In [3]:
```
import numpy as pd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
file_path=r"C:\Users\YOGESH\Downloads\telecom_churn_data.csv"
telecom_df=pd.read_csv(file_path)
telecom_df
```

Out[4]:

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

2000 rows × 16 columns

In [7]:
```
categorical = telecom_df.select_dtypes(include='object').columns
numerical = telecom_df.select_dtypes(exclude='object').columns

print('categorical :',categorical)
print('numerical   :',numerical)
```

```
categorical : Index(['phone_no', 'gender', 'multi_screen', 'mail_subscribed'], dt
ype='object')
numerical   : 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 [13]:
```
import numpy as np
np.shape(telecom_df)
```

Out[13]: (2000, 16)

In [15]:
```
np.size(telecom_df)
```

Out[15]: 32000

In [23]:
```
len(telecom_df)
```

Out[23]:
```
2000
```

In [25]:
```
np.tile(telecom_df,1)
```

Out[25]:
```
array([[2015, 100198, '409-8743', ..., 4.0, 1, 0.0],
       [2015, 100643, '340-5930', ..., 3.0, 2, 0.0],
       [2015, 100756, '372-3750', ..., 4.0, 5, 1.0],
       ...,
       [2015, 998474, '353-2080', ..., 5.0, 0, 0.0],
       [2015, 998934, '359-7788', ..., nan, 3, 0.0],
       [2015, 999961, '414-1496', ..., 3.0, 1, 1.0]], dtype=object)
```

In [27]:
```
telecom_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   year                    2000 non-null   int64  
 1   customer_id             2000 non-null   int64  
 2   phone_no                2000 non-null   object 
 3   gender                  1976 non-null   object 
 4   age                     2000 non-null   int64  
 5   no_of_days_subscribed   2000 non-null   int64  
 6   multi_screen            2000 non-null   object 
 7   mail_subscribed         2000 non-null   object 
 8   weekly_mins_watched     2000 non-null   float64
 9   minimum_daily_mins      2000 non-null   float64
 10  maximum_daily_mins      2000 non-null   float64
 11  weekly_max_night_mins   2000 non-null   int64  
 12  videos_watched          2000 non-null   int64  
 13  maximum_days_inactive   1972 non-null   float64
 14  customer_support_calls  2000 non-null   int64  
 15  churn                   1965 non-null   float64
dtypes: float64(5), int64(7), object(4)
memory usage: 250.1+ KB
```

In [29]:
```
telecom_df.head(6)
```

Out[29]:

	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen	ma
0	2015	100198	409-8743	Female	36	62	no	
1	2015	100643	340-5930	Female	39	149	no	
2	2015	100756	372-3750	Female	65	126	no	
3	2015	101595	331-4902	Female	24	131	no	
4	2015	101653	351-8398	Female	40	191	no	
5	2015	101953	329-6603	NaN	31	65	no	

In [31]:
```
numerical
```

Out[31]:
```
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 [33]:
```
telecom_df.dropna()
```

Out[33]:

	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen
0	2015	100198	409-8743	Female	36	62	no
1	2015	100643	340-5930	Female	39	149	no
2	2015	100756	372-3750	Female	65	126	no
3	2015	101595	331-4902	Female	24	131	no
4	2015	101653	351-8398	Female	40	191	no
...	...	...	...	...	...	...	...
1990	2015	993714	364-1969	Male	32	61	no
1991	2015	993815	387-5891	Male	49	50	yes
1992	2015	994954	329-3222	Female	42	119	no
1996	2015	998086	383-9255	Male	45	127	no
1999	2015	999961	414-1496	Male	37	73	no

1918 rows × 16 columns

In [35]:
```
import warnings
warnings.filterwarnings('ignore')
back=telecom_df.backfill()
back
```

Out[35]:

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

2000 rows × 16 columns

In [37]:
```
telecom_df.ffill()
```

Out[37]:

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

2000 rows × 16 columns

In [21]:
```
telecom_df.dropna()
```

Out[21]:

	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen
0	2015	100198	409-8743	Female	36	62	no
1	2015	100643	340-5930	Female	39	149	no
2	2015	100756	372-3750	Female	65	126	no
3	2015	101595	331-4902	Female	24	131	no
4	2015	101653	351-8398	Female	40	191	no
...	...	...	...	...	...	...	...
1990	2015	993714	364-1969	Male	32	61	no
1991	2015	993815	387-5891	Male	49	50	yes
1992	2015	994954	329-3222	Female	42	119	no
1996	2015	998086	383-9255	Male	45	127	no
1999	2015	999961	414-1496	Male	37	73	no

1918 rows × 16 columns

In [23]:
```
a=telecom_df['churn'].median(),telecom_df['age'].median(),telecom_df['customer_i
print(list(a),end=' ')
```

```
[0.0, 37.0, 567957.5, 1.0, 30.59, 3.0, 4.0, 2015.0, 101.0, 269.925, 10.2, 99.0]
```

In [25]:
```
from sklearn import impute

dir(impute)
```

Out[25]:
```
['KNNImputer',
 'MissingIndicator',
 'SimpleImputer',
 '__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__getattr__',
 '__loader__',
 '__name__',
 '__package__',
 '__path__',
 '__spec__',
 '_base',
 '_knn',
 'typing']
```

In [31]:
```
from sklearn.impute import KNNImputer

x=telecom_df['churn'],telecom_df['videos_watched'],telecom_df['no_of_days_subscr

imputer = KNNImputer(n_neighbors=2,weights='uniform',add_indicator=False)

z=pd.DataFrame(imputer.fit_transform(x))
```

In [33]:
```
pd.DataFrame(x)
```

Out[33]:

	0	1	2	3	4	5	6	7	8
churn	0.00	0.00	1.0	0.0	0.0	1.00	0.00	1.00	0.0
videos_watched	1.00	3.00	1.0	4.0	7.0	6.00	4.00	9.00	5.0
no_of_days_subscribed	62.00	149.00	126.0	131.0	191.0	65.00	59.00	50.00	205.0
weekly_mins_watched	148.35	294.45	87.3	321.3	243.0	193.65	239.25	196.65	263.7

4 rows × 2000 columns

In [35]:
```
z
```

Out[35]:

	0	1	2	3	4	5	6	7	8	9	...	1990	
0	0.00	0.00	1.0	0.0	0.0	1.00	0.00	1.00	0.0	0.0	...	0.0	
1	1.00	3.00	1.0	4.0	7.0	6.00	4.00	9.00	5.0	2.0	...	6.0	
2	62.00	149.00	126.0	131.0	191.0	65.00	59.00	50.00	205.0	63.0	...	61.0	5
3	148.35	294.45	87.3	321.3	243.0	193.65	239.25	196.65	263.7	316.8	...	67.5	46

4 rows × 2000 columns

In [47]:
```
telecom_df.head(3)
```

Out[47]:

	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen	ma
0	2015	100198	409-8743	Female	36	62	no	
1	2015	100643	340-5930	Female	39	149	no	
2	2015	100756	372-3750	Female	65	126	no	

In [49]:
```
telecom_df
```

Out[49]:

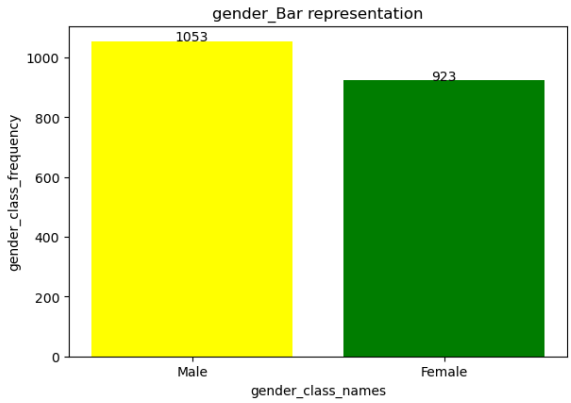
	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen
0	2015	100198	409-8743	Female	36	62	no
1	2015	100643	340-5930	Female	39	149	no
2	2015	100756	372-3750	Female	65	126	no
3	2015	101595	331-4902	Female	24	131	no
4	2015	101653	351-8398	Female	40	191	no
...	...	...	...	...	...	...	...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

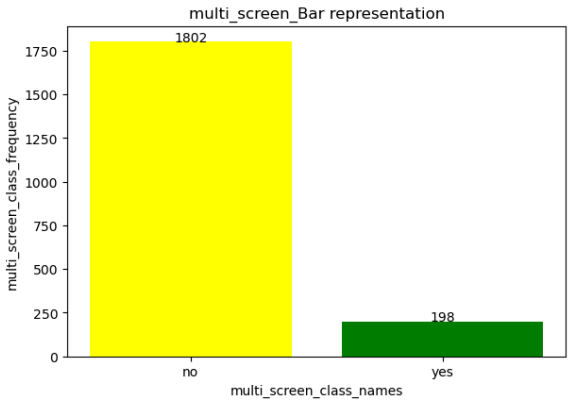
2000 rows × 16 columns

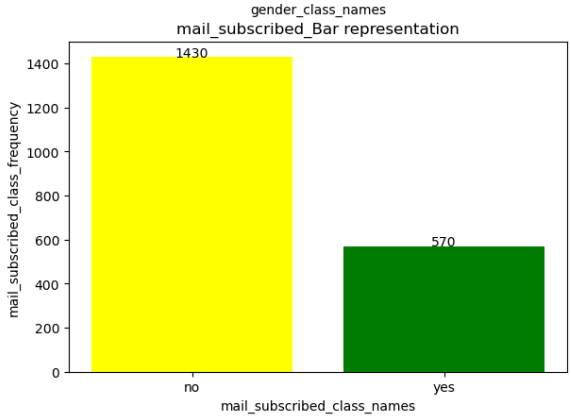
In [51]:
```
categorical
```

Out[51]:
```
Index(['phone_no', 'gender', 'multi_screen', 'mail_subscribed'], dtype='objec
t')
```

In [53]:
```
plt.figure(figsize=(15,10))
color=['yellow','green']
for i in range(1,4):
    keys = telecom_df[categorical[i]].value_counts().keys()
    values = telecom_df[categorical[i]].value_counts().values
    plt.subplot(2,2,i).bar(keys,values,color=color)
    plt.title(f'{categorical[i]}_Bar representation')
    plt.ylabel(f'{categorical[i]}_class_frequency')
    plt.xlabel(f'{categorical[i]}_class_names')
    for i,j in enumerate(values):
        plt.text(i,j+2,str(j),ha='center')
```







In [55]:
```
values1=telecom_df[categorical[1:2]].value_counts().values
keys1=telecom_df[categorical[1:2]].value_counts().keys()
df1=values1/2000*100
df1
```

Out[55]:
```
array([52.65, 46.15])
```

In [57]:
```
values2=telecom_df[categorical[2:3]].value_counts().values
keys2=telecom_df[categorical[2:3]].value_counts().keys()
df2=values2/2000*100
df2
```

Out[57]:
```
array([90.1,  9.9])
```

In [67]:
```
values3=telecom_df[categorical[3:4]].value_counts().values
keys3=telecom_df[categorical[3:4]].value_counts().keys
df3=values3/2000*100
df3
```

Out[67]:
```
array([71.5, 28.5])
```

In [69]:
```
s=df1,df2,df3
s[0]
```

Out[69]:
```
array([52.65, 46.15])
```

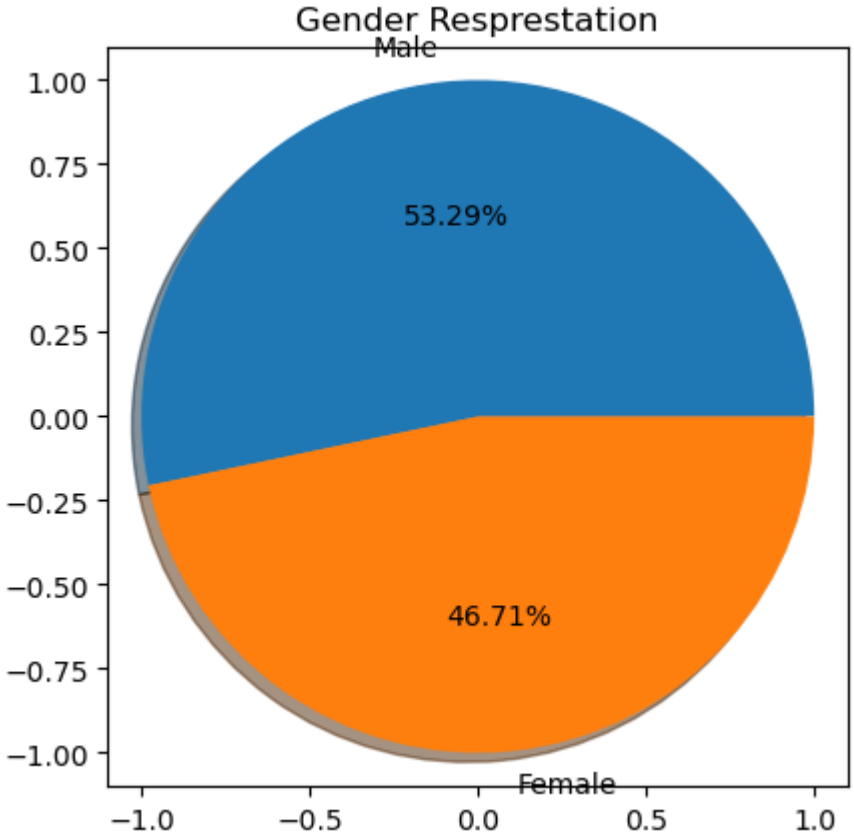
In [71]:
```
categorical
```

Out[71]:
```
Index(['phone_no', 'gender', 'multi_screen', 'mail_subscribed'], dtype='objec
t')
```

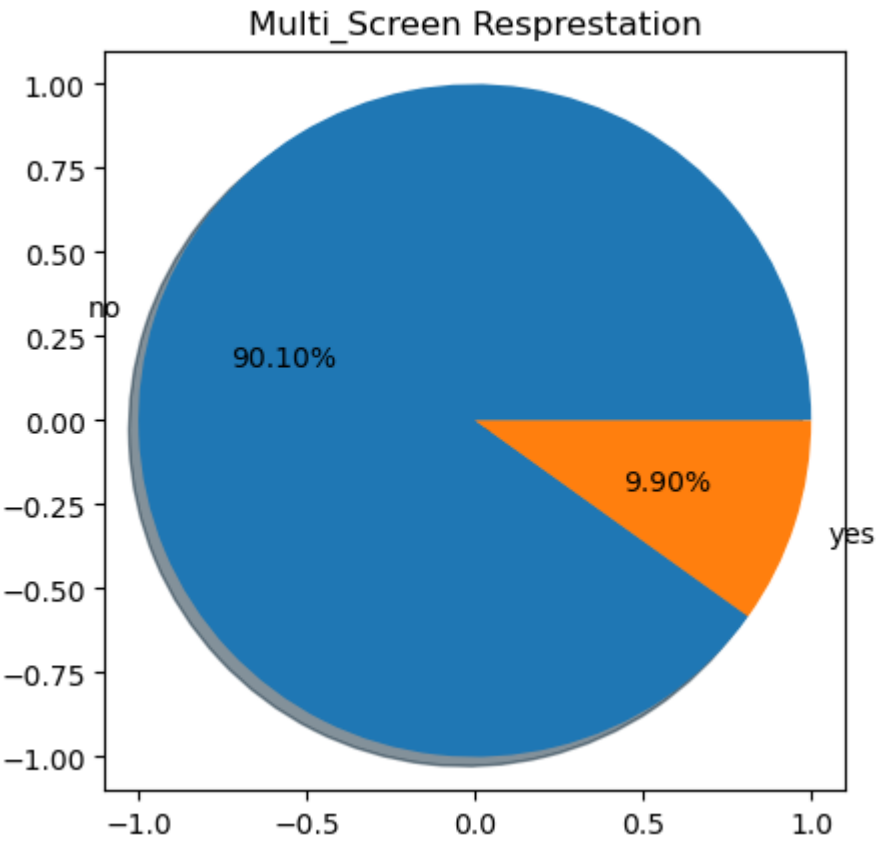
In [73]:
```
n=categorical[0:4]
n
```

Out[73]:
```
Index(['phone_no', 'gender', 'multi_screen', 'mail_subscribed'], dtype='objec
t')
```

In [77]:
```
keys1=telecom_df['gender'].value_counts().keys()
plt.pie(df1,labels=keys1,autopct='%0.2f%%',radius=1,frame=True,shadow=True)
plt.title('Gender Resprestation')
plt.show()
```

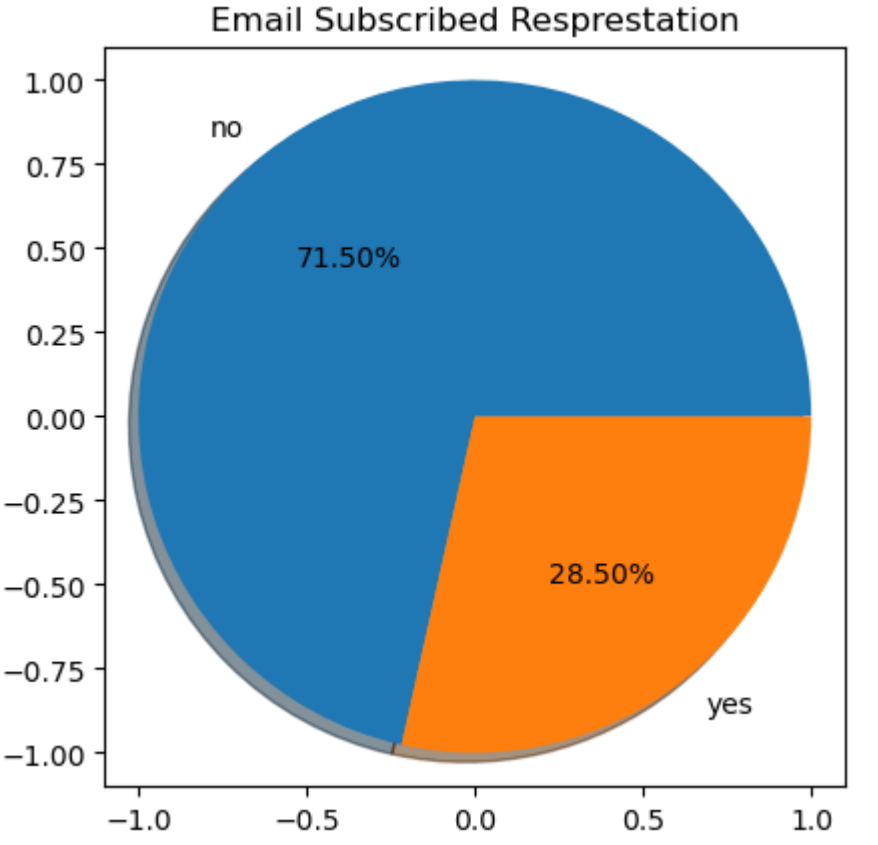


In [81]:
```
keys2=telecom_df['multi_screen'].value_counts().keys()
plt.pie(df2,labels=keys2,autopct='%0.2f%%',radius=1,frame=True,shadow=True)
plt.title('Multi_Screen Resprestation')
plt.show()
```



In [83]:
```
keys3=telecom_df[categorical[3]].value_counts().keys()
plt.pie(df3,labels=keys3,autopct='%0.2f%%',radius=1,frame=True,shadow=True)
plt.title('Email Subscribed Resprestation')
```

Out[83]: Text(0.5, 1.0, 'Email Subscribed Resprestation')



In [87]:
```
telecom_df[numerical[2: ]]
```

Out[87]:

	age	no_of_days_subscribed	weekly_mins_watched	minimum_daily_mins	maximu
0	36	62	148.35	12.2	
1	39	149	294.45	7.7	
2	65	126	87.30	11.9	
3	24	131	321.30	9.5	
4	40	191	243.00	10.9	
...	...	...	...	...	
1995	54	75	182.25	11.3	
1996	45	127	273.45	9.3	
1997	53	94	128.85	15.6	
1998	40	94	178.05	10.4	
1999	37	73	326.70	10.3	

2000 rows × 10 columns

In [89]:
```
telecom_df.describe()
```

Out[89]:

	year	customer_id	age	no_of_days_subscribed	weekly_mins_watched
count	2000.0	2000.000000	2000.00000	2000.000000	2000.000000
mean	2015.0	554887.157500	38.69050	99.750000	270.178425
std	0.0	261033.690318	10.20641	39.755386	80.551627
min	2015.0	100198.000000	18.00000	1.000000	0.000000
25%	2015.0	328634.750000	32.00000	73.000000	218.212500
50%	2015.0	567957.500000	37.00000	99.000000	269.925000
75%	2015.0	773280.250000	44.00000	127.000000	324.675000
max	2015.0	999961.000000	82.00000	243.000000	526.200000

In [91]:
```
count=(len(telecom_df))
count
```

Out[91]:
```
2000
```

In [97]:
```
numerical
```

Out[97]:
```
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 [107…
```
weekly_mins=telecom_df['weekly_max_night_mins']
Min=np.min(weekly_mins)
Mean=np.mean(weekly_mins)
Max=np.max(weekly_mins)
Median=np.median(weekly_mins)
Std=np.std(weekly_mins)
count=(len(weekly_mins))
data25=[count,Min,Mean,Max,Median,Std]
pd.DataFrame(data25,columns=['Weekly_Mins'],index=['count','Min','Mean','Max','M
```

Out[107…

	Weekly_Mins
count	2000.000000
Min	42.000000
Mean	100.415500
Max	175.000000
Median	101.000000
Std	19.524571

In [113…
```
print(np.percentile(telecom_df['weekly_max_night_mins'],50),np.quantile(telecom_
print(np.percentile(telecom_df['weekly_max_night_mins'],25),np.quantile(telecom_
print(np.percentile(telecom_df['weekly_max_night_mins'],75),np.quantile(telecom_
np.percentile(telecom_df['weekly_max_night_mins'],100),np.quantile(telecom_df['w
```

```
101.0 101.0
87.0 87.0
114.0 114.0
```

Out[113… (175.0, 175)

In [117…
```
l=[]
for i in numerical[2:]:
    data=telecom_df[i]
    mean=round(data.mean(),2)
    median=round(data.median(),2)
    Min=round(data.min(),2)
    Max=round(data.max(),2)

    p_25=round(np.percentile(data,25),2)
    p_50=round(np.percentile(data,50),2)
    p_75=round(np.percentile(data,75),2)

    index=['count','Min','Max','mean','median','p25','p50','p75']
    values=[count,Min,Max,mean,median,p_25,p_50,p_75]

    l.append(values)

l
```

Out[117…
```
[[2000, 18, 82, 38.69, 37.0, 32.0, 37.0, 44.0],
 [2000, 1, 243, 99.75, 99.0, 73.0, 99.0, 127.0],
 [2000, 0.0, 526.2, 270.18, 269.93, 218.21, 269.92, 324.68],
 [2000, 0.0, 20.0, 10.2, 10.2, 8.4, 10.2, 12.0],
 [2000, 0.0, 59.64, 30.62, 30.59, 24.74, 30.59, 36.8],
 [2000, 42, 175, 100.42, 101.0, 87.0, 101.0, 114.0],
 [2000, 0, 19, 4.48, 4.0, 3.0, 4.0, 6.0],
 [2000, 0.0, 6.0, 3.25, 3.0, nan, nan, nan],
 [2000, 0, 9, 1.55, 1.0, 1.0, 1.0, 2.0],
 [2000, 0.0, 1.0, 0.13, 0.0, nan, nan, nan]]
```

In [119…
```
pd.DataFrame(l,columns=['count','Min','Max','mean','median','p25','p50','p75'],i
```

Out[119…

	count	Min	Max	mean	median	p25	p50	p75
age	2000	18.0	82.00	38.69	37.00	32.00	37.00	44.00
no_of_days_subscribed	2000	1.0	243.00	99.75	99.00	73.00	99.00	127.00
weekly_mins_watched	2000	0.0	526.20	270.18	269.93	218.21	269.92	324.68
minimum_daily_mins	2000	0.0	20.00	10.20	10.20	8.40	10.20	12.00
maximum_daily_mins	2000	0.0	59.64	30.62	30.59	24.74	30.59	36.80
weekly_max_night_mins	2000	42.0	175.00	100.42	101.00	87.00	101.00	114.00
videos_watched	2000	0.0	19.00	4.48	4.00	3.00	4.00	6.00
maximum_days_inactive	2000	0.0	6.00	3.25	3.00	NaN	NaN	NaN
customer_support_calls	2000	0.0	9.00	1.55	1.00	1.00	1.00	2.00
churn	2000	0.0	1.00	0.13	0.00	NaN	NaN	NaN

In [125…
```
weekly_mean=weekly_mins.mean()
weekly_std=weekly_mins.std()
lb=weekly_mean-1*(weekly_std)
ub=weekly_mean+1*(weekly_std)
con1=weekly_mins>lb
con2=weekly_mins<ub
con= con1 & con2
telecom_df[con]
len(telecom_df[con])==68*2000/100
```

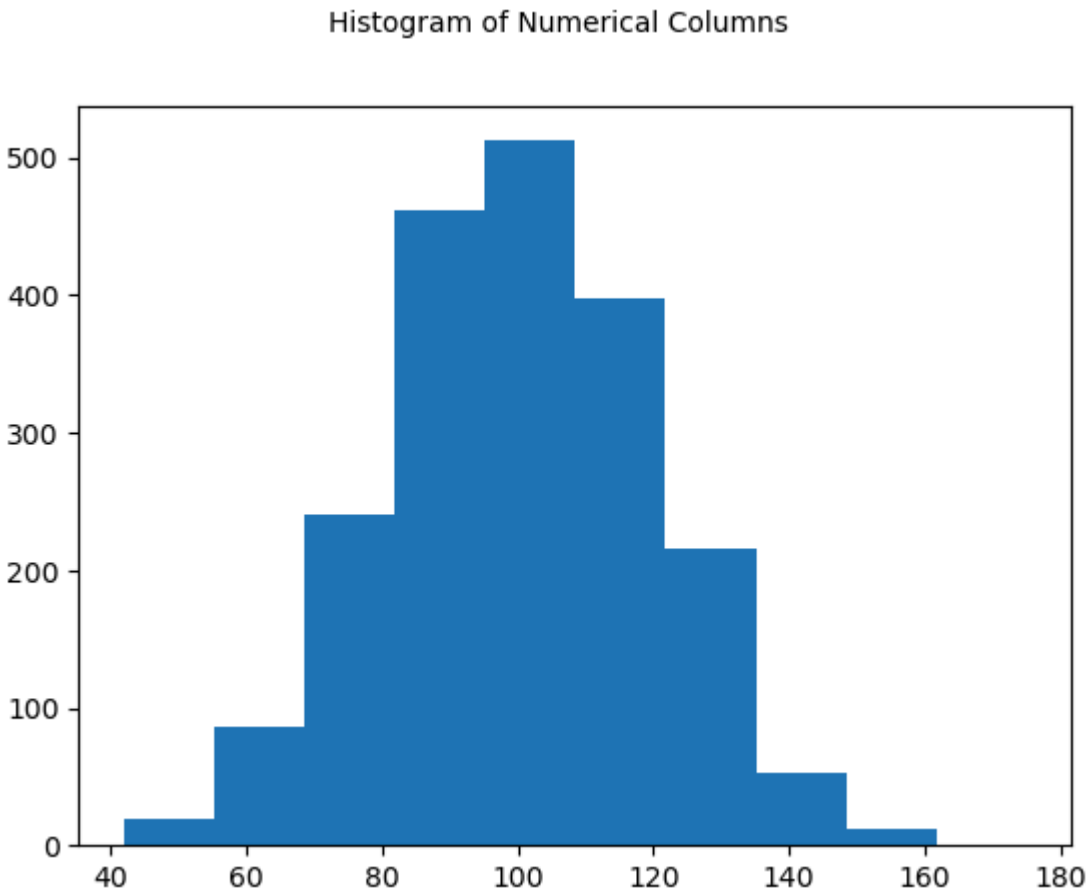
Out[125…
```
False
```

In [129…
```
weekly_mean=weekly_mins.mean()
weekly_std=weekly_mins.std()
lb=weekly_mean+1*(weekly_std)
ub=weekly_mean-1*(weekly_std)
con1=weekly_mins>lb
con2=weekly_mins<ub
con= con1 & con2
telecom_df[con]
len(telecom_df[con])==75*2000/100
```

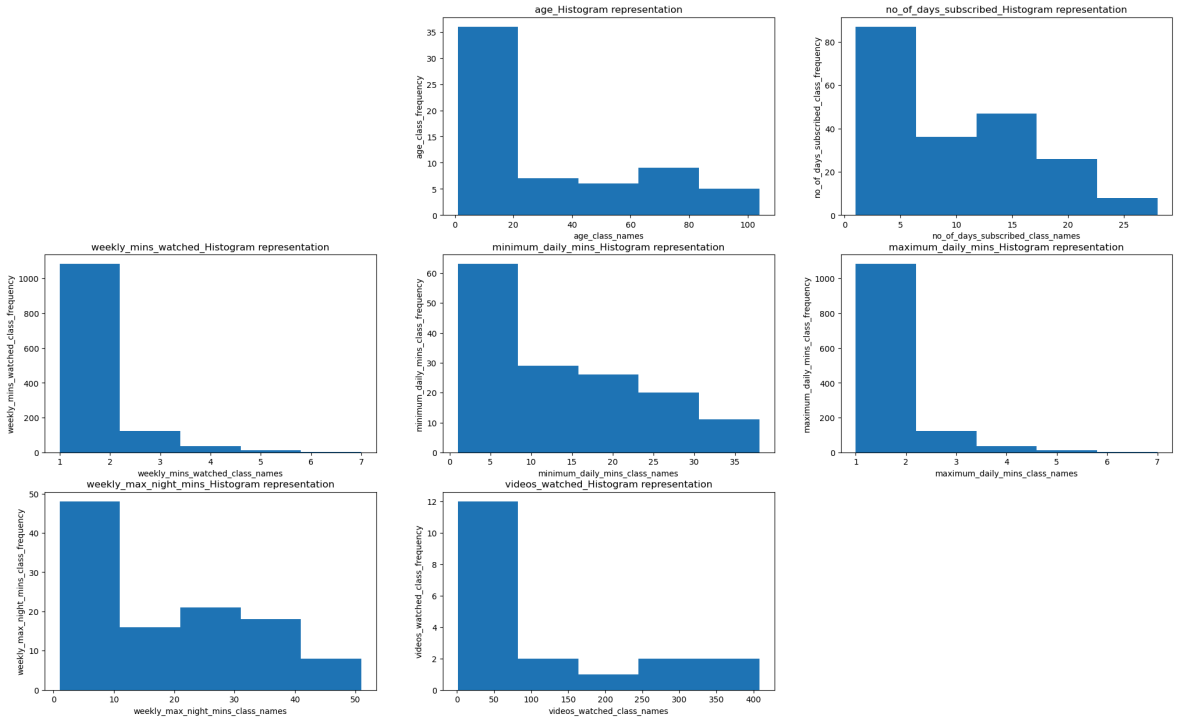
Out[129…
```
False
```

In [131…
```
plt.hist(weekly_mins,bins=10)
plt.suptitle("Histogram of Numerical Columns", fontsize=10)
```

Out[131… Text(0.5, 0.98, 'Histogram of Numerical Columns')

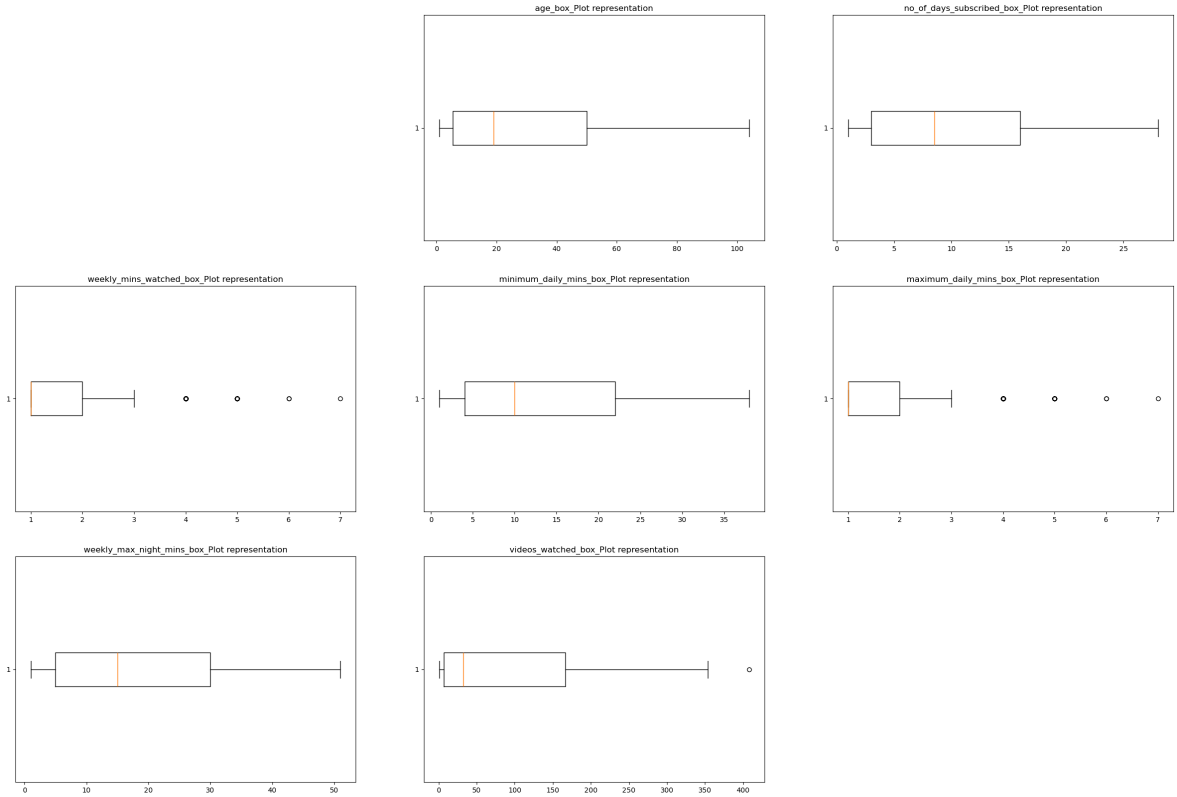


In [137…
```
plt.figure(figsize=(25,15))
for i in range(2,9):
    keys = telecom_df[numerical[i]].value_counts().keys()
    values = telecom_df[numerical[i]].value_counts().values
    plt.subplot(3,3,i).hist(values,5)
    plt.title(f'{numerical[i]}_Histogram representation')
    plt.ylabel(f'{numerical[i]}_class_frequency')
    plt.xlabel(f'{numerical[i]}_class_names')
```



In [141…

```
plt.figure(figsize=(30,20))
for i in range(2,9):
    keys = telecom_df[numerical[i]].value_counts().keys()
    values = telecom_df[numerical[i]].value_counts().values
    plt.subplot(3,3,i).boxplot(values,vert=False)
    plt.title(f'{numerical[i]}_box_Plot representation')
```



In [145…

```
q1=np.percentile(weekly_mins,25)
q3=np.percentile(weekly_mins,75)

Iqr=q3-q1

lb=q1-1.5*(Iqr)
```

```
ub=q3+1.5*(Iqr)

con1=weekly_mins<lb

con2=weekly_mins>ub

con=con1|con2

outliers=telecom_df[con]

outliers
```

Out[145…

	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen
95	2015	136635	335-3320	Male	31	118	no
148	2015	162977	382-6135	Male	30	100	no
383	2015	270563	415-6578	Male	55	79	no
1014	2015	576100	376-9249	Female	37	98	no
1446	2015	751882	370-5527	Male	36	23	no
1632	2015	836418	331-8909	Male	35	121	no
1748	2015	886737	332-2462	Male	37	39	no
1956	2015	979406	366-7360	Male	47	129	no
1980	2015	989058	396-8265	Female	29	151	no

In [147…

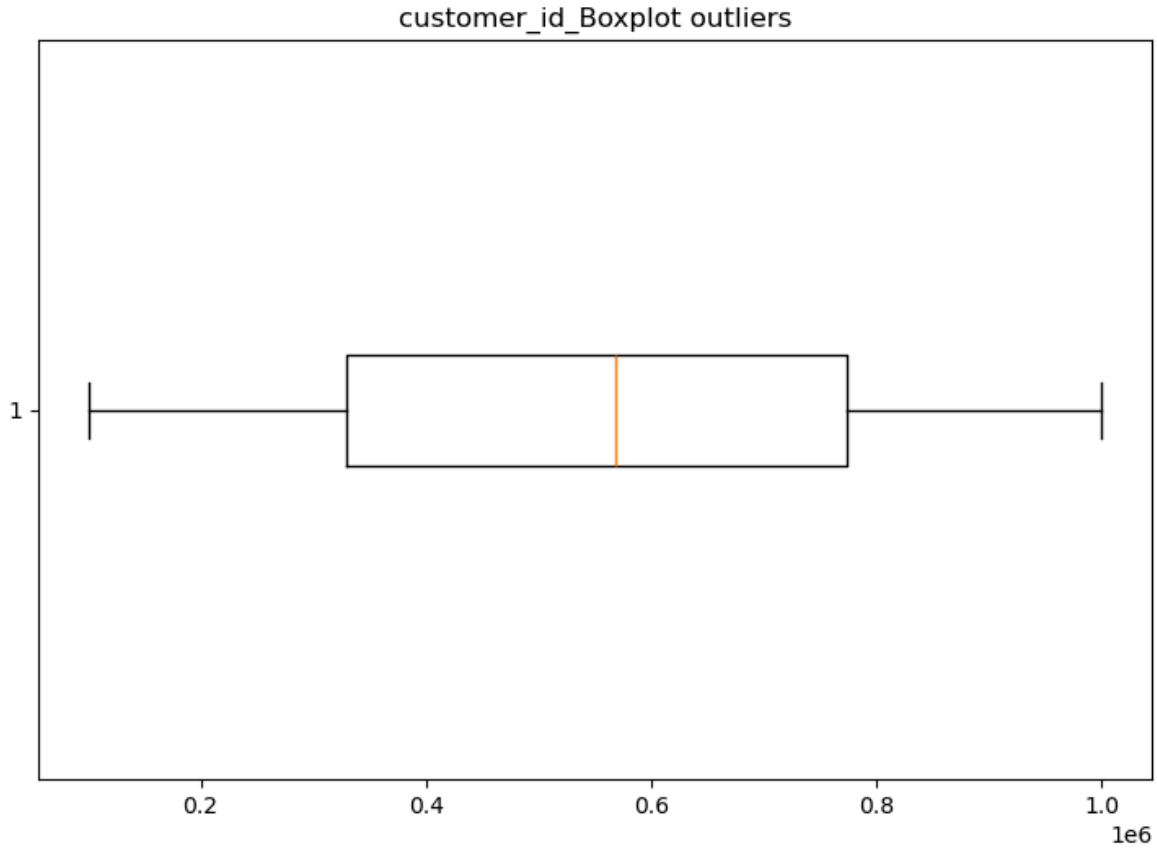
```
(9/200)*(100)
```

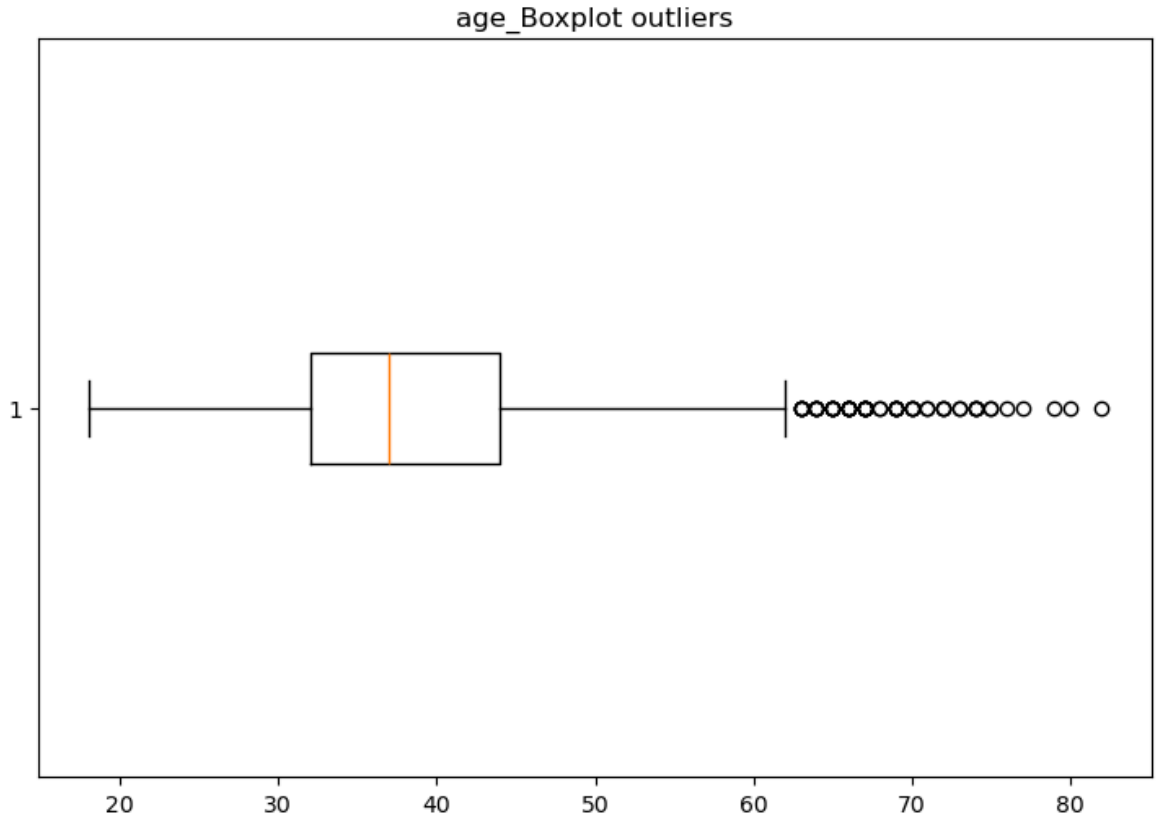
Out[147…

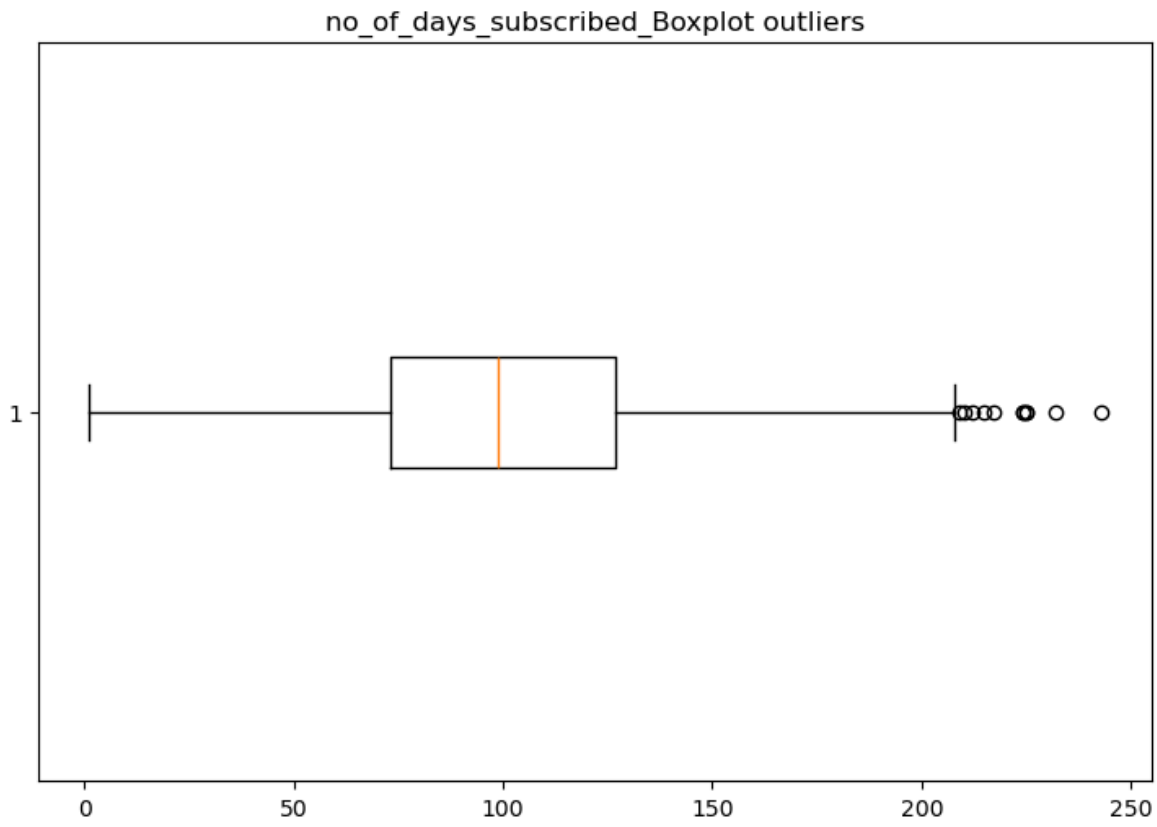
```
4.5
```

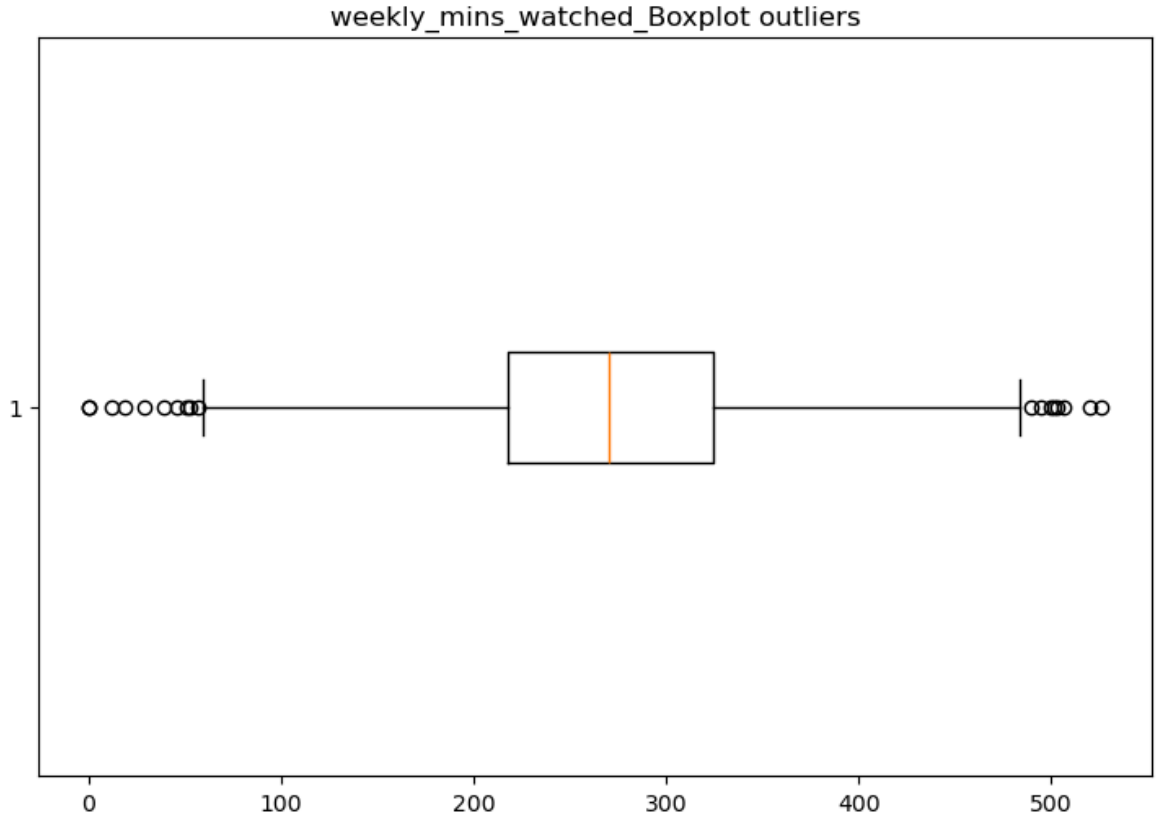
In [39]:

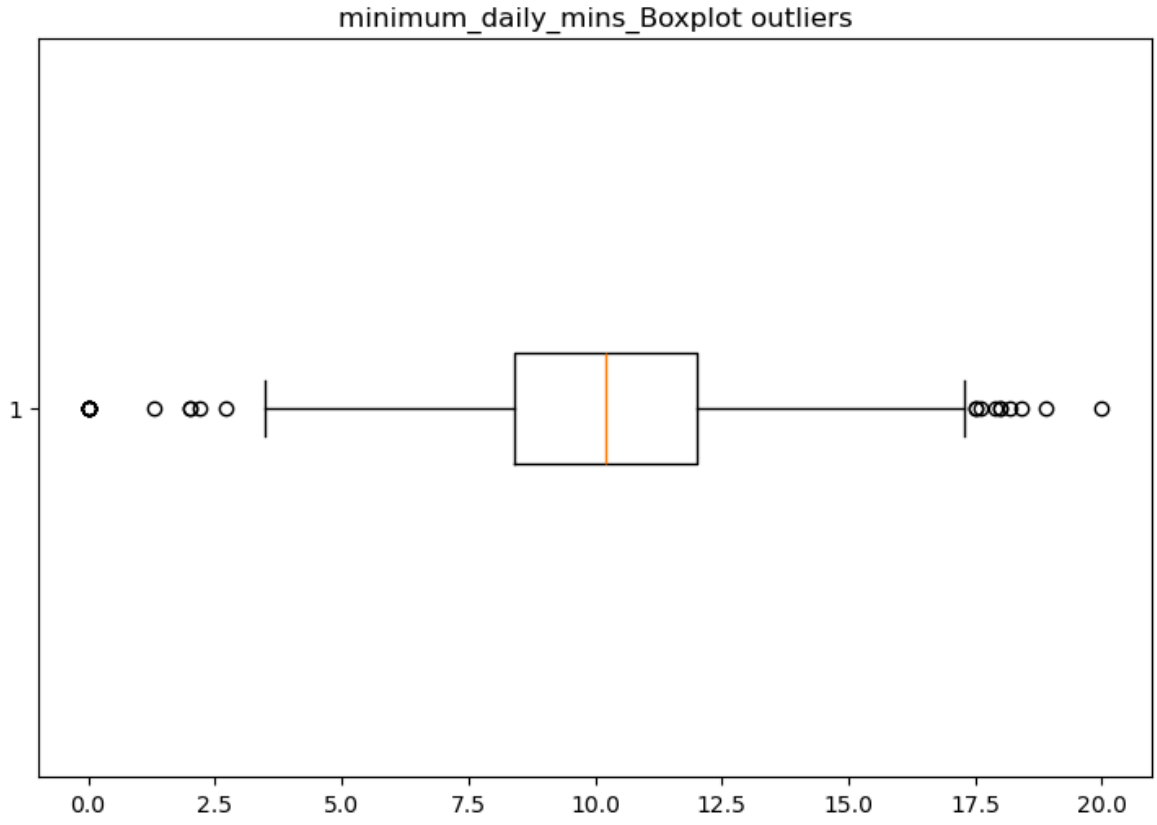
```
for i in range(1,9):
    plt.figure(figsize=(30,20))
    plt.subplot(3,3,i)
    plt.boxplot(telecom_df[numerical[i]],vert=False)
    plt.title(f'{numerical[i]}_Boxplot outliers')
    plt.show()
```

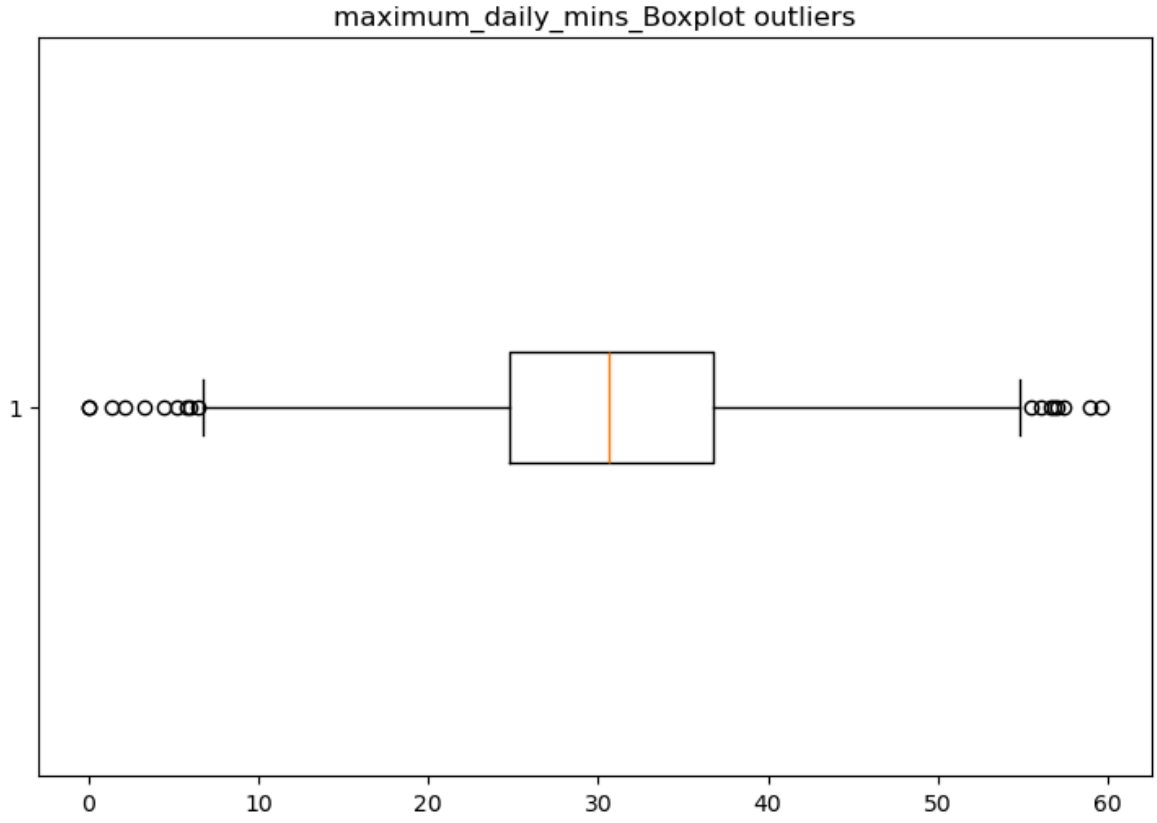

customer_id_Boxplot outliers
1
0.2
0.4
0.6
0.8
1.0
1e6

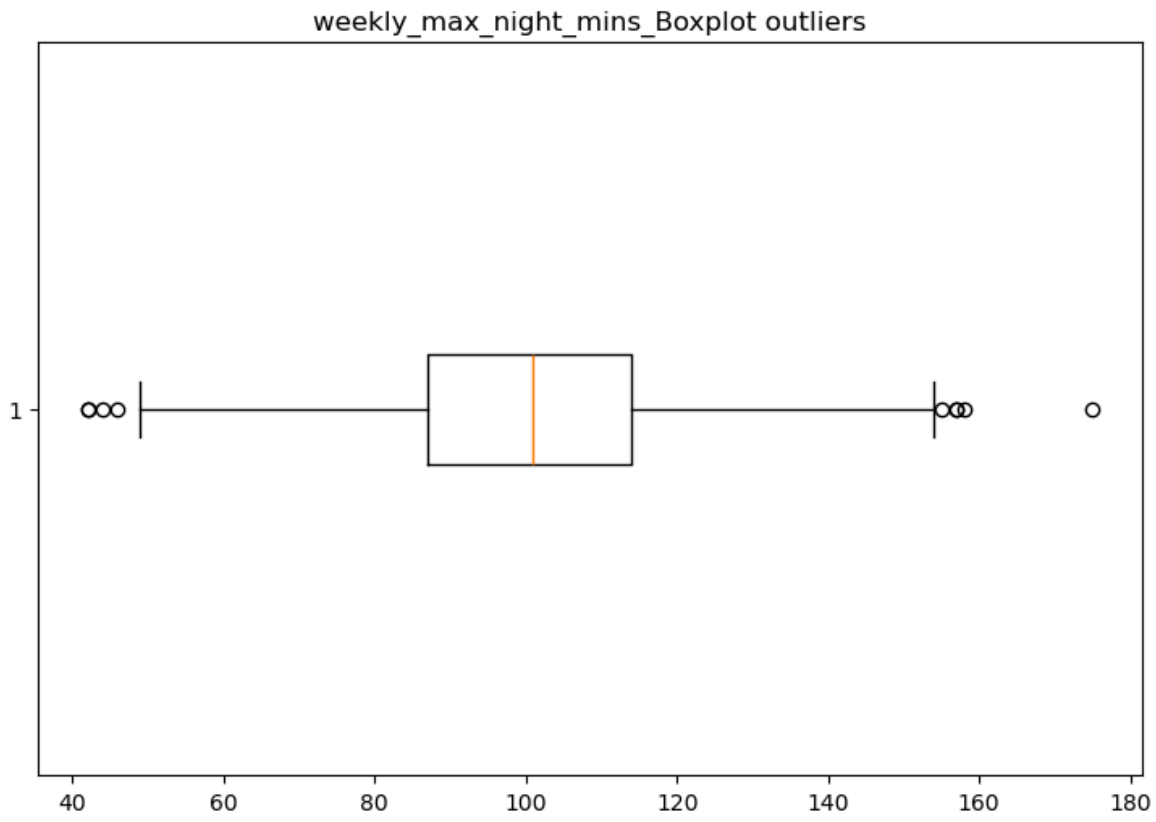

age_Boxplot outliers
1
20
30
40
50
60
70
80

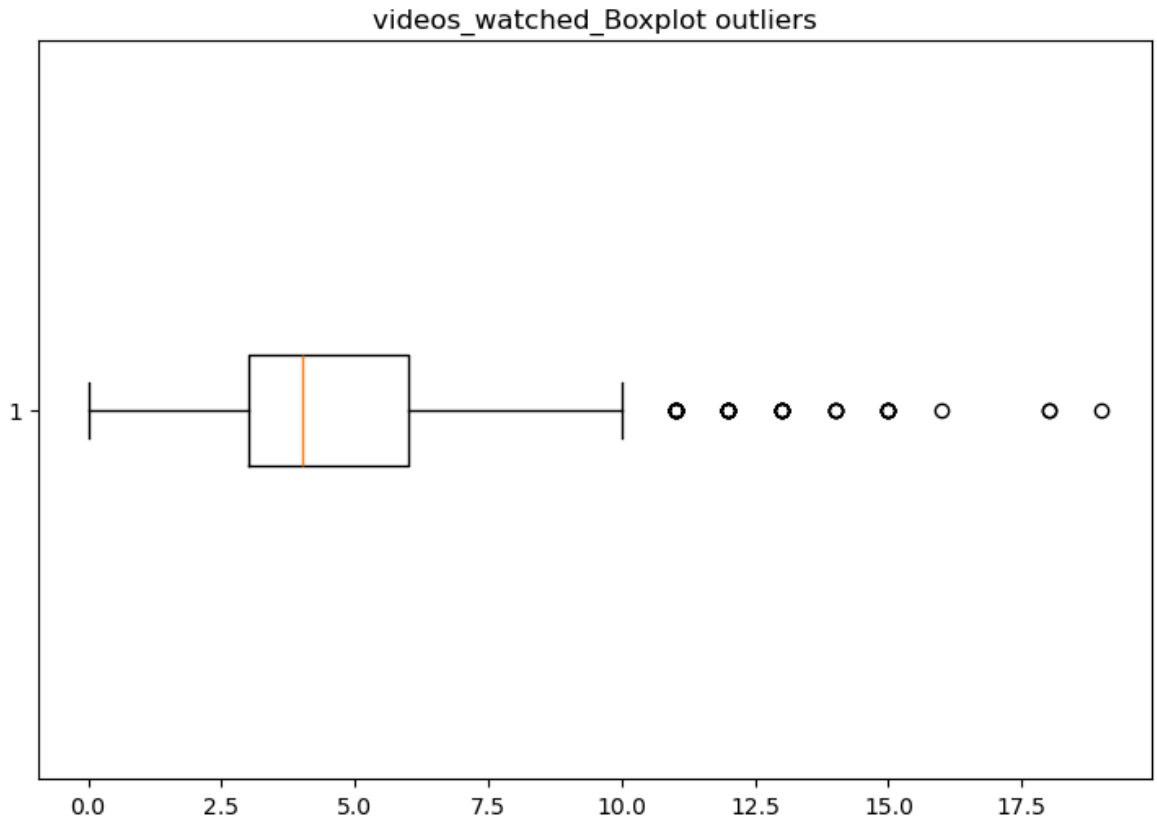

no_of_days_subscribed_Boxplot outliers
1
0
50
100
150
200
250


weekly_mins_watched_Boxplot outliers
1
0
100
200
300
400
500

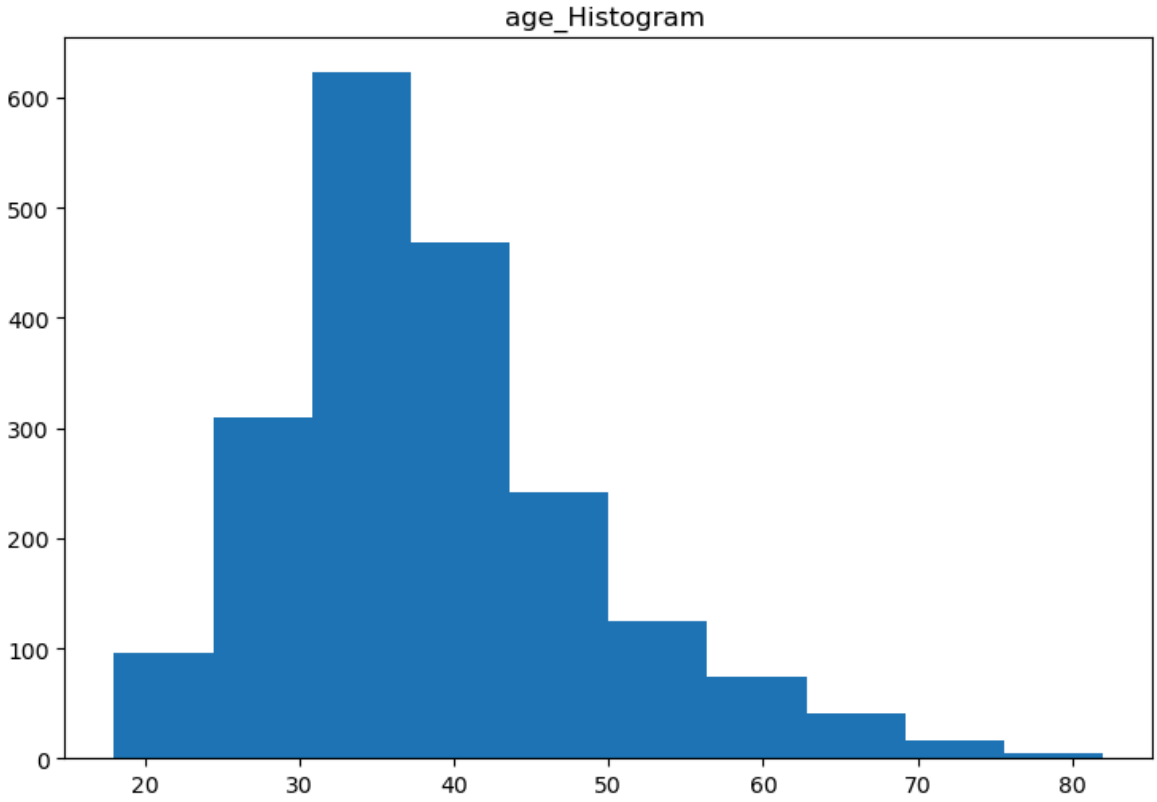

minimum_daily_mins_Boxplot outliers
1
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

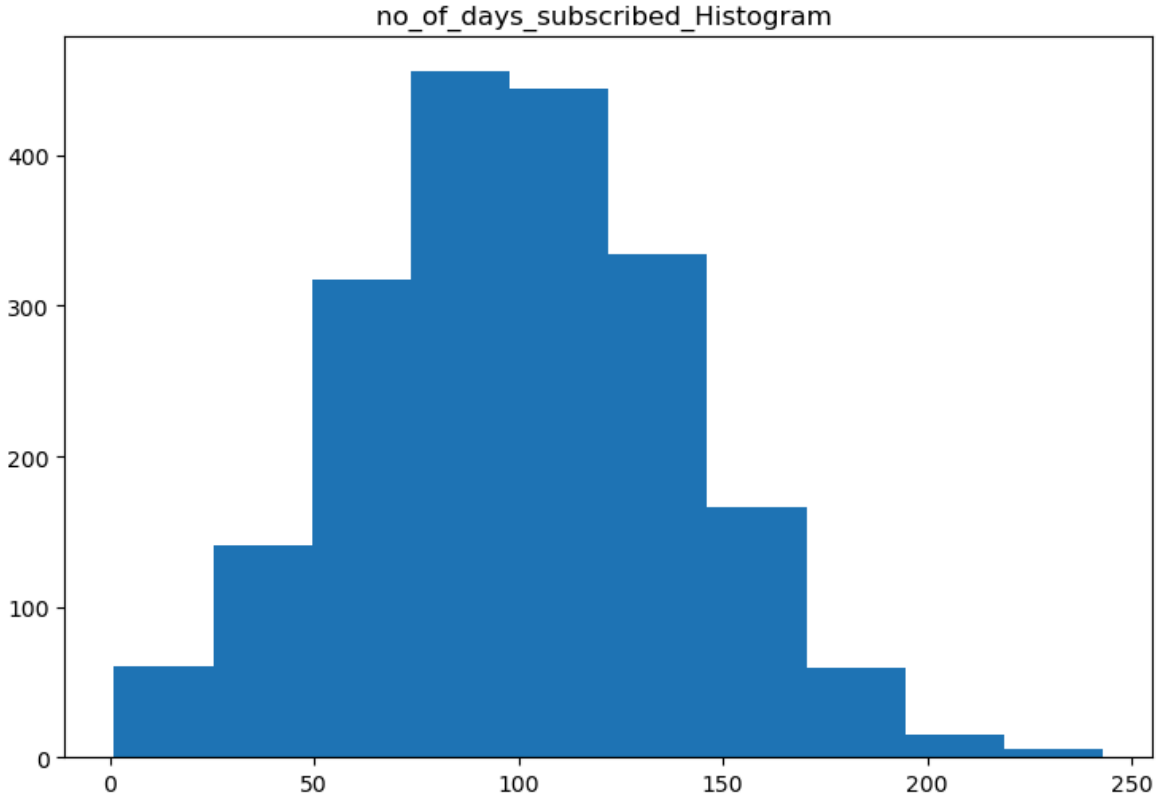

maximum_daily_mins_Boxplot outliers
1
0
10
20
30
40
50
60

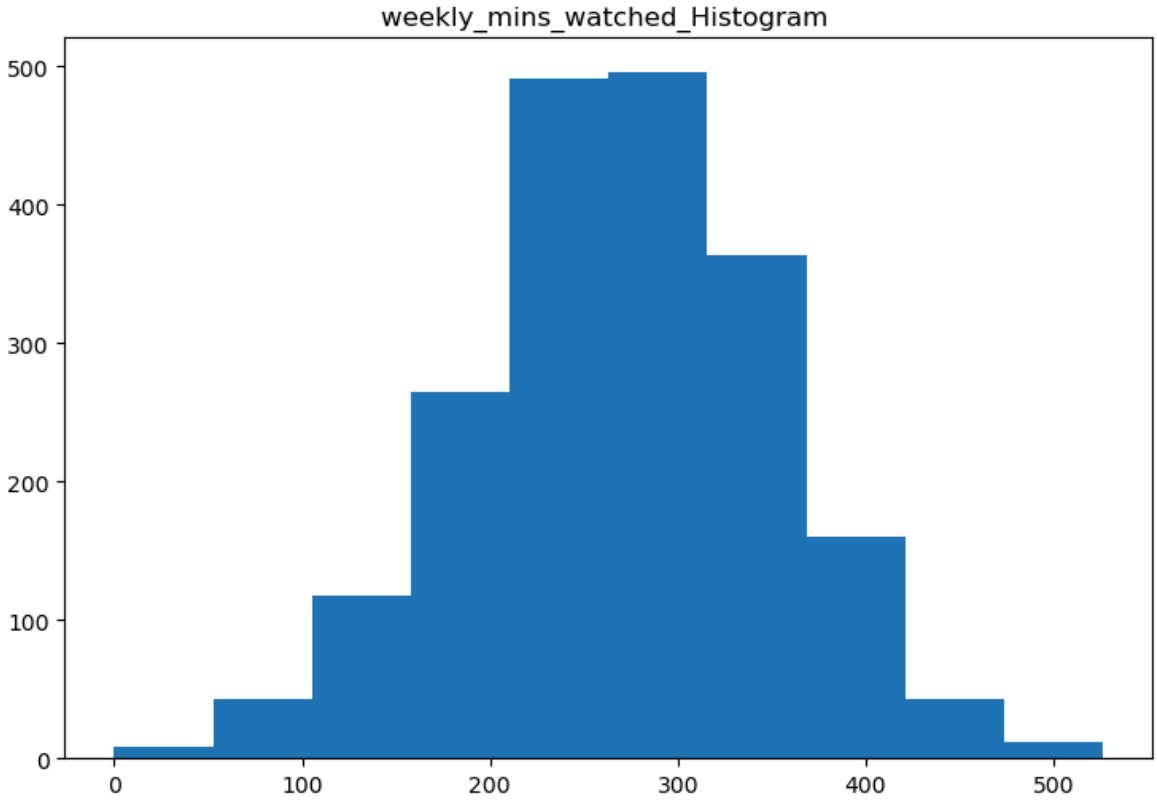


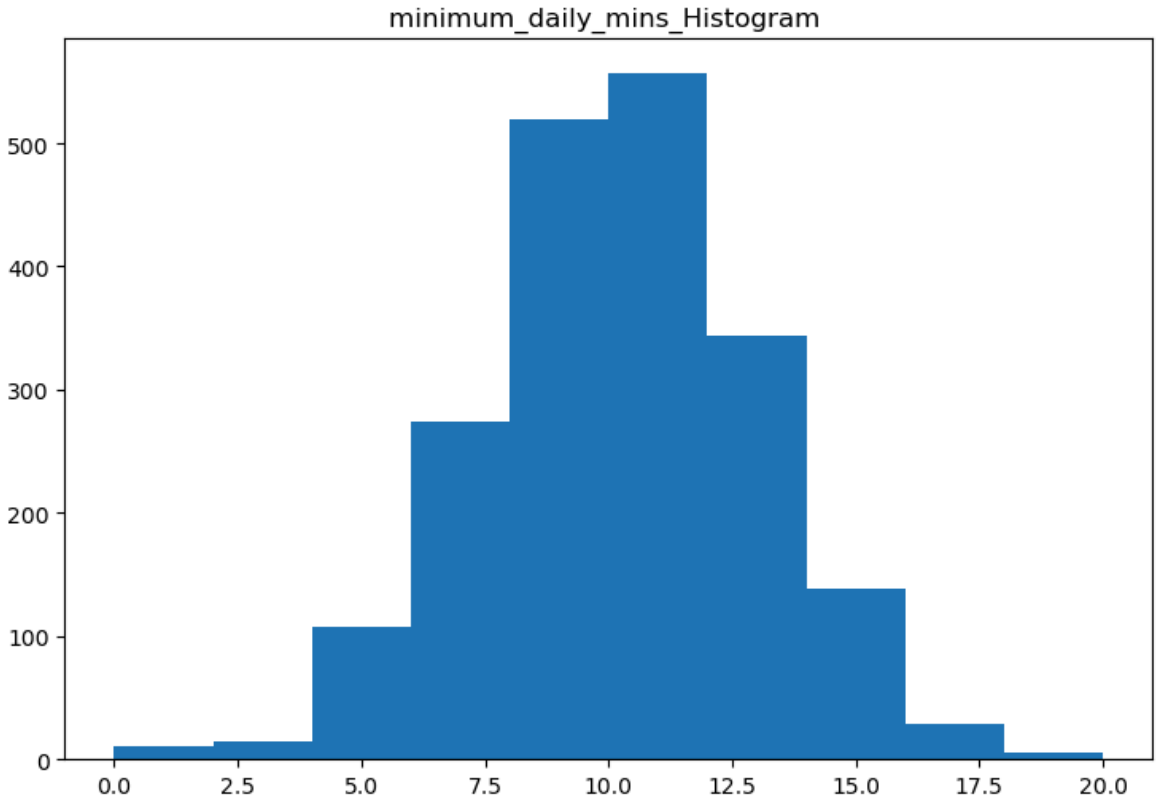


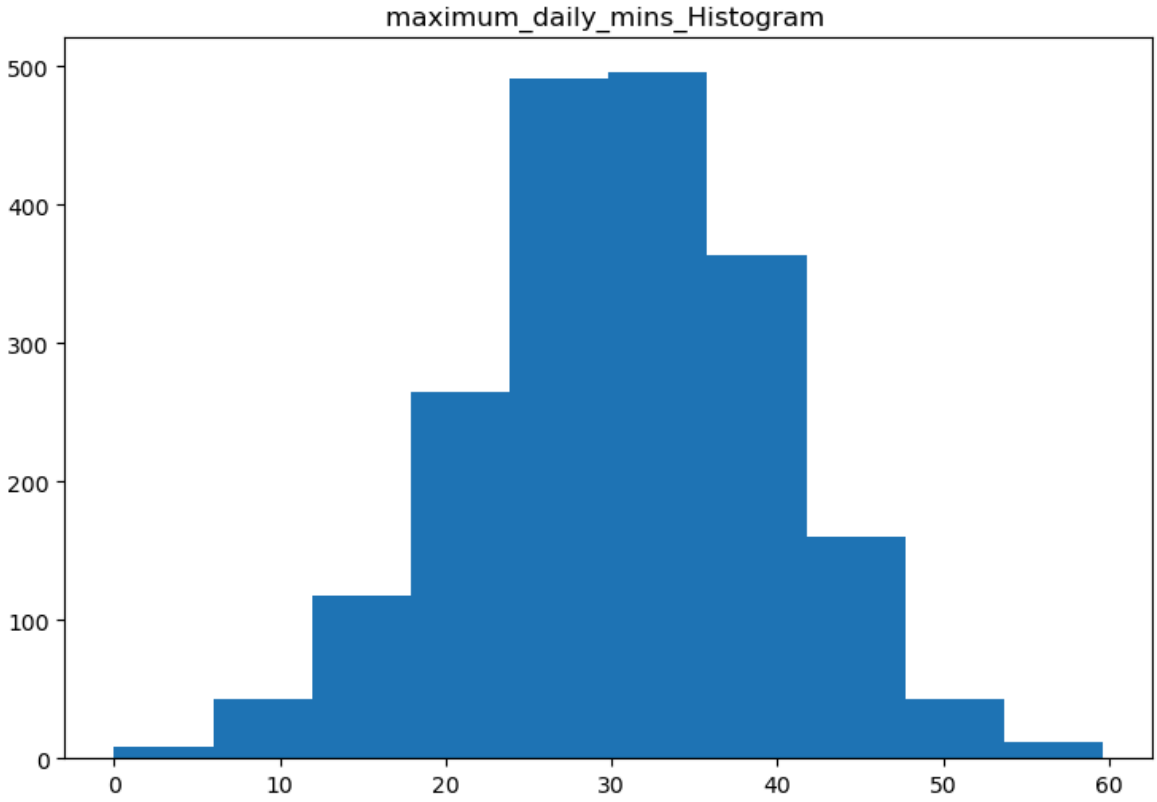
In [21]:
```
for i in range(2,9):
    plt.figure(figsize=(30,20))
    plt.subplot(3,3,i).hist(telecom_df[numerical[i]])
    plt.title(f'{numerical[i]}_Histogram ')
    plt.show()
```

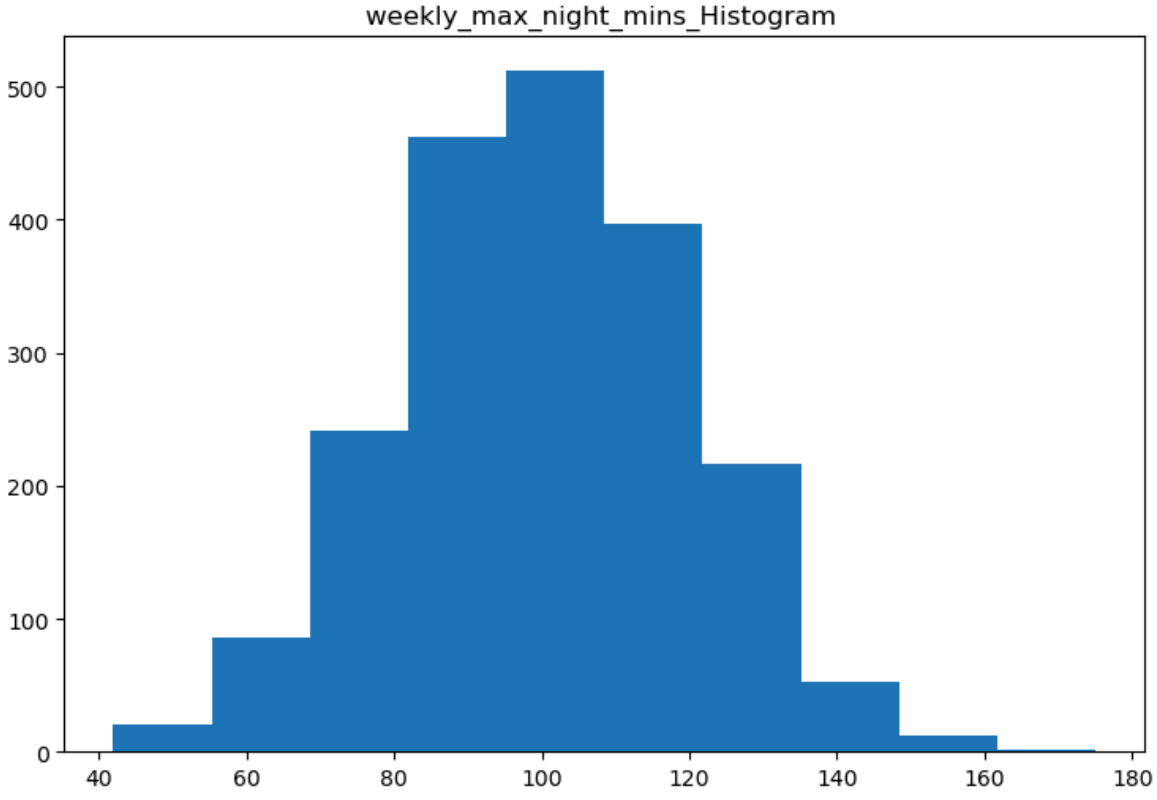

age_Histogram
600
500
400
300
200
100
0
20
30
40
50
60
70
80

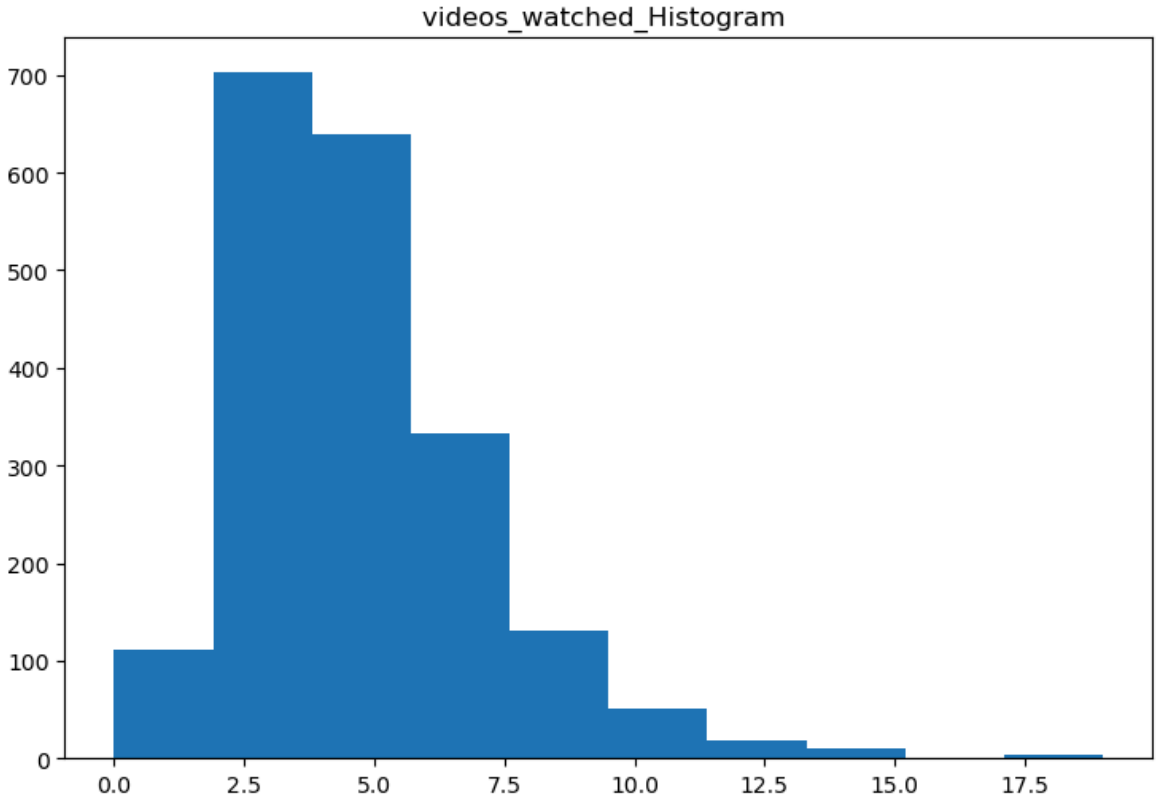

no_of_days_subscribed_Histogram
400
300
200
100
0
0
50
100
150
200
250


weekly_mins_watched_Histogram
500
400
300
200
100
0
0
100
200
300
400
500


minimum_daily_mins_Histogram
500
400
300
200
100
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


maximum_daily_mins_Histogram
500
400
300
200
100
0
0
10
20
30
40
50
60


weekly_max_night_mins_Histogram
500
400
300
200
100
0
40
60
80
100
120
140
160
180

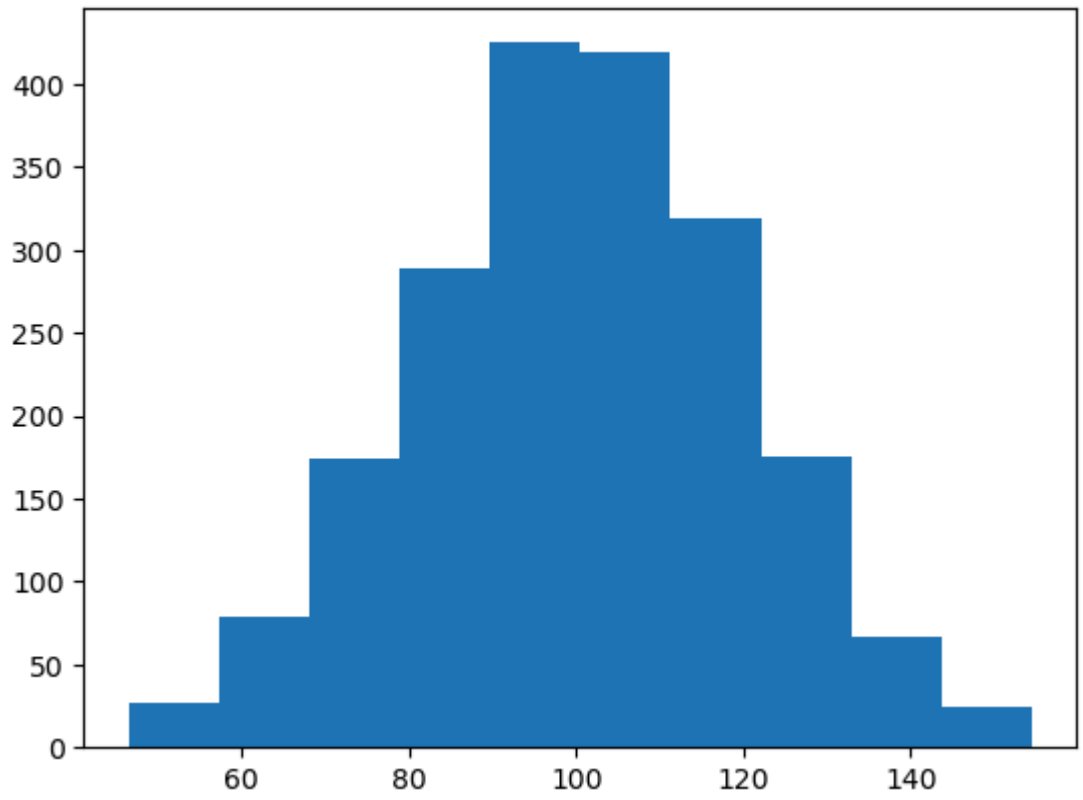


In [213…
```
np.size(l)
```

Out[213…
```
2000
```

In [215…
```
plt.hist(l)
```

Out[215…
```
(array([ 27.,  79., 174., 289., 425., 419., 319., 176.,  67.,  25.]),
 array([ 46.5,  57.3,  68.1,  78.9,  89.7, 100.5, 111.3, 122.1, 132.9,
        143.7, 154.5]),
 <BarContainer object of 10 artists>)
```



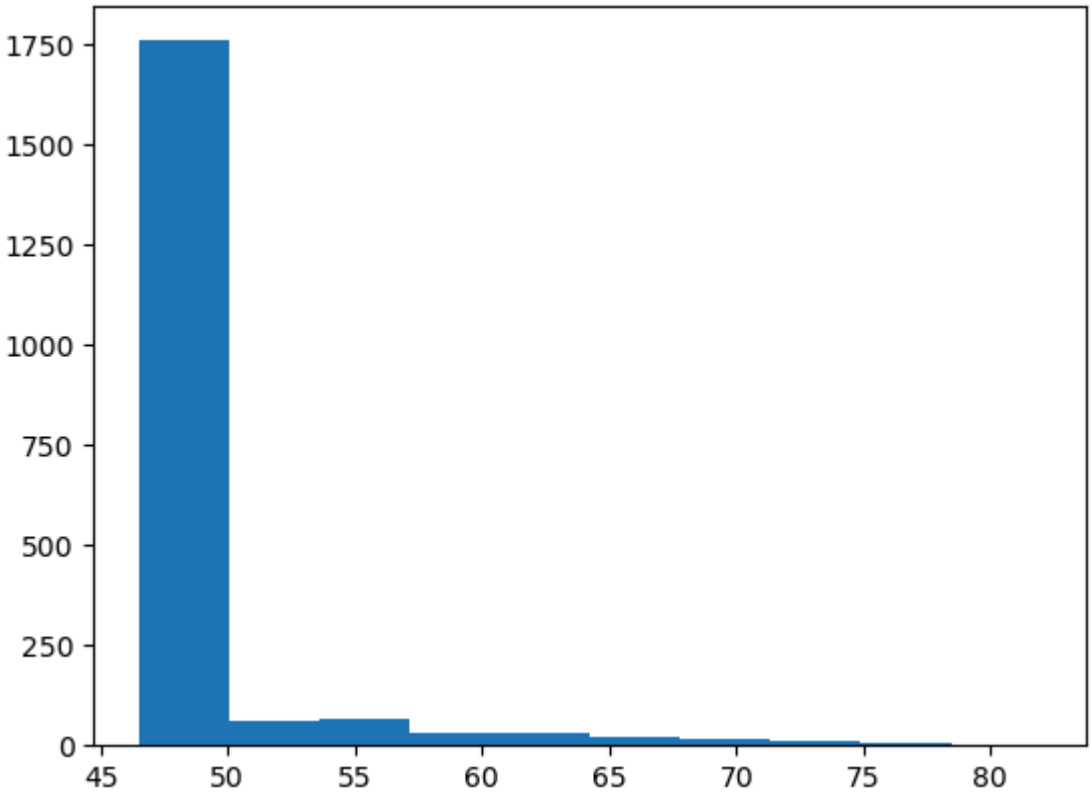
In [217…
```
l2=[]

for i in telecom_df['age']:
    if i < lb:
        l2.append(lb)
    elif i>ub:
        l2.append(ub)
    else:
        l2.append(i)

telecom_df['multi_screen']=l2
```

In [219…
```
len(l),len(l2)
```

Out[219… (2000, 2000)

In [225…
```
plt.hist(l2)
```

Out[225…
```
(array([1759.,   60.,   64.,   31.,   33.,   23.,   14.,    9.,    4.,
          3.]),
 array([46.5 , 50.05, 53.6 , 57.15, 60.7 , 64.25, 67.8 , 71.35, 74.9 ,
        78.45, 82.  ]),
 <BarContainer object of 10 artists>)
```



In [229…
```
id_col=telecom_df['gender']
cols=telecom_df['multi_screen']

pd.crosstab(id_col,cols)
```

Out[229…

multi_screen	46.5	47.0	48.0	49.0	50.0	51.0	52.0	53.0	54.0	55.0	...	70.0	71.0
gender													
Female	759	15	23	15	7	11	8	6	9	4	...	3	1
Male	863	12	15	18	13	10	17	7	9	7	...	1	1

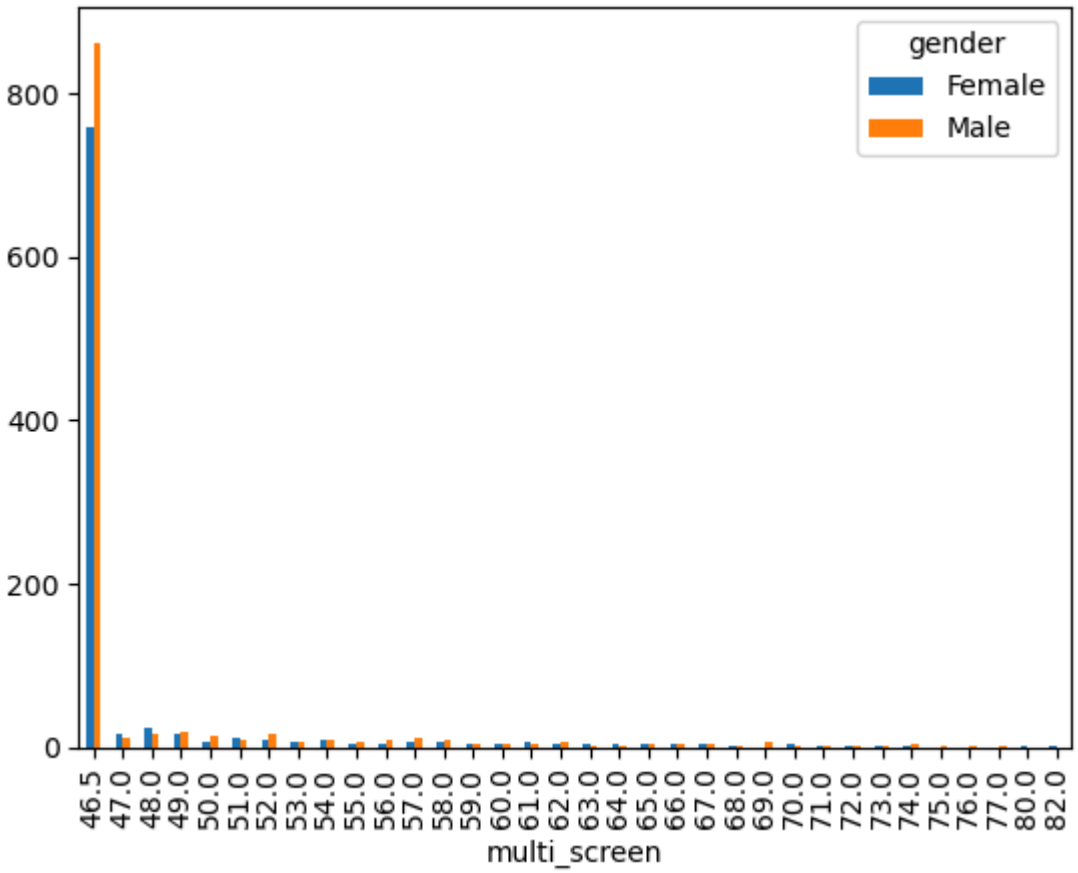
2 rows × 34 columns

In [231…

```
r1=pd.crosstab(cols,id_col)
r1.plot(kind='bar')
```

Out[231…

```
<Axes: xlabel='multi_screen'>
```

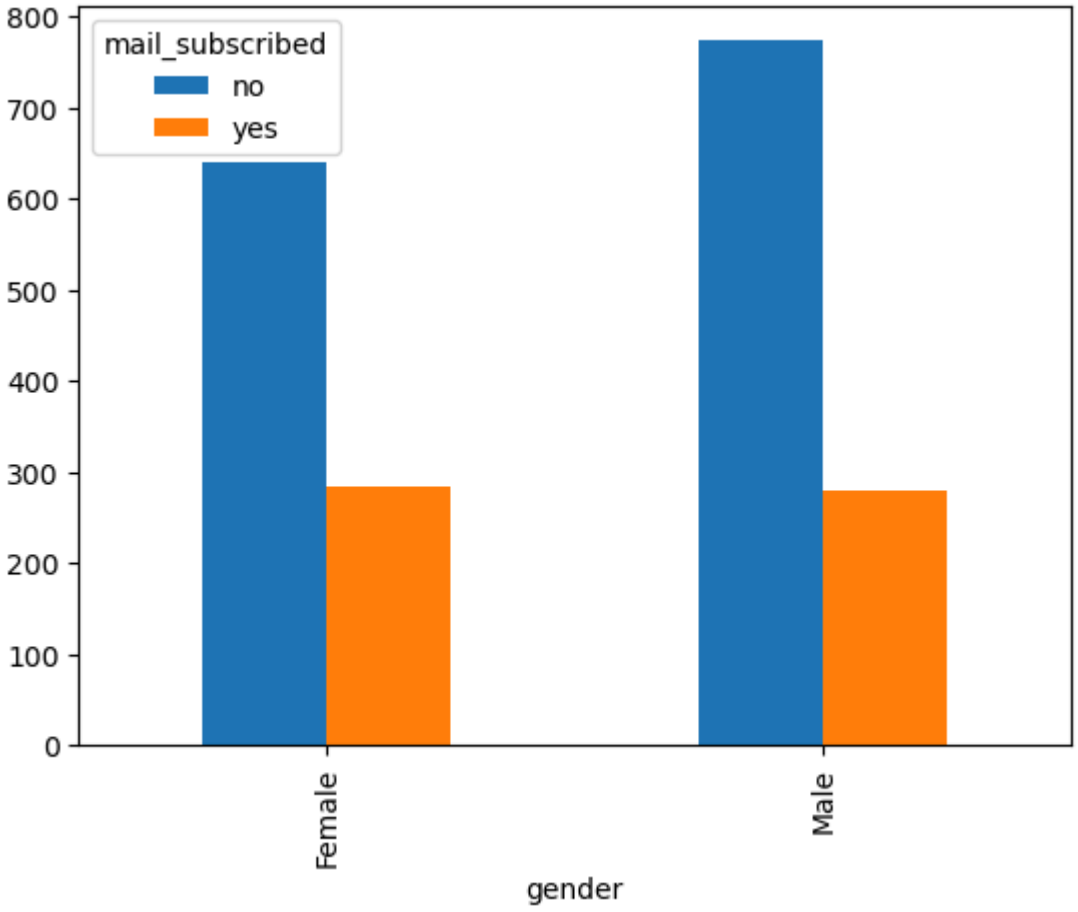


In [235…

```
id_col=telecom_df['gender']
cols=telecom_df['mail_subscribed']

r2=pd.crosstab(id_col,cols)

r2.plot(kind='bar')
```

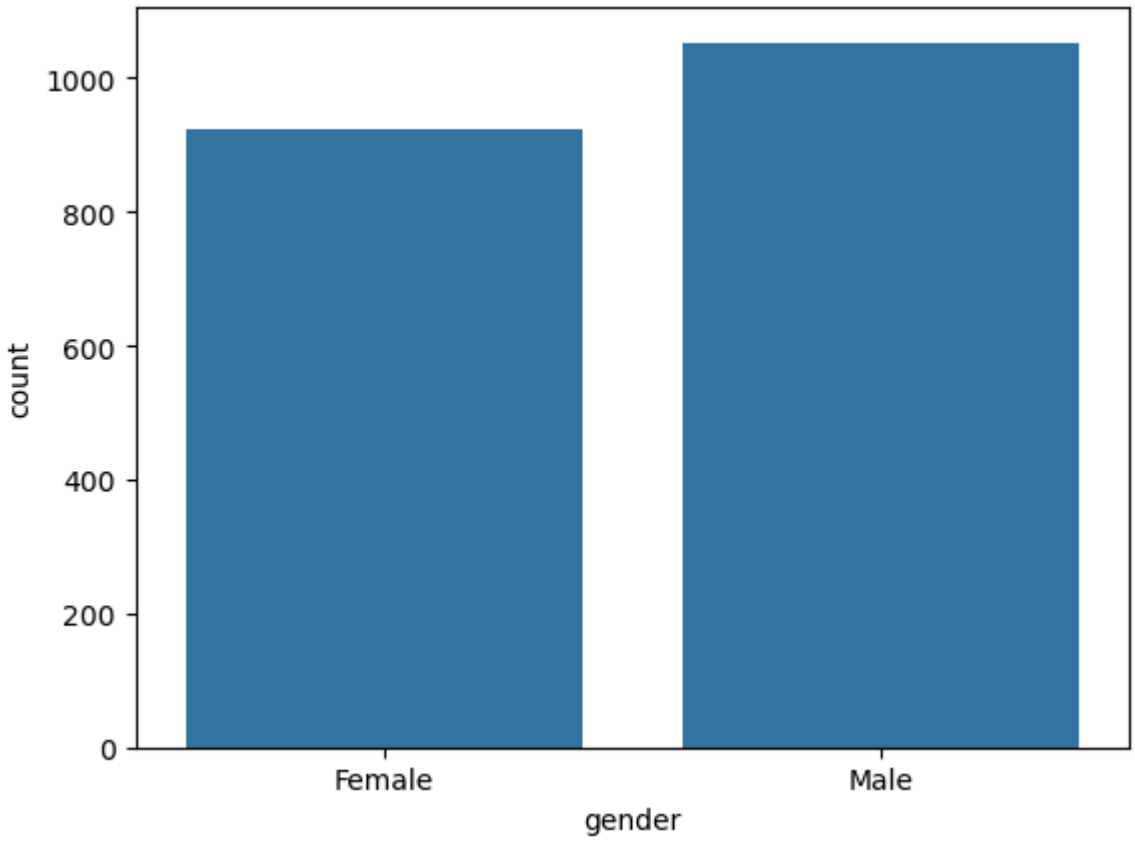
Out[235…

```
<Axes: xlabel='gender'>
```



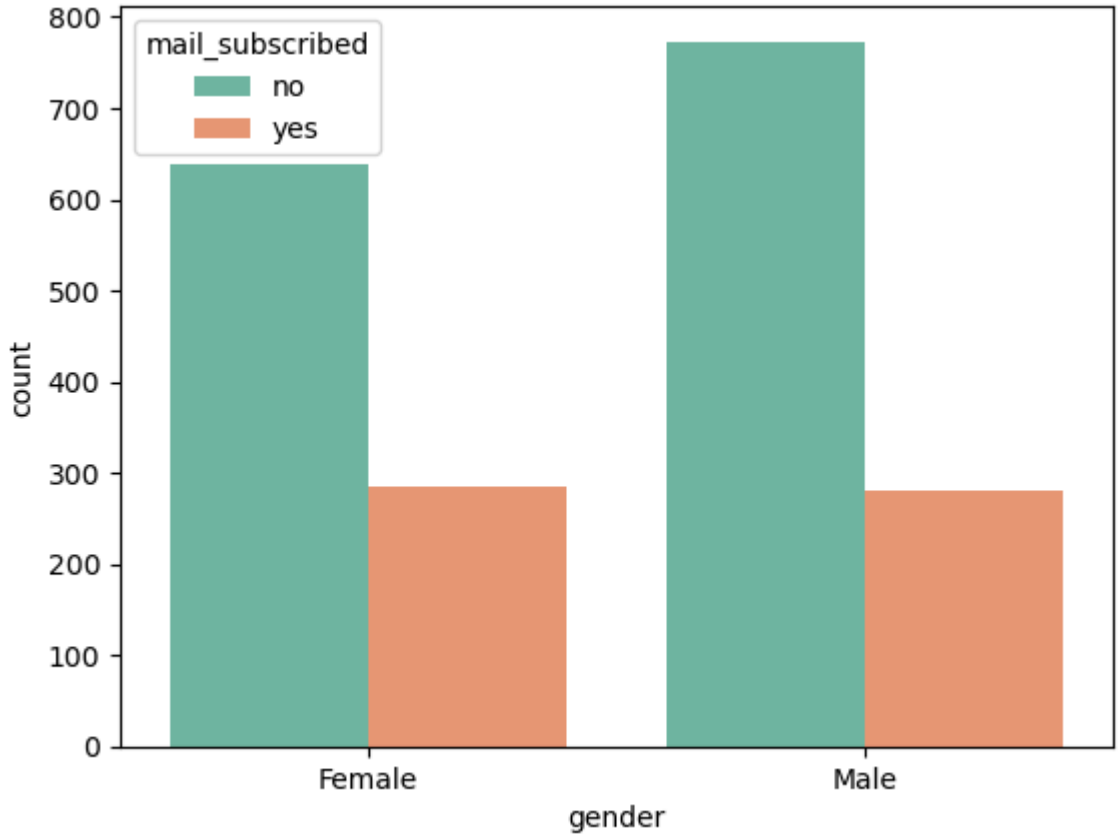
In [239…
```
sns.countplot(data=telecom_df,
              x='gender')
```

Out[239…
```
<Axes: xlabel='gender', ylabel='count'>
```



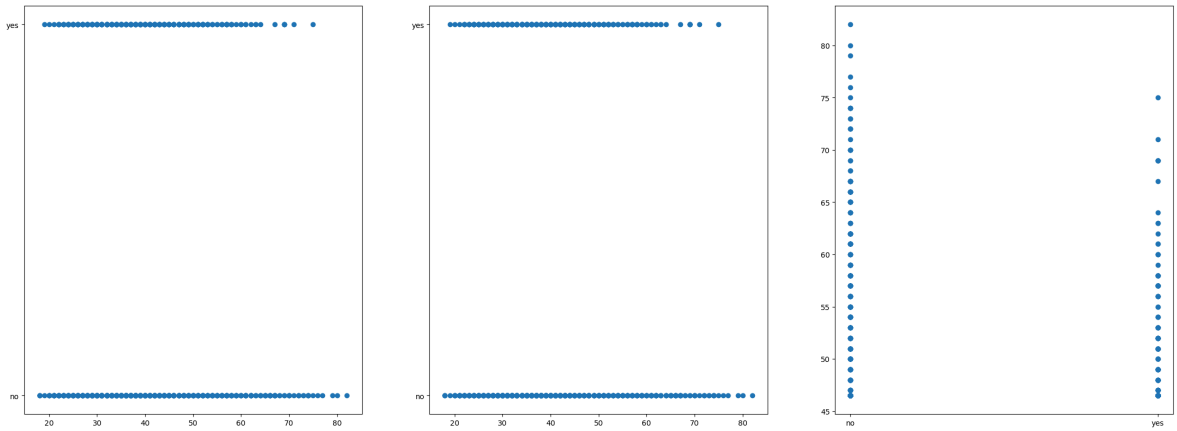
In [243…
```
sns.countplot(data=telecom_df,
              x='gender',hue='mail_subscribed',palette="Set2",width=0.8)
```

Out[243…
```
<Axes: xlabel='gender', ylabel='count'>
```



In [247…
```
plt.figure(figsize=(28,10))
col1=telecom_df['age']
col2=telecom_df['mail_subscribed']
col3=telecom_df['multi_screen']
plt.subplot(1,3,1).scatter(col1,col2)
plt.subplot(1,3,2).scatter(col1,col2)
plt.subplot(1,3,3).scatter(col2,col3)
```

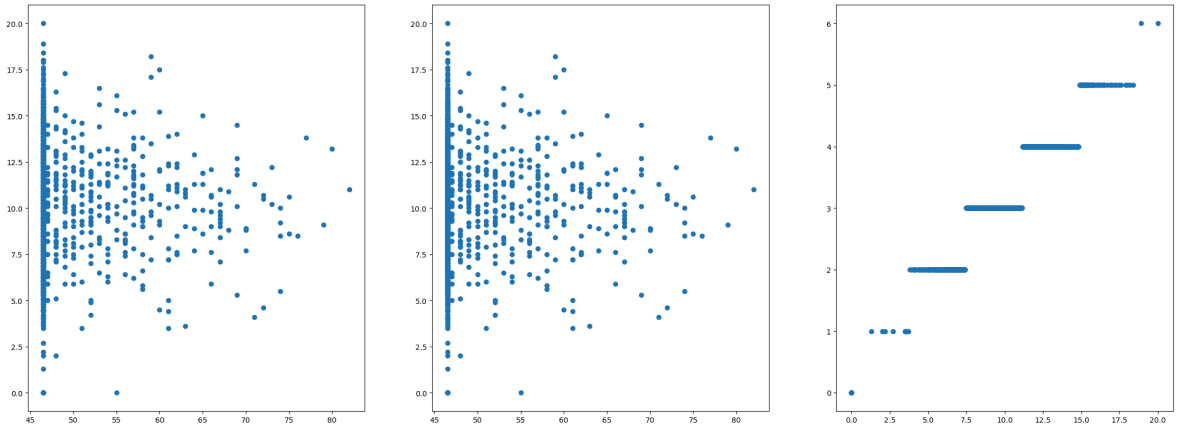
Out[247…
```
<matplotlib.collections.PathCollection at 0x1e584c46180>
```



In [249…
```
plt.figure(figsize=(28,10))
col1=telecom_df['multi_screen']
col2=telecom_df['minimum_daily_mins']
col3=telecom_df['maximum_days_inactive']
plt.subplot(1,3,1).scatter(col1,col2)
```

```
plt.subplot(1,3,2).scatter(col1,col2)
plt.subplot(1,3,3).scatter(col2,col3)
```

Out[249… `<matplotlib.collections.PathCollection at 0x1e5867cbe00>`



In [74]:
```
telecom_df.corr(numeric_only=True)
```

Out[74]:

	year	customer_id	age	no_of_days_subscribed	weekly_mi
year	NaN	NaN	NaN	NaN	
customer_id	NaN	1.000000	0.023830	-0.011457	
age	NaN	0.023830	1.000000	0.038822	
no_of_days_subscribed	NaN	-0.011457	0.038822	1.000000	
weekly_mins_watched	NaN	-0.013830	0.016316	0.000657	
minimum_daily_mins	NaN	0.033733	-0.000676	0.016329	
maximum_daily_mins	NaN	-0.013834	0.016328	0.000652	
weekly_max_night_mins	NaN	0.009713	0.016645	-0.001967	
videos_watched	NaN	0.063967	0.013076	0.019414	
maximum_days_inactive	NaN	0.045145	0.006690	0.017961	
customer_support_calls	NaN	-0.036009	0.000618	0.013419	
churn	NaN	-0.056777	0.011631	0.002528	

In [76]:
```
telecom_df.corr(method='pearson',numeric_only=True)
```

Out[76]:

	year	customer_id	age	no_of_days_subscribed	weekly_mi
year	NaN	NaN	NaN	NaN	
customer_id	NaN	1.000000	0.023830	-0.011457	
age	NaN	0.023830	1.000000	0.038822	
no_of_days_subscribed	NaN	-0.011457	0.038822	1.000000	
weekly_mins_watched	NaN	-0.013830	0.016316	0.000657	
minimum_daily_mins	NaN	0.033733	-0.000676	0.016329	
maximum_daily_mins	NaN	-0.013834	0.016328	0.000652	
weekly_max_night_mins	NaN	0.009713	0.016645	-0.001967	
videos_watched	NaN	0.063967	0.013076	0.019414	
maximum_days_inactive	NaN	0.045145	0.006690	0.017961	
customer_support_calls	NaN	-0.036009	0.000618	0.013419	
churn	NaN	-0.056777	0.011631	0.002528	

In [ ]: