Applied Finance Project

Payden & Rygel

Jingze Sun

12/11/2020

Applied Finance Project Final Report

Examine the Feasibility of Using Emissions Data to Predict the Yield-to-Worst Spread of Municipal Bonds

1. Introduction

U.S. investors have long been interested in the topic of Environmental, Social, and Governance (ESG) driven investment, which has been an expanding market in recent years. According to (Baker, 2020), the value of global assets applying [ESG] data to drive investment decisions has almost doubled over four years, and more than tripled over eight years, to $40.5 trillion in 2020. The client of our project, Payden & Rygel, was interested leveraging the municipal emissions data available on the CDP (formerly the Carbon Disclosure Project) website in order to make better investment decisions in their municipal bond strategy. In response to their request, we decided to examine the relationship between the yield-to-worst (YTW) spread of the municipal bond and the normalized total annual emissions of the corresponding municipality. We hoped that the results of our project would inform our client on whether emissions data could be used to assess the risk and thereby better price the municipal bonds in their portfolio.

Using the callable general obligation bond with the longest maturity for each municipality, we used the multivariate linear regression model and the random forest model to perform our analysis. The outputs of our models indicate that the normalized total annual emissions of the municipality is not predictive of the YTW spread, while controlling for other variables, such as the modified duration and the GDP of the municipality. Although further analysis may be required for our project, we believe that this result may suggest that metrics such as the modified duration and the credit rating of a bond may already be sufficient in reflecting its level of risk. Furthermore, given that most of the analyses in the current literature used proprietary ESG scores developed by specialized institutions, we think that emissions is not informative of the risk of the bond by itself.

2. Literature Review

1) The municipal bond market

US municipal bonds are the primary funding source for US infrastructure. State, county and local governments and agencies issue these tradeable debt instruments to build highways, airports, water and sewer plants, and other structures that provide essential services to the public.

With nearly $3.9 trillion in outstanding bonds, municipal bonds make up nearly 10% of the value of the $40.8 trillion U.S. bond market in year 2019 (MacKay Municipal Managers, 2019).

2) Previous research on ESG factors and bond pricing:

There is a growing literature on the relationship between ESG factors and bond characteristics. For instance, in the research paper published by Hermes Investment Management, the authors found that there is a negative correlation between the bond's previous year ESG score and its log annual average CDS spread. In addition, the paper showed that the log annual average CDS spread is negatively correlated with the quantified credit rating score (2017). Similarly, Kjerstensson and Nygren (2019) have found that, controlling for other bond characteristics such as maturity and duration, etc., the ESG score has a low yet statistically significant negative effect on corporate bond yield spread. In addition, Cadybury and Nemnov (2020) showed that the gradual removal of the top-polluting companies in a corporate bond investment portfolio leads to a decrease in the level of YTM. In summary, the evidence from the current literature shows that we should expect financial institutions with higher ESG performance to have lower level of risk in their fixed income instruments such as corporate bonds.

The most important study that this paper will follow was done by Chen and Gao (2011), in which they found that the firms' cost of debt, as measured by the bond yield-to-maturity spread, is positively correlated with carbon dioxide emission rates. The measurement of carbon dioxide emissions rates in their paper was defined as total emissions divided by electricity generation in MWh. In regards to control variables, they used various factors that had been shown to be associated with firms' cost of capital, such as firm size, market to book ratio, and leverage, etc.

3. Data Description

1) Greenhouse Gas (GHG) emissions data:

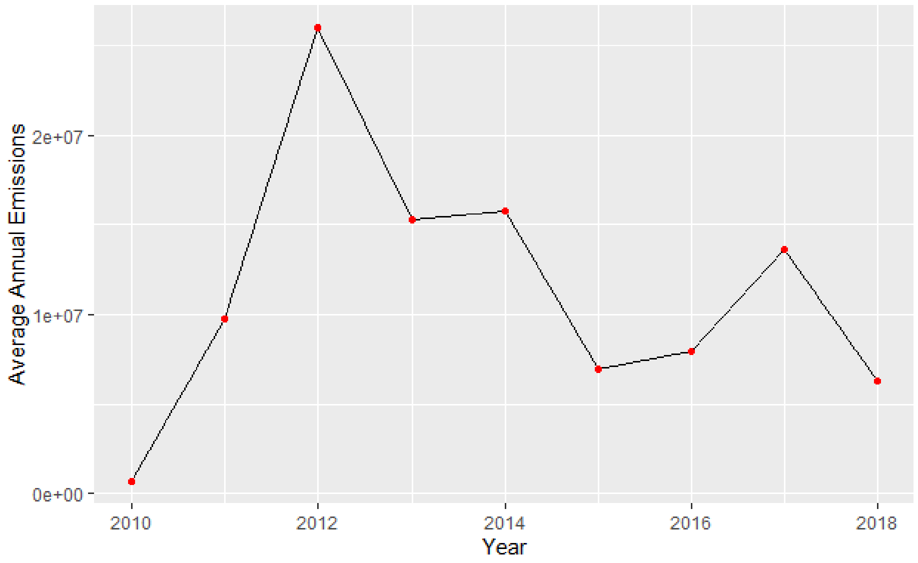
The municipal GHG emissions data was obtained from the Cities, states and regions data portal on the CDP website. Table 1 shows the number of cities and towns with valid observations for each in year from 2012 to 2018. Graph 1 shows the average emissions each year.

Table 1

Year	Number of Reporting Cities and Towns
2010	2
2011	4
2012	4
2013	9
2014	15
2015	24
2016	19
2017	16

2018	6

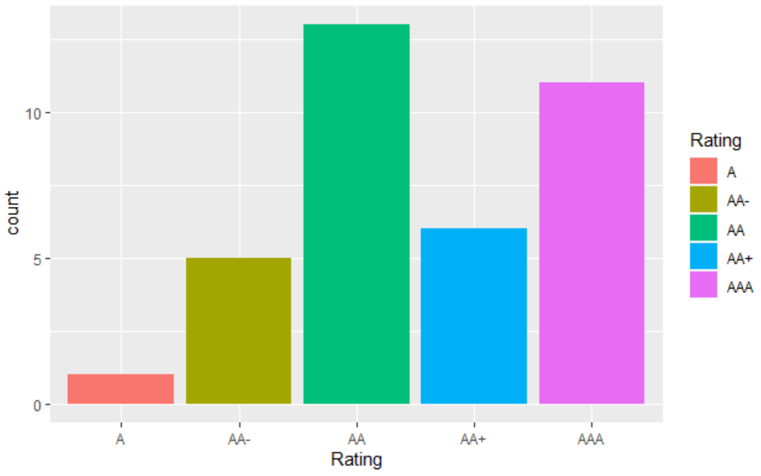
Graph 1: (Unit: metric tons)



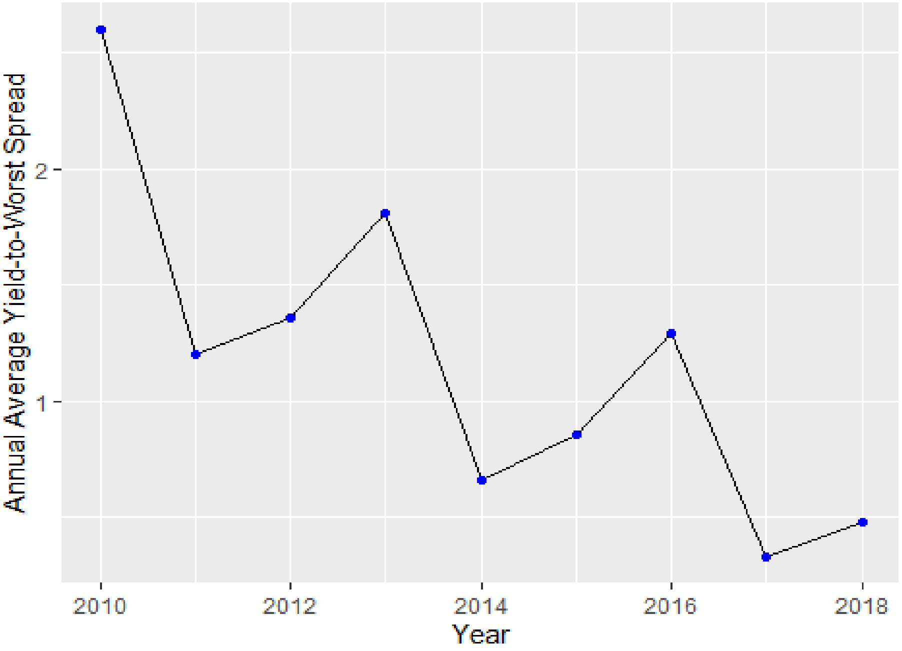
2) Municipal bond YTW data:

The municipal bond data was obtained from Bloomberg. Since each municipality has issued several municipal bonds over the years, we developed a set of conditions under which we selected one municipal bond for each municipality: 1) The bond has to be a general obligation bond. 2) It cannot have irregular status, such as “called” or “refunded”, etc. 3) It must be a callable bond, and 4) it cannot have a maturity greater than 30 years. Once we obtained a set of bonds that meet the above criterion, we picked the one that has the longest maturity among them. Graph 2 shows the distribution of bonds by their S&P credit ratings. Graph 3 shows the average YTW Spread by year.

Graph 2

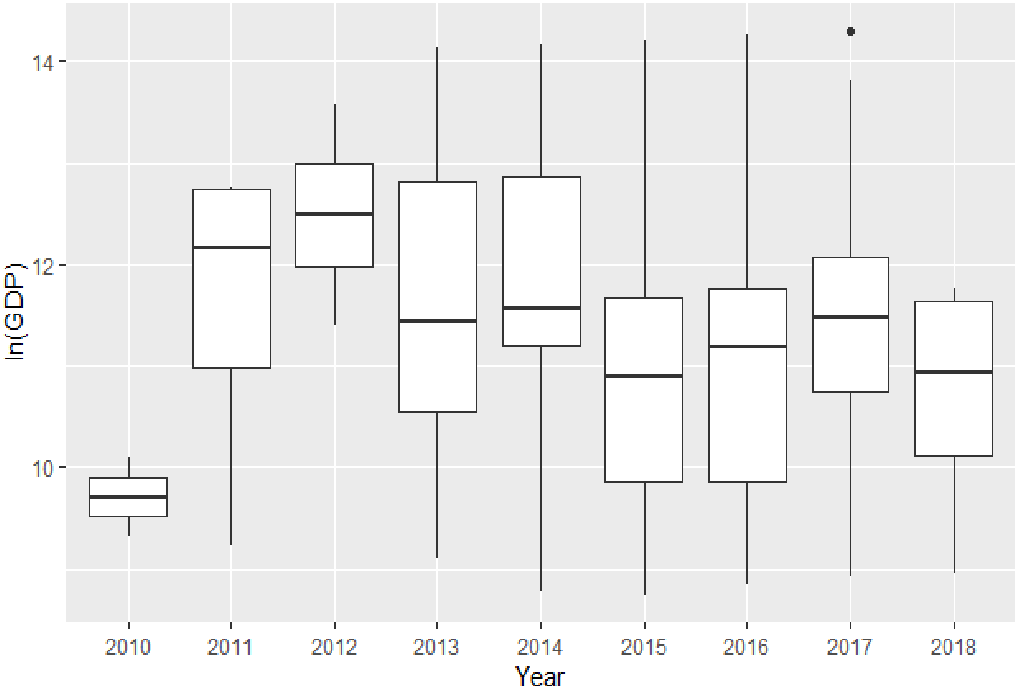


Graph 3



3) GDP data: All the GDP data of the municipalities included in our data set were collected from the Economic Research section of the Federal Reserve Bank of St. Louis's website. Graph 4 shows the boxplot of the natural log of GDP volume by each year.

Graph 4: (Unit: Natural Log of Millions of USD)



4. Methodology

1) Multivariate linear regression model:

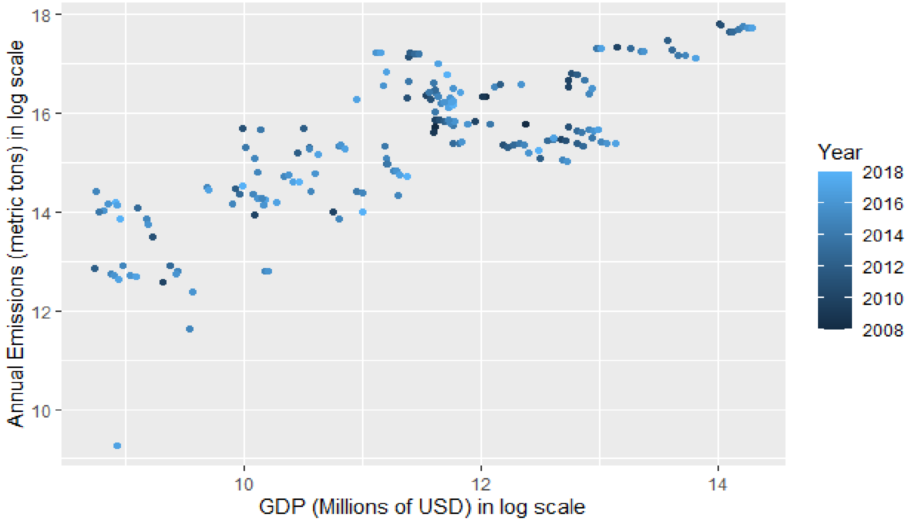
a) Model construction: We modified the empirical model used in Chen and Gao's study (2011), which is specified as the following:

$$YTW\_Spread_{it} = \beta_0 + \beta_0 I_{R\,it} + \vec{\beta}_1 \vec{X}_{it} + \varepsilon_{i,t}$$

, where $YTW\_Spread_{it} = Bond\ YTW_{it} - Benchmark\ Yield_t$, I_R is normalized total annual emissions, and $\vec{X}_{it}$ is the matrix consisting of other control variables in our model. The benchmark yield is defined as the best possible matching US Treasury yield, given the years to maturity of bond *i* at year *t*. For instance, if bond *i* has 3 years until its maturity, then its best possible matching US Treasury yield is the 5-year US Treasury yield. According to Tan, Wirjanto, and Fang, there are two definitions for normalized total annual emissions: 1) $I_R = \frac{AE}{RV}$ and 2) $I_P = \frac{AE}{P}$, where AE, RV, and P represents the company's annual emission, annual revenue and annual net profit (2018). In the case of our analysis, we substitute the denominator with the GDP of the municipality.

b) Feasibility of the normalized annual emissions regressor: Before we incorporate normalized annual emissions in our model, we want know if it is justifiable to normalize municipality's annual emissions by its GDP. In other words, we want to know there is a potential linear relationship between these two variables. Graph 5 shows the scatterplot of GDP (in millions of USD; log scale) and annual emissions (in metric tons; log scale).

Graph 5



We later ran a regression analysis on the two variables. Our result shows that the natural log of annual emissions is positively and significantly correlated with the natural log of GDP.

```
Log Annual Emissions and log GDP
==============================================
                  Dependent variable:
              ---------------------------
                        ln(AE)
----------------------------------------------
ln(GDP)                 0.784***
                       (0.046)

Constant               6.462***
                       (0.524)

----------------------------------------------
Observations              181
R2                       0.624
Adjusted R2              0.621
Residual Std. Error      0.863 (df = 179)
F Statistic              296.509*** (df = 1; 179)
==============================================
Note:          *p<0.1; **p<0.05; ***p<0.01
```

c) Collinearity check on possible regressors: Before we decided upon which regressors to use in our multivariate regression model, we had considered a set of possible candidates: Normalized annual total emissions (Annual emissions divided by GDP), modified duration, years until the bond's maturity year, natural log of GDP, and annual emissions in millions of metric tons. In order to avoid the issue of having collinearity in the model, we examined the correlation matrix of these regressor candidates.

Table 2 shows the correlation matrix of all the possible regressors. We found that there is strong correlation between the natural log of GDP and annual emissions.

Table 2

	Norm_AE	**Mod_Dur**	**Yr_t_Mat**	**ln_GDP**	**AE_Mil**
Norm_AE	1				
Mod_Dur	-0.16665938	1			
Yr_t_Mat	0.07090726	0.529649111	1		
ln_GDP	-0.26136070	0.123498266	-0.16016001	1	
AE_Mil	0.11865094	0.005351026	-0.15134068	0.7780765	1

d) Control variables: The set of control variables in the model are: i) Mid-adjusted modified duration of the bond; ii) the natural log of the GDP of the municipality, and iii) the credit rating of the bond, which has been converted to a dummy variable for each rating. Our decisions to include the control variables modified duration and credit rating were inspired by the control variables used in the studies mentioned the literature review section of this paper, while the choice for using the natural log of the GDP falls under the same logic as using a company's size as a control variable.

2) Random forest model:

a) Model construction: We split our data set with 80% as the training set and 20% as the test set. In addition to modified duration and the natural log of GDP, we added individual credit rating dummy variables into our random forest model, since we were interested in learning if there is a differentiating effect on YTW spread across different levels of credit rating.

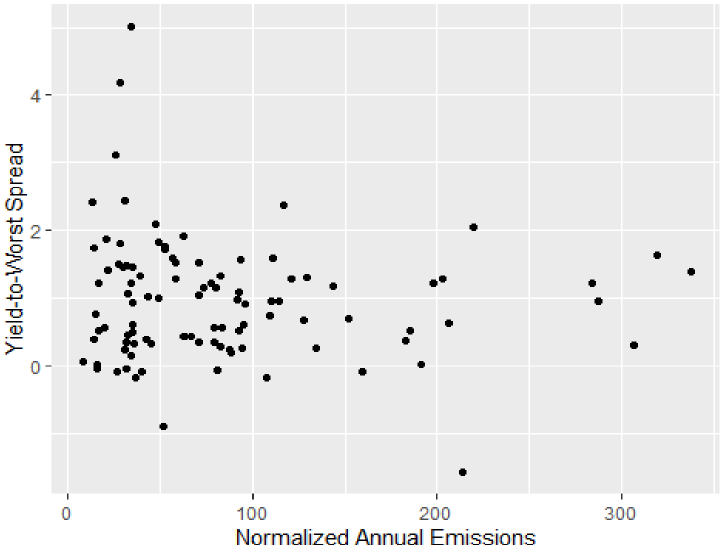
b) Hyperparameter tuning of the random forest model: In order to minimize the out-of-bag (OOB) error in the model trained from the training data set, we used the *tuneRF* function in R to identify the optimal parameter for the *m-try* parameter of the random model, which is the number of candidate regressors considered at each split. The output of the *tuneRF* shows that the OOB error is minimized when *m-try* is set to 2.

5. Empirical Analysis:

1) Relationship between YTW spread and normalized annual emissions:

Graph 6 shows the scatterplot of YTW spread and normalized annual emissions. There is no distinctive pattern indicated on the graph.

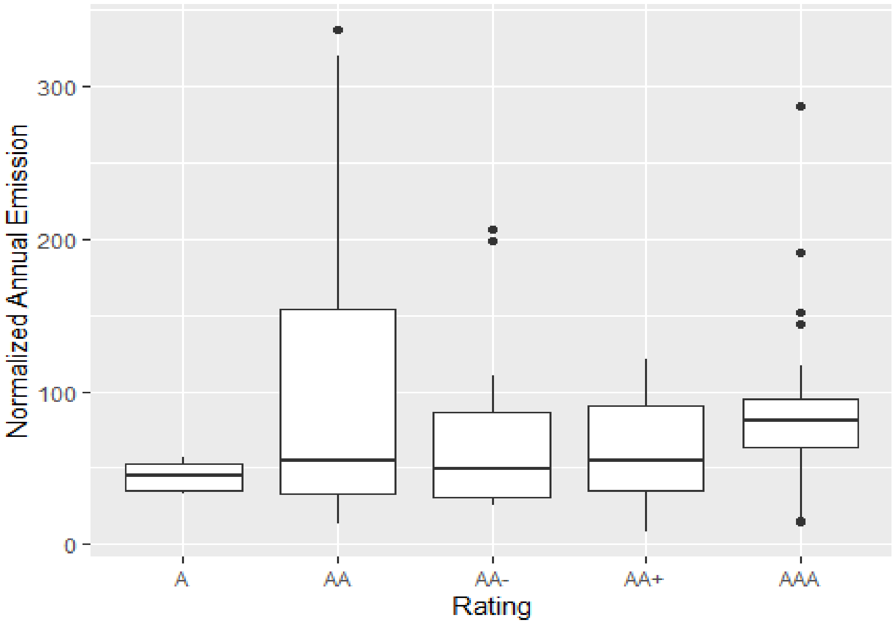
Graph 6



2) Relationship between normalized annual emissions and bond ratings:

We wanted to know if there was a disparity in the distribution of normalized annual emissions across different bond ratings. Graph 7 shows that the median normalized annual emissions is about the same for bonds with ratings from A to AA+, while the AAA bonds have slightly higher median normalized annual emissions.

Graph 7



3) Results of YTW spread on regressors:

We performed univariate regressions on each of the regressors. Table 3 shows the result of our analysis.

Table 3

Regressor	Coefficient	P value	Adjusted R Square
Normalized Annual Emission	-0.001198	0.345	-0.001015
Modified Duration	0.24530	9.18e-10	0.3149
ln(GDP)	-0.07553	0.2176	0.003517

The normalized annual emissions is not statistically significantly correlated with the YTW spread. Modified duration is statistically significantly correlated with the YTW spread, and the regression coefficient is positive. The natural log of GDP is not statistically significantly correlated with the YTW spread.

4) Multivariate regression model result:

```
Yield-to-Worst Spread on Normalized Annual Emissions with Controls
============================================
                     Dependent variable:
                   ---------------------------
                     Yield-to-Worst Spread

Normalized Annual Emissions           -0.001 (0.001)
Modified Duration                     0.233*** (0.034)
ln(GDP)                               -0.147*** (0.053)
Rating: AA                            -0.387 (0.327)
Rating: AA-                           0.271 (0.392)
Rating: AA+                           -0.524 (0.356)
Rating: AAA                           -0.787** (0.340)
Constant                              1.889** (0.748)

Observations            99
R2                      0.500
Adjusted R2             0.461
Residual Std. Error     0.670 (df = 91)
F Statistic             12.985*** (df = 7; 91)
============================================
```

Note: *p<0.1; p<0.05;* p<0.01

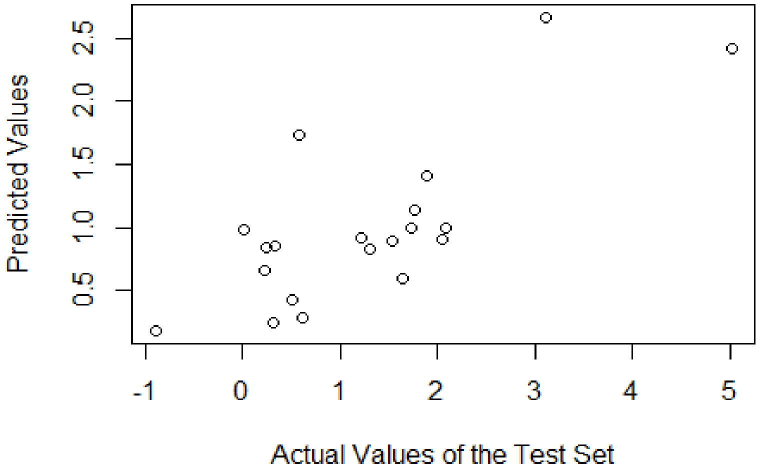
The normalized annual emissions does not have a statistically significant correlation with the YTW spread of the municipal bond. The modified duration has a positive and statistically significant correlation with YTW spread. Interestingly, our results also show that bonds with the highest AAA rating have significantly lower YTW spread, but other credit ratings are not predictive of the YTW spread.

5) Random forest model results:

a) Model performance on the training data: The random forest achieved a MSE of 0.4825779, which is higher than the MSE of the multivariate linear regression model (0.4121088).

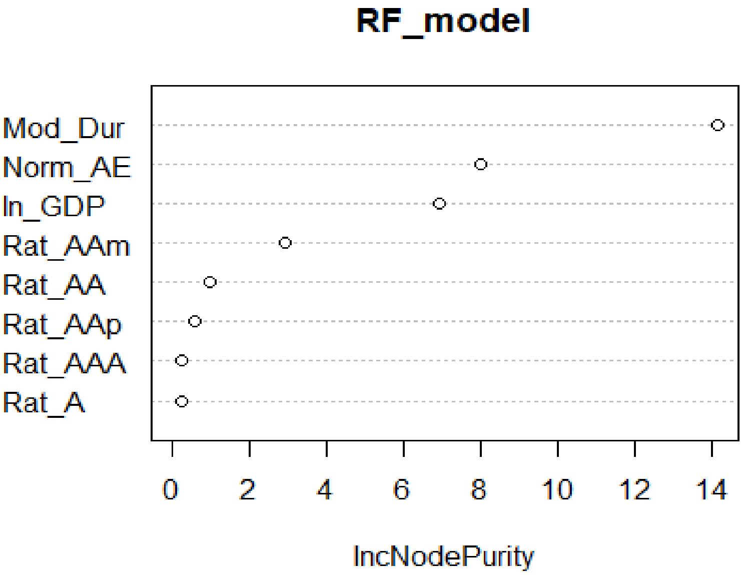
b) Model performance on the test data: The random forest achieved a MSE of 0.838424, which is significantly higher than the MSE of the multivariate linear regression model (0.4121088). This suggest that the random model does not perform very well out-of-sample, and that the multivariate linear regression model is the one with better performance between the two models. Graph 8 shows the scatterplot of the actual YTW spread values in the test data and the predicated YTW spread values.

Graph 8



c) Importance ranking of variables: Graph 9 shows the importance of variables used in the random forest, ranked by a measure of node impurity:

Graph 9



Modified duration is shown to be the most important variable. Normalized annual emissions and the natural log of GDP are ranked 2nd and 3rd respectively. The individual credit ratings are not nearly as important the others.

6. Discussion:

The results from both the multivariate regression model and the random forest model indicate that, controlling for modified duration, the natural log of the municipality's GDP, and the bond's credit rating, the normalized annual emissions of a municipality is not predictive of the YTW spread of the municipal bond. This conclusion is not consistent with the idea in the current literature that financial institutions with higher ESG performance have lower debt risk measured by factors such as average CDS spread. We think there could be several reasons that this is the case in our study.

First, our study only examines the effect of emissions on bond's YTW spread. This approach has its limitation, since it omits the effects of the social and governance aspects of the municipality. Hence, we think it would be interesting to examine the relationship between the ESG score of the municipality listed on the CDP website and the YTW spreads of the municipal bonds that the municipality had issued.

Second, as the source of the emissions data used our analysis, CDP does not have a consistent schedule for the publication of the emissions data. In some years, the emissions data set of the year was created at the end of the year, while in other years the data set was created a few months later in the year after. In addition, CDP continues to update the emissions data set in the years after it was published. Furthermore, the emissions data set published in a given year only include the emissions data measured in the years prior in most cases. Thus, we argue that the inconsistent publication schedule and the ongoing data set maintenance made it difficult for market participants to react to the information on the emissions in a timely manner, and therefore it was unlikely for the information to have a prompt influence on the level of risk of the municipal bond.

Third, although our study shows that only the AAA rating is predictive of the bond's YTW spread, the bond's risk may have already been "priced into" the bond's credit rating. Based on the definitions of credit rating provided by S&P Global (2020), the credit rating of a bond is reflective of the obligor's capability to meet its financial commitments. Therefore, once we control for the bond's credit rating, emissions may not have sufficient explanatory power on the changes in bond's YTW spread.

7. References:

Baker, S. (2020, July 2). *Global ESG-data driven assets hit $40.5 trillion*. Retrieved from Pensions & Investments: https://www.pionline.com/esg/global-esg-data-driven-assets-hit-405-trillion

Cadbury, R., & Nemnov, K. (2020, July 8). *ESG Investing Combining Performance and Impact in Fixed Income.* Retrieved from Climate Action: https://www.ssga.com/library-content/pdfs/insights/ESG_in_fixed_income.pdf

Chen, L. H., & Gao, L. S. (2011, October 7). *The Pricing of Climate Risk.* Retrieved from SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1940727

Hermes Investment Management. (2017, April). *Pricing ESG Risk in Credit Markets.* Retrieved from https://www.hermes-investment.com/ukw/wp-content/uploads/sites/80/2017/04/Credit-ESG-Paper-April-2017.pdf

Kjerstensson, L., & Nygren, H. (2019). *ESG Rating and Corporate Bond Perforamnce - An analysis of the effect of ESG rating on yield spread.* Retrieved from DiVA: https://www.diva-portal.org/smash/get/diva2:1333903/FULLTEXT01.pdf

MacKay Municipal Managers. (2019, January). *US Taxable Municipal Bonds: Beyond the Basics.* Retrieved from https://www.mackayshields.com/images/pdf/Beyond-the-Basics_FINAL.pdf

S&P Global. (2020, December 7). *S&P Global Ratings Definitions*. Retrieved from S&P Global Ratings: https://www.standardandpoors.com/en_US/web/guest/article/-/view/sourceId/504352#:~:text=B.,-Issuer%20Credit%20Ratings&text=An%20S%26P%20Global%20Ratings%20issuer,commitments%20as%20they%20come%20due.

Tan, K., Wirjanto, T. S., & Fang, M. (2018, February). *Managing Climate and Carbon Risk in Investment Portfolios.* Retrieved from https://www.soa.org/globalassets/assets/Files/resources/research-report/2018/managing-climate-carbon-risk.pdf

8. Appendixes & R Code

Jingze Sun

12/11/2020

Set up the environment:

```r
# Clear workspace:
rm(list = ls())
```

```r
# Suppress warnings:
options(warn = -1)
```

Load Packages:

```r
library(ggplot2)
library(data.table)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```r
library(writexl)
library(readxl)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(stargazer)
```

```
##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

Raw Individual Data Sets:

```r
# Create copies of raw data for future references:
CDP_2019_raw <- fread("Emissions_2019.csv")
CDP_2018_raw <- fread("Emissions_2018.csv")
CDP_2017_raw <- fread("Emissions_2017.csv")
CDP_2016_raw <- fread("Emissions_2016.csv")
CDP_2015_raw <- fread("Emissions_2015.csv")
CDP_2014_raw <- fread("Emissions_2014.csv")
CDP_2013_raw <- fread("Emissions_2013.csv")
CDP_2012_raw <- fread("Emissions_2012.csv")
```

Data Cleaning:

CDP Data:

```r
# Year 2012:
CDP_2012 <- fread("Emissions_2012.csv")
CDP_2012 <- CDP_2012[,c(6,7,2,1,10)]
colnames(CDP_2012) <- c("Rep_Year","Mea_Year","ID","City","AE")

# Correct the format of the annual emissions column:
CDP_2012$AE <- as.numeric(gsub(",", "", CDP_2012$AE))
CDP_2012[,GDP:=NA]
```

```r
# Year 2013:
CDP_2013 <- fread("Emissions_2013.csv")
CDP_2013 <- CDP_2013[,c(6,9,2,1,10)]
colnames(CDP_2013) <- c("Rep_Year","Mea_Year","ID","City","AE")
```

```r
# Correct the format of the measurement year column:
CDP_2013[,Mea_Year:=as.integer(str_sub(Mea_Year,1,4))]

# Correct the format of the annual emissions column:
CDP_2013$AE <- as.numeric(gsub(",", "", CDP_2013$AE))
CDP_2013[,GDP:=NA]
```

```r
# Year 2014:
CDP_2014 <- fread("Emissions_2014.csv")
CDP_2014 <- CDP_2014[,c(6,7,2,1,10)]
colnames(CDP_2014) <- c("Rep_Year","Mea_Year","ID","City","AE")

# Correct the format of the annual emissions column:
CDP_2014$AE <- as.numeric(gsub(",", "", CDP_2014$AE))

# Correct errors in the annual emissions column based on information in the
# "Reason for Increase/Decrease in emissions column:
# 1. Pittsburgh data should be 6.79 million
CDP_2014$AE[CDP_2014$City=="City of Pittsburgh"] <- 6.79*1000000
# 2. Seattle data should be 3.6 million
CDP_2014$AE[CDP_2014$City=="City of Seattle"] <- 3.6*1000000
CDP_2014[,GDP:=NA]

# Note: Chicago is missing reporting year 2014 data.
```

```r
# Year 2015:
CDP_2015 <- fread("Emissions_2015.csv")
CDP_2015 <- CDP_2015[Country=="USA",]
CDP_2015 <- CDP_2015[,c(6,7,2,1,10)]
colnames(CDP_2015) <- c("Rep_Year","Mea_Year","ID","City","AE")

# Correct the format of the measurement year column:
CDP_2015[,Mea_Year:=year(mdy_hm(Mea_Year))]

# Correct the format of the annual emissions column:
CDP_2015$AE <- as.numeric(gsub(",", "", CDP_2015$AE))
CDP_2015[,GDP:=NA]
```

```r
# Year 2016:
CDP_2016 <- fread("Emissions_2016.csv")
CDP_2016 <- CDP_2016[,c(6,7,1,2,12,19)]
colnames(CDP_2016) <- c("Rep_Year","Mea_Year","ID","City","AE","GDP")

# Correct the format of the measurement year column:
CDP_2016[,Mea_Year:=year(mdy_hms(str_remove(Mea_Year, " AM")))]

# Correct the format of the annual emissions and GDP columns:
CDP_2016$AE <- as.numeric(gsub(",", "", CDP_2016$AE))
CDP_2016$GDP <- as.numeric(gsub(",", "", CDP_2016$GDP))
```

```r
# Year 2017:
CDP_2017 <- fread("Emissions_2017.csv")
CDP_2017 <- CDP_2017[,c(8,9,1,2,16,17,14,23)]
colnames(CDP_2017) <- c("Rep_Year","Mea_Year","ID","City","S1","S2","AE","GDP")
CDP_2017$S1 <- as.numeric(gsub(",", "", CDP_2017$S1))
```

```
CDP_2017$S2 <- as.numeric(gsub(",", "", CDP_2017$S2))
CDP_2017$AE <- as.numeric(gsub(",", "", CDP_2017$AE))
CDP_2017$GDP <- as.numeric(gsub(",", "", CDP_2017$GDP))

# Correct the format of the measurement year column:
CDP_2017[,Mea_Year:=as.integer(str_sub(Mea_Year,1,4))]

# Handle missing emissions data:
CDP_2017$S1[is.na(CDP_2017$S1)] <- 0
CDP_2017$S2[is.na(CDP_2017$S2)] <- 0

CDP_2017$AE <- ifelse(!is.na(CDP_2017$AE), CDP_2017$AE,
                      CDP_2017$S1 + CDP_2017$S2)

# Keep only relevant columns:
CDP_2017 <- CDP_2017[,c(1:4,7,8)]
```

```
# Year 2018:
CDP_2018 <- fread("Emissions_2018.csv")
CDP_2018 <- CDP_2018[,c(1,10,2,3,22,23)]
colnames(CDP_2018) <- c("Rep_Year","Mea_Year","ID","City","S1","S2")

# Handle missing AE data:
CDP_2018$S1[is.na(CDP_2018$S1)] <- 0
CDP_2018$S2[is.na(CDP_2018$S2)] <- 0

# Obtain GDP and AE columns:
CDP_2018[,AE:=S1+S2]
CDP_2018[,GDP:=NA]
CDP_2018 <- CDP_2018[,c(1:4,7,8)]

# Correct the format of the measurement year column:
CDP_2018[,Mea_Year:=as.integer(str_sub(Mea_Year,1,4))]

# Remove rows with missing AE data:
CDP_2018 <- CDP_2018[!(CDP_2018$AE==0),]

# We make the assumption that if the measurement year is missing, it is the
# same as the reporting year:
CDP_2018[,Mea_Year:=ifelse(!is.na(Mea_Year),Mea_Year,Rep_Year)]
```

```
CDP_2019 <- fread("Emissions_2019.csv")
CDP_2019 <- CDP_2019[,c(1,10,2,3,16:19,22,23)]
colnames(CDP_2019) <- c("Rep_Year","Mea_Year","ID","City","S1_A",
                        "S1_B","S2_A","S2_B","S1","S2")

# Handle missing AE data:
CDP_2019$S1_A[is.na(CDP_2019$S1_A)] <- 0
CDP_2019$S1_B[is.na(CDP_2019$S1_B)] <- 0
CDP_2019$S2_A[is.na(CDP_2019$S2_A)] <- 0
CDP_2019$S2_B[is.na(CDP_2019$S2_B)] <- 0
CDP_2019$S1 <- ifelse(!is.na(CDP_2019$S1), CDP_2019$S1,
                      CDP_2019$S1_A + CDP_2019$S1_B)
CDP_2019$S2 <- ifelse(!is.na(CDP_2019$S2), CDP_2019$S2,
                      CDP_2019$S2_A + CDP_2019$S2_B)
CDP_2019[,AE:=S1+S2]

# Keep only relevant columns:
CDP_2019 <- CDP_2019[,c(1:4,11)]

# Add the empty GDP column:
CDP_2019[,GDP:=NA]

# Correct the format of the measurement year column:
CDP_2019[,Mea_Year:=as.integer(str_sub(Mea_Year,1,4))]

# Remove rows with missing AE data:
CDP_2019 <- CDP_2019[!(CDP_2019$AE==0),]
```

```
# Combine the data from each year:
CDP_raw <- rbind(CDP_2019, CDP_2018) %>% rbind(CDP_2017) %>%
           rbind(CDP_2016) %>% rbind(CDP_2015) %>% rbind(CDP_2014) %>%
           rbind(CDP_2013) %>% rbind(CDP_2012)
# Reorder the columns:
CDP_raw <- CDP_raw[,c(3,1,2,4:6)]
# Sort the CDP_raw data table:
setkey(CDP_raw, ID, Mea_Year, Rep_Year)
# Note: At the this point, the raw combined CDP emissions data is created.
```

```
# Keep only rows with non-missing emissions data:
CDP_raw_AE <- CDP_raw[!is.na(AE),]
```

```
# Clean the City Name column:

# 1. Remove "City of" and "Town of"
CDP_raw_AE$City = str_remove(CDP_raw_AE$City, "City of ")
CDP_raw_AE$City = str_remove(CDP_raw_AE$City, "Town of ")

# 2. Address the missing state info in city names:
City_Dummy <- CDP_raw_AE %>%
  group_by(ID) %>%
  slice(1)
```

```
# 2.1 Extract individual city names:
City_Dummy <- City_Dummy[,c(1,4)]

# 2.2 Correct small errors:
City_Dummy$City[City_Dummy$ID==35272] <- "New Haven"
City_Dummy$City[City_Dummy$ID==49335] <- "Nashville"
City_Dummy$City[City_Dummy$ID==52897] <- "Aspen"
City_Dummy$City[City_Dummy$ID==35883] <- "San Jose"

# 2.3 Export out the city name files:
#write_xlsx(City_Dummy,"C:\\Users\\Michael Sun\\Desktop\\City_names.xlsx")

# 3. Create the CDP raw data set with annual emissions:
CDP_raw_AE <- CDP_raw_AE %>%
  left_join(City_Dummy, by="ID")
```

```r
CDP_raw_AE <- as.data.table(CDP_raw_AE)

CDP_raw_AE <- CDP_raw_AE[,c(1:3,7,5,6)]

colnames(CDP_raw_AE) <- c("ID","Rep_Year","Mea_Year","City","AE", "GDP")

# Sort the CDP_raw_AE data table:
setkey(CDP_raw_AE, ID, Rep_Year, Mea_Year)
```

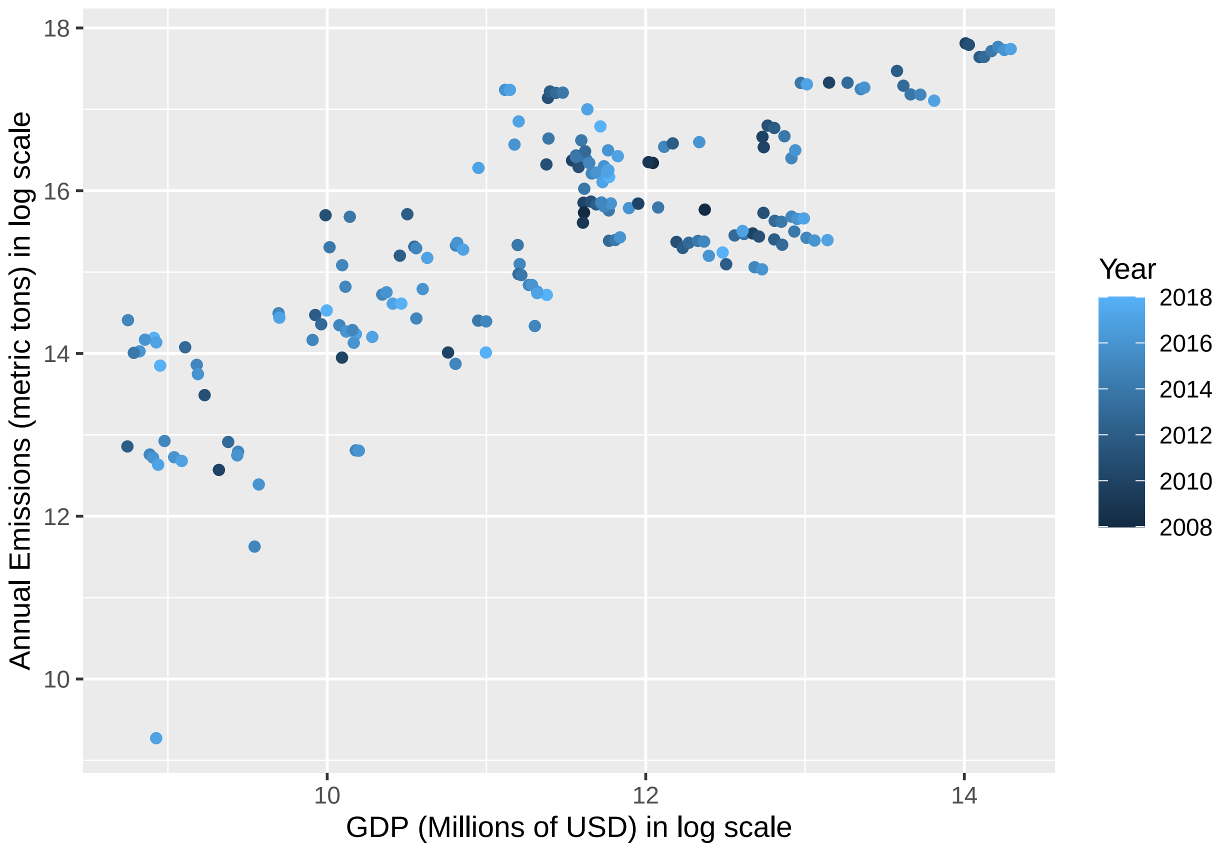
```r
# Note: At this point, there are duplicate rows and rows with wrong measurement
# year labels. To fix these two problems, we run two loops through the SORTED
# CDP_raw_AE data table:

# 1. The first loop addresses the problem with duplicates:
CDP_raw_AE[,AE:=round(AE)]
CDP_raw_AE[,Mea_Year_D:=NA]

for (i in 1:(nrow(CDP_raw_AE)-1)) {
  if ((CDP_raw_AE$City[i]==CDP_raw_AE$City[i+1]) &
      (CDP_raw_AE$Mea_Year[i]==CDP_raw_AE$Mea_Year[i+1]) &
      (CDP_raw_AE$AE[i]==CDP_raw_AE$AE[i+1])) {
    CDP_raw_AE$Mea_Year_D[i+1] <- 1
  } else {
    CDP_raw_AE$Mea_Year_D[i+1] <- 0
  }
}
CDP_raw_AE$Mea_Year_D[1] <- 0

# Remove duplicates:
CDP_raw_AE <- CDP_raw_AE[Mea_Year_D==0,]

# 2. The second loop ddresses the problem with wrong measurement year labels:
for (i in 1:(nrow(CDP_raw_AE)-1)) {
  if ((CDP_raw_AE$City[i]==CDP_raw_AE$City[i+1]) &
      (CDP_raw_AE$Mea_Year[i]>=CDP_raw_AE$Mea_Year[i+1]) &
      (CDP_raw_AE$AE[i]!=CDP_raw_AE$AE[i+1])) {
    CDP_raw_AE$Mea_Year[i+1] <- CDP_raw_AE$Mea_Year[i] +
      (CDP_raw_AE$Rep_Year[i+1] - CDP_raw_AE$Rep_Year[i])
  } else {
    CDP_raw_AE$Mea_Year[i+1] <- CDP_raw_AE$Mea_Year[i+1]
  }
}

# 3. Keep only the relevant columns:
CDP_raw_AE <- CDP_raw_AE[,c(1,3:6)]
colnames(CDP_raw_AE) <- c("ID","Year","City","AE", "GDP")
```

GDP Data:

```r
# Drop the GDP column from the CDP_raw_AE data table:
CDP_raw_AE[,GDP:=NULL]
```

```r
# Use a loop to read all the GDP data:
# Reference: https://medium.com/@MohWaitherero/getting-excel-files-into-r-7b82
# a8e3d89d
# 1. Temporarily change the working directory:
new_wd <- "D:/Academics/UCLA Anderson MFE/Applied Finance Project/2. Literature Review & Preliminary Ana
setwd(new_wd)

# 2. Create a list of gdp xls file names:
gdp.list <- list.files(pattern = '*.xls')

# 3. Load the GDP data into the GDP data frame:
GDP <- lapply(gdp.list, read_excel)

# 4. Unstack the GDP data frame:
GDP <- do.call("rbind.data.frame", GDP)

# 5. Correct the date column in the GDP data frame:
GDP <- as.data.table(GDP)
colnames(GDP) <- c("Date","GDP","City")
GDP$Date <- as.Date(GDP$Date)
GDP$Year <- year(GDP$Date)

# 6. Drop the unneeded column:
GDP <- as.data.table(GDP)
GDP[,Date:=NULL]

# 7. Merge GDP data with the CDP annual emissions data:
CDP_AE_GDP <- CDP_raw_AE %>% left_join(GDP, by=c("Year","City"))

# 8. Remove rows with missing GDP data:
CDP_AE_GDP <- as.data.table(CDP_AE_GDP)
CDP_AE_GDP <- CDP_AE_GDP[!is.na(GDP),]

# 9. Add log scale annual emissions and log scale GDP variables:
CDP_AE_GDP[,ln_GDP:=log(GDP)]
CDP_AE_GDP[,ln_AE:=log(AE)]
```

```r
# 9. Examine the idea of normalizing emissions using GDP:
ggplot(data=CDP_AE_GDP) +
  geom_point(mapping = aes(x=ln_GDP,y=ln_AE,color=Year)) +
  xlab("GDP (Millions of USD) in log scale") +
  ylab("Annual Emissions (metric tons) in log scale")
```



```
reg_AE_GDP <- lm(ln_AE ~ ln_GDP, data=CDP_AE_GDP)
print(summary(reg_AE_GDP))
```

```
##
## Call:
## lm(formula = ln_AE ~ ln_GDP, data = CDP_AE_GDP)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1839 -0.6589  0.1005  0.5538  2.0664
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.46208    0.52353   12.34   <2e-16 ***
## ln_GDP       0.78360    0.04551   17.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8629 on 179 degrees of freedom
## Multiple R-squared:  0.6236, Adjusted R-squared:  0.6215
## F-statistic: 296.5 on 1 and 179 DF,  p-value: < 2.2e-16
```

```
stargazer(reg_AE_GDP,
          title="Log Annual Emissions and log GDP",
          align=TRUE,
          type = 'text')
```

```
##
## Log Annual Emissions and log GDP
## ===============================================
##                          Dependent variable:
##                      ---------------------------
##                                 ln_AE
## -----------------------------------------------
## ln_GDP                       0.784***
##                               (0.046)
##
## Constant                     6.462***
##                               (0.524)
##
## -----------------------------------------------
## Observations                    181
## R2                             0.624
## Adjusted R2                    0.621
## Residual Std. Error      0.863 (df = 179)
## F Statistic          296.509*** (df = 1; 179)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

```
# 9. Compute the Normalized Emissions data:
CDP_AE_GDP[,Norm_AE:=AE/GDP]
```

Municipal Bonds Data:

```
# Use a loop to read all the Bond data:
# 1. Temporarily change the working directory:
new_wd_2 <- "D:/Academics/UCLA Anderson MFE/Applied Finance Project/2. Literature Review & Preliminary
setwd(new_wd_2)

# 2. Create a list of Bond xlsx file names:
bond.list <- list.files(pattern = '*.xlsx')

# 3. Load the Bond data into the Bond data frame:
Bond <- lapply(bond.list, read_excel)

# 4. Unstack the GDP data frame:
Bond <- do.call("rbind.data.frame", Bond)
```

```
# 5. Correct the date column in the Bond data frame:
Bond$Date <- as.Date(Bond$Date)
Bond$Year <- year(Bond$Date)

# 6. Drop the unneeded column:
Bond <- as.data.table(Bond)
Bond[,Date:=NULL]

# 7. Reorder and rename the columns:
Bond <- Bond[,c(6,5,1,4,2,3)]
colnames(Bond) <- c("Year","City","YTW","Mod_Dur","Mat_Year","Rating")

# 8. Merge with the Annual Emissions and GDP data table:
```

```
CDP_AE_GDP_Bond <- CDP_AE_GDP %>% left_join(Bond, by=c("Year","City"))

# 9. Remove rows with missing Bond data:
CDP_AE_GDP_Bond <- as.data.table(CDP_AE_GDP_Bond)
CDP_AE_GDP_Bond <- CDP_AE_GDP_Bond[!is.na(YTW),]

# 10. Compute years until bond's year of maturity:
CDP_AE_GDP_Bond[,Yr_t_Mat:=Mat_Year-Year]

# 11. Change bond ratings to factors:
CDP_AE_GDP_Bond$Rating <- as.factor(CDP_AE_GDP_Bond$Rating)

# 12. Keep only the variables that we need:
CDP_AE_GDP_Bond <- CDP_AE_GDP_Bond[,c(1:5,8:10,13,12)]
```

US Federal Reserve Treasury Bond Yield Data:

```
# The goal is here is to get the annual average CMT (or BEY):

# 1. Temporarily change the working directory:
new_wd_3 <- "D:/Academics/UCLA Anderson MFE/Applied Finance Project/2. Literature Review & Preliminary 
setwd(new_wd_3)

# 2. Create a list of UST rates xlsx file names:
rates.list <- list.files(pattern = '*.xlsx')

# 3. Load the UST rates data into the UST rates data frame:
Rates <- lapply(rates.list, read_excel)

# 4. Unstack the UST rates data frame:
Rates <- do.call("rbind.data.frame", Rates)

# 5. Correct the date column in the UST rates data frame:
Rates$Date <- as.Date(Rates$Date)
Rates$Year <- year(Rates$Date)

# 6. Keep the needed columns:
Rates <- as.data.table(Rates)
```

```
Rates <- Rates[,c(14,9,11,12,13)]
colnames(Rates) <- c("Year","yr_5","yr_10","yr_20","yr_30")
Rates$yr_5 <- as.numeric(Rates$yr_5)
Rates$yr_10 <- as.numeric(Rates$yr_10)
Rates$yr_20 <- as.numeric(Rates$yr_20)
Rates$yr_30 <- as.numeric(Rates$yr_30)

# 7. Obtain the annual average CMT of US Treasury Bonds:
Ann_Rates <- Rates[,list(Mean_5=mean(yr_5),
                         Mean_10=mean(yr_10),
                         Mean_20=mean(yr_20),
                         Mean_30=mean(yr_30)),
                   by = Year]
```

```
# 8. Map the annual average CMT to the CDP_AE_GDP_Bond data table:
CDP_ALL <- CDP_AE_GDP_Bond %>% left_join(Ann_Rates, by="Year")

# 9. Sort the entire data table
CDP_ALL <- as.data.table(CDP_ALL)
setkey(CDP_ALL, ID, Year)
```

Compute the YTW_spread:

```
# 1. Isolate the appropriate UST Bond Yield for the calculation:
tax_adj <- 1-0.35
CDP_ALL[,Cor_Rate:=tax_adj*ifelse(Yr_t_Mat<=5,Mean_5,
                          ifelse(Yr_t_Mat<=10,Mean_10,
                          ifelse(Yr_t_Mat<=20,Mean_20,Mean_30)))]

# 2. Get the YTW_spread variable:
CDP_ALL[,YTW_SP:=YTW-Cor_Rate]
```

Exploratory Data Analysis:

Table 1 - Number of Municipalities by Year:

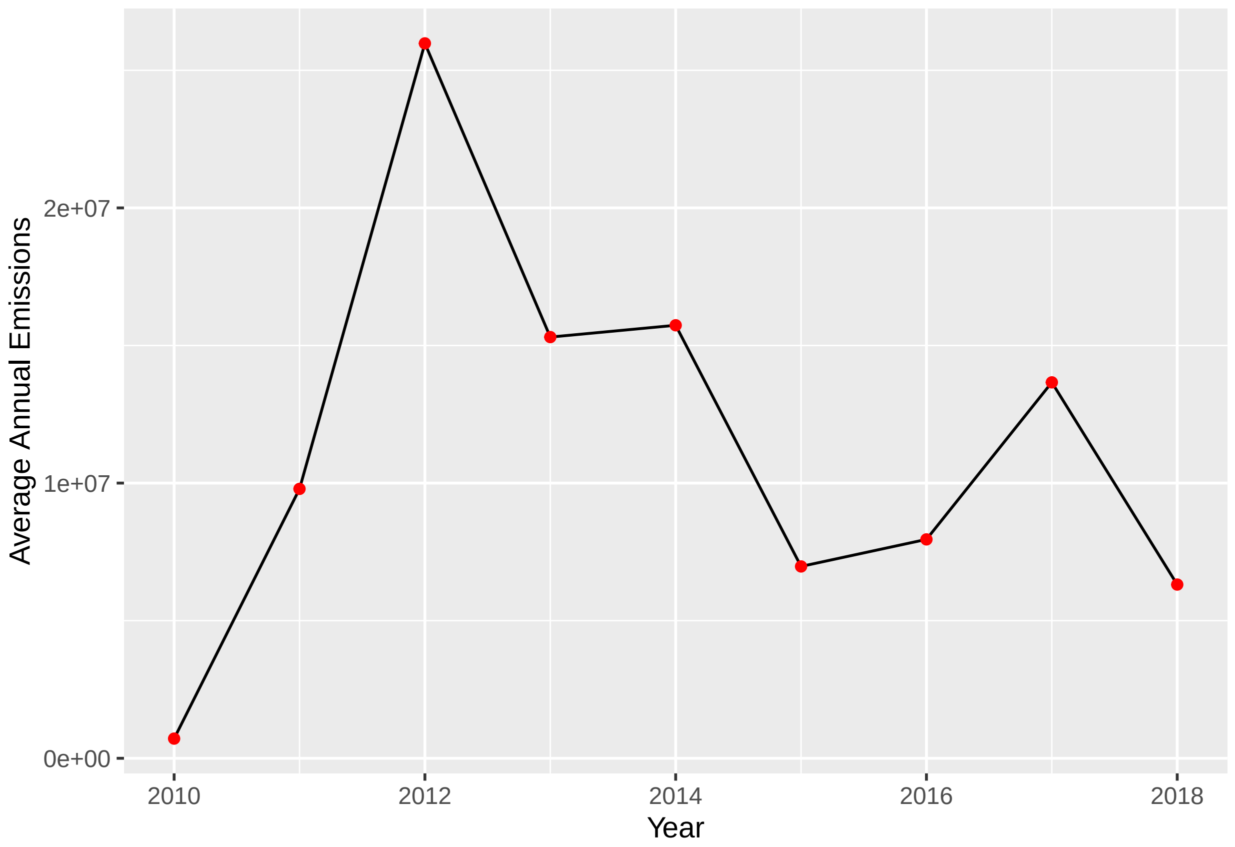
```
# Write a loop to print out the number of unique municipalities in the clean
# data table:
for (i in 2010:2018) {
  print(nrow(CDP_ALL[Year==i,]))
}
```

```
## [1] 2
## [1] 4
## [1] 4
## [1] 9
## [1] 15
## [1] 24
## [1] 19
## [1] 16
## [1] 6
```

Graph 1 - Emissions Level by Year:

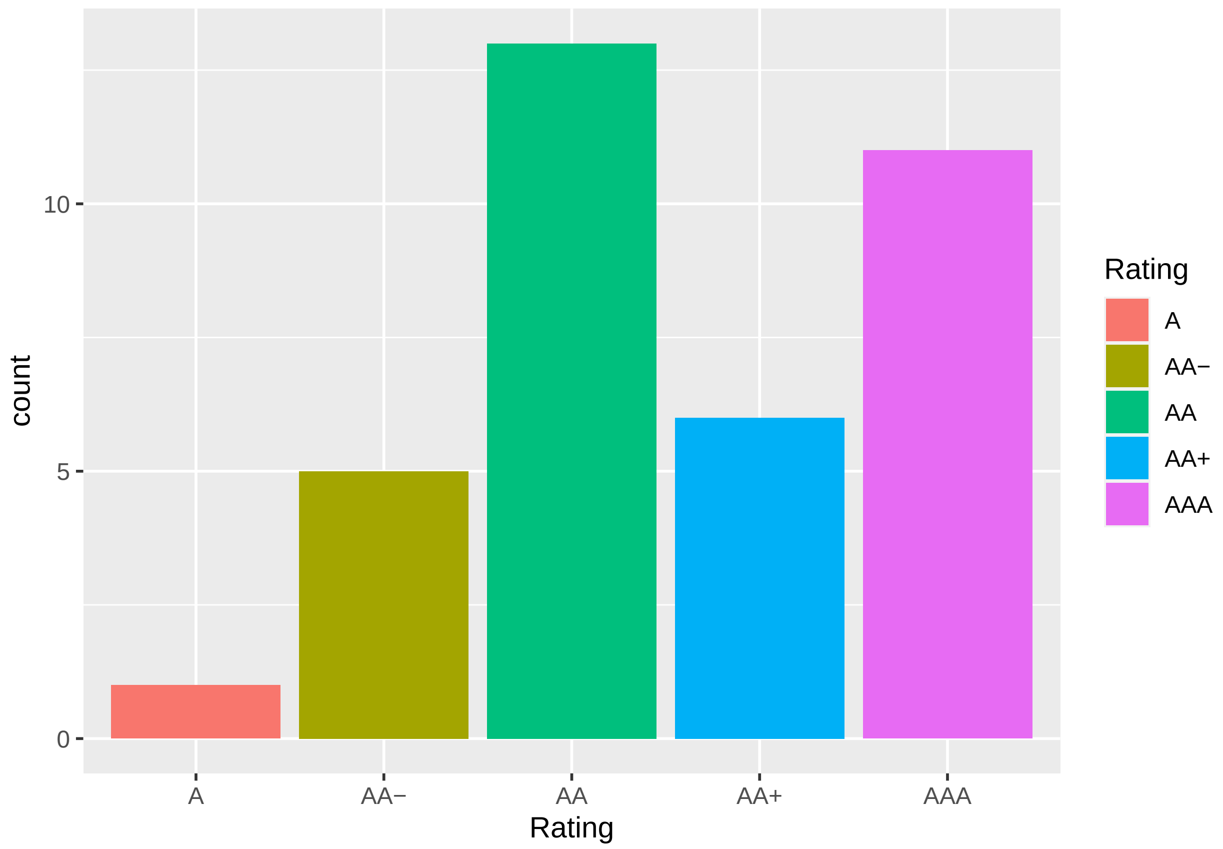
```
# Create a data table that contains the annual average emissions:
TSAE <- CDP_ALL[,list(MeanAE=mean(AE)), by = Year]
```

```
ggplot(data = TSAE, mapping = aes(x = Year, y = MeanAE)) +
  geom_line() + ylab("Average Annual Emissions") +
  geom_point(color="red")
```



Graph 2 - Distribution of Bond Ratings:

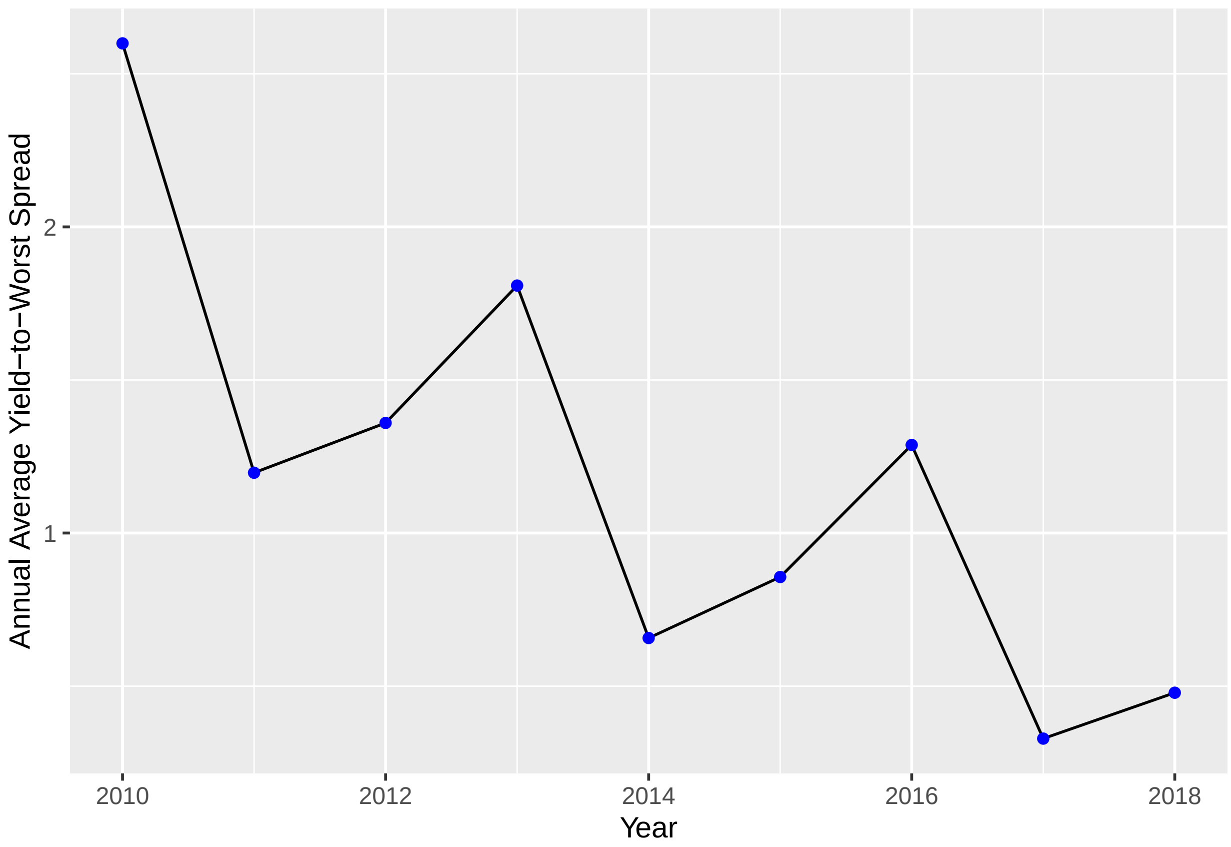
```
# Create a ratings data frame:
CDP_Ratings <- CDP_ALL %>%
  group_by(ID) %>%
  slice(1)

# Set the order of bond ratings
CDP_Ratings$Rating <- factor(CDP_Ratings$Rating,
                             levels=c("BBB+","A-","A","A+","AA-","AA",
                                      "AA+","AAA"))

ggplot(data=CDP_Ratings) +
  geom_bar(mapping=aes(x=Rating, fill=Rating))
```



Graph 3 - Annual Average of the YTW Spread:

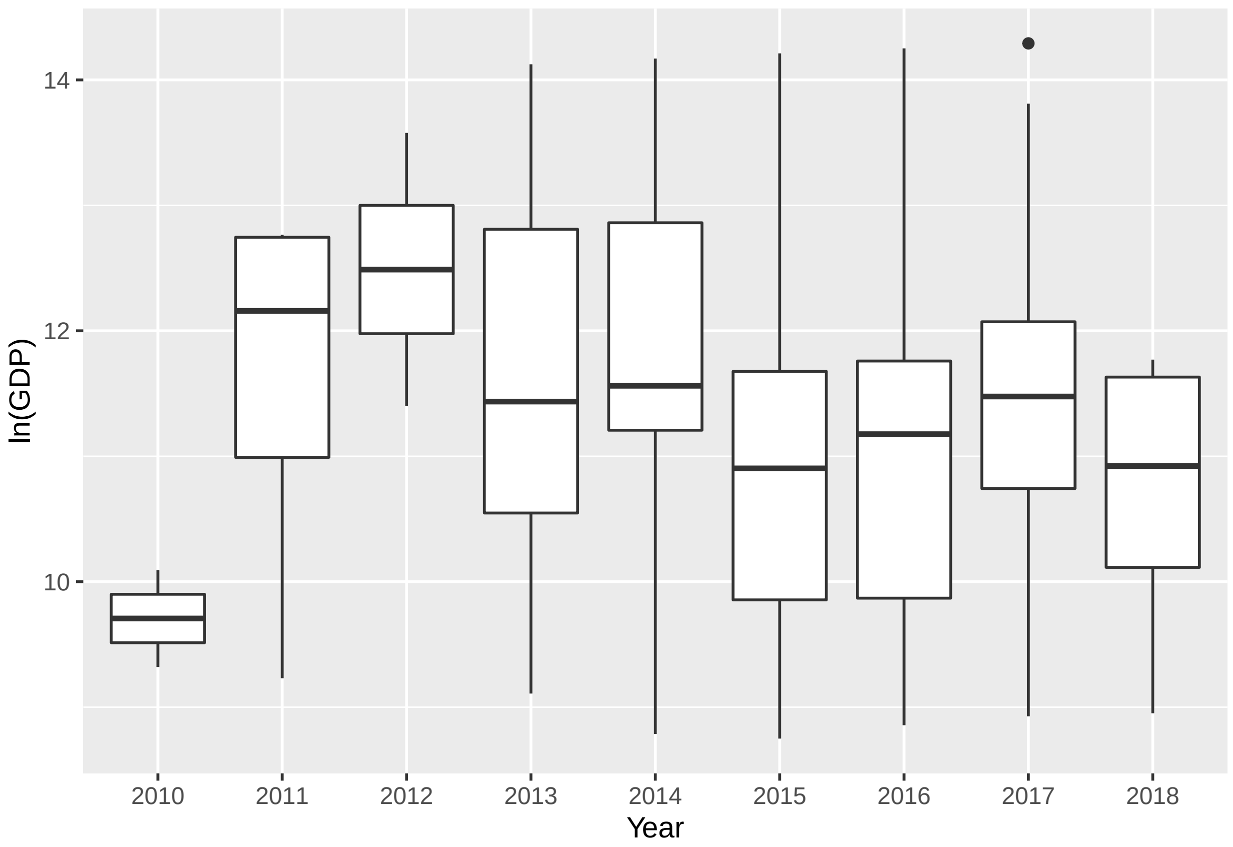
```
# Create a data table that contains the annual average of YTW spread:
Ann_Avg_Sp <- CDP_ALL[,list(MeanYTW_Sp=mean(YTW_SP)), by = Year]
```

```
ggplot(data = Ann_Avg_Sp, mapping = aes(x = Year, y = MeanYTW_Sp)) +
  geom_line() + ylab("Annual Average Yield-to-Worst Spread") +
  geom_point(color="blue")
```



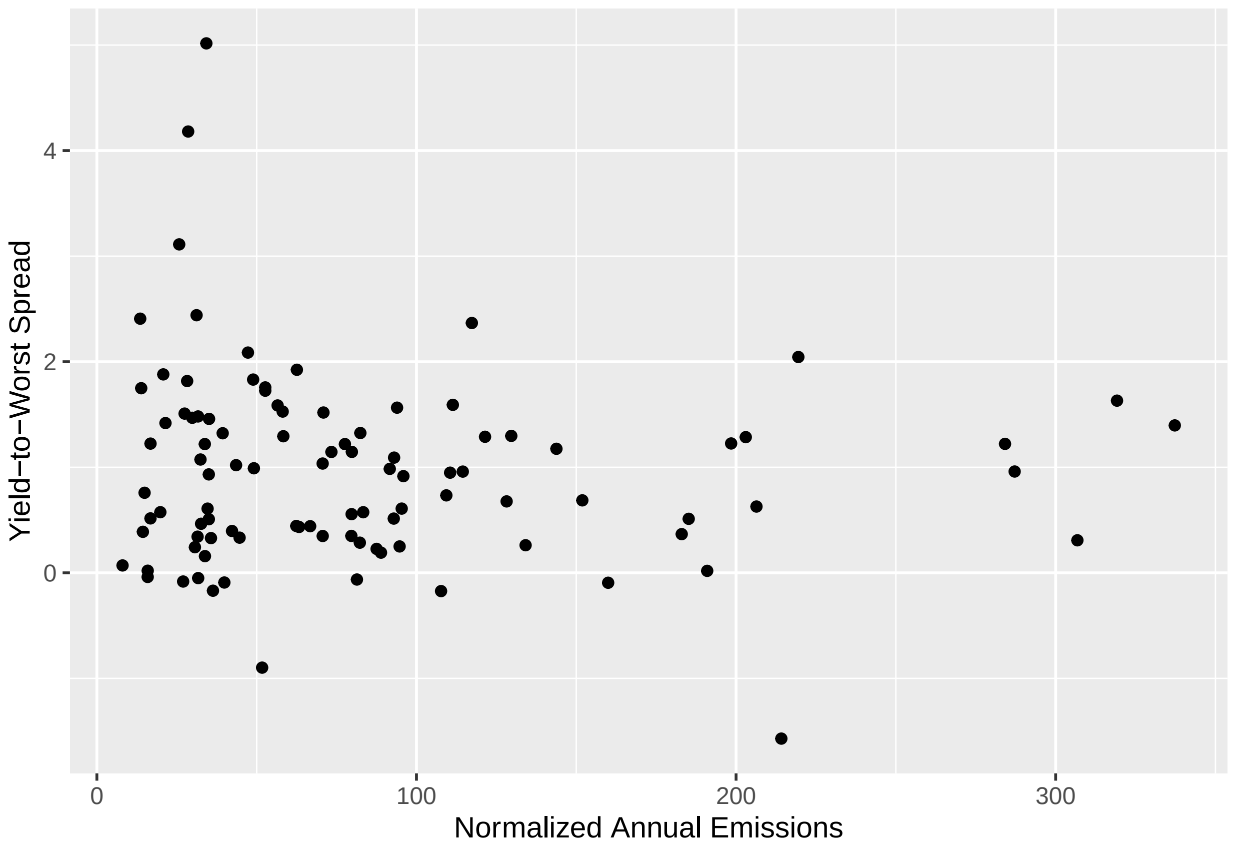
Graph 4 - Distribution of GDP by Year:

```
# Use a boxplot to distribue the distribution of GDP by year:
CDP_GDP <- CDP_ALL[,c("Year","City","GDP")]
CDP_GDP$ln_GDP <- log(CDP_GDP$GDP)

CDP_GDP$Year <- factor(CDP_GDP$Year)

ggplot(data=CDP_GDP, mapping=aes(x=Year,y=ln_GDP)) +
  geom_boxplot() +
  ylab("ln(GDP)")
```



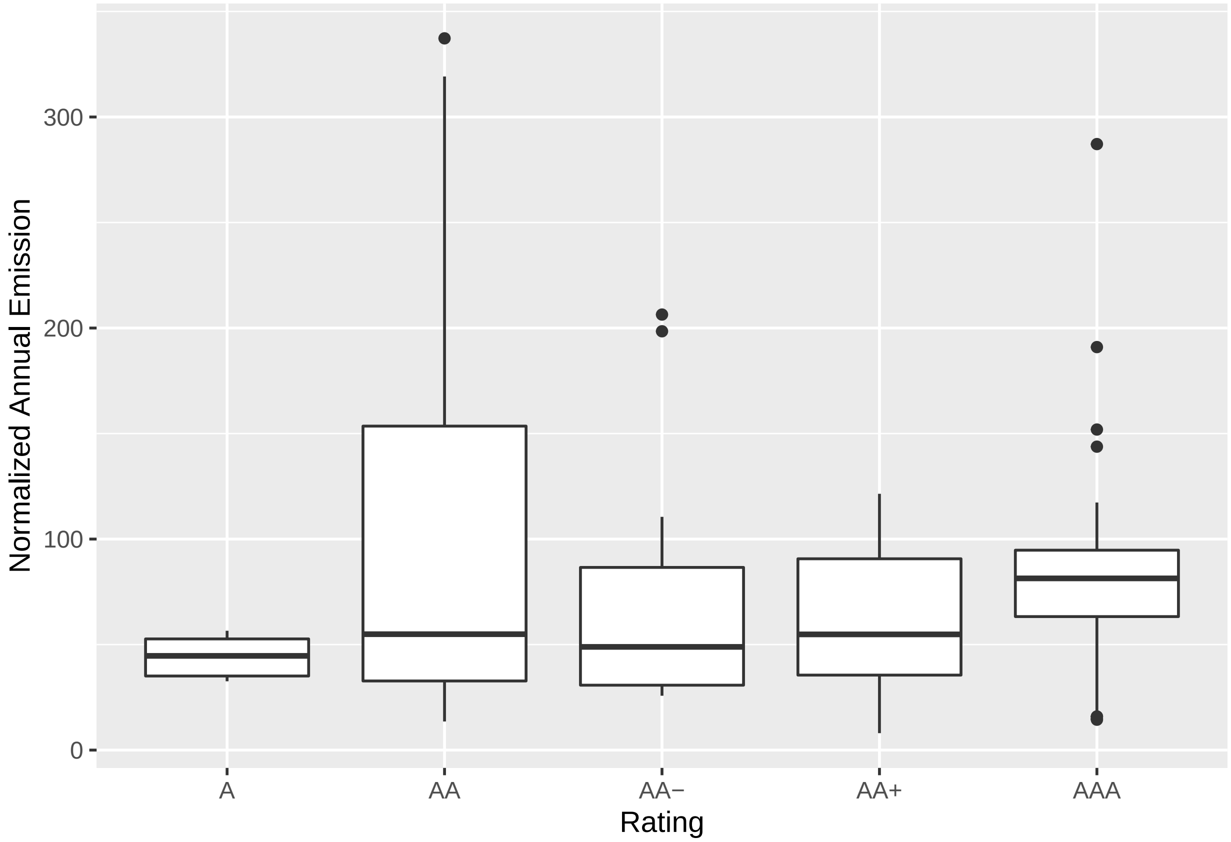
Graph 5 - Scatterplot of Yield-to-Worst Spread and Normalized Emissions

```r
ggplot(data=CDP_ALL) +
  geom_point(mapping = aes(x=Norm_AE,y=YTW_SP)) +
  xlab("Normalized Annual Emissions") +
  ylab("Yield-to-Worst Spread")
```



Graph 6 - Distribution of Normalized Annual Emissions by Ratings:

```
ggplot(data=CDP_ALL, mapping=aes(x=Rating,y=Norm_AE)) +
  geom_boxplot() +
  ylab("Normalized Annual Emission")
```



Empirical Analysis:

Prepare Data for Regression Analysis:

```
# Grab all the columns needed for regression:
CDP_Reg <- CDP_ALL[,c(1:6, 8:10, 16)]
CDP_Reg[,ln_GDP:=log(GDP)]
CDP_Reg[,AE_Mil:=AE/1000000]
print(colnames(CDP_Reg))
```

```
##  [1] "ID"       "Year"     "City"     "AE"       "GDP"      "Norm_AE"
##  [7] "Mod_Dur"  "Yr_t_Mat" "Rating"   "YTW_SP"   "ln_GDP"   "AE_Mil"
```

Correlation Matrix:

```
Reg_Matrix <- CDP_Reg[,c("Norm_AE","Mod_Dur","Yr_t_Mat","ln_GDP","AE_Mil")]
Reg_Matrix <- as.matrix(Reg_Matrix)
print(cor(Reg_Matrix))
```

```
##              Norm_AE      Mod_Dur    Yr_t_Mat     ln_GDP      AE_Mil
## Norm_AE   1.00000000 -0.166659378  0.07090726 -0.2613607  0.118650942
## Mod_Dur  -0.16665938  1.000000000  0.52964911  0.1234983  0.005351026
## Yr_t_Mat  0.07090726  0.529649111  1.00000000 -0.1601600 -0.151340675
## ln_GDP   -0.26136070  0.123498266 -0.16016001  1.0000000  0.778076487
## AE_Mil    0.11865094  0.005351026 -0.15134068  0.7780765  1.000000000
```

```
Reg_Matrix <- CDP_Reg[,c("Norm_AE","Mod_Dur","Yr_t_Mat","ln_GDP","AE_Mil")]
dur_mat <- lm(Mod_Dur ~ Yr_t_Mat, data=Reg_Matrix)
print(summary(dur_mat))
```

```
## 
## Call:
## lm(formula = Mod_Dur ~ Yr_t_Mat, data = Reg_Matrix)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.5859 -1.1127  0.0887  1.0463  5.0650 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   2.4504     0.4842   5.061 1.98e-06 ***
## Yr_t_Mat      0.2343     0.0381   6.150 1.73e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.799 on 97 degrees of freedom
## Multiple R-squared:  0.2805, Adjusted R-squared:  0.2731 
## F-statistic: 37.82 on 1 and 97 DF,  p-value: 1.73e-08
```

Univarite Regression Model:

```
reg_Norm_AE <- lm(YTW_SP ~ Norm_AE, data=CDP_Reg)
print(summary(reg_Norm_AE))
```

```
## 
## Call:
## lm(formula = YTW_SP ~ Norm_AE, data = CDP_Reg)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -2.3716 -0.6185 -0.0083  0.4664  3.9992 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1.057979   0.140806   7.514 2.85e-11 ***
## Norm_AE     -0.001198   0.001263  -0.949    0.345    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9127 on 97 degrees of freedom
## Multiple R-squared:  0.009199,   Adjusted R-squared:  -0.001015 
## F-statistic: 0.9006 on 1 and 97 DF,  p-value: 0.345
```

```
reg_Mod_Dur <- lm(YTW_SP ~ Mod_Dur, data=CDP_Reg)
print(summary(reg_Mod_Dur))
```

```
## 
## Call:
## lm(formula = YTW_SP ~ Mod_Dur, data = CDP_Reg)
## 
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.57918 -0.50143 -0.09733  0.37061  3.08537 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.32209    0.20315  -1.586    0.116    
## Mod_Dur      0.24530    0.03615   6.786 9.18e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.755 on 97 degrees of freedom
## Multiple R-squared:  0.3219, Adjusted R-squared:  0.3149 
## F-statistic: 46.04 on 1 and 97 DF,  p-value: 9.181e-10
```

```
reg_lnGDP <- lm(YTW_SP ~ ln_GDP, data=CDP_Reg)
print(summary(reg_lnGDP))
```

```
## 
## Call:
## lm(formula = YTW_SP ~ ln_GDP, data = CDP_Reg)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -2.4922 -0.6008 -0.0716  0.4905  3.8791 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)   
## (Intercept)  2.03185    0.70340   2.889  0.00477 **
## ln_GDP      -0.09541    0.06189  -1.542  0.12642   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9059 on 97 degrees of freedom
## Multiple R-squared:  0.02391,    Adjusted R-squared:  0.01385 
## F-statistic: 2.377 on 1 and 97 DF,  p-value: 0.1264
```

Multivariate Regression Model:

```
reg_multi <- lm(YTW_SP ~ Norm_AE+Mod_Dur+ln_GDP+Rating, data=CDP_Reg)
print(summary(reg_multi))
```

```
## 
## Call:
## lm(formula = YTW_SP ~ Norm_AE + Mod_Dur + ln_GDP + Rating, data = CDP_Reg)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.46194 -0.42418 -0.05481  0.36402  2.11620 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1.8888154  0.7482103   2.524  0.01332 *  
## Norm_AE     -0.0006762  0.0010047  -0.673  0.50260    
## Mod_Dur      0.2333901  0.0342628   6.812 1.01e-09 ***
## ln_GDP      -0.1471136  0.0529728  -2.777  0.00666 **
## RatingAA    -0.3866809  0.3266100  -1.184  0.23953
## RatingAA-    0.2705866  0.3922273   0.690  0.49203
## RatingAA+   -0.5244951  0.3559264  -1.474  0.14404
## RatingAAA   -0.7868249  0.3396505  -2.317  0.02277 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6696 on 91 degrees of freedom
## Multiple R-squared:  0.4997, Adjusted R-squared:  0.4612
## F-statistic: 12.98 on 7 and 91 DF,  p-value: 1.776e-11
```

```
MSE_reg_1 <- sum((reg_multi$residuals)^2)/nrow(CDP_Reg)
print(MSE_reg_1)
```

```
## [1] 0.4121088
```

```
stargazer(reg_multi,
          title="Yield-to-Worst Spread on Normalized Annual Emissions",
          align=TRUE,
          type = 'text')
```

Yield-to-Worst Spread on Normalized Annual Emissions

```
    Dependent variable:
---------------------------
          YTW_SP
```

Norm_AE -0.001 (0.001)
Mod_Dur 0.233*** (0.034)
ln_GDP -0.147*** (0.053)
RatingAA -0.387 (0.327)
RatingAA- 0.271 (0.392)
RatingAA+ -0.524 (0.356)
RatingAAA -0.787** (0.340)
Constant 1.889** (0.748)

Observations 99
R2 0.500
Adjusted R2 0.461
Residual Std. Error 0.670 (df = 91)
F Statistic 12.985*** (df = 7; 91)
=== Note: *p<0.1;* ***p<0.05;*** p<0.01

```
# Does using millions of metric tons of annual emissions make a difference?
reg_multi_2 <- lm(YTW_SP ~ AE_Mil+Mod_Dur+ln_GDP+Rating, data=CDP_Reg)
print(summary(reg_multi_2))
```

```
##
## Call:
## lm(formula = YTW_SP ~ AE_Mil + Mod_Dur + ln_GDP + Rating, data = CDP_Reg)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1.5557 -0.4225 -0.0628  0.3874  2.1497
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.6927891  0.9427314   1.796   0.0759 .
## AE_Mil      -0.0004155  0.0097160  -0.043   0.9660
## Mod_Dur      0.2355354  0.0343120   6.865 7.95e-10 ***
## ln_GDP      -0.1344756  0.0805974  -1.668   0.0987 .
## RatingAA    -0.4095482  0.3385122  -1.210   0.2295
## RatingAA-    0.2718847  0.3942045   0.690   0.4921
## RatingAA+   -0.5184382  0.3568334  -1.453   0.1497
## RatingAAA   -0.7955508  0.3403449  -2.337   0.0216 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6712 on 91 degrees of freedom
## Multiple R-squared:  0.4972, Adjusted R-squared:  0.4585
## F-statistic: 12.86 on 7 and 91 DF,  p-value: 2.198e-11
```

Random Forest Model:

```r
# Gather needed variables:
CDP_RF <- CDP_Reg[,c("Norm_AE","Mod_Dur","Rating","YTW_SP","ln_GDP")]
```

```r
# Change the ratings to dummy variables:
CDP_RF[,Rat_A:=ifelse(Rating=="A",1,0)]
CDP_RF[,Rat_AAm:=ifelse(Rating=="AA-",1,0)]
CDP_RF[,Rat_AA:=ifelse(Rating=="AA",1,0)]
CDP_RF[,Rat_AAp:=ifelse(Rating=="AA+",1,0)]
CDP_RF[,Rat_AAA:=ifelse(Rating=="A",1,0)]
CDP_RF$Rating <- NULL
```

```r
# Create the training and test set:
n <- nrow(CDP_RF)
n_train <- round(0.8*n)
set.seed(123)
train_indices <- sample(1:n, n_train)
CDP_RF_train <- CDP_RF[train_indices,]
CDP_test <- CDP_RF[-train_indices,]
```

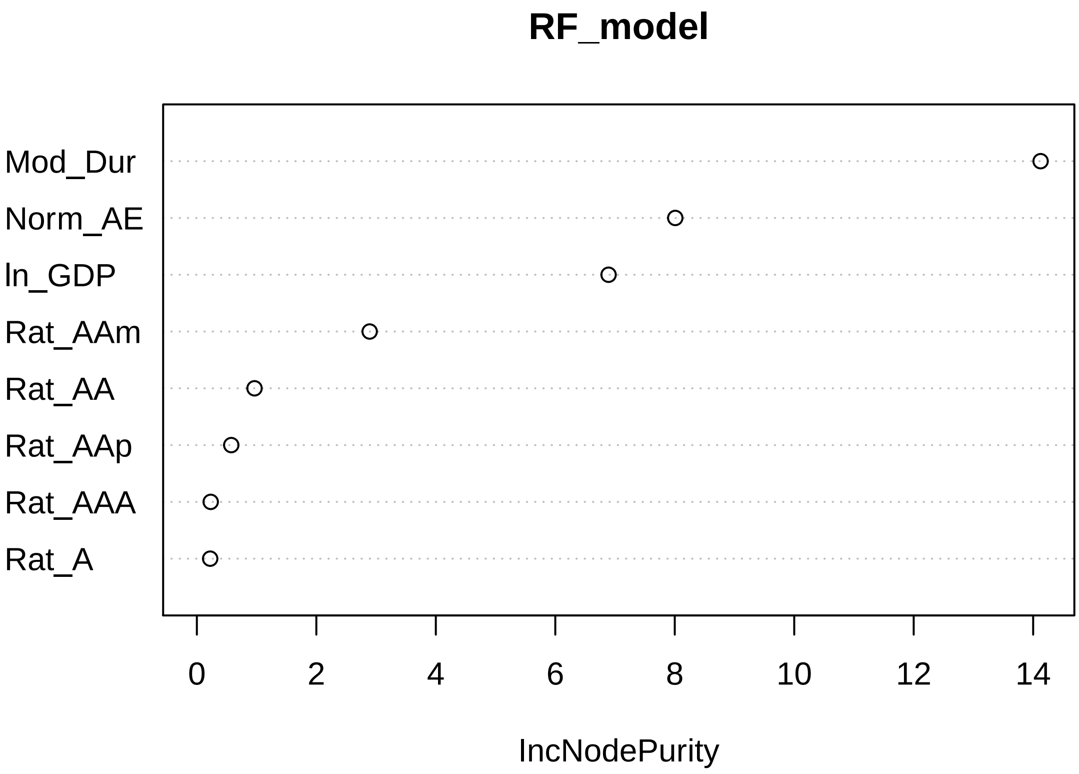
```r
# Create the RF model using the
RF_model <- randomForest(formula = YTW_SP ~ .,
                         data=CDP_RF_train,
                         mtry=2) # mtry = 2 is derived from the tuning.
print(RF_model)
```

```
##
## Call:
##  randomForest(formula = YTW_SP ~ ., data = CDP_RF_train, mtry = 2)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.4825779
##                     % Var explained: 20.26
```

```
# Check for variable importance:
importance(RF_model)
```

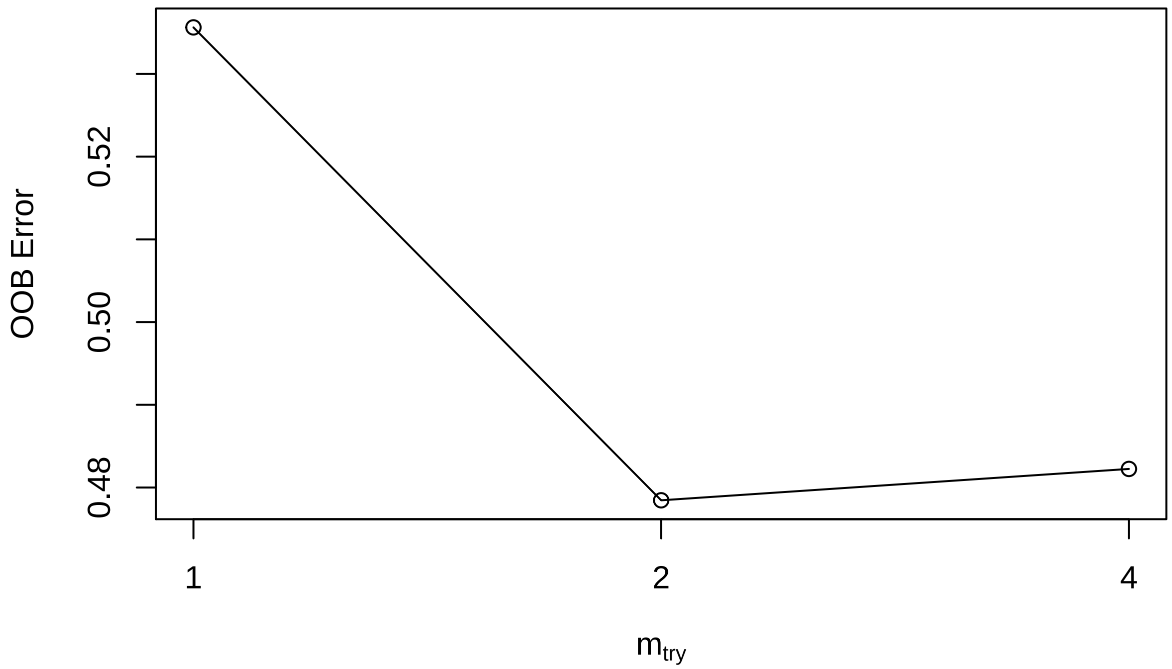
```
##           IncNodePurity
## Norm_AE       8.0083944
## Mod_Dur      14.1247243
## ln_GDP        6.8919692
## Rat_A         0.2224369
## Rat_AAm       2.8928433
## Rat_AA        0.9649671
## Rat_AAp       0.5755268
## Rat_AAA       0.2315288
```

```
# Plot variable importance
varImpPlot(RF_model)
```



```
# Tune the mtry input of the RF model:
res <- tuneRF(x = subset(CDP_RF_train, select = -YTW_SP),
              y = CDP_RF_train$YTW_SP,
              ntreeTry = 500)
```

```
## mtry = 2  OOB error = 0.4784568
## Searching left ...
## mtry = 1     OOB error = 0.5356256
## -0.1194858 0.05
## Searching right ...
## mtry = 4     OOB error = 0.4822471
## -0.00792197 0.05
```



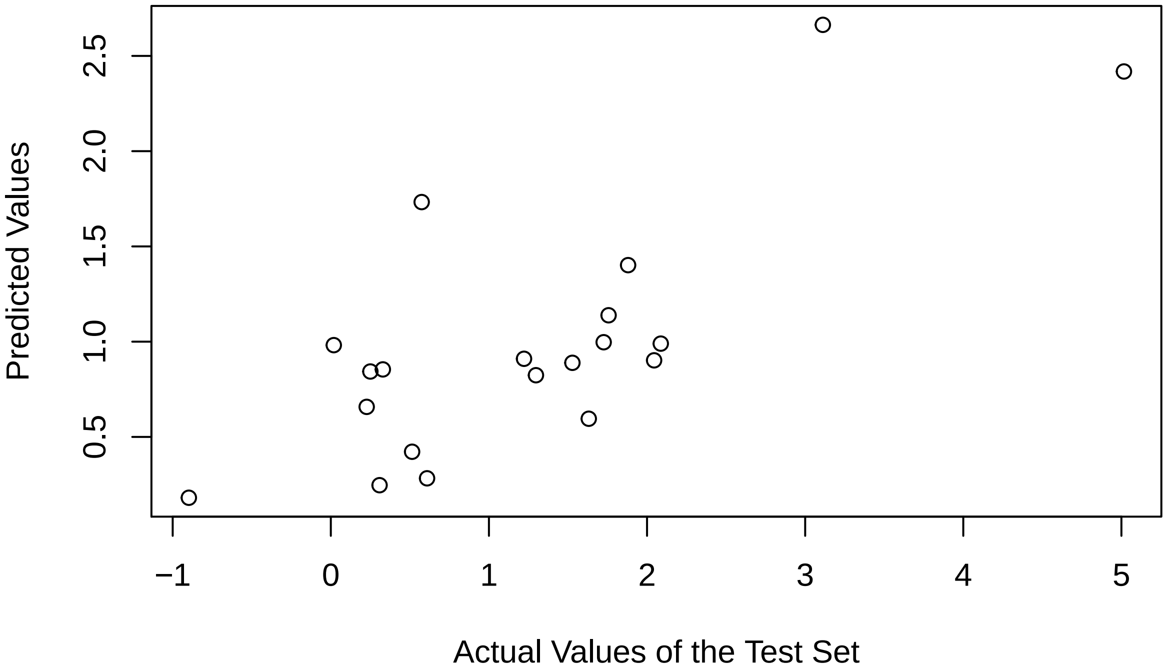
```
print(res)
```

```
##   mtry  OOBError
## 1    1 0.5356256
## 2    2 0.4784568
## 4    4 0.4822471
```

```
# Evaluate model performance on a test set:
CDP_pred <- predict(object = RF_model,
                    newdata = CDP_test)
CDP_test_act <- CDP_test$YTW_SP

MSE_RF_test <- mean((CDP_pred - CDP_test_act)^2)
print(MSE_RF_test)
```

```
## [1] 0.838424
```

```
plot(x=CDP_test_act, y=CDP_pred,
     xlab = "Actual Values of the Test Set",
     ylab = "Predicted Values")
```


Predicted Values
0.5
1.0
1.5
2.0
2.5
−1
0
1
2
3
4
5
Actual Values of the Test Set