



## Data Assignment #2 (due March 8, 2023 by 11:59pm)

This assignment addresses two related subjects: the fiscal impact of oil and gas in Alberta and the pipelines we use to transport our oil and gas.

Canada is an energy exporting country, and our energy exports are an important source of income to governments, in particular in Alberta.

Let's start with the pipelines that transport some of those exports. The [Canadian Energy Regulator](#) produces [pipeline profiles](#) for the largest oil and gas pipelines in Canada, and also tracks other export-related data. The pipeline throughput and capacity data are available [here](#). Crude oil export data sets are available [here](#), while gas trade data are available [here](#).

The US EIA also tracks [crude oil imports from Canada and elsewhere](#), and often provides much more detail than the Canadian export data. Unfortunately, their new Application Programming Interface (API) is very cumbersome and so I've skipped teaching you that step and downloaded some data from their API for you to use for this assignment.

There are five deliverables for this assignment, and they build on skills we have already learned:

- a graph of throughput by grade and capacity on the Enbridge mainline ex-Gretna;
- a graph of throughput by grade and destination on the TransMountain pipeline;
- a graph of throughput and capacity on the TC Energy Canadian Mainline;
- a graph of US oil imports by origin (Canada vs Rest of the World) and refining region (PADDs 1 through 5);
- a graph of Alberta resource royalties over time.

To execute my versions of the deliverables that you see below, I've used the following packages. Use this as a guide to set up your document:

```
library(kableExtra)
library(readxl)
library(janitor)
library(tidyverse)
```

```
library(lubridate)
library(scales)
library(viridis)
library(cowplot)
```

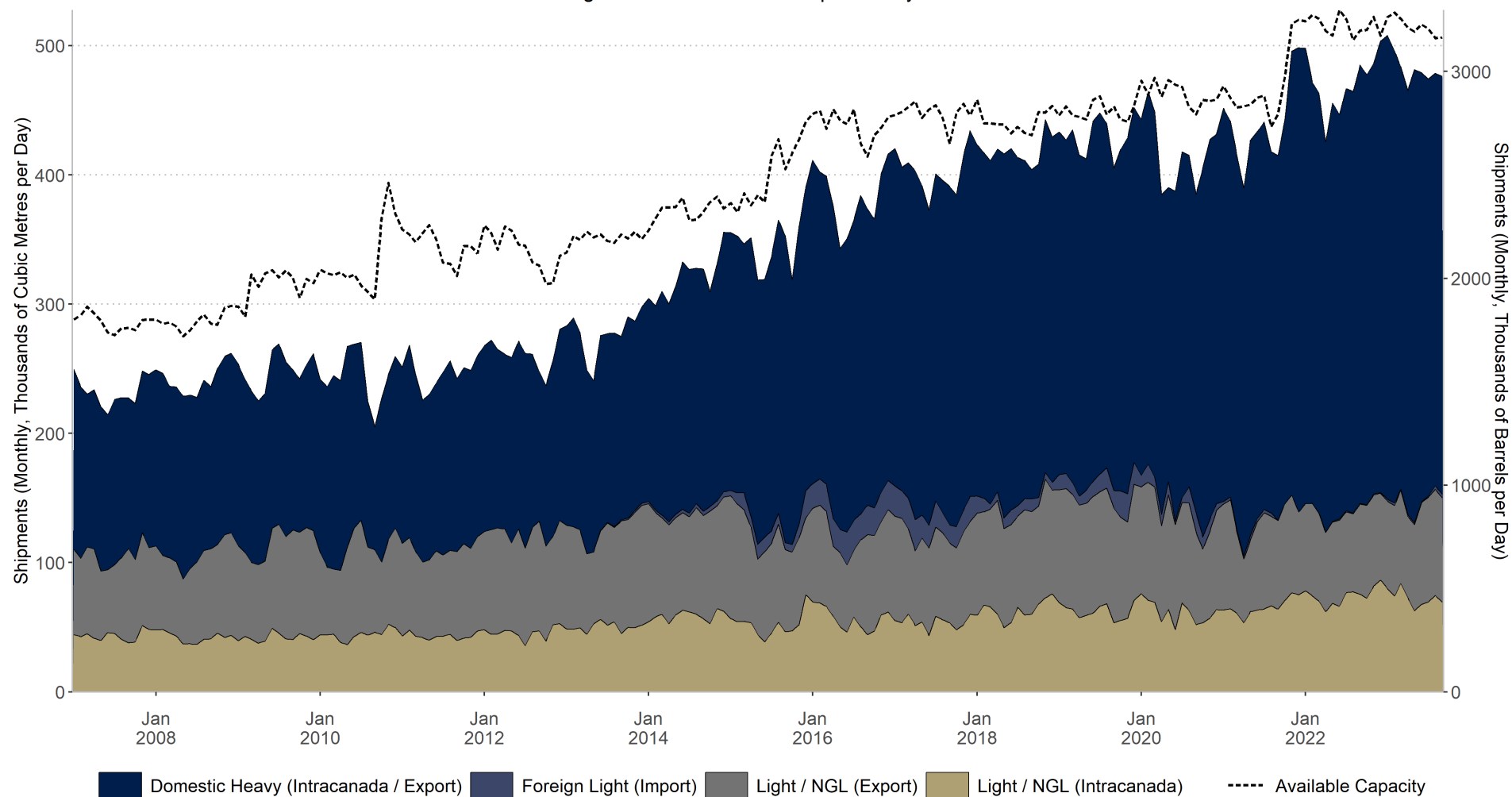
You may also find it useful to have a look at the [Functions Demo](#) before you start this assignment.

## Deliverable 1: the Enbridge Mainline

The Enbridge Mainline moves more crude oil than all other systems combined. For this graph, I expect you to download data from the pipeline profiles and produce a graph in R that shows exports ex-Gretna by grade as well as capacity at that point. You'll need to filter the data to make it useful for your graph, and you'll need to combine an area plot and a line plot.

You should produce something like (but not necessarily identical to) this:

## Enbridge Canadian Mainline Shipments by Product



Source: CER Data for Enbridge Mainline (ex-Gretna), graph by Andrew Leach.

## Hints:

To add the second y axis, use `scale_y_continuous(expand = c(0, 0), sec.axis = sec_axis( trans=~.*1/.16, name="Shipments (Monthly, Thousands of Barrels per Day)"))` which translates the units using the multiplier in `trans=~`;

You'll probably want to download a csv and look at it. You'll also likely want to do some fixing of cases. You can do this in R or in your CSV (but it's easier in R);

There are a couple of ways to make a variable out of two existing variables for graphing. We've previously used the `interaction()` function, e.g. `group=interaction(product,trade_type)`, or you can also use paste, as I did in another part of the functions demo. For example, you could create a new variable using `mutate(pair=paste(product,"(",trade_type,")",sep=""))` which will create, for example, a variable like **Light / NGL (Export)**;

If you're getting a weird looking graph, you're probably graphing by year. Check your data to see what you really should be using.

To add a capacity line, there are a couple of ways to do it, but I'll show you the way I did it here. First, make a separate data set for capacity, and use the mean in each month:

```
enb_capacity<-enb_data%>%group_by(date)%>%summarize(capacity=mean(available_capacity_1000_m3_d,na.rm=T))
```

And then, you can add a line into your graph using those data. This let's me show you a neat trick too - if you name the colour in a ggplot aesthetic, this will carry through to your legends.

```
geom_line(data=enb_capacity,aes(date,capacity,color="Available Capacity"),linewidth=.85, lty="21") +  
scale_color_manual("",values=c("black"))
```

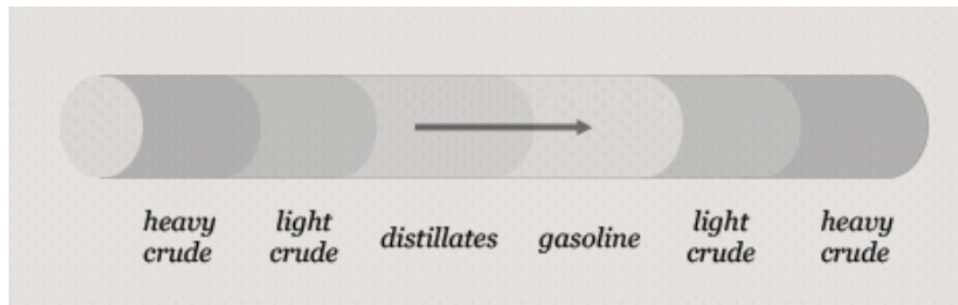
You should also note that, for a time before the opening of the Dakota Access Pipeline, there was some US-produced crude moving east on the Mainline.

## Deliverable 2: the Trans-Mountain Pipeline

The Trans-Mountain pipeline has been around since the 1950s and is the only major crude oil pipeline to cross the continental divide. Trans-Mountain is also unique in that it ships both crude oil and refined products in batches (other pipelines ship in batches too, just not with refined products in the mix). The following description is taken from the Trans-Mountain website"

Trans Mountain is the only pipeline in North America that carries both refined product and crude oil in batches.

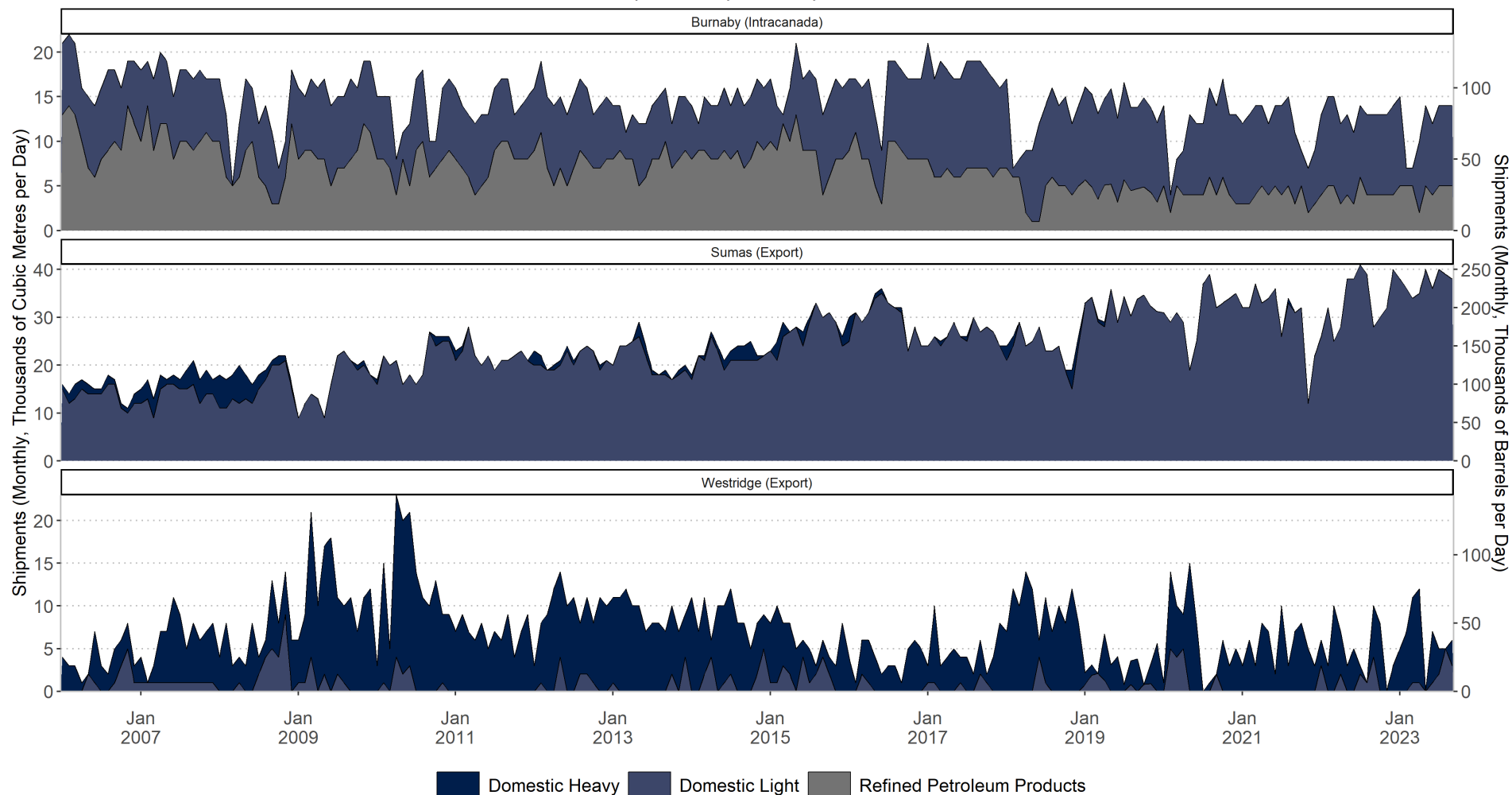
On any given day, the pipeline is used to move different grades or varieties of petroleum. Products moving next to each other in the pipeline can mix. This mixing – or product interface – is minimized by putting the products in a specific sequence.



Transmountain also has three delivery points: exports to the US via Sumas, deliveries to domestic refining in Burnaby, and a port delivery point at Westridge.

To see what gets delivered where on TransMountain, you should produce something like (but not necessarily identical to) this for TransMountain:

## Trans-Mountain Pipeline Shipments by Product and Destination

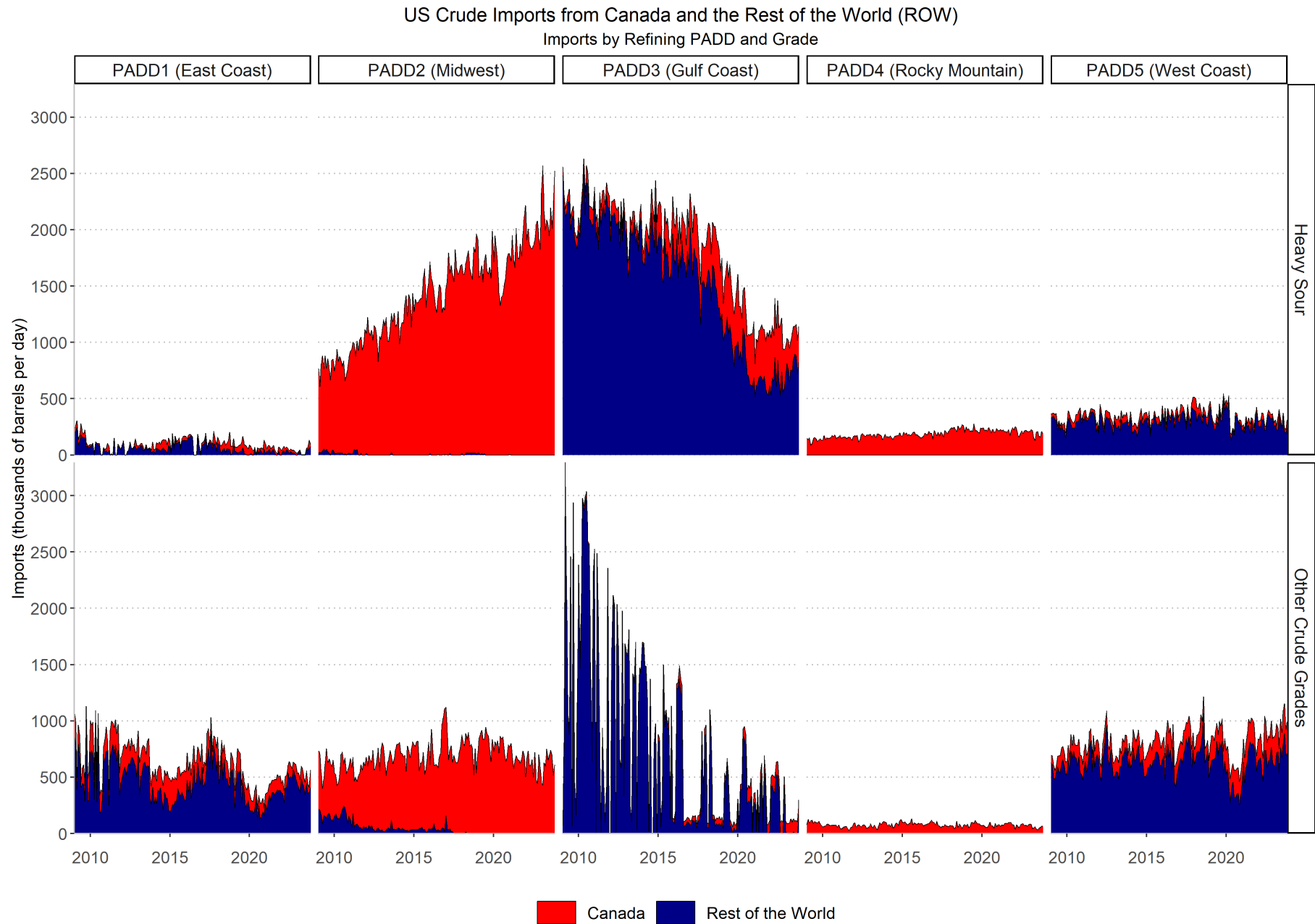


Source: CER Data for Trans-Mountain, graph by Andrew Leach.

Hint: use `facet_wrap(~key_point, ncol = 1, scales = "free_y")` to allow different y-axes on each of the plots, and to stack the plots vertically if you like. You are not required to present them this way, but I do want to see a three-facet plot

## Deliverable 3: US imports of Canadian crude

We haven't talked a lot about the US market, but it's worth having a sense of how the US relies on imports. Using data that I've provided for you [here](#), you should be able to reproduce a graph that looks like this one for use imports by grade and refining region. Note that refining regions give you a better sense of where demand lies, rather than a focus on the import port or pipeline border crossing point.



Data via US Energy Information Administration, current to November 2023. Graph by Andrew Leach.



## Hints:

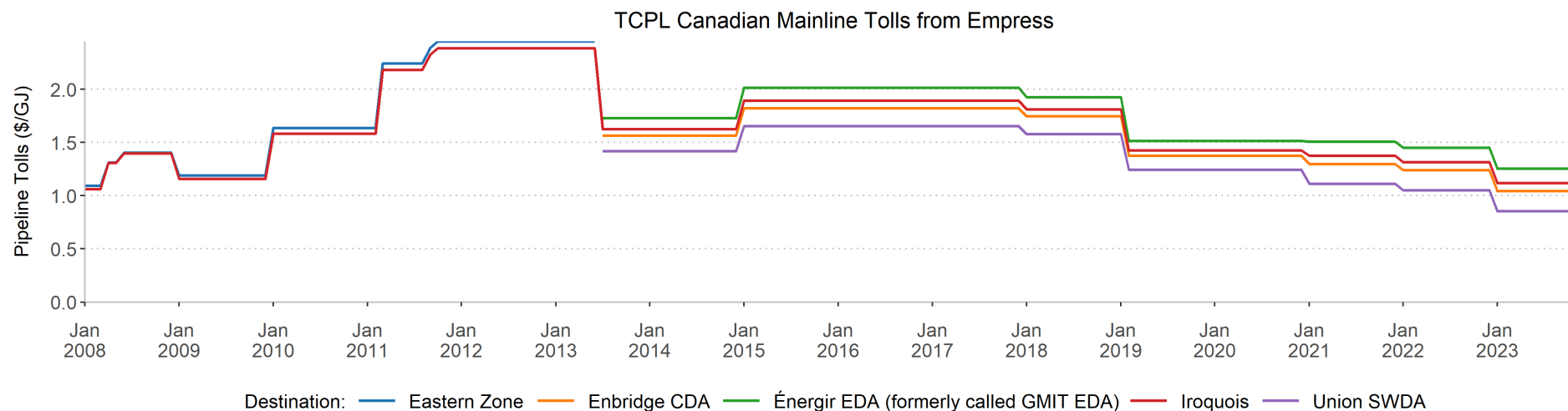
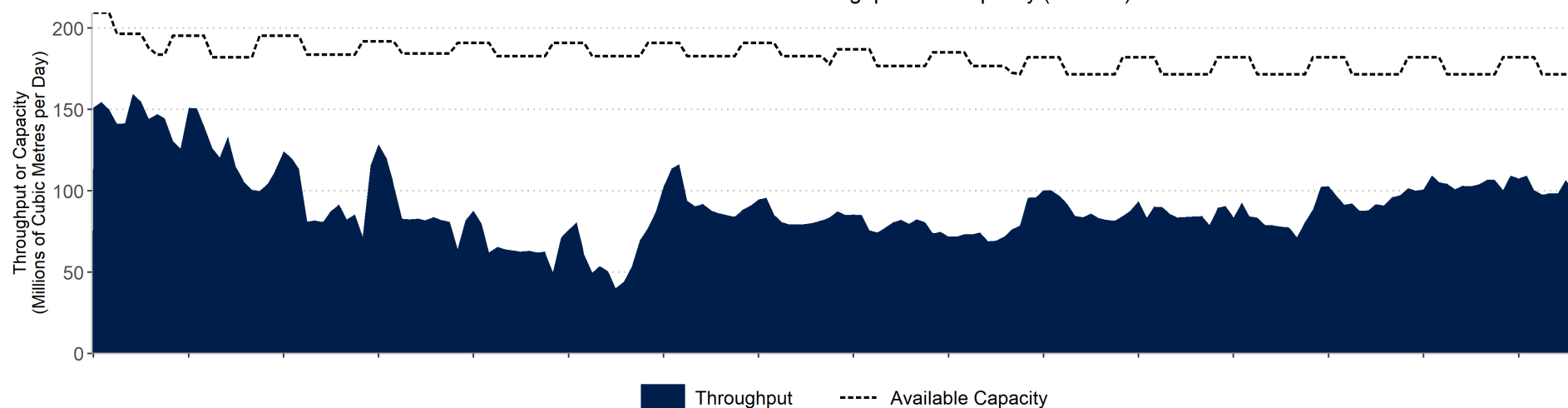
Think about your `facet_grid` elements (rows and cols), your grouping of data, and your fill aesthetic.

## Deliverable 4: The TC Energy Canadian Mainline

The first pipeline controversy in Canadian political history was not about an oil pipeline, but about a gas pipeline. The [Pipeline Debate](#) of 1956 eventually led to the construction of the Canadian mainline. The Mainline also became the subject of one of the most important regulatory decisions in recent history, as the NEB refused to allow the pipeline to enter into a *downward spiral* and capped tolls and directed TransCanada to find new business to improve the viability of its asset. Both tolls and throughputs are available on the pipeline profile site.

To see how flows have changed on the Mainline over time, and to see when the regulatory changes really took hold, you should produce a graph like this:

## TCPL Canadian Mainline Throughput and Capacity (Prairies)



Source: CER data for TCPL Canadian Mainline, graph by Andrew Leach.

## Hints:

The Mainline data files are large, so depending on the speed of your connection, you may get timeout errors. If you do, just add the command options(timeout=300) to your code **before** you call the download command. You likely also want to make sure you're not downloading them every time you run your code, so use an if(!file.exists..) in your code as we've done before;

You're going to want to use either plotgrid or cowplot to combine your plots. I used `plot_grid(throughput_plot, tolls, ncol = 1, align = "v") ;`

You'll want to use a `group_by` and `summarize` to make average monthly plots. For the toll data, I used `group_by(month, year, path) %>% summarize(toll = mean(toll, na.rm=T))` and `mutate(date = ymd(paste(year, month, 1, sep = "-")))` to accomplish this.

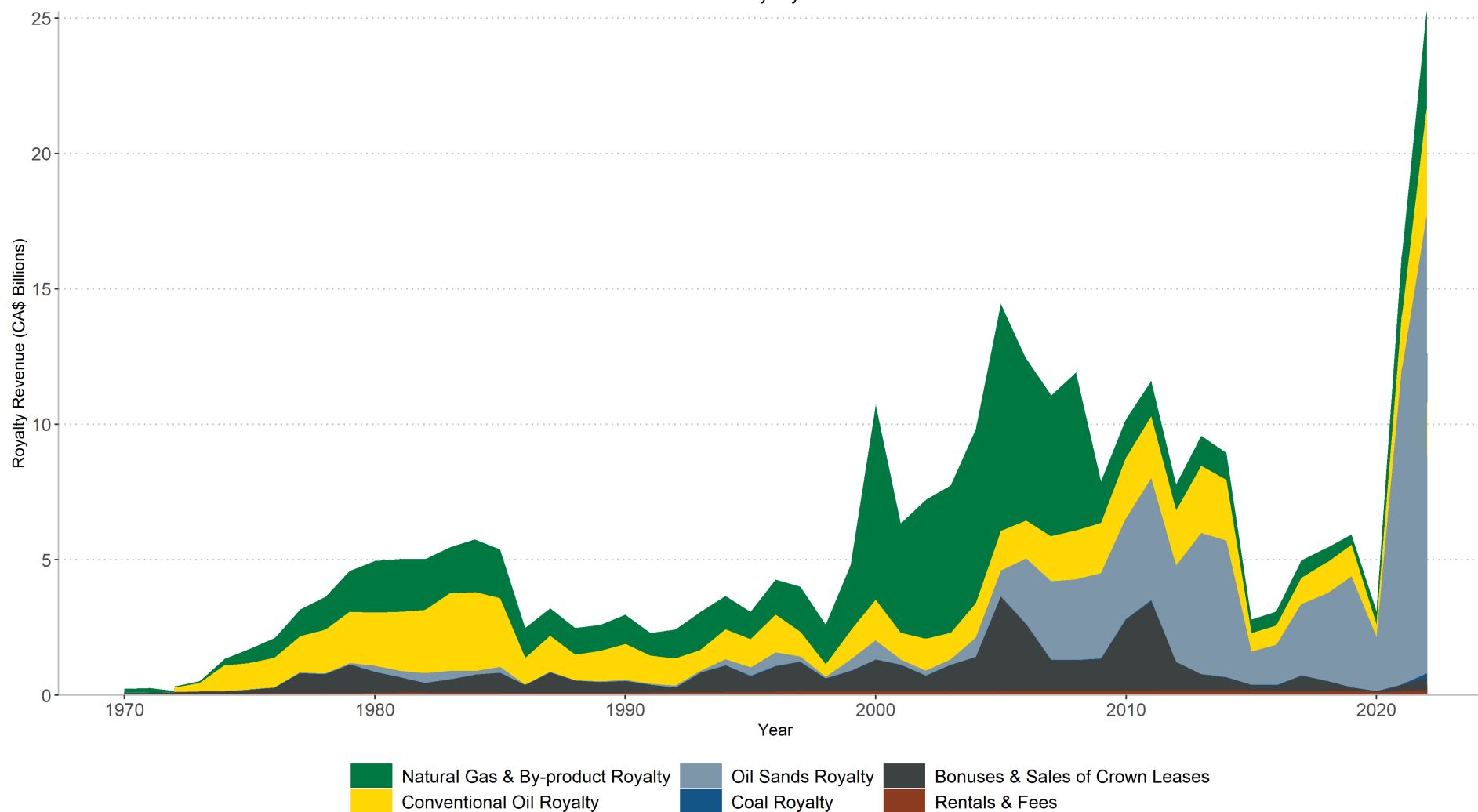
Use theme commands to suppress the x axis names and titles on your top graph;

Filter the data to keep the same starting date in both cases.

## Deliverable 5: Alberta Resource Royalties

Alberta's budget comes out this week, and resource royalties are always a big part of the story. In the last full fiscal year, resource royalties hit record levels and the forecast of future royalties will drive government spending and savings decisions. To have some perspective on these decisions, let's make a plot of Alberta resource royalties by product over time. Use the newest data [here](#) to create a graph similar to the following:

## Alberta Royalty Revenue



## Hints:

To turn the fiscal years into single years, I used this command: `mutate(year = as.numeric(gsub("x","",str_split(year, "_", simplify = TRUE)[ , 1])))`. This splits the string at the “\_”, and then replaces the x with a blank. So, for example, if your year entry was “x1970\_71”, you run this command and it first grabs the first part of the outcome of splitting the string at the “\_” (x1970) and then uses gsub to replace the x with a blank.

Other than altering how the fiscal years are read in, you should not need any other new commands. You do not need to replicate the U of A colours.

## RMD File and HTML/PDF Preparation

As before, use the basic [RMD file](#) to complete this (and future) assignments, just rename it assignment 2. You only need to submit the HTML.

---

Content © 2024 by [Andrew Leach](#)

All content licensed under a [CC BY-NC 4.0 International license](#) (CC BY-NC 4.0)

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

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

[View the source at GitHub](#)