



Inhaltsverzeichnis

1	Benutzer Referenz	4
1.1	Systemanforderungen	4
1.2	Installation	4
1.2.1	Einzelplatz	4
1.2.2	Netzwerk	4
1.2.3	Programmstart	4
1.3	Registrierung	5
1.4	Programmeinstellungen	5
1.5	Verfahrensbibliothek (VB)	8
1.5.1	Hinzufügen/Bearbeiten	9
1.5.2	Löschen	9
1.5.3	Sicherheit	9
1.5.4	Info	10
1.5.5	Drucken	10
1.5.6	Word	10
1.5.7	Importieren/Exportieren	10
1.5.8	Gruppieren	11
1.5.9	Referenzen	11
1.5.10	Info-Dateien	11
1.6	Daten	11
1.6.1	Anforderungs-Daten-Schnittstelle (ADS)	11
1.6.2	Arbeiten ohne ADS	12
1.7	Log und List	12
2	Technische Referenz	18
2.1	PIAFStat Sprachelemente	18
2.1.1	Grundlagen	18
2.1.2	Analyse-Merkmale	18
2.1.3	Klassifizierungs-Merkmale	19
2.1.4	Besondere Klassifizierungs-Merkmale	20
2.1.5	Besondere Platzhalter	20
2.1.6	Optionen	20
2.1.7	Blöcke	21
2.1.8	Funktionen	23
2.1.9	Layout	24
2.2	ASCII-Schnittstelle (ADS)	26
2.2.1	VERSUCHE.TXT	27
2.2.2	VMERKMAL.TXT	28
2.2.3	VFAKTOR.TXT	29
2.2.4	VSTUFE.TXT	29
2.2.5	VDATEN.TXT	30
2.2.6	VDATENSE.TXT	31
2.2.7	VGD.TXT	32
2.2.8	VUPDATE.INI	32
2.3	Programmcode	36
2.3.1	SAS	36

2.3.2	R	36
2.3.3	Globale Informationen	37
2.3.4	Versuche	38
2.3.4.1	SAS	38
2.3.4.2	R	39
2.3.5	Merkmale	40
2.3.5.1	SAS	40
2.3.5.1	R	43
2.3.6	Faktoren	50
2.3.6.1	SAS	50
2.3.6.1	R	51
2.3.7	Grenzdifferenzen	52
2.3.7.1	SAS	52
2.3.7.1	R	52
2.3.8	Zuordnung der K-Merkmale	53
2.3.8.1	SAS	53
2.3.8.1	R	54
2.3.9	Zuordnung aller Programmvariablen	54
2.3.9.1	SAS	54
2.3.9.1	R	55
2.3.10	Daten	56
2.3.10.1	SAS	56
2.3.10.1	R	58
2.3.11	SE-Daten	64
2.3.11.1	SAS	64
2.3.11.1	R	66
2.4	Programmkonfiguration	71
2.4.1	Programmparameter (/INI, /BIB und /RUN)	71
2.4.2	Ini-Datei (Programmparameter /INI)	72
2.4.3	Windows-Registry	73
2.4.4	Verfahrensabfrage (Programmparameter /BIB)	73
2.4.5	Hintergrundmodus (Programmparameter /RUN)	74
2.4.6	Logbuch Sitzung PIAFStat	74
2.4.7	Logbuch Eingaben des Dialoges SAS/R ausführen	75
2.4.8	Temporäres Verzeichnis für PIAFStat	75
2.5	Dateistruktur	75
2.6	Netzwerkbetrieb	77
2.7	Speicherort der Standard-Verfahrensbibliothek	77
2.8	Hintergrundmodus – ExitCodes	78
3	Anlagen	79
3.1	Fehlermeldungen bei Programmstart	79
3.2	Tastenkombinationen (Hotkey oder short cut)	79
3.3	MDAC - Microsoft Data Access Components	80
3.4	Boolsche Algebra	80
3.4.1	Theorie	80
3.4.2	Anwendung	81
3.5	SAS	82
3.5.1	Exit Code	82
3.5.2	Auszug aus der SAS-Dokumentation „SAS Companion for the Microsoft Windows Environment“	82

3.5.3	SAS-Kommandos für zusätzliche Parameter	83
3.6	R.....	84
3.7	Layout-Schriftattribute	85
3.8	Format für Word-Dateiname.....	86
3.9	Optionen der Verfahren.....	87
3.10	Programmbeispiele	89
3.10.1	Verwendung von Funktionen.....	89
3.10.2	Stapelverarbeitung (Batch-Jobs).....	93

© 2025

BioMath GmbH

1 Benutzer Referenz

PIAFStat stellt über eine Verfahrensbibliothek die Verbindung zwischen ASCII-Daten, insbesondere von Versuchsdaten der PIAF-Schnittstelle, und SAS bzw. R her. Die einzelnen Verfahren der Bibliothek sind SAS- bzw. R-Programme, die um syntaktische Elemente erweitert sind. Durch die Einführung dieser Elemente können die Verfahren unabhängig von den konkreten Daten entwickelt werden. Erst durch die Zuordnung der Daten zu einem Verfahren wird durch Bindung der Datenvariablen mit den syntaktischen Elementen ein lauffähiges SAS- bzw. R-Programm generiert. PIAFStat steuert danach das SAS- bzw. R-System und stellt die Ergebnisse für eine weitere Bearbeitung zur Verfügung.

In den ersten Abschnitten des Handbuches wird die Bedienung von PIAFStat kurz vorgestellt. Die *Technische Referenz* enthält die wichtigsten Informationen über die Konfiguration und Einbettung von PIAFStat in das Windowssystem sowie die Beschreibung der einzelnen Schnittstellen und der zusätzlichen Syntaxelemente.

1.1 Systemanforderungen

PIAFStat wurde für Windows7/ Windows10 als Desktopanwendung entwickelt und benötigt bei einer vollständigen Installation ca. 15 MB freien Festplattenspeicher. Es werden 128 MB RAM und eine Grafikauflösung von 800 x 600 Pixel empfohlen. Weiterhin wird die Microsoft Data Access-Komponente (MDAC), die bereits in Windows integriert ist, benötigt.

1.2 Installation

Die Installation benötigt Administratorrechte und kann sowohl als Einzelplatz als auch als Netzwerkversion erfolgen. Nach der Installation ist PIAFStat zu starten, um einerseits die Standard-Verfahrensbibliothek zu erzeugen und andererseits, um die Programmeinstellungen für die Schnittstellen zu konfigurieren. Nutzer, die mit PIAFStat arbeiten, benötigen Schreib- und Leserechte sowohl im Programmverzeichnis als auch in dessen Unterverzeichnissen.

1.2.1 Einzelplatz

Für die Installation führen Sie das Installationsprogramm auf der CD-ROM aus und folgen den Anweisungen auf dem Bildschirm. Die Installation muss auf einer lokalen Festplatte erfolgen. Rufen Sie danach über das Startmenü das Programm PIAFStat auf und konfigurieren Sie die Schnittstellen im Dialog „Optionen“.

1.2.2 Netzwerk

Für die Netzwerkinstallation lesen Sie die Anweisungen auf der CD-ROM.

1.2.3 Programmstart

Um PIAFStat im Dialogmodus auszuführen, sollte der Aufruf über das Startmenü erfolgen. Durch die Angabe von Parametern ist es möglich, die Art der Programmausführung (Dialogmodus, Hintergrundmodus, Einstellungen übernehmen usw.) festzulegen. Ausführliche Informationen dazu finden Sie in der *Technischen Referenz*.

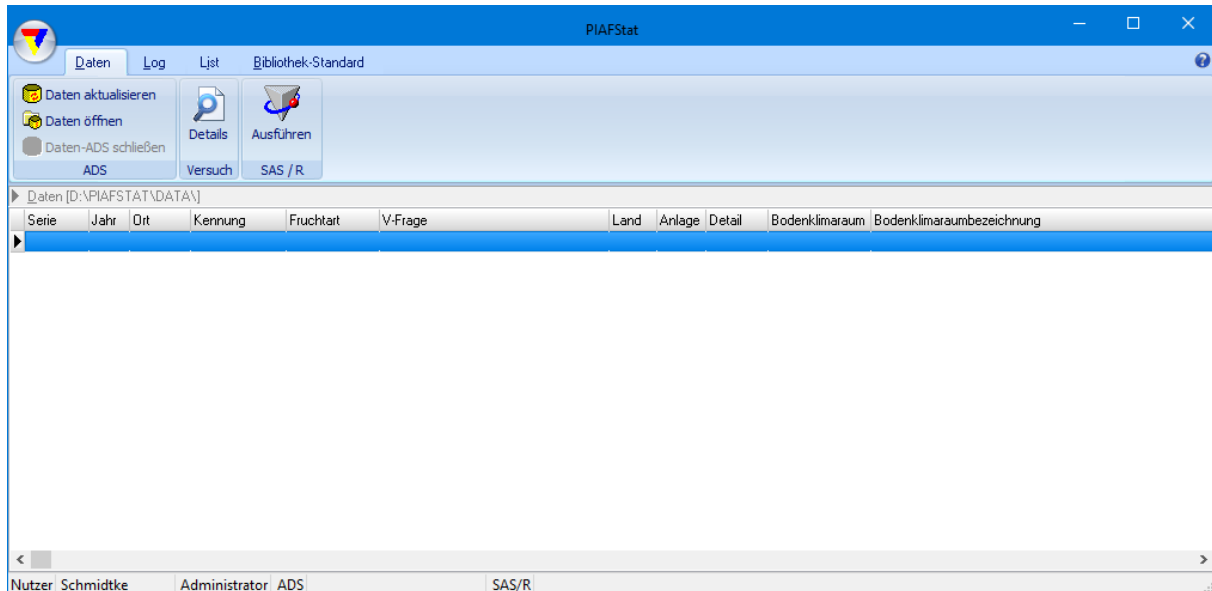
Die Liste der wichtigsten Fehlermeldungen, die beim Programmstart angezeigt werden können, ist in der Anlage 3.1 aufgeführt.

1.3 Registrierung

Nach der Installation muss PIAFStat registriert werden. Benutzen Sie das entsprechende Formular „Registrierung.pdf“.

1.4 Programmeinstellungen

Nach dem ersten Programmstart wird das Hauptfenster geöffnet.



Die einzelnen Arbeitsbereiche:

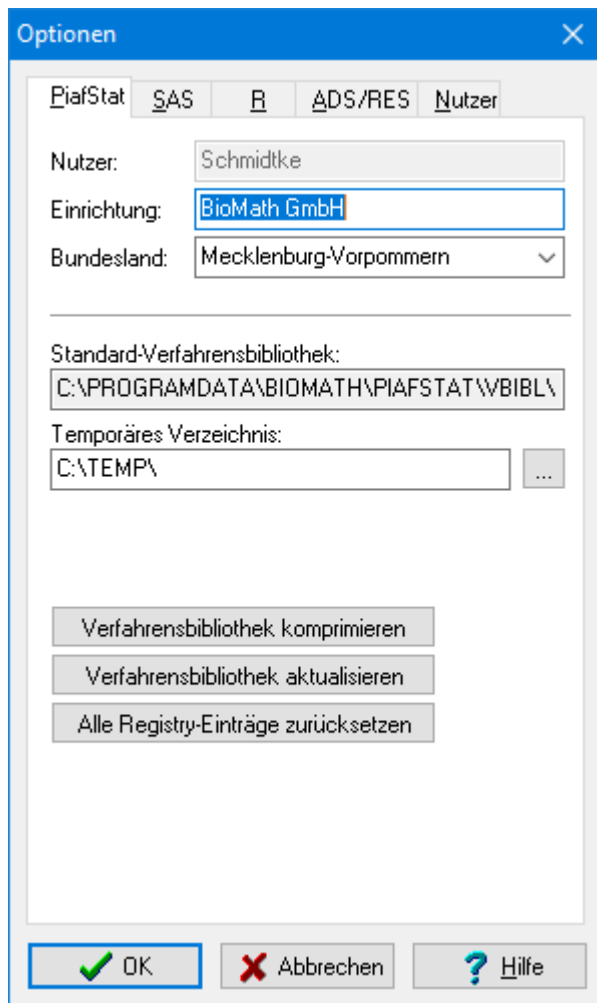
- Daten
- Log / Lis und
- Bibliothek (Standard-VB oder Nutzer-VB)

wählen Sie über die Reiter der Multifunktionsleiste aus. Für jeden Arbeitsbereich werden in der Multifunktionsleiste die spezifischen Aktionen bereitgestellt. Wichtige Aktionen werden Ihnen im Bearbeitungsbereich zusätzlich über die rechte Maustaste angeboten. Weiterhin stehen Tastenkombinationen (siehe Anhang) zur Verfügung.

In der unteren Statuszeile sind Informationen über den aktuellen Nutzer, den Zustand der ADS und der Kommunikation mit dem SAS- bzw. R-System zu finden.

PIAFStat verwendet häufig für die Anzeige ein Datengitter (z.B. im Arbeitsbereich Daten). Innerhalb eines solchen Gitters kann sowohl die Breite als auch die Reihenfolge der Spalten verändert werden. In der Regel gelten solche Änderungen nur für die aktuelle Sitzung. Die Einstellungen für die Gitter im Arbeitsbereich Daten und in der Detailanzeige der Merkmale werden über die aktuelle Sitzung hinaus gespeichert.

Für die Programmkonfiguration wählen Sie mit Hilfe des PIAFStat-Knopfes (oben links) den Menüpunkt „Optionen“:



Auf der Registerkarte „PIAFStat“ haben Sie die Möglichkeit, den Firmennamen und das Bundesland zu vergeben. Diese Angaben werden bei der Bearbeitung von statistischen Verfahren automatisch verwendet. Das Verzeichnis der Standard-Verfahrensbibliothek wird durch das System vergeben. Das temporäre Verzeichnis kann eingestellt werden und dient der Zwischenspeicherung von Informationen. Mit dem Knopf "Verfahrensbibliothek komprimieren" wird die Größe der Datei der aktivierten Verfahrensbibliothek reduziert. Während der Komprimierung ist der Zugriff auf die Verfahrensbibliothek gesperrt. Mit dem Knopf „Verfahrensbibliothek aktualisieren“ werden alle Tabellen der aktuellen Verfahrensbibliothek erneut ausgelesen. Diese Funktionalität ist nur bei Problemen im Netzwerkbetrieb bei der gemeinsamen Nutzung der Bibliotheken relevant. Mit dem Knopf "Alle Registry-Einträge zurücksetzen" werden die aktuellen Einstellungen von PIAFStat erneut in der Registry gespeichert und alle nicht benötigten PIAFStat-Registryeinträge gelöscht.

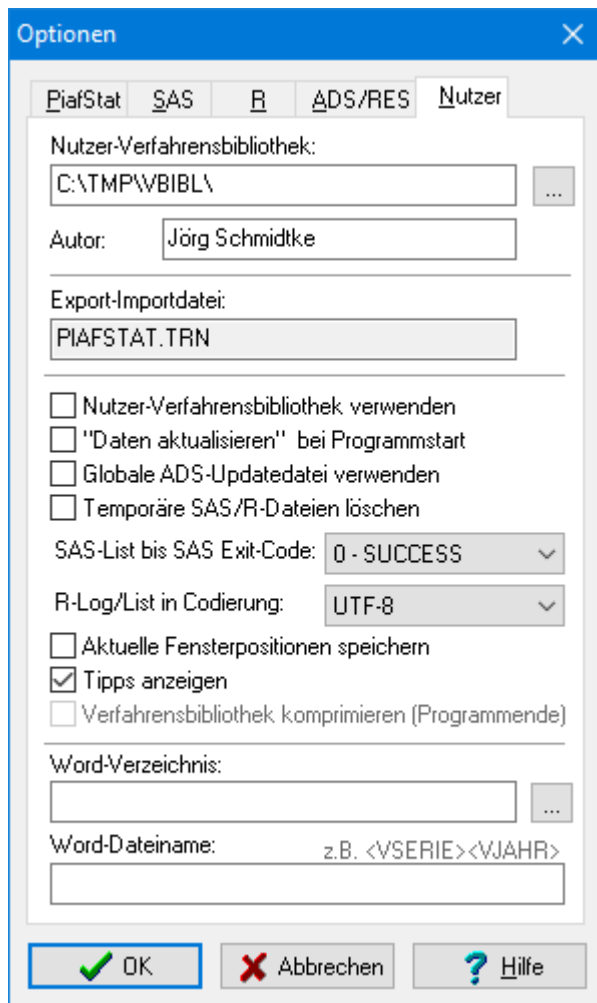
Auf der Registerkarte „SAS“ ist der SAS-Programmpfad festzulegen. Weiterhin können zusätzliche Parameter für den SAS-Aufruf angegeben werden¹.

Auf der Registerkarte „R“ ist der R-Programmpfad festzulegen. Weiterhin können zusätzliche Parameter für den R-Aufruf angegeben werden².

Auf der Registerkarte „ADS/RES“ sind die notwendigen Verzeichnisse und die global zu verwendende ADS-Updatedatei festzulegen. Benutzen Sie dort gegebenenfalls die entsprechenden Auswahlschalter.

¹ In der Anlage 3.5 ist der Verweis zur SAS-Dokumentation beschrieben.

² In der Anlage 3.6 ist der Verweis zur R-Dokumentation beschrieben.



Im Register „Nutzer“ kann das Verzeichnis für die Nutzer-Verfahrensbibliothek festgelegt werden.

Der Eintrag „Autor“ ist eine Voreinstellung für neue statistische Verfahren bzw. Funktionen. (Wird der Verfahrensbibliothek ein neues Verfahren oder eine neue Funktion hinzugefügt, so wird als Autor dieser Eintrag voreingestellt.)

Der Dateiname für den Export bzw. Import von statistischen Verfahren bzw. Funktionen ist durch das System festgelegt.

Weiterhin haben Sie auf dieser Registerkarte die Möglichkeiten:

- die Nutzer-Verfahrensbibliothek zu aktivieren
- das automatische Auslesen der ADS beim Programmstart zu aktivieren
- die Verwendung der globalen ADS-Updatedatei zu aktivieren
- das Löschen aller temporären SAS-Dateien vorzunehmen
- nachdem SAS-Aufruf das List-Fenster in Abhängigkeit vom SAS-Exitcode anzuzeigen
- die Text-Codierung der R-Ausgabedateien festzulegen
- das Speichern der Fensterpositionen zu aktivieren
- die Anzeigetipps in der Multifunktionsleiste zu aktivieren
- die aktivierte Verfahrensbibliothek automatisch bei Programmende zu komprimieren
- ein temporäres Word-Verzeichnis³ und
- ein Format für den Word-Dateinamen⁴ festzulegen

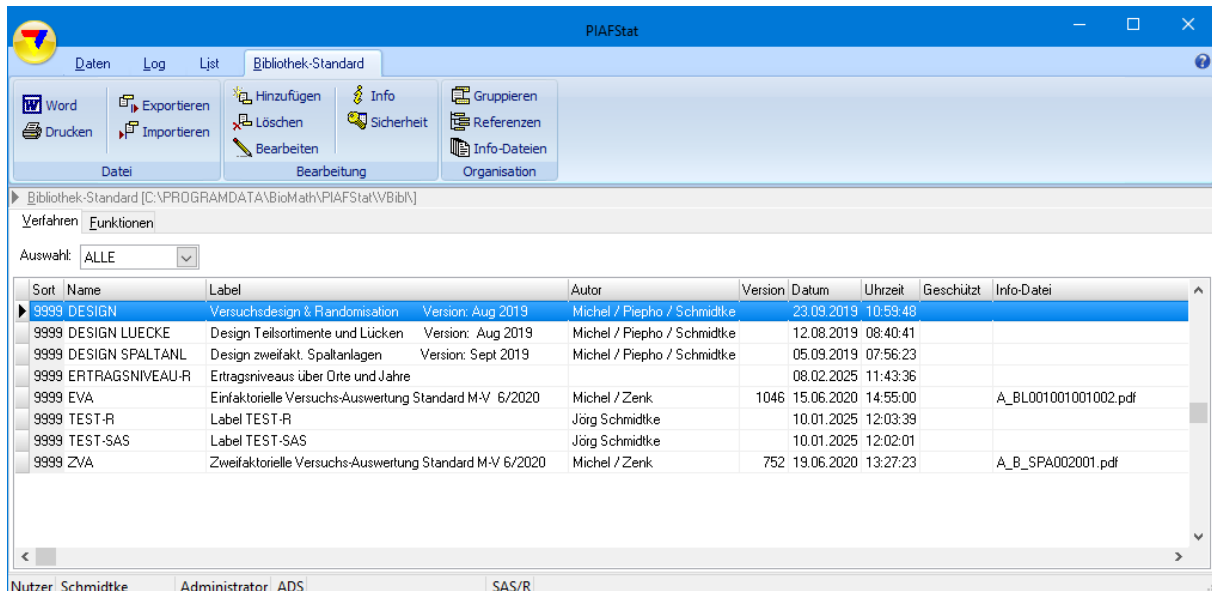
Alle Optionen werden in der Windows-Registry gespeichert.

³ Bei der Übernahme der Log/Lis-Ausgaben in die Textverarbeitung Word erfolgt der Datenaustausch in dem angegebenen Verzeichnis. Ist die Verzeichnisangabe leer, wird das nutzerspezifische temporäre Verzeichnis verwendet.

⁴ In der Anlage 3.8 sind die Formate und die Syntax beschrieben.

1.5 Verfahrensbibliothek (VB)

Die Verfahrensbibliothek besteht einerseits aus statistischen Verfahren und andererseits aus Funktionen. Innerhalb von Verfahren können nur Funktionen der gleichen Bibliothek verwendet werden. PIAFStat kann die Standard- oder Nutzer-Verfahrensbibliothek verwenden. Das Verzeichnis der aktuell verwendeten Bibliothek wird in der oberen Statuszeile angezeigt. Die Beschriftung des Reiters der Multifunktionsleiste gibt den Typ der Bibliothek an: Bibliothek-Standard oder Bibliothek-Nutzer.



Der Wechsel zwischen den Bibliotheken erfolgt im Dialog Optionen. Legen Sie dort ein Verzeichnis für die Nutzer-VB fest. Durch Aktivieren des Feldes „Nutzer-Verfahrensbibliothek verwenden“ können Sie auf diese Nutzer-VB zugreifen. Ist in dem angegebenen Verzeichnis noch keine Nutzer-VB vorhanden, erzeugt PIAFStat eine leere VB.

Ein Vollzugriff auf die Standard-VB ist nur durch administrative Rechte möglich (Ausführliche Hinweise finden Sie auf der CD-ROM). Auf die Nutzer-VB kann immer mit Vollzugriff gearbeitet werden.

Für die Verwaltung der statistischen Verfahren bzw. Funktionen wählen Sie zunächst den Arbeitsbereich „Bibliothek“. Dort werden alle verfügbaren Verfahren bzw. Funktionen mit „Name“, „Label“, „Autor“, „Version“, „Datum“, „Uhrzeit“ und „Info-Datei“ (nur bei Verfahren) aufgelistet. Die Versionsnummer bezieht sich nur auf den Programmcode. Die Reihenfolge der Auflistung ist durch das Feld „Sort“ festgelegt. Die Auflistung der Verfahren kann über die Auswahl einer Verfahrensgruppe eingeschränkt werden. Über die rechte Maustaste können ausführliche Informationen zu dem markierten Verfahren abgerufen werden.

Für das Navigieren bzw. für die Bearbeitung von Verfahren und Funktionen stehen die Tasten ENTER, EINFÜGEN und ENTFERNEN zur Verfügung.

1.5.1 Hinzufügen/Bearbeiten

Wählen Sie „Hinzufügen“ um ein neues Verfahren bzw. Funktion in die Bibliothek aufzunehmen. Der Doppelklick auf ein Verfahren startet automatisch die Bearbeitung. Innerhalb der Bearbeitung vergeben Sie einen eindeutigen Namen mit einer maximalen Länge von 16 Zeichen. Durch das Feld „Sort“ wird die Anzeigereihenfolge festgelegt. Kurzinformationen hinterlegen Sie im Feld „Label“. Für den eigentlichen Programmcode einschließlich Informationstext (Info-Datei) stehen die entsprechenden Register mit Werkzeugleiste zur Verfügung. Die zugeordneten Info-Dateien müssen sich im Unterverzeichnis „Info“ der aktivierten Verfahrensbibliothek befinden. Die Organisation dieser Dateien kann über den Knopf Info-Dateien erfolgen.

Unter „Details“ können zusätzliche Informationen (Ersteller, letzter Bearbeiter, Version und Autor) eingesehen bzw. festgelegt werden.

Ist dem Verfahren bzw. Funktion ein Kennwort zugewiesen worden, so ist im unteren Bereich des Dialogfensters ein Schloss sichtbar. Die Bearbeitung des Programmodes ist in diesem Falle nur durch eine Entriegelung (geöffnetes Schloss) möglich. Dazu klicken Sie auf das geschlossene Schloss und geben das entsprechende Kennwort ein.

Nach der Entriegelung sind in der aktuellen PIAFStat-Sitzung das Verfahren bzw. Funktion immer bearbeitbar. In einer neuen Sitzung müssen Sie das Verfahren bzw. Funktion zur Bearbeitung erneut entriegeln. Die Entriegelung kann auch über die Sicherheitseinstellungen vorgenommen werden.

Einstellung der Programmiersprache

Auf der Registerkarte „Details“ ist die verwendete Programmiersprache (SAS oder R) des Verfahrens festzulegen. Abhängig von der festgelegten Programmiersprache wird zur Ausführung des generierten Programmcodes automatisch SAS oder R aufgerufen.

Einstellungen für den Hintergrundmodus

Auf der Registerkarte „Details“ können sie die Verwendung des Verfahrens für den Hintergrundmodus festlegen. Ist für ein Verfahren der Modus Einzelversuche und auch der Modus Auswertungsserie aktiviert so werden die Daten bezüglich ihrer vorliegenden Struktur (Einzelversuche / Auswertungsserie) behandelt.

1.5.2 Löschen

Das markierte Verfahren bzw. Funktion wird nach einer Sicherheitsabfrage gelöscht.

1.5.3 Sicherheit

Durch die Vergabe eines Kennwortes ist ein Schutz vor Änderung eines Verfahrens bzw. Funktion gewährleistet.

Vorgehensweise: Erstellen Sie zunächst ein Verfahren bzw. Funktion und vergeben Sie dann ein Kennwort und aktivieren gegebenenfalls die Option „Code DEKLARATION nicht sperren“. Daraufhin wird ein geöffnetes Schloss (Code nicht gesperrt) eingeblendet. In der aktuellen PIAFStat-Sitzung ist dann das Verfahren bzw. Funktion immer bearbeitbar. In einer neuen Sitzung müssen Sie das Verfahren bzw. Funktion zur Bearbeitung erneut entriegeln. Dieses kann sowohl in den Sicherheitseinstellungen als auch im Bearbeitungsdialog erfolgen.

Der Export und Import von Verfahren bzw. Funktionen erfolgt einschließlich Kennwortschutz.

1.5.4 Info

Die dem markierten Verfahren zugeordnete Info-Datei wird mit der Standard-Windowsanwendung geöffnet.

1.5.5 Drucken

Das markierte Verfahren bzw. Funktion wird nach dem Druckerauswahldialog ausgedruckt.

1.5.6 Word

Das markierte Verfahren bzw. Funktion wird zur weiteren Verarbeitung in formatierter Form an die Textverarbeitung Word übergeben und angezeigt.

1.5.7 Importieren/Exportieren

Der Import und Export von Verfahren und Funktionen erfolgt über die Dateien, die in den Systemeinstellungen auf der Registerkarte „Nutzer“ festgelegt wurden. Sind einem Verfahren Info-Dateien zugeordnet, so werden diese automatisch in den Export bzw. Import eingeschlossen. Bei einem Export werden die Dateien grundsätzlich neu erstellt, d.h. bereits vorhandene exportierte Verfahren und Funktionen werden überschrieben. Um exportierte Verfahren und Funktionen anderen Nutzern zur Verfügung zu stellen, müssen Sie die Exportdatei PIAFSTAT.TRN weitergeben.

Verfahren:

Wählen Sie den Reiter „Verfahren“ und schränken Sie gegebenenfalls die Liste der Verfahren bezüglich einer Gruppierung ein. Benutzen Sie danach den entsprechenden Schalter, um den Import bzw. Export durchzuführen. Legen Sie in der folgenden Dialogbox das Import- bzw. Exportverzeichnis fest. In einer weiteren Dialogbox werden, je nach Einschränkung bezüglich der Gruppierung, die verfügbaren Verfahren gelistet. Markieren Sie in dieser Liste die entsprechenden Verfahren durch einen Haken. Sind bei den markierten Verfahren Funktionen eingebunden, so werden diese in der folgenden Dialogbox gelistet und für den Export bzw. Import markiert. Sollte bei einem Import ein Konflikt hinsichtlich des Namens (sowohl bei den Verfahren als auch bei den Funktionen) auftreten, werden Sie zu einer Umbenennung oder Ersetzung aufgefordert. Mit der Option „Ausgewählte Verfahren in der Bibliothek ersetzen“ werden bereits in der Bibliothek vorhandene Verfahren ohne Nachfrage ersetzt.

Funktionen:

Wählen Sie den Reiter „Funktionen“ und benutzen Sie die entsprechenden Schalter, um den Import bzw. Export durchzuführen. Legen Sie in der folgenden Dialogbox das Import- bzw. Exportverzeichnis fest. In einer weiteren Dialogbox werden alle verfügbaren Funktionen gelistet. Markieren Sie in dieser Liste die entsprechenden Funktionen durch einen Haken. Sollte bei einem Import ein Konflikt hinsichtlich des Namens auftreten, werden Sie zu einer Umbenennung oder Ersetzung aufgefordert. Mit der Option „Ausgewählte Funktionen in der Bibliothek ersetzen“ werden bereits in der Bibliothek vorhandene Funktionen ohne Nachfrage ersetzt.

Anmerkung:

Neu importierte Verfahren werden automatisch nur der Gruppe ALLE zugeordnet, d.h. ist im Arbeitsbereich „Bibliothek“ eine andere Gruppe zum Listing ausgewählt, so werden die neu importierten Verfahren bezüglich dieser Auswahl nicht angezeigt.

1.5.8 Gruppieren

Alle statistischen Verfahren sind in Gruppen eingeteilt. Jedes Verfahren kann dabei mehreren Gruppen angehören. Das System selbst verwaltet die Gruppe ALLE, d.h. innerhalb dieser Gruppe befinden sich immer alle verfügbaren Verfahren. Über den Schalter „Gruppieren“ rufen Sie die Gruppenverwaltung auf. Dort können weitere Gruppen definiert und entsprechende Verfahren zugeordnet werden. Analog den Verfahren sind die Felder „Gruppenname“, „Label“ und „Sort“ zu handhaben. Beachten Sie, dass bei einer späteren Verfahrensauswahl erst die Gruppe und danach das Verfahren gewählt werden.

1.5.9 Referenzen

Verfahren können Funktionen verwenden. Die zu verwendenden Funktionen werden im Deklarationsteil der Verfahren referenziert. Einen vollständigen Überblick über alle Verfahren, Funktionen und deren Referenzen erhalten Sie über den Knopf „Referenzen“. Funktionen, die nicht referenziert sind, können hier aus der Verfahrensbibliothek gelöscht werden.

1.5.10 Info-Dateien

Jedem Verfahren kann eine Info-Datei zugeordnet werden. Diese Dateien sind im Unterverzeichnis „Info“ der Verfahrensbibliothek gespeichert. Einen Überblick über alle Info-Dateien und allen Verfahrenszuordnungen erhalten Sie über den Knopf „Info-Dateien“.

1.6 Daten**1.6.1 Anforderungs-Daten-Schnittstelle (ADS)**

Die Daten stehen in der ADS (Anforderungsdatenschnittstelle) zur Verfügung. Nach dem Programmstart hat PIAFStat Zugriff auf das voreingestellte ADS-Verzeichnis (siehe Optionen). Immer wenn Sie „Daten aktualisieren“ wählen greift PIAFStat auf dieses ADS-Verzeichnis zu. Über „Daten öffnen“ kann die ADS aus einem anderen Verzeichnis gelesen werden. (Der zugehörige Schalter kann entsprechend umgeschaltet werden.)

Nachdem Auslesen der ADS, werden im Bearbeitungsbereich die Versuche oder A-Serie (zuzüglich Versuche) mit deren wesentlichen Eigenschaften angezeigt. Die ausführlichen Versuchsinformationen wie Merkmale, Faktoren, Daten, SE-Daten und Grenzdifferenzen können Sie über den Schalter „Details“ bzw. die rechte Maustaste abrufen. Fehlermeldungen, die hinsichtlich der Datenaktualisierung auftreten, werden in der Statuszeile angezeigt.

Wurde in der ADS eine A-Serie erkannt so wird diese in einem zusätzlichen Datengitter angezeigt. Die Aktionen „SAS/R-ausführen“ bzw. „Details“ beziehen sich dann auf das jeweils markierte Element (A-Serie, Einzelversuch und ausgewählte Einzelversuche).

Der Knopf „Daten aktualisieren“ importiert die ADS aus dem Verzeichnis, dass unter „Optionen“/„ADS-Verzeichnis“ festgelegt wurde. Mit Hilfe von „Daten öffnen...“ kann die ADS auch aus einem anderen Verzeichnis importiert werden.

Das Auslesen der ADS ist mit Hilfe einer Updatedatei so steuerbar, dass Änderungen wie z.B. die Umbenennung von Variablenamen, die Sortierreihenfolge der Merkmale usw. möglich sind. Diese Updatedatei kann sowohl im Verzeichnis der ADS als auch global (d.h. nutzbar für alle ADS-Verzeichnisse) bereitgestellt werden. Die globale ADS-Updatedatei wird nur dann für die Steuerung verwendet, wenn ihre Verwendung aktiviert und im ADS-Verzeichnis keine Updatedatei vorhanden ist.

Anmerkung:

In Abschnitt 2.2.8 ist die Struktur der Updatedatei beschrieben.

1.6.2 Arbeiten ohne ADS

Je nach Ablaufstruktur der Verfahren werden ADS-Daten nicht immer benötigt. Vor der Abarbeitung solcher Verfahren sollten bereits geöffnete ADS-Daten geschlossen werden. Wähle Sie dazu den Schalter „Daten-ADS schließen“.

Anmerkung:

Bei einer geschlossenen ADS werden dem abzuarbeitenden Verfahren keine Daten zur Verfügung gestellt. Im Dialogmodus kann der Nutzer im Dialog SAS/R-ausführen ein ADS-Verzeichnis für das Verfahren festlegen.

1.7 Log und List

Um eine statistische Analyse durchzuführen, wählen Sie zunächst die zu analysierenden Daten aus und benutzen den Schalter „SAS/R ausführen“. Je nach importierter ADS gibt es die folgenden Möglichkeiten der Datenauswahl:

Einzelversuch:

Nur ein Einzelversuch kann für die Analyse ausgewählt werden.

Versuchsserie:

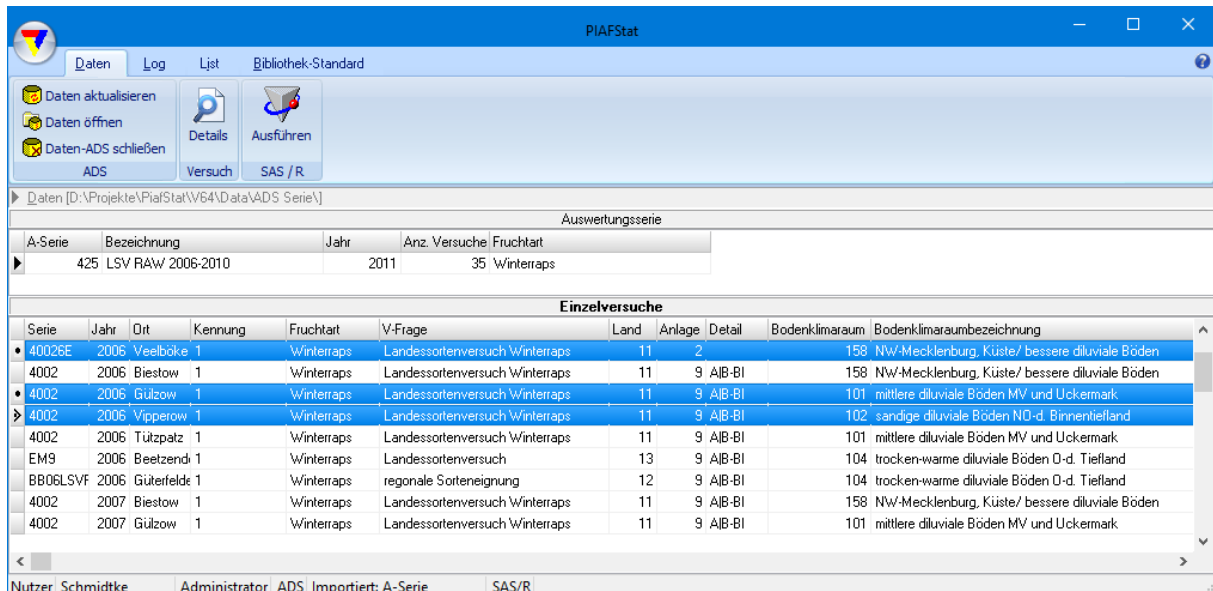
Serie oder
ein Einzelversuch oder
mehrere Einzelversuche können für die Analyse ausgewählt werden.

Die Auswahl von mehreren Einzelversuchen erfolgt in der Regel mit Hilfe der Maus, bei betätigter STRGTASTE. Sollen aufeinanderfolgende Versuche ausgewählt werden, klicken Sie auf den ersten Versuch, halten die UMSCHALTTASTE fest und benutzen die NACH-OBEN-TASTE bzw. NACH-UNTEN-TASTE.

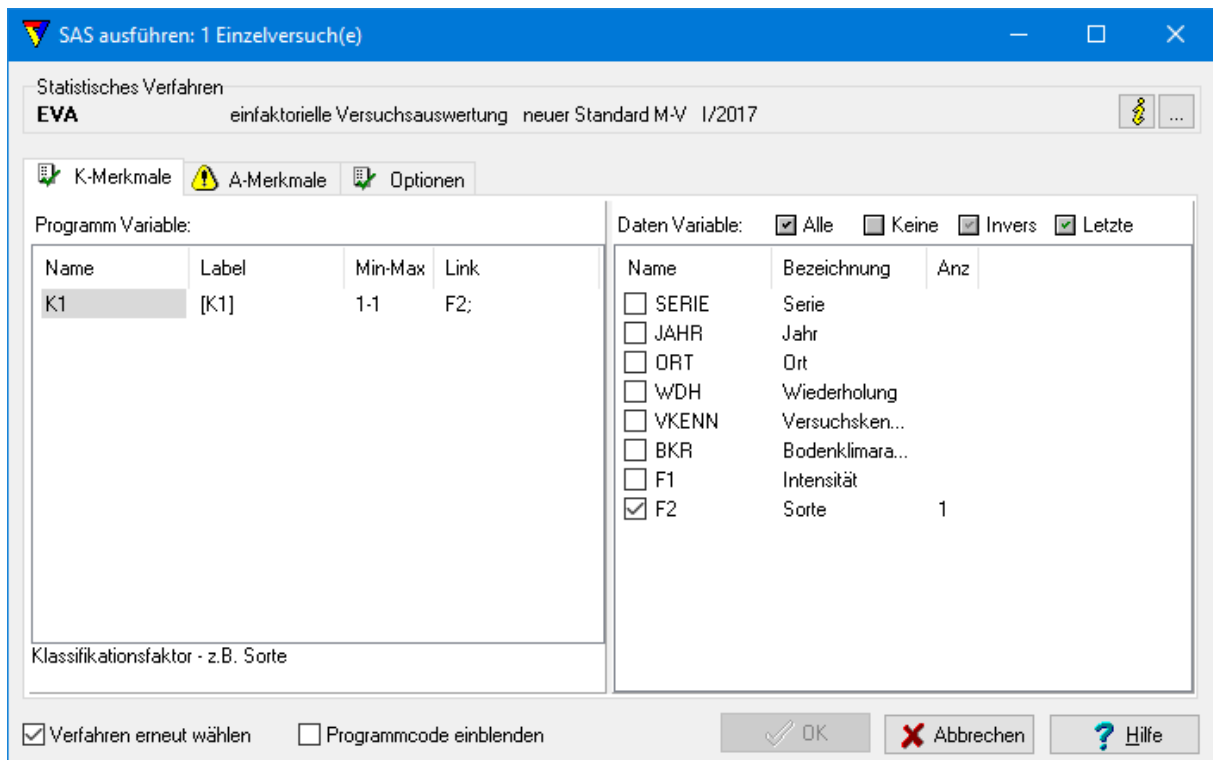
Über die rechte Maustaste im Titelbereich „Einzelversuche“ kann das Auswahlmenü:



angezeigt werden.



In der folgenden Dialogbox selektieren Sie eine (Verfahrens-) Gruppe und ein zugehöriges Verfahren. In einem weiteren Dialog werden alle Bindungen zwischen den Versuchsdaten und dem statistischen Verfahren festgelegt.



Im Titel des Dialoges wird angezeigt, ob es sich um ein SAS- oder R-Verfahren handelt.

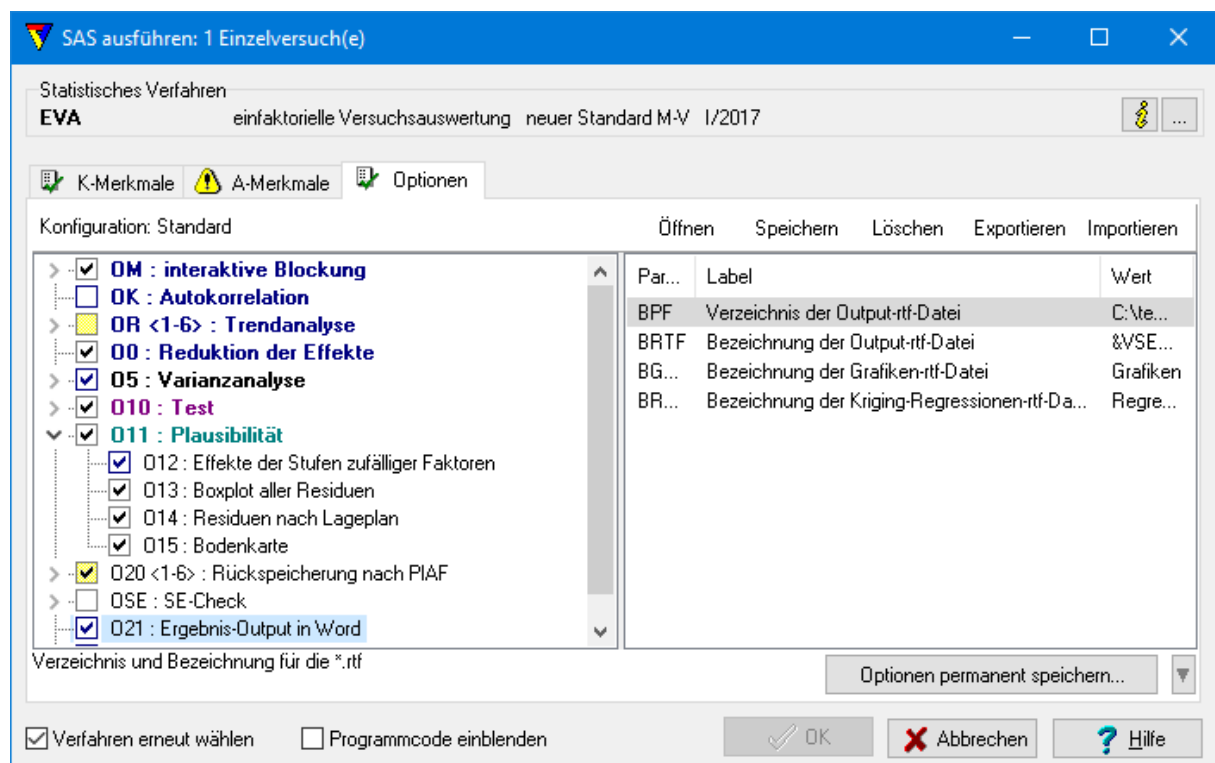
Die Anzeige der Registerseiten ist abhängig von den Versuchsdaten und dem gewählten Verfahren. Im oberen Bereich wird das gewählte statistische Verfahren einschließlich Label angezeigt. Mit dem „I“-Schalter können ausführliche Informationen zu dem Verfahren eingeblendet werden. Der rechte Schalter dient zur Auswahl eines Verfahrens.

Auf den Registerkarten der Merkmale werden auf der linken Seite jeweils alle Programm-Variablen, die im gewählten statistischen Verfahren definiert wurden, gelistet. In der Spalte „Min-Max“ ist die Anzahl der notwendigen Bindungen zu den Daten-Variablen sichtbar. Die Bezeichnung „INF“ bedeutet *beliebig viele* Bindungen. In der Spalte „Link“ werden die aktuellen Bindungen zu den Daten-Variablen angezeigt. Im unteren Bereich der Liste ist der ausführliche Informationstext der markierten Programm-Variablen eingeblendet. Auf der rechten Seite sind alle Daten-Variablen gelistet. Die Spalte „Anz“ gibt Auskunft über die Anzahl der Programm-Variablen, die an diese Daten-Variable aktuell gebunden sind. Um eine Programm-Variable an Daten-Variablen zu binden, markieren Sie zunächst die entsprechende Programm-Variable und „haken“ danach die Daten-Variablen an.

Mit den Schaltern „Alle“, „Keine“, „Invers“ und „Letzte“ können gleichzeitig mehrere Daten-Variable an- bzw. abgehakt werden. Der Schalter „Letzte“ hakt diejenigen Daten-Variablen an bzw. ab die beim letzten Aufruf des Dialoges ausgewählt wurden.

Wurden in dem statistischen Verfahren optionale Elemente definiert, so finden Sie diese Angaben auf dem entsprechenden Register.

Auf der Registerkarte Optionen können Sie zusätzliche Einstellung wählen.



Bei notwendigen Unteroptionen ist die Anzahl in Klammern angegeben. Im obigen Beispiel ist für die Option „OR“ mindestens eine der sechs Unteroption notwendig (<1-6>). Verfügt eine Option über Parameter so werden diese auf der rechten Seite zur Bearbeitung bereitgestellt. Verwenden Sie den Schalter „Bearbeiten“ oder führen Sie einen Doppelklick auf den Eintrag aus.

Anmerkung

Wurden nicht alle geforderten Bindungen getroffen, so ist der "OK-Knopf inaktiv und der entsprechende Reiter der Registerkarte mit einem Warnschild gekennzeichnet. Wurde das Kontrollfeld "Verfahren erneut wählen" markiert, so wird beim nächsten Aufruf die Auswahl eines statistischen Verfahrens angeboten.

Für die Einsicht in den generierten Code bzw. der deklarierten Elemente, aktivieren Sie das Kontrollfeld "Programmcode einblenden".

Nachdem alle Zuordnungen getroffen wurden, können Sie den automatischen SAS- bzw. R-Run über den Schalter „OK“ starten. Während des Runs wird PIAFStat minimiert und in der Taskleiste angezeigt. Nach Beendigung dieses Runs werden (je nach Exit Code) die Log- und die Listdateien in den entsprechenden Arbeitsbereichen geladen und angezeigt. Diese Ergebnisse können in die Zwischenablage kopiert, in eine Datei gespeichert, gedruckt, an Word übergeben und nach Begriffen durchsucht werden. Verwenden Sie dazu die untere Schalterleiste bzw. das Menü der rechten Maustaste.

Anmerkung

Bei jedem Run werden alle temporären Dateien überschrieben. In den Programmooptionen können Sie festlegen, ob diese temporären Dateien nach Programmende automatisch gelöscht werden.

Ausführliche Informationen zum SAS Exit Code sind in der Anlage 3.5.1 und zum R Exit Code in Anlage 3.6 dokumentiert.

Schaltergruppe Konfiguration der Optionen

Die Konfiguration der Optionen können für das ausgewählte Verfahren gespeichert, wieder abgerufen und gelöscht werden. Die Speicherung erfolgt nutzerspezifisch jeweils unter einer Bezeichnung. Es können mehrere Konfigurationen je Verfahren abgespeichert werden.

Ferner können gespeicherte Konfigurationen eines Verfahrens in eine Datei exportiert und aus einer Datei importiert werden. Die Dateinamen haben die folgende Struktur: PIAFSTAT-<Name des Verfahrens>.KOP. Z.B. für das Verfahren EVA lautet der Dateiname PIAFSTAT-EVA.KOP.

Benutzen Sie für den Export bzw. Import den entsprechenden Schalter. Legen Sie in der folgenden Dialogbox das Export- bzw. Importverzeichnis fest. In einer weiteren Dialogbox werden die verfügbaren Konfigurationen der Optionen gelistet. Markieren Sie in dieser Liste die gewünschten Konfigurationen durch einen Haken und beenden Sie den Dialog mit OK.

Beim Export wird die Datei grundsätzlich neu erstellt, d.h. in der Datei befinden sich nur die Konfigurationen, die exportiert wurden. Beim Import werden bereits vorhandene Konfigurationen überschrieben. Optionen mit Verzeichnisangaben können optional übernommen werden.

Optionen speichern

Je nach Nutzerrecht, Art der Verfahrensbibliothek und ausgewähltes Verfahren können die Options-Werte unterschiedlich gespeichert werden. Die zur Verfügung stehenden Möglichkeiten der Speicherung sind über den Schalter "Optionen ... speichern..." und über das Aufklappenmenü dieses Schalters zu erreichen. Es gibt die folgenden Möglichkeiten:

Optionen permanent speichern:

Die Options-Werte werden direkt in der Verfahrensbibliothek gespeichert.

Optionen nutzerspezifisch speichern:

Die Options-Werte werden nur für den Nutzer gespeichert. Die Options-Werte in der Verfahrensbibliothek werden dabei nicht verändert.

Optionen auf Standard zurücksetzen:

Die nutzerspezifisch gespeicherten Options-Werte werden gelöscht und auf die Options-Werte der Verfahrensbibliothek gesetzt.

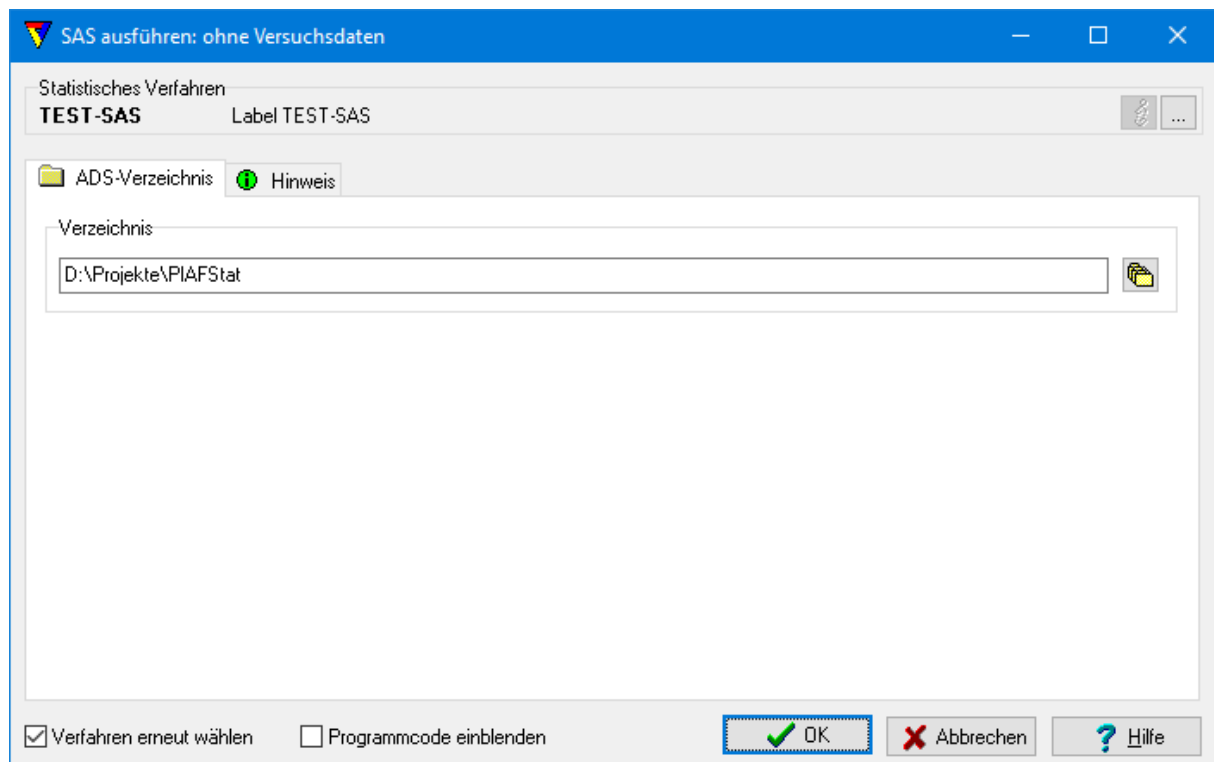
Nach Auswahl eines Verfahrens werden die Optionen entweder durch die nutzerspezifisch gespeicherten Options-Werte oder den Options-Werten der Verfahrensbibliothek initialisiert (siehe Anlage 3.9).

Die Einstellung nutzerspezifischer Options-Werte wird auf der Registerkarte Optionen durch einen Informationstext vermerkt.

Ausführliche Informationen zur Speicherung der Optionen sind in der Anlage 3.93.9 dokumentiert.

Ohne ADS

Bei einer geschlossenen ADS kann der Nutzer im Dialog SAS/R-ausführen ein ADS-Verzeichnis zur Verarbeitung für das Verfahren festlegen.



2 Technische Referenz

Die folgenden Abschnitte enthalten ausführliche Informationen über die Syntaxregeln, den Aufbau der ASCII-Schnittstellen sowie eine Beschreibung der Programmkonfiguration einschließlich Dateistruktur.

2.1 *PIAFStat* Sprachelemente

2.1.1 Grundlagen

Jedes Verfahren wird in die Abschnitte **Deklaration** und **Programm** unterteilt. Im Abschnitt **Deklaration** werden alle Platzhalter einschließlich deren Eigenschaften definiert. Im Abschnitt **Programm** wird das eigentliche SAS- bzw. R-Programm hinterlegt. Die in das Programm eingefügten Elemente (Verfahrens-Parameter) sind definierte Platzhalter, welche einerseits SAS- bzw. R-Statement konform platziert (z.B. für Variable) werden, aber andererseits auch ein Erzeugen von weiteren SAS- bzw. R-Statements u. a. (z.B. Label) bewirken können. Diese Platzhalter sind in eckige Klammern „[...]“ eingeschlossen. Es gibt verschiedene Typen:

- A: Analyse-Merkmale
- K: Klassifizierungs-Merkmale
- O: Optionale Abschnitte
- B: Blöcke
- P: Funktionen
- L: Layout

und besondere Platzhalter

Anmerkung:

Die Wörter „Deklaration“ und „Programm“ müssen jeweils einzeln in einer Zeile stehen, damit sie als Schlüsselwörter identifiziert werden können. Der Abschnitt **Deklaration** muss sich vor dem Abschnitt **Programm** befinden. Innerhalb des Deklarationsabschnittes werden Zeilen, die mit einem Stern (*) beginnen, als Kommentarzeilen behandelt. Die abschließende runde Klammer eines Deklarationselementes muss (bis auf Leerzeichen) das letzte Zeichen der aktuellen Zeile sein. Das Komma dient als Trennzeichen für die Parameter der Deklarationselemente. Innerhalb dieser Parameter sind Zeilenumbrüche erlaubt. Für den Parameter Programm-Code sind prinzipiell alle alphanumerischen Zeichen erlaubt.

2.1.2 Analyse-Merkmale

Der Platzhalter (im SAS- bzw. R-Programm) für ein Analyse-Merkmal ist **[Ax]**, wobei x eine Zeichenkette bestehend aus den Zeichen (A...Z;0..9;_) ist.

Die Deklaration ist:

Ax(L:Label, I:Info, M:Min-Max, Feldname = Feldinhalt | Feldname = Feldinhalt,...,
Feldname = Feldinhalt | Feldname = Feldinhalt, AUTO)

Alle Parameter innerhalb der Klammer sind optional.

Parameter	Bedeutung
L	Kurzer Informationstext
I	Informationstext
M	Min-Max: Anzahl der Datenbindung (Min>=0, Max<=512) Für Max kann der Bezeichner „INF“ (beliebig) verwendet werden Voreinstellung: 1-INF
Feldname	Feldname in der Datei VMerkmal.txt
Feldinhalt	Inhalt des Feldes Feldname (siehe Anmerkung)
AUTO	Sind alle A- und K-Merkmale mit AUTO deklariert und alle vorangestellten Bedingungen erfüllt, so erfolgt in der Nutzerschnittstelle ein Hinweis über die automatische Bindung an die Datenvariablen.

Anmerkung:

Die Mindestforderung eines Platzhalter vom Typ Analyse-Merkmal ist **Ax()**. Ist das letzte Zeichen von „Feldinhalt“ ein Stern (*) so wird dieses Zeichen als Platzhalterzeichen interpretiert. Der Stern (*) kann auch als alleiniger Platzhalter verwendet werden.

Beispiel:

A1(L:Ertragsmerkmal, I:Parzellenwerte aus 2f Spaltanlage, Name= ERT* , AUTO)

Es werden alle A-Merkmale deren Name mit „ERT“ beginnen automatisch ausgewählt.

A2(L:Ertragsmerkmal, I:Parzellenwerte aus 2f Spaltanlage, FELD1= * , AUTO)

Es werden alle A-Merkmale die im FELD1 ein Eintrag besitzt automatisch ausgewählt.

Das Trennzeichen „," (Komma) für „Feldname = Feldinhalt“ bedeutet eine UND-, das Zeichen „|“ (Strich) eine ODER-Verknüpfung. Als NOT-Operator dient das voranstellte Zeichen „!“ (Ausrufezeichen) vor dem „Feldinhalt“.

Weitere Informationen bezüglich der Operatoren befinden sich im Anhang.

2.1.3 Klassifizierungs-Merkmale

Der Platzhalter (im SAS- bzw. R-Programm) für ein Klassifizierungs-Merkmal ist **[Kx]**, wobei x eine Zeichenkette bestehend aus den Zeichen (A...Z;0..9;_) ist.

Die Deklaration ist:

Kx(L:Label, I:Info, M:Min-Max, Bezeichnung,...,Bezeichnung, AUTO)

Alle Parameter innerhalb der Klammer sind optional.

Parameter	Bedeutung
L	Kurzer Informationstext
I	Informationstext
M	Min-Max: Anzahl der Datenbindung (Min>=0, Max<=512) Für Max kann der Bezeichner „INF“ (beliebig) verwendet werden Voreinstellung: 1-INF
Bezeichnung	Feldinhalt des Feldes „BEZ“ der Datei Vfaktor.txt oder ein besonderer Platzhalter (WDH, ORT, SERIE, JAHR, VKENN und Fi)
AUTO	Sind alle A- und K-Merkmale mit AUTO deklariert und alle vorangestellten Bedingungen erfüllt, so erfolgt in der

Parameter	Bedeutung
	Nutzerschnittstelle ein Hinweis über die automatische Bindung an die Datenvariablen.

Anmerkung:

Die Mindestforderung eines Platzhalters vom Typ Klassifizierungs-Merkmal ist **Kx()**. Bei der automatischen Zuordnung wird der Wert der Bezeichnung zunächst mit der Bezeichnung des K-Merkmals verglichen. Konnte diesbezüglich keine Zuordnung gefunden werden erfolgt ein weiterer Vergleich mit dem Namen des K-Merkmals.

2.1.4 Besondere Klassifizierungs-Merkmale

Die K-Merkmale WDH, ORT, SERIE, JAHR, VKENN, BKR und Fi können im SAS- bzw. R-Programm direkt verwendet werden. Für diese Variablen ist kein Platzhalter notwendig.

Variable	Datei
WDH	Vdaten.txt
ORT	Feld „ORT“ aus Versuche.txt
SERIE	Vdaten.txt
JAHR	Vdaten.txt
VKENN	Versuche.txt
BKR	Versuche.txt
Fi	Feld „Fi“ aus Vdaten.txt

2.1.5 Besondere Platzhalter

Die besonderen Platzhalter sind vordefiniert (Sie brauchen im Abschnitt Deklaration nicht aufgeführt werden).

Platzhalter	Bedeutung
[A-ALL]	Alle Analysemerkmale
[K-ALL]	Alle Klassifizierungsmerkmale (WDH, ORT, SERIE, JAHR, VKENN, BKR und alle Faktoren Fi)
[F-ALL]	Alle Faktoren Fi (aus der Datei Vfaktor.txt)

Anmerkung:

Werden die besonderen Platzhalter verwendet, so können an dieser Stelle keine weiteren Einschränkungen durch den Nutzer erfolgen.

2.1.6 Optionen

Werden optionale Elemente im Dialogmodus verwendet, so hat der Nutzer immer die Möglichkeit, diese Optionen zu aktivieren bzw. zu deaktivieren. Im Hintergrundmodus werden grundsätzlich die Voreinstellungen verwendet.

Der Platzhalter (im SAS- bzw. R-Programm) für ein optionales Element ist **[Ox]** wobei x eine Zeichenkette bestehend aus den Zeichen (A...Z;0..9;_) ist.

Die Deklaration ist:

Ox(L:Label, I:Info, M:Min-Max, C:Programmcode, IsON:Optionales Element = ON | OFF +...+ Optionales Element = ON | OFF, IsOFF:Optionales Element = ON | OFF +...+ Optionales Element = ON | OFF, ON | OFF | FIX)

Parameter	Bedeutung
L	Kurzer Informationstext
I	Informationstext
M	Min-Max: Anzahl der Unteroptionen (Min>=0, Max<=512) Für Max kann der Bezeichner „INF“ (beliebig) verwendet werden Voreinstellung: 0-INF
C	Programmcode Innerhalb des Programmcodes können alle Platzhalter verwendet werden (A-Merkmale, K-Merkmale, Optionen, Blöcke u. Besondere Platzhalter)
IsON	Abhängigkeiten zu weiteren optionalen Elementen, wenn die Option aktiviert ist
IsOFF	Abhängigkeiten zu weiteren optionalen Elementen, wenn die Option deaktiviert ist
ON OFF FIX	Die Option ist eingeschaltet (ON), nicht eingeschaltet (OFF) oder immer eingeschaltet (FIX). Die Voreinstellung ist ON und nicht explizit anzugeben.

Anmerkung:

Die Mindestforderung eines Platzhalters vom Typ Option ist **Ox**(C:Programm-Code).

Besonderheit in R:

Geht der Programmcode über mehrere Zeilen, so ist jeder R-Befehl mit einem Semikolon abzuschließen.

Handhabung der Optionen

Innerhalb des Programmcodes einer Option können Optionsplatzhalter verwendet werden. Es sind nur zwei Hierarchiestufen möglich: Option – Unteroption. Die Reihenfolge der Deklaration ist beliebig. Unteroptionen brauchen bei Mehrfachverwendung nur einmal deklariert werden.

Ist der Deklarationsteil eines Verfahrens nicht gesperrt, so kann die Voreinstellung ON bzw. OFF durch einen autorisierten Nutzer⁵ im Dialog „SAS/R-ausführen“ geändert werden. Die Voreinstellung FIX ist mit Hilfe des Dialoges nicht änderbar.

Beispiel

O13(L:explorativer Ansatz, C:outpredictm, IsON: O14=OFF + O12=OFF, ON)

O14(L:vollständiges Modell, C:outpredict, IsON: O13=OFF, OFF)

Für eine Datensortierung werden die Option O13 bzw. O14 verwendet. Die Defaulteinstellung ist: O13 aktiviert und O14 deaktiviert (siehe ON bzw. OFF). In der Option O13 ist durch **IsOn: O14=OFF + O12=OFF** festgelegt, dass die Optionen O14 und O12 automatisch deaktiviert werden, wenn O13 aktiviert wurde. In der Option O14 ist durch **IsOn: O13=OFF** festgelegt, dass die Option O13 automatisch deaktiviert wird, wenn O14 aktiviert wurde.

2.1.7 Blöcke

Der Platzhalter (im SAS- bzw. R-Programm) für einen Block ist **[Bx]** wobei x eine Zeichenkette bestehend aus den Zeichen (A...Z;0..9;_) ist.

Die Deklaration ist:

⁵ Autorisiert ist ein Nutzer dann, wenn er auch manuell den Deklarationsteil ändern kann.

Bx(L:Label, I:Info, C:Programmcode,R: Range, OPTION oder OPTIONINT oder OPTIONNUM oder OPTIONLIST oder OPTIONPATH)

Parameter	Bedeutung
L	Kurzer Informationstext
I	Informationstext
C	Programmcode Innerhalb des Programmcodes können die Platzhalter: A-Merkmale, K-Merkmale, Blöcke u. Besondere Platzhalter verwendet werden
R	Legt den Eingabebereich für die Parameter OPTIONINT (ganze Zahl) und OPTIONNUM (numerische Zahl) oder die Auswahlliste für den Parameter OPTIONLIST fest Min-Max: Eingabebereich für OPTIONINT und OPTIONNUM (Das Dezimaltrennzeichen für numerische Zahlen ist der Punkt.) Eintag1/Eintrag2/ usw.: Auswahlliste für OPTIONLIST
OPTION	Block wird im Programmcode einer Option verwendet und ist änderbar
OPTIONINT	Block wird im Programmcode einer Option verwendet und ist änderbar. Im zugehörigen Eingabedialog wird geprüft, ob die Eingabe eine gültige ganze Zahl ist. Mit dem optional zu benutzenden Parameter R kann der Eingabebereich für die ganze Zahl festgelegt werden.
OPTIONNUM	Block wird im Programmcode einer Option verwendet und ist änderbar. Im zugehörigen Eingabedialog wird geprüft, ob die Eingabe eine gültige numerische Zahl ist. Mit dem optional zu benutzenden Parameter R kann der Eingabebereich für die numerische Zahl festgelegt werden. Das Dezimaltrennzeichen numerischer Zahlen ist der Punkt.
OPTIONLIST	Block wird im Programmcode einer Option verwendet und im zugehörigen Eingabedialog kann aus einer Liste ein Eintrag ausgewählt werden. Die Einträge der Liste werden mit dem Parameter R festgelegt.
OPTIONPATH	Block wird im Programmcode einer Option verwendet und ist änderbar. Im zugehörigen Eingabedialog für die Änderung des Optionsinhaltes steht dem Nutzer ein Verzeichnis-Auswahldialog zur Verfügung.

Anmerkung:

Die Mindestforderung eines Platzhalters vom Typ Block ist **Bx**(C: Programmcode).

Besonderheit in R:

Geht der Programmcode über mehrere Zeilen, so ist jeder R-Befehl mit einem Semikolon abzuschließen.

Handhabung der Blöcke

Blöcke sollten immer dann eingesetzt werden, wenn **gleicher Programmcode mehrmals** zum Einsatz kommen soll. Blöcke werden intern in „globale“ und „optionale“ Blöcke unterteilt. Globale Blöcke haben „fixen“ Programmcode und können im Programmcode und im Optionen-Programmcode verwendet werden.

Durch den Zusatz OPTION, OPTIONINT, OPTIONNUM, OPTIONLIST oder OPTIONPATH werden optionale Blöcke deklariert. Diese Blöcke finden **nur** innerhalb des Optionen-Programmcodes Verwendung und können zur Laufzeit durch den Nutzer verändert werden. (Für den Nutzer stellen sich diese Blöcke als „zusätzliche Parameter“ innerhalb der Optionen dar.) Autorisierte Nutzer⁶ können diese optionalen Blöcke gegebenenfalls im Dialog „SAS/R ausführen“ ändern. Bei Mehrfachverwendung von optionalen Blöcken brauchen diese nur einmal deklariert werden.

Beispiel1 – optionaler Block zur Eingabe einer ganzen Zahl

B01(L:Eingabe einer ganzen Zahl, I:Gib eine ganze Zahl aus dem Bereich von 1 bis 10 ein., R:1-10, C:5, OPTIONINT)

Beispiel2 – optionaler Block zur Eingabe einer numerischen Zahl

B02(L:Eingabe einer numerischen Zahl, I:Gib eine numerische Zahl aus dem Bereich von -0.125 bis 35 ein., C:4.37, R:-0.125-35, OPTIONNUM)

Beispiel3 – optionaler Block zur Auswahl eines Listenelementes

B03(L:Auswahl einer Farbe, I:Wähle eine Farbe aus:, C:Gelb, R:Rot/Gelb/Grün, OPTIONLIST)

Beispiel3 – optionaler Block zur Auswahl eines Verzeichnisses

B04(L:Eingabe eines Verzeichnisses, I:Wähle ein Verzeichnis aus:, C:Z:\Projekte, OPTIONPATH)

2.1.8 Funktionen

Der Platzhalter für eine Funktion ist **[Px]**, wobei x eine Zeichenkette bestehend aus den Zeichen (A...Z;0..9;_) ist.

Die Deklaration ist:

Px(L:Label, I:Info, F:Funktionsname, Averf = Afkt, Kverf = Kfkt, Obez = Ofkt, Bbez = Bfkt)

Parameter	Bedeutung
L	Kurzer Informationstext
I	Informationstext
F	Name der Funktion in der aktuellen Bibliothek
Averf = Afkt,	Zuweisung der A- und K-Merkmale
Kverf = Kfkt	(verf=Platzhalter des Verfahrens, fkt=Platzhalter in der Funktion)
Obez = Ofkt,	Umbenennung der Optionen und Blöcke
Bbez = Bfkt	(bez=neue Bezeichnung im Verfahren, fkt=Platzhalter in der Funktion)

⁶ Autorisiert ist ein Nutzer dann, wenn er auch manuell den Deklarationsteil ändern kann.

Anmerkung:

Die Mindestforderung eines Platzhalters vom Typ Funktion ist **Px**(F:Funktionsname).

Handhabung der Funktionen

Innerhalb einer Bibliothek können die statistischen Verfahren Funktionen verwenden. Funktionen sind wiederverwertbare Programmteile die einerseits A- und K-Merkmale verarbeiten und andererseits in Abhängigkeit von Optionen und Blöcken stehen können. Funktionen können selbst keine weiteren Funktionen aufrufen.

Im Deklarationsteil des Verfahrens ist der Funktionsaufruf durch **Px**(...) festgelegt.

Innerhalb des Klammersausdruckes werden nur die zu verwendenden Elemente der Funktion deklariert. Die A- und K-Merkmale der Funktion werden durch zugewiesene A- und K-Merkmale der Verfahren überladen. Die in der Funktion deklarierten Optionen und Blöcke werden umbenannt und in den Deklarationsteil des Verfahrens überführt. Sie sind dann im Verfahren wie ‚normale‘ Optionen und Blöcke zu nutzen. (Nicht deklarierte Optionen und Blöcke werden auch nicht in den Deklarationsteil des Verfahrens überführt.)

Im Programmteil des Verfahrens wird der Platzhalter **[Px]** durch den Programmabschnitt der Funktion ersetzt.

Funktionen können nur Deklarationen enthalten, die dann auch nur im Deklarationsabschnitt des Verfahrens wirksam werden, sie können nur einen Programmteil mit Programmcode enthalten, der sich dann auch im Verfahren im Programmteil wiederfindet. Funktionen können aber auch beides enthalten. Es ist nicht zwingend erforderlich das alle Elemente einer Funktion im Verfahren aktiviert werden.

2.1.9 Layout

Der Platzhalter für ein Layout-Element ist **[Lx]** wobei x eine Zeichenkette bestehend aus den Zeichen (A...Z;0..9;_) ist.

Die Deklaration ist:

Lx(L:Label, I:Info, E:Element, FColor=Wert, FStyle=Wert, Sub=OPEN | CLOSE, TFormat=Wert, TFormattype=Wert, File=Dateiname, Pagebreak=ON | OFF, Totaltime=ON | OFF)

Parameter	Bedeutung
L	Kurzer Informationstext
I	Informationstext
E	Element für das ein Layout festgelegt wird (derzeit nur für Optionen)
FColor	Schriftfarbe (clBlack, clGray,...)*
FStyle	Schriftstiel (fsBold, fsItalic)
Sub	Unteroptionen werden erweitert oder geschlossen angezeigt (OPEN oder CLOSE)
TFormat	Nur gültig für Lis/Log-Datei Zeichenkette die formatiert wird
TFormattype	Nur gültig für Lis/Log-Datei Typ der Formatierung (WORD, LINE, LEFT, RIGHT, ...) *
File	Nur gültig für Lis/Log-Datei Dateiname (Format txt oder rtf) zur Speicherung im RES-Verzeichnis**

Parameter	Bedeutung
Pagebreak	Nur gültig für Lis/Log-Datei Der Seitenumbruch wird ein- oder ausgeschaltet (ON oder OFF). Die Voreinstellung ist ON.
Totaltime	Nur gültig für Lis-Datei von SAS Die Ergänzung der Lis-Datei um die in der SAS-Log-Datei ausgegebene benötigte Gesamtzeit wird ein- oder ausgeschaltet (ON oder OFF) Die Voreinstellung ist OFF.

* Die Layout-Schriftattribute für FColor sind in der Anlage 0 aufgelistet.

** Pfadangaben bei Dateinamen werden nicht berücksichtigt. Nicht aufgeführte Dateiextensionen werden als txt behandelt.

Anmerkung:

Die Mindestforderung eines Platzhalters vom Typ Layout ist **Lx**(E: Element). Im Dialogmodus stehen alle Formatierungen uneingeschränkt zur Verfügung. Im Hintergrundmodus sind die Formatierungen nur in den RTF-Dateien des Layout-Elementes wirksam.

Beispiel

LD(L:Bitte diese Option beachten, E: OPlan, FColor=RED, FStyle=BOLD, Sub=CLOSE)

Mit dem Layoutelement „LD“ werden die Anzeigeattribute der Option „OPlan“ festgelegt. Die Schrift ist rot und fett und die Unteroptionen sind zunächst nicht sichtbar, d.h. nicht aufgeklappt.

Beispiele für Lis-Datei

LBR(L: Abschaltung des Seitenumbruchs, E: LIS, Pagebreak=OFF)

In der Lis-Datei werden alle Seitenumbrüche entfernt.

LTT(L: SAS-Gesamtzeit, E: LIS, Totaltime=ON)

Der Lis-Datei von SAS wird die benötigte SAS-Gesamtzeit angefügt.

LUEB(L: Zeile formatieren, E: LIS, TFormat= Autoren, TFormattyp=LINE, FStyle=BOLD)

In der Lis-Datei werden alle Zeilen, in denen die Zeichenkette „Autoren“ auftritt, fett formatiert.

LALP(L: alpha formatieren, E: LIS, TFormat= alpha, TFormattyp=WORDRIGHT, FColor=RED, FStyle=BOLD)

In der Lis-Datei werden die Zeichenketten „alpha“ einschließlich aller Zeichen bis zum Zeilenende rot und fett formatiert.

LSAV(L: Lis-Datei speichern, E: LIS, File=report05.txt)

Die Lis-Datei wird unter dem Namen report05.txt als Textdatei gespeichert. Die Speicherung erfolgt im voreingestellten RES-Verzeichnis (siehe PIAFStat-Optionen).

2.2 ASCII-Schnittstelle (ADS)

Alle Importdateien befinden sich im gleichen Verzeichnis. PIAFStat benötigt einen einfachen Lesezugriff für dieses Verzeichnis. Die bereitgestellten Daten können mehrere Versuche (bzw. A-Serien) beinhalten. Ein „Satz“ besteht aus folgenden Textdateien:

Dateiname	Inhalt
VERSUCHE.TXT	Beschreibung der Versuche
VMERKMAL.TXT	Merkmale der Versuche
VFAKTOR.TXT	Faktoren der Versuche
VSTUFE.TXT	Stufen der Versuche
VDATEN.TXT	Daten der Versuche
VDATENSE.TXT*	SE-Daten der Versuche
VGD.TXT*	Grenzdifferenzen der Versuche
VUPDATE.INI*	Modifikation der ADS

*Datei ist in der ADS nicht zwingend erforderlich

VAR

Interner Variablenname in PIAF, PIAFStat, SAS bzw. R. Für PIAFStat sind die Primärschlüssel fett formatiert.

Formate

In jeder Zeile einer Text-Datei steht ein Datensatz und ist mit 0D_{Hex} und 0A_{Hex} abgeschlossen. Die einzelnen Felder der Datensätze sind durch „#“ (23_{Hex}) getrennt. Das letzte Feld eines Datensatzes ist mit „#“ (23_{Hex}) abgeschlossen. Fehlende Werte sind durch ein Leerzeichen (20_{Hex}) charakterisiert. Das Dezimaltrennzeichen für numerische Werte ist entweder das Komma (2C_{Hex}) oder der Punkt (2E_{Hex}).

Typ

- Ai: alphanumerisches Feld der Länge i (Buchstaben und Ziffern)
- I: Integer (32bit Integer im Bereich -2147483648 und 2147483647)
- N: Numerisch (Zahlen mit Vorzeichen und Dezimaltrennzeichen im Bereich -10³⁰⁷ und 10³⁰⁸)
- D: Datum mit dem Format TT.MM.JJJJ

PFL

Pflichtfeld, d.h. in der Schnittstellendatei muss für diese Variable ein Wert vom definierten Typ vorhanden sein.

Wertebereich (WB)

Falls es sich um Integer oder numerische Felder handelt, kann ein gültiger Wertebereich festgelegt werden.

Fkt

- F (funktional): Diese Variable benötigt PIAFStat zur Organisation der Schnittstellendaten.
- B (beschreibend): Diese Variable wird in PIAFStat nur angezeigt.
- N (Nein): Diese Variable wird **nicht** durch PIAFStat ausgewertet.

2.2.1 VERSUCHE.TXT

PIAF					PIAFStat			SAS/R**
Var	Typ	PFL	WB	Bemerkung	Fkt	Var	Typ	Var
V1_nr	N8,0	Ja	> 0	Versuchsnummer	F	VNR	I	VNR
D_nr	N5,0	Ja	> 0	Bestandsnummer	N	DNR	I	DNR
S_nr	A15	Ja		Serienbezeichnung	F/B	SNR	A15	SNR
Jahr	N4	Ja		Jahr	F/B	JAHR	I	JAHR
Land	N2	Ja		Landesnummer	B	LAND	I	LAND
Ort_Key	A5	Ja		OrtID (Primärschlüssel)	F	ORTID	A5	ORTID
Ort_Bez	A30			Ortsbezeichnung	B	ORTBEZ	A30	ORTBEZ
V_Kenn	A50	Ja		Versuchskennung	F/B	VKENN	A50	VKENN
Fag	A30	Ja	Liste	Fruchtartbezeichnung	B	FAG	A30	FAG
Vfrage	A60			Versuchsfrage	B	VFRAGE	A60	VFRAGE
Wdh	N2,0	Ja	> 1		F/B	WDH	I	WDH
Anlage	N4,0	Ja	Liste		F/B	ANLAGE	I	ANLAGE
Detail	A20	Ja			F	DETAIL	A20	DETAIL
Ort	A4			Ort (Benutzer orientiert)	B	ORT	A5	ORT
AS_Nr	N8,0			Nummer der A-Serie	F	ASNR	I	ASNR
AS_Bez	A30			Bezeichnung der A-Serie	B	ASNAME	A30	ASNAME
AS_Jahr	N4			Jahr der A-Serie	B	ASJAHR	I	ASJAHR
BKR	N4			Bodenklimaraum	B	BKR	I	BKR
BKR_BEZ	A50			Bodenklimaraumbezeichnung	B	BKRBEZ	A50	BKRBEZ

* SAS-Dataset VERS

** R-Dataframe VERS und VERS_LABEL

Anmerkung

Die Felder AS_Nr, AS_Bez und AS_Jahr sind nur bei exportierten A-Serien vorhanden. Die als A-Serie exportierten Versuche haben alle die gleiche AS_Nr. Die einer solchen A-Serie zugehörigen Faktoren und Merkmale werden durch den ersten Versuch festgelegt.

2.2.2 VMERKMAL.TXT

PIAF					PIAFStat			SAS'R''
Var	Typ	PFL	WB	Bemerkung	Fkt	Var	Typ	Var
V1_nr	N8,0	Ja		Versuchsnummer	F	VNR	I	VNR
M_nr	N3,0	Ja	> 0	Merkmalsnummer	F	MNR	I	MNR
Name	A6	Ja		Name	B	NAME	A32	NAME
Objekt	A7	Ja		Objekt	B	OBJEKT	A7	OBJEKT
Bezug	A6	Ja		Bezug	B	BEZUG	A6	BEZUG
Methode	A6	Ja		Methode	B	METHODE	A6	METHODE
Herk	A3	Ja		Herkunft	B	HERK	A3	HERK
Variante	N3,0	Ja	> 0	Variante	B	VARIANTE	I	VARIANTE
Bez	A40			Merkmalsbezeichnung (lang)	B	BEZ	A40	BEZ
Mi	N3,0	Ja	i>0	Nummer der Datenspalten in VDATEN.TXT	F	MI	I	MI
Schad	A6			Zielorganismus	B	SCHAD	A6	SCHAD
Name1	A8			Merkmalsbezeichnung (kurz)	B	NAME1	A8	NAME1
Name2	A16			Merkmalsbezeichnung (mittel)	B	NAME2	A16	NAME2
Typ	A1		N,C,D	Merkmalstyp	F	TYP	A1	TYP
Dezimal	N1			Anzahl Dezimalstellen	F	DEZIMAL	I	DEZIMAL
Gruppenart	A4			Gruppenart	F	GRPART	A4	GRPART
Datum				Datum der Merkmalerfassung	B	DATUM	A10	DATUM
BBCH von	N		<100	Entwicklungsstadium (von)	B	BBCHVON	I	BBCHVON
BBCH bis	N		<100	Entwicklungsstadium (bis)	B	BBCHBIS	I	BBCHBIS
flagmwadjust	A1		0,1	Flag für adjust. Mittelwerte	F	FLAG	A1	FLAG
N_Arithmetis ch	N3			minimaler Prozentsatz an Werten für erlaubte arithmetische Mittelwerte	F	NARITH	I	NARITH
FELD1	A10			Analysetyp	F/B	FELD1	A10	FELD1
FELD2	A10			Skalierung	F/B	FELD2	A10	FELD2
FELD3	A10			Zusatz-Info3	F/B	FELD3	A10	FELD3
...			
FELD10	A10			Zusatz-Info10	F/B	FELD10	A10	FELD10
Schnitt	N			Schnittnummer	B	SCHNITT	I	SCHNITT
Werte_Status	A4		Liste	{gesp, repr, orig}}	B	WSTATUS	A4	WSTATUS
Termin	A3			Terminkürzel	B	TERMIN	A6	TERMIN
EV_BBCH von	N			Entwicklungsstadium (von) des Versuchs	B	EBBCHVON	I	EBBCHVON
EV_BBCH bis	N			Entwicklungsstadium (Haupt) des Versuchs	B	EBBCH	I	EBBCH
EV_BBCH bis	N			Entwicklungsstadium (bis) des Versuchs	B	EBBCHBIS	I	EBBCHBIS
EV_Schnitt	N			Schnittnummer des Versuchs	B	ESCHNITT	I	ESCHNITT
Eingabeposition	N			Position des Merkmals bei der Eingabe	B	EINPOS	I	EINPOS
MinValue	N			Min-Wert	B	MINVALUE	N	MINVALUE
MaxValue	N			Max-Wert	B	MAXVALUE	N	MAXVALUE
Anz_StPro	N			Anzahl der Stichproben	B	ANZSTPRO	I	ANZSTPRO
AGG_StPro	A3			Stichprobenberechnungsfunktion	B	AGGSTPRO	A3	AGGSTPRO
Flag_PB	A1			Flag Prüfbericht	B	FLAGPB	A1	FLAGPB
Kultur_Nr	N			Kulturnummer	B	KULNR	I	KULNR
Kultur	A5			Kulturcode	B	KULTUR	A5	KULTUR
Kultur_Bez	A30			Kulturbezeichnung	B	KULBEZ	A30	KULBEZ
Sorte	A15			Sorte	B	SORTE	A15	SORTE
Schad_Bez	A50			Bezeichnung des Schadorganismus	B	SBEZ	A50	SBEZ
Schad_BezLat	A50			Lateinische Bezeichnung des Schadorganismus	B	SBEZLAT	A50	SBEZLAT
Schad_Klasse	A10			Klasse des Schadorganismus	B	SKLASSE	A10	SKLASSE

PIAF					PIAFStat			SAS/R**
Var	Typ	PFL	WB	Bemerkung	Fkt	Var	Typ	Var
M_KEY	A8			M_Key	B	MKEY	A8	MKEY
BSA_NAME	A5			BSA Name	B	BSANAME	A5	BSANAME
FORMEL	A250			Formel	B	FORMEL	A250	FORMEL

* SAS-Dataset MERK und MERKVERS

** R-Dataframe MERK und MERKVERS sowie MERK_LABEL und MERKVERS_LABEL

Anmerkung

Die Merkmale werden nach VNR und MI sortiert. Liegt die PIAF-Schnittstelle in Version 1.xx vor, so wird MI aus der Reihenfolge von MNR bestimmt.

2.2.3 VFAKTOR.TXT

PIAF					PIAFStat			SAS/R**
Var	Typ	PFL	WB	Bemerkung	Fkt	Var	Typ	Var
V1_nr	N8,0	Ja		Versuchsnummer	F	VNR	I	VNR
F_key	N3,0	Ja		Faktor	F/B	FKEY	I	FKEY
Bez	A30			Faktorbezeichnung	B	BEZFA	A30	BEZFA
Fvar1Bez	A23			Bez. 1.Faktorvariable	B	BEZ1	A23	BEZ1
...								
FvarnBez	A23			Bez. n. Faktorvariable	B	BEZn	A23	BEZn

* SAS-Dataset FAK

** R-Dataframe FAK und FAK_LABEL

Anmerkung

Die Anzahl der Faktorvariablen (je Versuch) ist durch die Trennzeichen der PIAF-Schnittstelle festgelegt.

2.2.4 VSTUFE.TXT

PIAF					PIAFStat			SAS/R**
Var	Typ	PFL	WB	Bemerkung	Fkt	Var	Typ	Var
V1_nr	N8,0	Ja		Versuchsnummer	F	VNR	I	VNR
F_key	N3,0	Ja		Faktor	F/B	FKEY	I	FKEY
St_nr	N3,0	Ja		Stufennummer	F/B	FI	I	FI
Bez	A100			Stufenbezeichnung	B	BEZST	A100	BEZST
Wert1	A10			Wert der 1. Faktorvariable	B	WERT1	A20	WERT1
Wertn	A10			Wert der n. Faktorvariable	B	WERTn	A20	WERTn
SGWert1	A10			Wert der Sortengruppe1	B	SGWert1	A20	SGWert1
SGWert2	A10			Wert der Sortengruppe2	B	SGWert2	A20	SGWert2

* SAS-Dataset FAK

** R-Dataframe FAK und FAK_LABEL

Anmerkung

Die Inhalte (Zeichenketten) der Felder Wert1 bis SGWert2 können beliebig viele Leerzeichen enthalten.

2.2.5 VDATEN.TXT

PIAF					PIAFStat			SAS/R
Var	Typ	PFL	WB	Bemerkung	Fkt	Var	Typ	Var
V1_nr	N8,0	Ja		Versuchsnummer	F	VNR	I	VNR
Jahr	N4	Ja		Jahr	F/B	JAHR	I	JAHR
S_nr	A15	Ja		Seriennummer	F/B	SERIE	A15	SERIE
Ort_Key	A5	Ja		Ort_Key	F	ORTID	A17	ORTID
St_nr1	s.o.	Ja		Stufennummer Faktor 1	F/B	F1	s.o.	F1
St_nr2	s.o.			Stufennummer Faktor 2	F/B	F2	s.o.	F2
...								
St_nr<n>	s.o.			Stufennummer Faktor <n>	F/B	Fn	s.o.	Fn
Wdh	s.o.	Ja		Wiederholung	F/B	WDH	s.o.	WDH
Kurzel	A15	Ja		Parzellenkürzel	N	KUERZEL	A15	KUERZEL
Block1	A8			benutzerdefinierte Angabe aus dem Bereich der Parzellen	B	BLOCK1	A8	BLOCK1
Block2	A8			benutzerdefinierte Angabe aus dem Bereich der Parzellen	B	BLOCK2	A8	BLOCK2
M1	A10	Ja		Merkmalswert 1	F	M1	A10	NAME_1
...				Siehe VMERKMAL.TXT	...			
Mn	A10	Ja		Merkmalswert n	F	Mn	A10	NAME_n
Parznr	N5			Parzellennummer	F	PARZNR	I	PARZNR
Reihe	N4			Zeilennr. (Lageplan)	F/B	REIHE	I	REIHE
Spalte	N4			Spaltennr. (Lageplan)	F/B	SPALTE	I	SPALTE
				ORT aus VERSUCHE.TXT		ORT	A5	ORT
				ORTBEZ aus VERSUCHE.TXT		ORTBEZ	A30	ORTBEZ
				VKENN aus VERSUCHE.TXT		VKENN	A50	VKENN
				LAND aus VERSUCHE.TXT		LAND	I	LAND
				BKR aus VERSUCHE.TXT		BKR	I	BKR
				BKRBEZ aus VERSUCHE.TXT		BKRBEZ	A50	BKRBEZ

* SAS-Dataset DATEN

** R-Dataframe DATEN und DATEN_LABEL

Anmerkung

In der Datenstruktur gibt es (dynamischen) Spalten für die Faktoren und für die Merkmale. Die Anzahl und die Reihenfolge dieser Spalten richten sich nach der Anzahl und der Sortierreihenfolge der Dateien „VFaktor.txt“ sowie nach der Nummer der Datenspalte (Mi) in „VMerkmal.txt“. Es sind maximal 255 Datenspalten in „VDaten.txt“ zulässig.

Die Variable „NAME“ wird durch das Feld „Name“ des Merkmals bestimmt. Gibt es mehrfach gleiche Merkmalsnamen so werden die Variablenamen um die Nummer Mi ergänzt.

2.2.6 VDATENSE.TXT

PIAF					PIAFStat			SAS/R
Var	Typ	PFL	WB	Bemerkung	Fkt	Var	Typ	Var
V1_nr	N8,0	Ja		Versuchsnummer	F	VNR	I	VNR
Jahr	N4	Ja		Jahr	F/B	JAHR	I	JAHR
S_nr	A15	Ja		Seriennummer	F/B	SERIE	A15	SERIE
Ort_Key	A5	Ja		Ort_Key	F	ORTID	A17	ORTID
St_nr1	s.o.	Ja		Stufennummer Faktor 1	F/B	F1	s.o.	F1
St_nr2	s.o.			Stufennummer Faktor 2	F/B	F2	s.o.	F2
...								
St_nr<n>	s.o.			Stufennummer Faktor <n>	F/B	Fn	s.o.	Fn
Wdh	s.o.	Ja		Wiederholung	F/B	WDH	s.o.	WDH
Kurzel	A15	Ja		Parzellenkürzel	N	KUERZEL	A15	KUERZEL
Block1	A8			benutzerdefinierte Angabe aus dem Bereich der Parzellen	B	BLOCK1	A8	BLOCK1
Block2	A8			benutzerdefinierte Angabe aus dem Bereich der Parzellen	B	BLOCK2	A8	BLOCK2
SE1	A10	Ja		SE-Wert Merkmal 1	F	M1	A10	NAME_1
...				Siehe VMERKMAL.TXT	...			
SEn	A10	Ja		SE-Wert Merkmal n	F	Mn	A10	NAME_n
Parznr	N5			Parzellennummer	F	PARZNR	I	PARZNR
Reihe	N4			Zeilennr. (Lageplan)	F/B	REIHE	I	REIHE
Spalte	N4			Spaltennr. (Lageplan)	F/B	SPALTE	I	SPALTE
				ORT aus VERSUCHE.TXT		ORT	A5	ORT
				ORTBEZ aus VERSUCHE.TXT		ORTBEZ	A30	ORTBEZ
				VKENN aus VERSUCHE.TXT		VKENN	A50	VKENN
				LAND aus VERSUCHE.TXT		LAND	I	LAND
				BKR aus VERSUCHE.TXT		BKR	I	BKR
				BKRBEZ aus VERSUCHE.TXT		BKRBEZ	A50	BKRBEZ

* SAS-Dataset DATENSE

** R-Dataframe DATENSE und DATENSE_LABEL

Anmerkung

In der Datenstruktur gibt es die (dynamischen) Spalten für die Faktoren und Merkmale. Die Anzahl und die Reihenfolge dieser Spalten richten sich nach der Anzahl und der Sortierreihenfolge der Dateien „VFaktor.txt“ sowie nach der Nummer der Datenspalte (Mi) in „VMerkmal.txt“. Es sind maximal 255 Datenspalten in „VDatenSE.txt“ zulässig.

Die Variable „NAME“ wird durch das Feld „Name“ des Merkmals bestimmt. Gibt es mehrfach gleiche Merkmalsnamen so werden die Variablenamen um die Nummer Mi ergänzt.

2.2.7 VGD.TXT

PIAF 4.4					PIAFStat			SAS/R
Var	Typ	PFL	WB	Bemerkung	Fkt	Var	Typ	Var
V1_nr	N8,0	Ja		Versuchsnummer	F	VNR	I	VNR
M_nr	N3,0	Ja	> 0	Merkmalsnummer	F	MNR	I	MNR
Name	A6	Ja		Name	B	NAME	A32	NAME
Objekt	A7	Ja		Objekt	B	OBJEKT	A7	OBJEKT
Bezug	A6	Ja		Bezug	B	BEZUG	A6	BEZUG
Methode	A6	Ja		Methode	B	METHODE	A6	METHODE
Herk	A3	Ja		Herkunft	B	HERK	A3	HERK
Variante	N3,0	Ja	> 0	Variante	B	VARIANTE	I	VARIANTE
GrArt	A4	Ja		Gruppenart	B	GRART	A4	GRART
StVergl	A4	Ja		Gruppenart Stufenvergleich	B	STVERGL	A4	STVERGL
GD5	N			Grenzdifferenz	B	GD5	N	GD5

* SAS-Dataset GRENZD

** R-Dataframe GRENZD und GRENZD_LABEL

2.2.8 VUPDATE.INI

Die Datei „VUPDATE.INI“ ist optional anwendbar. Sie dient der Anpassung der Funktionsweise des Schnittstellenimports. Diese Updatedatei kann sowohl im Verzeichnis der ADS als auch global (d.h. nutzbar für alle ADS-Verzeichnisse) bereitgestellt werden. Die globale ADS-Updatedatei wird nur dann für die Steuerung verwendet, wenn ihre Nutzung aktiviert und im ADS-Verzeichnis keine Updatedatei vorhanden ist.

1. Änderung der Merkmalsnamen

[Merkmale]

Name = Feldname1 + Feldname2 + Feldname3 + ...

Der Eintrag **Name** im Abschnitt **[Merkmale]** bezieht sich auf alle Merkmale der ADS. Mit dieser Zuweisung wird der Name des Merkmals durch den Inhalt der Felder Feldname1 + Feldname2 + ... zusammengesetzt. Ist die Zeichenkette Feldname in Doppelhochkommata eingeschlossen so wird die Zeichenkette ohne Hochkommata für die Bildung des Namens verwendet.

Beispiel:

[Merkmale]

Name = Name + Feld1 + "OD"

Der Name aller Merkmale wird um den Inhalt von FELD1 (=Analysetyp) und der Zeichenkette OD erweitert.

Anmerkung

Bei der Zuweisung des Inhalts der Merkmalsvariablen werden Umlaute, Leer-, Sonder- und Steuerzeichen durch eine Unterstrich (_) ersetzt.

2. Festlegung der Anzeigereihenfolge der Merkmale

[Merkmale]

Sort = Feldname1(order) + Feldname2(order) + ...

Mit der Zuweisung **Sort** = Feldname1 + Feldname2 + ... wird für den Dialogmodus die Anzeigereihenfolge der Merkmale festgelegt:

- Dialog „Sas/R ausführen“
- Detailanzeige der Versuchsdaten

Die Sortierung erfolgt in der Reihenfolge der Felder und innerhalb eines jeden Feldes nach der durch *order* festgelegten Reihenfolge:

ASC	aufsteigend
DESC	absteigend
ASCNUM	aufsteigend (numerisch)
DESCNUM	absteigend (numerisch)

Die Voreinstellung für *order* ist aufsteigend und optional, d.h. die Angabe von ASC ist nicht notwendig.

Beispiel:

```
[Merkmale]
Sort = BEZ(ASC) + Feld3(DESC)
```

Die Merkmale werden zunächst nach dem Feld "BEZ" aufsteigend und danach bezüglich des Feldes „Feld3“ absteigend sortiert.

Anmerkung:

Die Schlüsselwörter ASCNUM bzw. DESCNUM können nur auf alphanumerische Felder angewendet werden. Bei einer aufsteigenden Reihenfolge werden erst die Merkmale ohne Feldinhalte, danach die mit numerischen Inhalten und zuletzt die mit alphanumerischen Feldinhalten gelistet.

3. Festlegung der Label der Merkmale

```
[Merkmale]
Label = Feldname1 + Feldname2 + Feldname3 + ...
```

Der Eintrag **Label** im Abschnitt **[Merkmale]** bezieht sich auf alle Merkmale der ADS. Mit dieser Zuweisung wird der Label des Merkmals durch den Inhalt der Felder Feldname1 + Feldname2 + ... zusammengesetzt. Ist die Zeichenkette Feldname in Doppelhochkommata eingeschlossen, so wird die Zeichenkette ohne Hochkommata für die Bildung des Labels verwendet.

Beispiel:

```
[Merkmale]
Label = BEZ + Feld1 + "OD"
```

Der Label aller Merkmale wird aus dem Inhalt des Feldes BEZ, des Feldes FELD1 (=Analysetyp) und der Zeichenkette OD zusammengesetzt.

4. A-Merkmale als K-Merkmale

[Klass]

Feldname = Feldinhalt1, Feldinhalt2, ...

Alle Merkmale deren Feldname den Inhalt Feldinhalt haben werden als K-Merkmale behandelt und stehen nicht mehr als A-Merkmale zur Verfügung.

Beispiel 1:

[Klass]

Name = Laenge, Breite

Alle Merkmale deren Namen Laenge oder Breite ist werden als K-Merkmale behandelt.

Beispiel 2:

[Klass]

Methode = Klass

Schnitt = 1

FELD1 = A, AV

Merkmale mit: Methode = Klass oder
 Schnitt = 1 oder
 Feld1 = A oder
 Feld1 = AV werden als K-Merkmale behandelt.

Anmerkung:

In der Updatedatei wird erst der Eintrag für die Modifikation der Merkmalsnamen und danach der Abschnitt Klass verarbeitet, d.h. erst erfolgt die Umbenennung, danach die Klassifizierung.

4. Harmonisierung einer Merkmals-Variablen in Versuchsserien

Versuchsserien werden in PIAF bezüglich der Merkmale harmonisiert. D.h. jedem Einzelversuch der Serie werden die Merkmale der Serie zugeordnet. Hat ein Einzelversuch keine Daten für ein Merkmal der Serie, so werden für diesen Versuch nicht alle Variablen⁷ des Merkmals in die ADS geschrieben.

Werden die nicht vorhandenen Variablen eines Merkmals zur Änderung der Merkmalsnamen herangezogen, entstehen zwangsläufig nicht klar definierte Merkmalsnamen.

Beispiel

Eine Versuchsserie besteht aus den Einzelversuchen 8849, 4537 und 3984. Diese Versuche wurden für das Merkmal *Gesamt-TM dt/ha* zu den Terminen ESM, EUS und ESMFJ harmonisiert.

⁷ Variable eines Merkmals sind z.B. Termin, Schnitt, Zielorganismus, Sorte, ...
(siehe PIAFStat-Variablenname in VMERKMAL.TXT)

Versuchsnummer	Nummer der Datenspalte	Merkmalsname	Merkmalsbezeichnung	Termin
8849	13	ERT_TM	Gesamt-TM dt/ha	ESM
8849	25	ERT_TM	Gesamt-TM dt/ha	
8849	126	ERT_TM	Gesamt-TM dt/ha	
4537	13	ERT_TM	Gesamt-TM dt/ha	ESM
4537	25	ERT_TM	Gesamt-TM dt/ha	EUS
4537	126	ERT_TM	Gesamt-TM dt/ha	ESMFJ
3984	13	ERT_TM	Gesamt-TM dt/ha	ESM
3984	25	ERT_TM	Gesamt-TM dt/ha	EUS
3984	126	ERT_TM	Gesamt-TM dt/ha	

Der Versuch 8849 besitzt nur Daten für den Termin ESM. Der Versuch 2537 verfügt über Daten zu allen Terminen. Der Versuch 3984 hat nur Daten für die Termine ESM und EUS. Wird nun in der VUpdate.ini die Änderung des Merkmalsnamens wie folgt festgelegt

```
[Merkmale]
```

```
Name = Name + "_" + Termin
```

so sind die Merkmalsnamen für nicht vorhandene Termine nicht mehr klar definiert.

Mit Hilfe der VUpdate.ini kann im Abschnitt [Serie] festgelegt werden, dass die fehlenden Einträge der Termine ergänzt werden.

```
[Serie]
```

```
HarmonMerkmalVariable = Merkmalsvariable
```

Die fehlenden Einträge in der PIAFStat-Merkmalsvariable <Merkmalsvariable> werden ergänzt. Diese Ergänzung erfolgt nur für Merkmale von Einzelversuchen in Versuchsserien. Sie wird nur dann durchgeführt, wenn eindeutig die Nummern der Datenspalten zu den Inhalten der Merkmalsvariablen zugeordnet werden können.

Beispiel:

```
[Serie]
```

```
HarmonMerkmalVariable = Termin
```

In einer Versuchsserie werden die fehlenden Einträge der Merkmalsvariablen Termin in den Einzelversuchen ergänzt.

2.3 Programmcode

2.3.1 SAS

Bei der Generierung des SAS-Codes werden folgende Dateien im temporären Nutzerverzeichnis erstellt:

Datei	Bemerkung
PIAFSTAT.SAS	Generierter SAS-Programmcode
PIAFSTAT.DAT	Datendatei für Dataset: DATEN
PIAFSTAT.DSE	Datendatei für Dataset: DATENSE
PIAFSTAT.VER	Datendatei für Dataset: VERS
PIAFSTAT.MER	Datendatei für Dataset: MERK
PIAFSTAT.MVE	Datendatei für Dataset: MERKVERS
PIAFSTAT.FAK	Datendatei für Dataset: FAK
PIAFSTAT.GRD	Datendatei für Dataset: GRENZD
PIAFSTAT.PDZ	Datendatei für Dataset: PDZUORD
PIAFSTAT.PVZ	Datendatei für Dataset: PVZUORD

Der Programmcode in der Datei „PIAFSTAT.SAS“ besteht aus den Abschnitten globale Informationen, Versuche, Merkmale, Faktoren, Grenzdifferenzen, Daten, SE-Daten und Statistische Verfahren.

Wird das Verfahren ohne ADS-Daten ausgeführt erfolgt in den Abschnitten Versuche, Merkmale, Faktoren, Grenzdifferenzen, Daten und SE-Daten keine Erstellung von Datasets.

2.3.2 R

Bei der Generierung des R-Codes werden folgende Dateien im temporären Nutzerverzeichnis erstellt:

Datei	Bemerkung
PIAFSTAT.R	Generierter R-Programmcode
PIAFSTAT.RDAT	Datendatei für Dataframe: DATEN und DATEN_LABEL
PIAFSTAT.RDSE	Datendatei für Dataframe: DATENSE und DATENSE_LABEL
PIAFSTAT.RVER	Datendatei für Dataframe: VERS und VERS_LABEL
PIAFSTAT.RMER	Datendatei für Dataframe: MERK und MERK_LABEL
PIAFSTAT.RMVE	Datendatei für Dataframe: MERKVERS und MERKVERS_LABEL
PIAFSTAT.RFAK	Datendatei für Dataframe: FAK und FAK_LABEL
PIAFSTAT.RGRD	Datendatei für Dataframe: GRENZD und GRENZD_LABEL
PIAFSTAT.RPDZ	Datendatei für Dataframe: PDZUORD und PDZUORD_LABEL
PIAFSTAT.RPVZ	Datendatei für Dataframe: PVZUORD und PVZUORD_LABEL

Der Programmcode in der Datei „PIAFSTAT.R“ besteht aus den Abschnitten globale Informationen, Versuche, Merkmale, Faktoren, Grenzdifferenzen, Daten, SE-Daten und Statistische Verfahren.

Wird das Verfahren ohne ADS-Daten ausgeführt erfolgt in den Abschnitten Versuche, Merkmale, Faktoren, Grenzdifferenzen, Daten und SE-Daten keine Erstellung von Dataframes.

2.3.3 Globale Informationen

In diesem Abschnitt werden die folgenden globalen Variablen für SAS bzw. R definiert:

Nr.	Variable	Bemerkung
1	Verzeichnisse	
2	VERZADS	Verzeichnis der aktuellen ADS (Wurden im Dialogmodus keine Daten aus der ADS gelesen hat diese Variable den Inhalt der Zeichenkette „ADS-Verzeichnis“ des Dialoges „SAS/R-ausführen“.)
3	VERZRES	Verzeichnis der aktuellen RES
4	VERZTMP	Temporäres Verzeichnis
5	MODA	Dateiname der MoDa
5*	LIS	Dateiname der List-Datei für R
6	Allgemein	
7	VERFAHRENAME	Name des statistischen Verfahrens
8	VERFAHRENLABEL	Label des statistischen Verfahrens
9	VERFAHRENAUTOR	Autor des statistischen Verfahrens
10	VERFAHRENVERSION	Versionsnummer des statistischen Verfahrens
11	VERFAHRENDATUM	Versionsdatum des statistischen Verfahrens
12	VERFAHRENZEIT	Versionsuhrzeit des statistischen Verfahrens
13	VERFAHRENERSTELLT	Ersteller (Name Firma und Bundesland getrennt mit Komma) des statistischen Verfahrens
14	VERFAHRENGEAENDERT	Letzte Änderung (Angabe analog Ersteller)
15	Serie	
16	ASNUMMER	Nummer der A-Serie
17	ASNAME	Name der A-Serie
18	ASJAHR	Jahr der A-Serie
19	ASANZVERS	Anzahl der Versuche
20	ASFRUCHTART	Fruchtart (DIVERSE - falls die Fruchtart in den Einzelversuchen unterschiedlich ist)
21	Versuch	
22	VSERIE	Serienbezeichnung
23	VJAHR	Jahr
24	VORTBEZ	Ortsbezeichnung
25	VKENNUNG	Versuchskennung
26	VFRUCHTART	Fruchtartbezeichnung
27	VFRAGE	Versuchsfrage
28	VLAND	Landesnummer
29	VANLAGE	Anlage
30	VDETAIL	Detail
31	VORT	Ortskürzel (Benutzer orientiert)
32	VBKR	Bodenklimaraum
33	VWDH	Anzahl der Wiederholungen
34	ANZFAKTOREN	Anzahl der Faktoren

Nr.	Variable	Bemerkung
35	ANZFAKTORVAR	Maximale Anzahl der Faktorvariablen
36	ANZSORTENGRVAR	Maximale Anzahl der Sortengruppen
37	ANZMERKMALE	Anzahl der Merkmale
38	ANZGRENZDIFF	Anzahl der Grenzdifferenzen
39 ...	FiBEZj	Bezeichnung der Faktorvariable j des Faktors i i=1..Anzahl der Faktoren und j= 1..Anzahl der Faktorvariablen
	FiSBEZj	Bezeichnung der Sortengruppe j des Faktors i i=1..Anzahl der Faktoren und j= 1..Anzahl der Sortengruppen

* Variable wird nur für R-Programme erzeugt

Anmerkung: Die globalen Variablen für A-Serien werden nur definiert, wenn eine A-Serie ausgewertet wird. Die Variablen „FiBEZj“ korrespondieren mit den PIAF-Schnittstellenvariablen „Fvar1Bez“ aus Vstufen.txt.

Wurden im Dialogmodus keine Daten aus der ADS gelesen werden die Serien- und Versuchsspezifischen globalen Variablen (ab Nr.15) nicht erzeugt.

2.3.4 Versuche

2.3.4.1 SAS

Für die Versuche wird das Dataset VERS erstellt. Dieses Set besteht aus allen Feldern der ADS-Datei Versuche.txt. Werden Einzelversuche ausgewertet so enthält dieses Dataset nur den Datensatz des auszuwertenden Versuches. Bei Auswertungsserien ist für jeden Versuch der Serie ein Datensatz vorhanden. Stehen keine ADS-Daten zur Verfügung so wird das Dataset VERS nicht erstellt.

```
/*Versuchdefinition*/
```

```
DATA VERS;
```

```
INFORMAT SNR $15. ORTID $5. ORTBEZ $30. VKENN $50. FAG $30. VFRAGE $60.  
DETAIL $20. ORT $5. ASNAME $30. BKRBEZ $50.;
```

```
INFILE "temporäres Nutzerverzeichnis\PIAFSTAT.VER" LRECL=128;
```

```
INPUT VNR DNR SNR & JAHR LAND ORTID & ORTBEZ & VKENN & FAG & VFRAGE & WDH  
ANLAGE DETAIL & ORT & ASNR ASNAME & ASJAHR BKR BKRBEZ & @;
```

```
INPUT ANZFVAR ANZSGVAR;
```

```
/*Label zuordnen*/
```

```
Label
```

```
VNR          = "Versuchsnummer"  
DNR          = "Bestandsnummer"  
SNR          = "Serie"  
JAHR         = "Jahr"  
LAND         = "Land"  
ORTID        = "OrtID"  
ORTBEZ       = "Ortsbezeichnung"  
VKENN        = "Kennung"  
FAG          = "Fruchtartbezeichnung"  
VFRAGE       = "V-Frage"  
WDH          = "Anz. Wdh."  
ANLAGE       = "Anlage"  
DETAIL       = "Detail"  
ORT          = "Ort"  
ASNR         = "A-Serie Nummer"
```

```

ASNAME      = "A-Serie Bezeichnung"
ASJAHR      = "A-Serie Jahr"
BKR         = "Bodenklimaraum"
BKRBEZ      = "Bodenklimaraumbezeichnung"
ANZFVAR     = "Anzahl Faktorvariable"
ANZSGVAR    = "Anzahl Sortengruppen" ;
RUN;

```

2.3.4.2 R

Für die Versuche werden die Dataframes VERS und VERS_LABEL erstellt. Diese Frames bestehen aus allen Feldern der ADS-Datei Versuche.txt. Werden Einzelversuche ausgewertet so enthält der Datenframe VERS nur den Datensatz des auszuwertenden Versuches. Bei Auswertungsserien ist für jeden Versuch der Serie ein Datensatz vorhanden.

Stehen keine ADS-Daten zur Verfügung so werden die Dataframes VERS und VERS_LABEL nicht erstellt.

```

# Versuchdefinition
tmpcolnames <- c(
  "VNR",
  "DNR",
  "SNR",
  "JAHR",
  "LAND",
  "ORTID",
  "ORTBEZ",
  "VKENN",
  "FAG",
  "VFRAGE",
  "WDH",
  "ANLAGE",
  "DETAIL",
  "ORT",
  "ASNR",
  "ASNAME",
  "ASJAHR",
  "BKR",
  "BKRBEZ",
  "ANZFVAR",
  "ANZSGVAR"
)

```

```

tmpcolClasses <-c(
  "integer",
  "integer",
  "character",
  "integer",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "character",
  "integer",
  "integer",
  "character",
  "character",
  "integer",
  "character",

```

```

"integer",
"integer",
"character",
"integer",
"integer"
)

VERS <- read.table(
  "temporäres Nutzerverzeichnis/PIAFSTAT.RVER",
  header      = FALSE,
  sep         = '\t',
  col.names   = tmpcolnames,
  colClasses  = tmpcolClasses,
  na.string   = ".",
  fileEncoding = "latin1"
)

VERS_LABEL <- c(
  "VNR"      = "Versuchsnummer",
  "DNR"      = "Bestandsnummer",
  "SNR"      = "Serie",
  "JAHR"     = "Jahr",
  "LAND"     = "Land",
  "ORTID"    = "OrtID",
  "ORTBEZ"   = "Ortsbezeichnung",
  "VKENN"    = "Kennung",
  "FAG"      = "Fruchtartbezeichnung",
  "VFRAGE"   = "V-Frage",
  "WDH"      = "Anz. Wdh.",
  "ANLAGE"   = "Anlage",
  "DETAIL"   = "Detail",
  "ORT"      = "Ort",
  "ASNR"     = "A-Serie Nummer",
  "ASNAME"   = "A-Serie Bezeichnung",
  "ASJAHR"   = "A-Serie Jahr",
  "BKR"      = "Bodenklimaraum",
  "BKRBEZ"   = "Bodenklimaraumbezeichnung",
  "ANZFVAR"  = "Anzahl Faktorvariable",
  "ANZSGVAR" = "Anzahl Sortengruppen"
)

# temporäre Objekte löschen
rm(tmpcolnames)
rm(tmpcolClasses)

```

2.3.5 Merkmale

2.3.5.1 SAS

Für die Merkmale werden die Datasets MERK und MERKVERS erstellt. Diese Sets bestehen aus den Feldern der ADS-Datei Vmerkmal.txt.

Werden Einzelversuche ausgewertet so enthält sowohl das Dataset MERK als auch das Dataset MERKVERS die Merkmale des auszuwertenden Einzelversuchs. Bei Auswertungsserien sind die serienbezogenen Merkmale im Dataset MERK, die der einzelnen Versuche der Serie im Dataset MERKVERS verfügbar.

Stehen keine ADS-Daten zur Verfügung so werden die Datasets MERK und MERKVERS nicht erstellt.

DataSet MERK

*/*Merkmaldefinition*/*

DATA MERK;

INFORMAT NAME \$32. OBJEKT \$7. BEZUG \$6. METHODE \$6. HERK \$3. BEZ \$40. SCHAD \$6. NAME1 \$8. NAME2 \$16. TYP \$1. GRPART \$4. DATUM DDMMYY8.;

INFORMAT FLAG \$1. FELD1 \$10. FELD2 \$10. FELD3 \$10. FELD4 \$10. FELD5 \$10. FELD6 \$10. FELD7 \$10. FELD8 \$10. FELD9 \$10. FELD10 \$10.;

INFORMAT WSTATUS \$4. TERMIN \$3. AGGSTPRO \$3. FLAGPB \$1. KULTUR \$5. KULBEZ \$30. SORTE \$15. SBEZ \$50. SBEZLAT \$50. SKLASSE \$10.;

INFORMAT MKEY \$8. BSANAME \$5. FORMEL \$250.;

INFILE "temporäres Nutzerverzeichnis\PIAFSTAT\PIAFSTAT.MER" LRECL=176;

INPUT VNR MI NAME & OBJEKT & BEZUG & METHODE & HERK & VARIANTE BEZ & MNR SCHAD & NAME1 & NAME2 & TYP & DEZIMAL GRPART & DATUM & BBCHVON @;

INPUT BBCHBIS FLAG & NARITH FELD1 & FELD2 & FELD3 & FELD4 & FELD5 & FELD6 & FELD7 & FELD8 & FELD9 & FELD10 & SCHNITT & WSTATUS & @;

INPUT TERMIN & EBBCHVON EBBCH EBBCHBIS ESCHNITT EINPOS MINVALUE MAXVALUE ANZSTPRO AGGSTPRO & FLAGPB & KULNR KULTUR & KULBEZ & SORTE & @;

INPUT SBEZ & SBEZLAT & SKLASSE & MKEY & BSANAME & FORMEL &;

FORMAT DATUM EURDFDD8.;

*/*Label zuordnen*/*

Label

MI	= "Nummer Datenspalte"
NAME	= "Name"
OBJEKT	= "Objekt"
BEZUG	= "Bezug"
METHODE	= "Methode"
HERK	= "Herkunft"
VARIANTE	= "Variante"
BEZ	= "Bezeichnung"
MNR	= "Merkmalsnummer"
SCHAD	= "Zielorganismus"
NAME1	= "Name1"
NAME2	= "Name2"
TYP	= "Merkmalstyp"
DEZIMAL	= "Dezimalstellen"
GRPART	= "Gruppenart"
DATUM	= "Datum"
BBCHVON	= "BBCH von"
BBCHBIS	= "BBCH bis"
FLAG	= "Adjustierung"
NARITH	= "zulässige Fehlwerte"
FELD1	= "Analysetyp"
FELD2	= "Skalierung"
FELD3	= "FELD3"
FELD4	= "FELD4"
FELD5	= "FELD5"
FELD6	= "FELD6"
FELD7	= "FELD7"
FELD8	= "FELD8"
FELD9	= "FELD9"
FELD10	= "FELD10"
SCHNITT	= "Schnittnummer"
WSTATUS	= "Wertestatus"
TERMIN	= "Terminkürzel"
EBBCHVON	= "Entwicklungsstadium (von) des Versuchs"

```

EBBCH      = "Entwicklungsstadium (Haupt) des Versuchs"
EBBCHBIS   = "Entwicklungsstadium (bis) des Versuchs"
ESCHNITT   = "Schnittnummer des Versuchs"
EINPOS     = "Position des Merkmals bei der Eingabe"
MINVALUE   = "Min-Wert"
MAXVALUE   = "Max-Wert"
ANZSTPRO   = "Anzahl der Stichproben"
AGGSTPRO   = "Stichprobenberechnungsfunktion"
FLAGPB     = "Flag Prüfbericht"
KULNR      = "Kulturnummer"
KULTUR     = "Kulturcode"
KULBEZ     = "Kulturbezeichnung"
SORTE      = "Sorte"
SBEZ       = "Bezeichnung des Schadorganismus"
SBEZLAT    = "Lateinische Bezeichnung des Schadorganismus"
SKLASSE    = "Klasse des Schadorganismus"
MKEY       = "M_Key"
BSANAME    = "BSA Name"
FORMEL     = "Formel" ;
RUN;

```

DataSet MERKVERS

```
/*Merkmal-Versuchdefinition*/
```

```
DATA MERKVERS;
```

```
INFORMAT NAME $32. OBJEKT $7. BEZUG $6. METHODE $6. HERK $3. BEZ $40. SCHAD $6. NAME1 $8. NAME2 $16. TYP $1. GRPART $4. DATUM DDMMYY8.;
```

```
INFORMAT FLAG $1. FELD1 $10. FELD2 $10. FELD3 $10. FELD4 $10. FELD5 $10. FELD6 $10. FELD7 $10. FELD8 $10. FELD9 $10. FELD10 $10.;
```

```
INFORMAT WSTATUS $4. TERMIN $3. AGGSTPRO $3. FLAGPB $1. KULTUR $5. KULBEZ $30. SORTE $15. SBEZ $50. SBEZLAT $50. SKLASSE $10.;
```

```
INFORMAT MKEY $8. BSANAME $5. FORMEL $250.;
```

```
INFILE "temporäres Nutzerverzeichnis\PIAFSTAT\PIAFSTAT.MVE" LRECL=192;
```

```
INPUT VNR MI NAME & OBJEKT & BEZUG & METHODE & HERK & VARIANTE BEZ & MNR SCHAD & NAME1 & NAME2 & TYP & DEZIMAL GRPART & DATUM & BBCHVON @;
```

```
INPUT BBCHBIS FLAG & NARITH FELD1 & FELD2 & FELD3 & FELD4 & FELD5 & FELD6 & FELD7 & FELD8 & FELD9 & FELD10 & SCHNITT & WSTATUS & @;
```

```
INPUT TERMIN & EBBCHVON EBBCH EBBCHBIS ESCHNITT EINPOS MINVALUE MAXVALUE ANZSTPRO AGGSTPRO & FLAGPB & KULNR KULTUR & KULBEZ & SORTE & @;
```

```
INPUT SBEZ & SBEZLAT & SKLASSE & MKEY & BSANAME & FORMEL &;
```

```
FORMAT DATUM EURDFDD8.;
```

```
/*Label zuordnen*/
```

```
Label
```

```

NVR      = "Versuchsnummer"
MI       = "Nummer Datenspalte"
NAME     = "Name"
OBJEKT   = "Objekt"
BEZUG    = "Bezug"
METHODE  = "Methode"
HERK     = "Herkunft"
VARIANTE = "Variante"
BEZ      = "Bezeichnung"
MNR      = "Merkmalsnummer"
SCHAD    = "Zielorganismus"
NAME1    = "Name1"
NAME2    = "Name2"

```

```

TYP           = "Merkmalstyp"
DEZIMAL       = "Dezimalstellen"
GRPART        = "Gruppenart"
DATUM         = "Datum"
BBCHVON       = "BBCH von"
BBCHBIS       = "BBCH bis"
FLAG          = "Adjustierung"
NARITH        = "zulässige Fehlwerte"
FELD1         = "Analysetyp"
FELD2         = "Skalierung"
FELD3         = "FELD3"
FELD4         = "FELD4"
FELD5         = "FELD5"
FELD6         = "FELD6"
FELD7         = "FELD7"
FELD8         = "FELD8"
FELD9         = "FELD9"
FELD10        = "FELD10"
SCHNITT       = "Schnittnummer"
WSTATUS       = "Wertestatus" ;
TERMIN        = "Terminkürzel" ;
EBBCHVON      = "Entwicklungsstadium (von) des Versuchs"
EBBCH         = "Entwicklungsstadium (Haupt) des Versuchs"
EBBCHBIS      = "Entwicklungsstadium (bis) des Versuchs"
ESCHNITT      = "Schnittnummer des Versuchs"
EINPOS        = "Position des Merkmals bei der Eingabe"
MINVALUE      = "Min-Wert"
MAXVALUE      = "Max-Wert"
ANZSTPRO      = "Anzahl der Stichproben"
AGGSTPRO      = "Stichprobenberechnungsfunktion"
FLAGPB        = "Flag Prüfbericht"
KULNR         = "Kulturnummer"
KULTUR        = "Kulturcode"
KULBEZ        = "Kulturbezeichnung"
SORTE         = "Sorte"
SBEZ          = "Bezeichnung des Schadorganismus"
SBEZLAT       = "Lateinische Bezeichnung des Schadorganismus"
SKLASSE       = "Klasse des Schadorganismus"
MKEY          = "M_Key"
BSANAME       = "BSA Name"
FORMEL        = "Formel" ;
RUN;

```

2.3.5.1 R

Für die Merkmale werden die Dataframes MERK, MERK_LABEL, MERKVERS und MERKVERS_LABEL erstellt. Diese Frames bestehen aus den Feldern der ADS-Datei Vmerkmal.txt.

Werden Einzelversuche ausgewertet so enthalten die Dataframes MERK und MERVERS die Merkmale des auszuwertenden Einzelversuchs. Bei Auswertungsserien sind die serienbezogenen Merkmale im Dataframe MERK, die der einzelnen Versuche der Serie im Dataframe MERKVERS verfügbar. Stehen keine ADS-Daten zur Verfügung so werden die Dataframes MERK, MERK_LABEL, MERKVERS und MERKVERS_LABEL nicht erstellt.

Dataframe MERK und MERK_LABEL

```
# Merkmaldefinition
tmpcolnames <- c(
  "VNR",
  "MI",
  "NAME",
  "OBJEKT",
  "BEZUG",
  "METHODE",
  "HERK",
  "VARIANTE",
  "BEZ",
  "MNR",
  "SCHAD",
  "NAME1",
  "NAME2",
  "TYP",
  "DEZIMAL",
  "GRPART",
  "DATUM",
  "BBCHVON",
  "BBCHBIS",
  "FLAG",
  "NARITH",
  "FELD1",
  "FELD2",
  "FELD3",
  "FELD4",
  "FELD5",
  "FELD6",
  "FELD7",
  "FELD8",
  "FELD9",
  "FELD10",
  "SCHNITT",
  "WSTATUS",
  "TERMIN",
  "EBBCHVON",
  "EBBCH",
  "EBBCHBIS",
  "ESCHNITT",
  "EINPOS",
  "MINVALUE",
  "MAXVALUE",
  "ANZSTPRO",
  "AGGSTPRO",
  "FLAGPB",
  "KULNR",
  "KULTUR",
  "KULBEZ",
  "SORTE",
  "SBEZ",
  "SBEZLAT",
  "SKLASSE",
  "MKEY",
  "BSANAME",
  "FORMEL",
  "NSORT"
)
```

```

tmpcolClasses <-c(
  "integer",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "character",
  "integer",
  "character",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "integer",
  "character",
  "character",
  "integer",
  "integer",
  "character",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "integer",
  "character",
  "character",
  "integer",
  "integer",
  "integer",
  "integer",
  "integer",
  "numeric",
  "numeric",
  "integer",
  "character",
  "character",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "integer"
)

MERK <- read.table(
  "temporäres Nutzerverzeichnis/PIAFSTAT.RMER",

```

```

Header      = FALSE,
sep         = '\t',
col.names   = tmpcolnames,
colClasses  = tmpcolClasses,
na.string   = ".",
fileEncoding = "latin1"
)

MERK_LABEL <- c(
  "VNR"      = "Versuchsnummer",
  "MI"       = "Nummer Datenspalte",
  "NAME"     = "Name",
  "OBJEKT"   = "Objekt",
  "BEZUG"    = "Bezug",
  "METHODE"  = "Methode",
  "HERK"     = "Herkunft",
  "VARIANTE" = "Variante",
  "BEZ"      = "Bezeichnung",
  "MNR"      = "Merkmalsnummer",
  "SCHAD"    = "Zielorganismus",
  "NAME1"    = "Name1",
  "NAME2"    = "Name2",
  "TYP"      = "Merkmalstyp",
  "DEZIMAL"  = "Dezimalstellen",
  "GRPART"   = "Gruppenart",
  "DATUM"    = "Datum",
  "BBCHVON"  = "BBCH von",
  "BBCHBIS"  = "BBCH bis",
  "FLAG"     = "Adjustierung",
  "NARITH"   = "notw. N für MW (%)",
  "FELD1"    = "Analysetyp",
  "FELD2"    = "Skalierung",
  "FELD3"    = "FELD3",
  "FELD4"    = "FELD4",
  "FELD5"    = "FELD5",
  "FELD6"    = "FELD6",
  "FELD7"    = "FELD7",
  "FELD8"    = "FELD8",
  "FELD9"    = "FELD9",
  "FELD10"   = "FELD10",
  "SCHNITT"  = "Schnittnummer",
  "WSTATUS"  = "Wertestatus",
  "TERMIN"   = "Terminkürzel",
  "EBBCHVON" = "Entwicklungsstadium (von) des Versuchs",
  "EBBCH"    = "Entwicklungsstadium (Haupt) des Versuchs",
  "EBBCHBIS" = "Entwicklungsstadium (bis) des Versuchs",
  "ESCHNITT" = "Schnittnummer des Versuchs",
  "EINPOS"   = "Position des Merkmals bei der Eingabe",
  "MINVALUE" = "Min-Wert",
  "MAXVALUE" = "Max-Wert",
  "ANZSTPRO" = "Anzahl der Stichproben",
  "AGGSTPRO" = "Stichprobenberechnungsfunktion",
  "FLAGPB"   = "Flag Prüfbericht",
  "KULNR"    = "Kulturnummer",
  "KULTUR"   = "Kulturcode",
  "KULBEZ"   = "Kulturbezeichnung",
  "SORTE"    = "Sorte",
  "SBEZ"     = "Bezeichnung des Schadorganismus",
  "SBEZLAT"  = "Lateinische Bezeichnung des Schadorganismus",
  "SKLASSE"  = "Klasse des Schadorganismus",
  "MKEY"     = "M_Key",

```

```

"BSANAME" = "BSA Name",
"FORMEL"  = "Formel",
"NSORT"   = "NSORT"
)

# temporäre Objekte löschen
rm(tmpcolnames)
rm(tmpcolClasses)

```

Dataframe MERKVERS und MERKVERS_LABEL

```

# Merkmal-Versuchsdefinition
tmpcolnames <- c(
  "VNR",
  "MI",
  "NAME",
  "OBJEKT",
  "BEZUG",
  "METHODE",
  "HERK",
  "VARIANTE",
  "BEZ",
  "MNR",
  "SCHAD",
  "NAME1",
  "NAME2",
  "TYP",
  "DEZIMAL",
  "GRPART",
  "DATUM",
  "BBCHVON",
  "BBCHBIS",
  "FLAG",
  "NARITH",
  "FELD1",
  "FELD2",
  "FELD3",
  "FELD4",
  "FELD5",
  "FELD6",
  "FELD7",
  "FELD8",
  "FELD9",
  "FELD10",
  "SCHNITT",
  "WSTATUS",
  "TERMIN",
  "EBBCHVON",
  "EBBCH",
  "EBBCHBIS",
  "ESCHNITT",
  "EINPOS",
  "MINVALUE",
  "MAXVALUE",
  "ANZSTPRO",
  "AGGSTPRO",
  "FLAGPB",
  "KULNR",
  "KULTUR",
  "KULBEZ",

```

```

"SORTE",
"SBEZ",
"SBEZLAT",
"SKLASSE",
"MKEY",
"BSANAME",
"FORMEL",
"NSORT"
)

tmpcolClasses <-c(
  "integer",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "character",
  "integer",
  "character",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "integer",
  "character",
  "character",
  "integer",
  "integer",
  "character",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",
  "integer",
  "character",
  "character",
  "integer",
  "integer",
  "integer",
  "integer",
  "integer",
  "numeric",
  "numeric",
  "integer",
  "character",
  "character",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "character",
  "character",

```



```

"character",
"character",
"character",
"character",
"integer"
)

MERKVERS <- read.table(
  "temporäres Nutzerverzeichnis/PIAFSTAT.RMVE",
  Header      = FALSE,
  sep         = '\t',
  col.names   = tmpcolnames,
  colClasses  = tmpcolClasses,
  na.string   = ".",
  fileEncoding = "latin1"
)

MERKVERS_LABEL <- c(
  "VNR"      = "Versuchsnummer",
  "MI"       = "Nummer Datenspalte",
  "NAME"     = "Name",
  "OBJEKT"   = "Objekt",
  "BEZUG"    = "Bezug",
  "METHODE"  = "Methode",
  "HERK"     = "Herkunft",
  "VARIANTE" = "Variante",
  "BEZ"      = "Bezeichnung",
  "MNR"      = "Merkmalsnummer",
  "SCHAD"    = "Zielorganismus",
  "NAME1"    = "Name1",
  "NAME2"    = "Name2",
  "TYP"      = "Merkmalstyp",
  "DEZIMAL"  = "Dezimalstellen",
  "GRPART"   = "Gruppenart",
  "DATUM"    = "Datum",
  "BBCHVON"  = "BBCH von",
  "BBCHBIS"  = "BBCH bis",
  "FLAG"     = "Adjustierung",
  "NARITH"   = "notw. N für MW (%)",
  "FELD1"    = "Analysetyp",
  "FELD2"    = "Skalierung",
  "FELD3"    = "FELD3",
  "FELD4"    = "FELD4",
  "FELD5"    = "FELD5",
  "FELD6"    = "FELD6",
  "FELD7"    = "FELD7",
  "FELD8"    = "FELD8",
  "FELD9"    = "FELD9",
  "FELD10"   = "FELD10",
  "SCHNITT"  = "Schnittnummer",
  "WSTATUS"  = "Wertestatus",
  "TERMIN"   = "Terminkürzel",
  "EBBCHVON" = "Entwicklungsstadium (von) des Versuchs",
  "EBBCH"    = "Entwicklungsstadium (Haupt) des Versuchs",
  "EBBCHBIS" = "Entwicklungsstadium (bis) des Versuchs",
  "ESCHNITT" = "Schnittnummer des Versuchs",
  "EINPOS"   = "Position des Merkmals bei der Eingabe",
  "MINVALUE" = "Min-Wert",
  "MAXVALUE" = "Max-Wert",
  "ANZSTPRO" = "Anzahl der Stichproben",
  "AGGSTPRO" = "Stichprobenberechnungsfunktion",

```

```

"FLAGPB"    = "Flag Prüfbericht",
"KULNR"     = "Kulturnummer",
"KULTUR"    = "Kulturcode",
"KULBEZ"    = "Kulturbezeichnung",
"SORTE"     = "Sorte",
"SBEZ"      = "Bezeichnung des Schadorganismus",
"SBEZLAT"   = "Lateinische Bezeichnung des Schadorganismus",
"SKLASSE"   = "Klasse des Schadorganismus",
"MKEY"      = "M_Key",
"BSANAME"   = "BSA Name",
"FORMEL"    = "Formel",
"NSORT"     = "NSORT"
)

```

```

# temporäre Objekte löschen
rm(tmpcolnames)
rm(tmpcolClasses)

```

2.3.6 Faktoren

2.3.6.1 SAS

Für die Faktoren und Faktorstufen wird das Dataset FAK erstellt. Dieses Set besteht aus den Feldern der ADS-Dateien Vfaktor.txt und Vstufe.txt. Stehen keine ADS-Daten zur Verfügung so wird das Dataset FAK nicht erstellt.

```

/*Faktordefinition*/
DATA FAK;
INFORMAT BEZFA $30. BEZ1 $23. BEZ2 $23. BEZ3 $23. SG1 $23. SG2 $23. BEZST
$100. WERT1 $30. WERT2 $30. WERT3 $30. SGWERT1 $20. SGWERT2 $20.;

INFILE "temporäres Nutzerverzeichnis\PIAFSTAT.FAK" LRECL=320;

INPUT VNR 1-10 FKEY 11-20 BEZFA 21-50 BEZ1 51-70 BEZ2 71-90 BEZ3 91-110 SG1
111-130 SG2 131-150 FI 151-160 BEZST 161-190 WERT1 191-220 WERT2 221-250
WERT3 251-280 SGWERT1 281-300 SGWERT2 301-320;

/*Label zuordnen*/
Label
VNR          = "Versuchsnummer"
FKEY         = "Faktor"
BEZFA        = "Bezeichnung"
BEZ1         = "Fvar1Bez"
BEZ2         = "Fvar2Bez"
BEZ3         = "Fvar3Bez"
BEZn         = "FvarnBez"
SG1          = "SG1Bez"
SG2          = "SG2Bez"
FI           = "Stufe"
BEZST        = "Bezeichnung"
WERT1        = "Fvar1Wert"
WERT2        = "Fvar2Wert"
WERT3        = "Fvar3Wert"
WERTn        = "FvarnWert"
SGWERT1      = "SG1Wert"
SGWERT2      = "SG2Wert" ;
RUN;

```

2.3.6.1 R

Für die Faktoren und Faktorstufen werden die Dataframes FAK und FAK_LABEL erstellt. Diess Dataframes bestehen aus den Feldern der ADS-Dateien Vfaktor.txt und Vstufe.txt.

Stehen keine ADS-Daten zur Verfügung so werden die Dataframes FAK und FAK_LABEL nicht erstellt.

```
# Faktordefinition
tmpcolnames <- c(
  "VNR",
  "FKEY",
  "BEZFA",
  "BEZ1",
  "SG1",
  "SG2",
  "FI",
  "BEZST",
  "WERT1",
  "SGWERT1",
  "SGWERT2"
)

tmpcolClasses <-c(
  "integer",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "integer",
  "character",
  "character",
  "character",
  "character"
)

FAK <- read.table(
  "TEMPORÄRES NUTZERVERZEICHNIS/PIAFSTAT.RFAK",
  header      = FALSE,
  sep         = '\t',
  col.names   = tmpcolnames,
  colClasses  = tmpcolClasses,
  na.string   = ".",
  fileEncoding = "latin1"
)

FAK_LABEL <- c(
  "VNR"      = "Versuchsnummer",
  "FKEY"     = "Faktor",
  "BEZFA"    = "Bezeichnung",
  "BEZ1"     = "Fvar1Bez",
  "SG1"      = "SG1Bez",
  "SG2"      = "SG2Bez",
  "FI"       = "Stufe",
  "BEZST"    = "Bezeichnung",
  "WERT1"    = "Fvar1Wert",
  "SGWERT1"  = "SG1Wert",
  "SGWERT2"  = "SG2Wert"
)
```

```
# temporäre Objekte löschen
rm(tmpcolnames)
rm(tmpcolClasses)
```

2.3.7 Grenzdifferenzen

2.3.7.1 SAS

Für die Grenzdifferenzen wird das Dataset GRENZD erstellt. Dieses Set besteht aus den Feldern der ADS-Datei Vgd.txt.

Stehen keine ADS-Daten zur Verfügung so wird das Dataset GRENZD nicht erstellt.

```
/*Grenzdifferenzdefinition*/
DATA GRENZD;
INFORMAT NAME $32. OBJEKT $7. BEZUG $6. METHODE $6. HERK $3. GRART $4.
STVERGL $4.;

INFILE "temporäres Nutzerverzeichnis\PIAFSTAT.GRD" LRECL=80;

INPUT VNR MI NAME & OBJEKT & BEZUG & METHODE & HERK & VARIANTE GRART &
STVERGL & GD5;

/*Label zuordnen*/
Label
VNR          = "Versuchsnummer"
MI           = "Merkmalsnummer"
NAME         = "Name"
OBJEKT       = "Objekt"
BEZUG        = "Bezug"
METHODE      = "Methode"
HERK         = "Herkunft"
VARIANTE     = "Variante"
GRART        = "Gruppenart"
STVERGL      = "Gruppenart Stufenvergleich"
GD5          = "Grenzdifferenz" ;
RUN;
```

2.3.7.1 R

Für die Grenzdifferenzen werden die Dataframes GRENZD und GRENZD_LABEL erstellt. Diese Dataframes bestehen aus den Feldern der ADS-Datei Vgd.txt. Stehen keine ADS-Daten zur Verfügung so werden die Dataframes GRENZD und GRENZD_LABEL nicht erstellt.

```
# Grenzdifferenzdefinition
tmpcolnames <- c(
  "VNR",
  "MI",
  "NAME",
  "OBJEKT",
  "BEZUG",
  "METHODE",
  "HERK",
  "VARIANTE",
  "GRART",
  "STVERGL",
  "GD5"
)
```

```

tmpcolClasses <-c(
  "integer",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "character",
  "integer",
  "character",
  "character",
  "numeric"
)

GRENZD <- read.table(
  "TEMPORÄRES NUTZERVERZEICHNIS/PIAFSTAT.RGRD",
  header      = FALSE,
  sep         = '\t',
  col.names   = tmpcolnames,
  colClasses  = tmpcolClasses,
  na.string   = ".",
  fileEncoding = "latin1"
)

GRENZD_LABEL <- c(
  "VNR"      = "Versuchsnummer",
  "MI"       = "Merkmalsnummer",
  "NAME"     = "Name",
  "OBJEKT"   = "Objekt",
  "BEZUG"    = "Bezug",
  "METHODE"  = "Methode",
  "HERK"     = "Herkunft",
  "VARIANTE" = "Variante",
  "GRART"    = "Gruppenart",
  "STVERGL"  = "Gruppenart Stufenvergleich",
  "GD5"      = "Grenzdifferenz"
)

# temporäre Objekte löschen
rm(tmpcolnames)
rm(tmpcolClasses)

```

2.3.8 Zuordnung der K-Merkmale

2.3.8.1 SAS

Für die Zuordnung der Programmvariablen der K-Merkmale zu den Datenvariablen wird das Dataset PDZUORD erstellt. Sind keine Zuordnungen vorhanden, so wird das Dataset PDZUORD nicht erstellt.

Stehen keine ADS-Daten zur Verfügung so wird das Dataset PDZUORD nicht erstellt.

```

/*Zuordnung der Programmvariablen zu den Datenvariablen*/
DATA PDZUORD;
INFORMAT PNAME $32. PBEZ $64. DNAME $16. DBEZ $40.;

INFILE "temporäres Nutzerverzeichnis\PIAFSTAT.PDZ" LRECL=208;

INPUT PNAME & PBEZ & DNAME & DBEZ &;

```

```

/*Label zuordnen*/
Label
PNAME ="Name der Programmvariable"
PBEZ  ="Label der Programmvariable"
DNAME ="Name der Datenvariable"
DBEZ  ="Bezeichnung der Datenvariable";
RUN;

```

2.3.8.1 R

Für die Zuordnung der Programmvariablen der K-Merkmale zu den Datenvariablen werden die Dataframes PDZUORD und PDZUORD_LABEL erstellt. Sind keine Zuordnungen vorhanden, so werden die Dataframes PDZUORD und PDZUORD_LABEL nicht erstellt.

Stehen keine ADS-Daten zur Verfügung so werden die Dataframes PDZUORD und PDZUORD_LABEL nicht erstellt.

```

# Zuordnung der Programmvariablen zu den Datenvariablen
tmpcolnames <- c(
  "PNAME",
  "PBEZ",
  "DNAME",
  "DBEZ"
)

tmpcolClasses <- c(
  "character",
  "character",
  "character",
  "character"
)

PDZUORD <- read.table(
  "TEMPORÄRES NUTZERWERZEICHNIS/PIAFSTAT.RPDZ",
  header      = FALSE,
  sep         = '\t',
  col.names   = tmpcolnames,
  colClasses  = tmpcolClasses,
  na.string   = ".",
  fileEncoding = "latin1"
)

PDZUORD_LABEL <- c(
  "PNAME" = "Name der Programmvariable",
  "PBEZ"  = "Label der Programmvariable",
  "DNAME" = "Name der Datenvariable",
  "DBEZ"  = "Bezeichnung der Datenvariable"
)

# temporäre Objekte löschen
rm(tmpcolnames)
rm(tmpcolClasses)

```

2.3.9 Zuordnung aller Programmvariablen

2.3.9.1 SAS

Für die Zuordnung der Programmvariablen zu den Datenvariablen wird das Dataset PVAZUORD erstellt. Sind keine Zuordnungen vorhanden, so wird das Dataset PVAZUORD nicht erstellt.

```

/*Inhalte der Programm-Variablen K-Merkmal, A-Merkmal und Optionen*/
DATA PVAZUORD;
INFORMAT PVARTYPE $32. PVARNAME $32. PVARLABEL $64. PVARVALUE $128.;

INFILE "temporäres Nutzerverzeichnis\PIAFSTAT.PVZ" LRECL=32;

INPUT PVARTYPE & PVARNAME & PVARLABEL & PVARVALUE &;

/*Label zuordnen*/
Label
PVARTYPE  ="Type der Programmvariable"
PVARNAME  ="Name der Programmvariable"
PVARLABEL ="Label der Programmvariable"
PVARVALUE ="Wert der Programmvariable";
RUN;

```

Die Variable PVARTYPE des SAS-Datasets PVAZUORD kennzeichnet den Typ der Programmvariable. Folgende Typen sind definiert:

PVARTYPE	Beschreibung
Verfahren	Informationen zum Verfahren
Funktion	Informationen zur Funktion
K-Merkmal Programm-Variable	Programmvariable des K-Merkmals
K-Merkmal Datenvariable-Variable	dem K-Merkmal zugeordnete Datenvariable
A-Merkmal Programm-Variable	Programmvariable des A-Merkmals
A-Merkmal Datenvariable-Variable	dem A-Merkmal zugeordnete Datenvariable
Option	(Haupt-)Option
Option-Block	optionaler Block der (Haupt-)Option
Sub-Option	Unteroption
Sub-Option-Block	optionaler Block der Unteroption

Anmerkung:

Das SAS-Dataset PVAZUORD wird während der PIAFStat-Sitzung chronologisch in einem Logbuch aufgezeichnet (siehe Abschnitt 2.4.7).

2.3.9.1 R

Für die Zuordnung der Programmvariablen zu den Datenvariablen werden die Dataframes PVAZUORD und PVAZUORD_LABEL erstellt. Sind keine Zuordnungen vorhanden, so werden die Dataframes PVAZUORD und PVAZUORD_LABEL nicht erstellt.

```

# Inhalte der Programm-Variablen K-Merkmal, A-Merkmal und Optionen
tmpcolnames <- c(
  "PVARTYPE",
  "PVARNAME",
  "PVARLABEL",
  "PVARVALUE"
)

tmpcolClasses <- c(
  "character",
  "character",
  "character",
  "character"
)

```

```

)

PVAZUORD <- read.table(
  "TEMPORÄRES NUTZERVERZEICHNIS/PIAFSTAT.RPVZ",
  header      = FALSE,
  sep         = '\t',
  col.names   = tmpcolnames,
  colClasses  = tmpcolClasses,
  na.string   = ".",
  fileEncoding = "latin1"
)

PVAZUORD_LABEL <- c(
  "PVARTYPE" = "Type der Programmvariable",
  "PVARNAME" = "Name der Programmvariable",
  "PVARLABEL" = "Label der Programmvariable",
  "PVARVALUE" = "Wert der Programmvariable"
)

# temporäre Objekte löschen
rm(tmpcolnames)
rm(tmpcolClasses)

```

Die Variable PVARTYPE des Dataframes PVAZUORD kennzeichnet den Typ der Programmvariable. Folgende Typen sind definiert:

PVARTYPE	Beschreibung
Verfahren	Informationen zum Verfahren
Funktion	Informationen zur Funktion
K-Merkmal Programm-Variable	Programmvariable des K-Merkmals
K-Merkmal Datenvariable-Variable	dem K-Merkmal zugeordnete Datenvariable
A-Merkmal Programm-Variable	Programmvariable des A-Merkmals
A-Merkmal Datenvariable-Variable	dem A-Merkmal zugeordnete Datenvariable
Option	(Haupt-)Option
Option-Block	optionaler Block der (Haupt-)Option
Sub-Option	Unteroption
Sub-Option-Block	optionaler Block der Unteroption

Anmerkung:

Der Dataframe PVAZUORD wird während der PIAFStat-Sitzung chronologisch in einem Logbuch aufgezeichnet (siehe Abschnitt 2.4.7).

2.3.10 Daten

2.3.10.1 SAS

Für den Zugriff auf die Daten wird das Dataset DATEN erstellt. Dieses Set ist bezüglich der Anzahl der Faktoren und Merkmale dynamisch aufgebaut. Das folgende Beispiel bezieht sich auf einen Versuch mit einem Faktor (F1) und zwei Merkmalen (ERT_86, D_AUFG).

Sowohl bei einer A-Serie als auch bei einem Einzelversuch wird die Formatvariable „\$ORTFmt“ für die Versuchsorte definiert. Dieses Format ordnet dem benutzerorientierten Ortskürzel die entsprechende Ortsbezeichnung zu.

Für die Faktoren sind folgenden Formate definiert:

Formatvariable	Bemerkung
FiFmt	Bezeichnung des Faktors i mit i=1..Anzahl der Faktoren
Fi_jFmt	Wert der Faktorvariablen j des Faktors i mit i=1..Anzahl der Faktoren und j= 1..Anzahl der Faktorvariablen
Fi_SjFmt	Wert der Sortengruppe j des Faktors i mit i=1..Anzahl der Faktoren und j= 1..Anzahl der Sortengruppen

Stehen keine ADS-Daten zur Verfügung so wird das Dataset DATEN nicht erstellt.

```
/*Datendefinition*/
```

```
DATA DATEN;
```

```
INFORMAT SERIE $15. ORTID $17. F1 WDH KUERZEL $15. NR1 $8. NR2 $8. D_AUFG  
DDMMYY8.;
```

```
INFORMAT ORT $5. ORTBEZ $30. VKENN $50.;
```

```
INFILE "temporäres Nutzerverzeichnis\PIAFSTAT\PIAFSTAT.DAT" LRECL=80;
```

```
INPUT VNR JAHR SERIE & ORTID & F1 WDH KUERZEL & NR1 & NR2 & ERT_86 & D_AUFG  
& PARZNR REIHE SPALTE ORT & ORTBEZ & VKENN & LAND BKR;
```

```
FORMAT D_AUFG EURDFDD8.;
```

```
/*Label zuordnen*/
```

```
Label
```

```
VNR          = "Versuch"  
JAHR         = "Jahr"  
SERIE        = "Serie"  
ORTID        = "OrtID"  
F1           = "Sorte"  
WDH          = "Wdh"  
KUERZEL      = "Kürzel"  
NR1          = "Nr. 1"  
NR2          = "Nr. 2"  
ERT_86       = "Kornertrag bei 86% TS dt/ha"  
D_AUFG       = "Datum des Aufgangs"  
PARZNR       = "Parzelle"  
REIHE        = "Reihe"  
SPALTE       = "Spalte"  
ORT          = "Ort"  
ORTBEZ       = "Ortsbezeichnung"  
VKENN        = "Kennung" ;  
LAND         = "Land" ;  
BKR          = "Bodenklimaraum" ;
```

```
RUN;
```

```
/*A-Serie: Wertelabel für die Orte*/
```

```
PROC FORMAT;
```

```
VALUE $ORTFmt
```

```
'514' = 'Grafenreuth'
```

```
' '    = 'Haar'
```

```
'612' = 'Bad Windsheim';
```

```
/*Wertelabel für die Faktoren und Faktorvariablen*/
```

```
PROC FORMAT;
```

```
VALUE F1Fmt
```

```
1 = 'Eiffel II'
```

```
2 = 'Classic'
```

```

3   = 'Miami'
4   = 'Nitouche'
5   = 'Metaxa'
6   = 'Granada'
7   = 'Sponsor'
8   = 'Power'
9   = 'Pinocchio'
10  = 'Attika'
11  = 'Madonna'
12  = 'Catania'
13  = 'Phönix';
VALUE F1_1Fmt
1   = 'EF 00288'
2   = 'EF 00374'
3   = 'EF 00352'
4   = 'EF 00366'
5   = 'EF 00393'
6   = 'EF 00423'
7   = 'EF 00426'
8   = 'EF 00439'
9   = 'EF 00441'
10  = 'EF 00444'
11  = 'EF 00445'
12  = 'EF 00447'
13  = '';
VALUE F1_S1Fmt
1   = 'ZE'
2   = 'ZE'
3   = 'ZE'
4   = 'ZE'
5   = 'ZA'
6   = 'ZE'
7   = 'ZE'
8   = 'ZE'
9   = 'P55'
10  = 'P55'
11  = 'ZA'
12  = 'ZA'
13  = 'ZA';
VALUE F1_S2Fmt
1   = 'TE'
2   = 'TE'
3   = 'TE'
4   = 'TE'
5   = 'TA'
6   = 'TE'
7   = 'TE'
8   = 'TE'
9   = 'QA'
10  = 'AS'
11  = 'ZD'
12  = 'ZD'
13  = 'ZD';
RUN;

```

2.3.10.1 R

Für den Zugriff auf die Daten werden die Dataframes DATEN und DATEN_LABEL erstellt. Diese Dataframes sind bezüglich der Anzahl der Faktoren und Merkmale dynamisch aufgebaut. Das folgende Beispiel bezieht sich auf einen Versuch mit 2 Faktoren und 23 Merkmalen.

Sowohl bei einer A-Serie als auch bei einem Einzelversuch wird der Dataframe ORT_FMT für die Versuchsorte definiert. Dieser Dataframe ordnet dem benutzerorientierten Ortskürzel die entsprechende Ortsbezeichnung zu.

Für die Faktoren sind folgenden Dataframes definiert:

Formatvariable	Bemerkung
Fi_Fmt	Bezeichnung des Faktors i mit i=1..Anzahl der Faktoren
Fi_j_Fmt	Wert der Faktorvariablen j des Faktors i mit i=1..Anzahl der Faktoren und j= 1..Anzahl der Faktorvariablen
Fi_Sj_Fmt	Wert der Sortengruppe j des Faktors i mit i=1..Anzahl der Faktoren und j= 1..Anzahl der Sortengruppen

Stehen keine ADS-Daten zur Verfügung so werden die Dataframes DATEN und DATEN_LABEL nicht erstellt.

```
# Datendefinition
tmpcolnames <- c(
  "VNR",
  "JAHR",
  "SERIE",
  "ORTID",
  "F1",
  "F2",
  "WDH",
  "KUERZEL",
  "NR1",
  "NR2",
  "AUF_GANG",
  "LAG_BLUE",
  "MNGLAUF",
  "MNGLN_WI",
  "PFLALFDM_5",
  "PFLALFDM_6",
  "PFL_QM_7",
  "PFL_QM_8",
  "PFLALANG",
  "SCLEROTI",
  "ALT_ERNA",
  "ERTR_KG",
  "TS_KORN",
  "ERTR91DT",
  "FETT91",
  "TKG_LUTR",
  "GLUC91",
  "SE_ERT",
  "WUECH",
  "MNGLV_WI",
  "LAGER_21",
  "LAGER_22",
  "PHO_MA19",
  "PARZNR",
  "REIHE",
  "SPALTE",
  "ORT",
  "ORTBEZ",
  "VKENN",
```

```

"LAND",
"BKR",
"BKRBEZ"
)

tmpcolClasses <-c(
  "integer",
  "integer",
  "character",
  "character",
  "integer",
  "integer",
  "integer",
  "character",
  "character",
  "character",
  "character",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "numeric",
  "integer",
  "integer",
  "integer",
  "character",
  "character",
  "character",
  "integer",
  "integer",
  "character"
)

DATEN <- read.table(
  "temporäres Nutzerverzeichnis/PIAFSTAT.RDAT",
  header      = FALSE,
  sep         = '\\t',
  col.names   = tmpcolnames,
  colClasses  = tmpcolClasses,
  na.string   = ".",
  fileEncoding = "latin1"
)

DATEN_LABEL <- c(

```

```

"VNR"      = "Versuchsnummer",
"JAHR"     = "Jahr",
"SERIE"    = "Serie",
"ORTID"    = "OrtID",
"F1"       = "Intensität",
"F2"       = "Sorte",
"WDH"      = "Wdh",
"KUERZEL"  = "Kürzel",
"NR1"      = "Nr. 1",
"NR2"      = "Nr. 2",
"AUF_GANG" = "Datum des Aufgangs",
"LAG_BLUE" = "Lager bei/nach Blüte",
"MNGLAUF"  = "Mängel im Stand nach Aufgang",
"MNGLN_WI" = "Mängel im Stand nach Winter",
"PFLALFDM_5" = "Pflanzenzahl je lfd. m in der Reihe",
"PFLALFDM_6" = "Pflanzenzahl je lfd. m in der Reihe",
"PFL_QM_7"  = "Bestandesdichte",
"PFL_QM_8"  = "Bestandesdichte",
"PFLALANG"  = "Pflanzenlänge zur Ernte cm",
"SCLEROTI"  = "Sclerotinia",
"ALT_ERNA"  = "Alternaria",
"ERTR_KG"   = "Kornertrag pro Parzelle kg",
"TS_KORN"   = "Trockensubstanz (Korn) %",
"ERTR91DT"  = "Kornertrag bei 91_% TS dt/ha",
"FETT91"    = "Öl bei 91_% TS (%)",
"TKG_LUTR"  = "Tausendkornmasse, lufttrocken g",
"GLUC91"    = "Glucosinolatgehalt bei 91_% TS (%)",
"SE_ERT"    = "Standardfehler des Mittelw. d. Ertrages",
"WUECH"     = "Wüchsigkeit",
"MNGLV_WI"  = "Mängel im Stand vor Winter",
"LAGER_21"  = "Lager",
"LAGER_22"  = "Lager",
"PHO_MA19"  = "Phoma (Parzellenbonitur)",
"PAR_ZNR"   = "Parzelle",
"REIHE"     = "Reihe",
"SPALTE"    = "Spalte",
"ORT"       = "Ort",
"ORTBEZ"    = "Ortsbezeichnung",
"VKENN"     = "Kennung",
"LAND"      = "Land",
"BKR"       = "Bodenklimaraum",
"BKRBEZ"    = "Bodenklimaraumbezeichnung"

```

```
)
```

```
# Wertelabel für die Orte
```

```
ORT_FMT <- c(
  "10" = "Biestow"
)
```

```
# Wertelabel für die Faktoren und Faktorvariablen
```

```
F1_FMT <- c(
  "1" = "ohne Behandlung",
  "2" = "mit Behandlung"
)
```

```
F1_1_FMT <- c(
  "1" = "",
  "2" = ""
)
```

```
F1_2_FMT <- c(
  "1" = "",
  "2" = ""
)
```

```

)
F1_S1_FMT <- c(
  "1" = "",
  "2" = ""
)
F1_S2_FMT <- c(
  "1" = "",
  "2" = ""
)
F2_FMT <- c(
  "1" = "Elektra",
  "2" = "Lorenz",
  "3" = "NK Nemax",
  "4" = "Ladoga",
  "5" = "Adriana",
  "6" = "Kadore",
  "7" = "PR45D03",
  "8" = "Visby",
  "9" = "Vision",
  "10" = "Horus",
  "11" = "NK Petrol",
  "12" = "NK Rapster",
  "13" = "PR 46 W 15",
  "14" = "Dimension",
  "15" = "Hammer",
  "16" = "Galileo",
  "17" = "Goya",
  "18" = "PR 46 W 20",
  "19" = "PR 45 D 04",
  "20" = "Xenon",
  "21" = "Compass",
  "22" = "Treffer"
)
F2_1_FMT <- c(
  "1" = "RAW 01647",
  "2" = "RAW 02152",
  "3" = "RAW 02183",
  "4" = "RAW 02241",
  "5" = "RAW 02562",
  "6" = "RAW 02620",
  "7" = "RAW 02592",
  "8" = "RAW 02551",
  "9" = "RAW 02631",
  "10" = "RAW 02543",
  "11" = "RAW 02508",
  "12" = "RAW 02503",
  "13" = "RAW 02595",
  "14" = "RAW 02766",
  "15" = "RAW 02762",
  "16" = "RAW 02813",
  "17" = "RAW 02814",
  "18" = "RAW 02796",
  "19" = "RAW 02802",
  "20" = "RAW 02890",
  "21" = "RAW 02969",
  "22" = "RAW 02974"
)
F2_2_FMT <- c(
  "1" = "Elektra",
  "2" = "Lorenz",
  "3" = "NK Nemax",

```

```

"4" = "Ladoga",
"5" = "Adriana",
"6" = "Kadore",
"7" = "PR45D03",
"8" = "Visby",
"9" = "Vision",
"10" = "Horus",
"11" = "NK Petrol",
"12" = "NK Rapster",
"13" = "PR 46 W 15",
"14" = "Dimension",
"15" = "Hammer",
"16" = "Galileo",
"17" = "Goya",
"18" = "PR 46 W 20",
"19" = "PR 45 D 04",
"20" = "Xenon",
"21" = "Compass",
"22" = "Treffer"
)
F2_S1_FMT <- c(
  "1" = "H",
  "2" = "L",
  "3" = "L",
  "4" = "L",
  "5" = "L",
  "6" = "",
  "7" = "H",
  "8" = "H",
  "9" = "L",
  "10" = "H",
  "11" = "H",
  "12" = "L",
  "13" = "H",
  "14" = "H",
  "15" = "H",
  "16" = "L",
  "17" = "L",
  "18" = "H",
  "19" = "H",
  "20" = "H",
  "21" = "H",
  "22" = "H"
)
F2_S2_FMT <- c(
  "1" = "",
  "2" = "",
  "3" = "",
  "4" = "",
  "5" = "",
  "6" = "",
  "7" = "",
  "8" = "",
  "9" = "",
  "10" = "",
  "11" = "",
  "12" = "",
  "13" = "",
  "14" = "",
  "15" = "",
  "16" = "",

```

```

"17" = "",
"18" = "",
"19" = "",
"20" = "",
"21" = "",
"22" = ""
)

# temporäre Objekte löschen
rm(tmpcolnames)
rm(tmpcolClasses)

```

2.3.11 SE-Daten

2.3.11.1 SAS

Für den Zugriff auf die SE-Daten wird das Dataset DATENSE erstellt. Dieses Set ist bezüglich der Anzahl der Faktoren und Merkmale dynamisch aufgebaut. Das folgende Beispiel bezieht sich auf einen Versuch mit einem Faktor (F1) und zwei Merkmalen (ERT_86, D_AUFG).

Sowohl bei einer A-Serie als auch bei einem Einzelversuch wird die Formatvariable „\$ORTFmt“ für die Versuchsorte definiert. Dieses Format ordnet dem benutzerorientierten Ortskürzel die entsprechende Ortsbezeichnung zu.

Für die Faktoren sind folgenden Formate definiert:

Formatvariable	Bemerkung
FiFmt	Bezeichnung des Faktors i mit i=1..Anzahl der Faktoren
Fi_jFmt	Wert der Faktorvariablen j des Faktors i mit i=1..Anzahl der Faktoren und j= 1..Anzahl der Faktorvariablen
Fi_SjFmt	Wert der Sortengruppe j des Faktors i mit i=1..Anzahl der Faktoren und j= 1..Anzahl der Sortengruppen

Stehen keine ADS-SE-Daten zur Verfügung so wird das Dataset DATENSE nicht erstellt.

```

/*SE-Datendefinition*/
DATA DATENSE;
INFORMAT SERIE $15. ORTID $17. F1 WDH KUERZEL $15. NR1 $8. NR2 $8. D_AUFG
DDMMYY8.;
INFORMAT ORT $5. ORTBEZ $30. VKENN $50.;

INFILE "temporäres Nutzerverzeichnis\PIAFSTAT\PIAFSTAT.DSE" LRECL=80;

INPUT VNR JAHR SERIE & ORTID & F1 WDH KUERZEL & NR1 & NR2 & ERT_86 & D_AUFG
& PARZNR REIHE SPALTE ORT & ORTBEZ & VKENN & LAND BKR;

FORMAT D_AUFG EURDFDD8.;

/*Label zuordnen*/
Label
VNR          = "Versuch"
JAHR         = "Jahr"
SERIE        = "Serie"
ORTID        = "OrtID"
F1           = "Sorte"

```



```

WDH          = "Wdh"
KUERZEL      = "Kürzel"
NR1          = "Nr. 1"
NR2          = "Nr. 2"
ERT_86       = "Kornertrag bei 86% TS dt/ha"
D_AÜFG       = "Datum des Aufgangs"
PARZNR       = "Parzelle"
REIHE        = "Reihe"
SPALTE       = "Spalte"
ORT          = "Ort"
ORTBEZ       = "Ortsbezeichnung"
VKENN        = "Kennung" ;
LAND         = "Land" ;
BKR          = "Bodenklimaraum" ;
RUN;

```

```
/*A-Serie: Wertelabel für die Orte*/
```

```
PROC FORMAT;
```

```
VALUE $ORTEFmt
```

```
'514' = 'Grafenreuth'
```

```
' ' = 'Haar'
```

```
'612' = 'Bad Windsheim';
```

```
/*Wertelabel für die Faktoren und Faktorvariablen*/
```

```
PROC FORMAT;
```

```
VALUE F1Fmt
```

```
1 = 'Eiffel II'
```

```
2 = 'Classic'
```

```
3 = 'Miami'
```

```
4 = 'Nitouche'
```

```
5 = 'Metaxa'
```

```
6 = 'Granada'
```

```
7 = 'Sponsor'
```

```
8 = 'Power'
```

```
9 = 'Pinochio'
```

```
10 = 'Attika'
```

```
11 = 'Madonna'
```

```
12 = 'Catania'
```

```
13 = 'Phönix';
```

```
VALUE F1_1Fmt
```

```
1 = 'EF 00288'
```

```
2 = 'EF 00374'
```

```
3 = 'EF 00352'
```

```
4 = 'EF 00366'
```

```
5 = 'EF 00393'
```

```
6 = 'EF 00423'
```

```
7 = 'EF 00426'
```

```
8 = 'EF 00439'
```

```
9 = 'EF 00441'
```

```
10 = 'EF 00444'
```

```
11 = 'EF 00445'
```

```
12 = 'EF 00447'
```

```
13 = '';
```

```
VALUE F1_S1Fmt
```

```
1 = 'ZE'
```

```
2 = 'ZE'
```

```
3 = 'ZE'
```

```
4 = 'ZE'
```

```
5 = 'ZA'
```

```
6 = 'ZE'
```

```
7 = 'ZE'
```

```

8   = 'ZE'
9   = 'P55'
10  = 'P55'
11  = 'ZA'
12  = 'ZA'
13  = 'ZA';
VALUE F1_S2Fmt
1   = 'TE'
2   = 'TE'
3   = 'TE'
4   = 'TE'
5   = 'TA'
6   = 'TE'
7   = 'TE'
8   = 'TE'
9   = 'QA'
10  = 'AS'
11  = 'ZD'
12  = 'ZD'
13  = 'ZD';
RUN;

```

2.3.11.1 R

Für den Zugriff auf die Daten werden die Dataframes DATENSE und DATENSE_LABEL erstellt. Diese Dataframes sind bezüglich der Anzahl der Faktoren und Merkmale dynamisch aufgebaut. Das folgende Beispiel bezieht sich auf einen Versuch mit 2 Faktoren und 23 Merkmalen.

Sowohl bei einer A-Serie als auch bei einem Einzelversuch wird der Dataframe ORT_FMT für die Versuchsorte definiert. Dieser Dataframe ordnet dem benutzerorientierten Ortskürzel die entsprechende Ortsbezeichnung zu.

Für die Faktoren sind folgenden Dataframes definiert:

Formatvariable	Bemerkung
Fi_Fmt	Bezeichnung des Faktors i mit i=1..Anzahl der Faktoren
Fi_j_Fmt	Wert der Faktorvariablen j des Faktors i mit i=1..Anzahl der Faktoren und j= 1..Anzahl der Faktorvariablen
Fi_Sj_Fmt	Wert der Sortengruppe j des Faktors i mit i=1..Anzahl der Faktoren und j= 1..Anzahl der Sortengruppen

Stehen keine ADS-Daten zur Verfügung so werden die Dataframes DATENSE und DATENSE_LABEL nicht erstellt.

```

# SE-Datendefinition
tmpcolnames <- c(
  "VNR",
  "JAHR",
  "SERIE",
  "ORTID",
  "F1",
  "F2",
  "WDH",
  "KUERZEL",
  "NR1",
  "NR2",
  "AUF_GANG",

```



```

"numeric",
"numeric",
"numeric",
"numeric",
"numeric",
"numeric",
"integer",
"integer",
"integer",
"character",
"character",
"character",
"integer",
"integer",
"character"
)

DATENSE <- read.table(
  "TEMPORÄRES NUTZERWERZEICHNIS/PIAFSTAT.RDSE",
  header      = FALSE,
  sep         = '\t',
  col.names   = tmpcolnames,
  colClasses  = tmpcolClasses,
  na.string   = ".",
  fileEncoding = "latin1"
)

DATENSE_LABEL <- c(
  "VNR"      = "Versuchsnummer",
  "JAHR"     = "Jahr",
  "SERIE"    = "Serie",
  "ORTID"    = "OrtID",
  "F1"       = "Intensität",
  "F2"       = "Sorte",
  "WDH"      = "Wdh",
  "KUERZEL"  = "Kürzel",
  "NR1"      = "Nr. 1",
  "NR2"      = "Nr. 2",
  "AUF_GANG" = "Datum des Aufgangs",
  "LAG_BLUE" = "Lager bei/nach Blüte",
  "MNGLAUFG" = "Mängel im Stand nach Aufgang",
  "MNGLV_WI" = "Mängel im Stand nach Winter",
  "PFLALFDM_5" = "Pflanzenzahl je lfd. m in der Reihe",
  "PFLALFDM_6" = "Pflanzenzahl je lfd. m in der Reihe",
  "PFL_QM_7"  = "Bestandesdichte",
  "PFL_QM_8"  = "Bestandesdichte",
  "PFLALANG"  = "Pflanzenlänge zur Ernte cm",
  "SCLEROTI"  = "Sclerotinia",
  "ALT_ERNA"  = "Alternaria",
  "ERTR_KG"   = "Kornertrag pro Parzelle kg",
  "TS_KORN"   = "Trockensubstanz (Korn) %",
  "ERTR91DT"  = "Kornertrag bei 91 % TS dt/ha",
  "FETT91"    = "Öl bei 91 % TS (%)",
  "TKG_LUTR"  = "Tausendkornmasse, lufttrocken g",
  "GLUC91"    = "Glucosinolatgehalt bei 91 % TS (%)",
  "SE_ERT"    = "Standardfehler des Mittelw. d. Ertrages",
  "WUECH"     = "Wüchsigkeit",
  "MNGLV_WI"  = "Mängel im Stand vor Winter",
  "LAGER_21"  = "Lager",
  "LAGER_22"  = "Lager",
  "PHO_MA19"  = "Phoma (Parzellenbonitur)",

```

```

"PARZNR"      = "Parzelle",
"REIHE"       = "Reihe",
"SPALTE"      = "Spalte",
"ORT"         = "Ort",
"ORTBEZ"      = "Ortsbezeichnung",
"VKENN"       = "Kennung",
"LAND"        = "Land",
"BKR"         = "Bodenklimaraum",
"BKRBEZ"      = "Bodenklimaraumbezeichnung"
)

# Wertelabel für die Orte
ORT_FMT <- c(
  "10" = "Biestow"
)

# Wertelabel für die Faktoren und Faktorvariablen
F1_FMT <- c(
  "1" = "ohne Behandlung",
  "2" = "mit Behandlung"
)
F1_1_FMT <- c(
  "1" = "",
  "2" = ""
)
F1_2_FMT <- c(
  "1" = "",
  "2" = ""
)
F1_S1_FMT <- c(
  "1" = "",
  "2" = ""
)
F1_S2_FMT <- c(
  "1" = "",
  "2" = ""
)
F2_FMT <- c(
  "1" = "Elektra",
  "2" = "Lorenz",
  "3" = "NK Nemax",
  "4" = "Ladoga",
  "5" = "Adriana",
  "6" = "Kadore",
  "7" = "PR45D03",
  "8" = "Visby",
  "9" = "Vision",
  "10" = "Horus",
  "11" = "NK Petrol",
  "12" = "NK Rapster",
  "13" = "PR 46 W 15",
  "14" = "Dimension",
  "15" = "Hammer",
  "16" = "Galileo",
  "17" = "Goya",
  "18" = "PR 46 W 20",
  "19" = "PR 45 D 04",
  "20" = "Xenon",
  "21" = "Compass",
  "22" = "Treffer"
)

```

```

F2_1_FMT <- c(
  "1" = "RAW 01647",
  "2" = "RAW 02152",
  "3" = "RAW 02183",
  "4" = "RAW 02241",
  "5" = "RAW 02562",
  "6" = "RAW 02620",
  "7" = "RAW 02592",
  "8" = "RAW 02551",
  "9" = "RAW 02631",
  "10" = "RAW 02543",
  "11" = "RAW 02508",
  "12" = "RAW 02503",
  "13" = "RAW 02595",
  "14" = "RAW 02766",
  "15" = "RAW 02762",
  "16" = "RAW 02813",
  "17" = "RAW 02814",
  "18" = "RAW 02796",
  "19" = "RAW 02802",
  "20" = "RAW 02890",
  "21" = "RAW 02969",
  "22" = "RAW 02974"
)
F2_2_FMT <- c(
  "1" = "Elektra",
  "2" = "Lorenz",
  "3" = "NK Nemax",
  "4" = "Ladoga",
  "5" = "Adriana",
  "6" = "Kadore",
  "7" = "PR45D03",
  "8" = "Visby",
  "9" = "Vision",
  "10" = "Horus",
  "11" = "NK Petrol",
  "12" = "NK Rapster",
  "13" = "PR 46 W 15",
  "14" = "Dimension",
  "15" = "Hammer",
  "16" = "Galileo",
  "17" = "Goya",
  "18" = "PR 46 W 20",
  "19" = "PR 45 D 04",
  "20" = "Xenon",
  "21" = "Compass",
  "22" = "Treffer"
)
F2_S1_FMT <- c(
  "1" = "H",
  "2" = "L",
  "3" = "L",
  "4" = "L",
  "5" = "L",
  "6" = "",
  "7" = "H",
  "8" = "H",
  "9" = "L",
  "10" = "H",
  "11" = "H",
  "12" = "L",

```

```

"13" = "H",
"14" = "H",
"15" = "H",
"16" = "L",
"17" = "L",
"18" = "H",
"19" = "H",
"20" = "H",
"21" = "H",
"22" = "H"
)
F2_S2_FMT <- c(
  "1" = "",
  "2" = "",
  "3" = "",
  "4" = "",
  "5" = "",
  "6" = "",
  "7" = "",
  "8" = "",
  "9" = "",
  "10" = "",
  "11" = "",
  "12" = "",
  "13" = "",
  "14" = "",
  "15" = "",
  "16" = "",
  "17" = "",
  "18" = "",
  "19" = "",
  "20" = "",
  "21" = "",
  "22" = ""
)

# temporäre Objekte löschen
rm(tmpcolnames)
rm(tmpcolClasses)

```

2.4 Programmkonfiguration

2.4.1 Programmparameter (/INI, /BIB und /RUN)

PIAFStat kann mit zusätzlichen Parametern aufgerufen werden:

Parameter	Beschreibung
Keine	PIAFStat wird im Dialogmodus ausgeführt.
/INI <i>Dateiname</i>	Die Programmkonfiguration wird aus der angegebenen Datei gelesen und in die Windows-Registry übertragen.
/BIB <i>Dateiname</i>	PIAFStat listet in der angegebenen Datei alle für den Hintergrundmodus verfügbaren Verfahren der Standardbibliothek auf.
/RUN <i>Dateiname</i>	PIAFStat wird im Hintergrundmodus mit der angegebenen Steuerdatei ausgeführt.

Anmerkung:

Dateinamen sind mit vollständigen Verzeichnispfad anzugeben. Befinden sich in dem vollständigen Dateinamen Leerzeichen so ist der Dateiname in Hochkommata einzuschließen.

2.4.2 Ini-Datei (Programmparameter /INI)

Mit dem Programmparameter /INI <dateiname> liest PIAFStat die angegebene Konfigurationsdatei ein und überträgt deren Einträge in die Windows-Registry unter den Schlüssel **HKEY_CURRENT_USER\Software\BioMath\PiafStat**. Die Konfigurationsdatei hat die folgende Struktur:

Schlüssel	Name	Wert
Lizenz	Nutzername	Name des Nutzers (siehe Anmerkung unten)
	Firma	Name der Einrichtung
	Bundesland	Name des Bundeslandes
Verzeichnis	SAS	Vollständige Angabe des SAS-Programms
	R	Vollständige Angabe des R-Programms
	ADS	Verzeichnis der ADS
	RES	Verzeichnis der RES
	ExportImport	zuletzt verwendetes Verzeichnis für den Import und Export von Verfahren
	Tmp	Temporäres Verzeichnis
Parameter	SAS	Zusätzliche Parameter beim Aufruf von SAS
	R	Zusätzliche Parameter beim Aufruf von R
System	ADSGlobalUpdate	Dateiname der globalen ADS-Updatedatei
	UseADSGlobalUpdate	Aktivierung der Nutzung der globalen ADS-Updatedatei {0=keine Aktivierung, 1=Aktivierung}

Anmerkung: PIAFStat verwendet als Standard den Windows-Nutzernamen. Durch einen Eintrag in der Konfigurationsdatei kann dieser Name (nur für PIAFStat) überschrieben werden.

Beispiel SAS:

```
[Lizenz]
Firma=BioMath GmbH
Bundesland=Mecklenburg-Vorpommern

[Verzeichnis]
SAS=C:\PROGRAMME\SAS INSTITUTE\SAS\V9\SAS.EXE
ADS=C:\PROJEKTE\PIAFSTAT\DATA\
RES=C:\PROJEKTE\PIAFSTAT\DATA\
ExportImport=C:\EIGENE DATEIEN\
Tmp=C:\Temp\

[Parameter]
SAS=

[System]
ADSGlobalUpdate=C:\EIGENE DATEIEN\
UseADSGlobalUpdate=1
```


Beispiel R:

```
[Lizenz]
Firma=BioMath GmbH
Bundesland=Mecklenburg-Vorpommern

[Verzeichnis]
R=C:\PROGRAMME\R\R-4.0.4\BIN\R.EXE
ADS=C:\PROJEKTE\PIAFSTAT\DATA\
RES=C:\PROJEKTE\PIAFSTAT\DATA\
ExportImport=C:\EIGENE DATEIEN\
Tmp=C:\Temp\

[Parameter]
R=

[System]
ADSGlobalUpdate=C:\EIGENE DATEIEN\
UseADSGlobalUpdate=1
```

2.4.3 Windows-Registry

Alle Programmeinstellungen werden in der Windows-Registry unter dem Schlüssel:

```
HKEY_CURRENT_USER\Software\BioMath\PiafStat
```

gespeichert.

2.4.4 Verfahrensabfrage (Programmparameter /BIB)

Mit dem Programmparameter „/BIB <dateiname>“ listet PIAFStat in der angegebenen Datei alle für den Hintergrundmodus verfügbaren Verfahren der Standardbibliothek auf:

Eintrag	Typ	Bemerkung
G-Name	A16	Bezeichnung der Gruppe
V-Name	A16	Bezeichnung des Verfahrens
V-Label	A64	Label des Verfahrens
EV-Flag	I	Verfahren für Einzelversuche {1 = ja, 0 = nein}
AS-Flag	I	Verfahren für Auswertungsserie {1 = ja, 0 = nein}

Anmerkung: Ai bedeutet – alphanumerisch: maximal i Zeichen; I – bedeutet numerisch ganzzahlig

Alle Einträge sind mit „Tab“ (09_{Hex}) getrennt und das Zeilenende ist mit „CR“ (0D_{Hex}0A_{Hex}) abgeschlossen.

Die Ausgabedatei wird bei jeder Abfrage neu erstellt. Ist diese Datei leer so verfügt die Standardbibliothek über keine Verfahren, die für den Hintergrundmodus geeignet sind. Die Verfahren können verschiedenen Gruppen zugeordnet sein, sie werden redundant für jede Gruppe vollständig gelistet.

Wurde die Verfahrensabfrage erfolgreich durchgeführt so ist der (Programm-)Exitcode gleich Null, anderenfalls ungleich Null. Die Liste der Exitcodes ist im Abschnitt „Hintergrundmodus – ExitCodes“ zu finden.

Bei der Verfahrensabfrage wird die Standardlogbuchdatei geführt.

2.4.5 Hintergrundmodus (Programmparameter /RUN)

Mit dem Programmparameter „/RUN <dateiname>“ wird PIAFStat im Hintergrundmodus ausgeführt und arbeitet die angegebene Steuerdatei ab. Die Steuerdatei besteht aus dem Abschnitt **[RUN]** mit Schlüsselwörtern und Wertzuweisung.

Es stehen folgende Schlüsselwörter zu Verfügung:

Schlüsselwort	Pflicht	Bemerkung
ADS	Nein	ADS-Verzeichnis (vollständiger Pfad) ⁸
RES	Ja	RES-Verzeichnis (vollständiger Pfad)
MODA	Ja	MoDa-Datei (nur Dateiname)
LGB	Ja	Logbuch-Datei (Dateiname mit vollständigem Pfad)
LST	Ja	List-Datei (Dateiname mit vollständigem Pfad)
LOG	Ja	Log-Datei (Dateiname mit vollständigem Pfad)
VNAME	Ja	Name des statistischen Verfahrens der Standardbibliothek

Anmerkung:

Pfad- und Dateinamen die Leerzeichen (20_{Hex}) enthalten sind in Hochkommata (22_{Hex}) einzuschließen.

Beispiel:

[RUN]

```
ADS      = C:\Projekte\Mais\V200X\  
RES      = C:\Projekte\Mais\V200X\  
MODA     = Mob.txt  
LGB      = C:\Projekte\Mais\Mon.lgb  
LST      = C:\Projekte\Mais\Mon.lst  
LOG      = C:\Projekte\Mais\Mon.log  
VNAME    = LIST
```

Ist der Eintrag für die Logbuchdatei ungültig, so wird die Standardlogbuchdatei verwendet.

Wurde die Steuerdatei erfolgreich verarbeitet so ist der (Programm-)Exitcode gleich Null, anderenfalls ungleich Null. Die Liste der Exitcodes ist im Abschnitt „Hintergrundmodus – ExitCodes“ zu finden.

Anmerkung:

Mithilfe der Stapelverarbeitung (Batch-Jobs) kann der Hintergrundmodus für die Automatisierung von Auswertungen genutzt werden. (Siehe Anhang Programmbeispiele)

2.4.6 Logbuch Sitzung PIAFStat

Während jeder PIAFStat-Sitzung wird ein Sitzungslogbuch geführt. Es befindet sich standardmäßig im nutzerspezifischen temporären Verzeichnis (piafstat.log). Die Logbuchdatei ist im ASCII-Format und hat folgenden Zeilenaufbau:

⁸ Ist kein ADS-Verzeichnis angegeben wird das Verfahren ohne ADS-Daten ausgeführt.

Spalte	Eintrag
1-10	Datum im Format dd.mm.yyyy
12-20	Uhrzeit im Format hh:mm:ss
22-Zeilenende	Logbucheintrag

2.4.7 Logbuch Eingaben des Dialoges SAS/R ausführen

Während jeder PIAFStat-Sitzung wird ein Logbuch mit den Eingaben des Dialoges *Sas/R ausführen* geführt. Das Logbuch wird nach Beendigung der Sitzung als Datei PIAFSTATSASRUNLGB.TXT im RES-Verzeichnis gespeichert. Die Datei ist eine ASCII-Datei und listet chronologisch das SAS-Dataset bzw. R-Dataframe PVAZUORD der Runs auf (siehe Abschnitt 2.3.9). Das SAS-Dataset bzw. R-Dataframe enthält alle Informationen zu den im Dialog *Sas/R ausführen* ausgewählten A-Merkmalen, K-Merkmalen und Optionen.

Jeder Run wird in einem Abschnitt des Logbuchs mit dem folgenden Aufbau gespeichert:

Zeile	Inhalt
1	Datum und Uhrzeit im Format dd.mm.yyyy hh:mm:ss
2	Namen der Variablen des SAS-Datasets bzw. R-Dataframes (Trennzeichen: Tabulator)
3 bis Blockende	Inhalte der Variablen des SAS-Datasets bzw. R-Dataframes (Trennzeichen: Tabulator)

2.4.8 Temporäres Verzeichnis für PIAFStat

Im temporären Verzeichnis erfolgt die Zwischenspeicherung der Arbeitsschritte. Als Standard wird das von Windows festgelegte nutzerspezifische temporäre Verzeichnis verwendet. Für die Spezifikation des temporären Verzeichnisses nutzt Windows als erstes die Umgebungsvariable TMP und wenn diese nicht definiert wurde die Variable TEMP. Eine Änderung des temporären Verzeichnisses kann in den Optionen erfolgen.

Anmerkung: Bei Windows7 kann nur über den systeminternen Namen auf das nutzerspezifische temporäre Verzeichnis zugegriffen werden.

2.5 Dateistruktur

Die Datenstruktur nach der vollständigen Installation von PIAFStat im Default-Verzeichnis: C:\Programme\BioMath\PIAFStat.

Verzeichnis: C:\Programme\BioMath\ PIAFStat

Programmverzeichnis

Datei	Bemerkung
_DISREG.ISR	Registryinformation
_ISREG32.DLL	Programmbibliothek
DELSL1.ISU	Uninstall information
PIAFSTAT.EXE	Hauptanwendung
PIAFSTAT.CHM	Hilfetext
PIAFSTAT.INI	Default Konfigurationsdatei
PIAFSTAT.BMR	Zusätzliche Konfigurationsdatei
SASDEFAULT.TXT	Default Syntaxtext für SAS und R

Datei	Bemerkung
HANDBUCH.PDF	Handbuch
REGISTRIERUNG.PDF	Formular zur Registrierung

Verzeichnis: C:\<programmdaten-verzeichnis>\BioMath\PIAFStat\Vbibl
Verfahrensbibliothek

Datei	Bemerkung
PIAFSTAT.VBL	Verfahrensbibliothek
PIAFSTAT.OLD ¹⁰	Sicherung der Verfahrensbibliothek nach erfolgreicher Komprimierung

Aufbau der Tabellen:

Tabelle: VERFAHREN

Feldname	Typ	Anmerkung
Vname	A16	Verfahrensname: Primärschlüssel
Vsort	A4	Sortierreihenfolge: Sekundärschlüssel
Label	A64	Label
Sas	Memo	Programmcode in SAS oder R
ENutzer	A64	Ersteller: Nutzernamen
Efirma	A64	Ersteller: Einrichtung
EBundesland	A64	Ersteller: Bundesland
LNutzer	A64	Letzter Bearbeiter: Nutzernamen
LFirma	A64	Letzter Bearbeiter: Einrichtung
LBundesland	A64	Letzter Bearbeiter: Bundesland
Autor	A64	Autor
Version	I	Versionsnummer
VDatum	D	Datum der letzten Änderung des Programmcode
VZeit	T	Uhrzeit der letzten Änderung des Programmcode
Zusatz1		Intern
Zusatz2		Intern
EVFlag	I	Verfahren geeignet für den Hintergrundmodus: Einzelversuche {0=nein, 1=ja}
ASFlag	I	Verfahren geeignet für den Hintergrundmodus: Auswertungsserie {0=nein, 1=ja}
RFlag	I	Programmiersprache {0=SAS, 1=R}
Infodatei	A255	Dateiname der Dokumentation

Tabelle: GRUPPEN

Feldname	Typ	Anmerkung
GName	A16	Gruppenname: Primärschlüssel
GSort	A4	Sortierreihenfolge: Sekundärschlüssel
Label	A64	Label

⁹ Ist das systemspezifische Programmdatenverzeichnis (z.B. WindowsXP: " C:\Dokumente und Einstellungen\All Users\Anwendungsdaten ", Windows7: "C:\ProgramData")

¹⁰ Ist nur nach erfolgreicher Komprimierung vorhanden.

Tabelle: GRUPPENVERFAHREN

Feldname	Typ	Anmerkung
GName	A16	Gruppenname: Primärschlüssel
VName	A16	Verfahrenname Primärschlüssel

Tabelle: FUNKTIONEN

Feldname	Typ	Anmerkung
Vname	A16	Funktionsname: Primärschlüssel
Vsort	A4	Sortierreihenfolge: Sekundärschlüssel
Label	A64	Label
Sas	Memo	Programmcode in SAS oder R
ENutzer	A64	Ersteller: Nutzernamen
Efirma	A64	Ersteller: Einrichtung
EBundesland	A64	Ersteller: Bundesland
LNutzer	A64	Letzter Bearbeiter: Nutzernamen
LFirma	A64	Letzter Bearbeiter: Einrichtung
LBundesland	A64	Letzter Bearbeiter: Bundesland
Autor	A64	Autor
Version	I	Versionsnummer
VDatum	D	Datum der letzten Änderung des Programmcode
VZeit	T	Uhrzeit der letzten Änderung des Programmcode
Zusatz1		Intern
Zusatz2		Intern

Verzeichnis: C:\<programmdaten-verzeichnis>\BioMath\PIAFStat\VBibl\Info

Die Dokumentationsdateien der Verfahren müssen in diesem Verzeichnis abgelegt werden.

2.6 Netzbetrieb

Erfolgte die Installation auf einem Netzlaufwerk, so können mehrere Nutzer gleichzeitig auf die Standard-Verfahrensbibliothek zugreifen. Die Standard-Verfahrensbibliothek befindet sich im Unterverzeichnis **VBibl** des Programm-Verzeichnisses. Die Dokumentationsdateien der Verfahren sind im VBibl-Unterverzeichnis **Info** abgelegt.

Durch den gemeinsamen Zugriff auf diese Datenbank ist es notwendig, dass die Hauptanwendung eine gewisse Aktualisierung durchführt. Diese Aktualisierung kann manuell über den Dialog Optionen erfolgen.

2.7 Speicherort der Standard-Verfahrensbibliothek

Der Speicherort der Standard-Verfahrensbibliothek ist vom Typ der Installation (Lokal und Netz) abhängig.

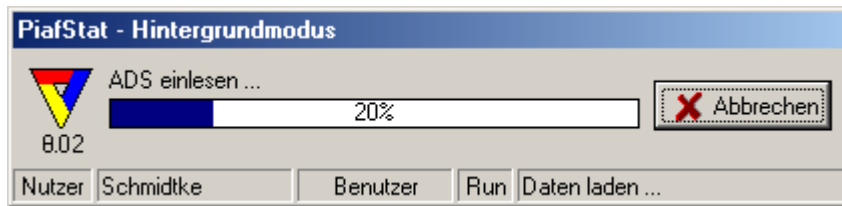
Im lokalen Betrieb befindet sich die Standard-Verfahrensbibliothek im betriebssystemspezifischen Programmdaten-Verzeichnis, im Netzbetrieb im Unterverzeichnis des Programm-Verzeichnisses.

Lokal: <laufwerk>\<programmdaten-verzeichnis>\BioMath\VBibl

Netz: <netzlaufwerk>\<pfad zum programm pifastat.exe>\VBibl

2.8 Hintergrundmodus – ExitCodes

PIAFStat wird mit dem Parameter /RUN <dateiname> im Hintergrundmodus aufgerufen und die Abarbeitungsschritte der Steuerdatei werden im Fenster „Hintergrundmodus“ angezeigt.



Der Abarbeitungsprozess kann mit dem Schalter „Abbrechen“ beendet werden. Bei Beendigung werden folgende Exit-Codes erzeugt:

Exitcode	Bemerkung
0	Die Steuerdatei konnte erfolgreich verarbeitet werden.
1	PIAFStat wurde nicht lizenziert bzw. registriert.
2	Nicht verwendet
3	PIAFStat wird bereits ausgeführt.
4	Die Standard-Verfahrensbibliothek ist nicht verfügbar.
5	PIAFStat konnte die Steuerdatei nicht vollständig und fehlerfrei verarbeiten. Die genaue Fehlerbeschreibung kann dem Logbuch entnommen werden.
6	Nutzerabbruch
7	Das nutzerspezifische temporäre Verzeichnis ist physisch nicht vorhanden.
11	SAS wurde mit Errorlevel 1 beendet – WARNING sind vorhanden
12	SAS wurde mit Errorlevel 2 beendet – ERROR sind vorhanden
13	R wurde mit Errorlevel 1 beendet – Fehler in Abarbeitung

Anmerkung:

Ist der Exitcode ungleich Null, so kann die genaue Fehlerbeschreibung dem Logbuch entnommen werden. Für den Hintergrundmodus wird die Logbuchdatei durch den entsprechenden Eintrag in der Steuerdatei festgelegt. Ist dieser Eintrag ungültig, so wird das Logbuch im nutzerspezifischen temporären Verzeichnis in der Datei „piafstat.log“ gespeichert.

3 Anlagen

3.1 Fehlermeldungen bei Programmstart

Fehlermeldung	Ursache
Lizenz-Benachrichtigungen	Im Programmverzeichnis befinden sich fehlerhafte Dateien. Setzen Sie sich mit der BioMath GmbH in Verbindung.
Registrierung-Benachrichtigungen	Die Zeit bis zur Registrierung ist abgelaufen. Fordern Sie die Registriernummer bei der BioMath GmbH an.
Die Nutzer-Verfahrensbibliothek im Verzeichnis: "C:\..." konnte nicht geöffnet werden. Es wird die Standard-Verfahrensbibliothek verwendet.	Das angegebene Verzeichnis oder die erforderliche Verzeichnis- und Dateistruktur ist ungültig. Stellen Sie sicher, dass im angegebenen Verzeichnis die Datei PIAFSTAT.VBL existiert und das Unterverzeichnis "Info" angelegt ist. Anmerkung: Der Verweis auf die Nutzer-VB wird automatisch im Optionen-Dialog deaktiviert.
Die Standard-Verfahrensbibliothek im Verzeichnis: "C:\..." konnte nicht geöffnet werden. PIAFStat wird beendet.	Der Zugriff auf die Standard-Verfahrensbibliothek ist nicht möglich. Stellen Sie sicher, dass im Bibliotheks-Verzeichnis "Vbibl" die Datei PIAFSTAT.VBL existiert und das Unterverzeichnis "Info" angelegt ist. Erscheint diese Fehlermeldung, obwohl die Datei- und Verzeichnisstruktur dem oben genannte entspricht, so ist die Datei PIAFSTAT.VBL defekt. Ersetzen Sie diese Datei durch eine Sicherungskopie.
Das nutzerspezifische temporäre Verzeichnis "C:\..." existiert nicht.'	Das nutzerspezifische temporäre Verzeichnis ist physisch nicht vorhanden, obwohl die Windows-Umgebungsvariable TEMP bzw. TMP definiert sind. Wenden Sie sich an den Systemadministrator.

3.2 Tastenkombinationen (Hotkey oder short cut)

Fenster	Hotkey	Befehl
Hauptfenster	Alt+D	Arbeitsbereich Daten
	Alt+L	Arbeitsbereich Log
	Alt+I	Arbeitsbereich Lis
	Alt+B	Arbeitsbereich Bibliothek
	Alt+O	Optionen
Arbeitsbereich Daten	F5	Daten erneut einlesen

Fenster	Hotkey	Befehl
	Strg+O	Daten öffnen
	Alt+S	Sas/R ausführen
	Strg+D	Details
Arbeitsbereich Log/Lis	Strg+A	Alles kopieren
	Strg+S	Speichern unter
	Strg+F	Suchen
	F3	Weitersuchen
	Strg+P	Drucken
Arbeitsbereich Bibliothek	Strg+I	Info-Dateien
	Strg+R	Gruppieren
	Strg+G	Referenzen
	F1	Info (Anzeige der Infodatei des markierten Verfahrens)

3.3 MDAC - Microsoft Data Access Components

Microsoft Data Access Components ist eine plattformübergreifende Technologie zum Zugriff auf Datenbanken. Ab Windows 2000 ist diese Technologie in das Betriebssystem integriert (MDAC Version 2.5).

PIAFStat verwaltet die Verfahrensbibliotheken mit der Microsoft Jet Engine 4.0, die Bestandteil der MDAC ist. Die aktuell installierte MDAC-Version wird im Info-Dialog von PIAFStat angezeigt.

3.4 Boolsche Algebra

3.4.1 Theorie

Die Boolsche Algebra geht von der Menge $\{0,1\}$ aus. Auf dieser Menge sind folgende drei Operationen definiert:

1. Die **Konjunktion (Und-Verknüpfung)** ist eine binäre Verknüpfung, hängt also von zwei Argumenten ab. Sie ist genau dann 1, wenn das erste *und* das zweite Argument 1 ist, und in jedem anderen Fall 0.
2. Auch die **Disjunktion (Oder-Verknüpfung)** ist eine binäre Verknüpfung. Sie ist genau dann 1, wenn das erste *oder* zweite Argument 1 ist, und sonst 0.
3. Die **Negation (Nicht-Operator)** verlangt nur ein Argument. Sie ist 0, wenn das Argument 1 ist, und 1, wenn das Argument 0 ist.

Diese drei Verknüpfungen können wie folgt mit Verknüpfungstabellen dargestellt werden.

Und	0	1
0	0	0
1	0	1

Oder	0	1
0	0	1
1	1	1

X	Nicht X
0	1
1	0

Um komplexere boolsche Ausdrücke zu erhalten, könne diese drei Operationen mehrfach hintereinander ausgeführt werden. Dabei ist zu beachten, dass die

Operationen unterschiedliche Prioritäten haben: „**Nicht**“ kommt vor „**Und**“, und „**Und**“ kommt vor „**Oder**“. Möchte man andere Prioritäten setzen, so muss man die entsprechenden Teilausdrücke in Klammern setzen.

Beispiel:

X	Y	X=A Und Y=B	X=A Oder Y=B	Nicht X=A Und Y=B	Nicht X=A Oder Y=B	X=A Und Nicht Y=B	X=A Oder Nicht Y=B	Nicht X=A Und Nicht Y=B	Nicht X=A Oder Nicht Y=B
A	A	0	1	0	0	1	1	0	1
A	B	1	1	0	1	0	1	0	0
B	A	0	0	0	1	0	1	1	1
B	B	0	1	1	1	0	0	0	1

Für die Operationen **Und**, **Oder** und **Nicht** gelten eine Reihe Rechengesetze wie Kommutativgesetz, Assoziativgesetz, Distributivgesetz usw.

Bezüglich des **Nicht**-Operators gelten die wichtigen Gesetze von de Morgan (x,y Elemente der Menge {0,1}):

1. **Nicht** (x **Und** y) = **Nicht** x **Oder** **Nicht** y
2. **Nicht** (x **Oder** y) = **Nicht** x **Und** **Nicht** y

3.4.2 Anwendung

Innerhalb der Deklaration der Syntaktischen Elemente für die A-Merkmale können die obigen Operationen für die automatische Bindung der Programmvariablen an die Datenvariablen verwendet werden. Die Deklaration lautet:

Ax(L:Label, I:Info, Feldname = Feldinhalt | Feldname = Feldinhalt,...,
Feldname = Feldinhalt, <Min Max>, AUTO)

Das Trennzeichen „|“ bedeutet eine **Und**-Verknüpfung, das Zeichen „|“ eine **Oder**-Verknüpfung. Als **Nicht**-Operator dient das Zeichen „!“, welches vor dem Feldinhalt stehen muss.

Beispiele:

1. Ax(L:Label, I:Info, Methode = !ZAEHL)
Es werden alle die Variablen ausgewählt deren Methode nicht "ZAEHL" ist.
2. Ax(L:Label, I:Info, Methode = !ZAEHL , Objekt = !PFLANZ)
Es werden alle die Variablen ausgewählt deren Methode nicht "ZAEHL" und dessen Objekt nicht "PFLANZ" ist.
3. Ax(L:Label, I:Info, Methode = !ZAEHL | Objekt = !KORN)
Es werden alle die Variablen ausgewählt deren Methode nicht "ZAEHL" oder dessen Objekt nicht "KORN" ist.

Var Name	Methode	Objekt	Bsp. 1	Bsp. 2	Bsp. 3
			Methode Nicht ZAEHL	Methode Nicht ZAEHL Und Objekt Nicht PFLANZ	Methode Nicht ZAEHL Oder Objekt Nicht KORN
V1	ZAEHL	PFLANZ	Nicht Ausgewählt	Nicht Ausgewählt	Ausgewählt
V2	ZAEHL	KORN	Nicht Ausgewählt	Nicht Ausgewählt	Nicht Ausgewählt
V3	ZAEHL	AEHRE	Nicht Ausgewählt	Nicht Ausgewählt	Ausgewählt
V4	BON19	PFLANZ	Ausgewählt	Nicht Ausgewählt	Ausgewählt
V5	BON19	KORN	Ausgewählt	Ausgewählt	Ausgewählt
V6	BON19	AEHRE	Ausgewählt	Ausgewählt	Ausgewählt

Durch Anwendung der Rechengesetze von de Morgan können die Formulierungen der Beispiele 2 und 3 wie folgt umgeformt werden:

Bsp. 2

Methode **Nicht** „ZAEHL“ **Und** Objekt **Nicht** „PFLANZ“ = **Nicht** (Methode gleich „ZAEHL“ **Oder** Objekt gleich „PFLANZ“)

*Es werden alle die Variablen **nicht** ausgewählt, deren Methode **gleich** „ZAEHL“ ist **oder** deren Objekt **gleich** „PFLANZ“ ist.*

Bsp. 3

Methode **Nicht** „ZAEHL“ **Oder** Objekt **Nicht** „KORN“ = **Nicht** (Methode gleich „ZAEHL“ **Und** Objekt gleich „KORN“)

*Es werden alle die Variablen **nicht** ausgewählt, deren Methode **gleich** „ZAEHL“ **und** dessen Objekt **gleich** „KORN“ ist.*

3.5 SAS

3.5.1 Exit Code

Nachdem automatischen SAS-Run wertet PIAFStat den SAS Exit Code (Return Code) aus. Ist diese Code Null wird automatisch die Listdatei angezeigt, anderenfalls die Logdatei.

3.5.2 Auszug aus der SAS-Dokumentation „SAS Companion for the Microsoft Windows Environment“

The return code for the completion of a SAS job is returned in the Windows batch variable, ERRORLEVEL. A value of 0 indicates normal termination. You can affect the value of ERRORLEVEL by using the ABORT statement. The ABORT statement takes an option argument, n, which is an integer. The ABORT statement also takes the RETURN or ABEND argument. If you issue these statements without specifying n, the ERRORLEVEL variable is set to the following values:

abort;	sets the ERRORLEVEL variable to 3.
abort return;	sets the ERRORLEVEL variable to 4.
abort abend;	sets the ERRORLEVEL variable to 5.

The n argument can range from 1 to 65,535. The ERRORLEVEL variable is used as a condition in the IF command in a Windows batch file. Refer to your Window's user's guide for more information on the ERRORLEVEL variable. The following table summarizes the values of the ERRORLEVEL variable.

Values for the ERRORLEVEL Variable

Condition	Severity	Return Code Value
All steps terminated normally	SUCCESS	0
SAS System issued warning(s)	WARNING	1
SAS issued error(s)	ERROR	2
User issued the ABORT statement	INFORMATIONAL	3
User issued the ABORT RETURN statement	FATAL	4
User issued the ABORT ABEND statement	FATAL	5
SAS internal error	INFORMATIONAL	6

TABLE NOTE 1:

If the SAS System terminates abnormally, the return code value is undefined. If a GP fault occurs, then the return code value is the same as the system error code for that fault.

3.5.3 SAS-Kommandos für zusätzliche Parameter

In der SAS-Dokumentation „Companion for Windows“ sind im Abschnitt „SAS System Options under Windows“ die möglichen SAS-Kommandos, die als zusätzliche Parameter in PIAFStat verwendet werden können, beschrieben.

Link zum Download des Dokumentes¹¹:

<http://support.sas.com/documentation/cdl/en/hostwin/67962/PDF/default/hostwin.pdf>

¹¹ Stand 08.05.2025

3.6 R

Ab Version 8 unterstützt PIAFStat auch R-Programme. Dazu ist in der Dialogbox „Verfahren bearbeiten“ auf der Seite „Details“ die Programmiersprache R auszuwählen.

Der Verfahrensabschnitt DEKLARATION wird für R genauso gehandhabt wie für SAS. Lediglich für den Programmcode in Blöcken und optionalen Blöcken ist zu beachten, dass jeder R-Befehl, der über mehrere Zeilen geht, mit einem Semikolon abgeschlossen werden muss.

Der Verfahrensabschnitt PROGRAMM wird für R genauso gehandhabt wie für SAS.

In SAS werden automatisch Log- und List-Dateien erzeugt, in R aber nur eine Ausgabedatei der abgearbeiteten R-Befehle. Diese Ausgabedatei wird in PIAFStat als Log-Datei behandelt. Soll für ein Verfahren in R auch eine List-Datei erzeugt werden, steht dafür die globale Variable LIS zur Verfügung. Mit dem R-Befehl `sink(LIS)` kann dann z.B. die Ausgabe in die List-Datei umgelenkt und mit dem Befehl `sink()` die Umlenkung wieder beendet werden.

PIAFStat ruft R im Hintergrund standardmäßig mit den folgenden zusätzlichen Parametern auf

- CMD BATCH
- --encoding=latin1
- --vanilla

Weitere Parameter können in den Optionen von PIAFStat festgelegt werden. Die vollständige Liste der zusätzlichen Parameter sind in „Introduction to R (Appendix B)“ zu finden.

Link zum Download des Dokumentes ¹²:

<https://cran.r-project.org/doc/manuals/R-intro.pdf>

Nachdem automatischen R-Run im Hintergrund wertet PIAFStat den Exit Code (Return Code) aus. Ist diese Code Null wird (falls vorhanden) automatisch die List-Datei angezeigt, anderenfalls die Log-Datei.

Anmerkung: R liefert als Exit Code nur die Null oder im Fehlerfall die Eins zurück. Eine Spezifikation des Fehlers ist mit dem Exit Code nicht möglich.

¹² Stand 08.05.2025

3.7 Layout-Schriftattribute

FColor

Wert	Bedeutung
DkGray	Dunkelgrün
Fuchsia	Fuchsia
Gray	Grau
Green	Grün
Lime	Lime
LtGray	Hellgrün
Maroon	Maroon
Navy	Marineblau
Olive	Olivgrün
Purple	Purpur
Red	Rot
Silver	Silber
Teal	Teal
White	Weiß
Yellow	Gelb
InactiveCaptionText	Aktuelle Farbe von Text in der Titelleiste von inaktiven Fenstern
BtnHighlight	Aktuelle Farbe von hervorgehobenem Text auf einer Schaltfläche
3DDkShadow	Dunkler Schatten von dreidimensional angezeigten Elementen
3DLight	Helle Farbe für dreidimensional angezeigte Elemente (für Ränder)
InfoText	Textfarbe für Kurzhinweise
InfoBk	Hintergrundfarbe für Kurzhinweise
Background	Aktuelle Hintergrundfarbe des Windows-Desktop
ActiveCaption	Aktuelle Farbe der Titelleiste aktiven Fenstern
InactiveCaption	Aktuelle Farbe der Titelleiste von inaktiven Fenstern
Menu	Aktuelle Hintergrundfarbe von Menüs
Window	Aktuelle Hintergrundfarbe von Fenstern
WindowFrame	Aktuelle Farbe von Fensterrahmen
MenuText	Aktuelle Farbe von Menütext
WindowText	Aktuelle Farbe von Fenstertext
CaptionText	Aktuelle Farbe von Text in der Titelleiste von aktiven Fenstern
ActiveBorder	Aktuelle Rahmenfarbe von aktiven Fenstern
InactiveBorder	Aktuelle Rahmenfarbe von inaktiven Fenstern
AppWorkSpace	Aktuelle Farbe der Arbeitsumgebung von Anwendungen
Highlight	Aktuelle Hintergrundfarbe von markiertem Text
HightlightText	Aktuelle Farbe von markiertem Text
BtnFace	Aktuelle Farbe von Schaltflächen
BtnShadow	Aktuelle Farbe des Schattens einer Schaltfläche
GrayText	Aktuelle Farbe von abgedunkeltem Text
BtnText	Aktuelle Farbe von Text auf einer Schaltfläche

3.8 Format für Word-Dateiname

In den Nutzeroptionen kann für die Übernahme der Log/Lis-Ausgaben in die Textverarbeitung Word das Format des Dateinamens mit Hilfe von Schlüsselwörtern festgelegt werden. Die Schlüsselwörter beziehen sich auf Variablen, die den ausgewerteten Versuch bzw. die Serie beschreiben. Die Schlüsselwörter sind in spitzen Klammern einzuschließen.

Schlüsselwort	Bedeutung
Verfahren	
<VERFAHRENNAME>	Name des statistischen Verfahrens
<VERFAHRENLABEL>	Label des statistischen Verfahrens
<VERFAHRENAUTOR>	Autor des statistischen Verfahrens
<VERFAHRENVERSION>	Versionsnummer des statistischen Verfahrens
<VERFAHRENDATUM>	Versionsdatum des statistischen Verfahrens im Format „ddmmyyyy“
<VERFAHRENZEIT>	Versionsuhrzeit des statistischen Verfahrens im Format „hhmmss“
<VERFAHRENERSTELLT>	Ersteller (Name Firma und Bundesland getrennt mit Komma) des statistischen Verfahrens
<VERFAHRENGEAENDERT>	Letzte Änderung (Angabe analog Ersteller)
Serie	
<ASNUMMER>	Nummer der A-Serie
<ASNAME>	Name der A-Serie
<ASJAHR>	Jahr der A-Serie
<ASANZVERS>	Anzahl der Versuche
<ASFRUCHTART>	Fruchtart
Versuch	
<VSERIE>	Serienbezeichnung
<VJAHR>	Jahr
<VORTBEZ>	Ortsbezeichnung
<VKENNUNG>	Versuchskennung
<VFRUCHTART>	Fruchtartbezeichnung
<VFRAGE>	Versuchsfrage
<VLAND>	Landesnummer
<VANLAGE>	Anlage
<VDETAIL>	Detail
<VORT>	Ortskürzel (Benutzer orientiert)
<VBKR>	Bodenklimaraum
<VWDH>	Anzahl der Wiederholungen
System	
<DATUM>	Aktuelles Systemdatum im Format „ddmmyyyy“
<UHRZEIT>	Aktuelle Systemzeit im Format „hhmmss“

Der generierte nutzerspezifische Word-Dateiname hat das Format:
 [Prefix]_[Zeichenkette mit ersetzten Schlüsselwörtern]_[Postfix]

Der Prefix ist entsprechend der Ausgabedatei entweder „Lis“ oder „Log“. Für den Postfix ist ein Zähler generiert, der bei jeder Word-Übernahme um eins erhöht wird. Dieser Zähler wird bei jedem Programmstart mit Null initialisiert.

Beispiel:

In den Nutzerspezifischen Optionen wurde für das Format des Word-Dateinamens die Zeichenkette:

<VJAHR>_<VORTBEZ>_Fruchtart_<VFRUCHTART>_<DATUM>

festgelegt. Nach einem aktuellen SAS/R-Run (am 28.04.2020) mit Daten eines Versuches aus dem Jahre 2019 in Tützpatz, wurde der folgende Lis-Dateiname generiert:

LIS_2019_Tützpatz_FRUCHTART_Winterweizen_28042020_1.doc

Anmerkung:

Die Abschnitte in der Zeichenkette die keine Schlüsselwörter sind, werden bei der Generierung des Dateinamens beibehalten. Nichtzulässige Zeichen für einen Dateinamen werden durch Leerzeichen ersetzt.

3.9 Optionen der Verfahren

Im Deklarationsteil der Verfahren wird für jedes optionale Element ein Default-Wert definiert. Im Dialogmodus gibt es die Möglichkeit diese Default-Werte nutzerspezifisch anzupassen oder auch permanent im Verfahren zu ändern. In Abhängigkeit von der verwendeten Verfahrensbibliothek und den Rechten des Nutzers gelten die folgenden Einschränkungen für die Speicherung der Optionen:

	Administrator		Nutzer	
	Optionen Permanent speichern	Optionen Nutzerspezifisch speichern	Optionen Permanent speichern	Optionen Nutzerspezifisch speichern
Standard-VB	Ja	Ja	Nein	Ja
Nutzer-VB	Ja	Nein	Ja	Nein

Anmerkung:

Wurde der Deklarationsteil des Verfahrens gesperrt, so können die Optionen nicht gespeichert werden.

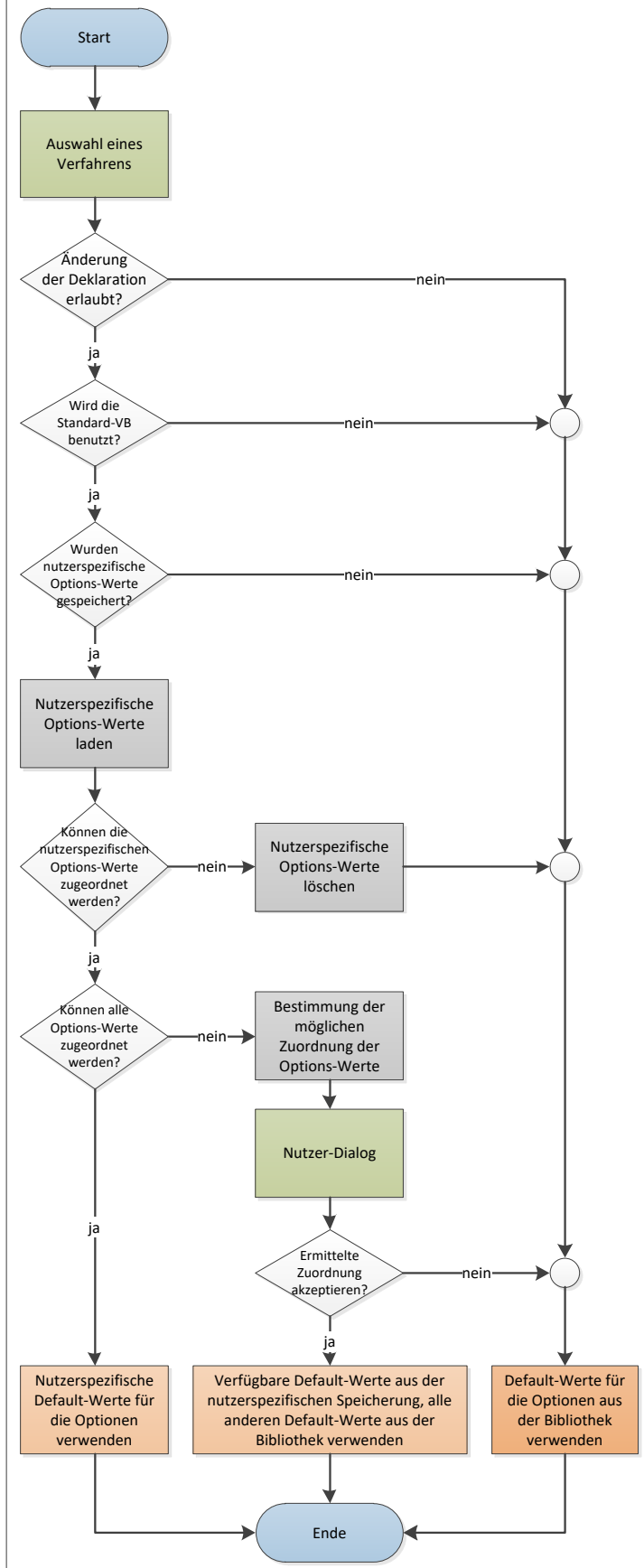
Die nutzerspezifische Speicherung der Optionen erfolgt in der Windows-Registry im Schlüssel HKEY_CURRENT_USER.

Es werden strukturiert gespeichert:

- Name des Verfahrens
- Bezeichner der Optionen und deren Zustände (ON oder OFF)
- Bezeichner der optionalen Blöcke und deren Inhalte

Die Listen von Konfigurationen der Optionen werden ebenfalls in der Windows-Registry im Schlüssel HKEY_CURRENT_USER abgelegt.

Einstellen der Default-Options-Werte nach Verfahrensauswahl



Nach Auswahl eines Verfahrens im Dialog „SAS/R ausführen“ wird das nebenstehende Struktogramm durchlaufen und die Options-Werte eingestellt.

Mit dieser Abarbeitungsfolge wird sichergestellt, dass gespeicherte nutzerspezifische Optionen auch genutzt werden können wenn Verfahren geändert wurden. Folgende Änderungen an den Optionen des Verfahrens führt zum Nutzer-Dialog:

- Bezeichner von Optionen werden geändert
- Optionen werden hinzugefügt oder entfernt
- Bezeichner von optionalen Blöcken werden geändert
- Optionale Blöcke werden hinzugefügt oder entfernt

Alle anderen Änderungen am Deklarations- und Programmteil haben keinen Einfluss auf das Handling der gespeicherten nutzerspezifischen Optionen.

3.10 Programmbeispiele

3.10.1 Verwendung von Funktionen

Die folgenden Beispiele zeigen die Anwendung von Funktionen. Die Codes in den Abschnitten PROGRAMM sind SAS-Codes, die aber durch R-Codes ersetzt werden können. Die Inhalte der Abschnitte DEKLARATION gelten sowohl für SAS als auch für R.

1. Beispiel

```
*****
Funktion = SORTK
*****
DEKLARATION
K1(L:Sortierung)
```

PROGRAMM

```
/*Sortieren nach K1*/
PROC SORT;
BY [K1];
RUN;
```

```
*****
Verfahren = LIST
*****
```

DEKLARATION

```
KSERIE(L:Serie, I:Wählen Sie bitte eine Versuchsserie.,M:1-1,serie,AUTO)
```

```
A1(L:Analysemerkmal, I:Bitte wählen Sie ein Merkmal deren Methode "WIEGEN"
ist., Methode=WIEGEN, AUTO)
```

```
PSORT(L:Sortierung, I:Sortierung nach Serie, F:SORTK, KSERIE=K1)
```

PROGRAMM

```
/*Sortieren nach KSerie*/
[PSort]
```

```
/*Liste*/
```

```
PROCPRINT;
VAR [KSERIE] [A1];
RUN;
```

```
*****
```

Es wird das folgende Programm dynamisch generiert:

DEKLARATION

```
KSERIE(L:Serie, I:Wählen Sie bitte eine Versuchsserie.,M:1-1,serie,AUTO)
```

```
A1(L:Analysemerkmal, I:Bitte wählen Sie ein Merkmal deren Methode "WIEGEN"
ist., Methode=WIEGEN, AUTO)
```

PROGRAMM

```
/*Sortieren nach K1*/
PROC SORT;
BY [KSERIE];
RUN;
```

```

/*Liste*/
PROCPRINT;
VAR [KSERIE] [A1];
RUN;

```

Im Verfahren „List“ werden zunächst den Programmvariablen KSERIE und A1 die entsprechenden Datenvariablen zugewiesen. Die Deklaration PSORT verweist auf das Funktion SORTK wobei durch die Zuweisung „KSERIE=K1“ in der Funktion SORTK an Stelle der Variable K1 die Variable KSERIE tritt. Es wird also erst nach KSERIE sortiert und dann die Variablen KSERIE und A1 gelistet.

2. Beispiel

```

*****
Funktion: MERGEN
*****

```

DEKLARATION

```
KX()
```

```

O12(L:Residuen nach Lageplan,I:Nur EINE der beiden Unteroptionen
wählen!!!,M:1-1, C: [O13][O14],ON)
O13(L:explorativer Ansatz oder!,I:Die Option O14 muss DEAKTIVIERT sein
siehe Info-Text!, C:outpredictm,ON)
O14(L:nach vollständigem Modell,I:Die Option O13 muss DEAKTIVIERT sein
siehe Info-Text! , C:outpredict,OFF)

```

PROGRAMM

```

/* Mergen der Urdaten - Reihe- und Spalte - mit Residuen aus OUTPM oder
OUTP*/

```

```

Proc sort data=daten;
  by [KX] wdh;
run;
Proc sort data=[O12];
  by [KX] wdh;
run;
DATA a;
  set DATEN;
  id = [KX]*1000+WDH*10;
run;
DATA b;
  set [O12];
  id = [KX]*1000+WDH*10;
run;
DATA ab;
  MERGE a b;
  by id;
  X='[O12]';
run;

```

```

*****
Verfahren: proc mixed + Mergen
*****

```

DEKLARATION

```
K1(L:.....)
```

```
PMERGEN(L: Mergen der Urdaten,F:MERGEN, K1=KX, OP1=O12,OP2=O13,OP3=O14)
```

PROGRAMM

```

proc mixed data=Daten ;
  class [K1] WDH ;
  model &MERKMAL = [K1] / OUTPRED=outpredict OUTPREDM=outpredictm;
  random WDH/s;
  lsmeans [K1] / alpha=&alpha pdiff cl;
* Erzeugen von output-Datasets (tables);
ods output tests3=vatab
           lsmeans=lsmeans
           covparms=covp
           solutionR=sol
           diffs=difftest
           ClassLevels=classlev;

run;

[PMERGEN]

*****

```

Es wird das folgende Programm dynamisch generiert:

DEKLARATION

K1(L:.....)

OP1(L:Residuen nach Lageplan,I:Nur EINE der beiden Unteroptionen
wählen!!!,M:1-1, C: [OP2][OP3],ON)
OP2(L:explorativer Ansatz oder!,I:Die Option O14 muss DEAKTIVIERT sein
siehe Info-Text!, C:outpredictm,ON)
OP3(L:nach vollständigem Modell,I:Die Option O13 muss DEAKTIVIERT sein
siehe Info-Text! , C:outpredict,OFF)

PROGRAMM

```

proc mixed data=Daten ;
  class [K1] WDH ;
  model &MERKMAL = [K1] / OUTPRED=outpredict OUTPREDM=outpredictm;
  random WDH/s;
  lsmeans [K1] / alpha=&alpha pdiff cl;
* Erzeugen von output-Datasets (tables);
ods output tests3=vatab
           lsmeans=lsmeans
           covparms=covp
           solutionR=sol
           diffs=difftest
           ClassLevels=classlev;

run;

/* Mergen der Urdaten - Reihe- und Spalte - mit Residuen aus OUTPM oder
OUTP*/

Proc sort data=daten;
  by [K1] wdh;
run;
Proc sort data=[OP1];
  by [K1] wdh;
run;
DATA a;
  set DATEN;
  id = [K1]*1000+WDH*10;
run;
DATA b;
  set [OP1];

```

```

id = [K1]*1000+WDH*10;
run;
DATA ab;
MERGE a b;
by id;
X='[OP1]';
run;

```

Im Verfahren „proc mixed + Mergen“ werden K1 und die zu verwendende Funktion MERGEN deklariert. In MERGEN wird KX durch K1 überladen, dem Verfahren „proc mixed + Mergen“ werden die Optionen mit den neuen Name OP1,OP2 und OP3 hinzugefügt.

3. Beispiel

Nicht deklarierte Blöcke und Optionen werden dem Verfahren nicht hinzugefügt. Damit ist es z.B. möglich, feste globale Blöcke zu verwenden.

```

*****
Funktion: VERZ
*****
DEKLARATION
BPFAD1 (C:C:\Tmp\)
BPFAD2 (C:C:\Tmp0815\)
BPFAD3 (C:C:\Doku\)
BPFAD4 (C:C:\Prog16\)

```

PROGRAMM

```

/**/
*****

```

Jeder hier deklarierte Block kann einzeln in einem Verfahren Verwendung finden:

Deklaration

```
PPfad (F:VERZ,BAUSGABE=BPFAD1)
```

In diesem Beispiel greift das Verfahren auf die Funktion VERZ zu und es wird in dem Verfahren ein Block mit der Bezeichnung BAUSGABE und dem Inhalt C:\Temp\ bereitgestellt.

4. Beispiel

Verzeichnisse für Ergebnisdateien.

```

*****
Funktion: ERGVERZ
*****
DEKLARATION
BPFADE (L:Verzeichnis für Ergebnisdateien, I:Pfad für Ergebnisdateien,
C:C:\ergebnis,OPTION)
OERG (L: Verzeichnis für Ergebnisdateien, I:Pfad für Ergebnisdateien,
C:[BPFADE],ON)
PROGRAMM
/**/
*****

```

Verfahren: RTFOUT

```

*****
DEKLARATION

```

```
PPFADERG(L: Verzeichnis zu den Ergebnisdateien, F: ERGVERZ,OPFADERG=OERG,  
BPFADERG=BPFADDE)
```

```
...
```

PROGRAMM

```
...
```

```
ods rtf file="[OPFADERG]\V&VSERIE.\V&VSERIE.M&VORT..doc" style=ms_style;
```

```
...
```

```
*****
```

Durch die Funktion ERGVERZ wird das Hauptverzeichnis für die Ergebnisdateien festgelegt. Im Verfahren RTFOUT kann zur Laufzeit dieses Verzeichnis verändert oder die Voreinstellung übernommen werden. Im Weiteren wird für jede Versuchsserie ein Unterverzeichnis angelegt.

3.10.2 Stapelverarbeitung (Batch-Jobs)

Das folgende Beispiel zeigt die Anwendung einer Stapelverarbeitung, wobei die Programmiersprache des Verfahrens (SAS oder R) automatisch erkannt und verarbeitet wird.

Beispiel

Eine ADS soll sowohl mit dem Verfahren VK als auch mit dem Verfahren A-BL ausgewertet werden.

Inhalt der Datei Job.bat:

```
C:\Progra~1\BioMath\PIAFStat\Piafstat.exe /RUN G:\SERIEN\PrgVK.txt  
C:\Progra~1\BioMath\PIAFStat\Piafstat.exe /RUN G:\SERIEN\PrgABL.txt
```

Inhalt der Datei PrgVK.txt:

[RUN]

```
ADS = C:\Programme\proPlant\piaf\ADS\  
RES = C:\Programme\proPlant\piaf\RES\  
MODA= moda.txt  
LGB = G:\Sta\PrgVK.lgb  
LST = G:\Sta\ PrgVK.lst  
LOG = G:\Sta\PrgVK.lgb  
VNAME = VK
```

Inhalt der Datei PrgABL.txt:

[RUN]

```
ADS = C:\Programme\proPlant\piaf\ADS\  
RES = C:\Programme\proPlant\piaf\RES\  
MODA= moda.txt  
LGB = G:\Sta\PrgABL.lgb  
LST = G:\Sta\ PrgABL.lst  
LOG = G:\Sta\PrgABL.lgb  
VNAME = A-BL
```

Die notwendigen Verfahren VK und A-BL müssen in der Standardverfahrensbibliothek für die Verwendung im Hintergrundmodus aktiviert sein. Die Dateien „PrgVK.txt“ und „PrgABL.txt“ sind im Verzeichnis G:\SERIEN zu speichern.