Clearing Assumptions & Modeling Heart Rate and Heart Disease with Linear and Logistic Regression

Kyle Desjardins, Kyle O'Neill

Wentworth Institute of Technology

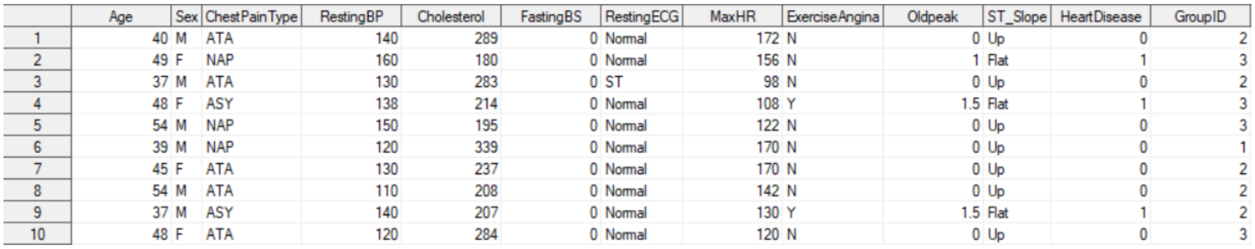
Abstract

Mathematical modeling is important in predicting future outcomes based on previous data. The two most frequently used models are linear and logistic regression. Linear regression is the process of predicting a continuous variable while logistic regression is a classification method. For example, it may be more appropriate to use linear regression when predicting heart rate of future patients. Logistic regression may be more accurate when you are predicting a binary outcome like if a new patient will have heart disease or not. Before continuing with either, there are assumptions of each that must be verified to ensure an accurate model. In the health field, it is critical these models are accurate as we are always creating new medicine.

Introduction

In the field of Statistical Programming, linear and logistic regression are regarded as commonplace methods for forecasting clinical data. In this project, we explain the process of clearing assumptions in order to utilize these methods. In addition, we provide the mathematics that power these important regression models. Our project offers digestible and impactful examples and theory to demonstrate the complexity of the methods used within Statistical Programming.

Dataset



	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease	GroupID
1	40	M	ATA	140	289	0	Normal	172	N	0	Up	0	2
2	49	F	NAP	160	180	0	Normal	156	N	1	Flat	1	3
3	37	M	ATA	130	283	0	ST	98	N	0	Up	0	2
4	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1	3
5	54	M	NAP	150	195	0	Normal	122	N	0	Up	0	3
6	39	M	NAP	120	339	0	Normal	170	N	0	Up	0	1
7	45	F	ATA	130	237	0	Normal	170	N	0	Up	0	2
8	54	M	ATA	110	208	0	Normal	142	N	0	Up	0	2
9	37	M	ASY	140	207	0	Normal	130	Y	1.5	Flat	1	2
10	48	F	ATA	120	284	0	Normal	120	N	0	Up	0	3

The dataset that is used for this project has 13 variables and 918 observations. All observations are split up into three equal groups shown by GroupID on the far right. Since this dataset is a practice dataset used for logistic regression, our logistic model was made to predict the chance, or odds, of getting heart disease.

Since our response variable has a binary outcome (0 they don't have heart disease, 1 they have heart disease), we are using logistic regression since this is a classification method of prediction. Let's do a quick rundown of what the coefficients mean for this. Take the variable age, for example. Say the coefficient is 0.0198. This means for every increase of age by one, it multiplies the odds of having heart disease by $e^{0.0198}$.

To understand linear regression, we are using a simple linear model:

Linear Regression: $Heart\ Rate = \beta_0 + \beta_1(Age) = 191.99\ - 1.031(Age)$

Since linear regression predicts continuous variables, we chose a response variable that is continuous. Now, 191.99 is just the average heart rate if we have an age of 0. The coefficient of age means that for every increase in age by one year, heart rate decreases by -1.031. The derivation on how to get these equations will be seen later in this paper.

Before continuing with the modeling, there are some assumptions that need to be verified before accurately modeling. We will first go through the assumptions of linear regression, then we will go through the assumptions of logistic regression.

Linear Regression

Assumption of Normality

For this assumption, the residuals of the model must follow a normal distribution. But what is a normal distribution?

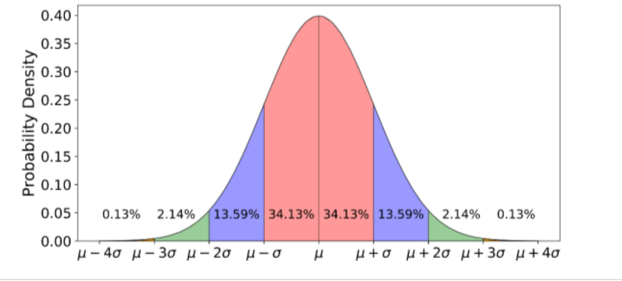


Figure 1: Normal Distribution [1]

Normally distributed data is when the data is symmetric around its mean. Within one standard deviation of the mean falls 68% of the data, within two standard deviations falls roughly 95% of the data, and within three standard deviations falls roughly 99% of the data. Smaller the

[1] Galarnyk, M. (2019, November 5). *Explaining the 68-95-99.7 rule for a normal distribution*. Medium. Retrieved July 28, 2022, from https://towardsdatascience.com/understanding-the-68-95-99-7-rule-for-a-normal-distribution-b7b7cbf760c2

standard deviation, the closer to the mean the data points are. In our case, we will take residual points (or errors) and standardize them to a standard normal curve where we know the probability underneath that curve. A residual is just the observed data point minus the fitted value from our model:

$$\hat{\varepsilon}_i = (obs.) - (fit)$$

A standard normal curve is the same as above, but with a mean of zero and standard deviation of one. The probability density function of a standard normal curve is given by:

$$f(x) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{\mathrm{x}}{2}}$$

Skewed data pulls the mean, median, and mode away from each other. Here is a quick graphic:

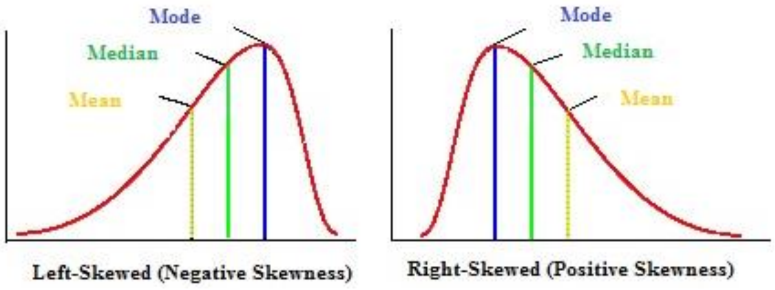
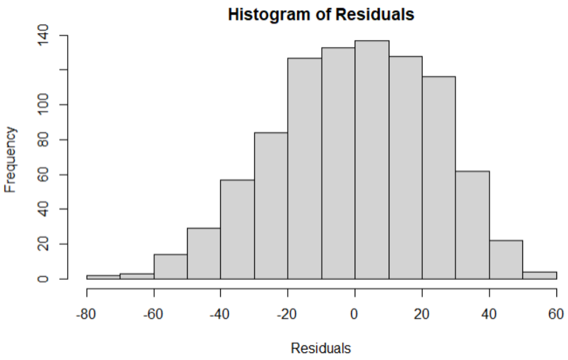
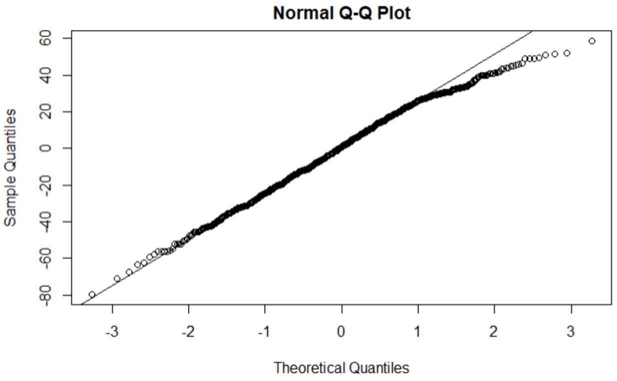

Mode
Median
Mean
Left-Skewed (Negative Skewness)
Mode
Median
Mean
Right-Skewed (Positive Skewness)

Figure 2: Skewed Distributions [2]

Having the mean, median, and the mode of a dataset not the center can cause bias and inaccurate results. In our case, if the residuals of the linear model are not normally distributed, then our model does not explain all the trends. Every point has a residual. So, we can use those values and put them into a histogram and Q-Q Plot to test for normality.

[2] Glen, S. (2022, February 23). *Skewed distribution: Definition, examples*. Statistics How To. Retrieved July 28, 2022, from https://www.statisticshowto.com/probability-and-statistics/skewed-distribution





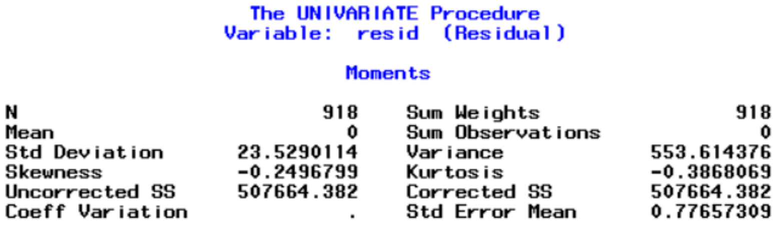
We can see with the histogram of the residuals that they follow roughly a bell shaped curve centered around zero. This is a good visual when deciding normality. Next, we look at the Q-Q Plot where we take theoretical quantiles from a standard normal distribution and see if the sample quantiles of the residuals are equal to that. If they are, they would sit on the Y=X line. In this case, we see some skewness at the top of the plot since it falls away from the line. Are the residuals skewed? We can measure skewness which is the measure of symmetry of data:

$$g = \frac{\sum_{i=1}^{N}(Y_i - \bar{Y})^3}{S^3}$$

Y_i is the data point, $\bar{Y}$ is the mean of the variable Y, and S is the standard deviation of the variable. Our variable is just residuals. Since we are dealing with a normal distribution, skewness will be centered around zero. Thus, symmetric data should be around zero. Negative values indicate the data is skewed left, and positive values indicate the right skewness. The threshold for skewness is ± 2. In addition, we can also measure kurtosis, the measure of heaviness of the tails of the distribution. High kurtosis means the tails are heavy (long) and have the potential for outliers. The equation for kurtosis is:

$$g = \frac{\sum_{i=1}^{N}(Y_i - \bar{Y})^4}{S^4} - 3$$

It is the next dimension of the skewness equation. We subtract three from this equation because kurtosis is centered around three. To make it easier for interpretation, we center it around 0. Therefore, negative values show the data is heavy tailed to the left, and positive values show heavier tail to the right. The closer the value is to zero, the more normal your data is.



```
                         The UNIVARIATE Procedure
                       Variable:  resid  (Residual)

                                 Moments

N                              918    Sum Weights                918
Mean                             0    Sum Observations             0
Std Deviation           23.5290114    Variance            553.614376
Skewness                -0.2496799    Kurtosis            -0.3868069
Uncorrected SS          507664.382    Corrected SS        507664.382
Coeff Variation                  .    Std Error Mean      0.77657309
```

Using the univariate procedure in SAS, we can find these values easily. This procedure also calculates all the basic statistics to look at for each variable we choose. But, for our case, we are just looking at skewness and kurtosis. Skewness shows -0.249 and kurtosis is -0.386 showing that this data is pretty symmetric around its mean of 0. A quick observation: One of the properties of residuals is that the mean of them will be zero because their sum is zero. This makes sense since there is a reference line (our model) that the values are getting subtracted from. Therefore, when you add them together, you will get zero.

Skewness and kurtosis are pretty vague ideas when it comes to their quantification. Looking at them could show the reader skewness, but we must verify with a statistical test. In this case, we will be using the Shapiro-Wilk test. This test gives a W score to test if the distribution of our sample is normal. W is bounded between 0 and 1 where 1 is a perfect fit. Since this is a goodness of fit test, we must standardize our residuals first and compare that to the actual dispersion of the data. Our test statistic is as follows:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

The null and alternative hypothesis are:

$$H_0 = Values\ deviate\ from\ a\ normal\ distribution$$

$$H_a = Values\ are\ normally\ distributed$$

Hypothesis testing is just the use of sample data to test the likelihood of some outcome. One of the two tests has to be true. There is some critical value that needs to be defined before the test. If the test statistic falls outside that critical value, that means there is enough evidence to reject the null. This means we can assume the alternative is true. This correlates to a low probability value (P-Value). This is just the probability of getting what we observed if the null hypothesis is true. The critical value used in our project was $\alpha = 0.05$. This means we expect to see a test statistic as severe as the one observed 5% of the time. This threshold depends on the study, so it should always be said at the beginning so readers can make their own interpretation. The numerator of the W score is the standardization of our data to a normal distribution. a_i is a constant generated from the covariances, variances, and means from a standard normal distribution. $x_{(i)}$ is the ranked order of the residuals. Ranked order is important when we want to see how each observation compares to each other. Dividing by the sum of squares gives us that connection from the spread of our data to the normal distribution. The closer these values are to each other, the better fit our data is to a normal distribution.

Let's dive into what a_i is:

$$a_i = (a_1,\ a_2, \dots,\ a_n) = \frac{m^T V^{-1}}{C}$$

Where

$$C = \left|\left|\mathrm{V}^{-1}\mathrm{m}\right|\right| = (\mathrm{m}^{\mathrm{T}}\mathrm{V}^{-1}\mathrm{m})^{\frac{1}{2}}$$

C is a vector norm where $m = (m_1, \ldots, m_n)^T$ is the expected values of the order statistics of random variables from a standard normal distribution. In other words, m is a transposed matrix from the standard normal. V is a covariance matrix. Now, what is a vector norm? First off, a vector is a quantity that has a direction and magnitude. Velocity is just one example of a vector.

A vector norm, denoted generally by $|x|$, has the following properties:

1. $|x| > 0$ when $x \neq 0$ and $|x| = 0$ if and only if $x = 0$

2. $|kx| = |k||x|$ for any scalar k

3. $|x + y| \leq |x| + |y|$: follows some form of the triangle inequality

The most common vector norm is the L2 Norm, or Euclidean Norm:

$$\left|\left|x\right|\right|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

For example, say we have the vector $< 1,2,3 >$. The Euclidean Distance of this vector is:

$$\sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$$

This is just the distance this vector is from the origin (0,0). Other dimensions exist like the L3 Norm, L4 Norm, and so on.

Now, let’s understand variance and covariance to understand V in the above equation. The formula for variance is:

$$var(x) = \frac{\sum_1^n (x_i - \bar{x})^2}{n - 1}$$

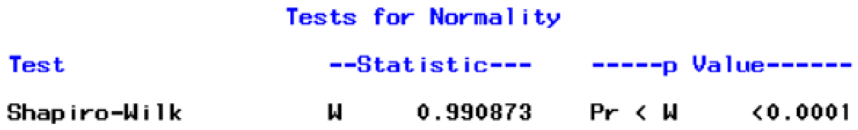
$\bar{x}$ is the mean of the variable. We take the sum of squares of a variable and divide by $n - 1$ to quantify the spread of the data away from its mean. We divide by $n - 1$ instead of n because we want an accurate estimate of the population variance. This is because the sample variance is calculated using the sample mean, but that mean was created based on the sample data. So, dividing by $n-1$ will give us an unbiased estimate of the population. Covariance is similar, but it is the variation of two random variables around their expected means:

$$cov(x,y)=\frac{\sum_1^n(x_i-\bar{x})(y_i-\bar{y})}{n-1}$$

One addition that covariance has over variance is that we can see how two variables move with each other. When one increases, does the other? When one increases, does the other decrease? We can gauge direction of two variables with this. So, a covariance matrix is the variance and covariances of two random variables given by:

$$\begin{bmatrix} var(x_1) & \cdots & cov(x_1,x_n) \\ \vdots & \ddots & \vdots \\ cov(x_n,x_1) & \cdots & var(x_n) \end{bmatrix}$$

Since we are dealing with a standard normal, all these values to fill these variables for a_i are found using that distribution. These are just constants that are found in a table online and are omitted from this paper. As stated previously, the numerator for the W score is just the standardization of our residuals to a standard normal. Then dividing by the sum of squares to see how it compares to our residuals. Let’s look at the results:



```
                         Tests for Normality

Test                  --Statistic---    -----p Value------

Shapiro-Wilk          W     0.990873    Pr < W     <0.0001
```

Since our W score is 0.990873, we can say that our residuals are normally distributed. Since our p-value is below $\alpha=0.05$, we reject the null and can confidently say our residuals are normally distributed. Even though the Q-Q Plot showed some skewness, we verified the normality of the errors with the Shapiro-Wilk test.

Assumption of Homogeneity of Variance

This is the assumption that there are equal variances between groups. Earlier, it was stated that all 918 observations were split up into three equal groups. The variance between those groups should be equal or it may cause inaccurate and biased results. To ensure the accuracy of the results, the group sizes must be close to equal. If the variances of the groups are not equal, that is called heteroscedastic. To understand the test for equal variances, we need to understand what a chi-squared distribution and a F-distribution is. A chi-squared distribution is related to a normal distribution. If Z is a normal random variable, Z^2 has a chi-squared (χ^2) distribution with one degree of freedom. A degree of freedom is the number of independent observations that go into calculating any distribution or estimate. For a distribution, degrees of freedom are just the number of observations minus 1.

If we have $Z_1, Z_2, \ldots, Z_k$ random variables, then $Z_1^2 + Z_2^2 + \cdots + Z_k^2$ has χ^2 distribution with k degrees of freedom. We can graph their curves with the probability density function:

$$f(x) = \frac{x^{\frac{k}{2}-1} e^{-\frac{x}{2}}}{2^{\frac{k}{2}} \Gamma\left(\frac{k}{2}\right)}$$

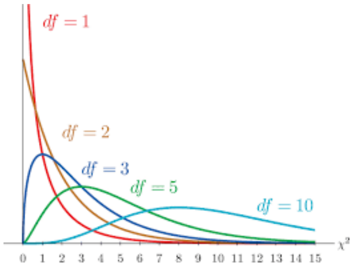


Figure 3: Chi-Squared Distributions [3]

[3] jbstatistics. (2013, November 21). *An introduction to the chi-square distribution*. YouTube. Retrieved July 28, 2022, from https://www.youtube.com/watch?v=hcDb12fsbBU

Since our pdf depends on the degrees of freedom, the distribution curve changes. The reason we need this is because a F-distribution is related to chi-squared. If a random variable U_1 has a χ^2 distribution with ν_1 degrees of freedom and U_2 has a χ^2 distribution with ν_2 degrees of freedom, F is the ratio of these two variables over their respective degrees of freedom. In other words, $\frac{\frac{U_1}{\nu_1}}{\frac{U_2}{\nu_2}}$ has a F-distribution with ν_1, ν_2 degrees of freedom. A F-distribution is used commonly in analysis of variance test and the pdf, just like a chi-squared distribution, relies on the degrees of freedom:

$$f(x) = \frac{\Gamma\left(\frac{\nu_1+\nu_2}{2}\right)\left(\frac{\nu_1}{\nu_2}\right)^{\frac{\nu_1}{2}} x^{\frac{\nu_1}{2}-1}}{\Gamma\left(\frac{\nu_1}{2}\right)\Gamma\left(\frac{\nu_2}{2}\right)\left(1+\frac{\nu_1}{\nu_2}x\right)^{\frac{\nu_1+\nu_2}{2}}}$$

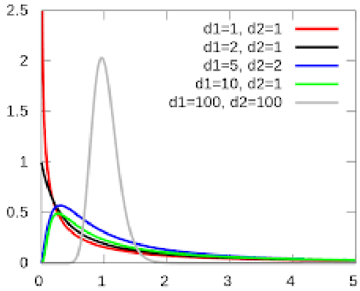


Figure 4: F-Distributions [4]

We see that the higher the degrees of freedom, the more normal the distribution is. This distribution is a one-sided test. Once we find a critical value (which is based off the degrees of freedom from the statistical test), if our F-value (that we calculate) falls outside that critical range, we will reject the null.

[4] jbstatistics. (2012, November 4). *An introduction to the F distribution*. YouTube. Retrieved July 29, 2022, from https://www.youtube.com/watch?v=G_RDxAZJ-ug&ab_channel=zedstatistics

The test we will be using for our dataset is called Levene's Test. This statistical test tests if k samples have equal variances giving us our null and alternative hypothesis:

$$H_0\colon\ \sigma_1^2 = \sigma_2^2 = \cdots = \sigma_k^2$$

$$H_a\colon \sigma_i^2 \neq \sigma_j^2,\ for\ at\ least\ one\ pair\ (i,\ j)$$

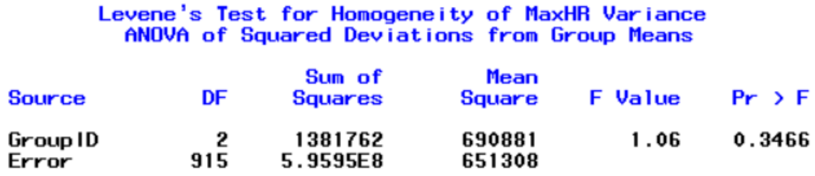
Our significance level is $\alpha = 0.05$. The test statistic is:

$$W = \frac{(N-k)}{k-1}\frac{\left(\sum_{i=1}^{k} N_i(\bar{Z}_{i.} - \bar{Z}_{..})^2\right)}{\sum_{i=1}^{k}\sum_{j=1}^{N_i}\left(Z_{ij} - \bar{Z}_{i.}\right)^2} = \frac{(N-k)}{k-1}\left(\frac{SS(G)}{SS(E)}\right)$$

N are the total observations. k are the number of samples (three in our case). N_i is the number of observations for group i. $\bar{Z}_{i.}$ is the mean for group i given by $\bar{Z}_{i.} = \frac{1}{N_i}\sum_{j=1}^{N_i} Z_{ij}$. $\bar{Z}_{..}$ is the grand mean of all 918 observations given by $\bar{Z}_{..} = \frac{1}{N}\sum_{i=1}^{k}\sum_{j=1}^{N_i} Z_{ij}$. The reason this test is used is because the data itself does not have to take on a normal distribution. It is a powerful and robust test because Z_{ij} can use the mean, median, or 10% trimmed mean based on the skewness of the data without losing its statistical power:

1. $Z_{ij} = \left|Y_{ij} - \bar{Y}_{i.}\right|\colon\ \bar{Y}_{i.}\ is\ the\ mean\ of\ the\ i^{th} subgroup$
2. $Z_{ij} = \left|Y_{ij} - \widetilde{Y}_{i.}\right|\colon\ \widetilde{Y}_{i.}\ is\ the\ median$
3. $Z_{ij} = \left|Y_{ij} - \bar{Y}'_{i.}\right|\colon\ \bar{Y}'_{i.}\ is\ the\ 10\%\ trimmed\ mean$

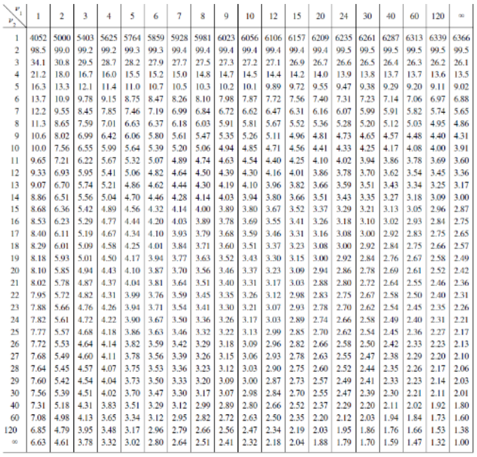
If the data follows a normal distribution, we can use the mean which is what is normally used. If the data is slightly skewed to begin with, we can use the median. Figure 2 shows that with skewed data, the median is more towards the middle of the data. Thus, that will give us more accurate results. If the data is heavy tailed, we can use the 10% trimmed mean. This means we take the mean from the 10th percentile to the 90th percentile and neglect the tails completely. The numerator for this is the sum of squares for the groups. This quantifies the variability between groups. The denominator is the sum of squares of the errors (residuals). This quantifies the variability within each group. Dividing them is a common way to compare them. This W score is equivalent to a F score in an ANOVA test which tests the equality of means between groups. The results of the test are as follows:



```
            Levene's Test for Homogeneity of MaxHR Variance
              ANOVA of Squared Deviations from Group Means

                                Sum of        Mean
Source              DF         Squares      Square    F Value    Pr > F

GroupID              2         1381762      690881       1.06    0.3466
Error              915       5.9595E8       651308
```

From this table, we know the degrees of freedom (DF) is 2 for the group since there are 3 groups. The degrees of freedom of the error are the total number of observations minus the total number of groups (918-3=915). We showed how to get the sum of squares of the group and the error. For an F distribution though, we divide the two sums of squares by their respective degrees of freedom as explained previously. That gives us a mean square of the group and mean square of the errors. The mean square is simply the average sum of squares. Dividing the mean square of the group by the mean square of the error gives us our F-Value of 1.06.

Now that we have our F-Value, we need to find our critical point. We will use a F table at an $\alpha = 0.05$ level:

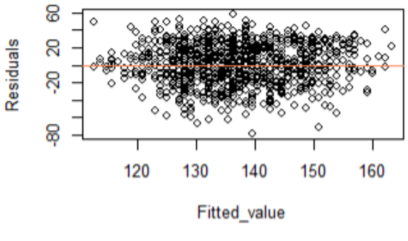


ν_1 / ν_2	1	2	3	4	5	6	7	8	9	10	12	15	20	24	30	40	60	120	∞
1	4052	5000	5403	5625	5764	5859	5928	5981	6023	6056	6106	6157	6209	6235	6261	6287	6313	6339	6366
2	98.5	99.0	99.2	99.2	99.3	99.3	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.5	99.5	99.5	99.5	99.5	99.5
3	34.1	30.8	29.5	28.7	28.2	27.9	27.7	27.5	27.3	27.2	27.1	26.9	26.7	26.6	26.5	26.4	26.3	26.2	26.1
4	21.2	18.0	16.7	16.0	15.5	15.2	15.0	14.8	14.7	14.5	14.4	14.2	14.0	13.9	13.8	13.7	13.7	13.6	13.5
5	16.3	13.3	12.1	11.4	11.0	10.7	10.5	10.3	10.2	10.1	9.89	9.72	9.55	9.47	9.38	9.29	9.20	9.11	9.02
6	13.7	10.9	9.78	9.15	8.75	8.47	8.26	8.10	7.98	7.87	7.72	7.56	7.40	7.31	7.23	7.14	7.06	6.97	6.88
7	12.2	9.55	8.45	7.85	7.46	7.19	6.99	6.84	6.72	6.62	6.47	6.31	6.16	6.07	5.99	5.91	5.82	5.74	5.65
8	11.3	8.65	7.59	7.01	6.63	6.37	6.18	6.03	5.91	5.81	5.67	5.52	5.36	5.28	5.20	5.12	5.03	4.95	4.86
9	10.6	8.02	6.99	6.42	6.06	5.80	5.61	5.47	5.35	5.26	5.11	4.96	4.81	4.73	4.65	4.57	4.48	4.40	4.31
10	10.0	7.56	6.55	5.99	5.64	5.39	5.20	5.06	4.94	4.85	4.71	4.56	4.41	4.33	4.25	4.17	4.08	4.00	3.91
11	9.65	7.21	6.22	5.67	5.32	5.07	4.89	4.74	4.63	4.54	4.40	4.25	4.10	4.02	3.94	3.86	3.78	3.69	3.60
12	9.33	6.93	5.95	5.41	5.06	4.82	4.64	4.50	4.39	4.30	4.16	4.01	3.86	3.78	3.70	3.62	3.54	3.45	3.36
13	9.07	6.70	5.74	5.21	4.86	4.62	4.44	4.30	4.19	4.10	3.96	3.82	3.66	3.59	3.51	3.43	3.34	3.25	3.17
14	8.86	6.51	5.56	5.04	4.70	4.46	4.28	4.14	4.03	3.94	3.80	3.66	3.51	3.43	3.35	3.27	3.18	3.09	3.00
15	8.68	6.36	5.42	4.89	4.56	4.32	4.14	4.00	3.89	3.80	3.67	3.52	3.37	3.29	3.21	3.13	3.05	2.96	2.87
16	8.53	6.23	5.29	4.77	4.44	4.20	4.03	3.89	3.78	3.69	3.55	3.41	3.26	3.18	3.10	3.02	2.93	2.84	2.75
17	8.40	6.11	5.19	4.67	4.34	4.10	3.93	3.79	3.68	3.59	3.46	3.31	3.16	3.08	3.00	2.92	2.83	2.75	2.65
18	8.29	6.01	5.09	4.58	4.25	4.01	3.84	3.71	3.60	3.51	3.37	3.23	3.08	3.00	2.92	2.84	2.75	2.66	2.57
19	8.18	5.93	5.01	4.50	4.17	3.94	3.77	3.63	3.52	3.43	3.30	3.15	3.00	2.92	2.84	2.76	2.67	2.58	2.49
20	8.10	5.85	4.94	4.43	4.10	3.87	3.70	3.56	3.46	3.37	3.23	3.09	2.94	2.86	2.78	2.69	2.61	2.52	2.42
21	8.02	5.78	4.87	4.37	4.04	3.81	3.64	3.51	3.40	3.31	3.17	3.03	2.88	2.80	2.72	2.64	2.55	2.46	2.36
22	7.95	5.72	4.82	4.31	3.99	3.76	3.59	3.45	3.35	3.26	3.12	2.98	2.83	2.75	2.67	2.58	2.50	2.40	2.31
23	7.88	5.66	4.76	4.26	3.94	3.71	3.54	3.41	3.30	3.21	3.07	2.93	2.78	2.70	2.62	2.54	2.45	2.35	2.26
24	7.82	5.61	4.72	4.22	3.90	3.67	3.50	3.36	3.26	3.17	3.03	2.89	2.74	2.66	2.58	2.49	2.40	2.31	2.21
25	7.77	5.57	4.68	4.18	3.86	3.63	3.46	3.32	3.22	3.13	2.99	2.85	2.70	2.62	2.54	2.45	2.36	2.27	2.17
26	7.72	5.53	4.64	4.14	3.82	3.59	3.42	3.29	3.18	3.09	2.96	2.82	2.66	2.58	2.50	2.42	2.33	2.23	2.13
27	7.68	5.49	4.60	4.11	3.78	3.56	3.39	3.26	3.15	3.06	2.93	2.78	2.63	2.55	2.47	2.38	2.29	2.20	2.10
28	7.64	5.45	4.57	4.07	3.75	3.53	3.36	3.23	3.12	3.03	2.90	2.75	2.60	2.52	2.44	2.35	2.26	2.17	2.06
29	7.60	5.42	4.54	4.04	3.73	3.50	3.33	3.20	3.09	3.00	2.87	2.73	2.57	2.49	2.41	2.33	2.23	2.14	2.03
30	7.56	5.39	4.51	4.02	3.70	3.47	3.30	3.17	3.07	2.98	2.84	2.70	2.55	2.47	2.39	2.30	2.21	2.11	2.01
40	7.31	5.18	4.31	3.83	3.51	3.29	3.12	2.99	2.89	2.80	2.66	2.52	2.37	2.29	2.20	2.11	2.02	1.92	1.80
60	7.08	4.98	4.13	3.65	3.34	3.12	2.95	2.82	2.72	2.63	2.50	2.35	2.20	2.12	2.03	1.94	1.84	1.73	1.60
120	6.85	4.79	3.95	3.48	3.17	2.96	2.79	2.66	2.56	2.47	2.34	2.19	2.03	1.95	1.86	1.76	1.66	1.53	1.38
∞	6.63	4.61	3.78	3.32	3.02	2.80	2.64	2.51	2.41	2.32	2.18	2.04	1.88	1.79	1.70	1.59	1.47	1.32	1.00

Figure 5: F Table at $\alpha = 0.05$ [5]

Our critical value is based on the degrees of freedom in the numerator (2) and degrees of freedom in the denominator (915). We move over to two on the top and down to infinity for ν_2 and we get a critical value of 4.61. Since this is a one-sided test, we fail to reject the null that the variances are equal since our F value does not fall past our critical value. We also can see our p-value is 0.3466 which is above 0.05 which is further evidence to failing to reject the null.

Another way we can test equality of variance is by a residual vs. predicted values plot. This type of plot is used to detect non-linearity, unequal error variances, and potential outliers. We will use a different method to predict outliers in this paper.

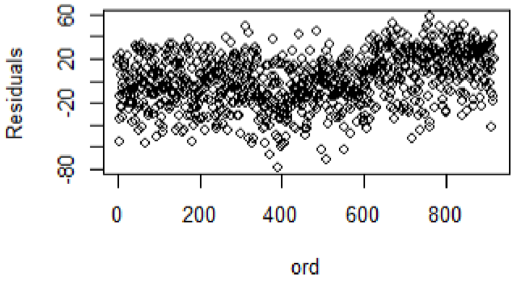


[5] Spiegel, M. R., Srinivasan, R. A., & Schiller, J. J. (2013). *Schaum's outline of probability and statistics* (Vol. 4). Schaum.

In this graph, we are looking for a horizontal band around the 0 line because that suggests equal variance. If there was a cone shape to the data (when the fitted value increased, the residuals got closer to zero), that would suggest heteroscedastic data and we would have to transform the data. But we can see the residuals are spread out evenly around its mean of 0. In addition, if the residuals bounce randomly around the 0 line, that satisfies the assumption of linearity of the residuals.

Assumption of Independence

An easy way to see that the data is independent of each other is through a residual vs. order plot:



There are 918 residuals, so we plot a scatterplot of the residuals and their order. We can see that there is no pattern. This satisfies the assumption that the observations are independent of each other. We can safely say that no observation influences any other observation.

Assumption of No Extreme Outliers

There are a lot of different ways to deal with outliers like the box-plot method. This project goes over a little more advanced topic called the Studentized Residual Approach. We know that a residual is the observation minus its fitted value. The idea behind studentized residuals is to delete an observation one at a time and refit the model to the remaining observations. Then we compare the observed results to their respective fitted values based on the model with that data point deleted. These points are called deleted residuals and standardizing the deleted residuals gives us studentized residuals.

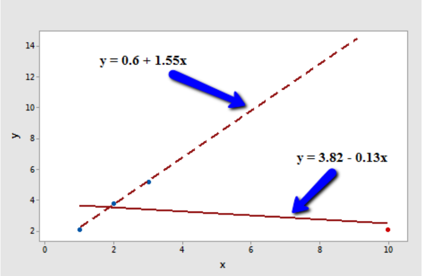


Figure 6: Deleted Residual Example [6]

The solid line represents the line of best fit for all four data points in figure 6. If we delete the red point, we create the new line of best fit given by the dashed line. In this simple example, we can see that the high leverage data point (red data point) is an outlier because it falls away from the natural trend. Leverage is the measurement of how far away each observation is from each other. High leverage data points tend to pull the line of best fit toward it and create inaccurate results. They have the potential to change coefficients drastically. In figure 6, the slopes of the two lines flip completely because of that one point.

How do we know what is considered a large enough change though? That is where the studentized residuals come in. This test follows a T-Distribution. This type of distribution is similar to a normal distribution, but the tails are heavier. You only use this when you are working with sample data and don't know the population standard deviation. The more degrees of freedom, the T-distribution will become a normal distribution:

[6] The Pennsylvania State University. (n.d.). *9.4 - Studentized Residuals*. 9.4 - Studentized Residuals | STAT 462. Retrieved July 31, 2022, from https://online.stat.psu.edu/stat462/node/247/

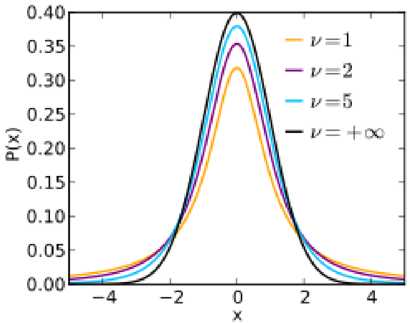


Figure 7: T-Distribution [7]

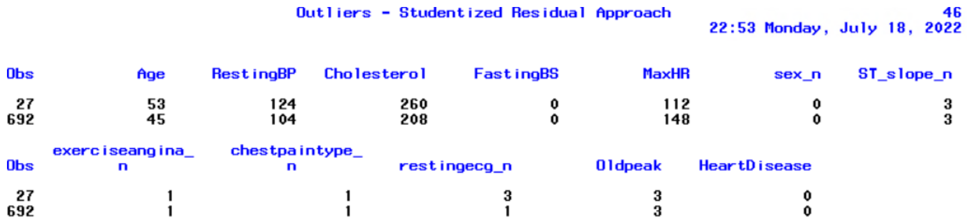
Since we have 918 observations, we are virtually just using a normal distribution since our degrees of freedom are 917. Our test statistic for each residual point is:

$$t_i = \frac{\hat{\varepsilon}_i}{sd(\hat{\varepsilon}_i)\sqrt{1 - h_{ii}}}$$

$\hat{\varepsilon}_i$ is the residual at point i. h_{ii} is the leverage which is given by:

$$h_{ii} = \frac{1}{n} - \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n}\left(x_j - \bar{x}\right)^2}$$

Leverage is bound between 0 and 1. High leverage data points can cause issues if they don't follow the natural trend of the rest of the data. Since we are using a normal distribution, if $t_i > |3|$, that falls outside our significance level of $\alpha = 0.05$. Thus, that residual will be considered an outlier.



```
                         Outliers - Studentized Residual Approach                    46
                                                               22:53 Monday, July 18, 2022

Obs          Age    RestingBP   Cholesterol     FastingBS       MaxHR       sex_n   ST_slope_n

 27           53          124           260             0         112           0            3
692           45          104           208             0         148           0            3

     exerciseangina_   chestpaintype_
Obs                n                n    restingecg_n      Oldpeak    HeartDisease

 27                1                1               3            3               0
692                1                1               1            3               0
```

[7] Glen, S. (2021, September 30). *T-distribution / student's t: Definition, step by step articles, video*. Statistics How To. Retrieved July 31, 2022, from https://www.statisticshowto.com/probability-and-statistics/t-distribution/

From our results, we can see that two observations fall outside that critical area and can be considered influential points. Since it is only two points, the next step is what to do about them. The easiest method is to just delete them from the dataset. Deleting two observations from the whole 918 will not change anything in the grand scheme.

Logistic Regression

Assumption of Binary Response Variable

This is an easy assumption to clear. For our logistic model, we are predicting heart disease. In the dataset, heart disease has two (binary) values: 0 they don't have heart disease, 1 they have heart disease. This assumption is verified.

Assumption of Large Sample Size

Our dataset has 918 observations. This is plenty large enough for the project we are doing.

Assumption of No Multicollinearity

To understand multicollinearity, we first need to understand what correlation is. Earlier, we learned that covariance measures the direction of two random variables. It didn't measure how strong that relationship is. That is where correlation comes in. Correlation is bounded between -1 and 1 and is given by:

$$r = \frac{cov(x,y)}{s_x s_y} = \frac{\frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{n-1}}{s_x s_y} = \cdots = \frac{(n\sum xy - \sum x \sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (y)^2]}}$$

We take the covariance and divide by the product of the two variable's respective standard deviations. After some derivation, the final equation is what is used in multiple programming languages since it is more straightforward for a computer to handle. What does the output represent?

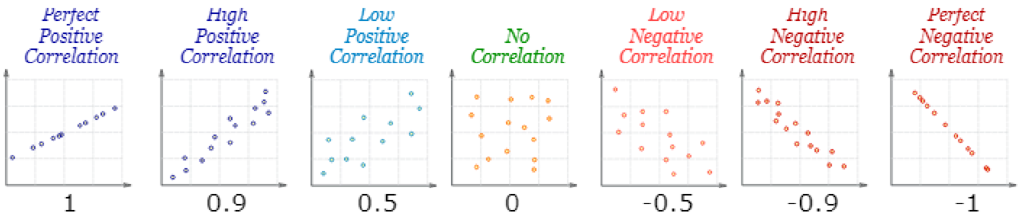


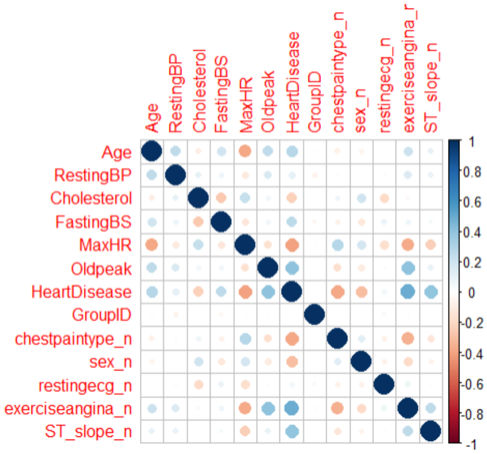
Figure 8: Correlation [8]

The closer the value is to 1, it means there is a positive correlation between the variables. When one increases, the other increases also. In addition, the higher correlation there is between variables, we can create a linear model between the two variables since one depends on the other variable. The closer the value is -1, it is the opposite. When one increases, the other tends to decrease. You can also create a linear model with those variables, but the linear relationship is negative.

For logistic regression, we don't want any of our independent variables in the model to be correlated with each other. That is where this assumption comes into play. Multicollinearity creates a problem in regression because all independent variables are influencing each other in some way. Thus, technically not being independent. It creates a couple problems. The first being that it is difficult to choose the significant variables to put into the model. In a sense, if two variables are highly correlated, it can be seen as double counting and one should be taken away from the model. Multicollinearity also creates an unstable model. This means change in one variable drastically changes the one it is highly correlated with. This makes the model difficult to

[8] Pierce, R. (2021). *Correlation*. Math is Fun. Retrieved July 28, 2022, from https://www.mathsisfun.com/data/correlation.html.

interpret. The last issue is overfitting. If we apply our model to a different sample, the accuracy of the model will drop significantly since it was “overfit” to the sample we trained on.

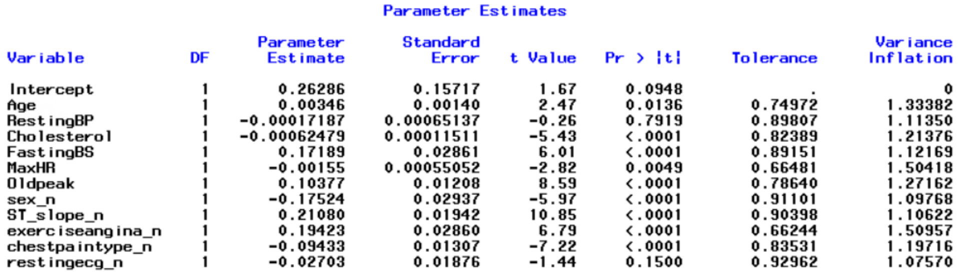


Using this correlation matrix, we can easily see the correlation of each variable of our dataset. The diagonal line makes sense that it is a perfect 1 because each variable is perfectly correlated with itself. From this matrix, we can safely say there is no correlation between any two variables that is too high. The threshold is ± 0.8. Anything above or below the threshold would be a cause for concern. Say we did have high correlation between two variables. Which variable would we take away from our model?

The way to choose which variable to get rid of is called the variance inflation factor. This is just the measure of how much the variance of the coefficient is inflated due to multicollinearity. It is equal to the ratio of the regression model variance to the model variance of a model that only includes the i^{th} variable:

$$VIF = \frac{1}{1 - R_i^2}$$

The VIF is calculated for every independent variable. It is bounded below by 1. The closer the VIF is to 1, that variable can stay in the model. If the value goes above five, there is cause for concern that that variable is an issue due to multicollinearity.



```
                                   Parameter Estimates

                        Parameter      Standard                                        Variance
Variable          DF     Estimate         Error    t Value    Pr > |t|     Tolerance    Inflation

Intercept          1      0.26286       0.15717       1.67      0.0948             .            0
Age                1      0.00346       0.00140       2.47      0.0136       0.74972      1.33382
RestingBP          1  -0.00017187    0.00065137      -0.26      0.7919       0.89807      1.11350
Cholesterol        1  -0.00062479    0.00011511      -5.43      <.0001       0.82389      1.21376
FastingBS          1      0.17189       0.02861       6.01      <.0001       0.89151      1.12169
MaxHR              1     -0.00155    0.00055052      -2.82      0.0049       0.66481      1.50418
Oldpeak            1      0.10377       0.01208       8.59      <.0001       0.78640      1.27162
sex_n              1     -0.17524       0.02937      -5.97      <.0001       0.91101      1.09768
ST_slope_n         1      0.21080       0.01942      10.85      <.0001       0.90398      1.10622
exerciseangina_n   1      0.19423       0.02860       6.79      <.0001       0.66244      1.50957
chestpaintype_n    1     -0.09433       0.01307      -7.22      <.0001       0.83531      1.19716
restingecg_n       1     -0.02703       0.01876      -1.44      0.1500       0.92962      1.07570
```

Using SAS, we can see that all our values are close to 1 and we can validate that there is no multicollinearity. Tolerance is another way to verify the VIF. Tolerance is just $\frac{1}{VIF}$, so the closer the tolerance is to 1 is more evidence that that variable can stay in the model.

Assumption of Linear Relationship Between Explanatory Variables and the Log-Odds of the Response Variable

To understand log-odds, we first need to understand odds. Say there is an 80% chance of winning a game. That means there is a 20% chance of losing that game. Therefore, the probability of success is $P(S) = 0.8$ and the probability of failure is $P(F) = 0.2$. The odds equation is:

$$\frac{P}{1-P} = \frac{P(S)}{P(F)} = \frac{0.8}{0.2} = 4$$

In our simple example, we have odds of 4 to 1. This means for every 4 games, there is 1 loss. Now, what are log-odds? Since probability must be positive and less than one, we can create our P and $1 - P$:

$$p = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

$$1 - p = \frac{1}{1 + e^{\beta_0 + \beta_1 x}}$$

Plugging this into our odds equation:

$$\frac{p}{1 - p} = e^{\beta_0 + \beta_1 x}$$

Now we need to take the natural log of both sides to get rid of e:

$$\ln\left(\frac{p}{1 - p}\right) = \beta_0 + \beta_1 x$$

Therefore, we can see that the log of the odds is equal to a linear equation. For simplicity, the derivation of this includes only one independent variable. The derivation does not change with more independent variables. To see this visually, we can take this easy example:

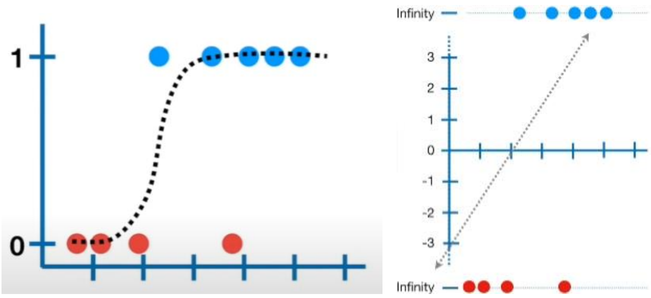


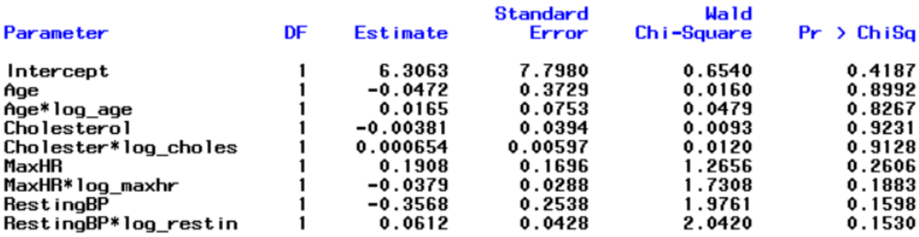
Figure 9: Log-Odds & Linearity [9]

For the graph on the left in figure 9, let's say the Y-axis is the probability of having heart disease and the X-axis is just the variable age. Instead of fitting a line to the data, we fit an "S" shaped logistic function that is bounded between 0 and 1. The curve tells us the probability that someone has heart disease based on the model's independent variables. For example, the higher the age, the more likely you are to have heart disease. The opposite is also true. If you

[9] Starmer, J. (2018, June 4). *Logistic regression details PT1: Coefficients*. YouTube. Retrieved July 28, 2022, from https://www.youtube.com/watch?v=vN5cNN2-HWE&ab_channel=StatQuestwithJoshStarmer

plug in the log-odds equation for every value between 0 and 1, we will be getting a line as shown on the right. For example, $\ln\left(\frac{0.5}{1-0.5}\right) = 0, \ln\left(\frac{.95}{1-.95}\right) = 3$, and so on. For now, we are assuming this is the line of best fit. This is just a visual way to see the linearity between the log-odds of the response variable and the explanatory variables.

The test for this assumption is called the Box-Tidwell Test. The basic idea of this test is to take the continuous variables and create an interaction term with the natural log of that variable. An interaction term is just multiplying the original variable by the natural log of the same variable. Then we use a chi-squared distribution and a Wald Chi-Squared Test to test if the interaction terms in a new model are significant or not. If they are significant, then that shows nonlinearity between the log-odds of heart disease and that variable. If it is not significant, then that shows linearity.



```
                                          Standard          Wald
Parameter               DF    Estimate       Error    Chi-Square    Pr > ChiSq

 Intercept               1      6.3063      7.7980        0.6540        0.4187
Age                      1     -0.0472      0.3729        0.0160        0.8992
Age*log_age              1      0.0165      0.0753        0.0479        0.8267
Cholesterol              1    -0.00381      0.0394        0.0093        0.9231
Cholester*log_choles     1    0.000654     0.00597        0.0120        0.9128
MaxHR                    1      0.1908      0.1696        1.2656        0.2606
MaxHR*log_maxhr          1     -0.0379      0.0288        1.7308        0.1883
RestingBP                1     -0.3568      0.2538        1.9761        0.1598
RestingBP*log_restin     1      0.0612      0.0428        2.0420        0.1530
```

For all the continuous variables in our dataset, we can see that the interaction terms are above our significance level of $\alpha = 0.5$. This means they are not significant, and we can safely say there is a linear relationship between the log-odds of heart disease and the explanatory variables.

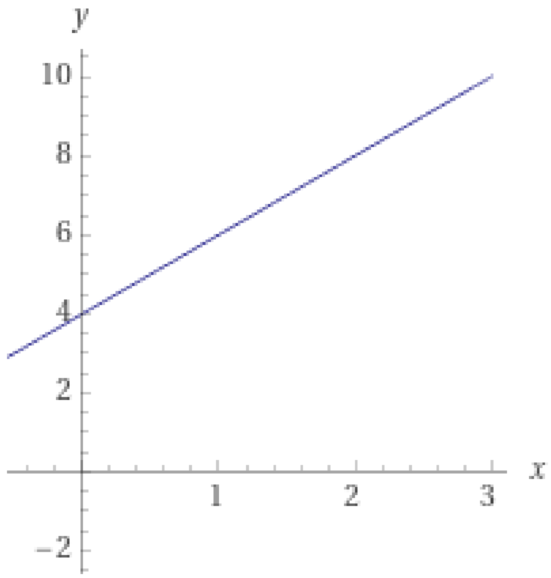
Now that the assumptions of both linear and logistic regression are validated, we can continue to the actual modeling, how we got each coefficient for both models, and the accuracy of our models.

Linear Regression Model:

Once the assumptions for linear regression have been cleared, we can get to work creating a linear regression model. First we can explain simply what linear regression is and when we use it. The use case of linear regression is when we are predicting a quantitative response variable Y with a predictor variable X. The equation for a linear regression line is

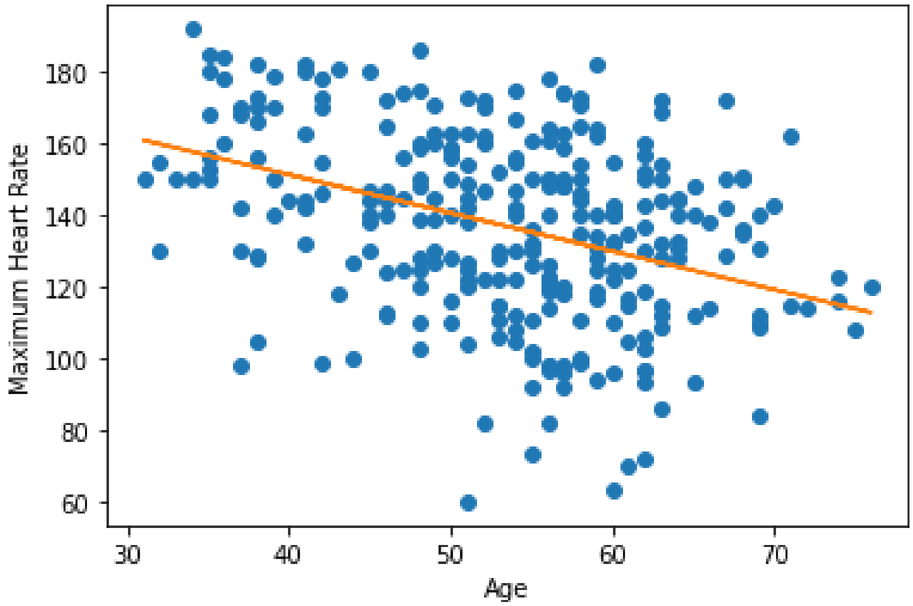
$$Y = \beta_0 + \beta_1 X$$

Where Y is the response, X is the predictor variable, and β_0 and β_1 are two unknown constants that represent the intercept and the slope in the linear model, known as the coefficients or parameters. The graph below is a simple representation of a line with a Y intercept of 4 and a slope of 2.



Line of Best Fit:

Once we have an idea of when to use linear regression and what it look like, we can get into estimating the coefficients. Lets look at the end product of a linear regression model to get a better understanding of why we create a model.



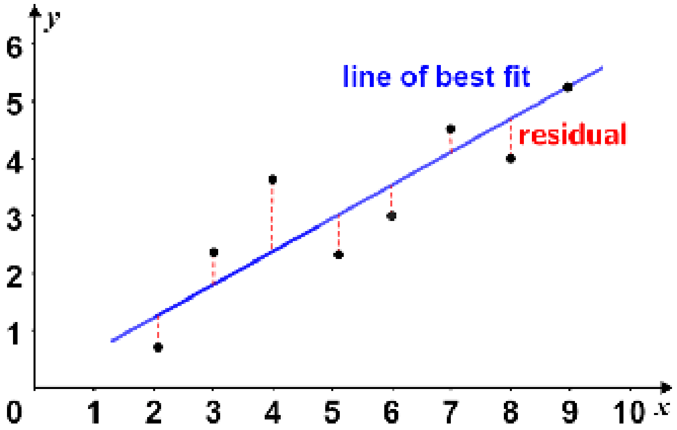
The graph above shows the line of best fit, which we will dive more into later, but we can explain what it is used for now. The blue dots here is a sample of the data known as the test data, and the orange line is the line that best fits the training data. Again, this will be explained later but for now we can view this orange line as the line that predicts the heart rate of a patient given the patients age. Suppose we are given the age of a patient that is 70. This orange line would predict that the Maximum heart rate of a 7- year old patient would be about 120 beats per minute. If we look a the circled blue dot in red, we can see that this patient has Maximum Heart Rate that is drastically below what our model predicted. This can be a good indication that further tests should be done on this patients to see of there are any problems. That is why we create these models, to help visualize when there may be problems regaurding a patients health.

Estimating Coefficeints:

Now that we understand when and why we use these models, we can start to break down the math behind them. We will be estimating the coefficents, of the β_0 and β_1 terms that we talked about earlier when explaining the equation for a linear regression line. In order to get the orange line, we need to solve for the line of best fit. We can do this by minimizing the residuals, where a residual is the difference between the observed value and predicted value. This is represented as

$$e_i = y_i - \widehat{y_i}$$

Where e_i is the residual at a point where x = i, and y_i is the observed value when x = i, and $\widehat{y_i}$ is the predicted value at the point x = i.



The graph above shows a visualization of the residual, where the blue line is the line of best fit and the black points are the actual observed values. The red dotted lines are the distance between the two. Our goal when creating a line of best fit is to create a line that minimizes the distance between predicted value and residuals. Because residuals can be positive or negative, we will end up squaring their values so we can add them together and get a positive value. But for now, this is how to calculate a residual and what they look like on a graph. When we add all the residuals squared together, we get the sum of squared residuals, or the RSS. This will be the key term that we will try to minimize that will allow us to get a line that best fits the data.

The Residual Sum of Squares represents the total variance in a model, so the smaller the RSS the better the line fits the data. That is represented as

$$e_i = y_i - \widehat{y_i}$$

$$\hat{y} = \widehat{B_0} + \widehat{B_1}x$$

$$y = B_0 + B_1x$$

$$RSS = e_1^2 + e_2^2 + \cdots + e_n^2$$

This can be rewritten as

$$RSS = \left(y_1 - \widehat{B_0} - \widehat{B_1}x_1\right)^2 + \left(y_2 - \widehat{B_0} - \widehat{B_1}x_2\right)^2 + \cdots$$

$$+ \left(y_n - \widehat{B_0} - \widehat{B_1}x_n\right)^2$$

Deriving linear regression coefficients by using partial derivatives:

To solve for the value of $\widehat{B_0}\ \widehat{B_1}$ we have to take the partial derivatives of the RSS with respect to each. The result is as follows where $\hat{Y}_i$ is the line of the predicted linear regression model, a is the value $\widehat{B_0}$, B is the value $\widehat{B_1}$, and S is the RSS.

$$\hat{Y}_i = a + Bx_i$$

$$S = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

$$S = \sum_{i=1}^{n} (Y_i - a - Bx_i)^2$$

$$\frac{\partial S}{\partial a}[\sum_{i=1}^{n} (Y_i - a - Bx_i)^2]$$

$$0 = \sum_{i=1}^{n} -2(Y_i - a - Bx_i)$$

$$0 = \sum_{i=1}^{n} (Y_i - a - Bx_i)$$

$$0 = \sum_{i=1}^{n} Y_i - \sum_{i=1}^{n} a - B\sum_{i=1}^{n} x_i$$

$$\sum_{i=1}^{n} a = na$$

$$0 = \sum_{i=1}^{n} Y_i - na - B\sum_{i=1}^{n} x_i$$

$$a = \frac{\sum_{i=1}^{n} Y_i - B\sum_{i=1}^{n} x_i}{n}$$

$$\frac{\partial S}{\partial b}[\sum_{i=1}^{n}(Y_i - a - Bx_i)^2]$$

$$0 = \sum_{i=1}^{n} -2x_i(Y_i - a - Bx_i)$$

$$0 = \sum_{i=1}^{n} x_i(Y_i - a - Bx_i)$$

$$0 = \sum_{i=1}^{n}(x_iY_i - ax_i - Bx_i^2)$$

$$a = \bar{Y} - B\bar{x}$$

$$0 = \sum_{i=1}^{n}(x_iY_i - (\bar{Y} - B\bar{x})x_i - Bx_i^2)$$

$$B = \frac{\sum_{i=1}^{n}(x_iY_i - \bar{Y}x_i)}{\sum_{i=1}^{n}(x_i^2 - \bar{x}x_i)}$$

We can simplify the last term to

$$\widehat{B_1} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\widehat{B_0} = \hat{y} - \widehat{B_1} * \bar{x}$$

$$\text{where } \bar{y} \equiv \frac{1}{n}\sum_{i=1}^{n} y_i \text{ and } \bar{x} \equiv \frac{1}{n}\sum_{i=1}^{n} x_i$$

This is the form of the coefficients of a linear regression model. $\widehat{B_1}$ can also be viewed as the covariance of (x,y) divided by the variance of x.

Logistic Regression Model:

Now that we have computed the linear regression model, we can move onto the logistic regression model. The first question we must answer is why would be use a logistic regression model, and what exactly is its output.

We use Logistic Regression when the Y value we are looking for has a binomial distribution. In other words when the response variable is categorical and can be place in one of two categories denoted as failure or success, 0 or 1.

A logistic regression model will produce a probability, p(x), that a response variable is in the category defined as a success, which when dealing with two categories is represented as 1 where the category representing a failure is represented as a 0.

Earlier in the paper we discuss log odds and how a logistic regression function looks.

Again, the formula is

$$p(X) = \frac{e^{B_O + B_1 X}}{1 + e^{B_O + B_1 X}}$$

We are tasked with finding the values of $\widehat{B_1}$ and $\widehat{B_0}$ that will give us the best fitting model. For logistic regression, this is done with the maximum likelihood function.

Because logistic regression predicts probabilities, rather than just classes, we

can fit it using likelihood. For each training data-point, we have a vector of

features, xi, and an observed class, yi. The probability of that class was either p, if y_i= 1, or 1 − p, if y_i= 0. The likelihood is then

$$l(\mathrm{B}_0, B_1) = \prod_{i:y_i} p(x_i) \prod_{i':y_i'} (1 - p(x_{i'}))$$

If we take the log-likelihood we can turn the product into summations

$$\begin{aligned}\ell(\beta_0, \beta) &= \sum_{i=1}^{n} y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)) \\ &= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i \log \frac{p(x_i)}{1 - p(x_i)} \\ &= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i(\beta_0 + x_i \cdot \beta) \\ &= \sum_{i=1}^{n} -\log\left(1 + e^{\beta_0 + x_i \cdot \beta}\right) + \sum_{i=1}^{n} y_i(\beta_0 + x_i \cdot \beta)\end{aligned}$$

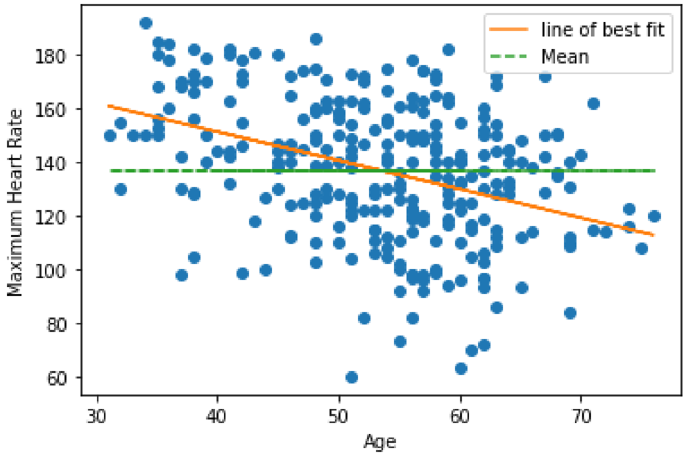
We can take the derivative with respects to the coefficients and solve, similarly to what we did when solving for the linear regression coefficients. Once we get values for $\widehat{B_0}$ $\widehat{B_1}$ we have our model, that will help us predict the probability that a patient has heart disease.

Machine Learning Key Concepts:

There are three basic principles with machine learning. First, we clean and split the data. This means we search for values in the data that don't make sense and deal with them. An example would be if we have the age of a patient inputted as a negative number. We know that age can never be a negative number, so we can either remove the data point completely, replace the datapoint with the average of all the data, or do some other process. Choosing how to deal with incorrect data is something that changes form project to project and needs to be noted when showing the final version of a model, as it can have an impact. For the data we chose, there were no datapoints we needed to clean, so we left it as it is. Next we split the data. This means we break the data into a training dataset, and a testing dataset. It is common to train the data on 70% of the data, and test on the remaining 30%. Other common splits are 50/50 and 60/40 which I will show in the code at the end. A training dataset is the data that we will fit our model to. So, for linear regression, we will create a line of best fit on the training data. Then we can test for accuracy on the test data.

Linear Regression Model in Python:

Here we show a simple linear regression model with two variables. The response variable is Maximum Heart Rate, and the independent variable is Age. We chose to show an example on these two variables because as we researched all the variables independently, we saw there was scientific evidence that with an increase in age there is shown to be a decrease in heart rate. This would be a good model to show simple linear regression.



We've shown the derivation of how to estimate the coefficients, now we can create a model. We plot our linear regression line against the test values. The estimates for the coefficients are as follows

$$\hat{y} = \widehat{B_0} + \widehat{B_1}x$$

Where:

$$\widehat{B_0} = 194.05965568$$

$$\widehat{B_1} = -1.06928335$$

The orange line is the line we created using the training data, and the blue dots are the observed values from the test data. We also added a green dotted line, which is the mean of the observed values and will help us when we discuss the accuracy of the model we created.

Assessing Accuracy:

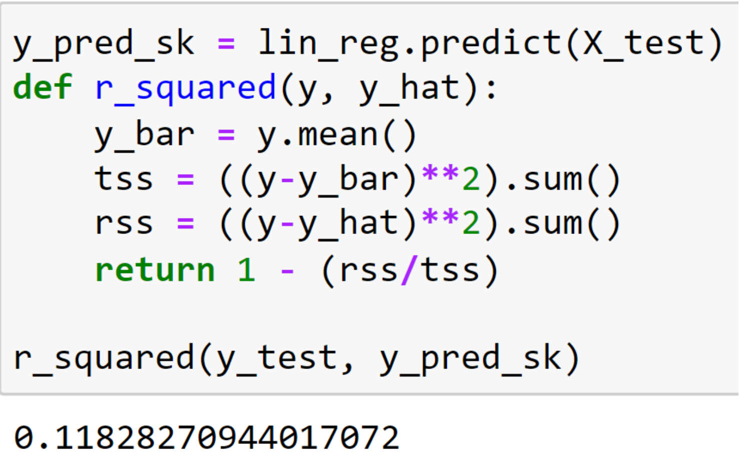
Once we have created our model, we can test its accuracy. We can do this with an R^2 value, which explains the proportion of variability of Y that can be explained by X. The R^2 is always between 0 and 1, if the value is closer to 1 then the regression explains a lot the variability of y, if it is close to zero then the regression does not explain that much variability of y.

The equation for R^2 is

$$R^2 = \frac{TSS - RSS}{TSS}$$

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

In python the code for calculating the R^2 value is



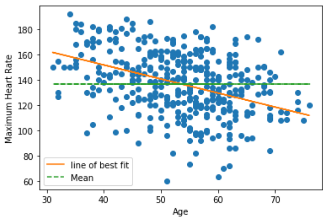
```
y_pred_sk = lin_reg.predict(X_test)
def r_squared(y, y_hat):
    y_bar = y.mean()
    tss = ((y-y_bar)**2).sum()
    rss = ((y-y_hat)**2).sum()
    return 1 - (rss/tss)

r_squared(y_test, y_pred_sk)
```

```
0.11828270944017072
```

This code in python is what was shown earlier with how to calculate the R^2 value. The result we attain when calculating the R^2 .1183, which means we can explain 11.83% of variance with our model. Most of the time in clinical datasets, around .1 is common for an R^2 value so this isn't too far off from what we would commonly expect.

These are the results for the other train/test splits we mentioned earlier:

60/40 Train Test Split



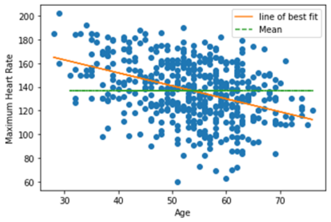
550 Training Values, 368 Test Values

$\widehat{B_0} = 196.2167823$

$\widehat{B_1} = -1.1098657$

$R^2 = 0.11147472075203857$

50/50 Train Test Split



459 Training Values, 459 Test Values

$\widehat{B_0} = 195.61381404$

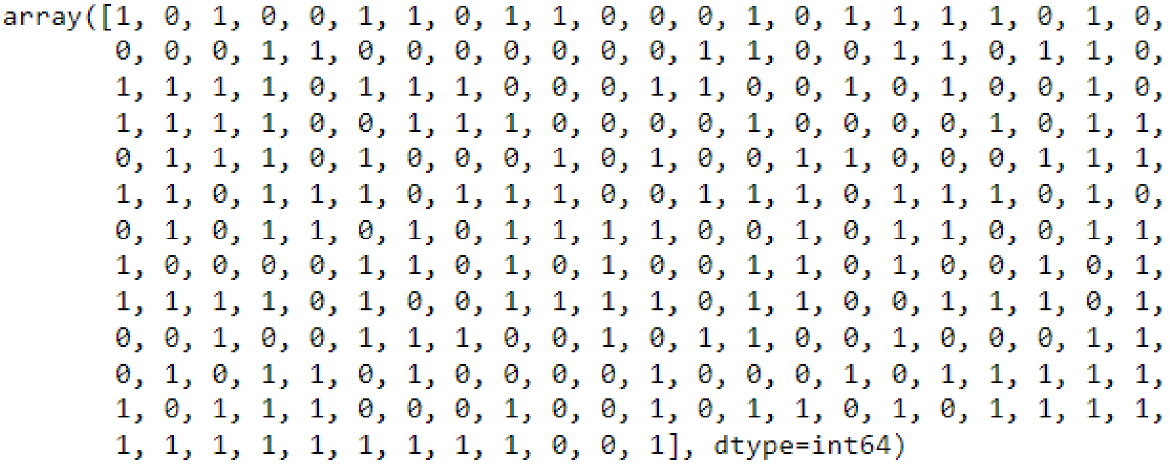
$\widehat{B_1} = -1.09495967$

$R^2 = 0.12606432732910777$

Logistic Regression Python Model:

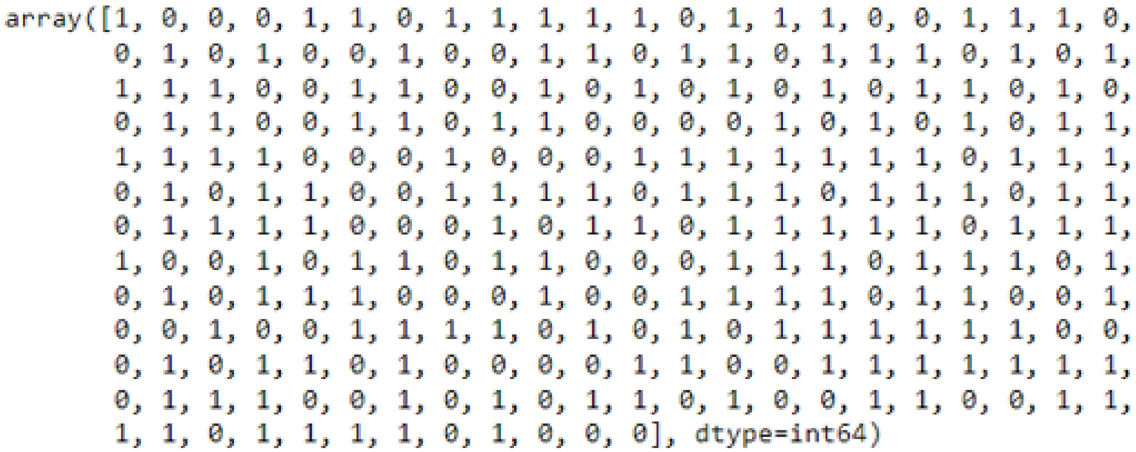
Like we have previously discussed, the output of a logistic regression model is the probability that given a value X, the response value Y is a success. With a success being labeled put into the category 1. In python, we can set our model to label the predictions for us. So, if a probability of success is over .5, we label the predicted value as a 1, and if the probability of success is below .5, we label the predicted values as a 0.

For our training dataset, the actual observed values are



```
array([1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0,
       1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0,
       1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1,
       0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1,
       1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1,
       1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1,
       0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1,
       0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1], dtype=int64)
```

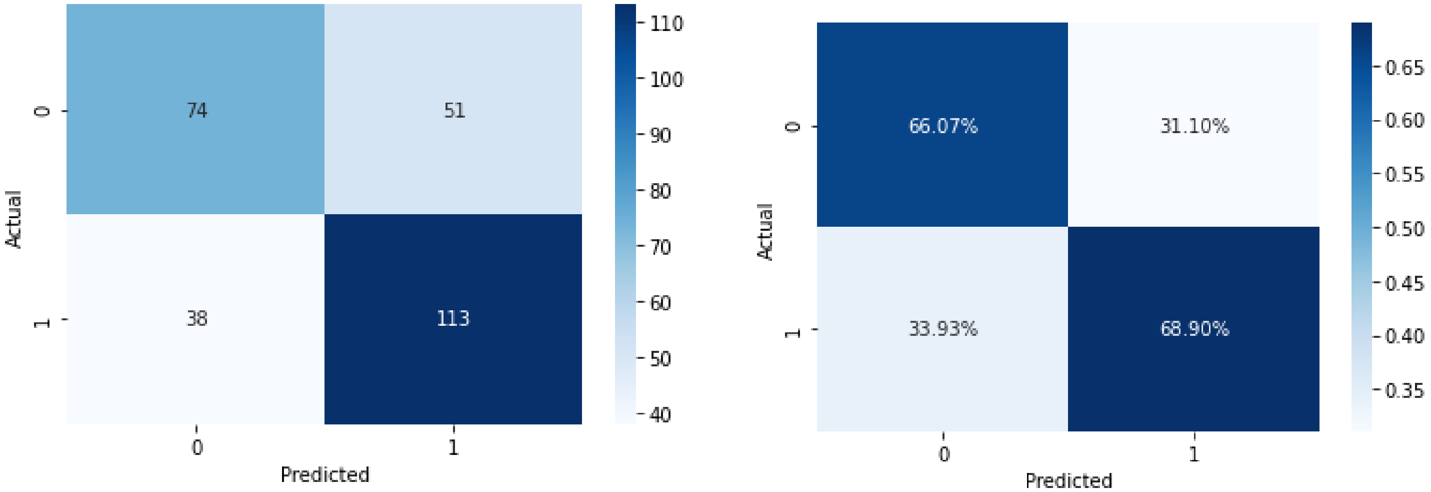
When we create our model, the predicted values of Y based on the observed values of X give us the results:



```
array([1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0,
       0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1,
       1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0,
       0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1,
       1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1,
       0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1,
       0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0,
       0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0], dtype=int64)
```

We can use a confusion matrix to observe the differences between our predicted categories and the actual observed values.

For a 70/30 train/test split we get the confusion matrices

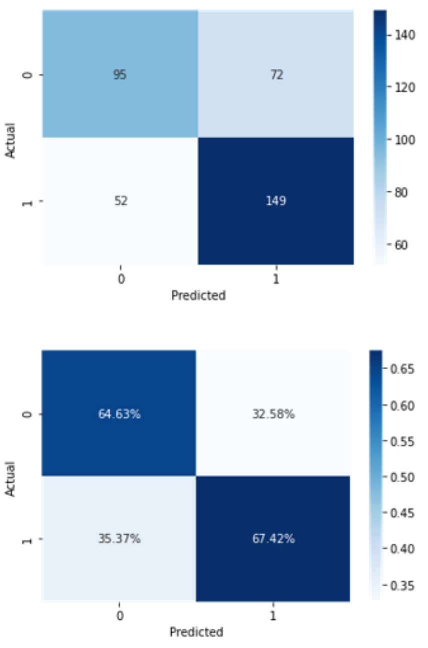


The matrix on the left has the number of observed/predicted values, while the matrix on the right has the percent of observed/predicted values. For the matrix on the right, the top left quadrant is when the observed value is 0 and the predicted value is 0. Which in our data, means when the patient does not have heart disease and when we predicted that the patient does not have heart disease. The top right quadrant is when the observed is 0 and the predicted is 1. This is when the patient does not have heart disease, but we predicted the patient does have heart disease. This is also known as a false positive. The bottom left quadrant is when the observed value is 1 and the predicted value is 0. In our data this means the patient does have heart disease, but we predicted the patient does not have heart disease. This type of error is known as a false negative. And lastly, the bottom right quadrant is when the observed value is 1 and the predicted value is 1. So, the patient does have heart disease and we predicted the patient does have heart disease.

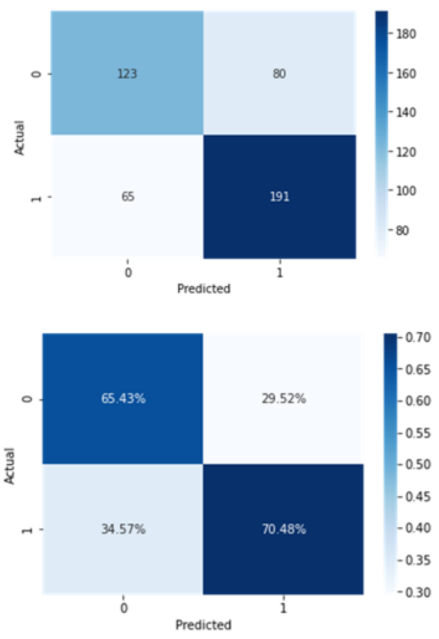
When dealing with clinical data, the implications of a false negative can be severe. With logistic regression, the cut off point we used was a probability that a success what above .5. If we were to make the probability of success a little more lenient, we could reduce the number of false negatives and in turn increasing the number of false positives. An example of this could be covid tests. It is more common to have a false positive than a false negative. This results in more positive cases being caught at the expense of more false positives. Depending on the application the cut off point could be adjusted to fit your needs.

These are the examples of 60/40 and 50/50 train test splits for logistic regression.

60/40 Train Test Split



50/50 Train Test Split



Conclusion:

In the field of statistical programming and clinical research linear and logistic regression models are powerful tools that are used repeatedly. In this project we broke down the lifecycle of the entire process, from getting the data, clearing the assumptions, and interpreting machine learning models.

References:

Beers, B. (2022, July 8). *What P-value tells us*. Investopedia. Retrieved July 29, 2022, from https://www.investopedia.com/terms/p/p-value.asp

Covariance. Corporate Finance Institute. (2022, January 24). Retrieved July 28, 2022, from https://corporatefinanceinstitute.com/resources/knowledge/finance/covariance/

Crowson, M. (2021, March 30). *Testing linearity in the logit using the box-tidwell transformation in SPSS (part 1 of 2)*. YouTube. Retrieved July 28, 2022, from https://www.youtube.com/watch?v=sciPFNcYqi8&ab_channel=MikeCrowson

Galarnyk, M. (2019, November 5). *Explaining the 68-95-99.7 rule for a normal distribution*. Medium. Retrieved July 28, 2022, from https://towardsdatascience.com/understanding-the-68-95-99-7-rule-for-a-normal-distribution-b7b7cbf760c2

Glen, S. (2020, December 16). *Variance inflation factor*. Statistics How To. Retrieved July 28, 2022, from https://www.statisticshowto.com/variance-inflation-factor/

Glen, S. (2021, June 1). *Degrees of freedom: What are they?* Statistics How To. Retrieved July 28, 2022, from https://www.statisticshowto.com/probability-and-statistics/hypothesis-testing/degrees-of-freedom/

Glen, S. (2021, September 30). *T-distribution / student's t: Definition, step by step articles, video*. Statistics How To. Retrieved July 31, 2022, from https://www.statisticshowto.com/probability-and-statistics/t-distribution/

Glen, S. (2022, February 23). *Skewed distribution: Definition, examples*. Statistics How To. Retrieved July 28, 2022, from https://www.statisticshowto.com/probability-and-statistics/skewed-distribution/

Grubber, J. (2019). (rep.). *The Thorn in My Side!! Logistic Regression Continuous Variables that Violate the Assumption of Linearity on the Log-odds (Logit) Scale: How to Identify and What to Do?* (pp. 1–4). Savannah, Georgia: Southeast SAS Users Group.

Hayes, A. (2022, July 8). *What is correlation in finance?* Investopedia. Retrieved July 28, 2022, from https://www.investopedia.com/terms/c/correlation.asp

jbstatistics. (2012, November 4). *An introduction to the F distribution*. YouTube. Retrieved July 29, 2022, from https://www.youtube.com/watch?v=G_RDxAZJ-ug&ab_channel=zedstatistics

jbstatistics. (2013, November 21). *An introduction to the chi-square distribution*. YouTube. Retrieved July 28, 2022, from https://www.youtube.com/watch?v=hcDb12fsbBU

King, A., & Eckersley, R. (2019). *Wilk test*. Wilk Test - an overview | ScienceDirect Topics. Retrieved July 28, 2022, from https://www.sciencedirect.com/topics/mathematics/wilk-test#:~:text=The%20Shapiro%E2%80%93Wilk%20test%20statistic,1%20being%20a%20perfect%20match.

Manish. (2020, June 26). *Logistic regression R: Introduction to logistic regression*. Analytics Vidhya. Retrieved July 28, 2022, from https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-r/#:~:text=log(p%2F1%2Dp,association%20in%20a%20linear%20way.&text=This%20is%20the%20equation%20used%20in%20Logistic%20Regression.

National Institute of Standards and Technology. (2019). *Bartlett's Test*. NIST: Engineering Statistics Handbook. Retrieved July 28, 2022, from https://www.itl.nist.gov/div898/handbook/eda/section3/eda357.htm#:~:text=The%20Bartlett%20test%20statistic%20is,for%20at%20least%20two%20groups.&text=where%20%5Cchi%5E2_%7B1,a%20significance%20level%20of%20%CE%B1.

National Institute of Standards and Technology. (2020). *Levene's Test for Equality of Variances*. NIST: Engineering Statistics Handbook. Retrieved July 28, 2022, from https://www.itl.nist.gov/div898/handbook/eda/section3/eda35a.htm#:~:text=Levene's%20test%20(%20Levene%201960)%20is,used%20to%20verify%20that%20assumption.

The Pennsylvania State University. (n.d.). *9.4 - Studentized Residuals*. 9.4 - Studentized Residuals | STAT 462. Retrieved July 31, 2022, from https://online.stat.psu.edu/stat462/node/247/

Pierce, R. (2021). *Correlation*. Math is Fun. Retrieved July 28, 2022, from https://www.mathsisfun.com/data/correlation.html.

Potter, C. (2022, July 26). *Variance inflation factor (VIF)*. Investopedia. Retrieved July 28, 2022, from https://www.investopedia.com/terms/v/variance-inflation-factor.asp

Schreiber-Gregory, D. N., & Bader, K. (2018). *Logistic and linear regression assumptions: Violation recognition and control*. LexJansen. Retrieved July 28, 2022, from https://proceedings.wuss.org/2018/130_Final_Paper_PDF.pdf

Spiegel, M. R., Srinivasan, R. A., & Schiller, J. J. (2013). *Schaum's outline of probability and statistics* (Vol. 4). Schaum.

Starmer, J. (2017, July 24). *Linear regression, clearly explained!!!* YouTube. Retrieved July 28, 2022, from https://www.youtube.com/watch?v=nk2CQITm_eo

Starmer, J. (2018, June 4). *Logistic regression details PT1: Coefficients*. YouTube. Retrieved July 28, 2022, from https://www.youtube.com/watch?v=vN5cNN2-HWE&ab_channel=StatQuestwithJoshStarmer

Starmer, J. (2018, March 5). *StatQuest: Logistic regression*. YouTube. Retrieved July 28, 2022, from https://www.youtube.com/watch?v=yIYKR4sgzI8

Weisstein, E. (n.d.). *Vector norm*. From Wolfram MathWorld. Retrieved July 28, 2022, from https://mathworld.wolfram.com/VectorNorm.html

Wu, S. (2021, June 5). *Multi-collinearity in regression*. Medium. Retrieved July 28, 2022, from https://towardsdatascience.com/multi-collinearity-in-regression-fe7a2c1467ea#:~:text=The%20second%20method%20to%20check,this%20variable%20and%20the%20rest.

YouTube. (2021, July 18). *Residuals, standardized residuals, and studentized residuals*. YouTube. Retrieved July 28, 2022, from https://www.youtube.com/watch?v=y4hRD7EWdJ4

Appendix

SAS Code:

```
DM 'OUTPUT; CLEAR; LOG; CLEAR;';
/* Importing csv file */
proc import datafile = "U:\Kyle's Docs\Final Project\heart1.csv"
out = work.heart
dbms = CSV;
run;

data heart1;
```

```
        set heart;
        log_age=log(age);
        log_restingbp=log(restingbp);
        log_maxhr=log(maxhr);
        log_cholesterol=log(cholesterol);
    run;

    /*Box Tidwell*/
    proc logistic data=heart1 descending;
        model heartdisease=age age*log_age cholesterol
cholesterol*log_cholesterol maxhr maxhr*log_maxhr restingbp
restingbp*log_restingbp;
    run;

    /*Shapiro-Wilk Test & Skewness & Kurtosis*/
    proc reg data=heart;
        model MaxHR=age / stb clb;
        output out=stdres p= predict r = resid;
    run;

    proc univariate data=stdres normal;
        var resid;
    run;

    /***** Testing for Homogeneity of Variance *****/

    /* Testing for Homogeneity of Variance - Levene's Test */
    proc glm data=heart;
        class groupid;
```

```
            model heartdisease = groupid;
            means groupid / hovtest=levene; /* can specify type=abs|square */
      run;
      quit;

      /* Multicolinearity */
      proc corr data=heart;
            var age restingbp cholesterol fastingbs maxhr sex_n st_slope_n
exerciseangina_n chestpaintype_n restingecg_n
                    oldpeak heartdisease;
            title 'Health Predictors - Examination of Correlation Matrix';
      run;

      proc reg data=heart;
            model heartdisease = age restingbp cholesterol fastingbs maxhr
oldpeak sex_n st_slope_n exerciseangina_n chestpaintype_n restingecg_n / vif
tol collin;
            title 'Health Predictors - Multicollinearity Investigation of VIF
and Tol';
      run;

      /* Studentized residuals - Check Outliers*/
      ods graphics on;
      proc reg data=heart;
            model heartdisease = age / stb clb;
            output out=stdres p= predict r = resid rstudent=r h=lev
cookd=cookd dffits=dffit;
      run;
      quit;
      ods graphics off;
      /* Print only those observations having absolute value of studentized
residual greater than 3*/
      proc print data=stdres;
            var age restingbp cholesterol fastingbs maxhr sex_n st_slope_n
exerciseangina_n chestpaintype_n restingecg_n
                    oldpeak heartdisease;
            title 'Outliers - Studentized Residual Approach';
            where abs(r)>=3;
      run;
```

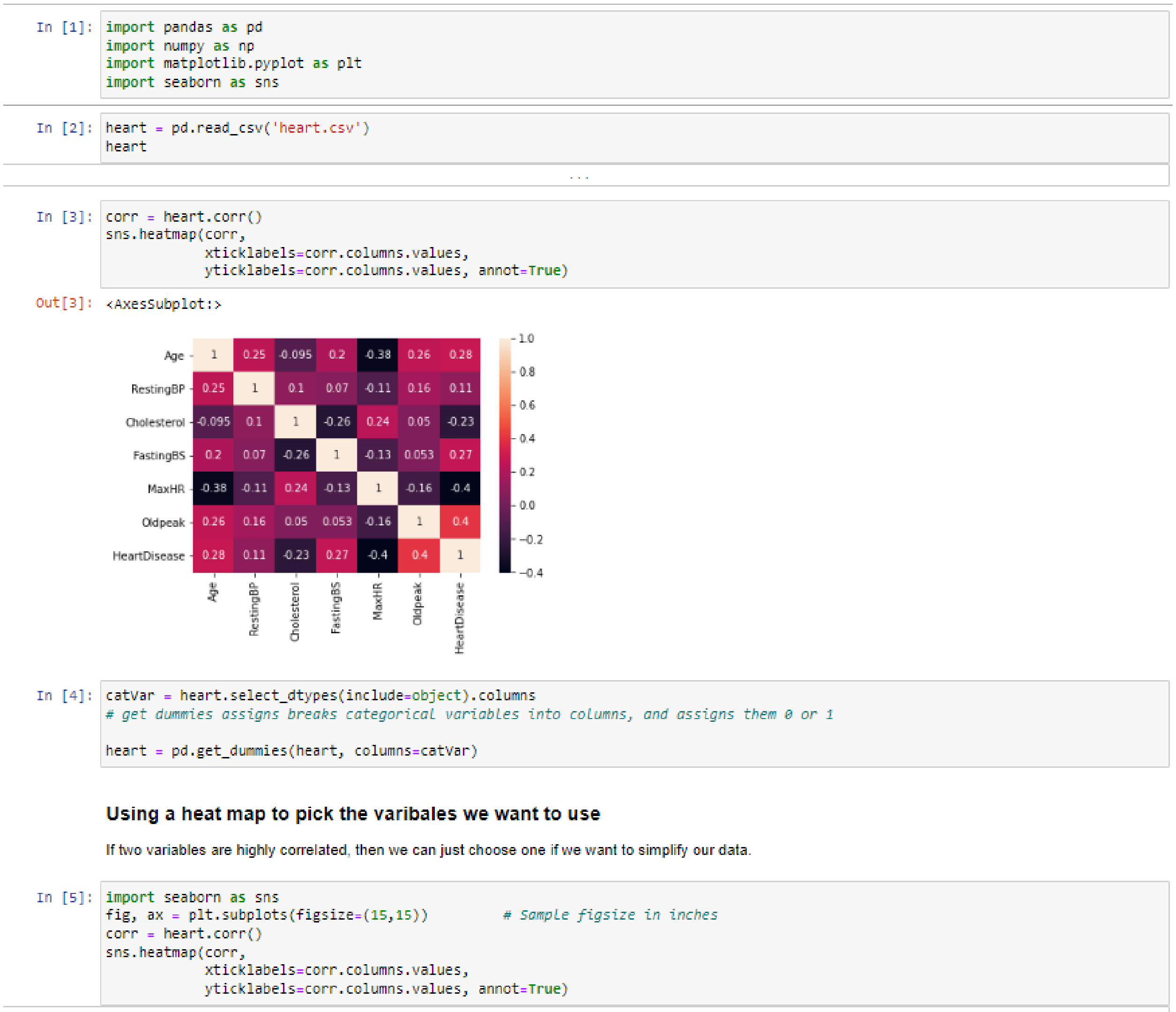
R Code:

```
library(corrplot)


data<-read.csv("heart1.csv")
head(data)
data = subset(data, select = -c(ChestPainType,ExerciseAngina,ST_Slope,Sex,RestingECG) )
data.cor<-cor(data)
head(round(data.cor, 2))
print(data.cor)
corrplot(data.cor)


attach(data)
Best.lm<-lm(MaxHR~Age)
print(Best.lm)
```

```
Residuals<-Best.lm$residuals

Fitted_value<-Best.lm$fitted.values #y-hat

qqnorm(Residuals)

qqline(Residuals)

hist(Residuals, freq = FALSE)

points(density(Residuals),type="l")

par(mfrow=c(2,2))

plot(Residuals~Age)

abline(h=0, col="red")

abline(h=0,col="red")


plot(Residuals~Fitted_value)

abline(h=0,col="coral")


ord<-seq(1,918,1)

plot(Residuals~ord)


qqnorm(Residuals)

qqline(Residuals)
```

Python Code:

Python Code:



```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
```

```
In [2]: heart = pd.read_csv('heart.csv')
        heart
```

...

```
In [3]: corr = heart.corr()
        sns.heatmap(corr,
                    xticklabels=corr.columns.values,
                    yticklabels=corr.columns.values, annot=True)
```

```
Out[3]: <AxesSubplot:>
```

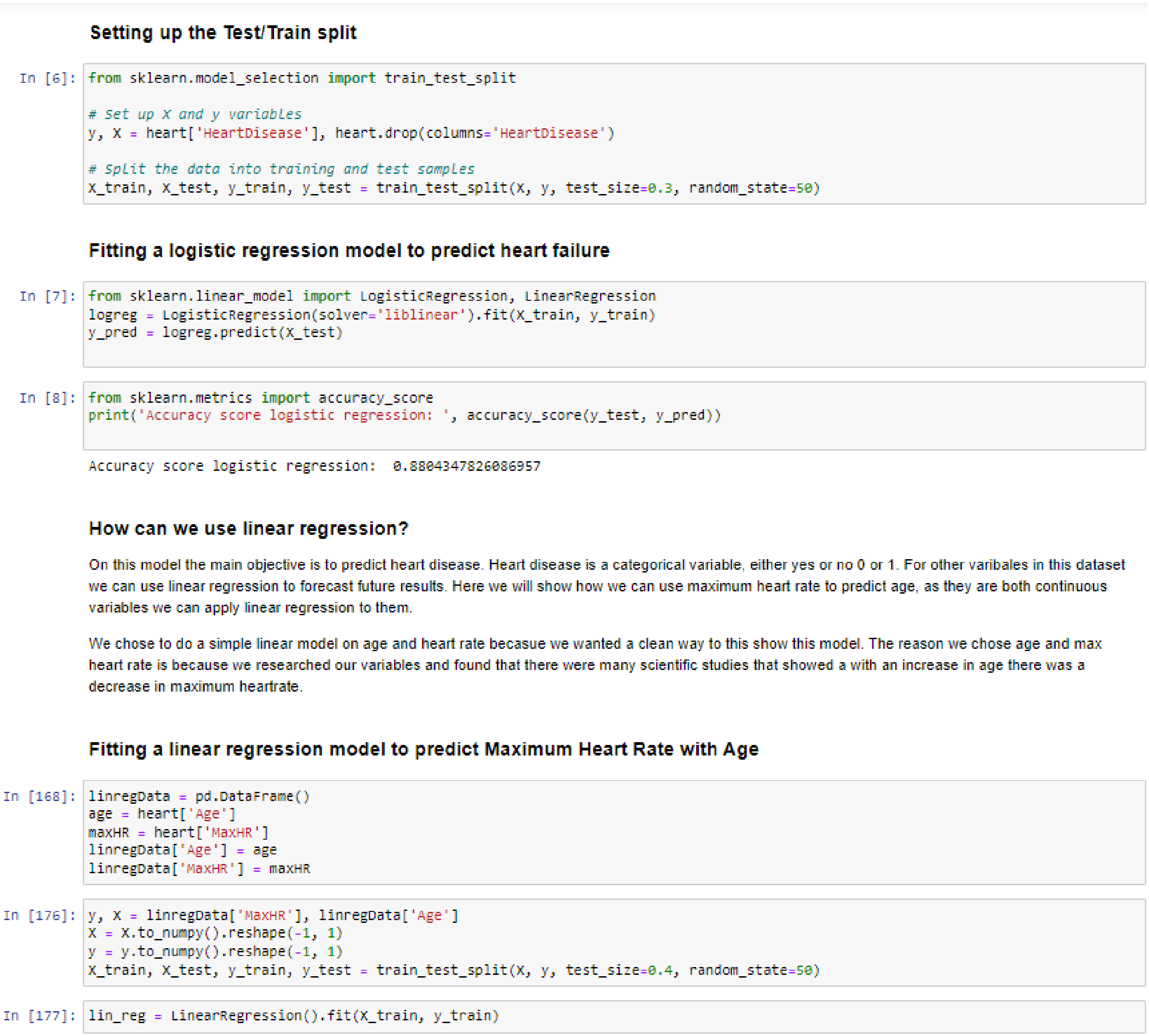
	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
Age	1	0.25	-0.095	0.2	-0.38	0.26	0.28
RestingBP	0.25	1	0.1	0.07	-0.11	0.16	0.11
Cholesterol	-0.095	0.1	1	-0.26	0.24	0.05	-0.23
FastingBS	0.2	0.07	-0.26	1	-0.13	0.053	0.27
MaxHR	-0.38	-0.11	0.24	-0.13	1	-0.16	-0.4
Oldpeak	0.26	0.16	0.05	0.053	-0.16	1	0.4
HeartDisease	0.28	0.11	-0.23	0.27	-0.4	0.4	1

```
In [4]: catVar = heart.select_dtypes(include=object).columns
        # get dummies assigns breaks categorical variables into columns, and assigns them 0 or 1

        heart = pd.get_dummies(heart, columns=catVar)
```

Using a heat map to pick the varibales we want to use

If two variables are highly correlated, then we can just choose one if we want to simplify our data.

```
In [5]: import seaborn as sns
        fig, ax = plt.subplots(figsize=(15,15))         # Sample figsize in inches
        corr = heart.corr()
        sns.heatmap(corr,
                    xticklabels=corr.columns.values,
                    yticklabels=corr.columns.values, annot=True)
```



Setting up the Test/Train split

In [6]:
```
from sklearn.model_selection import train_test_split

# Set up X and y variables
y, X = heart['HeartDisease'], heart.drop(columns='HeartDisease')

# Split the data into training and test samples
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=50)
```

Fitting a logistic regression model to predict heart failure

In [7]:
```
from sklearn.linear_model import LogisticRegression, LinearRegression
logreg = LogisticRegression(solver='liblinear').fit(X_train, y_train)
y_pred = logreg.predict(X_test)
```

In [8]:
```
from sklearn.metrics import accuracy_score
print('Accuracy score logistic regression: ', accuracy_score(y_test, y_pred))
```

```
Accuracy score logistic regression:  0.8804347826086957
```

How can we use linear regression?

On this model the main objective is to predict heart disease. Heart disease is a categorical variable, either yes or no 0 or 1. For other varibales in this dataset we can use linear regression to forecast future results. Here we will show how we can use maximum heart rate to predict age, as they are both continuous variables we can apply linear regression to them.

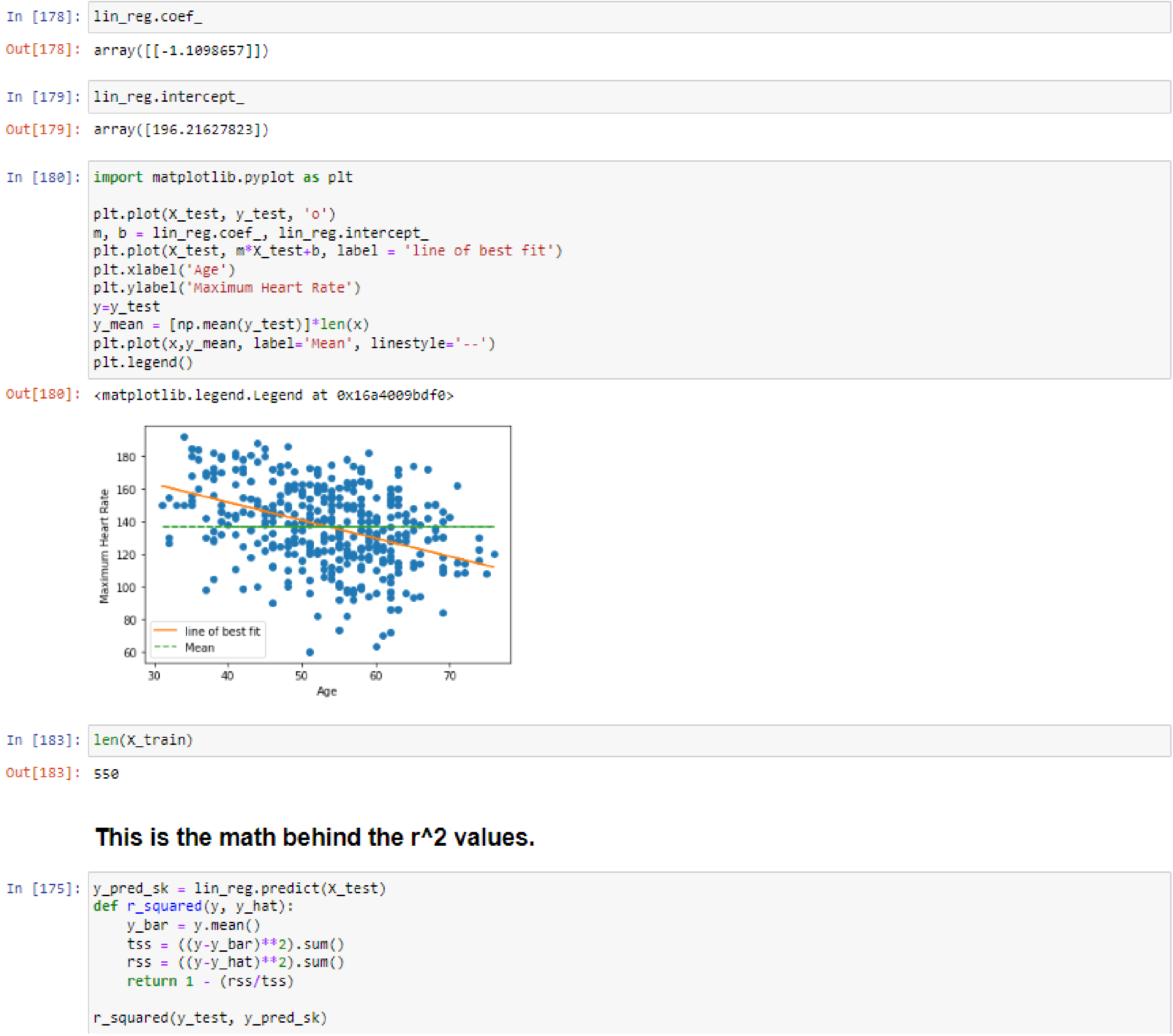
We chose to do a simple linear model on age and heart rate becasue we wanted a clean way to this show this model. The reason we chose age and max heart rate is because we researched our variables and found that there were many scientific studies that showed a with an increase in age there was a decrease in maximum heartrate.

Fitting a linear regression model to predict Maximum Heart Rate with Age

In [168]:
```
linregData = pd.DataFrame()
age = heart['Age']
maxHR = heart['MaxHR']
linregData['Age'] = age
linregData['MaxHR'] = maxHR
```

In [176]:
```
y, X = linregData['MaxHR'], linregData['Age']
X = X.to_numpy().reshape(-1, 1)
y = y.to_numpy().reshape(-1, 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=50)
```

In [177]:
```
lin_reg = LinearRegression().fit(X_train, y_train)
```

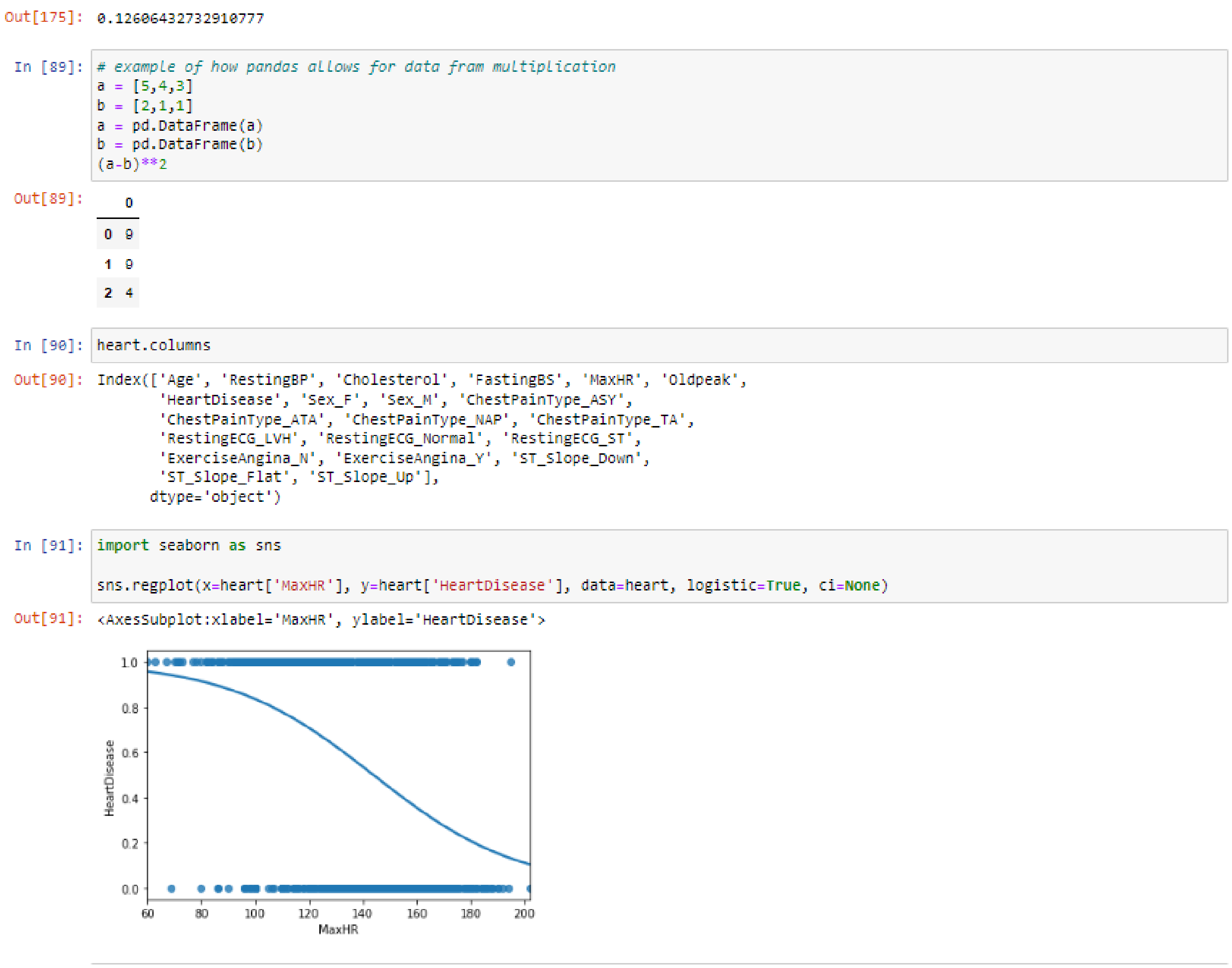


```
In [178]: lin_reg.coef_

Out[178]: array([[-1.1098657]])

In [179]: lin_reg.intercept_

Out[179]: array([196.21627823])

In [180]: import matplotlib.pyplot as plt

          plt.plot(X_test, y_test, 'o')
          m, b = lin_reg.coef_, lin_reg.intercept_
          plt.plot(X_test, m*X_test+b, label = 'line of best fit')
          plt.xlabel('Age')
          plt.ylabel('Maximum Heart Rate')
          y=y_test
          y_mean = [np.mean(y_test)]*len(x)
          plt.plot(x,y_mean, label='Mean', linestyle='--')
          plt.legend()

Out[180]: <matplotlib.legend.Legend at 0x16a4009bdf0>
```

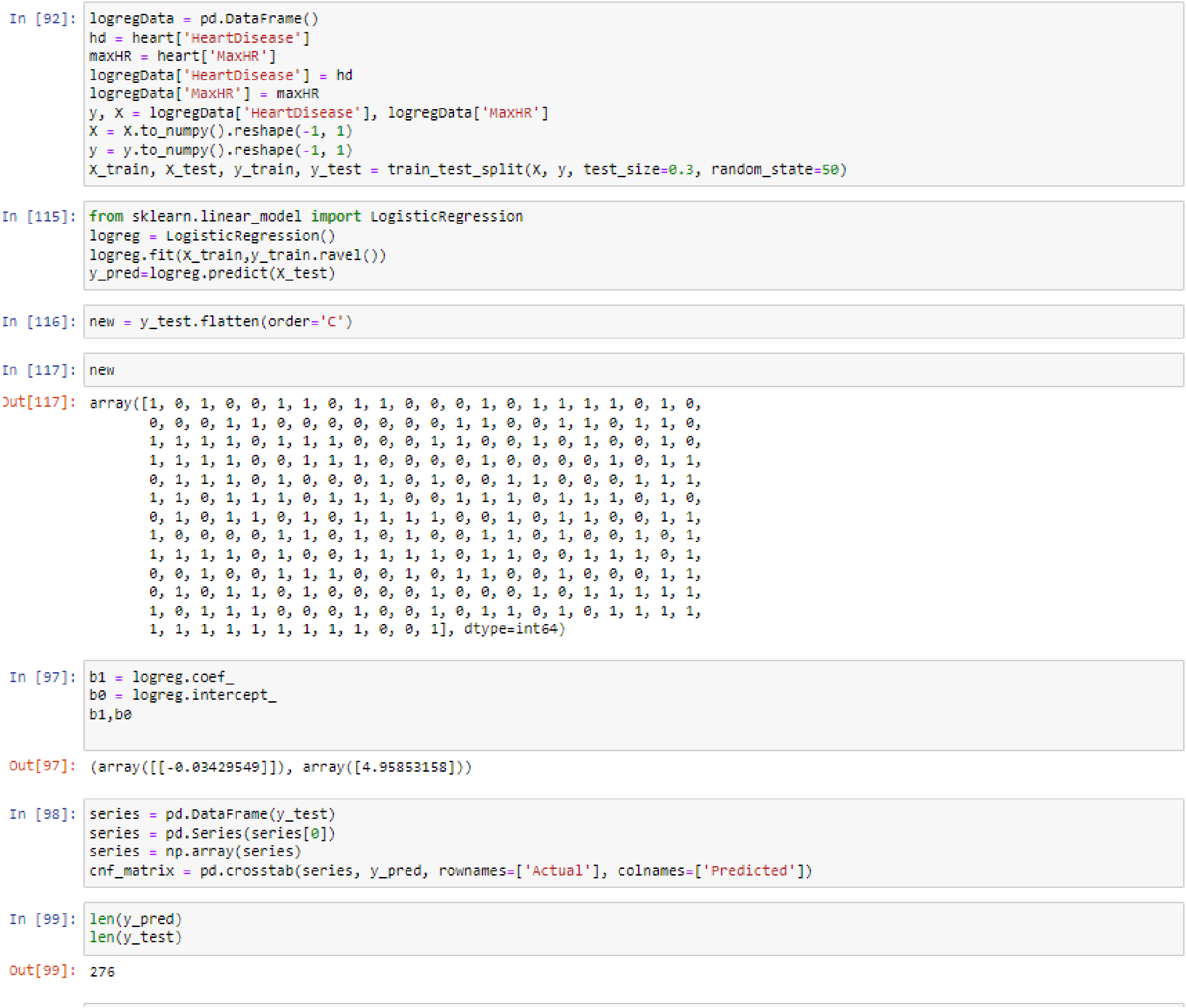
```
In [183]: len(X_train)

Out[183]: 550
```

This is the math behind the r^2 values.

```
In [175]: y_pred_sk = lin_reg.predict(X_test)
          def r_squared(y, y_hat):
              y_bar = y.mean()
              tss = ((y-y_bar)**2).sum()
              rss = ((y-y_hat)**2).sum()
              return 1 - (rss/tss)

          r_squared(y_test, y_pred_sk)
```



```
Out[175]: 0.12606432732910777

 In [89]: # example of how pandas allows for data fram multiplication
          a = [5,4,3]
          b = [2,1,1]
          a = pd.DataFrame(a)
          b = pd.DataFrame(b)
          (a-b)**2

 Out[89]:    0
          0  9
          1  9
          2  4

 In [90]: heart.columns

 Out[90]: Index(['Age', 'RestingBP', 'Cholesterol', 'FastingBS', 'MaxHR', 'Oldpeak',
                 'HeartDisease', 'Sex_F', 'Sex_M', 'ChestPainType_ASY',
                 'ChestPainType_ATA', 'ChestPainType_NAP', 'ChestPainType_TA',
                 'RestingECG_LVH', 'RestingECG_Normal', 'RestingECG_ST',
                 'ExerciseAngina_N', 'ExerciseAngina_Y', 'ST_Slope_Down',
                 'ST_Slope_Flat', 'ST_Slope_Up'],
                dtype='object')

 In [91]: import seaborn as sns

          sns.regplot(x=heart['MaxHR'], y=heart['HeartDisease'], data=heart, logistic=True, ci=None)

 Out[91]: <AxesSubplot:xlabel='MaxHR', ylabel='HeartDisease'>
```



```
In [92]: logregData = pd.DataFrame()
         hd = heart['HeartDisease']
         maxHR = heart['MaxHR']
         logregData['HeartDisease'] = hd
         logregData['MaxHR'] = maxHR
         y, X = logregData['HeartDisease'], logregData['MaxHR']
         X = X.to_numpy().reshape(-1, 1)
         y = y.to_numpy().reshape(-1, 1)
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=50)
```

```
In [115]: from sklearn.linear_model import LogisticRegression
          logreg = LogisticRegression()
          logreg.fit(X_train,y_train.ravel())
          y_pred=logreg.predict(X_test)
```

```
In [116]: new = y_test.flatten(order='C')
```

```
In [117]: new
```

```
Out[117]: array([1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0,
                 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0,
                 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0,
                 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1,
                 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1,
                 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0,
                 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1,
                 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1,
                 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1,
                 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1,
                 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1,
                 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1], dtype=int64)
```

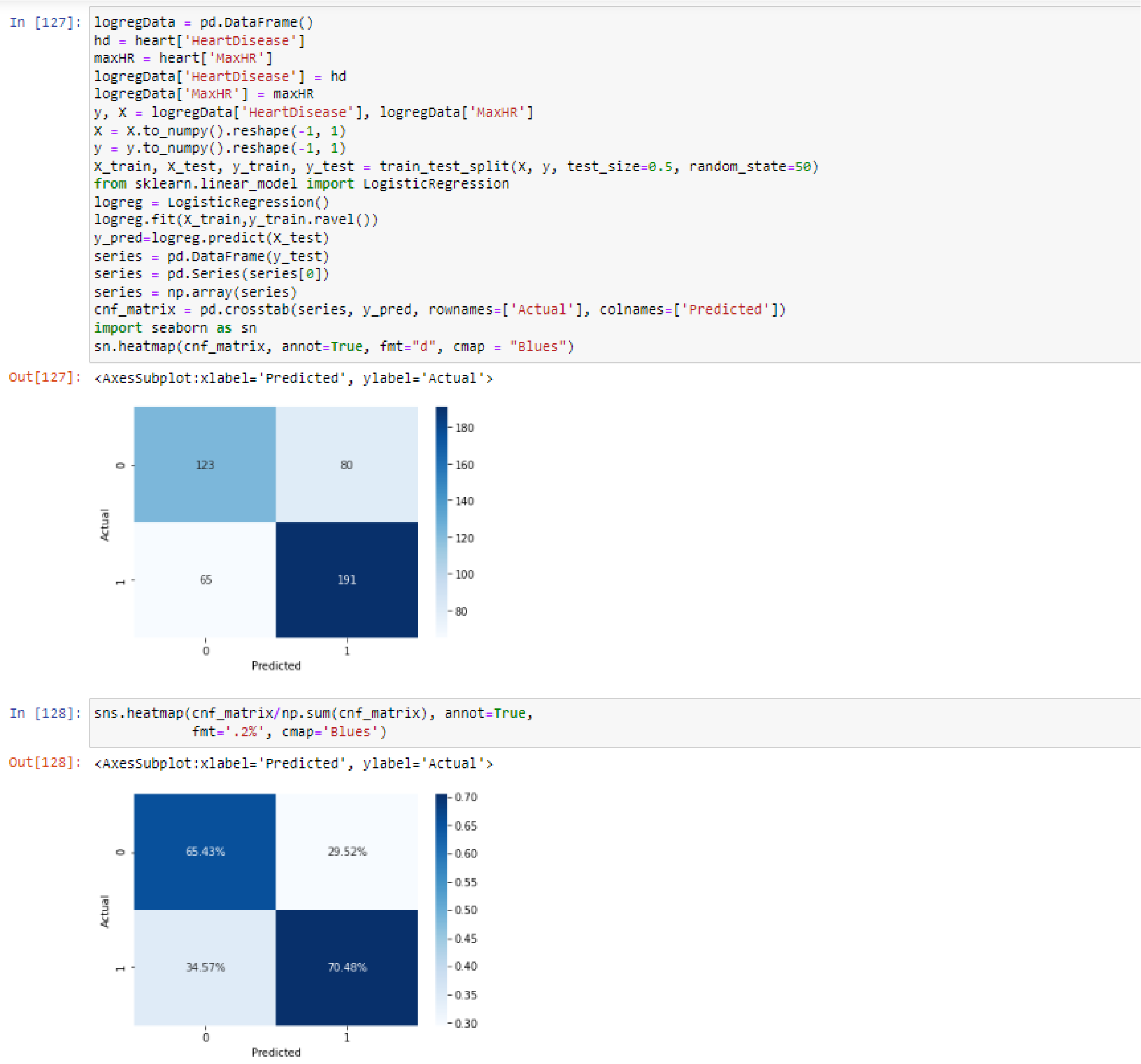
```
In [97]: b1 = logreg.coef_
         b0 = logreg.intercept_
         b1,b0
```

```
Out[97]: (array([[-0.03429549]]), array([4.95853158]))
```

```
In [98]: series = pd.DataFrame(y_test)
         series = pd.Series(series[0])
         series = np.array(series)
         cnf_matrix = pd.crosstab(series, y_pred, rownames=['Actual'], colnames=['Predicted'])
```

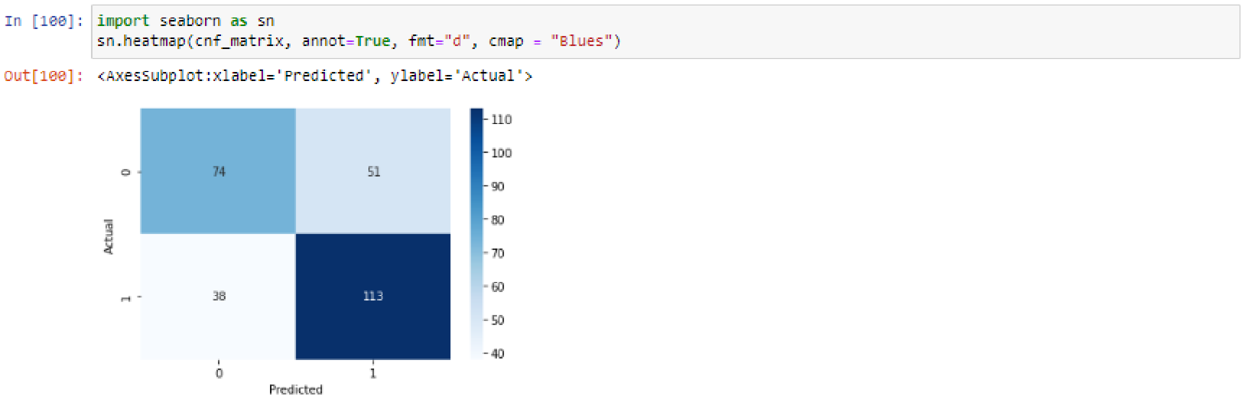
```
In [99]: len(y_pred)
         len(y_test)
```

```
Out[99]: 276
```

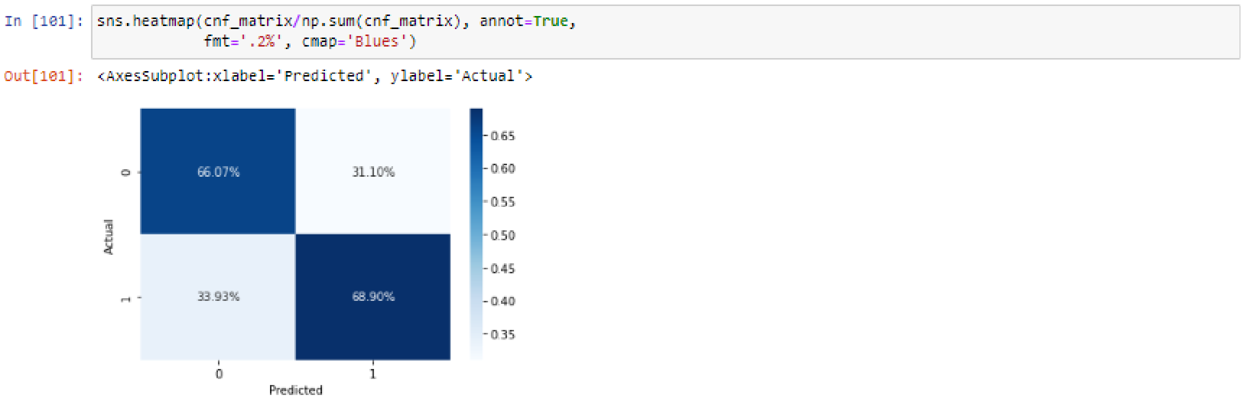


```
In [127]: logregData = pd.DataFrame()
          hd = heart['HeartDisease']
          maxHR = heart['MaxHR']
          logregData['HeartDisease'] = hd
          logregData['MaxHR'] = maxHR
          y, X = logregData['HeartDisease'], logregData['MaxHR']
          X = X.to_numpy().reshape(-1, 1)
          y = y.to_numpy().reshape(-1, 1)
          X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=50)
          from sklearn.linear_model import LogisticRegression
          logreg = LogisticRegression()
          logreg.fit(X_train,y_train.ravel())
          y_pred=logreg.predict(X_test)
          series = pd.DataFrame(y_test)
          series = pd.Series(series[0])
          series = np.array(series)
          cnf_matrix = pd.crosstab(series, y_pred, rownames=['Actual'], colnames=['Predicted'])
          import seaborn as sn
          sn.heatmap(cnf_matrix, annot=True, fmt="d", cmap = "Blues")

Out[127]: <AxesSubplot:xlabel='Predicted', ylabel='Actual'>
```

```
In [128]: sns.heatmap(cnf_matrix/np.sum(cnf_matrix), annot=True,
                      fmt='.2%', cmap='Blues')

Out[128]: <AxesSubplot:xlabel='Predicted', ylabel='Actual'>
```



```
In [100]: import seaborn as sn
          sn.heatmap(cnf_matrix, annot=True, fmt="d", cmap = "Blues")

Out[100]: <AxesSubplot:xlabel='Predicted', ylabel='Actual'>
```



```
In [101]: sns.heatmap(cnf_matrix/np.sum(cnf_matrix), annot=True,
                      fmt='.2%', cmap='Blues')

Out[101]: <AxesSubplot:xlabel='Predicted', ylabel='Actual'>
```