# Assignment No: 3

## Aim/Problem Statement:-Basic Statistics - Measures of Central Tendencies and Variance

Perform the following operations on any open source dataset (eg. data.csv)

1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.
2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor' of iris.csv dataset.

Provide the codes with outputs and explain everything that you do in this step. .

## Theory:-

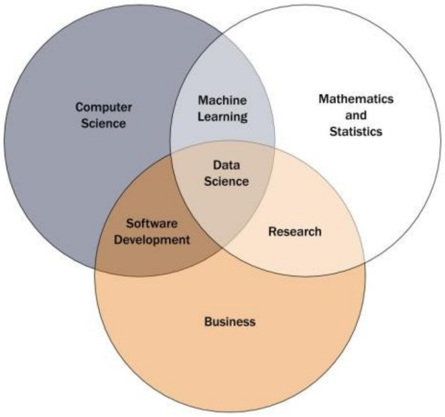
## Statistical Inference:

statistical inference as the process of generating conclusions about a population from a noisy sample. Without statistical inference we're simply living within our data. With statistical inference, we're trying to generate new knowledge.

Statistical analysis and probability influence our lives on a daily basis. Statistics is used to predict the weather, restock retail shelves, estimate the condition of the economy, and much more. Used in a variety of professional fields, statistics has the power to derive valuable insights and solve complex problems in business, science, and society. Without hard science, decision making relies on emotions and gut reactions. Statistics and data override intuition, inform decisions, and minimize risk and uncertainty.

In data science, statistics is at the core of sophisticated machine learning algorithms, capturing and translating data patterns into actionable evidence. Data scientists use statistics to gather, review, analyze, and draw conclusions from data, as well as apply quantified mathematical models to appropriate variables.

Data science knowledge is grouped into three main areas: computer science; statistics and mathematics; and business or field expertise. These areas separately result in a variety of careers, as displayed in the diagram below. Combining computer science and statistics without business knowledge enables professionals to perform an array of machine learning functions. Computer science and business expertise leads to software development skills. Mathematics and statistics (combined with business expertise) result in some of the most talented researchers. It is only with all three areas combined that data scientists can maximize their performance, interpret data, recommend innovative solutions, and create a mechanism to achieve improvements.



Statistical functions are used in data science to analyze raw data, build data models, and infer results. Below is a list of the key statistical terms:

- Population: the source of data to be collected.
- Sample: a portion of the population.
- Variable: any data item that can be measured or counted.
- Quantitative analysis (statistical): collecting and interpreting data with patterns and data visualization.
- Qualitative analysis (non-statistical): producing generic information from other non-data forms of media.
- Descriptive statistics: characteristics of a population.
- Inferential statistics: predictions for a population.
- Central tendency (measures of the center): mean (average of all values), median (central value of a data set), and mode (the most recurrent value in a data set).
- Measures of the Dispersion:
  - Range: the distance between each value in a data set.
  - Variance: the distance between a variable and its expected value.
  - Standard deviation: the dispersion of a data set from the mean.

### Statistical techniques for data scientists

There are a number of statistical techniques that data scientists need to master. When just starting out, it is important to grasp a comprehensive understanding of these principles, as any holes in knowledge will result in compromised data or false conclusions.

General statistics: The most basic concepts in statistics include bias, variance, mean, median, mode, and percentiles.

Probability distributions: Probability is defined as the chance that something will occur, characterized as a simple "yes" or "no" percentage. For instance, when weather reporting indicates a 30 percent chance of rain, it also means there is a 70 percent chance it will not rain. Determining the distribution calculates the probability that all those potential values in the study will occur. For example, calculating the probability that the 30 percent chance for rain will change over the next two days is an example of probability distribution.

Dimension reduction: Data scientists reduce the number of random variables under consideration through feature selection (choosing a subset of relevant features) and feature extraction (creating new features from functions of the original features). This simplifies data models and streamlines the process of entering data into algorithms.

Over and under sampling: Sampling techniques are implemented when data scientists have too much or too little of a sample size for a classification. Depending on the balance between two sample groups, data scientists will either limit the selection of a majority class or create copies of a minority class in order to maintain equal distribution.

Bayesian statistics: Frequency statistics uses existing data to determine the probability of a future event. Bayesian statistics, however, takes this concept a step further by accounting for factors we predict will be true in the future. For example, imagine trying to predict whether at least 100 customers will visit your coffee shop each Saturday over the next year. Frequency statistics will determine probability by analyzing data from past Saturday visits. But Bayesian statistics will determine probability by also factoring for a nearby art show that will start in the summer and take place every Saturday afternoon. This allows the Bayesian statistical model to provide a much more accurate figure.

**The goals of inference**

1. Estimate and quantify the uncertainty of an estimate of a population quantity (the proportion of people who will vote for a candidate).
2. Determine whether a population quantity is a benchmark value ("is the treatment effective?").
3. Infer a mechanistic relationship when quantities are measured with noise ("What is the slope for Hooke's law?")
4. Determine the impact of a policy? ("If we reduce pollution levels, will asthma rates decline?")
5. Talk about the probability that something occurs.

## Algorithm:-

**Step 1. Import Dataset:**

```
train_df=pd.read_csv('train.csv')
test_df=pd.read_csv('test.csv')
train_df.shape, test_df.shape

train_df['label']='train'
test_df['label']='test'

combined_data_df=pd.concat([train_df,test_df])
combined_data_df.shape
#The reasons for combining both training and test dataset are:

#To find missing values in both the datasets
#If we need transform/remove any features, we can do it in both datasets at one time
#To convert categorical variable to numerical variable in both datasets
```

**Step 2. Statistical Inference**

```
#Statistical Analysis
combined_data_df.mean()
combined_data_df.median()
combined_data_df.mode()
combined_data_df.std()
combined_data_df.describe()
combined_data_df.min()
combined_data_df.max()
combined_data_df.dtypes
```

**Step 3. Handling Missing Values**

```
#Handle the missing value
combined_data_df.info()
#Data types in data set:

#Categorical = 10
#Numerical = 5
#Target =1
combined_data_df.isnull().sum()
combined_data_df.dropna(subset=['workclass','occupation','native-country'],axis=0,inplace=True)
combined_data_df.isnull().sum()
combined_data_df.dropna(subset=['income_>50K'],axis=0,inplace=True)
combined_data_df.isnull().sum()
```

**Step 4. Data Visualization**

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set_theme(style="darkgrid")

#frequency distribution of work class
plt.figure(figsize=(10,10))
sns.countplot(data= combined_data_df, x = combined_data_df['workclass'])
combined_data_df.drop(combined_data_df.index[combined_data_df['workclass'] == 'Without-pay'], inplace=True)
combined_data_df.shape
plt.figure(figsize=(10,15))
sns.countplot(data= combined_data_df, y = "native-country")
#This graph clearly shows that the given dataset in from US. We can remove other countries since almost of them are US origin.
combined_data_df=combined_data_df[combined_data_df['native-country']=='United-States']
combined_data_df.shape
#Since, now we only have US as the only native country, this feature is useless. We can drop this feature altogether
combined_data_df=combined_data_df.drop(columns='native-country',axis=1)
combined_data_df.shape
#frequency distribution of education class

plt.figure(figsize=(20,10))
sns.countplot(data= combined_data_df, x = "education")
combined_data_df['education'] = combined_data_df['education'].replace(['1st-4th','5th-6th'],'elementary-school')
combined_data_df['education'] = combined_data_df['education'].replace(['7th-8th'],'middle-school')
combined_data_df['education'] = combined_data_df['education'].replace(['9th','10th','11th','12th'],'high-school')
combined_data_df['education'] = combined_data_df['education'].replace(['Doctorate','Bachelors','Some-college','Masters','Prof-school','Assoc-voc','Assoc-acdm'],'postsecondary-education')

plt.figure(figsize=(20,10))
sns.countplot(data= combined_data_df, x = "education")

plt.figure(figsize=(20,10))
sns.countplot(data= combined_data_df, x = "marital-status")

combined_data_df['marital-status'] = combined_data_df['marital-status'].replace(['Divorced','Never-married','Widowed'],'single')
```

```
combined_data_df['marital-status'] = combined_data_df['marital-status'].replace(['Married-civ-
spouse','Separated','Married-spouse-absent','Married-AF-spouse'],'married')
plt.figure(figsize=(20,10))
plt.figure()
sns.countplot(data= combined_data_df, x = "marital-status")

plt.figure(figsize=(20,10))
sns.countplot(data= combined_data_df, y = "occupation")

plt.figure(figsize=(20,10))
sns.countplot(data= combined_data_df, x = "relationship")
```

**Step 5. convert categorical variable to numerical variable**

```
#Split the dataset into categorical and numerical values
#categorical
 cat_columns = [ col for col in list(combined_data_df.columns) if combined_data_df[col].dtype
=='object' and col!= 'label']
cat_columns

#numberical num_columns = [ col for col in list(combined_data_df.columns) if
combined_data_df[col].dtype in ['int64','float64']]
num_columns fig= plt.figure(figsize=(15,15))
corr_matrix = combined_data_df.corr()
sns.heatmap(data=corr_matrix,annot=True)
plt.show()
combined_data_df.drop(columns='fnlwgt',inplace=True)
combined_data_df.shape
#Converting categorical variables to numerical ( Dummy variables )
#get dummies
features_df = pd.get_dummies(data=combined_data_df, columns=cat_columns)
features_df.shape
features_df.columns
#split your data
train_df = features_df[features_df['label'] == 'train']
test_df = features_df[features_df['label'] == 'test']
# Drop your labels
train_df = train_df.drop('label', axis=1)
test_df = test_df.drop(columns=['label','income_>50K'], axis=1)
train_df.shape, test_df.shape
train_df.columns
train_df.isnull().sum().sum()
```

**Input:** train.csv, test.csv

**Output:** Performed statistical analysis on the income prediction dataset and also converted categorical data into numerical data.

**Conclusion:-**

● Handled the missing value, by dropping them from the dataset<br>
● From the data visualization, combined/categorized the features<br>
● Using dummy variable, converted categorical variable to numerical variable to create better model.