Analysis on How Age Impacts the Attendance of Scheduled Healthcare Appointments

Project Report

Dustin Phan

A.) Summarize the following:

Research question:

What correlation does the patient's age have on the number of no shows after scheduling?

Scope:

The dataset used in this project was one from Kaggle a data sharing platform:

https://www.kaggle.com/datasets/joniarroba/noshowappointments?resource=download

The dataset contains information about general healthcare scheduling. Columns include appointment number, scheduled time, appointment day, age, neighborhood, scholarship, types of diseases, SMS received and no-show. The focus points in this project will be on the age and no-show columns. Referring to the question above, this project will offer an answer to the correlation between a patient's age and number of no-shows. This solution will provide insight for clinics in Brazil by explaining the relationship between a factor that is responsible for the number of patient no-shows/patient showing up. This gives clinics valuable knowledge on how they can rearrange or accommodate those patients who are not effectively able to get to their scheduled appointments. The project does not give solid next steps on what the clinics need to do but gives valuable insight to accomplish their goals. Recommendations will be provided.

Solution:

To effectively solve this question, the following technologies will be used:

- Microsoft Excel
- Jupyter Notebooks

- Python (programming language)
- Python libraries: Pandas, SciPy, Matplotlib, Sklearn and Plotnine. Pandas will be used to import the dataset from Excel csv to Jupyter Notebooks environment and manipulate the data as needed. SciPy is used on the data frames created in Python/Pandas to calculate statistical information. Matplotlib/SkLearn will be used when I visualize the data and apply linear regression. Plotnine was used to effectively visualize the linear regression data.
- Methodologies: Linear regression is used in this project because we are comparing a predictor variable and a response variable. The predictor is the Age, the response variable is the count or number of patients that fall into a category (no-show/showed-up). This will be shown during the test phase, to see if the general comparison (the first plot with both Age vs no-show/showed-up), aligns with the plots that the linear regression was performed, adding a best fit line.

B.) Project Plan

Project planning methodology:

During the planning of this project, it still used the waterfall methodology. To summarize:

- Requirements gathering: Compared to Task 2, I've used the same dataset that is listed in the appendices. Microsoft Excel, Python and Jupyter Notebooks was used. What I did not anticipate in part 2 that I've implemented in the project was a couple of libraries. The first one was SciPy to do statistical analysis on the data frames, the second was SkLearn to perform linear regression, and lastly, plotnine (ggplot) to visually inspect the linear regression data.

- Design: In comparison to task 2, for the design phase I did this by first importing the dataset into pandas and inspected information on my target columns (‘Age’ and ‘No-show’). I was looking for datatypes, missing values and duplicate rows. After this was completed, I then split the clean dataset into two separate data frames. The next steps are in the implementation stage of this method.

- Implementation: The original data frame was first split into a separate data frame after being cleaned consisting of only the ‘Age’ and ‘No-show’ columns, What I realized at this point and didn’t in task 2 was I needed to split this data frame even further into two separate data frames. Data frame 1 consisting of (Age and patients that showed up) and data frame 2 consisting of (Age and patients that did NOT show up). The next step was to visualize on one plot what the relationship between age and no-show columns were. To get a clearer picture, statistical analysis was done by applying a function from the stats library in SciPy. The output of utilizing this function gave me the slope, p-value and R-value of the data that was plotted.

- Testing: In the original plan, I thought of comparing different factors like neighborhood or diseases in correlation with the number of no-shows, to show that this solution worked for different factors. While going through the actual project, it only made sense to back up the question at hand, instead of getting different analyses for different factors. Instead, I decided to take those individual data frames and run further analyses on them. The method I used was linear regression. For the dataset comparing age to number of patients that showed up, I used a prediction model where a best fit line was created, then an r2 score value was calculated to get the percentage of variance that the age variable explains (in regard to patients that showed up). The same method was applied to the second data frame (Age vs patients that did NOT show up). Visualizations were created including the best fit line on the graph to solidify my findings.

- Delivery/Deployment: In the tentative proposal, I stated that I would give a presentation on my findings. The actual project will include a Panopto recording giving a presentation on the code and the visualizations. The project report will be included in the delivery as well.

Project timeline and milestones:

The actual timeline of this project was vastly different and ahead of schedule. In the tentative project proposal, I was unsure of what steps I will complete since I had to account for submission of tasks and review/approval of submissions. The actual timeline is as follows:

Milestone	**Projected Start Date**	**Projected End Date**	**Duration (days/hours)**
Requirements gathering	06/22/23	06/22/23	1 day
High-Level Design	06/23/23	06/23/23	1 day
Implementation	06/24/23	06/25/23	2 days
Testing	06/26/23	06/26/23	1 day
Delivery Deployment	06/27/23	06/28/23	2 days

C.) Methodology (data selection and collection)

Data selection/ collection process:

The data selection was the same as task 1. While prepping for this final project, I was already working on the data selection process. To recap the dataset below was used:

https://www.kaggle.com/datasets/joniarroba/noshowappointments?resource=download

This is the general dataset consisting of health care appointment schedule information. The collection method was simply to download that csv file and read it in using Pandas library from Python into Jupyter Notebooks.

Obstacles:

The obstacle I faced was determining what the data actually consisted of. This required navigating through the Kaggle link and finding meta information/column information on what they meant. I had to be careful of the columns that were required to answer my question carefully.

Handling unplanned data governance issues:

From task 2, I knew the data had to be free of null/missing values and duplicate row information. I did not anticipate the data being 'dirty' based off of the column description. For example, in this dataset there was a patient's age of the value -1. This is not feasible since people cannot be a negative age.

Advantages and limitations of the dataset used:

The advantage of using this dataset was that it was really clean and did not have too many columns, the data had no Na/missing values and no duplicate rows. I made sure to verify in Jupyter notebooks by running different Pandas code. That being said, this is where the data set's limitations lay as well. While analyzing the data, I thought a very beneficial added column could've been a 're-scheduled' column. This would be where the dataset identifies which appointment number had a reschedule and could've produced completely different data or means of interpretating the data.

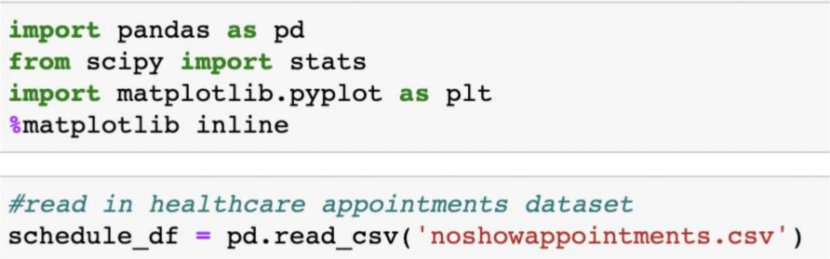
D) Data extraction and preparation (full code can be found in the PDF file in the appendices)

These steps are the exact steps I took for data extraction and preparation:

1.) Downloaded the dataset from this link: https://www.kaggle.com/datasets/joniarroba/noshowappointments?resource=download

2.) Imported the csv file into Jupyter Notebooks

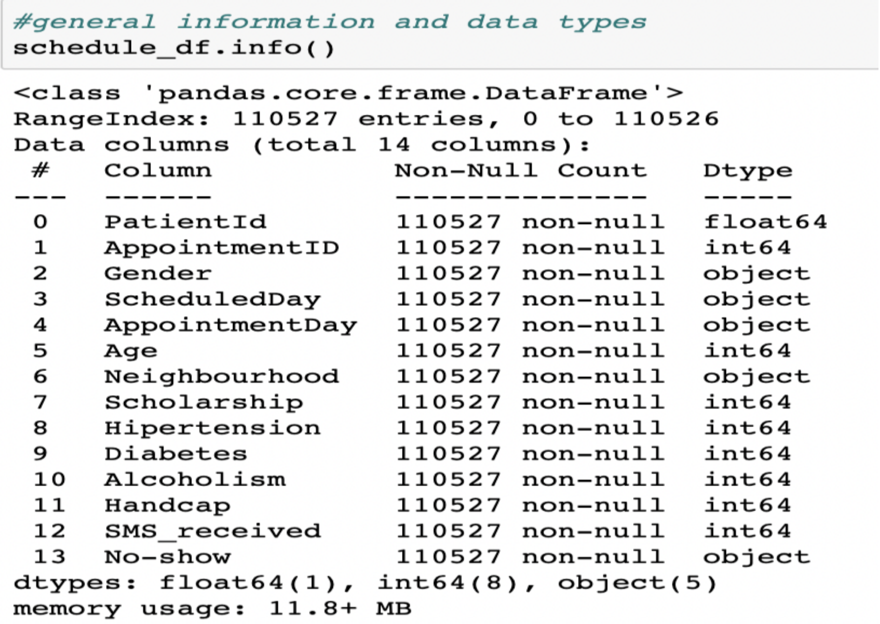
3.) Assigned the dataset into a data frame named 'schedule_df '.



```
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
%matplotlib inline
```

```
#read in healthcare appointments dataset
schedule_df = pd.read_csv('noshowappointments.csv')
```

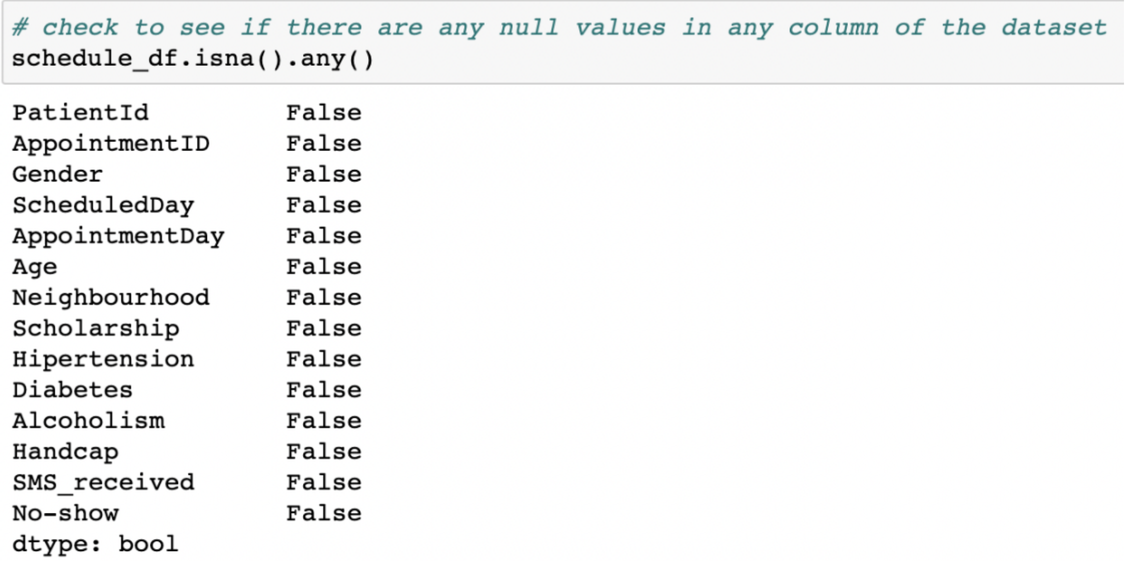
4.) Displayed information on columns, number of rows and datatypes.



```
#general information and data types
schedule_df.info()
```

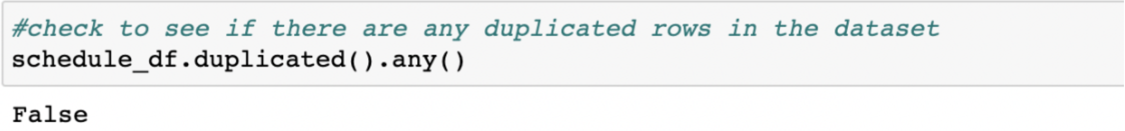
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   PatientId       110527 non-null  float64
 1   AppointmentID   110527 non-null  int64  
 2   Gender          110527 non-null  object 
 3   ScheduledDay    110527 non-null  object 
 4   AppointmentDay  110527 non-null  object 
 5   Age             110527 non-null  int64  
 6   Neighbourhood   110527 non-null  object 
 7   Scholarship     110527 non-null  int64  
 8   Hipertension    110527 non-null  int64  
 9   Diabetes        110527 non-null  int64  
 10  Alcoholism      110527 non-null  int64  
 11  Handcap         110527 non-null  int64  
 12  SMS_received    110527 non-null  int64  
 13  No-show         110527 non-null  object 
dtypes: float64(1), int64(8), object(5)
memory usage: 11.8+ MB
```

5.) Check for null values and duplicates.



```
# check to see if there are any null values in any column of the dataset
schedule_df.isna().any()
```

```
PatientId         False
AppointmentID     False
Gender            False
ScheduledDay      False
AppointmentDay    False
Age               False
Neighbourhood     False
Scholarship       False
Hipertension      False
Diabetes          False
Alcoholism        False
Handcap           False
SMS_received      False
No-show           False
dtype: bool
```



```
#check to see if there are any duplicated rows in the dataset
schedule_df.duplicated().any()
```

```
False
```

The tools (Python and Jupyter Notebooks) and methods (Pandas functions) shown in these snapshots were appropriate to read the dataset into an environment which is easy to manipulate the data. In the snapshots were examples on how I was able to verify the cleanliness of the data.

E) Data analysis Report (Full code in the appendices)

Methods used to analyze the data:

The methods that were used in this project were:

Data cleaning. Looking for data that can throw off the final analysis i.e., missing values, duplicate values and column values that don’t make sense such as ‘Age’ being ‘-1’ and making sure that datatypes are matching the column description.

Data wrangling. Separating columns in the dataset that are appropriate for the question that the analysis is trying to answer. In this case the ‘Age’ and ‘No-show’ columns.

Linear regression. This is the main point of the analysis and is what answers the question of the analysis. Linear regression is effective in comparing two variables and seeing the relationship/correlation between the variables.

Advantages and limitations of the tools used to analyze the data:

Data cleaning advantage: This is the first and a very important step in data analysis, without cleaning the data the end result will more than likely be inaccurate.

Data cleaning disadvantage: This is time consuming and sometimes the analyst can remove data that was deemed unimportant but in reality, is important.

Data wrangling advantage: the second step to the data analysis process is wrangling the data. This step narrows down the data, making answering questions more effective.

Data wrangling disadvantage: This is a tricky step because it requires the analyst to study the data and get to know the relationships between the variables. It can also be an iterative step because sometimes after the end result, the analyst might go back and add in or remove columns to more effectively answer the analysis questions.

Linear regression advantage: In this analysis using linear regression is incredibly effective since linear regression's purpose is to compare the relationship/correlation between two variables. Furthermore, a best fit line can be calculated on the data to predict values. For example, in this case, predict whether the patient will show up to their scheduled appointments based on age. (This will be shown in the step-by-step explanation).

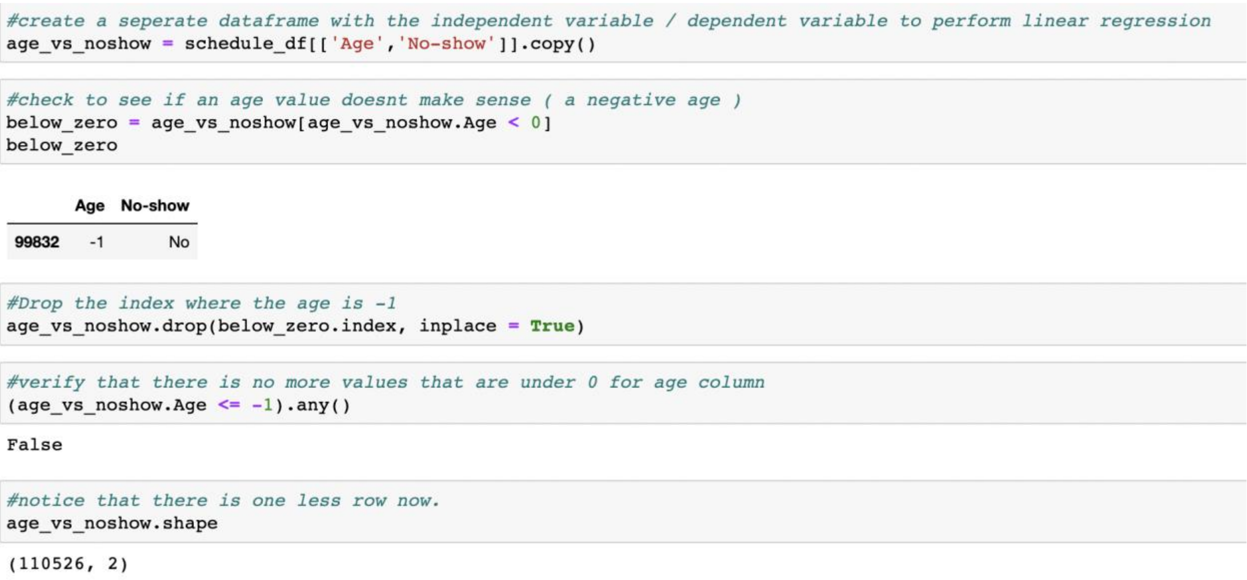
Linear regression disadvantage: The disadvantage of using linear regression is in the name itself. Sometimes data will not always have a linear correlation. In other words, it assumes that the two variables have a linear relationship. That means by using this method, the data is prone to noise due to any clustering data or outlier data.

Step-by-step explanation of analytical methods:

A step-by-step explanation including snapshots of the analytical methods used in this project will be presented in the following pages.

Continuing on from the cleaned dataset…

Step1.) look for any other data issues. In this dataset I found an issue where the value -1 did not make sense for the column ‘Age’ since patients cannot be a negative age. Once found, I dropped that row.



```
#create a seperate dataframe with the independent variable / dependent variable to perform linear regression
age_vs_noshow = schedule_df[['Age','No-show']].copy()
```

```
#check to see if an age value doesnt make sense ( a negative age )
below_zero = age_vs_noshow[age_vs_noshow.Age < 0]
below_zero
```

	Age	No-show
99832	-1	No

```
#Drop the index where the age is -1
age_vs_noshow.drop(below_zero.index, inplace = True)
```

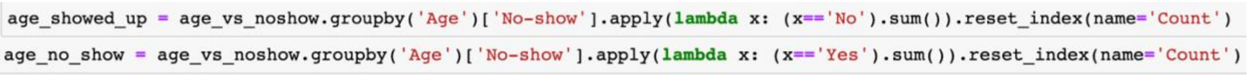
```
#verify that there is no more values that are under 0 for age column
(age_vs_noshow.Age <= -1).any()
```

```
False
```

```
#notice that there is one less row now.
age_vs_noshow.shape
```

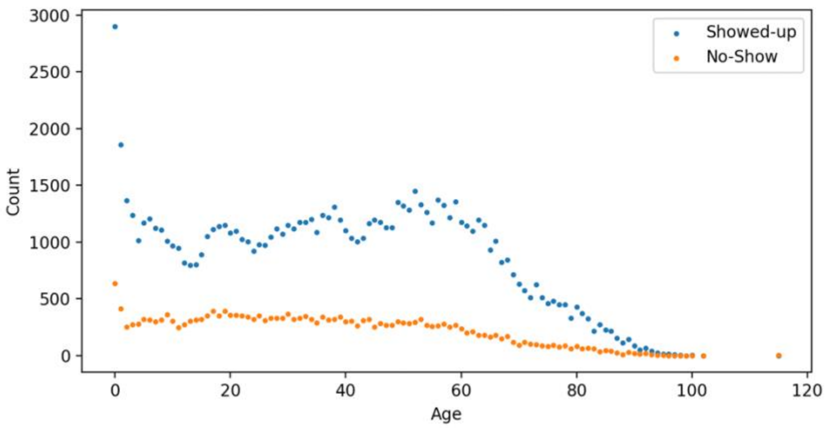
```
(110526, 2)
```

Step2.) Two data frames were created from the age_vs_noshow data frame, to perform linear regression. One called ‘age_showed_up’ and one called ‘age_no_show’. Representing (Age vs number of patients that showed up and Age vs number of patients that did NOT show up respectively). This was created by using grouping/lambda function to group the ages into categories by year (0,1,2,3,4,5… years old) and applying the lambda function on each value in the no-show dataset to get a count on how many no-shows/patients that showed up in each age category. The plot visual gives a general relationship.

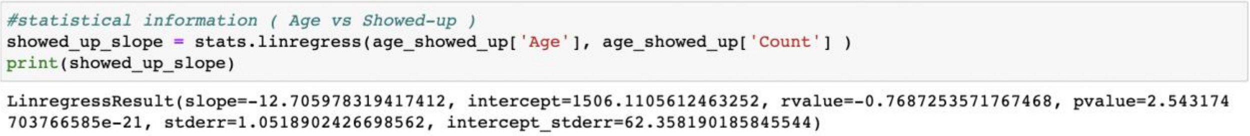


```
age_showed_up = age_vs_noshow.groupby('Age')['No-show'].apply(lambda x: (x=='No').sum()).reset_index(name='Count')
```

```
age_no_show = age_vs_noshow.groupby('Age')['No-show'].apply(lambda x: (x=='Yes').sum()).reset_index(name='Count')
```

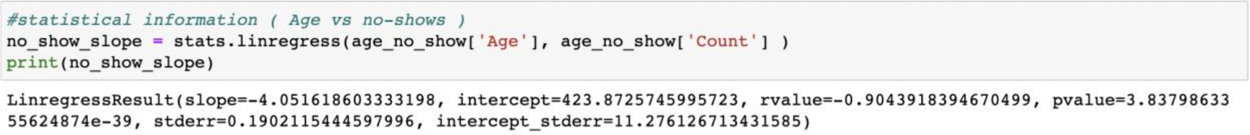


Step 3.) From step 2, I noticed that there was a negative relationship between the variables, so to get statistical information, the SciPy library was used to obtain the slope, p-value and R-value to confirm my findings.



```
#statistical information ( Age vs Showed-up )
showed_up_slope = stats.linregress(age_showed_up['Age'], age_showed_up['Count'] )
print(showed_up_slope)
```

```
LinregressResult(slope=-12.705978319417412, intercept=1506.1105612463252, rvalue=-0.7687253571767468, pvalue=2.543174
703766585e-21, stderr=1.0518902426698562, intercept_stderr=62.358190185845544)
```

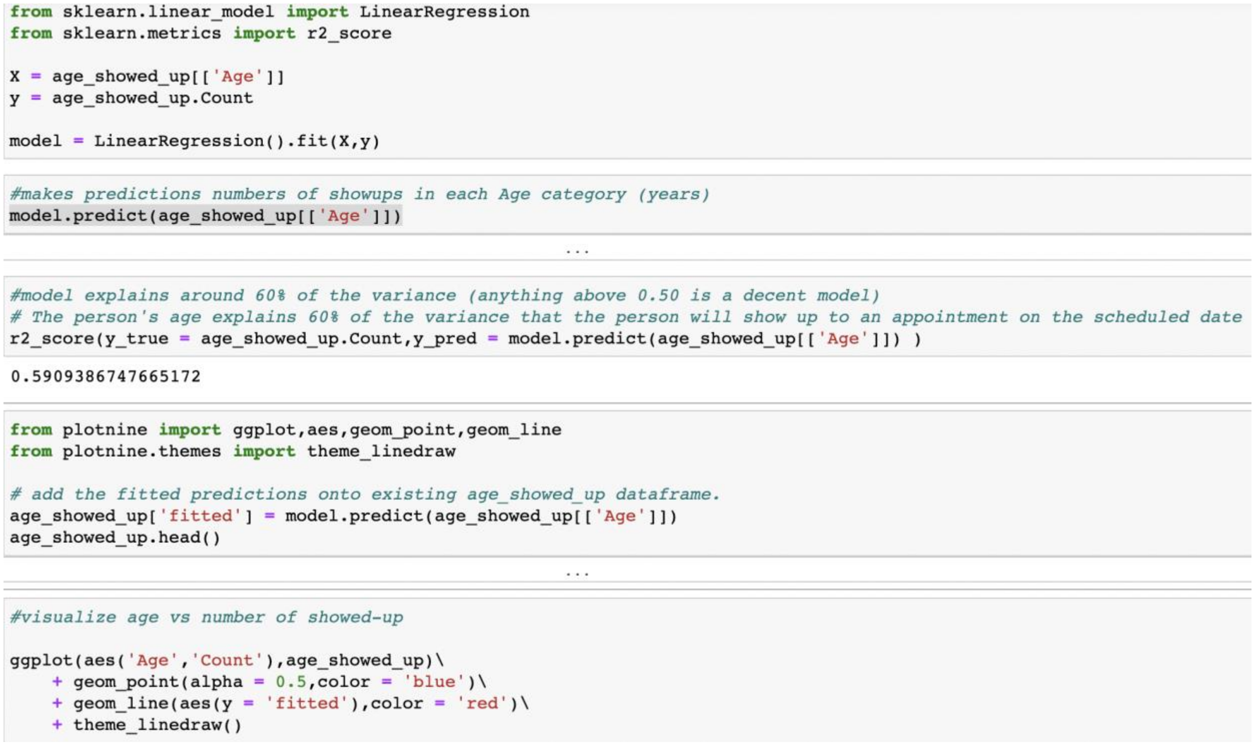


```
#statistical information ( Age vs no-shows )
no_show_slope = stats.linregress(age_no_show['Age'], age_no_show['Count'] )
print(no_show_slope)
```

```
LinregressResult(slope=-4.051618603333198, intercept=423.8725745995723, rvalue=-0.9043918394670499, pvalue=3.83798633
55624874e-39, stderr=0.1902115444597996, intercept_stderr=11.276126713431585)
```

As shown above the slope and R-value are negative. This represents a negative linear relationship. A negative slope represents a negative correlation. (As the patient gets older, the count of showing up is lower). The R-value represents the coefficient which tells me how strong the correlation is. (Scaling from -1 strong negative correlation and 1 being a strong positive correlation). The p-value measures significance. The threshold for P-value being significant is: 0.05, this means that there was a 5% that the results happened by chance. If our results show anything with 0.05% or under that means our results has a correlation with statistical significance. In other words, the two variables being compared has a linear relationship.

The last step before concluding.) the last step was to confirm my findings and do some extra tests to see if they match up. Therefore, concluding the methodology part of this project. The same coding method was used on the age_no_show data frame as well.



```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

X = age_showed_up[['Age']]
y = age_showed_up.Count

model = LinearRegression().fit(X,y)
```

```
#makes predictions numbers of showups in each Age category (years)
model.predict(age_showed_up[['Age']])
```

...

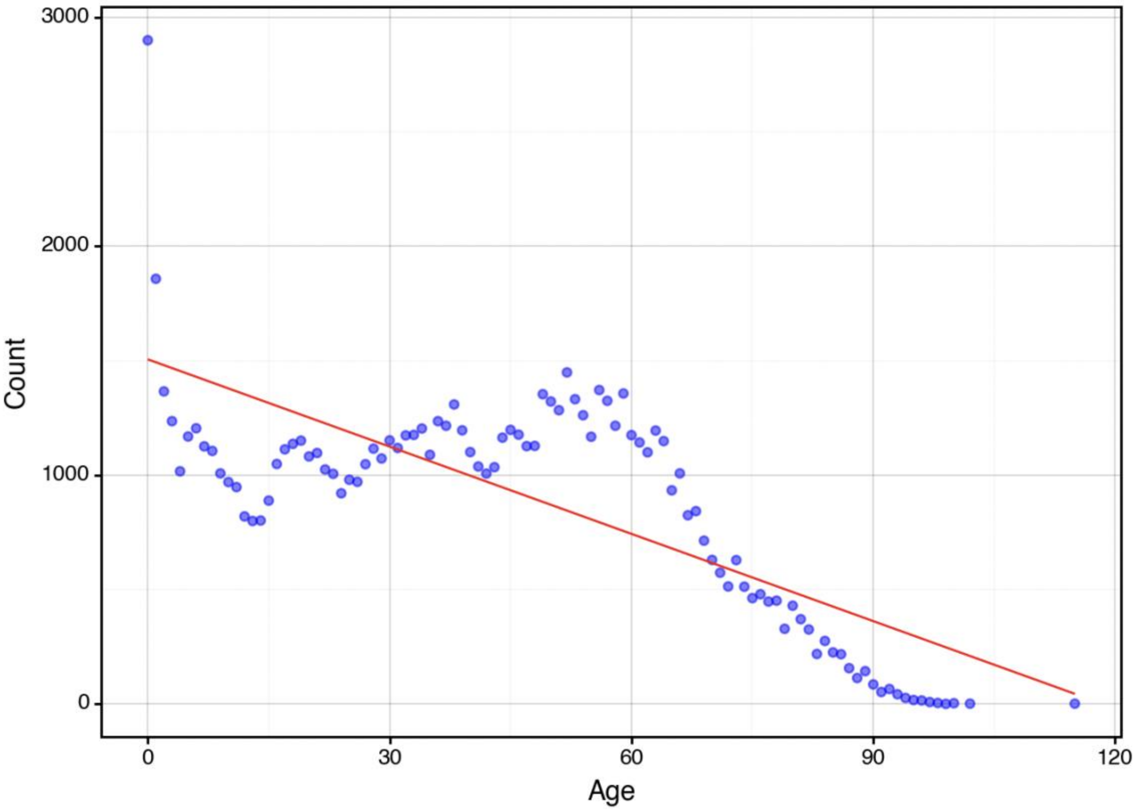
```
#model explains around 60% of the variance (anything above 0.50 is a decent model)
# The person's age explains 60% of the variance that the person will show up to an appointment on the scheduled date
r2_score(y_true = age_showed_up.Count,y_pred = model.predict(age_showed_up[['Age']]) )
```

```
0.5909386747665172
```

```
from plotnine import ggplot,aes,geom_point,geom_line
from plotnine.themes import theme_linedraw

# add the fitted predictions onto existing age_showed_up dataframe.
age_showed_up['fitted'] = model.predict(age_showed_up[['Age']])
age_showed_up.head()
```

...

```
#visualize age vs number of showed-up

ggplot(aes('Age','Count'),age_showed_up)\
    + geom_point(alpha = 0.5,color = 'blue')\
    + geom_line(aes(y = 'fitted'),color = 'red')\
    + theme_linedraw()
```



The code in the last page was used to perform linear regression on the previously built dataframes. ‘age_no_show’ and ‘age_showed_up’.

As a reminder the full code is provided in the appendices and will be presented in the Panopto presentation.

Using the linear regression function from the Sklearn library, I created a best fit line by assigning variable X with ‘Age’ and y with ‘count’. This line shows a relationship between the two variables. Not only that, but it gives a prediction on how many people of that age category will either show up or not show up.

The r2 score that can be pulled from the output represents:

The percentage of variance that the model explains in regard to the compared variables. In the practical world, a model with a r2 score higher than 0.5 is a decent model. As shown from the statistical calculations above, the age_showed_up dataframe gave me a 0.59 r2 score, this means that the person's age explains 60% of the variance that the person will show up to an appointment on the scheduled date.

F.) Evaluation of the project

Statistical significance:

Definitions:

- P-value measures the statistical significance. This value confirms that the result of the regression performed on the sample dataset was not by chance. the threshold for P-value being significant is: 0.05, this means that there was a 5% that the results happened by chance. If our results show anything with 0.05% or under that means our results has a correlation with statistical significance. In the two variables being compared has a linear relationship.

- R-value: is the correlation coefficient which ranges from -1 to 1, the close it is to -1 the stronger the negative relationship between the age and no-show /show-up variables. Likewise, with 1 it shows a strong positive correlation.

- Slope: Determines whether the relationship is a negative or positive relationship

- R2 score is a statistical measure that shows the percentage of variance for a dependent variable that’s explained by an independent variable a linear regression model. In practice, an R2 score of above 50% (0.5) is an acceptable model.

The results of my analysis:

Age_no_show:

- P-Value: 2.54e-21. This number is very close to 0. This indicates that the result of regression was not by chance and is statistically significant.
- R-Value: 0.904 Age and no-show have very strong negative correlation.
- Slope: - 4.1 confirms that it is a negative linear relationship between the two variables
- R2 score: 0.818. This shows that 81% of the variance in count of no-shows is explained by the age variable.

Age_showed_up:

- P-Value: 3.84e-39 which is. This number is also very close to 0. This number is very close to 0. This indicates that the result of regression was not by chance and is statistically significant.
- R-Value: - 0.769 Age and patients that show up have a negative correlation.
- Slope: -12.71 confirms that this is also a negative linear relationship between the two variables.
- R2 score: 0.591. This shows that 59% of the variance in count of showed up is explained by the age variable.

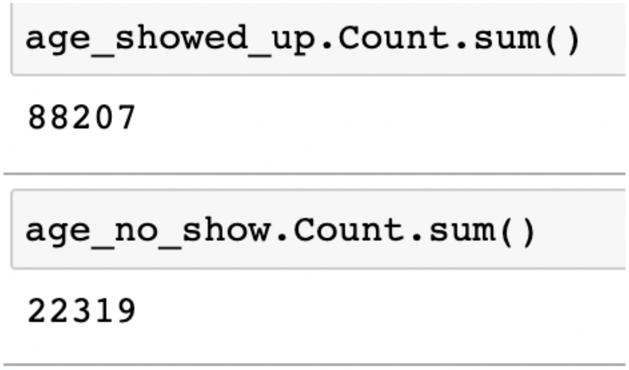
Practical significance:

From my findings, the data shows that comparing these two variables are significant. This is useful in the real world because based on this dataset, clinics can draw conclusions that age does have an effect on whether or not patients will show up to their scheduled appointments, whatever the reasons may be. The result of this analysis shows that as the 'Age' gets higher, the smaller number of patients showing up to their scheduled appointments lessen. The tricky part in this analysis is that the results also show that as 'Age' gets higher, the smaller number of patients NOT showing up to their appointments lessen as well. (meaning that they do show up to their appointments).

So, this means that according to this dataset, the number of people that show up or don't show up both lessen as age gets higher. (fewer elderly patients in general have appointments to even go to). In order for clinics to ensure effective care, the patients must have a way to make it to their appointments. The solution would be to target elderly people and figure out ways to reach out and provide the assistance as needed.

Overall success and effectiveness:

The results show that in both the number of patients that show up and not showing up in respect to age have a negative relationship. This analysis was effective in showing the relationships between age and the no-show columns. We have to keep in mind in this limited dataset, the total number of people who showed up is around 4x higher than the number of people who did not show up. Looking at the numbers below and taking a look at the slope for each of these data frames, we see that the number of age_showed_up (-12.71 compared to 4.1) is a much more negative slope. Meaning the number of patients that show-up as age grows drops significantly faster than patients that don't show up as age grows. A clear explanation of the interoperation is explained in the summary.



```
age_showed_up.Count.sum()
```

```
88207
```

```
age_no_show.Count.sum()
```

```
22319
```

G.) Summary of Analysis

Conclusions drawn:

- Age vs number of show ups have a negative linear relationship (less showed up with higher age)

- Age vs number of no-shows have a negative linear relationship (more showed up with higher age)

- This analysis can be tricky to interoperate, if less people showed up with higher age 'age_showed_up', and more people showed up with higher age when comparing 'age_no_show' dataframe, we have to look at it in separate perspectives. Around 88,207 people total showed up and 22,319 people did not show up. If the slope for the total people who showed up shows more negative, that means that the number of showed up drops significantly faster as the age increases. If the total number is much larger, and the slope drops quicker, that means the quantity of patients that are showing up drops a lot faster, as a result people are less likely to show up in comparison.
  This means we can conclude that as age increases, the less likely patients are going to show up for their scheduled appointments.

Effectiveness of graphical communication:

Graphical communication can vary depending on what the analyst chooses to graph. In this analysis it was pretty clear on what I needed to visualize. Again, to compare the 'Age' to 'No-show' columns. A scatter plot is commonly used to identify linear relationships and more so, correlations between variables. So that is why the scatter plot was effective in this analysis. During linear regression, a best fit line is used for prediction. The best fit line is the chosen tool for regression models because it measures the distance between the predicted values, and the actual values. The closer the values are to the line, the stronger the correlation.

2 Recommended courses of action:

Recommendation 1: Clinics in Brazil should reach out to their patients in their current databases, identify who is scheduled for an appointment and who's not. If time permits, find a way of contacting those patients who are elderly or not able to schedule an appointment and work with them to accommodate to their needs.

Recommendation 2: Use this analysis and with any new incoming patients, use the prediction model to infer if they will show up to their scheduled appointments or not. If they are more likely to not show up. Provide options up front to accommodate to their needs, resulting in a higher count of patients that show up to their schedules appointments. Allowing for better workflow, better work conditions and a stronger health scheduling system overall.

H.) Panopto Recording Link

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3d7a6897-86e7-4a6f-87be-b030006ec860#

I) Appendices / Sources

Dataset: https://www.kaggle.com/datasets/joniarroba/noshowappointments?resource=download

Tools :
- https://anaconda.org/anaconda/jupyter
- https://www.python.org/downloads/
- https://plotnine.readthedocs.io/en/stable/ (syntax for plotting using ggplot)

Code

In [297]:

```
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
%matplotlib inline
```

In [298]:

```
#read in healthcare appointments dataset
schedule_df = pd.read_csv('noshowappointments.csv')
```

In [299]:

```
#snapshot of columns and data in dataset
schedule_df.head()
```

Out[299]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	0	1	0	0	0	0	No
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	0	0	0	0	0	No
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	0	0	0	0	0	0	No
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	0	0	0	0	0	0	No
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	1	1	0	0	0	No

In [300]:

```
#general information and data types
schedule_df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   PatientId       110527 non-null  float64
 1   AppointmentID   110527 non-null  int64  
 2   Gender          110527 non-null  object 
 3   ScheduledDay    110527 non-null  object 
 4   AppointmentDay  110527 non-null  object 
 5   Age             110527 non-null  int64  
 6   Neighbourhood   110527 non-null  object 
 7   Scholarship     110527 non-null  int64  
 8   Hipertension    110527 non-null  int64  
 9   Diabetes        110527 non-null  int64  
 10  Alcoholism      110527 non-null  int64
 11  Handcap         110527 non-null  int64
 12  SMS_received    110527 non-null  int64
 13  No-show         110527 non-null  object
dtypes: float64(1), int64(8), object(5)
memory usage: 11.8+ MB
```

In [301]:

```
# check to see if there are any null values in any column of the dataset
schedule_df.isna().any()
```

Out[301]:

```
PatientId         False
AppointmentID     False
Gender            False
ScheduledDay      False
AppointmentDay    False
Age               False
Neighbourhood     False
Scholarship       False
Hipertension      False
Diabetes          False
Alcoholism        False
Handcap           False
SMS_received      False
No-show           False
dtype: bool
```

In [302]:

```
#check to see if there are any duplicated rows in the dataset
schedule_df.duplicated().any()
```

Out[302]:

```
False
```

In [303]:

```
#create a seperate dataframe with the independent variable / dependent
variable to perform linear regression
age_vs_noshow = schedule_df[['Age','No-show']].copy()
```

In [304]:

```
#check to see if an age value doesnt make sense ( a negative age )
below_zero = age_vs_noshow[age_vs_noshow.Age < 0]
below_zero
```

Out[304]:

	Age	No-show
99832	-1	No

In [305]:

```
#Drop the index where the age is -1
age_vs_noshow.drop(below_zero.index, inplace = True)
```

In [306]:

```
#verify that there is no more values that are under 0 for age column
```

```
(age_vs_noshow.Age <= -1).any()
```

Out[306]:

```
False
```

In [307]:

```
#notice that there is one less row now.
age_vs_noshow.shape
```

Out[307]:

```
(110526, 2)
```

In [308]:

```
age_vs_noshow.head()
```

Out[308]:

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

below data seperates the Age and counts of people who showed up and who did not show up

"age_showed_up" = patients categorized by age who SHOWED up.

"age_no_show" = patients categorized by age who did NOT show up.

In [309]:

```
age_showed_up = age_vs_noshow.groupby('Age')['No-show'].apply(lambda x:
(x=='No').sum()).reset_index(name='Count')
```

In [310]:

```
age_showed_up.head()
```

Out[310]:

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

In [311]:

```
age_no_show = age_vs_noshow.groupby('Age')['No-show'].apply(lambda x:
(x=='Yes').sum()).reset_index(name='Count')
```

In [312]:

```
age_no_show.head()
```

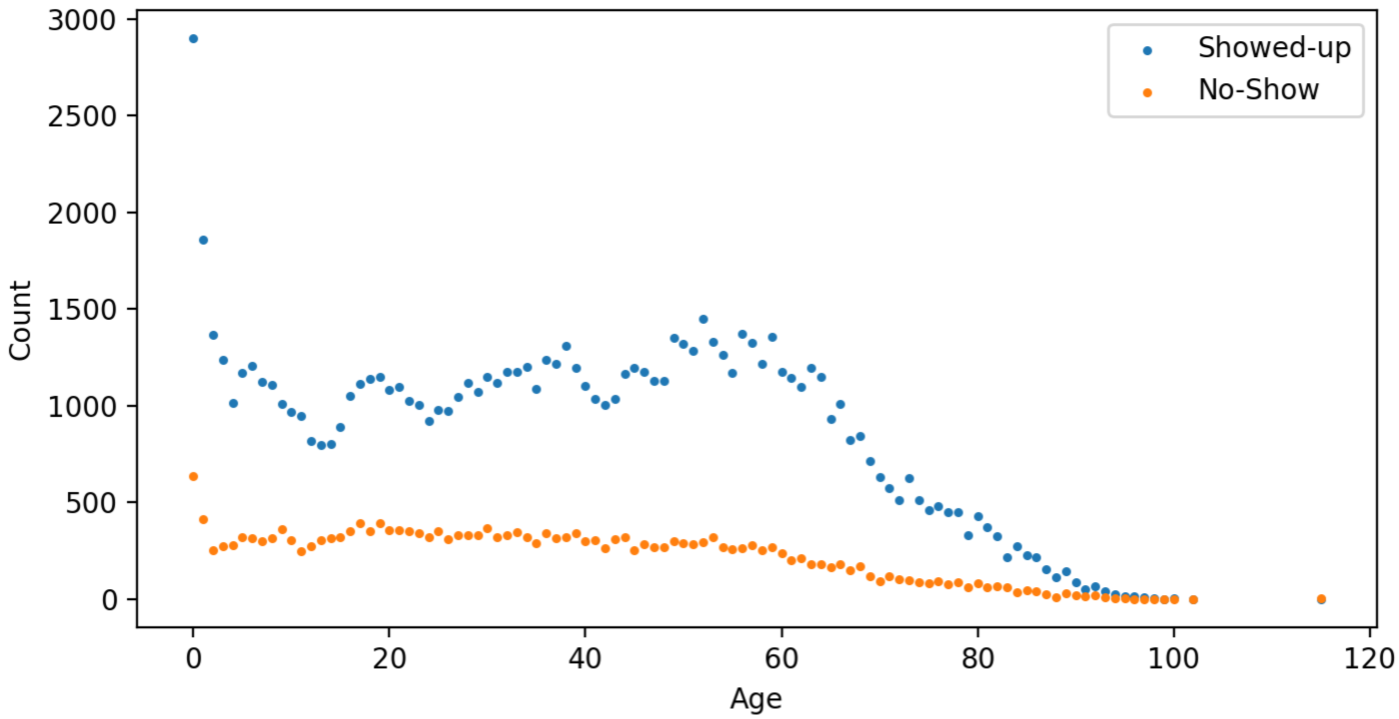
Out[312]:

	Age	Count
0	0	639
1	1	415
2	2	252

	Age	Count
3	3	277
4	4	282

In [313]:

```
w = 5
fig, ax = plt.subplots(figsize = (8,4))
ax.scatter(age_no_show.Age,age_showed_up.Count,w,label = 'Showed-up')
ax.scatter(age_no_show.Age,age_no_show.Count,w,label = 'No-Show')
ax.set_ylabel('Count')
ax.set_xlabel('Age')
ax.legend()
plt.show()
;
```



Out[313]:

```
''
```

Next two blocks of code shows the slope of patients that showed up and patients that did not show up according to age.

Slope: shows a general correlation

R-value: is the correlation coefficient which ranges from -1 to 1, the close it is to -1 the stronger the negative relationship between the age and no-show /show-up variables. Likewise, with 1 it shows a strong positive correlation.

P-value: P-value confirms that the result of the regression performed on the sample dataset was not by chance. the threshold for P-value being significant is: 0.05, this means that there was a 5% that the results happened by chance. If our results show anything with 0.05% or under that means our results has a correlation with statistical significance.In other words, the two variables being compared has a linear relationship.

In [314]:

```
#statistical information ( Age vs Showed-up )
showed_up_slope = stats.linregress(age_showed_up['Age'],
age_showed_up['Count'] )
print(showed_up_slope)
LinregressResult(slope=-12.705978319417412, intercept=1506.1105612463252,
rvalue=-0.7687253571767468, pvalue=2.543174703766585e-21,
stderr=1.0518902426698562, intercept_stderr=62.358190185845544)
```

In [315]:

```
#statistical information ( Age vs no-shows )
no_show_slope = stats.linregress(age_no_show['Age'], age_no_show['Count'] )
print(no_show_slope)
LinregressResult(slope=-4.051618603333198, intercept=423.8725745995723,
rvalue=-0.9043918394670499, pvalue=3.8379863355624874e-39,
stderr=0.1902115444597996, intercept_stderr=11.276126713431585)
```

In [316]:

```
# Testing ( Age vs Showed-up)
# purpose: fitting the line that minimizes the distance of the actual scores
from the predicted scores.
# Use fitted regression lines to illustrate the relationship between a
predictor variable (x) and
# a response variable (y)
# in this case, this is used to predict whether a person will show up based
on age.

from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

X = age_showed_up[['Age']]
y = age_showed_up.Count

model = LinearRegression().fit(X,y)
```

In [317]:

```
#makes predictions numbers of showups in each Age category (years)
model.predict(age_showed_up[['Age']])
```

Out[317]:

```
array([1506.11056125, 1493.40458293, 1480.69860461, 1467.99262629,
       1455.28664797, 1442.58066965, 1429.87469133, 1417.16871301,
       1404.46273469, 1391.75675637, 1379.05077805, 1366.34479973,
       1353.63882141, 1340.93284309, 1328.22686477, 1315.52088646,
       1302.81490814, 1290.10892982, 1277.4029515 , 1264.69697318,
       1251.99099486, 1239.28501654, 1226.57903822, 1213.8730599 ,
       1201.16708158, 1188.46110326, 1175.75512494, 1163.04914662,
       1150.3431683 , 1137.63718998, 1124.93121166, 1112.22523334,
       1099.51925502, 1086.81327671, 1074.10729839, 1061.40132007,
       1048.69534175, 1035.98936343, 1023.28338511, 1010.57740679,
        997.87142847,  985.16545015,  972.45947183,  959.75349351,
        947.04751519,  934.34153687,  921.63555855,  908.92958023,
        896.22360191,  883.51762359,  870.81164528,  858.10566696,
        845.39968864,  832.69371032,  819.987732  ,  807.28175368,
        794.57577536,  781.86979704,  769.16381872,  756.4578404 ,
        743.75186208,  731.04588376,  718.33990544,  705.63392712,
        692.9279488 ,  680.22197048,  667.51599216,  654.81001385,
        642.10403553,  629.39805721,  616.69207889,  603.98610057,
        591.28012225,  578.57414393,  565.86816561,  553.16218729,
        540.45620897,  527.75023065,  515.04425233,  502.33827401,
        489.63229569,  476.92631737,  464.22033905,  451.51436073,
        438.80838242,  426.1024041 ,  413.39642578,  400.69044746,
        387.98446914,  375.27849082,  362.5725125 ,  349.86653418,
        337.16055586,  324.45457754,  311.74859922,  299.0426209 ,
        286.33664258,  273.63066426,  260.92468594,  248.21870762,
        235.5127293 ,  210.10077267,   44.92305451])
```

In [318]:

```
#model explains around 60% of the variance (anything above 0.50 is a decent model)
# The person's age explains 60% of the variance that the person will show up to an appointment on the scheduled date
r2_score(y_true = age_showed_up.Count,y_pred = model.predict(age_showed_up[['Age']]) )
```

Out[318]:

```
0.5909386747665172
```

In [319]:

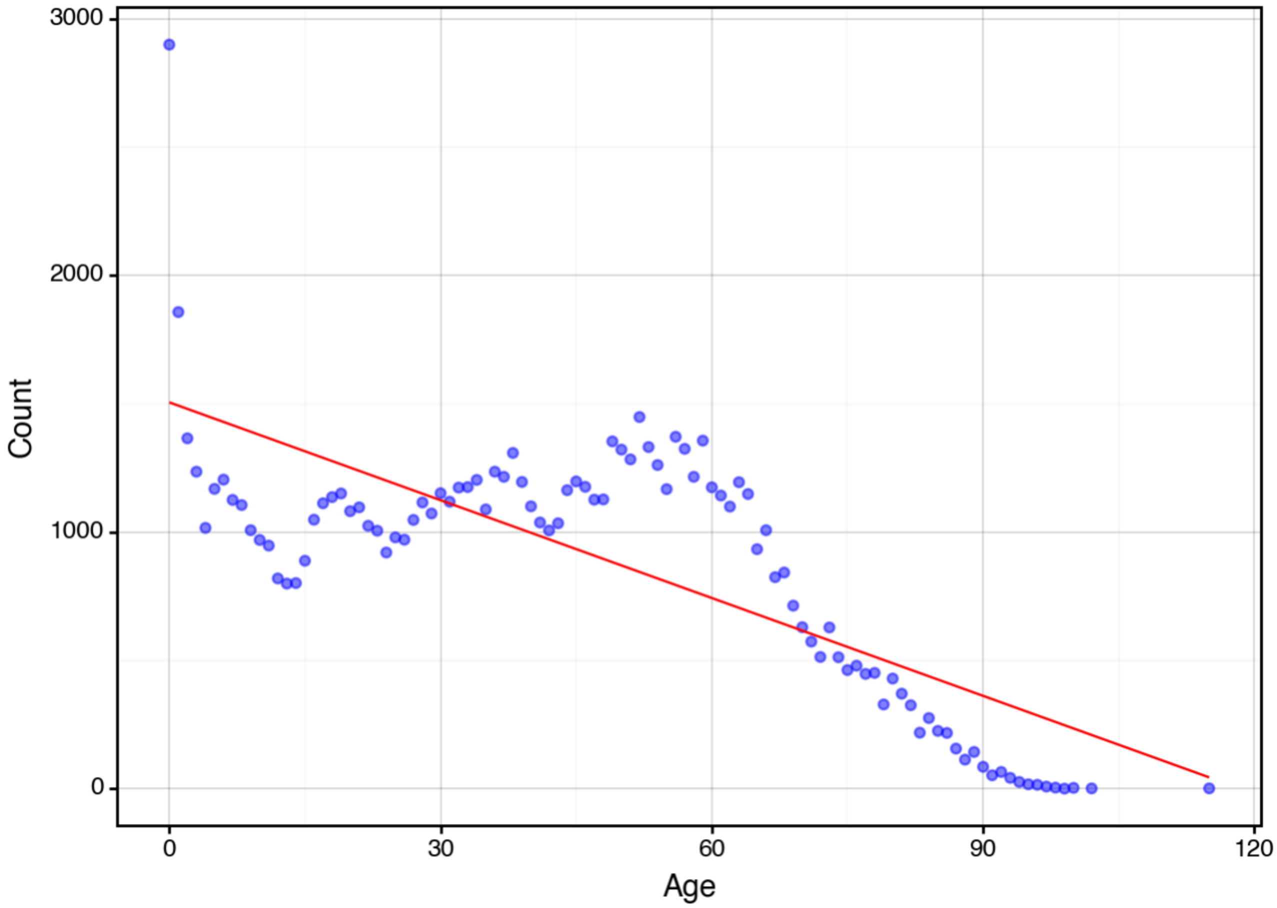
```
from plotnine import ggplot,aes,geom_point,geom_line
from plotnine.themes import theme_linedraw

# add the fitted predictions onto existing age_showed_up dataframe.
age_showed_up['fitted'] = model.predict(age_showed_up[['Age']])
age_showed_up.head()
```

Out[319]:

	Age	Count	fitted
0	0	2900	1506.110561
1	1	1858	1493.404583
2	2	1366	1480.698605
3	3	1236	1467.992626
4	4	1017	1455.286648

In [320]:

```
#visualize age vs number of showed-up

ggplot(aes('Age','Count'),age_showed_up)\
    + geom_point(alpha = 0.5,color = 'blue')\
    + geom_line(aes(y = 'fitted'),color = 'red')\
    + theme_linedraw()
```



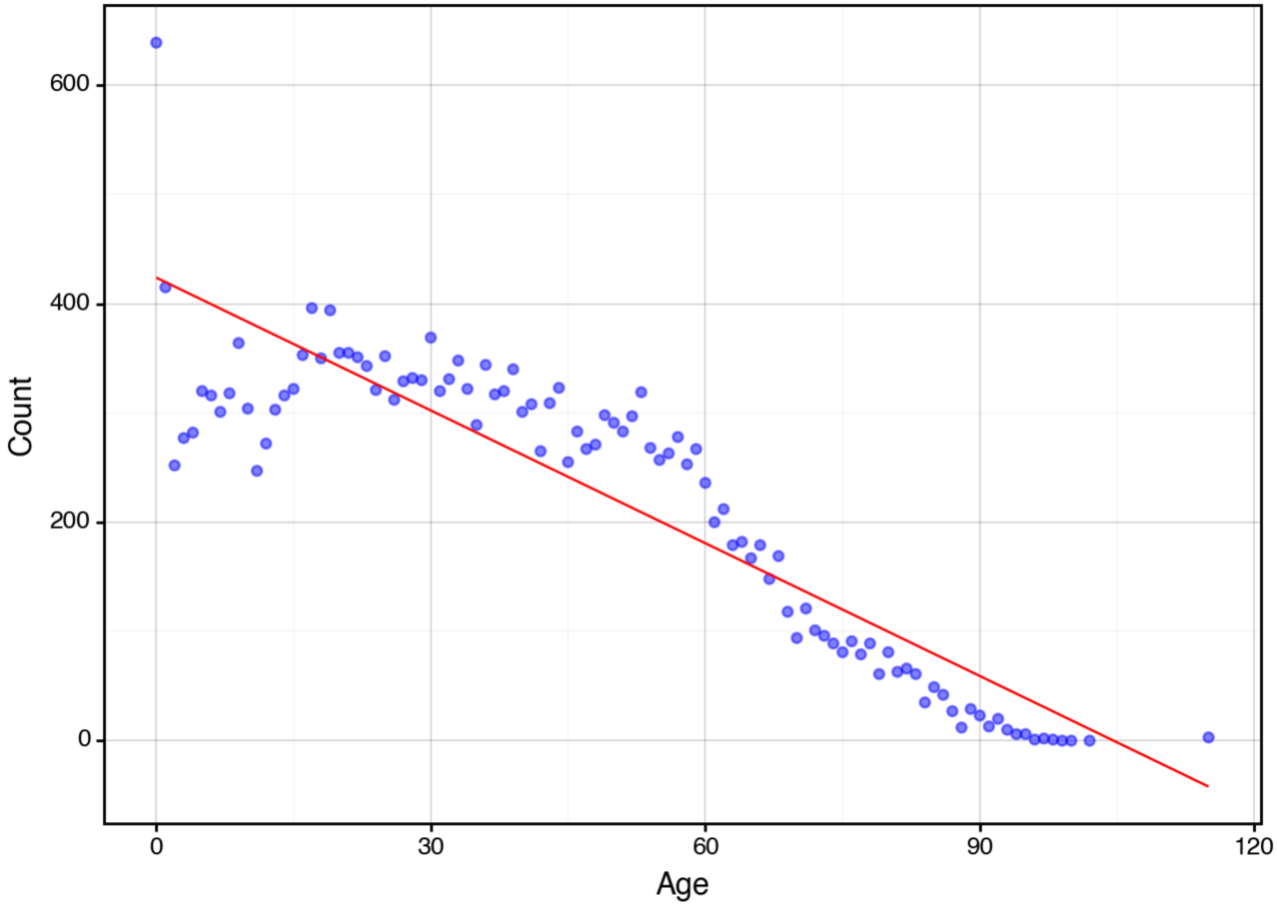
Out[320]:

AGE VS SHOWED UP

In [321]:

```
# from same logic as the showed-up, but this time with age vs people that
will not show up, based on age.
# testing (Age vs No-show)

X = age_no_show[['Age']]
y = age_no_show.Count

model = LinearRegression().fit(X,y)

model.predict(age_no_show[['Age']])
```

```
#the output from this print statement 0.817... means that this model explains
81% of variance age vs no-show
# r2 in this case better fits the data.
print(r2_score(y_true = age_no_show.Count,y_pred =
model.predict(age_no_show[['Age']])))

age_no_show['fitted'] = model.predict(age_no_show[['Age']])

ggplot(aes('Age','Count'),age_no_show)\
    + geom_point(alpha = 0.5,color = 'blue')\
    + geom_line(aes(y = 'fitted'),color = 'red')\
    + theme_linedraw()
0.8179245992945937
```



Out[321]:

AGE VS No-Show

In [322]:

```
age_showed_up.Count.sum()
```

Out[322]:

```
88207
```

In [323]:

```
age_no_show.Count.sum()
```

Out[323]:

```
22319
```

Conclusion:

for both the age vs showing up and age vs no show datasets, we see that there is a negative trend. We have to take into account the number of no shows and people who showed up.

88,207 patients showed up 22,319 patients did not over all, more patients showed up rather than not.

If we are seeing a negative trends while comparing both of those variables to age, we can fairly predict that with the data collected, for whatever reason, there are less elderly people making appointments. Taking a look at both graphs, at age 60 is where the 'Counts' begin to drop in a linear fashion. More importantly, look at the ratio of people who showed up in general vs people who did not show up. Looking at the graphs, we see that the slope for people who showed up is significantly higher than the no-shows. This means that the quantity that showed up as age grows higher drops in larger groups. Therefore we can conclude when age is higher, the less likely they're going to show up to their appointments.

In [ ]: