Group-4\_Project

Jamie Bigley

6/21/2021

# Group 4 Project

Europe’s Energy Production/Consumption Review

#Library that are used for this report  
library(sqldf)  
library(ggplot2)  
library(reshape2)  
library(tidyverse)  
library(dplyr)  
library(gridExtra)  
library(gsubfn)  
library(proto)

#Load data sets for project  
#Energy information from BP  
bp\_information <- read.csv("https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/xlsx/energy-economics/statistical-review/bp-stats-review-2020-consolidated-dataset-panel-format.csv")  
  
#World population information  
World\_Pop <- read.csv(file = "World\_Pop.csv", header = TRUE, sep = ",")  
  
#Countries Area information  
areas <- read.csv(file = "countries\_area.csv")  
  
#GDP information  
df\_gdp <- read.csv(file = "API\_GDP\_dataframe.csv")

#Prepare World Population 'World\_Pop' data.frame in preparation to merge with other data.frames  
  
#Remove all the 'X' from the column titles for the years  
names(World\_Pop) <- sub("^X", "", names(World\_Pop))  
  
#Rename the column of 'Country\_Name' to 'Country'  
names(World\_Pop)[1] <- "Country"  
  
#Need to convert each year column to a year column  
World\_Pop <- melt(World\_Pop, id.vars = c("Country"))  
World\_Pop <- World\_Pop[-c(1:792),]  
  
#Need to rename the two columns year and population  
names(World\_Pop)[2] <- "Year"  
names(World\_Pop)[3] <- "Population"

#Prepare Countries Area File in preparation to merge with other dataframes  
  
#Need to clean up file by removing 3rd row  
areas <- subset(areas, select = -c(3))  
  
#Need to rename columns to 'Country' and 'Area(kms)'  
names(areas)[1] <- paste("Country")  
names(areas)[2] <- paste("Area(km2)")  
  
#Need to remove the first row  
areas <- areas[-c(1),]  
  
#Need to change area(km2) to numeric vector  
areas$`Area(km2)` <- as.numeric(as.character(areas$`Area(km2)`))

#Prepare GDP file in preparation to merge with other dataframes  
  
#Need to remove unneeded columns  
df\_gdp <- df\_gdp[-c(2:4)]  
  
#Need to rename columns  
names(df\_gdp)[1] <- paste("Country")  
names(df\_gdp) <- gsub("X", "", names(df\_gdp))  
  
#Convert each column year into a column of years and remove the "NA".   
df\_gdp <- melt(df\_gdp, id.vars = c("Country"))  
df\_gdp <- na.omit(df\_gdp)  
  
#Rename columns for year and currency  
names(df\_gdp)[2] <- paste("Year")  
names(df\_gdp)[3] <- paste("GDP.US..")  
  
#Need to change the year to a numeric vector  
df\_gdp$Year <- as.numeric(as.character(df\_gdp$Year))

#Merging all the data.frame into one data set.  
  
#Merge Area and GDP into a project data.frame  
Project <- merge(df\_gdp, areas, by = "Country")  
  
#Now add World Population to the 'Project' dataframe  
Project <- merge(Project, World\_Pop, by.x = c("Country", "Year"))  
  
#Now adding BP information to the Project dataframe  
Project <- merge(Project, bp\_information, by.x = c("Country", "Year"))  
  
#Reducing the Project dataframe to just countries with a 'SubRegion' = Europe  
Project <- Project[c(which(Project$SubRegion == "Europe")),]  
  
#Some column are NA for all of Europe. Those columns will be removed.  
Project <- subset(Project, select = -c(biodiesel\_cons\_kbd, biodiesel\_cons\_kboed, biodiesel\_cons\_ktoe,   
 biodiesel\_prod\_kbd, biodiesel\_prod\_kboed, biodiesel\_cons\_pj, ethanol\_cons\_pj,  
 biodiesel\_prod\_ktoe, biodiesel\_prod\_pj, cobalt\_kt, ethanol\_cons\_kbd,   
 ethanol\_cons\_kboed, ethanol\_cons\_ktoe, ethanol\_prod\_kbd, ethanol\_prod\_kboed),)  
  
#Remaining 'NA' are equal to zero. Replacing all NA with 0  
Project[is.na(Project)] = 0  
  
#Correcting the units of measurement for 'biofuels'  
Project$biofuels\_prod\_pj <- Project$biofuels\_prod\_pj \* 0.001  
names(Project)[24] <- paste("biofuels\_prod\_ej")  
Project$biofuels\_cons\_pj <- Project$biofuels\_cons\_pj \* 0.001  
names(Project)[20] <- paste("biofuels\_cons\_ej")  
  
#Write to complete data set to in case files change  
write.csv(Project,"Project.csv")

#Because of the number of countries in the Europe "SubRegion" the data set will be reduced to the countries with the highest GDP  
Project\_2019 <- subset(Project, Year == 2019)  
TopGDP <- top\_n(Project\_2019, 10, GDP.US..)  
TopGDP <- TopGDP$Country  
  
Project <- Project[ Project$Country %in% TopGDP, ]

# Data Clean Up

Data set was reduced to Countries with the Highest GDP

|  |  |
| --- | --- |
| Countries | Countries |
| Belgium | France |
| Germany | Italy |
| Netherlands | Poland |
| Spain | Switzerland |
| Turkey | United Kingdom |

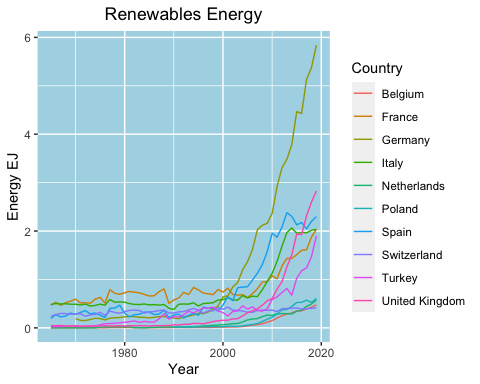
TopGDP

## [1] "Belgium" "France" "Germany" "Italy"   
## [5] "Netherlands" "Poland" "Spain" "Switzerland"   
## [9] "Turkey" "United Kingdom"

#Looking at renewables consumed from each Country.  
#First make a new data frame with just the renewable energy source.  
Total\_Renew <- Project[, c("Country", "Year", "GDP.US..", "Population",   
 "hydro\_ej", "ren\_power\_ej",   
 "renewables\_ej","solar\_ej", "wind\_ej")]  
Total\_Renew$Total <- rowSums(Total\_Renew[, 5:9])  
  
TotalRenewPlot <- ggplot(Total\_Renew, aes(x = Year, y = Total)) + geom\_line(aes(color = Country))  
TotalRenewPlot <- TotalRenewPlot + labs(title = "Renewables Energy", y = "Energy EJ") + theme(plot.title = element\_text(hjust = 0.5))  
TotalRenewPlot <- TotalRenewPlot + theme(panel.background = element\_rect(fill = "light blue",   
 color = "light blue",   
 size = 0.5,   
 linetype = "solid"))

# Renewable Energy

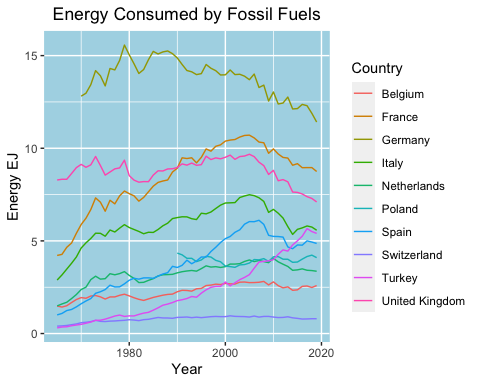
plot(TotalRenewPlot)



#Looking at fossil fuels consumed from each Country.  
#First make a new data frame with just the fossil fuels energy source.  
  
Total\_Fossil <- Project[ , c("Country", "Year", "GDP.US..", "Population",   
 "biofuels\_cons\_ej", "biogeo\_ej", "coalcons\_ej",  
 "gascons\_ej", "oilcons\_ej", "nuclear\_ej")]  
Total\_Fossil$Total <- rowSums(Total\_Fossil[, 5:10])  
  
TotalFossilPlot <- ggplot(Total\_Fossil, aes(x = Year, y = Total)) + geom\_line(aes(color = Country))  
TotalFossilPlot <- TotalFossilPlot + labs(title = "Energy Consumed by Fossil Fuels", y = "Energy EJ") +   
 theme(plot.title = element\_text(hjust = 0.5))  
TotalFossilPlot <- TotalFossilPlot + theme(panel.background = element\_rect(fill = "light blue",   
 color = "light blue",   
 size = 0.5,   
 linetype = "solid"))

# Fossil Fuels

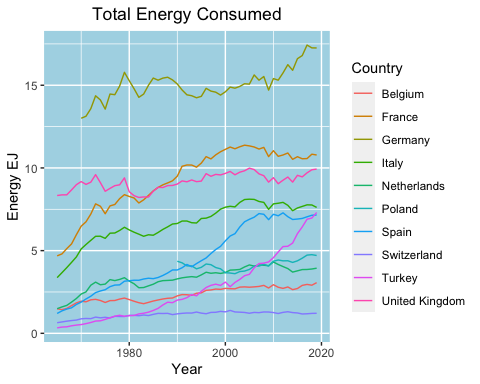
plot(TotalFossilPlot)



#Next we will look at Total Energy Consumed.  
#Combining the renewable and fossil data frames  
TotalEnergy <- Project[ , c("Country", "Year", "GDP.US..", "Population",   
 "biofuels\_cons\_ej", "biogeo\_ej", "coalcons\_ej",  
 "gascons\_ej", "oilcons\_ej", "hydro\_ej", "nuclear\_ej",   
 "ren\_power\_ej", "renewables\_ej", "solar\_ej", "wind\_ej")]  
TotalEnergy$Total <- rowSums(TotalEnergy[ , 5:15])  
TotalEnergyPlot <- ggplot(TotalEnergy, aes(x = Year, y = Total)) + geom\_line(aes(color = Country))  
TotalEnergyPlot <- TotalEnergyPlot + labs(title = "Total Energy Consumed", y = "Energy EJ") +   
 theme(plot.title = element\_text(hjust = 0.5))  
TotalEnergyPlot <- TotalEnergyPlot + theme(panel.background = element\_rect(fill = "light blue",   
 color = "light blue",   
 size = 0.5,   
 linetype = "solid"))

# Total Energy Consumed

plot(TotalEnergyPlot)

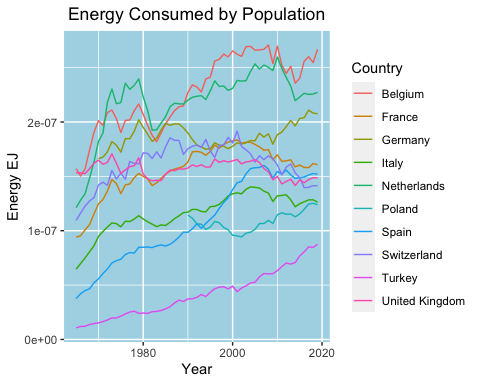


#Energy consumption looks pretty stable across each country. Try comparing this information to population and land area.

#First we look at energy per population  
#First we need to change 'Population' to a numeric   
TotalEnergy$Population <- as.numeric(TotalEnergy$Population)  
  
EnergyPerPopulationPlot <- ggplot(TotalEnergy, aes(x = Year, y = (Total/Population))) +   
 geom\_line(aes(color = Country)) + labs(title = "Energy Consumed by Population", y = "Energy EJ") +  
 theme(plot.title = element\_text(hjust = 0.5)) + theme(panel.background = element\_rect(fill = "light blue",  
 color = "light blue",  
 size = 0.5,  
 linetype = "solid"))

# Energy Consumed by Population

plot(EnergyPerPopulationPlot)

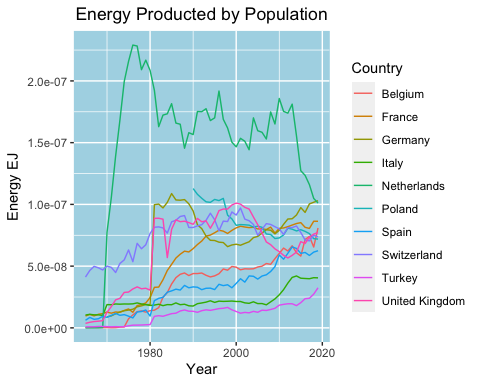


#lets but this in comparison with a population plot  
PopulationPlot <- ggplot(TotalEnergy, aes(x = Year, y = Population)) +   
 geom\_line(aes(color = Country)) + labs(title = "Population by Country", y = "Population") +  
 theme(plot.title = element\_text(hjust = 0.5)) + theme(panel.background = element\_rect(fill = "light blue",  
 color = "light blue",  
 size = 0.5,  
 linetype = "solid"))

#Production per population.  
#Create a data.frame for energy production source.   
EnergyProductionTotal <- Project[ , c("Country", "Year", "Population", "biofuels\_prod\_ej", "biogeo\_ej", "coalprod\_ej", "gasprod\_ej", "hydro\_ej", "nuclear\_ej", "ren\_power\_ej", "renewables\_ej", "solar\_ej", "wind\_ej")]  
  
#Create a column to sum all the energy production sources  
EnergyProductionTotal$Total\_Prod <- rowSums(EnergyProductionTotal[, 4:13])  
EnergyProductionTotal$Population <- as.numeric(EnergyProductionTotal$Population)  
   
  
ProductionPopulationPlot <- ggplot(EnergyProductionTotal, aes(x = Year, y = (Total\_Prod/Population))) +  
 geom\_line(aes(color = Country)) + labs(title = "Energy Producted by Population", y = "Energy EJ") +  
 theme(plot.title = element\_text(hjust = 0.5)) + theme(panel.background = element\_rect(fill = "light blue",  
 color = "light blue",  
 size = 0.5,  
 linetype = "solid"))

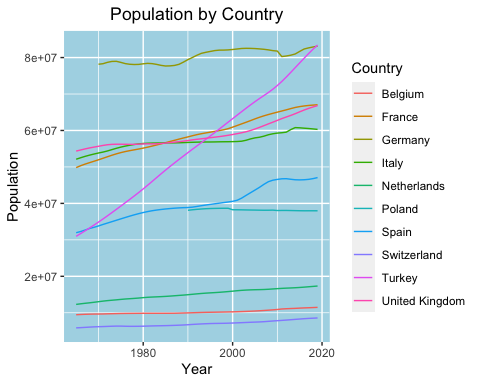
# Energy Produced by the Country Population

plot(ProductionPopulationPlot)



# Population by Country

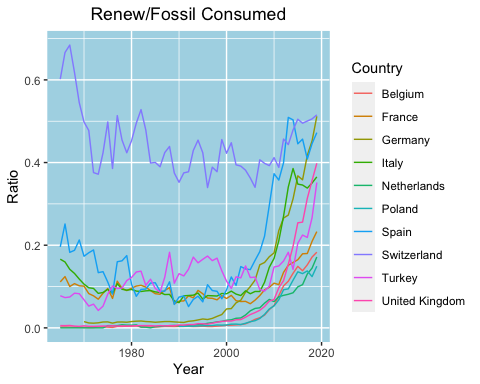
plot(PopulationPlot)



#Next lets look at the ratio of fossil fuels to renewable consumed.   
  
EnergyConRatio <- Total\_Renew[, c("Country", "Year", "Population", "Total")]  
names(EnergyConRatio)[4] <- paste("Renew\_Total(Ej)")  
EnergyConRatio$Total\_Fossil <- Total\_Fossil$Total  
names(EnergyConRatio)[5] <- paste("Total\_Fossil (Ej)")  
EnergyConRatio$Renew\_Fossil <- EnergyConRatio$`Renew\_Total(Ej)`/EnergyConRatio$`Total\_Fossil (Ej)`  
  
#Plot the ratio for each country each year  
EnergyConRatioPlot <- ggplot(EnergyConRatio, aes(x = Year, y = Renew\_Fossil)) + geom\_line(aes(color = Country)) +  
 labs(title = "Renew/Fossil Consumed", y = "Ratio") +  
 theme(plot.title = element\_text(hjust = 0.5)) + theme(panel.background = element\_rect(fill = "light blue",  
 color = "light blue",  
 size = 0.5,  
 linetype = "solid"))

# Rate of Renewable to Fossil Fuels Used

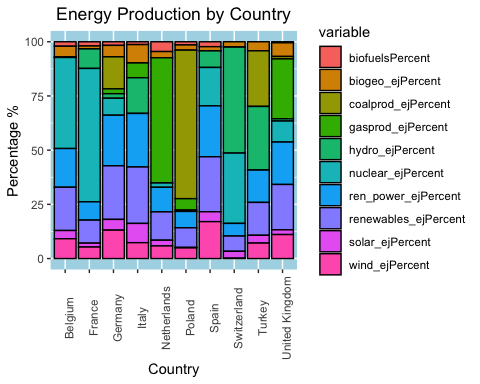
plot(EnergyConRatioPlot)



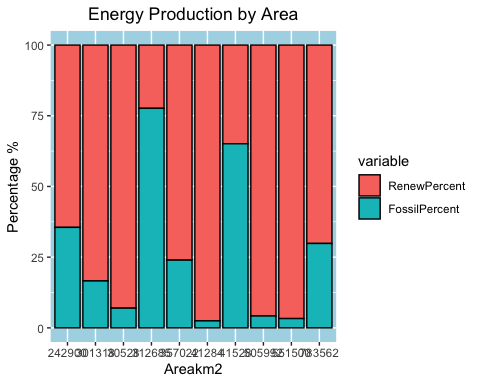
#Look at the different Energy Production based off the about of area in a country  
Prod\_2019 <- subset(Project, Year == 2019)  
ProductionArea <- Prod\_2019[, c("Country", "Area(km2)", "biofuels\_prod\_ej", "biogeo\_ej",  
 "coalprod\_ej", "gasprod\_ej", "hydro\_ej",   
 "nuclear\_ej", "ren\_power\_ej", "renewables\_ej", "solar\_ej", "wind\_ej")]  
  
ProductionArea$Total <- rowSums(ProductionArea[ , 3:12])  
ProdAreaPer <- Prod\_2019[, c("Country", "Area(km2)")]  
ProdAreaPer$biofuelsPercent <- (ProductionArea$biofuels\_prod\_ej / ProductionArea$Total)\*100  
ProdAreaPer$biogeo\_ejPercent <- (ProductionArea$biogeo\_ej / ProductionArea$Total)\*100  
ProdAreaPer$coalprod\_ejPercent <- (ProductionArea$coalprod\_ej / ProductionArea$Total)\*100  
ProdAreaPer$gasprod\_ejPercent <- (ProductionArea$gasprod\_ej / ProductionArea$Total)\*100  
ProdAreaPer$hydro\_ejPercent <- (ProductionArea$hydro\_ej / ProductionArea$Total)\*100  
ProdAreaPer$nuclear\_ejPercent <- (ProductionArea$nuclear\_ej / ProductionArea$Total)\*100  
ProdAreaPer$ren\_power\_ejPercent <- (ProductionArea$ren\_power\_ej / ProductionArea$Total)\*100  
ProdAreaPer$renewables\_ejPercent <- (ProductionArea$renewables\_ej / ProductionArea$Total)\*100  
ProdAreaPer$solar\_ejPercent <- (ProductionArea$solar\_ej / ProductionArea$Total)\*100  
ProdAreaPer$wind\_ejPercent <- (ProductionArea$wind\_ej / ProductionArea$Total)\*100  
names(ProdAreaPer)[2] <- paste("Areakm2")  
ProdAreaPer <- melt(ProdAreaPer, id.vars = c("Areakm2", "Country"))  
ProdAreaPer$Areakm2 <- as.character(ProdAreaPer$Areakm2)  
  
AreaPlot <- ggplot(ProdAreaPer, aes(x = Country, y = value, fill = variable)) + geom\_bar(position = 'stack', stat = 'identity', color = "black") + labs(title = "Energy Production by Country", y = "Percentage %", col = "Country") + theme(axis.text.x = element\_text(angle = 90)) +  
 theme(plot.title = element\_text(hjust = 0.5)) + theme(panel.background = element\_rect(fill = "light blue",  
 color = "light blue",  
 size = 0.5,  
 linetype = "solid"))

# Ratio of Energy Sources from Area of Country

plot(AreaPlot)



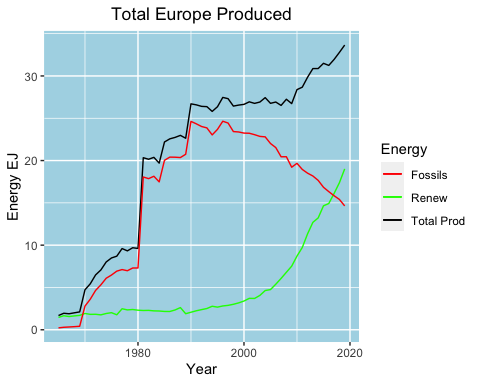
#Look at the different Energy Production based off the about of area in a country  
Prod\_2019 <- subset(Project, Year == 2019)  
ProductionArea <- Prod\_2019[, c("Country", "Area(km2)", "biofuels\_prod\_ej", "biogeo\_ej",  
 "coalprod\_ej", "gasprod\_ej", "hydro\_ej",   
 "nuclear\_ej", "ren\_power\_ej", "renewables\_ej", "solar\_ej", "wind\_ej")]  
  
ProductionArea$Total <- rowSums(ProductionArea[ , 3:12])  
ProductionArea$Renew\_Total <- rowSums(ProductionArea[ , 7:12])  
ProductionArea$Fossil\_Total <- rowSums(ProductionArea[ , 3:6])  
ProductionArea$RenewPercent <- (ProductionArea$Renew\_Total/ProductionArea$Total) \* 100  
ProductionArea$FossilPercent <- (ProductionArea$Fossil\_Total/ProductionArea$Total) \* 100  
ProdAreaPer <- ProductionArea[, c("Country", "Area(km2)", "RenewPercent", "FossilPercent")]  
names(ProdAreaPer)[2] <- paste("Areakm2")  
ProdAreaPer <- melt(ProdAreaPer, id.vars = c("Areakm2", "Country"))  
ProdAreaPer$Areakm2 <- as.character(ProdAreaPer$Areakm2)  
  
AreaPlot <- ggplot(ProdAreaPer, aes(x = Areakm2, y = value, fill = variable)) + geom\_bar(position = 'stack', stat = 'identity', color = "black") + labs(title = "Energy Production by Area", y = "Percentage %") + labs(col = "Energy") +   
 theme(plot.title = element\_text(hjust = 0.5)) + theme(panel.background = element\_rect(fill = "light blue",  
 color = "light blue",  
 size = 0.5,  
 linetype = "solid"))  
plot(AreaPlot)



#Looking at at Europe as a hole  
#Combine all 10 countries into Europe  
EuropeEnergyProd <- Project[, c("Year", "Population", "biofuels\_prod\_ej", "biogeo\_ej",   
"coalprod\_ej", "gasprod\_ej","nuclear\_ej", "hydro\_ej", "ren\_power\_ej", "renewables\_ej",   
"solar\_ej", "wind\_ej")]  
EuropeEnergyProd$Population <- as.numeric(EuropeEnergyProd$Population)  
EuropeEnergyProd$Total\_Prod <- rowSums(EuropeEnergyProd[, 3:12])  
EuropeEnergyProd$Renew\_Total <- rowSums(EuropeEnergyProd[, 8:12])  
EuropeEnergyProd$Fossil\_Total <- rowSums(EuropeEnergyProd[, 3:7])  
  
EuropeEnergyProd$RenewPercent <- (EuropeEnergyProd$Renew\_Total/EuropeEnergyProd$Total\_Prod) \* 100  
EuropeEnergyProd$FossilPercent <- (EuropeEnergyProd$Fossil\_Total/EuropeEnergyProd$Total\_Prod) \* 100  
  
#Plot total Production in Europe  
EuropeEnergyProdTotal <- EuropeEnergyProd[, c("Year", "Total\_Prod")]  
EuropeEnergyProdTotal <- aggregate(EuropeEnergyProdTotal, by = list(Category = EuropeEnergyProdTotal$Year), FUN = sum)  
EuropeEnergyProdTotal <- subset(EuropeEnergyProdTotal, select = -c(Year))  
names(EuropeEnergyProdTotal)[1] <- paste("Year")  
  
#Repeat this for the Renew and Fossil Totals  
EuropeRenew <- EuropeEnergyProd[, c("Year", "Renew\_Total")]  
EuropeRenew <- aggregate(EuropeRenew, by = list(Category = EuropeRenew$Year), FUN = sum)  
EuropeRenew <- subset(EuropeRenew, select = -c(Year))  
names(EuropeRenew)[1] <- paste("Year")  
  
EuropeFossil <- EuropeEnergyProd[, c("Year", "Fossil\_Total")]  
EuropeFossil <- aggregate(EuropeFossil, by = list(Category = EuropeFossil$Year), FUN = sum)  
EuropeFossil <- subset(EuropeFossil, select = -c(Year))  
names(EuropeFossil)[1] <- paste("Year")  
  
EuropeEnergyProdTotal$Renew\_Total <- EuropeRenew$Renew\_Total  
EuropeEnergyProdTotal$Fossil\_Total <- EuropeFossil$Fossil\_Total  
#EuropeEnergyProdTotal <- melt(EuropeEnergyProdTotal, id.vars = "Year")  
  
  
EuropeEnergyProdTotalPlot <- ggplot(EuropeEnergyProdTotal, aes(x = Year)) + geom\_line(aes(y = Total\_Prod, color = "Total Prod")) +   
 geom\_line(aes(y = Renew\_Total, color = "Renew")) +   
 geom\_line(aes(y = Fossil\_Total, color = "Fossils")) +   
 scale\_color\_manual(values = c("Red", "Green", "Black")) +   
 labs(title = "Total Europe Produced", y = "Energy EJ", col = "Energy") + theme(plot.title = element\_text(hjust = 0.5)) +   
 theme(panel.background = element\_rect(fill = "light blue", color = "light blue", size = 0.5, linetype = "solid"))

# Europe Energy Production

plot(EuropeEnergyProdTotalPlot)



#We will be unable to predict when renewable energy production will over take Fossil fuel energy production   
#energy because the graph above shows this has already taken place. Therefore, we use the information from 1985 - 2010   
#to train a prediction model and see how accurate it is.  
  
EurEnergylm <- Project[, c("Year", "Population", "biofuels\_prod\_ej", "biogeo\_ej",   
"coalprod\_ej", "gasprod\_ej","nuclear\_ej", "hydro\_ej", "ren\_power\_ej", "renewables\_ej",   
"solar\_ej", "wind\_ej")]  
EurEnergylm$Population <- as.numeric(EurEnergylm$Population)  
EurEnergylm <- aggregate(EurEnergylm, by = list(Category = EurEnergylm$Year), FUN = sum)  
EurEnergylm <- subset(EurEnergylm, select = -c(Year),)  
names(EurEnergylm)[1] <- paste("Year")  
  
#Adding Total or overall energy produced, then total by renewable and fossil fuels  
EurEnergylm$Total\_Prod <- rowSums(EurEnergylm[, 3:12])  
EurEnergylm$Renew\_Total <- rowSums(EurEnergylm[,8:12])  
EurEnergylm$Fossil\_Total <- rowSums(EurEnergylm[, 3:7])  
  
#Adding columns of percentages of renewable and fossil fuels produced.   
EurEnergylm$RenewPercent <- (EurEnergylm$Renew\_Total/EurEnergylm$Total\_Prod) \* 100  
EurEnergylm$FossilPercent <- (EurEnergylm$Fossil\_Total/EurEnergylm$Total\_Prod) \* 100  
  
  
Energylm <- EurEnergylm[21:55, ]  
  
Energylmodel <- lm(Energylm$RenewPercent ~ Energylm$biofuels\_prod\_ej + Energylm$biogeo\_ej +  
 Energylm$coalprod\_ej + Energylm$gasprod\_ej + Energylm$nuclear\_ej + Energylm$hydro\_ej +   
 Energylm$ren\_power\_ej + Energylm$renewables\_ej + Energylm$solar\_ej + Energylm$wind\_ej)  
summary(Energylmodel)

##   
## Call:  
## lm(formula = Energylm$RenewPercent ~ Energylm$biofuels\_prod\_ej +   
## Energylm$biogeo\_ej + Energylm$coalprod\_ej + Energylm$gasprod\_ej +   
## Energylm$nuclear\_ej + Energylm$hydro\_ej + Energylm$ren\_power\_ej +   
## Energylm$renewables\_ej + Energylm$solar\_ej + Energylm$wind\_ej)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.52666 -0.12967 -0.00896 0.09572 0.57717   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11.01250 1.03582 10.632 9.18e-11 \*\*\*  
## Energylm$biofuels\_prod\_ej -0.92740 3.48165 -0.266 0.79214   
## Energylm$biogeo\_ej 6.88053 1.38911 4.953 4.21e-05 \*\*\*  
## Energylm$coalprod\_ej -0.37760 0.06369 -5.929 3.45e-06 \*\*\*  
## Energylm$gasprod\_ej -0.31173 0.15807 -1.972 0.05975 .   
## Energylm$nuclear\_ej -0.49541 0.16813 -2.947 0.00686 \*\*   
## Energylm$hydro\_ej 2.82953 0.29068 9.734 5.51e-10 \*\*\*  
## Energylm$ren\_power\_ej -0.43957 1.76047 -0.250 0.80487   
## Energylm$renewables\_ej 6.19898 1.76354 3.515 0.00170 \*\*   
## Energylm$solar\_ej -1.51779 1.11847 -1.357 0.18690   
## Energylm$wind\_ej NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2504 on 25 degrees of freedom  
## Multiple R-squared: 0.9998, Adjusted R-squared: 0.9997   
## F-statistic: 1.428e+04 on 9 and 25 DF, p-value: < 2.2e-16

Energylmodel <- lm(Energylm$RenewPercent ~ Energylm$biogeo\_ej +Energylm$nuclear\_ej +   
 Energylm$hydro\_ej + Energylm$renewables\_ej)  
summary(Energylmodel)

##   
## Call:  
## lm(formula = Energylm$RenewPercent ~ Energylm$biogeo\_ej + Energylm$nuclear\_ej +   
## Energylm$hydro\_ej + Energylm$renewables\_ej)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.04210 -0.21894 0.03886 0.22154 0.89396   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.8260 0.9334 6.242 7.11e-07 \*\*\*  
## Energylm$biogeo\_ej 10.9222 1.3444 8.124 4.54e-09 \*\*\*  
## Energylm$nuclear\_ej -0.5403 0.1223 -4.417 0.00012 \*\*\*  
## Energylm$hydro\_ej 2.6767 0.3945 6.785 1.59e-07 \*\*\*  
## Energylm$renewables\_ej 4.9960 0.3339 14.964 1.87e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4349 on 30 degrees of freedom  
## Multiple R-squared: 0.9993, Adjusted R-squared: 0.9992   
## F-statistic: 1.064e+04 on 4 and 30 DF, p-value: < 2.2e-16

#Create Training dataframe  
EurEnergylm <- EurEnergylm[21:55, ] #Removing data before 1985.  
set.seed(23)  
nrows <- nrow(EurEnergylm)  
sample\_size <- round(0.8\*nrows)  
  
#Create random set of numbers  
training\_index <- sample(1:nrows, size = sample\_size, replace = FALSE)  
  
#Create training dataframe of 80%  
EnergyTrain <- EurEnergylm[training\_index, ]  
  
#Create test dataframe of 20%  
EnergyTest <- EurEnergylm[-training\_index, ]  
  
library(e1071)  
  
liner\_svm <- svm(RenewPercent ~ biofuels\_prod\_ej + biogeo\_ej + coalprod\_ej + gasprod\_ej + nuclear\_ej + hydro\_ej +   
 ren\_power\_ej + renewables\_ej + solar\_ej + wind\_ej, data = EnergyTrain, type = "eps-regression", kernel = "linear")  
  
liner\_svm

##   
## Call:  
## svm(formula = RenewPercent ~ biofuels\_prod\_ej + biogeo\_ej + coalprod\_ej +   
## gasprod\_ej + nuclear\_ej + hydro\_ej + ren\_power\_ej + renewables\_ej +   
## solar\_ej + wind\_ej, data = EnergyTrain, type = "eps-regression",   
## kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: eps-regression   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.1   
## epsilon: 0.1   
##   
##   
## Number of Support Vectors: 4

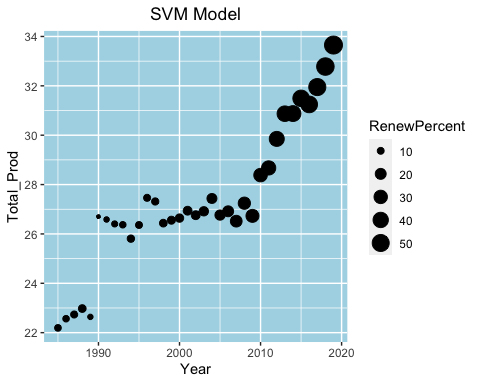
pred\_Renew <- predict(liner\_svm, newdata = EnergyTest)  
SVMError <- (EurEnergylm - pred\_Renew)  
  
sqrt(mean((EnergyTest$RenewPercent - pred\_Renew)^2))

## [1] 1.610993

library(Metrics)  
rmse(EnergyTest$RenewPercent, pred\_Renew)

## [1] 1.610993

SVMPlot <- ggplot(data = EurEnergylm, aes(x = Year, y = Total\_Prod)) + geom\_point(aes(size = RenewPercent)) +   
 labs(title = "SVM Model") + theme(plot.title = element\_text(hjust = 0.5)) + theme(panel.background = element\_rect(fill = "light blue", color = "light blue", size = 0.5, linetype = "solid"))  
plot(SVMPlot)



#Create a variable for "RenewGreat50"  
# If 0 Renew % < 50% or 1 then Renew % > 50%  
EurEnergylm$RenewGreat50 <- as.factor(ifelse(EurEnergylm$RenewPercent > 50, 1, 0))  
  
#Rerun training and test data frames.  
nrows <- nrow(EurEnergylm)  
sample\_size <- round(0.8\*nrows)  
training\_index <- sample(1:nrows, size = sample\_size, replace = FALSE)  
EnergyTrain <- EurEnergylm[training\_index, ]  
EnergyTest <- EurEnergylm[-training\_index, ]  
  
library(kernlab)  
modelksvm <- ksvm(RenewGreat50 ~., data = EnergyTrain, kernel = "rbfdot", cost = 50, scale = FALSE)  
modelksvmPred <- predict(modelksvm, EurEnergylm, type = "vote")  
comptable <- data.frame(EurEnergylm[,18], modelksvmPred[2,])  
table(comptable)

## modelksvmPred.2...  
## EurEnergylm...18. 0 1  
## 0 32 0  
## 1 0 3

#Calcutaling the error  
modelksvmError <- ((32 - 3)/ 36)  
modelksvmError

## [1] 0.8055556

#Creating the model   
modelksvmPlot <- data.frame(EurEnergylm$Year, EurEnergylm$Total\_Prod, modelksvmError)  
colnames(modelksvmPlot) <- c("Year", "Total\_Prod", "Error")  
ModelksvmPlot <- ggplot(modelksvmPlot, aes(x = Year, y = Total\_Prod)) + geom\_point(aes(size = modelksvmPred[2,], color = "Production"))  
ModelksvmPlot <- ModelksvmPlot + labs(title = "KSVM Model", y = "Energy Production") +  
 theme(plot.title = element\_text(hjust = 0.5)) +   
 theme(panel.background = element\_rect(fill = "light blue",color = "light blue", size = 0.5, linetype = "solid")) +  
 scale\_color\_manual(values = c("Orange")) + labs(col = "Energy Produced")  
  
plot(ModelksvmPlot)

