Healthcare Appointments Attendance Analysis

April 28, 2023

1 Project: Healthcare Appointments Attendance Analysis

1.1 Table of Contents

Introduction

1.1.1 Dataset Description

In this project I will be analyzing a dataset of medical appointments in Brazil. I will be exploring how different variables in the dataset may have an affect on whether or not patients show up to their scheduled appointments.

1.1.2 Question(s) for Analysis

please note that answers are based on the dataset and is tentative and nothing is definitive.

Question 1: Are patients more likely to schedule appointments closer or further than the actual appointment day? Does the length of days between the schedule date and the actual day of the appointment affect whether or not patients will show up to their scheduled appointments?

Question 2: What correlation does "Age" have on the number of no-shows? does the "Neighbourhood" and amount of scholarships per neighbourhood affet the number of no-shows?

In [53]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy import stats
```

Data Wrangling/Cleaning

1.1.3 General Inspection

```
In [54]: patient_df = pd.read_csv('noshowappointments-kagglev2-may-2016.csv')
         patient_df.head()
```

```
Out[54]:       PatientId  AppointmentID Gender          ScheduledDay  \
         0  2.987250e+13        5642903      F  2016-04-29T18:38:08Z
         1  5.589978e+14        5642503      M  2016-04-29T16:08:27Z
         2  4.262962e+12        5642549      F  2016-04-29T16:19:04Z
         3  8.679512e+11        5642828      F  2016-04-29T17:29:31Z
         4  8.841186e+12        5642494      F  2016-04-29T16:07:23Z

                  AppointmentDay  Age      Neighbourhood  Scholarship  Hipertension  \
         0  2016-04-29T00:00:00Z   62    JARDIM DA PENHA            0             1
         1  2016-04-29T00:00:00Z   56    JARDIM DA PENHA            0             0
         2  2016-04-29T00:00:00Z   62      MATA DA PRAIA            0             0
         3  2016-04-29T00:00:00Z    8  PONTAL DE CAMBURI            0             0
         4  2016-04-29T00:00:00Z   56    JARDIM DA PENHA            0             1

            Diabetes  Alcoholism  Handcap  SMS_received No-show
         0         0           0        0             0      No
         1         0           0        0             0      No
         2         0           0        0             0      No
         3         0           0        0             0      No
         4         1           0        0             0      No
```

1.1.4 *Column Descriptions*

PatientId = The ID number the patient is assigned to. **AppointmentID** = The ID the patient's appointment is assigned to. **Gender** = Classifies if the patient is Male or Female. **ScheduledDay** = The day that the patient made the appointment. **AppointmentDay** = The actual day of the appointment. **Age** = The patient's age. **Neighbourhood** = The neighborhood that the patient lives in. **Scholarship** = If the patient belongs to the social welfare program "Bolsa Familia". **Hipertension, Diabetes, Alcoholism, Handcap** = Marked 1 if patient has this disease, 0 if the patient doesn't. **SMS_recieved** = Whether or not a text message reminder was received. **No-Show** = If the column value was "No" then the patient showed up, If the column value was "Yes" that means that the patient did NOT show up.

1.1.5 Checking the number of rows/columns in this dataset.

We see below that this dataset has 110,527 rows and 14 columns.

```
In [55]: patient_df.shape
```

```
Out[55]: (110527, 14)
```

1.1.6 Checking for null values in this dataset.

When looking at the sum of nulls for every column, the output is 0, which means we have no null values.

In [56]:
```
patient_df.isna().sum()
```

Out[56]:
```
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
```

1.1.7 Checking for duplicate values in this dataset.

again, as we can see, the output is 0. So this dataset is free of duplicate values.

In [57]:
```
patient_df.duplicated().sum()
```

Out[57]:
```
0
```

Question 1: Length Between Scheduled day and Day of appointment influence on no-shows To answer the first question, we need 3 columns. So a copy of the patient dataframe will be created as a copy for the first question. We need specifically the ScheduledDay, AppointmentDay and No-show columns.

For question 1 notice that the ScheduledDay column has times that are after the AppointmentDay times, these records may indicate that the patient arrived before scheduling the appointment. This data will make the first question inaccurate if we kept this in the dataset because the No-show value for these rows will always be "No". During cleanup I will find the values that have a negative time and exclude it from the results.

First, There will be a subset of the patient dataframe which has the columns that are necessary for analysis.

Second, the "ScheduledDay" rows will be subtracted from the "AppointmentDay" rows to show the days between the appointment and the time of scheduling. To reconfirm my statement above, you see some differences in days that are -1.

In [58]:
```
length_days = patient_df[['ScheduledDay','AppointmentDay','No-show']].copy(deep=True)
length_days.head()
```

Out[58]:

	ScheduledDay	AppointmentDay	No-show
0	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	No
1	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	No
2	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	No
3	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	No
4	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	No

```
In [59]: length_days['AppointmentDay'] = pd.to_datetime(length_days['AppointmentDay'])
         length_days['ScheduledDay'] = pd.to_datetime(length_days['ScheduledDay'])

         length_days['Difference'] = (length_days['AppointmentDay'] - length_days['ScheduledDay'

         length_days.head()
```

```
Out[59]:          ScheduledDay AppointmentDay No-show  Difference
         0 2016-04-29 18:38:08     2016-04-29      No          -1
         1 2016-04-29 16:08:27     2016-04-29      No          -1
         2 2016-04-29 16:19:04     2016-04-29      No          -1
         3 2016-04-29 17:29:31     2016-04-29      No          -1
         4 2016-04-29 16:07:23     2016-04-29      No          -1
```

Continuing on, the first print statement in the block shows how many rows are in the Difference column.

The next print statement shows how many values in the difference column is equal to or below 0.

If we subtract the second print statement from the first statement, it should be equal to the amount of values that are above 0 (the third print statement).

Another way to confirm if there are any values less than or equal to 0 is to use the any() function. The output from the last print statement in the block outputs "False", which means there are no longer any values that are 0 or below in the Difference column.

```
In [60]: print(f'The intial amount of rows in the "Difference" column is: {(len(length_days.Diff
         print(f'The amount of days that is 0 or below is: {len(length_days[length_days.Differen
         length_days.drop(length_days[length_days.Difference <= 0 ].index, inplace = True)
         print(f'Here is the new amount of rows in the "Difference" column after the drop: {len(
         print(f'Are there any values left in the "Difference" column that is 0 or below? {(leng
```

```
The intial amount of rows in the "Difference" column is: 110527
The amount of days that is 0 or below is: 43781
Here is the new amount of rows in the "Difference" column after the drop: 66746
Are there any values left in the "Difference" column that is 0 or below? False
```

Question 2: Age vs No-show In question 2, in the 'Age vs No-show' correlation, it doesnt make sense that a patient is negative years old. Note that a person that is 0 or older may be a possibilty (newborns). below, I will first find out how many patients fall into the category of being negative years old, identify where that patient is on the data set and confirm/drop the patient from the dataset.

The "age_noshow" variable is a copy subset of the patient_df dataframe and will be used to answer question 2.

The "below_zero" variable is a container to use in the 4th step below to simplify the drop function that rids of the rows in the "age_noshow" dataframe where the column "Age" contains values that are <= -1.

The "neg_location" variable below shows a list of the locations of negative numbers.

In [61]:
```
age_noshow = patient_df[['Age','No-show']].copy(deep = True)
age_noshow.head()
```

Out[61]:

	Age	No-show
0	62	No
1	56	No
2	62	No
3	8	No
4	56	No

In [62]:
```
below_zero = age_noshow[age_noshow.Age < 0]
neg_location = np.where(age_noshow.Age < 0) #list of locations where 'Age' <= -1

print(f'Location of the values less than 0: {neg_location}')

print(f'Amount of values that are negative: {len(neg_location)}')

print(f'Initial amount of rows in the "Age" column: {len(age_noshow.Age)}')

age_noshow.drop(below_zero.index, inplace = True)

print(f'Amount of rows in the "Age" column after dropping negative values: {len(age_nos

print(f'Are there any values in the "Age" column that is under 0? {(age_noshow.Age <= -
```

```
Location of the values less than 0: (array([99832]),)
Amount of values that are negative: 1
Initial amount of rows in the "Age" column: 110527
Amount of rows in the "Age" column after dropping negative values: 110526
Are there any values in the "Age" column that is under 0? False
```

Question 2: Neighbourhood vs No-show In question 2, I will need to create another subset of the patient dataframe to display neighborhoods and the number of patients that arrived at their scheduled appointment, and the number that did not.

Below will compute the total number of No-shows by neighborhood, This is categorized by patients that made it to their appointments ('no_by_neighbourhood') and those who didn't ('yes_by_neighbourhood'). The sum of these counts will add up to 110,527 rows which was the total amount in the original patient dataframe.

Remember that "Yes" means that the patient is a no-show (did not make it to the appointment), "No" means that the patient showed up.

In [63]:
```
neighbourhood_noshow = patient_df[['Neighbourhood','No-show']].copy(deep = True)
```

In [64]:
```
neighbourhood_noshow.head()
```

Out[64]:

	Neighbourhood	No-show
0	JARDIM DA PENHA	No
1	JARDIM DA PENHA	No

```
2      MATA DA PRAIA      No
3  PONTAL DE CAMBURI      No
4    JARDIM DA PENHA      No
```

```
In [65]: no_by_neighbourhood =neighbourhood_noshow.groupby('Neighbourhood')['No-show'].apply(lam
         yes_by_neighbourhood =neighbourhood_noshow.groupby('Neighbourhood')['No-show'].apply(la
```

Confirmation of grouping by Neighbourhood and counting by No-show columns

```
In [66]: no_by_neighbourhood.head()

Out[66]:          Neighbourhood  Count
         0            AEROPORTO      7
         1           ANDORINHAS   1741
         2      ANTÔNIO HONÓRIO    221
         3  ARIOVALDO FAVALESSA    220
         4       BARRO VERMELHO    332
```

```
In [67]: yes_by_neighbourhood.head()

Out[67]:          Neighbourhood  Count
         0            AEROPORTO      1
         1           ANDORINHAS    521
         2      ANTÔNIO HONÓRIO     50
         3  ARIOVALDO FAVALESSA     62
         4       BARRO VERMELHO     91
```

```
In [68]: print(f'Total number of patients that showed up to their scheduled appointments: {no_by
         print(f'Total number of patients that did not show up to their scheduled appointments:
         print(f'Total number of patients that scheduled an appointment: {no_by_neighbourhood.Co

Total number of patients that showed up to their scheduled appointments: 88208
Total number of patients that did not show up to their scheduled appointments: 22319
Total number of patients that scheduled an appointment: 110527
```

Exploratory Data Analysis

1.1.8 Visual analysis on the affects of time between days and the number of no-shows

for this question, lets explore the data by displaying on a graph the numbers of attendance in relationship to the time between days.

"yes_count_days" is the number of no shows categorized by length of days between scheduling day and arrival day.

"no_count_days" is the number of patients that showed up categorized by length of days between scheduling day and arrival day.

```
In [69]: no_count_days = length_days.groupby('Difference')['No-show'].apply(lambda x: (x=='No').
         no_count_days.head()
```

```
Out[69]:    Difference  Count
         0           1   5123
         1           2   2093
         2           3   4059
         3           4   2405
         4           5   3036
```

```
In [70]: no_count_days.Count.sum()
```

```
Out[70]: 47337
```

```
In [71]: yes_count_days = length_days.groupby('Difference')['No-show'].apply(lambda x: (x=='Yes'
         yes_count_days.head()
```

```
Out[71]:    Difference  Count
         0           1   1602
         1           2    644
         2           3   1231
         3           4    872
         4           5   1001
```

```
In [72]: yes_count_days.Count.sum()
```

```
Out[72]: 19409
```

Question 1: Are patients more likely to schedule appointments closer or further than the actual appointment day? Does the length of days between the schedule date and the actual day of the appointment affect the patient's no show status?

To begin with, it does not seem like the length of days between the schedule date and the actual day of the appointment affect the patient's no_show status.(showing up vs no show) Referring to the tables and graphs below, within each Difference, patients seemed to show up more than not, consistantly. (refer to the ratio in no shows and showed up below).

```
In [73]: total_per_gap = yes_count_days.Count + no_count_days.Count
         ratio = {'Difference':no_count_days.Difference,
                  'showed-up':no_count_days.Count,
                  'No-show':yes_count_days.Count,
                  'total_appts':yes_count_days.Count + no_count_days.Count,
                  'ratio_no_show_to_total':yes_count_days.Count/(yes_count_days.Count + no_count
                  'ratio_showed_up_to_total':no_count_days.Count/(no_count_days.Count + yes_coun
         ratio_info = pd.DataFrame(ratio)

         ratio_info.head()
```

```
Out[73]:    Difference  showed-up  No-show  total_appts  ratio_no_show_to_total  \
         0           1       5123     1602         6725                0.238216
         1           2       2093      644         2737                0.235294
         2           3       4059     1231         5290                0.232703
         3           4       2405      872         3277                0.266097
```

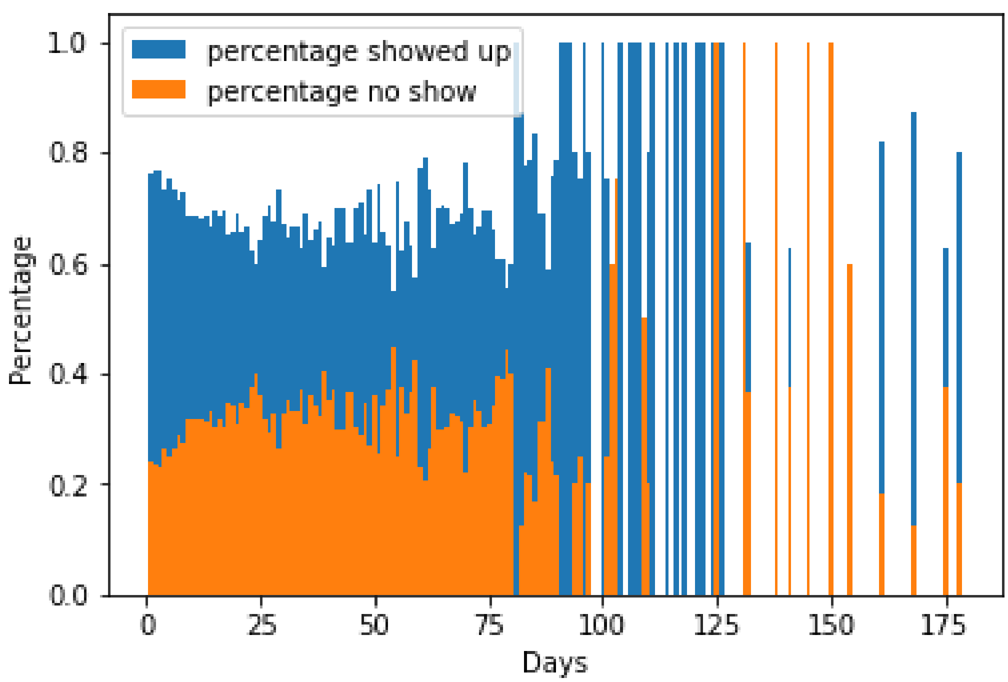
```
4             5        3036      1001           4037                    0.247956

   ratio_showed_up_to_total
0                  0.761784
1                  0.764706
2                  0.767297
3                  0.733903
4                  0.752044
```

As we can see below in the graph, almost every gap in days show that the percentage of patients that showed up outnumbered those that didn't.

```
In [74]: x = no_count_days.Difference
         width = 1

         fig, ax = plt.subplots()
         ax.bar(x,ratio_info.ratio_showed_up_to_total, width, label='percentage showed up')
         ax.bar(x,ratio_info.ratio_no_show_to_total, width, label='percentage no show')
         ax.set_ylabel('Percentage')
         ax.set_xlabel('Days')
         ax.legend()
         plt.show()
```



Next I will answer: are patients more likely to schedule appointments closer or further than the actual appointment day?

To do this, I will use percentiles to see the percentage of appointments that were made beneath and above the number of days between schedule/appointments.

through the method below, it's safe to say that patients are more likely to make appointments with a closer gap than a longer gap wether they show up or not. (in this case i used 60th percentile because 60 is over 50, which is just the mean. This means 50% of appointments were made before AND after the Difference).

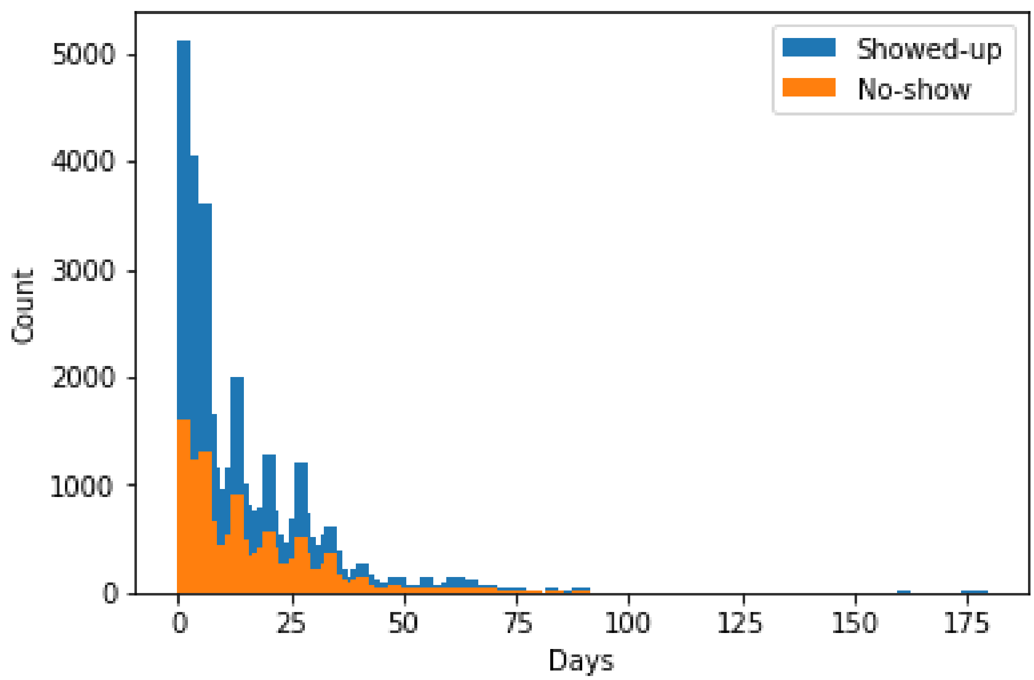
After finding the Difference in days, I compared it to the graph. The percentage for both no shows and for patients that showed up falls within 1-68 days and 1-78 days respectively.

```
In [75]: print(f'60% of no shows made appointments below {np.percentile(yes_count_days,60).round
         print(f'60% of patients that show up made appointments below {np.percentile(no_count_da
```

```
60% of no shows made appointments below 68.0 days, and 40% over 68 days.
60% of patients that show up made appointments below 78.0 days, and 40% over 78 days.
```

```
In [76]: x = no_count_days.Difference
         width = 3

         fig, ax = plt.subplots()
         ax.bar(x,no_count_days.Count, width, label='Showed-up')
         ax.bar(x,yes_count_days.Count, width, label='No-show')
         ax.set_ylabel('Count')
         ax.set_xlabel('Days')
         ax.legend()
         plt.show()
```



1.1.9 Visual analysis on the correlations of the "Age", "Neighborhood" and " Scholarship" vs. number of "No-shows"

In this section, I will give visual representations on the correlation between age/neighborhood vs the amount of patients that showed up to their appointments. Different views will be presented to give us a better understanding of the correlations and will answer the question below.

Question 2: What correlation does "Age" have on the number of no-shows? does the "Neighbourhood" and amount of scholarships per neighbourhood affet the number of no-shows?

Analyzing Age vs No-show "age_showed_up" = patients categorized by age who SHOWED up.
"age_no_show" = patients categorized by age who did NOT show up.

```
In [77]: age_showed_up = age_noshow.groupby('Age')['No-show'].apply(lambda x: (x=='No').sum()).r
```

```
In [78]: age_no_show = age_noshow.groupby('Age')['No-show'].apply(lambda x: (x=='Yes').sum()).re
```

```
In [79]: age_showed_up.head()
```

Out[79]:

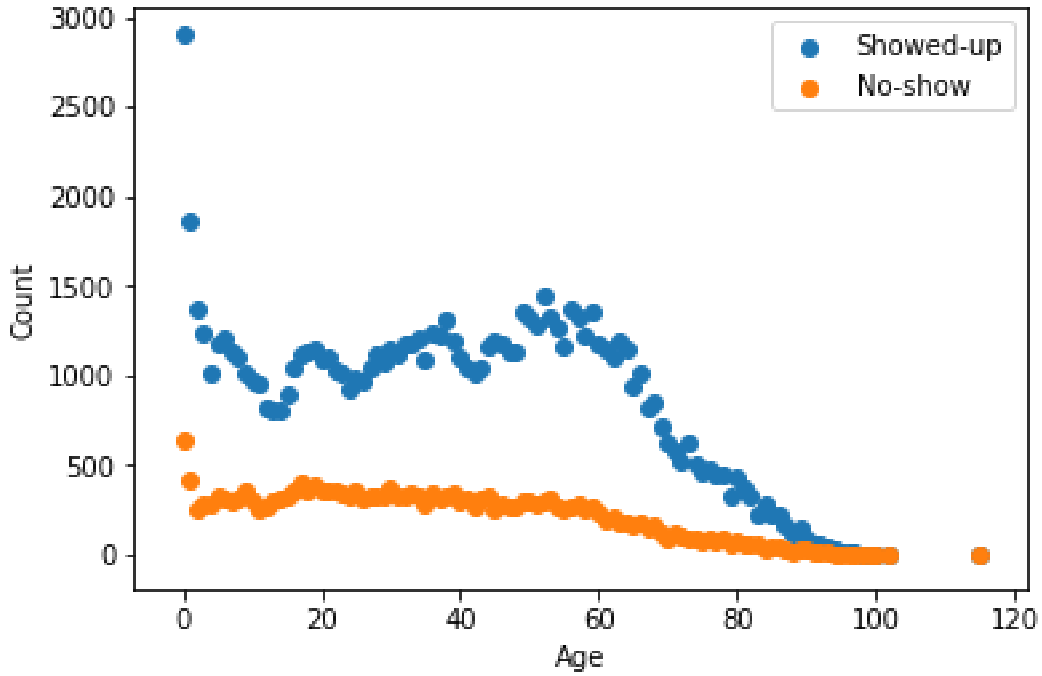
	Age	Count
0	0	2900
1	1	1858
2	2	1366
3	3	1236
4	4	1017

```
In [80]: age_no_show.head()
```

Out[80]:

	Age	Count
0	0	639
1	1	415
2	2	252
3	3	277
4	4	282

```
In [81]: x =age_no_show.Age
         fig, ax = plt.subplots()
         ax.scatter(x,age_showed_up.Count, label='Showed-up')
         ax.scatter(x,age_no_show.Count, label='No-show')
         ax.set_ylabel('Count')
         ax.set_xlabel('Age')
         ax.legend()
         plt.show()
```




```
In [82]: showed_up_slope = stats.linregress(age_showed_up['Age'], age_showed_up['Count'] )
         print(showed_up_slope)
```

```
LinregressResult(slope=-12.705978319417413, intercept=1506.1105612463255, rvalue=-0.768725357176
```

```
In [83]: no_show_slope = stats.linregress(age_no_show['Age'], age_no_show['Count'] )
         print(no_show_slope)
```

```
LinregressResult(slope=-4.051618603333198, intercept=423.87257459957232, rvalue=-0.904391839467C
```

I used a scatter plot to show a correlation between Age vs. No-shows. It seems that there is a negative correlation in both number of Showed up and number of No shows in respect to Age. According to the graph, the number of showed up has a higher negative correlation compared to the no shows. This is proven by taking the slope of each comparison above.

number of showed up has a slope of around -12.7. number of no show has a slope of around -4.1.

So what can this tell us?

Since the graph is comparing age and the counts of show ups vs no shows. We see that as the ages increase, both the number of no shows and show ups decrease. from the ages 0-60 on the graph, is where the slope starts to dip at a higher rate. With that being said, the number of patients that showed up was more sporadic. The graph's slope was alternating until the age of 60. While this was true for the patients that showed up, the no show patients showed a more consistant negative correlation. As age increased,less and less people no showed. Note that this is a double-negative. (since both the number of show ups and no shows decreased as age went up, we can infer that the number of appointments made in general went down as age increased. We also see that compared to the number of showups, the number of no shows was less in every age category as well.)

Analyzing the neighbourhood and scholarship columns Below grabs the necessary columns for this relationship: Neighborhood, No-show and Scholarship information.

```
In [84]: neigh_scholar = {'Neighbourhood': patient_df.Neighbourhood,
                          'no_shows':patient_df['No-show'],
                          'scholarship':patient_df.Scholarship}
         neighbourhood_scholar = pd.DataFrame(neigh_scholar)

         neighbourhood_scholar.head()
```

```
Out[84]:        Neighbourhood no_shows  scholarship
         0    JARDIM DA PENHA       No            0
         1    JARDIM DA PENHA       No            0
         2      MATA DA PRAIA       No            0
         3  PONTAL DE CAMBURI       No            0
         4    JARDIM DA PENHA       No            0
```

Next is grouping the number of No-shows and showed up based on neighbourhood.

```
In [85]: no_show = neighbourhood_scholar.groupby('Neighbourhood')['no_shows'].apply(lambda x: (x
```

```
In [86]: no_show
```

```
Out[86]:            Neighbourhood  Count
         0              AEROPORTO      1
         1             ANDORINHAS    521
         2         ANTÔNIO HONÓRIO     50
         3     ARIOVALDO FAVALESSA     62
         4          BARRO VERMELHO     91
         5              BELA VISTA    384
         6          BENTO FERREIRA    193
         7               BOA VISTA     58
         8                  BONFIM    550
         9               CARATOÍRA    591
         10                 CENTRO    703
         11                COMDUSA     56
         12              CONQUISTA    160
         13             CONSOLAÇÃO    237
         14             CRUZAMENTO    304
         15               DA PENHA    429
         16             DE LOURDES     47
         17              DO CABRAL     88
         18             DO MOSCOSO     92
         19              DO QUADRO    140
```

```
20       ENSEADA DO SUÁ     52
21          ESTRELINHA    106
22        FONTE GRANDE    149
23       FORTE SÃO JOÃO    346
24           FRADINHOS     48
25          GOIABEIRAS    137
26       GRANDE VITÓRIA    217
27            GURIGICA    456
28               HORTO     42
29    ILHA DAS CAIEIRAS    235
..                 ...    ...
51    PARQUE INDUSTRIAL      0
52       PARQUE MOSCOSO    179
53             PIEDADE     88
54    PONTAL DE CAMBURI     12
55       PRAIA DO CANTO    190
56         PRAIA DO SUÁ    294
57            REDENÇÃO    275
58           REPÚBLICA    143
59          RESISTÊNCIA    906
60               ROMÃO    474
61        SANTA CECÍLIA    123
62          SANTA CLARA    134
63         SANTA HELENA     37
64          SANTA LUÍZA     77
65          SANTA LÚCIA     86
66         SANTA MARTHA    496
67         SANTA TEREZA    272
68          SANTO ANDRÉ    508
69        SANTO ANTÔNIO    484
70        SANTOS DUMONT    369
71          SANTOS REIS    112
72      SEGURANÇA DO LAR     28
73         SOLON BORGES     69
74         SÃO BENEDITO    287
75        SÃO CRISTÓVÃO    363
76             SÃO JOSÉ    428
77            SÃO PEDRO    515
78          TABUAZEIRO    573
79        UNIVERSITÁRIO     32
80           VILA RUBIM    141

[81 rows x 2 columns]
```

In [87]:
```
showed_up = neighbourhood_scholar.groupby('Neighbourhood')['no_shows'].apply(lambda x:
showed_up
```

Out[87]:
```
          Neighbourhood  Count
0             AEROPORTO      7
```

1	ANDORINHAS	1741
2	ANTÔNIO HONÓRIO	221
3	ARIOVALDO FAVALESSA	220
4	BARRO VERMELHO	332
5	BELA VISTA	1523
6	BENTO FERREIRA	665
7	BOA VISTA	254
8	BONFIM	2223
9	CARATOÍRA	1974
10	CENTRO	2631
11	COMDUSA	254
12	CONQUISTA	689
13	CONSOLAÇÃO	1139
14	CRUZAMENTO	1094
15	DA PENHA	1788
16	DE LOURDES	258
17	DO CABRAL	472
18	DO MOSCOSO	321
19	DO QUADRO	709
20	ENSEADA DO SUÁ	183
21	ESTRELINHA	432
22	FONTE GRANDE	533
23	FORTE SÃO JOÃO	1543
24	FRADINHOS	210
25	GOIABEIRAS	563
26	GRANDE VITÓRIA	854
27	GURIGICA	1562
28	HORTO	133
29	ILHA DAS CAIEIRAS	836
..	...	...
51	PARQUE INDUSTRIAL	1
52	PARQUE MOSCOSO	623
53	PIEDADE	364
54	PONTAL DE CAMBURI	57
55	PRAIA DO CANTO	845
56	PRAIA DO SUÁ	994
57	REDENÇÃO	1278
58	REPÚBLICA	692
59	RESISTÊNCIA	3525
60	ROMÃO	1741
61	SANTA CECÍLIA	325
62	SANTA CLARA	372
63	SANTA HELENA	141
64	SANTA LUÍZA	351
65	SANTA LÚCIA	352
66	SANTA MARTHA	2635
67	SANTA TEREZA	1060
68	SANTO ANDRÉ	2063

```
69        SANTO ANTÔNIO   2262
70        SANTOS DUMONT    907
71          SANTOS REIS    435
72     SEGURANÇA DO LAR    117
73         SOLON BORGES    400
74         SÃO BENEDITO   1152
75        SÃO CRISTÓVÃO   1473
76             SÃO JOSÉ   1549
77            SÃO PEDRO   1933
78           TABUAZEIRO   2559
79        UNIVERSITÁRIO    120
80           VILA RUBIM    710

[81 rows x 2 columns]
```

Next is to clean up the data for the scholarship section. I noticed that neighbourhoods were repeated in the neighbourhood column. So my assumption was that everytime there was one scholarship for a neighbourhood then it would be on a seperate line. So i used the any function to see if there was any value in the scholarship column that was greater than one. Below shows false. So the assumption was true. To clean the data a bit more, I wanted to exclude any neighbourhood that did not receive a scholarship since I am analyzing for those neighbourhoods that DID, and it's affect on the number of no-shows.

```
In [88]: zero = neighbourhood_scholar[neighbourhood_scholar.scholarship == 0]
         neighbourhood_scholar.drop(zero.index, inplace = True)
```

```
In [89]: (neighbourhood_scholar['scholarship'] == 0).any()
```

```
Out[89]: False
```

```
In [90]: (neighbourhood_scholar['scholarship'] > 1).any()
```

```
Out[90]: False
```

```
In [91]: neighbourhood_scholar.head()
```

```
Out[91]:      Neighbourhood no_shows  scholarship
         12  NOVA PALESTINA       No            1
         17       CONQUISTA      Yes            1
         18  NOVA PALESTINA       No            1
         31  NOVA PALESTINA      Yes            1
         33   SÃO CRISTÓVÃO       No            1
```

Once the data was sorted out, I could then group the number of scholarships according to neighbourhood. As we see below with the data table and line chart, that scholarships were not distributed evenly around neighbourhoods. (note that on the original Kaggle website which this dataset was pulled from, states that scholarships were granted to those that were in need/ on social welfare).

```
In [92]: num_scholar = neighbourhood_scholar.groupby('Neighbourhood')['scholarship'].apply(lambd
         num_scholar
```

```
Out[92]:          Neighbourhood  Count
         0             ANDORINHAS    323
         1        ANTÔNIO HONÓRIO     14
         2    ARIOVALDO FAVALESSA     52
         3             BELA VISTA    225
         4         BENTO FERREIRA     23
         5              BOA VISTA     23
         6                 BONFIM    373
         7              CARATOÍRA    456
         8                 CENTRO    143
         9                COMDUSA     34
         10             CONQUISTA    141
         11            CONSOLAÇÃO    199
         12            CRUZAMENTO    170
         13              DA PENHA    292
         14            DE LOURDES      5
         15             DO CABRAL     97
         16            DO MOSCOSO    111
         17             DO QUADRO    113
         18        ENSEADA DO SUÁ      6
         19            ESTRELINHA     77
         20          FONTE GRANDE     86
         21       FORTE SÃO JOÃO    140
         22             FRADINHOS     12
         23            GOIABEIRAS     36
         24        GRANDE VITÓRIA    111
         25              GURIGICA    422
         26                 HORTO      6
         27     ILHA DAS CAIEIRAS    203
         28   ILHA DE SANTA MARIA     23
         29      ILHA DO PRÍNCIPE    579
         ..                   ...    ...
         43              NAZARETH      2
         44        NOVA PALESTINA    310
         45        PARQUE MOSCOSO     10
         46               PIEDADE    115
         47     PONTAL DE CAMBURI      5
         48          PRAIA DO SUÁ    151
         49              REDENÇÃO    156
         50             REPÚBLICA      9
         51           RESISTÊNCIA    468
         52                 ROMÃO    178
         53         SANTA CECÍLIA     25
         54           SANTA CLARA     30
         55          SANTA HELENA     35
```

```
56          SANTA LUÍZA      7
57          SANTA LÚCIA      8
58         SANTA MARTHA    441
59         SANTA TEREZA    201
60          SANTO ANDRÉ    334
61        SANTO ANTÔNIO    151
62        SANTOS DUMONT    235
63          SANTOS REIS    120
64     SEGURANÇA DO LAR      9
65         SOLON BORGES     36
66         SÃO BENEDITO    404
67        SÃO CRISTÓVÃO    174
68             SÃO JOSÉ    180
69            SÃO PEDRO    321
70           TABUAZEIRO    537
71        UNIVERSITÁRIO      5
72           VILA RUBIM     75

[73 rows x 2 columns]
```

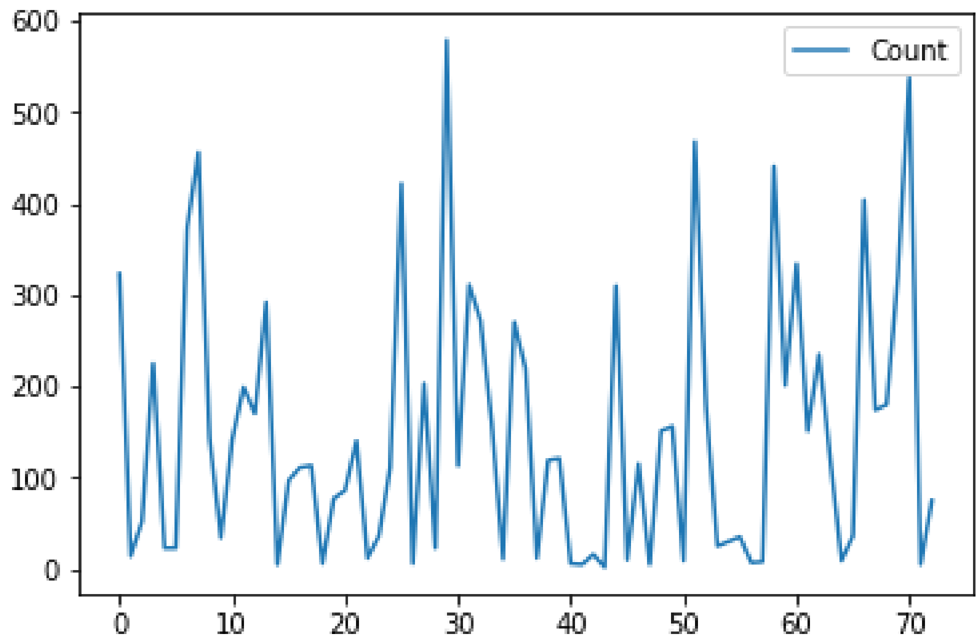
Finding the min and max amount of scholarships per neighbourhood could indicate poorer or richer neighborhoods. We can only assume that Ilha Do Principe is the poorest and Nazareth is the richest in the dataset. Keep in mind that statement is only an assumption. Or it could be that Nazareth didn't have the knowledge of the social welfare program.

```
In [93]: print(num_scholar.loc[num_scholar['Count']. idxmax()])
         print(num_scholar.loc[num_scholar['Count']. idxmin()])
```

```
Neighbourhood    ILHA DO PRÍNCIPE
Count                         579
Name: 29, dtype: object
Neighbourhood    NAZARETH
Count                   2
Name: 43, dtype: object
```

to verify the information above, refer to the graph below. Ilha Do Principe is name: 29, and has 579 scholarships. comparing to the graph the spike is the highest. Nazareth is name: 43, and has 2 scholarships. Again comparing that to the graph, we see the spike is at the lowest point in all of the other points. The data in the table is vizualized and confirmed by the graph.

```
In [94]: num_scholar.plot();
```



Moving on, I wanted to compare the number of no-shows/showed up and number of scholarships by neighbourhood. For each neighbourhood the number of no shows and the number of showed up summed should equal the total appointments made per neighborhood.

In [95]:
```
info = {'Neighbourhood': num_scholar.Neighbourhood,
                  'num_no_show':no_show.Count,
                  'num_showed_up':showed_up.Count,
                  'total_appts':no_show.Count+showed_up.Count,
                  'num_scholar':num_scholar.Count,
        }
scholar_no_show = pd.DataFrame(info)
scholar_no_show

non_exist = scholar_no_show[scholar_no_show.Neighbourhood.isna()]
scholar_no_show.drop(non_exist.index, inplace = True)

scholar_no_show
```

Out[95]:

	Neighbourhood	num_no_show	num_showed_up	total_appts	num_scholar
0	ANDORINHAS	1	7	8	323.0
1	ANTÔNIO HONÓRIO	521	1741	2262	14.0
2	ARIOVALDO FAVALESSA	50	221	271	52.0
3	BELA VISTA	62	220	282	225.0
4	BENTO FERREIRA	91	332	423	23.0
5	BOA VISTA	384	1523	1907	23.0
6	BONFIM	193	665	858	373.0
7	CARATOÍRA	58	254	312	456.0

8	CENTRO	550	2223	2773	143.0
9	COMDUSA	591	1974	2565	34.0
10	CONQUISTA	703	2631	3334	141.0
11	CONSOLAÇÃO	56	254	310	199.0
12	CRUZAMENTO	160	689	849	170.0
13	DA PENHA	237	1139	1376	292.0
14	DE LOURDES	304	1094	1398	5.0
15	DO CABRAL	429	1788	2217	97.0
16	DO MOSCOSO	47	258	305	111.0
17	DO QUADRO	88	472	560	113.0
18	ENSEADA DO SUÁ	92	321	413	6.0
19	ESTRELINHA	140	709	849	77.0
20	FONTE GRANDE	52	183	235	86.0
21	FORTE SÃO JOÃO	106	432	538	140.0
22	FRADINHOS	149	533	682	12.0
23	GOIABEIRAS	346	1543	1889	36.0
24	GRANDE VITÓRIA	48	210	258	111.0
25	GURIGICA	137	563	700	422.0
26	HORTO	217	854	1071	6.0
27	ILHA DAS CAIEIRAS	456	1562	2018	203.0
28	ILHA DE SANTA MARIA	42	133	175	23.0
29	ILHA DO PRÍNCIPE	235	836	1071	579.0
..	...	...	...	...	...
43	NAZARETH	1219	4586	5805	2.0
44	NOVA PALESTINA	424	1478	1902	310.0
45	PARQUE MOSCOSO	110	534	644	10.0
46	PIEDADE	166	658	824	115.0
47	PONTAL DE CAMBURI	16	80	96	5.0
48	PRAIA DO SUÁ	54	317	371	151.0
49	REDENÇÃO	29	106	135	156.0
50	REPÚBLICA	402	1862	2264	9.0
51	RESISTÊNCIA	0	1	1	468.0
52	ROMÃO	179	623	802	178.0
53	SANTA CECÍLIA	88	364	452	25.0
54	SANTA CLARA	12	57	69	30.0
55	SANTA HELENA	190	845	1035	35.0
56	SANTA LUÍZA	294	994	1288	7.0
57	SANTA LÚCIA	275	1278	1553	8.0
58	SANTA MARTHA	143	692	835	441.0
59	SANTA TEREZA	906	3525	4431	201.0
60	SANTO ANDRÉ	474	1741	2215	334.0
61	SANTO ANTÔNIO	123	325	448	151.0
62	SANTOS DUMONT	134	372	506	235.0
63	SANTOS REIS	37	141	178	120.0
64	SEGURANÇA DO LAR	77	351	428	9.0
65	SOLON BORGES	86	352	438	36.0
66	SÃO BENEDITO	496	2635	3131	404.0
67	SÃO CRISTÓVÃO	272	1060	1332	174.0

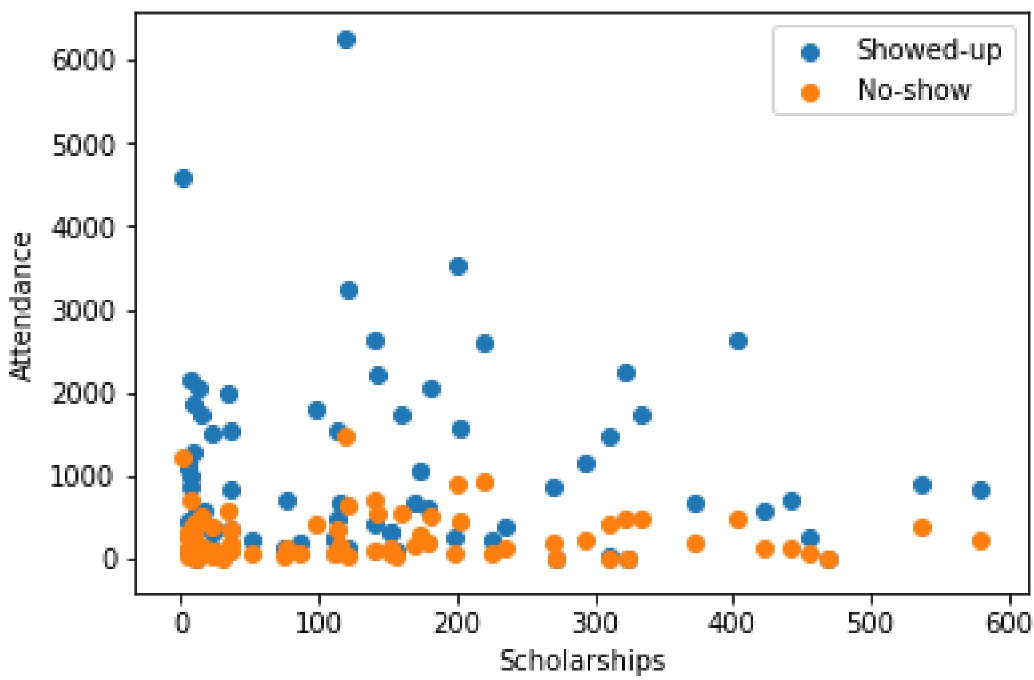
68	SÃO JOSÉ	508	2063	2571	180.0
69	SÃO PEDRO	484	2262	2746	321.0
70	TABUAZEIRO	369	907	1276	537.0
71	UNIVERSITÁRIO	112	435	547	5.0
72	VILA RUBIM	28	117	145	75.0

[73 rows x 5 columns]

In the graph below, there is no relationship amongst the scholarships and number of no shows. The dots are un-uniform and scattered. Meaning that for every amount of scholarships (by neighbourhood), the number of no-shows vary. The one thing we can see is that lower number of scholarships, there is a higher concentration of data points. So regardless of whether they show up or not, there was more appointments made at the lower number of scholarships.

```
In [96]: x = scholar_no_show.num_scholar
         width = 5

         fig, ax = plt.subplots()
         ax.scatter(x,scholar_no_show.num_showed_up,label='Showed-up')
         ax.scatter(x,scholar_no_show.num_no_show,label='No-show')
         ax.set_ylabel('Attendance')
         ax.set_xlabel('Scholarships')
         figsize=(8, 6)
         ax.legend()
         plt.show()
```



Conclusions

Question 1: Are patients more likely to schedule appointments closer or further than the actual appointment day? Does the length of days between the schedule date and the actual day of the appointment affect whether or not patients will show up to their scheduled appointments?

To conclude question 1, the length of days between the schedule date and the actual day of the appointment does not affect whether the patient shows up or not. Through the analysis above, we see that patients tend to show up more than not, regardless of the gap in days. We also answered the question "are patients more likely to schedule their appointment closer to the day the appointment was set?". This was true. Looking at the second graph, the numbers skewed right while the majority of values were closer to the left, which in the relationship of 'days vs count' means that the majority of the data points were at the shorter amount of days.

Question 2: What correlation does "Age" have on the number of no-shows? does the "Neighbourhood" and amount of scholarships per neighbourhood affet the number of no-shows?

With the "Age" variable, I found that the number of appointments went down as the patient's age increased. In that sense, the number of total appointments made in general had a strong negative correlation. This was achieved by comparing the number of no-shows and showed up by age. I found that both no-shows and showed up had a strong negative correlation in regard to age, so in turn can assume that the total amount of appointments made was negative as well.

With the "Neighbourhood" and "Scholarship" variables, People who received the scholarship were those in need of resources to get healthcare and those who need access to the social welfare program. Exploring the data, I found that Ilha Do Principe had the most scholarships and Nazareth had the least, which could mean that Ilha Do Principe was the poorest neighborhood and Nazareth is the richest, or it could mean that the number of patients within each neighborhood had varying amounts of knowledge about the program. Again, I can only make assumptions based on the context given to us (source Kaggle).

Next to finally answer the question, I compared the number of people who showed up/didn't show up to the number of scholarships given. Using a scatterplot to show the correlation, there was none. I discovered that the data points did not go in a negative or positive direction, which means that number of no-shows did not depend on number of scholarships. With that being said, there is a bigger concentration on the bottom left of the graph. excluding the outliers, it seems like those neighbourhoods that received less scholarships, made more appointments. Those that received more scholarships, didn't make appointments as much.

While comparing different variables to number of no-shows, there is a common trend where people tend to show up more to their appointments than not.

This goes to show that, even with indefinite answers, or answers to the wrong questions: We can still gain useful information by eliminating possible assumptions and build off of that to ask the right ones. > ### Limitation There was one limitation I thought of while working through the project. The person gathering the data could've added a column which contains appointment status. Within these columns, the values could be: "canceled","re-scheduled" and "Done". The canceled and done column would allow the db admin to rid of the information from the database, making analysis and cleaning easier. The Re-scheduled value paints a more accurate picture of what's actually going on with that patient instead of assuming they did not show up and nothing else. Having just the re-scheduled value opens up doors for further analysis. Questions like: 'what is the likely hood of a certain showing up for appointment?'and can be answered based on their previous patterns of no-show and reschedule.

Another column that can be added to the dataset could be "method". This method column contains HOW the patient made the appointment such as: through a previous visit, through e-mail, phone, website.

Questions like: "through which method of making the appointment did patients show up more ?" "which method do patients prefer to use the most?"

questions like these can help the hospitals reach out to their patients better, understand what method of communication to invest in and identify fake-appointments(spam,phishing attempts..etc).