Data Visualization
Part II

Plotting with matplotlib
and pandas

First, Read Data from CSV file

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib

sales = pd.read_csv("sample-salesv2.csv",
parse_dates=['date'])
sales.head()
sales.dtypes

sales.describe()

sales['unit price'].describe()
```

Customers

```
customers = sales[['name','ext price','date']]
customers.head()

customer_group = customers.groupby('name')
customer_group.size()

sales_totals = customer_group.sum()
sales_totals.sort_values('ext price').head()

my_plot = sales_totals.plot(kind='bar')
my_plot = sales_totals.plot(kind='barh')

# identical
my_plot = sales_totals.plot.bar()
```

Customers – Title and Labels

```
my_plot = sales_totals.sort_values('ext price',
ascending=False).plot(kind='bar', legend=None,
title="Total Sales by Customer")


my_plot.set_xlabel("Customers")


my_plot.set_ylabel("Sales ($)")
```

Customers with Product Category

```
customers = sales[['name', 'category', 'ext price',
'date']]
customers.head()
category_group =
customers.groupby(['name','category']).sum()
category_group.head(10)
category_group = category_group.unstack()
category_group.head(10)
my_plot = category_group.plot(kind='bar', stacked=True,
title="Total Sales by Customer")
my_plot.set_xlabel("Customers")
my_plot.set_ylabel("Sales ($)")
my_plot.legend(["Belts","Shirts","Shoes"], loc='best',
ncol=3)
```

Customers with Product Category – Sorted!

```
category_group = category_group.sort_values(('ext price', 'Belt'), ascending=False)
category_group.head()
my_plot = category_group.plot(kind='bar', stacked=True, title="Total Sales by Customer")


# sort by total without showing total!
category_group['total'] = category_group.sum(axis=1)
category_group = category_group.sort_values('total', ascending=False)
category_group.head()
category_group.drop('total', axis=1, inplace=True)
my_plot = category_group.plot(kind='bar', stacked=True, title="Total Sales by Customer")
```

Purchase Patterns

```
purchase_patterns = sales[['ext price','date']]
purchase_patterns.head()


purchase_plot = purchase_patterns['ext
price'].hist(bins=20)


# done many times now,
# but should always be done to make figure self-
explanatory
purchase_plot.set_title("Purchase Patterns")
purchase_plot.set_xlabel("Order Amount ($)")
purchase_plot.set_ylabel("Number of Orders")
```

Purchase Patterns – Timeline

```
purchase_patterns = purchase_patterns.set_index('date')
purchase_patterns.head()

# sorted by time
purchase_patterns.sort_index()

# resampled by months
purchase_plot =
purchase_patterns.resample('M').sum().plot(title="Total
Sales by Month", legend=None)

# save the figure
fig = purchase_plot.get_figure()
fig.savefig("total-sales.png")
```

Boxplot and Histogram

```
# Box and Whisker Plots
sales.boxplot()    # Not very useful!
sales.plot(kind='box', subplots=True, layout=(2,2),
sharex=False, sharey=False)

sales.boxplot(column="ext price", by="name")

# Histograms
sales.hist()
sales.plot(kind='hist', subplots=True, layout=(2,2),
sharex=False, sharey=False)  # "ignored", unfortunately

sales.hist(column="ext price", by="name", bins=30)
sales.hist(column="ext price", by="name", bins=30,
sharex=True, sharey=True)
```

First, Read Data from CSV file

```
import numpy as np
import matplotlib.pyplot as plt

# Load CSV using pandas
import pandas as pd
from pandas import read_csv
# AirBnB website visitors
filename = 'visitors.csv'
visitors = read_csv(filename, index_col='id_visitor')
print(visitors.head())

print(visitors.shape)
print(visitors.head())
print(visitors.dtypes)
```

Histograms, Density Plots, Box and Whisker Plots

```
# Univariate Histograms
visitors.hist()

# Univariate Density Plots
visitors.plot(kind='density', subplots=True,
layout=(2,2), sharex=False)

# Box and Whisker Plots
visitors.plot(kind='box', subplots=True, layout=(2,2),
sharex=False, sharey=False)
```

Correlation Matrix Plot

```
# correlation matrix
correlations = visitors.corr()
# plot correlation matrix (generic)
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(correlations, vmin=-1, vmax=1)
fig.colorbar(cax)

# change the tick labels
ticks = np.arange(0,4,1)
ax.set_xticks(ticks)
ax.set_yticks(ticks)
ax.set_xticklabels(visitors.columns)
ax.set_yticklabels(visitors.columns)
```

Scatter Plot Matrix

```
# Scatterplot Matrix
from pandas.plotting import scatter_matrix
scatter_matrix(visitors)
```

Additional Readings

- Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython by Wes McKinney (pub. yr. 2017). Chapter 9 and 10.
- Machine Learning Mastery with Python by Jason Brownlee (pub. yr. 2017). Chapter 6.
- https://github.com/chris1610/pbpython/blob/master/notebooks/Simple_Graphing.ipynb
- http://pbpython.com/simple-graphing-pandas.html
- https://www.analyticsvidhya.com/blog/2016/01/12-pandas-techniques-python-data-manipulation/

Additional Readings (cont'd)

- https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html
- https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.plot.html
- http://pandas.pydata.org/pandas-docs/version/0.20.3/generated/pandas.DataFrame.boxplot.html
- http://pandas.pydata.org/pandas-docs/version/0.20.3/generated/pandas.DataFrame.hist.html
- https://matplotlib.org/

- DataCamp:
  - Course: Intermediate Python for Data Science
    - » Chapter: Matplotlib
  - Introduction to Data Visualization with Python
    - » Chapter: Customizing Plots