AWS

MACHINE LEARNING

CERTIFICATION

DOMAIN #2: EXPLORATORY DATA ANALYSIS (24% EXAM)

INTRODUCTION

INTRODUCTION

- This part of the exam accounts for 24% and is not specific to AWS or SageMaker.
- This section of the test will assess your skills in feature engineering and data cleaning.
- This section will cover fundamentals such as normalization, one hot encoding, and filling in missing data values.
- This section relies on Python and famous packages such as pandas, numpy, and seaborn.
- Please note that the exam will not test you on python or ask you to write any code.

AWS ML CERTIFICATION EXAM DOMAINS

Domain	% of Examination
Domain 1: Data Engineering	20%
Domain 2: Exploratory Data Analysis	24%
Domain 3: Modeling	36%
Domain 4: Machine Learning Implementation and Operations	20%
TOTAL	100%

Source: https://d1.awsstatic.com/training-and-certification/docs-ml/AWS%20Certified%20Machine%20Learning%20-%20Specialty_Exam%20Guide%20(1).pdf

DOMAIN #2 OVERVIEW:

SECTION #5: JUPYTER NOTEBOOKS, SCIKIT LEARN, PYTHON PACKAGES, AND DISTRIBUTIONS

- Introduction
- Jupyter Notebooks and Scikit Learn
- Python Packages (Pandas, Numpy, Matplotlib and Seaborn
- Distributions (Normal, Standard, Poisson, Bernoulli)
- Time Series

SECTION #6: AMAZON ATHENA, QUICKSIGHT AND ELASTIC MAP REDUCE

- Amazon Athena Features
- Amazon Athena Deep Dive (Security, Cost, and glue integration)
- Amazon QuickSight Features
- Amazon QuickSight (integration with AWS services)
- Amazon QuickSight ML insights and Use Cases
- Elastic Map Reduce (EMR)
- Apache Hadoop with EMR
- Apache Spark with EMR

DOMAIN #2 OVERVIEW:

SECTION #7: FEATURE ENGINEERING

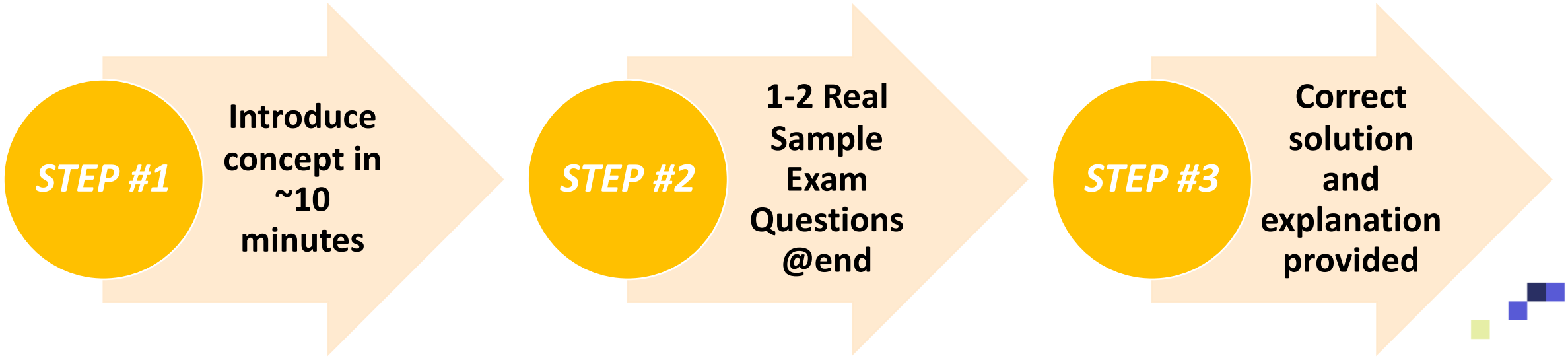
- Introduction to Feature Engineering
  - Amazon SageMaker GroundTruth
  - Feature Selection
  - Scaling
  - Imputation
  - Outliers
  - One Hot Encoding
  - Binning
  - Log Transformation
  - Shuffling, Feature Splitting, Unbalanced Datasets
  - Text Feature Engineering overview
  - Bag of words, punctuation, and dates (easy ones!)
  - Term Frequency Inverse Document Frequency (TF-IDF)
  - N-Grams (Unigram vs. Bigram vs. Trigram)
  - Orthogonal Sparse Bigram (OSB)
  - Cartesian Product Transformation

LECTURE DESIGN

- We know how hard it is to study for an exam especially if you have a busy schedule.
- This course is designed to be extremely on point and optimized to pass the exam.

No boring content. Zero unnecessary information.

- Here's the lecture structure that we will follow:



RECALL OUR MINI CHALLENGE AND PRIZE!

- For those of you who will successfully complete the entire section and watch the videos till the end, they will receive a valuable prize!

https://pixabay.com/illustrations/trophy-win-prize-transparent-1414791/

INTRODUCTION

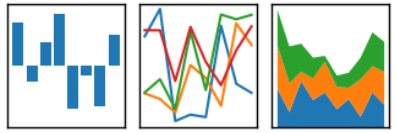
- In order to perform data visualization, there are generally two approaches: (1) Use Developer tools or (2) use Business intelligence tools.

DEVELOPER TOOLS

BUSINESS INTELLIGENCE TOOLS

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



jupyter

INSTALL PROJECT DOCUMENTATION BLOG DONATE

jupyter

Jupyter Notebook

The Jupyter Notebook is a web-based interactive computing platform that allows users to author data- and code-driven narratives that combine live code, equations, narrative text, visualizations, interactive dashboards and other media.

AMAZON SAGEMAKER

AMAZON QUICKSIGHT

+ a b | e a u

S O F T W A R E

TABLEAU

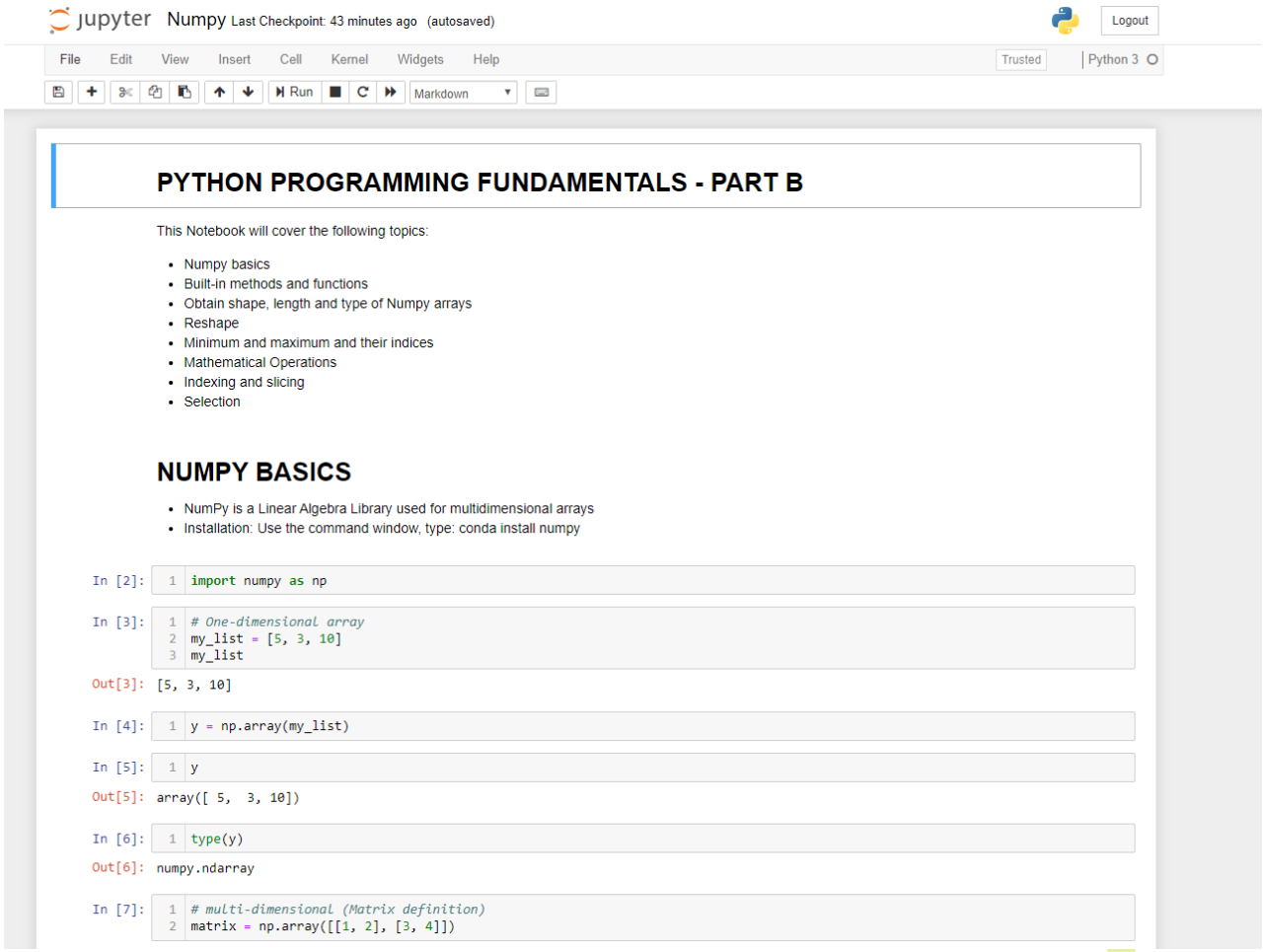
Microsoft Power BI

POWER BI

JUPYTER NOTEBOOKS AND SCIKIT LEARN

JUPYTER NOTEBOOKS

- Jupyter Notebooks are open-source web application that enable developers to develop and distribute codes, text, equations, and figures in one place.
- It's one of the top tools used for machine learning developers.
- In Jupyter notebooks, you can write in 40 programming languages such as Python, R, and Scala.
- You can Share notebooks including code results with other.
- Jupyter enable developers to leverage big data tools, such as Apache Spark, from Python, R and Scala.
- https://jupyter.org/



SCIKIT LEARN

- Scikit-learn is a free machine learning library developed for python.
- Scikit-learn offers several algorithms for classification, regression, clustering
- Several famous models are included such as support vector machines, random forests, gradient boosting, and k-means.
- Scikit learn can be efficiently used in data preprocessing.

```
In [24]:
1 from sklearn.model_selection import train_test_split
2
3 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state=5)
```

```
In [25]:
1 X_train.shape

Out[25]: (455, 30)
```

```
In [26]:
1 X_test.shape

Out[26]: (114, 30)
```

```
In [27]:
1 y_train.shape

Out[27]: (455,)
```

```
In [28]:
1 y_test.shape

Out[28]: (114,)
```

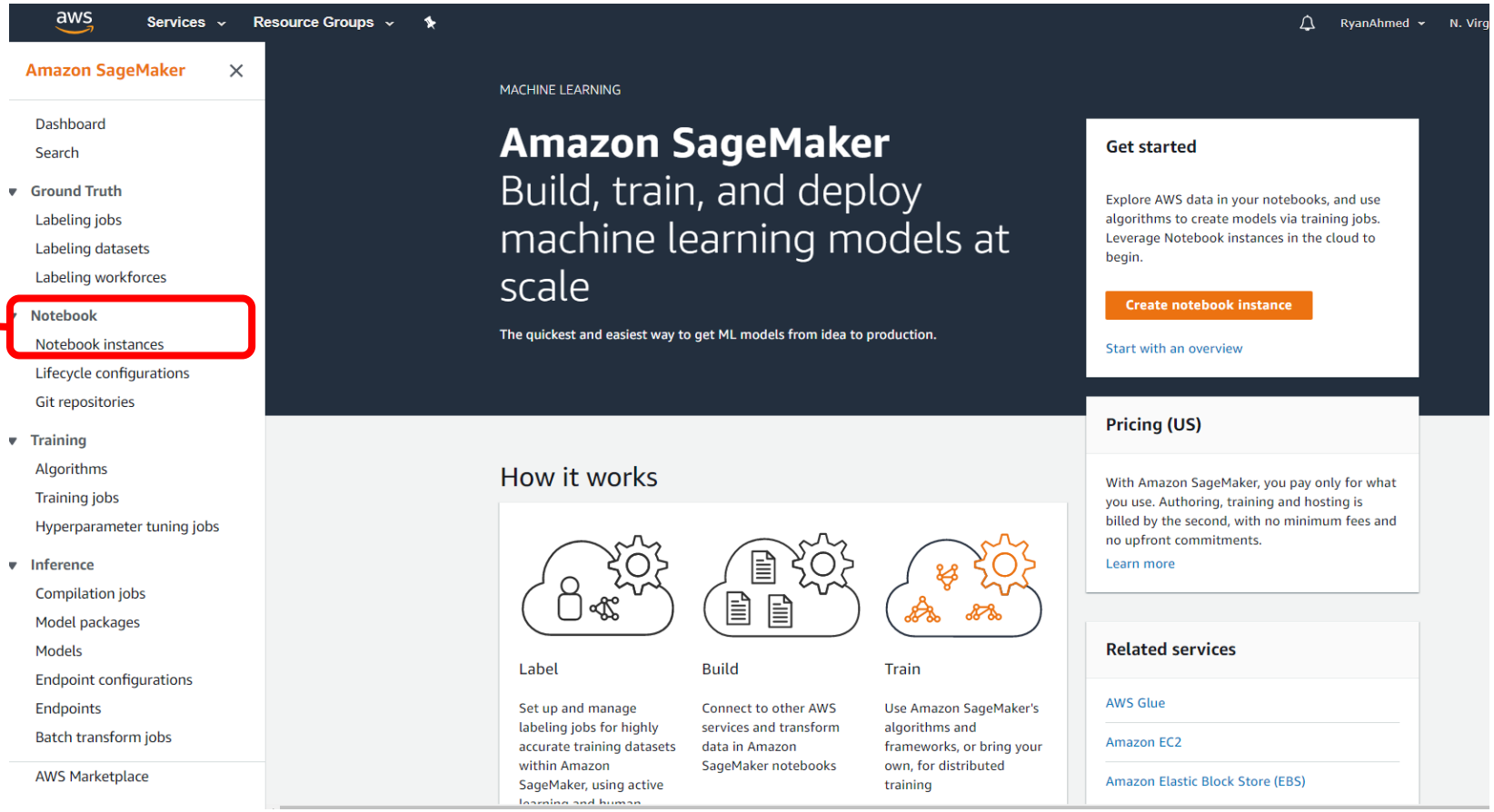
```
In [29]:
1 from sklearn.svm import SVC
2 from sklearn.metrics import classification_report, confusion_matrix
3
4 svc_model = SVC()
5 svc_model.fit(X_train, y_train)
```

SPLITTING THE DATA INTO TRAINING AND TESTING

TRAINING A SUPPORT VECTOR MACHINES

SAGEMAKER JUPYTER NOTEBOOKS

CREATE A NOTEBOOK INSTANCE IN SAGEMAKER



PYTHON PACKAGES

NUMPY: MATHEMATICAL OPERATIONS

MATHEMATICAL OPERATIONS

In [30]:
```
1 x = np.arange(1, 5)
2 x
```

Out[30]: array([1, 2, 3, 4])

In [50]:
```
1 y = np.arange(1, 5)
2 y
```

Out[50]: array([1, 2, 3, 4])

In [32]:
```
1 z = x+y
2 z
```

Out[32]: array([2, 4, 6, 8])

In [33]:
```
1 z = x**2
2 z
```

Out[33]: array([1, 4, 9, 16], dtype=int32)

In [34]:
```
1 k = np.sqrt(z)
2 k
```

Out[34]: array([1., 2., 3., 4.])

In [35]:
```
1 z = np.exp(y)
2 z
```

Out[35]: array([2.71828183, 7.3890561 , 20.08553692, 54.59815003])

NUMPY: MATRIX DEFINTIONS AND ELEMENTS SELECTION

ELEMENTS SELECTION

In [47]:
```
1 matrix = np.random.randint(1,10, (5, 5))
2 matrix
```

Out[47]:
```
array([[5, 9, 8, 8, 8],
       [3, 9, 2, 8, 4],
       [1, 9, 5, 8, 4],
       [3, 3, 3, 8, 9],
       [4, 4, 7, 8, 5]])
```

In [48]:
```
1 new_matrix = matrix[matrix>3]
2 new_matrix
```

Out[48]:
```
array([5, 9, 8, 8, 8, 9, 8, 4, 9, 5, 8, 4, 8, 9, 4, 4, 7, 8, 5])
```

In [49]:
```
1 new_matrix = matrix[matrix%2==0]
2 new_matrix
```

Out[49]:
```
array([8, 8, 8, 2, 8, 4, 8, 4, 8, 4, 4, 8])
```

PANDAS

- Pandas is an open source library that offers high-performance data structures and data analysis tools in python.
- Data can also be stored using pandas DataFrame.
- Think of it as using Microsoft excel in python/jupyter environment.

GETTING CSV DATA

In [14]:
```
1 import pandas as pd
2
3 df = pd.read_csv('sample_file.csv')
```

In [15]:
```
1 df
```

Out[15]:

	first	last	email	postal	gender	dollar
0	Joseph	Patton	daafeja@boh.jm	M6U 5U7	Male	$2,629.13
1	Noah	Moran	guutodi@bigwoc.kw	K2D 4M9	Male	$8,626.96
2	Nina	Keller	azikez@gahew.mr	S1T 4E6	Male	$9,072.02

In [16]:
```
1 # write to a csv file
2 df.to_csv('sample_output.csv',index=False)
```

MATPLOTLIB

BASIC PLOT

In [3]:
```
1 import numpy as np
2 x = np.arange(0, 10, 0.2) # evenly sampled time at 0.2 s intervals
3 x
```

Out[3]:
```
array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4,
       2.6, 2.8, 3. , 3.2, 3.4, 3.6, 3.8, 4. , 4.2, 4.4, 4.6, 4.8, 5. ,
       5.2, 5.4, 5.6, 5.8, 6. , 6.2, 6.4, 6.6, 6.8, 7. , 7.2, 7.4, 7.6,
       7.8, 8. , 8.2, 8.4, 8.6, 8.8, 9. , 9.2, 9.4, 9.6, 9.8])
```

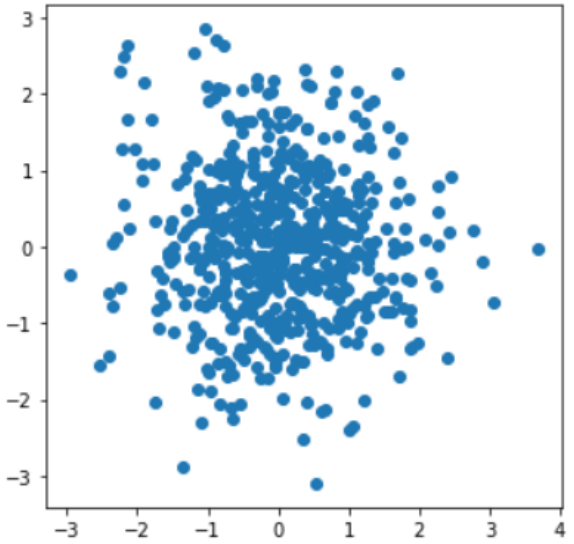
In [4]:
```
1 y = np.sin(x)
```

In [6]:
```
1 plt.plot(x, y)
2 plt.xlabel('Time')
3 plt.ylabel('Sine Wave')
4 plt.title('My first plotting exercise!')
```

Out[6]:
```
Text(0.5,1,'My first plotting exercise!')
```

In [6]:
```
1 import random
2
3 fig = plt.figure(figsize=(5,5))
4
5 X = np.random.randn(600)
6 Y = np.random.randn(600)
7
8 plt.scatter(X,Y)
9
```

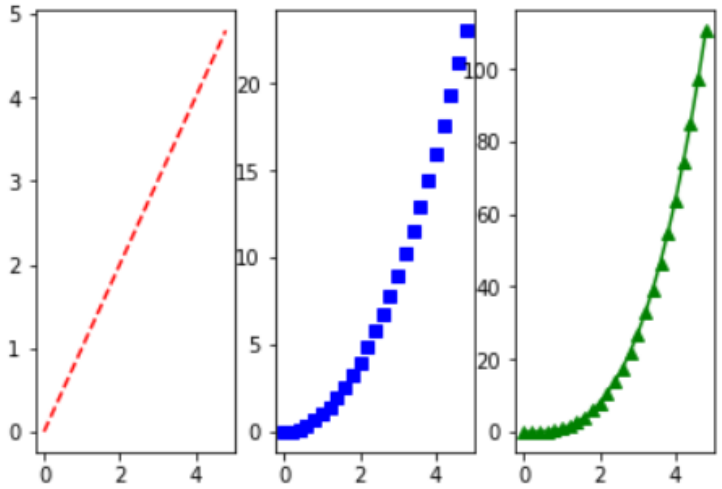
Out[6]:
```
<matplotlib.collections.PathCollection at 0x1a4214bb400>
```



MATPLOTLIB

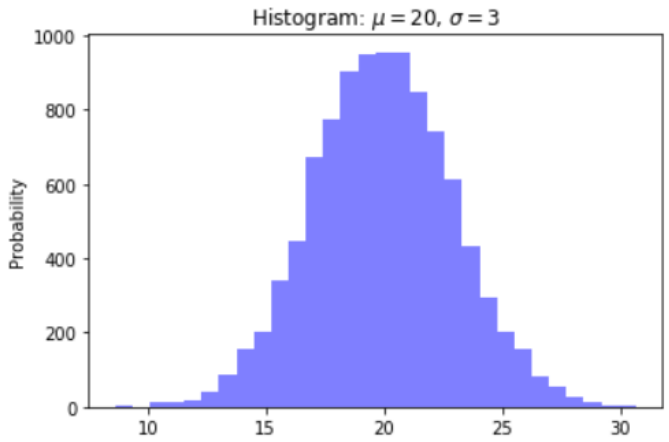
SUBPLOTS

In [8]:
```
plt.subplot(1, 3, 1)
plt.plot(t, t, 'r--');

plt.subplot(1, 3, 2)
plt.plot(t, t**2, 'bs')

plt.subplot(1, 3, 3)
plt.plot(t, t**3, 'g^-');
```



In [18]:
```
mu = 20 # mean of distribution
sigma = 3 # standard deviation of distribution
x = mu + sigma * np.random.randn(10000)

num_bins = 30

n, bins, patches = plt.hist(x, num_bins, facecolor='blue', alpha=0.5)

plt.ylabel('Probability')
plt.title(r'Histogram: $\mu=20$, $\sigma=3$')

```

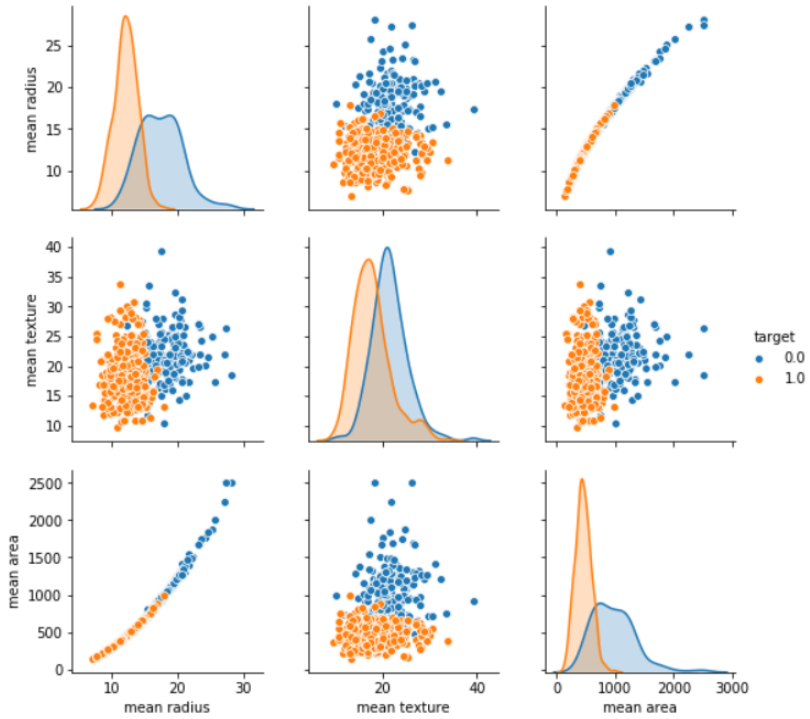
Out[18]: Text(0.5,1,'Histogram: $\\mu=20$, $\\sigma=3$')



SEABORN

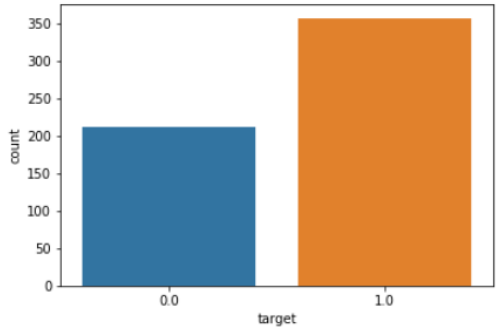
```
In [4]: 1 sns.pairplot(df_cancer, hue = 'target', vars = ['mean radius', 'mean texture', 'mean area'] )
```

```
Out[4]: <seaborn.axisgrid.PairGrid at 0x2cc12018cf8>
```



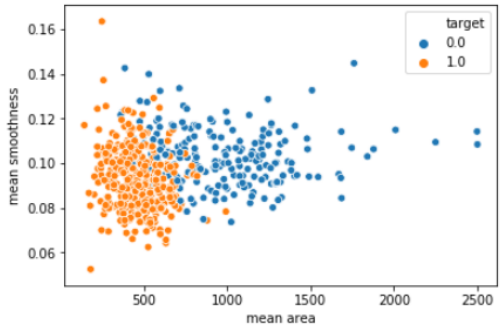

```
In [5]: 1 sns.countplot(df_cancer['target'], label = "Count")
```

```
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x2cc13756b38>
```




```
In [6]: 1 sns.scatterplot(x = 'mean area', y = 'mean smoothness', hue = 'target', data = df_cancer)
        2
```

```
Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x2cc13b139b0>
```



SEABORN: HEATMAPS

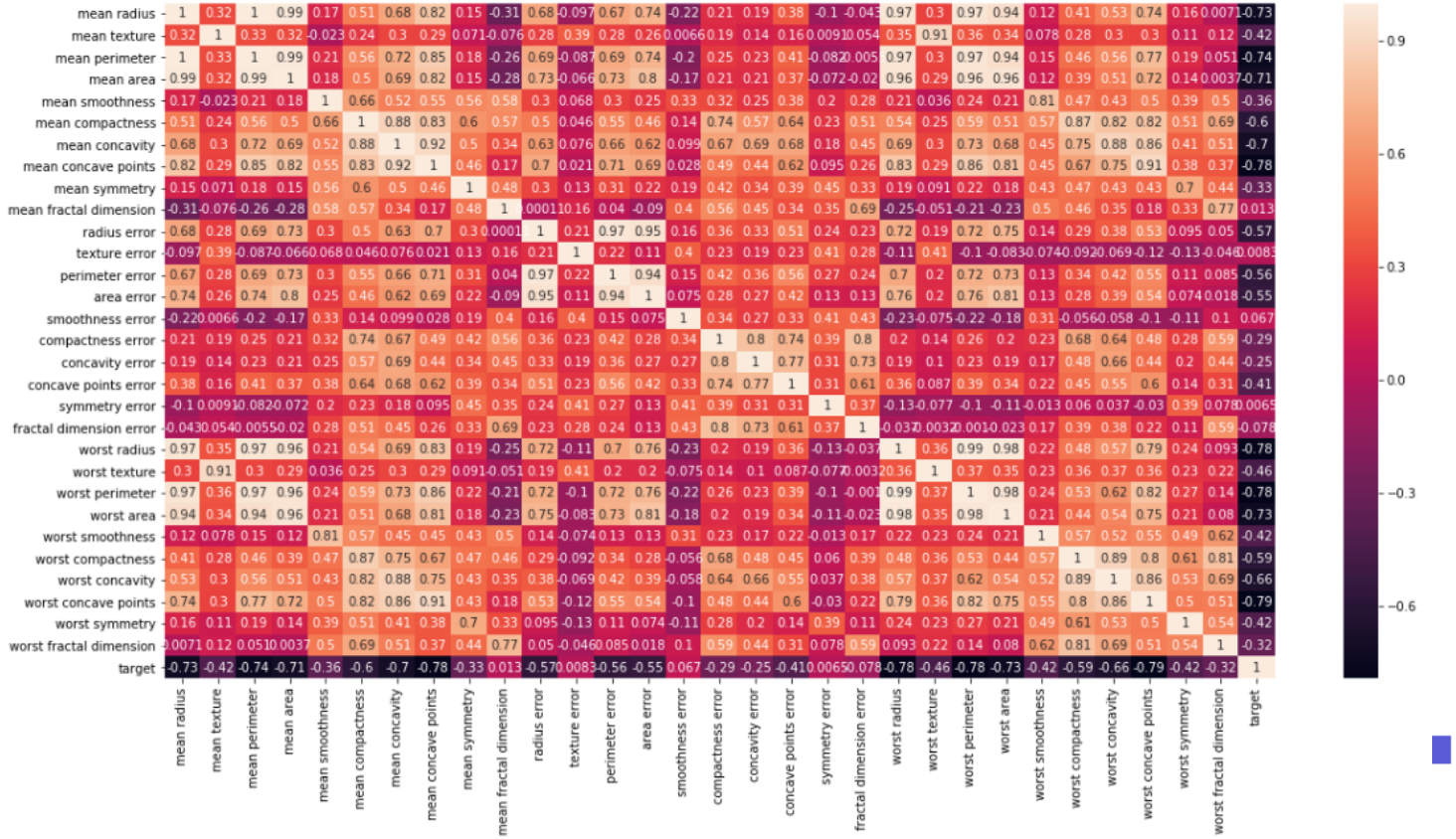
- Heat maps are used to represents values as colours.

In [7]:

```
# Strong correlation between the mean radius and mean perimeter, mean area and mean primeter
plt.figure(figsize=(20,10))
sns.heatmap(df_cancer.corr(), annot=True)
```

Out[7]:

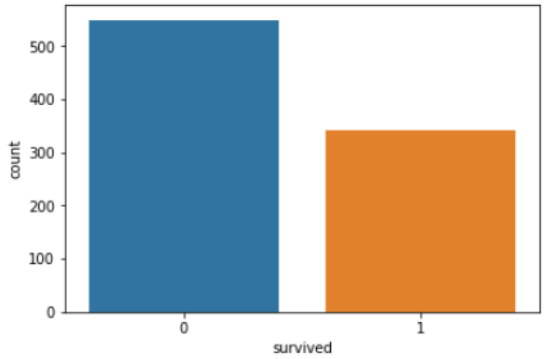
```
<matplotlib.axes._subplots.AxesSubplot at 0x2cc13b872b0>
```



SEABORN

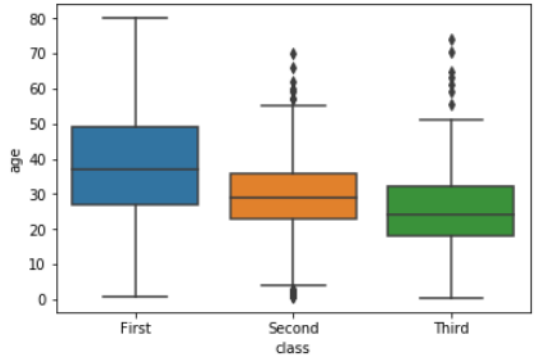
```
In [10]: 1 sns.countplot(x = 'survived', data = titanic_data)
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x2cc146dd6d8>
```



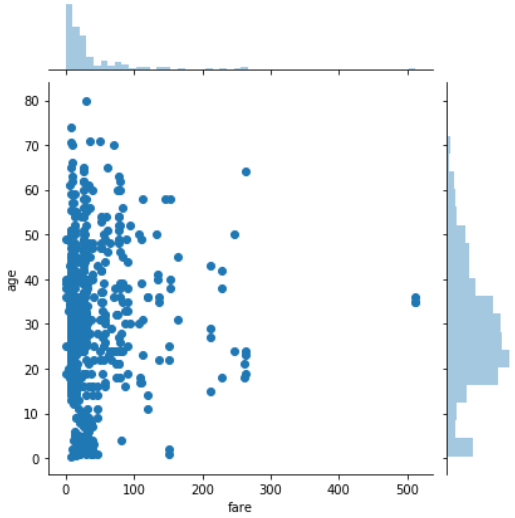

```
In [11]: 1 sns.boxplot(x='class', y='age', data=titanic_data)
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x2cc13cffc50>
```



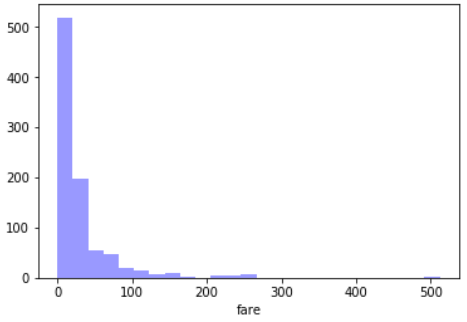

```
In [13]: 1 sns.jointplot(x='fare', y='age', data = titanic_data)
```

```
Out[13]: <seaborn.axisgrid.JointGrid at 0x2cc13ddaf98>
```




```
In [14]: 1 sns.distplot(titanic_data['fare'], bins = 25, kde= False,color = 'blue')
```

```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x2cc13e950b8>
```

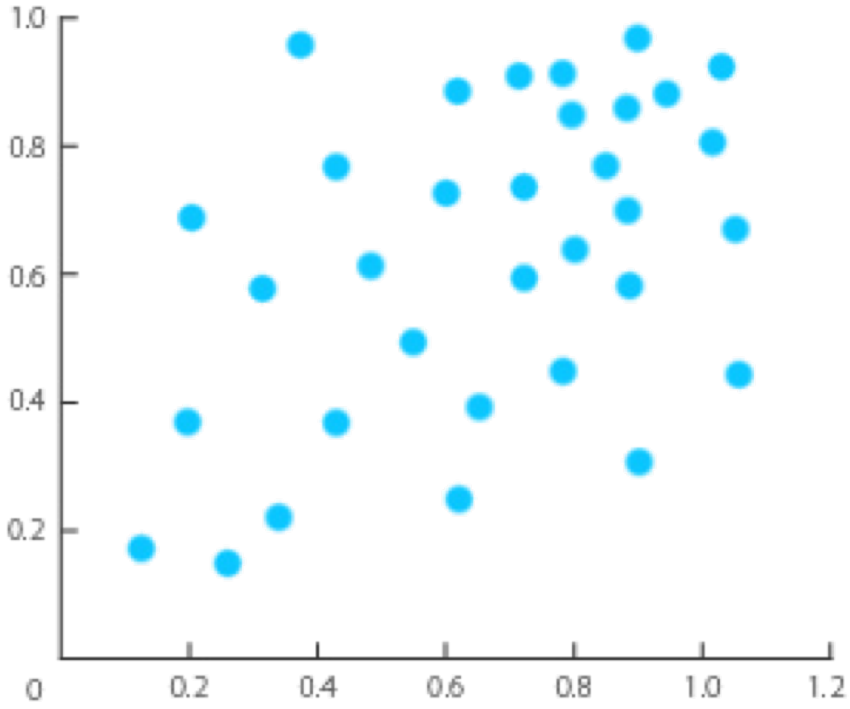


DATA VISUALIZATION

RELATIONSHIPS

SCATTERPLOT

"Scatterplot demonstrates the relationship between two variables (X, Y)"



BUBBLE CHART

"Bubble chart demonstrates the relationship between three variables (X, Y, Bubble Size)

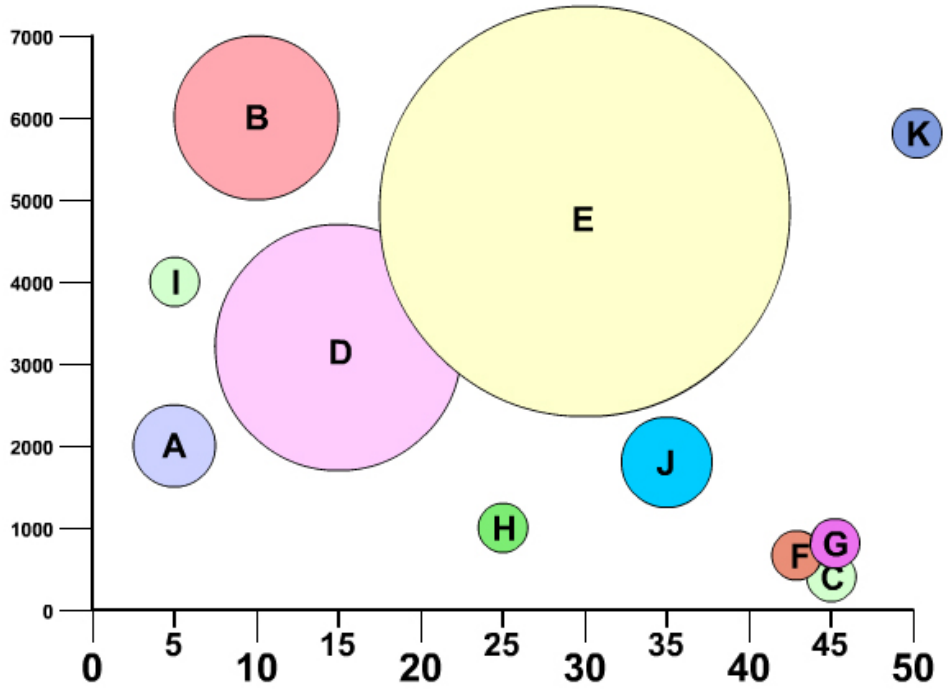
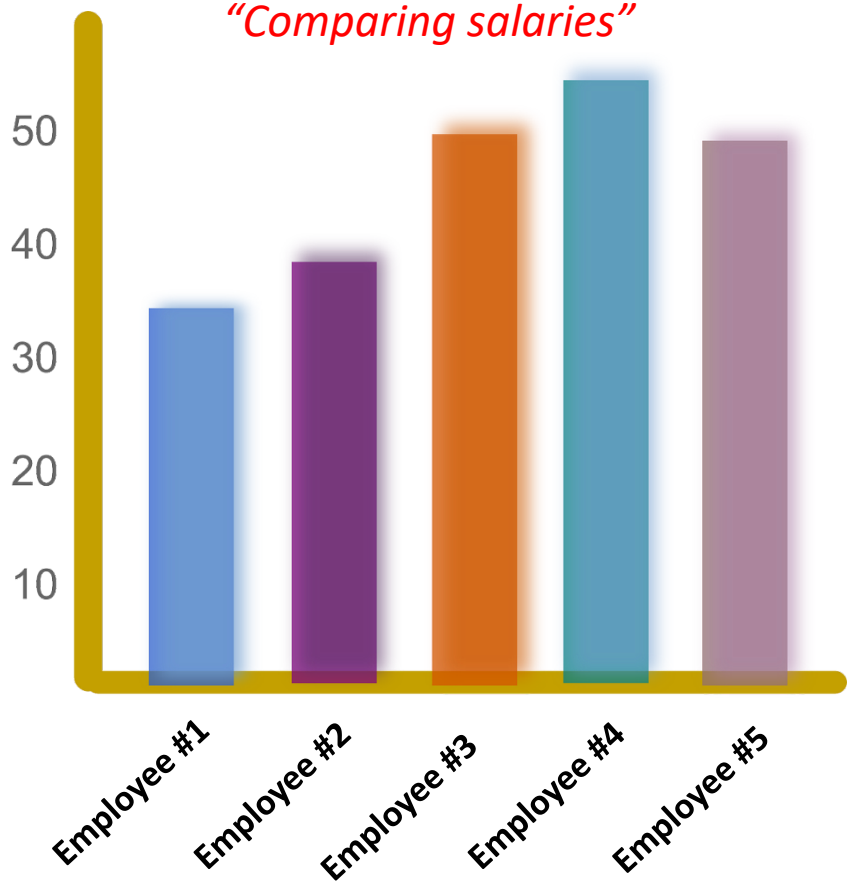


Photo Credit: https://commons.wikimedia.org/wiki/File:Example_of_Scatter_Plot.jpg
Photo Credit: https://commons.wikimedia.org/wiki/File:Bubble_chart.jpg

COMPARISONS

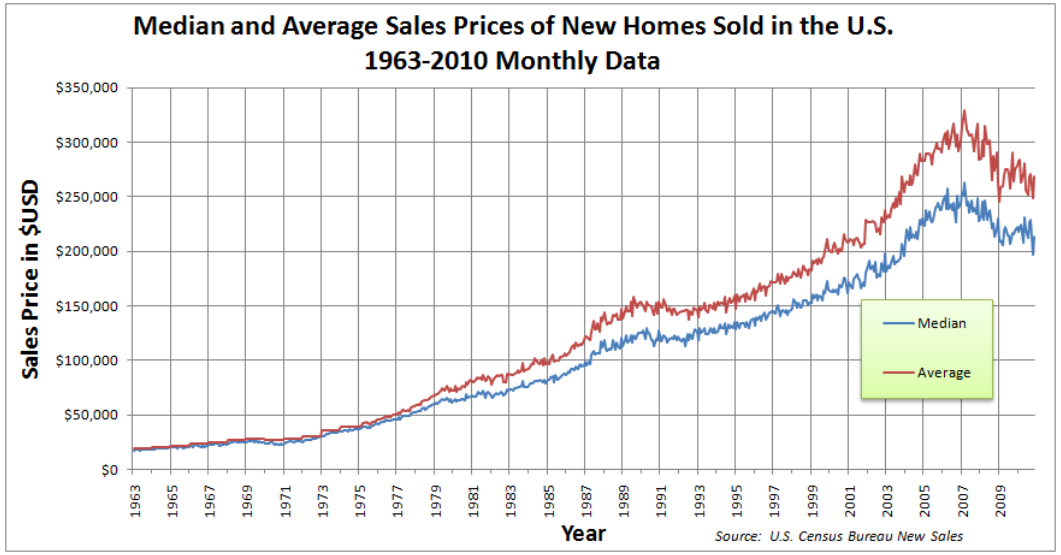
BAR CHART

"Comparing salaries"



LINE CHART

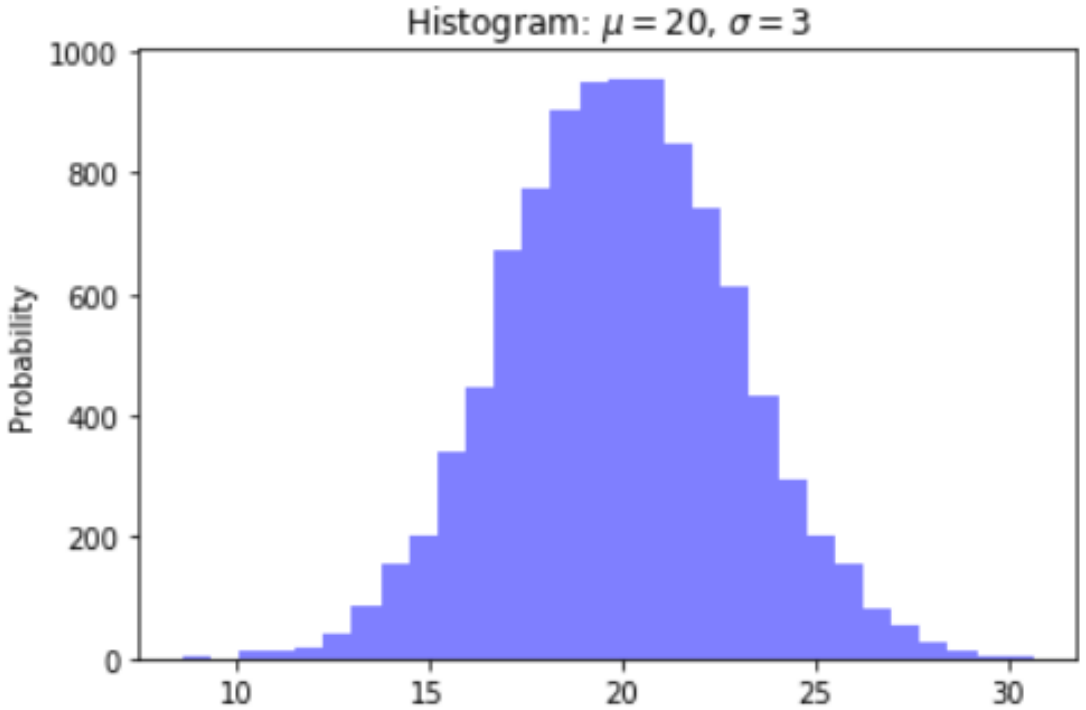
"Comparing median and average House prices over the years"



https://www.needpix.com/photo/89660/productivity-statistics-bar-chart-chart-graph-diagram-results
https://commons.wikimedia.org/wiki/File:Median_and_Average_Sales_Prices_of_New_Homes_Sold_in_the_US_1963-2010_Monthly.png

DISTRIBUTIONS

HISTOGRAMS



BOX PLOT

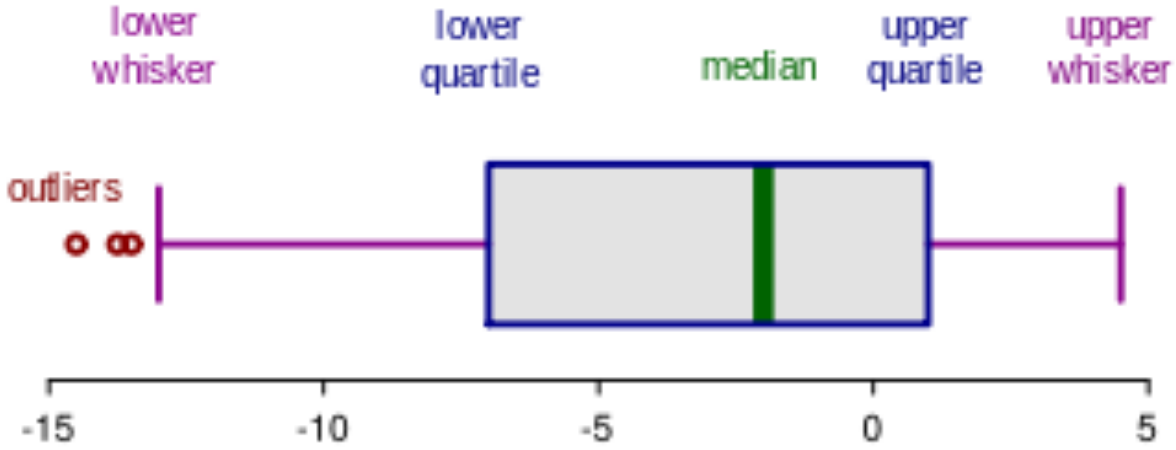


Photo Credit: https://commons.wikimedia.org/wiki/File:Elements_of_a_boxplot_en.svg

BOX PLOT

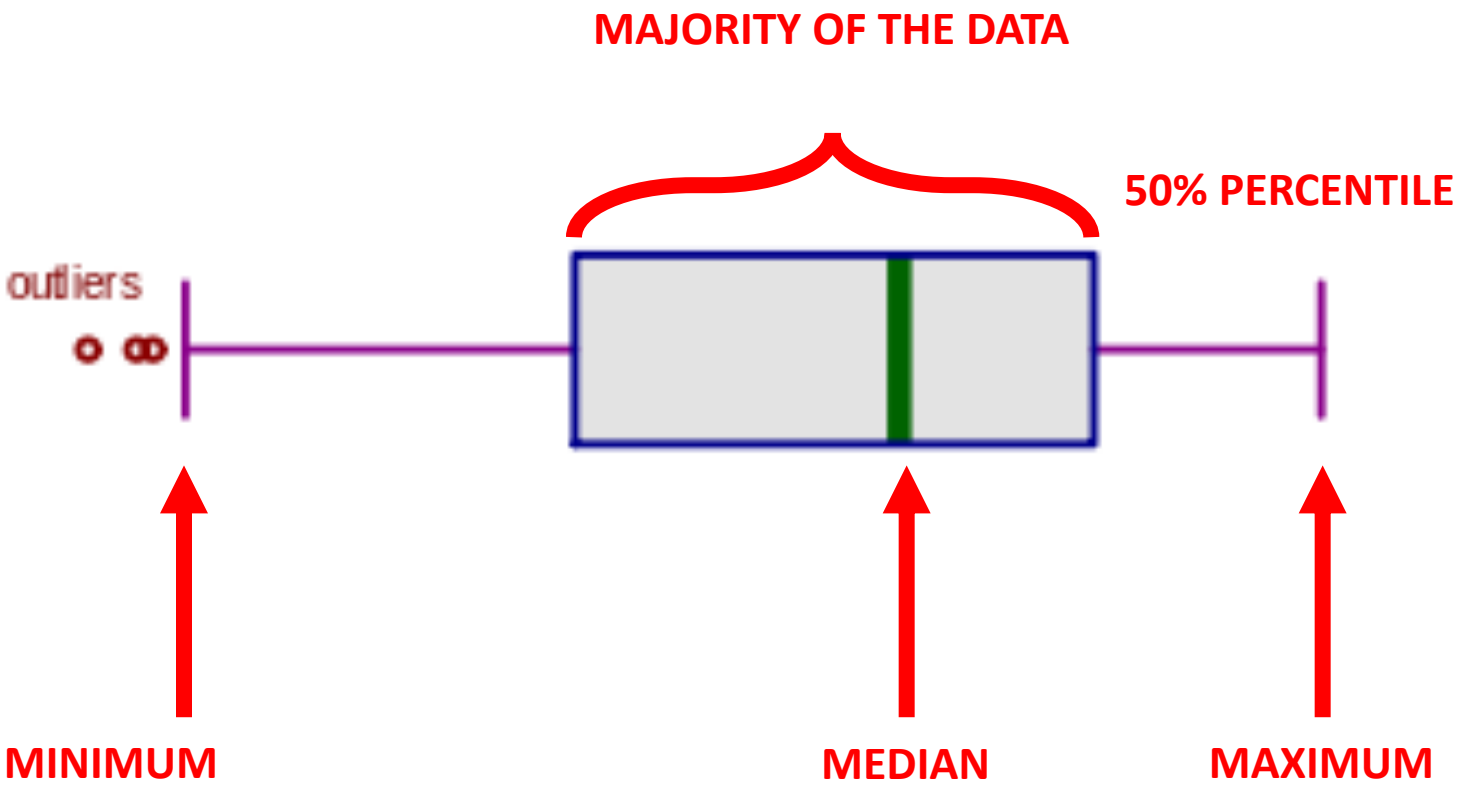
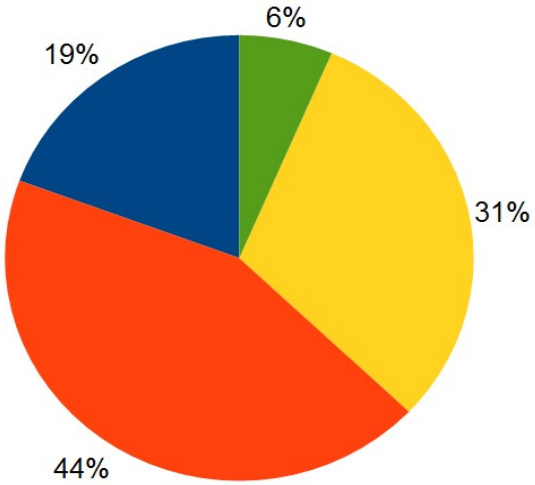


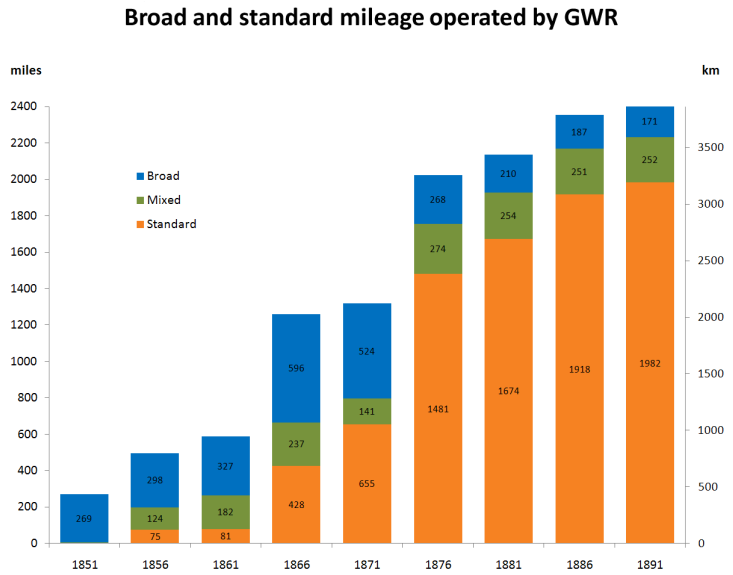
Photo Credit: https://commons.wikimedia.org/wiki/File:Elements_of_a_boxplot_en.svg

COMPOSITIONS

PIE CHART



STACKED BAR CHART



STACKED AREA CHART

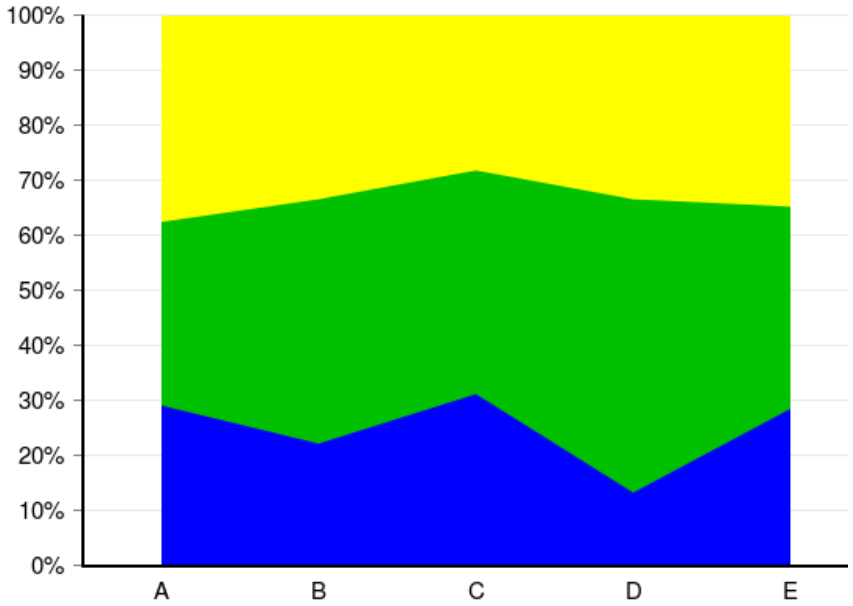


Photo Credit: https://commons.wikimedia.org/wiki/File:Broad_and_standard_mileage_operated_by_GWR.png
Photo Credit: https://commons.wikimedia.org/wiki/File:Charts_SVG_Example_12_-_Stacked_100%25_Area_Chart.svg
Photo Credit: https://commons.wikimedia.org/wiki/File:Pie-chart.jpg

DISTRIBUTIONS

1. NORMAL DISTRIBUTION

- A normal distribution is known as the bell curve because it looks like a bell!
- The density curve is symmetrical.
- Normal distribution is defined by its mean and standard deviation.
- Normal distribution is centered about its mean, with standard deviation indicating its spread.
- At point x, the height is represented as follows:

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

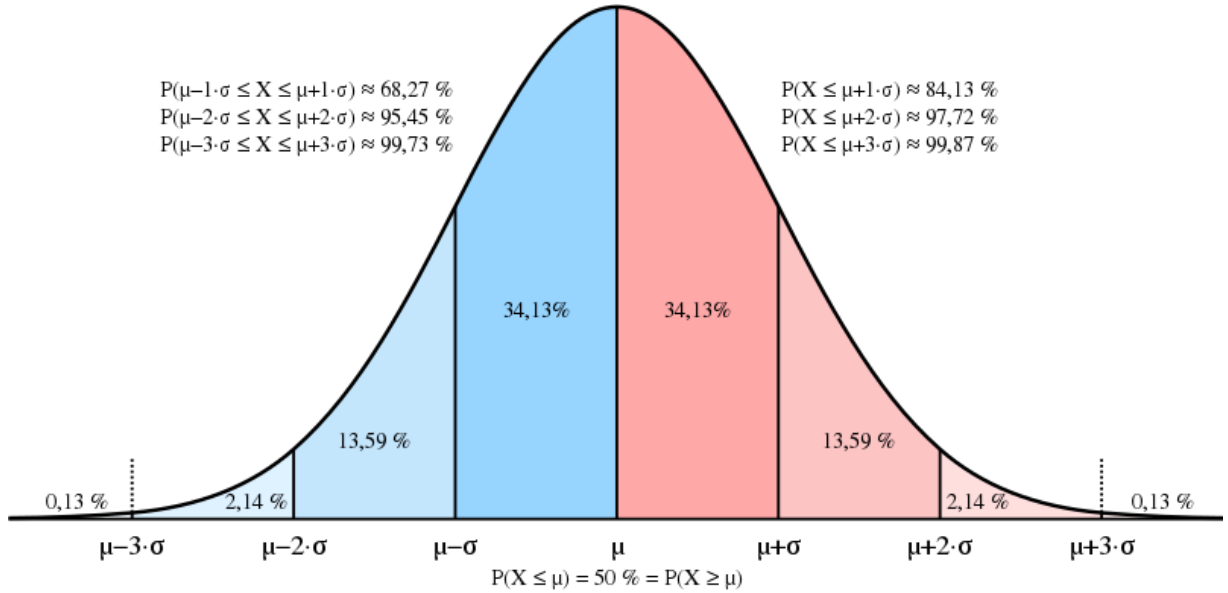


Photo Credit: https://commons.wikimedia.org/wiki/File:Normal_Distribution_Sigma.svg

1. STANDARD NORMAL DISTRIBUTION

- The Standard Normal distribution curve has:
  - Mean = 0
  - Standard deviation = 1

- We can convert data that is normally distributed to make it follow a standard normal by subtracting the mean and dividing by the standard deviation.

$$Z = \frac{X - \mu}{\sigma}$$

- For normally distributed data:
  - 68.3% of observations are within 1 standard deviation from the mean (-1,1).
  - 95% of observations are within 2 standard deviations of the mean (-2,2).
  - 99.7% of observations are within 3 standard deviations of the mean, interval (-3,3).

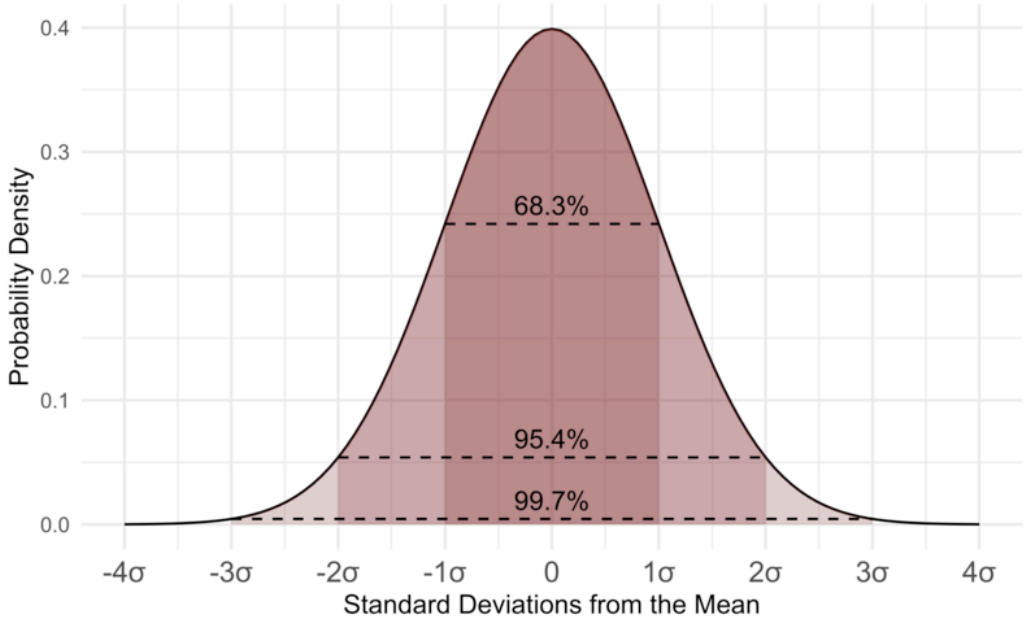


Photo Credit: https://commons.wikimedia.org/wiki/File:Standard_Normal_Distribution.png

2. POISSON DISTRIBUTION

- Poisson distribution is the discrete probability distribution of the number of events that occur in a specified period of time.
- Poisson distribution is extremely helpful for planning purposes as it enable managers to analyze customer behaviour as they visit a restaurant or store for example.
- Example: a restaurant manager want to know how many customers visit the store. He knows that the average visitors are 5 but actual number can change drastically.
- A Poisson distribution enable the manager to analyze the probability of various events.
    - Probability of 0 customers coming to the restaurant.
    - Probability of 7 or more customers visiting the restaurant.

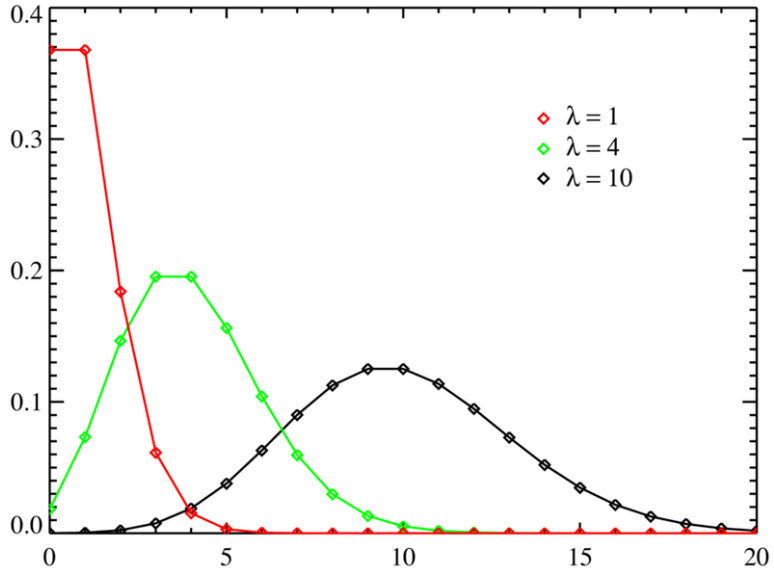


Photo Credit: https://commons.wikimedia.org/wiki/File:Poisson_distribution_PMF.png

2. POISSON DISTRIBUTION

- When to apply Poisson distribution?
  - Events are independent, If one event takes place, it does not impact the probability of another event taking place as well.
  - Rate of event occurrence is constant
  - Probability of an event taking place is proportional to time period.

- Poisson distribution formula:

$$P(x;\ \mu) = (e^{-\mu})\ (\mu^{x})\ /\ x!$$

2. POISSON DISTRIBUTION

EXAMPLE:
You are a manager of car dealership and the average number of cars sold is 2 cars per day. What is the probability that exactly 5 cars will be sold tomorrow?

Solution

- Since we have 2 cars sold per day, μ = 2.
- Since we want to know the likelihood that 5 cars being sold tomorrow, x = 5.
- We know that e = 2.71828 (constant).

Substitute in Poisson formula as follows:

$$P(x;\ \mu) = (e^{-\mu})\ (\mu^{x})\ /\ x!$$
$$P(5;\ 2) = (2.71828^{-2})\ (2^{5})\ /\ 5!$$
$$P(5;\ 2) = 0.036$$

Thus, the probability of selling 5 cars tomorrow is 0.036.

3. BINOMIAL DISTRIBUTION

- A binomial distribution measures the probability of success or failure outcome when the experiment is repeated several times (ex: outcomes of taking the AWS Machine Learning exam is: pass or fail).
- There are only two possible outcomes with fixed probabilities summing to one.
- For example, a coin toss has only two possible outcomes: heads or tails where the probability of each event is exactly = 0.5.
- For N trials, and probability = π, the binomial distribution is calculated as follows:

$$P(x) = \frac{N!}{x!(N-x)!}\pi^{x}(1-\pi)^{N-x}$$

- where P(x) is the probability of x successes out of N trials, N is the number of trials, and π is the probability of success.

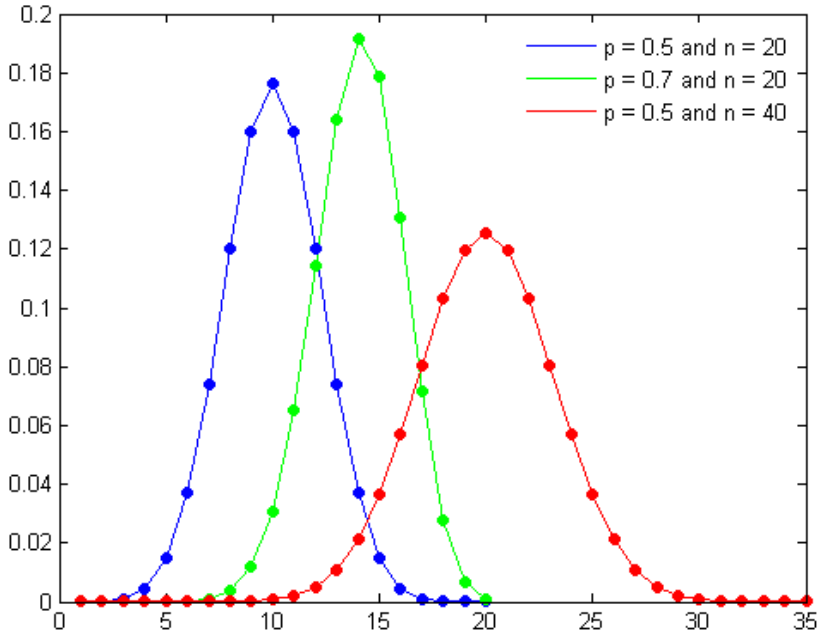


Photo Credit: https://en.wikipedia.org/wiki/File:Binomial_distribution_pdf.png

4. BERNOULLI DISTRIBUTION

- Bernoulli distribution is a special case of the binomial distribution for n = 1.
- Simply put, it is a binomial distribution with a single trial (one coin toss).
- Bernoulli distribution is a discrete probability distribution has only two outcomes (“Success” or a “Failure”).
- For example, when coin flipping:
  - Probability of head (success) = 0.5
  - Probability of tail (failure) = 1 – P = 0.5
- The probability of a failure is labeled on the x-axis as 0 and success is labeled as 1.
- As shown in figure, the probability of success (1) is 0.7, and the probability of failure (0) is 0.3:
-

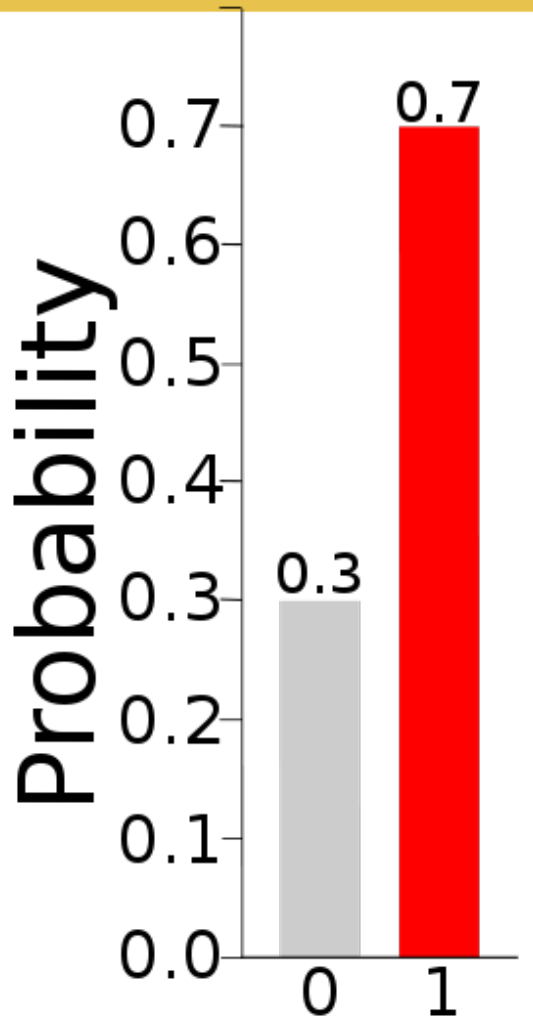


Photo Credit: https://commons.wikimedia.org/wiki/File:Bernoulli_0.7.svg

TIME SERIES

TIME SERIES: TRENDS

- In any time series, there are 4 important components:
  1. **Level:** average value in the series
  2. **Trend:** if the series is increasing or decreasing in value
  3. **Seasonality:** repeating short-term cycle in the series
  4. **Noise:** non-systematic random variation component

TIME SERIES: MULTIPLICATIVE VS. ADDITIVE MODELS

- There are two types of time series models: additive and multiplicative.
  - In additive models, the seasonality, trend and error components are added.
  - In multiplicative models, these components are multiplied.

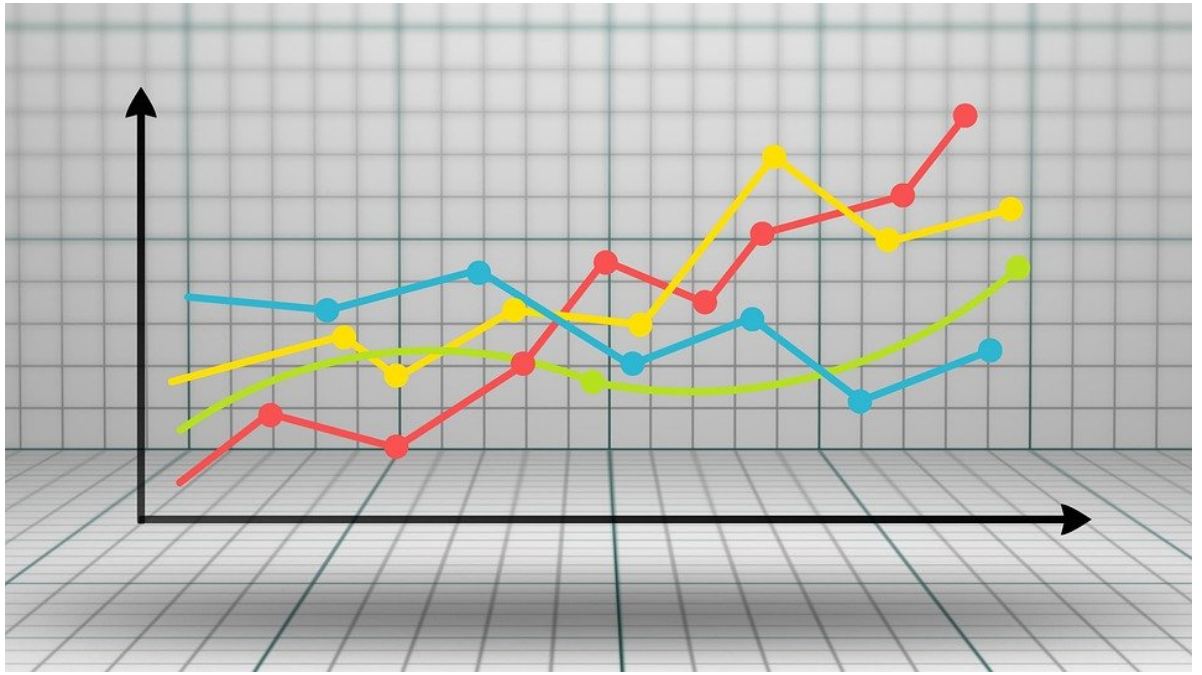


Photo Credit: https://pixabay.com/illustrations/graph-diagram-growth-written-report-3033203/