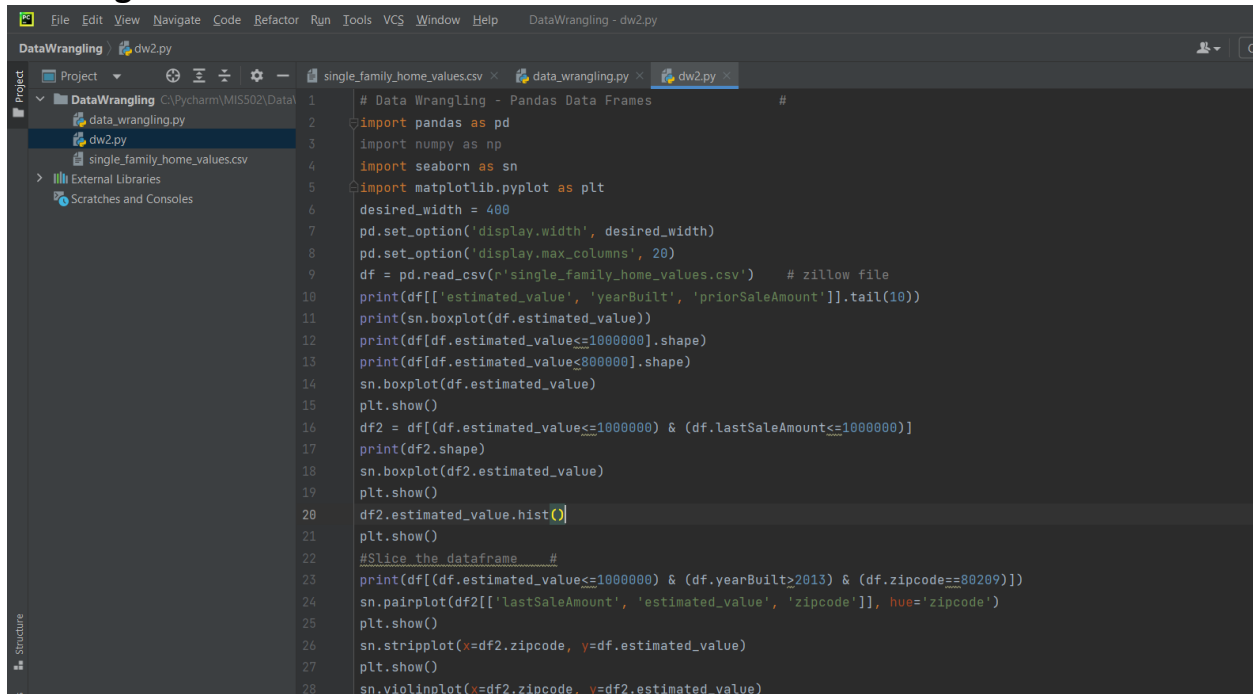


Tutorial 7

Reading a CSV file into a Pandas Data Frame:



```
1 # Data Wrangling - Pandas Data Frames #
2 import pandas as pd
3 import numpy as np
4 import seaborn as sn
5 import matplotlib.pyplot as plt
6 desired_width = 400
7 pd.set_option('display.width', desired_width)
8 pd.set_option('display.max_columns', 20)
9 df = pd.read_csv(r'single_family_home_values.csv') # zillow file
10 print(df[['estimated_value', 'yearBuilt', 'priorSaleAmount']].tail(10))
11 print(sn.boxplot(df.estimated_value))
12 print(df[df.estimated_value<=1000000].shape)
13 print(df[df.estimated_value<=800000].shape)
14 sn.boxplot(df.estimated_value)
15 plt.show()
16 df2 = df[(df.estimated_value<=1000000) & (df.lastSaleAmount<=1000000)]
17 print(df2.shape)
18 sn.boxplot(df2.estimated_value)
19 plt.show()
20 df2.estimated_value.hist()
21 plt.show()
22 #Slice the dataframe #
23 print(df[(df.estimated_value<=1000000) & (df.yearBuilt>2013) & (df.zipcode==80209)])
24 sn.pairplot(df2[['lastSaleAmount', 'estimated_value', 'zipcode']], hue='zipcode')
25 plt.show()
26 sn.stripplot(x=df2.zipcode, y=df2.estimated_value)
27 plt.show()
28 sn.violinplot(x=df2.zipcode, y=df2.estimated_value)
```

```
# Data Wrangling - Pandas DataFrame
import pandas as pd
import seaborn as sn
import numpy as np
import matplotlib.pyplot as plt

desired_width = 320
pd.set_option('display.width', desired_width)
pd.set_option('display.max_columns', 18)
df = pd.read_csv(r'single_family_home_values.csv') # zillow file
print(df.head(2))
print(df.tail(3))
print(type(df))
print(df.shape)
print(df.info())
```

#	Column	Non-Null Count	Dtype
0	id	15000 non-null	int64
1	address	15000 non-null	object
2	city	15000 non-null	object
3	state	15000 non-null	object
4	zipcode	15000 non-null	int64
5	latitude	14985 non-null	float64
6	longitude	14985 non-null	float64
7	bedrooms	15000 non-null	int64
8	bathrooms	15000 non-null	float64
9	rooms	15000 non-null	int64
10	squareFootage	15000 non-null	int64
11	lotSize	15000 non-null	int64

Df.describe()

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

desired_width = 320
pd.set_option('display.width', desired_width)
pd.set_option('display.max_columns', 18)
df = pd.read_csv(r'single_family_home_values.csv') # zillow file
print(df.head(2))
print(df.tail(3))
print(type(df))
print(df.shape)
print(df.info())
print(df.describe())
df = df.fillna(df.mean())
```

	id	zipcode	latitude	longitude	bedrooms	bathrooms	rooms	squareFootage	lotSize	yearBuilt	lastSaleAmount	priorSaleAmount
count	15000.00000	15000.00000	14985.00000	14985.00000	15000.00000	15000.00000	15000.00000	15000.00000	15000.00000	14999.00000	1.500000e+04	1.128700e+0
mean	5.176229e+07	80204.919467	39.740538	-104.964076	2.708400	2.195067	6.164133	1514.504400	5820.76620	1929.517168	4.053563e+05	2.594350e+0
std	6.190876e+07	9.715263	0.023555	0.039788	0.897231	1.166279	1.958601	830.635999	3013.27947	29.937051	7.756998e+05	3.379387e+0
min	1.433670e+05	80022.000000	39.614531	-105.108440	0.000000	0.000000	0.000000	350.000000	278.00000	1874.00000	2.590000e+02	0.000000e+0
25%	1.004802e+07	80205.000000	39.727634	-104.978737	2.000000	1.000000	5.000000	986.000000	4620.00000	1907.000000	1.940000e+05	1.100000e+0
50%	2.563241e+07	80206.000000	39.748048	-104.957689	3.000000	2.000000	6.000000	1267.500000	5950.00000	1925.000000	3.200000e+05	2.100000e+0
75%	5.114222e+07	80207.000000	39.758214	-104.937522	3.000000	3.000000	7.000000	1766.250000	6270.00000	1949.000000	4.632000e+05	3.302400e+0
max	3.209481e+08	80209.000000	39.888020	-104.830930	15.000000	12.000000	39.000000	10907.000000	122839.00000	2016.000000	4.560000e+07	1.600000e+0

Process finished with exit code 0

Median

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 desired_width = 320
4 pd.set_option('display.width', desired_width)
5 pd.set_option('display.max_columns', 18)
6 df = pd.read_csv(r'single_family_home_values.csv') # zillow file
7 print(df.head(2))
8 print(df.tail(3))
9 print(type(df))
10 print(df.shape)
11 print(df.info())
12 print(df.describe())
13 print(df.priorSaleAmount.median())
14
15
16
17
18
19

```

Run: data_wrangling.py

	id	zipcode	latitude	longitude	bedrooms	bathrooms	rooms	squareFootage	lotSize	yearBuilt	lastSaleAmount	priorSaleAmount
count	1.500000e+04	15000.000000	14985.000000	14985.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	14999.000000	1.500000e+04	1.128700e+04
mean	5.176229e+07	80204.919467	39.740538	-104.964076	2.708400	2.195067	6.164133	1514.504400	5820.76620	1929.517168	4.053563e+05	2.594350e+05
std	6.190876e+07	9.715263	0.023555	0.039788	0.897231	1.166279	1.958601	830.635999	3013.27947	29.937051	7.756998e+05	3.379380e+05
min	1.433670e+05	80022.000000	39.614531	-105.108440	0.000000	0.000000	0.000000	350.000000	278.000000	1874.000000	2.590000e+02	0.000000
25%	1.084802e+07	80205.000000	39.727634	-104.978737	2.000000	1.000000	5.000000	986.000000	4620.000000	1907.000000	1.940000e+05	1.100000e+05
50%	2.563241e+07	80206.000000	39.748048	-104.957689	3.000000	2.000000	6.000000	1267.500000	5950.000000	1925.000000	3.209000e+05	2.100000e+05
75%	5.114222e+07	80207.000000	39.758214	-104.937522	3.000000	3.000000	7.000000	1766.250000	6270.000000	1949.000000	4.632000e+05	3.302400e+05
max	3.209481e+08	80209.000000	39.888020	-104.830930	15.000000	12.000000	39.000000	10907.000000	122839.000000	2016.000000	4.560000e+07	1.600000e+07
210000.0												

Process finished with exit code 0

Dealing with missing data and outliers:

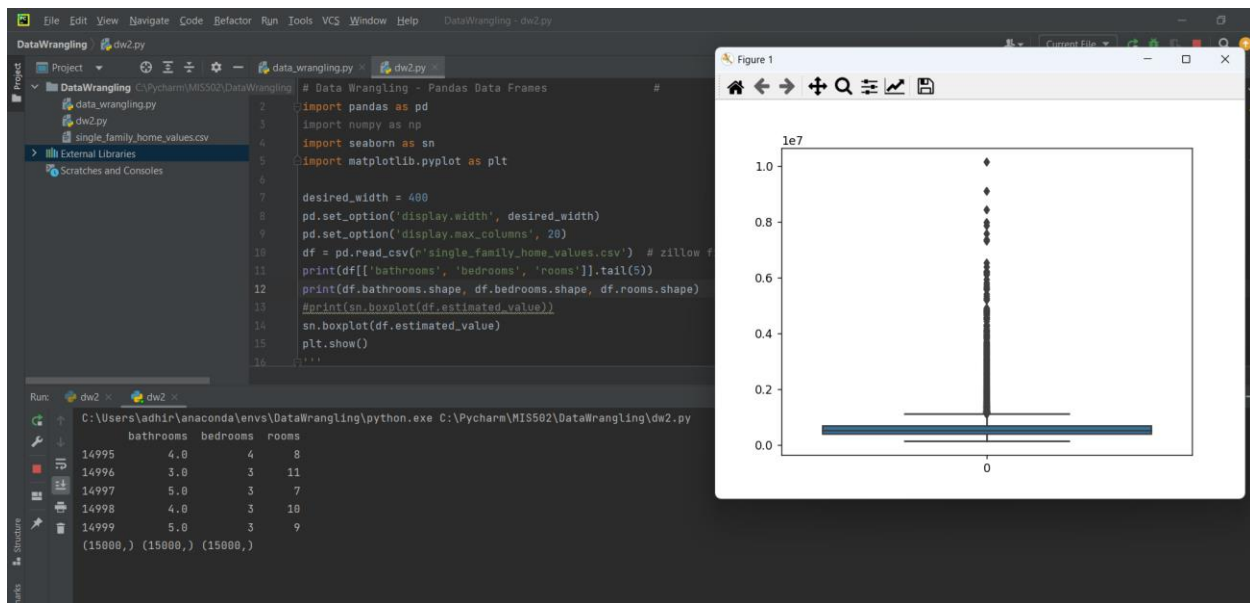
```

1 import pandas as pd
2 import numpy as np
3 import seaborn as sn
4 import matplotlib.pyplot as plt
5
6
7 desired_width = 400
8 pd.set_option('display.width', desired_width)
9 pd.set_option('display.max_columns', 20)
10 df = pd.read_csv(r'single_family_home_values.csv') # zillow file
11 print(df[['bathrooms', 'bedrooms', 'rooms']].tail(5))
12 print(df.bathrooms.shape, df.bedrooms.shape, df.rooms.shape)
13 print(sn.boxplot(df.estimated_value))
14 sn.boxplot(df.estimated_value)
15 plt.show()
16
17 df2 = df[(df.estimated_value <= 1000000) & (df.lastSaleAmount <= 1000000)]

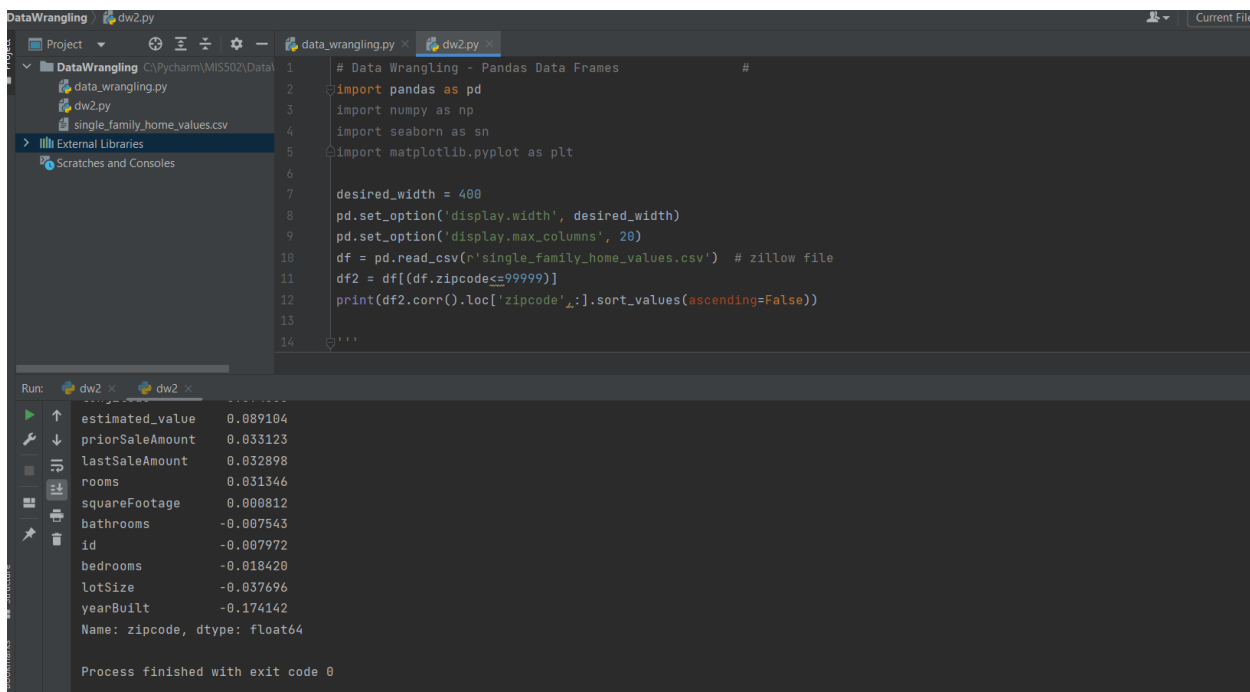
```

Run: dw2.py

	bathrooms	bedrooms	rooms
14995	4.0	4	8
14996	3.0	3	11
14997	5.0	3	7
14998	4.0	3	10
14999	5.0	3	9



Simple Statistics in Python:



Slicing a DataFrame:

```
DataWrangling - dw2.py
# Data Wrangling - Pandas Data Frames
1
2
3 import pandas as pd
4 import numpy as np
5 import seaborn as sn
6 import matplotlib.pyplot as plt
7 desired_width = 400
8 pd.set_option('display.width', desired_width)
9 pd.set_option('display.max_columns', 20)
10 df = pd.read_csv(r'single_family_home_values.csv')
11 print(df.zipcode.unique())
12 print(df[(df.estimated_value<=1000000) & (df.yearBuilt>2013) & (df.zipcode==80209)])
13
14 import pandas as pd
```

Run: dw2

	id	address	city	state	zipcode	latitude	longitude	bedrooms	bathrooms	rooms	squareFootage	lotSize	yearBuilt	lastSaleDate	lastSaleAmount
[80022 80033 80123 80202 80203 80204 80205 80206 80207 80209]															
13229	3455195	866 S York St	Denver	CO	80209	39.700791	-104.960246	3	2.0	8	3332	6250	2015.0	2014-04-09	560000
13276	39512040	764 S York St	Denver	CO	80209	39.702607	-104.960243	3	4.0	7	2567	6250	2016.0	2012-07-23	420000
13316	11586398	450 S Vine St	Denver	CO	80209	39.708351	-104.962546	3	5.0	7	2570	4680	2016.0	2015-02-11	571000
13428	11586742	636 S Williams St	Denver	CO	80209	39.704969	-104.966025	3	5.0	6	2578	4680	2016.0	2015-07-22	637600
13431	184305844	456 S High St	Denver	CO	80209	39.708197	-104.964857	3	5.0	10	3346	6240	2016.0	2016-01-11	725000
13766	7652681	408 S Franklin St	Denver	CO	80209	39.709074	-104.968369	3	5.0	8	3367	5060	2015.0	2014-04-21	513486
13812	39708952	611 S Washington St	Denver	CO	80209	39.705403	-104.979580	3	4.0	10	2090	4690	2014.0	2015-03-09	825000
13948	184305843	450 S High St	Denver	CO	80209	39.708333	-104.964858	3	5.0	7	3124	6240	2016.0	2014-02-20	607000
14162	30566405	876 S Williams St	Denver	CO	80209	39.708646	-104.966016	3	1.0	8	3394	6250	2015.0	2014-03-18	675000
14982	43208991	731 S Elizabeth St	Denver	CO	80209	39.703135	-104.956141	2	3.0	6	2680	6160	2016.0	2015-03-31	485000

Groupby merge etc.

```
DataWrangling - dw2.py
17 pd.set_option('display.width', desired_width)
18 pd.set_option('display.max_columns', 20)
19 df = pd.read_csv(r'single_family_home_values.csv') # zillow file
20 df['lastSaleDate2'] = pd.to_datetime(df.lastSaleDate)
21 print(df.info()) # returns df columns
22 print(df.lastSaleDate2.dt.year.head(2)) # returns lastSaleDate2 as year format
23 print(df.lastSaleDate2.dt.month.head(2)) # returns lastSaleDate2 as month format
24 print(df.lastSaleDate2.dt.week.head(2)) # returns lastSaleDate2 as week format
25 print(df.lastSaleDate2.dt.day.head(2)) # returns lastSaleDate2 as day format
26 print(df.lastSaleDate2.dt.dayofweek.head(2)) # returns lastSaleDate2 as dayofweek format
27
28
29 df3 = df.groupby('zipcode').estimated_value.median().reset_index() # reset_index() shifts pd series to df
```

Run: dw2

	Data	columns (total 19 columns):
#	Column	Non-Null Count Dtype
0	id	15000 non-null int64
1	address	15000 non-null object
2	city	15000 non-null object
3	state	15000 non-null object
4	zipcode	15000 non-null int64
5	latitude	14985 non-null float64
6	longitude	14985 non-null float64
7	bedrooms	15000 non-null int64
8	bathrooms	15000 non-null float64
9	rooms	15000 non-null int64
10	squareFootage	15000 non-null int64
11	lotSize	15000 non-null int64

```
dtypes: datetime64[ns](1), float64(5), int64(8), object(5)
```

```
memory usage: 2.2+ MB
```

```
None
```

```
0    2009
```

```
1    2004
```

```
Name: lastSaleDate2, dtype: int64
```

```
0    12
```

```
1     9
```

```
Name: lastSaleDate2, dtype: int64
```

```
0    51
```

```
1    39
```

```
Name: lastSaleDate2, dtype: int64
```

```
0    17
```

```
1    23
```

```
Name: lastSaleDate2, dtype: int64
```

```
0     3
```

```
1     3
```

```
Name: lastSaleDate2, dtype: int64
```