Week 10

import numpy as np

import matplotlib.pyplot as plt

people = ('G1','G2','G3','G4','G5','G6','G7','G8')

segments = 4

data = [[ 3.40022085, 7.70632498, 6.4097905, 10.51648577, 7.5330039,

7.1123587, 12.77792868, 3.44773477],

[ 11.24811149, 5.03778215, 6.65808464, 12.32220677, 7.45964195,

6.79685302, 7.24578743, 3.69371847],

[ 3.94253354, 4.74763549, 11.73529246, 4.6465543, 12.9952182,

4.63832778, 11.16849999, 8.56883433],

[ 4.24409799, 12.71746612, 11.3772169, 9.00514257, 10.47084185,

10.97567589, 3.98287652, 8.80552122]]

percentages = (np.random.randint(5,20, (len(people), segments)))

y\_pos = np.arange(len(people))

fig = plt.figure(figsize=(10,8))

ax = fig.add\_subplot(111)

colors ='rgwm'

patch\_handles = []

left = np.zeros(len(people))

for i, d in enumerate(data):

patch\_handles.append(ax.barh(y\_pos, d,

color=colors[i%len(colors)], align='center',

left=left))

left += d

for j in range(len(patch\_handles)):

for i, patch in enumerate(patch\_handles[j].get\_children()):

bl = patch.get\_xy()

x = 0.5\*patch.get\_width() + bl[0]

y = 0.5\*patch.get\_height() + bl[1]

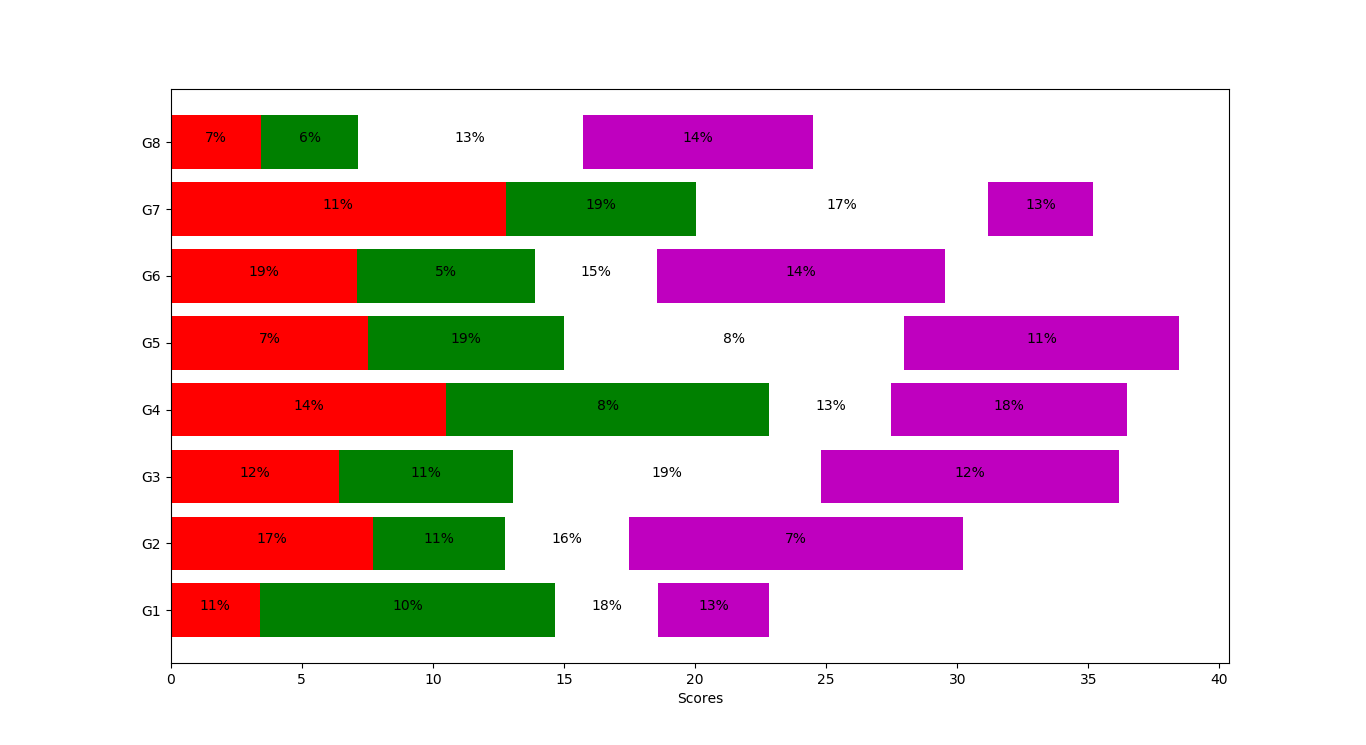
ax.text(x,y, "%d%%" % (percentages[i,j]), ha='center')

ax.set\_yticks(y\_pos)

ax.set\_yticklabels(people)

ax.set\_xlabel('Scores')

plt.show()



Week 8&9

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

iris=load\_iris()

print(iris['DESCR'])

iris.keys()

iris['target\_names']

iris['feature\_names']

iris['data'].shape

iris['data']

data1 = pd.DataFrame(data= np.c\_[iris['data'], iris['target']],columns= iris['feature\_names'] + ['target'])

data1['target']=pd.to\_numeric(data1['target'],downcast='integer')

data1

data1.dtypes

X\_train, X\_test, y\_train, y\_test = train\_test\_split(iris['data'], iris['target'], random\_state = 0)

print(X\_train.shape)

print(X\_test.shape)

fig, ax = plt.subplots(3, 3, figsize=(15,15))

for i in range(3):

for j in range(3):

ax[i,j].scatter(X\_train[:,j], X\_train[:, i + 1], c=y\_train, s=60)

ax[i,j].set\_xticks(())

ax[i,j].set\_yticks(())

if i == 2:

ax[i,j].set\_xlabel(iris['feature\_names'][j])

if j == 0:

ax[i,j].set\_ylabel(iris['feature\_names'][i + 1])

if j > i:

ax[i,j].set\_visible(False)

plt.show()

knn = KNeighborsClassifier(n\_neighbors = 1)

print(knn.fit(X\_train, y\_train))

X\_new = np.array([[5, 2.9, 1, 0.2]])

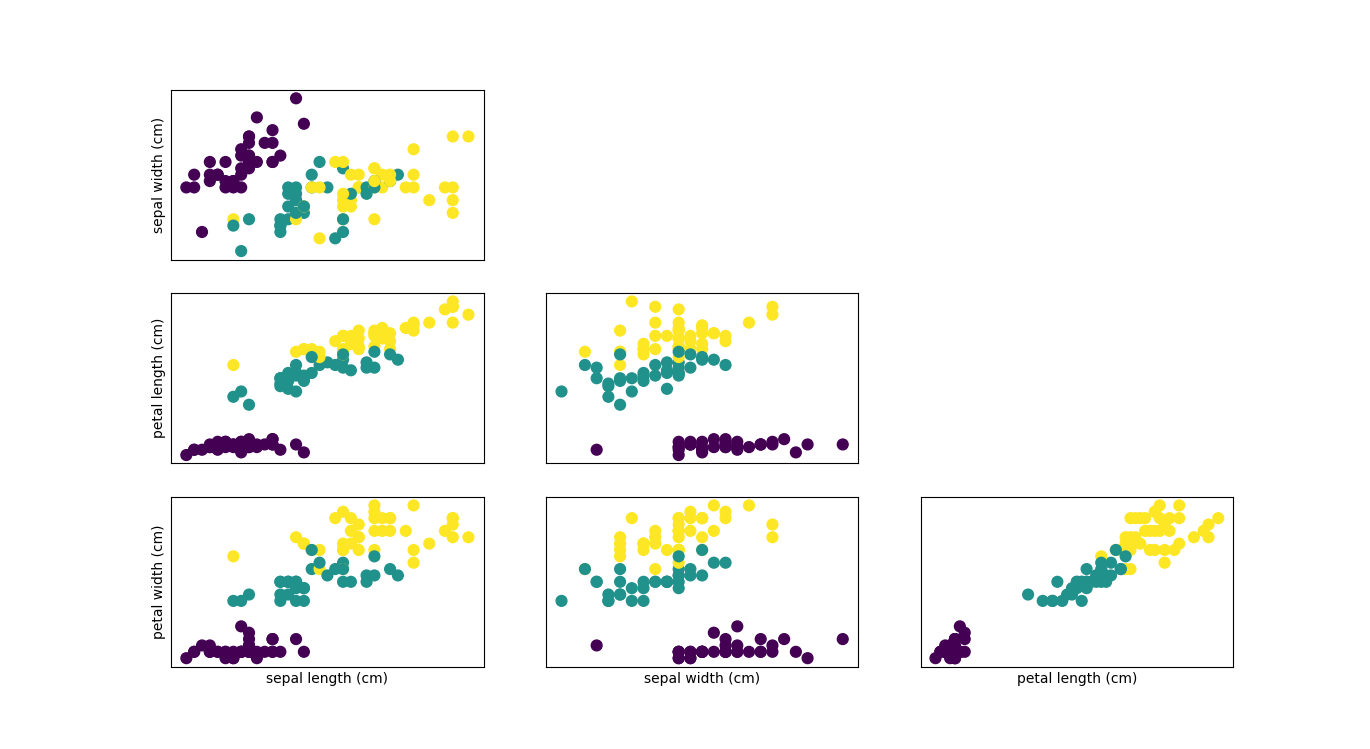
print(X\_new.shape)

prediction = knn.predict(X\_new)

print(prediction)

print(iris['target\_names'][prediction])

print(knn.score(X\_test, y\_test))



Week 5

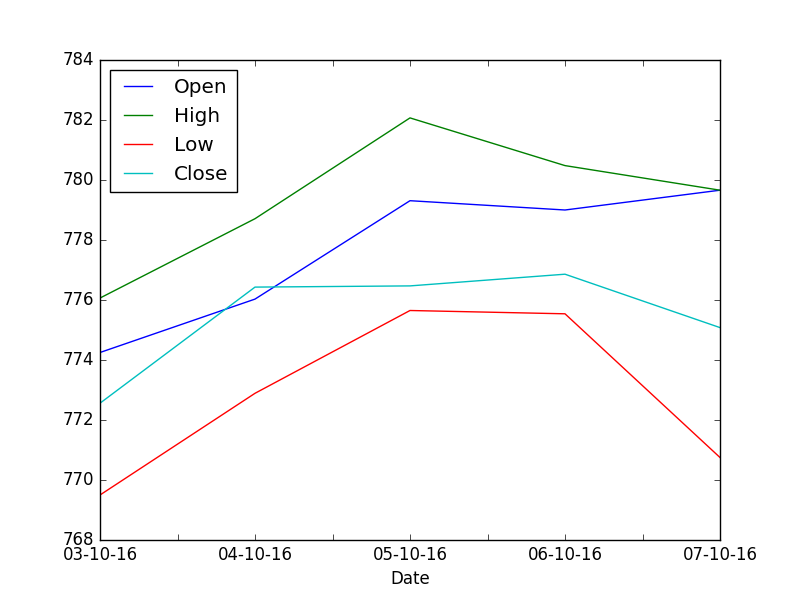
import matplotlib.pyplot as plt

import pandas as pd

df = pd.read\_csv('fdata.csv', index\_col=0)

df.plot(x=df.index, y=df.columns)

plt.show()



WEEK 4:

Write a Python program to create bar plots with errorbars on the same figure. Attach a text label above each bar displaying men means (integer value).

Sample Date

Mean velocity: 0.2474, 0.1235, 0.1737, 0.1824

Standard deviation of velocity: 0.3314, 0.2278, 0.2836, 0.2645

SOURCE CODE:

import numpy as np

import matplotlib.pyplot as plt

import matplotlib.patches as mpatches

N = 5

men\_means = (54.74, 42.35, 67.37, 58.24, 30.25)

men\_std= (4, 3, 4, 1, 5)

ind = np.arange(N) # the x locations for the groups

width = 0.35 # the width of the bars

fig, ax = plt.subplots()

rects1 = ax.bar(ind, men\_means, width, color='r', yerr=men\_std)

# add some text for labels, title and axes ticks

plt.ylabel('Scores')

plt.xlabel('Velocity')

plt.title('Scores by Velocity')

red\_patch = mpatches.Patch(color='red', label='Men')

plt.legend(handles=[red\_patch])

def autolabel(rects):

"""

Attach a text label above each bar displaying its height

"""

for rect in rects:

height = rect.get\_height()

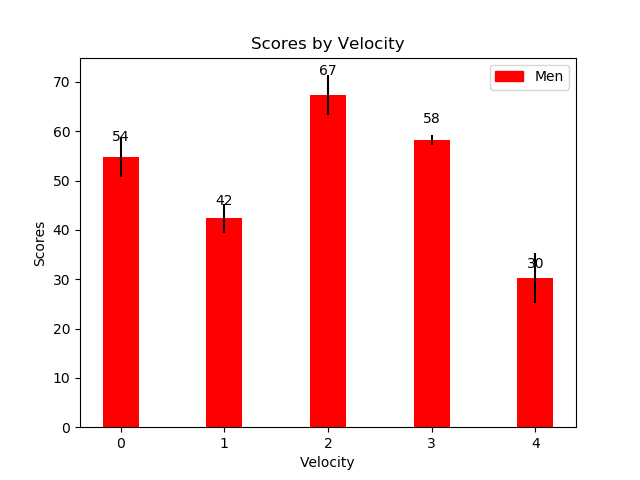
ax.text(rect.get\_x() + rect.get\_width()/2., 1.05\*height,

'%d' % int(height),

ha='center', va='bottom')

autolabel(rects1)

plt.show()



Program 1a

import csv

with open('C:/Users/cvr/Desktop/datascience.csv', newline='') as File:

reader = csv.reader(File)

for row in reader:

print(row)

outtput

['694', '192']

['405', '215']

['802', '215']

['1366', '274']

['963', '185']

['821', '212']

['714', '222']

['101.8', '276']

['887', '260']

['790', '221.5']

['696', '255']

['771', '260']

['1006', '293']

['1191', '375']

Program1b

import matplotlib.pyplot as plt

import pandas as pd

data = pd.read\_csv('C:/Users/cvr/Desktop/datascience.csv', sep=',',header=None, index\_col =0)

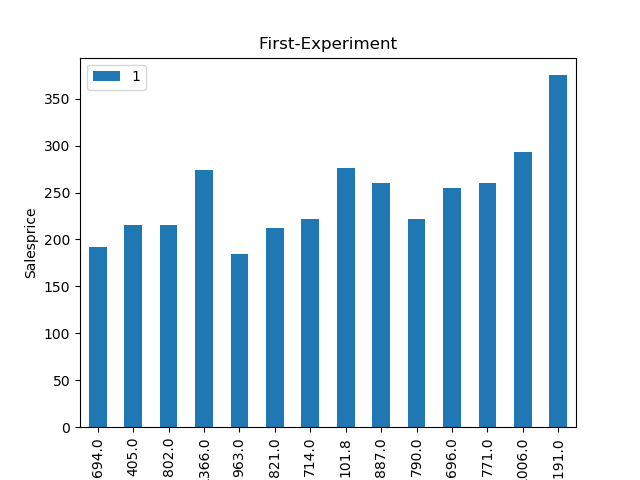
data.plot(kind='bar')

plt.ylabel('Salesprice')

plt.xlabel('area')

plt.title('First-Experiment')

plt.show()



import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import csv

in\_array1 = [694, 405, 802, 1366, 716, 963, 963, 821, 714, 101.8, 887, 790, 696, 771, 1006, 1191]

in\_array2 = [192, 215, 215, 274, 112.7, 185.0, 212.0, 222, 276, 260.0, 221.5, 225.0, 260, 293, 375, 222]

out\_array1 = np.log(in\_array1)

list1 = list(out\_array1)

out\_array2 = np.log(in\_array2)

list2 = list(out\_array2)

dat = np.array([list1, list2])

dat = dat.T

np.savetxt('C:/Users/CVR/Desktop/text2.txt', dat, delimiter = ',')

data = pd.read\_csv('C:/Users/CVR/Desktop/text2.txt',sep=',')

print(data)

data.plot(kind = 'bar')

plt.xlabel('salesprice')

plt.ylabel('area')

plt.title('First Program')

plt.show()

#for ggplot

plt.style.use('ggplot')

data.plot()

plt.show()

output

6.542471960506804685e+00 5.257495372027781499e+00

0 6.003887 5.370638

1 6.687109 5.370638

2 7.219642 5.613128

3 6.573680 4.724729

4 6.870053 5.220356

5 6.870053 5.356586

6 6.710523 5.402677

7 6.570883 5.620401

8 4.623010 5.560682

9 6.787845 5.400423

10 6.672033 5.416100

11 6.545350 5.560682

12 6.647688 5.680173

13 6.913737 5.926926

14 7.082549 5.402677

