Predicting Carbon Emissions From Demographics

Introduction:

The world is in the midst of a global warming crisis. If the current trend of global warming continues, then there will be dire consequences such as famines, floods, and forest fires. Our project aims to predict the CO_2output for each country based on several demographic, economic, and quality of life markers. This would be helpful because it could help indicate certain factors that would lead to an increased carbon footprint in a country. Lawmakers and politicians could then use these findings to mitigate certain factors that would increase the carbon emissions. Being able to identify predictors of CO_2output would also increase consciousness amongst the general public as to which of their activities are detrimental to the environment with regards to CO_2 emissions. The goal of this project was to explore the factors that influence and predict the per capita carbon emissions for the countries of the world. This would hopefully provide insight on the effect of policy proposals on per capita carbon emissions.

Imports and Data Cleaning:

The csv file that we read in was based on combining two other datasets (one being demographic data from CIA World Factbook and the other being carbon emission data from the World Bank). The CO_2output is measured as metric tonnes of CO_2 produced per capita.

The demographic data contained data such as literacy rate, net migration, GDP per capita, economic data, etc. These datasets were combined using Python (Python script is attached to submission), but some issues were encountered because the names of countries were different in both datasets (The Republic of Laos vs. Laos). Once we corrected these issues, we exported the combined data as a CSV, and we were ready to use R to create a model.

```
library(readr)
emissions = read_csv('country_project.csv')
```

Train-Test Split:

We want to split the data into train data and test data so that we can also compare the lasso and ridge regression models. We used a random 80-20 cut for our train-test split. This is because the dataset is organized in alphabetical order by country, and this could lead to some unwanted issues in the regression (i.e. Zambia & Zimbabwe will be grouped together and United States & United Kingdom will be group together)

```
set.seed(1)
train_cut = sample(1:nrow(emissions), floor(nrow(emissions)*0.8), replace=F)
train = emissions[train_cut, 3:22]
test = emissions[-train_cut,3:22]
rmse <- function(x,y) sqrt(mean((x-y)^2))
```

Basic Linear Model:
We first run a Naive Simple Linear Regression in order to get a baseline feel for the models and to eventually check the assumptions (normality assumption, constant variance assumption, and outliers check). The value of 4.633 is the RMSE value.

```
linear_model = lm(EM2016~., data = train)
summary(linear_model)
rmse(predict(linear_model, test), test$EM2016)
```

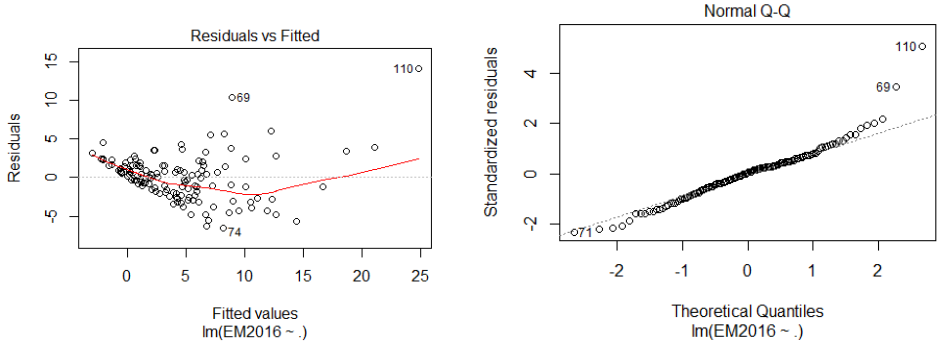
```
Multiple R-squared:  0.7127, Adjusted R-squared:  0.6189
F-statistic: 7.598 on 32 and 98 DF,  p-value: 3.23e-15

[1] 4.633618
```

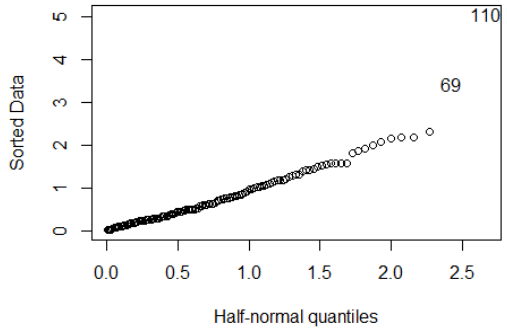
Checking Assumption for Linear Model

```
#Checking Assumptions
library(faraway)
library(car)

plot(linear_model)
```




```
halfnorm(rstandard(linear_model))
```



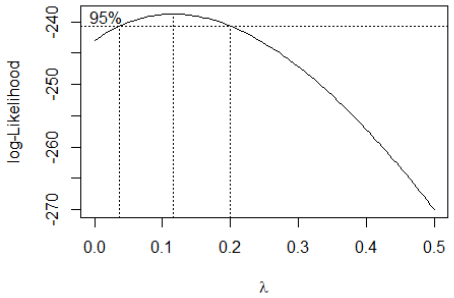
The linear model does not definitively pass the constant variance assumption because there are values that stand out.There seem to be two trends in the plot: in the dense region to the

left, the residuals decrease for higher fitted values and in the sparse region to the right, the residuals seem to increase for higher fitted values. The linear model has a mediocre performance in the normality assumption check, but if we exclude outliers like 110 and 69, the Normal Q-Q plot does not look as bad. Since points do not fall on a straight line in the halfnorm(), there are outliers in the data (69 and 110). We will try to utilize transformations in order to get a better result.

Linear Model using Box Cox Transformation:

The first transformation attempted is the Box Cox transformation below. This yields an R^2 value of 0.743.

```
library(MASS)
boxcox(linear_model, plotit = TRUE, lambda = seq(0, 0.5, 0.05))
```




```
#The best lambda value would be 0.1
boxcox_linear = lm(EM2016^0.1 ~., data = train)
summary(boxcox_linear)

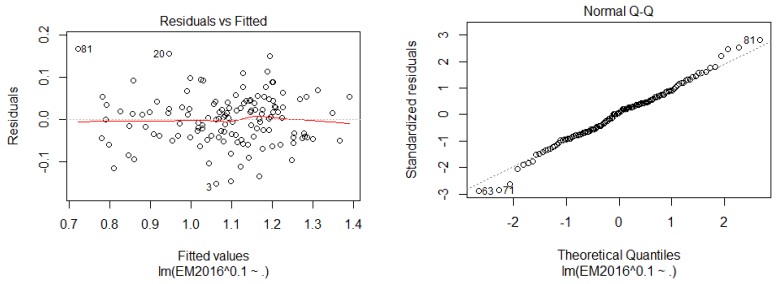
## Residual standard error: 0.06742 on 98 degrees of freedom
## Multiple R-squared:  0.8462, Adjusted R-squared:  0.796
## F-statistic: 16.85 on 32 and 98 DF,  p-value: < 2.2e-16

1-sum((train$EM2016-boxcox_linear$fit^10)^2)/sum((train$EM2016-mean(train$EM2
016))^2)

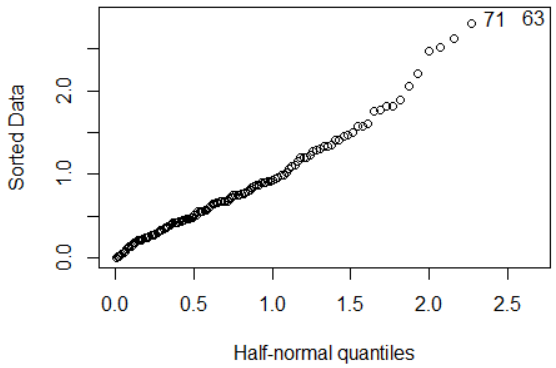
R2: ## [1] 0.742785
```

Check Assumptions for Linear Model for Box Cox Transformation

```
library(faraway)
plot(boxcox_linear)
```




```
halfnorm(rstandard(boxcox_linear))
```



The boxcox linear model seems to pass the constant variance assumption because the points seem to be pretty evenly distributed on the residuals vs. fitted values plot. The boxcox linear model does pass the normality assumption because the standardized residuals do not deviate from the theoretical quantiles. Since points do not fall on a straight line in the halfnorm(), there are outliers in the data (71 and 63)

Log Transformation Linear Model:

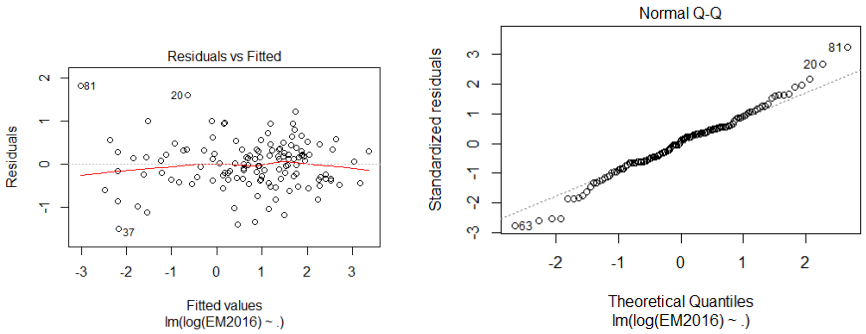
The first transformation attempted is the log transformation below. This yields an R^2 value of 0.985.

```
#Transformation
log_linear = lm(log(EM2016)~., data = train)
1-sum((train$EM2016-exp(log_linear$fit))/sum((train$EM2016-mean(train$EM2016)
)^2))

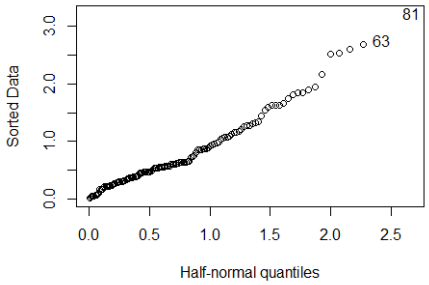
R2:  ## [1] 0.9854669
```

Checking Assumptions for Log Linear Model:

```
plot(log_linear)
```




```
halfnorm(rstandard(log_linear))
```



The log linear model seems to pass the constant variance assumption because the points on the chart are pretty evenly distributed. The scale for the residuals here is much smaller, which means that the log transformation performs better on the constant variation assumption check. The log linear model has a mediocre performance in the normality assumption check, but it does not look too bad, so we can say it passes the normality assumption check and proceed with caution. However, we do notice that this Normal Q-Q plot shows more favorable results than the Normal Q-Q plot for the strictly linear model. Since points do roughly fall on a straight line in the halfnorm(), there are no outliers in the data. We may try to utilize transformations in order to get an even better result.

Outliers:

There were some possible outliers in the plots above. However, we have decided not to remove the outliers. This is because the outliers are not due to incorrect data collection. We believe that our data is extremely reliable because it has been vetted and confirmed by various governmental agencies. Instead, we want our model to take into account these abnormal countries (such as the UAE). For example, the UAE is a country that has an

unusually high CO_2output per capita because of its oil industry. We do not want to exclude such countries; hence, we have decided to keep these outliers in the data.

Transformation Selection:

Moving forward, we should use the log-transformation instead of the Box Cox. This is because, despite both models being moderately valid and seemingly trustworthy, the log-transformation yields marginally better results in terms of checking all the assumptions. What we can say definitively though is that the log and Box Cox transformations both yield better results than simply continuing with non-transformed data.

Other Attempted Models (Cut for Brevity):
We performed transformation and tested different models. However, we were not able to include all of these transformations and models in our final report because it would exceed the 7 page limit and detract from the overall message/cohesiveness of the report. Some of these attempts include using stepwise regression, interaction terms, and polynomial regression. The output and code for some of these attempts are attached for reference in the appendix if you wish to review them.

Lasso Regression with Log Transform:

The lasso regression with the log transformation yielded an RMSE of 3.77.If we use the formula $1 - \frac{RMSE^2}{Variance}$, we get 53% . This isn't fantastic, but it isn't unusable either. It means that our factors predict some of the variance, but there are some factors left unaccounted for.

```
library(glmnet)

xr=model.matrix(~., data = train)
yr= train$EM2016
lasso.cv=cv.glmnet(xr,log(yr), standardize = TRUE)

lasso=glmnet(xr,log(yr),lambda = lasso.cv$lambda.min,standardize = TRUE)
summary(lasso)

coef(lasso)

ypred=predict.glmnet(lasso,newx = model.matrix(~., data = test))
rmse(exp(ypred), test$EM2016)

## [1] 3.77325
```

Ridge Regression with Log Transform
The ridge regression with the log transformation yielded an RMSE of 3.62, which is actually decent considering that the data has a variance of 30. If we use the formula $1 - \frac{RMSE^2}{Variance}$,

we get 57%, which means that 43% is unexplained (due to other predictors). We are moderately satisfied with our RMSE value here. The performance in the ridge regression seems to have a better RMSE value than the RMSE value for the lasso regression.

```
ridges =lm.ridge(log(train$EM2016) ~ ., data = train, lambda = seq(22,23,
len=25))
which.min(ridges$GCV)
##       14

ridge_model = lm.ridge(log(train$EM2016) ~ ., data = train, lambda =
which.min(ridges$GCV))

ypred_test <- (model.matrix(~., data = test[, -20])) %*%
as.matrix(coef(ridges)[14,])
rmse(exp(ypred_test), test$EM2016)

## [1] 3.623286
```

Conclusions:

Our findings tell us that the factors we selected were better at predicting variance within log(emissions) than within emissions, and we found that they worked best when fitted using ridge regression because the ridge regression yielded the smallest RMSE value. We found that for the linear model, the GDP per capita was the most significant predictor of per capita carbon emissions. This makes sense, since GDP describes the amount of production per person, and that production likely emits carbon. When you directly create a model based on just GDP in the simple linear model, however, the R^2 dips to .31, so we want to consider a greater variety of factors.

Through the more advanced models, and training, however, we found that this relation gets lost. We see other factors like birth rate and agriculture having stronger influence. Agriculture certainly makes sense, as a larger percentage of the economy being agriculture generally implies less industrial work. There are exceptions, however. Notably, the US has high carbon emissions and also produces a significant amount of crops. Tying this to current events, Joe Biden's plan to shift away from agriculture and towards industrial/manufacturing work. This may increase CO_2 output according to our model, so Biden's administration wants to consider the environmental impacts of their economic policy decisions.

From our data, we conclude that this problem is tough to solve based only on the factors we used., but intuitive factors like GDP per capita and %of economy in agriculture are strong predictors. For example, based on the demographic data we used, the United Arab Emirates seems like it should have much less per capita carbon emissions than the UK, but in reality produces 5x as much. This can make the problem difficult when we only have 1 data point for each country. Ultimately, our results fall somewhat in line with our intuition of the problem, but requires further investigation into matter using more comprehensive data. Factors like the difference in emissions from coal vs. natural gas

introduce wrinkles that aren't accounted for in our dataset, though they might provide a better understanding of the problem.

Appendix/Extra Models:

Ridge Regression with Log Transformation and Second Degree Polynomial

```
## Ridge Regression with Log Transformation and Second Degree Polynomial
```{r}
ridges =lm.ridge(log(train$EM2016)~ poly(Population, 2) + poly(Area,
    2) + poly(Density, 2) + poly(Coastline, 2) + poly(Migration,
    2) + poly(Infant, 2) + poly(GDP, 2) + poly(Literacy, 2) +
    poly(Phones, 2) + poly(Arable, 2) + poly(Crops, 2) + poly(Other,
    2)+ poly(Birthrate, 2) + poly(Deathrate,
    2) + poly(Agriculture, 2) + poly(Industry, 2) + poly(Service,
    2) , data = train, lambda = seq(38,40, len=25))
which.min(ridges$GCV)
ridge_model = lm.ridge(log(train$EM2016) ~ ., data = train, lambda = which.min(ridges$GCV))

ypred_test <- (model.matrix(~poly(Population, 2) + poly(Area,
    2) + poly(Density, 2) + poly(Coastline, 2) + poly(Migration,
    2) + poly(Infant, 2) + poly(GDP, 2) + poly(Literacy, 2) +
    poly(Phones, 2) + poly(Arable, 2) + poly(Crops, 2) + poly(Other,
    2)+ poly(Birthrate, 2) + poly(Deathrate,
    2) + poly(Agriculture, 2) + poly(Industry, 2) + poly(Service,
    2) , data = test[, -20])) %*% as.matrix(coef(ridges)[5,])
rmse(exp(ypred_test), test$EM2016)
```
```

```
38.33333
       5
[1] 25.56373
```

Polynomial Regression Model (Second Degree)

```
## Linear model

You can also embed plots, for example:

```{r pressure, echo=FALSE}
#Create a naive linear model (all predictors except for country name and index column)

# run regression
linear_model = lm(log(train$EM2016)~ poly(Population, 2) + poly(Area,
    2) + poly(Density, 2) + poly(Coastline, 2) + poly(Migration,
    2) + poly(Infant, 2) + poly(GDP, 2) + poly(Literacy, 2) +
    poly(Phones, 2) + poly(Arable, 2) + poly(Crops, 2) + poly(Other,
    2)+ poly(Birthrate, 2) + poly(Deathrate,
    2) + poly(Agriculture, 2) + poly(Industry, 2) + poly(Service,
    2) , data=train)

summary(linear_model)
rmse(predict(linear_model, test), test$EM2016)
```
```

```
[1] 5.419087
```

Linear Model with Interaction Terms

```
## Interaction

You can also embed plots, for example:

```{r pressure, echo=FALSE}
#Create a naive linear model (all predictors except for country name and index column)

# run regression
interaction_linear_model = lm(log(train$EM2016)~ .^2, data=train)

summary(interaction_linear_model)
rmse(predict(interaction_linear_model, test), test$EM2016)
```
```

```
Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:    NaN
F-statistic:   NaN on 130 and 0 DF,  p-value: NA

prediction from a rank-deficient fit may be misleading[1] 867.997
```

Stepwise Regression Code and Results

```r
library(MASS)
step_model = step(log_linear)
summary(step_model)
rmse(predict(step_model, test), test$EM2016)
```

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  3.263e+00  4.209e-01   7.753 4.10e-12 ***
RegionBALTICS               -1.995e-01  4.847e-01  -0.412 0.681377
RegionC.W. OF IND. STATES    3.010e-01  2.991e-01   1.006 0.316323
RegionEASTERN EUROPE        -1.899e-01  3.447e-01  -0.551 0.582797
RegionLATIN AMER. & CARIB   -5.138e-01  2.037e-01  -2.523 0.013028 *
RegionNEAR EAST              1.575e-01  3.038e-01   0.519 0.604999
RegionNORTHERN AFRICA       -4.307e-01  4.760e-01  -0.905 0.367479
RegionNORTHERN AMERICA      -3.532e-01  5.567e-01  -0.634 0.527043
RegionOCEANIA                1.582e-01  2.670e-01   0.593 0.554669
RegionSUB-SAHARAN AFRICA    -8.010e-01  2.172e-01  -3.687 0.000349 ***
RegionWESTERN EUROPE        -9.611e-01  2.941e-01  -3.268 0.001431 **
`Pop. Density (per sq. mi.)` -1.082e-04  4.080e-05  -2.651 0.009164 **
`GDP ($ per capita)`         3.670e-05  1.141e-05   3.218 0.001683 **
`Arable (%)`                -1.097e-02  5.033e-03  -2.180 0.031314 *
Birthrate                   -4.432e-02  1.077e-02  -4.116 7.33e-05 ***
Agriculture                 -4.476e+00  5.956e-01  -7.514 1.40e-11 ***
Service                     -1.043e+00  5.237e-01  -1.992 0.048768 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6302 on 114 degrees of freedom
Multiple R-squared:  0.8322,    Adjusted R-squared:  0.8087
F-statistic: 35.34 on 16 and 114 DF,  p-value: < 2.2e-16

[1] 5.729358
```

Simple Linear Based on GDP:

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.581e+00  5.494e-01   2.877   0.0047 **
`GDP ($ per capita)` 3.011e-04  3.883e-05   7.754 2.35e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.591 on 129 degrees of freedom
Multiple R-squared:  0.3179,    Adjusted R-squared:  0.3126
F-statistic: 60.12 on 1 and 129 DF,  p-value: 2.353e-12

[1] 4.239395
```