Musterlösungen zu Serie 6

Lösung 6.1

X_i = Inhalt (in Zentiliter) der i-ten Weinflasche, $i = 1, \ldots, n = 12$.

1. *Modell*: $X_1, \ldots, X_{12}$ i.i.d. $\sim \mathcal{N}(\mu, \sigma^2)$, $\sigma^2 = 1.5^2$ bekannt.
2. *Nullhypothese*:
$$H_0: \quad \mu = \mu_0 = 70$$
Alternative:
$$H_A: \quad \mu < \mu_0$$
3. *Verteilung der Teststatistik unter H_0*:
$$T: \quad \overline{X}_{12} \sim \mathcal{N}\left(70, \frac{1.5^2}{12}\right)$$
4. *Signifikanzniveau*:
$$\alpha = 5\,\%$$
5. *Verwerfungsbereich für die Teststatistik*: (zu **R**)
```
from scipy.stats import norm
import numpy as np
norm.ppf(q=0.05, loc=70, scale=1.5/np.sqrt(12))

## 69.28775748677653
```
$$K = (-\infty, 69.29]$$
6. *Testentscheid*: (zu **R**)
```
from pandas import Series

wein = Series([71, 69, 67, 68, 73, 72, 71, 71, 68, 72, 69, 72])

wein.mean()

## 70.25
```
$70.25 \notin K \rightarrow H_0$ beibehalten. Es ist also durchaus plausibel, dass der Weinhändler den Wein korrekt abfüllt.

L<U+FFFD><U+FFFD>sung mittels Standardisierung:

1. *Modell*: $X_1, \dots, X_{12}$ i.i.d. $\sim \mathcal{N}(\mu, \sigma^2)$, $\sigma^2 = 1.5^2$ bekannt.
2. *Nullhypothese*: $H_0 : \mu = \mu_0 = 70$ *Alternative*: $H_A : \mu < \mu_0$
3. *Teststatistik*:
$$Z = \sqrt{n}\frac{\overline{X}_n - \mu_0}{\sigma}$$
Verteilung der Teststatistik unter $H_0 : \ Z \sim \mathcal{N}(0, 1)$
4. *Signifikanzniveau*: $\alpha = 5\,\%$
5. *Verwerfungsbereich für die Teststatistik*: (zu **R**)
$$\Phi^{-1}(0.05) = -1.645 \Rightarrow K = (-\infty, -1.645]$$

```
from scipy.stats import norm
norm.ppf(q=0.05)

## -1.6448536269514729
```

6. *Testentscheid*:
$$z = \sqrt{12}\frac{70.25 - 70}{1.5} = 0.5774$$
$z \notin K \to H_0$ beibehalten. Es ist also durchaus plausibel, dass der Weinhändler den Wein korrekt abfüllt.

Lösung 6.2

1. *Modell*: $X_1, \dots, X_{12}$ i.i.d. $\sim \mathcal{N}(\mu, \sigma^2)$, σ^2 unbekannt; geschätzter Wert: $\hat{\sigma}_x^2 = 1.96^2$ (zu **R**)

```
from pandas import Series

wein = Series([71, 69, 67, 68, 73, 72, 71, 71, 68, 72, 69, 72])

wein.std()

## 1.9598237397554634
```

2. *Nullhypothese*: $H_0 : \mu = \mu_0 = 70$ *Alternative*: $H_A : \mu < \mu_0$
3. *Teststatistik*: t-Verteilung mit Freiheitsgrad 11
4. *Signifikanzniveau*: $\alpha = 5\,\%$
5. *Verwerfungsbereich für die Teststatistik*:

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

t.ppf(q=0.05, df=11, loc=70, scale=1.96/np.sqrt(12))

## 68.98388250815812
```

$$K = (-\infty, 68.98]$$

6. *Testentscheid*: (zu **R**)

```
from pandas import Series

wein = Series([71, 69, 67, 68, 73, 72, 71, 71, 68, 72, 69, 72])

wein.mean()

## 70.25
```

   $70.25 \notin K \rightarrow H_0$ beibehalten. Wir kommen also zum selben Ergebnis wie in Teilaufgabe a).

L<U+FFFD><U+FFFD>sung mittels Standardisierung:

1. *Modell*: $X_1, \ldots, X_{12}$ i.i.d. $\sim \mathcal{N}(\mu, \sigma^2)$, σ^2 unbekannt; geschätzter Wert: $\hat{\sigma}_x^2 = 1.96^2$
2. *Nullhypothese*: $H_0 : \mu = \mu_0 = 70$ *Alternative*: $H_A : \mu < \mu_0$
3. *Teststatistik*:
$$T = \sqrt{n}\frac{\overline{X}_n - \mu_0}{\hat{\sigma}_X}$$
   Verteilung der Teststatistik unter $H_0 : T \sim t_{n-1}$
4. *Signifikanzniveau*: $\alpha = 5\,\%$
5. *Verwerfungsbereich für die Teststatistik*: (zu **R**)
$$t_{11;0.05} = -1.796 \Rightarrow K = (-\infty, -1.796]$$

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

t.ppf(q=0.05, df=11)

## -1.7958848187036696
```

6. *Testentscheid*:

$$t = \sqrt{12}\frac{70.25 - 70}{1.96} = 0.441$$

$t \notin K \rightarrow H_0$ beibehalten. Wir kommen also zum selben Ergebnis wie in Teilaufgabe a).

Lösung 6.3

a) 1. *Modell*: X_i: i-te Ammoniumbestimmung, X_i i.i.d. $\mathcal{N}(\mu, \sigma^2)$ mit $\sigma = 10$.

2. *Nullhypothese*:

$$H_0: \quad \mu_0 = 200$$

Alternative:

$$H_A: \quad \mu > 200$$

(einseitiger Test nach oben)

3. *Verteilung der Teststatistik unter H_0*:

$$\overline{X}_{16} \sim \mathcal{N}\left(200, \frac{10^2}{16}\right)$$

4. *Signifikanzniveau*:

$$\alpha = 0.05$$

5. *Verwerfungsbereich für die Teststatistik*: (zu **R**)

$$K = [204.11, \infty)$$

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

norm.ppf(q=0.95, loc=200, scale=10/np.sqrt(16))

## 204.1121340673787
```

6. *Testentscheid*: Es gilt

$$204.2 \in K$$

also wird die Nullhypothese verworfen. Eine Grenzwertüberschreitung ist statistisch gesichert.

L<U+FFFD><U+FFFD>sung mittels Standardisierung:

1. *Modell*: X_i: i-te Ammoniumbestimmung, X_i i.i.d. $\mathcal{N}(\mu, \sigma^2)$ mit $\sigma = 10$.

2. *Nullhypothese*: $H_0 : X_i$ i.i.d. $\mathcal{N}(\mu_0, \sigma^2)$ mit $\mu_0 = 200$ und $\sigma = 10$

   Alternative: $H_A : X_i$ i.i.d. $\mathcal{N}(\mu, \sigma^2)$ mit $\mu > 200$ und $\sigma = 10$ (einseitig)

3. *Teststatistik*:

$$Z = \frac{\overline{X}_n - \mu_0}{\sigma / \sqrt{n}}$$

   Verteilung der Teststatistik unter H_0: $Z \sim \mathcal{N}(0, 1)$

4. *Signifikanzniveau*: $\alpha = 0.05$

5. *Verwerfungsbereich für die Teststatistik*: (zu **R**)

$$K = \{z : \Phi(z) > 0.95\} =]1.64, \infty[$$

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

norm.ppf(q=0.95)

## 1.6448536269514722
```

6. *Testentscheid*: Der Wert der Teststatistik ist

$$z = \frac{\overline{x}_n - \mu_0}{\sigma / \sqrt{n}} = \frac{204.2 - 200}{\sigma / \sqrt{16}} = 1.68$$

   $1.68 \in K$, also wird die Nullhypothese verworfen. Eine Grenzwertüberschreitung ist statistisch gesichert.

b) Aus der Teilaufgabe a) folgt, dass die Nullhypothese verworfen werden kann, falls der Mittelwert aller Messungen grösser als 204.11 ist,

$$\overline{x}_n > 204.11$$

Es gilt f<U+FFFD><U+FFFD>r die Wahrscheinlichkeit (zu **R**)

$$P(\overline{X}_n > 204.11)$$

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

1-norm.cdf(x=204.11, loc=205, scale=10/np.sqrt(16))
```

```
## 0.6390797174095532
```

Die Macht des Tests ist also rund 64 %.

L<U+FFFD><U+FFFD>sung mittels Standardisieren:

Um die Wahrscheinlichkeit zu berechnen, dass eine Grenzwertüberschreitung nachgewiesen werden kann (H_0 verworfen werden kann), geht man wieder zu einer standardisierten Zufallsvariablen über. Mit $\mu_A = 205$ und $\sigma = 10$ erhält man

$$
\begin{aligned}
P(\overline{X}_n > 204.1) &= P\left(\frac{\overline{X}_n - \mu_A}{\sigma/\sqrt{n}} > \frac{204.1 - \mu_A}{\sigma/\sqrt{n}}\right) \\
&= P\left(\frac{\overline{X}_n - \mu_A}{\sigma/\sqrt{n}} > -0.36\right) \\
&= P(Z > -0.36)
\end{aligned}
$$

Dies entspricht also der Wahrscheinlichkeit, dass eine normal-verteilte Zufallsvariable Z mit Varianz 1,

$$Z \sim \mathcal{N}(0,1)$$

einen Wert grösser als -0.36 annimmt. Diese Wahrscheinlichkeit ist wegen der Symmetrie der Normalverteilung gleich zu (zu **R**)

$$P(Z \leq 0.36) = 0.6406$$

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

norm.cdf(x=0.36)

## 0.6405764332179913
```

c) Dies ist genau das Niveau des Tests und war als 5 % vorgegeben.

d) Es ist schwieriger, eine Grenzüberschreitung nachzuweisen, wenn die Standardabweichung aus den Daten geschätzt wird. Die Verteilung der Teststatistik folgt einer t-Verteilung mit $n-1$ Freiheitsgraden, die langsch<U+FFFD><U+FFFD>nziger als eine Normalverteilung ist. Der t-Test wird folgendermassen formal durchgeführt:

1. *Modell*: X_i: i-te Ammoniumbestimmung

   X_i i.i.d. $\mathcal{N}(\mu, \sigma^2)$ mit σ unbekannt

2. *Nullhypothese*:
$$H_0: \quad \mu_0 = 200$$
Alternative:
$$H_A: \quad \mu > 200$$
(einseitig nach oben)

3. *Teststatistik*: t-Verteilung mit 15 Freiheitsgraden

4. *Signifikanzniveau*:
$$\alpha = 0.05$$

5. *Verwerfungsbereich für die Teststatistik*: (zu **R**)
$$K = (204.38, \infty)$$

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

t.ppf(q=0.95, df=15, loc=200, scale=10/np.sqrt(16))

## 204.38262588923138
```

6. *Testentscheid*:
$$204.2 \notin K$$
also kann die Nullhypothese nicht verworfen werden. Eine Grenzwertüberschreitung ist statistisch nicht gesichert.

Der Unterschied zum z-Test ist nicht sehr gross, führt hier aber gerade dazu, dass die Nullhypothese nicht mehr verworfen werden kann.

L<U+FFFD><U+FFFD>sung mittels Standardisierung:

1. *Modell*:

   X_i: i-te Ammoniumbestimmung, X_i i.i.d. $\mathcal{N}(\mu, \sigma^2)$ mit σ unbekannt

2. *Nullhypothese*:
$$H_0: \quad \mu_0 = 200$$
Alternative:
$$H_A: \quad \mu > 200$$
(einseitig nach oben)

3. *Teststatistik*:

$$T = \frac{\overline{X}_n - \mu_0}{\widehat{\sigma}/\sqrt{n}}$$

Verteilung der Teststatistik unter H_0:

$$T \sim t_{n-1}$$

4. *Signifikanzniveau*:

$$\alpha = 0.05$$

5. *Verwerfungsbereich für die Teststatistik*: (zu **R**)

$$K = \{t : t_{15,0.95} > 0.95\} = (1.753, \infty)$$

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

t.ppf(q=0.95, df=15)

## 1.7530503556925547
```

6. *Testentscheid*: Der Wert der Teststatistik ist

$$t = \frac{\widehat{\overline{x}}_n - \mu_0}{\widehat{\sigma}/\sqrt{n}} = \frac{204.2 - 200}{\widehat{\sigma}/\sqrt{16}} = 1.68$$

$1.68 \notin K$, also kann die Nullhypothese nicht verworfen werden. Eine Grenzwertüberschreitung ist statistisch nicht gesichert.

Der Unterschied zum z-Test ist nicht sehr gross, führt hier aber gerade dazu, dass die Nullhypothese nicht mehr verworfen werden kann.

R-Code

Bemerkung: F<U+FFFD><U+FFFD>r den t-Test *muss* in **R** standardisiert werden!

Aufgabe 6.1

(zu **Python**)

```
qnorm(0.05, 70, 1.5/sqrt(12))

## [1] 69.28776
```

(zu **Python**)

```
wein <- c(71, 69, 67, 68, 73, 72, 71, 71, 68, 72, 69,
    72)

mean(wein)

## [1] 70.25
```

(zu **Python**)

```
qnorm(0.05)

## [1] -1.644854
```

Aufgabe 6.2

(zu **Python**)

```
wein <- c(71, 69, 67, 68, 73, 72, 71, 71, 68, 72, 69,
    72)

sd(wein)

## [1] 1.959824
```

(zu **Python**)

```
wein <- c(71, 69, 67, 68, 73, 72, 71, 71, 68, 72, 69,
    72)

mean(wein)

## [1] 70.25
```

(zu **Python**)

```
qt(0.05, df = 11)

## [1] -1.795885
```

Aufgabe 6.3

a) (zu **Python**)

```
qnorm(0.95, mean = 200, sd = 10/sqrt(16))

## [1] 204.1121
```

```
qnorm(0.95)

## [1] 1.644854
```

(zu **Python**)

b) (zu **Python**)

```
1 - pnorm(204.11, mean = 205, sd = 10/sqrt(16))

## [1] 0.6390797
```

(zu **Python**)

```
pnorm(0.36)

## [1] 0.6405764
```

c)

d) (zu **Python**)

```r
qt(0.95, 15) * 10/4 + 200

## [1] 204.3826
```

(zu **Python**)

```r
qt(0.95, 15)

## [1] 1.75305
```