Project: Investigate a Dataset (Patients noshow appointments-may-2016.csv)

Table of Contents

Introduction

This dataset collects information from 100k medical appointments in Brazil and is focused on the question of whether or not patients show up for their appointment. A number of characteristics about the patient are included in each row.</br> ● 'ScheduledDay' tells us on what day the patient set up their appointment.</br> ● 'Neighborhood' indicates the location of the hospital.</br> ● 'Scholarship' indicates whether or not the patient is enrolled in Brasilian welfare program Bolsa Família.</br> ● Be careful about the encoding of the last column: it says 'No' if the patient showed up to their appointment, and 'Yes' if they did not show up.</br>

In [4]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

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 [7]:

```
# Load your data and print out a few lines. Perform operations to inspect data
#  types and look for instances of missing or possibly errant data.
df=pd.read_csv("no show appointments.csv")
df.head()
```

Out[7]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipe
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	0	
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	0	

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipe
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	0	
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	

In [9]:

```
#get the shape of data(rows and columns)
df.shape
```

Out[9]:

```
(110527, 14)
```

In [11]:

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

Out[11]:

```
0
```

In [20]:

```
#check unique values
x = df["PatientId"]
print("There are {} patients\n only {} patients are unique\n and {} patient are duplicated".format
```

```
There are 110527 patients
 only 62299 patients are unique
 and 48228 patient are duplicated
```

In [25]:

```
df.describe()
```

Out[25]:

	PatientId	AppointmentID	Age	Scholarship	Hipertension	Diabetes	Alcoholism	
count	1.105270e+05	1.105270e+05	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000	1
mean	1.474963e+14	5.675305e+06	37.088874	0.098266	0.197246	0.071865	0.030400	
std	2.560949e+14	7.129575e+04	23.110205	0.297675	0.397921	0.258265	0.171686	
min	3.921784e+04	5.030230e+06	-1.000000	0.000000	0.000000	0.000000	0.000000	
25%	4.172614e+12	5.640286e+06	18.000000	0.000000	0.000000	0.000000	0.000000	
50%	3.173184e+13	5.680573e+06	37.000000	0.000000	0.000000	0.000000	0.000000	
75%	9.439172e+13	5.725524e+06	55.000000	0.000000	0.000000	0.000000	0.000000	
max	9.999816e+14	5.790484e+06	115.000000	1.000000	1.000000	1.000000	1.000000	

in previous result we got age with -1 so we should drop it

In [55]:

```
 index = df.index[df["Age"]<= 0]
index
df["Age"].head()
```

Out[55]:

```
0    62
1    56
```

```
2    62
3     8
4    56
Name: Age, dtype: int64
```

In [ ]:

```

```

In [ ]:

```
df["Age"].head()
```

In [26]:

```
df.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
```

Data Cleaning (Replace this with more specific notes!)

In [59]:

```
# 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 section.
#remove age less than or equal 0
for i in index:
 df.drop(index= i,inplace= True)
```

In [65]:

```
df.duplicated().sum()
#we have no duplicates
```

Out[65]:

```
0
```

In [67]:

```
# drop all unneeded columns
df.drop(["PatientId",'AppointmentID'],axis=1,inplace= True)
```

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.

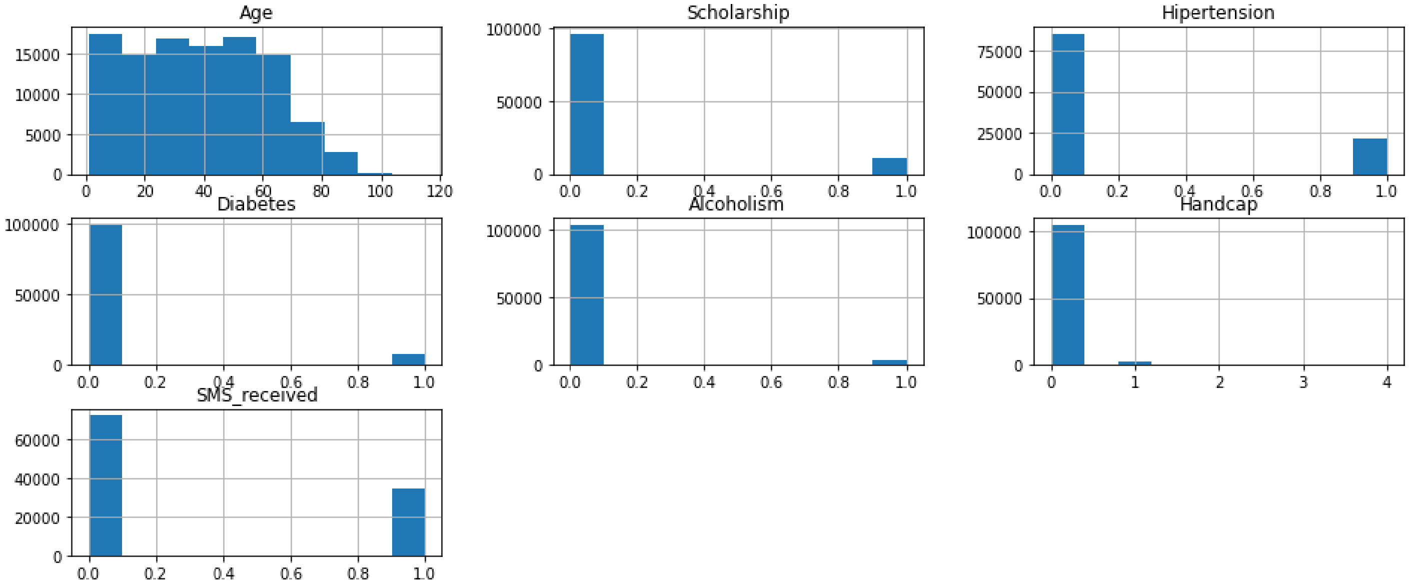
Overview for the whole data

In [68]:

```
df.hist(figsize=(16,6.5))
```

Out[68]:

```
array([[<AxesSubplot:title={'center':'Age'}>,
        <AxesSubplot:title={'center':'Scholarship'}>,
        <AxesSubplot:title={'center':'Hipertension'}>],
       [<AxesSubplot:title={'center':'Diabetes'}>,
        <AxesSubplot:title={'center':'Alcoholism'}>,
        <AxesSubplot:title={'center':'Handcap'}>],
       [<AxesSubplot:title={'center':'SMS_received'}>, <AxesSubplot:>,
        <AxesSubplot:>]], dtype=object)
```



Research Question 2 (Replace this header name!)

In [152…

```
# Continue to explore the data to address your additional research
#   questions. Add more headers as needed if you have more questions to
#   investigate.
# first we get split data into show and no show

show= df['No-show']=='No'
no_show= df['No-show']=="Yes"
```

In [95]:

```
# ratio between show and no show
x= show.count()/no_show.count()
x
```

Out[95]:

```
Gender            3.934825
ScheduledDay      3.934825
AppointmentDay    3.934825
Age               3.934825
Neighbourhood     3.934825
Scholarship       3.934825
Hipertension      3.934825
Diabetes          3.934825
Alcoholism        3.934825
Handcap           3.934825
SMS_received      3.934825
```

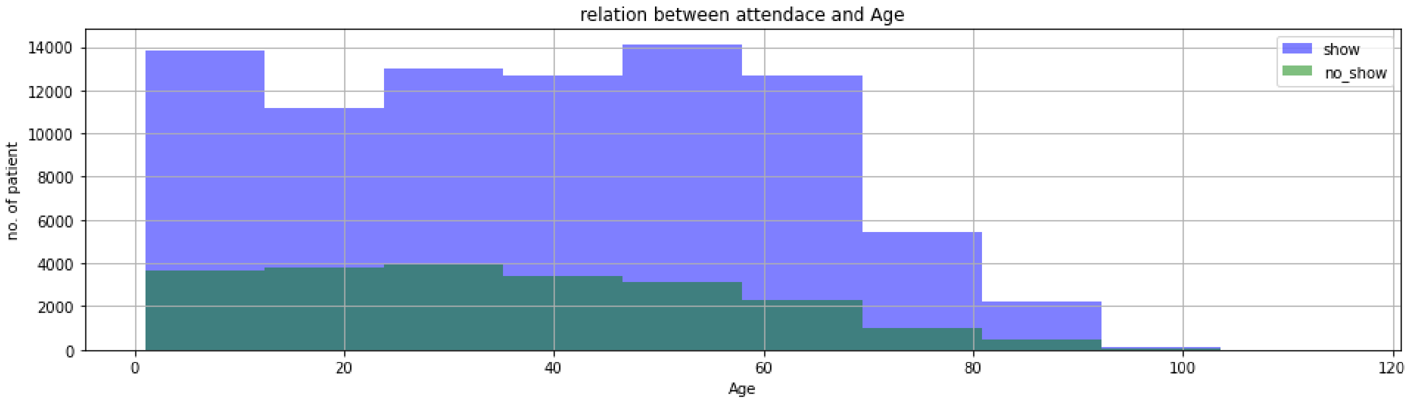
```
No-show            3.934825
dtype: float64
```

As shown above the amount of patients who attend is 4 time the patients who donot lets see the reasons

let us find the relation between no attendance and other vairbales

In [120…

```
#relation between attdence and Age scholarship (government welfare)
def attendace(df,col_name,attend,absent):
    plt.figure(figsize=[16,4])
    df[col_name][show].hist(alpha=0.5,bins=10,color='blue',label="show")
    df[col_name][no_show].hist(alpha=0.5,bins=10,color='green',label="no_show")
    plt.legend()
    plt.xlabel(col_name)
    plt.ylabel("no. of patient")
    plt.title("relation between attendace and {}".format(col_name))
attendace(df,"Age",show,no_show)
```

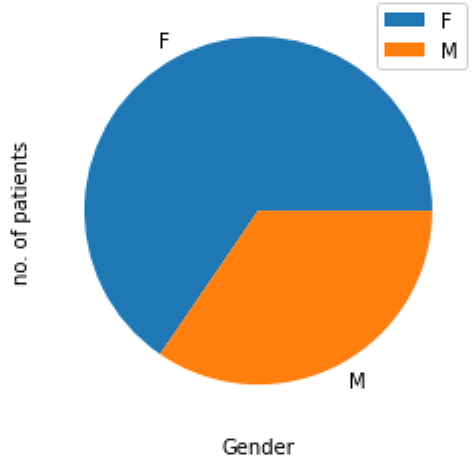


According to the above diagram it shows that age from greater that 0 to 65 approximately go to appointments but after that they dont go

Does gender affect attendace??

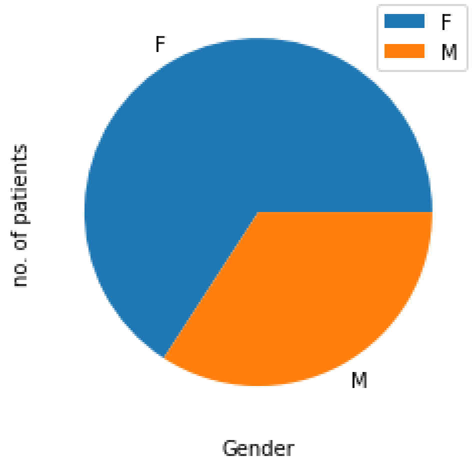
In [115…

```
    plt.figure(figsize=[16,4])
    df["Gender"][show].value_counts(normalize=True).plot(kind="pie",label="show")
    plt.legend()
    plt.xlabel("Gender")
    plt.ylabel("no. of patients")
    plt.show()
```



In [117…

```
    plt.figure(figsize=[16,4])
    df["Gender"][no_show].value_counts(normalize=True).plot(kind="pie",label="show")
    plt.legend()
    plt.xlabel("Gender")
    plt.ylabel("no. of patients")
    plt.show()
```

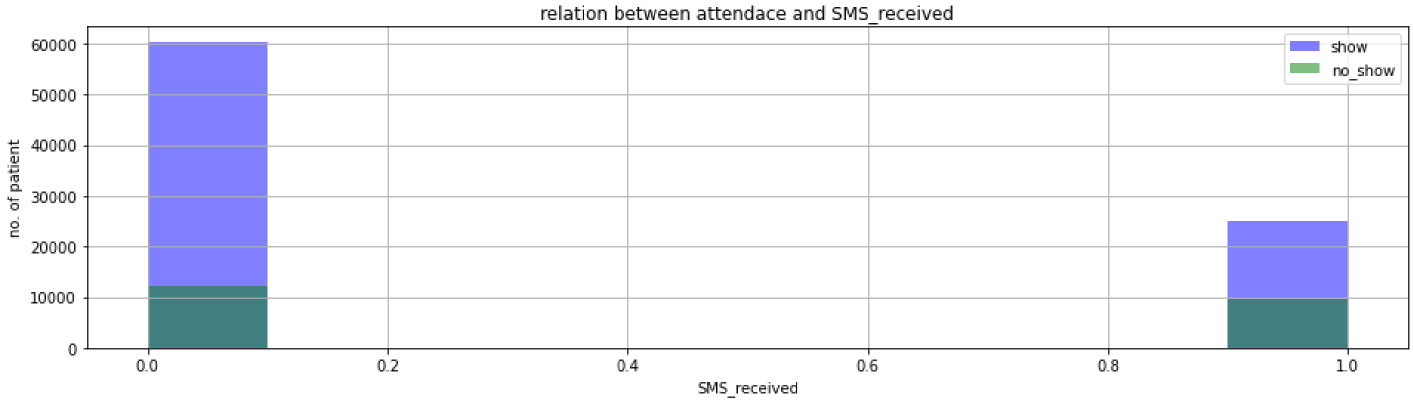


there is no relation between gender and attendance

Does recieving SMS affect attendance

In [121…

```
attendace(df,"SMS_received",show,no_show)
```



As shown here there is no relation bettween SMS recieving and attendance as the patient who donot recieve SMS still donot come

and majority come despite not recieving SMS

Does Scholarship affect attendance?

In [143…

```
# the Answer is no because only 12% of attended person are included in Scolarship
print("No of patients with Scolarships  is {}, And whe number attended Is {} ".format(df["Scholars
```

```
No of patients with Scolarships  is 10809, And whe number attended Is Gender            85307
ScheduledDay      85307
AppointmentDay    85307
Age               85307
Neighbourhood     85307
Scholarship       85307
Hipertension      85307
Diabetes          85307
Alcoholism        85307
Handcap           85307
SMS_received      85307
No-show           85307
dtype: int64
```

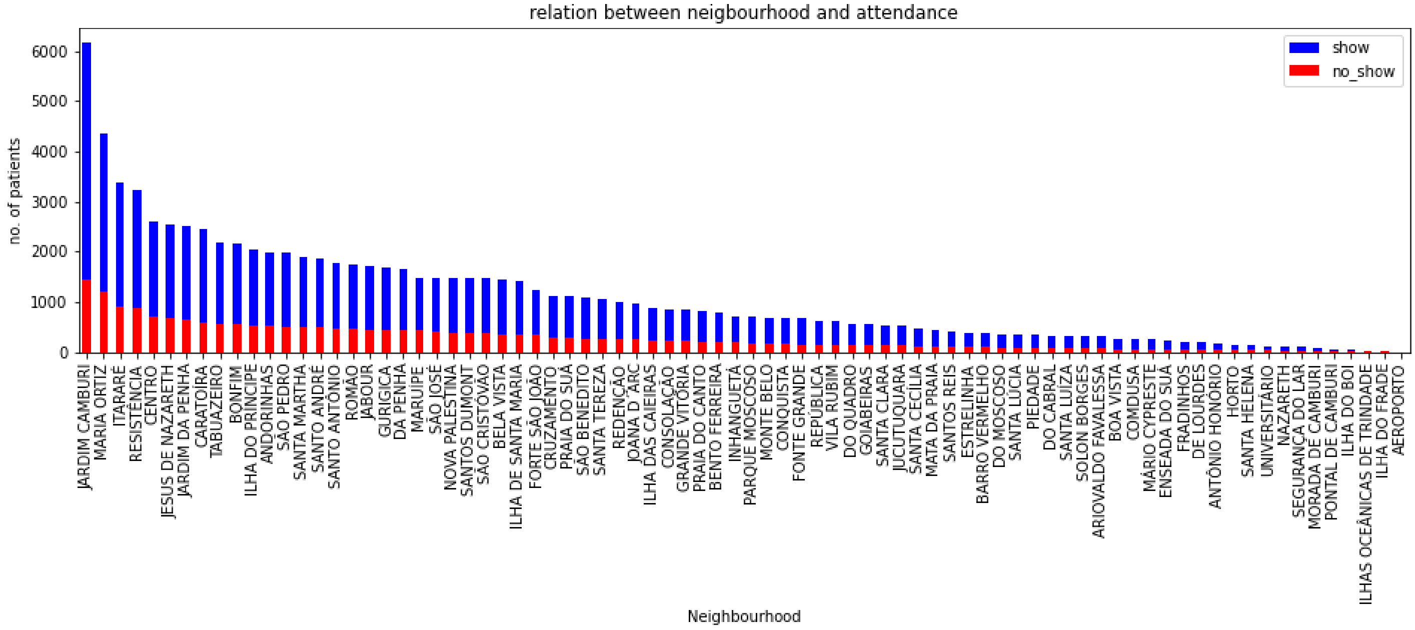
Is there a relation between neigbourhood and attendance

In [172…

```
plt.figure(figsize=[16,4])
df.Neighbourhood[show].value_counts().plot(kind='bar',color="blue",label="show")
df.Neighbourhood[no_show].value_counts().plot(kind="bar",color="red",label="no_show")
plt.legend()
plt.xlabel("Neighbourhood")
plt.ylabel("no. of patients")
plt.title("relation between neigbourhood and attendance")
```

Out[172…

```
Text(0.5, 1.0, 'relation between neigbourhood and attendance')
```



According to the previous graph Neighbourhood has a great influence on attendence

Conclusions

After many investigations it is shown that Neighbourhood has a great

influence on attendence