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

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

```
ls drive/MyDrive/'Colab Notebooks'/bmi.csv
```

```
'drive/MyDrive/Colab Notebooks/bmi.csv'
```

```
ls drive/MyDrive/'Colab Notebooks'/'income(1) (1).csv'
```

```
'drive/MyDrive/Colab Notebooks/income(1) (1).csv'
```

```
import pandas as pd
import numpy as np
from scipy import stats
```

```
df=pd.read_csv("drive/MyDrive/Colab Notebooks/bmi.csv")
df
```

```
     Gender  Height  Weight  Index
0      Male     174      96      4
1      Male     189      87      2
2    Female     185     110      4
3    Female     195     104      3
4      Male     149      61      3
...     ...     ...     ...    ...
495  Female     150     153      5
496  Female     184     121      4
497  Female     141     136      5
498    Male     150      95      5
499    Male     173     131      5

500 rows × 4 columns
```

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.head()
```

```
   Gender  Height  Weight  Index
0    Male     174      96      4
1    Male     189      87      2
2  Female     185     110      4
3  Female     195     104      3
4    Male     149      61      3
```

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.dropna()
```

```
     Gender  Height  Weight  Index
0      Male     174      96      4
1      Male     189      87      2
2    Female     185     110      4
3    Female     195     104      3
4      Male     149      61      3
...     ...     ...     ...    ...
495  Female     150     153      5
496  Female     184     121      4
497  Female     141     136      5
498    Male     150      95      5
499    Male     173     131      5
500 rows × 4 columns
```

```
max_vals = np.max(np.abs(df[['Height','Weight']]))
max_vals
max_vals
```

```
199
```

```
df1=pd.read_csv("drive/MyDrive/Colab Notebooks/bmi.csv")
df1
```

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

Next steps: Generate code with df1 View recommended plots New interactive sheet

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
df1[['Height','Weight']] = sc.fit_transform(df1[['Height','Weight']])
df.head(10)
```

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
5	Male	189	104	3
6	Male	147	92	5
7	Male	154	111	5
8	Male	174	90	3
9	Female	169	103	4

Next steps: Generate code with df View recommended plots New interactive sheet

```
from sklearn.preprocessing import MinMaxScaler
```

```
Scaler=MinMaxScaler()
df[['Height','Weight']]=Scaler.fit_transform(df[['Height','Weight']])
```

```
df.head(0)
```

```
   Gender  Height  Weight  Index
```

Next steps: Generate code with df View recommended plots New interactive sheet

```
df2=pd.read_csv("drive/MyDrive/Colab Notebooks/bmi.csv")
df2
```

```
      Gender  Height  Weight  Index
  0     Male     174      96      4
  1     Male     189      87      2
  2   Female     185     110      4
  3   Female     195     104      3
  4     Male     149      61      3
...      ...     ...     ...    ...
495   Female     150     153      5
496   Female     184     121      4
497   Female     141     136      5
498     Male     150      95      5
499     Male     173     131      5
500 rows × 4 columns
```

Next steps: Generate code with df2 View recommended plots New interactive sheet

```
from sklearn.preprocessing import Normalizer
Scaler=Normalizer()
df2[['Height','Weight']]=Scaler.fit_transform(df2[['Height','Weight']])
df2
```

	Gender	Height	Weight	Index
0	Male	0.875578	0.483077	4
1	Male	0.908381	0.418144	2
2	Female	0.859536	0.511075	4
3	Female	0.882353	0.470588	3
4	Male	0.925448	0.378875	3
...	...	...	...	...
495	Female	0.700071	0.714073	5
496	Female	0.835527	0.549450	4
497	Female	0.719753	0.694230	5
498	Male	0.844819	0.535052	5
499	Male	0.797227	0.603680	5

500 rows × 4 columns

Next steps: Generate code with df2 View recommended plots New interactive sheet

```
df3=pd.read_csv("drive/MyDrive/Colab Notebooks/bmi.csv")
df3
```

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

Next steps: Generate code with df3 View recommended plots New interactive sheet

```
from sklearn.preprocessing import MaxAbsScaler
Scaler=MaxAbsScaler()
df3[['Height','Weight']]=Scaler.fit_transform(df3[['Height','Weight']])
df3
```

	Gender	Height	Weight	Index
0	Male	0.874372	0.60000	4
1	Male	0.949749	0.54375	2
2	Female	0.929648	0.68750	4
3	Female	0.979899	0.65000	3
4	Male	0.748744	0.38125	3
...	...	...	...	...
495	Female	0.753769	0.95625	5
496	Female	0.924623	0.75625	4
497	Female	0.708543	0.85000	5
498	Male	0.753769	0.59375	5
499	Male	0.869347	0.81875	5

500 rows × 4 columns

Next steps: Generate code with df3 View recommended plots New interactive sheet

```
df4=pd.read_csv("drive/MyDrive/Colab Notebooks/bmi.csv")
```

```
from sklearn.preprocessing import RobustScaler
Scaler=RobustScaler()
df4[['Height','Weight']]=Scaler.fit_transform(df4[['Height','Weight']])
df4.head()
```

	Gender	Height	Weight	Index
0	Male	0.125000	-0.178571	4
1	Male	0.660714	-0.339286	2
2	Female	0.517857	0.071429	4
3	Female	0.875000	-0.035714	3
4	Male	-0.767857	-0.803571	3

Next steps: Generate code with df4 View recommended plots New interactive sheet

```
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import RFE
from sklearn.linear_model import RidgeCV,LassoCV,Ridge,Lasso
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import mutual_info_classif
from sklearn.feature_selection import mutual_info_regression
from sklearn.feature_selection import chi2
```

```
df=pd.read_csv('drive/MyDrive/Colab Notebooks/income(1) (1).csv')
df.columns
```

```
Index(['age', 'JobType', 'EdType', 'maritalstatus', 'occupation',
       'relationship', 'race', 'gender', 'capitalgain', 'capitalloss',
       'hoursperweek', 'nativecountry', 'SalStat'],
      dtype='object')
```

```
df1=df.drop(["Name","sex","Ticket","cabin","embarked"],axis=1)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-24-957512aa0a29> in <cell line: 1>()
----> 1 df1=df.drop(["Name","sex","Ticket","cabin","embarked"],axis=1)

                                  3 frames

/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in drop(self, labels, errors)
   7068         if mask.any():
   7069             if errors != "ignore":
-> 7070                 raise KeyError(f"{labels[mask].tolist()} not found in axis")
   7071             indexer = indexer[~mask]
   7072         return self.delete(indexer)

KeyError: "['Name', 'sex', 'Ticket', 'cabin', 'embarked'] not found in axis"
```

Next steps: Explain error

```
df1.columns
```

```
Index(['Gender', 'Height', 'Weight', 'Index'], dtype='object')
```

```
import pandas as pd
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
```

```
data=pd.read_csv('drive/MyDrive/Colab Notebooks/bmi.csv')
```

```
data=data.dropna()
```

```
df.columns
```

```
Index(['age', 'JobType', 'EdType', 'maritalstatus', 'occupation',
       'relationship', 'race', 'gender', 'capitalgain', 'capitalloss',
       'hoursperweek', 'nativecountry', 'SalStat'],
      dtype='object')
```

```
df
```

	age	JobType	EdType	maritalstatus	occupation	relationship	race	gender	capitalgain	capitalloss	hoursperweek	nativecountry	SalStat
0	45	Private	HS-grad	Divorced	Adm-clerical	Not-in-family	White	Female	0	0	28	United-States	less than or equal to 50,000
1	24	Federal-gov	HS-grad	Never-married	Armed-Forces	Own-child	White	Male	0	0	40	United-States	less than or equal to 50,000
2	44	Private	Some-college	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	40	United-States	greater than 50,000
3	27	Private	9th	Never-married	Craft-repair	Other-relative	White	Male	0	0	40	Mexico	less than or equal to 50,000
4	20	Private	Some-college	Never-married	Sales	Not-in-family	White	Male	0	0	35	United-States	less than or equal to 50,000
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Next Generate code with df View recommended plots New interactive sheet

```
import pandas as pd
import numpy as np
from scipy.stats import chi2_contingency
import seaborn as sns
tips=sns.load_dataset('tips')
tips.head()
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4