

```

# The Anscombe Quartet (Python)

# demonstration data from
# Anscombe, F. J. 1973, February. Graphs in statistical analysis.
# The American Statistician 27: 17-21.

# prepare for Python version 3x features and functions
from __future__ import division, print_function

# import packages for Anscombe Quartet demonstration
import pandas as pd # data frame operations
import numpy as np # arrays and math functions
import statsmodels.api as sm # statistical models (including regression)
import matplotlib.pyplot as plt # 2D plotting

# define the anscombe data frame using dictionary of equal-length lists
anscombe = pd.DataFrame({'x1' : [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
    'x2' : [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
    'x3' : [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
    'x4' : [8, 8, 8, 8, 8, 8, 8, 8, 19, 8, 8],
    'y1' : [8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68],
    'y2' : [9.14, 8.14, 8.74, 8.77, 9.26, 8.1, 6.13, 3.1, 9.13, 7.26, 4.74],
    'y3' : [7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73],
    'y4' : [6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.5, 5.56, 7.91, 6.89]})

# fit linear regression models by ordinary least squares
set_I_design_matrix = sm.add_constant(anscombe['x1'])
set_I_model = sm.OLS(anscombe['y1'], set_I_design_matrix)
print(set_I_model.fit().summary())

set_II_design_matrix = sm.add_constant(anscombe['x2'])
set_II_model = sm.OLS(anscombe['y2'], set_II_design_matrix)
print(set_II_model.fit().summary())

set_III_design_matrix = sm.add_constant(anscombe['x3'])
set_III_model = sm.OLS(anscombe['y3'], set_III_design_matrix)
print(set_III_model.fit().summary())

set_IV_design_matrix = sm.add_constant(anscombe['x4'])
set_IV_model = sm.OLS(anscombe['y4'], set_IV_design_matrix)
print(set_IV_model.fit().summary())

# create scatter plots
fig = plt.figure()
set_I = fig.add_subplot(2, 2, 1)
set_I.scatter(anscombe['x1'], anscombe['y1'])
set_I.set_title('Set I')
set_I.set_xlabel('x1')
set_I.set_ylabel('y1')
set_I.set_xlim(2, 20)
set_I.set_ylim(2, 14)

set_II = fig.add_subplot(2, 2, 2)
set_II.scatter(anscombe['x2'], anscombe['y2'])
set_II.set_title('Set II')
set_II.set_xlabel('x2')
set_II.set_ylabel('y2')
set_II.set_xlim(2, 20)
set_II.set_ylim(2, 14)

```

```

set_III = fig.add_subplot(2, 2, 3)
set_III.scatter(anscombe['x3'],anscombe['y3'])
set_III.set_title('Set III')
set_III.set_xlabel('x3')
set_III.set_ylabel('y3')
set_III.set_xlim(2, 20)
set_III.set_ylim(2, 14)

set_IV = fig.add_subplot(2, 2, 4)
set_IV.scatter(anscombe['x4'],anscombe['y4'])
set_IV.set_title('Set IV')
set_IV.set_xlabel('x4')
set_IV.set_ylabel('y4')
set_IV.set_xlim(2, 20)
set_IV.set_ylim(2, 14)

plt.subplots_adjust(left=0.1, right=0.925, top=0.925, bottom=0.1,
    wspace = 0.3, hspace = 0.4)
plt.savefig('fig_anscombe_Python.pdf', bbox_inches = 'tight', dpi=None,
    facecolor='w', edgecolor='b', orientation='portrait', papertype=None,
    format=None, transparent=True, pad_inches=0.25, frameon=None)

# Suggestions for the student:
# See if you can develop a quartet of your own,
# or perhaps just a duet, two very different data sets
# with the same fitted model.

```