Exploratory Data Analysis (EDA)

Justin Post

Big Picture

- Big Data characteristics
- Course split into four topics
    1. Programming in `python`
    2. Big Data Management
    3. Modeling Big Data (with `Spark` via `pyspark`)
    4. Streaming Data

Uses for Data

Four major goals with data:

1. Description
2. Prediction/Classification
3. Inference
4. Pattern Finding

- First step of most analysis is to get to know your data! Done through an **Exploratory Data Analysis**

EDA

- Essentially **Descriptive Statistics** with a bit more big picture stuff about your data
- EDA generally consists of a few steps:
    - Understand how your data is stored
    - Do basic data validation
    - Determine rate of missing values
    - Clean data up data as needed
    - Investigate distributions
        - Univariate measures/graphs
        - Multivariate measures/graphs
    - Apply transformations and repeat previous step

Understand how your data is stored

- Should know if your data has read in how you think it should!

Read in some data (we'll learn more about this later!)

```
import pandas as pd
wine_data = pd.read_csv("https://www4.stat.ncsu.edu/~online/datasets/winequality-full.csv")
wine_data.head()
```

```
##    fixed acidity  volatile acidity  citric acid  ...  alcohol  quality  type
## 0            7.4              0.70         0.00  ...      9.4        5   Red
## 1            7.8              0.88         0.00  ...      9.8        5   Red
## 2            7.8              0.76         0.04  ...      9.8        5   Red
## 3           11.2              0.28         0.56  ...      9.8        6   Red
## 4            7.4              0.70         0.00  ...      9.4        5   Red
##
## [5 rows x 13 columns]
```

Understand how your data is stored

- Should know if your data has read in how you think it should!

```
wine_data.info()
```

```
## <class 'pandas.core.frame.DataFrame'>
## RangeIndex: 6497 entries, 0 to 6496
## Data columns (total 13 columns):
##  #   Column                Non-Null Count  Dtype  
## ---  ------                --------------  -----  
##  0   fixed acidity         6497 non-null   float64
##  1   volatile acidity      6497 non-null   float64
##  2   citric acid           6497 non-null   float64
##  3   residual sugar        6497 non-null   float64
##  4   chlorides             6497 non-null   float64
##  5   free sulfur dioxide   6497 non-null   float64
##  6   total sulfur dioxide  6497 non-null   float64
##  7   density               6497 non-null   float64
##  8   pH                    6497 non-null   float64
##  9   sulphates             6497 non-null   float64
##  10  alcohol               6497 non-null   float64
##  11  quality               6497 non-null   int64  
##  12  type                  6497 non-null   object 
## dtypes: float64(11), int64(1), object(1)
## memory usage: 660.0+ KB
```

Do basic data validation

- Usually look at quick summary stats of all the data to check that things make sense

```
wine_data.describe()
```

```
##        fixed acidity  volatile acidity  ...      alcohol      quality
## count    6497.000000       6497.000000  ...  6497.000000  6497.000000
## mean        7.215307          0.339666  ...    10.491801     5.818378
## std         1.296434          0.164636  ...     1.192712     0.873255
## min         3.800000          0.080000  ...     8.000000     3.000000
## 25%         6.400000          0.230000  ...     9.500000     5.000000
## 50%         7.000000          0.290000  ...    10.300000     6.000000
## 75%         7.700000          0.400000  ...    11.300000     6.000000
## max        15.900000          1.580000  ...    14.900000     9.000000
##
## [8 rows x 12 columns]
```

Determine rate of missing values

- Every programming language has indicators for missing values
- In python, we use `NaN` for 'not a number' (in `pandas`) (might use other things for missing with other data objects/modules)

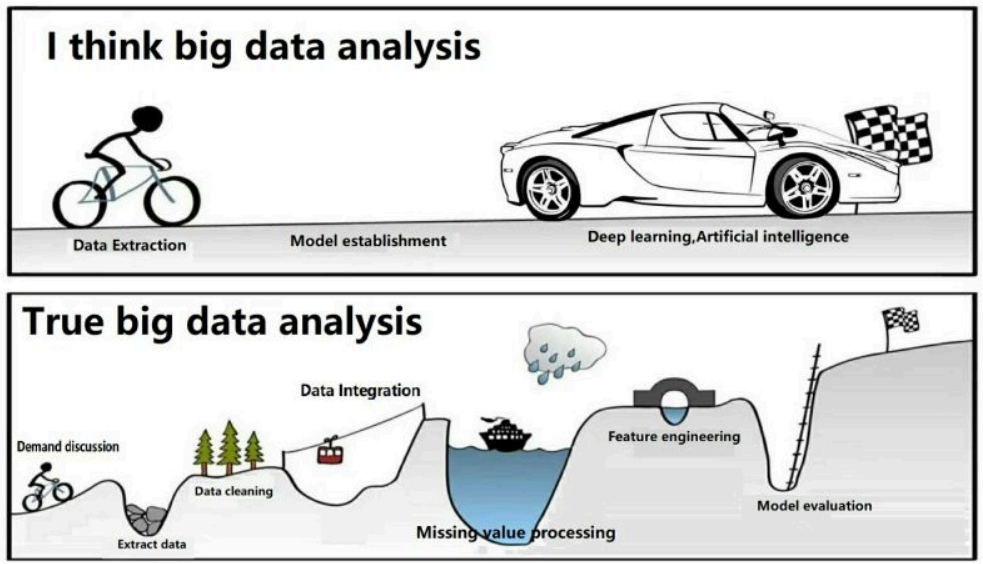
```
wine_data.isnull().sum()
```

```
## fixed acidity           0
## volatile acidity        0
## citric acid             0
## residual sugar          0
## chlorides               0
## free sulfur dioxide     0
## total sulfur dioxide    0
## density                 0
## pH                      0
## sulphates               0
## alcohol                 0
## quality                 0
## type                    0
## dtype: int64
```

Clean data up data as needed

May need to

- reread data with different specifications
- fill missing values
- remove some rows and/or columns
- check your data against some gold standard?



https://devrant.com/rants/2161326/rant

Investigate distributions

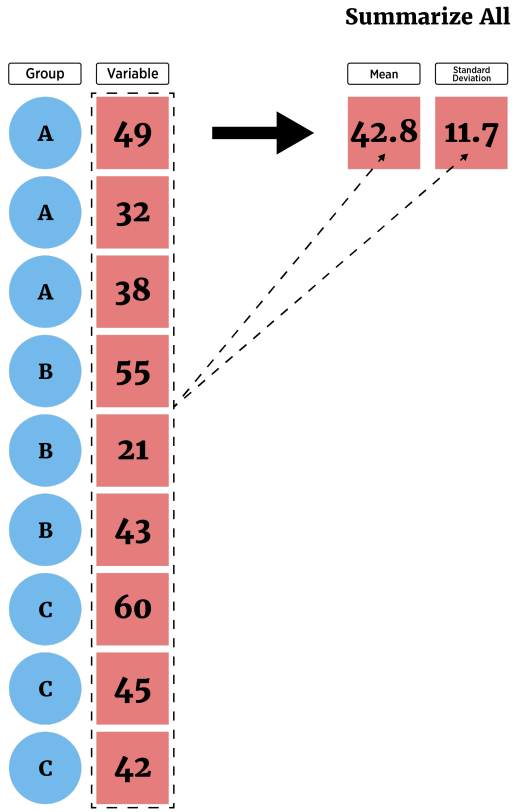
Goal: Understand types of data and their distributions

- Univariate measures/graphs
- Multivariate measures/graphs

Investigate distributions

Goal: Understand types of data and their distributions

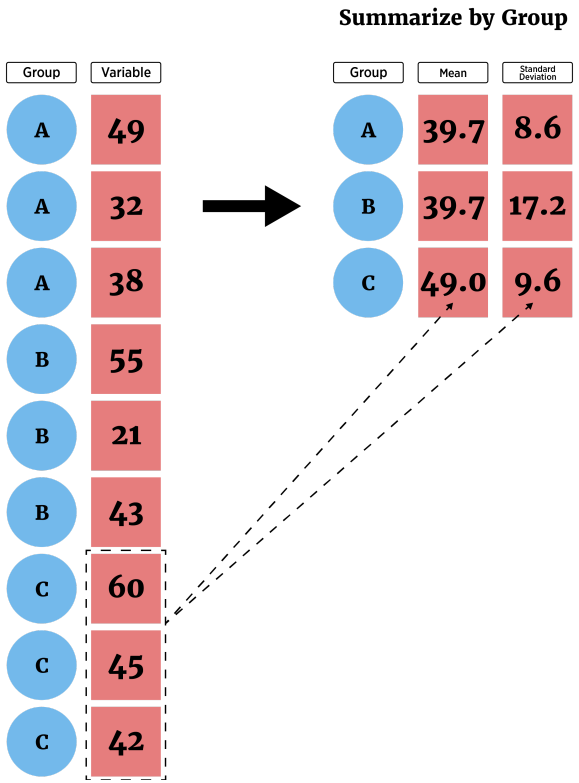
- Numerical summaries



Making Sense of Data

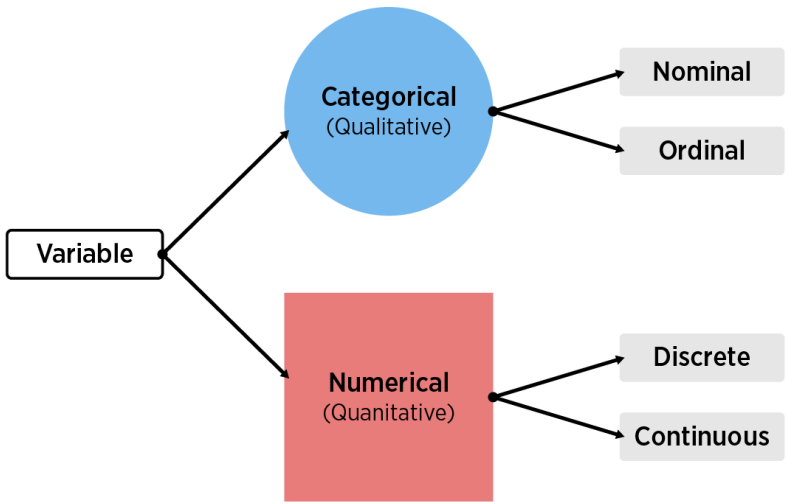
Goal: Understand types of data and their distributions

- Numerical summaries (across subgroups)



Types of Data

- How to summarize data depends on the type of data
  - Categorical (Qualitative) variable - entries are a label or attribute
  - Numeric (Quantitative) variable - entries are a numerical value where math can be performed



Making Sense of Data

Goal: Understand types of data and their distributions

- Numerical summaries (across subgroups)
    - Contingency Tables
    - Mean/Median
    - Standard Deviation/Variance/IQR
    - Quantiles/Percentiles

Categorical Data

Goal: Describe the **distribution** of the variable

- Distribution = pattern and frequency with which you observe a variable
- Categorical variable - entries are a label or attribute
    - Describe the relative frequency (or count) for each category
    - Called a **contingency table**

Categorical Variable Summary - One-way Table

- Count the # of times each category of **one** variable appears!

```
wine_data.type #treat like a numpy array
```

```
## 0         Red
## 1         Red
## 2         Red
## 3         Red
## 4         Red
##          ...
## 6492    White
## 6493    White
## 6494    White
## 6495    White
## 6496    White
## Name: type, Length: 6497, dtype: object
```

```
sum(wine_data.type == "Red")
```

```
## 1599
```

```
sum(wine_data.type == "White")
```

```
## 4898
```

Categorical Variable Summary - Two-way Table

- Count the # of times each **combination** of categories for *two* variables appear!
- Consider `quality` and `type`

```
sum((wine_data.type == "Red") & (wine_data.quality == 3))
```

```
## 10
```

```
sum((wine_data.type == "Red") & (wine_data.quality == 4))
```

```
## 53
```

```
sum((wine_data.type == "Red") & (wine_data.quality == 5))
```

```
## 681
```

```
#etc
```

Numeric Data

Goal: Describe the **distribution** of the variable

- Distribution = pattern and frequency with which you observe a variable
- Numeric variable - entries are a numerical value where math can be performed

For a single numeric variable, describe the distribution via

- Shape: Histogram, Density plot, ...
- Measures of center: Mean, Median, ...
- Measures of spread: Variance, Standard Deviation, Quartiles, IQR, ...

For two numeric variables, describe the distribution via

- Shape: Scatter plot; Measures of linear relationship: Covariance, Correlation

Numerical Variable Location Summary - Mean

- Sample mean: for a variable in our data set (call it y)

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

```
sum(wine_data.alcohol)/len(wine_data.alcohol)
```

```
## 10.491800831152855
```

Numerical Variable Location Summary - Median

- Sample median
  - Sort values
  - Value with 50% of data below and above is the median
  - If even number of observations, average middle two values

```
sorted_alcohol = wine_data.alcohol.sort_values()
sorted_alcohol
```

```
## 4864     8.00
## 4224     8.00
## 5438     8.40
## 5434     8.40
## 544      8.40
##          ...
## 4544    14.00
## 588     14.00
## 6102    14.05
## 5517    14.20
## 652     14.90
## Name: alcohol, Length: 6497, dtype: float64
```

```
len(sorted_alcohol)/2
```

```
## 3248.5
```

```
sorted_alcohol.values[3248]
```

```
## 10.3
```

Numerical Variable Spread Summary - Variance

- Sample variance is *almost* the average squared deviation from the mean

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

```
sub = wine_data[0:4].chlorides
sub
```

```
## 0    0.076
## 1    0.098
## 2    0.092
## 3    0.075
## Name: chlorides, dtype: float64
```

```
mean_chlorides = sum(sub)/4
mean_chlorides
```

```
## 0.08525
```

```
sub-mean_chlorides
```

```
## 0   -0.00925
## 1    0.01275
## 2    0.00675
## 3   -0.01025
## Name: chlorides, dtype: float64
```

```
(sub-mean_chlorides)**2
```

```
## 0    0.000086
## 1    0.000163
## 2    0.000046
## 3    0.000105
## Name: chlorides, dtype: float64
```

```
sum((sub-mean_chlorides)**2)/3
```

```
## 0.00013291666666666674
```

Numerical Variable Spread Summary - Standard Deviation

- Sample Standard Deviation = square root of sample variance
  - Puts metric on the scale of the variable

```
import numpy as np
np.sqrt(sum((sub-mean_chlorides)**2)/3)
```

```
## 0.011528949070347511
```

Numerical Variable Spread Summary - Quantiles/Percentiles

- Sample quantile - a generalization of the median
  - p^{th} quantile - value with p% of the values below it
  - Also called the 100*p%ile

```
len(sorted_alcohol)/2
```

```
## 3248.5
```

```
#obtain 0.25 quantile (median of lower half of the data)
(sorted_alcohol.values[1624]+sorted_alcohol.values[1623])/2
```

```
## 9.5
```

Numerical Variable Relationship Summary - Correlation

- Sample correlation - a measure of the **linear** relationship between two variables
  - Call the variables x and y
  - (x_i, y_i) are numeric variables observed on the same n units, $i = 1, \ldots, n$
  - Pearson's correlation coefficient:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Numerical Variable Relationship Summary - Correlation

- Sample correlation - a measure of the **linear** relationship between two variables

```
wine_data.loc[0:4, ["fixed acidity", "chlorides"]]
```

```
##    fixed acidity  chlorides
## 0            7.4      0.076
## 1            7.8      0.098
## 2            7.8      0.092
## 3           11.2      0.075
## 4            7.4      0.076
```

```
wine_data.loc[0:4, ["fixed acidity", "chlorides"]].corr()
```

```
##                fixed acidity  chlorides
## fixed acidity       1.000000  -0.322814
## chlorides          -0.322814   1.000000
```

Numerical Variable Relationship Summary - Correlation

- Sample correlation - a measure of the **linear** relationship between two variables
    - Sensitive to outliers
    - Spearman's correlation coefficient simply uses Pearson's correlation on the ranks of the data!

```
wine_data.loc[0:4, ["fixed acidity", "chlorides"]]
```

```
##    fixed acidity  chlorides
## 0            7.4      0.076
## 1            7.8      0.098
## 2            7.8      0.092
## 3           11.2      0.075
## 4            7.4      0.076
```

Recap

- EDA generally consists of a few steps:
    - Understand how your data is stored
    - Do basic data validation
    - Determine rate of missing values
    - Clean data up data as needed
    - Investigate distributions
        - Univariate measures/graphs
        - Multivariate measures/graphs
    - Apply transformations and repeat previous step
- Usually want summaries for different **subgroups of data**!!