- introduction

i used this dataset that collects information from 100k medical appointments in Brazil

- Question

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

In [31]:

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

- Data Wrangling

we will load the data and check for cleanliness.

In [32]:

```

#we will load the data and check for cleanliness
raw_df = pd.read_csv(r'C:\Users\Wello\Downloads\KaggleV2-May-2016.csv')
raw_df.head()
```

Out[32]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Sc
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	

In [33]:

```
df = raw_df.copy()
df.shape
```

Out[33]:

```
(110527, 14)
```

In [34]:

```
#No of missing value
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
```

Fortunately, there is no missing values in dataset.

In [35]:

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

Out[35]:

```
0
```

In [36]:

```
df['PatientId'].nunique()
```

Out[36]:

```
62299
```

In [37]:

```
df['PatientId'].duplicated().sum()
```

Out[37]:

```
48228
```

There is 48228 duplicated PatientId

In [38]:
```
df.describe()
```

Out[38]:

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

- Data Cleaning

There is -1 years old in an age column, which is impossible,so we will dorp it.

No-show column is confusing. Inorder to make it more clear, we can change the column name to 'Show', and change the object in this column. Some mistake in label name such as'Handcap' and 'Hipertension' should also be corrected.

If two rows has absolutely same information regardless of AppointmentID, we can treat them as duplicated information. The data types of scheduled day and appointment day are str, which need to be transfered to datatime, in order to be analyzed easily.

In order to make in-depth analysis of relationship between appointment time and presence, some new columns show be added.

In [39]:
```
df.query('Age==-1')
```

Out[39]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhooc
99832	4.659432e+14	5775010	F	2016-06-06T08:58:13Z	2016-06-06T00:00:00Z	-1	ROMÃC

In [40]:

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

Out[40]:

	PatientId	AppointmentID	Age	Scholarship	Hipertension	Diabetes
count	1.105260e+05	1.105260e+05	110526.000000	110526.000000	110526.000000	110526.000000
mean	1.474934e+14	5.675304e+06	37.089219	0.098266	0.197248	0.071865
std	2.560943e+14	7.129544e+04	23.110026	0.297676	0.397923	0.258266
min	3.921784e+04	5.030230e+06	0.000000	0.000000	0.000000	0.000000
25%	4.172536e+12	5.640285e+06	18.000000	0.000000	0.000000	0.000000
50%	3.173184e+13	5.680572e+06	37.000000	0.000000	0.000000	0.000000
75%	9.438963e+13	5.725523e+06	55.000000	0.000000	0.000000	0.000000
max	9.999816e+14	5.790484e+06	115.000000	1.000000	1.000000	1.000000

In [41]:

```
df.rename(columns={'Hipertension': 'Hypertension','Handcap': 'Handicap','No-show'

df.head()
```

Out[41]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Sc
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	

In [42]:
```
df['show']=df['show'].replace({'No':1,'Yes':0 })
df['show']=df['show'].astype('int')
df.head()
```

Out[42]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Sc
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	

In [43]:
```
#check number of PatientId and show duplicated
df.duplicated(['PatientId','show']).sum()
```

Out[43]:
```
38710
```

In [44]:
```
df.drop_duplicates(['PatientId','show'],inplace=True)
df.shape
```

Out[44]:
```
(71816, 14)
```

In [45]:
```
#remove un importatnt data
df.drop(['PatientId','AppointmentID','ScheduledDay','AppointmentDay'],axis= 1,inp
df.head()
```

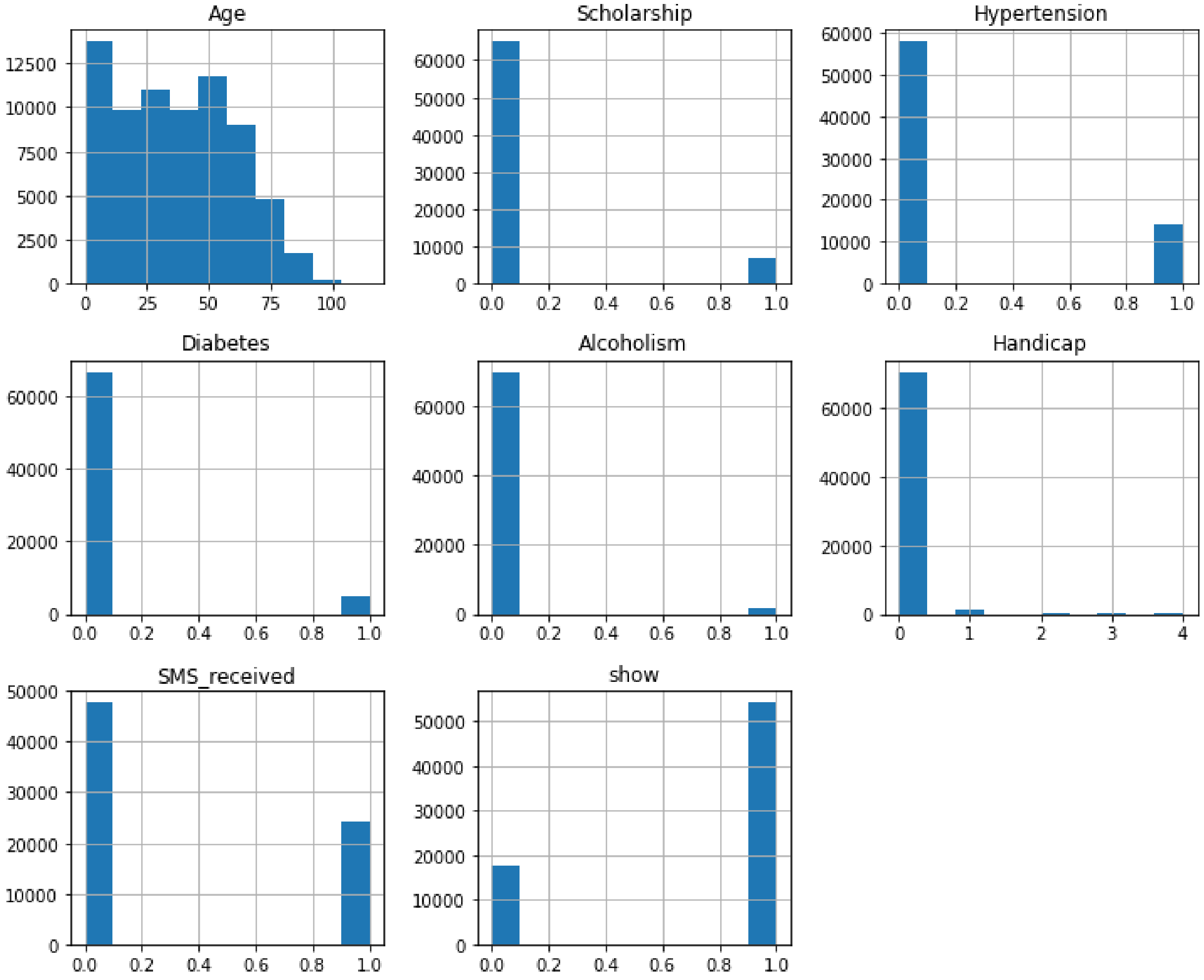
Out[45]:

	Gender	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handicap	SMS
0	F	62	JARDIM DA PENHA	0	1	0	0	0	
1	M	56	JARDIM DA PENHA	0	0	0	0	0	
2	F	62	MATA DA PRAIA	0	0	0	0	0	
3	F	8	PONTAL DE CAMBURI	0	0	0	0	0	
4	F	56	JARDIM DA PENHA	0	1	1	0	0	

- Exploratory Data Analysis

Now we ready to move on exploration

```
In [46]: df.hist(figsize=(12,10));
```



```
In [47]: show_data = df.show == 1
         noshow_data = df.show == 0
         df[show_data].count(), df[noshow_data].count()
```

```
Out[47]: (Gender            54153
          Age               54153
          Neighbourhood     54153
          Scholarship       54153
          Hypertension      54153
          Diabetes          54153
          Alcoholism        54153
          Handicap          54153
          SMS_received      54153
          show              54153
          dtype: int64,
          Gender            17663
          Age               17663
          Neighbourhood     17663
          Scholarship       17663
          Hypertension      17663
          Diabetes          17663
          Alcoholism        17663
          Handicap          17663
          SMS_received      17663
          show              17663
          dtype: int64)
```

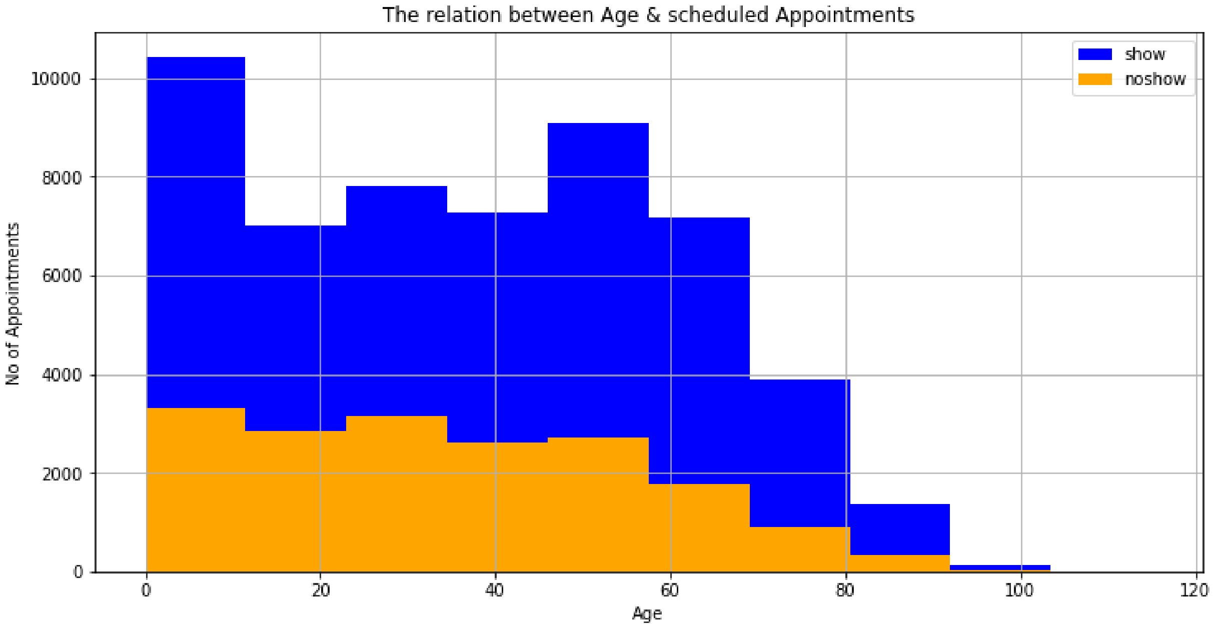
Number of showed patient(54153) is 3 times more than Non showed (17663)

```
In [48]: df[show_data].mean(),df[noshow_data].mean()
```

```
Out[48]: (Age             37.229166
          Scholarship      0.091334
          Hypertension     0.202944
          Diabetes         0.072868
          Alcoholism       0.023600
          Handicap         0.020904
          SMS_received     0.297232
          show             1.000000
          dtype: float64,
          Age             34.376267
          Scholarship      0.108419
          Hypertension     0.170922
          Diabetes         0.065108
          Alcoholism       0.029440
          Handicap         0.017777
          SMS_received     0.453094
          show             0.000000
          dtype: float64)
```

showed patient recieved sms less than unshowed ones that means we have to check sms campaign

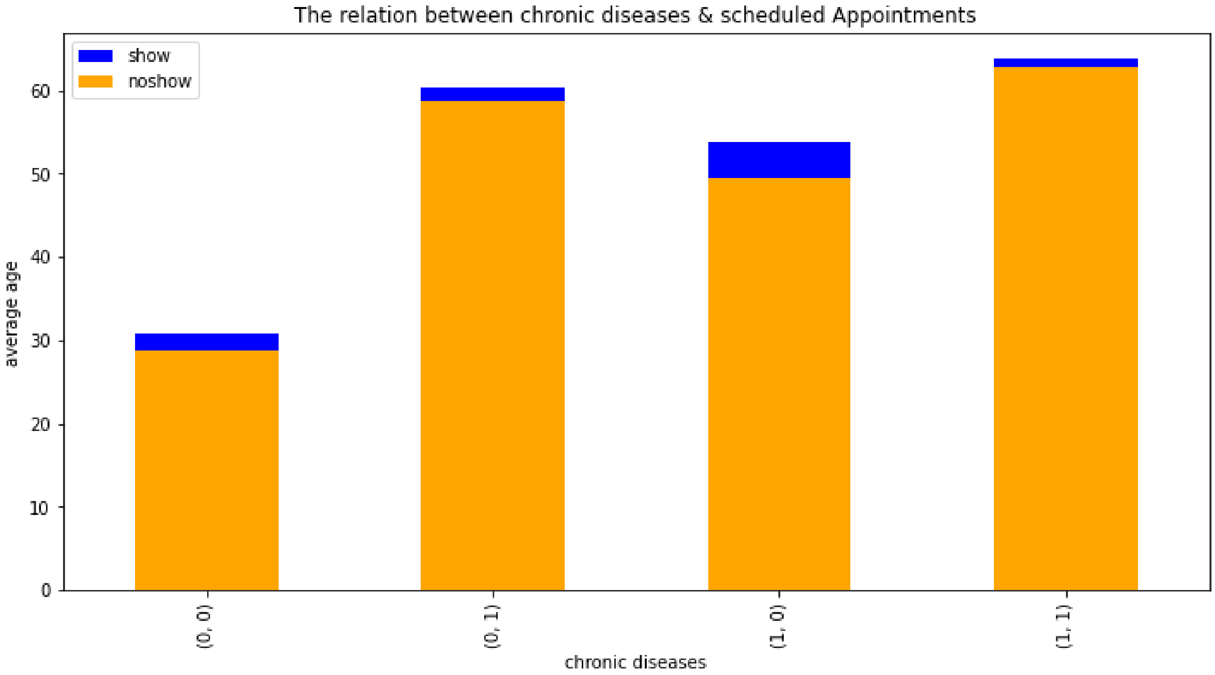
```
In [49]: # does Age affect the scheduled appointment?
         def  appointment(df,col_name,attendance,absence):
             plt.figure(figsize=[12,6])
             df[col_name][show_data].hist(color= 'blue',label='show')
             df[col_name][noshow_data].hist(color='orange',label='noshow')
             plt.legend();
             plt.title('The relation between Age & scheduled Appointments');
             plt.xlabel('Age');
             plt.ylabel('No of Appointments');
         appointment(df,'Age',show_data,noshow_data)
```



Ages from 0 ~ 9 are the most attending that means parents take care of their kids then Ages from 45 ~ 55 and people > 65 years old are the least attending.

In [50]:

```
# does the Age and chronic diseases affect the scheduled appointment?

plt.figure(figsize=[12,6])
df[show_data].groupby(['Diabetes','Hypertension']).mean()['Age'].plot(kind= 'bar'
df[noshow_data].groupby(['Diabetes','Hypertension']).mean()['Age'].plot(kind= 'ba
plt.title('The relation between chronic diseases & scheduled Appointments')
plt.legend();
plt.xlabel('chronic diseases')
plt.ylabel('average age');
```

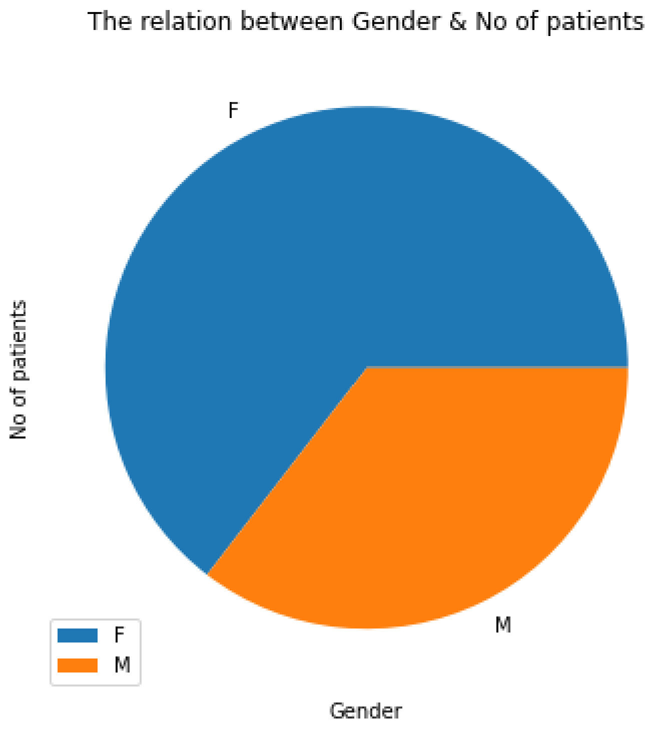


In [51]:
```
df[show_data].groupby(['Diabetes','Hypertension']).mean()['Age'],df[noshow_data].
```

Out[51]:
```
(Diabetes  Hypertension
 0         0               30.713360
           1               60.270517
 1         0               53.701370
           1               63.764303
 Name: Age, dtype: float64,
 Diabetes  Hypertension
 0         0               28.768691
           1               58.650380
 1         0               49.481172
           1               62.913282
 Name: Age, dtype: float64)
```

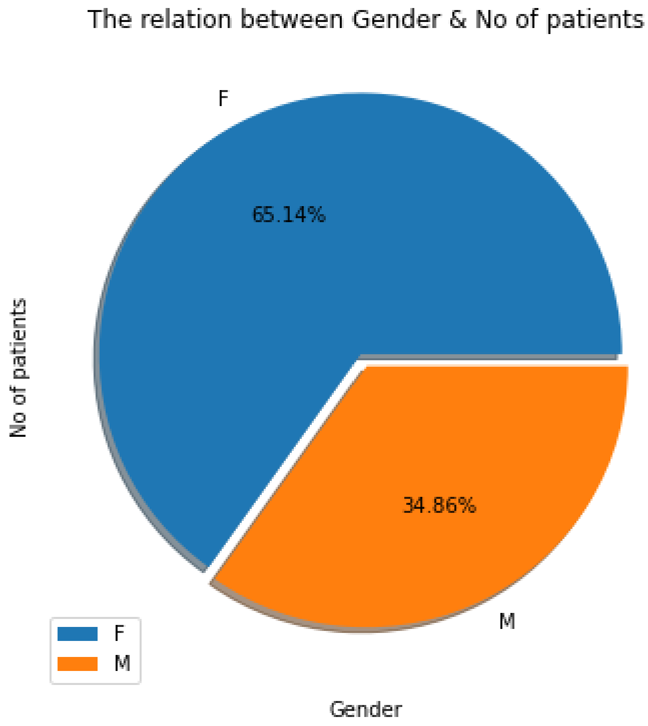
There is no correlation between mean Age of chronic diseases & scheduled Appointments

In [64]:
```
# Percentage of people who show-up ?

def  appointment(df,col_name,attendance,absence):
    plt.figure(figsize=[12,6])
    df[col_name][show_data].value_counts(normalize=True).plot(kind='pie',label='s
    plt.legend();
    plt.title('The relation between Gender & No of patients');
    plt.xlabel('Gender');
    plt.ylabel('No of patients');
appointment(df,'Gender',show_data,noshow_data)
```



In [65]:

```
# Percentage of people who show-up ?

def  appointment(df,col_name,attendance,absence):
    plt.figure(figsize=[12,6])
    df[col_name][noshow_data].value_counts(normalize=True).plot(kind='pie',label=
    plt.legend();
    plt.title('The relation between Gender & No of patients');
    plt.xlabel('Gender');
    plt.ylabel('No of patients');
appointment(df,'Gender',show_data,noshow_data)
```


The relation between Gender & No of patients

there is no correlation between the Gender & scheduled Appointments

In [66]:

```
df[show_data].groupby(['Gender']).mean()['Age'],df[noshow_data].groupby(['Gender'
```

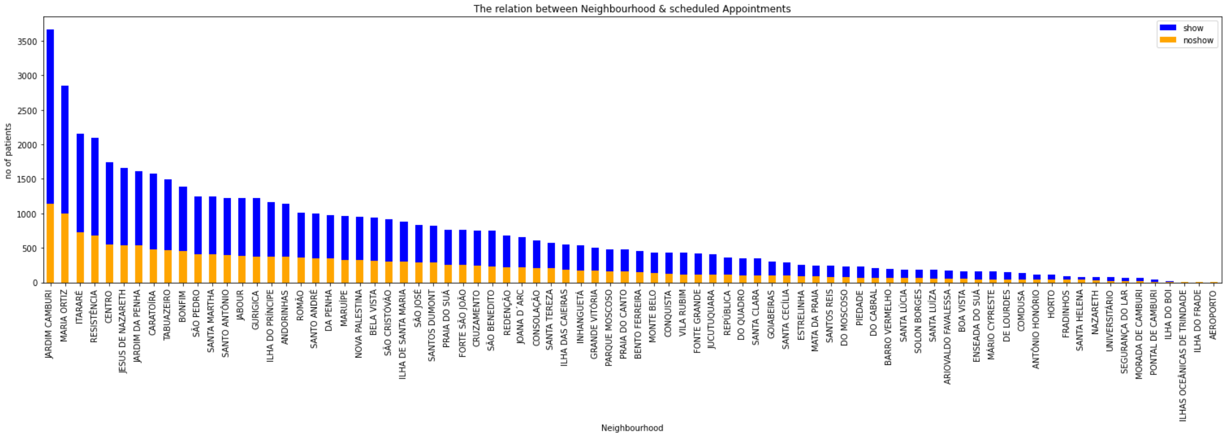
Out[66]:

```
(Gender
 F    39.130292
 M    33.766269
 Name: Age, dtype: float64,
 Gender
 F    36.06501
 M    31.22040
 Name: Age, dtype: float64)
```

there is no correlation between the average Age of Gender & scheduled Appointments

In [67]:

```
# does the Neighbourhood affects the Neighbourhood
plt.figure(figsize=[26,6])
df[show_data]['Neighbourhood'].value_counts().plot(kind= 'bar',color= 'blue', lab
df[noshow_data]['Neighbourhood'].value_counts().plot(kind= 'bar',color= 'orange',
plt.title('The relation between Neighbourhood & scheduled Appointments')
plt.legend();
plt.xlabel('Neighbourhood')
plt.ylabel('no of patients');
```



there is a correlation between the Neighbourhood & scheduled Appointments

- Conclusions

After making some questions on the dataset , we figured out some information about the patients and their behavior through questions as in the last question we found out that JARDIM CAMBURI seems to has most attending patients , in first question we found out Ages from 0 ~ 9 are the most attending that means parents take care of their kids then Ages from 45 ~ 55 and people > 65 years old are the least attending. for a nother question There is no correlation between mean Age of chronic diseases & scheduled Appointments.

- Limitations

There is no a clear correlation between Age & Choronic diseases and showing appointment

Type *Markdown* and LaTeX: α^2

In [ ]:

```
```