Project: No Show Medical Appointment Analysis

Conor Parke

Table of Contents

Contents

Introduction:

This data analysis explores a file containing data for 100,000 medical appointments in Brazil. Multiple variables are contained within the dataset, of which most could have been potentially used to explore. From this, 3 key independent variables and one dependent variable have been selected for analysis. The three independent variables were Age, Gender & Neighborhood. The dependent variable chosen being whether a patient turns up for their medical appointment or not. These three independent variables were ultimately selected because they would be insightful for determining whether or not patients show up at their medical appointments. This investigation from the data provided helps increase our perception of whether patients will appear for their medical appointments in the future in the areas provided.

This project will explore the following 4 questions:

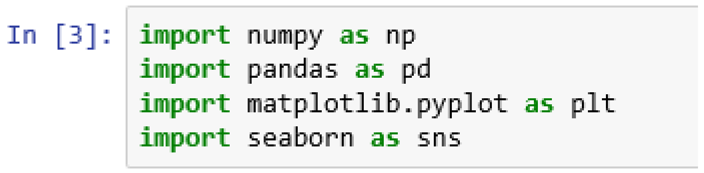
1. Can knowing a patient's gender help us to determine whether they will show up for their scheduled appointment?

2. Can knowing a patient's age help us to determine whether they will show up for their scheduled appointment?

3. Can knowing the neighborhood that a patient resides help us to determine whether they will show up for their scheduled appointment?

4. Can knowing the patient’s illness help us determine whether they will show up for their scheduled appointment?

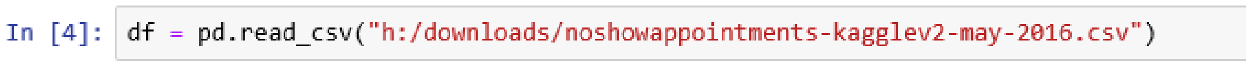
The following libraries have been imported to help us carry out this analysis:



```
In [3]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
```

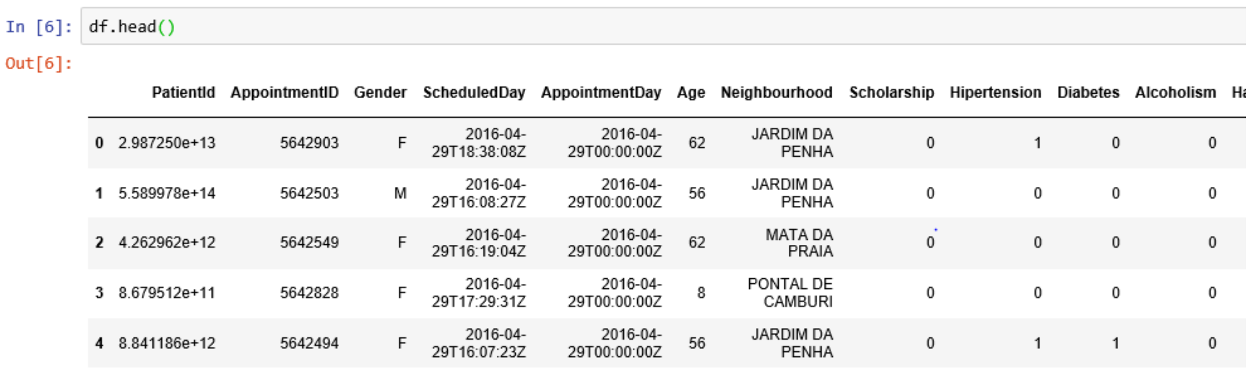
Data Wrangling:

In this section, data has been sorted in order to gain insights into necessary data cleaning, and then trimmed and cleaned to ensure accurate analysis. General Properties Below, the medical no show appointments data file is loaded my local h: \ drive using the Pandas csv reader.



```
In [4]: df = pd.read_csv("h:/downloads/noshowappointments-kagglev2-may-2016.csv")
```

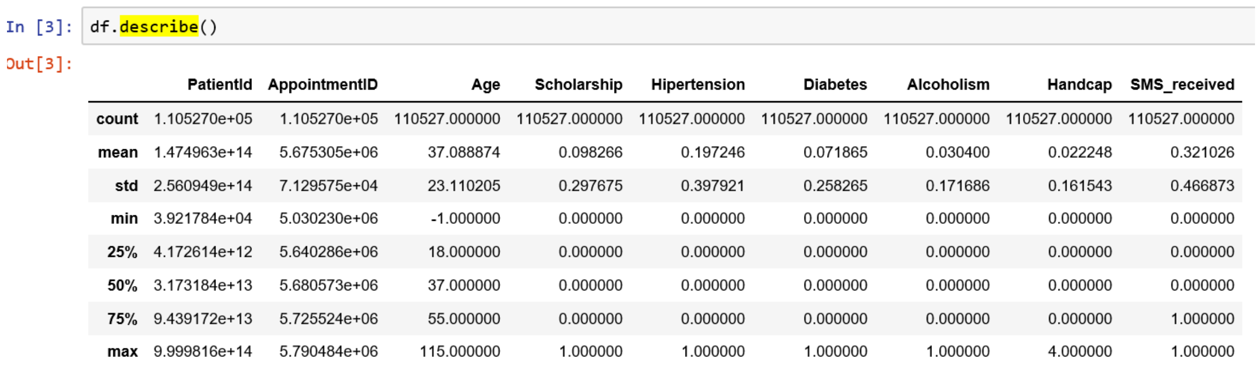
The first 5 rows of the data file are displayed using the df.head() function.



```
In [6]: df.head()
Out[6]:
```

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

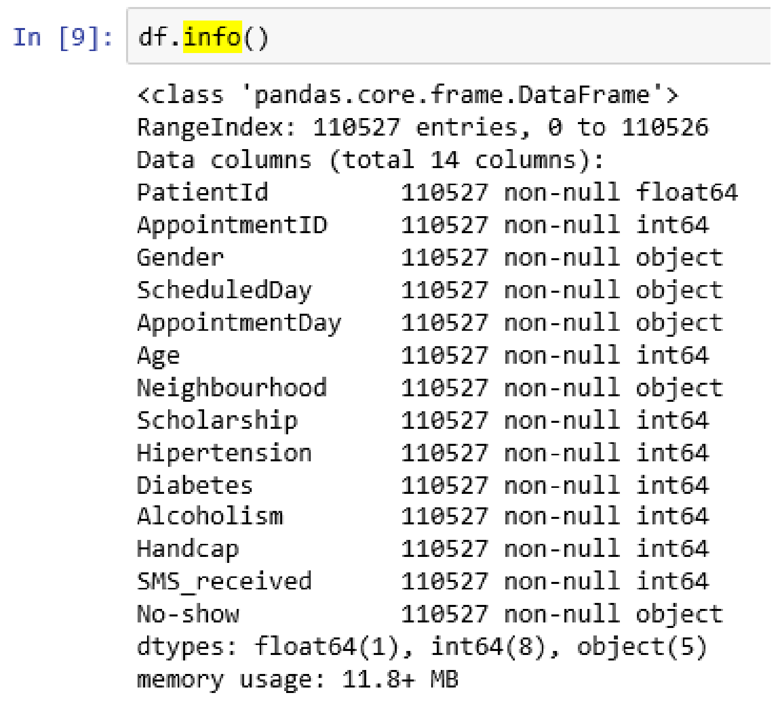
The describe function gives a breakdown of the data we are looking at, in this case we can see total count, mean(average), min and max for each of the columns in the dataset. For example we can see that the max age from patients was a 115 year old.



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

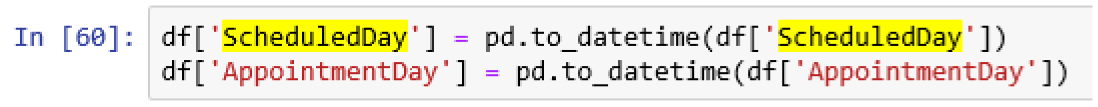
	PatientId	AppointmentID	Age	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received
count	1.105270e+05	1.105270e+05	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000
mean	1.474963e+14	5.675305e+06	37.088874	0.098266	0.197246	0.071865	0.030400	0.022248	0.321026
std	2.560949e+14	7.129575e+04	23.110205	0.297675	0.397921	0.258265	0.171686	0.161543	0.466873
min	3.921784e+04	5.030230e+06	-1.000000	0.000000	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	0.000000
50%	3.173184e+13	5.680573e+06	37.000000	0.000000	0.000000	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	0.000000	1.000000
max	9.999816e+14	5.790484e+06	115.000000	1.000000	1.000000	1.000000	1.000000	4.000000	1.000000

This gives us a feel for how the data is structured, df.info() is used to return information about the dataframe including the data types of each column.



```
In [9]: 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
```

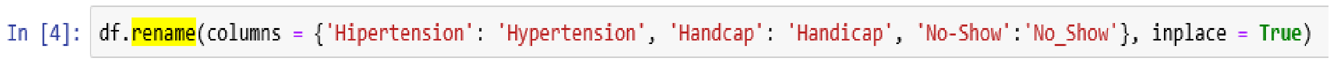
ScheduledDay & AppointmentDay data types were then converted from object to datetime, this makes the data easier to work with when carrying out further analysis. (See below)



```
In [60]: df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay'])
         df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay'])
```

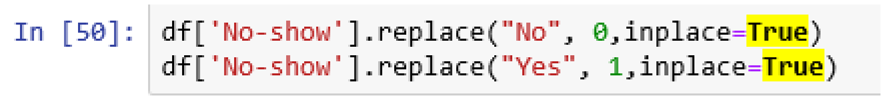
Data Cleaning

-Below, all necessary data cleaning steps were executed. The columns were renamed, as the spelling had to be corrected, and to ensure no issues occurred with columns regarding graphing the data.



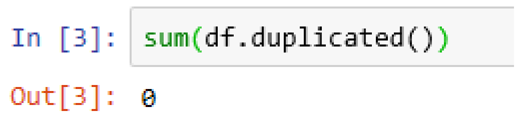
```
In [4]: df.rename(columns = {'Hipertension': 'Hypertension', 'Handcap': 'Handicap', 'No-Show':'No_Show'}, inplace = True)
```

The categorical column object data for no-show was replaced with integers 0 & 1 instead of “yes” & “no” to make this column more practical for statistical computations.



```
In [50]: df['No-show'].replace("No", 0,inplace=True)
         df['No-show'].replace("Yes", 1,inplace=True)
```

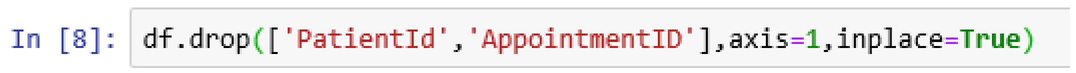
The following code was used to determine if the data had any duplicate rows.



```
In [3]: sum(df.duplicated())

Out[3]: 0
```

The output confirmed that no duplicate rows existed, and that there was no repeating Patient or AppointmentID's in the data.

Unnecessary columns like Patientid & AppointmentID were then dropped, as they were not required as part of the analysis, this was carried out as part of the data cleaning process.



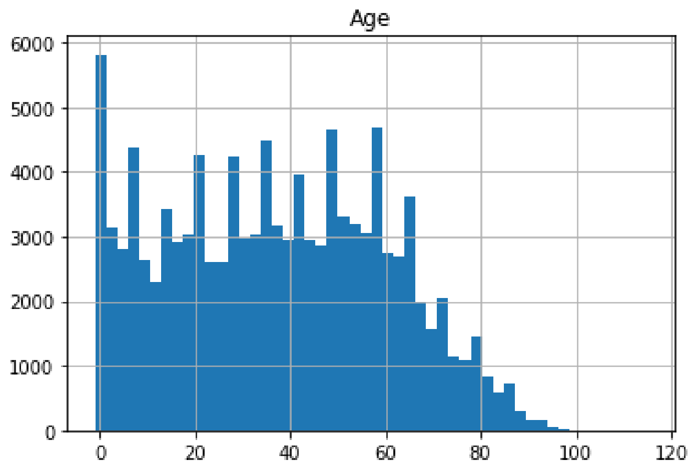
```
In [8]: df.drop(['PatientId','AppointmentID'],axis=1,inplace=True)
```

Exploratory Data Analysis

Research Question 1: Can knowing a patient's age help us to determine whether they will show up for their scheduled appointment?

The following histogram provides a clear breakdown of the number of employees who scheduled appointments in total by age. The highest bracket appears to be young children under 5 and adults aged 45-55.

```
In [29]: df.hist(column='Age', bins=50)

Out[29]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000023D3353D5F8>]],
               dtype=object)
```



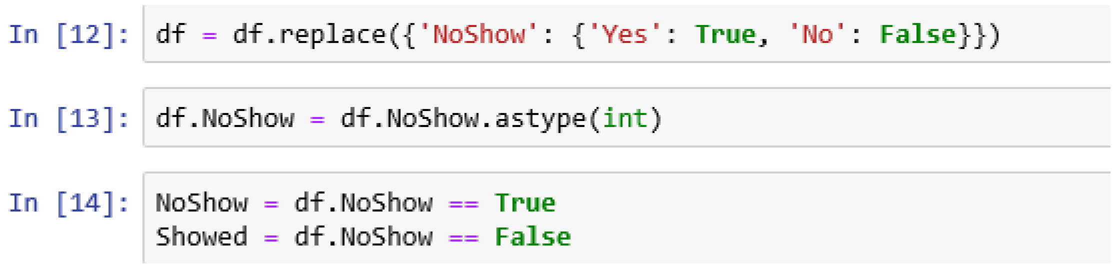
By breaking the age categories down into groups, we can get a mean of the number of no-show appointments per age group.

```
In [65]: cat = [0, 20, 40, 60, 100]
         age_groups = df.groupby(pd.cut(df.Age, cat))
         age_groups["No-show"].mean()

Out[65]: Age
         (0, 20]      0.228090
         (20, 40]     0.229790
         (40, 60]     0.185865
         (60, 100]    0.152012
         Name: No-show, dtype: float64
```

This data tells us that patients over 40 are more likely to attend their scheduled appointments that those under 40. Patients in the 60-100 age bracket are the best attenders of their appointments, while those in the 20-40 age bracket are the worst, accounting for 28.88% of the entire no-show population.

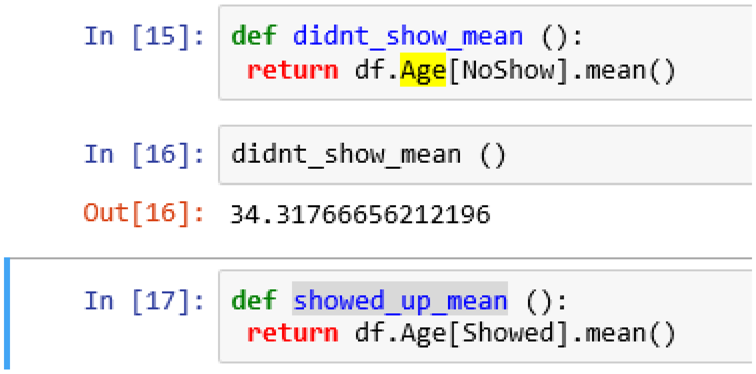
The average age of those who showed & didn't show for their medical appointments is shown below:

Replace NoShow indicator with true/false instead of yes & no. Changing the format to type Integer.

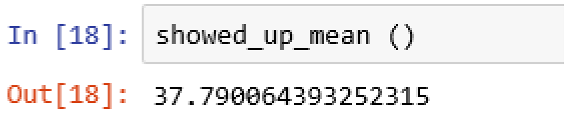


```
In [12]: df = df.replace({'NoShow': {'Yes': True, 'No': False}})

In [13]: df.NoShow = df.NoShow.astype(int)

In [14]: NoShow = df.NoShow == True
         Showed = df.NoShow == False
```

Mean calculated for those who showed & didn't show. Average age of no show's was 34 years old, and the average age of those who showed up in the dataset was 38 years old.

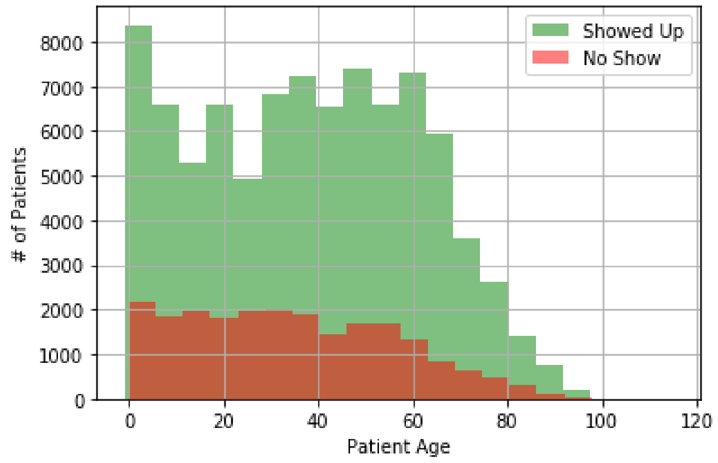


```
In [15]: def didnt_show_mean ():
          return df.Age[NoShow].mean()

In [16]: didnt_show_mean ()

Out[16]: 34.31766656212196

In [17]: def showed_up_mean ():
          return df.Age[Showed].mean()
```



```
In [18]: showed_up_mean ()

Out[18]: 37.790064393252315
```

```
In [19]: df.Age[Showed].hist(alpha=0.5, bins=20, label = 'Showed Up',color = 'green')
         df.Age[NoShow].hist(alpha=0.5,bins=20, label = 'No Show',color = 'red')
         plt.legend()
         plt.xlabel('Patient Age')
         plt.ylabel('# of Patients');
```



This histogram is of the age distribution of patients, clearly illustrating the difference between those who attended and those who didn't attend their appointments. It is clear to see that the highest attenders and no-shows are in the 0-5 age bracket as mentioned before.

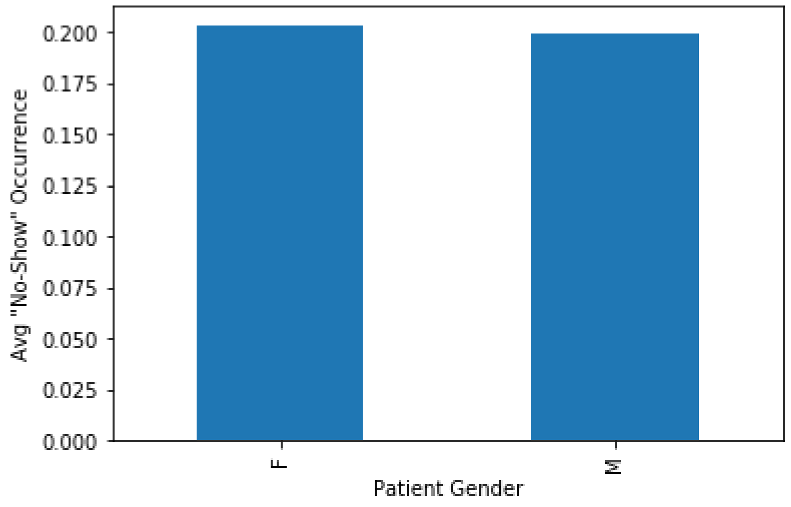
Research Question 2: Can knowing a patient's gender help us to determine whether they will show up for their scheduled appointment?

```
In [30]: df["Gender"].value_counts()

Out[30]: F    71840
         M    38687
         Name: Gender, dtype: int64
```

Value counts show that almost double of male appointments have been scheduled by females.

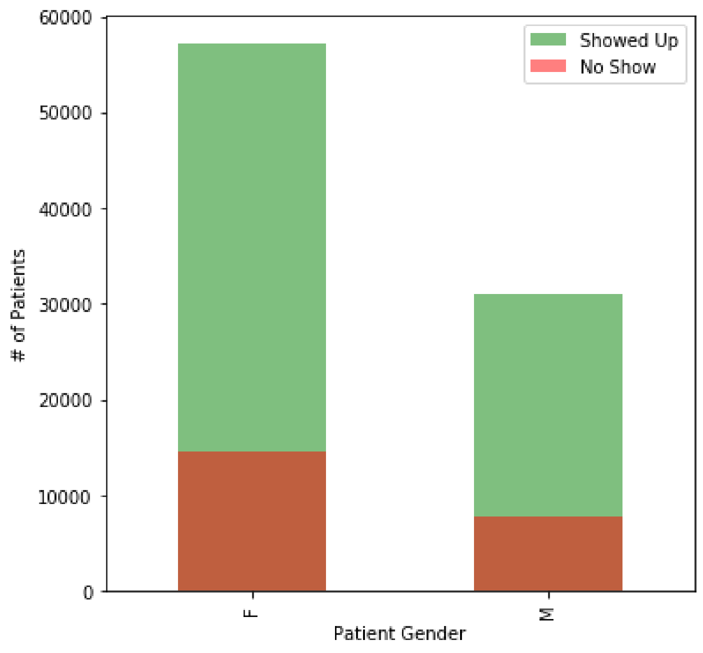
The following bar chart gives us the average no show occurrence by gender from this dataset. From this there is no real difference between both genders regarding no-shows for medical appointments, with females average being slightly higher than male.

```
In [21]: df.groupby('Gender').NoShow.mean().plot(kind='bar')
         plt.xlabel('Patient Gender')
         plt.ylabel('Avg "No-Show" Occurrence');
```



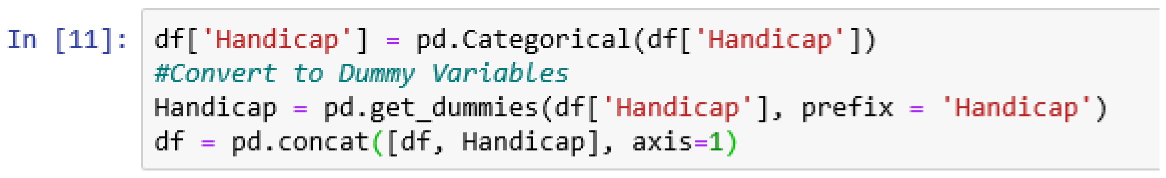
However, the bar chart below illustrates the total variance between male and female shows & no-shows for medical appointments. It is evident that there is a larger number of females who attend and do not show up for medical appointments from this chart.

```
In [24]: df.Gender[Showed].value_counts().plot(kind = 'bar',alpha=0.5,figsize = (6,6),
         label = 'Showed Up',color='green')
         df.Gender[NoShow].value_counts().plot(kind = 'bar', alpha=0.5,figsize = (6,
         6),label = 'No Show', color='red')
         plt.legend()
         plt.xlabel('Patient Gender')
         plt.ylabel('# of Patients');
```

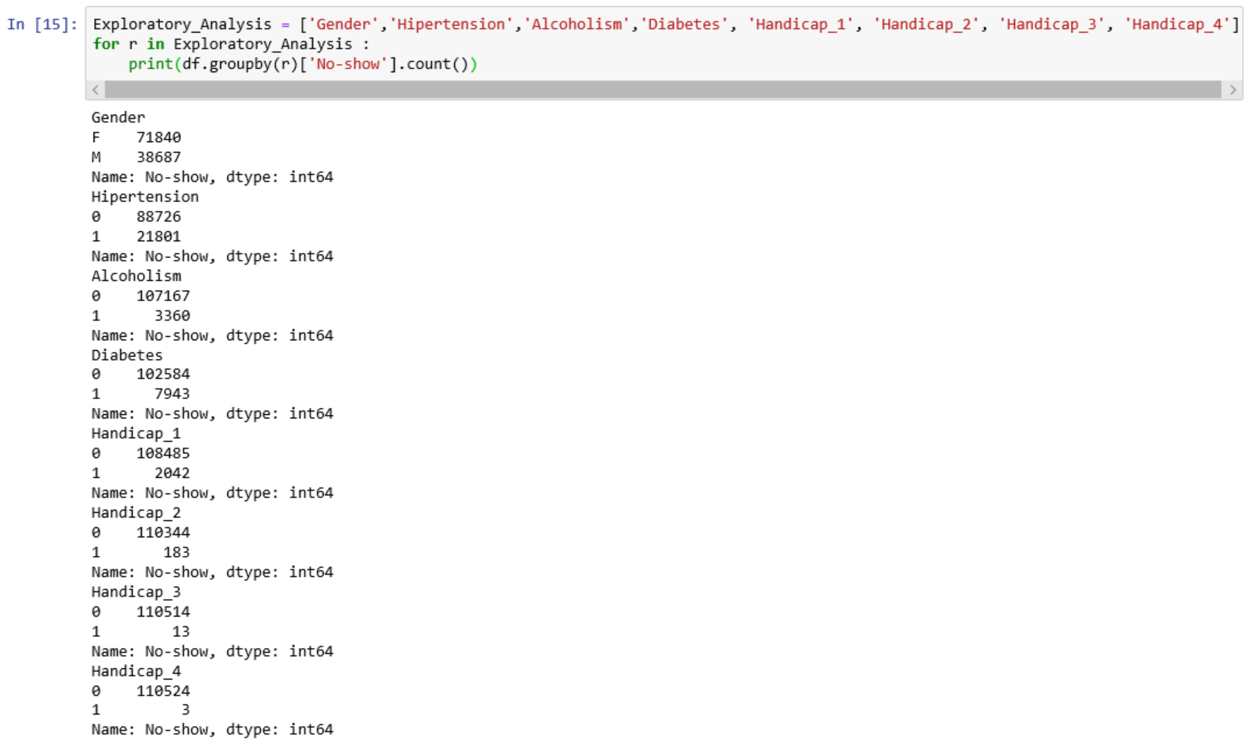


Research Question 3: Can knowing a patient's illness help us to determine whether they will show up for their scheduled appointment?

Variable was created to consider all illnesses of the patient & Gender to find the average number of no-shows for each of the categories: Handicap is broken down to 4 different categories. For this reason, it had to converted to a categorical variable & dummy variable had to be created for each of the categories in order to make this illness usable for data modelling. For each of the categories, it is evident that attended appointments is clearly higher than no-show appointments. For the physically challenged categories this is also reflected, were Handicap_1 having the highest number of no-shows of the four.



```
In [11]: df['Handicap'] = pd.Categorical(df['Handicap'])
         #Convert to Dummy Variables
         Handicap = pd.get_dummies(df['Handicap'], prefix = 'Handicap')
         df = pd.concat([df, Handicap], axis=1)
```

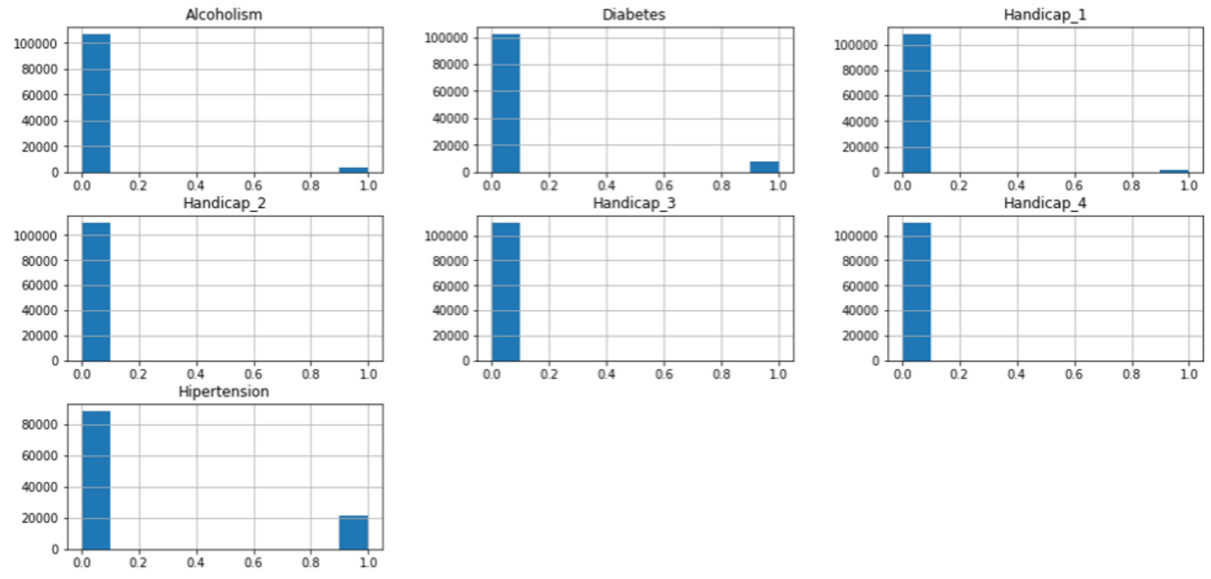


```
In [15]: Exploratory_Analysis = ['Gender','Hipertension','Alcoholism','Diabetes', 'Handicap_1', 'Handicap_2', 'Handicap_3', 'Handicap_4']
         for r in Exploratory_Analysis :
             print(df.groupby(r)['No-show'].count())
```

```
Gender
F    71840
M    38687
Name: No-show, dtype: int64
Hipertension
0    88726
1    21801
Name: No-show, dtype: int64
Alcoholism
0    107167
1      3360
Name: No-show, dtype: int64
Diabetes
0    102584
1      7943
Name: No-show, dtype: int64
Handicap_1
0    108485
1      2042
Name: No-show, dtype: int64
Handicap_2
0    110344
1       183
Name: No-show, dtype: int64
Handicap_3
0    110514
1        13
Name: No-show, dtype: int64
Handicap_4
0    110524
1         3
Name: No-show, dtype: int64
```

The histogram provides a graphical representation of the count breakdown by illness, this shows us that patients with Hyperextension are most likely to not show up for their medical appointment with over 21,000 patients a no-show with this illness. Second highest category of no show's was diabetics. With the physically challenged categories being clearly the best attenders of medical appointments with the least no-shows.

In [19]:
```
df.hist(Exploratory_Analysis,figsize = (17,8));
```

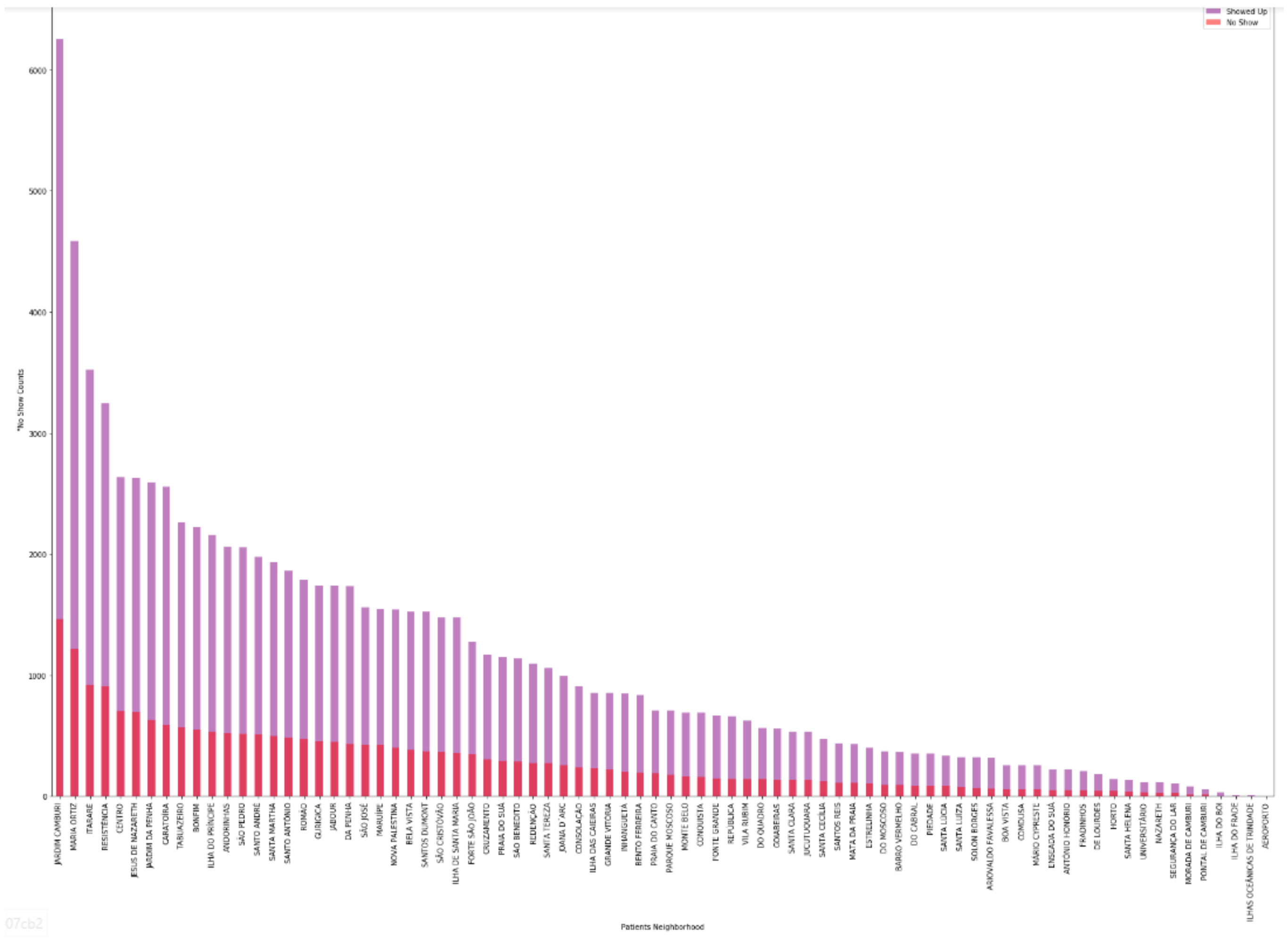


Research Question 4: Can knowing the town that a patient is from help us to determine whether they will show up for their scheduled appointment?

The chart below of each neighborhood provides a comprehensive overview of each town’s appointments with regard to attending compared to no-show’s. This is strongly helpful for our predictive analysis to determine whether a patient will attend their appointment depending on their location.

In [26]:
```
df.Neighbourhood[Showed].value_counts().plot(kind = 'bar',alpha=0.5,figsize =
(30,20),label = 'Showed Up',color='purple')
df.Neighbourhood[NoShow].value_counts().plot(kind = 'bar', alpha=0.5,figsize = (30,20),label = 'No Show', color='red')
plt.legend()
plt.xlabel('Patients Neighborhood')
plt.ylabel('"No Show Counts');
```

The purple shows the appointments where the patients showed up, with red representing the no-shows. Although JARDIM CAMBURI has the most no-show’s, it must be noted that this area has the most appointments scheduled also, MARIA ORTIZ has almost as many no-shows with over 1500 less attended appointments.



This graph proves that there is a strong correlation between the neighborhood a patient is from & the possibility of them turning up for their appointment. This analysis on the patient's location only really provides a high-level oversight and it is still difficult to determine the statistical importance of this link, a deeper analysis would therefore need to be carried out.

Limitations:

The results from this investigation were limited to insights that could only be extracted from the no-show medical appointments data set alone. We would need more details regarding personal information of the individual patient to carry out a more accurate comprehensive analysis of the statistics provided.

Conclusions:

After investigating this data file containing 100,000 medical appointments from brazil, and with particular interest being paid to the three independent variables age, gender & neighborhood, with the dependent variable used was determining whether a patient will turn up for their medical appointment or not. From the first 2 questions posed age and gender do not have strong enough links to help us properly determine whether a patient will show up for their appointment or not. However, I feel that the illness and neighborhood of the patient are both stronger factors in helping determine whether they will appear. For instance, there is almost a 1 in 4 chance that a patient with hypertension will not show up for their appointment & we can also see a strong link between where a patient is from and if they will show up for their appointment based of the data provided. It is worth noting that working off these descriptive statistics alone is largely inconclusive & would require something of a different approach using inferential statistics to delve deeper into the data available.