Lesson A-3

Getting to Know Your Data

- Data Exploration

CRISP-DM

- **Cross-industry standard process for data mining**
  - an open standard process model that describes common approaches used by data mining experts.
  - It is the most widely-used analytics model

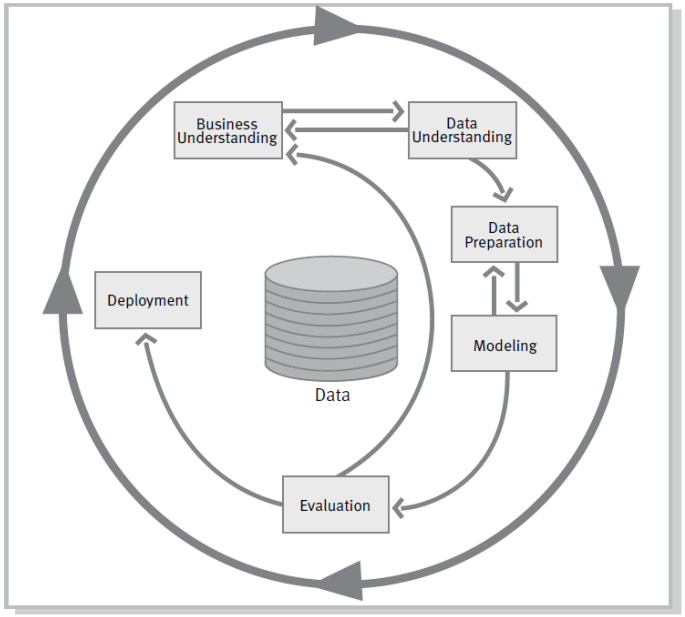
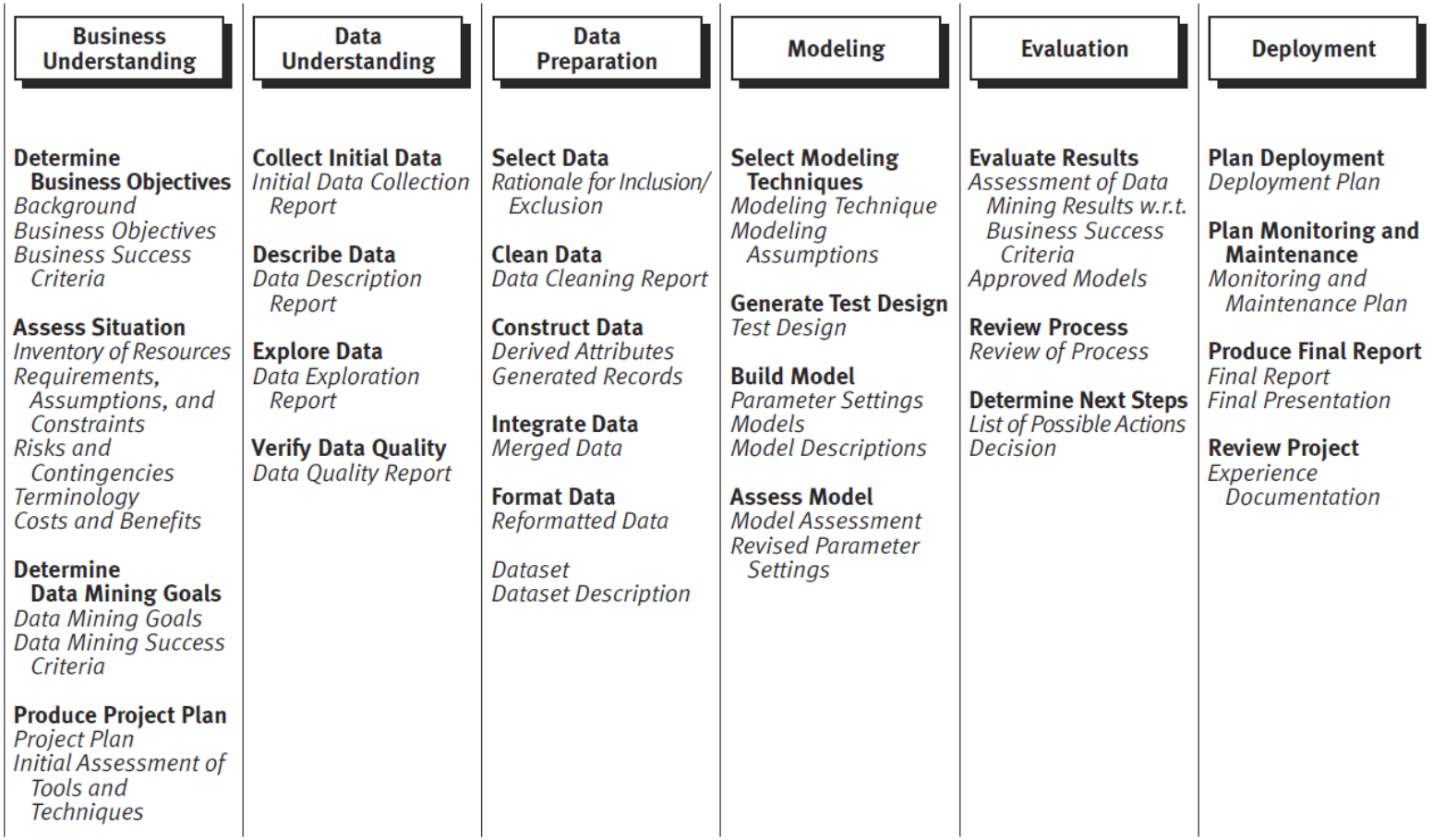


Figure 2: Phases of the CRISP-DM reference model

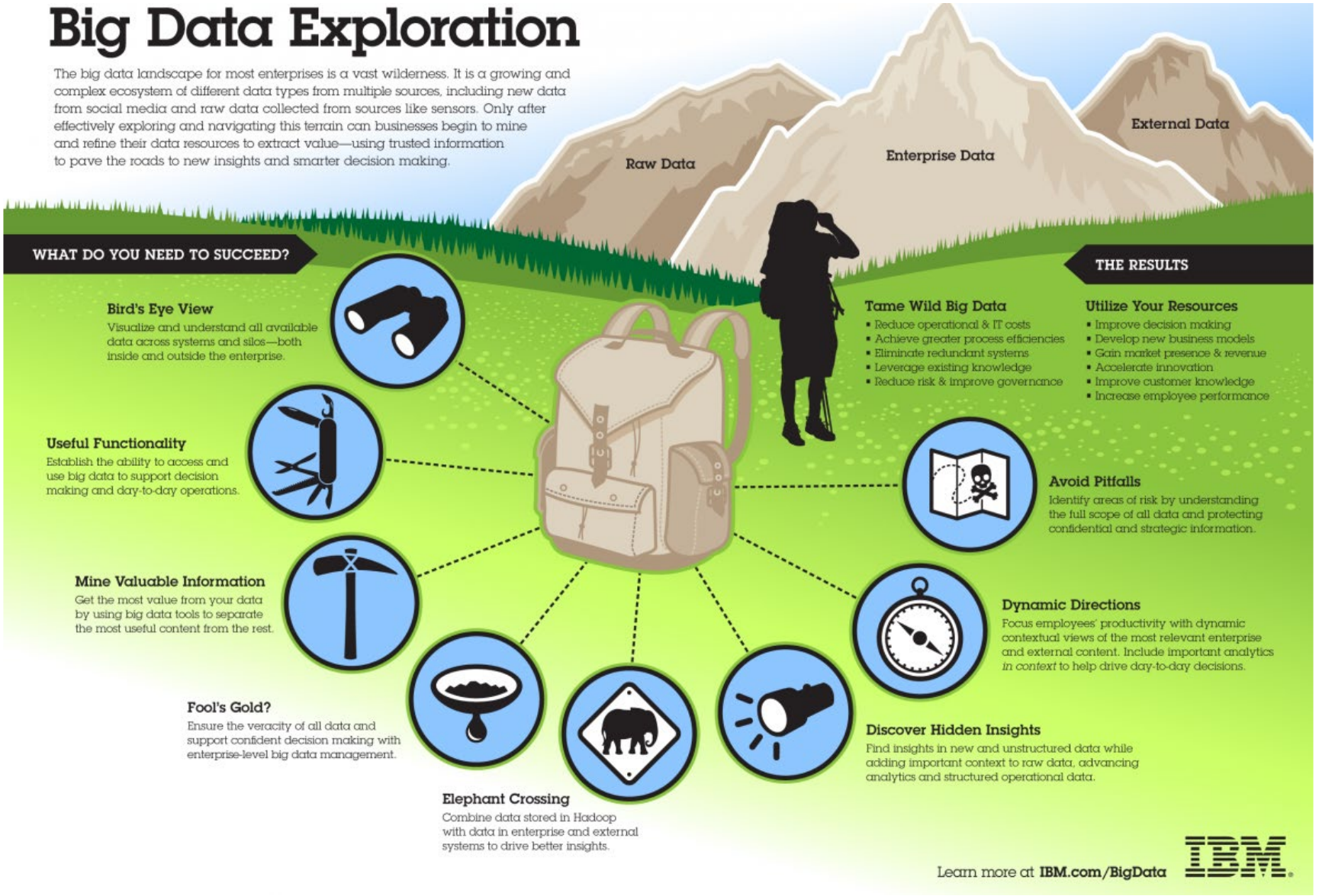

Business Understanding
Determine Business Objectives
Background
Business Objectives
Business Success Criteria
Assess Situation
Inventory of Resources
Requirements, Assumptions, and Constraints
Risks and Contingencies
Terminology
Costs and Benefits
Determine Data Mining Goals
Data Mining Goals
Data Mining Success Criteria
Produce Project Plan
Project Plan
Initial Assessment of Tools and Techniques
Data Understanding
Collect Initial Data
Initial Data Collection Report
Describe Data
Data Description Report
Explore Data
Data Exploration Report
Verify Data Quality
Data Quality Report
Data Preparation
Select Data
Rationale for Inclusion/ Exclusion
Clean Data
Data Cleaning Report
Construct Data
Derived Attributes
Generated Records
Integrate Data
Merged Data
Format Data
Reformatted Data
Dataset
Dataset Description
Modeling
Select Modeling Techniques
Modeling Technique
Modeling Assumptions
Generate Test Design
Test Design
Build Model
Parameter Settings
Models
Model Descriptions
Assess Model
Model Assessment
Revised Parameter Settings
Evaluation
Evaluate Results
Assessment of Data Mining Results w.r.t. Business Success Criteria
Approved Models
Review Process
Review of Process
Determine Next Steps
List of Possible Actions
Decision
Deployment
Plan Deployment
Deployment Plan
Plan Monitoring and Maintenance
Monitoring and Maintenance Plan
Produce Final Report
Final Report
Final Presentation
Review Project
Experience Documentation

What is Data Exploration

- *The purpose of exploratory analysis is to **"get to know"** the dataset.*
- Data exploration addresses data mining projects using statistics and visualization to investigate each type of attributes of a dataset

Big Data Exploration

The big data landscape for most enterprises is a vast wilderness. It is a growing and complex ecosystem of different data types from multiple sources, including new data from social media and raw data collected from sources like sensors. Only after effectively exploring and navigating this terrain can businesses begin to mine and refine their data resources to extract value—using trusted information to pave the roads to new insights and smarter decision making.



Learn more at **IBM.com/BigData**

IBM

Understand data

- Load dataset
- View dataset
- Shape of dataset
- Attribute types
- Statistical Techniques
  - Data summary based on statistical description
  - attribute skewness
  - Class distribution
  - Correlations between attributes
- Data visualization
  - bar charts, histograms, frequency polygons, pie charts, scatter plot , heatmap

Load and view the dataset

- pandas.DataFrame
  - Two-dimensional, size-mutable, potentially heterogeneous tabular data, labeled axes (rows and columns)

]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

Shape of data set and attribute type

```
data.shape # shape of dataframe (number of rows and columns)
```

```
: (768, 9)
```

```
data.dtypes
```

```
]: Pregnancies                 int64
   Glucose                     int64
   BloodPressure               int64
   SkinThickness               int64
   Insulin                     int64
   BMI                       float64
   DiabetesPedigreeFunction  float64
   Age                         int64
   Outcome                     int64
   dtype: object
```

Shape of data set and attribute type

```
data.info() # summary of dataframe
            # number rows,  name of feature,  number of non-null items, type of a feature
            # in each column
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnancies                 768 non-null int64
Glucose                     768 non-null int64
BloodPressure               768 non-null int64
SkinThickness               768 non-null int64
Insulin                     768 non-null int64
BMI                         768 non-null float64
DiabetesPedigreeFunction    768 non-null float64
Age                         768 non-null int64
Outcome                     768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Type of Attribute

- Categorical (Qualitative) data
  - Are categorical in nature. They describe the quality of something (someone)
    - Nominal data
    - Ordinal data
- Numeric (Quantitative) data
  - Are numerical in nature. They measure the quantity of something (someone)
    - Interval data
    - Ratio data

	attribute type	properties	examples	descriptive statistics	graph
categorical (qualitative)	**nominal**	Names, information to distinguish (compare) one object from another (= , $\neq$)	Names of employee, zip codes, employee ID, eye color, gender	frequency, percentage, mode	Bar Pie
	ordinal	The value of an ordinal attribute provide enough information to order objects ($<$, $>$)	GPA, professor rank, Likert scales,	frequency, percentage, mode, median	Bar Pie Boxplot

	attribute type	properties	examples	descriptive statistics	graph
numeric (quantitative)	**interval**	Differences between values are meaningful. (+, -)	calendar date, temperature in Celsius or Fahrenheit	frequency, percentage, mode mean median standard deviation	Bar Pie Box plot Density plot Histogram
	ratio	True 0 allows ratio statements (*, /)	temperature in Kelvin, time, money, counts, age, weight, length, electrical current,	frequency, percentage, mode mean median standard deviation	Bar Pie Boxplot Histogram Density plot

Data summary based on statistical description

`count`	Number of non-NA values
`describe`	Compute set of summary statistics for Series or each DataFrame column
`min, max`	Compute minimum and maximum values
`argmin, argmax`	Compute index locations (integers) at which minimum or maximum value obtained, respectively
`idxmin, idxmax`	Compute index labels at which minimum or maximum value obtained, respectively
`quantile`	Compute sample quantile ranging from 0 to 1
`sum`	Sum of values
`mean`	Mean of values
`median`	Arithmetic median (50% quantile) of values
`mad`	Mean absolute deviation from mean value
`prod`	Product of all values
`var`	Sample variance of values
`std`	Sample standard deviation of values
`skew`	Sample skewness (third moment) of values
`kurt`	Sample kurtosis (fourth moment) of values
`cumsum`	Cumulative sum of values
`cummin, cummax`	Cumulative minimum or maximum of values, respectively
`cumprod`	Cumulative product of values
`diff`	Compute first arithmetic difference (useful for time series)
`pct_change`	Compute percent changes

Pandas `describe()` function

- The Pandas `describe()` function lists 8 statistical properties of each “numeric” attribute
  - Count
  - Mean
  - Standard deviation
  - Minimum value
  - 25th percentile
  - 50th percentile (Median)
  - 75th percentile
  - Maximum value

```
data.describe()
```

]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

```
from pandas import set_option
set_option('precision', 3) # Displays precision for decimal numbers
data.describe()
```

]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000	768.000	768.000	768.000	768.000	768.000	768.000	768.000	768.000
mean	3.845	120.895	69.105	20.536	79.799	31.993	0.472	33.241	0.349
std	3.370	31.973	19.356	15.952	115.244	7.884	0.331	11.760	0.477
min	0.000	0.000	0.000	0.000	0.000	0.000	0.078	21.000	0.000
25%	1.000	99.000	62.000	0.000	0.000	27.300	0.244	24.000	0.000
50%	3.000	117.000	72.000	23.000	30.500	32.000	0.372	29.000	0.000
75%	6.000	140.250	80.000	32.000	127.250	36.600	0.626	41.000	1.000
max	17.000	199.000	122.000	99.000	846.000	67.100	2.420	81.000	1.000

	attribute type	properties	examples	descriptive statistics	graph
categorical (qualitative)	**nominal**	Names, information to distinguish (compare) one object from another (= , $\neq$)	Names of employee, zip codes, employee ID, eye color, gender	frequency, percentage, mode	Bar Pie
	ordinal	The value of an ordinal attribute provide enough information to order objects ($<$, $>$)	GPA, professor rank, Likert scales,	frequency, percentage, mode, median	Bar Pie Boxplot

	attribute type	**properties**	**examples**	**descriptive statistics**	**graph**
numeric (quantitative)	**interval**	Differences between values are meaningful. (+, -)	calendar date, temperature in Celsius or Fahrenheit	frequency, percentage, mode mean median standard deviation	Bar Pie Box plot Density plot Histogram
	ratio	True 0 allows ratio statements (*, /)	temperature in Kelvin, time, money, counts, age, weight, length, electrical current,	frequency, percentage, mode mean median standard deviation	Bar Pie Boxplot Histogram Density plot

count

- Count
  - Count non-NA cells for each column or row.

```
# count
data.count()
```

```
4]: Pregnancies                768
    Glucose                    768
    BloodPressure              768
    SkinThickness              768
    Insulin                    768
    BMI                        768
    DiabetesPedigreeFunction   768
    Age                        768
    Outcome                    768
    dtype: int64
```

Measuring the Central Tendency - Arithmetic Mean

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{x_1 + x_2 + \cdots + x_N}{N}.$$

$x_1, x_2, \ldots, x_N$ is a set of N values

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

the mean salary is _____

weighted arithmetic mean or the **weighted average**
Each value *xi* in a set may be associated with a weight w_i for $i = 1, \ldots, N$.

$$\bar{x} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_N x_N}{w_1 + w_2 + \cdots + w_N}.$$

Measuring the Dispersion of Data - Range

- *Let* $x_1, x_2, \ldots, x_N$ be a set of observations for some numeric attribute, **X**
- The range of the set is the difference between the largest value (max) and smallest (min) values

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

the range of the salary is _____

Measuring the Central Tendency – Midrange

- *Let* $x_1, x_2, \ldots, x_N$ be a set of observations for some numeric attribute, X
- It is the average of the largest and smallest values in the dataset.
- This measure is easy to computing using aggregating functions, `max()` and `min()`

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

the midrange of the salary is _____

Measuring the Central Tendency - Median

Median = the middle value of a set of ordered data.

Median = $\{(n + 1) \div 2\}^{th}$ value

n is the number of values

If *n* is an even number, the median is calculated by averaging the two middle values.

Median = (value below median + value above median) ÷ 2

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

the median salary is _____

```
# mean
data.mean()
```

```
2]: Pregnancies                 3.845
    Glucose                   120.895
    BloodPressure              69.105
    SkinThickness              20.536
    Insulin                    79.799
    BMI                        31.993
    DiabetesPedigreeFunction    0.472
    Age                        33.241
    Outcome                     0.349
    dtype: float64
```

```
# median
data.median()
```

```
1]: Pregnancies                 3.000
    Glucose                   117.000
    BloodPressure              72.000
    SkinThickness              23.000
    Insulin                    30.500
    BMI                        32.000
    DiabetesPedigreeFunction    0.372
    Age                        29.000
    Outcome                     0.000
    dtype: float64
```

```
# max
data.max()
```

```
Pregnancies                 17.00
Glucose                    199.00
BloodPressure              122.00
SkinThickness               99.00
Insulin                    846.00
BMI                         67.10
DiabetesPedigreeFunction     2.42
Age                         81.00
Outcome                      1.00
dtype: float64
```

```
# min
data.min()
```

```
Pregnancies                 0.000
Glucose                     0.000
BloodPressure               0.000
SkinThickness               0.000
Insulin                     0.000
BMI                         0.000
DiabetesPedigreeFunction    0.078
Age                        21.000
Outcome                     0.000
dtype: float64
```

```
# range
data.max() - data.min()
```

```
: Pregnancies                 17.000
  Glucose                    199.000
  BloodPressure              122.000
  SkinThickness               99.000
  Insulin                    846.000
  BMI                         67.100
  DiabetesPedigreeFunction     2.342
  Age                         60.000
  Outcome                      1.000
  dtype: float64
```

```
# mid-range
(data.max() - data.min())/2
```

```
: Pregnancies                 8.500
  Glucose                    99.500
  BloodPressure              61.000
  SkinThickness              49.500
  Insulin                   423.000
  BMI                        33.550
  DiabetesPedigreeFunction    1.171
  Age                        30.000
  Outcome                     0.500
  dtype: float64
```

Measuring the Central Tendency – Mode

- The mode is the value that occurs most frequently in the dataset.
- It can be determined for qualitative and quantitative attributes.
- Data sets with one, two, or three modes are respectively called unimodal, bimodal, and trimodal.
- In general, a data set with two or more modes is multimodal.

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

the mode(s) of the salary is (are) _____

```
# mode
data.mode()
```

:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	1.0	99	70.0	0.0	0.0	32.0	0.254	22.0	0.0
1	NaN	100	NaN	NaN	NaN	NaN	0.258	NaN	NaN

Measuring the Dispersion of Data - Quantiles

- Suppose that the data are sorted ascendingly.
- q-**quantiles** are cutting points that partition a finite set of values into q subsets of (nearly) equal sizes. There are $q - 1$ cutting points for the q-subsets
- There is one fewer quantile than the number of subset created.
- The 2-quantiles is the data point dividing the lower and upper halves of the data distribution. It corresponds to the median.

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: {30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110}.

What is (are) quantile(s) of the 2-quantiles of this dataset?

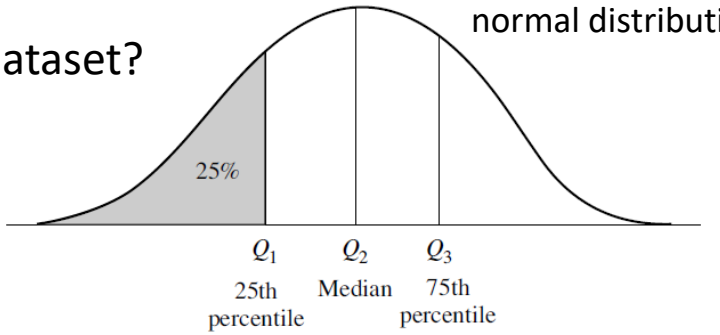
Measuring the Dispersion of Data - Quartiles

- The 4-quantiles are the three data points that split the data distribution into four (nearly) equal parts; each part represents one-fourth of the data distribution.
- The 4-quantiles are referred to as **quartiles**.

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: {30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110}.

What is (are) quantile(s) of the 4-quantiles of this dataset?

Quartiles of normal distribution



- The 100-quantiles are more commonly referred to as **percentiles**; they divide the data dataset into 100 equal-sized consecutive subsets.

- The median, quartiles, and percentiles are the most widely used forms of quantiles.

Measuring the Dispersion of Data - Interquartile Range (IQR)

- IQR is a measure of statistical dispersion and is the distance between the first and third quartiles. It is defined as

$$IQR = Q3 - Q1$$

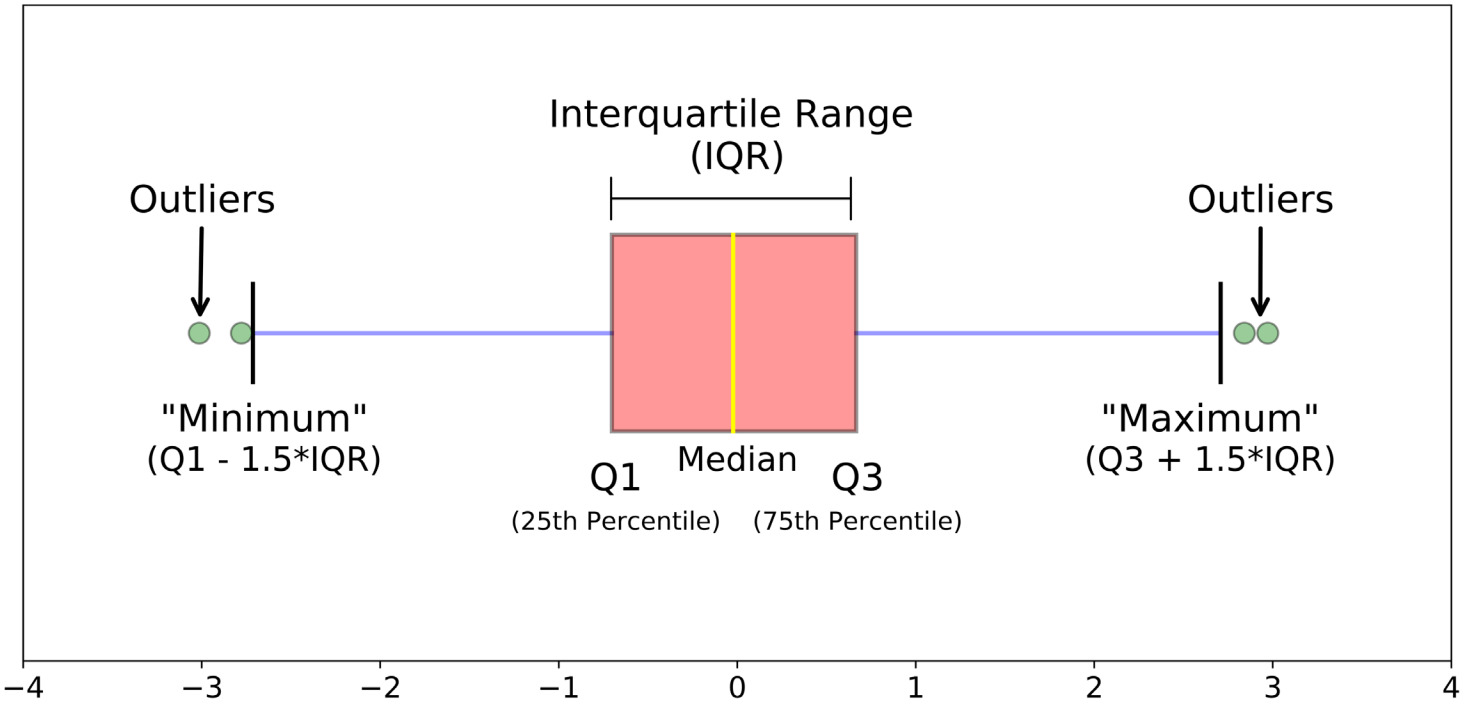
- Using IQR, we can identify suspected outliers that are values falling in at least 1.5 x IQR above Q3 and below Q1.

- Boxplot is a popular way to visualize a distribution based on minimum, Q1, median (Q2), Q3 and maximum

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: {30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110}.

What are the IQR of this dataset?

Boxplot


Interquartile Range
(IQR)
Outliers
Outliers
"Minimum"
(Q1 - 1.5*IQR)
Q1
Median
Q3
(25th Percentile)
(75th Percentile)
"Maximum"
(Q3 + 1.5*IQR)
−4
−3
−2
−1
0
1
2
3
4

```
# IQR
Q1 = data.quantile(0.25)
Q3 = data.quantile(0.75)
IQR = Q3 - Q1
IQR
```

```
Pregnancies                   5.000
Glucose                      41.250
BloodPressure                18.000
SkinThickness                32.000
Insulin                     127.250
BMI                           9.300
DiabetesPedigreeFunction      0.382
Age                          17.000
Outcome                       1.000
dtype: float64
```

Measuring the Dispersion of Data – variance and standard deviation

- **Variance and standard deviation** are measures of data dispersion and they indicate how spread out a data distribution is.

- A low standard deviation means that the data tend to be very close to the mean, while a high standard deviation indicates that the data are spread out over a large range of values.

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 = \left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right) - \bar{x}^2$$

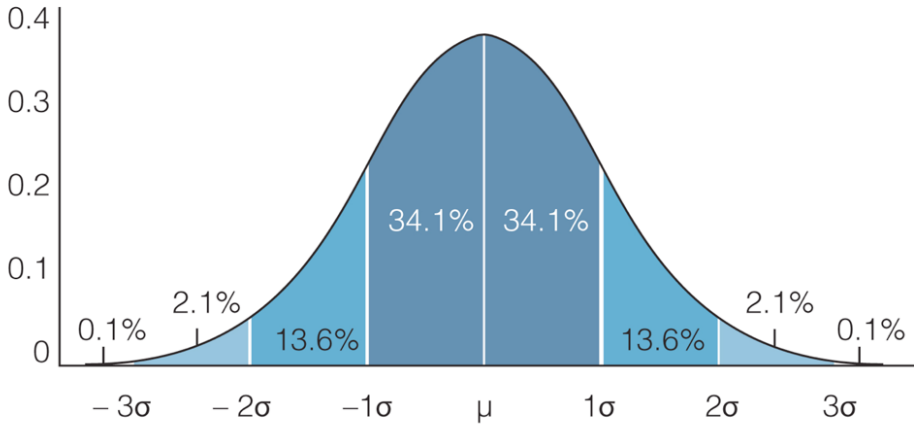
  - σ^2 is the **variance** of N values, $x_1, x_2, \ldots, x_N$,
  - $\overline{x}$ *is the mean of values*
  - The **standard deviation**, σ, is the square root of the variance, σ^2

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

What are the variance and standard deviation of this dataset?

Measuring the Dispersion of Data – standard deviation

- The basic properties of the standard deviation, as a measure of spread are as follows:
- σ measures spread about the mean and should be considered only when the mean is chosen as the measure of center.
- $\sigma = 0$ only when there is no spread, that is, when all observations have the same value. Otherwise, $\sigma > 0$ or $\sigma < 0$.



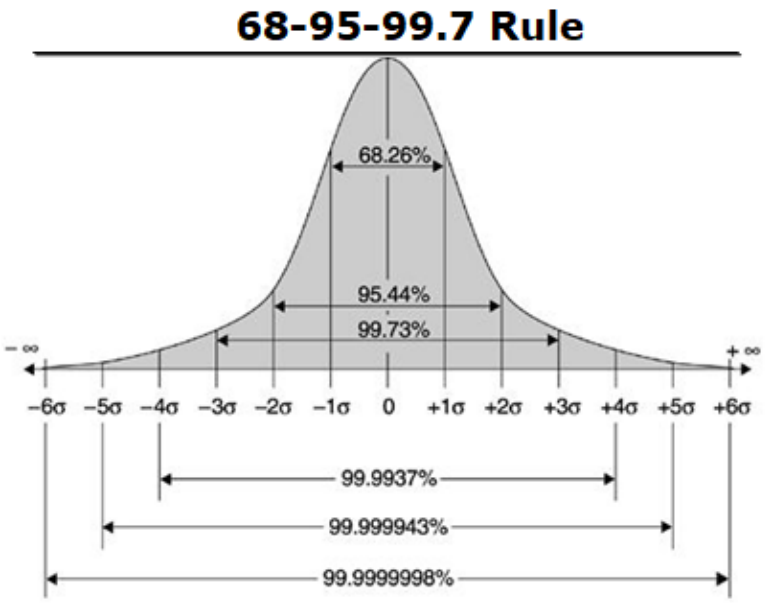
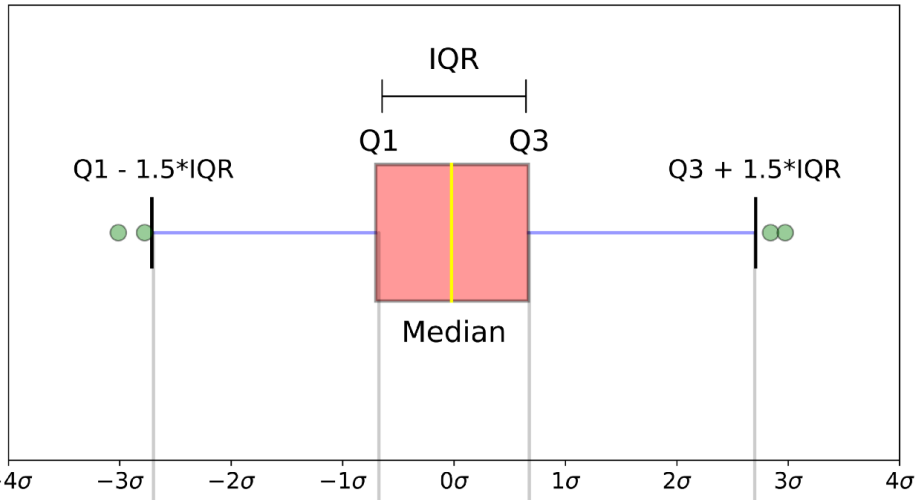
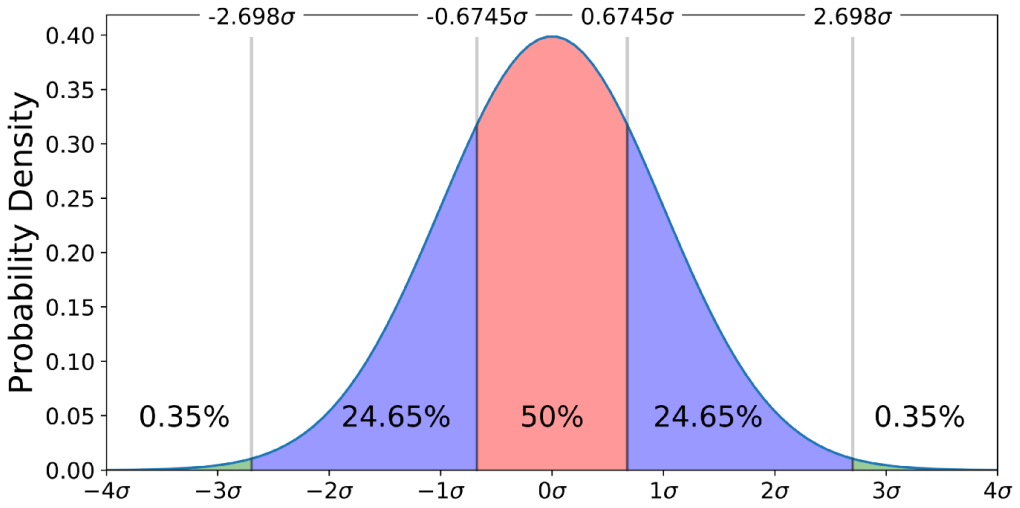
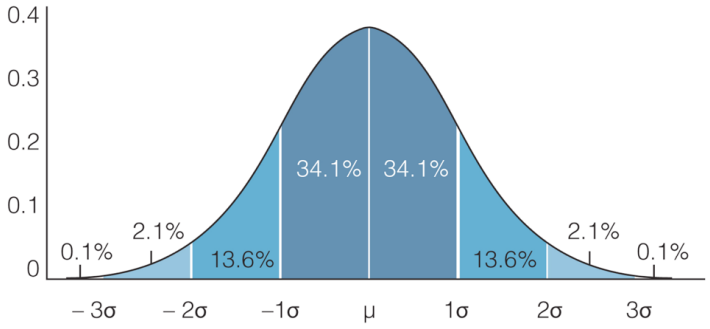


Image: Google

Boxplot on Standard Normal Distribution

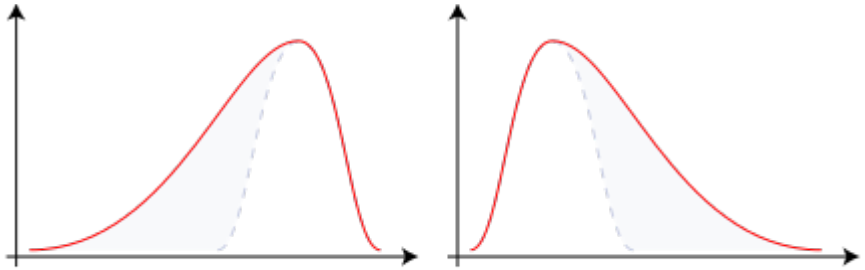


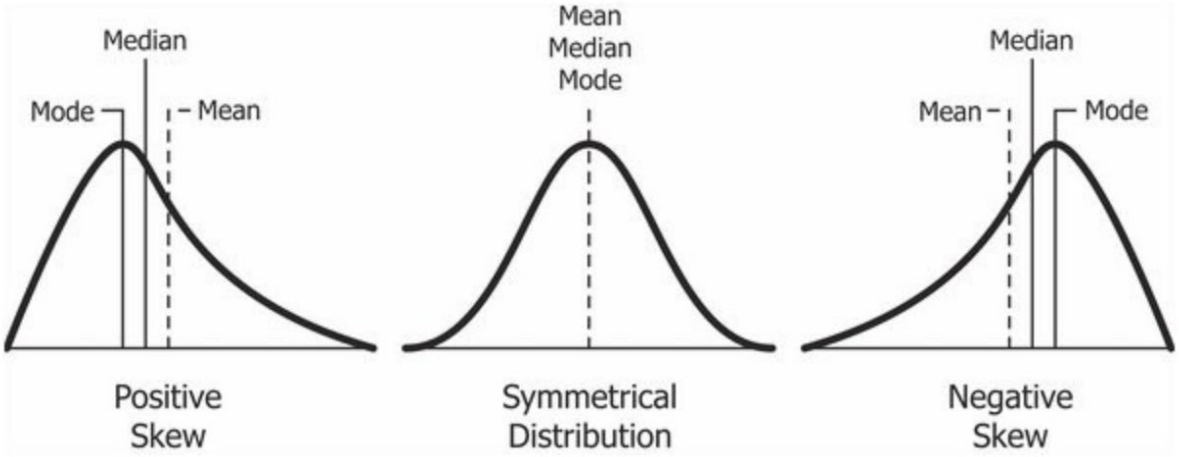




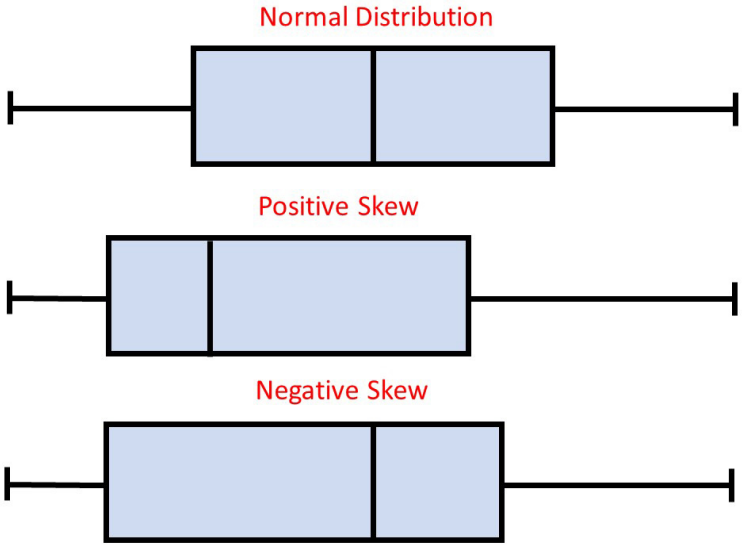
Skewness

- Skewness is a measure of asymmetry of a distribution
  - In a normal (Gaussian or bell curve) distribution, the mean divides the curve symmetrically into two equal parts at the median and the value of skewness is zero.
- When a distribution is asymmetrical the tail of the distribution is skewed to one side to the right or to the left.
  - When the value of the skewness is negative, the tail of the distribution is longer towards the left-hand side of the curve.
  - When the value of the skewness is positive, the tail of the distribution is longer towards the right-hand side of the curve





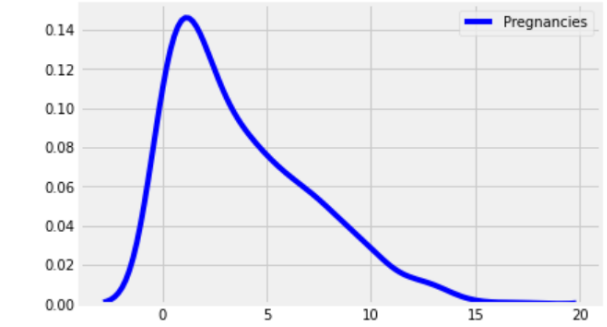
source: Wikipedia



Pearson Mode Skewness: skew = 3 * (Mean – Median) / Standard Deviation

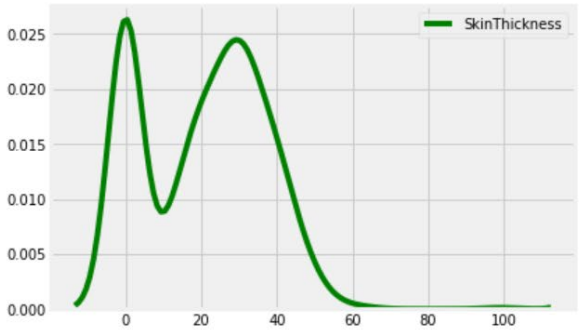
```
sns.kdeplot(data['Pregnancies'], color='b')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x18becc82f08>
```



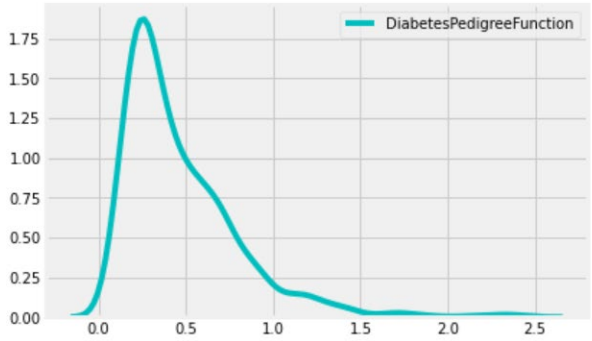

```
sns.kdeplot(data['SkinThickness'], color="g")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x18bedd12ac8>
```



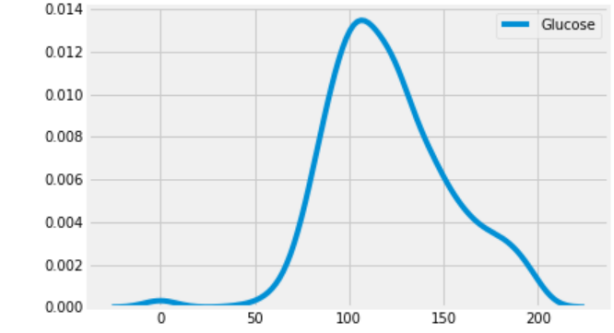

```
sns.kdeplot(data['DiabetesPedigreeFunction'], color="c")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x18bedeebec8>
```



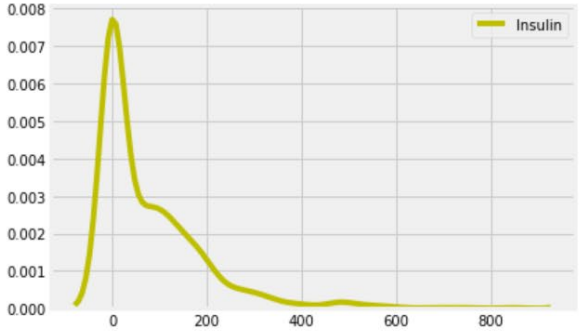

```
sns.kdeplot(data['Glucose'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x18bec872f48>
```



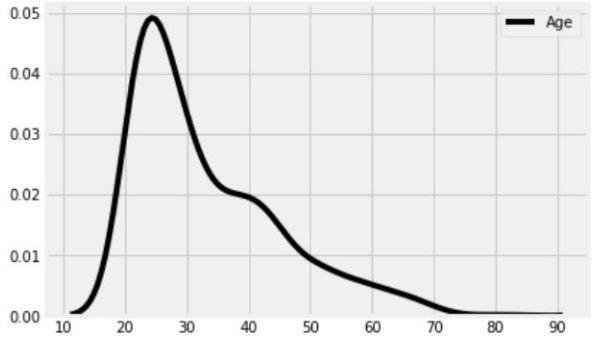

```
sns.kdeplot(data['Insulin'], color="y")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x18bedd97f88>
```



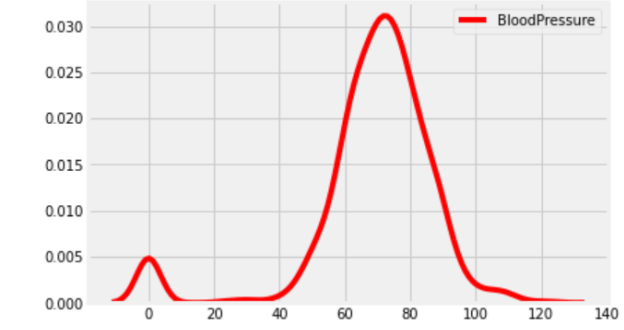

```
sns.kdeplot(data['Age'], color="k")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x18bee21b688>
```



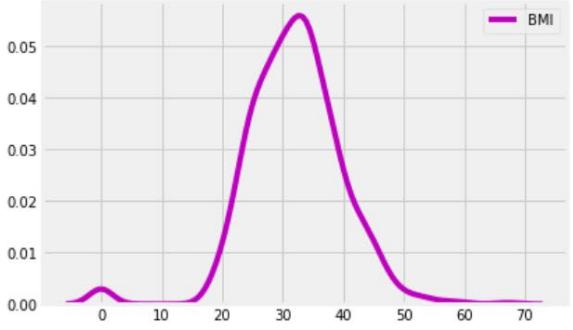

```
sns.kdeplot(data['BloodPressure'], color="r")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x18bedcc5488>
```




```
sns.kdeplot(data['BMI'], color="m")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x18bec59b888>
```



Pregnancies	0.902
Glucose	0.174
BloodPressure	-1.844
SkinThickness	0.109
Insulin	2.272
BMI	-0.429
DiabetesPedigreeFunction	1.920
Age	1.130

t[7]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000	768.000	768.000	768.000	768.000	768.000	768.000	768.000	768.000
mean	3.845	120.895	69.105	20.536	79.799	31.993	0.472	33.241	0.349
std	3.370	31.973	19.356	15.952	115.244	7.884	0.331	11.760	0.477
min	0.000	0.000	0.000	0.000	0.000	0.000	0.078	21.000	0.000
25%	1.000	99.000	62.000	0.000	0.000	27.300	0.244	24.000	0.000
50%	3.000	117.000	72.000	23.000	30.500	32.000	0.372	29.000	0.000
75%	6.000	140.250	80.000	32.000	127.250	36.600	0.626	41.000	1.000
max	17.000	199.000	122.000	99.000	846.000	67.100	2.420	81.000	1.000

```
# mode
data.mode()
```

:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	1.0	99	70.0	0.0	0.0	32.0	0.254	22.0	0.0
1	NaN	100	NaN	NaN	NaN	NaN	0.258	NaN	NaN

```
# skew
skew = data.skew() # calculate the skew of each feature using the skew() function on the DataFrame
skew
```

:

```
Pregnancies                0.902
Glucose                    0.174
BloodPressure             -1.844
SkinThickness              0.109
Insulin                    2.272
BMI                       -0.429
DiabetesPedigreeFunction   1.920
Age                        1.130
Outcome                    0.635
dtype: float64
```

Skewness

- If skewness is less than -1 or greater than 1, the distribution is highly skewed.
- If skewness is between -1 and -0.5 or between 0.5 and 1, the distribution is moderately skewed.
- If skewness is between -0.5 and 0.5, the distribution is approximately symmetric.

Class Distribution

- We need to know how balanced the class values are.
- Class imbalance problems are common
  - This results in models that have poor predictive performance, specifically for the minority class.
    - typically, the minority class is more important and therefore the problem is more sensitive to classification errors for the minority class than the majority class.
    - Medical diagnosis, fraud detection, claim prediction, spam detection, anomaly detection, outlier detection, intrusion detection
- We need special handling in data preprocessing and model evaluation as well

Metrics for Evaluating Classifier

Measure	*Formula*
accuracy, recognition rate	$\frac{TP+TN}{P+N}$
error rate, misclassification rate	$\frac{FP+FN}{P+N}$
sensitivity, true positive rate, recall	$\frac{TP}{P}$
specificity, true negative rate	$\frac{TN}{N}$
precision	$\frac{TP}{TP+FP}$
F, F_1, F-score, harmonic mean of precision and recall	$\frac{2 \times precision \times recall}{precision + recall}$
F_β, where β is a non-negative real number	$\frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P + N$

Confusion matrix, shown with totals for positive and negative tuples.

Classes	*buys_computer = yes*	*buys_computer = no*	*Total*	*Recognition (%)*
buys_computer = yes	**6954**	**46**	7000	99.34
buys_computer = no	**412**	**2588**	3000	86.27
Total	7366	2634	10,000	95.42

Confusion matrix for the classes buys computer = yes and buys computer = no

Classes	*yes*	*no*	*Total*	*Recognition (%)*
yes	**90**	**210**	300	30.00
no	**140**	**9560**	9700	98.56
Total	230	9770	10,000	96.40

Confusion matrix for medical data where the class values are *yes* and *no* for a class label attribute, *cancer*.

Class Distribution

- **Slight Imbalance**. An imbalanced classification problem where the distribution of instances is uneven by a small amount in the training dataset (e.g. 1:4 or less)
- **Severe Imbalance**. An imbalanced classification problem where the distribution of instances is uneven by a large amount in the training dataset (e.g. 1:100 or more)

- *"Most of the contemporary works in class imbalance concentrate on imbalance ratios ranging from 1:4 up to 1:100. …. In real-life applications such as fraud detection or cheminformatics we may deal with problems with imbalance ratio ranging from 1:1000 up to 1:5000."*

Learning from imbalanced data: open challenges and future directions, *Progress in AI*, Bartosz Krawczyk

```
# load in the data set
data = pd.read_csv("diabetes.csv") # load the CSV file using pandas.read_csv function
                                   # The function returns a pandas.DataFrame
```

```
# class distribution
class_counts = data.groupby('Outcome').size()
class_counts
```

```
Outcome
0    500
1    268
dtype: int64
```

```
class_counts[0]/data['Outcome'].size
```

```
0.6510416666666666
```

```
class_counts[1]/data['Outcome'].size
```

```
0.3489583333333333
```

Correlations between attributes

- Correlation refers to the relationship between two attributes and how they related in terms of change.
  - Correlation analysis is used to detect attribute redundancy and improve performance of model
  - The performance of some machine learning models like linear and logistic regression may not be good if there are highly correlated attributes in the dataset.
  - We need to review all of the pairwise correlations of the attribute in the dataset.

Correlation analysis

- Correlation analysis
  - Measure how strongly one attribute is related (dependent) with the other
  - Numeric data
    - Covariance
    - Pearson's Correlation Coefficient
    - `corr()` Pandas function
  - Categorical data
    - Chi-square test
    - `chi2_contingency()` SciPy function
    - `from sklearn.attribute_selection import chi2`

Covariance between attributes

Covariance measures the strength and the direction of the relationship between the observations of two attributes and the correlation is derived from the covariance.

The sample covariance between two variables, X and Y, is

$$\mathrm{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \,, \quad \bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

	X	Y
1	**1**	**3**
2	**-2**	**2**
3	**3**	**4**
4	**0**	**6**
5	**3**	**0**

Cov(X, Y) =

Covariance between attributes

- **Positive covariance**: If $cov(X, Y) > 0$, then X and Y both tend to be larger than their expected values.
- **Negative covariance**: If $cov(X, Y) < 0$ then if X is larger than its expected value, Y is likely to be smaller than its expected value.
- **Independence**: $cov(X, Y) = 0$

```
# Covariance between features
set_option('precision', 3) # Displays precision for decimal numbers
covariance = data.cov()
covariance
```

1]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	11.354	13.947	9.215	-4.390	-28.555	0.470	-0.037	21.571	0.357
Glucose	13.947	1022.248	94.431	29.239	1220.936	55.727	1.455	99.083	7.115
BloodPressure	9.215	94.431	374.647	64.029	198.378	43.005	0.265	54.523	0.601
SkinThickness	-4.390	29.239	64.029	254.473	802.980	49.374	0.972	-21.381	0.569
Insulin	-28.555	1220.936	198.378	802.980	13281.180	179.775	7.067	-57.143	7.176
BMI	0.470	55.727	43.005	49.374	179.775	62.160	0.367	3.360	1.101
DiabetesPedigreeFunction	-0.037	1.455	0.265	0.972	7.067	0.367	0.110	0.131	0.027
Age	21.571	99.083	54.523	-21.381	-57.143	3.360	0.131	138.303	1.337
Outcome	0.357	7.115	0.601	0.569	7.176	1.101	0.027	1.337	0.227

Pearson Correlation Coefficient between attributes

The correlation is derived from the covariance.

$$corr(X,Y) = r_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$$

- $r_{X,Y}$ can take a range of values from +1 to -1
- If $r_{X,Y} > 0$: X and Y are positively correlated (X's values increase as Y's).
- The higher, the stronger correlation.
- $r_{X,Y} = 0$: independent;
- $r_{X,Y} < 0$: negatively correlated

corr(X, Y) =

	X	Y
1	**1**	**3**
2	**-2**	**2**
3	**3**	**4**
4	**0**	**6**
5	**3**	**0**

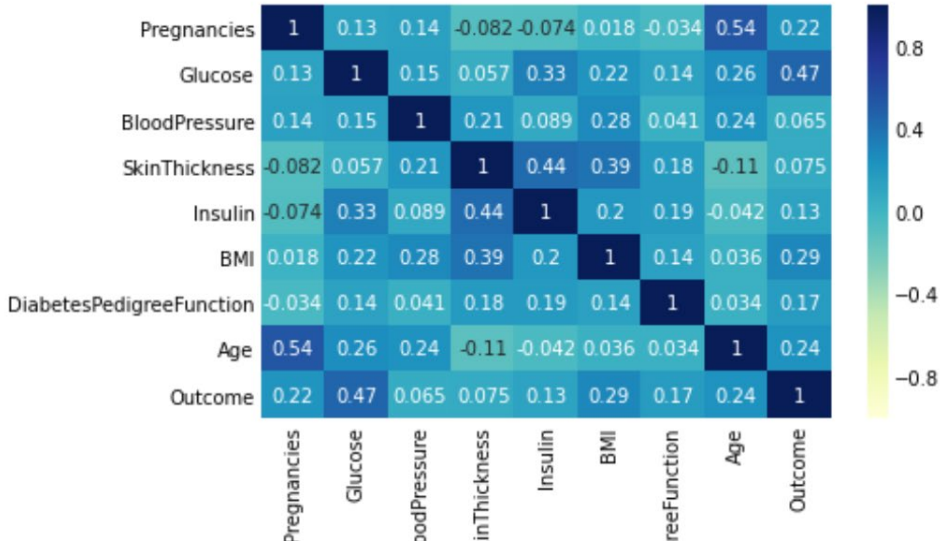
```
# Pairwise Pearson correlations
set_option('precision', 3) # Displays precision for decimal numbers
correlations = data.corr(method='pearson')
correlations
```

|:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000	0.129	0.141	-0.082	-0.074	0.018	-0.034	0.544	0.222
Glucose	0.129	1.000	0.153	0.057	0.331	0.221	0.137	0.264	0.467
BloodPressure	0.141	0.153	1.000	0.207	0.089	0.282	0.041	0.240	0.065
SkinThickness	-0.082	0.057	0.207	1.000	0.437	0.393	0.184	-0.114	0.075
Insulin	-0.074	0.331	0.089	0.437	1.000	0.198	0.185	-0.042	0.131
BMI	0.018	0.221	0.282	0.393	0.198	1.000	0.141	0.036	0.293
DiabetesPedigreeFunction	-0.034	0.137	0.041	0.184	0.185	0.141	1.000	0.034	0.174
Age	0.544	0.264	0.240	-0.114	-0.042	0.036	0.034	1.000	0.238
Outcome	0.222	0.467	0.065	0.075	0.131	0.293	0.174	0.238	1.000

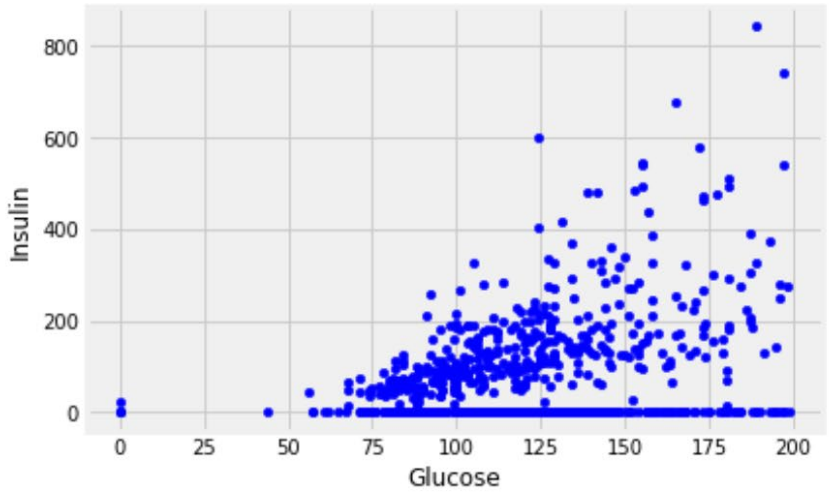
```
1 sns.heatmap(data.corr(), vmin=-1, vmax=1, cmap="YlGnBu", annot = True)
```

|: <matplotlib.axes._subplots.AxesSubplot at 0x1acaa3e9a48>



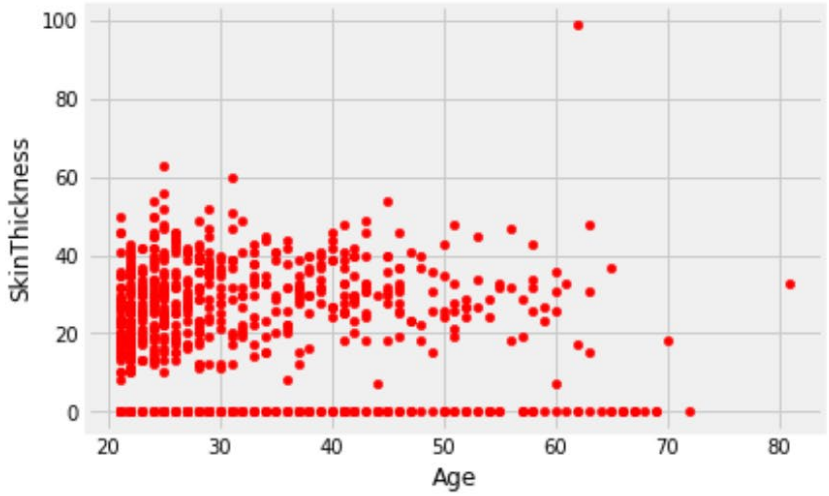
```
# Pandas Scatter Plot
data.plot(x='Glucose',y='Insulin',kind='scatter',color='B')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f08d8ee548>
```

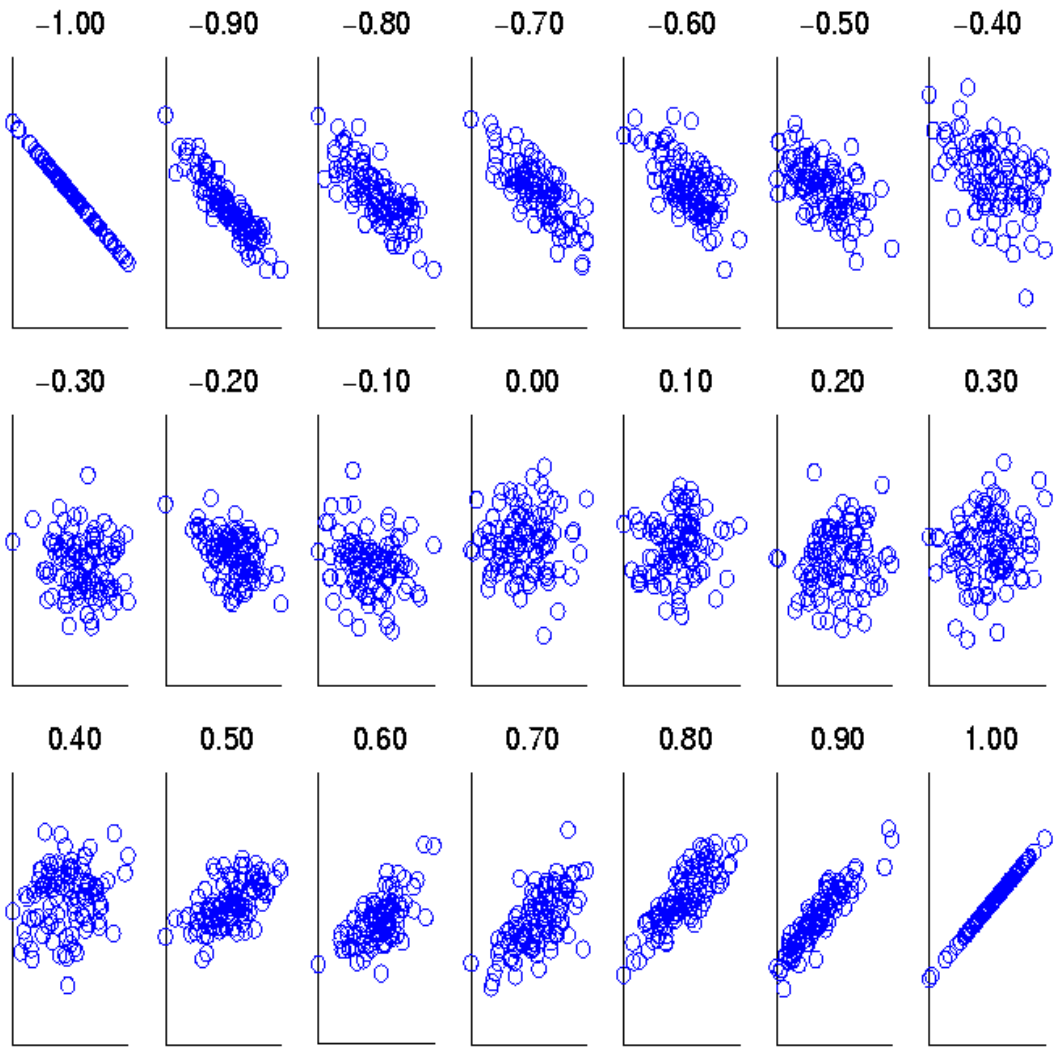



```
# Pandas Scatter Plot
data.plot(x='Age',y='SkinThickness',kind='scatter',color='R')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f08db397c8>
```



Evaluating Correlation



Scatter plots showing the similarity from –1 to 1.

Chi-Square Test of Independent

- Test the independence (or dependence) of two categorical attributes using the chi-square test
- The null hypothesis assumes true until we have evidence to go for it or go against it
  - H_0 (null hypothesis): the two attributes are independent
  - H_a (alternative hypothesis): the two attributes are dependent

Chi-Square Test

- Summarize the data in the two-way contingency table, the observed table
  - This table represents the observed counts of each cell
- Calculate the expected count for each cell in the table to find the expected table from observed counts
  - This table displays what the counts for each cell would be for sample data if there were not relations between two attributes
  - To find the expected count for each cell in the expected table we multiply the marginal row and column totals for that cell and divide by the overall total in the observed table
  - i.e. for each cell this is
  - $E = (row\ total \times column\ total) \,/\, total\ number\ of\ data$

Chi-Square Test

- Compute a Chi-Square test statistic as follows:
  - $\chi^2 = \Sigma(O_i - E_i)^2/E_i$
    - where O_i is an observed count of each cell of the contingency table, and E_i is an expected count of each cell of the contingency table
- With Chi-Square test statistic, a significance level (α) chosen, and the degree of freedom for the Chi-Square distribution, we can make a decision.
  - Degree of Freedom (df) = (number of rows – 1) x (numbers of columns – 1)
  - a significance level = 0.001

Chi-Square Test

- The decision is made by
  - Either comparing the value of test Chi-Square statistic to a critical chi-square value at a chosen significance level, α (rejection region approach)
    - Test statistic >= Critical value: reject null hypothesis, attributes are dependent (H_a)
    - Test statistic < Critical value: fail to reject null hypothesis, attributes are independent (H_0)
  - or finding the probability of getting of test Chi-Square statistic (p-value approach)
    - p-value <= α (a chosen significance level): reject null hypothesis, attributes are dependent (H_a)
    - p-value > α: fail to reject null hypothesis, attributes are independent (H_0)
  - http://courses.atlas.illinois.edu/spring2016/STAT/STAT200/pchisq.html

the hypothesis that *gender* and *preferred reading* are independent

	male	*female*	*Total*
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

Note: Are *gender* and *preferred_reading* correlated?

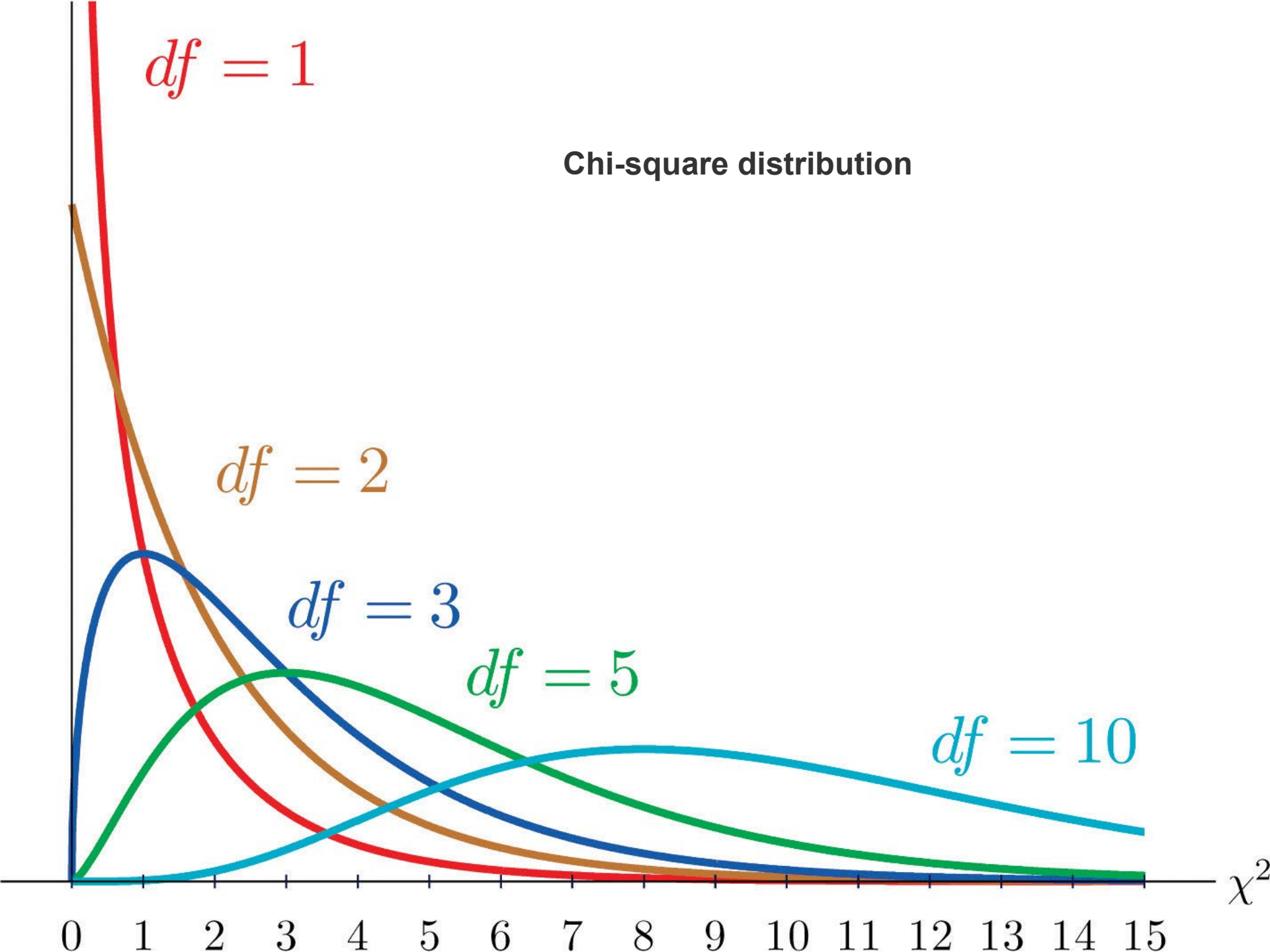
$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$
$$= 284.44 + 121.90 + 71.11 + 30.48 = 507.93.$$

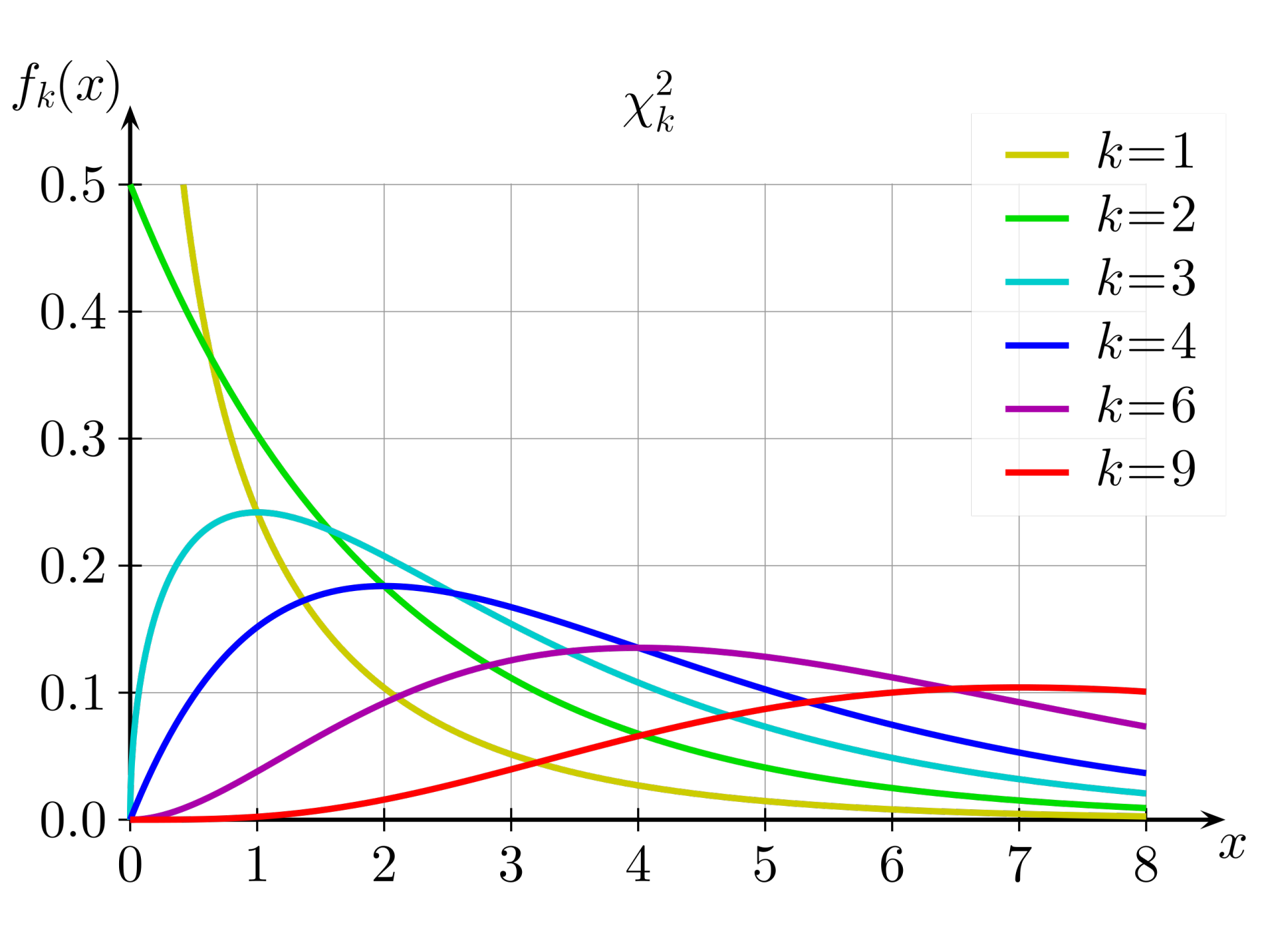
Data Mining, Jiawei

For the degree of freedom is 1 and the significance level, α = 0.001

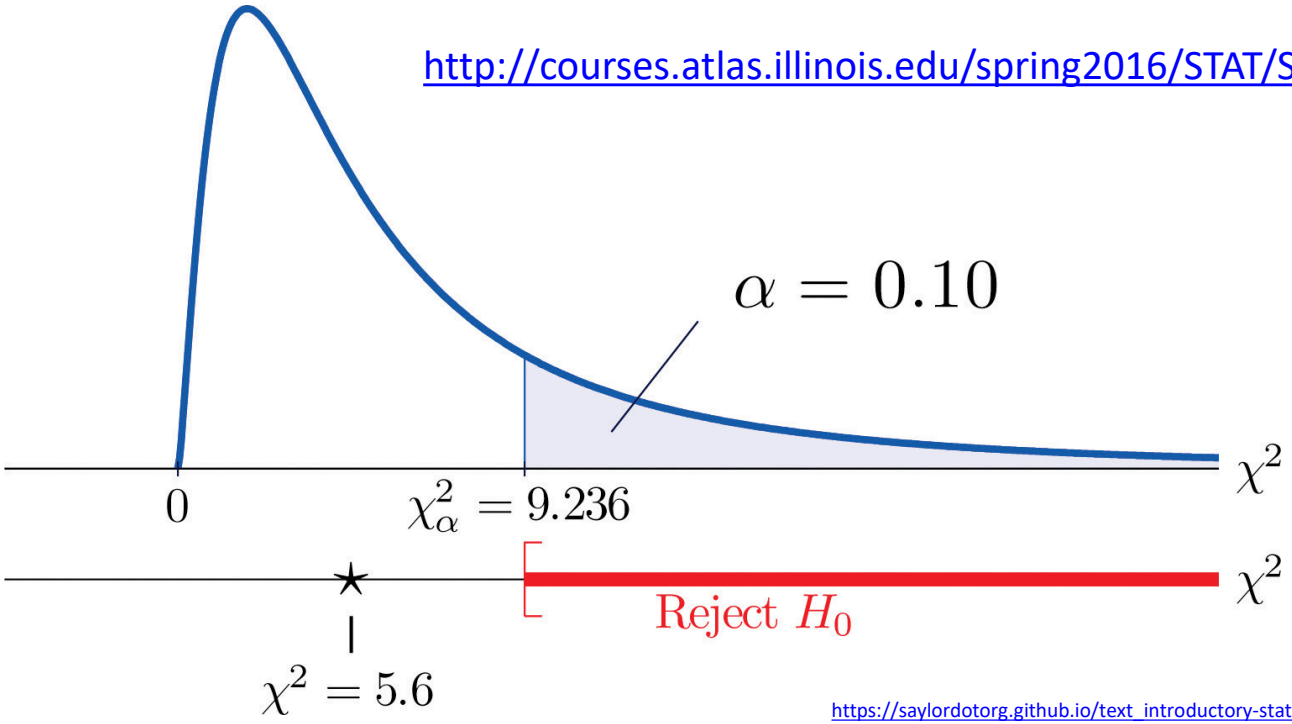
- the test Chi-Square statistic (χ^2) needed to reject the hypothesis at the significance level is equal to or greater than Chi-Square critical value (x_α^2), 10.828.
- the p-value of the test Chi-Square statistic needed to reject the hypothesis is equal to or less than the significance level, α, 0.001.

Since the computed Chi-Square value (χ^2) > x_α^2, (since p-value < α) we can reject the hypothesis that gender and preferred reading are independent and conclude that the two attributes are (strongly) correlated for the given group of people.


Chi-square distribution
df = 1
df = 2
df = 3
df = 5
df = 10
χ²
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15


$f_k(x)$
χ^2_k
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
7
8
x
$k=1$
$k=2$
$k=3$
$k=4$
$k=6$
$k=9$

http://courses.atlas.illinois.edu/spring2016/STAT/STAT200/pchisq.html



https://saylordotorg.github.io/text_introductory-statistics/s15-chi-square-tests-and-f-tests.html

Critical Value for Chi-Square

Select your significance level, input your degrees of freedom, and then hit "Calculate for Chi-Square".

Significance Level: 0.10 ▼

Degrees of Freedom: 5

Critical value = 9.236.

Calculate for Chi-Square

https://www.socscistatistics.com/tests/criticalvalues/default.aspx

Chi-square score: 5.6

DF: 5

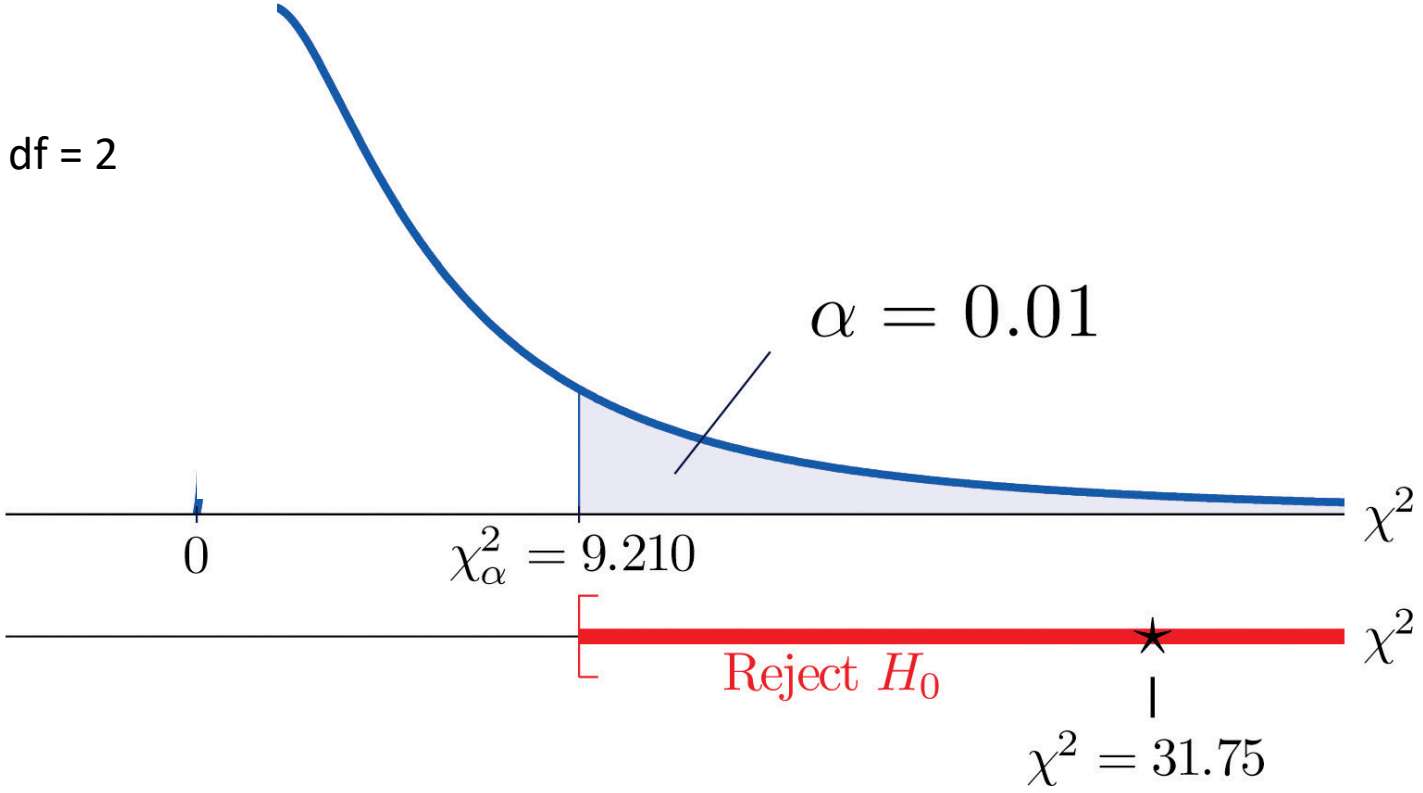
Significance Level:

- 0.01
- 0.05
- 0.10

The P-Value is .347105. The result is *not* significant at p < .10.

Calculate

https://www.socscistatistics.com/pvalues/chidistribution.aspx



Critical Value for Chi-Square

Select your significance level, input your degrees of freedom, and then hit "Calculate for Chi-Square"

Significance Level: 0.01

Degrees of Freedom: 2

Critical value = 9.21.

Calculate for Chi-Square

Chi-square score: 31.75

DF: 2

Significance Level:

- (•) 0.01
- ( ) 0.05
- ( ) 0.10

The P-Value is < .00001. The result is significant at p < .01.

Calculate

Critical values of the Chi-square distribution with *d* degrees of freedom

	Probability of exceeding the critical value						
d	0.05	0.01	0.001	*d*	0.05	0.01	0.001
1	3.841	6.635	10.828	11	19.675	24.725	31.264
2	5.991	9.210	13.816	12	21.026	26.217	32.910
3	7.815	11.345	16.266	13	22.362	27.688	34.528
4	9.488	13.277	18.467	14	23.685	29.141	36.123
5	11.070	15.086	20.515	15	24.996	30.578	37.697
6	12.592	16.812	22.458	16	26.296	32.000	39.252
7	14.067	18.475	24.322	17	27.587	33.409	40.790
8	15.507	20.090	26.125	18	28.869	34.805	42.312
9	16.919	21.666	27.877	19	30.144	36.191	43.820
10	18.307	23.209	29.588	20	31.410	37.566	45.315

***INTRODUCTION TO POPULATION GENETICS,* Table D.1**

© 2013 Sinauer Associates, Inc.

Degree of Freedom	Probability of Exceeding the Critical Value								
	0.99	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.01
1	0.000	0.004	0.016	0.102	0.455	1.32	2.71	3.84	6.63
2	0.020	0.103	0.211	0.575	1.386	2.77	4.61	5.99	9.21
3	0.115	0.352	0.584	1.212	2.366	4.11	6.25	7.81	11.34
4	0.297	0.711	1.064	1.923	3.357	5.39	7.78	9.49	13.28
5	0.554	1.145	1.610	2.675	4.351	6.63	9.24	11.07	15.09
6	0.872	1.635	2.204	3.455	5.348	7.84	10.64	12.59	16.81
7	1.239	2.167	2.833	4.255	6.346	9.04	12.02	14.07	18.48
8	1.647	2.733	3.490	5.071	7.344	10.22	13.36	15.51	20.09
9	2.088	3.325	4.168	5.899	8.343	11.39	14.68	16.92	21.67
10	2.558	3.940	4.865	6.737	9.342	12.55	15.99	18.31	23.21
11	3.053	4.575	5.578	7.584	10.341	13.70	17.28	19.68	24.72
12	3.571	5.226	6.304	8.438	11.340	14.85	18.55	21.03	26.22
13	4.107	5.892	7.042	9.299	12.340	15.98	19.81	22.36	27.69
14	4.660	6.571	7.790	10.165	13.339	17.12	21.06	23.68	29.14
15	5.229	7.261	8.547	11.037	14.339	18.25	22.31	25.00	30.58
16	5.812	7.962	9.312	11.912	15.338	19.37	23.54	26.30	32.00
17	6.408	8.672	10.085	12.792	16.338	20.49	24.77	27.59	33.41
18	7.015	9.390	10.865	13.675	17.338	21.60	25.99	28.87	34.80
19	7.633	10.117	11.651	14.562	18.338	22.72	27.20	30.14	36.19
20	8.260	10.851	12.443	15.452	19.337	23.83	28.41	31.41	37.57
22	9.542	12.338	14.041	17.240	21.337	26.04	30.81	33.92	40.29
24	10.856	13.848	15.659	19.037	23.337	28.24	33.20	36.42	42.98
26	12.198	15.379	17.292	20.843	25.336	30.43	35.56	38.89	45.64
28	13.565	16.928	18.939	22.657	27.336	32.62	37.92	41.34	48.28
30	14.953	18.493	20.599	24.478	29.336	34.80	40.26	43.77	50.89
40	22.164	26.509	29.051	33.660	39.335	45.62	51.80	55.76	63.69
50	27.707	34.764	37.689	42.942	49.335	56.33	63.17	67.50	76.15
60	37.485	43.188	46.459	52.294	59.335	66.98	74.40	79.08	88.38
	Not Significant							***Significant***	

```
# calculate Chi-Squared test
# Returns of scipy.stats.chi2_contingency()
# chi2: The test statistic.
# p: The p-value of the test
# dof: Degrees of freedom
# expected: The expected frequencies, based on the marginal sums of the table.
```

```
data = pd.read_csv('courses.csv')
```

```
data
```

:

	Gender	Interest
0	Male	Science
1	Male	Science
2	Male	Science
3	Male	Science
4	Male	Science
...	...	...
87	Female	Art
88	Female	Art
89	Female	Art
90	Female	Art
91	Female	Art

92 rows × 2 columns

]:
```
#Contingency Table
table = pd.crosstab(data["Gender"],data["Interest"])
```

]:
```
table
```

:[95]:

Interest	Art	Math	Science
Sex			
Female	30	20	10
Male	17	9	6

]:
```
# Observed Values
Observed_Values = table.values
Observed_Values
```

:109]:
```
array([[30, 20, 10],
       [17,  9,  6]], dtype=int64)
```

```
# calculate Chi-Squared test
# Returns of scipy.stats.chi2_contingency()
# chi2: The test statistic.
# p: The p-value of the test
# dof: Degrees of freedom
# expected: The expected frequencies, based on the marginal sums of the table.

import scipy.stats as sp

from scipy.stats import chi2_contingency
from scipy.stats import chi2

chi2_test_statistic, p, dof, expected = sp.chi2_contingency(table)

print('dof=%d' % dof)
```

```
dof=2
```

```
chi2_test_statistic, p, dof, expected
```

```
10]: (0.271574651504035,
      0.8730282833800731,
      2,
      array([[30.65217391, 18.91304348, 10.43478261],
             [16.34782609, 10.08695652,  5.56521739]]))
```

```
# interpret test-statistic

# Test Statistic >= Critical Value: reject null hypothesis, dependent (Ha)
# Test Statistic < Critical Value:  fail to reject null hypothesis,independent (Ho).
# chi2.ppf(q, df, loc=0, scale=1) inverset CDF.

from scipy.stats import chi2

prob = 0.95 # significant value = 1 - 0.95 = 0.05
critical = chi2.ppf(prob, dof)
```

```
round(critical, 3)
```

: 5.991

```
print('critical=%.3f, chi2_test_statistic=%.3f' % (critical, chi2_test_statistic))
```

```
critical=5.991, chi2_test_statistic=0.272
```

```
if chi2_test_statistic >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
Independent (fail to reject H0)
```

```
# interpret p-value
# p-value <= alpha: reject null hypothesis, dependent (Ha)
# p-value > alpha:  fail to reject null hypothesis,independent (Ho).
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
significance=0.050, p=0.873
Independent (fail to reject H0)
```