Deep Learning KU (708.220) WS22

Assignment 1: Maximum Likelihood Estimation

David Mihola
12211951

November 9th, 2022

Contents

First part – derivation

The assignment specifies a 3-dimensional Gaussian distribution with a mean vector

$$\boldsymbol{\mu} \in \mathbb{R}^3$$

and a covariance matrix

$$\Sigma \in \mathbb{R}^{3\times 3}\,,$$

for which there is additional information of

$$\Sigma = \sigma^2 I = \sigma^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}.$$

The determinant of the covariance matrix can be calculated using the Rule of Sarrus as

$$\det(\Sigma) = \det \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} = \sigma^2 \cdot \sigma^2 \cdot \sigma^2 + 0 \cdot 0 \cdot 0 + 0 \cdot 0 \cdot 0 - 0 \cdot \sigma^2 \cdot 0 - 0 \cdot 0 \cdot \sigma - \sigma \cdot 0 \cdot 0 = \sigma^6$$

and the inverse covariance matrix Σ^{-1} can be calculated with the knowledge that

$$\Sigma \times \Sigma^{-1} = \Sigma^{-1} \times \Sigma = I$$

and that

$$\Sigma = \sigma^2 I\,,$$

therefore the equation can be written as

$$\sigma^2 I \times \Sigma^{-1} = I\,.$$

Multiplying a matrix by the identity matrix is an identity operation, so the equation can be simplified to

$$\sigma^2 \Sigma^{-1} = I$$

and by multiplying each side of the equation by $\frac{1}{\sigma^2}$, the result is

$$\Sigma^{-1} = \frac{1}{\sigma^2} I = \begin{bmatrix} \frac{1}{\sigma^2} & 0 & 0 \\ 0 & \frac{1}{\sigma^2} & 0 \\ 0 & 0 & \frac{1}{\sigma^2} \end{bmatrix}.$$

Furthermore, the assignment specifies, that there is a collection of random variables, which is independently and identically distributed (i.i.d).

a) Likelihood of a single data point and the whole data set

The likelihood for a single data point $\boldsymbol{x}^m$, where $\boldsymbol{\theta} = \langle \boldsymbol{\mu}, \sigma \rangle$ and the covariance matrix and inverse covariance matrix are given as described above is

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}^m) = p(\boldsymbol{x}^m|\boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^3|\Sigma|}} e^{-\frac{1}{2}(\boldsymbol{x}^m-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}^m-\boldsymbol{\mu})} = \frac{1}{\sqrt{(2\pi)^3\sigma^6}} e^{-\frac{1}{2}s},$$

where s is the following substitution

$$s = (\boldsymbol{x}^m - \boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}^m - \boldsymbol{\mu}) = \begin{bmatrix} x_1^m - \mu_1 & x_2^m - \mu_2 & x_3^m - \mu_3 \end{bmatrix} \times \begin{bmatrix} \frac{1}{\sigma^2} & 0 & 0 \\ 0 & \frac{1}{\sigma^2} & 0 \\ 0 & 0 & \frac{1}{\sigma^2} \end{bmatrix} \times \begin{bmatrix} x_1^m - \mu_1 \\ x_2^m - \mu_2 \\ x_3^m - \mu_3 \end{bmatrix} =$$

$$\frac{1}{\sigma^2} \begin{bmatrix} x_1^m - \mu_1 & x_2^m - \mu_2 & x_3^m - \mu_3 \end{bmatrix} \times \begin{bmatrix} x_1^m - \mu_1 \\ x_2^m - \mu_2 \\ x_3^m - \mu_3 \end{bmatrix} =$$

$$\frac{1}{\sigma^2}\left((x_1^m - \mu_1)\cdot(x_1^m - \mu_1) + (x_2^m - \mu_2)\cdot(x_2^m - \mu_2) + (x_3^m - \mu_3)\cdot(x_3^m - \mu_3)\right) =$$

$$\frac{1}{\sigma^2}\left((x_1^m - \mu_1)^2 + (x_2^m - \mu_2)^2 + (x_3^m - \mu_3)^2\right).$$

Finally, the likelihood for a single data point $\boldsymbol{x}^m$ can be written as

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}^m) = \frac{1}{\sqrt{(2\pi)^3\sigma^6}} e^{-\frac{1}{2\sigma^2}\left((x_1^m-\mu_1)^2+(x_2^m-\mu_2)^2+(x_3^m-\mu_3)^2\right)}.$$

Using the equation derived above for a single data point, it can be generalized for the whole data set. Knowing that the samples of the data set are independent and identically distributed the likelihood of the whole data set is

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}) = p(\boldsymbol{X}|\boldsymbol{\theta}) = p(\boldsymbol{x^1}, \boldsymbol{x^2}, ..., \boldsymbol{x^N}|\boldsymbol{\theta}) \overset{\text{i.i.d}}{=} \prod_{n=1}^{N} \frac{1}{\sqrt{(2\pi)^3\sigma^6}} e^{-\frac{1}{2\sigma^2}\left((x_1^n-\mu_1)^2+(x_2^n-\mu_2)^2+(x_3^n-\mu_3)^2\right)}.$$

Deriving the log-likelihood of the data set means just applying the natural logarithm on the likelihood

$$\ell(\boldsymbol{\theta}|\boldsymbol{X}) = \ln\left(\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X})\right) \overset{\text{i.i.d}}{=} \ln\left(\prod_{n=1}^{N} \frac{1}{\sqrt{(2\pi)^3\sigma^6}} e^{-\frac{1}{2\sigma^2}\left((x_1^n-\mu_1)^2+(x_2^n-\mu_2)^2+(x_3^n-\mu_3)^2\right)}\right) =$$

$$\sum_{n=1}^{N}\left(\ln\left(\frac{1}{\sqrt{(2\pi)^3\sigma^6}}\right) + -\frac{1}{2\sigma^2}\left((x_1^n - \mu_1)^2 + (x_2^n - \mu_2)^2 + (x_3^n - \mu_3)^2\right)\right).$$

Additionally, additive and multiplicative constants and variables, that remain constant in regards to the summation, can be moved in front of the sum operator followingly

$$\ell(\boldsymbol{\theta}|\boldsymbol{X}) \overset{\text{i.i.d}}{=} N \cdot \ln\left(\frac{1}{\sqrt{(2\pi)^3\sigma^6}}\right) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left((x_1^n - \mu_1)^2 + (x_2^n - \mu_2)^2 + (x_3^n - \mu_3)^2\right).$$

b) Maximum-likelihood estimate of $\boldsymbol{\mu}$

The natural logarithm is a strictly increasing function, therefore the log-likelihood function can be used to estimate the $\boldsymbol{\mu}$ parameter, as the value of $\boldsymbol{\mu}$, for which the likelihood function has its maximum will not change

$$\hat{\boldsymbol{\mu}}_{ML} = \arg\max_{\boldsymbol{\mu}} \left(\ell(\boldsymbol{\theta}|\boldsymbol{X})\right) =$$

$$\arg\max_{\boldsymbol{\mu}} \left(N \cdot \ln\left(\frac{1}{\sqrt{(2\pi)^3\sigma^6}}\right) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} \left((x_1^n - \mu_1)^2 + (x_2^n - \mu_2)^2 + (x_3^n - \mu_3)^2 \right) \right) .$$

Furthermore, the 3-dimensional Gaussian distribution is a concave function. Therefore, the maximum-likelihood estimate of $\boldsymbol{\mu}$ can be obtained by solving

$$\frac{\partial \ell(\boldsymbol{\theta}|\boldsymbol{X})}{\partial \boldsymbol{\mu}} = 0 .$$

The equation above equates the partial derivative by the 3-dimensional vector $\boldsymbol{\mu}$ to zero. The same result can be also achieved by equating the derivatives of each component of $\boldsymbol{\mu}$ to zero and by solving these equations separately as

$$\begin{aligned} \tfrac{d\ell(\boldsymbol{\theta}|\boldsymbol{X})}{d\mu_1} &= 0 \\ \tfrac{d\ell(\boldsymbol{\theta}|\boldsymbol{X})}{d\mu_2} &= 0 \\ \tfrac{d\ell(\boldsymbol{\theta}|\boldsymbol{X})}{d\mu_3} &= 0 \end{aligned} ,$$

by applying the formula derived above

$$\begin{aligned} \frac{d\left(N\cdot\ln\left(\frac{1}{\sqrt{(2\pi)^3\sigma^6}}\right) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left((x_1^n-\mu_1)^2+(x_2^n-\mu_2)^2+(x_3^n-\mu_3)^2\right)\right)}{d\mu_1} &= 0 \\ \frac{d\left(N\cdot\ln\left(\frac{1}{\sqrt{(2\pi)^3\sigma^6}}\right) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left((x_1^n-\mu_1)^2+(x_2^n-\mu_2)^2+(x_3^n-\mu_3)^2\right)\right)}{d\mu_2} &= 0 \\ \frac{d\left(N\cdot\ln\left(\frac{1}{\sqrt{(2\pi)^3\sigma^6}}\right) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left((x_1^n-\mu_1)^2+(x_2^n-\mu_2)^2+(x_3^n-\mu_3)^2\right)\right)}{d\mu_3} &= 0 \end{aligned} ,$$

by calculating the derivatives

$$\begin{aligned} -\tfrac{1}{2\sigma^2}\textstyle\sum_{n=1}^{N} 2\left(x_1^n - \mu_1\right) &= 0 \\ -\tfrac{1}{2\sigma^2}\textstyle\sum_{n=1}^{N} 2\left(x_2^n - \mu_2\right) &= 0 \\ -\tfrac{1}{2\sigma^2}\textstyle\sum_{n=1}^{N} 2\left(x_3^n - \mu_3\right) &= 0 \end{aligned} ,$$

by cancellation of the constant value of 2 and by moving the constants and constant variables in regards to the summation in front of the sum

$$\begin{aligned} \tfrac{N\mu_1}{\sigma^2} - \tfrac{1}{\sigma^2}\textstyle\sum_{n=1}^{N} x_1^n &= 0 \\ \tfrac{N\mu_2}{\sigma^2} - \tfrac{1}{\sigma^2}\textstyle\sum_{n=1}^{N} x_2^n &= 0 \\ \tfrac{N\mu_3}{\sigma^2} - \tfrac{1}{\sigma^2}\textstyle\sum_{n=1}^{N} x_3^n &= 0 \end{aligned} ,$$

by multiplying the equations by $\frac{1}{\sigma^2}$

$$
\begin{aligned}
N \cdot \mu_1 - \sum_{n=1}^{N} x_1^n = 0 \\
N \cdot \mu_2 - \sum_{n=1}^{N} x_2^n = 0 \\
N \cdot \mu_3 - \sum_{n=1}^{N} x_3^n = 0
\end{aligned}
$$

and by moving μ_k to one side of the equation

$$
\begin{aligned}
\mu_1 = \frac{1}{N} \sum_{n=1}^{N} x_1^n \\
\mu_2 = \frac{1}{N} \sum_{n=1}^{N} x_2^n \\
\mu_3 = \frac{1}{N} \sum_{n=1}^{N} x_3^n
\end{aligned} .
$$

Furthermore, the final result can be rewritten using the vector notation again as

$$
\hat{\boldsymbol{\mu}}_{ML} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}^n \, .
$$

Second part - practical work

To analyze the data and to choose appropriate variables, I have crated a Jupyter notebook, see below. I have done a simple analysis mainly by plotting several chosen variables, which, I thought, were appropriate and might follow the Normal distribution.

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as scs
```

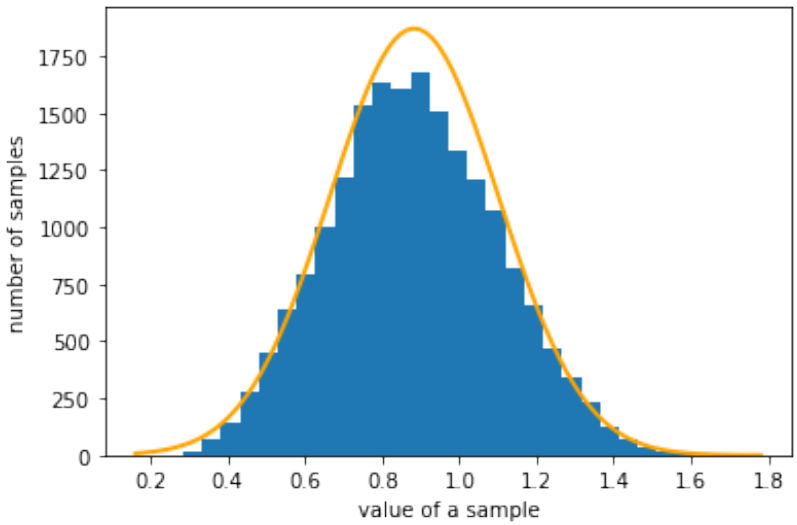
[2]:
```
def stats_for_variable(df, variable_id):
    col = np.array(df[[variable_id]].values).reshape(-1)
    col_cleaned = col[~np.isnan(col)] # remove NaN values
    sample_cnt = col_cleaned.shape[0]

    # MLE estimates
    col_mean_mle = col_cleaned.sum() / sample_cnt
    col_variance_mle = ((col_cleaned - col_mean_mle) ** 2).sum() /␣
 ↪sample_cnt

    print(f"Statistics for a variable \"{variable_id}\":")
    print(f"- number of not NaN samples: {sample_cnt},")
    print(f"- maximum-likelihood estimate for mean: {col_mean_mle:.3f},")
    print(f"- maximum-likelihood estimate for variance:␣
 ↪{col_variance_mle:.3f}.")
    print(f"Histogram overlayed with estimated normal distribution pdf:")
    bins_cnt = 30
    plt.hist(np.hstack(col_cleaned), bins=bins_cnt)

    # histogram overlay with the estimated Normal distribution
    xmin, xmax = plt.xlim()
    bin_width = (xmax - xmin) / bins_cnt
    x = np.linspace(xmin, xmax, 100)
    pdf = scs.norm.pdf(x, col_mean_mle, np.sqrt(col_variance_mle))
    plt.plot(x, pdf * sample_cnt * bin_width, linewidth=2,␣
 ↪color="orange")
    axes = plt.gca()
    axes.set_xlabel("value of a sample")
    axes.set_ylabel("number of samples")
    plt.show()
```

[3]:
```
sc_collage_df = pd.read_csv("data/social_capital_college.csv")
sc_county_df = pd.read_csv("data/social_capital_county.csv")
sc_high_school_df = pd.read_csv("data/social_capital_high_school.csv")
sc_zip_df = pd.read_csv("data/social_capital_zip.csv")
print("Number of samples for each data set (including NaN values):")
print(f"- data set \"social_capital_college.csv\": {sc_collage_df.
 ↪shape[0]},")
print(f"- data set \"social_capital_county.csv\": {sc_county_df.
 ↪shape[0]},")
print(f"- data set \"social_capital_high_school.csv\":␣
 ↪{sc_high_school_df.shape[0]},")
print(f"- data set \"social_capital_zip.csv\": {sc_zip_df.shape[0]}.")
```

```
Number of samples for each data set (including NaN values):
- data set "social_capital_college.csv": 2586,
- data set "social_capital_county.csv": 3089,
- data set "social_capital_high_school.csv": 17525,
- data set "social_capital_zip.csv": 23028.
```

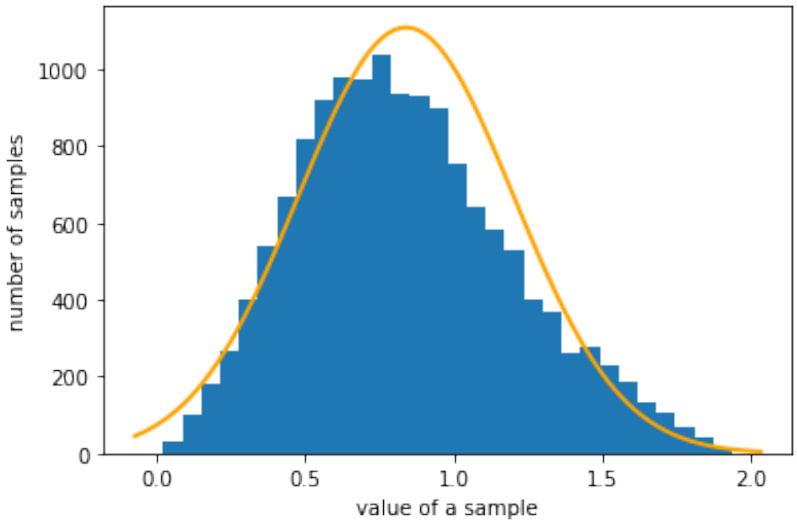
[4]:
```
stats_for_variable(sc_zip_df, "ec_zip")
```

```
Statistics for a variable "ec_zip":
- number of not NaN samples: 18980,
- maximum-likelihood estimate for mean: 0.883,
- maximum-likelihood estimate for variance: 0.048.
Histogram overlayed with estimated normal distribution pdf:
```



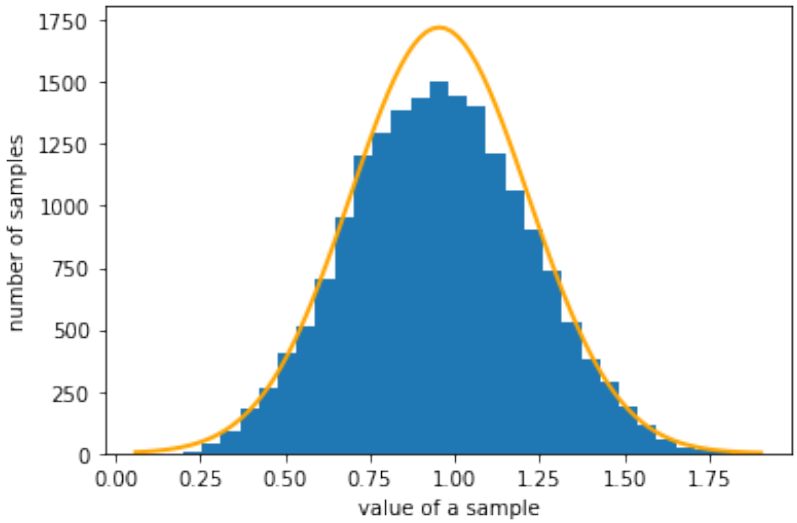
[5]:
```
stats_for_variable(sc_zip_df, "nbhd_ec_zip")
```

```
Statistics for a variable "nbhd_ec_zip":
- number of not NaN samples: 14285,
- maximum-likelihood estimate for mean: 0.840,
- maximum-likelihood estimate for variance: 0.130.
Histogram overlayed with estimated normal distribution pdf:
```



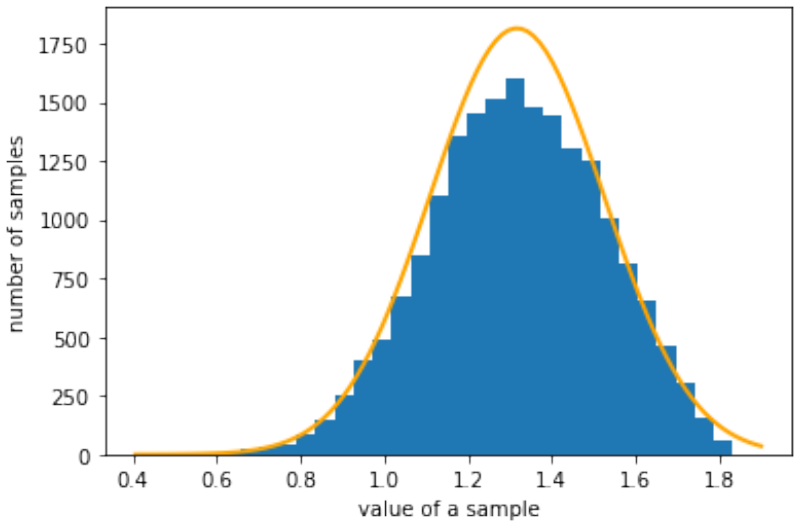
[6]:
```
stats_for_variable(sc_zip_df, "ec_grp_mem_zip")
```

```
Statistics for a variable "ec_grp_mem_zip":
- number of not NaN samples: 18337,
- maximum-likelihood estimate for mean: 0.955,
- maximum-likelihood estimate for variance: 0.068.
Histogram overlayed with estimated normal distribution pdf:
```



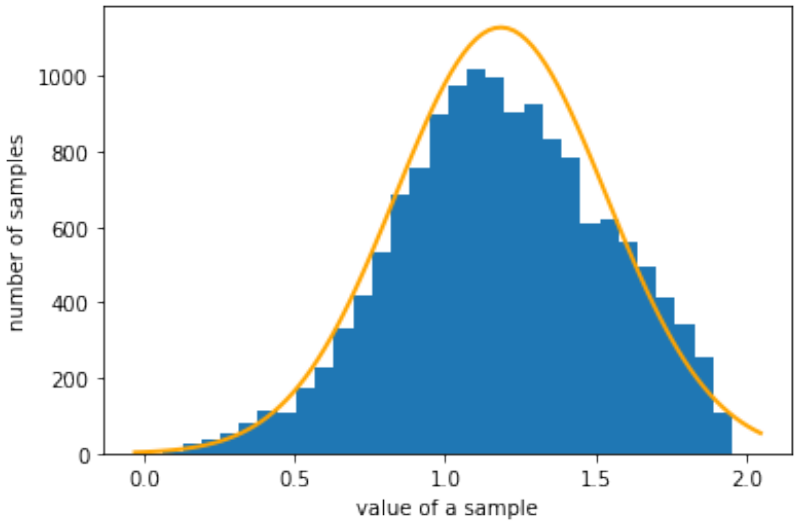
[7]:
```
stats_for_variable(sc_zip_df, "ec_high_zip")
```

```
Statistics for a variable "ec_high_zip":
- number of not NaN samples: 18980,
- maximum-likelihood estimate for mean: 1.317,
- maximum-likelihood estimate for variance: 0.043.
Histogram overlayed with estimated normal distribution pdf:
```



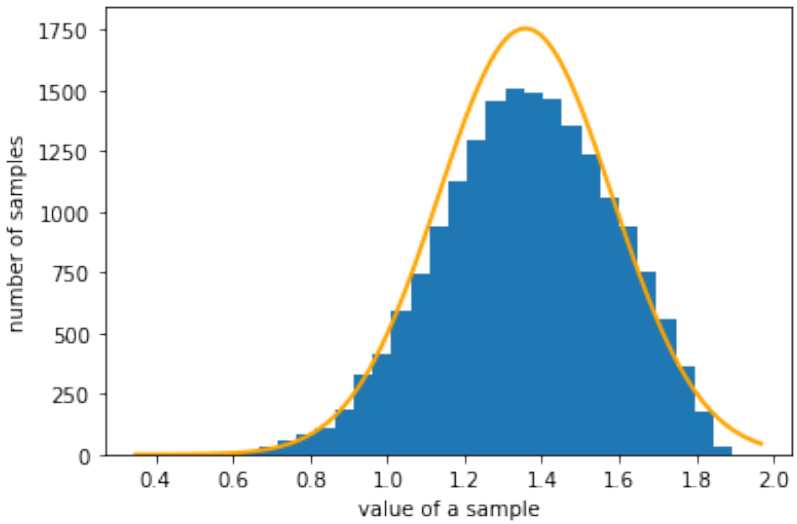
[8]:
```
stats_for_variable(sc_zip_df, "nbhd_ec_high_zip")
```

```
Statistics for a variable "nbhd_ec_high_zip":
- number of not NaN samples: 14285,
- maximum-likelihood estimate for mean: 1.185,
- maximum-likelihood estimate for variance: 0.122.
Histogram overlayed with estimated normal distribution pdf:
```



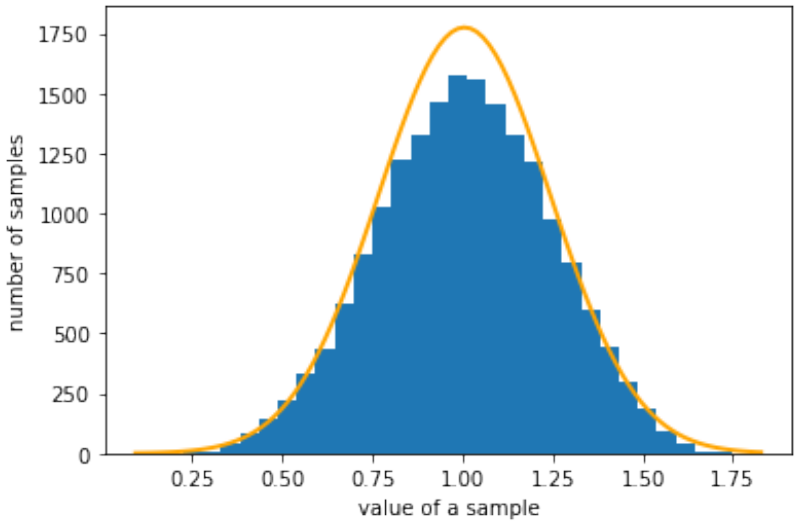
[9]:
```
stats_for_variable(sc_zip_df, "ec_grp_mem_high_zip")
```

```
Statistics for a variable "ec_grp_mem_high_zip":
- number of not NaN samples: 18337,
- maximum-likelihood estimate for mean: 1.357,
- maximum-likelihood estimate for variance: 0.051.
Histogram overlayed with estimated normal distribution pdf:
```



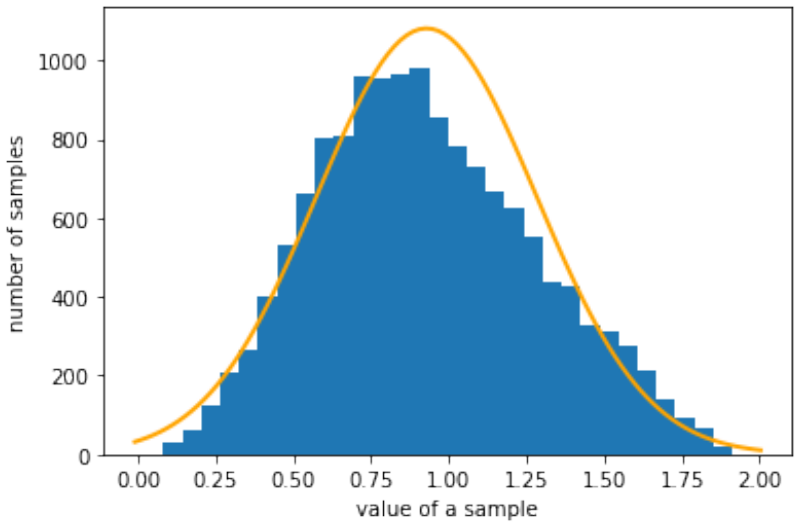
[10]:
```
stats_for_variable(sc_zip_df, "exposure_grp_mem_zip")
```

```
Statistics for a variable "exposure_grp_mem_zip":
- number of not NaN samples: 18337,
- maximum-likelihood estimate for mean: 1.005,
- maximum-likelihood estimate for variance: 0.057.
Histogram overlayed with estimated normal distribution pdf:
```



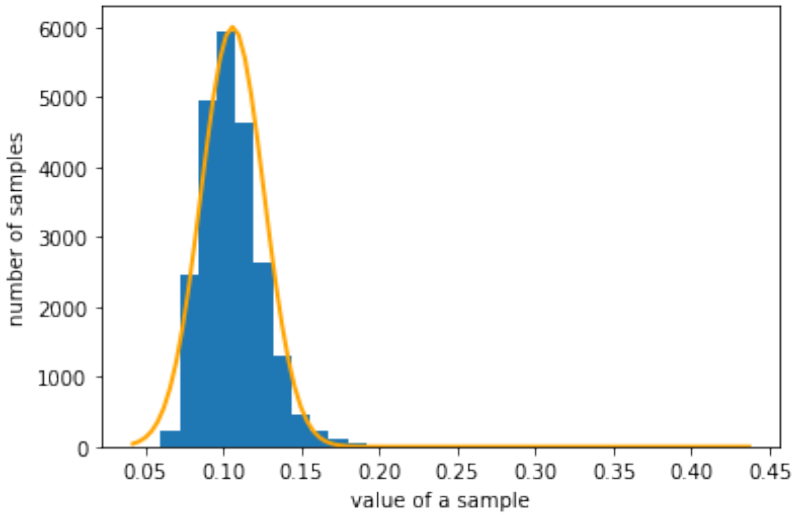
[11]:
```
stats_for_variable(sc_zip_df, "nbhd_exposure_zip")
```

```
Statistics for a variable "nbhd_exposure_zip":
- number of not NaN samples: 14285,
- maximum-likelihood estimate for mean: 0.929,
- maximum-likelihood estimate for variance: 0.125.
Histogram overlayed with estimated normal distribution pdf:
```



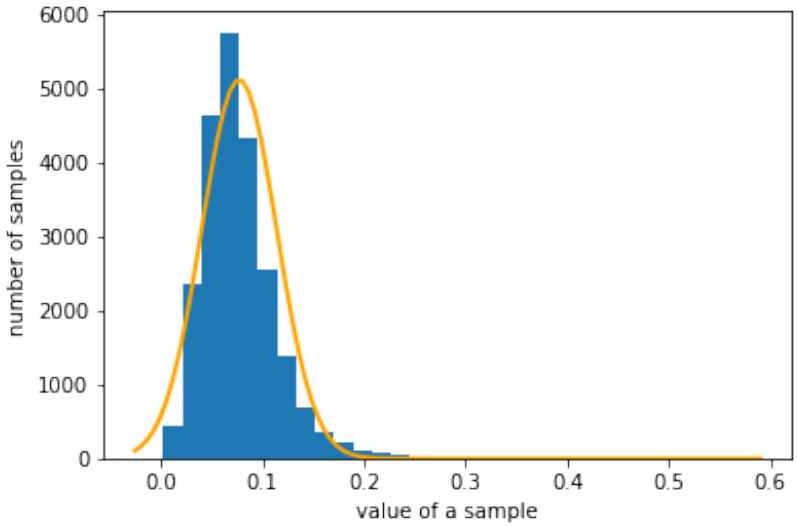
[12]:
```
stats_for_variable(sc_zip_df, "clustering_zip")
```

```
Statistics for a variable "clustering_zip":
- number of not NaN samples: 23028,
- maximum-likelihood estimate for mean: 0.106,
- maximum-likelihood estimate for variance: 0.000.
Histogram overlayed with estimated normal distribution pdf:
```



[13]:
```
stats_for_variable(sc_zip_df, "volunteering_rate_zip")
```

```
Statistics for a variable "volunteering_rate_zip":
- number of not NaN samples: 23025,
- maximum-likelihood estimate for mean: 0.077,
- maximum-likelihood estimate for variance: 0.001.
Histogram overlayed with estimated normal distribution pdf:
```



[14]:
```
selected_cols = sc_zip_df[["ec_zip", "ec_grp_mem_zip",
                           "exposure_grp_mem_zip"]]
print("Covariance matrix of the chosen columns:")
selected_cols.cov()
```

```
Covariance matrix of the chosen columns:
```

[14]:

	ec_zip	ec_grp_mem_zip	exposure_grp_mem_zip
ec_zip	0.047756	0.054757	0.048092
ec_grp_mem_zip	0.054757	0.068441	0.060381
exposure_grp_mem_zip	0.048092	0.060381	0.056931

[15]:
```
print("Correlation between chosen columns:")
selected_cols.corr()
```

```
Correlation between chosen columns:
```

[15]:

	ec_zip	ec_grp_mem_zip	exposure_grp_mem_zip
ec_zip	1.000000	0.966891	0.931110
ec_grp_mem_zip	0.966891	1.000000	0.967315
exposure_grp_mem_zip	0.931110	0.967315	1.000000

c) Selection and explanation of the chosen data

Usually the accuracy of estimation increases with the rise of a number of samples, on which the estimation is performed. Therefore I chose the `social_capital_zip.csv` data set, which has the larges number of rows.

Based on the assignment, the selected random variables should:

1. follow the Normal distribution,
2. all have the same variance,
3. have no correlation.

Considering the task of maximum likelihood estimation of the mean, the first point is the most important. But as can be seen from the plots above, all of these variables follow more or less the Normal distribution. Generally they are missing a bit of mass around the mean, i.e. less values are close to the mean than the estimated Normal distribution suggests, and the missing mass appears at around $\mu - 2\sigma$ and $\mu + 2\sigma$. There are also some variables that are either left or right leaning, i.e. the number of values in the left tail is higher/lower than in the right tail, this can be f.e. seen on the plot of variables `nbhd_ec_zip` and `ec_grp_mem_high_zip`.

The variables, that seems to follow the Normal distribution the best, i.e. the histogram fills the area under the overlaid estimated Normal distribution without overflowing it, are `ec_zip`, `ec_grp_mem_zip` and `exposure_grp_mem_zip`. These variables have also relatively similar variances, ranging from 0.048 to 0.068. The problem might only be with the covariance matrix, which suggest that the variables are correlated. Actually the correlation is very strong, as can be seen above, which is given by the nature of the data set. But considering the derived equation for obtaining the maximum-likelihood of the mean in b), the result will not be hindered by this discrepancy.

d) Maximum-likelihood estimate of μ in Python

See the code submitted in `multivariet_normal_dist_MLE-Mihola.py`. The result is

$$\hat{\boldsymbol{\mu}}_{ML} = \begin{bmatrix} 0.879 & 0.955 & 1.005 \end{bmatrix} .$$

A plot illustrating the position of the estimated mean in 3 dimensional space can be seen on the next page.

150 randomly selected samples (blue) and the estimated mean (orange).