Investigate_a_Dataset

September 2, 2022

1 Project: Investigate a Dataset - No-show appointments

1.1 Table of Contents

1.1.1 Dataset Description

This data set collected info. about patients in Brazil which have booked appointments with the doctors and recieved SMS notifications and all the instruction and still skipped thier appointment in order to try and find a reason and a correlation behind this.

1.1.2 This dataset contains 14 columns :

- PatientId : a unique no for each patient
- AppointmentID : a unique no for each appointment
- Gender : The Sex of the patient
- ScheduledDay : Tells us when did the patient make his appointment
- AppointmentDay : The day where the patient is scheduled to make his visit to the doctor
- Age : Patient's age
- Neighbourhood : The neighbourhood in which the hospital resides
- Scholarship : The patient is enrolled or not enrolled in the Brazilian wellfare program
- [Hipertension, Diabetes, Alcoholism, Handcap] : Some of the illnesses that might be related to the patient not showing
- SMS_received : Weather the patient recieved an SMS notification or not.
- No-show : The patient showed up to the appointment or not

1.1.3 Question(s) for Analysis

We'll start this analysis by posing the following questions:
1 - What is the percentage of people showing up to appointments to those who didn't show up
2 - Is there a relation between thier age and the patient not showing up
3 - Which hospitals have a higher rate of "No show" than others ?

4 - is there any relation between the person being signed up in the Brazilian wellfare program and not showing up to appointments

5 - Are SMS notifications helpful reminders for the patients to show up ?

```
In [1]: # Use this cell to set up import statements for all of the packages that you
        #   plan to use.

        # Remember to include a 'magic word' so that your visualizations are plotted
        #   inline with the notebook. See this page for more:
        #   http://ipython.readthedocs.io/en/stable/interactive/magics.html
        import os
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline

In [2]: # Upgrade pandas to use dataframe.explode() function.
        !pip install --upgrade pandas==0.25.0

Collecting pandas==0.25.0
  Downloading https://files.pythonhosted.org/packages/1d/9a/7eb9952f4b4d73fbd75ad1d5d6112f407e69
    100% || 10.5MB 2.1MB/s eta 0:00:01 0% |                                | 102kB 4.8MB/s eta 0:
Requirement already satisfied, skipping upgrade: python-dateutil>=2.6.1 in /opt/conda/lib/python
Requirement already satisfied, skipping upgrade: pytz>=2017.2 in /opt/conda/lib/python3.6/site-p
Collecting numpy>=1.13.3 (from pandas==0.25.0)
  Downloading https://files.pythonhosted.org/packages/45/b2/6c7545bb7a38754d63048c7696804a0d9473
    100% || 13.4MB 2.3MB/s eta 0:00:01   47% |                 | 6.3MB 24.0MB/s eta 0:00:01    6
Requirement already satisfied, skipping upgrade: six>=1.5 in /opt/conda/lib/python3.6/site-packa
tensorflow 1.3.0 requires tensorflow-tensorboard<0.2.0,>=0.1.0, which is not installed.
Installing collected packages: numpy, pandas
  Found existing installation: numpy 1.12.1
    Uninstalling numpy-1.12.1:
      Successfully uninstalled numpy-1.12.1
  Found existing installation: pandas 0.23.3
    Uninstalling pandas-0.23.3:
      Successfully uninstalled pandas-0.23.3
Successfully installed numpy-1.19.5 pandas-0.25.0
```

Data Wrangling

1.1.4 General Properties

```
In [2]: # Load your data and print out a few lines. Perform operations to inspect data
        #   types and look for instances of missing or possibly errant data.
        df = pd.read_csv('Database_No_show_appointments/noshowappointments-kagglev2-may-2016.csv
        df.head()
```

Out[2]:

	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	Hipertension	\
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 [3]:
```
# Checking dataset describtion
df.shape
```

Out[3]: (110527, 14)

In [4]:
```
#checking data types
df.dtypes
```

Out[4]:
```
PatientId         float64
AppointmentID       int64
Gender             object
ScheduledDay       object
AppointmentDay     object
Age                 int64
Neighbourhood      object
Scholarship         int64
Hipertension        int64
Diabetes            int64
Alcoholism          int64
Handcap             int64
SMS_received        int64
No-show            object
dtype: object
```

In [5]:
```
df.describe()
```

Out[5]:

	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
```

In [6]:
```
#checking for NaN values
df.isna().any()
```

Out[6]:
```
PatientId         False
AppointmentID     False
Gender            False
ScheduledDay      False
AppointmentDay    False
Age               False
Neighbourhood     False
Scholarship       False
Hipertension      False
Diabetes          False
Alcoholism        False
Handcap           False
SMS_received      False
No-show           False
dtype: bool
```

In [7]:
```
#Checking for duplicates
df.duplicated().any()
```

Out[7]: False

1.1.5 Data Cleaning

In [8]:
```
# After discussing the structure of the data and any problems that need to be
#   cleaned, perform those cleaning steps in the second part of this section.
#adjusting columns names to more pythonic names
df.columns=df.columns.str.lower()
df.rename(columns = {"no-show":"no_show"}, inplace = True)
df.head()
```

Out[8]:

	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

In [9]:
```
# Defining a function to replace columns with ones and zeros to yes and no

def replace_values(values_to_replace):
    for value in values_to_replace :
        df[value].replace({0:'No',
                           1:'Yes'}, inplace = True)
```

In [10]:
```
# Changing values in columns with zeroes and ones to yes and no
columns_need_change = ['scholarship','hipertension', 'diabetes','alcoholism','sms_recei
replace_values(columns_need_change)
df.head()
```

Out[10]:

	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	No	Yes	No	No	0	0	No
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	No	No	No	No	0	0	No
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	No	No	No	No	0	0	No
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	No	No	No	No	0	0	No

```
4  8.841186e+12        5642494      F  2016-04-29T16:07:23Z

          appointmentday  age      neighbourhood scholarship hipertension  \
0  2016-04-29T00:00:00Z   62    JARDIM DA PENHA          No          Yes
1  2016-04-29T00:00:00Z   56    JARDIM DA PENHA          No           No
2  2016-04-29T00:00:00Z   62      MATA DA PRAIA          No           No
3  2016-04-29T00:00:00Z    8  PONTAL DE CAMBURI          No           No
4  2016-04-29T00:00:00Z   56    JARDIM DA PENHA          No          Yes

  diabetes alcoholism  handcap sms_received no_show
0       No         No        0           No      No
1       No         No        0           No      No
2       No         No        0           No      No
3       No         No        0           No      No
4      Yes         No        0           No      No
```

```
In [11]: #handcap column shows how many handcappings this person has , so it will not be changed
         df.handcap.unique()
```

```
Out[11]: array([0, 1, 2, 3, 4])
```

```
In [12]: # The age column has a negative value , lets see what can be done
         df.query('age == -1')
```

```
Out[12]:           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          No           No

               diabetes alcoholism  handcap sms_received no_show
         99832       No         No        0           No      No
```

```
In [13]: # Lets replace it by the mean age
         mean_age = df['age'].mean()
         df['age'].replace({
             -1 : mean_age
         }, inplace = True)
         # checking if the value still exists
         df.query('age == -1')
```

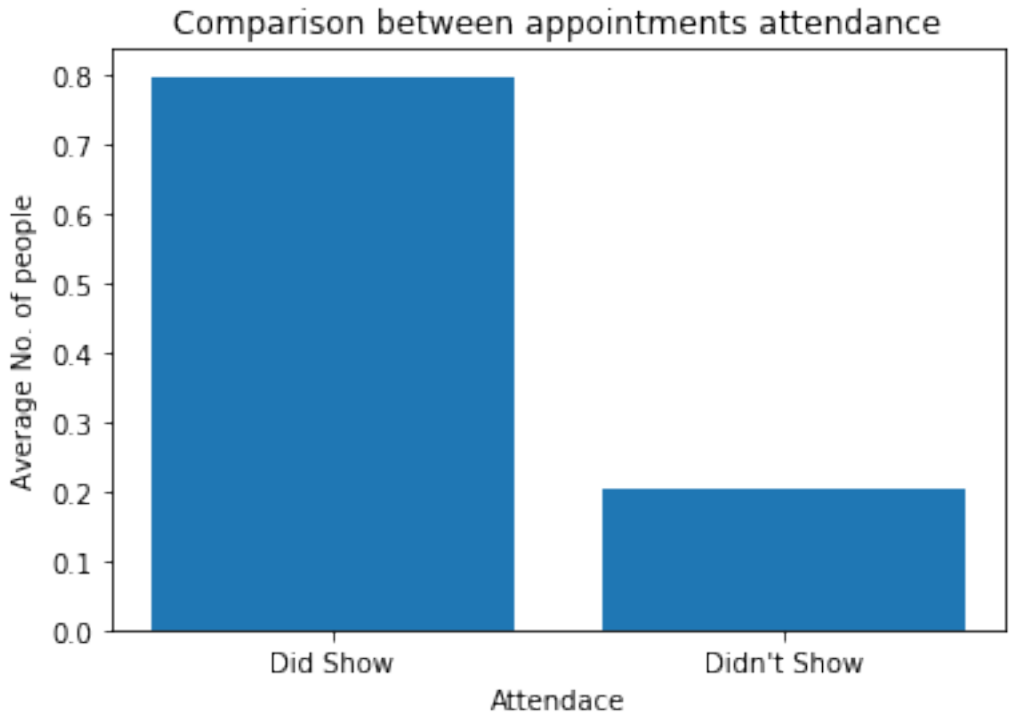
```
Out[13]: Empty DataFrame
         Columns: [patientid, appointmentid, gender, scheduledday, appointmentday, age, neighbou
         Index: []
```

Exploratory Data Analysis

1.1.6 Research Question 1 What is the percentage of people showing up to appointments to those who didn't show up

In [14]:
```
# Use this, and more code cells, to explore your data. Don't forget to add
#   Markdown cells to document your observations and findings.
# We calculate The total number of patients and then we find the related number of peop
total_no_show = df.no_show.count()
did_show = df.no_show.value_counts()[0]
didnt_show = df.no_show.value_counts()[1]
perc_did_show= did_show/total_no_show
perc_didnt_show = didnt_show/total_no_show
```

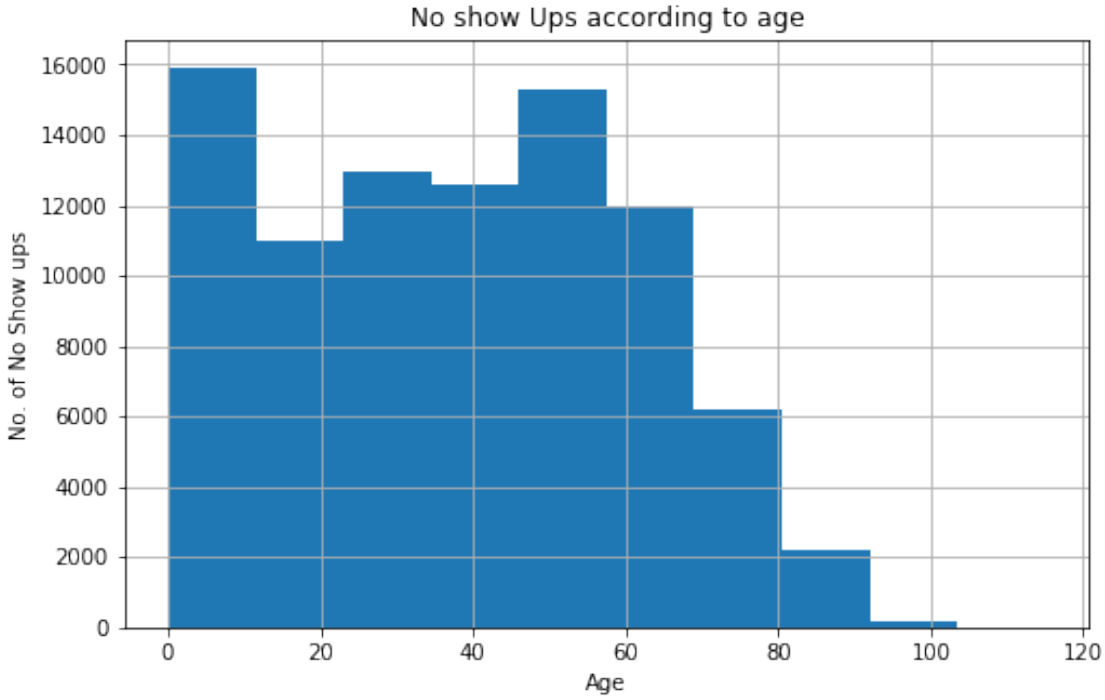
In [15]:
```
locations = [1, 2]
heights = [perc_did_show, perc_didnt_show]
labels = ['Did Show', "Didn't Show"]
plt.bar(locations, heights, tick_label=labels)
plt.title('Comparison between appointments attendance')
plt.xlabel('Attendace')
plt.ylabel('Average No. of people');
```



From the previous plot we can see that there is a high number of attendace compared to a small percentage of people who don't show up to thier appointments

1.1.7 Research Question 2 Is there a relation between thier age and the patient not showing up?

```
In [16]: # Continue to explore the data to address your additional research
         #   questions. Add more headers as needed if you have more questions to
         #   investigate.
         # We check this by dividing people who don't attend to subgroups and plotting them to a
         plt.subplots(figsize = (8,5))
         df.query('no_show == "No"')['age'].hist(bins = 10)
         plt.title('No show Ups according to age')
         plt.xlabel('Age')
         plt.ylabel('No. of No Show ups');
```



From the Visual we can clearly see that the number of No Show ups is higher in the younger people while older people tend to not miss thier appointments

1.1.8 Research Question 3 - Which hospitals have a higher rate of "No show" than others ?

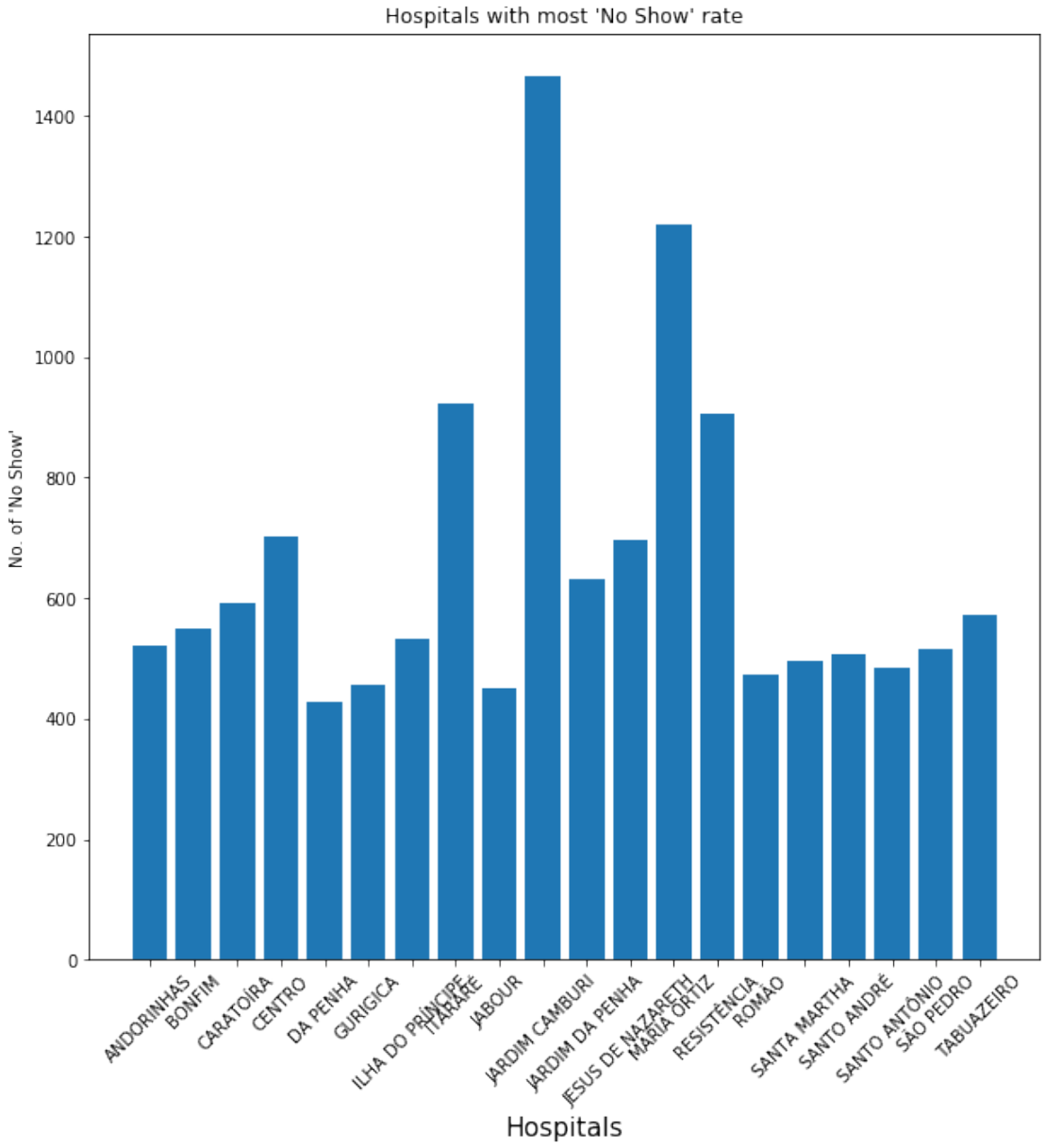
```
In [17]: # Extracting the series' data into a list in order to obtain values

         neighbourhood_name = df.query('no_show == "Yes"')['neighbourhood'].value_counts().keys(
         neighbourhood_num = df.query('no_show == "Yes"')['neighbourhood'].value_counts().tolist

         print(f"The neighbourhood with the most 'No Show'is :{neighbourhood_name[0]}")
         print(f"With a total number of {neighbourhood_num[0]}")
```

```
The neighbourhood with the most 'No Show'is :JARDIM CAMBURI
With a total number of 1465
```

```
In [18]: # Plotting the first top 20 resulting Hostpitals with the most "No show" rate
         plt.subplots(figsize = (10,10))
         plt.title("Hospitals with most 'No Show' rate")
         plt.xticks(rotation = 45, fontsize = 10)
         plt.xlabel("Hospitals", fontsize = 15)
         plt.ylabel("No. of 'No Show'")
         plt.bar(neighbourhood_name[:20],neighbourhood_num[:20]);
```

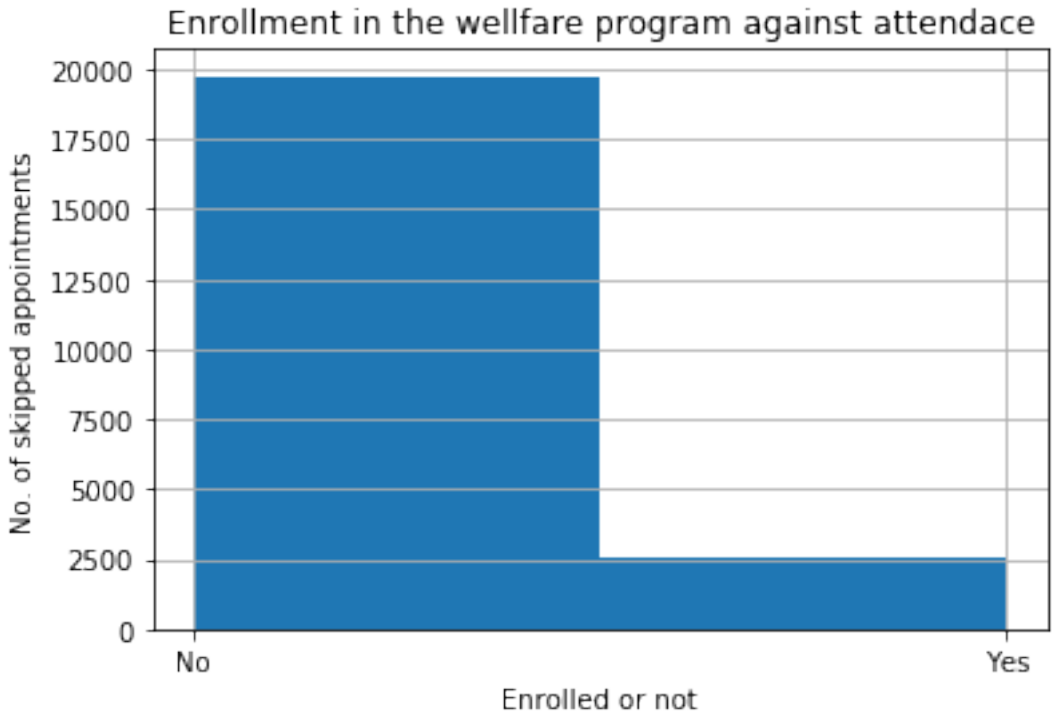


As shown in the prevoius figure and results , some hospitals do have a very high rate of "No Show" Which shows that there are issues to address regarding these hospitals.

1.1.9 Research Question 4 - is there any relation between the person being signed up in the Brazilian wellfare program and not showing up to appointments

Next We want to check if there is a relation between enrolling in the Brazilian wellfare program and the appointments attendance

```
In [19]: # First we plot the relation between "No show " and the Scholarship
         df.query('no_show == "Yes"')['scholarship'].hist(bins = 2);
         plt.title("Enrollment in the wellfare program against attendace")
         plt.xlabel("Enrolled or not")
         plt.ylabel("No. of skipped appointments");
```



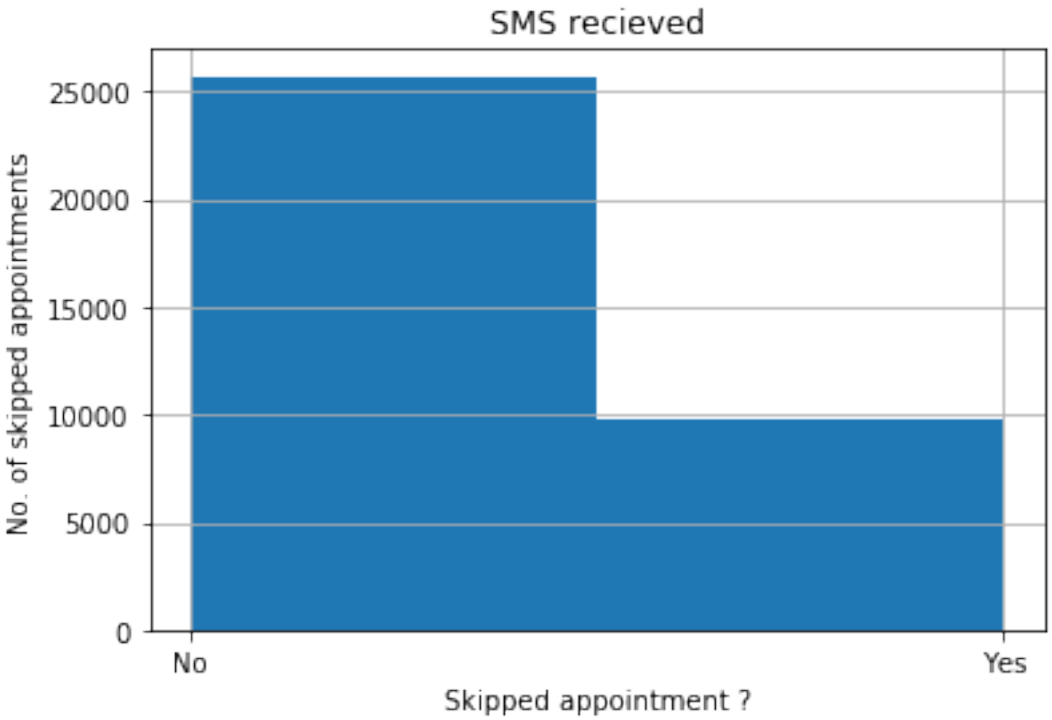
The previous plot shows that people not enrolled in the Brazilian wellfare program are more likely to skip attendance for the medical appointments

1.1.10 Research Question 5 - Are SMS notifications helpful reminders for the patients to show up ?

In order to check the relation , lets check the probability of attendance if the person recieved an SMS

```
In [20]: df.query('sms_received == "Yes"')['no_show'].hist(bins = 2)
         plt.title("SMS recieved")
```

```
plt.xlabel("Skipped appointment ?")
plt.ylabel("No. of skipped appointments");
```



We can see from the previous plot that people that recieved an SMS are more likely to attend thier appointments

Conclusions

From the previous study we've found out the following : 1 - There is a high percentage of people showing up to thier appointments than not although some improvements could be done towards identifying the causes that prevent people from showing up

2- We concluded that the majority of young people skip thier appointments while the older people don't

3 - We have found some Hospitals to have a very high rate of people not showing up

4 - We have found that people enrolled in the Brazilian wellfare program are more likely to attend thier appointments rather than thier counterparts which have a higher "No show " rate

5 - The SMS reminders have been successful to an extent , as people who recieve SMS reminders are more likely to show up to thier appointments

1.2 Limitations

1 - The main limitation was that there was no more available data about the hospital services to determine what caused some hospitals to have a higher "no-show" rate than others , we should further study these cases (no. of staff , no. of beds , available equipment,...etc)

```
In [21]: from subprocess import call
         call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])
```

```
Out[21]: 0
```