Problem Set CarbonReductionCosts

Table of Contents

# Problem Set: Estimating CO2 Reduction Costs

Author: Daniel Dreyer

!! ,fig.width=9, fig.height=6, dev=‘svg’

Welcome to this Interactive Problem Set. The world’s energy demand are constantly increasing and at the same time so are carbon emission. High efforts are made to find technological improvement as well as find economic incentives to avoid long term effects of climate change through high emission. During this problem set, you will examine how carbon pricing would effect emissions in the U.S. electricity sector. The analysis is based on **“Inferring Carbon Abatement Costs in Electricity Markets: A Revealed Preference Approach using the Shale Revolution”** by Joseph A. Cullen, Erin T. Mansur (2016) - further referred as Cullen (2016). You can find the paper and further ressources here: <a href=“<https://www.aeaweb.org/articles?id=10.1257/pol.20150388> target =”\_blank"> https://www.aeaweb.org/articles?id=10.1257/pol.20150388

\_\_\_\_ ergebnisse \_\_\_\_

We use the statistical programming language R to replicate the analyisis proposed by Cullen (2016). Therefore, you are required to have basic knowledge of Statistics and R.

Therefore, I will provide information of concepts and functions along the way, but you are required to have basic knowledge of Statistics and R, but . If you are completely new to programming in R you should’t feel left behind though. Understanding the basic concepts is pretty straight forward, you can find a useful beginners guide [here](https://cran.r-project.org/manuals.html). Below you will find an overview of content and a guideline on how to deal with upcoming interactive problems.

### Contents

The main idea behind … is to find effect of …

Overview

1 TBA

2 TBA

2.1

3 TBA

4 TBA

5 Conclusion

References

## Exercise Content

*Exercise 1* - Motivation (rename later)

*Exercise 2* - TBA

*Exercise 3* - TBA

*Exercise 4* - TBA

*Exercise 5* - TBA

### ReadME

The problem set offers different ways of interaction. Some provide you additional information or test your knowledge, others need you to fill in small gaps of code. Coding exercises are marked as **TASK**. Below you can read find the different types of interactions with their corresponding functions:

* *Info Boxes*: Contain additional information on technical terms or documentation of functions.
* *Quizzes*: Evaluate your knownledge of topics before we dive in our analysis.
* *Code Chunks*: Require you to complete small parts of Code. The work flow of solving code chunks is intuitive and as followed:
* edit : Click to start editing the Code chunk.
* run chunk: Runs the chunk and displays outputs or errors in the corresponding console.
* check: Check your input against the solution.
* hint : You can request a hint if you have problem solving a task.

Additional functions:

* data: Redirects you to the data browser - you can navigate through the data set in an interface.
* solution : Shows the solution of the task.

After finishing a exercise, click Go to next exercise at the bottom or navigate around with the help of the bar on top.

## Exercise 1 – Insights into Energy Markets

The world’s energy consumption keeps rising daily. New technologies are being developed to produce more green electricity, but we are still heavily dependant on fossil fuels. Coal is hereby one of the biggest and most important sources of energy, but at the same time the biggest cause of carbon emissions. It is easy to store, to transport, doesn’t alter and can be mined in huge quantities all around the world. This makes it desirable and cheap to use as a reliable energy source. Given that coal supplies will last for centuries, it is a major goal to find incentitives to reduce its use and therefore emissions. Another major energy ressource is natural gas. Even though still being a fossil fuel, its’ emissions and heat density are way superior to coal. Nevertheless, natural gas wasn’t considered to be an real alternative to coal because of higher costs through difficult extracting methods and handling.  
This changed due to the Shale Revolution, which started in the early 2000s in the United States. Natural gas began to not only be a by-product of the oil industry, but could now be extracted in large quantities and in a targeted manner. Lanfrancois (2012) estimates, that based on the Clean Power Plan introduced by the Obama Administration in 2015, carbon dioxide emissions from the electricity industry could be reduced by roughly 23 to 42 percent by replacing existing coal to gas fired generators.

### Info: Shale-Revolution

Shale gas is a form of natural gas and is extracted from shale formation, a technology known as fracking. Since the start of this century, shale gas became a increasingly more popular form of natural gas in the United States. Whereas in the year 2000 shale gas only provided 1% of extracted natural gas in the U.S., nowadays it makes up roughly 50%. The long term effects on the environment however are highly unceratin and heavily debated.

You can find further information on this topic below:  
<a href=“<https://en.wikipedia.org/wiki/Shale_ga> target =”\_blank“> https://en.wikipedia.org/wiki/Shale\_gass  
<a href=”<https://en.wikipedia.org/wiki/Hydraulic_fracturing> target = "\_blank"> https://en.wikipedia.org/wiki/Hydraulic\_fracturing

Enough background for now, let’s start with some interactive tasks. In Exercise 1 we will get a feeling for the energy sector and explore time-series data of historic prices and generated power.

**Task:** Use the read\_csv command to *load the data set* and store it in a variable called dat. Read\_csv displays the parsed type of each column after you execute the command. This will come handy later on in this exercise. In future exercises we will diable this message.

### Info: read.csv() and write.csv()

The command read\_csv() reads in a **csv file** and stores it into a given name. Csv is a format for data, that uses commas as seperators and periods as decimals. Another version of are **csv2 files**, where semicolons are used for seperators and commas for decimals. We only use csv files in this problem set.

Given the file you want to read it is in the same working directory, you can use the command below:

example <- read\_csv("example.csv")

If the file is saved outside of your working directory you have to differentiate between two possibilities.  
For files that are “above” your current working directory, you have to provide the full path to the file. You can view your working directory with getwd():

example <- read\_csv("C:/Data/example.csv")

For files that are located “below” the working directory it is sufficient to specify the file path starting with your current working directory:

example <- read\_csv("./Data/example.csv")

Csv files can be saved using the write\_csv command:

write\_csv(example, file="example.csv")

For further information, use help(write\_csv).

# ... <- read\_csv("./Data/exercise1.csv")  
dat <- read\_csv("./Data/exercise1.csv")

## Parsed with column specification:  
## cols(  
## year = col\_double(),  
## month = col\_double(),  
## date = col\_character(),  
## generated\_coal = col\_double(),  
## generated\_gas = col\_double(),  
## price\_naturalgas\_USA = col\_double(),  
## price\_naturalgas\_Europe = col\_double(),  
## natural\_gas\_index = col\_double()  
## )

**Task:** The first thing you should do after loading a new data set is looking at the structure. Use the head() command to show the first **eight** rows of the data set named dat.

When not given a second argument, head() displays the first 5 rows on default. Alternatively, the problem set provides a seperate data tab. If you want to take a closer look at the data, especially later on in the problem set, you can click data and get redirected to the Data Explorer Tab, where you can navigate along the data frame and see some basic statistics.

# head(dat, ...)  
head(dat, 8)

## year month date generated\_coal generated\_gas price\_naturalgas\_USA  
## 1 2002 1 1/1/2002 164.3580 49.33555 2.25  
## 2 2002 2 2/1/2002 143.0488 45.06809 2.31  
## 3 2002 3 3/1/2002 151.4856 52.11839 3.03  
## 4 2002 4 4/1/2002 142.3046 50.03635 3.42  
## 5 2002 5 5/1/2002 151.4064 51.18480 3.49  
## 6 2002 6 6/1/2002 164.6677 66.63968 3.22  
## 7 2002 7 7/1/2002 183.1947 84.98841 2.98  
## 8 2002 8 8/1/2002 179.9555 85.59359 3.09  
## price\_naturalgas\_Europe natural\_gas\_index  
## 1 3.06 44.72  
## 2 3.03 45.25  
## 3 2.97 53.62  
## 4 2.81 57.57  
## 5 2.83 58.55  
## 6 2.90 55.64  
## 7 2.95 53.30  
## 8 3.01 54.78

The data frame contains monthly numbers of generated MWh of coal and gas as well as prices of natural gas in Europe and the United States as well as the natural gas index from the year 2002 to the end of 2019. Keep in mind that different commodities are traded in different units. Coal in metric tons (\$/mt) and gas in British thermal units ($/mmBTU). The natural gas index is set to 100 in the year 2010 and changes accordingly. While reading in our current data set I mentioned that it would come handy to know which data types your columns inherit. R requires dates to be stored as type Date, otherwise we will get weird outputs when trying to plot along a time line. The date column is currently stored as type character. Therefore we have to convert the date variable into a date format by using as.Date().

**Task:** Transform the data type with as.Date() and store the transformed variable to the same name.

# ... <- ...(dat$date, format="%m/%d/%Y")  
dat$date <- as.Date(dat$date, format="%m/%d/%Y")

Now that we imported our data set, got a first look on its structure and data, lets start creating our first plot. Basic R is the way to go if you want some quick visualisations. Additionally, R provides several packages to create highly customizable plots. In this problem set we will mostly use the package ggplot2. It is widely considered to be one of the most powerful packages to create plots, but having the downside that you have to get used to the handling. Because of that this proble mset doesn’t require you to create whole plots on your own but requires you to fill in gaps. This way you will learn how the logic behind ggplot2 works and at the same time won’t be frustrated if the output doesn’t meet the requirements.

The first plot we create should tell us more about the development of gas prices along the time frame. As we have mentioned at the beginning of this exercise, the Shale-Revolution started in the beginning of this century. Therefore we expect falling gas prices for the U.S. market. Since commodity prices are split for regions/markets, the European Gas price should differ. Let’s see if these assumptions are true.

**Task:** Use the data frame dat to plot historic prices of gas in Europe and the United States. Just press *check*.

### Info: ggplot2

The package ggplot2 is a powerful data visualization package for R, which is part of the tidyverse environment. Besides basic functions that are also provided by native R, it allows the user to highly modify graphs by **altering**, **adding** or **removing** components.

You are not required to have deeper knowledge about the functionality of ggplot2. However, you can find further documentation [here](https://ggplot2.tidyverse.org/) or type help(ggplot2).

ggplot(data=dat, aes(x=date)) +  
 geom\_line(aes(y=dat$price\_naturalgas\_USA, color="Gas US"), size=1) +  
 geom\_line(aes(y=dat$price\_naturalgas\_Europe, color="Gas Europe"), size=1) +  
 labs(x = "Year", y = "", title = "Natural Gas Prices", subtitle = "Y=$/mmBTU")+  
 scale\_color\_manual(name="Region",values = c("blue","red"))



Looking at the plot our assumptions are confirmed. The spikes in 2005 and 2008 are mostly consequences of the iraq war and the financial crisis. Apart from that, we observe a strict fall of gas prices in the U.S. from just over $12 in 2008 to under $2 in 2016. Since ten the prices remained at roughly the same level. Natural gas prices in Europe have remained at a significantly higher level. One reason being is that fracking isn’t as popular in Europe, therefore a major part of gas in Europe gets imported from Russia which drives prices. On the other hand Europe has a different energy mix. According to the European Environment Agency (EEA) the consumption of gas decreased in average by 1.4% per year since 2005, whereas in the U.S. natural gas reached new all-time highs nearly every year. Because of the lack of data provided by state officials in Europe, the analysis in this problem set will further purely focus on the U.S. market.

Before getting an insight into the generated power, take your first quiz:

Quiz: Do you think that the proportion of gas generated electicity compared to coal generated electricity has increased or decreased over time?

* increased [x]
* decreased [ ]

**Task:** Create a new ggplot2 for electricity generation. Plot the date on the x-Axisand the generated electricity on the y-Axis. We want seperate line for gas generated power as well as coal generated power. Lines are drawn with geom\_line. For reference you can look at the code from the previous graph or conduct the help() function.

#ggplot(data=\_\_\_, aes(x=\_\_\_)) +  
# \_\_\_\_(aes(\_=dat$generated\_coal, color="Coal Generation (TWh)"), size=1, alpha=0.5) +  
# \_\_\_\_(aes(\_=dat$generated\_gas, color="Natural Gas Generation (TWh)"), size=1, alpha=0.5) +  
# labs(x = "Year", y = "", title = "Monthly Generation by Fueltype", subtitle = "Y=Monthly Electricty (TWh)")+  
# scale\_color\_manual(name="",values = c("black","red"))+  
# scale\_y\_continuous(breaks=seq(0,200,25))  
ggplot(data=dat, aes(x=date)) +  
 geom\_line(aes(y=dat$generated\_coal, color="Coal Generation (TWh)"), size=1, alpha=0.5) +  
 geom\_line(aes(y=dat$generated\_gas, color="Natural Gas Generation (TWh)"), size=1, alpha=0.5) +  
 labs(x = "Year", y = "", title = "Monthly Generation by Fueltype", subtitle = "Y=Monthly Electricty (TWh)")+  
 scale\_color\_manual(name="",values = c("black","red"))+  
 scale\_y\_continuous(breaks=seq(0,200,25))



The results are in line with what we should expect. With gas decreasing in price, gas plants increased their share in electricity production over time. Giving you a bit more background, coal-fired plants have lower operating costs than gas-fired generators, but are slow to adjust to fluctating demands and are expensive to start. Gas generators typically fill these gaps. There are two different types of gas-generators, one being peaker plants, which run in high demand hours due to their fast start-up times but high marginal costs. The other type being combined cycle gas turbines (CCGT), having low heat rates (high efficiency of turning fuel into power) and are used to provide baseline power generation throughout the day. Because of these factors we can assume that the two types of fuel generators are “switchable”.  
However, the mechanism called fuel switching is a lot more complex than just about changing fuel prices. Factors like capacity of plants, transmission grid limits (Mansur & White 2012, Davis & Hausman 2015) or firms market power (Bushnell, Mansur & Saravia 2008) also play a role here. We try to include some but not all of them in our analysis later on.  
To wrap this up, we observe severe fluctuations, that occur yearly with a smaller peak in summer and a bigger in the winter season. The fluctuation are largely driven by Residential using air conditioning in summer and space heating in winter (EIA, “Today in Energy”, 2020/3).

### Award: Artist

You created your first plot! You will earn more awards throughout the problem set. After you completed all exercises you will see how many awards you got.

In the next exercise we will go through necessary theory for our main analysis. Click Go to next exercise to continue.

## Exercise 2 – Theory

### Relationship between Fuel and Carbon Prices

### Info: Variable names in this exercise

: Marginal Costs  
: Heat Rate, mmBTU/MWh  
: Cost of burning coal  
: Cost of burning gas  
: Coal Price, $/mmBTU  
: Gas Price, $/mmBTU  
: Carbon content of coal, tons/mmBTU  
: Carbon content of gas, tons/mmBTU  
: Carbon Price , $/ton

As seen in the last exercise there are more factors to marginal costs than just the commodity prices itself. With that in mind we define the marginal costs of fossil-powered plants as a Equation of heat rate and the costs of burning fuel. Variable names of this exercise are explained in the info-box above:

$$\tag{1}MC=HR\cdot(P\_{fuel}+CO\_{2,fuel}\cdot P\_{co2})=HR\cdot C\_{fuel}$$

In Exercise 1 we observed generated electricity of coal and gas and their variance over time. Because of the fact that the cost efficiency of coal is generally better, we have to introduce a incentive to reduce the use of coal in relation to gas in electricity generation. One way to introdude a change in cost efficiency is to propose carbon prices. This is especially viable in this case because coal contains approximately twice as much as natural gas and therefore stronger effects the cost efficiency. Another factor that drives this momentum is that gas generators generally have a more efficient heat rate than coal generators (as metioned at the end of last exercise).  
Combining these two facts of cost and generation efficiency lead to steeper marginal cost for coal generators when carbon prices are introduced. Reflecting this to our upcoming analysis, we don’t observe any carbon prices, but a time-series of fuel prices that are transformed into burning costs with equation (1). To implement the mechanism of fuel change we introduce a coal-gas ratio, which we define as followed:

$$\tag{2}costratio=\frac{C\_{coal}}{C\_{gas}}$$

Combining Equation 1 and 2, we can explain cost ratios as a function of fuel prices, carbon content and carbon prices:

$$\tag{3}costratio=\frac{C\_{coal}}{C\_{gas}}=\frac{P\_{coal}+CO\_{2,coal}\cdot P\_{co2}}{P\_{gas}+CO\_{2,gas}\cdot P\_{co2}}$$

For completeness we introduce values for carbon content and , that will be fixed based on predictions for the year 2025. The EIA reports carbon content and forcasts for fuel prices. We will use these values later on in our analysis:

* Average delivered coal price $2.25/mmBTU and gas prices $5.75/mmBTU (forcast for 2025)
* Carbon content Natural Gas: 117 lbs carbon/MMBTU or 0.0585 tons/MMBTU
* Carbon content Coal: 210.8 lbs carbon/MMBTU or 0.1054 tons/MMBTU

Lets quickly interpret Equation 2: The price ratio will rise for higher costs of burning coal or lower costs of burning gas. To make this even clearer, lets visualize the relationship, which replicates Figure 4 of Cullen.

Source: Cullen (2016)

Panel (a) shows the relationship between fixed costs of coal and gas when carbon prices get introduced. As shown before we observe higher marginal costs for coal and at a certain value of carbon prices, gas becomes more cost efficient. Panel (c) shows the results when transforming fuel prices in price ratios as defined in Equation 2. We can observe the same in absence of carbon prices with fixed coal prices and variable gas prices as seen in panel (b) and (d).

Quiz: Do you think we can create any given price ratio under the assumption that we introduce carbon prices under fixed fuel prices? The answer is basically shown in the graphs above.

* Yes [x]
* No [ ]

The answer is the central idea to our analysis. We can create every variation of cost ratios, either by introducing carbon prices to fixed fuel prices, or without carbon prices by varying costs of coal and gas. We use this variation in the cost ratios observed in our data to understand how emissions change when gas generators become more competitive with coal plants (Cullen 2016).

$$\tag{4}{P\_{co2}}=\frac{costratio\cdot{P\_{gas}}-{P\_{coal}}}{CO\_{2,coal}-costratio\cdot CO\_{2,gas}}$$

We will use this equation in the second part of the analysis (4.2) to transform our fuel costs into carbon prices. Therefore we will be able to predict the impact of carbon taxes on emissions and calculate abatement costs.

Since it is not common to have electricity grid in Europe that are formed like the ones in the United States, I will introduce the concept of interconnection in the following sub-section.

### Interconnections

Source: Cullen (2016)

The American Electric system is made up of three major interconnections, which in turn concist of different balancing authorities (responsible for maintaining the electricity balance within the region). Local electricity grids are hereby connected to form a network, which provides higher stablity and reliability. These interconnections operate mostly independent from each other and exchange little to no electricity. This is a huge difference to the European electricity grid, where grid stability is ensured across borders.

In the graph below you can see the 3 interconnection we include data of in this problem set. I listed a few details for each below: - Eastern Interconnection (EAST): Consists of 36 balancing authorities and extends from the East Coast to the Rocky Mountains. - Western Interconnection (WECC): Involves 37 balancing authorities, which are located in the West of North America. - Electric Reliability Council of Texas (ERCOT): Consists of large parts of Texas

In the first part of this exercise we already hinted that our model will we some form of regression of price ratios against emissions. Since we are essentially observing three different energy markets we should confirm the distribution of our data points. Therefore we create a graph that plots all price ratios within our data set against emissions. We run the model seperately on each interconnection if find a large distribution across interconnections. We will present the data set for the analysis in detail in upcoming exercises. For not it’s enough to get an impression on the distribution.

**Task:** Just press *check*.

### Info: geom\_encircle

Geom\_encircle() is part of the package ggalt which extends functionality of ggplot2. Besides standard functionality for plotting points or lines that is provided by basic ggplot2, it provides a advanced framework for visualising your data. Geom\_encircle() automatically encloses points in a polygon and can be used to visualise differences in groups of data.

Call help(geom\_encircle) to get further information.

//////////// rename data set

dat <- read\_csv("Data/exercise3.csv")  
ggplot(dat,aes(x=coalprice/gasprice,y=co2mass/1000,color=intercn))+  
 geom\_point(alpha=0.2) +  
 geom\_encircle(aes(group=intercn,fill=intercn),alpha=0.3, s\_shape=1) +  
 theme\_bw()+   
 labs(y="",   
 x="Price Ratio",   
 subtitle="y=CO2 Emissions in 1000 tons/day",  
 title="Distribution of data")



As you can the distribution of data point for the Eastern interconnection is way different than the other two interconnections. We could probably find a model that fits ERCOT and WECC, but its unlikely to fit well for EAST. Because of this we will perform our analysis on each interconnection seperately.

### Award: Theorist

You learned about the necessary theory. You are ready to start with the analysis!

In the next chapter we will begin constructing a model that will implement the theory we introduced in this exercise. Click Go to next exercise.

## Exercise 3.1 – Emission Response Curves - A simple Approach

A short recapture: We got an idea about fuel shares in the U.S. energy market and introduced interconnection. Furthermore we introduced the concept of fuel switching and necessary theory behind mapping carbon pricing. In Chapter 3 we will combine these findings and start developing a regression model. #########We will start by proposing a simple linear regression, analyze the statistics and work our way along to a fitting model.

If you are purely interested in the final model and the economic impact, you can skip to Exercise 4.1. Chapter 3 will focus on regression theory, which will guide you to the final model.

**Task:** To get started, load the data set exercise3.csv, press edit and check afterwards.

dat <- read\_csv("Data/exercise3.csv")  
head(dat,3)

## date month year intercn co2mass gasprice coalprice load  
## 1 1/1/2006 1 2006 EAST 4548660.4 9.34131 2.05989 6454708.0  
## 2 1/1/2006 1 2006 ERCOT 452107.9 7.58753 1.66738 616973.4  
## 3 1/1/2006 1 2006 WECC 775122.7 7.81944 1.62992 1587397.1

The data are gathered from several official U.S. agencies and aggregated to a daily level and seperated by interconnection. Along the way we will extend the data set with further variables. For now you can find the variables and it’s associated units below:

co2mass: emissions in tons  
gasprice: Capacity weighted average daily gas price per interconnetion ($ per million BTU)  
coalprice: Price of Coal ($ per million BTU)  
load: Ddaily electricity consumption per interconnection in MWh

Because of the size of the original data sets we wont do the data preparation in this problem set but rather provide the data sources. Emission data are measured by the Continuous Emissions Monitoring System (CEMS) of the Environmental Protection Agency (EPA). The U.S. Energy Information Administration (EIA) collects data of coal prices in Form 923. Spot prices for gas can be found at the Intercontinental Exchange (ICE) and data for electricity consumption or load are provided by the Federal Energy Regulatory Commission (FERC) in Form 714.

In Exercise 2 we saw that it makes sense for us to run our model on each interconnection seperatly. For this purpose we will filter our data set for interconnection EAST and develop the model based on these data. Once we found a fitting model we will run it on each interconnection. Before we start with modelling we alter our data set with the implications from above. For ease of interpretation we convert emissions and load from tons to million tons.

**Task:** Use the pipe operator %>% to combine following tasks: Filter the data set dat for interconnection EAST and store the new data frame in variable dat\_east. Additionally, calculate the cost ratio between coalprice and gasprice according to Equation 2.

### Info: Pipe, Select, Filter, Mutate

The pipe operator %>% is a feature provided by the dplyr Package. Essentially it allows to exectute multiple operation on a dataframe at once. Essentially, every pipe operator returns a dataframe and passes it to the next connected function.  
Select allows you to filter for certain columns of data, whereas filter does the same for rows.  
If you want to create new columns or alter existing, you can use mutate. There are several more functions to alter your data sets, below you find a handy cheat sheet.

example %>%   
 select(1:5) %>% # keep certain columns from index 1 to 5   
 mutate(new\_column = old\_column+1) %>% # create or alter columns   
 filter(columnname=="value")

You can find a more detailed cheat sheet [here](https://rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf) or use the help() function.

# \_\_\_ <- data \_\_\_   
# mutate(costratio = \_\_\_/\_\_\_,  
# co2mass = co2mass/1000000),  
# load = load/1000000) %>%  
# filter(intercn=="\_\_\_")  
dat\_east <- dat %>%   
 mutate(costratio = coalprice/gasprice,  
 co2mass = co2mass/1000000,  
 load = load/1000000) %>%  
 filter(intercn=="EAST")

### Stage 1: Linear Regression

We start by proposing a simple linear regression model. In a mathematical way we can express the relationship as followed:

### Info: Linear Regression with lm()

lm() is part of the stats package and is part of basic R. As shown in the syntax example below, it enables you to regress y on the indepedent variables x1 and x2. The model itself can be handled just like every other variable. You can give it a name or use it in other functions. Another popular function to solve linear regressions in R is felm(), that provides further functionality to include fixed effects. In this problem set we will strictly use lm().

example <- lm(y~x1+x2, data=dat\_east)

You can find more information on the lm() function [here](https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/lm) or use the help() function.

**Task:** Run a regression with co2mass as the dependent variable and costratio as independent variable. Store it in the variable fit1.

# ... <- lm(...~... , data=dat\_east)  
fit <- lm(co2mass ~ costratio, data=dat\_east)

**Task:** To show the statistics of our regression we will use stargazer(). Since we won’t need every information that is shown by the default function. We will define our own and at the same time introduce custom functions which will be useful later on. Read the Info Boxes to find out more about stargazer and custom functions and continue with the next task. Press **check** to define the functiom.

### Info: Stargazer

stargazer provides specialised HTML formatting for regression tables and summary statistics tables. It is easy to use, supports a large number of model types and formats data in a more pleasing way. The basic function is called by stargazer() and but can be customized heavily. For more information run help(stargazer) or read on [here](https://cran.r-project.org/web/packages/stargazer/).

### Info: Custom functions

R provides an easy framework to add your own functions. The syntax is quite similar to other programming languages you could be familiar with. As long as you are in the same session you can then use this function. Below you find the basic structure and an example:

`myfunction <- function(arg1, arg2, ... ){`   
`statements`   
`return(object)`

Here’s an simple functions that adds 5 to the input and returns it.

`example <- function(x) {`   
 `x + 5`   
`}`   
`example(1)`

You find more detailed information [here](https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/function).

show.regression= function(...){  
 library(stargazer)  
 stargazer(...,   
 type = "text",   
 style = "aer",   
 digits = 3,  
 df = FALSE,  
 report = "vct\*",  
 star.cutoffs = c(0.05, 0.01, 0.001),  
 model.names = FALSE,  
 object.names = TRUE,  
 model.numbers = FALSE,   
 omit.stat=c("f", "ser")  
 )  
}

Quiz: Before looking at the summary, what impact do changing cost ratios have on emissions?

* With increasing cost ratios emissions will increase. [ ]
* With increasing cost ratios emissions will fall. [x]

**Task:** Use the function we just defined to show the summary of fit1.

show.regression(fit)

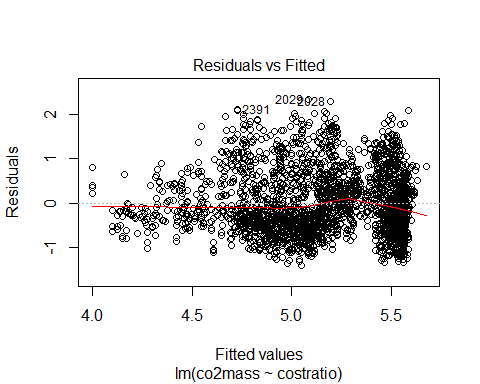
##   
## =====================================================  
## co2mass   
## fit   
## -----------------------------------------------------  
## costratio -1.306   
## t = -25.065\*\*\*   
##   
## Constant 5.877   
## t = 184.852\*\*\*   
##   
## Observations 2,557   
## R2 0.197   
## Adjusted R2 0.197   
## -----------------------------------------------------  
## Notes: \*\*\*Significant at the 0.1 percent level.  
## \*\*Significant at the 1 percent level.   
## \*Significant at the 5 percent level.

We observe a positive intercept term with 5.87704 and a negative coefficient for of -1.30554, meaning that emissions fall with an increasing cost ratio. The three stars next to our coefficents imply, that the p-value is smaller than 0.1. That suggests that our data our inconsistent with the null hypothesis and that the null hyphithesis may be rejected. In other words the probabiliy of finding an estimator that is at least as high as the one we predicted is smaller than one percent.

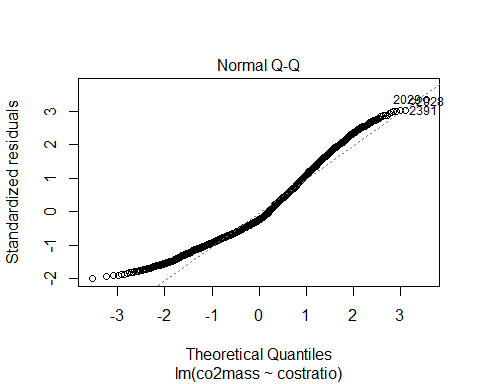
Our eventual goal is to plot the rensponse curves of changing ??AWD prices

**Task:** To

plot(fit, 1)



plot(fit, 2)



QQ PLOT RESIDUALS?

### Info: RMSE?

asd

### Stage 2: Multivariate Regression Model

If we look back at the implications of the energy market from Exercise 1, we know that there are several other factors that should influence emissions. To control for other effects we can use multivariate regression models and include additional factors. For now we add the effects of generated electricity. Our regression formula changes to following:

**Task:** Just press **check**

fit1 = lm(co2mass ~ costratio + load, data=dat\_east)  
  
show.regression(fit,fit1)

##   
## ======================================================  
## co2mass   
## fit fit1   
## ------------------------------------------------------  
## costratio -1.306 -0.751   
## t = -25.065\*\*\* t = -46.157\*\*\*   
##   
## load 0.774   
## t = 157.897\*\*\*   
##   
## Constant 5.877 -0.196   
## t = 184.852\*\*\* t = -4.942\*\*\*   
##   
## Observations 2,557 2,557   
## R2 0.197 0.925   
## Adjusted R2 0.197 0.925   
## ------------------------------------------------------  
## Notes: \*\*\*Significant at the 0.1 percent level.   
## \*\*Significant at the 1 percent level.   
## \*Significant at the 5 percent level.

TEXT

### Info: R-squared

R-Squared is a statistical measurement that represents the correlation between fitted values and observed values. R-squared is hereby always positive and ranges from 0 to 1. A value closer to 1 indicates that the suggested model explains a majority of the variance in the outcome variable. Mathematically we can descibe this with the following:

A problem with the R-squared measurement is, that it always increases with higher numbers of variables in the model, even if these variables are only weakly responsible for the predicted values. An solution is to take the number of variables into account, this is called Adjusted R-Squared and also shown in the summary output.

Our goal is to find the response curves of emissions to changing fuel price. Therefore, we want to plot our fitted values of emissions against costs. First off, we need to predict the emission value of our model.

**Task:** Predict the fitted values of fit2 on dat\_east with help of the predict() function.

### Info: Predict

predict()

co2.hat1 = predict(fit1, dat\_east)

As explained before our eventual goal is to plot the response curves of emissions to changing fuel prices (or cost ratios). Simply evaluating the fitted values on our intitial data frame as we have done in the last task won’t give us an interpretable curve though. A multivariate regression with two dependent variables spans three dimensions. To illustrate this problem run the code below.

**Task:** Just press **check**

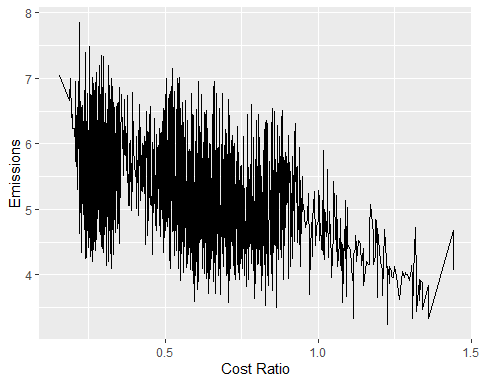
#take sample from data to limit points shown in plot  
temp<- dat\_east %>% cbind(co2.hat1)  
temp1<- temp[sample(nrow(temp), 150), ]  
#assign vairables to axises  
x <- temp1$costratio  
z <- temp1$co2.hat1  
y <- temp1$load  
#set a nice color scheme  
numbercol <- 8 # number of colors  
plotcolor <- brewer.pal(numbercol,"Greys") # use Brewer to get color scheme   
colornum <- cut(rank(z), numbercol, labels=FALSE)  
colorcode <- plotcolor[colornum] # assign color  
#plot  
s3d <- scatterplot3d(x,y,z, col.axis="gray", col.grid="gray", type="h",color=colorcode,   
angle = 35, scale.y = 1, pch = 19, xlab="Cost Ratio", ylab = "Load", zlab = "Emissions")  
s3d$plane3d(fit1, lty = "dotted")



As you can see we can a regression plane that spans over three dimensions and there no exact “line” to be drawn here. For purpose of illustration, run the code below to see what would happen if we simply plot emissions against cost ratios.

**Task:** Just press **check**

ggplot(dat\_east,aes(x=costratio,y=co2.hat1))+  
 geom\_line(aes(x=dat\_east$costratio, y=co2.hat1))+  
 labs(y="Emissions", x="Cost Ratio")



R hereby follows it’s intential function and connects all points, which is not the desired output we want to see. We could follow the interpretation of falling emission for higher cost ratios but a more detailed analysis isn’t possible. To solve this issue we will use a work around in the way that we won’t predict our values on the original data we build our model on. Instead we will evaluate all independent variables but cost ratios at their respective mean.  
We are able to do so, because the average of the fitted values is equal to the average of the actual values :

This is true for linear regressions with intercept term. The sum of the residuals are in this case zero.

To implement what we have just adopted we create a new data frame. Since we just added load to our model for now, let’s create a new dataframe with costratio and the mean value of load. This will serve as our data frame we use to run predict() on.

**Task:** Just press **check**.

pred\_dat = tibble(costratio = dat\_east$costratio) %>%  
 cbind(load=mean(dat\_east$load))

Until now we have assumed that we are dealing with a simple linear relationship. This is highly unlikely in a real-world scenario.

**Task:** Just press **check**.

fit2 = lm(co2mass ~ poly(costratio,3) + load, data=dat\_east)  
show.regression(fit, fit1, fit2)

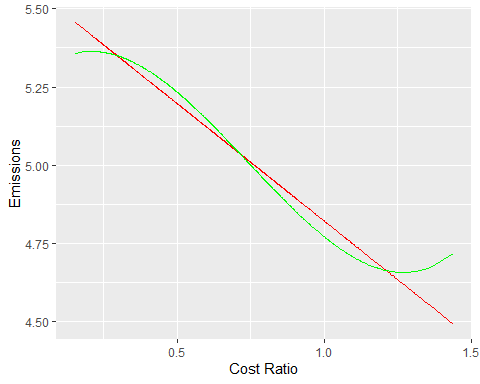
##   
## ================================================================  
## co2mass   
## fit fit1 fit2   
## ----------------------------------------------------------------  
## costratio -1.306 -0.751   
## t = -25.065\*\*\* t = -46.157\*\*\*   
##   
## poly(costratio, 3)1 -10.059   
## t = -46.615\*\*\*  
##   
## poly(costratio, 3)2 -0.287   
## t = -1.360   
##   
## poly(costratio, 3)3 1.444   
## t = 6.852\*\*\*   
##   
## load 0.774 0.773   
## t = 157.897\*\*\* t = 158.893\*\*\*  
##   
## Constant 5.877 -0.196 -0.602   
## t = 184.852\*\*\* t = -4.942\*\*\* t = -16.495\*\*\*  
##   
## Observations 2,557 2,557 2,557   
## R2 0.197 0.925 0.927   
## Adjusted R2 0.197 0.925 0.927   
## ----------------------------------------------------------------  
## Notes: \*\*\*Significant at the 0.1 percent level.   
## \*\*Significant at the 1 percent level.   
## \*Significant at the 5 percent level.

**Task:** Run predict() on fit1 and fit2. Use the data frame pred\_dat to calculate the fitted values.

#co2.hat1 = \_\_\_  
#co2.hat2 = \_\_\_  
co2.hat1 = predict(fit1, pred\_dat)  
co2.hat2 = predict(fit2, pred\_dat)

**Task:** Create a new ggplot. Use geom\_line() to plot one line each for co2.hat1 and co2.hat2, with cost ratios on the x-axis and emissions on the y-axis.

#ggplot(pred\_dat,aes(x=costratio,y=dat\_east$co2mass))+  
# geom\_line(aes(\_\_\_), color="red") +  
# geom\_line(aes(\_\_\_), color="green") +  
ggplot(pred\_dat,aes(x=costratio,y=dat\_east$co2mass))+  
 geom\_line(aes(x=costratio, y=co2.hat1), color="red") +  
 geom\_line(aes(x=costratio, y=co2.hat2), color="green") +  
 labs(x="Cost Ratio", y="Emissions")



As we can see in the graph above our output shows us nice lines for every fit that we could interpret.

### Award: Linear Regression Expert2

TEXT

## Exercise 3.2 – Restricted Spline Regressions

### Stage 3: Cubic Spline

In this exercise we will introduce the concept of spline regression. We

In mathematical terms a spline regression can be noted the following way:

$$\tag{4}CO\_{2t}=s(CR\_{t}|\beta)$$

Methodically there doesn’t change much to normal polynomial regressions

### Info: ns()

Ns() is part of the splines package and allows us to perform natural splines regressions on our data. It takes several input arguments, the only one we will use is df which sets the degrees of freedom.

WIP

### Award: Linear Regression Expert

TEXT

## Exercise 4 – Examining Carbon Abatement

In the following chapter we shift our focus away from theory and work our way towards our final results. To approach our analysis we split it into two parts:

* First, we will use the methods we introduced to determine the response curves to changing fuel costs. (Exercise 4.1)
* Second, we transform these fuel prices into carbon prices and plot the abatement curves. (Exercise 4.2)

To this point we limited our data to fuel prices and generated electricity (load). In this exercise we will present the final data set and explain the additional effects we will include in our model. ?????????

**Task:** Load the data set main\_data.csv into a variable dat and show the first 3 rows.

#... <- read\_csv("./Data/...")  
#...  
dat <- read\_csv("./Data/main\_data.csv")  
head(dat,3)

## date month year intercn co2mass coalprice gasprice costratio  
## 1 1/1/2006 1 2006 EAST 4548660.4 2.05989 9.34131 0.22051  
## 2 1/1/2006 1 2006 ERCOT 452107.9 1.66738 7.58753 0.21975  
## 3 1/1/2006 1 2006 WECC 775122.7 1.62992 7.81944 0.20844  
## load tlsd tlmin tlmax nonfossil meant netNSflow  
## 1 6454708.0 18396.854 241122.34 302880.47 75239808 41.66000 2618123  
## 2 616973.4 2818.439 21748.79 30423.37 4278971 63.46470 0  
## 3 1587397.1 6752.520 56996.01 78514.23 26915852 46.78953 -1170264  
## so2price  
## 1 1513.86  
## 2 1513.86  
## 3 1513.86

The data frame is a combination of data we used previous exercises and additional factors. It consists of daily information for each interconnection from 2006 to 2012. The additional variables we add here are explained below:

tlsd, tlmin, tlmax: Standard deviation, minimum and maximum of load. meant: Average daily temperature per interconnection  
nonFossil: Electricity generation with non-fossil fuel in MWh  
so2price: Permit prices of ($/ton)  
netNSflow: Electricity flowing from Canada to US by interconnection and month in MWh

The data are gathered from several official U.S. agencies and continue to be aggregated to a daily level and seperated by interconnection. The U.S. Energy Information Administration (EIA) provides data for Non-fossil energy production, permit prices for are collected from the EPA Clean Air Markets and finally, net imports of electricity from Canada are gathered from the National Energy Board of Canada. Links to the sources can be found in the References section.

In Exercise 1, we saw heavy seasonal influences for generated power, especially we observed peaks in summer and winter. To adjust for this effect we create a seasonal dummy variable that has three states for seasons (off-season/summer/winter) and takes the year in consideration.

**Task:** Calculate the seaonal variable and show the first few rows. Just press *check*.

dat\_final <- dat %>%   
 mutate(season=(month>3) + (month>6) + (month>9),  
 yearseason=year\*10+season)  
  
head(dat\_final,3)

## date month year intercn co2mass coalprice gasprice costratio  
## 1 1/1/2006 1 2006 EAST 4548660.4 2.05989 9.34131 0.22051  
## 2 1/1/2006 1 2006 ERCOT 452107.9 1.66738 7.58753 0.21975  
## 3 1/1/2006 1 2006 WECC 775122.7 1.62992 7.81944 0.20844  
## load tlsd tlmin tlmax nonfossil meant netNSflow  
## 1 6454708.0 18396.854 241122.34 302880.47 75239808 41.66000 2618123  
## 2 616973.4 2818.439 21748.79 30423.37 4278971 63.46470 0  
## 3 1587397.1 6752.520 56996.01 78514.23 26915852 46.78953 -1170264  
## so2price season yearseason  
## 1 1513.86 0 20060  
## 2 1513.86 0 20060  
## 3 1513.86 0 20060

**Task:** Use group\_by() to group the dataset dat by intercn. Afterwards use summarise\_all() to calculate the means of every column. For formatting we use the package kable. Just press **checK**.

### Info: Group\_by() and summarise()

Group\_by() allows you to group a data frame by specific variables. Operations that are run on the grouped data frame are then performed on each group.

The function Summarise() runs on **grouped data** and can perform operations e.g. calculating means (mean()) or finding minimums (min()). There are several pre-implemented version of summarise functions in R. The one we use here is summarise\_all(), which performs these operations on all columns.

To give you an example in code form, lets pretend we have a dataframe data with several car manufacturer and their respective car models with prices and we want to calculate the average car price per manufacturer.

example <- data %>%   
 group\_by(manufacturer) %>%  
 summarise(mean\_price = mean(price))

Call help(group\_by) or help(summarize) for further information.

### Info: kable()

hallo hier entsteht eine info box

**Task:** Just press **check**.

dat\_final %>%   
 select(intercn, co2mass, load, coalprice, gasprice, costratio, nonfossil) %>%   
 group\_by(intercn) %>%   
 mutate(co2mass = co2mass/1000, load=load/1000, "Emission Rate"=co2mass/load, nonfossil=nonfossil/100000) %>%   
 summarise\_all("mean") %>%   
 kable(format="html", align="c", col.names = c("intercn","Emissions","Load","Gas Price","Coal Price","Price Ratio","Non Fossil","Emission Rate")) %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"), position = "center", full\_width = F)

intercn

Emissions

Load

Gas Price

Coal Price

Price Ratio

Non Fossil

Emission Rate

EAST

5159.0120

7456.084

2.499007

5.487308

0.5499861

704.27030

0.6901824

ERCOT

561.4024

866.491

2.204418

5.104193

0.5243940

50.71451

0.6502459

WECC

888.6862

1834.824

1.841604

5.042629

0.4303330

249.70453

0.4838995

The table reports the mean of important variables for each interconnection. We can clearly see that EAST is by far the largest of the observed energy markets. The emissions seem to increase proportionally to the electricity consumption if we take the electricity production from non-fossil sources in consideration. This gets clearer if the look at the emission rates, which are defined as . Therefore the Western interconnection has by far the greenest energy production, followed with some distance by Texas (ERCOT) and the Eastern interconnection. Furthermore we observe clear variations in price ratios. This is in line with our assumptions from Section 2, where we stated that each interconnection has vastly different conditions and therefore should conduct our regression on each seperatly. In the next exercise we will use this data set and the regression theory we presented in Chapter 3 to trace out the emission response curves.

### Award: ?

wip

Click Go to next exercise to continue.

## Exercise 4.1 – Estimated CO2 Response to Fuel Prices

In this exercise we expand the regression model from the previous Exercise and determine the response curves to changing fuel prices. In the first step we will perform a regresson based on the methods we introduced in the previous chapter. Afterwards we will use this model to predict emissions and plot a response curve that gives us percentage changes based on predicted fuel prices of 2025.

As a side note, the code isn’t meant to be the shortest or the most efficient, but should allow you to follow each step we take to get our final results. Based on this we will explain and carry out the analysis step by step for interconnection EAST, afterwards apply it to ERCOT and WECC and interpret our results.

Building upon the model we defined in the last chapter, we will expand our regression model with several control variables as followed. The meaning and source of data to every variable is explained in the info box below:

First, we load and prepare the data to run the ///////// SEASONAL VARIABLE

**Task:** Load the dataframe and store it in dat. create the seasonable dummy variable.

dat <- read\_csv("Data/main\_data.csv") %>%   
 mutate(season=(month>3) + (month>6) + (month>9),  
 yearseason=year\*10+season)  
head(dat,3)

## date month year intercn co2mass coalprice gasprice costratio  
## 1 1/1/2006 1 2006 EAST 4548660.4 2.05989 9.34131 0.22051  
## 2 1/1/2006 1 2006 ERCOT 452107.9 1.66738 7.58753 0.21975  
## 3 1/1/2006 1 2006 WECC 775122.7 1.62992 7.81944 0.20844  
## load tlsd tlmin tlmax nonfossil meant netNSflow  
## 1 6454708.0 18396.854 241122.34 302880.47 75239808 41.66000 2618123  
## 2 616973.4 2818.439 21748.79 30423.37 4278971 63.46470 0  
## 3 1587397.1 6752.520 56996.01 78514.23 26915852 46.78953 -1170264  
## so2price season yearseason  
## 1 1513.86 0 20060  
## 2 1513.86 0 20060  
## 3 1513.86 0 20060

**Task:** Filter the data set for intercn EAST.

#east <- filter(...,...)  
east <- filter(dat, intercn=="EAST")

**Task:** Calculate the mean of every variable we use in our regression. First, select the necessary columns and then use summarise\_all to get the mean. Store the result in mean\_east.

# ... <- east %>%   
# select(load, tlsd, tlmin, tlmax, meant, nonfossil, so2price, netNSflow, yearseason) %>%   
# summarise\_all(...)  
mean\_east <- east %>%   
 select(load, tlsd, tlmin, tlmax, meant, nonfossil, so2price, netNSflow, yearseason) %>%   
 summarise\_all(mean)

**Task:** To make our life a bit easier we will create a new data set with gasprice, priceratio and the results of the last task. Use tibble(), which creates a new data frame and cbind(), which takes a sequence of columns and combines them with another data frame.

#temp\_east <- ...(gasprice = east$gasprice, priceratio = east$priceratio) %>%   
# ...(mean\_east)  
predict\_east <- tibble(date=east$date, intercn=east$intercn, co2mass=east$co2mass, gasprice = east$gasprice, coalprice=east$coalprice, costratio = east$costratio) %>%   
 cbind(mean\_east)

#We will perform a reduced-form regression which expands the cubic spline regression from exercise 3.2 and builds upon the theory of exercise 2. We define the model as followed: ??????????

$$\tag{5}CO\_{2t}=s(priceratio\_{t}|\beta)+s(load\_{t}|\theta) + s(temp\_t|\omega)+X\_t\psi+D\_\gamma+\epsilon\_t$$

### Info: Model variables

We already introduced the meaning of some variables in previous exercises. To get one complete view I listed them below again: ????????? : CO2 emissions in tons  
: Cost ratio of coal over gas : daily electricity consumption per interconnection in MWh  
: average daily temperature per interconnection  
: Factors like non-fossil electricity production (e.g. solar, hydro or wind), price, net imports of electricity from Canada and variance in load  
: Dummy variable for seasonal variation to absorb fluctuations, e.g. by renewable energies.

**Task:** Perform the regression model we described in the beginning of this exercise for interconnection EAST. Use ns() for variables with spline regression (as explained in 3.3). We want to use 5 degrees of freedom. Store the resulting model in reg\_east.

#... <- lm(co2mass ~ ...(priceratio, df=...) + ...(load, df=...) + tlsd + tlmin + tlmax + ...(meant, df=...) + nonfossil + so2price + netNSflow + yearseason, data=east)  
reg\_east <- lm(co2mass ~ ns(costratio, df=5) + ns(load, df=5) + tlsd + tlmin + tlmax + ns(meant, df=5) + nonfossil + so2price + netNSflow + yearseason, data=east)

**Task:** Predict emissions based on the regression model reg\_east and temp\_east. We set interval to confidence to get the mean interval and be able to plot a confidence band later on. Just press *check*.

### Info: Confidence interval

A confidence interval answers the question for which defined probability the data points lie within the interval. Mathematically, given we have observations and a confidence level , a confidence interval has a probability to contain the true underlying parameter. Most commonly, and also in our case, we use the 95% confidence interval and is defined as:

where is the fitted response, the t-value with n-2 degrees of freedom and the equation inside the square root represents the standard error.

/// reg\_robust <- lm\_robust(y1 ~ x, mydat, se\_type = ‘stata’) /// <https://ditraglia.com/econ224/lab07.pdf>

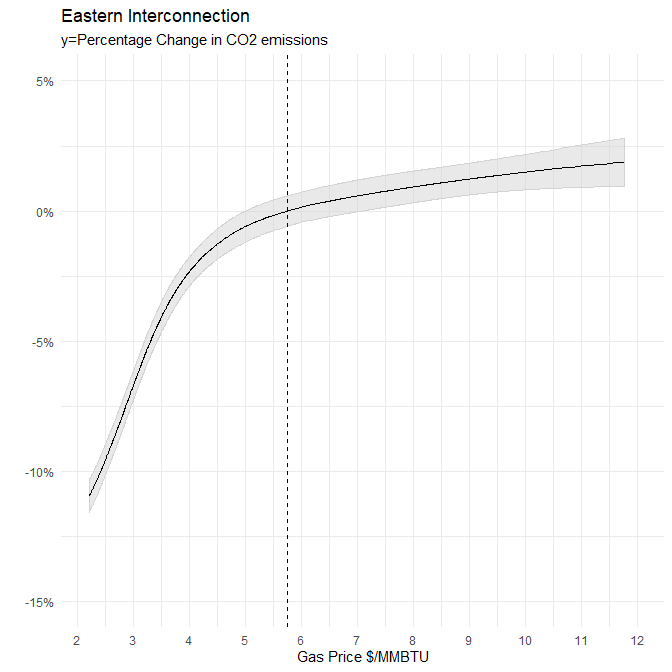
fit\_east <- predict\_east %>%   
 cbind(as.data.frame(predict(reg\_east, newdata = predict\_east, interval = 'confidence')))

**Task:** Just press *check*.

basecoal=2.25  
basegas=5.75  
  
diff\_base <- abs(east$costratio - (basecoal/basegas))  
min\_dif <- min(diff\_base)  
closest <- min\_dif==diff\_base  
base\_emit <- mean(fit\_east$fit[which(closest)])  
  
final\_east <- fit\_east %>%   
 mutate(fit.transformed=fit/base\_emit-1,  
 lwr.transformed=lwr/base\_emit-1,  
 upr.transformed=upr/base\_emit-1)

**Task:** Just press *check*.

plot\_east <- ggplot(final\_east, aes(x=basecoal/costratio, y=fit.transformed)) +  
 geom\_ribbon(aes\_string(ymin = final\_east$lwr.transformed, ymax = final\_east$upr.transformed),  
 colour="lightgrey", fill="lightgrey", alpha=0.5,) +  
 geom\_line(aes(y=fit.transformed)) +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1), limits=c(-.15,.05)) +  
 geom\_vline(xintercept = 5.75, linetype = "dashed") +   
 scale\_x\_continuous(breaks=c(2,3,4,5,6,7,8,9,10,11,12), limits=c(2.2, 12)) +  
 labs(title="Eastern Interconnection", subtitle = "y=Percentage Change in CO2 emissions",  
 y="", x="Gas Price $/MMBTU") +  
 theme\_minimal()  
  
plot\_east



TEXT////

**Task:** Run the code for the ERCOT interconnection. Just press *check*.

#filter for interconnection ERCOT  
ercot <- filter(dat, intercn=="ERCOT")  
  
#calculate the mean for regression coefficients  
mean\_ercot <- ercot %>%   
 select(load, tlsd, tlmin, tlmax, meant, nonfossil, so2price, netNSflow, yearseason) %>%   
 summarise\_all(mean)  
  
#join data  
predict\_ercot <- tibble(date=ercot$date, intercn=ercot$intercn, gasprice = ercot$gasprice, coalprice=ercot$coalprice, priceratio = ercot$priceratio) %>% cbind(mean\_ercot)

## All columns in a tibble must be 1d or 2d objects:  
## \* Column `priceratio` is NULL

#run model   
reg\_ercot <- lm(co2mass ~ ns(priceratio, df=5) + ns(load, df=5) + tlsd + tlmin + tlmax + ns(meant, df=5) + nonfossil + so2price + netNSflow + yearseason, data=ercot)

## Error in ns(priceratio, df = 5): object 'priceratio' not found

#predict new co2 emission and join with predict\_ercot, ERCOT has no net imports of electricity, predict throws warning  
fit\_ercot <- predict\_ercot %>%   
 cbind(as.data.frame(predict(reg\_ercot, newdata = predict\_ercot, interval = 'confidence')))

## Error in eval(lhs, parent, parent): object 'predict\_ercot' not found

#find baselevel of emission in relation to baseline price of coal and gas  
diff\_base <- abs(ercot$priceratio - (basecoal/basegas))  
min\_dif <- min(diff\_base)  
closest <- min\_dif==diff\_base  
base\_emit <- mean(fit\_ercot$fit[which(closest)])

## Error in mean(fit\_ercot$fit[which(closest)]): object 'fit\_ercot' not found

#transform predicted emission into percentage change to baseline emissions  
final\_ercot <- fit\_ercot %>%   
 mutate(fit.transformed=fit/base\_emit-1,  
 lwr.transformed=lwr/base\_emit-1,  
 upr.transformed=upr/base\_emit-1)

## Error in eval(lhs, parent, parent): object 'fit\_ercot' not found

#plot result  
plot\_ercot <- ggplot(final\_ercot, aes(x=basecoal/priceratio, y=fit.transformed)) +  
 geom\_ribbon(aes\_string(ymin = final\_ercot$lwr.transformed, ymax = final\_ercot$upr.transformed),  
 colour="lightgrey", fill="lightgrey", alpha=0.5) +  
 geom\_line(aes(y=fit.transformed)) +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1), limits=c(-.15,.05)) +  
 geom\_vline(xintercept = 5.75, linetype = "dashed") +   
 scale\_x\_continuous(breaks=c(2,3,4,5,6,7,8,9,10,11,12), limits=c(2.2, 12)) +  
 labs(title="ERCOT Interconnection", subtitle = "y=Percentage Change in CO2 emissions",  
 y="", x="Gas Price $/MMBTU") +  
 theme\_minimal()

## Error in ggplot(final\_ercot, aes(x = basecoal/priceratio, y = fit.transformed)): object 'final\_ercot' not found

**Task:** Run the code for the WECC interconnection. Just press *check*.

wecc <- filter(dat, intercn=="WECC")  
  
mean\_wecc <- wecc %>%   
 select(load, tlsd, tlmin, tlmax, meant, nonfossil, so2price, netNSflow, yearseason) %>%   
 summarise\_all(mean)  
  
predict\_wecc <- tibble(date=wecc$date, intercn=wecc$intercn, gasprice = wecc$gasprice, coalprice=wecc$coalprice, priceratio = wecc$priceratio) %>% cbind(mean\_wecc)

## All columns in a tibble must be 1d or 2d objects:  
## \* Column `priceratio` is NULL

reg\_wecc <- lm(co2mass ~ ns(priceratio, df=5) + ns(load, df=5) + tlsd + tlmin + tlmax + ns(meant, df=5) + nonfossil + so2price + netNSflow + yearseason, data=wecc)

## Error in ns(priceratio, df = 5): object 'priceratio' not found

fit\_wecc <- predict\_wecc %>%   
 cbind(as.data.frame(predict(reg\_wecc, newdata = predict\_wecc, interval = 'confidence')))

## Error in eval(lhs, parent, parent): object 'predict\_wecc' not found

base\_emit <- mean(fit\_wecc$fit[which(min(abs(wecc$priceratio - (basecoal/basegas)))==abs(wecc$priceratio - (basecoal/basegas)))])

## Error in mean(fit\_wecc$fit[which(min(abs(wecc$priceratio - (basecoal/basegas))) == : object 'fit\_wecc' not found

final\_wecc <- fit\_wecc %>%   
 mutate(fit.transformed=fit/base\_emit-1,  
 lwr.transformed=lwr/base\_emit-1,  
 upr.transformed=upr/base\_emit-1)

## Error in eval(lhs, parent, parent): object 'fit\_wecc' not found

plot\_wecc <- ggplot(final\_wecc, aes(x=basecoal/priceratio, y=fit.transformed)) +  
 geom\_ribbon(aes\_string(ymin = final\_wecc$lwr.transformed, ymax = final\_wecc$upr.transformed),  
 colour="lightgrey", fill="lightgrey", alpha=0.5) +  
 geom\_line(aes(y=fit.transformed)) +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1), limits=c(-.15,.05)) +  
 geom\_vline(xintercept = 5.75, linetype = "dashed") +   
 scale\_x\_continuous(breaks=c(2,3,4,5,6,7,8,9,10,11,12), limits=c(2.2, 12)) +  
 labs(title="Western Interconnection", subtitle = "y=Percentage Change in CO2 emissions",  
 y="", x="Gas Price $/MMBTU") +  
 theme\_minimal()

## Error in ggplot(final\_wecc, aes(x = basecoal/priceratio, y = fit.transformed)): object 'final\_wecc' not found

**Task:** Use grid.arrange() to display the plots next to each other. Just press *check*.

grid.arrange(plot\_east, plot\_ercot, plot\_wecc, nrow=1)

## Error in arrangeGrob(...): object 'plot\_ercot' not found

//TEXT, east oben, hier verlgeich zu ercot, wecc

///////// test

main\_data2 <- rbind(final\_east, final\_ercot, final\_wecc)

## Error in rbind(final\_east, final\_ercot, final\_wecc): object 'final\_ercot' not found

main\_data2 <- main\_data2[order(as.Date(main\_data2$date, format="%d/%m/%Y")),]

## Error in eval(expr, envir, enclos): object 'main\_data2' not found

write.csv(main\_data2, "X:\\libraries\\RTutor\_BA\\main\_data2.csv", row.names = FALSE)

## Error in is.data.frame(x): object 'main\_data2' not found

temp <- read\_csv("main\_data2.csv")

## Error: 'main\_data2.csv' does not exist in current working directory ('X:/libraries/RTutor\_BA').

### Award: Sp(l)ine Surgeon

This was a critical operation, but you mastered it! Keep going, the hardest part is over.

In the next exercise we will map the emission response curves from this exercise into carbon prices and estimate the effects of carbon prices on emissions as well as the associated costs. Click Go to next exercise to continue.

## Exercise 4.2 – Imputed CO2 Response to Carbon Prices

During exercise 4.1, we found that emissions could be greatly reduced with changing cost ratios of coal and gas. In this exercise we will follow the analysis of Cullen (2016) and transform these ratios into carbon prices. This will allow us to estimate the effect of carbon taxes on emissions and we can approximate the costs of such reductions. The exercise is split in two logical parts, first we will transform the ratios with the help of Equation 3. Afterwards we will plot the emission response curves on carbon taxes and interpret the results. ?????????????

To get the same percentage change as in 4.1 we assume the same base line prices. Additionally the fuels associated carbon contents:

* Average delivered coal price $2.25/mmBTU and gas prices $5.75/mmBTU for 2025.
* Carbon content Natural Gas: 117 lbs carbon/MMBTU or 0.0585 tons/MMBTU
* Carbon content Coal: 210.8 lbs carbon/MMBTU or 0.1054 tons/MMBTU (averaged on weighted fuel consumption according to EIA Form 923)

To transform the response curves we will use the same fitted model as in of Exercise 4.1. To avoid repeating the same steps as before, I prepared a data set that includes the predicted emissions as well as the tranformed emissions with confidence intervals. Load the data set main\_data2.csv and store it in dat. Use the head() command to show the first rows. To save us some time, we also declare the variables for base prices and carbon content.

**Task:** Just press *check*.

dat <- read\_csv("Data/main\_data2.csv")  
head(dat,3)

## date intercn gasprice coalprice priceratio fit lwr  
## 1 1/1/2006 EAST 9.34131 2.05989 0.22051 5103749.0 5068459.4  
## 2 1/1/2006 ERCOT 7.58753 1.66738 0.21975 587032.3 581503.9  
## 3 1/1/2006 WECC 7.81944 1.62992 0.20844 880353.8 873111.1  
## upr fit.transformed lwr.transformed upr.transformed  
## 1 5139038.6 0.01544 0.00842 0.02246  
## 2 592560.7 0.01015 0.00063 0.01966  
## 3 887596.6 0.00741 -0.00088 0.01569

basegas <- 5.75  
basecoal <- 2.25  
gas\_cc <- 0.0585  
coal\_cc <- 0.1054

In the next step we go ahead and transform the cost ratios into carbon prices using Equation 3. For reference I included it below once again. Based on the cost ratios we calculate we will also get negative carbon ratios. Since we don’t consider negative taxes we will filter them out.

As we have all needed data now, we can calculate the carbon taxes in the next step. We will get negative tax values because of our method, therefore we will filter them out. It would’t make much sense for us here to consider negative taxes. For reference or when you decided to skip to this exercise I once again include the Equation 3.

$$\tag{3}{P\_{co2}}=\frac{CR\cdot{P\_{gas}}-{P\_{coal}}}{CO\_{2,coal}-CR\cdot CO\_{2,gas}}$$

**Task:** Create a column carbontax using Equation 3 and filter the just calculated taxes for the interval [0,80]. Save the result into data frame tax and display the first rows.

#... <- dat %>%   
# mutate(carbontax = (priceratio\*...-...)/(coal\_cc-priceratio\*gas\_cc)) %>%   
# filter(carbontax >= ... & carbontax <= ...)  
#  
#head(...,...)  
tax <- dat %>%   
 mutate(carbontax = (priceratio\*basegas-basecoal)/(coal\_cc-priceratio\*gas\_cc)) %>%   
 filter(carbontax >= 0 & carbontax <= 80)  
  
head(tax,3)

## date intercn gasprice coalprice priceratio fit lwr  
## 1 9/15/2006 EAST 4.35576 1.95501 0.44883 4996909.2 4966759  
## 2 9/15/2006 WECC 3.76465 1.51456 0.40231 873742.4 867020  
## 3 9/16/2006 EAST 4.35576 1.95501 0.44883 4996909.2 4966759  
## upr fit.transformed lwr.transformed upr.transformed carbontax  
## 1 5027059.0 -0.00582 -0.01181 0.00018 4.17970  
## 2 880464.8 -0.00016 -0.00785 0.00753 0.77307  
## 3 5027059.0 -0.00582 -0.01181 0.00018 4.17970

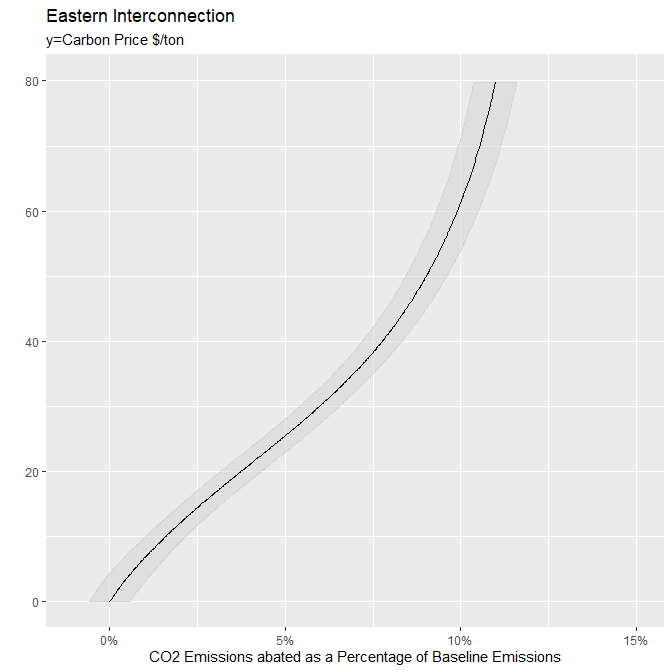
The overall approach is similar to the one in exercise 4.1. We will filter for each interconnection seperatly and plot our result with ggplot2. First we will create a plot for interconnection EAST and interpret the results. Afterwards we will run the code on the other interconnections and compare them to another. In contrast to exercise 4.1 though, we won’t plot the emission change, but the abated emissions.

**Task:** Filter the data frame tax for interconnection EAST and save it to tax\_east. Since we filter for a string, don’t forget to put quotes around the argument.

#  
tax\_east <- filter(tax, intercn=="EAST")

**Task:** Just press *check*.

plot\_east <- ggplot(tax\_east, aes(x=carbontax, y=-fit.transformed)) +  
 geom\_ribbon(aes\_string(ymin = -tax\_east$lwr.transformed, ymax = -tax\_east$upr.transformed),  
 colour="lightgrey", fill="lightgrey", alpha=0.5) +  
 geom\_line(aes(y=-fit.transformed)) +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1), limits=c(-.01,0.15)) +  
 scale\_x\_continuous(limits=c(0, 80)) +  
 labs(title="Eastern Interconnection",  
 y="CO2 Emissions abated as a Percentage of Baseline Emissions", x="", subtitle="y=Carbon Price $/ton") +  
 coord\_flip()  
  
plot\_east



///////TEXT

**Task:** Run the code for the ERCOT interconnection. Just press *check*.

tax\_ercot <- filter(tax, intercn=="ERCOT")  
  
plot\_ercot <- ggplot(tax\_ercot, aes(x=carbontax, y=-fit.transformed)) +  
 geom\_ribbon(aes\_string(ymin = -tax\_ercot$lwr.transformed, ymax = -tax\_ercot$upr.transformed),  
 colour="lightgrey", fill="lightgrey", alpha=0.5) +  
 geom\_line(aes(y=-fit.transformed)) +   
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1), limits=c(-.01,0.15)) +  
 scale\_x\_continuous(limits=c(0, 80)) +  
 labs(title="Ercot Interconnection",  
 y="CO2 Emissions abated as a Percentage of Baseline Emissions", x="", subtitle="y=Carbon Price $/ton") +   
 coord\_flip()

**Task:** Run the code for the WECC interconnection. Just press *check*.

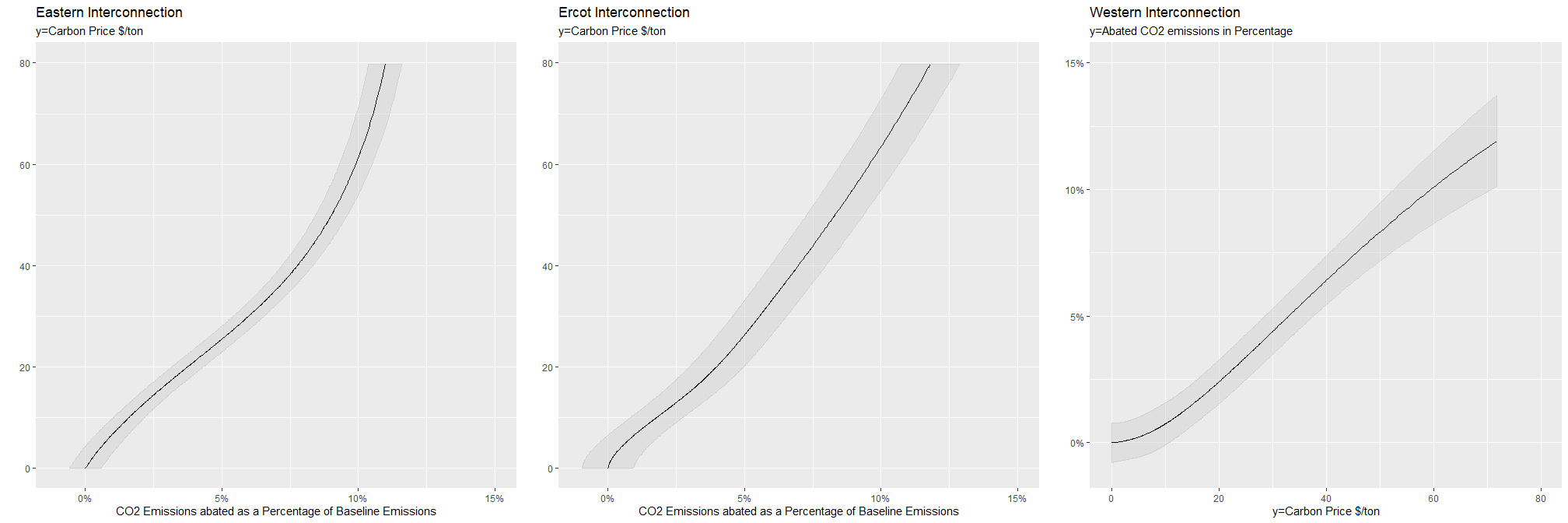
tax\_wecc <- filter(tax, intercn=="WECC")  
  
plot\_wecc <- ggplot(tax\_wecc, aes(x=carbontax, y=-fit.transformed)) +  
 geom\_ribbon(aes\_string(ymin = -tax\_wecc$lwr.transformed, ymax = -tax\_wecc$upr.transformed),  
 colour="lightgrey", fill="lightgrey", alpha=0.5) +  
 geom\_line(aes(y=-fit.transformed)) +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1), limits=c(-.01,0.15)) +  
 scale\_x\_continuous(limits=c(0, 80)) +  
 labs(title="Western Interconnection",  
 y="", x="y=Carbon Price $/ton", subtitle="y=Abated CO2 emissions in Percentage") #+   
 #coord\_flip()

Quiz: We have seen in an earlier exercise, that interconnection ERCOT has higher generation of renewable energies than EAST. Do you expect ERCOT to have a higher or lower emission abatement in comparison to EAST?

* higher [ ]
* lower [x]

**Task:** Just press *check*.

grid.arrange(plot\_east, plot\_ercot, plot\_wecc, nrow=1)



TEXT

### Award: 2?

?

**Task:** Just press *check*.

find\_closest\_value <- function(a) {  
 vector <- 1:9  
 for(i in 0:8){  
 vector[i+1] = mean(a$fit[which.min(abs(a$carbontax - i\*10))])  
 }  
 return(vector)  
}  
temp1 = data.frame(tax=seq(0,80,10),east\_emission=round(find\_closest\_value(tax\_east)/100000,1), ercot\_emission=round(find\_closest\_value(tax\_ercot)/100000,1), wecc\_emission=round(find\_closest\_value(tax\_wecc)/100000,1))  
  
temp2 <- temp1 %>%   
 mutate(perc\_east=round((1-temp1$east\_emission/temp1$east\_emission[1])\*100,1),  
 perc\_ercot=round((1-temp1$ercot\_emission/temp1$ercot\_emission[1])\*100,1),  
 perc\_wecc=round((1-temp1$wecc\_emission/temp1$wecc\_emission[1])\*100,1),  
 emission\_all=east\_emission+ercot\_emission+wecc\_emission)  
  
temp3 <- temp2 %>%   
 mutate(perc\_all=round((1-temp2$emission\_all/temp2$emission[1])\*100,1))  
   
table <- temp3 %>%  
 select(c(1,2,5,3,6,4,7,8,9)) %>%  
 kable(format="html", col.names = c("Tax","abs.","%","abs.","%","abs.","%","abs.","%"), align="c", caption = "Precited Emissions and Percentage Abatement") %>%  
 add\_header\_above(c(" "=1, "East" = 2, "ERCOT" = 2, "West" = 2, "Total" = 2)) %>%  
 kable\_styling(bootstrap\_options = c("striped", "hover", "condensed"), position = "center", full\_width = F)%>%  
 #column\_spec(1:9, width = "0.4") %>%  
 footnote(general = "Predicted emission are in 100.000 tons/day. Change to baseline (Tax=0).")  
  
table

Precited Emissions and Percentage Abatement

East

ERCOT

West

Total

Tax

abs.

%

abs.

%

abs.

%

abs.

%

0

50.3

0.0

5.8

0.0

8.7

0.0

64.8

0.0

10

49.5

1.6

5.7

1.7

8.7

0.0

63.9

1.4

20

48.4

3.8

5.6

3.4

8.5

2.3

62.5

3.5

30

47.2

6.2

5.5

5.2

8.4

3.4

61.1

5.7

40

46.4

7.8

5.4

6.9

8.2

5.7

60.0

7.4

50

45.7

9.1

5.3

8.6

8.0

8.0

59.0

9.0

60

45.3

9.9

5.3

8.6

7.9

9.2

58.5

9.7

70

45.0

10.5

5.2

10.3

7.7

11.5

57.9

10.6

80

44.7

11.1

5.1

12.1

7.7

11.5

57.5

11.3

Note:

Predicted emission are in 100.000 tons/day. Change to baseline (Tax=0).

The result corresponds to Table 2 in Cullen(2016).

TEXT

testvector <- tax\_east$carbontax  
  
test <- tax\_east %>%   
 mutate(fx1=carbontax,  
 fx2=carbontax[+1])  
  
  
  
#gen fx1=carbontax  
# gen fx2 = carbontax[\_n+1]  
# gen x1 = base\_emit - emithat //the height of the curve when the base emission are the zero line  
# gen x2 = base\_emit -emithat[\_n+1]   
#gen area = 0.5\*(x2-x1)\*(fx2+fx1)

To conclude our analysis we can calculate the abatement costs for certain carbon taxes. Lets say we introduce a carbon tax of 40$/ton

Calculating costs

Concluding, our results seem to be robust to parameter changes.

## Exercise 5 – Conclusion

awards()

## Hi , you have earned 0 awards:

The methods used here can be further used to generate additoinal results. This could be a point where you can go on.

## Exercise 6 – Bonus - Robutness Tests

//// einschub robustness test mit anderen definition von priceratios , appendix a2< number of knots NUR FUER EAST, danach kurz interpretieren

# show that all integers between 0 and 10  
1:10

## [1] 1 2 3 4 5 6 7 8 9 10

# show that all integers between 0 and 10  
1:10

## [1] 1 2 3 4 5 6 7 8 9 10

## Exercise References

### Bibliography

* Bushnell, J. B., Mansur, E. T. & Saravia, C. (2008), ‘Vertical arrangements, market structure, and competition: An analysis of restructured US electricity markets’, American Economic Review 98(1), 237–66.
* Cullen, Joseph A., and Erin T. Mansur. 2017. “Inferring Carbon Abatement Costs in Electricity Markets: A Revealed Preference Approach Using the Shale Revolution.” American Economic Journal: Economic Policy, 9 (3): 106-33.
* Davis, L. & Hausman, C. (2015), Market impacts of a nuclear power plant closure, Ei @ haas working paper wp-248.
* Lafrancois, B. A. (2012), ‘A lot left over: Reducing CO2 emissions in the United States’ electric power sector through the use of natural gas’, Energy Policy 50, 428–435.
* Mansur, E. T. & White, M. (2012), ‘Market organization and efficiency in electricity markets’, (Working Paper).
* Swensson, Wretman (1992), “Model Assisted Survey Sampling”, Springer-Verlag.

### R Packages

* Auguie, B. (2017): gridExtra: “Functions in Grid graphics”, R package version 2.3, <http://cran.r-project.org/web/packages/gridExtra/index.html>
* Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.2. <https://CRAN.R-project.org/package=stargazer>
* Kranz, S. (2020): RTutor. “Creating R problem sets with automatic assessment of student’s solutions”, R package version 2020.03.13, <https://github.com/skranz/RTutor>
* Ligges, U., Maechler, M., Schnackenberg, S. (2018), scatterplot3d: “Plots a three dimensional (3D) point cloud.”, R package version 0.3-41, <https://cran.r-project.org/web/packages/scatterplot3d/index.html>
* Neuwirth, E. (2014), RColorBrewer: “Provides color schemes for maps (and other graphics)”. R package version 1.1-2, <https://cran.r-project.org/web/packages/RColorBrewer/index.html>
* Rudis, B. (2017): ggalt: “Extra Coordinate Systems, ‘Geoms’, Statistical Transformations, Scales and Fonts for ‘ggplot2’”, R package version 0.4.0, <https://cran.r-project.org/web/packages/ggalt/index.html>
* Venables ,W., Bates, D. (2019), splines: “Regression spline functions and classes.”, R package version 3.6.2, Part of R.3.6.2
* Wickham, H. (2016): ggplot2. “Elegant Graphics for Data Analysis”, Springer-Verlag, New York, R package version 3.2.1, <http://CRAN.R-project.org/package=ggplot2>
* Wickham, H., Francois, R., Henry, L., Muller, K., (2018): dplyr. “A Grammar of Data Manipulation”, R package version 0.8.3, <http://CRAN.R-project.org/package=dplyr>
* Zhu, H. (2019), kableExtra: “Construct Complex Table with ‘kable’ and Pipe Syntax”, R package version 1.1.0, <https://cran.r-project.org/web/packages/kableExtra/index.html>

### Data Sources

* European Environment Agency, “Primary Energy Consumption by Fuel”, <https://www.eea.europa.eu/data-and-maps/indicators/primary-energy-consumption-by-fuel-6/assessment-2>
* Federal Energy Regulatory Comission, “Form No. 714 Annual Electric Balancing Authority Area and Planning Area Report”, <https://www.ferc.gov/docs-filing/forms/form-714/data.asp>
* Government of Canada, “Canada Energy Regulator”, <https://www.cer-rec.gc.ca/bts/ctrg/gnnb/lctrctxprts/index-eng.html>
* Intercontinental Exchange, “Commodity Prices”, <https://www.theice.com/marketdata/reports>
* National Centers for environmental Information (NOAA), Climate data, <https://www.ncdc.noaa.gov/cag/statewide/time-series>
* The World Bank , “Commodity Markets Monthly Prices”, <https://www.worldbank.org/en/research/commodity-markets>
* United States Environmental Protection Agency, “SO2 Trading Program”, <https://ampd.epa.gov/ampd/>
* U.S. Energy Information Administration, “FAQ”, (<https://www.eia.gov/tools/faqs/faq.php?id=73&t=11>)
* U.S. Energy Information Administration, “Form EIA-923”, <https://www.eia.gov/electricity/data/eia923/>
* U.S. Energy Information Administration, “Today in Energy”, <https://www.eia.gov/todayinenergy/detail.php?id=43035>

*All of the above links were accessable as of March 31, 2020.*