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.
> - **No-show**: it says 'No' if the patient showed up to their appointment, and 'Yes' if they did not show up.
> - **PatientId**: The Patient Identification number in the hospital
> - **AppointmentID**: The appointment identification number in the hospital.
> - **Gender**: Tell us about the patient sex.
> - **AppointmentDay**: tells us on what day the patient show up their appointment.
> - **Age**: tells us about the age of pattient.
> - **Hipertension**: If the patient has this disease.
> - **Diabetes**: If the patient has this disease.
> - **Alcoholism**: If the patient has this disease.
> - **Handcap***: If the patient has this disease.
> - **SMS_received**: If the patient received notification for the appointment.

Question(s) for Analysis

> 1. Is the number of gender equals through the dataset ?
> 2. Is the case that a patient get scholarship will help to show up ?
> 3. What days of week patient sow up easily for they appointment ?
> 4. Is patient going to show up wether they are sick or not ?

In [86]:
```
# Use this cell to set up import statements for all of the packages that you
#   plan to use.
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
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
```

In [87]:
```
# Upgrade pandas to use dataframe.explode() function.
#!pip install --upgrade pandas==0.25.0
```

Data Wrangling

General Properties

> 1. Here we are reading our dataset to get known about it contents.
> 2. After that reload dataset with columns renamed in a way to get ease in our process
> 3. At this we can try show data sample

In [88]:
```
# Load your data and print out a few lines. Perform operations to inspect data
renamed_columns = ['patient_id', 'appointment_id', 'gender', 'scheduled_day',
       'appointment_day', 'age', 'neighbourhood', 'scholarship', 'hypertension',
       'diabetes', 'alcoholism', 'handicap', 'sms_received', 'no_show']
df = pd.read_csv('noshowappointments-kagglev2-may-2016.csv', header = 0, names = renamed
df.sample(10)
```

Out[88]:

	patient_id	appointment_id	gender	scheduled_day	appointment_day	age	neighbourhood	scholar
97130	2.444762e+14	5733857	M	2016-05-24T14:34:16Z	2016-06-07T00:00:00Z	86	JUCUTUQUARA	
89085	8.866935e+12	5751313	M	2016-05-31T09:57:10Z	2016-06-06T00:00:00Z	0	SANTO ANTÔNIO	
9086	7.972185e+13	5677525	F	2016-05-10T07:19:55Z	2016-05-10T00:00:00Z	54	BELA VISTA	
11715	1.831626e+14	5693271	F	2016-05-13T07:01:41Z	2016-05-16T00:00:00Z	61	SANTO ANDRÉ	
32198	7.638849e+13	5647368	F	2016-05-02T13:29:07Z	2016-05-10T00:00:00Z	24	MARIA ORTIZ	
6537	1.886486e+14	5719136	F	2016-05-19T10:38:53Z	2016-05-24T00:00:00Z	65	ILHA DO PRÍNCIPE	
85418	2.714528e+12	5770189	M	2016-06-03T09:52:46Z	2016-06-03T00:00:00Z	17	DO MOSCOSO	
10160	4.529479e+13	5665607	F	2016-05-05T14:54:44Z	2016-05-17T00:00:00Z	25	CRUZAMENTO	
107873	6.195752e+11	5783528	F	2016-06-07T14:06:16Z	2016-06-07T00:00:00Z	56	ILHA DE SANTA MARIA	
7410	8.722785e+11	5726547	M	2016-05-20T13:54:22Z	2016-05-30T00:00:00Z	40	REPÚBLICA	

Data info and some quick stats about all information in the dataset

> Using the following cell we can describe all our dataset With the information about the dataset we can get all columns data types, numbers of rows and columns and also if there some missing data, duplicated or incorrect data.

In [89]:
```
#   types and look for instances of missing or possibly errant data.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   patient_id       110527 non-null  float64
 1   appointment_id   110527 non-null  int64  
 2   gender           110527 non-null  object 
 3   scheduled_day    110527 non-null  object 
 4   appointment_day  110527 non-null  object 
 5   age              110527 non-null  int64  
 6   neighbourhood    110527 non-null  object 
 7   scholarship      110527 non-null  int64  
 8   hypertension     110527 non-null  int64  
 9   diabetes         110527 non-null  int64  
 10  alcoholism       110527 non-null  int64  
 11  handicap         110527 non-null  int64  
 12  sms_received     110527 non-null  int64  
 13  no_show          110527 non-null  object 
dtypes: float64(1), int64(8), object(5)
memory usage: 11.8+ MB
```

In [90]:
```
df.describe()
```

Out[90]:

	patient_id	appointment_id	age	scholarship	hypertension	diabetes	alcoholis
count	1.105270e+05	1.105270e+05	110527.000000	110527.000000	110527.000000	110527.000000	110527.0000
mean	1.474963e+14	5.675305e+06	37.088874	0.098266	0.197246	0.071865	0.0304
std	2.560949e+14	7.129575e+04	23.110205	0.297675	0.397921	0.258265	0.1716
min	3.921784e+04	5.030230e+06	-1.000000	0.000000	0.000000	0.000000	0.0000
25%	4.172614e+12	5.640286e+06	18.000000	0.000000	0.000000	0.000000	0.0000
50%	3.173184e+13	5.680573e+06	37.000000	0.000000	0.000000	0.000000	0.0000
75%	9.439172e+13	5.725524e+06	55.000000	0.000000	0.000000	0.000000	0.0000
max	9.999816e+14	5.790484e+06	115.000000	1.000000	1.000000	1.000000	1.0000

In [91]:
```
# Let's first check and correct data about the patients ages
df.loc[df['age'] < 0]
```

Out[91]:

	patient_id	appointment_id	gender	scheduled_day	appointment_day	age	neighbourhood	scholarsh
99832	4.659432e+14	5775010	F	2016-06-06T08:58:13Z	2016-06-06T00:00:00Z	-1	ROMÃO	

In [92]:
```
df.drop(df.loc[df['age'] < 0].index, inplace=True)
```

In [93]:
```
# Now let's check our info again
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 110526 entries, 0 to 110526
Data columns (total 14 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   patient_id       110526 non-null  float64
 1   appointment_id   110526 non-null  int64  
```

```
 2   gender           110526 non-null  object
 3   scheduled_day    110526 non-null  object
 4   appointment_day  110526 non-null  object
 5   age              110526 non-null  int64
 6   neighbourhood    110526 non-null  object
 7   scholarship      110526 non-null  int64
 8   hypertension     110526 non-null  int64
 9   diabetes         110526 non-null  int64
 10  alcoholism       110526 non-null  int64
 11  handicap         110526 non-null  int64
 12  sms_received     110526 non-null  int64
 13  no_show          110526 non-null  object
dtypes: float64(1), int64(8), object(5)
memory usage: 12.6+ MB
```

In [94]:
```
df.duplicated().sum()
```

Out[94]:
```
0
```

In [95]:
```
# Now we are checking for unique values
{i : df[i].unique() for i in df.columns}
```

Out[95]:
```
{'patient_id': array([2.98724998e+13, 5.58997777e+14, 4.26296230e+12, ...,
        7.26331493e+13, 9.96997666e+14, 1.55766317e+13]),
 'appointment_id': array([5642903, 5642503, 5642549, ..., 5630692, 5630323, 5629448]),
 'gender': array(['F', 'M'], dtype=object),
 'scheduled_day': array(['2016-04-29T18:38:08Z', '2016-04-29T16:08:27Z',
        '2016-04-29T16:19:04Z', ..., '2016-04-27T16:03:52Z',
        '2016-04-27T15:09:23Z', '2016-04-27T13:30:56Z'], dtype=object),
 'appointment_day': array(['2016-04-29T00:00:00Z', '2016-05-03T00:00:00Z',
        '2016-05-10T00:00:00Z', '2016-05-17T00:00:00Z',
        '2016-05-24T00:00:00Z', '2016-05-31T00:00:00Z',
        '2016-05-02T00:00:00Z', '2016-05-30T00:00:00Z',
        '2016-05-16T00:00:00Z', '2016-05-04T00:00:00Z',
        '2016-05-19T00:00:00Z', '2016-05-12T00:00:00Z',
        '2016-05-06T00:00:00Z', '2016-05-20T00:00:00Z',
        '2016-05-05T00:00:00Z', '2016-05-13T00:00:00Z',
        '2016-05-09T00:00:00Z', '2016-05-25T00:00:00Z',
        '2016-05-11T00:00:00Z', '2016-05-18T00:00:00Z',
        '2016-05-14T00:00:00Z', '2016-06-02T00:00:00Z',
        '2016-06-03T00:00:00Z', '2016-06-06T00:00:00Z',
        '2016-06-07T00:00:00Z', '2016-06-01T00:00:00Z',
        '2016-06-08T00:00:00Z'], dtype=object),
 'age': array([ 62,  56,   8,  76,  23,  39,  21,  19,  30,  29,  22,  28,  54,
         15,  50,  40,  46,   4,  13,  65,  45,  51,  32,  12,  61,  38,
         79,  18,  63,  64,  85,  59,  55,  71,  49,  78,  31,  58,  27,
          6,   2,  11,   7,   0,   3,   1,  69,  68,  60,  67,  36,  10,
         35,  20,  26,  34,  33,  16,  42,   5,  47,  17,  41,  44,  37,
         24,  66,  77,  81,  70,  53,  75,  73,  52,  74,  43,  89,  57,
         14,   9,  48,  83,  72,  25,  80,  87,  88,  84,  82,  90,  94,
         86,  91,  98,  92,  96,  93,  95,  97, 102, 115, 100,  99]),
 'neighbourhood': array(['JARDIM DA PENHA', 'MATA DA PRAIA', 'PONTAL DE CAMBURI',
        'REPÚBLICA', 'GOIABEIRAS', 'ANDORINHAS', 'CONQUISTA',
        'NOVA PALESTINA', 'DA PENHA', 'TABUAZEIRO', 'BENTO FERREIRA',
        'SÃO PEDRO', 'SANTA MARTHA', 'SÃO CRISTÓVÃO', 'MARUÍPE',
        'GRANDE VITÓRIA', 'SÃO BENEDITO', 'ILHA DAS CAIEIRAS',
        'SANTO ANDRÉ', 'SOLON BORGES', 'BONFIM', 'JARDIM CAMBURI',
        'MARIA ORTIZ', 'JABOUR', 'ANTÔNIO HONÓRIO', 'RESISTÊNCIA',
        'ILHA DE SANTA MARIA', 'JUCUTUQUARA', 'MONTE BELO',
        'MÁRIO CYPRESTE', 'SANTO ANTÔNIO', 'BELA VISTA', 'PRAIA DO SUÁ',
        'SANTA HELENA', 'ITARARÉ', 'INHANGUETÁ', 'UNIVERSITÁRIO',
        'SÃO JOSÉ', 'REDENÇÃO', 'SANTA CLARA', 'CENTRO', 'PARQUE MOSCOSO',
        'DO MOSCOSO', 'SANTOS DUMONT', 'CARATOÍRA', 'ARIOVALDO FAVALESSA',
        'ILHA DO FRADE', 'GURIGICA', 'JOANA D´ARC', 'CONSOLAÇÃO',
        'PRAIA DO CANTO', 'BOA VISTA', 'MORADA DE CAMBURI', 'SANTA LUÍZA',
```

```
        'SANTA LÚCIA', 'BARRO VERMELHO', 'ESTRELINHA', 'FORTE SÃO JOÃO',
        'FONTE GRANDE', 'ENSEADA DO SUÁ', 'SANTOS REIS', 'PIEDADE',
        'JESUS DE NAZARETH', 'SANTA TEREZA', 'CRUZAMENTO',
        'ILHA DO PRÍNCIPE', 'ROMÃO', 'COMDUSA', 'SANTA CECÍLIA',
        'VILA RUBIM', 'DE LOURDES', 'DO QUADRO', 'DO CABRAL', 'HORTO',
        'SEGURANÇA DO LAR', 'ILHA DO BOI', 'FRADINHOS', 'NAZARETH',
        'AEROPORTO', 'ILHAS OCEÂNICAS DE TRINDADE', 'PARQUE INDUSTRIAL'],
       dtype=object),
 'scholarship': array([0, 1]),
 'hypertension': array([1, 0]),
 'diabetes': array([0, 1]),
 'alcoholism': array([0, 1]),
 'handicap': array([0, 1, 2, 3, 4]),
 'sms_received': array([0, 1]),
 'no_show': array(['No', 'Yes'], dtype=object)}
```

In [96]:
```
df.columns
```

Out[96]:
```
Index(['patient_id', 'appointment_id', 'gender', 'scheduled_day',
       'appointment_day', 'age', 'neighbourhood', 'scholarship',
       'hypertension', 'diabetes', 'alcoholism', 'handicap', 'sms_received',
       'no_show'],
      dtype='object')
```

Data Cleaning

> With the last cell we saw that there is **110527** rows and **14** columns. It also mention that we to update some date type especially first for scheduled and appointment days
>
> ### More important we are almost ready to start our exploration but we first need to transform a little bit our dataframe

In [97]:
```
# 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.
df.drop(['patient_id', 'appointment_id'], axis = 1, inplace = True)
df['appointment_day'] = pd.to_datetime(df['appointment_day'])
df['scheduled_day'] = pd.to_datetime(df['scheduled_day'])
df.head()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 110526 entries, 0 to 110526
Data columns (total 12 columns):
 #   Column           Non-Null Count   Dtype              
---  ------           --------------   -----              
 0   gender           110526 non-null  object             
 1   scheduled_day    110526 non-null  datetime64[ns, UTC]
 2   appointment_day  110526 non-null  datetime64[ns, UTC]
 3   age              110526 non-null  int64              
 4   neighbourhood    110526 non-null  object             
 5   scholarship      110526 non-null  int64              
 6   hypertension     110526 non-null  int64              
 7   diabetes         110526 non-null  int64              
 8   alcoholism       110526 non-null  int64              
 9   handicap         110526 non-null  int64              
 10  sms_received     110526 non-null  int64              
 11  no_show          110526 non-null  object             
dtypes: datetime64[ns, UTC](2), int64(7), object(3)
memory usage: 11.0+ MB
```

In [98]:
```
# Let's transform the no_show coulumns but not required
df['no_show'].replace({'Yes': 1, 'No': 0}, inplace = True)
```

In [99]:
```
df.sample(10)
```

Out[99]:

	gender	scheduled_day	appointment_day	age	neighbourhood	scholarship	hypertension	diabetes	al
97350	F	2016-05-24 10:31:45+00:00	2016-06-03 00:00:00+00:00	74	PARQUE MOSCOSO	0	0	0	
39025	F	2016-05-13 18:13:09+00:00	2016-05-17 00:00:00+00:00	23	RESISTÊNCIA	0	0	0	
47450	F	2016-05-19 13:14:17+00:00	2016-05-19 00:00:00+00:00	66	MARUÍPE	0	1	0	
87353	F	2016-06-01 07:09:25+00:00	2016-06-01 00:00:00+00:00	43	SANTA LUÍZA	0	0	0	
82546	F	2016-04-15 16:12:44+00:00	2016-05-02 00:00:00+00:00	57	ITARARÉ	0	0	0	
66812	F	2016-05-09 11:27:22+00:00	2016-05-09 00:00:00+00:00	66	MARIA ORTIZ	0	1	1	
29420	F	2016-04-25 09:30:36+00:00	2016-05-10 00:00:00+00:00	18	FORTE SÃO JOÃO	0	0	0	
102098	F	2016-05-30 06:41:24+00:00	2016-06-01 00:00:00+00:00	21	MARIA ORTIZ	0	0	0	
36540	F	2016-05-10 11:43:01+00:00	2016-05-12 00:00:00+00:00	21	SÃO BENEDITO	0	0	0	
51506	F	2016-04-20 09:10:06+00:00	2016-05-04 00:00:00+00:00	74	CENTRO	0	1	0	

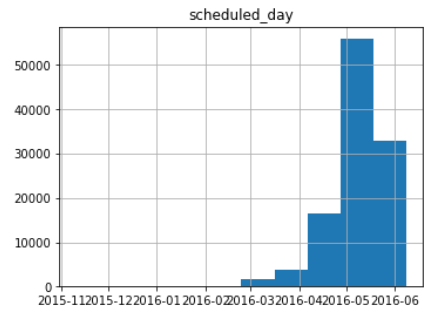
Exploratory Data Analysis

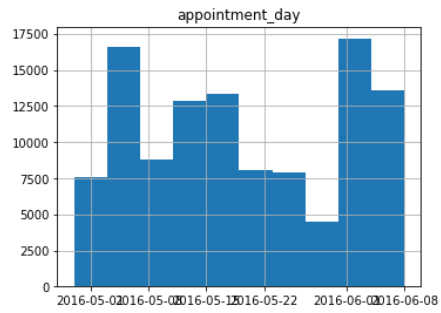
> Let's use some quick graphics to get answer for our early asked questions
>
> First , let's check hist plot for all feature for quick exploration of distribution on each feature

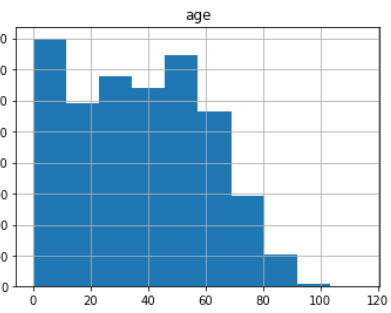
Research Question 1 (Is the number of gender equals through the dataset ?)

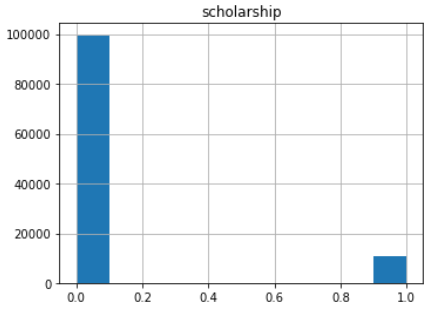
In [100…
```
# Use this, and more code cells, to explore your data. Don't forget to add
#   Markdown cells to document your observations and findings.
df.hist(figsize = [20, 20])
```

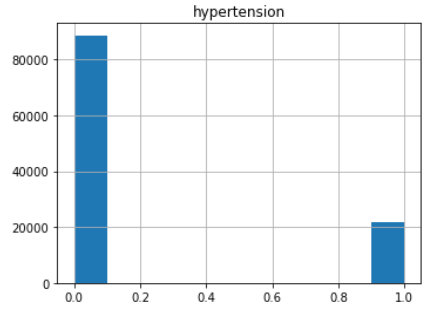
Out[100]:
```
array([[<AxesSubplot:title={'center':'scheduled_day'}>,
        <AxesSubplot:title={'center':'appointment_day'}>,
        <AxesSubplot:title={'center':'age'}>],
       [<AxesSubplot:title={'center':'scholarship'}>,
        <AxesSubplot:title={'center':'hypertension'}>,
        <AxesSubplot:title={'center':'diabetes'}>],
       [<AxesSubplot:title={'center':'alcoholism'}>,
        <AxesSubplot:title={'center':'handicap'}>,
        <AxesSubplot:title={'center':'sms_received'}>],
       [<AxesSubplot:title={'center':'no_show'}>, <AxesSubplot:>,
        <AxesSubplot:>]], dtype=object)
```

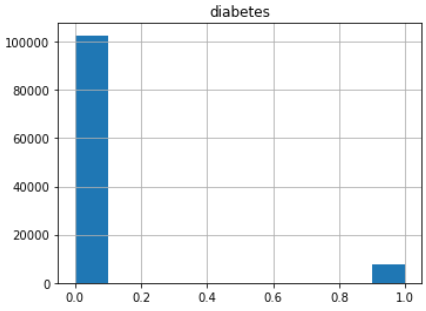


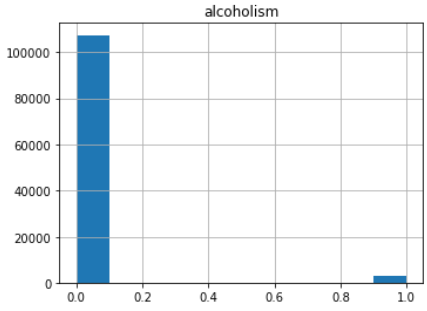


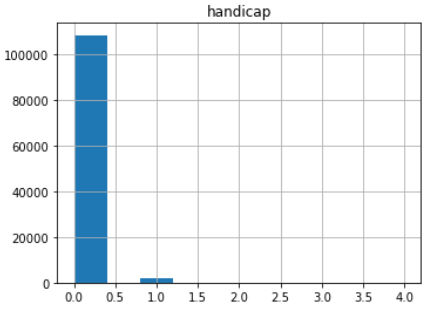


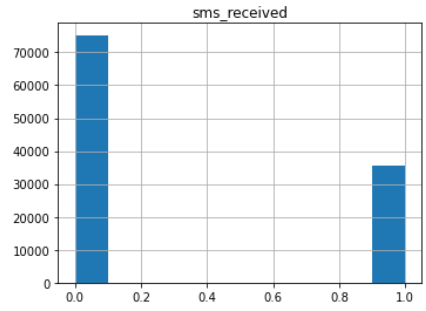


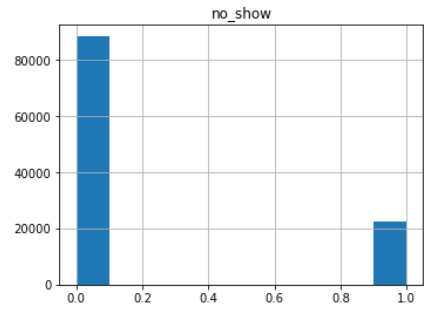






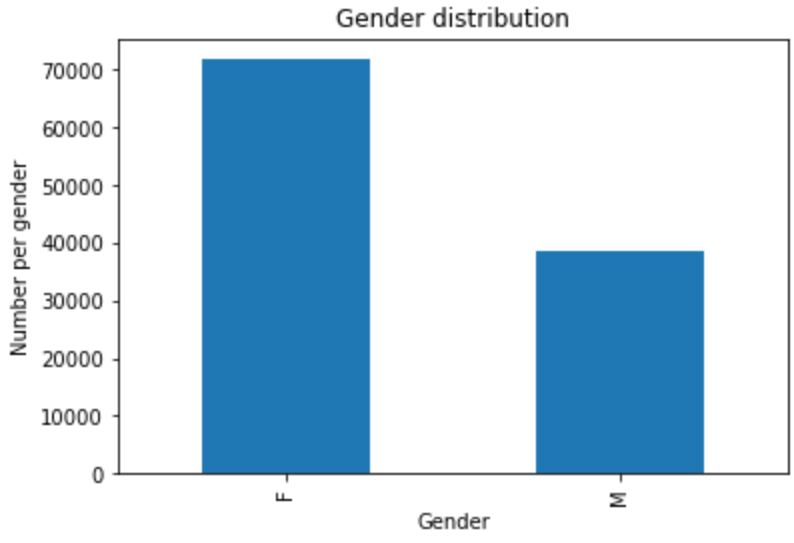






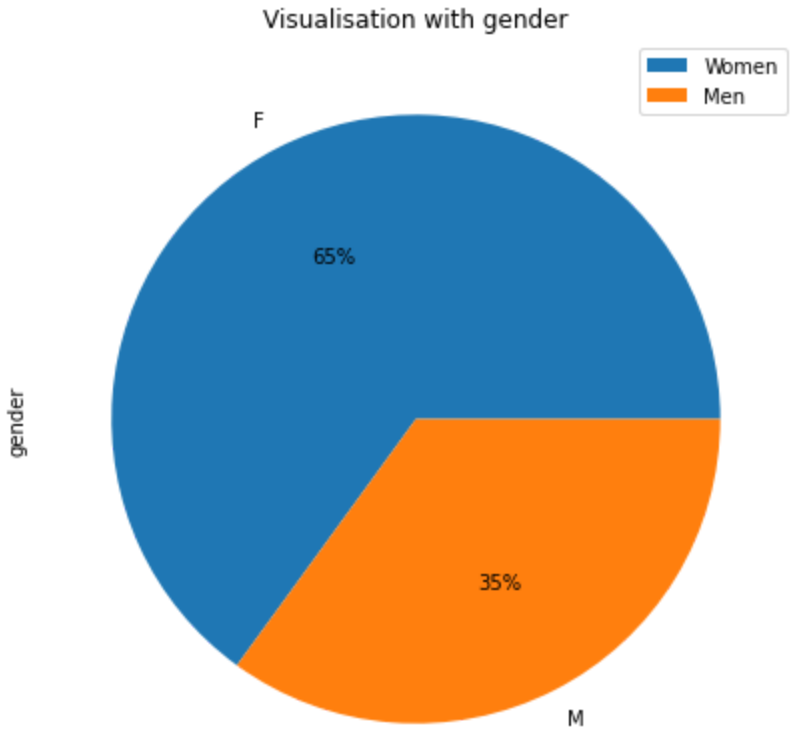
In [101…
```
df['gender'].value_counts().plot(kind = 'bar', xlabel= 'Gender', ylabel= 'Number per gen
```

Out[101]:
```
<AxesSubplot:title={'center':'Gender distribution'}, xlabel='Gender', ylabel='Number pe
r gender'>
```

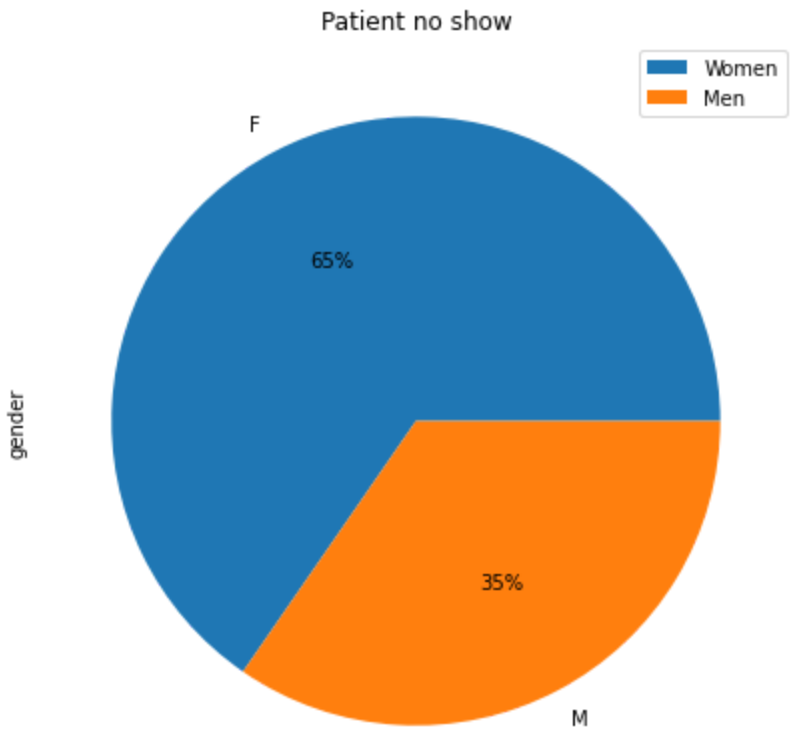


In [102…
```
def show_pie(data, title, legend):
    fig, ax = plt.subplots()
    data.value_counts().plot(kind='pie', autopct='%.0f%%', ax=ax, figsize= (7,7))
    ax.legend(legend);
    ax.set_title(title)
```

In [103…
```
show_pie(df.gender, 'Visualisation with gender',["Women", "Men"])
```

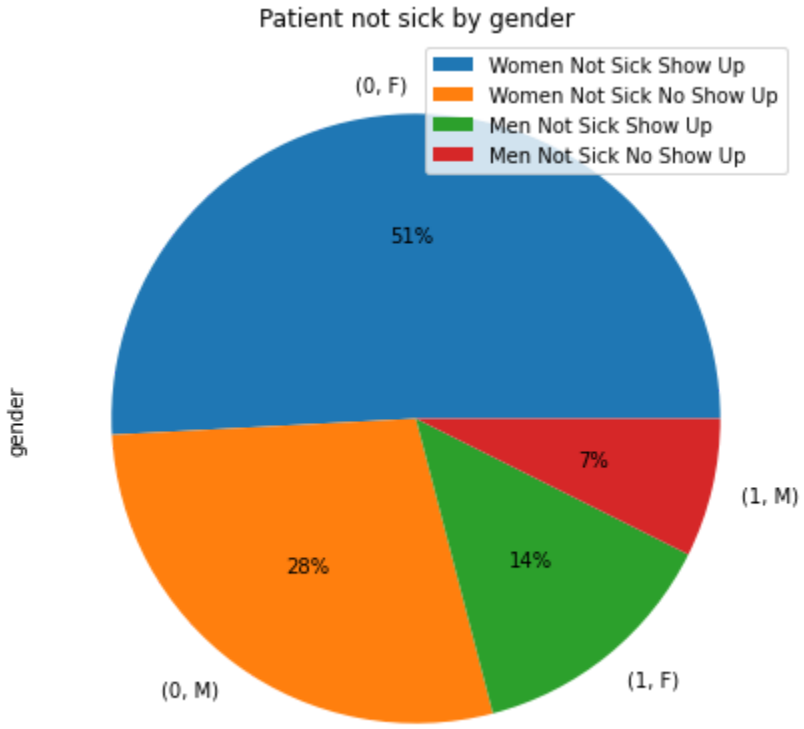


In [104…
```
df_noshow = df.query('no_show == 1')
show_pie(df_noshow.gender, "Patient no show", ["Women", "Men"])
```



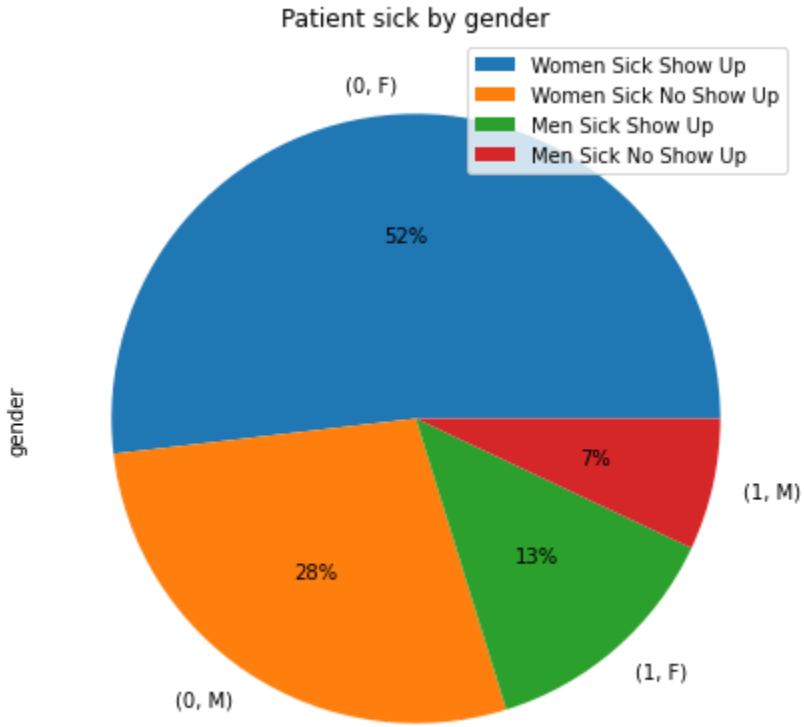
In [105…
```
df_no_show_not_sick =  df.query('alcoholism == 0 and hypertension == 0 and diabetes == 0
```

```
show_pie(df_no_show_not_sick.gender, 'Patient not sick by gender', ["Women Not Sick Show
```



In [106…

```
df_no_show_seek =  df.query('alcoholism == 0 or hypertension == 0 or diabetes == 0  or h
show_pie(df_no_show_seek.gender, 'Patient sick by gender', ["Women Sick Show Up", "Women
```

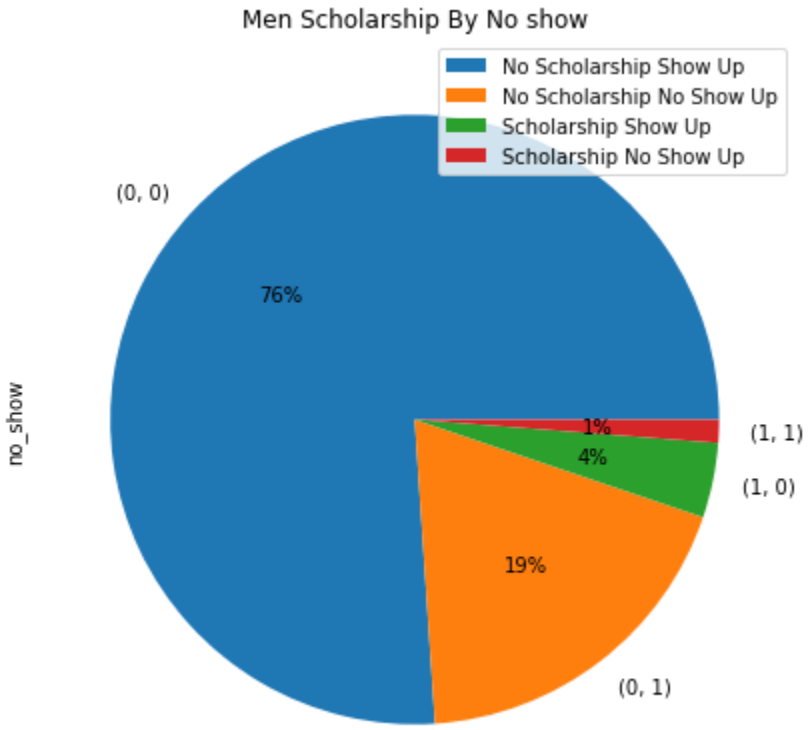


Research Question 2 and 4 (Is the case that a patient get scholarship will help to show up ? / Is patient going to show up wether they are sick or not ?)

In [107…

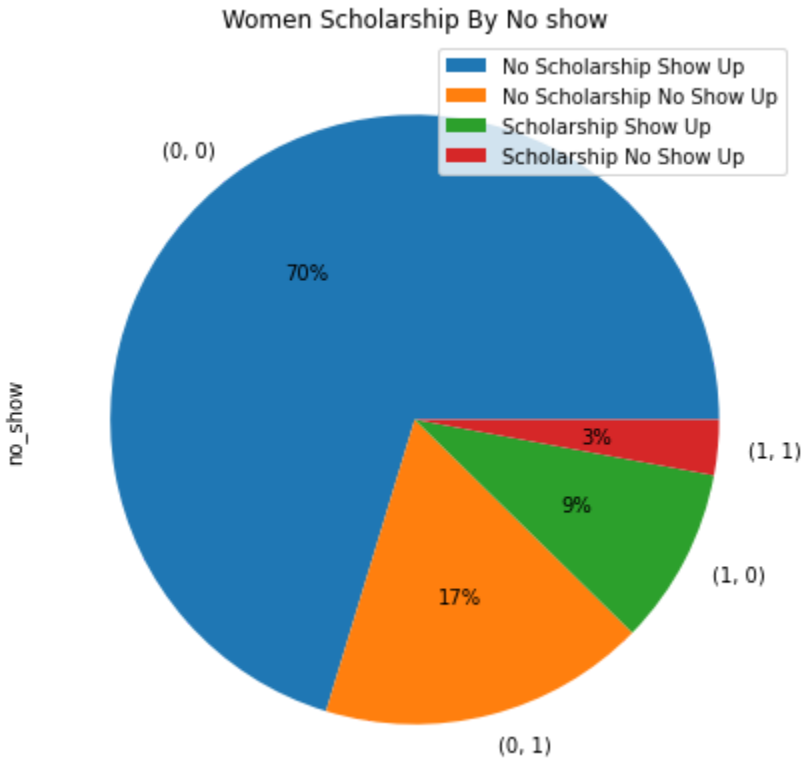
```
# Explore the dataset based on scholarship
df_scholar_men = df.query('gender == "M"').groupby(['scholarship'])
```

```
show_pie(df_scholar_men.no_show, 'Men Scholarship By No show', ["No Scholarship Show Up"
```



In [108…

```
df_scholar_women = df.query('gender == "F"').groupby(['scholarship'])

show_pie(df_scholar_women.no_show, 'Women Scholarship By No show', ["No Scholarship Show
```



Research Question 3 (What days of week patient sow up easily for they appointment ?)

In [109…

```
# Let's first transform day to get what days is suitable for patients.
```

```
df_day_week = df.copy()
df_day_week['appointment_day'] =  df['appointment_day'].dt.dayofweek
df_day_week['scheduled_day'] =  df['scheduled_day'].dt.dayofweek
df_day_week.head()
```

Out[109]:

	gender	scheduled_day	appointment_day	age	neighbourhood	scholarship	hypertension	diabetes	alcoho
0	F	4	4	62	JARDIM DA PENHA	0	1	0	
1	M	4	4	56	JARDIM DA PENHA	0	0	0	
2	F	4	4	62	MATA DA PRAIA	0	0	0	
3	F	4	4	8	PONTAL DE CAMBURI	0	0	0	
4	F	4	4	56	JARDIM DA PENHA	0	1	1	

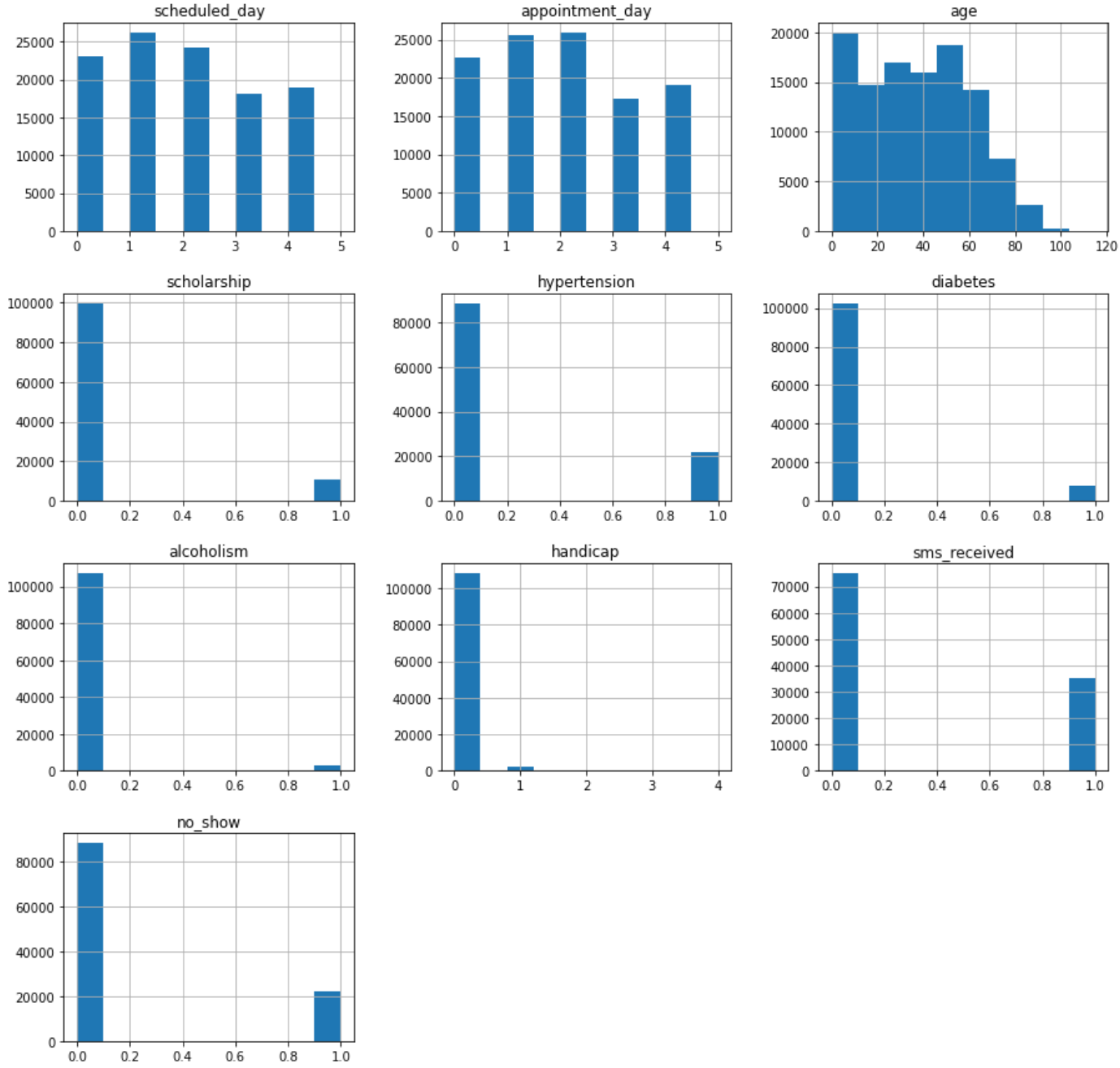
In [110…
```
df_day_week.sample(10)
```

Out[110]:

	gender	scheduled_day	appointment_day	age	neighbourhood	scholarship	hypertension	diabetes	a
75390	M	2	2	67	NOVA PALESTINA	1	1	1	
96891	F	2	2	4	MARUÍPE	0	0	0	
80390	M	0	3	63	DA PENHA	0	1	0	
89372	F	2	2	29	CONQUISTA	0	0	0	
58438	F	3	3	55	MARUÍPE	0	0	0	
18943	F	2	0	7	FORTE SÃO JOÃO	0	0	0	
108310	F	3	3	25	ROMÃO	0	0	0	
97226	F	3	0	72	ANDORINHAS	0	1	1	
219	M	4	4	45	ANDORINHAS	0	0	0	
98950	F	2	2	63	REPÚBLICA	0	1	0	

In [111…
```
df_day_week.hist(figsize= [15,15])
```

Out[111]:
```
array([[<AxesSubplot:title={'center':'scheduled_day'}>,
        <AxesSubplot:title={'center':'appointment_day'}>,
        <AxesSubplot:title={'center':'age'}>],
       [<AxesSubplot:title={'center':'scholarship'}>,
        <AxesSubplot:title={'center':'hypertension'}>,
        <AxesSubplot:title={'center':'diabetes'}>],
       [<AxesSubplot:title={'center':'alcoholism'}>,
        <AxesSubplot:title={'center':'handicap'}>,
        <AxesSubplot:title={'center':'sms_received'}>],
       [<AxesSubplot:title={'center':'no_show'}>, <AxesSubplot:>,
        <AxesSubplot:>]], dtype=object)
```



In [112…

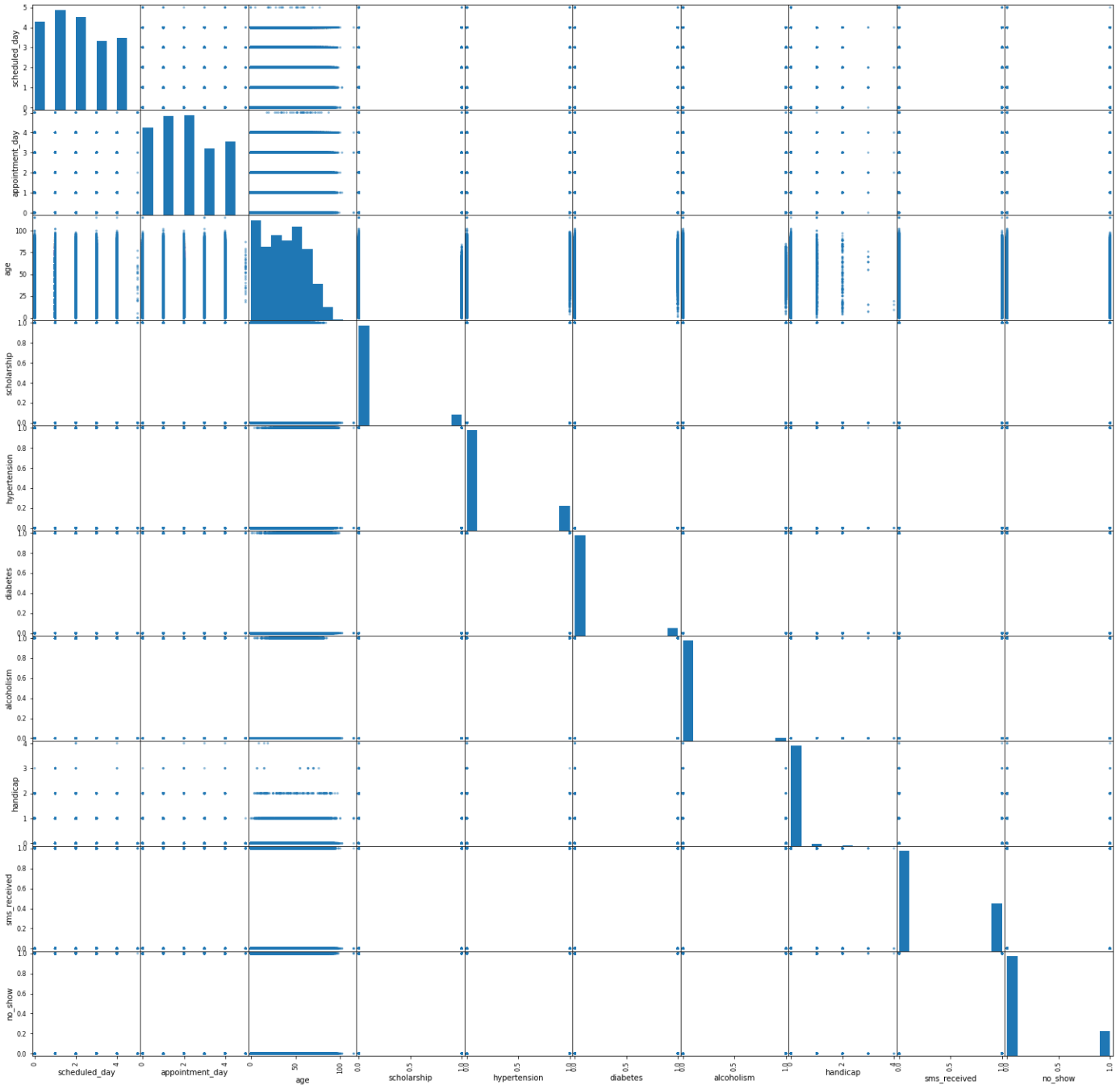
```
# Quick Explore with neighbourhood
df_day_week_group_age = df_day_week.query('no_show == 1').groupby(['neighbourhood'])

pd.plotting.scatter_matrix(df_day_week, figsize = (25,25))
```

Out[112]:

```
array([[<AxesSubplot:xlabel='scheduled_day', ylabel='scheduled_day'>,
        <AxesSubplot:xlabel='appointment_day', ylabel='scheduled_day'>,
        <AxesSubplot:xlabel='age', ylabel='scheduled_day'>,
        <AxesSubplot:xlabel='scholarship', ylabel='scheduled_day'>,
        <AxesSubplot:xlabel='hypertension', ylabel='scheduled_day'>,
        <AxesSubplot:xlabel='diabetes', ylabel='scheduled_day'>,
        <AxesSubplot:xlabel='alcoholism', ylabel='scheduled_day'>,
        <AxesSubplot:xlabel='handicap', ylabel='scheduled_day'>,
        <AxesSubplot:xlabel='sms_received', ylabel='scheduled_day'>,
        <AxesSubplot:xlabel='no_show', ylabel='scheduled_day'>],
       [<AxesSubplot:xlabel='scheduled_day', ylabel='appointment_day'>,
        <AxesSubplot:xlabel='appointment_day', ylabel='appointment_day'>,
        <AxesSubplot:xlabel='age', ylabel='appointment_day'>,
        <AxesSubplot:xlabel='scholarship', ylabel='appointment_day'>,
        <AxesSubplot:xlabel='hypertension', ylabel='appointment_day'>,
        <AxesSubplot:xlabel='diabetes', ylabel='appointment_day'>,
        <AxesSubplot:xlabel='alcoholism', ylabel='appointment_day'>,
        <AxesSubplot:xlabel='handicap', ylabel='appointment_day'>,
```

```
 <AxesSubplot:xlabel='sms_received', ylabel='appointment_day'>,
 <AxesSubplot:xlabel='no_show', ylabel='appointment_day'>],
[<AxesSubplot:xlabel='scheduled_day', ylabel='age'>,
 <AxesSubplot:xlabel='appointment_day', ylabel='age'>,
 <AxesSubplot:xlabel='age', ylabel='age'>,
 <AxesSubplot:xlabel='scholarship', ylabel='age'>,
 <AxesSubplot:xlabel='hypertension', ylabel='age'>,
 <AxesSubplot:xlabel='diabetes', ylabel='age'>,
 <AxesSubplot:xlabel='alcoholism', ylabel='age'>,
 <AxesSubplot:xlabel='handicap', ylabel='age'>,
 <AxesSubplot:xlabel='sms_received', ylabel='age'>,
 <AxesSubplot:xlabel='no_show', ylabel='age'>],
[<AxesSubplot:xlabel='scheduled_day', ylabel='scholarship'>,
 <AxesSubplot:xlabel='appointment_day', ylabel='scholarship'>,
 <AxesSubplot:xlabel='age', ylabel='scholarship'>,
 <AxesSubplot:xlabel='scholarship', ylabel='scholarship'>,
 <AxesSubplot:xlabel='hypertension', ylabel='scholarship'>,
 <AxesSubplot:xlabel='diabetes', ylabel='scholarship'>,
 <AxesSubplot:xlabel='alcoholism', ylabel='scholarship'>,
 <AxesSubplot:xlabel='handicap', ylabel='scholarship'>,
 <AxesSubplot:xlabel='sms_received', ylabel='scholarship'>,
 <AxesSubplot:xlabel='no_show', ylabel='scholarship'>],
[<AxesSubplot:xlabel='scheduled_day', ylabel='hypertension'>,
 <AxesSubplot:xlabel='appointment_day', ylabel='hypertension'>,
 <AxesSubplot:xlabel='age', ylabel='hypertension'>,
 <AxesSubplot:xlabel='scholarship', ylabel='hypertension'>,
 <AxesSubplot:xlabel='hypertension', ylabel='hypertension'>,
 <AxesSubplot:xlabel='diabetes', ylabel='hypertension'>,
 <AxesSubplot:xlabel='alcoholism', ylabel='hypertension'>,
 <AxesSubplot:xlabel='handicap', ylabel='hypertension'>,
 <AxesSubplot:xlabel='sms_received', ylabel='hypertension'>,
 <AxesSubplot:xlabel='no_show', ylabel='hypertension'>],
[<AxesSubplot:xlabel='scheduled_day', ylabel='diabetes'>,
 <AxesSubplot:xlabel='appointment_day', ylabel='diabetes'>,
 <AxesSubplot:xlabel='age', ylabel='diabetes'>,
 <AxesSubplot:xlabel='scholarship', ylabel='diabetes'>,
 <AxesSubplot:xlabel='hypertension', ylabel='diabetes'>,
 <AxesSubplot:xlabel='diabetes', ylabel='diabetes'>,
 <AxesSubplot:xlabel='alcoholism', ylabel='diabetes'>,
 <AxesSubplot:xlabel='handicap', ylabel='diabetes'>,
 <AxesSubplot:xlabel='sms_received', ylabel='diabetes'>,
 <AxesSubplot:xlabel='no_show', ylabel='diabetes'>],
[<AxesSubplot:xlabel='scheduled_day', ylabel='alcoholism'>,
 <AxesSubplot:xlabel='appointment_day', ylabel='alcoholism'>,
 <AxesSubplot:xlabel='age', ylabel='alcoholism'>,
 <AxesSubplot:xlabel='scholarship', ylabel='alcoholism'>,
 <AxesSubplot:xlabel='hypertension', ylabel='alcoholism'>,
 <AxesSubplot:xlabel='diabetes', ylabel='alcoholism'>,
 <AxesSubplot:xlabel='alcoholism', ylabel='alcoholism'>,
 <AxesSubplot:xlabel='handicap', ylabel='alcoholism'>,
 <AxesSubplot:xlabel='sms_received', ylabel='alcoholism'>,
 <AxesSubplot:xlabel='no_show', ylabel='alcoholism'>],
[<AxesSubplot:xlabel='scheduled_day', ylabel='handicap'>,
 <AxesSubplot:xlabel='appointment_day', ylabel='handicap'>,
 <AxesSubplot:xlabel='age', ylabel='handicap'>,
 <AxesSubplot:xlabel='scholarship', ylabel='handicap'>,
 <AxesSubplot:xlabel='hypertension', ylabel='handicap'>,
 <AxesSubplot:xlabel='diabetes', ylabel='handicap'>,
 <AxesSubplot:xlabel='alcoholism', ylabel='handicap'>,
 <AxesSubplot:xlabel='handicap', ylabel='handicap'>,
 <AxesSubplot:xlabel='sms_received', ylabel='handicap'>,
 <AxesSubplot:xlabel='no_show', ylabel='handicap'>],
[<AxesSubplot:xlabel='scheduled_day', ylabel='sms_received'>,
 <AxesSubplot:xlabel='appointment_day', ylabel='sms_received'>,
 <AxesSubplot:xlabel='age', ylabel='sms_received'>,
 <AxesSubplot:xlabel='scholarship', ylabel='sms_received'>,
```

```
 <AxesSubplot:xlabel='hypertension', ylabel='sms_received'>,
 <AxesSubplot:xlabel='diabetes', ylabel='sms_received'>,
 <AxesSubplot:xlabel='alcoholism', ylabel='sms_received'>,
 <AxesSubplot:xlabel='handicap', ylabel='sms_received'>,
 <AxesSubplot:xlabel='sms_received', ylabel='sms_received'>,
 <AxesSubplot:xlabel='no_show', ylabel='sms_received'>],
[<AxesSubplot:xlabel='scheduled_day', ylabel='no_show'>,
 <AxesSubplot:xlabel='appointment_day', ylabel='no_show'>,
 <AxesSubplot:xlabel='age', ylabel='no_show'>,
 <AxesSubplot:xlabel='scholarship', ylabel='no_show'>,
 <AxesSubplot:xlabel='hypertension', ylabel='no_show'>,
 <AxesSubplot:xlabel='diabetes', ylabel='no_show'>,
 <AxesSubplot:xlabel='alcoholism', ylabel='no_show'>,
 <AxesSubplot:xlabel='handicap', ylabel='no_show'>,
 <AxesSubplot:xlabel='sms_received', ylabel='no_show'>,
 <AxesSubplot:xlabel='no_show', ylabel='no_show'>]], dtype=object)
```



Conclusions

1. Is the number of gender equals through the dataset ? **Answer** : We have **65%** Women and **35%** Men. So no the dataset is not equally provided for gender feature

2. Is the case that a patient get scholarship will help to show up ? **Answer** :

```
The scholarship is not deterministic for this exploration.
```

3. What days of week patient sow up easily for they appointment ? **Answer** : Quick comparaison of appointment days of week and sheduled days of week show that patients come regularly for appointment on (Monday, Wednesday and Friday). **Tuesday and Thursday patients number decrease a little**
4. Is patient going to show up wether they are sick or not ? **Answer** : No matter they are sick or not the pies charts show us that we have arround the same behaviour for all patients

So patients sheduled for (Monday, Wednesday and Friday) will mostly show up

> With our exploration there is nothing that realy show why patient show up or not.

Submitting your Project

Limitations

> With our exploration there is nothing that realy show why patient show up or not. Maybe others features should be added, like we can try to find other patterns why patient show or not.

In [113…

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