Statistische Datenanalyse mit R, online, Lösungen

Dr. Andrea Denecke
Leibniz Universität IT-Services

Übung 0.1

Führen Sie die folgenden Berechnungen durch:

1 plus 3 plus 5 plus 7	`1+3+5+7`
1 geteilt durch 30	`1/30`
Logarithmus der Zahl 2 zur Basis 10	`log10(2)`
Fakultät von 3	`factorial(3)`
Verdopplung der Zahlen 1 bis 10	`1:10*2`

Verdopplung der geraden Zahlen von 1 bis 10

`c(2,4,6,8,10)*2`

eleganter geht dies mit der Funktion `seq (Startwert, Endwert,by= Schrittweite)` → `seq(2,10,by=2)*2`

Übung 0.2

Wir hatten uns eine Matrix erzeugt über

```
> b <- matrix(1:12, nrow=3)
```

Erreichen Sie, dass die Zahlen von 1 bis 12 nicht spaltenweise, sondern reihenweise in die Matrix gelegt werden!

Unter `help(matrix)` gibt es eine Auflistung der möglichen Argumente mit kurzer Beschreibung. Geeignet daher scheint das Argument byrow=TRUE, also

```
b <- matrix(1:12, nrow=3, byrow=TRUE)
```

Verändern Sie die Zahl 8 in den Wert 11

Die Zahl 8 hat in der Matrix jetzt die Position Reihe 2, Zeile 4. Dieser Position wird der Wert 11 zugewiesen:

```
b[2,4] <- 11
```

Übung 1

1. *Erzeugen Sie in R einen Datensatz „bundesliga“ basierend auf der CSV-Datei „bundesliga.csv“*
   `bundesliga <- read.csv2(file.choose())` im sich öffnenden Explorer die csv-Datei „bundesliga“ heraus suchen, anklicken.
2. *Verschaffen Sie sich einen Überblick über die Daten: welche Variablen gibt es?, wie viele Fälle?, etc.*
   Diese Infos erhält man z.B. über `summary(bundesliga)`, `str(bundesliga)`, `names(bundesliga)` …
3. *Lassen Sie sich die Werte der Variable „wochtag“ anzeigen*
   `bundesliga[,"wochtag"]`
4. *Lassen Sie sich die ersten 10 Werte der Heimtore und Gasttore anzeigen*
   `bundesliga[1:10,c(2,4)]`
5. *Lassen Sie sich die Spiele anzeigen, in denen die Gäste mindestens zwei Tore geschossen haben*
   `subset(bundesliga, gasttore>=2)`

Übung 2

1. *Berechnen Sie in der gleichen Weise die Variable „tore_gegn“*
```
bund_Hamb$tore_gegn <- ifelse(test =
bund_Hamb$spielort == „2“,
yes=bund_Hamb$heimtore,no=bund_Hamb$gasttore)
```
2. *Berechnen Sie nun eine Variable „tore_diff“ (Tordifferenz, aus Sicht von Hamburg) aus den Variablen „tore_Hamb“ und „tore_gegn“*
```
bund_Hamb$tore_diff <-
bund_Hamb$tore_Hamb – bund_Hamb$tore_gegn
```
3. *Berechnen Sie die jeweiligen Mittelwerte für die beiden Variablen* „tore_Hamb“ und „tore_gegn“
```
mean(bund_Hamb$tore_Hamb); mean(bund_Hamb$tore_gegn)
```
4. *Geben Sie den Befehl* `str(bund_Hamb)` *ein.*

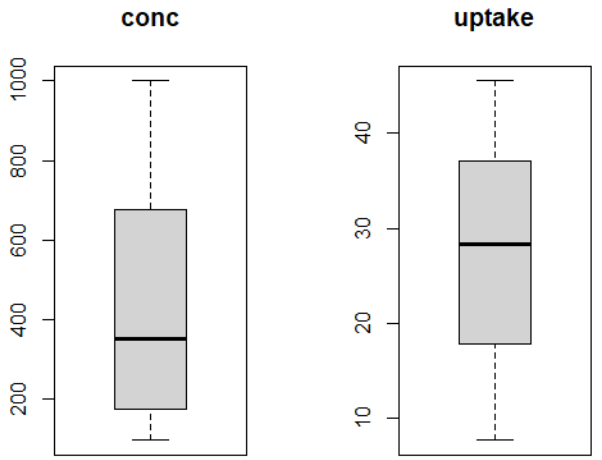
Übung 3

1. *Erzeugen Sie in dem Datensatz „bund_Hamb“ eine Variable „ergebnis“, die angibt, ob es sich um einen Sieg von Hamburg, eine Niederlage oder ein Unentschieden handelt*
   ```
   bund_Hamb$ergebnis <- recode(bund_Hamb$tore_diff, 'lo:-1="Niederlage"; 0="unentschieden";1:hi="Sieg"')
   ```
2. *Wie oft hat Hamburg gewonnen, wie oft verloren und wie oft unentschieden gespielt? Wie ist die Verteilung in Prozent?*
   ```
   table(bund_Hamb$ergebnis)
   prop.table(table(bund_Hamb$ergebnis))
   ```
3. *Erzeugen Sie ein Balkendiagramm für die Variable „ergebnis“. Verändern Sie die Farbe der Balken. Fügen Sie einen Titel „Spielergebnisse des Hamburger SV“ hinzu. Probieren Sie ggf. noch andere low-level plotting commands aus!*
   ```
   barplot(table(bund_Hamb$ergebnis),
   col=„tomato“,main=„Spielergebnisse des HSV“, …)
   ```
4. *Speichern Sie dieses Diagramm als .jpeg* : im Grafikfenster auf „Export“, „Save as Image“.

Übung 3.1

1. *Untersuchen Sie die Variablen „conc“ und „uptake“ auf Normalverteilung (grafisch, Shapiro-Wilk-Test, Schiefe, Kurtosis,…)*
   Hochladen und Ansehen der Testdatei: `data(„CO2“); View(CO2)`
   grafische Darstellung: `par(mfrow=c(1,2)); boxplot(CO2$conc); boxplot(CO2$uptake)` oder `qqnorm(CO2$uptake); qqline(CO2$uptake)`
   Paket e1071 installieren (`install.packages(e1071)`) und hochladen (`library(e1071)`).
   `skewness(CO2$conc), kurtosis(CO2$conc)`
   `shapiro.test(CO2$conc)`, entsprechend das Gleiche für „uptake“.

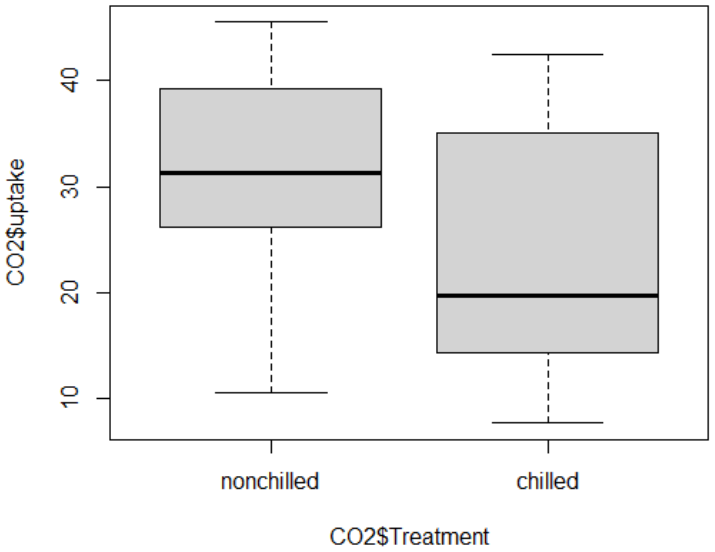
   „Conc“ zeigt rechtsschiefe (0.72) und abgeflachte (-0.68) Verteilung, allerdings im unkritischen Bereich. „uptake“ ist wenig schief (-0.1), aber der Kurtosis-Wert ist mit -1.35 kritisch, Verteilung also abgeflacht. Beide Verteilungen eher nicht als NV ansehen, Shapiro-Wilk Test bestätigt dies (5.146e-07 & 0.0007908).



Übung 3.1

2. *Die Variable „Treatment“ dient nun als Gruppierungsvariable. Wie sieht es bei der Variable “uptake“ mit der Normalverteilung in den beiden Gruppen aus?*

   `boxplot(CO2$uptake ~ CO2$Treatment)`
   `tapply(CO2$uptake,CO2$Treatment, shapiro.test)`, „Chilled“ deutlich rechtsschief, beide Gruppen nicht NV (0.043 und 0.0012)



   Berechnen Sie den Mittelwert von „uptake“ in den beiden „Treatment“-Gruppen.
   `tapply(CO2$uptake,CO2$Treatment,mean)`
   nonchilled 30.64286; chilled 23.78333

   Sind die beiden Gruppen varianzhomogen?
   Auch wieder
   `boxplot(CO2$uptake ~ CO2$Treatment)` sowie
   `leveneTest(CO2$uptake ~ CO2$Treatment)`
   p-Wert= 0.26, können also als varianzhomogen angesehen werden

Übung 4

1. *Erstellen Sie eine Variable, die den Viehbestand (livestock) in die Gruppen 1 (niedrig, <59000) und 2 (hoch, >59000) einteilt.*
   ```
   nitrate$ls_g <- recode(nitrate$livestock, 'lo:59000=1; 59000:hi=2')
   ```
2. *Untersuchen Sie die Hypothese, dass Regionen mit hohen Viehbeständen (Gruppe 2) signifikant höhere Nitratgehalte (nitrate) haben als solche mit geringeren Viehbeständen.*
   ```
   boxplot(nitrate$nitrate ~ nitrate$ls_g)
   ```
   Gr. 1 halbwegs normalverteilt, etwas schief, Gr.2 etwas deutlicher schief und langschwänzig.
   ```
   tapply(nitrate$nitrate, nitrate$ls_g, shapiro.test)
   ```
   Gruppe 1:0.2461, Gruppe 2:0.06991, man kann also noch NV annehmen.
   ```
   leveneTest(nitrate$nitrate~ nitrate$ls_g)
   ```
   p-Wert 0.089, also nicht varianzhomogen

Übung 4

2ff. Also würde die Wahl auf einen parametrischen Test fallen, der keine homogenen Varianzen erfordert: t-Test nach Welch. Zu beachten ist die einseitige Fragestellung (Gr. 1< Gr.2).

```
t.test(nitrate$nitrate~ nitrate$ls_g, var.equal=F, alternative=„l“)
```

Es ergibt sich ein p-Wert von 0.0066, d.h. Regionen mit hohen Viehbeständen haben signifikant höhere Nitratgehalte im Boden als solche mit geringeren Viehbeständen.

3. *Versuchen Sie, grafisch den Nitratgehalt (nitrate) der verschiedenen Bodenarten (soil) darzustellen.*

z.B. `boxplot(nitrate$nitrate ~ nitrate$soil)` oder

```
nitbod <- tapply(nitrate$nitrate, nitrate$soil, mean)
barplot(nitbod)
```

Übung 4

4. *Bestehen Unterschiede hinsichtlich des Nitratgehalts zwischen den vier Bodenarten? (Verzichten Sie auf die paarweisen Vergleiche)*
   Eigentlich sollte man sich zuerst die Boxplots anschauen, dann `tapply(nitrate$nitrate, nitrate$soil, shapiro.test)` um zu sehen, ob NV angenommen werden kann. Aber: `table(nitrate$soil)` zeigt, dass wir sehr geringe Fallzahlen haben, nur 4 Fälle bei „s“, „t“ 7, „l“ 9 und „u“ 10. Man sollte hier also von Vorneherein einen nicht-parametrischen Test wählen: Kruskal-Wallis.
   `kruskal.test(nitrate$nitrate ~ nitrate$soil)`

   Der p-Wert ist 0.0093, es liegen also signifikante Unterschiede zwischen den Gruppen vor.

Für Schnelle: *Lösen Sie Aufgabe 1 mithilfe der Funktion cut*.

Z.B. funktioniert

```
nitrate$ls_g2 <- cut(nitrate$livestock, breaks=2,
labels=c(1,2))
```

Übung 5

Berechnen Sie den Korrelationskoeffizienten von der Bodenart (soil) und dem Nitratgehalt. (Die Bodenart muss zuvor als numerische Variable angelegt werden: Reihenfolge ist sult)

```
nitrate$soilN <- recode(nitrate$soil,
'"s"=1;"u"=2;"l"=3;"t"=4', as.factor=F)
```

Da „soilN“ eine ordinale Variable ist, sparen wir uns die Untersuchung auf NV und wählen gleich einen nicht-parametrischen Test (hier Kendall‘s Tau, da kleine Fallzahlen)

```
cor.test(nitrate$nitrate, nitrate$soilN, method=„k")
```

Es ergibt sich ein Korrelationskoeffizient von -0.42, d.h. es besteht eine relativ geringe, negative Korrelation.

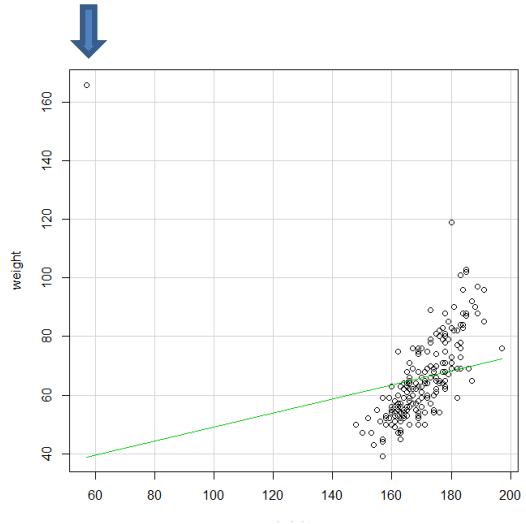
Übung 6

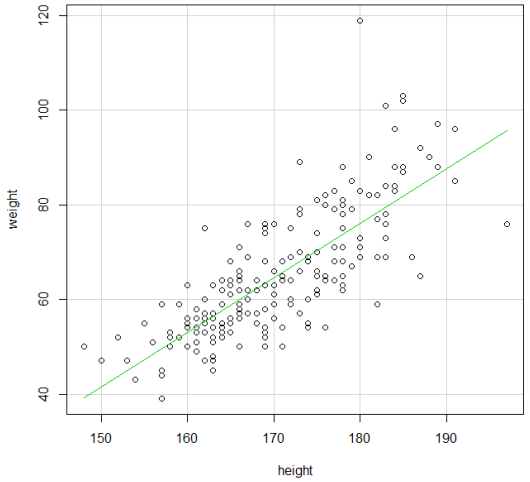
Laden Sie den Testdatensatz „Davis“ (Paket „car“). Inspizieren Sie die Daten sorgfältig!

1. Zeigen Sie, dass ein linearer Zusammenhang zwischen der Größe und dem Gewicht besteht (Korrelation).
   - Streudiagramm
   - Normalverteilte Variablen?
   - p-Wert

2. Führen Sie anhand der Daten eine lineare Regression durch („height“ ist vorherzusagende Variable, „weight“ die erklärende).
   - Streudiagramm (wie oben)
   - Regressionsmodell
   - Residuenanalyse auf
     - Unabhängigkeit
     - Normalverteilung
     - Homogene Varianzen

Übung 6



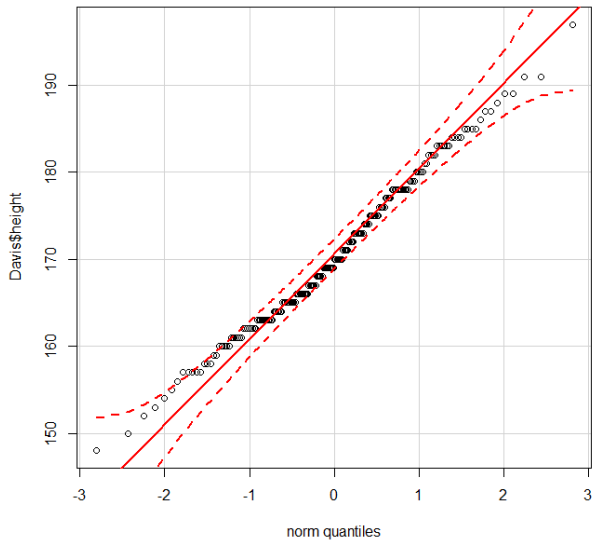


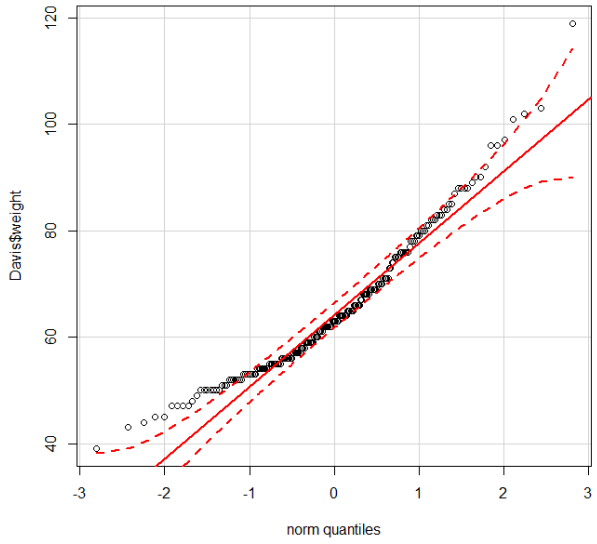
Im Streudiagramm der Daten werden Sie einen extremen Ausreißer entdecken (Pfeil). Beim Inspizieren der Daten fällt auf, dass diese Person ein Gewicht von 166 kg bei einer Größe von 57 cm aufweist, was natürlich unmöglich ist. Da das „reported weight“ (repwt) 56 kg betrug und die „reported height“ (repht) 163 cm, kann man von einer Vertauschung der Daten ausgehen.

Daher kann man in diesem Fall die Werte austauschen. Das neue Streudiagramm sieht deutlich sinnvoller aus.

Ob die Variablen Gewicht (weight) und Größe (height) normalverteilt sind, kann man z.B. durch Q-Q-plots feststellen (nächste Seite).

Übung 6





Anhand des Q-Q-plots von „height“ (hier mal andere QQ-Plot-Funktion {car}: `qqPlot(Davis$height,„norm“)`) kann man keine Abweichung von der Normalverteilung feststellen. Der Shapiro-Wilk-Test bestätigt diese Annahme (p=0.1697) (`shapiro.test(Davis$height)`).
Für die Variable „weight“ zeigt der Q-Q-plot eine Abweichung von der Normalverteilung, der Shapiro-Wilk-Test bestätigt dies (p=8.434e-07).
Da die Voraussetzung der Normalverteilung nicht gegeben ist, sollte der „Spearman‘s“ Test verwendet werden.
`cor.test(Davis$height,Davis$weight,method=„s“)`
Ergebnis: rho= 0.79 → eine mittelstarke positive Korrelation zwischen der Größe und dem Gewicht liegt vor.

Übung 6

Lineare Regression der Größe und des Gewichts:

```
Residuals:
     Min       1Q   Median       3Q      Max
-18.3492  -3.7151  -0.2466   3.5349  20.8802
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 136.83054    2.02039   67.72   <2e-16 ***
weight        0.51696    0.03034   17.04   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.702 on 198 degrees of freedom
Multiple R-squared: 0.5946,       Adjusted R-squared: 0.5925
F-statistic: 290.4 on 1 and 198 DF,  p-value: < 2.2e-16
```

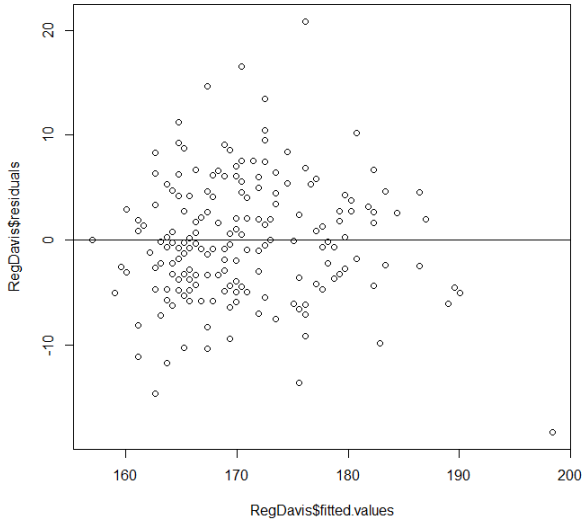
Regressionsmodell→ height = 0.51696*weight + 136.83054

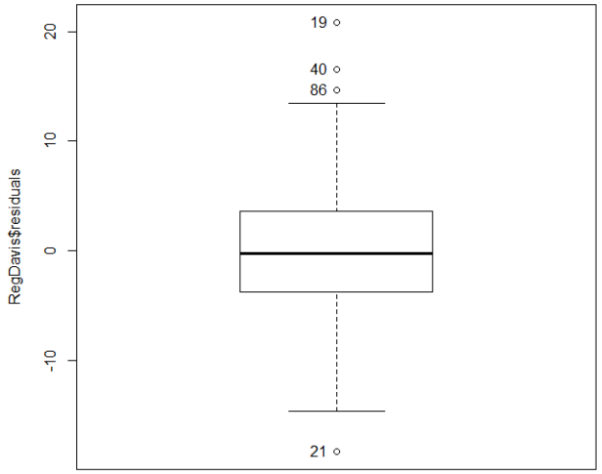
R^2= 0.5946, besagt, dass fast 60% der Größe einer Person durch dessen Gewicht erklärt werden (in diesem Modell).

Übung 6

Die Güte des Modells muss überprüft werden:

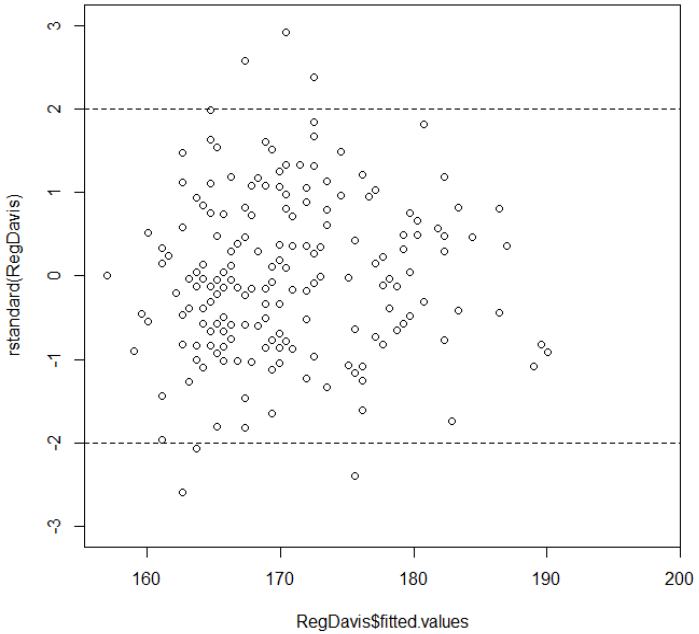
1. Unabhängigkeit der Residuen: `plot(fitted.values, residuals, data=reg.davis); abline(h=0);` kein Muster erkennbar; `DurbinWatsonTest(reg.davis)` bestätigt dies (1.94)
2. Normalverteilung der Residuen: zur Abwechselung mal ein `boxplot(reg.Davis$residuals)`, `qqnorm(reg.davis$residuals)` wäre auch gegangen. Die visuelle Inspektion des Boxplots zeigt keine Abweichung von der Normalverteilung (nur einige Ausreißer), der Shapiro-Wilk Test bestätigt dies (p= 0.2317838).





Übung 6

3. Varianzen der Residuen sind gleich :
`plot(reg.davis$fitted.values, rstandard(reg.davis)); abline(h=c(-2,2), lty=2)` Ausreißer <5% und die Punkte sind annähernd gleich verteilt, kein Muster oder „Trichter“ erkennbar, daher kann die Gleichheit der Varianzen angenommen werden.



Übung 7

1. *Erzeugen Sie eine Funktion zur Berechnung der Photosyntheseleistung pho der Pflanzen: pho=u*a*b*

```
pho <- function (u,a,b) {u*a*b}
```
#Anlegen der function pho

2. *Wenden Sie diese Funktion an (Datensatz CO2)*

```
CO2$pho <- pho(CO2$uptake,a <- ifelse(CO2$Type=="Quebec", 1,1.5),
                          b <- ifelse(CO2$Treatment=="chilled", 0.8, 0.5))
```
man passt mit der ifelse-Funktion den jeweiligen a und b Wert an.

3. *Wie hoch ist nach der Binomialverteilung die Wahrscheinlichkeit, bei 5 Kindern*

- *genau 2 Jungen zu bekommen* `dbinom(2,5,0.5)`, also 0.3125.
- *höchstens zwei Jungen zu bekommen?* `pbinom(2,5,0.5)` oder `dbinom(0,5,0.5) + dbinom(1,5,0.5) + dbinom(2,5,0.5)`, also 0.5.
- *Grafische Darstellung* `plot(dbinom(0:5, 5, 0.5))`