Lab 3: NumPy in Python

Matthew Loh

Table of contents

1 A. Creating dataframes : reading data files or converting arrays

```
import pandas as pd
data = pd.read_csv('./brain_size.csv', sep=';', na_values=".")
data
```

	Unnamed: 0	Gender	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count
0	1	Female	133	132	124	118.0	64.5	816932
1	2	Male	140	150	124	NaN	72.5	1001121

	Unnamed: 0	Gender	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count
2	3	Male	139	123	150	143.0	73.3	1038437
3	4	Male	133	129	128	172.0	68.8	965353
4	5	Female	137	132	134	147.0	65.0	951545
5	6	Female	99	90	110	146.0	69.0	928799
6	7	Female	138	136	131	138.0	64.5	991305
7	8	Female	92	90	98	175.0	66.0	854258
8	9	Male	89	93	84	134.0	66.3	904858
9	10	Male	133	114	147	172.0	68.8	955466
10	11	Female	132	129	124	118.0	64.5	833868
11	12	Male	141	150	128	151.0	70.0	1079549
12	13	Male	135	129	124	155.0	69.0	924059
13	14	Female	140	120	147	155.0	70.5	856472
14	15	Female	96	100	90	146.0	66.0	878897
15	16	Female	83	71	96	135.0	68.0	865363
16	17	Female	132	132	120	127.0	68.5	852244
17	18	Male	100	96	102	178.0	73.5	945088
18	19	Female	101	112	84	136.0	66.3	808020
19	20	Male	80	77	86	180.0	70.0	889083
20	21	Male	83	83	86	NaN	NaN	892420
21	22	Male	97	107	84	186.0	76.5	905940
22	23	Female	135	129	134	122.0	62.0	790619
23	24	Male	139	145	128	132.0	68.0	955003
24	25	Female	91	86	102	114.0	63.0	831772
25	26	Male	141	145	131	171.0	72.0	935494
26	27	Female	85	90	84	140.0	68.0	798612
27	28	Male	103	96	110	187.0	77.0	1062462
28	29	Female	77	83	72	106.0	63.0	793549
29	30	Female	130	126	124	159.0	66.5	866662
30	31	Female	133	126	132	127.0	62.5	857782
31	32	Male	144	145	137	191.0	67.0	949589
32	33	Male	103	96	110	192.0	75.5	997925
33	34	Male	90	96	86	181.0	69.0	879987
34	35	Female	83	90	81	143.0	66.5	834344
35	36	Female	133	129	128	153.0	66.5	948066
36	37	Male	140	150	124	144.0	70.5	949395
37	38	Female	88	86	94	139.0	64.5	893983
38	39	Male	81	90	74	148.0	74.0	930016
39	40	Male	89	91	89	179.0	75.5	935863

```python
import numpy as np
t = np.linspace(-6, 6, 20)
sin_t = np.sin(t)
cos_t = np.cos(t)

pd.DataFrame({'t': t, 'sin': sin_t, 'cos': cos_t})
```

	t	sin	cos
0	-6.000000	0.279415	0.960170
1	-5.368421	0.792419	0.609977
2	-4.736842	0.999701	0.024451
3	-4.105263	0.821291	-0.570509
4	-3.473684	0.326021	-0.945363
5	-2.842105	-0.295030	-0.955488
6	-2.210526	-0.802257	-0.596979
7	-1.578947	-0.999967	-0.008151
8	-0.947368	-0.811882	0.583822
9	-0.315789	-0.310567	0.950551
10	0.315789	0.310567	0.950551
11	0.947368	0.811882	0.583822
12	1.578947	0.999967	-0.008151
13	2.210526	0.802257	-0.596979
14	2.842105	0.295030	-0.955488
15	3.473684	-0.326021	-0.945363
16	4.105263	-0.821291	-0.570509
17	4.736842	-0.999701	0.024451
18	5.368421	-0.792419	0.609977
19	6.000000	-0.279415	0.960170

2 B. Manipulating data

```python
data.shape  # 40 rows and 8 columns
data.columns  # it has columns

print(data['Gender'])
# Simpler selector
data[data['Gender'] == 'Female']['VIQ'].mean()
```

```
groupby_gender = data.groupby('Gender')
for gender, value in groupby_gender['VIQ']:
    print((gender, value.mean()))

groupby_gender.mean()
```

```
0     Female
1       Male
2       Male
3       Male
4     Female
5     Female
6     Female
7     Female
8       Male
9       Male
10    Female
11      Male
12      Male
13    Female
14    Female
15    Female
16    Female
17      Male
18    Female
19      Male
20      Male
21      Male
22    Female
23      Male
24    Female
25      Male
26    Female
27      Male
28    Female
29    Female
30    Female
31      Male
32      Male
33      Male
34    Female
35    Female
```

```
36      Male
37    Female
38      Male
39      Male
Name: Gender, dtype: object
('Female', 109.45)
('Male', 115.25)
```

Gender	Unnamed: 0	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count
Female	19.65	111.9	109.45	110.45	137.200000	65.765000	862654.6
Male	21.35	115.0	115.25	111.60	166.444444	71.431579	954855.4

3 Exercise 3.1

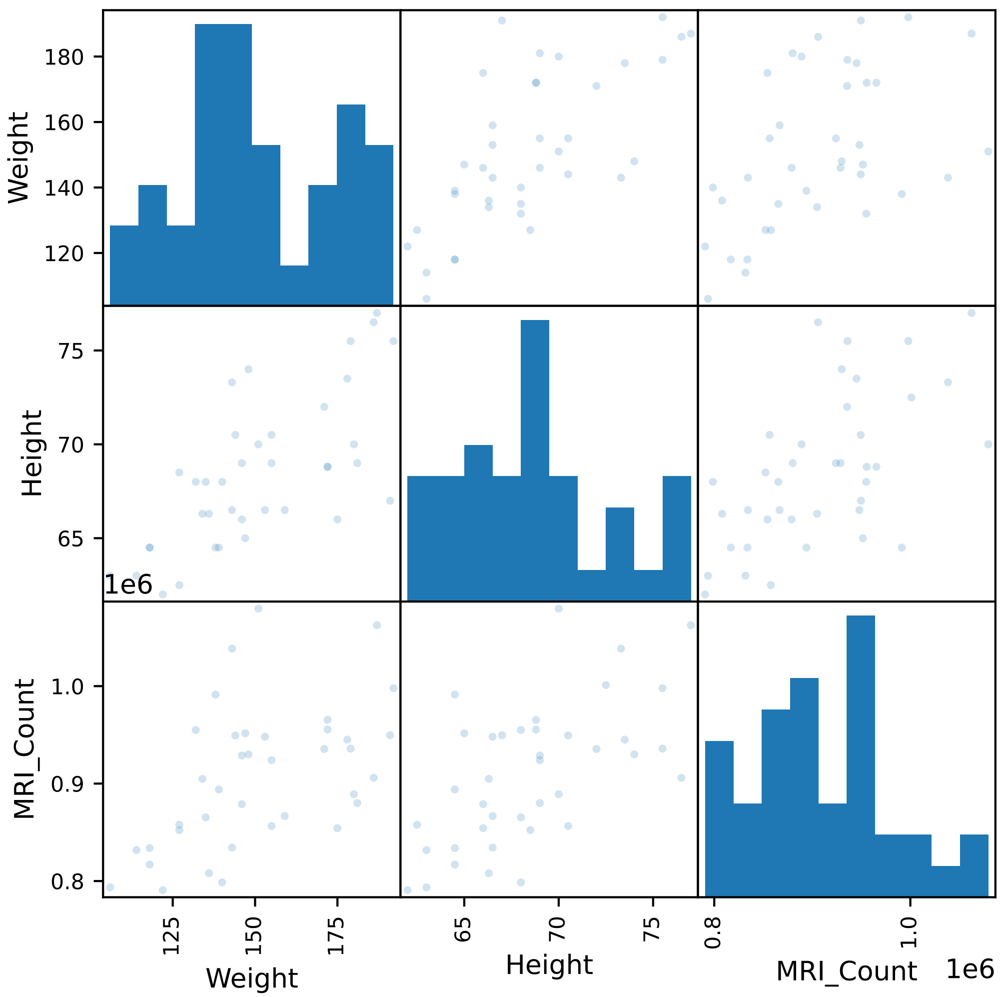
- What is the mean value for VIQ for the full population?
- What is the average value of MRI counts, for males and females?

```python
# Mean value for VIQ
print(data['VIQ'].mean())

# Average value of MRI counts

print(groupby_gender['MRI_Count'].mean())
# Average value of MRI counts for males
print(data[data['Gender'] == 'Male']['MRI_Count'].mean())
print(data[data['Gender'] == 'Female']['MRI_Count'].mean())
```

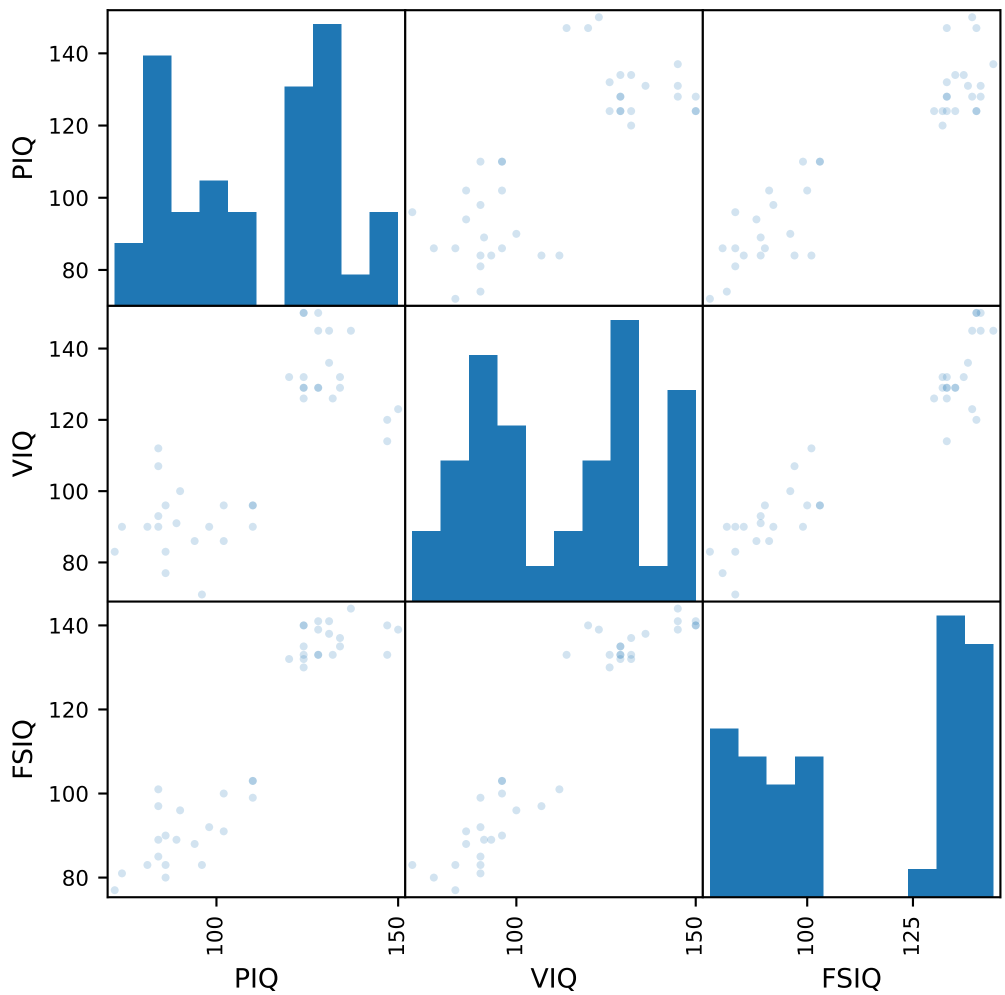
```
112.35
Gender
Female    862654.6
Male      954855.4
Name: MRI_Count, dtype: float64
954855.4
862654.6
```

4 C. Plotting Data

```
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
data = pd.read_csv('./brain_size.csv', sep=';', na_values=".")

column1 = ['Weight', 'Height', 'MRI_Count']
column2 = ['PIQ', 'VIQ', 'FSIQ']
scatter_matrix(data[column1],
               alpha=0.2, figsize=(6, 6), diagonal='hist')
scatter_matrix(data[column2],
               alpha=0.2, figsize=(6, 6), diagonal='hist')
plt.show()

from statsmodels.formula.api import ols
model = ols("VIQ ~ Gender + MRI_Count + Height", data).fit()
print(model.summary())
```



Weight
Height
MRI_Count
180
160
140
120
75
70
65
1e6
1.0
0.9
0.8
125
150
175
65
70
75
0.8
1.0
Weight
Height
MRI_Count
1e6



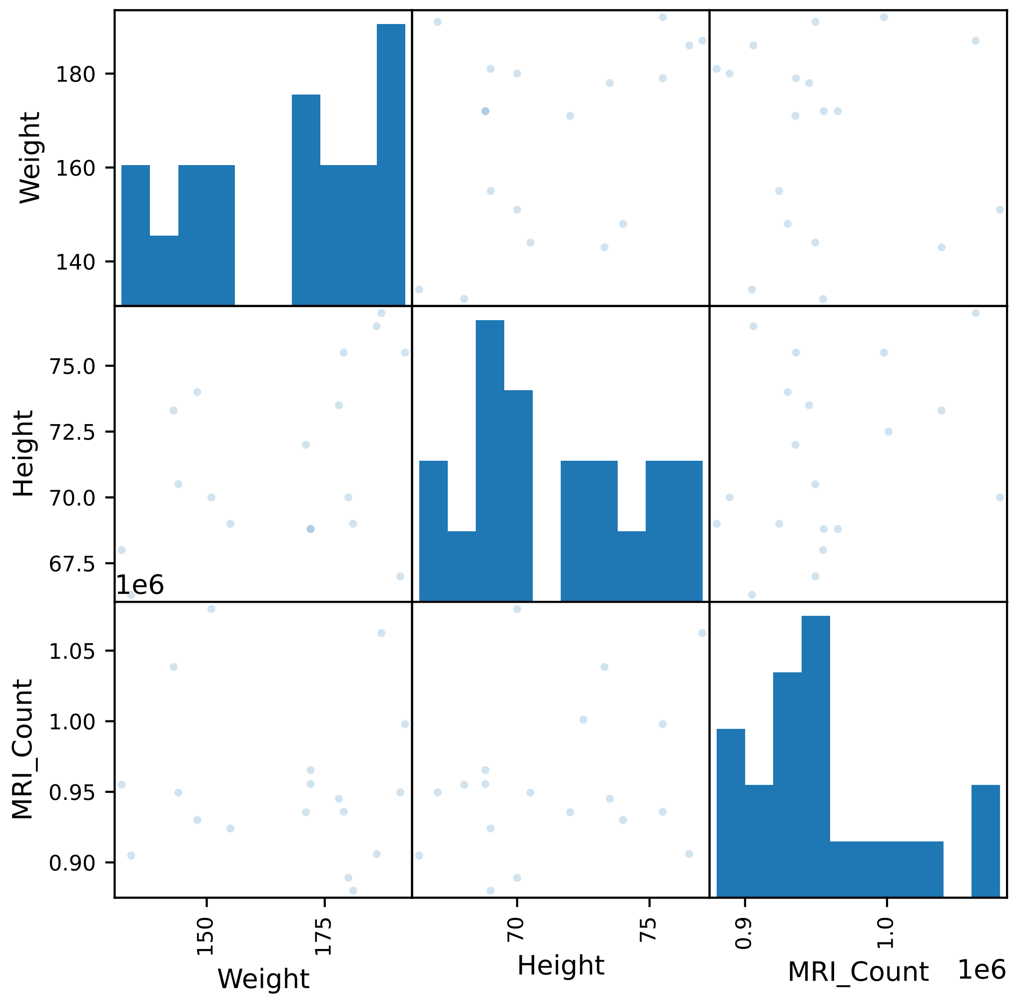

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    VIQ   R-squared:                       0.246
Model:                            OLS   Adj. R-squared:                  0.181
Method:                 Least Squares   F-statistic:                     3.809
Date:                Wed, 08 May 2024   Prob (F-statistic):             0.0184
Time:                        11:41:00   Log-Likelihood:                -172.34
No. Observations:                  39   AIC:                             352.7
Df Residuals:                      35   BIC:                             359.3
Df Model:                           3
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
```

```
==============================================================================
Intercept        166.6258     88.824      1.876      0.069     -13.696     346.948
Gender[T.Male]     8.8524     10.710      0.827      0.414     -12.890      30.595
MRI_Count          0.0002   6.46e-05      2.615      0.013    3.78e-05       0.000
Height            -3.0837      1.276     -2.417      0.021      -5.674      -0.494
==============================================================================
Omnibus:                        7.373   Durbin-Watson:                   2.109
Prob(Omnibus):                  0.025   Jarque-Bera (JB):                2.252
Skew:                           0.005   Prob(JB):                        0.324
Kurtosis:                       1.823   Cond. No.                     2.40e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.4e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

5 Exercise 3.2

- Plot the scatter matrix for males only, and for females only

```python
scatter_matrix = pd.plotting.scatter_matrix(
    data[data['Gender'] == 'Male'][column1],
    alpha=0.2, figsize=(6, 6), diagonal='hist')
```

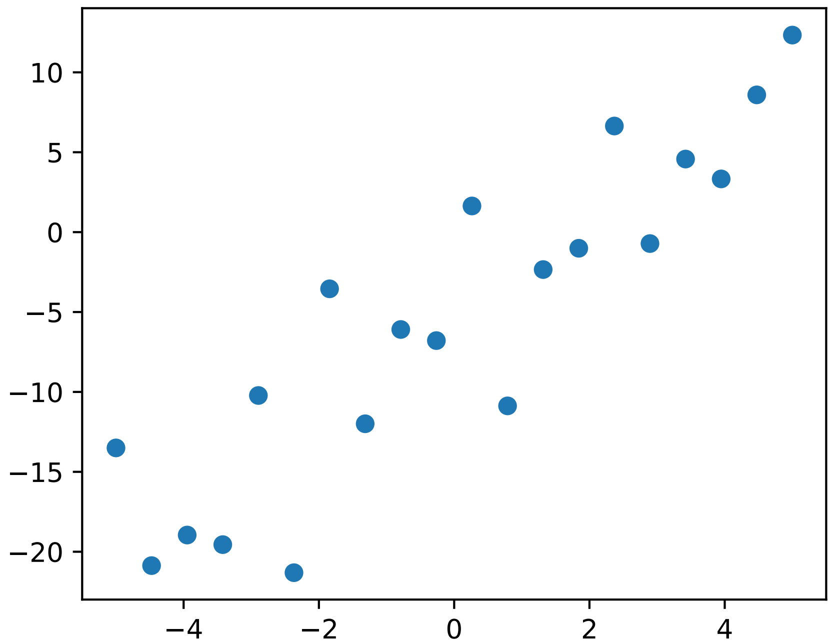


6 D-1: Linear models, multiple factors, and analysis of variance

```
from statsmodels.formula.api import ols
import numpy as np
x = np.linspace(-5, 5, 20)
np.random.seed(1)
# normal distributed noise
y = -5 + 3*x + 4 * np.random.normal(size=x.shape)
# Plot the data
plt.figure(figsize=(5, 4))
plt.plot(x, y, 'o')
```

```python
# Create a data frame containing all the relevant variables
data = pd.DataFrame({'x': x, 'y': y})

model = ols("y ~ x", data).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.804
Model:                            OLS   Adj. R-squared:                  0.794
Method:                 Least Squares   F-statistic:                     74.03
Date:                Wed, 08 May 2024   Prob (F-statistic):           8.56e-08
Time:                        11:41:00   Log-Likelihood:                -57.988
No. Observations:                  20   AIC:                             120.0
Df Residuals:                      18   BIC:                             122.0
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -5.5335      1.036     -5.342      0.000      -7.710      -3.357
x              2.9369      0.341      8.604      0.000       2.220       3.654
==============================================================================
Omnibus:                        0.100   Durbin-Watson:                   2.956
Prob(Omnibus):                  0.951   Jarque-Bera (JB):                0.322
Skew:                          -0.058   Prob(JB):                        0.851
Kurtosis:                       2.390   Cond. No.                         3.03
==============================================================================

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



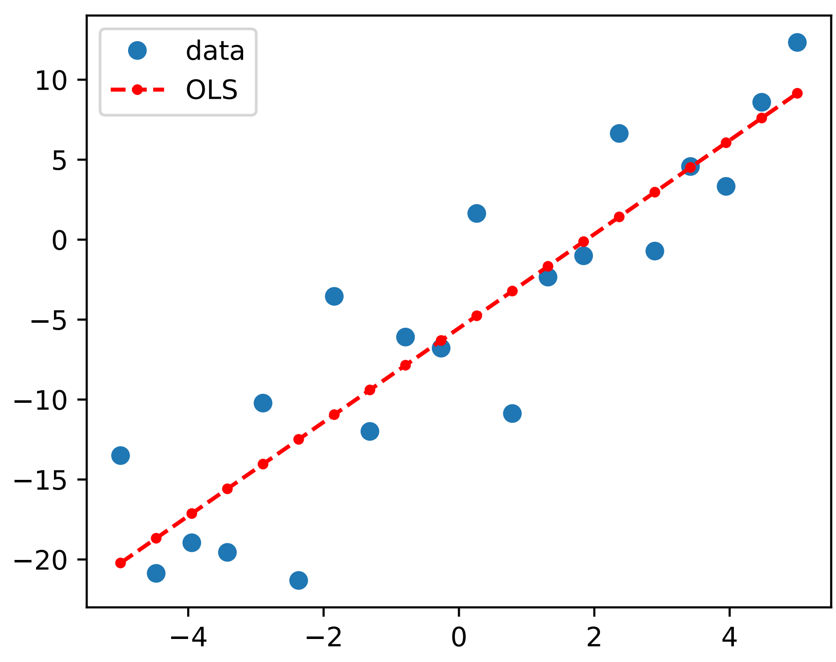
7 Exercise 3.3

- Similar to the model above, use Analysis of Variance (ANOVA) on linear models, plot the fitted model and retrieve the parameter estimates.

```
model = ols("y ~ x", data).fit()
print(model.summary())

# Plot the data
plt.figure(figsize=(5, 4))
plt.plot(x, y, 'o', label="data")
plt.plot(x, model.fittedvalues, 'r--.', label="OLS")
plt.legend()
plt.show()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.804
Model:                            OLS   Adj. R-squared:                  0.794
Method:                 Least Squares   F-statistic:                     74.03
Date:                Wed, 08 May 2024   Prob (F-statistic):           8.56e-08
```

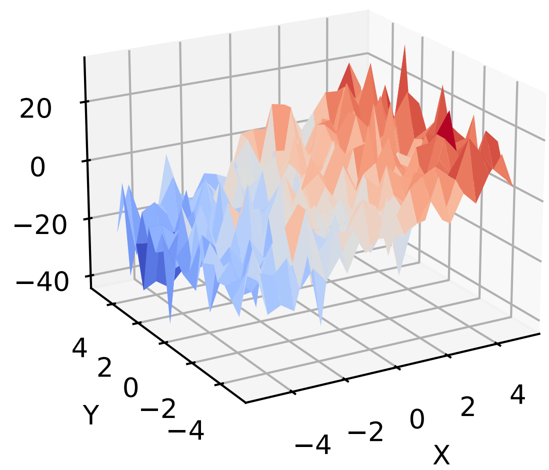
```
Time:                        11:41:01   Log-Likelihood:                -57.988
No. Observations:                  20   AIC:                             120.0
Df Residuals:                      18   BIC:                             122.0
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -5.5335      1.036     -5.342      0.000      -7.710      -3.357
x              2.9369      0.341      8.604      0.000       2.220       3.654
==============================================================================
Omnibus:                        0.100   Durbin-Watson:                   2.956
Prob(Omnibus):                  0.951   Jarque-Bera (JB):                0.322
Skew:                          -0.058   Prob(JB):                        0.851
Kurtosis:                       2.390   Cond. No.                         3.03
==============================================================================

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



8 D - 2: Multiple Regression: including multiple factors

```
"""
Multiple Regression
====================
Calculate using 'statsmodels' just the best fit, or all the corresponding
statistical parameters.
Also shows how to make 3d plots. Original author: Thomas Haslwanter
"""

from statsmodels.stats.anova import anova_lm
from statsmodels.formula.api import ols
import pandas
import matplotlib.pyplot as plt
import numpy as np

# For statistics. Requires statsmodels 5.0 or more

##############################################################################
# Generate and show the data
x = np.linspace(-5, 5, 21)
# We generate a 2D grid
X, Y = np.meshgrid(x, x)

# To get reproducable values, provide a seed value
np.random.seed(1)

# Z is the elevation of this 2D grid
Z = -5 + 3*X - 0.5*Y + 8 * np.random.normal(size=X.shape)

# Plot the data
# For 3d plots. This import is necessary to have 3D plotting below
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
surf = ax.plot_surface(X, Y, Z, cmap=plt.cm.coolwarm,
                       rstride=1, cstride=1)
ax.view_init(20, -120)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
```

```
################################################################################
# Multilinear regression model, calculating fit, P-values, confidence
# intervals etc.
# Convert the data into a Pandas DataFrame to use the formulas framework
# in statsmodels

# First we need to flatten the data: it's 2D layout is not relevant.
X = X.flatten()
Y = Y.flatten()
Z = Z.flatten()
data = pandas.DataFrame({'x': X, 'y': Y, 'z': Z})

# Fit the model
model = ols("z ~ x + y", data).fit()
plt.show()

# Print the summary
print(model.summary())
print("\nRetrieving manually the parameter estimates:")
print(model._results.params)

# Analysis of Variance (ANOVA) on linear models

# Peform analysis of variance on fitted linear model
anova_results = anova_lm(model)

print('\nANOVA results')
print(anova_results)
```



OLS Regression Results

Dep. Variable:	z	R-squared:	0.594
Model:	OLS	Adj. R-squared:	0.592
Method:	Least Squares	F-statistic:	320.4
Date:	Wed, 08 May 2024	Prob (F-statistic):	1.89e-86
Time:	11:41:01	Log-Likelihood:	-1537.7
No. Observations:	441	AIC:	3081.
Df Residuals:	438	BIC:	3094.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-4.5052	0.378	-11.924	0.000	-5.248	-3.763
x	3.1173	0.125	24.979	0.000	2.872	3.363
y	-0.5109	0.125	-4.094	0.000	-0.756	-0.266

Omnibus:	0.260	Durbin-Watson:	2.057
Prob(Omnibus):	0.878	Jarque-Bera (JB):	0.204
Skew:	-0.052	Prob(JB):	0.903
Kurtosis:	3.015	Cond. No.	3.03

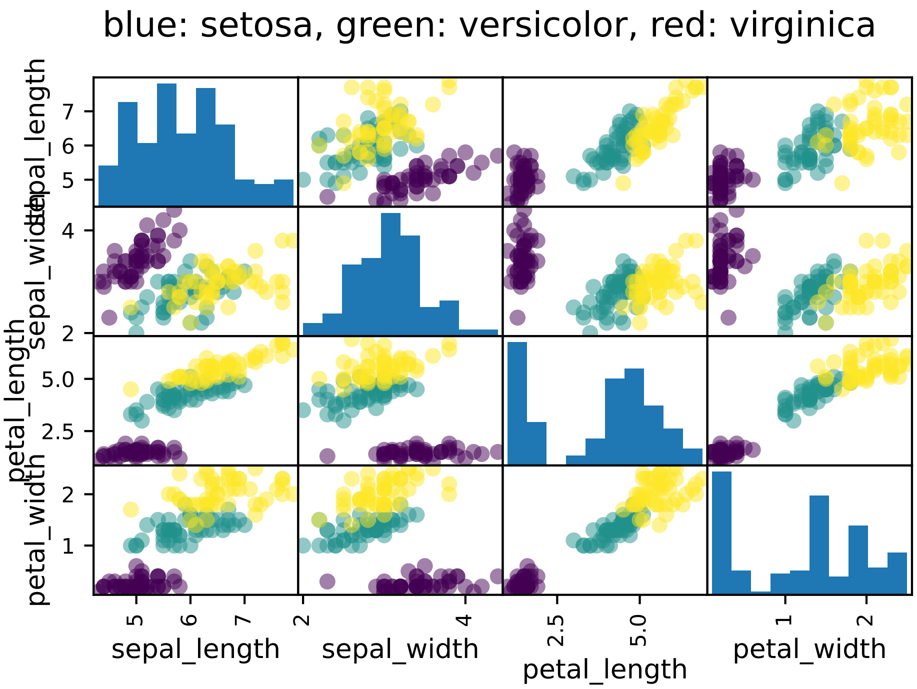
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
Retrieving manually the parameter estimates:
[-4.50523303  3.11734237 -0.51091248]

ANOVA results
             df        sum_sq       mean_sq           F        PR(>F)
x           1.0  39284.301219  39284.301219  623.962799  2.888238e-86
y           1.0   1055.220089   1055.220089   16.760336  5.050899e-05
Residual  438.0  27576.201607     62.959364         NaN           NaN
```

9 the iris data

```python
import matplotlib.pyplot as plt
import pandas
from pandas.plotting import scatter_matrix
from statsmodels.formula.api import ols

# Data
data = pandas.read_csv('iris.csv')

categories = pandas.Categorical(data['name'])

# The parameter 'c' is passed to plt.scatter and will control the color
scatter_matrix(data, c=categories.codes, marker='o')

fig = plt.gcf()
fig.suptitle("blue: setosa, green: versicolor, red: virginica", size=13)
```

```
Text(0.5, 0.98, 'blue: setosa, green: versicolor, red: virginica')
```




```
# Let us try to explain the sepal length as a function of the petal
# width and the category of iris

model = ols('sepal_width ~ name + petal_length', data).fit()
print(model.summary())

# Now formulate a "contrast", to test if the offset for versicolor and virginica are identica
print('Testing the difference between effect of versicolor and virginica')
print(model.f_test([0, 1, -1, 0]))
plt.show()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            sepal_width   R-squared:                       0.478
Model:                            OLS   Adj. R-squared:                  0.468
Method:                 Least Squares   F-statistic:                     44.63
Date:                Wed, 08 May 2024   Prob (F-statistic):           1.58e-20
Time:                        11:41:01   Log-Likelihood:                -38.185
No. Observations:                 150   AIC:                             84.37
Df Residuals:                     146   BIC:                             96.41
Df Model:                           3
```

```
Covariance Type:            nonrobust
=========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept                  2.9813      0.099     29.989      0.000       2.785       3.178
name[T.versicolor]        -1.4821      0.181     -8.190      0.000      -1.840      -1.124
name[T.virginica]         -1.6635      0.256     -6.502      0.000      -2.169      -1.158
petal_length               0.2983      0.061      4.920      0.000       0.178       0.418
==============================================================================
Omnibus:                        2.868   Durbin-Watson:                   1.753
Prob(Omnibus):                  0.238   Jarque-Bera (JB):                2.885
Skew:                          -0.082   Prob(JB):                        0.236
Kurtosis:                       3.659   Cond. No.                         54.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
Testing the difference between effect of versicolor and virginica
<F test: F=3.2453353465741515, p=0.07369058781700982, df_denom=146, df_num=1>
```