MACHINE LEARNING ESSENTIALS

Notes by Aniket Sahoo - Part I

Disclaimer : These are the personal notes prepared by me while undergoing the PGDMLAI course by Upgrad for my future reference. It includes all the topics (both mandatory and optional sections) and may be a couple of extra topics not included in the course from various scholarly articles over the internet. Please do use it for your personal reading only and refrain from sharing it over public domains such as LinkedIn, Github, Facebook, GoogleDrive etc as it might infringe upon various copyrights. I have tried my best to avoid any errors, still if you find any please let me know by dropping a mail to sahooaniket@gmail.com so that it can be rectified.

CONTENTS

1. BASICS

PYTHON FOR DATA SCIENCE

1.1.1. PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. It is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a batteries included language due to its comprehensive standard library. The language's core philosophy can be summarized as follows,

1. Beautiful is better than ugly.
2. Explicit is better than implicit.
3. Simple is better than complex.
4. Complex is better than complicated.
5. Readability counts.

A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Anaconda is a widely used free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution includes data-science packages suitable for Windows, Linux, and MacOS.

Installing Python with Anaconda

1. Search Anaconda on the browser and open the link with the address https://anaconda.org/.
2. Download Anaconda from the Anaconda homepage by selecting the operating system configuration.
3. Install Anaconda.
4. Create your own virtual python environment using the following commands on the conda command prompt.

```
(base)$ pip install virtualenv
(base)$ virtualenv -p /usr/env_path virtualenv_name
(base)$ source virtualenv_name/bin/activate
(virtualenv_name)$ pip install pandas
(virtualenv_name)$ deactivate
```

5. Open Jupyter Notebook on your browser and start coding.

Python Code - Basic Coding

```
Data Types
>> sentence = '  This is my first line of code in python  '
>> print(sentence)              # This is my first line of code in python
>> sentence.upper()             # '  THIS IS MY FIRST LINE OF CODE IN PYTHON  '
>> sentence.lower()             # '  this is my first line of code in python  '
>> sentence.strip()             # 'this is my first line of code in python'
```

```
>> sentence.lstrip()           # 'this is my first line of code in python  '
>> sentence.rstrip()           # '  this is my first line of code in python'
>> a, b = 6, 7
>> print(a + b, a - b, a * b, a / b,  a // b, a % b, a ** b)
   # 13 -1 42 0.8571428571428571 0 6 279936
```

Slicing

```
>> sentence[0]                 # 'T'
>> sentence[33:39]             # 'python'
>> sentence[:29]               # 'this is my first line of code'
>> sentence[:-10]              # 'this is my first line of code'
>> sentence[33:]               # 'python'
>> sentence[-6:]               # 'python'
>> sentence[0:39:2]            # 'Ti sm is ieo oei yhn'
```

List

```
>> empty_list = []
>> DA_languages = ['R','Python', 'SAS', 'Scala', 42]
>> DA_languages[0]             # R
>> DA_languages[-1]            # 42
>> DA_languages[1:3]           # ['Python', 'SAS']
>> DA_languages.append('Java')  # ['R', 'Python', 'SAS', 'Scala', 42, 'Java']
>> DA_languages.pop()          # 'Java'
>> DA_languages.pop(2)         # 'SAS'
>> DA_languages.append('SAS')   # ['R', 'Python', 'Scala', 42, 'SAS']
>> DA_languages.remove('SAS')   # ['R', 'Python', 'Scala', 42]
>> new_list = DA_languages
>> another_list = DA_languages.copy()
>> print(id(DA_languages))      # 1419340537672
>> print(id(new_list))          # 1419340537672
>> print(id(another_list))      # 1419340535112
>> sentence.split()
   # ['This', 'is', 'my', 'first', 'line', 'of', 'code', 'in', 'python']
>> sentence.split('i')
   # ['  Th', 's ', 's my f', 'rst l', 'ne of code ', 'n python  ']
>> '_'.join(sentence.split())   # This_is_my_first_line_of_code_in_python
>> nums_1, nums_2 = [1,2], [3,4]
>> nums_1*3                    # [1, 2, 1, 2, 1, 2]
>> nums_1.extend(nums_2)       # [1, 2, 3, 4]
>> nums = nums_2 + nums_1      # [3, 4, 1, 2]
>> len(nums)                   # 4
>> sorted(nums)                # [1, 2, 3, 4]
>> list(reversed(nums))        # [2, 1, 4, 3]
>> max(nums)                   # 4
>> min(nums)                   # 1
>> nums_1.append(nums_2)       # [1, 2, [3, 4]]
>> nums_1[0]                   # 1
>> nums_1[2]                   # [3, 4]
```

List Comprehension

```
>> squares_list = [x**2 for x in range(1,10)]
   # [1, 4, 9, 16, 25, 36, 49, 64, 81]
>> single_word_list  = [word for word in sentence.split()]
   # ['This', 'is', 'my', 'first', 'line', 'of', 'code', 'in', 'python']
```

Dictionary

```
>> empty_dictionary = {}
>> bio_data = {'Name': 'Bob Marley', 'Age':35, 'Height':'5.6 ft', 'Hobby': 'Music'}
>> bio_data['Hobby']                  # 'Music'
>> bio_data.get('Profession','NA')    # 'NA'
>> bio_data['Profession'] = 'Singer'
   # {'Name': 'Bob Marley', 'Age': 35, 'Height': '5.6 ft', 'Hobby': 'Music',
'Profession': 'Singer'}
>> bio_data.keys()     # ['Name', 'Age', 'Height', 'Hobby', 'Profession']
>> bio_data.values()   # ['Bob Marley', 35, '5.6 ft', 'Music', 'Singer']
>> new_dictionary = dict(Country='Jamaica', Songs=['One Love','Misty Morning'])
>> bio_data.update(new_dictionary)
   # {'Name': 'Bob Marley', 'Age': 35, 'Height': '5.6 ft', 'Hobby': 'Music',
'Profession': 'Singer', 'Country': 'Jamaica', 'Songs': ['One Love', 'Misty Morning']}
>> del bio_data['Songs']
   # {'Name': 'Bob Marley', 'Age': 35, 'Height': '5.6 ft', 'Hobby': 'Music',
'Profession': 'Singer'}
```

Dictionary Comprehension

```
>> squared_dict = {num : num**2 for num in range(0, 25)}
   # {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
>> even_sq_dict = {num:square for num, square in squared_dict.items() if num%2==0}
   # {0: 0, 2: 4, 4: 16, 6: 36, 8: 64}
```

Set

```
>> students_set_1 = set(['A','A','B','C','C','B'])   # {'A', 'B', 'C'}
>> students_set_2 = set(['A','E','D','E','A'])       # {'A', 'D', 'E'}
>> students_set_1.intersection(students_set_2)       # {'A'}
>> students_set_1.union(students_set_2)              # {'A', 'B', 'C', 'D', 'E'}
>> students_set_1.difference(students_set_2)         # {'B', 'C'}
```

Tuple

```
>> city_tuple = ('Mumbai', 18.9949521, 72.8141853)
   # ('Mumbai', 18.9949521, 72.8141853)
```

Functions

```
>> def square(num):
>>     out = num**2
>>     return(out)

>> def addition(*args):
>>     return(sum(args))

>> def factorial(n):
>>     if n>1:
>>         return n*factorial(n-1)
>>     else:
>>         return n

>> product = lambda x, y : x*y

>> list_of_names = ['nikola', 'james', 'albert']
>> list_of_names2 = ['tesla','watt','einstein']
>> proper = lambda x, y: x[0].upper()+x[1:] +' '+ y[0].upper()+y[1:]
```

```
>> list(map(proper, list_of_names,list_of_names2))
   # ['Nikola Tesla', 'James Watt', 'Albert Einstein']

>> divby3 = lambda x:  x % 3 == 0
>> list_of_nums = [3,4,7,8,9,5,6]
>> filter(divby3, list_of_nums)                        # [3, 6, 9]
>> reduce(lambda x,y: x if x>y else y, list_of_nums)   # 9
```

1.1.2. NUMPY

NumPy (pronounced NUM-pee) is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It stands for numerical python. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. The most basic object in NumPy is the ndarray, or simply an array which is an n-dimensional, homogeneous array. By homogenous, one means that all the elements in a NumPy array have to be of the same data type, which is commonly numeric (float or integer).

Installing Numpy

One can install numpy using the following command on the conda command prompt.

```
(base)$ source virtualenv_name/bin/activate
(virtualenv_name)$ pip install numpy
```

Python Code - Numpy

```
>> import numpy as np

Array Creation
# axis = 0 refers to the rows
# axis = 1 refers to the columns
>> np.array([2, 4, 5, 6, 7, 9])         # 1-D array
>> np.array([[2, 3, 4], [5, 8, 7]])     # 2-D array
>> np.ones((5, 3))                      # Array of ones
>> np.ones((5, 3), dtype = np.int)      # Change dtype (default float64)
>> np.zeros(4, dtype = np.int)          # Array of zeros
>> np.random.random([3, 4])             # Array of random numbers
>> np.arange(10, 100, 5)                # Array of numbers 10 to 100 with a step of 5
>> np.linspace(15, 18, 25)              # Array of length 25 between 15 and 18
>> np.full((4,3), 7)                    # Array of 7's
>> np.tile(some_array, 3)               # Array of repeating sequence
>> np.eye(3, dtype = int)               # Array of identity matrix
>> np.random.randint(0, 10, (4,4))      # Array of random integers ranging from 0 to 9

Array Inspection
>> rand_array.shape                     # shape
>> rand_array.dtype                     # dtype
>> rand_array.ndim                      # dimensions
>> rand_array.itemsize                  # itemsize
```

Array Slicing

```
>> array_1d[2]                          # Third element
>> array_1d[2:]                         # Third element onwards
>> array_1d[:3]                         # First three elements
>> array_1d[2:7]                        # Third to seventh elements
>> array_1d[0::2]                       # Subset starting 0 at increment of 2
>> array_2d[2, 1]                       # Third row second column
>> array_2d[1, :]                       # Second row, and all columns
>> array_2d[:, 2]                       # All rows and the third column
>> array_2d[:, :3]                      # All rows and the first three columns
```

Array Manipulation

```
>> some_array.reshape(2, 3, 4)          # Array reshape
>> some_array.reshape(4, -1)            # Array reshape with automatic dimensions
>> some_array.T                         # Array Transpose
>> np.vstack((array_1, array_2))        # Array vertical stacking
>> np.hstack((array_1, array_2))        # Array horizontal stacking
```

Array Operations

```
>> array_1 * array_2                    # multiplication
>> some_array ** 2                      # squared
>> np.sin(some_array)                   # sin
>> np.exp(some_array)                   # exponential
>> np.log(some_array)                   # logarithm
>> f = np.vectorize(lambda x: func(x)) # custom function
>> f(some_array)                        # apply custom function
>> some_array.T                         # transpose
>> np.linalg.matrix_rank(some_array)    # rank of array
>> np.linalg.inv(some_array)            # inverse
>> np.linalg.det(some_array)            # determinant
>> np.add(array_1, array_2)             # addition
>> np.subtract(array_1, array_2)        # subtraction
>> np.dot(array_1, array_2)             # multiplication
>> np.divide(array_1, array_2)          # division
>> eigen_val, eigen_vec = np.linalg.eig(some_array)
                                        # eigen operation
```

1.1.3. PANDAS

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. The name is derived from the term “panel data”, an econometrics term for data sets that include observations over multiple time periods for the same individuals. One shall use Pandas heavily for data manipulation, visualisation, building machine learning models, etc. There are two main data structures in Pandas - Series and Dataframes. The default way to store data is dataframes, and thus manipulating dataframes quickly is probably the most important skill set for data analysis.

Installing Pandas

One can install pandas using the following command on the conda command prompt.

```
(base)$ source virtualenv_name/bin/activate
(virtualenv_name)$ pip install pandas
```

Python Code - Pandas

```
>> import pandas as pd

Creation
# A pandas series
>> pd.Series([2, 4, 5, 6, 9])

# A pandas series with explicit indexing
>> pd.Series([0, 1, 2], index = ['a', 'b', 'c'])

# A pandas dataframe
>> pd.DataFrame({'name': ['Vinay', 'Kushal', 'Aman', 'Saif'], 'age': [22, 25, 24,
28], 'occupation': ['engineer', 'doctor', 'data analyst', 'teacher']})

# A pandas dataframe by importing files
>> data = pd.read_csv('path/file.csv')

Structure Changing
>> data.set_index('col_1', inplace = True)            # set col_1 as index
>> data.sort_index(axis = 0, ascending = False)       # sort by index
>> data.sort_values(by = 'col_1', ascending = False)  # sort by col_1

Inspection
>> data.shape                                         # shape
>> data.head()                                        # top 5 rows
>> data.tail()                                        # bottom 5 rows
>> data.info()                                        # metadata summary
>> data.describe()                                    # statistical summary
>> data.memory_usage()                                # memory consumed by columns
>> data.columns                                       # columns
>> data.values                                        # values as an array
>> pd.options.display.max_info_columns = 100          # display 100 columns

Slicing
>> data['col_1']                        # series of col_1 data
>> data.col_1                           # series of col_1 data
>> data[['col_1', 'col_2', 'col_3']]    # dataframe of col_1, col_2
>> data[2:7]                            # rows from indices 2 to 6
>> data[5::2]                           # alternate rows from index 5

Index based slicing
>> data.iloc[2, 4]                      # single element 3rd row and 5th column
>> data.iloc[5]                         # single row and all  columns
>> data.iloc[5, :]                      # single row and all  columns
>> data.iloc[[3, 7, 8]]                 # multiple rows using a list of indices
>> data.iloc[[3, 7, 8], :]              # multiple rows using a list of indices
>> data.iloc[[3, 7, 8], ]               # multiple rows using a list of indices
>> data.iloc[4:8]                       # rows using a range of integer indices
>> data.iloc[4:8, :]                    # rows using a range of integer indices
>> data.iloc[4:8, ]                     # rows using a range of integer indices
>> data.iloc[:, 2]                      # single column
>> data.iloc[:, 3:8]                    # single column
>> data.iloc[3:6, 2:5]                  # multiple rows and columns
```

Label based slicing

```
>> data.loc[2, 'Sales']                # single element row label 2 and column sales
>> data.loc[5]                         # single row with label 5
>> data.loc[5, :]                      # single row with label 5
>> data.loc[[3, 7, 8]]                 # multiple rows with label 3, 7, 8
>> data.loc[[3, 7, 8], :]              # multiple rows with label 3, 7, 8
>> data.loc[4:8]                       # multiple rows using a range of labels
>> data.loc[4:8, ]                     # multiple rows using a range of labels
>> data.loc[4:8, :]                    # multiple rows using a range of labels
>> data.loc[[1, 2], 'Sales':'Profit']  # multiple rows using labels and columns
>> data.loc[[True, True, False, True]] # multiple rows corresponding to True
```

Condition based slicing

```
>> data.col_1 > 3000                   # all rows where col_1 > 3000
>> data.loc[data.col_1 > 3000]         # all rows where col_1 > 3000
>> data.loc[data['col_1'] > 3000, :]   # all rows where col_1 > 3000
>> data.loc[(data.col_1 == 3000), :]   # all rows where col_1 = 3000
>> data.loc[(data.col_1 != 3000), :]   # all rows where col_1 <> 3000
>> data.loc[(data.col_1 > 2000) & (data.col_1 < 3000) & (data.col_2 > 100), :]
# all rows where 2000 < col_1 < 3000 and col_2 > 100
>> data.loc[(data.col_1 > 2000) | (data.col_1 > 100), :]
# all rows where 2000 < col_1 or col_2 > 100
>> data.loc[data['col_1'].isin(list_of_values), :]
# all rows where col_1 having values in list_of_values
>> data.loc[(data.col_1 > 2000) & (data.col_1 < 3000) & (data.col_2 > 100), ['col_1',
'col_2', 'col_3']]
# all rows where 2000 < col_1 < 3000 and col_2 > 100 and only selected columns
```

Operations

```
>> pd.merge(data_1, data_2, how='inner', on=['col_1', 'col_2'])
# inner join of dataframes on a specific columns
>> pd.concat([data_1, data_2], axis = 0)
# concatenate dataframes one on top of the other
>> pd.concat([data_1, data_2], axis = 1)
# concatenate dataframes side by side
>> data_1.append(data_2)
# alternative to concatenate along the rows
>> data_1 + data_2
# add two dataframes (gives NaN when values not present in both)
>> data_1.add(data_2, fill_value = 0)
# add two dataframes (does not give NaN when values not present in both)
>> data['new_col'] = any_value
# add a new column
>> data['new_col'] = data['col_1']/data['col_2']
# add a new column
>> data.groupby(['col_1', 'col_2'])
# group the data by specific columns
>> data.groupby(['col_1', 'col_2'])['col_3'].sum()
# sum of col_3 of the grouped data
>> data['col_1'].apply(lambda x: function(x))
# applying an operation to a column
>> data.pivot_table(values = 'aggregation_col', index = 'group_by_row', columns =
'group_by_col', aggfunc = 'mean')
# pivot data
```

```
>> pd.melt(data, id_vars = ['col_1', 'col_2'], value_vars = ['col_3', 'col_4'])
# unpivot a pivoted data
>> data['col_1'].unique()
# unique data in a specific column
>> data['col_1'].quantile(0.75)
# 75th quantile of a specific column
>> data.corr()
# correlation data
>> pd.to_datetime(data['col_1'])
# change datatype of a specific column
```

1.1.4. WORKING WITH DATA

There are multiple ways of getting data into Python, depending on where the data is stored. The simplest case is when one has the data in CSV files, but often, one may need to get data from other formats, sources and documents, such as text files, relational databases, websites, APIs, PDF documents, etc. After the data is extracted one has to then deal with nuances that inevitably come while getting data from various sources i.e cleaning the data.

Python Code - Getting data from Text files

```
>> import pandas as pd

# import delimited data
>> data = pd.read_csv('path/file.csv', sep = 'separator', encoding = 'encoding')
```

Python Code - Getting data from Relational Databases

```
>> import pymysql

# create a connection object 'conn'
>> conn = pymysql.connect(host='localhost', user='user_name', passwd='password',
db='db_name')
# create a cursor object c
>> c = conn.cursor()
# execute a query using c.execute
>> c.execute('query')
# getting the data as a tuple
>> all_rows = c.fetchall()
# import the data to pandas
>> pd.DataFrame(list(all_rows), columns=['col_1', 'col_2', 'col_3'])
```

Python Code - Getting data from Websites

```
>> import requests, bs4

# scraping the data from website
>> req = requests.get(url)
>> soup = bs4.BeautifulSoup(req.text, 'html5lib')
>> soup.select('html_tag')
```

Python Code - Getting data from API's

```
>> import requests, re, json

# getting the data from api
>> data = re.sub('', '+', data)
>> url = 'complete_url_address&key={1}'.format(api_key)
>> req = requests.get(url)
>> r_dict = json.loads(req.text)
```

Python Code - Getting data from PDF files

```
>> import PyPDF2

# reading the pdf file
>> pdf_file_object = open('file_path', 'rb')     # open a file to read
>> pdf_reader = PyPDF2.PdfFileReader(pdf_file_object)   # read file
>> pdf_reader.numPages                           # number of pages in the PDF file
>> page_object = pdf_reader.getPage(page_no)     # read a specific page
>> page_object.extractText()                     # extract data from specific page
>> pdf_file_object.close()                       # close file object
# writing a pdf file
>> pdf_writer = PyPDF2.PdfFileWriter()
>> pdf_writer.addPage(page_object)               # add a specific page
>> new_pdf_file_object = open('file_path', 'wb') # open a file to write
>> pdf_writer.write(new_pdf_file_object)         # write to file
>> new_pdf_file_object.close()                   # close file object
```

Python Code - Cleaning Data

```
>> df.isnull()
# find all nulls
>> df.isnull().sum()
# find count of all nulls
>> df.isnull().any(axis=0)
# columns with at least one missing value
>> df.isnull().all(axis=0)
# columns with all missing values
>> df.isnull().sum(axis=0)
# sum of missing values in each column
>> df.isnull().all(axis=0).sum()
# how many columns have all missing values
>> df.isnull().any(axis=1)
# rows with at least one missing value
>> round(100*(df.isnull().sum()/len(df.index)), 2)
# percentage of missing values in columns
>> df.drop('col_1', axis=1)
# removing specific column
>> df.dropna(how='all', axis='columns')
# remove columns with missing values
>> df.fillna('value')
```

```
# fill missing data with specific data
>> df.fillna(df.mean())
# fill missing data with mean data
>> df.fillna(method='ffill', limit=1)
# fill missing data with preceding value
>> len(df[df.isnull().sum(axis=1) > 5].index)
# count of rows with > 5 missing values
>> df[df.isnull().sum(axis=1) <= 5]
# retaining rows with <= 5 missing values
>> df[~np.isnan(df['col_1'])]
# removing rows of specific column with missing data
>> df.loc[np.isnan(df['col_1']), ['col_1']] = df['col_1'].mean()
# imputing specific column by mean values
>> df.loc[np.isnan(df['col_1']), ['col_1']] = 'value'
# imputing specific column by specific values
>> df['col_1'].astype('category')
# converting to category type
>> df['col_1'].value_counts()
# frequencies of each category
>> df.drop_duplicates()
# removes duplicate rows
>> df.drop_duplicates(['col_1', 'col_2'], keep='first')
# removes duplicate rows based on specific columns
>> df.replace('regex', np.nana, regex=True)
# replace data based on regex
>> df.interpolate(method='linear', axis=0)
# interpolate missing values according to different methods
>> df['col_1'].apply(lambda x : str(x).encode('encoding', 'ignore').decode('ascii',
'ignore'))
# clean bomb characters
```

Python Code - Datetime Formatting

```
>> import datetime as dt
>> import pytz as tz

>> curr_dt = dt.date.today()
# current date = datetime.date(2016, 07, 26)
>> curr_dt.weekday()
# current week day = 1 (Monday 0, Sunday 6)
>> curr_dt.isoweekday()
# current week day = 2 (Monday 1, Sunday 7)
>> fut_dt = curr_dt + dt.timedelta(days=7)
# add 7 days to current date = datetime.date(2016, 8, 2)
>> new_dt = dt.date(2016,9,24)
# set new date = datetime.date(2016, 9, 24)
>> till_dt = new_dt - curr_dt
# days remaining till new date = datetime.timedelta(days=60)
>> till_dt.days
# only days remaining to given date = 60
>> till_dt.total_seconds()
# only seconds remaining to given date = 5184000.0
```

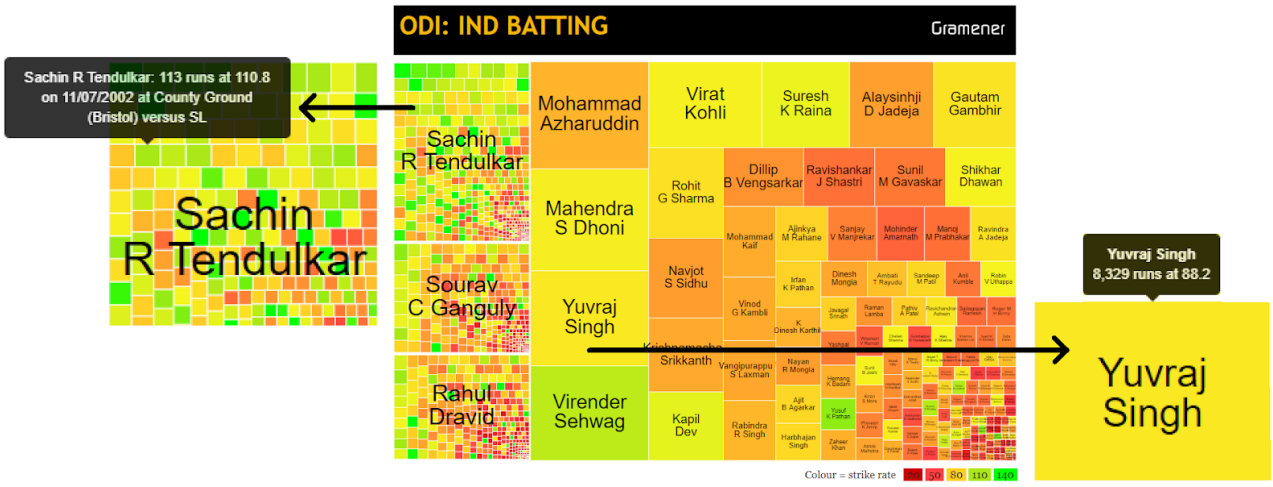
```
>> new_tm = dt.time(12, 30, 15, 100000)
# set new time = datetime.time(12, 30, 15, 100000)
>> new_tm.hour
# hours in time = 12
>> new_dt_tm = dt.datetime(2016, 7, 26, 12, 30, 15, 100000)
# set date new time = datetime.datetime(2016, 7, 26, 12, 30, 15, 100000)
>> new_dt_tm.year
# year in date time = 2016
>> dt_tm_now = dt.datetime.now()
# local date time = datetime.datetime(2016, 7, 26, 12, 30, 15, 100000)
>> dt_tm_now = dt.datetime.utcnow()
# utc date time = datetime.datetime(2016, 7, 26, 7, 00, 15, 100000)
>> all_tz = [timezones for timezones in tz.all_timezones]
# list of all time zones
>> new_dt_tm = dt.datetime(2016, 7, 26, 12, 30, 15, tzinfo = tz.UTC)
# utc date time = datetime.datetime(2016, 7, 26, 7, 00, 15, tzinfo=<UTC>)
>> arizona_dt_tm = new_dt_tm.astimezone(tz.timezone('US/Arizona'))
# arizona date time = datetime.datetime(2016, 7, 26, 5, 30, 15, tzinfo=<DstTzInfo
'US/Arizona' MST-1 day, 17:00:00 STD>)
>> dt_tm_now.strftime('date_time_format')
# date time in provided date_time_format
>> dt.datetime.strptime(datetime_str, 'date_time_format')
# converts string to datetime type
```

DATA VISUALIZATION IN PYTHON

1.2.1. DATA VISUALISATION

Data visualisation can add value to the information one wants to convey. Graphics and visuals, if used intelligently and innovatively, can convey a lot more than what raw data alone can convey. The various advantages of data visualization are as follows.

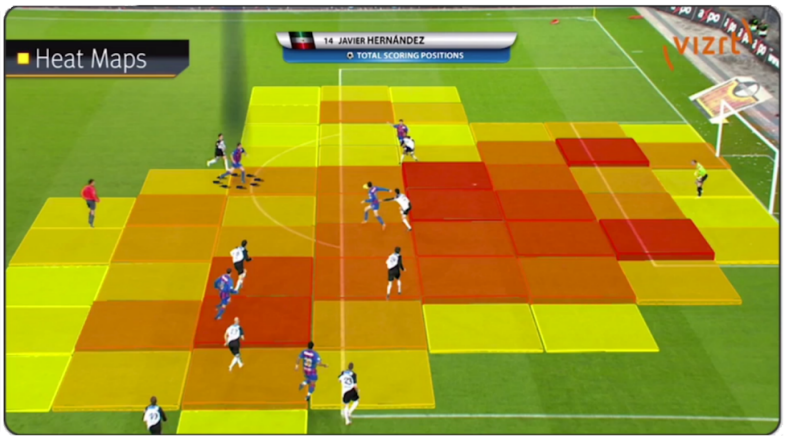
One can easily improve data density and improve the amount of information being conveyed. For example, imagine the difficulty in interpreting a spreadsheet with the score of each inning of each batsman being recorded along with his strike rate. The following data visualization helps one to figure out many insights just by looking at the plot.



Visualisation can help in visual exploratory analytics. For example, the following visualization helps in understanding the connections between different software and clustering them together based on their features.

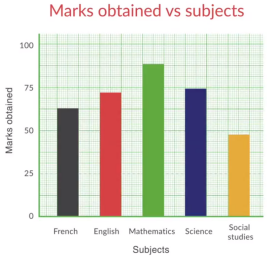
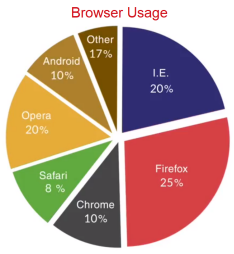
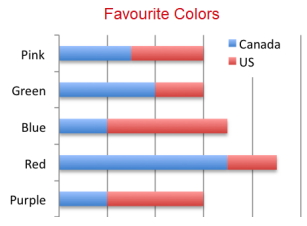
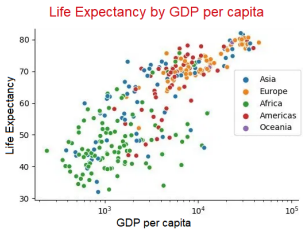
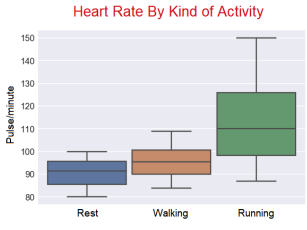
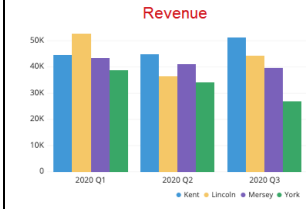
A picture is worth a thousand words. That's the power of visual imagery. A message which cannot be conveyed through a large set of texts and tabular data can easily be presented through visuals. One often uses graphics to make sense of large and complex sets of information. This makes data visualisation a very important step for data understanding. For example, the following visualization of a data from an incomprehensible tabular format in a heat map helped a football coach formulate the defence strategy for his team.

Player Name	X Grid	X Grid Prob	Y Grid	Y Grid Prob	Goals Density
Javier Hernández	13	0.976	98	0.876	7.98
Andrés Guardado	15	0.65	97	0.95	5.4
Oribe Peralta	13	0.77	99	0.97	6.9
Rafael Márquez	9	0.654	96	0.9654	6.4
Raúl Jiménez	7	0.914	98	0.914	2.1
Paul Aguilar	11	0.64	89	0.44	2.1
Miguel Layún	12	0.73	97	0.73	1.02
Héctor Herrera	9	0.84	95	0.84	0.97
Néstor Araujo	10	0.74	94	0.74	1.3
Jorge Torres Nilo	13	0.78	92	0.7	1.1
Guillermo Ochoa	11	0.96	3	0.97	0
Javier Hernández	13	0.976	98	0.876	7.98
Andres Iniesta	15	0.65	97	0.95	5.4
Zlatan Ibrahimovic	13	0.77	99	0.97	6.9
Radamel Falcao	9	0.654	96	0.9654	6.4
Robin van Persie	7	0.914	98	0.914	2.1
Andrea Pirlo	11	0.64	89	0.44	2.1
Yaya Toure	12	0.73	97	0.73	1.02
Edinson Cavani	9	0.84	95	0.84	0.97
Sergio Busquets	11	0.96	3	0.97	0



Understanding Basic Chart Types

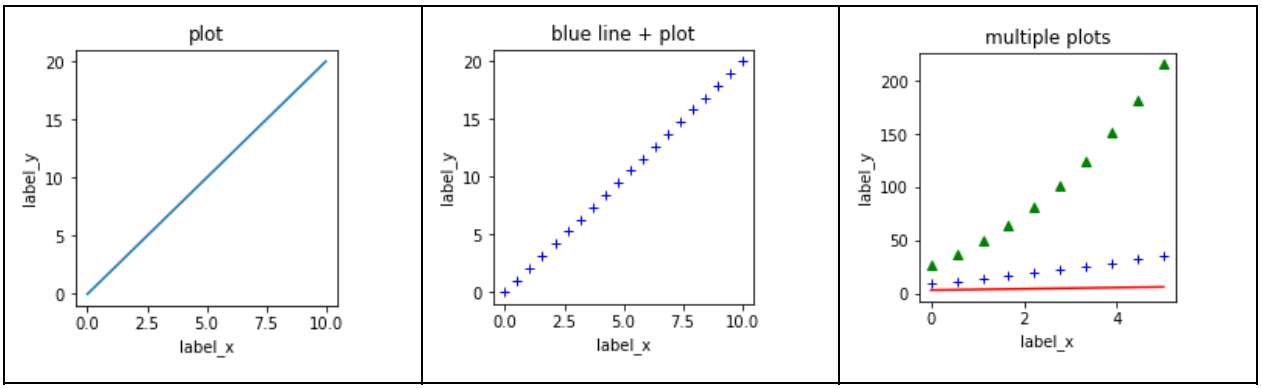
The pre-attentive attributes make graphics easy to understand whereas attentive attributes are relatively difficult to grasp. However, to make sense through visuals, it's important to first understand the different types of visuals and their common usage. The various types of basic plots which are most often are,

Line chart	Bar chart	Histogram	Pie chart
To represent a time-dependent trend	To represent a few dimensions on a linear scale	To represent frequencies of groups	To summarize the share of different components in an aggregate whole
			
Stacked Bar chart	**Scatter plot**	**Box plot**	**Grouped Bar chart**
To compare the share and contribution of categories across different sectors	To summarize the variation of data points across two parameters	To represent the quartile, percentile and outliers values	To represent different sub-groups among the main categories
			

Python Code - Matplotlib

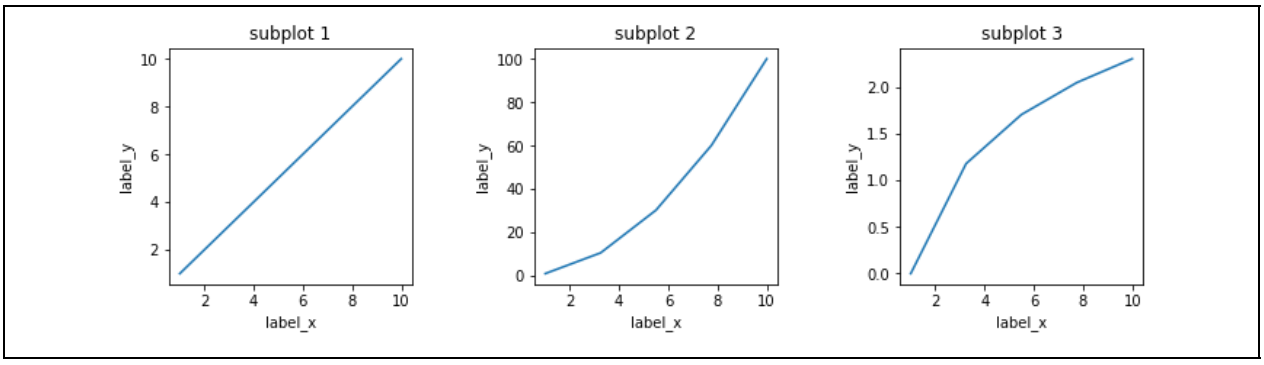
```
>> import matplotlib.pyplot as plt

Plots
>> plt.xlabel('label_x')                    # change x label
>> plt.ylabel('label_y')                    # change y label
>> plt.title('title')                       # change title
>> plt.xlim([initial_value, final_value])   # x-axis limits
>> plt.ylim([initial_value, final_value])   # y-axis limits
```

```
>> plt.plot(data_x, data_y)                    # plot two arrays
>> plt.yscale('log')                           # using y axis with log scale
>> plt.xticks([])                              # disable ticks in x axis
>> plt.show()                                  # show the plot
>> plt.savefig('output.png')                   # save plot as image
>> plt.plot(data_x, data_y, 'b+')              # plot color blue and line type '+'
>> plt.plot(data_x, data_y, 'b+', data_x, data_y, 'g^', data_x, data_y, 'r-')
                                               # multiple plots
```



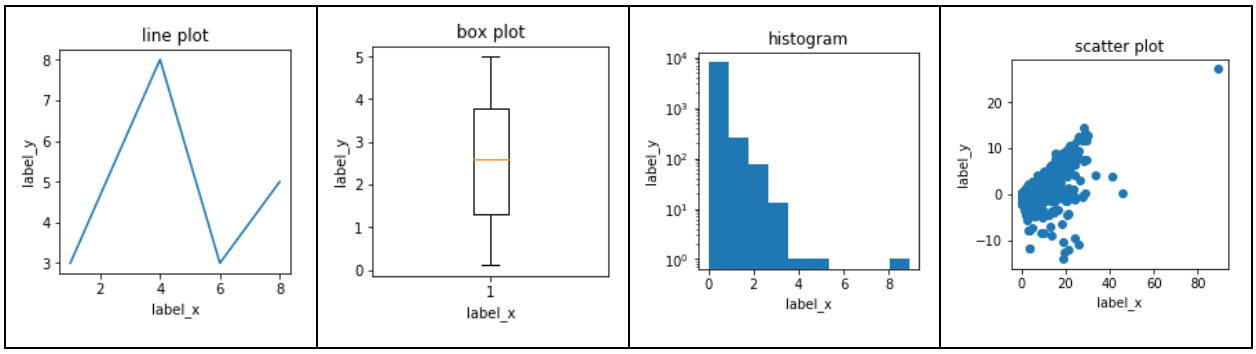
Subplots

```
>> plt.figure(1, (width, height))              # initiating new figure explicitly
>> plt.subplots_adjust(hspace=1, wspace=1)     # space between subplots
# subplot for 1 row 3 columns
>> plt.subplot(1, 3, 1)                        # set subplot 1
>> plt.title('subplot 1')                      # title for 1st position
>> plt.plot(data_x, data_y)                    # plot for 1st position
>> plt.subplot(1, 3, 2)                        # set subplot 2
>> plt.title('subplot 2')                      # title for 2nd position
>> plt.plot(data_x, data_y)                    # plot for 2nd position
>> plt.subplot(1, 3, 3)                        # set subplot 3
>> plt.title('subplot 3')                      # title for 3rd position
>> plt.plot(data_x, data_y)                    # plot for 3rd position
```



Plot Types

```
>> plt.plot(data_x, data_y)                    # line plot
>> plt.boxplot(data_x)                         # box plot
>> plt.hist(data_x)                            # histogram
>> plt.scatter(data_x, data_y)                 # scatter plot
>> image = plt.imread('image_path')            # read image
>> plt.imshow(image)                           # plot image
```



Pandas Plot

```
>> df.plot(kind='plot_type', x='col_1', legend=True, figsize=(width, height),
title='title')
```

1.2.2. DATA DISTRIBUTION

Data distributions tell how the data is distributed across values of different attributes of the data. These distributions could be across one variable, or more than one variable too, or categories or over time.

Univariate Distributions

A univariate distribution shows how data points of one variable are distributed. Univariate plots can be visualized using histograms, density plots, rug plots and box plots.

Bivariate Distributions

A bivariate distribution shows how two variables interact with each other. Bivariate plots can be visualized using scatter plots. One can also visualise pairwise relationships between multiple variables using heatmaps.

Categorical Distributions

It shows how the data is distributed across multiple segments or categories. This data can be visualized using various categorical plots like box plot, bar plot, count plot etc.

Time Series Distributions

The time-series data involves chronological information of one or more variables (one of the attributes in the data being time). One can simply observe the trend of a variable using a simple line plot. But, if one wants to spot interesting and hidden trends in a variable one can use any of the following two ways,

1. Plotting time on the x-axis and the value (usually aggregated using mean, median etc.) of a variable on the y-axis.

2. Plotting a heat map with year/ month/day along the axes and the values denoted by colour.

Python Code - Seaborn

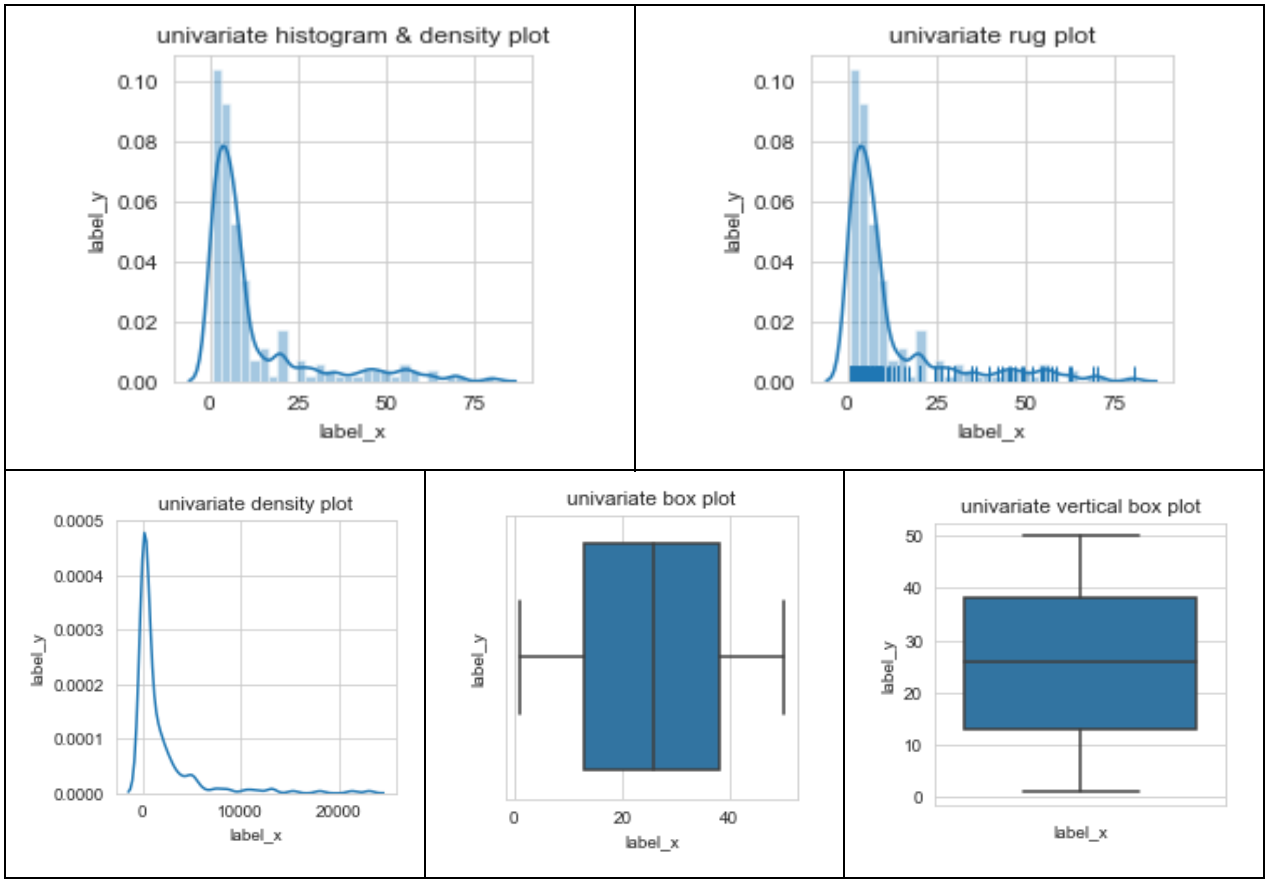
```
>> import seaborn as sns
```

Plots

```
sns.set_style('whitegrid')                          # set style
sns.set_context('paper', font_scale=1)              # set context
```

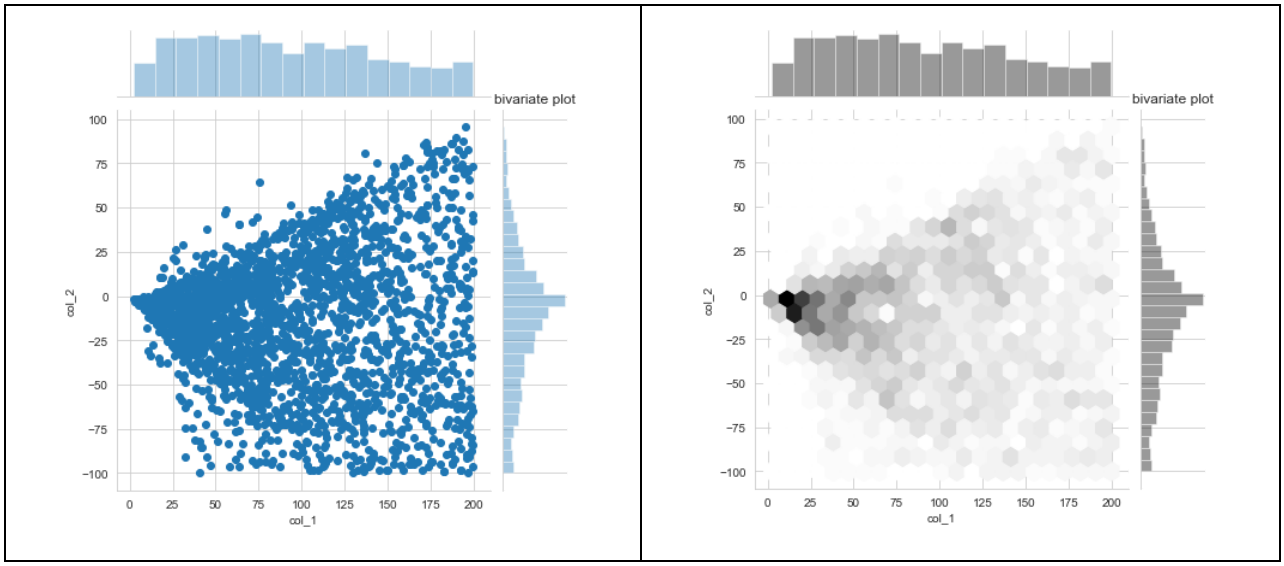
Univariate Distribution

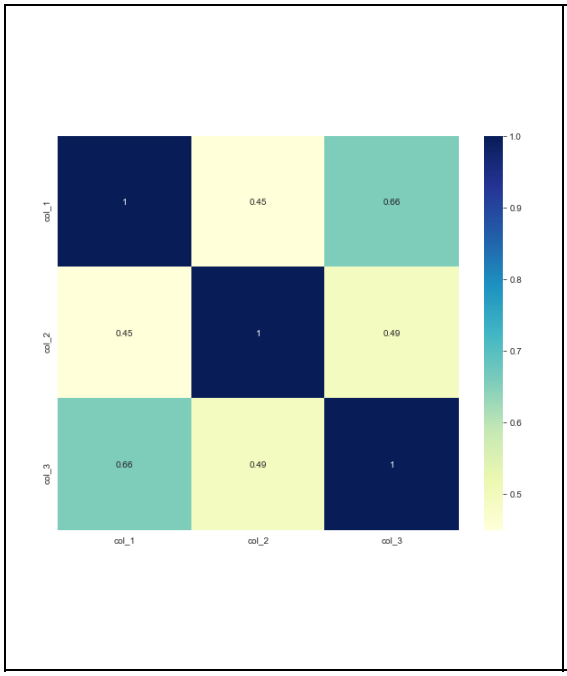
```
>> sns.distplot(df['col_1'])                    # density & histogram plot
>> sns.distplot(df['col_1'], hist=False)        # density plot
>> sns.distplot(df['col_1'], rug=True)          # rug plot
>> sns.boxplot(df['col_1'])                     # box plot
>> sns.boxplot(y=df['col_1'])                   # vertical box plot
```

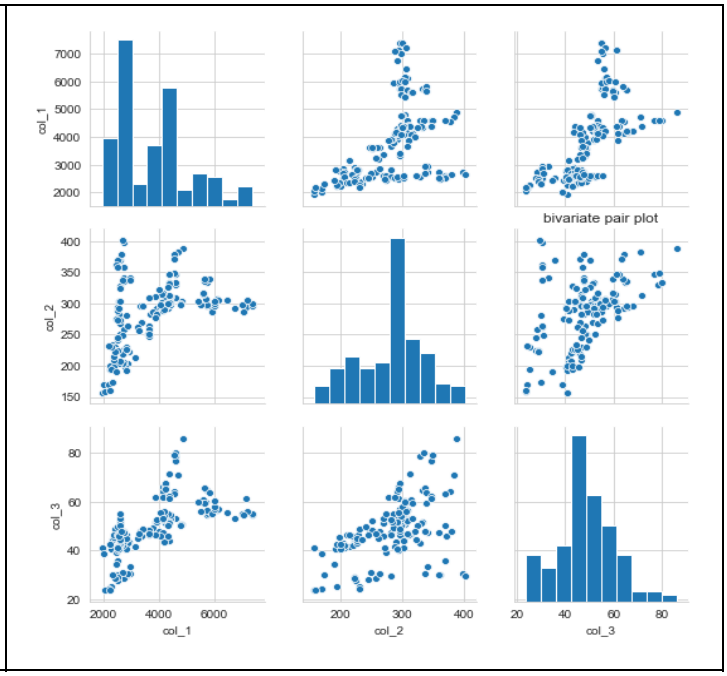


Bivariate Distribution

```
>> sns.jointplot('col_1', 'col_2', df)                    # joint plot
>> sns.jointplot('col_1', 'col_2', df, kind="hex", color="k") # joint plot
>> corr_matrix = df[['col_1', 'col_2', 'col_3']].corr()    # correlation matrix
>> sns.heatmap(corr_matrix, cmap="YlGnBu", annot=True)     # heat map
>> sns.pairplot(df[['col_1', 'col_2', 'col_3']])          # pairwise plot
```

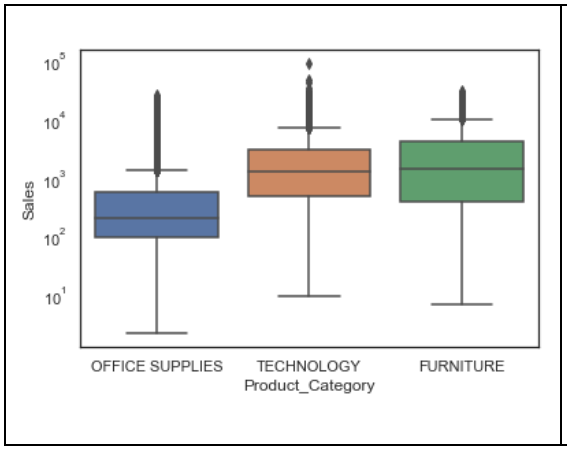


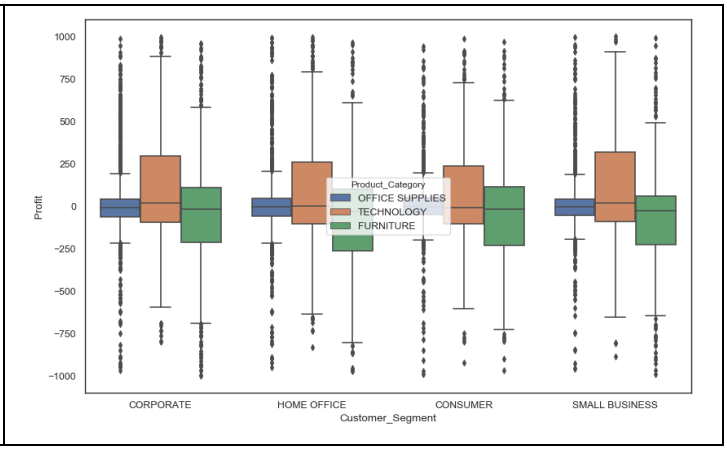




Categorical Distribution

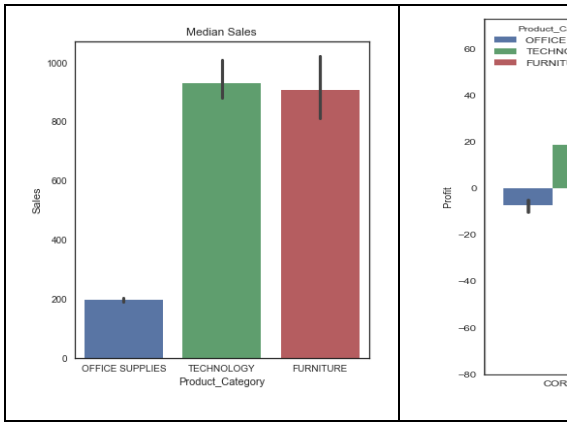
```
>> sns.boxplot(x='col_1', y='col_2', data=df)              # box plot
>> sns.boxplot(x='col_1', y='col_2', hue='col_3', data=df) # box plot
```

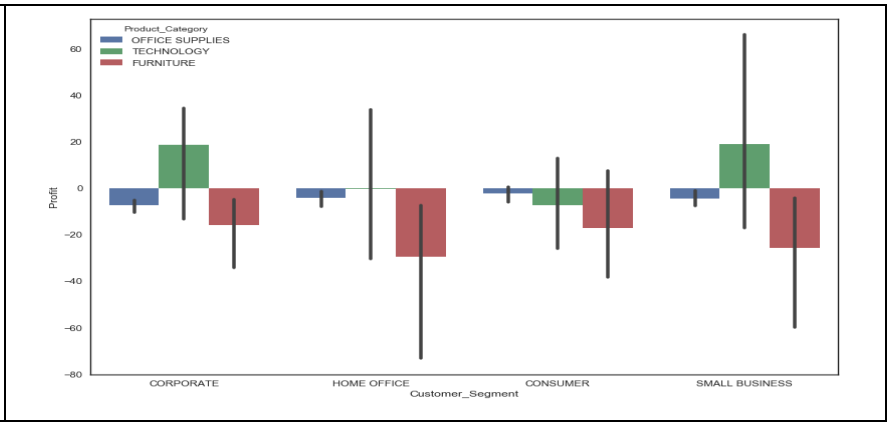


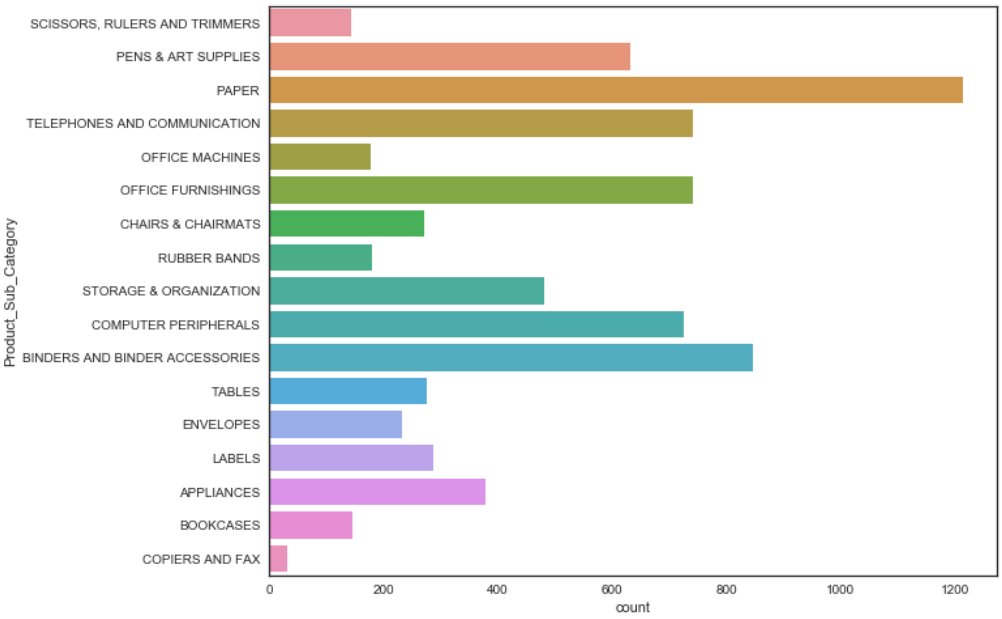


Categorical Distribution (aggregated values)

```
>> sns.barplot(x='col_1', y='col_2', data=df, estimator=np.median) # bar plot
>> sns.barplot(x='col_1', y='col_2', hue='col_3', data=df, estimator=np.median)
                                                                   # bar plot
>> sns.countplot(y='col_1', data=df)                             # count plot
```

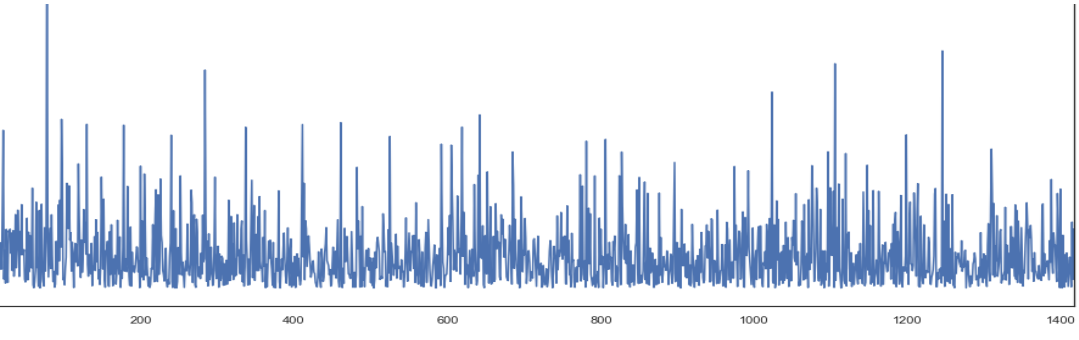


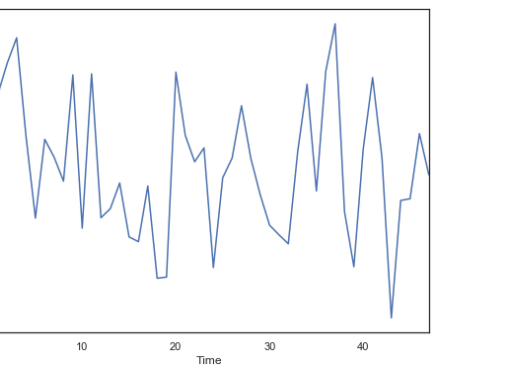


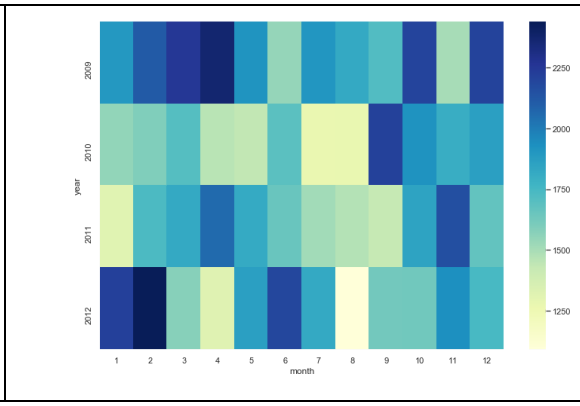


Time Series Distribution

```
>> sns.tsplot(data=df)                                          # time series plot
>> new_df = df.groupby(['year', 'month']).col_1.mean()          # group by year, month
>> sns.tsplot(new_df )                                          # time series plot
>> year_month_df = pd.pivot_table(df, values='col_1', index='year', columns='month',
aggfunc='mean')                                                 # pivot table
>> sns.heatmap(year_month_df , cmap="YlGnBu")                   # heat map
```







MATH FOR MACHINE LEARNING

1.3.1. VECTORS AND VECTOR SPACES

The ability to visualise data is one of the most useful skills to possess as a data science professional, and a solid foundation in linear algebra enables one to do that. In fact, some concepts and algorithms are quite easy to understand if one can visualise them as vectors and matrices, rather than looking at the data as lists and arrays of numbers. Linear Algebra is the workhorse of Data Science and ML. While training a machine learning model using a library (such as in R or Python), much of what happens behind the scenes is a bunch of matrix operations. The most popular deep learning library today, Tensorflow, is essentially an optimised (i.e. fast and reliable) matrix manipulation library. So is scikit-learn, the Python library for machine learning.

Vectors

1. Vectors are usually represented in two ways - as ordered lists, such as $x = [x_1, x_2, \ldots x_n]$ or or using the 'hat' notation, such as $x = x_1 \hat{i} + x_2 \hat{j} + x_3 \hat{k}$ where $\hat{i}, \hat{j}, \hat{k}$ represent the three perpendicular directions (or axes).

2. The number of elements in a vector is the dimensionality of the vector. For e.g. $x = [x_1, x_2]$ is two dimensional (2-D) vector , $x = [x_1, x_2, x_3]$ is a 3-D vector and so on.

3. The magnitude of a vector is the distance of its tip from the origin. For an n-dimensional vector $x = [x_1, x_2, \ldots x_n]$, the magnitude is given by, $||a|| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$.

4. A unit vector is one whose distance from the origin is exactly 1 unit. For e.g. the vectors $\hat{i}, \hat{j}, \hat{i}/\sqrt{2} + \hat{j}/\sqrt{2}$ are unit vectors.

Vector Operations

1. Vector Addition/Subtraction : It is the element-wise sum/difference of two vectors. Mathematically,

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{bmatrix}$$

2. Scalar Multiplication/Division : It is the element-wise multiplication/division of the scalar value. Mathematically,

$$a \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} ax_1 \\ ax_2 \\ \vdots \\ ax_n \end{bmatrix}$$

3. Vector Multiplication or Dot Product : It is the element-wise product of the two vectors. It is also known as the dot product of two vectors. The dot product of two vectors returns a scalar quantity. Mathematically,

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} . \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$$

Geometrically,

$$dot(\vec{x}, \vec{y}) = \|x\| \|y\| \cos\theta$$
$$where, \ \theta \ is \ the \ angle \ between \ two \ vectors$$

The dot product of two perpendicular vectors (also called orthogonal vectors) is 0. The dot product can be used to compute the angle between two vectors using the formula,

$$\cos\theta = \frac{\vec{x} . \vec{y}}{\|x\| \|y\|}$$

This simple property of the dot product is extensively used in data science applications. One such example is the spam detection, which is a popular application of machine learning, where the spam mails are separated from genuine mails. Spam detection algorithms make a decision based on the words in an email i.e. if the email contains phrases such as *"easy money"*, *"free!!"*, *"hurry up"* etc., it is more likely to be a spam mail. On the other hand, if it contains words such as *"meeting"*, *"powerpoint"*, *"client"* etc., it is probably genuine mail. Each mail is represented as a vector based on the words it contains. Each mail is then classified accordingly by checking for its similarity with known spam mails by finding the angle between the vector representation of these mails (the smaller the angle the more similar are the mails). This cosine similarity technique is an extremely useful technique and can be extended to any set of text documents. In fact, it is a very general technique used in a variety of machine learning techniques such as recommender systems, web and document search etc.

Vector Spaces

1. Basis Vector : A basis vector of a vector space V is defined as a subset $(v_1, v_2, \ldots v_n)$ of vectors in vector space V, that are linearly independent and span vector space V. Consequently, if $(v_1, v_2, \ldots v_n)$ is a list of vectors in vector space V, then these vectors form a vector basis if and only if every v in vector space V can be uniquely written as,

$$v = a_1 v_1, a_2 v_2, \ldots a_n v_n$$

The vectors $\hat{i}$ and $\hat{j}$ are chosen as the default basis vectors, though one can choose to have a completely different, valid choice of basis vectors.

2. Span : The span of two or more vectors is the set of all possible vectors that one can get by changing the scalars and adding them.

3. Linear Combination : The linear combination of two vectors is the sum of the scaled vectors.

4. Linearly Dependent : A set of vectors is called linearly dependent if any one or more of the vectors can be expressed as a linear combination of the other vectors.

5. Linearly Independent : If none of the vectors in a set can be expressed as a linear combination of the other vectors, the vectors are called linearly independent.

1.3.2. LINEAR TRANSFORMATIONS AND MATRICES

Matrices, is a time tested and powerful data structure used to perform numerical computations. Briefly, a matrix is a collection of values stored as rows and columns, i.e.

$$A = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

Matrices

1. Rows : Rows are horizontal. The matrix A has m rows. Each row itself is a vector, so they are also called row vectors.

2. Columns : Columns are vertical. The matrix A has n columns. Each column itself is a vector, so they are also called column vectors.

3. Entities : Entities are individual values in a matrix. For a given matrix A, value of row i and column j is represented as A_{ij}.

4. Dimensions : The number of rows and columns. For m rows and n columns, the dimensions are $(m \times n)$.

5. Square Matrices : These are matrices where the number of rows is equal to the number of columns, i.e $m = n$.

6. Diagonal Matrices : These are square matrices where all the off-diagonal elements are zero, i.e,

$$A = \begin{bmatrix} x_{11} & 0 & \cdots & 0 \\ 0 & x_{22} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & x_{mn} \end{bmatrix}$$

7. Identity Matrices : These are diagonal matrices where all the diagonal elements are 1, i.e,

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Matrix Operations

1. Matrix Addition/Subtraction : It is the element-wise sum/difference of two matrices. Mathematically,

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} + \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{bmatrix} = \begin{bmatrix} x_{11}+y_{11} & x_{12}+y_{12} & \cdots & x_{1n}+y_{1n} \\ x_{21}+y_{21} & x_{22}+y_{22} & \cdots & x_{2n}+y_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1}+y_{m1} & x_{m2}+y_{m2} & \cdots & x_{mn}+y_{mn} \end{bmatrix}$$

2. Matrix Multiplication/Division : It is the element-wise multiplication/division of the scalar value. Mathematically,

$$a \times \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} = \begin{bmatrix} ax_{11} & ax_{12} & \cdots & ax_{1n} \\ ax_{21} & ax_{22} & \cdots & ax_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ ax_{m1} & ax_{m2} & \cdots & ax_{mn} \end{bmatrix}$$

3. Matrix Multiplication or Dot Product : It is the element-wise product of the two matrices i.e the (i,j) element of the output matrix is the dot product of the i^{th} row of the first matrix and the j^{th} column of the second matrix. Mathematically,

$$\underset{(m \times n)}{\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}} \cdot \underset{(o \times p)}{\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1o} \\ y_{21} & y_{22} & \cdots & y_{2o} \\ \vdots & \vdots & \cdots & \vdots \\ y_{o1} & y_{o2} & \cdots & y_{op} \end{bmatrix}}$$

$$= \underset{(m \times p)}{\begin{bmatrix} x_{11}y_{11}+x_{12}y_{21}+\ldots+x_{1n}y_{o1} & \cdots & x_{11}y_{1p}+x_{12}y_{2p}+\ldots+x_{1n}y_{op} \\ x_{21}y_{11}+x_{22}y_{21}+\ldots+x_{2n}y_{o1} & \cdots & x_{21}y_{1p}+x_{22}y_{2p}+\ldots+x_{2n}y_{op} \\ \vdots & \cdots & \vdots \\ x_{m1}y_{11}+x_{m2}y_{21}+\ldots+x_{mn}y_{o1} & \cdots & x_{m1}y_{1p}+x_{m2}y_{2p}+\ldots+x_{mn}y_{op} \end{bmatrix}}$$

Not all matrices can be multiplied with each other. For the matrix multiplication AB to be valid, the number of columns in A should be equal to the number of rows in B. i.e for two matrices A and B with dimensions $(m \times n)$ and $(o \times p)$, AB exists if and only if $m = p$ and BA exists if and only if $o = n$. Matrix multiplication is not commutative i.e $AB \neq BA$.

4. Matrix Inverse : The inverse of a matrix A is a matrix such that $AA^{-1} = I\ (Identity\ Matrix)$.

5. Matrix Transpose : The transpose of a matrix produces a matrix in which the rows and columns are interchanged. Mathematically,

$$A^T = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{m1} \\ x_{12} & x_{22} & \cdots & x_{m2} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{nm} \end{bmatrix} \qquad where,\ A = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

Linear Transformations

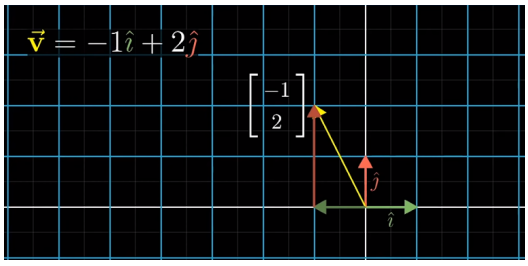
Any transformation can be geometrically visualised as the distortion of the n-dimensional space (it can be squishing, stretching, rotating etc.). The distortion of space can be visualised as a distortion of the grid lines that make up the coordinate system. Space can be distorted in several different ways. A linear transformation, however, is a special distortion with two distinct properties,

1. Straight lines remain straight and parallel to each other
2. The origin remains fixed

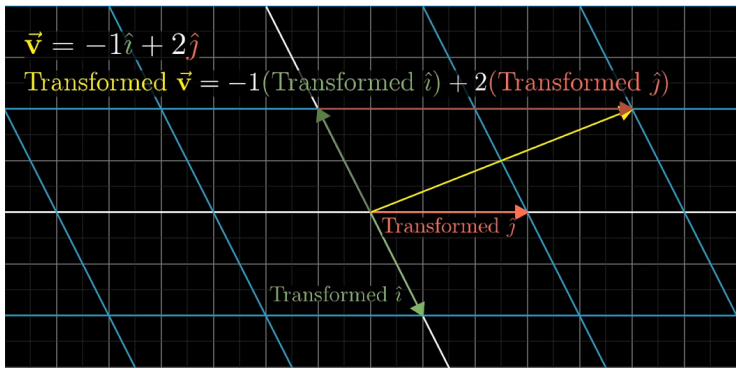
Consider a linear transformation where the original basis vectors $\hat{i}$ and $\hat{j}$ move to the new points, $\hat{i} = [1,-2]$ and $\hat{j} = [3,0]$. This means that $\hat{i}$ moves to $(1,-2)$ from $(1,0)$ and $\hat{j}$ moves to $(3,0)$ from $(0,1)$ in the linear transformation. This transformation simply stretches the space in the y-direction by three units while stretching the space in x-direction by two units and rotating it by sixty degrees in clockwise direction. One can combine the two vectors where $\hat{i}$ and $\hat{j}$ land and write them as a single matrix, i.e,

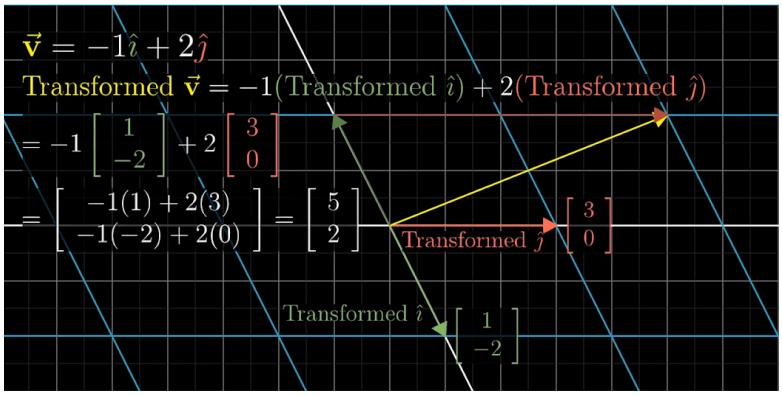
$$L = \begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix}$$

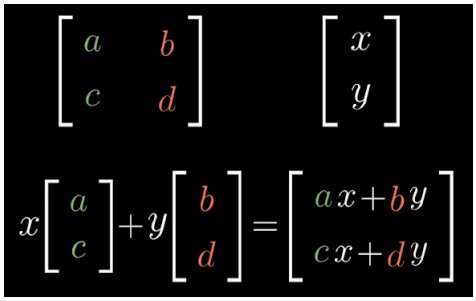
As can be seen, each of these vectors form one column of the matrix (and hence are often called column vectors). This matrix fully represents the linear transformation. Now, if one wants to find where any given vector v would land after this transformation, one simply needs to multiply the vector v with the matrix L, i.e $v_{new} = L.v$. It is convenient to think of this matrix as a function which describes the transformation, i.e it takes the original vector v as the input and returns the new vector v_{new}. The following figures represent the linear transformation.



Vector $v = -1\hat{i} + 2\hat{j}$ being multiplied with matrix L, i.e the transformed basis vectors.







Any vector v undergoing a linear transformation is a matrix multiplication.

Formally, a transformation is linear if it satisfies the following two properties,

1. Additivity or Distributivity, i.e $L(v+w) = L(v) + L(w)$.
2. Associativity of Homogeneity, i.e $L(cv) = cL(v)$ where c is a scalar.

Composite Transformation

One can also apply multiple linear transformations one after the other. For example, one can rotate the space 90 degrees counterclockwise, then apply positive shear, and then rotate it back again 90 degrees clockwise. Let's say these matrices are called A, B and C respectively. Mathematically, if one imagines these transformations being applied to a vector v, then the final vector would be, $v_{final} = C.B.A.v$. That is, one first applies A to vector v to get the matrix $v_{transformation\ A} = A.v$, then B to vector $A.v$ to get $v_{transformation\ B} = B.A.v$ and so on to finally get $v_{final} = C.B.A.v$. One can represent the matrix product of $C.B.A$ as another matrix $L = C.B.A$. The matrix L represents the three transformations done one after the other or in other words a composite transformation matrix (doing the three consecutive transformations is equivalent to the single transformation).

Determinants

The determinant of a matrix A, usually denoted as $|A|$ is a numerical value associated with a matrix. Mathematically,

$$|A| = x_{11} \cdot x_{21} - x_{12} \cdot x_{21} \qquad where,\ A = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}$$

$$|B| = x_{11} \cdot det\left(\begin{bmatrix} x_{22} & x_{23} \\ x_{32} & x_{33} \end{bmatrix}\right) - x_{12} \cdot det\left(\begin{bmatrix} x_{21} & x_{23} \\ x_{31} & x_{33} \end{bmatrix}\right) + x_{13} \cdot det\left(\begin{bmatrix} x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix}\right)$$

$$where,\ B = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$$

The determinant represents the magnitude by which the area (for 2D matrix), volume (for 3D matrix) and so on is scaled upon linear transformation. For example if the determinant of a 2D matrix is zero, then it represents a transformation that squishes the 2D space into a straight line or a single point.

System of Linear Equations

A system of linear equations is a set of linear equations involving the same set of variables. For example the n linear equations involving n variables $x_1, x_2, \ldots x_n$ will be of the form,

$$\begin{aligned} A_{11}x_1 + A_{12}x_2 + \ldots + A_{1n}x_n &= b_1 \\ A_{21}x_1 + A_{22}x_2 + \ldots + A_{2n}x_n &= b_2 \\ &\vdots \\ A_{n1}x_1 + A_{n2}x_2 + \ldots + A_{nn}x_n &= b_n \end{aligned}$$

Solving this system means finding a combination of $x_1, x_2, \ldots x_n$ that satisfies all the equations. One can solve this system of equations algebraically, but in most practical applications one will have to solve really large sets of equations and variables. Thus, one needs to automate the process of solving such systems. Matrices give a very nifty way to express and solve these equations. The equations above can be rewritten in the matrix form as,

$$Ax = b \qquad where,\ A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Now, solving the system of linear equations boils down to just solving the matrix equation $Ax = b$, i.e. finding a vector x which satisfies the condition $Ax = b$. Thus, solving a system of equations (no matter how many of them) gets reduced to computing the inverse of a matrix and multiplying it by a vector. More importantly, since matrix operations can be parallelised, large systems can be solved in a matter of seconds.

Inverse Matrix

Solving a system of linear equations $Ax = b$ is equivalent to finding the unique vector x that lands on the vector b after the transformation A, i.e. $x = A^{-1}b$. A^{-1} is known as the inverse of matrix A. In most cases, A represents a transformation wherein the span is maintained (i.e 2D stays 2D, 3D stays 3D, etc.). But in the rare cases when A happens to squish the space into a lower dimensional one, such as squishing a 3D space onto a straight line, it becomes quite unlikely that a vector x exists which lands on b. The problem is that the vector Ax lies on that 2D plane, but the vector b does not lie on that plane. Hence, this problem becomes unsolvable in an exact sense. In such cases one can say that the system of equations does not have a solution. Such situations are reflected by the fact that the determinant of A is zero, and equivalently the inverse A^{-1} does not exist. Non-invertible matrices are also called singular matrices.

Rank of a Matrix

The rank of a matrix is the dimensionality of the output of the transformation. For example consider the matrix,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}$$

It has a rank of one because the two column vectors are collinear (i.e. they are the same vectors), and so this matrix squishes the 2D space onto a straight line of dimension one.

Column Space

The column space of a matrix is the span of the columns of the matrix. For example consider the matrix,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}$$

The column space of the matrix is a straight line. Thus, rank is equal to the number of dimensions in the column space of the matrix.

Null Space

The null space is the space of all vectors that land on the zero vector under the transformation.

Least Squares Approximation

For a system of equations that do not have a solution, one tries to find an approximate solution to such equations, i.e to find the vector x which will come as close to b as possible after the transformation A. One such technique being used to do that is called the least squares approximation. The least squares approximate solution to the system of equations $Ax = b$ is given by,

$$x = (A^T A)^{-1} A^T b$$

This is an important and very frequently used technique as in most of the real-world phenomena, the matrix A is not invertible. This is because many real-world systems are usually not represented by square matrices.

1.3.3. EIGENVALUES AND EIGENVECTORS

Eigenvectors of a matrix A are special vectors which do not change their direction (or span) under the transformation A. They just get scaled in the same direction, and the magnitude by which they get scaled is called the eigenvalue of that eigenvector. If the vector reverses its direction, the eigenvalue is negative. Mathematically, this can be represented by the equation,

$$Av = \lambda v$$

Here, v is the eigenvector of the matrix A and λ is the eigenvalue.

Eigenvalues And Eigenvectors

The eigenvector-eigenvalue equation $Av = \lambda v$ can be rewritten as,

$$Av - \lambda v = 0 \quad or \quad (A - \lambda I)\, v = 0$$

Now, the quantity $(A - \lambda I)$ itself is a matrix, and one knows that the a matrix-vector multiplication is zero only when the matrix (transformation) squishes the space into a lower dimensional one. This is represented by the fact that the determinant of the matrix is zero. Thus, solving the equation $det(A - \lambda I) = 0$ will give the eigenvectors.

Consider the following example for calculating the eigenvalues and eigenvectors. Let's define the matrix A as,

$$A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$$

Solving the equation $det(A - \lambda I) = 0$ gives,

$$det\left(\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

$$or,\ det\left(\begin{bmatrix} 2-\lambda & 3 \\ 2 & 1-\lambda \end{bmatrix}\right) = 0$$

$$or,\ (2-\lambda)(1-\lambda) - (3 \times 6) = 0$$

$$or,\ \lambda^2 - 3\lambda - 4 = 0$$

Solving the quadratic equation gives,

$$\lambda_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} = \frac{3-5}{2} = -1$$

$$\lambda_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} = \frac{3+5}{2} = 4$$

Thus, the two eigenvalues λ_1 and λ_2 have been determined. A square matrix of size $n \times n$ always has exactly n eigenvalues, each with a corresponding eigenvector. The eigenvalue specifies the size of the eigenvector. Now using the equation $Av = \lambda v$ for the two eigenvalues λ_1 and λ_2 one can find the corresponding eigenvectors.

$$For\ \lambda_1 = -1,$$

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = -1 \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}$$

$$or,\ 2x_{11} + 3x_{12} = -x_{11}$$

$$2x_{11} + 1x_{12} = -x_{12}$$

$$or,\ x_{11} = -x_{12}$$

$$For\ \lambda_1 = 4,$$

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = 4 \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}$$

$$or,\ 2x_{11} + 3x_{12} = 4x_{11}$$

$$2x_{11} + 1x_{12} = 4x_{12}$$

$$or,\ x_{11} = \frac{3}{2} x_{12}$$

Since an eigenvector simply represents an orientation (the corresponding eigenvalue represents the magnitude), all scalar multiples of the eigenvector are vectors that are parallel to this eigenvector, and are therefore equivalent (i.e. upon normalizing the vectors, they would all be equal). Thus, instead of further solving the above system of equations, one can freely choose a real value for either x_{11} or x_{12}, and determine the other one by using the final equations. For example, if one chooses arbitrarily $x_{12} = 1$ for $\lambda = -1$ then $x_{11} = -1$ and $x_{12} = 1$ for $\lambda = 4$ then $x_{11} = 3/2$ giving the corresponding eigenvectors as,

$$v_{\lambda_1} = v_{-1} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad and \quad v_{\lambda_2} = v_4 = \begin{bmatrix} 3/2 \\ 1 \end{bmatrix}$$

The eigenvectors are usually reported in the normalised form even though there can be an infinite number of vectors, conventionally one would report the above eigenvectors. But it is not incorrect to say that the following vectors are also the eigenvectors of this matrix too.

$$v_{\lambda_1} = v_{-1} = \begin{bmatrix} -5 \\ 5 \end{bmatrix} \quad and \quad v_{\lambda_2} = v_4 = \begin{bmatrix} 30 \\ 20 \end{bmatrix}$$

Eigendecomposition of a Matrix

Similar to the prime factorisation of numbers (i.e. breaking down an integer as a product of its prime factors, such as $12 = 2 \times 2 \times 3$), one can write a matrix as a product of other matrices. These other matrices are matrices formed by the eigenvectors and eigenvalues of the original matrix (though there's no analogy between eigenvalues and prime numbers). One can decompose any matrix diagonalisable A as,

$$A = Q\Sigma Q^{-1}$$

Here, Q is a matrix whose each column is an eigenvector of A and Σ is a diagonal matrix whose diagonal entries are the eigenvalues of A. This is called eigendecomposition of the matrix A. Most of the square matrices are diagonalisable, with some exceptions. A matrix is diagonalisable if there exists some invertible matrix P such that $A = P^{-1}AP$.

Consider a (2×2) matrix A with eigenvectors v_1 and v_2 and the corresponding eigenvalues λ_1 and λ_2. Then,

$$Av_1 = \lambda_1 v_1 \quad and \quad Av_2 = \lambda_2 v_2$$

Using the matrix trick the eigenvectors v_1 and v_2 can be arranged as column vectors $Q = [v_1\ v_2]$. Also, another diagonal matrix Σ can be created with the eigenvalues as the diagonal entries. Then the above equations can be written compactly in one matrix equation as,

$$A\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$or,\ AQ = Q\Sigma$$

$$or,\ A = Q\Sigma Q^{-1}$$

Consider the following example.

$$For,\ A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} with\ eigenvalues\ -1\ and\ 4,\ eigenvectors \begin{bmatrix} -1 \\ 1 \end{bmatrix} and \begin{bmatrix} 3/2 \\ 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} -1 & 3/2 \\ 1 & 1 \end{bmatrix}\ \Sigma = \begin{bmatrix} -1 & 0 \\ 0 & 4 \end{bmatrix} and\ Q^{-1} = \begin{bmatrix} -2/5 & 3/5 \\ 2/5 & 2/5 \end{bmatrix}$$

$$A \text{ can be represented as } A = Q\Sigma Q^{-1}$$

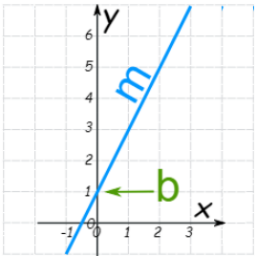
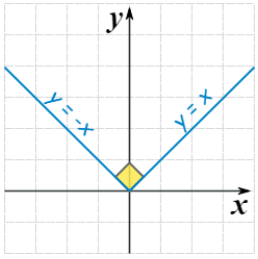
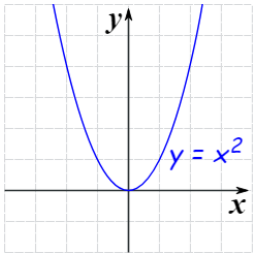
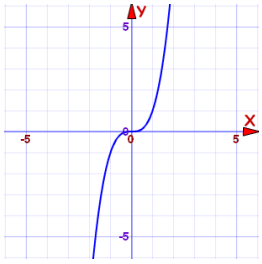
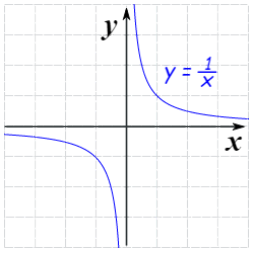
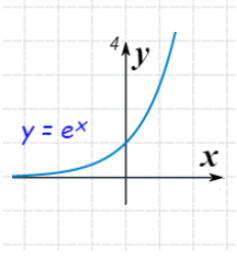
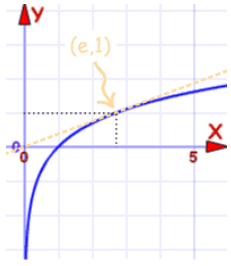
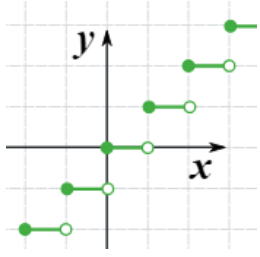
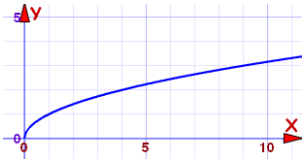
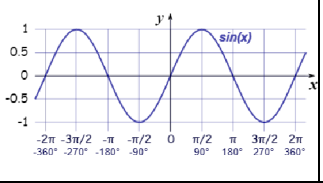
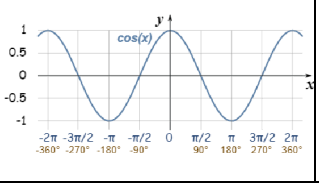
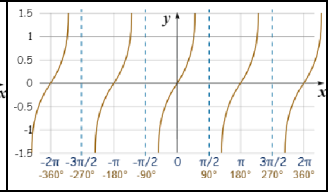
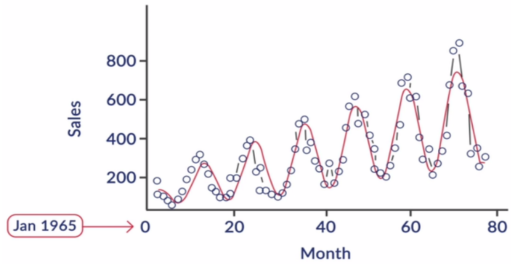
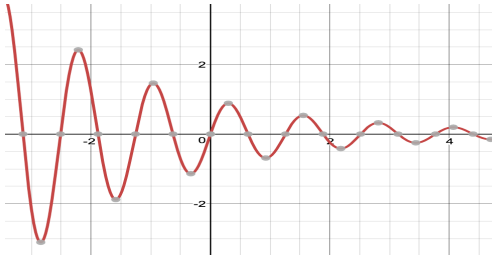
$$or,\ A = \begin{bmatrix} -1 & 3/2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} -2/5 & 3/5 \\ 2/5 & 2/5 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$$

1.3.4. MULTIVARIABLE CALCULUS

Calculus is one of the most important branches of mathematics having extensive applications in data sciences and machine learning. At its core, calculus is the mathematical study of continuous change. In the context of machine learning, calculus is heavily used in optimisation techniques which are the workhorses of most ML models.

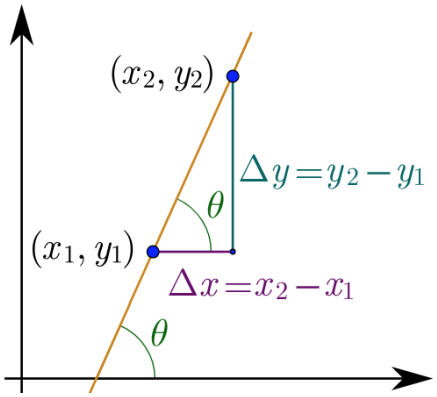
Functions

A function is a relationship, or a mapping, between inputs and outputs. To quickly reiterate, if a function f takes the input x and returns the output y, then this relation is denoted as $y = f(x)$. The element x is called the argument of the function and y is the value of the function. Some of the common functions have been given in the following table.

$y = mx + b$	$y = \|x\|$	$y = x^2$	$y = x^3$
			
$y = 1/x$	$y = e^x$	$y = ln\ x$	$y = floor(x)$
			
$y = \sqrt{x}$	$y = sin\ (x)$	$y = cos\ (x)$	$y = tan\ (x)$
			
$A\sqrt{x}\ cos(Bx) + Cx + D$		$e^{-x/A}\ sin\ (x)$	
			

Derivatives

The study of functions naturally led the way to the study of the rate of change of functions. In fact, the scientists who had invented calculus had spent a majority of their time trying to model the rates of change of functions. The basic quantity being used to measure the rate of change of a function is the slope. The slope is simply the ratio of the change in the value of the function (rise) to the change in the input (run). The following figure shows the slope of a linear function $y = mx + c$.



$$slope = \frac{\Delta y}{\Delta x}$$

$$or,\ m = \frac{vertical\ change}{horizontal\ change}$$

$$or,\ m = \frac{rise}{run}$$

Differentiation

The slope of a function at a point is basically the derivative of the function computed at that point. Differentiation is the action of computing a derivative. The derivative of a function $y = f(x)$ is a measure of the rate at which the value y of the function changes with respect to the change in the variable x. It is called the derivative of f with respect to x.

Although in practice one usually computes derivatives using the rules, it is important to understand the derivatives by first principles. This method is the foundation for the rest of differential calculus. Every differentiation rule and identity in calculus is based on the concept of derivatives by first principles. The rule is as follows,

$$m = \frac{change\ in\ y}{change\ in\ x} = \frac{\Delta y}{\Delta x}$$

$$or,\ m = \frac{\Delta f(a)}{\Delta a} = \frac{f(a+h) - f(a)}{(a+h) - a} = \frac{f(a+h) - f(a)}{h}$$

$$or,\ f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$$

Some of the common differentiation rules are given in the following table.

Sl.	Function	Derivative	Sl	Function	Derivative
1	cf	cf'	2	x^2	$n\,x^{n-1}$
3	$f+g$	$f'+g'$	4	$f-g$	$f'-g'$
5	fg	$fg'+f'g$	6	f/g	$(f'g-g'f)/g^2$
7	$1/f$	$-f'/f^2$	8	$f(g(x))$	$f'g(x).g'(x)$

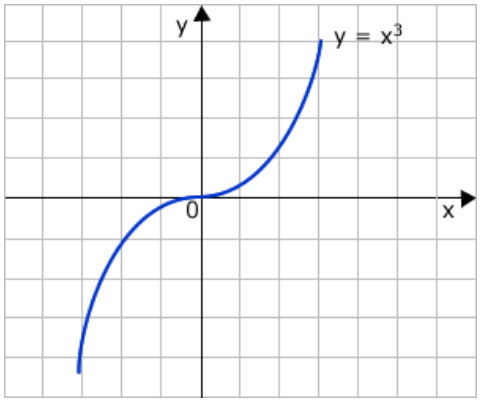
Differentiation of some of the common functions are given in the following table.

Sl.	Function	Derivative	Sl	Function	Derivative
1	c	0	2	x	1
3	ax	a	4	x^2	$2x$
5	x^3	$3x^2$	6	$\sqrt{x}$	$\frac{1}{2}\, x^{-1/2}$
7	e^x	e^x	8	a^x	$ln(a)\, a^x$
9	$ln\, x$	$1/x$	10	$log_a\, x$	$1/(x\, ln(a))$
11	$sin\, x$	$cos\, x$	12	$cos\, x$	$-\, sin\, x$
13	$tan\, x$	$sec^2\, x$	14	$sin^{-1}\, x$	$1/\sqrt{1-x^2}$
15	$cos^{-1}\, x$	$-\, 1/\sqrt{1-x^2}$	16	$tan^{-1}\, h$	$1/\sqrt{1+x^2}$

Critical Points, Maxima and Minima

An important use of derivatives is that they help in identifying the critical points of functions, i.e. points at which something interesting happens, such as the function reaches a maximum or a minimum value, changes its curvature, etc. Finding the maxima or minima of functions is a very general and frequently occurring problem in calculus. In machine learning, almost all model training algorithms boil down to finding the minimum of a function (the cost function).

Critical Points : Critical points are points where the function reaches a local or global maximum or minimum value (or sometimes an inflexion point). A function $f(x)$ has a critical point x if $f'(x) = 0$ at that point, or if the function is non-differentiable. It is not necessary that if $f'(x) = 0$ then the point needs to be either a maximum or a minimum, such points are called inflexion points. The following figure explains the same.

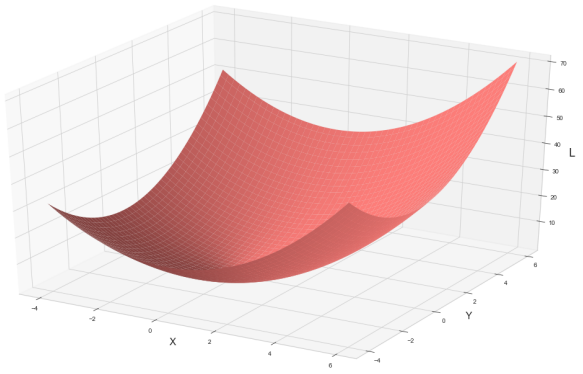
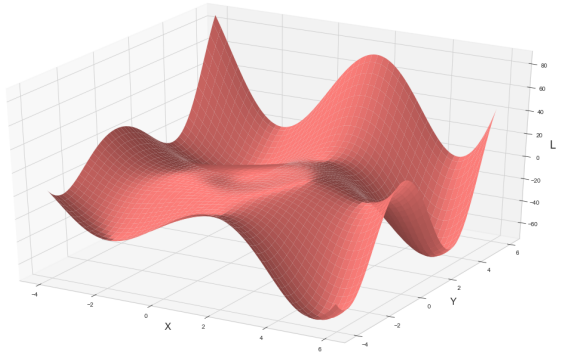


For example, the plot shows the function $y = x^3$, which has an inflexion point at $x = 0$. The slope at $x = 0$ is 0 ($f'(x) = 3x^2 = 0$), though one can see that there is no minimum or maximum at $x = 0$.

Maxima and Minima : A differentiable function's maxima and minima points satisfy the condition $f'(x) = 0$. To find whether a point is maxima or minima, one can compute the derivatives in the vicinity of the point. If the derivative is positive to the right and negative to the left of the critical point, then it is a minima and vice versa. But if the derivative does not change the sign from right to the left of the critical point, then it is an inflexion point. There also is another way to deduce whether a critical point is maxima or minima by computing the double derivative. If $f''(x) > 0$, it is a minimum, if $f''(x) < 0$, it is a maximum and if $f''(x) = 0$ then it is an inflexion point.

Multivariable Functions

In many real-world phenomena, quantities of interest depend on more than one variable. For example, one would have seen the ideal gas equation $PV = nRT$ or $P(n, V, T) = nRT/V$. This is an example of a multivariable function. In machine learning, many of the optimisation problems boil down to minimising or maximising multivariable functions. Some examples of multivariable functions are given in the following table.

$f(x,y) = x^2 + y^2$	$f(x,y) = (x^2 + y^2)\ (sin\ x + cos\ y)$
	

Partial Derivatives

There is often the need to compute the rate of change of multivariate functions $f(x,y)$ with respect to the variables x and y. This can be done by using partial derivatives, i.e. derivatives of a function computed with respect to only one variable. For example, the partial derivative of $f(x,y) = x^2 + y^2$ with respect to x and y respectively are,

$$\frac{\partial f(x,y)}{\partial x} = \frac{\partial f\left(x^2+y^2\right)}{\partial x} = \frac{\partial\left(x^2\right)}{\partial x} + \frac{\partial\left(y^2\right)}{\partial x} = 2x$$

$$\frac{\partial f(x,y)}{\partial y} = \frac{\partial f\left(x^2+y^2\right)}{\partial y} = \frac{\partial\left(x^2\right)}{\partial y} + \frac{\partial\left(y^2\right)}{\partial y} = 2y$$

This represents the rate at which the value of the function changes with the change in value of x or y, while the value of y or x is kept constant (while computing a partial derivative, all other variables of the function are kept constant).

Total Derivatives

In an function $f(x,y)$, the variables x and y are usually assumed to be independent. However, in some situations, they may be dependent on some other common variables. For example, both x and y themselves may be varying with time t, i.e. $x = x(t)$ and $y = y(t)$. In such cases, one cannot assume that x and y are independent and thus, one cannot compute the partial derivatives assuming so. Thus, comes into play total derivatives. The total derivatives are somewhat analogous to the rate of change of a function with respect to all its variables. For example, the total derivative of $f(x,y) = x^2 + y^2$ with respect to t is,

$$\frac{\partial f(x,y)}{\partial t} = \frac{\partial f\left(x^2+y^2\right)}{\partial t} = \frac{\partial\left(x^2+y^2\right)}{\partial x} \cdot \frac{\partial x}{\partial t} + \frac{\partial\left(x^2+y^2\right)}{\partial y} \cdot \frac{\partial y}{\partial t} = 2x\,\frac{\partial x}{\partial t} + 2y\,\frac{\partial y}{\partial t}$$

Vector-Valued Functions

The functions being used till now have a single scalar output. For example, the function $f(x) = x^2$ has a single number as the output. Similarly, the multivariate function $f(x, y) = x^2 + y^2$ also outputs a single number. But, in many cases, it is convenient to have functions having more than one scalars as the output. These are called vector-valued functions. For example, the following function $F(t)$ has a vector of size two as the output (the two components $S(t)$ and $v(t)$ representing the distance covered by a car and its speed at time t respectively),

$$F(t) = \begin{bmatrix} S(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} 10 + 3t \\ \frac{2 + 5t}{t} \end{bmatrix}$$

Although distance and speed have different units, this is valid since the elements of a vector-values function are pretty much independent of each other. One can also think about derivatives of vector functions. The derivative $F'(t)$ will also be a vector of size two. Its first component will represent the rate of change of the distance $S(t)$ with time (i.e. the speed), while its second component will represent the rate of change of the speed $v(t)$ with time (i.e. its acceleration). Thus,

$$F'(t) = \begin{bmatrix} S'(t) \\ v'(t) \end{bmatrix} = \begin{bmatrix} 3 \\ -\frac{2}{t^2} \end{bmatrix}$$

Vector functions can also be multivariable,

$$F(x, y) = \begin{bmatrix} x + y \\ x^2 + y^2 \end{bmatrix}$$

So, the vector functions are almost a trivial extension of the usual univariate and multivariable functions, with the main advantage being that they make the notation compact, i.e. they can store multiple output variables (such as distance and speed) in one single vector, rather than maintaining many variables.

Jacobian

The derivatives of a multivariate function can be stored in a vector called the Jacobian. For example,

$$J_f(x, y) = J_f\left(x^2 + y^2\right) = \begin{bmatrix} \frac{\partial f}{\partial x}, & \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x, & 2y \end{bmatrix}$$

Geometrically, the Jacobian tells the slope of the function in all directions. The first component tells the rate of change of f with respect to x (or the slope in the x direction) while the second component tells the rate of change of f with respect to y (or the slope in the y direction). For example, the Jacobian of the above function at the point $(1, 0)$ is $J_f(1, 0) = [x^2 + y^2]_{(1,0)} = [2, 0]$. Thus, at the point $(1, 0)$, $\partial f/\partial x = 2$ and $\partial f/\partial y = 0$, which means that the function's slope is 2 in the x direction and 0 in the y direction. In other words, if one is standing at the point $(1, 0)$ and takes a small step in the positive x-direction, one will move a little uphill (since the slope is positive). On the

other hand, if one takes a small step in the y-direction, ones altitude won't change at all since the slope in y-direction is 0.

Jacobian Matrix

In the most general case, the Jacobian can be extended to vector-valued functions. In such cases, the Jacobian is a matrix. For example, consider the following vector-valued function,

$$F(x,y) = \begin{bmatrix} f_1(x,y) \\ f_2(x,y) \end{bmatrix} = \begin{bmatrix} x+y \\ x^2+y^2 \end{bmatrix}$$

This is most general case of a function, it has multiple outputs (f_1, f_2) (i.e. vector-valued) and multiple inputs (x,y) (i.e. multivariable). The derivative of this function means to compute the rate of change of all combinations, i.e. f_1 with respect to x, f_1 with respect to y, f_2 with respect to x and f_2 with respect to y. This can be stored in a Jacobian matrix as follows,

$$J_f(x,y) = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2x & 2y \end{bmatrix}$$

In general, for a vector-valued function F having n variables $x_1, x_2 \ldots, x_n$ and m components $f_1, f_2 \ldots, f_m$, the Jacobian is a $n \times m$ matrix given by,

$$J = \begin{bmatrix} \frac{\partial F}{\partial x_1} & \cdots & \frac{\partial F}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Hessian Matrix

The Hessian matrix can be thought of as a simple extension of the Jacobian. Just like the Jacobian contains the first order partial derivatives of a function, the Hessian matrix contains the second order partial derivatives. The Hessian matrix is heavily used in optimisation algorithms of multivariate functions, i.e. to find the maxima or minima of functions that depend on multiple variables.

Geometrically, the Hessian is a measure of the curvature of a function. The Hessian will be large at the curvy regions and smaller at the flat ones. The curvature (what the Hessian measures) is not the same as slope (what the Jacobian measures). The slope tells of the straight line that best represents the shape of the function at a point, whereas the curvature tells the inverse of the radius of the circle which best hugs the shape of the function at that point. The Hessian matrix H of a multivariable function f is a square $n \times n$ matrix, usually defined and arranged as follows,

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

Taylor Series and Linearisation

By Taylor expansion, any continuous function can be approximated by the linear combination of its first order gradient and the quadratic function of the second order gradient and so on. A Taylor series is a series expansion of a function about a point, Suppose that the function $f(x)$ is infinitely differentiable (smooth) at $x = a$. The Taylor series for $f(x)$ centred at $x = a$ is then given by,

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$$

or

$$f(x+h) = f(x) + \frac{f'(x)}{1!}(h) + \frac{f''(x)}{2!}(h)^2 + \frac{f'''(x)}{3!}(h)^3 + \cdots$$

2. STATISTICS

INFERENTIAL STATISTICS

2.1.1. BASICS OF PROBABILITY

Many times, one requires a very large amount of data for analysis which may need too much time and resources to acquire. In such situations, one is forced to work with a smaller sample of the data, instead of having the entire data to work with. The process of inferring insights from sample data is called Inferential Statistics.

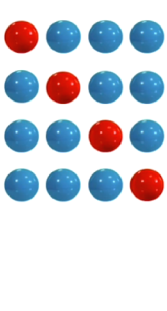
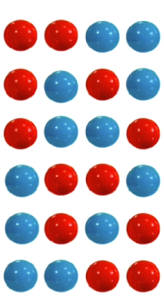
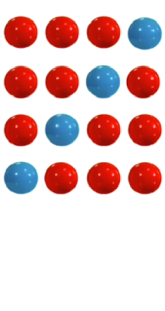
Note that even after using inferential statistics, one would only be able to estimate the population data from the sample data, but not find the exact values. This is because when one doesn't have the exact data, one can only make reasonable estimates about it with a limited level of certainty. Therefore, when certainty is limited, one talks in terms of probability.

Let's consider the following game. There is a bag with three red and two blue balls. Everyone playing the game gets four chances to pick a random ball from the bag. Everytime a ball is picked, its color is noted and then the ball is placed back in the bag for the next pick till all the four chances are complete. Anyone who gets a combination of four red balls for all the four picks will receive a reward of 150 points and for any other combination will get a penalty of 10 points. The question out here is whether in the long run (i.e. if the game is played a lot of times), is a player going to end up with positive or negative points. For finding the answer to this question, one has to find the following details. Each of these steps explains a concept which is very useful for finding the answer.

1. Finding all the possible combinations of picks (Random Variables).
2. Finding the probability of each combination (Probability Distributions).
3. Using the probabilities to estimate the points earned per player (Expected Value).

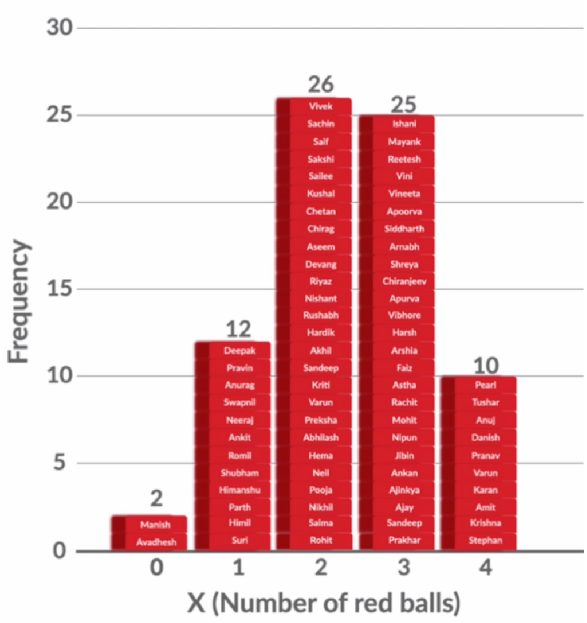
Random Variables

The following figure shows all the possible outcomes/combinations that can occur for the four picks. In all there are 16 different possible outcomes. These outcomes can be quantified by using some variable X representing the number of red balls picked or number of blue balls picked or the difference in the number of red and blue balls picked. This random variable X basically converts the outcomes of experiments to something measurable, converting the data entirely into quantitative terms, making it possible to perform a number of statistical analyses on the data.

Possible Outcomes					
Possible Combinations of picks	**0 Red Balls 4 Blue Balls**	**1 Red Balls 3 Blue Balls**	**2 Red Balls 2 Blue Balls**	**3 Red Balls 1 Blue Balls**	**4 Red Balls 0 Blue Balls**
					
Random Variables					
No. of red balls	$X = 0$	$X = 1$	$X = 2$	$X = 3$	$X = 4$
No. of blue balls	$X = 4$	$X = 3$	$X = 2$	$X = 1$	$X = 0$
Difference in red and blue balls	$X = -4$	$X = -2$	$X = 0$	$X = 2$	$X = 4$

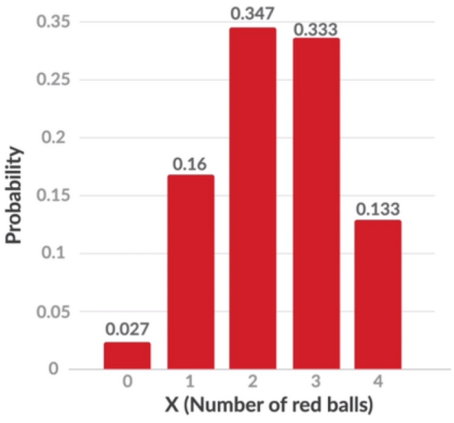
Probability Distributions

The random variable X, which helped in converting the outcomes of the experiment to something measurable has been defined as the number of red balls picked (the count of red balls being more relevant for the current experiment). Moving on to the next step one needs to find the probability of each of these combinations. For this a sample set of 75 people were allowed to pick the balls and the result of the picks are given in the following figure. It gives the frequency of the number of red balls picked for the sample set and the probability distribution.



$$Probability\ P(X) = \frac{Favourable\ Outcomes}{Total\ Outcomes}$$

X	$P(X) = x/75$
0	2/75 = 0.027
1	12/75 = 0.160
2	26/75 = 0.347
3	25/75 = 0.333
4	10/75 = 0.133



So basically, a probability distribution is ANY form of representation that tells about the probability for all possible values of X. It could be an equation or table or chart. In a valid, complete probability distribution, there are no negative values, and the total of all probability values adds up to 1.

Expected Value

The expected value for a variable X is the value of X one would expect to get after performing the experiment once. It should be interpreted as the average value one gets after the experiment has been conducted an infinite number of times. It is also called the expectation or average or mean value. Mathematically, for a random variable X that can take values $x_1, x_2, x_3 \ldots, x_n$, the expected value (EV) is given by,

$$EV(X) = x_1 \cdot P\left(X = x_1\right) + x_2 \cdot P\left(X = x_2\right) + \ldots + x_n \cdot P\left(X = x_n\right)$$

Now using the probabilities calculated and above equation one can find the expected value for the number of red balls.

$$EV(No\ of\ red\ balls) = (0 \times 0.027) + (1 \times 0.160) + (2 \times 0.347) + (3 \times 0.333) + (4 \times 0.133)$$

The expected value for the number of red balls is 2.385, which means that if the experiment is conducted infinite times, the average number of red balls getting picked per game would be 2.385. But this expected value is not going to be of any help in estimating the points earned per player. For this, one needs to consider a more relevant random variable X such as points earned after each game. Thus,

$$X = \textit{Points earned in each game}$$
$$P(X = +150) = P(X = 4) = 0.133$$
$$P(X = -10) = P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3) = 0.867$$
$$EV(\textit{Points earned per game}) = (150 \times 0.133) + (-10 \times 0.867) = 11.28$$

So, on an average each player is going to earn 11.28 points in every game. Thus, there is a very high chance of a player going to end up with positive points.

This same principle can be used in finding whether a game in a casino is going to be profitable for the house or the player in the long run. If this same game is played in a casino where the points are replaced with currency then in the long run the house is going to lose money. In order for the house to always win, the expected value needs to be negative. To achieve this the house can either bring down the value of rewards on winning or increase the value of penalty on losing the game or may change the rules of the game. Consider another example, the game of roulette. The European roulette wheel contains the numbers 0 to 36 written in an irregular sequence. The players can bet on any number starting from 0 to 36. For example, let's say one bets ₹1000 on the number 5. Upon spinning the wheel if the ball lands on the pocket marked 5, one would win ₹$1000 \times$ ₹$36 =$ ₹36000, resulting in net winnings of ₹$36000 -$ ₹$1000 =$ ₹35000. However, if the ball lands on any other pocket, one would not win anything, resulting in net winnings of ₹$0 -$ ₹$1000 =$ $-$₹1000. The probability of winning the game if one bets on the number 5 is $1/37 = 0.027$. The expected value of X where it is the random variable for net winnings is $(35000 \times P(X = 35000)) + (-1000 \times P(X = -1000)) = -2.70$. Thus, the game of roulette is designed to ensure a negative expected value, helping the house to always win in the long run.

2.1.2. DISCRETE PROBABILITY DISTRIBUTIONS

The different probabilities of picking the balls while playing the game were found out by conducting experiments. Specifically, out here, 75 people were asked to play the game and based on the data of these people, a frequency distribution was created. Then, using this distribution the probability distribution was built. However, this is a lengthy process, but there are shorter processes for finding the probabilities where one doesn't need to repeat the experiments.

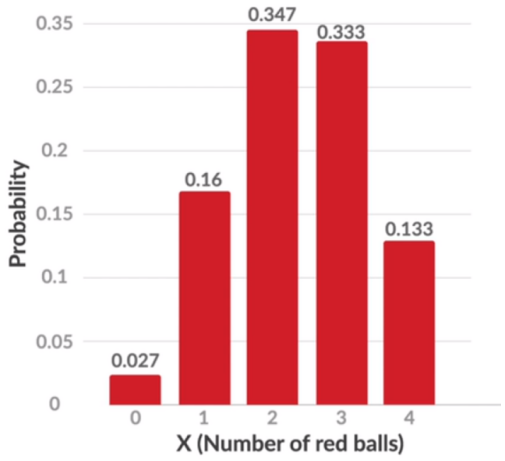
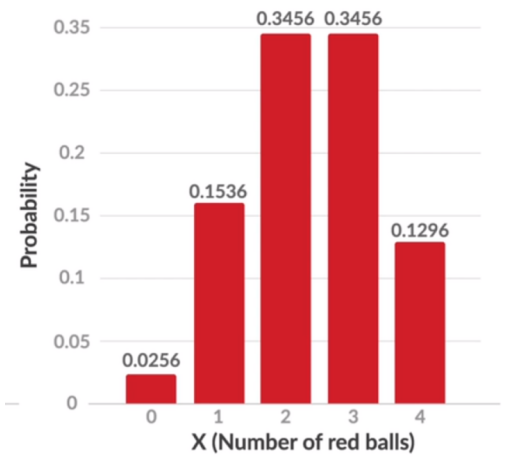
Probability Without Experiment

One can find the probability distribution of picking red balls for the same game without conducting any experiment by using the basic concepts of probability such as addition rule of probability for mutually exclusive events, multiplication rule of probability for independent events, combinatorics etc. The following table gives the theoretical probability values for the experiments performed.

Experiment	Theoretical Probability
Probability of getting 0 red and 4 blue balls $P(X = 0)$	$1 \times [(2/5) \times (2/5) \times (2/5) \times (2/5)] = 0.0256$
Probability of getting 1 red and 3 blue balls $P(X = 1)$	$4 \times [(3/5) \times (2/5) \times (2/5) \times (2/5)] = 0.1536$
Probability of getting 2 red and 2 blue balls $P(X = 2)$	$6 \times [(3/5) \times (3/5) \times (2/5) \times (2/5)] = 0.3456$

Probability of getting 3 red and 1 blue balls $P(X=3)$	$4 \times [(3/5) \times (3/5) \times (3/5) \times (2/5)] = 0.3456$
Probability of getting 4 red and 0 blue balls $P(X=3)$	$1 \times [(3/5) \times (3/5) \times (3/5) \times (3/5)] = 0.1296$

As can be seen in the following figure the theoretical (calculated) values of probability are actually quite close to the experimental value. The small differences that can be noticed exist because of the lower number of experiments done.

Observed Probability Distribution	Theoretical Probability Distribution
	

Binomial Distribution

The binomial distribution gives the discrete probability distribution $P(X=r)$ of obtaining exactly r successes out of n Bernoulli trials (where the result of each Bernoulli trial is true with probability p and false with probability $1-p$). The binomial distribution is given by,

$$P(X=r) = {}^{n}C_{r}\,(p)^{r}\,(1-p)^{n-r}$$

Inorder to be able to apply the binomial formula the following conditions needs to be satisfied,

1. The total number of trials should be fixed at n.
2. The n trials are independent.
3. Each trial is binary, i.e., has only two possible outcomes, success or failure.
4. Probability of success is same in all trials, denoted by p.
5. The random variable X is the number of successes in the n trials.

Negative Binomial Distribution

Similarly if X denotes the number of trials until the r^{th} success, then the probability distribution is given by,

$$P(X=r) = {}^{n-1}C_{r-1}\,(p)^{r}\,(1-p)^{n-r}$$

Geometric Distribution

A geometric distribution is a special case of a negative binomial distribution with $r=1$. Let X denote the number of trials until the first success, then the probability distribution is given by,

$$P(X=r) = p(1-p)^{n-1}$$

Poisson Distribution

Let the discrete random variable X denote the number of times an event occurs in an interval of time (or space). Then X may be a Poisson random variable with $r = 0, 1, 2...$, $\lambda > 0$ (λ being both the mean and the variance of X) and the probability distribution is given by,

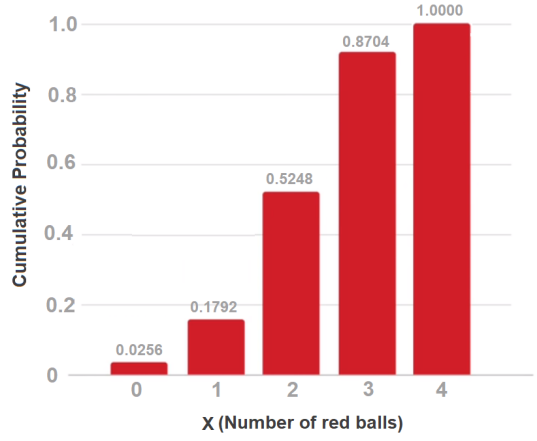
$$P(X=r) = \frac{e^{-\lambda}\lambda^{r}}{r!}$$

Cumulative Probability

Till now only the probability of getting an exact value (for example, $P(X=4)$) has been discussed. But, the casino may need to know the probability of getting three or less red balls (i.e $P(X \leq 3)$), as this is where the players will lose and the house will make money. Sometimes, talking in less than is more useful, and cumulative probability is used for such cases. The cumulative probability of X is defined as the probability of the variable being less than or equal to x. It is denoted by,

$$F(x) = P(X \leq x) = \sum_{r=0}^{n} P(X=r)$$

The following figure gives the cumulative probability of the ball picking game.



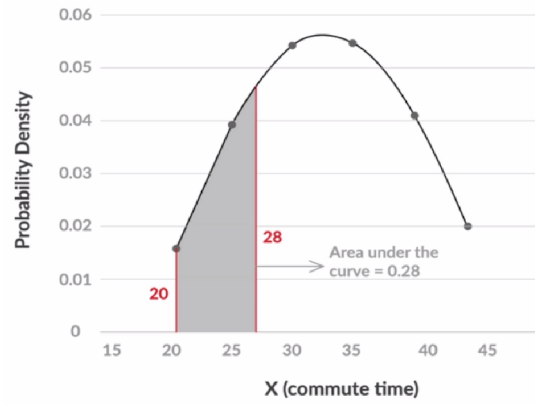
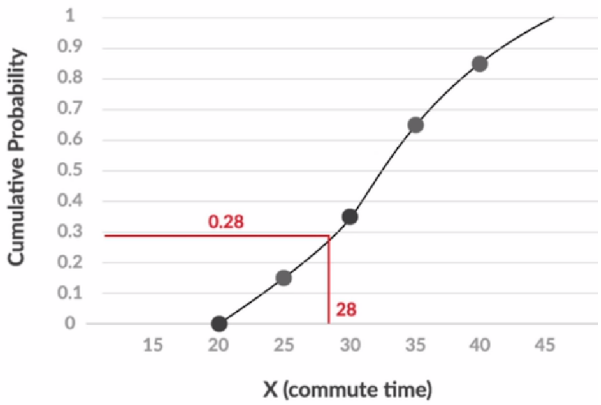
x	$F(x) = P(X \leq x)$
0	0.0000 + 0.0256 = 0.0256
1	0.0256 + 0.1536 = 0.1792
2	0.1792 + 0.3456 = 0.5248
3	0.5248 + 0.3456 = 0.8704
4	0.8704 + 0.1296 = 1.0000

2.1.3. CONTINUOUS PROBABILITY DISTRIBUTIONS

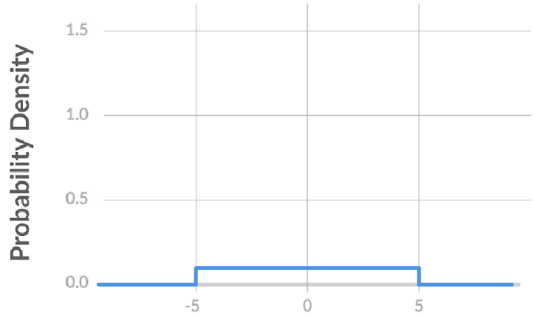
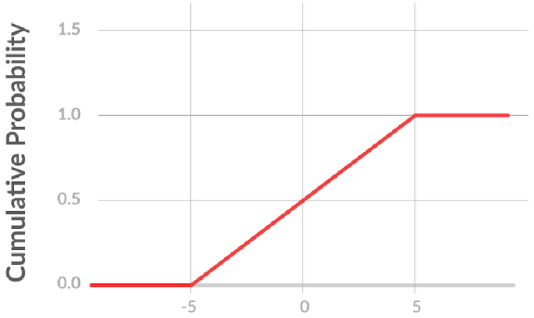
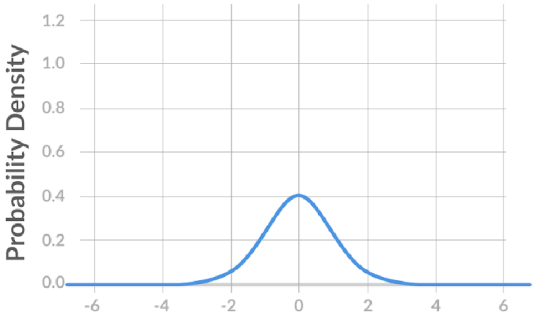
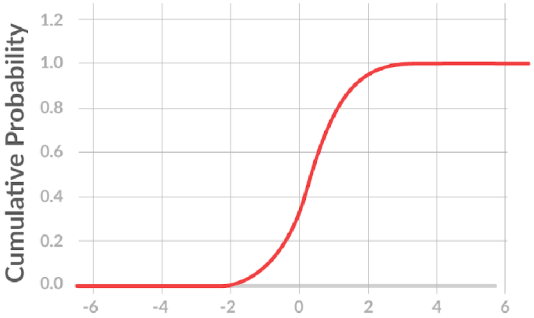
Till now one had been dealing with only discrete random variables (such as number of balls, number of patients, cars, wickets, pasta packets, etc.). Now, one is going to learn about the probability of continuous random variables (such as time, weight etc.).

Probability Density Functions

The CDF and PDF are two functions that talk about probabilities in terms of intervals rather than exact values. The Cumulative Distribution Function is a distribution which plots the cumulative probability of X against X. However, the Probability Density Function is a function in which the area under the curve gives the cumulative probability. The following figure shows a CDF and PDF.

Probability Density Function	Cumulative Distribution Function
	

The main difference between the cumulative probability distribution of a continuous random variable and a discrete one, is the way they are plotted. While the continuous variables' cumulative distribution is a curve, the distribution for discrete variables looks more like a bar chart. The reason for plotting both of these so differently is that, for discrete variables, the cumulative probability does not change very frequently. In the discrete example, one only cares about what the probability is for $0, 1, 2, 3, \ldots$ (this is because the cumulative probability does not change between, say, 3 and 3.999999 and is equal to 0.8704). For continuous variables, PDFs are more commonly used in real life as it is much easier to see the patterns in PDFs as compared to CDFs. The following figure gives the CDF and PDF for a few random variables.

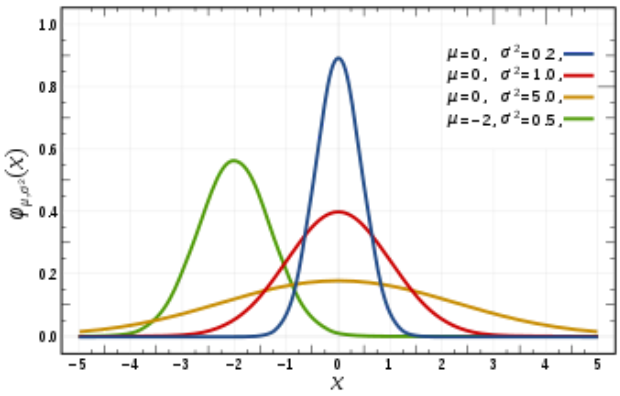
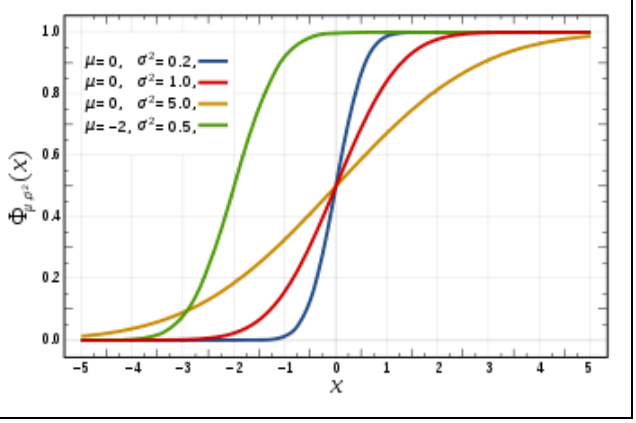
Uniformly Distributed Variable	
PDF	CDF
	
Symmetrically Distributed Variable	
PDF	CDF
	

Normal Distribution

In probability theory, a normal (Gaussian/Gauss/Laplace-Gauss/z) distribution is a type of continuous probability distribution for a real-valued random variable. A normal distribution is sometimes informally called a bell curve. Normal distributions are important in statistics and are often used in the natural and social sciences to represent real-valued random variables whose distributions are not known. Their importance is partly due to the central limit theorem. Moreover, Gaussian distributions have some unique properties that are valuable in analytic studies. For instance, any linear combination of a fixed collection of normal deviates is a normal deviate. Many results and methods (such as propagation of uncertainty and least squares parameter fitting) can be derived analytically in explicit form when the relevant variables are normally distributed. The general form of its probability density function is,

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

The parameter μ is the mean or expectation of the distribution (and also its median and mode) and σ is its standard deviation. The variance of the distribution is σ^2. A random variable with a Gaussian distribution is said to be normally distributed. The following figures give the PDF and CDF of some normally distributed functions.

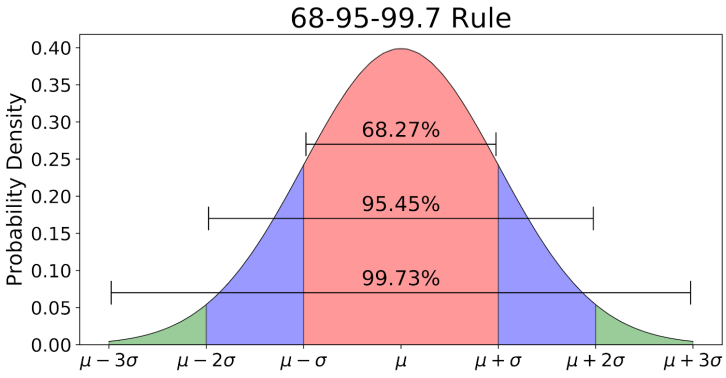
Probability Density Function	Cumulative Distribution Function
	

As can be seen, the value of σ is an indicator of how wide the graph is. This is true for any graph, not just the normal distribution. A low value of σ means that the graph is narrow, while a high value implies that the graph is wider. This happens because the wider graph will clearly have more values away from the mean, resulting in a high standard deviation.

Any data that is normally distributed follow the 1-2-3 rule. This rule states that,

1. There is a 68.27 % probability of the variable lying within 1 standard deviation of the mean.
2. There is a 95.45 % probability of the variable lying within 2 standard deviations of the mean.
3. There is a 99.73 % probability of the variable lying within 3 standard deviations of the mean.

The following figure gives an insight into the same,



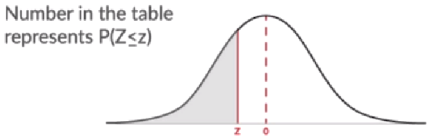
If one wants to find the probability, it doesn't matter what the value of μ and σ is. All one needs to know is how far the value of X is from μ, specifically, what multiple of σ is the difference between X and μ. Consider the following example, where the mean $(\mu) = 35$ and standard deviation $(\sigma) = 5$. Then the various probabilities for random variable X can be easily found out like,

1. $P(25 < X < 45) = P(\mu - 2\sigma < X < \mu + 2\sigma) \approx 47.5\% + 47.5 \approx 95\%$
2. $P(25 < X < 50) = P(\mu - 2\sigma < X < \mu + 3\sigma) \approx 47.5\% + 49.85\% \approx 97.35\%$
3. $P(X < 40) = P(0 < X < \mu + \sigma) \approx 50\% + 34\% \approx 84\%$

In the above examples one could see that for finding the probability of a random variable X, one is basically finding out how far is the random variable X form the mean μ. For example, the random variable $X = 43.5$ is 8.5 units away from the mean. But in standard terms it can be said as $1.65\,\sigma$ or $(8.5/5)$ standard deviations away from the mean. This value of 1.65 is called as the standardised random variable or $z - score$ which is given by,

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

Basically, it tells how many standard deviations away from the mean μ, the random variable X is. The standardised random variable Z is a much more informative variable than the unstandardized random variable X while dealing with cumulative probabilities. A positive value of Z means that the value is to the right of the center implying high cumulative probability and vice versa. The cumulative probability corresponding to a given value of Z (say 0.68) can be found out using the Z-table as shown in the following table.



z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
−1.9	.0287	.0281	.0274	.0268	.0262	.0256	.0250	.0244	.0239	.0233
−1.8	.0359	.0351	.0344	.0336	.0329	.0322	.0314	.0307	.0301	.0294
−1.7	.0446	.0436	.0427	.0418	.0409	.0401	.0392	.0384	.0375	.0367
−1.6	.0548	.0537	.0526	.0516	.0505	.0495	.0485	.0475	.0465	.0455
−1.5	.0668	.0655	.0643	.0630	.0618	.0606	.0594	.0582	.0571	.0559
−1.4	.0808	.0793	.0778	.0764	.0749	.0735	.0721	.0708	.0694	.0681
−1.3	.0968	.0951	.0934	.0918	.0901	.0885	.0869	.0853	.0838	.0823
−1.2	.1151	.1131	.1112	.1093	.1075	.1056	.1038	.1020	.1003	.0985
−1.1	.1357	.1335	.1314	.1292	.1271	.1251	.1230	.1210	.1190	.1170
−1.0	.1587	.1562	.1539	.1515	.1492	.1469	.1446	.1423	.1401	.1379
−0.9	.1841	.1814	.1788	.1762	.1736	.1711	.1685	.1660	.1635	.1611
−0.8	.2119	.2090	.2061	.2033	.2005	.1977	.1949	.1922	.1894	.1867
−0.7	.2420	.2389	.2358	.2327	.2296	.2266	.2236	.2206	.2177	.2148
−0.6	.2743	.2709	.2676	.2643	.2611	.2578	.2546	.2514	.2483	.2451
−0.5	.3085	.3050	.3015	.2981	.2946	.2912	.2877	.2843	.2810	.2776
−0.4	.3446	.3409	.3372	.3336	.3300	.3264	.3228	.3192	.3156	.3121
−0.3	.3821	.3783	.3745	.3707	.3669	.3632	.3594	.3557	.3520	.3483
−0.2	.4207	.4168	.4129	.4090	.4052	.4013	.3974	.3936	.3897	.3859
−0.1	.4602	.4562	.4522	.4483	.4443	.4404	.4364	.4325	.4286	.4247
−0.0	.5000	.4960	.4920	.4880	.4840	.4801	.4761	.4721	.4681	.4641

z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	.5000	.5040	.5080	.5120	.5160	.5199	.5239	.5279	.5319	.5359
0.1	.5398	.5438	.5478	.5517	.5557	.5596	.5636	.5675	.5714	.5753
0.2	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
0.3	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
0.4	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
0.5	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
0.6	.7257	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7517	.7549
0.7	.7580	.7611	.7642	.7673	.7704	.7734	.7764	.7794	.7823	.7852
0.8	.7881	.7910	.7939	.7967	.7995	.8023	.8051	.8078	.8106	.8133
0.9	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621
1.1	.8643	.8665	.8686	.8708	.8729	.8749	.8770	.8790	.8810	.8830
1.2	.8849	.8869	.8888	.8907	.8925	.8944	.8962	.8980	.8997	.9015
1.3	.9032	.9049	.9066	.9082	.9099	.9115	.9131	.9147	.9162	.9177
1.4	.9192	.9207	.9222	.9236	.9251	.9265	.9279	.9292	.9306	.9319
1.5	.9332	.9345	.9357	.9370	.9382	.9394	.9406	.9418	.9429	.9441
1.6	.9452	.9463	.9474	.9484	.9495	.9505	.9515	.9525	.9535	.9545
1.7	.9554	.9564	.9573	.9582	.9591	.9599	.9608	.9616	.9625	.9633
1.8	.9641	.9649	.9656	.9664	.9671	.9678	.9686	.9693	.9699	.9706
1.9	.9713	.9719	.9726	.9732	.9738	.9744	.9750	.9756	.9761	.9767

Standard Normal Distribution

The simplest case of a normal distribution is known as the standard normal distribution. This is a special case when $\mu = 0$ and $\sigma = 1$, and is described by the probability density function,

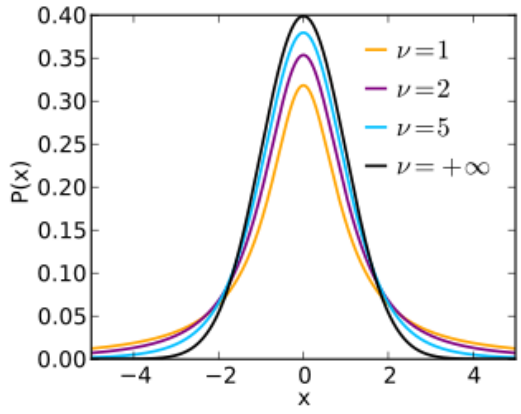
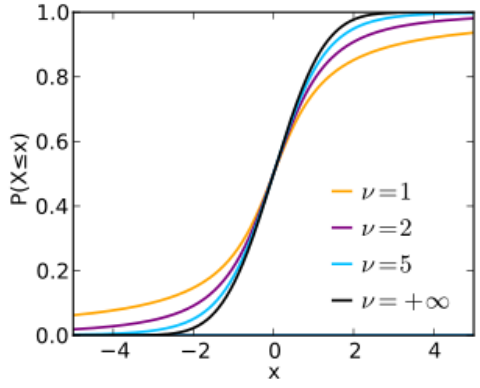
$$\varphi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$$

Student's T-Distribution

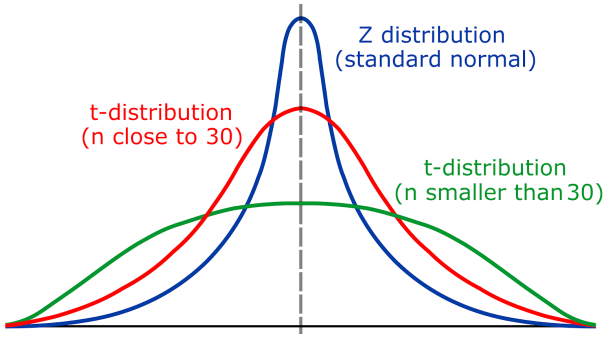
In real life scenarios most of the time one does not have all the information regarding a population (such as population standard deviation etc.). In such scenario the t-distribution is used. It is a continuous probability distribution that arises while estimating the mean of a normally distributed population in situations where the sample size is small and the population standard deviation is unknown. It is similar to the normal distribution in many cases (for example, it is symmetrical about its central tendency). However, it is shorter than the normal distribution and has a flatter tail, implying that it has a larger standard deviation. The general form of its probability density function is,

$$f(x, v) = \frac{\Gamma\left((v+1)/2\right)}{\sqrt{v\pi}\,\Gamma\left(v/2\right)} \left(1 + \frac{x^2}{v}\right)^{-\left((v+1)/2\right)}$$

The parameter $v = n - 1$ is the degrees of freedom where n is the number of observations. The following figures give the PDF and CDF of some *t*-distributed functions and comparison with the normal distribution.

Probability Density Function	Cumulative Distribution Function
	

***t*-distribution *vs* *z*-distribution**



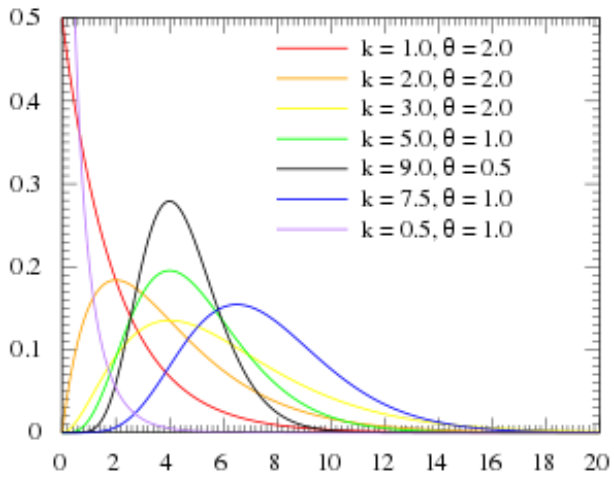
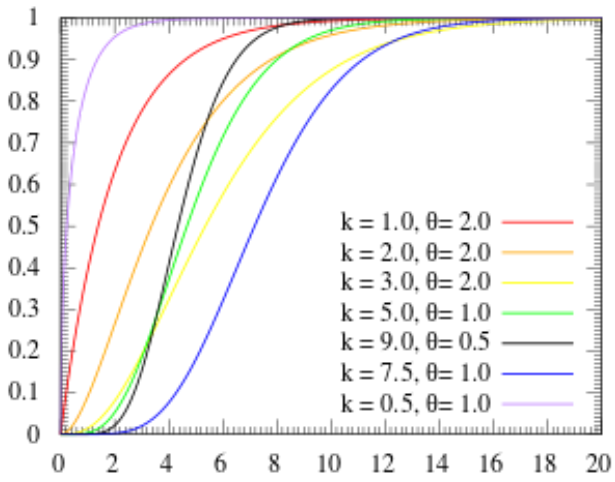
The most important use of the t-distribution is that one can approximate the value of the population standard deviation σ from the sample standard deviation S. However, as the sample size increases (i.e $n \geq 30$), the t-distribution tends to be similar to the z-distribution. Thus, the t-distribution is used whenever the standard deviation of the population is unknown and the sample size is less than 30. But, if the standard deviation of the population is known, one has to use z-distribution, irrespective of the value of the sample size.

Gamma Distribution

The gamma distribution is a two-parameter family of continuous probability distributions. The exponential distribution and chi-squared distribution are special cases of the gamma distribution. It is frequently used to model waiting times. The continuous random variable X follows a gamma distribution for x (the waiting time) until the k^{th} event occurs, if its probability density function is,

$$f(x,k) = \frac{1}{\Gamma(k)\,\theta^k}\, x^{k-1}\, e^{-\frac{x}{\theta}}$$

The following figures give the PDF and CDF of some gamma distributed functions.

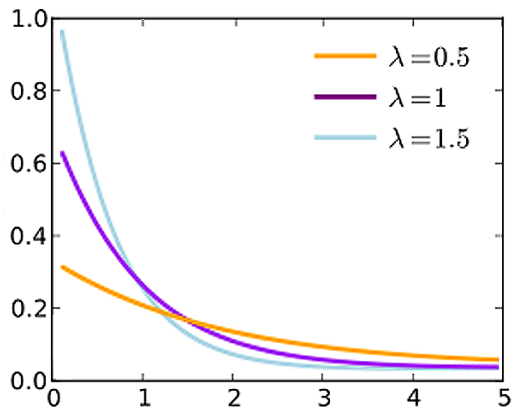
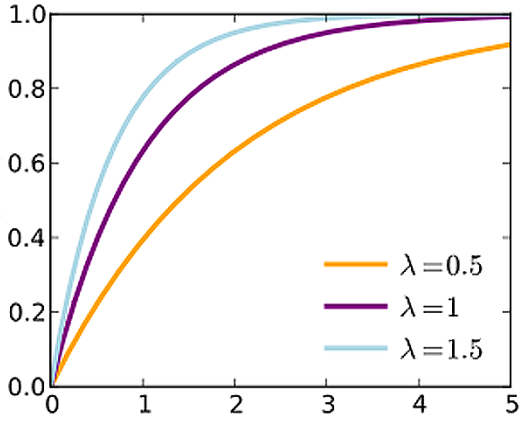
Probability Density Function	Cumulative Distribution Function
	

Exponential Distribution

The exponential distribution is the probability distribution of the time between events in a Poisson point process, i.e., a process in which events occur continuously and independently at a constant average rate. The continuous random variable X follows an exponential distribution if its probability density function is,

$$f(x,\lambda) = \begin{cases} \lambda\, e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

The following figures give the PDF and CDF of some exponentially distributed functions.

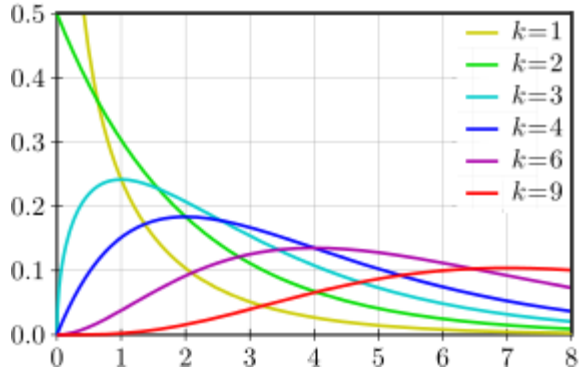
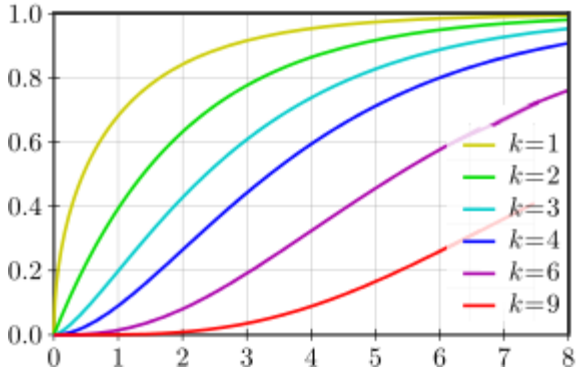
Probability Density Function	Cumulative Distribution Function
	

Chi-Squared Distribution

The chi-square distribution (also chi-squared or χ^2-distribution) with k degrees of freedom is the distribution of a sum of the squares of k independent standard normal random variables. It is one of the most widely used probability distributions in inferential statistics, notably in hypothesis testing and in construction of confidence intervals. The continuous random variable X follows an chi-squared distribution if its probability density function is,

$$f(x,k) = \frac{1}{2^{k/2}\,\Gamma\left(\frac{k}{2}\right)}\, x^{\left(\frac{k}{2}-1\right)} e^{-\frac{x}{2}}$$

The following figures give the PDF and CDF of some chi-squared distributed functions.

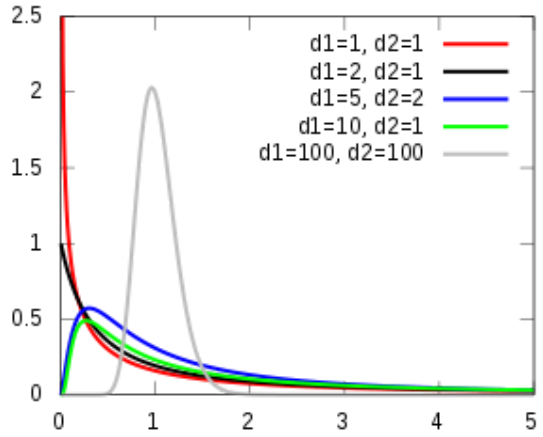
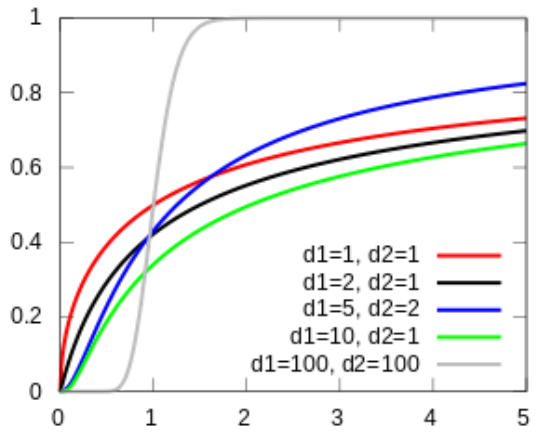
Probability Density Function	Cumulative Distribution Function
	

F Distribution

The F-distribution, also known as Snedecor's F distribution or the Fisher–Snedecor distribution is a continuous probability distribution that arises frequently as the null distribution of a test statistic, most notably in the analysis of variance. The general form of its probability density function is,

$$f\left(x, d_1, d_2\right) = \sqrt{\frac{\left(d_1 x\right)^{d_1} d_2^{d_2}}{\left(d_1 x + d_2\right)^{d_1+d_2}}} \Big/ x\, B\left(\frac{d_1}{2}, \frac{d_2}{2}\right)$$

The following figures give the PDF and CDF of some gamma distributed functions.

Probability Density Function	Cumulative Distribution Function
	

2.1.4. CENTRAL LIMIT THEOREM

The central limit theorem states that if there is a population with mean μ and standard deviation σ and if sufficiently large random samples are taken from the population with replacement, then the distribution of the sample means will be approximately normally distributed. This holds true regardless of whether the source population is normal or skewed, provided the sample size is sufficiently large (usually with $n \geq 30$). But, if the population is normal, then the theorem holds true even for smaller samples (usually with $n < 30$). In fact, this also holds true even if the population is binomial, provided that $min\,(np,\ n(1-p)) > 5$, where n is the sample size and p is the probability of success in the population. This means that one can use the normal probability model to quantify uncertainty when making inferences about a population mean based on the sample mean.

Samples

A population is an aggregate of creatures, things, cases and so on. A population commonly contains too many individuals to study conveniently, so an investigation is often restricted to one or more samples drawn from it. A well chosen sample contains most of the information about a particular population parameter but the relation between the sample and the population must be such as to allow true inferences to be made about a population from that sample. So, the first important attribute of a sample is that every individual in the population from which it is drawn must have a known non-zero chance of being included in it. Statistics (such as averages, standard deviations etc.) when taken from populations are referred to as population parameters. The following table gives the notations and formulae related to populations and their samples.

Type	Term	Notation	Formula
Common	Median	$Median$	Value separating the higher half from the lower half
	Mode	$Mode$	Value that appears most often
	Range	$Range$	Difference between the largest and smallest values
	First Quartile / Lower	$Q1$	25^{th} percentile splitting off the lowest 25%

	Quartile		of data from the highest 75%
	Second Quartile / Median	$Q2$	50^{th} percentile cutting data set in half
	Third Quartile / Upper Quartile	$Q3$	75^{th} percentile splitting off the highest 25% of data from the lowest 75%
	Interquartile Range	IQR	$IQR = Q3 - Q1$
	Mid-Range	M	Halfway between minimum & maximum $M = \frac{max\ value + min\ value}{2}$
Population $(X_1, X_2 \ldots X_N)$	Population Size	N	Number if elements in population
	Population Mean	μ	$\mu = \frac{1}{N}\sum_{i=1}^{N} X_i$
	Population Variance	σ^2	Mean squared deviation of each point from mean $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left(X_i - \mu\right)^2$
	Population Standard Deviation	σ	$\sigma = \sqrt{\sigma^2}$
	Mean Absolute Deviation	MAD	Mean of absolute deviation of each point from mean $MAD = \frac{1}{N}\sum_{i=1}^{N}\left\|x_i - \mu\right\|$
Sample $(X_1, X_2 \ldots X_n)$ (Sample of population)	Sample Size	n	Number if elements in sample
	Sample Mean	$\overline{X}$	$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$
	Sample Variance	S^2	$S^2_{n-1} = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2$ $(n-1)$ is used for un-biasing the result
	Sample Standard Deviation	S	$S = \sqrt{S^2_{n-1}}$

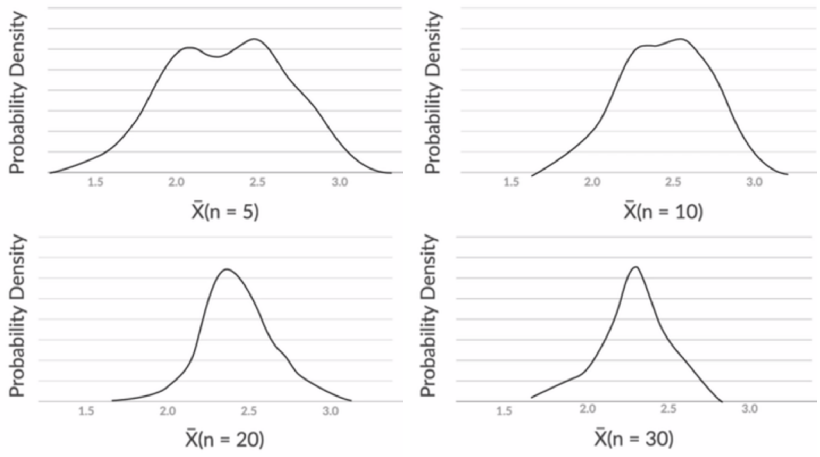
Sampling Distributions

The sampling distribution, specifically the sampling distribution of the sample means, is a probability density function for the sample means of a population. This distribution has some very interesting properties, which will later help one in estimating the sampling error. The following table gives the sampling distributions for the ball picking game.

Population	Sampling Distributions

Serial No.	Name	X (No. of red balls)
1	Manish	3
2	Rohit	2
3	Pearl	4
4	Prakhar	3
.	.	.
.	.	.
39	Rajiv	2
40	Ajay	3
41	Romil	3
42	Nikhil	2
43	Himanshu	2
44	Parth	3
45	Raman	2
.	.	.
.	.	.
70	Sachin	1
71	Salma	2
72	Sakshi	3
73	Sandeep	3
74	Kushal	4
75	Suri	1

$\bar{x}_1 = 2.8$
$\bar{x}_2 = 2.0$
$\bar{x}_3 = 2.6$
$\bar{x}_4 = 3.2$
.
.
$\bar{x}_{39} = 2.0$
$\bar{x}_{40} = 2.8$
$\bar{x}_{41} = 2.2$
$\bar{x}_{42} = 2.6$
$\bar{x}_{43} = 2.0$
$\bar{x}_{44} = 2.6$
$\bar{x}_{45} = 3.0$
.
.
$\bar{x}_{95} = 2.8$
$\bar{x}_{96} = 2.6$
$\bar{x}_{97} = 1.8$
$\bar{x}_{98} = 2.4$
$\bar{x}_{99} = 2.4$
$\bar{x}_{100} = 2.6$



The following table gives the notations and formulae related to sampling distributions.

Type	Term	Notation	Formula
Sampling Distribution of Sample Mean $(\overline{X}_1, \overline{X}_2 \ldots \overline{X}_k)$ (k sample means)	Sampling Distribution's Size	k	No convention
	Sampling Distribution's Mean	$\mu_{\overline{X}}$	Mean of sample means $\mu_{\overline{X}} = \mu$
	Sampling Distribution's Standard Deviation or Standard Error	SE	Dispersion of sample means around the population mean $SE = \sigma/\sqrt{n}$

Central Limit Theorem

The central limit theorem says that, for any kind of data, provided a high number of samples has been taken, the following properties hold true,

1. Sampling distribution's mean $(\mu_{\overline{X}})$ = Population mean (μ).
2. Sampling distribution's standard deviation $(SE) = \sigma/\sqrt{n}$.
3. For $n > 30$, the sampling distribution becomes a normal distribution.

Thus using the CLT, instead of collecting the data from the whole population only a good enough sample population can be used to collect the data for getting inference about the whole population. However, it would not be fair to infer that the population mean data is going to be exactly equal to the sample mean data. This is because the flaws of the sampling process must have led to some error. Hence, the sample mean's value needs to be reported with some margin of error. This margin of error is known as the confidence interval.

If there is a sample with sample size n, mean $\overline{X}$ and standard deviation S, then the confidence interval corresponding to y percentage of confidence level for μ is given by the range,

$$y = \left(\overline{X} - \frac{Z^* S}{\sqrt{n}}, \overline{X} + \frac{Z^* S}{\sqrt{n}} \right)$$

Here, Z^* is the $z-score$ associated with a y percentage of confidence level, i.e. the population mean and sample mean differ by a margin of error given by $Z^* S/\sqrt{n}$. It should be noted that one should use the standard deviation of the entire population, but if there are enough observations (i.e. $n \geq 30$) then the standard deviation for the sample can be used. Some of the most commonly used values of Z^* are as follows.

Confidence Interval	Z^*
90 %	± 1.65
95 %	± 1.96
99 %	± 2.58

At this point, it is important to address a very common misconception. Sampling distributions are just a theoretical exercise and one is not actually expected to make one in real life. If one wants to estimate the population mean, one needs to just take a sample rather than create an entire sampling distribution. The theoretical study of sampling distributions helped in learning more about CLT so that one can make all the assumptions as stated above. Consider the following examples.

1. The maximum permissible of lead in any food product is 2.5 ppm. The aim is to find the average content of lead in one of the food products. A sample population of $n = 100$ was taken having $\overline{X} = 2.3\ ppm$ and $S = 0.3\ ppm$. So before reporting the population mean μ one needs to find the confidence interval. This being a very sensitive task upon which the entire business depends so a high confidence level of 99 % needs to be chosen giving the confidence interval of $\overline{X} \pm (Z^* S/\sqrt{n})$, i.e 2.3 ± 0.077.

   So, one can conclude that the mean lead content in the product lies in the range of $2.223\ ppm$ to $2.377\ ppm$ with a confidence level of 99 %.

2. The amount of paracetamol specified by the drug regulatory authorities is $500\ mg$ with an allowed error of 10 %. Anything below $450\ mg$ is a quality issue as the drug becomes ineffective, while anything above $550\ mg$ is a serious regulatory issue. In a pharma company there are a number of identical manufacturing plants, each of which produces approximately $10,000$ tablets per hour. The aim is to ensure that the manufacturing process is running successfully by measuring the average amount of paracetamol in each tablet. A sample population of $n = 100$ tablets were taken having $\overline{X} = 530\ mg$ and $S = 100\ mg$. Upon calculating the average amount of paracetamol in each tablet it was found that the content of paracetamol was in the range of $513.5 - 546.5\ mg$, $510.4 - 549.6\ mg$ and $504.2 - 555.8\ mg$ for 90 %, 95 % and 99 % of confidence level respectively.

   Thus, it can be claimed that the tablet is fit to consume and effective only at a confidence level of 90 %.

3. A certain website is surveying which of the two features A and B is better. The survey was conducted on a sample population of $n = 10,000$. Now it was found that 50.5 % of people preferred the feature B over the feature A. If the random variable X is taken to be the proportion of people that prefer feature B over the feature A, then, for this sample

$\overline{X} = 0.505$ ($50.5\,\%$) and assume $S = 0.2$ ($20\,\%$). So, to find the actual population mean μ one needs to first find the margin of error.

If the margin of error is taken to be $1\,\%$, then that would mean that μ, which is the proportion of people that prefer feature B over the feature A, lies between the range $50.5 \pm 1\,\%$, i.e. $49.5\,\%$ to $51.5\,\%$. It would then be difficult for anyone to say with certainty whether μ would be more than $50\,\%$ or not. So, even though the proportion of people that prefer feature B over the feature A is more than $50\,\%$ in the sample, one would not be able to say with certainty that this proportion would be more than $50\,\%$ for the entire population.

On the other hand, if the margin of error is taken to be $0.3\,\%$, then one would be able to say that the population mean lies within the range $50.5 \pm 0.3\,\%$, i.e. $50.2\,\%$ to $50.8\,\%$. So, one would be able to say with certainty that the proportion of people that prefer feature B over the feature A is more than $50\,\%$ in the sample and for the entire population too. The margin of error corresponding to $90\,\%$ confidence level is given by $Z^* S/\sqrt{n} = 0.0033$ ($0.33\,\%$), and the population mean lies between $50.5 \pm 0.33\,\%$, i.e. $50.17\,\%$ to $50.83\,\%$.

Hence, one can say that feature B should replace feature A with a confidence of $90\,\%$.

HYPOTHESIS TESTING

2.2.1. CONCEPTS OF HYPOTHESIS TESTING

The statistical analyses learnt in Inferential Statistics enable one to make inferences about the population mean from the sample data when there is no idea of the population mean. However, sometimes one needs to have some starting assumption about the population mean and requires to confirm those assumptions using the sample data. It is here that the hypothesis testing comes into play.

Hypothesis Testing

One should not confuse between inferential statistics and hypothesis testing owing to their similar terminologies. The inferential statistics is used to find some population parameter (mostly population mean) when there is no initial number to start with. So, one starts with the sampling activity and finds out the sample mean. Then, the population mean is estimated from the sample mean using the confidence interval. Whereas the Hypothesis testing is used to confirm a conclusion (or hypothesis) about the population parameter (which is known from EDA or intuition). Through hypothesis testing, one can determine whether there is enough evidence to conclude if the hypothesis about the population parameter is true or not. The various steps in Hypothesis Testing are,

1. Formulating null and alternate hypotheses.
2. Making a decision.

Null and Alternate Hypotheses

The Hypothesis Testing starts with the formulation of these two hypotheses,

1. Null hypothesis (H_0): The status quo or prevailing belief about the population.
2. Alternate hypothesis (H_1): The challenge to the status quo or complement of the null hypothesis.

Some examples of null and alternate hypotheses are given in the following table.

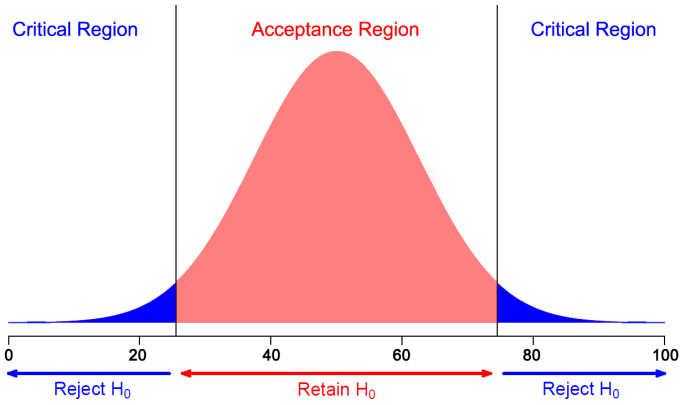
Scenario	Hypotheses
Criminal Trial	H_0: Defendant is innocent. H_1: Defendant is not innocent.
Average lead content in food products should be less than $2.5\ ppm$	H_0: Average lead content is less than or equal to $2.5\ ppm$. H_1: Average lead content is more than $2.5\ ppm$.
A company claims its total valuation is at least $\$10$ billion	H_0: Total valuation is more than or equal to $\$10$ H_1: Total valuation is less than $\$10$
Average demand of AC units is 350 units per month in summers	$H_0: \mu = 350$ $H_1: \mu \neq 350$

As can be seen for instances where the claim statement has words like *"at least"*, *"at most"*, *"less than"* or *"greater than"*, then one cannot formulate the null hypothesis just from the claim statement as it is not necessary that the claim is always about the status quo. In such scenarios one can use the following rules to formulate the null and alternate hypotheses,

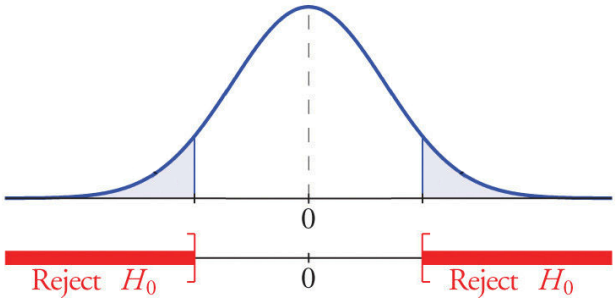
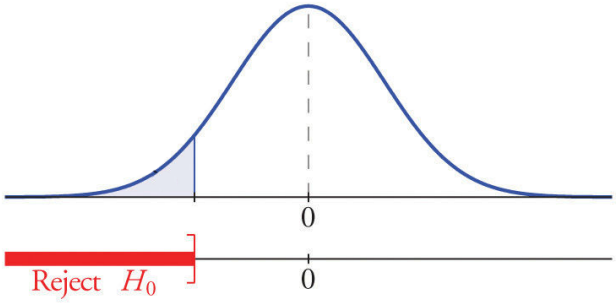
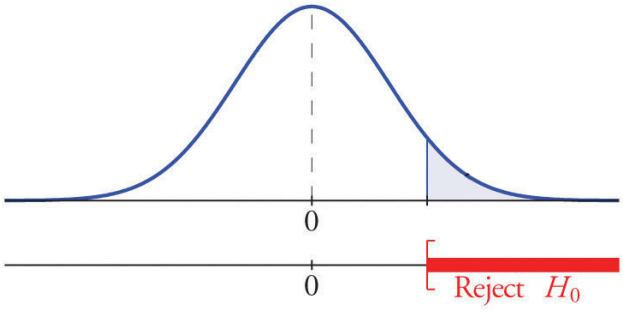
1. The null hypothesis (H_0) always has the following signs : $=\ or\ \leq\ or\ \geq$
2. The alternate hypothesis (H_1) always has the following signs : $\neq\ or\ >\ or\ <$

Making a Decision

The next step in Hypothesis Testing is making the decision to either reject or fail to reject the null hypothesis. If the sample mean lies in the criticals region then one can reject the null hypothesis. But, if the sample mean lies in the acceptance region then one fails to reject the null hypothesis.



The formulation of the null and alternate hypotheses determines the type of the test and the position of the critical regions in the normal distribution. One can tell the type of the test and the position of the critical region on the basis of the sign in the alternate hypothesis. The following table gives the same.

Sign in H_1	Test Type	Graphical Representation
$H_1 : \mu \neq \mu_0$	Two-tailed Test - Rejection region on both sides of distribution	
$H_1 : \mu < \mu_0$	Lower-tailed Test - Rejection region on left side of distribution	
$H_1 : \mu > \mu_0$	Upper-tailed Test - Rejection region on right side of distribution	

It is also important to understand that one can never accept the null hypothesis, rather can only fail to reject it. This is because the whole testing takes place with an aim to reject the present status quo, which is what the alternative hypothesis is. So, one tries to gather support for the alternative hypothesis so that one can reject the null hypothesis. If one is unable to reject the null hypothesis then it does not automatically imply that one has accepted the null hypothesis. Rather, it only means that there is lack of sufficient evidence to prove that the sample has properties significantly different from the null hypothesis. Similarly, one can never reject the alternative hypothesis, rather can only fail to reject the null hypothesis. Some of the widely used techniques for hypothesis testing are z-Test, t-Test, Chi-Square Test and ANOVA Test.

Types of Errors

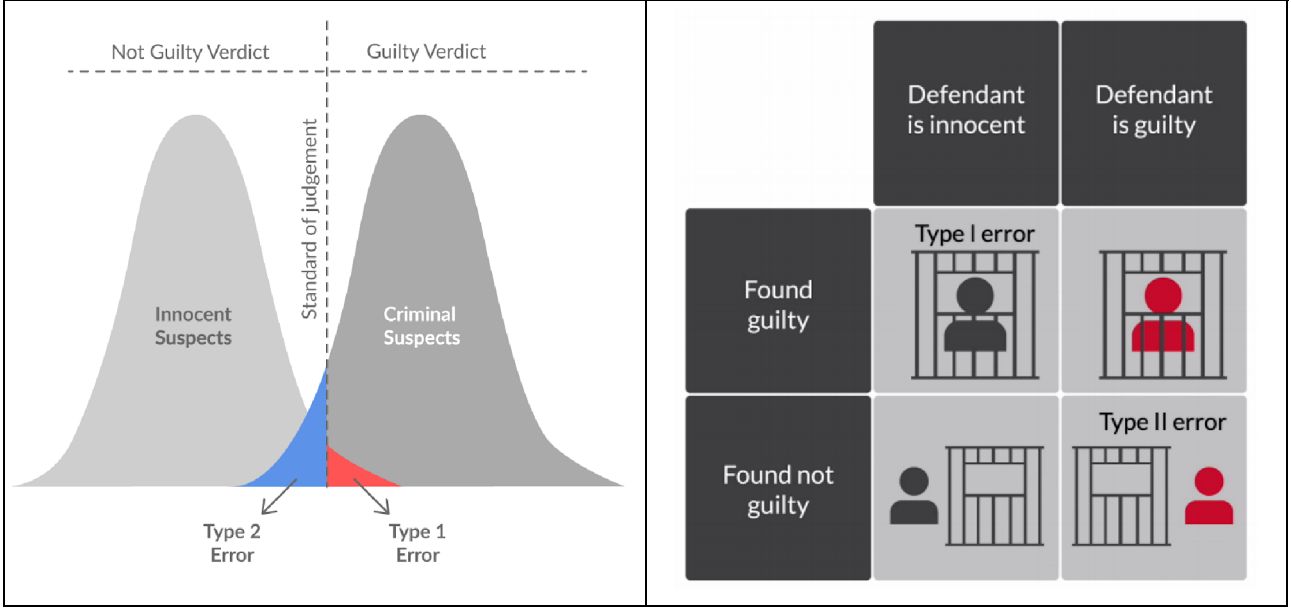
There are two types of errors that can result during the hypothesis testing process.

1. Type-I Error : It is represented by α and occurs when a true null hypothesis is rejected.
2. Type-II Error : It is represented by β and occurs when a false null hypothesis is failed to be rejected.

Type of Error		Null Hypothesis	
		True	**False**
Decision	**Fail to Reject Null Hypothesis**	Correct Decision	**Type-II Error** (β)
	Reject Null Hypothesis	**Type-I Error** (α)	Correct Decision

The power of any hypothesis test is defined by $1-\beta$. If one goes back to the analogy of the criminal trial example, one would find that the probability of making a Type-I error would be more if the jury convicts the accused even on less substantial evidence. The Type-I error can be reduced if the jury adopts more stringent criteria to convict an accused party. However, reducing the probability of a Type-I error may increase the probability of making a Type-II error, i.e. if the jury becomes very liberal in acquitting the people on trial, there would be a higher probability that an actual criminal walks free. The following figure represents the same.

Jury Example for Error Types



Z-Test

A z-Test is a statistical test used to determine whether two population means are different when the variances are known and the sample size is large. The test statistic is assumed to have a normal distribution, and nuisance parameters such as standard deviation should be known in order to perform the test accurately. The test statistic used is given by $z = (\overline{X} - \mu)/(\sigma/\sqrt{n})$.

Z-Test : Critical Value Method

1. The null and alternate hypotheses are formulated.

2. A new quantity α (also known as the significance level) is defined for the test. It refers to the proportion of the sample mean lying in the critical region. This value corresponds to the probability of observing such an extreme value by chance or the probability of rejecting the null hypothesis when it is true. One can take any value for α such as $0.01\ (1\ \%)$, $0.05\ (5\ \%)$, $0.1\ (10\ \%)$ and so on as per the sensitivity of the test. A significance level of 0.05 indicates that there is $5\ \%$ risk of concluding that a difference exists when actually there is no difference.

3. The cumulative probability of the UCV (upper critical value) is calculated from the value of α. Then the $z-score$ (Z_c) is calculated using the UCV and the Z-table as shown in the following table.

z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	.5000	.5040	.5080	.5120	.5160	.5199	.5239	.5279	.5319	.5359
0.1	.5398	.5438	.5478	.5517	.5557	.5596	.5636	.5675	.5714	.5753
0.2	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
0.3	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
0.4	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
0.5	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
0.6	.7257	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7517	.7549
0.7	.7580	.7611	.7642	.7673	.7704	.7734	.7764	.7794	.7823	.7852
0.8	.7881	.7910	.7939	.7967	.7995	.8023	.8051	.8078	.8106	.8133
0.9	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621
1.1	.8643	.8665	.8686	.8708	.8729	.8749	.8770	.8790	.8810	.8830
1.2	.8849	.8869	.8888	.8907	.8925	.8944	.8962	.8980	.8997	.9015
1.3	.9032	.9049	.9066	.9082	.9099	.9115	.9131	.9147	.9162	.9177
1.4	.9192	.9207	.9222	.9236	.9251	.9265	.9279	.9292	.9306	.9319
1.5	.9332	.9345	.9357	.9370	.9382	.9394	.9406	.9418	.9429	.9441
1.6	.9452	.9463	.9474	.9484	.9495	.9505	.9515	.9525	.9535	.9545
1.7	.9554	.9564	.9573	.9582	.9591	.9599	.9608	.9616	.9625	.9633
1.8	.9641	.9649	.9656	.9664	.9671	.9678	.9686	.9693	.9699	.9706
1.9	.9713	.9719	.9726	.9732	.9738	.9744	.9750	.9756	.9761	.9767
2.0	.9772	.9778	.9783	.9788	.9793	.9798	.9803	.9808	.9812	.9817
2.1	.9821	.9826	.9830	.9834	.9838	.9842	.9846	.9850	.9854	.9857
2.2	.9861	.9864	.9868	.9871	.9875	.9878	.9881	.9884	.9887	.9890
2.3	.9893	.9896	.9898	.9901	.9904	.9906	.9909	.9911	.9913	.9916
2.4	.9918	.9920	.9922	.9925	.9927	.9929	.9931	.9932	.9934	.9936
2.5	.9938	.9940	.9941	.9943	.9945	.9946	.9948	.9949	.9951	.9952
2.6	.9953	.9955	.9956	.9957	.9959	.9960	.9961	.9962	.9963	.9964
2.7	.9965	.9966	.9967	.9968	.9969	.9970	.9971	.9972	.9973	.9974
2.8	.9974	.9975	.9976	.9977	.9977	.9978	.9979	.9979	.9980	.9981
2.9	.9981	.9982	.9982	.9983	.9984	.9984	.9985	.9985	.9986	.9986
3.0	.9987	.9987	.9987	.9988	.9988	.9989	.9989	.9989	.9990	.9990

The $z-score$ is the sum of the $z-values$ of the corresponding horizontal and vertical axis of the table.

For example if UCV is equal to 0.975 then the $z-score$ is equal to $1.9 + 0.06 = 1.96$.

4. The critical values, UCV and LCV are calculated from the value of Z_c using the formula $\mu \pm (Z_c S/\sqrt{n})$.

5. Finally the decision is made on the basis of the value of the sample mean x with respect to the critical values UCV and LCV.

Consider the following examples using the critical value method to make a decision about an hypothesis.

1. A manufacturer claims that the average life of its product is 36 months. An auditor selects a sample of size $n = 49$ units of the product having average life of $\overline{X} = 34.5$ months and standard deviation of $S = 4$ months. The hypotheses being considered are, $H_0: \mu = 36$ months and $H_1: \mu \neq 36$ months. This being a two-tailed test, for a significance level of $3\,\%$ (i.e. $\alpha = 0.03$), the UCV can be calculated as $UCV = 1 - (0.03/2) = 0.9850$. The $z - score$ for 0.9850 can be calculated from Z-table as 2.17 (2.1 on the horizontal axis and 0.07 on the vertical axis). The LCV and UCV can then be calculated as $LCV = 36 - (2.17 \times 4/\sqrt{49}) = 34.76$ and $UCV = 36 + (2.17 \times 4/\sqrt{49}) = 37.24$.

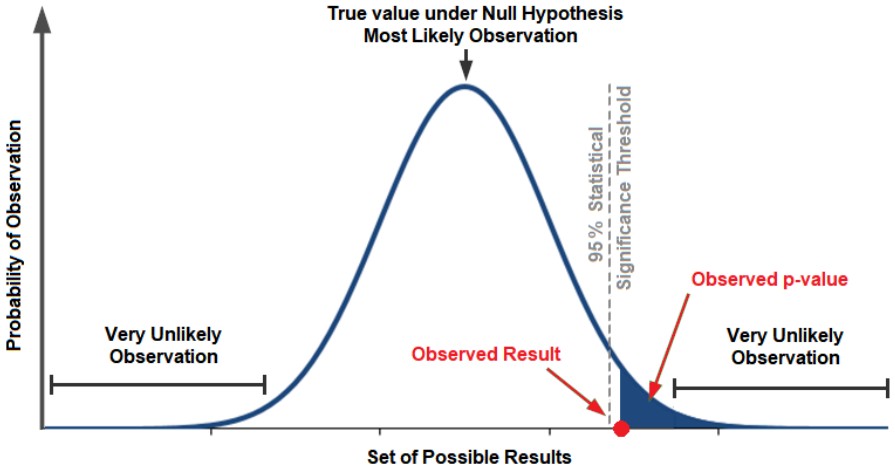
   It can be seen that the sample mean in this case is 34.5 months, which is less than the LCV. This implies that the sample mean lies in the critical region and thus the null hypothesis can be rejected.

2. A retail chain claims that the average demand of AC units is at most 350 units per month in summers. A sample of $n = 36$ stores were taken having average sales of $\overline{X} = 370.16$ units and standard deviation of $S = 90$ units. The hypotheses being considered are, $H_0: \mu \leq 350$ units and $H_1: \mu > 350$ units. This being a one-tailed test (more specifically a upper-tailed test), upon taking a significance level of $5\,\%$ (i.e. $\alpha = 0.05$), the UCV can be calculated as $UCV = 1 - 0.05 = 0.9500$. The $z - score$ for 0.9500 can be calculated from Z-table as 1.645 (as the value 0.9500 is not present in the Z-table, the nearest values 0.9495 and 0.9505 can be taken and the average of their $z - values$ be considered i.e $(1.64 + 1.65)/2 = 1.645$). The UCV can then be calculated as $UCV = 350 + (1.645 \times 90/\sqrt{36}) = 374.67$.

   It can be seen that the sample mean in this case is 370.16 units, which is less than the UCV. This implies that the sample mean lies in the acceptance region and thus one fails to reject the null hypothesis.

Z-Test : *p*-Value Method

A $p - value$ measures the strength of evidence in support of a null hypothesis. Suppose the test statistic in a hypothesis test is equal to K. Then the $p - value$ is the probability of observing a test statistic as extreme as K, assuming that the null hypothesis is true. If the $p - value$ is less than the significance level, then the null hypothesis is rejected. The following figure shows the interpretation of $p - value$.



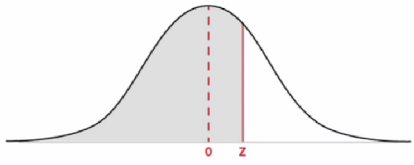
It should be noted that $p-values$ are not the probability of making a mistake for rejecting a true null hypothesis rather they are the probability of obtaining an effect at least as extreme as the one in the sample data, assuming the truth of the null hypothesis. The following steps describe the p-value method for hypothesis testing.

1. The null and alternate hypotheses are formulated.

2. The $z-score$ is calculated for the sample mean point $\overline{X}$ on the distribution using the formula $z-score=(\overline{X}-\mu)/(\sigma/\sqrt{n})$.

3. The $p-value$ is then calculated from the cumulative probability for the given calculated $z-score$ using the Z-table. To find the correct $p-value$ from the $z-score$, the cumulative probability is first found out by simply checking the Z-table (which gives the area under the curve till that point) as shown in the following table.

z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	.5000	.5040	.5080	.5120	.5160	.5199	.5239	.5279	.5319	.5359
0.1	.5398	.5438	.5478	.5517	.5557	.5596	.5636	.5675	.5714	.5753
0.2	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
0.3	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
0.4	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
0.5	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
0.6	.7257	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7517	.7549
0.7	.7580	.7611	.7642	.7673	.7704	.7734	.7764	.7794	.7823	.7852
0.8	.7881	.7910	.7939	.7967	.7995	.8023	.8051	.8078	.8106	.8133
0.9	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621
1.1	.8643	.8665	.8686	.8708	.8729	.8749	.8770	.8790	.8810	.8830
1.2	.8849	.8869	.8888	.8907	.8925	.8944	.8962	.8980	.8997	.9015
1.3	.9032	.9049	.9066	.9082	.9099	.9115	.9131	.9147	.9162	.9177
1.4	.9192	.9207	.9222	.9236	.9251	.9265	.9279	.9292	.9306	.9319
1.5	.9332	.9345	.9357	.9370	.9382	.9394	.9406	.9418	.9429	.9441
1.6	.9452	.9463	.9474	.9484	.9495	.9505	.9515	.9525	.9535	.9545
1.7	.9554	.9564	.9573	.9582	.9591	.9599	.9608	.9616	.9625	.9633
1.8	.9641	.9649	.9656	.9664	.9671	.9678	.9686	.9693	.9699	.9706
1.9	.9713	.9719	.9726	.9732	.9738	.9744	.9750	.9756	.9761	.9767
2.0	.9772	.9778	.9783	.9788	.9793	.9798	.9803	.9808	.9812	.9817
2.1	.9821	.9826	.9830	.9834	.9838	.9842	.9846	.9850	.9854	.9857
2.2	.9861	.9864	.9868	.9871	.9875	.9878	.9881	.9884	.9887	.9890
2.3	.9893	.9896	.9898	.9901	.9904	.9906	.9909	.9911	.9913	.9916
2.4	.9918	.9920	.9922	.9925	.9927	.9929	.9931	.9932	.9934	.9936
2.5	.9938	.9940	.9941	.9943	.9945	.9946	.9948	.9949	.9951	.9952
2.6	.9953	.9955	.9956	.9957	.9959	.9960	.9961	.9962	.9963	.9964
2.7	.9965	.9966	.9967	.9968	.9969	.9970	.9971	.9972	.9973	.9974
2.8	.9974	.9975	.9976	.9977	.9977	.9978	.9979	.9979	.9980	.9981
2.9	.9981	.9982	.9982	.9983	.9984	.9984	.9985	.9985	.9986	.9986
3.0	.9987	.9987	.9987	.9988	.9988	.9989	.9989	.9989	.9990	.9990
3.1	.9990	.9991	.9991	.9991	.9992	.9992	.9992	.9992	.9993	.9993
3.2	.9993	.9993	.9994	.9994	.9994	.9994	.9994	.9995	.9995	.9995
3.3	.9995	.9995	.9995	.9996	.9996	.9996	.9996	.9996	.9996	.9997
3.4	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9998



Area under curve till the z-value

The $p-value$ is the sum of the $z-values$ of the corresponding horizontal and vertical axis of the table.

For example if $z-score$ is equal to 1.34 then the $p-value$ is equal to $1-0.9099=0.0901$

4. A decision is made on the basis of the $p-value$ with respect to the given significance value α. It should be noted that the $p-value$ is multiplied by two for a two-tailed test.

Consider the following examples using the $p-value$ method to make a decision about an hypothesis.

1. A manufacturer claims that the average life of its product is 36 months. An auditor selects a sample of size $n=49$ units of the product having average life of $\overline{X}=34.5$ months and standard deviation of $S=4$ months. The hypotheses being considered are, $H_0:\ \mu=36$ months and $H_1:\ \mu\neq 36$ months. The value of the $z-score$ can be calculated for the sample mean point $(\overline{X}=34.5)$ as $(34.5-36)/(4/\sqrt{49})=-2.62$ (since the sample mean lies on the left side of the hypothesised mean of 36 months, the $z-score$ comes out to be negative). The $z-value$ for -2.62 can be calculated from Z-table as 0.0044 (value corresponding to

-2.6 on the horizontal axis and 0.02 on the vertical axis). Since the sample mean is on the left side of the distribution and this is a two-tailed test, the $p-value$ would be $2 \times 0.0044 = 0.0088$.

For a significance level of 3%, it can be seen that the $p-value$ is less than the significance level $(0.0088 < 0.03)$. Thus, the null hypothesis can be rejected.

2. A retail chain claims that the average demand of AC units is at most 350 units per month in summers. A sample of $n = 36$ stores were taken having average sales of $\overline{X} = 370.16$ units and standard deviation of $S = 90$ units. The hypotheses being considered are, $H_0: \mu \leq 350$ units and $H_1: \mu > 350$ units. The value of the $z-score$ can be calculated for the sample mean point $(\overline{X} = 370.16)$ as $(370.16 - 350)/(90/\sqrt{36}) = 1.344$ (since the sample mean lies on the right side of the hypothesised mean of 350 units, the $z-score$ comes out to be positive). The $p-value$ for 1.344 can be calculated from Z-table as $1 - 0.90988 = .0895$ (value corresponding to 1.3 on the horizontal axis and 0.04 on the vertical axis). Since the sample mean is on the right side of the distribution and this is a upper-tailed test, the $p-value$ would remain the same $.0895$.

   For a significance level of 5%, it can be seen that the $p-value$ is more than the significance level $(0.0895 > 0.05)$. Thus, one fails to reject the null hypothesis.

T-Test

A t-Test is a statistical test used to compare the mean of two given samples. Similar to z-test, the t-test also assumes a normal distribution of the sample. It is used when the population parameters (such as mean μ, standard deviation σ etc.) are not known. The test statistic used is given by $t = (\overline{X} - \mu)/(S/n)$.

The method for hypothesis testing is similar to that of z-Test critical value method, except that the T-Table is used instead of the Z-Table while calculating the value of Z_c. The T-table contains the values of Z_c for a given degree of freedom and significance level as shown in the following table.

Degrees of Freedom	.005 (1-tailed) .01 (2-tailed)	.01 (1-tailed) .02 (2-tailed)	.023 (1-tailed) .05 (2-tailed)	.05 (1-tailed) .10 (2-tailed)	.10 (1-tailed) .20 (2-tailed)	.25 (1-tailed) .50 (2-tailed)
1	63.657	31.821	12.706	6.314	3.078	1.000
2	9.925	6.965	4.303	2.920	1.886	.816
3	5.841	4.541	3.182	2.353	1.638	.765
4	4.604	3.747	2.776	2.132	1.533	.741
5	4.032	3.365	2.571	2.015	1.476	.727
6	3.707	3.143	2.447	1.943	1.440	.718
7	3.500	2.896	2.365	1.895	1.415	.711
8	3.355	2.896	2.306	1.960	1.397	.706
9	3.250	2.821	2.262	1.833	1.383	.703
10	3.169	2.764	2.228	1.812	1.372	.700
11	3.106	2.718	2.201	1.769	1.363	.697
12	3.054	2.681	2.179	1.782	1.356	.696
13	3.012	2.650	2.160	1.771	1.350	.694
14	2.977	2.625	2.145	1.761	1.345	.692
15	2.947	2.602	2.132	1.753	1.341	.691

The $z-score$ is the corresponding value for the degree of freedom and the significance level.

For example if degree of freedom is 8 and significance level is 0.05 (single-tailed) then the $z-score$ is equal to 1.960.

Consider the following examples for finding the $z-score$ using the T-Table.

1. The standard deviation of a sample of size $n=25$. For a two-tailed hypothesis test of a significance level of 5% the value of $Z_c=2.064$ ($z-score$ for $df=25-1=24$ and $\alpha=0.05$ using T-Table).

2. The standard deviation of a sample of size $n=32$. For a two-tailed hypothesis test of a significance level of 5% the value of $Z_c=1.96$ ($z-score$ for $df=32-1=31$ and $\alpha=0.05$ using T-Table). As the degrees of freedom is greater than 29 the value of Z_c can also be found using the Z-Table which gives $Z_c=1.96$ ($z-score$ for $1-(0.05/2)=0.975$). This is because for a sample size ≥ 30, the t-distribution is the same as the z-distribution.

T-Test : One-Sample Mean Test

This test is used to compare the mean of a sample to a pre-specified value and tests for a deviation from that value. Consider the following examples.

1. The average birth weight for babies born in cities in India is 2.9 Kg. One wants to compare the average birth weight of a sample of babies born in villages to this value.

2. The recommended total cholesterol level by doctors is below 200 mg/dl. One wants to test if the samples gathered from all the metro cities are statistically different, on average, from this recommended level.

T-Test : Paired Two-Sample Mean Test

This test is used when the sample observations are from the same individual or object. During this test, the same subject is tested twice. The paired two-sample test and the one-sample test are actually the same test in disguise. Consider the following examples.

1. There is a new drug being tested. One would need to compare the sample before and after the drug is taken to check if the results are different or not.

2. There is a hypothesis that Virat Kohli performs better or as good in the second innings of a test match as the first innings. This would be a two-sample mean test, where sample 1 would contain his score from the first innings and sample 2 would contain his score from the second innings. This would be a paired test since each row in the data would correspond to the same match.

T-Test : Unpaired Two-Sample Mean Test

This test is used when the sample observations are independent. During this test, the same subject is not tested twice. Consider the following examples.

1. There is a new drug being tested. One needs to compare its effectiveness to that of the standard available drug. So, one needs to take a sample of patients who have consumed the new drug and compare it with another sample who have consumed the standard drug.

2. There is a hypothesis that Virat Kohli performs better than Sachin Tendulkar in the second innings of a ODI match. This would be a two-sample mean test, where sample 1 would contain Kohli's score from the second innings and sample 2 would contain Sachin's score

from the second innings. This would be an unpaired test since each row in the data would correspond to different matches.

T-Test : Two-Sample Proportion Test

This test is used when the sample observations are categorical, with two categories (it could be True/False, 1/0, Yes/No, Male/Female, Success/Failure etc.). Consider the following example.

1. Two drugs are being compared for the effectiveness of two drugs. The desired outcome of the drug is defined as success. So, one needs to take a sample of patients who have consumed the new drug and record the number of successes and compare it with successes in another sample who have consumed the standard drug.

T-Test : A/B Testing

A/B testing is a direct industry application of the two-sample proportion test. Consider the following example.

1. During the development of an e-commerce website, there could be different opinions about the choices of various elements, such as the shape of buttons, the text on the call-to-action buttons, the colour of various UI elements, the copy on the website, or numerous other such things. Often, the choice of these elements are very subjective and it is very difficult to predict which option would perform better. To resolve such conflicts, one can use A/B-Testing. It provides a way to test two different versions of the same element and check which one performs better.

Chi-Square Test

A Chi-Square Test is a statistical test used to determine whether there is a statistically significant difference (i.e. a magnitude of difference that is unlikely to be due to chance alone) between the expected frequencies and the observed frequencies in one or more categories of a so-called contingency table. The test statistic used is given by $\chi^2 = \Sigma[\ (Observed - Expected)^2 / Expected\ ]$.

The method for hypothesis testing is similar to the z-Test p-value method, except that the Chi-Square Table is used instead of the Z-Table while calculating the p-value. The Chi-Square Table contains the Chi-Square values for a given degree of freedom and significance level as shown in the following table.

d.f.	.995	.99	.975	.95	.9	.1	.05	.025	.01
1	0.00	0.00	0.00	0.00	0.02	2.71	3.84	5.02	6.63
2	0.01	0.02	0.05	0.10	0.21	4.61	5.99	7.38	9.21
3	0.07	0.11	0.22	0.35	0.58	6.25	7.81	9.35	11.34
4	0.21	0.30	0.48	0.71	1.06	7.78	9.49	11.14	13.28
5	0.41	0.55	0.83	1.15	1.61	9.24	11.07	12.83	15.09
6	0.68	0.87	1.24	1.64	2.20	10.64	12.59	14.45	16.81
7	0.99	1.24	1.69	2.17	2.83	12.02	14.07	16.01	18.48
8	1.34	1.65	2.18	2.73	3.49	13.36	15.51	17.53	20.09
9	1.73	2.09	2.70	3.33	4.17	14.68	16.92	19.02	21.67
10	2.16	2.56	3.25	3.94	4.87	15.99	18.31	20.48	23.21
11	2.60	3.05	3.82	4.57	5.58	17.28	19.68	21.92	24.72
12	3.07	3.57	4.40	5.23	6.30	18.55	21.03	23.34	26.22
13	3.57	4.11	5.01	5.89	7.04	19.81	22.36	24.74	27.69
14	4.07	4.66	5.63	6.57	7.79	21.06	23.68	26.12	29.14
15	4.60	5.23	6.26	7.26	8.55	22.31	25.00	27.49	30.58
16	5.14	5.81	6.91	7.96	9.31	23.54	26.30	28.85	32.00
17	5.70	6.41	7.56	8.67	10.09	24.77	27.59	30.19	33.41

The $p - value$ is the corresponding value for the degree of freedom and the significance level.

For example if degree of freedom is 8 and significance level is 0.05 then the $p - value$ is equal to 15.51.

Chi-Square Test : Independence Test

This test is used to determine if there is a significant relationship between two nominal (categorical) variables from a single population. Consider the following example.

1. A public opinion poll surveyed a sample of $n = 1000$ voters. Respondents were classified by the first categorical variable gender (male or female) and by second categorical variable voting preference (Republican, Democrat, or Independent). Results are shown in the following contingency table.

	Republican	Democrat	Independent	Total
Male	200	150	50	400
Female	250	300	50	600
Total	450	450	100	1000

The hypotheses being considered are, H_0: There is no relationship between gender and voting preference and H_1: There is a relationship between gender and voting preference. The expected values are calculated assuming null hypothesis is correct. Thus, expected values are $E_{Male,\ Republican} = (400 \times 450)/1000 = 180$, $E_{Female,\ Republican} = (600 \times 450)/1000 = 270$ and so on. The test statistic $\chi^2 = [(200-180)^2/180] + [(250-270)^2/270] + ... = 16.2$. For a significance level of $5\,\%$ and $df = (levels\ of\ gender - 1) \times (levels\ of\ voting\ preference - 1) = (2-1) \times (3-1) = 2$, the $p - value = 0.0003$ (i.e. the probability that the test statistic is more extreme than 16.2).

Since, the $p - value$ is less than the significance level $(0.0003 < 0.05)$, the null hypothesis can be rejected.

Chi-Square Test : Goodness of Fit

This test is used to determine whether the sample data is consistent with a hypothesized distribution when there is one nominal (categorical) variable from a single population. Consider the following example.

1. A Toy Company prints cricket cards. The company claims that $30\,\%$ of the cards are rookies, $60\,\%$ veterans but not All-Stars, and $10\,\%$ are veteran All-Stars. The hypotheses being considered are, H_0: The proportion of rookies, veterans and All-stars is $30\,\%$, $60\,\%$ and $10\,\%$ respectively and H_1: At least one of the proportions is not true. For a sample of $n = 100$ cards the count of various types were 50 rookies, 45 veterans and 5 All-Stars. The expected values are calculated assuming null hypothesis is correct. Thus, expected values are $E_{rookie} = 0.3 \times 100 = 30$, $E_{veteran} = 0.6 \times 100 = 60$ and so on. The test statistic $\chi^2 = [(50-30)^2/30] + [(45-60)^2/60] + ... = 19.58$. For a significance level of $5\,\%$ and $df = (levels\ of\ player - 1) = 3 - 1 = 2$, the $p - value = 0.0001$ (i.e. the probability that the test statistic is more extreme than 19.58).

Since, the $p - value$ is less than the significance level $(0.0001 < 0.05)$, the null hypothesis can be rejected.

F-Test : ANOVA

It is a statistical test in which the test statistic has a F-distribution under the null hypothesis. The F-Test is used to assess whether the expected values of a quantitative variable within several pre-defined groups differ from each other.

The Two Sample t-Tests can only validate hypotheses containing only two groups at a time. For samples involving three or more groups, the t-Test then becomes tedious as one has to perform the tests for each combination of the groups. Also, the Type-I error increases in this process. So, analysis of variance (ANOVA) is used to statistically assess the equality of means. ANOVA can determine whether the means of three or more groups are different. It uses F-Tests to statistically test the equality of means. Consider the following example.

1. A test was conducted in a workplace, and the feedback on three e-commerce platforms were recorded in a dataset. The following table shows the data.

Amazon	7.5	8.5	6	10	8.5	8	8	6	9.5	10	6.5
Flipkart	7	9.5	10	6	7.5	8.5	10	6.5	6.5	9	10
Snapdeal	5	7.5	8.5	3	6	5	7				

The hypotheses being considered are, H_0: All the platforms are equally popular $(\mu_1 = \mu_2 = ... = \mu_p)$ and H_1: At least one of the platforms has different popularity from the rest $(\mu_1 \neq \mu_2 \neq ... \neq \mu_p)$. The following table gives the method of calculation of F-Ratio.

Source of Variation	Sum Of Squares
Between : Variation of mean of the groups from the grand mean	$SSB = \sum_{j=1}^{p} n_j \left(\bar{x}_j - \bar{x} \right)^2$
Within : Variation of observations of a group from the group mean	$SSW = \sum_{j=1}^{p} \sum_{i=1}^{n_j} \left(x_{ij} - \bar{x}_j \right)^2$
Total : Variation of all observations from the grand mean	$TSS = \sum_{j=1}^{p} \sum_{i=1}^{n_j} \left(x_{ij} - \bar{x} \right)^2$ or $TSS = SSB + SSW$
i - observations in group, j - particular group, n_j - total observations in group, x_{ij} - all observations, $\bar{x}_j$ - particular group mean, $\bar{x}$ - grand mean, p - groups	

Sum of Squares	Degrees of Freedom	Mean Square Value	F-Ratio
SSB	$df_B = p - 1$	$MSB = SSB/df_B$	$F = \frac{MSB}{MSW}$
SSW	$df_W = x_{ij} - p$	$MSW = SSW/df_W$	
TSS	$df_T = df_B + df_W$		

The calculated values are given in the following table.

Variation	Sum of Squares	Degrees of Freedom	Mean Square Value	F-Ratio
Between	$SSB = 24.4184$	$df_B = 3 - 1 = 2$	$MSB = 12.2092$	$F = 4.7800$
Within	$SSW = 66.4090$	$df_W = 29 - 3 = 26$	$MSW = 2.5541$	
Total	$TSS = 90.8275$	$df_T = 2 + 26 = 28$		

For a significance level of $5\,\%$, $df_B = 2$ and $df_W = 26$, the $critical\ value = 3.3690$ ($F - score$ for $df_B = 2$, $df_W = 26$ and $\alpha = 0.05$ using F-Table). The F-table for $\alpha = 0.05$ contains the critical values for the various degrees of freedom (rows represent denominator degrees of freedom and the columns represent numerator degrees of freedom) as shown in the following table.

F - Distribution ($\alpha = 0.05$ in the Right Tail)

df_2 \ df_1	1	2	3	4	5	6	7	8	9
1	161.45	199.50	215.71	224.58	230.16	233.99	236.77	238.88	240.54
2	18.513	19.000	19.164	19.247	19.296	19.330	19.353	19.371	19.385
3	10.128	9.5521	9.2766	9.1172	9.0135	8.9406	8.8867	8.8452	8.8123
4	7.7086	6.9443	6.5914	6.3882	6.2561	6.1631	6.0942	6.0410	6.9988
5	6.6079	5.7861	5.4095	5.1922	5.0503	4.9503	4.8759	4.8183	4.7725
6	5.9874	5.1433	4.7571	4.5337	4.3874	4.2839	4.2067	4.1468	4.0990
7	5.5914	4.7374	4.3468	4.1203	3.9715	3.8660	3.7870	3.7257	3.6767
8	5.3177	4.4590	4.0662	3.8379	3.6875	3.5806	3.5005	3.4381	3.3881
9	5.1174	4.2565	3.8625	3.6331	3.4817	3.3738	3.2927	3.2296	3.1789
10	4.9646	4.1028	3.7083	3.4780	3.3258	3.2172	3.1355	3.0717	3.0204
11	4.8443	3.9823	3.5874	3.3567	3.2039	3.0946	3.0123	2.9480	2.8962
12	4.7472	3.8853	3.4903	3.2592	3.1059	2.9961	2.9134	2.8486	2.7964
13	4.6672	3.8056	3.4105	3.1791	3.0254	2.9153	2.8321	2.7669	2.7144
14	4.6001	3.7389	3.3439	3.1122	2.9582	2.8477	2.7642	2.6987	2.6458
15	4.5431	3.6823	3.2874	3.0556	2.9013	2.7905	2.7066	2.6408	2.5876
16	4.4940	3.6337	3.2389	3.0069	2.8524	2.7413	2.6572	2.5911	2.5377
17	4.4513	3.5915	3.1968	2.9647	2.8100	2.6987	2.6143	2.5480	2.4943
18	4.4139	3.5546	3.1599	2.9277	2.7729	2.6613	2.5767	2.5102	2.4563
19	4.3807	3.5219	3.1274	2.8951	2.7401	2.6283	2.5435	2.4768	2.4227
20	4.3512	3.4928	3.0984	2.8661	2.7109	2.5990	2.5140	2.4471	2.3928
21	4.3248	3.4668	3.0725	2.8401	2.6848	2.5727	2.4876	2.4205	2.3660
22	4.3009	3.4434	3.0491	2.8167	2.6613	2.5491	2.4638	2.3965	2.3419
23	4.2793	3.4221	3.0280	2.7955	2.6400	2.5277	2.4422	2.3748	2.3201
24	4.2597	3.4028	3.0088	2.7763	2.6207	2.5082	2.4226	2.3551	2.3002
25	4.2417	3.3852	2.9912	2.7587	2.6030	2.4904	2.4047	2.3371	2.2821
26	4.2252	3.3690	2.9752	2.7426	2.5868	2.4741	2.3883	2.3205	2.2655
27	4.2100	3.3541	2.9604	2.7278	2.5719	2.4591	2.3732	2.3053	2.2501
28	4.1960	3.3404	2.9467	2.7141	2.5581	2.4453	2.3593	2.2913	2.2360
29	4.1830	3.3277	2.9340	2.7014	2.5454	2.4324	2.3463	2.2783	2.2229
30	4.1709	3.3158	2.9223	2.6896	2.5336	2.4205	2.3343	2.2662	2.2107
40	4.0847	3.2317	2.8387	2.6060	2.4495	2.3359	2.2490	2.1802	2.1240
60	4.0012	3.1504	2.7581	2.5252	2.3683	2.2541	2.1665	2.0970	2.0401
120	3.9201	3.0718	2.6802	2.4472	2.2899	2.1750	2.0868	2.0164	1.9588
∞	3.8415	2.9957	2.6049	2.3719	2.2141	2.0986	2.0096	1.9384	1.8799

The $F - critical\ value$ is the corresponding value for the various degrees of freedom for a particular significance level.

For example if degree of freedom is 2 and 26 for a significance level of 0.05 then the $F - critical\ value$ is equal to 3.3690.

Since, the $F - critical\ value$ is less than the calculated $F - value$ $(3.3690 < 4.7800)$, the null hypothesis can be rejected.

Python Code - Hypothesis Testing

Chi-Square Test

```
>> from scipy.stats import chi2_contingency
>> stat, p, dof, expected = chi2_contingency([data['col_1'], data['col_2']])
```

One-Sample Mean t-Test

```
>> from scipy.stats import ttest_1samp()
>> stat, p = stats.ttest_1samp(data['col_1'], mean_value)
```

Unpaired Two-Sample Mean t-Test

```
>> from scipy.stats import ttest_ind
>> stat, p = ttest_ind(data['col_1'], data['col_2'])
```

Paired Two-Sample Mean t-Test

```
>> from scipy.stats import ttest_rel
>> stat, p = ttest_rel(data['col_1'], data['col_2'])
```

F-Test ANOVA

```
>> from scipy.stats import f_oneway
>> stat, p = f_oneway(data['col_1'], data['col_2'], data['col_3'])
```

EXPLORATORY DATA ANALYSIS

2.3.1. DATA SOURCING

EDA is the process of exploring data with the aim of extracting useful and actionable information from it. It is arguably the most important and revelatory step in any kind of data analysis. The various steps involved in EDA are,

1. Data Sourcing
2. Data Cleaning
3. Univariate Analysis
4. Bivariate Analysis
5. Derived Metrics

Data Sourcing

There are two major kinds of data which can be classified according to the source,

1. Public data : A large amount of data collected by the government or other public agencies is made public for the purposes of research. Such data sets do not require special permission for access and are therefore called public data.

2. Private data : It is that data which is sensitive to organisations and is thus not available in the public domain. Banking, telecom, retail, and media are some of the key private sectors that rely heavily on this data to make decisions.

It should be noted that public data isn't always relevant and private data isn't always easily available. The following data sources are handy when looking for data sets.

1. https://github.com/awesomedata/awesome-public-datasets
2. https://data.gov.in/
3. https://github.com/datameet

Public Data

Public data is available on the internet on various platforms. A lot of data sets are available for direct analysis, whereas some of the data have to be manually extracted and converted into a format that is fit for analysis.

Private Data

A large number of organisations seek to leverage data analytics to make crucial decisions. As organisations become customer-centric, they utilise insights from data to enhance customer experience, while also optimising their daily processes. While banks use data to make credit related decisions, telecoms use it to optimise plans for customers and predict customer churn. While retail data analytics helps drive decisions such as pricing and stocking, the HR data analytics helps identify and predict employee behaviour. The media industry uses the data extensively to target viewers better, while the advertisers use it to identify best avenues for targeting customers and the journalists use the same data for visualisation to aid information.

2.3.2. DATA CLEANING

Once the data has been procured, the next step is to clean it to get rid of data quality issues. There are various types of quality issues when it comes to data, and that's why data cleaning is one of the most time-consuming steps of data analysis. For example, there could be formatting errors (such as

rows and columns are ill-formatted, unclearly named etc.), missing values, repeated rows, spelling inconsistencies etc. These issues could make it difficult to analyse data and could lead to errors or irrelevant results. Thus, these issues need to be corrected using data cleaning before the data is analysed.

Formatting Errors

Formatting errors such as ill-formatted and unclearly named rows and columns need to be addressed first. The following steps are used to correct some of these issues at the level of rows and columns.

Fixing Rows :

1. Delete summary rows such as total, subtotal rows etc.
2. Delete incorrect rows such as header rows, footer rows etc.
3. Delete extra rows such as column number, indicators, blank rows, page number etc.

Fixing Columns :

1. Add column names if missing.
2. Rename columns which are abbreviated or encoded consistently to provide proper information.
3. Delete unnecessary columns.
4. Align the misaligned or shifted columns properly.
5. Split columns for more data (such as addresses can be splitted to get street, city, state etc.) to analyse each separately.
6. Merge columns for creating unique identifiers (such as name with address) if required.

Missing Values

Another common data quality issue is the missing values. If there are reliable external sources then one can replace the missing values with the information. But often, it is better to let missing values be and continue with the analysis rather than extrapolating the available information as good methods add information, bad methods exaggerate the information. The following steps are used to treat missing values.

1. Identify values that indicate missing data but are not recognised by the software as such (disguised missing values such as blank strings, *“NA”*, *“XX”*, *“999”*, etc.) as missing.

2. Either add the reliable data from external sources or better keep it as such rather than exaggerating the data.

3. Delete rows if the number of missing values are significant in number, as this would not impact the analysis. Similarly, delete columns if the missing values are quite significant in number.

4. Fill partial missing values using business judgement (such as missing time zone, century, etc.) as such values are easily identifiable.

Standardising Values

Scaling of data ensures that all the values have a common scale, which makes analysis easier. Along with this, removing outliers is another important step in data cleaning as it may disproportionately affect the results of the analysis and may lead to faulty interpretations. There is no fixed definition of

an outlier. It is left up to the judgment of the analyst to decide the criteria on which data can be categorised as abnormal or an outlier. The following steps are used to standardise values.

1. Standardise all the values so as to ensure that all observations under a variable have a common and consistent unit (such as converting lbs to kgs, miles/hr to km/hr, etc.)

2. Scale the values if required to ensure that the observations under a variable have a common scale.

3. Standardise the precision for better presentation of data (such as 4.5312341 kgs to 4.53 kgs etc.).

4. Remove outliers that would disproportionately affect the results of analysis.

5. Remove extra characters (such as common prefix/suffix, leading/trailing/multiple spaces, etc.) in the textual values as these are irrelevant during analysis.

6. Standardise the case for textual data (such as using uppercase, lowercase, title case, sentence case etc.).

7. Standardise the format wherever required (such as 23/10/16 to 2016/10/20, *"Modi, Narendra"* to *"Narendra Modi"*, etc.).

Invalid Values

A data set can contain invalid values in various forms. Some of the values could be truly invalid (such as a string *"tr8ml"* in a variable containing mobile numbers, a height of $11\,ft$ in a set containing heights of children etc.) On the other hand, some other invalid values can be corrected (such as a numeric value with a data type of string, junk characters due to wrong encoding etc.). The following steps are used to clean invalid values.

1. Encode the data properly while reading values (such as changing the encoding to CP1252 instead of UTF-8 if the data is being read as junk characters etc.).

2. Convert the incorrect data types (such as converting numeric values stored as strings into number, strings into date etc.).

3. Correct the values that go beyond logical range (such as temperature less than $-273^\circ\,C$, height of human more than $20\,ft$ etc.) A close look at the data helps in checking if there is scope for correction in the value or if the value needs to be removed.

4. Delete the invalid values and treat them as missing values (such as wrong data, values not belonging to the categorical list etc.).

5. Correct the wrong structure or remove the values that don't follow a defined structure (such as 12 digit pin code, 20 digit mobile contact number etc.)

6. Validate the internal rules (such as a date of delivery must definitely be after the date of the order etc.).

Filtering Data

The last stage of data cleaning is the filtering of data. Though there will be a largely accurate data set by now, one might not need the entire data set for analysis. It is important to understand what one needs to infer from the data and then choose the relevant parts of the data set for analysis. Thus, one needs to filter the data to get what one needs for analysis. The following steps are used to filter data.

1. Remove duplicate data (such as identical rows, rows where some columns are identical etc.).

2. Filter the rows to get only the rows relevant to the analysis (such as filter by segment, filter by date period etc.).

3. Filter the columns by picking only the columns relevant to the analysis.

4. Aggregate the data by grouping the data by required keys and aggregate the rest.

2.3.3. UNIVARIATE ANALYSIS

The term univariate itself suggests that it deals with analysing variables one at a time. It is important to separately understand each variable before moving on to analysing multiple variables together. Given a clean data set, the first step is to understand what it contains. Information about a data set can be gained simply by looking at its metadata. Metadata, in simple terms, is the data that describes each variable in detail (information such as the size of the data set, how and when the data set was created, what the rows and variables represent, etc.). There are various ways of classifying variables. The most simple way of classifying variables is as,

1. Categorical variables.
2. Quantitative/Numeric variables.

Similarly, there is one other method of classifying variables known as Steven's typology. Steven's typology classifies variables into four types.

1. Nominal variables.
2. Ordinal variables.
3. Interval variables.
4. Ratio variables.

Categorical Variables

These are qualitative data which are used to categorize data into various categories (such as male/female, yes/no, etc.). These are again divided into,

1. Unordered - categorical data with no notion of high-low, more-less etc. (such as red-blue-green, home-personal-auto loan etc.).

2. Ordered - categorical data with some kind of ordering (such as high-medium-low, graduate-masters-doctorate etc.).

Quantitative/Numeric Variables

These are simply numeric variables which can be added up, multiplied, divided etc. (such as salary, number of bank accounts, runs scored etc.).

Nominal Variables

These are categorical variables, where the categories differ only by their names, there is no order among categories (such as red/blue/green, male/female etc.). These are the most basic forms of categorical variables.

Ordinal Variables

These are categorical variables, where the categories follow a certain order, but the mathematical difference between categories is not meaningful (such as primary school/high school/college, high/medium/low, bad/good/excellent etc.). The ordinal variables are nominal as well.

Interval Variables

These are categorical variables which follow a certain order where the mathematical difference between categories is meaningful but division or multiplication is not (such as temperature in degrees celsius, dates etc.). The interval variables are both nominal and ordinal.

Ratio Variables

These are categorical variables which follow a certain order where apart from the mathematical difference, the ratio (division/multiplication) is possible (such as sales in dollars, marks of students etc.). The ratio variables are nominal, ordinal and interval type.

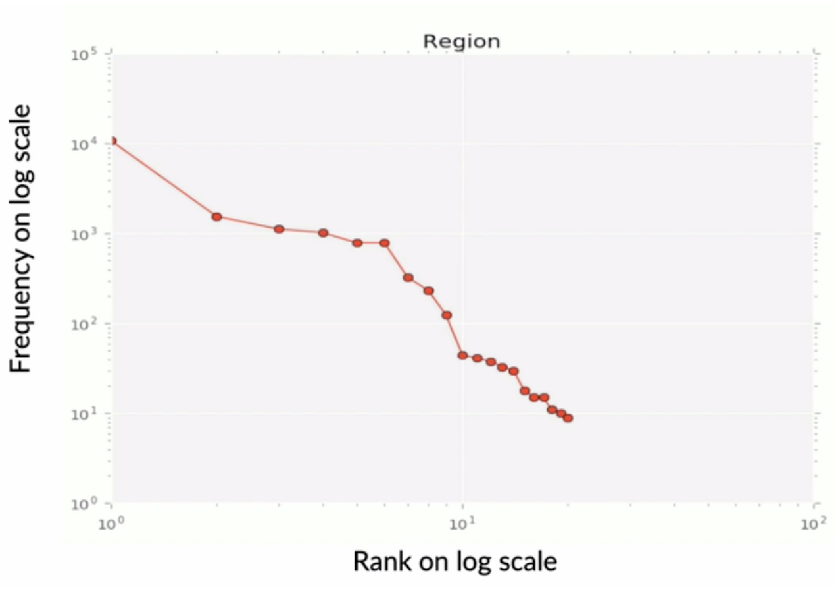
Univariate Analysis - Unordered Categorical Variables

Plots are immensely helpful in identifying hidden patterns in the data. Using simple rank-frequency plots one can extract meaning even from seemingly trivial unordered categorical variables (such as country, name of an artist, name of a github user etc.). Rank-frequency plots of unordered categorical variables, when plotted on a log-log scale, typically result in a power law distribution. Plotting on a log scale compresses the values to a smaller scale which makes the plot easy to read. The following figure shows an example of rank-frequency plot.

Rank-Frequency Table

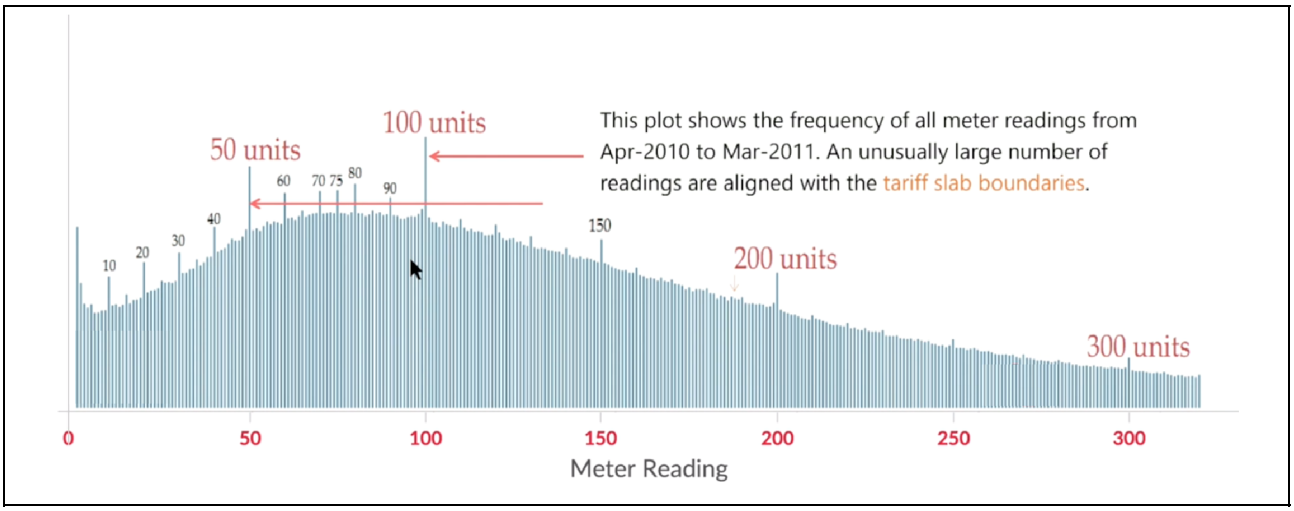
Rank	Region	Count
1	India	10780
2	Headstrong	1554
3	China	1130
4	Philippines	1030
5	US	792
6	Romania	788
7	Mexico	324
8	Guatemala	233
9	Poland	124
10	Brazil	45
11	Hungary	41
12	Colombia	38
13	Netherlands	33
14	South Africa	30
15	UK	18
16	UAE	15
17	GMS India	15
18	Japan	11
19	Czech Republic	10
20	Kenya	9

Power Law Distribution on Log-Log Plot



Univariate Analysis - Ordered Categorical Variables

Simple histogram plots can reveal very interesting insights for continuous or ordered categorical variables. Consider the following figure showing the histogram of power meter readings across households of a power company which gives very interesting and unexpected insights.

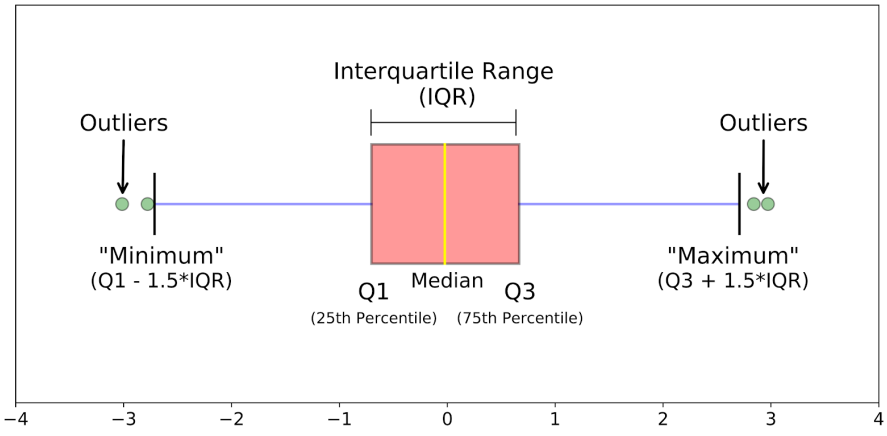


From the histogram one can see that there are huge spikes in readings aligned with the tariff slab boundaries as these are the ones where the power theft is being done and the reading adjusted accordingly to pay lower bills. The other spikes being seen are the ones where the readings have been cooked up by the agents sitting at offices so that they would not have to work to get the actual readings.

Univariate Analysis - Quantitative Variables

Mean and median are single values that broadly give a representation of the entire data. It is very important to understand when to use these metrics to avoid doing inaccurate analysis. While mean gives an average of all the values, median gives a typical value that could be used to represent the entire group. As a simple rule of thumb, use of mean should be questioned, since median is almost always a better measure of representativeness.

Standard deviation and interquartile difference are both used to represent the spread of the data. Interquartile difference is a much better metric than standard deviation if there are outliers in the data. This is because the standard deviation is influenced by outliers while the interquartile difference simply ignores them. Simple box plots can be used to check the spread of data as shown in the following figure.



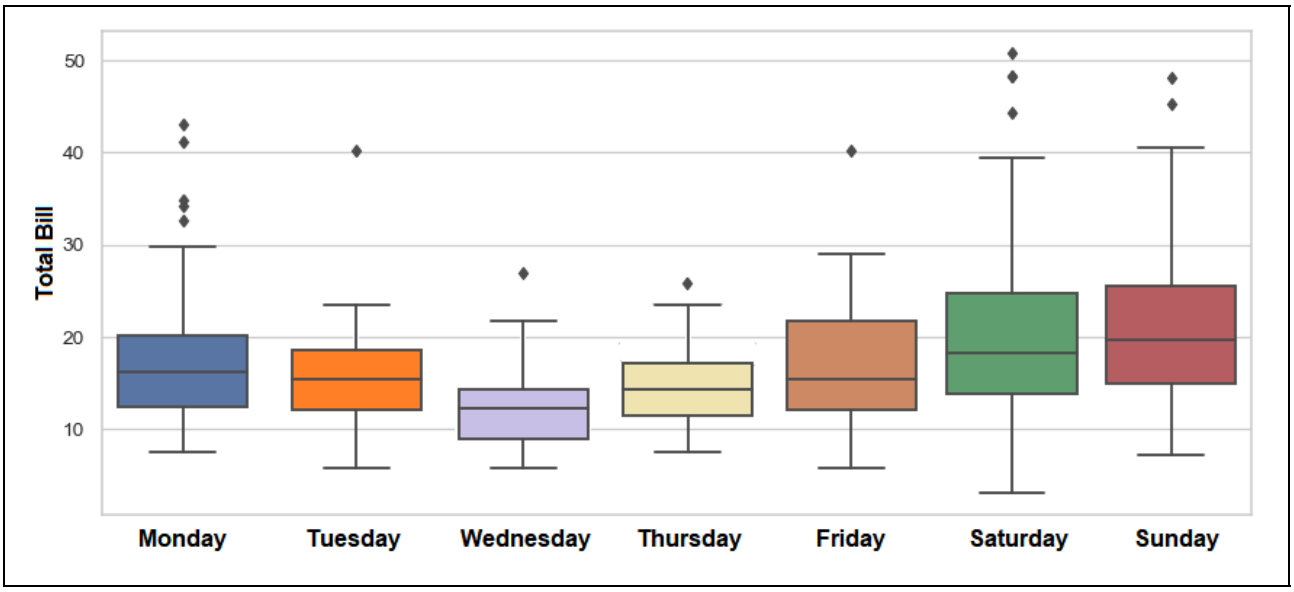
Segmented Univariate

Simple univariate analysis is tremendously useful in many cases, but the real strength of univariate analysis often lies in segmented univariate analysis. All the categorical variables can be segmented and the different segments be compared to get insights. Even continuous variables can be segmented by creating buckets.

The entire segmentation process can be divided into four parts,

1. Fetching the raw data.
2. Grouping by dimensions.
3. Summarising the data using a relevant metric such as mean, median, etc.
4. Comparing the aggregated metric across groups/categories.

The following figure gives one such example of segmented univariate analysis for the total bill on days of a week.



2.3.4. BIVARIATE ANALYSIS

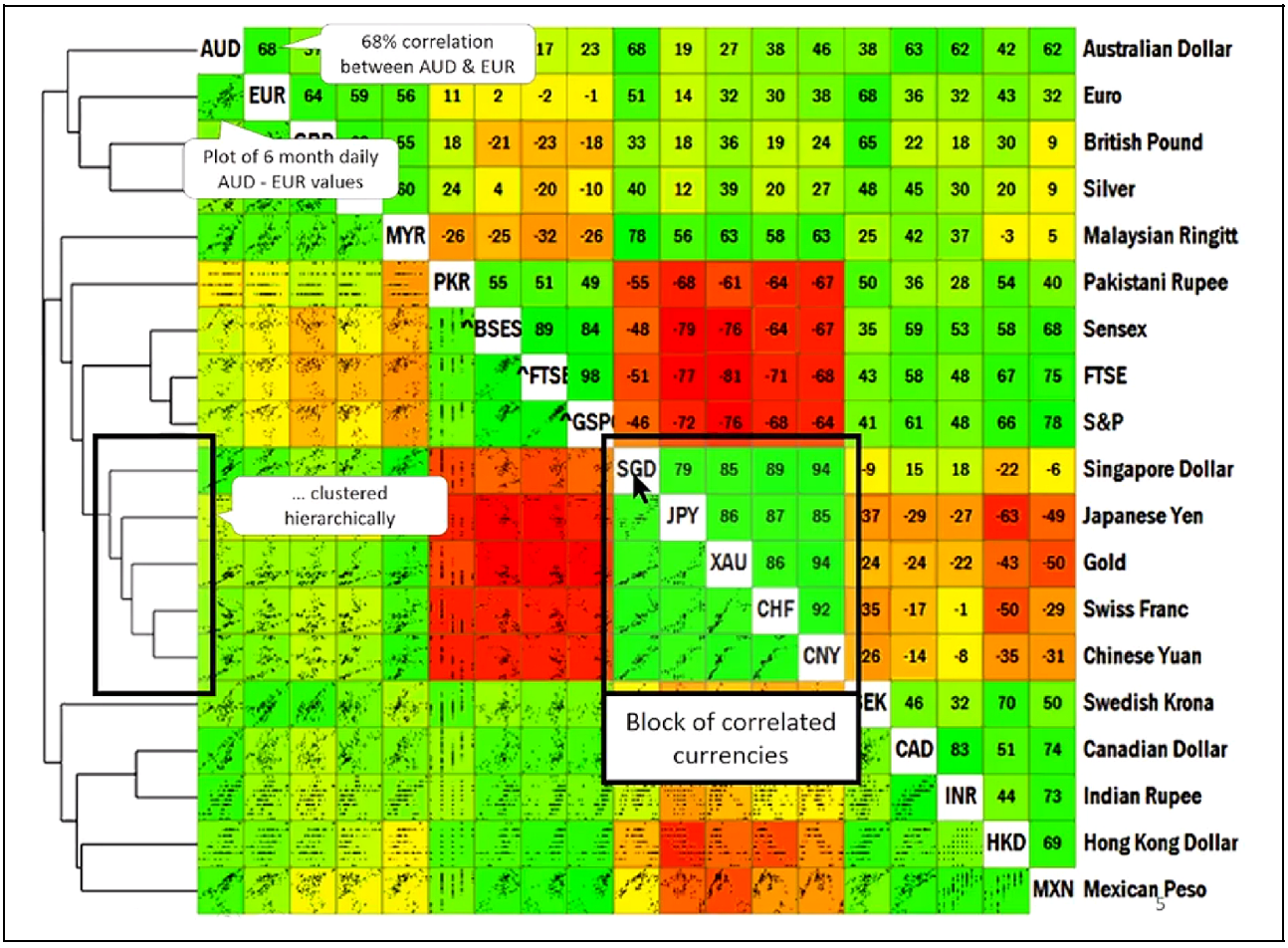
Bivariate Analysis gives the relationship between two variables. Bivariate analysis is one of the simplest forms of statistical analysis which involves the analysis of two variables for the purpose of determining the empirical relationship between them. It helps in determining to what extent one can predict a value for one variable (possibly a dependent variable) if the value of the other variable (possibly the independent variable) is known.

Bivariate Analysis on Continuous Variables

The most common kind of analysis being done between pairs of continuous variables is the correlation analysis. Correlation is a number between -1 and 1 which quantifies the extent to which two variables *'correlate'* with each other, i.e.

1. If one increases as the other increases, the correlation is positive.
2. If one decreases as the other increases, the correlation is negative.
3. If one stays constant as the other varies, the correlation is zero.

For example an increase in rain is accompanied by an increase in humidity, it shows a positive correlation. Similarly, as the price of a commodity decreases its demand increases shows a negative correlation. A perfect positive correlation means that the correlation coefficient is exactly 1 (i.e. as one variable moves either up or down, the other one moves in the same direction). Similarly, a perfect negative correlation means that the correlation coefficient is exactly -1 (i.e. the two variables move in opposite directions). A zero correlation (correlation coefficient is 0) implies no relationship at all. The following figure gives the correlation matrix of stock prices of different countries providing a real sense of the relationship between many variables.



As can be seen the correlated variables have been grouped by similarities, and the correlation has also been calculated for groups of variables. This is called clustering, where the idea is to form a hierarchy of similar groups of variables. The top-right half represents the correlation coefficient and the left bottom half has the scatter plot between the two variables.

Bivariate Analysis on Categorical Variables

The categorical bivariate analysis is essentially an extension of the segmented univariate analysis to another categorical variable. In segmented univariate analysis, one compares the metrics such as mean of the variable across various segments of a categorical variable (such as median income of educated parents is higher than that of uneducated ones, etc.). Whereas in the categorical bivariate analysis, this comparison is extended to other categorical variables (such as checking if the median income of educated parents is higher than that of uneducated ones in all states, etc.). One can even drill down into another categorical variable and get closer to the true patterns in the data. Such as check if the median income of educated parents is higher than that of uneducated ones (variable 1), in all states (variable 2), for all age groups (variable 3) etc. This can be called as a trivariate analysis and even though it gives a more granular version of the truth, it gets a bit complex to make sense of

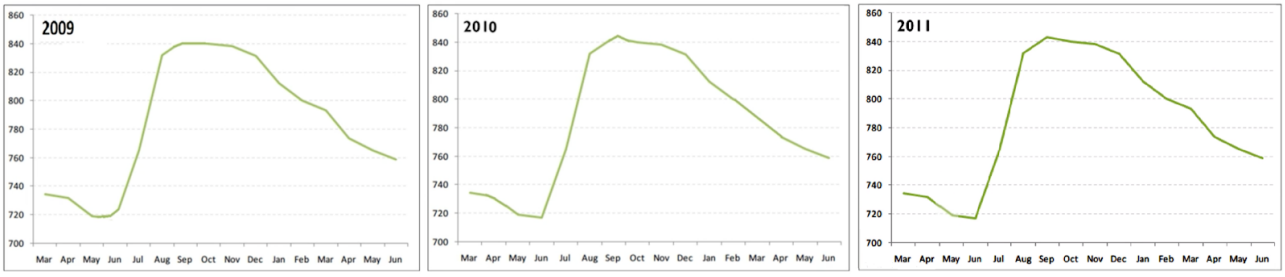
and explain to others and hence it is not usually done in EDA. Usually, one does not analyse more than two variables at a time, though there are ways to do that (machine learning models are essentially a way to do that).

A segmented univariate analysis may deceive one into thinking that a certain phenomenon is true without asking the question whether it is true for all sub-populations or is it true only when aggregated across the entire population. So, bivariate analysis is performed to check the influence of the variables. In general, there are two fundamental aspects of analysing categorical variables to draw conclusions, they are,

1. Checking the distribution of two categorical variables (such as checking the distribution of the occupations of parents across their educational qualifications using cross tables etc.).
2. Checking the distribution of two categorical variables with one continuous variable (such as checking the distribution of incomes of parents across their educational qualifications and occupation etc.).

2.3.5. DERIVED METRICS

Sometimes, one would not get the most valuable insights by analysing the data available. So, one often needs to create new variables using the existing ones to get meaningful insights. New variables can be created based on the business understanding or as suggested by the clients. The business understanding plays a very important role in deriving new variables. Consider the following figure showing the plot for marks scored against month of birth.



By plotting the marks against the month of birth (derived variable), one could observe that the children who were born after June would have missed the cutoff by a few days and gotten admission at the age of 5 instead of 4. The ones being born after June (such as July, August, etc.) were intellectually and emotionally more mature than their peers because of their higher age, resulting in better performance. This unexpected insight could not have been discovered without the derived variable. Broadly, there are three different types of derived metrics,

1. Type-driven metrics.
2. Business-driven metrics.
3. Data-driven metrics.

Type-Driven Metrics

These metrics can be derived by understanding the variable's typology. Understanding the types of variables enables one to derive new metrics of types different from the same column. For example, age in years is a ratio attribute, but one can convert it into an ordinal type by binning it into categories such as children (< 13 years), teenagers ($13-19$ years), young adults ($20-25$ years), etc. This helps in getting insights about the children, teenagers, young adults etc. such as are

teenagers better than children in studying, are young adults more likely to play than the teenagers and children and so on. Some more examples of type driven derived metrics are,

1. Emails : domain, host etc.
2. Image URL : format, size, type, etc.
3. Web : host, parameter, hashtag, etc.
4. Date : month, day, quarter, season, week, etc.
5. Name : first name, middle name, last name, title, etc.
6. Time : hour, minute, am/pm, morning, evening, shift, etc.
7. Dimensions : city, state, country, north, south, time-zone, etc.

Business-Driven Metrics

Every business has certain rules, based on which metrics can be derived. These are the metrics derived from the business perspective and completely domain specific. Extracting meaningful information from existing variables (such as month from date etc.) can be easy, but extracting information that requires business expertise is not an easy task. It requires a decent domain experience. Without understanding the domain correctly, deriving insights can be difficult and prone to errors. Some more examples of business driven derived metrics are,

1. Student marks : pass/fail, CGPA cut-off etc.
2. Banking : number of transactions, minimum average balance, rate of interest etc.
3. Cricket : scored a century, took five fer etc.

Data-Driven Metrics

These metrics can be created based on the variables present in the existing data set or on certain analysis. For example, if there are two variables in the data set such as weight and height which are highly correlated, one can derive a new metric Body Mass Index (BMI) for analysis instead of analysing weight and height variables separately. Once the BMI is found, one can easily categorise people based on their fitness (such as $BMI < 18.5$ should be considered as an underweight while category, while $BMI > 30.0$ can be considered as obese). This is how data-driven metrics can help one discover the hidden patterns out of the data.