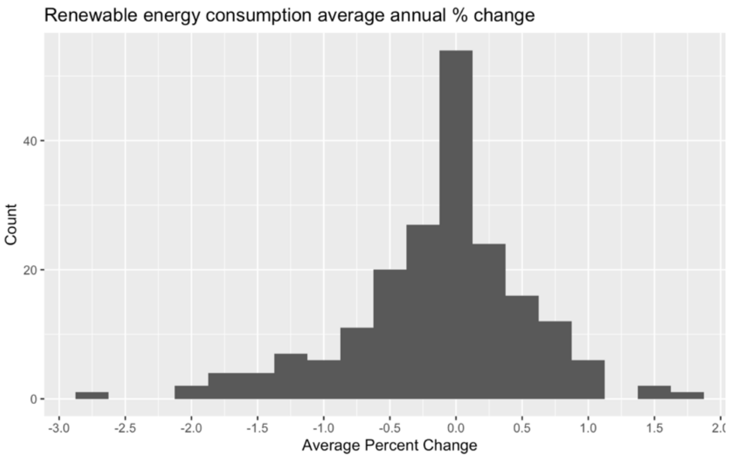
An Exploration into Logistic Regression: CO_2 Emissions as Predicted by Urban Population, Forest Area, and Renewable Energy Consumption

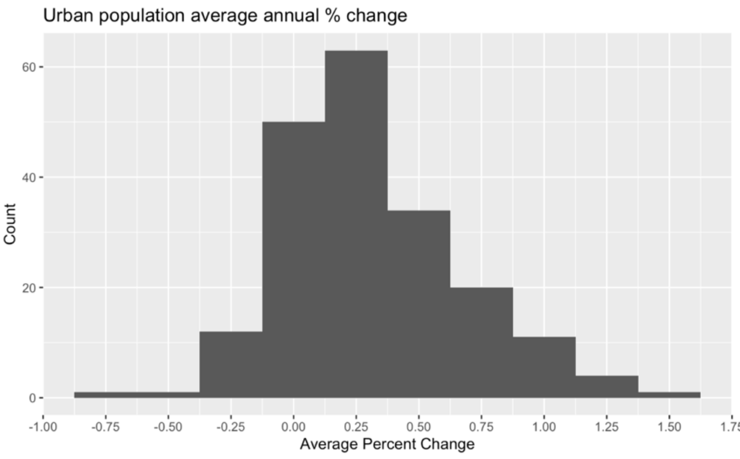
Carbon dioxide (CO_2) emissions heavily impact the current climate crisis. There are many factors that impact CO_2 emissions; in our research we explored three of them: percent of a population living in urban areas, percent of energy consumption using renewable resources, and percent of land area that is classified as forest.

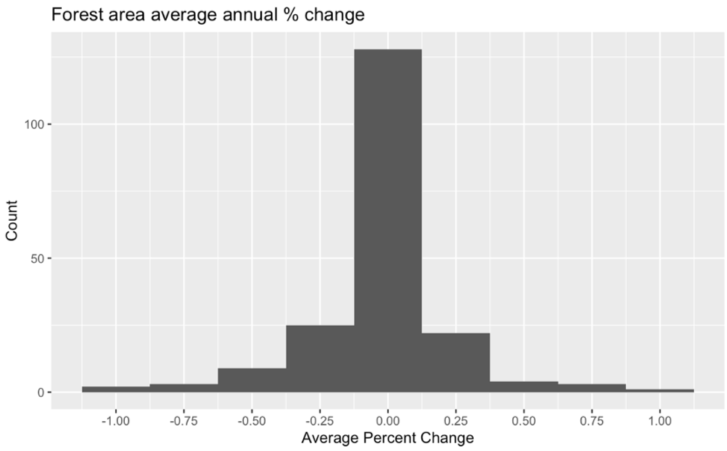
We used four datasets from the World Bank data set (*citation*): CO_2 Emissions (kt), Urban Population (% of total population), Renewable Energy Consumption (% of total final energy consumption), and Forest Area (% of land area). Data were reported for all countries and territories over years 1964-2019. The first step we took with the .csv file of our data was to remove records that were not associated with countries (eg- “low income”, and “South America”). Then we limited the data to the years 2001-2014; as these were the years that had complete data for our indicators, and we wanted the data to be relevant to the current time. We also removed 22 countries for which the data was incomplete.

Our research question is to explore the relationship between forest area, urban population, and renewable energy as predictors for CO_2 emissions. We anticipate that as forest area increases, we will see a smaller CO_2 output since forests clean CO_2 from the atmosphere and deforestation requires the burning of fossil fuels (a major contributor to CO_2 emissions). As renewable energy consumption increases, we expect to see a smaller CO_2 output as it replaces consumption of energy used by burning fossil fuels. As urban populations increase, we expect to see larger CO_2 emissions since urban populations tend to burn more fossil fuels. We hope to explore this relationship further and find a model for it.

To run logistic regression in R, we needed to prepare a final data frame with a column for each of our predictors and a column for CO_2 as a binary outcome. For urban population, renewable energy, and forest area, the data were reported as percentages for each year and each country. We first found the average annual percent change for each country for each of these datasets over our time frame. Below are three histograms showing count of countries vs average annual percentage change for each of our predictors.



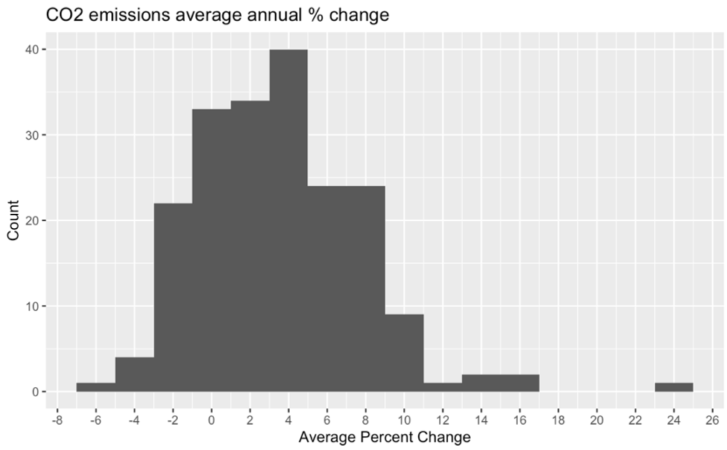


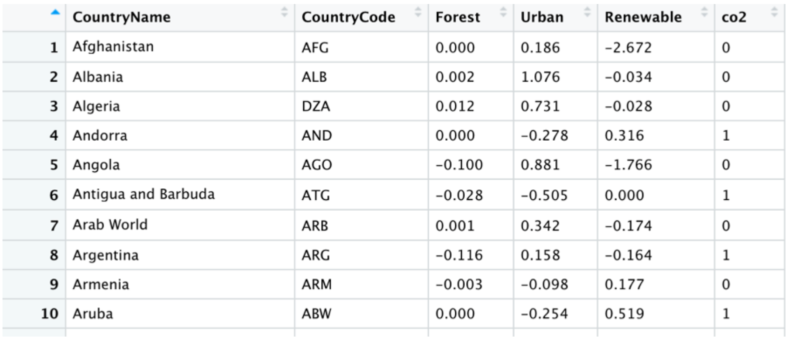


Reporting the CO_2 emissions as a binary outcome took a bit more manipulation. The CO_2 emission data was reported in kilotons. We took these steps:

1. Calculated average percentage change in CO_2 emissions for each country from 2001-2014.
2. Calculated the average of all the averages calculated in step-1: The overall average = 3.551%
3. We defined success for the model as low CO_2emissions. Countries with average annual percentage change in CO_2 emissions < 3.551% were given 1 as a binary outcome for the dependent variable of CO_2 emissions.
4. We defined failure for the model as high CO_2emissions. Countries with average annual percentage change in CO_2 emissions >= 3.551% were given 0 as a binary outcome for the dependent variable of CO_2 emissions.

The CO_2 histogram and the final dataframe are shown below.





	CountryName	CountryCode	Forest	Urban	Renewable	co2
1	Afghanistan	AFG	0.000	0.186	−2.672	0
2	Albania	ALB	0.002	1.076	−0.034	0
3	Algeria	DZA	0.012	0.731	−0.028	0
4	Andorra	AND	0.000	−0.278	0.316	1
5	Angola	AGO	−0.100	0.881	−1.766	0
6	Antigua and Barbuda	ATG	−0.028	−0.505	0.000	1
7	Arab World	ARB	0.001	0.342	−0.174	0
8	Argentina	ARG	−0.116	0.158	−0.164	1
9	Armenia	ARM	−0.003	−0.098	0.177	0
10	Aruba	ABW	0.000	−0.254	0.519	1

Complete dataframe

Before implementing logistic regression in R, we first needed to understand the math behind it. Logistic regression is closely related to a linear relationship; however, for a logistic regression model the output will be a probability, which, of course, must be restricted to the interval [0,1]. If we start with a linear model (simplified to one predictor variable):

$$y = \beta_0 + \beta_1 x$$

We can see that this will give us values outside of the range [0,1]. To address this, we can alter the equation to this:

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

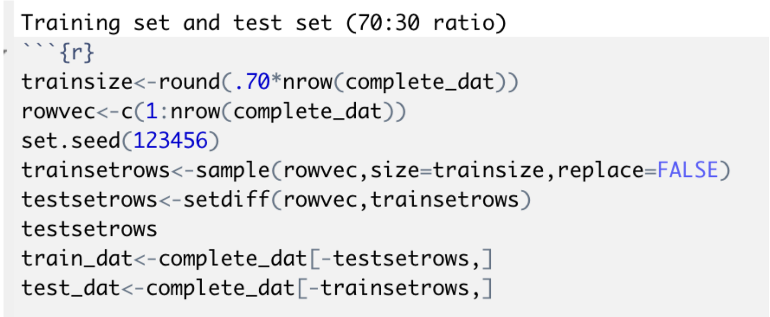
The exponential function ensures that values stay positive, and the fraction will keep the value at 1 or below. In this form, we can see that our output, p, will be in the form of a probability. This is what is graphed when we see a logistic plot, as we will show for our data later. If we put this formula back into the form of a linear model, we get what's called the logit (or log odds) function:

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

This will map probabilities back to the full range of real numbers. The coefficients in the logistic regression function are calculated using maximum likelihood. This is the same method that we used to calculate the probability of binomial distributions. This is what R does when the glm method is implemented, which we will discuss furthers.

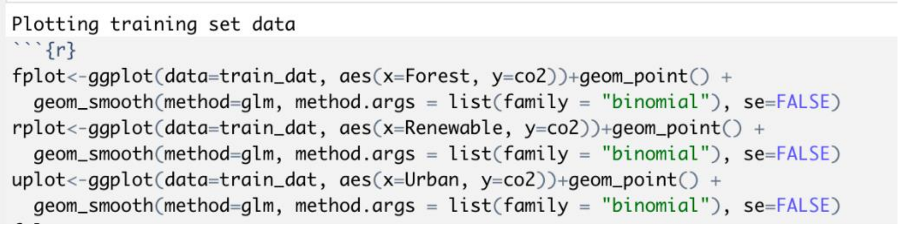
To run logistic regression, we first confirmed that our data satisfy the requirements for the method. The first requirement is to define the independent and dependent variables for our logistic model. The independent variables are used to predict the probability of a successful outcome. In our case the predictors are forest area, renewable energy usage and urban population. The independent variables should not be collinear or highly correlated because it impacts the performance of the model. Also, the high correlation between independent variables contributes to masking of coefficients for those particular predictors which directly impacts the predicted values of the model. The dependent variable is a categorical variable (having values as different categories as 0 or 1, true or false or more than 2 categories names). In our case, we performed binary classification, so the dependent variable CO_2 had values as 0 or 1 as the binary outcomes (categories). This was the first part for the data satisfaction requirements to define and evaluate our variables for the model.

Next, after we have the required condensed dataset with all the variables, we need to bifurcate it as training dataset and testing data set, further used for data modelling and model testing respectively. We took 70:30 ratio of the complete dataset as training dataset values and testing data set values respectively. We randomly selected 70% of the complete dataset as the training dataset for data modelling and rest 30% for testing the model. It is important to have a good mixture of 0s and 1s in the training dataset because it may impact the decision-making of the model. For our model, the ratio of 1s and 0s in the training set is 75:63, that signifies none of the binary outcome is a dominating outcome and our model's performance will not be impacted towards any of the binary outcome. The code we used is below:



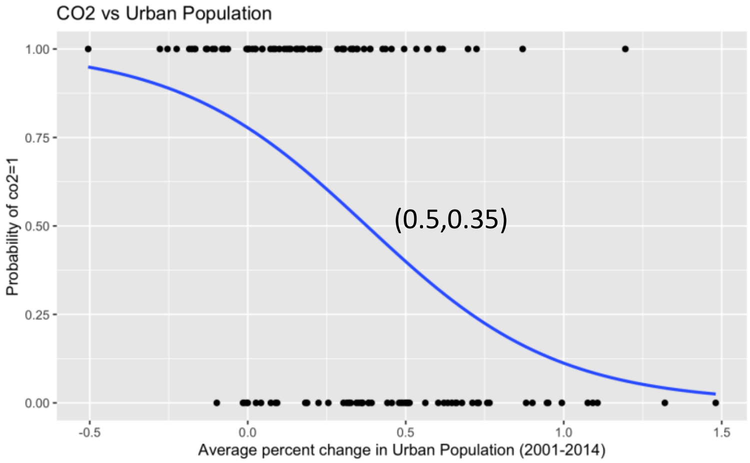
~~~
Training set and test set (70:30 ratio)
```{r}
trainsize<-round(.70*nrow(complete_dat))
rowvec<-c(1:nrow(complete_dat))
set.seed(123456)
trainsetrows<-sample(rowvec,size=trainsize,replace=FALSE)
testsetrows<-setdiff(rowvec,trainsetrows)
testsetrows
train_dat<-complete_dat[-testsetrows,]
test_dat<-complete_dat[-trainsetrows,]
~~~

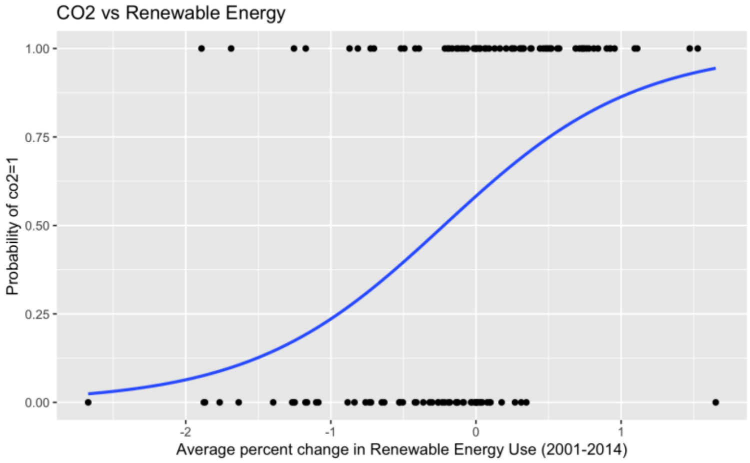
To begin exploring the relationship between each of our predictor variables and CO_2 emissions as a binary outcome, we plotted each variable as a logistic plot using geom_smooth as shown below.

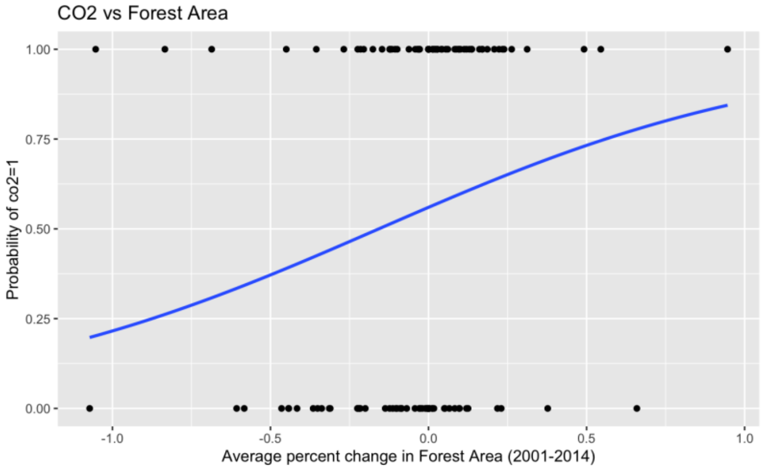


~~~
Plotting training set data
```{r}
fplot<-ggplot(data=train_dat, aes(x=Forest, y=co2))+geom_point() +
  geom_smooth(method=glm, method.args = list(family = "binomial"), se=FALSE)
rplot<-ggplot(data=train_dat, aes(x=Renewable, y=co2))+geom_point() +
  geom_smooth(method=glm, method.args = list(family = "binomial"), se=FALSE)
uplot<-ggplot(data=train_dat, aes(x=Urban, y=co2))+geom_point() +
  geom_smooth(method=glm, method.args = list(family = "binomial"), se=FALSE)
~~~

Below we have included each of these three plots







The interpretation of these graphs is as follows:

1) All the black solid points represents all the countries.

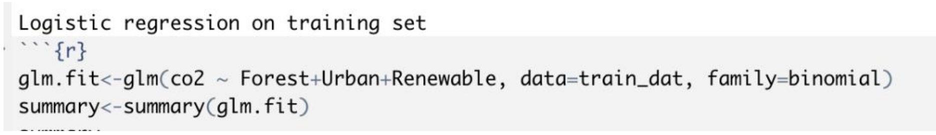
2) The y-axis represents the binary outcomes for CO_2 emissions i.e. 0 or 1. Also, it represents the predicted probability of low CO_2 emissions(success) for each country.

3) The x-axis represents the average annual percentage change in urban population for each country.

4) Any point on the curve can be interpreted in the following way: The point (0.5, 0.35) on the urban population graph means for a country with 0.5 average annual percentage change in urban population, it has 0.35 probability of having low CO_2 emissions.

We can see from these plots that our predictions hold: for urban population, we see more 1s for CO_2 where the urban population decreased (or increased more slowly). For forest area, we see more 1s when forest increased (or decreased less rapidly). For renewable energy consumption, we see more 1s when the consumption increased.

After satisfying all the data requirements for the method, we implemented logistic in R using the glm function as can be seen below:



~~~
Logistic regression on training set
```{r}
glm.fit<-glm(co2 ~ Forest+Urban+Renewable, data=train_dat, family=binomial)
summary<-summary(glm.fit)
~~~

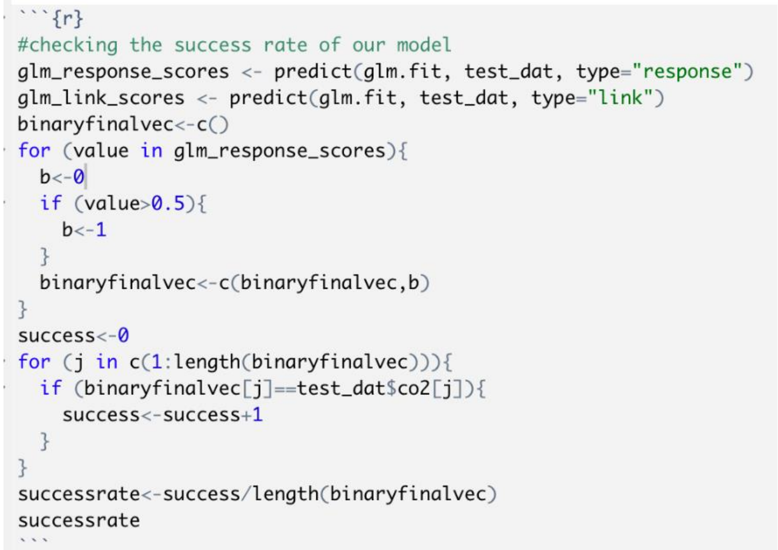
glm stands for generalized linear model. Logistic regression is a special case under linear modelling (as demonstrated above) This function requires three inputs:

1) Dependent ~ independent variables: in our case CO_2 emissions ~ Urban population, Forest area and Renewable Energy Usage

2) Training dataset (the 70% randomly selected data from the complete condensed dataset)

3) Family type – binomial (since we have a binary classification)

After model implementation, we first needed to test the accuracy of our model using the test data set. To do this, we used the predict function and the following code:

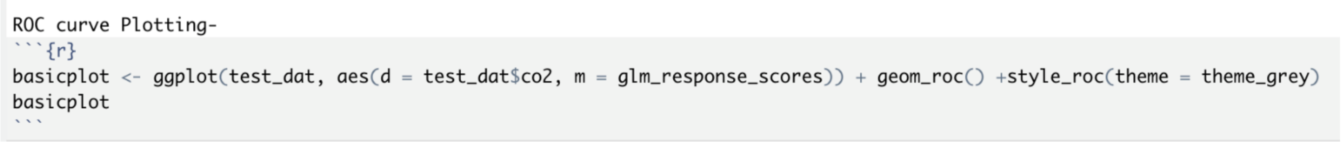


~~~
```{r}
#checking the success rate of our model
glm_response_scores <- predict(glm.fit, test_dat, type="response")
glm_link_scores <- predict(glm.fit, test_dat, type="link")
binaryfinalvec<-c()
for (value in glm_response_scores){
  b<-0
  if (value>0.5){
    b<-1
  }
  binaryfinalvec<-c(binaryfinalvec,b)
}
success<-0
for (j in c(1:length(binaryfinalvec))){
  if (binaryfinalvec[j]==test_dat$co2[j]){
    success<-success+1
  }
}
successrate<-success/length(binaryfinalvec)
successrate
```
~~~

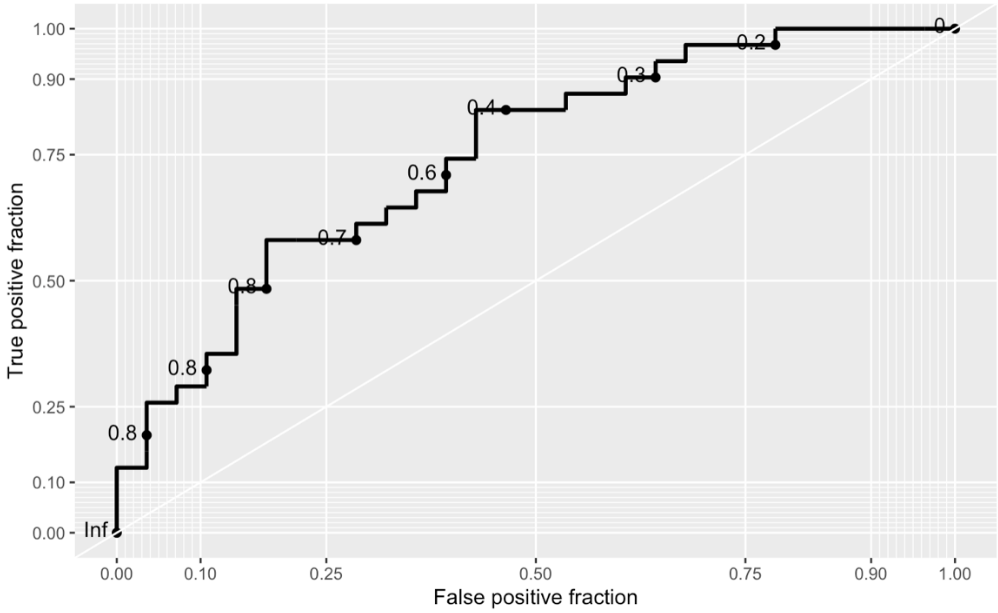
We found our model to have 68% accuracy with our test dataset. We set the threshold at 0.5; if the predicted probability of low CO_2 emissions (success) for one instance in our test set was above 0.5, we called it a 1. Otherwise, it was a 0. In our research we found 0.5 to be the standard threshold value. Also logically, if the probability is above 0.5, it's more likely to be a success than a failure, and vice versa. The 68% was found by comparing these predicted binary outputs (using predict function) to our actual binary outcomes for CO_2 emissions. We had a match for approximately 68% between predicted and actual test dataset values.

We then wanted to check whether our success rate of 68% could have happened randomly. We know that by randomly selecting 0s and 1s, we would, on average, get a 50% match with our test data. How likely is it that we would get a 68% match with a model that was no better than random? To explore this, we performed 10,000 random samples of 0s and 1s in a vector with the same number of elements as our test data. We then compared each of these vectors to the observed outcomes of our test data. As expected, we had approximately 50% accuracy on average. There were only 44 samples of the 10,000 for which we had 68% accuracy or greater. Based on this model testing, there is a probability of 0.0044 that our model was essentially random, which is highly unlikely. Hence, we can conclude that 68% accuracy for our model would not have happened if the model was no better than a random binary classifier.

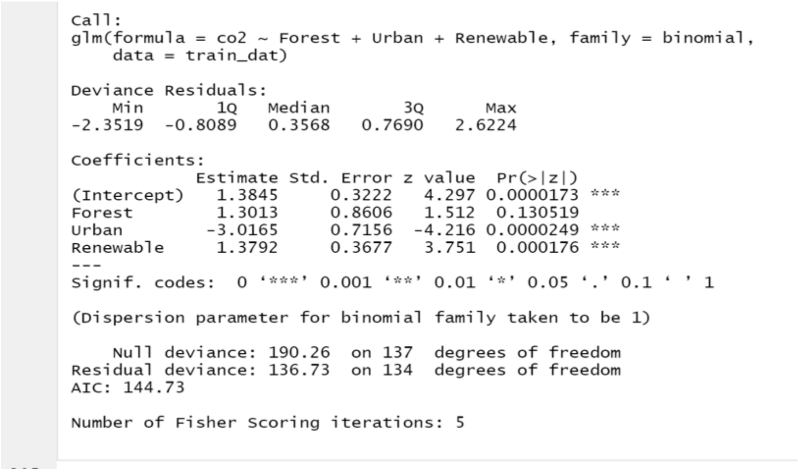
We further visualized the performance of our model using a confusion matrix and ROC curve. A confusion matrix provides the count of true positives, true negatives, false positives and false negatives for the model which further helps to calculate the specificity and sensitivity. A ROC curve is a plot of true positive vs false positive of values predicted by the model. It has a diagonal representing the random classifier; the area under the diagonal is 50% to represent accuracy of random classification. The ROC curve for our model was above the random classifier (diagonal) as its performance was better than random classifier. The area under our model's curve will be 0.68 (68% success rate). Each value on the ROC curve is calculated as sensitivity/specificity. As a model's accuracy improves, the area under the curve will get closer to 1 (true positives increase and false positivces decrease). The ROC curve and code is shown below:



~~~
ROC curve Plotting-
```{r}
basicplot <- ggplot(test_dat, aes(d = test_dat$co2, m = glm_response_scores)) + geom_roc() +style_roc(theme = theme_grey)
basicplot
```
~~~



After examining the summary of the glm function (shown below), we reflected back on our hypothesis for each predictor. From the sign of the coefficients, we see that renewable energy and forest area both have a positive correlation with a low CO_2 output, whereas urban population has a negative correlation. This is what we anticipated. However, when looking at the p-values for each coefficient, we can only confidently say that urban population and renewable energy consumption are correlated with CO_2 emissions based on our data and model. We cannot reject the null hypothesis that forest area and CO_2 emissions are not related.


```
Call:
glm(formula = co2 ~ Forest + Urban + Renewable, family = binomial,
    data = train_dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3519  -0.8089   0.3568   0.7690   2.6224

Coefficients:
            Estimate Std. Error z value  Pr(>|z|)
(Intercept)   1.3845     0.3222   4.297 0.0000173 ***
Forest        1.3013     0.8606   1.512  0.130519
Urban        -3.0165     0.7156  -4.216 0.0000249 ***
Renewable     1.3792     0.3677   3.751  0.000176 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 190.26  on 137  degrees of freedom
Residual deviance: 136.73  on 134  degrees of freedom
AIC: 144.73

Number of Fisher Scoring iterations: 5
```

In future research, we could run various diagnostics to improve the model's accuracy, such as Breusch Pagan and R-step regression. In the Breusch Pagan test we can check the variance of the residuals to test for homoscedasticity or heteroscedasticity of the model. Additionally, stepwise regression (or stepwise selection) consists of iteratively adding and removing predictors, in the predictive model, in order to find the subset of variables in the data set resulting in the best performing model (a model that lowers prediction error). These diagnostics (and others) can be used to further improve the model and get optimal predictions.

Our model performed on our test set with 68% accuracy. We have shown that this likely did not happen due to chance, but we still wanted to hypothesize what we could do in future research to increase the accuracy of this model. Since we're not confident that forest area and CO_2 are correlated in our data, we could drop the forest area set and run the model again. We also could look at additional greenhouse gases (methane and nitrous oxide) rather than just for CO_2 emissions. We also discussed the threshold that we set for success and failure; as 3.551% (the CO_2 average annual percent change for all countries) can be replaced by alternative central measure values that could hold meaning for us, such as a percentage goal that countries may have, perhaps related to international agreements like the Paris Climate Accord.

Bibliography

- https://en.wikipedia.org/wiki/Logistic_regression
- https://www.statisticssolutions.com/assumptions-of-logistic-regression/
- https://rstudio-pubs-static.s3.amazonaws.com/363468_e9d568ed27184ef3b41848e441f42496.html
- https://www.youtube.com/watch?v=YMJtsYIp4kg
- https://climate.nasa.gov/causes/
- https://towardsdatascience.com/understanding-logistic-regression-step-by-step-704a78be7e0a
- https://cran.r-project.org/web/packages/plotROC/vignettes/examples.html
- https://towardsdatascience.com/roc-curve-in-machine-learning-fea29b14d133
- https://www.youtube.com/watch?v=vN5cNN2-HWE
- https://www.youtube.com/watch?v=WflqTUOvdik