DS501 STATISTICAL AND MATHEMATICAL METHODS FOR DATA SCIENCE

Dr. Muhammad Wasim

PhD, MS, M.Phil, M.Sc, MCS

Certified Data Analyst [KARACHI.AI]

Lecture Week 02-03

- Probability

PROBABILITY

- The chance that something will happen
- Probability as a mathematical framework for reasoning about uncertainty
- Given infinite observations of an event, the proportion of observations where a given outcome happens
- Strength of belief that something is true
- “Mathematical language for quantifying uncertainty” – Wasserman
- **Ω** : Sample Space, set of all outcomes of a random experiment
- ***A*** : Event ($A \subseteq \Omega$), collection of possible outcomes of an experiment
- **P(*A*)**: Probability of event ***A***, **P** is a function: events$\longrightarrow\mathbb{R}$

PROBABILITY

- **P(Ω)** = 1
- **P(*A*)** ≥ 0 **,** for all ***A***
- If *A1*, *A2*, … are disjoint events then: $\mathrm{P}(\bigcup_i^\infty A_i) = \sum_i^\infty \mathrm{P}(A_i)$

Some Properties:

✓ If ***B*** ⊆ ***A*** then **P(*A*)** ≥ **P(*B*)**

✓ **P(*A*** ⋃ ***B*)** ≤ **P(*A*) + P(*B*)**

✓ **P(*A*** ⋂ ***B*)** ≤ **min(P(*A*), P(*B*))**

✓ **P(¬*A*) = P(Ω** / ***A*) = 1 - P(*A*)**

/ is set difference

P(*A* ⋂ *B*) will be notated as P(*A*, *B*)

PROBABILITY

Probabilistic models:

– sample space

– probability law

- Axioms of probability
- Simple examples

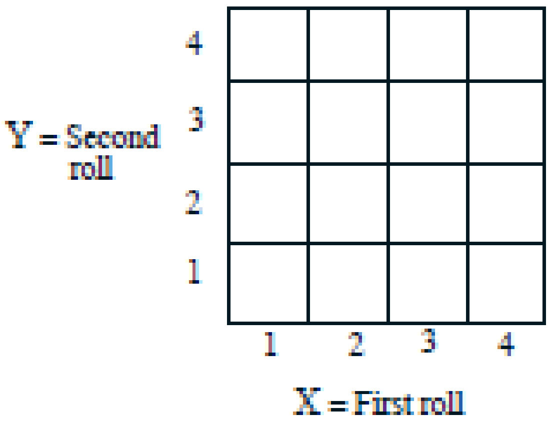
Sample space Ω:

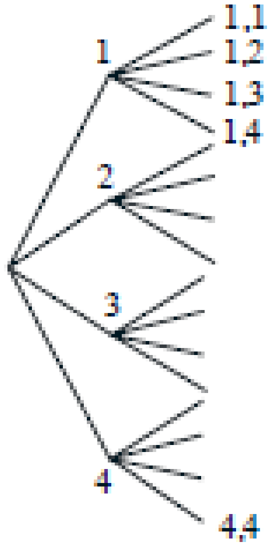
- "List" (set) of possible outcomes
- List must be:
  – Mutually exclusive
  – Collectively exhaustive

PROBABILITY

Sample space: Discrete example

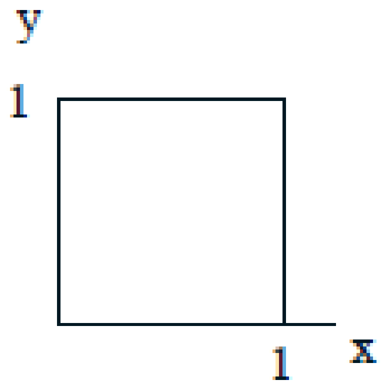
- Two rolls of a tetrahedral die
  - Sample space vs. sequential description





Sample space: Continuous example

$$\Omega = \{(x, y) \mid 0 \leq x, y \leq 1\}$$



PROBABILITY

Probability axioms

- **Event:** a subset of the sample space
- Probability is assigned to events

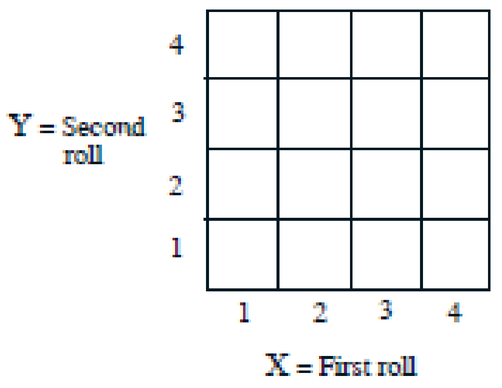
Axioms:

1. **Nonnegativity:** $\mathrm{P}(A) \geq 0$
2. **Normalization:** $\mathrm{P}(\Omega) = 1$
3. **Additivity:** If $A \cap B = \emptyset$, then $\mathrm{P}(A \cup B) = \mathrm{P}(A) + \mathrm{P}(B)$

- $$\begin{aligned}\mathrm{P}(\{s_1, s_2, \ldots, s_k\}) &= \mathrm{P}(\{s_1\}) + \cdots + \mathrm{P}(\{s_k\}) \\ &= \mathrm{P}(s_1) + \cdots + \mathrm{P}(s_k)\end{aligned}$$

- Axiom 3 needs strengthening

Probability law: Example with finite sample space



- Let every possible outcome have probability 1/16
  - $\mathrm{P}((X, Y)$ is $(1,1)$ or $(1,2)) =$
  - $\mathrm{P}(\{X = 1\}) =$
  - $\mathrm{P}(X + Y$ is odd$) =$
  - $\mathrm{P}(\min(X, Y) = 2) =$

PROBABILITY (REVIEW)

Discrete uniform law

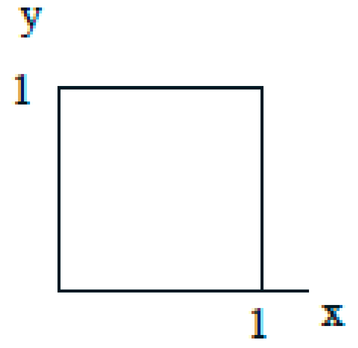
- Let all outcomes be equally likely
- Then,

$$P(A) = \frac{\text{number of elements of } A}{\text{total number of sample points}}$$

- Computing probabilities $\equiv$ counting
- Defines fair coins, fair dice, well-shuffled decks

Continuous uniform law

- Two "random" numbers in $[0, 1]$.

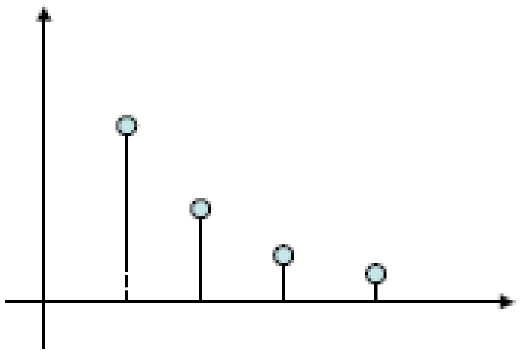


- **Uniform** law: Probability $=$ Area
  - $P(X + Y \leq 1/2) = ?$
  - $P(\,(X, Y) = (0.5, 0.3)\,)$

PROBABILITY

Probability law: Ex. w/countably infinite sample space

- Sample space: $\{1, 2, \ldots\}$
  - We are given $\mathrm{P}(n) = 2^{-n}$, $n = 1, 2, \ldots$
  - Find P(outcome is even)



$$\mathrm{P}(\{2,4,6,\ldots\}) \;=\; \mathrm{P}(2)+\mathrm{P}(4)+\cdots = \frac{1}{2^2}+\frac{1}{2^4}+\frac{1}{2^6}+\cdots = \frac{1}{3}$$

- Countable additivity axiom (needed for this calculation):
  If $A_1, A_2, \ldots$ are disjoint events, then:

$$\mathrm{P}(A_1 \cup A_2 \cup \cdots) = \mathrm{P}(A_1) + \mathrm{P}(A_2) + \cdots$$

PROBABILITY

Example (from the reading)

Experiment: toss a fair coin, report heads or tails.

Sample space: $\Omega = \{H, T\}$.

Probability function: $P(H) = .5, \quad P(T) = .5$.

Use tables:

Outcomes	H	T
Probability	1/2	1/2

(Tables can really help in complicated examples)

PROBABILITY

Events

Events are sets:

- Can describe in words
- Can describe in notation
- Can describe with Venn diagrams

Experiment: toss a coin 3 times.

Event:

You get 2 or more heads = { HHH, HHT, HTH, THH}

PROBABILITY

Events, sets and words

Experiment: toss a coin 3 times.

Which of following equals the event "exactly two heads"?

$$A = \{THH, HTH, HHT, HHH\}$$
$$B = \{THH, HTH, HHT\}$$
$$C = \{HTH, THH\}$$

To keep the notation cleaner, let's use $P(T) = (1 - p) = q$.
Since the flips are independent (we'll discuss this next week) the probabilities multiply. This gives the following 2×2 table.

		second flip	
		H	T
first flip	H	p^2	pq
	T	qp	q^2

If probability of H is p and probability of T is 1-p, the write down possible mathematical expression for A, B & C.

PROBABILITY

Events, sets and words

Experiment: toss a coin 3 times.

Which of the following describes the event $\{THH, HTH, HHT\}$?

(1) "exactly one head"
(2) "exactly one tail"
(3) "at most one tail"
(4) none of the above

PROBABILITY

Events, sets and words

Experiment: toss a coin 3 times.

The events "exactly 2 heads" and "exactly 2 tails" are disjoint.

(1) True (2) False

answer: True: $\{THH, HTH, HHT\} \cap \{TTH, THT, HTT\} = \emptyset$.

PROBABILITY

Events, sets and words

Experiment: toss a coin 3 times.

The event "at least 2 heads" implies the event "exactly two heads".

(1) True (2) False

False. It's the other way around:
$\{THH, HTH, HHT\} \subset \{THH, HTH, HHT, HHH\}$.

PROBABILITY

Probability rules in mathematical notation

Sample space: $S = \{\omega_1, \omega_2, \ldots, \omega_n\}$

Outcome: $\omega \in S$

Probability between 0 and 1: $0 \leq P(\omega) \leq 1$

Total probability is 1: $\sum_{j=1}^{n} P(\omega_j) = 1, \quad \sum_{\omega \in S} P(\omega) = 1$

Event A: $P(A) = \sum_{\omega \in A} P(\omega)$

PROBABILITY

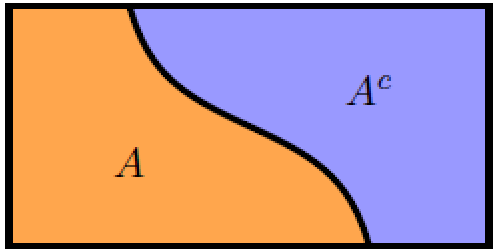
Probability and set operations on events

Events A, L, R

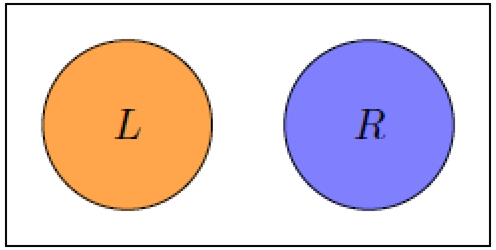
Rule 1. Complements: $P(A^c) = 1 - P(A)$.

Rule 2. Disjoint events: If L and R are disjoint then
$P(L \cup R) = P(L) + P(R)$.

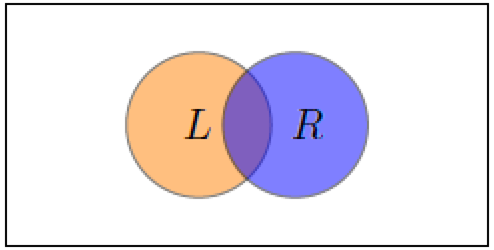
Rule 3. Inclusion-exclusion principle: For any L and R:
$P(L \cup R) = P(L) + P(R) - P(L \cap R)$.



$\Omega = A \cup A^c$, no overlap



$L \cup R$, no overlap



$L \cup R$, overlap $= L \cap R$

PERMUTATION & COMBINATION

Permutations

Lining things up. How many ways can you do it?

'abc' and 'cab' are different permutations of {a, b, c}

Permutations of k from a set of n

Give all permutations of 3 things out of $\{a, b, c, d\}$

$$\begin{array}{cccccc} abc & abd & acb & acd & adb & adc \\ bac & bad & bca & bcd & bda & bdc \\ cab & cad & cba & cbd & cda & cdb \\ dab & dac & dba & dbc & dca & dcb \end{array}$$

Would you want to do this for 7 from a set of 10?

PERMUTATION & COMBINATION

Combinations

Choosing subsets – order doesn't matter.

How many ways can you do it?

Combinations of k from a set of n

Give all combinations of 3 things out of $\{a, b, c, d\}$

Answer: {a,b,c}, {a,b,d}, {a,c,d}, {b,c,d}

PERMUTATION & COMBINATION

Permutations and Combinations

abc	acb	bac	bca	cab	cba	$\{a, b, c\}$
abd	adb	bad	bda	dab	dba	$\{a, b, d\}$
acd	adc	cad	cda	dac	dca	$\{a, c, d\}$
bcd	bdc	cbd	cdb	dbc	dcb	$\{b, c, d\}$

Permutations:

$${}_4P_3$$

Combinations:

$$\binom{4}{3} = {}_4C_3$$

$$\binom{4}{3} = {}_4C_3 = \frac{{}_4P_3}{3!}$$

PERMUTATION & COMBINATION

Board Question

(a) Count the number of ways to get exactly 3 heads in 10 flips of a coin.

(b) For a fair coin, what is the probability of exactly 3 heads in 10 flips?

answer: (a) We have to 'choose' 3 out of 10 flips for heads: $\boxed{\binom{10}{3}.}$

(b) There are 2^{10} possible outcomes from 10 flips (this is the rule of product). For a fair coin each outcome is equally probable so the probability of exactly 3 heads is

answer: $$\frac{\binom{10}{3}}{2^{10}} = \frac{120}{1024} = 0.117$$

ESTIMATION OF LOCATION (CENTRAL TENDENCY)

- Variables with measured or count data might have thousands of distinct values.
- A basic step in exploring data is getting a “typical value” for each feature (variable): an estimate of where most of the data is located (i.e., its central tendency).

ESTIMATION OF LOCATION (CENTRAL TENDENCY)

KEY TERM S FOR ESTIM ATES OF LOCATION

1. Mean:

The sum of all values divided by the number of values.

Synonyms: average

$$\text{Mean} = \bar{x} = \frac{\sum_{i}^{n} x_i}{n}$$

ESTIMATION OF LOCATION (CENTRAL TENDENCY)

2. Weighted mean

The sum of all values times a weight divided by the sum of the weights

There are two main motivations for using a weighted mean:

1. Some values are intrinsically more variable than others, and highly variable observations are given a lower weight. For example, if we are taking the average from multiple sensors and one of the sensors is less accurate, then we might down weight the data from that sensor.

ESTIMATION OF LOCATION (CENTRAL TENDENCY)

2. The data collected does not equally represent the different groups that we are interested in measuring. For example, because of the way an online experiment was conducted, we may not have a set of data that accurately reflects all groups in the user base. To correct that, we can give a higher weight to the values from the groups that were underrepresented.

Synonyms: weighted average

$$\text{Weighted mean} = \bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i}^{n} w_i}$$

ESTIMATION OF LOCATION (CENTRAL TENDENCY)

3. Median

- The value such that one-half of the data lies above and below
- The *median* is the middle number on a sorted list of the data
- If there is an even number of data values, the middle value is one that is not actually in the data set, but rather the average of the two values that divide the sorted data into upper and lower halves

Synonyms: 50th percentile

ESTIMATION OF LOCATION (CENTRAL TENDENCY)

4. Weighted median

- The value such that one-half of the sum of the weights lies above and below the sorted data
- For the same reasons that one uses a weighted mean, it is also possible to compute a *weighted median*. As with the median, we first sort the data, although each data value has an associated weight. Instead of the middle number, the weighted median is a value such that the sum of the weights is equal for the lower and upper halves of the sorted list.
- Like the median, the weighted median is robust to outliers.

ESTIMATION OF LOCATION (CENTRAL TENDENCY)

5. Trimmed mean

- The average of all values after dropping a fixed number of extreme values
- A trimmed mean eliminates the influence of extreme values

Synonyms: truncated mean

$$\text{Trimmed mean} = \bar{x} = \frac{\sum_{i=p+1}^{n-p} x_{(i)}}{n-2p}$$

ESTIMATION OF LOCATION (CENTRAL TENDENCY)

6. Robust

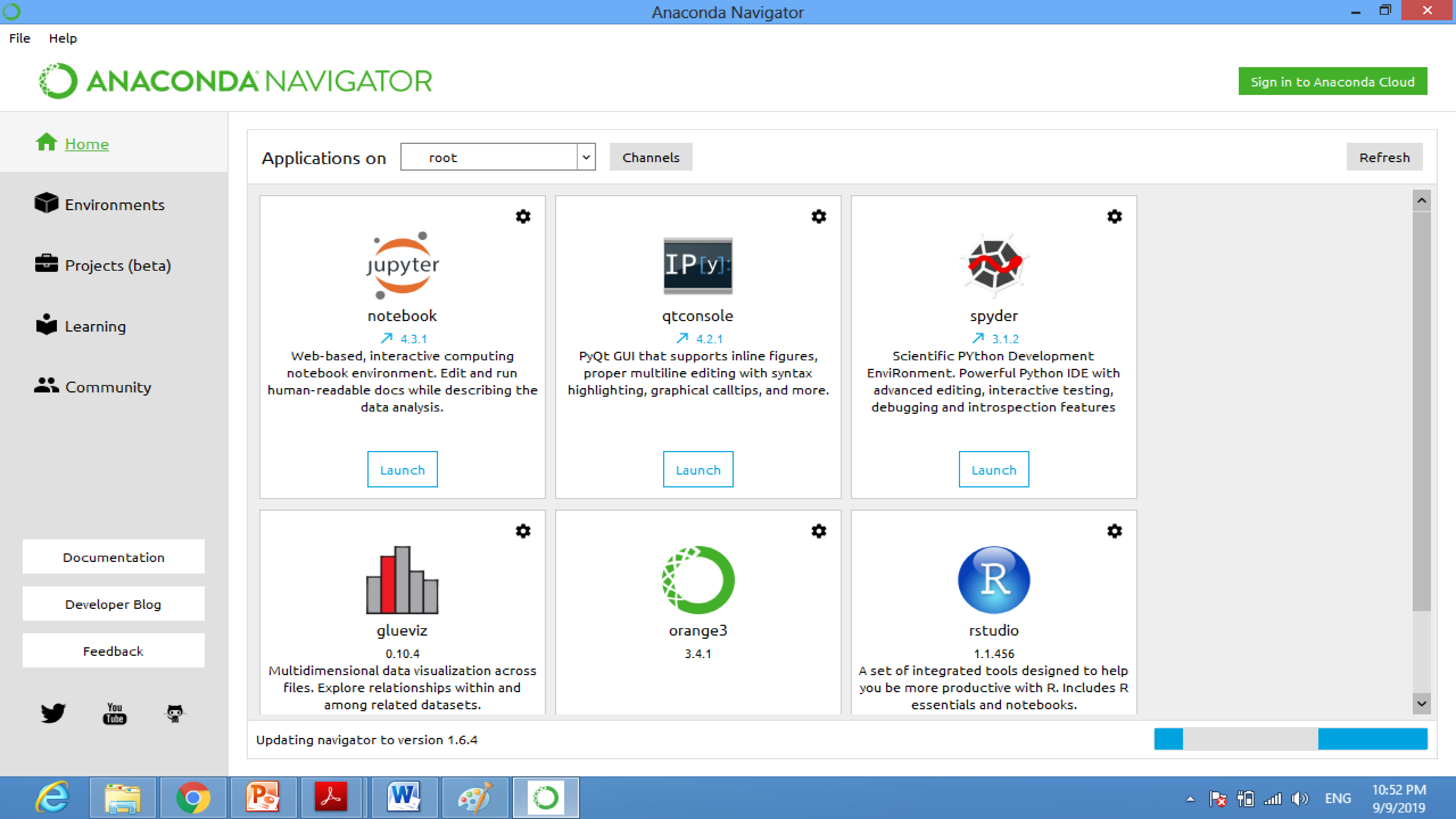
Not sensitive to extreme values.

Synonyms: resistant

7. Outlier

- A data value that is very different from most of the data
- An outlier is any value that is very distant from the other values in a data set

Synonyms: extreme value


Anaconda Navigator
File
Help
ANACONDA NAVIGATOR
Sign in to Anaconda Cloud
Home
Environments
Projects (beta)
Learning
Community
Documentation
Developer Blog
Feedback
Applications on
root
Channels
Refresh
jupyter
notebook
4.3.1
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
IP[y]:
qtconsole
4.2.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
spyder
3.1.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
glueviz
0.10.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.
orange3
3.4.1
R
rstudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Updating navigator to version 1.6.4
ENG
10:52 PM
9/9/2019

EXAMPLE: LOCATION ESTIMATES OF POPULATION AND MURDER RATES

Compute the mean, trimmed mean, and median for the population using R:

```
> state <- read.csv(file="/Users/andrewbruce1/book/state.csv")
> mean(state[["Population"]])
[1] 6162876
> mean(state[["Population"]], trim=0.1)
[1] 4783697
> median(state[["Population"]])
[1] 4436370
```

```
> weighted.mean(state[["Murder.Rate"]], w=state[["Population"]])
[1] 4.445834
> library("matrixStats")
> weightedMedian(state[["Murder.Rate"]], w=state[["Population"]])
[1] 4.4
```

	State	Population	Murder rate
1	Alabama	4,779,736	5.7
2	Alaska	710,231	5.6
3	Arizona	6,392,017	4.7
4	Arkansas	2,915,918	5.6
5	California	37,253,956	4.4
6	Colorado	5,029,196	2.8
7	Connecticut	3,574,097	2.4
8	Delaware	897,934	5.8

EXAMPLE: LOCATION ESTIMATES OF POPULATION AND MURDER RATES

Working with pandas in Python:

jupyter Muhammad Wasim - univariate_analysis_assignment_Solution Last Checkpoint: 01/24/2019 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Python 3

Code CellToolbar

In [35]:

```python
import pandas as pd
import os
print(os.path.abspath('../'))
# set the path where you have downloaded the data files
#file_path = os.path.abspath('../Projects/Project 1/')
file_path = os.path.abspath('C:/Users/Hussain Computer/Desktop/Tasks-KAI/Projects/Project 1')
print(file_path)
#print('C:\Users\Hussain Computer\Desktop\Tasks-KAI\Day 2')
data_df = pd.read_csv(file_path+'\\Muhammad Wasim - titanic1.csv')
#test = pd.read_csv(file_path+'\Day 2\\test_income_data_AAII.csv')
# For windows, if the above paths doesn't works
# import os
# f_path = os.path.join(*['C:', 'Users', 'user', 'Desktop', 'train_income_data_AAII.csv'])
# train = pd.read_csv(f_path)
# or maybe try following to reach to the path
# absolute_path = os.path.abspath(os.path.dirname('train_income_data_AAII.csv'))
```

```
C:\Users\Hussain Computer\Desktop\Tasks-KAI\Projects
C:\Users\Hussain Computer\Desktop\Tasks-KAI\Projects\Project 1
```

ESTIMATES OF VARIABILITY

Estimates of Variability

- Location is just one dimension in summarizing a feature. A second dimension, *variability*, also referred to as *dispersion*, measures whether the data values are tightly clustered or spread out.
- At the heart of statistics lies variability:
- Measuring it, reducing it, distinguishing random from real variability, identifying the various sources of real variability, and making decisions in the presence of it.

KEY TERMS FOR VARIABILITY METRICS

Deviations

The difference between the observed values and the estimate of location.

Synonyms: errors, residuals

Variance

The sum of squared deviations from the mean divided by $n - 1$ where n is the number of data values.

Synonyms: mean-squared-error

KEY TERMS FOR VARIABILITY METRICS

Standard deviation

The square root of the variance.

Synonyms: l2-norm, Euclidean norm

Mean absolute deviation

The mean of the absolute value of the deviations from the mean.

Synonyms: l1-norm, Manhattan norm

Median absolute deviation from the median

The median of the absolute value of the deviations from the median.

KEY TERM S FOR VARIAB ILITY M ETRICS

Range

The difference between the largest and the smallest value in a data set.

Order statistics

Metrics based on the data values sorted from smallest to biggest.

Synonyms: ranks

Percentile

The value such that *P* percent of the values take on this value or less and (100–P) percent take on this value or more.

Synonyms: quantile

Interquartile range

The difference between the 75th percentile and the 25th percentile.

Synonyms: IQR

KEY TERMS FOR VARIABILITY METRICS

- Just as there are different ways to measure location (mean, median, etc.) there are also different ways to measure variability.
- The most widely used estimates of variation are based on the differences, or *deviations*, between the estimate of location and the observed data. For a set of data {1, 4, 4}, the mean is 3 and the median is 4.
- The deviations from the mean are the differences: $1 – 3 = –2$, $4 – 3 = 1$, $4 – 3 = 1$.
- These deviations tell us how dispersed the data is around the central value.

KEY TERMS FOR VARIABILITY METRICS

Mean absolute deviation:

- The sum of the deviations from the mean is precisely zero. Instead, a simple approach is to take the average of the absolute values of the deviations from the mean.
- In the preceding example, the absolute value of the deviations is {2 1 1} and their average is (2 + 1 + 1) / 3 = 1.33.
- This is known as the *mean absolute deviation* and is computed with the formula:

$$\text{Mean absolution deviation} = \frac{\sum_{i=1}^{n} | x_i - \bar{x} |}{n}$$

KEY TERMS FOR VARIABILITY METRICS

The best-known estimates for variability are the *variance* and the *standard deviation*, which are based on squared deviations. The variance is an average of the squared deviations, and the standard deviation is the square root of the variance.

$$\text{Variance} = s^2 = \frac{\sum (x - \overline{x})^2}{n - 1}$$

$$\text{Standard deviation} = s = \sqrt{\text{Variance}}$$

KEY TERMS FOR VARIABILITY METRICS

- A robust estimate of variability is the *median absolute deviation from the median*

or MAD:

$$\text{Median absolute deviation} = \text{Median}\left(| x_1 - m |,\ | x_2 - m |,\ \ldots,\ | x_N - m | \right)$$

- where *m* is the median. Like the median, the MAD is not influenced by extreme values. It is also possible to compute a trimmed standard deviation analogous to the trimmed mean.

EXAMPLE: VARIABILITY ESTIMATES OF STATE POPULATION

```
> sd(state[["Population"]])
[1] 6848235
> IQR(state[["Population"]])
[1] 4847308
> mad(state[["Population"]])
[1] 3849870
```

	State	Population	Murder rate
1	Alabama	4,779,736	5.7
2	Alaska	710,231	5.6
3	Arizona	6,392,017	4.7
4	Arkansas	2,915,918	5.6
5	California	37,253,956	4.4
6	Colorado	5,029,196	2.8
7	Connecticut	3,574,097	2.4
8	Delaware	897,934	5.8

Working with pandas in Python: Data File: “titanic1.csv”

In [35]:
```
import pandas as pd
import os
print(os.path.abspath('../'))
# set the path where you have downloaded the data files
#file_path = os.path.abspath('../Projects/Project 1/')
file_path = os.path.abspath('C:/Users/Hussain Computer/Desktop/Tasks-KAI/Projects/Project 1')
print(file_path)
#print('C:\Users\Hussain Computer\Desktop\Tasks-KAI\Day 2')
data_df = pd.read_csv(file_path+'\\Muhammad Wasim - titanic1.csv')
#test = pd.read_csv(file_path+'\Day 2\\test_income_data_AAII.csv')
# For windows, if the above paths doesn't works
# import os
# f_path = os.path.join(*['C:', 'Users', 'user', 'Desktop', 'train_income_data_AAII.csv'])
# train = pd.read_csv(f_path)
# or maybe try following to reach to the path
# absolute_path = os.path.abspath(os.path.dirname('train_income_data_AAII.csv'))
```

```
C:\Users\Hussain Computer\Desktop\Tasks-KAI\Projects
C:\Users\Hussain Computer\Desktop\Tasks-KAI\Projects\Project 1
```

Working with pandas in Python: Data File: “titanic1.csv”

In [10]:
```
print(type())
data_df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
```

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [11]:
```
# Get dimensions of your data
data_df.shape
```

Out[11]: (891, 12)

Working with pandas in Python: Data File: “titanic1.csv”

```
In [12]: data_df.dtypes

Out[12]: PassengerId      int64
         Survived         int64
         Pclass           int64
         Name            object
         Sex             object
         Age            float64
         SibSp            int64
         Parch            int64
         Ticket          object
         Fare           float64
         Cabin           object
         Embarked        object
         dtype: object
```

```
In [13]: categorical_var = data_df.dtypes.loc[data_df.dtypes=='object'].index
         print(categorical_var)

         Index(['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked'], dtype='object')
```

```
In [14]: data_df[categorical_var].apply(lambda x:len(x.unique()))

Out[14]: Name        891
         Sex           2
         Ticket      681
         Cabin       148
         Embarked      4
         dtype: int64
```

Working with pandas in Python: Data File: “titanic1.csv”

```
In [15]: # Here we will use .replace which we discussed in class topic "Data Preprocessing"
         data_df['Survived'].replace({0:'Not Survived'}, inplace=True)

In [16]: data_df['Survived'].replace({1:'Survived'}, inplace=True)

In [17]: data_df.dtypes

Out[17]: PassengerId      int64
         Survived        object
         Pclass           int64
         Name            object
         Sex             object
         Age            float64
         SibSp            int64
         Parch            int64
         Ticket          object
         Fare           float64
         Cabin           object
         Embarked        object
         dtype: object

In [18]: data_df['Survived'].unique()

Out[18]: array(['Not Survived', 'Survived'], dtype=object)
```

Working with pandas in Python: Data File: “titanic1.csv”

```
In [19]: data_df['Survived'].unique()

Out[19]: array(['Not Survived', 'Survived'], dtype=object)

In [20]: data_df['Survived'].value_counts()

Out[20]: Not Survived    549
         Survived        342
         Name: Survived, dtype: int64

In [21]: data_df['Survived'].value_counts()/data_df.shape[0]

Out[21]: Not Survived    0.616162
         Survived        0.383838
         Name: Survived, dtype: float64

In [22]: # Here we observe that around 61.6 % are 'Not Survived' and around 38.3% are 'Survived'

In [23]: data_df.dtypes

Out[23]: PassengerId      int64
         Survived        object
         Pclass           int64
         Name            object
         Sex             object
         Age            float64
         SibSp            int64
         Parch            int64
         Ticket          object
         Fare           float64
         Cabin           object
         Embarked        object
```

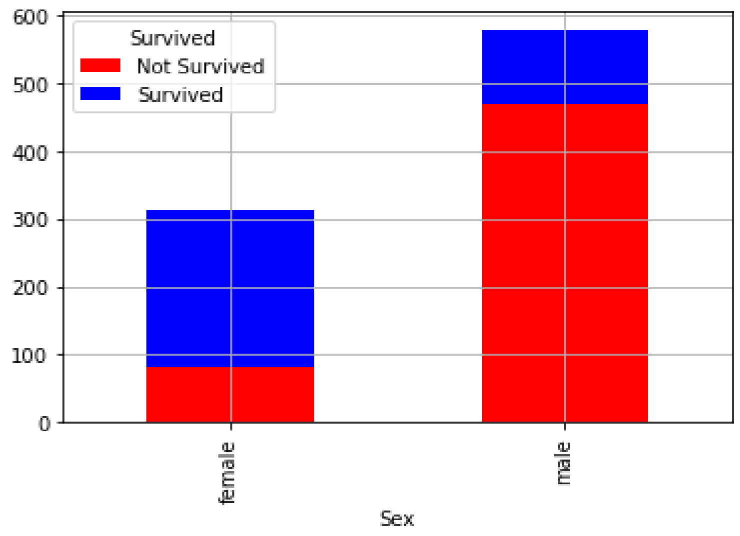
Working with pandas in Python: Data File: “titanic1.csv”

In [24]:
```
cross_tab = pd.crosstab(data_df['Sex'],data_df['Survived'],margins=True)
print(cross_tab)
```

```
Survived  Not Survived  Survived  All
Sex
female              81       233  314
male               468       109  577
All                549       342  891
```

In [25]:
```
%matplotlib inline
cross_tab.iloc[:-1,:-1].plot(kind='bar',stacked=True, color=['red','blue'], grid=True)
```

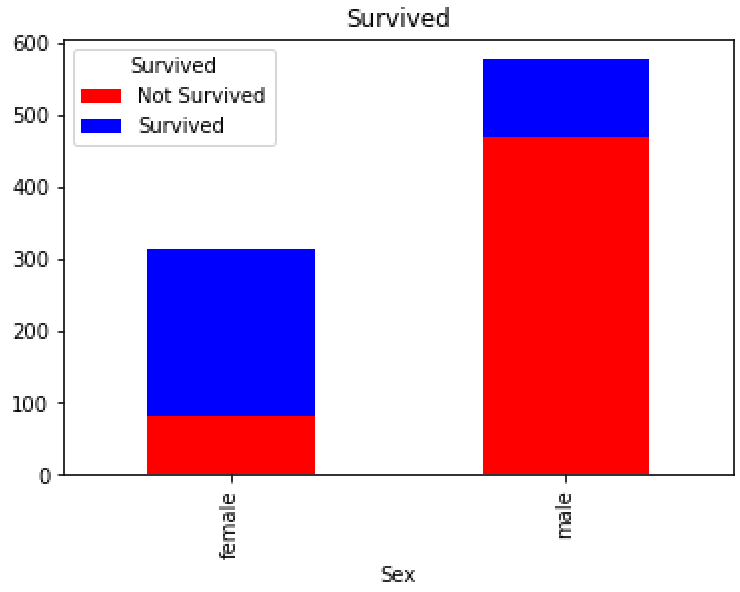
Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x579b1adf98>
```



Working with pandas in Python: Data File: “titanic1.csv”

```
In [26]: cross_tab.iloc[:-1,:-1].plot(kind='bar', stacked=True, color=['red','blue'], grid=False, title='Survived')
```

```
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x579d288b70>
```




```
In [28]: df=data_df['Survived'].value_counts()/data_df.shape[0]
```

```
Out[28]: Not Survived    0.616162
         Survived        0.383838
         Name: Survived, dtype: float64
```

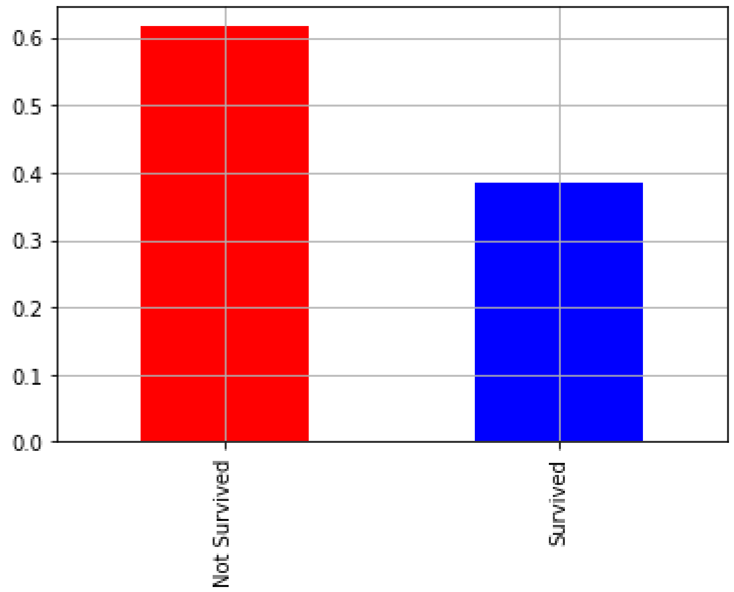
Working with pandas in Python: Data File: “titanic1.csv”

In [28]:
```
df=data_df['Survived'].value_counts()/data_df.shape[0]
```

Out[28]:
```
Not Survived    0.616162
Survived        0.383838
Name: Survived, dtype: float64
```

In [30]:
```
df=data_df['Survived'].value_counts()/data_df.shape[0]
%matplotlib inline
df.plot(kind='bar',stacked=True, color=['red','blue'], grid=True)
```

Out[30]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x579d3476d8>
```



Working with pandas in Python: Data File: “titanic1.csv”

In [ ]:
```
# Pclass int64 in continoeus data
```

In [31]:
```
data_df['Pclass'].value_counts()
```

Out[31]:
```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```

In [32]:
```
t=data_df['Pclass'].value_counts()/data_df.shape[0]
```

Out[32]:
```
3    0.551066
1    0.242424
2    0.206510
Name: Pclass, dtype: float64
```

In [34]:
```
t=data_df['Pclass'].value_counts()/data_df.shape[0]
t.plot(kind='bar',stacked=True, color=['red','blue','orange'], grid=True)
```

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x579d342358>