Serie 9

Aufgabe 9.1

Wir verwenden den Datensatz `InsectSprays.csv`:

```
import pandas as pd
import numpy as np

df = pd.read_csv("../../../Themen/Varianzanalyse/Jupyter_Notebooks_de/InsectSpra
index_col=0)

df.head()
```

```
##    count spray
## 1     10     A
## 2      7     A
## 3     20     A
## 4     14     A
## 5     14     A
```

Dabei wurden 6 verschiedene Insektensprays verwendet, die auf verschiedenen Feldern versprüht wurden. Danach wurde die Anzahl Insekten gezählt, die sich auf dem entsprechenden Feld nach dem Besprühen befanden. (Beall, G., (1942) The Transformation of data from entomological field experiments, Biometrika, 29, 243–262.)

a) Wir wollen zunächst die Mittelwerte der einzelnen Sprays bestimmen. Dazu verwenden wir den **Python**-Methode **`.groupby(...)`**

```
df.groupby("spray").mean()
```

```
##            count
## spray
## A      14.500000
## B      15.333333
## C       2.083333
## D       4.916667
## E       3.500000
## F      16.666667
```

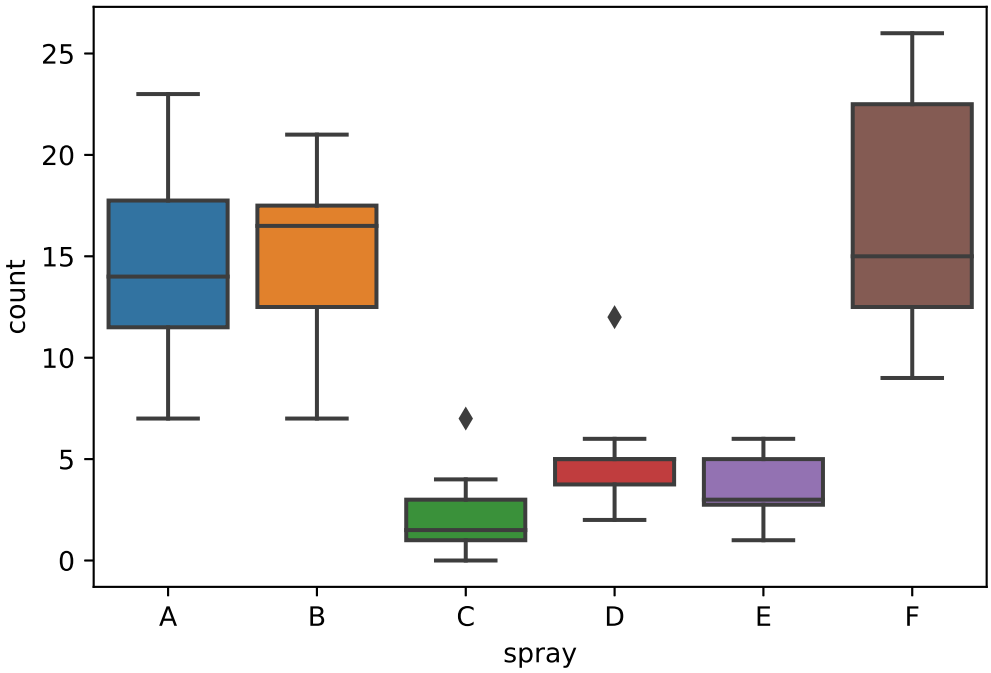
Diese Methode **`.groupby(...)`** gruppiert das Dataframe nach der Spalte **`spray`**.

Die Mittelwerte sind sehr unterschiedlich. Die Sprays *C*, *D* und *E* scheinen wesentlich effizienter zu sein als die Sprays *A*, *B* und *F*. Die tieferen Werte zeigen hier weniger Insekten an.

b) Wir wollen nun noch einen Boxplot der Daten machen.

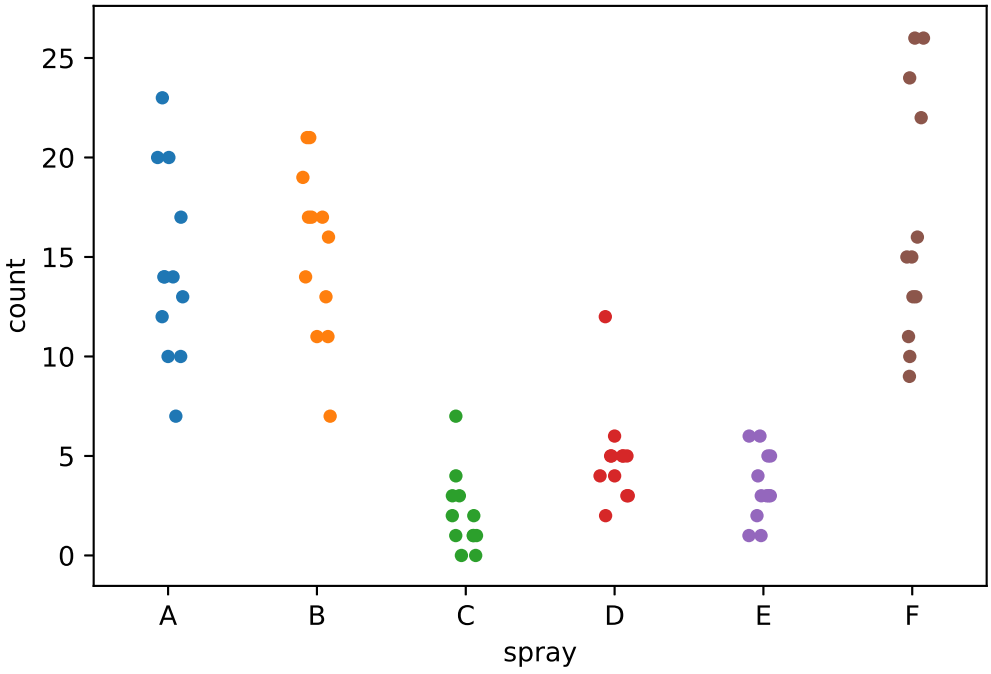
Das Module **seaborn** wurde speziell zu Darstellung von statistischen Daten entwickelt.

```
import seaborn as sns

sns.boxplot(x="spray", y="count", data=df)
```



Auch hier ist offensichtlich, dass die Sprays *C*, *D* und *E* wesentlich effizienter erscheinen zu sein als die Sprays *A* und *E*.

```
sns.stripplot(x="spray", y="count", data=df)
```



Auch hier wieder dasselbe Resultat.

Wir wollen nun noch mit einem Hypothesentest untersuchen, ob die Unterschiede statistisch signifikant sind.

c) Die Nullhypothese ist, dass alle Mittelwerte gleich sind. Wir testen auf Signifikanzniveau von 5 %.

```
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm

fit = ols("count~spray", data=df).fit()
fit.summary()

## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                  count   R-squared:                       0.724
## Model:                            OLS   Adj. R-squared:                  0.704
## Method:                 Least Squares   F-statistic:                     34.70
## Date:                Tue, 21 Apr 2020   Prob (F-statistic):           3.18e-17
## Time:                        05:25:23   Log-Likelihood:                -197.42
## No. Observations:                  72   AIC:                             406.8
## Df Residuals:                      66   BIC:                             420.5
## Df Model:                           5
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
```

```
## Intercept     14.5000      1.132     12.807      0.000      12.240      16.760
## spray[T.B]     0.8333      1.601      0.520      0.604      -2.363       4.030
## spray[T.C]   -12.4167      1.601     -7.755      0.000     -15.613      -9.220
## spray[T.D]    -9.5833      1.601     -5.985      0.000     -12.780      -6.387
## spray[T.E]   -11.0000      1.601     -6.870      0.000     -14.197      -7.803
## spray[T.F]     2.1667      1.601      1.353      0.181      -1.030       5.363
## ==============================================================================
## Omnibus:                        3.201   Durbin-Watson:                   1.753
## Prob(Omnibus):                  0.202   Jarque-Bera (JB):                2.421
## Skew:                           0.411   Prob(JB):                        0.298
## Kurtosis:                       3.360   Cond. No.                         6.85
## ==============================================================================
##
## Warnings:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

Der Output ist sehr ausführlich. Hier interessieren wir uns nur für den p-Wert der F-Statistik.

```
fit.f_pvalue
```

```
## 3.182583726145126e-17
```

Der p-Wert ist kleiner als $3.2 \cdot 10^{-17}$, also bei weitem unter dem Signifikanzniveau. Damit wird die Nullhypothese verworfen. Es gibt also Unterschiede in der Wirksamkeit der Sprays. Dies ist aber nach den Überlegungen von a) und b) nicht sonderlich überraschend.

d) Wir können nun noch untersuchen, ob einer der drei Sprays C, D und E statistisch signifikant unterschiedlich zu den beiden anderen ist.

Dazu erstellen wir eine neues Dataframe, das nur noch die Daten der Sprays C, D und E enthält.

Dazu müssen wir zuerst entscheiden, in welchen Zeilen in der zweiten Spalte ein C, D oder E vorkommt. Dies geschieht mit folgendem Befehl (es gibt mehrere Varianten)

```
df_n = df.loc[df['spray'].isin(["C", "D", "E"])]
df_n.head()
```

```
##     count spray
## 25      0     C
## 26      1     C
## 27      7     C
## 28      2     C
## 29      3     C
```

Wir wollen diesen Befehl noch kurz untersuchen:

```
df['spray'].isin(["C", "D", "E"])
```

```
## 1     False
## 2     False
## 3     False
## 4     False
## 5     False
##       ...
## 68    False
## 69    False
## 70    False
## 71    False
## 72    False
## Name: spray, Length: 72, dtype: bool
```

Die Methode `.isin(...)` steht für „ist Element von". Falls `spray` einer dieser Werte ist, so wird der Wert `True` ausgegeben, ansonsten `False`. Dieser Vektor hat die Länge 72

```
df['spray'].isin(["C", "D", "E"]).size
```

```
## 72
```

was der Anzahl Zeilen des ursprünglichen Dataframes entspricht

```
df.shape
```

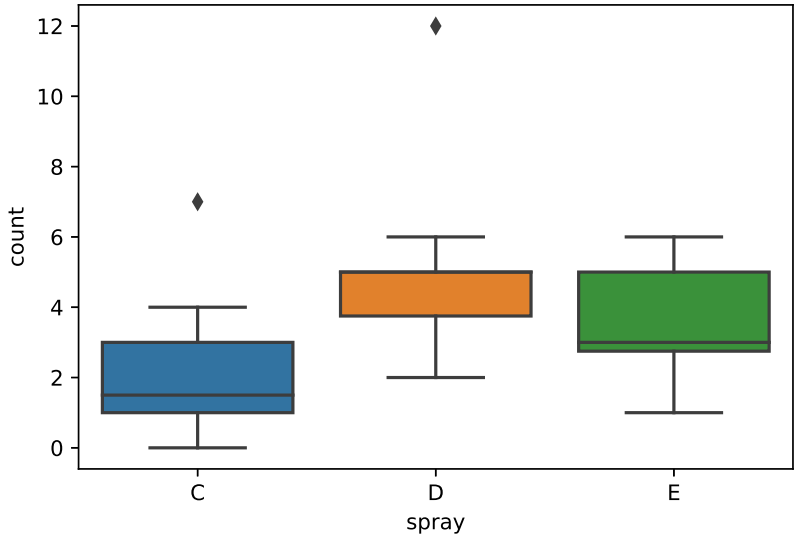
```
## (72, 2)
```

Die Einträge des Vektor sind `False`, falls `spray` den Wert `A`, `B` oder `F`, ansonsten ist der Wert `True`.

Der Befehl `df.loc(...)` wählt nun die Zeilen aus, wo der Eintrag des Vektors `True` ist.

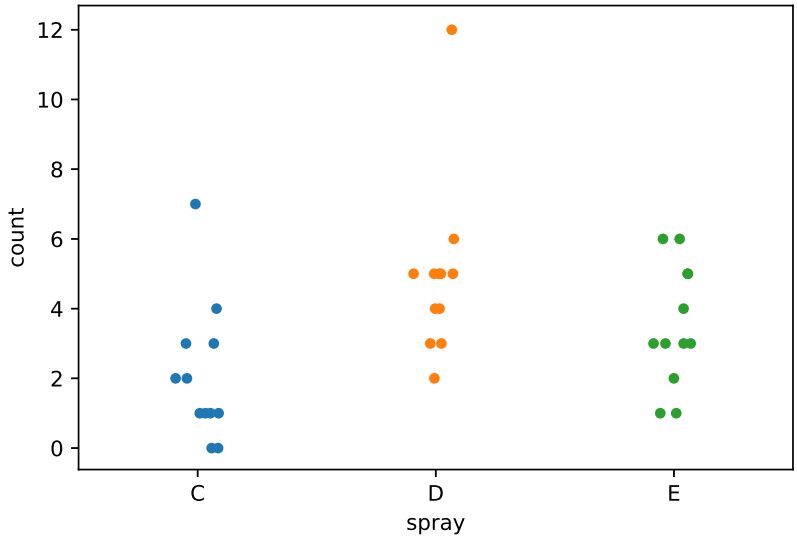
Nun können wir wieder einen Boxplot erstellen.

```
import seaborn as sns

sns.boxplot(x="spray", y="count", data=df_n)
```



oder den Stripchart

```
import seaborn as sns

sns.stripplot(x="spray", y="count", data=df_n)
```



Hier „scheint" Spray *C* besser zu sein. Aber ob dieser Unterschied signifikant ist, müssen wir mit einem Hypothesentest überprüfen.

Nullhypothese ist wieder, dass es keinen Unterschied zwischen den Sprays gibt.

```
fit_n = ols("count~spray", data=df_n).fit()
fit_n.f_pvalue
```

```
## 0.00876264225426873
```

Der p-Wert ist 0.00876 und damit unter dem Signifikanzniveau von 5 %. Somit ist ein Spray unterschiedlich von den beiden anderen. Aus dem Boxplot können wir vermuten, dass der Spray C nach unten abweicht. Dieser Spray wäre dann statistisch signifikanter Testsieger.

Aufgabe 9.2

In der Datei `Diet.csv` sind 76 Personen aufgelistet, die jeweils einer der Diäten 1,2 oder 3 für 6 Wochen machten.

```
import pandas as pd
import numpy as np

df = pd.read_csv("../../../Themen/Varianzanalyse/Jupyter_Notebooks_de/Diet.csv")

df.head()
```

```
##    Person gender  Age  Height  pre.weight  Diet  weight6weeks
## 0      25          41     171          60     2          60.0
## 1      26          32     174         103     2         103.0
## 2       1      0   22     159          58     1          54.2
## 3       2      0   46     192          60     1          54.0
## 4       3      0   55     170          64     1          63.3
```

In der Datei ist das Gewicht **`pre.weight`** vor der Diät und das Gewicht **`weight6weeks`** nach 6 Wochen aufgeführt. Wir interessieren uns für den Gewichtsverlust. Dazu führen wir zu der Datei eine Spalte **`weight.loss`** hinzu. Dies geht folgendermassen:

```
df["weight_loss"] = df["weight6weeks"] - df["pre.weight"]

df.head()
```

```
##    Person gender  Age  Height  pre.weight  Diet  weight6weeks  weight_loss
## 0      25          41     171          60     2          60.0          0.0
## 1      26          32     174         103     2         103.0          0.0
## 2       1      0   22     159          58     1          54.2         -3.8
## 3       2      0   46     192          60     1          54.0         -6.0
## 4       3      0   55     170          64     1          63.3         -0.7
```

Führen Sie nun die Teilaufgaben in Aufgabe 1 für **`weight.loss`** und **`Diet`** durch. Interpretieren Sie jeweils die Resultate.

Aufgabe 9.3

24 Tiere werden zufällig zu 4 unterschiedlichen Ernährungsdiäten zugeordnet, um den Effekt auf die Blutkoagulationszeit zu untersuchen.

Behandlung	Koagulationszeit							
A	62	60	63	59				
B	63	67	71	64	65	66		
C	68	66	71	67	68	68		
D	56	62	60	61	63	64	63	59

a) Geben Sie die Daten selber in **Python** ein, und stellen Sie sie mit Stripcharts und Boxplots dar.

Python-Hinweise: Die Daten werden in ein Dataframe mit zwei Spalten eingelesen: eine Spalte mit Druckfestigkeitsangaben und eine Spalte mit Hüllentyp:

```
from pandas import DataFrame
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st

df=DataFrame({
  "Behandlung": np.repeat(["A", "B", "C", "D"], [4, 6, 6, 8]),
  "Koagulationszeit" : [62, 60, 63, ..]
})
sns.stripplot(x="Behandlung", y="Koagulationszeit", data=df)
plt.xlabel("Behandlung")
plt.ylabel("Koagulationszeit")
plt.show()

sns.boxplot(x="Behandlung", y="Koagulationszeit", data=df)
plt.xlabel("Behandlung")
plt.ylabel("Koagulationszeit")
plt.show()
```

b) Besteht ein signifikanter Unterschied zwischen den Behandlungsarten in Bezug auf die Koagulationszeit? Führen Sie einen statistischen Hypothesentest auf dem 5 % Niveau durch.

Stellen Sie die Nullhypothese auf.

Aufgabe 9.4

Der Fachartikel *Compression of Single-Wall Corrugated Containers Using Fixed and Floating Test Platens* (J. Testing and Evaluation, 1992: 318-320) beschreibt ein Experiment, in dem verschiedene Typen von Container-Hüllen in Bezug auf Druckfestigkeit (lb) verglichen wurden.

Typ	Druckfestigkeit					
1	655.5	788.3	734.3	721.4	679.1	699.4
2	789.2	772.5	786.9	686.1	732.1	774.8
3	737.1	639.0	696.3	671.7	717.2	727.1
4	535.1	628.7	542.4	559.0	586.9	520.0

a) Geben Sie die Daten selber in **Python** ein, und stellen Sie sie mit Stripcharts und Boxplots dar.

Python-Hinweise: Die Daten werden in ein Dataframe mit zwei Spalten eingelesen: eine Spalte mit Druckfestigkeitsangaben und eine Spalte mit Hüllentyp:

```
from pandas import DataFrame
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st

df=DataFrame({
  "Typ": np.repeat(["T1", "T2", "T3", "T4"], [6, 6, 6, 6]),
  "Druckfestigkeit" : [655.5, 788.3, 734.3, ..]
})
sns.stripplot(x="Typ", y="Druckfestigkeit", data=df)
plt.xlabel("Typ")
plt.ylabel("Druckfestigkeit")
plt.show()

sns.boxplot(x="Typ", y="Druckfestigkeit", data=df)
plt.xlabel("Typ")
plt.ylabel("Druckfestigkeit")
plt.show()
```

b) Wie lautet ein Gruppenmittelmodell passend zum Datensatz und zur Fragestellung? Schätzen Sie die Parameter Ihres Modelles.

c) Besteht ein Unterschied zwischen den Hüllentypen? Führen Sie einen statistischen Hypothesentest auf dem 5 % Niveau durch.

Kurzlösungen vereinzelter Aufgaben

A 9.3:

b) $\mu = 64 \quad \mu_A = 61 \quad \mu_B = 66 \quad \mu_C = 68 \quad \mu_D = 61$

c) $s_A^2 = 3.333 \quad s_B^2 = 8 \quad s_C^2 = 2.8 \quad s_D^2 = 6.85$

d) $\text{MS}_E = 5.6$

e) $\text{MS}_G = 76$

A 9.4:

c) P-Wert $5.5e-7$

Musterlösungen zu Serie 9

Lösung 9.1

Lösung 9.2

a) Gruppenmittel:

```
df.groupby("Diet").mean()
```

```
##       Person        Age      Height  pre.weight  weight6weeks  weight_loss
## Diet
## 1       12.5  40.875000  170.291667   72.875000     69.575000    -3.300000
## 2       38.0  39.000000  174.851852   71.111111     68.085185    -3.025926
## 3       65.0  37.777778  167.259259   73.629630     68.481481    -5.148148
```

oder einfacher:

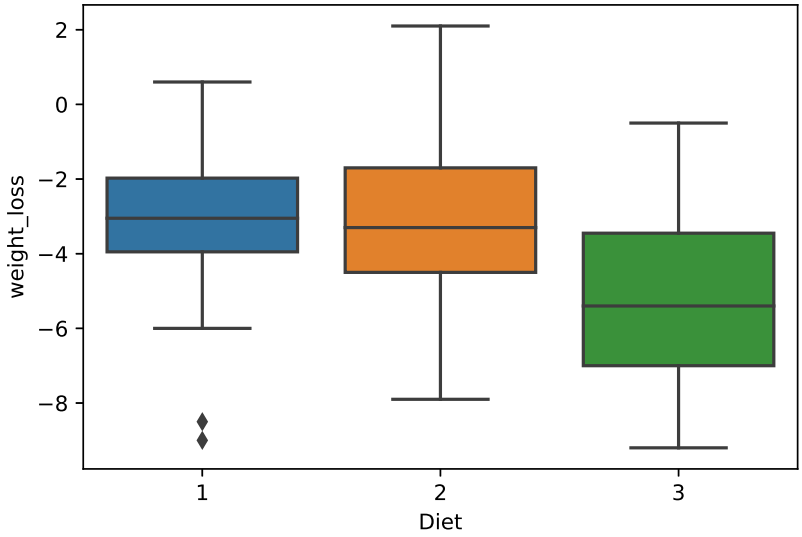
```
df.groupby("Diet")["weight_loss"].mean()
```

```
## Diet
## 1   -3.300000
## 2   -3.025926
## 3   -5.148148
## Name: weight_loss, dtype: float64
```

Die Diäten 1 und 2 führen zu einem durchschnittlichen Gewichtsverlust von etwa 3 Kilogramm. Diät 5 hingegen ist der durchschnittliche Gewichtsverlust 5 Kilogramm. Ist dies nun statistisch signifikant ein grösser Gewichtsverlust als bei Diäten 1 und 2?

b) Graphische Darstellung durch Boxplot:

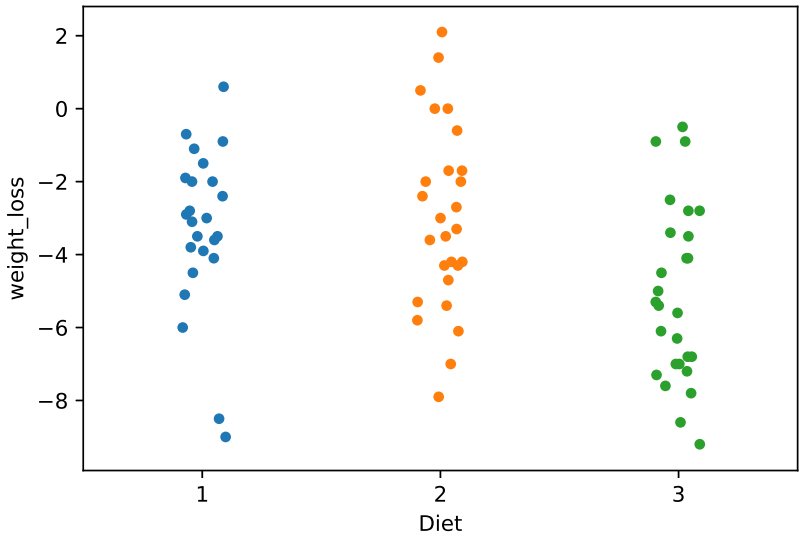
```
import seaborn as sns

sns.boxplot(x="Diet", y="weight_loss", data=df)
```



Boxplot bestätigt Vermutung aus Teilaufgabe a).

Stripchart:

```
sns.stripplot(x="Diet", y="weight_loss", data=df)
```



c) Die Nullhypothese ist, dass alle Mittelwerte gleich sind. Das heisst, alle Diäten sind gleich wirksam. Wir testen auf Signifikanzniveau von 5 %.

```
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
```

```
fit = ols("weight_loss~Diet", data=df).fit()
fit.f_pvalue

## 0.007164023698650354
```

Der p-Wert ist mit 0.007 kleiner als das Signifikanzniveau und somit wird die Nullhypothese verworfen. Das heisst, Diät 3 führt zu statistisch signifikant mehr Gewichtsverlust als die Diäten 1 und 2.

Lösung 9.3

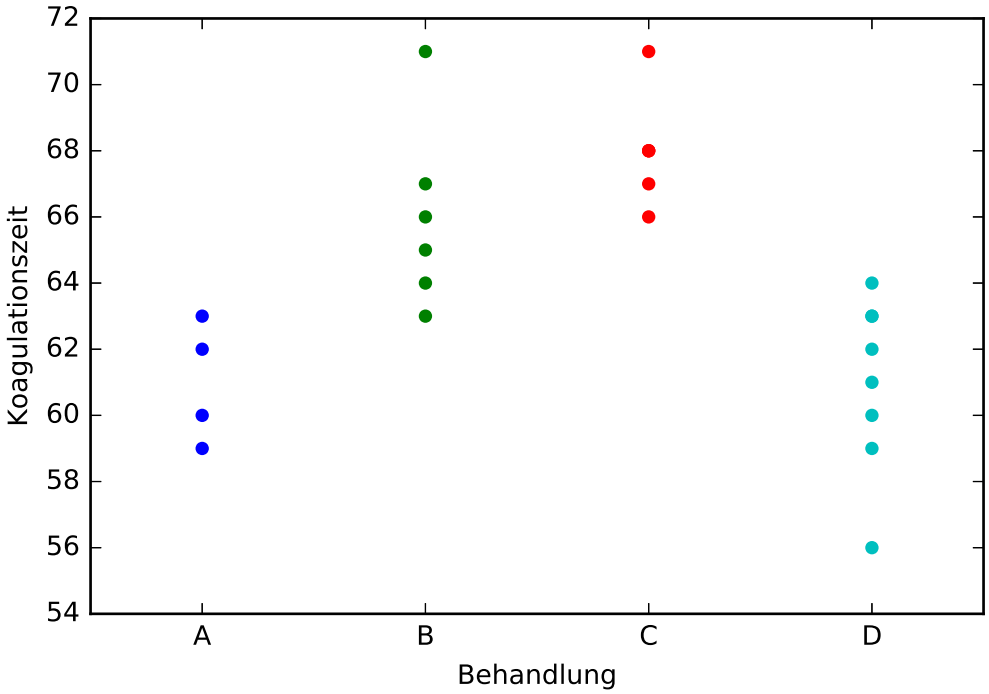
a) Die Daten werden wie folgt als Data Frame in **Python** eingelesen : (zu **R**)

```
from pandas import DataFrame
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st

df=DataFrame({
  "Behandlung": np.repeat(["A", "B", "C", "D"], [4, 6, 6, 8]),
  "Koagulationszeit" : [62, 60, 63, 59, 63, 67,
                        71, 64, 65, 66, 68, 66,
                        71, 67, 68, 68, 56, 62,
                        60, 61, 63, 64, 63, 59]
})
print(df)

##    Behandlung  Koagulationszeit
## 0           A                62
## 1           A                60
## 2           A                63
## 3           A                59
## 4           B                63
## 5           B                67
## 6           B                71
## 7           B                64
## 8           B                65
## 9           B                66
## 10          C                68
## 11          C                66
## 12          C                71
## 13          C                67
## 14          C                68
## 15          C                68
```

```
## 16           D                   56
## 17           D                   62
## 18           D                   60
## 19           D                   61
## 20           D                   63
## 21           D                   64
## 22           D                   63
## 23           D                   59
```

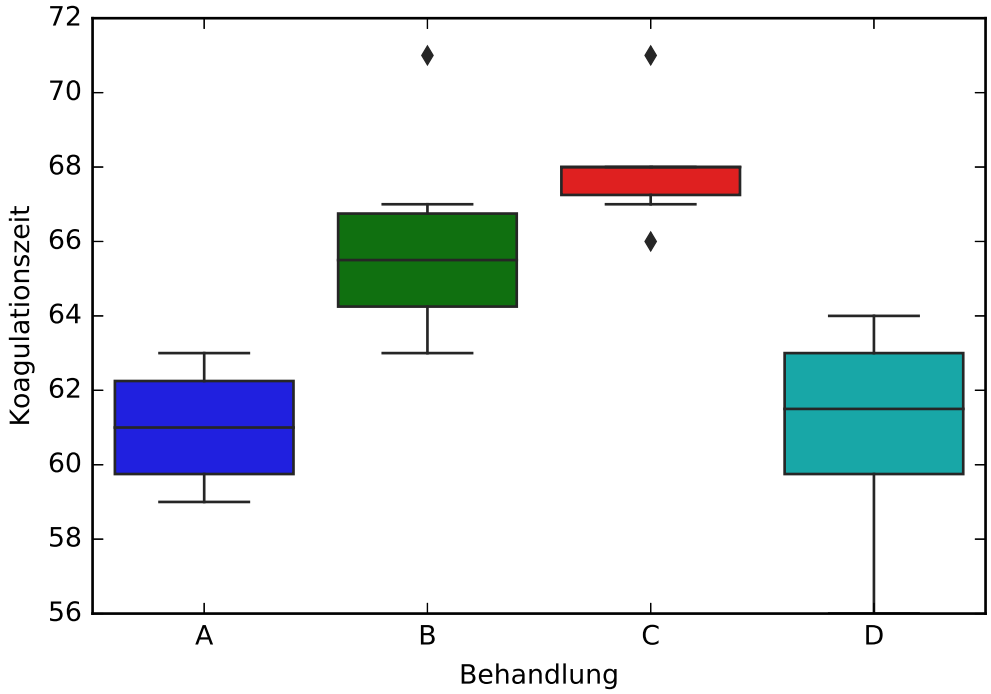
Wir erzeugen eine Stripchart Graphik wie folgt:

```
sns.stripplot(x="Behandlung", y="Koagulationszeit", data=df)
plt.xlabel("Behandlung")
plt.ylabel("Koagulationszeit")
plt.show()
```



Die entsprechenden Boxplots sind:

```
sns.boxplot(x="Behandlung", y="Koagulationszeit", data=df)
plt.xlabel("Behandlung")
plt.ylabel("Koagulationszeit")
plt.show()
```



Man sieht deutliche Unterschiede in der Lage der vier Stichproben. Vor allem die Stichprobe der Behandlung C hat deutlich höhere Werte als die drei anderen. Bezüglich der Streuung gibt es auch Unterschiede : Behandlung C weist eine kleine innere Streuung auf. Ansonsten ist aber die Streuung innerhalb der Gruppen klein im Vergleich zur Streuung zwischen den Gruppen.

b) (zu **R**)

c) Die Null-Hypothese lautet, dass sich die Behandlungsgruppen nicht unterscheiden, also dass die Gruppenmittelwerte

$$\mu_1 = \mu_2 = \mu_3 = \mu_4$$

sind, oder die Behandlungseffekte (eng. *treatment effects*)

$$\tau_2 = \tau_3 = \tau_4 = 0$$

```
from pandas import DataFrame
import pandas as pd
import numpy as np
import seaborn as sns
```

```
import scipy.stats as st
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
from statsmodels.graphics.factorplots import interaction_plot
from statsmodels.stats.outliers_influence import summary_table
import matplotlib.pyplot as plt
from scipy import stats
import warnings
warnings.filterwarnings("ignore")


df=DataFrame({
  "Behandlung": np.repeat(["A", "B", "C", "D"], [4, 6, 6, 8]),
  "Koagulationszeit" : [62, 60, 63, 59, 63, 67,
                        71, 64, 65, 66, 68, 66,
                        71, 67, 68, 68, 56, 62,
                        60, 61, 63, 64, 63, 59]
})
fit = ols("Koagulationszeit~Behandlung", data=df).fit()

fit.params


## Intercept          6.100000e+01
## Behandlung[T.B]    5.000000e+00
## Behandlung[T.C]    7.000000e+00
## Behandlung[T.D]   -3.197442e-14
## dtype: float64
```

```
anova_lm(fit)
```

```
##                df  sum_sq  mean_sq          F    PR(>F)
## Behandlung    3.0   228.0     76.0  13.571429  0.000047
## Residual     20.0   112.0      5.6        NaN       NaN
```

Da die transformierte Teststatistik mit $F = 13.57$ einen P-Wert von $5e-5$ hat und somit kleiner als das Niveau 5 % ist, wird die Nullhypothese verworfen und es gilt die Alternative. Dies war ja schon ersichtlich aus dem Boxplot.

Lösung 9.4

a) Die Daten werden wie folgt als Data Frame in **Python** eingelesen : (zu **R**)

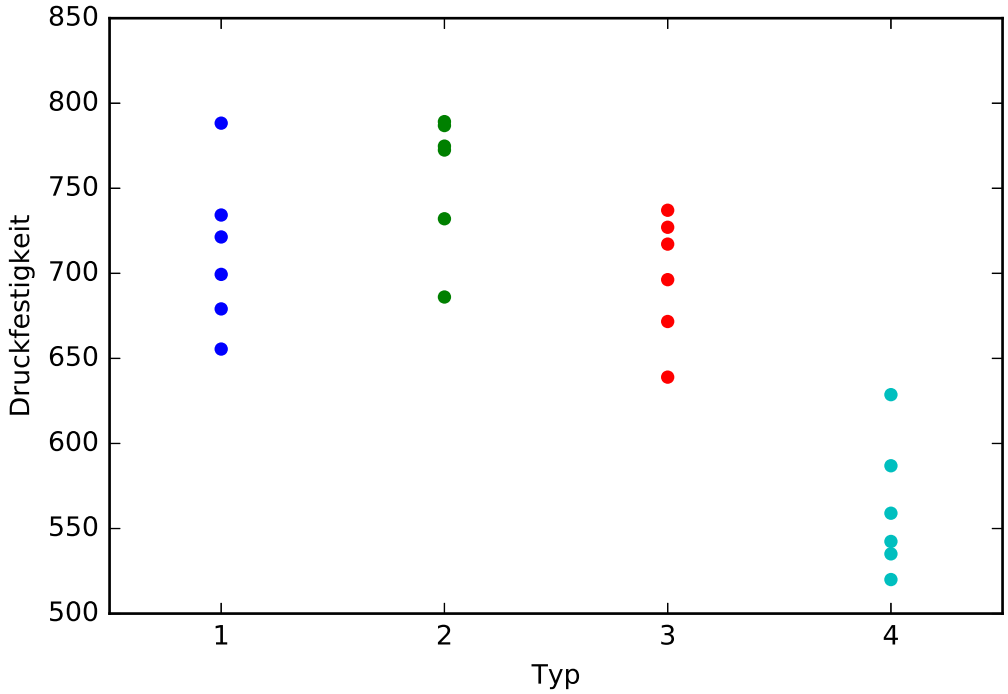
```
from pandas import DataFrame
import pandas as pd
import numpy as np
import seaborn as sns
```

```
import matplotlib.pyplot as plt
import scipy.stats as st

df=DataFrame({
  "Typ": np.repeat(["T1", "T2", "T3", "T4"], [6, 6, 6, 6]),
  "Druckfestigkeit" : [655.5, 788.3, 734.3, 721.4, 679.1, 699.4,
                       789.2, 772.5, 786.9, 686.1, 732.1, 774.8,
                       737.1, 639.0, 696.3, 671.7, 717.2, 727.1,
                       535.1, 628.7, 542.4, 559.0, 586.9, 520.0]
})
print(df)
```

```
##    Typ  Druckfestigkeit
## 0   T1            655.5
## 1   T1            788.3
## 2   T1            734.3
## 3   T1            721.4
## 4   T1            679.1
## 5   T1            699.4
## 6   T2            789.2
## 7   T2            772.5
## 8   T2            786.9
## 9   T2            686.1
## 10  T2            732.1
## 11  T2            774.8
## 12  T3            737.1
## 13  T3            639.0
## 14  T3            696.3
## 15  T3            671.7
## 16  T3            717.2
## 17  T3            727.1
## 18  T4            535.1
## 19  T4            628.7
## 20  T4            542.4
## 21  T4            559.0
## 22  T4            586.9
## 23  T4            520.0
```

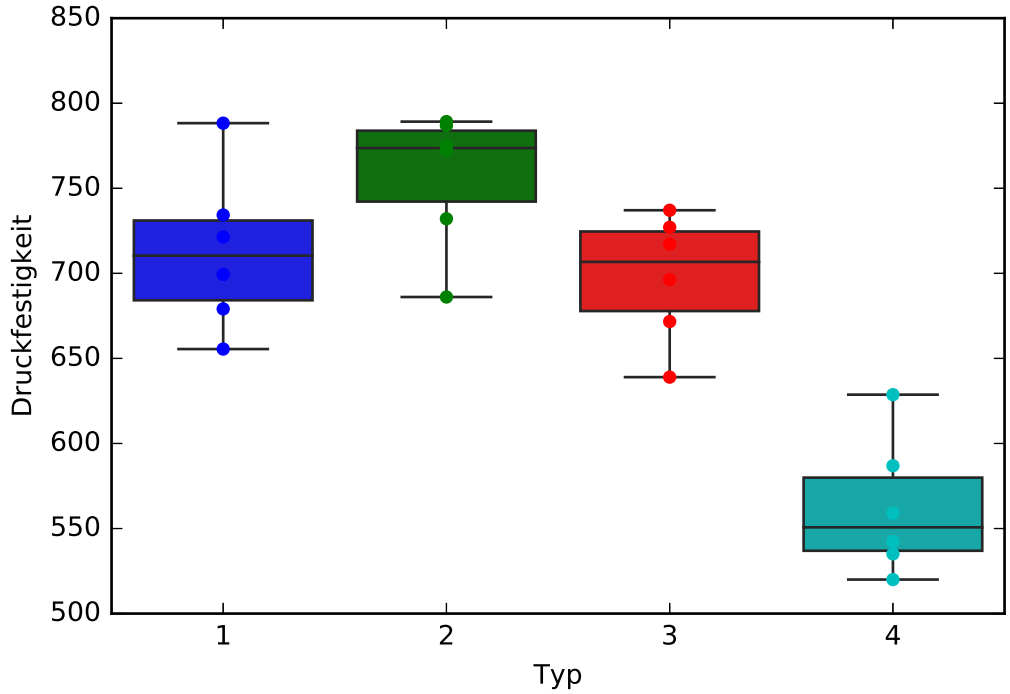
Wir erzeugen eine Stripchart Graphik wie folgt:

```
sns.stripplot(x="Typ", y="Druckfestigkeit", data=df)
plt.xlabel("Typ")
plt.ylabel("Druckfestigkeit")
plt.show()
```



Die entsprechenden Boxplots sind:

```
sns.boxplot(x="Typ", y="Druckfestigkeit", data=df)
plt.xlabel("Typ")
plt.ylabel("Druckfestigkeit")
plt.show()
```



Man sieht deutliche Unterschiede in der Lage der vier Stichproben. Vor allem die Stichprobe für den Typ 4 hat deutlich tiefere Werte als die drei anderen. Bezüglich der Streuung sind sich alle in etwa gleich (d.h. die Boxhöhe ist bei allen etwa gleich). Stichprobe zu Typ 2 ist linksschief, während Stichprobe zu Typ 4 rechtsschief ist.

b) Ein Gruppenmittelmodell lautet :

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij}$$

wobei μ einen globalen Parameter bezeichnet, den alle Gruppen miteinander teilen, τ_i bezeichnet die behandlungsspezifische Abweichung vom globalen Parameter μ und ε_{ij} ist der Fehlerterm. Wir wählen die Parametrisierung $\mu = \mu_1$, d.h., die behandlungsspezifische Abweichung τ_1 ist 0. Mit **Python** ergibt sich dann folgende Parameterschätzung für diese Modell: (zu **R**)

```
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
fit = ols("Druckfestigkeit~Typ", data=df).fit()

fit.params
```

```
## Intercept    713.000000
## Typ[T.T2]     43.933333
## Typ[T.T3]    -14.933333
## Typ[T.T4]   -150.983333
## dtype: float64
```

Die behandlungsspezifischen Abweichung lauten somit

$$\tau_{\text{TypT1}} = 0 \quad \tau_{\text{TypT2}} = 43.93333 \quad \tau_{\text{TypT3}} = -14.93333 \quad \tau_{\text{TypT4}} = -150.98333$$

c) Die Null-Hypothese lautet, dass sich die Typen nicht unterscheiden, also dass die Gruppenmittelwerte

$$\mu_1 = \mu_2 = \mu_3 = \mu_4$$

sind, oder die Behandlungseffekte (eng. *treatment effects*)

$$\tau_2 = \tau_3 = \tau_4 = 0$$

Die Alternative besagt, dass sich mindestens ein Gruppenpaar i und j im Gruppenmittelwert unterscheidet, d.h., $\mu_i \neq \mu_j$. (zu **R**)

```
anova_lm(fit)
```

```
##              df         sum_sq       mean_sq          F        PR(>F)
## Typ         3.0  127374.754583  42458.251528  25.094289  5.525450e-07
## Residual   20.0   33838.975000   1691.948750        NaN           NaN
```

Da die transformierte Teststatistik einen P-Wert von $5.5 \cdot 10^{-7}$ hat und somit kleiner als das Niveau 5 % ist, wird die Nullhypothese verworfen und es gilt die Alternative. Dies ist ja schon ersichtlich aus dem Boxplot: Typ 4 unterscheidet sich wesentlich von den anderen drei Typen; evt. auch Typ 2 von den Typen 1 und 3.

R-Code