Part 1: Loading and Inspecting our data

Before we begin inspecting our data we will need to load it by using pandas built in function read_csv. This will read our CSV file and load our data in a pandas DataFrame Object.

Pandas

Pandas is a flexible python library we will be using to create our DataFrame for data manipulation and analysis.

Matplotlib/Pyplot

Matplotlib is a plotting python 2D library. Pyplot is a command style function that makes Matplotlib work like MATLAB. Simpler version of Seaborn.

Seaborn

Seaborn is a python library used for visualization, a more enhanced version of Matplotlib.

```
# Import the new library libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# This line is used to make plots show inline
%matplotlib inline
```

```
# This retrieves our data simply read it as a csv file and assign it to a respective variable
df = pd.read_csv('Patient2Treatment-Outcome2Drug-Mock2-Complete.csv')
```

Health Care Data Analysis

We will be exploring an oncology treatment data set which contains information about artificial patients.

```
# Allows us to look at the first 5 rows of our DataFrame
df.head()
```

	PatientID	TreatmentStart	Drug	Dosage
0	PT0	2014-Chapter	14D	20
1	PT0	12/15 Chapter	14D	20
2	PT1	12/15 Chapter	14D	20
3	PT1	12/15 Chapter	14D	20
4	PT1	2014-Chapter	14D	20
5	PT1	2014-Chapter	20D	20

```
# Gives us some some descriptive statistics on our dataframe
df.describe()
```

	Dosage
count	24.000000
mean	20.000000
std	10.454545
min	14.000000
25%	14.000000
50%	20.000000
75%	20.000000
max	100.000000

Note: The first thing I noticed after observing basic statistics is the high number of low Dosage. Compared to the other values, low Dosage is significantly higher. This may indicate a typo.

```
# Returns the data types of our columns
df.dtypes
```

```
PatientID      object
TreatmentStart object
Drug           object
Dosage         int64
dtype: object
```

```
# Returns the number of rows of our data set
len(df)
```

```
24
```

Accessing columns in our dataframe

```
df.columns
```

```
Index(['PatientID', 'TreatmentStart', 'Drug', 'Dosage'], dtype='object')
```

```
# Output each row in our PatientID column in tabular form. Notice the low brackets which gives it a DataFrame format
df[['PatientID']].head()
```

	PatientID
0	PT0
1	PT0
2	PT1
3	PT1
4	PT1

```
df[['TreatmentStart']].head()
```

	TreatmentStart
0	2014-
1	12/15
2	12/15
3	2014-
4	2014-

```
df[['Drug']].head()
```

	Drug
0	14D
1	14D
2	14D
3	14D
4	20D

```
df[['Dosage']].head()
```

	Dosage
0	20
1	20
2	20
3	20
4	20

```
# Returns the unique values for patientID in Series form.
df['PatientID'].unique()
```

```
array(['PT0', 'PT1', 'PT2', 'PT3', 'PT4', 'PT5', 'PT6', 'PT7',
       'PT8', 'PT9', 'PT10', 'PT11', 'PT12', 'PT13', 'PT14',
       'PT15', 'PT16', 'PT17', 'PT18', 'PT19', 'PT20'],
      dtype='object')
```

```
# Returns the unique drugs in our DataFrame in Series form
df['Drug'].unique()
```

```
array(['14D', '20D', '20D', '20D', '20D', '20D', '20D', '20D',
       '20D', '20D', '20D', '20D', '20D', '20D', '20D', '20D',
       '20D', '20D', '20D', '20D', '20D', '20D'],
      dtype='object')
```

```
# Returns the first row in our DataFrame
df.iloc[0]
```

PatientID	PT0
TreatmentStart	2014-Chapter 14D
Drug	14D
Dosage	20

```
# Returns a row in our DataFrame excluding the first row
df.iloc[1:]
```

	PatientID	TreatmentStart	Drug	Dosage
1	PT0	12/15 Chapter	14D	20
2	PT1	12/15 Chapter	14D	20
3	PT1	12/15 Chapter	14D	20
4	PT1	2014-Chapter	14D	20
5	PT1	2014-Chapter	20D	20

```
# Returns the last row in our DataFrame
df.iloc[-1:]
```

	PatientID	TreatmentStart	Drug	Dosage
23	PT19	2015-Chapter	20D	20

```
# Returns all rows where PatientID is PT19 in the Drug in tiraglatin
df[df['PatientID'] == 'PT19']
df[df['Drug'] == 'tiraglatin']
```

	PatientID	TreatmentStart	Drug	Dosage
19	PT19	2015-Chapter	14D	20
20	PT19	2015-Chapter	20D	20
21	PT19	2015-Chapter	20D	20
22	PT19	2015-Chapter	20D	20
23	PT19	2015-Chapter	20D	20

```
# Returns the unique values for PatientID in Series form
df['PatientID'].unique()
```

```
array(['PT0', 'PT1', 'PT2', 'PT3', 'PT4', 'PT5', 'PT6', 'PT7',
       'PT8', 'PT9', 'PT10', 'PT11', 'PT12', 'PT13', 'PT14',
       'PT15', 'PT16', 'PT17', 'PT18', 'PT19', 'PT20'],
      dtype='object')
```

Part 2: Data Cleaning

Lets clean up our Dosage Column by updating a row

```
# The row that had an error in it in our Dosage column, we have a max value of 1000 which is way above our normal value
# With pandas we can use the loc function to update the row
df.loc[df['Dosage'] == 1000, 'Dosage'] = 20
```

```
/Applications/Anaconda3/lib/python3.6/site-packages/pandas/core/indexing.py:179: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_with_indexer(indexer, value)
```

```
# Update our row with updated Dosage column
df
```

	PatientID	TreatmentStart	Drug	Dosage
0	PT0	2014-Chapter	14D	20
1	PT0	12/15 Chapter	14D	20
2	PT1	12/15 Chapter	14D	20
3	PT1	2014-Chapter	14D	20
4	PT1	2014-Chapter	20D	20
5	PT2	2014-Chapter	14D	20
6	PT2	2014-Chapter	14D	20
7	PT2	2014-Chapter	14D	20
8	PT2	2014-Chapter	20D	20
9	PT3	2014-Chapter	14D	20
10	PT3	2014-Chapter	14D	20
11	PT3	2014-Chapter	20D	20
12	PT4	2014-Chapter	14D	20
13	PT4	2014-Chapter	14D	20
14	PT4	2014-Chapter	20D	20
15	PT5	2014-Chapter	14D	20
16	PT5	2014-Chapter	14D	20
17	PT5	2014-Chapter	20D	20
18	PT6	2014-Chapter	14D	20
19	PT6	2014-Chapter	14D	20
20	PT6	2014-Chapter	20D	20
21	PT6	2014-Chapter	20D	20
22	PT6	2014-Chapter	20D	20
23	PT6	2014-Chapter	20D	20

```
# Lets delete the row that contains
df.drop(19, inplace=True)
```

Note: Notice that the row index is no longer in order.

Lets clean up our PatientStart column

```
# Our first task is convert the TreatmentStart column to a datetime object then assign it to our data column
df['TreatmentStart'] = pd.to_datetime(df['TreatmentStart'])
```

```
# Updated our new data column
df.head()
```

	PatientID	TreatmentStart	Drug	Dosage
0	PT0	2014-01-01	Chapter 14D	20
1	PT0	2015-12-01	Chapter 14D	20
2	PT1	2015-12-01	Chapter 14D	20
3	PT1	2014-01-01	Chapter 14D	20
4	PT1	2014-01-01	Chapter 20D	20

```
# Again we have updated our column to our TreatmentStart column
df.sort_values('TreatmentStart', inplace=True)
```

```
# Updated our new data column
df.head()
```

	PatientID	TreatmentStart	Drug	Dosage
0	PT0	2014-01-01	Chapter 14D	20
3	PT1	2014-01-01	Chapter 14D	20
4	PT1	2014-01-01	Chapter 20D	20
5	PT2	2014-01-01	Chapter 14D	20
6	PT2	2014-01-01	Chapter 14D	20

```
# Lets sort our dataset by our PatientID column in ascending order then the TreatmentStart in descending order
df.sort_values('PatientID', 'TreatmentStart', ascending = [True, False])\.head()
```

	PatientID	TreatmentStart	Drug	Dosage
0	PT0	2014-01-01	Chapter 14D	20
1	PT0	2015-12-01	Chapter 14D	20
2	PT1	2015-12-01	Chapter 14D	20
3	PT1	2014-01-01	Chapter 14D	20
4	PT1	2014-01-01	Chapter 20D	20

```
# Lets the DataFrame by Dosage in descending order
df.sort_values('Dosage', ascending = False)
```

	PatientID	TreatmentStart	Drug	Dosage
23	PT19	2015-01-01	Chapter 20D	20
17	PT5	2014-01-01	Chapter 20D	20
20	PT6	2014-01-01	Chapter 20D	20
21	PT6	2014-01-01	Chapter 20D	20
22	PT6	2014-01-01	Chapter 20D	20
16	PT5	2014-01-01	Chapter 14D	20
15	PT5	2014-01-01	Chapter 14D	20
14	PT4	2014-01-01	Chapter 20D	20
13	PT4	2014-01-01	Chapter 14D	20
12	PT4	2014-01-01	Chapter 14D	20
11	PT3	2014-01-01	Chapter 20D	20
10	PT3	2014-01-01	Chapter 14D	20
9	PT3	2014-01-01	Chapter 14D	20
8	PT2	2014-01-01	Chapter 20D	20
7	PT2	2014-01-01	Chapter 14D	20
6	PT2	2014-01-01	Chapter 14D	20
5	PT2	2014-01-01	Chapter 14D	20
4	PT1	2014-01-01	Chapter 20D	20
3	PT1	2014-01-01	Chapter 14D	20
2	PT1	2015-12-01	Chapter 14D	20
1	PT0	2015-12-01	Chapter 14D	20
0	PT0	2014-01-01	Chapter 14D	20

```
# Lets delete the column
df.drop('Drug', inplace=True, axis=1)
```

```
# Updated our new data
df.head()
```

	PatientID	TreatmentStart	Dosage
0	PT0	2014-01-01	20
3	PT1	2014-01-01	20
4	PT1	2014-01-01	20
5	PT2	2014-01-01	20
6	PT2	2014-01-01	20

```
# Lets put some information
df.head(10)
```

	PatientID	TreatmentStart	Dosage
0	PT0	2014-01-01	20
3	PT1	2014-01-01	20
4	PT1	2014-01-01	20
5	PT2	2014-01-01	20
6	PT2	2014-01-01	20
7	PT2	2014-01-01	20
8	PT2	2014-01-01	20
9	PT3	2014-01-01	20
10	PT3	2014-01-01	20
11	PT3	2014-01-01	20

```
# Lets put some information
df.head(20)
```

	PatientID	TreatmentStart	Dosage
0	PT0	2014-01-01	20
3	PT1	2014-01-01	20
4	PT1	2014-01-01	20
5	PT2	2014-01-01	20
6	PT2	2014-01-01	20
7	PT2	2014-01-01	20
8	PT2	2014-01-01	20
9	PT3	2014-01-01	20
10	PT3	2014-01-01	20
11	PT3	2014-01-01	20
12	PT4	2014-01-01	20
13	PT4	2014-01-01	20
14	PT4	2014-01-01	20
15	PT5	2014-01-01	20
16	PT5	2014-01-01	20
17	PT5	2014-01-01	20
18	PT6	2014-01-01	20
19	PT6	2014-01-01	20

```
# Lets delete the column
df.drop('Dosage', inplace=True, axis=1)
```

```
# Updated our new data
df.head()
```

	PatientID	TreatmentStart
0	PT0	2014-01-01
3	PT1	2014-01-01
4	PT1	2014-01-01
5	PT2	2014-01-01
6	PT2	2014-01-01

```
# Lets put some information
df.head(10)
```

	PatientID	TreatmentStart
0	PT0	2014-01-01
3	PT1	2014-01-01
4	PT1	2014-01-01
5	PT2	2014-01-01
6	PT2	2014-01-01
7	PT2	2014-01-01
8	PT2	2014-01-01
9	PT3	2014-01-01
10	PT3	2014-01-01
11	PT3	2014-01-01

```
# Lets put some information
df.head(20)
```

	PatientID	TreatmentStart
0	PT0	2014-01-01
3	PT1	2014-01-01
4	PT1	2014-01-01
5	PT2	2014-01-01
6	PT2	2014-01-01
7	PT2	2014-01-01
8	PT2	2014-01-01
9	PT3	2014-01-01
10	PT3	2014-01-01
11	PT3	2014-01-01
12	PT4	2014-01-01
13	PT4	2014-01-01
14	PT4	2014-01-01
15	PT5	2014-01-01
16	PT5	2014-01-01
17	PT5	2014-01-01
18	PT6	2014-01-01
19	PT6	2014-01-01

Data Analysis

```
# Returns the number of patients
len(df['PatientID'])
```

```
24
```

```
# Returns the number of unique patients
len(df['PatientID'].unique())
```

```
20
```

Note: As you can see there are 20 unique patients and the length of patients is 24, which means there are 4 Patients on two different medications.

Patients treated at practice

How many Patients does the practice treat?

```
print("The practice treats %s patients".format(len(df.patient[:], 'patients')
The practice treats 20 patients
```

Drug usage at practice

What drugs are used at practice?

```
# Implement the groupby function to group the drugs in a DataFrame
df.groupby('Drug').count()
```

Drug	PatientID	TreatmentStart	Dosage
Chapter 14D	10	10	10
Nexlosmab	10	10	10

```
# Stores the above code correctly for better reading by using read_select function
p = df.groupby('Drug').count()[['PatientID', 'TreatmentStart', 'Dosage']].reset_index()
```

Drug	PatientID	TreatmentStart	Dosage
Chapter 14D	10	10	10
Nexlosmab	10	10	10

```
# Get dataframe to print in the format we want. The above code did not permanently modify our DataFrame.
# We permanently modify the DataFrame we could be used inplace=True.
p[['Drug', 'PatientID']]
```

	Drug	PatientID
0	Chapter 14D	10
1	Nexlosmab	10

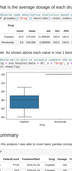
Before plotting the comparison between Drug & Patient Count we have to assign that DataFrame to a variable for flexibility.

```
count = df.groupby('Drug').count()[['PatientID']].reset_index()
```

```
count[['Drug', 'PatientID']]
```

	Drug	PatientID
0	Chapter 14D	10
1	Nexlosmab	10

```
# First begin by using a count variable to compare our 2 columns
fig = sns.barplot(data = count, x = 'Drug', y = 'PatientID')
fig.set_title('Drug Count')
plt.xlabel('Drug')
plt.ylabel('# Patients')
plt.show(fig)
```



Note: As shown above twice as many patients are taking Chapter compared to Nexlosmab.

Changes to treatment overtime

Our current Dataset shows treatments by start date. But we would like to compare the changes in treatments by the month to see how many patients start a given drug overtime

```
# Begin by creating a new column based only on TreatmentStart month
df['TreatmentStartMonth'] = df['TreatmentStart'].apply('month.format')
```

Do we see any changes, patterns in treatment overtime?

```
# Updated our new column
df.head()
```

	PatientID	TreatmentStart	Drug	Dosage	TreatmentStartMonth
0	PT0	2014-01-01	Chapter 14D	20	2014-01-01
3	PT1	2014-01-01	Chapter 14D	20	2014-01-01
4	PT1	2014-01-01	Chapter 20D	20	2014-01-01
5	PT2	2014-01-01	Chapter 14D	20	2014-01-01
6	PT2	2014-01-01	Chapter 14D	20	2014-01-01

```
# Now implement the groupby function to compare TreatmentStartMonth, Drug, with counting Patients.
drug_by_month = df.groupby('TreatmentStartMonth', 'Drug').count()[['PatientID']].reset_index()
```

```
# Compare using to the top drug count started based on the month
drug_by_month.head()
```

	TreatmentStartMonth	Drug	PatientID
0	2014-01-01	Chapter 7	
1	2014-01-01	Chapter 7	
2	2014-01-01	Chapter 1	
3	2014-01-01	Nexlosmab 1	
4	2014-01-01	Chapter 1	

```
# Lets make our Seaborn plot again to visualize how many patients started each drug by month
# The hue argument allows us to see the breakdown of each month and each drug
fig = sns.lineplot(data = drug_by_month, x = 'TreatmentStartMonth', y = 'PatientCount', hue = 'Drug')
fig.set_title('Monthly Drug Comparison')
plt.show(fig)
```

What is the average dosage of each drug?

```
# Generate some descriptive statistics based on the drug column
df.groupby('Drug').describe().reset_index()
```

	Drug	count	mean	std	min	25%	50%	75%	max
0	Chapter 14D	10.0	20.0	0.000000	20.0	20.0	20.0	20.0	20.0
1	Nexlosmab	14.0	240.000	3.000000	240.0	240.0	240.0	240.0	240.0

Note: As shown above there are 10 count for Chapter 14D and 14 for Nexlosmab with a standard deviation of 240 which means each patient who was under Nexlosmab recieved a total dosage of 240.

```
# Generate an plot to visually compare the average dosage of each drug
fig = sns.boxplot(data = df, x = 'Drug', y = 'Dosage')
fig.set_title('Drug Dosage')
plt.show(fig)
```


Summary

In this analysis I was able to cover basic pandas concepts. Overall I was able to explore our data set with the help of pandas and seaborn library. Some data cleaning was also done to further analyze our data.

```
df
```

	PatientID	TreatmentStart	Drug	Dosage	TreatmentStartMonth
0	PT0	2014-01-01	Chapter 14D	20	2014-01-01
3	PT1	2014-01-01	Chapter 14D	20	2014-01-01
4	PT1	2014-01-01	Chapter 20D	20	2014-01-01
5	PT2	2014-01-01	Chapter 14D	20	2014-01-01
6	PT2	2014-01-01	Chapter 14D	20	2014-01-01
7	PT2	2014-01-01	Chapter 14D	20	2014-01-01
8	PT2	2014-01-01	Chapter 20D	20	2014-01-01
9	PT3	2014-01-01	Chapter 14D	20	2014-01-01
10	PT3	2014-01-01	Chapter 14D	20	2014-01-01
11	PT3	2014-01-01	Chapter 20D	20	2014-01-01
12	PT4	2014-01-01	Chapter 14D	20	2014-01-01
13	PT4	2014-01-01	Chapter 14D	20	2014-01-01
14	PT4	2014-01-01	Chapter 20D	20	2014-01-01
15	PT5	2014-01-01	Chapter 14D	20	2014-01-01
16	PT5	2014-01-01	Chapter 14D	20	2014-01-01
17	PT5	2014-01-01	Chapter 20D	20	2014-01-01
18	PT6	2014-01-01	Chapter 14D	20	2014-01-01
19	PT6	2014-01-01	Chapter 14D	20	2014-01-01
20	PT6	2014-01-01	Chapter 20D	20	2014-01-01
21	PT6	2014-01-01	Chapter 20D	20	2014-01-01
22	PT6	2014-01-01	Chapter 20D	20	2014-01-01
23	PT6	2014-01-01	Chapter 20D	20	2014-01-01