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 question we are trying to answer here is why 30% of patients miss their scheduled appointment. We are trying to predict the most important factors that affect the attendance of the patient.

Some questions we can ask to help us explore the data:

1)Does the patient's gender has a relation with the attendance?
2)Does the neighbourhood play a role in making patients don't show up? "Location of the hospital"
3)Which patients show up more? Does old age take care of their health more than youth?
4)Does the disease type affect the patient's show-up?

In [1]:
```
# importing all important packages
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
# Load data into a panda's data frame
data = pd.read_csv('Dataset.csv')
data.head()
```

Out[2]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	H
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	0	
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	0	
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	0	
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	

In [3]:
```
# Checking the shape of the data frame
data.shape
```

Out[3]:
```
(110527, 14)
```

In [4]:
```
## Information of data set
data.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 [5]:
```
#summary statistics
data.describe()
```

Out[5]:

	PatientId	AppointmentID	Age	Scholarship	Hipertension	Diabetes	Alcoholism
count	1.105270e+05	1.105270e+05	110527.000000	110527.000000	110527.000000	110527.000000	110527.000000
mean	1.474963e+14	5.675305e+06	37.088874	0.098266	0.197246	0.071865	0.030400
std	2.560949e+14	7.129575e+04	23.110205	0.297675	0.397921	0.258265	0.171686
min	3.921784e+04	5.030230e+06	-1.000000	0.000000	0.000000	0.000000	0.000000
25%	4.172614e+12	5.640286e+06	18.000000	0.000000	0.000000	0.000000	0.000000
50%	3.173184e+13	5.680573e+06	37.000000	0.000000	0.000000	0.000000	0.000000
75%	9.439172e+13	5.725524e+06	55.000000	0.000000	0.000000	0.000000	0.000000
max	9.999816e+14	5.790484e+06	115.000000	1.000000	1.000000	1.000000	1.000000

In [6]:
```
#checking any null values exist data frame
data.isnull().sum()
```

Out[6]:
```
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 [7]:
```
#checking any duplicated values exit data frame
data.duplicated().sum()
```

Out[7]:
```
0
```

In [8]:
```
#droping patient-id and appointment-id as the columns are not useful for the present ana
data = data.drop(['PatientId','AppointmentID'],axis=1)
data.head()
```

Out[8]:

	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholis

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

In [9]:
```
# change datatype for columns ScheduledDay, AppointmentDay to Date Time
data['ScheduledDay'] = pd.to_datetime(data['ScheduledDay'], errors='coerce')
data['AppointmentDay'] = pd.to_datetime(data['AppointmentDay'], errors='coerce')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 12 columns):
 #   Column          Non-Null Count   Dtype              
---  ------          --------------   -----              
 0   Gender          110527 non-null  object             
 1   ScheduledDay    110527 non-null  datetime64[ns, UTC]
 2   AppointmentDay  110527 non-null  datetime64[ns, UTC]
 3   Age             110527 non-null  int64              
 4   Neighbourhood   110527 non-null  object             
 5   Scholarship     110527 non-null  int64              
 6   Hipertension    110527 non-null  int64              
 7   Diabetes        110527 non-null  int64              
 8   Alcoholism      110527 non-null  int64              
 9   Handcap         110527 non-null  int64              
 10  SMS_received    110527 non-null  int64              
 11  No-show         110527 non-null  object             
dtypes: datetime64[ns, UTC](2), int64(7), object(3)
memory usage: 10.1+ MB
```

In [10]:
```
#checking no of unique values existing data frame
data.nunique()
```

Out[10]:
```
Gender                2
ScheduledDay      103549
AppointmentDay        27
Age                  104
Neighbourhood         81
Scholarship            2
Hipertension           2
Diabetes               2
Alcoholism             2
Handcap                5
SMS_received           2
No-show                2
dtype: int64
```

In [11]:
```
data.Age.value_counts()
```

Out[11]:
```
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
```

In [12]:
```
# In the Age column age will never be -1 so dropping the complete row
data = data[data['Age']>0]
data.Age.value_counts()
```

Out[12]:
```
1      2273
52     1746
49     1652
53     1651
56     1635
       ...
98        6
115       5
100       4
102       2
99        1
Name: Age, Length: 102, dtype: int64
```

In [13]:
```
data.Handcap.value_counts()
```

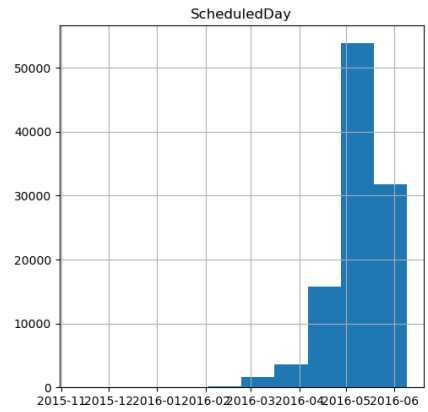
Out[13]:
```
0    104747
1      2041
2       183
3        13
4         3
Name: Handcap, dtype: int64
```

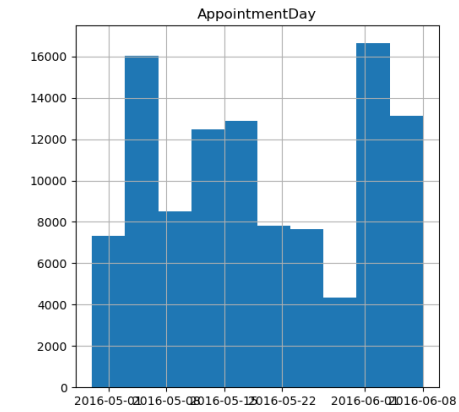
In [14]:
```
# dropping columns which greater than or equal to 2 as they are not making any sense
data = data[data['Handcap']<2]
data.Handcap.value_counts()
```

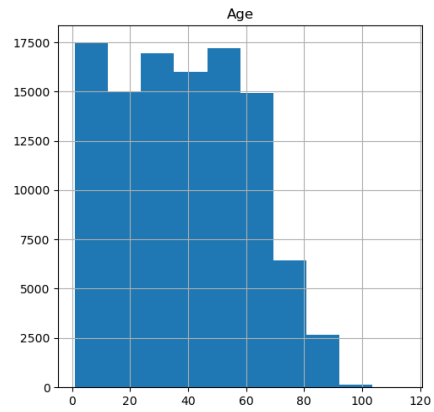
Out[14]:
```
0    104747
1      2041
Name: Handcap, dtype: int64
```

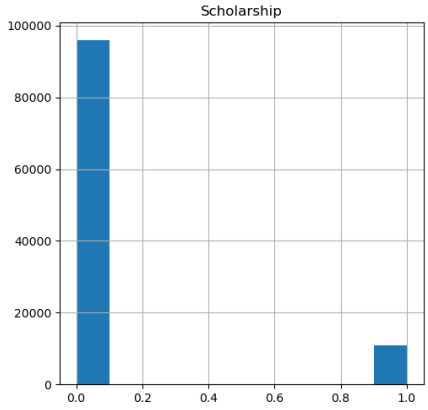
In [15]:
```
data.hist(figsize=(20,20))
```

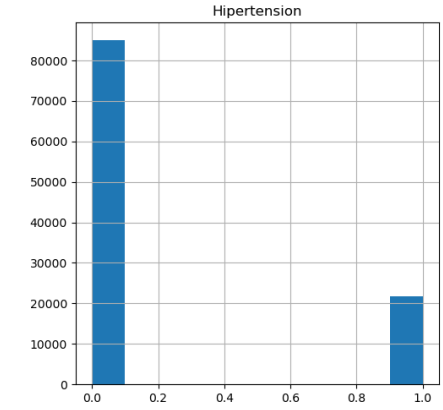
Out[15]:
```
array([[<AxesSubplot:title={'center':'ScheduledDay'}>,
        <AxesSubplot:title={'center':'AppointmentDay'}>,
        <AxesSubplot:title={'center':'Age'}>],
       [<AxesSubplot:title={'center':'Scholarship'}>,
        <AxesSubplot:title={'center':'Hipertension'}>,
        <AxesSubplot:title={'center':'Diabetes'}>],
       [<AxesSubplot:title={'center':'Alcoholism'}>,
        <AxesSubplot:title={'center':'Handcap'}>,
        <AxesSubplot:title={'center':'SMS_received'}>]], dtype=object)
```

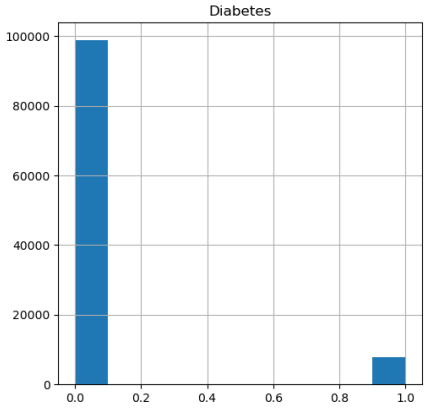


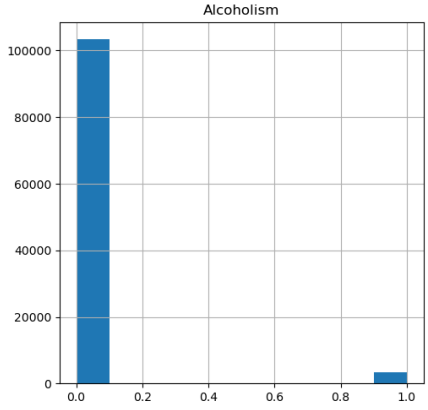


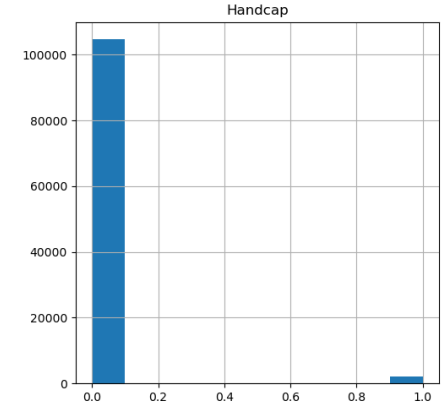


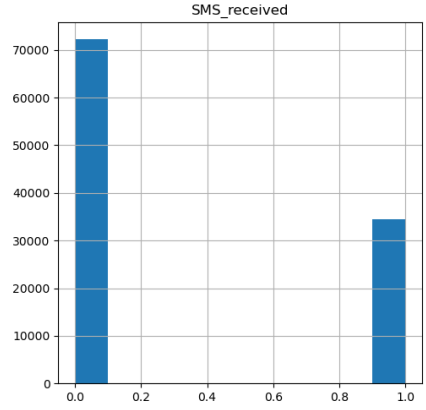












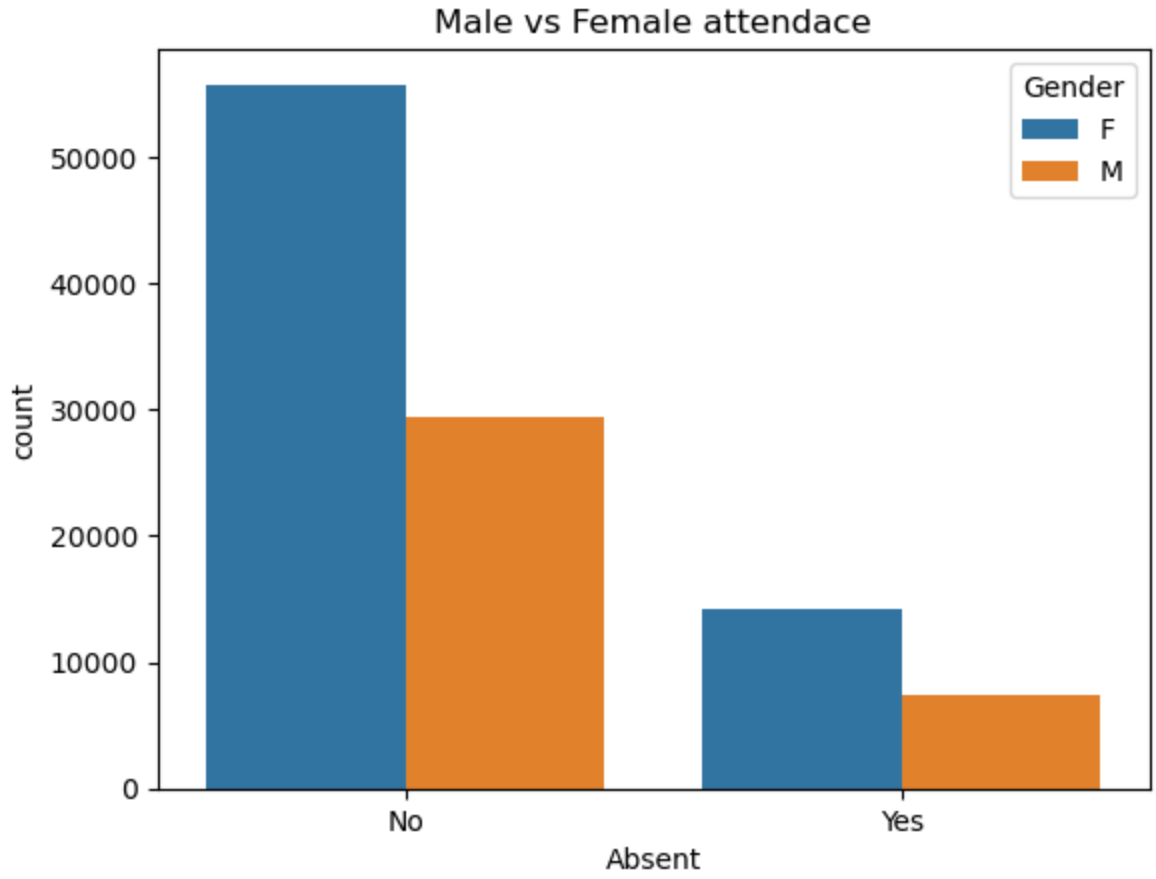
In [16]:
```
# Rename incorrect columns names
data = data.rename(columns={'Handcap':'Handicap', 'Hipertension':'Hypertension','No-show
data.head()
```

Out[16]:

	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholi
0	F	2016-04-29 18:38:08+00:00	2016-04-29 00:00:00+00:00	62	JARDIM DA PENHA	0	1	0	
1	M	2016-04-29 16:08:27+00:00	2016-04-29 00:00:00+00:00	56	JARDIM DA PENHA	0	0	0	
2	F	2016-04-29 16:19:04+00:00	2016-04-29 00:00:00+00:00	62	MATA DA PRAIA	0	0	0	
3	F	2016-04-29 17:29:31+00:00	2016-04-29 00:00:00+00:00	8	PONTAL DE CAMBURI	0	0	0	
4	F	2016-04-29 16:07:23+00:00	2016-04-29 00:00:00+00:00	56	JARDIM DA PENHA	0	1	1	

Does the patient's gender has a relation with the attendance?

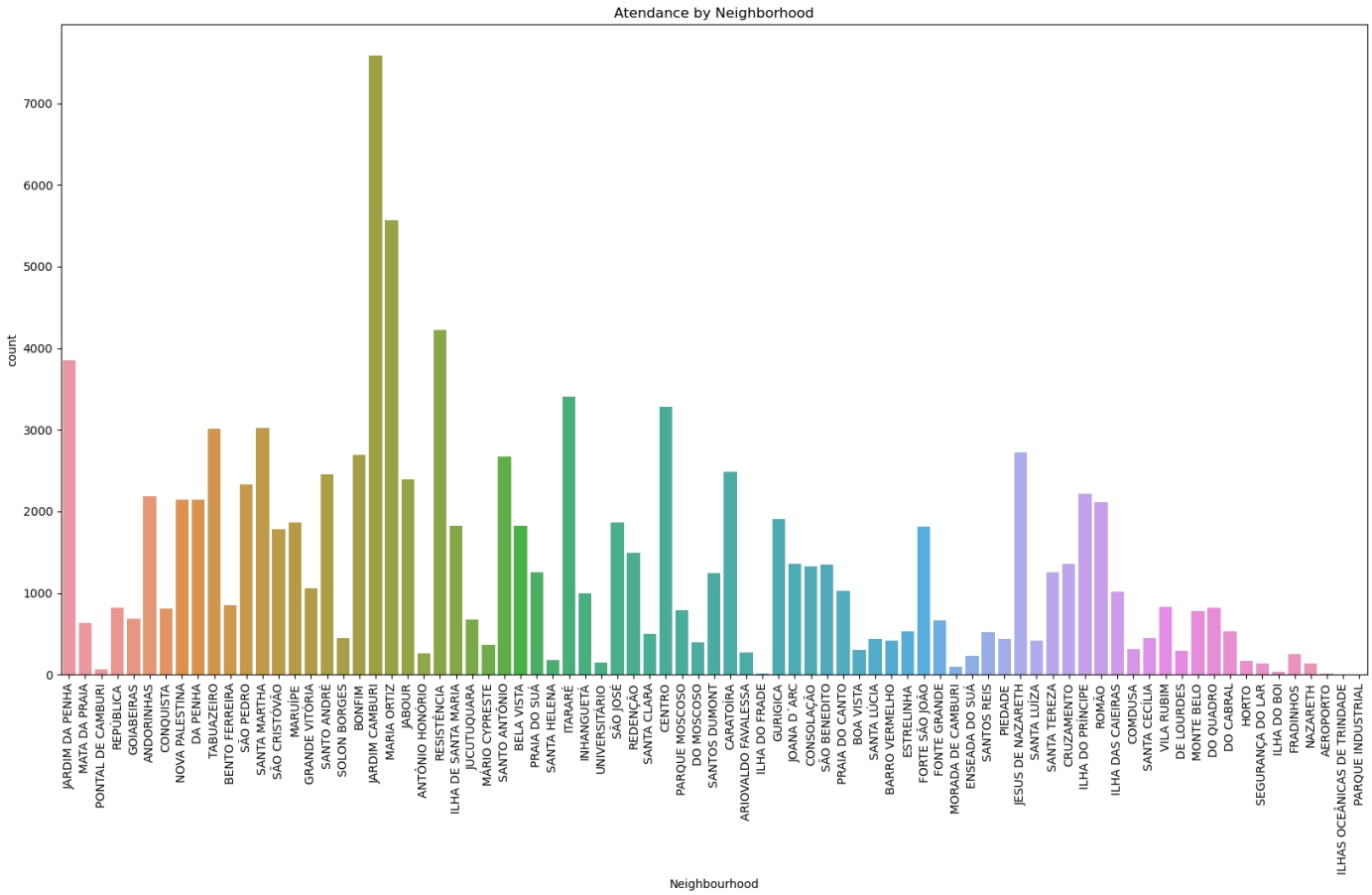
In [17]:
```
sns.countplot(x=data['Absent'], hue=data['Gender']);
plt.title('Male vs Female attendace');
```



The number of females showing up is greater than the number of males. Maybe because we have more data on females but that also shows that they visit hospitals more in general.

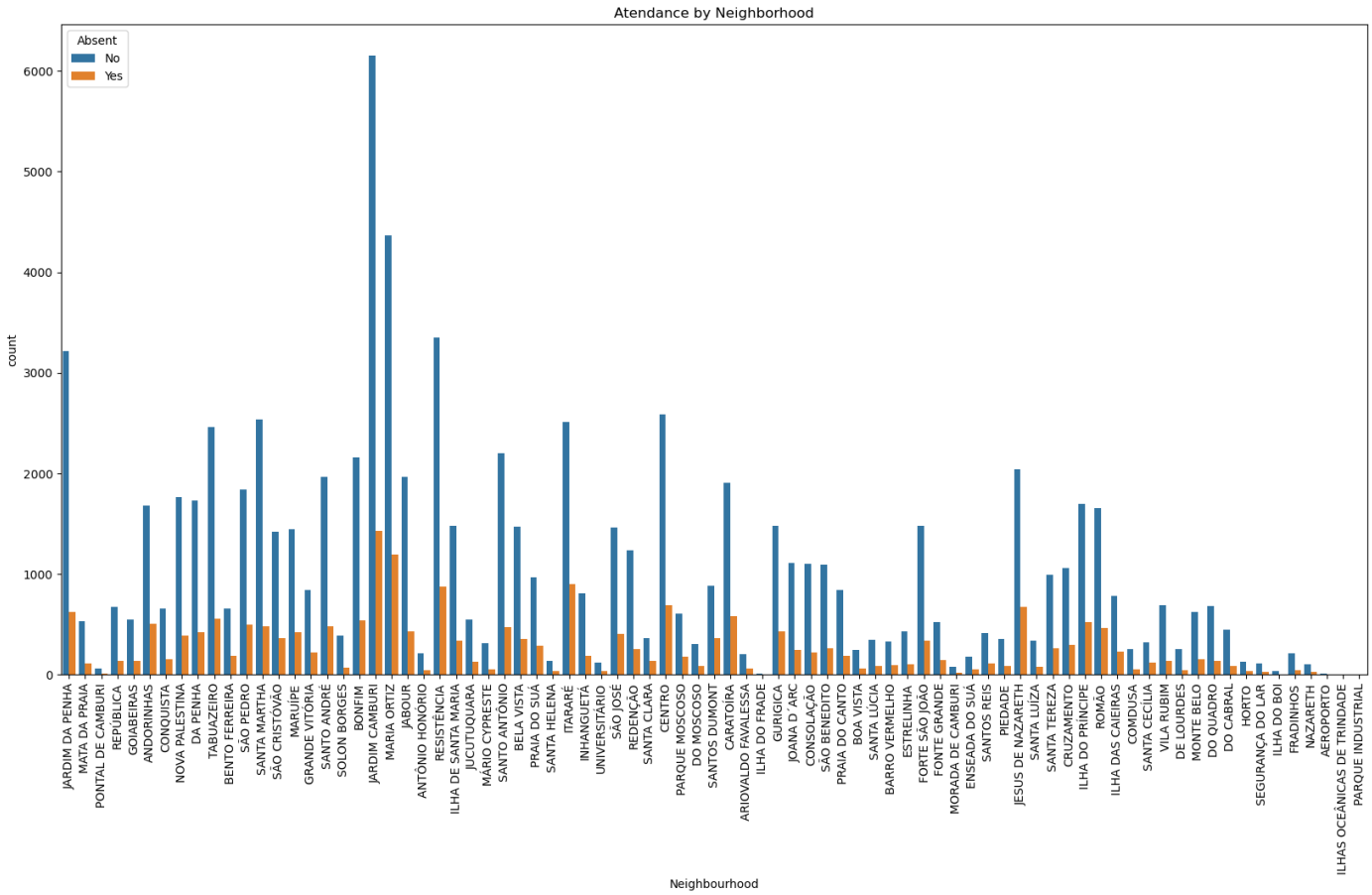
Does the neighbourhoods play a role in making patients don't show up? "Location of the hospital

In [18]:
```
plt.figure(figsize=(20,10))
sns.countplot(x=data['Neighbourhood']);
plt.title('Atendance by Neighborhood');
plt.xticks(rotation=90);
```



We see that some neighborhood have more people show up for their appointment and this indicates that this area have increase in diseases

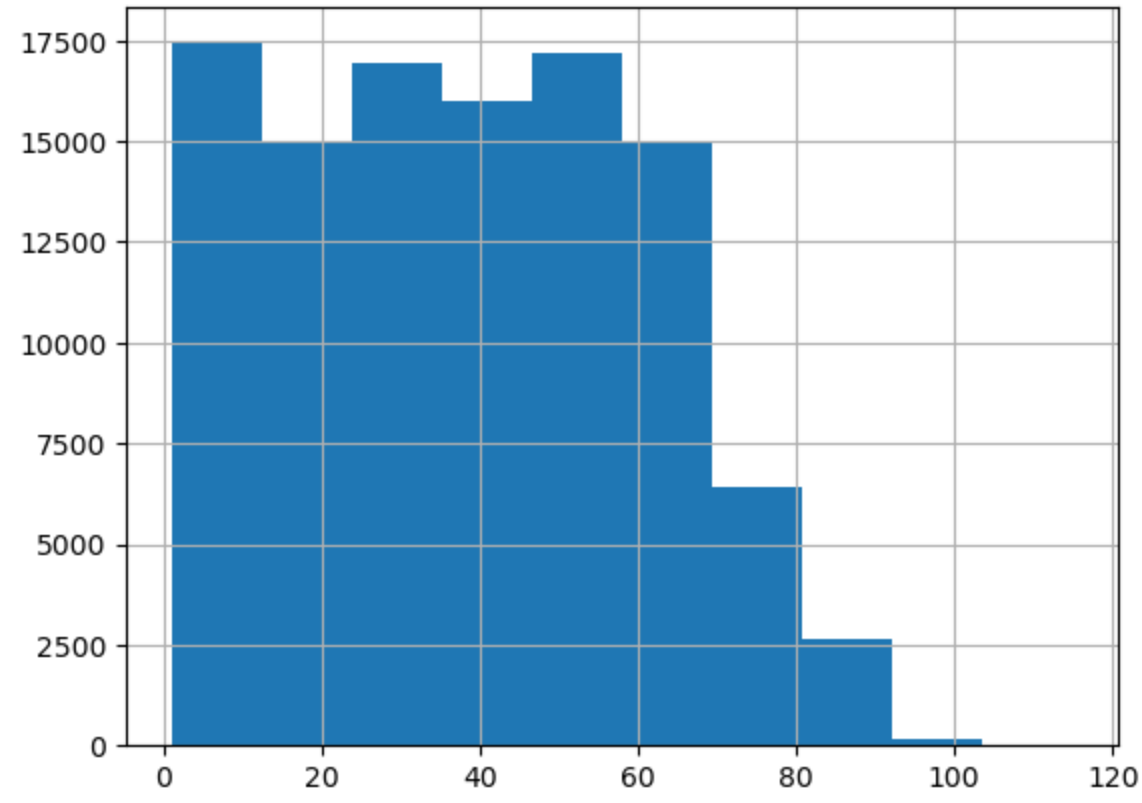
In [19]:
```
plt.figure(figsize=(20,10))
sns.countplot(x=data['Neighbourhood'],hue=data['Absent']);
plt.title('Atendance by Neighborhood');
plt.xticks(rotation=90);
```



We see that some neighbourhoods have more people showing up for their appointment and this indicates that this area has an increase in diseases

Which patients show up more? Does old age take care of their health more than youth?

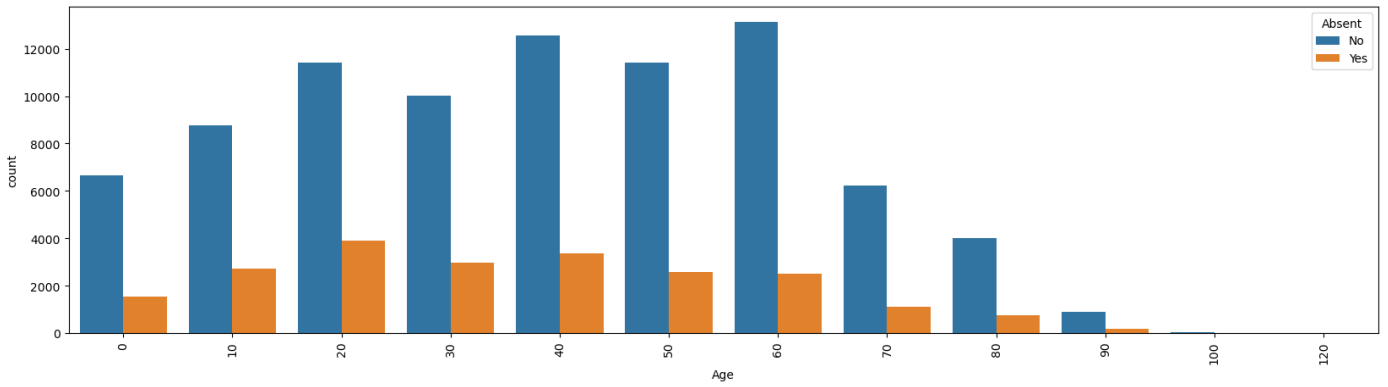
In [20]:
```
data['Age'].hist(bins=10);
```



In [21]:
```
data['Age'] = [round(a,-1) for a in data['Age']]  # this trick makes age easier as I div
                                                  #it easier visualizing
data['Age'].value_counts()
```

Out[21]:
```
40     15939
60     15605
20     15310
50     13995
30     13002
10     11504
0       8190
70      7356
80      4744
90      1074
100       64
120        5
Name: Age, dtype: int64
```

In [22]:
```
plt.figure(figsize=(20,5))
sns.countplot(x=data['Age'], hue=data['Absent'])
plt.xticks(rotation=90);
```



This shows that the ratio is close but youth still show up more which is the opposite of what we argued at the beginning

Does the disease type affect the patient's show-up?

In [23]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 106788 entries, 0 to 110526
Data columns (total 12 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Gender          106788 non-null  object
 1   ScheduledDay    106788 non-null  datetime64[ns, UTC]
 2   AppointmentDay  106788 non-null  datetime64[ns, UTC]
 3   Age             106788 non-null  int64
 4   Neighbourhood   106788 non-null  object
 5   Scholarship     106788 non-null  int64
 6   Hypertension    106788 non-null  int64
 7   Diabetes        106788 non-null  int64
 8   Alcoholism      106788 non-null  int64
 9   Handicap        106788 non-null  int64
 10  SMS_received    106788 non-null  int64
 11  Absent          106788 non-null  object
dtypes: datetime64[ns, UTC](2), int64(7), object(3)
memory usage: 10.6+ MB
```

In [24]:
```
disease_columns = data[['Hypertension','Diabetes','Alcoholism','Handicap']]
```

```python
disease_columns.head()
```

Out[24]:

	Hypertension	Diabetes	Alcoholism	Handicap
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	1	1	0	0

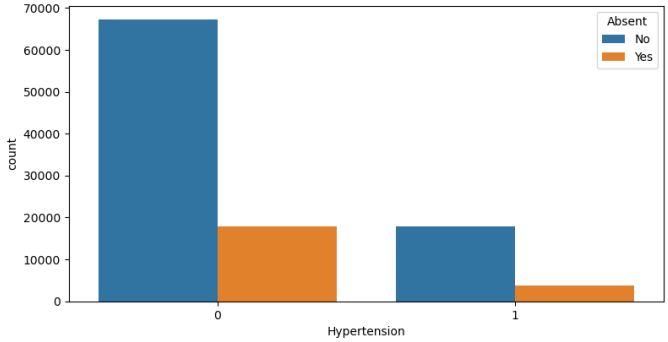
In [25]:

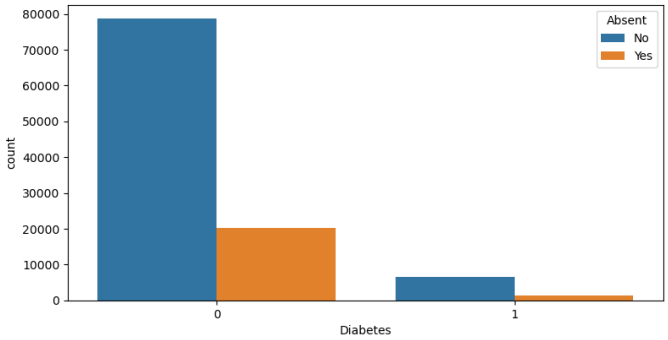
```python
plt.figure(figsize=(20,10));
plt.subplot(2,2,1)
sns.countplot(disease_columns['Hypertension'],hue=data['Absent'])
plt.subplot(2,2,2)
sns.countplot(disease_columns['Diabetes'],hue=data['Absent'])
plt.subplot(2,2,3)
sns.countplot(disease_columns['Alcoholism'],hue=data['Absent'])
plt.subplot(2,2,4)
sns.countplot(disease_columns['Handicap'],hue=data['Absent'])
```

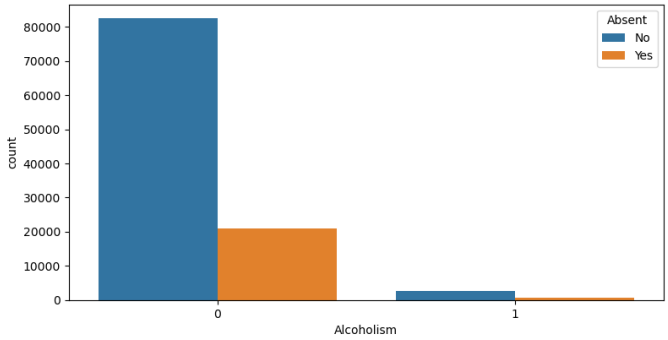
```
C:\Users\SAI RAM\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: P
ass the following variable as a keyword arg: x. From version 0.12, the only valid positi
onal argument will be `data`, and passing other arguments without an explicit keyword wi
ll result in an error or misinterpretation.
  warnings.warn(
C:\Users\SAI RAM\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: P
ass the following variable as a keyword arg: x. From version 0.12, the only valid positi
onal argument will be `data`, and passing other arguments without an explicit keyword wi
ll result in an error or misinterpretation.
  warnings.warn(
C:\Users\SAI RAM\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: P
ass the following variable as a keyword arg: x. From version 0.12, the only valid positi
onal argument will be `data`, and passing other arguments without an explicit keyword wi
ll result in an error or misinterpretation.
  warnings.warn(
C:\Users\SAI RAM\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: P
ass the following variable as a keyword arg: x. From version 0.12, the only valid positi
onal argument will be `data`, and passing other arguments without an explicit keyword wi
ll result in an error or misinterpretation.
  warnings.warn(
```

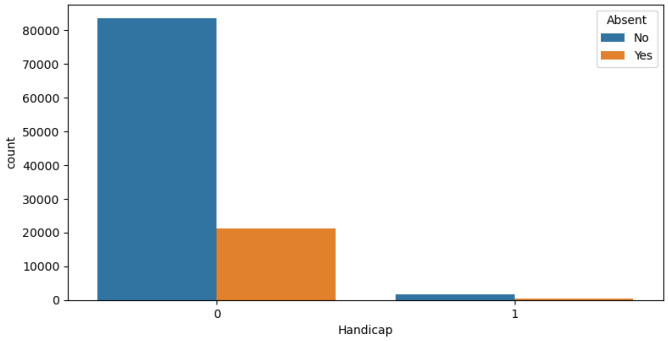
Out[25]:

```
<AxesSubplot:xlabel='Handicap', ylabel='count'>
```









We see that most of them don't have a disease and show up for appointments but we notice that patients with hypertension show up either when they are infected or not which is a mark that hypertension will probably show up more.

Conclusions

Now we can see the factors that affect the absence of the patients more clearly. Gender and age are the most important factor as we saw earlier that females and youth show up for their appointment more than males and old people. Neighbourhoods and hypertension come after gender and age as there are some neighbourhoods where the diseases are spread and patients with hypertension tend to show up if they have it or not. So we need to search for more factors to help the patient remember their appointments and show up.

Always open for feedback and suggestions.If it helps Thumbs Up !!!