In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from jupyterthemes import jtplot
jtplot.style(theme='onedork')
sns.set(style='white')
```

In [2]:

```
df = pd.read_csv('./KaggleV2-May-2016.csv')
df.head()
```

Out[2]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA
1	5.589980e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA
2	4.262960e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA
3	8.679510e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI
4	8.841190e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 17 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 
 14  ScheduledDay.1    110527 non-null  object 
 15  AppointmentDay.1  110527 non-null  object 
 16  watingday         110527 non-null  object 
dtypes: float64(1), int64(8), object(8)
memory usage: 14.3+ MB
```

In [4]:

```
# 데이터 중복 여부 확인
df.duplicated().sum()
```

Out[4]: 0

In [5]:

```
# 데이터 null값 확인
print(df.isnull().sum())
print('-'*30)
print(df.nunique())
```

```
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
ScheduledDay.1      0
AppointmentDay.1    0
watingday           0
dtype: int64
------------------------------
PatientId            61744
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
ScheduledDay.1         111
AppointmentDay.1        27
watingday              131
dtype: int64
```

In [6]:

```
df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay']).dt.date.astype('datetime64[ns
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay']).dt.date.astype('datetime6
df['WeekDay']=df['AppointmentDay'].dt.weekday
df['Waiting']=(df['AppointmentDay']-df['ScheduledDay']).dt.days
df['Waiting_str']=(df['AppointmentDay']-df['ScheduledDay']).dt.days
df['Age_str'] = df['Age']
```

In [7]:

```
columns = df.columns
```

In [8]:

```
waiting2 = df.groupby(by=['Waiting_str','No-show'])

waiting2 = waiting2.count()['PatientId'].unstack()

waiting2.fillna(value=0, inplace=True)
waiting2.reset_index(drop=False, inplace=True)
waiting2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 131 entries, 0 to 130
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Waiting_str  131 non-null    int64  
 1   No           131 non-null    float64
 2   Yes          131 non-null    float64
dtypes: float64(2), int64(1)
memory usage: 3.2 KB
```

In [9]:

```
Waitingsort = pd.Series(['Same day: 0', 'Week: 1-7','Month: 8-30', 'Quarter: 31-92',
```

In [10]:

```
waiting2['Waiting_str'] = pd.cut(waiting2.Waiting_str, bins = [-1,0,7,30,92,181,500],
df['Waitingsort'] = pd.cut(df.Waiting_str, bins = [-1,0,7,30,92,181,500], labels=Wait
```

In [11]:

```
waiting2 = waiting2.groupby('Waiting_str').sum()
```

In [12]:

```
waiting2['No-showing rate'] = (waiting2.Yes / waiting2.No)*100
```

In [13]:

```
waiting2
```

Out[13]:

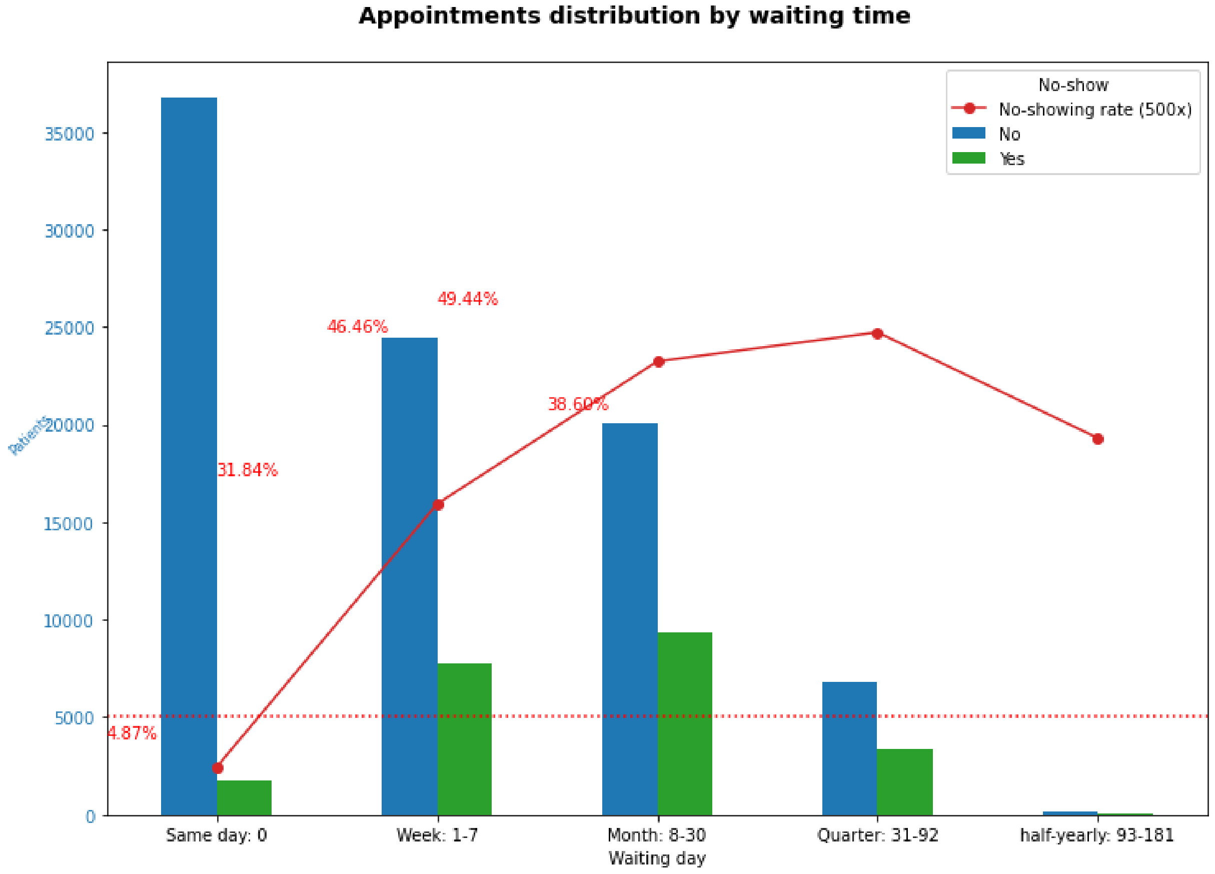
No-show	No	Yes	No-showing rate
Waiting_str			
Same day: 0	36771.0	1792.0	4.873406
Week: 1-7	24413.0	7772.0	31.835497
Month: 8-30	20071.0	9325.0	46.460067
Quarter: 31-92	6839.0	3381.0	49.437052
half-yearly: 93-181	114.0	44.0	38.596491
Very long: >181	0.0	0.0	NaN

In [14]:

```
eda_waiting2 = waiting2.copy()
eda_waiting2.reset_index(drop=False, inplace=True)
eda_waiting2.drop(5, inplace=True)


# 'No-showing rate'를 백분율 값이있는 문자열로 변환
eda_waiting2['No-show percentual'] = eda_waiting2['No-showing rate'].apply(lambda x:
# 동일한 척도로 그려지기 위해 비율 값에 500 배를 곱함
eda_waiting2['No-showing rate (500x)'] = eda_waiting2['No-showing rate']*500
```

In [15]:

```
## 그래프 매개 변수 설정 :https://codetorial.net/matplotlib/two_types_of_graphs.html
fig1, ax = plt.subplots(figsize=[12,8])  # 그래프 창 크기를 정의
fig1.subplots_adjust(top=0.92)
plt.suptitle('Appointments distribution by waiting time ', fontsize=14, fontweight='b

colors = ['tab:blue', 'tab:green', 'tab:red']  # 사용할 색상을 정의

ax.set_ylabel('Patients', color=colors[0], fontsize=8,rotation=45) #y 축 색상 및 레이
ax.tick_params(axis='y', labelcolor=colors[0])

## 꺾은 선형 차트 그리기 :
eda_waiting2[['Waiting_str', 'No-showing rate (500x)']].plot(x='Waiting_str', linesty
# 선 차트 마커 라벨 설정
x = ax.get_xticks()  #Getting the x-axis ticks to plot the label
for a,b,c in zip(x,eda_waiting2['No-showing rate (500x)'], eda_waiting2['No-show perc
    plt.text(a,b+1500,c, color='red', fontsize=10)
plt.axhline(5000, color="red", linestyle=":")


## 막대 차트 플로팅 :
eda_waiting2[['Waiting_str', 'No', 'Yes']].plot(x='Waiting_str', kind='bar', ax=ax, c
ax.set_xticklabels(ax.get_xticklabels(), rotation=360,fontsize=10)
ax.set_xlabel('Waiting day', fontsize=10)   #y 축 색상 및 레이블 설정

plt.show()
```



In [16]:

```
sort_age = pd.Series(['underage_age: 0-19', 'Adult: 20-39', 'Adult2: 40-59', 'Senior:
#나이대별로 분류
```

In [17]:

```
age_str = df.groupby(by=['Age_str','No-show'])
```

In [18]:

```
age_str = age_str.count()['PatientId'].unstack()
```

In [19]:

```
age_str.fillna(value=0, inplace=True)
age_str.reset_index(drop=False, inplace=True)
age_str.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 104 entries, 0 to 103
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Age_str  104 non-null    int64
 1   No       104 non-null    float64
 2   Yes      104 non-null    float64
dtypes: float64(2), int64(1)
memory usage: 2.6 KB
```

In [20]:

```
age_str['Age_str'] = pd.cut(age_str.Age_str, bins = [-1,19,39,59,79,99,150], labels=s
df['sort_age'] = pd.cut(df.Age_str, bins = [-1,19,39,59,79,99,150], labels=sort_age)
```

In [21]:

```
age_str = age_str.groupby('Age_str').sum()
age_str['No-showing rate'] = (age_str.Yes / age_str.No)*100
age_str
```

Out[21]:

No-show	No	Yes	No-showing rate
Age_str			
underage_age: 0-19	23670.0	6741.0	28.479087
Adult: 20-39	22190.0	6680.0	30.103650

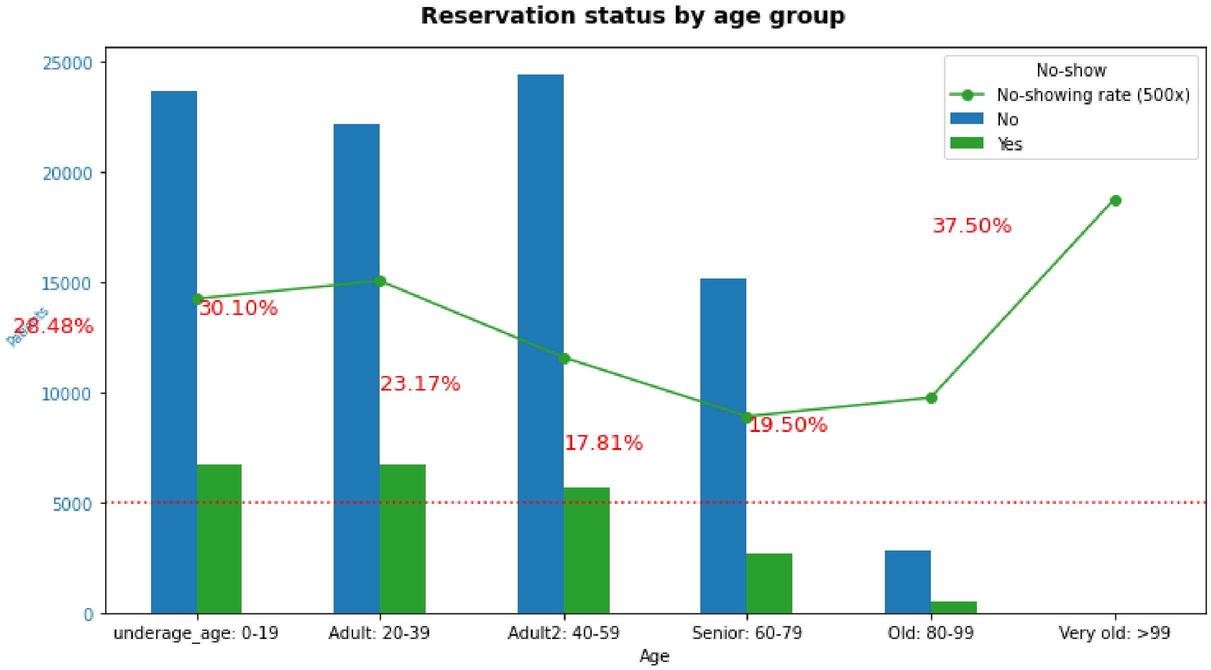
No-show Age_str	No	Yes	No-showing rate
Adult2: 40-59	24416.0	5656.0	23.165138
Senior: 60-79	15118.0	2692.0	17.806588
Old: 80-99	2805.0	547.0	19.500891
Very old: >99	8.0	3.0	37.500000

In [22]:
```
# 데이터 프레임 조정
edit_age_str = age_str.copy()
edit_age_str.reset_index(drop=False, inplace=True) #인덱스를 열로
# 'No-showing rate'를 백분율 값이있는 문자열로 변환
edit_age_str['No-show %'] = edit_age_str['No-showing rate'].apply(lambda x: '{0:.2f}%
# 동일한 척도로 그려지기 위해 비율 값에 500 배를 곱함
edit_age_str['No-showing rate (500x)'] = edit_age_str['No-showing rate']*500
edit_age_str
```

Out[22]:

No-show	Age_str	No	Yes	No-showing rate	No-show %	No-showing rate (500x)
0	underage_age: 0-19	23670.0	6741.0	28.479087	28.48%	14239.543726
1	Adult: 20-39	22190.0	6680.0	30.103650	30.10%	15051.825146
2	Adult2: 40-59	24416.0	5656.0	23.165138	23.17%	11582.568807
3	Senior: 60-79	15118.0	2692.0	17.806588	17.81%	8903.294087
4	Old: 80-99	2805.0	547.0	19.500891	19.50%	9750.445633
5	Very old: >99	8.0	3.0	37.500000	37.50%	18750.000000

In [23]:
```
## 그래프 매개 변수 설정 :
fig1, ax = plt.subplots(figsize=[12,6]) # 그래프 창 크기를 정의
fig1.subplots_adjust(top=0.92)
plt.suptitle('Reservation status by age group ', fontsize=14, fontweight='bold')
#plt.legend(shadow=True,borderpad=1)
colors = ['tab:blue', 'tab:green', 'tab:red'] # 사용할 색상을 정의

ax.set_ylabel('Patients', color=colors[0], fontsize=8,rotation=45) #y 축 색상 및 레이
ax.tick_params(axis='y', labelcolor=colors[0])

## 꺾은 선형 차트 그리기 :
edit_age_str[['Age_str', 'No-showing rate (500x)']].plot(x='Age_str', linestyle='-',
# 선 차트 마커 라벨 설정
x = ax.get_xticks() #x 축 눈금을 가져와 레이블 플로팅하기
for a,b,c in zip(x,edit_age_str['No-showing rate (500x)'], edit_age_str['No-show %'])
    plt.text(a,b-1500,c, color='red', fontsize=13)



## 막대 차트 플로팅 :
edit_age_str[['Age_str', 'No', 'Yes']].plot(x='Age_str', kind='bar', ax=ax, color=col
ax.set_xticklabels(ax.get_xticklabels(), rotation=360,fontsize=10)
ax.set_xlabel('Age', fontsize=10) #y 축 색상 및 레이블 설정
plt.axhline(5000, color="red", linestyle=":")
plt.show()
#위의 차트를 통해 대기 시간이 길어질수록 노쇼 비율이 증가
```



In [24]:
```
df.drop(df[df['Neighbourhood'] == 'ILHAS OCEÂNICAS DE TRINDADE'].index, inplace=True
```

In [25]:
```
df['No_show'] = df['No-show']
```

In [26]:
```
neighborhood = df.Neighbourhood.unique()
neighborhood.sort()
nei_hos = df.groupby(by='Neighbourhood').No_show.value_counts().sort_index()
```

In [27]:
```
## ## 데이터 조작 :
nei_hos = nei_hos.unstack()  #groupby 객체를 데이터 셋으로 변환
nei_hos.fillna(value=0, inplace=True)  #NaN 값을 0으로 바꾸기
print(nei_hos.head(3))

def get_total(dataframe):

    return dataframe.sum(axis=1)

def df_row_normalize(dataframe):

    return dataframe.div(dataframe.sum(axis=1), axis=0)
```

```
No_show              No     Yes
Neighbourhood
AEROPORTO           7.0     1.0
ANDORINHAS       1741.0   521.0
ANTÔNIO HONÓRIO   221.0    50.0
```

In [28]:
```
## 미리 정의 된 함수를 사용하여 데이터 정규화 :
normalnei = df_row_normalize(nei_hos)
print(normalnei.head(3))


nei_hos['Total'] = get_total(nei_hos)
normalnei['Total'] = get_total(normalnei)
```

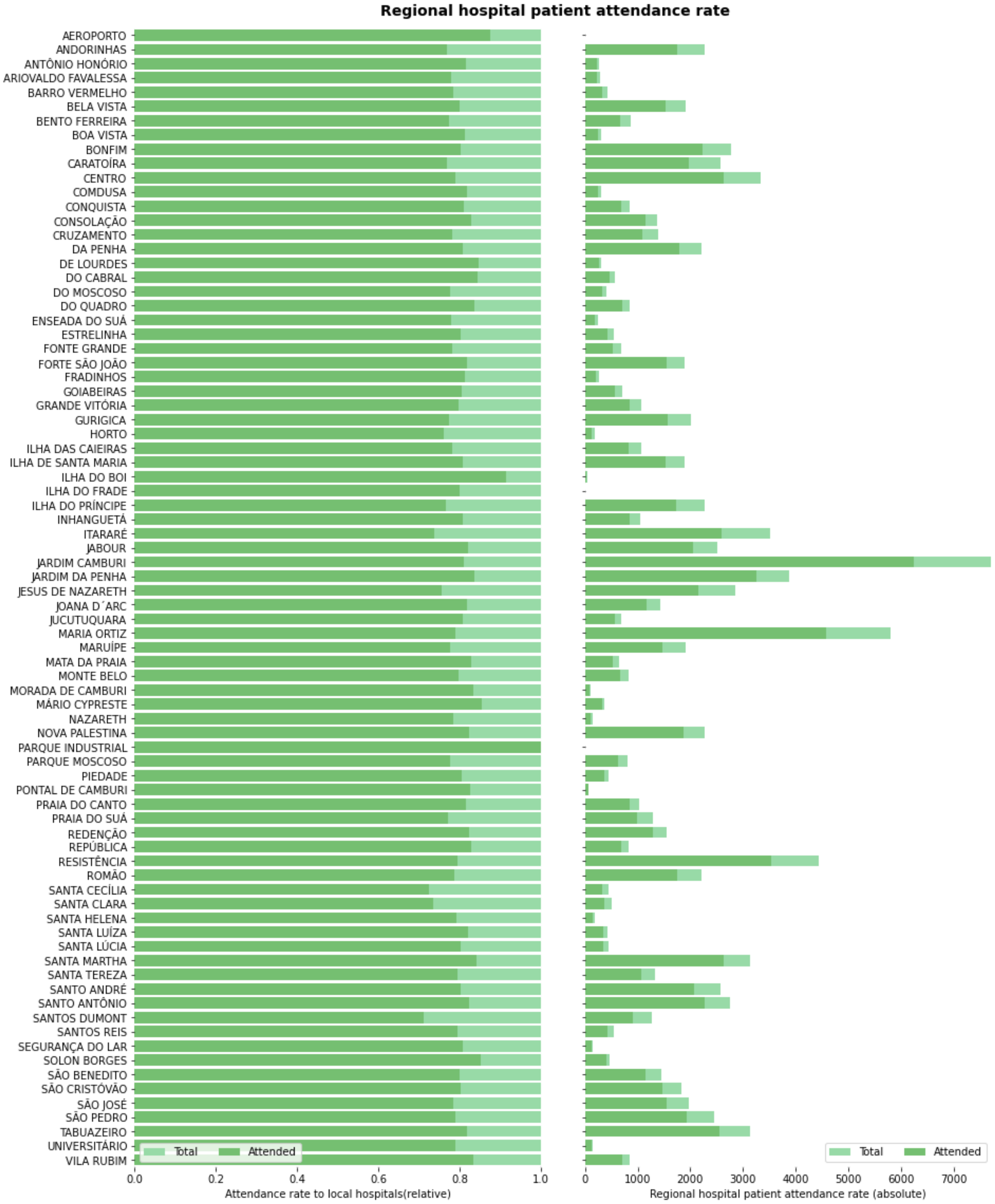
```
No_show                No       Yes
Neighbourhood
AEROPORTO        0.875000  0.125000
ANDORINHAS       0.769673  0.230327
ANTÔNIO HONÓRIO  0.815498  0.184502
```

In [29]:
```
# 'neighbourhood'인덱스 재설정 및 열로 만들기
```

```
nei_hos.reset_index(inplace=True)
normalnei.reset_index(inplace=True)
```

In [30]:

```
fig2, (ax1, ax2) = plt.subplots(1,2, figsize=(12,16), sharey=False)
fig2.tight_layout()
plt.suptitle('Regional hospital patient attendance rate', fontsize=14, fontweight='bo
fig2.subplots_adjust(top=0.96)

## 지역병원별로 노쇼(상대)
#총 예약
sns.set_color_codes("pastel")
sns.barplot(x="Total", y="Neighbourhood", data=normalnei, label="Total", color="g", a
#참석한 예약
sns.set_color_codes("muted")
sns.barplot(x="No", y="Neighbourhood", data=normalnei, label="Attended", color="g", a
## 범례 ,축 레이블 추가
ax1.legend(ncol=2, loc="lower left", frameon=True)
ax1.set(xlim=(0, 1), ylabel="", xlabel="Attendance rate to local hospitals(relative)"
sns.despine(left=True, bottom=True,ax=ax1)

# 지역병원별로 노쇼(절대)
#총 예약
sns.set_color_codes("pastel")
sns.barplot(x="Total", y="Neighbourhood", data=nei_hos, label="Total", color="g",ax=a
# 참석한 예약
sns.set_color_codes("muted")
sns.barplot(x="No", y="Neighbourhood", data=nei_hos, label="Attended", color="g", ax=
# 범례 ,축 레이블 추가
ax2.legend(ncol=2, loc="lower right", frameon=True)
ax2.set(xlim=(0, 7720), ylabel="", xlabel="Regional hospital patient attendance rate 
ax2.set_yticklabels([''])
sns.despine(left=True, bottom=True, ax=ax2)


plt.show()
```



In [31]:

```
df['No-show'].replace("Yes", 1, inplace=True)
df['No-show'].replace("No", 0, inplace=True)
df['Past'] = df.sort_values(['ScheduledDay']).groupby(['PatientId'])['No-show'].cumsu
```

In [ ]:

In [32]:

```
columns = df.columns

for i in range(len(columns)):
    print(df.iloc[:,i].value_counts(),'₩n','-'*30)
```

```
8.221460e+14    88
9.963767e+10    84
2.688610e+13    70
3.353480e+13    65
6.264200e+12    62
                ..
```

```
2.471290e+14     1
4.999710e+13     1
8.483290e+14     1
1.338260e+11     1
3.367740e+13     1
Name: PatientId, Length: 61742, dtype: int64
 ------------------------------
5769215    1
5651786    1
5733701    1
5707080    1
5702986    1
          ..
5686470    1
5582192    1
5586290    1
5584243    1
5771266    1
Name: AppointmentID, Length: 110525, dtype: int64
 ------------------------------
F    71838
M    38687
Name: Gender, dtype: int64
 ------------------------------
2016-05-03    4238
2016-05-02    4216
2016-05-16    4120
2016-05-05    4095
2016-05-10    4024
              ...
2016-04-09       1
2015-11-10       1
2016-01-19       1
2016-06-04       1
2016-03-19       1
Name: ScheduledDay, Length: 111, dtype: int64
 ------------------------------
2016-06-06    4692
2016-05-16    4613
2016-05-09    4520
2016-05-30    4514
2016-06-08    4479
2016-05-11    4474
2016-06-01    4464
2016-06-07    4416
2016-05-12    4394
2016-05-02    4376
2016-05-18    4373
2016-05-17    4372
2016-06-02    4310
2016-05-10    4308
2016-05-31    4279
2016-05-05    4273
2016-05-19    4270
2016-05-03    4256
2016-05-04    4168
2016-06-03    4090
2016-05-24    4009
2016-05-13    3985
2016-05-25    3909
2016-05-06    3879
2016-05-20    3828
2016-04-29    3235
2016-05-14      39
Name: AppointmentDay, dtype: int64
 ------------------------------
 0      3539
 1      2273
 52     1746
 49     1652
 53     1651
        ...
 115       5
 100       4
 102       2
 99        1
-1         1
Name: Age, Length: 104, dtype: int64
 ------------------------------
JARDIM CAMBURI       7717
MARIA ORTIZ          5805
RESISTÊNCIA          4431
JARDIM DA PENHA      3877
ITARARÉ              3514
                     ...
PONTAL DE CAMBURI      69
ILHA DO BOI            35
ILHA DO FRADE          10
AEROPORTO               8
PARQUE INDUSTRIAL       1
Name: Neighbourhood, Length: 80, dtype: int64
 ------------------------------
0    99664
1    10861
Name: Scholarship, dtype: int64
 ------------------------------
0    88724
1    21801
Name: Hipertension, dtype: int64
 ------------------------------
0    102582
1      7943
Name: Diabetes, dtype: int64
 ------------------------------
0    107165
1      3360
Name: Alcoholism, dtype: int64
 ------------------------------
0    108284
1      2042
2       183
3        13
4         3
Name: Handcap, dtype: int64
 ------------------------------
0    75043
1    35482
Name: SMS_received, dtype: int64
 ------------------------------
0    88208
1    22317
Name: No-show, dtype: int64
 ------------------------------
2016-05-03    4238
2016-05-02    4216
2016-05-16    4120
2016-05-05    4095
2016-05-10    4024
              ...
2016-01-27       1
2016-04-09       1
2016-03-05       1
2015-12-03       1
2016-01-04       1
Name: ScheduledDay.1, Length: 111, dtype: int64
 ------------------------------
2016-06-06    4692
2016-05-16    4613
```

```
2016-05-09    4520
2016-05-30    4514
2016-06-08    4479
2016-05-11    4474
2016-06-01    4464
2016-06-07    4416
2016-05-12    4394
2016-05-02    4376
2016-05-18    4373
2016-05-17    4372
2016-06-02    4310
2016-05-10    4308
2016-05-31    4279
2016-05-05    4273
2016-05-19    4270
2016-05-03    4256
2016-05-04    4168
2016-06-03    4090
2016-05-24    4009
2016-05-13    3985
2016-05-25    3909
2016-05-06    3879
2016-05-20    3828
2016-04-29    3235
2016-05-14      39
Name: AppointmentDay.1, dtype: int64
 ------------------------------
 -       38563
 2        6725
 4        5290
 1        5213
 7        4906
         ...
 139         1
 146         1
 127         1
 123         1
 125         1
Name: watingday, Length: 131, dtype: int64
 ------------------------------
2    25867
1    25640
0    22715
4    19017
3    17247
5       39
Name: WeekDay, dtype: int64
 ------------------------------
 0      38563
 2       6725
 4       5290
 1       5213
 7       4906
        ...
 117        1
 146        1
 82         1
-6          1
 127        1
Name: Waiting, Length: 131, dtype: int64
 ------------------------------
 0      38563
 2       6725
 4       5290
 1       5213
 7       4906
        ...
 117        1
 146        1
 82         1
-6          1
 127        1
Name: Waiting_str, Length: 131, dtype: int64
 ------------------------------
 0      3539
 1      2273
 52     1746
 49     1652
 53     1651
        ... 
 115       5
 100       4
 102       2
 99        1
-1         1
Name: Age_str, Length: 104, dtype: int64
 ------------------------------
Same day: 0           38563
Week: 1-7             32185
Month: 8-30           29394
Quarter: 31-92        10220
half-yearly: 93-181     158
Very long: >181           0
Name: Waitingsort, dtype: int64
 ------------------------------
underage_age: 0-19    30411
Adult2: 40-59         30070
Adult: 20-39          28870
Senior: 60-79         17810
Old: 80-99             3352
Very old: >99            11
Name: sort_age, dtype: int64
 ------------------------------
No     88208
Yes    22317
Name: No_show, dtype: int64
 ------------------------------
0     74926
1     27553
2      5554
3      1508
4       485
5       220
6       105
7        51
8        35
9        28
10       19
11       19
12        6
13        5
14        4
15        3
16        2
17        1
18        1
Name: Past, dtype: int64
 ------------------------------
```

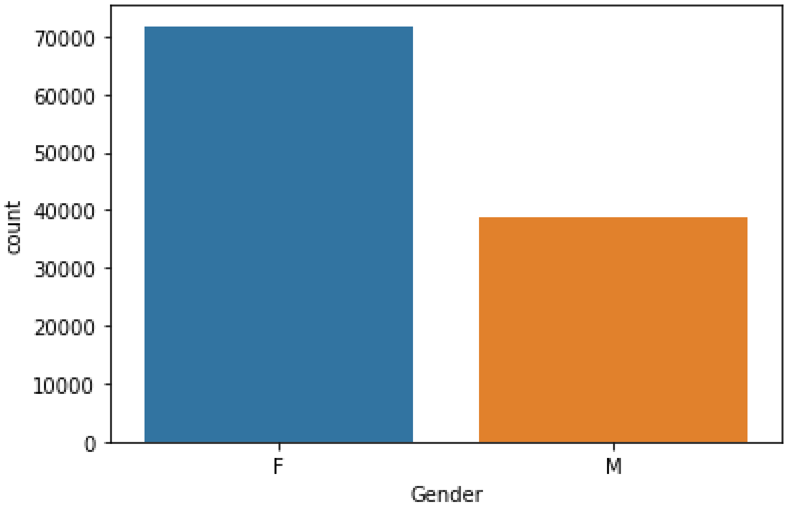
In [33]:

```
df = df[df['Age']>=0]
df = df[df['Waiting']>=0]

df.drop(['PatientId','AppointmentID'], axis=1, inplace=True)
```

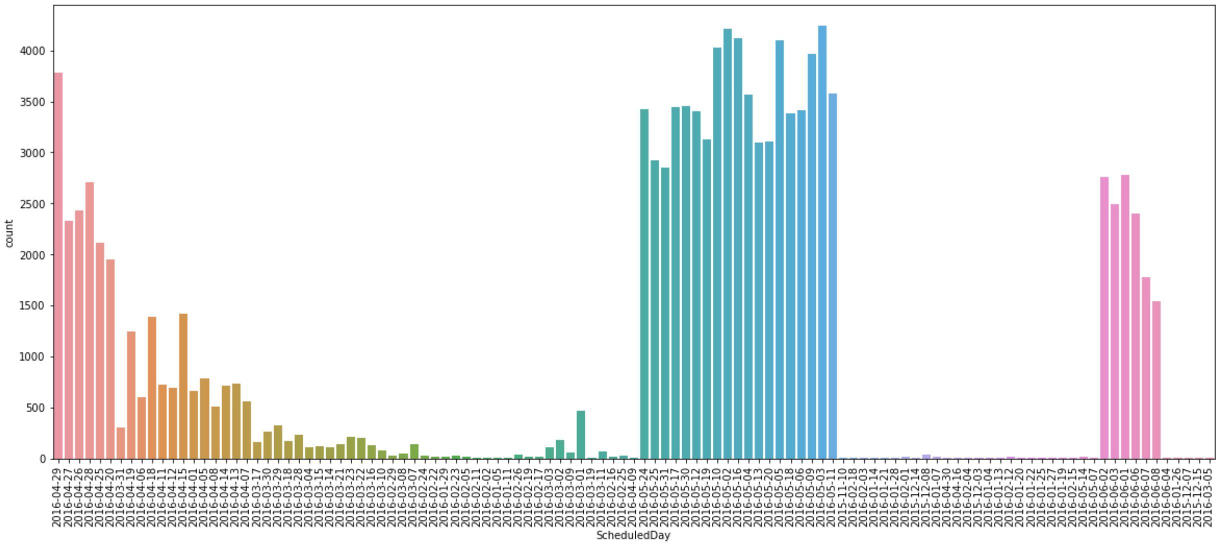
In [34]:

```
# 각 변수별로 value 확인 및 그래프
columns = df.columns
```

```
# ScheduledDay, AppointmentDay 날짜형식변환
df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay']).dt.date.astype('datetime64[ns
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay']).dt.date.astype('datetime6

for i in range(len(columns)):
    if i>=1 and i<=4 or i==13:
        if i==1 or i==2 or i==4:
            plt.figure(figsize=(20,8))
            sns.countplot(data=df, x=columns[i])
            plt.xticks(rotation=90)
            plt.show()
        else:
            plt.figure(figsize=(25,4))
            sns.countplot(data=df, x=columns[i])
            plt.show()
    else:
        sns.countplot(data=df, x=columns[i])
        plt.show()

    print(df.iloc[:,i].value_counts(),'₩n','-'*30)
```

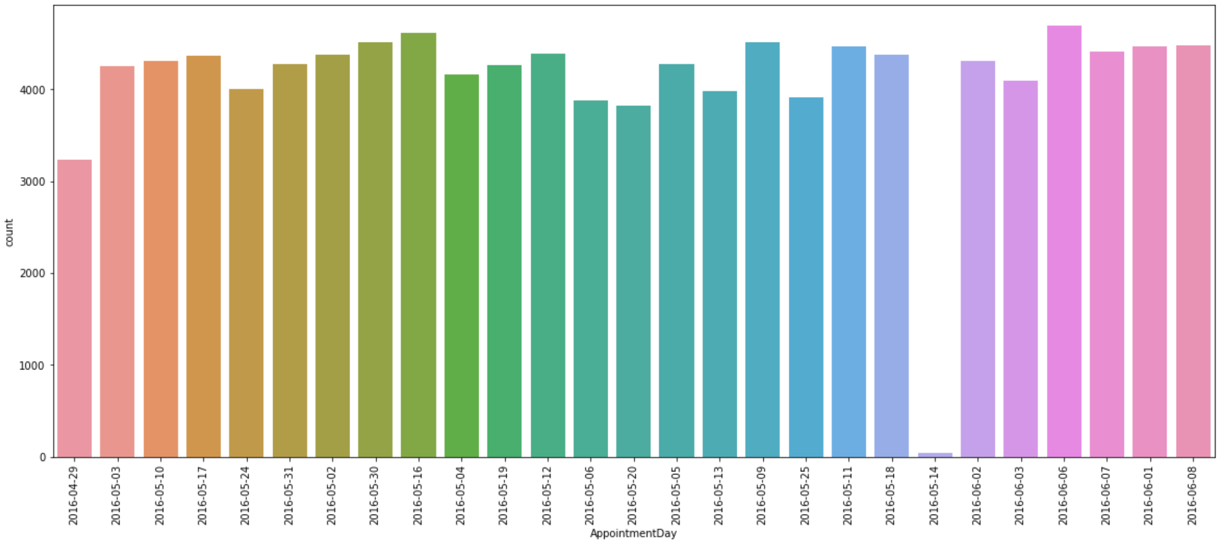


```
F    71834
M    38685
Name: Gender, dtype: int64
 ------------------------------
```

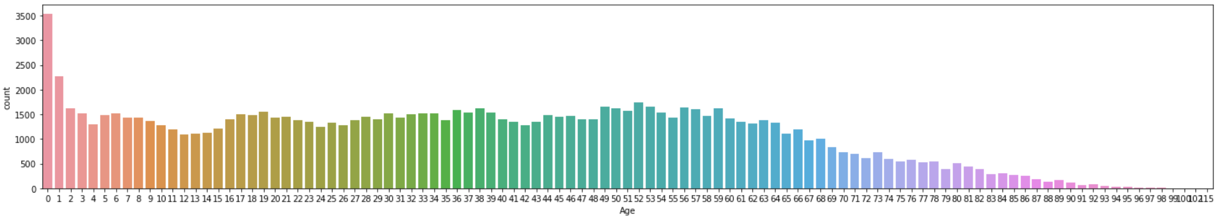


```
2016-05-03    4238
2016-05-02    4216
2016-05-16    4120
2016-05-05    4094
2016-05-10    4023
              ...
2016-01-27       1
```

```
2016-04-09       1
2016-03-05       1
2015-12-03       1
2016-01-04       1
Name: ScheduledDay, Length: 111, dtype: int64
 ------------------------------
```

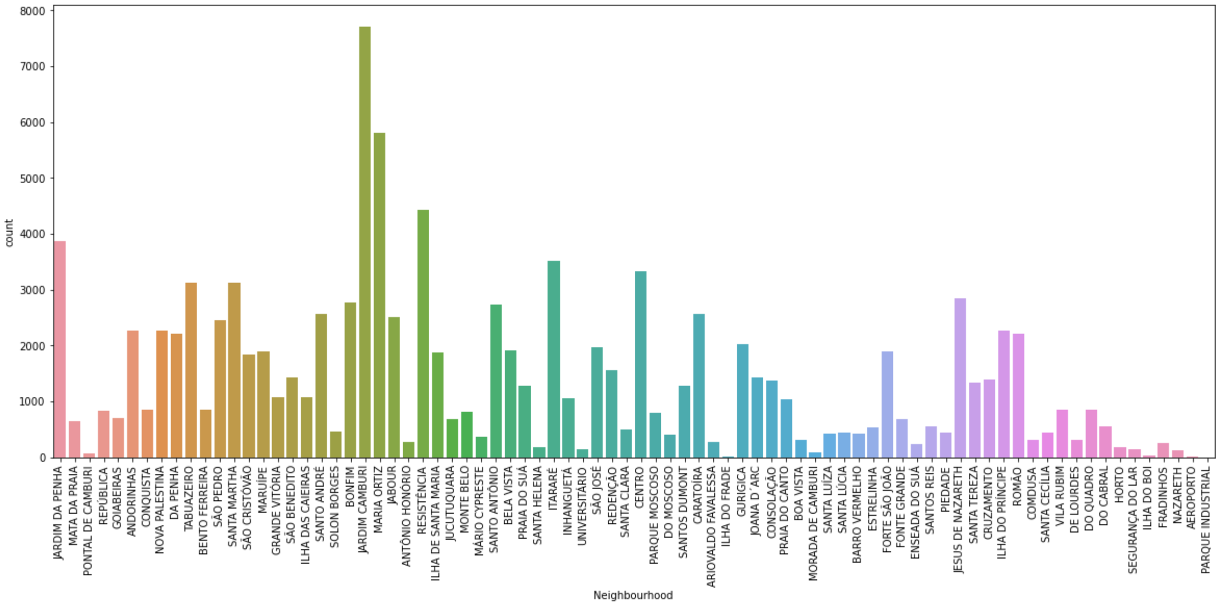


```
2016-06-06    4691
2016-05-16    4613
2016-05-09    4519
2016-05-30    4514
2016-06-08    4479
2016-05-11    4474
2016-06-01    4464
2016-06-07    4416
2016-05-12    4394
2016-05-02    4376
2016-05-18    4373
2016-05-17    4371
2016-06-02    4310
2016-05-10    4308
2016-05-31    4279
2016-05-05    4272
2016-05-19    4270
2016-05-03    4255
2016-05-04    4167
2016-06-03    4090
2016-05-24    4009
2016-05-13    3985
2016-05-25    3909
2016-05-06    3879
2016-05-20    3828
2016-04-29    3235
2016-05-14      39
Name: AppointmentDay, dtype: int64
 ------------------------------
```

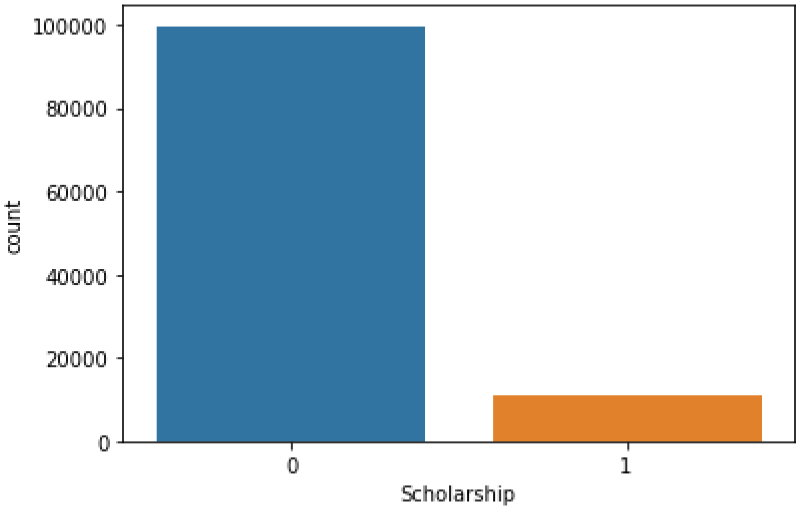


```
0         3539
1         2273
52        1746
49        1652
53        1651
          ...
98           6
115          5
```

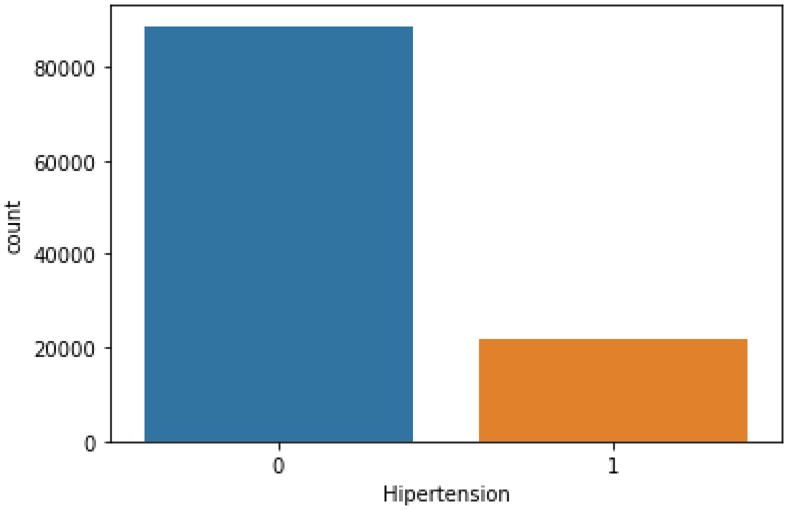
```
100       4
102       2
99        1
Name: Age, Length: 103, dtype: int64
 ------------------------------
```



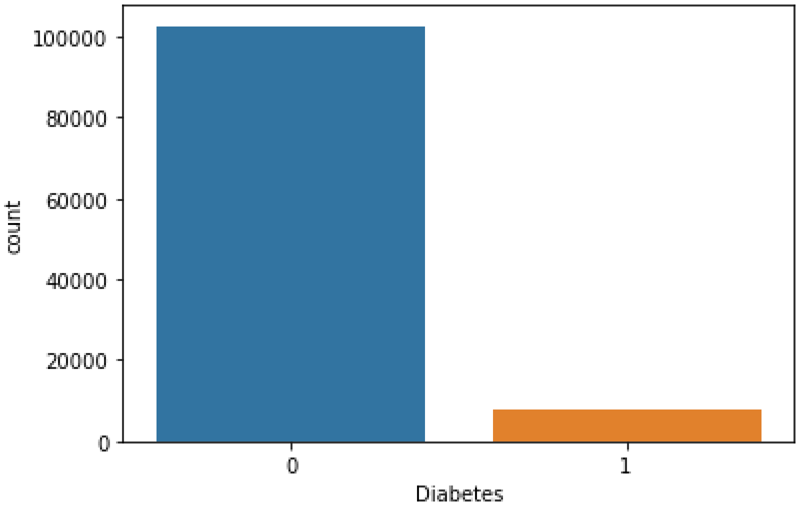
```
JARDIM CAMBURI       7717
MARIA ORTIZ          5805
RESISTÊNCIA          4430
JARDIM DA PENHA      3877
ITARARÉ              3514
                     ...
PONTAL DE CAMBURI      69
ILHA DO BOI            35
ILHA DO FRADE          10
AEROPORTO               8
PARQUE INDUSTRIAL       1
Name: Neighbourhood, Length: 80, dtype: int64
 ------------------------------
```



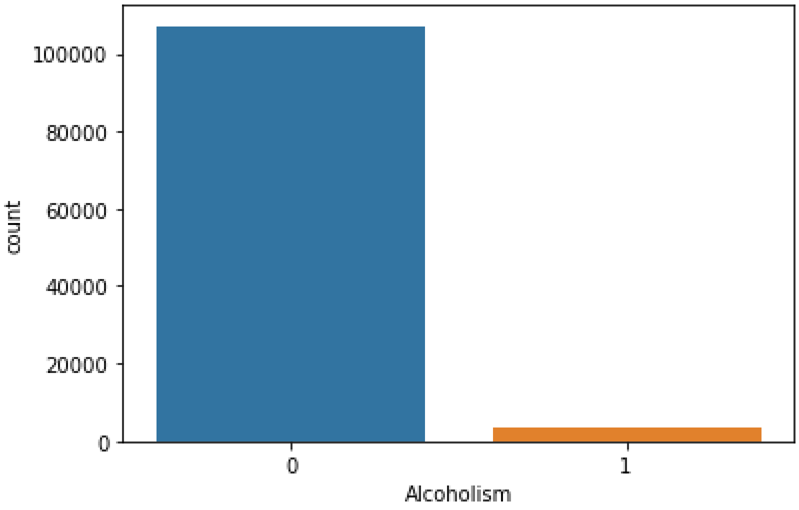
```
0    99658
1    10861
Name: Scholarship, dtype: int64
 ------------------------------
```



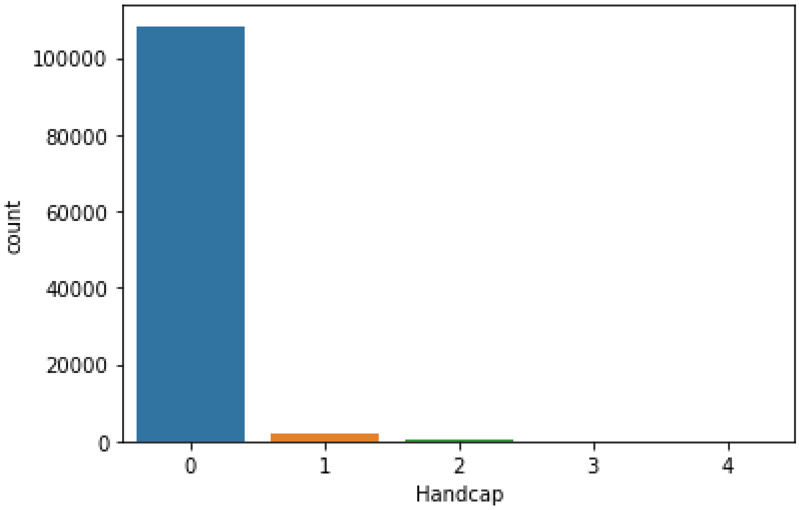
```
0    88718
1    21801
Name: Hipertension, dtype: int64
 ------------------------------
```



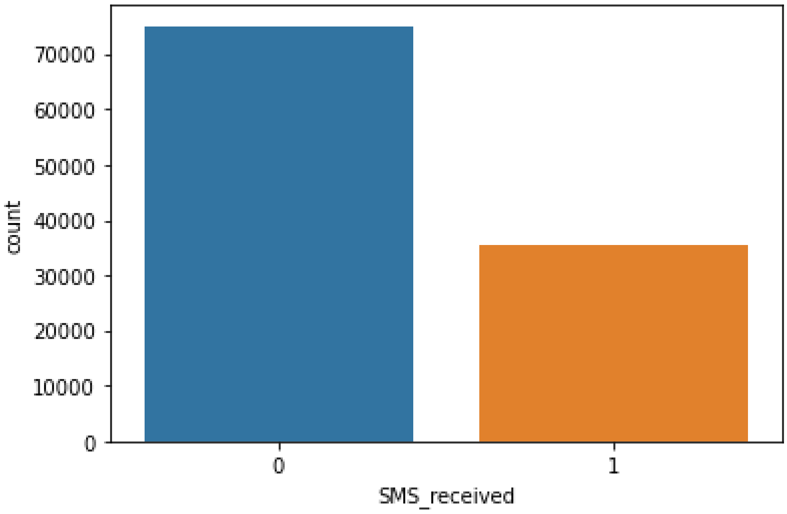
```
0    102576
1      7943
Name: Diabetes, dtype: int64
 ------------------------------
```



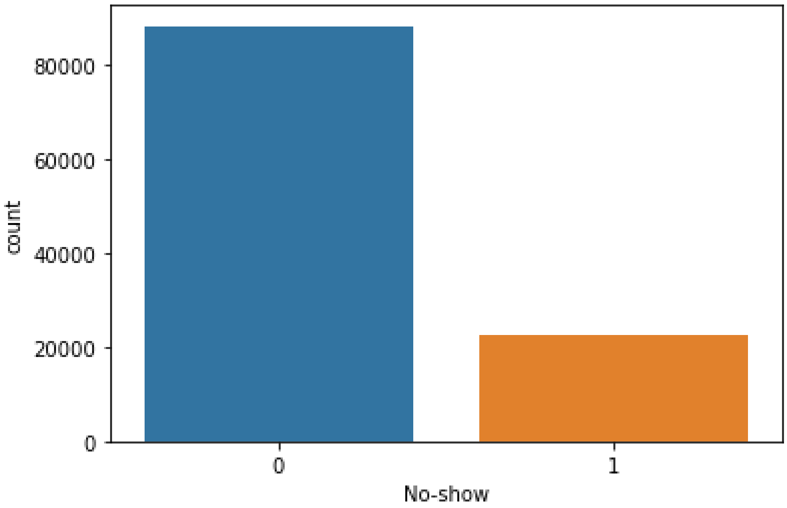
```
0    107159
1      3360
Name: Alcoholism, dtype: int64
 ------------------------------
```



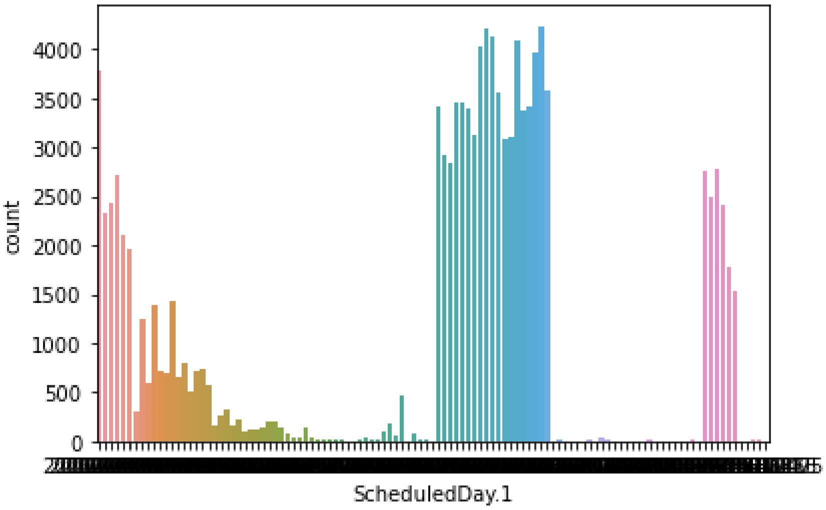
```
0    108280
1      2040
2       183
3        13
4         3
Name: Handcap, dtype: int64
 ------------------------------
```



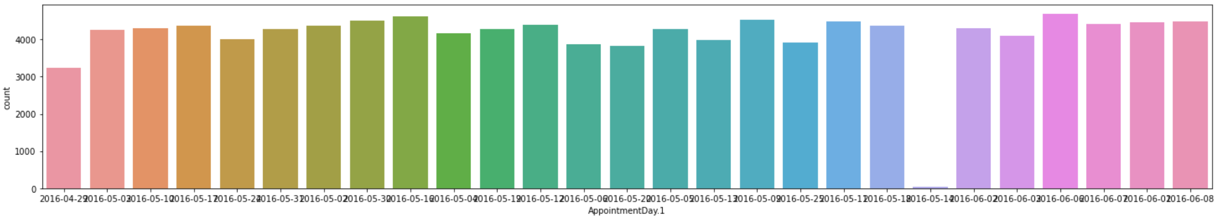
```
0    75037
1    35482
Name: SMS_received, dtype: int64
 ------------------------------
```



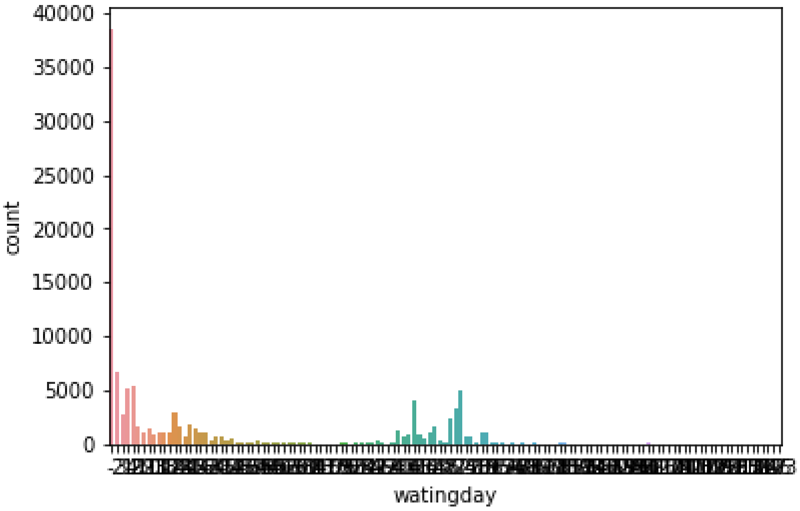
```
0    88207
1    22312
Name: No-show, dtype: int64
 ------------------------------
```



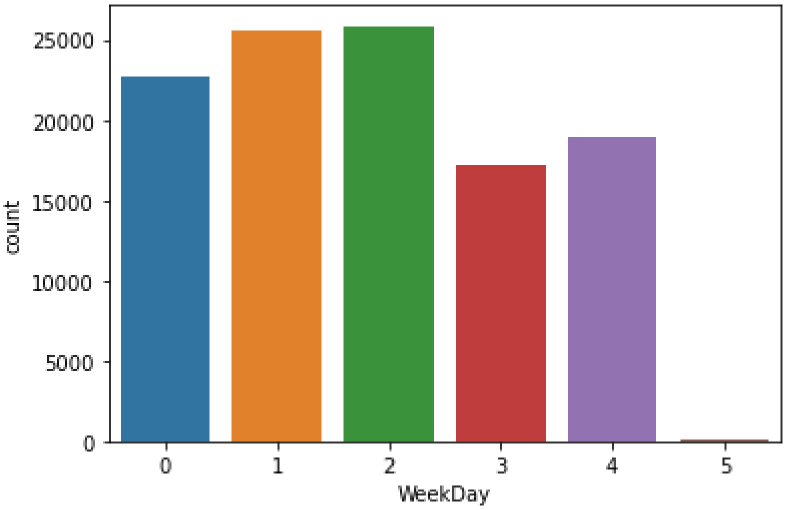
```
2016-05-03    4238
2016-05-02    4216
2016-05-16    4120
2016-05-05    4094
2016-05-10    4023
              ...
2016-01-27       1
2016-04-09       1
2016-03-05       1
2015-12-03       1
2016-01-04       1
Name: ScheduledDay.1, Length: 111, dtype: int64
 ------------------------------
```



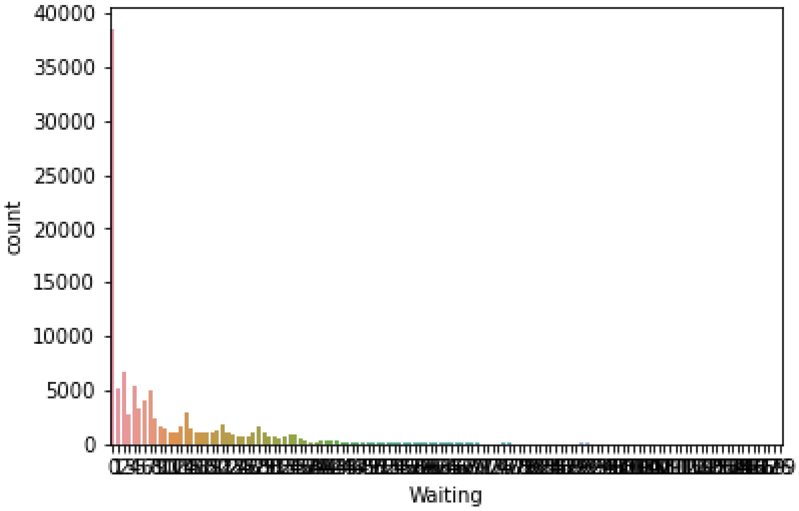
```
2016-06-06    4691
2016-05-16    4613
2016-05-09    4519
2016-05-30    4514
2016-06-08    4479
2016-05-11    4474
2016-06-01    4464
2016-06-07    4416
2016-05-12    4394
2016-05-02    4376
2016-05-18    4373
2016-05-17    4371
2016-06-02    4310
2016-05-10    4308
2016-05-31    4279
2016-05-05    4272
2016-05-19    4270
2016-05-03    4255
2016-05-04    4167
2016-06-03    4090
2016-05-24    4009
2016-05-13    3985
2016-05-25    3909
2016-05-06    3879
2016-05-20    3828
2016-04-29    3235
2016-05-14      39
Name: AppointmentDay.1, dtype: int64
 ------------------------------
```



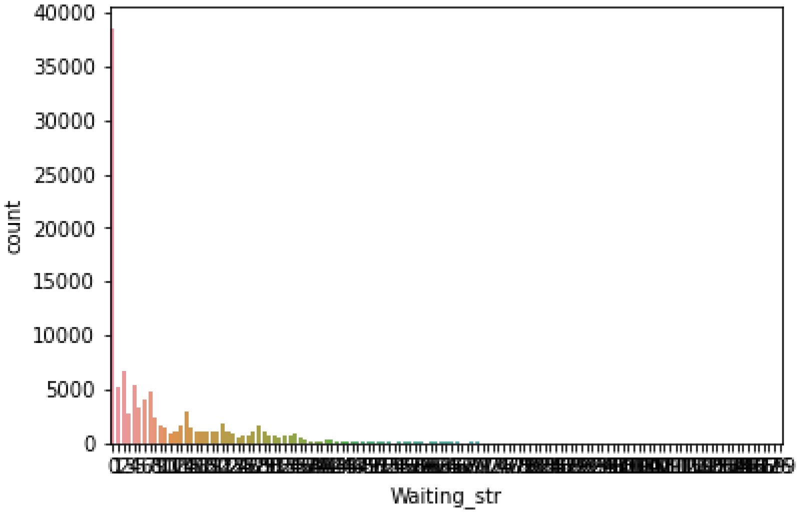
```
-         38562
2          6725
4          5290
1          5213
7          4906
          ...
82            1
146           1
132           1
123           1
127           1
Name: watingday, Length: 129, dtype: int64
 ------------------------------
```



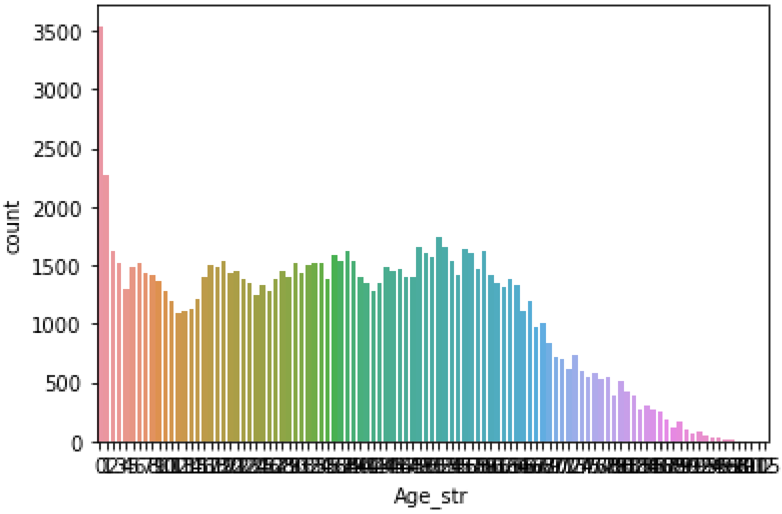
```
2    25866
1    25638
0    22713
4    19017
3    17246
5       39
Name: WeekDay, dtype: int64
 ------------------------------
```



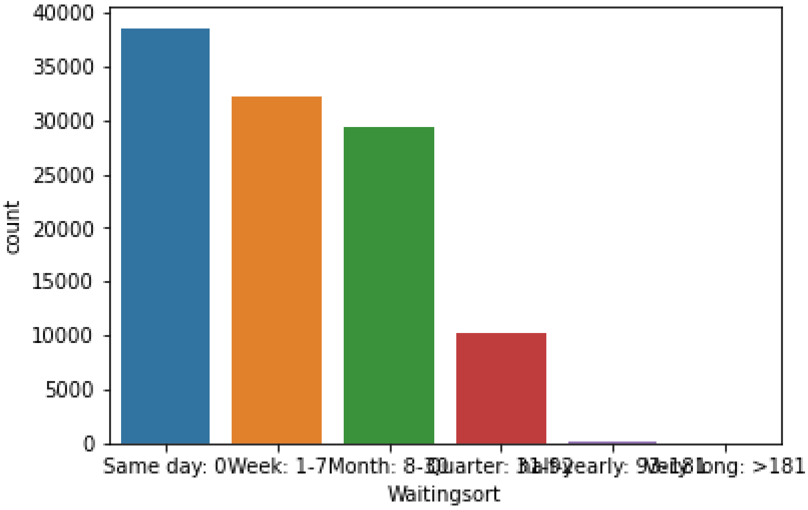
```
0      38562
2       6725
4       5290
1       5213
7       4906
        ...
101        1
132        1
151        1
146        1
127        1
Name: Waiting, Length: 129, dtype: int64
 ------------------------------
```



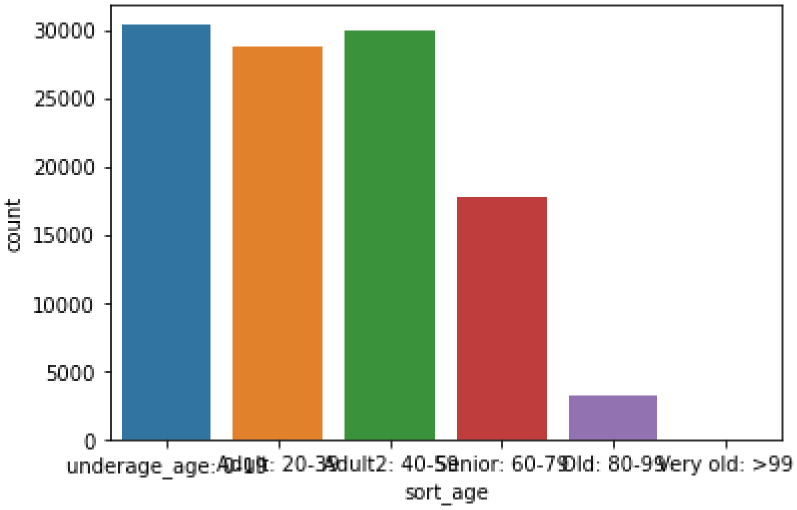
```
0      38562
2       6725
4       5290
1       5213
7       4906
        ...
101        1
132        1
151        1
146        1
127        1
Name: Waiting_str, Length: 129, dtype: int64
 ------------------------------
```



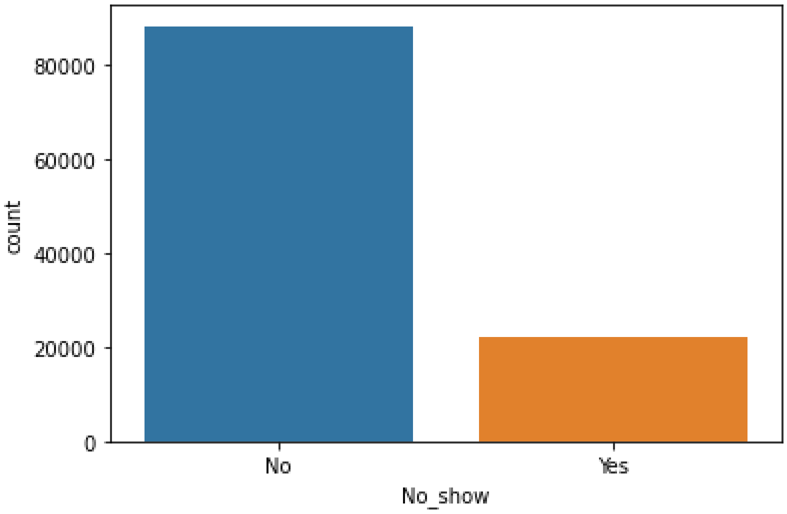
```
0      3539
1      2273
52     1746
49     1652
53     1651
       ...
98        6
115       5
100       4
102       2
99        1
Name: Age_str, Length: 103, dtype: int64
 ------------------------------
```



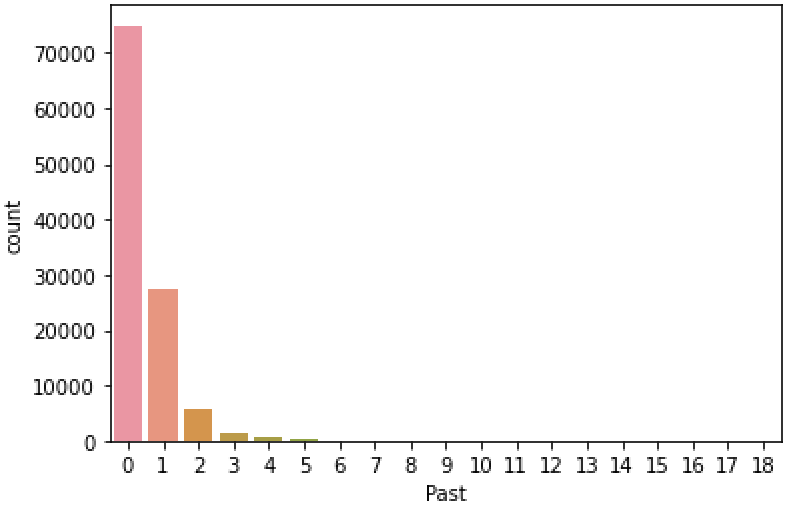
```
Same day: 0            38562
Week: 1-7              32185
Month: 8-30            29394
Quarter: 31-92         10220
half-yearly: 93-181      158
Very long: >181            0
Name: Waitingsort, dtype: int64
 ------------------------------
```



```
underage_age: 0-19    30409
Adult2: 40-59         30070
Adult: 20-39          28868
Senior: 60-79         17810
Old: 80-99             3351
Very old: >99            11
Name: sort_age, dtype: int64
 ------------------------------
```



```
No     88207
Yes    22312
Name: No_show, dtype: int64
 ------------------------------
```



```
0      74925
1      27550
2       5552
3       1508
```

```
4        485
5        220
6        105
7         51
8         35
9         28
10        19
11        19
12         6
13         5
14         4
15         3
16         2
17         1
18         1
Name: Past, dtype: int64
 ------------------------------
```

In [35]:

```
df.columns
```

Out[35]:

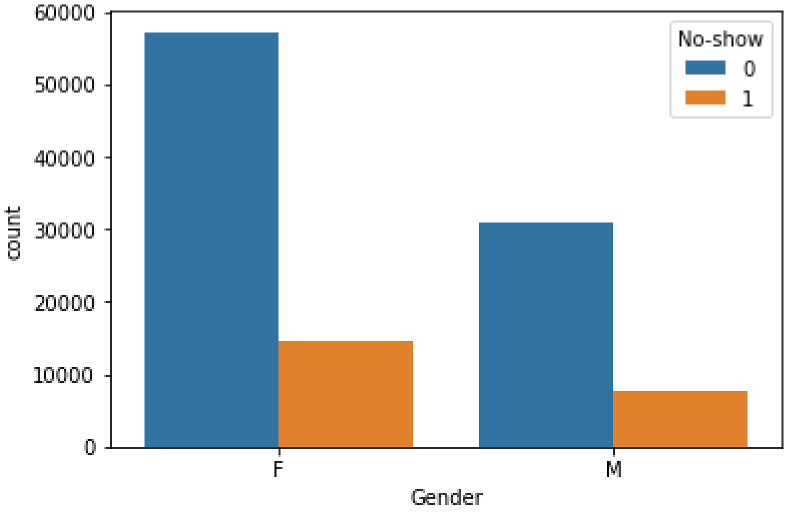
```
Index(['Gender', 'ScheduledDay', 'AppointmentDay', 'Age', 'Neighbourhood',
       'Scholarship', 'Hipertension', 'Diabetes', 'Alcoholism', 'Handcap',
       'SMS_received', 'No-show', 'ScheduledDay.1', 'AppointmentDay.1',
       'watingday', 'WeekDay', 'Waiting', 'Waiting_str', 'Age_str',
       'Waitingsort', 'sort_age', 'No_show', 'Past'],
      dtype='object')
```

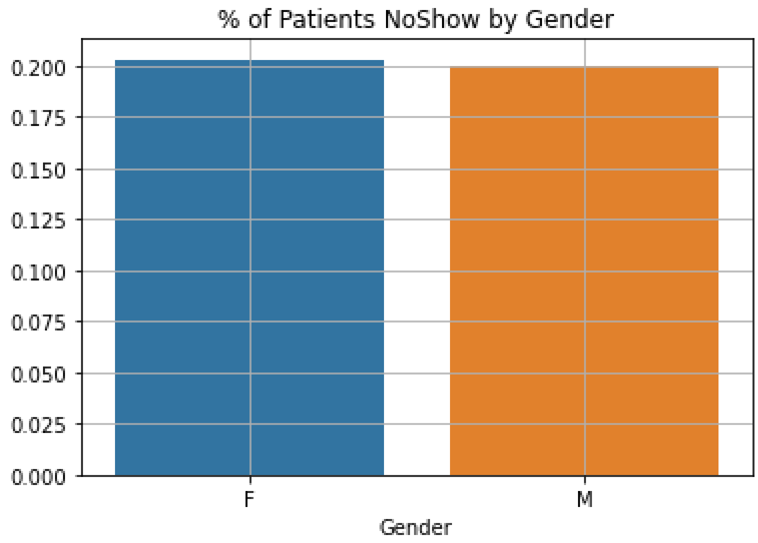
In [36]:

```
def ratio(col):
    ratio_ = df[df['No-show']==1].groupby([col]).size()/df.groupby([col]).size()
    return ratio_
```

In [37]:

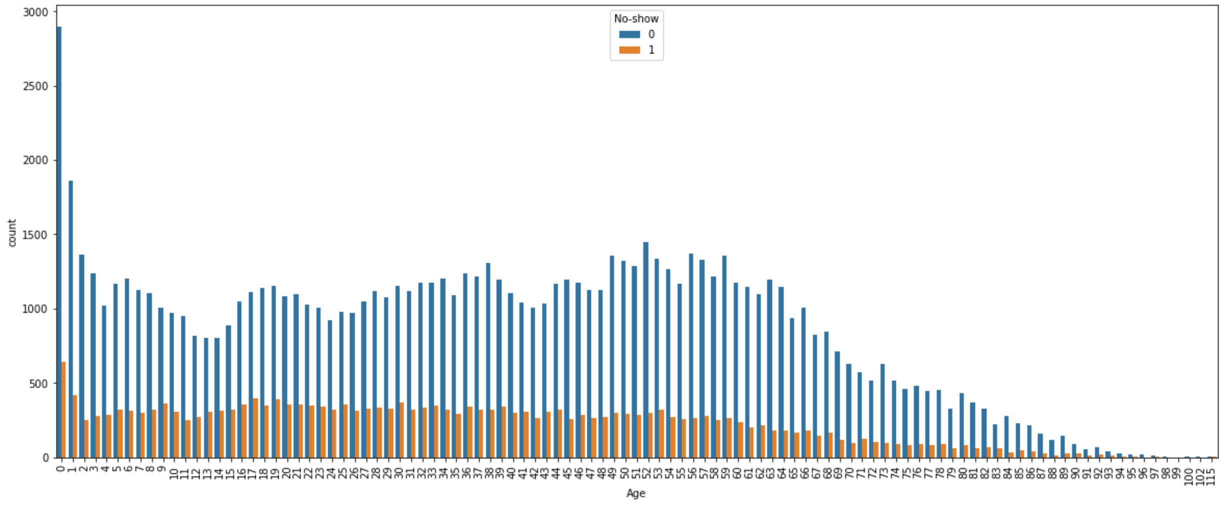
```
sns.countplot(data=df, x='Gender', hue='No-show')
plt.show()

sns.barplot(x=ratio('Gender').index, y=ratio('Gender'))
plt.title('% of Patients NoShow by Gender')
plt.grid()
plt.show()
```

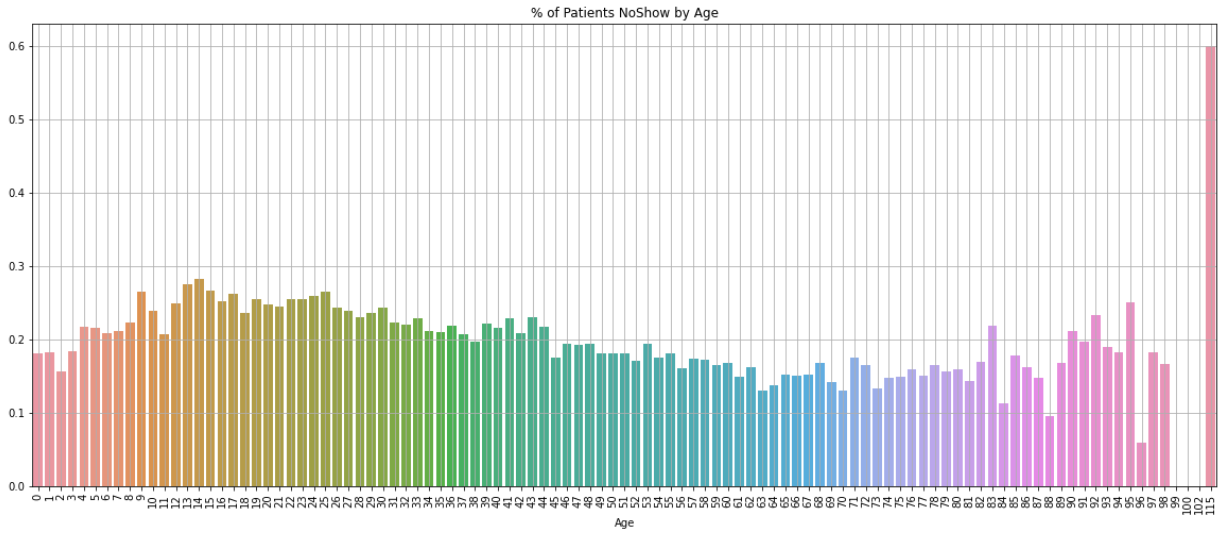




In [38]:

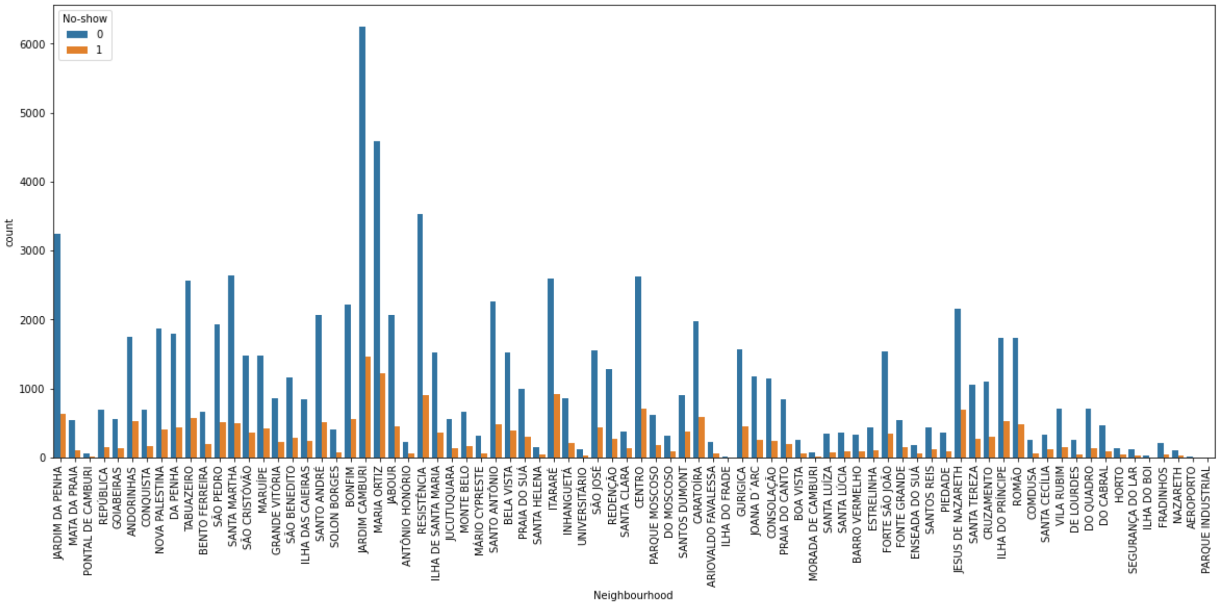
```
plt.figure(figsize=(20,8))
sns.countplot(data=df, x='Age', hue='No-show')
plt.xticks(rotation=90)
plt.show()

plt.figure(figsize=(20,8))
sns.barplot(x=ratio('Age').index, y=ratio('Age'))
plt.xticks(rotation=90)
plt.title('% of Patients NoShow by Age')
plt.grid()
plt.show()
```

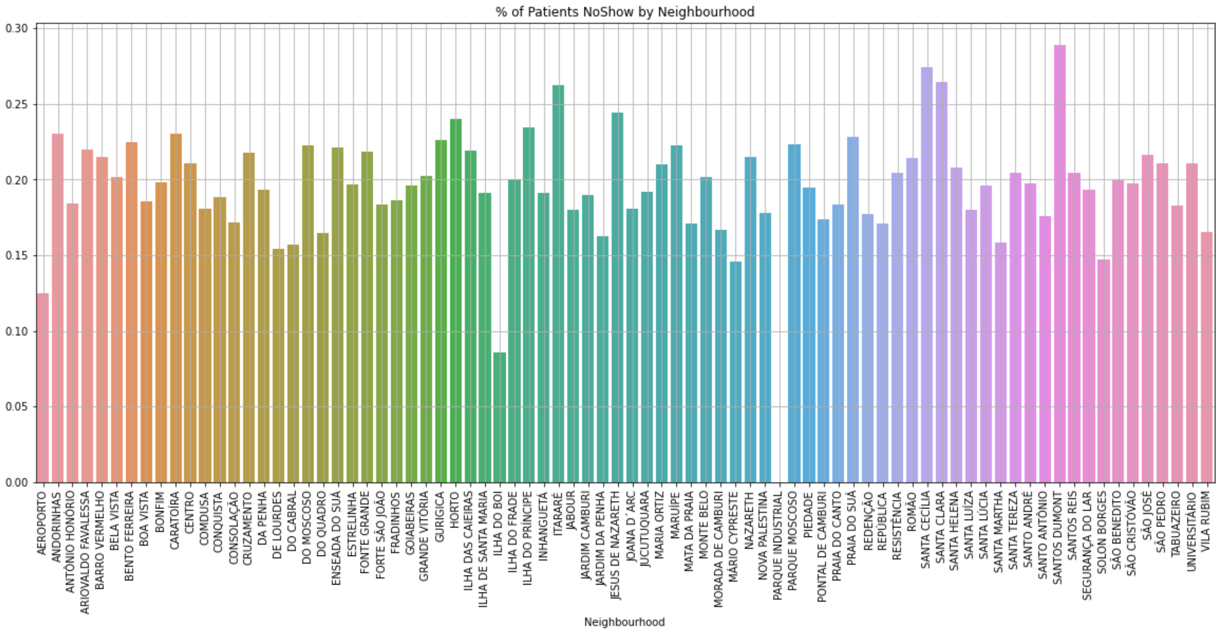




In [39]:

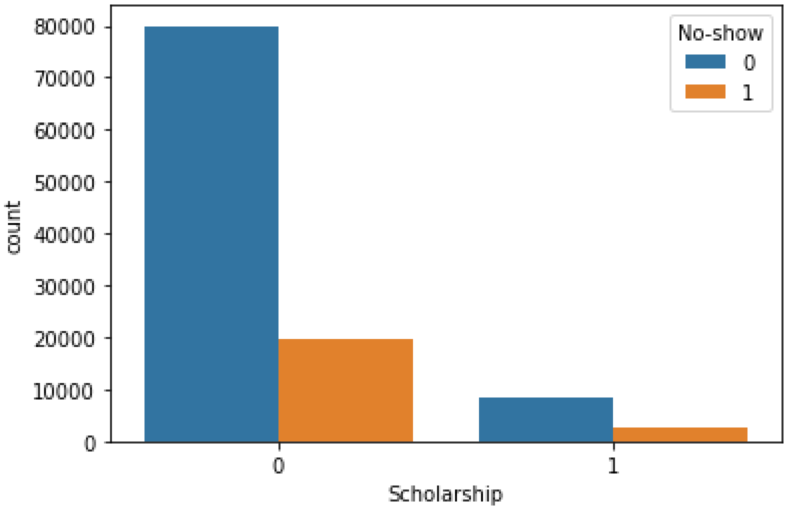
```
plt.figure(figsize=(20,8))
sns.countplot(data=df, x='Neighbourhood', hue='No-show')
plt.xticks(rotation=90)
plt.show()

plt.figure(figsize=(20,8))
sns.barplot(x=ratio('Neighbourhood').index, y=ratio('Neighbourhood'))
plt.xticks(rotation=90)
plt.title('% of Patients NoShow by Neighbourhood')
plt.grid()
plt.show()
```

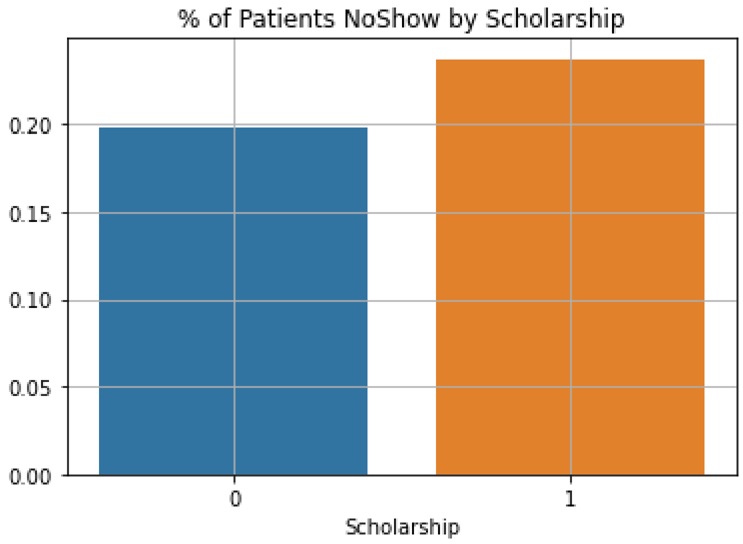




In [40]:

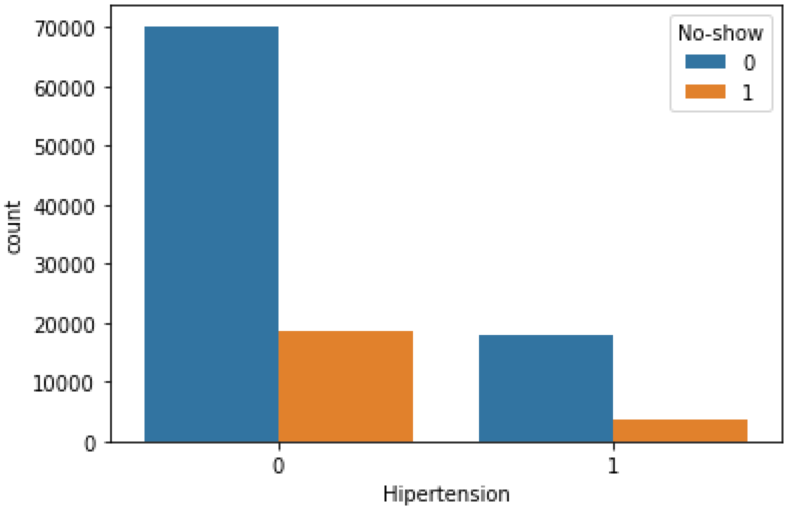
```
sns.countplot(data=df, x='Scholarship', hue='No-show')
plt.show()

sns.barplot(x=ratio('Scholarship').index, y=ratio('Scholarship'))
plt.title('% of Patients NoShow by Scholarship')
plt.grid()
plt.show()
```

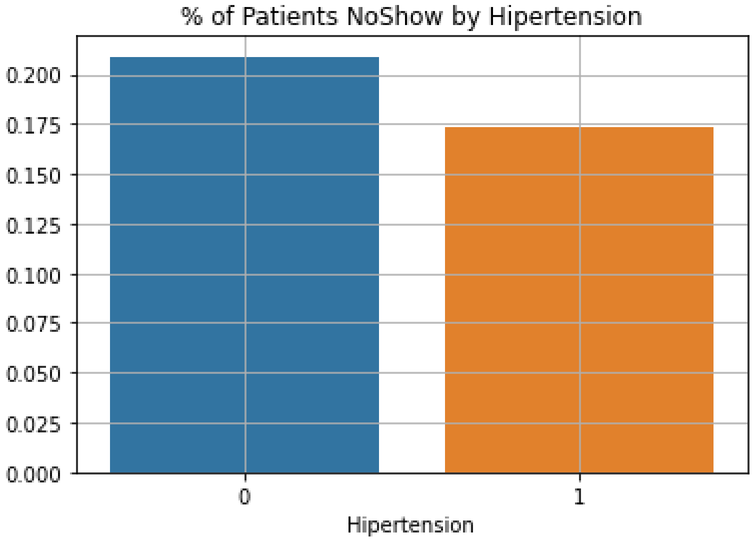




In [41]:

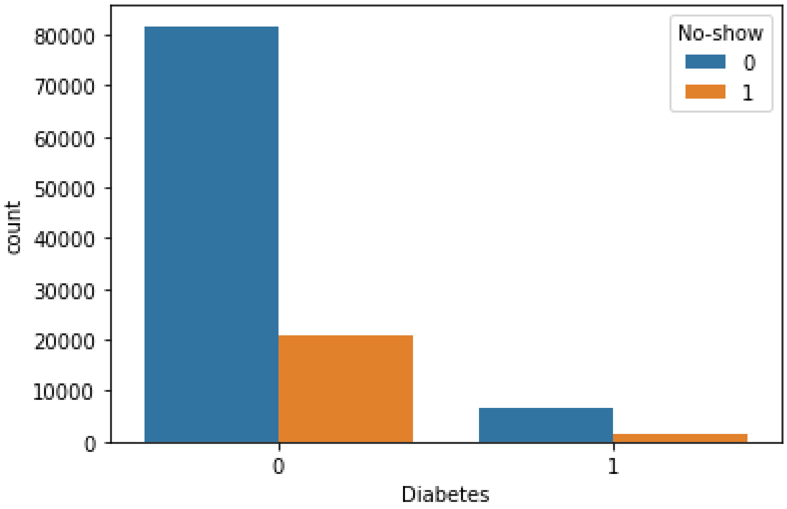
```
sns.countplot(data=df, x='Hipertension', hue='No-show')
plt.show()

sns.barplot(x=ratio('Hipertension').index, y=ratio('Hipertension'))
plt.title('% of Patients NoShow by Hipertension')
plt.grid()
plt.show()
```

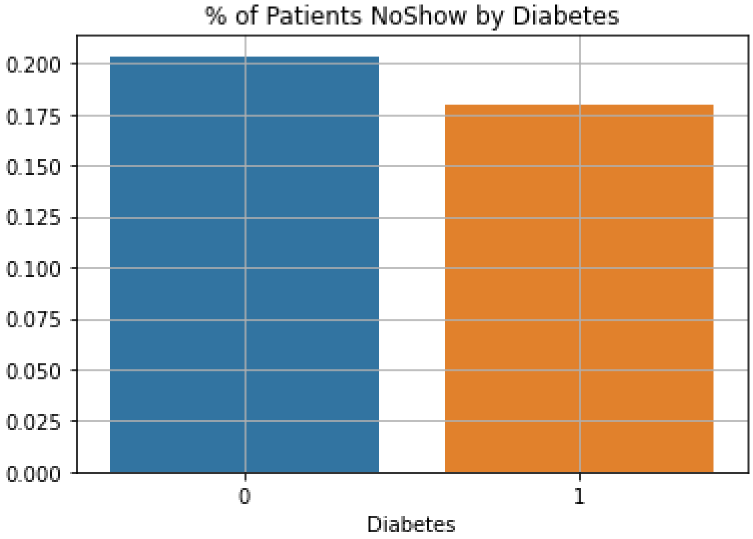




In [42]:

```
sns.countplot(data=df, x='Diabetes', hue='No-show')
plt.show()

sns.barplot(x=ratio('Diabetes').index, y=ratio('Diabetes'))
plt.title('% of Patients NoShow by Diabetes')
plt.grid()
plt.show()
```

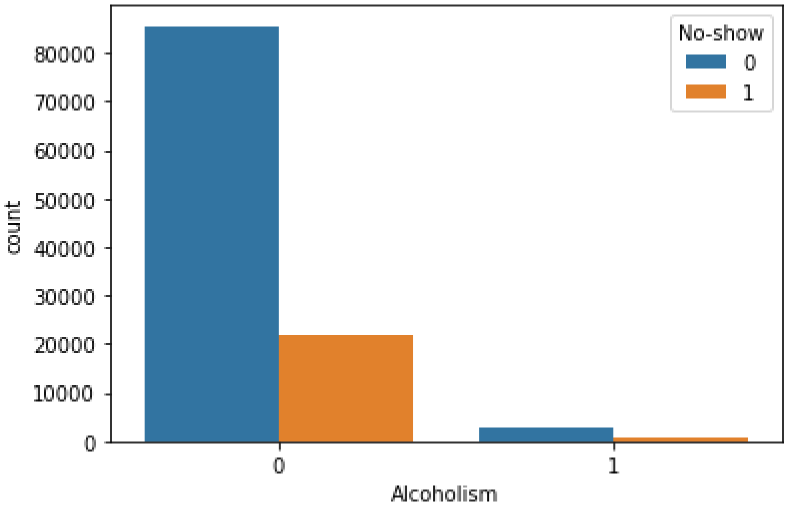


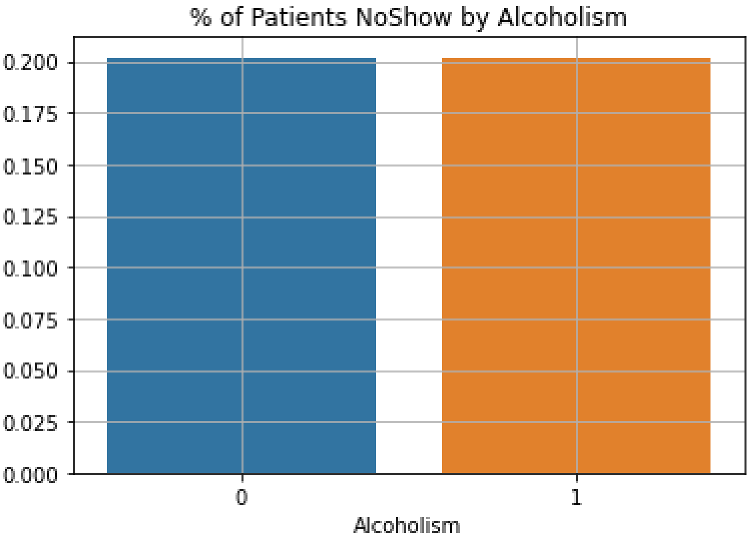


In [43]:

```
sns.countplot(data=df, x='Alcoholism', hue='No-show')
plt.show()
```

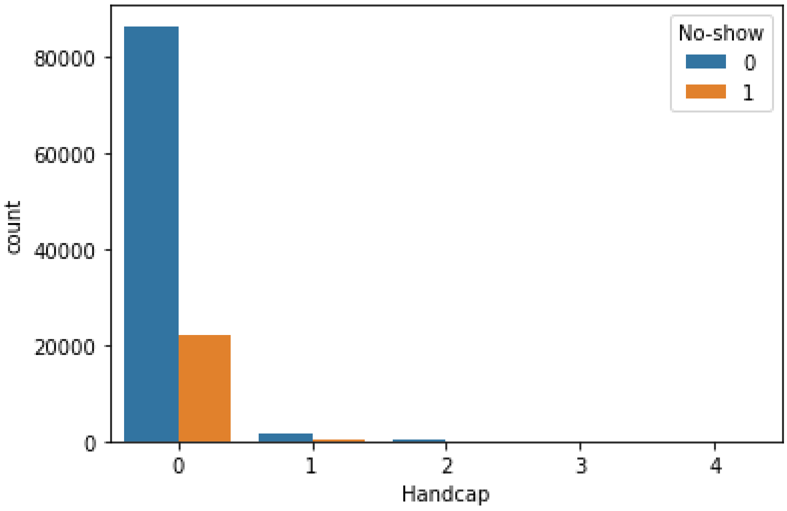
```
sns.barplot(x=ratio('Alcoholism').index, y=ratio('Alcoholism'))
plt.title('% of Patients NoShow by Alcoholism')
plt.grid()
plt.show()
```

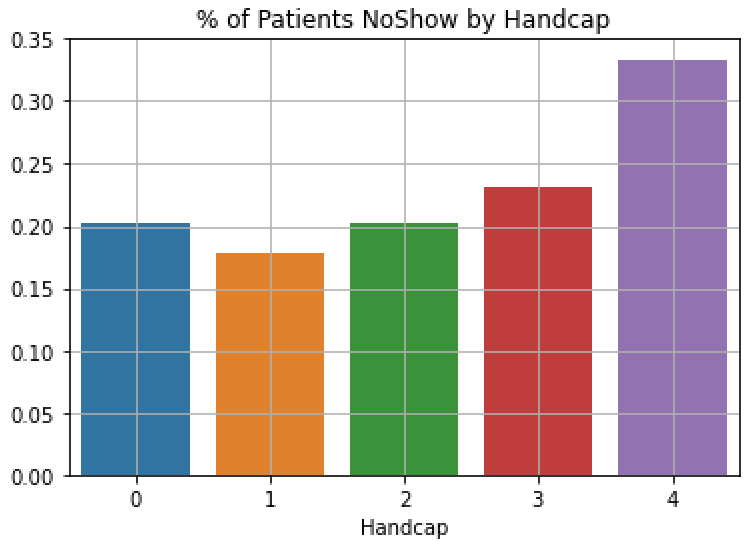




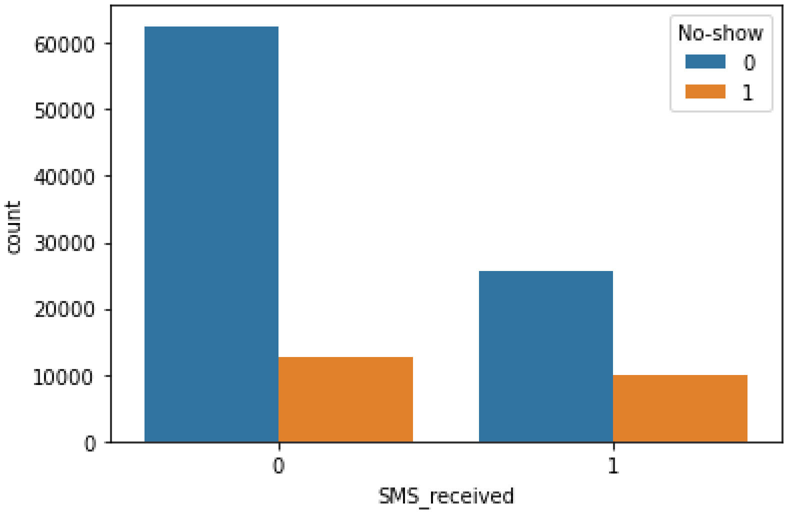
In [44]:

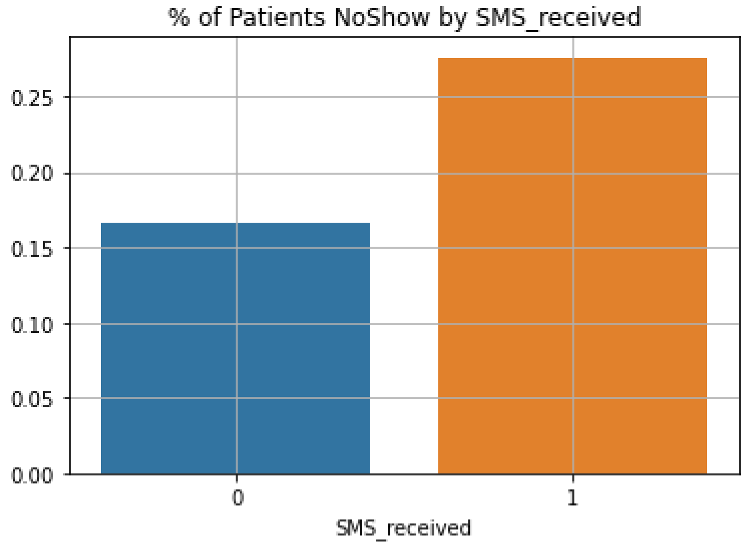
```
sns.countplot(data=df, x='Handcap', hue='No-show')
plt.show()

sns.barplot(x=ratio('Handcap').index, y=ratio('Handcap'))
plt.title('% of Patients NoShow by Handcap')
plt.grid()
plt.show()
```





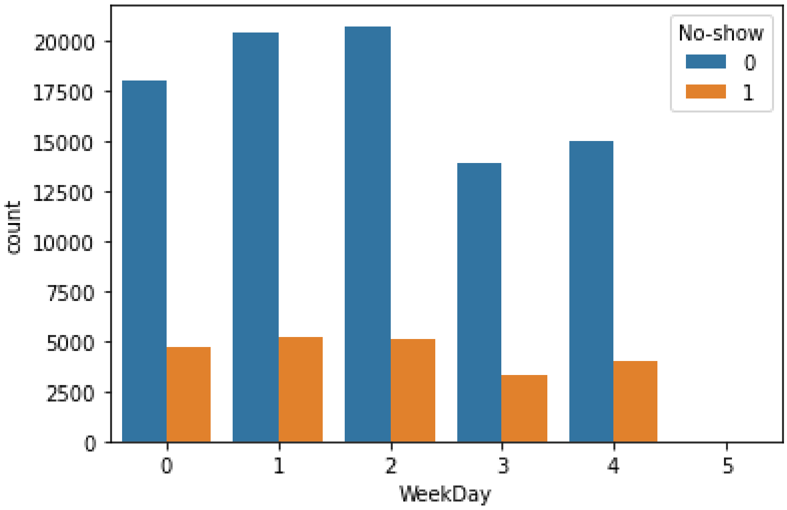
In [45]:
```
sns.countplot(data=df, x='SMS_received', hue='No-show')
plt.show()

sns.barplot(x=ratio('SMS_received').index, y=ratio('SMS_received'))
plt.title('% of Patients NoShow by SMS_received')
plt.grid()
plt.show()
```

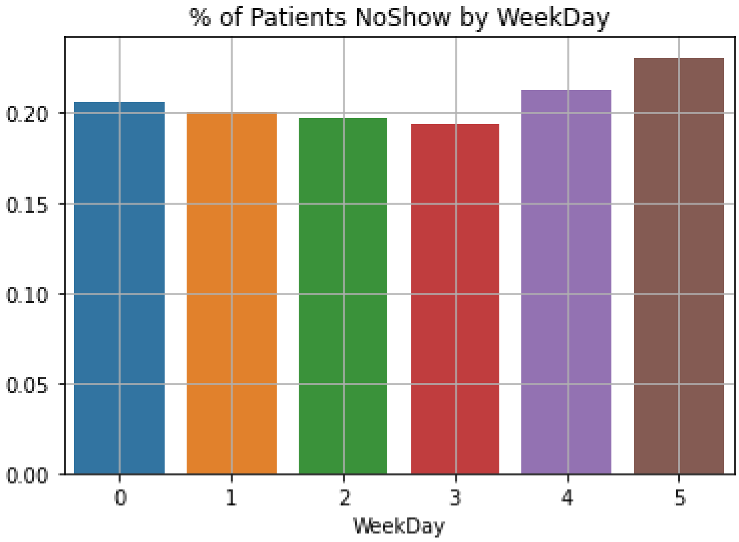




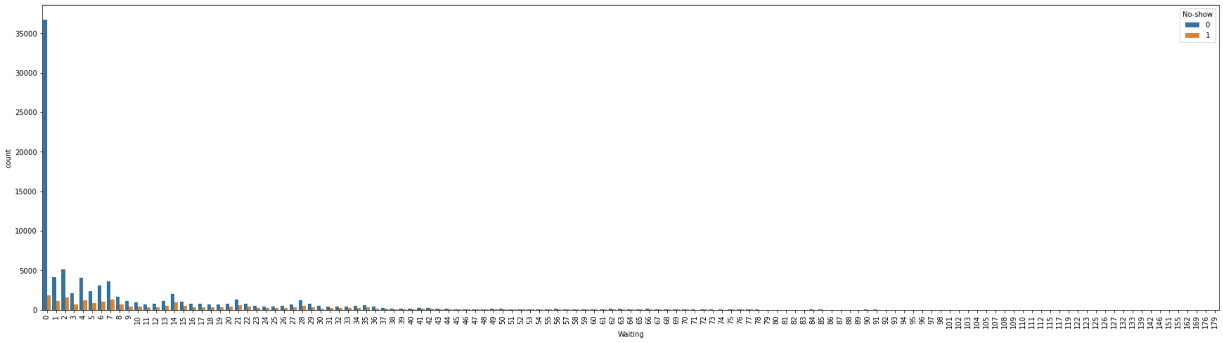
In [46]:
```
sns.countplot(data=df, x='WeekDay', hue='No-show')
plt.show()
```

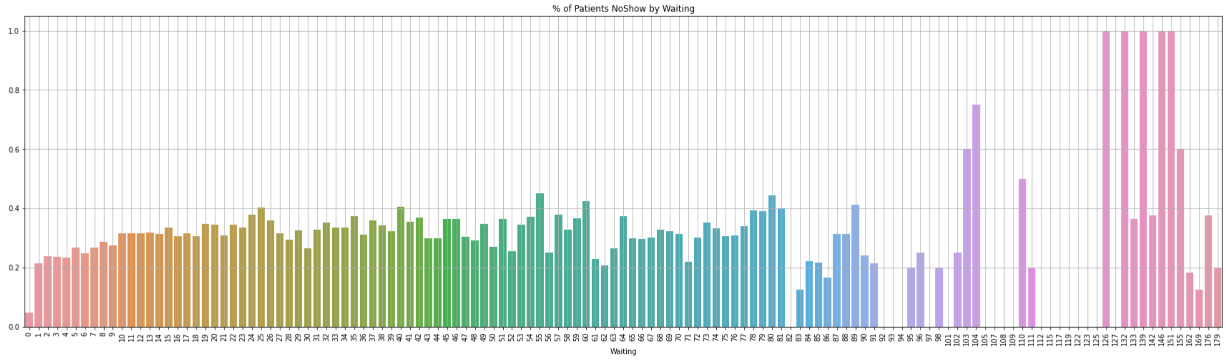
```
sns.barplot(x=ratio('WeekDay').index, y=ratio('WeekDay'))
plt.title('% of Patients NoShow by WeekDay')
plt.grid()
plt.show()
```





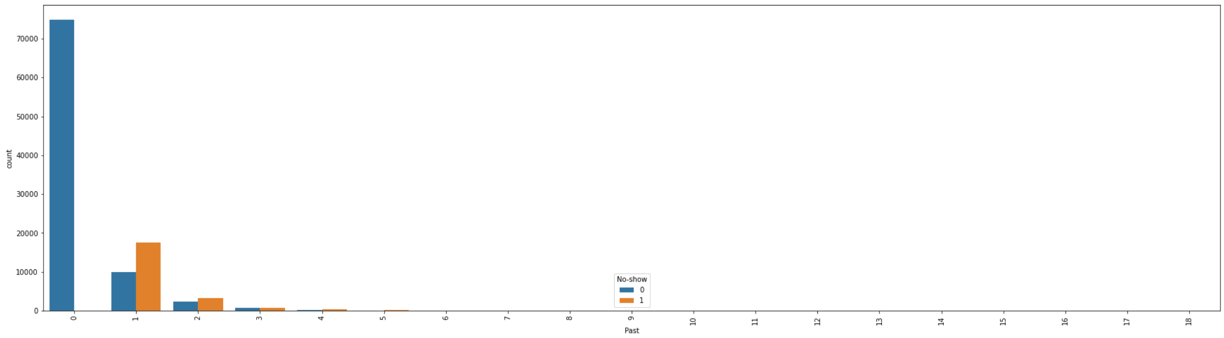
In [47]:
```
plt.figure(figsize=(30,8))
sns.countplot(data=df, x='Waiting', hue='No-show')
plt.xticks(rotation=90)
plt.show()

plt.figure(figsize=(30,8))
sns.barplot(x=ratio('Waiting').index, y=ratio('Waiting'))
plt.xticks(rotation=90)
plt.title('% of Patients NoShow by Waiting')
plt.grid()
plt.show()
```

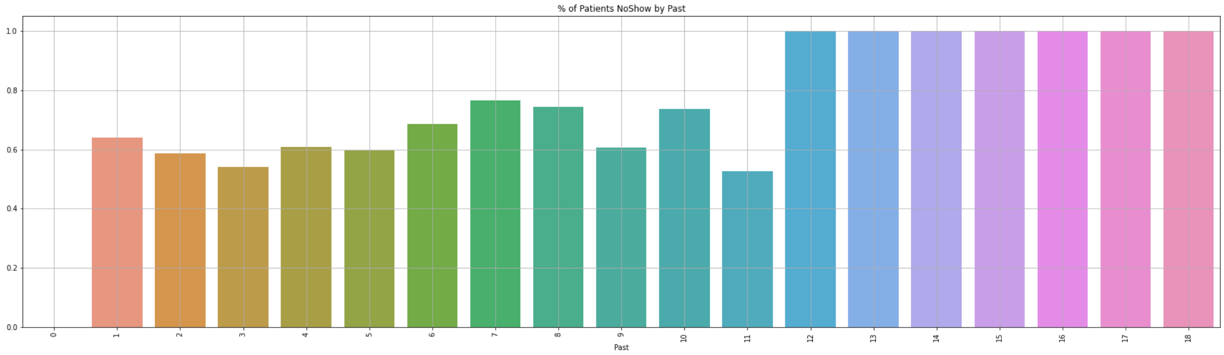




In [48]:

```
plt.figure(figsize=(30,8))
sns.countplot(data=df, x='Past', hue='No-show')
plt.xticks(rotation=90)
plt.show()

plt.figure(figsize=(30,8))
sns.barplot(x=ratio('Past').index, y=ratio('Past'))
plt.xticks(rotation=90)
plt.title('% of Patients NoShow by Past')
plt.grid()
plt.show()
```





In [49]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 110519 entries, 0 to 110526
Data columns (total 23 columns):
 #   Column           Non-Null Count   Dtype 
---  ------           --------------   ----- 
 0   Gender           110519 non-null  object
 1   ScheduledDay     110519 non-null  object
 2   AppointmentDay   110519 non-null  object
 3   Age              110519 non-null  int64 
 4   Neighbourhood    110519 non-null  object
 5   Scholarship      110519 non-null  int64 
 6   Hipertension     110519 non-null  int64 
 7   Diabetes         110519 non-null  int64 
 8   Alcoholism       110519 non-null  int64 
 9   Handcap          110519 non-null  int64 
 10  SMS_received     110519 non-null  int64 
 11  No-show          110519 non-null  int64 
 12  ScheduledDay.1    110519 non-null  object
 13  AppointmentDay.1  110519 non-null  object
 14  watingday         110519 non-null  object
 15  WeekDay           110519 non-null  int64
 16  Waiting           110519 non-null  int64
 17  Waiting_str       110519 non-null  int64
 18  Age_str           110519 non-null  int64
 19  Waitingsort       110519 non-null  category
 20  sort_age          110519 non-null  category
 21  No_show           110519 non-null  object
 22  Past              110519 non-null  int64
dtypes: category(2), int64(13), object(8)
memory usage: 23.8+ MB
```

In [50]:

```
df.describe()
```

Out[50]:

	Age	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap
count	110519.000000	110519.000000	110519.000000	110519.000000	110519.000000	110519.000000
mean	37.089071	0.098273	0.197260	0.071870	0.030402	0.022231
std	23.109970	0.297684	0.397932	0.258274	0.171692	0.161495
min	0.000000	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	0.000000
max	115.000000	1.000000	1.000000	1.000000	1.000000	4.000000

In [51]:

```
import numpy as np
import pandas as pd

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

import tensorflow as tf

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,

tf.random.set_seed(500)

df =pd.read_csv('./KaggleV2-May-2016.csv',encoding='latin-1')
```

In [52]:

```
#
df['No-show'].replace("Yes", 1, inplace=True)
df['No-show'].replace("No", 0, inplace=True)

df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay']).dt.date.astype('datetime64[ns
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay']).dt.date.astype('datetime6
df['WeekDay']=df['AppointmentDay'].dt.weekday#요일 숫자로
df['Waiting']=(df['AppointmentDay']-df['ScheduledDay']).dt.days
df['Past'] = df.sort_values(['ScheduledDay']).groupby(['PatientId'])['No-show'].cumsu

#
df = df[df['Age']>=0]
df = df[df['Waiting']>=0]
df['NoShow'] = df['No-show']
df.drop(['PatientId','AppointmentID','No-show'], axis=1, inplace=True)
df.drop(['ScheduledDay'], axis=1, inplace=True)
df.drop(['AppointmentDay'], axis=1, inplace=True)
df.drop(df[df['Neighbourhood'] == 'ILHAS OCEÂNICAS DE TRINDADE'].index, inplace=True

df.Gender = df.Gender.apply(lambda x: 1 if x == 'M' else 0)
```

In [53]:

```
def encoding(df, column, prefix):
    df = df.copy()
    dumy1 = pd.get_dummies(df[column], prefix=prefix)
    df = pd.concat([df, dumy1], axis=1)
    df = df.drop(column, axis=1)
    return df
#https://devuna.tistory.com/67,https://rfriend.tistory.com/tag/pd.get_dummies%28%29%2(
```

In [54]:

```
df = encoding(df, 'Neighbourhood', prefix='N')
```

In [55]:

```
y = df['NoShow'].copy()
X = df.drop('NoShow', axis=1).copy()
scaler = StandardScaler()

X = scaler.fit_transform(X)
```

In [56]:

```
Xtrain, xtest, ytrain, ytest = train_test_split(X, y, train_size=0.8, random_state=10
```

In [57]:

```
inputs = tf.keras.Input(shape=(X.shape[1]))
x = tf.keras.layers.Dense(64, activation='relu')(inputs)
x = tf.keras.layers.Dense(64, activation='relu')(x)
outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)

model = tf.keras.Model(inputs, outputs)
model.compile(
    loss='binary_crossentropy',
    optimizer='adam',
    metrics=[
        'accuracy',
        tf.keras.metrics.AUC(name='auc')
    ]
)

history = model.fit(
    Xtrain,
    ytrain,
    validation_split=0.2,
    epochs=500,
    batch_size=52,
    callbacks=[
        tf.keras.callbacks.EarlyStopping(
            monitor='val_loss',
            patience=4,
            restore_best_weights=True
        )
    ]
)
#참조 https://hwiyong.tistory.com/96 ,https://www.tensorflow.org/guide/keras/sequentia
```

```
Epoch 1/500
1361/1361 [==============================] - 2s 1ms/step - loss: 0.2335 - accuracy: 0.
8837 - auc: 0.9453 - val_loss: 0.1857 - val_accuracy: 0.9038 - val_auc: 0.9616
Epoch 2/500
1361/1361 [==============================] - 2s 1ms/step - loss: 0.1789 - accuracy: 0.
9082 - auc: 0.9646 - val_loss: 0.1795 - val_accuracy: 0.9087 - val_auc: 0.9650
Epoch 3/500
1361/1361 [==============================] - 2s 1ms/step - loss: 0.1741 - accuracy: 0.
9118 - auc: 0.9667 - val_loss: 0.1770 - val_accuracy: 0.9114 - val_auc: 0.9659
Epoch 4/500
1361/1361 [==============================] - 2s 1ms/step - loss: 0.1714 - accuracy: 0.
```

```
9144 - auc: 0.9678 - val_loss: 0.1743 - val_accuracy: 0.9155 - val_auc: 0.9674
Epoch 5/500
1361/1361 [==============================] - 2s 1ms/step - loss: 0.1693 - accuracy: 0.
9160 - auc: 0.9687 - val_loss: 0.1739 - val_accuracy: 0.9137 - val_auc: 0.9670
Epoch 6/500
1361/1361 [==============================] - 2s 1ms/step - loss: 0.1679 - accuracy: 0.
9163 - auc: 0.9695 - val_loss: 0.1747 - val_accuracy: 0.9125 - val_auc: 0.9668
Epoch 7/500
1361/1361 [==============================] - 2s 1ms/step - loss: 0.1664 - accuracy: 0.
9181 - auc: 0.9700 - val_loss: 0.1722 - val_accuracy: 0.9161 - val_auc: 0.9676
Epoch 8/500
1361/1361 [==============================] - 2s 1ms/step - loss: 0.1652 - accuracy: 0.
9186 - auc: 0.9704 - val_loss: 0.1728 - val_accuracy: 0.9163 - val_auc: 0.9675
Epoch 9/500
1361/1361 [==============================] - 2s 2ms/step - loss: 0.1641 - accuracy: 0.
9198 - auc: 0.9710 - val_loss: 0.1726 - val_accuracy: 0.9151 - val_auc: 0.9673
Epoch 10/500
1361/1361 [==============================] - 2s 1ms/step - loss: 0.1628 - accuracy: 0.
9201 - auc: 0.9714 - val_loss: 0.1724 - val_accuracy: 0.9179 - val_auc: 0.9677
Epoch 11/500
1361/1361 [==============================] - 2s 1ms/step - loss: 0.1615 - accuracy: 0.
9212 - auc: 0.9720 - val_loss: 0.1733 - val_accuracy: 0.9165 - val_auc: 0.9673
```

In [58]:

```
model.evaluate(xtest, ytest)
```

```
691/691 [==============================] - 1s 726us/step - loss: 0.1758 - accuracy: 0.
9116 - auc: 0.9659
```

Out[58]:

```
[0.17580853402614594, 0.9115584492683411, 0.9658643007278442]
```

In [59]:

```
y_true = np.array(ytest)
y_pred = np.squeeze(np.array(model.predict(xtest) >= 0.5, dtype=np.int))
print("분류:₩n₩n", classification_report(y_true, y_pred))
```

```
분류:

              precision    recall  f1-score   support

           0       0.96      0.93      0.94     17683
           1       0.75      0.85      0.79      4422

    accuracy                           0.91     22105
   macro avg       0.85      0.89      0.87     22105
weighted avg       0.92      0.91      0.91     22105
```

In [60]:

```
print("Confusion Matrix:₩n", confusion_matrix(y_true, y_pred))
```

```
Confusion Matrix:
 [[16402  1281]
 [  674  3748]]
```

Analysis

DT

In [61]:

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

df = pd.read_csv('./KaggleV2-May-2016_1.csv')
df.head()
df.info()

#
```

```
df['No-show'].replace("Yes", 1, inplace=True)
df['No-show'].replace("No", 0, inplace=True)

df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay']).dt.date.astype('datetime64[ns
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay']).dt.date.astype('datetime6
df['WeekDay']=df['AppointmentDay'].dt.weekday
df['Waiting']=(df['AppointmentDay']-df['ScheduledDay']).dt.days
df['Past'] = df.sort_values(['ScheduledDay']).groupby(['PatientId'])['No-show'].cumsu

#
df = df[df['Age']>=0]
df = df[df['Waiting']>=0]

df.drop(['PatientId','AppointmentID'], axis=1, inplace=True)
df.drop(['ScheduledDay'], axis=1, inplace=True)
df.drop(['AppointmentDay'], axis=1, inplace=True)
df.drop(df[df['Neighbourhood'] == 'ILHAS OCEÂNICAS DE TRINDADE'].index, inplace=True
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 17 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 
 14  ScheduledDay.1    110527 non-null  object 
 15  AppointmentDay.1  110527 non-null  object 
 16  watingday         110527 non-null  object 
dtypes: float64(1), int64(8), object(8)
memory usage: 14.3+ MB
```

In [62]:

```
df.head(20)
```

Out[62]:

	Gender	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS
0	F	62	JARDIM DA PENHA	0	1	0	0	0	
1	M	56	JARDIM DA PENHA	0	0	0	0	0	
2	F	62	MATA DA PRAIA	0	0	0	0	0	
3	F	8	PONTAL DE CAMBURI	0	0	0	0	0	
4	F	56	JARDIM DA PENHA	0	1	1	0	0	
5	F	76	REPÚBLICA	0	1	0	0	0	
6	F	23	GOIABEIRAS	0	0	0	0	0	
7	F	39	GOIABEIRAS	0	0	0	0	0	

	Gender	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS
8	F	21	ANDORINHAS	0	0	0	0	0	
9	F	19	CONQUISTA	0	0	0	0	0	
10	F	30	NOVA PALESTINA	0	0	0	0	0	
11	M	29	NOVA PALESTINA	0	0	0	0	0	
12	F	22	NOVA PALESTINA	1	0	0	0	0	
13	M	28	NOVA PALESTINA	0	0	0	0	0	
14	F	54	NOVA PALESTINA	0	0	0	0	0	
15	F	15	NOVA PALESTINA	0	0	0	0	0	
16	M	50	NOVA PALESTINA	0	0	0	0	0	
17	F	40	CONQUISTA	1	0	0	0	0	
18	F	30	NOVA PALESTINA	1	0	0	0	0	
19	F	46	DA PENHA	0	0	0	0	0	

In [63]:

```
from sklearn import tree
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,
from sklearn.model_selection import cross_validate, learning_curve, validation_curve
from sklearn.pipeline import make_pipeline

X = df.drop(['No-show'], axis=1)
y = df['No-show']

X.keys()

X = pd.get_dummies(X[['Gender', 'Age', 'Neighbourhood','Scholarship', 'Hipertension',
                     'Alcoholism', 'Handcap', 'SMS_received', 'WeekDay', 'Waiting', '
                   columns=['Gender','Neighbourhood','WeekDay'],
                   drop_first=True)
```

In [64]:

```
Xtr, Xts, ytr, yts = train_test_split(X, y, test_size=0.25, random_state = 1, stratif

tree = DecisionTreeClassifier()
tree.fit(Xtr, ytr)
y_pred = tree.predict(Xts)
print('Training Acc: {:.3f}'.format(tree.score(Xtr, ytr)))
print('Test Acc: {:.3f}'.format(tree.score(Xts, yts)))
```

```
Training Acc: 0.995
Test Acc: 0.903
```

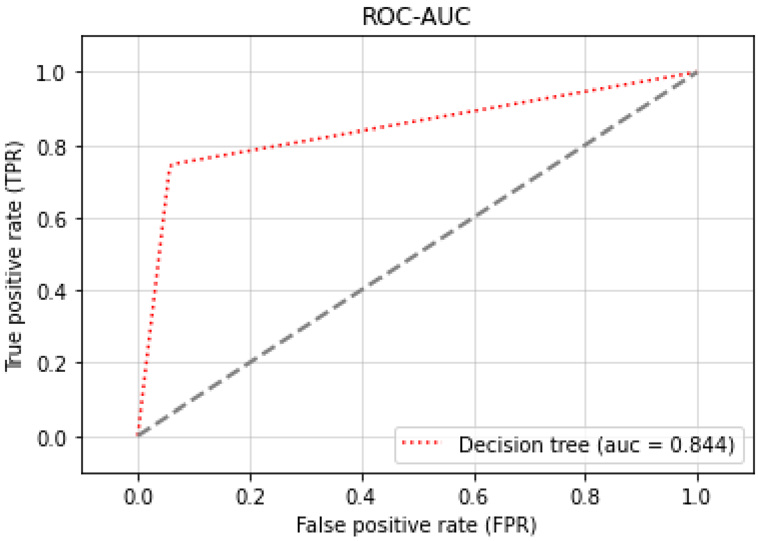
In [65]:

```
confmat = pd.DataFrame(confusion_matrix(yts,y_pred),
                       index=['True[0]','True[1]'],
                       columns=['Predict[0]','Predict[1]'])
print('confusion_matrix','₩n',confmat,'₩n₩n')

cl_report = classification_report(yts,y_pred)
print('classification_report','₩n',cl_report,'₩n₩n')

print('잘못 분류된 샘플 개수: %d' %(yts != y_pred).sum())
print('정확도: %.3f' % accuracy_score(yts,y_pred))
print('정밀도: %.3f' % precision_score(yts,y_pred))
print('재현율: %.3f' % recall_score(yts,y_pred))
print('F1: %.3f' % f1_score(yts,y_pred))
```

```
confusion_matrix
          Predict[0]  Predict[1]
True[0]       20827        1225
True[1]        1445        4133


classification_report
               precision    recall  f1-score   support

           0       0.94      0.94      0.94     22052
           1       0.77      0.74      0.76      5578

    accuracy                           0.90     27630
   macro avg       0.85      0.84      0.85     27630
weighted avg       0.90      0.90      0.90     27630



잘못 분류된 샘플 개수: 2670
정확도: 0.903
정밀도: 0.771
재현율: 0.741
F1: 0.756
```

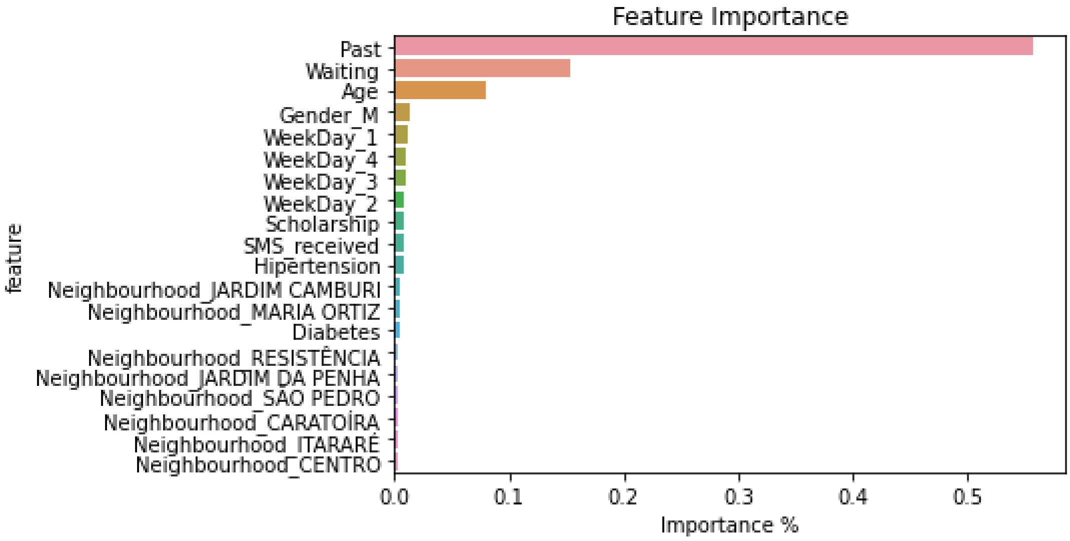
In [66]:

```
colors = ['red', 'black', 'blue', 'green']
linestyles = [':', '--', '-.', '-']
all_clf = [tree]
clf_labels = ['Decision tree']

for clf, label, clr, ls in zip(all_clf, clf_labels, colors, linestyles):
    clf.fit(Xtr, ytr)
    y_pred = clf.predict_proba(Xts)[:, 1]
    fpr, tpr, thresholds = roc_curve(y_true=yts,
                                     y_score=y_pred)
    roc_auc = auc(x=fpr, y=tpr)
    plt.plot(fpr, tpr,
             color=clr,
             linestyle=ls,
             label='%s (auc = %0.3f)' % (label, roc_auc))

plt.legend(loc='lower right')
plt.plot([0, 1], [0, 1],
         linestyle='--',
         color='gray',
         linewidth=2)

plt.xlim([-0.1, 1.1])
plt.ylim([-0.1, 1.1])
plt.grid(alpha=0.5)
plt.xlabel('False positive rate (FPR)')
plt.ylabel('True positive rate (TPR)')
plt.title('ROC-AUC')
plt.show()
```



In [67]:
```
feature_importance = pd.DataFrame({'feature' : X.columns,
                                   'importances' : tree.feature_importances_})
ordered = feature_importance.sort_values(['importances'], ascending = False)
top = ordered[:20]
print(top)

sns.barplot(x = 'importances', y = 'feature', data = top)
plt.xlabel('Importance %')
plt.title('Feature Importance')
plt.show()
```

```
                        feature  importances
8                          Past     0.557562
7                       Waiting     0.152605
0                           Age     0.079665
9                      Gender_M     0.013882
89                    WeekDay_1     0.010848
92                    WeekDay_4     0.009521
91                    WeekDay_3     0.009454
90                    WeekDay_2     0.008734
1                   Scholarship     0.007597
6                  SMS_received     0.007433
2                  Hipertension     0.007205
46   Neighbourhood_JARDIM CAMBURI     0.005060
51      Neighbourhood_MARIA ORTIZ     0.004734
3                      Diabetes     0.003638
67      Neighbourhood_RESISTÊNCIA     0.003551
47  Neighbourhood_JARDIM DA PENHA     0.003302
85        Neighbourhood_SÃO PEDRO     0.003296
18        Neighbourhood_CARATOÍRA     0.003218
44          Neighbourhood_ITARARÉ     0.003189
19           Neighbourhood_CENTRO     0.003144
```


Feature Importance
Past
Waiting
Age
Gender_M
WeekDay_1
WeekDay_4
WeekDay_3
WeekDay_2
Scholarship
SMS_received
Hipertension
Neighbourhood_JARDIM CAMBURI
Neighbourhood_MARIA ORTIZ
Diabetes
Neighbourhood_RESISTÊNCIA
Neighbourhood_JARDIM DA PENHA
Neighbourhood_SÃO PEDRO
Neighbourhood_CARATOÍRA
Neighbourhood_ITARARÉ
Neighbourhood_CENTRO
feature
0.0
0.1
0.2
0.3
0.4
0.5
Importance %