Project: Investigate a Dataset (Replace this with something more specific!)

Table of Contents

Introduction

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.

The questions that will be answered will be two typical sterotypes - 1. Does older people tned to book and plan eariler than younger people? And 2. Are people who frequent doctors more healthy?

In [1]:

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

Data Wrangling

General Properties

In [2]:

```
df = pd.read_csv('noshowappointments-may-2016.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	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
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	1	1

Data Columns Explained

AppointmentID - Identification of each appointment

Gender = Male or Female . Female is the greater proportion, woman takes way more care of they health in comparison to man.

ScheduledDay = The day someone called or registered the appointment, this is before appointment of course.

AppointmentDay = The day of the actuall appointment, when they have to visit the doctor.

Age = How old is the patient.

Neighbourhood = Where the appointment takes place.

Scholarship = Ture of False . Observation, this is a broad topic, consider reading this article
https://en.wikipedia.org/wiki/Bolsa_Fam%C3%ADlia

Hipertension = True or False

Diabetes = True or False

Alcoholism = True or False

Handcap = True or False SMS_received = 1 or more messages sent to the patient.

No-show = True or False.

In [4]:

```
df.shape
```

Out[4]:

```
(110527, 14)
```

In [5]:

```
df.describe()
```

Out[5]:

	PatientId	AppointmentID	Age	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS
count	1.105270e+05	1.105270e+05	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000	1105
mean	1.474963e+14	5.675305e+06	37.088874	0.098266	0.197246	0.071865	0.030400	0.022248	
std	2.560949e+14	7.129575e+04	23.110205	0.297675	0.397921	0.258265	0.171686	0.161543	
min	3.921784e+04	5.030230e+06	-1.000000	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	
50%	3.173184e+13	5.680573e+06	37.000000	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	
max	9.999816e+14	5.790484e+06	115.000000	1.000000	1.000000	1.000000	1.000000	4.000000	

In [6]:

```
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
```

Data Cleaning (Replace this with more specific notes!)

In [7]:

```
df.rename(columns={'No-show': 'no_show'}, inplace= True)
df.rename(columns=lambda x: x.strip().lower().replace(" ",""), inplace=True)
```

Rename No-show to no_show and renaming every column to delete space and all lower case

In [9]:

```
def convert_to_datetime():
    df['scheduledday'] = pd.to_datetime(df['scheduledday'])
    df['appointmentday'] = pd.to_datetime(df['appointmentday'])
    return df.head(1)
convert_to_datetime()
```

Out[9]:

	patientid	appointmentid	gender	scheduledday	appointmentday	age	neighbourhood	scholarship	hipertension	diabetes	alc
0	2.987250e+13	5642903	F	2016-04-29 18:38:08	2016-04-29	62	JARDIM DA PENHA	0	1	0	

function for converting date time from str to datetime

In [10]:

```
df['scheduleddate'] = [d.date() for d in df['scheduledday']]
df['scheduledtime'] = [d.time() for d in df['scheduledday']]
df['appointmentdate'] = [d.date() for d in df['appointmentday']]
df['appointmenttime'] = [d.time() for d in df['appointmentday']]
df.drop(['scheduledday','appointmentday'], axis=1, inplace=True)
```

Change the ScheduledDay and AppointmentDay from str to date time format then split ScheduledDay and AppointmentDay into sperate date and time colums ScheduledDate, ScheduledTime, AppointmentDate, AppointmentTime.

In [11]:

```
df['no_show']= df['no_show'].map({'Yes':1, 'No':0}).astype(int)
```

Replace yes and no with 1 and two for no_show

In [12]:

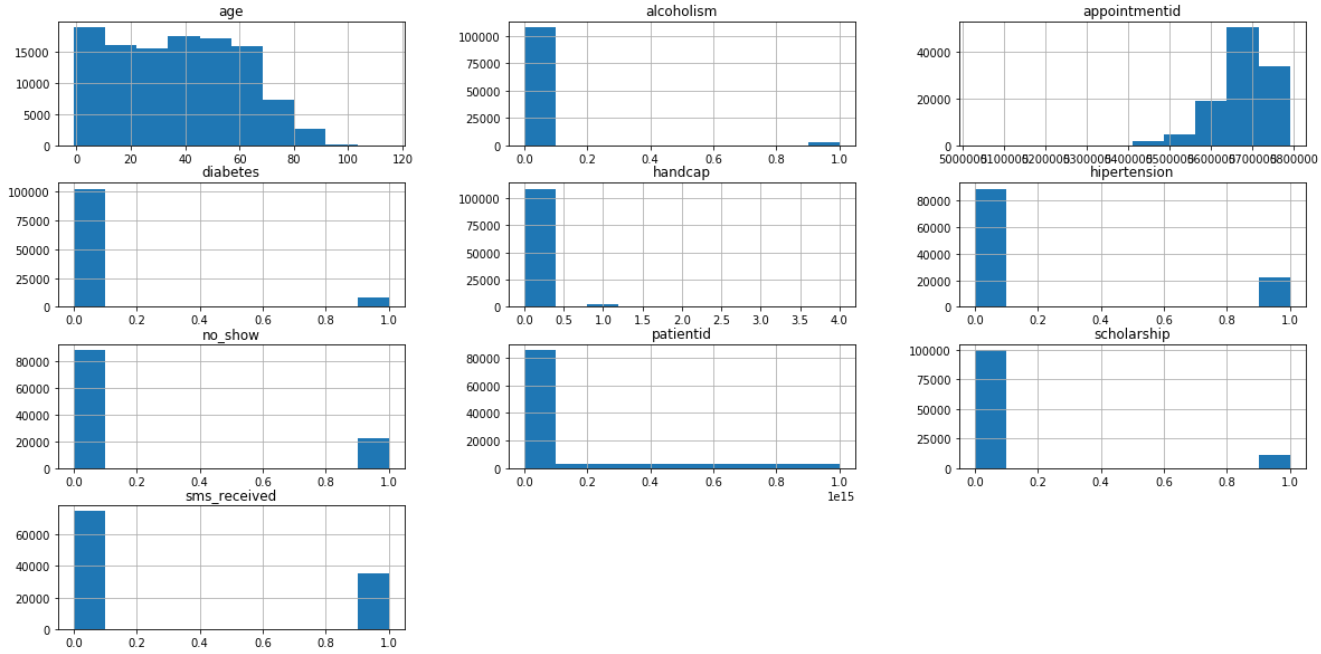
```
df.head(5)
```

Out[12]:

	patientid	appointmentid	gender	age	neighbourhood	scholarship	hipertension	diabetes	alcoholism	handcap	sms_received
0	2.987250e+13	5642903	F	62	JARDIM DA PENHA	0	1	0	0	0	0
1	5.589978e+14	5642503	M	56	JARDIM DA PENHA	0	0	0	0	0	0
2	4.262962e+12	5642549	F	62	MATA DA PRAIA	0	0	0	0	0	0
3	8.679512e+11	5642828	F	8	PONTAL DE CAMBURI	0	0	0	0	0	0
4	8.841186e+12	5642494	F	56	JARDIM DA PENHA	0	1	1	0	0	0

In [13]:

```
df.hist(figsize=(20,10));
```



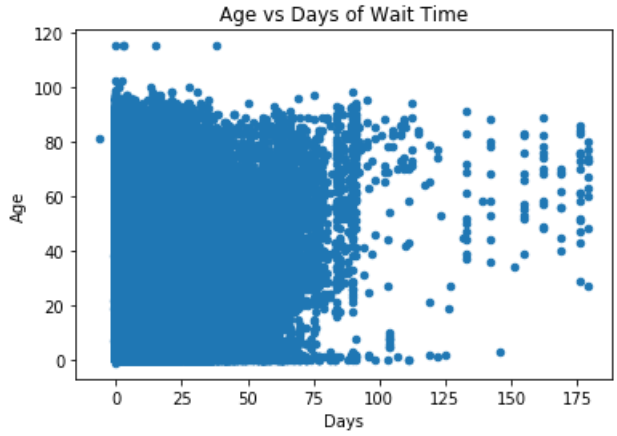
Here we can start to see some trends.

Exploratory Data Analysis

Does older people schedual their appoint more in advanced?

In [14]:

```
df['waitdays'] = df.appointmentdate - df.scheduleddate
df.waitdays = df.waitdays.dt.days
df.plot(x='waitdays', y='age', kind='scatter')
plt.title('Age vs Days of Wait Time')
plt.xlabel('Days')
plt.ylabel('Age');
```

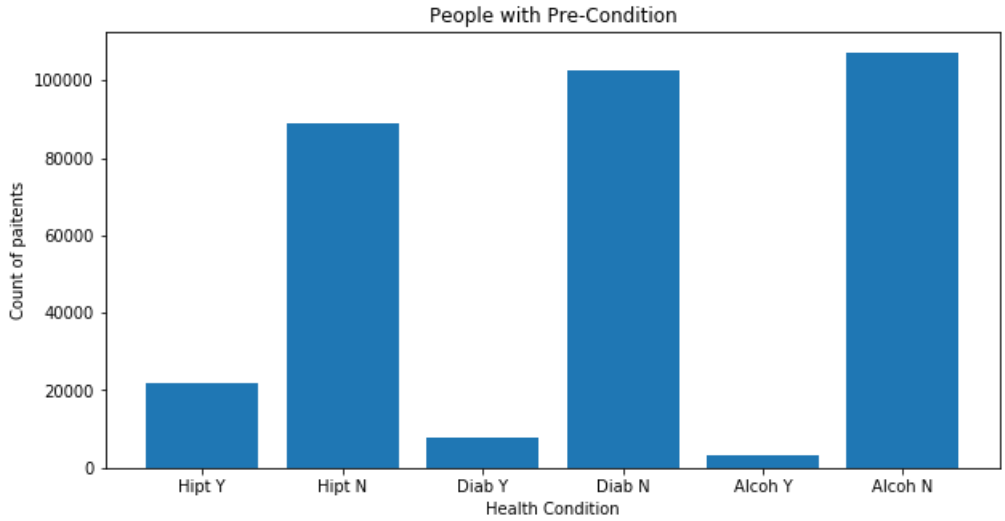


Here we have a scatter plot showing the amount of time people schedual their appointment in advance compare to their age. Based on this information, older people does not schedual their time more in advanced compared to younger people.

How healthy are the people going to the appointments?

In [16]:

```
hipertension_yes = df.query('hipertension == 1')['age'].count()
hipertension_no = df.query('hipertension == 0')['age'].count()
diabetes_yes = df.query('diabetes == 1')['age'].count()
diabetes_no = df.query('diabetes == 0')['age'].count()
alcoholism_yes = df.query('alcoholism == 1')['age'].count()
alcoholism_no = df.query('alcoholism == 0')['age'].count()
locations = [1, 2, 3, 4, 5, 6]
heights = [hipertension_yes, hipertension_no, diabetes_yes, diabetes_no, alcoholism_yes, alcoholism
_no]
labels = ['Hipt Y', 'Hipt N', 'Diab Y', 'Diab N', 'Alcoh Y', 'Alcoh N']
plt.bar(locations, heights, tick_label=labels)
plt.title('People with Pre-Condition')
plt.ylabel('Count of paitents')
plt.xlabel('Health Condition')
plt.rcParams["figure.figsize"] = [10,5]
```



By counting people with different conditions, this chart is able to be drawn. It appeares that majority amount of people does not have any serious pre-conditions.

Conclusions

After looking at paitnet data provided to us from Brazil's medical appointments it has clearly answered both of our answers. First is that older people do no have tendency to book appointments any eariler than younger people. Second people with no serious health problems tends to go to the doctors more than those that have serious health problems.

Limitations

This data doesn't provide enough information for accurate prediction of if paitents will show up. For example the distance between the paitnet and the clinc and the reason paitent booked the apointment all are significant to predicting if paitent will show up for appointment or not.

In [ ]:

```
```