

da-on-healthcare-appointment-data

June 12, 2025

```
[1]: from google.colab import files
      uploaded = files.upload()
```

<IPython.core.display.HTML object>

Saving Data.csv to Data.csv

```
[2]: import pandas as pd
      import numpy as np
      import datetime
      from time import strftime
      import matplotlib.pyplot as plt
      %matplotlib inline

      import seaborn as sns
```

```
[3]: # Reading the datasett

      base_data = pd.read_csv('Data.csv')
```

```
[4]: base_data
```

```
[4]:
```

| | 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 | |
| ... | ... | ... | ... | ... | |
| 110522 | 2.572134e+12 | 5651768 | F | 2016-05-03T09:15:35Z | |
| 110523 | 3.596266e+12 | 5650093 | F | 2016-05-03T07:27:33Z | |
| 110524 | 1.557663e+13 | 5630692 | F | 2016-04-27T16:03:52Z | |
| 110525 | 9.213493e+13 | 5630323 | F | 2016-04-27T15:09:23Z | |
| 110526 | 3.775115e+14 | 5629448 | F | 2016-04-27T13:30:56Z | |

| | AppointmentDay | Age | Neighbourhood | Scholarship | \ |
|---|----------------------|-----|-----------------|-------------|---|
| 0 | 2016-04-29T00:00:00Z | 62 | JARDIM DA PENHA | 0 | |
| 1 | 2016-04-29T00:00:00Z | 56 | JARDIM DA PENHA | 0 | |

| | | | | |
|--------|----------------------|-----|-------------------|-----|
| 2 | 2016-04-29T00:00:00Z | 62 | MATA DA PRAIA | 0 |
| 3 | 2016-04-29T00:00:00Z | 8 | PONTAL DE CAMBURI | 0 |
| 4 | 2016-04-29T00:00:00Z | 56 | JARDIM DA PENHA | 0 |
| ... | ... | ... | ... | ... |
| 110522 | 2016-06-07T00:00:00Z | 56 | MARIA ORTIZ | 0 |
| 110523 | 2016-06-07T00:00:00Z | 51 | MARIA ORTIZ | 0 |
| 110524 | 2016-06-07T00:00:00Z | 21 | MARIA ORTIZ | 0 |
| 110525 | 2016-06-07T00:00:00Z | 38 | MARIA ORTIZ | 0 |
| 110526 | 2016-06-07T00:00:00Z | 54 | MARIA ORTIZ | 0 |

| | Hipertension | Diabetes | Alcoholism | Handcap | SMS_received | No-show |
|--------|--------------|----------|------------|---------|--------------|---------|
| 0 | 1 | 0 | 0 | 0 | 0 | No |
| 1 | 0 | 0 | 0 | 0 | 0 | No |
| 2 | 0 | 0 | 0 | 0 | 0 | No |
| 3 | 0 | 0 | 0 | 0 | 0 | No |
| 4 | 1 | 1 | 0 | 0 | 0 | No |
| ... | ... | ... | ... | ... | ... | ... |
| 110522 | 0 | 0 | 0 | 0 | 1 | No |
| 110523 | 0 | 0 | 0 | 0 | 1 | No |
| 110524 | 0 | 0 | 0 | 0 | 1 | No |
| 110525 | 0 | 0 | 0 | 0 | 1 | No |
| 110526 | 0 | 0 | 0 | 0 | 1 | No |

[110527 rows x 14 columns]

```
[5]: base_data.shape #how many rows and columns data set has
```

```
[5]: (110527, 14)
```

```
[6]: #info gives the information about data frame
base_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   PatientId             110527 non-null float64
1   AppointmentID         110527 non-null int64
2   Gender                110527 non-null object
3   ScheduledDay          110527 non-null object
4   AppointmentDay        110527 non-null object
5   Age                   110527 non-null int64
6   Neighbourhood         110527 non-null object
7   Scholarship           110527 non-null int64
8   Hipertension          110527 non-null int64
9   Diabetes              110527 non-null int64
```

```

10 Alcoholism      110527 non-null  int64
11 Handcap         110527 non-null  int64
12 SMS_received    110527 non-null  int64
13 No-show         110527 non-null  object
dtypes: float64(1), int64(8), object(5)
memory usage: 11.8+ MB

```

```

[7]: #modifying the date and time into standard form
# Derived matrix converting object time into a date time
base_data['ScheduledDay'] = pd.to_datetime(base_data['ScheduledDay']).dt.date.
    ↳astype('datetime64[ns]')
base_data['AppointmentDay'] = pd.to_datetime(base_data['AppointmentDay']).dt.
    ↳date.astype('datetime64[ns]')

```

```

[8]: base_data.head(5)

```

```

[8]:      PatientId  AppointmentID  Gender  ScheduledDay  AppointmentDay  Age  \
0  2.987250e+13      5642903      F    2016-04-29    2016-04-29    62
1  5.589978e+14      5642503      M    2016-04-29    2016-04-29    56
2  4.262962e+12      5642549      F    2016-04-29    2016-04-29    62
3  8.679512e+11      5642828      F    2016-04-29    2016-04-29     8
4  8.841186e+12      5642494      F    2016-04-29    2016-04-29    56

      Neighbourhood  Scholarship  Hipertension  Diabetes  Alcoholism  \
0  JARDIM DA PENHA           0           1           0           0
1  JARDIM DA PENHA           0           0           0           0
2  MATA DA PRAIA           0           0           0           0
3  PONTAL DE CAMBURI        0           0           0           0
4  JARDIM DA PENHA           0           1           1           0

      Handcap  SMS_received  No-show
0           0           0      No
1           0           0      No
2           0           0      No
3           0           0      No
4           0           0      No

```

for the schedule day and appointment day storing the weekdays only into a variable

```

[9]: # 5 is Saturday, 6 is Sunday

base_data['sch_weekday'] = base_data['ScheduledDay'].dt.dayofweek

```

```

[10]: base_data['app_weekday'] = base_data['AppointmentDay'].dt.dayofweek

```

```

[11]: base_data['sch_weekday'].value_counts()

```

```
[11]: sch_weekday
      1    26168
      2    24262
      0    23085
      4    18915
      3    18073
      5         24
      Name: count, dtype: int64
```

```
[12]: base_data['app_weekday'].value_counts()
```

```
[12]: app_weekday
      2    25867
      1    25640
      0    22715
      4    19019
      3    17247
      5         39
      Name: count, dtype: int64
```

```
[13]: base_data.columns
```

```
[13]: Index(['PatientId', 'AppointmentID', 'Gender', 'ScheduledDay',
            'AppointmentDay', 'Age', 'Neighbourhood', 'Scholarship', 'Hiptertension',
            'Diabetes', 'Alcoholism', 'Handcap', 'SMS_received', 'No-show',
            'sch_weekday', 'app_weekday'],
            dtype='object')
```

```
[14]: #changing the name of some cloumns
base_data= base_data.rename(columns={'Hiptertension': 'Hypertension', 'Handcap': 'Handicap', 'SMS_received': 'SMSReceived', 'No-show': 'NoShow'})
```

```
[15]: base_data.columns
```

```
[15]: Index(['PatientId', 'AppointmentID', 'Gender', 'ScheduledDay',
            'AppointmentDay', 'Age', 'Neighbourhood', 'Scholarship', 'Hypertension',
            'Diabetes', 'Alcoholism', 'Handicap', 'SMSReceived', 'NoShow',
            'sch_weekday', 'app_weekday'],
            dtype='object')
```

```
[16]: base_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   PatientId             110527 non-null float64
```

```

1 AppointmentID 110527 non-null int64
2 Gender        110527 non-null object
3 ScheduledDay  110527 non-null datetime64[ns]
4 AppointmentDay 110527 non-null datetime64[ns]
5 Age          110527 non-null int64
6 Neighbourhood 110527 non-null object
7 Scholarship   110527 non-null int64
8 Hypertension  110527 non-null int64
9 Diabetes      110527 non-null int64
10 Alcoholism   110527 non-null int64
11 Handicap     110527 non-null int64
12 SMSReceived  110527 non-null int64
13 NoShow       110527 non-null object
14 sch_weekday  110527 non-null int32
15 app_weekday  110527 non-null int32
dtypes: datetime64[ns](2), float64(1), int32(2), int64(8), object(3)
memory usage: 12.6+ MB

```

```

[17]: # dropping some columns which have no significance
base_data.drop(['PatientId', 'AppointmentID', 'Neighbourhood'], axis=1,
               ↪inplace=True)

```

```
[18]: base_data
```

```

[18]:
      Gender ScheduledDay AppointmentDay Age Scholarship Hypertension \
0         F   2016-04-29   2016-04-29   62             0           1
1         M   2016-04-29   2016-04-29   56             0           0
2         F   2016-04-29   2016-04-29   62             0           0
3         F   2016-04-29   2016-04-29    8             0           0
4         F   2016-04-29   2016-04-29   56             0           1
...
110522    F   2016-05-03   2016-06-07   56             0           0
110523    F   2016-05-03   2016-06-07   51             0           0
110524    F   2016-04-27   2016-06-07   21             0           0
110525    F   2016-04-27   2016-06-07   38             0           0
110526    F   2016-04-27   2016-06-07   54             0           0

      Diabetes Alcoholism Handicap SMSReceived NoShow sch_weekday \
0             0           0         0           0    No           4
1             0           0         0           0    No           4
2             0           0         0           0    No           4
3             0           0         0           0    No           4
4             1           0         0           0    No           4
...
110522    0           0         0           1    No           1
110523    0           0         0           1    No           1
110524    0           0         0           1    No           2

```

| | | | | | | |
|--------|---|---|---|---|----|---|
| 110525 | 0 | 0 | 0 | 1 | No | 2 |
| 110526 | 0 | 0 | 0 | 1 | No | 2 |

| | |
|--------|-------------|
| | app_weekday |
| 0 | 4 |
| 1 | 4 |
| 2 | 4 |
| 3 | 4 |
| 4 | 4 |
| ... | ... |
| 110522 | 1 |
| 110523 | 1 |
| 110524 | 1 |
| 110525 | 1 |
| 110526 | 1 |

[110527 rows x 13 columns]

```
[19]: base_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Gender                 110527 non-null object
1   ScheduledDay            110527 non-null datetime64[ns]
2   AppointmentDay          110527 non-null datetime64[ns]
3   Age                    110527 non-null int64
4   Scholarship             110527 non-null int64
5   Hypertension            110527 non-null int64
6   Diabetes                110527 non-null int64
7   Alcoholism              110527 non-null int64
8   Handicap                110527 non-null int64
9   SMSReceived             110527 non-null int64
10  NoShow                  110527 non-null object
11  sch_weekday              110527 non-null int32
12  app_weekday              110527 non-null int32
dtypes: datetime64[ns](2), int32(2), int64(7), object(2)
memory usage: 10.1+ MB
```

```
[20]: base_data.describe()
```

```
[20]:
```

| | | | |
|-------|-------------------------------|-------------------------------|---|
| | ScheduledDay | AppointmentDay | \ |
| count | 110527 | 110527 | |
| mean | 2016-05-08 20:33:18.179630080 | 2016-05-19 00:57:50.008233472 | |
| min | 2015-11-10 00:00:00 | 2016-04-29 00:00:00 | |

| | | |
|-----|---------------------|---------------------|
| 25% | 2016-04-29 00:00:00 | 2016-05-09 00:00:00 |
| 50% | 2016-05-10 00:00:00 | 2016-05-18 00:00:00 |
| 75% | 2016-05-20 00:00:00 | 2016-05-31 00:00:00 |
| max | 2016-06-08 00:00:00 | 2016-06-08 00:00:00 |
| std | NaN | NaN |

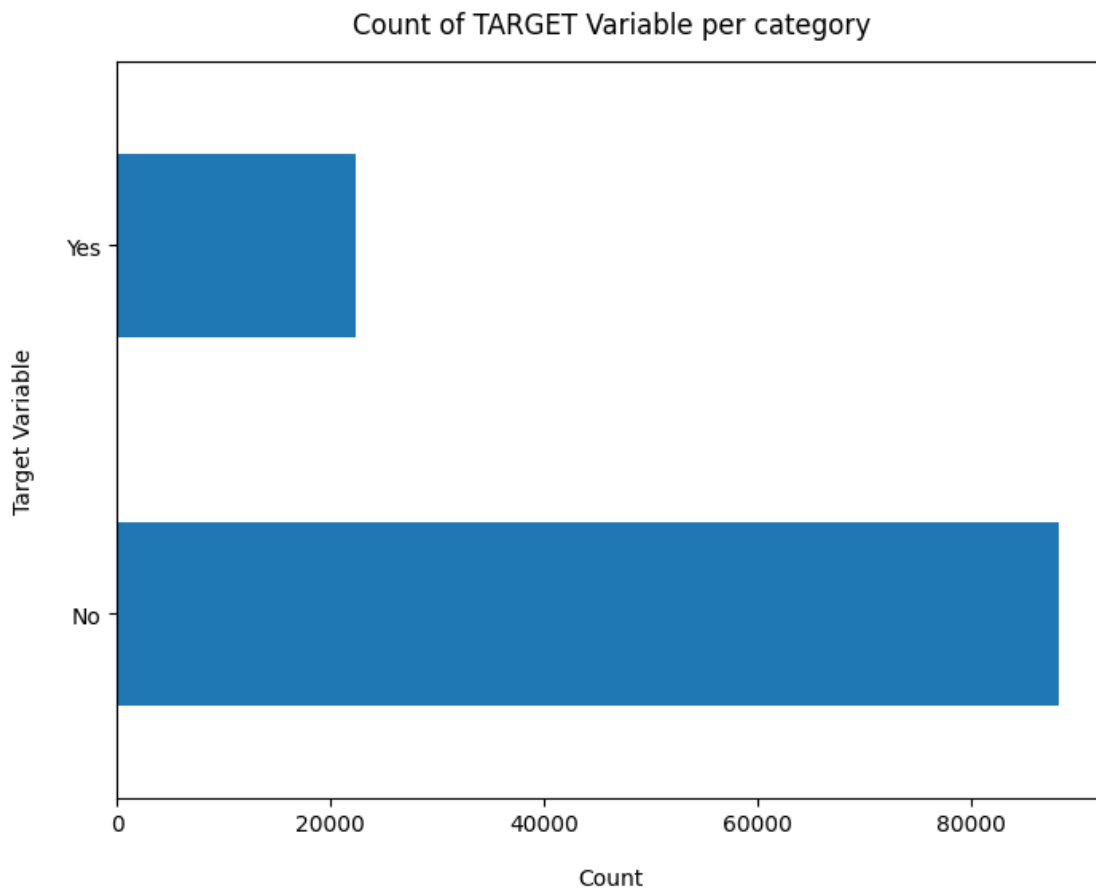
| | Age | Scholarship | Hypertension | Diabetes \ |
|-------|---------------|---------------|---------------|---------------|
| count | 110527.000000 | 110527.000000 | 110527.000000 | 110527.000000 |
| mean | 37.088874 | 0.098266 | 0.197246 | 0.071865 |
| min | -1.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 18.000000 | 0.000000 | 0.000000 | 0.000000 |
| 50% | 37.000000 | 0.000000 | 0.000000 | 0.000000 |
| 75% | 55.000000 | 0.000000 | 0.000000 | 0.000000 |
| max | 115.000000 | 1.000000 | 1.000000 | 1.000000 |
| std | 23.110205 | 0.297675 | 0.397921 | 0.258265 |

| | Alcoholism | Handicap | SMSReceived | sch_weekday \ |
|-------|---------------|---------------|---------------|---------------|
| count | 110527.000000 | 110527.000000 | 110527.000000 | 110527.000000 |
| mean | 0.030400 | 0.022248 | 0.321026 | 1.851955 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| 50% | 0.000000 | 0.000000 | 0.000000 | 2.000000 |
| 75% | 0.000000 | 0.000000 | 1.000000 | 3.000000 |
| max | 1.000000 | 4.000000 | 1.000000 | 5.000000 |
| std | 0.171686 | 0.161543 | 0.466873 | 1.378520 |

| | app_weekday |
|-------|---------------|
| count | 110527.000000 |
| mean | 1.858243 |
| min | 0.000000 |
| 25% | 1.000000 |
| 50% | 2.000000 |
| 75% | 3.000000 |
| max | 5.000000 |
| std | 1.371672 |

```
[21]: base_data['NoShow'].value_counts().plot(kind='barh', figsize=(8, 6))
plt.xlabel("Count", labelpad=14)
plt.ylabel("Target Variable", labelpad=14)
plt.title("Count of TARGET Variable per category", y=1.02)
```

```
[21]: Text(0.5, 1.02, 'Count of TARGET Variable per category')
```



```
[22]: # calculating the % of appointments or not
100*base_data['NoShow'].value_counts()/len(base_data['NoShow'])
```

```
[22]: NoShow
No      79.806744
Yes     20.193256
Name: count, dtype: float64
```

```
[23]: base_data['NoShow'].value_counts()
```

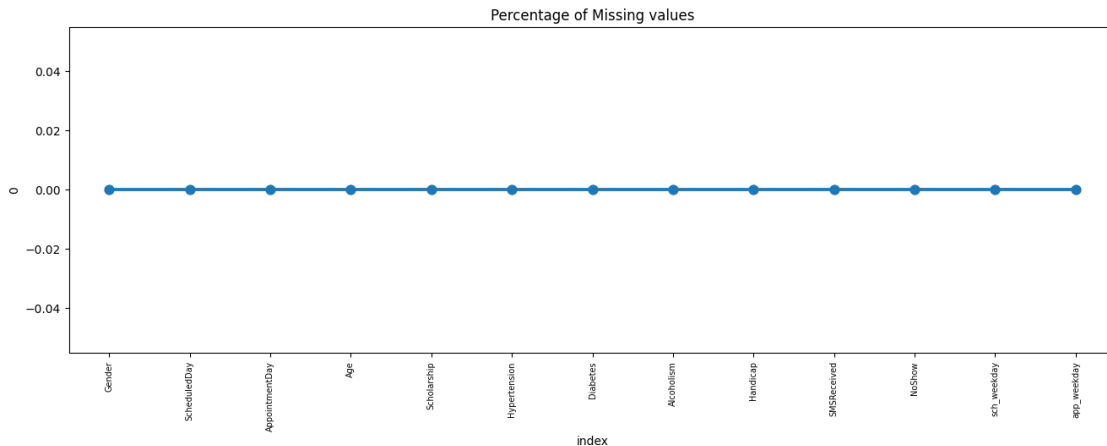
```
[23]: NoShow
No      88208
Yes     22319
Name: count, dtype: int64
```

```
[26]: # Having a look that data contains missing values or not

missing = pd.DataFrame((base_data.isnull().sum())*100/base_data.shape[0]).
    ↪reset_index()
```



```
plt.figure(figsize=(16,5))
ax = sns.pointplot(x='index', y=0, data=missing)
plt.xticks(rotation=90, fontsize=7)
plt.title("Percentage of Missing values")
plt.show()
```



0.0.1 Missing Data - Initial Intuition

- Here, we don't have any missing data.

General Thumb Rules:

- For features with less missing values- can use regression to predict the missing values or fill with the mean of the values present, depending on the feature.
- For features with very high number of missing values- it is better to drop those columns as they give very less insight on analysis.
- As there's no thumb rule on what criteria do we delete the columns with high number of missing values, but generally you can delete the columns, if you have more than 30-40% of missing values.

0.1 Data Cleaning

1. Create a copy of base data for manipulation & processing

```
[27]: new_data = base_data.copy()
```

```
[28]: new_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---

```

```

0   Gender          110527 non-null object
1   ScheduledDay    110527 non-null datetime64[ns]
2   AppointmentDay  110527 non-null datetime64[ns]
3   Age             110527 non-null int64
4   Scholarship     110527 non-null int64
5   Hypertension    110527 non-null int64
6   Diabetes        110527 non-null int64
7   Alcoholism      110527 non-null int64
8   Handicap        110527 non-null int64
9   SMSReceived     110527 non-null int64
10  NoShow          110527 non-null object
11  sch_weekday     110527 non-null int32
12  app_weekday     110527 non-null int32
dtypes: datetime64[ns](2), int32(2), int64(7), object(2)
memory usage: 10.1+ MB

```

As we don't have any null records, there's no data cleaning required

```
[29]: # Get the max tenure
print(base_data['Age'].max()) #72
```

115

```
[30]: # Group the tenure in bins of 12 months
labels = ["{0} - {1}".format(i, i + 20) for i in range(1, 118, 20)]

base_data['Age_group'] = pd.cut(base_data.Age, range(1, 130, 20), right=False,
↳ labels=labels)
```

```
[31]: base_data.drop(['Age'], axis=1, inplace=True)
```

0.2 Data Exploration

```
[32]: list(base_data.columns)
```

```
[32]: ['Gender',
      'ScheduledDay',
      'AppointmentDay',
      'Scholarship',
      'Hypertension',
      'Diabetes',
      'Alcoholism',
      'Handicap',
      'SMSReceived',
      'NoShow',
      'sch_weekday',
      'app_weekday',
      'Age_group']
```

[33]: *#having a loook into the values of count of each columns and there count in*
↪respect to NoShow column

```
for i, predictor in enumerate(base_data.drop(columns=['NoShow'])):
    print('-'*10,predictor,'-'*10)
    print(base_data[predictor].value_counts())
    plt.figure(i)
    sns.countplot(data=base_data, x=predictor, hue='NoShow')
```

----- Gender -----

Gender

F 71840

M 38687

Name: count, dtype: int64

----- ScheduledDay -----

ScheduledDay

2016-05-03 4238

2016-05-02 4216

2016-05-16 4120

2016-05-05 4095

2016-05-10 4024

...

2016-01-27 1

2016-01-19 1

2016-06-04 1

2016-01-26 1

2016-03-05 1

Name: count, Length: 111, dtype: int64

----- AppointmentDay -----

AppointmentDay

2016-06-06 4692

2016-05-16 4613

2016-05-09 4520

2016-05-30 4514

2016-06-08 4479

2016-05-11 4474

2016-06-01 4464

2016-06-07 4416

2016-05-12 4394

2016-05-02 4376

2016-05-18 4373

2016-05-17 4372

2016-06-02 4310

2016-05-10 4308

2016-05-31 4279

2016-05-05 4273

2016-05-19 4270

2016-05-03 4256

```

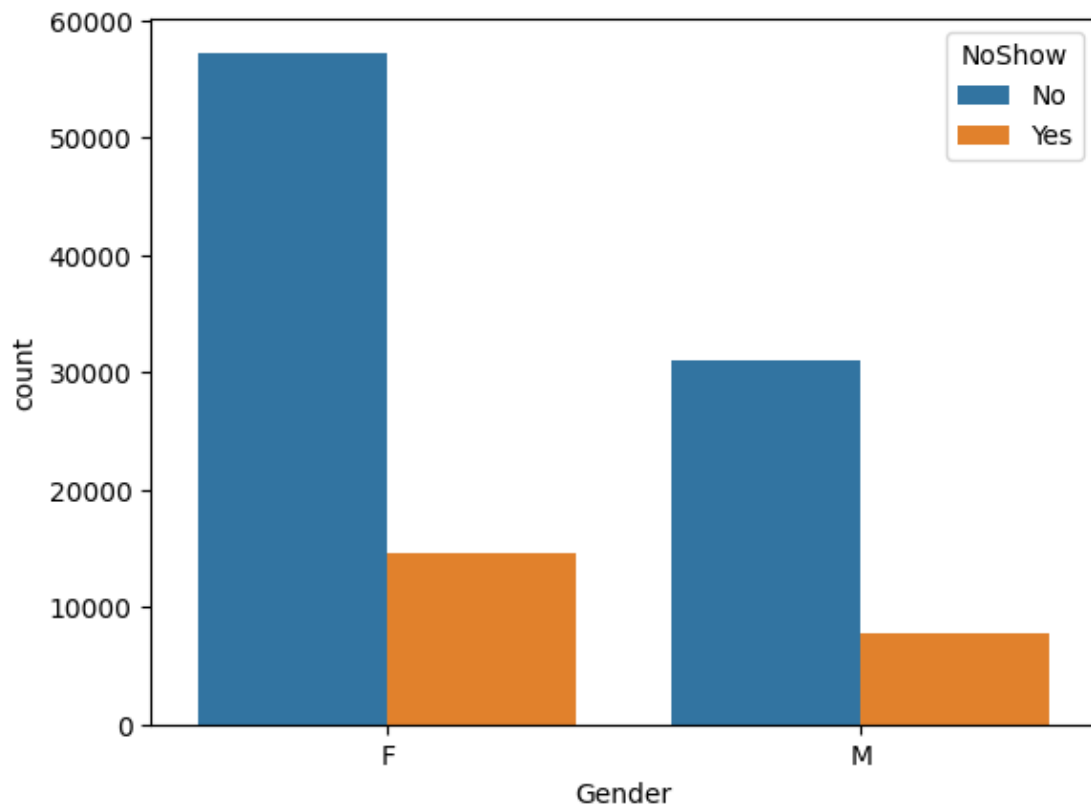
2016-05-04    4168
2016-06-03    4090
2016-05-24    4009
2016-05-13    3987
2016-05-25    3909
2016-05-06    3879
2016-05-20    3828
2016-04-29    3235
2016-05-14     39
Name: count, dtype: int64
----- Scholarship -----
Scholarship
0    99666
1    10861
Name: count, dtype: int64
----- Hypertension -----
Hypertension
0    88726
1    21801
Name: count, dtype: int64
----- Diabetes -----
Diabetes
0    102584
1     7943
Name: count, dtype: int64
----- Alcoholism -----
Alcoholism
0    107167
1     3360
Name: count, dtype: int64
----- Handicap -----
Handicap
0    108286
1     2042
2        183
3         13
4          3
Name: count, dtype: int64
----- SMSReceived -----
SMSReceived
0    75045
1    35482
Name: count, dtype: int64
----- sch_weekday -----
sch_weekday
1    26168
2    24262
0    23085

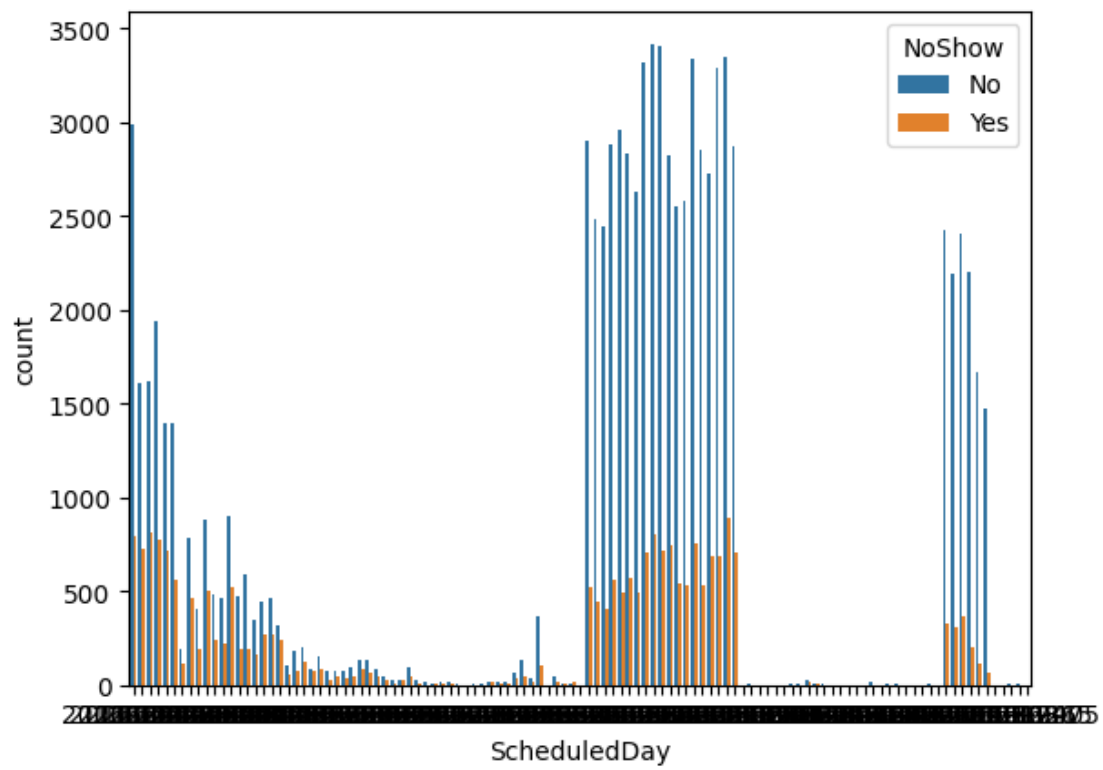
```

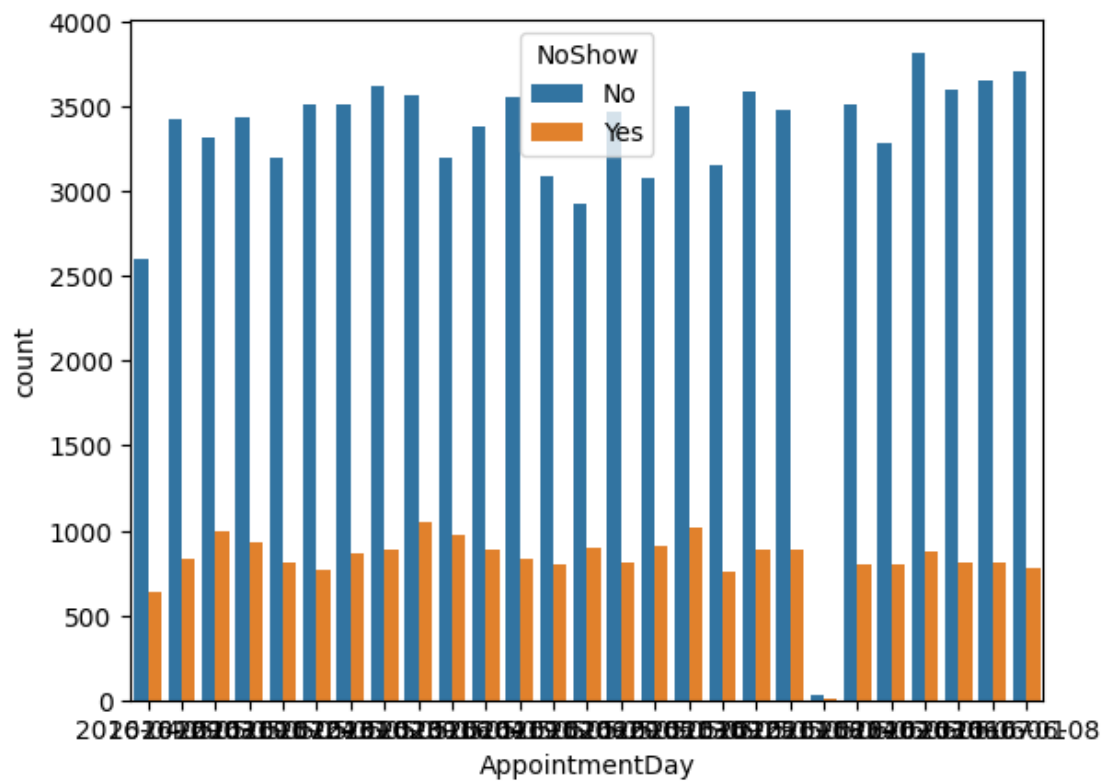
```

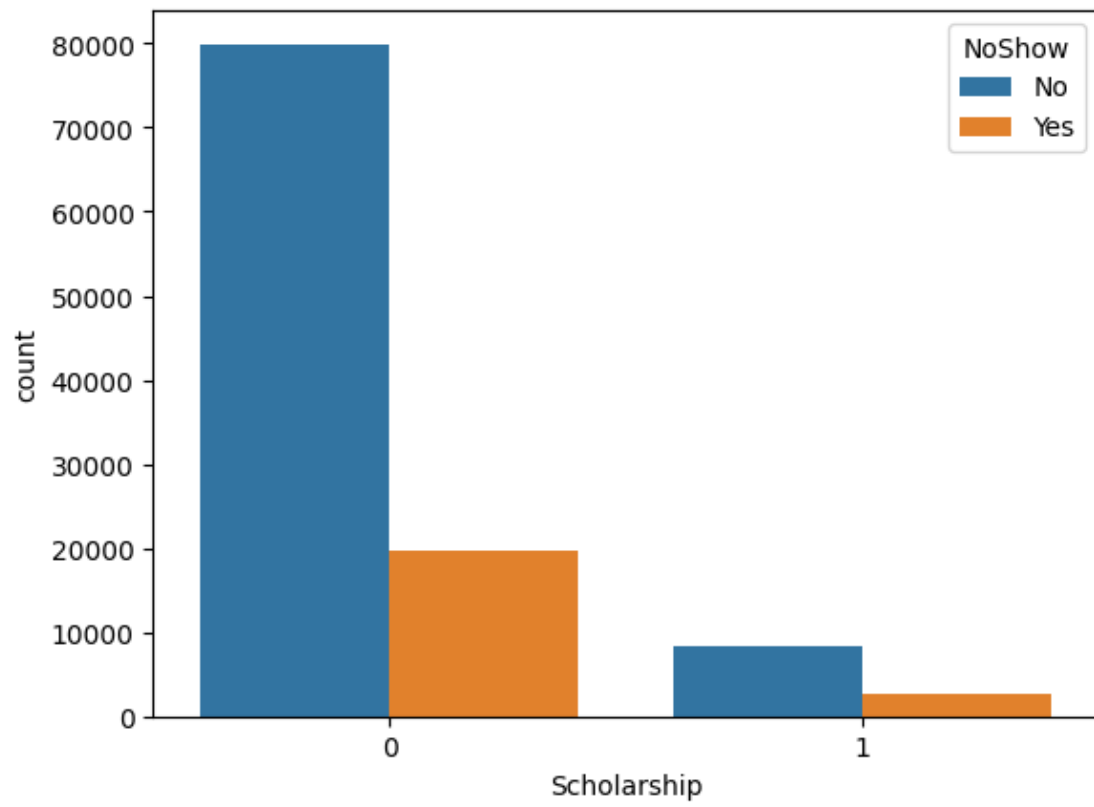
4    18915
3    18073
5      24
Name: count, dtype: int64
----- app_weekday -----
app_weekday
2    25867
1    25640
0    22715
4    19019
3    17247
5       39
Name: count, dtype: int64
----- Age_group -----
Age_group
41 - 61    30081
21 - 41    28835
1 - 21     28309
61 - 81    16910
81 - 101    2845
101 - 121      7
Name: count, dtype: int64

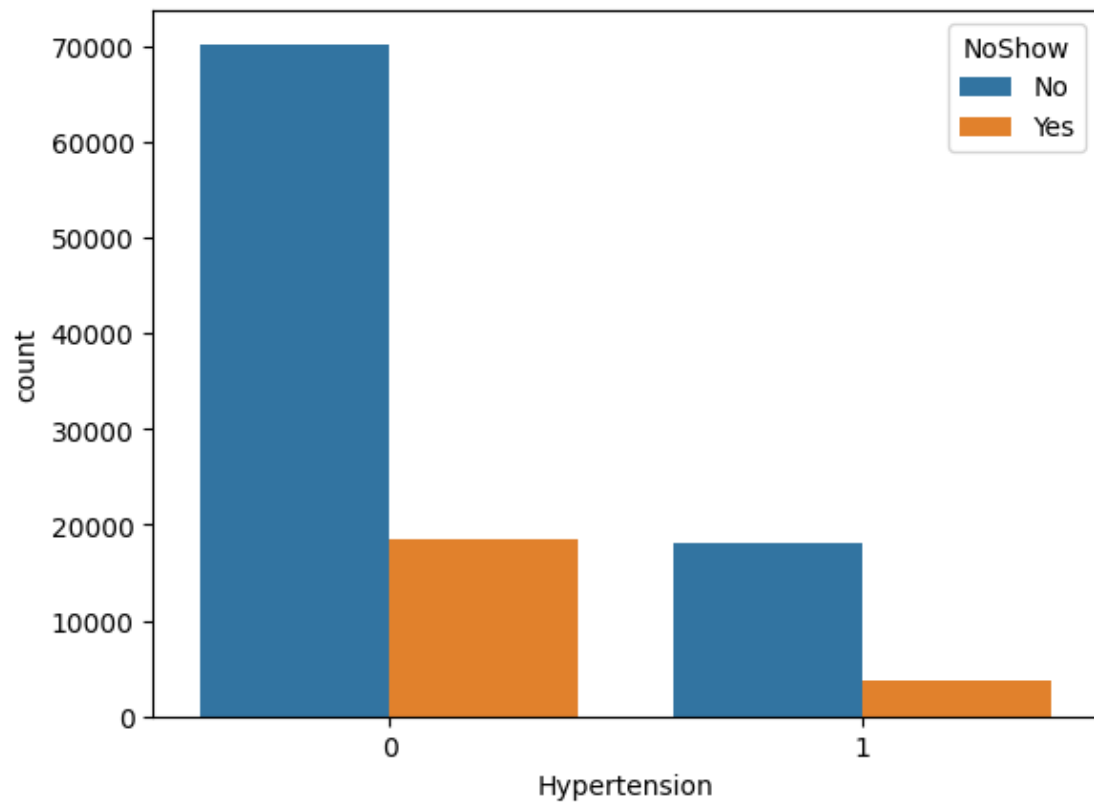
```

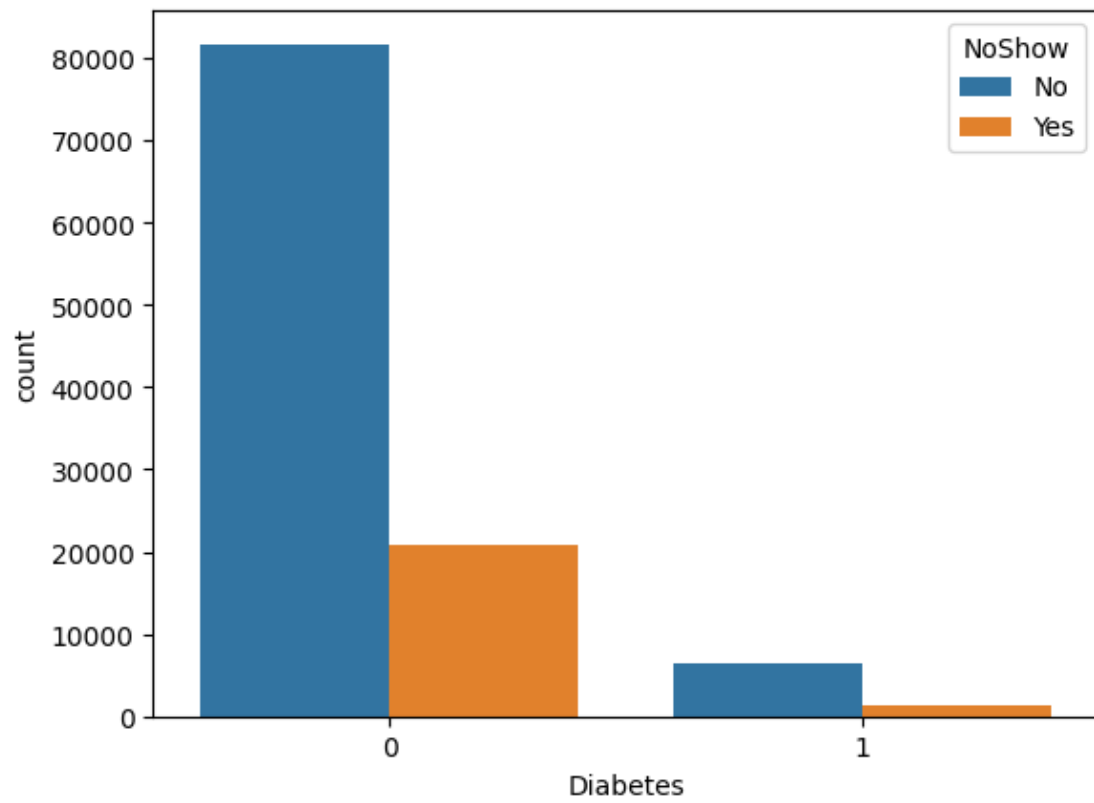


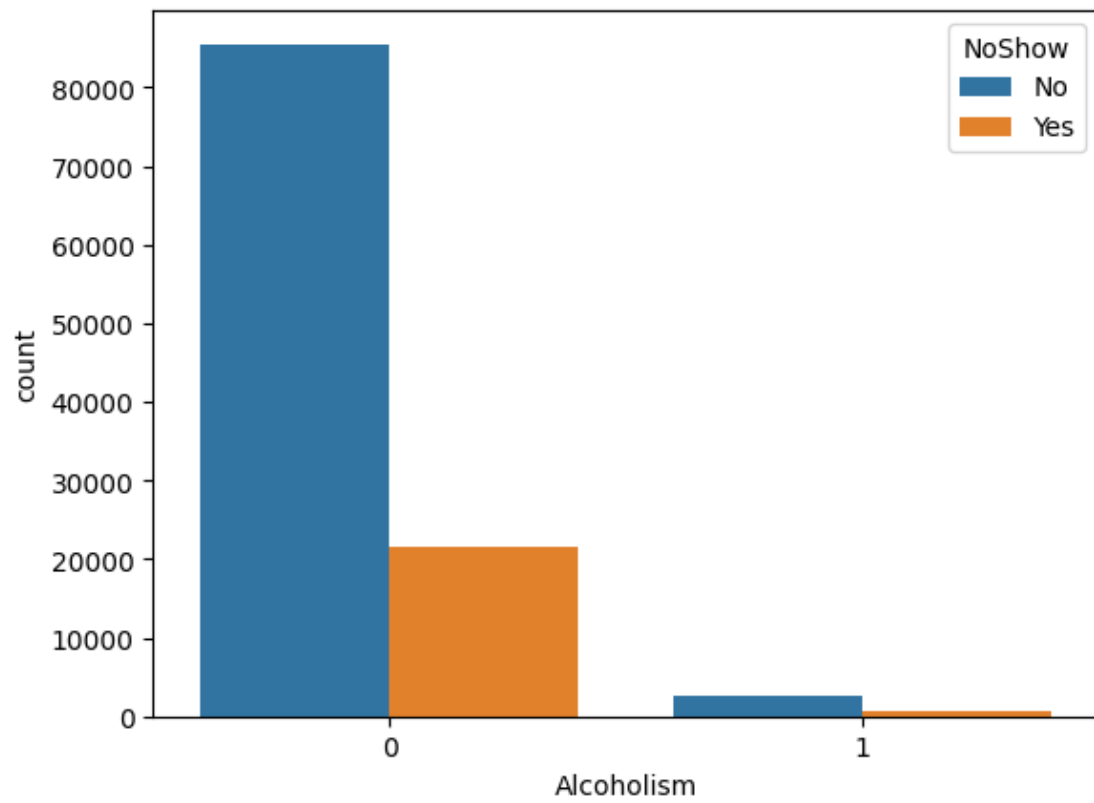


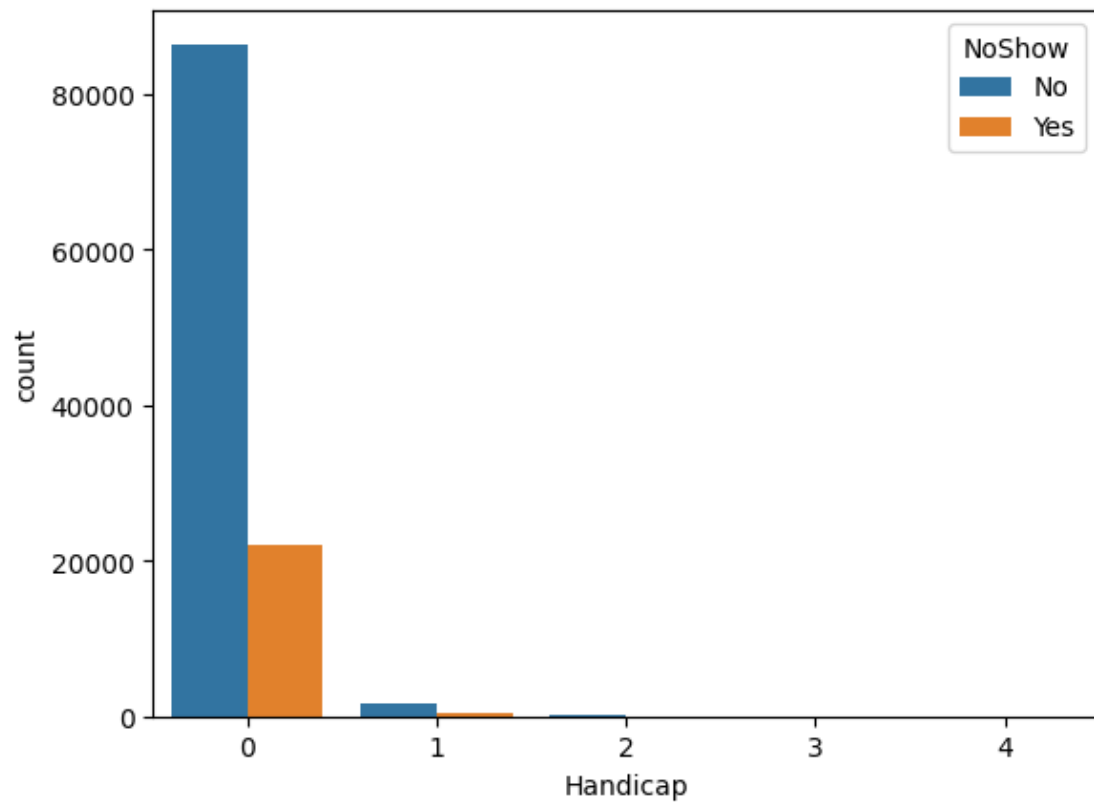


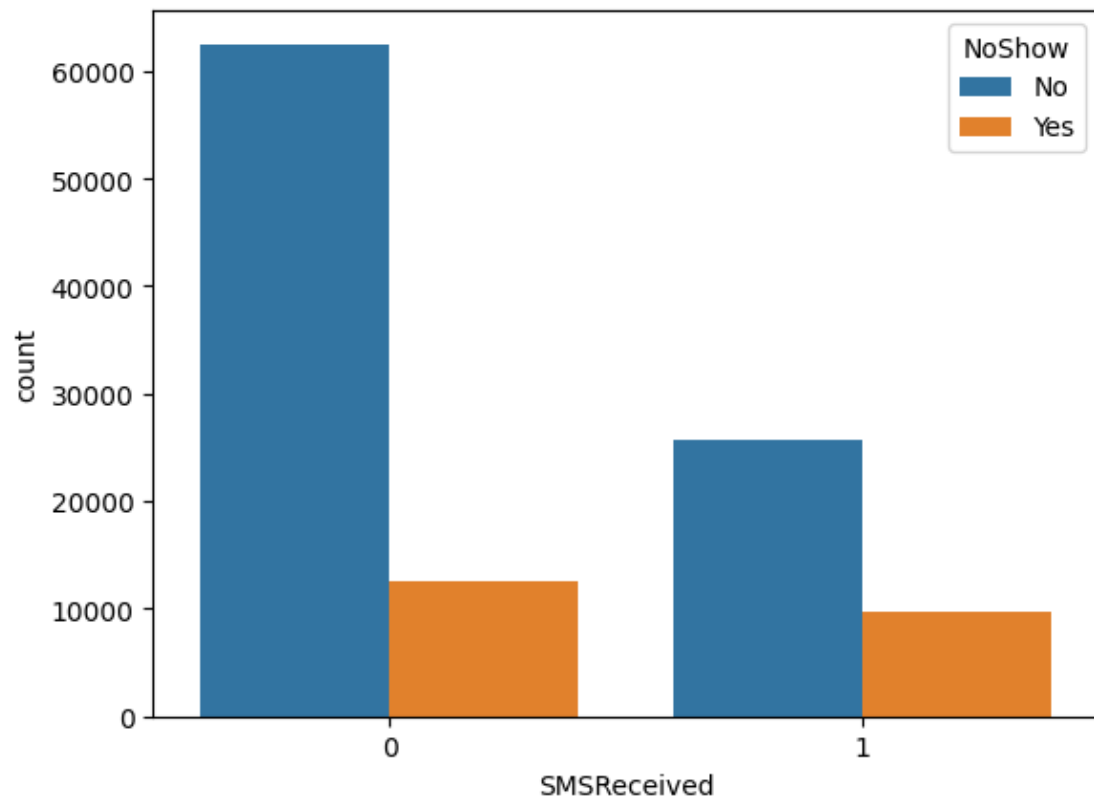


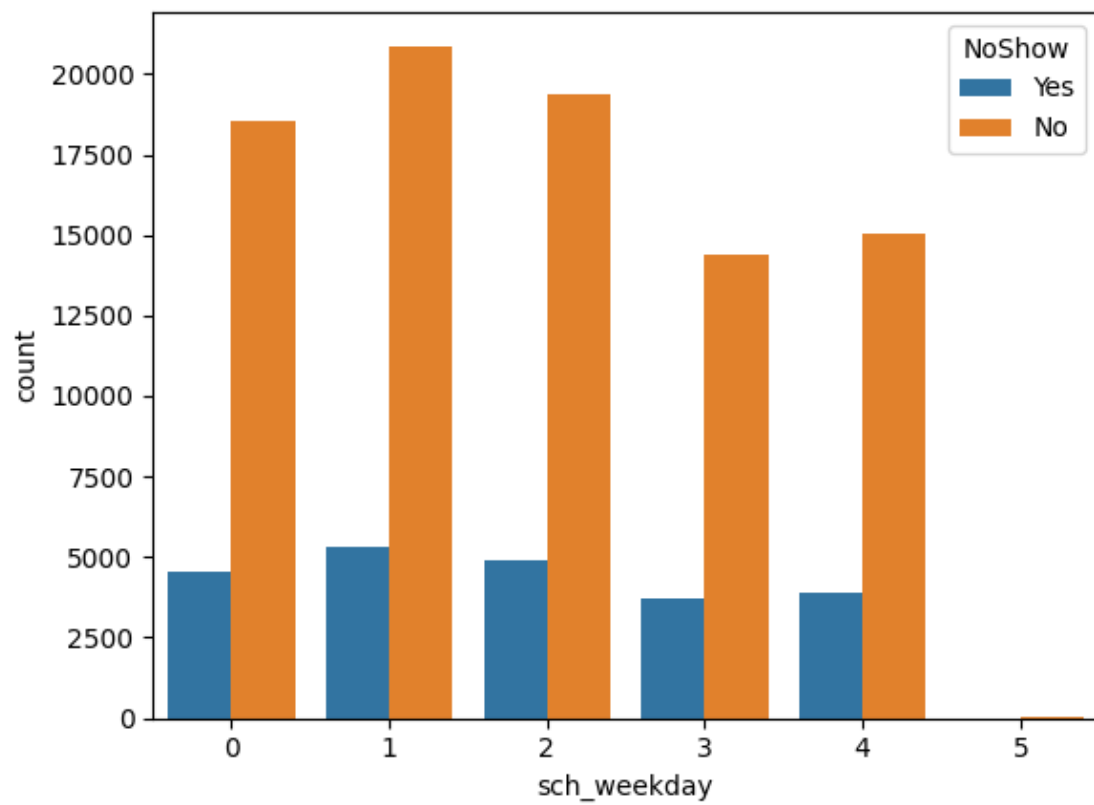


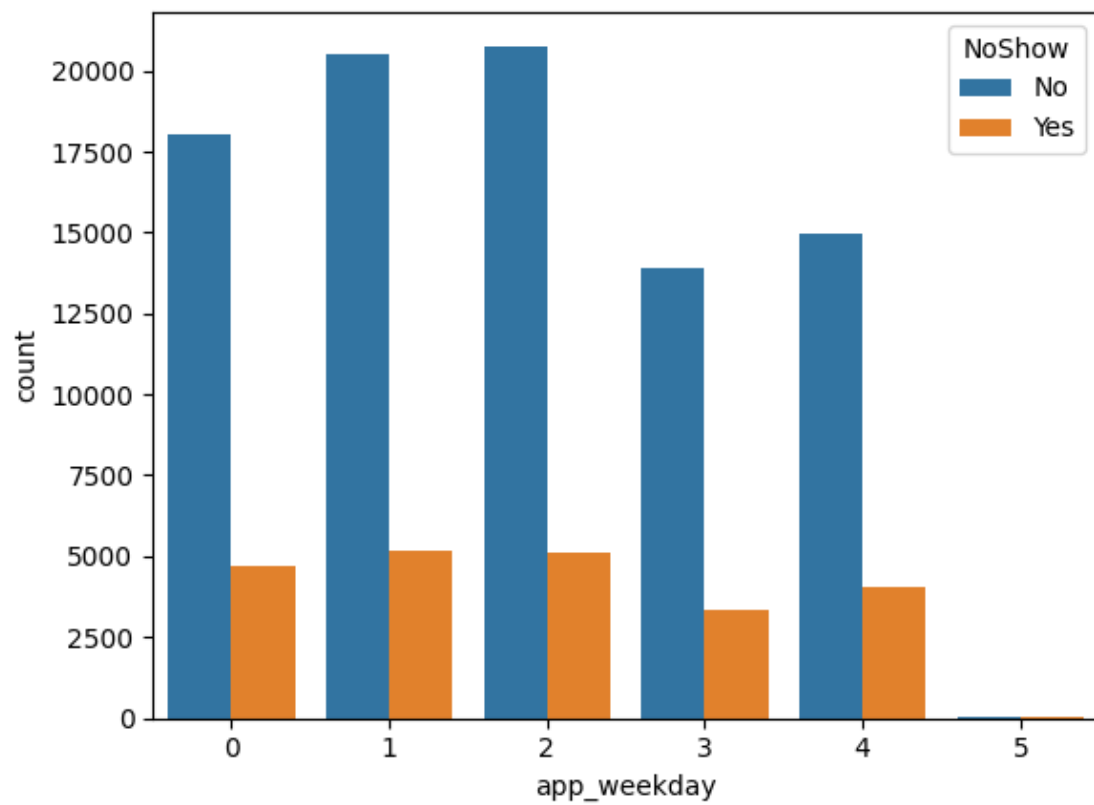


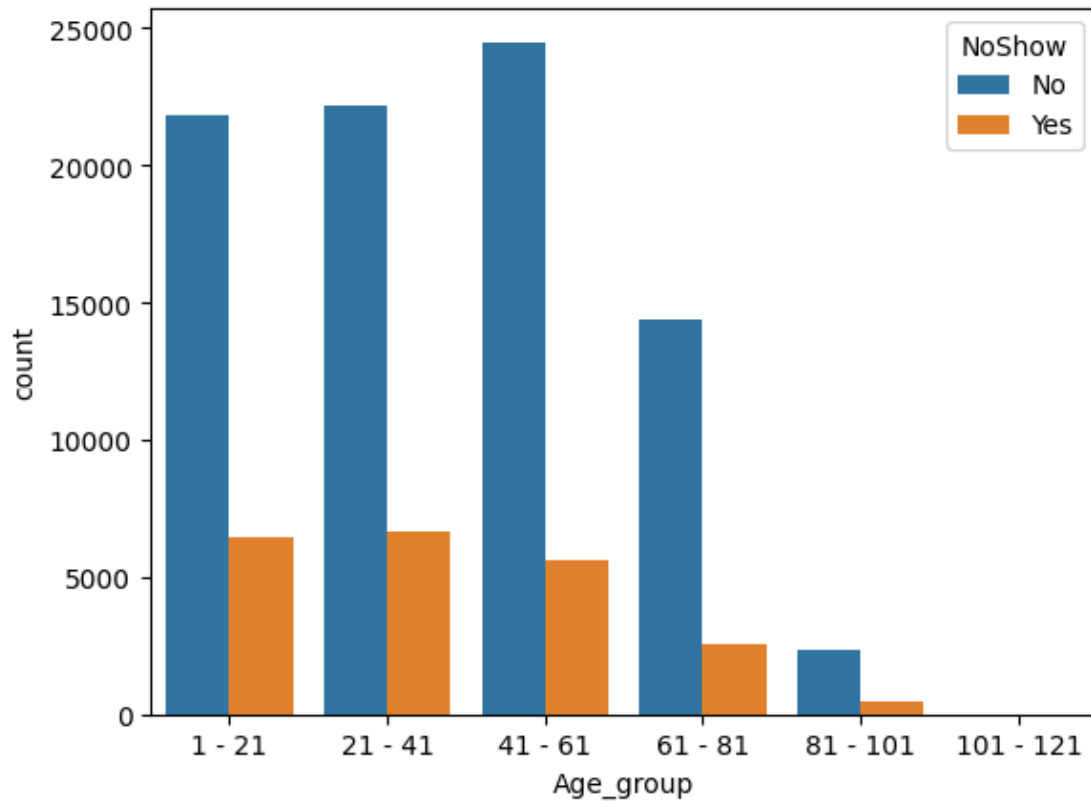












```
[34]: base_data['NoShow'] = np.where(base_data.NoShow == 'Yes',1,0)
```

```
[35]: base_data.NoShow.value_counts()
```

```
[35]: NoShow
0    88208
1     22319
Name: count, dtype: int64
```

Convert all the categorical variables into dummy variables

```
[36]: base_data_dummies = pd.get_dummies(base_data)
base_data_dummies.head()
```

```
[36]: ScheduledDay AppointmentDay Scholarship Hypertension Diabetes \
0    2016-04-29    2016-04-29            0            1            0
1    2016-04-29    2016-04-29            0            0            0
2    2016-04-29    2016-04-29            0            0            0
3    2016-04-29    2016-04-29            0            0            0
4    2016-04-29    2016-04-29            0            1            1
```


| | Alcoholism | Handicap | SMSReceived | NoShow | sch_weekday | app_weekday | \ |
|---|------------|----------|-------------|--------|-------------|-------------|---|
| 0 | 0 | 0 | 0 | 0 | 4 | 4 | |
| 1 | 0 | 0 | 0 | 0 | 4 | 4 | |
| 2 | 0 | 0 | 0 | 0 | 4 | 4 | |
| 3 | 0 | 0 | 0 | 0 | 4 | 4 | |
| 4 | 0 | 0 | 0 | 0 | 4 | 4 | |

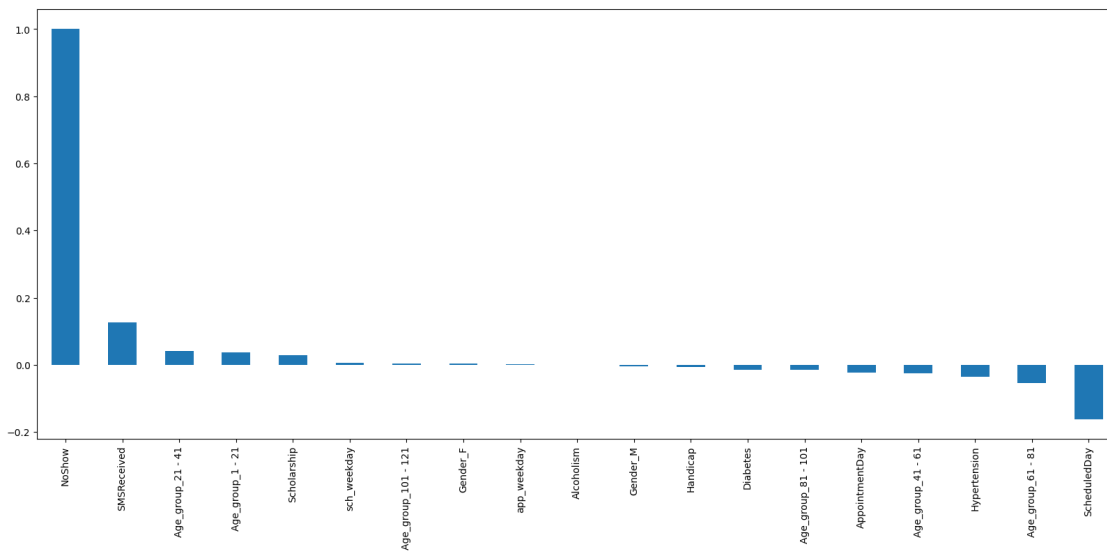
| | Gender_F | Gender_M | Age_group_1 - 21 | Age_group_21 - 41 | Age_group_41 - 61 | \ |
|---|----------|----------|------------------|-------------------|-------------------|-------|
| 0 | True | False | False | False | False | False |
| 1 | False | True | False | False | False | True |
| 2 | True | False | False | False | False | False |
| 3 | True | False | True | False | False | False |
| 4 | True | False | False | False | False | True |

| | Age_group_61 - 81 | Age_group_81 - 101 | Age_group_101 - 121 |
|---|-------------------|--------------------|---------------------|
| 0 | True | False | False |
| 1 | False | False | False |
| 2 | True | False | False |
| 3 | False | False | False |
| 4 | False | False | False |

Build a correlation of all predictors with 'NoShow'

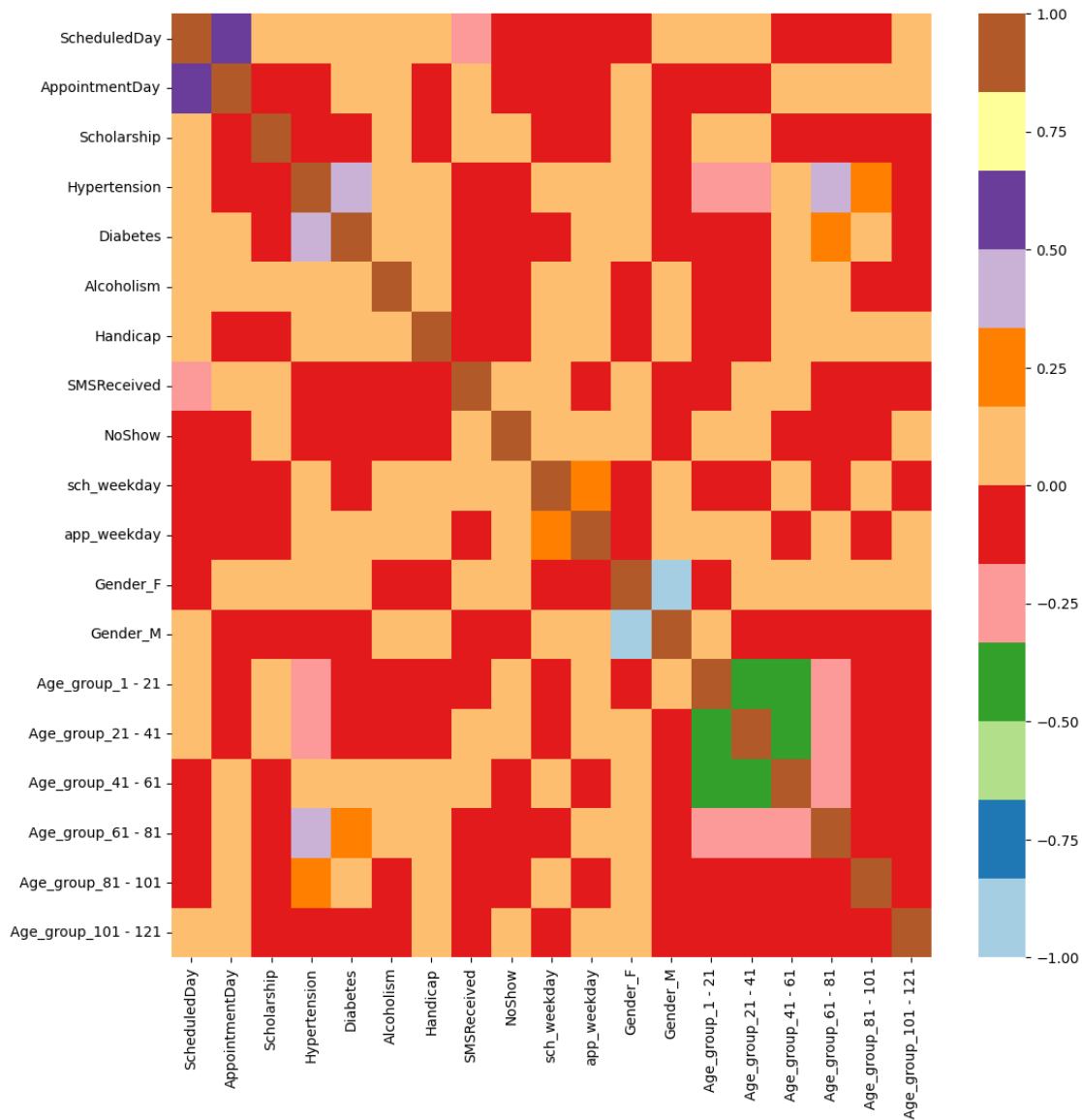
```
[37]: plt.figure(figsize=(20,8))
base_data_dummies.corr()['NoShow'].sort_values(ascending = False).
      plot(kind='bar')
```

[37]: <Axes: >



```
[38]: plt.figure(figsize=(12,12))
sns.heatmap(base_data_dummies.corr(), cmap="Paired")
```

[38]: <Axes: >



0.3 Bivariate Analysis

```
[39]: new_df1_target0=base_data.loc[base_data["NoShow"]==0]
new_df1_target1=base_data.loc[base_data["NoShow"]==1]
```

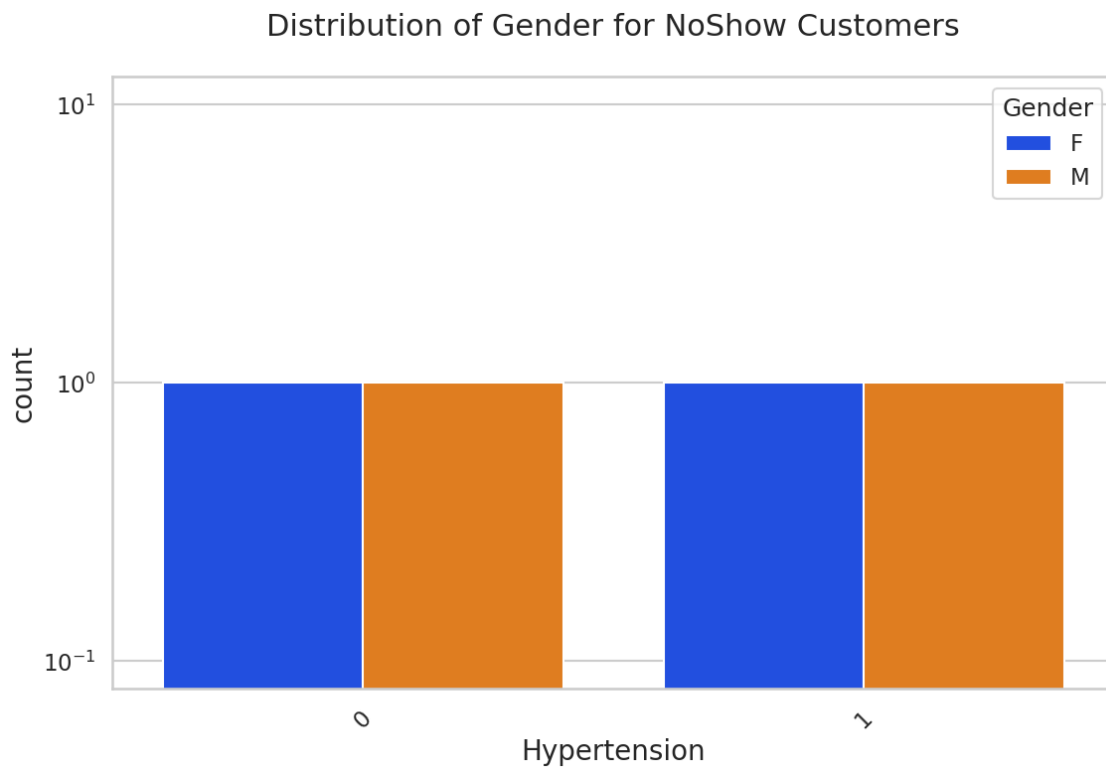
```
[40]: def unipLOT(df,col,title,hue =None):

    sns.set_style('whitegrid')
    sns.set_context('talk')
    plt.rcParams["axes.labelsize"] = 20
    plt.rcParams['axes.titlesize'] = 22
    plt.rcParams['axes.titlepad'] = 30

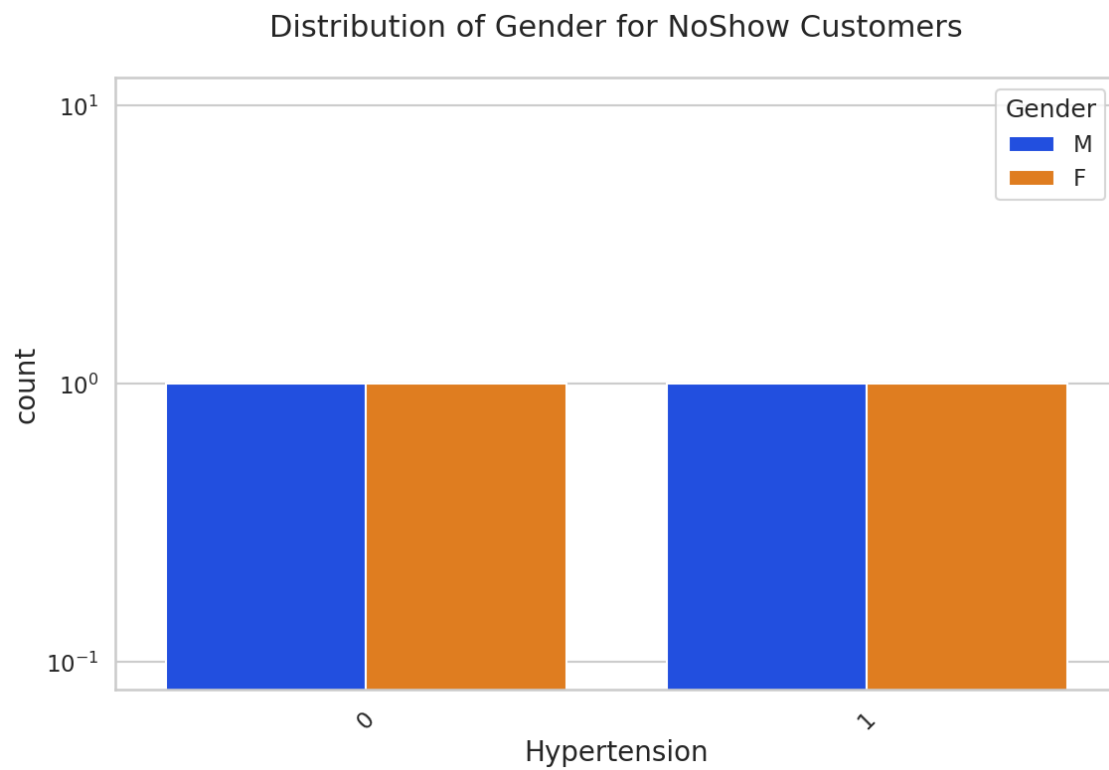
    temp = pd.Series(data = hue)
    fig, ax = plt.subplots()
    width = len(df[col].unique()) + 7 + 4*len(temp.unique())
    fig.set_size_inches(width , 8)
    plt.xticks(rotation=45)
    plt.yscale('log')
    plt.title(title)
    ax = sns.countplot(data = df, x= col, order=df[col].value_counts().
↳index,hue = hue,palette='bright')

    plt.show()
```

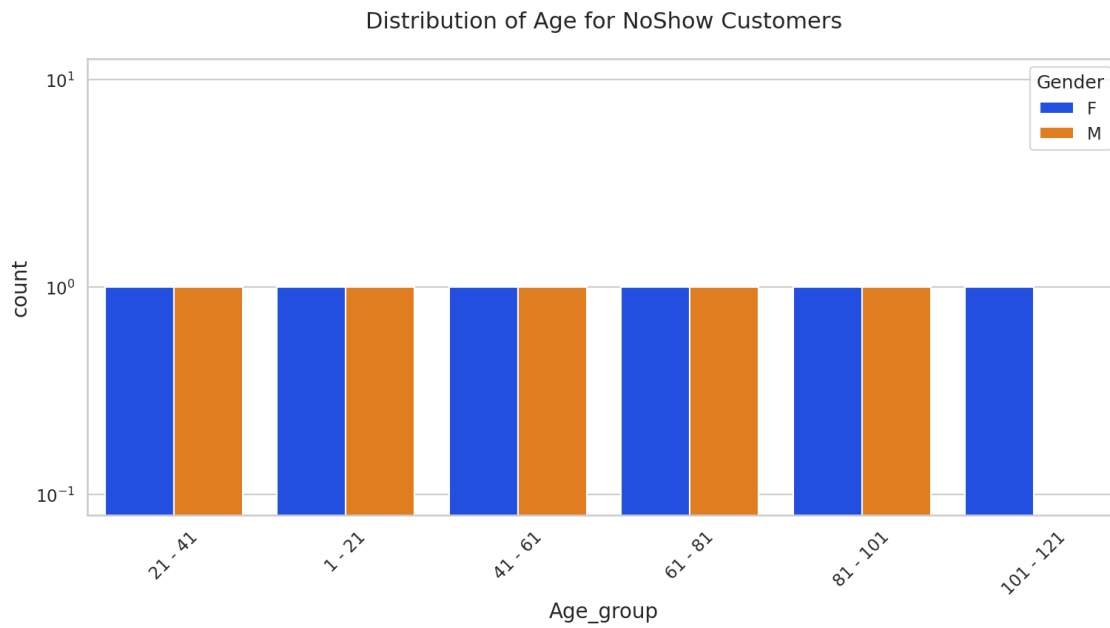
```
[41]: unipLOT(new_df1_target1,col='Hypertension',title='Distribution of Gender for_
↳NoShow Customers',hue='Gender')
```



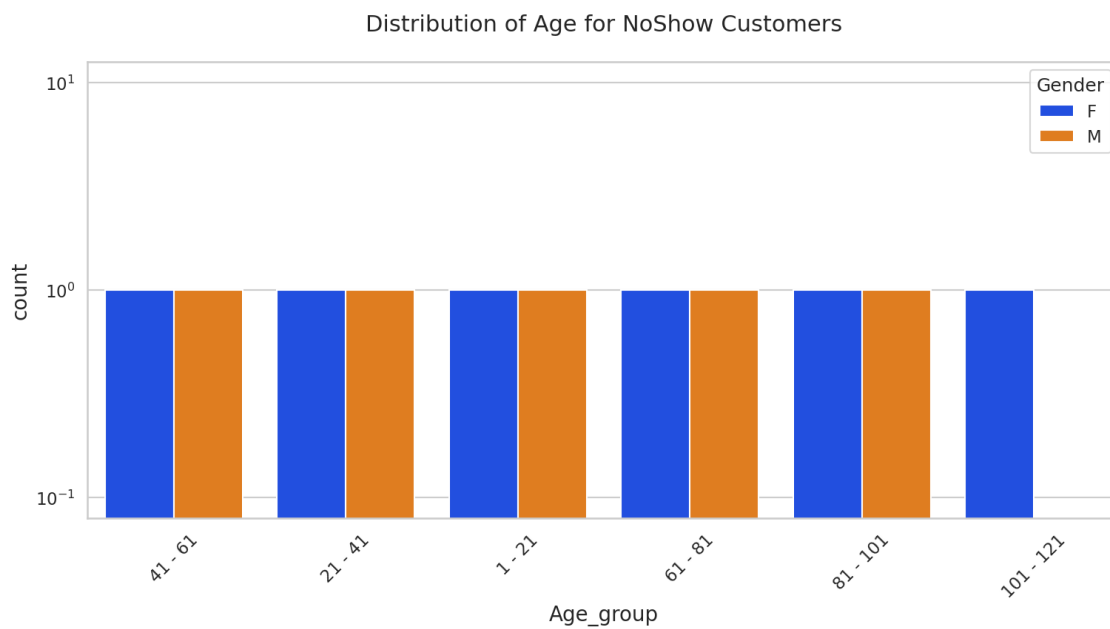
```
[42]: uniplot(new_df1_target0,col='Hypertension',title='Distribution of Gender for_
↳NoShow Customers',hue='Gender')
```



```
[43]: uniplot(new_df1_target1,col='Age_group',title='Distribution of Age for NoShow_
↳Customers',hue='Gender')
```



```
[44]: uniplot(new_df1_target0,col='Age_group',title='Distribution of Age for NoShow_
↳Customers',hue='Gender')
```



```
[ ]:
```

Findings

1. Female patients have taken more appointments then male patients
2. Ratio of Nohow and Show is almost equal for age group except Age 0 and Age 1 with 80% show rate for each age group
3. Each Neighbourhood have almost 80% show rate
4. There are 99666 patients without Scholarship and out of them around 80% have come for the visit and out of the 21801 patients with Scholarship around 75% of them have come for the visit.
5. there are around 88,726 patients without Hypertension and out of them around 78% have come for the visit and Out of the 21801 patients with Hypertension around 85% of them have come for the visit.
6. there are around 102,584 patients without Diabetes and out of them around 80% have come for the visit and Out of the 7,943 patients with Diabetes around 83% of them have come for the visit.
7. there are around 75,045 patients who have not received SMS and out of them around 84% have come for the visit and out of the 35,482 patients who have received SMS around 72% of them have come for the visit.
8. there is no appointments on sunday and on saturday appointments are very less in comparision to other week days