Investigate_a_Dataset

February 1, 2019

> **Tip**: Welcome to the Investigate a Dataset project! You will find tips in quoted sections like this to help organize your approach to your investigation. Before submitting your project, it will be a good idea to go back through your report and remove these sections to make the presentation of your work as tidy as possible. First things first, you might want to double-click this Markdown cell and change the title so that it reflects your dataset and investigation.

1 Project: Investigate a Dataset (no-shows at medical appointments in Brazil-may-2016)

1.1 Table of Contents

> **Tip**: In this section of the report, provide a brief introduction to the dataset you've selected for analysis. At the end of this section, describe the questions that you plan on exploring over the course of the report. 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.
>
> If you haven't yet selected and downloaded your data, make sure you do that first before coming back here. In order to work with the data in this workspace, you also need to upload it to the workspace. To do so, click on the jupyter icon in the upper left to be taken back to the workspace directory. There should be an 'Upload' button in the upper right that will let you add your data file(s) to the workspace. You can then click on the .ipynb file name to come back here.

```
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 time to change appointment_day to adate without day time 00:00:00
```

```
import datetime
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 steps carefully and justify your cleaning decisions.

1.1.1 General Properties

In [3]:
```
# 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.
#change appointment_day to adate without day time 00:00:00
df_noshowappointments = pd.read_csv('noshowappointments-kagglev2-may-2016.csv',index_col
                                    parse_dates=["ScheduledDay", "AppointmentDay"], infe
df_noshowappointments.head()
```

Out[3]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	\
0	2.987250e+13	5642903	F	2016-04-29 18:38:08	2016-04-29	62	
1	5.589978e+14	5642503	M	2016-04-29 16:08:27	2016-04-29	56	
2	4.262962e+12	5642549	F	2016-04-29 16:19:04	2016-04-29	62	
3	8.679512e+11	5642828	F	2016-04-29 17:29:31	2016-04-29	8	
4	8.841186e+12	5642494	F	2016-04-29 16:07:23	2016-04-29	56	

	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	\
0	JARDIM DA PENHA	0	1	0	0	
1	JARDIM DA PENHA	0	0	0	0	
2	MATA DA PRAIA	0	0	0	0	
3	PONTAL DE CAMBURI	0	0	0	0	
4	JARDIM DA PENHA	0	1	1	0	

	Handcap	SMS_received	No-show
0	0	0	No
1	0	0	No
2	0	0	No
3	0	0	No
4	0	0	No

In [4]:
```
# investigate no rows and no of columns
df_noshowappointments.shape
```

Out[4]: (110527, 14)

```
In [5]: # investigate data type and missing values
        df_noshowappointments.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 datetime64[ns]
AppointmentDay    110527 non-null datetime64[ns]
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: datetime64[ns](2), float64(1), int64(8), object(3)
memory usage: 11.8+ MB
```

```
In [6]: # do some summary statistics to the data set
        df_noshowappointments.describe()
```

Out[6]:

	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
```

so dataset has no missing values

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.

Tip: Make sure that you keep your reader informed on the steps that you are taking in your investigation. Follow every code cell, or every set of related code cells, with a markdown cell to describe to the reader what was found in the preceding cell(s). Try to make it so that the reader can then understand what they will be seeing in the following cell(s).

1.1.2 Data Cleaning (Replace this with more specific notes!)

In [7]:
```
# 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.
#drop non-important columns for my analysis
df_noshowappointments.drop(['PatientId','AppointmentID','Handcap'], axis=1, inplace = Tr
```

In [8]:
```
# I do not want to replace values like  0 by no and 1 by yes in rows to do easily descri
#renaming camelcase with lowercase and snakecase
df_noshowappointments.rename(columns={'Hipertension':'hypertension','ScheduledDay':'sche
                                      'Diabetes':'diabetes', 'Alcoholism':'alcoholism','
                                      'sms_received','Scholarship':'scholarship', 'Age':
                                      'appointment_day','No-show':'no_show','Neighbourho
                                      'neighbourhood', 'Gender':'gender'}, inplace=True)

df_noshowappointments.head()
```

Out[8]:

	gender	scheduled_day	appointment_day	age	neighbourhood	\
0	F	2016-04-29 18:38:08	2016-04-29	62	JARDIM DA PENHA	
1	M	2016-04-29 16:08:27	2016-04-29	56	JARDIM DA PENHA	
2	F	2016-04-29 16:19:04	2016-04-29	62	MATA DA PRAIA	
3	F	2016-04-29 17:29:31	2016-04-29	8	PONTAL DE CAMBURI	
4	F	2016-04-29 16:07:23	2016-04-29	56	JARDIM DA PENHA	

	scholarship	hypertension	diabetes	alcoholism	sms_received	no_show

```
0            0            1            0            0            0       No
1            0            0            0            0            0       No
2            0            0            0            0            0       No
3            0            0            0            0            0       No
4            0            1            1            0            0       No
```

Exploratory Data Analysis

Tip: Now that you've trimmed and cleaned your data, you're ready to move on to exploration. Compute statistics and create visualizations with the goal of addressing the research questions that you posed in the Introduction section. It is recommended that you be systematic with your approach. Look at one variable at a time, and then follow it up by looking at relationships between variables.

1.1.3 Research Question 1 (by the mean what are the types of different diseases?what is the most disease present in brazil in 2016 ? and what is the most age suffer from that disease?)

In [9]:
```
# do summary statistics to the data set
df_noshowappointments.describe()
```

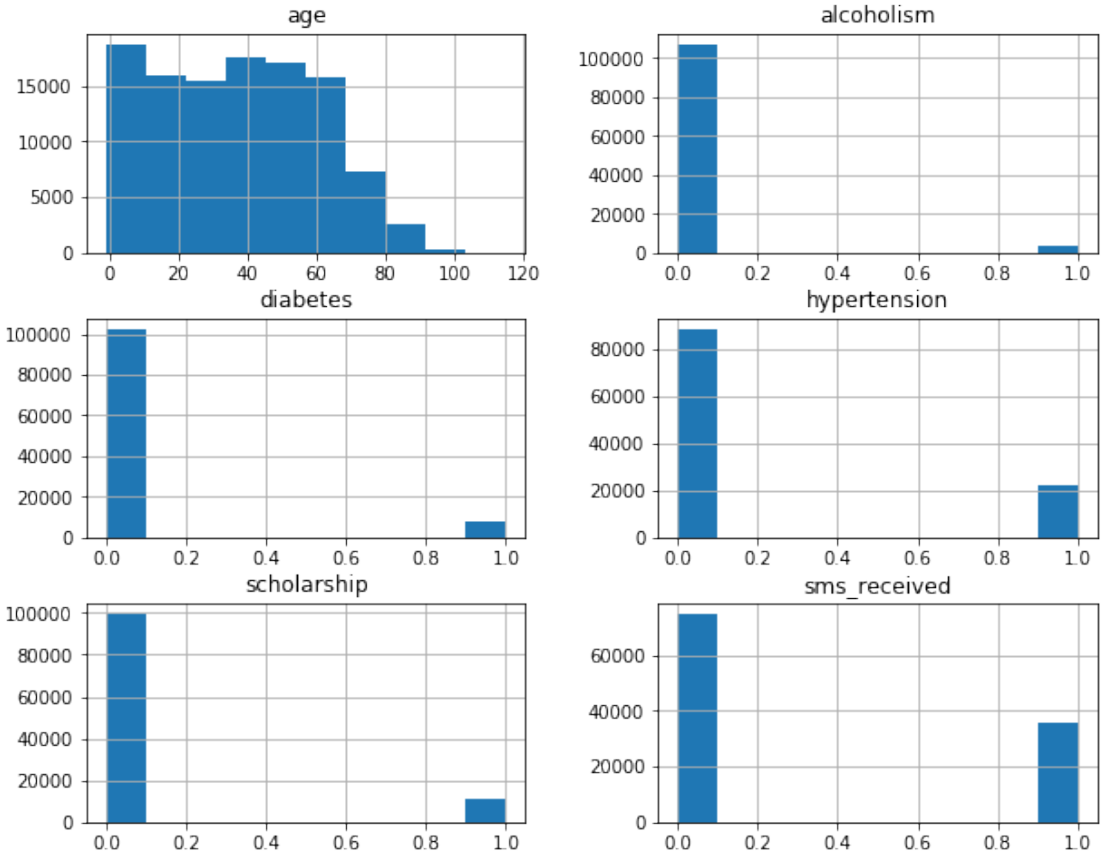
Out[9]:

	age	scholarship	hypertension	diabetes	alcoholism	sms_received
count	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000
mean	37.088874	0.098266	0.197246	0.071865	0.030400	0.321026
std	23.110205	0.297675	0.397921	0.258265	0.171686	0.466873
min	-1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	18.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	37.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	55.000000	0.000000	0.000000	0.000000	0.000000	1.000000
max	115.000000	1.000000	1.000000	1.000000	1.000000	1.000000

In [10]:
```
# Use this, and more code cells, to explore your data. Don't forget to add
#   Markdown cells to document your observations and findings.
df_noshowappointments.hist(figsize=(10,8))
```

Out[10]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f87000402e8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f86ffb9f668>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f86ffdfbac8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f86ffdb3b38>],
```

```
[<matplotlib.axes._subplots.AxesSubplot object at 0x7f86ffd68470>,
 <matplotlib.axes._subplots.AxesSubplot object at 0x7f86ffd684a8>]], dtype=objec
```



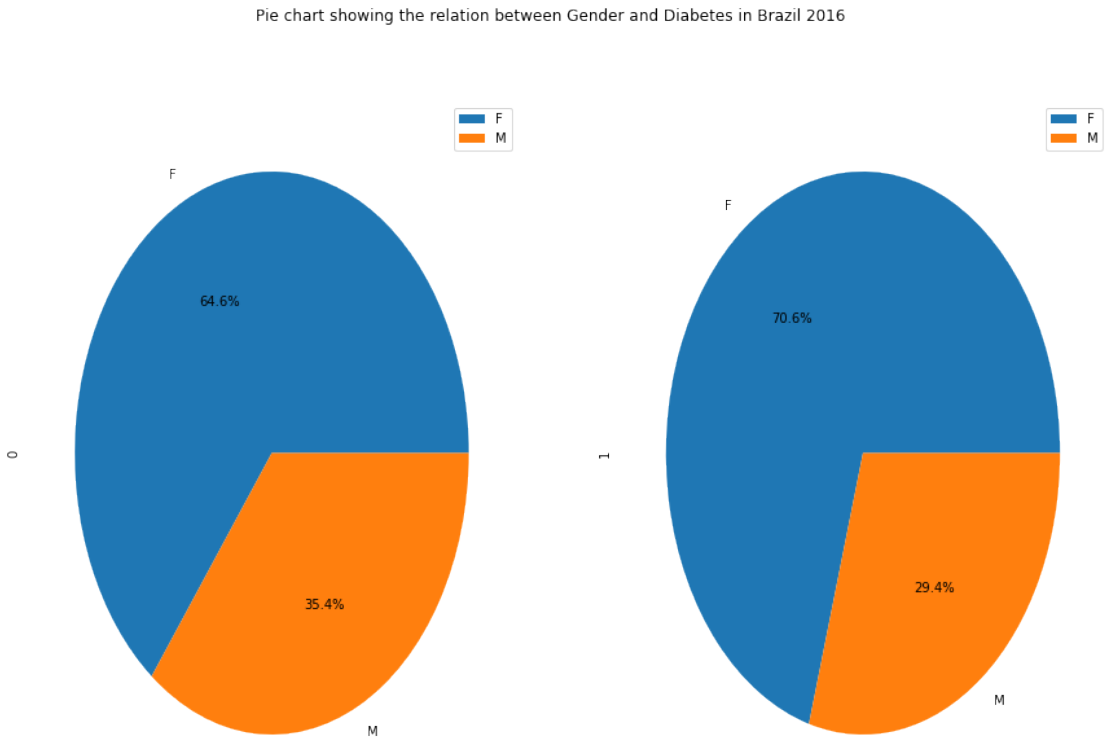
by the mean the types of different diseases are hypertension then diabetes then alcoholism the most disease present in brazil in 2016 is hypertension and the most age suffer from that disease 37 years old

1.1.4 Research Question 2 (Is there a relation between disease and gender in Brazil 2016?)

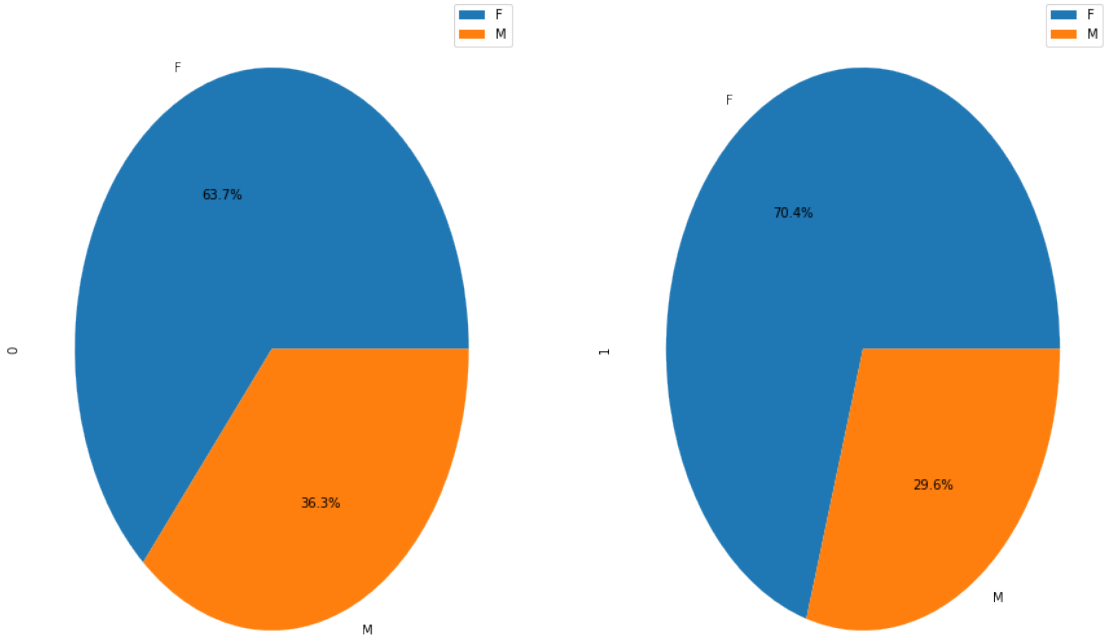
```
In [11]: # Continue to explore the data to address your additional research
         #   questions. Add more headers as needed if you have more questions to
         #   investigate.
         grouping_gender_diabetes = df_noshowappointments.groupby(['gender','diabetes'])
         grouping_gender_diabetes.size().unstack().plot(kind='pie', subplots = True,
                                                        autopct='%1.1f%%', figsize=(15,10)
                                                        ,title = 'Pie chart showing the relation be
         grouping_gender_hypertension = df_noshowappointments.groupby(['gender','hypertension'])
         grouping_gender_hypertension.size().unstack().plot(kind='pie', subplots = True,
                                                        autopct='%1.1f%%', figsize=(15,10)
                                                        ,title = 'Pie chart showing the relation be
         grouping_gender_alcoholism = df_noshowappointments.groupby(['gender','alcoholism'])
```

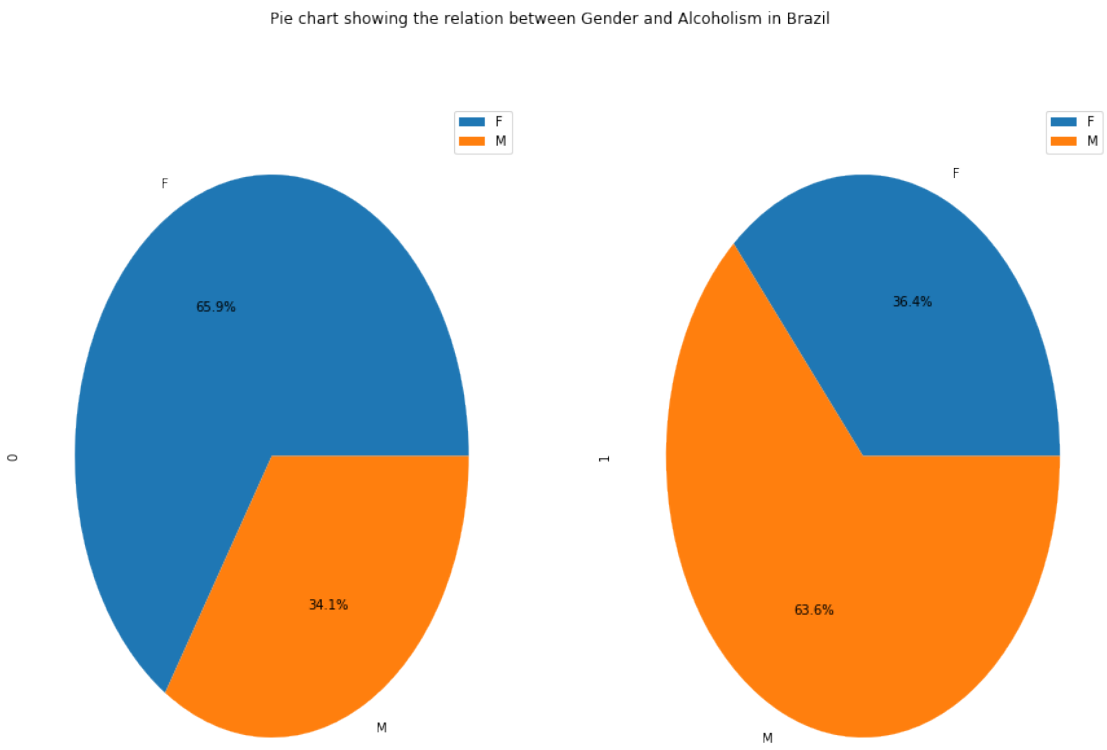
```
grouping_gender_alcoholism.size().unstack().plot(kind='pie', subplots = True,
                                                 autopct='%1.1f%%', figsize=(15,10)
                                                  ,title = 'Pie chart showing the relation b
```

```
Out[11]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x7f86ff9e0080>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x7f86ff98af98>], dtype=object)
```



Pie chart showing the relation between Gender and Hypertension in Brazil in 2016

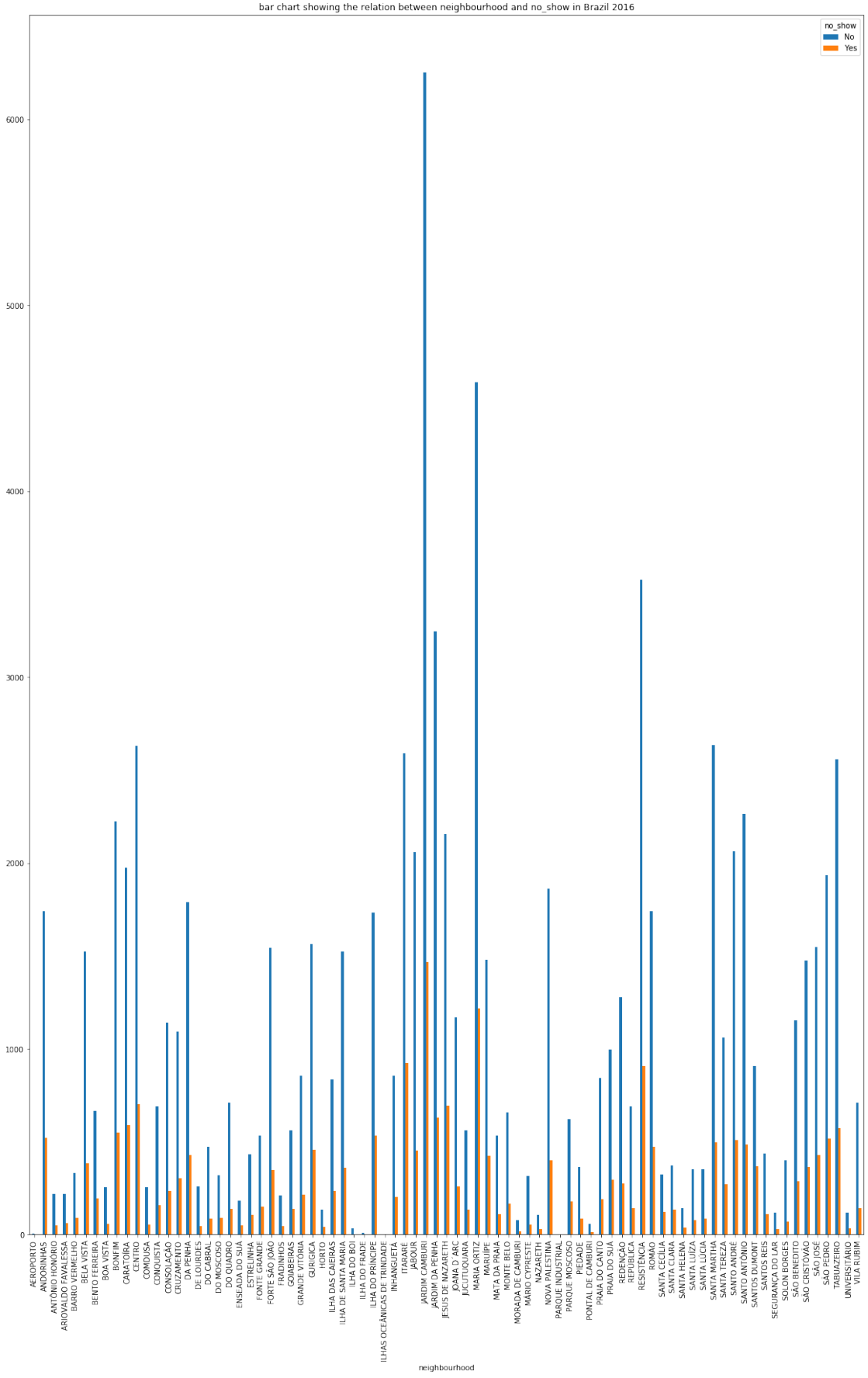




What is the most area has most numbers of no_show?

```
In [12]: grouping_gender_diabetes = df_noshowappointments.groupby(['neighbourhood', 'no_show'])
         grouping_gender_diabetes.size().unstack().plot(kind='bar', figsize=(20,30),title = 'bar
```

```
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f86ffca0128>
```

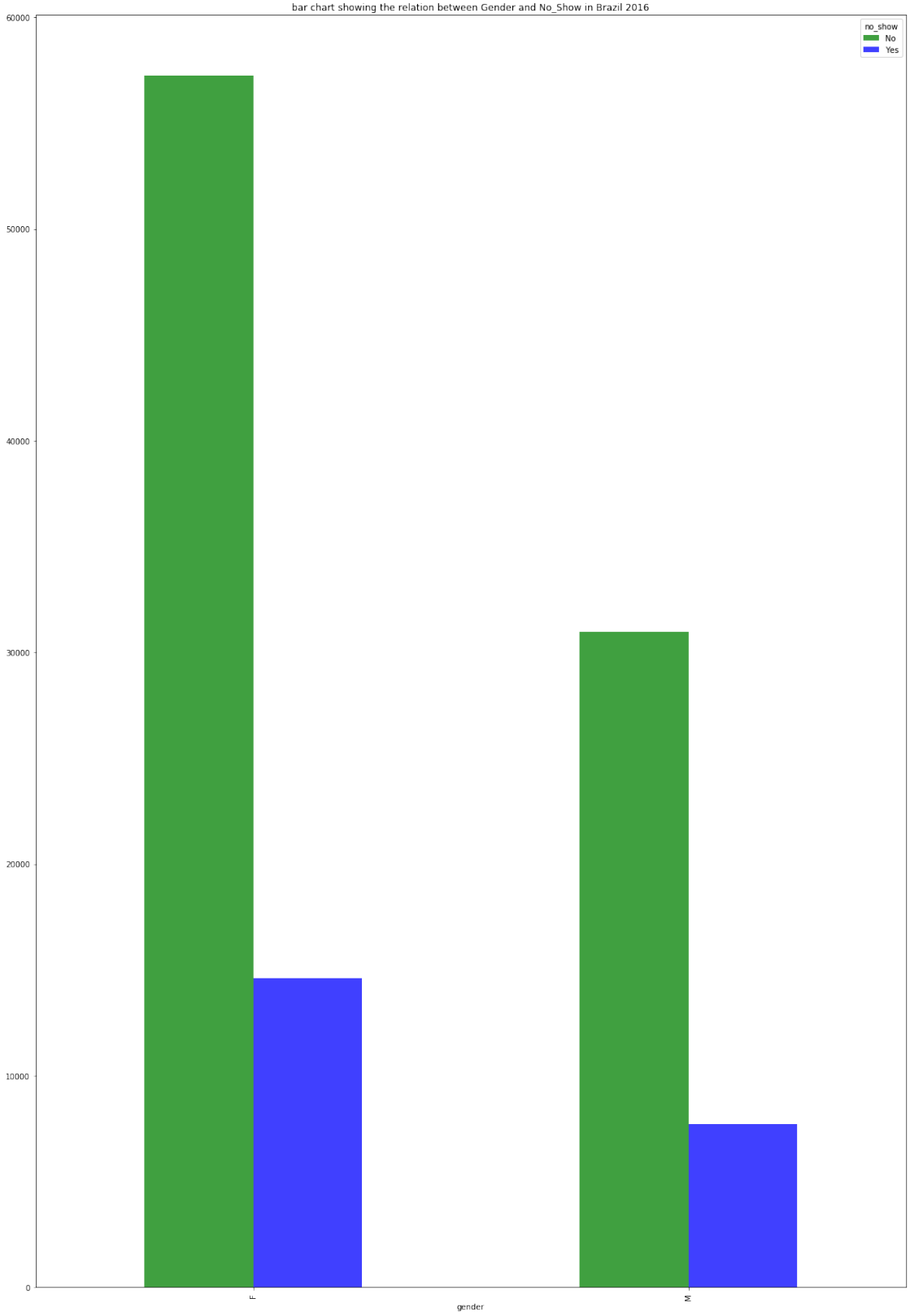

bar chart showing the relation between neighbourhood and no_show in Brazil 2016
no_show
No
Yes
6000
5000
4000
3000
2000
1000
0
AEROPORTO
ANDORINHAS
ANTÔNIO HONÓRIO
ARIOVALDO FAVALESSA
BARRO VERMELHO
BELA VISTA
BENTO FERREIRA
BOA VISTA
BONFIM
CARATOÍRA
CENTRO
COMDUSA
CONQUISTA
CONSOLAÇÃO
CRUZAMENTO
DA PENHA
DE LOURDES
DO CABRAL
DO MOSCOSO
DO QUADRO
ENSEADA DO SUÁ
ESTRELINHA
FONTE GRANDE
FORTE SÃO JOÃO
FRADINHOS
GOIABEIRAS
GRANDE VITÓRIA
GURIGICA
HORTO
ILHA DAS CAIEIRAS
ILHA DE SANTA MARIA
ILHA DO BOI
ILHA DO FRADE
ILHA DO PRÍNCIPE
ILHAS OCEÂNICAS DE TRINDADE
INHANGUETÁ
ITARARÉ
JABOUR
JARDIM CAMBURI
JARDIM DA PENHA
JESUS DE NAZARETH
JOANA D´ARC
JUCUTUQUARA
MARIA ORTIZ
MARUÍPE
MATA DA PRAIA
MONTE BELO
MORADA DE CAMBURI
MÁRIO CYPRESTE
NAZARETH
NOVA PALESTINA
PARQUE INDUSTRIAL
PARQUE MOSCOSO
PIEDADE
PONTAL DE CAMBURI
PRAIA DO CANTO
PRAIA DO SUÁ
REDENÇÃO
REPÚBLICA
RESISTÊNCIA
ROMÃO
SANTA CECÍLIA
SANTA CLARA
SANTA HELENA
SANTA LUÍZA
SANTA LÚCIA
SANTA MARTHA
SANTA TEREZA
SANTO ANDRÉ
SANTO ANTÔNIO
SANTOS DUMONT
SANTOS REIS
SEGURANÇA DO LAR
SOLON BORGES
SÃO BENEDITO
SÃO CRISTÓVÃO
SÃO JOSÉ
SÃO PEDRO
TABUAZEIRO
UNIVERSITÁRIO
VILA RUBIM
neighbourhood

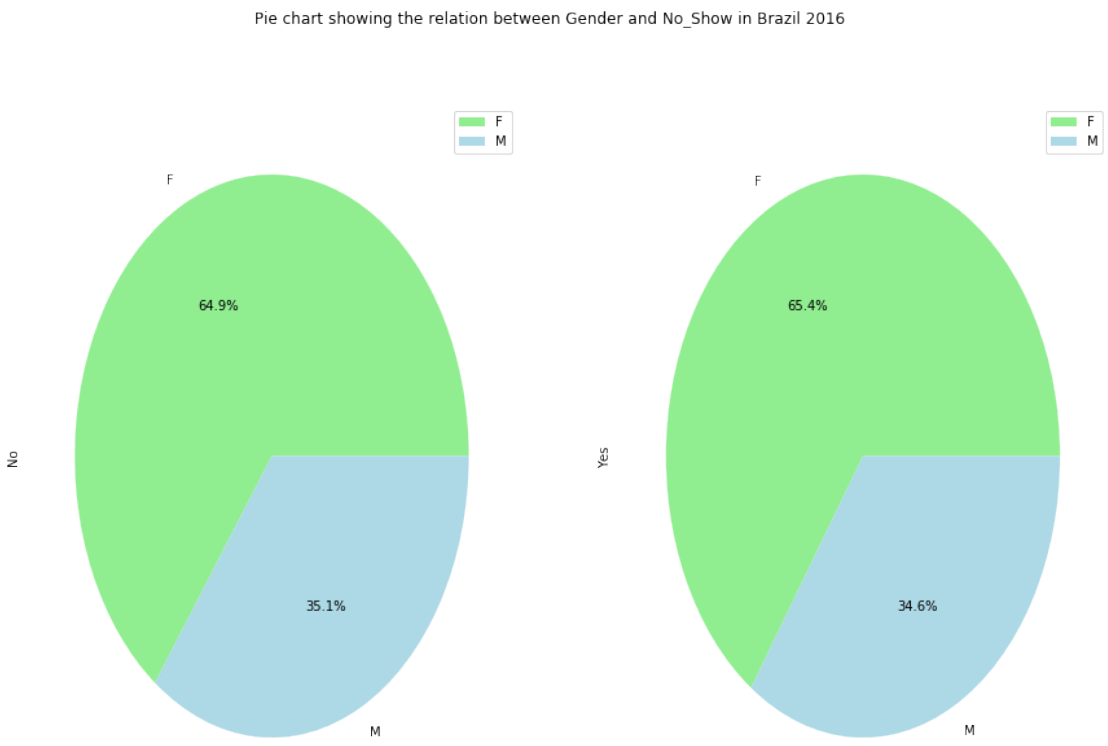
who has the most no_show males or females?

```
In [41]: grouping_gender_no_show = df_noshowappointments.groupby(['gender','no_show'])
         grouping_gender_no_show.size().unstack().plot(kind='bar', figsize=(20,30),colors=('g','
         grouping_gender_no_showno_show = df_noshowappointments.groupby(['gender','no_show'])
         grouping_gender_no_show.size().unstack().plot(kind='pie', subplots = True,colors=('ligh
                                                       autopct='%1.1f%%', figsize=(15,10)
                                                        ,title = 'Pie chart showing the relation b

/opt/conda/lib/python3.6/site-packages/pandas/plotting/_core.py:179: UserWarning: 'colors' is be
  warnings.warn(("'colors' is being deprecated. Please use 'color'"


Out[41]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x7f86fa1bd2e8>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x7f86fbd1d400>], dtype=object)
```


bar chart showing the relation between Gender and No_Show in Brazil 2016
no_show
No
Yes
60000
50000
40000
30000
20000
10000
0
F
M
gender



Conclusions By the mean the types of different diseases are hypertension then diabetes then alcoholism the most disease present in brazil in 2016 is hypertension and the most age suffer from that disease 37 years old. males have alcoholism issue more than females in brazil so medical advice about alcoholism will mostly cosidered for males. hypertension is the most disease present in brazil 2016 and most age suffer from diseases 37 years old, that age needs to start medical care and attention. women in brazil 2016 suffer from hypertension and diabetes more than males so females in brazil need more medical care and to focus preventive medicine on them. most of no-show happened from females in brazil 2016 so they suffer from diseases like diabetes and hypertension more than men. men in brazil 2016 need special attention and medical care and guidance for excessive alcoholism. > **Tip**: Finally, summarize your findings and the results that have been performed. Make sure that you are clear with regards to the limitations of your exploration. If you haven't done any statistical tests, do not imply any statistical conclusions. And make sure you avoid implying causation from correlation!

> **Tip**: Once you are satisfied with your work here, check over your report to make sure that it is satisfies all the areas of the rubric (found on the project submission page at the end of the lesson). You should also probably remove all of the "Tips" like this one so that the presentation is as polished as possible.

1.2 Submitting your Project

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).

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.

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

```
Out[43]: 0
```

```
In [ ]:
```