DATA ANALYST NANODEGREE - UDACITY
3RD PROJECT

INVESTIGATE A DATASET

MEDICAL APPOINMENTS EXPLORATION

PREPARED BY
MANAR S. ELMASSAH
Aspiring Business Analyst

AUGUST 2019

Project: Investigate a Dataset (Medical Appoinments Exploration)

Table of Contents

Introduction

In this project we're going to explore a dataset provided through Kaggle "No-Show Appointments"

> 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.

Where going to preform the data analysis process from cleaning and wrangling to exploration and finally delivering findings.

The questions we're going to ask are:

```
1. Does recieving a reminder (SMS) affect the rate of attendence?
2.Does being enrolled in the wellfare program (Scholership) affect the attendance rate?
```

In [67]:

```
#  import statements for all of the packages that will be used.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
```

Data Wrangling

> In this section we're going to take an overview of the dataset, see if it needs steps further to optmize it for analysis.

General Properties

In [68]:

```
# Load the dataset and print out a few lines to inspect it
df= pd.read_csv("noshowappointments-may-2016.csv")
df.head()
```

Out[68]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	0	1	0
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	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
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	0	0	0

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes
4	8.841186e+12	5642494	F	29T16:07:23Z	29T00:00:00Z	56	PENHA	0	1	1

Checking for Null Values

After comparing the findings it's found that there are no missing data but there is an unsuitable data type in the columns "ScheduledDay" and "AppointmentDay".

In [69]:

```
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
```

Checking for duplicates

After running the code it's found that there are no duplicated rows withing the dataset

In [70]:

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

Out[70]:

```
0
```

Data Cleaning

Overall the dataset seems pretty clean and doesn't need any further steps to start analyzing it.

However we're going to trim the data parts we don't need to answer our questions to optimize the dataset.

In [71]:

```
# Dropping Unnecessary columns
df.drop(columns=['PatientId','ScheduledDay','AppointmentDay','Neighbourhood'] , inplace = True)
df.head()
```

Out[71]:

	AppointmentID	Gender	Age	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	5642903	F	62	0	1	0	0	0	0	No
1	5642503	M	56	0	0	0	0	0	0	No
2	5642549	F	62	0	0	0	0	0	0	No
3	5642828	F	8	0	0	0	0	0	0	No
4	5642494	F	56	0	1	1	0	0	0	No

Exploratory Data Analysis

Now We're ready to start exploring the data we have to answer the questions posed earlier.

Does recieving a reminder (SMS) affect the rate of attendence?

In order for us to investigate this answer we need to compare the percentage of show ups and the No-shows based on who recieved a notification message.

In [72]:

```
#Overview of the dataset's statistics

df.describe()
```

Out[72]:

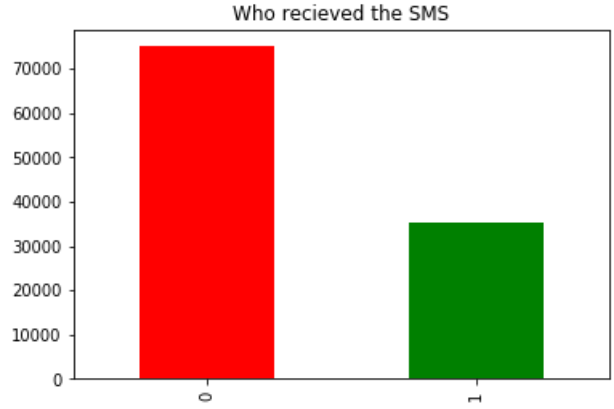
	AppointmentID	Age	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received
count	1.105270e+05	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000
mean	5.675305e+06	37.088874	0.098266	0.197246	0.071865	0.030400	0.022248	0.321026
std	7.129575e+04	23.110205	0.297675	0.397921	0.258265	0.171686	0.161543	0.466873
min	5.030230e+06	-1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	5.640286e+06	18.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	5.680573e+06	37.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	5.725524e+06	55.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
max	5.790484e+06	115.000000	1.000000	1.000000	1.000000	1.000000	4.000000	1.000000

In [73]:

```
#Analyzing the statistics of 'SMS' column
sms_count =df['SMS_received'].value_counts()
```

In [74]:

```
#Plotting the rate of people who recieved the SMS
plot1 =sms_count.plot.bar(color = ["red","green"]);
plot1.set_title("Who recieved the SMS" );
```



From what's showed above it's found that only 32.1% of the total popullation recieved the SMS

Let's take a look over the data of who attended and who didn't

In [75]:

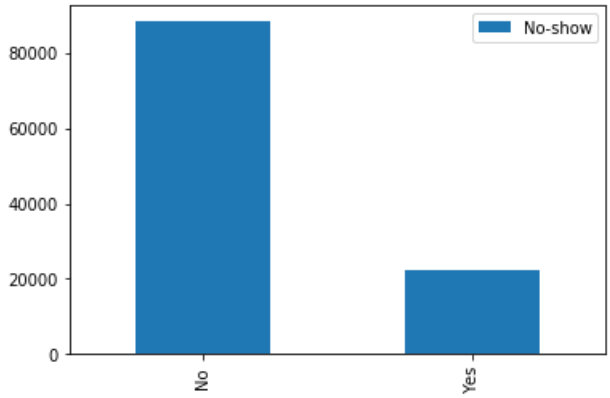
```
#Analyzing the statistics of 'No-Show' column
```

```
no_shows= df['No-show'].value_counts()
print(no_shows)
```

```
No      88208
Yes     22319
Name: No-show, dtype: int64
```

In [76]:

```
#Plotting the rate of people who showed-up
plot2 = no_shows.plot.bar();
plot2.legend();
```



In [77]:

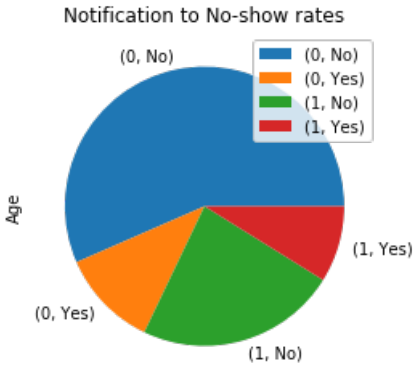
```
df.groupby(by = ['SMS_received','No-show']).count()
```

Out[77]:

		AppointmentID	Gender	Age	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap
SMS_received	**No-show**								
0	**No**	62510	62510	62510	62510	62510	62510	62510	62510
	Yes	12535	12535	12535	12535	12535	12535	12535	12535
1	**No**	25698	25698	25698	25698	25698	25698	25698	25698
	Yes	9784	9784	9784	9784	9784	9784	9784	9784

In [78]:

```
# Comparison between Show-Ups and No-Shows and whither they recived notification
pie = df.groupby(by = ['SMS_received','No-show']).count()['Age'].plot(kind = 'pie');
pie.set_title("Notification to No-show rates");
pie.legend();
```



Result

From what's found after analysing the data we can see that:

```
1.Most people showed up for their appointments even without recieving a notification
2.The number of people who missed the appointment after being notified is less than who we
ren't
```

Does being enrolled in the wellfare program (Scholership) affect the attendance rate?

In [79]:

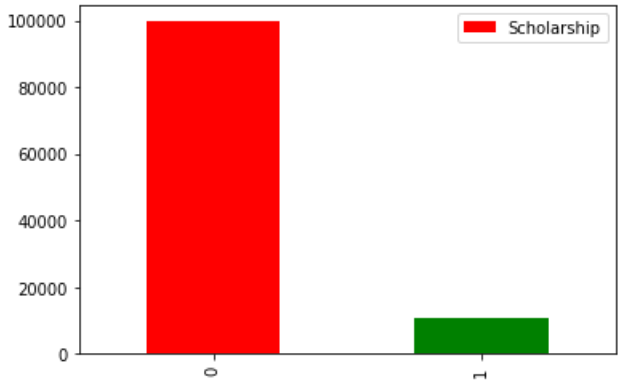
```
# The wellfare programs stats
wellfare_count =df['Scholarship'].value_counts()

print(wellfare_count)
```

```
0    99666
1    10861
Name: Scholarship, dtype: int64
```

In [80]:

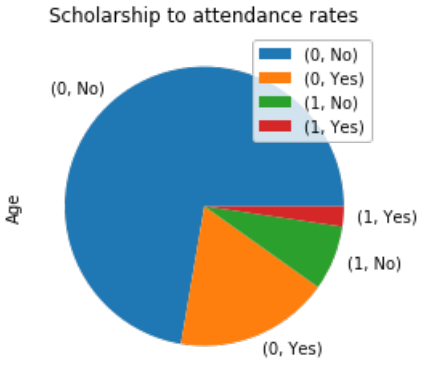
```
plot3 = wellfare_count.plot.bar(color = ['red' , 'green']);
plot3.legend();
```



We can see here that the majority of people in our dataset are not enrolled in the wellfare program (almost 90%).

In [81]:

```
# Comparison between Show-Ups and No-Shows and whither they are enrolled in the wellfare program
pie2 = df.groupby(by = ['Scholarship','No-show']).count()['Age'].plot(kind = 'pie');
pie2.set_title("Scholarship to attendance rates");
pie2.legend();
```



Result

From what's found after analysing the data we can see that:

1.Most people showed up for their appointments even without being enrolled in the wellfare program
2.The number of people who missed the appointment will being a part of the health program i s much less than who aren't

Conclusions

> So at the end of our analysis we can conclude that the attendance rate to medical appointments is not affected by reminders or being a part of health care programs. The rates do improve a little if those conditions are met but it's merely trivial.
>
> This is because medical appointments are mostly needed with urgency that it wouldn't be affected much with secondary circumstances.

Limitations

Through this report I was faced with some difficulties like:

1. The data was rough to handle in regards to the string columns
2.The modification of plot label for ticks
3.The data was mostly inconclusive in regards to the questions I asked at the beginning

References

1. https://matplotlib.org/3.1.1/gallery/pie_and_polar_charts/pie_and_donut_labels.html#sphx-glr-gallery-pie-and-polar-charts-pie-and-donut-labels-py

2.https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.groupby.html

In [82]:

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

Out[82]:

0