

Wintersemester 2022

Jens Klenke

## R Propädeutikum

### Lösung Übungsaufgaben 3

#### 1 Verteilungen und Zufallszahlen

**1.1** Sei  $X \sim t(5)$ . Berechne  $P(X < 6)$ ,  $P(3 < X \leq 7)$  und  $P(X > 4)$ .

- $P(X < 6)$

```
pt(6, df = 5)
```

```
## [1] 0.9990769
```

- $P(3 < X \leq 7)$

```
pt(7, df = 5) - pt(3, df = 5)
```

```
## [1] 0.01459125
```

- $P(X > 4)$ .

```
1 - pt(4, df = 5)
```

```
## [1] 0.005161708
```

**1.2** Berechne das 0.95-Quantil einer  $F(4, 5)$ -verteilten Zufallsvariable.

```
qf(0.95, df1 = 4, df2 = 5)
```

```
## [1] 5.192168
```

- 1.3 Berechne die Wahrscheinlichkeit dafür, den Jackpot im Lotto zu gewinnen (d.h. 6 Richtige aus 49). Vernachlässige Zusatz- und Superzahlen. (Hinweis: Benutze die hypergeometrische Verteilung.)

```
dhyp(6, m = 6, n = 43, k = 6)
```

```
## [1] 7.151124e-08
```

```
# oder per Binomialkoeffizient:  
# 1/choose(49, 6)
```

- 1.4 Erzeuge 20  $\chi^2(5)$ -verteilte Zufallszahlen ohne (!) dabei die `rchisq()`-Funktion zu benutzen.

Hinweis:  $\chi^2(n) = \sum_{i=1}^n Z_i^2$  mit  $Z_i \sim \mathcal{N}(0, 1)$  für alle  $i = 1, \dots, n$ .

```
# seed damit bei beiden Befehlen die Ergebnisse gleich sind  
set.seed(1549)  
# schnell:  
replicate(20, sum(rnorm(5)^2))
```

```
## [1] 7.091977 4.967954 5.514013 4.676718 3.931540 7.028298  
## [7] 13.070791 6.461356 7.643946 3.144983 6.943869 10.710932  
## [13] 5.536349 6.237411 5.422414 1.965647 3.319014 3.732634  
## [19] 8.127858 4.358846
```

```
# seed damit bei beiden Befehlen die Ergebnisse gleich sind  
set.seed(1549)  
# mit schleife:  
rn <- numeric()  
for(i in 1:20){  
  rn[i] <- sum(rnorm(5)^2)  
}  
rn
```

```
## [1] 7.091977 4.967954 5.514013 4.676718 3.931540 7.028298  
## [7] 13.070791 6.461356 7.643946 3.144983 6.943869 10.710932  
## [13] 5.536349 6.237411 5.422414 1.965647 3.319014 3.732634  
## [19] 8.127858 4.358846
```

- 1.5 Ziehe 10 mal  $n = 10000$  standardnormalverteilte Zufallszahlen und berechne für jede Ziehung das arithmetische Mittel. Betrachte die 10 Mittelwerte. Was fällt auf?**

```
norm <- numeric()
for(i in 1:10){
  norm[i] <- mean(rnorm(10000))
}
norm
```

```
## [1] -0.007993253  0.000205263 -0.023010730  0.008032737 -0.015314207
## [6] -0.006583092 -0.010186764 -0.002472349 -0.001919459 -0.006736107
```

Alle 10 Mittelwerte liegen sehr nahe bei 0. Nicht überraschend, da wir Zufallszahlen mit Erwartungswert 0 ziehen.

**Zusatzaufgabe:** Wdh. mit Cauchy-Verteilung

```
cauchy <- numeric()
for(i in 1:10){
  cauchy[i] <- mean(rt(10000, df = 1))
}
cauchy
```

```
## [1]  2.86061674  0.28687124 20.24954554 -2.26699594 -0.93806437
## [6] -0.09852194  0.06981833  0.03190819  0.16937363  0.36625186
```

Nun liegen die Mittelwerte nicht wirklich nahe beieinander. Erklärung: die Cauchy-Verteilung hat keinen Erwartungswert.

- 1.6 Zusatz: Prüfe, dass das Integral über die Dichtefunktion einer  $\chi^2(15)$ -verteilten Zufallsvariable 1 ist.**

```
integrate(dchisq, lower = 0, upper = Inf, df = 15)
```

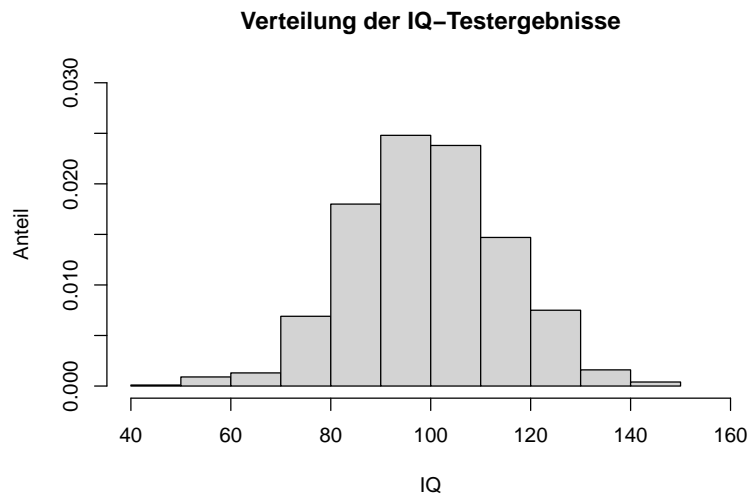
```
## 1 with absolute error < 1.5e-05
```

## 2 Grafiken

```
set.seed(385)
results <- rnorm(1000, mean = 100, sd = 15)
```

- 2.1 Kopieren Sie obigen Code und nehmen Sie an, dass dieser eine IQ-Testreihe mit 1000 Probanden simuliert. Zeichnen Sie ein Histogramm der Ergebnisse. Geben Sie Ihrem Plot anschließend eine passende Überschrift sowie passende Achsenbeschriftungen. Spezifizieren Sie darüber hinaus den Bereich von x- und y-Achse auf  $[40, 160]$  bzw.  $[0, 0.03]$ .

```
set.seed(385)
results <- rnorm(1000, mean = 100, sd = 15)
hist(results,
      freq = F,
      xlab = 'IQ',
      ylab = 'Anteil',
      main = 'Verteilung der IQ-Testergebnisse',
      xlim = c(40, 160),
      ylim = c(0, 0.03))
```

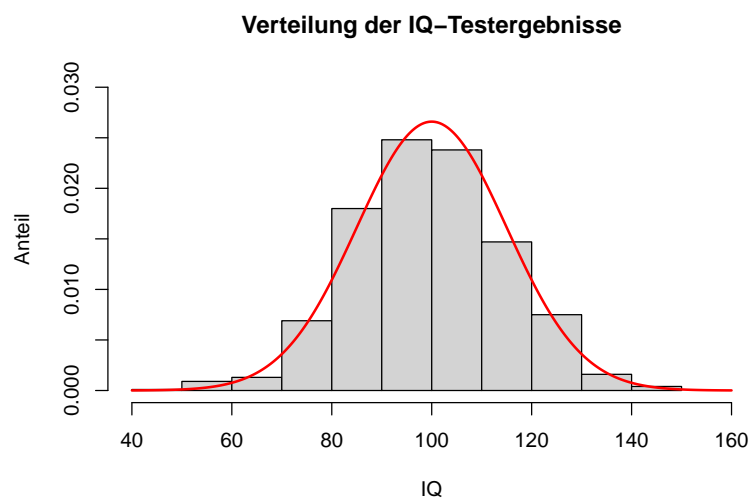


- 2.2 Hinterlegen Sie dem Plot die dem IQ zugrundeliegende, theoretische Dichtefunktion, d.h. eine Normalverteilung mit  $\mu = 100$  und  $\sigma = 15$ . Wählen Sie als Zeichenfarbe Rot und machen Sie die einzuzeichnende Linie etwas breiter.

```

set.seed(385)
results <- rnorm(1000, mean = 100, sd = 15)
hist(results, freq = F, xlab = 'IQ', ylab = 'Anteil',
      main = 'Verteilung der IQ-Testergebnisse',
      xlim = c(40, 160), ylim = c(0, 0.03))
x <- seq(40, 160, 0.01)
y <- dnorm(x, mean = 100, sd = 15)
lines(x, y, col = 'red', lwd = 2)

```



**2.3 Zeichnen Sie einen Punkt in Form eines Dreiecks an das Maximum der theoretischen Dichte. Wählen Sie als Farbe Blau.**

```

hist(results,
      freq = F,
      xlab = 'IQ',
      ylab = 'Anteil',
      main = 'Verteilung der IQ-Testergebnisse',
      xlim = c(40, 160),
      ylim = c(0, 0.03))

x <- seq(40, 160, 0.01)
y <- dnorm(x, mean = 100, sd = 15)

lines(x,
      y,
      col = 'red',

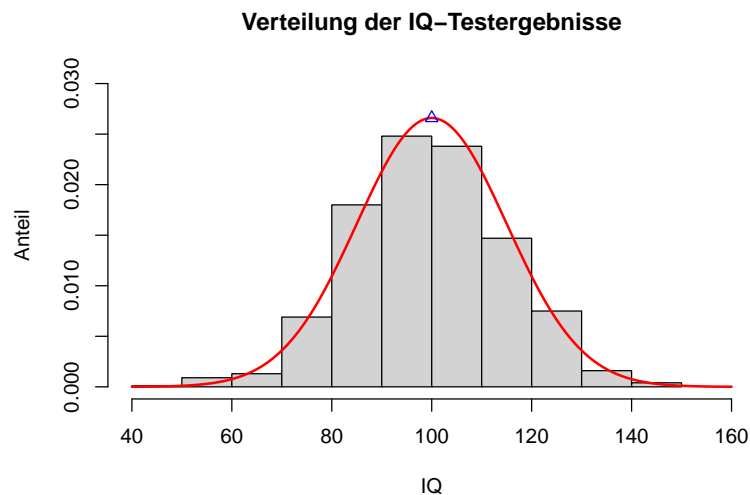
```

```

lwd = 2)

points(100,
       dnorm(100, mean = 100, sd = 15),
       pch = 2,
       col = 'blue')

```



**2.4 Kennzeichnen Sie sowohl das 0.025- als auch das 0.975-Quantil der theoretischen Dichte, in dem Sie Vertikalen an diesen Punkten einzeichnen. Wählen Sie als Farbe Grün.**

```

hist(results,
     freq = F,
     xlab = 'IQ',
     ylab = 'Anteil',
     main = 'Verteilung der IQ-Testergebnisse',
     xlim = c(40, 160),
     ylim = c(0, 0.03))

x <- seq(40, 160, 0.01)
y <- dnorm(x, mean = 100, sd = 15)

lines(x, y, col = 'red', lwd = 2)

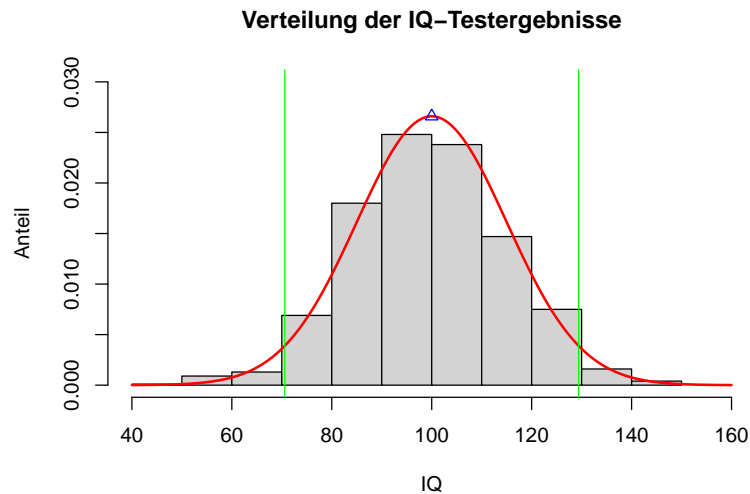
points(100,
       dnorm(100, mean = 100, sd = 15),
       pch = 2,

```

```
col = 'blue')
```

```
abline(v = qnorm(0.025, mean = 100, sd = 15), col = 'green')
```

```
abline(v = qnorm(0.975, mean = 100, sd = 15), col = 'green')
```



### 3 Lineare Regression

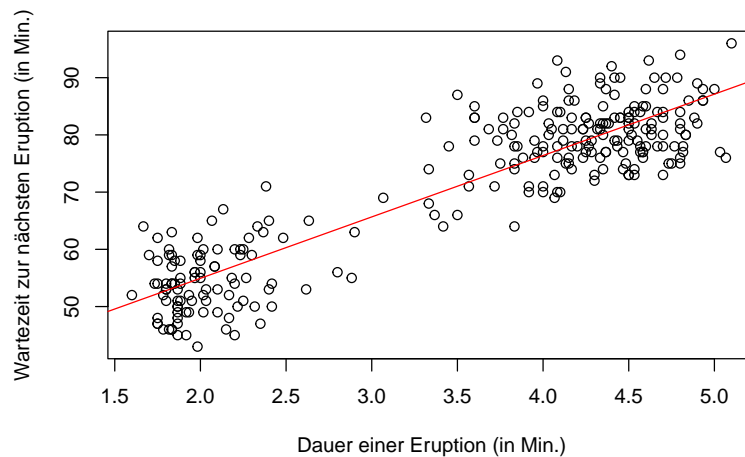
- 3.1 Betrachten Sie im Folgenden den Datensatz `faithful`, der Daten zum Old Faithful Geysir im Yellowstone Nationalpark enthält. Sowohl die Dauer einer Eruption in Min. (`eruptions`) als auch die Wartezeit bis zur nächsten Eruption in Min. (`waiting`) sind als Variable im Datensatz verfügbar. Unterstellen Sie nachfolgendes Regressionsmodell und schätzen Sie die entsprechenden Parameter  $\hat{\beta}_0, \hat{\beta}_1$ . Zeichnen Sie anschließend eine geeignete Grafik und interpretieren Sie diese.

$$\text{waiting}_t = \beta_0 + \beta_1 \text{eruptions}_t + u_t$$

```
model_ff <- lm(waiting ~ eruptions, data = faithful)
```

```
plot(faithful$eruptions,  
     faithful$waiting,  
     xlab = 'Dauer einer Eruption (in Min.)',  
     ylab = 'Wartezeit zur nächsten Eruption (in Min.)')
```

```
abline(model_ff, col = 'red')
```



Grundsätzlich: Je länger die Dauer einer Eruption ist, desto länger ist die Wartezeit bis zur nächsten Eruption.

**3.2 Verschaffen Sie sich mit `summary()` einen Überblick über ihr in 3.1) erhaltenes Ergebnis. Interpretieren Sie die geschätzten Koeffizienten und speichern Sie anschließend das  $R^2$  (Multiple R-squared) in der Variablen R2 ab. (Hinweis: Schauen Sie sich die, beim Ausführen von `summary()`, ausgegebene Datenstruktur genauer an.)**

```
summary(model_ff)
```

```
##
## Call:
## lm(formula = waiting ~ eruptions, data = faithful)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0796  -4.4831   0.2122   3.9246  15.9719
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.4744     1.1549   28.98  <2e-16 ***
## eruptions    10.7296     0.3148   34.09  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.914 on 270 degrees of freedom
```



```
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

- Achsenabschnitt (Intercept)
  - Bei einer Eruptionsdauer von 0 Minuten, beträgt die Wartezeit bis zur nächsten Eruption etwa 33.5 Minuten (nicht wirklich sinnvoll interpretierbar).
- Steigungskoeffizient (eruptions)
  - Steigt die Dauer einer Eruption um eine Minute, steigt die Wartezeit bis zur nächsten Eruption um etwa 10.7 Minuten.

```
R2 <- summary(model_ff)$r.squared
R2
```

```
## [1] 0.8114608
```

Summary erstellt ebenfalls eine Liste, in der u. a. auch das  $R^2$  enthalten ist.

**3.3 Angenommen Sie beobachten einen zusätzlichen Datenpunkt für die Dauer einer Eruption von  $X_{new} = 4$ . Sagen Sie die entsprechende Wartezeit bis zur nächsten Eruption vorher.**

```
new_value <- data.frame(eruptions = 4)
predict(model_ff, newdata = new_value)
```

```
##          1
## 76.39296
```

Bei einer Eruptionsdauer von 4 Min. beträgt die mit unserem Modell geschätzte Wartezeit etwa. 76.4 Min.