Untitled

August 21, 2022

1 Project: Investigate a Dataset - [Dataset-name]

1.1 Table of Contents

Introduction
Data Wrangling
Exploratory Data Analysis
Conclusions
Introduction

1.1.1 Dataset Description

Tip: In this section of the report, provide a brief introduction to the dataset you've selected/downloaded for analysis. Read through the description available on the homepage-links present here. List all column names in each table, and their significance. In case of multiple tables, describe the relationship between tables.

1.1.2 Question(s) for Analysis

Tip: Clearly state one or more questions that you plan on exploring over the course of the report. You will address these questions in the **data analysis** and **conclusion** sections. Try to build your report around the analysis of at least one dependent variable and three independent variables. If you're not sure what questions to ask, then make sure you familiarize yourself with the dataset, its variables and the dataset context for ideas of what to explore.

Tip: Once you start coding, use NumPy arrays, Pandas Series, and DataFrames where appropriate rather than Python lists and dictionaries. Also, **use good coding practices**, such as, define and use functions to avoid repetitive code. Use appropriate comments within the code cells, explanation in the mark-down cells, and meaningful variable names.

In [2]:
```
# 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
```

```
%matplotlib inline
from pandas import read_excel
# 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
```

In [3]:
```
#df = read_excel(noshowappointments-kagglev2-may-2016, sheet_name = my_sheet)
#df = read_excel(file_name, sheet_name = my_sheet)
#dfs = pd.read_excel(xlsx_file, sheetname="sheet1")
#df = pd.read_excel(noshowappointments-kagglev2-may-2016.xlsx, index_col=0)
df = pd.read_csv("noshowappointments-kagglev2-may-2016.csv")
#df = pd.read_csv("tmdb-movies.csv")
```

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

Out[4]:

	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 [5]:
```
# to get the dimension of our data
df.shape
```

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

In [6]:
```
#Display the the datatype of each column and the number of values on each column
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
```

In [7]:
```
##Statistical description of the data
df.describe()
```

Out[7]:

	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	0.000000
max	9.999816e+14	5.790484e+06	115.000000	1.000000	1.000000	1.000000	1.000000	4.000000	1.000000

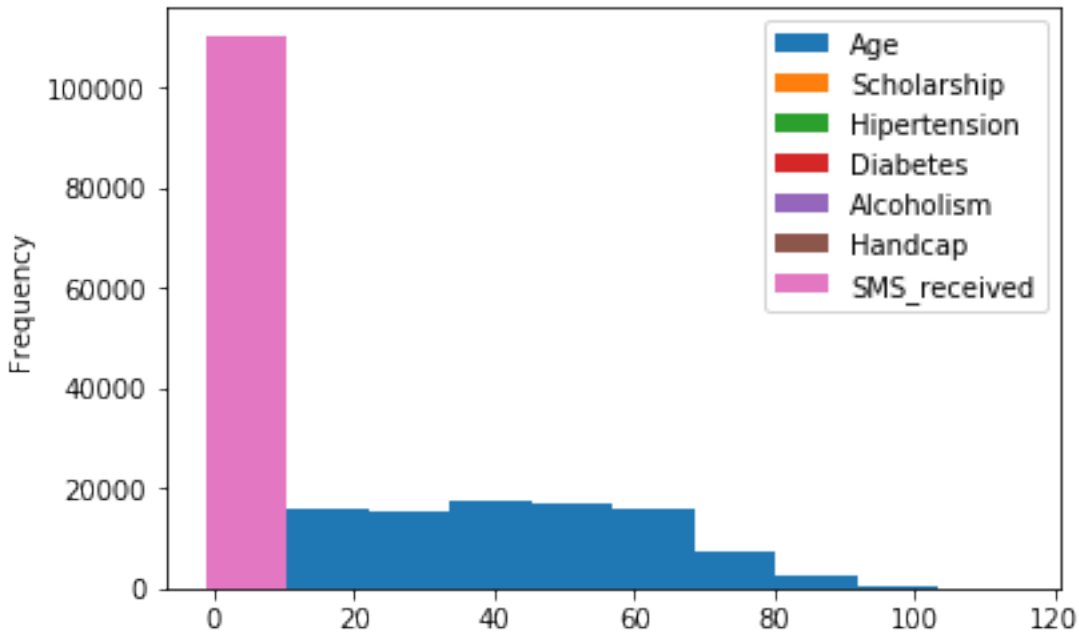
```
In [8]: ##Statistical description of the data
        df.duplicated().sum()

Out[8]: 0

In [9]: df.nunique()

Out[9]: PatientId          62299
        AppointmentID     110527
        Gender                 2
        ScheduledDay      103549
        AppointmentDay        27
        Age                  104
        Neighbourhood         81
        Scholarship            2
        Hipertension           2
        Diabetes               2
        Alcoholism             2
        Handcap                5
        SMS_received           2
        No-show                2
        dtype: int64

In [ ]:

In [10]: h = [2, 4, 7, 9, 0]
         h.pop(0)

Out[10]: 2

In [11]: df['ScheduledDay'][0]

Out[11]: '2016-04-29T18:38:08Z'

In [ ]:

In [ ]:

In [12]: df.drop('PatientId', axis =1, inplace = True)
         #df.drop("column_name", axis=1, inplace=True)

In [13]: df.drop('AppointmentID', axis = 1, inplace = True)

In [14]: df.drop('ScheduledDay', axis = 1, inplace = True)

In [15]: df.drop('AppointmentDay', axis = 1, inplace = True)

In [16]: df.head()
```

Out[16]:

	Gender	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	F	62	JARDIM DA PENHA	0	1	0	0	0	0	No
1	M	56	JARDIM DA PENHA	0	0	0	0	0	0	No
2	F	62	MATA DA PRAIA	0	0	0	0	0	0	No
3	F	8	PONTAL DE CAMBURI	0	0	0	0	0	0	No
4	F	56	JARDIM DA PENHA	0	1	1	0	0	0	No

In [17]:
```
df.plot(kind = 'hist')
```

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7f726863d128>

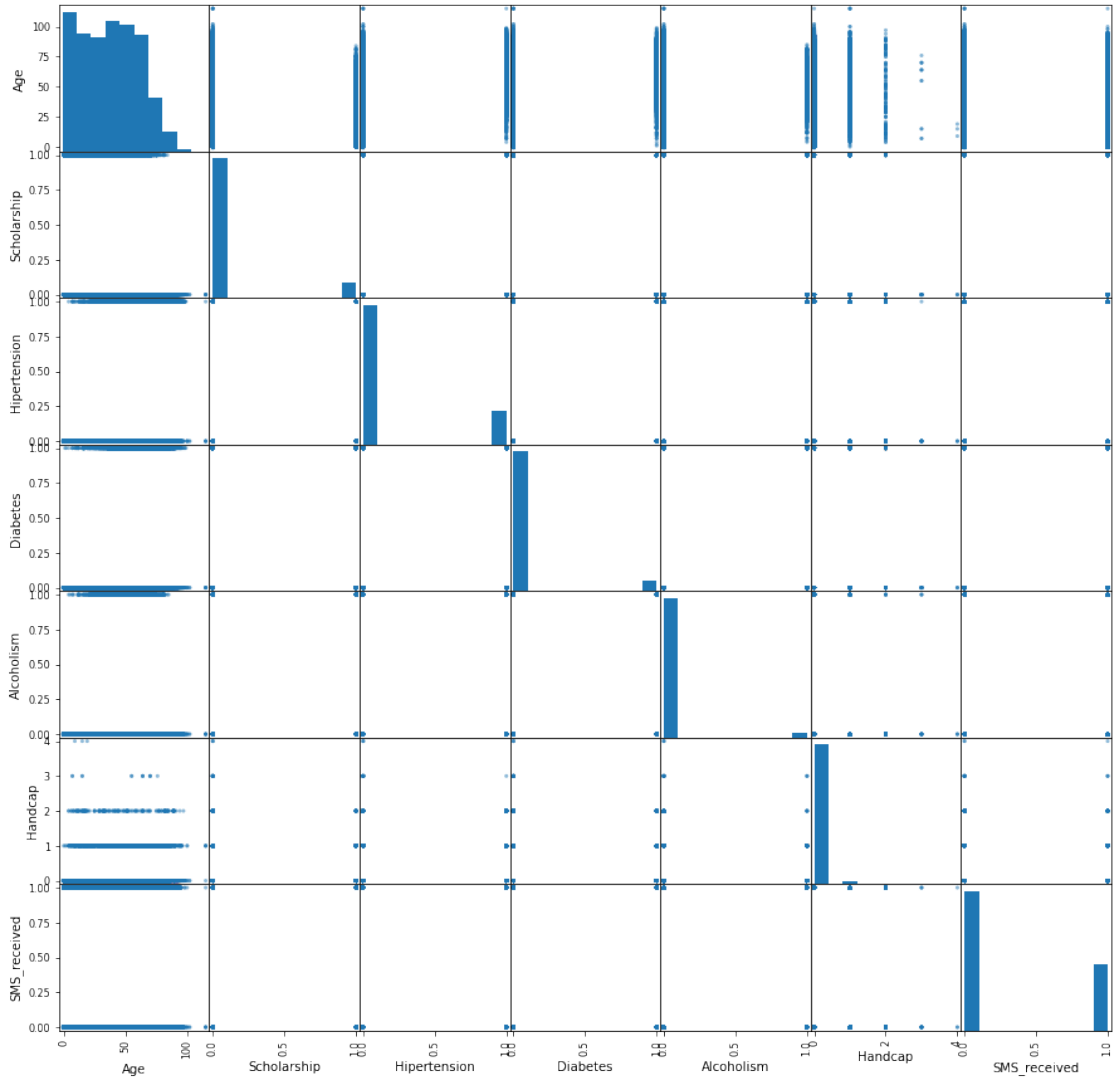


In [18]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 10 columns):
Gender          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: int64(7), object(3)
memory usage: 8.4+ MB
```

```
In [19]: pd.plotting.scatter_matrix(df, figsize = (15,15));
```



1.2 The plots above shows that majority of the patients were reminded of their appointments with a text message

Majority of the children did not receive a text message Further data wrangling can be done on the 'AppointmentDay' feature

In [ ]: