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

Table of Contents

Introduction

Dataset Description

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.

Question(s) for Analysis

predict if a patient will show up for their scheduled appointment or not?

In [1]:
```
# Use this cell to set up import statements for all of the packages that you
#   plan to use.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 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
% matplotlib inline
```

Data Wrangling

> **Tip**: In this section of the report, you will load in the data, check for cleanliness, and then trim and clean your dataset for analysis. Make sure that you **document your data cleaning steps in mark-down cells precisely and justify your cleaning decisions.**

General Properties

> **Tip**: You should *not* perform too many operations in each cell. Create cells freely to explore your data. One option that you can take with this project is to do a lot of explorations in an initial notebook. These don't have to be organized, but make sure you use enough comments to understand the purpose of each code cell. Then, after you're done with your analysis, create a duplicate notebook where you will trim the excess and organize your steps so that you have a flowing, cohesive report.

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('noshowappointments-kagglev2-may-2016.csv')
```

In [3]:
```
#inspect data types and look for missing data
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
```

No missing values

In [4]:
```
df.head()
```

Out[4]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood
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 [5]:
```
#find num of rows and columns
df.shape
```

Out[5]:
```
(110527, 14)
```

In [6]:
```
df.describe()
```

Out[6]:

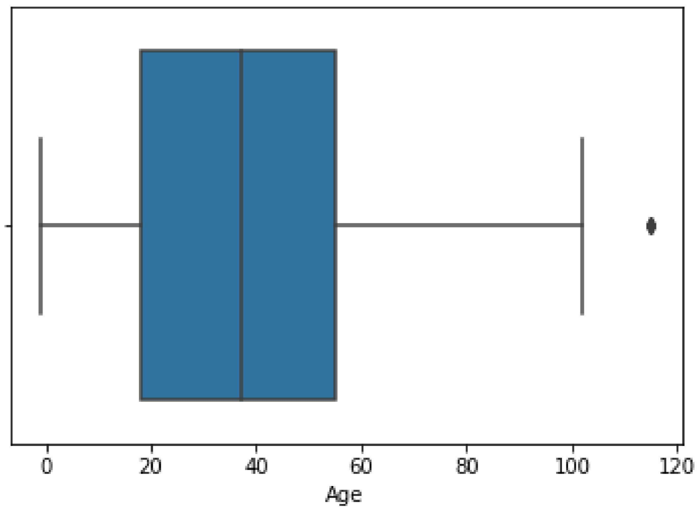
	PatientId	AppointmentID	Age	Scholarship	Hipertension	Diabetes
count	1.105270e+05	1.105270e+05	110527.000000	110527.000000	110527.000000	110527.00000
mean	1.474963e+14	5.675305e+06	37.088874	0.098266	0.197246	0.07186
std	2.560949e+14	7.129575e+04	23.110205	0.297675	0.397921	0.25826
min	3.921784e+04	5.030230e+06	-1.000000	0.000000	0.000000	0.00000
25%	4.172614e+12	5.640286e+06	18.000000	0.000000	0.000000	0.00000
50%	3.173184e+13	5.680573e+06	37.000000	0.000000	0.000000	0.00000
75%	9.439172e+13	5.725524e+06	55.000000	0.000000	0.000000	0.00000
max	9.999816e+14	5.790484e+06	115.000000	1.000000	1.000000	1.00000

mean of age is 37 years max age is 115 min age is -1 max and min age seems to be wrong

Data Cleaning

The outliers in the Age field seems to be be errors. The youngest person is -1 and the oldest is 115. Boxplots provide a way to visually identify outliers.

In [7]:
```
# create a boxplot of age using seaborn
sns.boxplot(df.Age)
plt.show()
```



The boxplot confirms that there are outliers at the end of the range.

In [8]:
```
df[df.Age < 0]
```

Out[8]:

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

Only one record was found seems to be an error

In [9]:
```
df[df.Age > 100]
```

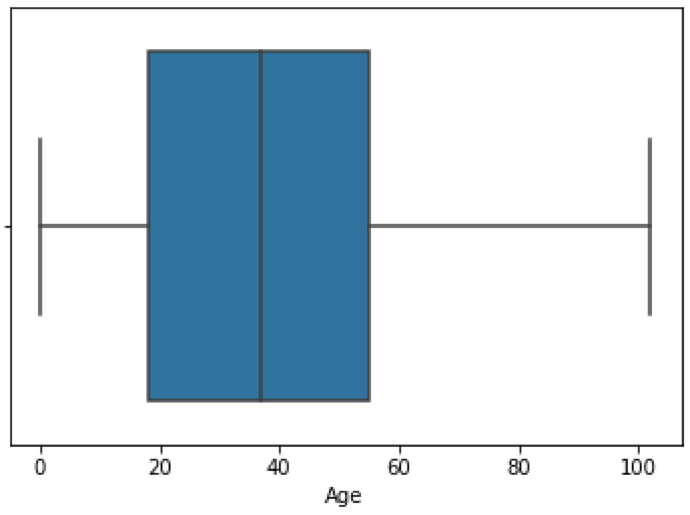
Out[9]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourho
58014	9.762948e+14	5651757	F	2016-05-03T09:14:53Z	2016-05-03T00:00:00Z	102	CONQUIS
63912	3.196321e+13	5700278	F	2016-05-16T09:17:44Z	2016-05-19T00:00:00Z	115	ANDORINH
63915	3.196321e+13	5700279	F	2016-05-16T09:17:44Z	2016-05-19T00:00:00Z	115	ANDORINH
68127	3.196321e+13	5562812	F	2016-04-08T14:29:17Z	2016-05-16T00:00:00Z	115	ANDORINH
76284	3.196321e+13	5744037	F	2016-05-30T09:44:51Z	2016-05-30T00:00:00Z	115	ANDORINH
90372	2.342836e+11	5751563	F	2016-05-31T10:19:49Z	2016-06-02T00:00:00Z	102	MARIA OR
97666	7.482346e+14	5717451	F	2016-05-19T07:57:56Z	2016-06-03T00:00:00Z	115	SÃO JC

There are two patients who were 115 years old ,this very rare seems to be an error

remove the records with outlier ages.

In [10]:
```
df = df[(df.Age > -1) & (df.Age < 115)]
# display the boxplot again to verify our new outcome
sns.boxplot(df.Age)
plt.show()
```

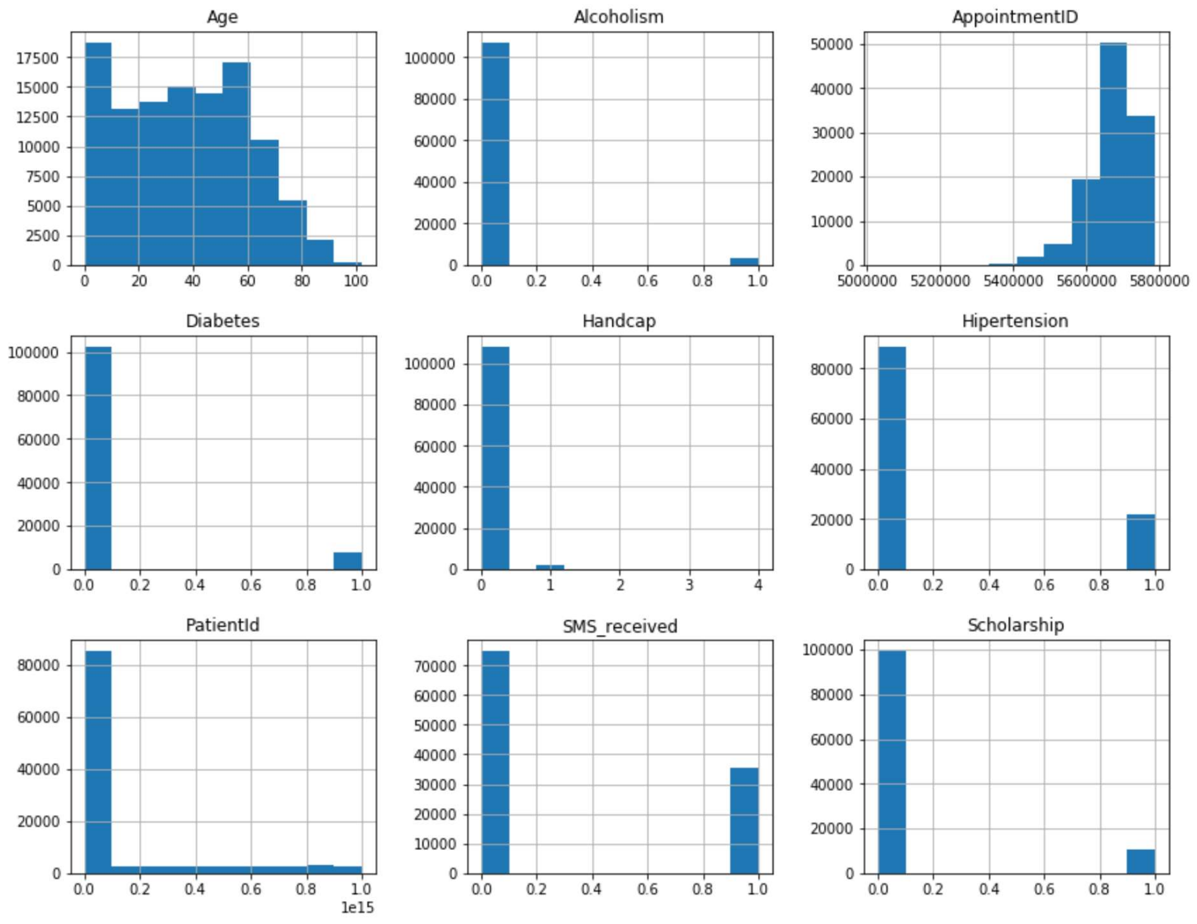


Exploratory Data Analysis

General look of the data

```
In [11]: df.hist(figsize = (14,11))
```

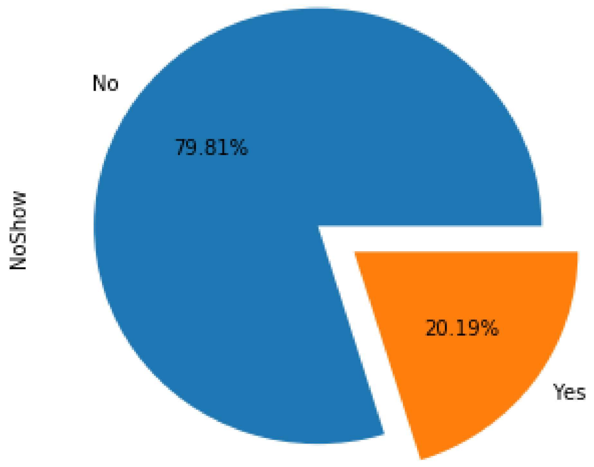
```
Out[11]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f1d19190f98>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7f1d190a7940>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7f1d1905fe80>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x7f1d19025400>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7f1d18fc9208>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7f1d18fc9278>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x7f1d18f442b0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7f1d18efd1d0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7f1d18f371d0>]], d
         type=object)
```



most of patient between 0 and 60
around 20% have hypertension
most of them do not have diabetes or any chronic disease
around half of theme received SMS

In [17]:
```
df.rename(columns={'No-show':'NoShow'}, inplace=True)
```

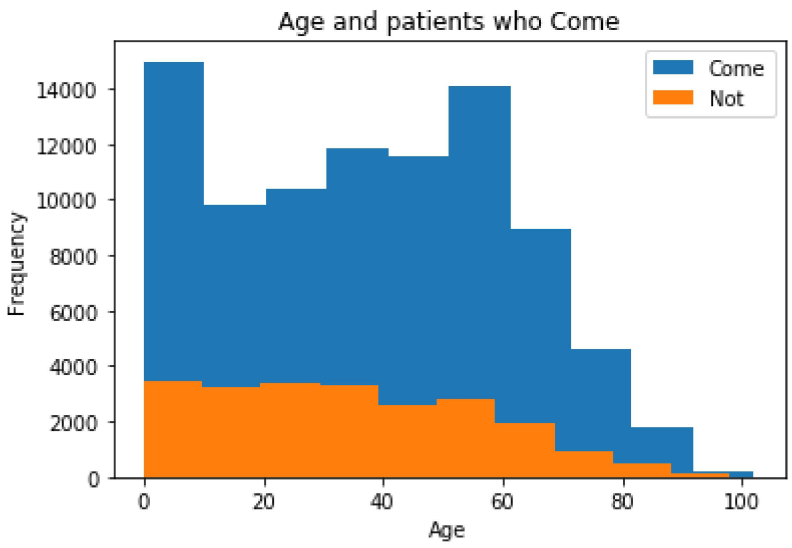
In [13]:
```
df.NoShow.value_counts().plot.pie(figsize=(5,5), autopct='%.2f%%', explode=(0,
.2))
plt.show()
```



Around 20% not come to their appointments

Research Question 1: Is Age associated with No Shows?

In [14]:
```
Come = (df.NoShow == 'No')
Not = (df.NoShow == 'Yes')
df['Come'] = Come
df['Not'] = Not
df[Come].Age.plot.hist()
df[Not].Age.plot.hist()
plt.xlabel("Age")
plt.title("Age and patients who Come")
plt.legend(['Come', 'Not'])
plt.show()
print('Come Mean Age:{:.2f}'.format(df[Come].Age.mean()))
print('Not Mean Age:{:.2f}'.format(df[Not].Age.mean()))
```



```
Come Mean Age:37.79
Not Mean Age:34.31
```

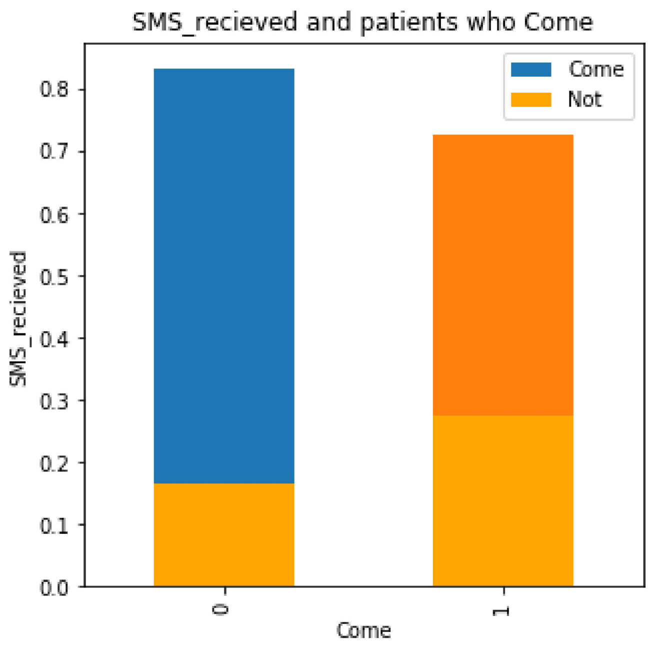
May be age affect a little

Research Question 2: Are SMS notifications associated with No Shows?

Look strange people who received message not Come compare to people who do not!

In [40]:
```
df.groupby('SMS_received')['Come'].mean().plot(kind='bar',figsize=(5,5));
df.groupby('SMS_received')['Not'].mean().plot(kind='bar',figsize=(5,5),color=
'Orange');
plt.xlabel("Come")
plt.ylabel("SMS_recieved")
plt.title("SMS_recieved and patients who Come")
plt.legend()
```

Out[40]:
```
<matplotlib.legend.Legend at 0x7f1d18cb5518>
```



Conclusions

there was a small difference in average age between patients who missed their appointments

Look strange people who received message not Come compare to people who do not!

Limitations

all of data comapare to SMS_received column like chronic disease most people do not have it ,so i think maybe an improvment to have more columns such as SMS_received

Submitting your Project

> **Tip**: Before you submit your project, you need to create a .html or .pdf version of this notebook in the workspace here. To do that, run the code cell below. If it worked correctly, you should get a return code of 0, and you should see the generated .html file in the workspace directory (click on the orange Jupyter icon in the upper left).
>
> **Tip**: Alternatively, you can download this report as .html via the **File** > **Download as** submenu, and then manually upload it into the workspace directory by clicking on the orange Jupyter icon in the upper left, then using the Upload button.
>
> **Tip**: Once you've done this, you can submit your project by clicking on the "Submit Project" button in the lower right here. This will create and submit a zip file with this .ipynb doc and the .html or .pdf version you created. Congratulations!

In [16]:
```
from subprocess import call
call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset-Copy2.ipynb'])
```

Out[16]:
```
0
```

In [ ]: