Lecture Advanced Statistics (Prof. Dr. Kauffeldt)

Table of Contents

Chapter 1: Methods

Preliminaries

- Load Libaries
- Load Data

In [2]:
```
import httpimport
with httpimport.remote_repo(url='https://raw.githubusercontent.com/ProfKauf/Modules
    import profK_libraries, profK_statistics
from profK_libraries import *
from profK_statistics import *
```

In [3]:
```
#data
data= pd.read_excel ('C:\\Users\\kauffeldt\\Dropbox\\Teaching\\3_Programme\\Data\\w
data.head()
```

Out[3]:

	Music	Slow songs or fast songs	Dance	Folk	Country	Classical music	Musical	Pop	Rock	Metal or Hardrock	...	Age
0	5.0	3.0	2.0	1.0	2.0	2.0	1.0	5.0	5.0	1.0	...	20.0
1	4.0	4.0	2.0	1.0	1.0	1.0	2.0	3.0	5.0	4.0	...	19.0
2	5.0	5.0	2.0	2.0	3.0	4.0	5.0	3.0	5.0	3.0	...	20.0
3	5.0	3.0	2.0	1.0	1.0	1.0	1.0	2.0	2.0	1.0	...	22.0
4	5.0	3.0	4.0	3.0	2.0	4.0	3.0	5.0	3.0	1.0	...	20.0

5 rows × 150 columns

Section 1.1: Descriptive Methods

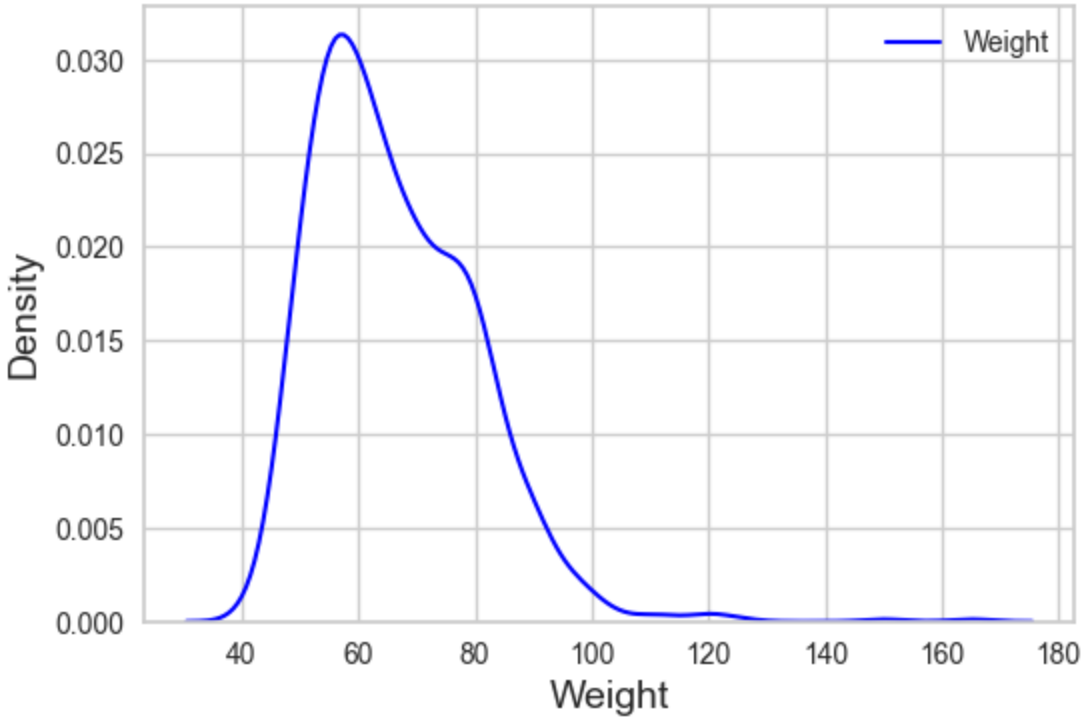
1.1.1 Graphical Methods

If the level of measurement of the target variable is at least **quantitative**:

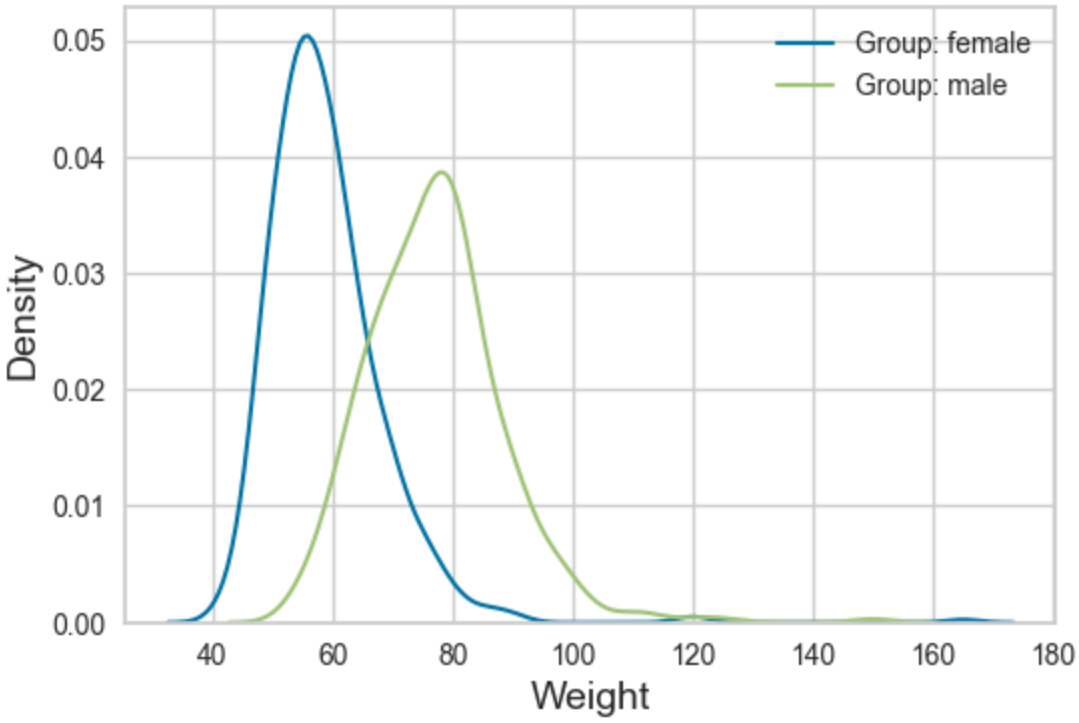
- Distribution Plot
- Bar Plot

Distribution Plots:

In [4]:
```
plots.dist(data['Weight'],fig=[6,4],labelsize=14,ticksize=10,legsize=10,linewidth=1
```



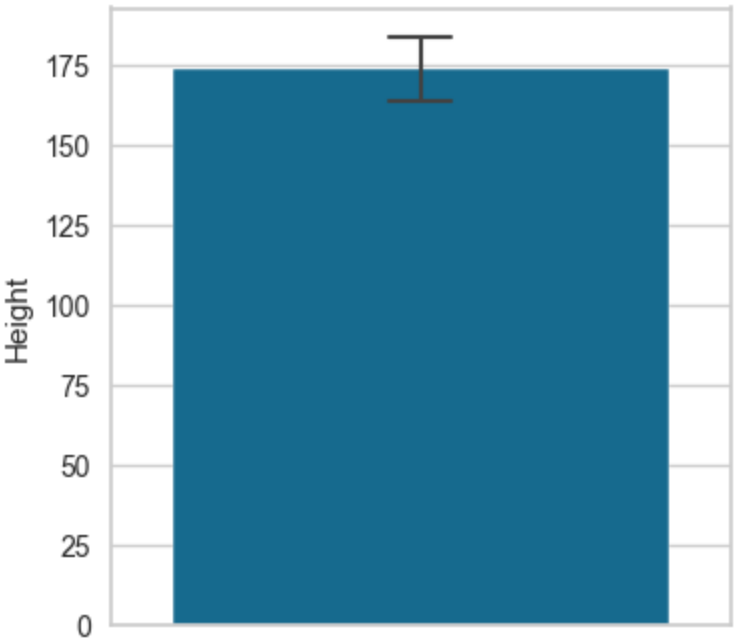
In [5]:
```
plots.dist(data=data,var='Weight',groupvar='Gender',fig=[6,4],labelsize=14,ticksize
```



Bar Plots:

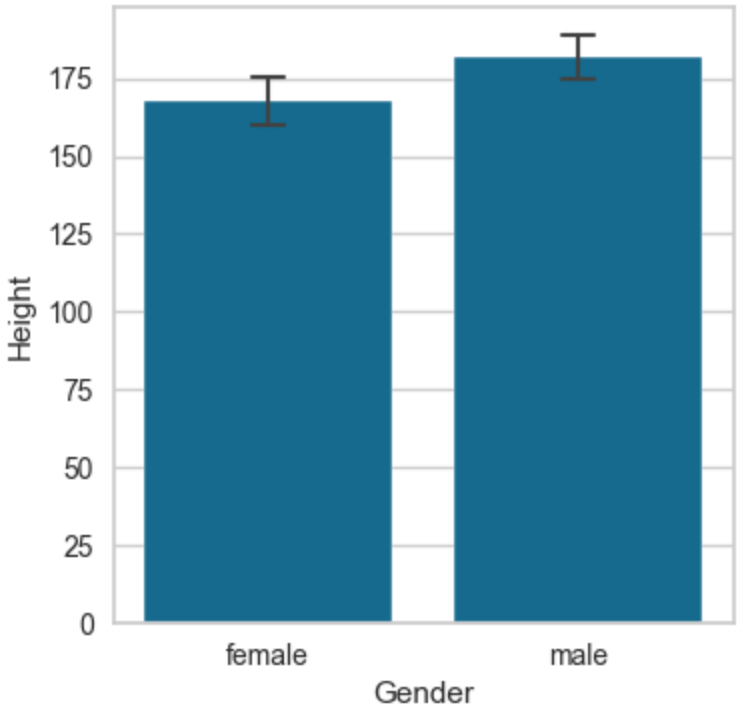
In [6]:
```
plt.figure(figsize=(4,4))
sns.barplot(y='Height',data=data,ci='sd',capsize=.1,errwidth=1.5)
```

Out[6]: <Axes: ylabel='Height'>



In [7]:
```
plt.figure(figsize=(4,4))
sns.barplot(x='Gender',y='Height',data=data,ci='sd',capsize=.1,errwidth=1.5)
```

Out[7]: <Axes: xlabel='Gender', ylabel='Height'>



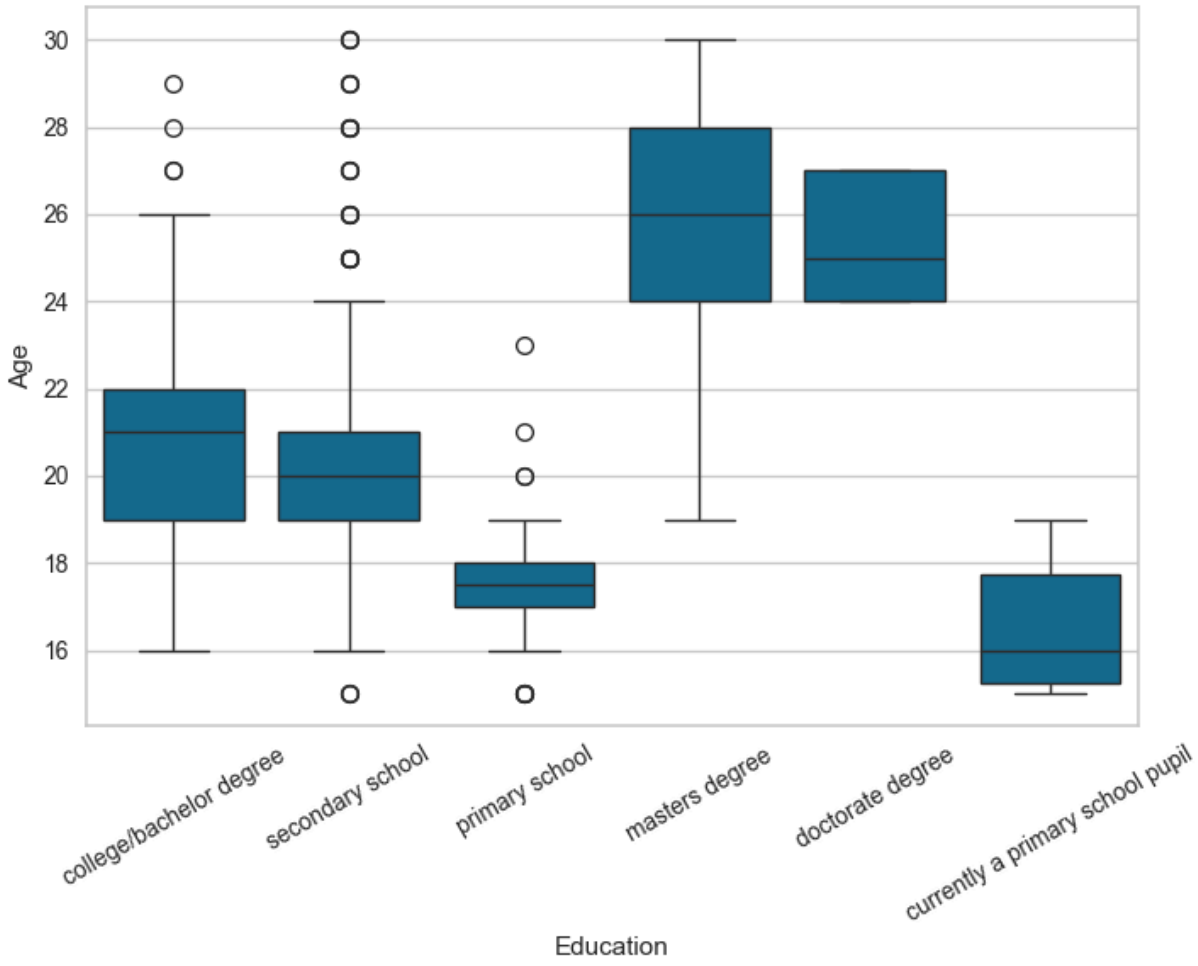
If the level of measurement of the target variable is at least **ordinal**:

- Box Plot
- Cat Plot
- Violin Plot

Box Plot:

In [8]:
```
sns.boxplot(x='Education',y='Age',data=data)
plt.xticks(rotation=30)
```

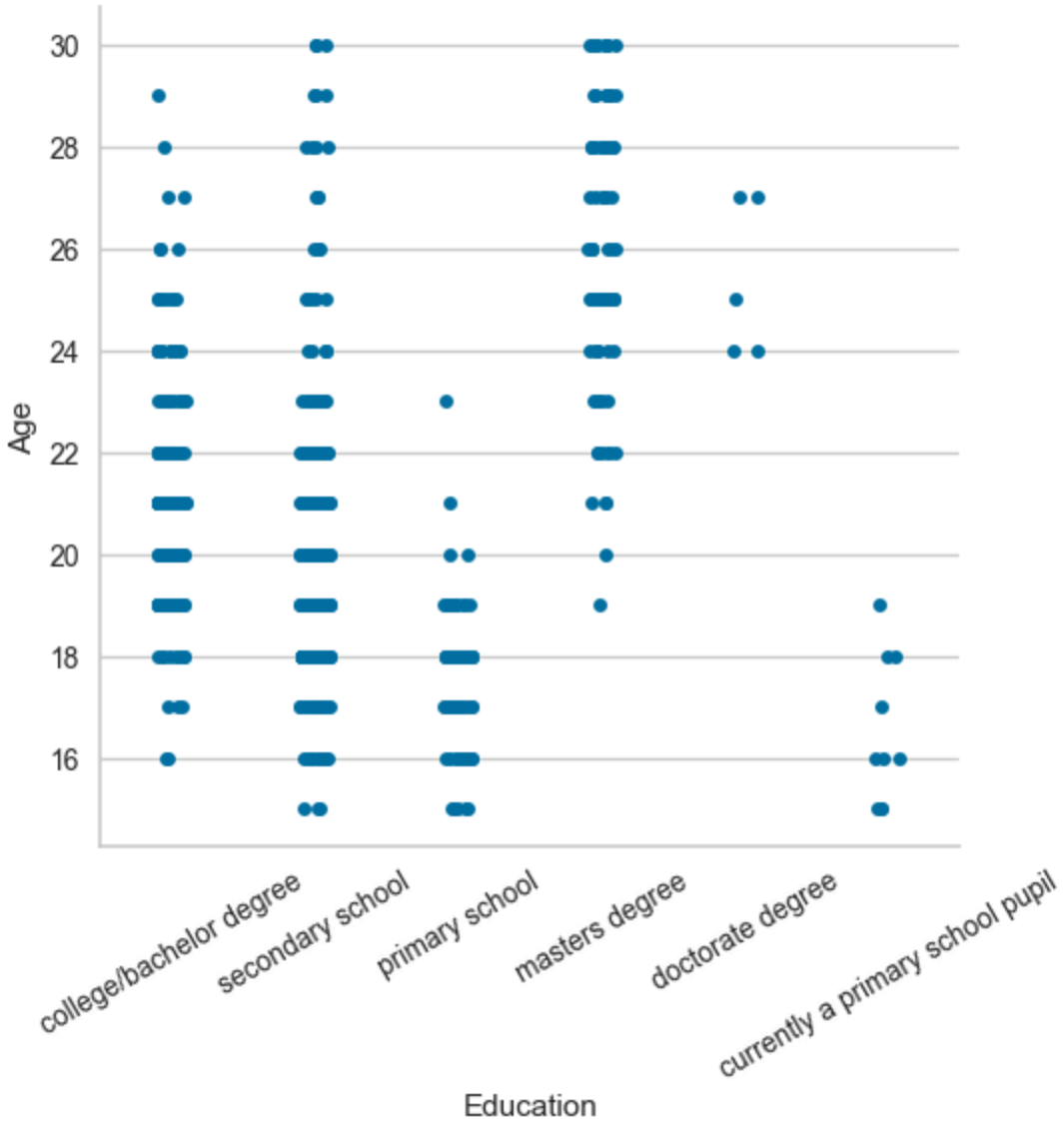
Out[8]:
```
([0, 1, 2, 3, 4, 5],
 [Text(0, 0, 'college/bachelor degree'),
  Text(1, 0, 'secondary school'),
  Text(2, 0, 'primary school'),
  Text(3, 0, 'masters degree'),
  Text(4, 0, 'doctorate degree'),
  Text(5, 0, 'currently a primary school pupil')])
```



Cat Plot:

In [9]:
```
sns.catplot(x='Education',y='Age',data=data)
plt.xticks(rotation=30)
```

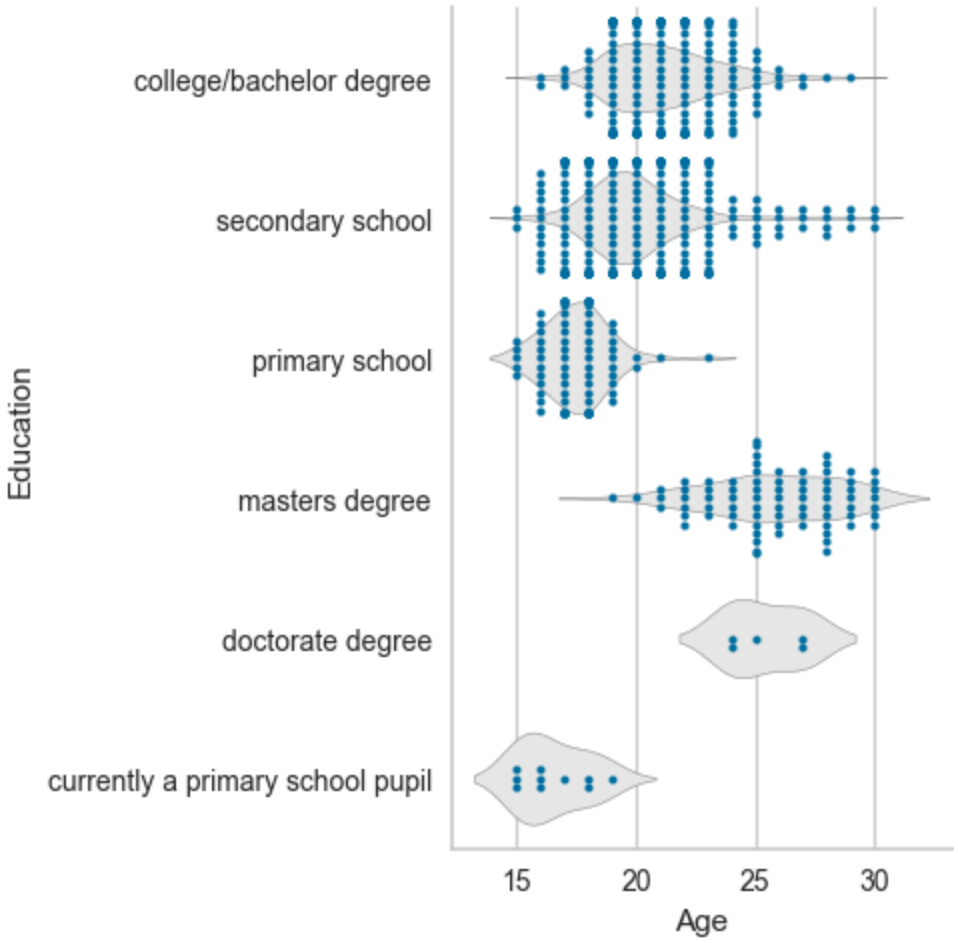
Out[9]:
```
([0, 1, 2, 3, 4, 5],
 [Text(0, 0, 'college/bachelor degree'),
  Text(1, 0, 'secondary school'),
  Text(2, 0, 'primary school'),
  Text(3, 0, 'masters degree'),
  Text(4, 0, 'doctorate degree'),
  Text(5, 0, 'currently a primary school pupil')])
```



Violin Plot:

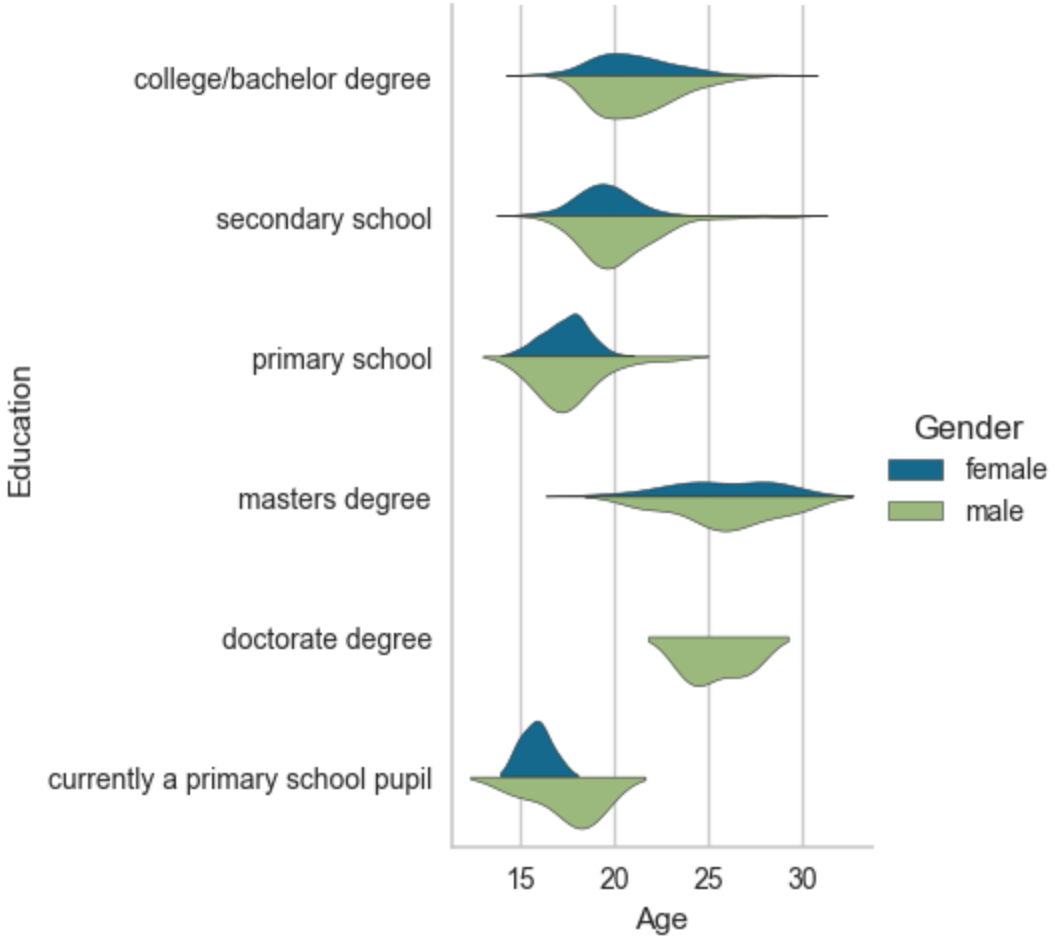
In [10]:
```
sns.catplot(data=data, y="Education", x="Age", kind="violin", color=".9", inner=Non
sns.swarmplot(data=data, y="Education", x="Age", size=3)
```

Out[10]:
```
<Axes: xlabel='Age', ylabel='Education'>
```



In [11]:
```
sns.catplot(data=data, y="Education", x="Age", hue='Gender', kind="violin",inner=No
```

Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x15ccc919c70>
```

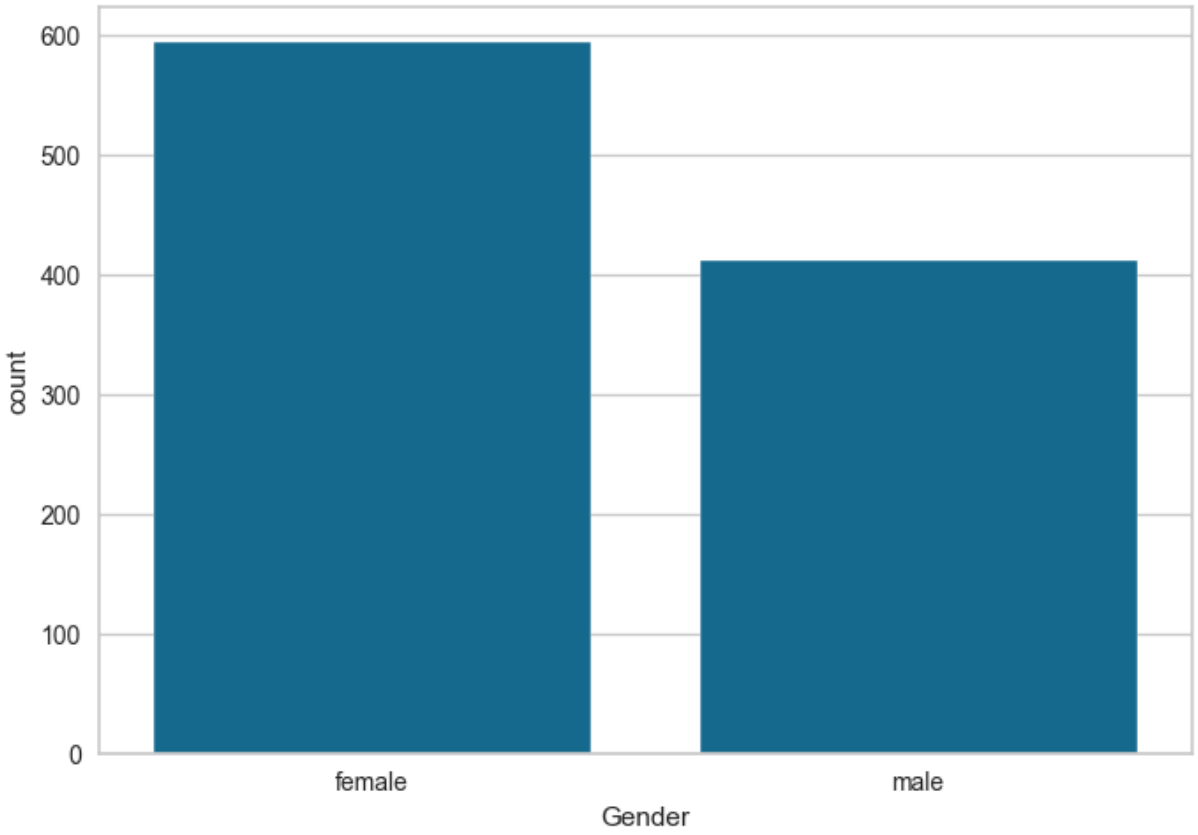


If the level of measurement of the target variable is at least **nominal**:

- Count Plot

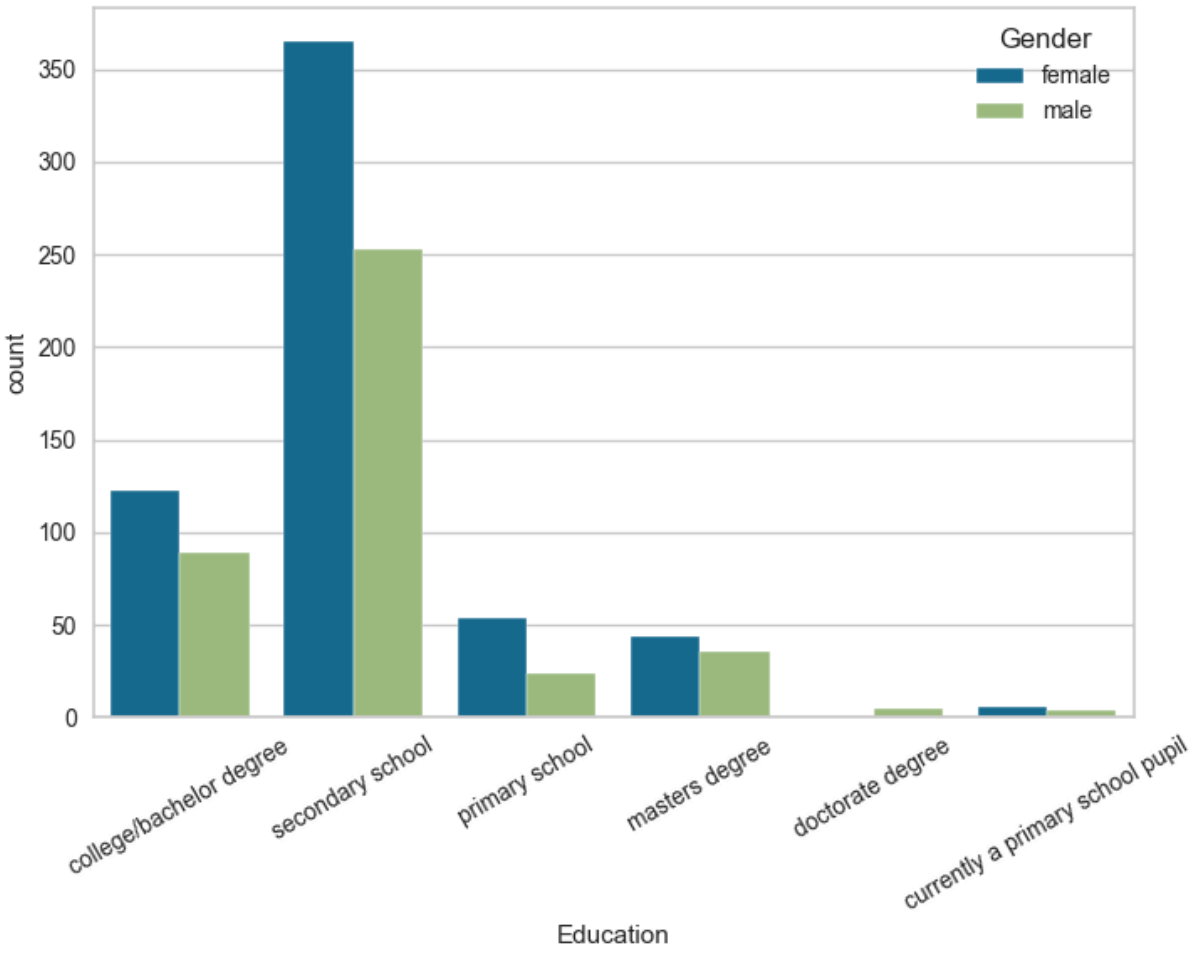
In [12]:
```
sns.countplot(x=data['Gender'])
```

Out[12]:
```
<Axes: xlabel='Gender', ylabel='count'>
```



In [13]:
```
sns.countplot(x=data['Education'],hue=data['Gender'])
plt.xticks(rotation=30)
```

Out[13]:
```
([0, 1, 2, 3, 4, 5],
 [Text(0, 0, 'college/bachelor degree'),
  Text(1, 0, 'secondary school'),
  Text(2, 0, 'primary school'),
  Text(3, 0, 'masters degree'),
  Text(4, 0, 'doctorate degree'),
  Text(5, 0, 'currently a primary school pupil')])
```



1.1.2 Statstics

In [14]:
```
data_des=data[['Gender','Movies','Age']]
data_des=data_des.dropna()
```

In [15]:
```
des=describe.data(data_des,ordinal=['Movies'],nominal=['Gender'])
```

In [16]:
```
des.table()
```

Out[16]:

	Age
count	992.000000
mean	20.423387
std	2.808409
min	15.000000
25%	19.000000
50%	20.000000
75%	22.000000
max	30.000000

In [17]:
```
des.table(show='ordinal')
```

Out[17]:

	Movies
count	992.0
categories	5.0
iqr	1.0
min	1.0
25%	4.0
50%	5.0
75%	5.0
max	5.0

In [18]:
```
des.table(show='nominal')
```

Out[18]:

	Gender
count	992
mode	female
categories	2
least freq	male(40.93%)
most freq	female(59.07%)

Section 1.2: Test Methods for Parameters

1.2.1 Preliminaries

Types of Tests

1. **One-sample Tests:** Testing one sample (one group) against a prespecified value of the parameter.

   Examples:
   - Is there a change in the average IQ of students (IQ so far: μ_0=101)?
   - Is there a decrease in average yearly sales of a company (so far: μ_0=41'000)?

   Hypotheses for test on mean:
   - Two-sided: $H0 : mean = \mu_0$ and $HA : mean \neq \mu_0$
   - Left-sided: $H0 : mean \geq \mu_0$ and $HA : mean < \mu_0$ (for right-sided just reverse the inequality signs)

2. **Two-sample Tests:** Comparing two samples (two groups).

   Examples:
   - Is there a difference in the average income of men (X) and women (Y)?
   - Are the number of burglaries in homes with alarm devices (X) lower than those without (Y) ?

   Hypotheses for test on mean difference:
   - Two-sided: $H0 : meanX - meanY = \mu_0$ and $HA : meanX - meanY \neq \mu_0$
   - Left-sided: $H0 : meanX - meanY \geq \mu_0$ and $HA : meanX - meanY < \mu_0$ (for right-sided just reverse the inequality signs)

   (often μ_0=difference=0)

Steps when running a test

1. Step: Formulate Research Question and Hypotheses
2. Step: Data Preprocessing (Slice Data,Remove NaN,Encoding)
3. Step: Pre-analyses
4. Step: Check Requirements
5. Step: Run and Interprete Test

We will explain these steps with the help of a **Two-sample t-Test on mean difference**.

Step 1: Research Question: Are men on average taller than women?

Hypotheses: $H0 : mean\ \ height\ \ men \leq mean\ \ height\ \ women$ and
$HA : mean\ \ height\ \ men > mean\ \ height\ \ women$

1.2.2 Data Preprocessing

Slice Data

Keep only those columns you need.

In [19]:
```
data_ttest=data[['Height','Gender']]
```

In [20]:
```
data_ttest.head(2)
```

Out[20]:

	Height	Gender
0	163.0	female
1	163.0	female

Remove Nan

NaN = not a number = missing values (must be removed)

In [21]:
```
nan=dataprep.nan(data_ttest)
```

analysis:

In [22]:
```
nan.analysis
```

Out[22]:

	Column	Missing Values
Analysis Missing Values	Height	20
	Gender	6
Number of Rows with NaNs	25	

drop rows with missing values:

In [23]:
```
data_ttest2=nan.drop
```

In [24]:
```
data_ttest.shape
```

Out[24]:
```
(1010, 2)
```

In [25]:
```
data_ttest2.shape
```

Out[25]:
```
(985, 2)
```

Encoding

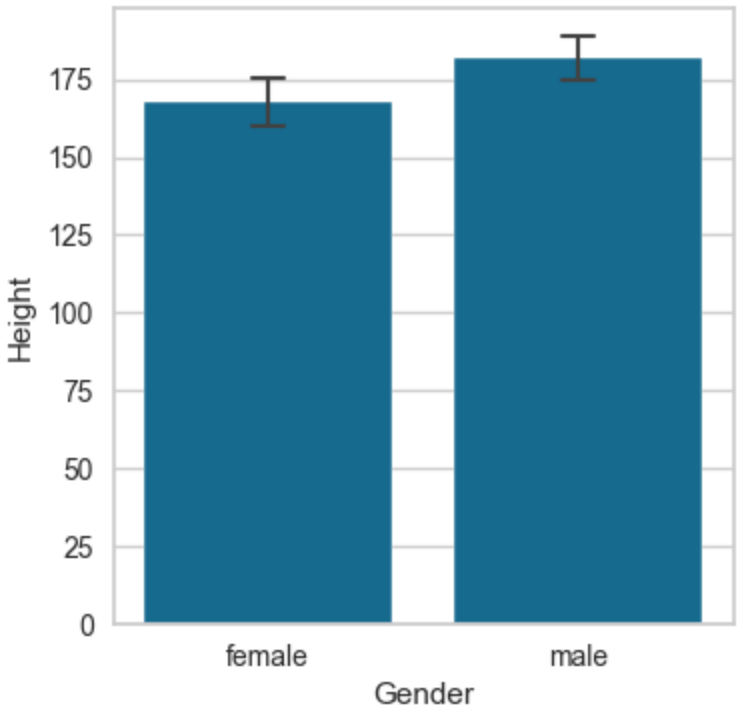
Not necessary -> gender is the grouping variable and height is quantitative.

1.2.3 Pre-analyses

Graphical (here Barplot because height is quantitative)

In [26]:
```
plt.figure(figsize=(4,4))
sns.barplot(x='Gender',y='Height',data=data_ttest2,ci='sd',capsize=.1,errwidth=1.5)
```

Out[26]:
```
<Axes: xlabel='Gender', ylabel='Height'>
```



Groupwise Descriptive Statistics

In [27]:
```
data_ttest2.groupby('Gender').describe().round(3)
```

Out[27]:

	Height							
	count	mean	std	min	25%	50%	75%	max
Gender								
female	580.0	167.771	7.520	62.0	164.0	168.0	172.0	186.0
male	405.0	181.758	6.965	159.0	178.0	182.0	186.0	203.0

1.2.4 Check Requirements

Assumptions t-Test

- **T1. Quantitative dependent variable.**
- **T2. Normality.** The population(s) follow a normal distribution.
- **T3. Independence.** The measurements within and between groups are independent.
- **T4. No outliers.**

- **T5. Binary grouping variable.** There are exactly two groups to compare. [*]
- **T6. Homoscedasticity.** Variance homogeneity: variance group 1 = variance group 2. [*]

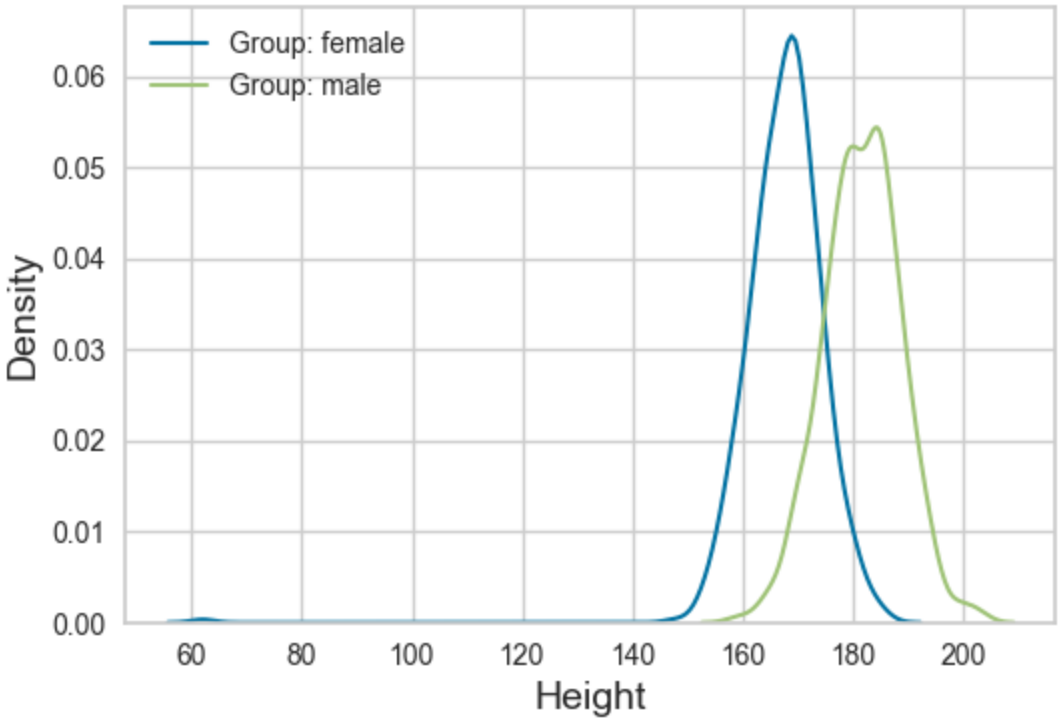
[*] Only for two-sample t-test.

T1. Dependent variable is quantitative

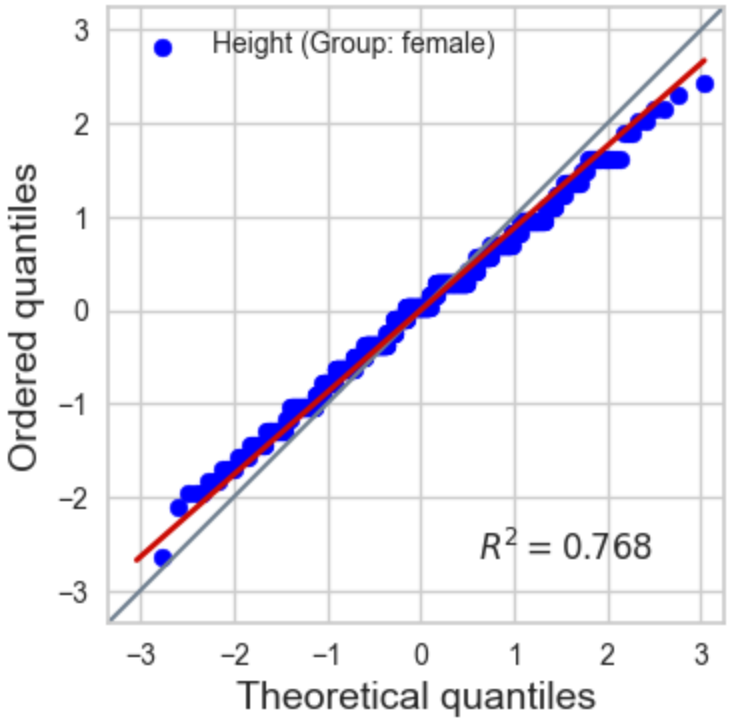
Height is a quantitative variable, so this is true.

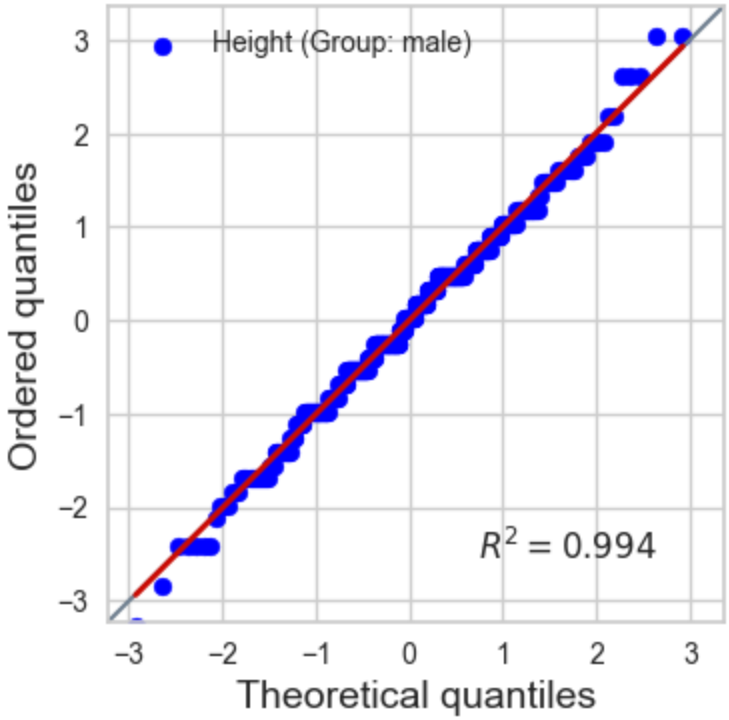
T2. Populations follow a normal distribution

In [28]:
```
plots.dist(data=data_ttest2,var='Height',groupvar='Gender',fig=[6,4],labelsize=14,t
```



In [29]:
```
plots.qq(data=data_ttest2,var='Height',groupvar='Gender',fig=[6,4],labelsize=14,tic
```





T3. Independent measurements

- The measurements within groups are independent if no subject submitted more than once her height.
- The measurements between groups are independent if the height of women and men are independent. This should be the case as long as there are not too many genetic dependencies in the sample.

T4. No Outliers

What are outliers?

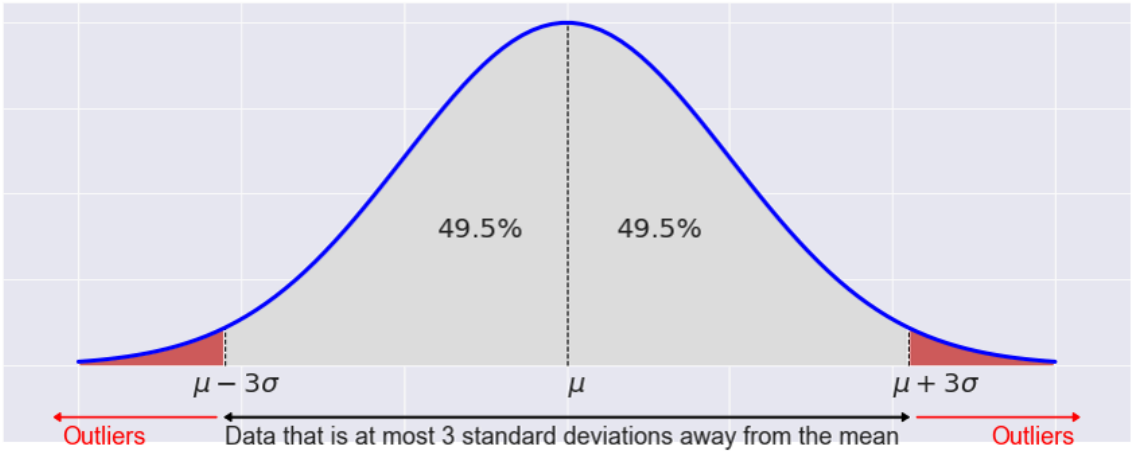
- an outlier is a data point that differs significantly from other observations (see)

What should we do with outliers?

- If outliers represent natural variations in the population they should be left. Otherwise, if they are, e.g., due to measurement errors, they should be removed.
- Statistical methods can tell you which data points are likely to be outliers but not if they must be removed.
- Univariate outlier detection methods include zscore, iqr and mad.

Method zscore:

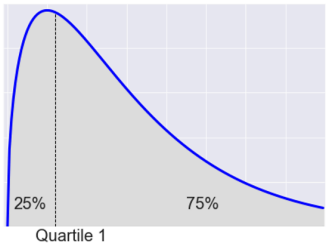
An observation is a potential outlier if it is more than 3 standard deviations away from the mean.

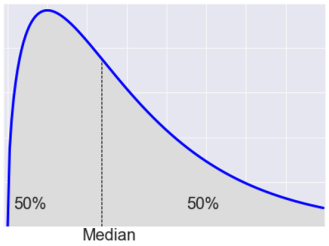


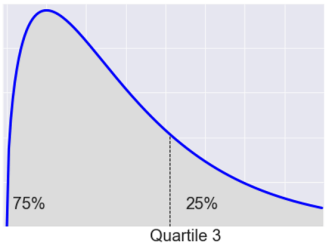
Method interquartile range (iqr)

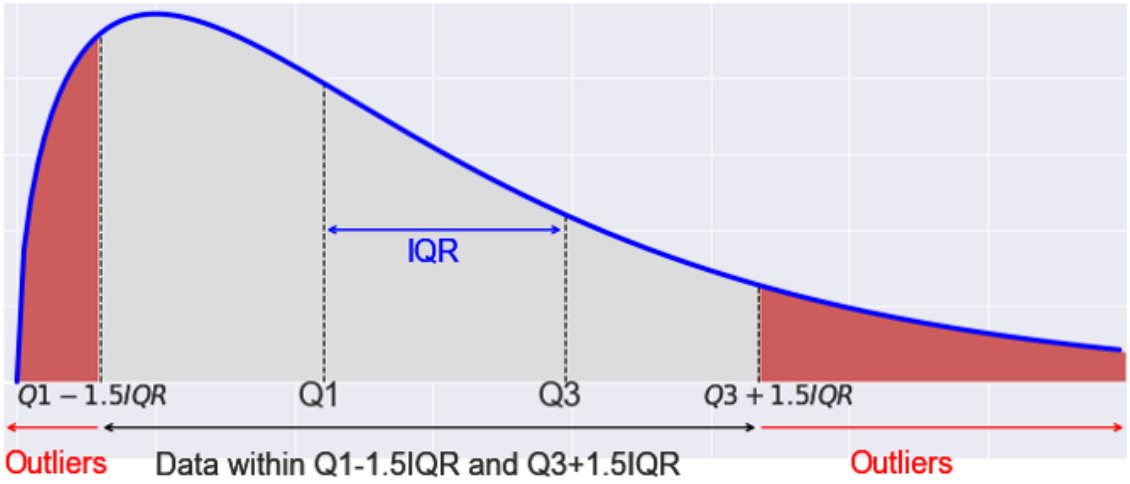
interquartile range = q3(quartile 3) - q1(quartile 1)

An observation is a potential outlier if it is < q1 - 1.5iqr or > q3 + 1.5 iqr









Method median absolute deviation (mad)

mad = median of the absolute value of the differences between observation and data median (= median(|observation - M|))

An observation is a potential outlier if it is < M - 2.24mad or > M + 2.24mad

Outlier Analysis

seperate data set according to groups:

In [30]:
```
groupdata=dataprep.group_sep(data=data_ttest2,groupvar='Gender')
```

In [31]:
```
out_female=outlier.univariate(groupdata[0]['Height'])
out_male=outlier.univariate(groupdata[1]['Height'])
```

In [32]:
```
out_female.analysis
```

Out[32]:

	method	pot. outlier	proportion
extreme value	zscore	1	0.17%
analysis	iqr	23	3.97%
	mad	90	15.52%
	E[ND] (>3 std from mean)	1	0.27%

In [33]:
```
out_male.analysis
```

Out[33]:

	method	pot. outlier	proportion
extreme value	zscore	3	0.74%
analysis	iqr	25	6.17%
	mad	79	19.51%
	E[ND] (>3 std from mean)	1	0.27%

Iqr and mad identify way too many data points as potential outliers. These leaves us with method 'zscore'. Let's have a look at the potential outliers:

In [34]:
```
groupdata[0].loc[out_female.show(method='zscore')]
```

Out[34]:

	Height	Gender
676	62.0	female

Height of 62 cm seems to be a measurement error that needs to be removed.

In [35]:
```
data_ttest3=data_ttest2.drop(out_female.show(method='zscore'))
```

In [36]:
```
groupdata[1].loc[out_male.show(method='zscore')]
```

Out[36]:

	Height	Gender
85	159.0	male
547	203.0	male
799	203.0	male

These heights are not necessarily measurement erros.

As a further robustness test, we run the Tietjen-Moore test to test if it is likely that there are 3 or 2 outliers in the data.
Reference: Tietjen and Moore (1972): Some Grubbs-Type Statistics for the Detection of Outliers, Technometrics, 14, pp. 583-597

In [37]:
```
outliers_tietjen(groupdata[1]['Height'],k=3,hypo=True)
```

Out[37]:
```
False
```

In [38]:
```
outliers_tietjen(groupdata[1]['Height'],k=2,hypo=True)
```

Out[38]:
```
False
```

There seem to be no outliers for male.

T5. Independent variable is binary
True. In this data set gender is binary.

T6. Homogeneity (equal variances in groups)
We test this requirement by running a Levene's test of equal variances:

H0: The variances of all groups are equal and HA: The variances of at least two groups differ

In [39]:
```
tests.equal_var.levene(data=data_ttest3,var='Height',groupvar='Gender',rem=True)
```

Out[39]:

	var	group		f	dof1	dof2	p-val	remark
Levenes Test	Height	Gender	Mean	8.879683	1	982	0.002955	for symmetric, moderate-tailed distributions
of Equal Variances			Median	8.720196	1	982	0.003222	for skewed (non-normal) distributions
			Trimmed	10.901975	1	982	0.000999	for heavy-tailed distributions

We may take the Levene's test based on mean. Here the p-value is 0.3% < 5%. Hence, we can reject the null hypothesis.

Remark: degrees of freedom of Levene's Test

- dof1 = number of groups (k) - 1
- dof2 = number of observations (n) - k

1.2.5 Run Test and Interprete Result

In [40]:
```
tests.t.two_sample(data=data_ttest3,var='Height',groupvar='Gender',alternative='les
```

Out[40]:

	var	group	mean	variances	t	dof	alternative	p-val	CI95%	co
Two-Sample	Height	female	167.9534	equal	-32.9292	982.000	less	0.0	[-inf, -13.11]	2.
t-Test		male	181.7580	unequal	-32.1730	794.406		0.0	[-inf, -13.1]	2.

The p-value is 0% < 5%. Hence, we can reject the null hypothesis and have evidence that men are on average taller than women. The power is 1, so the hypothesis test is very good at detecting a false null hypothesis

Remark: degrees of freedom of t-test. When computing a mean, we lose one degree of freedmom. Hence:

- one-sample t-test: dof = n - 1
- two-sample t-test: dof = n - 2

The degrees of freedom must always be taken from the first row. The second row shows the Satterthwaite corrected degrees of freedom for the case of unequal variances.

Effect size

$$Cohen's\ d = \frac{mean1 - mean2}{pooled\ standard\ deviation}$$

- Cohen's d suggests a large effect size:

Cohen´s d effect size	Interpretation	Differences in SD
d=.0–.19	Trivial effect	<1/5 from a SD
d=.20	Small effect	1/5 from a SD
d=.50	Medium effect	1/2 from a SD
d=.80 or higher	Large effect	8/10 from a SD

1.2.6 What to do if the requirements of the t-test are violated?

You get less robust results. Sometimes this could mean that you must use other tests. For example:

- If the dependent variable is ordinal or the data is not normally distributed → use a sign- or Mann-Whitney U test
- If you have got a paired sample (no between group independence) → use a paired t-test or Wilcoxon signed-rank test

Overview:

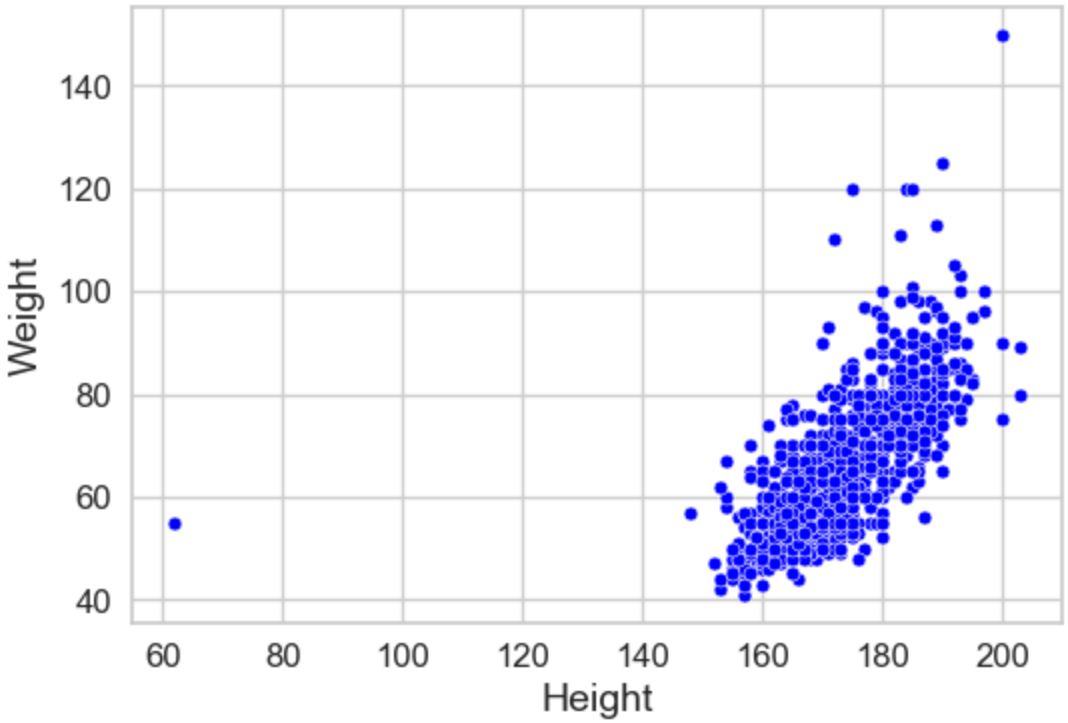
Level of Measurement	Test for	One-Sample Test	Two-Sample Test	
			Independent Samples	Dependent Samples
Quantitative	Mean	t-test	t-Test Welch's t-test if variances not equal	paired t-test
Ordinal	Median	sign-test	Mann–Whitney U test	Wilcoxon signed-rank test

Section 1.3: Test Methods for Correlations and Associations

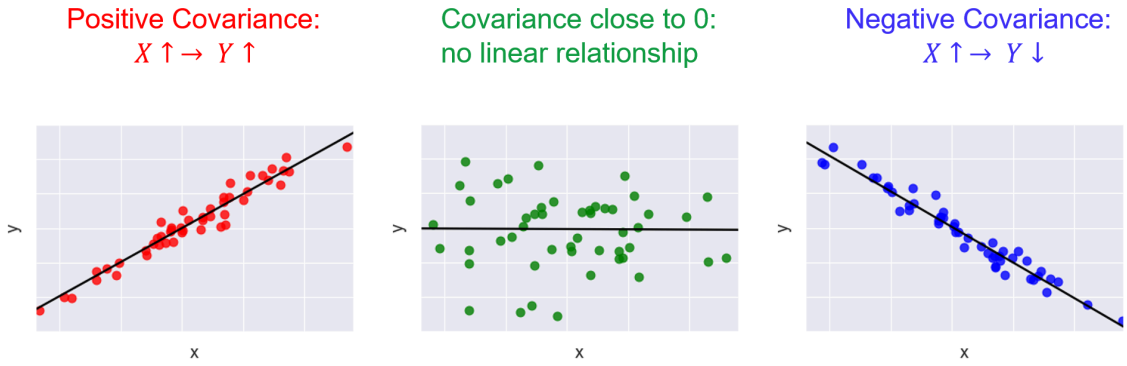
1.3.1 Correlation Quantitative vs Quantitative Variable

Example: Does weight increase in height?

```
In [41]: plots.scatter(data['Height'],data['Weight'],fig=[6,4],dotsize=25,labelsize=14,ticks
```



The covariance measures the linear relationship between two variables:



$$cov(X,Y) = \frac{(x_1 - mean_x)(y_1 - mean_y) + \ldots + (x_n - mean_x)(y_n - mean_y)}{n-1}$$

Unfortunately, the covariance depends on the unit of measurement:

- height in cm:

In [42]:
```
data['Height'].cov(data['Weight'])
```

Out[42]: 94.62917179129056

- height in m:

In [43]:
```
height_in_m=data['Height']/100
```

In [44]:
```
height_in_m.cov(data['Weight'])
```

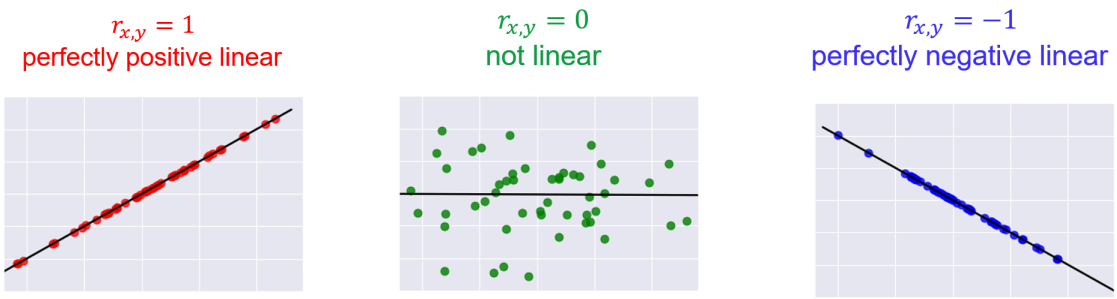
Out[44]:
```
0.9462917179129056
```

Therefore, we use a standardized version of the covariance: Pearson's correlation coefficient r:

$$r = \frac{cov(X,Y)}{standdev_X \cdot standdev_Y}$$

where the covariance of the variables is divided by the product of their standard deviations.

The correlation coeffient can only take on values between -1 and +1, where -1 indicates a perfect negative linear relationship and +1 a perfect positive linear relationship:



Test the correlation

Step 1. Research Question and Hypotheses
Height and weight are positively correlated: $H0 : r \leq 0$ and $HA : r > 0$

Step 2. Data preprocessing

In [45]:
```
data_corr=data[['Height','Weight']] #slice the data
```

In [46]:
```
nan=dataprep.nan(data_corr) #remove nan
```

In [47]:
```
nan.analysis
```

Out[47]:

	Column	Missing Values
Analysis Missing Values	Height	20
	Weight	20
Number of Rows with NaNs	30	

In [48]:
```
data_corr=nan.drop
```

Step 3. Pre-analyses

correlation matrix

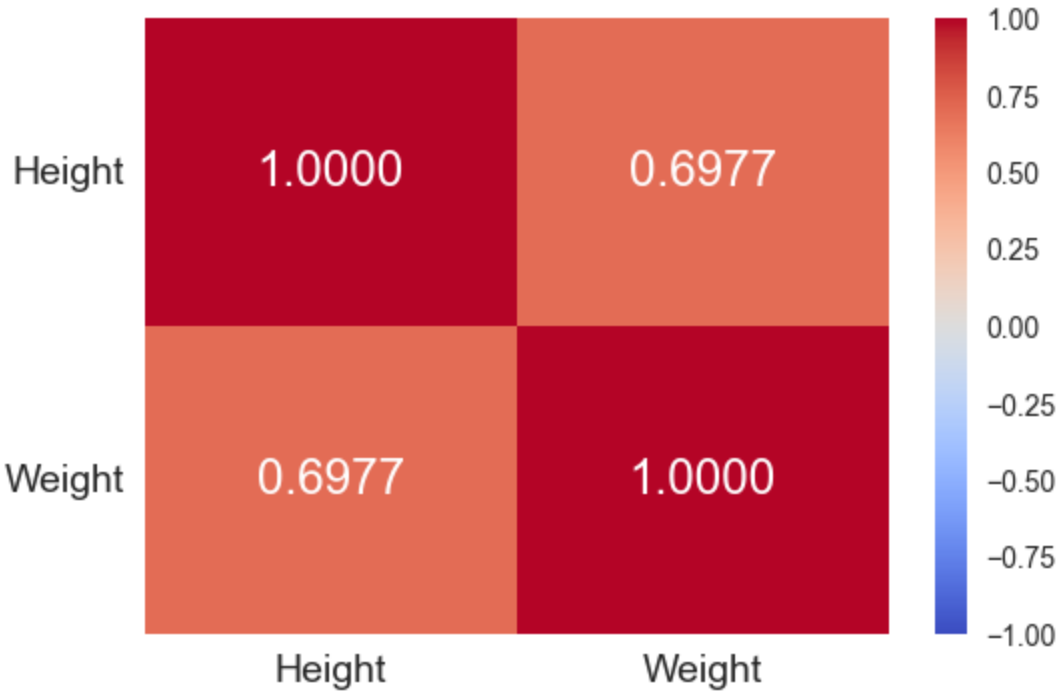
In [49]:
```
cm=describe.corrmat(data_corr)
```

In [50]:
```
cm.table.round(3)
```

Out[50]:

	Height	Weight
Height	1.000	0.698
Weight	0.698	1.000

In [51]:
```
cm.heatmap(fig=[6,4],nsize=35,lsize=14)
```



In [52]:
```
cm2=describe.corrmat(data_corr,utri=False)
```

In [53]:
```
cm2.table
```

Out[53]:

	Height	Weight
Height		
Weight	0.6977	

Step 4. Check Requirements.

Assumptions Pearson Correlation Test

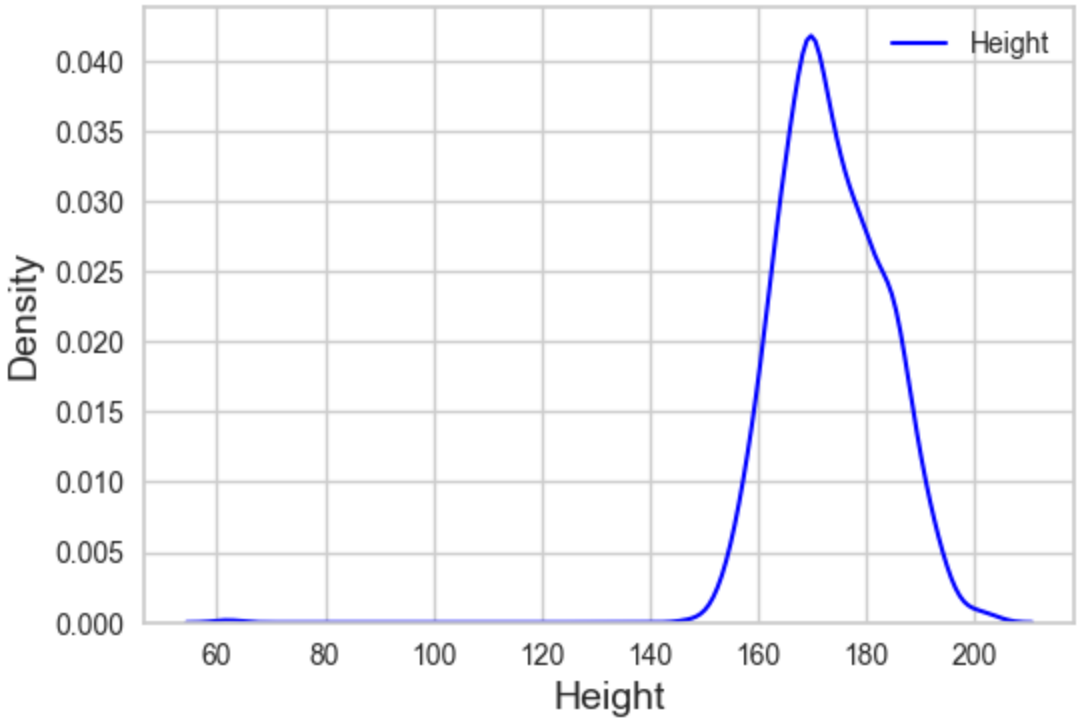
- **PCC1. Quantitative Variables.**

- **PCC2. Normality.** The populations follow a normal distribution.
- **PCC3. No Outliers.**
- **PCC4. Independence.** The observations within a variable are independent.

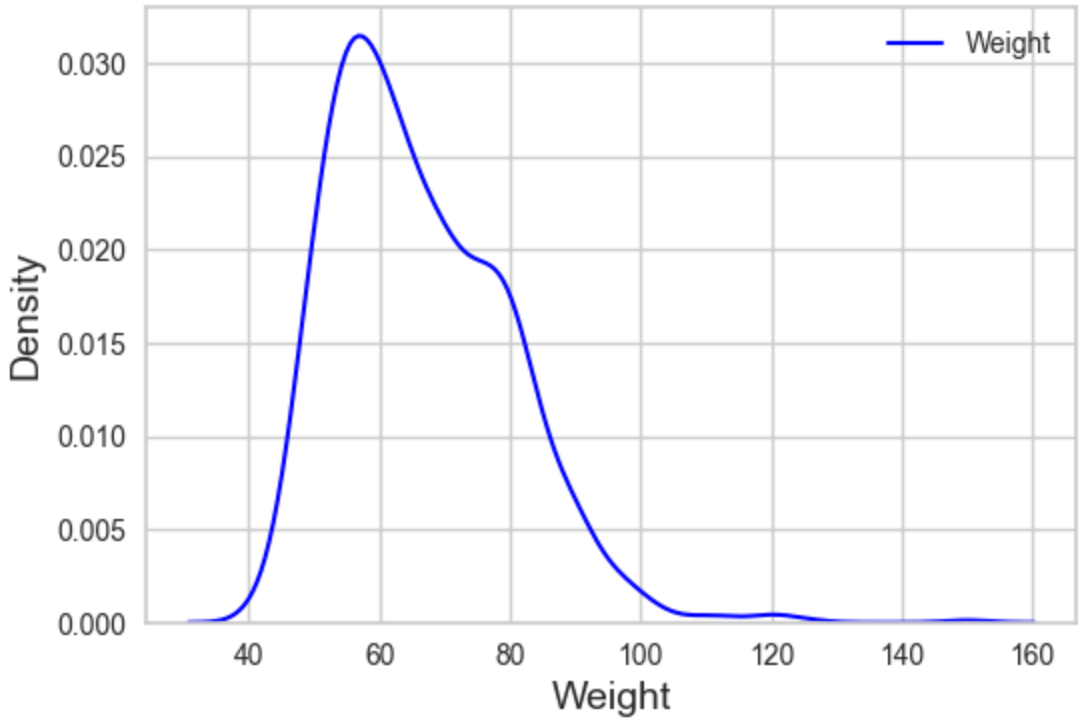
Height and weight are quantitative (PCC1 true) and we may assume that no subject submitted twice as well as no strong family connections (PCC4 true).

PCC2. Bivariate normal.

In [54]:
```
plots.dist(data_corr['Height'],fig=[6,4],labelsize=14,ticksize=10,legsize=10,linewi
```



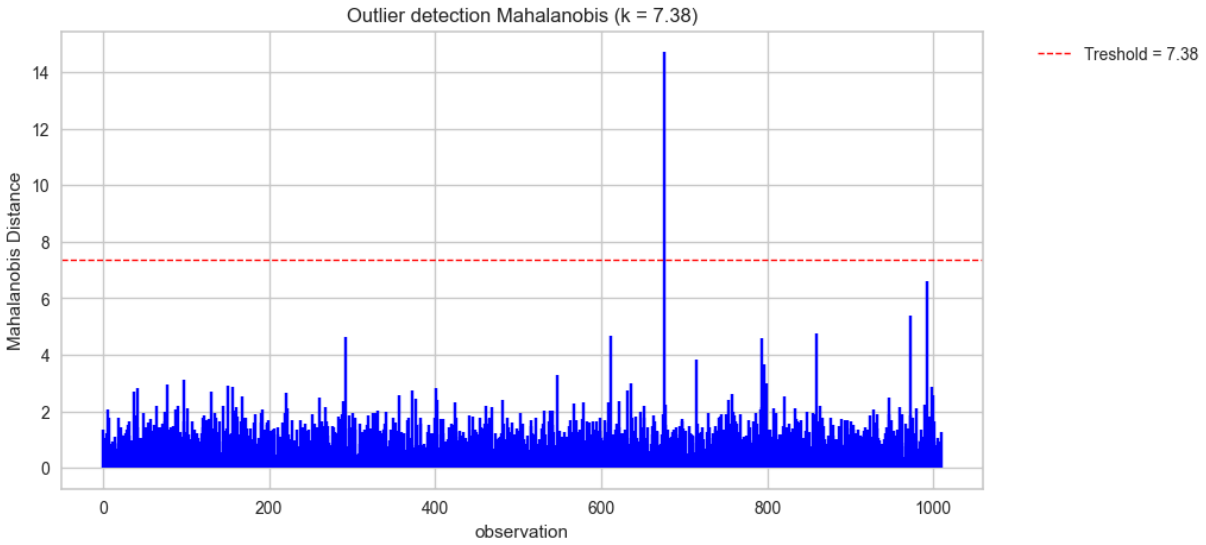
In [55]:
```
plots.dist(data_corr['Weight'],fig=[6,4],labelsize=14,ticksize=10,legsize=10,linewi
```



-> seems to be fairly normal.

PCC3. No outlier.

In [56]:
```
plots.outlier(x=data_corr[['Height']],y=data_corr['Weight'],method='Mahalanobis',dt
```



Step 5. Run test and interprete

In [57]:
```
tests.correlation.simple(data_corr['Height'],data_corr['Weight'],alternative='great
```

Out[57]:

	var1	var2	n	r (pearson)	CI95%	alternative	p-val	BF10	power
pearson Test of Correlation	Height	Weight	980	0.6977	[0.67, 1.0]	greater	0.0	1.888e+140	1.0

The p-value is 0% < 5%. Hence, we can reject the null hypothesis and have evidence that there is a positive correlation. The power is 1, so the hypothesis test is very good at detecting a false null hypothesis

Effect size

- r = 0.7346 suggests a high correlation

Interpretation	Correlation value
Small correlation	0.10 to 0.29
Medium correlation	0.30 to 0.49
Large correlation	0.50 to 1.0

Furthermore $r^2 \approx 54\%$ means that the variables share 54% of their variance.

1.3.2 Correlation Ordinal vs Quantitative/Ordinal Variable

In case of ordinal variables, we cannot compute the covariance as we cannot compute a mean. Correlation coefficients are determined using **rank-based** approaches that order the data from lowest to highest and assign a rank to each observation depending on its position. Popular rank-based correlation coefficients are:

- Spearman's ρ, Kendall's τ, Goodman and Kruskal's γ

Spearman's ρ
This coefficient works just like Pearson's r with the difference that it computes the covariance and the standard deviations with respect to the ranks instead of the values of the variable:

$$r^S = \frac{cov(rank(X), rank(Y))}{std(rank(X)) \cdot std(rank(Y))}$$

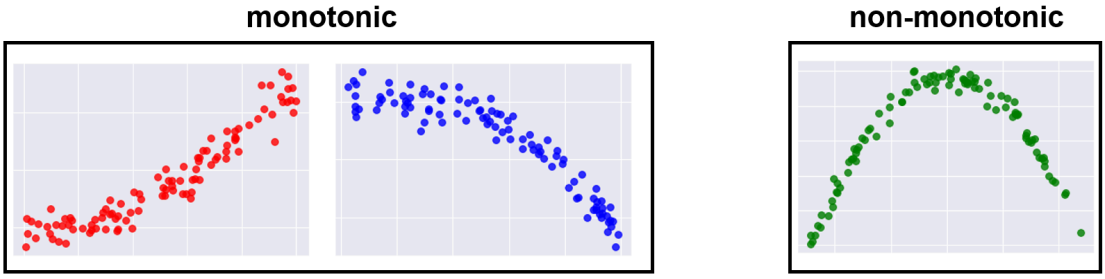
Kendall's τ, Goodman and Kruskal's γ

These coefficients are based on the number of concordant and discordant pairs in a data set. Given two variables X and Y, two pairs of observations (x_i, y_i) and (x_j, y_j) are

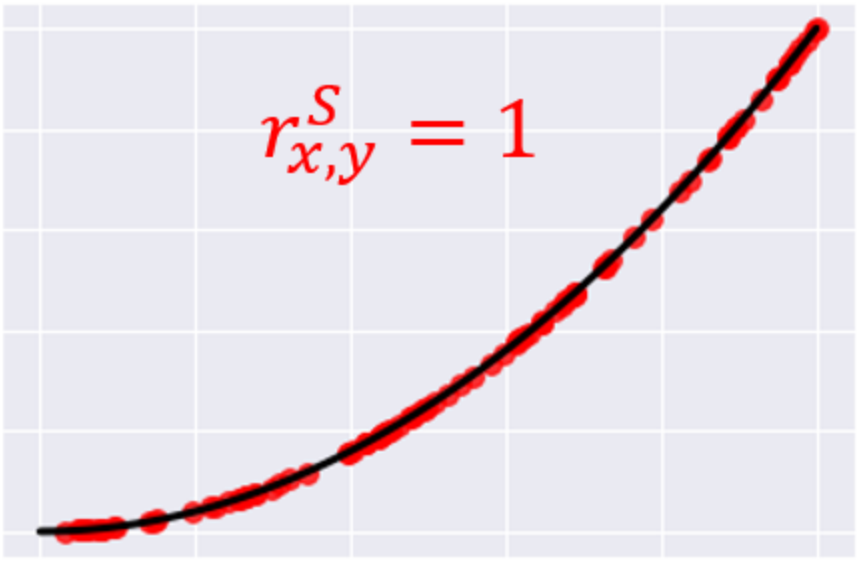
- *concordant* if $x_i > x_j$ and $y_i > y_j$ or if $x_i *discordant* if $x_i > x_j$ and $y_i x_j and y_i
  Examples:
  - (1,3) and (6,9) are concordant
  - (3,1) and (6,9) are discordant

Relationship between Spearman and Kendall coefficient: $Kendall \approx 0.7 \cdot Spearman$

Rank-based approaches identify more general monotonic relationships:



A coeffient = 1 indicates a perfectly positive monotonic relationship:



Correlation Test

Step 1. Research Question:
Does preference for classical music increase in education? $HA : r^S \leq 0$ and $HA : r^S > 0$

Step 2. Data Preprocessing

In [58]:
```
data_rank=data[['Education','Classical music']] #slice data
data_rank=data_rank.dropna() #drop NaNs
```

Education is not encoded yet. Thus, we need to encode it:

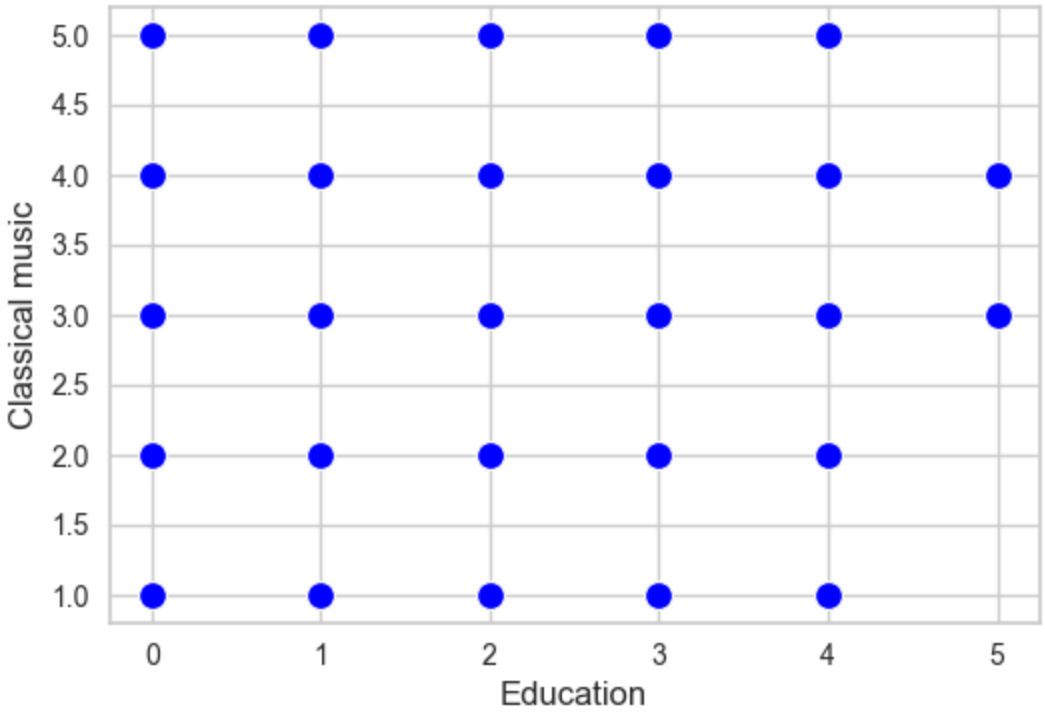
In [59]:
```
enc=dataprep.encoder(order={'Education':['currently a primary school pupil','second
                                         'college/bachelor degree', 'masters degree
data_rank2=enc.fit_transform(data_rank)
```

In [60]:
```
data_rank2.Education.unique()
```

Out[60]:
```
array([3., 1., 2., 4., 5., 0.])
```

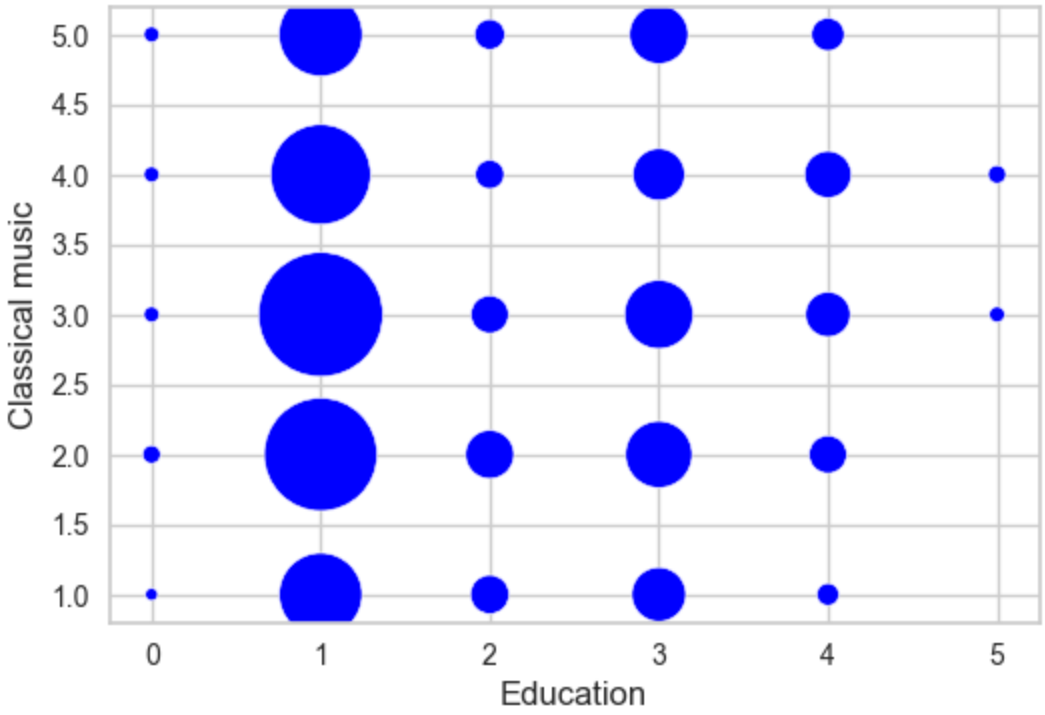
Step 3. Pre-analyses

In [61]:
```
plots.scatter(data_rank2['Education'],data_rank2['Classical music'],fig=[6,4],ticks
```



This is not really informative -> we need to enable the ordinal option.

In [62]:
```
plots.scatter(data_rank2['Education'],data_rank2['Classical music'],fig=[6,4],ticks
```



Does rather look like a non-monotonic relationship.

Step 4. Check assumptions

Assumptions Rank-based Correlation Tests

- **SK1. Ordinal.** Both variables are at least ordinal.
- **SK2. Independence.** The observations within a variable are independent.

Education and preference for classical music are ordinal. Furthermore, we may assume independence.

Step 5. Run test and interprete.

In [63]:
```
tests.correlation.simple(data_rank2['Education'],data_rank2['Classical music'],alte
```

Out[63]:

	var1	var2	n	r (spearman)	CI95%	alternative	p-val	power
spearman Test of Correlation	Education	Classical music	1002	0.0276	[-0.02, 1.0]	greater	0.1918	0.2197

In [64]:
```
tests.correlation.simple(data_rank2['Education'],data_rank2['Classical music'],alte
```

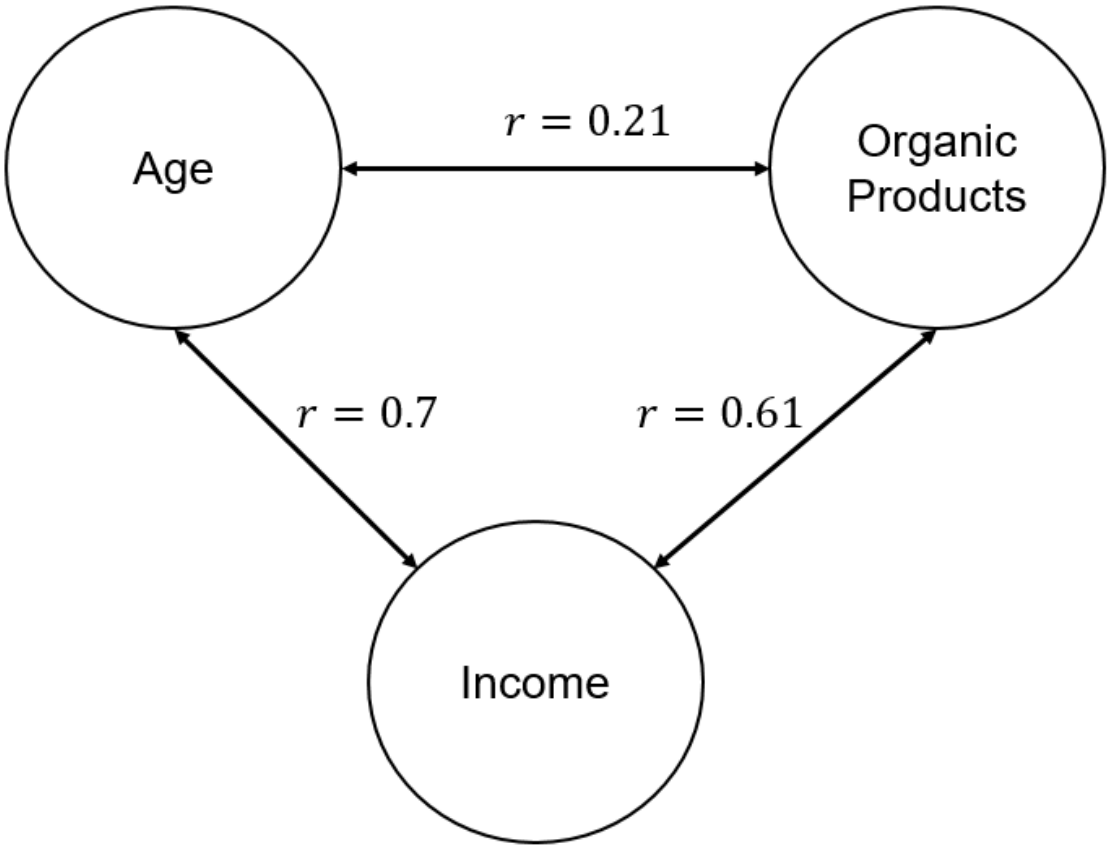
Out[64]:

	var1	var2	n	r (kendall)	CI95%	alternative	p-val	power
kendall Test of Correlation	Education	Classical music	1002	0.0223	[-0.03, 1.0]	greater	0.2401	0.174

The p-value of both coefficients is below 5%. Hence, we may reject the null hypothesis. There is evidence that education and preference for classical music is positively correlated. However, the power is at 0.2 to 0.25, which indicates that the tests are not good at detecting a false null hypothesis

1.3.3 Partial Correlations

When testing correlations, we need to take into account potential confounding variables. Say, we would like to test if age is correlated with buying organic products. Then, we also have to take into account that age is correlated with income, which might be also correlated with buying (the more expensive) organic products:



The correlation of 0.21 might be partly due the positive correlation between age and income. Hence, we need to eliminate the effect from income. Partial correlation analysis offers a way to do this.

Partial correlation coefficient adjustment:
Let X,Y and Z be three variables. Suppose, we would like to examine the correlation between X and Y while controlling for Z. The adjusted correlation coefficient is then:

$$r_{XY,Z} = \frac{r_{XY} - r_{XZ} \cdot r_{YZ}}{\sqrt{1 - r_{XZ}^2} \cdot \sqrt{1 - \cdot r_{YZ}^2}}$$

In [65]:
```
data_part=data[['Age','Height','Weight']]
data_part=data_part.dropna()
```

In [66]:
```
data_part['Age'].corr(data_part.Weight)
```

Out[66]:
```
0.23708368338501684
```

In [67]:
```
tests.correlation.partial(data=data_part,var1='Age',var2='Height',covar=['Weight'])
```

Out[67]:

	var1	var2	covar	n	r (pearson)	CI95%	alternative	p-val
pearson Partial Correation Test	Age	Height	[Weight]	978	-0.072924	[-0.14, -0.01]	two-sided	0.022637

1.3.4 Association Nominal vs Nominal/Ordinal

In order to test if a nominal and a nominal or ordinal variable are associated, we may use a Test of Independence: χ^2 or an exact Test in case of 2×2 contingency tables.

Hypotheses: **H0: X and Y are independent** and **HA: X and Y are dependent**

Example: Independence Test

Step 1. Research Question and Hypotheses
Does preference for classical music depend on gender?
H0: Gender and Preference are independent and **HA: Gender and Preference are dependent**

Step 2. Data preprocessing

In [68]:
```
data_ind=data[['Gender','Classical music']] #slice data
```

In [69]:
```
nan=dataprep.nan(data_ind) #nan
nan.analysis
```

Out[69]:

	Column	Missing Values
Analysis Missing Values	Gender	6
	Classical music	7
Number of Rows with NaNs	12	

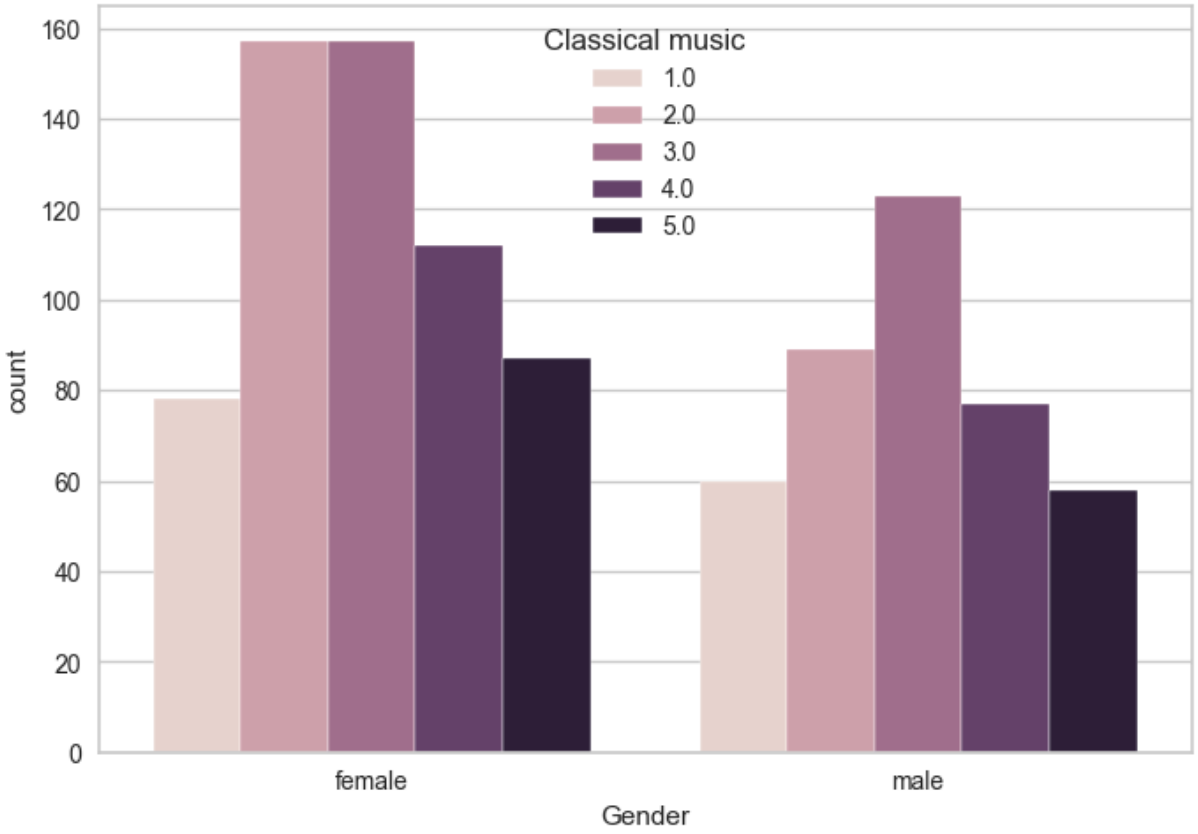
In [70]:
```
data_ind1=nan.drop
```

Step 3. Pre-analyses

Countplot:

In [71]:
```
sns.countplot(x='Gender',hue='Classical music',data=data_ind1)
```

Out[71]:
```
<Axes: xlabel='Gender', ylabel='count'>
```



Contingency tables:

In [72]:
```
x,y=data_ind1['Gender'], data_ind1['Classical music']
```

In [73]:
```
describe.contingency(x,y,show='observed')
```

Out[73]:

Classical music	1.0	2.0	3.0	4.0	5.0
Gender					
female	78	157	157	112	87
male	60	89	123	77	58

In [74]:
```
describe.contingency(x,y,show='expected').round(2)
```

Out[74]:

Classical music	1.0	2.0	3.0	4.0	5.0
Gender					
female	81.72	145.68	165.81	111.92	85.87
male	56.28	100.32	114.19	77.08	59.13

In [75]:
```
describe.contingency(x,y,show='deviations')
```

Out[75]:

Classical music	1.0	2.0	3.0	4.0	5.0
Gender					
female	-5.0%	8.0%	-5.0%	0.0%	1.0%
male	7.0%	-11.0%	8.0%	-0.0%	-2.0%

Step 4. Check assumptions

Assumptions Chi2 Independence Test

- **C1. Categorical Variables.**
- **C2. Large Sample.** Thumb rule: n > 50.
- **C3. Sufficient Expected Frequencies.** All expected frequencies are > 5.
- **C4. Independence.** Observations within variables are independent.

Assumptions Exact Independence Test

- **E1. Binary Categorical Variables.**
- **E2. Independence.**

We need to use χ^2 because we do not have a 2×2 contingency table.

C1. Gender and classical music are both categorical -> True
C2. Sample Size = 998 -> True:

In [76]:
```
len(data_ind1)
```

Out[76]:
```
998
```

C3. Expected frequencies > 5 (see contingency tables) -> True
C4. Independence (within variables) may be assumed to be true.

Step 5. Run test and interprete

Idea of the test: the test compares observed frequencies with the expected frequencies provided independence. If the difference is too large, we may reject the null hypothesis (independence).

Observed frequencies:

Classical music	1.0	2.0	3.0	4.0	5.0
Gender					
female	78	157	157	112	87
male	60	89	123	77	58

Expected frequencies:

Classical music	1.0	2.0	3.0	4.0	5.0
Gender					
female	81.72	145.68	165.81	111.92	85.87
male	56.28	100.32	114.19	77.08	59.13

How to compute expected frequencies?

$$E_{row\ i, column\ j} = \frac{(observations\ row\ i) \times (observations\ column\ j)}{Total\ observations}$$

Example: row 1, column 1:

$$E_{1,1} = \frac{(78+60)\cdot(78+157+157+112+87)}{998} \approx 81.72$$

How to compute the test statistic χ^2?
for a $n \times m$ contingency table:

$$\chi^2 = \frac{(observed_{1,1} - expected_{1,1})^2}{expected_{1,1}} + \cdots + \frac{(observed_{n,m} - expected_{n,m})^2}{expected_{n,m}}$$

In our example:

$$\chi^2 = \frac{(78-81.72)^2}{81.72} + \cdots + \frac{(58-59.13)^2}{59.13} \approx 3.76$$

In [77]:
```
tests.independence.chi2(x,y)
```

Out[77]:

	vars	no. categories	test	chi2	dof	p-val	cramer	power
Chi2 Tests	Gender	2	pearson	3.758539	4.0	0.439669	0.061368	0.301622
of Independence	Classical music	5	cressie-read	3.764047	4.0	0.438879	0.061413	0.302043
			G(log-likelihood)	3.776706	4.0	0.437068	0.061516	0.303009
			freeman-tukey	3.787661	4.0	0.435505	0.061606	0.303846
			mod-log-likelihood	3.799891	4.0	0.433764	0.061705	0.304780
			neyman	3.828268	4.0	0.429746	0.061935	0.306948

We focus on the first (pearson) and the third (G) row both yield a p-value > 5%. Hence, we cannot reject the null hypothesis. However, as the power shows the test is not really good at detecting false null hypothesis.

Effect Size

- Cramer's V is a measure of the effect size (if there is any)

$$Cramer's\ V = \sqrt{\frac{\chi^2/n}{min(r-1, k-1)}},$$

where n = number of observations, r = number of rows and k= number of columns

- In our case: negligible effect size

Value of φ or Cramer's V	Description
.00 and under .10	Negligible association
.10 and under .20	Weak association
.20 and under .40	Moderate association
.40 and under .60	Relatively strong association
.60 and under .80	Strong association
.80 to 1.00	Very strong association

Reference: Rea, L. M., and Parker, R. A. (1992). Designing and conducting survey research. San Francisco: Jossey-Boss.

1.3.5 Summary

Association: Nominal vs quantitative

For nominal vs quantitative variables, we can use the measure eta (η) or a point-biserial

correlation when the nominal variable is binary. However, for analyses nominal vs. quantitative, it is often advisable to apply a regression model.

Which coefficient for which levels of measurement?

		Variable 2		
		Quantitative	Ordinal	Nominal
Variable 1	Quantitative	Pearson Correlation Coefficient		
	Ordinal	Spearman's ρ / Kendall's τ / GK γ	Spearman's ρ / Kendall's τ / GK γ	
	Nominal	Eta η / If nominal is binary: Point-biserial correlation	Cramer's V / If nominal is binary: Rank-biserial correlation	Cramer's V

General correlation matrix:

In [78]:
```
data.columns
```

Out[78]:
```
Index(['Music', 'Slow songs or fast songs', 'Dance', 'Folk', 'Country',
       'Classical music', 'Musical', 'Pop', 'Rock', 'Metal or Hardrock',
       ...
       'Age', 'Height', 'Weight', 'Number of siblings', 'Gender',
       'Left - right handed', 'Education', 'Only child', 'Village - town',
       'House - block of flats'],
      dtype='object', length=150)
```

In [79]:
```
data_gen=data[['Gender','Classical music','Age','Height','Left - right handed']]
nominal=['Gender','Left - right handed']
ordinal=['Classical music']
data_gen=data_gen.dropna()
```

In [80]:
```
cm=describe.corrmat(data_gen,ordinal=ordinal, nominal=nominal,show_nominal=True,utr
```

In [81]:
```
cm.table
```

Out[81]:

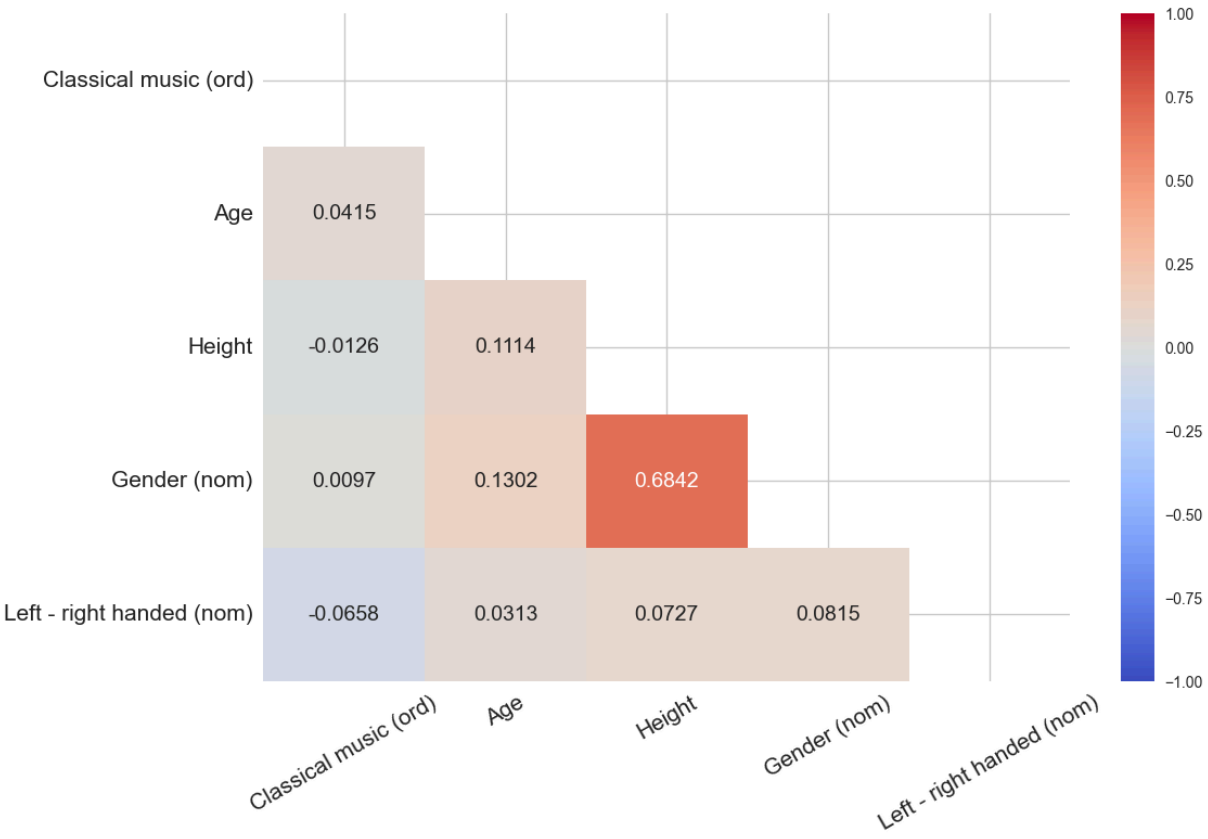
	Classical music (ord)	Age	Height	Gender (nom)	Left - right handed (nom)
Classical music (ord)					
Age	0.0415				
Height	-0.0126	0.1114			
Gender (nom)	0.0097	0.1302	0.6842		
Left - right handed (nom)	-0.0658	0.0313	0.0727	0.0815	

In [82]:
```
cm.coef
```

Out[82]:

	Classical music (ord)	Age	Height	Gender (nom)	Left - right handed (nom)
Classical music (ord)					
Age	spearman				
Height	spearman	pearson			
Gender (nom)	rbc	pbc	pbc		
Left - right handed (nom)	rbc	pbc	pbc	cramer	

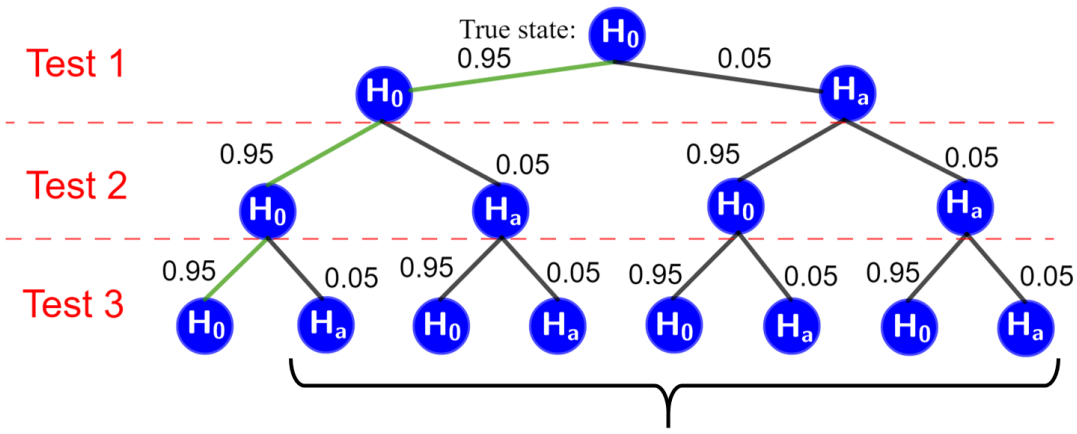
In [83]:
```
cm.heatmap(rotx=30)
```



Section 1.4: Multiple Tests

Repeated tests (e.g., for pairwise comparisons) inflate the family-wise error (α error).

Example: pairwise comparisons between 3 groups

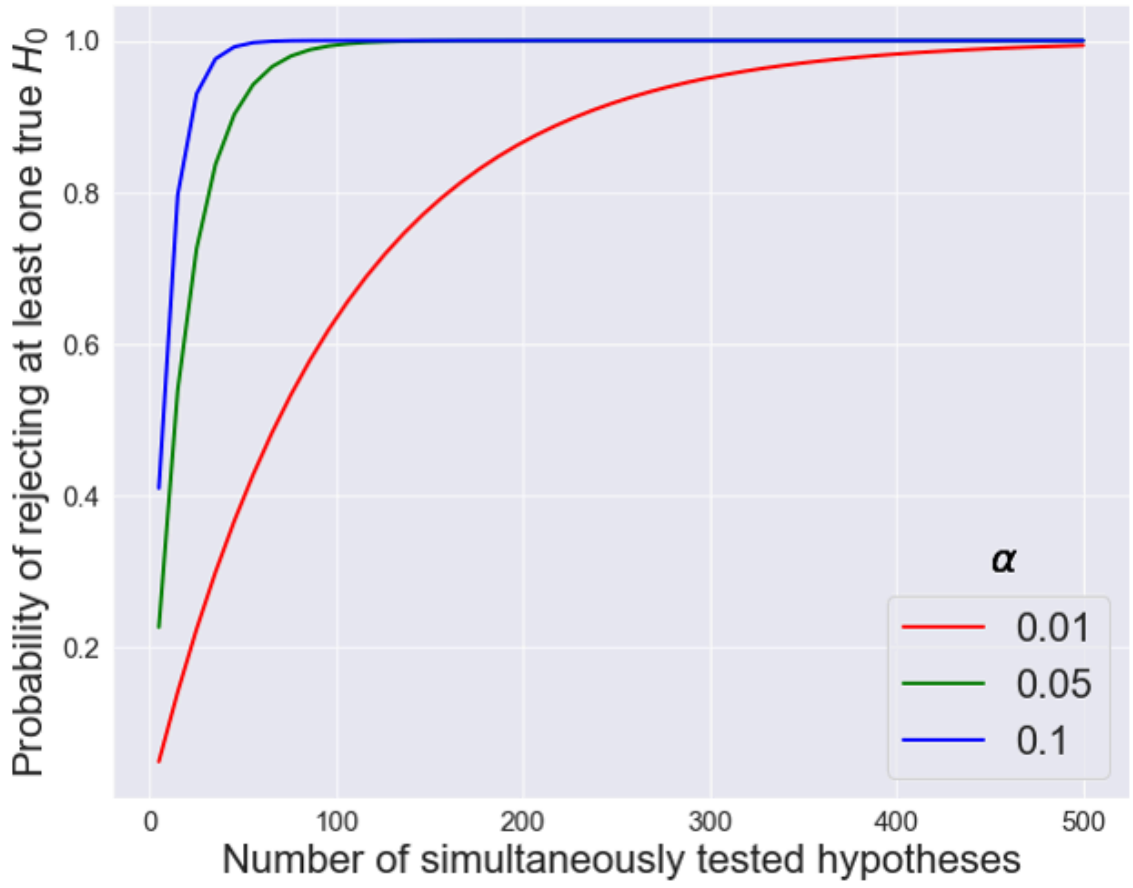


$P(\textit{all tests are correct}) = 0.95^3$

$P(\textit{at least 1 test is not correct}) = 1 - 0.95^3 \approx 14.3\%$

For each test, we set the probability for a false-positive test result (α) to 5%. Triple testing inflates this probability to 14.3%. In general:

$$Family-wise\ error\ rate \leq 1-(1-\alpha)^{number\ of\ simultaneous\ tests}$$

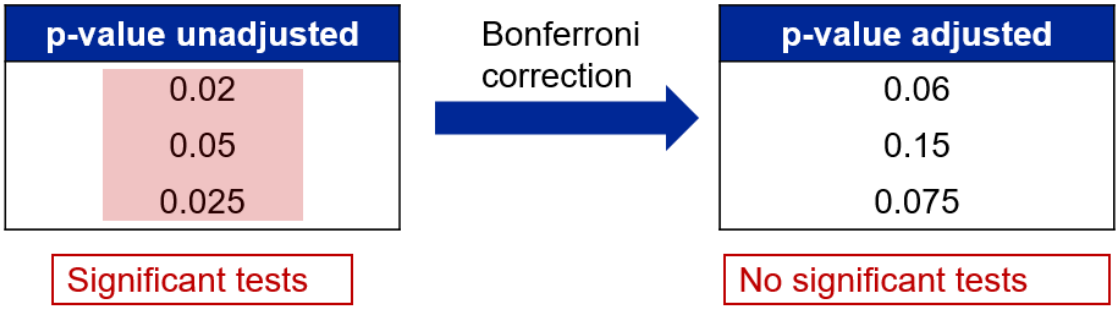


We must take into account the inflated family-wise error. One strategy is to adjust the p-value. Popular methods are:

1. Bonferroni (bonf)
2. Sidak (sidak)
3. Holm-Bonferroni (holm)
4. Benjamini Hochberg (bh)

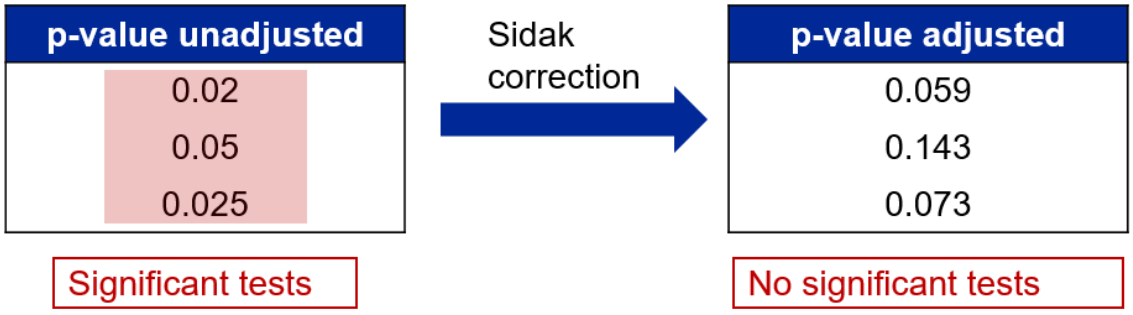
Bonferroni: $pval_{adj} = pval_{unadj} \cdot (number\ tests)$

Example:



Sidak : $pval_{adj} = 1 - (1 - pval_{unadj})^{(number\ tests)}$

Example:



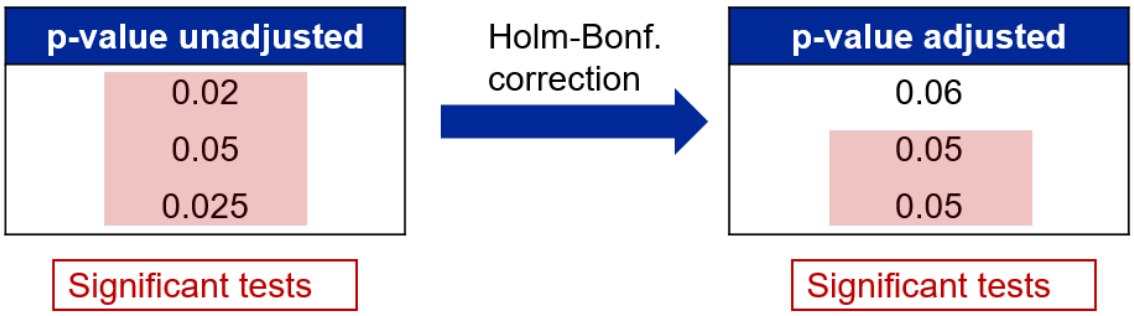
Holm-Bonferroni:

- Sort the unadjusted p-values from lowest to highest: $p(1)<\dots Adjust the ith p-value according to the following rule depending on its ranking position:

  $$pval_{adj} = (number\ tests - rank_i + 1) \cdot pval_{unadj}$$

  for $i = 1, \dots, number\ of\ tests$

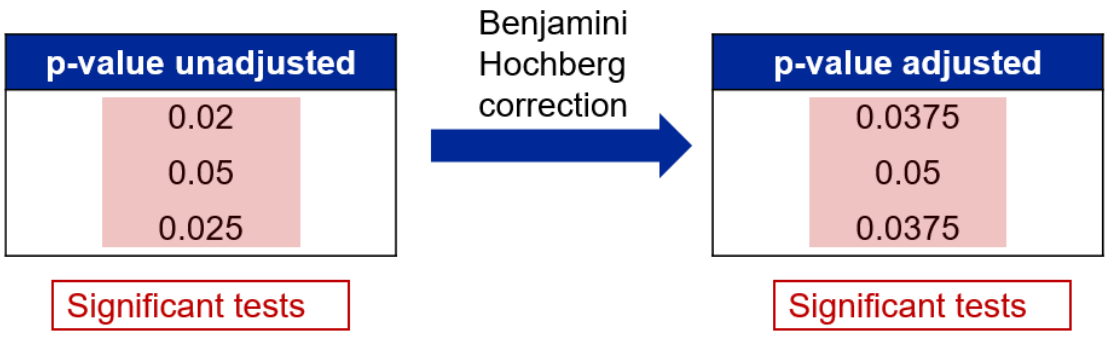
Example:



Benjamini Hochberg:

- Sort the unadjusted p-values from lowest to highest: $p(1)<\dots Multiply each p-value by the number of tests and divide it by its rank.
- The resulting sequence should be non-decreasing. If it starts to decrease, set the preceding p-value equal to the subsequent. Repeat this until the sequence is non-decreasing.

Example:



Pairwise t-tests with Bonferroni correction (see padjust):

In [84]:
```
data_pair=data[['Age','Education']]
```

In [85]:
```
data_pair.pairwise_tests(dv='Age', between='Education',alternative="two-sided",
                         interaction=False,padjust='bonf').round(3)
```

Out[85]:

	Contrast	A	B	Paired	Parametric	T	dof	alternative
0	Education	college/bachelor degree	currently a primary school pupil	False	True	9.615	11.109	two-sided
1	Education	college/bachelor degree	doctorate degree	False	True	-6.195	4.408	two-sided
2	Education	college/bachelor degree	masters degree	False	True	-13.793	116.290	two-sided
3	Education	college/bachelor degree	primary school	False	True	16.507	217.682	two-sided
4	Education	college/bachelor degree	secondary school	False	True	6.707	350.933	two-sided
5	Education	currently a primary school pupil	doctorate degree	False	True	-10.909	7.691	two-sided
6	Education	currently a primary school pupil	masters degree	False	True	-17.016	18.713	two-sided
7	Education	currently a primary school pupil	primary school	False	True	-2.085	11.260	two-sided
8	Education	currently a primary school pupil	secondary school	False	True	-7.437	9.647	two-sided
9	Education	doctorate degree	masters degree	False	True	-0.565	5.799	two-sided
10	Education	doctorate degree	primary school	False	True	11.349	4.438	two-sided
11	Education	doctorate degree	secondary school	False	True	8.001	4.127	two-sided
12	Education	masters degree	primary school	False	True	24.115	154.000	two-sided
13	Education	masters degree	secondary school	False	True	18.464	89.234	two-sided
14	Education	primary school	secondary school	False	True	-13.625	127.922	two-sided

Pairwise correlation tests with Benjamini Hochberg correction (see padjust):

In [86]:
```
data_pair2=data[['Age','Height','Weight']]
```

In [87]:
```
pg.pairwise_corr(data_pair2, method='pearson', alternative='greater', padjust='fdr_
```

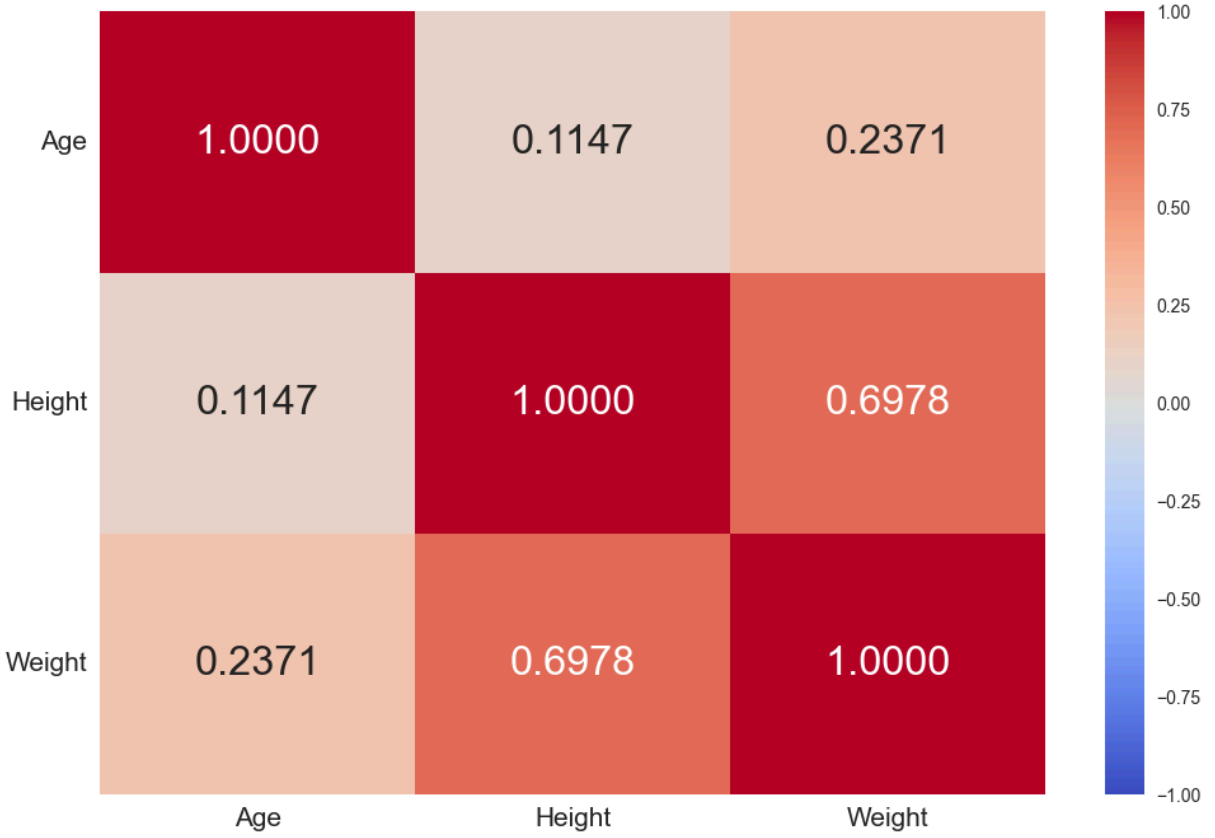
Out[87]:

	X	Y	method	alternative	n	r	CI95%	p-unc	p-corr	p-adjust	BF10
0	Age	Height	pearson	greater	988	0.115	[0.06, 1.0]	0.0	0.0	fdr_bh	54.741
1	Age	Weight	pearson	greater	987	0.238	[0.19, 1.0]	0.0	0.0	fdr_bh	2.084e+11
2	Height	Weight	pearson	greater	980	0.698	[0.67, 1.0]	0.0	0.0	fdr_bh	1.888e+140

In [88]:
```
data_pair2=data_pair2.dropna()
cm=describe.corrmat(data=data_pair2,utri=False,stars=True)
```

In [89]:
```
cm=describe.corrmat(data=data_pair2)
```

In [90]:
```
cm.heatmap()
```



In [91]:
```
cm.table
```

Out[91]:

	Age	Height	Weight
Age	1.000000	0.114687	0.237084
Height	0.114687	1.000000	0.697786
Weight	0.237084	0.697786	1.000000

Chapter 2: Models

"All models are wrong but some are useful"

Section 2.1: Regression Models

2.1.1 Fundamental Idea of Linear Regression

In the social sciences, we often aim at examining whether a independent variable (X) has an effect on a dependent variable (Y). For instance, do marketing expenses (X) increase sales (Y)?

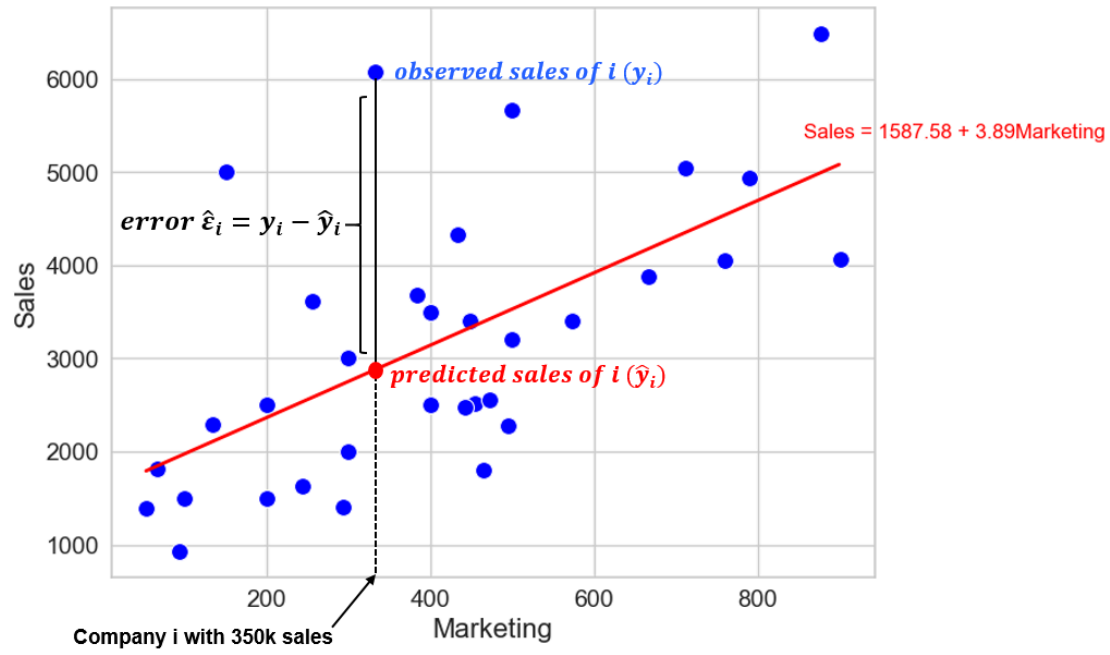
The idea of **linear regression** is to fit a line to the data. The theoretical equation of such a model is:

$$Sales = \beta_0 + \beta_1 \cdot Marketing + \varepsilon,$$

where

- β_0, β_1 = axis intercept, slope
- ε = error term that accounts for variables that are not in the equation (e.g., reputation of the company, product quality)

Finding the optimal line:



Each line leads to specific errors. We would like to find the line that minimizes the errors. That is the line that is located nearest to the data.

How do we aggregate the errors?

- The total sum of errors ($\varepsilon_1 + \cdots + \varepsilon_n$) implies that negative and positive deviations cancel each other partly out.

The solution is to aim at minimizing the total sum of **squared** errors (**Ordinary Least Squares (OLS)** approach):

$$SSR = \varepsilon_1^2 + \cdots + \varepsilon_n^2$$

In case of a simple linear regression (1 dependent Y, 1 independent X), the estimated regression coefficients can be computed as follows:

$$\hat{\beta}_1 = \frac{covariance(X,Y)}{variance(X)} \ \ and \ \ \hat{\beta}_0 = \frac{\Sigma y_i - \hat{\beta}_1 \Sigma x_i}{n}.$$

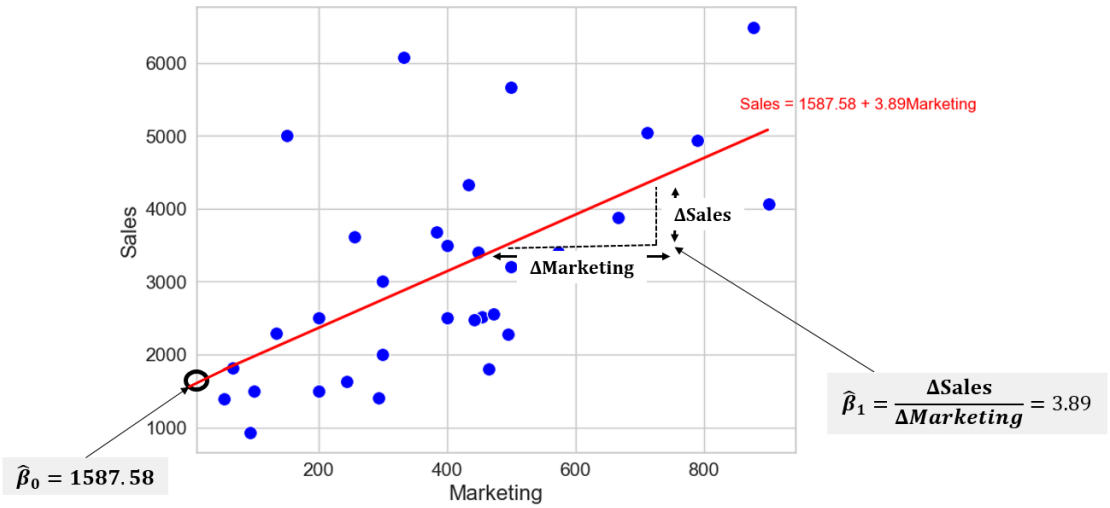
In the example above:

- covariance(Marketing, Sales) = 207797.65
- variance(Marketing) = 53472.15
- Sum Sales = 104432.81
- Sum Marketing = 13392
- n = 33

Hence (differences to estimated line above are due to rounding differences):

$$\hat{\beta}_1 = \frac{207797.65}{53472.15} \approx 3.89 \ \ and \ \ \hat{\beta}_0 = \frac{104432.81 - 3.89 \cdot 13392}{33} \approx 1585.99.$$

Interpreting the coefficients



- Intercept ($\hat{\beta}_0$): The estimated sales of a company with 0 marketing expenses is 1'587'580 $
- Slope ($\hat{\beta}_1$): A unit increase in marketing expenses increases sales about 3.89 $

2.1.2 Multiple Linear Regression

We may want to add further independent variables to our model. The theoretical regression equation with k independent variables looks as follows:

$$Y = \beta_0 + \beta_1 \cdot X_1 + \cdots + \beta_k \cdot X_k + \varepsilon$$

Conceptually, we distinguish between target variables and control variables. Say, X_1 and X_2 are the variables of interest. We examine their influence on Y while controlling for $X_3, \ldots, X_k$.

The math works similarly to the simple model but now we have to imagine a regression plane (or hyperplane).

Example:

$$Sales = \beta_0 + \beta_1 \cdot Marketing + \beta_2 \cdot Quality + \varepsilon,$$

where Quality is measured by the average product lifespan.

In [92]:
```
#data
data2= pd.read_excel ('C:\\Users\\kauffeldt\\Dropbox\\Teaching\\3_Programme\\Data\\
```

In [93]:
```
#Separate into DV and IVs
X=data2[['Marketing','Quality']]
y=data2['Sales']
```

2.1.3 Categorical Independent Variables (Dummy Encoding)

Categorical variables (nominal or ordinal) take on categories as values. A regression equation can only deal with numbers. Therefore, we have to convert these variables into indicator variables. Example: Variable "eye color" that can take on the categories blue, brown, green.

Variable Eye Color
Blue
Brown
Green

Dummy Encoding

3 Indicator Variables

Eye color	Dummy blue	Dummy brown	Dummy green
Blue	1	0	0
Brown	0	1	0
Green	0	0	1

However, we cannot use all 3 indicator variables because two always predict the third perfectly. Thid dependence would cause the regression model to break down. Hence, we have to drop one of the categories (which does not matter). The dropped category serves as reference category: all effects are measured with respect to this category. Dropping, e.g. blue, yields:

Eye color	Dummy brown	Dummy green
Blue	0	0
Brown	1	0
Green	0	1

In section 2.1.5, we explain how dummy encoding is done in Python.

2.1.4 Steps Regressionanalysis

1. Step. Write down hypothesized cause and effect relationship with control variables.
2. Step. Data preprocessing
3. Step. Check requirements
4. Step. Write down estimated regression equation and interprete result.
5. Step. Further robustness checks.

We will explain these steps with the help of the following example: We would like to examine the how marketing expenses affect sales while controlling for the reputation of a company.

Step 1. Theoretical regression equation:

$$Sales = \beta_0 + \beta_1 \cdot Marketing + \beta_2 \cdot Reputation$$

2.1.5 Data Preprocessing

Slice data and remove NaN

In [94]:
```
data_reg=data2[['Marketing','Sales','Reputation']]
data_reg=data_reg.dropna()
```

Separate data in dependent and independent

In [95]:
```
X,y=data_reg.drop('Sales',axis=1),data_reg['Sales']
```

Dummy encoding
Reputation takes on the categories low, medium, high. Hence, we have to dummy encode the matrix of the independents.

Define encoder:

In [96]:
```
enc=dataprep.onehot(cats=['Reputation'],drop=None)
```

Fit encoder to data and transform data:

In [97]:
```
enc.fit(X)
X_dum=enc.transform(X)
```

We decide to drop category low. Which category we drop does not matter as mentioned earlier.

In [98]:
```
X_dum=X_dum.drop('Reputation_Low',axis=1)
```

In [99]:
```
X_dum.head(3)
```

Out[99]:

	Reputation_High	Reputation_Medium	Marketing
0	1.0	0.0	500.0
1	1.0	0.0	876.0
2	0.0	1.0	759.0

2.1.5 Check requirements

Assumptions Linear Regression

- **L1. Linearity.** There is a linear relationship between dependent and independents.
- **L2. Lack of perfect (Multi)collinearity.** There is no perfect linear relationship between some independent variables.
- **L3. Strict Exogeneity.** The conditional means of the errors are zero ($E[\varepsilon_i \mid x_i] = 0$).
- **L4. Homoscedasticity.** Errors have equal conditional variances ($var(\varepsilon_i \mid x_i) = var(\varepsilon_j \mid x_j)$) for all i,j.
- **L5. No Autocorrelation.** Errors are not correlated ($cov(\varepsilon_i, \varepsilon_j) = 0$) for all i,j.
- **L6. Normality.** Errors follow a multivariate normal distribution.

L1 and L2 refer to the variables. The remaining assumptions refer to errors. In order to test these assumptions, we fit the regression model.

In [100…
```
reg=regression(X_dum,y)
```

In [101…
```
pip install lxml
```

```
Requirement already satisfied: lxml in c:\users\kauffeldt\anaconda3\envs\profkaufenv
\lib\site-packages (6.0.0)
Note: you may need to restart the kernel to use updated packages.
```

L1. Linearity

In [102…
```
reg.datafit.round(4)
```

Out[102…

	dv	dof resid	dof model	R2	adj. R2	omnibus (F)	omnibus (p-val)	LL
linear reg. fit	Sales	29.0	3.0	0.7077	0.6775	23.4053	0.0	-266.3585

Degrees of freedom:

- dof model = number of independent variables (k)
- dof resid = n - k - 1

R2:

$$R^2 = \frac{Variation\ explained\ by\ the\ model}{Total\ variation} = \frac{(\hat{y}_1 - mean\ y)^2 + \cdots + (\hat{y}_n - mean\ y)^2}{(y_1 - mean\ y)^2 + \cdots + (y_n - mean\ y)^2}$$

- $R^2 \approx 0.7077$ means that approx. 71% of the variations in Sales can be explained by variations in marketing expenses and reputation, which suggest that the model fits the data well.

Omnibus:
The omnibus test tests the null hypothesis that no independent affects the dependent against the alternative that some independnets affect the dependent.

$$H0 : \beta_1 = \cdots = \beta_k = 0 \ \ (R^2 = 0) \ \ and \ \ HA : \text{there is a } \beta_i \neq 0 \ \ (R^2 \neq 0)$$

- The p-value of the test is 0, thus we can rejet the null hypothesis, which supports the previous result.

In [103…
```
ass=reg.asstest
```

In [104…
```
ass
```

Out[104…

	test	statistic	p-val
linear reg.	Jarque-Bera	0.5845	0.7466
assumptions	Breusch-Pagan	15.5276	0.0014
	Durbin-Watson	2.1078	
	Ramsey RESET	0.2877	0.8832

The Ramsey RESET test tests if there are non-linear dependencies between dependent and independents. It tests the null hypothesis that a non-linear model has has a higher explanatory power than the linear one against the alternative that this is not the case.

The p-value is 55.7%>5%. We cannot reject the null hypothesis and found no evidence for relevant non-linear dependencies.

L2. No (Multi)collinearity

What is (multi)collinearity? -> will be answered in the lecture.

Collinearity: Correlation Matrix

In [105…
```
describe.corrmat(X_dum,stars=True,utri=False).table
```

Out[105…

	Reputation_High	Reputation_Medium	Marketing
Reputation_High		**	
Reputation_Medium	-0.559**		
Marketing	0.2885	0.1539	

An indicator for collinearity (pairwise linearity) is a correlation coefficient greater than 0.7 or less than -0.7. This is not the case.

Multicollinearity: Variance inflation factors (VIF)

In [106…

```
reg.vif
```

Out[106…

	var	vif
variance inflation	intercept	5.040589
factors	Reputation_High	1.838889
	Reputation_Medium	1.726673
	Marketing	1.294894

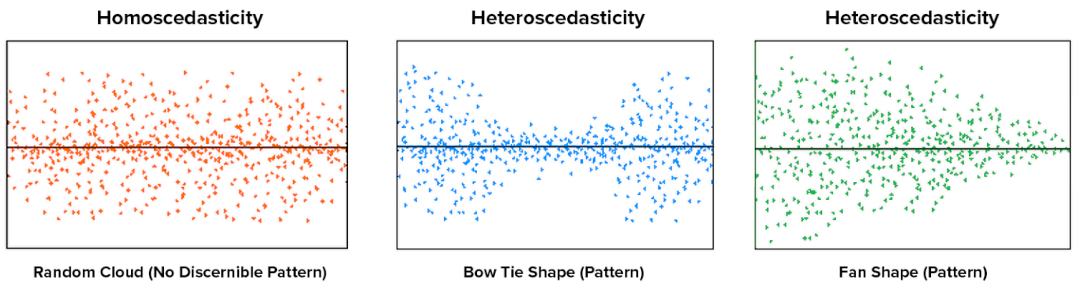
Multicollinearity (if three or more independents have a strong linear relationship) inflates the variance if there are small changes. The VIFs indicate the magnitude of variance inflations. If they are below 10, we may assume that there is no multicollinearity.

L3. Exogeneity

Exogeneity is established through theoretical or qualitative arguments in observational studies, not by statistical tests. In randomized control trials, the treatment would be exogenous by design (if the trial was executed properly and subjects complied with the design, etc.).

L4. Homoscedasticity

What is homoscedasticity?

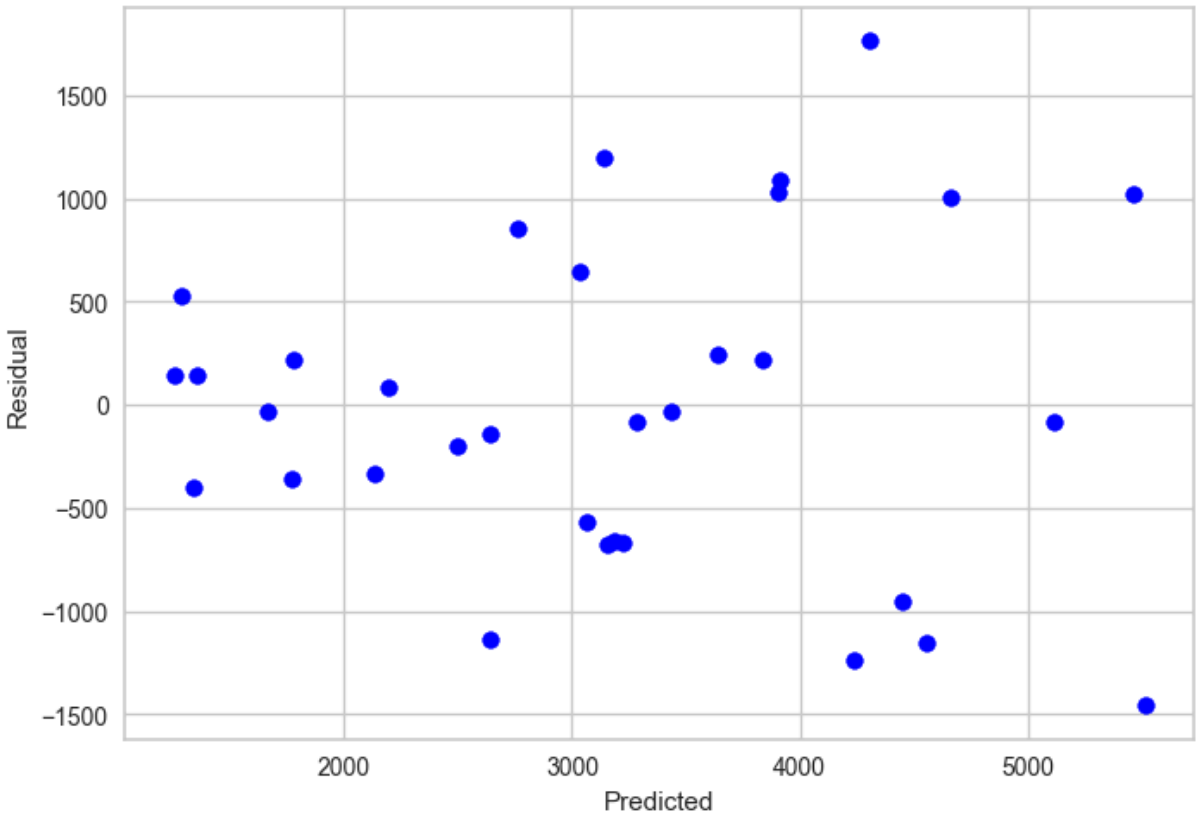


Source: https://www.fireblazeaischool.in/blogs/assumptions-of-linear-regression/

In [107…
```
pred=reg.pred
resid=reg.resid
```

In [108…
```
plt.scatter(pred,resid,color='blue')
plt.xlabel('Predicted')
plt.ylabel('Residual')
```

Out[108… `Text(0, 0.5, 'Residual')`



In [109…
```
ass
```

Out[109…

	test	statistic	p-val
linear reg.	Jarque-Bera	0.5845	0.7466
assumptions	Breusch-Pagan	15.5276	0.0014
	Durbin-Watson	2.1078	
	Ramsey RESET	0.2877	0.8832

The Breusch-Pagan test tests the null hypothesis that the residuals have constant variance against the alternative that this is not the case.

The p-value is 0.14% < 5%. Therefore, we found evidence that assumption L4 is violated. This is supported by the scatterplot above.

L5. No Autocorrelation

In [110…
```
ass
```

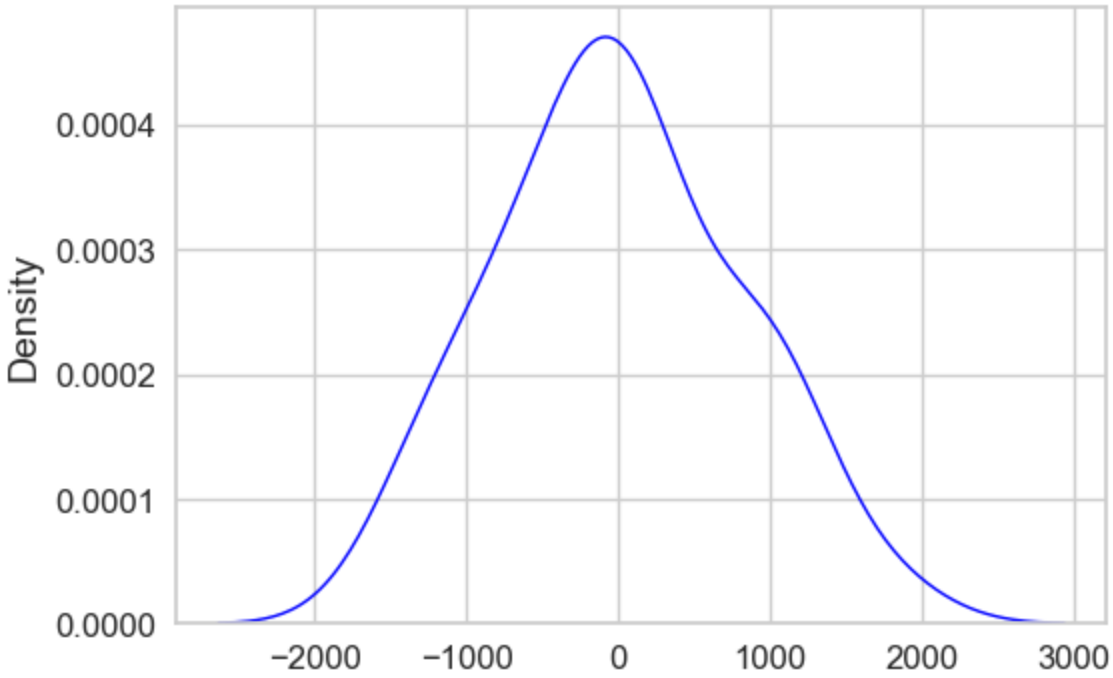
Out[110…

	test	statistic	p-val
linear reg.	Jarque-Bera	0.5845	0.7466
assumptions	Breusch-Pagan	15.5276	0.0014
	Durbin-Watson	2.1078	
	Ramsey RESET	0.2877	0.8832

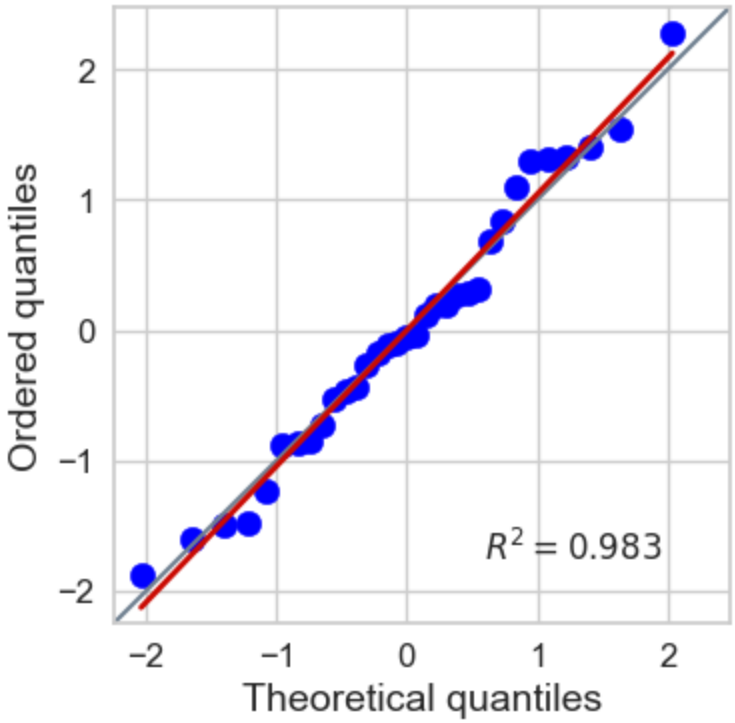
The Durbin-Watson test statistic tells us if the errors might be autocorrelated. If the DW statistics is between 1.5 and 2.5, we may assume that there is no autocorrelation. This is the case.

L6. Normality

In [111…
```
plots.dist(resid,fig=[6,4],labelsize=14,ticksize=12,linewidth=1)
```



In [112…
```
plots.qq(resid,fig=[6,4],labelsize=14,ticksize=12)
```



In [113…
```
ass
```

Out[113…

	test	statistic	p-val
linear reg.	Jarque-Bera	0.5845	0.7466
assumptions	Breusch-Pagan	15.5276	0.0014
	Durbin-Watson	2.1078	
	Ramsey RESET	0.2877	0.8832

The Jarque-Bera test tests the null hypothesis that there is normality against the alternative that errors are not normally distributed.

The p-value is 74.66% > 5%. We thus found no evidence that this assumption is violated.

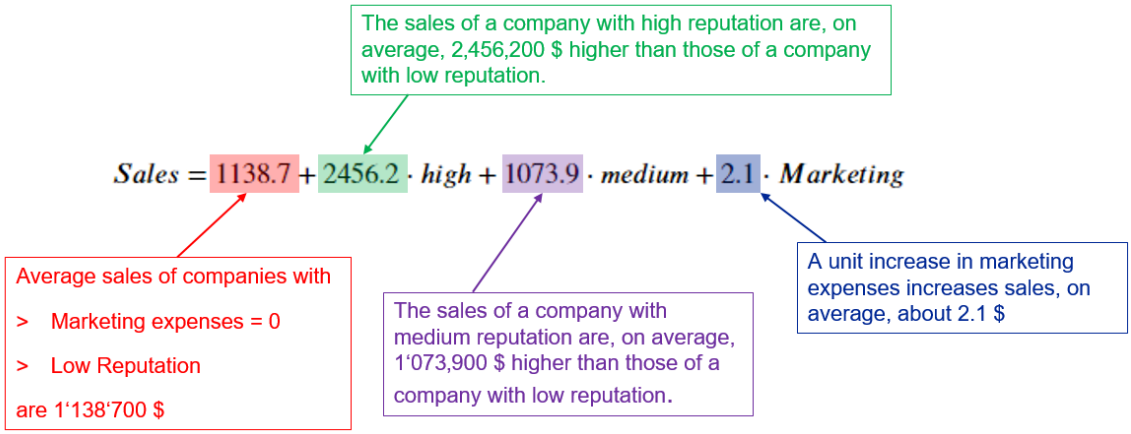
2.1.6 Interprete Result

In [114…
```
reg.coef.round(4)
```

Out[114...

		coef	stand. coef	std err	t	P>|t|	[0.025	0.97
linear reg.	intercept	1138.7134	-0.0334	322.998	3.525	0.001	478.109	1799.3
coefficients	Reputation_High	2456.2189	1.4808	438.050	5.607	0.000	1560.306	3352.13
	Reputation_Medium	1073.9194	-0.1078	379.661	2.829	0.008	297.425	1850.4
	Marketing	2.1386	-1.3396	0.719	2.975	0.006	0.668	3.6

Estimated regression equation and coefficient interpretation



Significance

Column $P > |t|$ shows the p-values of the t-tests for each coefficient of the independents with hypotheses:

- $H0$: The independent does not influence the dependent $(\beta_i = 0)$
- HA : The independent does influence the dependent $(\beta_i \neq 0)$

We found evidence that all variables significantly affect the dependent (p-values = 0, 0.008, 0.006).

Effects

High has the strongest influence on sales (standardized coefficient is farthest away from zero). The least influence has medium.

Robustness

The standard error (std err) shows the variance of the coefficient estimates.

2.1.8 Robustness Checks

1. Influential observations (outlier)
2. Overfitting
3. Moderation Effect

4. Omitted Variable Bias (OVB)
5. Mediation Effect

Overfitting

Including additional independents to the model always increases R^2. However, it is not clear if this is due to a true causal effect or purely mechanical: if we add independents, we lose degrees of freedom. This may lead to an artificial increase of R^2. For instance, if there are no degrees of freedom (number of groups - 1 (k-1) = number of observations (n), we always end up with $R^2 = 1$, purely due to mechanical mathematical reasons. An artificial increase of the explanatory power is known as **overfitting**, which means that we fitted the model too much to the data.

A possibility to detect overfitting is the adjusted R^2_{adj}. In contrast to R^2, it takes into account the degress of freedom and may decrease when we including additional independent variables:

$$R^2_{adj} = \left(\boldsymbol{R^2} - \frac{\boldsymbol{k}}{\boldsymbol{n-1}}\right) \cdot \left(\frac{\boldsymbol{n-1}}{\boldsymbol{n-k-1}}\right)$$

In [115…]
```
reg.datafit
```

Out[115…]

	dv	dof resid	dof model	R2	adj. R2	omnibus (F)	omnibus (p-val)	LL
linear reg. fit	Sales	29.0	3.0	0.707708	0.677471	23.405266	6.746427e-08	-266.358535

Data fit without variable "reputation":

In [116…]
```
X_dum2=X_dum.drop(['Reputation_High','Reputation_Medium'],axis=1)
```

In [117…]
```
reg2=regression(X_dum2,y)
```

In [118…]
```
reg2.datafit
```

Out[118…]

	dv	dof resid	dof model	R2	adj. R2	omnibus (F)	omnibus (p-val)	LL
linear reg. fit	Sales	31.0	1.0	0.381321	0.361364	19.106759	0.000129	-278.730776

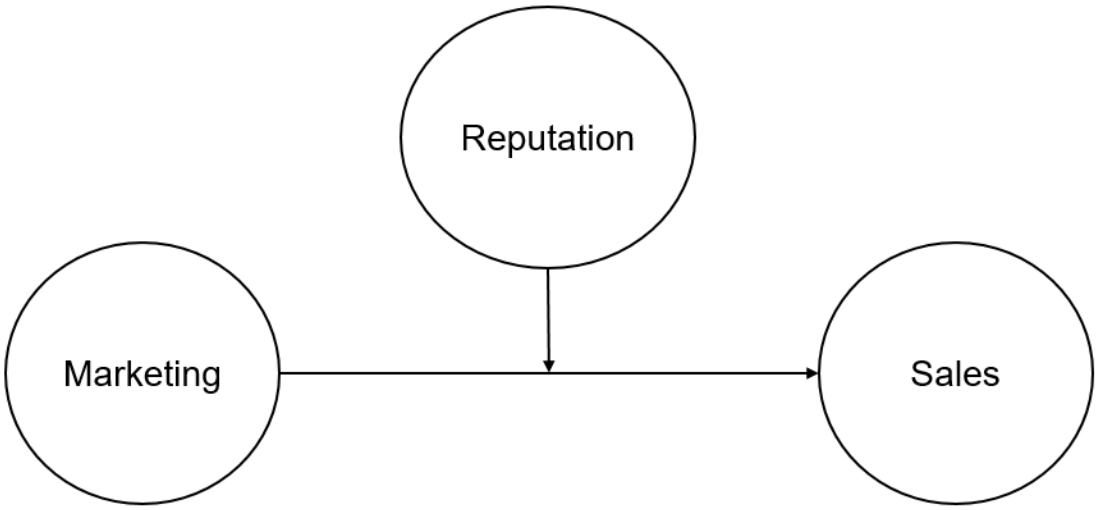
Conclusion:

The adjusted R^2 of the model without Reputation is less than that of the model with reputation. Hence, there is no indication that our model is overfitted.

Moderation Effect

Moderator = third variable that determines the magnitude of the effect.

Marketing might have a weaker effect on Sales if the company has less reputation:

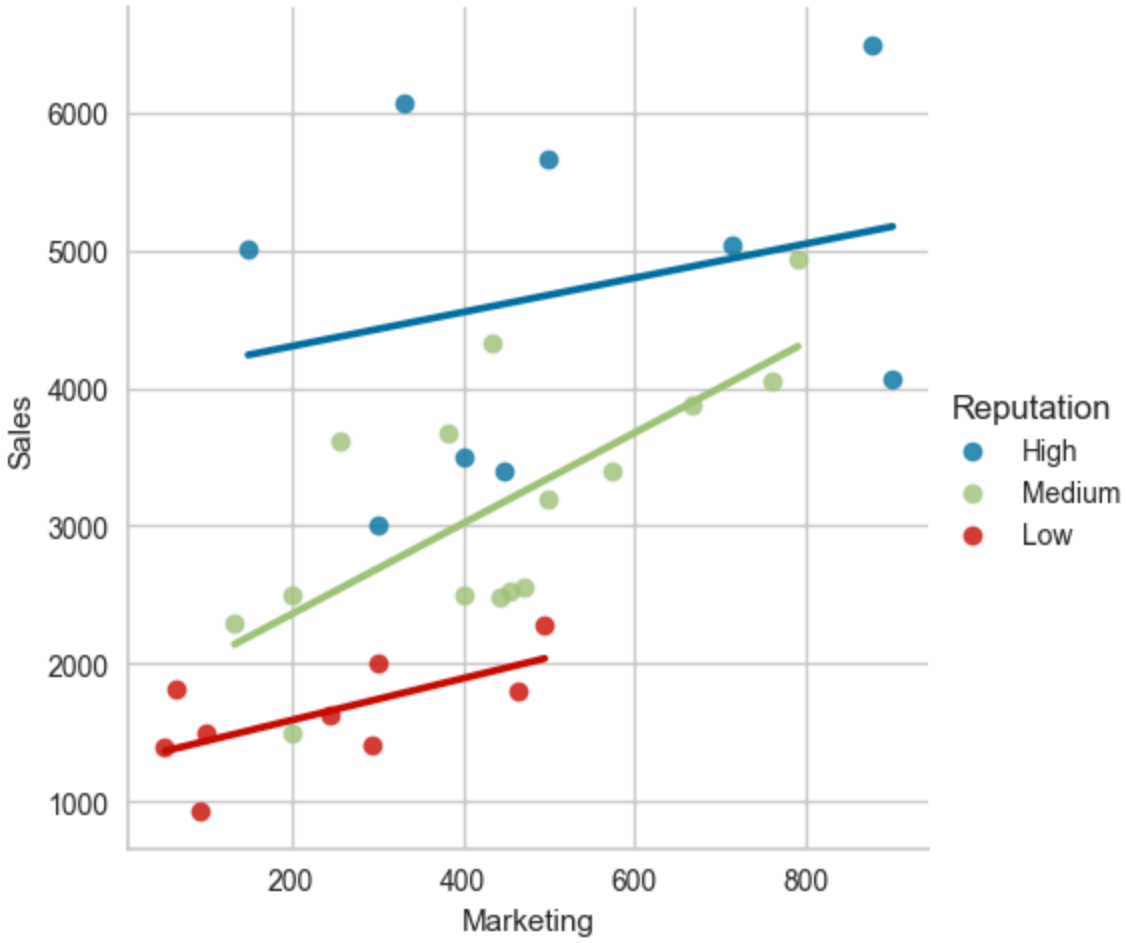


Let's have a look at the groupwise regressions:

In [119…
```
data_reg
data_reg=data_reg.dropna()
```

In [120…
```
sns.lmplot(data=data_reg, x="Marketing", y="Sales", hue="Reputation",ci=None)
```

Out[120…
```
<seaborn.axisgrid.FacetGrid at 0x15cd1114830>
```



The slopes of reputation 'low' and 'high' seems fairly similar. The slope of 'medium' hower is steeper.

In order to test if there is indeed a moderation, we need to include the interactions $marketing \times medium$ and $marketing \times high$ to the model. If these interactions are significant, we found evidence for a moderation effect.

In [121…
```
X_mod = X_dum.copy()
X_mod['marketing_medium']=X_mod['Marketing']*X_mod['Reputation_Medium']
X_mod['marketing_high']=X_mod['Marketing']*X_mod['Reputation_High']
```

In [122…
```
reg_mod=regression(X_mod,y)
```

In [123…
```
reg_mod.coef.round(4)
```

Out[123...

		coef	stand. coef	std err	t	P>|t|	[0.025	0.9
linear reg.	intercept	1283.6331	0.5312	492.034	2.609	0.015	274.062	2293.
coefficients	Reputation_High	2770.5907	1.9983	805.225	3.441	0.002	1118.405	4422.
	Reputation_Medium	414.4227	-0.3265	732.487	0.566	0.576	-1088.517	1917.
	Marketing	1.5208	-0.7339	1.736	0.876	0.389	-2.041	5.
	marketing_medium	1.7763	-0.7336	2.067	0.859	0.398	-2.465	6.
	marketing_high	-0.2773	-0.7356	2.065	-0.134	0.894	-4.515	3.

The p-values of both interactions terms are greater than 5%. Hence, we found no evidence for a moderation effect.

Omitted Variable Bias (OVB)
The results of a regression are biased if a relevant independent variable is omitted. A variable is relevant if the following conditions are met:

1. The variable must have an impact on the dependent.
2. The variable must be correlated with another independent variable.

Our data set contains another variable that might be relevant: product quality measured by the average lifespan of products.

What would be the case if quality was relevant? For the sake of simplicity, we omit reputation in the following:

- True causal relationship: $Sales = \beta_0 + \beta_1 Marketing + \beta_2 Quality$
- Our model: $Sales = \gamma_0 + \gamma_1 Marketing$

If quality and marketing expenses are correlated it holds that:
$Quality = \alpha_0 + \alpha_1 Marketing$. Inserting this equation into the true causal one yields:

$$\gamma_1 = \beta_1 + \beta_2 \alpha_1$$

Hence, instead of estimating the true effect of marketing on sales (β_1), we estimated γ_1, which might be higher or lower than β_1 depending on the direction of the correlation between marketing and quality.

Let's check if quality leads to an omitted variable bias:

Condition 1: Is quality correlated with another independent? $\rightarrow$ correlation matrix.

In [124…
```
X_ovb=X_dum.copy()
X_ovb['Quality']=data2['Quality']
```

In [125…
```
describe.corrmat(X_ovb,stars=True,utri=False).table
```

Out[125…

	Reputation_High	Reputation_Medium	Marketing	Quality
Reputation_High		**		**
Reputation_Medium	-0.559**			
Marketing	0.2885	0.1539		****
Quality	0.5841**	-0.0625	0.6951****	

Quality is highly correlated with marketing.

Condition 2: Does quality affect sales? → regression

In [126…
```
reg_ovb=regression(X_ovb,y)
```

In [127…
```
reg_ovb.coef
```

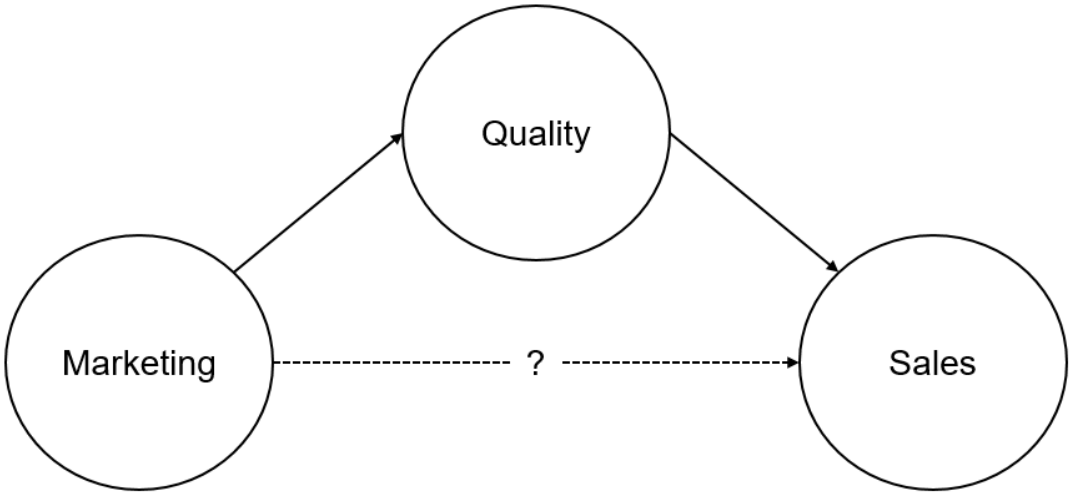
Out[127…

		coef	stand. coef	std err	t	P>|t|	[0.025	0.9
linear reg.	intercept	882.1838	0.201650	294.417	2.996	0.006	279.097	1485.2
coefficients	Reputation_High	1681.7791	1.621210	456.236	3.686	0.001	747.223	2616.3
	Reputation_Medium	880.3159	0.198334	338.118	2.604	0.015	187.713	1572.9
	Marketing	0.5870	-1.363490	0.801	0.733	0.470	-1.053	2.2
	Quality	398.1351	-0.657704	127.010	3.135	0.004	137.966	658.3

The p-value of quality is lower than 5%. Together with condition 1, this suggests a omitted variable bias. Notice that marketing is not significant anymore when we control for quality.

Mediation Effect

Mediator = third variable through which an effect occurs



Remark: This example surely is to some degree artifical because it is not clear why marketing should cause higher quality. However, it is preferable to using a complete new data set.

Test if quality is a mediator:

In [128…
```
data_med=X_ovb
data_med['Sales']=y
```

In [129…
```
pg.mediation_analysis(data=data_med, x='Marketing', m='Quality',covar=['Reputation_
```

Out[129…

	path	coef	se	pval	CI[2.5%]	CI[97.5%]	sig
0	Quality ~ X	0.003897	0.000920	0.000211	0.002015	0.005779	Yes
1	Y ~ Quality	455.685659	99.042733	0.000077	253.120524	658.250793	Yes
2	Total	2.138628	0.718943	0.005857	0.668225	3.609031	Yes
3	Direct	0.587043	0.800796	0.469607	-1.053314	2.227400	No
4	Indirect	1.551585	0.550407	0.008000	0.588977	2.713985	Yes

The indirect effect (also referred to as average causal mediation effect or ACME) of marketing on sales through mediator quality quantifies the estimated difference in sales resulting from a one-unit change in marketing through a sequence of causal steps in which marketing affects quality, which in turn affects sales. It is considered significant if the specified confidence interval does not include 0.

The indirect effect is highly significant, while the direct is not. This suggests a mediation effect.

2.1.7 Regression Models with Categorical Dependent Variable

The linear regression model works only properly when the dependent variable is quantitative. For categorical dependent variable it is advisable to use another model. The following table summarizes the suggested models.

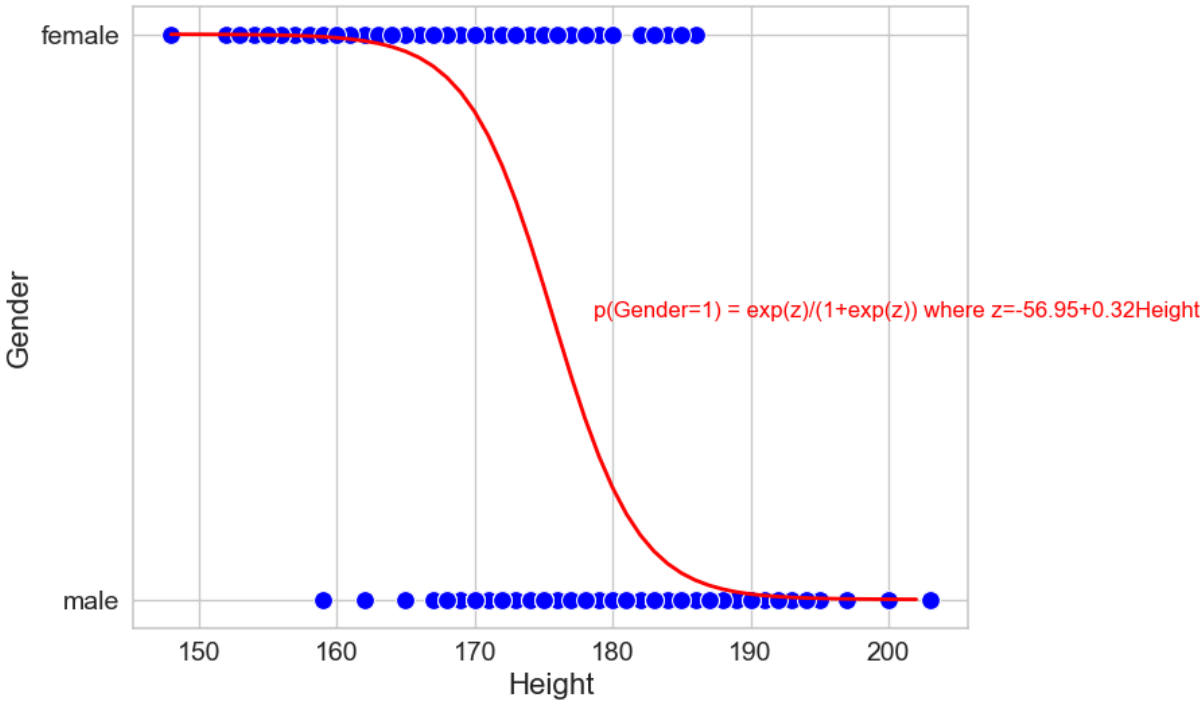
		Model	Python-implementation via regression object
Dependent Variable	Quantitative	Linear Regression	regression(X,y)
	Ordinal	Ordinal Regression	regression(X,y,method='ordinal')
	Binary	Logistic Regression	regression(X,y,method='logistic')
	Nominal	Multinomial Regression	regression(X,y,method='multinomial')

Example: Logistic Regression
Logistic Regression deals with binary dependent variables. It predicts the probabilities that the dependent takes on the categories. For instance, we may predict gender based on height.

In [130…
```
data_log=data[['Gender','Height']]
data_log=data_log.dropna()
data_log=data_log[data_log['Height']>70]
```

Visualize Logistic Regression:

In [131…
```
plots.scatter(data_log['Height'],data_log['Gender'],regression='logistic',intext=Tr
```



Results Logistic Regression:

In [132…
```
reg_log=regression(pd.DataFrame(data_log['Height']),data_log['Gender'],method='logi
```

```
Optimization terminated successfully.
         Current function value: 0.322937
         Iterations 8
```

In [133…
```
reg_log.coef.round(4)
```

Out[133…

		coef	exp(coef)	std err	z	P>|z|	[0.025	0.975]
logistic reg.	intercept	-56.9686	0.0000	3.541	-16.090	0.0	-63.908	-50.029
coefficients	Height	0.3243	1.3831	0.020	16.016	0.0	0.285	0.364