

Hausübung 1

Organisatorisches zur Abgabe

- Bitte geben Sie die Hausübung über Moodle ab. Bitte geben Sie nur ihre R-Syntax-Datei ab.
- Benennen Sie die R-Syntax-Datei bitte 'ProgStat15_H1_Nachname_Vorname.R', wobei Sie Nachname und Vorname durch Ihren eigenen Namen ersetzen.
- Gruppenarbeit von bis zu drei Personen ist möglich. Pro Gruppe bitte nur einmal abgeben! (Jede Hausübung kann prinzipiell in einer neuen Gruppe abgegeben werden.)
- Um den Schein zu erwerben, muss jede Hausübung einzeln positiv bestanden werden.
- Eine verbindliche Anmeldung zur Vorlesung (d.h. Scheinerwerb) erfolgt mit Abgabe der ersten Hausübung!
- **Letztmöglichster Abgabetermin: Mittwoch, den 27. Mai 2015 (23:59 Uhr)**

Vorbereitung

- Legen Sie folgende Ordnerstruktur an: Hauptordner 'H1' mit Unterordnern 'Daten', 'Programme', 'Ergebnisse'. Speichern Sie in 'Daten' Ihre Daten, in 'Programme' Ihr Programm und in 'Ergebnisse' Ihre Ergebnisse.
- Öffnen Sie eine neue Programmdatei und speichern Sie diese im Programme-Ordner unter folgendem Dateinamen: 'ProgStat15_H1_Nachname_Vorname.R', wobei Sie Nachname und Vorname durch Ihren eigenen Namen ersetzen. (Sie können auch die Vorlage auf der Homepage verwenden.)
- Geben Sie bitte in den ersten Zeilen der R-Datei die Namen aller Gruppenmitglieder an. (Ordnen Sie diese bitte möglichst alphabetisch nach dem Nachnamen. Verwenden Sie den ersten Namen als Dateinamen, falls Sie als Gruppe abgeben.)
- Verwenden Sie bitte den H1-Ordner als Working Directory.
- Schreiben Sie bitte zu jeder 'Lösung' die Aufgabennummer.
- Achten Sie darauf, dass Ihr Programm-Code grundsätzlich nachvollziehbar ist.
- Alle Teilschritte, die aufgrund eines fehlerhaften Programm-Codes eine nicht beabsichtigte Fehlermeldung erzeugen, werden mit 0 Punkten bewertet.
- **Geben Sie eine sinnvoll und ordentlich kommentierte Syntax ab.**

Aufgabe 1: Zum sos-Paket

(10 Punkte)

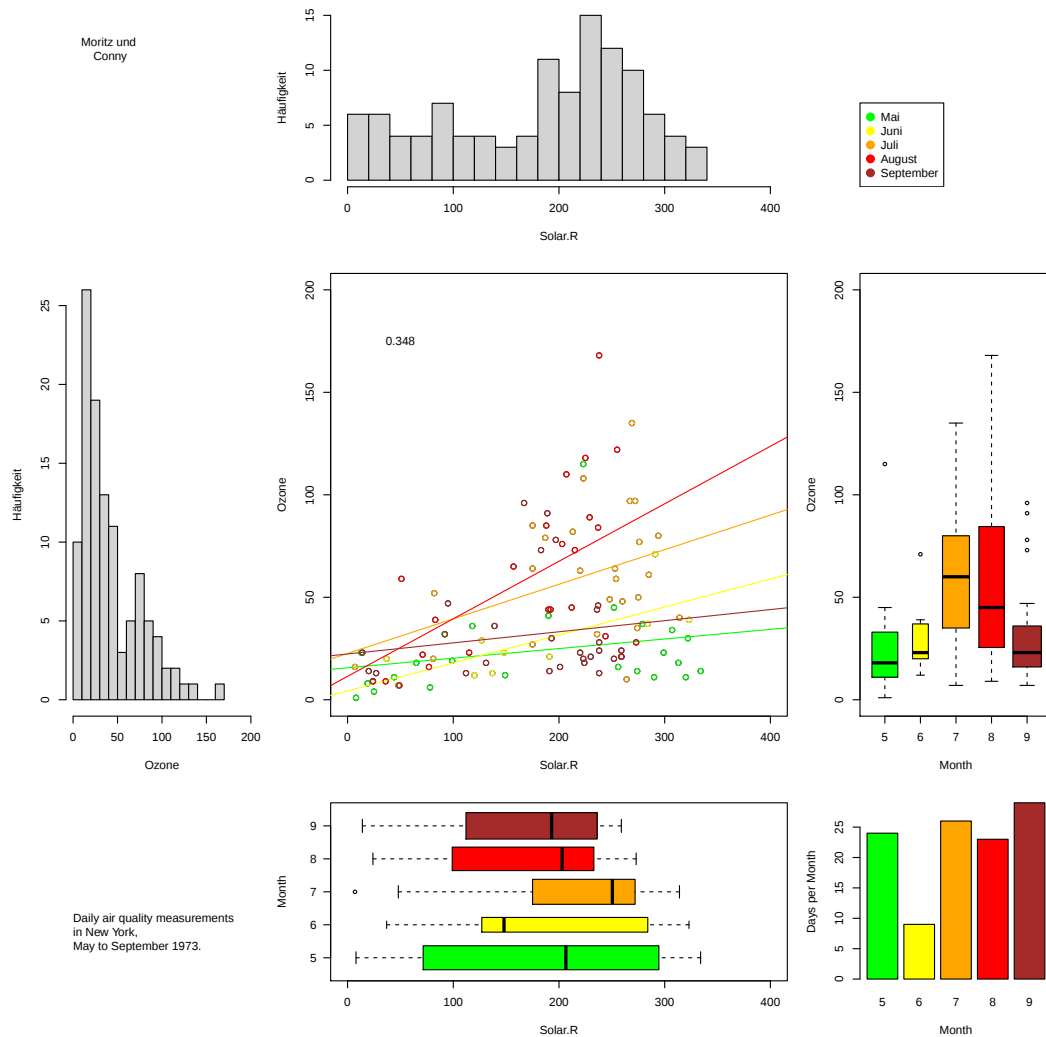
- a) Verwenden Sie das sos-Paket. Suchen Sie mit einer geeigneten Funktion daraus nach dem Begriff 'rasch model'.
- b) Speichern Sie in einem Vektor namens 'n-Treffer' die Anzahl an Treffern.
- c) Speichern Sie in einem Vektor namens 'Namen.Pakete' die Namen der gefundenen Pakete. Dabei soll jeder Paketname nur einmal in dem Vektor erscheinen. (Das Programm sollte so allgemein geschrieben sein, dass es unabhängig von der Anzahl an Treffern und deren Reihenfolge funktioniert.)
- d) Speichern Sie in einem Vektor namens 'n.Pakete' die Anzahl an verschiedenen Paketen, in denen der Suchbegriff gefunden wurde.
- e) Speichern Sie in einem Objekt namens 'letzten_20_Zeilen' die letzten 20 Zeilen des Suchergebnisses.
- f) Speichern Sie in einem Objekt namens 'genau_5-Treffer' diejenigen Zeilen des Suchergebnisses, die die Pakete mit genau 5 Treffern enthalten.
- g) Speichern Sie in einem Objekt namens 'zwischen_10_und_15-Treffer' diejenigen Zeilen des Suchergebnisses, die die Pakete mit 10 bis 15 Treffern enthalten.

Aufgabe 2: Graphiken

(22 Punkte)

- a) Verwenden Sie den Datensatz 'airquality' aus dem datasets-Paket. Speichern Sie den Datensatz in einem Objekt namens 'data'. Verwenden Sie für die nachfolgende Graphik diesen Datensatz.
- b) Entfernen Sie aus 'data' die Beobachtungen, in denen fehlende Werte in mindestens einer der beiden Variablen 'Ozone' oder 'Solar.R' vorkommen.
- c) Erzeugen Sie untenstehende Graphik. Sie benötigen dazu die Variablen 'Ozone', 'Solar.R' und 'Month' aus dem Datensatz.
 - (i) Achten Sie bei den einzelnen Graphiken auf die Skalierung und Beschriftung der Achsen.
 - (ii) Wählen Sie bei den Histogrammen dieselbe Balkenbreite wie in der Vorlage.
 - (iii) Wählen Sie die Farben möglichst ähnlich.
 - (iv) Schreiben Sie in das Feld links oben die Namen aller Gruppenmitglieder.
 - (v) Schreiben Sie in das Streudiagramm links oben die auf drei Nachkommastellen gerundete Korrelation zwischen den beiden Variablen.
 - (vi) Wählen Sie eine möglichst ähnliche Aufteilung für die 9 Felder (Größenverhältnisse).
 - (vii) Berücksichtigen Sie, dass der Abstand zwischen den einzelnen Graphiken nicht zu groß sein sollte.
- d) Speichern Sie die Graphik unter Verwendung von Syntax als pdf-Datei in Ihrem Ergebnisse-Ordner. Verwenden Sie als Dateiname 'Nachname_Vorname_A2'.

Hinweis: Bitte nehmen Sie zur Kenntnis, dass die gewählten Graphiken die Struktur der Daten nicht vollständig berücksichtigen. Die Zeitreihenstruktur der Daten wird nur auf Monatsbasis, aber nicht auf Tagesbasis berücksichtigt. Bei den gewählten Graphiken wird fälschlicherweise angenommen, dass die Messungen an den einzelnen Tagen unabhängig voneinander sind.



Aufgabe 3: ggplot2

(9 Punkte)

- a) Verwenden Sie den modifizierten Datensatz aus Aufgabe 2. Sie benötigen diesmal die Variablen 'Ozone', 'Solar.R', 'Month' und 'Day'. Erzeugen Sie unter Verwendung einer Funktion aus dem Paket ggplot2 eine einzige Graphik mit den folgenden Eigenschaften.
- Es handelt sich bei der Graphik um ein Streudiagramm, welches den Zusammenhang zwischen 'Ozone' und 'Solar.R' darstellt. Dabei befindet sich 'Solar.R' auf der x-Achse und 'Ozone' auf der y-Achse.
 - Für jeden Monat wird ein getrenntes Graphikfeld verwendet. Die 5 Felder werden horizontal in einer Zeile angeordnet.
 - Die Achsen der Graphik und die Graphikfelder sollten sinnvoll beschriftet werden.
 - Die Punkte innerhalb des Streudiagramms sollten so groß sein, dass sie gut sichtbar sind.
 - In jedem der Graphikfelder sollen die Punkte mithilfe eines Farbverlaufs gemäß der Variable 'Day' eingefärbt werden.
 - In jedes der 5 Streudiagramme wird die zugehörige Regressionsgerade eingezeichnet.

- b) Speichern Sie die Graphik unter Verwendung von Syntax als pdf-Datei in Ihrem Ergebnisse-Ordner. Verwenden Sie als Dateiname 'Nachname_Vorname_A3'.

Aufgabe 4: Eine eigene Funktion schreiben

(26 Punkte)

- a) Gegeben sind zwei numerische Vektoren `x` und `y`, sowie eine Faktorvariable `'group'`. Die Variable `'group'` soll genau 3 Faktorstufen haben. Schreiben Sie eine Funktion, die für den Vektor `x` und den Vektor `y` die Korrelation zwischen `x` und `y` auf jeder Faktorstufe der Variable `'group'` berechnet.
- (i) Nennen Sie die Funktion `'cor_by_group'`.
 - (ii) Die Funktion erhält als Parameter `x`, `y`, `group` und evtl. weitere benötigte Parameter.
 - (iii) Prüfen Sie die Eingaben für `x`, `y` und `group` auf Richtigkeit. Die Funktion soll abbrechen, falls mindestens eine der Eingaben nicht vom richtigen Typ ist. Außerdem soll sichergestellt werden, dass `'group'` genau drei Faktorstufen hat. Weiterhin soll die Funktion nur ein Ergebnis liefern, falls alle drei Vektoren gleich lang sind.
 - (iv) Als Ergebnis soll für alle drei Gruppen aus `'group'` die Korrelation zwischen `x` und `y` zurückgegeben werden. Diese soll auf drei Nachkommastellen gerundet sein.
 - (v) Die Funktion soll so flexibel gestaltet sein, dass die Korrelationen nach Pearson, Kendall oder Spearman berechnet werden können.
 - (vi) Die Funktion soll so flexibel gestaltet sein, dass beim Aufruf der Funktion entschieden werden kann, wie mit fehlenden Werten umgegangen wird.
- b) Testen Sie Ihre Funktion, so dass die gesamte Funktionalität Ihrer Funktion überprüft wird.
- Die Funktion sollte für folgende, beispielhafte, Eingaben (einschließlich möglicher Kombinationen) eine sinnvolle Ausgabe liefern:

– **Für `x`:**

```
x1 <- 1:30

x2 <- rnorm(30, mean=0, sd=1)

x3 <- rnorm(n=30, mean=0, sd=1)
x3[c(5)] <- NA

x4 <- rnorm(n=30, mean=0, sd=1)
x4[c(5, 15, 25)] <- NA

x5 <- c(letters[1:15], letters[1:15])

x6 <- data.frame(var1=1:30, var2=30:1)

x7 <- rnorm(29, mean=0, sd=1)
```

– **Für `y`:**

```
y1 <- 1:30

y2 <- rnorm(30, mean=0, sd=1)

y3 <- rnorm(n=30, mean=0, sd=1)
```

```

y3[c(13, 27)] <- NA

y4 <- rnorm(n=30, mean=0, sd=1)
y4[c(5, 16, 27)] <- NA

y5 <- c(letters[1:10], letters[1:10], letters[1:10])

y6 <- data.frame(var1=1:30, var2=30:1)

y7 <- rnorm(25, mean=0, sd=1)

```

– **Für group:**

```

group1 <- factor(c(rep(1,10), rep(2,10), rep(3,10)), levels=1:3)

group2 <- factor(rep((1:3), times=10), levels=1:3, labels=c("a", "b", "c"))

group3 <- c(rep(1,10), rep(2,10), rep(3,10))

group4 <- factor(c(rep(1,15), rep(2,15)), levels=1:2)

group5 <- factor(c(rep(1,9), rep(2,8), rep(3,9)), levels=1:3)

group6 <- factor(sample(1:3, 30, replace=TRUE), levels=1:3)

```

- Berechnen Sie jede Art der Korrelation (Pearson, Kendall, Spearman) mindestens einmal.
 - Falls in Ihren Eingaben fehlende Werte vorkommen, rufen Sie die Funktion bitte mehrmals mit unterschiedlichen Vorgaben in Bezug auf die fehlenden Werte auf.
- c) Schreiben Sie eine neue Funktion `cor_by_group5`, die dieselbe Funktionalität wie `cor_by_group` hat, aber für group 5 Faktorstufen erwartet.
- d) Wenden Sie die Funktion `cor_by_group5` auf den `airquality`-Datensatz (im Original) an. Übergeben Sie für `x` und `y` die Variablen `Solar.R` und `Ozone` und für `group` die Variable `Month`. Wählen Sie gegebenenfalls weitere benötigte Parameter so, dass Sie ein sinnvolles Ergebnis erhalten.