Genetische Statistik

WS 2021/2022 R - Übung 1 - Grundlagen

Dr. Janne Pott (janne.pott@uni-leipzig.de)

November 02, 2021

Aufgabe 1: R als Taschenrechner

Berechnen Sie folgende Terme:

- $|3^5 - 2^{10}|$
- $sin(\frac{3}{4}\pi)$
- $\frac{16!}{5!11!}$
- $\sqrt{37-8} + \sqrt{11}$
- $e^{-2.7}/0.1$
- $2.3^8 + \ln(7.4) - \tan(0.3\pi)$
- $\log_{10}(27)$
- $\ln(\pi)$
- $\ln(-1)$

Aufgabe 1: Lösung

```
abs(3^5 - 2^10)
sin((3/4)*pi)
factorial(16)/(factorial(5)*factorial(11))
sqrt(37-8) + sqrt(11)
exp(-2.7)/0.1
2.3^8 + log(7.4) - tan(0.3*pi)
log10(27)
log(pi)
log(-1)
```

Aufgabe 2: Variablen und Folgen

Erzeugen Sie für $n = 1, ..., 10$:

- $a_n = 3^n$
- $b_n = e^{-n}$
- $c_n = (1 + \frac{1}{n})^n$
- $d_n = \sin(n\frac{\pi}{10})$

Aufgabe 2: Lösung

```
n<-seq(1:10)
a<-3^n
a
b<-exp(-n)
b
c<-(1 + 1/n)^n
c
d<-sin(n*pi/10)
d
```

Aufgabe 3: Funktionen

- $h(x) = \sin(\sqrt{x})$ an 0, 0.1, 0.2, ..., 0.9, und 1.
- $g_1(a, b, c) = \frac{a*b}{a*b+(1-c)*(1-a)}$ und $g_2(a, b, c) = \frac{c*(1-a)}{c*(1-a)+(1-b)*a}$ für $a \in [0, 1]$, $b = 0.7$ und $c = 0.95$
- Plot von g_1 und g_2 für $a \in [0, 1]$, $b = 0.7$ und $c = 0.95$.

Aufgabe 3: Lösung (1)

```
h<-function(x){sin(sqrt(x))}
x<-seq(0,1,0.1)
options(width = 60)
h(x)
```

```
##  [1] 0.0000000 0.3109836 0.4324548 0.5207443 0.5911271
##  [6] 0.6496369 0.6994279 0.7424097 0.7798507 0.8126489
## [11] 0.8414710
```

Aufgabe 3: Lösung (2)

```
g1<-function(a,b,c){return(b*a/(b*a+(1-c)*(1-a)))}
g2<-function(a,b,c) {return(c*(1-a)/(c*(1-a)+(1-b)*a))}
g1(x,0.7,0.95)
```

```
##  [1] 0.0000000 0.6086957 0.7777778 0.8571429 0.9032258
##  [6] 0.9333333 0.9545455 0.9702970 0.9824561 0.9921260
## [11] 1.0000000
```

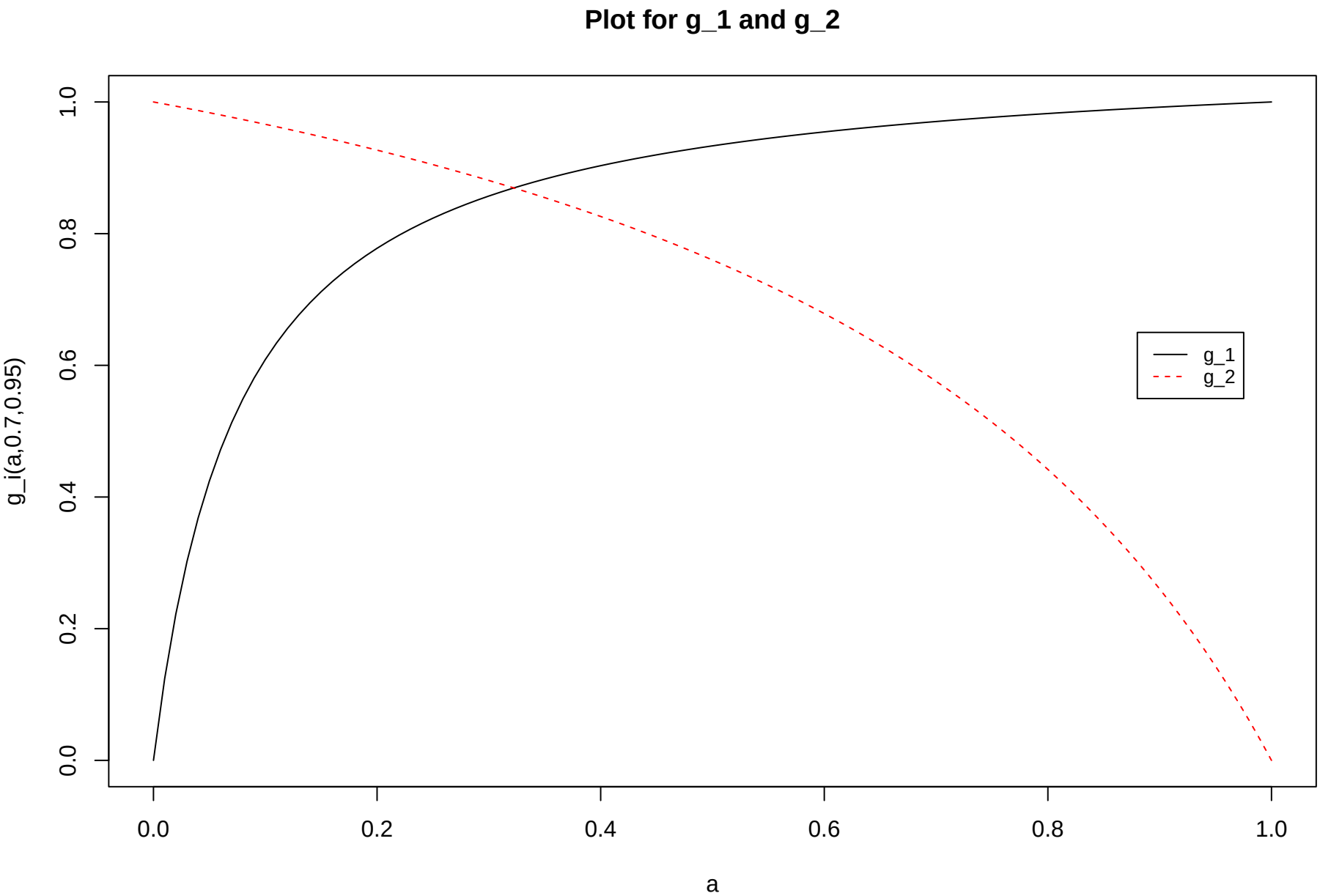
```
g2(x,0.7,0.95)
```

```
##  [1] 1.0000000 0.9661017 0.9268293 0.8807947 0.8260870
##  [6] 0.7600000 0.6785714 0.5757576 0.4418605 0.2602740
## [11] 0.0000000
```

Aufgabe 3: Lösung (3)

```
curve(g1(x,0.7,0.95),0,1,
      main = "Plot for g_1 and g_2",
      xlab = "a",
      ylab = "g_i(a,0.7,0.95)")
curve(g2(x,0.7,0.95),add=TRUE,col="red",lty="dashed")
legend(0.88, 0.65, legend=c("g_1", "g_2"),
       col=c("black", "red"), lty=1:2, cex=0.8)
```

Aufgabe 3: Lösung (3)



Aufgabe 4: Vektoren & Matrizen

- Vektor A mit den Quadratzahlen 1, 4, 9, . . . , 400
- Vektoren B und C aus den ersten bzw. letzten zehn Einträgen von A.
- Vektor D mit 50 Einträgen mit Muster ACCB
- Erzeugen Sie aus D die 10x5 Matrix M.

Aufgabe 4: Lösung

```
options(width = 50)
n<-c(1:20)
n
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
## [16] 16 17 18 19 20
```

```
A<-n^2
A
```

```
##  [1]   1   4   9  16  25  36  49  64  81 100 121
## [12] 144 169 196 225 256 289 324 361 400
```

```
B<-A[1:10]
C<-A[11:20]
D<-c(A,C,C,B)

M<-matrix(D,nrow = 10)
```

Aufgabe 5: Schleifen

- Erstellen Sie einen Vektor **iters** für Anzahl der Iterationen, beginnend bei 10, endend bei 100, und in 10er Schritten.
- Erstellen Sie einen Outputvektor **times**, in dem die Zeit eingetragen werden soll.
- Definieren Sie die erste *for*-Schleife von 1 bis zur Länge von **iters**, die
  - sich die Anzahl der gewünschten Iterationen aus **iters** zieht
  - die Zeitmessung startet (x=Sys.time())
  - pro Iteration eine normalverteilte Zufallsvariable mit n=10000 Ziehungen erstellt (dummy=rnorm(1e5), zweite Schleife) und die Summary davon bestimmt (dummy2<-summary(dummy), entspricht Min., Max., Quantile)
  - die Zeit in der Variablen **times** abspeichert
- Plotten Sie **iters** gegen **times**!

Aufgabe 5: Lösung (1)

```
#iterations to time
iters<-seq(10,100,by=10)

#output time vector for  iteration sets
times<-numeric(length(iters))
```

Aufgabe 5: Lösung (2)

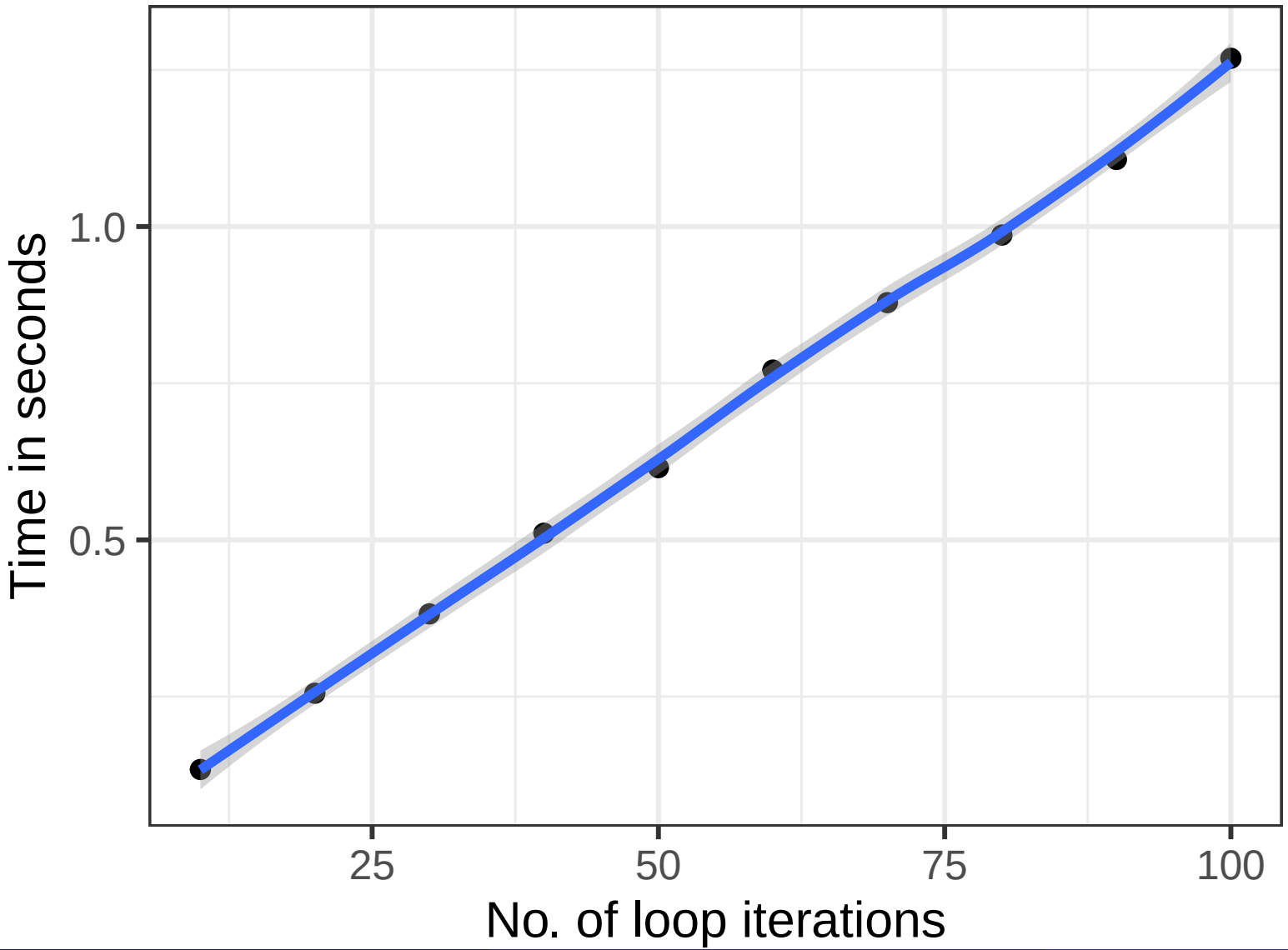
```
#loop over iteration sets
for(val in 1:length(iters)){
  cat(val,' of ', length(iters),'\n')
  to.iter<-iters[val]

  #start time
  strt<-Sys.time()

  #same for loop as before
  for(i in 1:to.iter){
    to.ls<-rnorm(1e5)
    to.ls<-summary(to.ls)
  }

  #end time
  times[val]<-Sys.time()-strt
}
```

Aufgabe 5: Lösung (3)

```
#plot the times
to.plot<-data.frame(iters,times)
ggplot2::ggplot(to.plot,aes(x=iters,y=times)) +
    geom_point() +
    geom_smooth() +
    theme_bw() +
    scale_x_continuous('No. of loop iterations') +
    scale_y_continuous ('Time in seconds')
```

Aufgabe 5: Lösung (4)

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Aufgabe 6: Dateneingabe

- Laden Sie den Datensatz *iris*.
- Ändern Sie die Klasse von *data.frame* zu *data.table*.
- Wie viele Einträge sind pro Spezies vorhanden?
- Wie lang und breit sind im Mittel die Blätter pro Spezie? Nutzen Sie dazu die Funktion *lapply()*.
- Definieren Sie eine neue Spalte als Produkt der Kelchblattlänge und -breite.
- Wie groß ist die mittlere Differenz der Blattlänge (Kelch - Blüte) in der Spezies *setosa*?

Aufgabe 6: Lösung (1)

```
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length
## 1          5.1         3.5          1.4
## 2          4.9         3.0          1.4
## 3          4.7         3.2          1.3
## 4          4.6         3.1          1.5
## 5          5.0         3.6          1.4
## 6          5.4         3.9          1.7
##   Petal.Width Species
## 1         0.2  setosa
## 2         0.2  setosa
## 3         0.2  setosa
## 4         0.2  setosa
## 5         0.2  setosa
## 6         0.4  setosa
```

Aufgabe 6: Lösung (2)

```
getDTthreads()
```

```
## [1] 4
```

```
setDTthreads(1)

setDT(iris)

iris[,.N,Species]
```

```
##       Species  N
## 1:     setosa 50
## 2: versicolor 50
## 3:  virginica 50
```

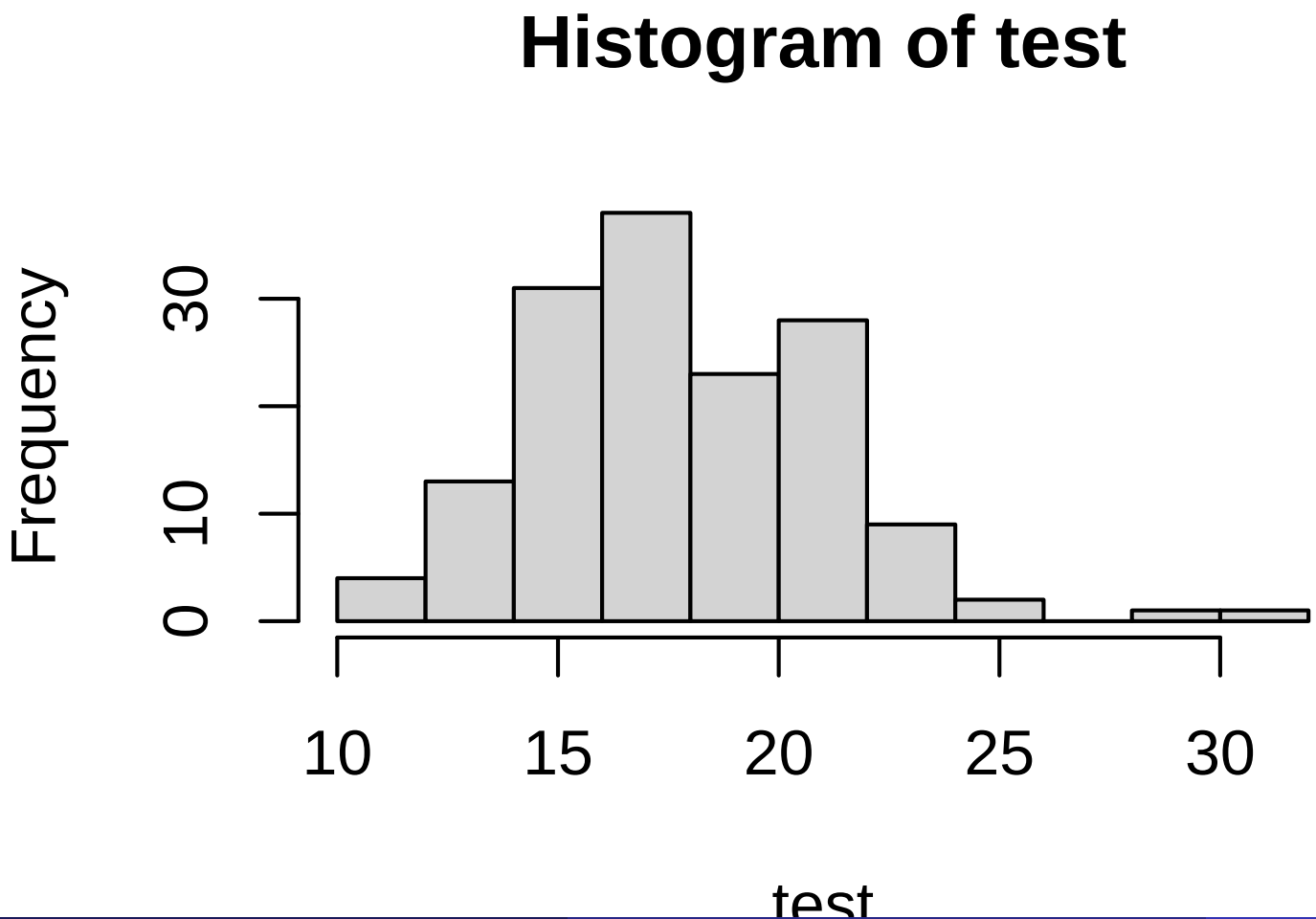
Aufgabe 6: Lösung (3)

```
iris[,lapply(.SD,mean),Species]
```

```
##       Species Sepal.Length Sepal.Width
## 1:     setosa        5.006       3.428
## 2: versicolor        5.936       2.770
## 3:  virginica        6.588       2.974
##    Petal.Length Petal.Width
## 1:        1.462       0.246
## 2:        4.260       1.326
## 3:        5.552       2.026
```

Aufgabe 6: Lösung (4)

```
iris[,test := Sepal.Length*Sepal.Width]
iris[,hist(test)]
```



Aufgabe 6: Lösung (5)

```
iris[Species=="setosa",mean(Sepal.Length - Petal.Length)]
```

```
## [1] 3.544
```

```
iris[,mean(Sepal.Length - Petal.Length),Species]
```

```
##       Species    V1
## 1:     setosa 3.544
## 2: versicolor 1.676
## 3:  virginica 1.036
```