‖‖ Chapter 3

Probability and simulation (solutions to exercises)

Contents

Import Python packages

```python
# Import all needed python packages
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.stats.power as smp
```

3.1 Concrete items

||| **Exercise 3.1** **Concrete items**

A construction company receives concrete items for a construction. The length of the items are assumed reasonably normally distributed. The following requirements for the length of the elements are made

$$\mu = 3000 \text{ mm.}$$

The company samples 9 items from a delevery which are then measured for control. The following measurements (in mm) are found:

3003	3005	2997	3006	2999	2998	3007	3005	3001

a) Compute the following three statistics: the sample mean, the sample standard deviation and the standard error of the mean, and what are the interpretations of these statistics?

||| **Solution**

From the data we get the sample mean and sample standard deviation

$$\bar{x} = 3002.33\text{mm and } s = 3.708\text{mm.}$$

```
x = np.array([3003, 3005, 2997, 3006, 2999, 2998, 3007, 3005, 3001])
print(np.mean(x))

3002.3333333333335

print(np.std(x, ddof=1))

3.708099243547832
```

From Definition **??**, we get the standard error of the mean as

$$SE_{\bar{x}} = \frac{3.708}{\sqrt{9}} = 1.236.$$

The interpretations of these are:

$\bar{x} = 3002.33$ The best estimate we can make of the true mean length of such concrete items

$s = 3.708$ The population of concrete item lenghts has a standard deviation estimated at 3.7. We estimate the average deviation from the mean for concrete items to be 3.7 mm. So most concrete items will be within the range of approximately ± 7.4 mm of the mean

$SE_{\bar{x}} = 1.236$ All of the following are versions of the same story given by this number:

- The standard deviation of the sampling distribution of the sample mean (seen as a random variable) is (estimated at) 1.24

- And also: The standard deviation of the sampling distribution of the difference between sample mean and the population mean (seen as a random variable) is(estimated at) 1.24

- So from sample to sample (of size $n = 9$) the sample mean will be different. And the size of these differences, that is, the difference between the the sample mean and the true population mean is on average 1.24

- The sample mean is on average 1.24 away from the target: the population mean

- The error we will make on average in using the sample mean for estimating the population mean is 1.24

|||| **Remark 3.1**

Please, think about the difference between the story told by s and the story told by $SE_{\bar{x}}$ (both are estimated standard deviations, but for two VERY different concepts).

---

b) In a construction process, 5 concrete items are joined together to a single construction with a length which is then the complete length of the 5 concrete items. It is very important that the length of this new construction is within 15 m plus/minus 1 cm. How often will it happen that such a

construction will be more than 1 cm away from the 15 m target (assume that the population mean concrete item length is $\mu = 3000$ mm and that the population standard deviation is $\sigma = 3$)?

||| Solution

Let Y denote the length of the joined construction. So

$$Y = \sum_{i=1}^{5} X_i,$$

where X_i is the length of a randomly selected concrete item. So using the rules for mean and variance calculations from Section **??**, we can find that

$$E(Y) = \sum_{i=1}^{5} E(X_i) = \sum_{i=1}^{5} 3000 = 5 \cdot 3000 = 15000,$$

and

$$V(Y) = \sum_{i=1}^{5} V(X_i) = \sum_{i=1}^{5} 3^2 = 5 \cdot 9 = 45.$$

We can also state that, since the concrete item distribution is a normal

$$X_i \sim N(\mu, \sigma^2), \ i = 1, \ldots, 5,$$

then the sum of five (independent) of such will be normal (Theorem **??**), so

$$Y \sim N(15000, 45).$$

(Actually, the normality result is expressed in Theorem **??** for the sample mean, however, the sum is just a simple scaling of the mean, so then the normality also holds for the sum) So we can now find the answer to the question

$$P(|Y - 15000| > 10) = 2 \cdot P(\frac{(Y - 15000)}{\sqrt{45}} > 10/\sqrt{45}) = 2 \cdot P(Z > 1.4907) = 0.136.$$

```
print(2*(1-stats.norm.cdf(15010, loc=15000, scale=np.sqrt(45))))

0.1360371281141437

print(2*stats.norm.cdf(-1.4907))

0.13604027609838776
```

In between 13 and 14 cases out of 100 the joined construction is beyond 1 cm away from the target – maybe a new supplier should be considered!

c) Find the 95% confidence interval for the mean μ.

‖‖‖ **Solution**

Since the 97.5%-quantile, $t_{0.975}$ of the t-distribution with 8 degrees of freedom equals $t_{0.975} = 2.306$ (found in Python as: `stats.t.ppf(0.975, 8)`, we get

$$3002.33 \pm 2.306 \cdot \frac{3.708}{\sqrt{9}} \Leftrightarrow [2999.5; 3005.2].$$

Or everything in Python:

```python
x = np.array([3003, 3005, 2997, 3006, 2999, 2998, 3007, 3005, 3001])
print(np.mean(x))
```

```
3002.3333333333335
```

```python
print(np.std(x, ddof=1))
```

```
3.708099243547832
```

```python
# Quantile
print(stats.t.ppf(0.975, df=8))
```

```
2.3060041350333704
```

```python
# Confidence interval directly
(CI_low,CI_high) = stats.ttest_1samp(x, popmean=0).confidence_interval()
print(CI_low,CI_high)
```

```
2999.4830359370885 3005.1836307295785
```

---

d) Find the 99% confidence interval for μ. Compare with the 95% one from above and explain why it is smaller/larger!

||| **Solution**

Since the 99.5%-quantile, $t_{0.995}$ of the t-distribution with 8 degrees of freedom equals $t_{0.995} = 3.355$ (found in Python as: `stats.t.ppf(0.995,8)`, we get

$$3002.33 \pm 3.335 \cdot \frac{3.708}{\sqrt{9}} \Leftrightarrow [2998.2; 3006.5].$$

Or everything in Python:

```
print(stats.t.ppf(0.995,df=8))

3.3553873313333957


(CI_low,CI_high) = stats.ttest_1samp(x, popmean=0).confidence_interval(
  confidence_level=0.99)
print(CI_low,CI_high)

2998.1859635916244 3006.4807030750426
```

It makes good sense that the 99% confidence interval becomes larger than the 95% one, as the consequence of wanting to be more confident on capturing the true mean μ will make us having to state a larger interval.

e) Find the 95% confidence intervals for the variance σ^2 and the standard deviation σ.

||| **Solution**

We use the formula for the variance confidence interval

$$\left[\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}}; \frac{(n-1)s^2}{\chi^2_{\alpha/2}} \right].$$

where the quantiles come from a χ^2-distribution with $\nu = n - 1 = 8$ degrees of

freedom

$$\left[\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}}; \frac{(n-1)s^2}{\chi^2_{\alpha/2}}\right] = \left[\frac{8 \cdot 3.708^2}{\chi^2_{0.975}}; \frac{8 \cdot 3.708^2}{\chi^2_{0.025}}\right] = \left[\frac{8 \cdot 13.75}{17.535}; \frac{8 \cdot 13.75}{2.180}\right]$$

$$= [6.273; 50.465].$$

And for the standard deviation:

$$\left[\sqrt{6.273}; \sqrt{50.465}\right] = [2.50; 7.10].$$

In Python:

```
print(stats.chi2.ppf((0.975,0.025), df=8))
```

```
[17.53454614  2.17973075]
```

```
CI_var = 8*np.var(x, ddof=1)/stats.chi2.ppf((0.975,0.025), df=8)
print(CI_var)
```

```
[ 6.27333032 50.46494854]
```

```
CI_std = np.sqrt(CI_var)
print(CI_std)
```

```
[2.50466172 7.10386856]
```

f) Find the 99% confidence intervals for the variance σ^2 and the standard deviation σ.

||| **Solution**

$$\left[\frac{8 \cdot 13.75}{21.955}; \frac{8 \cdot 13.75}{1.344}\right] = [5.010; 81.820].$$

And for the standard deviation:

$$\left[\sqrt{5.010}; \sqrt{81.820}\right] = [2.24; 9.05].$$

In Python:

```python
print(stats.chi2.ppf((0.995,0.005), df=8))

[21.95495499  1.34441309]


CI_var = 8*np.var(x, ddof=1)/stats.chi2.ppf((0.995,0.005), df=8)
print(CI_var)

[ 5.01025851 81.82009017]


CI_std = np.sqrt(CI_var)
print(CI_std)

[2.23836067 9.04544582]
```

3.2   Aluminum profile

||| **Exercise 3.2**   **Aluminum profile**

The length of an aluminum profile is checked by taking a sample of 16 items whose length is measured. The measurement results from this sample are listed below, all measurements are in mm:

180.02	180.00	180.01	179.97	179.92	180.05	179.94	180.10
180.24	180.12	180.13	180.22	179.96	180.10	179.96	180.06

From data is obtained: $\bar{x} = 180.05$ and $s = 0.0959$.

It can be assumed that the sample comes from a population which is normal distributed.

a) A 90%-confidence interval for μ becomes?

||| **Solution**

Since the 95%-quantile, $t_{0.95}$ of the t-distribution with 15 degrees of freedom equals $t_{0.95} = 1.753$ (found in Python as `stats.t.ppf(0.95,15)`), we get

$$180.05 \pm 1.753 \cdot \frac{0.0959}{\sqrt{16}} = [180.00, 180.10].$$

Or everything in Python:

```
x = np.array([180.02, 180.00, 180.01, 179.97, 179.92, 180.05, 179.94,
180.10,
180.24, 180.12, 180.13, 180.22, 179.96, 180.10, 179.96, 180.06])
print(np.mean(x))

180.05


print(np.std(x, ddof=1))

0.09591663046625545


print(stats.t.ppf(0.95,df = 15))

1.7530503556925547


(CI_low,CI_high) = stats.ttest_1samp(x, popmean=180).confidence_interval(
  confidence_level=0.9)
print(CI_low,CI_high)

180.0079633292111 180.09203667078893
```

b) A 99%-confidence interval for σ becomes?

||| Solution

We use the formula for the variance confidence interval

$$\left[\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}}; \frac{(n-1)s^2}{\chi^2_{\alpha/2}}\right],$$

where the quantiles come from a χ^2-distribution with $\nu = n - 1 = 15$ degrees of freedom

$$\left[\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}}; \frac{(n-1)s^2}{\chi^2_{\alpha/2}}\right] = \left[\frac{15 \cdot 0.0092}{\chi^2_{0.995}}; \frac{15 \cdot 0.0092}{\chi^2_{0.005}}\right] = \left[\frac{15 \cdot 0.0092}{32.801}; \frac{15 \cdot 0.0092}{4.601}\right]$$

$$= [0.00421; 0.02999].$$

And for the standard deviation:

$$\left[\sqrt{0.00421};\ \sqrt{0.02999}\right] = [0.065;\ 0.173].$$

In Python:

```python
print(stats.chi2.ppf((0.995,0.005),df=15))

[32.80132065  4.60091557]


CI_var = 15*np.var(x, ddof=1)/stats.chi2.ppf((0.995,0.005),df=15)
print(CI_var)

[0.00420715 0.02999403]


CI_std = np.sqrt(CI_var)
print(CI_std)

[0.06486253 0.17318785]
```

3.3 Concrete items (hypothesis testing)

||||| **Exercise 3.3**   **Concrete items (hypothesis testing)**

This is a continuation of Exercise 1, so the same setting and data is used (read the initial text of it).

a) To investigate whether the requirement to the mean is fulfilled (with $\alpha = 5\%$), the following hypothesis should be tested

$$H_0 : \mu = 3000$$
$$H_1 : \mu \neq 3000.$$

Or similarly asked: what is the evidence against the null hypothesis?

||||| **Solution**

This is a one-sample situation. In Python it could be handled by:

```
x = np.array([3003,3005,2997,3006,2999,2998,3007,3005,3001])
t_obs, p_val = stats.ttest_1samp(x,popmean=3000)
print(t_obs)
```

```
1.8877596148972005
```

```
print(p_val)
```

```
0.09575551807233935
```

from which the answer can be seen. One could also explicitly do it as

$$t_{\text{obs}} = \frac{3002.333 - 3000}{3.708/\sqrt{9}} = 1.885.$$

And then find the p-value as (using a t-distribution with $\nu = 8$ degrees of freedom)

$$2 * P(T > 1.885) = 0.096.$$

(in Python as: `2*(1-stats.t.cdf(1.885,df=8))`). So although there is a weak evidence aginst the null, cf. the p-value interpretation table in Section **??**, when using an α of 0.05 the null hypothesis is not rejected, but must be accepted.

b) What would the level $\alpha = 0.01$ critical values be for this test, and what are the interpretation of these?

|||| **Solution**

The critical values would be $\pm t_{0.995} = \pm 3.355$:

```
print(stats.t.ppf(0.995, df=8))
```

```
3.3553873313333957
```

This means that, in a new experiment, the standardized difference between the data and the null hypothesis, also called t_{obs}, must be either larger than 3.355 or smaller than -3.355 to lead to a significant result of the experiment.

c) What would the level $\alpha = 0.05$ critical values be for this test (compare also with the values found in the previous question)?

|||| **Solution**

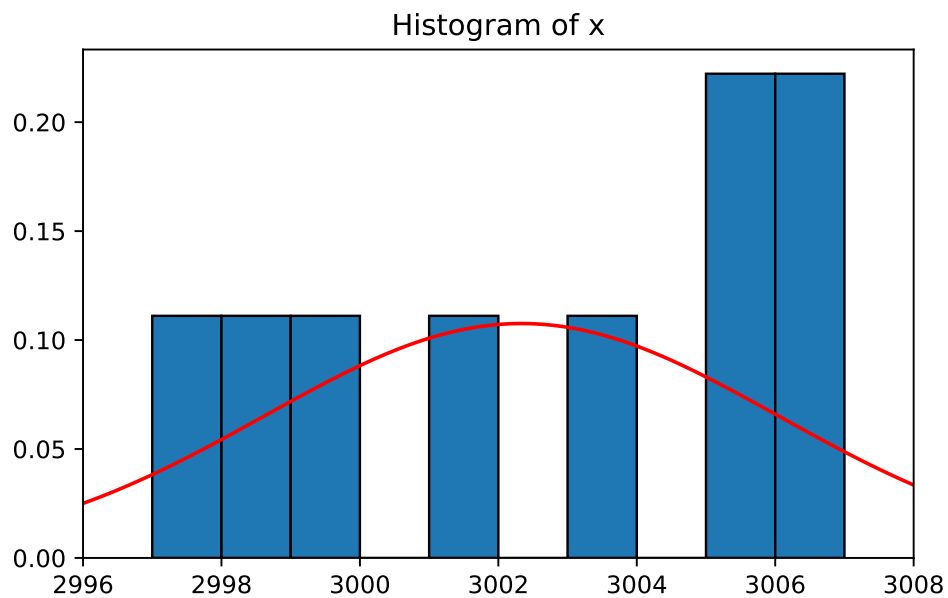The critical values would be $\pm t_{0.975} = \pm 2.306$:

```
print(stats.t.ppf(0.975, df=8))
```

```
2.3060041350333704
```

This means that, again in a new experiment, it is easier to detect an effect with significance level $\alpha = 0.05$ than on level $\alpha = 0.01$.

d) Investigate, by some plots, whether the data here appears to be coming from a normal distribution (as assumed until now)?
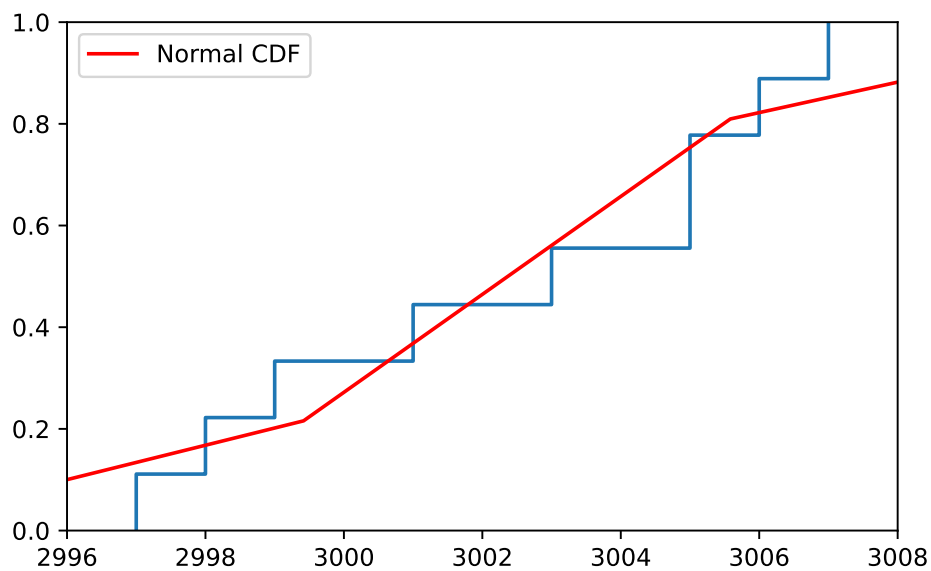
▕▕▕▕ **Solution**

```python
x = np.array([3003,3005,2997,3006,2999,2998,3007,3005,3001])
plt.hist(x, edgecolor='black', density=True)
# add normal curve
xlim = plt.xlim(2996, 3008)
x_plot = np.linspace(2996, 3008, 100)
y = stats.norm.pdf(x_plot, np.mean(x), np.std(x,ddof=1))
plt.plot(x_plot,y, color='red')
plt.title("Histogram of x")
plt.show()
```
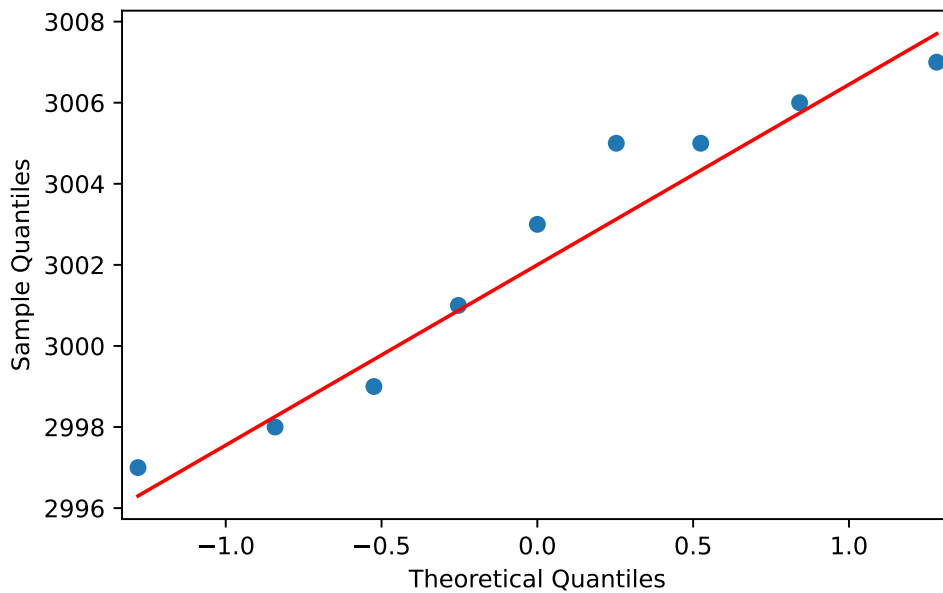

Histogram of x

▏▏▏▏ **Solution**

```python
x_plot = np.linspace(0.9*min(x), 1.1*max(x), 100)
y = stats.norm.cdf(x_plot, np.mean(x), np.std(x, ddof=1))
plt.ecdf(x)
plt.plot(x_plot, y, color='red', label='Normal CDF')
plt.legend()
xlim = plt.xlim(2996, 3008)
plt.show()
```


▏▏▏▏ **Solution**

```python
sm.qqplot(x, line ='q')
plt.show()
```
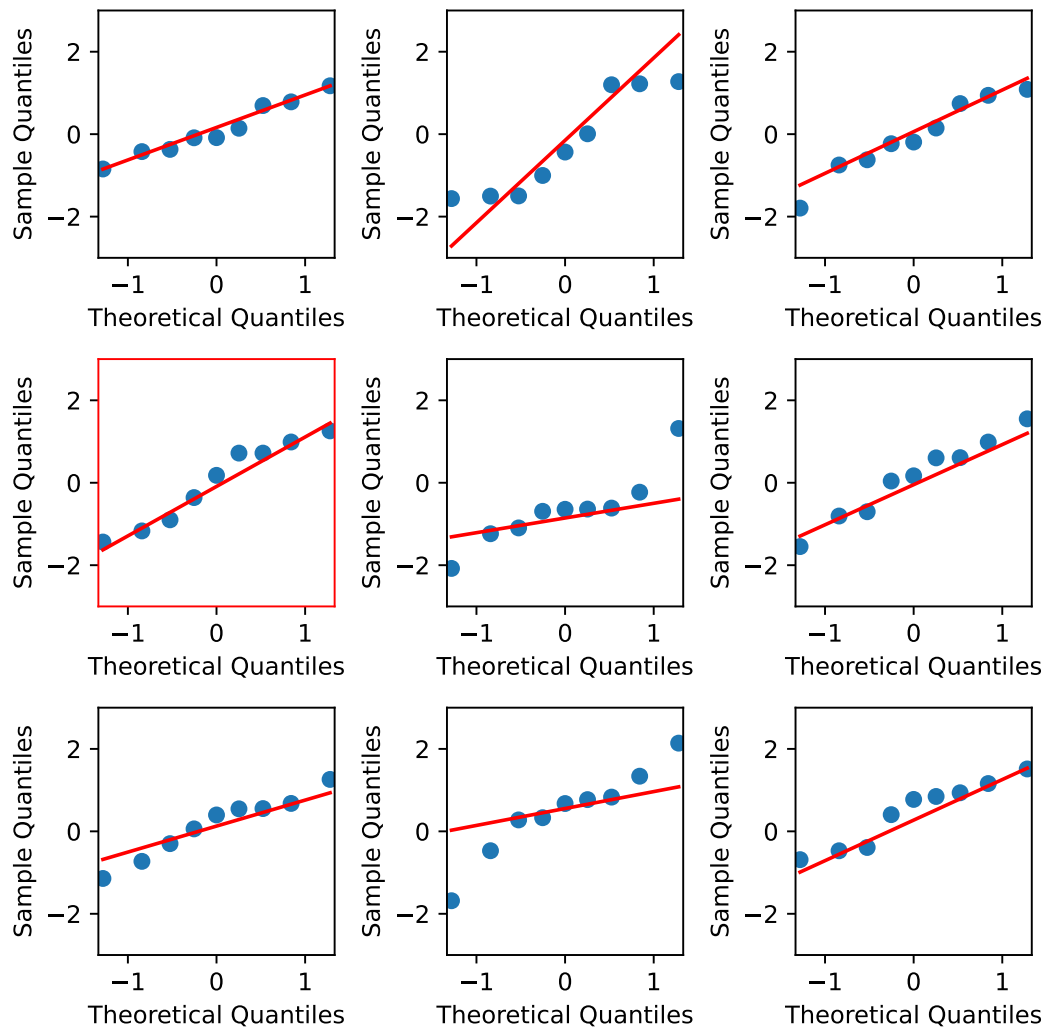
‖‖ **Solution**

Compare with 9 simulated ones

```python
W = stats.randint.rvs(0,3,size=2)
# Generate 9 plots with QQ-plots from the assumed distribution
n = len(x)
fig, axs = plt.subplots(3,3)
for ax in axs.flat:
    sm.qqplot(stats.norm.rvs(size=n),line="q",ax=ax)
    ylim = ax.set(ylim=[-3,3])
# Clear the randomly selected plot and replace with the plot from data

axs[W[0],W[1]].clear()
sm.qqplot((x-x.mean())/x.std(ddof=1),line="q",ax=axs[W[0],W[1]]) #
Standardized
ylim = axs[W[0],W[1]].set(ylim=[-3,3])
temp = plt.setp(axs[W[0],W[1]].spines.values(), color="red")
# Generate plot
plt.tight_layout()
plt.show()
```
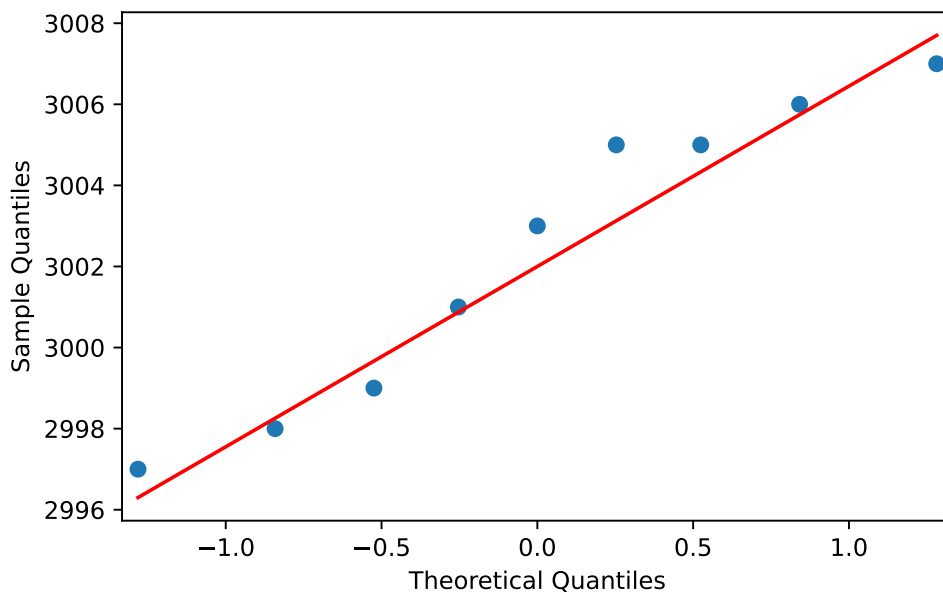
||| Solution

The nine data points do not differ more from the line than what truly normally distributed samples of size $n = 9$ do, so we cannot falsify the normality assumption (Did we prove normality?...no, we accept that they are normal distributed (like when we accept the null hypothesis)).

e) Assuming that you, maybe among different plots, also did the normal q-q plot above, the question is now: What exactly is plotted in that plot? Or more specifically: what are the x- and y-coordinates of e.g. the two points to the lower left in this plot?

▎▎▎ **Solution**

Let us look at the Normal q-q plot again:

```
sm.qqplot(x, line ='q')
plt.show()
```


The y-coordinates of the nine points from left to right in the plot are the ordered observations $x_{(1)}, \ldots, x_{(9)}$:

```
print(np.sort(x))

[2997 2998 2999 3001 3003 3005 3005 3006 3007]
```
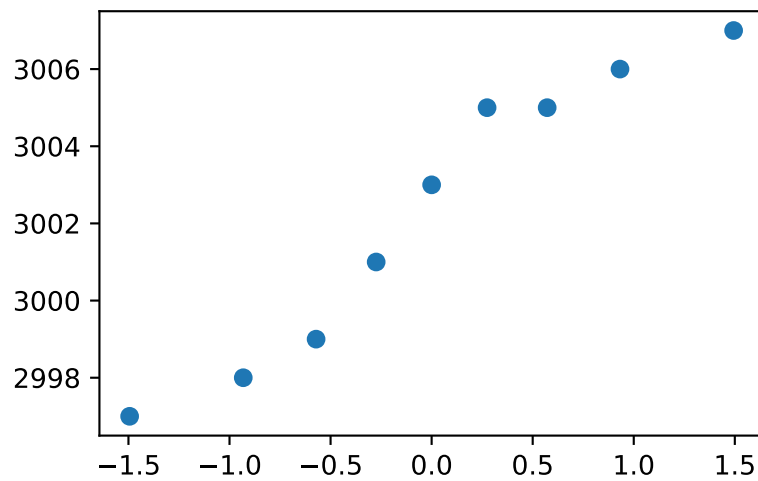
▎▎▎ **Solution**

The x-coordinates are quantiles from the standard normal distribution. Method **??** tell us exactly which quantiles, that are used by Python (in the current case there is less than $n = 10$ observations):

$$p_i = \frac{i - 3/8}{9 + 1/4}, \ i = 1, \ldots, 9$$

```
i = np.arange(1,10)
pi = (i - 3/8)/(9+1/4)
print(stats.norm.ppf(pi))

[-1.49415491 -0.93197131 -0.57163753 -0.27439149  0.          0.27439149
   0.57163753  0.93197131  1.49415491]


plt.plot(stats.norm.ppf(pi), np.sort(x), 'o')
plt.show()
```

3.4 Aluminum profile (hypothesis testing)

||| **Exercise 3.4**   **Aluminium profile (hypothesis testing)**

We use the same setting and data as in Exercise 2, so read the initial text of it.

a) Find the evidence against the following hypothesis:
$$H_0 : \mu = 180.$$

||| **Solution**

First we compute the observed t-statistic
$$t_{obs} = \frac{180.05 - 180}{0.0959/\sqrt{16}} = 2.085,$$
and the p-value is
$$p\text{-value} = 2 \cdot P(T > 2.085) = 2 \cdot 0.0273 = 0.055.$$

```
mu0 = 180
tobs = (180.05 - mu0)/(0.0959/np.sqrt(16))
print(tobs)
```

```
2.085505735141246
```

```
print(2*(1-stats.t.cdf(tobs,df=15)))
```

```
0.05452040573484673
```

Or in completely solve by Python:

```
x = np.array([180.02, 180.00, 180.01, 179.97, 179.92, 180.05, 179.94,
180.10,
180.24, 180.12, 180.13, 180.22, 179.96, 180.10, 179.96, 180.06])
t_obs,p_val = stats.ttest_1samp(x,popmean=180)
print(t_obs,p_val)
```

```
2.085144140571199 0.05455742828541606
```

‖ Hence there is *weak evidence against* H_0, cf. Table **??**.

b) If the following hypothesis test is carried out

$$H_0 : \mu = 180,$$
$$H_1 : \mu \neq 180.$$

What are the level $\alpha = 1\%$ critical values for this test?

‖‖‖ **Solution**

The critical values are the 0.005 and 0.995 quantiles of the t-distribution with $\nu = n - 1 = 15$ degrees of freedom, $\pm t_{0.995} = \pm 2.947$:

```
print(stats.t.ppf(0.995, df=15))
```

```
2.946712883338615
```

c) What is the 99%-confidence interval for μ?

‖‖‖ **Solution**

The formula gives

$$180.05 \pm t_{0.995} \frac{0.0959}{\sqrt{16}} = [179.98;\ 180.12].$$

```
(CI_low,CI_high) = stats.ttest_1samp(x, popmean=180).confidence_interval(
    confidence_level=0.99)
print(CI_low,CI_high)
```

```
179.9793403073197 180.12065969268033
```

d) Carry out the following hypothesis test

$$H_0 : \mu = 180,$$
$$H_1 : \mu \neq 180,$$

using $\alpha = 5\%$.

‖‖ **Solution**

We already found the p-value $= 0.055$ above, and as this is larger than α we cannot reject the null hypothesis of $\mu = 180$.

3.5   Transport times

||| **Exercise 3.5**   **Transport times**

A company, MM, selling items online wants to compare the transport times for two transport firms for delivery of the goods. To compare the two companies recordings of delivery times on a specific route were made, with a sample size of n = 9 for each firm. The following data were found:

$$\text{Firm A: } \bar{y}_A = 1.93 \text{ d and } s_A = 0.45 \text{ d,}$$
$$\text{Firm B: } \bar{y}_B = 1.49 \text{ d and } s_B = 0.58 \text{ d.}$$

note that d is the SI unit for days. It is assumed that data can be regarded as stemming from normal distributions.

a) We want to test the following hypothesis

$$H_0 : \mu_A = \mu_B$$
$$H_1 : \mu_A \neq \mu_B$$

What is the p-value, interpretation and conclusion for this test (at $\alpha = 5\%$ level)?

||| **Solution**

This is the independent samples (non-directional) t-test (Welch) in Method **??** (or Method **??**). We first find the t-test statistic

$$t_{\text{obs}} = \frac{1.93 - 1.49}{\sqrt{0.45^2/9 + 0.58^2/9}} = 1.8,$$

and then the degrees of freedom

$$\nu = \frac{\left(\frac{0.45^2}{9} + \frac{0.58^2}{9} \right)^2}{\frac{(0.45^2/9)^2}{8} + \frac{(0.58^2/9)^2}{8}} = 15.0,$$

and the p-value is then found to

$$2 \cdot P(T > 1.8) = 0.0922,$$

using a t-distribution with 15.0 degrees of freedom. So even though, according to Table **??**, there is weak evidence against the null hypothesis, when we use $\alpha = 0.05$ we cannot reject the null hypothesis of the two firms being equally fast. In Python it can be found as

```
ms = np.array([1.93,1.49])
vs = np.array([0.45**2, 0.58**2])
ns = np.array([9,9])
t_obs = (ms[0]-ms[1])/np.sqrt(vs[0]/ns[0]+vs[1]/ns[1])
nu = sum(vs / ns)**2 / sum((vs / ns)**2 / (ns - 1))
print(t_obs)
```

```
1.7981248351046575
```

```
print(nu)
```

```
15.069646790733499
```

```
print(2*(1-stats.t.cdf(t_obs,df=nu)))
```

```
0.09221726347624926
```

b) Find the 95% confidence interval for the mean difference $\mu_A - \mu_B$.

‖‖ Solution

We need the degrees of freedom - we found that number above: $\nu = 15.0$. Since the relevant t-quantile then is, using $\nu = 15.0$,

```
print(stats.t.ppf(0.975,df=15.0))
```

```
2.13149545559323
```

$$t_{0.975} = 2.131,$$

the confidence interval becomes

$$1.93 - 1.49 \pm 2.131 \sqrt{0.45^2/9 + 0.58^2/9}.$$

Which becomes

```
CI = ms[0]-ms[1] + np.array([-1,1])*stats.t.ppf(0.975,df=15.0)*np.sqrt(
  vs[0]/9+vs[1]/9)
print(CI)
```

```
[-0.08156434  0.96156434]
```

Hence the answer is: $[-0.082; 0.962]$.

c) What is the power of a study with $n = 9$ observations in each of the two samples of detecting a potential mean difference of 0.4 between the firms (assume that $\sigma = 0.5$ and that we use $\alpha = 0.05$)?

|||| **Solution**

```
n = 9
delta = 0.4
sd = 0.5
alpha = 0.05
print(smp.TTestIndPower().solve_power(nobs1=n, effect_size = delta/sd,
alpha=alpha))
```

```
0.35794895150011175
```

So the power is only 0.36 - not nearly good enough for a reasonable study.

d) What effect size (mean difference) could be detected with $n = 9$ observations in each of the two samples with a power of 0.8 (assume that $\sigma = 0.5$ and that we use $\alpha = 0.05$)?

||| Solution

```
n = 9
power = 0.8
sd = 0.5
alpha = 0.05
effect = smp.TTestIndPower().solve_power(nobs1=n, alpha=alpha,
power=power)
delta = effect*sd
print(delta)

0.7034623345972915
```

So a potential mean difference of 0.70 is detectable with probability 0.8 by such a study.

e) How large a sample size (from each firm) would be needed in a new investigation, if we want to detect a potential mean difference of 0.4 between the firms with probability 0.90, that is with power=0.90 (assume that $\sigma = 0.5$ and that we use $\alpha = 0.05$)?

||| Solution

```
power = 0.9
sd = 0.5
delta = 0.4
alpha = 0.05
print(smp.TTestIndPower().solve_power(effect_size=delta/sd,
alpha=alpha, power=power))

33.82554383684386
```

So $n = 34$ in each sample would do the job!

3.6 Cholesterol

|||| **Exercise 3.6**   **Cholesterol**

In a clinical trial of a cholesterol-lowering agent, 15 patients' cholesterol (in mmol/L) has been measured before treatment and 3 weeks after starting treatment. Data are listed in the following table:

Patient	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Before	9.1	8.0	7.7	10.0	9.6	7.9	9.0	7.1	8.3	9.6	8.2	9.2	7.3	8.5	9.5
After	8.2	6.4	6.6	8.5	8.0	5.8	7.8	7.2	6.7	9.8	7.1	7.7	6.0	6.6	8.4

The following is run in Python:

```python
x1 = np.array([9.1, 8.0, 7.7, 10.0, 9.6, 7.9, 9.0, 7.1,
       8.3, 9.6, 8.2, 9.2, 7.3, 8.5, 9.5])
x2 = np.array([8.2, 6.4, 6.6, 8.5, 8.0, 5.8, 7.8, 7.2,
       6.7, 9.8, 7.1, 7.7, 6.0, 6.6, 8.4])

# Independent
t_obs,p_val = stats.ttest_ind(x1,x2,equal_var=False)
print(t_obs, p_val)

3.3206371615181602 0.0025776851099507973


# Paired
t_obs,p_val = stats.ttest_rel(x1,x2)
print(t_obs,p_val)

7.340653674886342 3.6722518645935146e-06
```

a) Can there, based on these data be demonstrated a significant decrease in cholesterol levels with $\alpha = 0.001$?

‖‖ **Solution**

This is clearly a *paired* setting so only the results from the last of the Python-calls are relevant, where we can read off the results:

The (non-directional) p-value is 0.00000367, so there is very strong evidence against the null hypothesis, and we can beyond any reasonable doubts conclude that the mean cholesterol level has decreased after the 3 weeks.

3.7 Pulse

||| **Exercise 3.7**   **Pulse**

13 runners had their pulse measured at the end of a workout and 1 minute after again and we got the following pulse measurements:

Runner	1	2	3	4	5	6	7	8	9	10	11	12	13
Pulse end	173	175	174	183	181	180	170	182	188	178	181	183	185
Pulse 1min	120	115	122	123	125	140	108	133	134	121	130	126	128

The following was run in Python:

```
Pulse_end = np.array([173,175,174,183,181,180,170,
                      182,188,178,181,183,185])
Pulse_1min = np.array([120,115,122,123,125,140,108,
                       133,134,121,130,126,128])
print(np.mean(Pulse_end))

179.46153846153845


print(np.mean(Pulse_1min))

125.0


print(np.std(Pulse_end, ddof=1))

5.189980485117207


print(np.std(Pulse_1min, ddof=1))

8.406346808612328


print(np.std(Pulse_end-Pulse_1min, ddof=1))

5.767948575466911
```

a) What is the 99% confidence interval for the mean pulse drop (meaning the drop during 1 minute from end of workout)?

|||| **Solution**

We use the paired sample t-test version of the confidence interval (see Section **??**) with $s_{\text{dif}} = 5.768$, $n = 13$ and 12 degrees of freedom for the t-quantile $t_{0.005}$:

```
print(stats.t.ppf(0.995, df=12))
```

```
3.0545395883368704
```

Or in completely in Python:

```
res = stats.ttest_rel(Pulse_end,Pulse_1min)
print(res.statistic, res.pvalue)
```

```
34.043970207871624 2.624171406710289e-13
```

```
# Confidence interval
(CI_low,CI_high) = res.confidence_interval(confidence_level=0.99)
print(CI_low,CI_high)
```

```
49.57506593037534 59.34801099270158
```

So, the answer is:

$$54.46 \pm 3.054 \cdot \frac{5.768}{\sqrt{13}} = [49.58; 59.35].$$

b) Consider now the 13 pulse end measurements (first row in the table). What is the 95% confidence interval for the standard deviation of these?

‖‖‖ **Solution**

Using **??** we find the 95% confidence interval for the variance to

$$\frac{12 \cdot 5.18998^2}{\chi^2_{0.975}} < \sigma^2 < \frac{12 \cdot 5.18998^2}{\chi^2_{0.025}},$$

which then for the standard deviation becomes

$$\sqrt{\frac{12 \cdot 5.18998^2}{23.34}} < \sigma < \sqrt{\frac{12 \cdot 5.18998^2}{4.40}}.$$

Or in Python:

```
ttest = stats.ttest_rel(Pulse_end,Pulse_1min)
print(ttest.statistic, ttest.pvalue)

34.043970207871624 2.624171406710289e-13


CI_var = (12 * np.var(Pulse_end,ddof=1)) / stats.chi2.ppf((0.975,0.025),
df=12)
CI_std = np.sqrt(CI_var)
print(CI_var)

[13.85077006 73.39834071]


print(CI_std)

[3.72166227 8.56728316]
```

So, the answer is that we accept that $\sigma \in [3.72; 8.57]$ or we could write $3.72 < \sigma < 8.57$.

3.8 Foil production

||| **Exercise 3.8**   **Foil production**

In the production of a certain foil (film), the foil is controlled by measuring the thickness of the foil in a number of points distributed over the width of the foil. The production is considered stable if the mean of the difference between the maximum and minimum measurements does not exceed 0.35 mm. At a given day, the following random samples are observed for 10 foils:

Foil	1	2	3	4	5	6	7	8	9	10
Max. in mm (y_{max})	2.62	2.71	2.18	2.25	2.72	2.34	2.63	1.86	2.84	2.93
Min. in mm (y_{min})	2.14	2.39	1.86	1.92	2.33	2.00	2.25	1.50	2.27	2.37
Max-Min (D)	0.48	0.32	0.32	0.33	0.39	0.34	0.38	0.36	0.57	0.56

The following statistics may potentially be used

$$\bar{y}_{max} = 2.508, \ \bar{y}_{min} = 2.103, \ s_{y_{max}} = 0.3373, \ s_{y_{min}} = 0.2834, \ s_D = 0.09664.$$

a) What is a 95% confidence interval for the mean difference?

||| **Solution**

The 95% confidence interval is given by:

```
CI = (2.508 - 2.103) + np.array([-1, 1]) * stats.t.ppf(0.975, df=10-1)
* 0.09664 / np.sqrt(10)
print(CI)
```

```
[0.33586791 0.47413209]
```

The confidence interval contains those values of the mean difference that we believe in based on the data. Notice that the CI contains $\mu_D = 0.35$, hence we know that we will not reject the null hypothesis $H_0 = 0.35$, which is the next question.

b) How much evidence is there that the mean difference is different from 0.35? State the null hypothesis, t-statistic and p-value for this question.

‖‖‖ Solution

The t-statistic is found using Method 9:

$$t_{\text{obs}} = \frac{(2.508 - 2.103) - 0.35}{0.09664/\sqrt{10}} = 1.80.$$

```python
t_obs = ((2.508 - 2.103) - 0.35) / (0.09664 / np.sqrt(10))
print(t_obs)
```

```
1.7997234200047632
```

The p-value for this assessment is

$$p\text{-value} = 2 \cdot P(T > t_{\text{obs}}) = 0.1054.$$

where T has a t-distribution with 9 degrees of freedom:

```python
print(2 * (1-stats.t.cdf(t_obs,df=10-1)))
```

```
0.10543681559704599
```

According to Table ?? there is little or no evidence that against the null hypothesis that $\mu_D = 0.35$.

3.9 Course project

||| **Exercise 3.9** **Course project**

At a specific education it was decided to introduce a project, running through the course period, as a part of the grade point evaluation. In order to assess whether it has changed the percentage of students passing the course, the following data was collected:

	Before introduction of project	After introduction of project
Number of students evaluated	50	24
Number of students failed	13	3
Average grade point $\bar{x}$	6.420	7.375
Sample standard deviation s	2.205	1.813

a) As it is assumed that the grades are approximately normally distributed in each group, the following hypothesis is tested:

$$H_0 : \mu_{\text{Before}} = \mu_{\text{After}},$$
$$H_1 : \mu_{\text{Before}} \neq \mu_{\text{After}}.$$

The test statistic, the p-value and the conclusion for this test become?

||| **Solution**

The (Welch) t-test statistic for this setup is found using Method :

$$t = \frac{6.42 - 7.375}{\sqrt{2.205^2/50 + 1.813^2/24}} = -1.97.$$

From the t-distribution with $\nu = 54.4$ we can find that the (non-directional) p-value is

$$2 \cdot P(T > 1.97) = 0.054.$$

```
print(2*(1-stats.t.cdf(1.97,54.4)))
```

```
0.05393786409352663
```

In Python we could do it by the following:

```python
ms = np.array([6.42,7.375])
vs = np.array([2.205**2, 1.813**2])
ns = np.array([50,24])
t_obs = (ms[0]-ms[1])/np.sqrt(vs[0]/ns[0]+vs[1]/ns[1])
nu = sum(vs / ns)**2 / sum((vs / ns)**2 / (ns - 1))
print(t_obs)
```

```
-1.9733867395024223
```

```python
print(nu)
```

```
54.38590559936726
```

On a 5% level we cannot conclude a significant difference in the grade point means before and after.

b) A 99% confidence interval for the mean grade point difference is?

|||| **Solution**

We need the degrees of freedom - we found that number above: $\nu = 54.4$. Since the relevant t-quantile then is, using $\nu = 54.4$,

```python
print(stats.t.ppf(0.995,54.4))
```

```
2.669269209086161
```

Hence $t_{0.995} = 2.669$, and the confidence interval becomes

$$6.42 - 7.375 \pm 2.669\sqrt{2.205^2/50 + 1.813^2/24},$$

Which is:

```
CI = (6.42-7.375) + np.array([-1,1])*stats.t.ppf(0.995,df=54.4)*np.sqrt(
     2.205**2/50 + 1.813**2/24)
print(CI)
```

```
[-2.24676509  0.33676509]
```

and the answer is: we accept that the mean difference in the interval [-2.247; 0.337].

c) A 95% confidence interval for the grade point standard deviation after the introduction of the project becomes?

||| **Solution**

The confidence interval formula for a sample variance is used WITH the square-root applied to everything:

$$\frac{\sqrt{(n-1)\cdot s^2}}{\sqrt{\chi^2_{0.975}}} < \sigma < \frac{\sqrt{(n-1)\cdot s^2}}{\sqrt{\chi^2_{0.025}}},$$

so

$$\frac{\sqrt{23\cdot 1.813^2}}{\sqrt{38.076}} < \sigma_{\text{After}} < \frac{\sqrt{23\cdot 1.813^2}}{\sqrt{11.689}}.$$

The values found in Python:

```
CI_std = np.sqrt((23*1.813**2)/stats.chi2.ppf((0.975,0.025),23))
print(CI_std)
```

```
[1.40908823 2.54320499]
```

So the answer is, that we accept that $\sigma_{\text{After}} \in [1.41; 2.54]$.

3.10   Concrete items (sample size)

‖‖ **Exercise 3.10   Concrete items (sample size)**

This is a continuation of Exercise 1, so the same setting and data is used (read the initial text of it).

a) A study is planned of a new supplier. It is expected that the standard deviation will be approximately 3, that is, $\sigma = 3$ mm. We want a 90% confidence interval for the mean value in this new study to have a width of 2 mm. How many items should be sampled to achieve this?

‖‖ **Solution**

We use the sample size formula (Method **??**) with wanted margin of error $ME = 1$ (as the width of the confidence interval is twice the margin of error)

$$n = \left(\frac{z_{0.95}\sigma}{ME}\right)^2 = \left(\frac{1.645 \cdot 3}{1}\right)^2 = 24.35.$$

(in Python: `stats.norm.ppf(0.95)` to get $z_{0.95} = 1.645$). Hence the answer becomes: at least 25.

b) Answer the sample size question above but requiring the 99% confidence interval to have the (same) width of 2 mm.

‖‖ **Solution**

The same formula as above:

$$n = \left(\frac{z_{0.995}\sigma}{ME}\right)^2 = \left(\frac{2.576 \cdot 3}{1}\right)^2 = 59.72.$$

(using Python: `stats.norm.ppf(0.995)` to get $z_{0.995} = 2.576$). Hence the answer becomes: at least 60.

c) (Warning: This is a difficult question about a challenging abstraction - do not worry, if you do not make this one) For the two sample sizes found in the two previous questions find the probability that the corresponding confidence interval in the future study will actually be more than 10% wider than planned for (still assuming and using that the population variance is $\sigma^2 = 9$).

|||| **Solution**

The random width of the confidence interval is due to the randomly changing sample variance in the formula for the (random) half width of the interval

$$\text{'The half width of CI'} = t_{1-\alpha/2} \frac{S}{\sqrt{n}}.$$

The (sampling) distribution of the variance estimator, S, is a χ^2-distribution, as stated in Section **??** (around Equation (**??**)). Thus: let S^2 be the variance of a sample of size n from a normal distribution with variance $\sigma^2 = 3^2 = 9$. Then

$$\chi^2 = \frac{(n-1) \cdot S^2}{9},$$

is a stochastic variable following the χ^2-distribution with $v = n - 1$ degrees of freedom.

|||| **Solution**

So, as the wanted margin error was $ME = 1$, we are asked to first find with $\alpha = 0.10$ and $n = 25$, and hence $t_{0.95} = 1.711$ (in Python `stats.t.ppf(0.95, df=24)`)

$$
\begin{aligned}
P\left(\text{'Half width of CI'} > 1.1\right) &= P\left(t_{0.95} \cdot \frac{S}{\sqrt{n}} > 1.1\right) \\
&= P\left(t_{0.95}^2 \cdot \frac{S^2}{n} > 1.1^2\right) \\
&= P\left(t_{0.95}^2 \cdot \frac{9 \cdot \chi^2}{24 \cdot 25} > 1.1^2\right) \\
&= P\left(\chi^2 > \frac{1.1^2 \cdot 24 \cdot 25}{t_{0.95}^2 \cdot 9}\right) \\
&= P\left(\chi^2 > 27.56\right) \\
&= 0.28.
\end{aligned}
$$

```
print(stats.t.ppf(0.95, df=24))
```

```
1.7108820799094275
```

```
print(1.1**2*25*24/(stats.t.ppf(0.95, df=24)**2*9))
```

```
27.55839726451134
```

```
print(1-stats.chi2.cdf(1.1**2*25*24/(stats.t.ppf(0.95, df=24)**2*9),
24))
```

```
0.27910525870084113
```

Therefore in almost 30% of cases an experiment planned for a $90\% ME = 1$ would actually end up wih a confidence interval of half width more than 1.1.

For the 99% case, and $n = 60$ the same computation gives:

```
print(stats.t.ppf(0.995, df=59))
```

```
2.6617587534986837
```

```
print(1.1**2*60*59/(stats.t.ppf(0.995, df=59)**2*9))
```

```
67.17516486334
```

```
print(1-stats.chi2.cdf(1.1**2*60*59/(stats.t.ppf(0.995, df=59)**2*9),
59))
```

```
0.21736213064733256
```

So in this case it only happens in 22% of the cases. In the next part of this topic we will learn how we can plan experiments such that we are more in control of the risk of the experiments not really meeting our needs.

d) Now a new experiment is to be planned. In the first part above, given some wanted margin of error (ME) a sample size of $n = 25$ was found. What are each of the probabilities that an experiment with $n = 25$ will detect effects corresponding to ("end up significant for") $\mu_1 = 3001, 3002, 3003$

respectively? Assume that we use the typical $\alpha = 0.05$ level and that $\sigma = 3$?

|||| Solution

We can only solve this by the in-built `smp.TTestPower().solve_power()` function in Python. If we specify everything but the power - it will compute the power for us. With e.g. $\mu_1 = 3001$, we have that $\mu_0 - \mu_1 = -1$, so in the Python-function the `delta` should be set to either 1 or -1. And in fact one can insert a list of the three relevant deltas to get the three answers by a single call to Python: (But remember that the Python-function takes the effect size, which corresponds to $\frac{delta}{\sigma}$)

```
n = 25
delta = np.array([1,2,3])
sd = 3
alpha = 0.05
print(smp.TTestPower().solve_power(nobs=n, effect_size = delta/sd,
alpha=alpha))

[0.3596554  0.89201704 0.99768458]
```
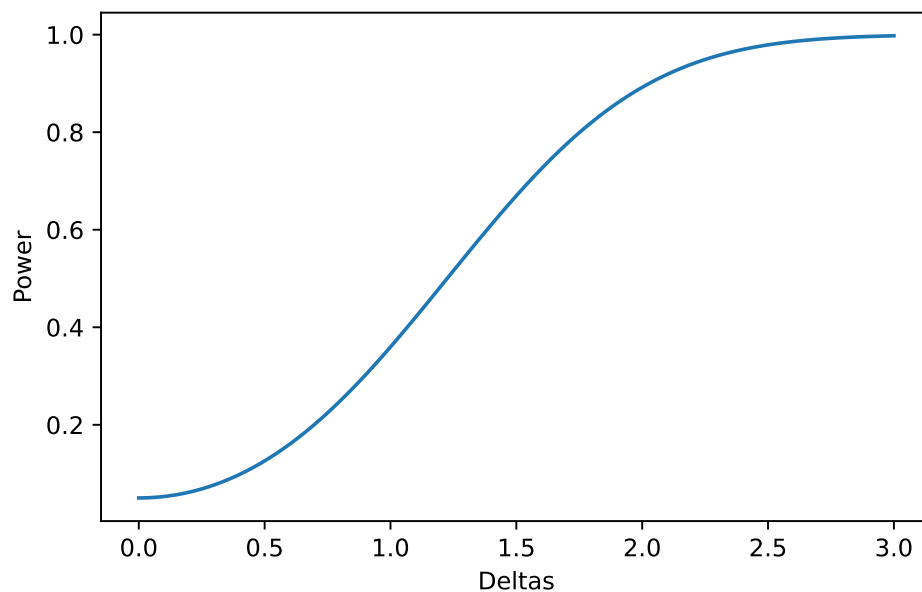
So the three probabilities are 0.36, 0.89 and 0.998. A difference of 1 would not be reasonably detectable by this experiment but a difference of 2 has a high probability of being detected and even more so for 3.

|||| Solution

A plot of all possible powers for all possible effect sizes could now easily be made:

```
deltas = np.linspace(0,3,100)
powers25 = smp.TTestPower().power(effect_size=deltas/sd, nobs=n,
alpha=alpha)
plt.plot(deltas,powers25)
plt.xlabel("Deltas")
plt.ylabel("Power")
plt.show()
```

e) One of the sample size computation above led to $n = 60$ (it is not so important how/why). Answer the same question as above using $n = 60$.

|||| Solution

```
print(smp.TTestPower().solve_power(nobs=60, effect_size =
np.array([1,2,3])/3, alpha=0.05))

[0.71897939 0.99909076 1.        ]
```
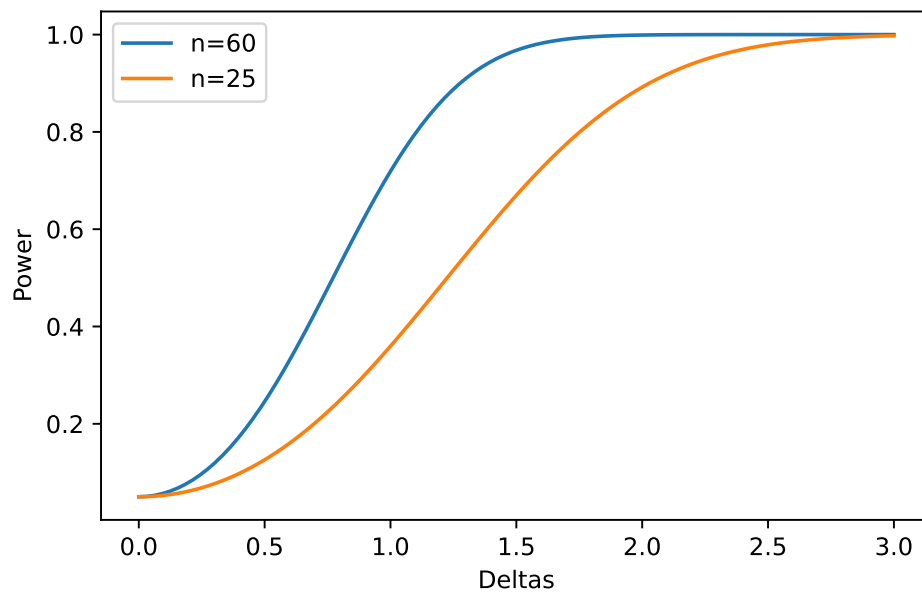
So the three probabilities are 0.72, 0.999 and 1.00000. A difference of 1 would still not be reasonably detectable by this experiment but a difference of 2 and 3 has extremely high power.

|||| **Solution**

A plot of all possible powers for all possible effect sizes could now easily be made:

```python
deltas = np.linspace(0,3,100)
powers60 = smp.TTestPower().power(effect_size=deltas/3, nobs=60,
alpha=0.05)
plt.plot(deltas,powers60, label="n=60")
plt.plot(deltas,powers25, label="n=25")
plt.xlabel("Deltas")
plt.ylabel("Power")
plt.legend()
plt.show()
```


f) What sample size would be needed to achieve a power of 0.80 for an effect of size 0.5?

|||| **Solution**

The approximate sample size formula, Method 3.47, is only formally given for one-sided tests, so at first sight again we have no formulas for this (as we test two-sided here). However, in fact the formula also works nicely for the two-sided case,

substituting $z_{1-\alpha}$ by $z_{1-\alpha/2}$, as the only error that is made would be the left hand tail of the rejection area, cf. the power plots on page 46, which essentially is zero for relevant effect sizes. So IF we use this adapted (and no-where stated) formula we would get:

$$n = \left(3 \cdot \frac{z_{0.8} + z_{0.975}}{0.5} \right)^2 = 282.6.$$

```
n = (3*(stats.norm.ppf(0.8)+stats.norm.ppf(0.975))/0.5)**2
print(n)
```

```
282.55967043656716
```

However, it would actually be (slightly) better to simply use the Python-function again as the Python-function does not rely on any normal approximation of the more correct t-distributions:

```
n = smp.TTestPower().solve_power(effect_size=0.5/3, alpha=0.05,
power=0.8)
print(n)
```

```
284.48568685343037
```

This would lead to $n = 285$. So 0.5 is an effect size that it would be pretty expensive to be able to detect.

g) Assume that you only have the finances to do an experiment with $n = 50$. How large a difference would you be able to detect with probability 0.8 (i.e. Power= 0.80)?

▊ Solution

```
effect = smp.TTestPower().solve_power(nobs=50, alpha=0.05, power=0.8)
sd = 3
delta = effect*sd
print(delta)
```

```
1.2125490152103784
```

The answer is 1.21, so a true alternative mean of 2998.8 (or smaller) or 3001.2 (or larger) would be detected by this experiment with probability 0.80

3.11   Concrete items 2

|||| **Exercise 3.11**   **Concrete items 2**

A construction company receives concrete items for a construction. The length of the items are assumed reasonably normally distributed, with mean $\mu = 3000$ mm and standard deviation $\sigma = 3$ mm.

a) In a construction process, 5 concrete items are joined together to a single construction with a length which is then the complete length of the 5 concrete items. It is very important that the length of this new construction is within 15 m plus/minus 2 cm. How often will it happen that such a construction will be more than 2 cm away from the 15 m target (assume that the population mean concrete item length is $\mu = 3000$ mm and that the population standard deviation is $\sigma = 3$)?

|||| **Solution**

Let X_i denote the length of an individual concrete item: $X_i \sim N(3000, 3^2)$.
Let Y denote the length of the joined construction. So

$$Y = \sum_{i=1}^{5} X_i,$$

From Theorem 2.40 we know that Y will also follow a normal distribution:
$Y_i \sim N(\mu_Y, \sigma_Y^2)$.

Using the rules for mean and variance calculations from Section **??**, we can find that

$$\mu_Y = E(Y) = \sum_{i=1}^{5} E(X_i) = \sum_{i=1}^{5} 3000 = 5 \cdot 3000 = 15000$$

and

$$\sigma_Y^2 = V(Y) = \sum_{i=1}^{5} V(X_i) = \sum_{i=1}^{5} 3^2 = 5 \cdot 9 = 45$$

and we have:

$$Y \sim N(15000, 45).$$

We now calculate the probability that Y is either larger than 15020 mm (15 m plus 2 cm) or smaller than 14980 mm (15 m minus 2 cm):

$$P(Y > 15020) = 1 - F(15020)$$
$$P(Y < 14980) = F(14980)$$

where $F(x)$ refers to the cdf of a normal distribution with mean 15000 and variance 45. The probabilities are found using Python:

```python
print(1-stats.norm.cdf(15020, loc=15000, scale=np.sqrt(45)))
```

```
0.001434556396038289
```

```python
print(stats.norm.cdf(14980, loc=15000, scale=np.sqrt(45)))
```

```
0.0014345563960383074
```

Therefore the total probability of the joined construction to deviate more than 2 cm from the target is $2 \cdot 0.001435 = 0.002869$.

The construction company, wants to check that the delivered concrete items do in fact have an average length of 3000 mm and a standard deviation of 3 mm (still assuming that the lengths follow a normal distribution). The company samples 9 items from a delivery which are then measured for control. The following measurements (in mm) are found:

3003	3005	2997	3006	2999	2998	3007	3005	3001

b) Compute the following three statistics: the sample mean, the sample standard deviation and the standard error of the mean. Explain the difference between the sample standard deviation and the standard error of the mean.

|||| **Solution**

From the data we get the sample mean and sample standard deviation

$$\bar{x} = 3002.33\text{mm and } s = 3.708\text{mm}.$$

```
x = np.array([3003, 3005, 2997, 3006, 2999, 2998, 3007, 3005, 3001])
print(np.mean(x))

3002.3333333333335

print(np.std(x, ddof=1))

3.708099243547832
```

From Definition **??**, we get the standard error of the mean as

$$SE_{\bar{x}} = \frac{3.708}{\sqrt{9}} = 1.236.$$

The sample standard deviation relates to the variation of the (lengths of the) individual concrete items. The standard error of the mean relates to the variation of the estimated sample mean (i.e. if the construction company was to do a new sample (of the same size, $n = 9$) we would not expect the get the exact same value of $\bar{x}$, but we would expect the variance of such estimates of $\bar{x}$ to be around 1.236^2).

c) Find the 95% confidence interval for the mean μ.

||| **Solution**

To compute the 95% confidence interval we need the 2.5%-quantile (a negative number) or the 97.5%-quantile (a positive number) from the t-distribution with $n - 1 = 8$ degrees of freedom. Since the 97.5%-quantile, $t_{0.975}$ of the t-distribution with $n - 1 = 8$ degrees of freedom equals $t_{0.975} = 2.306$ (found in Python as: `stats.t.ppf(0.975, 8)`, we get the 95% confidence interval

$$3002.33 \pm 2.306 \cdot \frac{3.708}{\sqrt{9}} \Leftrightarrow [2999.5; 3005.2].$$

Or everything in Python:

```
x = np.array([3003, 3005, 2997, 3006, 2999, 2998, 3007, 3005, 3001])
xbar = np.mean(x)
se_xbar = np.std(x, ddof=1)/np.sqrt(9)
t975 = stats.t.ppf(0.975, df=8)
mu_low = xbar - t975*se_xbar
mu_high = xbar + t975*se_xbar
print([mu_low, mu_high])


[2999.4830359370885, 3005.1836307295785]
```

d) Find the 99% confidence interval for μ. Compare with the 95% one from above and explain why it is smaller/larger!

▕▏▎▍ **Solution**

Since the 99.5%-quantile, $t_{0.995}$ of the t-distribution with 8 degrees of freedom equals $t_{0.995} = 3.355$ (found in Python as: `stats.t.ppf(0.995,8)`, we get

$$3002.33 \pm 3.335 \cdot \frac{3.708}{\sqrt{9}} \Leftrightarrow [2998.2; 3006.5].$$

Or everything in Python:

```
x = np.array([3003, 3005, 2997, 3006, 2999, 2998, 3007, 3005, 3001])
xbar = np.mean(x)
se_xbar = np.std(x, ddof=1)/np.sqrt(9)
t995 = stats.t.ppf(0.995, df=8)
mu_low = xbar - t995*se_xbar
mu_high = xbar + t995*se_xbar
print([mu_low, mu_high])


[2998.1859635916244, 3006.4807030750426]
```

The 99% confidence interval is larger than the 95% one, as the consequence of wanting to be more confident on capturing the true mean μ will make us having to state a larger interval.

e) Find the 95% confidence intervals for the variance σ^2 and the standard deviation σ.

▐▐▐▐ **Solution**

We use the formula for the variance confidence interval

$$\left[\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}} ; \frac{(n-1)s^2}{\chi^2_{\alpha/2}} \right].$$

where the quantiles come from a χ^2-distribution with $\nu = n - 1 = 8$ degrees of freedom

$$\left[\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}} ; \frac{(n-1)s^2}{\chi^2_{\alpha/2}} \right] = \left[\frac{8 \cdot 3.708^2}{\chi^2_{0.975}} ; \frac{8 \cdot 3.708^2}{\chi^2_{0.025}} \right] = \left[\frac{8 \cdot 13.75}{17.535} ; \frac{8 \cdot 13.75}{2.180} \right]$$

$$= [6.273; 50.465].$$

And for the standard deviation:

$$\left[\sqrt{6.273}; \sqrt{50.465} \right] = [2.50; 7.10].$$

In Python:

```
print(stats.chi2.ppf((0.975,0.025), df=8))

[17.53454614  2.17973075]


CI_var = 8*np.var(x, ddof=1)/stats.chi2.ppf((0.975,0.025), df=8)
print(CI_var)

[ 6.27333032 50.46494854]


CI_std = np.sqrt(CI_var)
print(CI_std)

[2.50466172 7.10386856]
```

f) Based on the 95% confidence intervals, what should the construction company conclude about the delivered concrete items? Can they trust that the

lengths of the individual items have mean 3000 mm and standard deviation 3 mm?

|||| **Solution**

While the construction company cannot be absolutely sure about the exact true values of μ and σ, they find that the 95% confidence interval for μ includes the value 3000 mm and the 95% confidence interval for σ includes the value 3 mm, so it seems that these values *could* be the true values (even though the exact estimates ($\bar{x}$ and s) are not exactly equal to 3000 and 3).

g) What would be the conclusion (based on 95% confidence intervals) if similar sample estimates had been based on a much larger sample? Say the construction company makes a sample of $n = 100$ concrete items and from this sample they obtain $\bar{x} = 3002.3$ and $s = 3.7$.

|||| **Solution**

If this was the case the 95% confidence intervals would be:

```python
xbar = 3002.3
se_xbar = 3.7/np.sqrt(100)
t975 = stats.t.ppf(0.975, df=99)
mu_low = xbar - t975*se_xbar
mu_high = xbar + t975*se_xbar
print([mu_low, mu_high])


[3001.565839727942, 3003.034160272058]


print(stats.chi2.ppf((0.975,0.025), df=99))


[128.42198864  73.36108019]


CI_var = 99*3.7**2/stats.chi2.ppf((0.975,0.025), df=99)
CI_std = np.sqrt(CI_var)
print(CI_std)


[3.24862533 4.29819844]
```

And the construction company would therefore suspect (and probably conclude) that the true population μ and the true population σ were both a little larger than the expected values of 3000 mm and 3 mm.

3.12 Concrete items 2 (hypothesis testing)

▐▐▐▐ **Exercise 3.12** **Concrete items (hypothesis testing)**

This is a continuation of Exercise 11:

A construction company receives concrete items for a construction. The length of the items are assumed reasonably normally distributed, with mean $\mu = 3000$ mm and standard deviation $\sigma = 3$ mm.

The company samples 9 items from a delivery which are then measured for control. The following measurements (in mm) are found:

3003	3005	2997	3006	2999	2998	3007	3005	3001

a) To investigate whether the requirement to the mean is fulfilled (with $\alpha = 5\%$), the following hypothesis should be tested

$$H_0 : \mu = 3000$$
$$H_1 : \mu \neq 3000.$$

Or similarly asked: what is the evidence against the null hypothesis?

▐▐▐▐ **Solution**

From Exercise 11 we have already found $\bar{x} = 3002.33$ and $s = 3.708$.
We calculate t_{obs}:

$$t_{obs} = \frac{3002.33 - 3000}{3.708/\sqrt{9}} = 1.885.$$

Then we calculate the corresponding p-value (using a t-distribution with $\nu = 9 - 1$ degrees of freedom):

$$p = 2 * P(T > 1.885) = 0.096.$$

(in Python we find: $P(T > 1.885) = $ `2*(1-stats.t.cdf(1.885,df=8))`).

Or solve using Python:

```
x = np.array([3003, 3005, 2997, 3006, 2999, 2998, 3007, 3005, 3001])
t_obs,p_val = stats.ttest_1samp(x,popmean=3000)
print(t_obs,p_val)

1.8877596148972005 0.09575551807233935
```

Conclusion: There is a "weak evidence against the null-hypothesis" (see table 3.1), and using a significance level α of 0.05 the null hypothesis is not rejected.

b) At significance level $\alpha = 0.01$, what are the critical values for t_{obs}? And what is the interpretation of these critical values?

▐▐▐ **Solution**

The critical values would be $\pm t_{0.995} = \pm 3.355$:

```
print(stats.t.ppf(0.995, df=8))

3.3553873313333957
```

This means that (for a sample of size $n = 9$), the calculated t_{obs}, must be either larger than 3.355 or smaller than -3.355 to lead to a significant result at significance level $\alpha = 0.01$.

c) At significance level $\alpha = 0.05$, what are the critical values for t_{obs}? And what is the interpretation of these critical values (compare also with the values found in the previous question)?

▏▎▍▌ **Solution**

The critical values would be $\pm t_{0.975} = \pm 2.306$:

```
print(stats.t.ppf(0.975, df=8))

2.3060041350333704
```
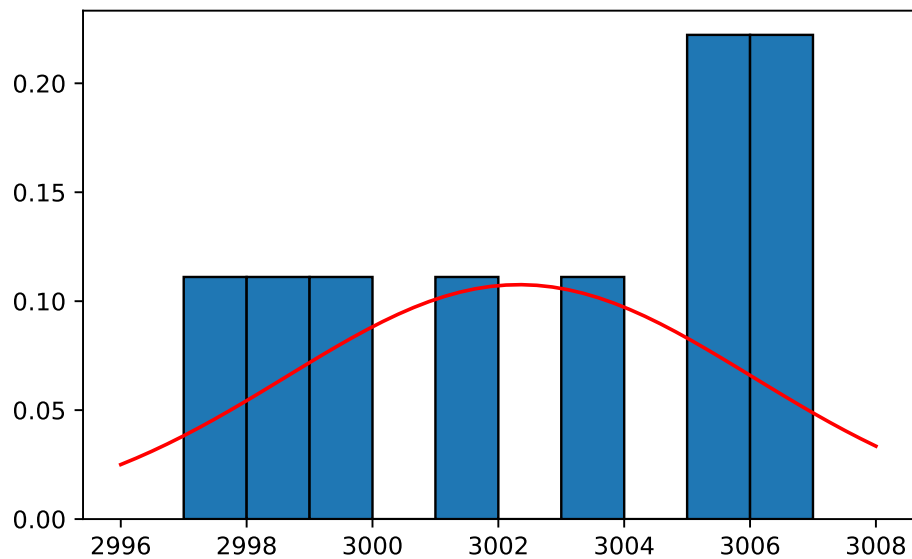
This means that (for a sample of size $n = 9$), the calculated t_{obs}, must be either larger than 2.306 or smaller than -2.306 to lead to a significant result at significance level $\alpha = 0.05$.

Comparing with the result from the previous question, we see that a smaller α is related to larger (absolute values of) critical values. When choosing a smaller α we demand the observed t_{obs} to be "more extreme" before we decide to reject the null hypothesis.

d) The company would now like to check the assumption that the data follows a normal distribution. Plot a histogram of the data and compare to a normal distribution pdf with same mean and standard deviation. Also plot the ecdf and compare to a normal distribution cdf with same mean and standard deviation (ecdf/cdf comparison has the advantage that no histogram-binsize is needed).
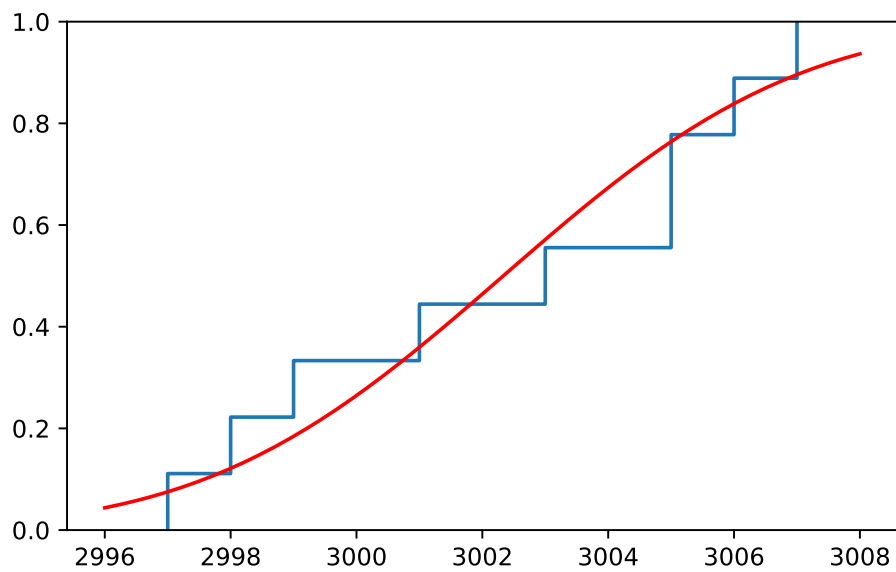
▏▎▍▌ **Solution**

```
x = np.array([3003,3005,2997,3006,2999,2998,3007,3005,3001])
plt.hist(x, edgecolor='black', density=True)
# add normal curve
x_values = np.linspace(2996, 3008)
pdf_values = stats.norm.pdf(x_values, loc=x.mean(),
scale=x.std(ddof=1))
plt.plot(x_values, pdf_values, color='red')
plt.show()
```
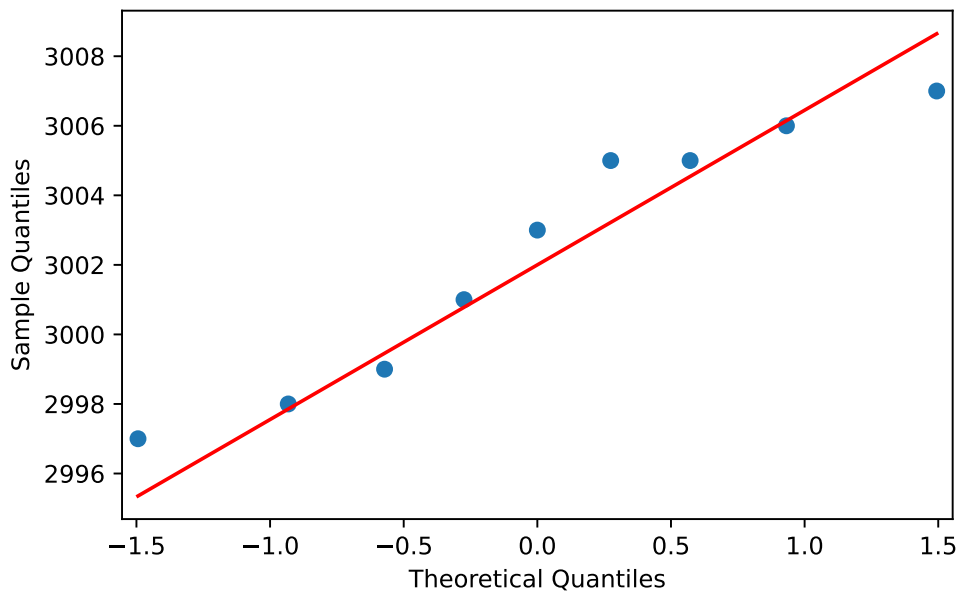
▏▏▏▏ **Solution**

```python
x = np.array([3003,3005,2997,3006,2999,2998,3007,3005,3001])
plt.ecdf(x)
# add normal curve
x_values = np.linspace(2996, 3008)
cdf_values = stats.norm.cdf(x_values, loc=x.mean(),
scale=x.std(ddof=1))
plt.plot(x_values, cdf_values, color='red')
plt.show()
```

e) Plot a QQ-plot of the data (see Method 3.42). Do you think the normal assumption is valid?

▥ **Solution**

```python
sm.qqplot(x, line ='q',a=3/8)
plt.show()
```
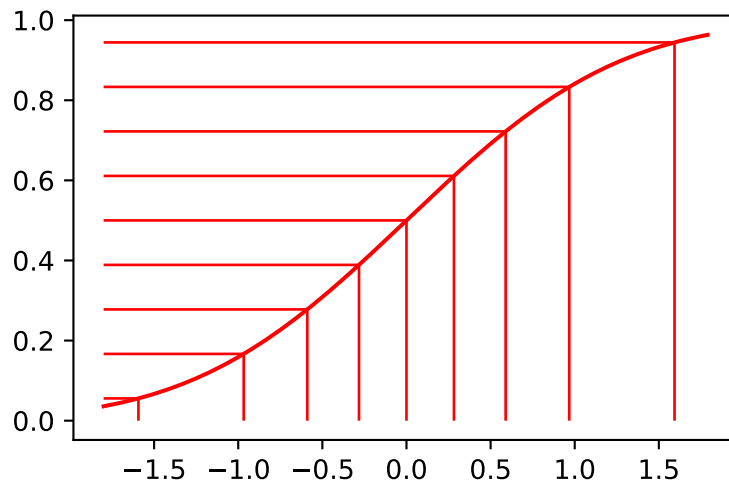
f) Take a closer look at your QQ-plot. What is actually plotted on the y-axis? And what is plotted on the x-axis?
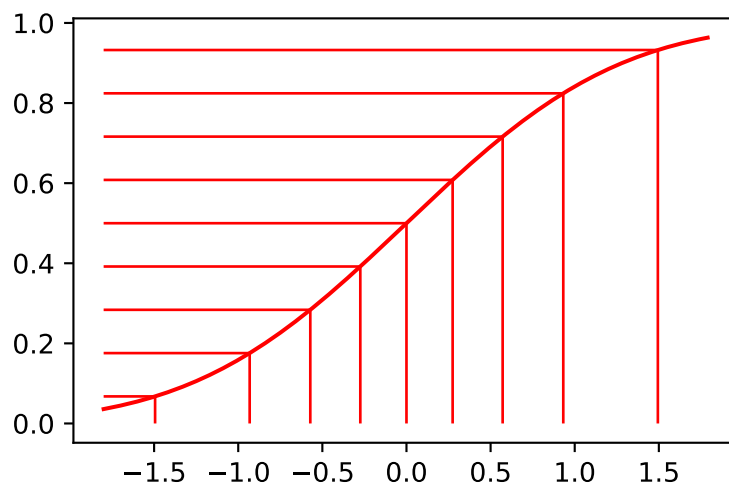
▥ Solution

The values plotted along the y-axis are the (sorted) values in the sample ($y_i = \{2997, 2998, 2999, 3001, 3003, 3005, 3005, 3006, 3007\}$).

Each value in the sample represents $1/9$'th of the total dataset (there are 9 data-points in total). Looking at Method 3.42, we see that the usual definition of p_i for datapoint number i is $p_i = (i - 0.5)/n$, so for our dataset this would be: $p_i = \{0.5/9, 1.5/9, 2.5/9, 3.5/9, 4.5/9, 5.5/9, 6.5/9, 7.5/9, 8.5/9\}$.

The usual values to be plotted on the x-axis are the corresponding **quantiles in a standard normal distribution**. These values can be computed in Python as `stats.norm.ppf`(p_i), and are visualized in the following plot (of the Standard normal cdf):

However, when $n \leq 10$ it is custom to choose values of p_i that are slightly more compressed towards the middle $p_i = (i - 3/8)/(n + 1/4)$ as visualized in the following plot (of the Standard normal cdf):

(did you specify a=3/8 in your QQ-plot in the previous question? If not the values will not be the same).

g) In order to evaluate the QQ-plot visually, plot QQ-plots of 9 simulated random samples from a normal distribution (with sample size $n = 9$ as in the data). Then replace one of the plots (e.g, the middle one) with the QQ-plot of the real data and evaluate visually. Does the QQ-plot of the real data look "as good" as the simulated ones?

You can use the following code:

```python
# Enter data into variable x:
x = np.array([3003,3005,2997,3006,2999,2998,3007,3005,3001])

# Generate 9 plots
fig, axs = plt.subplots(3, 3, figsize=(10, 10))

# Create 9 QQ plots of random normal samples
for ax in axs.flat:
    sm.qqplot(stats.norm.rvs(size=9), line="q", ax=ax)
    ax.set_ylim([-3, 3])

# Replace middle plot with real data
axs[1,1].clear()
sm.qqplot((x - x.mean())/x.std(ddof=1), line="q", ax=axs[1,1])
axs[1,1].set_ylim([-3, 3])

# Highlight middle plot with red spines
plt.setp(axs[1,1].spines.values(), color="red")

# Generate the plot
plt.tight_layout()
plt.show()
```
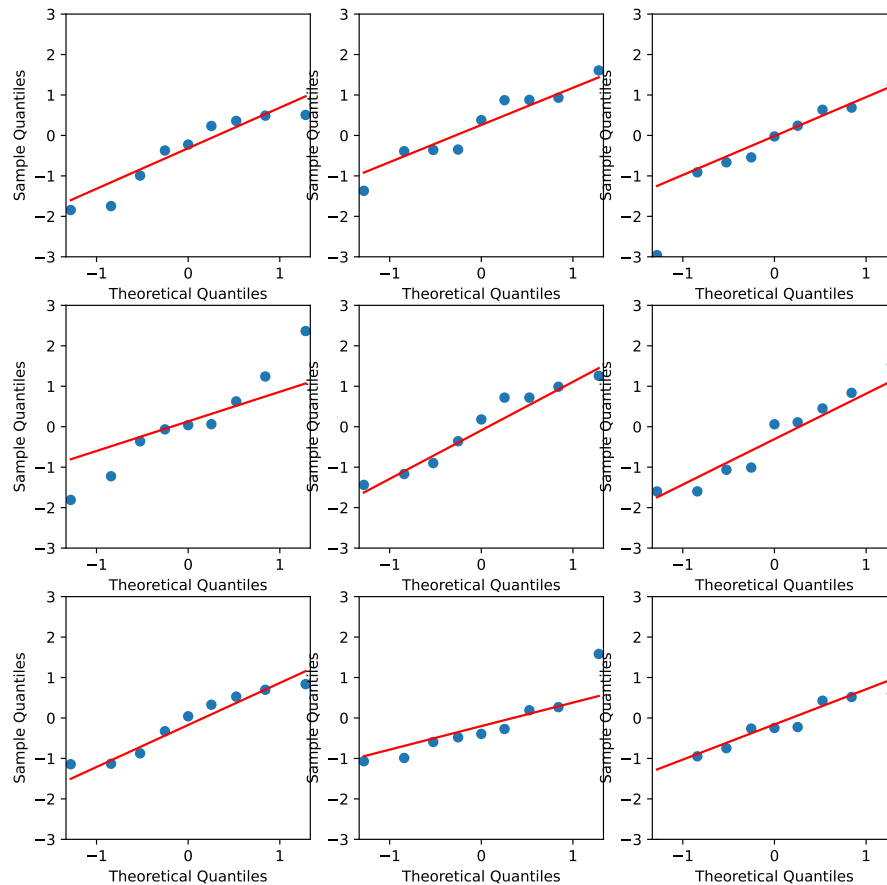
▐▐▐▐ **Solution**

The nine data points do not differ more from the line than what truly normally distributed samples of size $n = 9$ do, so from this visual inspection and comparison we cannot falsify the normality assumption (Did we prove normality?...no, we accept that they are normal distributed (like when we accept the null hypothesis)).