Project: Analysis of medical appointments(in Brazil)

Table of Contents

Introduction

No-Show appointments

> Data set consists of data of medical appointments in Brazil, is focused on the question whether people turn up for appointment or not. Characteristics of the data set
>
> 1. Neighbourhood-gives location
> 2. Gender
> 3. Age
> 4. Scholarship
> 5. Hipertension
> 6. No-show-This shows whether appointment taken or not if value is No, then appointment taken and vice-versa

Posing Questions

1. Observing people appointments with respect to scholarship?
2. Based on location and hipertension how people appointments affected?
3. Male vs Female scholarship and its effect on appointment
4. Which age people are not taking appointments mostly with respect to Neighbourhood?
5. People with hipertension,diabetes appointment vs Ppl without hipertension,diabetes appointment

In [1]:
```
# Use this cell to set up import statements for all of the packages that you
#   plan to use.

# Remember to include a 'magic word' so that your visualizations are plotted
#   inline with the notebook. See this page for more:
#   http://ipython.readthedocs.io/en/stable/interactive/magics.html
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
%matplotlib notebook
```

Data Wrangling

> **Tip**: In this section of the report, you will load in the data, check for cleanliness, and then trim and clean your dataset for analysis. Make sure that you document your steps carefully and justify your cleaning decisions.

General Properties

In [2]:
```
#reading the data into a dataframe using pd.read_csv('filepath/name')
data=pd.read_csv('noshowappointments-kagglev2-may-2016.csv')
#checking whether data read correctly or not and data.head() as it returns fir
st five rows here we used it since dataset is big
print(data.head())
```

```
      PatientId  AppointmentID Gender          ScheduledDay  \
0  2.987250e+13        5642903      F  2016-04-29T18:38:08Z
1  5.589978e+14        5642503      M  2016-04-29T16:08:27Z
2  4.262962e+12        5642549      F  2016-04-29T16:19:04Z
3  8.679512e+11        5642828      F  2016-04-29T17:29:31Z
4  8.841186e+12        5642494      F  2016-04-29T16:07:23Z

        AppointmentDay  Age      Neighbourhood  Scholarship  Hipertension  \
0  2016-04-29T00:00:00Z   62    JARDIM DA PENHA            0             1
1  2016-04-29T00:00:00Z   56    JARDIM DA PENHA            0             0
2  2016-04-29T00:00:00Z   62      MATA DA PRAIA            0             0
3  2016-04-29T00:00:00Z    8  PONTAL DE CAMBURI            0             0
4  2016-04-29T00:00:00Z   56    JARDIM DA PENHA            0             1

   Diabetes  Alcoholism  Handcap  SMS_received No-show
0         0           0        0             0      No
1         0           0        0             0      No
2         0           0        0             0      No
3         0           0        0             0      No
4         1           0        0             0      No
```

In [3]:
```
#convert the dataframe data into a data frame that allows access using a uniqu
e value
#this allows to remove duplicates in the data frame
df=data.set_index('PatientId')
print(df.head())
```

```
              AppointmentID Gender          ScheduledDay  \
PatientId
2.987250e+13        5642903      F  2016-04-29T18:38:08Z
5.589978e+14        5642503      M  2016-04-29T16:08:27Z
4.262962e+12        5642549      F  2016-04-29T16:19:04Z
8.679512e+11        5642828      F  2016-04-29T17:29:31Z
8.841186e+12        5642494      F  2016-04-29T16:07:23Z

                    AppointmentDay  Age      Neighbourhood  Scholarship  \
PatientId
2.987250e+13  2016-04-29T00:00:00Z   62    JARDIM DA PENHA            0
5.589978e+14  2016-04-29T00:00:00Z   56    JARDIM DA PENHA            0
4.262962e+12  2016-04-29T00:00:00Z   62      MATA DA PRAIA            0
8.679512e+11  2016-04-29T00:00:00Z    8  PONTAL DE CAMBURI            0
8.841186e+12  2016-04-29T00:00:00Z   56    JARDIM DA PENHA            0

              Hipertension  Diabetes  Alcoholism  Handcap  SMS_received  \
PatientId
2.987250e+13             1         0           0        0             0
5.589978e+14             0         0           0        0             0
4.262962e+12             0         0           0        0             0
8.679512e+11             0         0           0        0             0
8.841186e+12             1         1           0        0             0

             No-show
PatientId
2.987250e+13      No
5.589978e+14      No
4.262962e+12      No
8.679512e+11      No
8.841186e+12      No
```

In [4]:
```
df.columns = df.columns.str.replace('No-show','NoShow')
df.columns
```

Out[4]:
```
Index(['AppointmentID', 'Gender', 'ScheduledDay', 'AppointmentDay', 'Age',
       'Neighbourhood', 'Scholarship', 'Hipertension', 'Diabetes',
       'Alcoholism', 'Handcap', 'SMS_received', 'NoShow'],
      dtype='object')
```

Data Cleaning

In [5]:
```
#getting dataframe with required col's
#refering to the questions posed above the col's are selected
df_investigate=df[['Gender','ScheduledDay','Age','Neighbourhood','Scholarship'
,'Hipertension','Diabetes','NoShow']]
print(df_investigate.head())
```

```
             Gender          ScheduledDay  Age      Neighbourhood  \
PatientId
2.987250e+13      F  2016-04-29T18:38:08Z   62    JARDIM DA PENHA
5.589978e+14      M  2016-04-29T16:08:27Z   56    JARDIM DA PENHA
4.262962e+12      F  2016-04-29T16:19:04Z   62      MATA DA PRAIA
8.679512e+11      F  2016-04-29T17:29:31Z    8  PONTAL DE CAMBURI
8.841186e+12      F  2016-04-29T16:07:23Z   56    JARDIM DA PENHA

              Scholarship  Hipertension  Diabetes NoShow
PatientId
2.987250e+13            0             1         0     No
5.589978e+14            0             0         0     No
4.262962e+12            0             0         0     No
8.679512e+11            0             0         0     No
8.841186e+12            0             1         1     No
```

In [6]:
```
df_investigate.dtypes #checking whether all col's are with proper data types(e
xample : age can't be string but must be int)
```

Out[6]:
```
Gender          object
ScheduledDay    object
Age              int64
Neighbourhood   object
Scholarship      int64
Hipertension     int64
Diabetes         int64
NoShow          object
dtype: object
```

In [7]:
```
#after checking the data types of the col's in the data frame to investigate c
onverting them to appropriate data type
#converting data in the ScheduledDay from dtype object to dtype datatime
df_investigate1=df_investigate[df_investigate['ScheduledDay']==pd.to_datetime(
df['ScheduledDay'])].copy()
df_investigate.head()
```

Out[7]:

PatientId	Gender	ScheduledDay	Age	Neighbourhood	Scholarship	Hipertension
2.987250e+13	F	2016-04-29T18:38:08Z	62	JARDIM DA PENHA	0	1
5.589978e+14	M	2016-04-29T16:08:27Z	56	JARDIM DA PENHA	0	0
4.262962e+12	F	2016-04-29T16:19:04Z	62	MATA DA PRAIA	0	0
8.679512e+11	F	2016-04-29T17:29:31Z	8	PONTAL DE CAMBURI	0	0
8.841186e+12	F	2016-04-29T16:07:23Z	56	JARDIM DA PENHA	0	1

In [14]:
```
#knowing the data types of the data in the col's after conversion into proper
 dtypes
df_investigate.dtypes
```

Out[14]:
```
Gender           object
ScheduledDay     object
Age               int64
Neighbourhood    object
Scholarship       int64
Hipertension      int64
Diabetes          int64
NoShow           object
dtype: object
```

In [15]:
```
#removing noisy data ,here as of interest we got data Frame with required co
l's
#age can't be negative so removing such noisy data if any this is achieved as
 follows
df_investigate=df_investigate[df_investigate['Age']>=0]
#performing a comparision on  a col and obtaining the filtered data
```

In [16]:
```
# After discussing the structure of the data and any problems that need to
 be
#   cleaned, perform those cleaning steps in the second part of this sectio
n.
#Create a new function:
def num_missing(x):
    return sum(x.isnull())

#Applying per column:
print("Missing values per column:")
print(df_investigate.apply(num_missing, axis=0)) #axis=0 defines that funct
ion is to be applied on each column

#Applying per row:
#print ("\nMissing values per row:")
#print (df_investigate.apply(num_missing, axis=1).head())
df_investigate.head()
```

```
Missing values per column:
Gender          0
ScheduledDay    0
Age             0
Neighbourhood   0
Scholarship     0
Hipertension    0
Diabetes        0
NoShow          0
dtype: int64
```

Out[16]:

PatientId	Gender	ScheduledDay	Age	Neighbourhood	Scholarship	Hipertension
2.987250e+13	F	2016-04-29T18:38:08Z	62	JARDIM DA PENHA	0	1
5.589978e+14	M	2016-04-29T16:08:27Z	56	JARDIM DA PENHA	0	0
4.262962e+12	F	2016-04-29T16:19:04Z	62	MATA DA PRAIA	0	0
8.679512e+11	F	2016-04-29T17:29:31Z	8	PONTAL DE CAMBURI	0	0
8.841186e+12	F	2016-04-29T16:07:23Z	56	JARDIM DA PENHA	0	1

from the above finding no missing values hence imputing missing values is not necessary

Exploratory Data Analysis

> **Tip**: Now that you've trimmed and cleaned your data, you're ready to move on to exploration. Compute statistics and create visualizations with the goal of addressing the research questions that you posed in the Introduction section. It is recommended that you be systematic with your approach. Look at one variable at a time, and then follow it up by looking at relationships between variables.

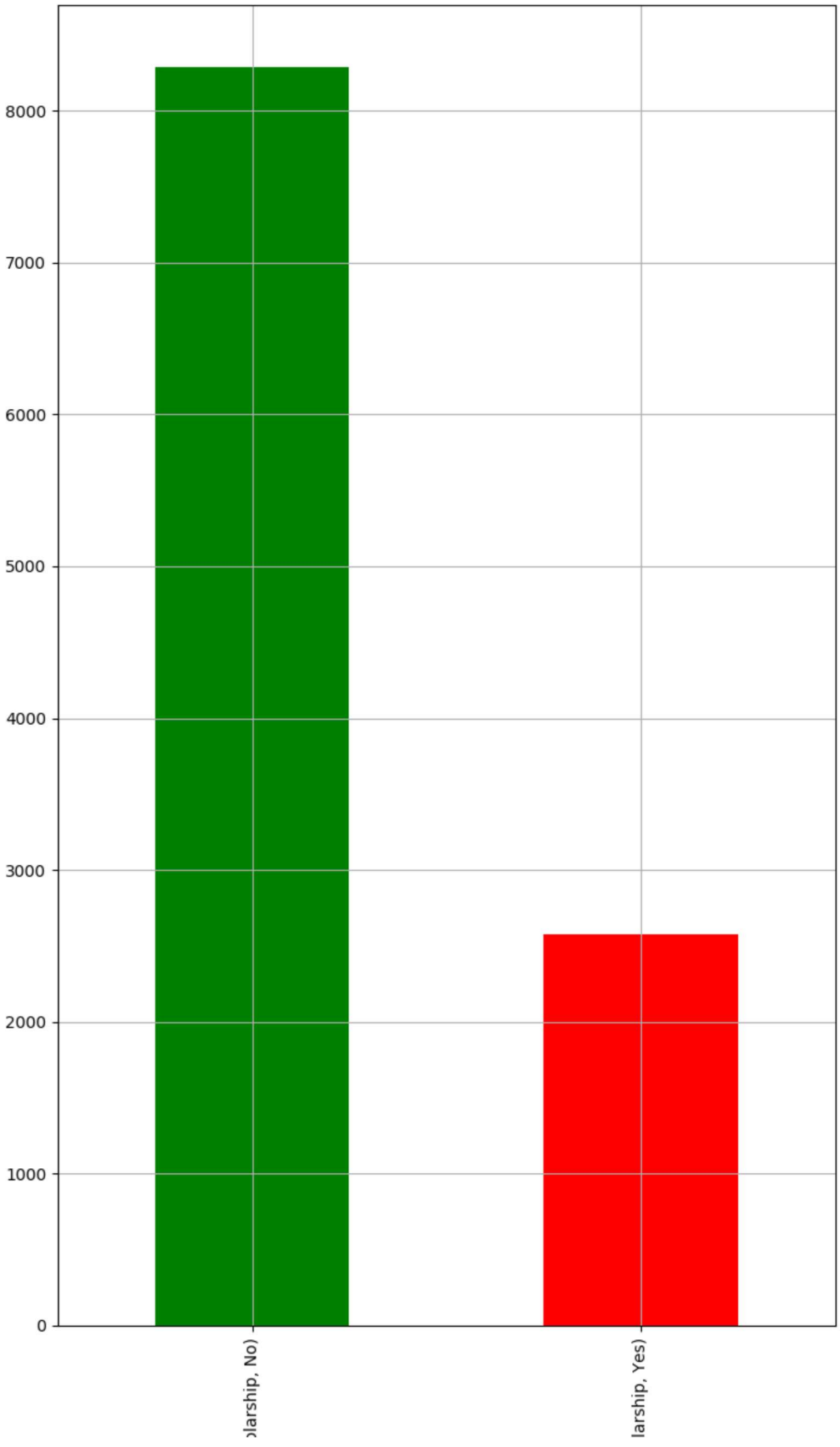
In [17]:
```
# Use this, and more code cells, to explore your data. Don't forget to add
# Markdown cells to document your observations and findings.
df_show=df_investigate[['Gender','Neighbourhood','Scholarship','NoShow','Hiper
tension']] #getting dataframe with required features
print(df_show.head())
```

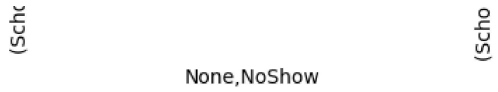
```
              Gender      Neighbourhood  Scholarship NoShow  Hipertension
PatientId
2.987250e+13      F    JARDIM DA PENHA            0     No             1
5.589978e+14      M    JARDIM DA PENHA            0     No             0
4.262962e+12      F      MATA DA PRAIA            0     No             0
8.679512e+11      F  PONTAL DE CAMBURI            0     No             0
8.841186e+12      F    JARDIM DA PENHA            0     No             1
```

Based on Scholarship ,how appointment is affected ?

> **Parameters**
>
> 1. Scholarship - Independent variable
> 2. Appointment(NoShow)-dependent variable on above variable

In [18]:
```
#data frame of interested investigation
df_Scholarship=df_show[['Scholarship','NoShow']]
#to visualize the scholarship with No-->NoShow and yes-->NoShow
view_Scholarship=df_Scholarship.groupby(['NoShow']).sum()
view_Scholarship.unstack().plot(kind='bar',stacked=True,  color=['green','red'
], grid=True)
```


8000
7000
6000
5000
4000
3000
2000
1000
0
larship, No)
larship, Yes)



Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x2a0e16d1128>

In [19]:
```
#data Frame with group by NoShow
var_df_Scholarship1=df_Scholarship.groupby(['NoShow'],as_index=False).sum()
#getting the probability of scholarship making to turn up for appointment
print('probability of a person to turnup for appointment on getting Scholarshi
p : ',(var_df_Scholarship1[var_df_Scholarship1['NoShow']=='No']['Scholarship']
.sum())/var_df_Scholarship1['Scholarship'].sum())
```

```
probability of a person to turnup for appointment on getting Scholarship :
0.762636957923
```

Based on Neighbourhood and Scholarship ,how appointment is affected ?

Parameters

1. Neighbourhood - Independent variable
2. Scholarship - Independent variable
3. Appointment(NoShow)-dependent variable on above variables

In [20]:
```
df_location_Scholarship=df_show[['Neighbourhood','Scholarship','NoShow']]
print(len(df_location_Scholarship))
#view_loc_Scholar=df_location_Scholarship.groupby(['Neighbourhood','NoShow'],as_index=False).sum()
view_loc_Scholar = df_location_Scholarship.groupby( [ "Neighbourhood", "NoShow",'Scholarship'],as_index=False )
#adding a new column count as NoShow is categorical and here we take count of similar values
g=pd.DataFrame(view_loc_Scholar.size().reset_index(name = "Count"))
g
```

```
110526
```

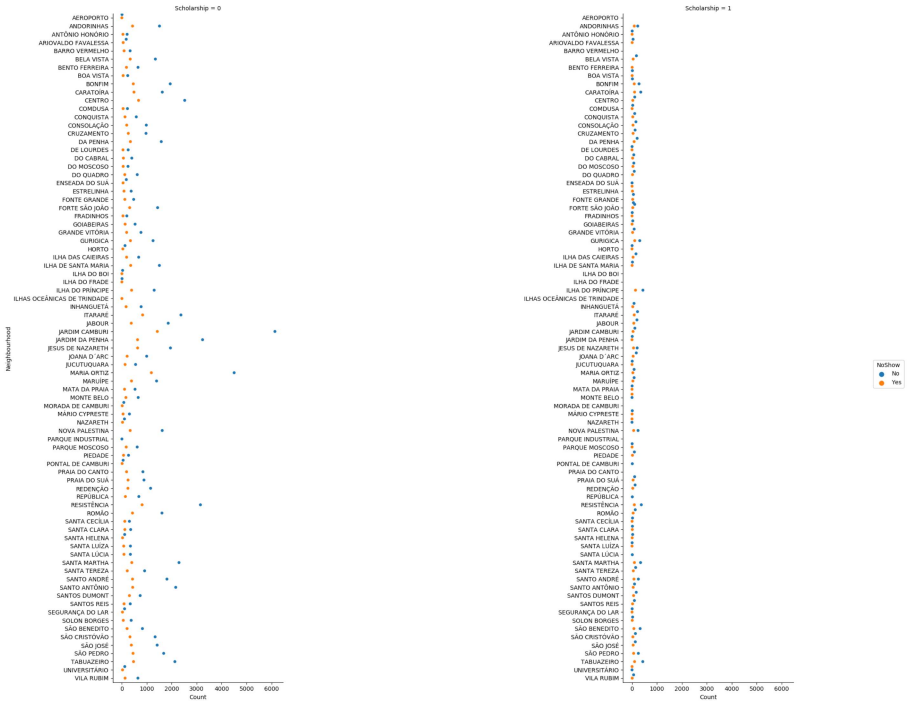
Out[20]:

	Neighbourhood	NoShow	Scholarship	Count
0	AEROPORTO	No	0	7
1	AEROPORTO	Yes	0	1
2	ANDORINHAS	No	0	1510
3	ANDORINHAS	No	1	231
4	ANDORINHAS	Yes	0	429
5	ANDORINHAS	Yes	1	92
6	ANTÔNIO HONÓRIO	No	0	212
7	ANTÔNIO HONÓRIO	No	1	9
8	ANTÔNIO HONÓRIO	Yes	0	45
9	ANTÔNIO HONÓRIO	Yes	1	5
10	ARIOVALDO FAVALESSA	No	0	176
11	ARIOVALDO FAVALESSA	No	1	44
12	ARIOVALDO FAVALESSA	Yes	0	54
13	ARIOVALDO FAVALESSA	Yes	1	8
14	BARRO VERMELHO	No	0	332
15	BARRO VERMELHO	Yes	0	91
16	BELA VISTA	No	0	1345
17	BELA VISTA	No	1	178
18	BELA VISTA	Yes	0	337
19	BELA VISTA	Yes	1	47
20	BENTO FERREIRA	No	0	649
21	BENTO FERREIRA	No	1	16
22	BENTO FERREIRA	Yes	0	186
23	BENTO FERREIRA	Yes	1	7
24	BOA VISTA	No	0	237
25	BOA VISTA	No	1	17
26	BOA VISTA	Yes	0	52
27	BOA VISTA	Yes	1	6
28	BONFIM	No	0	1940
29	BONFIM	No	1	283
...	...	...	...	...
273	SOLON BORGES	Yes	0	58

	Neighbourhood	NoShow	Scholarship	Count
274	SOLON BORGES	Yes	1	11
275	SÃO BENEDITO	No	0	825
276	SÃO BENEDITO	No	1	327
277	SÃO BENEDITO	Yes	0	210
278	SÃO BENEDITO	Yes	1	77
279	SÃO CRISTÓVÃO	No	0	1335
280	SÃO CRISTÓVÃO	No	1	138
281	SÃO CRISTÓVÃO	Yes	0	327
282	SÃO CRISTÓVÃO	Yes	1	36
283	SÃO JOSÉ	No	0	1416
284	SÃO JOSÉ	No	1	133
285	SÃO JOSÉ	Yes	0	381
286	SÃO JOSÉ	Yes	1	47
287	SÃO PEDRO	No	0	1678
288	SÃO PEDRO	No	1	255
289	SÃO PEDRO	Yes	0	449
290	SÃO PEDRO	Yes	1	66
291	TABUAZEIRO	No	0	2126
292	TABUAZEIRO	No	1	433
293	TABUAZEIRO	Yes	0	469
294	TABUAZEIRO	Yes	1	104
295	UNIVERSITÁRIO	No	0	118
296	UNIVERSITÁRIO	No	1	2
297	UNIVERSITÁRIO	Yes	0	29
298	UNIVERSITÁRIO	Yes	1	3
299	VILA RUBIM	No	0	645
300	VILA RUBIM	No	1	65
301	VILA RUBIM	Yes	0	131
302	VILA RUBIM	Yes	1	10

303 rows × 4 columns

In [21]:
```
sns.factorplot(x="Count", y="Neighbourhood", hue="NoShow", data=g, col ="Scholarship", kind="swarm")
#requesting to view the image --> in new tab for proper view (open in .ipynb )
```



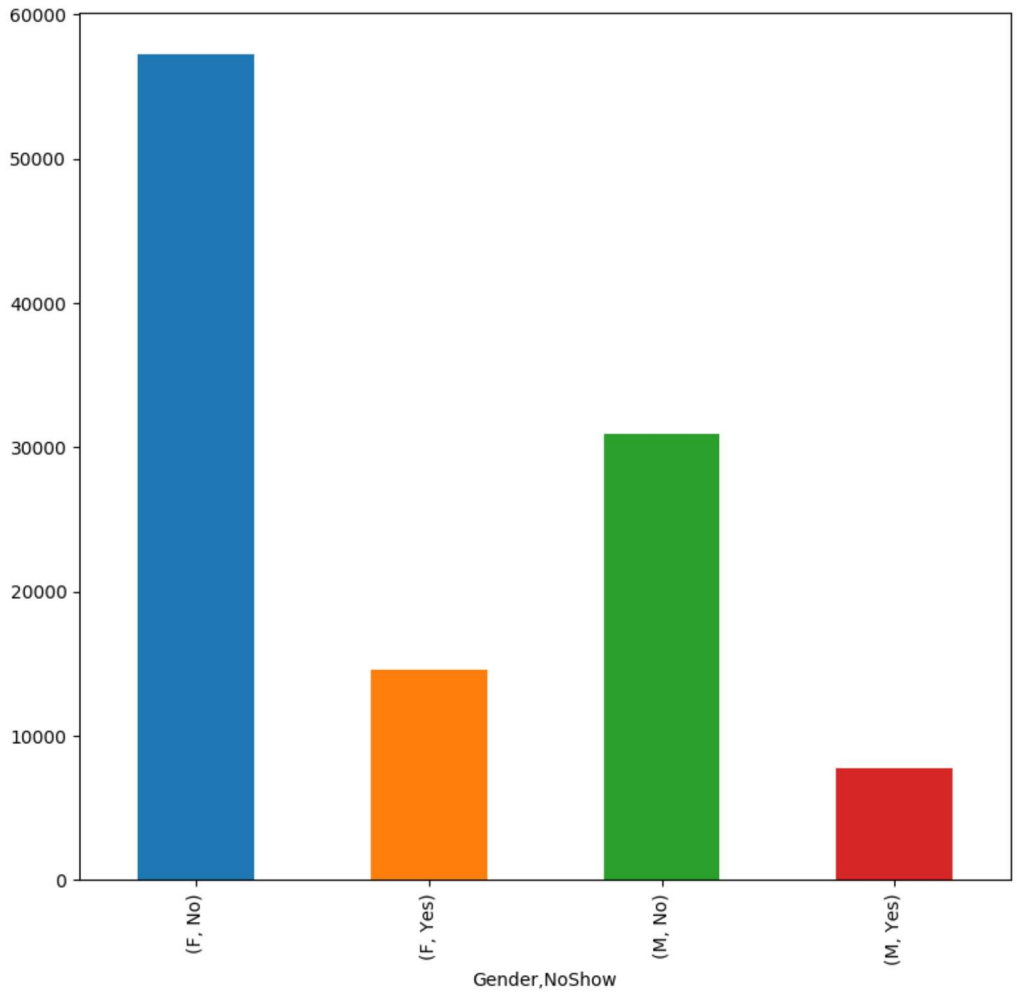
Out[21]: <seaborn.axisgrid.FacetGrid at 0x2a0e16d16d8>

MALE Vs FEMALE, Scholarship and its effect on NoShow Appointment

Parameters

1. Scholarship - Independent variable
2. Gender -Independent Variable
3. Appointment(NoShow)-dependent variable on above variables

In [22]:
```
#getting data frame to investigate
df_gender_Scholarship=df_show[['Gender','Scholarship','NoShow']]
#grouping data frame based on gender,NoShow and ploting a histogram
df_gender_Scholarship.groupby(['Gender','NoShow']).size().plot(kind='bar')
```



Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x2a0e16d13c8>

In [23]:
```
var_df_gender=df_gender_Scholarship.groupby(['Gender','NoShow'],as_index=False
).size()
#getting the probability of scholarship making to turn up for appointment
var_df_gender
```

Out[23]:
```
Gender  NoShow
F       No        57245
        Yes       14594
M       No        30962
        Yes        7725
dtype: int64
```

In [24]:
```
#female patients probability of taking appointment
p=57245+14594
p=57245/p
print('female patients probability of taking appointment : ',p)
q=30962+7725
q=30962/q
print('male patients probability of taking appointment : ',q)
```

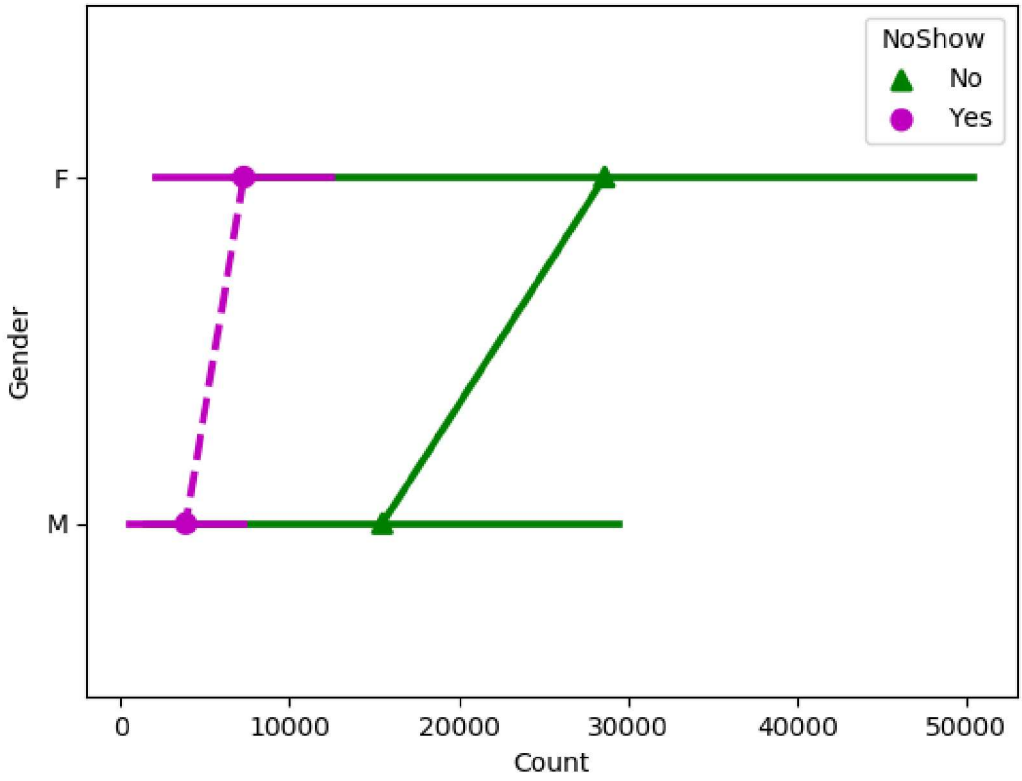
```
female patients probability of taking appointment :  0.7968512924734475
male patients probability of taking appointment :  0.8003205211052808
```

In [25]:
```
view_Gen_Scholar = df_gender_Scholarship.groupby( [ "Gender", "NoShow",'Schola
rship'],as_index=False )
g_df=pd.DataFrame(view_Gen_Scholar.size().reset_index(name = "Count"))
g_df
```

Out[25]:

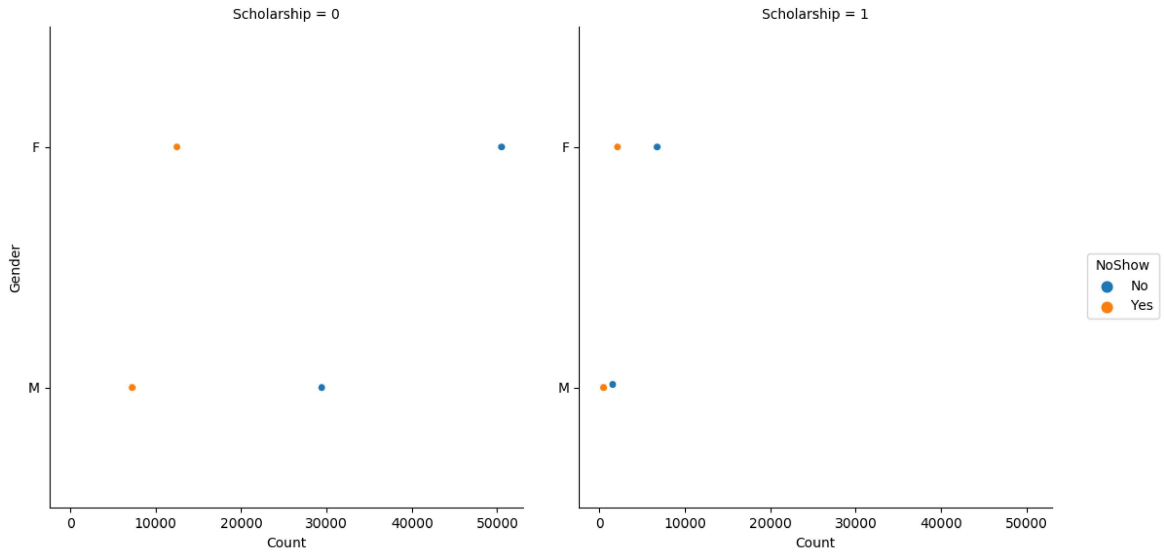
	Gender	NoShow	Scholarship	Count
0	F	No	0	50498
1	F	No	1	6747
2	F	Yes	0	12488
3	F	Yes	1	2106
4	M	No	0	29426
5	M	No	1	1536
6	M	Yes	0	7253
7	M	Yes	1	472

In [26]:
```
sns.pointplot(x="Count", y="Gender", hue="NoShow", data=g_df,palette={"No":
"g", "Yes": "m"},markers=["^", "o"], linestyles=["-", "--"])
```



Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x2a0e5496b38>

In [27]:
```
#to know how scholarship effects appointment based on gender
#plot a swarmplot using seaborn
sns.factorplot(x="Count", y="Gender", hue="NoShow", data=g_df, col ="Scholarship", kind="swarm")
```



Out[27]: <seaborn.axisgrid.FacetGrid at 0x2a0e47f8908>

Conclusions

Findings

Based on Neighbourhood and Scholarship ,how appointment is affected ?

findings Based on Scholarship as independent variable and NoShow as dependent

following observation has been made :

- Individual who is getting Scholarship there is a probabiity of 0.76 that he/she takes appointment

findings based on Neighbourhood , Scholarship and NoShow

from the visuvalization,

- No Location is getting scholarship for more than 1000 persons
- Many Locations are having almost less than 1000 persons that dont turnup for appointment who are not getting any Scholarship
- Highest no.of persons turnup for appointment even though no Scholarship (considering single location) is around 6000

MALE Vs FEMALE Scholarship based on Neighbourhood and its effect on NoShow Appointment

From the Visualizations,

- No.of Female patients are greater than no.of male patients
- Female patients dont turn up for appointment comparitively to male patients
- Patients who are getting scholarship that dont turnup for appointment are more females than males

REFERENCES

- https://www.analyticsvidhya.com/blog/2015/05/data-visualization-python/
- https://stackoverflow.com/questions/36367986/how-to-make-inline-plots-in-jupyter-notebook-larger
- https://seaborn.pydata.org/tutorial/categorical.html