Aaron's Week 6 Update

Aaron Swoboda

2025-05-09

This document implements the Climate and Economic models described in sections 13.1 and 13.2. It first describes the five box carbon dioxide model as a function of yearly emissions. We then calculate temperature anomalies from additional radiative forcing in the atmosphere and ocean with increased carbon dioxide concentrations. We then apply the model to historic emissions and various future scenarios. We then use the Kaya Identity to decompose emissions into the four drivers of emissions values, namely Population, GDP per capita, energy intensity of GDP, and emissions intensity of energy. We estimate historic trends in these variables using past data that form the basis of future forecasts. The economic module also incorporates insights from the Solow Growth Model to better predict economic output that accounts for depreciation of capital, declining marginal product of capital and labor, and even technological growth. Finally, we disaggregate the data - instead of only examining global variables, we allow the four Kaya Identity variables to differ across economic output levels from rich, middle, and poorest regions of the world.

Climate Module (Lab 1)

Emissions to CO2

Five Box model of carbon dioxide concentrations. Each box follows equation 13.1:

$$CO2_{i,t} = (1 - CO2decay) * CO2_{i,t-1} + CO2convert * CO2share_i * CO2emissions_{t-1}$$

Make this into a function for regular and repeated use.

```
CO2difference <- function(CO2previous, newCO2emissions) {
  CO2decay <- c(0, 1-exp(-1/363), 1-exp(-1/74), 1-exp(-1/17), 1-exp(-1/2))
  CO2share <- c(0.13, 0.20, 0.32, 0.25, 0.10);
  CO2convert <- 1/2.13/1000;

  CO2concnew = (1-CO2decay)*CO2previous + CO2convert*CO2share*newCO2emissions # Equation 13.1
  CO2concnew
}
```

Now we just need some emissions data and starter values for the model.

```
# emissions data
emissionsSince1750 <- read_sheet("https://docs.google.com/spreadsheets/d/15gnvwp5HWqPVb4_hODz6j1DlLiaQv
```

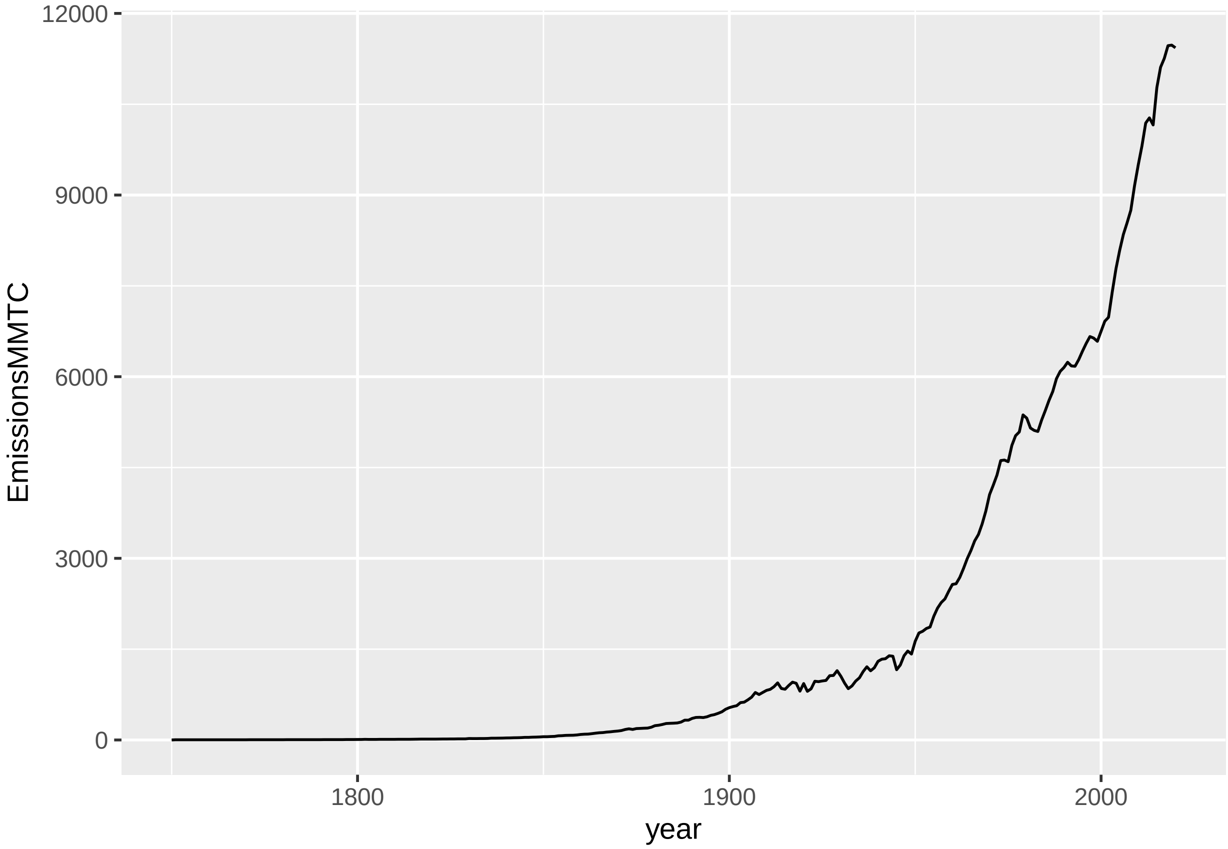
Load some data

```
## v Reading from "ECON269-Emissions-data".

## v Range '2:10000000'.

## New names:
## * `` -> `...1`
```

```r
names(emissionsSince1750) <- c("year", "EmissionsMMTC")
head(emissionsSince1750)
```

```
## # A tibble: 6 x 2
##    year EmissionsMMTC
##   <dbl>         <dbl>
## 1  1750             0
## 2  1751             3
## 3  1752             3
## 4  1753             3
## 5  1754             3
## 6  1755             3
```

```r
# Graph emissions over time
emissionsSince1750 %>%
  ggplot(aes(year, EmissionsMMTC)) +
  geom_line()
```



Prepare the data for the CO2 model calculations.

```r
HistoricData <- emissionsSince1750

HistoricData$Box1 <- 0
HistoricData$Box2 <- 0
HistoricData$Box3 <- 0
HistoricData$Box4 <- 0
HistoricData$Box5 <- 0
HistoricData$Box1[1] <- 275
```

```
(Box1col <- which(colnames(HistoricData) == "Box1"))
```

```
## [1] 3
```

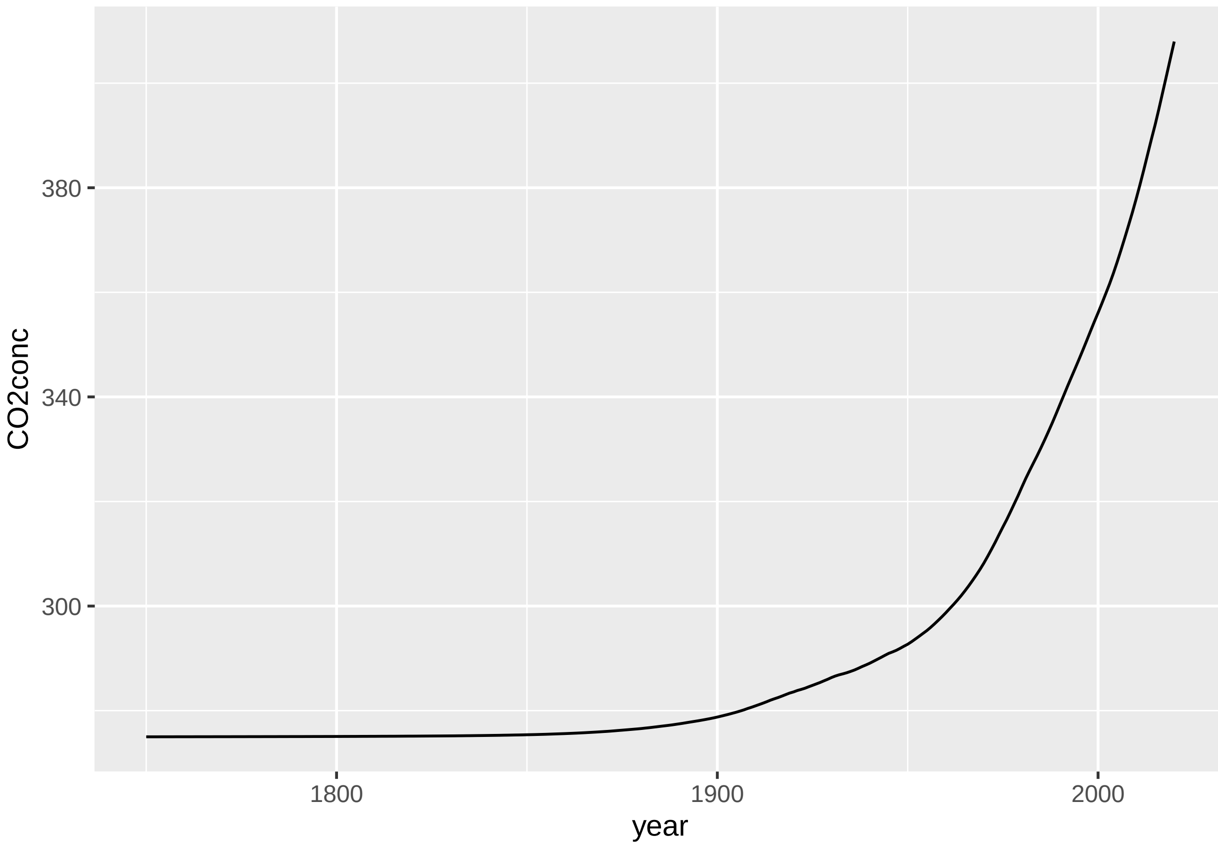
Now we implement it, year by year to use the previous year's box values and the emissions to calculate new box values.

```
for (i in 2:length(HistoricData$EmissionsMMTC)) {
 HistoricData[i, Box1col:(Box1col+4)] <- CO2difference(HistoricData[i - 1, Box1col:(Box1col+4)],
                                                     HistoricData$EmissionsMMTC[i - 1])
}

HistoricData$CO2conc <- HistoricData$Box1 + HistoricData$Box2 + HistoricData$Box3 +
  HistoricData$Box4 + HistoricData$Box5

tail(HistoricData)
```

```
## # A tibble: 6 x 8
##    year EmissionsMMTC  Box1  Box2  Box3  Box4  Box5 CO2conc
##   <dbl>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1  2015        10775.  300.  34.8  41.2  15.2  1.19    392.
## 2  2016        11110.  300.  35.7  42.3  15.6  1.23    395.
## 3  2017        11257.  301.  36.6  43.4  16.0  1.27    398.
## 4  2018        11467.  302.  37.6  44.5  16.4  1.30    402.
## 5  2019        11477.  302.  38.6  45.6  16.8  1.33    405.
## 6  2020        11432.  303.  39.5  46.7  17.2  1.34    408.
```

We can plot the CO2 concentration over time.

```
HistoricData %>%
  ggplot(aes(x = year, y = CO2conc)) +
  geom_line()
```



CO2 concentrations to Temperatures

Radiative Forcing (Equation 13.3) and Temperature anomalies (Equations 13.4 and 13.5)

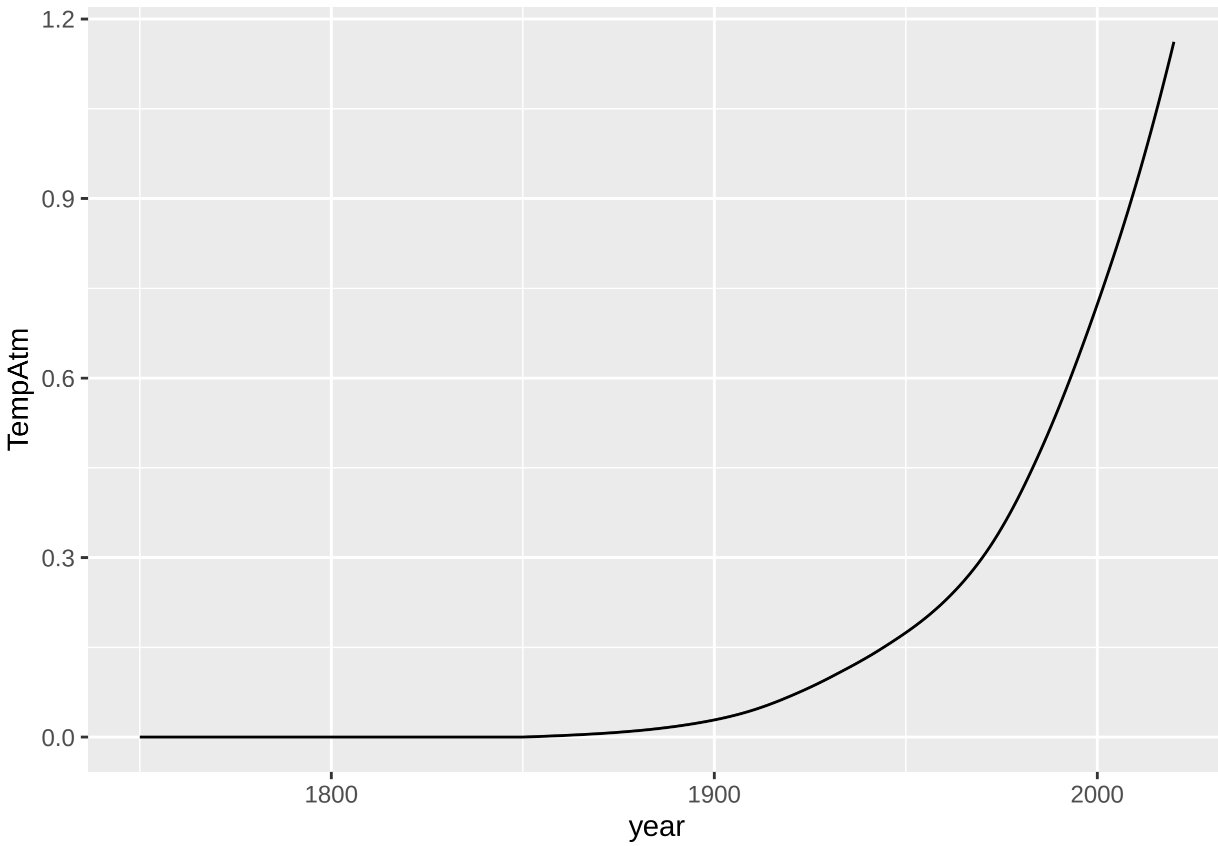
```
RadForc <-function(CO2) {
  5.35*log(CO2/275);
}

Temps <- function(atmtempold, oceantempold, radforc) {
  par1 <- 1.15
  par2 <- 0.0256
  par3 <- 0.00738
  par4 <- 0.00568

  atmtempnew   = atmtempold   +
      par2*(par1*radforc-atmtempold) +
      par3*(oceantempold-atmtempold)

  oceantempnew = oceantempold +
    par4*(atmtempold-oceantempold)

  temps <- c(atmtempnew, oceantempnew)
  names(temps) <- c("atm", "ocean")
  temps
}
```

```
HistoricData$RF <- RadForc(HistoricData$CO2conc)
```

```r
HistoricData$TempAtm = 0
HistoricData$TempOcean = 0

StartYear <- 1850
EndYear <- 2020

for (i in (which(HistoricData$year == StartYear)+1):which(HistoricData$year == EndYear)) {
  temp <- Temps(HistoricData$TempAtm[i-1], HistoricData$TempOcean[i-1], HistoricData$RF[i])
  HistoricData$TempAtm[i] <- temp["atm"]
  HistoricData$TempOcean[i] <- temp["ocean"]
}
```

We can now plot temp over time

```r
HistoricData %>%
  ggplot(aes(year, TempAtm)) +
  geom_line()
```

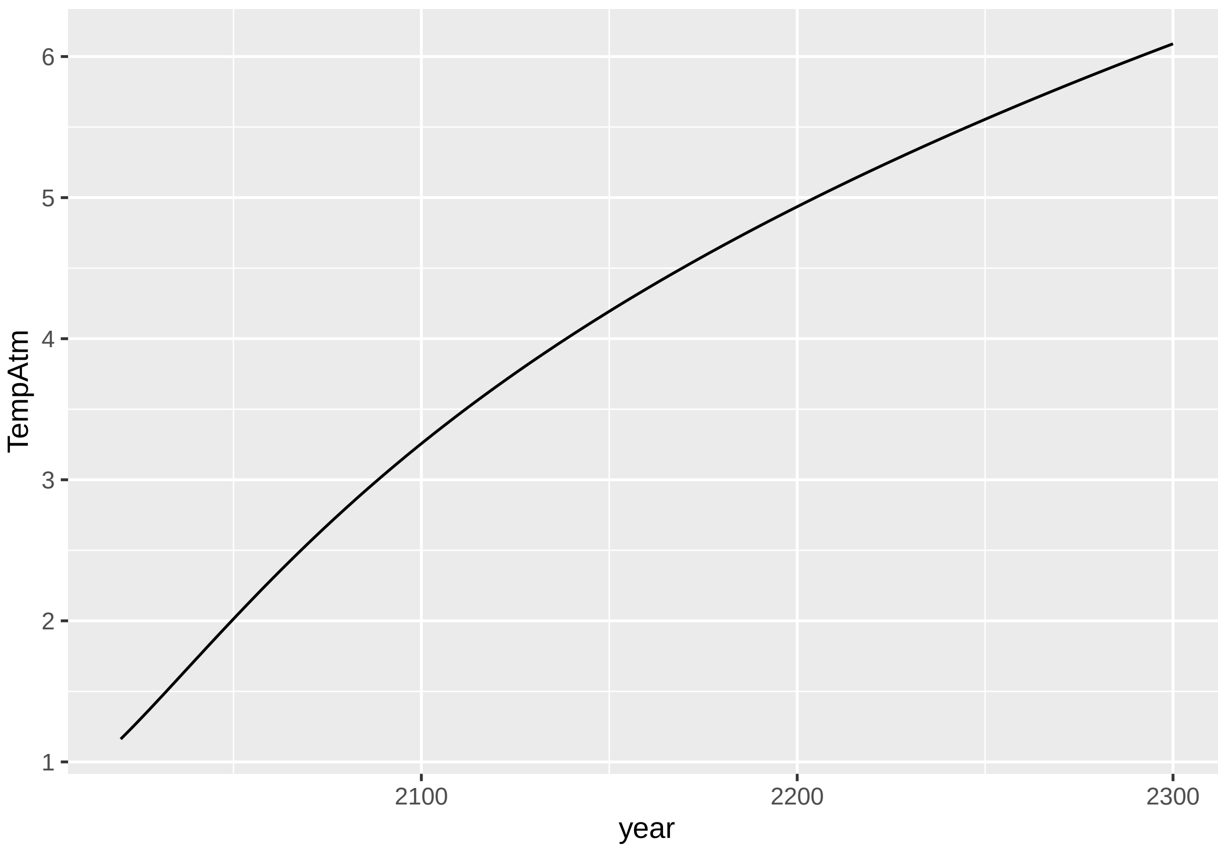


Future Scenario Example: Constant Emissions

```r
years <- 2020:2300

Scenario1 <- HistoricData[which(HistoricData$year == 2020), ]
Scenario1[2:length(years), ] <- 0

Scenario1$year <- years
```

```r
Scenario1$EmissionsMMTC = Scenario1$EmissionsMMTC[1]

#Implement Five box model for CO2 concentrations
for (i in 2:length(Scenario1$EmissionsMMTC)) {
 Scenario1[i, Box1col:(Box1col+4)] <- CO2difference(Scenario1[i - 1, Box1col:(Box1col+4)],
                                                      Scenario1$EmissionsMMTC[i - 1])
 }
# Calculate total CO2 concentration in atmosphere
Scenario1$CO2conc <- Scenario1$Box1 + Scenario1$Box2 + Scenario1$Box3 +
  Scenario1$Box4 + Scenario1$Box5
# Calculate radiative forcing
Scenario1$RF <- RadForc(Scenario1$CO2conc)
# Calculate Temperatures
for (i in 2:length(Scenario1$EmissionsMMTC)) {
    temp <- Temps(Scenario1$TempAtm[i-1], Scenario1$TempOcean[i-1], Scenario1$RF[i])
  Scenario1$TempAtm[i] <- temp["atm"]
  Scenario1$TempOcean[i] <- temp["ocean"]
}
```

```r
Scenario1 %>% ggplot(aes(year, TempAtm)) +
  geom_line()
```



Economic Module (Lab 2)

Let's use the global data as an example.

```r
# global kaya data
global.df <- read_sheet("https://docs.google.com/spreadsheets/d/1SBco_xnwTeglMsO8X_h9JDGQ3SgUVd1tTLBt0Z
```

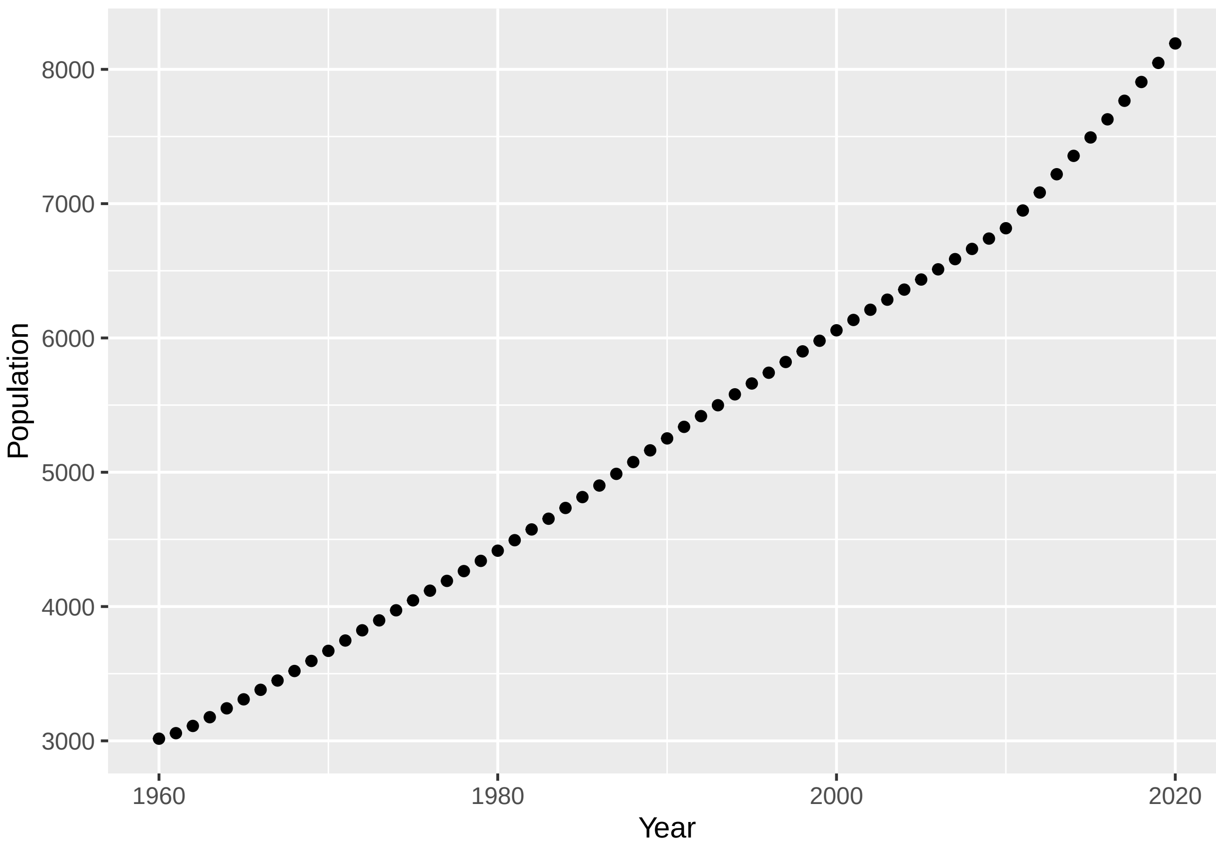
```
## v Reading from "Kaya Identity Data".
## v Range 'Sheet1'.
```

```r
head(global.df)
```

```
## # A tibble: 6 x 5
##    Year Emissions Population   GDP Energy
##   <dbl>     <dbl>      <dbl> <dbl>  <dbl>
## 1  1960      2569       3016  8651   2975
## 2  1961      2580       3057  9070   2976
## 3  1962      2686       3111  9576   3088
## 4  1963      2833       3176 10046   3274
## 5  1964      2995       3242 10716   3440
## 6  1965      3130       3309 11362   3710
```

Population

```r
global.df %>%
  ggplot(aes(Year, Population)) +
  geom_point()
```




```r
global.df$Population[which(global.df$Year == 2020)]/global.df$Population[which(global.df$Year == 2019)]
```

```
## [1] 0.0180169
```

```r
round(global.df$Population[2:61]/global.df$Population[1:60]-1 , 3)
```

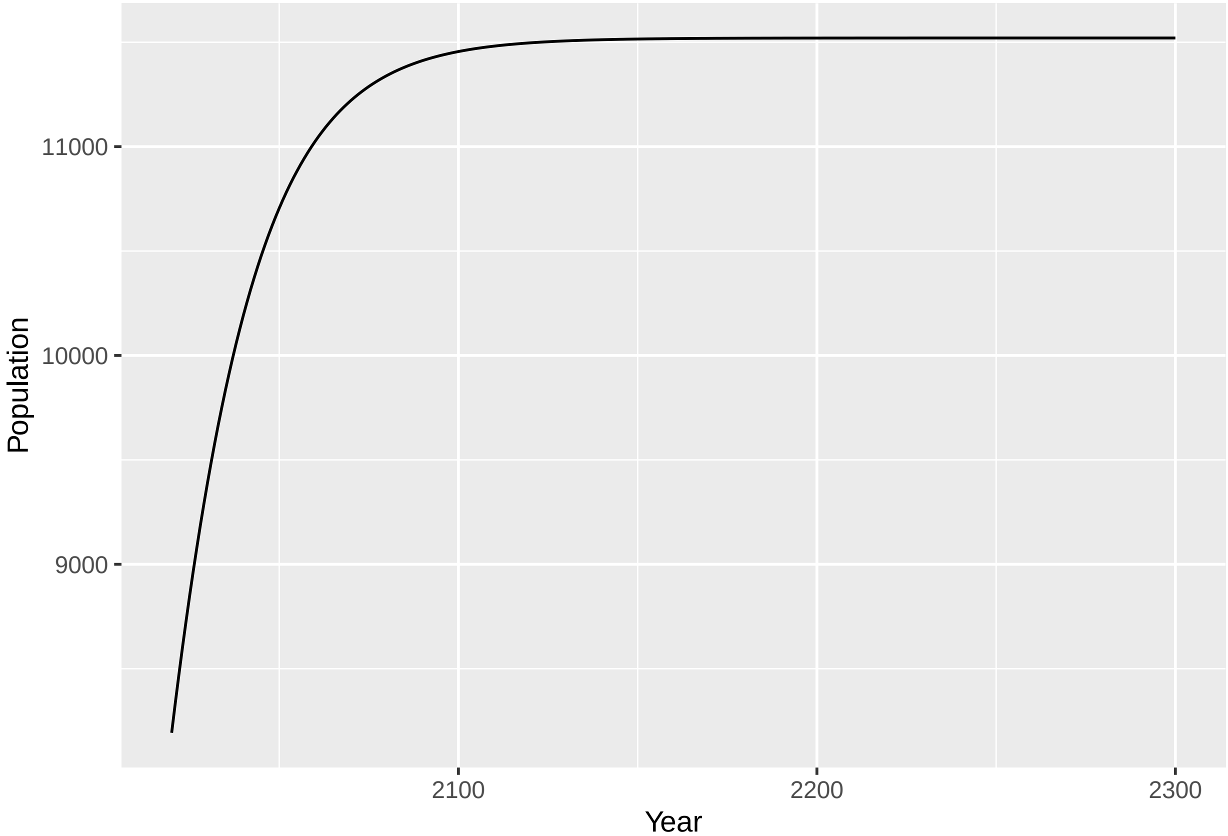
```
##  [1] 0.014 0.018 0.021 0.021 0.021 0.021 0.020 0.021 0.021 0.021 0.021 0.020
## [13] 0.019 0.019 0.019 0.018 0.018 0.017 0.018 0.018 0.018 0.018 0.017 0.017
## [25] 0.017 0.018 0.018 0.018 0.017 0.017 0.016 0.015 0.015 0.015 0.015 0.014
## [37] 0.014 0.014 0.013 0.013 0.013 0.012 0.012 0.012 0.012 0.012 0.012 0.012
## [49] 0.012 0.011 0.019 0.019 0.019 0.019 0.019 0.018 0.018 0.018 0.018 0.018
```

```r
PopulationModel = data.frame(Year = 2020:2300,
                             Population = global.df$Population[which(global.df$Year == 2020)],
                             GrowthRate =  global.df$Population[which(global.df$Year == 2020)]/global.d

PopulationModel[1, ]
```

```
##   Year Population GrowthRate
## 1 2020       8193  0.0180169
```

```r
for (i in 2:length(PopulationModel$Year)) {
  PopulationModel$GrowthRate[i] <- PopulationModel$GrowthRate[i-1]*.95
  PopulationModel$Population[i] <- PopulationModel$Population[i-1]*(1 + PopulationModel$GrowthRate[i])
}
```

```r
PopulationModel %>% ggplot(aes(Year, Population)) +
  geom_line()
```

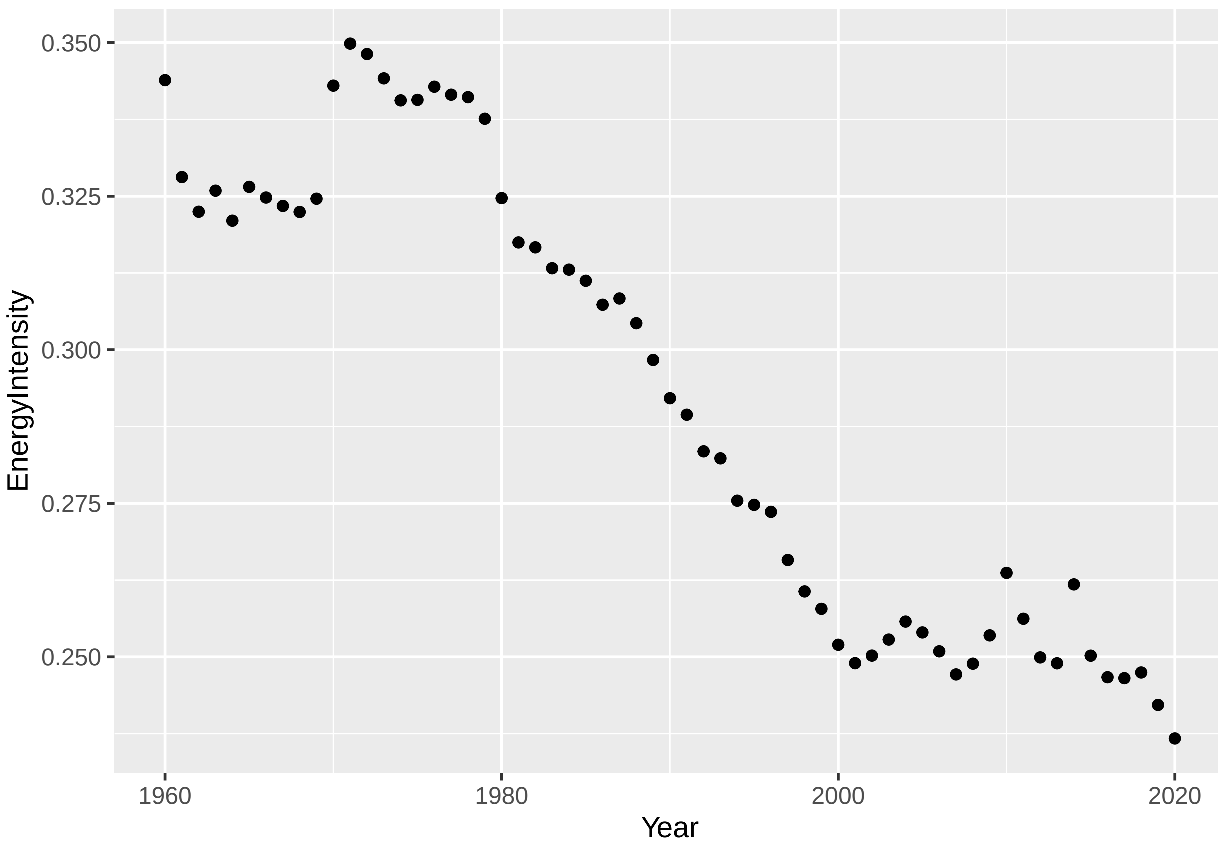


Energy Intensity

We need to calculate the energy intensity of GDP before we can model it in the future.

```r
global.df$EnergyIntensity <- global.df$Energy/global.df$GDP
```

```r
global.df %>%
  ggplot(aes(Year, EnergyIntensity)) +
  geom_point()
```




```r
round(global.df$EnergyIntensity[2:61]/global.df$EnergyIntensity[1:60]-1 , 3)
```

```
##  [1] -0.046 -0.017  0.011 -0.015  0.017 -0.005 -0.004 -0.003  0.007  0.057
## [11]  0.020 -0.005 -0.011 -0.010  0.000  0.006 -0.004 -0.001 -0.010 -0.038
## [21] -0.022 -0.003 -0.011 -0.001 -0.006 -0.013  0.003 -0.013 -0.020 -0.021
## [31] -0.009 -0.021 -0.004 -0.024 -0.002 -0.004 -0.029 -0.019 -0.011 -0.023
## [41] -0.012  0.005  0.010  0.012 -0.007 -0.012 -0.015  0.007  0.019  0.040
## [51] -0.028 -0.025 -0.004  0.052 -0.044 -0.014 -0.001  0.004 -0.021 -0.023
```

```r
aveEnergyIntGrowth <- mean(global.df$EnergyIntensity[2:61]/global.df$EnergyIntensity[1:60]-1)
aveEnergyIntGrowth
```

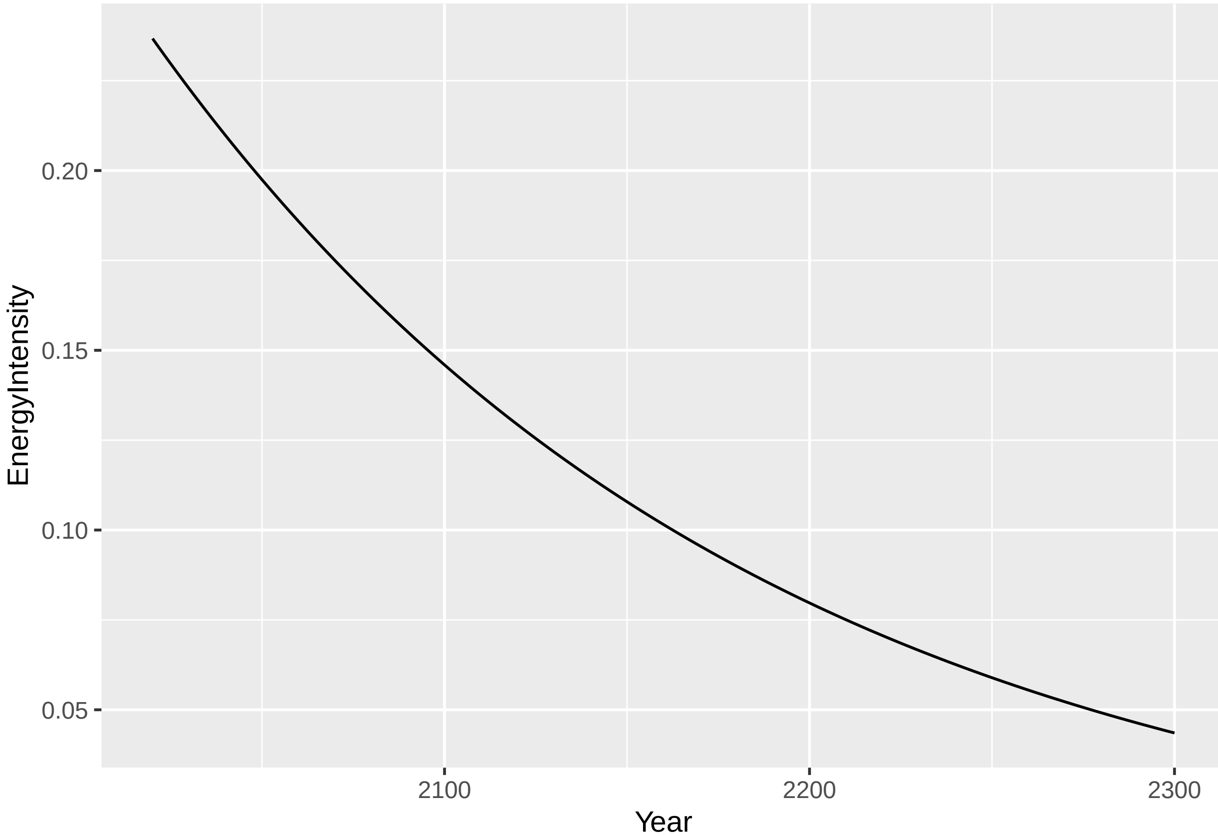
```
## [1] -0.006028318
```

```r
EnergyIntensityModel = data.frame(Year = 2020:2300,
                                  EnergyIntensity = global.df$EnergyIntensity[which(global.df$Year == 2020)]
                                  GrowthRate =  aveEnergyIntGrowth)

EnergyIntensityModel[1, ]
```

```
##   Year EnergyIntensity   GrowthRate
## 1 2020       0.2367197 -0.006028318
```

```r
for (i in 2:length(EnergyIntensityModel$Year)) {
  EnergyIntensityModel$EnergyIntensity[i] <- EnergyIntensityModel$EnergyIntensity[i-1]*(1 + EnergyIntens
}
```

```r
EnergyIntensityModel %>% ggplot(aes(Year, EnergyIntensity)) +
  geom_line()
```

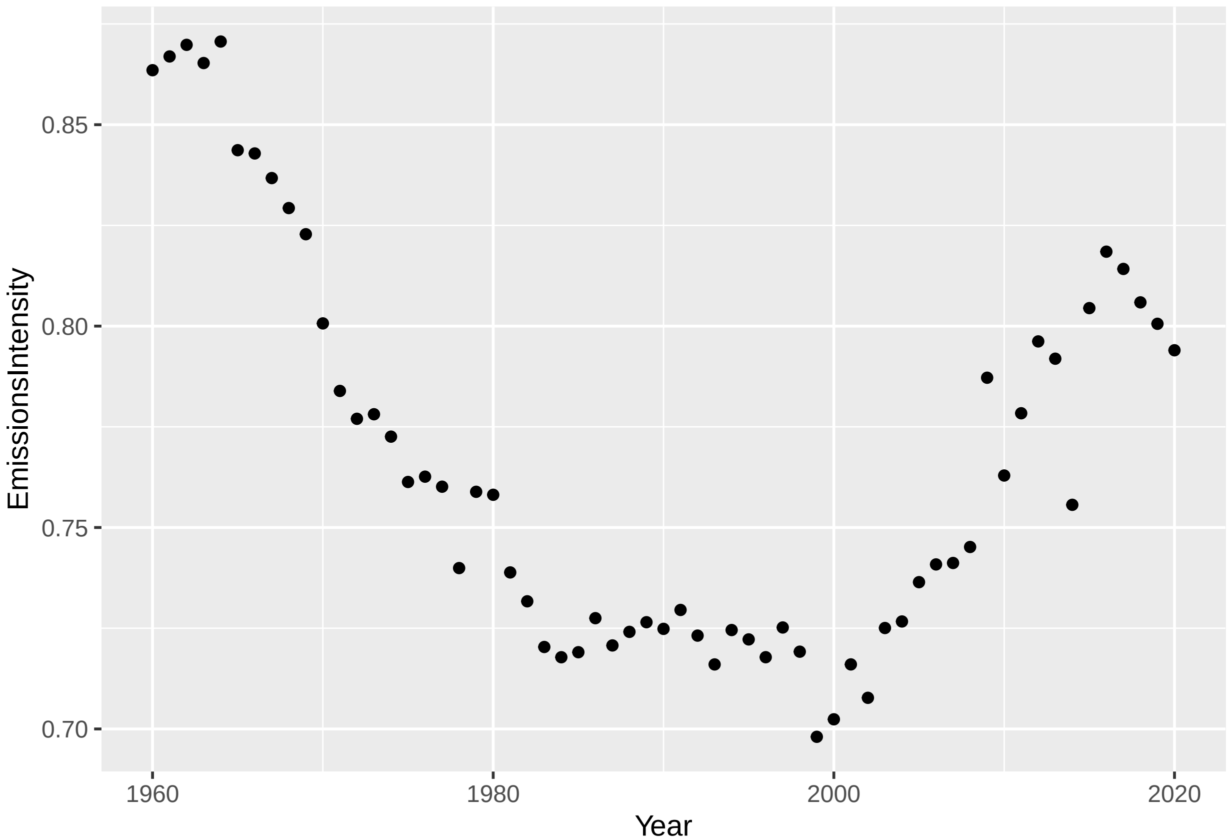


Emissions Intensity

We need to calculate the emissions intensity of ebnergy before we can model it in the future.

```r
global.df$EmissionsIntensity <- global.df$Emissions/global.df$Energy
```

```r
global.df %>%
  ggplot(aes(Year, EmissionsIntensity)) +
  geom_point()
```




```
round(global.df$EmissionsIntensity[2:61]/global.df$EmissionsIntensity[1:60]-1 , 3)
```

```
##  [1]  0.004  0.003 -0.005  0.006 -0.031 -0.001 -0.007 -0.009 -0.008 -0.027
## [11] -0.021 -0.009  0.001 -0.007 -0.015  0.002 -0.003 -0.027  0.026 -0.001
## [21] -0.025 -0.010 -0.015 -0.004  0.002  0.012 -0.009  0.005  0.003 -0.002
## [31]  0.006 -0.009 -0.010  0.012 -0.003 -0.006  0.010 -0.008 -0.029  0.006
## [41]  0.019 -0.012  0.024  0.002  0.013  0.006  0.000  0.005  0.056 -0.031
## [51]  0.020  0.023 -0.005 -0.046  0.065  0.017 -0.005 -0.010 -0.007 -0.008
```

```
aveEmissionsIntGrowth <- mean(global.df$EmissionsIntensity[2:61]/global.df$EmissionsIntensity[1:60]-1)
aveEmissionsIntGrowth
```

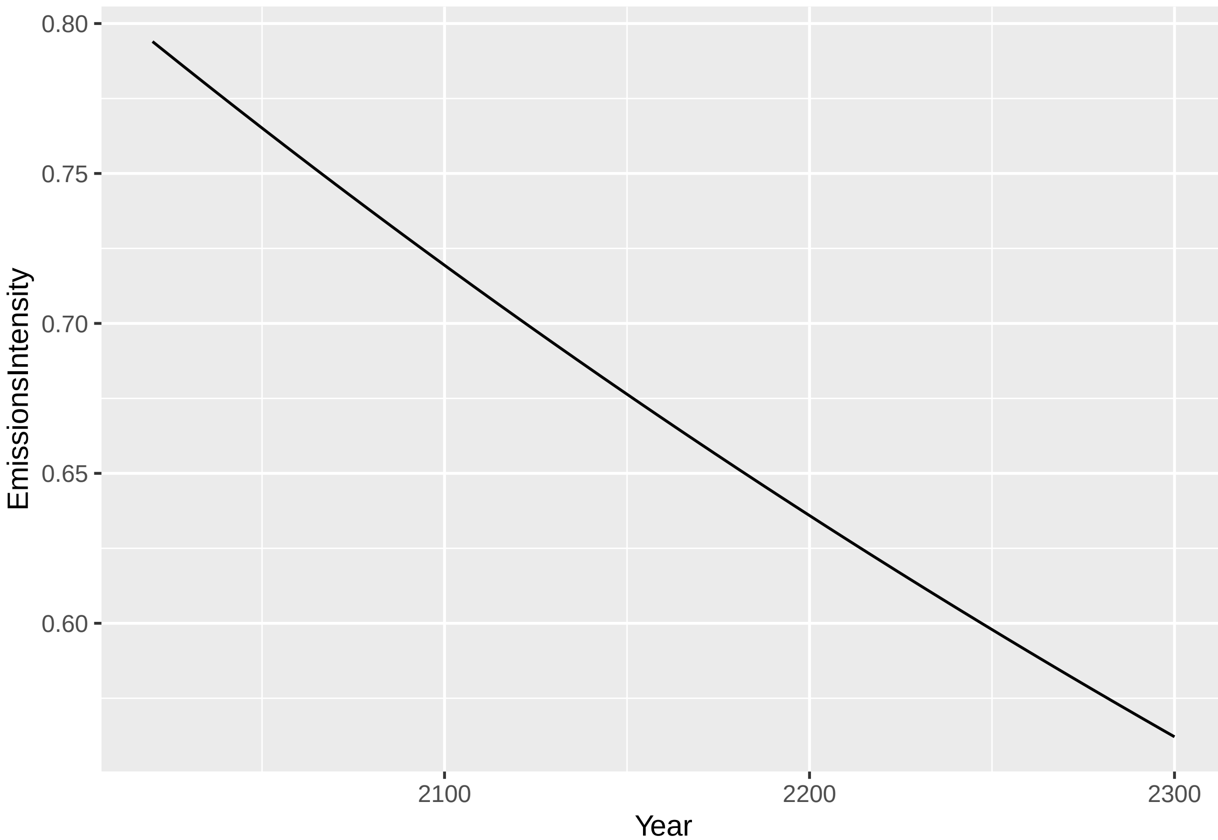
```
## [1] -0.001232515
```

```
EmissionsIntensityModel = data.frame(Year = 2020:2300,
                             EmissionsIntensity = global.df$EmissionsIntensity[which(global.df$Year == 
                             GrowthRate =  aveEmissionsIntGrowth)

EmissionsIntensityModel[1, ]
```

```
##   Year EmissionsIntensity   GrowthRate
## 1 2020          0.7939992 -0.001232515
```

```
for (i in 2:length(EmissionsIntensityModel$Year)) {
  EmissionsIntensityModel$EmissionsIntensity[i] <- EmissionsIntensityModel$EmissionsIntensity[i-1]*(1 +
}
```

```
EmissionsIntensityModel %>% ggplot(aes(Year, EmissionsIntensity)) +
  geom_line()
```



GDP

For GDP we will use the Cobb-Douglass Production function and the Solow Growth model.

$$GDP = A * K^{\alpha} * L^{(1-\alpha)}$$

where K is capital, L is labor, and A is the Total Factor Productivity.

```
CobbDouglassalpha <- .2
CobbDouglassGDP <- function(A, K, L, alpha = CobbDouglassalpha) {
  A*K^alpha*L^(1-alpha)
}
```

We are also told that capital also follows a difference equation, changing over time depending on depreciation, investment and savings.

$$K_t = K_{t-1} - \delta K_{t-1} + Investmentt$$

and because Investment = Savings, and we assume Savings is a constant share of output, $Savings = s * GDP$, then

$$K_t = K_{t-1} - \delta K_{t-1} + s * GDP_{t-1}$$

```
depreciation <- .1
savingsrate <- .2
Kapital <- function(Kold, GDPold, d = depreciation, s = savingsrate) {
```

```
  (1-d)*Kold + s*GDPold
}
```

Add some new columns to our data to use for calculations Text says that we should calibrate the model with changing A levels and assume a starting value of capital with

$$K_{1960} = \frac{s * A}{\delta}^{\frac{1}{1-\alpha}} * Population$$

```
global.df$A = 1 # text says use 1 as a starter value
global.df$K = 0
global.df$GDPmodeled = 0

global.df$K[1] = (savingsrate*global.df$A[1]/depreciation)^(1/(1-CobbDouglassalpha))*global.df$Populati
global.df$GDPmodeled[1] = CobbDouglassGDP(global.df$A[1], global.df$K[1], global.df$Population[1])

global.df %>% select(Year, Population, GDP, A, K, GDPmodeled)
```

```
## # A tibble: 61 x 6
##     Year Population   GDP     A     K GDPmodeled
##    <dbl>      <dbl> <dbl> <dbl> <dbl>      <dbl>
##  1  1960       3016  8651     1 7173.      3587.
##  2  1961       3057  9070     1    0          0
##  3  1962       3111  9576     1    0          0
##  4  1963       3176 10046     1    0          0
##  5  1964       3242 10716     1    0          0
##  6  1965       3309 11362     1    0          0
##  7  1966       3380 12011     1    0          0
##  8  1967       3449 12538     1    0          0
##  9  1968       3520 13336     1    0          0
## 10  1969       3595 14153     1    0          0
## # i 51 more rows
```

We see that the modeled GDP value is well below the actual GDP observed. We need to adjust our A value until the model is calibrated to yield the observed value. Note that when we adjust A, we also need to adjust our steady state capital value too.

```
global.df$A = 2.03 # I tried various values until the modeled value was close to the actual value.
global.df$K = 0
global.df$GDPmodeled = 0

global.df$K[1] = (savingsrate*global.df$A[1]/depreciation)^(1/(1-CobbDouglassalpha))*global.df$Populati
global.df$GDPmodeled[1] = CobbDouglassGDP(global.df$A[1], global.df$K[1], global.df$Population[1])

global.df %>% select(Year, Emissions, Population, GDP, A, K, GDPmodeled) %>% head()
```

```
## # A tibble: 6 x 7
##    Year Emissions Population   GDP     A      K GDPmodeled
##   <dbl>     <dbl>      <dbl> <dbl> <dbl>  <dbl>      <dbl>
## 1  1960      2569       3016  8651  2.03 17382.      8691.
## 2  1961      2580       3057  9070  2.03     0          0
## 3  1962      2686       3111  9576  2.03     0          0
## 4  1963      2833       3176 10046  2.03     0          0
## 5  1964      2995       3242 10716  2.03     0          0
## 6  1965      3130       3309 11362  2.03     0          0
```

Now that we have a starting value for A and Capital, we can implement our capital difference equation and use the new Population variable to estimate GDP over time.

```
for (i in 2:length(global.df$Year)) {
  global.df$K[i] = Kapital(global.df$K[i-1], global.df$GDPmodeled[i-1])
  global.df$GDPmodeled[i] = CobbDouglassGDP(global.df$A[i], global.df$K[i], global.df$Population[i])
}
global.df %>% select(Year, Emissions, Population, GDP, A, K, GDPmodeled) %>% tail()
```

```
## # A tibble: 6 x 7
##    Year Emissions Population   GDP     A      K GDPmodeled
##   <dbl>     <dbl>      <dbl> <dbl> <dbl>  <dbl>      <dbl>
## 1  2015     10775       7493 53534  2.03 36126.     20835.
## 2  2016     11110       7628 55025  2.03 36680.     21199.
## 3  2017     11257       7766 56081  2.03 37252.     21572.
## 4  2018     11467       7906 57500  2.03 37841.     21951.
## 5  2019     11477       8048 59196  2.03 38447.     22337.
## 6  2020     11432       8193 60823  2.03 39070.     22731.
```

While the modeled GDP is larger than in 1960, it hasn't grown as much as the observed GDP. We will now let A, Total Factor Productivity, also grow over time.

```
TFPgrowth <- 0.02 # this 2% value is suggested by the text

for (i in 2:length(global.df$Year)) {
  global.df$A[i] = global.df$A[i-1]*(1+TFPgrowth)
  global.df$K[i] = Kapital(global.df$K[i-1], global.df$GDPmodeled[i-1])
  global.df$GDPmodeled[i] = CobbDouglassGDP(global.df$A[i], global.df$K[i], global.df$Population[i])
}
global.df %>% select(Year, Population, GDP, A, K, GDPmodeled) %>% tail()
```

```
## # A tibble: 6 x 6
##    Year Population   GDP     A       K GDPmodeled
##   <dbl>      <dbl> <dbl> <dbl>   <dbl>      <dbl>
## 1  2015       7493 53534  6.03 109747.     77323.
## 2  2016       7628 55025  6.15 114237.     80648.
## 3  2017       7766 56081  6.28 118943.     84126.
## 4  2018       7906 57500  6.40 123874.     87754.
## 5  2019       8048 59196  6.53 129037.     91538.
## 6  2020       8193 60823  6.66 134441.     95492.
```

Now modeled GDP is higher than observed, so it seems like TFP growth should be less than 0.02

```
TFPgrowth <- 0.0137 # adjusted this value until the 2020 GDP and modeled GDP are similar in 2020

for (i in 2:length(global.df$Year)) {
  global.df$A[i] = global.df$A[i-1]*(1+TFPgrowth)
  global.df$K[i] = Kapital(global.df$K[i-1], global.df$GDPmodeled[i-1])
  global.df$GDPmodeled[i] = CobbDouglassGDP(global.df$A[i], global.df$K[i], global.df$Population[i])
}
global.df %>% select(Year, Population, GDP, A, K, GDPmodeled) %>% tail()
```

```
## # A tibble: 6 x 6
##    Year Population   GDP     A      K GDPmodeled
##   <dbl>      <dbl> <dbl> <dbl>  <dbl>      <dbl>
## 1  2015       7493 53534  4.29 77175.     51255.
## 2  2016       7628 55025  4.35 79708.     53046.
```

```
## 3  2017       7766 56081  4.41 82347.     54906.
## 4  2018       7906 57500  4.47 85093.     56831.
## 5  2019       8048 59196  4.53 87950.     58824.
## 6  2020       8193 60823  4.59 90920.     60890.
```

The text says to model future GDP assuming total factor productivity growth rate is 0.95 times the previous time period growth rate for Total Factor Productivity.

```r
GDPModel = PopulationModel
GDPModel$GrowthRate = TFPgrowth
GDPModel$A = global.df$A[which(global.df$Year == 2020)]
GDPModel$K = global.df$K[which(global.df$Year == 2020)]
GDPModel$GDPmodeled = global.df$GDPmodeled[which(global.df$Year == 2020)]

head(GDPModel)
```

```
##   Year Population GrowthRate        A     K GDPmodeled
## 1 2020   8193.000     0.0137 4.592644 90920   60890.44
## 2 2021   8333.232     0.0137 4.592644 90920   60890.44
## 3 2022   8468.732     0.0137 4.592644 90920   60890.44
## 4 2023   8599.551     0.0137 4.592644 90920   60890.44
## 5 2024   8725.748     0.0137 4.592644 90920   60890.44
## 6 2025   8847.395     0.0137 4.592644 90920   60890.44
```

```r
for (i in 2:length(GDPModel$Year)) {
  GDPModel$GrowthRate[i] = GDPModel$GrowthRate[i-1]*.99
  GDPModel$A[i] = GDPModel$A[i-1]*(1 + GDPModel$GrowthRate[i])
  GDPModel$K[i] = Kapital(GDPModel$K[i-1], GDPModel$GDPmodeled[i-1])
  GDPModel$GDPmodeled[i] = CobbDouglassGDP(GDPModel$A[i], GDPModel$K[i], GDPModel$Population[i])
}

GDPModel$GDPperCapita = GDPModel$GDPmodeled/GDPModel$Population

tail(GDPModel)
```

```
##     Year Population   GrowthRate        A        K GDPmodeled GDPperCapita
## 276 2295   11520.29 0.0008637719 16.29188 883352.8   447063.1     38.80657
## 277 2296   11520.29 0.0008551342 16.30581 884430.1   447554.5     38.84923
## 278 2297   11520.29 0.0008465829 16.31961 885498.0   448041.5     38.89150
## 279 2298   11520.29 0.0008381170 16.33329 886556.5   448524.2     38.93340
## 280 2299   11520.29 0.0008297359 16.34684 887605.7   449002.6     38.97492
## 281 2300   11520.29 0.0008214385 16.36027 888645.7   449476.7     39.01607
```

Kaya Identity

Now that we have modeled the four elements of the Kaya Identity (Population, GDP per capita, Energy Intensity, and Emissions Intensity), we can use this to model future emissions.

```r
Kaya.global <- PopulationModel[, c("Year", "Population")] %>%
  left_join(GDPModel[, c("Year", "GDPperCapita")]) %>%
  left_join(EnergyIntensityModel[, c("Year", "EnergyIntensity")]) %>%
  left_join(EmissionsIntensityModel[, c("Year", "EmissionsIntensity")])
```

```
## Joining with `by = join_by(Year)`
## Joining with `by = join_by(Year)`
## Joining with `by = join_by(Year)`
```

```
head(Kaya.global)
```

```
##   Year Population GDPperCapita EnergyIntensity EmissionsIntensity
## 1 2020   8193.000     7.432007       0.2367197          0.7939992
## 2 2021   8333.232     7.557568       0.2352926          0.7930206
## 3 2022   8468.732     7.685585       0.2338742          0.7920431
## 4 2023   8599.551     7.815884       0.2324644          0.7910669
## 5 2024   8725.748     7.948309       0.2310630          0.7900919
## 6 2025   8847.395     8.082713       0.2296701          0.7891181
```

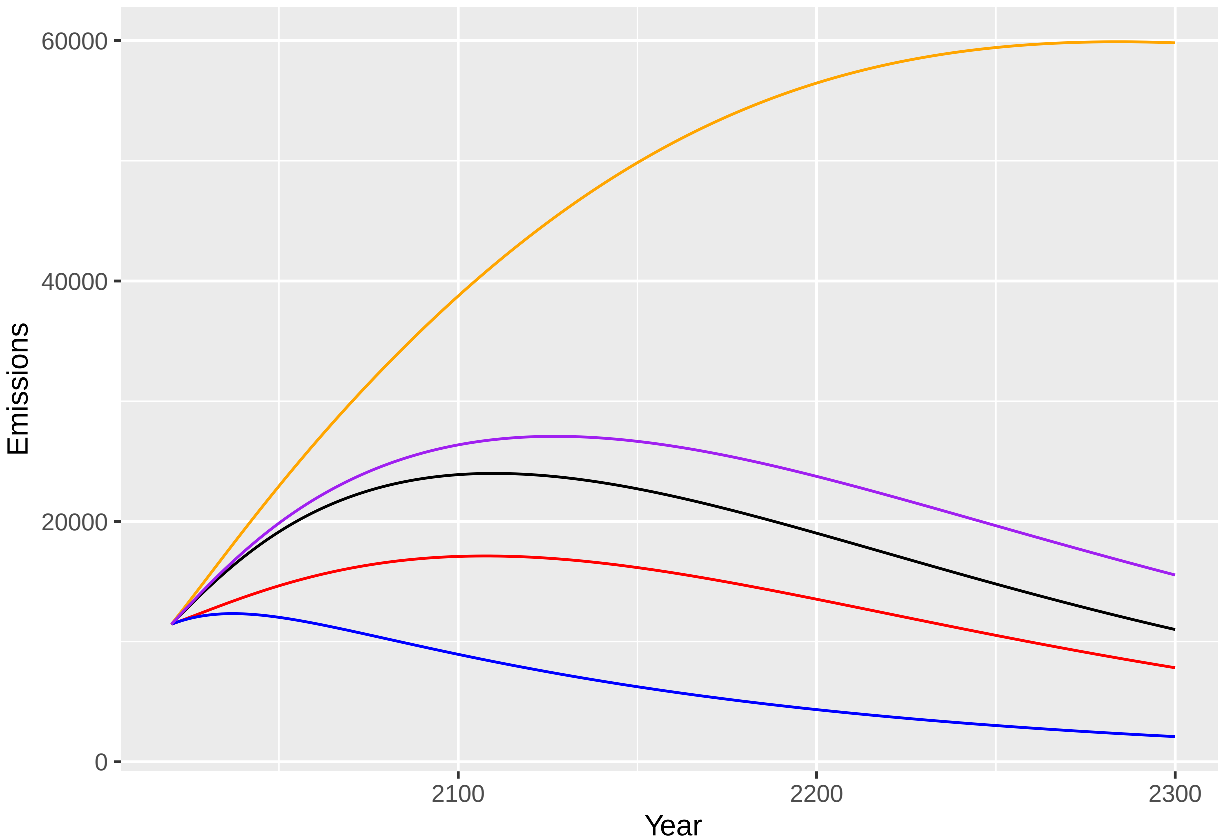
Now calculate the Kaya Identity with all of the modeled variables and again four separate times with each of the four variables held constant at their 2020 levels.

```
Kaya.global = Kaya.global %>%
  mutate(Emissions = Population*GDPperCapita*EnergyIntensity*EmissionsIntensity,
         EmissionsPopConstant = Population[1]*GDPperCapita*EnergyIntensity*EmissionsIntensity,
         EmissionsGDPperCapConstant = Population*GDPperCapita[1]*EnergyIntensity*EmissionsIntensity,
         EmissionsEnergyIntensityConstant = Population*GDPperCapita*EnergyIntensity[1]*EmissionsIntensi
         EmissionsEmissionsIntensityConstant = Population*GDPperCapita*EnergyIntensity*EmissionsIntensi
head(Kaya.global)
```

```
##   Year Population GDPperCapita EnergyIntensity EmissionsIntensity Emissions
## 1 2020   8193.000     7.432007       0.2367197          0.7939992  11444.67
## 2 2021   8333.232     7.557568       0.2352926          0.7930206  11751.37
## 3 2022   8468.732     7.685585       0.2338742          0.7920431  12056.65
## 4 2023   8599.551     7.815884       0.2324644          0.7910669  12360.14
## 5 2024   8725.748     7.948309       0.2310630          0.7900919  12661.51
## 6 2025   8847.395     8.082713       0.2296701          0.7891181  12960.42
##   EmissionsPopConstant EmissionsGDPperCapConstant
## 1             11444.67                   11444.67
## 2             11553.61                   11556.13
## 3             11664.10                   11658.85
## 4             11775.81                   11753.07
## 5             11888.46                   11839.05
## 6             12001.80                   11917.03
##   EmissionsEnergyIntensityConstant EmissionsEmissionsIntensityConstant
## 1                         11444.67                            11444.67
## 2                         11822.64                            11765.87
## 3                         12203.33                            12086.42
## 4                         12586.40                            12405.96
## 5                         12971.48                            12724.12
## 6                         13358.23                            13040.58
```

We can now feed these emissions predictions into our climate model to predict how CO2 concentrations and temperatures might be under the various scenarios. However, before we do that, let's plot the emissions variables in the difference scenarios.

```
Kaya.global %>%
  ggplot(aes(x = Year)) +
  geom_line(aes(y = Emissions)) +
  geom_line(aes(y = EmissionsPopConstant), color = "red") +
  geom_line(aes(y = EmissionsGDPperCapConstant), color = "blue") +
  geom_line(aes(y = EmissionsEnergyIntensityConstant), color = "orange") +
  geom_line(aes(y = EmissionsEmissionsIntensityConstant), color = "purple")
```



Calculate CO2 concentrations and temperature for emissions paths

```
Kaya.global <- left_join(Kaya.global, HistoricData, by = c("Year" = "year") )

Box1col <- which(colnames(Kaya.global) == "Box1")

#Implement Five box model for CO2 concentrations
for (i in 2:length(Kaya.global$Year)) {
 Kaya.global[i, Box1col:(Box1col+4)] <- CO2difference(Kaya.global[i - 1, Box1col:(Box1col+4)],
                                                       Kaya.global$Emissions[i - 1])
 }
# Calculate total CO2 concentration in atmosphere
Kaya.global$CO2conc <- Kaya.global$Box1 + Kaya.global$Box2 + Kaya.global$Box3 +
  Kaya.global$Box4 + Kaya.global$Box5
# Calculate radiative forcing
Kaya.global$RF <- RadForc(Kaya.global$CO2conc)
# Calculate Temperatures
for (i in 2:length(Kaya.global$Year)) {
    temp <- Temps(Kaya.global$TempAtm[i-1], Kaya.global$TempOcean[i-1], Kaya.global$RF[i])
  Kaya.global$TempAtm[i] <- temp["atm"]
  Kaya.global$TempOcean[i] <- temp["ocean"]
}
```

Let's extract the emissions, CO2 concentrations, and atmospheric temperature for the different scenarios and put them in a separate dataframe for convenience.

```
Exercise13.5 <- Kaya.global %>%
  select(Year, Emissions, CO2conc, TempAtm) %>%
```

```
  mutate(ConstantVariable = "None")
```

Now we rerun the Climate model for the other four scenarios and grab the same variables. Start with Population.

```
#Implement Five box model for CO2 concentrations
for (i in 2:length(Kaya.global$Year)) {
 Kaya.global[i, Box1col:(Box1col+4)] <- CO2difference(Kaya.global[i - 1, Box1col:(Box1col+4)],
                                                      Kaya.global$EmissionsPopConstant[i - 1]) # CHANG
 }
# Calculate total CO2 concentration in atmosphere
Kaya.global$CO2conc <- Kaya.global$Box1 + Kaya.global$Box2 + Kaya.global$Box3 +
  Kaya.global$Box4 + Kaya.global$Box5
# Calculate radiative forcing
Kaya.global$RF <- RadForc(Kaya.global$CO2conc)
# Calculate Temperatures
for (i in 2:length(Kaya.global$Year)) {
    temp <- Temps(Kaya.global$TempAtm[i-1], Kaya.global$TempOcean[i-1], Kaya.global$RF[i])
  Kaya.global$TempAtm[i] <- temp["atm"]
  Kaya.global$TempOcean[i] <- temp["ocean"]
}

Exercise13.5 <- Kaya.global %>%
  select(Year, EmissionsPopConstant, CO2conc, TempAtm) %>%
  rename(Emissions = EmissionsPopConstant) %>%
  mutate(ConstantVariable = "Population") %>%
  full_join(Exercise13.5)
```

```
## Joining with `by = join_by(Year, Emissions, CO2conc, TempAtm,
## ConstantVariable)`
```

Keep Per capita GDP constant

```
#Implement Five box model for CO2 concentrations
for (i in 2:length(Kaya.global$Year)) {
 Kaya.global[i, Box1col:(Box1col+4)] <- CO2difference(Kaya.global[i - 1, Box1col:(Box1col+4)],
                                                      Kaya.global$EmissionsGDPperCapConstant[i - 1]) #
 }
# Calculate total CO2 concentration in atmosphere
Kaya.global$CO2conc <- Kaya.global$Box1 + Kaya.global$Box2 + Kaya.global$Box3 +
  Kaya.global$Box4 + Kaya.global$Box5
# Calculate radiative forcing
Kaya.global$RF <- RadForc(Kaya.global$CO2conc)
# Calculate Temperatures
for (i in 2:length(Kaya.global$Year)) {
    temp <- Temps(Kaya.global$TempAtm[i-1], Kaya.global$TempOcean[i-1], Kaya.global$RF[i])
  Kaya.global$TempAtm[i] <- temp["atm"]
  Kaya.global$TempOcean[i] <- temp["ocean"]
}

Exercise13.5 <- Kaya.global %>%
  select(Year, EmissionsGDPperCapConstant, CO2conc, TempAtm) %>%
  rename(Emissions = EmissionsGDPperCapConstant) %>%
  mutate(ConstantVariable = "GDPperCapita") %>%
  full_join(Exercise13.5)
```

```
## Joining with `by = join_by(Year, Emissions, CO2conc, TempAtm,
## ConstantVariable)`
```

Keep Energy Intensity constant

```
#Implement Five box model for CO2 concentrations
for (i in 2:length(Kaya.global$Year)) {
 Kaya.global[i, Box1col:(Box1col+4)] <- CO2difference(Kaya.global[i - 1, Box1col:(Box1col+4)],
                                                      Kaya.global$EmissionsEnergyIntensityConstant[i -
 }
# Calculate total CO2 concentration in atmosphere
Kaya.global$CO2conc <- Kaya.global$Box1 + Kaya.global$Box2 + Kaya.global$Box3 +
  Kaya.global$Box4 + Kaya.global$Box5
# Calculate radiative forcing
Kaya.global$RF <- RadForc(Kaya.global$CO2conc)
# Calculate Temperatures
for (i in 2:length(Kaya.global$Year)) {
    temp <- Temps(Kaya.global$TempAtm[i-1], Kaya.global$TempOcean[i-1], Kaya.global$RF[i])
  Kaya.global$TempAtm[i] <- temp["atm"]
  Kaya.global$TempOcean[i] <- temp["ocean"]
}

Exercise13.5 <- Kaya.global %>%
  select(Year, EmissionsEnergyIntensityConstant, CO2conc, TempAtm) %>%
  rename(Emissions = EmissionsEnergyIntensityConstant) %>%
  mutate(ConstantVariable = "EnergyIntensity") %>%
  full_join(Exercise13.5)
```

```
## Joining with `by = join_by(Year, Emissions, CO2conc, TempAtm,
## ConstantVariable)`
```

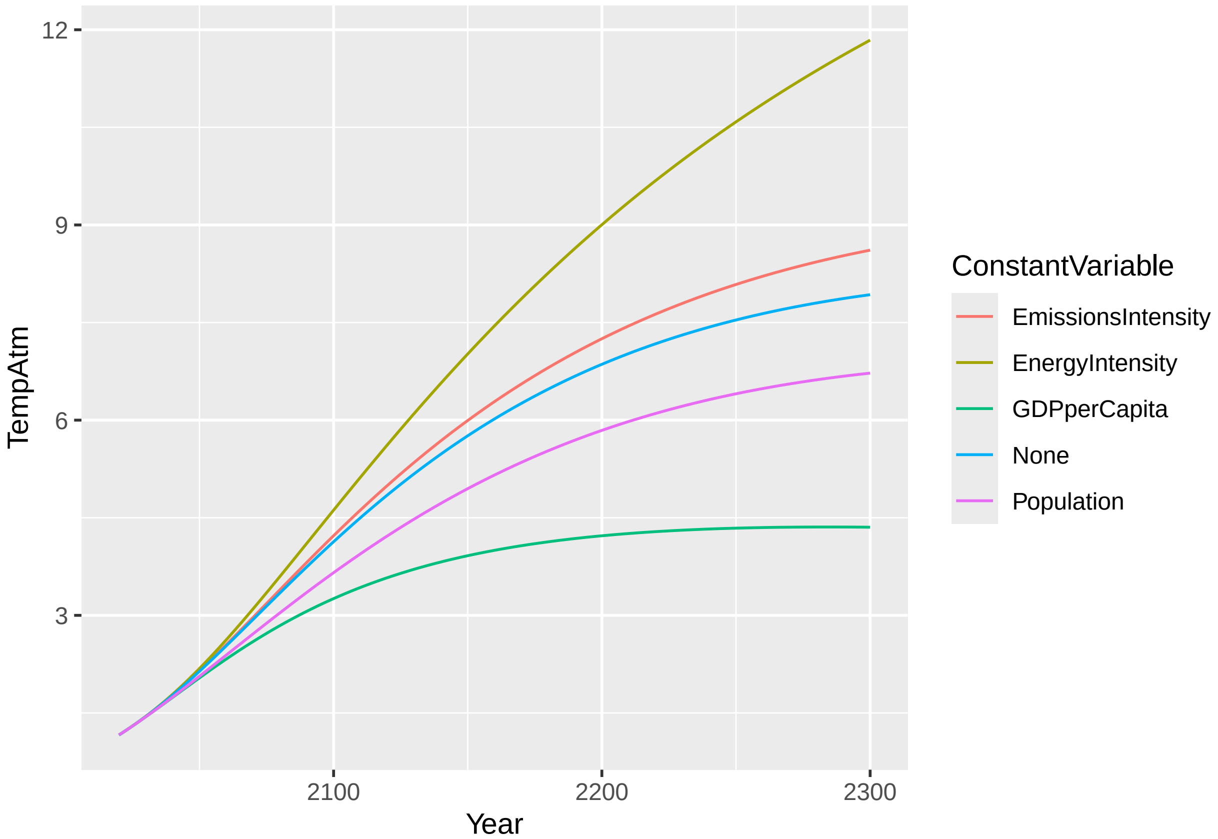
Keep Emissions Intensity constant

```
#Implement Five box model for CO2 concentrations
for (i in 2:length(Kaya.global$Year)) {
 Kaya.global[i, Box1col:(Box1col+4)] <- CO2difference(Kaya.global[i - 1, Box1col:(Box1col+4)],
                                                      Kaya.global$EmissionsEmissionsIntensityConstant[
 }
# Calculate total CO2 concentration in atmosphere
Kaya.global$CO2conc <- Kaya.global$Box1 + Kaya.global$Box2 + Kaya.global$Box3 +
  Kaya.global$Box4 + Kaya.global$Box5
# Calculate radiative forcing
Kaya.global$RF <- RadForc(Kaya.global$CO2conc)
# Calculate Temperatures
for (i in 2:length(Kaya.global$Year)) {
    temp <- Temps(Kaya.global$TempAtm[i-1], Kaya.global$TempOcean[i-1], Kaya.global$RF[i])
  Kaya.global$TempAtm[i] <- temp["atm"]
  Kaya.global$TempOcean[i] <- temp["ocean"]
}

Exercise13.5 <- Kaya.global %>%
  select(Year, EmissionsEmissionsIntensityConstant, CO2conc, TempAtm) %>%
  rename(Emissions = EmissionsEmissionsIntensityConstant) %>%
  mutate(ConstantVariable = "EmissionsIntensity") %>%
  full_join(Exercise13.5)
```

```
## Joining with `by = join_by(Year, Emissions, CO2conc, TempAtm,
```

```
## ConstantVariable)`
```

```r
Exercise13.5 %>% ggplot(aes(x = Year, y = TempAtm, color = ConstantVariable)) +
  geom_line()
```




```r
Exercise13.5 %>% filter(Year == 2300) %>%
  select(ConstantVariable, TempAtm) %>%
  arrange(-TempAtm) %>%
  mutate(across(where(is.numeric), \(x) round(x, 2)))
```

```
##     ConstantVariable TempAtm
## 1    EnergyIntensity   11.84
## 2 EmissionsIntensity    8.61
## 3               None    7.93
## 4         Population    6.72
## 5       GDPperCapita    4.35
```

Using Righ-Middle-Poor Region

Access the regional data

```r
# regional kaya data
regional.df <- read_sheet("https://docs.google.com/spreadsheets/d/1fgR_jtz7zCLynudKfqpjCr2LbORW_tXqHrdGI
```

```
## v Reading from "Lab 02 Regional Data".
```

```
## v Range 'Sheet1'.
```

```
## New names:
## * `` -> `...3`
```

```
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...7`
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...11`
## * `` -> `...12`
## * `` -> `...13`
## * `` -> `...15`
## * `` -> `...16`
```

```
# note the regional data has some issues with the names, etc. because the data starts on row 4, with re
```

```
Rich.df <- regional.df[-(1:3), c(1, 2, 6, 10, 14)]
head(Rich.df)
```

```
## # A tibble: 6 x 5
##    Year Population GDP       Energy    Emissions
##   <dbl> <list>     <list>    <list>    <list>
## 1  1960 <dbl [1]>  <dbl [1]> <dbl [1]> <dbl [1]>
## 2  1961 <dbl [1]>  <dbl [1]> <dbl [1]> <dbl [1]>
## 3  1962 <dbl [1]>  <dbl [1]> <dbl [1]> <dbl [1]>
## 4  1963 <dbl [1]>  <dbl [1]> <dbl [1]> <dbl [1]>
## 5  1964 <dbl [1]>  <dbl [1]> <dbl [1]> <dbl [1]>
## 6  1965 <dbl [1]>  <dbl [1]> <dbl [1]> <dbl [1]>
```

```
Rich.df <- Rich.df %>% mutate(across(Population:Emissions, as.numeric))
head(Rich.df)
```

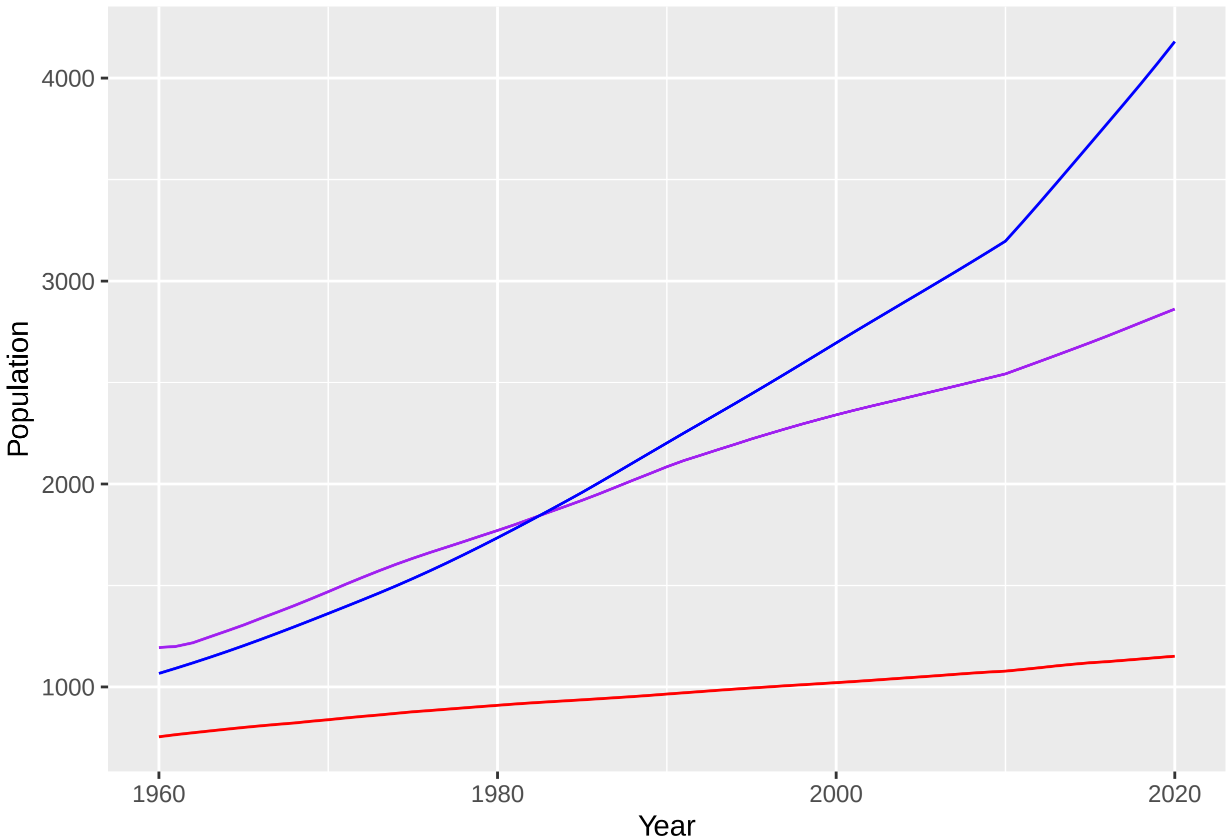
```
## # A tibble: 6 x 5
##    Year Population   GDP Energy Emissions
##   <dbl>      <dbl> <dbl>  <dbl>     <dbl>
## 1  1960       755. 7556.  1823.     1635
## 2  1961       765. 7939.  1870.     1684.
## 3  1962       774. 8405.  1971.     1776.
## 4  1963       783. 8839.  2097.     1875.
## 5  1964       792. 9403.  2194.     1976.
## 6  1965       801. 9928.  2372.     2044.
```

```
Middle.df <- regional.df[-(1:3), c(1, 3, 7, 11, 15)]
names(Middle.df) <- names(Rich.df)
Middle.df <- Middle.df %>% mutate(across(Population:Emissions, as.numeric))

Poor.df <- regional.df[-(1:3), c(1, 4, 8, 12, 16)]
names(Poor.df) <- names(Rich.df)
Poor.df <- Poor.df %>% mutate(across(Population:Emissions, as.numeric))
```

Population

The graph below shows that the population of the three different regions seems to have been growing at different rates since 1960.

```
ggplot() +
  geom_line(data = Rich.df, aes(x= Year, y = Population), color = "red") +
  geom_line(data = Middle.df, aes(x= Year, y = Population), color = "purple") +
  geom_line(data = Poor.df, aes(x= Year, y = Population), color = "blue")
```



Rich Region Kaya Modeling

```
Rich.df$Population[which(Rich.df$Year == 2020)]/Rich.df$Population[which(Rich.df$Year == 2019)]-1
```

```
## [1] 0.005839959
```

```
round(Rich.df$Population[2:61]/Rich.df$Population[1:60]-1 , 3)
```

```
##  [1] 0.014 0.012 0.011 0.011 0.011 0.010 0.009 0.008 0.010 0.009 0.010 0.009
## [13] 0.008 0.009 0.009 0.007 0.007 0.007 0.007 0.007 0.007 0.006 0.006 0.005
## [25] 0.005 0.006 0.006 0.006 0.006 0.007 0.006 0.006 0.006 0.006 0.006 0.005
## [37] 0.006 0.005 0.005 0.005 0.005 0.005 0.006 0.006 0.006 0.006 0.006 0.006
## [49] 0.005 0.004 0.008 0.008 0.008 0.007 0.007 0.005 0.006 0.006 0.006 0.006
```

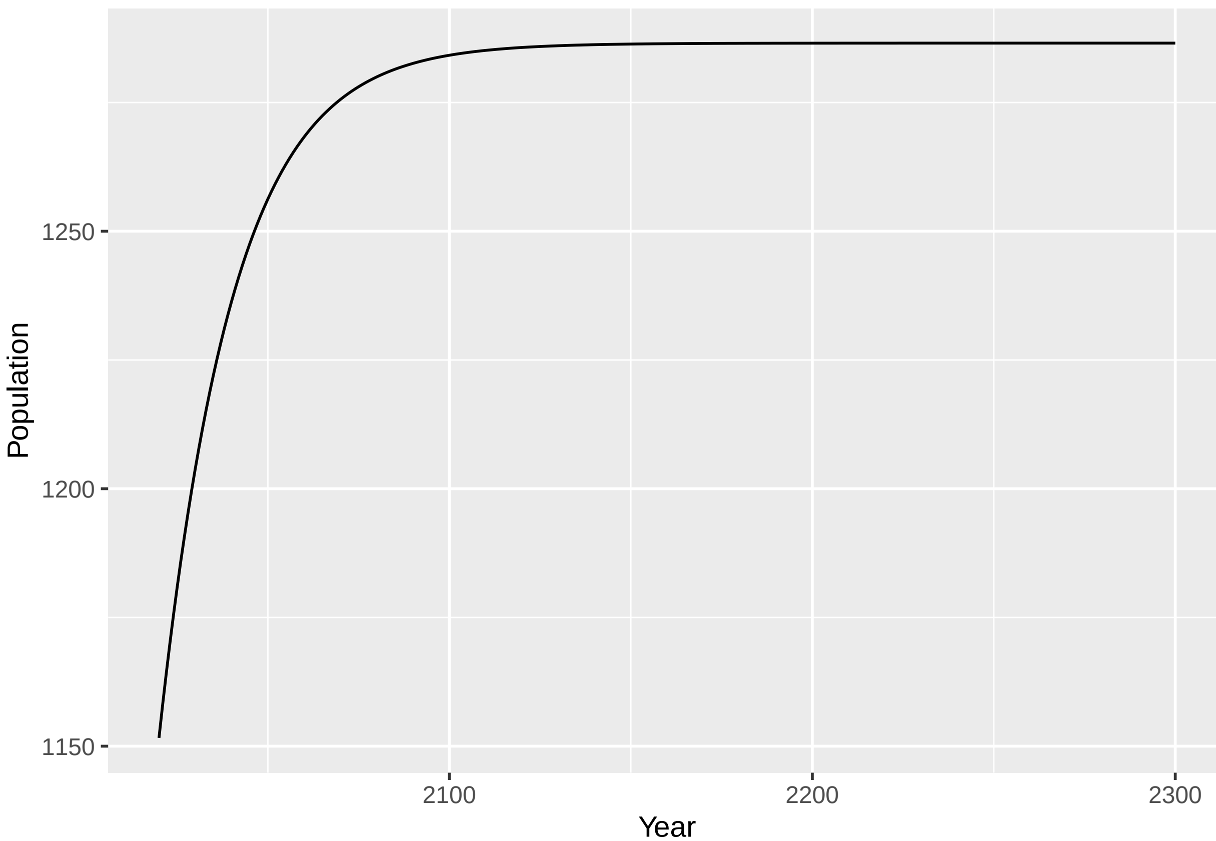
```
PopulationModel = data.frame(Year = 2020:2300,
                             Population = Rich.df$Population[which(Rich.df$Year == 2020)],
                             GrowthRate =  Rich.df$Population[which(Rich.df$Year == 2020)]/Rich.df$Popul

head(PopulationModel)
```

```
##   Year Population  GrowthRate
## 1 2020   1151.598 0.005839959
## 2 2021   1151.598 0.005839959
## 3 2022   1151.598 0.005839959
## 4 2023   1151.598 0.005839959
## 5 2024   1151.598 0.005839959
## 6 2025   1151.598 0.005839959
```

```
for (i in 2:length(PopulationModel$Year)) {
  PopulationModel$GrowthRate[i] <- PopulationModel$GrowthRate[i-1]*.95
  PopulationModel$Population[i] <- PopulationModel$Population[i-1]*(1 + PopulationModel$GrowthRate[i])
}
head(PopulationModel)
```

```
##   Year Population  GrowthRate
## 1 2020   1151.598 0.005839959
## 2 2021   1157.987 0.005547961
## 3 2022   1164.091 0.005270563
## 4 2023   1169.919 0.005007035
## 5 2024   1175.484 0.004756683
## 6 2025   1180.796 0.004518849
```

```
PopulationModel %>% ggplot(aes(Year, Population)) +
  geom_line()
```

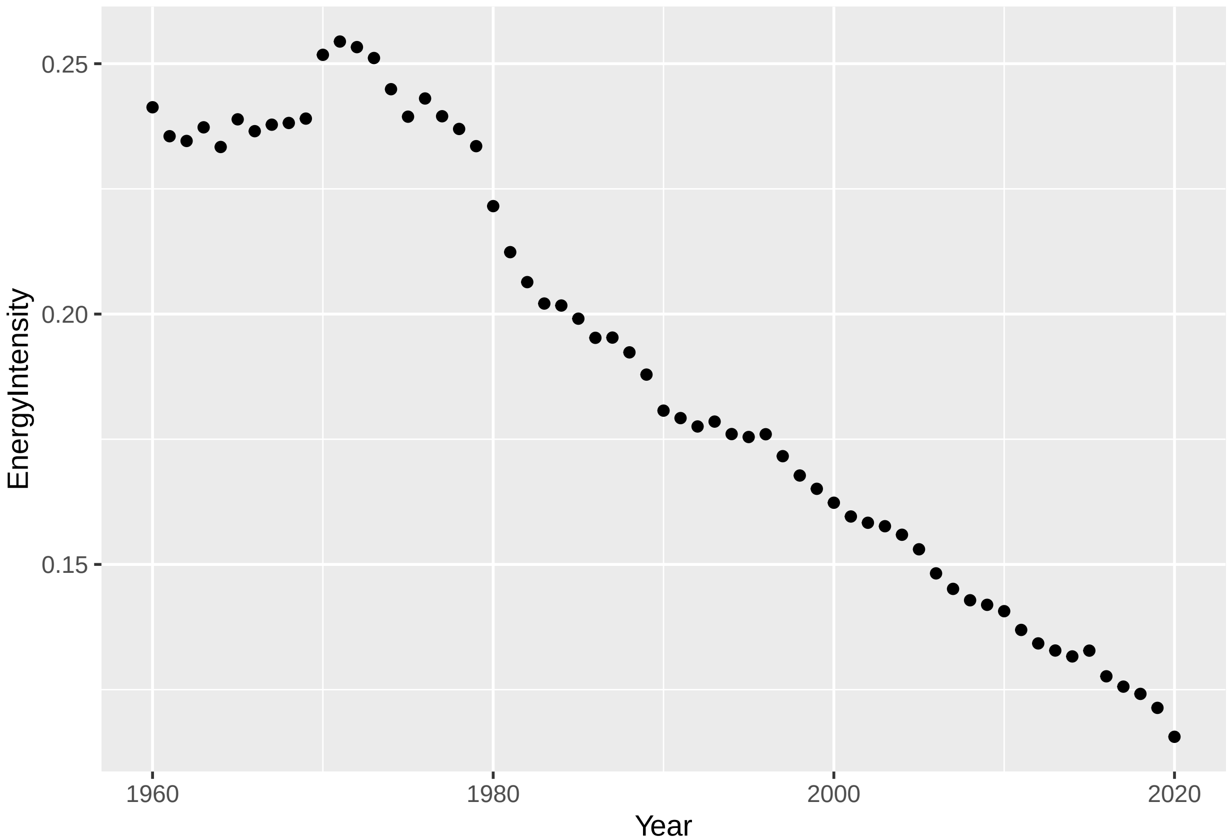


Energy Intensity

We need to calculate the energy intensity of GDP before we can model it in the future.

```
Rich.df$EnergyIntensity <- Rich.df$Energy/Rich.df$GDP
```

```
Rich.df %>%
  ggplot(aes(Year, EnergyIntensity)) +
  geom_point()
```




```
round(Rich.df$EnergyIntensity[2:61]/Rich.df$EnergyIntensity[1:60]-1 , 3)
```

```
##  [1] -0.024 -0.004  0.012 -0.017  0.024 -0.010  0.005  0.001  0.004  0.053
## [11]  0.011 -0.004 -0.009 -0.025 -0.022  0.015 -0.015 -0.011 -0.014 -0.051
## [21] -0.041 -0.028 -0.021 -0.002 -0.013 -0.019  0.000 -0.015 -0.023 -0.038
## [31] -0.008 -0.009  0.005 -0.014 -0.003  0.003 -0.025 -0.023 -0.016 -0.017
## [41] -0.017 -0.008 -0.004 -0.011 -0.019 -0.031 -0.021 -0.016 -0.006 -0.009
## [51] -0.027 -0.020 -0.011 -0.009  0.009 -0.039 -0.016 -0.012 -0.023 -0.048
```

```
aveEnergyIntGrowth <- mean(Rich.df$EnergyIntensity[2:61]/Rich.df$EnergyIntensity[1:60]-1)
aveEnergyIntGrowth
```

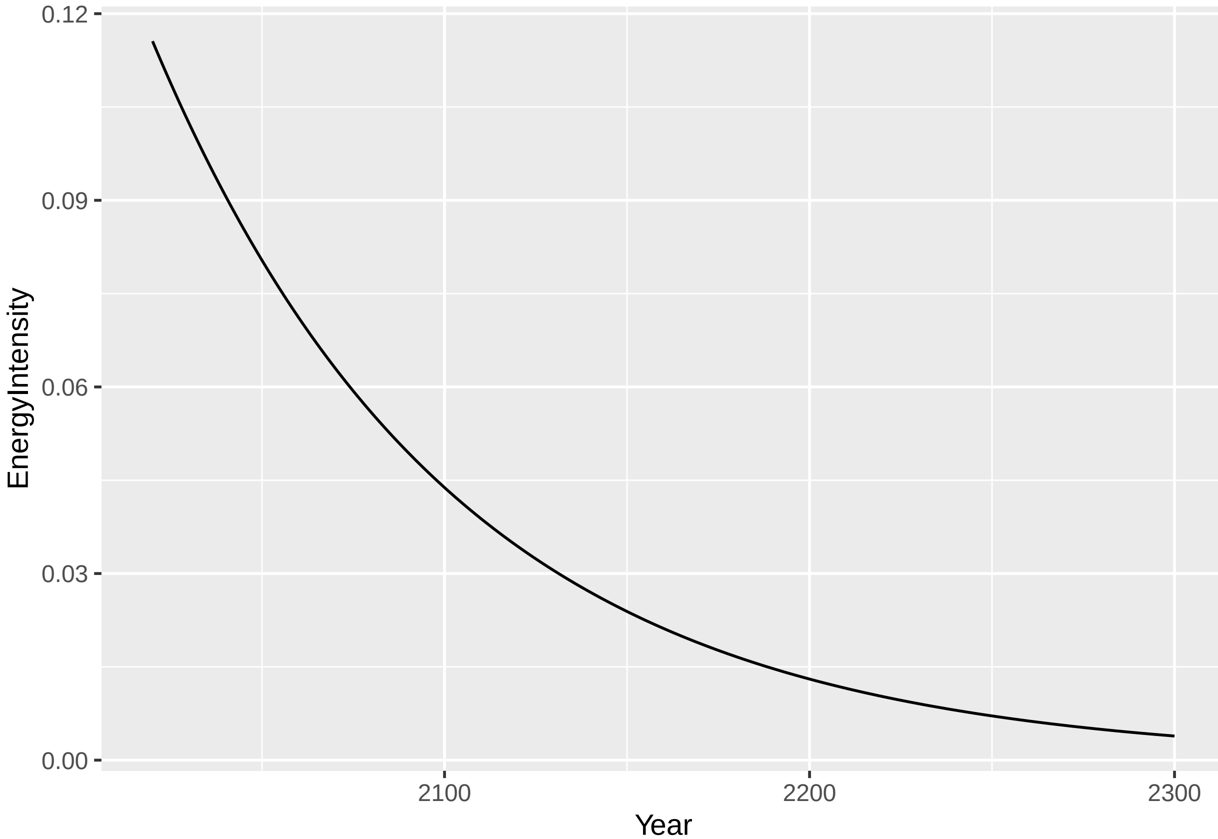
```
## [1] -0.01205049
```

```
EnergyIntensityModel = data.frame(Year = 2020:2300,
                             EnergyIntensity = Rich.df$EnergyIntensity[which(Rich.df$Year == 2020)],
                             GrowthRate =  aveEnergyIntGrowth)

EnergyIntensityModel[1, ]
```

```
##   Year EnergyIntensity  GrowthRate
## 1 2020       0.1155825 -0.01205049
```

```
for (i in 2:length(EnergyIntensityModel$Year)) {
  EnergyIntensityModel$EnergyIntensity[i] <- EnergyIntensityModel$EnergyIntensity[i-1]*(1 + EnergyIntens
}
```

```
EnergyIntensityModel %>% ggplot(aes(Year, EnergyIntensity)) +
  geom_line()
```

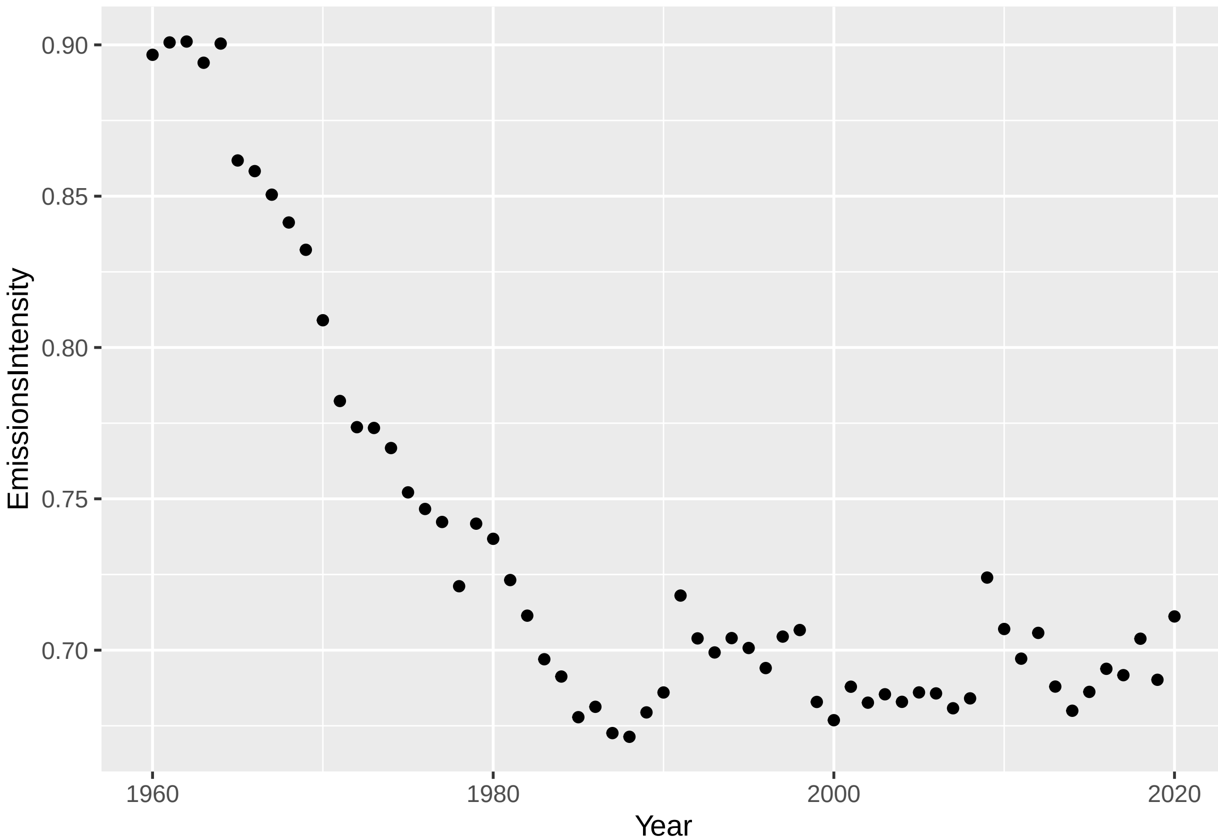


Emissions Intensity

We need to calculate the emissions intensity of ebnergy before we can model it in the future.

```
Rich.df$EmissionsIntensity <- Rich.df$Emissions/Rich.df$Energy
```

```
Rich.df %>%
  ggplot(aes(Year, EmissionsIntensity)) +
  geom_point()
```




```r
round(Rich.df$EmissionsIntensity[2:61]/Rich.df$EmissionsIntensity[1:60]-1 , 3)
```

```
##  [1]  0.005  0.000 -0.008  0.007 -0.043 -0.004 -0.009 -0.011 -0.011 -0.028
## [11] -0.033 -0.011  0.000 -0.009 -0.019 -0.007 -0.006 -0.029  0.029 -0.007
## [21] -0.019 -0.016 -0.020 -0.008 -0.019  0.005 -0.013 -0.002  0.012  0.010
## [31]  0.047 -0.020 -0.007  0.007 -0.005 -0.009  0.015  0.003 -0.034 -0.009
## [41]  0.016 -0.008  0.004 -0.004  0.005  0.000 -0.007  0.005  0.058 -0.023
## [51] -0.014  0.012 -0.025 -0.012  0.009  0.011 -0.003  0.017 -0.019  0.030
```

```r
aveEmissionsIntGrowth <- mean(Rich.df$EmissionsIntensity[2:61]/Rich.df$EmissionsIntensity[1:60]-1)
aveEmissionsIntGrowth
```

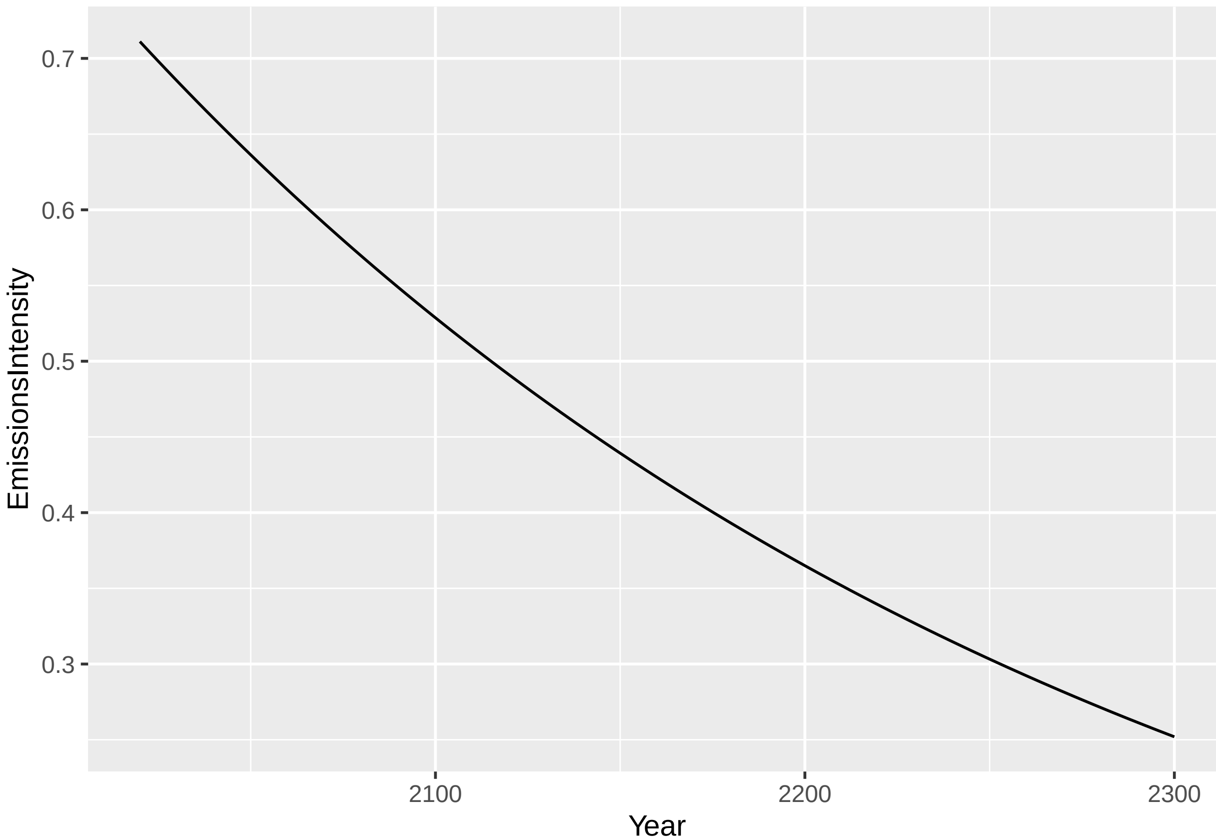
```
## [1] -0.003698849
```

```r
EmissionsIntensityModel = data.frame(Year = 2020:2300,
                             EmissionsIntensity = Rich.df$EmissionsIntensity[which(Rich.df$Year == 2020)
                             GrowthRate =  aveEmissionsIntGrowth)

EmissionsIntensityModel[1, ]
```

```
##   Year EmissionsIntensity   GrowthRate
## 1 2020          0.7111335 -0.003698849
```

```r
for (i in 2:length(EmissionsIntensityModel$Year)) {
  EmissionsIntensityModel$EmissionsIntensity[i] <- EmissionsIntensityModel$EmissionsIntensity[i-1]*(1 +
}
```

```r
EmissionsIntensityModel %>% ggplot(aes(Year, EmissionsIntensity)) +
  geom_line()
```



GDP

For GDP we will use the Cobb-Douglass Production function and the Solow Growth model.

```
Rich.df$A = 5.5 # text says use 1 as a starter value
Rich.df$K = 0
Rich.df$GDPmodeled = 0

Rich.df$K[1] = (savingsrate*Rich.df$A[1]/depreciation)^(1/(1-CobbDouglassalpha))*Rich.df$Population[1]
Rich.df$GDPmodeled[1] = CobbDouglassGDP(Rich.df$A[1], Rich.df$K[1], Rich.df$Population[1])

Rich.df %>% select(Year, Population, GDP, A, K, GDPmodeled) %>% head()
```

```
## # A tibble: 6 x 6
##    Year Population   GDP     A      K GDPmodeled
##   <dbl>      <dbl> <dbl> <dbl>  <dbl>      <dbl>
## 1  1960       755. 7556.   5.5 15117.      7559.
## 2  1961       765. 7939.   5.5     0          0
## 3  1962       774. 8405.   5.5     0          0
## 4  1963       783. 8839.   5.5     0          0
## 5  1964       792. 9403.   5.5     0          0
## 6  1965       801. 9928.   5.5     0          0
```

Now that we have a starting value for A and Capital, we can implement our capital difference equation and use the new Population variable to estimate GDP over time.

```
TFPgrowth <- 0.0183 # this 2% value is suggested by the text

for (i in 2:length(Rich.df$Year)) {
```

```r
  Rich.df$A[i] = Rich.df$A[i-1]*(1+TFPgrowth)
  Rich.df$K[i] = Kapital(Rich.df$K[i-1], Rich.df$GDPmodeled[i-1])
  Rich.df$GDPmodeled[i] = CobbDouglassGDP(Rich.df$A[i], Rich.df$K[i], Rich.df$Population[i])
}
tail(Rich.df)
```

```
## # A tibble: 6 x 10
##    Year Population    GDP Energy Emissions EnergyIntensity EmissionsIntensity
##   <dbl>      <dbl>  <dbl>  <dbl>     <dbl>           <dbl>              <dbl>
## 1  2015      1119. 38311.  5087.     3491            0.133              0.686
## 2  2016      1125. 39054.  4986.     3459.           0.128              0.694
## 3  2017      1131. 39594.  4973.     3440            0.126              0.692
## 4  2018      1138. 40190.  4990.     3512.           0.124              0.704
## 5  2019      1145. 41056.  4982.     3439.           0.121              0.690
## 6  2020      1152. 42083.  4864.     3459            0.116              0.711
## # i 3 more variables: A <dbl>, K <dbl>, GDPmodeled <dbl>
```

The text says to model future GDP assuming total factor productivity growth rate is 0.99 times the previous time period growth rate for Total Factor Productivity.

```r
GDPModel = PopulationModel
GDPModel$GrowthRate = TFPgrowth
GDPModel$A = Rich.df$A[which(Rich.df$Year == 2020)]
GDPModel$K = Rich.df$K[which(Rich.df$Year == 2020)]
GDPModel$GDPmodeled = Rich.df$GDPmodeled[which(Rich.df$Year == 2020)]

for (i in 2:length(GDPModel$Year)) {
  GDPModel$GrowthRate[i] = GDPModel$GrowthRate[i-1]*.99
  GDPModel$A[i] = GDPModel$A[i-1]*(1 + GDPModel$GrowthRate[i])
  GDPModel$K[i] = Kapital(GDPModel$K[i-1], GDPModel$GDPmodeled[i-1])
  GDPModel$GDPmodeled[i] = CobbDouglassGDP(GDPModel$A[i], GDPModel$K[i], GDPModel$Population[i])
}

GDPModel$GDPperCapita = GDPModel$GDPmodeled/GDPModel$Population

tail(GDPModel)
```

```
##     Year Population  GrowthRate        A        K GDPmodeled GDPperCapita
## 276 2295   1286.535 0.001153798 88.42443 813123.7   413181.5     321.1584
## 277 2296   1286.535 0.001142260 88.52543 814447.7   413788.1     321.6299
## 278 2297   1286.535 0.001130837 88.62554 815760.5   414389.5     322.0973
## 279 2298   1286.535 0.001119529 88.72476 817062.3   414985.7     322.5608
## 280 2299   1286.535 0.001108333 88.82310 818353.3   415576.8     323.0203
## 281 2300   1286.535 0.001097250 88.92056 819633.3   416162.9     323.4758
```

Emissions from Kaya Calculation Rich

Now that we have modeled the four elements of the Kaya Identity (Population, GDP per capita, Energy Intensity, and Emissions Intensity), we can use this to model future emissions.

```r
Kaya.Rich <- PopulationModel[, c("Year", "Population")] %>%
  left_join(GDPModel[, c("Year", "GDPperCapita")]) %>%
  left_join(EnergyIntensityModel[, c("Year", "EnergyIntensity")]) %>%
  left_join(EmissionsIntensityModel[, c("Year", "EmissionsIntensity")])
```

```
## Joining with `by = join_by(Year)`
```

```
## Joining with `by = join_by(Year)`
## Joining with `by = join_by(Year)`
```

```r
head(Kaya.Rich)
```

```
##   Year Population GDPperCapita EnergyIntensity EmissionsIntensity
## 1 2020   1151.598     36.61494       0.1155825          0.7111335
## 2 2021   1157.987     37.45132       0.1141896          0.7085031
## 3 2022   1164.091     38.30156       0.1128136          0.7058825
## 4 2023   1169.919     39.16536       0.1114541          0.7032715
## 5 2024   1175.484     40.04244       0.1101111          0.7006702
## 6 2025   1180.796     40.93253       0.1087842          0.6980785
```

Now calculate the Kaya Identity with all of the modeled variables and again four separate times with each of the four variables held constant at their 2020 levels.

```r
Kaya.Rich = Kaya.Rich %>%
  mutate(Emissions = Population*GDPperCapita*EnergyIntensity*EmissionsIntensity,
         EmissionsPopConstant = Population[1]*GDPperCapita*EnergyIntensity*EmissionsIntensity,
         EmissionsGDPperCapConstant = Population*GDPperCapita[1]*EnergyIntensity*EmissionsIntensity,
         EmissionsEnergyIntensityConstant = Population*GDPperCapita*EnergyIntensity[1]*EmissionsIntensi
         EmissionsEmissionsIntensityConstant = Population*GDPperCapita*EnergyIntensity*EmissionsIntensi
head(Kaya.Rich)
```

```
##   Year Population GDPperCapita EnergyIntensity EmissionsIntensity Emissions
## 1 2020   1151.598     36.61494       0.1155825          0.7111335  3465.791
## 2 2021   1157.987     37.45132       0.1141896          0.7085031  3508.645
## 3 2022   1164.091     38.30156       0.1128136          0.7058825  3550.562
## 4 2023   1169.919     39.16536       0.1114541          0.7032715  3591.511
## 5 2024   1175.484     40.04244       0.1101111          0.7006702  3631.465
## 6 2025   1180.796     40.93253       0.1087842          0.6980785  3670.400
##   EmissionsPopConstant EmissionsGDPperCapConstant
## 1             3465.791                   3465.791
## 2             3489.286                   3430.288
## 3             3512.459                   3394.212
## 4             3535.268                   3357.634
## 5             3557.674                   3320.624
## 6             3579.642                   3283.244
##   EmissionsEnergyIntensityConstant EmissionsEmissionsIntensityConstant
## 1                         3465.791                            3465.791
## 2                         3551.441                            3521.671
## 3                         3637.706                            3576.974
## 4                         3724.542                            3631.661
## 5                         3811.912                            3685.694
## 6                         3899.776                            3739.041
```

Middle Income

Now we repeat the same exercise with the Middle Income region data.

```r
Middle.df$Population[which(Middle.df$Year == 2020)]/Middle.df$Population[which(Middle.df$Year == 2019)]
```

```
## [1] 0.01170783
```

```r
round(Middle.df$Population[2:61]/Middle.df$Population[1:60]-1 , 3)
```

```
##  [1] 0.004 0.015 0.024 0.023 0.023 0.025 0.023 0.023 0.024 0.024 0.024 0.023
## [13] 0.022 0.020 0.019 0.017 0.016 0.016 0.016 0.016 0.016 0.017 0.017 0.016
```

```
## [25] 0.016 0.017 0.017 0.017 0.016 0.016 0.015 0.012 0.013 0.012 0.012 0.011
## [37] 0.011 0.011 0.010 0.010 0.009 0.009 0.008 0.008 0.008 0.008 0.008 0.008
## [49] 0.008 0.008 0.012 0.012 0.012 0.012 0.012 0.012 0.012 0.012 0.012 0.012
```

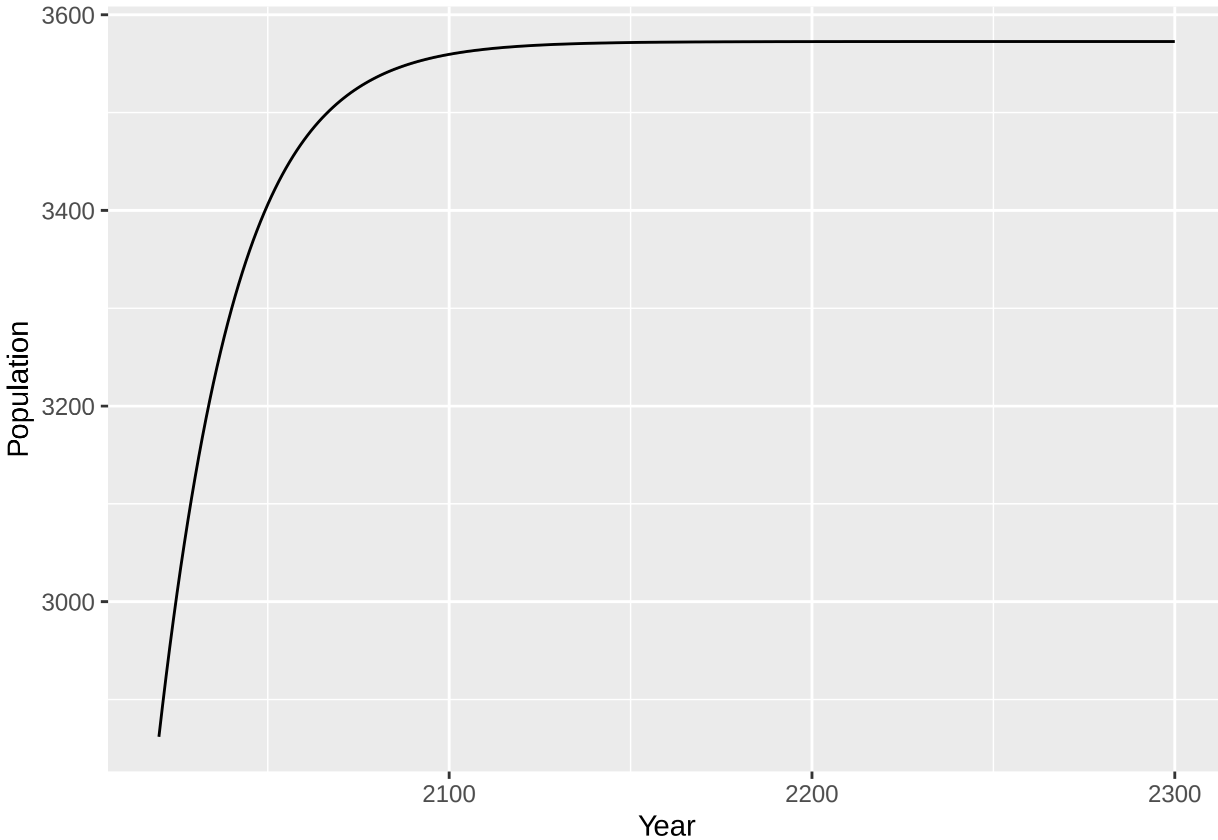
```
PopulationModel = data.frame(Year = 2020:2300,
                             Population = Middle.df$Population[which(Middle.df$Year == 2020)],
                             GrowthRate =  Middle.df$Population[which(Middle.df$Year == 2020)]/Middle.d

head(PopulationModel)
```

```
##   Year Population GrowthRate
## 1 2020   2861.881 0.01170783
## 2 2021   2861.881 0.01170783
## 3 2022   2861.881 0.01170783
## 4 2023   2861.881 0.01170783
## 5 2024   2861.881 0.01170783
## 6 2025   2861.881 0.01170783
```

```
for (i in 2:length(PopulationModel$Year)) {
  PopulationModel$GrowthRate[i] <- PopulationModel$GrowthRate[i-1]*.95
  PopulationModel$Population[i] <- PopulationModel$Population[i-1]*(1 + PopulationModel$GrowthRate[i])
}
head(PopulationModel)
```

```
##   Year Population  GrowthRate
## 1 2020   2861.881 0.011707830
## 2 2021   2893.712 0.011122438
## 3 2022   2924.288 0.010566316
## 4 2023   2953.642 0.010038000
## 5 2024   2981.808 0.009536100
## 6 2025   3008.821 0.009059295
```

```
PopulationModel %>% ggplot(aes(Year, Population)) +
  geom_line()
```

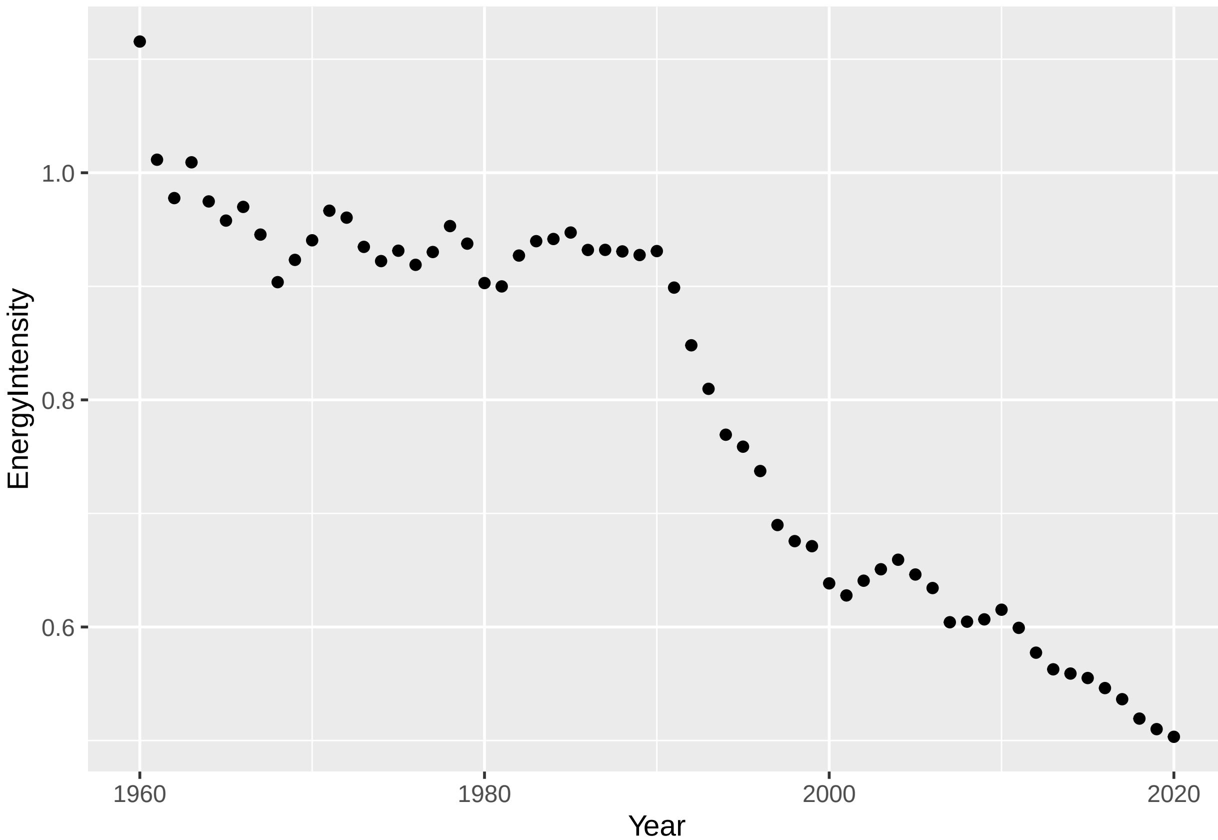


Energy Intensity

We need to calculate the energy intensity of GDP before we can model it in the future.

```
Middle.df$EnergyIntensity <- Middle.df$Energy/Middle.df$GDP
```

```
Middle.df %>%
  ggplot(aes(Year, EnergyIntensity)) +
  geom_point()
```




```
round(Middle.df$EnergyIntensity[2:61]/Middle.df$EnergyIntensity[1:60]-1 , 3)
```

```
##  [1] -0.093 -0.033  0.032 -0.034 -0.017  0.013 -0.025 -0.044  0.022  0.019
## [11]  0.028 -0.006 -0.027 -0.013  0.010 -0.013  0.012  0.025 -0.016 -0.037
## [21] -0.003  0.030  0.014  0.002  0.006 -0.016  0.000 -0.001 -0.003  0.004
## [31] -0.035 -0.057 -0.045 -0.050 -0.014 -0.028 -0.064 -0.021 -0.007 -0.049
## [41] -0.017  0.021  0.016  0.013 -0.020 -0.018 -0.047  0.001  0.003  0.014
## [51] -0.026 -0.036 -0.025 -0.007 -0.007 -0.016 -0.018 -0.032 -0.018 -0.013
```

```
aveEnergyIntGrowth <- mean(Middle.df$EnergyIntensity[2:61]/Middle.df$EnergyIntensity[1:60]-1)
aveEnergyIntGrowth
```

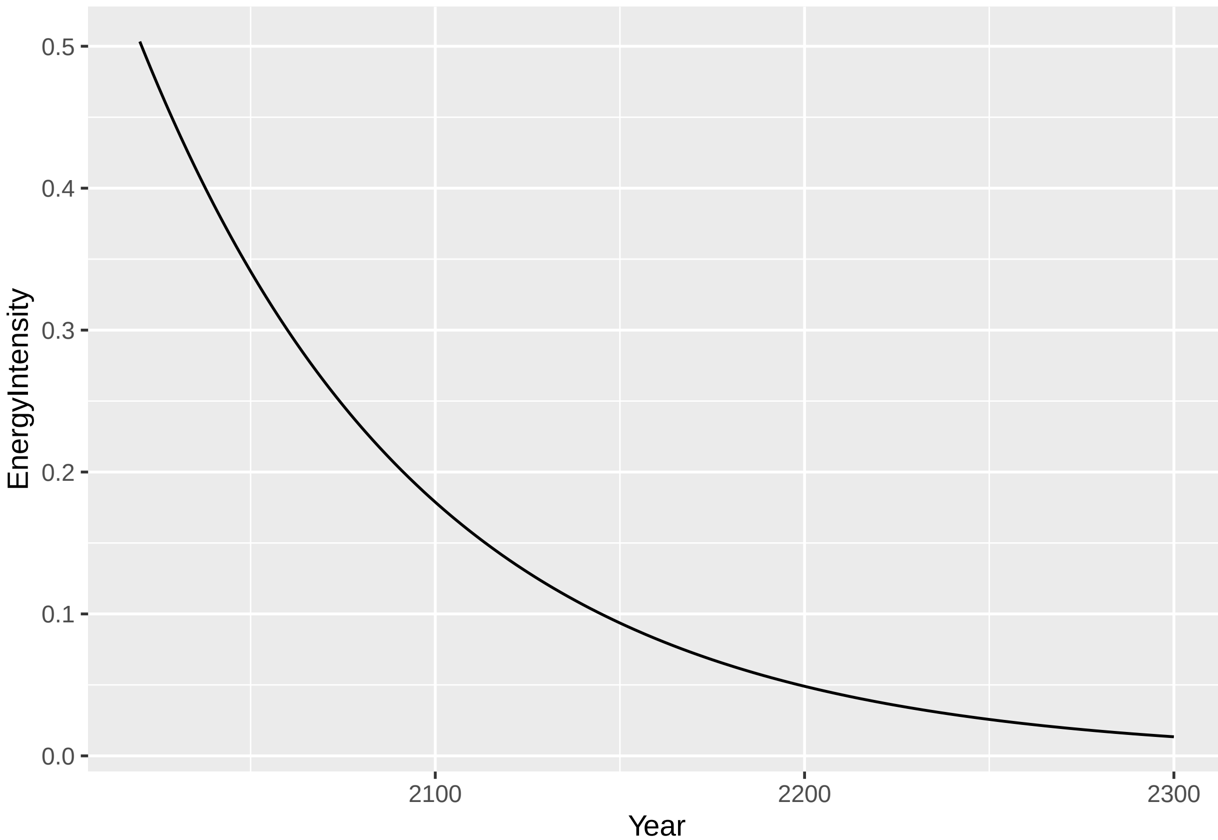
```
## [1] -0.01285456
```

```
EnergyIntensityModel = data.frame(Year = 2020:2300,
                             EnergyIntensity = Middle.df$EnergyIntensity[which(Middle.df$Year == 2020)]
                             GrowthRate =  aveEnergyIntGrowth)

EnergyIntensityModel[1, ]
```

```
##   Year EnergyIntensity  GrowthRate
## 1 2020       0.5033195 -0.01285456
```

```
for (i in 2:length(EnergyIntensityModel$Year)) {
  EnergyIntensityModel$EnergyIntensity[i] <- EnergyIntensityModel$EnergyIntensity[i-1]*(1 + EnergyIntens
}
```

```
EnergyIntensityModel %>% ggplot(aes(Year, EnergyIntensity)) +
  geom_line()
```

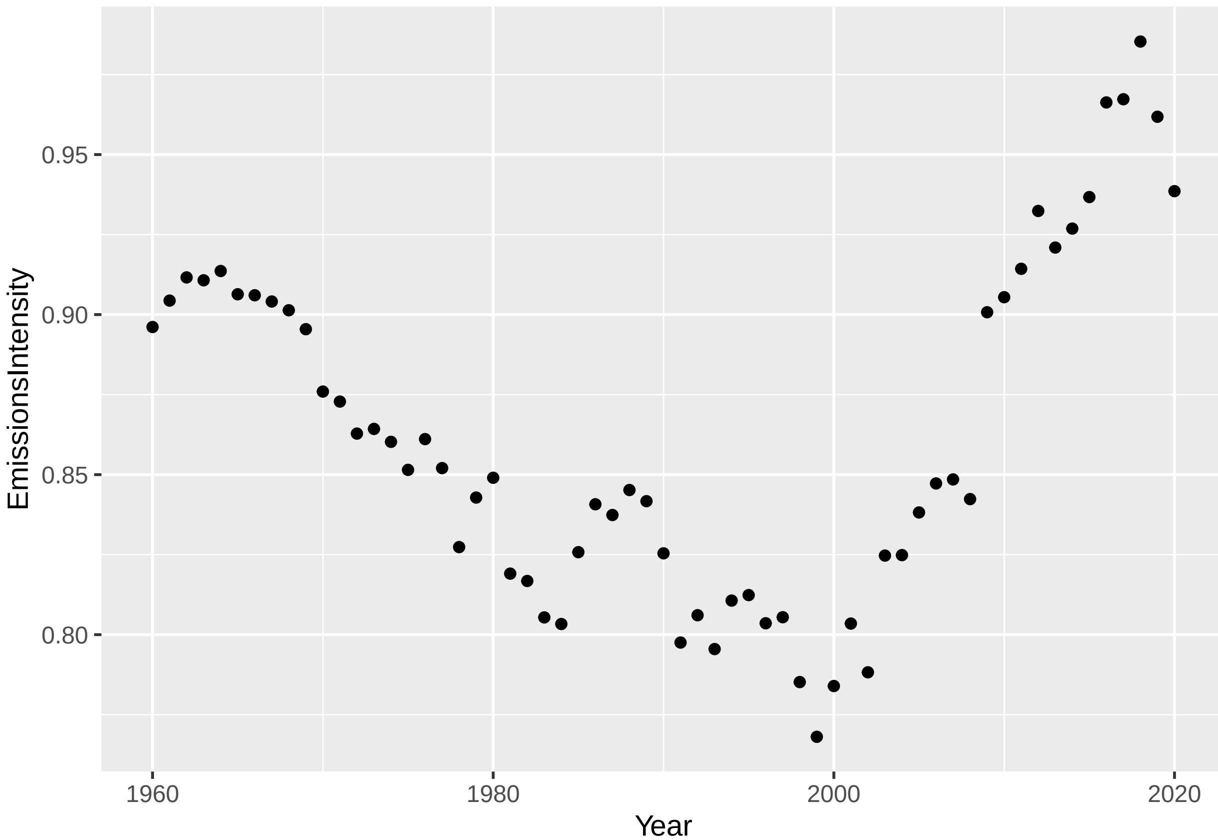


Emissions Intensity

We need to calculate the emissions intensity of ebnergy before we can model it in the future.

```
Middle.df$EmissionsIntensity <- Middle.df$Emissions/Middle.df$Energy
```

```
Middle.df %>%
  ggplot(aes(Year, EmissionsIntensity)) +
  geom_point()
```




```
round(Middle.df$EmissionsIntensity[2:61]/Middle.df$EmissionsIntensity[1:60]-1 , 3)
```

```
##  [1]  0.009  0.008 -0.001  0.003 -0.008  0.000 -0.002 -0.003 -0.007 -0.022
## [11] -0.004 -0.011  0.002 -0.005 -0.010  0.011 -0.011 -0.029  0.019  0.007
## [21] -0.035 -0.003 -0.014 -0.003  0.028  0.018 -0.004  0.009 -0.004 -0.019
## [31] -0.034  0.011 -0.013  0.019  0.002 -0.011  0.002 -0.025 -0.022  0.021
## [41]  0.025 -0.019  0.046  0.000  0.016  0.011  0.001 -0.007  0.069  0.005
## [51]  0.010  0.020 -0.012  0.006  0.011  0.032  0.001  0.019 -0.024 -0.024
```

```
aveEmissionsIntGrowth <- mean(Middle.df$EmissionsIntensity[2:61]/Middle.df$EmissionsIntensity[1:60]-1)
aveEmissionsIntGrowth
```

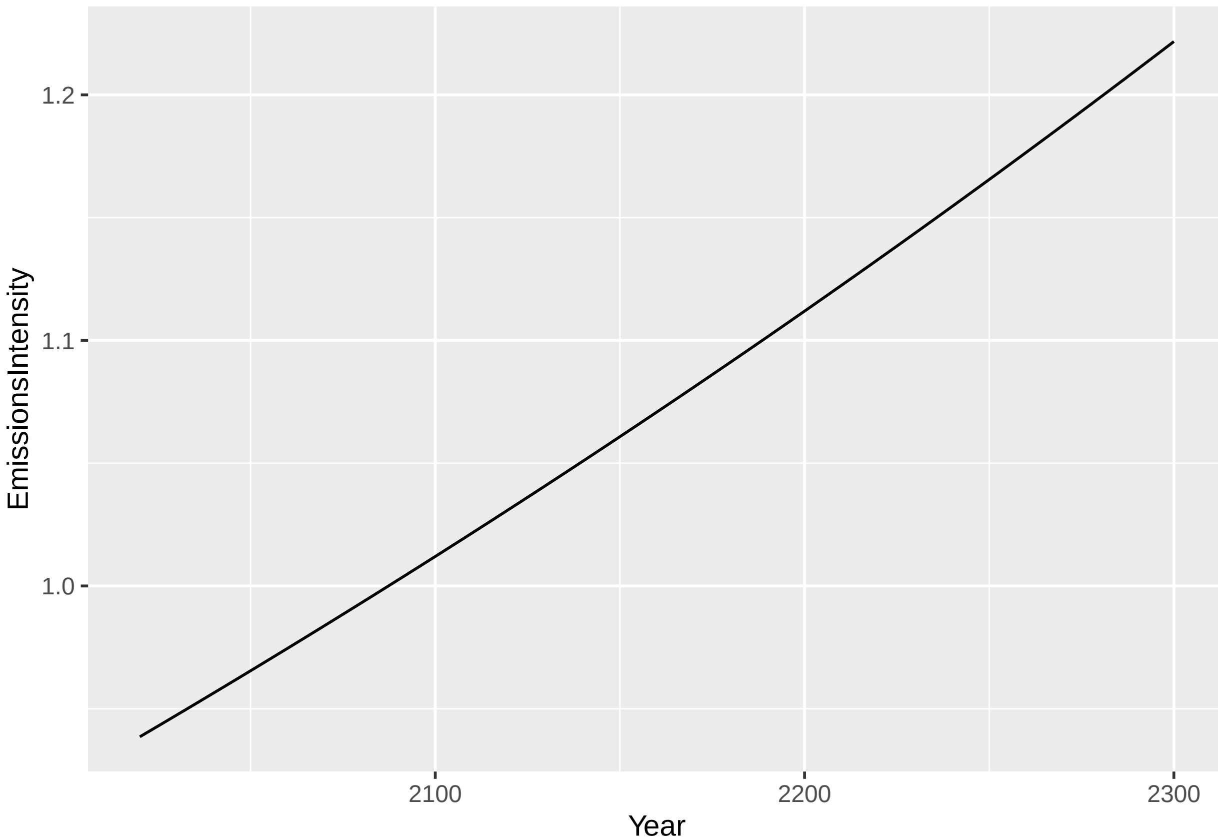
```
## [1] 0.0009419863
```

```
EmissionsIntensityModel = data.frame(Year = 2020:2300,
                             EmissionsIntensity = Middle.df$EmissionsIntensity[which(Middle.df$Year == 2
                             GrowthRate =  aveEmissionsIntGrowth)

EmissionsIntensityModel[1, ]
```

```
##   Year EmissionsIntensity   GrowthRate
## 1 2020          0.9385783 0.0009419863
```

```
for (i in 2:length(EmissionsIntensityModel$Year)) {
  EmissionsIntensityModel$EmissionsIntensity[i] <- EmissionsIntensityModel$EmissionsIntensity[i-1]*(1 +
}
```

```
EmissionsIntensityModel %>% ggplot(aes(Year, EmissionsIntensity)) +
  geom_line()
```



GDP

For GDP we will use the Cobb-Douglass Production function and the Solow Growth model.

```
Middle.df$A = .65 # text says use 1 as a starter value
Middle.df$K = 0
Middle.df$GDPmodeled = 0

Middle.df$K[1] = (savingsrate*Middle.df$A[1]/depreciation)^(1/(1- CobbDouglassalpha))*Middle.df$Populati
Middle.df$GDPmodeled[1] = CobbDouglassGDP(Middle.df$A[1], Middle.df$K[1], Middle.df$Population[1])

Middle.df %>% select(Year, Population, GDP, A, K, GDPmodeled) %>% head()
```

```
## # A tibble: 6 x 6
##    Year Population   GDP     A     K GDPmodeled
##   <dbl>      <dbl> <dbl> <dbl> <dbl>      <dbl>
## 1  1960      1194.  833.  0.65 1658.       829.
## 2  1961      1200.  859.  0.65    0          0
## 3  1962      1218.  889.  0.65    0          0
## 4  1963      1247.  906.  0.65    0          0
## 5  1964      1276.  994.  0.65    0          0
## 6  1965      1305. 1084.  0.65    0          0
```

Now that we have a starting value for A and Capital, we can implement our capital difference equation and use the new Population variable to estimate GDP over time.

```
TFPgrowth <- 0.0282 # 2% value is suggested by the text

for (i in 2:length(Middle.df$Year)) {
```

```
  Middle.df$A[i] = Middle.df$A[i-1]*(1+TFPgrowth)
  Middle.df$K[i] = Kapital(Middle.df$K[i-1], Middle.df$GDPmodeled[i-1])
  Middle.df$GDPmodeled[i] = CobbDouglassGDP(Middle.df$A[i], Middle.df$K[i], Middle.df$Population[i])
}
Middle.df %>% select(Year, Population, GDP, A, K, GDPmodeled) %>% tail()
```

```
## # A tibble: 6 x 6
##    Year Population    GDP     A      K GDPmodeled
##   <dbl>      <dbl>  <dbl> <dbl>  <dbl>      <dbl>
## 1  2015      2696. 11204.  3.00 15742.     11514.
## 2  2016      2729. 11921.  3.09 16471.     12060.
## 3  2017      2762. 12569.  3.17 17236.     12635.
## 4  2018      2795. 13259.  3.26 18039.     13237.
## 5  2019      2829. 13913.  3.35 18883.     13867.
## 6  2020      2862. 14534.  3.45 19768.     14524.
```

The text says to model future GDP assuming total factor productivity growth rate is 0.99 times the previous time period growth rate for Total Factor Productivity.

```
GDPModel = PopulationModel
GDPModel$GrowthRate = TFPgrowth
GDPModel$A = Middle.df$A[which(Middle.df$Year == 2020)]
GDPModel$K = Middle.df$K[which(Middle.df$Year == 2020)]
GDPModel$GDPmodeled = Middle.df$GDPmodeled[which(Middle.df$Year == 2020)]

for (i in 2:length(GDPModel$Year)) {
  GDPModel$GrowthRate[i] = GDPModel$GrowthRate[i-1]*.99
  GDPModel$A[i] = GDPModel$A[i-1]*(1 + GDPModel$GrowthRate[i])
  GDPModel$K[i] = Kapital(GDPModel$K[i-1], GDPModel$GDPmodeled[i-1])
  GDPModel$GDPmodeled[i] = CobbDouglassGDP(GDPModel$A[i], GDPModel$K[i], GDPModel$Population[i])
}

GDPModel$GDPperCapita = GDPModel$GDPmodeled/GDPModel$Population

tail(GDPModel)
```

```
##     Year Population  GrowthRate        A         K GDPmodeled GDPperCapita
## 276 2295    3572.61 0.001777983 46.26167  993945.0   509427.3     142.5925
## 277 2296    3572.61 0.001760203 46.34310  996435.9   510579.5     142.9150
## 278 2297    3572.61 0.001742601 46.42386  998908.2   511722.8     143.2350
## 279 2298    3572.61 0.001725175 46.50395 1001362.0   512857.2     143.5525
## 280 2299    3572.61 0.001707923 46.58337 1003797.2   513982.8     143.8676
## 281 2300    3572.61 0.001690844 46.66214 1006214.1   515099.6     144.1802
```

Emissions from Kaya Calculation Rich

Now that we have modeled the four elements of the Kaya Identity (Population, GDP per capita, Energy Intensity, and Emissions Intensity), we can use this to model future emissions.

```
Kaya.Middle <- PopulationModel[, c("Year", "Population")] %>%
  left_join(GDPModel[, c("Year", "GDPperCapita")]) %>%
  left_join(EnergyIntensityModel[, c("Year", "EnergyIntensity")]) %>%
  left_join(EmissionsIntensityModel[, c("Year", "EmissionsIntensity")])
```

```
## Joining with `by = join_by(Year)`
## Joining with `by = join_by(Year)`
```

```
## Joining with `by = join_by(Year)`
```

```
head(Kaya.Middle)
```

```
##   Year Population GDPperCapita EnergyIntensity EmissionsIntensity
## 1 2020   2861.881     5.074934       0.5033195          0.9385783
## 2 2021   2893.712     5.253066       0.4968496          0.9394624
## 3 2022   2924.288     5.436524       0.4904628          0.9403474
## 4 2023   2953.642     5.625308       0.4841581          0.9412331
## 5 2024   2981.808     5.819419       0.4779344          0.9421198
## 6 2025   3008.821     6.018863       0.4717908          0.9430072
```

Now calculate the Kaya Identity with all of the modeled variables and again four separate times with each of the four variables held constant at their 2020 levels.

```
Kaya.Middle = Kaya.Middle %>%
  mutate(Emissions = Population*GDPperCapita*EnergyIntensity*EmissionsIntensity,
         EmissionsPopConstant = Population[1]*GDPperCapita*EnergyIntensity*EmissionsIntensity,
         EmissionsGDPperCapConstant = Population*GDPperCapita[1]*EnergyIntensity*EmissionsIntensity,
         EmissionsEnergyIntensityConstant = Population*GDPperCapita*EnergyIntensity[1]*EmissionsIntensi
         EmissionsEmissionsIntensityConstant = Population*GDPperCapita*EnergyIntensity*EmissionsIntensi
head(Kaya.Middle)
```

```
##   Year Population GDPperCapita EnergyIntensity EmissionsIntensity Emissions
## 1 2020   2861.881     5.074934       0.5033195          0.9385783  6861.138
## 2 2021   2893.712     5.253066       0.4968496          0.9394624  7095.328
## 3 2022   2924.288     5.436524       0.4904628          0.9403474  7332.225
## 4 2023   2953.642     5.625308       0.4841581          0.9412331  7571.615
## 5 2024   2981.808     5.819419       0.4779344          0.9421198  7813.287
## 6 2025   3008.821     6.018863       0.4717908          0.9430072  8057.036
##   EmissionsPopConstant EmissionsGDPperCapConstant
## 1             6861.138                   6861.138
## 2             7017.279                   6854.724
## 3             7175.749                   6844.549
## 4             7336.387                   6830.816
## 5             7499.039                   6813.723
## 6             7663.559                   6793.463
##   EmissionsEnergyIntensityConstant EmissionsEmissionsIntensityConstant
## 1                         6861.138                            6861.138
## 2                         7187.723                            7088.650
## 3                         7524.428                            7318.431
## 4                         7871.275                            7550.258
## 5                         8228.282                            7783.916
## 6                         8595.470                            8019.195
```

Lowest Income

Now we repeat the same exercise with the lowest income region data.

```
Poor.df$Population[which(Poor.df$Year == 2020)]/Poor.df$Population[which(Poor.df$Year == 2019)]-1
```

```
## [1] 0.02584889
```

```
round(Poor.df$Population[2:61]/Poor.df$Population[1:60]-1 , 3)
```

```
##  [1] 0.024 0.024 0.024 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.024 0.024
## [13] 0.024 0.024 0.024 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.024
## [25] 0.024 0.024 0.024 0.024 0.023 0.023 0.022 0.021 0.021 0.021 0.020 0.020
```

```
## [37] 0.020 0.020 0.020 0.019 0.019 0.018 0.018 0.017 0.017 0.017 0.017 0.017
## [49] 0.017 0.017 0.029 0.029 0.029 0.028 0.027 0.027 0.026 0.026 0.026 0.026
```

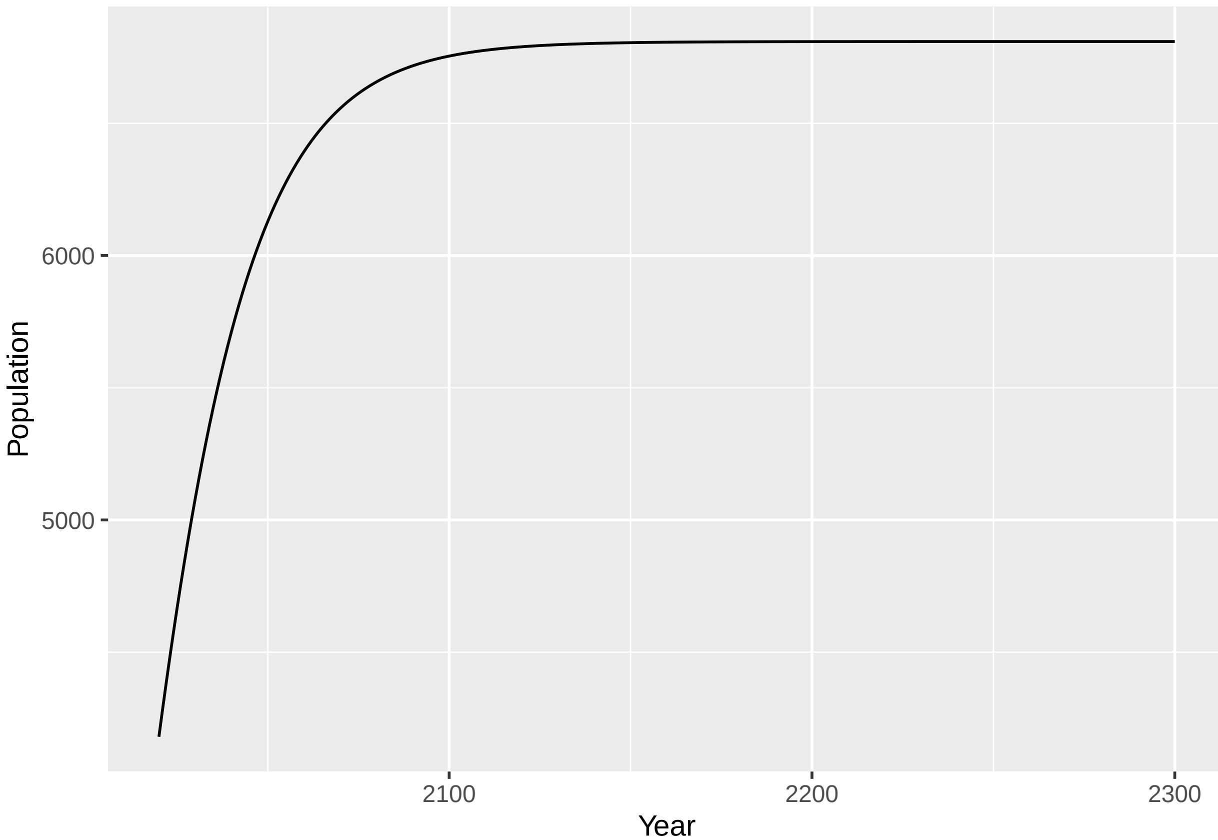
```
PopulationModel = data.frame(Year = 2020:2300,
                             Population = Poor.df$Population[which(Poor.df$Year == 2020)],
                             GrowthRate =  Poor.df$Population[which(Poor.df$Year == 2020)]/Poor.df$Popul

head(PopulationModel)
```

```
##   Year Population GrowthRate
## 1 2020   4179.822 0.02584889
## 2 2021   4179.822 0.02584889
## 3 2022   4179.822 0.02584889
## 4 2023   4179.822 0.02584889
## 5 2024   4179.822 0.02584889
## 6 2025   4179.822 0.02584889
```

```
for (i in 2:length(PopulationModel$Year)) {
  PopulationModel$GrowthRate[i] <- PopulationModel$GrowthRate[i-1]*.95
  PopulationModel$Population[i] <- PopulationModel$Population[i-1]*(1 + PopulationModel$GrowthRate[i])
}
head(PopulationModel)
```

```
##   Year Population GrowthRate
## 1 2020   4179.822 0.02584889
## 2 2021   4282.464 0.02455644
## 3 2022   4382.367 0.02332862
## 4 2023   4479.490 0.02216219
## 5 2024   4573.802 0.02105408
## 6 2025   4665.284 0.02000138
```

```
PopulationModel %>% ggplot(aes(Year, Population)) +
  geom_line()
```

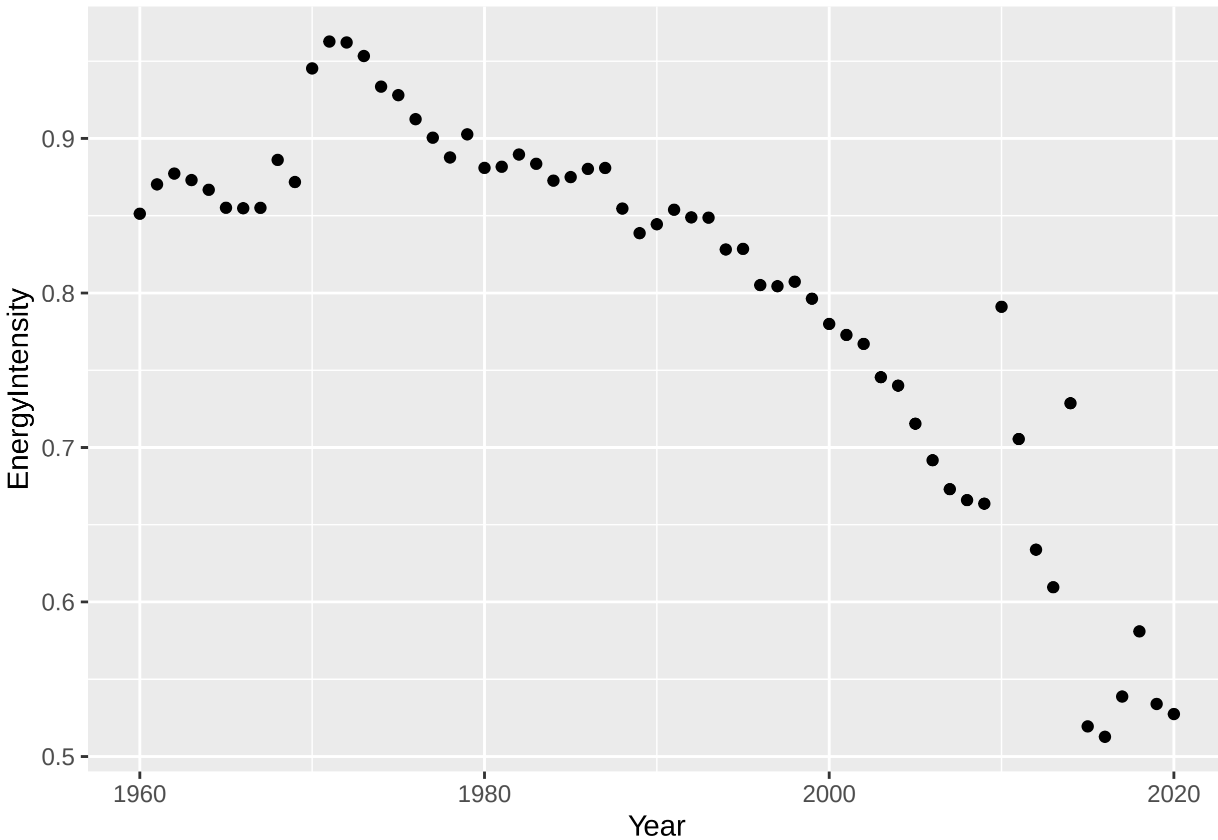


Energy Intensity

We need to calculate the energy intensity of GDP before we can model it in the future.

```
Poor.df$EnergyIntensity <- Poor.df$Energy/Poor.df$GDP
```

```
Poor.df %>%
  ggplot(aes(Year, EnergyIntensity)) +
  geom_point()
```




```
round(Poor.df$EnergyIntensity[2:61]/Poor.df$EnergyIntensity[1:60]-1 , 3)
```

```
##  [1]  0.022  0.008 -0.005 -0.007 -0.013  0.000  0.000  0.036 -0.016  0.084
## [11]  0.018 -0.001 -0.009 -0.021 -0.006 -0.017 -0.013 -0.014  0.017 -0.024
## [21]  0.001  0.009 -0.007 -0.012  0.003  0.006  0.001 -0.030 -0.019  0.007
## [31]  0.011 -0.006  0.000 -0.024  0.000 -0.028 -0.001  0.004 -0.014 -0.021
## [41] -0.009 -0.008 -0.028 -0.007 -0.033 -0.033 -0.027 -0.011 -0.003  0.192
## [51] -0.108 -0.101 -0.038  0.195 -0.287 -0.013  0.051  0.078 -0.081 -0.012
```

```
aveEnergyIntGrowth <- mean(Poor.df$EnergyIntensity[2:61]/Poor.df$EnergyIntensity[1:60]-1)
aveEnergyIntGrowth
```

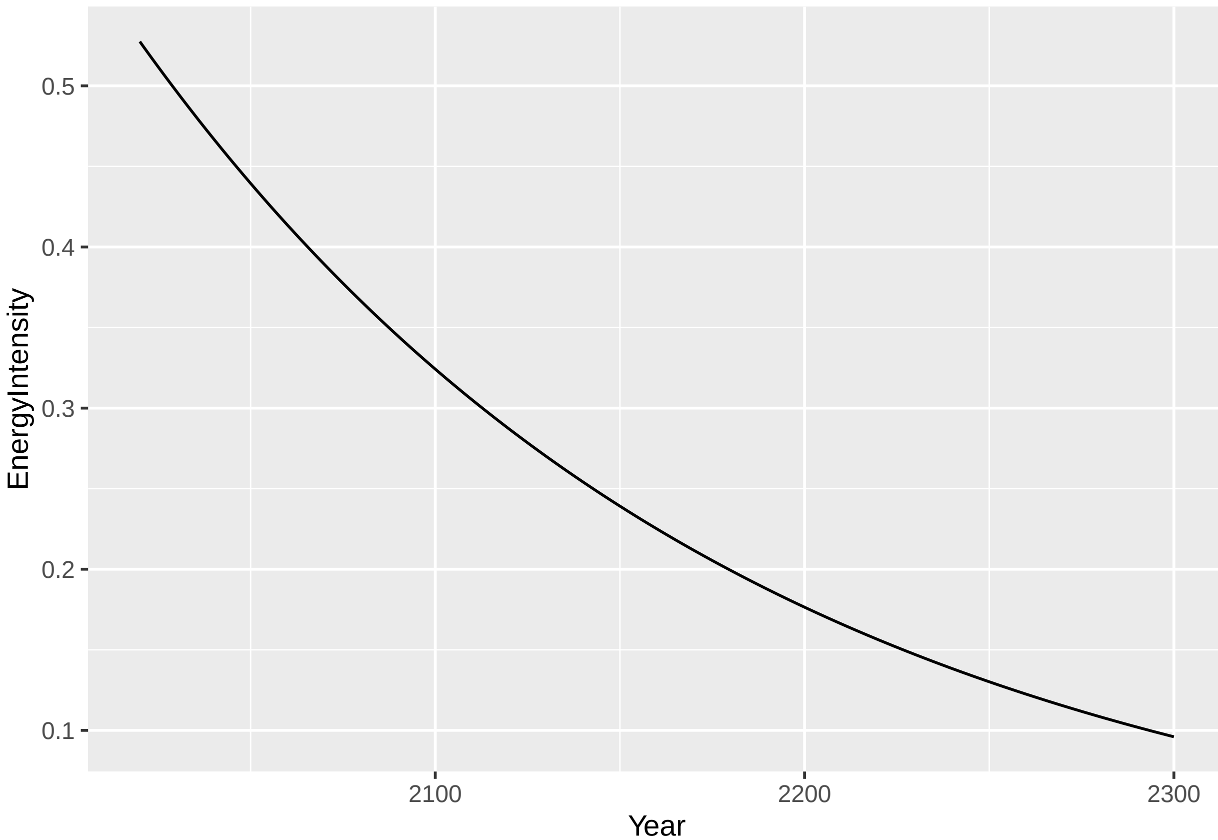
```
## [1] -0.006066107
```

```
EnergyIntensityModel = data.frame(Year = 2020:2300,
                              EnergyIntensity = Poor.df$EnergyIntensity[which(Poor.df$Year == 2020)],
                              GrowthRate =  aveEnergyIntGrowth)

EnergyIntensityModel[1, ]
```

```
##   Year EnergyIntensity   GrowthRate
## 1 2020       0.5274792 -0.006066107
```

```
for (i in 2:length(EnergyIntensityModel$Year)) {
  EnergyIntensityModel$EnergyIntensity[i] <- EnergyIntensityModel$EnergyIntensity[i-1]*(1 + EnergyInten
}
```

```
EnergyIntensityModel %>% ggplot(aes(Year, EnergyIntensity)) +
  geom_line()
```

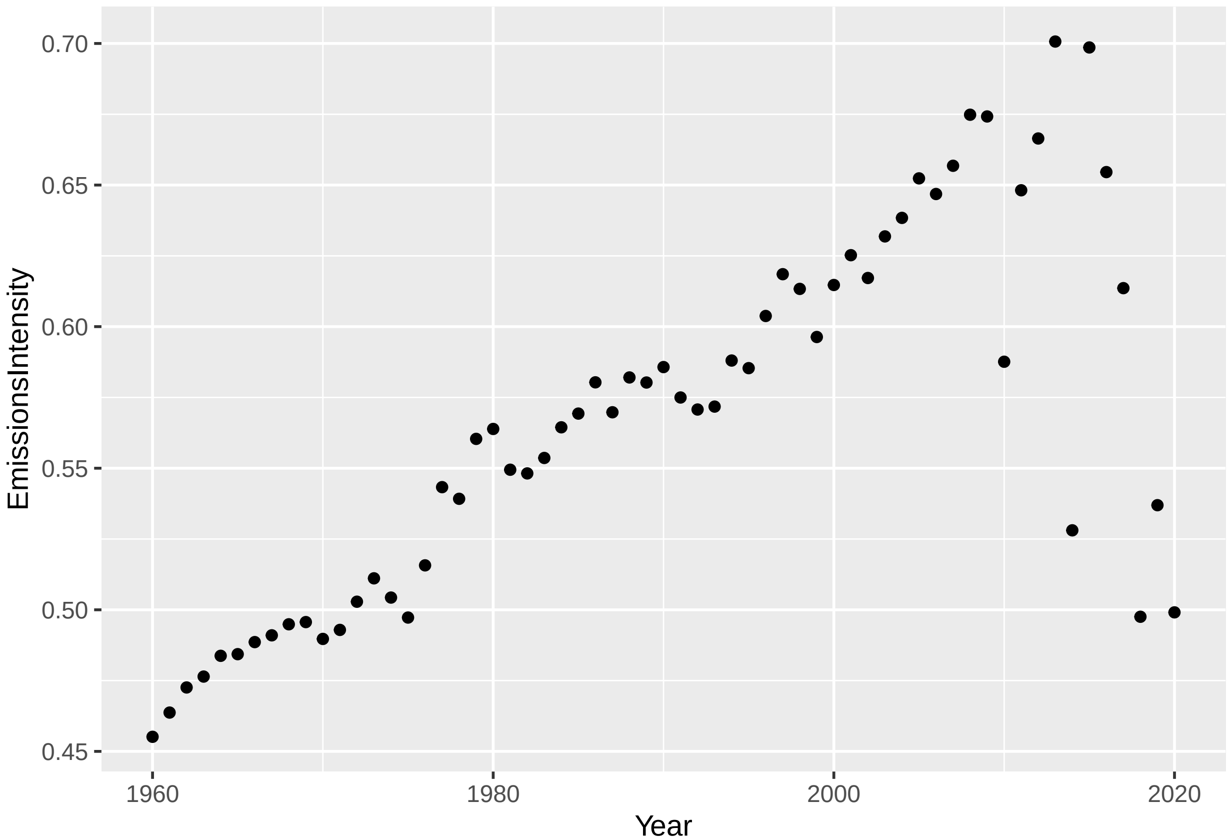


Emissions Intensity

We need to calculate the emissions intensity of ebnergy before we can model it in the future.

```
Poor.df$EmissionsIntensity <- Poor.df$Emissions/Poor.df$Energy
```

```
Poor.df %>%
  ggplot(aes(Year, EmissionsIntensity)) +
  geom_point()
```




```
round(Poor.df$EmissionsIntensity[2:61]/Poor.df$EmissionsIntensity[1:60]-1 , 3)
```

```
##  [1]  0.019  0.019  0.008  0.015  0.001  0.009  0.005  0.008  0.002 -0.012
## [11]  0.007  0.020  0.016 -0.013 -0.014  0.037  0.054 -0.008  0.039  0.006
## [21] -0.026 -0.002  0.010  0.020  0.009  0.019 -0.018  0.022 -0.003  0.009
## [31] -0.018 -0.007  0.002  0.028 -0.005  0.031  0.024 -0.008 -0.028  0.031
## [41]  0.017 -0.013  0.024  0.010  0.022 -0.008  0.015  0.027 -0.001 -0.128
## [51]  0.103  0.028  0.051 -0.246  0.323 -0.063 -0.063 -0.189  0.079 -0.070
```

```
aveEmissionsIntGrowth <- mean(Poor.df$EmissionsIntensity[2:61]/Poor.df$EmissionsIntensity[1:60]-1)
aveEmissionsIntGrowth
```

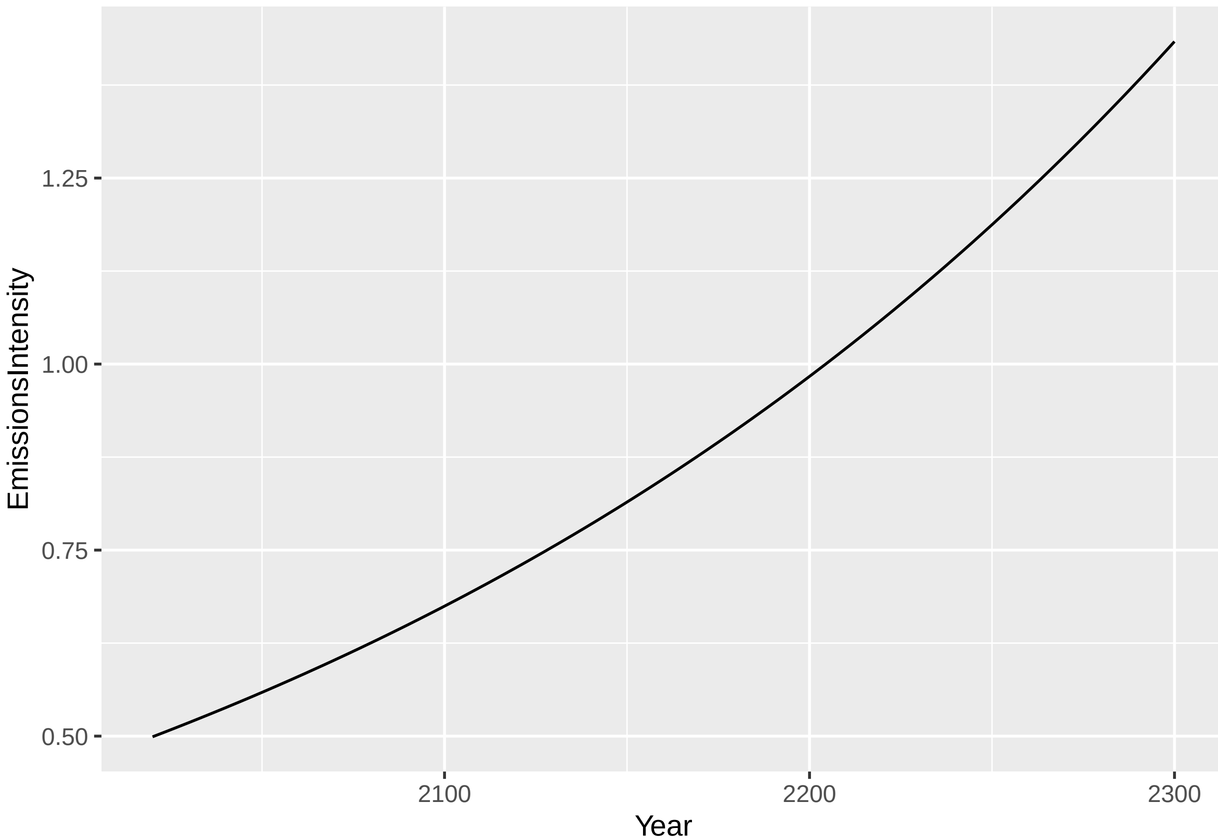
```
## [1] 0.003775118
```

```
EmissionsIntensityModel = data.frame(Year = 2020:2300,
                             EmissionsIntensity = Poor.df$EmissionsIntensity[which(Poor.df$Year == 2020)
                             GrowthRate =  aveEmissionsIntGrowth)

EmissionsIntensityModel[1, ]
```

```
##   Year EmissionsIntensity  GrowthRate
## 1 2020          0.4990987 0.003775118
```

```
for (i in 2:length(EmissionsIntensityModel$Year)) {
  EmissionsIntensityModel$EmissionsIntensity[i] <- EmissionsIntensityModel$EmissionsIntensity[i-1]*(1 +
}
```

```
EmissionsIntensityModel %>% ggplot(aes(Year, EmissionsIntensity)) +
  geom_line()
```



GDP

For GDP we will use the Cobb-Douglass Production function and the Solow Growth model.

```
Poor.df$A = .283 # text says use 1 as a starter value
Poor.df$K = 0
Poor.df$GDPmodeled = 0

Poor.df$K[1] = (savingsrate*Poor.df$A[1]/depreciation)^(1/(1- CobbDouglassalpha))*Poor.df$Population[1]
Poor.df$GDPmodeled[1] = CobbDouglassGDP(Poor.df$A[1], Poor.df$K[1], Poor.df$Population[1])

Poor.df %>% select(Year, Population, GDP, A, K, GDPmodeled) %>% head()
```

```
## # A tibble: 6 x 6
##    Year Population   GDP     A     K GDPmodeled
##   <dbl>      <dbl> <dbl> <dbl> <dbl>      <dbl>
## 1  1960      1067.  262. 0.283  524.       262.
## 2  1961      1092.  272. 0.283    0          0
## 3  1962      1119.  283. 0.283    0          0
## 4  1963      1146.  301. 0.283    0          0
## 5  1964      1174.  320. 0.283    0          0
## 6  1965      1203.  351. 0.283    0          0
```

Now that we have a starting value for A and Capital, we can implement our capital difference equation and use the new Population variable to estimate GDP over time.

```
TFPgrowth <- 0.0205 # 2% value is suggested by the text

for (i in 2:length(Poor.df$Year)) {
```

```r
  Poor.df$A[i] = Poor.df$A[i-1]*(1+TFPgrowth)
  Poor.df$K[i] = Kapital(Poor.df$K[i-1], Poor.df$GDPmodeled[i-1])
  Poor.df$GDPmodeled[i] = CobbDouglassGDP(Poor.df$A[i], Poor.df$K[i], Poor.df$Population[i])
}
Poor.df %>% select(Year, Population, GDP, A, K, GDPmodeled) %>% tail()
```

```
## # A tibble: 6 x 6
##    Year Population   GDP     A     K GDPmodeled
##   <dbl>      <dbl> <dbl> <dbl> <dbl>      <dbl>
## 1  2015      3677. 4019. 0.864 4426.      3297.
## 2  2016      3775. 4050. 0.882 4643.      3469.
## 3  2017      3873. 3918. 0.900 4873.      3649.
## 4  2018      3973. 4051. 0.918 5115.      3837.
## 5  2019      4075. 4228. 0.937 5371.      4035.
## 6  2020      4180. 4207. 0.956 5641.      4244.
```

The text says to model future GDP assuming total factor productivity growth rate is 0.99 times the previous time period growth rate for Total Factor Productivity.

```r
GDPModel = PopulationModel
GDPModel$GrowthRate = TFPgrowth
GDPModel$A = Poor.df$A[which(Poor.df$Year == 2020)]
GDPModel$K = Poor.df$K[which(Poor.df$Year == 2020)]
GDPModel$GDPmodeled = Poor.df$GDPmodeled[which(Poor.df$Year == 2020)]

for (i in 2:length(GDPModel$Year)) {
  GDPModel$GrowthRate[i] = GDPModel$GrowthRate[i-1]*.99
  GDPModel$A[i] = GDPModel$A[i-1]*(1 + GDPModel$GrowthRate[i])
  GDPModel$K[i] = Kapital(GDPModel$K[i-1], GDPModel$GDPmodeled[i-1])
  GDPModel$GDPmodeled[i] = CobbDouglassGDP(GDPModel$A[i], GDPModel$K[i], GDPModel$Population[i])
}

GDPModel$GDPperCapita = GDPModel$GDPmodeled/GDPModel$Population

tail(GDPModel)
```

```
##     Year Population  GrowthRate        A        K GDPmodeled GDPperCapita
## 276 2295   6809.638 0.001292505 6.338111 159239.7   81071.70     11.90544
## 277 2296   6809.638 0.001279580 6.346221 159530.0   81205.02     11.92501
## 278 2297   6809.638 0.001266785 6.354260 159818.0   81337.23     11.94443
## 279 2298   6809.638 0.001254117 6.362229 160103.7   81468.32     11.96368
## 280 2299   6809.638 0.001241576 6.370128 160387.0   81598.32     11.98277
## 281 2300   6809.638 0.001229160 6.377958 160667.9   81727.22     12.00170
```

Emissions from Kaya Calculation Rich

Now that we have modeled the four elements of the Kaya Identity (Population, GDP per capita, Energy Intensity, and Emissions Intensity), we can use this to model future emissions.

```r
Kaya.Poor <- PopulationModel[, c("Year", "Population")] %>%
  left_join(GDPModel[, c("Year", "GDPperCapita")]) %>%
  left_join(EnergyIntensityModel[, c("Year", "EnergyIntensity")]) %>%
  left_join(EmissionsIntensityModel[, c("Year", "EmissionsIntensity")])
```

```
## Joining with `by = join_by(Year)`
## Joining with `by = join_by(Year)`
```

```
## Joining with `by = join_by(Year)`
```

```
head(Kaya.Poor)
```

```
##   Year Population GDPperCapita EnergyIntensity EmissionsIntensity
## 1 2020   4179.822     1.015320       0.5274792          0.4990987
## 2 2021   4282.464     1.041114       0.5242795          0.5009828
## 3 2022   4382.367     1.067610       0.5210992          0.5028741
## 4 2023   4479.490     1.094782       0.5179381          0.5047725
## 5 2024   4573.802     1.122604       0.5147963          0.5066781
## 6 2025   4665.284     1.151053       0.5116734          0.5085909
```

Now calculate the Kaya Identity with all of the modeled variables and again four separate times with each of the four variables held constant at their 2020 levels.

```
Kaya.Poor = Kaya.Poor %>%
  mutate(Emissions = Population*GDPperCapita*EnergyIntensity*EmissionsIntensity,
         EmissionsPopConstant = Population[1]*GDPperCapita*EnergyIntensity*EmissionsIntensity,
         EmissionsGDPperCapConstant = Population*GDPperCapita[1]*EnergyIntensity*EmissionsIntensity,
         EmissionsEnergyIntensityConstant = Population*GDPperCapita*EnergyIntensity[1]*EmissionsIntensi
         EmissionsEmissionsIntensityConstant = Population*GDPperCapita*EnergyIntensity*EmissionsIntensi
head(Kaya.Poor)
```

```
##   Year Population GDPperCapita EnergyIntensity EmissionsIntensity Emissions
## 1 2020   4179.822     1.015320       0.5274792          0.4990987  1117.255
## 2 2021   4282.464     1.041114       0.5242795          0.5009828  1171.056
## 3 2022   4382.367     1.067610       0.5210992          0.5028741  1226.030
## 4 2023   4479.490     1.094782       0.5179381          0.5047725  1282.123
## 5 2024   4573.802     1.122604       0.5147963          0.5066781  1339.280
## 6 2025   4665.284     1.151053       0.5116734          0.5085909  1397.446
##   EmissionsPopConstant EmissionsGDPperCapConstant
## 1             1117.255                   1117.255
## 2             1142.988                   1142.042
## 3             1169.365                   1165.980
## 4             1196.352                   1189.063
## 5             1223.916                   1211.289
## 6             1252.030                   1232.657
##   EmissionsEnergyIntensityConstant EmissionsEmissionsIntensityConstant
## 1                         1117.255                            1117.255
## 2                         1178.203                            1166.652
## 3                         1241.041                            1216.826
## 4                         1305.742                            1267.712
## 5                         1372.275                            1319.245
## 6                         1440.613                            1371.364
```

Bringing Together the Kaya Emissions Data across Regions

We have worked to create emissions predictions for three different regions across five different scenarios, business as usual, and then population, GDP per capita, energy intensity, and emissions intensity held constant.

What might some interesting graphs be to examine these data?

To complete Exercise 13.5, we need to add together the emissions across regions before we input these emissions into the Five Box Carbon Dioxide model.

Previously, we calculated the global Kaya Identity Emissions directly globally. In this exercise we estimated the emissions for each region, so the global emissions are the sum of the emissions for each region.

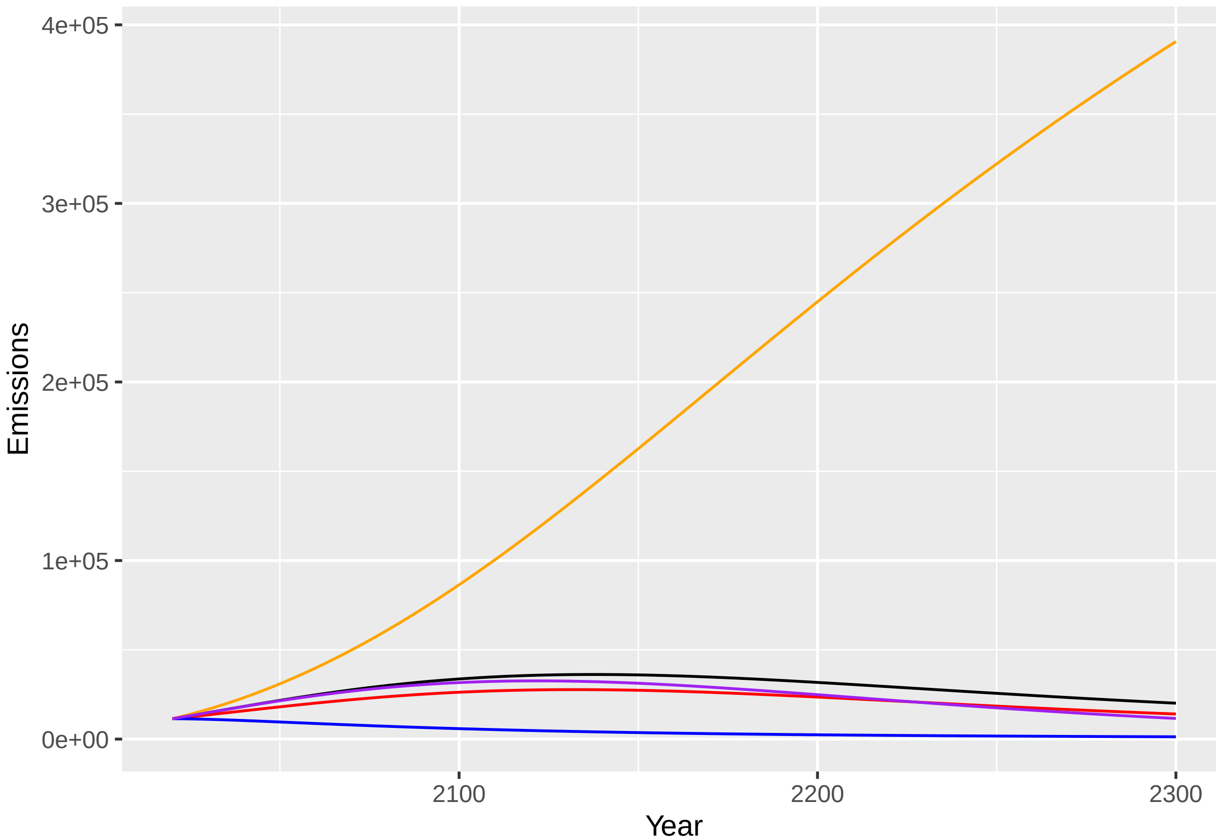
```r
Kaya.global <- Kaya.Rich

names(Kaya.global)
```

```
##  [1] "Year"                                "Population"
##  [3] "GDPperCapita"                        "EnergyIntensity"
##  [5] "EmissionsIntensity"                  "Emissions"
##  [7] "EmissionsPopConstant"                "EmissionsGDPperCapConstant"
##  [9] "EmissionsEnergyIntensityConstant"    "EmissionsEmissionsIntensityConstant"
```

```r
Kaya.global$Emissions <-  Kaya.global$Emissions + Kaya.Middle$Emissions + Kaya.Poor$Emissions
Kaya.global$EmissionsPopConstant <-  Kaya.global$EmissionsPopConstant +
  Kaya.Middle$EmissionsPopConstant + Kaya.Poor$EmissionsPopConstant
Kaya.global$EmissionsGDPperCapConstant <-  Kaya.global$EmissionsGDPperCapConstant +
  Kaya.Middle$EmissionsGDPperCapConstant + Kaya.Poor$EmissionsGDPperCapConstant
Kaya.global$EmissionsEnergyIntensityConstant <-  Kaya.global$EmissionsEnergyIntensityConstant +
  Kaya.Middle$EmissionsEnergyIntensityConstant + Kaya.Poor$EmissionsEnergyIntensityConstant
Kaya.global$EmissionsEmissionsIntensityConstant <-  Kaya.global$EmissionsEmissionsIntensityConstant +
  Kaya.Middle$EmissionsEmissionsIntensityConstant + Kaya.Poor$EmissionsEmissionsIntensityConstant
```

We can now feed these emissions predictions into our climate model to predict how CO2 concentrations and temperatures might be under the various scenarios. However, before we do that, let's plot the emissions variables in the difference scenarios.

```r
Kaya.global %>%
  ggplot(aes(x = Year)) +
  geom_line(aes(y = Emissions)) +
  geom_line(aes(y = EmissionsPopConstant), color = "red") +
  geom_line(aes(y = EmissionsGDPperCapConstant), color = "blue") +
  geom_line(aes(y = EmissionsEnergyIntensityConstant), color = "orange") +
  geom_line(aes(y = EmissionsEmissionsIntensityConstant), color = "purple")
```



Calculate CO2 concentrations and temperature for emissions paths

```
Kaya.global <- left_join(Kaya.global, HistoricData, by = c("Year" = "year") )

Box1col <- which(colnames(Kaya.global) == "Box1")

#Implement Five box model for CO2 concentrations
for (i in 2:length(Kaya.global$Year)) {
 Kaya.global[i, Box1col:(Box1col+4)] <- CO2difference(Kaya.global[i - 1, Box1col:(Box1col+4)],
                                                       Kaya.global$Emissions[i - 1])
 }
# Calculate total CO2 concentration in atmosphere
Kaya.global$CO2conc <- Kaya.global$Box1 + Kaya.global$Box2 + Kaya.global$Box3 +
  Kaya.global$Box4 + Kaya.global$Box5
# Calculate radiative forcing
Kaya.global$RF <- RadForc(Kaya.global$CO2conc)
# Calculate Temperatures
for (i in 2:length(Kaya.global$Year)) {
    temp <- Temps(Kaya.global$TempAtm[i-1], Kaya.global$TempOcean[i-1], Kaya.global$RF[i])
  Kaya.global$TempAtm[i] <- temp["atm"]
  Kaya.global$TempOcean[i] <- temp["ocean"]
}
```

Let's extract the emissions, CO2 concentrations, and atmospheric temperature for the different scenarios and put them in a separate dataframe for convenience.

```
Exercise13.5 <- Kaya.global %>%
  select(Year, Emissions, CO2conc, TempAtm) %>%
```

```
  mutate(ConstantVariable = "None")
```

Now we rerun the Climate model for the other four scenarios and grab the same variables. Start with Population.

```
#Implement Five box model for CO2 concentrations
for (i in 2:length(Kaya.global$Year)) {
 Kaya.global[i, Box1col:(Box1col+4)] <- CO2difference(Kaya.global[i - 1, Box1col:(Box1col+4)],
                                                       Kaya.global$EmissionsPopConstant[i - 1]) # CHANG
 }
# Calculate total CO2 concentration in atmosphere
Kaya.global$CO2conc <- Kaya.global$Box1 + Kaya.global$Box2 + Kaya.global$Box3 +
  Kaya.global$Box4 + Kaya.global$Box5
# Calculate radiative forcing
Kaya.global$RF <- RadForc(Kaya.global$CO2conc)
# Calculate Temperatures
for (i in 2:length(Kaya.global$Year)) {
    temp <- Temps(Kaya.global$TempAtm[i-1], Kaya.global$TempOcean[i-1], Kaya.global$RF[i])
  Kaya.global$TempAtm[i] <- temp["atm"]
  Kaya.global$TempOcean[i] <- temp["ocean"]
}

Exercise13.5 <- Kaya.global %>%
  select(Year, EmissionsPopConstant, CO2conc, TempAtm) %>%
  rename(Emissions = EmissionsPopConstant) %>%
  mutate(ConstantVariable = "Population") %>%
  full_join(Exercise13.5)
```

```
## Joining with `by = join_by(Year, Emissions, CO2conc, TempAtm,
## ConstantVariable)`
```

Keep Per capita GDP constant

```
#Implement Five box model for CO2 concentrations
for (i in 2:length(Kaya.global$Year)) {
 Kaya.global[i, Box1col:(Box1col+4)] <- CO2difference(Kaya.global[i - 1, Box1col:(Box1col+4)],
                                                       Kaya.global$EmissionsGDPperCapConstant[i - 1]) #
 }
# Calculate total CO2 concentration in atmosphere
Kaya.global$CO2conc <- Kaya.global$Box1 + Kaya.global$Box2 + Kaya.global$Box3 +
  Kaya.global$Box4 + Kaya.global$Box5
# Calculate radiative forcing
Kaya.global$RF <- RadForc(Kaya.global$CO2conc)
# Calculate Temperatures
for (i in 2:length(Kaya.global$Year)) {
    temp <- Temps(Kaya.global$TempAtm[i-1], Kaya.global$TempOcean[i-1], Kaya.global$RF[i])
  Kaya.global$TempAtm[i] <- temp["atm"]
  Kaya.global$TempOcean[i] <- temp["ocean"]
}

Exercise13.5 <- Kaya.global %>%
  select(Year, EmissionsGDPperCapConstant, CO2conc, TempAtm) %>%
  rename(Emissions = EmissionsGDPperCapConstant) %>%
  mutate(ConstantVariable = "GDPperCapita") %>%
  full_join(Exercise13.5)
```

```
## Joining with `by = join_by(Year, Emissions, CO2conc, TempAtm,
## ConstantVariable)`
```

Keep Energy Intensity constant

```
#Implement Five box model for CO2 concentrations
for (i in 2:length(Kaya.global$Year)) {
 Kaya.global[i, Box1col:(Box1col+4)] <- CO2difference(Kaya.global[i - 1, Box1col:(Box1col+4)],
                                                       Kaya.global$EmissionsEnergyIntensityConstant[i -
 }
# Calculate total CO2 concentration in atmosphere
Kaya.global$CO2conc <- Kaya.global$Box1 + Kaya.global$Box2 + Kaya.global$Box3 +
  Kaya.global$Box4 + Kaya.global$Box5
# Calculate radiative forcing
Kaya.global$RF <- RadForc(Kaya.global$CO2conc)
# Calculate Temperatures
for (i in 2:length(Kaya.global$Year)) {
    temp <- Temps(Kaya.global$TempAtm[i-1], Kaya.global$TempOcean[i-1], Kaya.global$RF[i])
  Kaya.global$TempAtm[i] <- temp["atm"]
  Kaya.global$TempOcean[i] <- temp["ocean"]
}

Exercise13.5 <- Kaya.global %>%
  select(Year, EmissionsEnergyIntensityConstant, CO2conc, TempAtm) %>%
  rename(Emissions = EmissionsEnergyIntensityConstant) %>%
  mutate(ConstantVariable = "EnergyIntensity") %>%
  full_join(Exercise13.5)
```

```
## Joining with `by = join_by(Year, Emissions, CO2conc, TempAtm,
## ConstantVariable)`
```

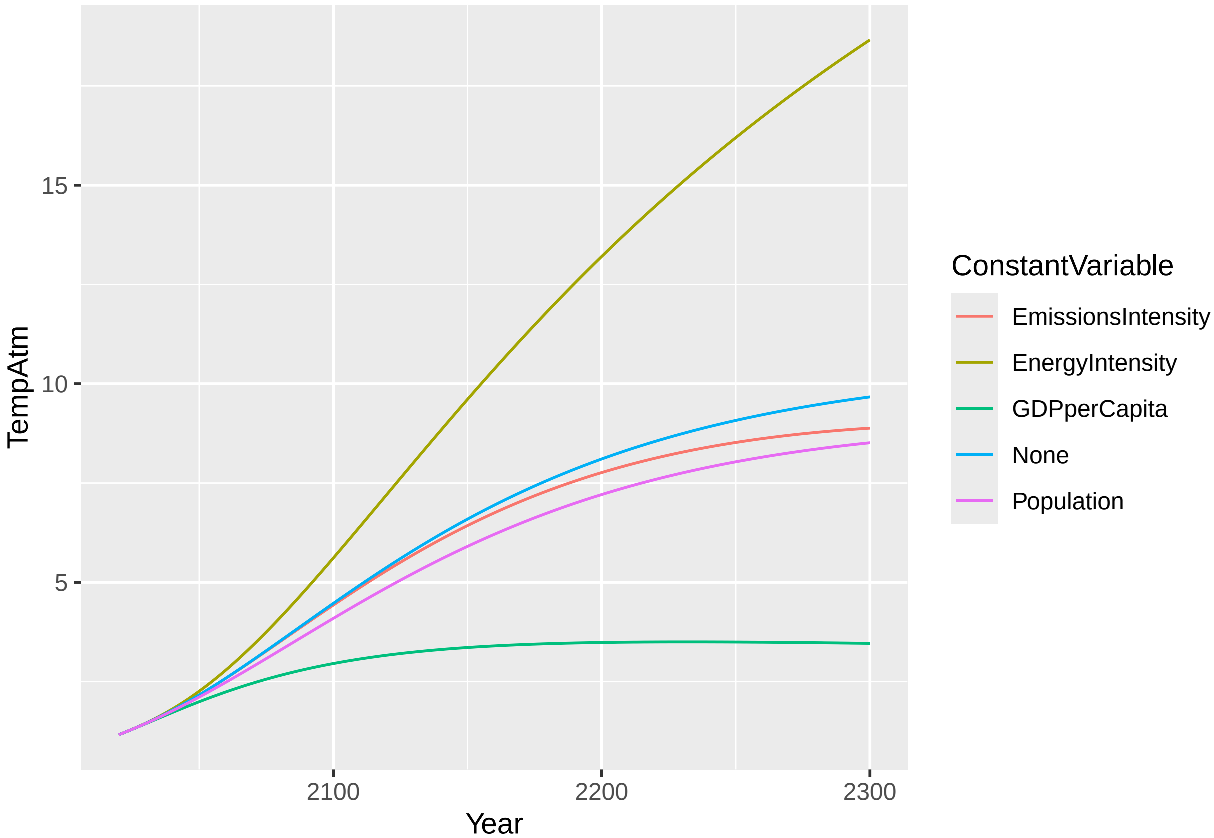
Keep Emissions Intensity constant

```
#Implement Five box model for CO2 concentrations
for (i in 2:length(Kaya.global$Year)) {
 Kaya.global[i, Box1col:(Box1col+4)] <- CO2difference(Kaya.global[i - 1, Box1col:(Box1col+4)],
                                                       Kaya.global$EmissionsEmissionsIntensityConstant[i
 }
# Calculate total CO2 concentration in atmosphere
Kaya.global$CO2conc <- Kaya.global$Box1 + Kaya.global$Box2 + Kaya.global$Box3 +
  Kaya.global$Box4 + Kaya.global$Box5
# Calculate radiative forcing
Kaya.global$RF <- RadForc(Kaya.global$CO2conc)
# Calculate Temperatures
for (i in 2:length(Kaya.global$Year)) {
    temp <- Temps(Kaya.global$TempAtm[i-1], Kaya.global$TempOcean[i-1], Kaya.global$RF[i])
  Kaya.global$TempAtm[i] <- temp["atm"]
  Kaya.global$TempOcean[i] <- temp["ocean"]
}

Exercise13.5 <- Kaya.global %>%
  select(Year, EmissionsEmissionsIntensityConstant, CO2conc, TempAtm) %>%
  rename(Emissions = EmissionsEmissionsIntensityConstant) %>%
  mutate(ConstantVariable = "EmissionsIntensity") %>%
  full_join(Exercise13.5)
```

```
## Joining with `by = join_by(Year, Emissions, CO2conc, TempAtm,
```

```
## ConstantVariable)`
```

```
Exercise13.5 %>% ggplot(aes(x = Year, y = TempAtm, color = ConstantVariable)) +
  geom_line()
```




```
Exercise13.5 %>% filter(Year == 2300) %>%
  select(ConstantVariable, TempAtm) %>%
  arrange(-TempAtm) %>%
  mutate(across(where(is.numeric), \(x) round(x, 2)))
```

```
##     ConstantVariable TempAtm
## 1    EnergyIntensity   18.66
## 2               None    9.67
## 3 EmissionsIntensity    8.88
## 4         Population    8.51
## 5       GDPperCapita    3.46
```