Data Reading

```
#task 1
from google.colab import drive
drive.mount('/content/drive')

data="/content/drive/MyDrive/dataset/medical_insurance.csv"

#dataset: https://www.kaggle.com/datasets/harishkumardatalab/medical-insurance-price-prediction
```

```
Mounted at /content/drive
```

```
#task 2
import pandas as pd

dataframe = pd.read_csv(data)
```

+ Code + Text

```
#task 3
dataframe.head(5)
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

```
#task4
dataframe.tail(5)
```

	age	sex	bmi	children	smoker	region	charges
2767	47	female	45.320	1	no	southeast	8569.86180
2768	21	female	34.600	0	no	southwest	2020.17700
2769	19	male	26.030	1	yes	northwest	16450.89470
2770	23	male	18.715	0	no	northwest	21595.38229
2771	54	male	31.600	0	no	southwest	9850.43200

Adding Headers

```
#task 5

headers = ["AAA", "Sex", "BMI", "Children", "Smoker", "Region", "Charges"]
dataframe.columns = headers

dataframe.head(10)
```

	AAA	Sex	BMI	Children	Smoker	Region	Charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
5	31	female	25.740	0	no	southeast	3756.62160
6	46	female	33.440	1	no	southeast	8240.58960
7	37	female	27.740	3	no	northwest	7281.50560
8	37	male	29.830	2	no	northeast	6406.41070
9	60	female	25.840	0	no	northwest	28923.13692

```
#task 6
dataframe=dataframe.rename(columns={'AAA':'Age'})
dataframe
```

	Age	Sex	BMI	Children	Smoker	Region	Charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
2767	47	female	45.320	1	no	southeast	8569.86180
2768	21	female	34.600	0	no	southwest	2020.17700
2769	19	male	26.030	1	yes	northwest	16450.89470
2770	23	male	18.715	0	no	northwest	21595.38229
2771	54	male	31.600	0	no	southwest	9850.43200

2772 rows × 7 columns

Basic Insight of Dataset

```
#task 7
print(dataframe.dtypes)
```

```
Age           int64
Sex          object
BMI         float64
Children      int64
Smoker       object
Region       object
Charges     float64
dtype: object
```

```
#task 8 (float to int of column: Charges)
dataframe['Charges'] = dataframe['Charges'].astype(int)
print(dataframe.dtypes)
dataframe.head(3)
```

```
Age           int64
Sex          object
BMI         float64
Children      int64
Smoker       object
Region       object
Charges       int64
dtype: object
```

	Age	Sex	BMI	Children	Smoker	Region	Charges
0	19	female	27.90	0	yes	southwest	16884
1	18	male	33.77	1	no	southeast	1725
2	28	male	33.00	3	no	southeast	4449

```
#task 9(description)
print(f"description:")
dataframe.describe()
```

```
description:
```

	Age	BMI	Children	Charges
count	2772.000000	2772.000000	2772.000000	2772.000000
mean	39.109668	30.701349	1.101732	13260.875180
std	14.081459	6.129449	1.214806	12151.768709
min	18.000000	15.960000	0.000000	1121.000000
25%	26.000000	26.220000	0.000000	4687.000000
50%	39.000000	30.447500	1.000000	9332.500000
75%	51.000000	34.770000	2.000000	16577.000000
max	64.000000	53.130000	5.000000	63770.000000

```
#task 9(information)
print(f"information of this dataset:")
dataframe.info()
```

```
information of this dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2772 entries, 0 to 2771
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Age       2772 non-null   int64  
 1   Sex       2772 non-null   object 
 2   BMI       2772 non-null   float64
 3   Children  2772 non-null   int64  
 4   Smoker    2772 non-null   object 
 5   Region    2772 non-null   object 
 6   Charges   2772 non-null   int64  
dtypes: float64(1), int64(3), object(3)
memory usage: 151.7+ KB
```

```
#task 10 (Selections of data using ‘loc’: selecting rows with specified BMI value)
df=dataframe.loc[(dataframe.BMI>22.5) & (dataframe.BMI<25.5)]
display(df)
```

	Age	Sex	BMI	Children	Smoker	Region	Charges
3	33	male	22.705	0	no	northwest	21984
15	19	male	24.600	1	no	southwest	1837
17	23	male	23.845	0	no	northeast	2395
26	63	female	23.085	0	no	northeast	14451
48	60	female	24.530	0	no	southeast	12629
...	...	...	...	...	...	...	...
2740	28	male	24.300	5	no	southwest	5615
2745	19	female	23.400	2	no	southwest	2913
2756	39	female	24.225	5	no	northwest	8965
2759	18	male	23.210	0	no	southeast	1121
2766	18	male	23.320	1	no	southeast	1711

354 rows × 7 columns

```
#task 10 (Selections of data using ‘iloc’: selecting indices 1,5,7,12,50,67,1199)
df1=dataframe.iloc[[1,5,7,12,50,67,1199]]
display(df1)
```

	Age	Sex	BMI	Children	Smoker	Region	Charges
1	18	male	33.770	1	no	southeast	1725
5	31	female	25.740	0	no	southeast	3756
7	37	female	27.740	3	no	northwest	7281
12	23	male	34.400	0	no	southwest	1826
50	18	female	35.625	0	no	northeast	2211
67	40	male	26.315	1	no	northwest	6389
1199	31	female	25.800	2	no	southwest	4934

```
'''TASK 11: Write a function [2 marks] that calculates the logarithm of the mean value of a column
and ‘apply’ this to your columns [1 mark], returning both the mean value and its logarithm.
Display your results.
'''
import pandas as pd
import numpy as np

def log_of_mean(data):
  if pd.api.types.is_numeric_dtype(data):
    mean=data.mean()
    log_mean=np.log(mean)
    return mean, log_mean

result_df=dataframe.apply(log_of_mean)
result_df['Measure']=['Mean', 'Log(Mean)']
print(result_df)
```

```
              Age   Sex        BMI  Children Smoker Region       Charges    Measure
     0  39.109668  None  30.701349  1.101732   None   None  13260.875180       Mean
     1   3.666370  None   3.424307  0.096883   None   None      9.492573  Log(Mean)
```

Correlation and Covariance

```
# TASK 12: Determining the correlation between two of the numeric attributes AAA and BMI

correlation=dataframe["Age"].corr(dataframe["BMI"])
print("Correlation between Age and BMI: ", correlation )
```

```
     Correlation between Age and BMI:  0.11304845107996202
```

```
# TASK 13: Determine the covariance between two of the numeric attributes

covv=dataframe["Age"].cov(dataframe["BMI"])
print("Covariance between Age and BMI: ", covv )
```

 # TASK 14: Difference between correlation and covariance

As we know correlation offers a standardized view of the linear association between variables, while covariance provides insights into the direction and relative magnitude of change considering the data units. The key differences is being discussed between Correlation and Covariance below:

1. Unit:

- Covariance: Measured in the units of the product of the two variables.
- Correlation: Unitless, ranging from -1 to +1.List item

2. Scale Sensitivity:

- Covariance: Sensitive to the scale of the data. Doubling one variable will double the covariance.
- Correlation: Not affected by the scale of the data.

3. Interpretation:

- Covariance: Indicates the direction and magnitude of the linear relationship. In positive covariance variables tend to move in the same direction while variables tend to move in opposite directions in negative covariance. Covariance of zero indicate noo linear relationship between two variable.
- Correlation: Represents the strength and direction of the linear relationship. Values closer to 1 means strong positive linear relationship Values closer to -1 indicates strong negative linear relationship and values closer to 0 indicates weak or no linear relationship.

```
# TASK 15   : Creating a new column with random values

low_bound,high_bound=1,5
dataframe["randValue"] = np.random.randint(low_bound, high_bound, size=dataframe.shape[0])
display(dataframe)
```

	Age	Sex	BMI	Children	Smoker	Region	Charges	randValue
0	19	female	27.900	0	yes	southwest	16884	3
1	18	male	33.770	1	no	southeast	1725	2
2	28	male	33.000	3	no	southeast	4449	2
3	33	male	22.705	0	no	northwest	21984	3
4	32	male	28.880	0	no	northwest	3866	4
...	...	...	...	...	...	...	...	...
2767	47	female	45.320	1	no	southeast	8569	2
2768	21	female	34.600	0	no	southwest	2020	2
2769	19	male	26.030	1	yes	northwest	16450	4
2770	23	male	18.715	0	no	northwest	21595	1
2771	54	male	31.600	0	no	southwest	9850	4

2772 rows × 8 columns

```
# TASK 16: Correlation and covariance between this new column and your previous two columns

corr1=dataframe["Age"].corr(dataframe["randValue"])
print("Correlation between Age and randValue: ", corr1 )
```