

```
In [4]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
In [5]: gym=pd.read_csv("gym_membership.csv")
gym.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   id               1000 non-null   int64  
 1   gender            1000 non-null   object  
 2   birthday          1000 non-null   object  
 3   Age                1000 non-null   int64  
 4   abonoment_type    1000 non-null   object  
 5   visit_per_week     1000 non-null   int64  
 6   days_per_week      1000 non-null   object  
 7   attend_group_lesson 1000 non-null   bool    
 8   fav_group_lesson    503 non-null    object  
 9   avg_time_check_in  1000 non-null   object  
 10  avg_time_check_out 1000 non-null   object  
 11  avg_time_in_gym    1000 non-null   int64  
 12  drink_abo          1000 non-null   bool    
 13  fav_drink           496 non-null    object  
 14  personal_training   1000 non-null   bool    
 15  name_personal_trainer 518 non-null   object  
 16  uses_sauna          1000 non-null   bool    
dtypes: bool(4), int64(4), object(9)
memory usage: 105.6+ KB
```

```
In [6]: gym.head()
```

	id	gender	birthday	Age	abonoment_type	visit_per_week	days_per_week	attend_
0	1	Female	1997-04-18	27	Premium		4	Mon, Sat, Tue, Wed
1	2	Female	1977-09-18	47	Standard		3	Mon, Sat, Wed
2	3	Male	1983-03-30	41	Premium		1	Sat
3	4	Male	1980-04-12	44	Premium		3	Sat, Tue, Wed
4	5	Male	1980-09-10	44	Standard		2	Thu, Wed

```
In [7]: gym.tail()
```

Out[7]:

	id	gender	birthday	Age	abonoment_type	visit_per_week	days_per_week	at
995	996	Female	1984-09-22	40	Standard		3	Thu, Tue, Wed
996	997	Female	2008-11-19	15	Standard		3	Fri, Mon, Sun
997	998	Male	1984-10-05	40	Standard		2	Fri, Tue
998	999	Male	2001-02-22	23	Standard		4	Mon, Sun, Thu, Tue
999	1000	Female	2006-05-07	18	Premium		2	Thu, Tue

Splitting Birthday for Predicting most member of age of particular Year

In [8]:

```
gym[['Year', 'Month', 'Day']] = gym['birthday'].str.split('-', expand=True)
gym.head()
```

Out[8]:

	id	gender	birthday	Age	abonoment_type	visit_per_week	days_per_week	attend
0	1	Female	1997-04-18	27	Premium		4	Mon, Sat, Tue, Wed
1	2	Female	1977-09-18	47	Standard		3	Mon, Sat, Wed
2	3	Male	1983-03-30	41	Premium		1	Sat
3	4	Male	1980-04-12	44	Premium		3	Sat, Tue, Wed
4	5	Male	1980-09-10	44	Standard		2	Thu, Wed

In [9]:

```
gym.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   id               1000 non-null   int64  
 1   gender            1000 non-null   object  
 2   birthday          1000 non-null   object  
 3   Age                1000 non-null   int64  
 4   abonoment_type    1000 non-null   object  
 5   visit_per_week     1000 non-null   int64  
 6   days_per_week      1000 non-null   object  
 7   attend_group_lesson 1000 non-null   bool    
 8   fav_group_lesson    503 non-null    object  
 9   avg_time_check_in   1000 non-null   object  
 10  avg_time_check_out  1000 non-null   object  
 11  avg_time_in_gym     1000 non-null   int64  
 12  drink_abo          1000 non-null   bool    
 13  fav_drink           496 non-null    object  
 14  personal_training   1000 non-null   bool    
 15  name_personal_trainer 518 non-null   object  
 16  uses_sauna          1000 non-null   bool    
 17  Year                1000 non-null   object  
 18  Month               1000 non-null   object  
 19  Day                 1000 non-null   object  
dtypes: bool(4), int64(4), object(12)
memory usage: 129.0+ KB
```

```
In [10]: gym['Year'] = gym['Year'].astype(int)
```

```
In [11]: gym.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   id               1000 non-null   int64  
 1   gender            1000 non-null   object  
 2   birthday          1000 non-null   object  
 3   Age                1000 non-null   int64  
 4   abonoment_type    1000 non-null   object  
 5   visit_per_week     1000 non-null   int64  
 6   days_per_week      1000 non-null   object  
 7   attend_group_lesson 1000 non-null   bool    
 8   fav_group_lesson    503 non-null    object  
 9   avg_time_check_in   1000 non-null   object  
 10  avg_time_check_out  1000 non-null   object  
 11  avg_time_in_gym     1000 non-null   int64  
 12  drink_abo          1000 non-null   bool    
 13  fav_drink           496 non-null    object  
 14  personal_training   1000 non-null   bool    
 15  name_personal_trainer 518 non-null   object  
 16  uses_sauna          1000 non-null   bool    
 17  Year                1000 non-null   int64  
 18  Month               1000 non-null   object  
 19  Day                 1000 non-null   object  
dtypes: bool(4), int64(5), object(11)
memory usage: 129.0+ KB
```

Spilting days of week for predicting most days visited in Gym.

```
In [12]: gym[['Day1', 'Day2', 'Day3', 'Day4', 'Day5']] = gym['days_per_week'].str.split(',', expand=True)
gym.head()
```

Out[12]:

	id	gender	birthday	Age	abonoment_type	visit_per_week	days_per_week	attend_
0	1	Female	1997-04-18	27	Premium	4	Mon, Sat, Tue, Wed	
1	2	Female	1977-09-18	47	Standard	3	Mon, Sat, Wed	
2	3	Male	1983-03-30	41	Premium	1	Sat	
3	4	Male	1980-04-12	44	Premium	3	Sat, Tue, Wed	
4	5	Male	1980-09-10	44	Standard	2	Thu, Wed	

5 rows × 25 columns



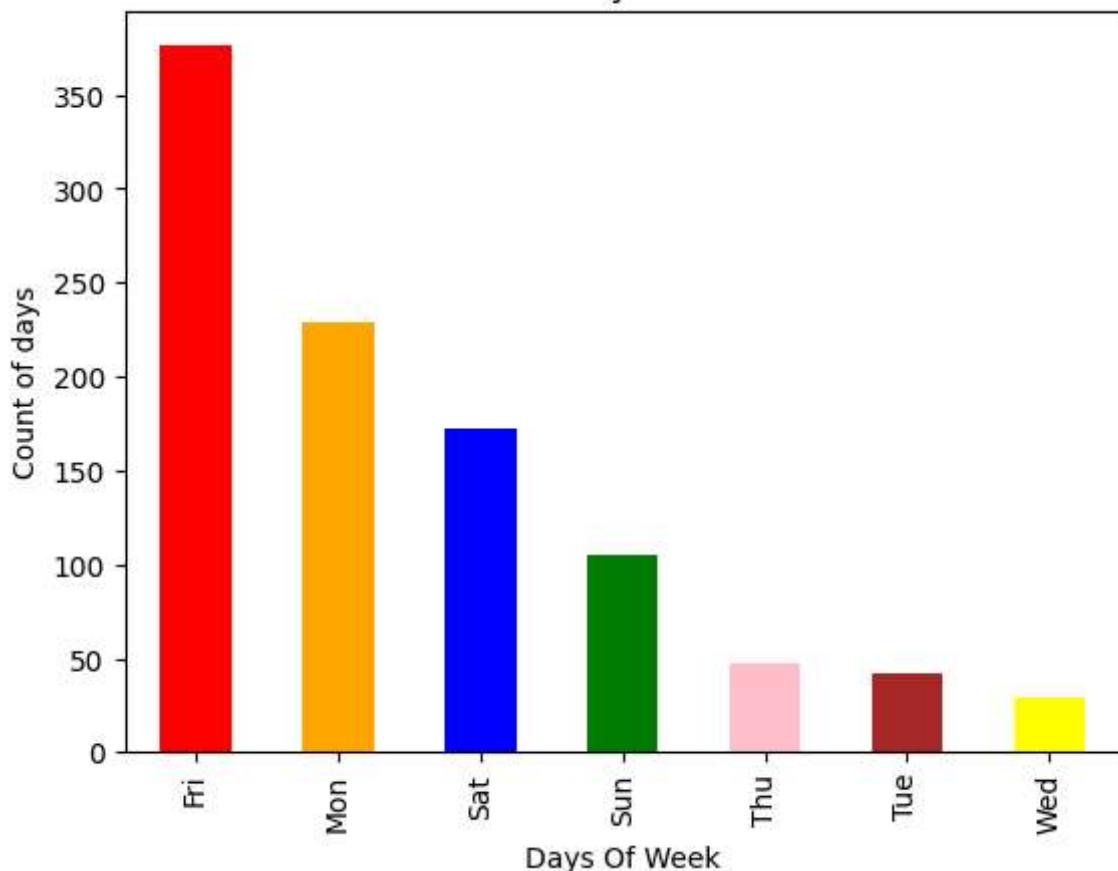
Counting Most days visited.

```
In [13]: Day_1=gym['Day1'].value_counts()
print(Day_1)
```

```
Day1
Fri    376
Mon    229
Sat    172
Sun    105
Thu     47
Tue     42
Wed     29
Name: count, dtype: int64
```

```
In [14]: Day_1.plot(kind='bar', color=['red', 'orange', 'blue', 'green', 'pink', 'brown', 'yellow'])
plt.xlabel('Days Of Week')
plt.ylabel('Count of days')
plt.title("Analysis 1")
plt.show()
```

Analysis 1

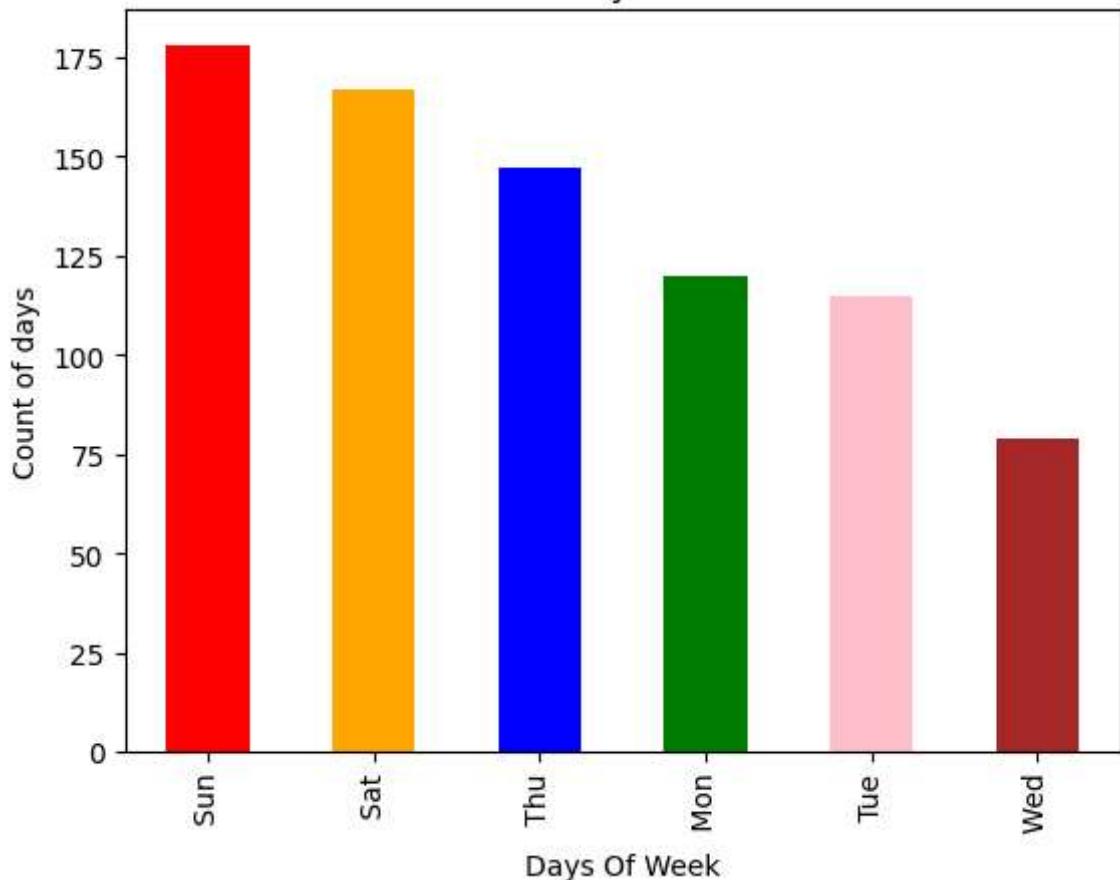


```
In [15]: Day_2=gym['Day2'].value_counts()  
print(Day_2)
```

```
Day2  
Sun    178  
Sat    167  
Thu    147  
Mon    120  
Tue    115  
Wed     79  
Name: count, dtype: int64
```

```
In [16]: Day_2.plot(kind='bar',color=['red','orange','blue','green','pink','brown','yellow'])  
plt.xlabel('Days Of Week')  
plt.ylabel('Count of days')  
plt.title("Analysis 2")  
plt.show()
```

Analysis 2

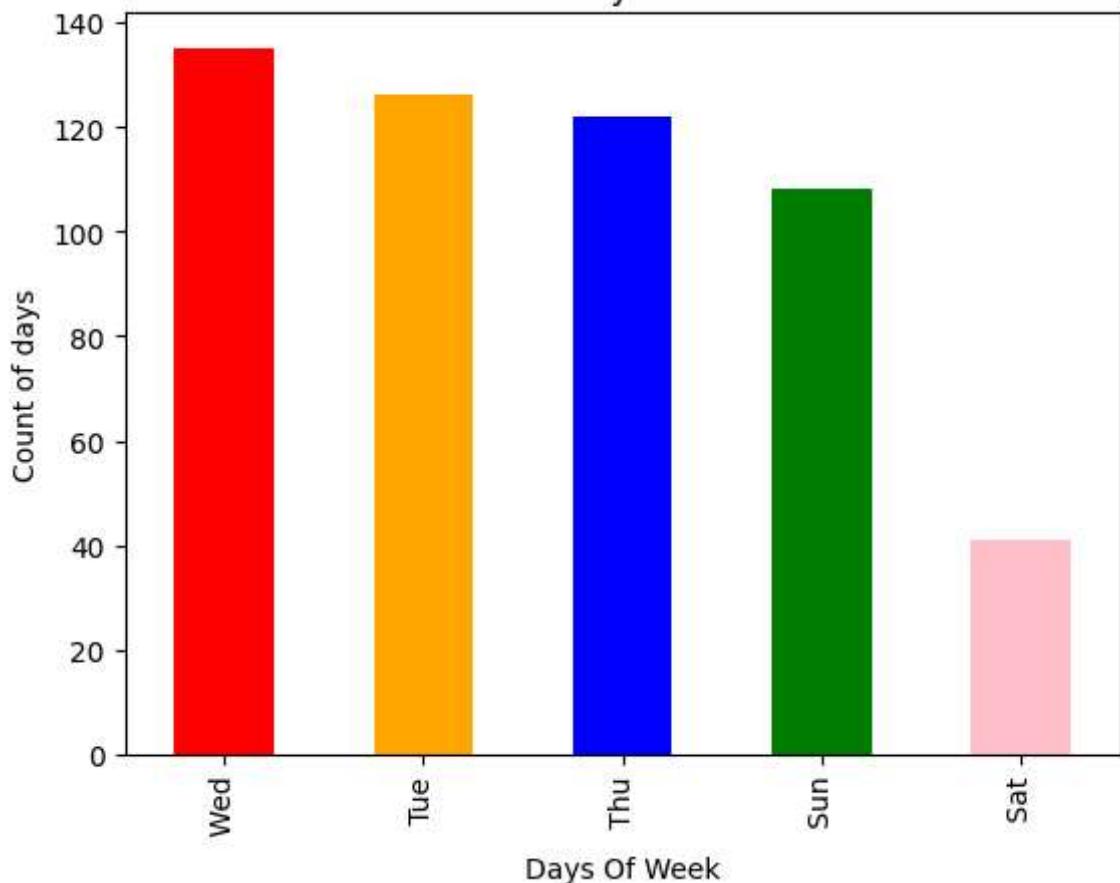


```
In [17]: Day_3=gym['Day3'].value_counts()  
print(Day_3)
```

```
Day3  
Wed    135  
Tue    126  
Thu    122  
Sun    108  
Sat     41  
Name: count, dtype: int64
```

```
In [18]: Day_3.plot(kind='bar',color=['red','orange','blue','green','pink','brown','yellow'])  
plt.xlabel('Days Of Week')  
plt.ylabel('Count of days')  
plt.title("Analysis 3")  
plt.show()
```

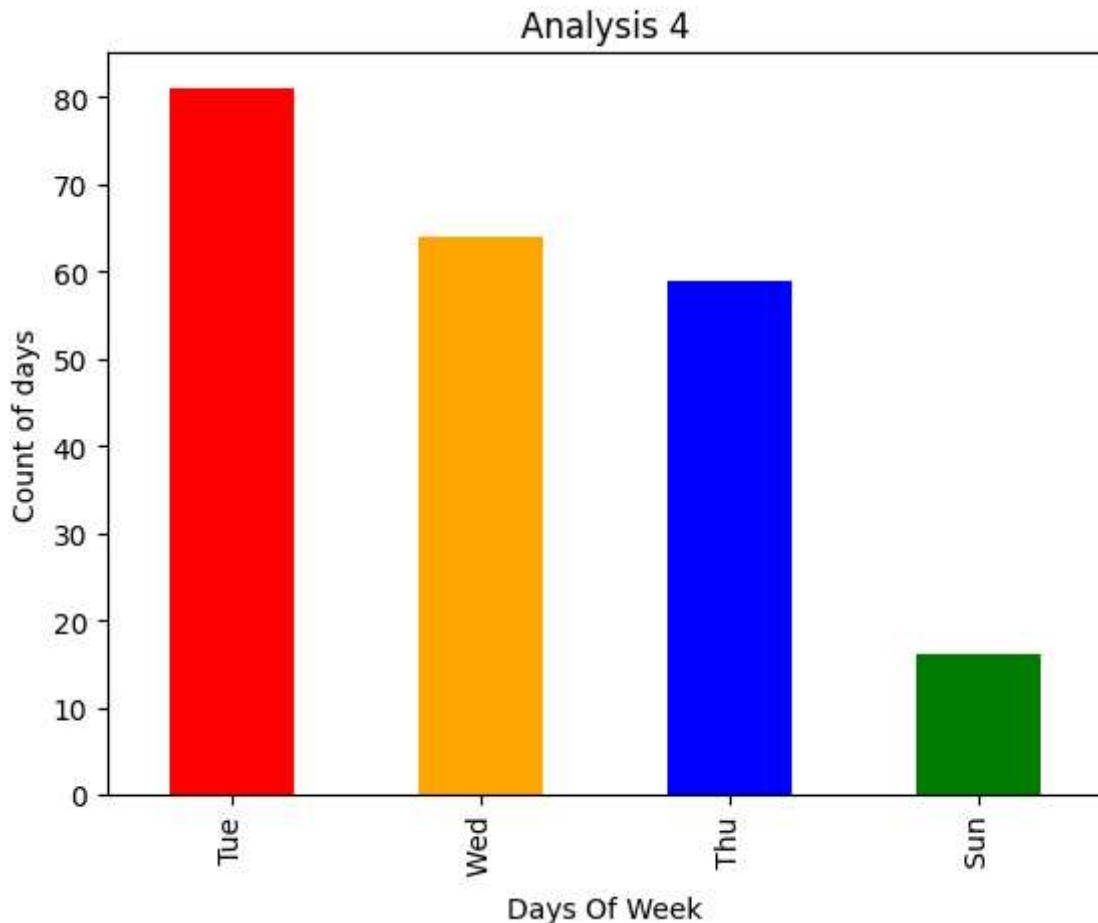
Analysis 3



```
In [19]: Day_4=gym['Day4'].value_counts()
print(Day_4)
```

```
Day4
Tue    81
Wed    64
Thu    59
Sun    16
Name: count, dtype: int64
```

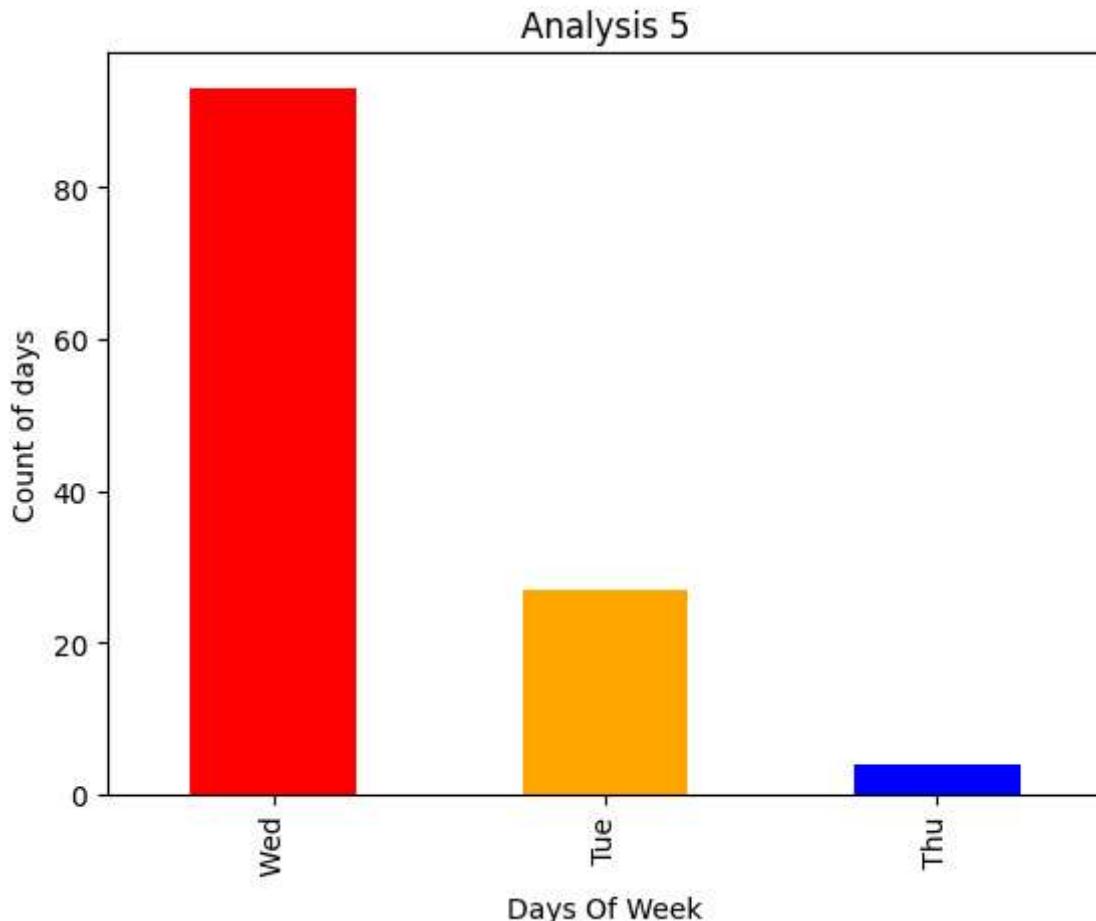
```
In [20]: Day_4.plot(kind='bar',color=['red','orange','blue','green','pink','brown','yellow'])
plt.xlabel('Days Of Week')
plt.ylabel('Count of days')
plt.title("Analysis 4")
plt.show()
```



```
In [21]: Day_5=gym['Day5'].value_counts()
print(Day_5)
```

```
Day5
Wed    93
Tue    27
Thu     4
Name: count, dtype: int64
```

```
In [22]: Day_5.plot(kind='bar',color=['red','orange','blue','green','pink','brown','yellow'])
plt.xlabel('Days Of Week')
plt.ylabel('Count of days')
plt.title("Analysis 5")
plt.show()
```



From Above Analysis we understand that Most People attend GYM at Sunday.

```
In [23]: gym.columns
```

```
Out[23]: Index(['id', 'gender', 'birthday', 'Age', 'abonnement_type', 'visit_per_week',
       'days_per_week', 'attend_group_lesson', 'fav_group_lesson',
       'avg_time_check_in', 'avg_time_check_out', 'avg_time_in_gym',
       'drink_abo', 'fav_drink', 'personal_training', 'name_personal_trainer',
       'uses_sauna', 'Year', 'Month', 'Day', 'Day1', 'Day2', 'Day3', 'Day4',
       'Day5'],
      dtype='object')
```

Splitting Check in time of gym for checking at which hour people check in to Gym the most.

```
In [24]: gym[['in_hour', 'in_min', 'in_sec']] = gym['avg_time_check_in'].str.split(':', expand=True)
```

```
In [25]: gym[['in_hour', 'in_min', 'in_sec']]
```

Out[25]:

	in_hour	in_min	in_sec
0	19	31	00
1	19	31	00
2	08	29	00
3	09	54	00
4	08	29	00
...
995	20	56	00
996	09	14	00
997	17	21	00
998	10	23	00
999	16	41	00

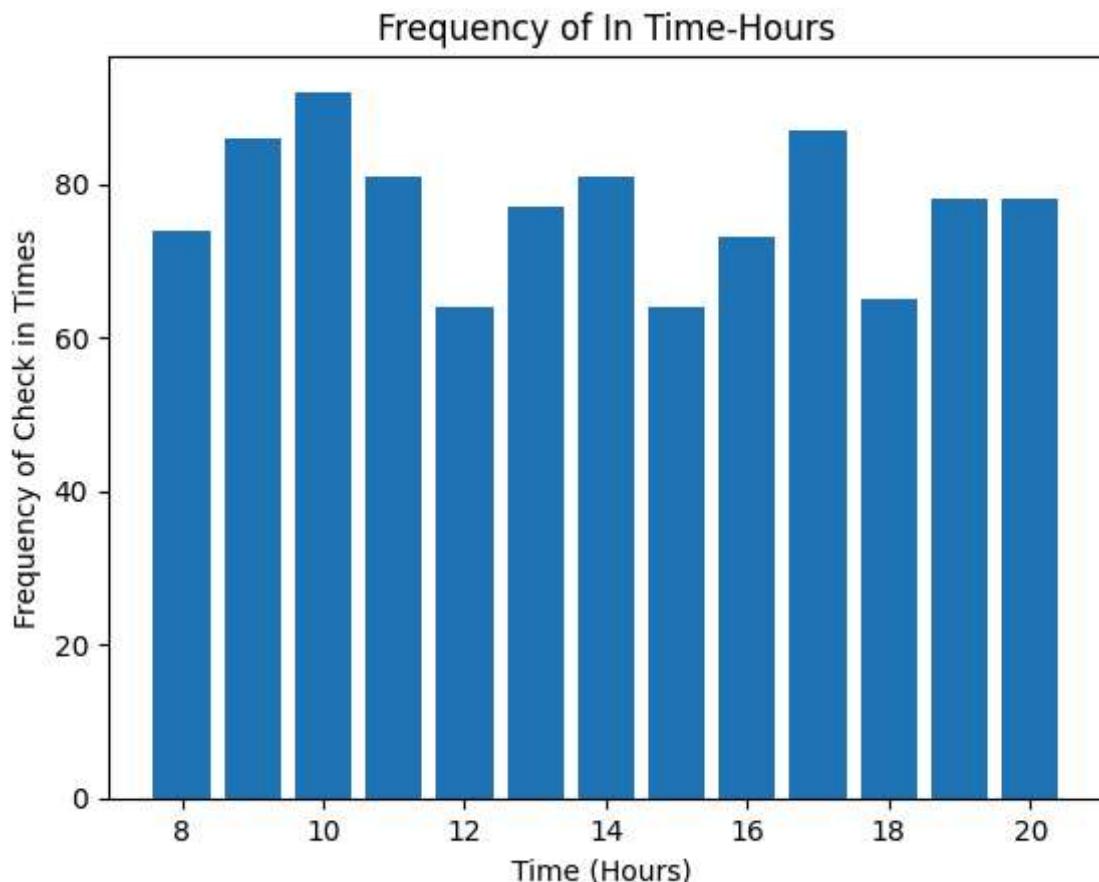
1000 rows × 3 columns

In [26]: gym['in_hour'] = gym['in_hour'].astype(int)

In [27]: hour_count=gym['in_hour'].value_counts().sort_index()

In [28]: hour_count

Out[28]: in_hour
8 74
9 86
10 92
11 81
12 64
13 77
14 81
15 64
16 73
17 87
18 65
19 78
20 78
Name: count, dtype: int64In [29]: plt.bar(hour_count.index,hour_count.values)
plt.xlabel("Time (Hours) ")
plt.ylabel("Frequency of Check in Times")
plt.title("Frequency of In Time-Hours")
plt.show()



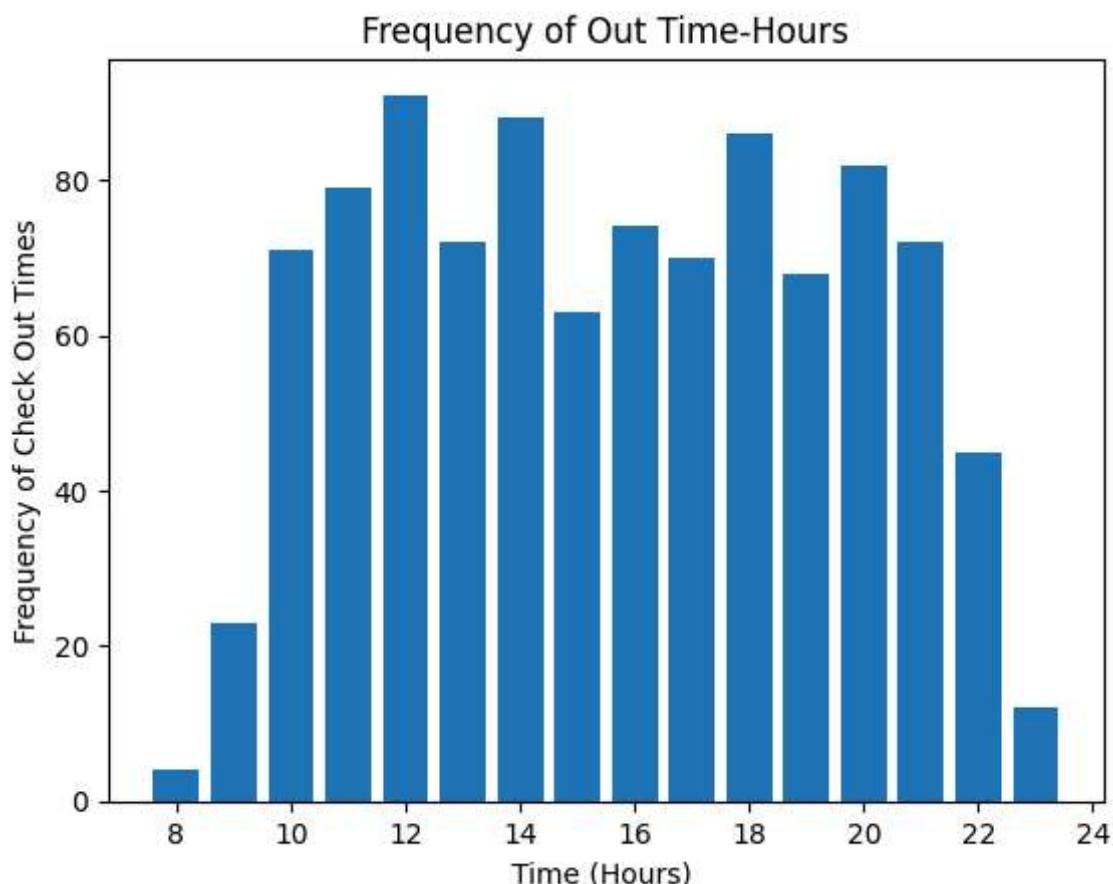
Most People Visit gym at check in time at 10AM

Spliting Check in time of gym for checking at which hour people check out to Gym the most.

```
In [30]: gym[['out_hour', 'out_min', 'out_sec']] = gym['avg_time_check_out'].str.split(':', expand=True)
In [31]: gym['out_hour'] = gym['out_hour'].astype(int)
In [32]: out_hr = gym['out_hour'].value_counts().sort_index()
In [33]: out_hr
```

```
Out[33]: out_hour  
8      4  
9     23  
10     71  
11     79  
12     91  
13     72  
14     88  
15     63  
16     74  
17     70  
18     86  
19     68  
20     82  
21     72  
22     45  
23     12  
Name: count, dtype: int64
```

```
In [34]: plt.bar(out_hr.index,out_hr.values)  
plt.xlabel("Time (Hours) ")  
plt.ylabel("Frequency of Check Out Times")  
plt.title("Frequency of Out Time-Hours")  
plt.show()
```



Most People Visit gym at check OUT time at 12AM

```
In [35]: gender_count = gym['gender'].value_counts().sort_index()  
print(gender_count)
```

```
gender
Female    503
Male      497
Name: count, dtype: int64
```

female candidate visites Comparative more to GYM.

```
In [36]: gym['Year']
```

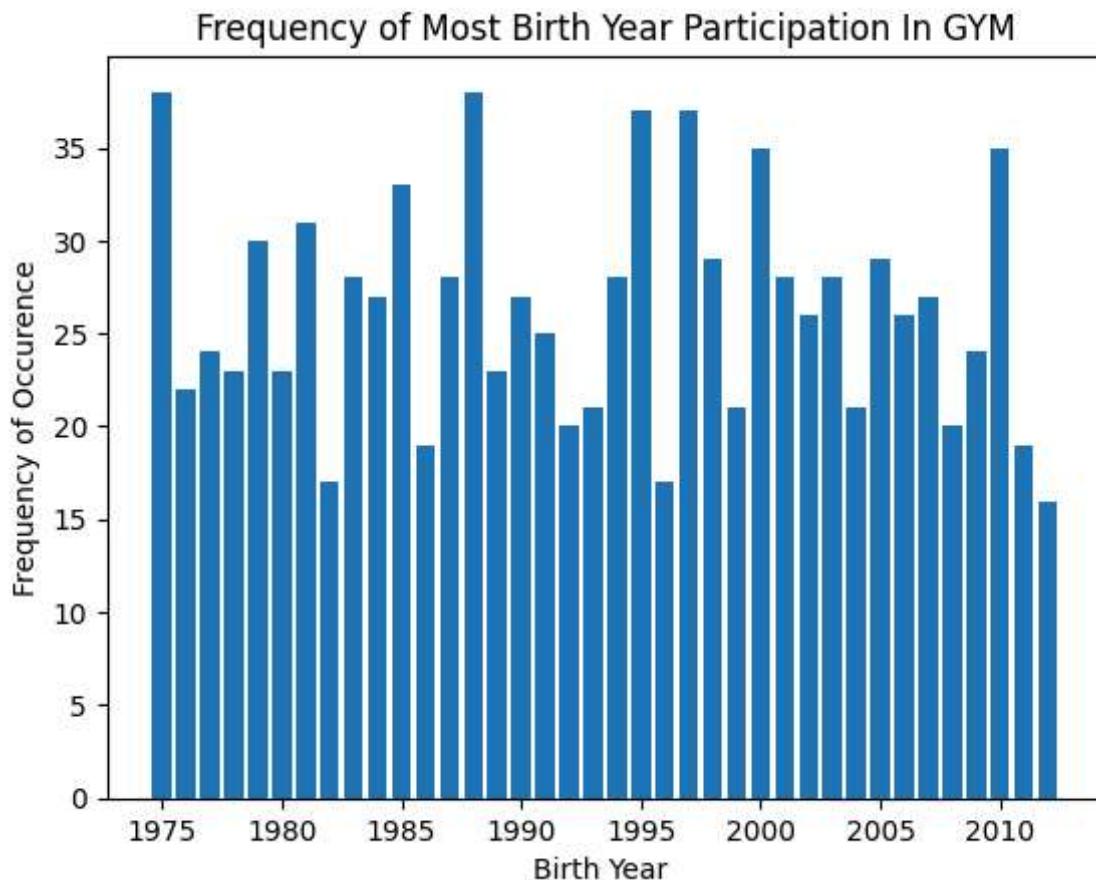
```
Out[36]: 0      1997
         1      1977
         2      1983
         3      1980
         4      1980
         ...
        995     1984
        996     2008
        997     1984
        998     2001
        999     2006
Name: Year, Length: 1000, dtype: int64
```

```
In [37]: birthday=gym['Year'].value_counts()
```

```
In [38]: print(birthday)
```

```
Year
1975    38
1988    38
1997    37
1995    37
2000    35
2010    35
1985    33
1981    31
1979    30
1998    29
2005    29
1983    28
1987    28
1994    28
2003    28
2001    28
2007    27
1990    27
1984    27
2006    26
2002    26
1991    25
1977    24
2009    24
1980    23
1978    23
1989    23
1976    22
1999    21
2004    21
1993    21
2008    20
1992    20
1986    19
2011    19
1982    17
1996    17
2012    16
Name: count, dtype: int64
```

```
In [39]: plt.bar(birthday.index,birthday.values)
plt.xlabel("Birth Year ")
plt.ylabel("Frequency of Occurrence")
plt.title("Frequency of Most Birth Year Participation In GYM")
plt.show()
```



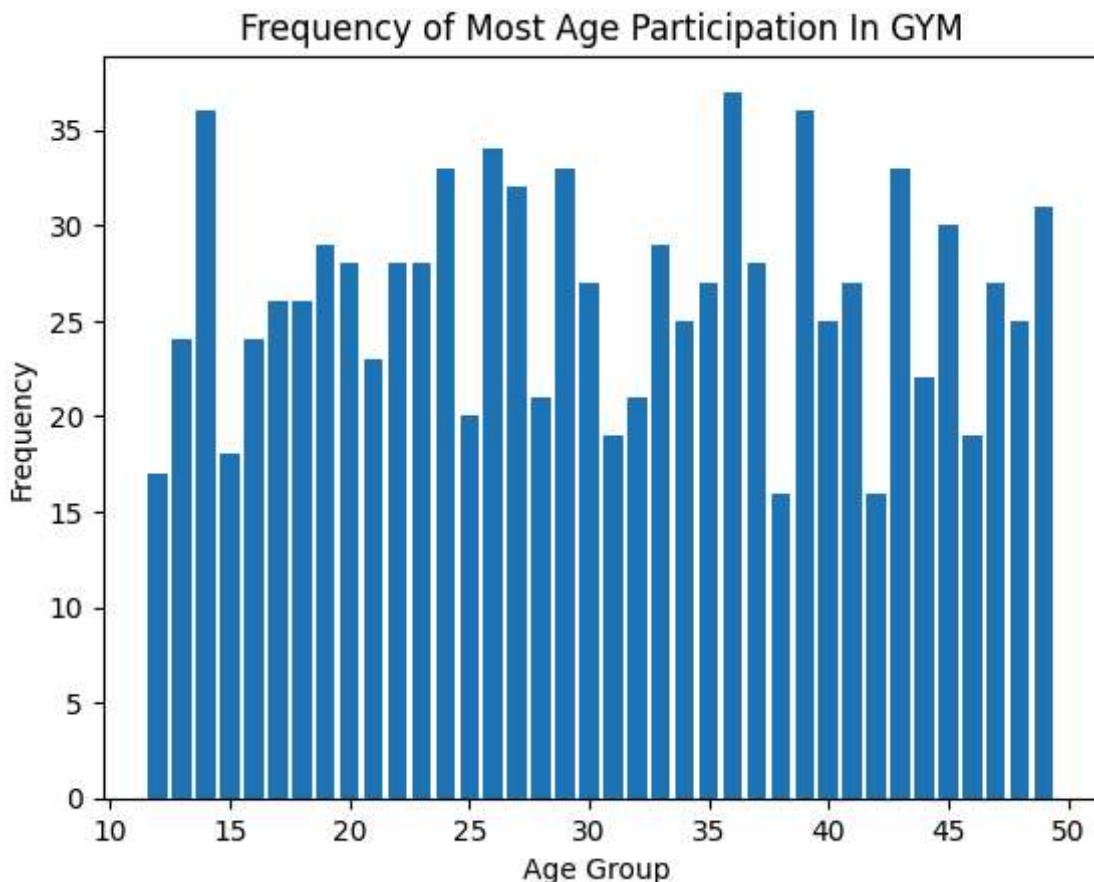
People born in year 1975 , 1988 , 1997 , 1995 are Most GymHolic

```
In [40]: age=gym[ 'Age' ].value_counts()
```

```
In [41]: age.head(5)
```

```
Out[41]: Age
36    37
14    36
39    36
26    34
29    33
Name: count, dtype: int64
```

```
In [42]: plt.bar(age.index,age.values)
plt.xlabel("Age Group ")
plt.ylabel("Frequency")
plt.title("Frequency of Most Age Participation In GYM")
plt.show()
```



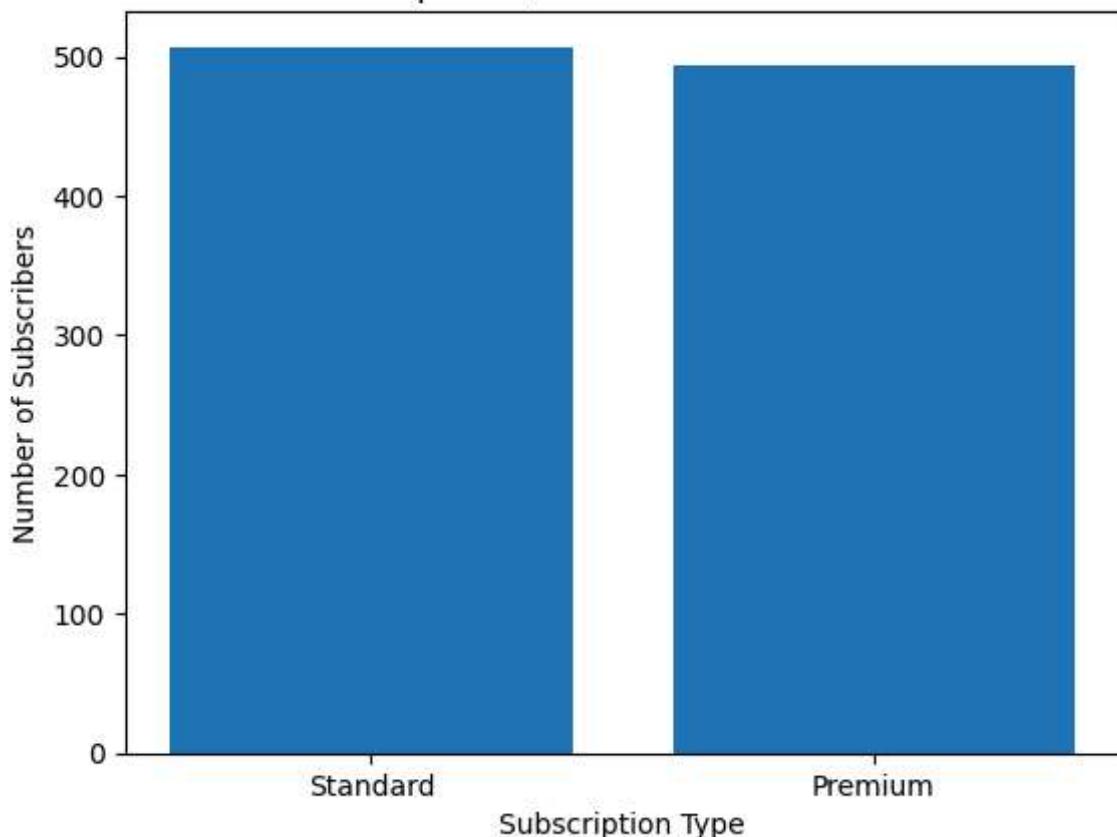
Age Group of 36 , 14 are most visitors of GYM.

```
In [43]: abonoment_type=gym[ 'abonoment_type' ].value_counts()  
abonoment_type
```

```
Out[43]: abonoment_type  
Standard      507  
Premium       493  
Name: count, dtype: int64
```

```
In [44]: plt.bar(abonoment_type.index,abonoment_type.values)  
plt.xlabel("Subscription Type")  
plt.ylabel("Number of Subscribers")  
plt.title("Subscipton v/s Number of Subscriber")  
plt.show()
```

Subscription v/s Number of Subscribers

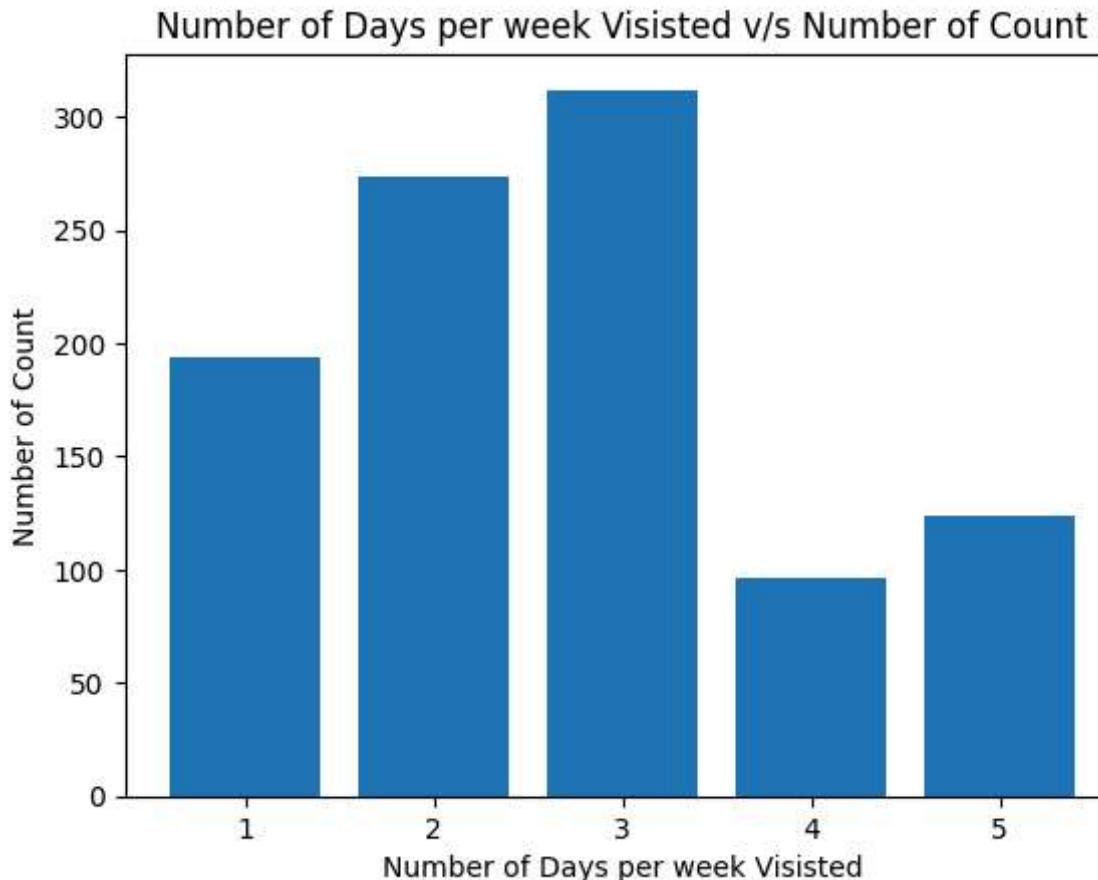


Subscription Type of standard GYM are subscribed by more.

```
In [45]: visits_per_week=gym['visit_per_week'].mode()  
print(visits_per_week)
```

```
0    3  
Name: visit_per_week, dtype: int64
```

```
In [46]: visits_per_week_graph=gym['visit_per_week'].value_counts()  
plt.bar(visits_per_week_graph.index,visits_per_week_graph.values)  
plt.xlabel("Number of Days per week Visisted")  
plt.ylabel("Number of Count")  
plt.title("Number of Days per week Visisted v/s Number of Count")  
plt.show()
```



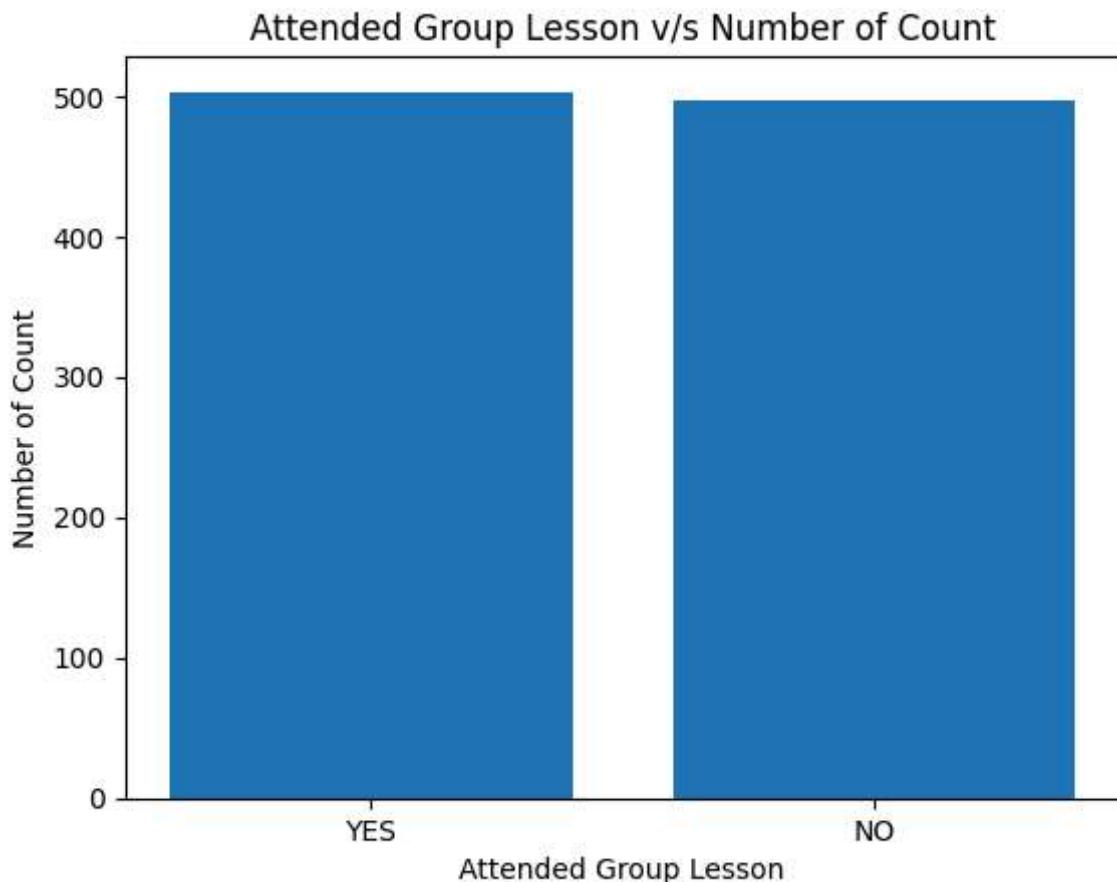
```
In [47]: print("people visits mostly 3 days per Week !!!! ")
```

```
people visits mostly 3 days per Week !!!!
```

```
In [48]: attended_group_lesson=gym[ 'attend_group_lesson' ].value_counts()  
attended_group_lesson
```

```
Out[48]: attend_group_lesson  
True      503  
False     497  
Name: count, dtype: int64
```

```
In [49]: custom_labels=["YES", "NO"]  
plt.bar(attended_group_lesson.index.astype(str),attended_group_lesson.values,tic  
plt.xlabel("Attended Group Lesson")  
plt.ylabel("Number of Count")  
plt.title("Attended Group Lesson v/s Number of Count")  
plt.show()
```



```
In [50]: avg_time=gym['avg_time_in_gym'].mean()
avg_time=int(avg_time)
print(avg_time)
```

105

```
In [51]: minutes=105
hours=minutes//60
remaining_minute=minutes%60
print("Average time of Gym is ::",hours,"hr",remaining_minute,'min')
```

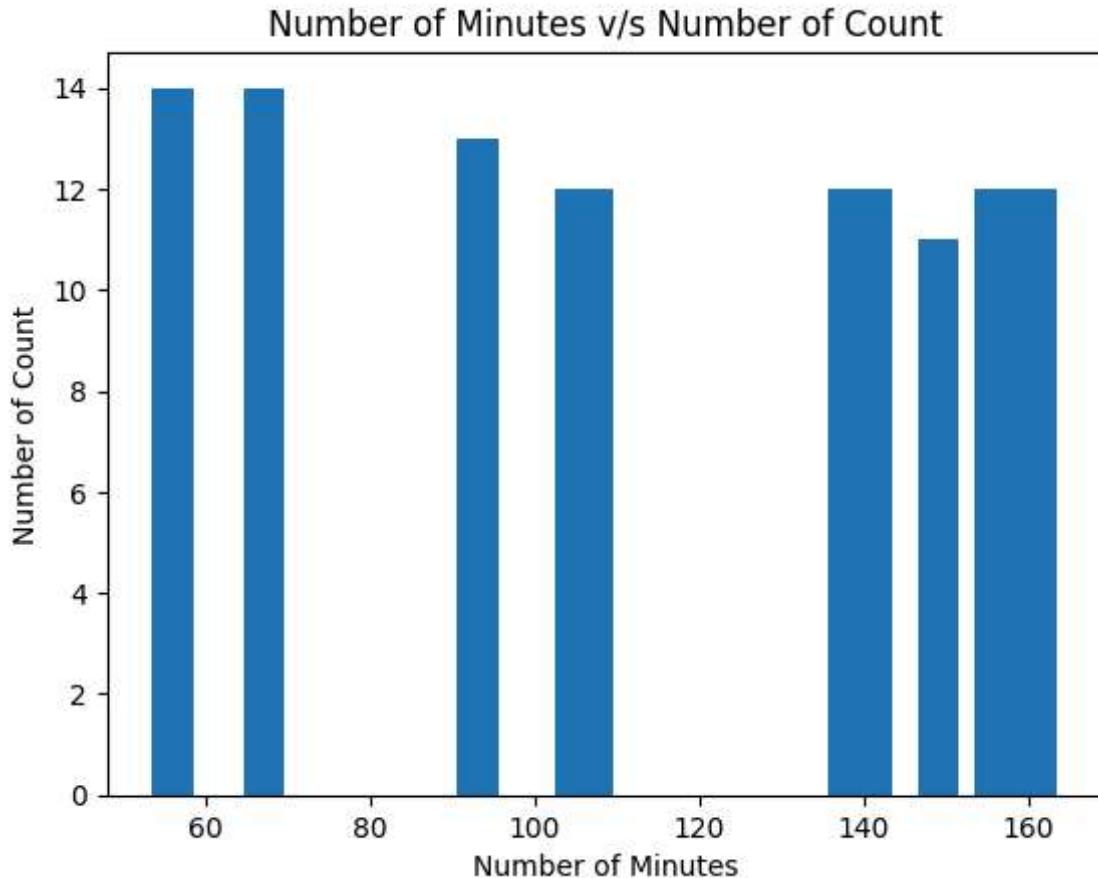
Average time of Gym is :: 1 hr 45 min

```
In [52]: avg_time_1=gym['avg_time_in_gym'].value_counts()
top_5=avg_time_1.head(10)
print(top_5)
```

```
avg_time_in_gym
56      14
67      14
93      13
107     12
161     12
141     12
156     12
105     12
138     12
149     11
Name: count, dtype: int64
```

```
In [53]: plt.bar(top_5.index,top_5.values,width=5)
plt.xlabel("Number of Minutes")
plt.ylabel("Number of Count")
```

```
plt.title("Number of Minutes v/s Number of Count")
plt.show()
```



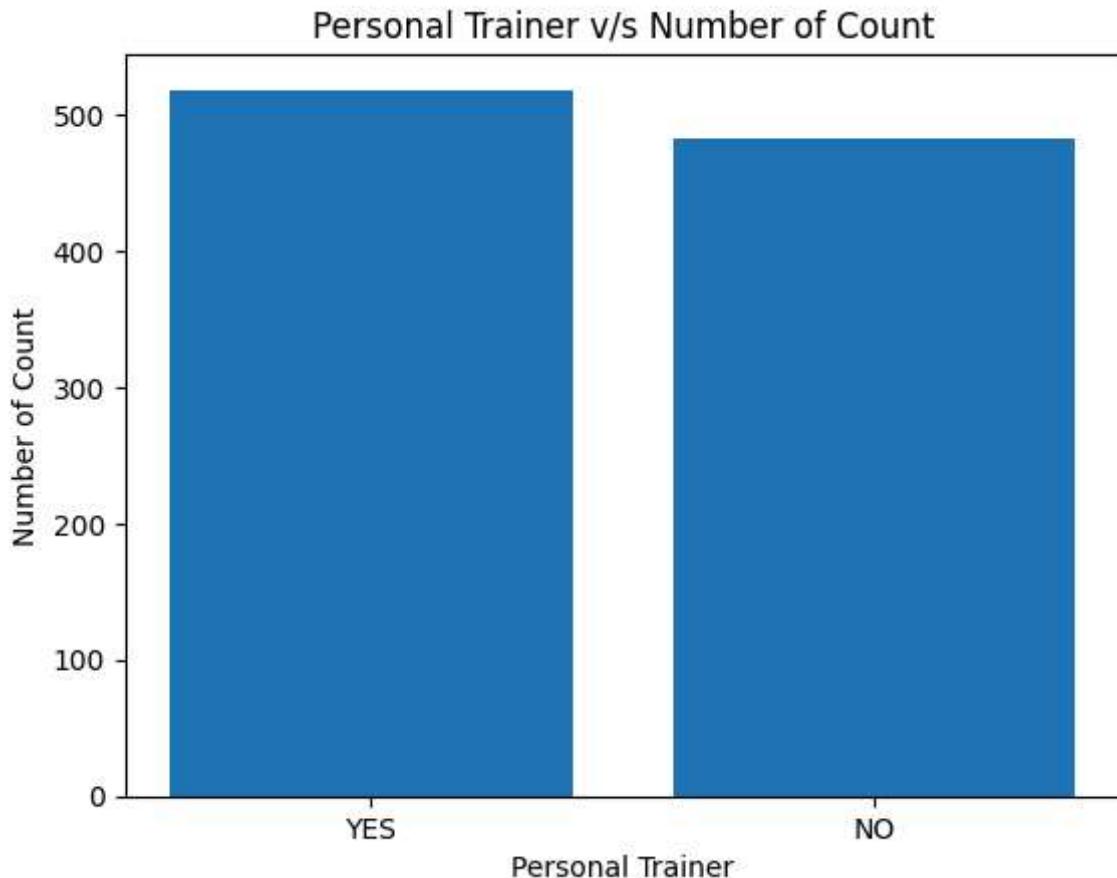
```
In [54]: print("Mostly People GYM for 50-70 minutes or 130-170 minutes.")
```

Mostly People GYM for 50-70 minutes or 130-170 minutes.

```
In [55]: personal_trainer=gym['personal_training'].value_counts()
personal_trainer
```

```
Out[55]: personal_training
True      518
False     482
Name: count, dtype: int64
```

```
In [56]: custom_labels_2=["YES", "NO"]
plt.bar(personal_trainer.index.astype(str),personal_trainer.values,tick_label=custom_labels_2)
plt.xlabel("Personal Trainer")
plt.ylabel("Number of Count")
plt.title("Personal Trainer v/s Number of Count")
plt.show()
```



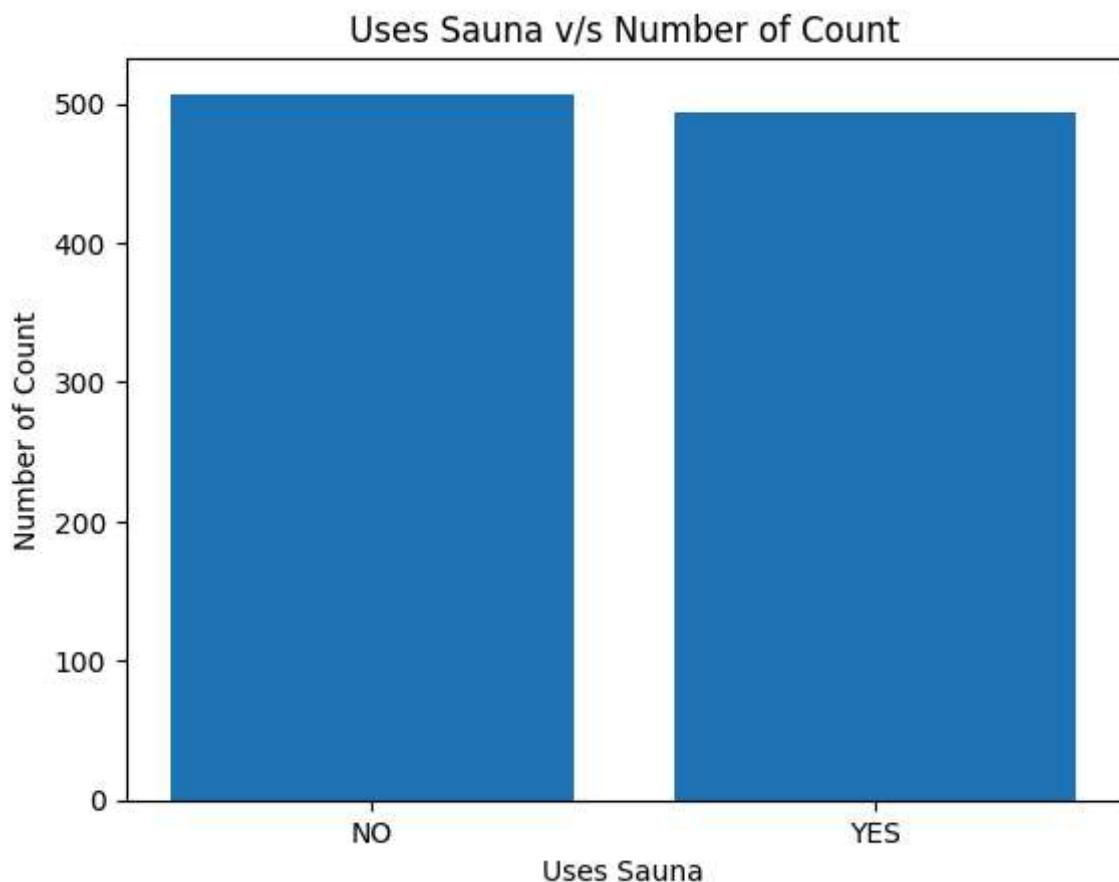
```
In [57]: print("Most of people Have peronal Trainer in GYM membership.")
```

Most of people Have peronal Trainer in GYM membership.

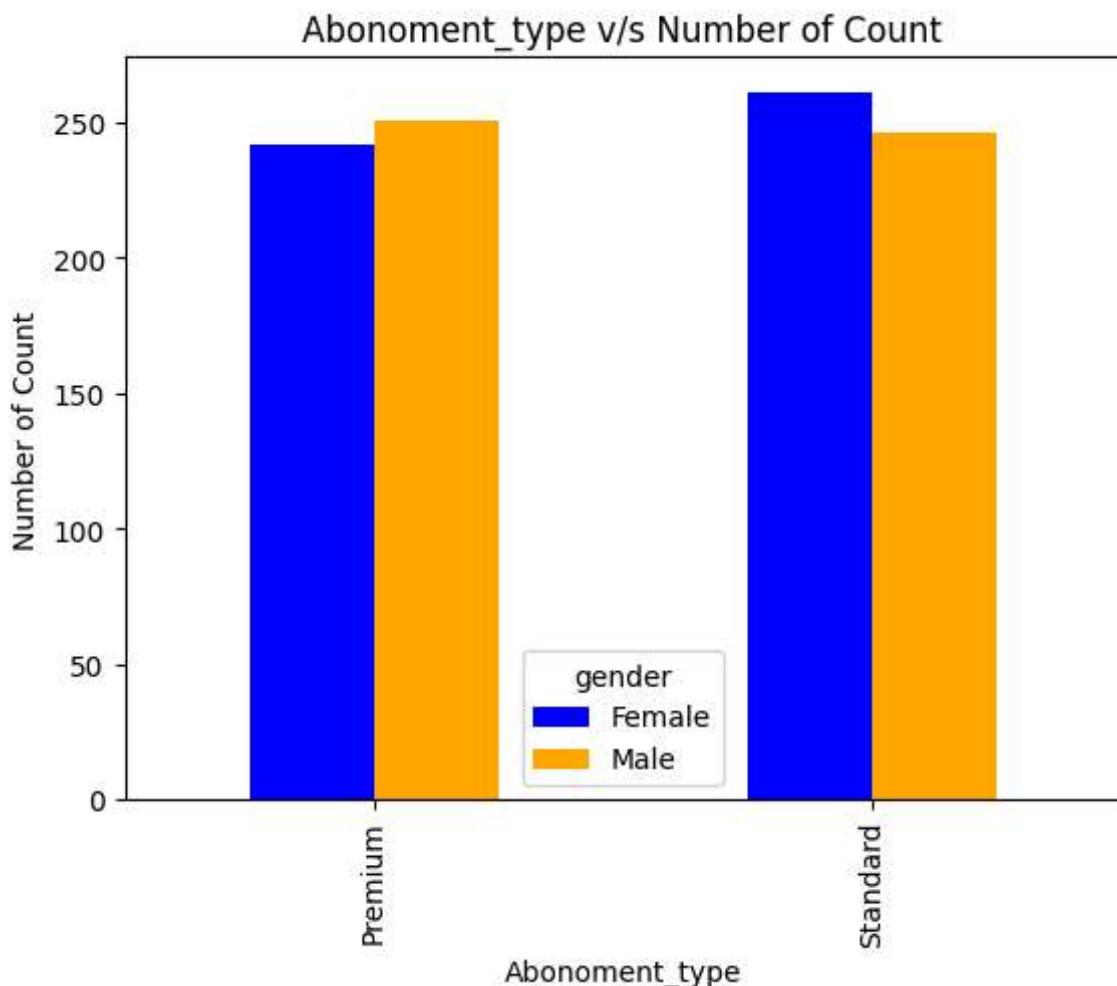
```
In [58]: sauna_users=gym['uses_sauna'].value_counts()  
print(sauna_users)
```

```
uses_sauna  
False    507  
True     493  
Name: count, dtype: int64
```

```
In [59]: custom_labels_4=["NO", "YES"]  
plt.bar(sauna_users.index.astype(str),sauna_users.values,tick_label=custom_label  
plt.xlabel("Uses Sauna")  
plt.ylabel("Number of Count")  
plt.title("Uses Sauna v/s Number of Count")  
plt.show()
```



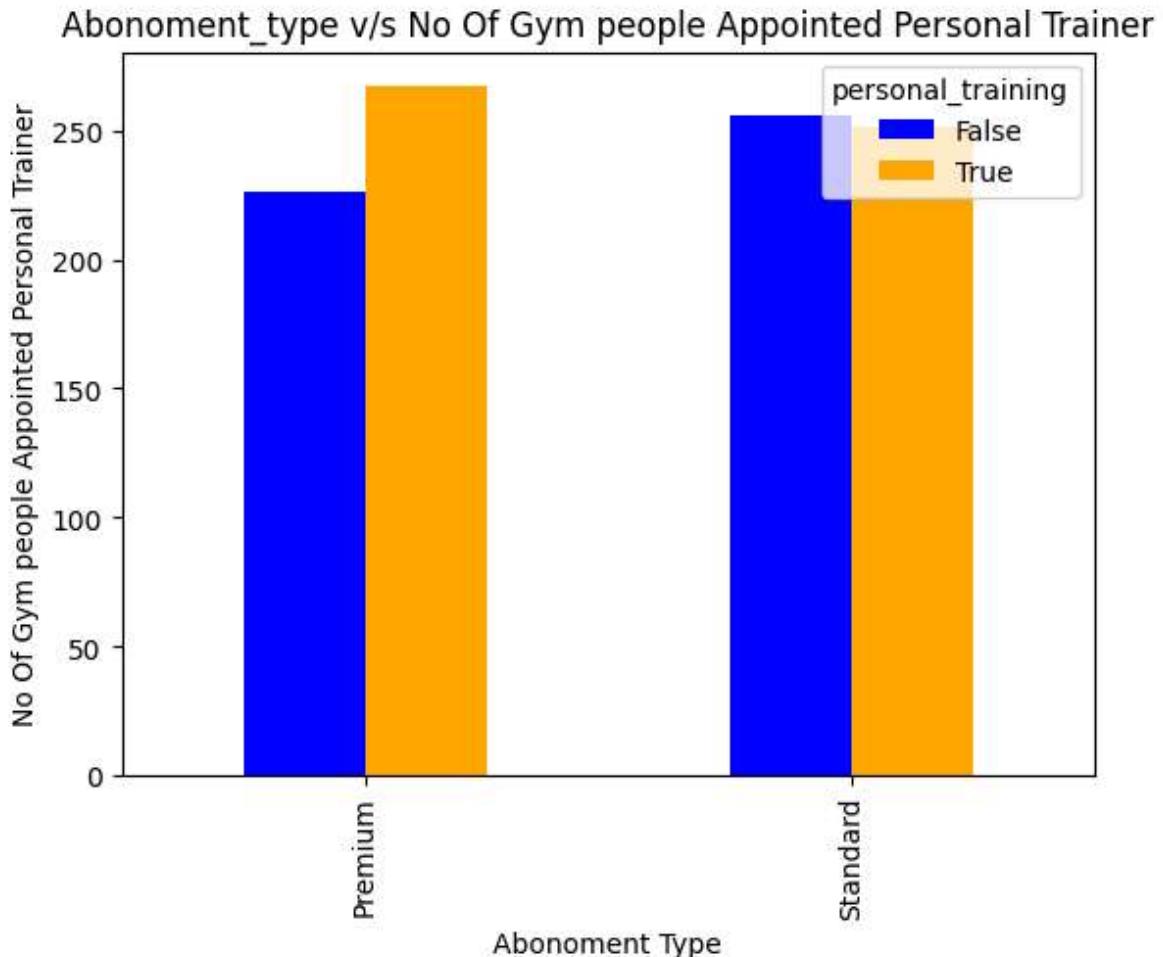
```
In [60]: gender_abonoment_type=pd.crosstab(gym['abonoment_type'], gym['gender'])
gender_abonoment_type.plot(kind='bar',stacked=False,color=['blue','orange'])
plt.xlabel("Abonoment_type")
plt.ylabel("Number of Count")
plt.title("Abonoment_type v/s Number of Count")
plt.show()
```



```
In [61]: print("Female Candidates are with Most Standard Abonoment type whereas Male are More with Premium Abonoment type")
```

Female Candidates are with Most Standard Abonoment type whereas Male are More with Premium Abonoment type

```
In [62]: training_abonoment_type=pd.crosstab(gym['abonoment_type'], gym['personal_trainin  
training_abonoment_type.plot(kind='bar',stacked=False,color=['blue','orange'])  
plt.xlabel("Abonoment Type")  
plt.ylabel("No Of Gym people Appointed Personal Trainer")  
plt.title("Abonoment_type v/s No Of Gym people Appointed Personal Trainer")  
plt.show()
```



```
In [63]: training_abonoment_type
```

```
Out[63]: personal_training  False  True
```

abonoment_type	Personal Training	No Of Gym people Appointed Personal Trainer
Premium	False	226
Premium	True	267
Standard	False	256
Standard	True	251

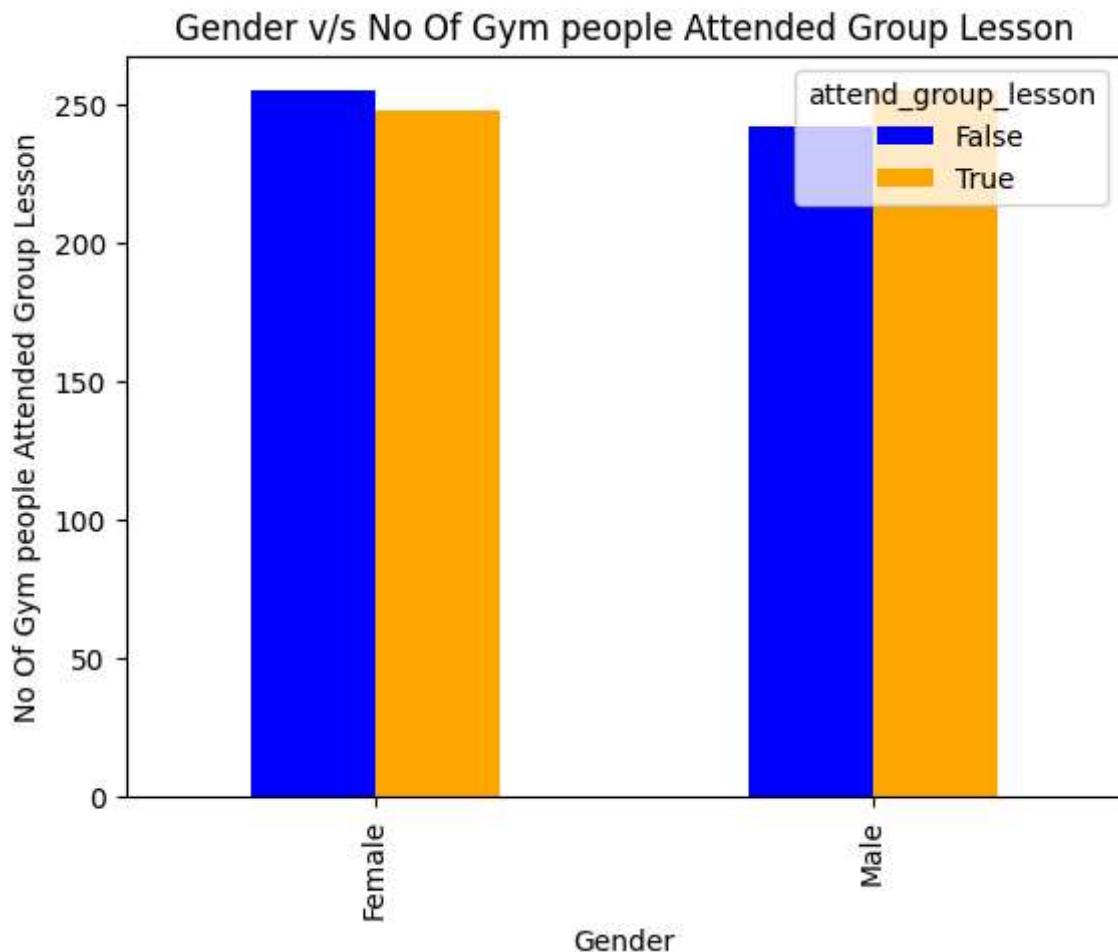
```
In [64]: print("There are most of the people who had Joined Gym with Premium Abonoment Ty
```

There are most of the people who had Joined Gym with Premium Abonoment Type and has also took the Premium Subscription.

```
In [65]: print("There are least of the people who had Joined Gym with Premium Abonoment T
```

There are least of the people who had Joined Gym with Premium Abonoment Type and has not took the Premium Subscription.

```
In [66]: gender_attend_group_lesson=pd.crosstab(gym['gender'].astype(str), gym['attend_group_lesson'], margins=True)
gender_attend_group_lesson.plot(kind='bar', stacked=False, color=['blue', 'orange'])
plt.xlabel("Gender")
plt.ylabel("No Of Gym people Attended Group Lesson")
plt.title("Gender v/s No Of Gym people Attended Group Lesson")
plt.show()
```



```
In [67]: gender_attend_group_lesson
```

```
Out[67]: attend_group_lesson  False  True
```

gender		
Female	255	248
Male	242	255

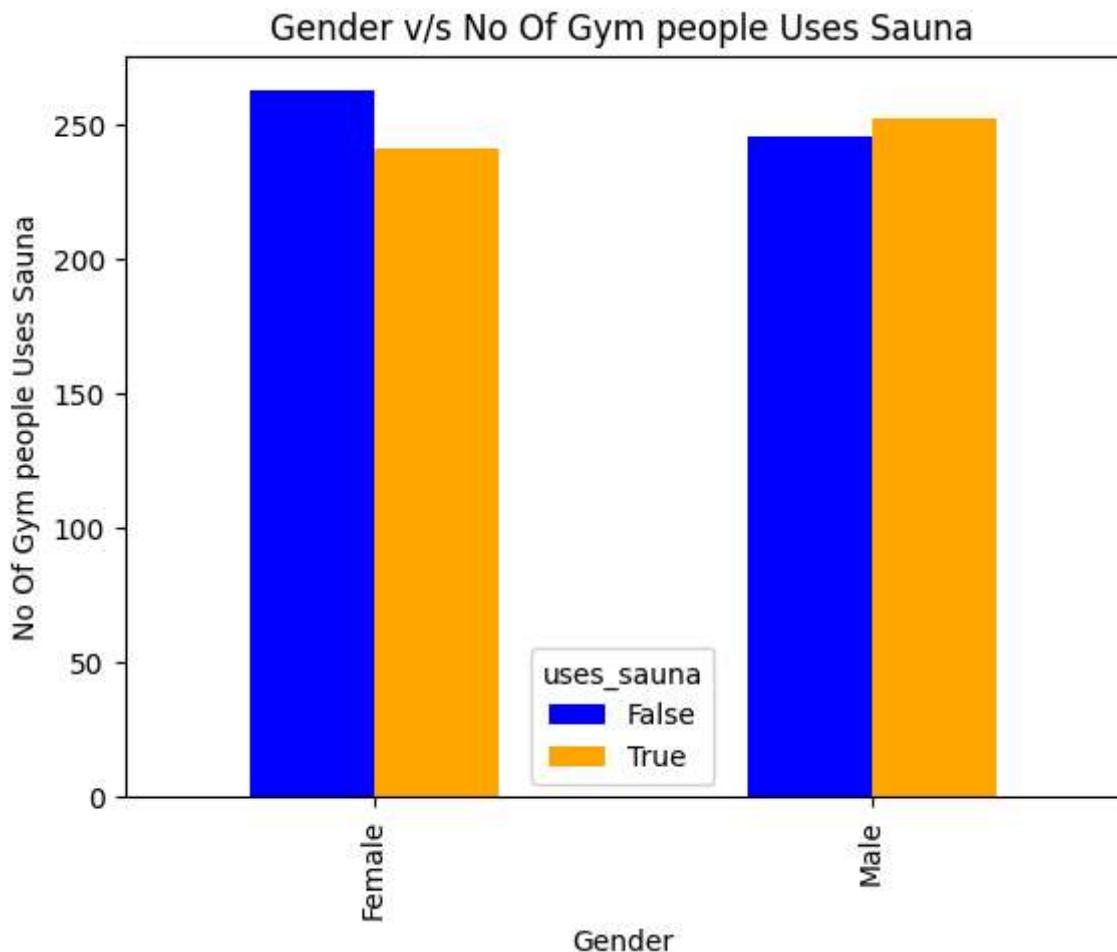
```
In [68]: print("Most of the Males has Not Attended Group Lesson.")
```

Most of the Males has Not Attended Group Lesson.

```
In [69]: print("Number of female attended group lesson and number of male attended Group
```

Number of female attended group lesson and number of male attended Group lesson are same.

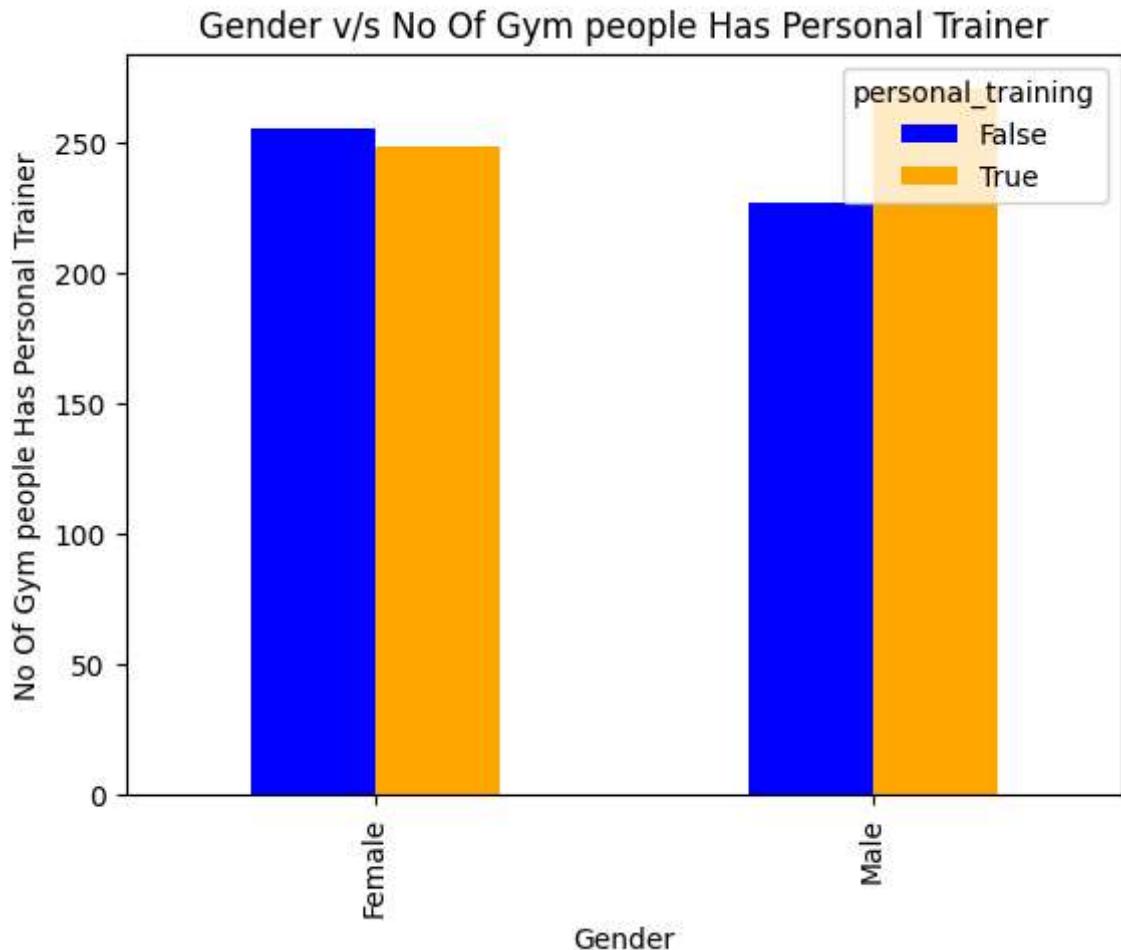
```
In [70]: gender_sauna=pd.crosstab(gym['gender'].astype(str), gym['uses_sauna'].astype(str))
gender_sauna.plot(kind='bar',stacked=False,color=['blue','orange'])
plt.xlabel("Gender")
plt.ylabel("No Of Gym people Uses Sauna")
plt.title("Gender v/s No Of Gym people Uses Sauna")
plt.show()
```



```
In [71]: print("Mens take more Sauna bath compared to Female.")
```

Mens take more Sauna bath compared to Female.

```
In [72]: gender_personal_trainer=pd.crosstab(gym['gender'].astype(str), gym['personal_trainer'])
gender_personal_trainer.plot(kind='bar', stacked=False, color=['blue', 'orange'])
plt.xlabel("Gender")
plt.ylabel("No Of Gym people Has Personal Trainer")
plt.title("Gender v/s No Of Gym people Has Personal Trainer")
plt.show()
```



```
In [73]: gender_personal_trainer
```

```
Out[73]: personal_training  False  True
```

gender		
Female	255	248
Male	227	270

```
In [142...]: print("Males have Hired More Personal trainer than female.")
```

Males have Hired More Personal trainer than female.

FAV Group Lesson

```
In [143...]: gym[['Act 1','Act 2','Act 3']] = gym['fav_group_lesson'].str.split(',', expand=True)
gym.head()
```

Out[143...]

	id	gender	birthday	Age	abonoment_type	visit_per_week	days_per_week	attend_
0	1	Female	1997-04-18	27	Premium	4	Mon, Sat, Tue, Wed	
1	2	Female	1977-09-18	47	Standard	3	Mon, Sat, Wed	
2	3	Male	1983-03-30	41	Premium	1	Sat	
3	4	Male	1980-04-12	44	Premium	3	Sat, Tue, Wed	
4	5	Male	1980-09-10	44	Standard	2	Thu, Wed	

5 rows × 34 columns



In [151...]

```
most_repeated_1 = gym['Act 1'].value_counts()
most_repeated_1
```

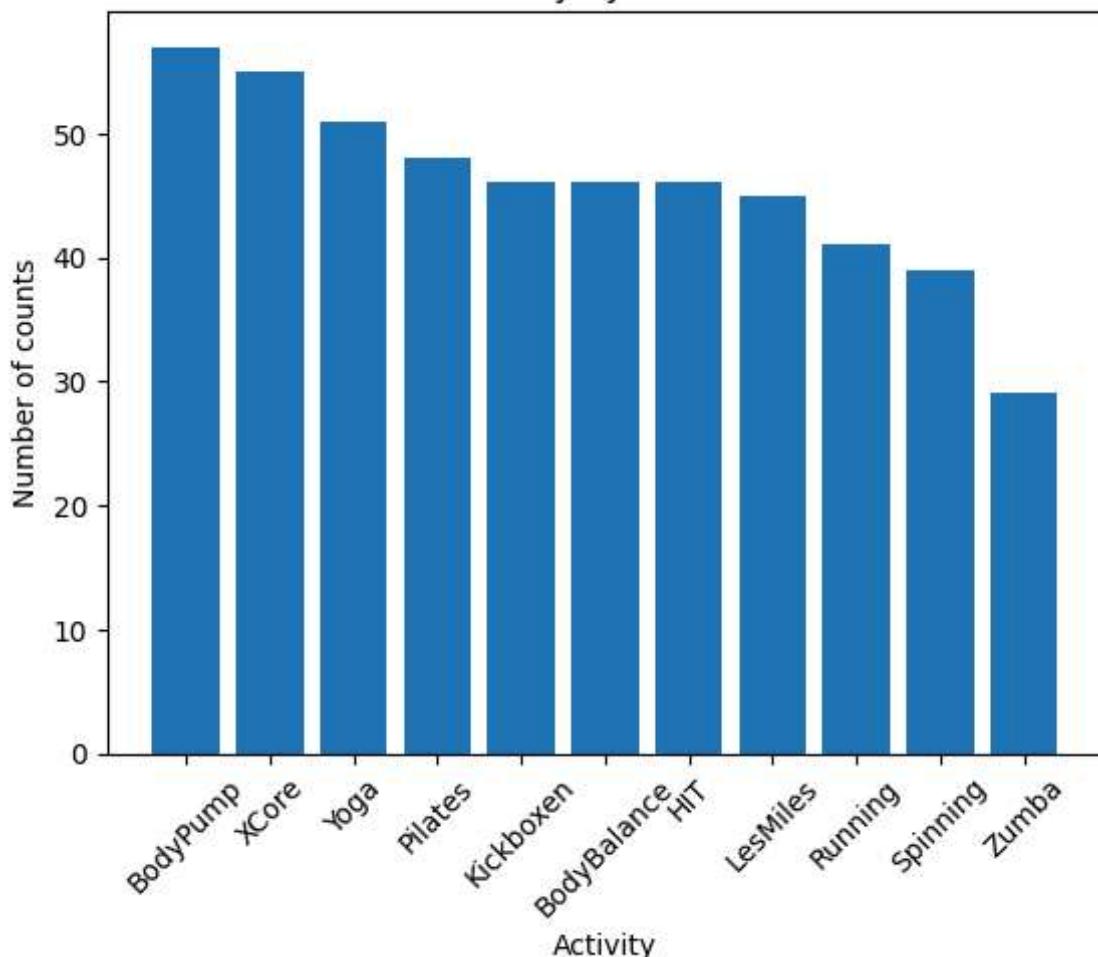
Out[151...]

```
Act 1
BodyPump      57
XCore         55
Yoga          51
Pilates        48
Kickboxen     46
BodyBalance    46
HIT            46
LesMiles       45
Running        41
Spinning        39
Zumba           29
Name: count, dtype: int64
```

In [160...]

```
plt.bar(most_repeated_1.index,most_repeated_1.values)
plt.xticks(rotation=45)
plt.xlabel("Activity")
plt.ylabel("Number of counts")
plt.title("Activity by count 1")
plt.show()
```

Activity by count 1

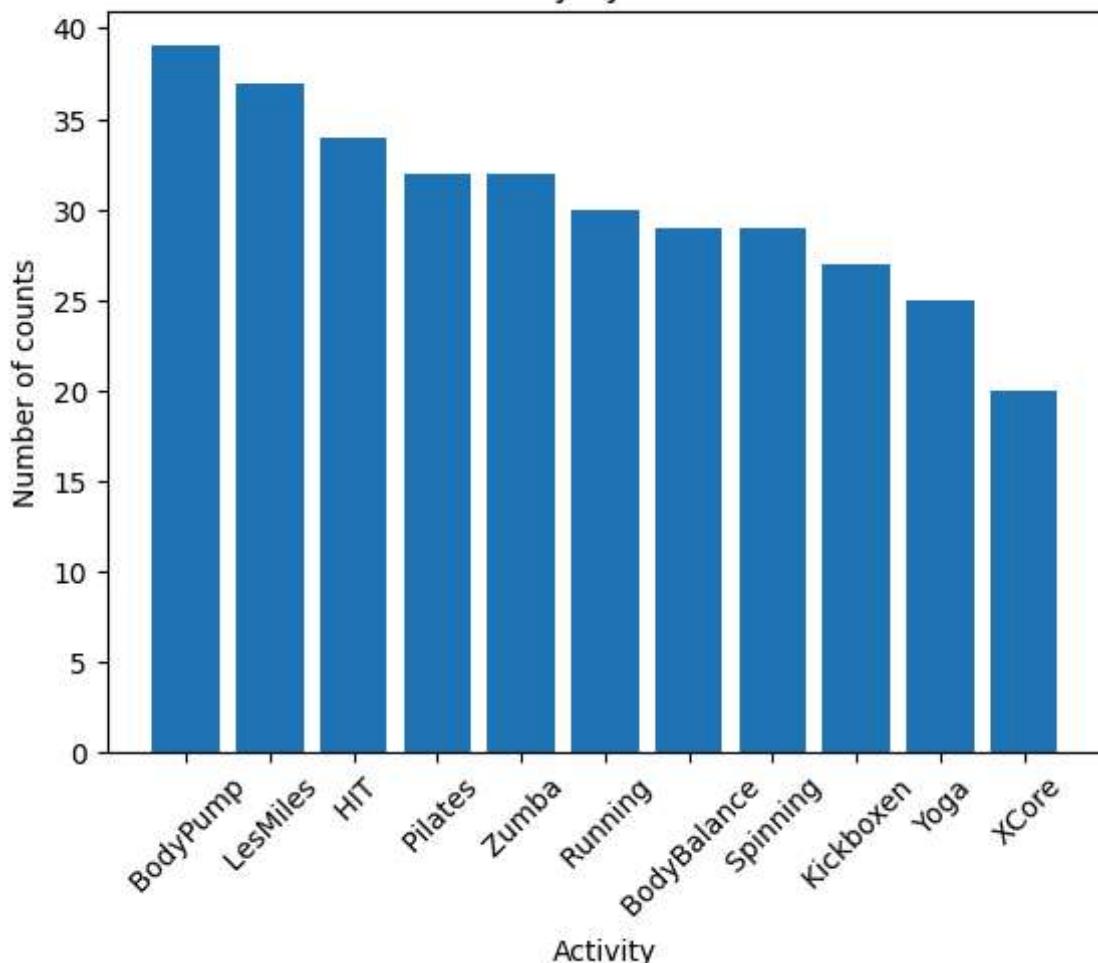


```
In [152...]: most_repeated_2 = gym['Act 2'].value_counts()
most_repeated_2
```

```
Out[152...]: Act 2
BodyPump      39
LesMiles      37
HIT           34
Pilates        32
Zumba          32
Running         30
BodyBalance    29
Spinning        29
Kickboxen      27
Yoga            25
XCore           20
Name: count, dtype: int64
```

```
In [161...]: plt.bar(most_repeated_2.index, most_repeated_2.values)
plt.xticks(rotation=45)
plt.xlabel("Activity")
plt.ylabel("Number of counts")
plt.title("Activity by count 2")
plt.show()
```

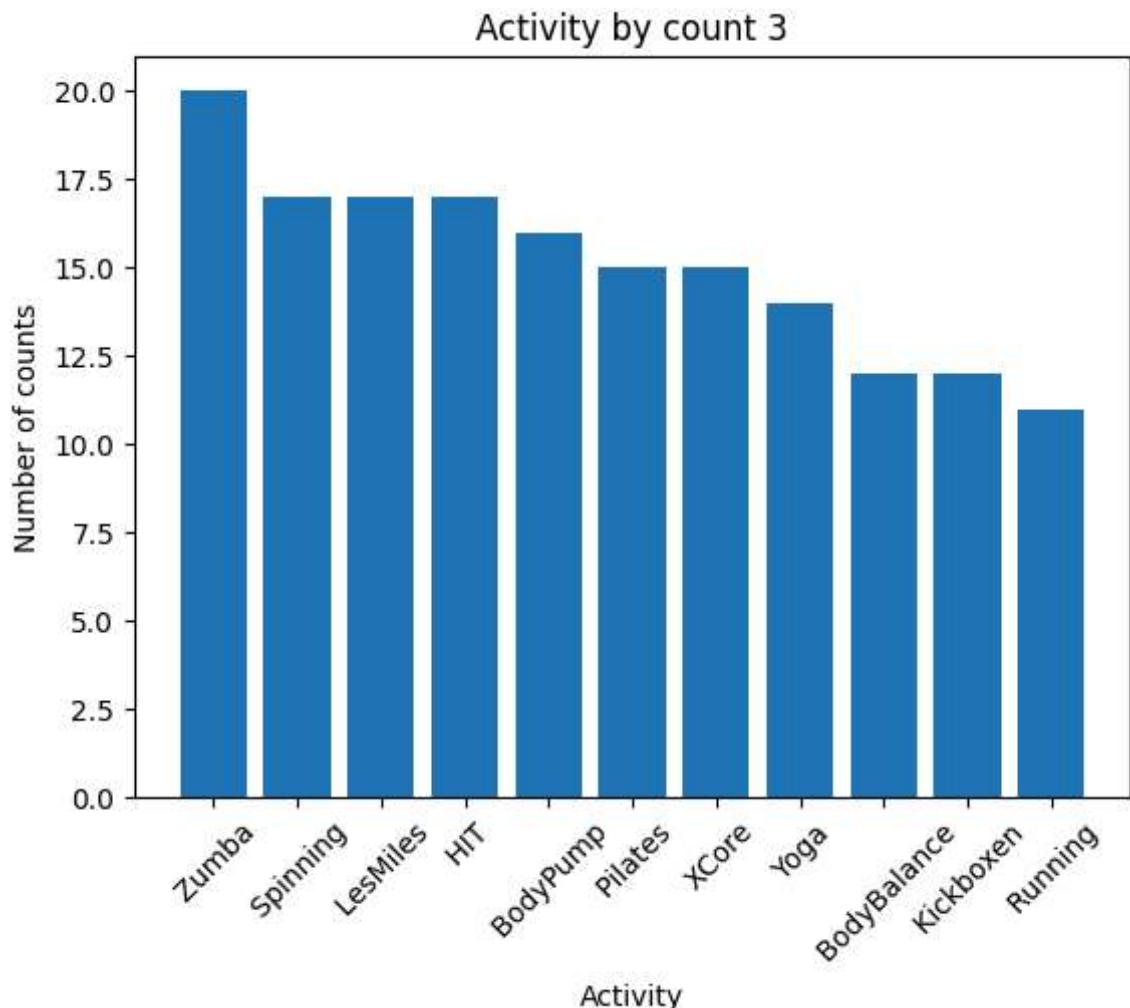
Activity by count 2



```
In [153...]: most_repeated_3 = gym['Act 3'].value_counts()
most_repeated_3
```

```
Out[153...]: Act 3
Zumba      20
Spinning   17
LesMiles   17
HIT        17
BodyPump   16
Pilates    15
XCore      15
Yoga       14
BodyBalance 12
Kickboxen  12
Running    11
Name: count, dtype: int64
```

```
In [162...]: plt.bar(most_repeated_3.index,most_repeated_3.values)
plt.xticks(rotation=45)
plt.xlabel("Activity")
plt.ylabel("Number of counts")
plt.title("Activity by count 3")
plt.show()
```



```
In [163]: print("Most favourite Activity of Gym Member is Bodypump.")
```

Most favourite Activity of Gym Member is Bodypump.

```
In [ ]:
```