Investigate_a_Dataset

July 15, 2022

1 Project: no show dataset

1.1 Table of Contents

Introduction
    Data Wrangling
    Exploratory Data Analysis
    Conclusions
    ## Introduction

1.1.1 Dataset Description

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. 'ScheduledDay' tells us on what day the patient set up their appointment. 'Neighborhood' indicates the location of the hospital. 'Scholarship' indicates whether or not the patient is enrolled in Brasilian welfare program Bolsa Família. 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.

1.2 importing libraries

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt

        import seaborn as snb
        %matplotlib inline
        # to show plots in jupyter notebook
        # libraries that i will use
```

```
In [ ]:
```

    ## Data Wrangling

```
In [2]: # Loading the data

        df= pd.read_csv("noshowappointments-kagglev2-may-2016.csv")
        df.head()
```

Out[2]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	0	1	0	0	0	0	No
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	0	0	0	0	0	No
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	0	0	0	0	0	0	No
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	0	0	0	0	0	0	No
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	1	1	0	0	0	No

2 reading data

3 here is a mathmatical summary of data

In [3]:
```
df.describe()
```

```
# min year is -1 and this is impossible ,so that is a wrong value so i will deal with th
```

Out[3]:

	PatientId	AppointmentID	Age	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap
count	1.105270e+05	1.105270e+05	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000
mean	1.474963e+14	5.675305e+06	37.088874	0.098266	0.197246	0.071865	0.030400	0.022248
std	2.560949e+14	7.129575e+04	23.110205	0.297675	0.397921	0.258265	0.171686	0.161543
min	3.921784e+04	5.030230e+06	-1.000000	0.000000	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	0.000000
50%	3.173184e+13	5.680573e+06	37.000000	0.000000				
75%	9.439172e+13	5.725524e+06	55.000000	0.000000				
max	9.999816e+14	5.790484e+06	115.000000	1.000000				

```
50%          0.000000       0.000000       0.000000       0.000000
75%          0.000000       0.000000       0.000000       0.000000
max          1.000000       1.000000       1.000000       4.000000

        SMS_received
count  110527.000000
mean        0.321026
std         0.466873
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         1.000000
```

Average age is 37-38 , Oldest age is 115

```
In [4]: df.duplicated().sum()
```

```
Out[4]: 0
```

that is mean that there is no duplicate values

```
In [5]: df["PatientId"].duplicated().sum()
```

```
Out[5]: 48228
```

means there is 48228 duplicated ID

```
In [6]: df['PatientId'].nunique()
```

```
Out[6]: 62299
```

there is 62299 unique values out of 110527

```
In [7]: df.duplicated(["PatientId","No-show"]).sum()
```

```
Out[7]: 38710
```

mean that there is a number 38710 of duplicated tries for patients to attend so , i will drop these

```
In [8]: df.drop_duplicates(['PatientId','No-show'],inplace= True)
        df.shape
```

```
Out[8]: (71817, 14)
```

I had dropped the duplicated tries for the patients. the dataset contains 71817 appointments.

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [9]: df.drop(['PatientId','AppointmentID','ScheduledDay','AppointmentDay'],axis=1,inplace=Tru
        #checking
        df.head()
```

Out[9]:

	Gender	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	F	62	JARDIM DA PENHA	0	1	0	0	0	0	No
1	M	56	JARDIM DA PENHA	0	0	0	0	0	0	No
2	F	62	MATA DA PRAIA	0	0	0	0	0	0	No
3	F	8	PONTAL DE CAMBURI	0	0	0	0	0	0	No
4	F	56	JARDIM DA PENHA	0	1	1	0	0	0	No

Iam dropping these columns (patient id ,appointmentid,ScheduledDay,AppointmentDay) As it will not help me in my analysis

```
In [10]: df.info()
         # here  i found that there is no empty cell
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 71817 entries, 0 to 110524
Data columns (total 10 columns):
Gender           71817 non-null object
Age              71817 non-null int64
Neighbourhood    71817 non-null object
Scholarship      71817 non-null int64
Hipertension     71817 non-null int64
Diabetes         71817 non-null int64
Alcoholism       71817 non-null int64
Handcap          71817 non-null int64
SMS_received     71817 non-null int64
No-show          71817 non-null object
dtypes: int64(7), object(3)
memory usage: 6.0+ MB
```

4 there is no missing values

5 note

There Is a wrong value in age=-1 so I will delete that appointment

```
In [11]: wrong_Ans=df.query("Age==-1")
         wrong_Ans
```

Out[11]:

	Gender	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
99832	F	-1	ROMÃO	0	0	0	0	0	0	No

```
In [12]: df.drop(index=99832,inplace=True)
         # here i found that there is only one appointment containing -1 age so i will drop that
         # checking
         wrong_Ans=df.query("Age==-1")
         wrong_Ans
```

```
Out[12]: Empty DataFrame
         Columns: [Gender, Age, Neighbourhood, Scholarship, Hipertension, Diabetes, Alcoholism,
         Index: []
```

```
In [ ]:
```

6 I will rename No-show to No_show

```
In [13]: df.rename(columns={"No-show":"No_show"},inplace=True)
         df.rename(columns={"Hipertension":"Hypertension"},inplace=True)
         df.head()
```

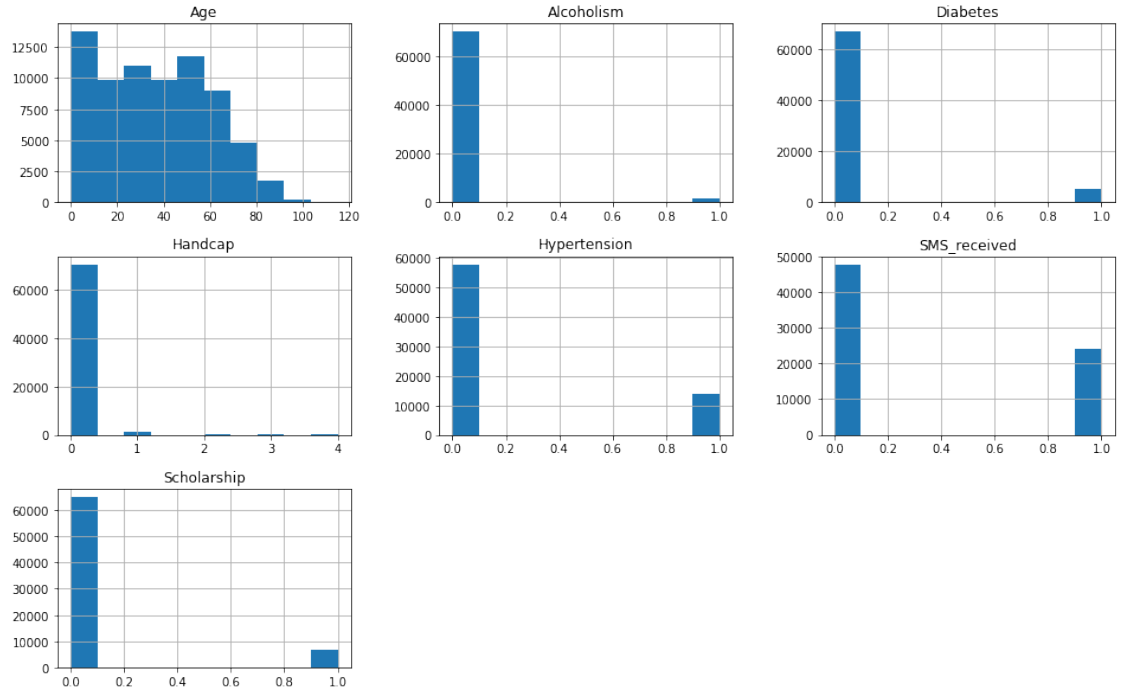
Out[13]:

	Gender	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	No_show
0	F	62	JARDIM DA PENHA	0	1	0	0	0	0	No
1	M	56	JARDIM DA PENHA	0	0	0	0	0	0	No
2	F	62	MATA DA PRAIA	0	0	0	0	0	0	No
3	F	8	PONTAL DE CAMBURI	0	0	0	0	0	0	No
4	F	56	JARDIM DA PENHA	0	1	1	0	0	0	No

6.0.1 Research Question 1 (Age of customers , get to know customers)

here is a describe of the customer that we deal with
most of them are not taking alcohol or diabetes
most of them are not HANDCAP
most of them are not hipertension but there is a percent of hipertension(15000)
most of them have received sms

```
In [14]: df.hist(figsize=(16,10));
```



7 i will divide appointments into two group, the group that had attend and the group that hadnot

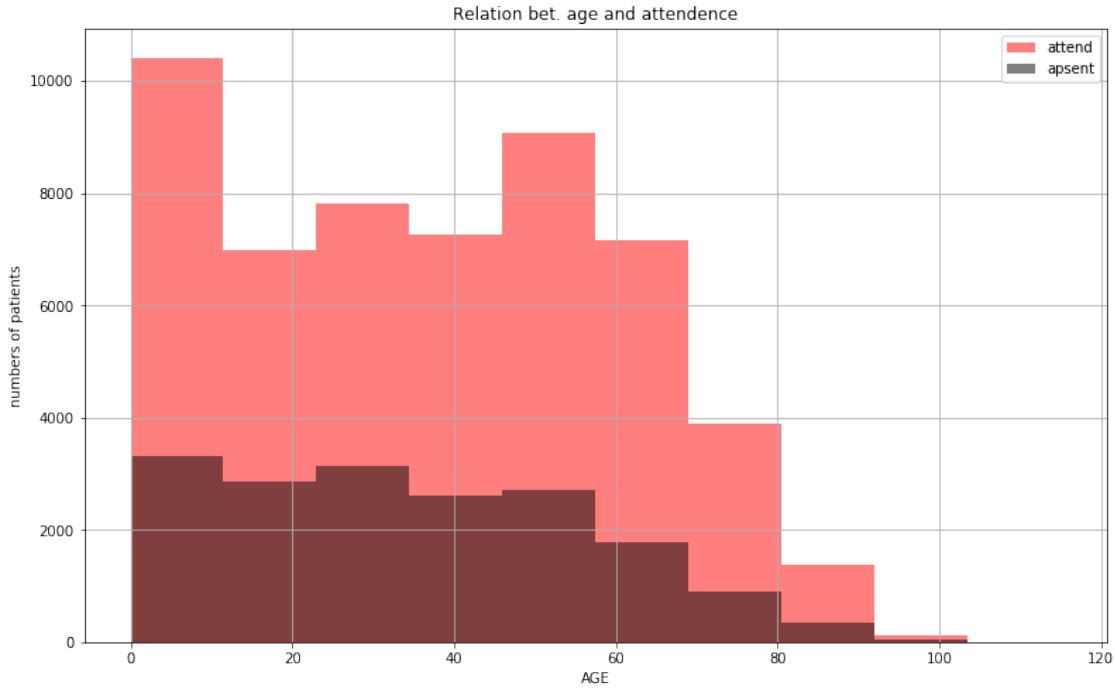
```
In [15]: noshow = df.No_show == 'Yes'
         show = df.No_show == 'No'


         df[show].count(),df[noshow].count()
```

```
Out[15]: (Gender           54153
          Age              54153
          Neighbourhood    54153
          Scholarship      54153
          Hypertension     54153
          Diabetes         54153
          Alcoholism       54153
          Handcap          54153
          SMS_received     54153
          No_show          54153
          dtype: int64, Gender           17663
          Age              17663
```

```
Neighbourhood     17663
Scholarship       17663
Hypertension      17663
Diabetes          17663
Alcoholism        17663
Handcap           17663
SMS_received      17663
No_show           17663
dtype: int64)
```

there is a 54153 patients that had attend
there is a 17663 patients that had not attend
which means patients that had attend is 3 times that hadnot

```
In [16]: def AGE_vs_attend(df,col_name,attend,apsent):
             plt.figure(figsize=[13,8])
             df[col_name][show].hist(alpha=.5,bins=10,color='red',label='attend')
             df[col_name][noshow].hist(alpha=.5,bins=10,color='black',label='apsent')
             plt.legend()
             plt.title('Relation bet. age and attendence')
             plt.xlabel('AGE')
             plt.ylabel('numbers of patients');
         AGE_vs_attend(df,'Age',show,noshow)
```



Age from(0-8) are most showing, which means that there is a lot of parents showing up
Age from(45-55) are the second showing patients
the least attending patients is at Age (65-85)

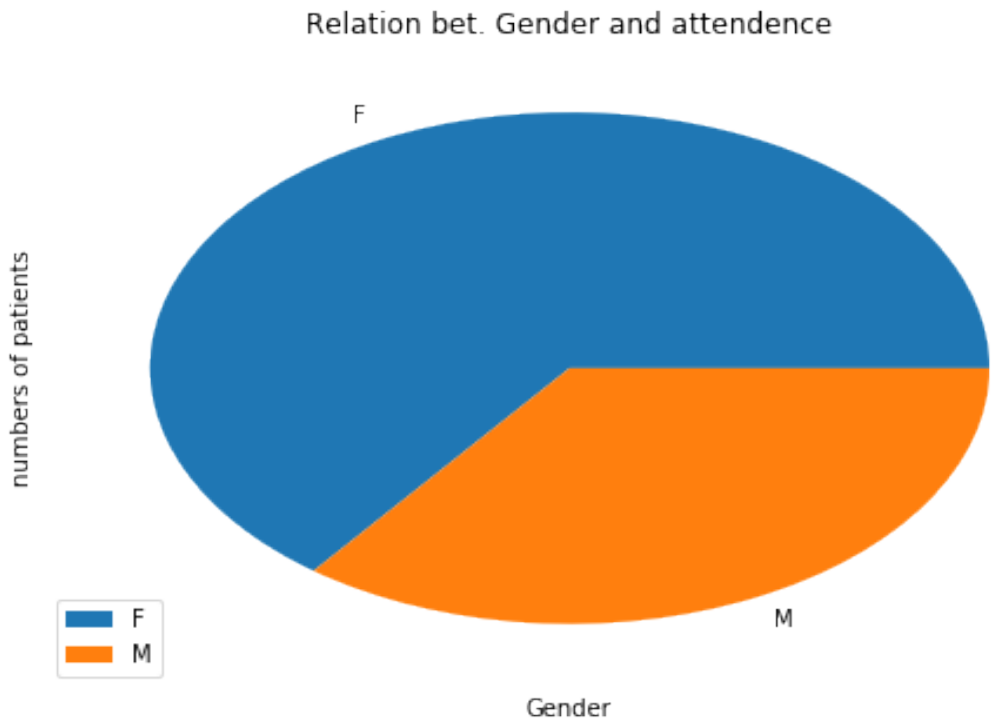
In [ ]:

7.0.1 Research Question 2 (Factors affecting attendence)

In [ ]:

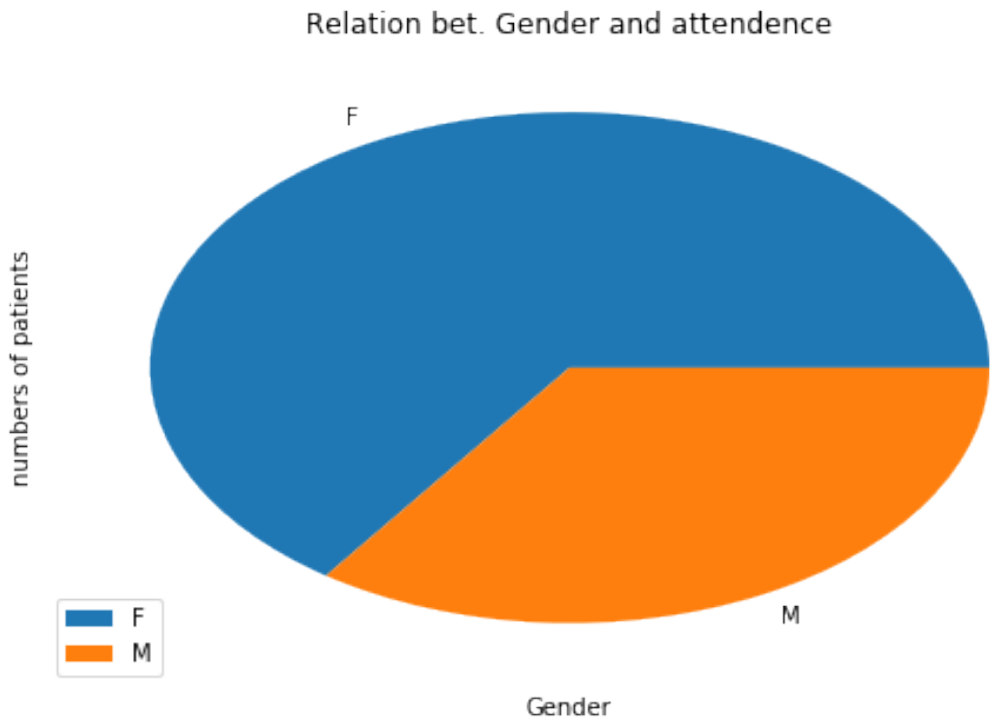
In [ ]:

```
In [20]: def Gender_vs_attend(df,col_name,attend,apsent):
             plt.figure(figsize=[8,5])
             df[col_name][show].value_counts(normalize=True).plot(kind='pie',label='show')
             plt.legend()
             plt.title('Relation bet. Gender and attendence')
             plt.xlabel('Gender')
             plt.ylabel('numbers of patients');

         Gender_vs_attend(df,'Gender',show,noshow)
```



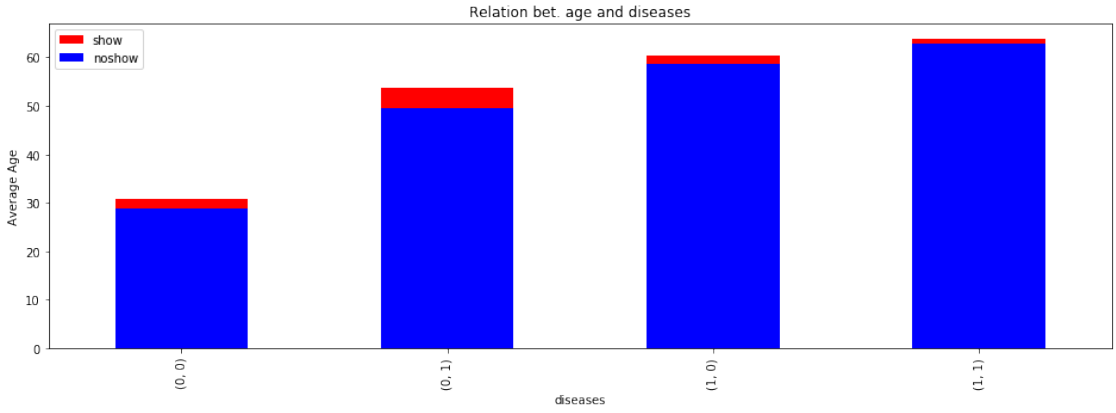

```
In [19]: def Gender_vs_attend(df,col_name,attend,apsent):
             plt.figure(figsize=[8,5])
             df[col_name][noshow].value_counts(normalize=True).plot(kind='pie',label='show')
             plt.legend()
             plt.title('Relation bet. Gender and attendence')
             plt.xlabel('Gender')
```

```
    plt.ylabel('numbers of patients');

Gender_vs_attend(df,'Gender',show,noshow)
```



there is no relation bet attendence and gender

```
In [21]: plt.figure(figsize=[16,5])
         df[show].groupby(['Hypertension','Diabetes']).mean()['Age'].plot(kind='bar',color='red'
         df[noshow].groupby(['Hypertension','Diabetes']).mean()['Age'].plot(kind='bar',color='bl
         plt.legend();
         plt.title("Relation bet. age and diseases")
         plt.xlabel('diseases')
         plt.ylabel('Average Age');
```



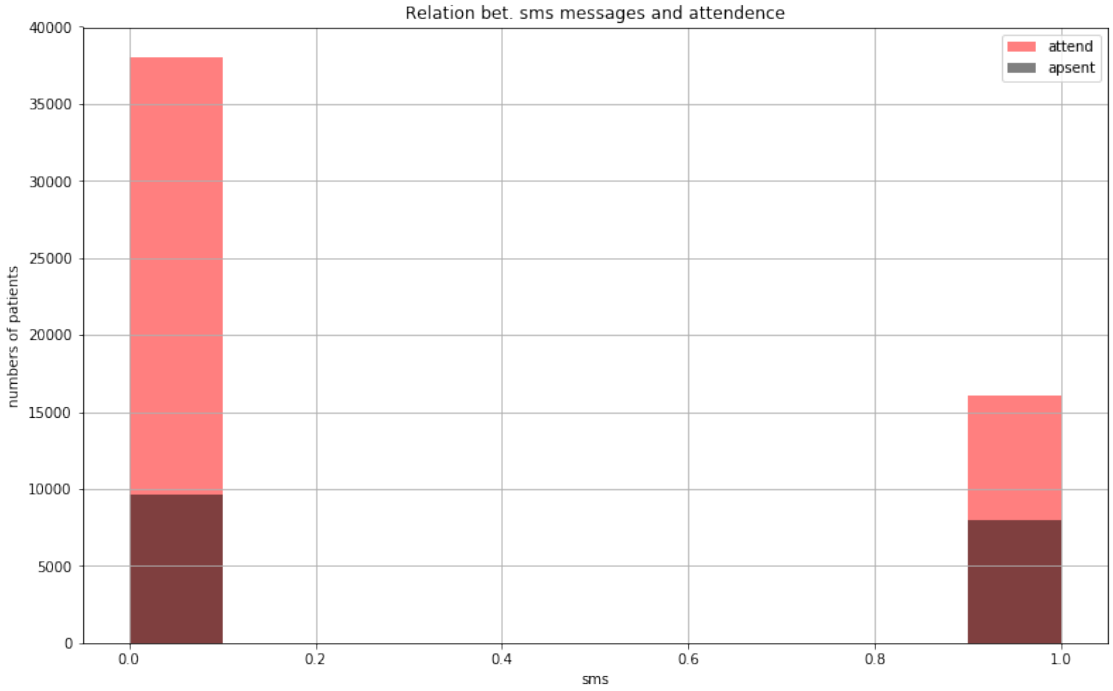
```
In [22]: df[noshow].groupby(["Hypertension","Diabetes"]).mean()['Age']

Out[22]: Hypertension  Diabetes
         0             0           28.768691
                       1           49.481172
         1             0           58.650380
                       1           62.913282
         Name: Age, dtype: float64
```

mean age of non choronic diseases that had not attend is 28 years
mean age of Hypertension patients that hadnot attend is 58 , and 49 of diabetes patients

```
In [23]: df[show].groupby(["Hypertension","Diabetes"]).mean()['Age']

Out[23]: Hypertension  Diabetes
         0             0           30.713360
                       1           53.701370
         1             0           60.270517
                       1           63.764303
         Name: Age, dtype: float64
```

mean age of non choronic diseases that had attend is 30 years
mean age of Hypertension patients that had attend is 60 , and 53 of diabetes patients
that means there is a relation bet age and choronic disease

```
In [34]: def Sms_vs_attend(df,col_name,attend,apsent):
             plt.figure(figsize=[13,8])
             df[col_name][show].hist(alpha=.5,bins=10,color='red',label='attend')
             df[col_name][noshow].hist(alpha=.5,bins=10,color='black',label='apsent')
             plt.legend()
             plt.title('Relation bet. sms messages and attendence')
             plt.xlabel('sms')
             plt.ylabel('numbers of patients');
         Sms_vs_attend(df,'SMS_received',show,noshow)
```



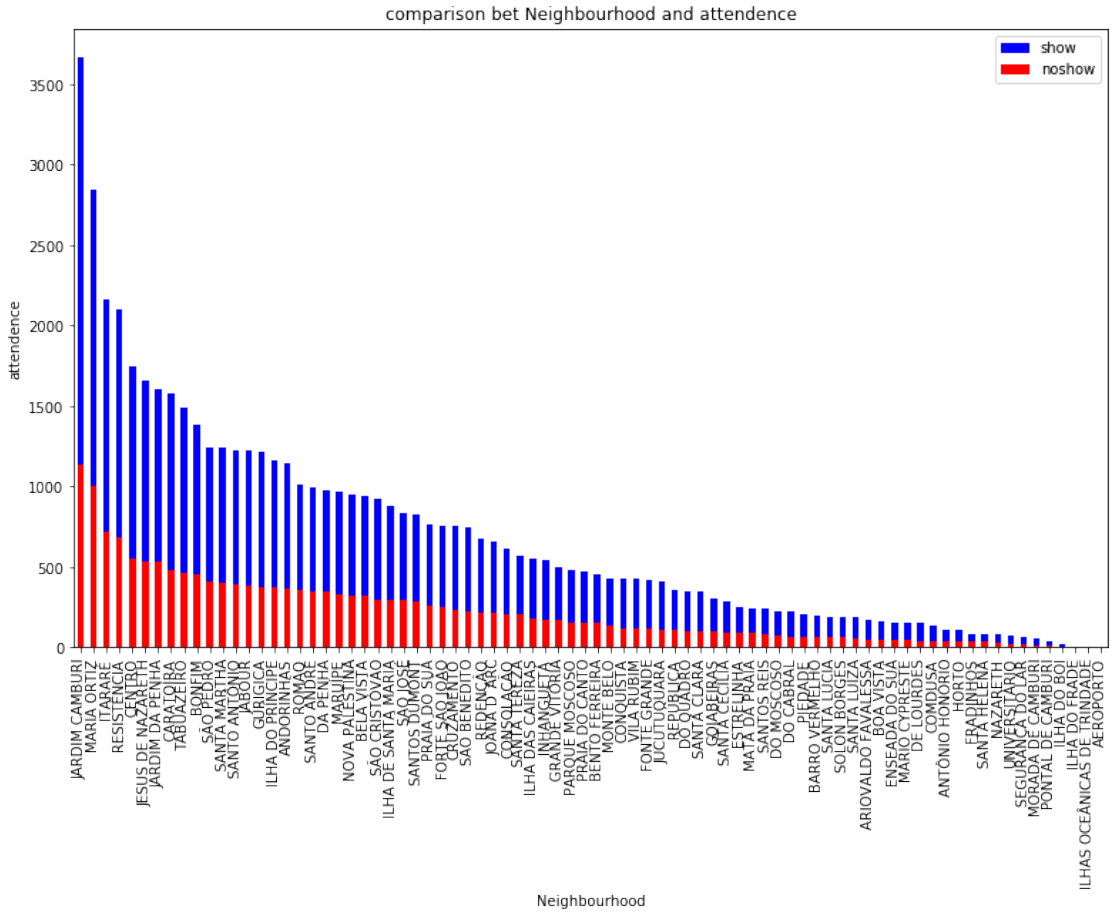
looks like there is alot of patients(double) that attend without receiving sms , so we need to change the way of writing sms

In [ ]:

In [ ]:

In [ ]:

In [35]:
```
plt.figure(figsize=[13,8])
df.Neighbourhood[show].value_counts().plot( kind= 'bar', color='blue', label='show')
df.Neighbourhood[noshow].value_counts().plot( kind= 'bar', color='red', label='noshow')
plt.legend()
plt.title("comparison bet Neighbourhood and attendence")
plt.xlabel("Neighbourhood")
plt.ylabel("attendence")
plt.show();
```



there is a huge effect bet Neighbourhood and attendence , jardim camburi has the greatest showing rate and appointments

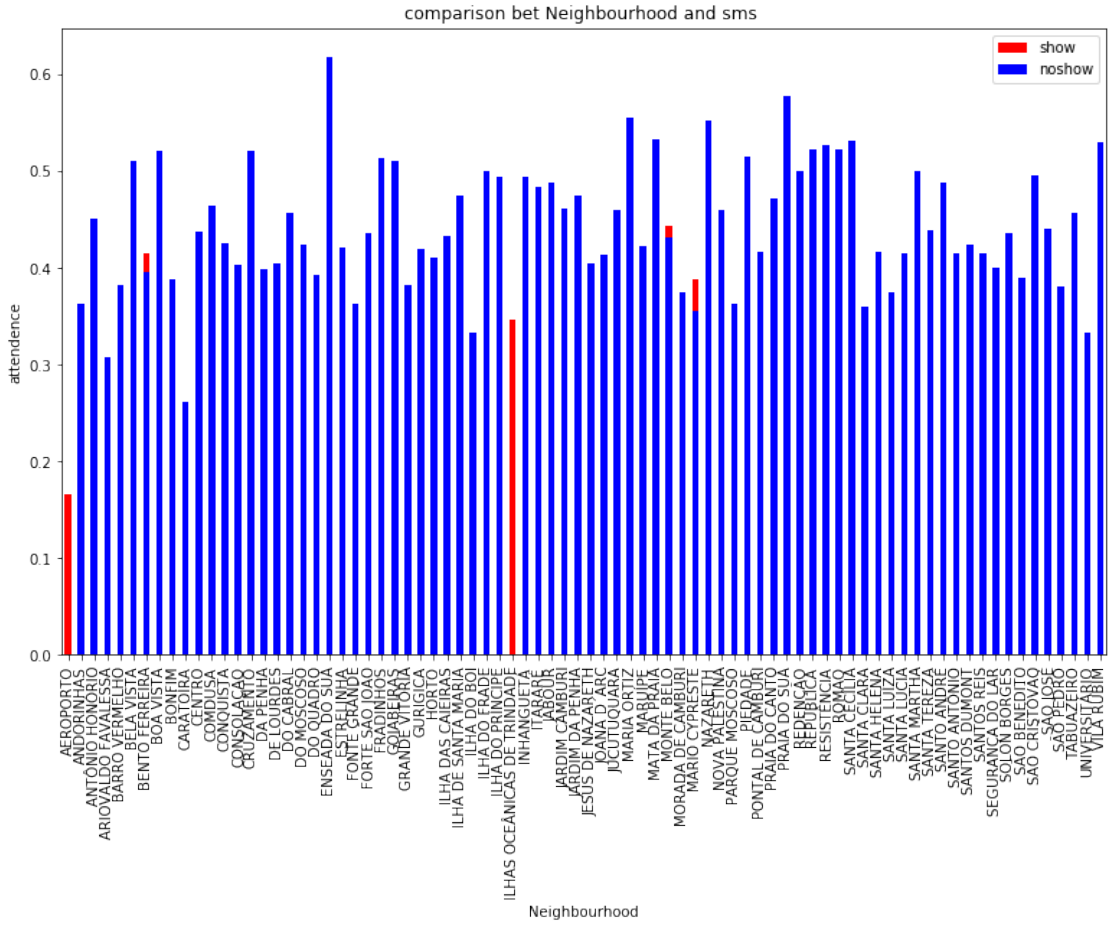
note: there is no a dataset that tells me the census of each city to get the percentage more acurate

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [36]: plt.figure(figsize=(13,8))
         df[show].groupby('Neighbourhood').SMS_received.mean().plot(kind= 'bar',color='red',labe
         df[noshow].groupby('Neighbourhood').SMS_received.mean().plot(kind= 'bar',color='blue',l
         plt.legend()
         plt.title("comparison bet Neighbourhood and sms ")
         plt.xlabel("Neighbourhood")
         plt.ylabel("attendence");
```



only 5 response to sms in Neighbourhoods ,ilhas oceanicas de trindade has the most rate of reponse

the 5 regions that shows response has the least attending rate

which means there is a huge issue with sms senders ,maybe they donot send to the other regions

In [ ]:

In [ ]:

In [ ]:

Conclusions

Tip: there is a huge relation bet. Neighbourhood and attendence, jardim camburi has the greatest showing rate and appointments note: there is no a dataset that tells me the census of each city to get the percentage more acurate

Tip: there is a huge issue with the sms senders technique, as most of patients had attend without receiving sms , the number of showing patients without receiving sms is double the number of patients that attend by the sms

Tip: Age has clear effect on attendence as the patients with (0-8) is the most showing ,and for sure with there parents. Age from(45-55) are the second showing patients. the least attending patients is at Age (65-85) so there is an inversaly propotional relation bet age and attendence.

Tip: to increase the amount of patients showing: fix the sms issue as patients are not giving attention to it give attention to regions or Neighbourhood
Limitations **Tip**: there is no relation bet. gender and attendence

```
In [25]: from subprocess import call
         call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])
```

```
Out[25]: 0
```

```
In [ ]:
```