Aufgabe 8

Häufigkeiten in R und Umgang mit fehlenden Werten

Häufigkeitsauszählungen können in R mit `table()` erstellt werden. `cumsum()` bildet kumulierte Summen. `mean()` berechnet das arithmetische Mittel, `median()` den Median

a) Lesen Sie von `http://goo.gl/dZkICg` die Daten der Vorlesungsumfrage ein und bilden Sie jeweils eine Tabelle der absoluten und relativen sowie der absoluten kumulierten und relativen kumulierten Häufigkeiten des Merkmals Alter.

b) Wandeln Sie das Merkmal `MatheZufr` (Antwort auf „Waren Sie zufrieden mit Ihrer Leistung in der Matheklausur) in ein ordinales Merkmal mit sinnvoller Reihenfolge um.

c) Berechnen Sie den Median und das arithmetische Mittel aller metrischen Merkmale der eingelesenen Daten.

Ausprägungen fehlender Werte werden in R mit `NA` (Not Available) dargestellt, Objekte mit fehlenden Werten können mittels `na.omit()` gelöscht werden. Die Funktion `sort()` gibt einen metrisch oder ordinal skalierten Vektor in aufsteigender Reihenfolge zurück.

d) Für ordinale Merkmale ist der Median zwar definiert, mit der eingebauten Funktion in `R` erhält man aber eine Fehlermeldung. Lösen Sie das Problem und berechnen Sie den Median aller (vorhandenen) Ausprägungen des Merkmals `MatheZufr`.

Lösungshinweis:

a)

```
# Umfragedaten einlesen
Umfrage <- read.csv("http://goo.gl/yMeyJp", sep = ";", dec = ",")
```

```
T <- table(Umfrage$Alter)  # absolute Häufigkeiten
n <- length(Umfrage$Alter)  # Anzahl der Objekte
T
```

```
##
## 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
## 24 58 71 53 43 40 25 11 11  8 12  7  2  2  5  2  1  1
## 36
##  1
```

```
cumsum(T)  # kumuliert
```

```
##  18  19  20  21  22  23  24  25  26  27  28  29  30
##  24  82 153 206 249 289 314 325 336 344 356 363 365
##  31  32  33  34  35  36
## 367 372 374 375 376 377
```

```
T/n  # relative Häufigkeiten
```


Prof.Dr. Stefan Etschberger – Hochschule Augsburg – Statistik – Sommersemester 2015 – Aufgabensammlung – (Seite 25 von 121)

```
##
##         18         19         20         21         22
## 0.06366048 0.15384615 0.18832891 0.14058355 0.11405836
##         23         24         25         26         27
## 0.10610080 0.06631300 0.02917772 0.02917772 0.02122016
##         28         29         30         31         32
## 0.03183024 0.01856764 0.00530504 0.00530504 0.01326260
##         33         34         35         36
## 0.00530504 0.00265252 0.00265252 0.00265252
```

```
round(T/n, 3)  # auf drei Kommastellen gerundet
```

```
##
##    18    19    20    21    22    23    24    25    26
## 0.064 0.154 0.188 0.141 0.114 0.106 0.066 0.029 0.029
##    27    28    29    30    31    32    33    34    35
## 0.021 0.032 0.019 0.005 0.005 0.013 0.005 0.003 0.003
##    36
## 0.003
```

```
round(cumsum(T)/n, 3)  # kumulierte rel. Häufigkeiten
```

```
##    18    19    20    21    22    23    24    25    26
## 0.064 0.218 0.406 0.546 0.660 0.767 0.833 0.862 0.891
##    27    28    29    30    31    32    33    34    35
## 0.912 0.944 0.963 0.968 0.973 0.987 0.992 0.995 0.997
##    36
## 1.000
```

```
cbind(as.data.frame(T), as.data.frame(cumsum(T)), as.data.frame(T/n),
    as.data.frame(cumsum(T)/n))
```

```
##    Var1 Freq cumsum(T) Var1       Freq cumsum(T)/n
## 18   18   24        24   18 0.06366048  0.06366048
## 19   19   58        82   19 0.15384615  0.21750663
## 20   20   71       153   20 0.18832891  0.40583554
## 21   21   53       206   21 0.14058355  0.54641910
## 22   22   43       249   22 0.11405836  0.66047745
## 23   23   40       289   23 0.10610080  0.76657825
## 24   24   25       314   24 0.06631300  0.83289125
## 25   25   11       325   25 0.02917772  0.86206897
## 26   26   11       336   26 0.02917772  0.89124668
## 27   27    8       344   27 0.02122016  0.91246684
## 28   28   12       356   28 0.03183024  0.94429708
## 29   29    7       363   29 0.01856764  0.96286472
## 30   30    2       365   30 0.00530504  0.96816976
## 31   31    2       367   31 0.00530504  0.97347480
## 32   32    5       372   32 0.01326260  0.98673740
## 33   33    2       374   33 0.00530504  0.99204244
## 34   34    1       375   34 0.00265252  0.99469496
## 35   35    1       376   35 0.00265252  0.99734748
## 36   36    1       377   36 0.00265252  1.00000000
```

b)
```
Umfrage$MatheZufr <- ordered(Umfrage$MatheZufr, levels = c("nicht",
    "geht so", "zufrieden", "sehr"))
```


Prof.Dr. Stefan Etschberger – Hochschule Augsburg – Statistik – Sommersemester 2015 – Aufgabensammlung – (Seite 26 von 121)

```r
Mathe.sortiert <- sort(Umfrage$MatheZufr)
n <- length(Mathe.sortiert)
is.integer(n/2)  # FALSE, also n ungerade
```

```
## [1] FALSE
```

```r
Mathe.Median <- Mathe.sortiert[(n + 1)/2]
Mathe.Median
```

```
## [1] geht so
## Levels: nicht < geht so < zufrieden < sehr
```

Aufgaben zur deskriptiven Statistik

Aufgabe 9

Deskriptiv: Häufigkeit

Ein Einzelhändler registriert für einen Exklusivartikel im Verlauf von 30 Verkaufstagen folgende Verkaufszahlen:

Tag	1	2	3	4	5	6	7	8	9	10
Anzahl	5	2	3	0	0	1	3	6	0	2
Tag	11	12	13	14	15	16	17	18	19	20
Anzahl	1	0	1	0	2	3	5	1	0	0
Tag	21	22	23	24	25	26	27	28	29	30
Anzahl	3	5	3	1	0	0	0	6	3	1

a) Berechnen Sie die absoluten und relativen Häufigkeiten der Ausprägungen sowie die absolute kumulierte Häufigkeit für $x = 4$.
b) Erstellen Sie das zugehörige Balkendiagramm und das Kreissektorendiagramm mithilfe der absoluten Häufigkeiten.

Lösungshinweis:

```
x <- c(5, 2, 3, 0, 0, 1, 3, 6, 0, 2, 1, 0, 1, 0, 2, 3, 5,
    1, 0, 0, 3, 5, 3, 1, 0, 0, 0, 6, 3, 1)
T <- table(x)
T  # Ausgabe der Häufigkeiten
cumsum(T)  # Ausgabe der kumulierten Häufigkeiten
```

Ausprägung	0	1	2	3	5	6
Häufigkeit	10	6	3	6	3	2
kumuliert	10	16	19	25	28	30

```
plot(table(x))
pie(table(x))
```

Zusätzliche Aufgaben

Aufgabe 99

Z: Einfache Grafiken in R

Grafiken

Grafiken kann man in R mit `plot()` erzeugen. `plot` versucht anhand der übergebenen Merkmalstypen aus dem Kontext zu entscheiden, welche Art von Grafik erzeugt werden soll:

a) Probieren Sie folgende plots in R aus:

```
# Umfragedaten einlesen
Daten <- read.csv("http://goo.gl/yMeyJp", sep = ";", dec = ",")
```

```
# plot mit einem nominalen Merkmal als Argument zeigt ein
# Balkendiagramm
plot(Daten$Farbe)
# Farben können als optionales Argument angegeben werden Ausrichtung
# der Beschriftung über las
plot(Daten$Farbe, col = c("blue", "yellow", "red", "black", "grey", "white"),
    las = 2)
# zwei nominale Merkmale werden als 'spineplot' gegenübergestellt
plot(Daten$Farbe, Daten$Geschlecht)
# ein metrisches und ein nominales Merkmal wird als Liste von Boxplots
# gezeichnet
plot(Daten$Geschlecht, Daten$Alter)
# zwei metrische Merkmale verarbeitet plot() als Streuplot
plot(Daten$AlterV, Daten$AlterM)
# mehr als zwei metrische Merkmale führen zu einer Streuplotmatrix:
plot(Daten[, c("Alter", "AlterM", "AlterV")])
```

`plot()` versteht je nach Kontext viele Parameter zur Anpassung der Grafiken. Die Liste aller Grafikeinstellungen von `plot()` und deren möglichen Werte findet man unter `?par`. Einige wichtige Parameter sind:

- *`main`: Überschrift der Grafik*
- *`xlab, ylab`: Beschriftung der Abszisse bzw. Ordinate*
- *`pch, col, cex`: ein Wert oder ein Vektor, der das Symbol, das gezeichnet wird, beinflusst; dabei steht `pch` für die Form des Symbols, `col` für die Farbe und `cex` für die Größe des Symbols.*

Mit der Funktion `rgb()` kann man beliebige Farben mischen und deren Transparenz festlegen.

b) Probieren Sie folgende Ausdrücke in R aus und experimentieren Sie mit Änderungen der Parameter

Prof.Dr. Stefan Etschberger – Hochschule Augsburg – Statistik – Sommersemester 2015 – Aufgabensammlung – (Seite 119 von 121)

```r
# Speichere zunächst 2 Farben
# rgb(): "r"ot, "g"rün, "b"lau;
# Argumente 1  bis 3: Anteile an rot, grün, blau
# 4. Argument: Transparenz (1 ist undurchsichtig, 0 ist komplett transparent)
Farben = c(rgb(1, 0, 0, 0.2), rgb(0, 0, 1, 0.2))
Symbole = c(20,18) # Kreis und Raute

plot(Daten$AnzSchuhe, Daten$AusgSchuhe,
     col=Farben[Daten$Geschlecht],  # farbliche Markierung über Geschlecht
     pch=Symbole[Daten$Geschlecht], # unterschiedliche Symbole je nach Geschlecht
     xlab="Wieviele Paar Schuhe besitzen Sie?", # Beschriftung der Abszisse
     ylab="Ausgaben in den letzten 12 Monaten für Schuhe") # Ordinate

plot(Daten[,c("Alter", "AlterV", "AlterM")],
     col=Farben[Daten$Geschlecht],  # farbliche Markierung über Geschlecht
     pch=Symbole[Daten$Geschlecht]) # unterschiedliche Symbole je nach Geschlecht
```

```r
# Alternative zu spineplot: Mosaikplot
mosaicplot(Daten$Farbe ~ Daten$Geschlecht,
           shade=TRUE, # farbige Markierung
           las=2, # Ausrichtung der Achsenbeschriftung
           xlab="Geschlecht",
           ylab="Wunschfarbe", main="")
```

```r
# Boxplot, horizontale Ausrichtung, eingefärbt
plot(Daten$Geschlecht, Daten$Alter, col = rgb(0, 0, 1, 0.2), horizontal = TRUE,
    las = 2, lwd = 1.5, pch = 20)
```

c) Generieren Sie einen Streuplot, in dem die Ausgaben für Schuhe und die Anzahl der Schuhe gegenübergestellt werden. Trennen Sie die Farben nach dem Merkmal „Farbe“ und färben Sie mit der jeweils in der Umfrage genannten Lieblingsfarbe ein, jeweils mit 30% Transparenz. Zeichnen Sie für die Frauen ausgefüllte Dreiecke und für die Männer ausgefüllte Quadrate.

Lösungshinweis:

a) Folgende Grafiken ergeben sich:

b) Resultate:

c)
```r
levels(Daten$Farbe)
```

```
## [1] "blau"    "gelb"    "rot"     "schwarz" "silber"  "weiss"
```

```r
Farben=c(rgb(0,0,1,0.3), # blau
         rgb(.9,.9,0,0.3), # gelb
         rgb(1,0,0,0.3), # rot
         rgb(0,0,0,0.3), # schwarz
         rgb(0.5,0.5,0.5,0.3), # grau (silber),
         rgb(0.2,0.2,0.2,0.3)) # (weiss) hier schwierig, also: helleres grau
Symbole=c(17,15)
plot(Daten$AnzSchuhe, Daten$AusgSchuhe, col=Farben[Daten$Farbe],
     pch=Symbole[Daten$Geschlecht])
grid()
```


Prof.Dr. Stefan Etschberger – Hochschule Augsburg – Statistik – Sommersemester 2015 – Aufgabensammlung – (Seite 120 von 121)

Aufgabe 100

Empirische Verteilungsfunktion, Quantile, Boxplot

mit der Funktion `ecdf()` wird eine empirische Verteilungsfunktion erstellt. Wird das Ergebnis von `ecdf()` in die Funktion `plot()` eingesetzt erhält man eine grafische Darstellung dieser Verteilungsfunktion. Die Funktion `quantile()` liefert empirische Quantile einer Urliste. Dabei gibt es verschiedene Varianten, wie diese Quantile berechnet werden können; mit dem Parameter `type` kann man diese unterschiedlichen Definitionen ansprechen. In der Vorlesung wurde `type = 2` definiert.

Gegeben ist die folgende Urliste:

```
x <- c(1, 2, 1, 2, 10, 10, 20, 1, 2, 1)
```

Lösen Sie folgende Aufgaben jeweils zuerst auf Papier und dann mit R:

a) Geben Sie die empirische Verteilungsfunktion $F(x)$ an und zeichnen Sie den Graph von F.
b) Berechnen Sie die folgenden empirischen Quantile $\tilde{x}_{p_i}$ mit

```
p <- c(0.2, 0.25, 0.4, 0.5, 0.6, 0.75, 0.99)
```

Lösungshinweis:

a) Sortierte Urliste und relative kumulierte Häufigkeiten

```
sort(x)
```

```
##  [1]  1  1  1  1  2  2  2 10 10 20
```

```
cumsum(table(x))/length(x)
```

```
##   1   2  10  20
## 0.4 0.7 0.9 1.0
```

damit ist

$$F(x) = \begin{cases} 0 & \text{für} \quad x < 1 \\ 0,4 & \text{für} \quad 1 \leqq x < 2 \\ 0,7 & \text{für} \quad 2 \leqq x < 10 \\ 0,9 & \text{für } 10 \leqq x < 20 \\ 1,0 & \text{für } 20 \leqq x \end{cases}$$

Graph von $F(x)$:

```
plot(ecdf(x), col = rgb(1, 0, 0, 0.7), lwd = 2, main = "")
```

b) Empirische Quantile und Boxplot

```
quantile(x, probs = p, type = 2)
```

```
##  20%  25%  40%  50%  60%  75%  99%
##  1.0  1.0  1.5  2.0  2.0 10.0 20.0
```

```
boxplot(x, col = "aliceblue", horizontal = TRUE, lwd = 1.5)
```


Prof.Dr. Stefan Etschberger – Hochschule Augsburg – Statistik – Sommersemester 2015 – Aufgabensammlung – (Seite 121 von 121)