Exercise 13.9 - Tradable Emissions Permits

Aaron Swoboda

2025-05-19

This document performs Exercise 13.9 from the Integrated Assessment Model chapter of Climate Economics. It runs the underlying climate and economic modeling tasks from sections 13.1 and 13.2. It also adds the new emissions reduction rates, $R_{i,t}$ to the model, calculates the relative and absolute cost of the emissions reductions, and explores the impact on carbon dioxide concentrations and warming.

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_hODz6j1DlLiaQvr
```

Load historic emissions data

```
## Auto-refreshing stale OAuth token.

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

## v Range '2:10000000'.

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

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

```
# Graph emissions over time
emissionsSince1750 %>%
```

```r
  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.

```r
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.

```r
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)

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

Temps <- function(atmtempold, oceantempold, radforc) {
  par1 <- 1.15
```

```r
  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
}
```

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

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

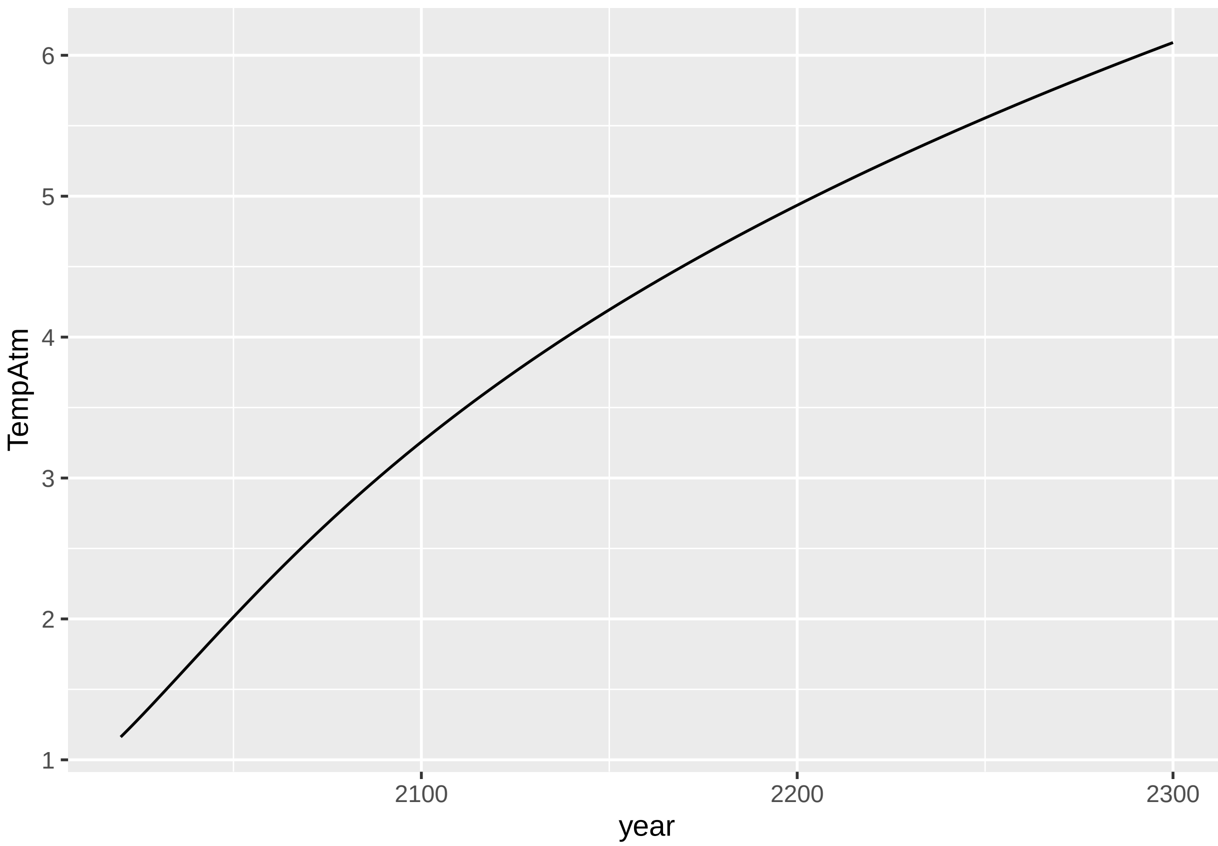
```r
HistoricData %>%
  ggplot(aes(year, TempAtm)) +
  geom_line()
```

Future Scenario Example: Constant Emissions

```r
years <- 2020:2300

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

Scenario1$year <- years
Scenario1$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 +
```

```r
  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)

Kaya Identity

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.

```r
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

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

Using Righ-Middle-Poor Region

Access the regional data

```
# 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 regional labels, etc. in earlier rows
```

```
Rich.df <- regional.df[-(1:3), c(1, 2, 6, 10, 14)]
#head(Rich.df)
Rich.df <- Rich.df %>% mutate(across(Population:Emissions, as.numeric))
#head(Rich.df)
```

```
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))
```

Rich Region Kaya Modeling

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

#head(PopulationModel)
```

```
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)
```

```
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)
```

```
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 + EnergyInten
}
```

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

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

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

```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.

```r
Rich.df$A = 5.5 #
Rich.df$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])

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

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

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 = 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)
```

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.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)`
```

```
#head(Kaya.Rich)
```

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.Rich = Kaya.Rich %>%
  mutate(Emissions = Population*GDPperCapita*EnergyIntensity*EmissionsIntensity)
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
```

Middle Income

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

```
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.df$Population[which(Middle.df$Year == 2019)]-1)
```

```
#head(PopulationModel)
```

```
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)
```

```
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()
```

```
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, ]
```

```
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.

```
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)
```

```
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 == 
```

```
                              GrowthRate =  aveEmissionsIntGrowth)

#EmissionsIntensityModel[1, ]
```

```
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$Populat
Middle.df$GDPmodeled[1] = CobbDouglassGDP(Middle.df$A[1], Middle.df$K[1], Middle.df$Population[1])

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()
```

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)
```

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.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)`
```

```r
#head(Kaya.Middle)
```

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.Middle = Kaya.Middle %>%
  mutate(Emissions = Population*GDPperCapita*EnergyIntensity*EmissionsIntensity)
#head(Kaya.Middle)
```

Lowest Income

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

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

#head(PopulationModel)
```

```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])
}
#head(PopulationModel)
```

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

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

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

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

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

```r
# EnergyIntensityModel[1, ]
```

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

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

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

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

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

# EmissionsIntensityModel[1, ]
```

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

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

GDP

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

```r
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])

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

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

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 = 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)
```

Emissions from Kaya Calculation

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.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)
```

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)
#head(Kaya.Poor)
```

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?

```
Kaya.global <- Kaya.Rich

names(Kaya.global)
```

```
## [1] "Year"                "Population"          "GDPperCapita"
## [4] "EnergyIntensity"     "EmissionsIntensity" "Emissions"
```

```
Kaya.global$Emissions <-  Kaya.global$Emissions + Kaya.Middle$Emissions + Kaya.Poor$Emissions
```

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")
```

Abatement and Tradable Permits

Now we add in the potential for emissions abatement as described by the text in section 13.3 and 13.4

```
Abatement.df <- data.frame(Year = 2020:2300,
                           R.rich = 0.1, R.mid = 0.1, R.poor = 0.1)
```

Pollution reductions reduce output by equation 13.13.

```
GDPModel = PopulationModel %>% left_join(Abatement.df)
```

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

```
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])
}
```

```r
GDPModel$GDPperCapita = GDPModel$GDPmodeled/GDPModel$Population

#tail(GDPModel)
```