# **Assessment of Results**

* **Covid-19 Risk Tracker**

### **Business Requirements and Project Purpose**

This program fully met the business requirements provided. It is a tool which could be very helpful to public health educators by allowing them to generate a custom risk profile tailored to their community members as well as provide simple, clear visualizations.

This program met the initial business requirements by: predicting at what level of vaccinations are needed for Covid-19 eradication, generating a personalized Covid-19 risk based on age and the vaccination rate of the patient’s local area, and by showing how much higher their risk would be at 0% vaccination in their area. It does this using multiple clear visualizations, and analyzes the best fitting method for the final data analysis. These visualizations are easily exportable, and can be used to draw helpful conclusions for patients, or to summarize statistical results for the health educators themselves.

The main descriptive method was a Multi-linear regression analysis which shows your personal risk based on age, and community vaccination rate along with your risk with no community vaccination. The analysis of personal risk was performed with multi-linear regression by analyzing how multiple factors coalesce to give a more tailored risk profile. These two factors were selected because they are available in the best data sets, and age shows the higher predictive validity than sex and ethnicity (CDC [1], 2021) A secondary descriptive method was an array of linear regression by ages, that showde a tailored risk profile for each age group at your community vaccination rate.

The main predictive method was based on 3 polynomial regression models. Two were nonlinear regression models (Quadratic, and Complex), while the third was a linear regression model. These were each compared using Standard Error of Regression testing generated on the fly, and the best fit model is used to display results to the end user. All three models are graphed so that the end user can visually assess their accuracy in order to make the predictive power of the trend feel more real.

The program met the expectations of the overall business vision it set out to fulfill as a tool for health professionals to make tailored risk profiles, and generate visuals to make the impact of vaccination more palpable.

### 

### **Assessment of Hypothesis**

The accuracy of the hypothesis here is a complicated issue. Since June, the delta variant of the Covid-19 virus has become 99.7% of Covid-19 cases in the country by the end of the data used, and while the vaccine is still protective for adverse outcomes with the delta variant, it is considered less effective at halting the spread of the delta variant (CDC [2], 2021). By using the end of the data set the model is trained on, it begins to lose predictive power for tracking the spread of Covid-19 during the new outbreak of the Delta variant. When excluding the delta variant by taking into account that the strain the model was trained on is only 0.0-0.3% of shown cases by the end, the model becomes much more accurate. Unfortunately the data sets available do not give precise statistics as to which variant is being tested, and it cannot be ruled out that a contribution of the decline of the original virus is because it is being outcompeted by the delta variant (CDC [2], 2021).

The method directly assessing the main hypothesis shows an increase in the estimated vaccination rate needed for covid eradication going up from 82.4% vaccination needed on 08/06/21 to 84.6% needed on 09/18/21. This would appear to disprove the hypothesis alone, however, the strain initially modeled has reached 0.0%-0.03% of all cases at a real value of 64.5% vaccination rate in the country, which in isolation could be seen as confirmation of the hypothesis. The CDC only publishes the percentages of the Delta Variant’s spread after June 2021 however, when it was already on its way to being half of all cases, so it is not possible to fully map the decline of the original strain against the vaccination rate in order to see which had greater magnitude on its eradication.

The model could be used on the delta variant once we have more data regarding the current vaccine’s impact on the delta variant specifically, or once a new vaccine is introduced specific to the Delta variant, but because the data provided by the CDC doesn’t include exact numbers for the delta variant, and since both variations of the virus are intermixed without clear delineation, there is no way to confirm or deny the hypothesis with the data available. Because of this I can only say that the hypothesis is uncertain at this time, but it is on track to be proven correct pending verification from further data which will not be available for months at least. The program as a whole still meets all its deliverables, and is still highly useful for patient education especially when looking at the strong impact vaccination had on the original strain of Covid-19 prior to the delta outbreak.

My assumption about the data for under 12 year olds was also proved true. The youngest age group never achieved vaccination rates over 0.3% and maintained less than 1 per 100k infection rate for the bulk of the outbreak. This age group’s data was excluded from overall analysis for being 3 standard deviations outside the mean (Frost, April 2021).

### 

### **Code Review**

For the first 4 modules of the program detailed code analysis will not be necessary as they are extremely simple and straightforward. Here are the summaries:

####LOAD AND FILTER DATASET####

Loads the CSV file of the Covid-19 tracking data. This method filters out columns not being used, and only leaves the columns of: Age Group, Percent of that Age Group Fully Vaccinated, and Covid-19 cases for that Week. Additionally the 0-11 year old age group is excluded due to having extremely little data for either cases or vaccinations with still far under 1% of this age group having been vaccinated (CDC [1], 2021).

####USER VARIABLES####

Creates a basic user interface from text prompts, and repeats the prompt with a printed warning if values fall outside of accepted ranges.

####AGES FROM GROUPS####

This method takes the age groups in written form, and replaces them with an average age for the age group so that the data can then be loaded more directly into the statistical methods later.

####GROUP INTO AGES####

This method is used as a precursor for the full multilinear regression that will come later, but it is used for a visualization of Covid-19 risk by age bracket that is extremely helpful. It breaks the Covid-19 data into separate data by age, then uses a simple, linear regression based array to extrapolate the data for each age group separately looking at user input. This, while less accurate overall, gives a broad picture of the data, allowing the user to see how Covid-19 rates affect age groups differently.

#Create GroupBy object of age groups and create list of index names

seperateCases = datasetCases.groupby('AgeGroupVacc')

#Create index of age group names #print(seperateCasesIndex)

seperateCasesIndex = []

for key, value in seperateCases.indices.items() :

seperateCasesIndex.append(key)

#scatterplot matrix using age and user input for vaccination

ageGroupCases = [] #the collection of Y values corresponding to the X labels of the bar chart

for title, group in seperateCases:

X = group[['Series\_Complete\_Pop\_pct\_agegroup']]

y = group['7-day\_avg\_group\_cases\_per\_100k']

model = linear\_model.LinearRegression()

model.fit(X, y)

#Use user input of vaccination for all values

predictedCases = model.predict([[masterPerc]])

#exclude results that fall below 0 risk, as that is not physically possible

if predictedCases[0] > 0:

ageGroupCases.append(predictedCases[0])

else:

ageGroupCases.append(0.0)

####TRAIN MULTILINEAR REGRESSION####

This is the main descriptive method for this program. This takes the two main independent variables, and generates a multilinear regression model to allow extrapolation and risk assessment using both variables. The coefficients for age groups[-45.64], and complete vaccinations [0.150] percentage show a strong correlation toward worse covid risk with increasing age, and a strong correlation for increased vaccination with decreasing covid risk.

#take X and y columns

X = datasetCases[['Series\_Complete\_Pop\_pct\_agegroup', 'AgeGroupVacc']]

y = datasetCases['7-day\_avg\_group\_cases\_per\_100k']

#Generate multilinear regression model

model = linear\_model.LinearRegression()

model.fit(X, y)

####CREATE CHARTS####

This code creates 3 charts for visualization from the already derived data. The first is a bar chart which shows the expected rate of infection for each age group based on the vaccination rate the user input.

#bar for ages#

pyplot.bar(ageGroups, ageGroupCases, bottom=0)

pyplot.title('Estimated Cases per 100k by Age Groups at your Local Vaccination Rate of ' + str(masterPerc\*100) + '%')

pyplot.xlabel('Age Group')

pyplot.ylabel('7 day cases per 100k')

pyplot.xticks(rotation = 45)

pyplot.show()

The second two charts are scatterplots. These show how increasing vaccination correlated with decreasing rates of Covid. The first one splits the age groups by color to allow a 3d picture of how vaccination worked with each age group. The second scatterplot is left completely blue so that the various lines of best fit derived from non-linear regression can be drawn on top in order to better visualize their accuracy.

#scatterplot broken down

fig, ax = pyplot.subplots()

datasetCases.plot(kind='scatter', x='Series\_Complete\_Pop\_pct\_agegroup', y='7-day\_avg\_group\_cases\_per\_100k', c="AgeGroupVacc", title='New Cases vs Percent Vaccinated by Age Group', cmap="viridis" ,ylabel='7 day cases per 100k', xlabel='Complete vaccination percentage', ax=ax)

#scatterplot for trendline

datasetCases.plot.scatter(x='Series\_Complete\_Pop\_pct\_agegroup', y='7-day\_avg\_group\_cases\_per\_100k', title='New Cases vs Percent Vaccinated Overall with Trendlines', ylabel='7 day cases per 100k', xlabel='Complete vaccination percentage')

modelToFit = datasetCases['Series\_Complete\_Pop\_pct\_agegroup']

####TRAIN NONLINEAR REGRESSION####

This is only a few lines of code, but it is some of the most powerful in the program. The numpy.polyfit method fits a polynomial to the scatterplot of data. This is what allows graphic projections as well as future projections for the predictive methods. The numpy.poly1d method then allows for many functions which would be difficult statistical methods themselves to be acted upon these equations. This is used later when calculating Y intercept values as well as the two forms of accuracy gathered.

#generate models

linearFit = numpy.polyfit(modelToFit, y, 1)

quadFit = numpy.polyfit(modelToFit, y, 2)

cubicFit = numpy.polyfit(modelToFit, y, 3)

linearFited = numpy.poly1d(linearFit)

quadFited = numpy.poly1d(quadFit)

cubicFited = numpy.poly1d(cubicFit)

#####DRAW EQUATIONS####

This module formats the 3 nonlinear regression equations and draws them as lines over the scatterplot of data so the end user can visualize the accuracy of the result. It also displays the generated equations so the user can take the raw data for themselves if they wish to extrapolate or statistically alter it.

####MATH FOR NONLINEAR REGRESSION MODELS####

This method calculates accuracy values for the data methods and formats them to be displayed during the PRINT method later. Firstly, raw data from the original file is gathered, then matching predictive data is generated for each of the nonlinear regression models. A result array is then created and formatted to hold the results of this. Created a full object class was considered, but this was far quicker and more linear for this relatively simple use case.

#fill lists with real Y valyes

newYFormat = datasetCases['7-day\_avg\_group\_cases\_per\_100k']

newXFormat = datasetCases['Series\_Complete\_Pop\_pct\_agegroup']

#generate lists of predicted Y values corresponding to X values to check accuracy

linearXList = linearFited(newXFormat)

quadXList = quadFited(newXFormat)

cubicXList = cubicFited(newXFormat)

#create results objects in format: [name, S, R, yIntercept]

linearResults = ['yelow Linear',0.0,0.0,0.0]

quadResults = ['orange Quadratic',0.0,0.0,0.0]

cubicResults = ['red Complex Polynomial',0.0,0.0,0.0]

S or Standard Error of Regression results are calculated here by checking the real arrays against simulated arrays. This is the most important metric for nonlinear regression analysis as P accuracy and R^2 are not considered nearly as viable for determining a best model type (Frost, 2021). The S value can be used as an expected level of deviation from the normal.

# find the S value, standard error of regression for 3 models by comparing against list of real Y values

linearResults[1] = numpy.sqrt(mean\_squared\_error(newYFormat,linearXList))

quadResults[1] = numpy.sqrt(mean\_squared\_error(newYFormat,quadXList))

cubicResults[1] = numpy.sqrt(mean\_squared\_error(newYFormat,cubicXList))

R^2 values are then generated next. While these are not as useful for model comparisons as S, they are still useful as a judge of overall model reliability in explaining data variance (Frost, 2021).

# R^2 values

linearResults[2] = r2\_score(newYFormat,linearXList)

quadResults[2] = r2\_score(newYFormat,quadXList)

cubicResults[2] = r2\_score(newYFormat,cubicXList)

The X intercept value is an essential part of the prescriptive method as it is the level of vaccination required for Covid-19 eradication. This is stored by default as part of the numpy.poly1d class used to derive these equations.

#find y intercept for each model

linearResults[3] = linearFited.r.real[0]

quadResults[3] = quadFited.r.real[0]

cubicResults[3] = cubicFited.r.real[0]

Models are compared according to S value here and the best fit model will be used to display the results to the end user.

#Compare y values for best fit using Standard Error of Regression value

bestResults = linearResults

if quadResults[1] < bestResults[1]:

bestResults = quadResults

if cubicResults[1] < bestResults[1]:

bestResults = cubicResults

These methods work with the multilinear regression model. They extrapolate from the user input values to predict the user’s current risk, and their risk at 0% vaccination. Values less than 0 are removed since that is not physically possible.

#percent vaccination and age, user outputs using MultiLinear model

predictedCases = model.predict([[masterPerc, userAge]])

predictedCasesZero = model.predict([[0, userAge]])

#find R2 value of multilinear regression model against data

multiResult = model.score(X,y)

#remove possibility of displaying negative risk values

if predictedCases[0] <0.0:

predictedCases[0] = 0.0

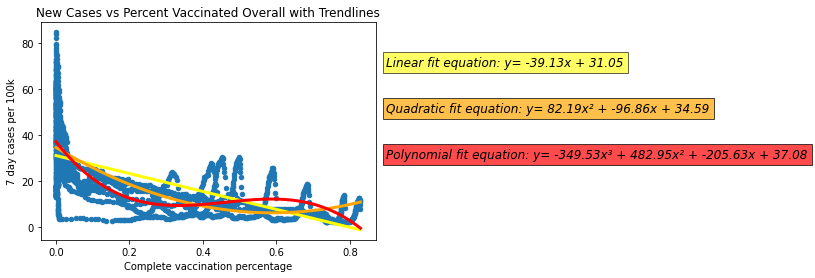
####PRINT RESULTS####

This is a simple method which prints the summary of outcomes and statistical accuracy for the model. Both of these categories were generated live previously in the program and will automatically update as different data is given to the model, or datasets are updated. This is the main, simple display of the predictive and descriptive methods as it simplifies and personally tailors the information into three numbers. Those numbers are: your personal risk of contracting covid, your risk if your local area had 0% vaccination, and the percent vaccination needed for covid-19 eradication.

### 

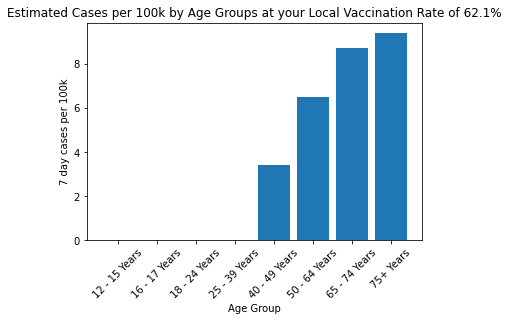
### **Visualizing the Process**

Of the three graphic visualizations, the third may be the most impactful.

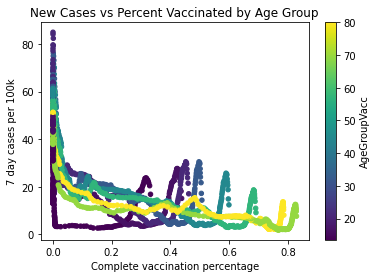


Lines on a graphic may come across as opaque, but these lines represent the nonlinear and linear regression methods used when directly tracking the impact of vaccinations against Covid-19 cases. The linear and Complex Polynomial equations both clearly show a trend in vaccinations toward 0 weekly covid cases at the X intercept, well before 100% vaccination is reached in this country. This graphic directly shows a potential end to the pandemic, and the equations quite possibly give it more weight, because even though a layman might not enjoy reading a string of numbers, they may enjoy visually tracking the line down to 0 Covid cases.

This graphic also showed the data preparation of my model. The scatterplot below the equations was the raw data plotted, and the equations themselves represented 3 of the training models.



The cases by age group trend graph also helps make the data real. It shows the real impact of vaccination rates on the community, so that people can see how the elderly are affected, even if they themselves have low risk. This is based on the array of linear regression analysis broken down by age. While this model is less accurate overall (by around 0.1 on the R^2 score), since it doesn’t weight the values of the oldest people with data trained on the youngest, it seems to have a more accurate risk assessment at the extremes of age (CDC [1], 2019), but the statistical analysis to confirm this was well outside the scope of this project.



The last visualization colors each point of the data by age. This allows a palpable visual to see that the trendlines of vaccine impact aren’t only for the elderly, but that the same pattern exists in practically every age group. This also shows that the phenomenon of the aberrant spikes at the tail of each color are for each age group, but the overall trend remains the same. These spikes were found to be caused by the new outbreak of the delta variant when the data was broken down by date, but this was first observed via the visualization. This visualization overall has a subtle 3d effect that was found to be appealing and attention grabbing when tested by end users.

IMPACT OF VACCINATION

Your daily personal covid risk with a 0% vaccination rate in your community would be:

30.62 per 100k. That is 13.5 times higher than it would be at the current vaccination rate.

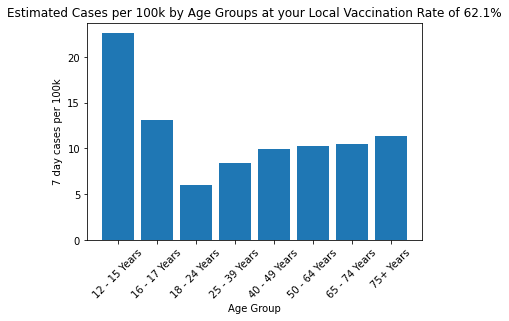
Lastly this data summary output may be the most impactful. It instantly compares your current risk with your simulated risk without vaccination. Your risk without vaccination is calculated using a plethora of real data, since there were no vaccinations for almost the first half of the pandemic. The final scale of risk comparison may be the most helpful part of the summary, and it was certainly seen as helpful by end users.

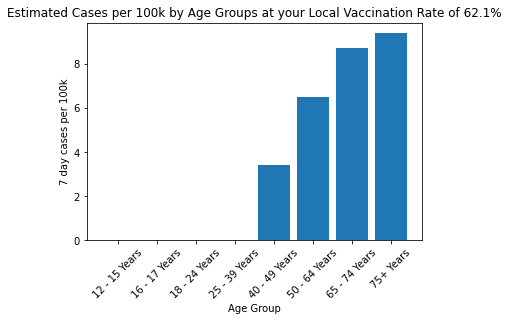
### 

### **Assessment of Accuracy**

The application is very accurate in displaying data to the end user. It uses 3 helpful visualizations, and outputs raw numbers directly to the end user for comparison. These are labeled with objective indicators of scale whilst visualizing as many available variables as is practical.

The data models themselves were initially well within accepted margins for showing impact with an R^2 for the descriptive method of 0.437 based on the Multilinear Regression Model. This is an acceptable score for an epistemological model in my view because it was able to explain nearly half of the variance in covid rates without taking into account seasonal variation, outbreak containment procedures, or even just increased testing. The predictive power has declined in the meantime however due to the new outbreak of the delta variant not being delineated from the previous outbreak.

Example risk calculation from data up to 09/18/21:

Previous data from 08/06/21:

These two breakdowns by age from the array of linear regressions model clearly show the massive difference between the Delta Variant and the original outbreak. The main functional difference between the two is that the Delta Variant spreads vastly more among the youngest age groups (CDC [2], 2021).

For exact numbers of the model’s change in accuracy: between August 6th and September 18th the R^2 value of the Multilinear Regression descriptive model went from 43.7 to 30.7, dropping in predictive power by almost a third while the Standard Error of Regression for the best Nonlinear regression model went from 11.5 to 12.5. This accuracy can actually be expected to go up again in the future however once the data set is fully trained on the delta variant outbreak. This data most likely will not be available for another 6 months however based on the timing of the last two major outbreaks. When it is available, it would be advisable to fully separate the Delta variant data to be trained in isolation if that is possible.

Additional statistical results are generated in the program output, but the S for Nonlinear regression and the R^2 for multi-linear regression are seen as the most reliable (Frost, 2021). The overall accuracy for these models should be considered acceptable to quite good since the decline in program accuracy with new data most likely reflects real world changes outside of the scope of this program or its data set.

### 

### **Testing Protocols**

The main tests performed were unit and integration testing. The first unit tests performed were when loading and formatting the initial CSV file of raw data. These were performed using a print test to manually check random segments of data, whilst using data summaries to ensure the correct number of rows and columns were being modified. Using a basic print test was necessary as no other systems had been set up to verify data in the program yet.

Initially the plan was to set up visualizations last after all data methods were completed, however, upon loading the pandas library in Jupyter notebook, it became clear that good visualizations of data could be used as tests themselves as development went on. Various displays of the data were tested, but a basic scatter plot worked best for the main prescriptive method. This test used the main dependent variable (Covid-19 cases) as the x and the main independent variable (percent vaccinated) as the y. This was used as a simple test of the implemented data methods by allowing a fast visual check of basic accuracy and data fit. This visualization format generated just for testing ended up influencing the final design because it was so clearly descriptive as to the impact of vaccinations on Covid-19 cases.

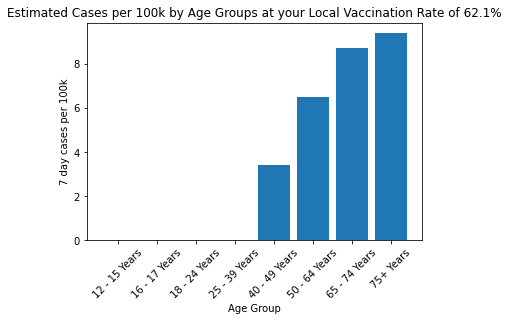
This choice somewhat complicated the testing plan because this required that integration testing between data methods and visualization methods be performed concurrently with the unit tests. Overall however, it seemed to greatly expedite unit testing since bad and strange data were often easily noticed visually. and it made sure data methods were compatible with visualization types before investing too much time in systems which would not be compatible together.

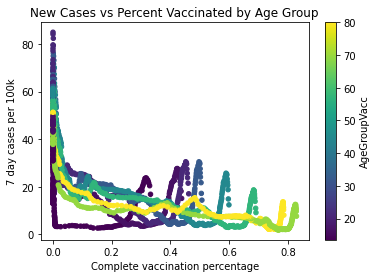
A simple end user test was also performed by allowing non-technical acquaintances to use the program to calculate their results. One of them was a healthcare administrator and another was a public educator. They gave feedback as to how to explain statistical information more clearly for those not acquainted with the statistical methods. This feedback led to the creation of the scatterplot showing the linear-regression analysis alongside the quadratic regression equation and the complex regression equation. This allowed for a visual depiction of best fit while adding a visual element to the statistical summary generated at the end of the report.

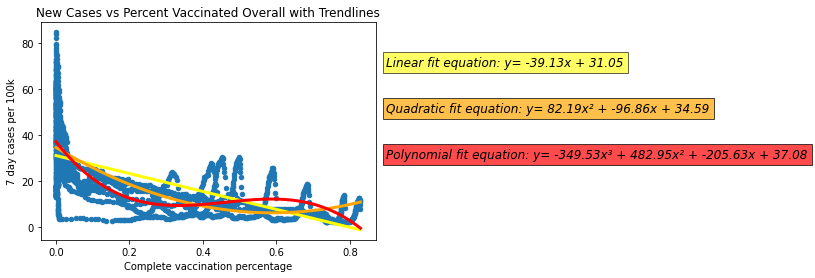
### 

### **Final Results**

All results are taken directly from the program using the dataset ending August 6th 2021 unless otherwise noted.

Raw output from program for 27 year old at 62.1% community vaccination:





YOUR PERSONAL RISK

Your daily personal risk of contracting covid based on your age (27) and local vaccination rate of 62.1%:

2.27 per 100k

IMPACT OF VACCINATION

Your daily personal covid risk with a 0% vaccination rate in your community would be:

30.62 per 100k. That is 13.5 times higher than it would be at the current vaccination rate.

COVID-19 ERADICATION

Estimated overall vaccination rate needed for covid eradication:

82.4% Vaccination

STATISTICAL INFORMATION

The multiple variable model used to derive your personal risk was able to predict 43.7% of the variance in Covid-19 rates when looking at age and vaccination rates (Based on calculated R^2 score).

The estimated vaccination rate needed to end the outbreak was derived from the red Complex Polynomial equation shown above by extrapolating to the X intercept at 0 cases per 100k overall.

This model was able to predict 53.3% of the variance observed in Covid-19 rates, and was the best fit of the three linear and non-linear models used.

Covid-19 rates are expected to vary by around 11.5 per 100k people compared to the predicted numbers, based on the statistical variance of our data for the best fit method (based on S or Standard Error of Regression).

Output with data from September 18th, 2021 for comparison:

YOUR PERSONAL RISK

Your daily personal risk of contracting covid based on your age (27) and local vaccination rate of 62.1%:

9.08 per 100k

IMPACT OF VACCINATION

Your daily personal covid risk with a 0% vaccination rate in your community would be:

30.80 per 100k. That is 3.4 times higher than it would be at the current vaccination rate.

COVID-19 ERADICATION

Estimated overall vaccination rate needed for covid eradication:

84.6% Vaccination

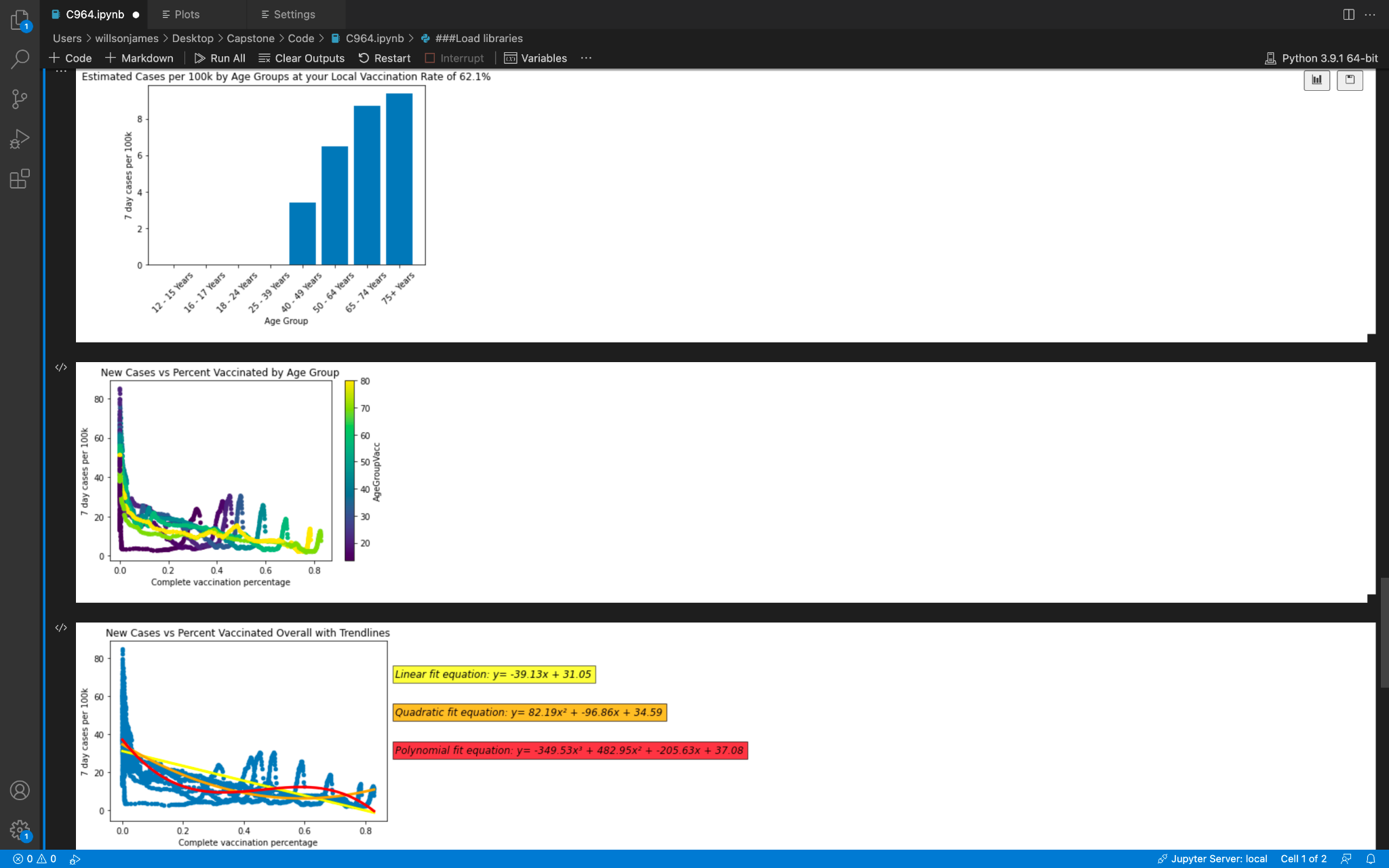
STATISTICAL INFORMATION

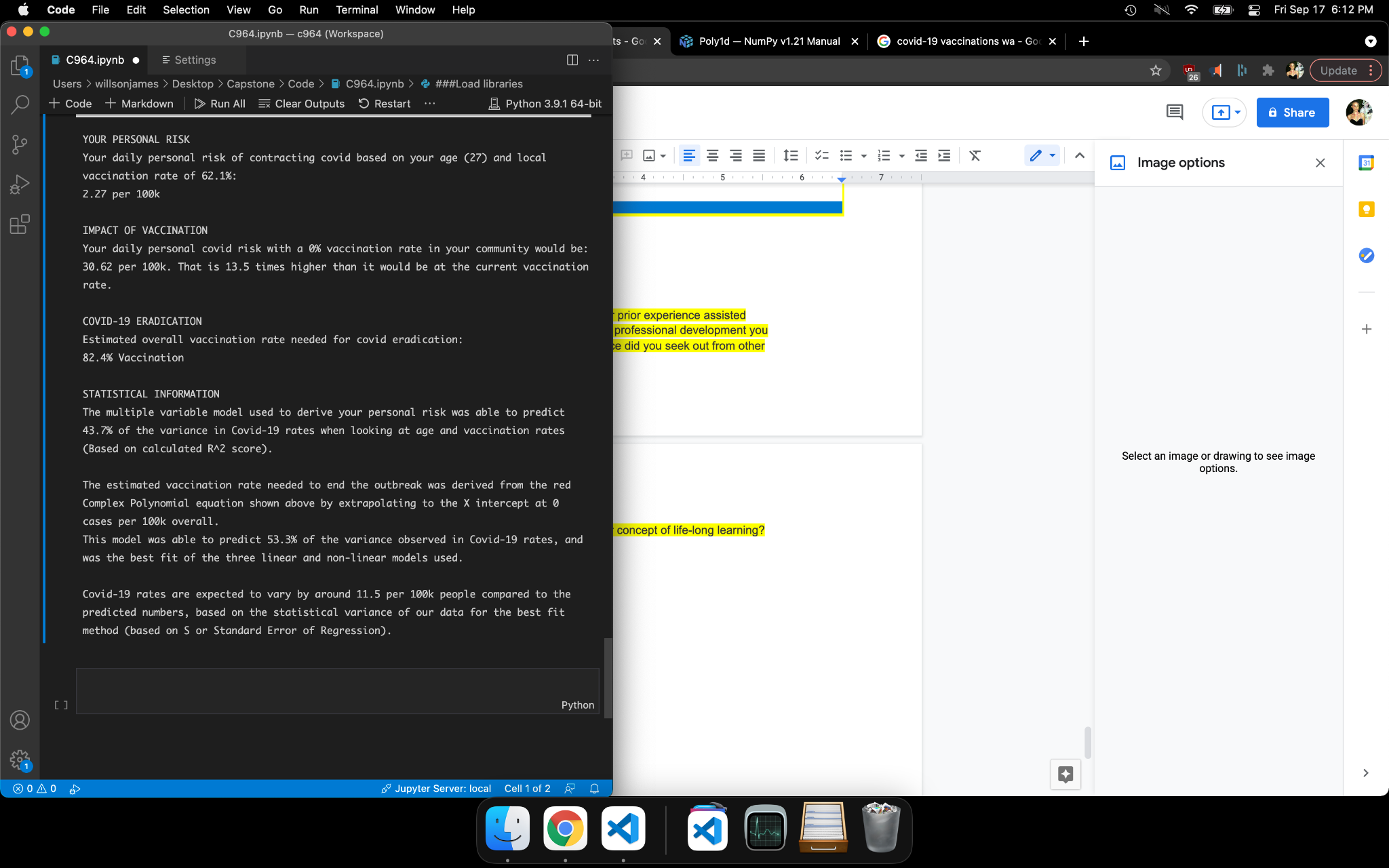
The multiple variable model used to derive your personal risk was able to predict 30.7% of the variance in Covid-19 rates when looking at age and vaccination rates (Based on calculated R^2 score).

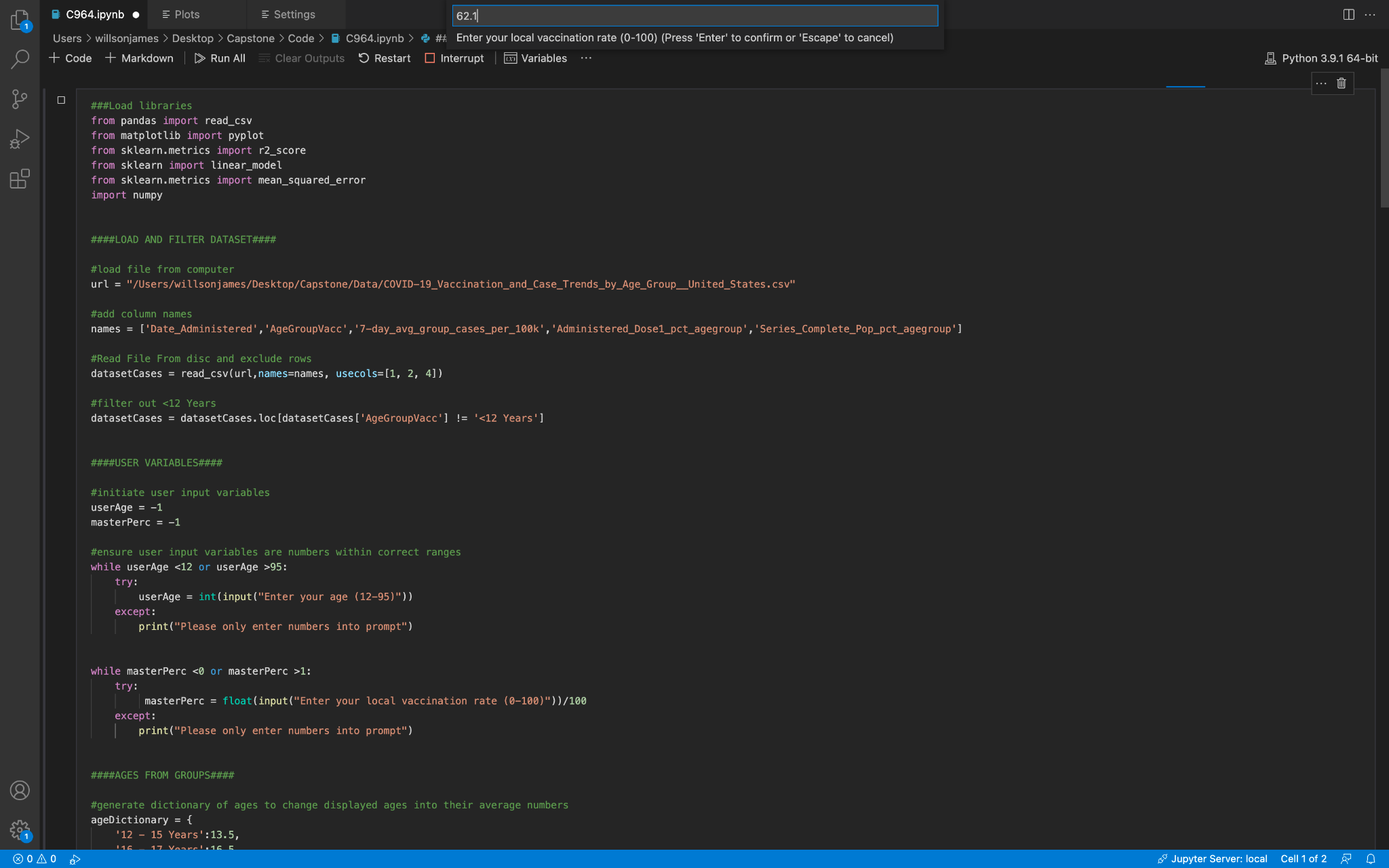
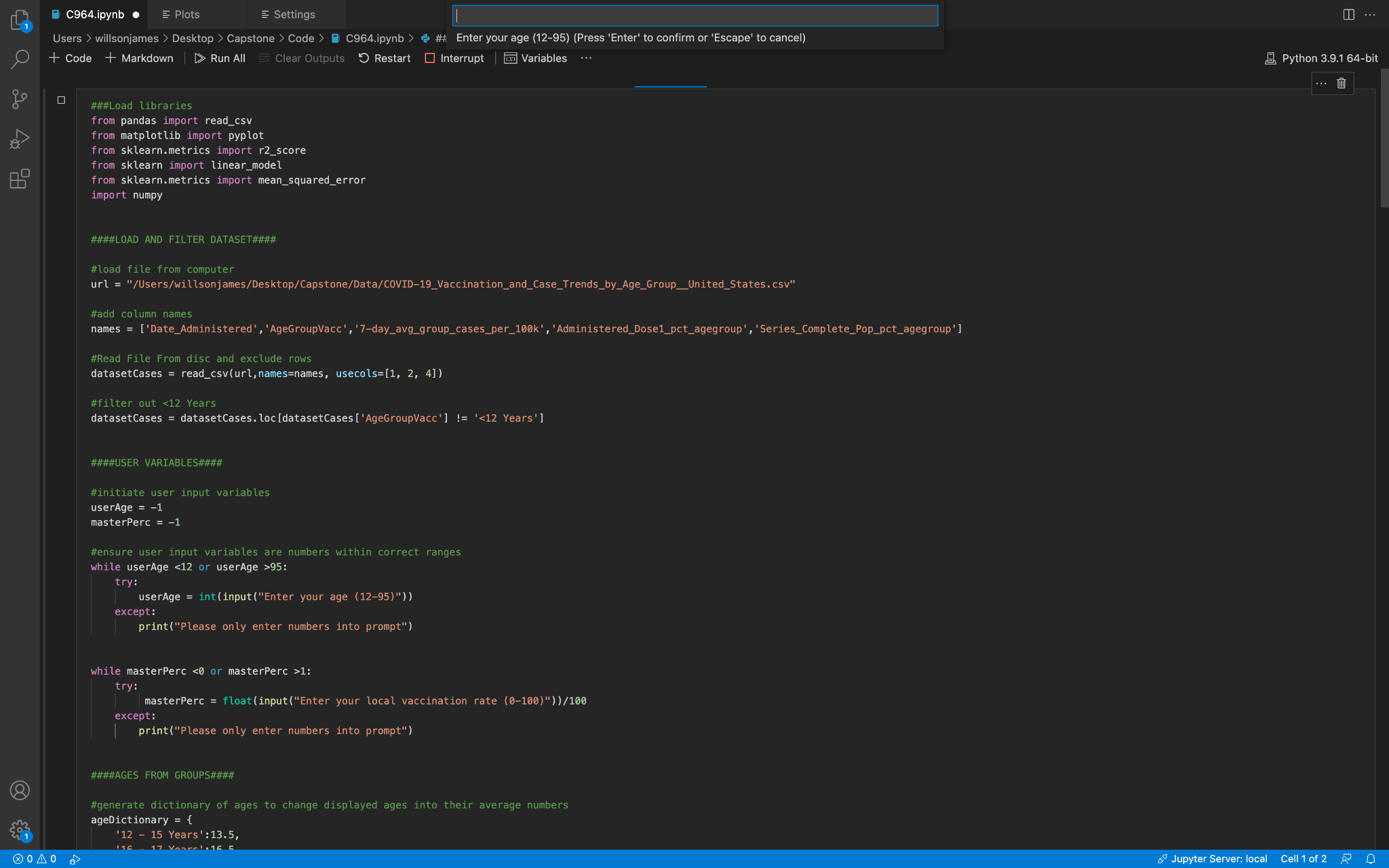
The estimated vaccination rate needed to end the outbreak was derived from the red Complex Polynomial equation shown above by extrapolating to the X intercept at 0 cases per 100k overall.

This model was able to predict 41.8% of the variance observed in Covid-19 rates, and was the best fit of the three linear and non-linear models used.

Covid-19 rates are expected to vary by around 12.5 per 100k people compared to the predicted numbers, based on the statistical variance of our data for the best fit method (based on S or Standard Error of Regression).

Screenshots:



User Prompts:

### 

### Lessons Learned

My personal history had a lot to do with my ability to finish this project. While this project was almost completely unrelated to anything else I’d ever worked on in terms of the tech stack, the skills I learned in my undergrad degree in biology were essential here. Without prior experience in literature review, and practice with epistemology and statistical methods I would have been completely lost.

More importantly, my background as a medical assistant and radiology tech greatly informed my decisions for this project. As a medical assistant, nearly half my job was in patient education, and I personally worked all of 2020 at an urgent care, testing people for Covid-19 and filing tracking reports to the CDC. I brought all this to the project in order to create the tool I would have wanted when doing public health outreach, in order to inform patients of their personal risk directly so that they could make informed choices about their health. I believe that the tool succeeded at this mission. I was able to volunteer at two Covid-19 vaccination drives prior to this project, and the information I had access to through that has also greatly informed my understanding here.

Prior to the project, I had no experience with machine learning or applying detailed statistical methods to python. I turned to many online resources, but especially wolfram alpha and khan academy. I believe that this project developed me professionally nearly incalculably, as the data manipulation tools I learned have already allowed me to complete projects that would have taken hours by hand, in only minutes.

Unfortunately, however, I do have to say that I developed professionally by seeing exactly what not to do from the Python communities of Leetcode and Stack Overflow. Even educational tutorials from these communities at times will use complete cryptic variable names of single letters, and throw in advanced methods without explanation when giving advice to beginner projects. Answers will be highly rated for being “elegant”, but are often actually completely unreadable and unmaintainable in the end. I have found that many python libraries also lack the highly detailed documentation I got used to while programming in Java. This is a completely subjective analysis of course, but it inspired me to be even more clear with my commenting, to pick a descriptive system of variable names and stick with it, and to do the work right ahead of time, doing something correctly the first time in order to save myself lots of headache and confusion earlier.

For outside assistance I also turned to a health administrator and a public educator that I know. They gave essential feedback of my program that helped make it feel more like a professional data product, while also giving more clear summaries to the end users. This feedback gives me a paradigm I will bring to future products. A user interface should not be what the programmer finds convenient, but should be based on what will be impactful to the end users. This advice helped me generate visuals which I found to be highly impactful relative to my early attempts. These lessons will stick with me professionally as lifelong learning.

### Citations

Centers for Disease Control and Prevention [1]. (2021). *COVID-19 vaccination and Case trends by age Group, United States*. CDC. Retrieved September 18, 2021, from https://data.cdc.gov/Vaccinations/COVID-19-Vaccination-and-Case-Trends-by-Age-Group-/gxj9-t96f.

Centers for Disease Control and Prevention [2]. (2021). *Delta variant: What we know about the science*. CDC. Retrieved September 18, 2021, from https://www.cdc.gov/coronavirus/2019-ncov/variants/delta-variant.html.

Frost, J. (2021, April 24). *Standard error of the regression vs. r-squared*. Statistics By Jim. Retrieved September 18, 2021, from https://statisticsbyjim.com/regression/standard-error-regression-vs-r-squared/.