Exploratory Data Analysis

Understanding your data

Contents

- Exploratory Data Analysis
- Interactive notebooks
- Data loading & processing
- Data Visualization
- Example
- Hackathon goals

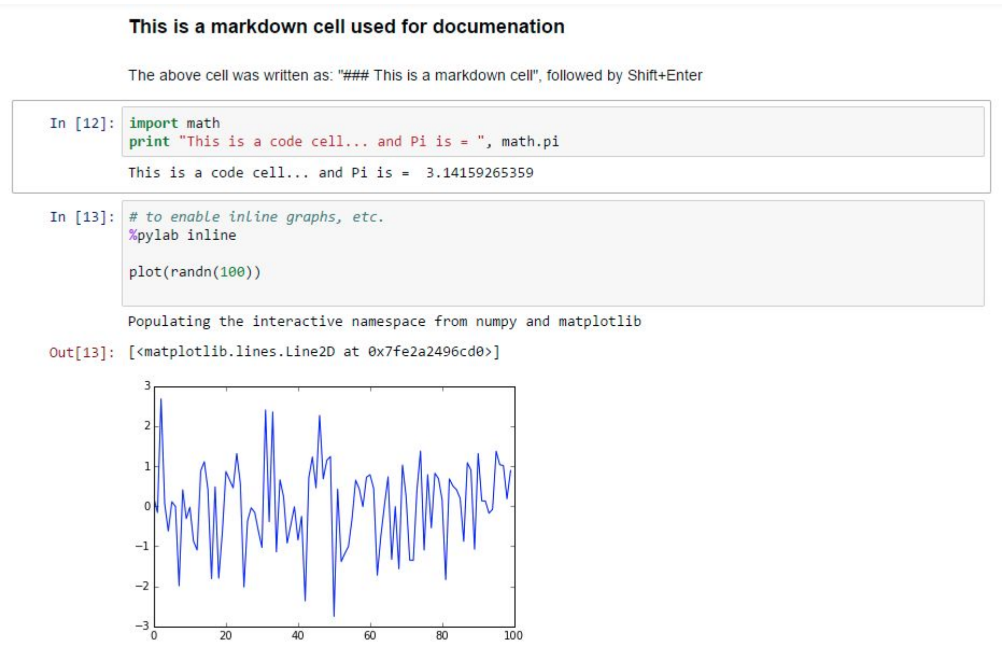
Exploratory Data Analysis

- Understanding data
- Spotting patterns & relations
- Machine Learning “Preconditions”

Jupyter notebooks

- Interactive (python) shell

- Code, story & charts



This is a markdown cell used for documenation

The above cell was written as: "### This is a markdown cell", followed by Shift+Enter

```
In [12]: import math
         print "This is a code cell... and Pi is = ", math.pi

         This is a code cell... and Pi is =  3.14159265359
```

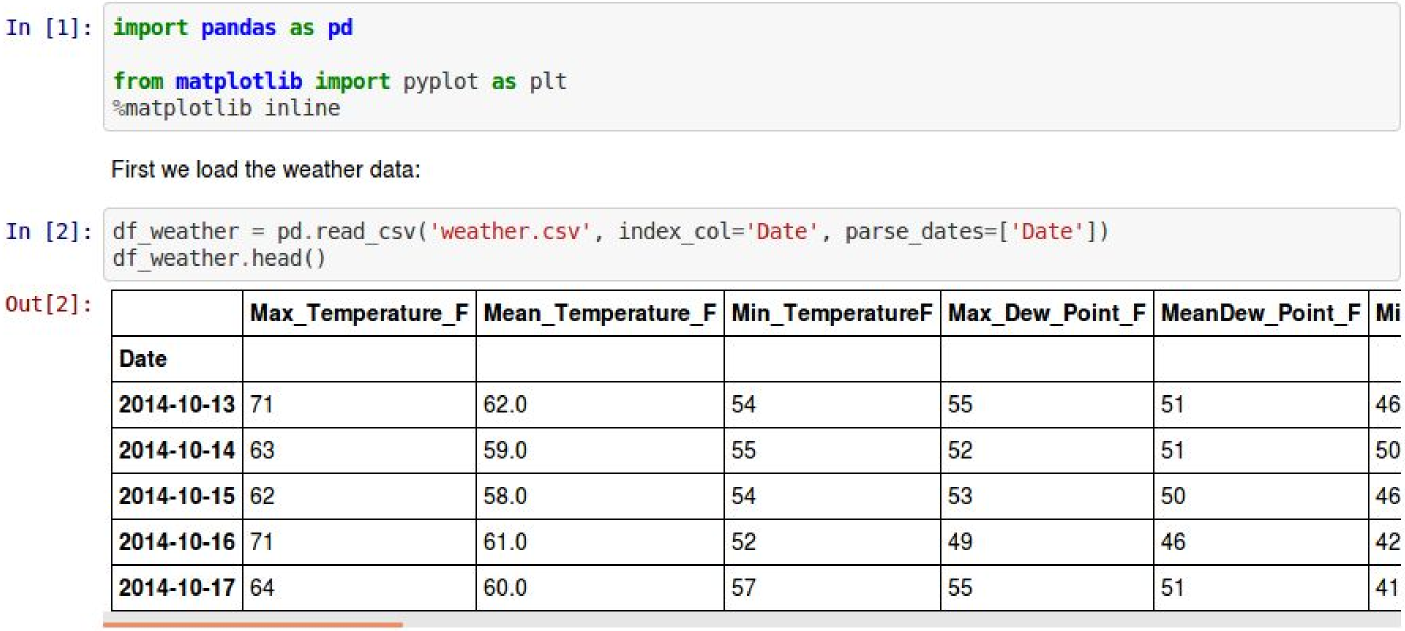
```
In [13]: # to enable inline graphs, etc.
         %pylab inline

         plot(randn(100))

         Populating the interactive namespace from numpy and matplotlib
Out[13]: [<matplotlib.lines.Line2D at 0x7fe2a2496cd0>]
```

Pandas - Data processing

Pandas cheat sheet

- Load, show data
- Concat, join dataframes
- Filter, sort rows
- Summarize data
- Apply functions, calculations
- Create/drop columns
- Group by, aggregate

Pandas - Data processing



In [1]:

```
import pandas as pd

from matplotlib import pyplot as plt
%matplotlib inline
```

First we load the weather data:

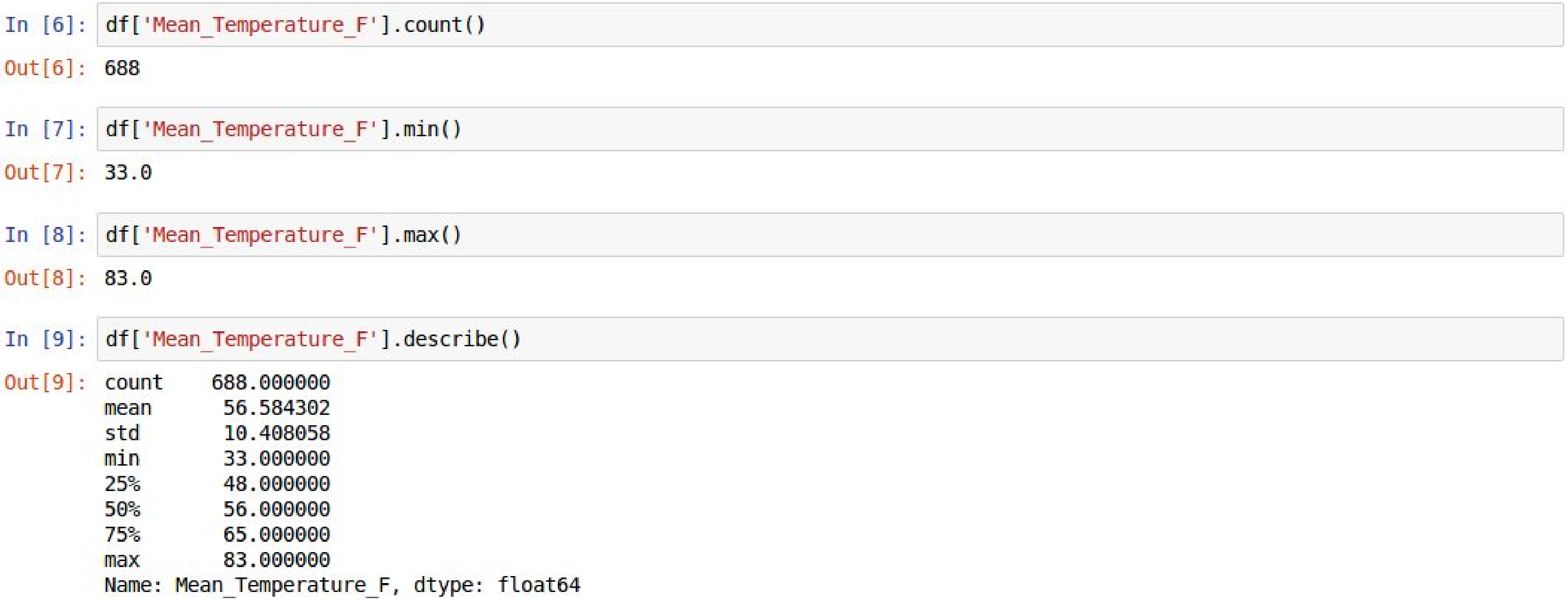
In [2]:

```
df_weather = pd.read_csv('weather.csv', index_col='Date', parse_dates=['Date'])
df_weather.head()
```

Out[2]:

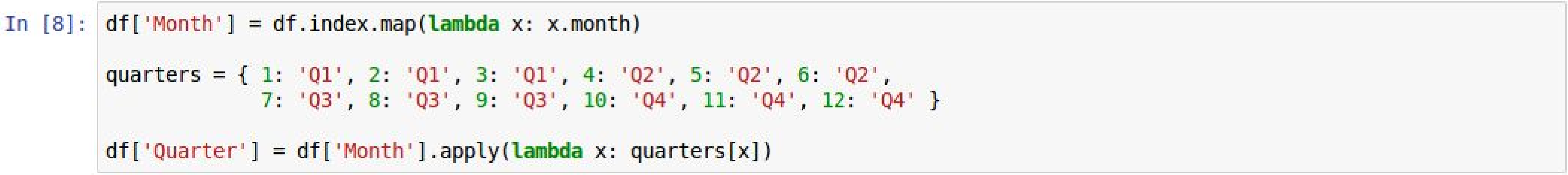
	Max_Temperature_F	Mean_Temperature_F	Min_TemperatureF	Max_Dew_Point_F	MeanDew_Point_F	Mi
Date						
2014-10-13	71	62.0	54	55	51	46
2014-10-14	63	59.0	55	52	51	50
2014-10-15	62	58.0	54	53	50	46
2014-10-16	71	61.0	52	49	46	42
2014-10-17	64	60.0	57	55	51	41

Pandas - Data processing



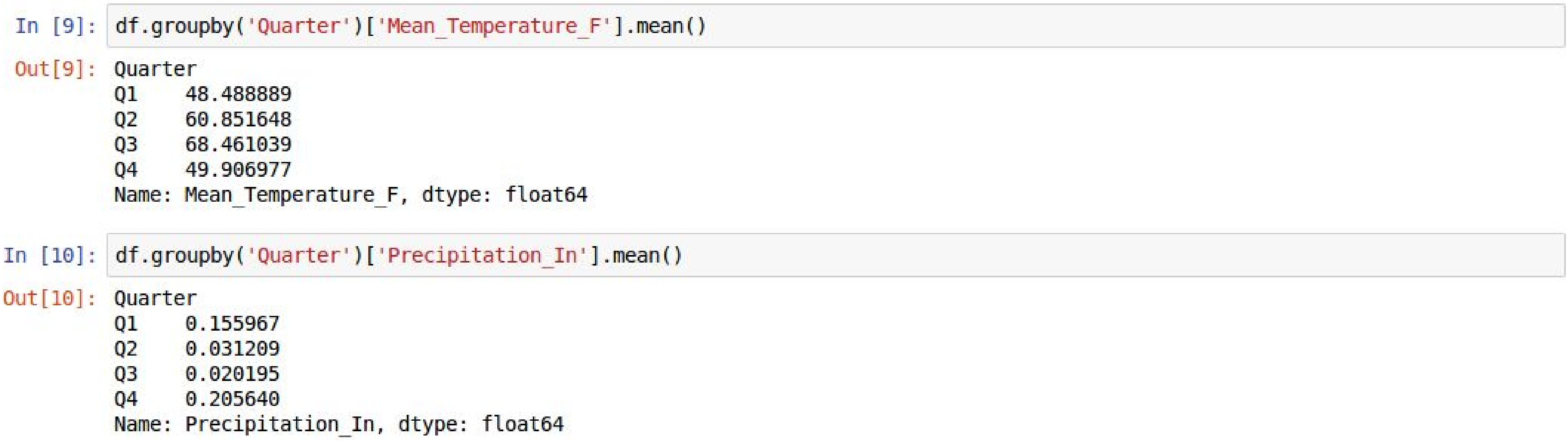
```
In [6]: df['Mean_Temperature_F'].count()

Out[6]: 688

In [7]: df['Mean_Temperature_F'].min()

Out[7]: 33.0

In [8]: df['Mean_Temperature_F'].max()

Out[8]: 83.0

In [9]: df['Mean_Temperature_F'].describe()

Out[9]: count    688.000000
        mean      56.584302
        std       10.408058
        min       33.000000
        25%       48.000000
        50%       56.000000
        75%       65.000000
        max       83.000000
        Name: Mean_Temperature_F, dtype: float64
```

Pandas - Data processing



```
In [8]: df['Month'] = df.index.map(lambda x: x.month)

        quarters = { 1: 'Q1', 2: 'Q1', 3: 'Q1', 4: 'Q2', 5: 'Q2', 6: 'Q2',
                     7: 'Q3', 8: 'Q3', 9: 'Q3', 10: 'Q4', 11: 'Q4', 12: 'Q4' }

        df['Quarter'] = df['Month'].apply(lambda x: quarters[x])
```

Pandas - Data processing



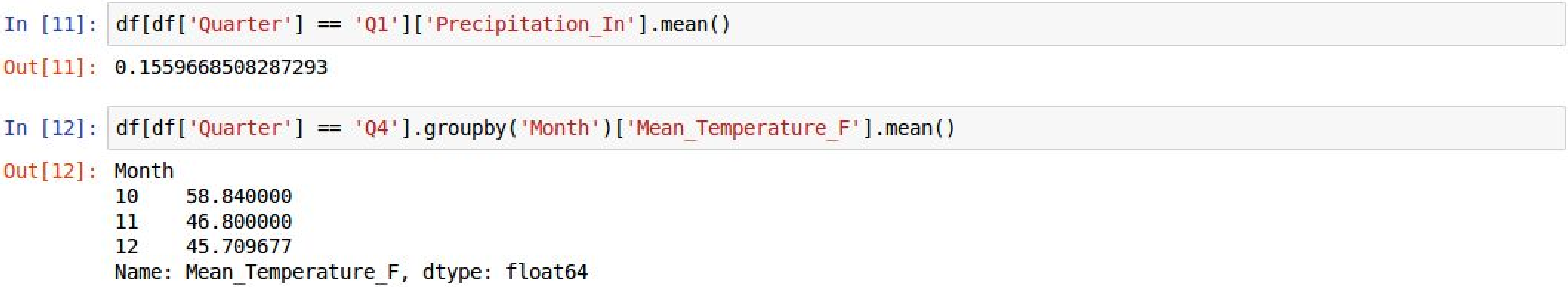
```
In [9]: df.groupby('Quarter')['Mean_Temperature_F'].mean()
```

```
Out[9]: Quarter
        Q1    48.488889
        Q2    60.851648
        Q3    68.461039
        Q4    49.906977
        Name: Mean_Temperature_F, dtype: float64
```

```
In [10]: df.groupby('Quarter')['Precipitation_In'].mean()
```

```
Out[10]: Quarter
         Q1    0.155967
         Q2    0.031209
         Q3    0.020195
         Q4    0.205640
         Name: Precipitation_In, dtype: float64
```

Pandas - Data processing



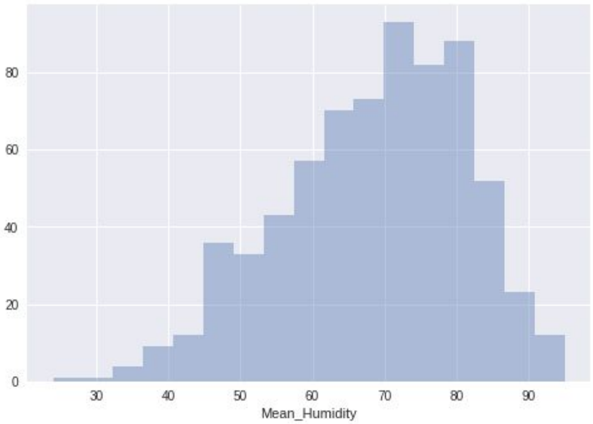
```
In [11]: df[df['Quarter'] == 'Q1']['Precipitation_In'].mean()

Out[11]: 0.1559668508287293

In [12]: df[df['Quarter'] == 'Q4'].groupby('Month')['Mean_Temperature_F'].mean()

Out[12]: Month
         10    58.840000
         11    46.800000
         12    45.709677
         Name: Mean_Temperature_F, dtype: float64
```

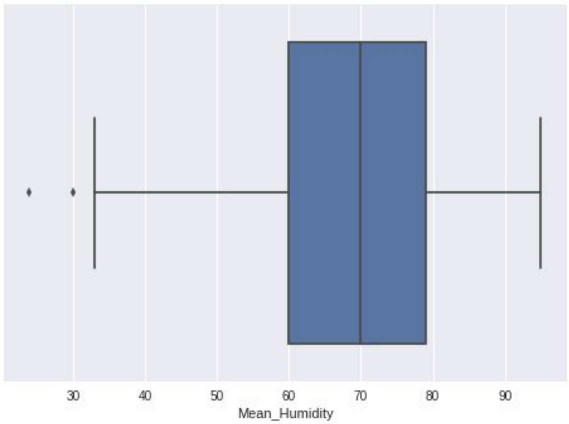
Pandas - Understanding

- Goal: Understand data
    - Individual variables
        - Types (numerical, categorical)
        - Range
        - Distribution shape
    - Relations
        - Correlation
        - Hypothesis tests
        - ...

Seaborn - Data Visualization

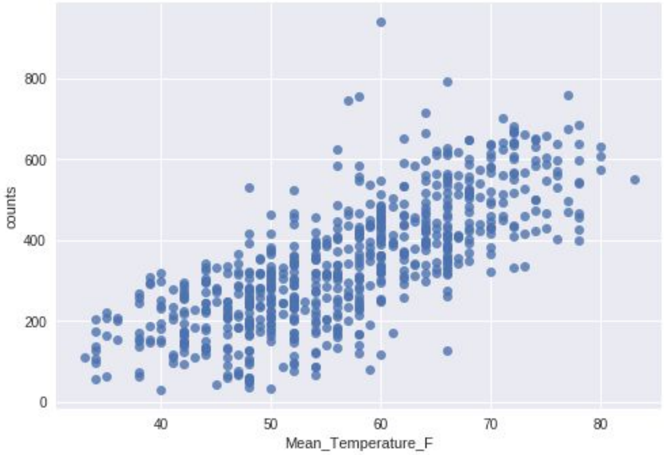
- Plot one variable
  - Histogram & boxplot

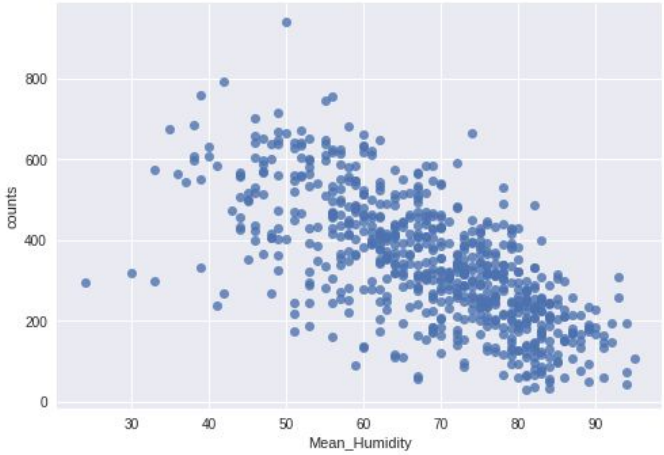




Seaborn - Data Visualization

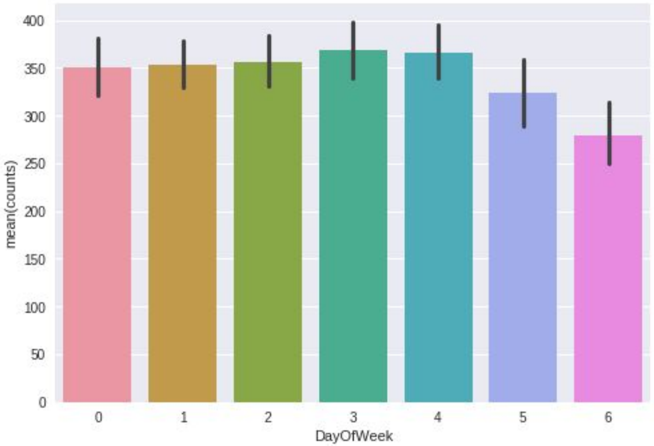
- Plot relations between (numerical) variables
  - Scatterplot

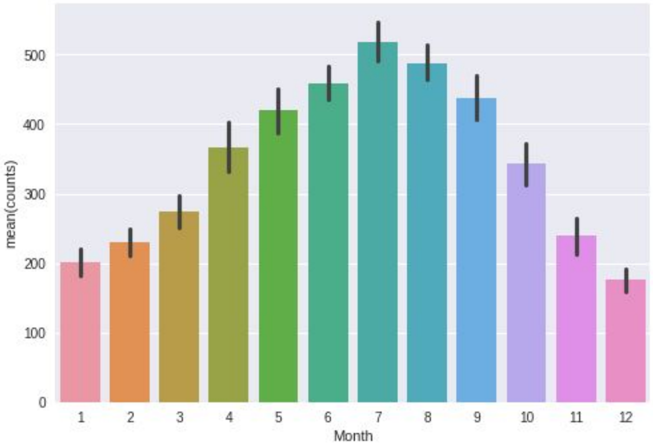




Seaborn - Data Visualization

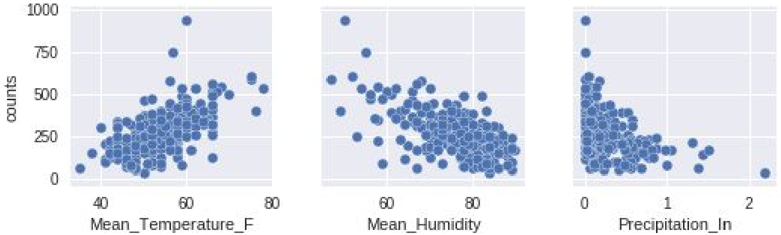
- Plot relations between (categorical) variables
  - Barplot & boxplot

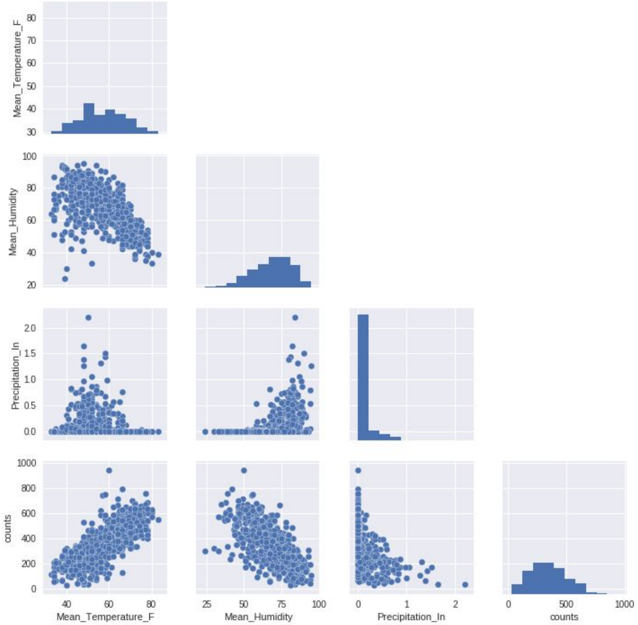




Seaborn - Data Visualization

- Pairplot - plot many relations





Seaborn - Understanding

- Goal: Understand data
  - Individual variables
  - Relations

Example

- Data Analysis Titanic Survivors

Hackathon goals

- Understand the data
- Find some research questions
  - Popular weekdays of stations (any differences?)
  - Popular stations (influence of new nearby stations?)
  - Upward/downward trends of station usage?
  - What makes a station popular? (Close to city center? Close to public transport?)
  - Be creative!
- See if data supports your ideas

Questions ?

- Let’s start hacking !

Advanced topics

- 3D plot of multi-variable relations ?
- Geo visualization: Plotting on a map ?
- Be creative !