Statistical analysis of the “Iris” dataset

Dorota Bjöörn, AI22

Introduction

The “Iris” dataset used in the current report was originally collected by Ronald Fisher in 1936. By measuring width and length of sepals and petals, respectively (*Figure 1)*, Fisher created a mathematical model distinguishing between three types of iris flowers: vercicolor, setosa and virginica (*Figure 2*). His findings were published the same year: Fisher,R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936). [*https://en.wikipedia.org/wiki/Iris_flower_data_set*]

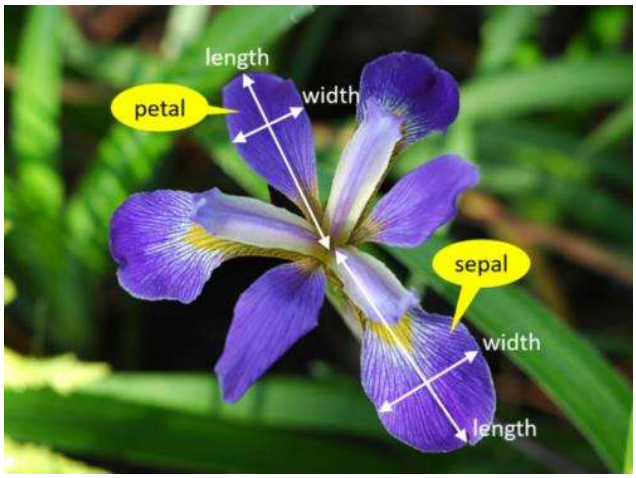


Figure 1. A photograph of an iris flower indicating the four measurements gathered in the iris dataset: petal length, petal width, sepal length and sepal width.[https://www.integratedots.com/determine-number-of-iris-species-with-k-means/]

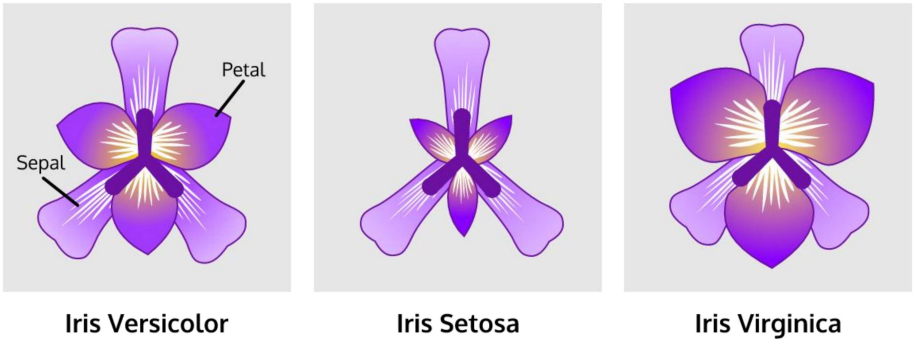


Figure 2. Images of the three iris species described in the iris dataset: versicolor, setosa and virginica.

[https://www.codecademy.com/courses/machine-learning/lessons/machine-learning-clustering/exercises/iris-dataset]

In the current report data is analysed with SciPy and Statsmodels with focus on descriptive analysis, confidence intervals, hypothesis testing, correlation and linear regression.

Methods

Dataset was used as provided [https://www.kaggle.com/datasets/arshid/iris-flower-dataset].

Analysis of the data was performed with NumpPy, Pandas, SciPy, Statsmodels. Graphs were generated with Seaborn and Matplotlib.

Results and discussion

In the following a simple analysis of the dataset is reported. Firstly, the dataset is described and adjusted for further analysis. Secondly the following questions are answered using basic statistical analysis in Python:

Q1 Construct a 95% confidence interval for means for sepal width for each species. Is there a significant difference between sample means for sepal width for iris species?

Q2 Is sepal length for virginica different from sepal length for versicolor?

Q3 Visualize data for pair-wise dependencies.

Q4 Describe dependency between petal length and petal width for setosa and compare with versicolor

All measurements are in cm. Only results are presented and discussed. The underlying code can be found in Appendix.

Dataset

Dataset consisted of 5 series: sepal length, sepal width, petal length, petal width and class. There were 150 entries; 50 for each of the iris species: setosa, versicolour and virginica. There were no missing values or Nan-values. Numerical values in dataset were already provided as float, while 'class' series classifying each entry into a species was an object.

Descriptive analysis

Descriptive analysis of the dataset summarizing each series showing for example mean, standard deviation (std) and median (50%) is shown in *Table 1*. Values given are for all species pooled together.

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Table 1. Descriptive analyses of the dataset with all species pooled together.

Of more interest is descriptive analysis of each iris species separately as shown in Tables 2. Results show lower measures of spread as expected.

Descriptive analysis for Iris-setosa:

	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.000000	50.000000	50.00000
mean	5.00600	3.418000	1.464000	0.24400
std	0.35249	0.381024	0.173511	0.10721
min	4.30000	2.300000	1.000000	0.10000
25%	4.80000	3.125000	1.400000	0.20000
50%	5.00000	3.400000	1.500000	0.20000
75%	5.20000	3.675000	1.575000	0.30000
max	5.80000	4.400000	1.900000	0.60000

Descriptive analysis for Iris-versicolor:

	sepal_length	sepal_width	petal_length	petal_width
count	50.000000	50.000000	50.000000	50.000000
mean	5.936000	2.770000	4.260000	1.326000
std	0.516171	0.313798	0.469911	0.197753
min	4.900000	2.000000	3.000000	1.000000
25%	5.600000	2.525000	4.000000	1.200000
50%	5.900000	2.800000	4.350000	1.300000
75%	6.300000	3.000000	4.600000	1.500000
max	7.000000	3.400000	5.100000	1.800000

Descriptive analysis for Iris-virginica:

	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.000000	50.000000	50.00000
mean	6.58800	2.974000	5.552000	2.02600
std	0.63588	0.322497	0.551895	0.27465
min	4.90000	2.200000	4.500000	1.40000
25%	6.22500	2.800000	5.100000	1.80000
50%	6.50000	3.000000	5.550000	2.00000
75%	6.90000	3.175000	5.875000	2.30000
max	7.90000	3.800000	6.900000	2.50000

Table 2. Descriptive analyses of each species separately.

Q1. Construct a 95% confidence interval for means for sepal width for each species

To construct confidence intervals, the t-distribution is used since sigma of the population is not known. The following confidence intervals were obtained at 95% level:

- Iris-setosa: 3.31-3.53
- Iris-versicolor: 2.68-2.86
- Iris-virginica: 2.88-3.07

None of the confidence intervals is overlapping thus mean values for all three species are significantly different.

Q2 Is sepal length for Iris-virginica different from sepal length for Iris-versicolor?

Variances for the two sample groups are not equal since F statistic (= 1.518) is larger than F critical for 49 and 49 degrees of freedom (= 0.622). H0 can thus be discarded.

Following hypothesis is to be tested on the two means:

H0: μ_virginica = μ_versicolor
Ha: μ_virginica ≠ μ_versicolor

p-value for the t-test is 1.866e-07, which is much smaller than alpha (0.05). Thus H0 hypothesis can be discarded concluding that the sepal lengths for virginica and versicolor are different. A plot of sepal length data points is shown in Figure 3, where visually sepal length of virginica is larger/different than versicolor, supporting the hypothesis test result.

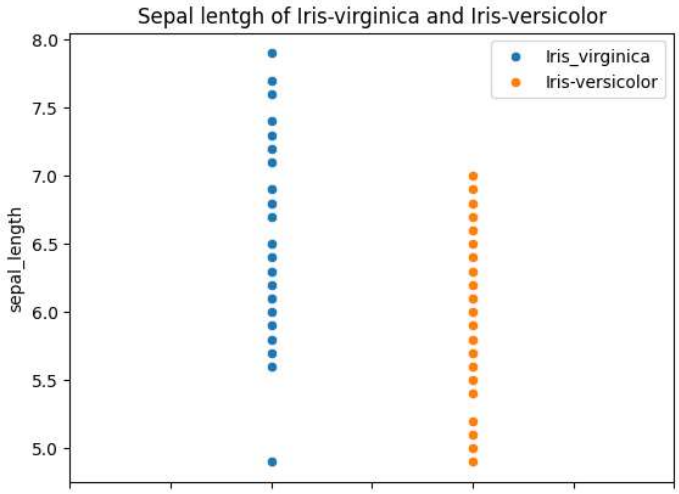


Figure 3. Sepal length of Iris-virginica and Iris-vesicolor.

Q3 Visualize data for pair-wise dependencies.

Visualization of all series pairwise (Figure 4) with species indicated shows that iris setosa (blue) can be distinguished from the other species as the blue circles are clearly isolated from green and orange for all combinations.

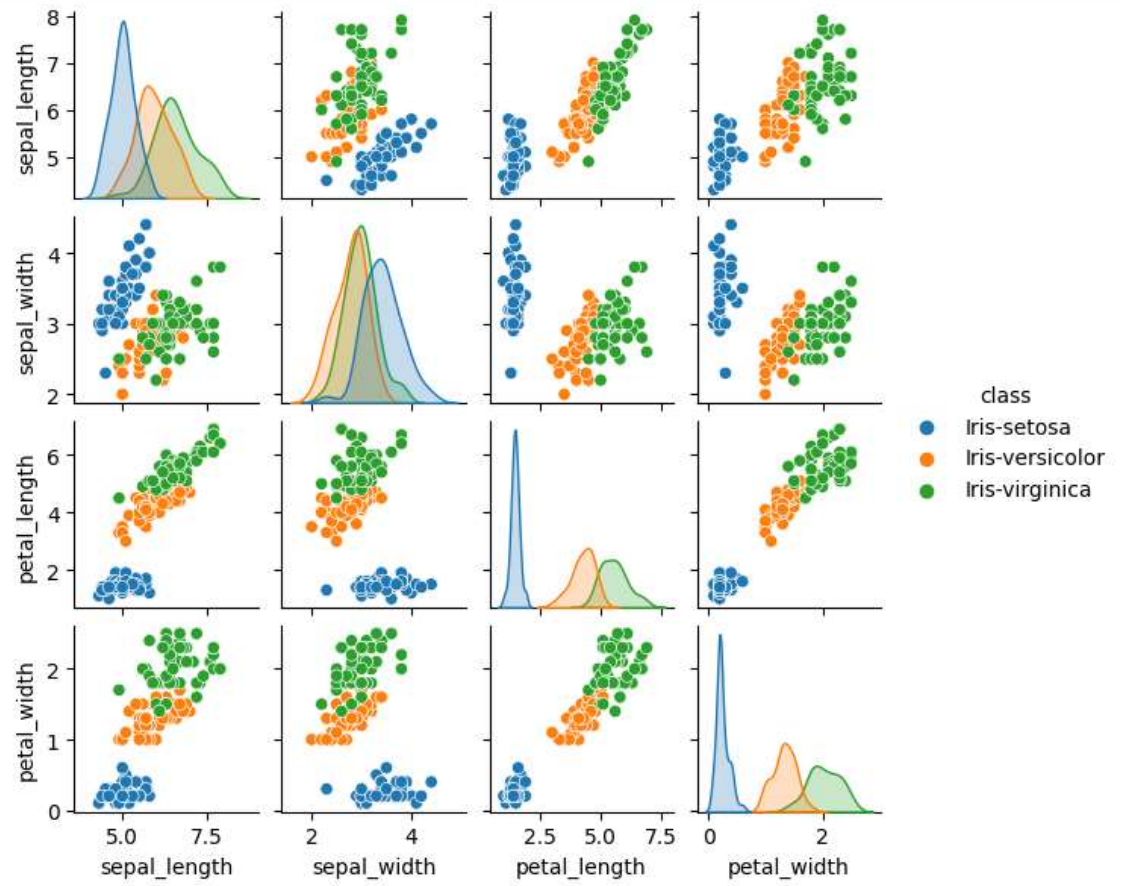


Figure 4. Pairwise dependance plots for the iris dataset.

Q4: Describe dependency between petal length and petal width for setosa and compare with versicolor.

Ordinary least squares (OLS) regression of petal length and width are shown in Table 3 and Table 4. Correlation is very week for setosa with R2 = 0.094 and stronger for versicolor with R2 = 0.619.

Same trends can be seen in graphs in Figure 5, where data is much more scattered with broad confidence intervals for setosa compared to versicolor. Both linear regression models approach 0 for petal width when petal length approaches 0, which is expected.

OLS Regression Results

Dep. Variable:	petal_length	R-squared:	0.094
Model:	OLS	Adj. R-squared:	0.075
Method:	Least Squares	F-statistic:	4.970
Date:	Sat, 04 Feb 2023	Prob (F-statistic):	0.0305
Time:	16:59:37	Log-Likelihood:	19.597
No. Observations:	50	AIC:	-35.19
Df Residuals:	48	BIC:	-31.37
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.3430	0.059	22.698	0.000	1.224	1.462
petal_width	0.4957	0.222	2.229	0.031	0.049	0.943

Omnibus:	1.837	Durbin-Watson:	1.817
Prob(Omnibus):	0.399	Jarque-Bera (JB):	1.025
Skew:	0.077	Prob(JB):	0.599
Kurtosis:	3.685	Cond. No.	9.99

Table 3 Results from OLS regression of petal length and width for setosa.

OLS Regression Results

Dep. Variable:	petal_length	R-squared:	0.619
Model:	OLS	Adj. R-squared:	0.611
Method:	Least Squares	F-statistic:	77.93
Date:	Sat, 04 Feb 2023	Prob (F-statistic):	1.27e-11
Time:	17:03:08	Log-Likelihood:	-8.5674
No. Observations:	50	AIC:	21.13
Df Residuals:	48	BIC:	24.96
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.7813	0.284	6.276	0.000	1.211	2.352
petal_width	1.8693	0.212	8.828	0.000	1.444	2.295

Omnibus:	2.041	Durbin-Watson:	2.149
Prob(Omnibus):	0.360	Jarque-Bera (JB):	1.188
Skew:	-0.312	Prob(JB):	0.552
Kurtosis:	3.425	Cond. No.	14.2

Table 4. Results from OLS regression of petal length and width for versicolor.

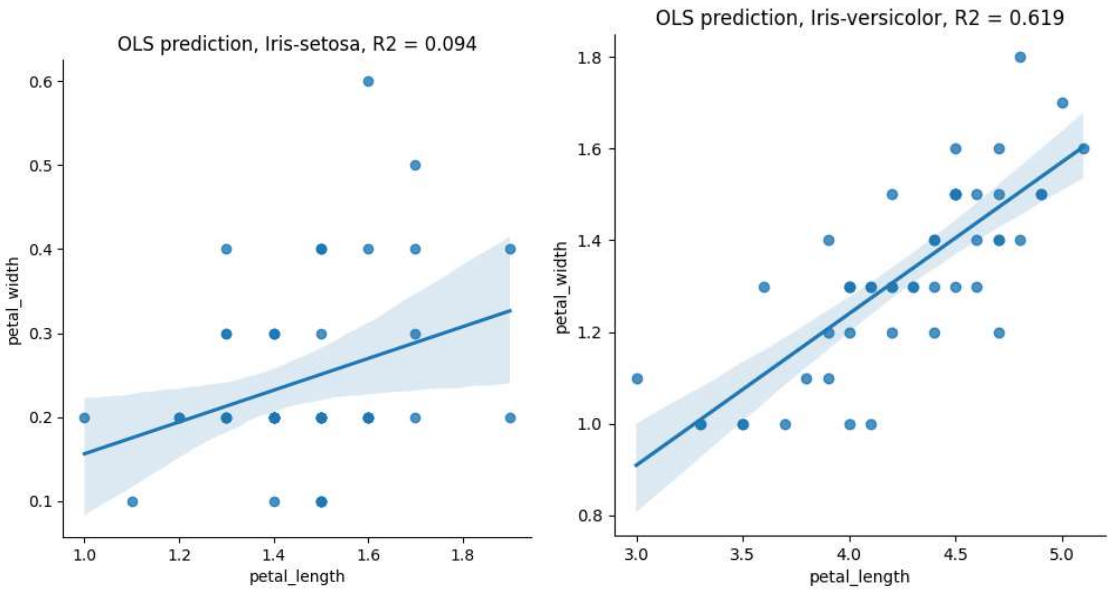


Figure 5. OLS predictions for correlation of petal width and length for setosa (left) and versicolor (right)

Conclusions

Scipy and Statsmodels have together with Pandas and Numpy been a powerful and simple tool for data analysis of Iris dataset.

Appendix

In [148…

```
# Importing necessary packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as scs
import seaborn as sns

from scipy.stats import t, f, ttest_ind
from statsmodels.formula.api import ols
```

In [149…

```
# Importing iris dataset as a Pandas dataframe
df = pd.read_csv('../Dataset/iris.csv')
```

In [150…

```
# Initial visualizatin of the dataframe
df.head(3)
```

Out[150…

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa

In [151…

```
# renaming colmns to remove space in he beginning of series name
df=df.rename(columns={' sepal_width':'sepal_width', ' petal_length':'petal_length', ' petal_width':'petal_width', ' class':'class'})
```

In [152…

```
# Initial check of data shows that there are 5 series, 150 entries, no NaN values.
# Series expected to contain measurements are of type float
# Series expected to contain text is of type object
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
```

```
 4   class         150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [153…

```
# There are 3 species of iris in series 'class' with 50 entries each.
df['class'].value_counts()
```

Out[153…

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: class, dtype: int64
```

In [154…

```
# Descriptive analysis of the dataframe shows for mean, standard deviation (std) and median (50%) for each data series.
df.describe()
```

Out[154…

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [155…

```
# Descriptive analysis of each species separately shows mean, standard deviation (std) and median (50%) for each data series.

species_list = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'] # list of species in dataframe

for species in species_list:
    print(f'Descriptive analysis for {species}:') # print a heading line
    print(f"{df[df['class'] == species].describe()}\n") # print result from describe() funktion on dataframe filtered for each species
```

```
Descriptive analysis for Iris-setosa:
       sepal_length  sepal_width  petal_length  petal_width
count      50.00000    50.000000     50.000000     50.00000
mean        5.00600     3.418000      1.464000      0.24400
std         0.35249     0.381024      0.173511      0.10721
min         4.30000     2.300000      1.000000      0.10000
25%         4.80000     3.125000      1.400000      0.20000
```

```
50%         5.00000     3.400000      1.500000      0.20000
75%         5.20000     3.675000      1.575000      0.30000
max         5.80000     4.400000      1.900000      0.60000

Descriptive analysis for Iris-versicolor:
       sepal_length  sepal_width  petal_length  petal_width
count     50.000000    50.000000     50.000000    50.000000
mean       5.936000     2.770000      4.260000     1.326000
std        0.516171     0.313798      0.469911     0.197753
min        4.900000     2.000000      3.000000     1.000000
25%        5.600000     2.525000      4.000000     1.200000
50%        5.900000     2.800000      4.350000     1.300000
75%        6.300000     3.000000      4.600000     1.500000
max        7.000000     3.400000      5.100000     1.800000

Descriptive analysis for Iris-virginica:
       sepal_length  sepal_width  petal_length  petal_width
count      50.00000    50.000000     50.000000     50.00000
mean        6.58800     2.974000      5.552000      2.02600
std         0.63588     0.322497      0.551895      0.27465
min         4.90000     2.200000      4.500000      1.40000
25%         6.22500     2.800000      5.100000      1.80000
50%         6.50000     3.000000      5.550000      2.00000
75%         6.90000     3.175000      5.875000      2.30000
max         7.90000     3.800000      6.900000      2.50000
```

In [157…

```python
# Q1. Construct a 95% confidence interval for means for sepal width for each species
# filtering used in the for loop for example for setosa: setosa = df[df['class'] == 'Iris-setosa']

species_list = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'] # list of species in dataframe
alpha = 0.05  # confidence interval
n= 50 # number of samples for each species

for species in species_list:                                    # calculations for each species in the list
    sem = scs.sem(df[df['class'] == species]['sepal_width'])    # Standard error for the specific species
    mean = df[df['class'] == species]['sepal_width'].mean()     # Average for the specific species
    lower, upper = t.interval(confidence=1-alpha, df=n-1, loc=mean, scale=sem) # Calculating lower and higher limits of the confidence i
    print(f'Confidence interval for {species} is : {lower:.2f}-{upper:.2f}') # Printing confidence intervall
```

```
Confidence interval for Iris-setosa is : 3.31-3.53
Confidence interval for Iris-versicolor is : 2.68-2.86
Confidence interval for Iris-virginica is : 2.88-3.07
```

In [165…

```python
# Q2. Is sepal length for Iris-virginica different from sepal length for Iris-versicolor?

# Test is done in 2 steps:
# 1. test if variances for the two populations are equal
```

```
# 1. test if variances for the two populations are equal
        # H_a: sigma_virginica != sigma_versicolor
        # H_0: sigma_virginica = sigma_versicolor
# 2. Hypothesis testing where:
        # H_a: μ_virginica != μ_versicolor
        # H_0: μ_virginica = μ_versicolor


filt_virginica = (df['class'] == 'Iris-virginica')      # Filter to filter out Iris-virginica from full dataframe df
filt_versicolor = (df['class'] == 'Iris-versicolor')    # Filter to filter out Iris-versicolor from full dataframe df
n = 50                                                  # Both series contain 50 samples
alpha = 0.05

# Test of variances:
f_statistic = df[filt_virginica]["sepal_length"].var()/df[filt_versicolor]["sepal_length"].var() # Statistica for samples: var_virginica
f_crit_49_49 = f.ppf(q=0.05, dfn=n-1, dfd=n-1) # Critical F value for 49,49 degrees of freedom
print(f'Test of variances:\n{f_statistic = :.3f}, {f_crit_49_49 = :.3f}') # printing results with 3 decimals
print(f'H0 can be discarded since f_statistica is larger than f_crit')
print(f'Conclusion: Variances for the two populations can not be assumed equal.\n')

# Test of means:
# a and b are series sepal_lengt filtered out for each class of iris; False based on test of variances above; two-sided since question w
ttest_means = ttest_ind(a=df[filt_virginica]["sepal_length"], b=df[filt_versicolor]["sepal_length"], equal_var=False, alternative="two-s
print(f'Test of means:\n{ttest_means = }')
print(f'H0 hypothsis can be discarded, since the p-value from t-test (1.87e-07) is smaller than alpha (0.05)')
print(f'Conclusion: The sepial lengths for virginica and versicolor are different.')
```

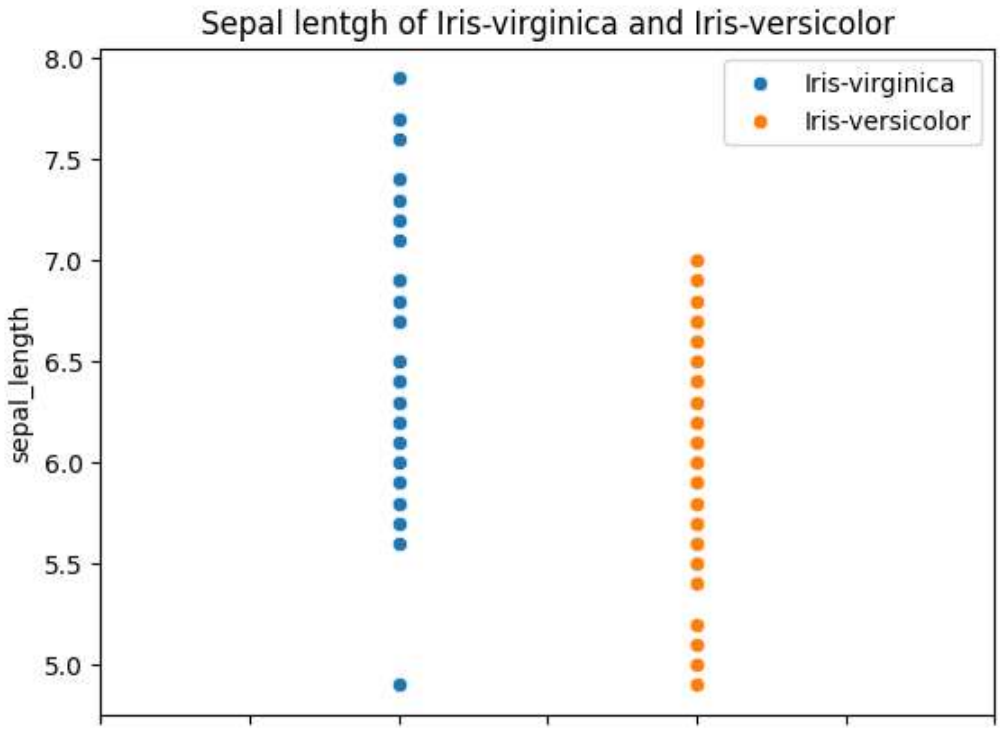
```
Test of variances:
f_statistic = 1.518, f_crit_49_49 = 0.622
H0 can be discarded since f_statistica is larger than f_crit
Conclusion: Variances for the two populations can not be assumed equal.

Test of means:
ttest_means = Ttest_indResult(statistic=5.629165259719801, pvalue=1.8661443873771216e-07)
H0 hypothsis can be discarded, since the p-value from t-test (1.87e-07) is smaller than alpha (0.05)
Conclusion: The sepial lengths for virginica and versicolor are different.
```

In [159…

```
# Q2 Plot of sepal length for Iris-virginica and Iris-versicolor

# Arrays of ones and twos to be used as x values so sepal length is not plotted as function of index
x_1 = np.ones(50, dtype = int)
x_2 = 2 * x_1

sns.scatterplot(x=x_1, y=df[filt_virginica]["sepal_length"]).set(xticklabels=[]) # Sepal length for virginica plotted with x = 1, x-labe
sns.scatterplot(x=x_2, y=df[filt_versicolor]["sepal_length"]).set(xticklabels=[]) # # Sepal length for versicolor plotted with x = 2, x-
plt.title("Sepal lentgh of Iris-virginica and Iris-versicolor") # Plot title
plt.xlim(0, 3) # x-values are 0 to 3 so x_1 and x_2 are centered in figure
plt.legend(labels=['Iris-virginica', 'Iris-versicolor']) # legend is added
```

Out[159…

```
<matplotlib.legend.Legend at 0x1b00038ba90>
```

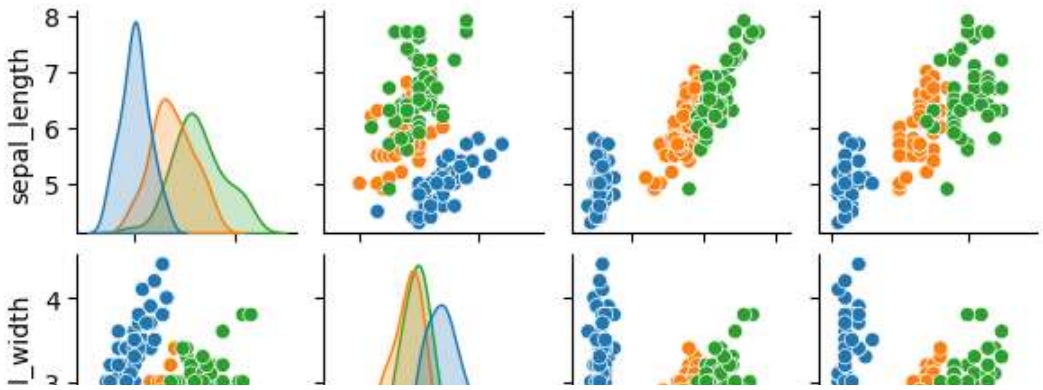


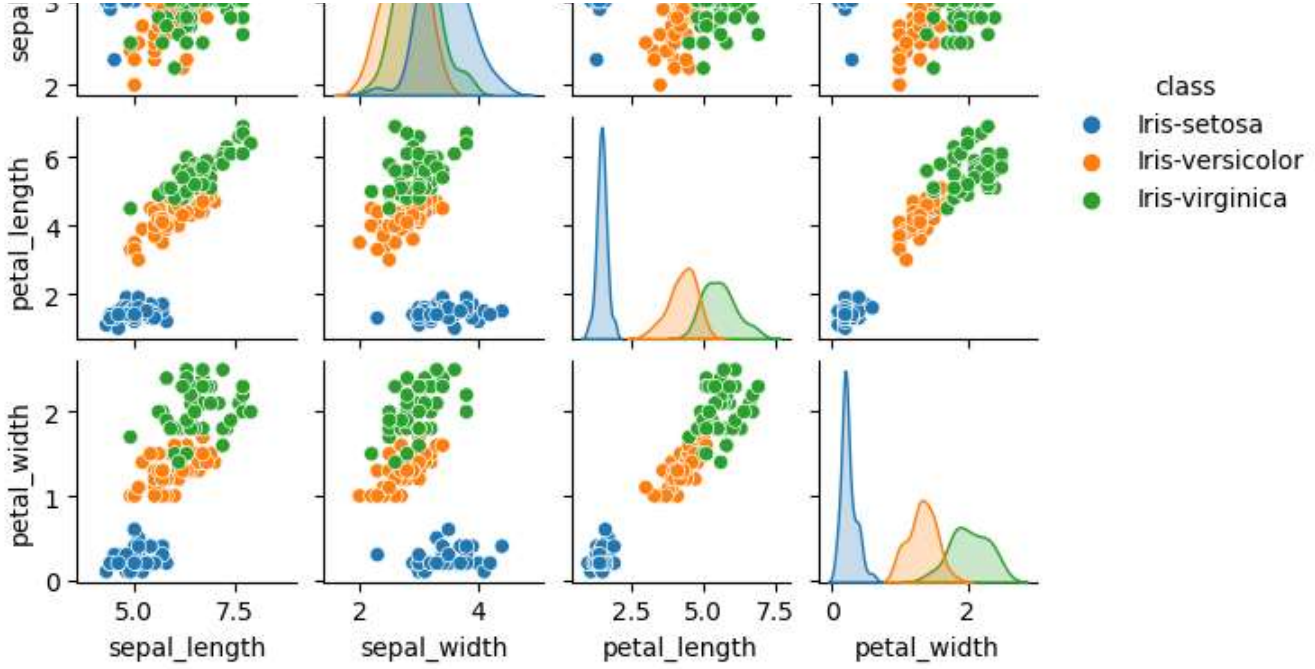
In [160…

```
# Q3. Visulazation of all series relationships with class as hue
# ref: https://www.geeksforgeeks.org/exploratory-data-analysis-on-iris-dataset/
sns.pairplot(data=df, hue='class', height=1.5)
```

Out[160…

```
<seaborn.axisgrid.PairGrid at 0x1b0003cd490>
```





In [161…

```
# Q4. Describe dependency between petal lenght and petal width for Iris-setosa and compare with Iris-versicolor

# Regression with ordinary least squares
filt_setosa = (df['class'] == 'Iris-setosa')      # Filter to filter out Iris-setosa from full dataframe df
model = ols('petal_length ~ petal_width', data=df[filt_setosa]).fit()
print(model.summary())
print('Ordinary least squares shows poor dependency with R-squared 0.094')
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           petal_length   R-squared:                       0.094
Model:                            OLS   Adj. R-squared:                  0.075
Method:                 Least Squares   F-statistic:                     4.970
Date:                Sun, 05 Feb 2023   Prob (F-statistic):             0.0305
Time:                        01:34:15   Log-Likelihood:                 19.597
No. Observations:                  50   AIC:                            -35.19
Df Residuals:                      48   BIC:                            -31.37
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.3430      0.059     22.698      0.000       1.224       1.462
petal_width    0.4957      0.222      2.229      0.031       0.049       0.943
```

```
==============================================================================
Omnibus:                        1.837   Durbin-Watson:                   1.817
Prob(Omnibus):                  0.399   Jarque-Bera (JB):                1.025
Skew:                           0.077   Prob(JB):                        0.599
Kurtosis:                       3.685   Cond. No.                         9.99
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

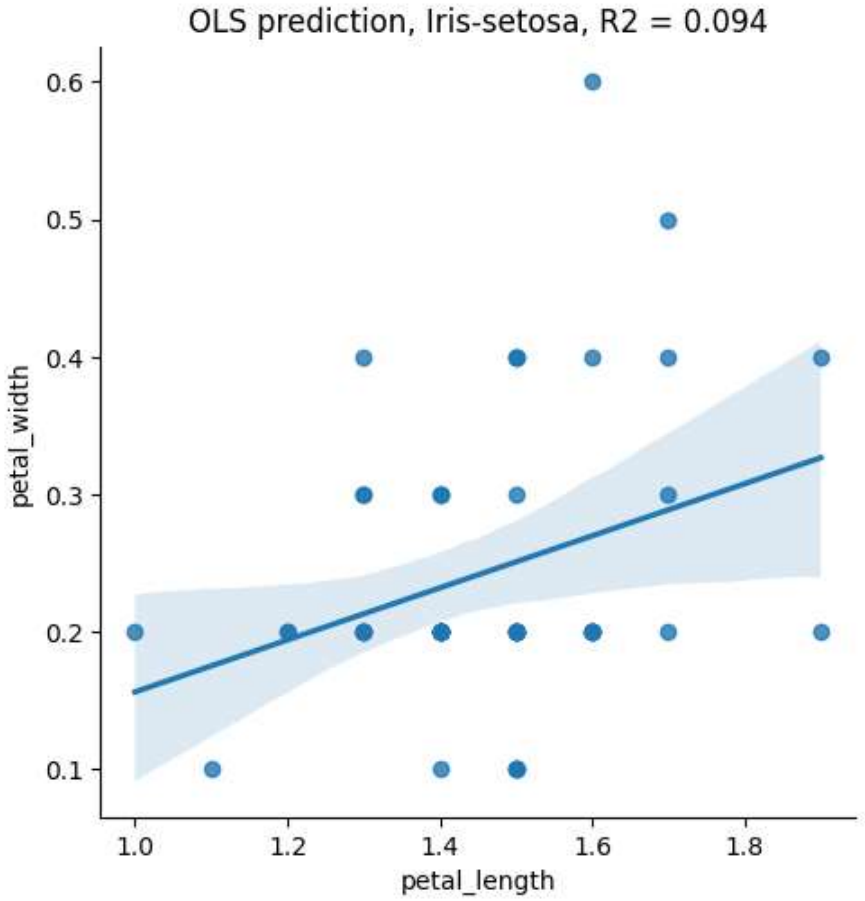
Ordinary least squares shows poor dependency with R-squared 0.094

In [162…

```
# Q4: In sample prediction for Isis-setosa
sns.lmplot(data=df[filt_setosa], x='petal_length', y='petal_width').set(title = "OLS prediction, Iris-setosa, R2 = 0.094")
```

Out[162…

```
<seaborn.axisgrid.FacetGrid at 0x1b000c44eb0>
```



In [163…

```
# Q4. Describe dependency between petal lenght and petal width for Iris-versicolor

# Regression with ordinary least squares
filt_versicolor = (df['class'] == 'Iris-versicolor')      # Filter to filter out Iris-versicolor from full dataframe df
model = ols('petal_length ~ petal_width', data=df[filt_versicolor]).fit()
print(model.summary())
print('Ordinary least squares shows better dependency with R-squared 0.619')
```

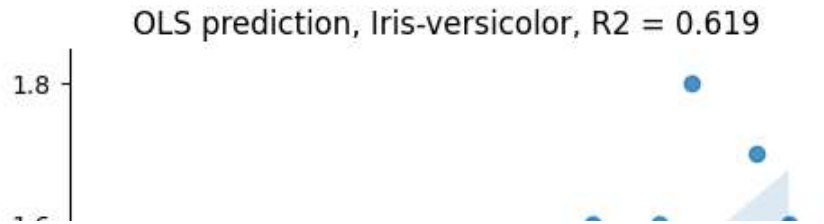
```
                            OLS Regression Results
==============================================================================
Dep. Variable:           petal_length   R-squared:                       0.619
Model:                            OLS   Adj. R-squared:                  0.611
Method:                 Least Squares   F-statistic:                     77.93
Date:                Sun, 05 Feb 2023   Prob (F-statistic):           1.27e-11
Time:                        01:34:16   Log-Likelihood:                -8.5674
No. Observations:                  50   AIC:                             21.13
Df Residuals:                      48   BIC:                             24.96
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.7813      0.284      6.276      0.000       1.211       2.352
petal_width    1.8693      0.212      8.828      0.000       1.444       2.295
==============================================================================
Omnibus:                        2.041   Durbin-Watson:                   2.149
Prob(Omnibus):                  0.360   Jarque-Bera (JB):                1.188
Skew:                          -0.312   Prob(JB):                        0.552
Kurtosis:                       3.425   Cond. No.                         14.2
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
Ordinary least squares shows better dependency with R-squared 0.619
```

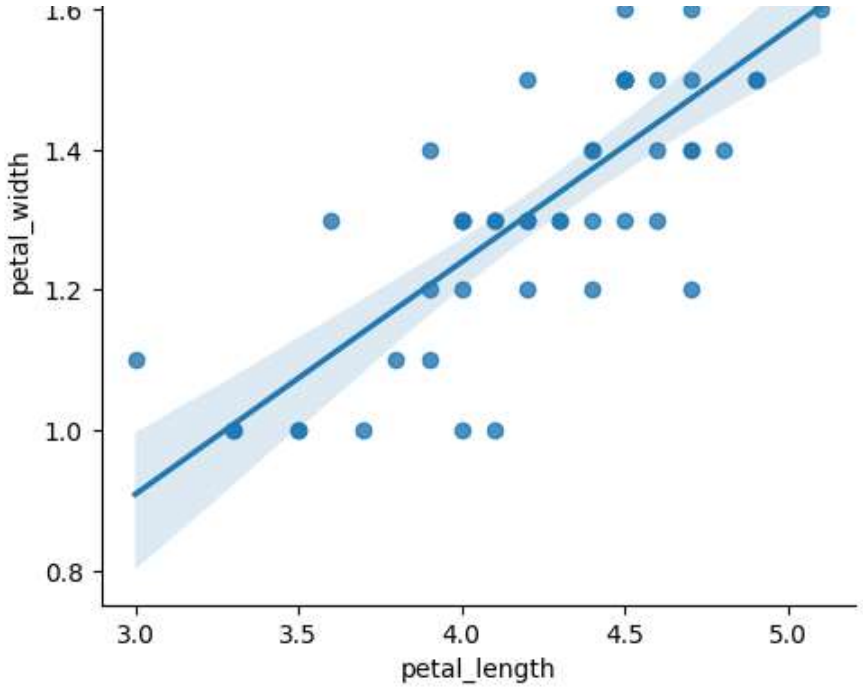
In [164…

```
# Q4: In sample prediction for Iris-versicolor
sns.lmplot(data=df[filt_versicolor], x='petal_length', y='petal_width').set(title = "OLS prediction, Iris-versicolor, R2 = 0.619")
```

Out[164…

```
<seaborn.axisgrid.FacetGrid at 0x1b001063a00>
```




1.6
1.4
1.2
1.0
0.8
petal_width
3.0
3.5
4.0
4.5
5.0
petal_length