Project: No-Show Appointment DataSet

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. The main qeustion we are trying to answer here is What factors are important for us to know in order to predict if a patient will show up for their scheduled appointment?

And every patient has its own :

- 1.PatientId / Identification of a patient.
- 2.AppointmentID / Identification of each appointment.
- 3.Gender / Male or Female.
- 4.ScheduledDay /tells us on what day the patient set up their appointment. .
- 5.AppointmentDay / The day of the appointment.
- 6.Age / How old the patient is in years.
- 7.Neighbourhood / the location of the hospital.
- 8.Scholarship / 1 for True, 0 for False,the patient has government medical support.
- 9.Hipertension / 1 for True, 0 for False.
- 10.Diabetes / 1 for True, 0 for False.
- 11.Alcoholism / if the patient is Alcoholism; 1 for True, 0 for False.
- 12.Handcap / number of disabilities a patient has.
- 13.SMS_received / 1 for True, 0 for False.
- 14.No-show / Yes or No, if the patient showup = No, if he didn't = Yes.

> Questions that we could explore:
>
> - 1- Is there a relationship between gender and show up?
> - 2- Does the age of patients affect why they are showup or not?
> - 3- Which neighborhood has the most no show rate?
> - 4- Does the patient who have scholarship show up or not?
> - 5- Does the hypertension affect the patient's show up?
> - 6- Does the diabetes affect the patient's show up?
> - 7- Does the alcoholism affect the patient's show up?
> - 8- Does the handicap affect the patient's show up?
> - 9- Did the patients receiving sms keep their appointment?

importing revelant libraries

In [1]:
```
# importing important libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
plt.style.use('ggplot')
```

Data Wrangling

> Now we got the data we need in a form we can work with in three steps:
>
> - 1- first we will gather the data we need to answer our questions.
> - 2- seconed we will assess our data to idintify any proplems in data's quality or structer.
> - 3- finally clean our data by modifing,replacing or removing data to ensure that our dataset is of the heighest quality.

General Properties

In [2]:
```
# Load the dataset:
df = pd.read_csv('noshowappointments-kagglev2-may-2016.csv')
```

In [3]:
```
#Let's look at the first five rows of the data:
df.head(5)
```

Out[3]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	S
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 [4]:
```
#Let's look at the last five rows of the data:
df.tail(5)
```

Out[4]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhc

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhc
110522	2.572134e+12	5651768	F	2016-05-03T09:15:35Z	2016-06-07T00:00:00Z	56	MARIA OF
110523	3.596266e+12	5650093	F	2016-05-03T07:27:33Z	2016-06-07T00:00:00Z	51	MARIA OF
110524	1.557663e+13	5630692	F	2016-04-27T16:03:52Z	2016-06-07T00:00:00Z	21	MARIA OF
110525	9.213493e+13	5630323	F	2016-04-27T15:09:23Z	2016-06-07T00:00:00Z	38	MARIA OF
110526	3.775115e+14	5629448	F	2016-04-27T13:30:56Z	2016-06-07T00:00:00Z	54	MARIA OF

In [5]:
```
#the basic information about the 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   PatientId       110527 non-null  float64
 1   AppointmentID   110527 non-null  int64  
 2   Gender          110527 non-null  object 
 3   ScheduledDay    110527 non-null  object 
 4   AppointmentDay  110527 non-null  object 
 5   Age             110527 non-null  int64  
 6   Neighbourhood   110527 non-null  object 
 7   Scholarship     110527 non-null  int64  
 8   Hipertension    110527 non-null  int64  
 9   Diabetes        110527 non-null  int64  
 10  Alcoholism      110527 non-null  int64  
 11  Handcap         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 [6]:
```
#exploration data shape:
df.shape
```

Out[6]:
```
(110527, 14)
```

In [7]:
```
#Find out the null values in the data:
df.isnull().sum()
```

Out[7]:
```
PatientId         0
AppointmentID     0
Gender            0
ScheduledDay      0
AppointmentDay    0
Age               0
Neighbourhood     0
Scholarship       0
Hipertension      0
```

```
Diabetes          0
Alcoholism        0
Handcap           0
SMS_received      0
No-show           0
dtype: int64
```

In [8]:
```
#chek for any duplicated values:
df.duplicated().sum()
```

Out[8]:
```
0
```

In [9]:
```
# for knowing statistical information:
df.describe()
```

Out[9]:

	PatientId	AppointmentID	Age	Scholarship	Hipertension	Diabetes
count	1.105270e+05	1.105270e+05	110527.000000	110527.000000	110527.000000	110527.000000
mean	1.474963e+14	5.675305e+06	37.088874	0.098266	0.197246	0.071865
std	2.560949e+14	7.129575e+04	23.110205	0.297675	0.397921	0.258265
min	3.921784e+04	5.030230e+06	-1.000000	0.000000	0.000000	0.000000
25%	4.172614e+12	5.640286e+06	18.000000	0.000000	0.000000	0.000000
50%	3.173184e+13	5.680573e+06	37.000000	0.000000	0.000000	0.000000
75%	9.439172e+13	5.725524e+06	55.000000	0.000000	0.000000	0.000000
max	9.999816e+14	5.790484e+06	115.000000	1.000000	1.000000	1.000000

Data Cleaning (Replace this with more specific notes!)

- 1- First I'll delete the columns I don't need:
  - PatientId > drop.
  - AppointmentID > drop.
- 2- Secondly I need to 'rename' some columns names and put underscore in some columns to make them more readalbe.
  - Frist No-show column before change it to show means 'No' if the patient showed up to their appointment, and 'Yes' if they did not show up.
  - So after I changed the column to show: '1' means that the patient showed up to their appointment, and '0' means that they didn't show up.
  - All the dataset now (1 = True) and (0 = False).
- 3- Third I need to change some coulmns type from object to date time:
  - ScheduledDay: string >> datetime
  - appointmentday: string >> datetime
- 4- Fourthly when we make describe to our data we find out a negative value in 'age' so we need to check that column to figer out what is the proplem .

In [10]:
```
#Delete columns that I will not use
df.drop(['PatientId','AppointmentID'], axis=1, inplace = True)
```

In [11]:
```
#fix and put underscore in some columns:
new_col = {'ScheduledDay':'Scheduled_Day','AppointmentDay':'Appointment_Day','Hipert
df.rename(columns = new_col , inplace = True)
```

In [12]:
```
#replace uppercase to lowercase:
df.rename(columns = lambda x : x.lower(), inplace=True)
```

In [13]:
```
# change the show column values
# 1=show , 0= noshow
df['show'] = df['show'].apply( lambda x: 1 if x == 'No' else 0)
```

In [14]:
```
#see the data after fixing:
df.head(5)
```

Out[14]:

	gender	scheduled_day	appointment_day	age	neighbourhood	scholarship	hypertension	diabe
0	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	0	1	
1	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	0	
2	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	0	0	
3	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	0	0	
4	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	1	

In [15]:
```
# Converting the date information in string to datetime type:
df['scheduled_day']=pd.to_datetime(df['scheduled_day'])
df['appointment_day']=pd.to_datetime(df['appointment_day'])
```

In [16]:
```
# find out all negative values in age:
df_age = df.query('age < 0')
df_age
```

Out[16]:

	gender	scheduled_day	appointment_day	age	neighbourhood	scholarship	hypertension	d
99832	F	2016-06-06 08:58:13+00:00	2016-06-06 00:00:00+00:00	-1	ROMÃO	0	0	

In [17]:
```
# drop the age which is less than 0:
df.drop(df_age.index, inplace=True)
```

In [18]:
```
#check the data after fixing:
```

```
df_age = df.query('age < 0')
df_age
```

Out[18]:

gender	scheduled_day	appointment_day	age	neighbourhood	scholarship	hypertension	diabete

In [19]:

```
df.dtypes
```

Out[19]:

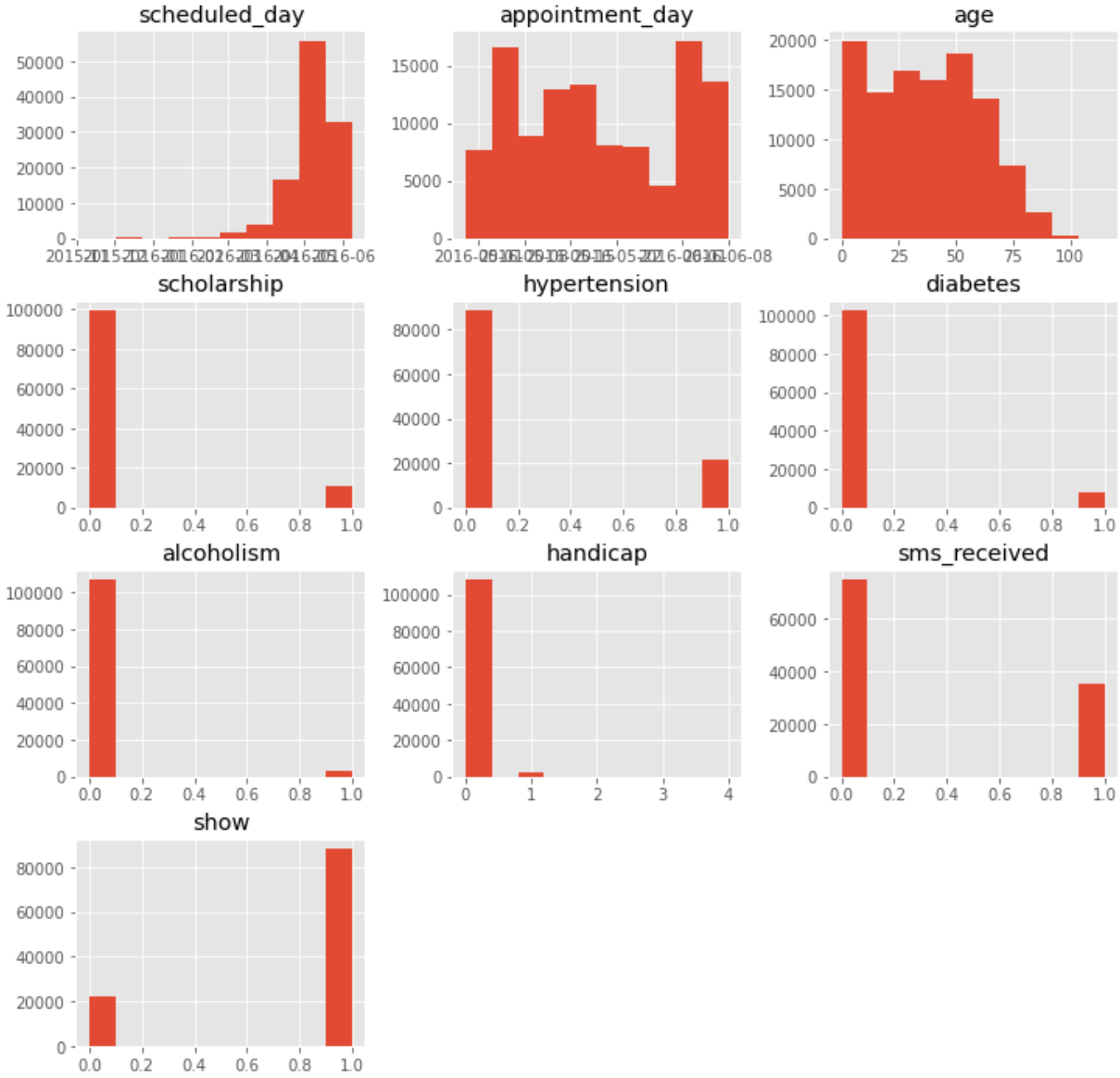
```
gender                          object
scheduled_day      datetime64[ns, UTC]
appointment_day    datetime64[ns, UTC]
age                              int64
neighbourhood                   object
scholarship                      int64
hypertension                     int64
diabetes                         int64
alcoholism                       int64
handicap                         int64
sms_received                     int64
show                             int64
dtype: object
```

Exploratory Data Analysis

> Now we will explore and then augment our data to maximize the potential of our analyses. after explore we can creat better feature from our data.
>
> - Exploring involves finding patterns in our data.
> - Visualizing relationships in our data.
> - Building intuition about what we're working.

In [20]:

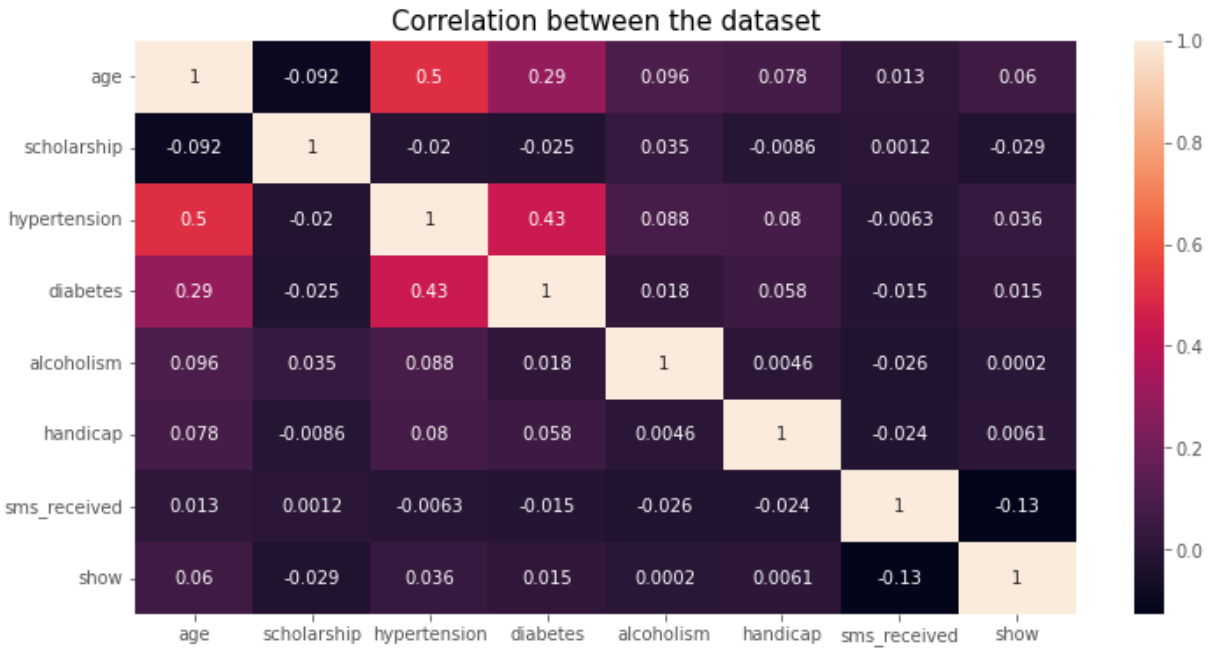
```
#explore dataset:
df.hist(figsize=(12,12));
```



> From histograms we can see that the majority of Patients:
>
> - are bellow 60 years old.
> - doesn't suffer from alcoholism/diabetes/hypertension.
> - are not handicapped.
> - have no Scholarship.
> - doesn't received a reminder sms.
> - doesn't missed the appointment.

In [21]:

```
# find out the correlation between the dataset(visualize correlation matrix):
plt.figure(figsize=(12, 6))
sns.heatmap(df.corr(),annot=True)
plt.title('Correlation between the dataset', fontsize=15);
```



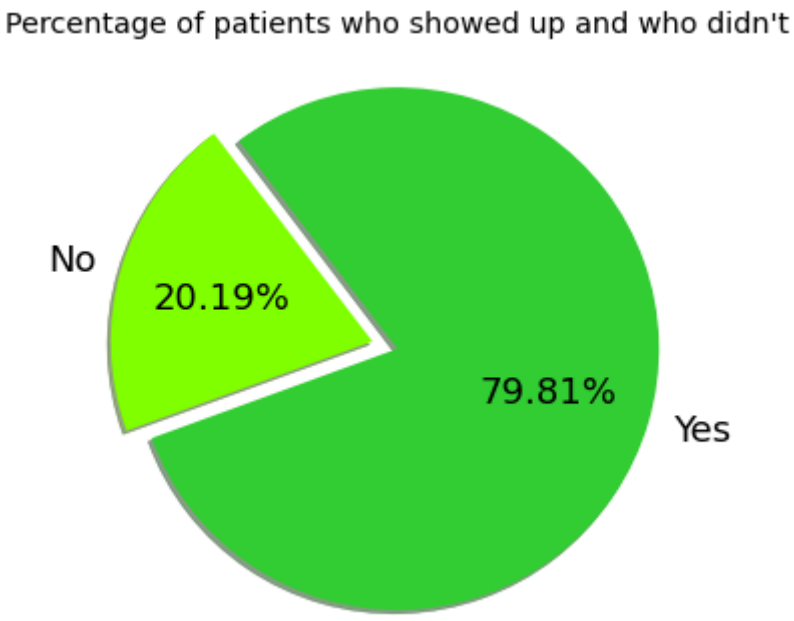
- Hypertension and Diabetes have medium postive correlation(0.43).
- Hypertension and age have strong postive correlation(0.5).
- Scholarship and show have negative correlation(-0.02).
- Alcoholism and show don't have any relationship(0.0002).
- Sms_received and show have strong negative correlation(-0.13).

Question 1

Is there a relationship between gender and show up?

In [22]:
```
def ratio_calculate(df, column_name1, column_name2):
    """
    calculate the ratio of given variables.

    input:

    df - the original dataframe.
    column_name1 - the name of the column I will use to groupby data.
    column_name2 - the name of the column we need to get value counts.

    output:

    pandas series contains the ratio :


    """
    ratio = df.groupby(column_name1)[column_name2].value_counts(normalize=True).unst

    return ratio
```

In [23]:
```
#pie chart explain the correlation between the ratio of patients who showed up and w
labels = ['Yes','No']
data = df['show'].value_counts()
color=['LimeGreen','Chartreuse']
explode = (0, 0.15)
plt.pie(data,radius=1.5,colors=color,labels=labels,explode=explode,autopct='%0.2f%%'
plt.title("Percentage of patients who showed up and who didn't",y=1.2);
```



> from the pie chart we find out that:
>
> - The patients who have attend about 80%
> - The patients who have miss their appointment about 20%

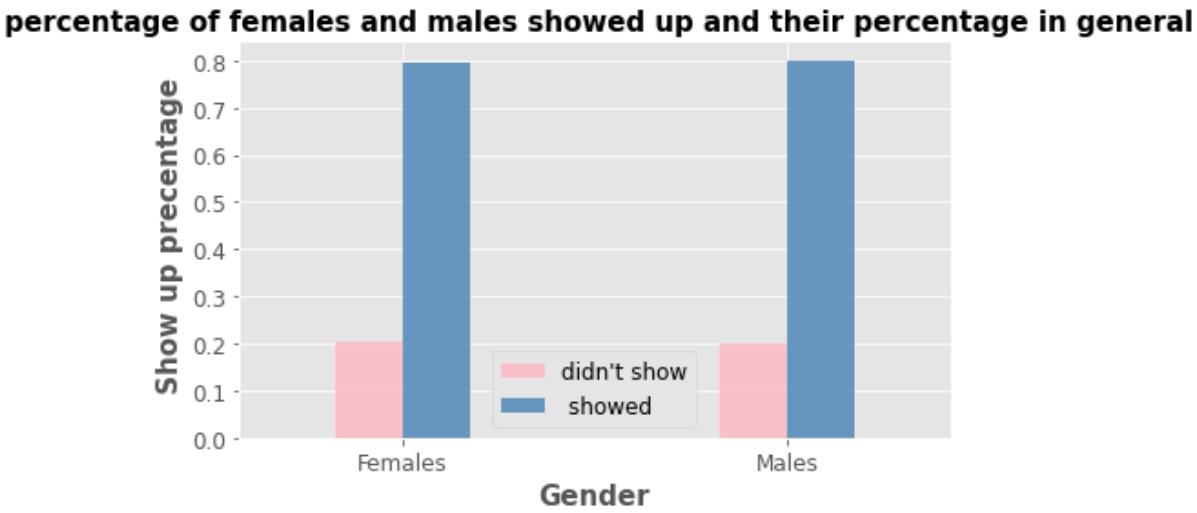
In [24]:
```
#create a data frame that contains patients who attend in their appointment:
df_show = df.query('show == 1')
```

In [25]:
```
#Percentage of females and males who show up compared to their total in general
gender_ratio = ratio_calculate(df,'gender',"show")
gender_ratio
```

Out[25]:

show	0	1
gender		
F	0.203149	0.796851
M	0.199679	0.800321

In [26]:
```
# making bar for females and males ratio ompared to their total in general
gender_ratio.plot(kind = "bar",
                  width =.35,
                  rot = 0,
                  color=['LightPink', 'SteelBlue'],
                  alpha = 0.8,
                  fontsize = 12,
                  figsize=(7,4))
plt.title(" percentage of females and males showed up and their percentage in genera
plt.ylabel(' Show up precentage', fontsize=15,weight='bold');
plt.xlabel('Gender', fontsize=15,weight='bold');
locations = ['Females', 'Males']
plt.xticks(np.arange(len(locations)),locations);
plt.legend(["didn't show"," showed"], fontsize=12);
```



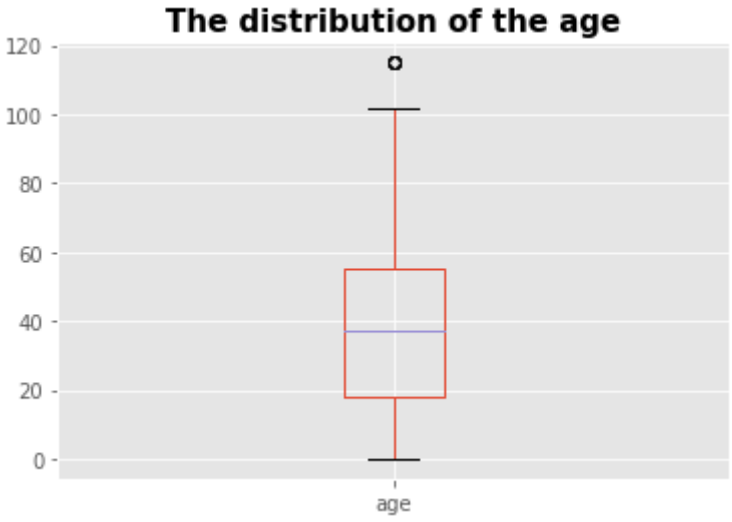
There are No relationship between the gender and showing up.

- The gender doesn't affect showing up of the patient,because the female and male almost equal.

Question 2

Does the age of patients affect why they are showup or not ?

In [27]:
```
#explore the age:
df.boxplot(column=['age'])
plt.title('The distribution of the age',fontsize=15,weight='bold' );
```



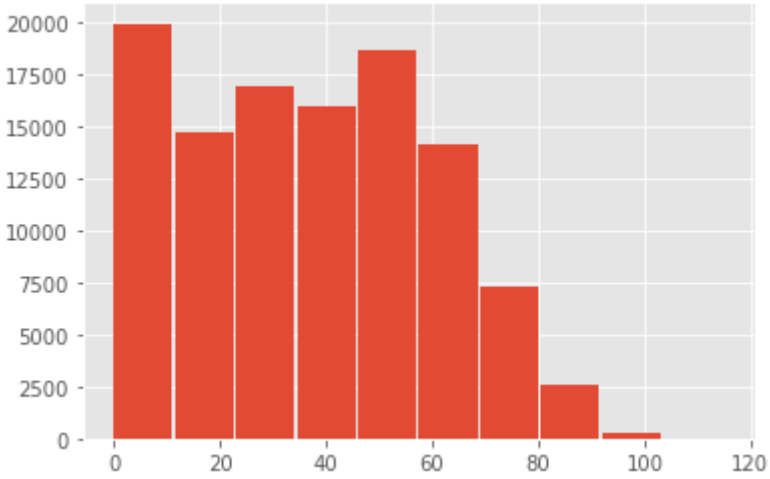
From boxplot chart we conclude that:

- The minimum of age is : 0
- The maximum of age is : 115
- The median of age is : 37
- The Median age is around 30 .

I think that the age of 0 relates to newborns

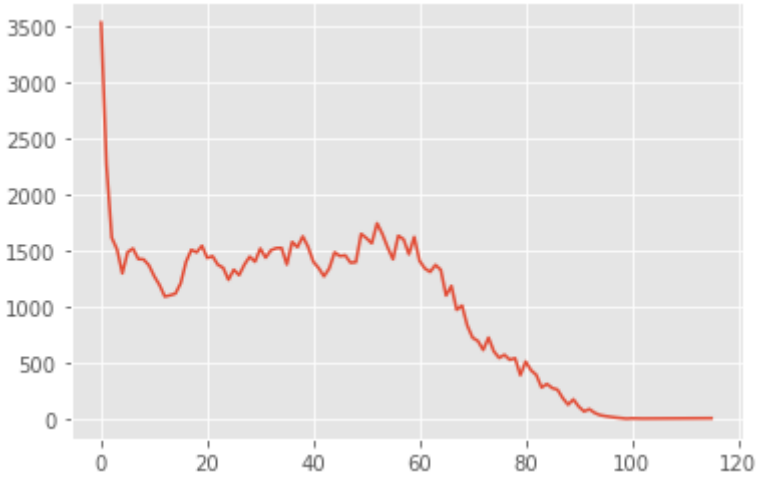
boxplot shows few datapoints as outliers we will not consider them as true outliers for this case.

In [28]:
```
plt.hist(df['age'], width=11);
```



> Age distributed equally till the age of 70. There is less patients above the age of 70.

In [29]:
```
# line chart explain the value count of "age":
df["age"].value_counts().sort_index().plot.line();
```



> The line plot above shows us that:
>
> - The largest number of patients who attended (0 age),and it doesn't make sense I think that the age of 0 relates to newborns.
> - The number of patients decreases until we reach the elderly patients, their number is very few.

In [30]:
```
#create a data frame that contains Patients under the age of 13:
df_kid = df.query('age < 13')
#create a data frame that contains Patients between the ages of 13 and 21 years:
df_teen = df.query('age > 13 and age < 21')
#create a data frame that contains Patients between the ages of 21 and 20 years:
df_adult = df.query('age > 21 and age < 50')
#create a data frame that contains Patients over 50 years old:
df_boomer = df.query('age > 50 ')
```

In [31]:
```
#create a data frame that contains attended patients under the age of 13
```

```
df_show_kid = df_show.query('age < 13')
#create a data frame that contains attended patients between the ages of 13 and 21 y
df_show_teen = df_show.query('age > 13 and age < 21')
#create a data frame that contains attended patients between the ages of 21 and 20 y
df_show_adult = df_show.query('age > 21 and age < 50')
#create a data frame that contains attended patients over 50 years old:
df_show_boomer = df_show.query('age > 50')
```

In [32]:

```
# calculate the total number of kids (show up or not ):
total_num_kid = df_kid["age"].value_counts().sum()
# calculate the total number of teens (show up or not ):
total_num_teen = df_teen["age"].value_counts().sum()
# calculate the total number of adults (show up or not ):
total_num_adult = df_adult["age"].value_counts().sum()
# calculate the total number of boomer (show up or not ):
total_num_boomer = df_boomer["age"].value_counts().sum()
```

In [33]:

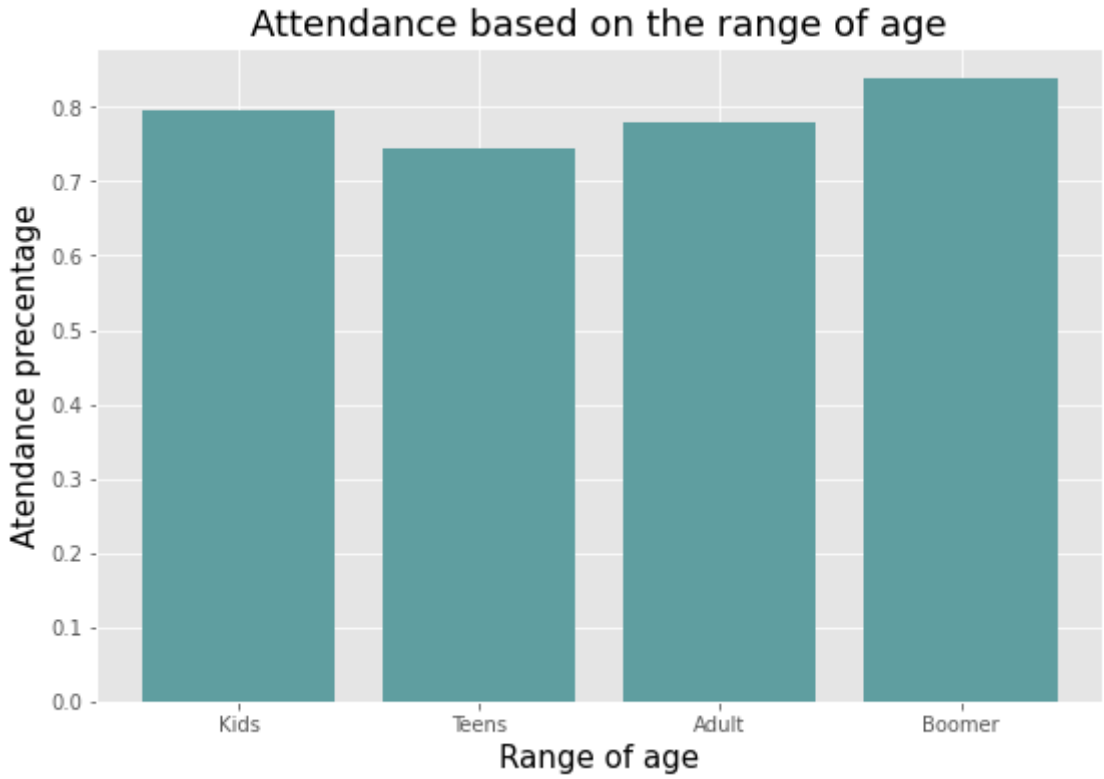
```
# calculate the  number of kids (show up):
num_kid_show = df_show_kid['age'].value_counts().sum()
# calculate the number of teens (show up):
num_teen_show = df_show_teen['age'].value_counts().sum()
# calculate the number of adults (show up):
num_adult_show = df_show_adult['age'].value_counts().sum()
# calculate the number of boomer(show up) :
num_boomer_show = df_show_boomer['age'].value_counts().sum()
```

In [34]:

```
# percentage between kids who showed up and total number of kids (show up or not):
ratio_show_kid = num_kid_show /total_num_kid
# percentage between teens who showed up and total number of teens (show up or not):
ratio_show_teen = num_teen_show /total_num_teen
# percentage between adults who showed up and total number of adults (show up or not
ratio_show_adult = num_adult_show /total_num_adult
# percentage between boomer who showed up and total number of boomer (show up or not
ratio_show_boomer = num_boomer_show /total_num_boomer
```

In [35]:

```
locations = ['Kids','Teens',"Adult","Boomer"]
hights = [ratio_show_kid ,ratio_show_teen,ratio_show_adult,ratio_show_boomer]
plt.figure(figsize=(9,6))
plt.bar(locations, hights,color="CadetBlue")
plt.title('Attendance based on the range of age', fontsize=18,color="black")
plt.ylabel('Atendance precentage', fontsize=15,color="black");
plt.xlabel('Range of age', fontsize=15,color="black");
```



> No,there are no relationship between age and show up.
>
> - As we can see in the chart the 'Boomer' are the most likely to keep their appointment.
> - come after them 'kids'and 'adults'.
> - The graph also shows that 'Teens' are the most frequently missed their appointment.
> - when the people getting older the opportunity of showing up in the appointment day augment.

Question 3

Which neighbourhood has the most no show rate?

In [36]:
```
#create a data frame that contains patients who missed their appointment.
df_no_show = df.query('show == 0')
```

In [37]:
```
df_no_show['neighbourhood'].value_counts()
```

Out[37]:
```
JARDIM CAMBURI                 1465
MARIA ORTIZ                    1219
ITARARÉ                         923
RESISTÊNCIA                     906
CENTRO                          703
                               ...
PONTAL DE CAMBURI                12
ILHA DO BOI                       3
ILHAS OCEÂNICAS DE TRINDADE       2
ILHA DO FRADE                     2
AEROPORTO                         1
Name: neighbourhood, Length: 80, dtype: int64
```

In [38]:
```
df_no_show['neighbourhood'].mode()[0]
```

Out[38]:
```
'JARDIM CAMBURI'
```

The neighborhood with the largest number of patients who miss their appointment is:

JARDIM CAMBURI

- Also most of the patients live in Jardim Camburi.

Question 4

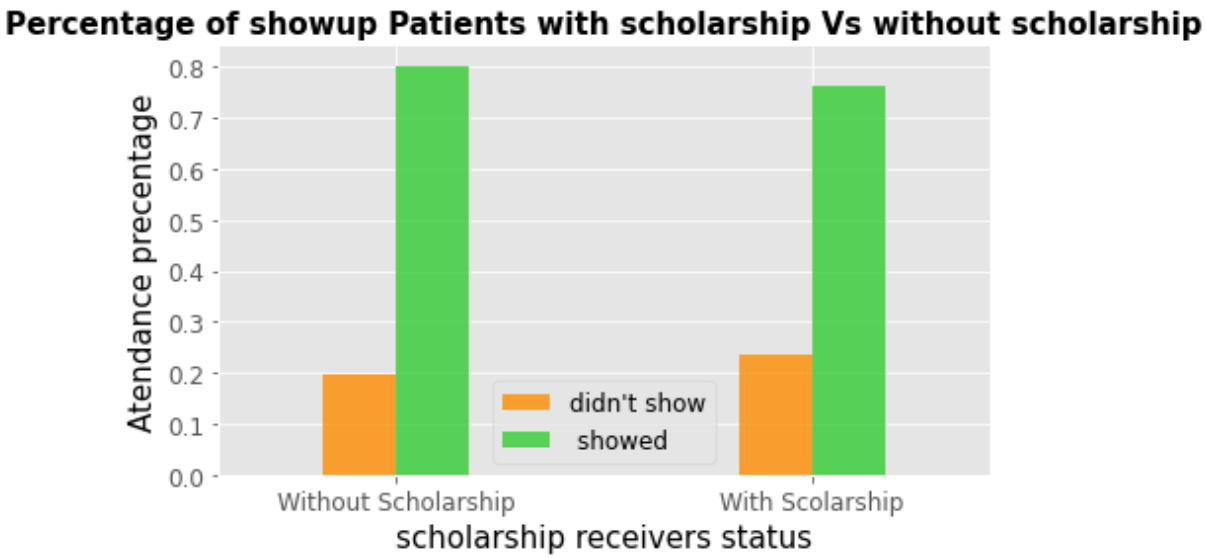
Does the patients who have scholarship show up or not?

In [39]:
```
#precentage of patients who have and haven't scholarship who showed up
scholarship_ratio = ratio_calculate(df,"scholarship","show")
scholarship_ratio
```

Out[39]:

show	0	1
scholarship		
0	0.198074	0.801926
1	0.237363	0.762637

In [40]:
```
# making bar for patients with and without scholcarship compared to their total in g
scholarship_ratio.plot(kind = "bar",
                       width =.35,
                       rot = 0,
                       color=['darkorange',"limegreen"],
                       alpha = 0.8,
                       fontsize = 12,
                       figsize=(7,4))
plt.title('Percentage of showup Patients with scholarship Vs without scholarship', f
plt.ylabel('Atendance precentage', fontsize=15,color="black");
plt.xlabel('scholarship receivers status', fontsize=15,color="black");
locations = ['Without Scholarship','With Scolarship']
plt.xticks(np.arange(len(locations)),locations);
plt.legend(["didn't show"," showed"], fontsize=12);
```



There are no relationship between patients who have scholarship and showing up.

- As we can see on the bar chart that the attendees with and without a scholarship almost equal.
- Patients who attend without scholarship are about 80%
- Patients who attend with scholarship are about 77%

Question 5

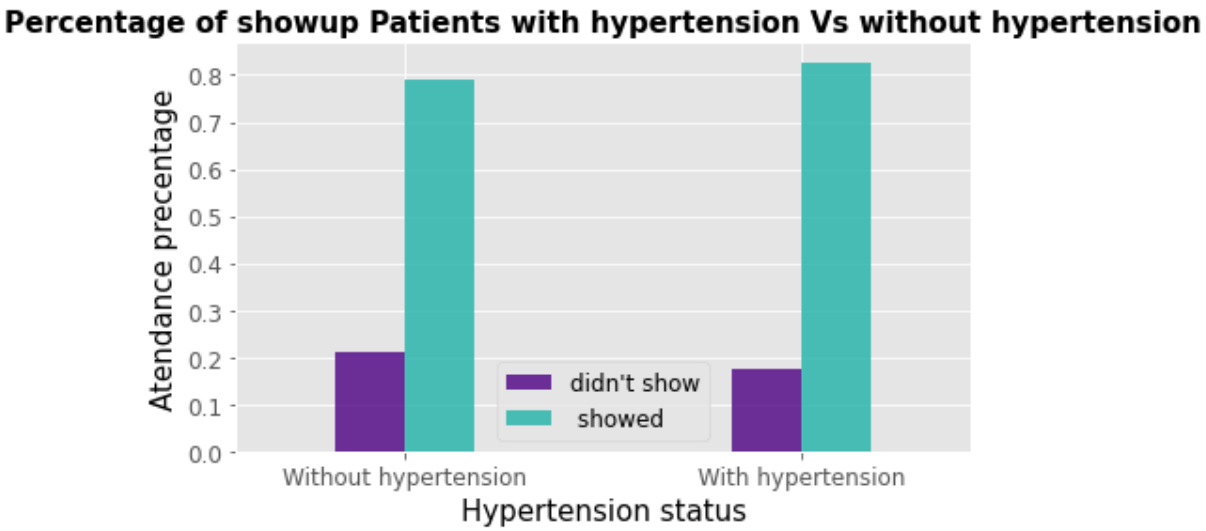
Does the hypertension affect the patient's show up?

In [41]:
```
#precentage of patients who have and haven't hypertension
hypertension_ratio = ratio_calculate(df,'hypertension',"show")
hypertension_ratio
```

Out[41]:

show	0	1
hypertension		
0	0.209039	0.790961
1	0.173020	0.826980

In [42]:
```
# making bar for patients with and without scholcarship compared to their total in g
hypertension_ratio.plot(kind = "bar",
                        width =.35,
                        rot = 0,
                        color=['Indigo',"lightseagreen"],
                        alpha = 0.8,
                        fontsize = 12,
                        figsize=(7,4))
plt.title('Percentage of showup Patients with hypertension Vs without hypertension',
plt.ylabel('Atendance precentage', fontsize=15,color="black");
plt.xlabel('Hypertension status', fontsize=15,color="black");
locations = ['Without hypertension','With hypertension']
plt.xticks(np.arange(len(locations)),locations);
plt.legend(["didn't show"," showed"], fontsize=12);
```



There are No relationship between patients who have hypertension and showing up.

- As we can see on the bar chart that the attendees patients with and without hypertension are approximately equal

Question 6

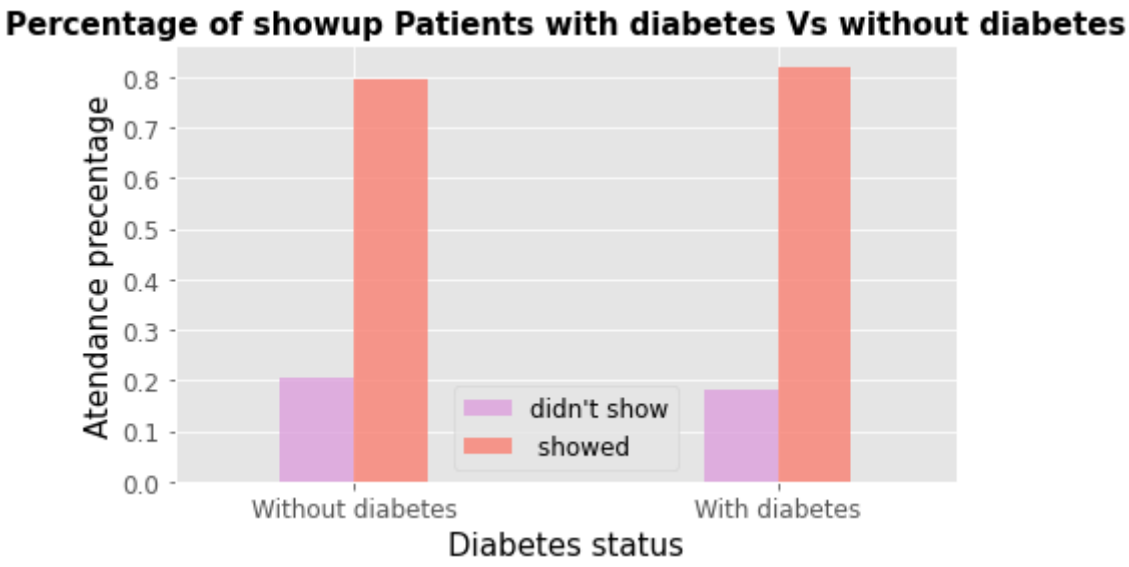
Does the diabetes affect the patient's show up?

In [43]:
```
#precentage of patients who have and haven't diabetes
diabetes_ratio = ratio_calculate(df,'diabetes',"show")
diabetes_ratio
```

Out[43]:

show	0	1
diabetes		
0	0.203630	0.796370
1	0.180033	0.819967

In [44]:
```
# making bar for patients with and without diabetes compared to their total in gener
diabetes_ratio.plot(kind = "bar",
                    width =.35,
                    rot = 0,
                    color=['Plum',"salmon"],
                    alpha = 0.8,
                    fontsize = 12,
                    figsize=(7,4))
plt.title('Percentage of showup Patients with diabetes Vs without diabetes', fontsiz
plt.xlabel('Diabetes status', fontsize=15,color="black");
plt.ylabel('Atendance precentage', fontsize=15,color="black");
locations = ['Without diabetes','With diabetes']
plt.xticks(np.arange(len(locations)),locations);
plt.legend(["didn't show"," showed"], fontsize=12);
```



> There are no relationship between patients who have diabetes and showing up.
>
> - As we can see on the bar chart that the attendees patients with and without diabetes are approximately equal.

Question 7

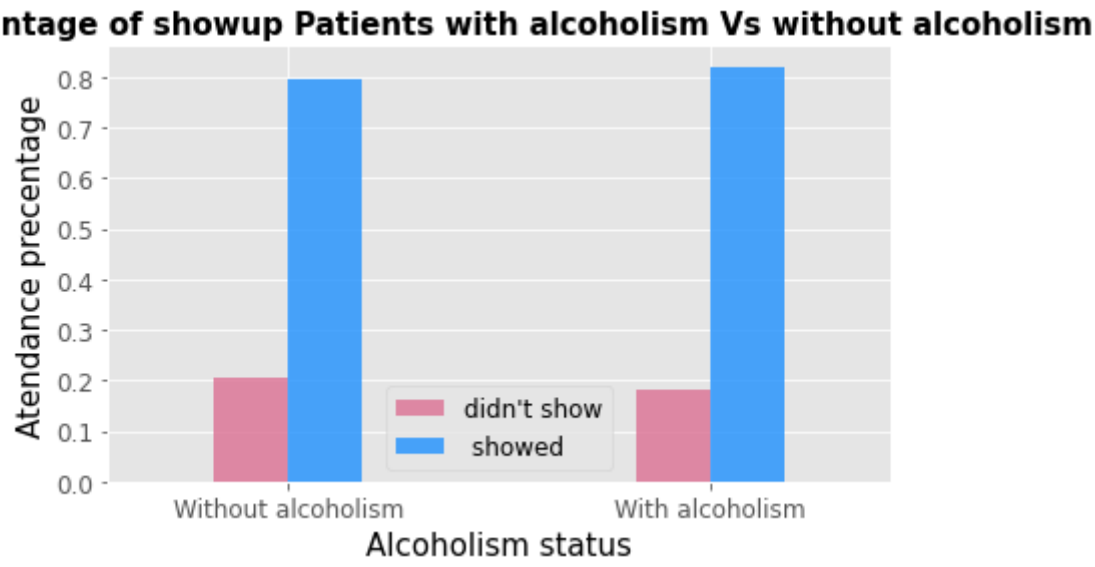
Does the alcoholism affect the patient's show up?

In [45]:
```
#precentage of patients who have and haven't Alcoholism
alcoholism_ratio = ratio_calculate(df,'alcoholism',"show")
alcoholism_ratio
```

Out[45]:

show	0	1
alcoholism		
0	0.201948	0.798052
1	0.201488	0.798512

In [46]:
```
# making bar for patients with and without scholcarship compared to their total in g
diabetes_ratio.plot(kind = "bar",
                    width =.35,
                    rot = 0,
                    color=['PaleVioletRed',"dodgerblue"],
                    alpha = 0.8,
                    fontsize = 12,
                    figsize=(7,4))
plt.title('Percentage of showup Patients with alcoholism Vs without alcoholism', fon
plt.ylabel('Atendance precentage', fontsize=15,color="black");
plt.xlabel('Alcoholism status', fontsize=15,color="black");
locations = ['Without alcoholism','With alcoholism']
plt.xticks(np.arange(len(locations)),locations);
plt.legend(["didn't show"," showed"], fontsize=12);
```



> There are no relationship between patients who have alcoholism and showing up.
>
> - As we can see on the bar chart that the attendees patients with and without alcoholism are equal.

Question 8

Does the handicap affect the patient's show up?

In [47]:
```
#chick up the unique values in the handicap column:
df["handicap"].nunique()
```

Out[47]:
```
5
```

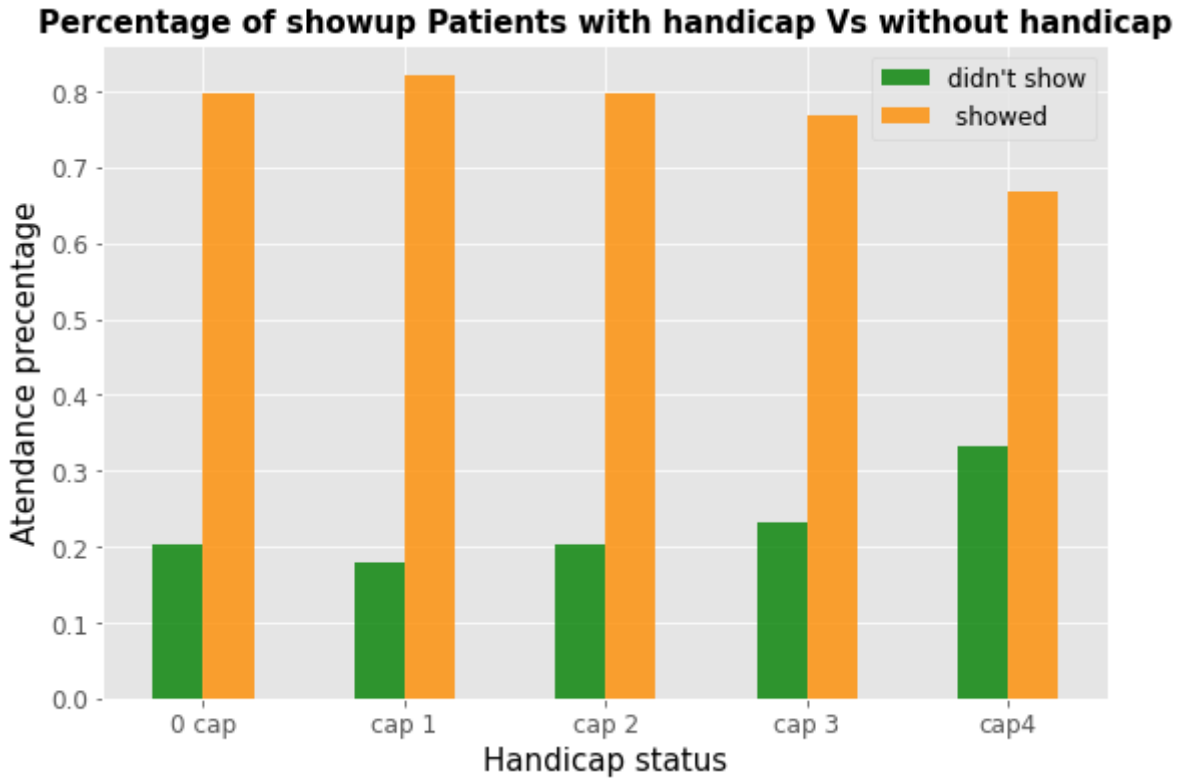
In [48]:
```
# find out the value counts of the handicap column:
df["handicap"].value_counts()
```

Out[48]:
```
0    108285
1      2042
2       183
3        13
4         3
Name: handicap, dtype: int64
```

In [49]:
```
#precentage of patients who have and haven't Handicap
handicap_ratio = ratio_calculate(df,'handicap', 'show')
```

In [50]:
```
# making bar for patients with and without handicap compared to their total in gener
handicap_ratio.plot(kind = "bar",
                    width =.5,
                    rot = 0,
                    color=['green',"DarkOrange"],
                    alpha = 0.8,
                    fontsize = 12,
                    figsize=(9,6))
plt.title('Percentage of showup Patients with handicap Vs without handicap', fontsiz
plt.ylabel('Atendance precentage', fontsize=15,color="black");
plt.xlabel('Handicap status', fontsize=15,color="black");
locations = ['0 cap','cap 1',"cap 2","cap 3", "cap4"]
```

```
plt.xticks(np.arange(len(locations)),locations);
plt.legend(["didn't show"," showed"], fontsize=12);
```



In handicap coulmn 0 means not handicapped,(1,2,3,4) means the person is handicapped.

> There are 'No relationship' between patients who have handicap and showing up.
>
> - As we can see on the bar chart that:
> - The patients with 'one handicap' is about 82% . follow them,
> - Ptients with 'No handicap' and 'two handicap' are about 80 % . follow them,
> - Patients with 'three handicap' is about 77%.
> - At the latest patients with 'four handicap'. The ratio among all patients with or without a disability is close

Question 9

Did the patients receiving sms keep their appointment?

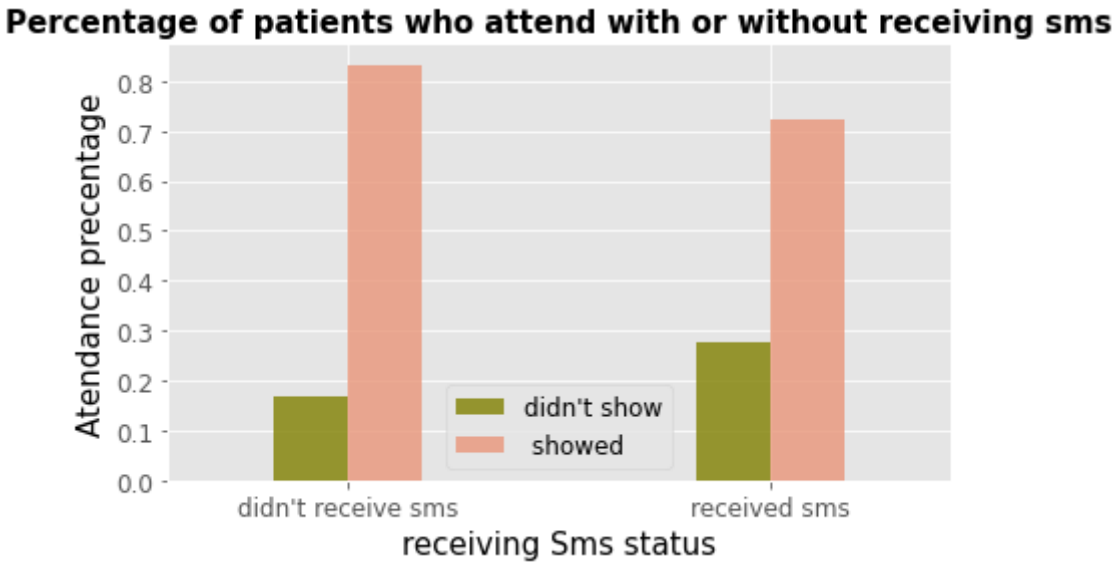
In [51]:
```
#precentage of patients who recieved and didn't receive sms
sms_received_ratio = ratio_calculate(df,'sms_received',"show")
sms_received_ratio
```

Out[51]:

show	0	1
sms_received		
0	0.167035	0.832965
1	0.275745	0.724255

In [52]:

```
# making bar for patients who recieved and didn't receive smscompared to their total
sms_received_ratio.plot(kind = "bar",
                        width =.35,
                        rot = 0,
                        color=['olive',"darksalmon"],
                        alpha = 0.8,
                        fontsize = 12,
                        figsize=(7,4))
plt.title('Percentage of patients who attend with or without receiving sms',fontsize
plt.xlabel('receiving Sms status',fontsize=15,color="black")
plt.ylabel('Atendance precentage',fontsize=15,color="black");
locations = ["didn't receive sms",'received sms']
plt.xticks(np.arange(len(locations)),locations);
plt.legend(["didn't show"," showed"], fontsize=12);
```



> There are no relationship between patients who have receiving SMS and showing up.
>
> - As we can see on the bar chart that the difference between attendees patients with and without receiving SMS is not far.

Conclusions

- Percentage of patients who show up on their appointments represents 79.8%
- Percentage of patients who Don't show up on their appointments represents 20.2%
- Showing rate for men and women are similar
- "JARDIM CAMBURI" is the most frequent place.
- Older patients are more commited to their appointments' schedules than younger ones.
- > When it comes to show up, there is no effect of these factors:
- Age :possibility of showing up increase when the people getting older.
- Being diabetic or not.

- Receiving SMS or not.
- Having the scholarship or not.
- Being alcoholic or not.

Limitations

- Age column contain value of "0" and it doesn't make sense I think that the age of 0 relates to newborns.
- Most of the variables are categorical, which doesn't allow for a high level of statistical method.
- Some information about the columns of the data set was unclear the time details in the ScheduledDay & AppointmentDay coulmns.
- Handicap column has five different values(0,1,2,3,4),the value of (0) means not handicapped, (1,2,3,4) means the person is handicapped.
- some patients who marked as no show up, in real they may show up but on another day that confused me.