Übungen zu Korrelation und linearer Regression **Serie 11**

Kommentar: Alle Datensätze und das Jupyter Notebook sind in einem Order gespeichert worden, damit Sie alles einfacher herunterladen können. Zudem habe ich Ihnen die Theorie zu Korrelation und linearer Regression im Anschluss an die Übungen noch aufgeschrieben. Vielleicht hilft es Ihnen! Sie schaffen das!

Aufgabe 1 Wir nehmen nochmals Ihre Daten zur Hand. Auf Moodle sind die Daten im File `Stats_Height_Weight.xlsx` zu finden. Wir möchten überprüfen, ob es einen linearen Zusammenhang zwischen der Grösse und dem Gewicht eines Menschen gibt, und eine lineare Regression durchführen. Im Theorieteil ganz am Schluss ist die gesamte Analyse aufgeschrieben, und auf Moodle finden Sie das zugehörige Jupyter Notebook `Stats_Height_Weight.ipynb`.

(a) Lesen Sie die Theorie auf den Seiten 7 bis 10 gut durch und versuchen Sie die Befehle im Jupyter Notebook zu verstehen und zu reproduzieren.

(b) Was ist nach Modell die Vorhersage für Ihr Gewicht? Stimmt dieser Wert einigermassen mit Ihrem tatsächlichen Gewicht überein?

(c) Wie gross ist die Abweichung zwischen Ihrem tatsächlichen und Ihrem geschätzen Gewicht?

Aufgabe 2 Die folgende Tabelle enthält die Länge und das Gewicht von 9 Schlangen. Auf Moodle sind diese Daten im File `Snakes.xlsx` gespeichert.

Nummer	Länge X [cm]	Gewicht Y [g]
1	60	136
2	69	198
3	66	194
4	64	140
5	54	93
6	67	172
7	59	116
8	65	174
9	63	145

(a) Stellen Sie die Daten in einem Streudiagramm dar.

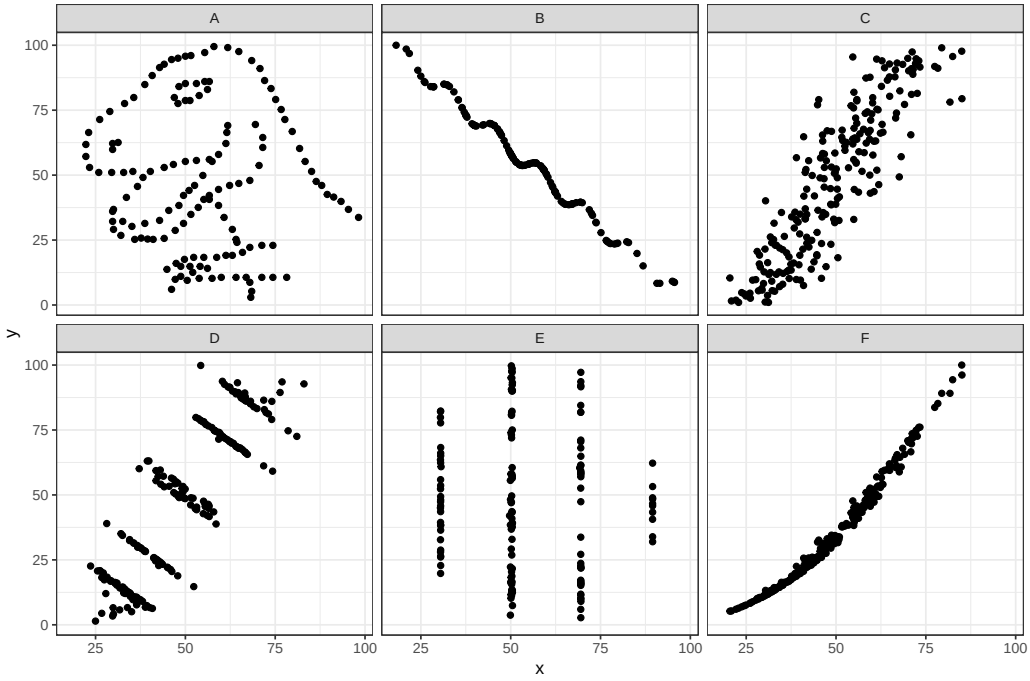
(b) Ist es aufgrund des Plots sinnvoll, von einem linearen Zusammenhang zwischen X und Y auszugehen?

(c) Berechnen Sie von Hand die Regressionsgerade mit Y als abhängige und X als erklärende Variable. Berechnen Sie ebenfalls die Residuen von Hand.

(d) Die Residuen müssen normalverteilt sein. Diese Annahme kann man mit einem Q-Q-Plot überprüfen. Sind Sie nun in der Lage den zugehörige Q-Q-Plot von Hand zu zeichnen? Falls es nicht klappt, machen Sie sich keine Sorgen, wir kommen darauf züruck.

(e) Führen Sie nun eine lineare Regression mit Python durch. Erhalten Sie die gleichen Werte für die Koeffizienten wie bei der Berechnung von Hand?

Aufgabe 3 Betrachten Sie die folgenden 6 Streudiagramme.



(a) Ist ein linearer und/oder monotoner Zusammenhang zwischen den Variablen y und x in den Datensätzen A-F ersichtlich? Achtung: Jeder Datensatz muss separat betrachtet werden.

	A		B		C		D		E		F	
	ja	nein	ja	nein	ja	nein	ja	nein	ja	nein	ja	nein
linear	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
monoton	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

(b) Schätzen Sie den zugehörigen Pearsons Korrelationskoeffizienten anhand der Streudiagramme.

(c) Auf Moodle stehen Ihnen die 6 Datensätzen A-F in einen File Names `Dino.xlsx` zur Verfügung. Achtung: Sie müssen jeweils die richtigen Zeilen selektieren. Der Datensatz befindet sich in dem sogenannten „long“ Format. Berechnen Sie für jeden Datensatz den zugehörigen Pearsons Korrelationskoeffizienten. Sie können dafür die folgenden Befehle in Python verwenden:

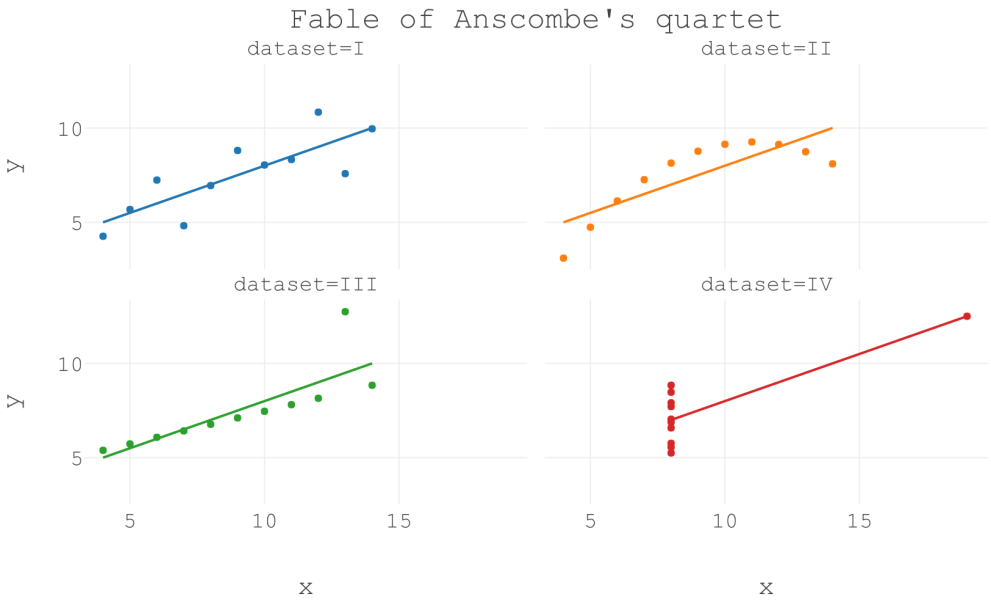
```
> from scipy import stats
# use the names that you choose to replace data.x and data.y
> Pearsons_r, p_value = stats.pearsonr(data.x, data.y)
> print(Pearsons_r)
```

Die erste Ausgabe ist Pearsons Korrelationskoeffizient und die zweite Ausgabe ist der zugehörige p-Wert. In diesem Fall lautet die Nullhypothese, dass es keinen linearen Zusammenhang zwischen den Daten gibt.

(d) Vergleichen Sie nun die Ergebnisse von Python mit Ihren Schätzungen.

Aufgabe 4 Die 4 berühmten Datensätze names *Anscombe's Quartet* vom Mathematiker Anscomb sind auf Moodle im File `anscombes_quartet.xlsx` gespeichert. Lösen Sie diese Aufgabe nur mit Python.

(a) Visualisieren Sie die 4 Datensätzen in einer Graphik. Ihr Ergebnis muss nicht genau gleich aussehen, aber als Beispiel ist der folgende Plot gemeint:



(b) Berechnen Sie für jeden Datensatz den zugehörigen Pearsons Korrelationskoeffizienten und vergleichen Sie die Werte, die Sie erhalten. Achtung: Sie müssen jeweils die richtigen Zeilen selektieren. Die Daten befindet sich in dem sogenannten „long" Format.

(c) Führen Sie für jeden Datensatz eine lineare Regressionsanalyse mit Python durch. Vergleichen Sie die Koeffizienten der verschiedenen Modellierungen miteinander.

(d) Machen Sie Q-Q-Plots um die Normalität der Residuen zu überprüfen (muss für jeden Datensatz separat ausgeführt werden). Falls es nicht klappt, machen Sie sich keine Sorgen, wir kommen darauf züruck.

(e) Was sind Ihre Erkenntnisse?

Für den Theorieteil beziehe ich mich auf Unterlagen, die von Walter Businger, Horst Heck, Reto Spöhel und Jasmin Wandel erarbeitet wurden. Ich bedanke mich herzlichst bei ihnen für den Gebrauch ihrer Unterlagen und für die grosszügige Unterstützung, die ich stets erfahren habe.

Korrelation

Das *Streudiagramm* liefert uns einen visuellen Eindruck von der gemeinsamen Verteilung zweier numerischer Variablen in der Stichprobe. Wir nennen die Variablen X und Y und bezeichnen die Werte in unserer Stichprobe als (x_i, y_i) $(i = 1, 2, \ldots, n)$. Im Streudiagramm wird jedes Datenpaar (x_i, y_i) als Punkt in einem zweidimensionalen Koordinatensystem eingezeichnet.

Abbildung 1 zeigt Streudiagramme von vier verschiedenen (fiktiven) Datensätzen. Im linken oberen Diagramm erkennt man einen deutlichen *linearen* Zusammenhang zwischen x- und y-Werten: die Punktepaare liegen in etwa auf einer Geraden mit positiver Steigung. Im rechten oberen Diagramm sieht man ebenfalls einen linearen Zusammenhang, allerdings etwas schwächer und die angedeutete Gerade weist negative Steigung auf. Im linken unteren Diagramm erkennt man keinerlei Zusammenhang zwischen beiden Variablen, und das rechte untere Diagramm zeigt einen deutlichen *nichtlinearen* Zusammenhang.

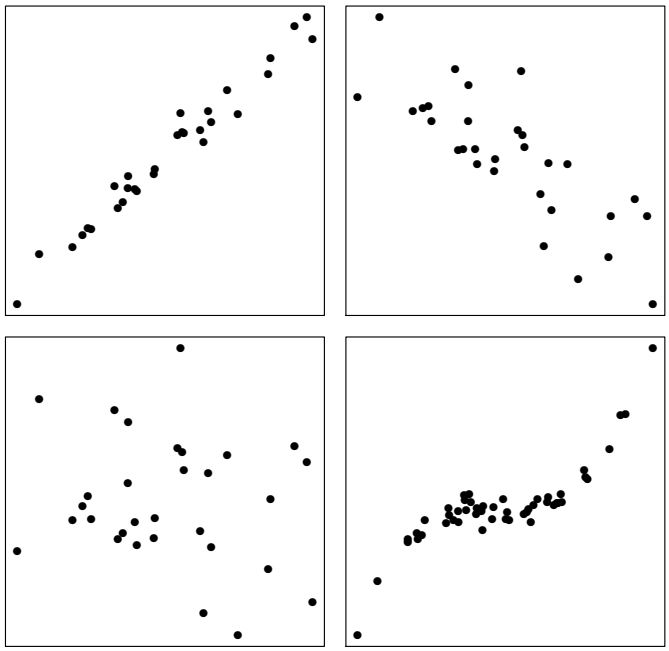


Abbildung 1 – Beispiele von Streudiagrammen

Wir haben bereits einige numerische Kenngrössen für einzelne numerische Variablen kennengelernt. Analog dazu gibt es auch Kenngrössen, um den Zusammenhang zwischen zwei numerischen Variablen zu beschreiben. Wir beschränken uns hier bloss auf Kenngrössen, die *lineare* Zusammenhänge messen (Pearsons Korrelation).

Eine solche Kenngrösse ist die *empirische Kovarianz* oder *Stichprobenkovarianz* von Stichprobenwerten (x_i, y_i) $(i = 1, \ldots, n)$ zweier Zufallsvariablen X und Y:

$$s_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})$$

Besser interpretierbar ist die normierte Form der empirischen Kovarianz, der sogenannte empirische *Korrelationskoeffizient*:

$$r = \frac{s_{xy}}{s_x \cdot s_y} \; ;$$

s_x und s_y bezeichnen dabei die empirische Standardabweichung der Stichprobenwerte x_i bzw. y_i $(i = 1, \ldots, n)$. In Python lässt sich Pearsons Korrelationskoeffizient mit den Funktionen `stats.pearsonr` folgendermassen berechnen

```
> from scipy import stats
## data is the name of the dataset
> Pearsons_r, p_value = stats.pearsonr(data.x,  data.y)
> print(Pearsons_r)
> print(p_value)
```

Die erste Ausgabe von `stats.pearsonr` ist Pearsons Korrelationskoeffizient und die zweite Ausgabe ist der zugehörige p-Wert. Die Nullhypothese lautet, dass es keinen linearen Zusammenhang zwischen den Daten gibt.

Der Korrelationskoeffizient besitzt folgende Eigenschaften:

- $r \in [-1, 1]$
- $r = 1$ oder $r = -1$ genau dann, wenn alle Punkte (x_i, y_i) *exakt* auf einer Geraden mit positiver, bzw. negativer Steigung liegen.
- r hat positives (bzw. negatives) Vorzeichen, falls ein linearer Zusammenhang zwischen X und Y mit positiver (bzw. negativer) Steigung besteht. Der Betrag des Korrelationskoeffizienten sagt *nichts* über die Steigung des linearen Zusammenhangs aus, aber: Je näher der Betrag bei 1 ist, desto *stärker* ist der lineare Zusammenhang zwischen den beiden Variablen, d.h. desto weniger streuen die Punkte um eine Gerade.
- $r = 0$ bedeutet, dass kein linearer Zusammenhang zwischen X und Y besteht. *Achtung*: Dies bedeutet *nicht*, dass kein Zusammenhang zwischen X und Y besteht – er ist unter Umständen einfach nur nicht linearer Natur.
- r bleibt unverändert, wenn man
  - die Rollen von X und Y vertauscht,
  - zu allen x- oder y-Werten eine Konstante addiert,
  - alle x- oder y-Werte mit einer positiven Konstanten multipliziert.

Zur Interpretation des empirischen Korrelationskoeffizienten ist die folgende Faustregel hilfreich:

$\lvert r\rvert = 0$	kein linearer Zusammenhang (zwischen X und Y)
$0 < \lvert r\rvert < 0.5$	schwacher linearer Zusammenhang
$0.5 \leq \lvert r\rvert < 0.8$	mittlerer linearer Zusammenhang
$0.8 \leq \lvert r\rvert < 1$	starker linearer Zusammenhang
$\lvert r\rvert = 1$	perfekter linearer Zusammenhang

Allgemein sollte man den Korrelationskoeffizienten immer nur zusammen mit dem Streudiagramm interpretieren! Z.B. zeigt die Abbildung 2 sechs unterschiedliche Streudiagramme (simulierter Datensätze), wobei die zugehörigen Korrelationskoeffizienten bei allen sechs Diagrammen identisch sind (nämlich $r = 0.8$).

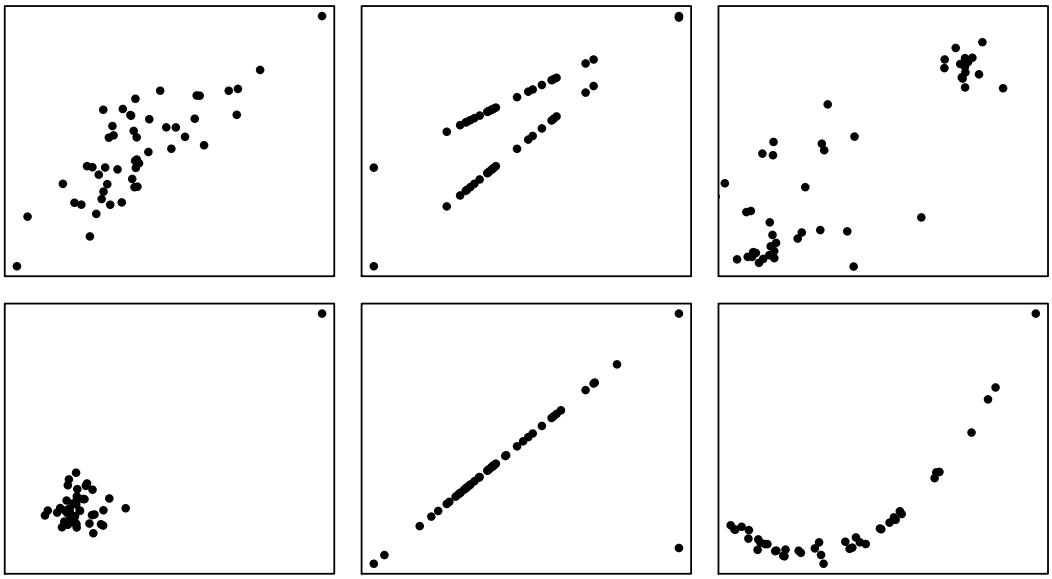


Abbildung 2 – Verschiedene Streudiagramme mit identischem Korrelationskoeffizient

Die häufigste Fehlinterpretation des Korrelationskoeffizienten betrifft die Kausalität: Ein linearer Zusammenhang zwischen zwei Variablen beantwortet die Frage nach Kausalität *nicht*!

Dazu das folgende amusante Beispiel

In Schweden wurde zwischen 1972 und 1985 eine hohe Korrelation zwischen der Storchenpopulation und der Geburtenrate festgestellt. Dennoch hängen diese beiden Ereignisse nicht kausal miteinander zusammen. Stattdessen sind hier Drittvariablen wie etwa die Industrialisierung von Bedeutung, die mutmasslich sowohl zu einem Absinken der Geburtenrate als auch zu einer verringerten Storchenpopulation führten.

Lineare Regression

Haben wir mit Hilfe eines Streudiagramms einen linearen Zusammenhang zwischen zwei Variablen festgestellt, können wir mit Hilfe der *linearen Regression* ein sogenanntes *lineares Modell* an die Daten anpassen. Mit dem linearen Modell machen wir nun insbesondere eine Annahme über die Richtung eines vermuteten kausalen Zusammenhangs.

Ein lineares Modell sieht wie folgt aus:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \quad i = 1, 2, \ldots, n \ .$$

Die Zahlen β_0 und β_1 heissen *Regressionskoeffizienten* und beschreiben Achsenabschnitt und Steigung des linearen Zusammenhangs zwischen X und Y. Die *Fehlerterme* ε_i heissen *Residuen*. Diese brauchen wir, weil wir in der Realität nie perfekte lineare Zusammenhänge finden. Man nimmt in der Regel an, dass die Fehler ε_i normalverteilte, unabhängige Zufallsvariablen mit Erwartungswert 0 sind. Diese Bedingung schreiben wir mathematisch so auf

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)\,, \quad i.i.d, \quad i = 1, 2, \cdots, n$$

Vergessen Sie nicht, diese Annahme jeweils mit Hilfe eines Q-Q-Plots zu überprüfen! Diese normalverteilten Fehlerterme bilden den Kern der *statistischen* linearen Regression. Die Varianz σ^2 ist eine Unbekannte.

Wie schätzt man nun für einen gegebenen Datensatz die Koeffizienten β_0 und β_1? Das Vorgehen ist ganz einfach: Wir müssen die Werte so wählen, dass die Fehlerquadratsumme

$$F(\beta_0,\, \beta_1) := \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_i)\right)^2$$

möglichst klein wird. Man kann einfach zeigen, dass $F(\beta_0,\, \beta_1)$ minimal wird für

$$\widehat{\beta_1} = \frac{\frac{1}{n}\left(\sum_{i=1}^{n} x_i y_i\right) - \overline{x} \cdot \overline{y}}{\frac{1}{n}\left(\sum_{i=1}^{n} x_i^2\right) - \overline{x}^2}, \qquad \widehat{\beta_0} = \overline{y} - \widehat{\beta_1}\overline{x}.$$

Bemerkung

Im Gegensatz zur Korrelation können bei der linearen Regression die Rollen von X und Y nicht einfach getauscht werden (das obige Minimierungsproblem würde zu einer anderen Lösung führen). Es ist also wichtig zu überlegen, welche Variable Y wir mit welcher Variable X erklären wollen, bzw. in welche Richtung wir einen kausalen Zusammenhang vermuten! Zum Beispiel ist klar, dass man *nicht* das Gewicht eines Autos mit Hilfe des Verbrauchs schätzen möchte. Umgekehrt macht es aber durchaus Sinn eine Schätzung für den durchschnittlichen Verbrauch in Abhängigkeit des Autogewichts zu haben, da vermutet wird, dass das Gewicht des Autos einen Einfluss auf den Verbrauch hat (und eben nicht umgekehrt).

Vorgehensweise in der Praxis (Teil I)

Wenn wir eine lineare Regressionsanalyse (oder irgendeine Modellierung) machen, gibt es eine klare Reihenfolge von Schritten, die betrachten werden müssen. Wenn man am Code schreiben ist, überspringt man natürlich gewisse Schritte. Wenn wir aber unsere Ergebnisse mit anderen Menschen teilen, ist es sehr wichtig, dass wir die Diskussion sauber aufschreiben, korrekt aufgleisen und versuchen eine sinnvolle Reihenfolge zu beachten.

Wir haben Ihre und meine Grösse und Gewicht gemessen. Als Beispiel nehmen wir nochmals den Datensatz `Stats_Height_Weight.xlsx` zur Hand und gehen diese Schritte durch (hier nur Teil I, Teil II wird nächtste Woche folgen).

Schritt 1. Als erstes machen wir einen Scatterplot der Daten um zu sehen, ob überhaupt ein linearer Zusammenhang zwischen Daten für die Zufallsvariablen X und Y erkennbar ist. Es muss klar sein, welche die abhängige und welche die unabhängige Variable ist. Wir möchten das Gewicht eines Menschen anhand seiner Grösse vorhersagen und nicht umgekehrt!

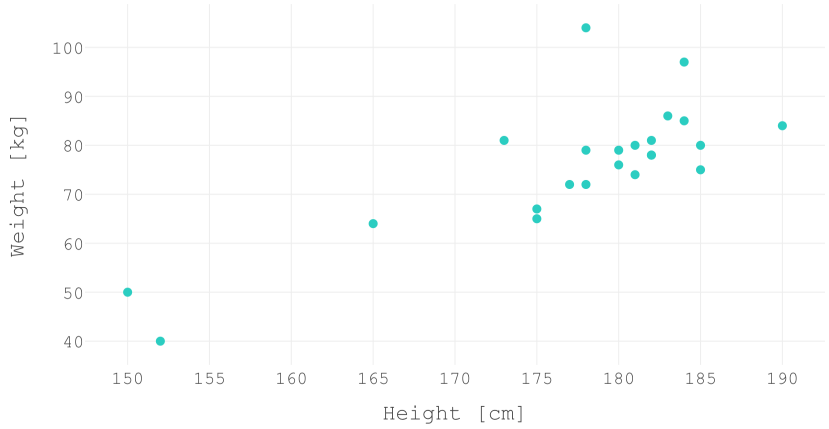


Abbildung 3 – Gewicht und Grösse von 23 Personen

Wir sehen bereits, dass es bei diesem Datensatz mehrere Problem gibt:

(i) Als Erstes sehen wir, dass wir zu wenig Datenpunkte für die Frauen haben. Das Grössen-Interval von ca. 155 cm bis ca. 170 cm ist sehr stark untervertreten. Allgemein haben wir wahrscheinlich zu wenig Datenpunkte.

(ii) Als Zweites hätten wir wahrscheinlich das Merkmal Geschlecht aufnehmen müssen. Im nachhinein ist es schwierig die Daten der Männer von den Daten der Frauen zu trennen. Grundsätzlich würde man in diesem Fall zwei getrennte Modelle machen, d.h. ein Modell für die Männer und ein Modell für die Frauen.

Der Datensatz ist nicht gut geeignet für eine lineare Regression. Dennoch fahren wir jetzt mal weiter und versuchen herauszufinden, was Python mit diesen Daten anstellt, wenn wir modellieren.

Schritt 2. Unser Modell lautet

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \quad i = 1, 2, \ldots, n\,,$$

wobei die Residuen ε_i normalverteilte, unabhängige Zufallsvariablen mit Erwartungswert 0 sind. Diese Bedingung schreiben wir mathematisch so auf

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)\,, \quad i.i.d, \quad i = 1, 2, \cdots, n$$

Wir möchten Schätzungen $\hat{\beta}_0$ und $\hat{\beta}_1$ für die Koeffizienten β_0 und β_1 so bestimmen, dass die Fehlerquadratsumme F minimiert wird, d.h.

$$F = \sum_{i=1}^{n} \varepsilon_i^2 \quad \longrightarrow \quad \text{Minimum}$$

Schritt 3. Wir lesen den Datensatz in Python ein und führen mit Hilfe der Befehle von `statsmodels.api` ganz einfach eine lineare Regression durch. Achtung `statsmodels` muss natürlich im Vorfeld installiert werden. So sieht z.B. die Codierung aus:

```
> import pandas as pd
> import statsmodels.api as sm
```

```
> wing = pd.read_excel("Stats_Height_Weight.xlsx")

## X usually means our input variables (or independent variables)
> X = wing["Groesse [cm]"]
## y usually means our output/dependent variable
> y = wing["Gewicht [kg]"]
## let's add an intercept (beta_0) to our model
> X = sm.add_constant(X)
# Note the difference in argument order
model = sm.OLS(y, X).fit()  ## sm.OLS(output, input)
# Print out the statistics
> model.summary()
```

Falls alles richtig funktioniert, erhalten Sie den folgenden Output:

data.txt

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           Gewicht [kg]   R-squared:                       0.621
Model:                            OLS   Adj. R-squared:                  0.603
Method:                 Least Squares   F-statistic:                     34.48
Date:                Thu, 14 May 2020   Prob (F-statistic):           7.91e-06
Time:                        08:58:59   Log-Likelihood:                -80.593
No. Observations:                  23   AIC:                             165.2
Df Residuals:                      21   BIC:                             167.5
Df Model:                           1
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const           -116.4556     32.816     -3.549      0.002    -184.701     -48.211
Groesse [cm]       1.0847      0.185      5.872      0.000       0.701       1.469
==============================================================================
Omnibus:                       19.593   Durbin-Watson:                   2.064
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               24.305
Skew:                           1.786   Prob(JB):                     5.28e-06
Kurtosis:                       6.549   Cond. No.                     3.32e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.32e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Schritt 4. An dieser Stelle diskutiert man die erhaltenen Ergebnisse. Wir sind noch nicht in der Lage das zu tun, aber das kommt noch. Nächste Woche in Teil II dieser Vorlesung werden wir den Output von Python detaillierter anschauen.

Schritt 5. Alle nötigen Annahmen für das Modell müssen überprüft werden. In unserem Fall heisst das, dass wir überprüfen müssen, ob die Residuen normalverteilt sind. Das machen wir einerseits visuell mit einem Q-Q-Plot und andererseits mit speziellen Hypothesentests, die die Normalität der Daten testen. Ebenfalls werden wir diese Schritte im Unterricht noch behandeln.

Ein Paar wichtige Informationen haben wir aber bereits gesammelt.

(i) R-squared: 0.621 ist nichts anderes als Pearsons Korrelationskoeffizient im Quadrat, d.h. r^2. Das können wir sehr einfach überprüfen, indem wir r berechnen und das Ergebnis quadrieren

```
> from scipy import stats
> Pearsons_r, p_value = stats.pearsonr(wing["Groesse [cm]"],
                                       wing["Gewicht [kg]"])
> Persons_r**2
ans = 0.6215
```

Die erste Ausgabe von `stats.pearsonr` ist Pearsons Korrelationskoeffizient und die zweite Ausgabe ist der zugehörige p-Wert. Die Nullhypothese lautet, dass es keinen linearen Zusammenhang zwischen den Daten gibt. Wir erhalten einen sehr kleinen p-Wert von 10^{-6}, d.h. wir können die Nullhypothese verwerfen. Rein vom Test her, scheint es einen linearen Zusammen in unseren Daten zu geben, aber das muss man immer visuell kontrollieren, so wie wir es gemacht haben. Wir sehen einen mittelstarken linearen Zusammenhang da $r \approx 0.788$.

(ii) Unsere Schätzung für den Achsenabschnitt lautet $\hat{\beta}_0 = -116.4556$. und für die Steigung erhalten wir $\hat{\beta}_1 = 1.0847$. D.h. die modellierte Gerade ist gegeben durch die Formel:

$$\begin{aligned} \hat{y} &= \hat{\beta}_0 + \hat{\beta}_1 \cdot x \\ &= -116.4556 + 1.0847 \cdot x \end{aligned}$$

Ich kann das Modell nun verwenden um Vorhersagen zu machen. Z.B. kann ich meine Grösse von 152 cm eingeben und nach Modell wäre ich

$$-116.4556 + 1.0847 \cdot 152 \approx 48.42 \text{ kg}\,.$$

Ebenfalls kann ich nun die `predict` Funktionalität folgendermassen verwenden und erhalte das gleiche Ergebnis

```
> predictions = model.predict([1, 152])
ans = [48.41517726]
```

Welche Vorhersage erhalten Sie für Ihr Gewicht?

(iii) Die Residuen (Fehlerterme) können wir nun problemlos berechnen; diese sind nichts anderes als die Differenz zwischen den Messwerten y_i und den geschätzen Werten $\hat{y}_i$ (also die Vorhersagen des Models). Für mein Gewicht ist der zugehörige Fehlerterm also ca. $40 - 48.42 = -8.42$ kg.

In Python kann ich alle Residuen ganz einfach folgendermassen ausgeben lassen

```
> print(model.resid)
```

Und um die Normalverteiltheit der Residuen zu Überprüfen kann ich einen Q-Q-Plot folgendermassen machen

```
> fig = sm.qqplot(model.resid, dist = stats.norm, fit = True, line="45")
```

Falls Sie Schwierigkeiten mit Q-Q-Plots haben, ist das kein Problem. Wir behandeln das Thema nochmals im Unterricht.