```
import pandas as pd
import numpy as np
```

```
df=pd.read_csv("/content/bike_buyers_cleaned.csv")
df
```

	Unnamed: 0	ID	Marital Status	Gender	Income	Children	Education	Occupation	Home Owner	Cars	Commute Distance	Region	Age	Purchased Bike
0	0	12496	Married	Female	40000.0	1.0	Bachelors	Skilled Manual	Yes	0.0	0-1 Miles	Europe	42.0	No
1	1	24107	Married	Male	30000.0	3.0	Partial College	Clerical	Yes	1.0	0-1 Miles	Europe	43.0	No
2	2	14177	Married	Male	80000.0	5.0	Partial College	Professional	No	2.0	2-5 Miles	Europe	60.0	No
3	3	24381	Single	Male	70000.0	0.0	Bachelors	Professional	Yes	1.0	5-10 Miles	Pacific	41.0	Yes
4	4	25597	Single	Male	30000.0	0.0	Bachelors	Clerical	No	0.0	0-1 Miles	Europe	36.0	Yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	995	23731	Married	Male	60000.0	2.0	High School	Professional	Yes	2.0	2-5 Miles	North America	54.0	Yes
996	996	28672	Single	Male	70000.0	4.0	Graduate Degree	Professional	Yes	0.0	2-5 Miles	North America	35.0	Yes
997	997	11809	Married	Male	60000.0	2.0	Bachelors	Skilled Manual	Yes	0.0	0-1 Miles	North America	38.0	Yes
998	998	19664	Single	Male	100000.0	3.0	Bachelors	Management	No	3.0	1-2 Miles	North America	38.0	No
999	999	12121	Single	Male	60000.0	3.0	High School	Professional	Yes	2.0	10+ Miles	North America	53.0	Yes

1000 rows × 14 columns

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

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Unnamed: 0        1000 non-null   int64  
 1   ID                1000 non-null   int64  
 2   Marital Status    1000 non-null   object 
 3   Gender            1000 non-null   object 
 4   Income            994 non-null    float64
 5   Children          992 non-null    float64
 6   Education         1000 non-null   object 
 7   Occupation        1000 non-null   object 
 8   Home Owner        1000 non-null   object 
 9   Cars              991 non-null    float64
 10  Commute Distance  1000 non-null   object 
 11  Region            1000 non-null   object 
 12  Age               992 non-null    float64
 13  Purchased Bike    1000 non-null   object 
dtypes: float64(4), int64(2), object(8)
memory usage: 109.5+ KB
```

```
df.describe()
```

	Unnamed: 0	ID	Income	Children	Cars	Age
count	1000.000000	1000.000000	994.000000	992.000000	991.000000	992.000000
mean	499.500000	19965.992000	56267.605634	1.910282	1.455096	44.181452
std	288.819436	5347.333948	31067.817462	1.626910	1.121755	11.362007
min	0.000000	11000.000000	10000.000000	0.000000	0.000000	25.000000
25%	249.750000	15290.750000	30000.000000	0.000000	1.000000	35.000000
50%	499.500000	19744.000000	60000.000000	2.000000	1.000000	43.000000
75%	749.250000	24470.750000	70000.000000	3.000000	2.000000	52.000000
max	999.000000	29447.000000	170000.000000	5.000000	4.000000	89.000000

```
df.head()
```

	Unnamed: 0	ID	Marital Status	Gender	Income	Children	Education	Occupation	Home Owner	Cars	Commute Distance	Region	Age	Purchased Bike
0	0	12496	Married	Female	40000.0	1.0	Bachelors	Skilled Manual	Yes	0.0	0-1 Miles	Europe	42.0	No
1	1	24107	Married	Male	30000.0	3.0	Partial College	Clerical	Yes	1.0	0-1 Miles	Europe	43.0	No
2	2	14177	Married	Male	80000.0	5.0	Partial College	Professional	No	2.0	2-5 Miles	Europe	60.0	No
3	3	24381	Single	Male	70000.0	0.0	Bachelors	Professional	Yes	1.0	5-10 Miles	Pacific	41.0	Yes
4	4	25597	Single	Male	30000.0	0.0	Bachelors	Clerical	No	0.0	0-1 Miles	Europe	36.0	Yes

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

```
df.tail()
```

	Unnamed: 0	ID	Marital Status	Gender	Income	Children	Education	Occupation	Home Owner	Cars	Commute Distance	Region	Age	Purchased Bike
995	995	23731	Married	Male	60000.0	2.0	High School	Professional	Yes	2.0	2-5 Miles	North America	54.0	Yes
996	996	28672	Single	Male	70000.0	4.0	Graduate Degree	Professional	Yes	0.0	2-5 Miles	North America	35.0	Yes
997	997	11809	Married	Male	60000.0	2.0	Bachelors	Skilled Manual	Yes	0.0	0-1 Miles	North America	38.0	Yes
998	998	19664	Single	Male	100000.0	3.0	Bachelors	Management	No	3.0	1-2 Miles	North America	38.0	No
999	999	12121	Single	Male	60000.0	3.0	High School	Professional	Yes	2.0	10+ Miles	North America	53.0	Yes

```
df.isnull().mean()
```

	0
Unnamed: 0	0.000
ID	0.000
Marital Status	0.000
Gender	0.000
Income	0.006
Children	0.008
Education	0.000
Occupation	0.000
Home Owner	0.000
Cars	0.009
Commute Distance	0.000
Region	0.000
Age	0.008
Purchased Bike	0.000

dtype: float64

NUMERICAL DATA FILLING

Filling Data by Mean.

```
# Percentage of null values in Income.
df["Income"].isnull().mean()
```

```
0.006
```

```
# Filling Data
df["Income"].fillna(df["Income"].mean(),inplace=True)
```

```
<ipython-input-30-0da5aa769eed>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignm
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting valu

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].me


  df["Income"].fillna(df["Income"].mean(),inplace=True)
```

```
df["Income"].isnull().mean()
```

```
0.0
```

```
df["Income"].to_string()
```

```
'0       40000.000000\n1       30000.000000\n2       80000.000000\n3       70000.000000\n4       30000.000000\n5       10000.000000\n6
160000.000000\n7       40000.000000\n8       20000.000000\n9       56267.605634\n10      30000.000000\n11      90000.000000\n12     170
000.000000\n13      40000.000000\n14      60000.000000\n15      10000.000000\n16      30000.000000\n17      30000.000000\n18      4000
0.000000\n19      20000.000000\n20      40000.000000\n21      80000.000000\n22      40000.000000\n23      80000.000000\n24      40000.0
00000\n25      30000.000000\n26      30000.000000\n27     100000.000000\n28      70000.000000\n29      20000.000000\n30      20000.0000
00\n31      10000.000000\n32      20000.000000\n33      80000.000000\n34      90000.000000\n35      10000.000000\n36      10000.000000
\n37      30000.000000\n38      20000.000000\n39      10000.000000\n40      30000.000000\n41      40000.000000\n42      10000.000000\n4
3     170000.000000\n44      20000.000000\n45      20...'
```

Filling Data by Median.

```
# Check the isnull values in Children.
df["Children"].isnull().mean()
```

```
0.008
```

```
df["Children"].fillna(df["Children"].median(),inplace=True)
```

```
n-input-34-ca3d5b94e71d>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment usi
avior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values alwa

mple, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(va


hildren"].fillna(df["Children"].median(),inplace=True)
```

```
df["Children"].isnull().mean()
```

```
0.0
```

```
df["Children"].to_string()
```

```
'0      1.0\n1      3.0\n2      5.0\n3      0.0\n4      0.0\n5      2.0\n6      2.0\n7      1.0\n8      2.0\n9      2.0\n10     3.0\n11
0.0\n12     5.0\n13     2.0\n14     1.0\n15     2.0\n16     3.0\n17     1.0\n18     2.0\n19     2.0\n20     0.0\n21     0.0\n22     2.0
\n23     5.0\n24     2.0\n25     1.0\n26     0.0\n27     0.0\n28     5.0\n29     0.0\n30     2.0\n31     0.0\n32     0.0\n33     2.0\n3
4     5.0\n35     5.0\n36     2.0\n37     0.0\n38     0.0\n39     4.0\n40     2.0\n41     2.0\n42     1.0\n43     4.0\n44     3.0\n45
1.0\n46     1.0\n47     2.0\n48     2.0\n49     0.0\n50     0.0\n51     0.0\n52     1.0\n53     4.0\n54     0.0\n55     4.0\n56     0.0
\n57     4.0\n58     1.0\n59     2.0\n60     1.0\n61     2.0\n62     2.0\n63     4.0\n64     1.0\n65     2.0\n66     0.0\n67     0.0\n6
8     4.0\n69     0.0\n70     0.0\n71     0.0\n72     3.0\n73     0.0\n74     3.0\n75     4.0\n76     0.0\n77     0.0\n78     2.0\n79
2.0\n80     4.0\n81     4.0\n82     0.0\n83  …'
```

Filling Data by Mode.

```
# percentage of isnull vales in Cars.
df["Cars"].isnull().mean()
```

```
0.009
```

```
# Filling the Cars Data.
df["Cars"].fillna(df["Cars"].mode()[0],inplace=True)
```

```
<ipython-input-39-44217a1b17d7>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignm
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting valu

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].me


  df["Cars"].fillna(df["Cars"].mode()[0],inplace=True)
```

```
df["Cars"].isnull().mean()
```

```
0.0
```