03_regression_numpy_random_numbers

May 7, 2025

[ ]:
```
import os
user = os.getenv('USER')
os.chdir(f'/scratch/cd82/{user}/notebooks')
```

0.0.1 Random Numbers in Python and Statistical Significance

Reference:
"Machine Learning with Python for Everyone", Mark E. Fenner, Addison-Wesley 2020, ISBN-13: 978-0-13-484562-3, pages 21-22

0.0.2 Set up a Random Number Generator

Set the seed value so the output is repeatable

[1]:
```
import numpy as np
import matplotlib.pyplot as plt

seed_seq = np.random.SeedSequence(42)  # The seed for the random number␣
 ↪generator.
# Create a random number generator instance
rng = np.random.default_rng(seed_seq)
print(rng.integers(1, 7, 10))
# The above code is a replacement for often seen functions of the older parts␣
 ↪of the np.random library.
# e.g.
np.random.seed(42)  # this is a a seperate generator to the one above and a␣
 ↪different seed too.
print(np.random.randint(1,7, 10))
```

```
[1 5 4 3 3 6 1 5 2 1]
[4 5 3 5 5 2 3 3 3 5]
```

0.0.3 The Gaussian distribution

Also known as a *normal distribution*, is characterised by the symmetric bell shape. It is important in statistics due to it being found as the distribution of many natural phenomena, and it is the basis of assumptions in many statistical tests.

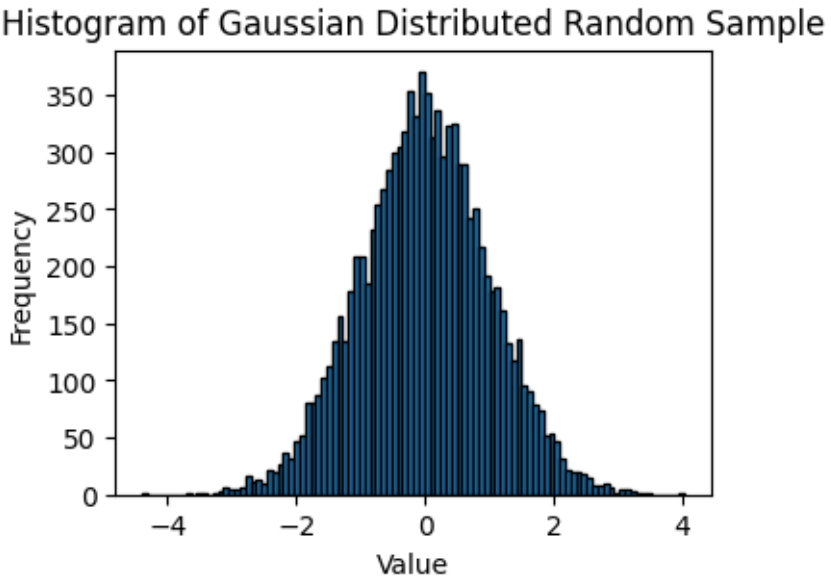
The t-test. We will use 2 sets of samples as an example of use of the `t-test` and resulting `p` value to test if the sets are different.

[2]:
```
mean = 0  # Mean of the distribution
std_dev = 1  # Standard deviation of the distribution
sample_size = 10000  # Number of samples

# Generate a single sample from a normal distribution with mean 0 and standard␣
 ↪deviation 1
gaussian_sample = rng.normal(loc=mean, scale=std_dev, size=sample_size)

gauss_counts = np.histogram(
    gaussian_sample,
    bins=np.arange(.5, 7.5))[0]

plt.figure(figsize=(4, 3))
plt.hist(gaussian_sample, bins=100, edgecolor='black')
plt.title('Histogram of Gaussian Distributed Random Sample')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```



0.0.4 Statistical testing

[3]:
```
import numpy as np
# import scipy.stats
from scipy import stats
```

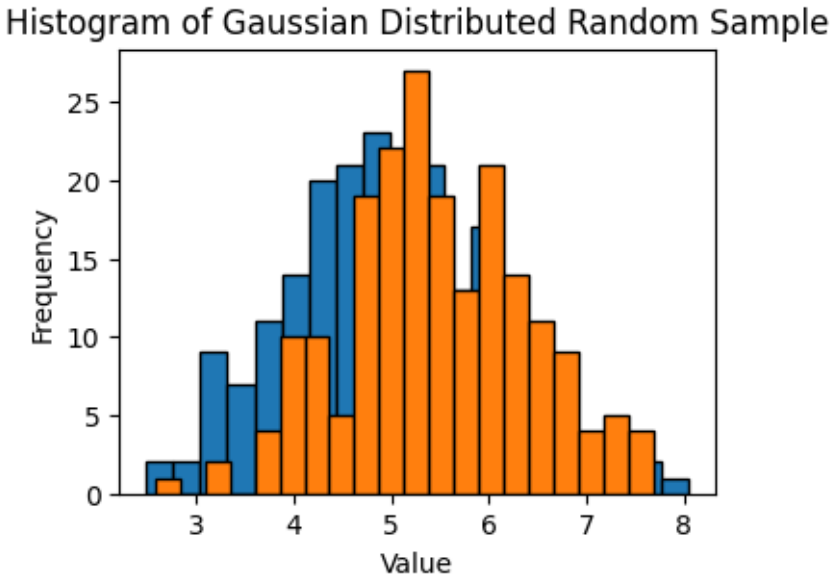
```
from scipy.stats import ttest_ind
from matplotlib import pyplot as plt

# Generate two random samples
s1 = rng.normal(loc=5.0, scale=1, size=200)
s2 = rng.normal(loc=5.5, scale=1, size=200)

print(np.average(s1))
print(np.average(s2))

# plot the two samples
plt.figure(figsize=(4, 3))
plt.hist(s1,  bins=20, edgecolor='black')
plt.hist(s2,  bins=20, edgecolor='black')
plt.title('Histogram of Gaussian Distributed Random Sample')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```

```
4.9175348490963255
5.45972477722924
```



Perform one-sample 1-tailed t-test Answers question: Is one set greater/less than a particular value?

[4]:
```
# Is the mean of s1 greater than 4.0?
t11_stat, p11_value = stats.ttest_1samp(s1, 4.0, alternative='greater')
```

```
print(f"1 sample 1 tailed: T-statistic: {t11_stat}")
print(f"1 sample 1 tailed: P-value: {p11_value}")
# A high t-test value and small p value indicates our question is validated by␣
  ↪the data
# The null-hypothesis can be rejected
```

```
1 sample 1 tailed: T-statistic: 12.56837011576658
1 sample 1 tailed: P-value: 2.3716480789407362e-27
```

Perform the 2 sample 2-tailed t-test Answers the question: Is sample mean of s1 not equal to the sample mean of the second sample s2.

It is the absolute value of the t value that indicates sample difference.
N.B. A negative t-statistic indicates that the sample mean of the first is less than the second

[5]:
```
# Answers question: Are they different?
t22_stat, p22_value = stats.ttest_ind( s1, s2)

print(f"2 sample 2 tailed: T-statistic: {t22_stat}")
print(f"2 sample 2 tailed: P-value: {p22_value}")
```

```
2 sample 2 tailed: T-statistic: -5.463809036137664
2 sample 2 tailed: P-value: 8.232078962658742e-08
```

Two sample 1-tailed test Answers question: Is mean of s2 greater than mean of s1?

[7]:
```
t21_stat, p21_value = stats.ttest_ind(s2, s1, alternative='less')
print(f"2 sample 1 tailed: T-statistic: {t21_stat}")
print(f"2 sample 1 tailed: P-value: {p21_value}")
```

```
2 sample 1 tailed: T-statistic: 5.463809036137664
2 sample 1 tailed: P-value: 0.9999999588396051
```

We would reject the *null hypthesis* that the mean of s2 is less than s1.