Serie 6

Aufgabe 6.1

Ein Weinhändler behauptet, dass die von ihm gefüllten Weinflaschen 70 Zentiliter enthalten. Ein skeptischer Konsument vermutet aber, dass der Weinhändler zu wenig Wein abfüllt und will diese Behauptung überprüfen. Deshalb kauft er 12 Weinflaschen und misst ihren Inhalt. Er findet:

$$71, 69, 67, 68, 73, 72, 71, 71, 68, 72, 69, 72 \quad \text{(in Zentiliter)}$$

Nehmen Sie zunächst an, dass die Standardabweichung der Abfüllung im Voraus bekannt ist. Sie beträgt $\sigma = 1.5$ Zentiliter. Da die Standardabweichung der Messungen bekannt ist, können wir einen z-Test durchführen. Führen Sie den (einseitigen; in welche Richtung?) Test auf dem 5 %- Signifikanzniveau durch.

Geben Sie die Modellannahmen, H_0, H_A, den Verwerfungsbereich, den Wert der Teststatistik und das Testergebnis explizit an.

Formulieren Sie in einem Satz die Schlussfolgerung für den kritischen Konsumenten.

Aufgabe 6.2

Ein Weinhändler behauptet, dass die von ihm gefüllten Weinflaschen 70 Zentiliter enthalten. Ein skeptischer Konsument vermutet aber, dass der Weinhändler zu wenig Wein abfüllt und will diese Behauptung überprüfen. Deshalb kauft er 12 Weinflaschen und misst ihren Inhalt. Er findet:

$$71, 69, 67, 68, 73, 72, 71, 71, 68, 72, 69, 72 \quad \text{(in Zentiliter)}$$

Die Standardabweichung der Abfüllungen ist nicht bekannt. Man muss sie also aus den gemachten Stichproben schätzen:

$$\hat{\sigma}_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 \approx 1.96^2$$

Da die Standardabweichung geschätzt wurde und nicht mehr exakt bekannt ist, kann der z-Test nicht durchgeführt werden. Verwenden Sie nun den t-Test auf dem 5 %-Signifikanzniveau.

Geben Sie die Modellannahmen, H_0, H_A, den Verwerfungsbereich, den Wert der Teststatistik und das Testergebnis explizit an.

Was ändert sich an obigem Test?

Aufgabe 6.3

Unterhalb einer Kläranlage wurden 16 unabhängige Wasserproben aus einem Fluss entnommen und jeweils deren Ammoniumkonzentration X_i (in µg NH_4-N/l) mit einem Messgerät bestimmt. Der Mittelwert der Proben ergab $\overline{x}_{16} = 204.2$.

Wir wollen nun wissen, ob mit diesem Experiment eine Überschreitung des Grenzwerts von 200 µg NH_4-N/l nachgewiesen werden kann (auf dem 5 % Niveau).

a) Nehmen Sie an, die Standardabweichung der Messungen sei im Voraus aufgrund früherer Studien bekannt. Sie betrage 10 µg NH_4-N/l.

   Führen Sie unter dieser Annahme einen z-Test durch, um zu prüfen, ob eine Grenzwertüberschreitung nachgewiesen werden kann.

   Geben Sie die Modellannahmen, H_0, H_A, den Verwerfungsbereich, den Wert der Teststatistik und das Testergebnis explizit an.

b) Wie wahrscheinlich ist es, dass man mit 16 unabhängigen Wasserproben eine Grenzwertüberschreitung nachweisen kann, wenn die wahre Ammoniumkonzentration tatsächlich über dem Grenzwert, und zwar bei 205 µg NH_4-N/l liegt?

c) Wie wahrscheinlich ist es, dass man mit 16 unabhängigen Wasserproben fälschlicherweise eine Grenzwertüberschreitung nachweist, obwohl die wahre Ammoniumkonzentration bei 200 µg NH-4-N/l liegt und den Grenzwert somit genau einhält?

d) Nehmen Sie an, dass die Standardabweichung von 10 µg/l aus den 16 Proben geschätzt worden ist. Deshalb ist nun ein t-Test (Nullhypothese $\mu_0 = 200$ µg/l) und nicht ein z-Test angebracht. Führen Sie den t-Test durch.

e) Welche Annahmen des t-Testes könnte verletzt sein und dazu führen, dass der t-Test schlechte Macht hat?

Kurzlösungen vereinzelter Aufgaben

A 6.3:

a) $K = (204.11, \infty)$ oder standardisiert $K = (1.64, \infty)$

b) $P(\overline{X}_n > 204.1) = 0.6406$

d) $K = (204.38, \infty)$

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

print(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

print(wein.mean())
```

```
## 70.25
```

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

Lösung mittels Standardisierung:

1. *Modell*: $X_1, \ldots, X_{12}$ i.i.d. $\sim \mathcal{N}(\mu, \sigma^2)$, $\sigma^2 = 1.5^2$ bekannt.
2. *Nullhypothese*: $H_0 : \mu = \mu_0 = 70$ *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
print(norm.ppf(q=0.05))
## -1.6448536269514729
```

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

Lösung 6.2

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

```
from pandas import Series

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

wein.std()
```

```
## 1.9598237397554634
print(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
print(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
print(wein.mean())
## 70.25
```

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

Lösung 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

print(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**)

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

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

norm.ppf(q=0.95, loc=200, scale=10/np.sqrt(16))
## 204.1121340673787
print(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.

Lösung 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**)

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

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

norm.ppf(q=0.95)
## 1.6448536269514722
```

```
print(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 für 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
print(1-norm.cdf(x=204.11, loc=205, scale=10/np.sqrt(16)))
## 0.6390797174095532
```

Die Macht des Tests ist also rund 64 %.

Lösung 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

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

$$= P(Z > -0.36)$$

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**)

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

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

norm.cdf(x=0.36)
## 0.6405764332179913
print(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 langschwänziger 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
print(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.

Lösung 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

print(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: Für 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**)

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

mean(wein)

## [1] 70.25
```

(zu **Python**)

```r
qt(0.05,df=11)

## [1] -1.795885
```

Aufgabe 6.3

a) (zu **Python**)

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

## [1] 204.1121
```

```r
qnorm(0.95)

## [1] 1.644854
```

(zu **Python**)

b) (zu **Python**)

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

## [1] 0.6390797
```

(zu **Python**)

```r
pnorm(0.36)

## [1] 0.6405764
```

c)

d) (zu **Python**)

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

## [1] 204.3826
```

(zu **Python**)

```
qt(0.95,15)

## [1] 1.75305
```