Submit_final_noShowAppointments

May 24, 2019

Investigation of TMDb movie dataset

0.1 Table of Contents

Introduction

> **Tip**: In this section of the report, provide a brief introduction to the dataset you've selected for analysis. At the end of this section, describe the questions that you plan on exploring over the course of the report. Try to build your report around the analysis of at least one dependent variable and three independent variables.
>
> If you haven't yet selected and downloaded your data, make sure you do that first before coming back here. If you're not sure what questions to ask right now, then make sure you familiarize yourself with the variables and the dataset context for ideas of what to explore.

Home

0.1.1 Context

A person makes a doctor appointment, receives all the instructions and no-show. Who to blame? If this is help, dont't forget to upvote :) Greatings!

0.1.2 Content

300k medical appointments and its 15 variables (characteristics) of each. The most important one if the patient show-up or no-show the appointment. Variable names are self-explanatory, if you have doubts, just let me know!

scholarship variable means this concept = https://en.wikipedia.org/wiki/Bolsa_Fam%C3%ADlia

0.1.3 Data Dictionary

PatientId - Identification of a patient AppointmentID - Identification of each appointment Gender = Male or Female . Female is the greater proportion, woman takes way more care of they health in comparison to man. DataMarcacaoConsulta = The day of the actuall appointment, when they have to visit the doctor. DataAgendamento = The day someone called or registered the appointment, this is before appointment of course. Age = How old is the patient. Neighbourhood = Where the appointment takes place. Scholarship = Ture of False . Observation, this is a broad topic, consider reading this article https://en.wikipedia.org/wiki/Bolsa_Fam%C3%ADlia Hipertension = True or False Diabetes = True or False Alcoholism = True or False Handcap = True or False SMS_received = 1 or more messages sent to the patient. No-show = True or False.

0.1.4 Inspiration

What if that possible to predict someone to no-show an appointment?

Questions - - Ratio of people missing appointments - Absence of people based on gender - Appointments based on hour of the day - Appointments based on the days of the week - Appointments based on month - Most important factor which leads to their absence

Home

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.

Home ### General Properties

[426]:
```
# Import all necessary libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

%matplotlib inline
```

[427]:
```
# Load data
df = pd.read_csv('data/noshowappointments.csv', parse_dates=True)
```

[428]:
```
df.head()
```

[428]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	0	1
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	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
```

[429]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
PatientId         110527 non-null float64
AppointmentID     110527 non-null int64
Gender            110527 non-null object
ScheduledDay      110527 non-null object
AppointmentDay    110527 non-null object
Age               110527 non-null int64
Neighbourhood     110527 non-null object
Scholarship       110527 non-null int64
Hipertension      110527 non-null int64
Diabetes          110527 non-null int64
Alcoholism        110527 non-null int64
Handcap           110527 non-null int64
SMS_received      110527 non-null int64
No-show           110527 non-null object
dtypes: float64(1), int64(8), object(5)
memory usage: 11.8+ MB
```

[430]:
```
df.shape
```

[430]:
```
(110527, 14)
```

[431]:
```
df.columns
```

[431]:
```
Index(['PatientId', 'AppointmentID', 'Gender', 'ScheduledDay',
       'AppointmentDay', 'Age', 'Neighbourhood', 'Scholarship', 'Hipertension',
       'Diabetes', 'Alcoholism', 'Handcap', 'SMS_received', 'No-show'],
      dtype='object')
```

0.2 Todo -

- Modify data types : PatientId, AppointmentId
- Format Date : ScheduledDay, AppointmentDay
- Get dummies for No-show and Gender

- Reorder Columns

[432]:
```
# Modify Data types

df['PatientId'] = df['PatientId'].astype('str')
df['AppointmentID'] = df['AppointmentID'].astype('str')
df['PatientId'] = df['PatientId'].str.split('.', expand=True)[0]
```

[433]:
```
# Format Date

df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay'], format="%Y-%m-%dT%H:%M:
 ↪%SZ")
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay'], format="%Y-%m-%dT%H:
 ↪%M:%SZ")
```

[434]:
```
# Get dummies for No-show and Gender columns

df[['Present','Absent']] = pd.get_dummies(df['No-show'])
# df[['Female','Male']] = pd.get_dummies(df['Gender'])
```

[435]:
```
# Drop extra columns

df.drop(['Absent','No-show'], inplace=True, axis=1)
# df.drop(['Gender','Female'], inplace=True, axis=1)
```

[437]:
```
# No of Duplicated values

for _ in df.columns:
    print(_,sum(df[_].duplicated()))
```

```
PatientId 48228
AppointmentID 0
Gender 110525
ScheduledDay 6978
AppointmentDay 110500
Age 110423
Neighbourhood 110446
Scholarship 110525
Hipertension 110525
Diabetes 110525
Alcoholism 110525
Handcap 110522
SMS_received 110525
Present 110525
```

[438]:
```
# No of Unique values

for _ in df.columns:
    print(_,len(df[_].unique()))
```

```
    print((df[_].unique()),'\n')
```

```
PatientId 62299
['29872499824296' '558997776694438' '4262962299951' ... '72633149253362'
 '996997666245785' '15576631729893']

AppointmentID 110527
['5642903' '5642503' '5642549' ... '5630692' '5630323' '5629448']

Gender 2
['F' 'M']

ScheduledDay 103549
['2016-04-29T18:38:08.000000000' '2016-04-29T16:08:27.000000000'
 '2016-04-29T16:19:04.000000000' ... '2016-04-27T16:03:52.000000000'
 '2016-04-27T15:09:23.000000000' '2016-04-27T13:30:56.000000000']

AppointmentDay 27
['2016-04-29T00:00:00.000000000' '2016-05-03T00:00:00.000000000'
 '2016-05-10T00:00:00.000000000' '2016-05-17T00:00:00.000000000'
 '2016-05-24T00:00:00.000000000' '2016-05-31T00:00:00.000000000'
 '2016-05-02T00:00:00.000000000' '2016-05-30T00:00:00.000000000'
 '2016-05-16T00:00:00.000000000' '2016-05-04T00:00:00.000000000'
 '2016-05-19T00:00:00.000000000' '2016-05-12T00:00:00.000000000'
 '2016-05-06T00:00:00.000000000' '2016-05-20T00:00:00.000000000'
 '2016-05-05T00:00:00.000000000' '2016-05-13T00:00:00.000000000'
 '2016-05-09T00:00:00.000000000' '2016-05-25T00:00:00.000000000'
 '2016-05-11T00:00:00.000000000' '2016-05-18T00:00:00.000000000'
 '2016-05-14T00:00:00.000000000' '2016-06-02T00:00:00.000000000'
 '2016-06-03T00:00:00.000000000' '2016-06-06T00:00:00.000000000'
 '2016-06-07T00:00:00.000000000' '2016-06-01T00:00:00.000000000'
 '2016-06-08T00:00:00.000000000']

Age 104
[ 62  56   8  76  23  39  21  19  30  29  22  28  54  15  50  40  46   4
  13  65  45  51  32  12  61  38  79  18  63  64  85  59  55  71  49  78
  31  58  27   6   2  11   7   0   3   1  69  68  60  67  36  10  35  20
  26  34  33  16  42   5  47  17  41  44  37  24  66  77  81  70  53  75
  73  52  74  43  89  57  14   9  48  83  72  25  80  87  88  84  82  90
  94  86  91  98  92  96  93  95  97 102 115 100  99  -1]

Neighbourhood 81
['JARDIM DA PENHA' 'MATA DA PRAIA' 'PONTAL DE CAMBURI' 'REPÚBLICA'
 'GOIABEIRAS' 'ANDORINHAS' 'CONQUISTA' 'NOVA PALESTINA' 'DA PENHA'
 'TABUAZEIRO' 'BENTO FERREIRA' 'SÃO PEDRO' 'SANTA MARTHA' 'SÃO CRISTÓVÃO'
 'MARUÍPE' 'GRANDE VITÓRIA' 'SÃO BENEDITO' 'ILHA DAS CAIEIRAS'
 'SANTO ANDRÉ' 'SOLON BORGES' 'BONFIM' 'JARDIM CAMBURI' 'MARIA ORTIZ'
```

```
 'JABOUR' 'ANTÔNIO HONÓRIO' 'RESISTÊNCIA' 'ILHA DE SANTA MARIA'
 'JUCUTUQUARA' 'MONTE BELO' 'MÁRIO CYPRESTE' 'SANTO ANTÔNIO' 'BELA VISTA'
 'PRAIA DO SUÁ' 'SANTA HELENA' 'ITARARÉ' 'INHANGUETÁ' 'UNIVERSITÁRIO'
 'SÃO JOSÉ' 'REDENÇÃO' 'SANTA CLARA' 'CENTRO' 'PARQUE MOSCOSO'
 'DO MOSCOSO' 'SANTOS DUMONT' 'CARATOÍRA' 'ARIOVALDO FAVALESSA'
 'ILHA DO FRADE' 'GURIGICA' 'JOANA D‡ARC' 'CONSOLAÇÃO' 'PRAIA DO CANTO'
 'BOA VISTA' 'MORADA DE CAMBURI' 'SANTA LUÍZA' 'SANTA LÚCIA'
 'BARRO VERMELHO' 'ESTRELINHA' 'FORTE SÃO JOÃO' 'FONTE GRANDE'
 'ENSEADA DO SUÁ' 'SANTOS REIS' 'PIEDADE' 'JESUS DE NAZARETH'
 'SANTA TEREZA' 'CRUZAMENTO' 'ILHA DO PRÍNCIPE' 'ROMÃO' 'COMDUSA'
 'SANTA CECÍLIA' 'VILA RUBIM' 'DE LOURDES' 'DO QUADRO' 'DO CABRAL' 'HORTO'
 'SEGURANÇA DO LAR' 'ILHA DO BOI' 'FRADINHOS' 'NAZARETH' 'AEROPORTO'
 'ILHAS OCEÂNICAS DE TRINDADE' 'PARQUE INDUSTRIAL']

Scholarship 2
[0 1]

Hipertension 2
[1 0]

Diabetes 2
[0 1]

Alcoholism 2
[0 1]

Handcap 5
[0 1 2 3 4]

SMS_received 2
[0 1]

Present 2
[1 0]
```

0.2.1 Check if all the todo's are done or not

[439]:
```
df.head()
```

[439]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism
0	29872499824296	5642903	F	2016-04-29 18:38:08	2016-04-29						
1	558997776694438	5642503	M	2016-04-29 16:08:27	2016-04-29						
2	4262962299951	5642549	F	2016-04-29 16:19:04	2016-04-29						
3	867951213174	5642828	F	2016-04-29 17:29:31	2016-04-29						
4	8841186448183	5642494	F	2016-04-29 16:07:23	2016-04-29						

```
0   62    JARDIM DA PENHA            0             1         0           0
1   56    JARDIM DA PENHA            0             0         0           0
2   62      MATA DA PRAIA            0             0         0           0
3    8  PONTAL DE CAMBURI            0             0         0           0
4   56    JARDIM DA PENHA            0             1         1           0

   Handcap  SMS_received  Present
0        0             0        1
1        0             0        1
2        0             0        1
3        0             0        1
4        0             0        1
```

[440]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
PatientId         110527 non-null object
AppointmentID     110527 non-null object
Gender            110527 non-null object
ScheduledDay      110527 non-null datetime64[ns]
AppointmentDay    110527 non-null datetime64[ns]
Age               110527 non-null int64
Neighbourhood     110527 non-null object
Scholarship       110527 non-null int64
Hipertension      110527 non-null int64
Diabetes          110527 non-null int64
Alcoholism        110527 non-null int64
Handcap           110527 non-null int64
SMS_received      110527 non-null int64
Present           110527 non-null uint8
dtypes: datetime64[ns](2), int64(7), object(4), uint8(1)
memory usage: 11.1+ MB
```

> **Tip**: You should *not* perform too many operations in each cell. Create cells freely to explore your data. One option that you can take with this project is to do a lot of explorations in an initial notebook. These don't have to be organized, but make sure you use enough comments to understand the purpose of each code cell. Then, after you're done with your analysis, create a duplicate notebook where you will trim the excess and organize your steps so that you have a flowing, cohesive report.

> **Tip**: Make sure that you keep your reader informed on the steps that you are taking in your investigation. Follow every code cell, or every set of related code cells, with a markdown cell to describe to the reader what was found in the preceding cell(s). Try to make it so that the reader can then understand what they will be seeing in the following cell(s).

0.2.2 Data Cleaning (Replace this with more specific notes!)

[454]:
```
df.describe()
```

[454]:
```
                Age    Scholarship   Hipertension       Diabetes  \
count  110526.000000  110526.000000  110526.000000  110526.000000
mean       37.089219       0.098266       0.197248       0.071865
std        23.110026       0.297676       0.397923       0.258266
min         0.000000       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

          Alcoholism        Handcap   SMS_received                WaitingTime  \
count  110526.000000  110526.000000  110526.000000                     110526
mean        0.030400       0.022248       0.321029  -10 days +06:51:17.952047
std         0.171686       0.161543       0.466874    15 days 05:51:31.240428
min         0.000000       0.000000       0.000000        -179 days +10:40:59
25%         0.000000       0.000000       0.000000  -15 days +16:18:22.250000
50%         0.000000       0.000000       0.000000          -4 days +12:37:27
75%         0.000000       0.000000       1.000000            0 days 08:18:28
max         1.000000       4.000000       1.000000            6 days 13:49:20

                Hour        Present
count  110526.000000  110526.000000
mean       10.774542       0.798066
std         3.216192       0.401445
min         6.000000       0.000000
25%         8.000000       1.000000
50%        10.000000       1.000000
75%        13.000000       1.000000
max        21.000000       1.000000
```

[442]:
```
# Removing the record with negative Age

df[df['Age'] < 0].index
df.drop(df[df['Age'] < 0].index, inplace=True)
```

[443]:
```
# Confirming that the data is removed

df[df['Age'] < 0]
```

[443]:
```
Empty DataFrame
Columns: [PatientId, AppointmentID, Gender, ScheduledDay, AppointmentDay, Age,
Neighbourhood, Scholarship, Hipertension, Diabetes, Alcoholism, Handcap,
SMS_received, Present]
Index: []
```

[444]:
```
# Adding time difference between Appointment and Scheduled

df['WaitingTime'] = df['ScheduledDay'] - df['AppointmentDay']
dates = df['WaitingTime'].abs()

def dayCount(dates):
    return dates.days

dates = dates.map(dayCount)
df.head()
```

[444]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	Present	WaitingTime
0	29872499824296	5642903	F	2016-04-29 18:38:08	2016-04-29	62	JARDIM DA PENHA	0	1	0	0	0	0	1	18:38:08
1	558997776694438	5642503	M	2016-04-29 16:08:27	2016-04-29	56	JARDIM DA PENHA	0	0	0	0	0	0	1	16:08:27
2	4262962299951	5642549	F	2016-04-29 16:19:04	2016-04-29	62	MATA DA PRAIA	0	0	0	0	0	0	1	16:19:04
3	867951213174	5642828	F	2016-04-29 17:29:31	2016-04-29	8	PONTAL DE CAMBURI	0	0	0	0	0	0	1	17:29:31
4	8841186448183	5642494	F	2016-04-29 16:07:23	2016-04-29	56	JARDIM DA PENHA	0	1	1	0	0	0	1	16:07:23

[445]:
```
# Adding month of Appointment and Day of Appointment

df['Month'] = df['AppointmentDay'].dt.month_name()
df['Day'] = df['AppointmentDay'].dt.day_name()
df['Hour'] = df['ScheduledDay'].dt.hour
```

[446]:
```
# Reorder Columns

column_order = df.columns.tolist()
column_order =␣
 ↪['PatientId','AppointmentID','Gender','ScheduledDay','AppointmentDay','Age','Neighbourhood'
                ␣
 ↪'Hipertension','Diabetes','Alcoholism','Handcap','SMS_received','WaitingTime','Month','Day'
                 'Present']
df = df[column_order]
df.shape
```

[446]:
```
(110526, 18)
```

[447]:
```
df.head()
```

[447]:
```
          PatientId AppointmentID Gender        ScheduledDay AppointmentDay  \
0   29872499824296       5642903      F 2016-04-29 18:38:08     2016-04-29
1  558997776694438       5642503      M 2016-04-29 16:08:27     2016-04-29
2    4262962299951       5642549      F 2016-04-29 16:19:04     2016-04-29
3     867951213174       5642828      F 2016-04-29 17:29:31     2016-04-29
4    8841186448183       5642494      F 2016-04-29 16:07:23     2016-04-29

   Age      Neighbourhood  Scholarship  Hipertension  Diabetes  Alcoholism  \
0   62    JARDIM DA PENHA            0             1         0           0
1   56    JARDIM DA PENHA            0             0         0           0
2   62      MATA DA PRAIA            0             0         0           0
3    8  PONTAL DE CAMBURI            0             0         0           0
4   56    JARDIM DA PENHA            0             1         1           0

   Handcap  SMS_received WaitingTime  Month     Day  Hour  Present
0        0             0    18:38:08  April  Friday    18        1
1        0             0    16:08:27  April  Friday    16        1
2        0             0    16:19:04  April  Friday    16        1
3        0             0    17:29:31  April  Friday    17        1
4        0             0    16:07:23  April  Friday    16        1
```

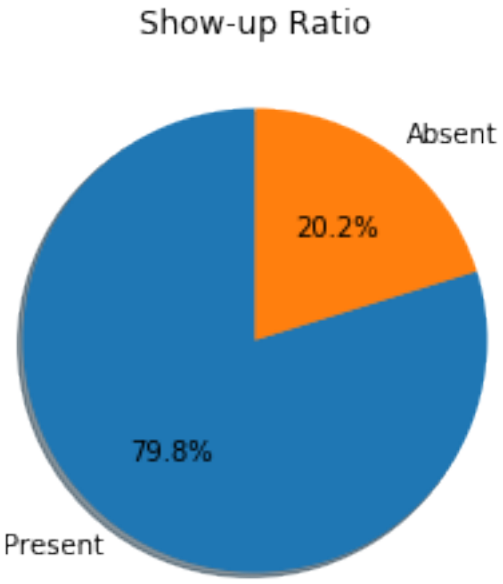
Exploratory Data Analysis

> **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.

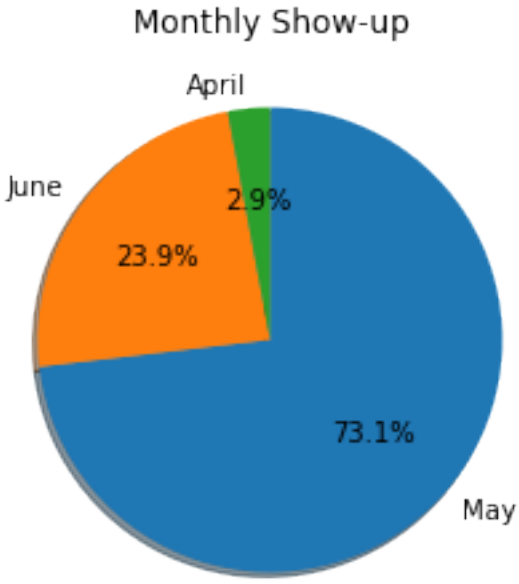
Home ### Research Question 1 (Replace this header name!)

0.2.3 Basic plots based on the available columns

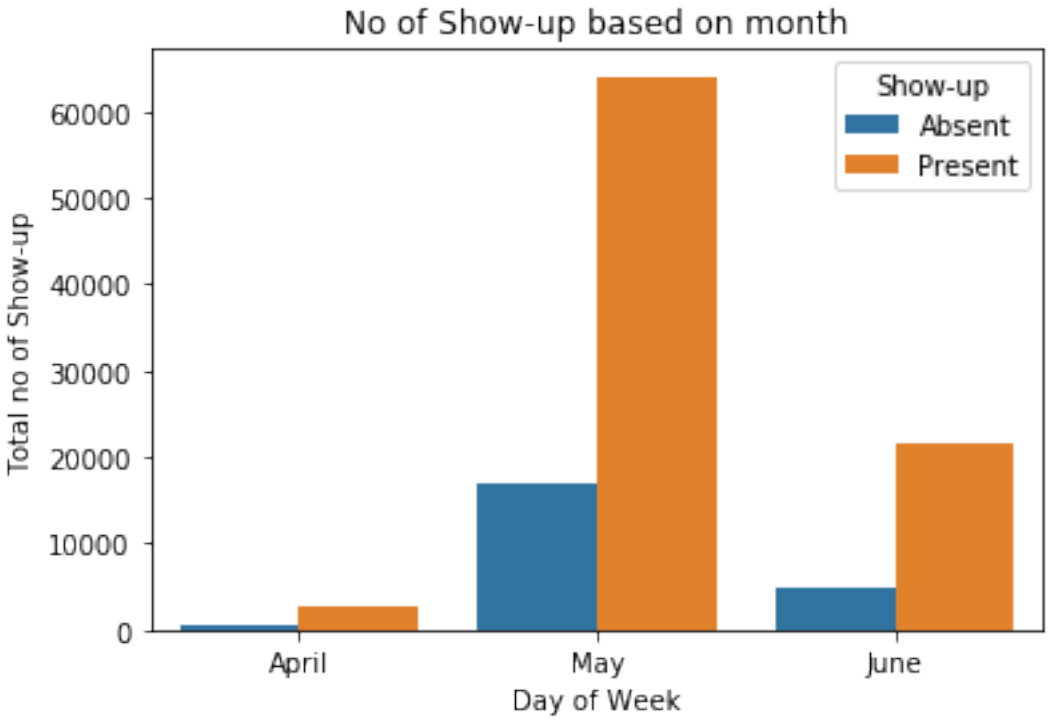
[401]:
```
temp = df['Present'].value_counts().to_list()
x_marker = ['Present', 'Absent']
plt.pie(temp, labels = x_marker, autopct='%1.1f%%', shadow=True, startangle=90)
plt.title('Show-up Ratio');
```



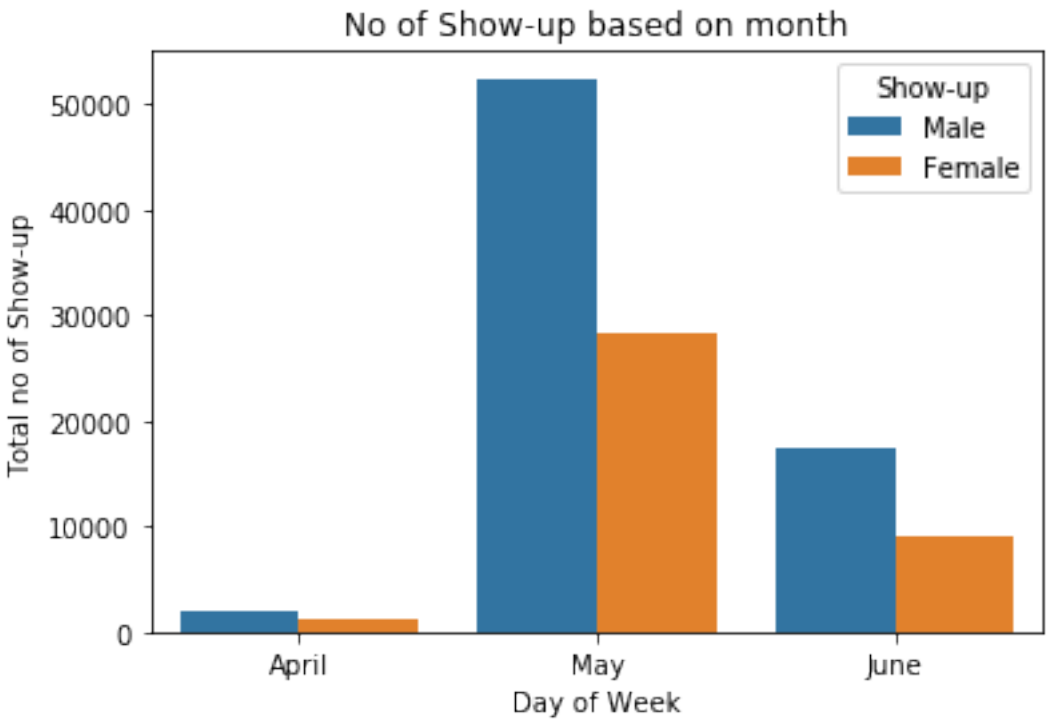
[402]:
```
temp = df['Month'].value_counts().to_list()
x_marker = df['Month'].value_counts().index.tolist()
plt.pie(temp, labels = x_marker, autopct='%1.1f%%', shadow=True,␣
 ↪startangle=90,counterclock=False)
plt.title('Monthly Show-up');
```



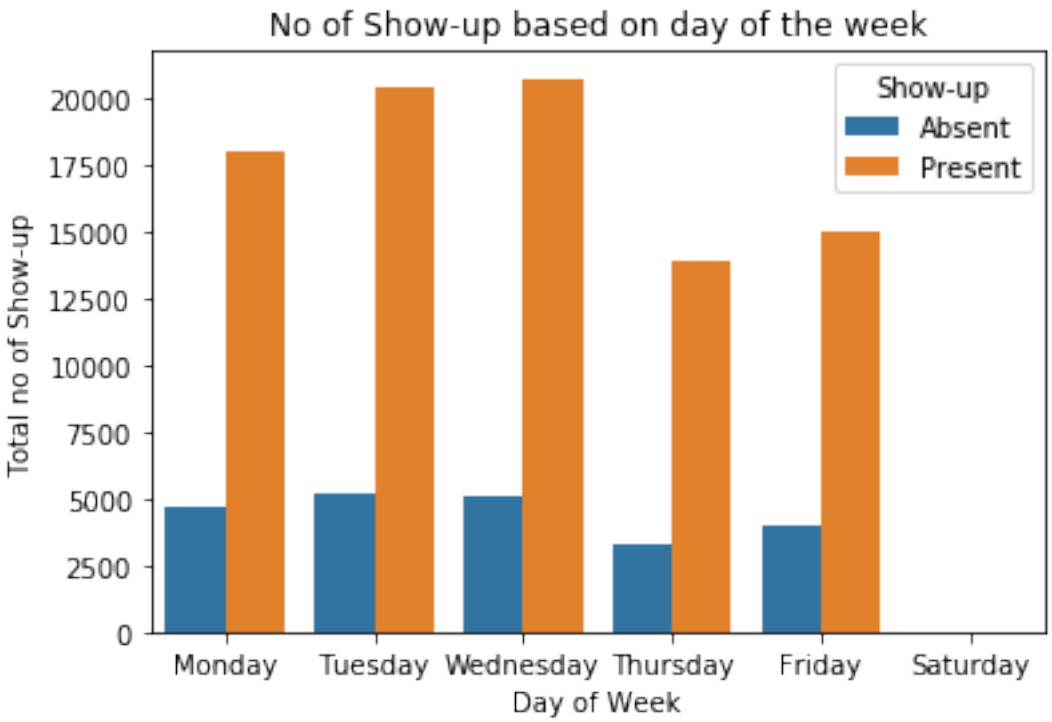
[403]:
```
sb.countplot(data=df, x='Month', hue='Present');
plt.title('No of Show-up based on month')
plt.legend(['Absent','Present'], title='Show-up');
plt.xlabel('Day of Week')
plt.ylabel('Total no of Show-up');
```



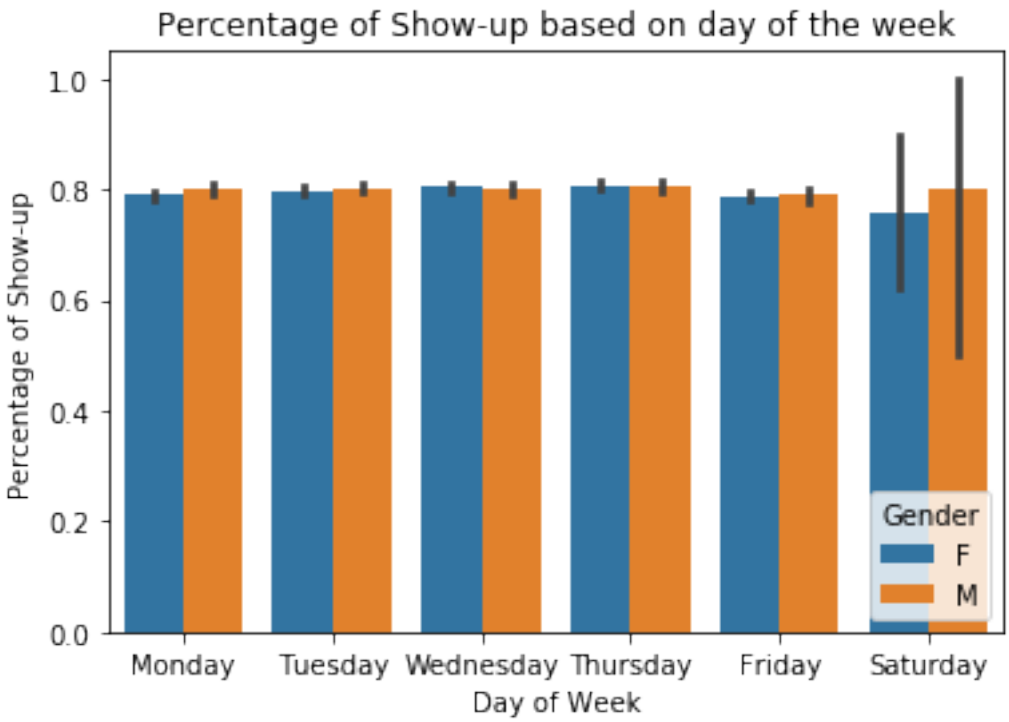
[404]:
```
sb.countplot(data=df, x='Month', hue='Gender');
plt.title('No of Show-up based on month')
plt.legend(['Male','Female'], title='Show-up');
plt.xlabel('Day of Week')
plt.ylabel('Total no of Show-up');
```



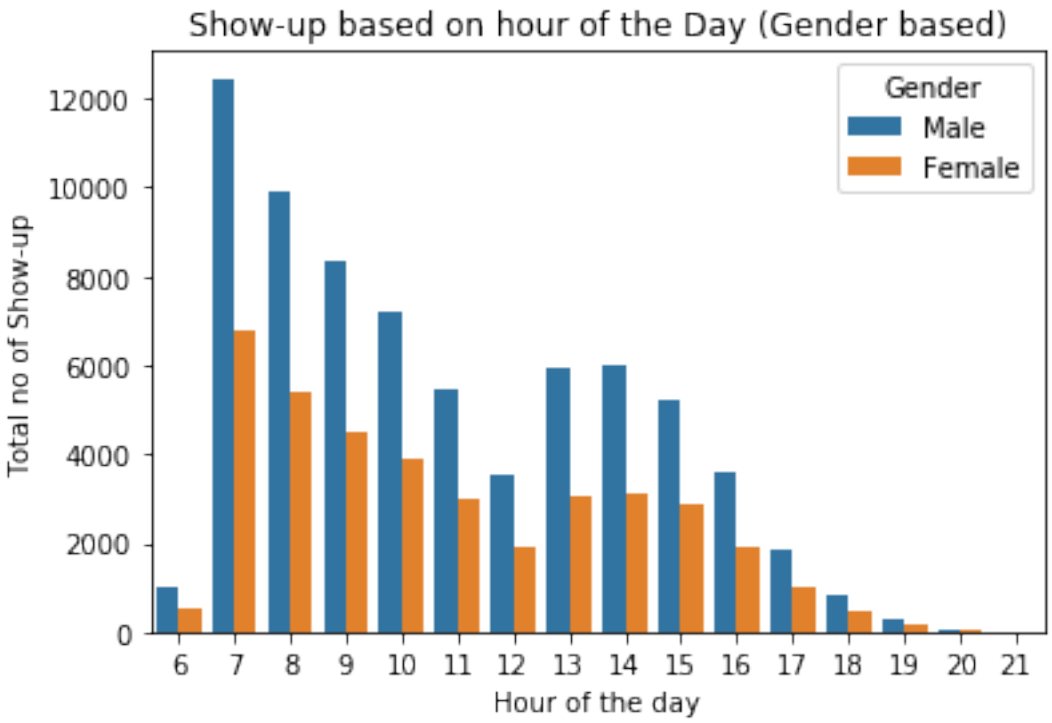
[405]:
```
x_marker = ['Monday', 'Tuesday', 'Wednesday', 'Thursday','Friday', 'Saturday']
sb.countplot(data=df, x='Day', hue='Present', order=x_marker);
plt.title('No of Show-up based on day of the week')
plt.legend(['Absent','Present'], title='Show-up');
plt.xlabel('Day of Week')
plt.ylabel('Total no of Show-up');
```



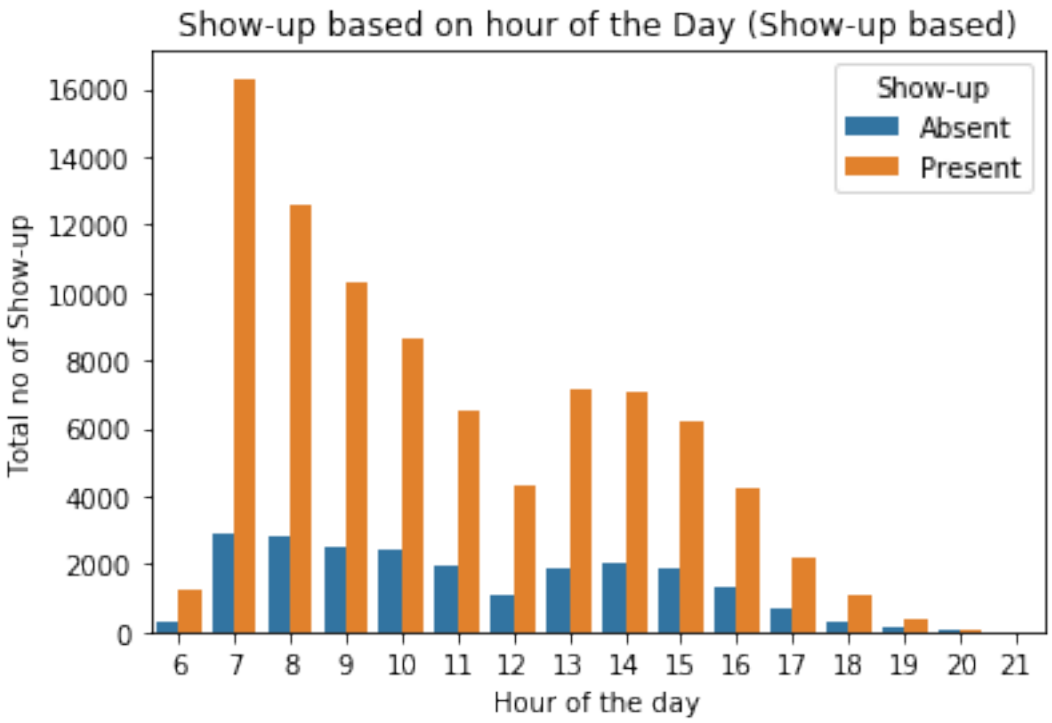
[406]:
```
x_marker = ['Monday', 'Tuesday', 'Wednesday', 'Thursday','Friday', 'Saturday']
sb.barplot(data=df, x='Day', y='Present', hue='Gender', order=x_marker);
plt.title('Percentage of Show-up based on day of the week');
plt.legend(loc='lower right', title='Gender');
plt.xlabel('Day of Week')
plt.ylabel('Percentage of Show-up');
```



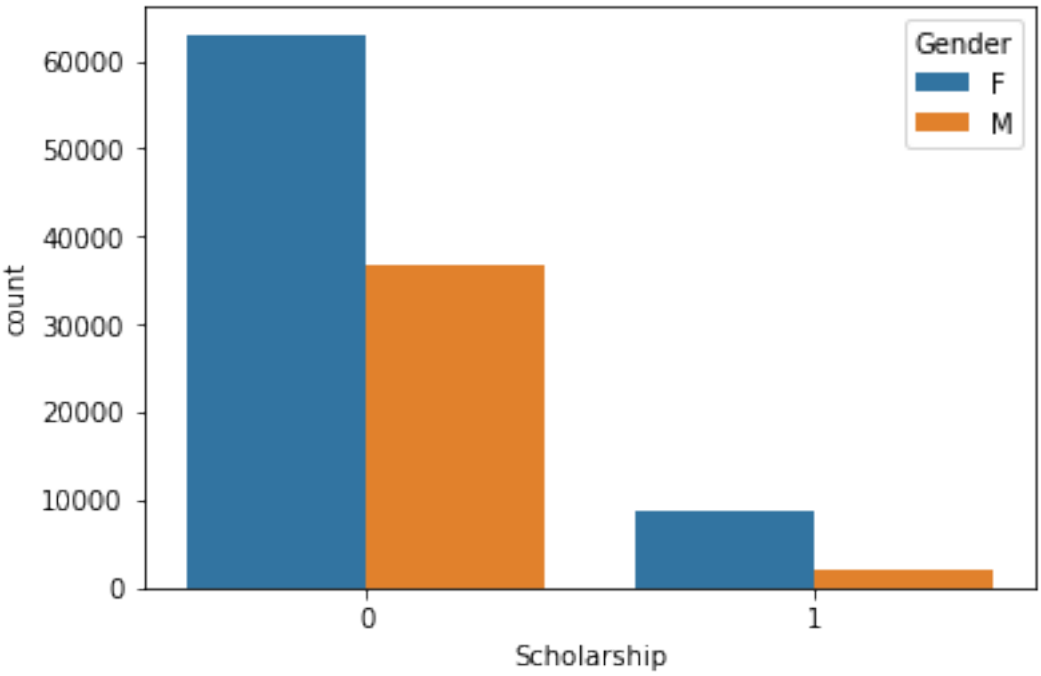
[407]:
```
df.groupby('Hour').count()
# x_marker = ['Monday', 'Tuesday', 'Wednesday', 'Thursday','Friday',␣
 ↪'Saturday']
sb.countplot(data=df, x='Hour', hue='Gender');
plt.legend(['Male','Female'], title='Gender');
plt.xlabel('Hour of the day')
plt.ylabel('Total no of Show-up')
plt.title('Show-up based on hour of the Day (Gender based)');
```



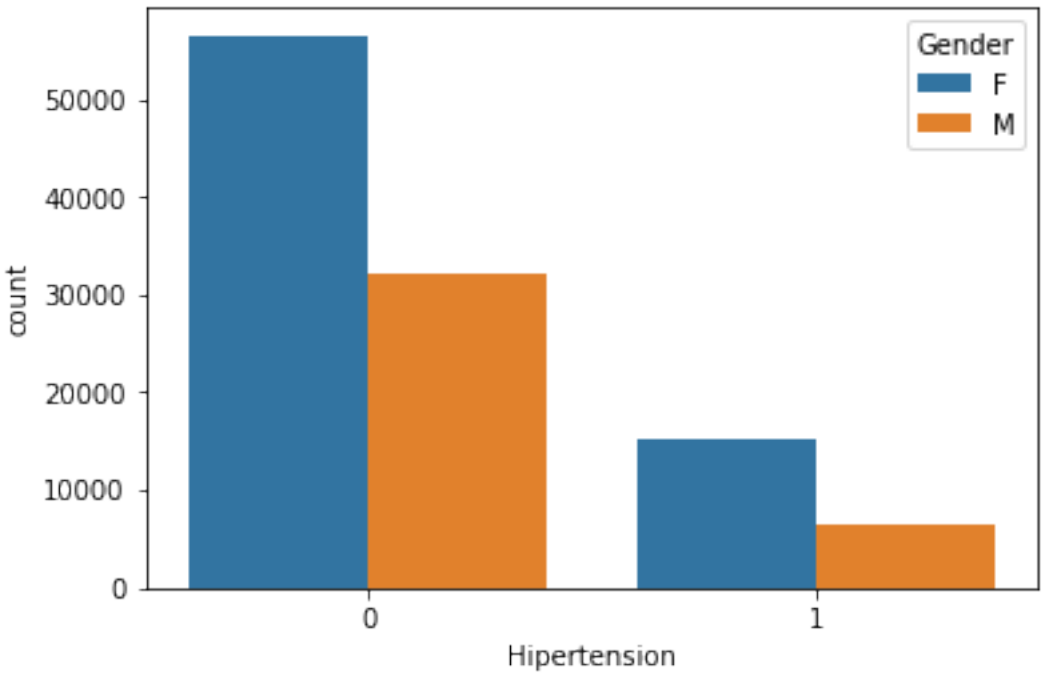
[408]:
```
sb.countplot(data=df, x='Hour', hue='Present');
plt.legend(['Absent','Present'], title='Show-up');
plt.xlabel('Hour of the day')
plt.ylabel('Total no of Show-up')
plt.title('Show-up based on hour of the Day (Show-up based)');
```



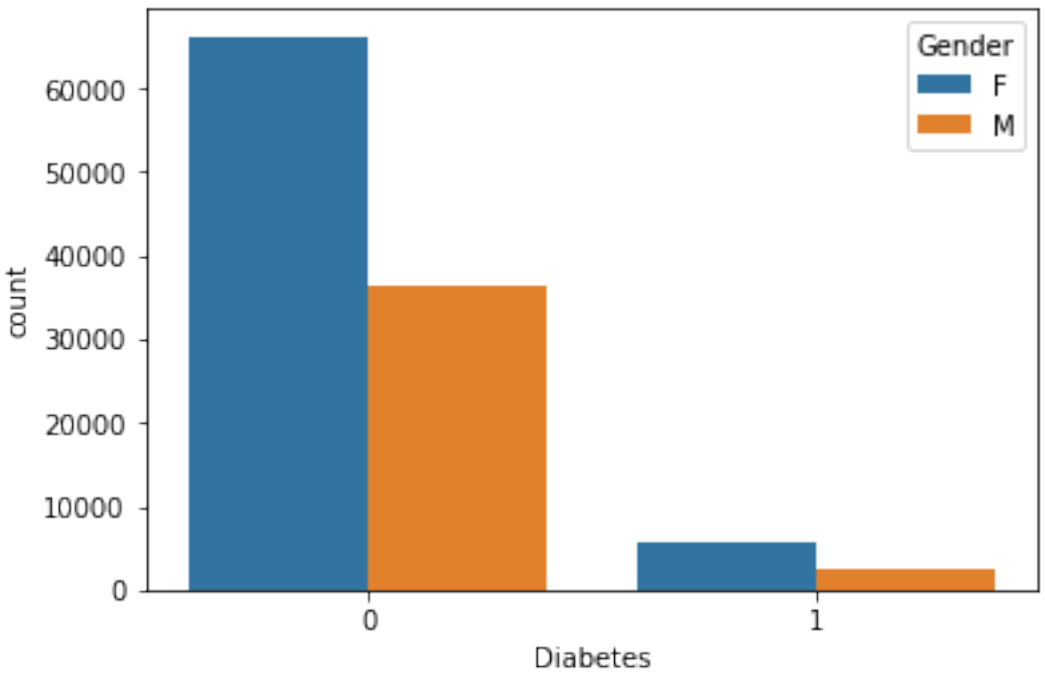
[409]:
```
sb.countplot(data=df, x='Scholarship', hue='Gender');
```



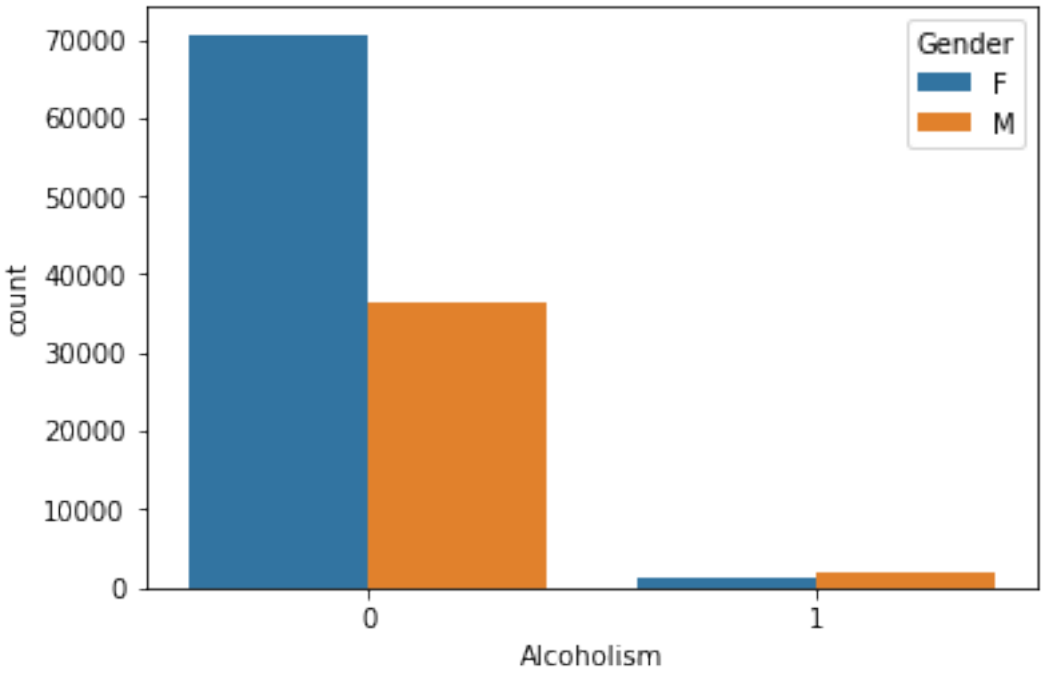
[410]:
```
sb.countplot(data=df, x='Hipertension', hue='Gender');
```



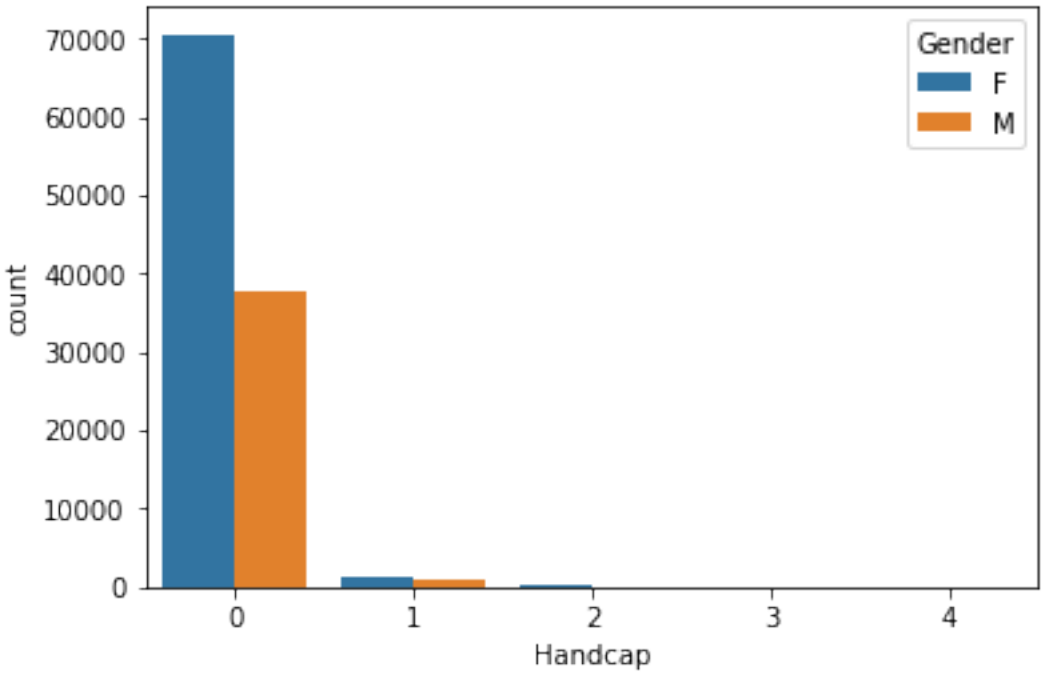
[411]:
```
sb.countplot(data=df, x='Diabetes', hue='Gender');
```



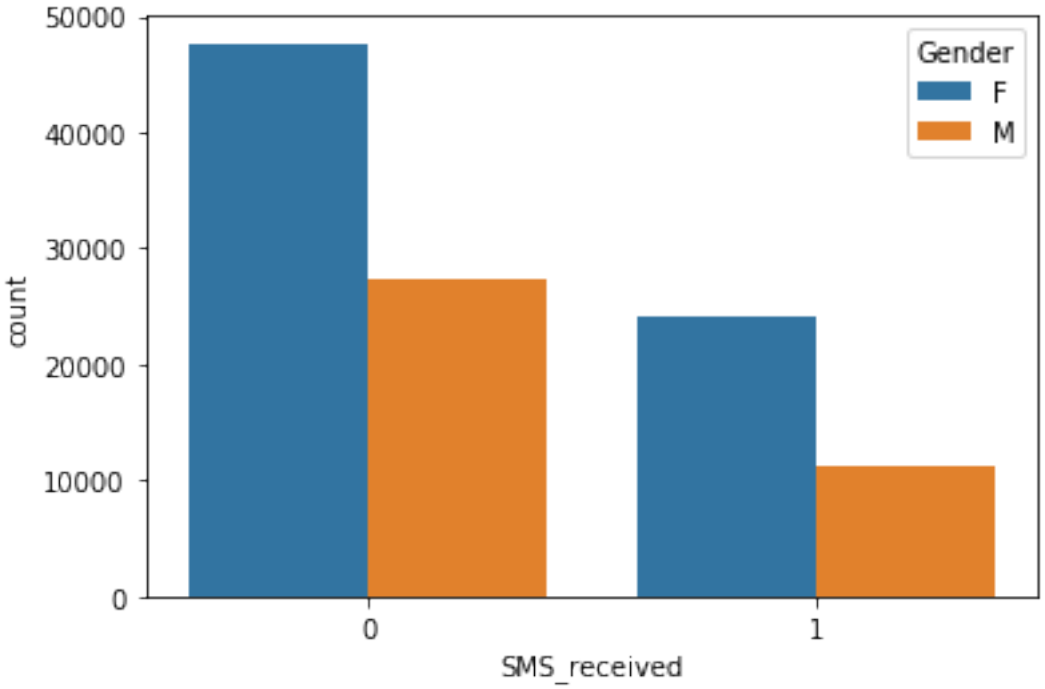
[412]:
```
sb.countplot(data=df, x='Alcoholism', hue='Gender');
```



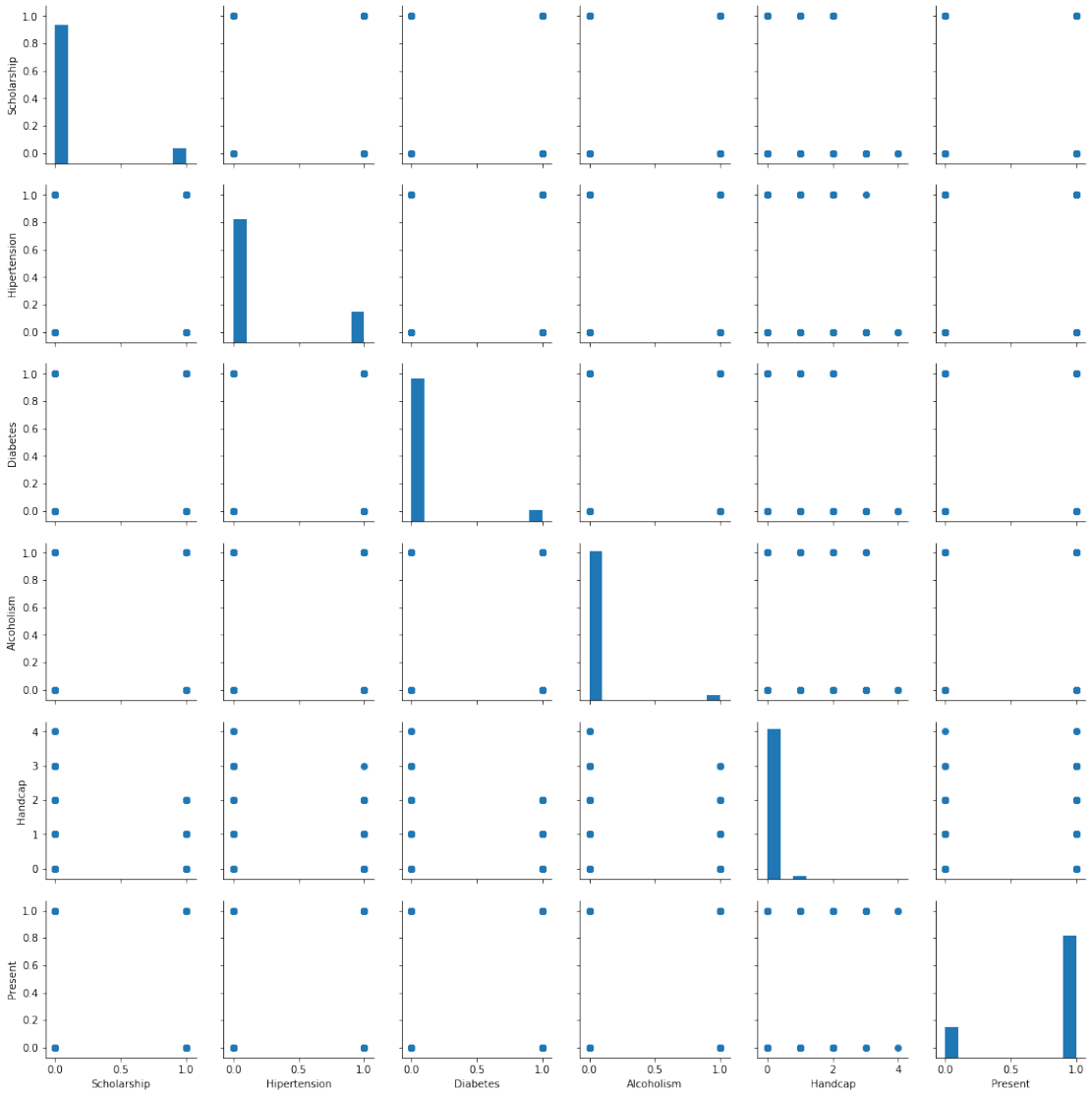
[413]:
```
sb.countplot(data=df, x='Handcap', hue='Gender');
```



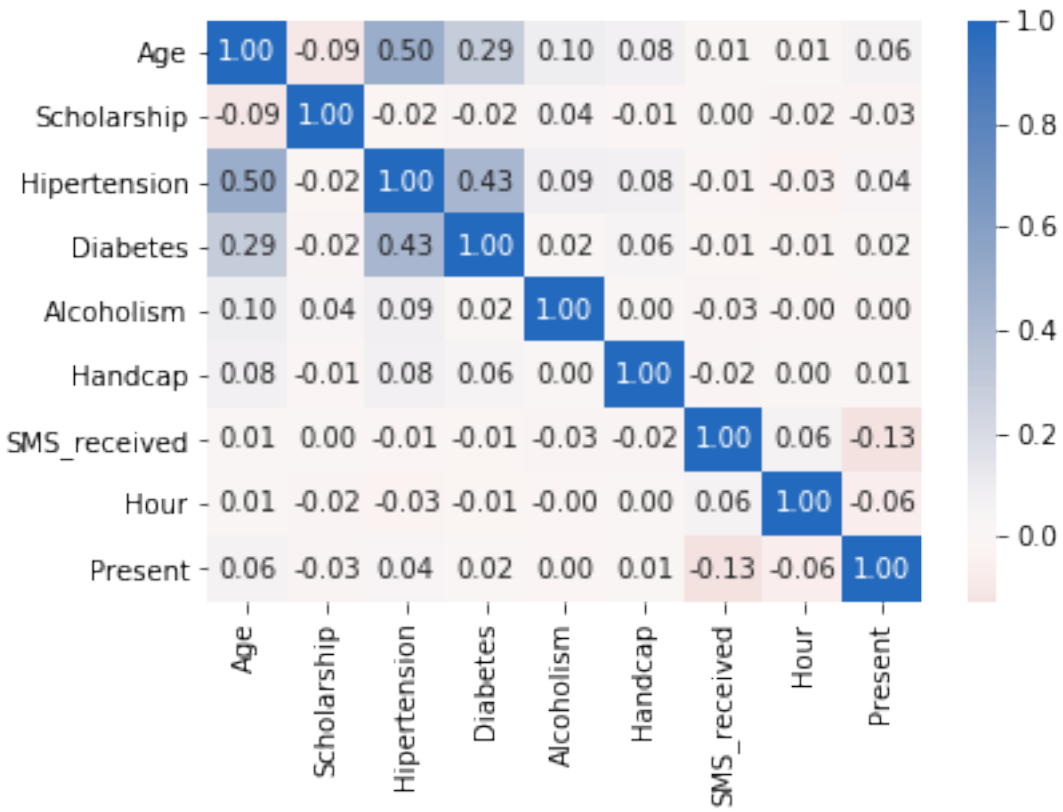
[414]:
```
sb.countplot(data=df, x='SMS_received', hue='Gender');
```



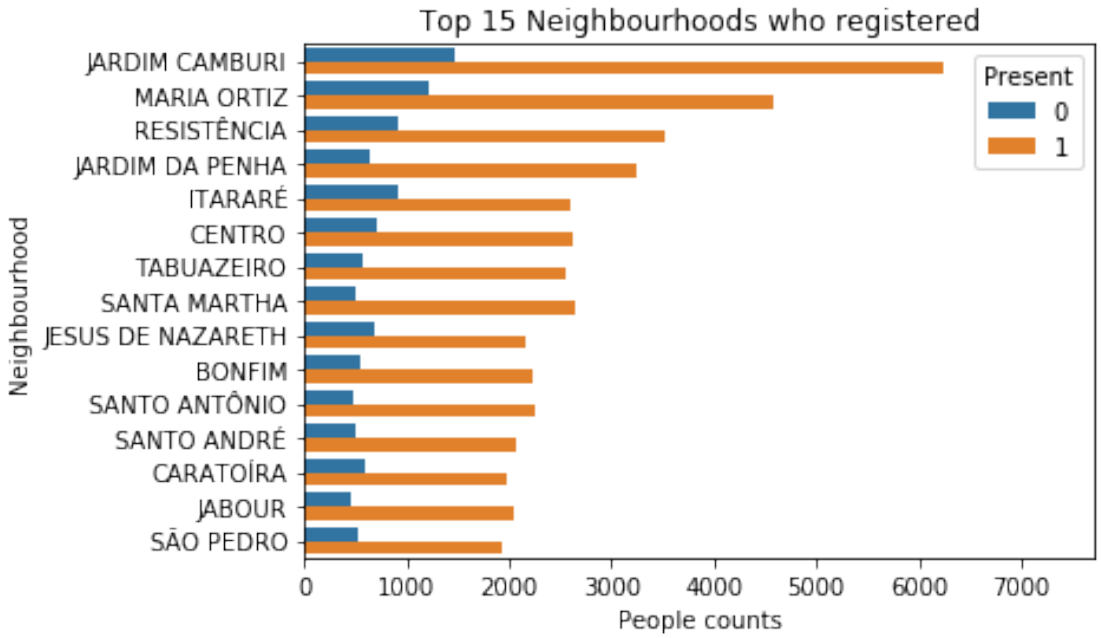
[423]:
```
g = sb.PairGrid(data = df, vars = ['Scholarship', 'Hipertension', 'Diabetes',␣
↪'Alcoholism', 'Handcap', 'Present'])
g.map_diag(plt.hist)
g.map_offdiag(plt.scatter);
```



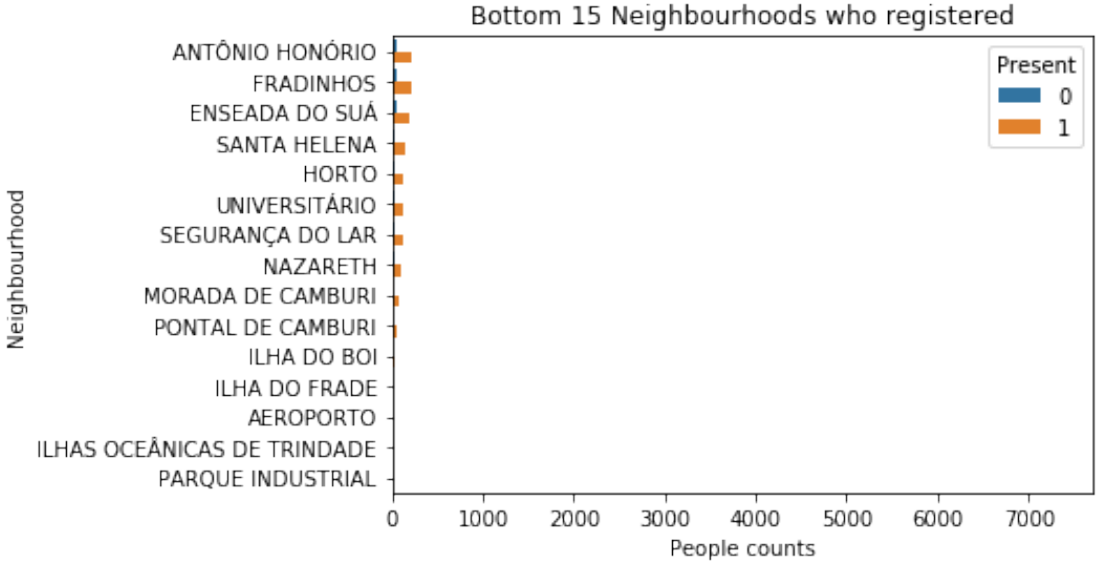
[451]:
```
sb.heatmap(df.corr(), annot = True, fmt = '.2f', cmap = 'vlag_r', center = 0);
```



[417]:
```
neighbourhood_counts = df['Neighbourhood'].value_counts()
neighbourhood_order = neighbourhood_counts.index
plt.xlim(0,df['Neighbourhood'].value_counts().max())
# base_color = sb.color_palette()[3]
sb.countplot(data = df, y = 'Neighbourhood', hue='Present', order =␣
 ↪neighbourhood_order[:15])
plt.xlabel('People counts')
plt.title('Top 15 Neighbourhoods who registered');
```



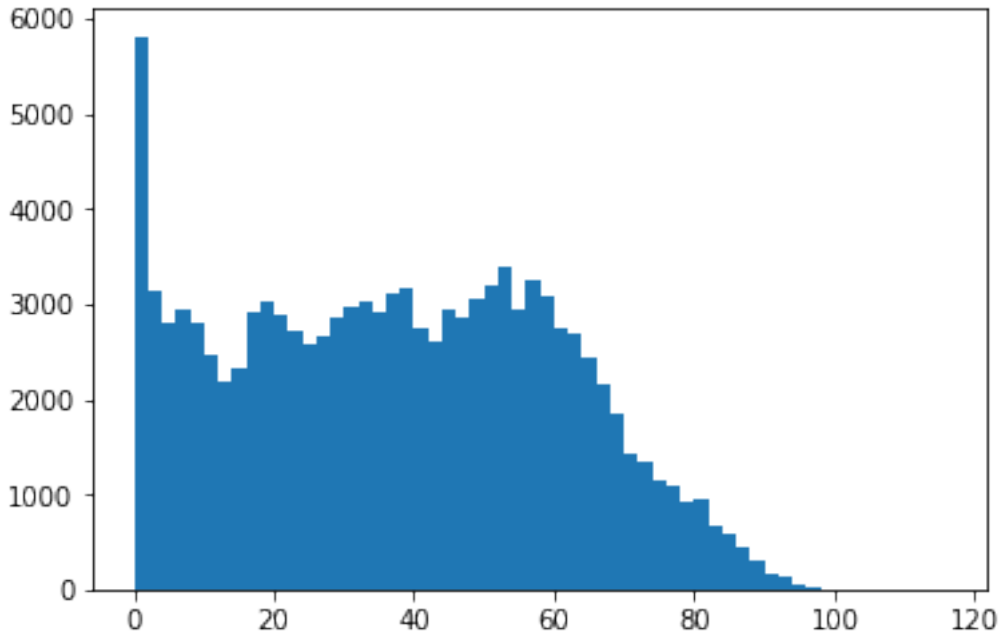
[418]:
```
sb.countplot(data = df, y = 'Neighbourhood', hue='Present', order =␣
 ↪neighbourhood_order[-15:])
plt.xlim(0,df['Neighbourhood'].value_counts().max())
plt.xlabel('People counts')
plt.title('Bottom 15 Neighbourhoods who registered');
# df['Neighbourhood'].value_counts().max()
```



0.2.4 Research Question 2 (Replace this header name!)

Dependency of show-up as based on the ages of the people

[419]:
```
bin_size = np.arange(df['Age'].min(), df['Age'].max()+2, 2)
plt.hist(df['Age'], bins=bin_size);
```



Conclusions

Tip: Finally, summarize your findings and the results that have been performed. Make sure that you are clear with regards to the limitations of your exploration. If you haven't done any statistical tests, do not imply any statistical conclusions. And make sure you avoid implying causation from correlation!

Tip: Once you are satisfied with your work, you should save a copy of the report in HTML or PDF form via the **File** > **Download as** submenu. Before exporting your report, check over it to make sure that the flow of the report is complete. You should probably remove all of the "Tip" quotes like this one so that the presentation is as tidy as possible. Congratulations!

Home

0.2.5 Age

Mostly infants less than 2 years of age attend the appointment

0.2.6 Waiting Time

0.2.7 Gender

More no of males show-up for the appointments

0.2.8 Hipertension and Diabetes

These are the two most important factors which influenced the show-up ratio

0.2.9 Month

Most appointments were made in May

0.2.10 Day

Wednesday has most sshow-ups and Saturday has the least

0.2.11 Hour

The graph is bimodal with 1st peak between 7 and 9 and 2nd peak between 13 and 15. These two time periods have the most show-ups

0.2.12 Neighbourhood

The top 3 neighbourhood having most show-up are - JARDIM CAMBURI, MARIA ORTIZ, RESISTÊNCIA

The top 3 neighbourhood having least show-up are - AEROPORTO, ILHAS OCEÂNICAS DE TRINDADE, PARQUE INDUSTRIAL'

Home

[ ]: