Investigate_a_Dataset

December 17, 2021

1 Project: Investigate a Dataset - [No show appointments]

1.1 Table of Contents

Introduction
Data Wrangling
Exploratory Data Analysis
Conclusions
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. '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.

1.1.1 Dataset Description

We have a csv file that contains a data for the tobic that we mentioned we are going to analyze and answer questions about it.

1.1.2 Question(s) for Analysis

What factors are important for us to know in order to predict if a patient will show up for their scheduled appointment?

```
In [83]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import seaborn as snb
         % matplotlib inline
```

Data Wrangling
In this section, we will load in the data, to check for cleanlines, missing values any issues with the data and Processing it for analyzing.

1.1.3 General Properties

```
In [84]: # Load your data and print out a few lines. Perform operations to inspect data
         df=pd.read_csv('noshowappointments-kagglev2-may-2016.csv')
         df.head()
```

Out[84]:

	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

Here if we look at the word "Hipertension" the head of the 9th column we find that it wrote with "i" instead of "y". I will clean it in the data cleaning section. onther thing in the head of the last column "No_show" instead of "no-show".

```
In [85]: df.shape
```

```
Out[85]: (110527, 14)
```

The data consists of 110527 raws & 14 column

```
In [86]: df.duplicated().sum()
```

```
Out[86]: 0
```

There is no duplicated raws in the data. but maybe there are nonunique values.

```
In [87]: #check for non unique values
         df['PatientId'].nunique()
```

```
Out[87]: 62299
```

That's mean unique values are 62299 and 48228 are duplicated ID

```
In [88]: df.duplicated(['PatientId','No-show']).sum()
```

```
Out[88]: 38710

In [89]: df['PatientId'].duplicated().sum()

Out[89]: 48228
```

Thus, 38710 are patient IDs that have the same status of showing or no showing.

```
In [90]: 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
```

These infromations tell us that there are no missing values.

```
In [91]: df.describe()

Out[91]:           PatientId  AppointmentID            Age    Scholarship  \
         count  1.105270e+05   1.105270e+05  110527.000000  110527.000000
         mean   1.474963e+14   5.675305e+06      37.088874       0.098266
         std    2.560949e+14   7.129575e+04      23.110205       0.297675
         min    3.921784e+04   5.030230e+06      -1.000000       0.000000
         25%    4.172614e+12   5.640286e+06      18.000000       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

                 Hipertension       Diabetes     Alcoholism        Handcap  \
         count  110527.000000  110527.000000  110527.000000  110527.000000
         mean        0.197246       0.071865       0.030400       0.022248
         std         0.397921       0.258265       0.171686       0.161543
```

```
min          0.000000       0.000000       0.000000       0.000000
25%          0.000000       0.000000       0.000000       0.000000
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
```

Here if we look at the minimum raw we will find that the minimum age is (-1) and that's illogical. so I will process this mitake.

```
In [92]: mask=df.query('Age=="-1"')
         mask
```

```
Out[92]:           PatientId  AppointmentID Gender          ScheduledDay  \
         99832  4.659432e+14        5775010      F  2016-06-06T08:58:13Z

                    AppointmentDay  Age Neighbourhood  Scholarship  Hipertension  \
         99832  2016-06-06T00:00:00Z   -1         ROMÃO            0             0

                Diabetes  Alcoholism  Handcap  SMS_received No-show
         99832         0           0        0             0      No
```

1.1.4 Data Cleaning

In this section, we will clean our data from any error to be ready for analysis.

```
In [93]: #Correcting "Hipertension" to "Hypertension"
         df.rename(columns={'Hipertension':'Hypertension'},inplace=True)
         df.head()
```

```
Out[93]:       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  Hypertension  \
         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 [94]: #Removing duplicates
         df.drop_duplicates(['PatientId','No-show'],inplace=True)
         df.shape
```

```
Out[94]: (71817, 14)
```

df.duplicated(['PatientId','No_show']).sum()

2 Removing the value of minimum age which has a negative value(-1)

df.drop(index=99832,inplace=True)

```
In [14]: df.describe()
```

```
Out[14]:           PatientId  AppointmentID           Age   Scholarship  Hypertension  \
         count  7.181700e+04   7.181700e+04  71817.000000  71817.000000  71817.000000
         mean   1.466294e+14   5.666495e+06     36.526978      0.095534      0.195065
         std    2.544927e+14   7.313144e+04     23.378518      0.293954      0.396254
         min    3.921784e+04   5.030230e+06     -1.000000      0.000000      0.000000
         25%    4.175978e+12   5.631622e+06     17.000000      0.000000      0.000000
         50%    3.189717e+13   5.672884e+06     36.000000      0.000000      0.000000
         75%    9.457487e+13   5.716568e+06     55.000000      0.000000      0.000000
         max    9.999816e+14   5.790484e+06    115.000000      1.000000      1.000000

                    Diabetes    Alcoholism       Handcap  SMS_received
         count  71817.000000  71817.000000  71817.000000  71817.000000
         mean       0.070958      0.025036      0.020135      0.335561
         std        0.256757      0.156235      0.155337      0.472190
         min        0.000000      0.000000      0.000000      0.000000
         25%        0.000000      0.000000      0.000000      0.000000
         50%        0.000000      0.000000      0.000000      0.000000
         75%        0.000000      0.000000      0.000000      1.000000
         max        1.000000      1.000000      4.000000      1.000000
```

The negative value in minimum Age was deleted.

```
In [15]: #Removing unnecessary data
         df.drop(['PatientId','ScheduledDay','AppointmentID','AppointmentDay'],axis=1,inplace=Tr
```

```
In [16]: df.head()
```

```
Out[16]:   Gender  Age      Neighbourhood  Scholarship  Hypertension  Diabetes  \
          0      F   62    JARDIM DA PENHA            0             1         0
          1      M   56    JARDIM DA PENHA            0             0         0
          2      F   62      MATA DA PRAIA            0             0         0
          3      F    8  PONTAL DE CAMBURI            0             0         0
          4      F   56    JARDIM DA PENHA            0             1         1

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

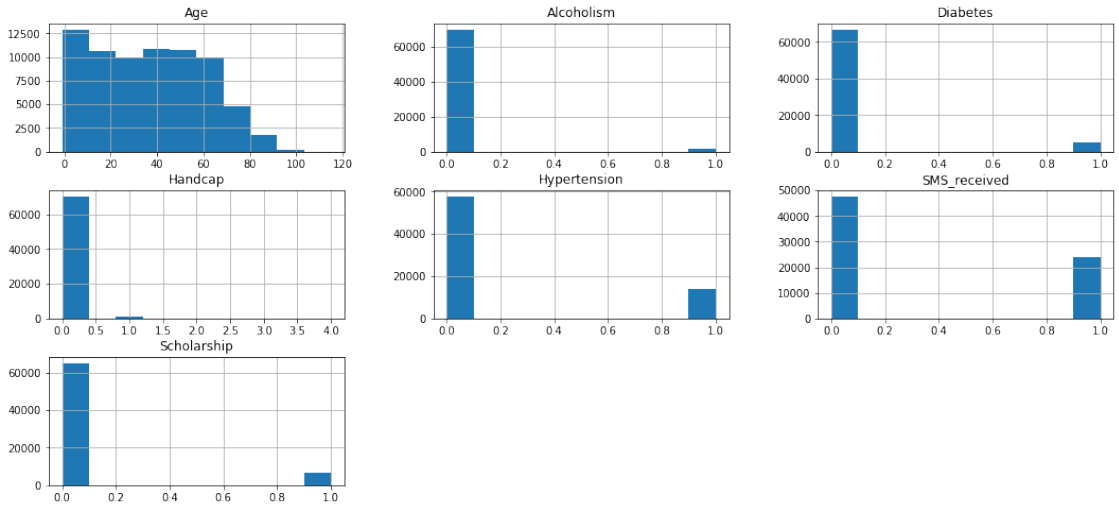
2.1 summary

After we gathering and assessing our data we cleaned it from evry error we started with correcting the weong words,removing the ngative value in the age section, also, removing duplicates in our data and lastly we removed unneccesary data now our data is cleaned and have no errors or duplications so we are ready to move on to exploration.

Exploratory Data Analysis Now that we've trimmed and cleaned our data, we're ready to move on to exploration. we are going to Compute statistics and create visualizations with the goal of addressing the research questions that we posed in the Introduction section.

2.1.1 Insights

```
In [17]: # Use this, and more code cells, to explore your data.
         df.hist(figsize=(18,8));
```



2.1.2 Research Question 2 (Replace this header name!)

```
In [32]: #Now we will divid the data into two groups to "show" or "not show".
         show=df.No_show=='No'
         noshow=df.No_show=='Yes'

In [33]: df[show].count()

Out[33]: Gender           54154
         Age              54154
         Neighbourhood    54154
         Scholarship      54154
         Hypertension     54154
         Diabetes         54154
         Alcoholism       54154
         Handcap          54154
         SMS_received     54154
         No_show          54154
         dtype: int64

In [34]: df[noshow].count()

Out[34]: Gender           17663
         Age              17663
         Neighbourhood    17663
         Scholarship      17663
         Hypertension     17663
         Diabetes         17663
         Alcoholism       17663
         Handcap          17663
         SMS_received     17663
         No_show          17663
         dtype: int64
```

Thus, Number of patient showed is greater than patient that no showed by 3 times.

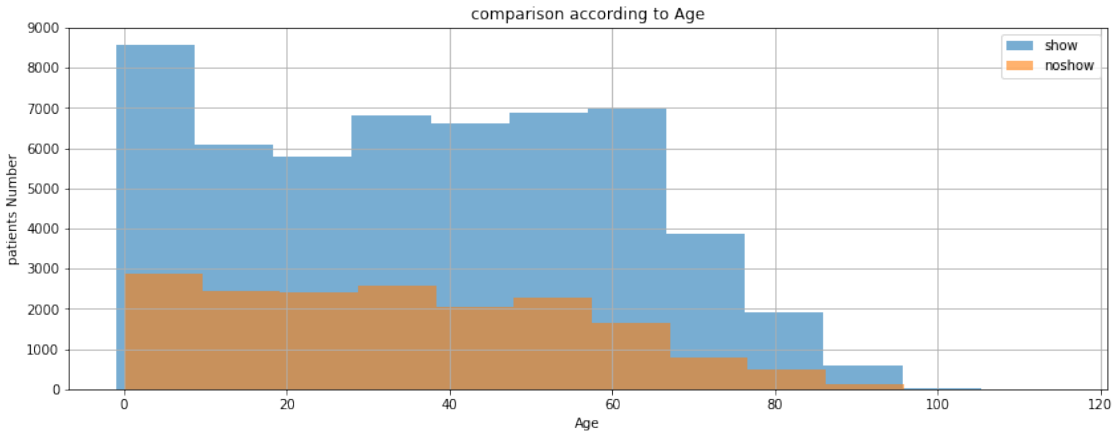
```
In [36]: #We will get the mean of each group
         df[show].mean(),df[noshow].mean()

Out[36]: (Age             37.228460
          Scholarship      0.091332
          Hypertension     0.202940
          Diabetes         0.072866
          Alcoholism       0.023599
          Handcap          0.020903
          SMS_received     0.297226
          dtype: float64, Age             34.376267
          Scholarship      0.108419
          Hypertension     0.170922
```

```
Diabetes          0.065108
Alcoholism        0.029440
Handcap           0.017777
SMS_received      0.453094
dtype: float64)
```

Notice here patients in [show] group recieved sms less than patients in [no show] group so we should reconsidering this issue with the marketing department.

2.2 Analyzing the factors

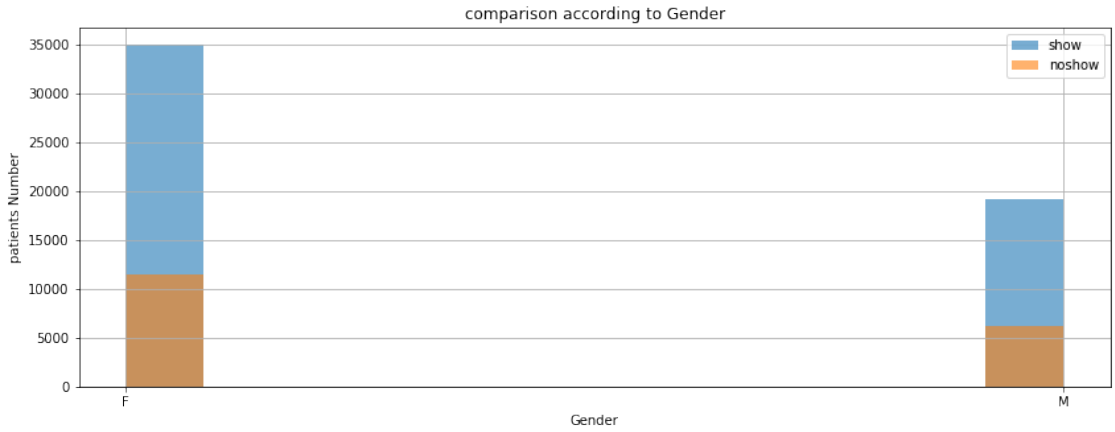
```
In [70]: #Analyzing the "Age" factor
         def attendance(df,Age,attended,Notattended):

             plt.figure(figsize=[14,5])
             df.Age[show].hist(alpha=.6,bins=12,label='show')
             df.Age[noshow].hist(alpha=.6,bins=12,label='noshow')
             plt.legend();
             plt.title('comparison according to Age')
             plt.xlabel('Age')
             plt.ylabel('patients Number');
         attendance(df,'Age',show,noshow)
```



The highest rate is between 0-10, the lowst is between 80-100 There is a negative relationship between the age and the attendence.

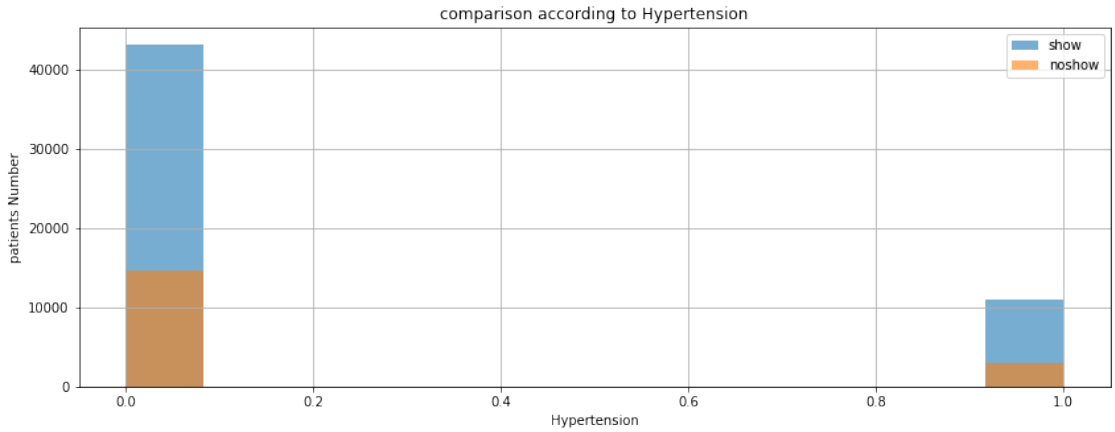
```
In [73]: #Analyzing the "Gender" factor
         def attendance(df,Gender,attended,Notattended):

             plt.figure(figsize=[14,5])
             df.Gender[show].hist(alpha=.6,bins=12,label='show')
             df.Gender[noshow].hist(alpha=.6,bins=12,label='noshow')
             plt.legend();
```

```
        plt.title('comparison according to Gender')
        plt.xlabel('Gender')
        plt.ylabel('patients Number');
    attendance(df,'Gender',show,noshow)
```



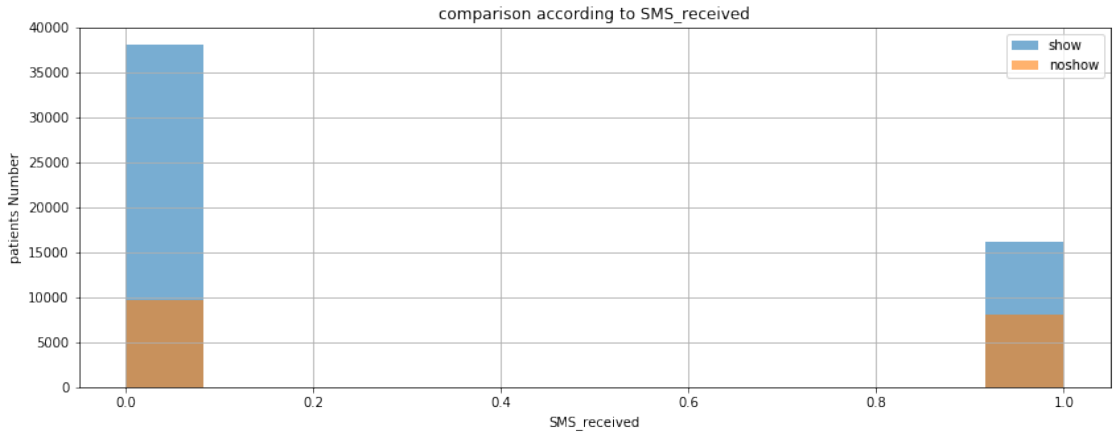
This figure illustrates that females have higher rate than males. but it not a significant factor for our analyze

```
In [74]: #Analyzing the "Hypertension" factor
         def attendance(df,Hypertension,attended,Notattended):

             plt.figure(figsize=[14,5])
             df.Hypertension[show].hist(alpha=.6,bins=12,label='show')
             df.Hypertension[noshow].hist(alpha=.6,bins=12,label='noshow')
             plt.legend();
             plt.title('comparison according to Hypertension')
             plt.xlabel('Hypertension')
             plt.ylabel('patients Number');
         attendance(df,'Hypertension',show,noshow)
```



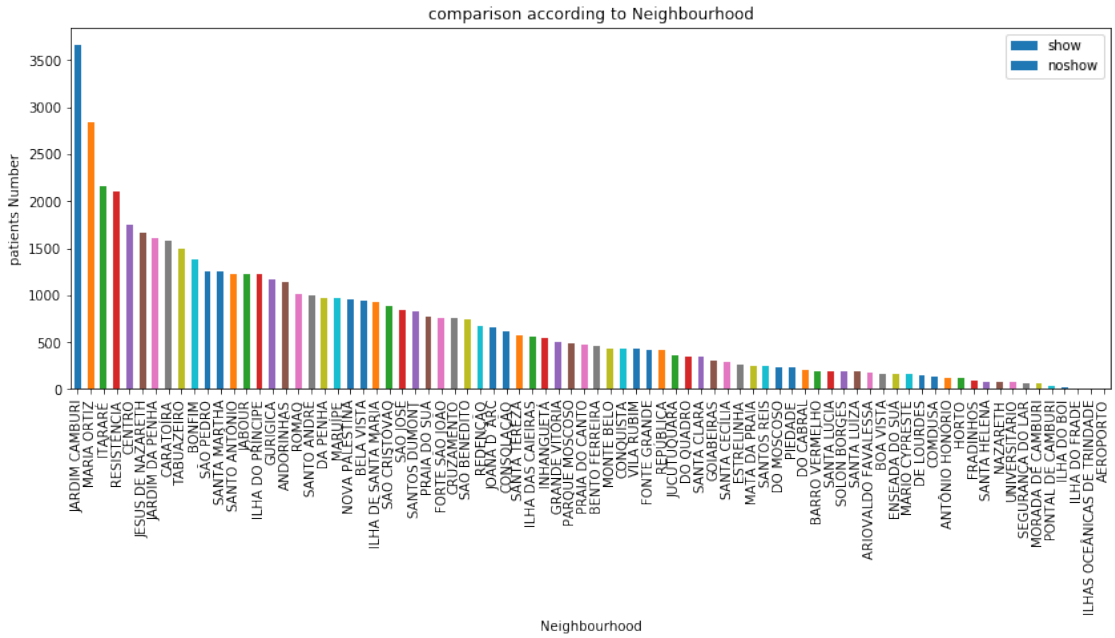
Hypertension is insignificant

```
In [75]: #Analyzing the "Hypertension" factor
         def attendance(df,SMS_received,attended,Notattended):

             plt.figure(figsize=[14,5])
             df.SMS_received[show].hist(alpha=.6,bins=12,label='show')
             df.SMS_received[noshow].hist(alpha=.6,bins=12,label='noshow')
             plt.legend();
             plt.title('comparison according to SMS_received')
             plt.xlabel('SMS_received')
             plt.ylabel('patients Number');
         attendance(df,'SMS_received',show,noshow)
```



patients in "show" group recieved sms less than patients in "no show" group so we should reconsidering this issue with the marketing department.

```
In [78]: #Analyzing the "Neighbourhood" factor
         def attendance(df,Neighbourhood,attended,Notattended):

             plt.figure(figsize=[14,5])
             df.Neighbourhood[show].value_counts().plot(kind='bar',label='show')
             df.Neighbourhood[noshow].value_counts().plot(kind='bar',label='noshow')
             plt.legend();
             plt.title('comparison according to Neighbourhood')
             plt.xlabel('Neighbourhood')
             plt.ylabel('patients Number');
         attendance(df,'Neighbourhood',show,noshow)
```



In [ ]: Here we can see that the neighbourhood or the district has a positive relationship with

Conclusions

After we analyze our data and make statisticals we can now answer the report question which is: What factors are important for us to know in order to predict if a patient will show up for their scheduled appointment? We have a lot of factors like: Age, Hypertension, Diabetes, SMS_received, Neighbourhood, Gender, etc... some of these factors are insignificant and other are. we can say that:

1.Age factor has a negative relationship with the attendence, the older the patient, the lower the attendance rate, we noticed that the highest rate lies down between 0 to 10 years.

2.Neighbourhood factor has a significant influence, patients in JARDIM CAMBURI is the most attendees and other towns has lower rates, the far the town is, the fewer attendees will come.

3.Also patients that attended recieved sms less than patients that don't attend, so we should reconsidering this issue with the customer service.but it means that sms is insignificant.

2.2.1 Limitations

we couldn't build our investigation of no show based on factors like gender, chronic diseases, enrollment in the welfare program.

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