Technion CS-236756

Introduction to Machine Learning

Welcome to the course!

Lecturer: Dr. Yonatan Belinkov.
Teaching assistants: Itay, Arkadi, Yonatan, Shani, and Edan.

Introduction

- Updates & Materials: in the [webcourse]
- Questions & Discussions: in the [Piazza forum]
- Grade
    - 74% - Exam (must pass)
    - 18% - Major programming assignments (3 assignments in pairs)
    - 8% - Short assignments (4 assignments, submitted individually)

Short assignment 1 is already published

Longer than the other short assignments, but shorter than the major "wet" assignments.

1. Get familiar with important python libraries for data science
2. Two probability questions
3. Two algebra questions

Major wet assignments

- 3 assignments, each is 6% of the final grade.
- Submissions in pairs only.
- Let you experience the more practical aspects of learning while applying ideas and algorithms learned in class.
- Rough partition:
    1. Data preparation
    2. Algorithm implementation
    3. Modeling & Classification

Technion CS-236756

Introduction to Machine Learning

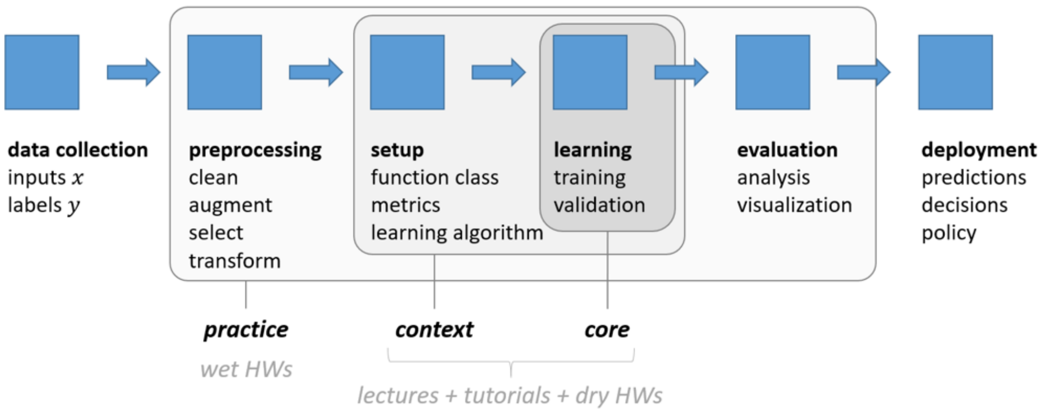
Tutorial 1: Data Exploration with Python

Ronen Nir & Itay Evron

Tutorial outline

- The learning pipeline
- Overview and motivations
- Steps of Data Exploration
    - Variable Identification
    - Univariate Analysis
    - Bi-variate Analysis

The learning pipeline



Data Exploration Overview

- Analyze datasets to reveal their main characteristics
- Find what the data can tell us
    - Some patterns can be revealed through visualizations and charts
- Using statistical methods to identify variables

- Using statistical methods to identify variables

Motivation

- With a new dataset in hand, the first thing you do is data exploration
    - Data exploration and data preparation take up to 80% of the time in many ML projects

- Exploring the data gives insights about what you can and cannot do
- Garbage in, garbage out
    - Exploring the data can help you make it better later on
    - *More data beats cleverer algorithms, better data beats more data*
        - (Peter Norvig)

Data Exploration Steps

1. Understanding the data
    - Variable Identification
    - Variable Analysis
2. Improving the data
    - Missing values
    - Outliers
    - Variable Engineering

Packages Relevant to Data Exploration

First, import relevant packages.

```
import pandas as pd # data analysis and manipulation tool
import numpy as np # Numerical computing tools
import seaborn as sns  # visualization library
import matplotlib.pyplot as plt  # another visualization library
import warnings
warnings.filterwarnings('ignore')
```

Step I: Understanding the Data

Throughout this tutorial we shall use the `tips` dataset for demonstration and examples from [Kaggle's visualization tutorial](#).

```
tips = sns.load_dataset('tips')
tips.shape
```

```
(244, 7)
```

Variable Identification

> What do we need to know about the variables in a dataset?

```
tips.head()
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Variable Types

We focus on two types of variables:

1. Continuous variables (`total_bill`, `tip`)
2. Categorical variables (`sex`, `smoker`, `day`, `time`)

Discussion

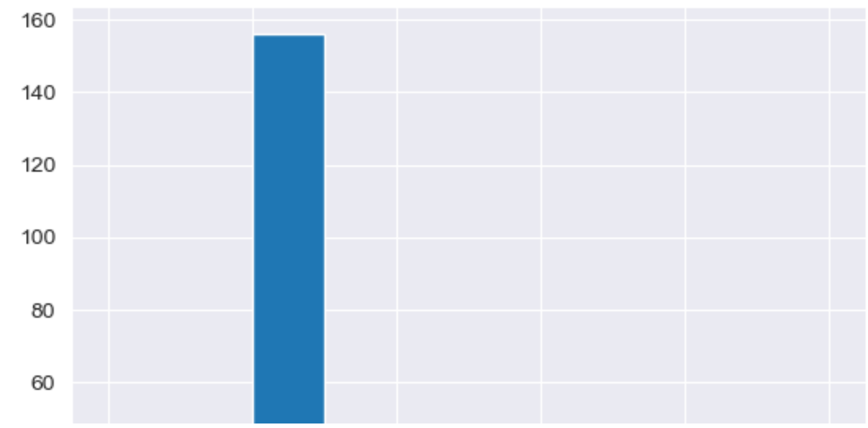
> How would you treat the 'size' variable?

```
tips['size'].value_counts()
```

```
size
2    156
3     38
4     37
5      5
1      4
6      4
Name: count, dtype: int64
```

```
tips['size'].hist()
```

```
<Axes: >
```



Features & Target Variables

- Target variables are what we care about, and we want to infer from the features (predictor variables)
- The features are often denoted as X and target variables are denoted y

Discussion

> What variables are the *predictor variables* and what variables are *target variables*?

```
tips.sample(5)
```

	total_bill	tip	sex	smoker	day	time	size
103	22.42	3.48	Female	Yes	Sat	Dinner	2
127	14.52	2.00	Female	No	Thur	Lunch	2
7	26.88	3.12	Male	No	Sun	Dinner	4
210	30.06	2.00	Male	Yes	Sat	Dinner	3
32	15.06	3.00	Female	No	Sat	Dinner	2

Data Understanding - Important Tip

- Pandas assumes a certain variable type to each column
- Doublecheck it with the attribute `dtypes`

```
tips.dtypes
```

```
total_bill     float64
tip            float64
sex           category
smoker        category
day           category
time          category
size             int64
dtype: object
```

Univariate Analysis

Explores the variables one by one

- Continuous variables
    - Use statistical metrics and visualization methods to understand the nature of the variable
- Categorical variables
    - Tables that describe distribution of each category

Univariate Analysis (Continuous Variables)

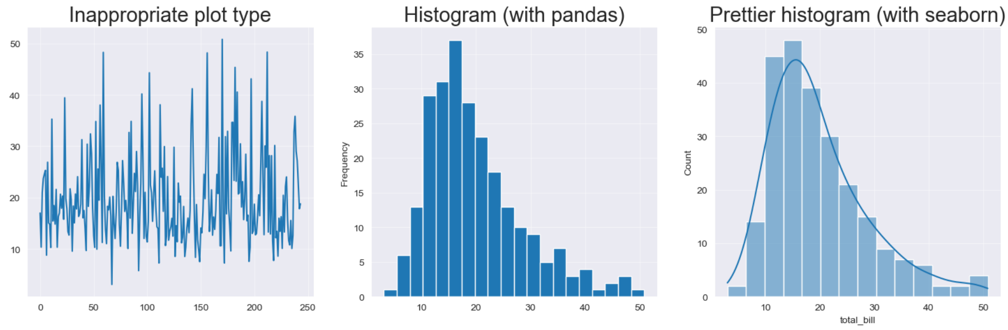
```
fig, axes = plt.subplots(1,3, figsize=(18, 5))

tips['total_bill'].plot(ax=axes[0])  # don't do that
axes[0].set_title("Inappropriate plot type", fontsize=21)

tips.total_bill.plot(kind='hist', bins=20, ax=axes[1]) # better
axes[1].set_title("Histogram (with pandas)", fontsize=21)

sns.histplot(tips.total_bill, kde=True, ax=axes[2]) # prettier with seaborn
axes[2].set_title("Prettier histogram (with seaborn)", fontsize=21)

for ax in axes:
    ax.grid(alpha=0.5)
```



Univariate Analysis (Categorical Variables)

```
tips['sex'].value_counts()
```

```
sex
Male      157
Female     87
Name: count, dtype: int64
```

```
tips.groupby('sex').smoker.value_counts()
```

```
sex     smoker
Male    No        97
        Yes       60
Female  No        54
        Yes       33
Name: count, dtype: int64
```

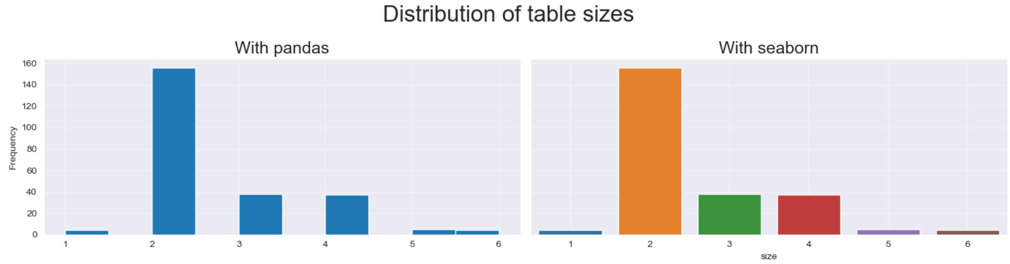
Visualization of Categorical Variables

```
fig, (ax1, ax2) = plt.subplots(ncols=2, sharey=True, figsize=(15,4))
tips["size"].plot(kind="hist", ax=ax1) # plot with pandas
ax1.grid(alpha=0.5)
```

```
ax1.set_title("With pandas", fontsize=18)

sns.countplot(data = tips, x="size", ax=ax2)  # prettier with seaborn
ax2.grid(alpha=0.5)
ax2.set_title("With seaborn", fontsize=18)

plt.suptitle("Distribution of table sizes", fontsize=24)
_ = plt.tight_layout()
```



Bi-variate Analysis

Explore the relationship between two variables

1. Continuous and continuous
2. Categorical and continuous
3. Categorical and categorical

Bi-Variate Analysis of 2 Continuous Variables

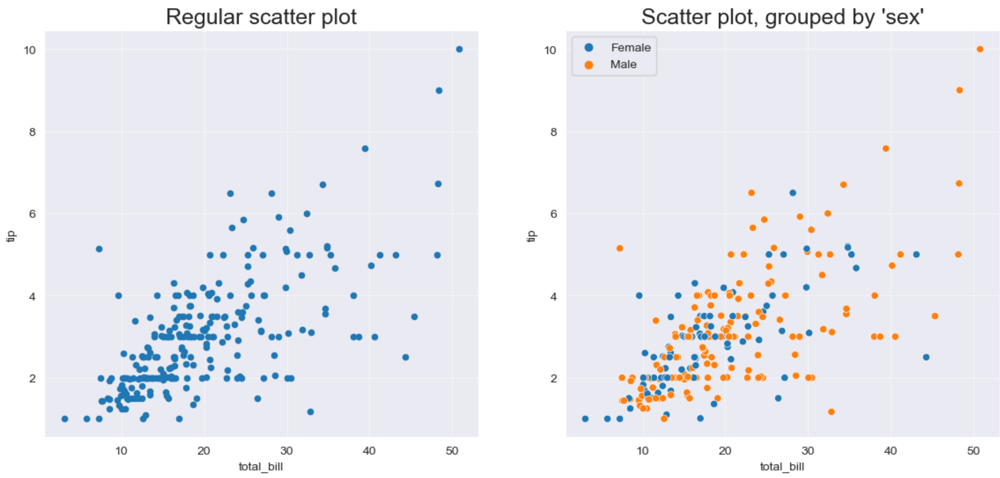
1. Your first visualization of 2 variables should be a *scatter plot*
    1. Keep it simple!
2. Use statistical methods to check the relationship between two variables
    1. e.g. computing the correlation between 2 variables

```
fig, axes = plt.subplots(1,2, figsize=(14, 6))

tips.plot(x='total_bill', y='tip', kind='scatter', ax=axes[0]) # plot with panc
axes[0].set_title("Regular scatter plot", fontsize=18)

sns.scatterplot(x='total_bill', y='tip', hue=tips.sex.to_list(), data=tips, ax=
axes[1].set_title("Scatter plot, grouped by 'sex'", fontsize=18)

for ax in axes:
    ax.grid(alpha=0.5)
```



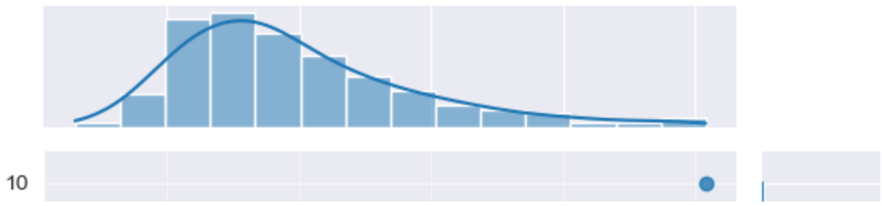
Correlation

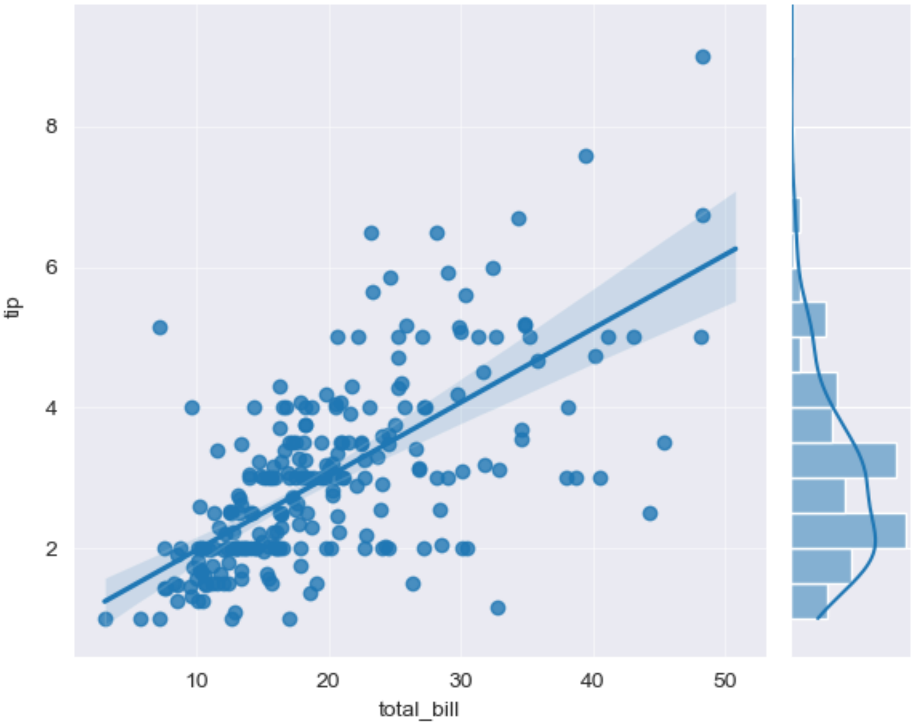
- Tests how strongly pairs of variables are *linearly* related
    - For example, height and weight are related
- The Correlation between two variables (X,Y) is defined to be: $\frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}$

```
x = tips.total_bill
y = tips.tip
# Compute correlation
print("Correlation is: {:.3f}".format(tips['total_bill'].corr(tips['tip'])))

g = sns.jointplot(data=tips, x="total_bill", y="tip", kind="reg")
_ = g.ax_joint.grid(alpha=0.5)
```

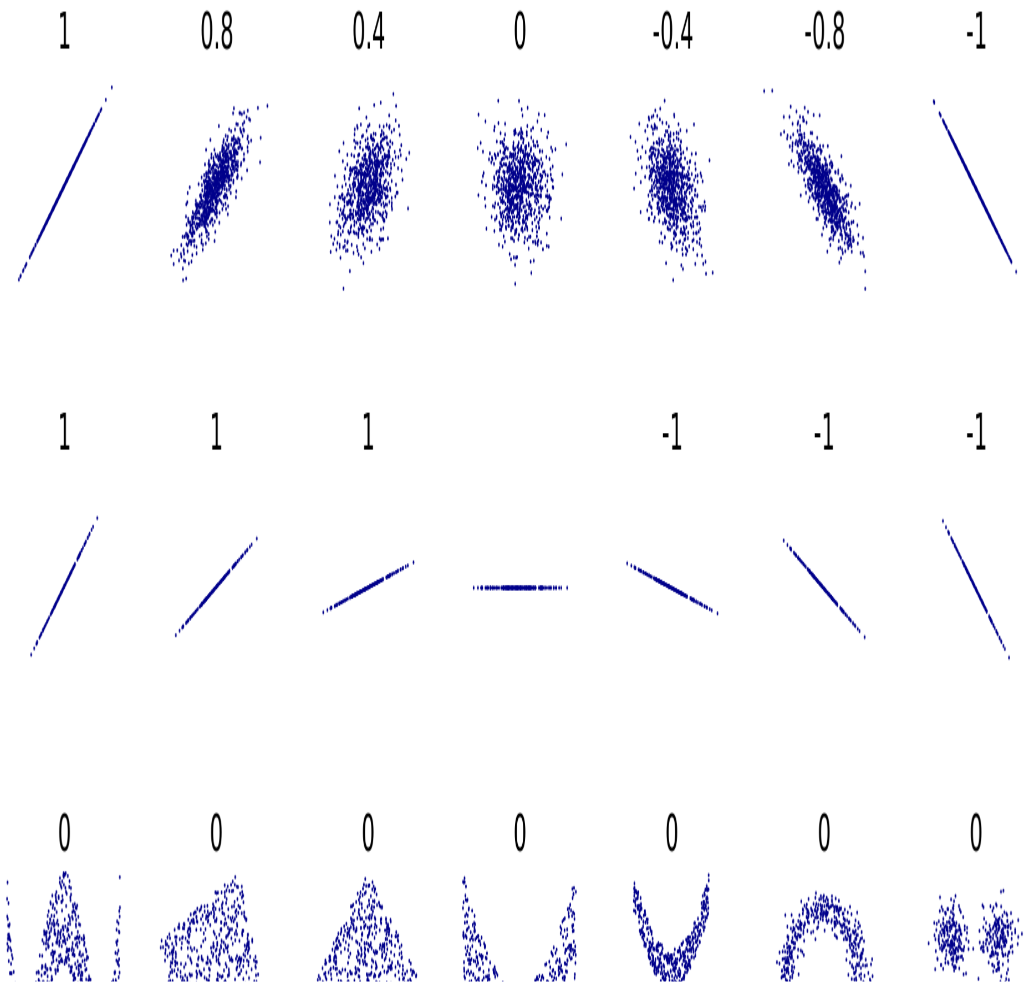
```
Correlation is: 0.676
```





Discussion (feature selection)

> What will you do if you find two correlative variables? Delete them? Keep them?
> It depends!

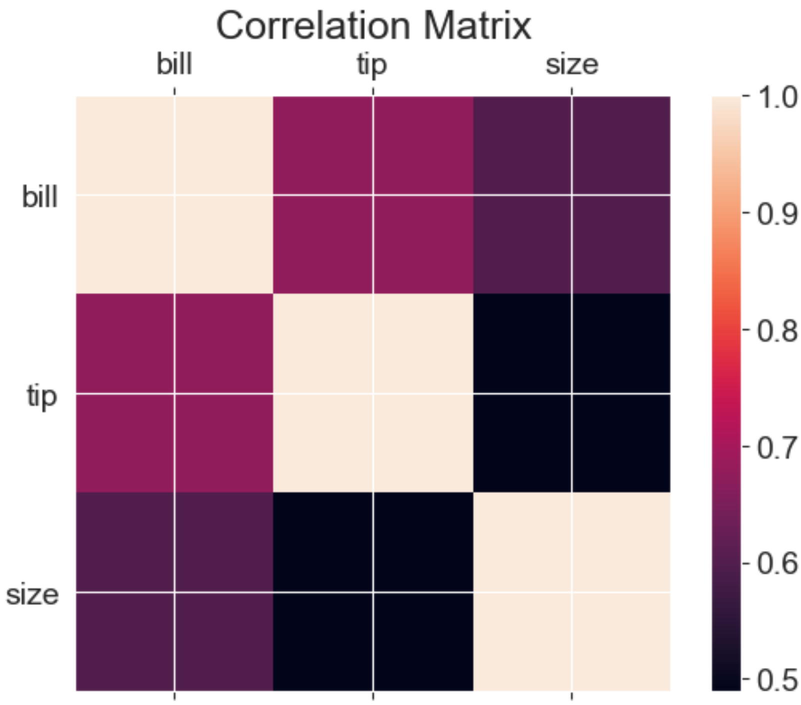


Correlation between continuous variables in `tips`

```
tips.corr(numeric_only=True)
```

	total_bill	tip	size
total_bill	1.000000	0.675734	0.598315
tip	0.675734	1.000000	0.489299
size	0.598315	0.489299	1.000000

```
f = plt.figure()
plt.matshow(tips.corr(numeric_only=True), fignum=f.number)
plt.xticks(range(3), ['bill', 'tip', 'size'], fontsize=14)
plt.yticks(range(3), ['bill', 'tip', 'size'], fontsize=14)
cb = plt.colorbar()
cb.ax.tick_params(labelsize=14)
_ = plt.title('Correlation Matrix', fontsize=18)
```



Bi-Variate Analysis of 2 Categorical Variables

1. Two-way table
2. Stacked Bar Chart
3. Statistical tests like Chi-square (out of scope)

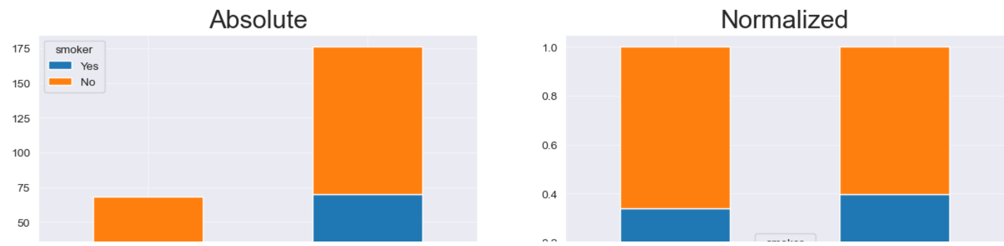
Examples of Bi-Variate Analysis in `tips`

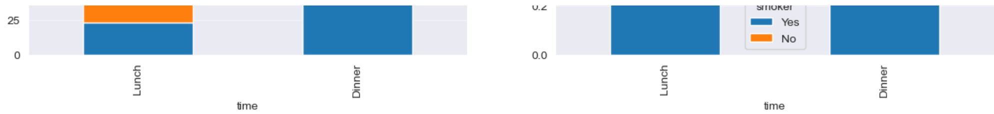
```
print(tips.groupby('time')['smoker'].value_counts(normalize=True))
pd.crosstab(tips['time'], tips['smoker'])
```

```
time    smoker
Lunch   No        0.661765
        Yes       0.338235
Dinner  No        0.602273
        Yes       0.397727
Name: proportion, dtype: float64
```

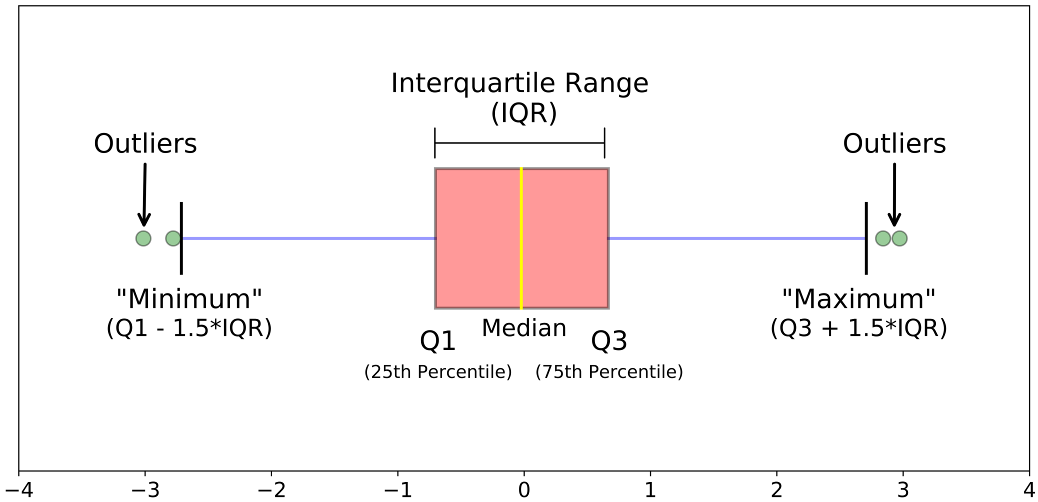
smoker	Yes	No
time		
Lunch	23	45
Dinner	70	106

```
fig, (ax1, ax2) = plt.subplots(ncols=2, sharey=False, figsize=(15,4))
ax1.grid(alpha=0.5)
ax2.grid(alpha=0.5)
ax1.set_title("Absolute", fontsize=22)
ax2.set_title("Normalized", fontsize=22)
_ = pd.crosstab(tips['time'], tips['smoker']).plot(kind='bar', stacked=True, ax
_ = pd.crosstab(tips['time'], tips['smoker'], normalize="index").plot(kind='bar
```





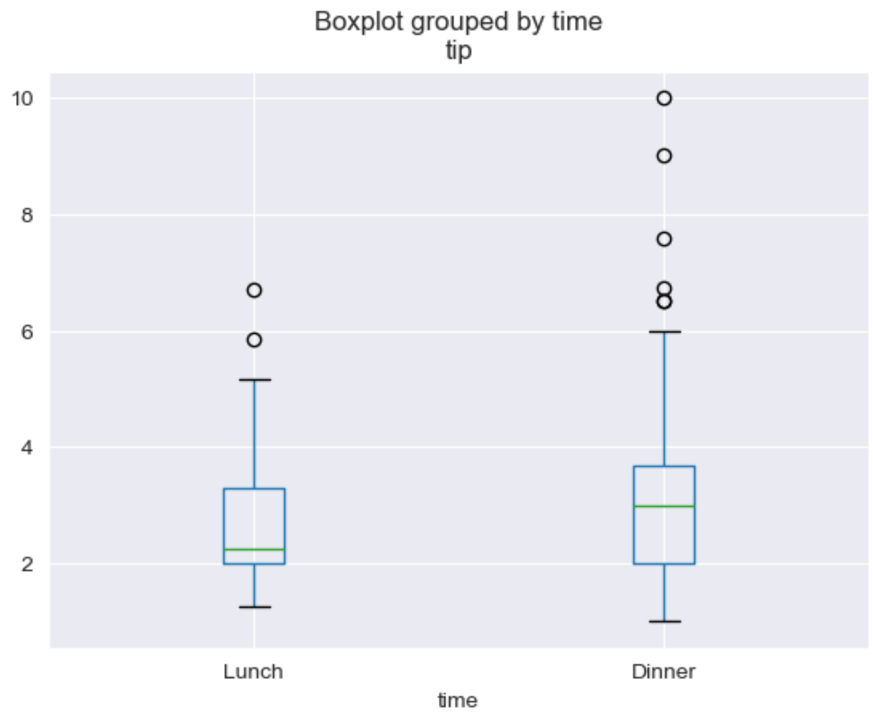
Analysis of Categorical and Continuous with Box plots



Credit: [towards data science]()

Box Plots in `tips`

```
_ = tips.boxplot(by='time', column='tip', grid=True)
#_ = tips.boxplot(by='day', column='tip', grid=True)
```



(Bonus) Creating New Variables

- Use expert/common knowledge to improve the data
- E.g. Humans like round numbers so customers tend to round the payment

> Can we design a variable that emphasizes this?

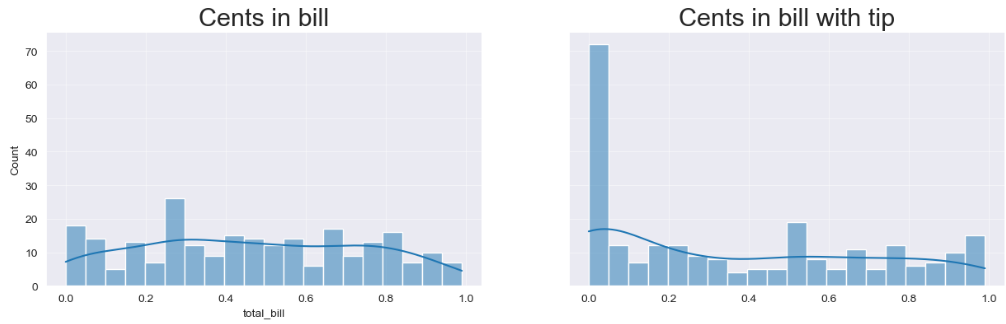
```
bill_with_tip = tips['total_bill'] + tips['tip']

fig, (ax1, ax2) = plt.subplots(ncols=2, sharey=True, figsize=(15,4))

sns.histplot((tips.total_bill - np.floor(tips.total_bill)), ax=ax1, kde=True, b
ax1.set_title("Cents in bill", fontsize=22)

sns.histplot(bill_with_tip - np.floor(bill_with_tip), ax=ax2, kde=True, bins=20
ax2.set_title("Cents in bill with tip", fontsize=22)

_ = ax1.grid(alpha=0.5), ax2.grid(alpha=0.5)
```



Step II: Improving the data

In major assignment 1