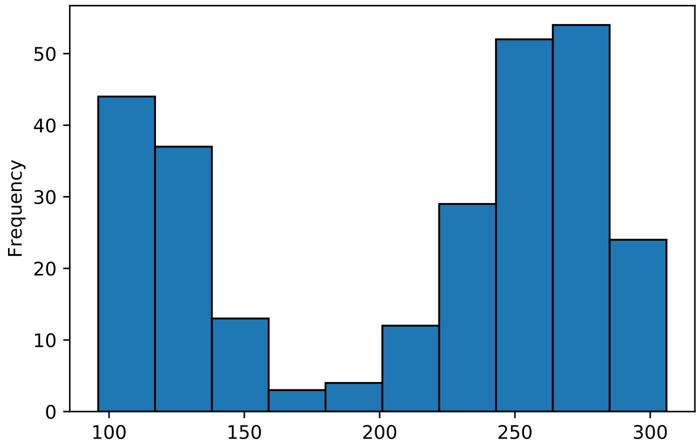
Serie 7

Aufgabe 7.1

Old Faithful ist ein Geysir im Yellowstone National Park in Wyoming:

https://en.wikipedia.org/wiki/Old_Faithful

Im Datensatz `oldfaithful.txt` sind Längen von 272 aufeinanderfolgenden Ausbrüchen aufgeführt. Hier ist ein Histogramm der Daten.



a) Schätzen Sie den Mittelwert der Eruptionsdauern ab und geben Sie ein 95 % Vertrauensintervall mit dem vereinfachten Bootstrap-Verfahren im Skript an. Verwenden Sie dazu 1000 Bootstraps.

Setzen Sie für $*$ den Pfad, wo sich Ihr `oldfaithful.txt`

```
import numpy as np

x = np.loadtxt(r"*/oldfaithful.txt")

n = np.size(x)

nboot = ....

tmpdata = np.random.choice(x, n*nboot, replace=True)

bootstrapsample = np.reshape(tmpdata, (n, nboot))

xbarstar = np.mean(bootstrapsample, axis=0)
```

```
d = np.percentile(xbarstar, q=[2.5, 97.5])

print('Vertrauensintervall: ',d)
```

b) Machen Sie dasselbe für den Median.

c) Schätzen Sie die Wahrscheinlichkeit

$$P(|\overline{x} - \mu| > 5)$$

ab.

Ersetzen Sie zur Abschätzung μ durch den Mittelwert der Daten und $\overline{x}$ durch die Bootstraps.

Aufgabe 7.2

Aus der uniformen Verteilung $X \sim \text{Uniform}([0, 10])$ soll eine Stichprobe vom Umfang n gezogen werden.

a) Es sei $n = 60$. Bestimmen Sie ein symmetrisches Intervall

$$I = [\mu_X - e, \mu_X + e]$$

um den Erwartungswert μ_X so, dass sich das arithmetische Mittel der Stichprobe, also $\overline{X}_{60}$, mit der Wahrscheinlichkeit von 95% in I befindet. Ein solches Intervall heisst **Prognoseintervall**. *Hinweis:* Standardisieren Sie das arithmetische Mittel $\overline{X}_n$ und benützen Sie den Zentralen Grenzwertsatz und $P(\overline{X}_n \in I) = 0.95$.

b) Umgekehrt: Wie gross muss n gewählt werden, damit $e = 0.2$ wird?

c) Überprüfen Sie a) experimentell, d. h. mit **Python** : ziehen Sie viele Stichproben (z. B. 200) und zählen Sie, wie viele ausserhalb von I liegen. **Python** -Hinweise:

```
from scipy.stats import uniform
import matplotlib.pyplot as plt
import numpy as np
n = 60   # Anzahl Stichproben
# X_1,...,X_n simulieren und in einer
# n-spaltigen Matrix (mit 200 Zeilen) anordnen
sim = uniform.rvs(size=200*n, loc=0, scale=1)
sim = sim.reshape(200, n)
#In jeder Matrixzeile Mittelwert berechnen
```

```
sim_mean = sim.mean(axis=1)
plt.plot(np.arange(1,201,1),sim_mean)
# Zeichnen Sie mit axhline(y=...) die
# Intervallgrenzen des Prognoseintervalls
# in der obigen Graphik ein.
```

Aufgabe 7.3

Aufgrund langjähriger Untersuchungen ist bekannt, dass der Bleigehalt X von gewissen Bodenproben annähernd normal-verteilt ist

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

a) Es wurden in 10 Bodenproben der Bleigehalt X gemessen. Dabei wurde ein Mittelwert von $\overline{x}_{10} = 31$ ppb erhalten. Die Standardabweichung sei bekannt und beträgt 6 ppb.

   Geben Sie das zweiseitige 99 % Vertrauensintervall für den Mittelwert an.

b) Wie viele Beobachtungen sind nötig, um die Breite des in Teilaufgabe a) bestimmten zweiseitigen Vertrauensintervalls auf die Hälfte zu reduzieren?

   Wie viele (unabhängige) Bestimmungen des Bleigehalts müssen geplant werden, falls der Bleigehalt mit einer Stichprobe „1 ppb genau" bestimmt werden soll, d. h., wenn die Breite des 99 % des Konfidenzintervalls nicht grösser als 1 ppb sein soll?

c) Normalerweise ist die Standardabweichung σ unbekannt. Um welchen Faktor verändert sich die Breite des zweiseitigen Vertrauensintervalls in Teilaufgabe a), wenn man die Standardabweichung aus den Daten geschätzt hat, also $\widehat{\sigma} = 6$?

Aufgabe 7.4

Im National Bureau of Standards (USA) wurden regelmässig Wägungen des 10-Gramm - Standardgewichtstücks durchgeführt. Bei 9 Wägungen erhielt man als durchschnittliche Differenz -403 Mikrogramm vom 10 Gramm-Sollgewicht und eine Standardabweichung von 3.127 Mikrogramm für eine einzelne Wägung.

a) Geben Sie das exakte, zweiseitige 95 %-Vertrauensintervall für die wahre Differenz an, unter der Annahme, dass die Messfehler normal-verteilt sind.

b) Könnte die wahre Differenz $-400.0\,\mu$g betragen? Entscheiden Sie aufgrund des Resultats in Aufgabe a). (Kurze Begründung)

Aufgabe 7.5

Eine Brücke soll aufgrund des höheren Verkehrsaufkommens renoviert werden. Im Bau wurden damals Schrauben mit einer mittleren Festigkeit von $500\,\mathrm{N/mm^2}$ benutzt. Da dies für nicht mehr sicher genug gehalten wird, sollen diese nun durch Schrauben mit einer mittleren Festigkeit von mehr als $500\,\mathrm{N/mm^2}$ ersetzt werden.

Um diesen Anforderungen gerecht zu werden, hat der alte Schraubenlieferant ein neues Verfahren entwickelt. Zur Baustelle werden allerdings unbeschriftete Schrauben geliefert, aus denen nicht sofort hervorgeht, ob es sich um die alten 500er oder um die neuen verbesserten Schrauben handelt.

Vor dem Verbau will der leitende Ingenieur zuerst sicherstellen, dass es sich um die besseren Schrauben handelt. Um dies herauszufinden, werden einige der Schrauben vermessen und ein statistischer Test durchgeführt. Je nach Ergebnis sollen die Schrauben verbaut oder zurückgeschickt werden.

	1	2	3	4	5
Schraubenfestigkeit (X_i)	520	512	499	524	505

Für den empirischen Mittelwert und empirische Varianz ergeben sich bei obiger Stichprobe $\overline{x}_5 = 512$ und $s_x^2 = 106.5$.

Wir modellieren die Daten mit einer Normalverteilung, d. h. X_i i.i.d. $\sim \mathcal{N}(\mu, \sigma^2)$.

a) Stellen Sie die geeigneten Null- und Alternativhypothesen auf und begründen Sie Ihre Wahl.

b) Sie führen nun einen einseitigen t-Test zum Niveau $\alpha = 0.05$ durch (unabhängig von Ihrer obigen Antwort). Stellen Sie die Teststatistik T auf und berechnen Sie deren Wert. Geben Sie die Verteilung der Teststatistik T unter H_0 und den Verwerfungsbereich des Tests an. Was ist der Testentscheid?

c) Berechnen Sie ein (zweiseitiges) 95 %-Vertrauensintervall für μ.

d) Wie würde das entsprechende Vertrauensintervall von c) aussehen, wenn wir die Streuung als bekannt voraussetzen würden (mit dem gleichen Wert wie der beobachtete)?

e) Betrachten Sie (unabhängig von dem oben aufgeführten Beispiel) einen einseitigen t-Test von $H_0 : \mu = 0$ gegen $H_A : \mu > 0$ zum Niveau 0.05.

   Obwohl die beobachteten n Datenpunkte einen empirischen Mittelwert grösser Null haben, ergeben die Berechnungen, dass die Nullhypothese nicht verworfen wird.

Entscheiden Sie, ob folgende Aussagen wahr oder falsch sind.

	Richtig	**Falsch**
Man verwirft H_0 für kein Niveau $\alpha < 0.05$	◯	◯
Es gibt ein Niveau $\alpha < 1$, bei dem man H_0 verwirft.	◯	◯
Der p-Wert ist strikt kleiner als 0.5.	◯	◯
Führt man statt eines einseitigen einen zweiseitigen Test zum Niveau 0.05 durch, verwirft man H_0 nicht.	◯	◯
Wenn man die Daten immer öfter kopiert (d. h., man betrachtet jeden Datenpunkt k-Mal, so dass man insgesamt $k \cdot n$ Datenpunkte erhält), verwirft man H_0 für ein grosses k beim Niveau 0.05.	◯	◯

Kurzlösungen vereinzelter Aufgaben

A 7.2:

a) $[4.27, 5.73]$

b) $n = 800$

A 7.3:

a) $[26.1, 35.9]$

b) 959

c) $[24.8, 37.2]$

A 7.4:

a) $[-405.4, -400.6]$

Musterlösungen zu Serie 7

Lösung 7.1

a) Für den Mittelwert gilt

```
import numpy as np


x = np.loadtxt("../../../Themen/Statistik_Messdaten/Uebungen_de/Daten/oldfa

np.mean(x)
print(np.mean(x))

## Error in running command /usr/local/bin/python3
```

Setzen **nboot=1000**:

```
import numpy as np

x = np.loadtxt("../../../Themen/Statistik_Messdaten/Uebungen_de/Daten/oldfa

n = np.size(x)

nboot = 1000

tmpdata = np.random.choice(x, n*nboot, replace=True)

bootstrapsample = np.reshape(tmpdata, (n, nboot))

xbarstar = np.mean(bootstrapsample, axis=0)

d = np.percentile(xbarstar, q=[2.5, 97.5])

print('Vertrauensintervall: ',d)

## Error in running command /usr/local/bin/python3
```

Das Vertrauensintervall ist etwa

$$I = [201, 217]$$

Da der Bootstrap vom Zufall abhängt, ergeben sich auch leicht verschiedene Resultate bei jedem Durchlauf.

b) Für den Median gilt

```
import numpy as np

x = np.loadtxt("../../../Themen/Statistik_Messdaten/Uebungen_de/Daten/oldfa

np.median(x)

## Error in running command /usr/local/bin/python3
```

Setzen `nboot=1000`:

```
import numpy as np

x = np.loadtxt("../../../Themen/Statistik_Messdaten/Uebungen_de/Daten/oldfa

n = x.size

nboot = 1000

tmpdata = np.random.choice(x, n*nboot, replace=True)

bootstrapsample = np.reshape(tmpdata, (n, nboot))

xbarstar = np.median(bootstrapsample, axis=0)

d = np.percentile(xbarstar, q=[2.5, 97.5])

print('Vertrauensintervall: ',d)

## Error in running command /usr/local/bin/python3
```

Das Vertrauensintervall ist (etwa)

$$I = [230, 247]$$

c) Code:

```
import numpy as np
x = np.loadtxt("../../../Themen/Statistik_Messdaten/Uebungen_de/Daten/oldfa

n = x.size

nboot = 1000

tmpdata = np.random.choice(x, n*nboot, replace=True)
```

```
bootstrapsample = np.reshape(tmpdata, (n, nboot))

xbarstar = np.mean(bootstrapsample, axis=0) - np.mean(x)

l = np.sum(xbarstar < -5)

u = np.sum(xbarstar > 5)

ratio = (l+u)/nboot

## Error in running command /usr/local/bin/python3
```

Die Wahrscheinlichkeit ist also etwa 0.23, dass der Durchschnitt der Messungen mehr als 5 Einheiten vom wahren Mittelwert abweicht.

Lösung 7.2

a) Für die uniforme Verteilung $X \sim \text{Uniform}([0,10])$ gilt $\text{E}(X) = \mu_X = 5$, $\sigma_X = \frac{10-0}{\sqrt{12}} = \frac{5}{\sqrt{3}}$ (siehe Kennzahlen für die uniforme Wahrscheinlichkeitsverteilung). Dann folgt aufgrund des Zentralen Grenzwertsatzes

$$\overline{X}_n \sim \mathcal{N}(\mu_X, \sigma_X^2/n)$$

Wird der arithmetische Mittelwert $\overline{X}_n$ der Stichprobe vom Umfang n standardisiert, so gilt für die standardisierte Zufallsvariable

$$Z_n = \frac{\overline{X}_n - \mu_X}{\sigma_X/\sqrt{n}} = \frac{\overline{X}_n - 5}{5/\sqrt{3n}} \sim \mathcal{N}(0,1)$$

Es gilt dann für das gesuchte Intervall $[\mu - e, \mu + e]$, dass

$$\begin{aligned}
P(\mu_X - e \leq \overline{X}_n \leq \mu_X + e) &= P\left(-\frac{e}{5/\sqrt{3n}} \leq \frac{\overline{X}_n - 5}{5/\sqrt{3n}} \leq \frac{e}{5/\sqrt{3n}}\right) \\
&= P\left(-\frac{e}{5/\sqrt{3n}} \leq Z_n \leq \frac{e}{5/\sqrt{3n}}\right) \\
&= \Phi\left(\frac{e}{5/\sqrt{3n}}\right) - \Phi\left(-\frac{e}{5/\sqrt{3n}}\right) \\
&= 0.95
\end{aligned}$$

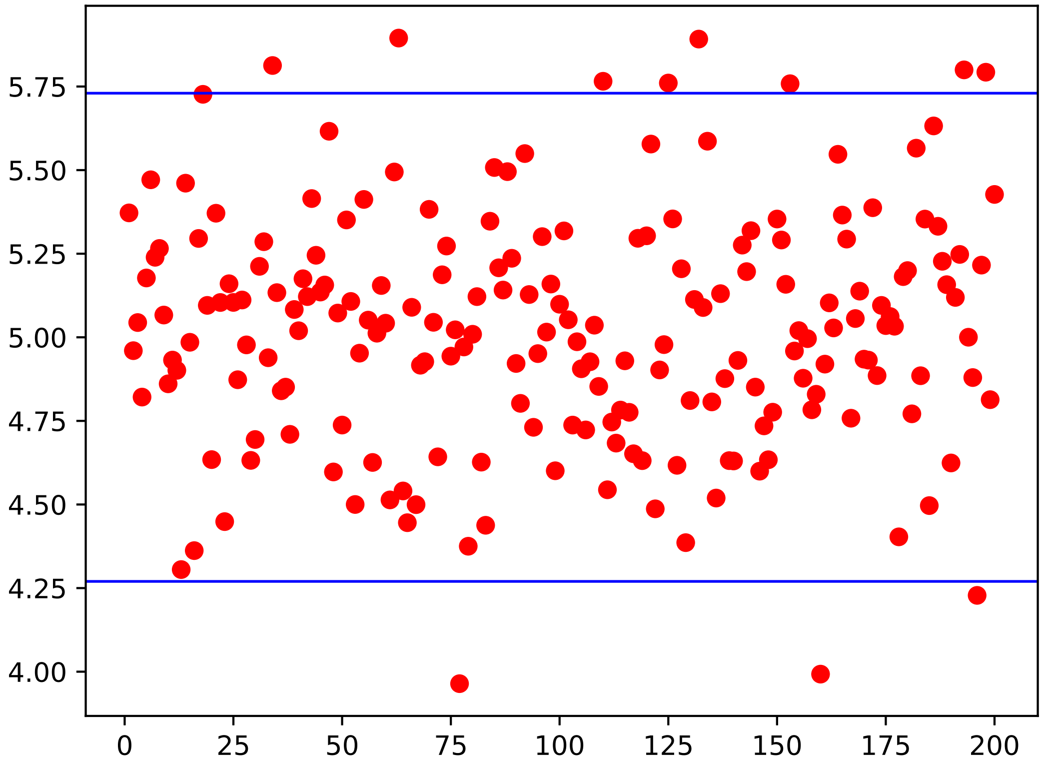
Aufgrund der Symmetrie der Standardnormalverteilung genügt es, eine Seite der Verteilung zu betrachten: $\Phi\left(\frac{e}{5/\sqrt{3n}}\right)$ soll also 97.5 % der Fläche unter der Gesamtkurve entsprechen. Das 97.5 %-Quantil $q_{0.975}$ der Standardnormalverteilung

ist
$$\frac{e}{5/\sqrt{3n}} = q_{0.975} = \Phi^{-1}(0.975) = 1.96\,.$$
Also ist
$$e = \frac{5}{\sqrt{3n}} \cdot q_{0.975} = \frac{5}{\sqrt{3 \cdot 60}} \cdot 1.96 = 0.73$$
Somit lautet das Prognoseintervall: $I = [5 - 0.73, 5 + 0.73] = [4.27, 5.73]$.

b) Auflösen der Gleichung $e = 0.2 = \frac{5}{\sqrt{3n}} \cdot 1.96$ nach n liefert $n = 800$.

c) $I = [5 - 0.73, 5 + 0.73]$, $n = 200$. Man erwartet ca. $0.05 \cdot 200 = 10$ Datenpunkte, die ausserhalb des Prognoseintervalls liegen. (zu **R**)

```
from scipy.stats import uniform
import matplotlib.pyplot as plt
import numpy as np
n = 60  # Anzahl Stichproben
# X_1,...,X_n simulieren und in einer
# n-spaltigen Matrix (mit 200 Zeilen) anordnen
sim = uniform.rvs(size=200*n, loc=0, scale=10)
sim = sim.reshape(200, n)
#In jeder Matrixzeile Mittelwert berechnen
sim_mean = sim.mean(axis=1)
plt.plot(np.arange(1,201,1),sim_mean)
axhline(y=5.73, linewidth=4, color='b')
axhline(y=4.27, linewidth=4, color='b')
```

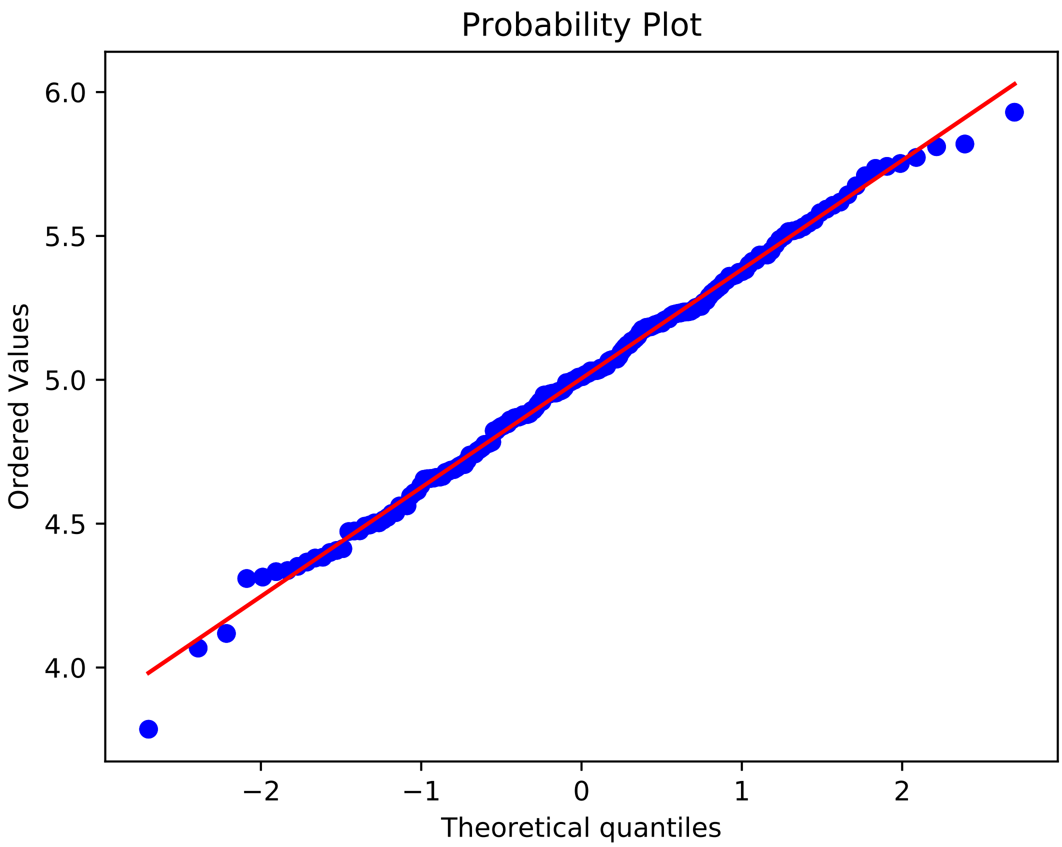



```
from scipy.stats import uniform
import matplotlib.pyplot as plt
import numpy as np
n = 60
sim = uniform.rvs(size=200*n, loc=0, scale=10)
sim = sim.reshape(200, n)
d = np.sum(sim_mean>5.73) + np.sum(sim_mean<4.27)
print(d)

## Error in running command /usr/local/bin/python3
```

Weiter bestätigt sich auch der Zentrale Grenzwertsatz: (zu **R**)

```
from scipy.stats import probplot
probplot(sim_mean, plot=plt)
```



Lösung 7.3

a) Wir bezeichnen mit X_i den Bleigehalt in der i-ten Bodenprobe. Es gilt

$$X_i \sim \mathcal{N}(\mu, \sigma^2)\,.$$

μ ist der wahre Mittelwert der Verteilung, wobei wir diesen Wert in der Praxis natürlich nicht kennen. $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ ist auch normalverteilt, allerdings mit Standardabweichung $\sigma/\sqrt{n}$, dem sogenannten *Standardfehler*

$$\overline{X}_n \sim \mathcal{N}(\mu, \sigma^2/n)$$

Nun betrachten wir in dieser Aufgabe $\overline{X}_{10}$, den Mittelwert von 10 Stichproben. $\overline{X}_{10}$ ist also normalverteilt mit Varianz $\sigma^2/n = \frac{36}{10} = 3.6$ und μ unbekannt. Der Fall, dass σ bekannt ist, μ aber nicht, ist natürlich nicht realistisch. Wir werden in einer späteren Teilaufgabe den Fall sehen, wo sowohl σ wie μ unbekannt sind. Da $\Phi(2.58) = 0.995$, liegen 99 % aller Beobachtungen von der standardisierten

Zufallsvariable

$$\frac{\overline{X}_{10}-\mu}{\sigma/\sqrt{n}} = \frac{\overline{X}_{10}-\mu}{6/\sqrt{10}}$$

in dem Intervall $[-2.58, 2.58]$. Es gilt also

$$P\left(\frac{\overline{X}_{10}-\mu}{6/\sqrt{10}} \in [-2.58, 2.58]\right) = 0.99$$

D. h. angenommen wir kennen das wahre μ, und wir bestimmen in 100 Messreihen mit jeweils 10 Stichproben $\overline{x}_{10}$, dann erwarten wir, dass der Wert $\frac{\overline{x}_{10}-\mu}{6/\sqrt{10}}$ in 99 Messreihen im Intervall $\in [-2.58, 2.58]$ liegt. Also liegen 99 % aller Beobachtungen von $\overline{X}_{10}-\mu$ im Intervall $[-2.58\cdot 6/\sqrt{10}, 2.58\cdot 6/\sqrt{10}]$. Somit gilt einerseits $\overline{X}_{10}-\mu \leq 2.58\cdot 6/\sqrt{10}$ und andererseits $-2.58\cdot 6/\sqrt{10} \leq \overline{X}_{10}-\mu$. Das 99 % Vertrauensintervall für μ ist demnach gegeben durch

$$\left[\overline{X}_{10} - 2.58\cdot\frac{6}{\sqrt{10}}, \overline{X}_{10} + 2.58\cdot\frac{6}{\sqrt{10}}\right],$$

d. h. die Intervallgrenzen sind abhängig von der Zufallsvariablen $\overline{X}_{10}$. In unserem Fall ist die Realisierung von $\overline{X}_{10} : \overline{x}_{10} = 31$. Man erhält also für ein 99 % Vertrauensintervall (für jede Realisierung von $\overline{X}_{10}$ erhält man ein leicht anderes Vertrauensintervall):

$$[26.1, 35.9]$$

Mit **Python**: (zu **R**)

```
from scipy.stats import norm, t
import numpy as np
norm.interval(alpha=0.99, loc=31, scale=6/np.sqrt(10))

## Error in running command /usr/local/bin/python3
```

b) Aus Teilaufgabe a) sieht man, dass die Breite des Vertrauensintervalls wie $1/\sqrt{n}$ abfällt mit der Anzahl n von Beobachtungen. Also sind viermal so viele Beobachtungen, $4\cdot 10$ nötig, um die Breite des Vertrauensintervalls zu halbieren. Wie aus Teilaufgabe a) ersichtlich ist die Breite des 99 % Vertrauensintervalls

$$2\cdot 2.58\cdot\frac{\sigma}{\sqrt{n}}.$$

Um die Breite des Vertrauensintervalls kleiner als 1 ppb zu erhalten, muss die Anzahl n der Beobachtungen entsprechend gross werden:

$$2\cdot 2.58\cdot\frac{\sigma}{\sqrt{n}} \leq 1$$

$$30.96 \le \sqrt{n}$$
$$n \ge 959$$

Es müssen mindestens 959 Beobachtungen vorliegen, um ein 99 % Vertrauensintervall von weniger als 1 ppb Breite zu erhalten.

c) Da σ_x unbekannt, verwenden wir eine t-Verteilung mit 9 Freiheitsgraden.

Mit **Python**: (zu **R**)

```
from scipy.stats import norm, t
import numpy as np
t.interval(alpha=0.99, df=9, loc=31, scale=6/np.sqrt(10))

## Error in running command /usr/local/bin/python3
```

Für $\hat{\sigma} = 6$, $n = 10$ und $\overline{x}_{10} = 31$ ergibt sich das Vertrauensintervall

$$[24.8, 37.2]$$

Durch Vergleich mit a) findet man, dass das Vertrauensintervall einen Faktor $3.25/2.58$, also um 26 % grösser geworden ist.

Lösung durch Standardisierung:

Die standardisierte Zufallsvariable

$$\frac{\overline{X}_n - \mu}{\hat{\sigma}/\sqrt{n}}$$

mit Schätzwert $\hat{\sigma}$ für σ ist nicht mehr normal-verteilt (wie für ein bekanntes, festes σ), sondern folgt einer t-Verteilung mit 9 Freiheitsgraden.

Das 99.5 % Quantil dieser Verteilung ist bei (zu **R**)

```
from scipy.stats import t

t.ppf(q=0.995, df=9)

## Error in running command /usr/local/bin/python3
```

Somit fallen 99 % der Beobachtungen von

$$\frac{\overline{X}_n - \mu}{\hat{\sigma}/\sqrt{n}}$$

in das Intervall $[-3.25, 3.25]$. Es gilt also

$$P\left(\frac{\overline{X}_n - \mu}{\widehat{\sigma}/\sqrt{n}} \in [-3.25, 3.25]\right) = 0.99$$

Ein Vertrauensintervall für μ ist daher gegeben durch

$$\left[\overline{x}_{10} - 3.25 \cdot \frac{\widehat{\sigma}}{\sqrt{n}}, \overline{x}_{10} + 3.25 \cdot \frac{\widehat{\sigma}}{\sqrt{n}}\right]$$

Für $\widehat{\sigma} = 6$, $n = 10$ und $\overline{x}_{10} = 31$ ergibt sich das Vertrauensintervall

$$[24.8, 37.2]$$

Durch Vergleich mit a) findet man, dass das Vertrauensintervall einen Faktor 3.25/2.58, also um 26 % grösser geworden ist.

Lösung 7.4

a) $\left[-403 \pm t_{9-1;97.5\,\%} \cdot \frac{3.127}{\sqrt{9}}\right] = [-403 \pm 2.31 \cdot 1.042] = [-405.4, -400.6]$

Mit **Python**: (zu **R**)

```
from scipy.stats import norm, t
import numpy as np

t.interval(alpha=0.95, df=8, loc=-403, scale=3.127/np.sqrt(9))

## Error in running command /usr/local/bin/python3
```

b) Da -400.0 nicht im 95 %-Vertrauensintervall liegt, würde die Nullhypothese $H_0 : \mu = -400.0$ zu Gunsten der Alternative $H_A : \mu \neq -400.0$ auf dem 5 %-Signifikanzniveau verworfen werden.

Die Beobachtungen und die Hypothese $H_0 : \mu = -400.0$ passen also nicht gut zusammen und daher ist die wahre Differenz wohl nicht -400.0.

Lösung 7.5

a) Wir wählen

$$H_0 : \quad \mu = 500$$

als Nullhypothese und

$$H_A : \quad \mu > 500$$

als Alternativhypothese.

Man möchte auf jeden Fall den Fehler „Es handelt sich um 500er Schrauben, doch wir denken, es seien die neuen verbesserten Schrauben." vermeiden, da dies im schlimmsten Fall zum Brückeneinsturz führen kann. Daher wird dies der Fehler 1. Art.

Die Geschichte schliesst $\mu < 500$ als Teil der Alternative aus. Die Alternative $H_A : \mu > 500$ führt zu einer grösseren Macht als $H_A : \mu \neq 500$ (eines der beiden „Argumente" reicht).

b) Mit **Python** (zu **R**)

```
import numpy as np
from scipy.stats import norm, t
from pandas import Series

x = Series([520, 512, 499, 524, 505])

x.mean()
x.var()

t.ppf(q=0.95, df=x.size-1, loc=500, scale=x.std()/np.sqrt(x.size))


## Error in running command /usr/local/bin/python3
```

Der Verwerfungsbereich ist also

$$I = [509.8, \infty)$$

Der Wert 512 für den Mittelwert liegt im Verwerfungsbereich, somit wird die Nullhypothese verworfen. Die Schrauben sind also statistisch signifikant stärker.

Berechnung von Hand:

Die Teststatistik berechnet sich als

$$T = \sqrt{5}\frac{\overline{X}_n - 500}{\hat{\sigma}}$$

wobei $\overline{x}_5 = \frac{1}{5}\sum_{i=1}^{5} x_i = 512$ der empirische Mittelwert und

$$\hat{\sigma}^2 = \frac{1}{4}\sum_{i=1}^{5}(x_i - \bar{x}_5)^2 = 106.5$$

die empirische Varianz ist.

Aus unserem Datensatz ergibt sich $t = \frac{512-500}{\sqrt{106.5/5}} \approx 2.6$ T ist t-verteilt mit vier Freiheitsgraden. Der Verwerfungsbereich auf dem 5 % Niveau ist gegeben durch

(zu **R**)

$$K = [2.132, \infty)$$

```
from scipy.stats import t

t.ppf(q=0.95, df=4)

## Error in running command /usr/local/bin/python3
```

Die Nullhypothese wird demnach abgelehnt.

c) Das zweiseitige Vertrauensintervall ist gegeben durch (zu **R**)

$$\text{VI} := \left[\overline{x}_5 - \frac{\hat{\sigma} t_{n-1;1-\alpha/2}}{\sqrt{5}}; \overline{x}_5 + \frac{\hat{\sigma} t_{n-1;1-\alpha/2}}{\sqrt{5}}\right]$$

```
import numpy as np
from scipy.stats import norm, t
from pandas import Series

x = Series([520, 512, 499, 524, 505])

t.ppf(q=[0.025,0.975], df=x.size-1, loc=x.mean(),
scale=x.std()/np.sqrt(x.size))

## Error in running command /usr/local/bin/python3
```

Für obige Daten ergibt sich $VI \approx [499.1882; 524.8118]$.

d) Setzt man die Streuung σ als bekannt voraus, so ist das zweiseitige Vertrauensintervall gegeben durch

$$\text{VI} := \left[\hat{\mu} - \frac{\sigma z_{1-\alpha/2}}{\sqrt{5}}; \hat{\mu} + \frac{\sigma z_{1-\alpha/2}}{\sqrt{5}}\right]$$

wobei $z_{1-\alpha/2}$ das $1-\alpha$-Quantil einer Standard-Normalverteilung ist. (zu **R**)

```
import numpy as np
from scipy.stats import norm, t
from pandas import Series

x = Series([520, 512, 499, 524, 505])

norm.ppf(q=[0.025,0.975], loc=x.mean(), scale=x.std()/np.sqrt(x.size))
```

```
## Error in running command /usr/local/bin/python3
```

Für obige Daten ergibt sich dann VI $\approx [502.9544; 521.0456]$.

e) R,R,R,R,R

R-Code

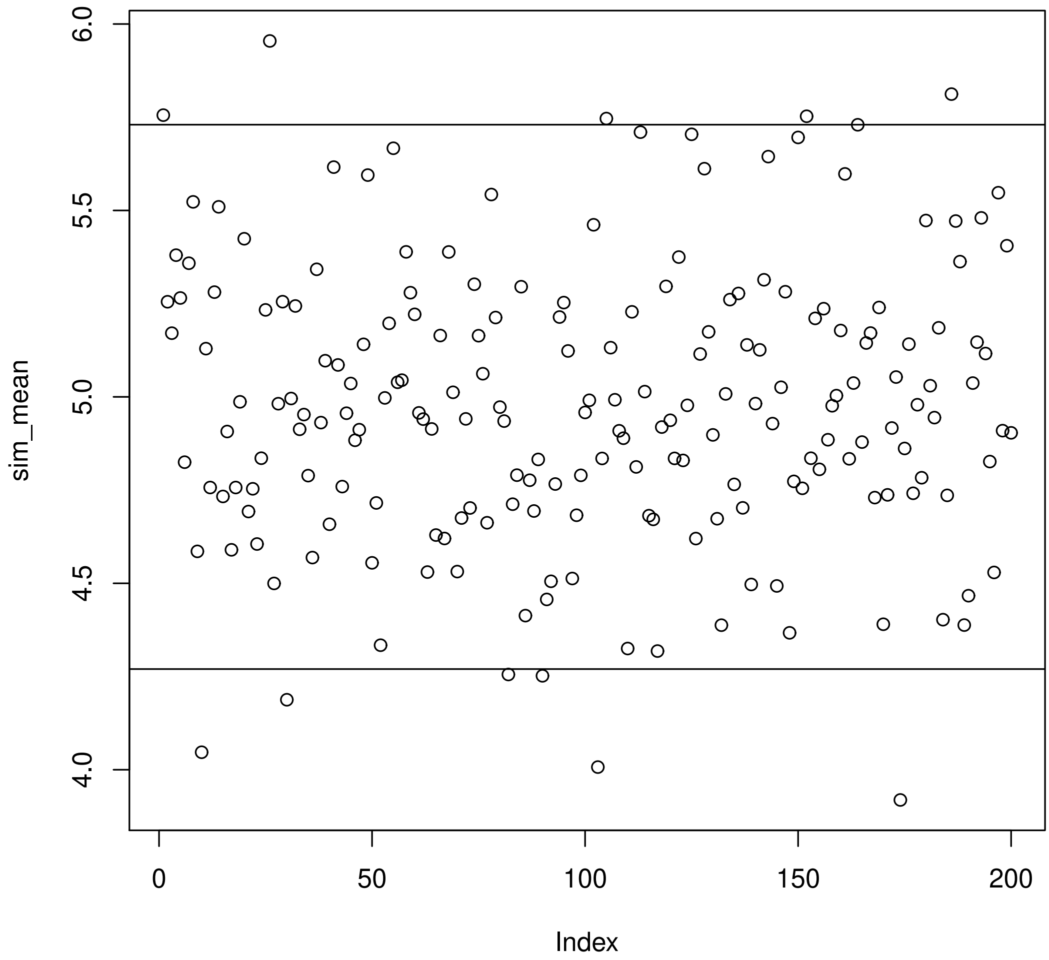
Aufgabe 7.2

a)

b)

c) (zu **Python**)

```
n <- 60
sim <- matrix(runif(n*200, min=0, max=10), ncol=n)
sim_mean <- apply(sim, 1, "mean")
plot(sim_mean)
abline(h=5.73)
abline(h=4.27)
```

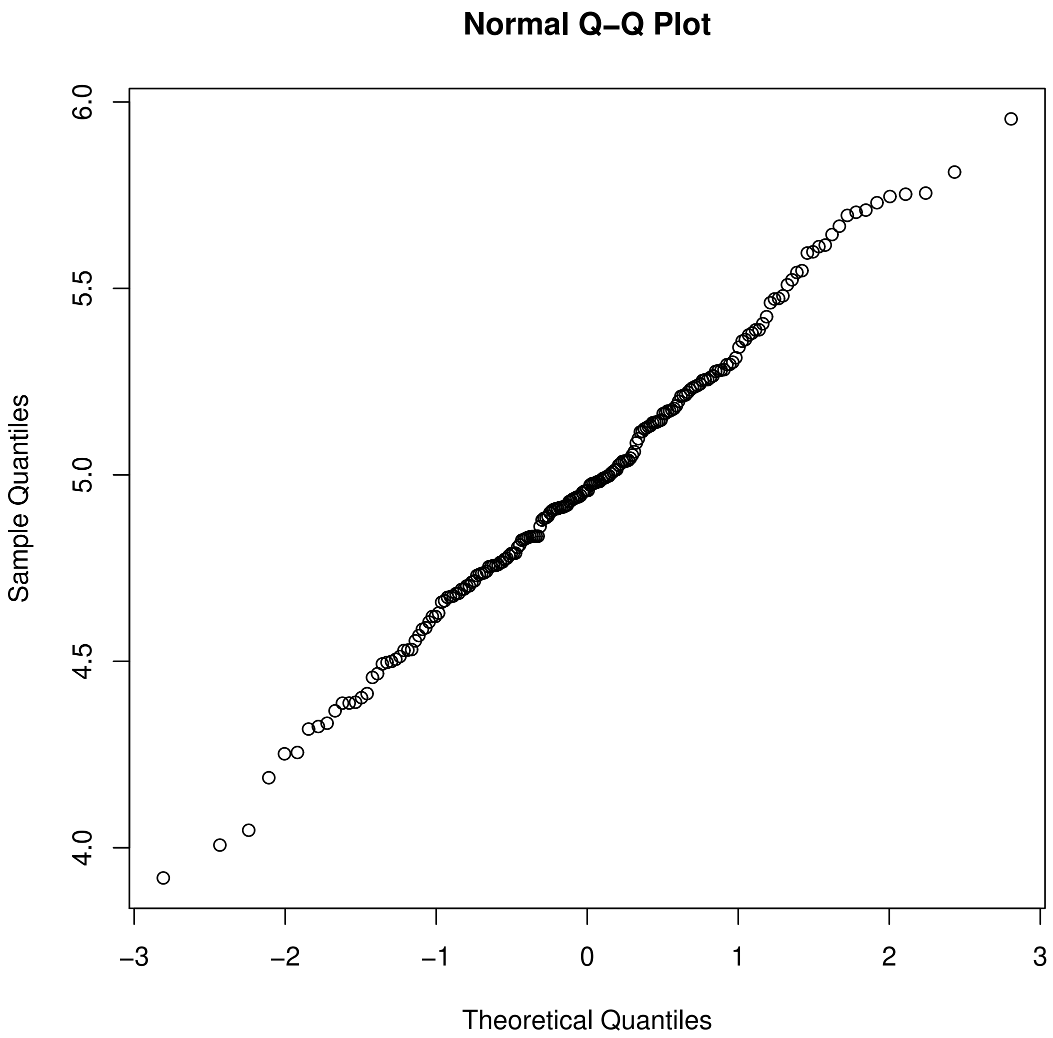



```
d <- sum(sim_mean>5.73) + sum(sim_mean<4.27)
```

In unserem Beispiel sind es 11.

Weiter bestätigt sich auch der Zentrale Grenzwertsatz: (zu **Python**)

```
qqnorm(sim_mean)
```



Aufgabe 7.3

a) (zu **Python**)

```
qnorm(c(0.005,0.995), 31, 6/sqrt(10))

## [1] 26.11271 35.88729
```

b)

c) (zu **Python**)

Diese Option gibt es bei **R** nicht.

(zu **Python**)

```r
qt(0.995, 9)

## [1] 3.249836
```

Aufgabe 7.4

a) (zu **Python**)

Dieser Befehl existiert in **R** nicht.

Aufgabe 7.5

a)

b) (zu **Python**)

```r
x <- c(520, 512, 499, 524, 505)

mean(x)

## [1] 512

var(x)

## [1] 106.5

length(x)-1

## [1] 4

qt(0.95, df=length(x)-1)*(sd(x)/sqrt(length(x)))+500

## [1] 509.8389
```

(zu **Python**)

```r
qt(0.95, df=4)

## [1] 2.131847
```

c) (zu **Python**)

```r
x <- c(520, 512, 499, 524, 505)

qt(c(0.025,0.975), df=length(x)-1)*(sd(x)/sqrt(length(x)))+512

## [1] 499.1862 524.8138
```

d) (zu **Python**)

```
x <- c(520, 512, 499, 524, 505)

qnorm(c(0.025,0.975), mean=mean(x), sd=sd(x)/sqrt(length(x)))

## [1] 502.9544 521.0456
```