Descriptive Statistics

Measures of Central Tendency

In statistics, a central tendency (or measure of central tendency) is a central or typical value for a probability distribution. It may also be called a center or location of the distribution. The most common measures of central tendency are the arithmetic mean, the median, and the mode

In [1]:
```
import pandas as pd
```

In [5]:
```
df=pd.read_csv(r'C:\Users\acer\Desktop\Sem 1\data science\DataSet\stats.csv')
df
```

Out[5]:

	Name	Salary	Country
0	Dan	40000	USA
1	Elizabeth	32000	Brazil
2	Jon	45000	Italy
3	Maria	54000	USA
4	Mark	72000	USA
5	Bill	62000	Brazil
6	Jess	92000	Italy
7	Julia	55000	USA
8	Jeff	35000	Italy
9	Ben	48000	Brazil

In [6]:
```
mean1=df['Salary'].mean()
mean1
```

Out[6]: 53500.0

In [7]:
```
sum1=df['Salary'].sum()
sum1
```

Out[7]: 535000

In [8]:
```
max1=df['Salary'].max()
max1
```

Out[8]: 92000

In [9]:
```
min1=df['Salary'].min()
min1
```

Out[9]: 32000

In [10]:
```
count1=df['Salary'].count()
count1
```

Out[10]: 10

In [11]:
```
median=df['Salary'].median()
median
```

Out[11]:
```
51000.0
```

In [12]:
```
mode1=df['Salary'].mode()
mode1
```

Out[12]:
```
0    32000
1    35000
2    40000
3    45000
4    48000
5    54000
6    55000
7    62000
8    72000
9    92000
dtype: int64
```

In [13]:
```
countrywise_sum=df.groupby(['Country'])['Salary'].sum()
countrywise_sum
```

Out[13]:
```
Country
Brazil    142000
Italy     172000
USA       221000
Name: Salary, dtype: int64
```

In [14]:
```
countrywise_count=df.groupby(['Country']).count()
countrywise_count
```

Out[14]:

Country	Name	Salary
Brazil	3	3
Italy	3	3
USA	4	4

Variance

In probability theory and statistics, variance is the expectation of the squared deviation of a random variable from its population mean or sample mean. Variance is a measure of dispersion, meaning it is a measure of how far a set of numbers is spread out from their average value.

In [15]:
```
var1=df['Salary'].var()
var1
```

Out[15]:
```
332055555.5555556
```

Standard Deviation

In statistics, the standard deviation is a measure of the amount of variation or dispersion of a set of values. A low standard deviation indicates that the values tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the values are spread out over a wider range

In [16]:
```
std1=df['Salary'].std()
std1
```

Out[16]:
```
18222.391598128816
```

Measure of Symmetry

Skewness

In probability theory and statistics, skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive, zero, negative, or undefined

In [17]:
```
skew1=df.skew(axis=0, skipna=True)
skew1
```

Out[17]:
```
Salary    1.021551
dtype: float64
```

Covariance and Correlation

Covariance

In probability theory and statistics, covariance is a measure of the joint variability of two random variables. If the greater values of one variable mainly correspond with the greater values of the other variable, and the same holds for the lesser values (that is, the variables tend to show similar behavior), the covariance is positive. In the opposite case, when the greater values of one variable mainly correspond to the lesser values of the other, (that is, the variables tend to show opposite behavior), the covariance is negative. The sign of the covariance therefore shows the tendency in the linear relationship between the variables.

Correlation

In statistics, correlation or dependence is any statistical relationship, whether causal or not, between two random variables or bivariate data. In the broadest sense correlation is any statistical association, though it actually refers to the degree to which a pair of variables are linearly related. Familiar examples of dependent phenomena include the correlation between the height of parents and their offspring, and the correlation between the price of a good and the quantity the consumers are willing to purchase, as it is depicted in the so-called demand curve

In [19]:
```
bw=pd.read_csv(r'C:\Users\acer\Desktop\Sem 1\data science\DataSet\BirthWeight.csv')
bw.head()
```

Out[19]:

	Infant ID	Gestational Age (Weeks)	Birth Weight (Grams)
0	1	34.7	1895
1	2	36.0	2030
2	3	29.3	1440
3	4	40.1	2835
4	5	35.7	3090

In [20]:
```
bw.set_index('Infant ID', inplace=True)
bw.head()
```

Out[20]:

Infant ID	Gestational Age (Weeks)	Birth Weight (Grams)
1	34.7	1895
2	36.0	2030
3	29.3	1440
4	40.1	2835
5	35.7	3090

In [21]:
```
bw.cov()
```

```
bw.cov()
```

Out[21]:

	Gestational Age (Weeks)	Birth Weight (Grams)
Gestational Age (Weeks)	9.963824	1798.025
Birth Weight (Grams)	1798.025000	485478.750

In [22]:
```
bw.corr(method="pearson")
```

Out[22]:

	Gestational Age (Weeks)	Birth Weight (Grams)
Gestational Age (Weeks)	1.000000	0.817519
Birth Weight (Grams)	0.817519	1.000000

In [24]:
```
#Covariance indicates that there is correlation exists between two
#Correlation coefficient of 0.818 indicates the relationship between two is positive and strong
```

In [23]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import skew
from scipy.stats import kurtosis
```

In [25]:
```
pd.set_option("display.max_columns",None)
pd.options.display.float_format = "{:,.2f}".format
```

Format : A data frame with 53940 rows and 10 variables

Description : A dataset containing the prices and other attributes of almost 54,000 diamonds.

The variables are as follows:

price: price in US dollars (326−− 18,823) carat: weight of the diamond (0.2--5.01) cut: quality of the cut (Fair, Good, Very Good, Premium, Ideal) colour: diamond colour, from J (worst) to D (best) clarity: a measurement of how clear the diamond is (IF (best), VVS1, VVS2,VS1, VS2, SI1, SI2, I1 (worst)) popularity: popularity of this specs (Good, Fair, Poor) x: length in mm (0--10.74) y: width in mm (0--58.9) z: depth in mm (0--31.8) depth: total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43--79) table: width of top of diamond relative to widest point (43--95)

In [27]:
```
xls = pd.read_csv(r'C:\Users\acer\Desktop\Sem 1\data science\DataSet\diamonds.csv')
xls.head()
```

Out[27]:

	id	carat	cut	color	clarity	depth	table	price	x	y	z
0	1	0.23	Ideal	E	SI2	61.50	55.00	326	3.95	3.98	2.43
1	2	0.21	Premium	E	SI1	59.80	61.00	326	3.89	3.84	2.31
2	3	0.23	Good	E	VS1	56.90	65.00	327	4.05	4.07	2.31
3	4	0.29	Premium	I	VS2	62.40	58.00	334	4.20	4.23	2.63
4	5	0.31	Good	J	SI2	63.30	58.00	335	4.34	4.35	2.75

In [32]:
```
des_df = xls.drop(['id'],axis = 1)
for col in des_df:
    if des_df[col].dtype == 'object':
        des_df = des_df.drop([col], axis = 1)
des_r = des_df.describe()
des_r = des_r.rename(index={'50%':'median/50%'})
des_r
```

Out[32]:

	carat	depth	table	price	x	y	z
count	53,940.00	53,940.00	53,940.00	53,940.00	53,940.00	53,940.00	53,940.00
mean	0.80	61.75	57.46	3,932.80	5.73	5.73	3.54
std	0.47	1.43	2.23	3,989.44	1.12	1.14	0.71
min	0.20	43.00	43.00	326.00	0.00	0.00	0.00
25%	0.40	61.00	56.00	950.00	4.71	4.72	2.91

median/50%	0.70	61.80	57.00	2,401.00	5.70	5.71	3.53
75%	1.04	62.50	59.00	5,324.25	6.54	6.54	4.04
max	5.01	79.00	95.00	18,823.00	10.74	58.90	31.80

In [33]:

```
var_r = des_df.var()
varlist = []
for col in des_df.columns:
    if des_df[col].dtype == 'object':
        continue
    varlist.append(round(des_df[col],5))

df = pd.DataFrame([varlist],columns=des_r.columns, index=['var'])
mct = des_r.append(df)
mct
```

Out[33]:

	carat	depth	table	price	x	y	z
count	53,940.00	53,940.00	53,940.00	53,940.00	53,940.00	53,940.00	53,940.00
mean	0.80	61.75	57.46	3,932.80	5.73	5.73	3.54
std	0.47	1.43	2.23	3,989.44	1.12	1.14	0.71
min	0.20	43.00	43.00	326.00	0.00	0.00	0.00
25%	0.40	61.00	56.00	950.00	4.71	4.72	2.91
median/50%	0.70	61.80	57.00	2,401.00	5.70	5.71	3.53
75%	1.04	62.50	59.00	5,324.25	6.54	6.54	4.04
max	5.01	79.00	95.00	18,823.00	10.74	58.90	31.80
var	0 0.23 1 0.21 2 0.23 3 ...	0 61.50 1 59.80 2 56.90 3 ...	0 55.00 1 61.00 2 65.00 3 ...	0 326 1 326 2 327 3 ...	0 3.95 1 3.89 2 4.05 3 ...	0 3.98 1 3.84 2 4.07 3 ...	0 2.43 1 2.31 2 2.31 3 ...

In [ ]: