

W-Rechnung und Statistik für Ingenieure

Übung 2

Einladen von Datensätzen in R

Es gibt verschiedene Möglichkeiten verschiedene Arten von Daten in R zu laden

- `read.table` liest Daten im Tabellenformat ein und erzeugt Datenmenge in R deren Struktur sich nach der verwendeten Tabelle richtet

Einladen von Datensätzen in R

Es gibt verschiedene Möglichkeiten verschiedene Arten von Daten in R zu laden

- `read.table` liest Daten im Tabellenformat ein und erzeugt Datenmenge in R deren Struktur sich nach der verwendeten Tabelle richtet
- `read.csv`, `read.csv2` liest Daten mit csv-Format ein (csv: 'comma separated values')

Einladen von Datensätzen in R

Es gibt verschiedene Möglichkeiten verschiedene Arten von Daten in R zu laden

- `read.table` liest Daten im Tabellenformat ein und erzeugt Datenmenge in R deren Struktur sich nach der verwendeten Tabelle richtet
- `read.csv`, `read.csv2` liest Daten mit csv-Format ein (csv: 'comma separated values')
- `scan` liest Daten aus der R-Konsole oder einer Datei als Vektor oder Liste ein

Einladen von Datensätzen in R

Es gibt verschiedene Möglichkeiten verschiedene Arten von Daten in R zu laden

- `read.table` liest Daten im Tabellenformat ein und erzeugt Datenmenge in R deren Struktur sich nach der verwendeten Tabelle richtet
- `read.csv`, `read.csv2` liest Daten mit csv-Format ein (csv: 'comma separated values')
- `scan` liest Daten aus der R-Konsole oder einer Datei als Vektor oder Liste ein
- `read.xls` liest bei geladenem Paket `xlsReadWrite` MS-EXCEL Dateien ein

Einladen von Datensätzen in R

Es gibt verschiedene Möglichkeiten verschiedene Arten von Daten in R zu laden

- `read.table` liest Daten im Tabellenformat ein und erzeugt Datenmenge in R deren Struktur sich nach der verwendeten Tabelle richtet
- `read.csv`, `read.csv2` liest Daten mit csv-Format ein (csv: 'comma seperated values')
- `scan` liest Daten aus der R-Konsole oder einer Datei als Vektor oder Liste ein
- `read.xls` liest bei geladenem Paket `xlsReadWrite` MS-EXCEL Dateien ein
- Mit passenden Funktionen können Daten auch aus der Zwischenablage geladen werden, z.B.:
`read.table('Clipboard')`

Einladen von Datensätzen in R

Es gibt verschiedene Möglichkeiten verschiedene Arten von Daten in R zu laden

- `read.table` liest Daten im Tabellenformat ein und erzeugt Datenmenge in R deren Struktur sich nach der verwendeten Tabelle richtet
- `read.csv`, `read.csv2` liest Daten mit csv-Format ein (csv: 'comma separated values')
- `scan` liest Daten aus der R-Konsole oder einer Datei als Vektor oder Liste ein
- `read.xls` liest bei geladenem Paket `xlsReadWrite` MS-EXCEL Dateien ein
- Mit passenden Funktionen können Daten auch aus der Zwischenablage geladen werden, z.B.:
`read.table('Clipboard')`
- ...

*Se
setzt
getrun*

Aufgabe 1

Erstellen einer Häufigkeitstabelle und von Kreisdiagrammen von Daten im csv-Format.

- Der Datensatz `Druckfestigkeit.csv` kann z.B. eingeladen werden mit `read.csv2('Druckfestigkeit.csv')`

Aufgabe 1

Erstellen einer Häufigkeitstabelle und von Kreisdiagrammen von Daten im csv-Format.

- Der Datensatz `Druckfestigkeit.csv` kann z.B. eingeladen werden mit `read.csv2('Druckfestigkeit.csv')`
- Dazu ist das R Arbeitsverzeichnis so zu wählen, dass die Datei in diesem liegt oder ein kompletter Dateipfad anzugeben
ACHTUNG: R benötigt umgekehrte 'Slashes'

Aufgabe 1

Erstellen einer Häufigkeitstabelle und von Kreisdiagrammen von Daten im csv-Format.

- Der Datensatz `Druckfestigkeit.csv` kann z.B. eingeladen werden mit `read.csv2("Druckfestigkeit.csv")`
- Dazu ist das R Arbeitsverzeichnis so zu wählen, dass die Datei in diesem liegt oder ein kompletter Dateipfad anzugeben
ACHTUNG: R benötigt umgekehrte 'Slashes'
- Auswahl des Arbeitsverzeichnisses mit `setwd("C:/Users/produnis/R")`

Aufgabe 1

Erstellen einer Häufigkeitstabelle und von Kreisdiagrammen von Daten im csv-Format.

- Der Datensatz `Druckfestigkeit.csv` kann z.B. eingeladen werden mit `read.csv2("Druckfestigkeit.csv")`
- Dazu ist das R Arbeitsverzeichnis so zu wählen, dass die Datei in diesem liegt oder ein kompletter Dateipfad anzugeben
ACHTUNG: R benötigt umgekehrte 'Slashes'
- Auswahl des Arbeitsverzeichnisses mit `setwd("C:/Users/produnis/R")`
- Erstelle eine Häufigkeitstabelle für die Schleiforte mithilfe von `table()`

Aufgabe 1

Erstellen einer Häufigkeitstabelle und von Kreisdiagrammen von Daten im csv-Format.

- Der Datensatz `Druckfestigkeit.csv` kann z.B. eingeladen werden mit `read.csv2("Druckfestigkeit.csv")`
- Dazu ist das R Arbeitsverzeichnis so zu wählen, dass die Datei in diesem liegt oder ein kompletter Dateipfad anzugeben
ACHTUNG: R benötigt umgekehrte 'Slashes'
- Auswahl des Arbeitsverzeichnisses mit `setwd("C:/Users/produnis/R")`
- Erstelle eine Häufigkeitstabelle für die Schleiferte mithilfe von `table()`
- Erstelle ein Kreis-(Torten-)Diagramm mithilfe von `pie()`

Aufgabe 1 : Datensatz

```
>druck<-read.csv2("Druckfestigkeit.csv")
```

```
> druck
```

	Druck	Festbetonrohddichte	H	S	P
1	168.500	2.495	Aachen	Aachen	Kassel
2	167.100	2.516	Aachen	Aachen	Kassel
3	158.700	2.485	Aachen	Aachen	Kassel
4	150.500	2.485	Aachen	Kassel	Kassel
5	151.800	2.487	Aachen	Kassel	Kassel
6	174.400	2.507	Aachen	Kassel	Kassel
7	165.600	2.510	Aachen	Aachen	Aachen
8	139.800	2.500	Aachen	Aachen	Aachen
9	156.900	2.520	Aachen	Aachen	Aachen
...					
122	147.700	2.525	Kassel	Kassel	Kassel
123	145.300	2.495	Kassel	Kassel	Kassel
124	144.700	2.525	Kassel	Kassel	Kassel
125	142.100	2.502	Kassel	Kassel	Kassel

Aufgabe 1 : Datensatz der Schleiforte

Auf Grundlage der Datenstruktur lässt sich aus den eingelesenen Daten die Menge der Schleif- und Prüforte extrahieren.

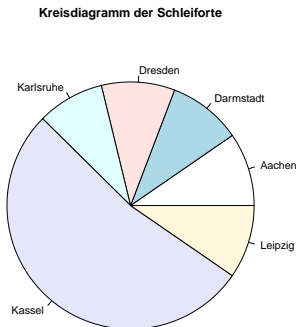
```
> druck$$
[1] Aachen      Aachen      Aachen      Kassel      Kassel      Kassel      Aachen
[8] Aachen      Aachen      Aachen      Aachen      Aachen      Kassel      Kassel
[15] Kassel      Aachen      Aachen      Aachen      Kassel      Kassel      Kassel
[22] Kassel      Kassel      Kassel      Kassel      Kassel      Kassel      Darmstadt
[29] Darmstadt   Darmstadt   Kassel      Kassel      Kassel      Darmstadt   Darmstadt
[36] Darmstadt   Darmstadt   Darmstadt   Darmstadt   Kassel      Kassel      Kassel
[43] Darmstadt   Darmstadt   Darmstadt   Kassel      Kassel      Kassel      Dresden
[50] Dresden     Dresden     Kassel      Kassel      Kassel      Dresden     Dresden
[57] Dresden     Dresden     Dresden     Dresden     Kassel      Kassel      Kassel
[64] Dresden     Dresden     Dresden     Kassel      Kassel      Kassel      Kassel
[71] Kassel      Kassel      Kassel      Kassel      Kassel      Karlsruhe   Karlsruhe
[78] Karlsruhe   Kassel      Kassel      Kassel      Karlsruhe   Karlsruhe   Karlsruhe
[85] Karlsruhe   Karlsruhe   Kassel      Kassel      Kassel      Karlsruhe   Karlsruhe
[92] Karlsruhe   Kassel      Kassel      Kassel      Leipzig     Leipzig     Leipzig
[99] Kassel      Kassel      Kassel      Leipzig     Leipzig     Leipzig     Leipzig
[106] Leipzig     Leipzig     Kassel      Kassel      Kassel      Leipzig     Leipzig
[113] Leipzig     Kassel      Kassel      Kassel      Kassel      Kassel      Kassel
[120] Kassel      Kassel      Kassel      Kassel      Kassel      Kassel      Kassel
```

Aufgabe 1 : Auswertung der Schleiforte

```
>table(druck$S)
```

Aachen	Darmstadt	Dresden	Karlsruhe	Kassel	Leipzig
12	12	12	11	66	12

```
>pie(table(druck$S), main='\"Kreisdiagramm der  
Schleiforte\"')
```

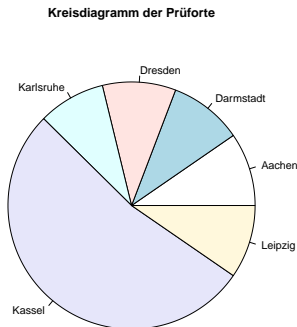


Aufgabe 1 : Auswertung der Prüforte

```
>table(druck$P)
```

Aachen	Darmstadt	Dresden	Karlsruhe	Kassel	Leipzig
12	12	12	11	66	12

```
>pie(table(druck$P), main='\"Kreisdiagramm der  
Prüforte\"')
```



Darstellung des Datensatzes STEEL.DAT.

- Der Datensatz STEEL.DAT kann z.B. eingeladen werden mit
`>steel<-scan("STEEL.DAT")`

Darstellung des Datensatzes STEEL.DAT.

- Der Datensatz STEEL.DAT kann z.B. eingeladen werden mit
`>steel<-scan("STEEL.DAT")`
- Messergebnisse der beiden Linien in einem Vektor, 1-10: Linie 1 und 11-20: Linie 2

Darstellung des Datensatzes STEEL.DAT.

- Der Datensatz STEEL.DAT kann z.B. eingeladen werden mit
`>steel<-scan("STEEL.DAT")`
- Messergebnisse der beiden Linien in einem Vektor, 1-10: Linie 1 und 11-20: Linie 2
- Teile den Vektor auf z.B. durch
`>linie1<-steel[1:10], >linie2<-steel[11:20]`

Darstellung des Datensatzes STEEL.DAT.

- Der Datensatz STEEL.DAT kann z.B. eingeladen werden mit
`>steel<-scan("STEEL.DAT")`
- Messergebnisse der beiden Linien in einem Vektor, 1-10: Linie 1 und 11-20: Linie 2
- Teile den Vektor auf z.B. durch
`>linie1<-steel[1:10], >linie2<-steel[11:20]`
- Erstelle ein Histogramm mithilfe von `hist()`

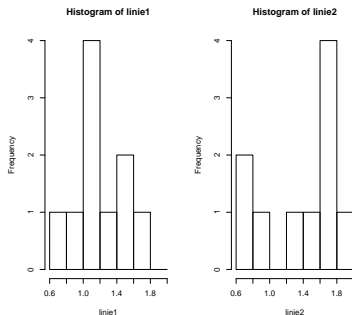
Aufgabe 2: Histogramme

Um zwei Diagramme in einer Grafik zu erhalten kann `par(mfrow=c(1,2))` benutzt werden (2 Grafiken in einer Zeile, `par(mfrow=c(2,1))` liefert 2 Grafiken untereinander)

```
>par(mfrow=c(1,2))
```

```
>hist(linie1,breaks=seq(0.6,2,0.2),include.lowest=T)
```

```
>hist(linie2,breaks=seq(0.6,2,0.2),include.lowest=T)
```



—, hist
plotter

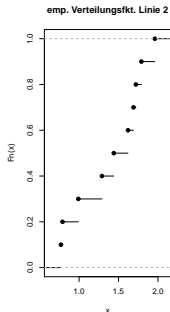
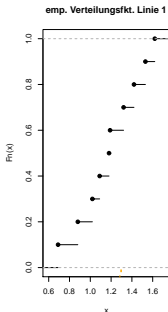
Aufgabe 2: Empirische Verteilungsfunktion

Die Funktion `ecdf('Daten')` erstellt die empirische Verteilungsfunktion bezüglich der gegebenen Daten. Diese kann dann wie üblich geplottet werden.

```
>plot(ecdf(linie1),main='emp. Verteilungsfkt. Linie 1')
```

```
>plot(ecdf(linie2),main='emp. Verteilungsfkt. Linie 2')
```

$$F(x) = P(X \leq x)$$



Aufgabe 2: Berechnen der Häufigkeiten

Bestimmen der absoluten, relativen und Summenhäufigkeiten mit R (ersetze ... durch die Argumente von oben). Zur Bestimmung werden die Daten der Histogramme gespeichert und verwendet.

```
>hist1<-hist(linie1,...,plot=F)
```

```
>hist2<-hist(linie2,...,plot=F)
```

- absolute Häufigkeiten `>hist1$counts`

Aufgabe 2: Berechnen der Häufigkeiten

Bestimmen der absoluten, relativen und Summenhäufigkeiten mit R (ersetze ... durch die Argumente von oben). Zur Bestimmung werden die Daten der Histogramme gespeichert und verwendet.

```
>hist1<-hist(linie1,...,plot=F)
```

```
>hist2<-hist(linie2,...,plot=F)
```

- absolute Häufigkeiten `>hist1$counts`
- relative Häufigkeiten `>hist1$counts/length(linie1)`

Aufgabe 2: Berechnen der Häufigkeiten

Bestimmen der absoluten, relativen und Summenhäufigkeiten mit R (ersetze ... durch die Argumente von oben). Zur Bestimmung werden die Daten der Histogramme gespeichert und verwendet.

```
>hist1<-hist(linie1,...,plot=F)
```

```
>hist2<-hist(linie2,...,plot=F)
```

- absolute Häufigkeiten `>hist1$counts`
- relative Häufigkeiten `>hist1$counts/length(linie1)`
- Summenhäufigkeiten
`>cumsum(hist1$counts/length(linie1))`

Aufgabe 2: Berechnen der Häufigkeiten

absolute Häufigkeiten

	[0.6, 0.8)	[0.8, 1)	[1, 1.2)	[1.2, 1.4)	[1.4, 1.6)	[1.6, 1.8)	[1.8, 2]
L1	1	1	4	1	2	1	0
L2	2	1	0	1	1	4	1

relative Häufigkeiten

	[0.6, 0.8)	[0.8, 1)	[1, 1.2)	[1.2, 1.4)	[1.4, 1.6)	[1.6, 1.8)	[1.8, 2]
L1	0.1	0.1	0.4	0.1	0.2	0.1	0.0
L2	0.2	0.1	0.0	0.1	0.1	0.4	0.1

Summenhäufigkeiten

	[0.6, 0.8)	[0.8, 1)	[1, 1.2)	[1.2, 1.4)	[1.4, 1.6)	[1.6, 1.8)	[1.8, 2]
L1	0.1	0.2	0.6	0.7	0.9	1.0	1.0
L2	0.2	0.3	0.3	0.4	0.5	0.9	1.0

Aufgabe 3: Histogramm zu Lebenszeiten von Glühbirnen

Datensatz kann mit `read.table` eingelesen werden. In der Tabelle sieht man direkt Klassen und absolute Häufigkeiten. Daraus kann ein Histogramm hergeleitet werden.

950-1000	1000-1050	1050-1100	1100-1150	1150-1200	1200-1250
2	2	3	6	7	12

1250-1300	1300-1350	1350-1400	1400-1450	1450-1500	1500-1550
16	20	24	27	29	29

1550-1600	1600-1650	1650-1700	1700-1750	1750-1800	1800-1850
28	25	21	16	12	8

1850-1900	1900-1950	1950-2000	2000-2500	2500-2100
6	3	2	1	1

#

-

950

1550

Aufgabe 3: Histogramm zu Lebenszeiten von Glühbirnen

Trägt man die Klassen und Häufigkeiten ab, erhält man die graphische Darstellung des Histogramms.

