



Data Assignment 1

The [Canadian Energy Regulator](#) produces a semi-annual-ish report called [Canada's Energy Future](#) which provides scenario-based analysis of global and local energy issues and the effects of these trends and issues on Canada's Energy Economy.

For your first data assignment, I'm going to ask you to produce a few graphs based on publicly-available data from the [2023 Canada's Energy Future Report](#) and answer some questions on those graphs. This assignment tests your skills in filtering, grouping, and presenting data in table and graph form. It also asks you to draw some basic conclusions from the data and to use a graph to support your conclusions in some way.

For what you see below, I've used the following packages so use this as a guide to set up your document:

```
library(kableExtra)
library(readxl)
library(janitor)
library(tidyverse)
library(lubridate)
library(scales)
library(viridis)
```

There are **multiple ways** for you to access the [data](#) for this report, but I'd **recommend** going directly to the [Open Government portal](#), and specifically to the [Benchmark Prices](#) data set to complete the main deliverables for this assignment. You'll need to figure out how to access the data you need and how to read it successfully into R.

There are **five (5) graded deliverables** in this assignment.

I am going to ask you for **three (3) specific R outputs** in this assignment, each of which you can complete using coding techniques that you've already seen, with some minor modifications. Each of these is worth 2 points with a grading key as follows : 0 for not attempted, 1 for attempted with reasonable effort but not completed, 1.5 for satisfactory completion, 2 for excellent work.

Next, in addition to these specific visualization deliverables, which you can complete using code that we've already used during the term, I'm going to ask you to answer a question on the data and to support your answer with a graph of your choosing. Here, again, I'll grade you on the scale of 0 for not attempted, 1 for attempted with reasonable effort but not completed, 1.5 for a solid explanation with no errors, and 2 for compelling explanation of the data. And, for the graph, I'll use the same scale as above.

Deliverable 1

The first deliverable for this assignment is a simple table, but based on filtered data. I would like you filter your prices data to have Western Canadian Select (WCS) prices for the *Canada Net-Zero* scenario, and only for years that are multiples of 5 (2005, 2010, 2015, ..., 2045, 2050) **starting** from 2020.

Western Canadian Select (WCS) prices in the Canada Net-Zero scenario of the Canada Energy Regulator's Canada's Energy Future (2023) report	
Year	Price (\$US 2022/bbl)
2020	29.61
2025	62.33
2030	49.00
2035	48.00
2040	47.00
2045	46.00
2050	45.00

I'll give you a couple of hints for how to make this happen:

1. So far, we've been using xls or xlsx files. Depending on how you access the data, you may need a `read_csv` or `read.csv` command if you're using a CSV data file. Also, make sure you don't give the file a different extension when you download it (e.g. don't set your `destfile="foo.xlsx"` if you're downloading a CSV file. Use `destfile="foo.csv"`);
2. If you want to force a number to have two decimal places, you can use `format(value, nsmall = 2)`, but remember that it will create a character string not a numerical value once you do that. Test it out: type `format(20.2565, nsmall = 2)` and you'll get back "20.26", a character string;
3. I don't want you to get caught up on styling your tables, so here are the three lines of code that I used to create the table itself above:

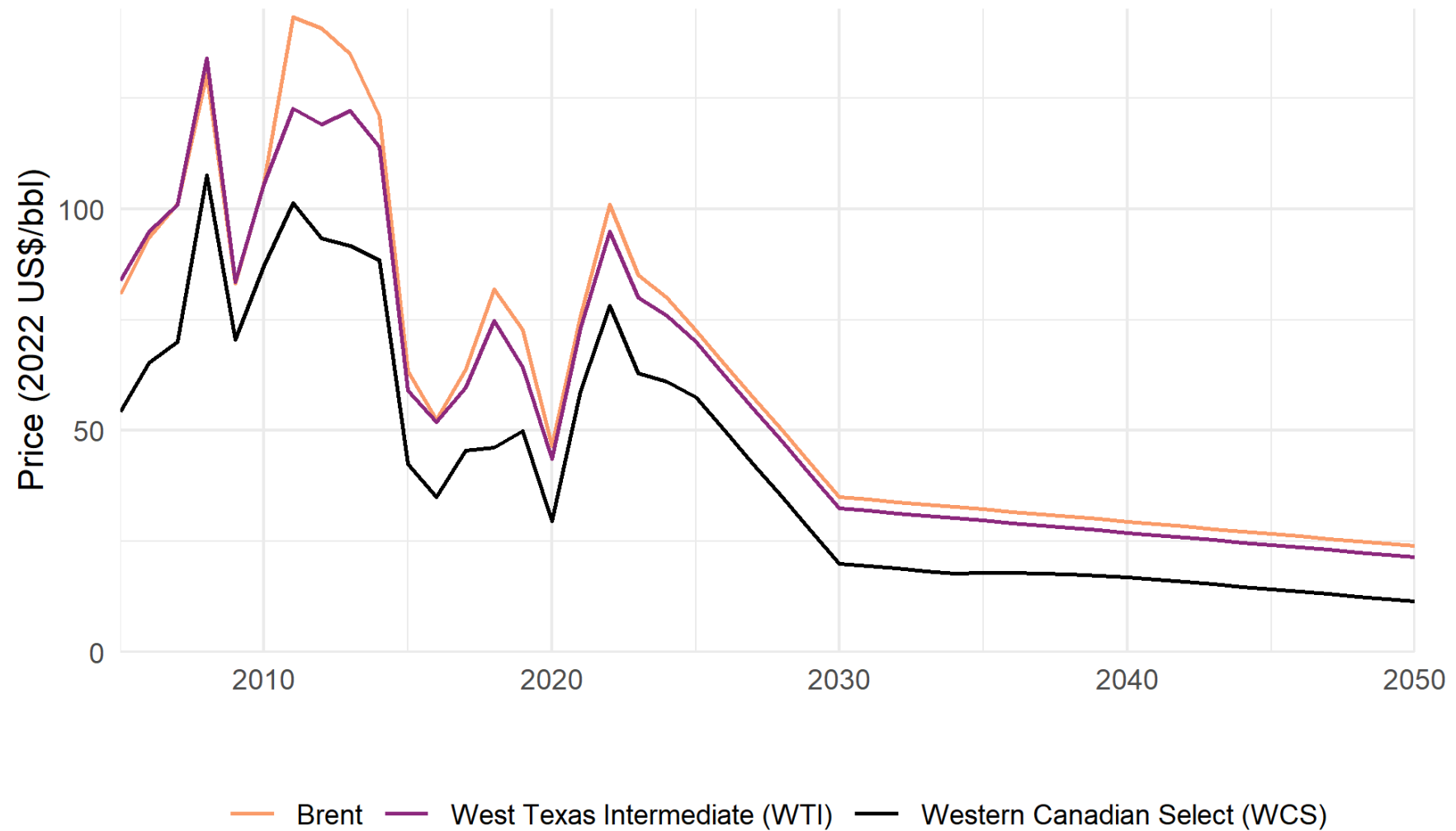
```
kbl(escape = FALSE, table.attr = "style='width:80%;'", digits=2, align=rep('c', 2)) %>%  
  kable_styling(fixed_thead = T, bootstrap_options = c("hover", "condensed", "responsive"), full_width = T) %>%  
  add_header_above(header = c("Western Canada Select (WCS) prices in the Canada Net-Zero scenario of the Canadian Energy Regulator"))
```

4. R will calculate a modulo (remainder) using `%%`, so, if you were to enter something like `2025%%5`, R would return 0, which you can use to design your filtering for years that divide evenly by 5.

Deliverable 2

The second deliverable for this assignment is a graph of oil prices, which you'll again have to base on filtered data. I'd like you to graph Brent, WTI, and WCS prices for the *Global Net-zero* scenario over time.

Crude Oil Prices, Global Net-zero Scenario



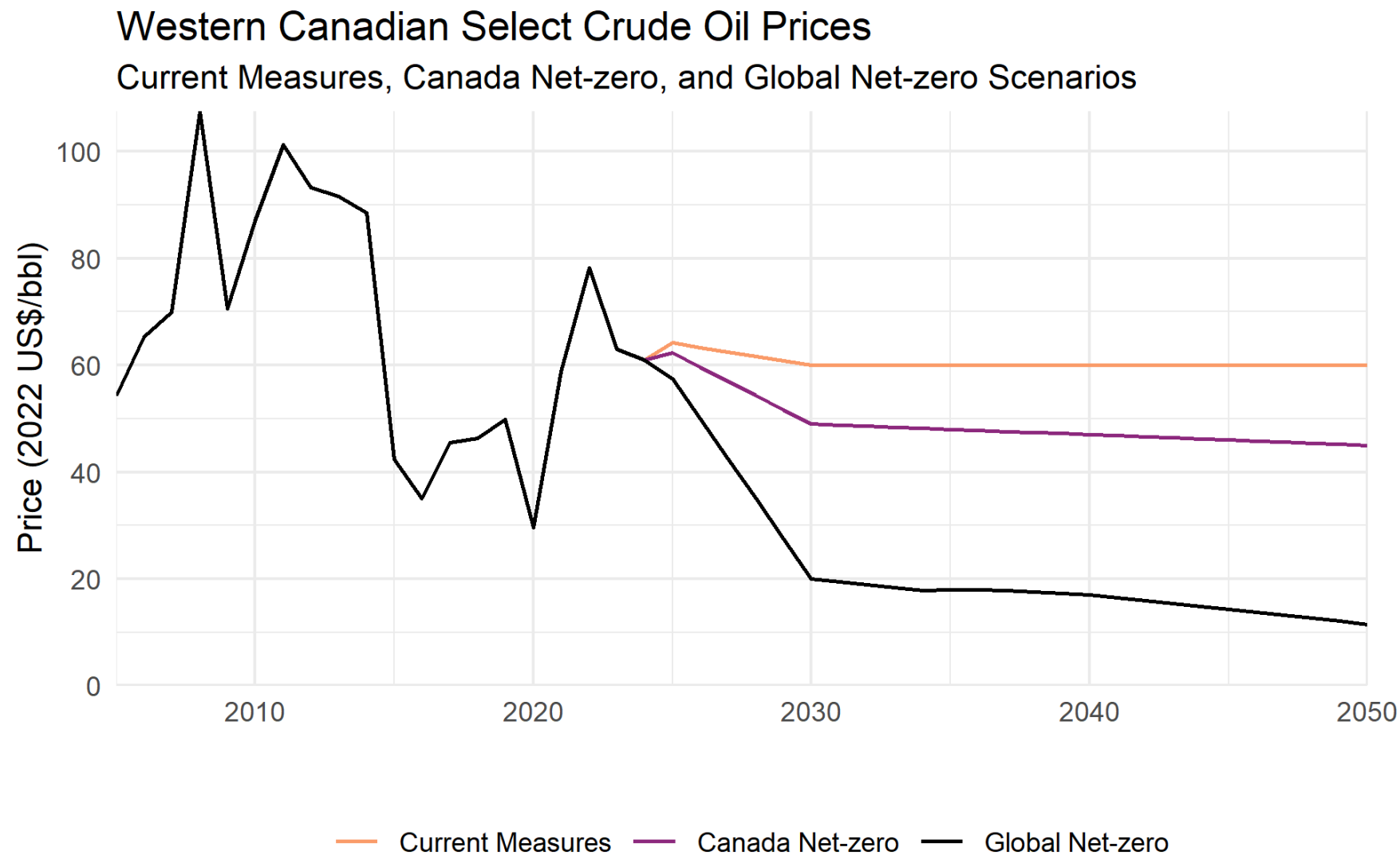
Data via Canada Energy Regulator, Energy Futures (2023), graph by Andrew Leach

I'll give you a couple of hints for how to make this happen:

1. I used `gsub` to strip the 2022 US \$/bbl from each of the variable names, to clean up the graph a bit. You can see a little bit of help for how to do that [here](#). I used `variable= gsub(" - 2022 US\\$/bbl","",variable)` to make that happen. Dollar signs are generally used as indicators for math in markdown, so if you're using a dollar sign in your text, you need to lead it with a `\` like this: `\$`. In `gsub`, you need to lead symbols with `\\` to tell it to look for the specific symbol, otherwise it will think you're building what's called a *regular expression*;
2. I used `expand_limits(y=c(0,145))` to make my axes look nice, and `scale_color_viridis("",option = "A",discrete=T,begin = 0,end = .8,direction=-1)` for the color palette. I've also used `theme(legend.position = "bottom")` for the position of the legend;
3. You will likely need to change the margins around your plot so that things aren't getting cut off. Use `theme(plot.margin = unit(c(1,1,1,1), "cm"))` and the four values are in order top, right, bottom, left (think trouble: t r b l);
4. If you want to filter your data to include multiple items, use this: `filter(variable %in% c("string 1","string 2","string 3"))`. The `c()` creates a vector of elements, and `%in%` is telling it to filter based on whether the variable is an element of the set. If you wanted to eliminate any variable that is an element of the set, you could use `filter(!variable %in% c("string 1","string 2","string 3"))`, where the `!` acts as a 'not' symbol.

Deliverable 3

The third deliverable for this assignment should be easy if you got the last one. I want you to graph WCS prices for each of the report's scenarios over time.



Data via Canadian Energy Regulator, Energy Futures (2023), graph by Andrew Leach

You should only need one hint for this one: in this case, your observation *groups* are going to be the three scenarios, so you need something like `geom_line(aes(year,value,group=scenario,color=scenario),linewidth=.65)` to get the graph to separate the three series. I also used `mutate(scenario=as_factor(scenario))` to order the scenarios in the same order as they appear in the data, and the same color scale as the previous graph to set the line colors here. You don't have to replicate all of these elements exactly, and I encourage you to do what works for you.

Deliverable 4

Canada is largely an oil price taker. You've made two graphs and a table on oil prices and how they might respond in each of these three scenarios. What can you tell me about Canada's potential exposure to global and domestic action on climate change and/or global energy technology evolution? Make sure you tell briefly me what each scenario assumes about global and domestic action on climate change, what each scenario holds for oil prices, and how that might affect Canadian oil production. (200 words, maximum)

Deliverable 5

Insert a graph of your choosing using data from the CER Energy Futures Report to support your explanation above. Don't try to do too much - a simple replication of one of the graphs above with one of the other data sets from the CER, or a further manipulation of the data we've just used is all you need.

RMD File and HTML/PDF Preparation

I have made you a basic [RMD file](#) for your use in completing this (and future) assignments.

Before you start making any changes to the markdown, test your ability to knit to html with [this file](#), and make sure you can make an html file before you proceed.

For example, with the document open as in the image below, use the Knit button to create your html.

The image shows the RStudio interface with the following components:

- Top Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for file operations, a search bar labeled "Go to file/function", and a "Knit" button.
- Source Editor:** Displays the R Markdown code for "data_assignment_1.rmd". The code includes options for message, warning, and include, followed by a code chunk for loading packages.
- Console:** Shows the R 4.2.2 startup message and the command `R> pagedown::chrome_print("data_assignment_1.rmd")`.

```
10 knitr::opts_chunk$set(message=F,
11                        warning=F,
12                        include=T)
13 #include all your code, but not warnings or loading messages
14 ```
15
16
17 ```{r packages-data,echo=T,output=F, warning=FALSE,include=T}
18 #load your packages
19 library(kableExtra)
20 library(readxl)
21 library(janitor)
22 library(tidyverse)
```

22:19 | Chunk 2: packages-data | R Markdown

R 4.2.2 · ~/

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Hi Andrew, welcome to R

R> pagedown::chrome_print("data_assignment_1.rmd")

Content © 2024 by [Andrew Leach](#)

All content licensed under a [Creative Commons](#)

[Attribution-NonCommercial 4.0 International license \(CC BY-NC 4.0\)](#)

Made with [R](#) and [Quarto](#)

[View the source at](#) [GitHub](#)