In [ ]:

```
================================================== EDA STEPS==================================================
1. Read the data
2. Separate Categorical columns and Numerical columns
3. Data quick checks
    shape,columns,dtypes
4. Null value analysis
    A. Check if any null values are present
    B. Fill the null values with median or KNNImputer for numerical columns
    C. Fill the null values with mode for Categorical columns
5. Do some data preprocessing
     If any columns are corrupted
    Ex- Numerical values in categorical columns
    ex- Categorical values in Numerical columns
6. Drop the id columns
    which means a data has more unique lables
    Drop the single value columns
7. Categorical column analysis
    a. Frequency tables
    b. Bar charts
    c. pie charts
8. Numerical columns analysis
    a. Histogram :
    b. Distribution
    c. Box plot
9. Outliers analysis
    Impute the outliers with median
10. Find the correlation between numerical columns
    heat maps
11. Convert Categorical to numerical
    a. LabelEncoder
    b. One hot Encoder
12. Scale the data
    a. Z standardization
    b. Normalization

By the time of 12 steps, we achieve 3 things
    1. Cleaned data
    2. Data in the form of complete Numerical
    3. We have some understanding the data
```

```
13. We will try to select the important features for ML model
    a. PCA: Principle Component Analysis (We will covered it in ML time)


Finally we achieve 3 dataset

1. Till step-11
           without scaling but data in numerical format
2. Till step-12
           with scaling data in numerical format

3. Till step-13
           PCA data
```

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
telecom_df=pd.read_csv(r"C:\Users\suman\Desktop\Desktop\DATASCIENCE & AI\DATA FILES\telecom_churn_data.csv")
telecom_df
```

Out[3]:

	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched
0	2015	100198	409-8743	Female	36	62	no	no	148.35
1	2015	100643	340-5930	Female	39	149	no	no	294.45
2	2015	100756	372-3750	Female	65	126	no	no	87.30
3	2015	101595	331-4902	Female	24	131	no	yes	321.30
4	2015	101653	351-8398	Female	40	191	no	no	243.00
...	...	...	...	...	...	...	...	...	...
1995	2015	997132	385-7387	Female	54	75	no	yes	182.25
1996	2015	998086	383-9255	Male	45	127	no	no	273.45
1997	2015	998474	353-2080	NaN	53	94	no	no	128.85
1998	2015	998934	359-7788	Male	40	94	no	no	178.05
1999	2015	999961	414-1496	Male	37	73	no	no	326.70

2000 rows × 16 columns

In [5]:
```
cat=telecom_df.select_dtypes(include='object').columns
num=telecom_df.select_dtypes(exclude='object').columns
cat,num
```

Out[5]:
```
(Index(['phone_no', 'gender', 'multi_screen', 'mail_subscribed'], dtype='object'),
 Index(['year', 'customer_id', 'age', 'no_of_days_subscribed',
        'weekly_mins_watched', 'minimum_daily_mins', 'maximum_daily_mins',
        'weekly_max_night_mins', 'videos_watched', 'maximum_days_inactive',
        'customer_support_calls', 'churn'],
       dtype='object'))
```

In [7]:
```
telecom_df['gender'].isnull().sum()
```

Out[7]:
```
24
```

In [9]:
```
gender_mode=telecom_df['gender'].mode()
telecom_df['gender']=telecom_df['gender'].fillna(gender_mode.values[0])
telecom_df
```

Out[9]:

	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched
0	2015	100198	409-8743	Female	36	62	no	no	148.35
1	2015	100643	340-5930	Female	39	149	no	no	294.45
2	2015	100756	372-3750	Female	65	126	no	no	87.30
3	2015	101595	331-4902	Female	24	131	no	yes	321.30
4	2015	101653	351-8398	Female	40	191	no	no	243.00
...	...	...	...	...	...	...	...	...	...
1995	2015	997132	385-7387	Female	54	75	no	yes	182.25
1996	2015	998086	383-9255	Male	45	127	no	no	273.45
1997	2015	998474	353-2080	Male	53	94	no	no	128.85
1998	2015	998934	359-7788	Male	40	94	no	no	178.05
1999	2015	999961	414-1496	Male	37	73	no	no	326.70

2000 rows × 16 columns

In [11]:
```
for i in cat[1:]:
    modes=telecom_df[i].mode()
    telecom_df[i]=telecom_df[i].fillna(modes.values[0])


telecom_df
```

Out[11]:

	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched
0	2015	100198	409-8743	Female	36	62	no	no	148.35
1	2015	100643	340-5930	Female	39	149	no	no	294.45
2	2015	100756	372-3750	Female	65	126	no	no	87.30
3	2015	101595	331-4902	Female	24	131	no	yes	321.30
4	2015	101653	351-8398	Female	40	191	no	no	243.00
...	...	...	...	...	...	...	...	...	...
1995	2015	997132	385-7387	Female	54	75	no	yes	182.25
1996	2015	998086	383-9255	Male	45	127	no	no	273.45
1997	2015	998474	353-2080	Male	53	94	no	no	128.85
1998	2015	998934	359-7788	Male	40	94	no	no	178.05
1999	2015	999961	414-1496	Male	37	73	no	no	326.70

2000 rows × 16 columns

In [13]:
```
telecom_df['age'].isnull().sum()
```

Out[13]:
```
0
```

In [15]:
```
age_median=round(telecom_df['age'].median())
age_median
```

Out[15]:
```
37
```

In [17]:
```
age_median=round(telecom_df['age'].median())
telecom_df['age']=telecom_df['age'].fillna(age_median)
telecom_df
```

Out[17]:

	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched
0	2015	100198	409-8743	Female	36	62	no	no	148.35
1	2015	100643	340-5930	Female	39	149	no	no	294.45
2	2015	100756	372-3750	Female	65	126	no	no	87.30
3	2015	101595	331-4902	Female	24	131	no	yes	321.30
4	2015	101653	351-8398	Female	40	191	no	no	243.00
...	...	...	...	...	...	...	...	...	...
1995	2015	997132	385-7387	Female	54	75	no	yes	182.25
1996	2015	998086	383-9255	Male	45	127	no	no	273.45
1997	2015	998474	353-2080	Male	53	94	no	no	128.85
1998	2015	998934	359-7788	Male	40	94	no	no	178.05
1999	2015	999961	414-1496	Male	37	73	no	no	326.70

2000 rows × 16 columns

In [19]:
```
telecom_df.drop(['year','customer_id','phone_no'],axis=1,inplace=True)
telecom_df
```

Out[19]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched	minimum_daily_mins	maximum
0	Female	36	62	no	no	148.35	12.2	
1	Female	39	149	no	no	294.45	7.7	
2	Female	65	126	no	no	87.30	11.9	
3	Female	24	131	no	yes	321.30	9.5	
4	Female	40	191	no	no	243.00	10.9	
...	...	...	...	...	...	...	...	
1995	Female	54	75	no	yes	182.25	11.3	
1996	Male	45	127	no	no	273.45	9.3	
1997	Male	53	94	no	no	128.85	15.6	
1998	Male	40	94	no	no	178.05	10.4	
1999	Male	37	73	no	no	326.70	10.3	

2000 rows × 13 columns

In [21]:
```
for i in num[2:]:
    medians=round(telecom_df[i].median())
    telecom_df[i]=round(telecom_df[i].fillna(medians))

telecom_df
```

Out[21]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched	minimum_daily_mins	maximum
0	Female	36	62	no	no	148.0	12.0	
1	Female	39	149	no	no	294.0	8.0	
2	Female	65	126	no	no	87.0	12.0	
3	Female	24	131	no	yes	321.0	10.0	
4	Female	40	191	no	no	243.0	11.0	
...	...	...	...	...	...	...	...	
1995	Female	54	75	no	yes	182.0	11.0	
1996	Male	45	127	no	no	273.0	9.0	
1997	Male	53	94	no	no	129.0	16.0	
1998	Male	40	94	no	no	178.0	10.0	
1999	Male	37	73	no	no	327.0	10.0	

2000 rows × 13 columns

In [23]:
```
cat=telecom_df.select_dtypes(include='object').columns
num=telecom_df.select_dtypes(exclude='object').columns
cat,num
```

Out[23]:
```
(Index(['gender', 'multi_screen', 'mail_subscribed'], dtype='object'),
 Index(['age', 'no_of_days_subscribed', 'weekly_mins_watched',
        'minimum_daily_mins', 'maximum_daily_mins', 'weekly_max_night_mins',
        'videos_watched', 'maximum_days_inactive', 'customer_support_calls',
        'churn'],
       dtype='object'))
```

FREQUENCY TABLE

In [25]:
```
keys=telecom_df['gender'].value_counts().keys()
values=telecom_df['gender'].value_counts().values
```

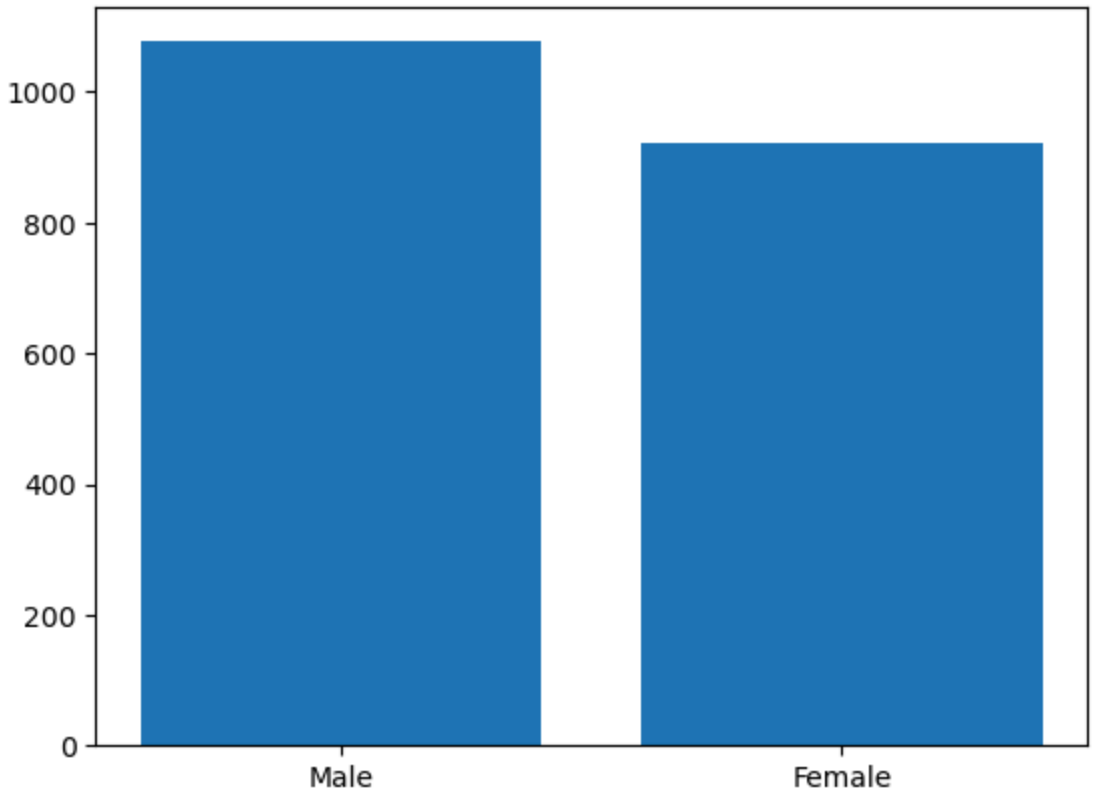
```
dff=pd.DataFrame(zip(keys,values))
dff.to_csv('gender_table.csv')
```

In [27]:
```
import os
folder='CHURN DATASET'
path=os.getcwd()
new_dir=os.path.join(path,folder)

try:
    os.makedirs(new_dir)
except Exception as e:
    print(e)


for i in cat:
    keys1=telecom_df[i].value_counts().keys()
    values1=telecom_df[i].value_counts().values
    col=['TYPES','NO OF TYPES']
    name=f'{i}_table.csv'
    new_path=os.path.join(new_dir,name)
    df1=pd.DataFrame(zip(keys1,values1),columns=col)
    df1.to_csv(new_path)
```

```
[WinError 183] Cannot create a file when that file already exists: 'C:\\Users\\suman\\OneDrive\\Documents\\NARESH IT \\EDA\\CHURN DATASET'
```

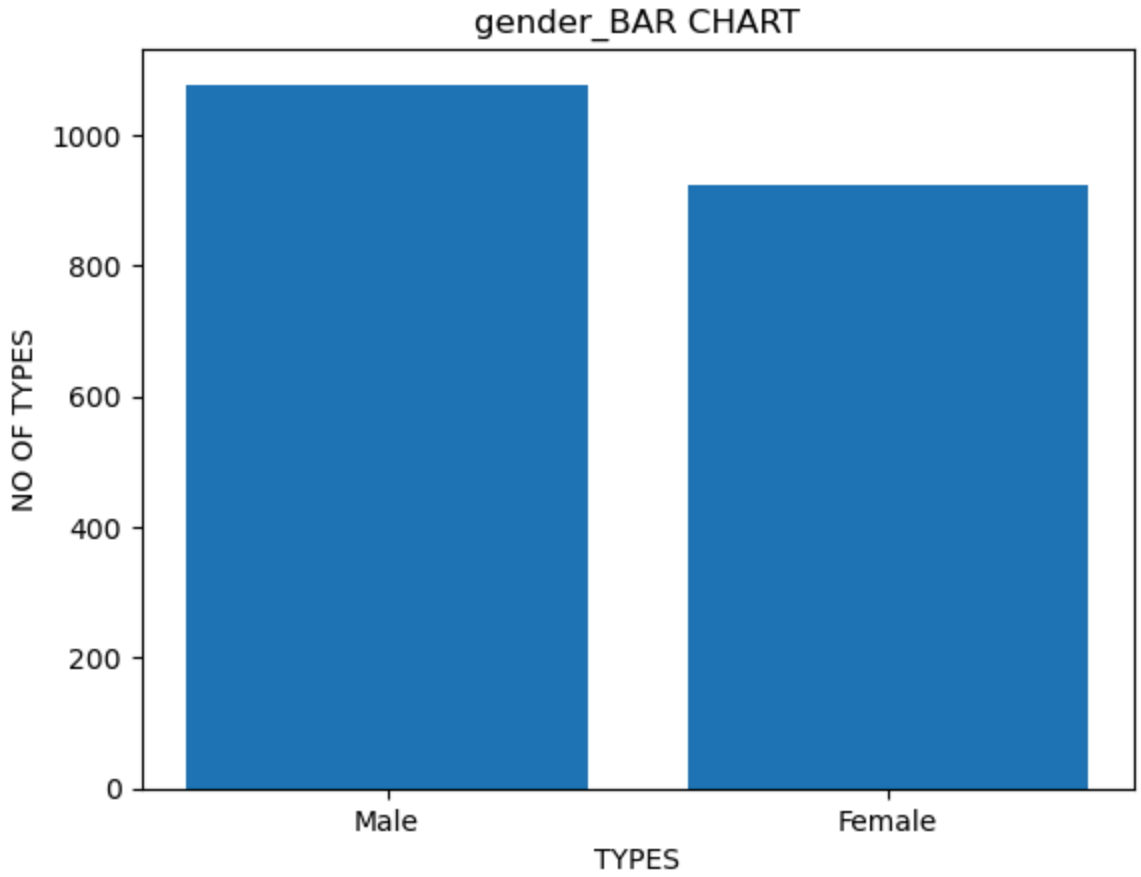
CATEGORICAL COLUMN

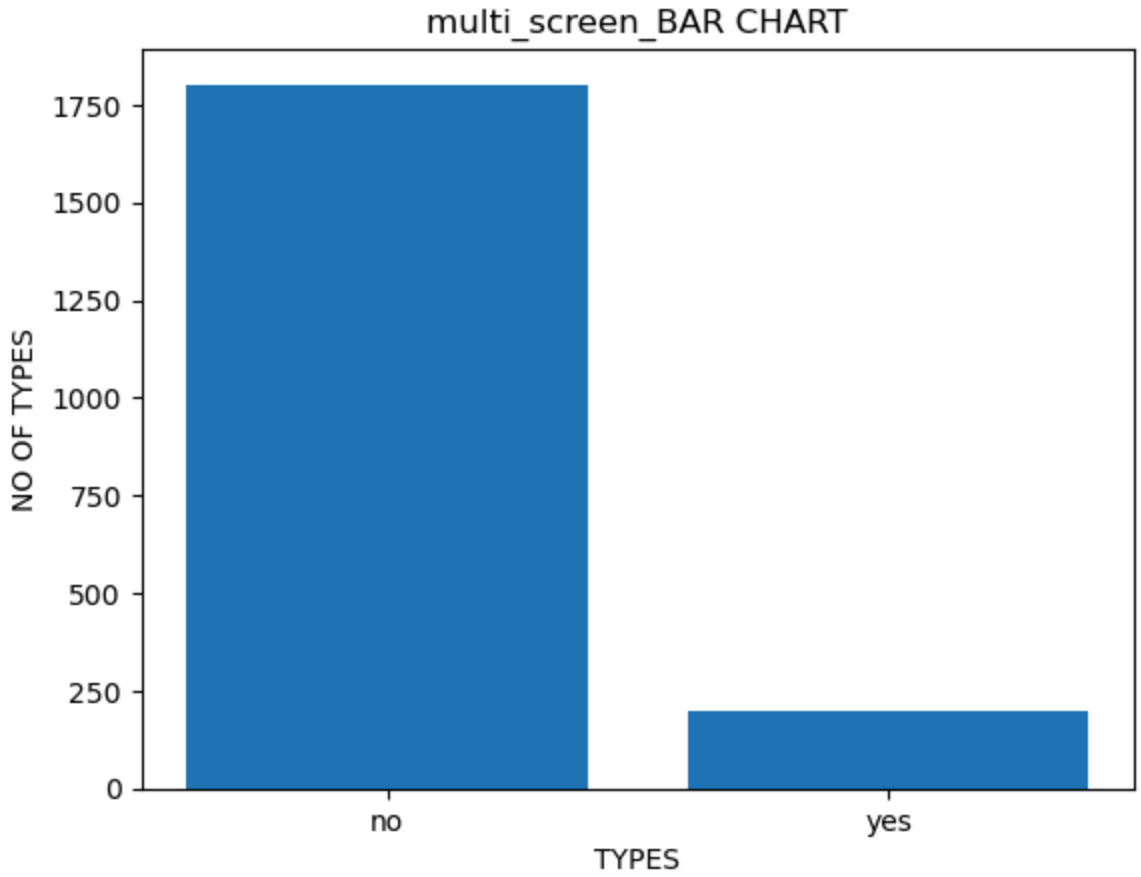
In [277…
```
keys1=telecom_df['gender'].value_counts().keys()
values1=telecom_df['gender'].value_counts().values
plt.bar(keys1,values1)
plt.show()
```

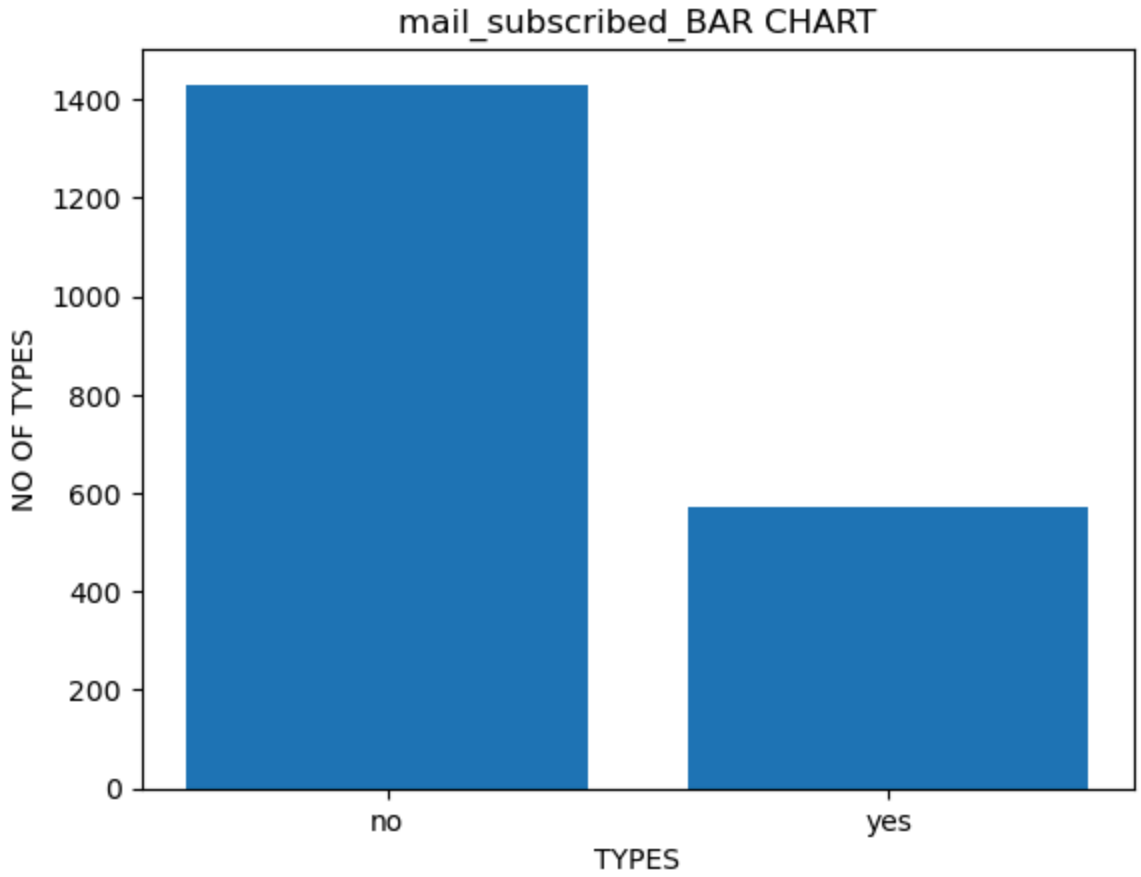


In [279…

```
for i in cat:
    keys1=telecom_df[i].value_counts().keys()
    values1=telecom_df[i].value_counts().values
    plt.bar(keys1,values1)
    plt.title(f'{i}_BAR CHART')
    plt.xlabel('TYPES')
    plt.ylabel('NO OF TYPES')
    name1=f'{i}_bar_chart.jpg'
    new_path1=os.path.join(new_dir,name1)
    plt.savefig(new_path1)
    plt.show()
```

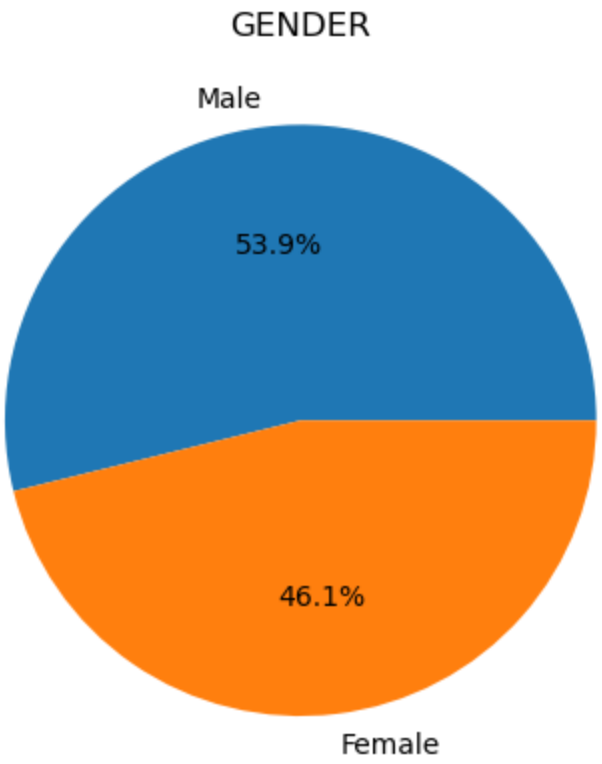

gender_BAR CHART
1000
800
600
400
200
0
NO OF TYPES
Male
Female
TYPES


multi_screen_BAR CHART
NO OF TYPES
1750
1500
1250
1000
750
500
250
0
no
yes
TYPES



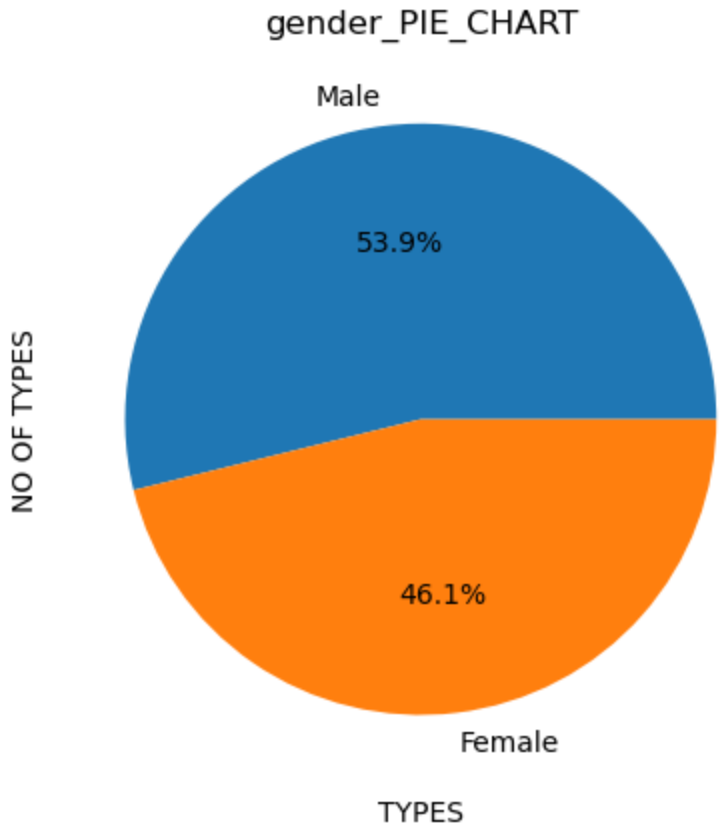
In [281…

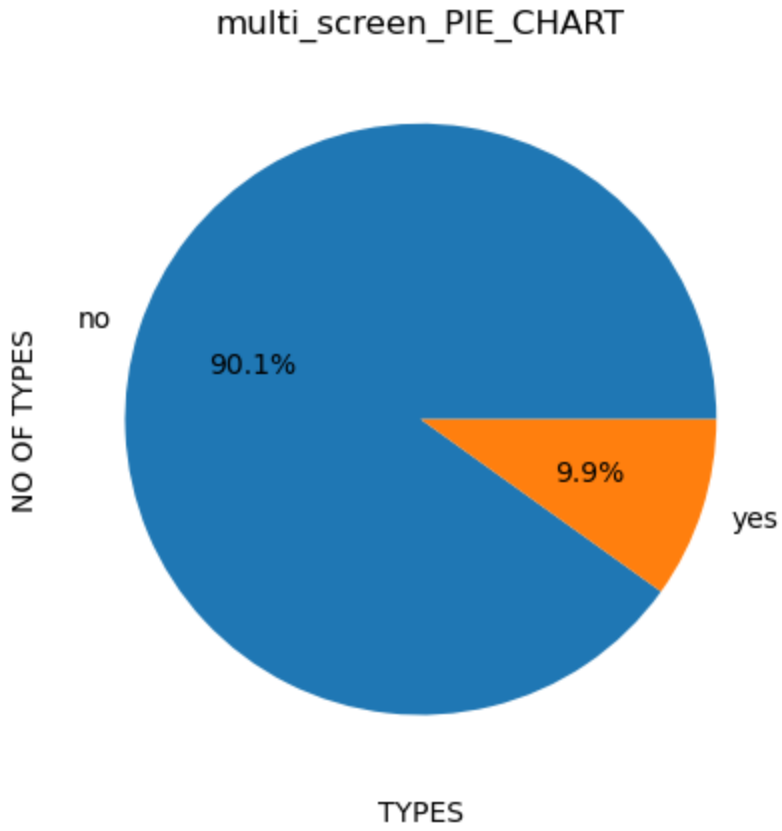
```
keys1=telecom_df['gender'].value_counts().keys()
values1=telecom_df['gender'].value_counts().values
plt.pie(x=values1,labels=keys1,autopct='%0.1f%%')
plt.title('GENDER')
plt.show()
```

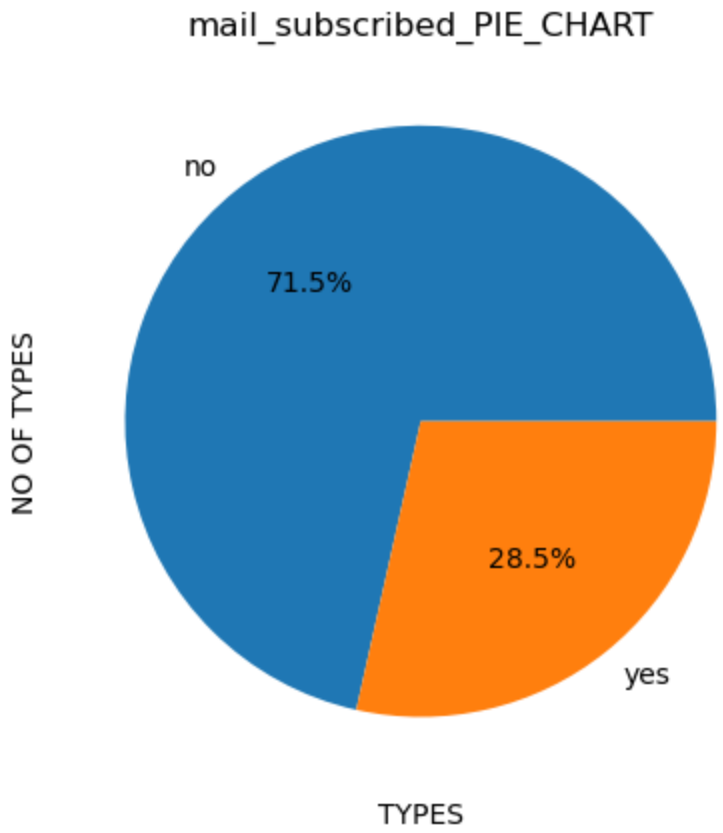


In [283…

```
for i in cat:
    keys1=telecom_df[i].value_counts().keys()
    values1=telecom_df[i].value_counts().values
    plt.pie(values1,labels=keys1,autopct="%0.1f%%")
    plt.title(f'{i}_PIE_CHART')
    plt.xlabel('TYPES')
    plt.ylabel('NO OF TYPES')
    name1=f'{i}_bar_chart.jpg'
    new_path1=os.path.join(new_dir,name1)
    plt.savefig(new_path1)
    plt.show()
```


gender_PIE_CHART
Male
53.9%
46.1%
Female
NO OF TYPES
TYPES

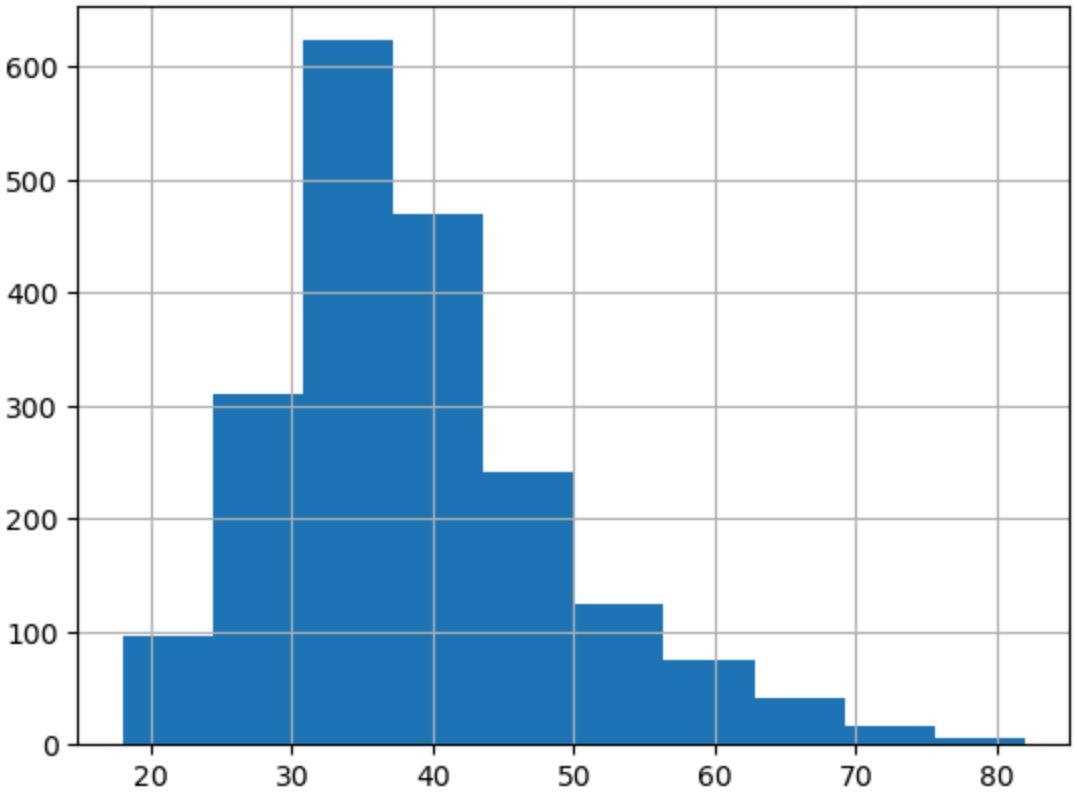

multi_screen_PIE_CHART
NO OF TYPES
no
90.1%
9.9%
yes
TYPES



NUMERICAL COLUMNS ANALYSIS

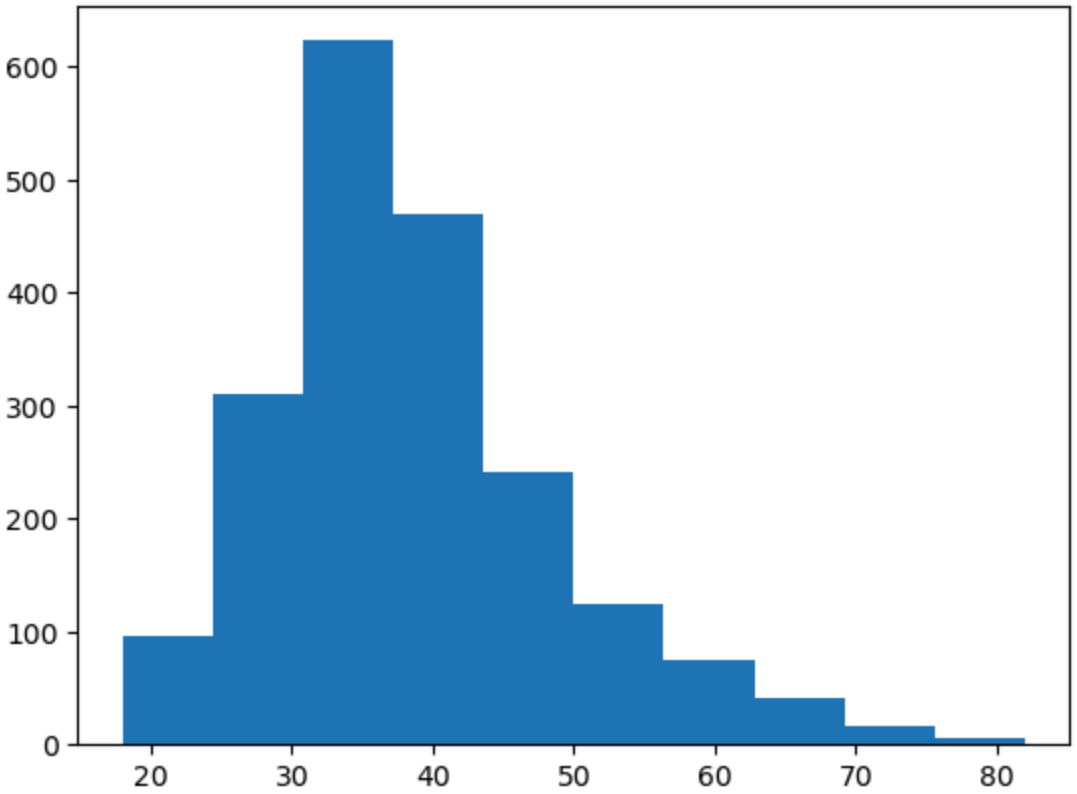
In [168…
```
telecom_df['age'].hist()
```

Out[168… <Axes: >



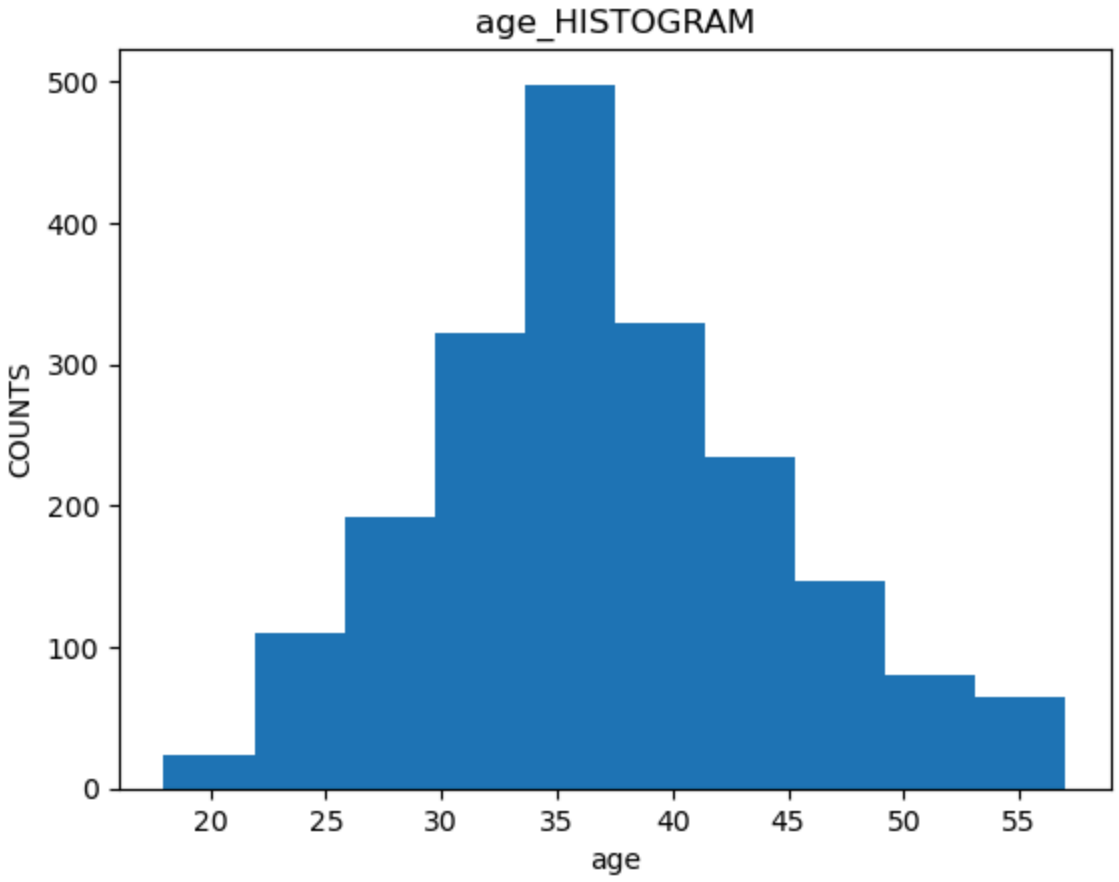
In [188…
```
plt.hist(telecom_df['age'])
```

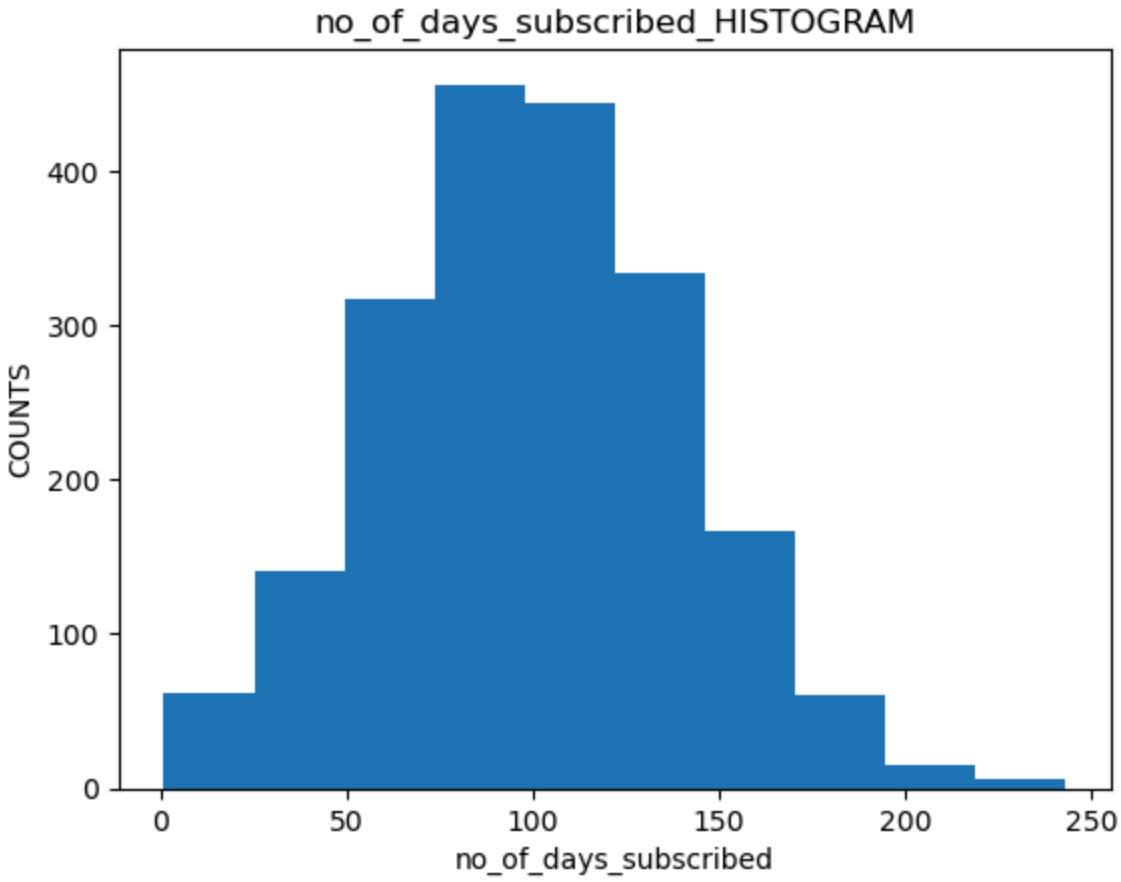
Out[188…
```
(array([ 96., 310., 623., 469., 241., 124.,  74.,  41.,  17.,   5.]),
 array([18. , 24.4, 30.8, 37.2, 43.6, 50. , 56.4, 62.8, 69.2, 75.6, 82. ]),
 <BarContainer object of 10 artists>)
```

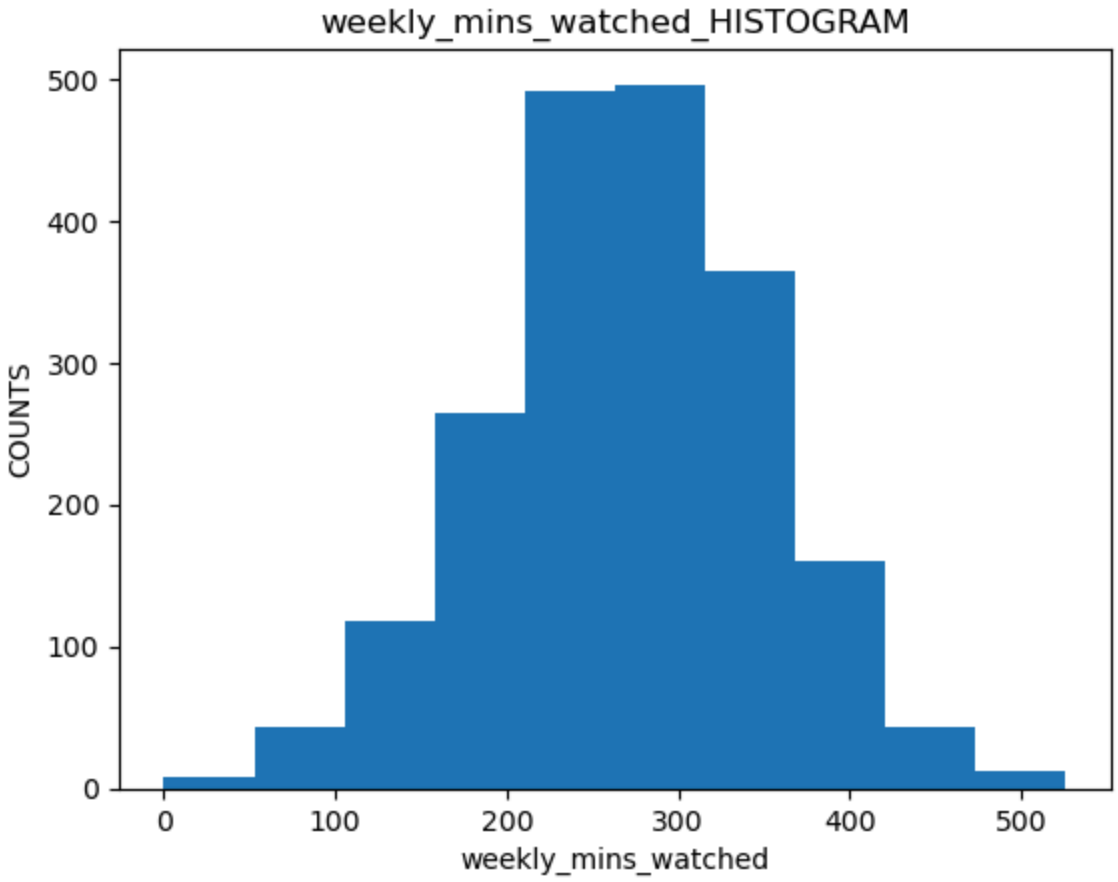


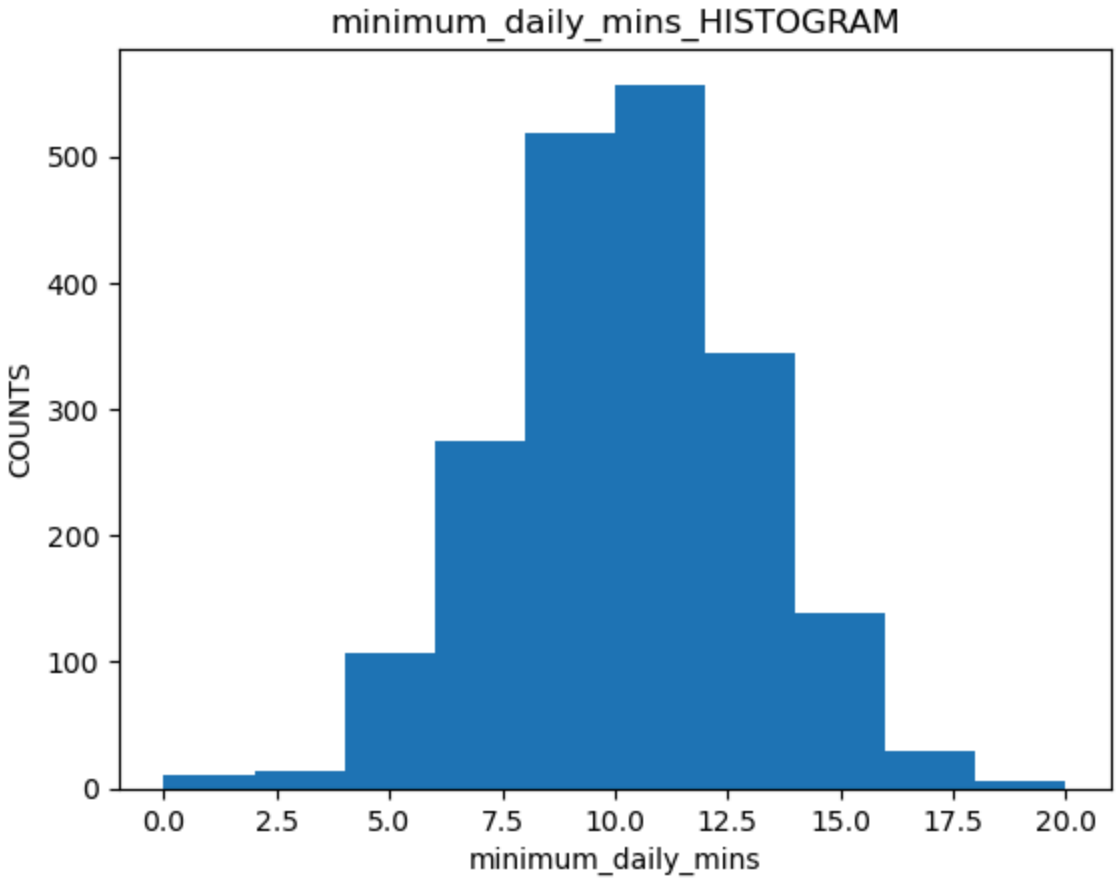
In [285…

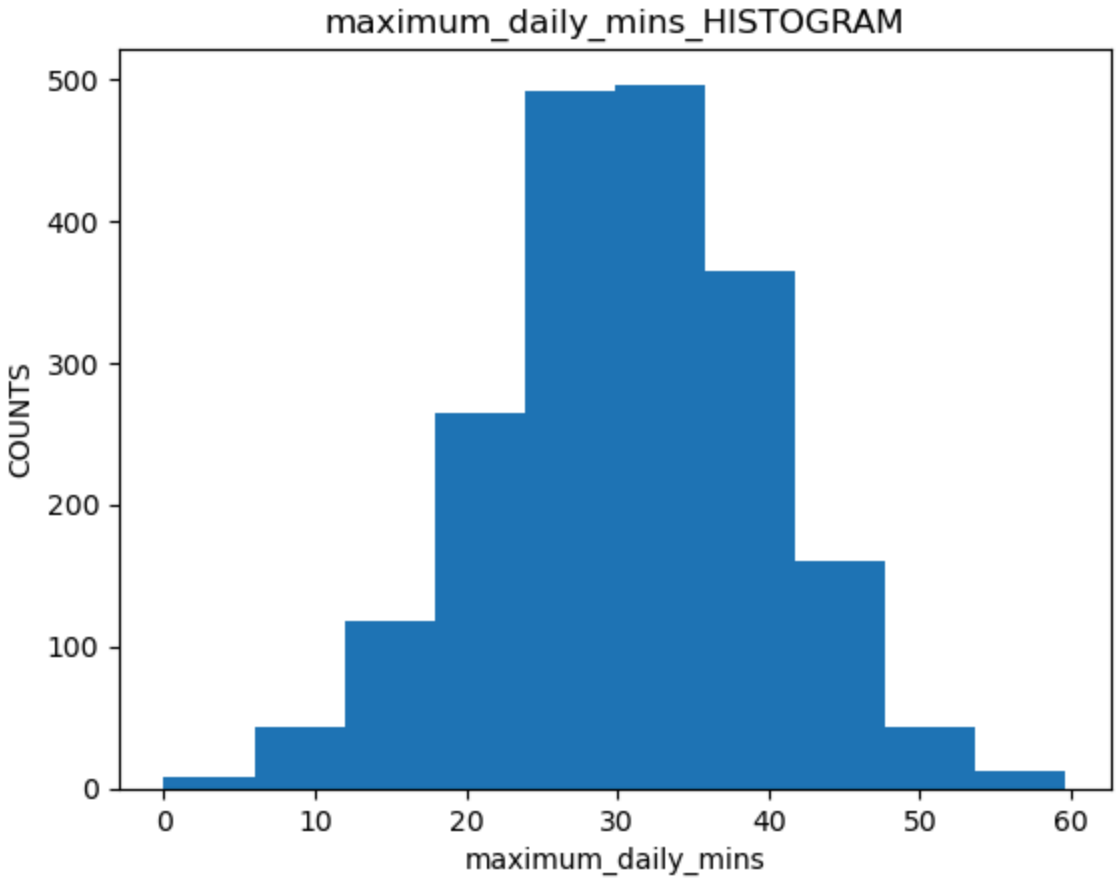
```
for i in num:
    plt.hist(telecom_df[i])
    plt.title(f'{i}_HISTOGRAM')
    plt.xlabel(f'{i}')
    plt.ylabel('COUNTS')
    name2=f'{i}_histogram.jpg'
    new_path2=os.path.join(new_dir,name2)
    plt.savefig(new_path2)
    plt.show()
```

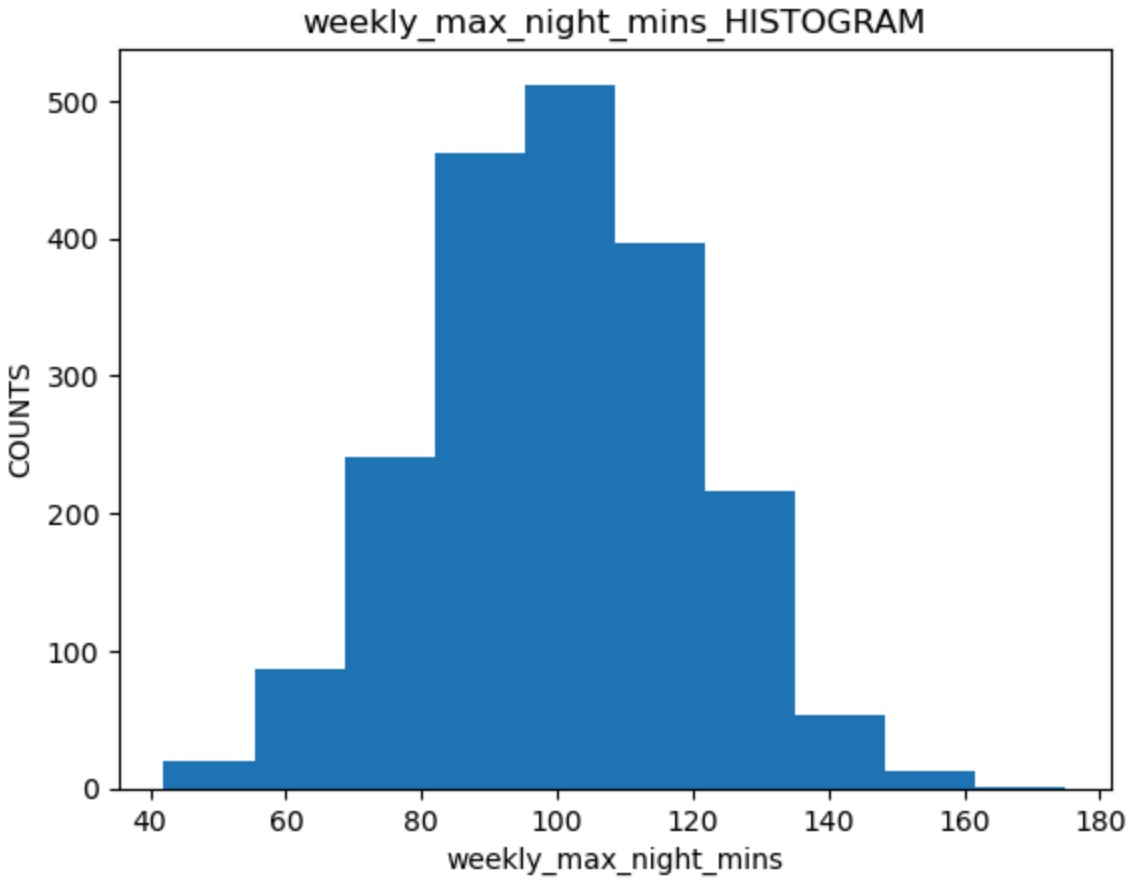

age_HISTOGRAM
COUNTS
0
100
200
300
400
500
20
25
30
35
40
45
50
55
age

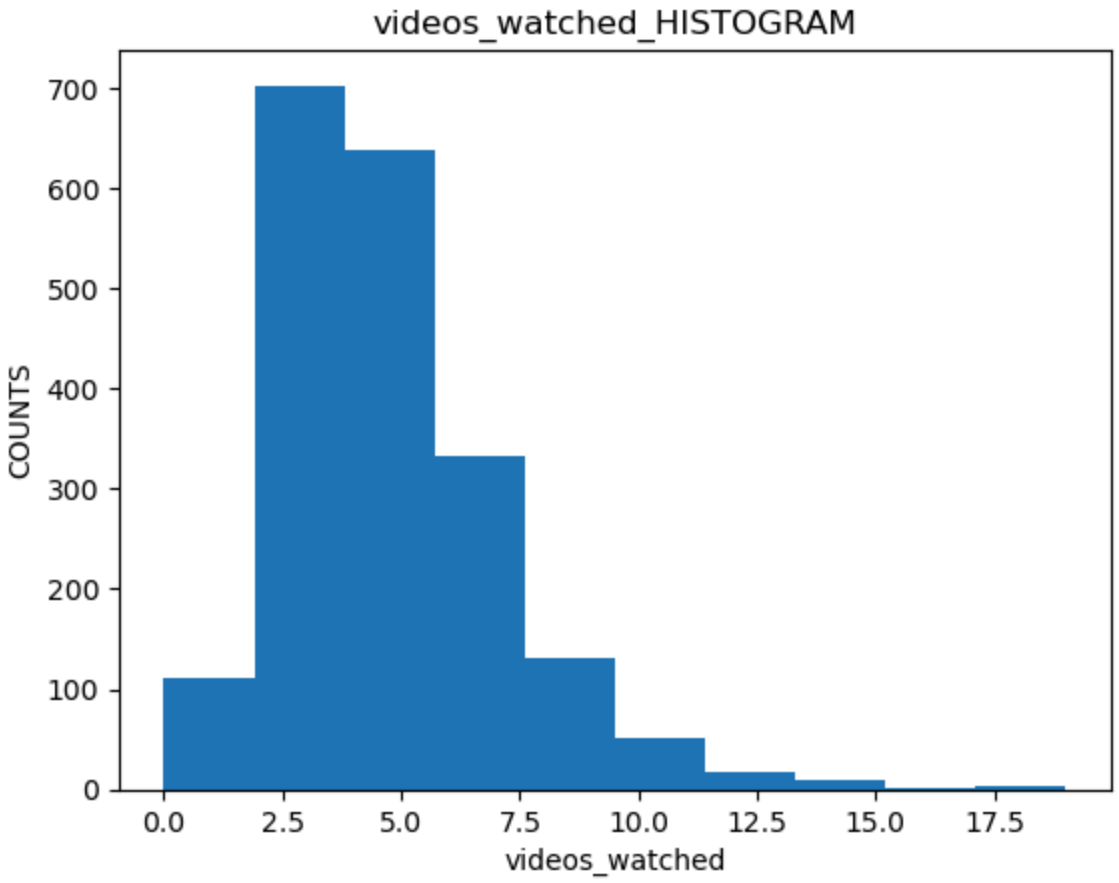

no_of_days_subscribed_HISTOGRAM
COUNTS
0
100
200
300
400
0
50
100
150
200
250
no_of_days_subscribed

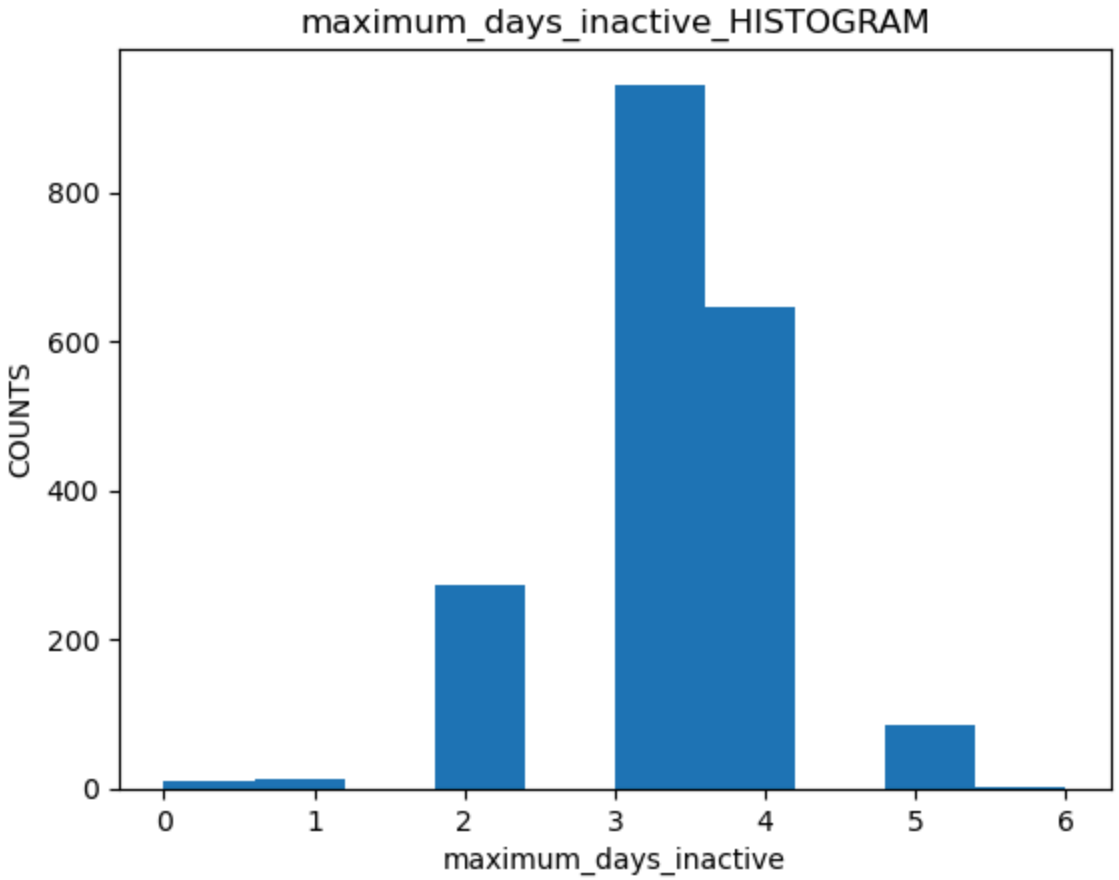

weekly_mins_watched_HISTOGRAM
COUNTS
0
100
200
300
400
500
weekly_mins_watched

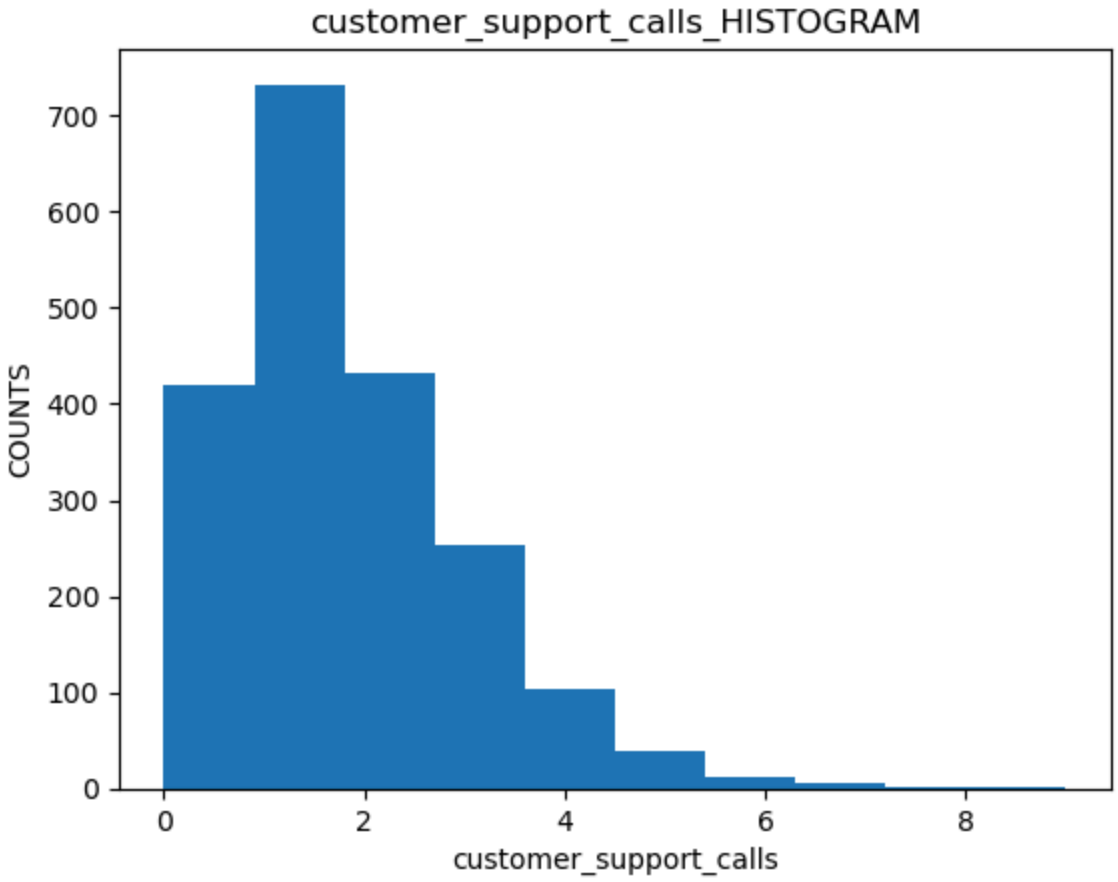

minimum_daily_mins_HISTOGRAM
COUNTS
0
100
200
300
400
500
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
minimum_daily_mins

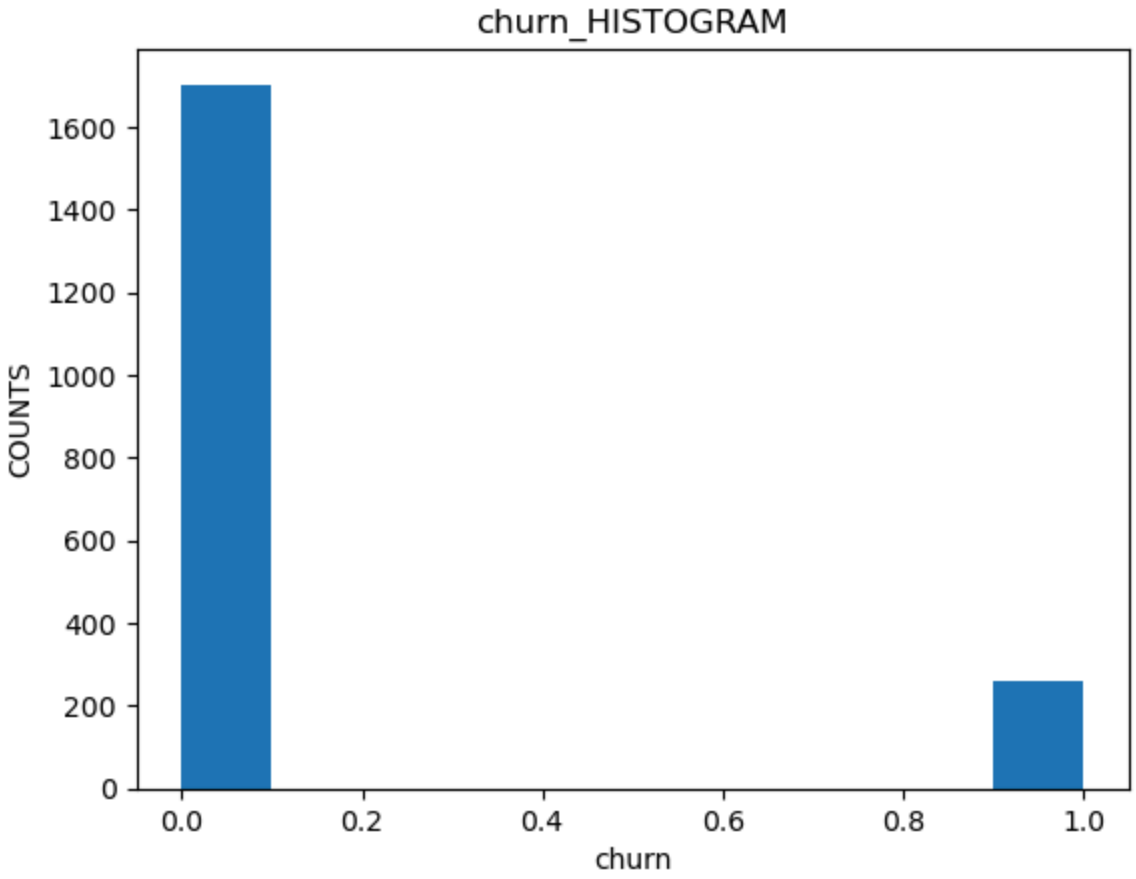

maximum_daily_mins_HISTOGRAM
COUNTS
maximum_daily_mins
0
100
200
300
400
500
0
10
20
30
40
50
60


weekly_max_night_mins_HISTOGRAM
COUNTS
0
100
200
300
400
500
40
60
80
100
120
140
160
180
weekly_max_night_mins


videos_watched_HISTOGRAM
COUNTS
0
100
200
300
400
500
600
700
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
videos_watched


maximum_days_inactive_HISTOGRAM
COUNTS
0
200
400
600
800
0
1
2
3
4
5
6
maximum_days_inactive


customer_support_calls_HISTOGRAM
COUNTS
0
100
200
300
400
500
600
700
0
2
4
6
8
customer_support_calls



In [29]:
```
Q1=np.quantile(telecom_df['age'],0.25)
Q3=np.quantile(telecom_df['age'],0.75)
IQR=Q3-Q1
lb=Q1-1.5*IQR
ub=Q3+1.5*IQR
con1=telecom_df['age']<lb
con2=telecom_df['age']>ub
con3=con1|con2
outliers_data=telecom_df[con3]
outliers_data
```

Out[29]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched	minimum_daily_mins	maximum
2	Female	65	126	no	no	87.0	12.0	
30	Female	63	106	no	no	281.0	11.0	
71	Male	67	163	no	no	372.0	10.0	
87	Male	64	21	no	yes	199.0	13.0	
154	Female	66	68	no	no	223.0	12.0	
...	...	...	...	...	...	...	...	
1852	Male	65	58	no	no	352.0	10.0	
1855	Female	72	143	no	no	304.0	5.0	
1884	Male	69	73	no	yes	123.0	12.0	
1970	Female	67	144	yes	no	225.0	9.0	
1981	Female	70	93	no	no	285.0	9.0	

63 rows × 13 columns

In [31]:
```
Q1=np.quantile(telecom_df['age'],0.25)
Q3=np.quantile(telecom_df['age'],0.75)
IQR1=Q3-Q1
lb1=Q1-1.5*IQR1
ub1=Q3+1.5*IQR1
con4=telecom_df['age']>lb1
con5=telecom_df['age']<ub1
con6=con4&con5
non_outliers_data=telecom_df[con6]
non_outliers_data
```

Out[31]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched	minimum_daily_mins	maximum
0	Female	36	62	no	no	148.0	12.0	
1	Female	39	149	no	no	294.0	8.0	
3	Female	24	131	no	yes	321.0	10.0	
4	Female	40	191	no	no	243.0	11.0	
5	Male	31	65	no	no	194.0	13.0	
...	...	...	...	...	...	...	...	
1995	Female	54	75	no	yes	182.0	11.0	
1996	Male	45	127	no	no	273.0	9.0	
1997	Male	53	94	no	no	129.0	16.0	
1998	Male	40	94	no	no	178.0	10.0	
1999	Male	37	73	no	no	327.0	10.0	

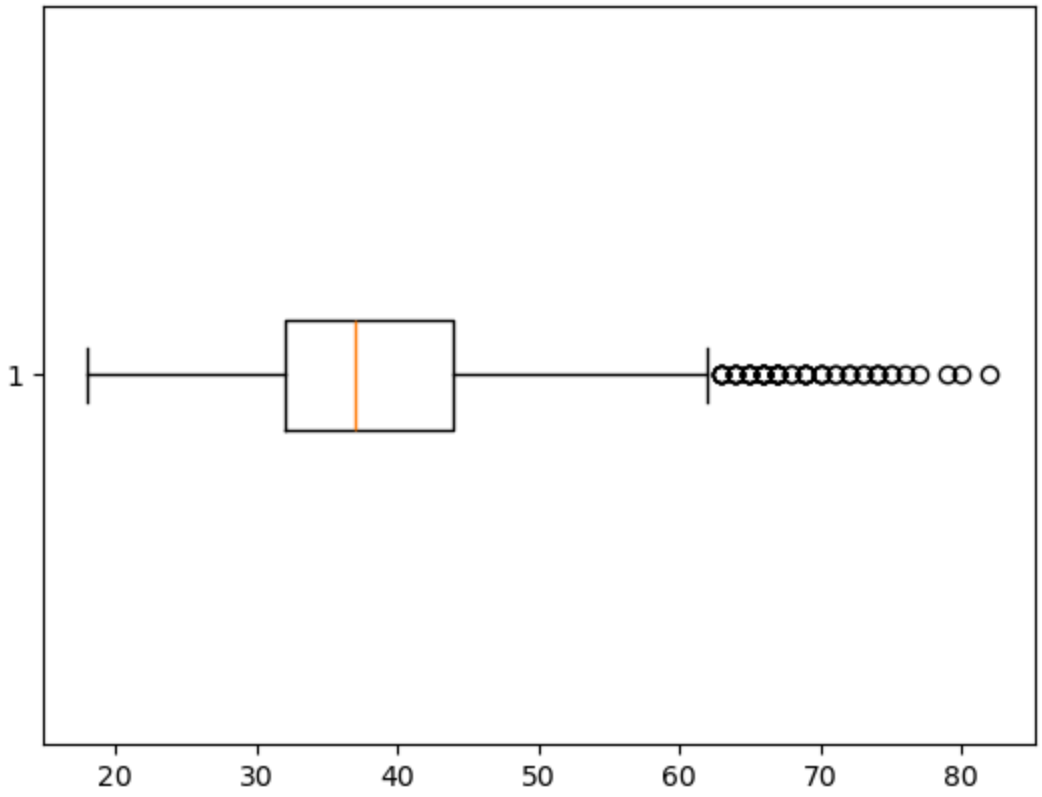
1926 rows × 13 columns

In [321…

```
plt.boxplot(telecom_df['age'],vert=False)
```

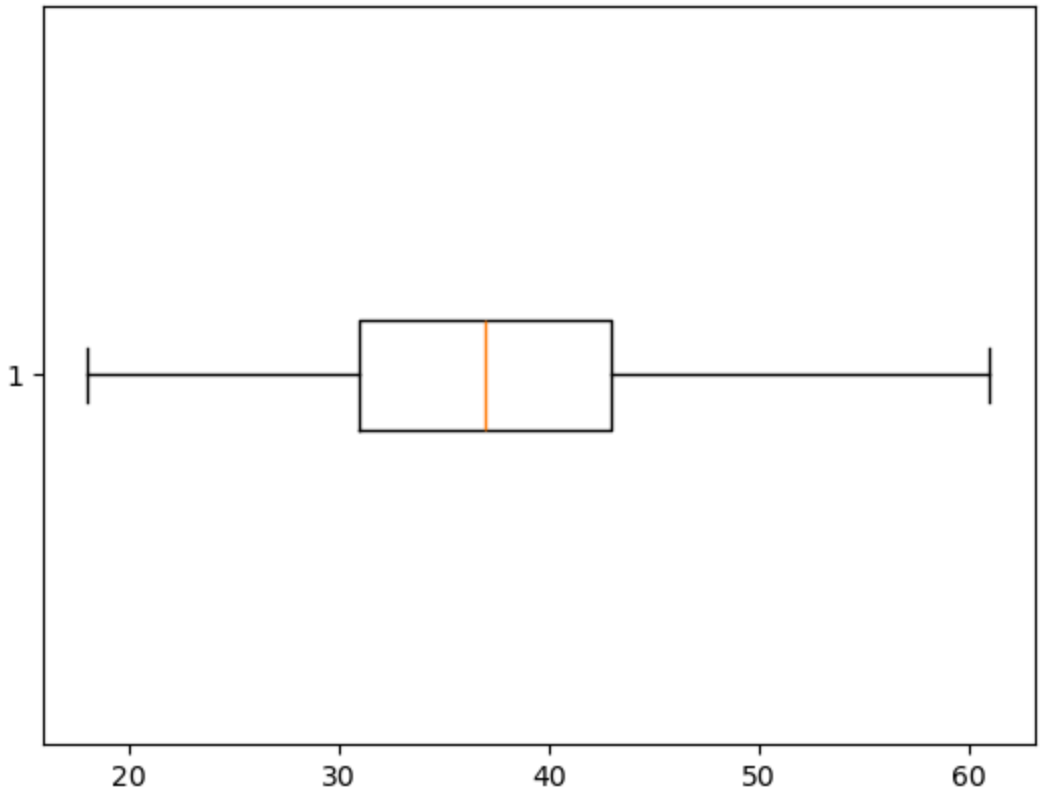
Out[321…

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1a70ee84fb0>,
  <matplotlib.lines.Line2D at 0x1a70ecc0d40>],
 'caps': [<matplotlib.lines.Line2D at 0x1a70ee85550>,
  <matplotlib.lines.Line2D at 0x1a70ee85700>],
 'boxes': [<matplotlib.lines.Line2D at 0x1a70713fd10>],
 'medians': [<matplotlib.lines.Line2D at 0x1a70ee85790>],
 'fliers': [<matplotlib.lines.Line2D at 0x1a70ee85ee0>],
 'means': []}
```



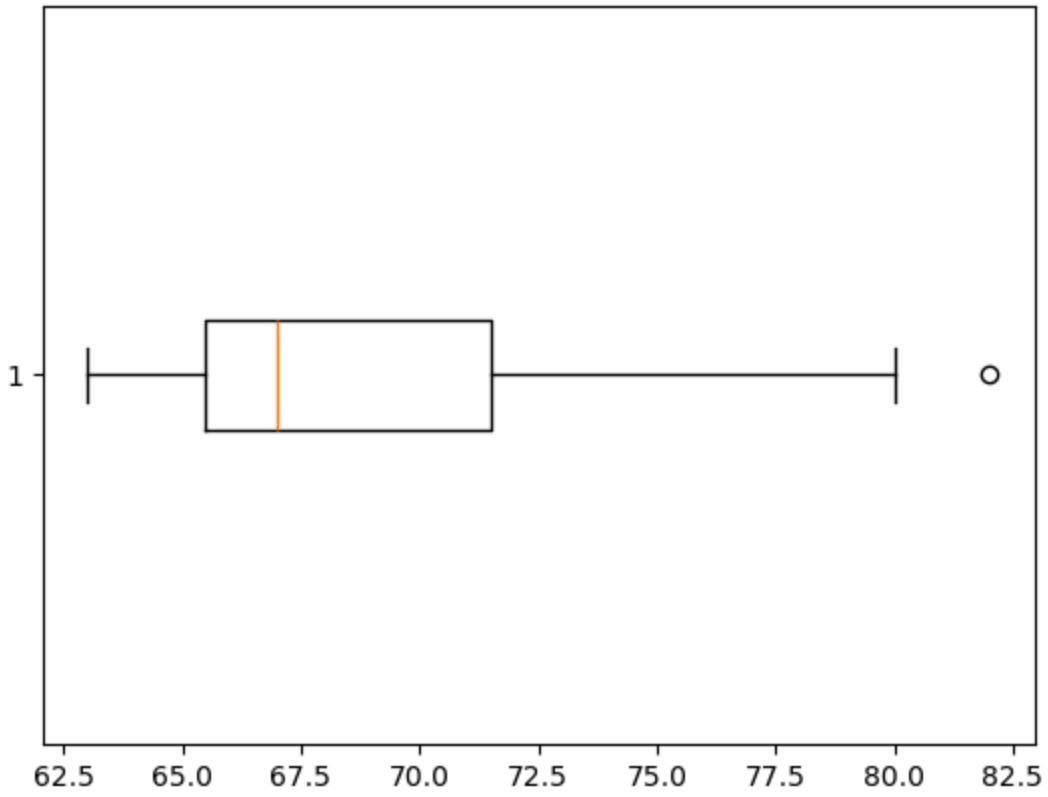
In [323…
```
plt.boxplot(non_outliers_data['age'],vert=False)
```

Out[323…
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1a70ef0c920>,
  <matplotlib.lines.Line2D at 0x1a70ef0cc50>],
 'caps': [<matplotlib.lines.Line2D at 0x1a70ef0cdd0>,
  <matplotlib.lines.Line2D at 0x1a70ef0d0d0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1a70ef0c710>],
 'medians': [<matplotlib.lines.Line2D at 0x1a70ef0d3a0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1a70ef0d670>],
 'means': []}
```



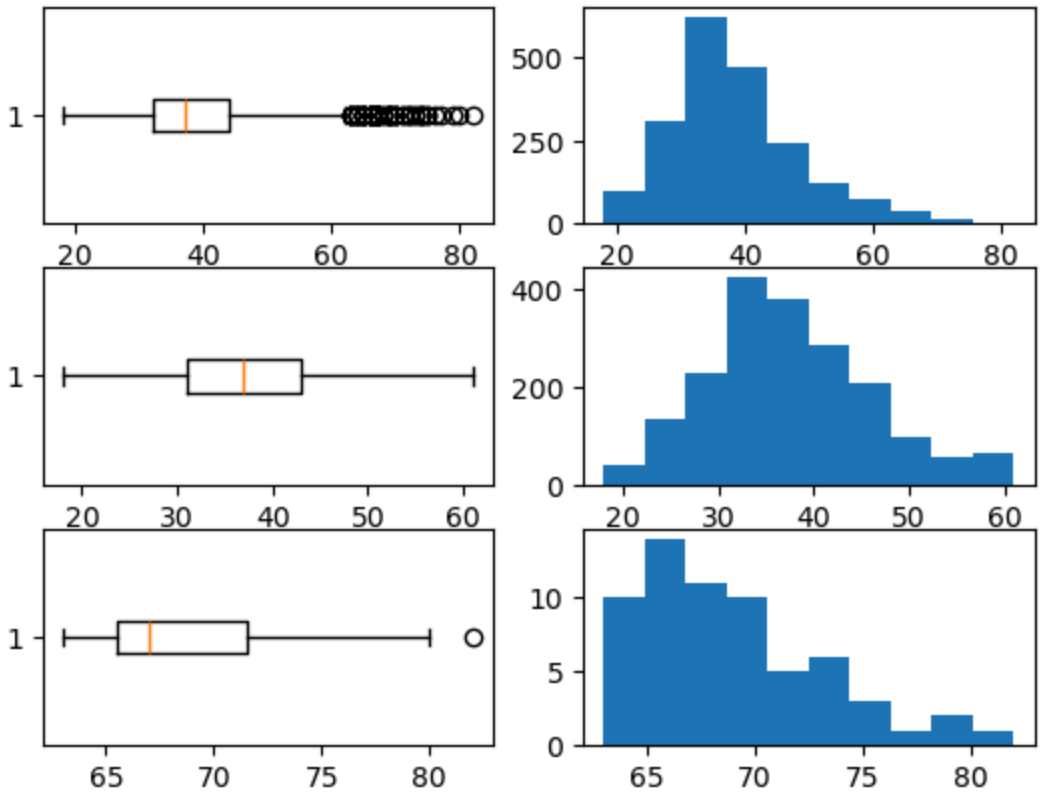
In [325…
```
plt.boxplot(outliers_data['age'],vert=False)
```

Out[325…
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1a70ef96000>,
  <matplotlib.lines.Line2D at 0x1a70ef2d190>],
 'caps': [<matplotlib.lines.Line2D at 0x1a70ef2cb60>,
  <matplotlib.lines.Line2D at 0x1a70ef2c1a0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1a70ef1aa20>],
 'medians': [<matplotlib.lines.Line2D at 0x1a70ecf3410>],
 'fliers': [<matplotlib.lines.Line2D at 0x1a70ef1bc50>],
 'means': []}
```



In [327…

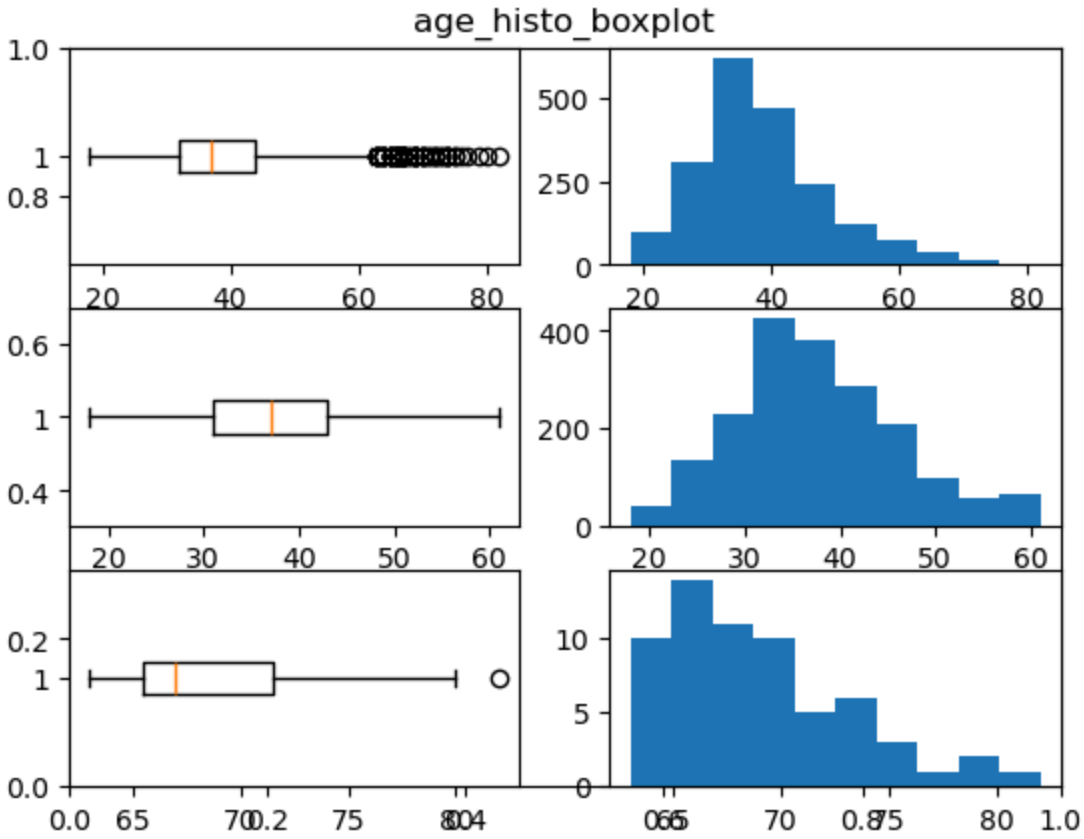
```
plt.subplot(3,2,1).boxplot(telecom_df['age'],vert=False)
plt.subplot(3,2,2).hist(telecom_df['age'])
plt.subplot(3,2,3).boxplot(non_outliers_data['age'],vert=False)
plt.subplot(3,2,4).hist(non_outliers_data['age'])
plt.subplot(3,2,5).boxplot(outliers_data['age'],vert=False)
plt.subplot(3,2,6).hist(outliers_data['age'])
plt.show()
```

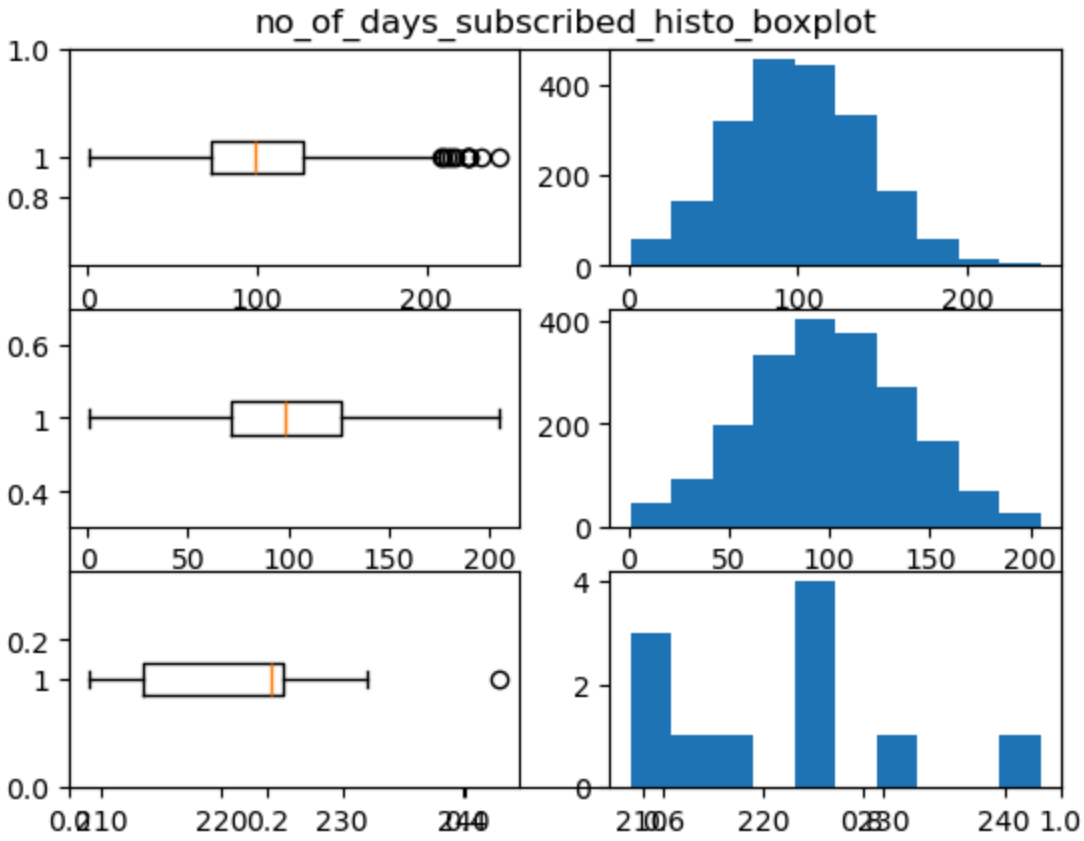


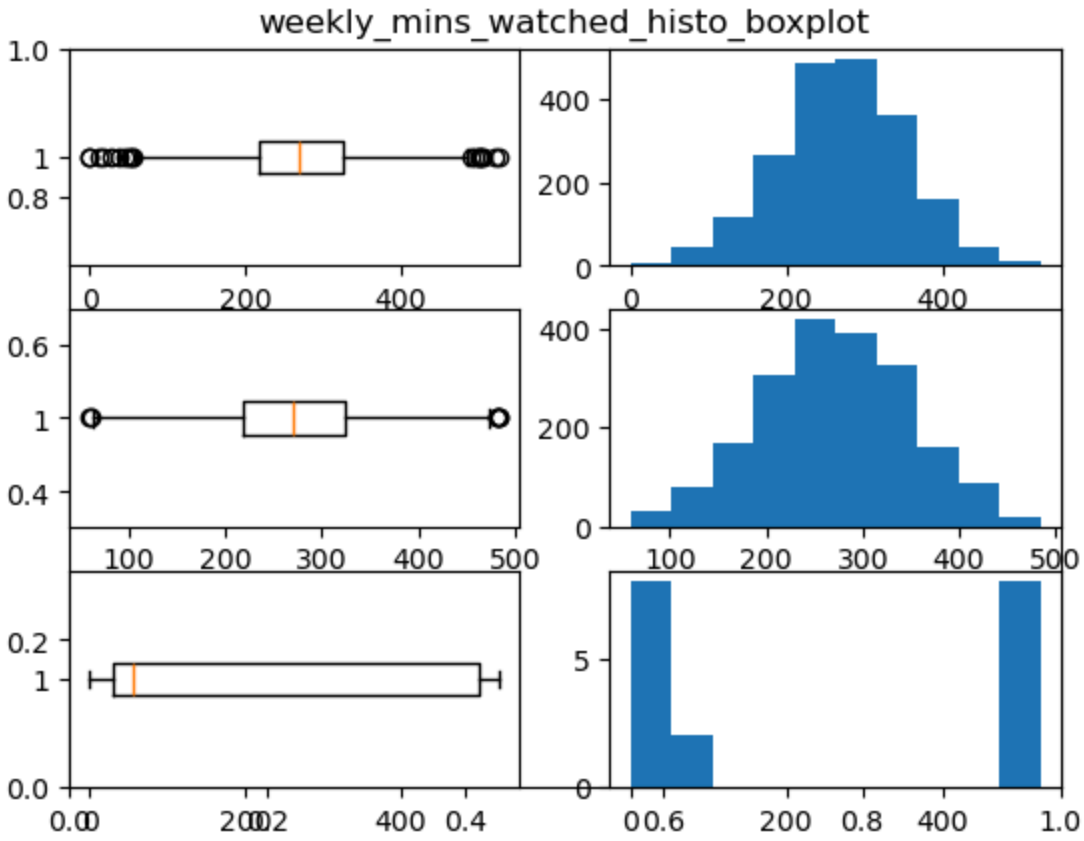
In [329…

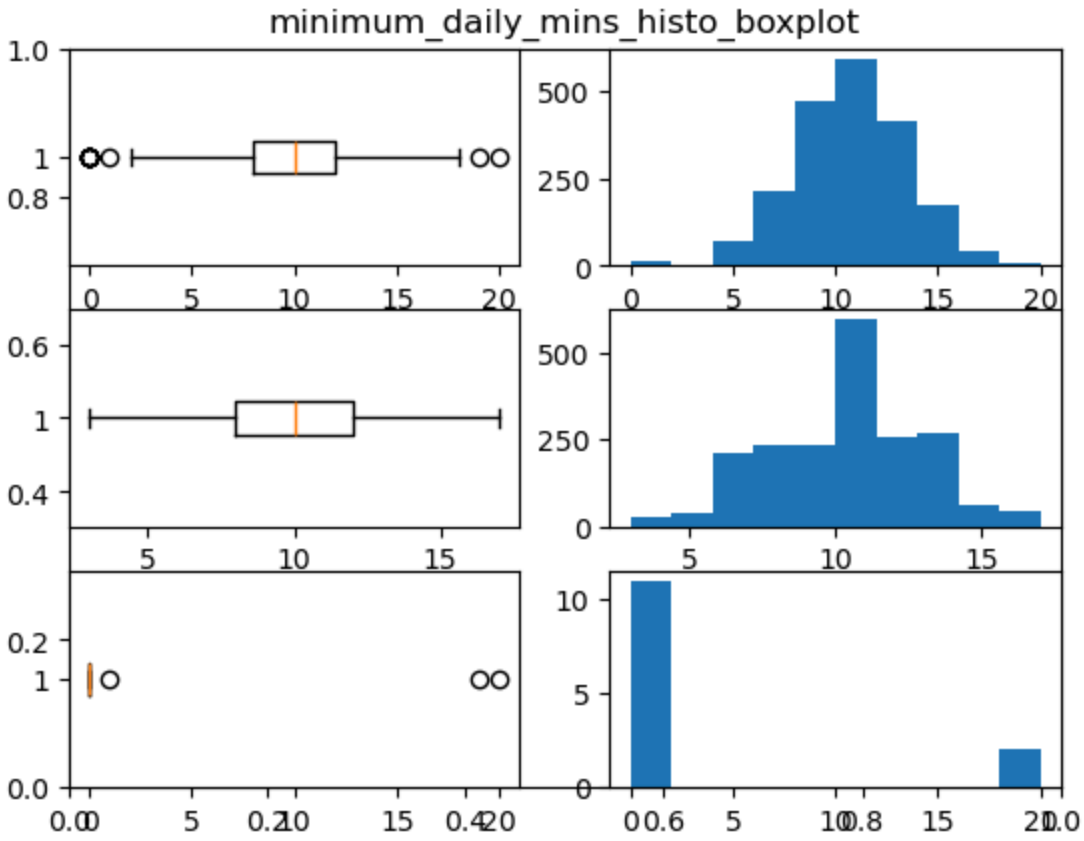
```
for i in num:
    Q1=np.quantile(telecom_df[i],0.25)
    Q3=np.quantile(telecom_df[i],0.75)
    IQR=Q3-Q1
    lb=Q1-1.5*IQR
    ub=Q3+1.5*IQR
    con1=telecom_df[i]<lb
    con2=telecom_df[i]>ub
    con3=con1|con2
    outliers_data=telecom_df[con3]
    con4=telecom_df[i]>lb
    con5=telecom_df[i]<ub
    con6=con4&con5
    non_outliers_data=telecom_df[con6]
    plt.title(f'{i}_histo_boxplot')
    plt.subplot(3,2,1).boxplot(telecom_df[i],vert=False)
```

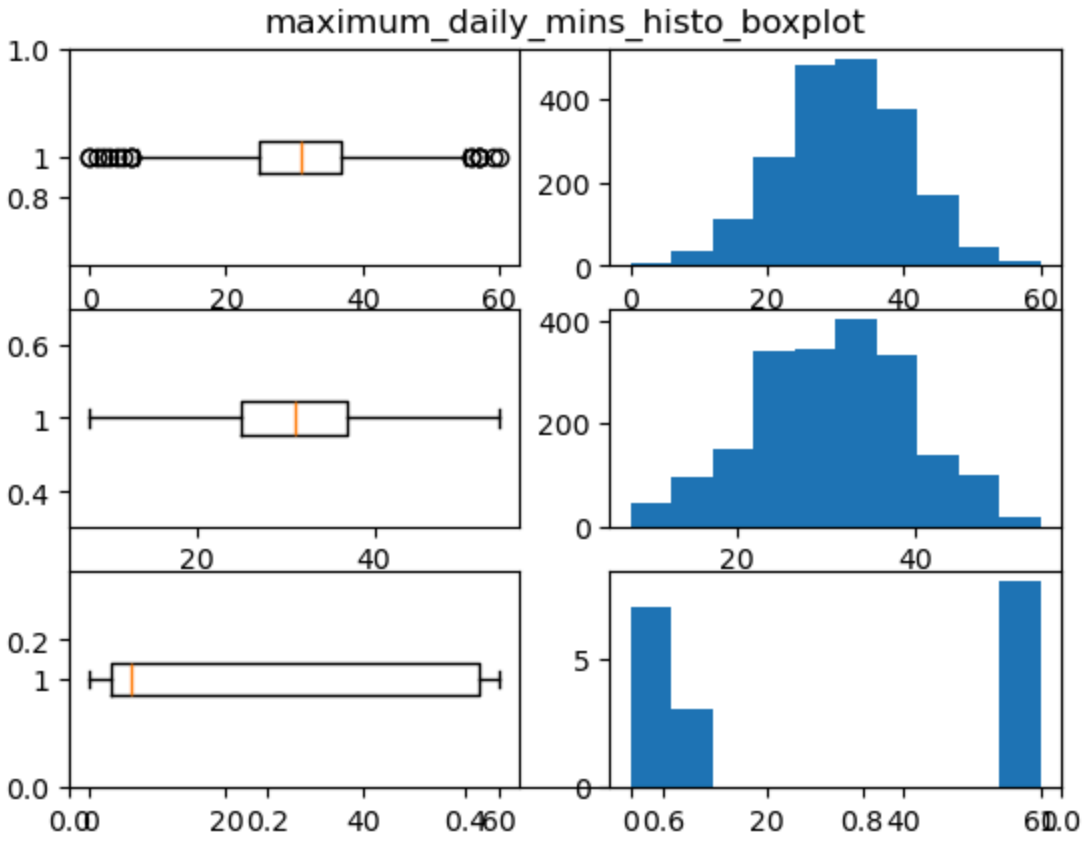
```
    plt.subplot(3,2,2).hist(telecom_df[i])
    plt.subplot(3,2,3).boxplot(non_outliers_data[i],vert=False)
    plt.subplot(3,2,4).hist(non_outliers_data[i])
    plt.subplot(3,2,5).boxplot(outliers_data[i],vert=False)
    plt.subplot(3,2,6).hist(outliers_data[i])
    plt.show()
```

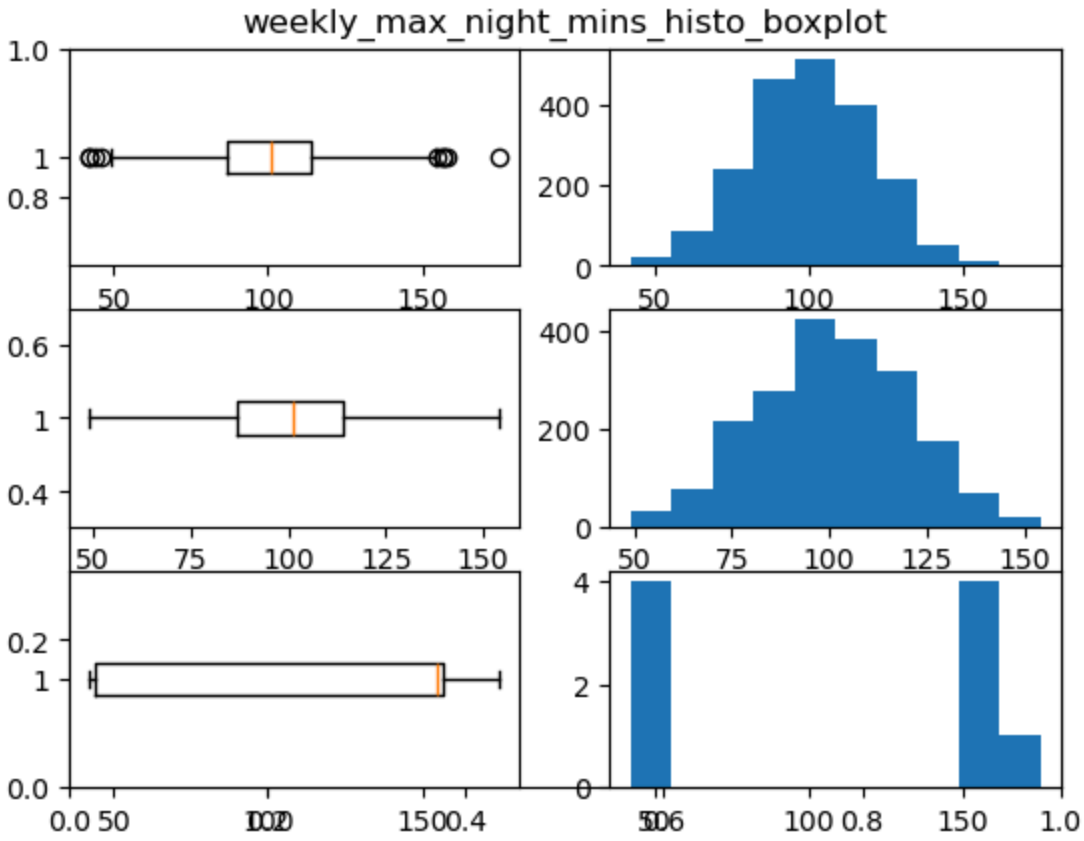


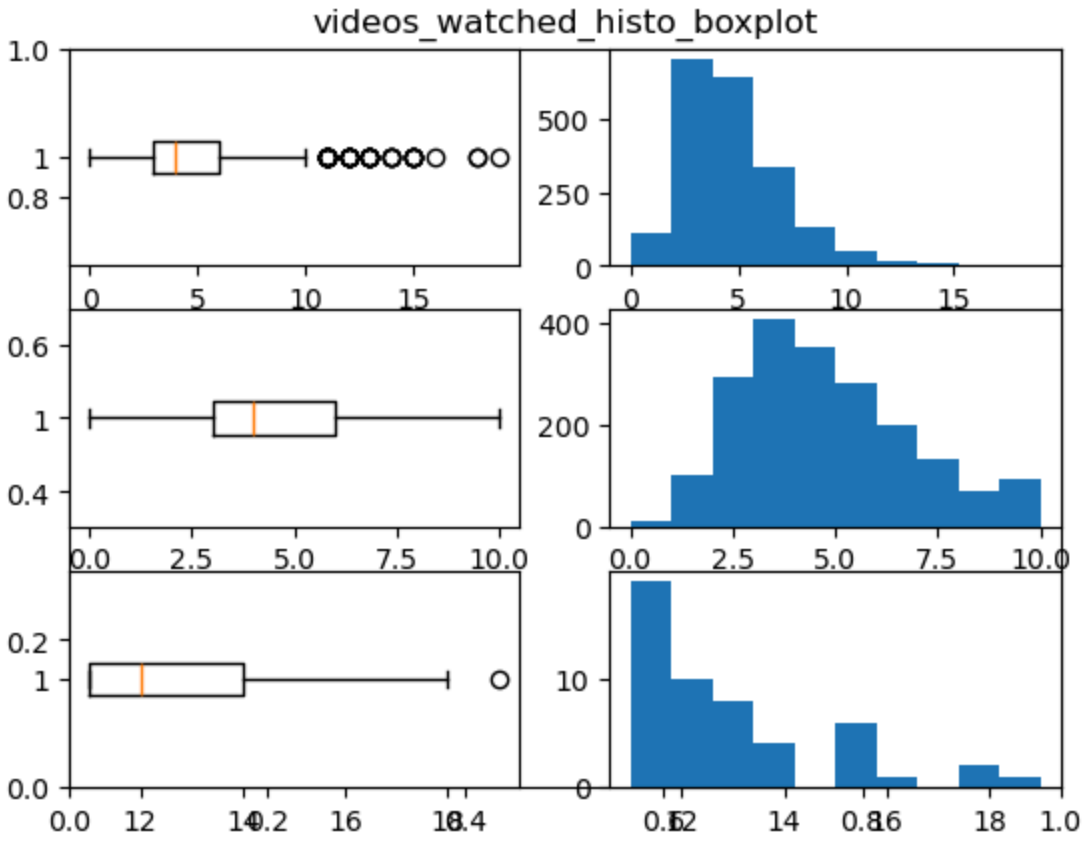

no_of_days_subscribed_histo_boxplot

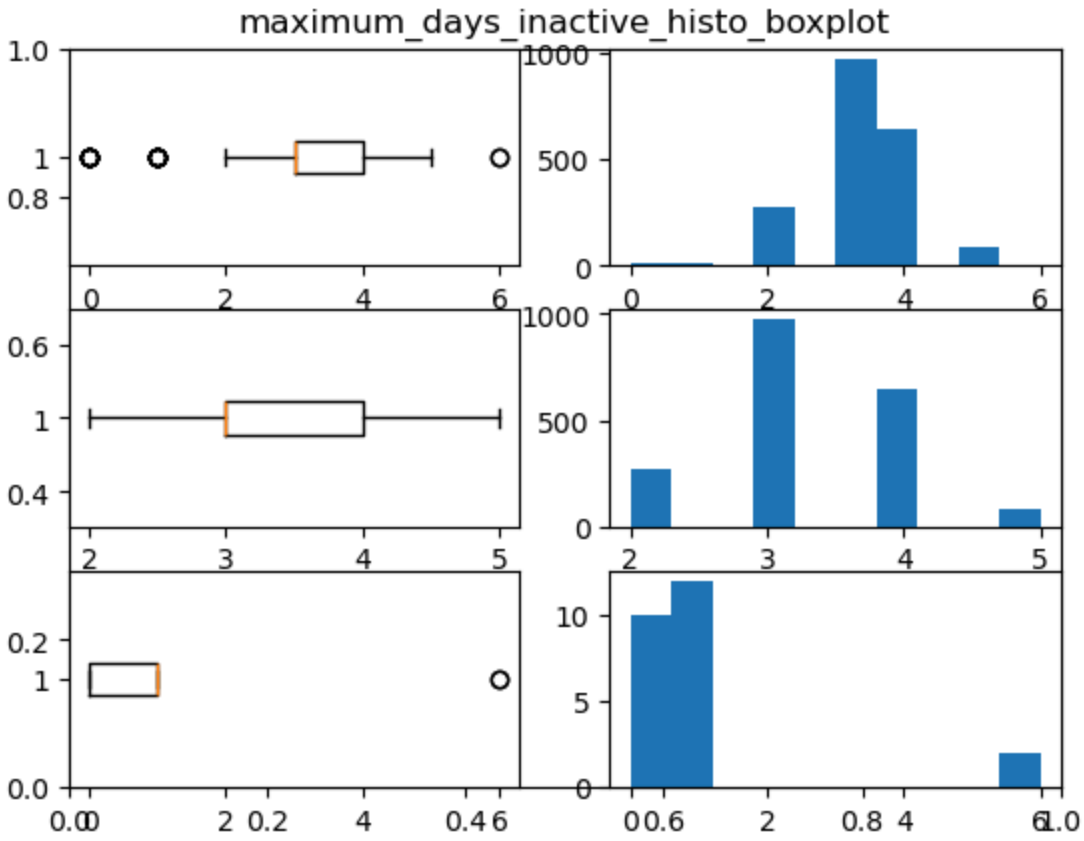

weekly_mins_watched_histo_boxplot
1.0
1
0.8
0
200
400
400
200
0
0
200
400
0.6
1
0.4
100
200
300
400
500
400
200
0
100
200
300
400
500
0.2
1
0.0
0.0
200
400
0.4
5
0
0
0.6
200
0.8
400
1.0

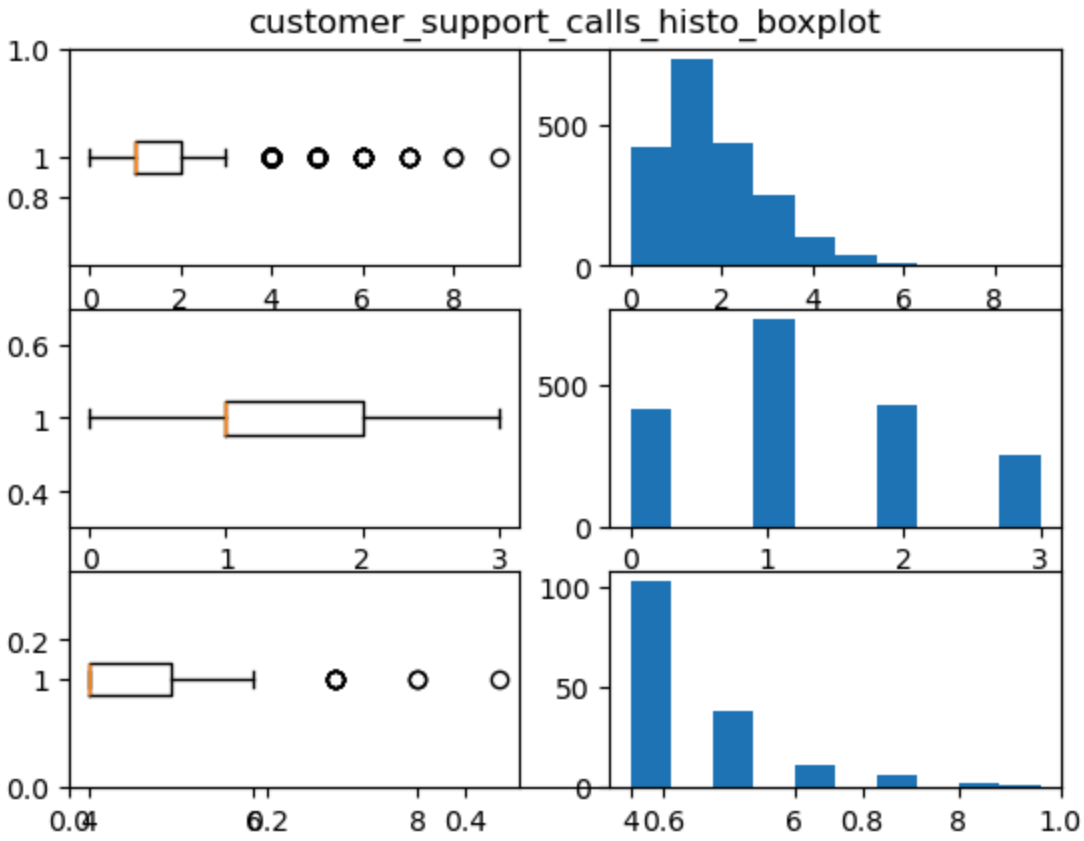

minimum_daily_mins_histo_boxplot

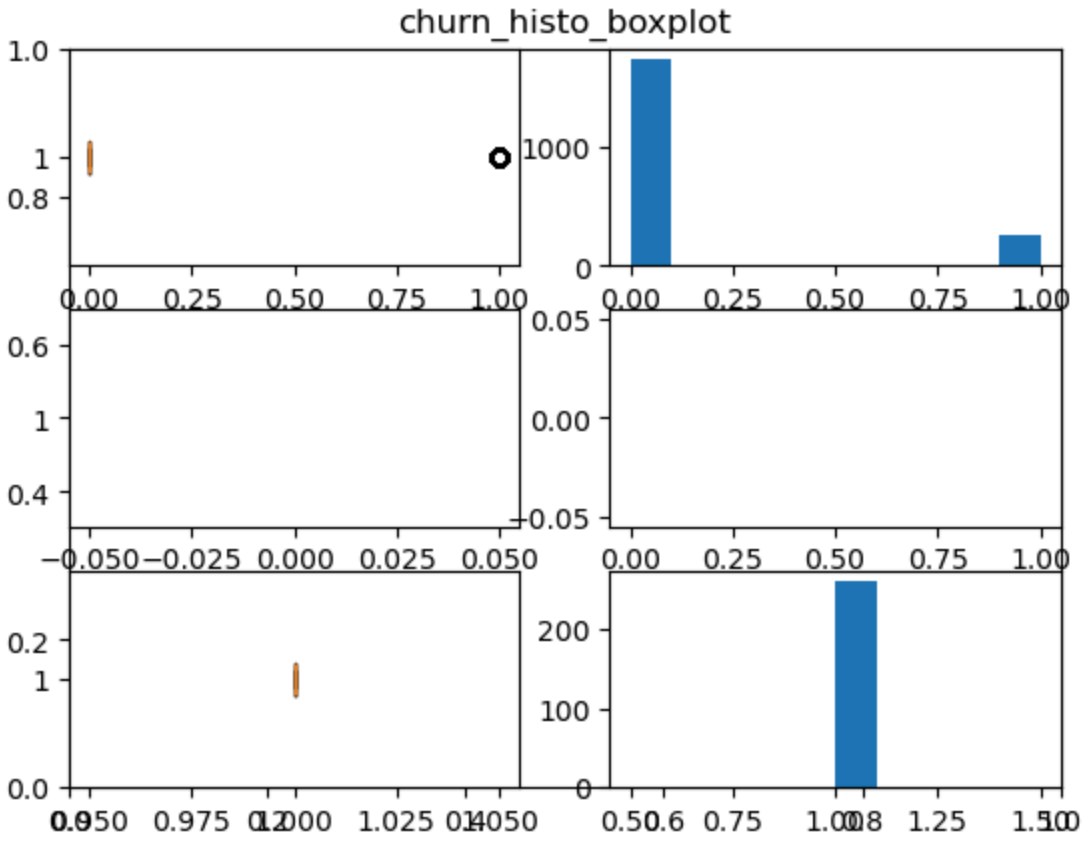

maximum_daily_mins_histo_boxplot


weekly_max_night_mins_histo_boxplot
1.0
1
0.8
0.6
1
0.4
0.2
1
0.0
400
200
0
400
200
0
4
2
0
50
100
150
50
75
100
125
150
0.0
0.4
0.8
1.0


videos_watched_histo_boxplot


maximum_days_inactive_histo_boxplot


customer_support_calls_histo_boxplot
1.0
1
0.8
0.6
1
0.4
0.2
1
0.0
0
2
4
6
8
500
0
0
1
2
3
500
0
0
1
2
3
100
50
0
0.4
0.6
0.8
1.0



OUTLIERS ANALYSIS

In [33]:
```
q1=np.quantile(telecom_df['age'],0.25)
q3=np.quantile(telecom_df['age'],0.75)
med=round(telecom_df['age'].median())
iqr=q3-q1
lb1=q1-1.5*iqr
ub1=q3+1.5*iqr
new_data=[]
for i in telecom_df['age']:
    if i<lb1 or i>ub1:
        new_data.append(med)
    else:
        new_data.append(i)
```

```
telecom_df['age']=new_data
telecom_df
```

Out[33]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched	minimum_daily_mins	maximum
0	Female	36	62	no	no	148.0	12.0	
1	Female	39	149	no	no	294.0	8.0	
2	Female	37	126	no	no	87.0	12.0	
3	Female	24	131	no	yes	321.0	10.0	
4	Female	40	191	no	no	243.0	11.0	
...	...	...	...	...	...	...	...	
1995	Female	54	75	no	yes	182.0	11.0	
1996	Male	45	127	no	no	273.0	9.0	
1997	Male	53	94	no	no	129.0	16.0	
1998	Male	40	94	no	no	178.0	10.0	
1999	Male	37	73	no	no	327.0	10.0	

2000 rows × 13 columns

In [35]:

```
for j in num[:1]:
    q1=np.quantile(telecom_df[j],0.25)
    q3=np.quantile(telecom_df[j],0.75)
    med=round(telecom_df[j].median())
    iqr=q3-q1
    lb1=q1-1.5*iqr
    ub1=q3+1.5*iqr
    new_data=[]
    for i in telecom_df[j]:
        if i<lb1 or i>ub1:
            new_data.append(med)
        else:
            new_data.append(i)
```

```
    telecom_df[j]=new_data
telecom_df
```

Out[35]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched	minimum_daily_mins	maximum
0	Female	36	62	no	no	148.0	12.0	
1	Female	39	149	no	no	294.0	8.0	
2	Female	37	126	no	no	87.0	12.0	
3	Female	24	131	no	yes	321.0	10.0	
4	Female	40	191	no	no	243.0	11.0	
...	...	...	...	...	...	...	...	
1995	Female	54	75	no	yes	182.0	11.0	
1996	Male	45	127	no	no	273.0	9.0	
1997	Male	53	94	no	no	129.0	16.0	
1998	Male	40	94	no	no	178.0	10.0	
1999	Male	37	73	no	no	327.0	10.0	

2000 rows × 13 columns

In [47]:

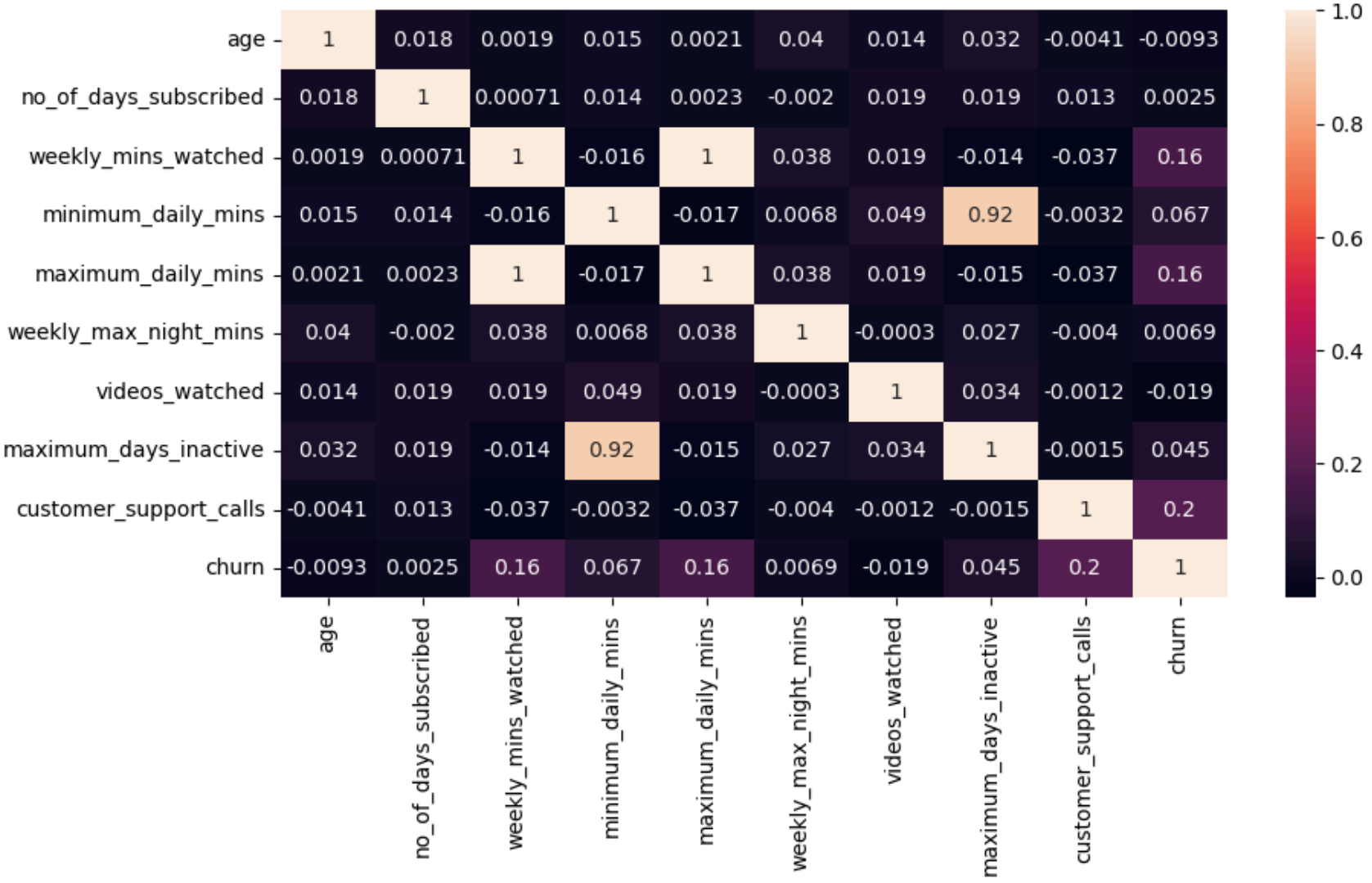
```
num_corr=telecom_df.corr(numeric_only=True)
num_corr
```

Out[47]:

	age	no_of_days_subscribed	weekly_mins_watched	minimum_daily_mins	maximum_daily_mins	we
age	1.000000	0.017936	0.001937	0.015210	0.002090	
no_of_days_subscribed	0.017936	1.000000	0.000706	0.014317	0.002278	
weekly_mins_watched	0.001937	0.000706	1.000000	-0.016341	0.999493	
minimum_daily_mins	0.015210	0.014317	-0.016341	1.000000	-0.017131	
maximum_daily_mins	0.002090	0.002278	0.999493	-0.017131	1.000000	
weekly_max_night_mins	0.040461	-0.001967	0.037780	0.006799	0.038193	
videos_watched	0.014284	0.019414	0.018619	0.048514	0.019366	
maximum_days_inactive	0.032164	0.019338	-0.014064	0.920389	-0.014779	
customer_support_calls	-0.004074	0.013419	-0.036866	-0.003236	-0.036526	
churn	-0.009296	0.002517	0.162977	0.066680	0.162561	

In [53]:
```
plt.figure(figsize=(10,5))
sns.heatmap(num_corr,annot=True)
```

Out[53]: <Axes: >



CATEGORICAL TO NUMERICAL COLUMNS

In [58]:
```
from sklearn.preprocessing import LabelEncoder
label=LabelEncoder()
for i in cat:
    telecom_df[i]=label.fit_transform(telecom_df[i])

telecom_df
```

Out[58]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched	minimum_daily_mins	maximum
0	0	36	62	0	0	148.0	12.0	
1	0	39	149	0	0	294.0	8.0	
2	0	37	126	0	0	87.0	12.0	
3	0	24	131	0	1	321.0	10.0	
4	0	40	191	0	0	243.0	11.0	
...	...	...	...	...	...	...	...	
1995	0	54	75	0	1	182.0	11.0	
1996	1	45	127	0	0	273.0	9.0	
1997	1	53	94	0	0	129.0	16.0	
1998	1	40	94	0	0	178.0	10.0	
1999	1	37	73	0	0	327.0	10.0	

2000 rows × 13 columns

In [62]:

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
for i in num:
    telecom_df[i]=scaler.fit_transform(telecom_df[[i]])

telecom_df
```

Out[62]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched	minimum_daily_mins	max
0	0	-0.137921	-0.949794	0	0	-1.517013	0.644145	
1	0	0.248232	1.239136	0	0	0.295695	-0.790876	
2	0	-0.009203	0.660453	0	0	-2.274378	0.644145	
3	0	-1.682534	0.786254	0	1	0.630922	-0.073365	
4	0	0.376950	2.295860	0	0	-0.337511	0.285390	
...	...	...	...	...	...	...	...	
1995	0	2.178998	-0.622713	0	1	-1.094876	0.285390	
1996	1	1.020539	0.685613	0	0	0.034963	-0.432121	
1997	1	2.050280	-0.144671	0	0	-1.752914	2.079167	
1998	1	0.376950	-0.144671	0	0	-1.144539	-0.073365	
1999	1	-0.009203	-0.673033	0	0	0.705417	-0.073365	

2000 rows × 13 columns

In [64]:

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
for i in cat:
    telecom_df[i]=scaler.fit_transform(telecom_df[[i]])

telecom_df
```

Out[64]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched	minimum_daily_mins	m
0	-1.080207	-0.137921	-0.949794	-0.331478	-0.631349	-1.517013	0.644145	
1	-1.080207	0.248232	1.239136	-0.331478	-0.631349	0.295695	-0.790876	
2	-1.080207	-0.009203	0.660453	-0.331478	-0.631349	-2.274378	0.644145	
3	-1.080207	-1.682534	0.786254	-0.331478	1.583910	0.630922	-0.073365	
4	-1.080207	0.376950	2.295860	-0.331478	-0.631349	-0.337511	0.285390	
...	...	...	...	...	...	...	...	
1995	-1.080207	2.178998	-0.622713	-0.331478	1.583910	-1.094876	0.285390	
1996	0.925748	1.020539	0.685613	-0.331478	-0.631349	0.034963	-0.432121	
1997	0.925748	2.050280	-0.144671	-0.331478	-0.631349	-1.752914	2.079167	
1998	0.925748	0.376950	-0.144671	-0.331478	-0.631349	-1.144539	-0.073365	
1999	0.925748	-0.009203	-0.673033	-0.331478	-0.631349	0.705417	-0.073365	

2000 rows × 13 columns

In [ ]: