Chapter 2

Probability and simulation (solutions to exercises)

Contents

```
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import statsmodels as sm
import statsmodels.stats.proportion as smprop
```

2.1 Discrete random variable

|||| Exercise 2.1 Discrete random variable

a) Let X be a stochastic variable. When running the Python-command `stats.binom.pmf(4, 10, 0.6)` Python returns 0.1115. written as:

```
print(stats.binom.pmf(4, 10, 0.6))


0.11147673600000009
```

What distribution is applied and what does 0.1115 represent?

|||| Solution

The distribution applied is the binomial distribution with $n = 10$ observations and $p = 0.6$ the probability for success. The value, 0.1115 output by Python is the value of the probability density function (pdf) for $x = 4$, hence the probability of getting exactly 4 successes in 10 draws with replacement with a success probability of 60%.

b) Let X be the same stochastic variable as above. The following are results from Python:

```
print(stats.binom.cdf(4,10,0.6))

0.16623861760000005

print(stats.binom.cdf(5,10,0.6))

0.3668967424000001
```

Calculate the following probabilities: $P(X \leq 5)$, $P(X < 5)$, $P(X > 4)$ and $P(X = 5)$.

|||| Solution

```
## P(X <= 5)
print(stats.binom.cdf(5,10,0.6))


0.3668967424000001


## P(X < 5)
print(stats.binom.cdf(4,10,0.6))


0.16623861760000005


## P(X > 4)
print(1 - stats.binom.cdf(4,10,0.6))


0.8337613824


## P(X = 5)
print(stats.binom.cdf(5,10,0.6) - stats.binom.cdf(4,10,0.6))


0.20065812480000003
```

c) Let X be a stochastic variable. From Python we get:

```
print(stats.poisson.pmf(k=4,mu=3))


0.16803135574154085
```

What distribution is applied and what does 0.16803 represent?

|||| Solution

The Poisson distribution and the value is the probability of getting $x = 4$ events per interval when the average events per interval $\lambda = 3$ (i.e. the mean).

d) Let X be the same stochastic variable as above. The following are results from Python:

```
print(stats.poisson.cdf(4,3))

0.8152632445237722

print(stats.poisson.cdf(5,3))

0.9160820579686966
```

Calculate the following probabilities: $P(X \leq 5)$, $P(X < 5)$, $P(X > 4)$ and $P(X = 5)$.

|||| Solution

```
## P(X <= 5))
print(stats.poisson.cdf(5,3))


0.9160820579686966


## P(X < 5)
print(stats.poisson.cdf(4,3))


0.8152632445237722


## P(X > 4)
print(1 - stats.poisson.cdf(4,3))


0.1847367554762278


## P(X = 5)
print(stats.poisson.cdf(5,3) - stats.poisson.cdf(4,3))


0.10081881344492438
```

2.2 Course passing proportions

|||| Exercise 2.2 Course passing proportions

a) If a passing proportion for a course given repeatedly is assumed to be 0.80 on average, and there are 250 students who are taking the exam each time, what is the expected value, μ and standard deviation, σ, for the number of students who do not pass the exam for a randomly selected course?

|||| Solution

If X is the number of students not passing a randomly selected course, this random variable follows the binomial distribution with $n = 250$ and $p = 0.20$, so we use the formula for the mean and variance of the binomial

$$\mu = np = 0.2 \cdot 250 = 50, \quad \sigma^2 = np(1-p) = 250 \cdot 0.2 \cdot 0.8 = 40 = 6.32^2.$$

So the answer is: $\mu = 50$ and $\sigma = 6.32$.

2.3 Notes in a box

|||| Exercise 2.3 Notes in a box

A box contains 6 notes:

On 1 of the notes there is the number 1
On 2 of the notes there is the number 2
On 2 of the notes there is the number 3
On 1 of the notes there is the number 4

Two notes are drawn at random from the box, and the following random variable is introduced: X, which describes the number of notes with the number 4 among the 2 drawn. The two notes are drawn without replacement.

a) The mean and variance for X, and $P(X = 0)$ are?

|||| Solution

X follows the hypergeometric distribution (2.24) with $N = 6$, $a = 1$, and $n = 2$ so the mean and variance formula (2.25) for this distribution is used to find

$$\mu_x = n\frac{a}{N} = 2/6,$$

and

$$\sigma_x^2 = n\frac{a(N-a)}{N^2}\frac{N-n}{N-1} = 2\frac{1\cdot(6-1)\cdot(6-2)}{6^2\cdot(6-1)} = \frac{2\cdot 5\cdot 4}{36\cdot 5} = 8/36 = 2/9.$$

And the hypergeometric probability formula (2-25) gives

$$P(X=0) = \frac{\binom{1}{0}\binom{5}{2}}{\binom{6}{2}} = \frac{5\cdot 4\cdot 2}{2\cdot 6\cdot 5} = 2/3.$$

So the correct answer is: $\mu_x = 1/3$, $\sigma_x^2 = 2/9$ and $P(X = 0) = 2/3$.

b) The 2 notes are now drawn with replacement. What is the probability that none of the 2 notes has the number 1 on it?

Solution

The binomial pdf (2-20) is used in Python:

```
print(stats.binom.pmf(0,n=2,p=1/6))


0.6944444444444443
```

Another way is, since it is zero successes, then there is $5/6$ probability of not getting a success and since we want that to happen in both two draws, then we can simply multiply the probabilities

$$P(X=0)=\frac{5}{6}\cdot\frac{5}{6}=\frac{25}{36}$$

So the correct answer is: 0.694 or $\frac{25}{36}$.

2.4 Consumer survey

|||| Exercise 2.4 Consumer survey

In a consumer survey performed by a newspaper, 20 different groceries (products) were purchased in a grocery store. Discrepancies between the price appearing on the sales slip and the shelf price were found in 6 of these purchased products.

a) At the same time a customer buys 3 random (different) products within the group consisting of the 20 goods in the store. The probability that no discrepancies occurs for this customer is?

|||| Solution

Let X denote the number of discrepancies when purchasing 3 random (different) products within the group of the 20 goods in the store. X then follows the hypergeometric distribution (NOT the binomial!!) (why not binomial: because you don't potentially by two goods of the same kind - you DO buy 3 DIFFERENT ones and hence having bought one - you do NOT "put it back" again and the randomly select - it is WITHOUT replacement). Therefore

$$P(X=0) = \frac{\binom{6}{0}\binom{14}{3}}{\binom{20}{3}} = \frac{14\cdot 13\cdot 12\cdot 3\cdot 2}{20\cdot 19\cdot 18\cdot 3\cdot 2} = \frac{91}{15\cdot 19} = 0.3192982.$$

Hence the answer is: 0.319.

2.5 Hay delivery quality

|||| Exercise 2.5 Hay delivery quality

A horse owner receives 20 bales of hay in a sealed plastic packaging. To control the hay, 3 bales of hay are randomly selected, and each checked whether it contains harmful fungal spores.

It is believed that among the 20 bales of hay 2 bales are infected with fungal spores. A random variable X describes the number of infected bales of hay among the three selected.

a) The mean of X, (μ_X), the variance of X, (σ_X^2) and $P(X \geq 1)$ are?

|||| Solution

The hypergeometric distribution with $N = 20$, $a = 2$ and $n = 3$ is used ("sampling without replacement"). First the mean and variance formulas for the hypergeometric distribution gives

$$\mu_x = 3\frac{2}{20} = 0.3,$$

and

$$\sigma_x^2 = 3\frac{2}{20}(1 - \frac{2}{20})(\frac{20-3}{20-1}) = 0.2415789.$$

Then we find

$$P(X \geq 1) = 1 - P(X = 0) = 1 - \frac{\binom{18}{3}\binom{2}{0}}{\binom{20}{3}} = 0.2842,$$

So the answer is: $\mu_x = 0.3$, $\sigma_x^2 = 0.242$ and $P(X \geq 1) = 0.2842$.

b) Another supplier advertises that no more than 1% of his bales of hay are infected. The horse owner buys 10 bales of hay from this supplier, and decides to buy hay for the rest of the season from this supplier if the 10 bales are error-free.
What is the probability that the 10 purchased bales of hay are error-free, if 1% of the bales from a supplier are infected (p_1) and the probability that the 10 purchased bales of hay are error-free, if 10% of the bales from a supplier are infected (p_{10})?

|||| Solution

We use the binomial distribution(s) ("sampling with replacement"="sampling from an infinte population").

We can use the pdf or cdf in Python. For p_1:

```
print(stats.binom.pmf(0,n=10,p=0.01))

0.9043820750088041

print(stats.binom.cdf(0,n=10,p=0.01))

0.9043820750088045
```

and for p_2:

```
print(stats.binom.pmf(0,n=10,p=0.1))

0.34867844009999993

print(stats.binom.cdf(0,n=10,p=0.1))

0.34867844009999993
```

This 10 independent events in series where each must be non-infected, i.e. a success is then $1 - 0.01$ and $1 - 0.1$, and can thus also be calculated simply by multiplying the probability of success in each event

$$p_1 = P(X_1 = 0) = 0.99^{10} = 0.9044,$$

and

$$p_{10} = P(X_{10} = 0) = 0.90^{10} = 0.3487.$$

So the answer becomes: $P_1 = 0.9044$ and $P_{10} = 0.3487$.

2.6 Newspaper consumer survey

|||| Exercise 2.6 Newspaper consumer survey

In a consumer survey performed by a newspaper, 20 different groceries (products) were purchased in a grocery store. Discrepancies between the price appearing on the sales slip and the shelf price were found in 6 of these purchased products.

a) Let X denote the number of discrepancies when purchasing 3 random (different) products within the group of the 20 products in the store. What is the mean and variance of X?

|||| Solution

We must use the hypergeometric distribution, since we draw $n = 3$ of the $N = 20$ products, where $a = 6$ have discrepancies, with no replacement (we cannot draw the same product twice).

The mean and variance of the hypergeometric distribution is found in Theorem 2.25, thus

$$\mu_X = n \cdot \frac{a}{N} = 3\frac{6}{20} = 0.90,$$

and

$$\sigma_X^2 = n \cdot \frac{a}{N}\left(1 - \frac{a}{N}\right)\left(\frac{N-n}{N-1}\right) = 3\frac{6}{20} \cdot \left(1 - \frac{6}{20}\right)\left(\frac{20-3}{20-1}\right) = 0.5636842.$$

Hence the answer is: $\mu_X = 0.90$ and $\sigma_X^2 = 0.56$.

2.7 A fully automated production

|||| Exercise 2.7 A fully automated production

On a large fully automated production plant items are pushed to a side band at random time points, from which they are automatically fed to a control unit. The production plant is set up in such a way that the number of items sent to the control unit on average is 1.6 item pr. minute. Let the random variable X denote the number of items pushed to the side band in 1 minute. It is assumed that X follows a Poisson distribution.

a) What is the probability that there will arrive more than 5 items at the control unit in a given minute is?

|||| Solution

With $\lambda = 1.6$, we find that

$$P(X > 5) = 1 - P(X \leq 5) = 1 - 0.994 = 0.006$$

where the 0.994 can be found with Python by:

```
print(1-stats.poisson.cdf(k=5,mu=1.6))
```

```
0.006040291111581331
```

So the answer is: approximately 0.6%.

b) What is the probability that no more than 8 items arrive to the control unit within a 5-minute period?

Solution

With $\lambda_{5minutes} = 8$, we find that

$$P(X \leq 8) = 0.593$$

where the 0.593 can be found by Python:

```
print(stats.poisson.cdf(k=8,mu=8))


0.5925473414375915
```

So the answer is: approximately 59.3%.

2.8 Call center staff

|||| Exercise 2.8 Call center staff

The staffing for answering calls in a company is based on that there will be 180 phone calls per hour randomly distributed. If there are 20 calls or more in a period of 5 minutes the capacity is exceeded, and there will be an unwanted waiting time, hence there is a capacity of 19 calls per 5 minutes.

a) What is the probability that the capacity is exceeded in a random period of 5 minutes?

|||| Solution

The 60 minutes mean of 180 calls corresponds to a 5 minutes mean of $\mu_{5min} = 180/12 = 15$ and the event of exceeding capacity is the event of observing at least 20 calls within 5 minutes. Let X represent the number of calls within a randomly chosen 5 minutes interval, then we need to find $P(X \geq 20)$, which in Python:

```
print(1-stats.poisson.cdf(k=19, mu=15))


0.12478121503252493
```

So the correct answer is: $P(X \geq 20) = 0.125$, where $X \sim Po(15)$.

b) If the probability should be at least 99% that all calls will be handled without waiting time for a randomly selected period of 5 minutes, how large should the capacity per 5 minutes then at least be?

Solution

Let X (as above) represent the number of calls in a randomly chosen 5 minutes interval, i.e. $X \sim Po(15)$. It is required that

$$P(\text{“All calls will be handled”}) = P(X \leq x_{\text{capacity}}) \geq 0.99$$

where x_{capacity} must be the smallest capacity which keeps the probability above 0.99. Using Python to find $P(X \leq x_{\text{capacity}})$ for 22,23,...,26:

```
print(stats.poisson.cdf(k=[22,23,24,25,26], mu=15))


[0.967 0.981 0.989 0.994 0.997]


# Or directly using the 99th quantile
print(stats.poisson.ppf(q=0.99, mu=15))


25.0
```

shows that the first (smallest) capacity level achieving this is 25.

So the correct answer is: the capacity must be at least 25 per 5 minutes

2.9 Continuous random variable

|||| Exercise 2.9 Continuous random variable

a) The following Python commands and results are given:

```
print(stats.norm.cdf(2))

0.9772498680518208

print(stats.norm.cdf(2,1,1))

0.8413447460685429

print(stats.norm.cdf(2,1,2))

0.6914624612740131
```

Specify which distributionsare used and explain the resulting probabilities (preferably by a sketch).

|||| Solution

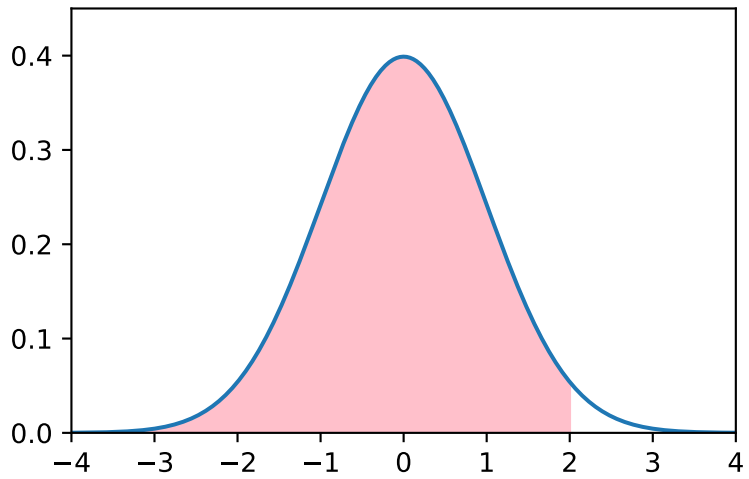
The normal distribution function (or normal cumulated density function cdf). The found probabilities are

- $P(X \leq 2)$ (or $P(X < 2)$) for $X \sim N(\mu = 0, \sigma^2 = 1)$
- $P(X \leq 2)$ (or $P(X < 2)$) for $X \sim N(\mu = 1, \sigma^2 = 1)$
- $P(X \leq 2)$ (or $P(X < 2)$) for $X \sim N(\mu = 1, \sigma^2 = 4)$

A sketch for the first, $P(Z \leq 2)$ (using Z indicates that is follows the standard normal distribution $N(0,1)$):

```
xseq = np.linspace(-4, 4, 1000)
yseq = stats.norm.pdf(xseq)

plt.plot(xseq, yseq)
xseq_fill = np.linspace(-4, 2, 1000) # fill from -4 to 2
yseq_fill = stats.norm.pdf(xseq_fill) # fill from -4 to 2
plt.fill_between(xseq_fill, yseq_fill, color='pink')
xlim = plt.xlim(-4, 4)
ylim = plt.ylim(0, 0.45)
plt.show()
```



b) What is the result of the following command: `stats.norm.ppf(stats.norm.cdf(2))`?

Solution

`stats.norm.ppf` and `stats.norm.cdf` are each others inverse so the result is the same as the argument: 2

c) The following Python commands and results are given:

```
print(stats.norm.ppf(0.975))
```

```
1.959963984540054

print(stats.norm.ppf(0.975,1,1))

2.959963984540054

print(stats.norm.ppf(0.975,1,2))

4.919927969080108
```

State what the numbers represent in the three cases (preferably by a sketch).

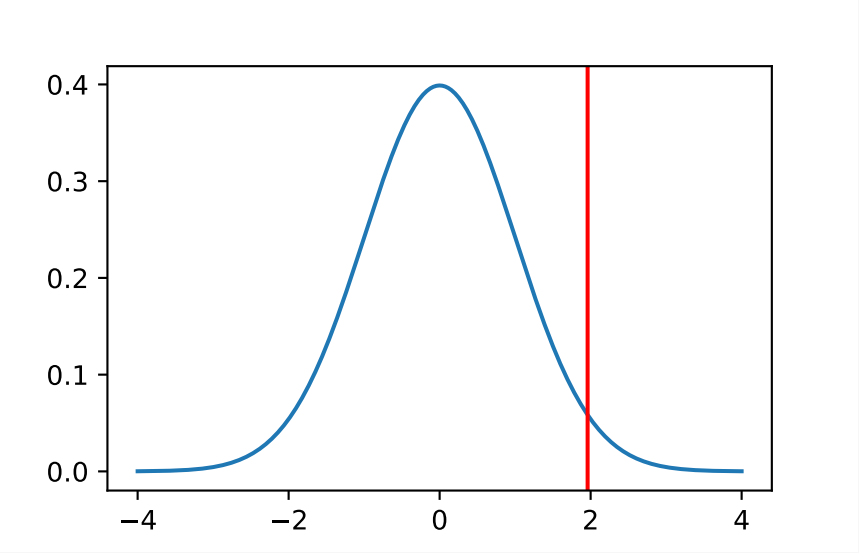
|||| Solution

The 97.5% percentiles for

- $N(\mu = 0, \sigma^2 = 1)$
- $N(\mu = 1, \sigma^2 = 1)$
- $N(\mu = 1, \sigma^2 = 4)$

A sketch for the first:

```
## Plot the standard normal distribution
xseq = np.linspace(-4, 4, 1000)
yseq = stats.norm.pdf(xseq)
plt.plot(xseq, yseq)
## Add the vertical line at the 0.975 quantile
plt.axvline(stats.norm.ppf(0.975), color='red')
plt.show()
```



2.10 The normal pdf

|||| Exercise 2.10 The normal pdf

a) Which of the following statements regarding the probability density function of the normal distribution $N(1, 2^2)$ is false?

1. The total area under the curve is equal to 1.0
2. The mean is equal to 1^2
3. The variance is equal to 2
4. The curve is symmetric about the mean
5. The two tails of the curve extend indefinitely
6. Don't know

|||| Solution

We need to find the one false statement, and go through the claims one at a time:

1. True, the total area under the curve is one, since this is true for all probability distributions, see Definition 2.32
2. True. The mean value is one, and we have that $1^2 = 1$
3. False, the standard deviation is two and the variance is four
4. True, the distribution is symmetric around the mean value
5. True, the normal density is defined between $-\infty$ and ∞

Correct answer is 3.

Let X be normally distributed with mean 24 and variance 16

b) Calculate the following probabilities:
- $P(X \leq 20)$
- $P(X > 29.5)$
- $P(X = 23.8)$

|||| Solution

- $P(X \leq 20)$

```
print(stats.norm.cdf(20, loc=24, scale=4))


0.15865525393145707
```

- $P(X > 29.5)$

```
print(1 - stats.norm.cdf(29.5, loc=24, scale=4))


0.08456572235133564
```

- The probability of a continuous random variable to have the outcome equal to a single value is zero, i.e. $P(X = 23.8) = 0$.

2.11 Computer chip control

|||| Exercise 2.11 Computer chip control

A machine for checking computer chips uses on average 65 milliseconds per check with a standard deviation of 4 milliseconds. A newer machine, potentially to be bought, uses on average 54 milliseconds per check with a standard deviation of 3 milliseconds. It can be used that check times can be assumed normally distributed and independent.

a) What is the probability that the time savings per check using the new machine is less than 10 milliseconds is?

|||| Solution

Let $X_{\text{old}} \sim N(65, 4^2)$ and $X_{\text{new}} \sim N(54, 3^2)$. If we let U denote the time saving per check, we have that $U = X_{\text{old}} - X_{\text{new}}$. We now from Theorem 2.40 that a linear combination of normal random variables is also normal and from Theorem 2.56

$$\mathrm{E}(U) = \mathrm{E}(X_{\text{old}} - X_{\text{new}}) = \mathrm{E}(X_{\text{old}}) - \mathrm{E}(X_{\text{new}}) = 65 - 54 = 11,$$

and

$$\mathrm{V}(U) = \mathrm{V}(X_{\text{old}} - X_{\text{new}}) = \mathrm{V}(X_{\text{old}}) + \mathrm{V}(X_{\text{new}}) = 16 + 9 = 25.$$

Hence $U \sim N(11, 5^2)$.

We are asked to find $P(U < 10)$, so

```
print(stats.norm.cdf(x=10,loc=11,scale=5))
```

```
0.42074029056089696
```

Another way to solve this is via a transformation to the standard normal distribution

using Theorem 2.43. Then

$$\begin{aligned} P(U < 10) &= P\left(Z < \frac{10 - \mathrm{E}(U)}{\sqrt{\mathrm{V}(U)}}\right) \\ &= P\left(Z < \frac{10 - (65 - 54)}{\sqrt{3^2 + 4^2}}\right) \\ &= P\left(Z < \frac{-1}{5}\right) \\ &= P\left(Z < -0.2\right) \\ &= 0.4207, \end{aligned}$$

where the latter can be found in Python:

```
z = (10-11)/5
print(z)


-0.2


print(stats.norm.cdf(z))


0.42074029056089696
```

So the answer is: approximately 42%.

b) What is the mean (μ) and standard deviation (σ) for the total time use for checking 100 chips on the new machine is?

|||| Solution

Let U be the total time use for checking 100 chips on the new machine, that is

$$U = \sum_{i=1}^{100} X_i$$

where $X_i \sim N(54, 3^2)$. So we find, using mean and variance identities in Theorem

2.56, that

$$\begin{aligned}\mu = \mathrm{E}(U) = \mathrm{E}\left(\sum_{i=1}^{100} X_i\right) &= \mathrm{E}(X_1 + X_2 + \cdots + X_{100}) \\ &= \mathrm{E}(X_1) + \mathrm{E}(X_2) + \cdots + \mathrm{E}(X_{100}) \\ &= \sum_{i=1}^{100} E(X_i) = \sum_{i=1}^{100} 54 = 100 \cdot 54 = 5400,\end{aligned}$$

and

$$\begin{aligned}\sigma^2 = \mathrm{V}(U) = \mathrm{V}\left(\sum_{i=1}^{100} X_i\right) &= \mathrm{V}(X_1 + X_2 + \cdots + X_{100}) \\ &= \mathrm{V}(X_1) + \mathrm{V}(X_2) + \cdots + \mathrm{V}(X_{100}) \\ &= \sum_{i=1}^{100} \mathrm{V}(X_i) = \sum_{i=1}^{100} 9 = 100 \cdot 9.\end{aligned}$$

So the answer is: $\mu = 100 \cdot 54 = 5400$ ms and $\sigma = 3 \cdot \sqrt{100} = 30$ ms.

2.12 Concrete items

|||| Exercise 2.12 Concrete items

A manufacturer of concrete items knows that the length (L) of his items are reasonably normally distributed with $\mu_L = 3000$ mm and $\sigma_L = 3$ mm. The requirement for these elements is that the length should be not more than 3007 mm and the length must be at least 2993 mm.

a) The expected error rate in the manufacturing will be?

|||| Solution

The expected error rate is:

$$\begin{aligned} P(L \leq 2993) + P(L \geq 3007) &= P(L \leq 2993) + 1 - P(L \leq 3007) \\ &= 2 \cdot P(L \leq 2993) = 0.01963066. \end{aligned}$$

In Python:

```
print(stats.norm.cdf(2993,loc=3000,scale=3) + (1-stats.norm.cdf(3007,loc=3000,scale=3))


0.01963065725729065


# Or
print(2 * (stats.norm.cdf(2993,loc=3000,scale=3)))


0.019630657257290667
```

So the answer becomes: approximately 2%.

b) The concrete items are supported by beams, where the distance between the beams is called L_{beam} and can be assumed normal distributed. The concrete items length is still called L. For the items to be supported correctly, the following requirements for these lengths must be fulfilled: 90 mm $<$

$L - L_{\text{beam}} < 110$ mm. It is assumed that the mean of the distance between the beams is $\mu_{\text{beam}} = 2900$ mm. How large may the standard deviation σ_{beam} of the distance between the beams be if you want the requirement fulfilled in 99% of the cases?

|||| Solution

The following should be fulfilled

$$P(90 < L - L_{\text{beam}} < 110) = 0.99.$$

We know that $\mathrm{E}(L - L_{\text{beam}}) = 3000 - 2900 = 100$ and that

$$\mathrm{V}(L - L_{\text{beam}}) = 9 + \sigma^2_{\text{beam}}.$$

So transforming to the standard normal gives

$$0.99 = P(90 < L - L_{\text{beam}} < 110) = P\left(\frac{-10}{\sqrt{9 + \sigma^2_{\text{beam}}}} < Z < \frac{10}{\sqrt{9 + \sigma^2_{\text{beam}}}}\right).$$

So since for the standard normal, we can find that

$$0.99 = P(-z_{0.005} < Z < z_{0.005}),$$

where $z_{0.005} = 2.576$ (in Python: `stats.norm.ppf(0.995)`), we can solve

$$2.576 = \frac{10}{\sqrt{9 + \sigma^2_{\text{beam}}}},$$

for σ_{beam}

$$\sigma_{\text{beam}} = \sqrt{\left(\frac{10}{2.576}\right)^2 - 9} = 2.464.$$

So the answer becomes: $\sigma_{\text{beam}} = 2.46$ mm.

2.13 Online statistic video views

|||| Exercise 2.13 Online statistic video views

In 2013, there were 110,000 views of the DTU statistics videos that are available online. Assume first that the occurrence of views through 2014 follows a Poisson process with a 2013 average: $\lambda_{365days} = 110000$.

a) What is the probability that in a randomly chosen half an hour there is no occurrence of views?

|||| Solution

The half hour intensity is

$$\lambda_{30min} = \lambda_{365days}/(365 \cdot 48) = \frac{110000}{17520} = 6.28.$$

So if X is the number of views in half an hour then, $X \sim Po(6.28)$ and the wanted probability is

$$P(X = 0) = \exp(-6.28) = 0.00187.$$

Or in Python:

```
lambda30min = 110000/(365*24*2)
print(stats.poisson.cdf(k=0,mu=lambda30min))


0.0018761399841118903
```

So the correct answer is: 0.002.

b) There has just been a view, what is the probability that you have to wait more than fifteen minutes for the next view?

|||| Solution

This can be solved either using the Poisson distribution: the 15 minutes rate is

$$\lambda_{15min} = \lambda_{365days}/(365 \cdot 96) = \frac{110000}{2 \cdot 17520} = 3.14.$$

So if X is the number of views in 15 minutes then, $X \sim Po(3.14)$ and the wanted probability is found in Python:

```
lambda30min = 110000/(365*24*2)
lambda15min = lambda30min/2
print(stats.poisson.cdf(k=0,mu=lambda15min))


0.04331443159169806
```

$$P(X = 0) = \exp(-3.14) = 0.043.$$

Or using the exponential distribution: the mean waiting time for a view is (in minutes)

$$\beta = 365 \cdot 24 \cdot 60/110000 = 4.778.$$

which in Python:

```
beta = 365*24*60/110000
print(1-stats.expon.cdf(x=15,scale=beta))


0.04331443159169801
```

So, the correct answer is: 0.043.

2.14 Body mass index distribution

|||| Exercise 2.14 Body mass index distribution

The so-called BMI (Body Mass Index) is a measure of the weight-height-relation, and is defined as the weight (W) in kg divided by the squared height (H) in meters:

$$BMI = \frac{W}{H^2}.$$

Assume that the population distribution of BMI is a log-normal distribution with $\alpha = 3.1$ and $\beta = 0.15$ (hence that $\log(BMI)$ is normal distributed with mean 3.1 and standard deviation 0.15).

a) A definition of "being obese" is a BMI-value of at least 30. How large a proportion of the population would then be obese?

|||| Solution

$$P(BMI > 30) = P(\log(BMI) > \log(30)) = P(Z > \frac{\log(30) - 3.1}{0.15}) = P(Z > 2.008) = 0.0223$$

where Z is a standard normal variable $Z \sim N(0,1)$. Or in Python:

```
al = 3.1
be = 0.15
print(1-stats.norm.cdf((np.log(30)-al)/be))
```

```
0.02232257330614873
```

```
# Or using log normal directly
print(1-stats.lognorm.cdf(30,s=be,scale=np.exp(al)))
```

```
0.02232257330614884
```

So the correct answer is: 2.23%.

2.15 Bivariate normal

|||| Exercise 2.15 Bivariate normal

a) In the bivariate normal distribution (see Example 2.73), show that if Σ is a diagonal matrix then (X_1, X_2) are also independent and follow univariate normal distributions.

|||| Solution

Σ is diagonal so

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix},$$

hence $|\Sigma| = \sigma_1^2\sigma_2^2$ and

$$\Sigma^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix},$$

the joint density is therefore

$$\begin{aligned} f(x_1, x_2) &= \frac{1}{2\pi\sqrt{\sigma_1^2\sigma_2^2}} e^{-\frac{1}{2}\frac{(x_1-\mu_1)^2}{\sigma_1^2} - \frac{1}{2}\frac{(x_2-\mu_2)^2}{\sigma_2^2}} \\ &= \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{1}{2}\frac{(x_1-\mu_1)^2}{\sigma_1^2}} \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{1}{2}\frac{(x_2-\mu_2)^2}{\sigma_2^2}}, \end{aligned}$$

which can be recognized as the product between two univariate normal pdf's (with $X_1 \sim N(\mu_1, \sigma_1^2)$ and $X_2 \sim N(\mu_2, \sigma_2^2)$). As the density is the product of two univariate densities they are independent, see Theorem 2.75.

b) Assume that Z_1 and Z_2 are independent standard normal random variables. Now let X and Y be defined by

$$\begin{aligned} X &= a_{11}Z_1 + c_1, \\ Y &= a_{12}Z_1 + a_{22}Z_2 + c_2. \end{aligned}$$

Show that an appropriate choice of $a_{11}, a_{12}, a_{22}, c_1, c_2$ can give any bivariate normal distribution for the random vector (X, Y), i.e. find $a_{11}, a_{12}, a_{22}, c_1, c_2$ as a function of μ_X, μ_Y and the elements of Σ.

Note that $\Sigma_{ij} = \text{Cov}(X_i, X_j)$ (i.e. here $\Sigma_{12} = \Sigma_{21} = \text{Cov}(X, Y)$), and that any linear combination of random normal variables will result in a random normal variable.

|||| Solution

$$\begin{aligned}
\text{E}(X) &= c_1 \Rightarrow c_1 = \mu_X, \\
\text{E}(Y) &= c_2 \Rightarrow c_2 = \mu_Y, \\
\text{V}(X) &= a_{11}^2 \Rightarrow a_{11} = \sqrt{\Sigma_{11}}, \\
\text{Cov}(X, Y) &= a_{11} a_{12} \\
&= \sqrt{\Sigma_{11}} a_{12} \Rightarrow a_{12} = \frac{\Sigma_{12}}{\sqrt{\Sigma_{11}}}, \\
\text{V}(Y) &= a_{12}^2 + a_{22}^2 \\
&= \frac{\Sigma_{12}^2}{\Sigma_{11}} + a_{22}^2 \Rightarrow a_{22} = \sqrt{\Sigma_{22} - \frac{\Sigma_{12}^2}{\Sigma_{11}}}.
\end{aligned}$$

c) Use the result to simulate 1000 realization of a bivariate normal random variable with $\boldsymbol{\mu} = (1, 2)$ and

$$\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

and make a scatter plot of the bivariate random variable.

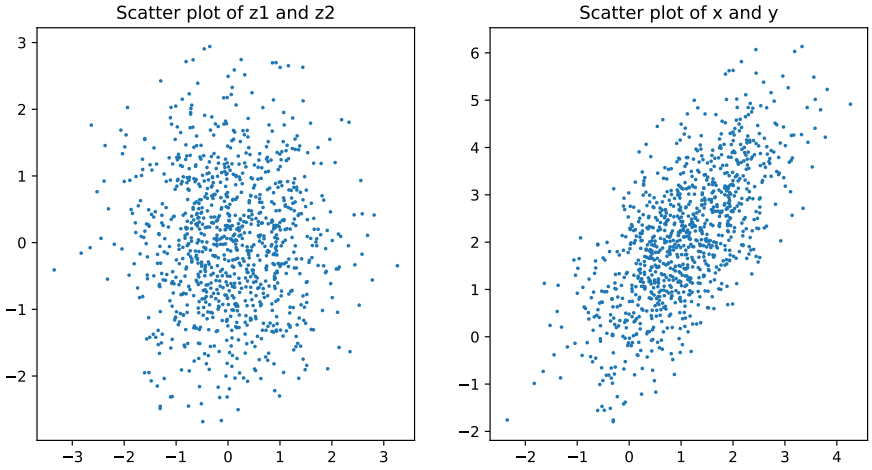
|||| Solution

In Python:

```
# Set the seed
np.random.seed(125)
# The parameters
Sigma = np.array([[1, 1], [1, 2]])
mu = np.array([1, 2])
c1, c2 = mu
a11 = np.sqrt(Sigma[0, 0])
a12 = Sigma[0, 1] / np.sqrt(Sigma[0, 0])
a22 = np.sqrt(Sigma[1, 1] - Sigma[0, 1]**2 / Sigma[0, 0])
# Simulate
k = 1000
z1 = np.random.normal(size=k)
z2 = np.random.normal(size=k)
# The simulation of X and Y
x = a11 * z1 + c1
y = a12 * z1 + a22 * z2 + c2
# The sample covariance
sample_cov = np.cov(np.vstack((z1, z2)))
print(sample_cov)


[[ 0.934 -0.022]
 [-0.022  1.073]]


# Make the scatter plots
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.scatter(z1, z2, s=2)
plt.title('Scatter plot of z1 and z2')
plt.subplot(1, 2, 2)
plt.scatter(x, y, s=2)
plt.title('Scatter plot of x and y')
plt.show()
```


Scatter plot of z1 and z2
Scatter plot of x and y

2.16 Sample distributions

|||| Exercise 2.16 Sample distributions

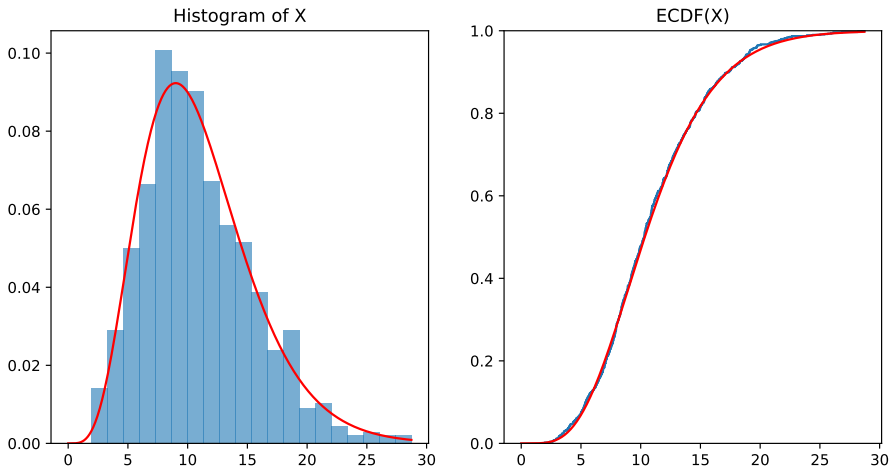
a) Verify by simulation that $\frac{n_1+n_2-2}{\sigma^2}S_p^2 \sim \chi^2(n_1+n_2-2)$ (See Example 2.85). You may use $n_1 = 5$, $n_2 = 8$, $\mu_1 = 2$, $\mu_2 = 4$, and $\sigma^2 = 2$.

|||| Solution

In Python:

```
# Set the seed for reproducibility
np.random.seed(125)
# Set parameters
k = 1000
n1, n2 = 5, 8
mu1, mu2 = 2, 4
sigma = np.sqrt(2)
# Simulate sample variances
s1 = np.var(np.random.normal(mu1, sigma, (n1,k)), axis=0,ddof=1)
s2 = np.var(np.random.normal(mu2, sigma, (n2,k)), axis=0,ddof=1)
# Compute pooled sample variance
sp = ((n1 - 1) * s1 + (n2 - 1) * s2) / (n1 + n2 - 2)
# Calculate x
x = sp * (n1 + n2 - 2) / sigma**2

# Plot histogram and chi-squared distribution
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.hist(x, bins=20, density=True, alpha=0.6)
xseq = np.linspace(0, np.max(x), 1000)
plt.plot(xseq, stats.chi2.pdf(xseq, df=n1+n2-2), 'r-')
plt.title('Histogram of X')
# Plot ECDF and chi-squared CDF
plt.subplot(1, 2, 2)
plt.ecdf(x)
plt.plot(xseq, stats.chi2.cdf(xseq, df=n1+n2-2), 'r-')
plt.title('ECDF(X)')
plt.show()
```



b) Show that if $X \sim N(\mu_1, \sigma^2)$ and $Y \sim N(\mu_2, \sigma^2)$, then

$$\frac{\bar{X} - \bar{Y} - (\mu_1 - \mu_2)}{S_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t(n_1 + n_2 - 2).$$

Verify the result by simulation. You may use $n_1 = 5$, $n_2 = 8$, $\mu_1 = 2$, $\mu_2 = 4$, and $\sigma^2 = 2$.

|||| Solution

First we consider Theorem 2.87, which makes us think how to find an expression which is standard normal distributed and include the right variables. Since,

$$\begin{aligned} \mathrm{E}(\bar{X} - \bar{Y}) &= \mu_1 - \mu_2, \\ \mathrm{V}(\bar{X} - \bar{Y}) &= \mathrm{V}(\bar{X}) + \mathrm{V}(\bar{Y}) = \sigma^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right), \end{aligned}$$

then we can standardize $\bar{X} - \bar{Y}$ by

$$\frac{\bar{X} - \bar{Y} - (\mu_1 - \mu_2)}{\sqrt{\sigma^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \sim N(0, 1).$$

Then we look for something which is χ^2-distributed and find

$$\frac{n_1 + n_2 - 2}{\sigma^2} S_p^2 \sim \chi^2(n_1 + n_2 - 2).$$

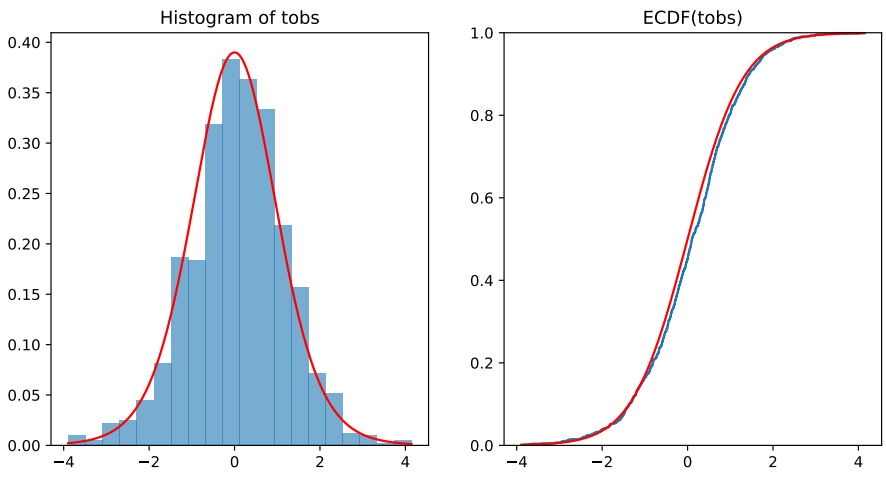
These two expressions with a little more enables us using Theorem 2.87 to setup

$$\frac{\frac{\bar{X}-\bar{Y}-(\mu_1-\mu_2)}{\sqrt{\sigma^2\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}}}{\sqrt{\frac{n_1+n_2-2}{\sigma^2}S_p^2\cdot\frac{1}{n_1+n_2-2}}} = \frac{\bar{X}-\bar{Y}-(\mu_1-\mu_2)}{S_p\sqrt{\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}} \sim t(n_1+n_2-2)$$

Verify by simulating the left side and comparing this with the right side:

```
# Set the seed for reproducibility
np.random.seed(125)
# Set parameters
n1 = 5
n2 = 8
mu1 = 2
mu2 = 4
sigma = np.sqrt(2)
# Simulate
k = 1000
X = np.random.normal(mu1, sigma, (n1, k))
Y = np.random.normal(mu2, sigma, (n2, k))
s1 = np.var(X, axis=0, ddof=1)
s2 = np.var(Y, axis=0, ddof=1)
m1 = np.mean(X, axis=0)
m2 = np.mean(Y, axis=0)
sp = ((n1 - 1) * s1 + (n2 - 1) * s2) / (n1 + n2 - 2)
tobs = (m1 - m2 - (mu1 - mu2)) / np.sqrt(sp * (1/n1 + 1/n2))

# Plot Histogram
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.hist(tobs,bins=20, density=True, alpha=0.6)
xseq = np.linspace(np.min(tobs), np.max(tobs), 1000)
plt.plot(xseq, stats.t.pdf(xseq, df=n1+n2-2), 'r-')
plt.title('Histogram of tobs')
# Plot ECDF
plt.subplot(1, 2, 2)
plt.ecdf(tobs)
plt.plot(xseq, stats.t.cdf(xseq, df=n1+n2-2), 'r-')
plt.title('ECDF(tobs)')
plt.show()
```


Histogram of tobs
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−2
0
2
4
ECDF(tobs)
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4

2.17 Sample distributions 2

|||| Exercise 2.17 Sample distributions 2

Let $X_1, ..., X_n$ and $Y_1, ..., Y_n$, with $X_i \sim N(\mu_1, \sigma^2)$ and $Y_i \sim N(\mu_2, \sigma^2)$ be independent random variables. Hence, two samples before they are taken. S_1^2 and S_2^2 are the sample variances based on the X's and the Y's respectively. Now define a new random variable

$$Q = \frac{S_1^2}{S_2^2} \tag{2-1}$$

a) For n equal $2, 4, 8, 16$ and 32 find:

1. $P(Q < 1)$
2. $P(Q > 2)$
3. $P\left(Q < \frac{1}{2}\right)$
4. $P\left(\frac{1}{2} < Q < 2\right)$

|||| Solution

From Theorem 2.96 we know that Q follows an F-distribution with degrees of freedom $\nu_1 = \nu_2 = n - 1$, and we find the required probabilities with Python:

```
# Set n as a list to get the results for all the n
n = np.array([2, 4, 8, 16, 32])
# P(Q<1)
print(stats.f.cdf(1, n-1,n-1))


[0.500 0.500 0.500 0.500 0.500]


# P(Q>2)
print(1 - stats.f.cdf(2,n-1,n-1))


[0.392 0.292 0.190 0.096 0.029]


# P(Q<0.5)
print(stats.f.cdf(0.5,n-1,n-1))


[0.392 0.292 0.190 0.096 0.029]


# P(0.5<Q<2)
print(stats.f.cdf(2,n-1,n-1) - stats.f.cdf(0.5,n-1,n-1))


[0.216 0.416 0.619 0.809 0.942]
```

b) For at least one value of n illustrate the results above by direct simulation from independent normal distributions. You may use any values of μ_1, μ_2 and σ^2.

|||| Solution

In Python:

```
## Set the seed for reproducibility
np.random.seed(125)
## Set parameters
mu1,mu2 = 2,1
sigma = 2
## Simulate
n,k = 8,100000 # k is the number of replications for the simulation
S1sq = np.var(np.random.normal(mu1,sigma,(n,k)),axis=0,ddof=1)
S2sq = np.var(np.random.normal(mu2,sigma,(n,k)),axis=0,ddof=1)
Q = S1sq/S2sq # We know that Q will follow a F-distribution
```

```
## P(Q<1)
# Theoretical value
print(stats.f.cdf(1,n-1,n-1))


0.5000000000000002


# Empirical value (simulation)
print(np.sum(Q<1)/k)


0.50291


## P(Q>2)
# Theoretical value
print(1-stats.f.cdf(2,n-1,n-1))


0.19035659083843182


# Empirical value (simulation)
print(np.sum(Q>2)/k)


0.18744


## P(0.5< Q < 2)
# Theoretical value
print(stats.f.cdf(2,n-1,n-1) - stats.f.cdf(0.5,n-1,n-1))


0.6192868183231363


print(np.sum((Q>0.5) & (Q<2))/k)


0.62077
```