Install and load libraries

We need to first install all the required packages for the data analysis process. If you have never used the packages below, it is more likely that you have not installed them on your machine either. Please make sure you install each of the packages below using the following command:

```
install.packages("here")
install.packages("janitor")
# install.packages('tidyverse')
install.packages("dplyr")
install.packages("stringr")
install.packages("ggplot2")
install.packages("treemapify")
install.packages("knitr")
install.packages("extrafont")
# font_import() # Only do this once
```

Then you need to load the following packages:

```
library(here)
library(janitor)
# library(tidyverse)
library(dplyr)
library(stringr)
library(ggplot2)
library(scales)
library(treemapify)
library(knitr)
library(extrafont)
loadfonts(device = "win")
```

Import clean dataset

```
unfccc_emissions <- utils::read.csv(here("scripts/cleaning/unfccc-emissions",
    "unfccc-emissions-clean.csv"), stringsAsFactors = TRUE)
```

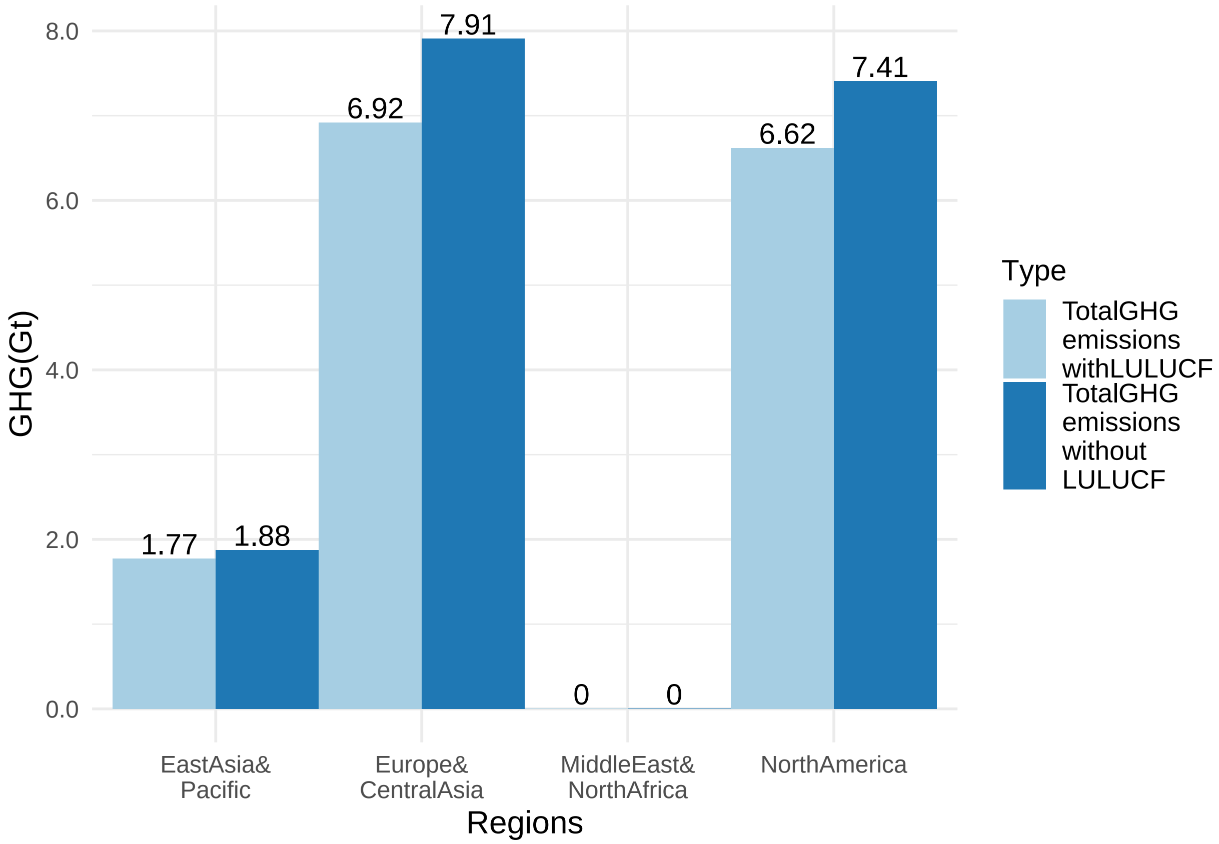
Data visualizations

Now that we've analyzed the data it's time to produce some visualizations. Given that there are many missing values within the Non-Annex I countries, we should avoid making any inference about their emissions based on this dataset. However, we can visualize Annex I countries' cumulative and annual emissions and figure the most contributing emitters based on their region.

First, let's explore the highest emitting regions in 2018.

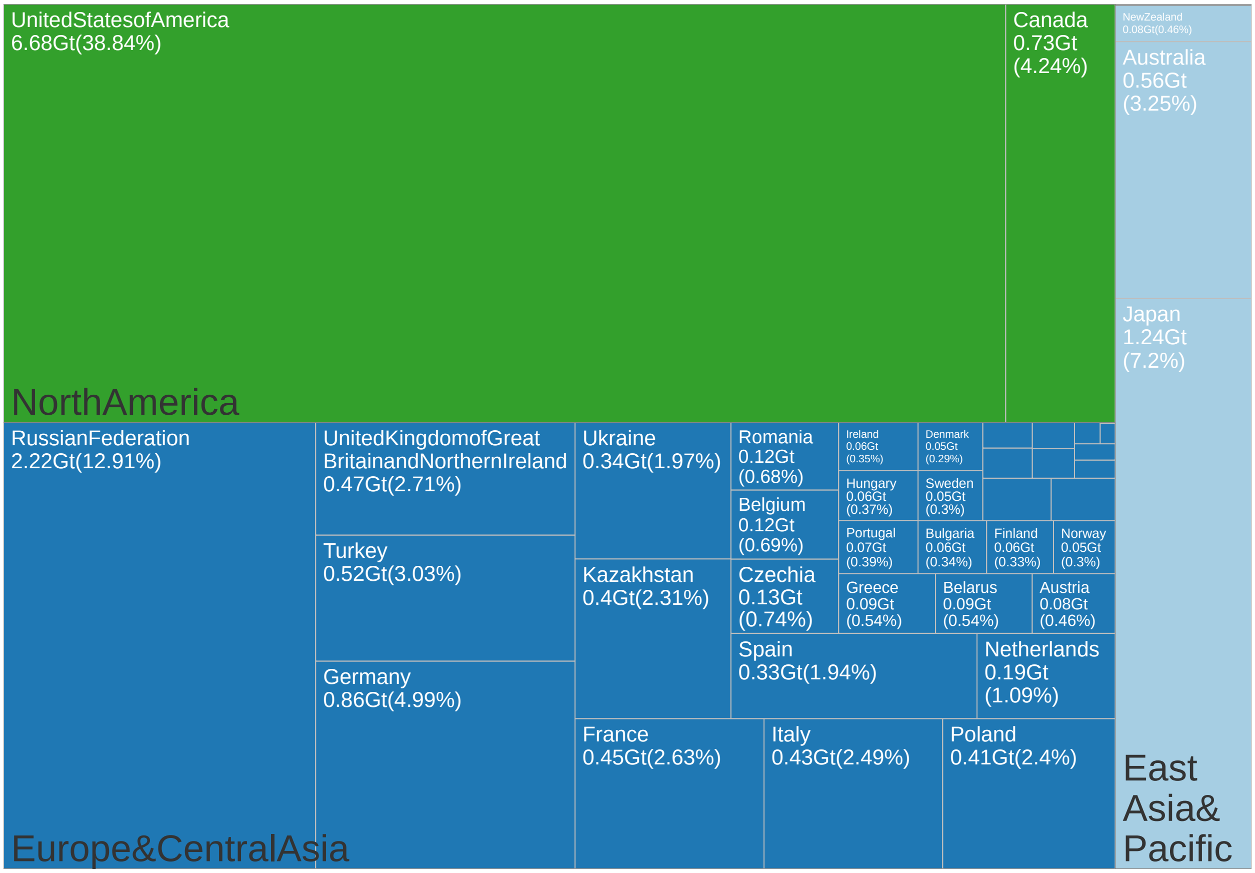
```
unfccc_emissions %>%
    dplyr::filter(year == "2018", group ==
        "Annex I", !(country %in% c("European Union (Convention)",
        "European Union (KP)"))) %>%
    dplyr::group_by(region, type) %>%
    dplyr::summarise(total = sum(ghg)) %>%
    ggplot2::ggplot(aes(x = as.factor(region),
        y = total, fill = stringr::str_wrap(as.factor(type),
            12))) + ggplot2::scale_fill_brewer(palette = "Paired",
    name = "Type") + ggplot2::geom_bar(stat = "identity",
    position = position_dodge(), width = 1) +
    ggplot2::geom_text(aes(label = round(total/1e+06,
        digits = 2)), vjust = -0.2, position = position_dodge(width = 0.9)) +
    ggplot2::theme_minimal() + ggplot2::xlab("Regions") +
    ggplot2::ylab("GHG (Gt)") + ggplot2::scale_y_continuous(labels = scales::unit_format
    scale = 1e-06)) + ggplot2::scale_x_discrete(labels = function(x) stringr::str_wrap(x
    width = 15)) + # theme(legend.position = 'none') +
ggplot2::theme(text = element_text(family = "Arial",
    size = 11), axis.title.x = element_text(family = "Arial",
    size = 12), axis.title.y = element_text(family = "Arial",
    size = 12), legend.text = element_text(family = "Arial",
    size = 10))
```

```
## `summarise()` has grouped output by 'region'. You can override using the `.groups` ar
```



When we graph by region and whether it's produced or net emissions (with/without LULUCF), we see that net emissions give Annex I countries the chance to report lower emissions. We can also see that the majority of emissions are located in Europe and Central Asia and North America for Annex I countries so we need to interrogate why that is.

```
unfccc_emissions %>%
    dplyr::filter(year == "2018", group ==
        "Annex I", type == "Total GHG emissions without LULUCF",
        !(country %in% c("European Union (Convention)",
            "European Union (KP)"))) %>%
    dplyr::mutate(percent = round(ghg/sum(ghg) *
        100, digits = 2)) %>%
    dplyr::group_by(region, type) %>%
    ggplot2::ggplot(aes(area = ghg, fill = region,
        subgroup = region, label = paste(country,
            paste(round(ghg/1e+06, digits = 2),
                " Gt (", percent, "%)", sep = ""),
            sep = "\n"))) + ggplot2::scale_fill_brewer(palette = "Paired") +
    treemapify::geom_treemap() + treemapify::geom_treemap_subgroup_border(size = 0) +
    treemapify::geom_treemap_subgroup_text(grow = FALSE,
        place = "bottomleft", size = 14,
        color = "grey20", reflow = TRUE,
        family = "Arial") + ggplot2::theme_void() +
    treemapify::geom_treemap_text(colour = "white",
        place = "topleft", reflow = T, size = 8,
        family = "Arial") + ggplot2::theme(legend.position = "none")
```

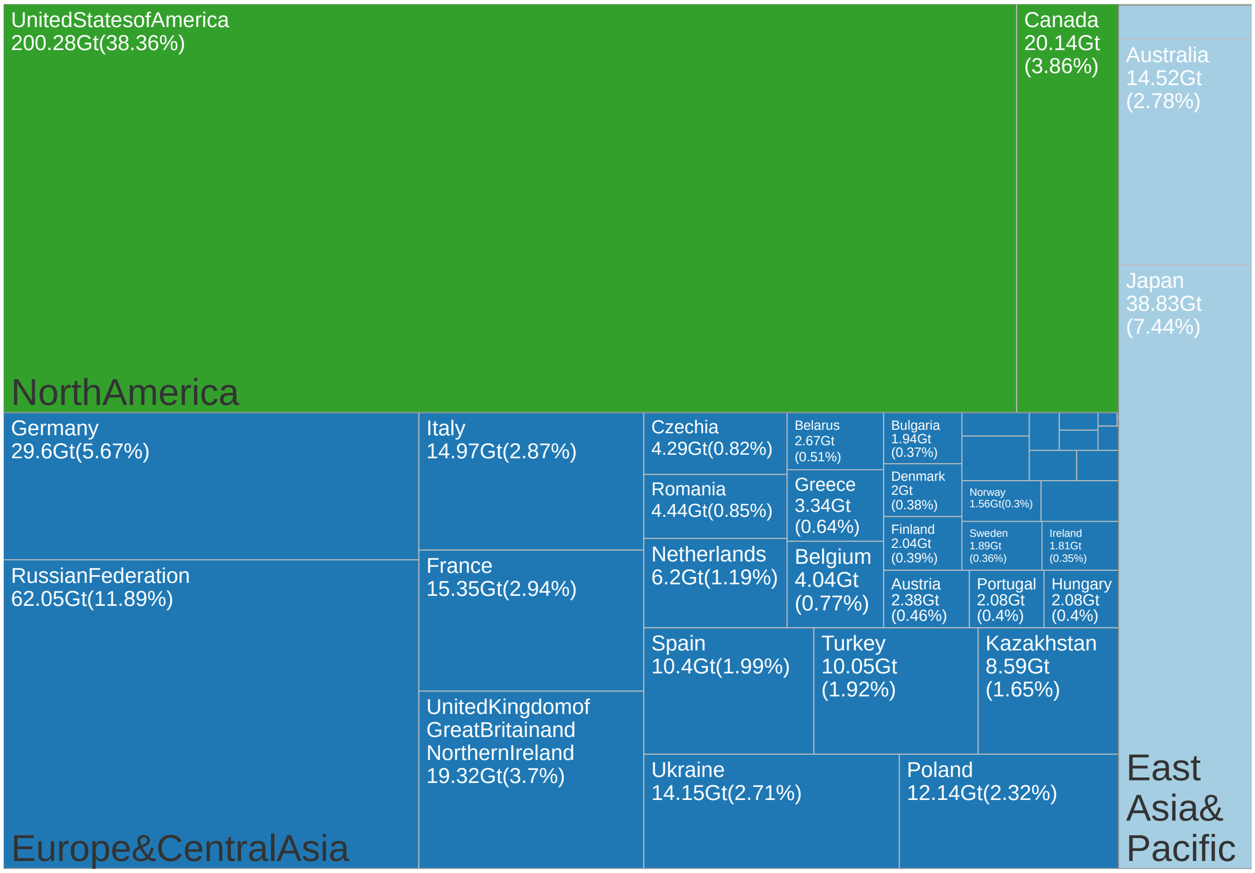


```
ggplot2::ggsave(here("images", "annex-i-ghg-2018.svg"),
    device = "svg", dpi = 300)
```

```
## Saving 6.5 x 4.5 in image
```

Now let's see if it's the same story when we look at the cumulative GHG emissions from 1990 to 2018.

```
unfccc_emissions %>%
    dplyr::filter(year != "base_year", group ==
        "Annex I", type == "Total GHG emissions without LULUCF",
        !(country %in% c("European Union (Convention)",
            "European Union (KP)"))) %>%
    dplyr::group_by(iso, country, region) %>%
    dplyr::summarise_if(is.numeric, sum,
        na.rm = TRUE) %>%
    dplyr::ungroup() %>%
    dplyr::mutate(percent = round(ghg/sum(ghg) *
        100, digits = 2)) %>%
    dplyr::group_by(country, region) %>%
    ggplot2::ggplot(aes(area = ghg, fill = region,
        subgroup = region, label = paste(country,
            paste(round(ghg/1e+06, digits = 2),
                " Gt (", percent, "%)", sep = ""),
            sep = "\n"))) + ggplot2::scale_fill_brewer(palette = "Paired") +
    treemapify::geom_treemap() + treemapify::geom_treemap_subgroup_border(size = 0) +
    treemapify::geom_treemap_subgroup_text(grow = FALSE,
        place = "bottomleft", size = 14,
        color = "grey20", reflow = TRUE,
        family = "Arial") + ggplot2::theme_void() +
    treemapify::geom_treemap_text(colour = "white",
        place = "topleft", reflow = T, size = 8,
        family = "Arial") + ggplot2::theme(legend.position = "none")
```



```
ggplot2::ggsave(here("images", "annex-i-ghg-cumulative.svg"),
    device = "svg", dpi = 300)
```

```
## Saving 6.5 x 4.5 in image
```

```
rm(unfccc_emissions)
```

Export as an R script for future use

Only run this chunk manually once within the .Rmd file. It produces an error when knitting it as a whole because of chunk label duplicates. As of May 12, 2021, there hasn't been a viable solution to run the code below when as part of the knitting process.

```
knitr::purl("unfccc-emissions-visualizations.Rmd",
    "unfccc-emissions-visualizations.R")
knitr::write_bib(.packages(), "packages.bib")
```

Software used

Firke, Sam. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*, 2021. https://github.com/sfirke/janitor.

Müller, Kirill. *Here: A Simpler Way to Find Your Files*, 2020. https://CRAN.R-project.org/package=here.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2021. https://www.R-project.org/.

Wickham, Hadley. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. https://ggplot2.tidyverse.org.

———. *Stringr: Simple, Consistent Wrappers for Common String Operations*, 2019. https://CRAN.R-project.org/package=stringr.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*, 2020. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. *Dplyr: A Grammar of Data Manipulation*, 2021. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Dana Seidel. *Scales: Scale Functions for Visualization*, 2020. https://CRAN.R-project.org/package=scales.

Wilkins, David. *Treemapify: Draw Treemaps in Ggplot2*, 2021. https://wilkox.org/treemapify/.

Winston Chang. *Extrafont: Tools for Using Fonts*, 2014. https://github.com/wch/extrafont.

Xie, Yihui. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC, 2015. https://yihui.org/knitr/.

———. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC, 2014. http://www.crcpress.com/product/isbn/9781466561595.

———. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*, 2021. https://yihui.org/knitr/.