In [1]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [92]:

```python
df= pd.read_excel ("D:\\assignment\\facebook_user_data.xlsx")
```

In [93]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99003 entries, 0 to 99002
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   userid                 99003 non-null  int64  
 1   age                    99003 non-null  int64  
 2   dob_day                99003 non-null  int64  
 3   dob_year               99003 non-null  int64  
 4   dob_month              99003 non-null  int64  
 5   gender                 98828 non-null  object 
 6   tenure                 99001 non-null  float64
 7   friend_count           99003 non-null  int64  
 8   friendships_initiated  99003 non-null  int64  
 9   likes                  99003 non-null  int64  
 10  likes_received         99003 non-null  int64  
 11  mobile_likes           99003 non-null  int64  
 12  mobile_likes_received  99003 non-null  int64  
 13  www_likes              99003 non-null  int64  
 14  www_likes_received     99003 non-null  int64  
dtypes: float64(1), int64(13), object(1)
memory usage: 11.3+ MB
```

In [4]:

```python
df.head(5)
```

Out[4]:

	userid	age	dob_day	dob_year	dob_month	gender	tenure	friend_count	friendships_initiated	likes	likes_received	mobile_likes	mobile_likes_received	www_likes	www_likes_received
0	2094382	14	19	1999	11	male	266.0	0	0	0	0	0	0	0	0
1	1192601	14	2	1999	11	female	6.0	0	0	0	0	0	0	0	0
2	2083884	14	16	1999	11	male	13.0	0	0	0	0	0	0	0	0
3	1203168	14	25	1999	12	female	93.0	0	0	0	0	0	0	0	0
4	1733186	14	4	1999	12	male	82.0	0	0	0	0	0	0	0	0

Imputation of missing values:

In [5]:

```python
df.columns
```

Out[5]:

```
Index(['userid', 'age', 'dob_day', 'dob_year', 'dob_month', 'gender', 'tenure',
       'friend_count', 'friendships_initiated', 'likes', 'likes_received',
       'mobile_likes', 'mobile_likes_received', 'www_likes',
       'www_likes_received'],
      dtype='object')
```

In [6]:

```python
df.isna().sum()
```

Out[6]:

```
userid                     0
age                        0
dob_day                    0
dob_year                   0
dob_month                  0
gender                   175
tenure                     2
friend_count               0
friendships_initiated      0
likes                      0
likes_received             0
mobile_likes               0
mobile_likes_received      0
www_likes                  0
www_likes_received         0
dtype: int64
```

Replace the null values (NA) of gender column with its mode or median and explain why mode/median used to replace NA values

In [19]:

```python
df.duplicated().sum()
```

Out[19]: 0

In [8]:

```python
df['gender'].fillna(df['gender'].mode,inplace= True)
```

ANS- Its a categorical variable so i used mode.

In [9]:

```python
df.isna().sum()
```

Out[9]:

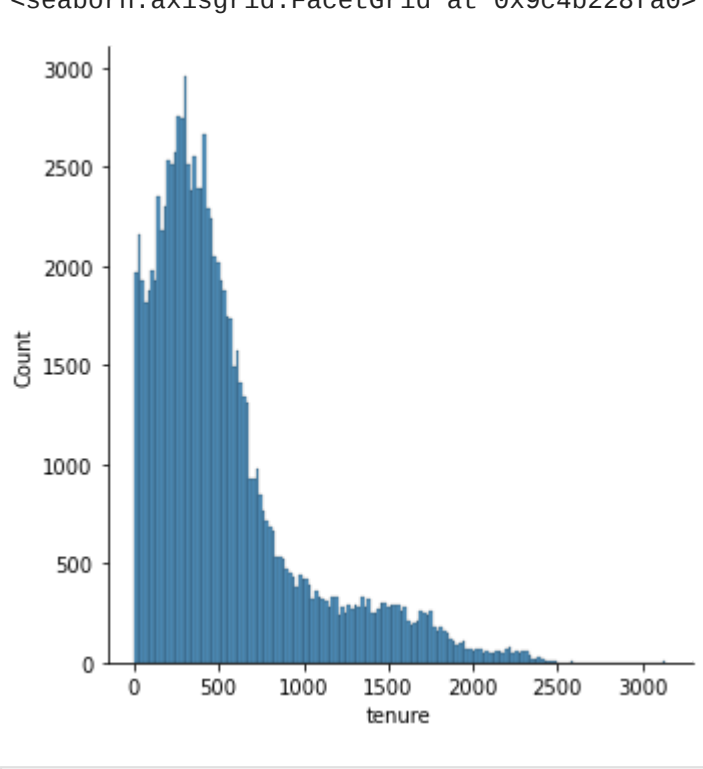
```
userid                   0
age                      0
dob_day                  0
dob_year                 0
dob_month                0
gender                   0
tenure                   2
friend_count             0
friendships_initiated    0
likes                    0
likes_received           0
mobile_likes             0
mobile_likes_received    0
www_likes                0
www_likes_received       0
dtype: int64
```

Replace the null values (NA) of tenure column (numerical variable) with its median,and explain why mode/median used to replace NA values

In [10]:

```python
sns.displot(df.tenure)
```

Out[10]: <seaborn.axisgrid.FacetGrid at 0x9c4b228fa0>



In [11]:

```python
df['tenure'].fillna(df['tenure'].median,inplace= True)
```

When the data is skewed, it is good to consider using the median value for replacing the missing values.

In [12]:

```python
df.isna().sum()
```

Out[12]:

```
userid                   0
age                      0
dob_day                  0
dob_year                 0
dob_month                0
gender                   0
tenure                   0
friend_count             0
friendships_initiated    0
likes                    0
likes_received           0
mobile_likes             0
mobile_likes_received    0
www_likes                0
www_likes_received       0
dtype: int64
```

2) Plot heatmap / correlation matrix on all the columns.

In [13]:

```python
df.corr()
```

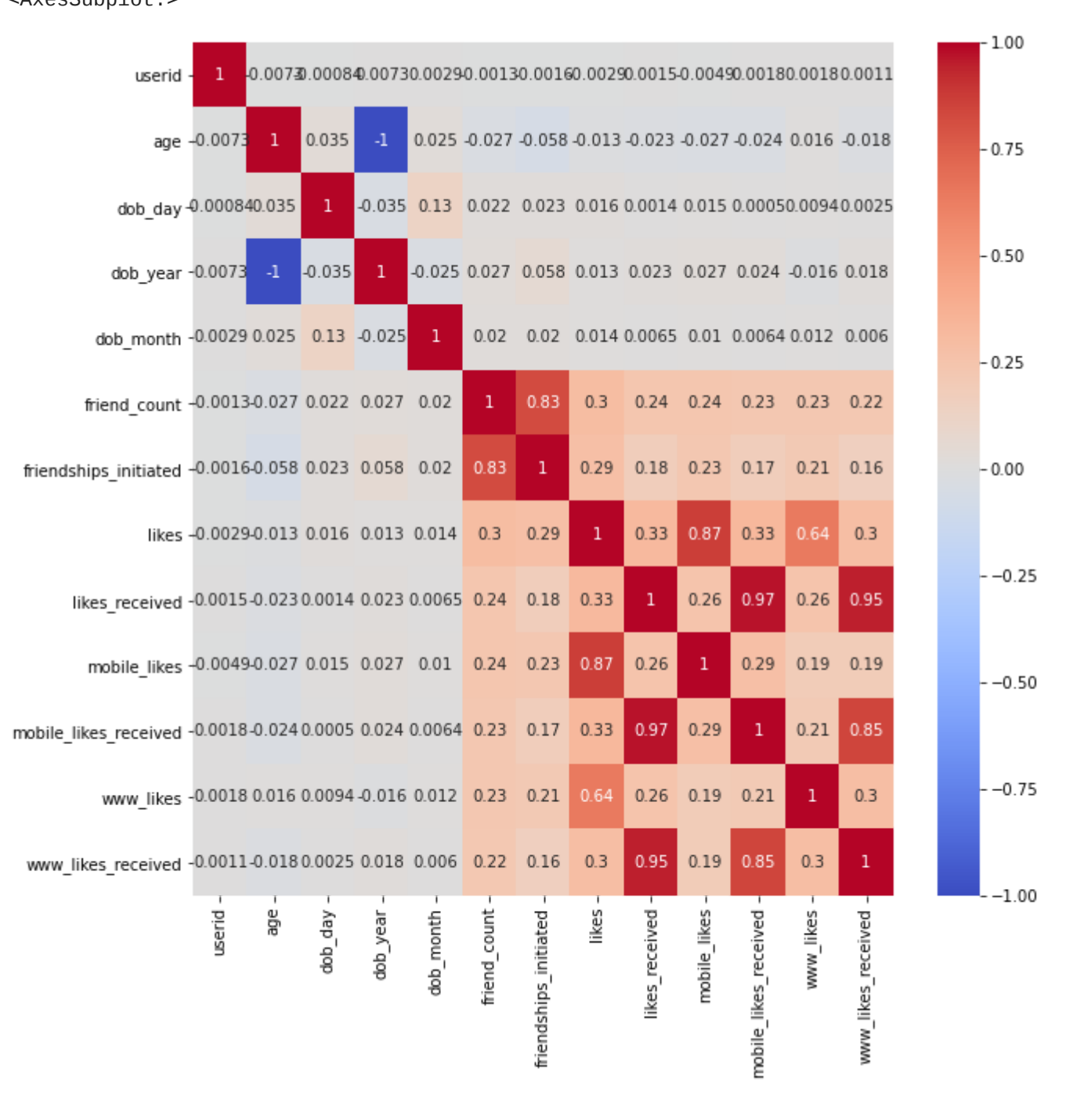
Out[13]:

	userid	age	dob_day	dob_year	dob_month	friend_count	friendships_initiated	likes	likes_received	mobile_likes	mobile_likes_received	www_likes	www_likes_re
userid	1.000000	-0.007265	-0.000839	0.007265	0.002924	-0.001314	-0.001591	-0.002875	0.001526	-0.004868	0.001753	0.001828	0.
age	-0.007265	1.000000	0.035035	-1.000000	0.025167	-0.027407	-0.058059	-0.013009	-0.022570	-0.026715	-0.024248	0.015585	-0.
dob_day	-0.000839	0.035035	1.000000	-0.035035	0.129443	0.021961	0.022999	0.015980	0.001367	0.014541	0.000497	0.009353	0.
dob_year	0.007265	-1.000000	-0.035035	1.000000	-0.025167	0.027407	0.058059	0.013009	0.022570	0.026715	0.024248	-0.015585	0.
dob_month	0.002924	0.025167	0.129443	-0.025167	1.000000	0.019804	0.020075	0.014147	0.006495	0.010400	0.006435	0.012136	0.
friend_count	-0.001314	-0.027407	0.021961	0.027407	0.019804	1.000000	0.825850	0.298017	0.236463	0.235656	0.232701	0.229803	0.
friendships_initiated	-0.001591	-0.058059	0.022999	0.058059	0.020075	0.825850	1.000000	0.285592	0.175131	0.229808	0.173804	0.214022	0.
likes	-0.002875	-0.013009	0.015980	0.013009	0.014147	0.298017	0.285592	1.000000	0.327376	0.871652	0.329258	0.644960	0.
likes_received	0.001526	-0.022570	0.001367	0.022570	0.006495	0.236463	0.175131	0.327376	1.000000	0.256998	0.973679	0.255365	0.
mobile_likes	-0.004868	-0.026715	0.014541	0.026715	0.010400	0.235656	0.229808	0.871652	0.256998	1.000000	0.288513	0.187619	0.
mobile_likes_received	0.001753	-0.024248	0.000497	0.024248	0.006435	0.232701	0.173804	0.329258	0.973679	0.288513	1.000000	0.209997	0.
www_likes	0.001828	0.015585	0.009353	-0.015585	0.012136	0.229803	0.214022	0.644960	0.255365	0.187619	0.209997	1.000000	0.
www_likes_received	0.001074	-0.018224	0.002460	0.018224	0.006003	0.220727	0.161439	0.295687	0.947990	0.190173	0.850490	0.296053	1.

In [16]:

```python
plt.figure(figsize=(10,10))
sns.heatmap(df.corr(), annot =True, cmap="coolwarm")
```

Out[16]: <AxesSubplot:>



3) Analysis based on gender of the users

- What is composition of male and female users?

In [112...

```python
df['gender'].value_counts()
```

Out[112...

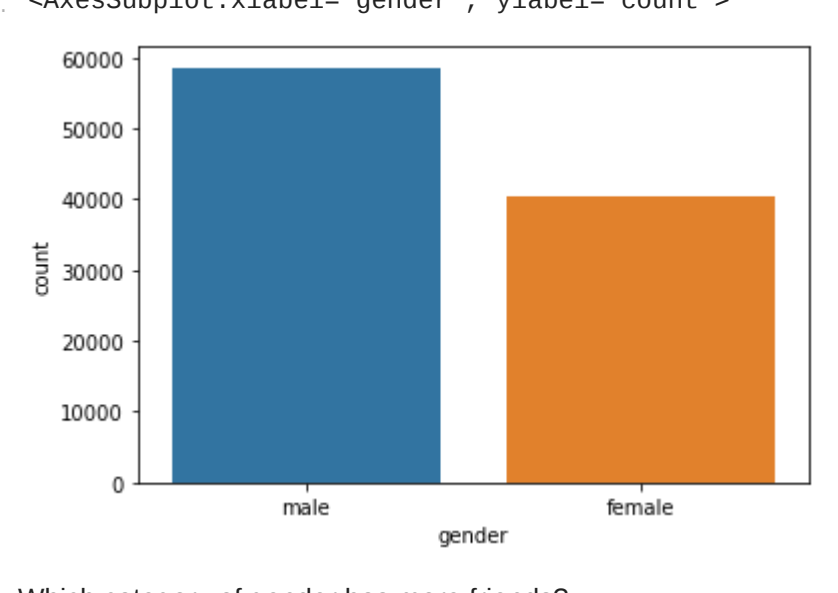
```
male      58574
female    40254
Name: gender, dtype: int64
```

In [111...

```python
sns.countplot(df['gender'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[111... <AxesSubplot:xlabel='gender', ylabel='count'>



Which category of gender has more friends?

In [113...

```python
df.groupby('gender').agg({'friend_count': ['sum']})
```

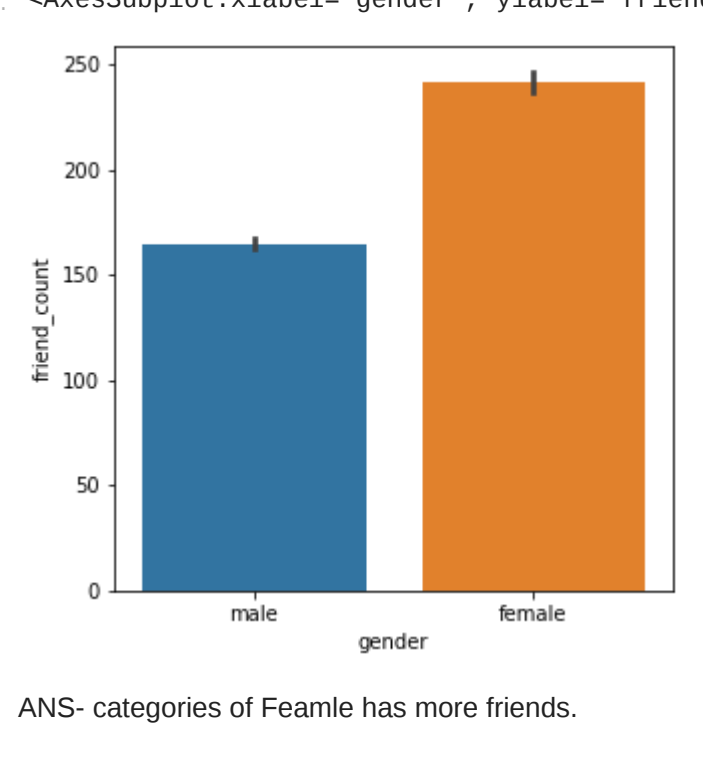
Out[113...

	friend_count
	sum
gender	
female	9740258
male	9666787

In [114...

```python
plt.figure(figsize=(5,5))
sns.barplot(x=df.gender,y=df.friend_count)
```

Out[114... <AxesSubplot:xlabel='gender', ylabel='friend_count'>



ANS- categories of Feamle has more friends.

Which category of gender initiated more friendships?

In [116...

```python
df.groupby('gender').agg({'friendships_initiated': ['sum']})
```

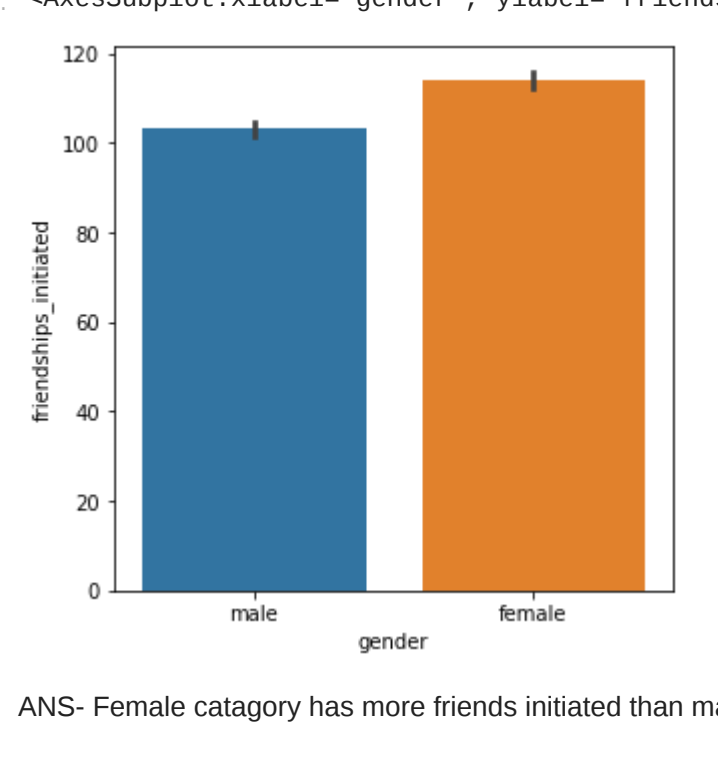
Out[116...

	friendships_initiated
	sum
gender	
female	4584894
male	6037023

In [124...

```python
plt.figure(figsize=(5,5))
sns.barplot(x=df.gender,y=df.friendships_initiated)
```

Out[124... <AxesSubplot:xlabel='gender', ylabel='friendships_initiated'>



ANS- Female catagory has more friends initiated than male.

What is the distribution of tenure across different categories of gender?

In [117...

```python
df.groupby('gender').agg({'tenure': ['sum']})
```

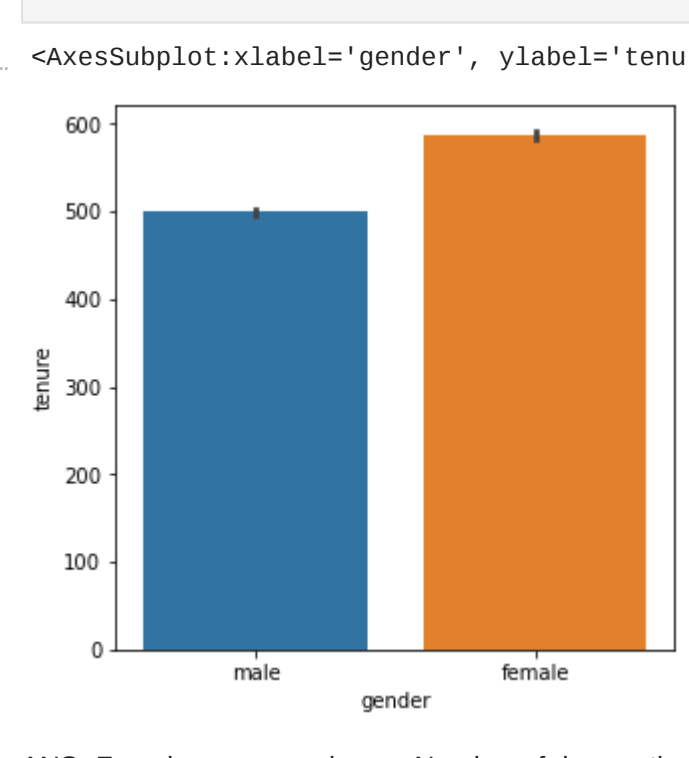
Out[117...

	tenure
	sum
gender	
female	23637151.0
male	29298972.0

In [126...

```python
plt.figure(figsize=(5,5))
sns.barplot(y=df.tenure,x=df.gender)
```

Out[126... <AxesSubplot:xlabel='gender', ylabel='tenure'>



ANS- Females are spend more Number of days active on Facebook than males.

4) Analysis based on the least active users on Facebook

- How many users have no friends?

In [96]:

```python
df.friend_count.value_counts()
```

Out[96]:

```
0       1962
1       1816
2       1117
3        860
5        789
        ... 
3299       1
4576       1
2339       1
4384       1
2047       1
Name: friend_count, Length: 2562, dtype: int64
```

ANS- 1962 users don't have friends

How many users did not like any posts?

In [104...

```python
df.likes.value_counts()
```

Out[104...

```
0       22308
1        6928
2        4434
3        3240
4        2507
        ...  
2087        1
3467        1
6119        1
4068        1
2047        1
Name: likes, Length: 2924, dtype: int64
```

ANS-22308 users did not like any post

How many users did not receive any likes?

In [102...

```python
df.likes_received.value_counts()
```

Out[102...

```
0       24428
1        7305
2        4541
3        3347
4        2669
        ...  
1610        1
3659        1
3723        1
2660        1
2047        1
Name: likes_received, Length: 2681, dtype: int64
```

ANS-24428 no of users did'nt recive any likes

Analysis based on the user accessibility (Mobile Devices vs. Web Devices)

What is the average number of posts liked by users (based on gender) through web vs.mobile devices?

In [153...

```python
a=df.groupby('gender').agg({'mobile_likes': ['mean']})
a
```

Out[153...

	mobile_likes
	mean
gender	
female	172.912928
male	60.261328

In [154...

```python
b=df.groupby('gender').agg({'www_likes': ['mean']})
b
```

Out[154...

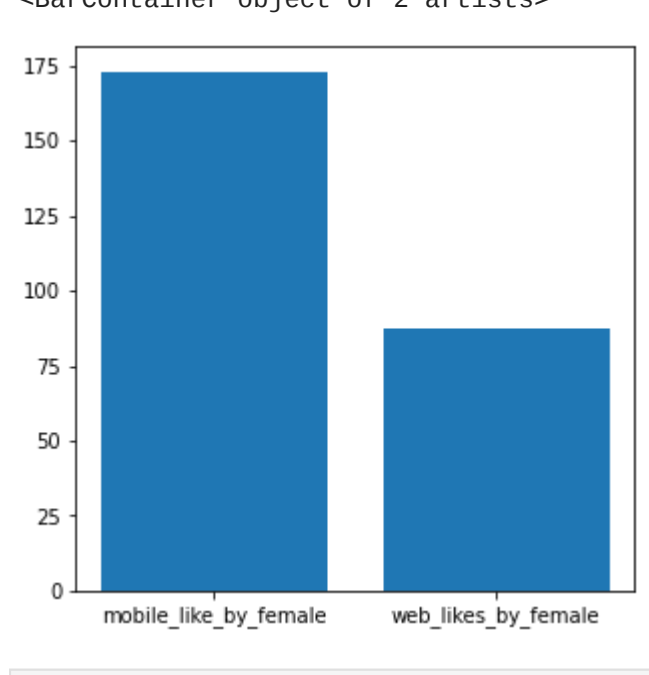
	www_likes
	mean
gender	
female	87.138297
male	24.416550

In [157...

```python
plt.figure(figsize=(5,5))
x=["mobile_like_by_female","web_likes_by_female"]
y=[list(a.loc['female'])][0],list(b.loc['female'])[0]
plt.bar(x,y)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\numpy\lib\stride_tricks.py:536: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which
is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' wh
en creating the ndarray.
  args = [np.array(_m, copy=False, subok=subok) for _m in args]
C:\ProgramData\Anaconda3\lib\site-packages\numpy\core\_asarray.py:102: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a
list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when cr
eating the ndarray.
  return array(a, dtype, copy=False, order=order)
```

Out[157... <BarContainer object of 2 artists>

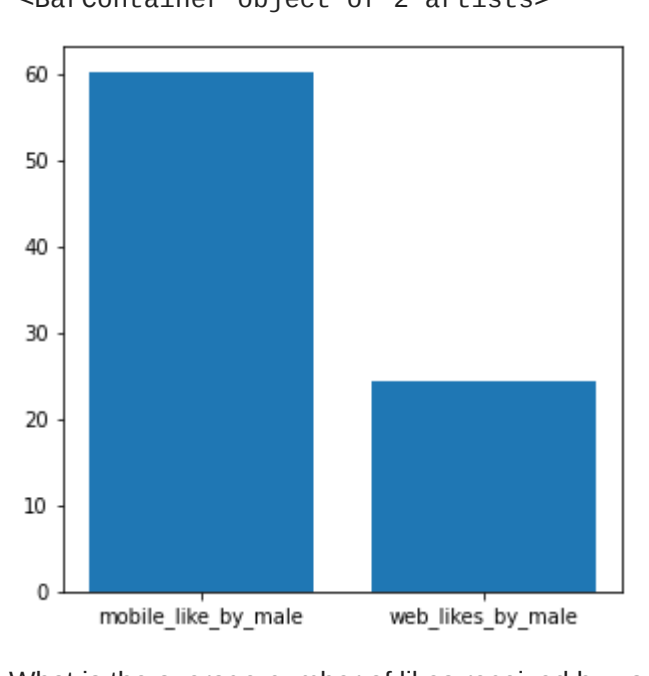


In [161...

```python
plt.figure(figsize=(5,5))
x=["mobile_like_by_male","web_likes_by_male"]
y=[list(a.loc['male'])][0],list(b.loc['male'])[0]
plt.bar(x,y)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\numpy\lib\stride_tricks.py:536: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which
is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' wh
en creating the ndarray.
  args = [np.array(_m, copy=False, subok=subok) for _m in args]
C:\ProgramData\Anaconda3\lib\site-packages\numpy\core\_asarray.py:102: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a
list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when cr
eating the ndarray.
  return array(a, dtype, copy=False, order=order)
```

Out[161... <BarContainer object of 2 artists>



What is the average number of likes received by users (based on gender) through web vs. mobile devices?

In [162...

```python
d=df.groupby('gender').agg({'mobile_likes_received': ['mean']})
d
```

Out[162...

	mobile_likes_received
	mean
gender	
female	147.100884
male	40.833015

In [163...

```python
e=df.groupby('gender').agg({'www_likes_received': ['mean']})
e
```

Out[163...

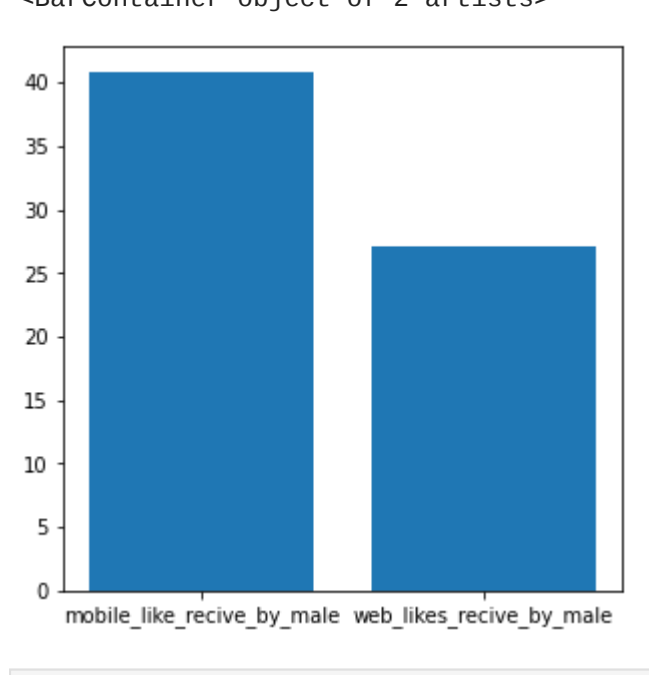
	www_likes_received
	mean
gender	
female	104.334451
male	27.078533

In [164...

```python
plt.figure(figsize=(5,5))
x=["mobile_like_recive_by_male","web_likes_recive_by_male"]
y=[list(d.loc['male'])][0],list(e.loc['male'])[0]
plt.bar(x,y)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\numpy\lib\stride_tricks.py:536: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which
is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' wh
en creating the ndarray.
  args = [np.array(_m, copy=False, subok=subok) for _m in args]
C:\ProgramData\Anaconda3\lib\site-packages\numpy\core\_asarray.py:102: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a
list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when cr
eating the ndarray.
  return array(a, dtype, copy=False, order=order)
```

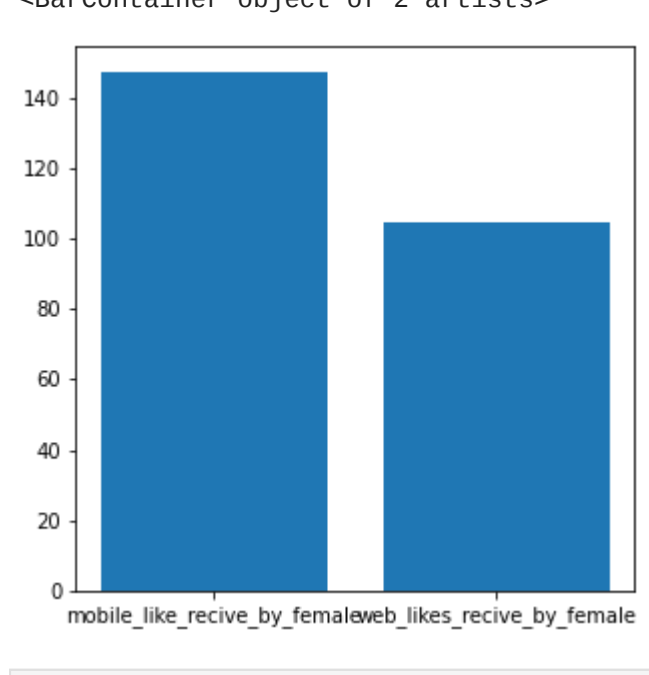
Out[164... <BarContainer object of 2 artists>



In [165...

```python
plt.figure(figsize=(5,5))
x=["mobile_like_recive_by_female","web_likes_recive_by_female"]
y=[list(d.loc['female'])][0],list(e.loc['female'])[0]
plt.bar(x,y)
```

Out[165... <BarContainer object of 2 artists>



In [ ]: