Sankey Diagrams: A Method to Visualize Data Flow

Special thanks to Dr. Scerri and Dr. Lucero

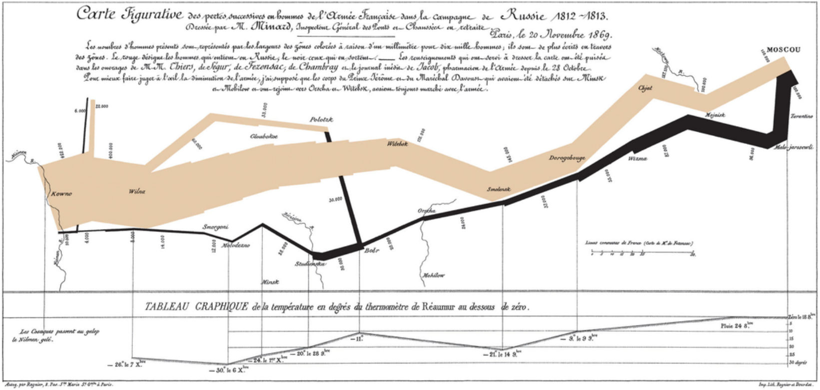
Daniel Palamarchuk

April 28, 2022

Introduction

- What is a Sankey Diagram?
  - Simply put, method to visualize data that "flows" between different processes
  - Example use cases: linking majors to careers, energy consumption, life-time of bills
- Sankey diagrams are named after a man named Matthew Henry Sankey who used it to demonstrate the efficiency of energy transfer within a steam engine

Examples



Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zônes colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zônes. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. —— Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.
Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.

Another Example

image source: Ben Schmidt

Example I found here

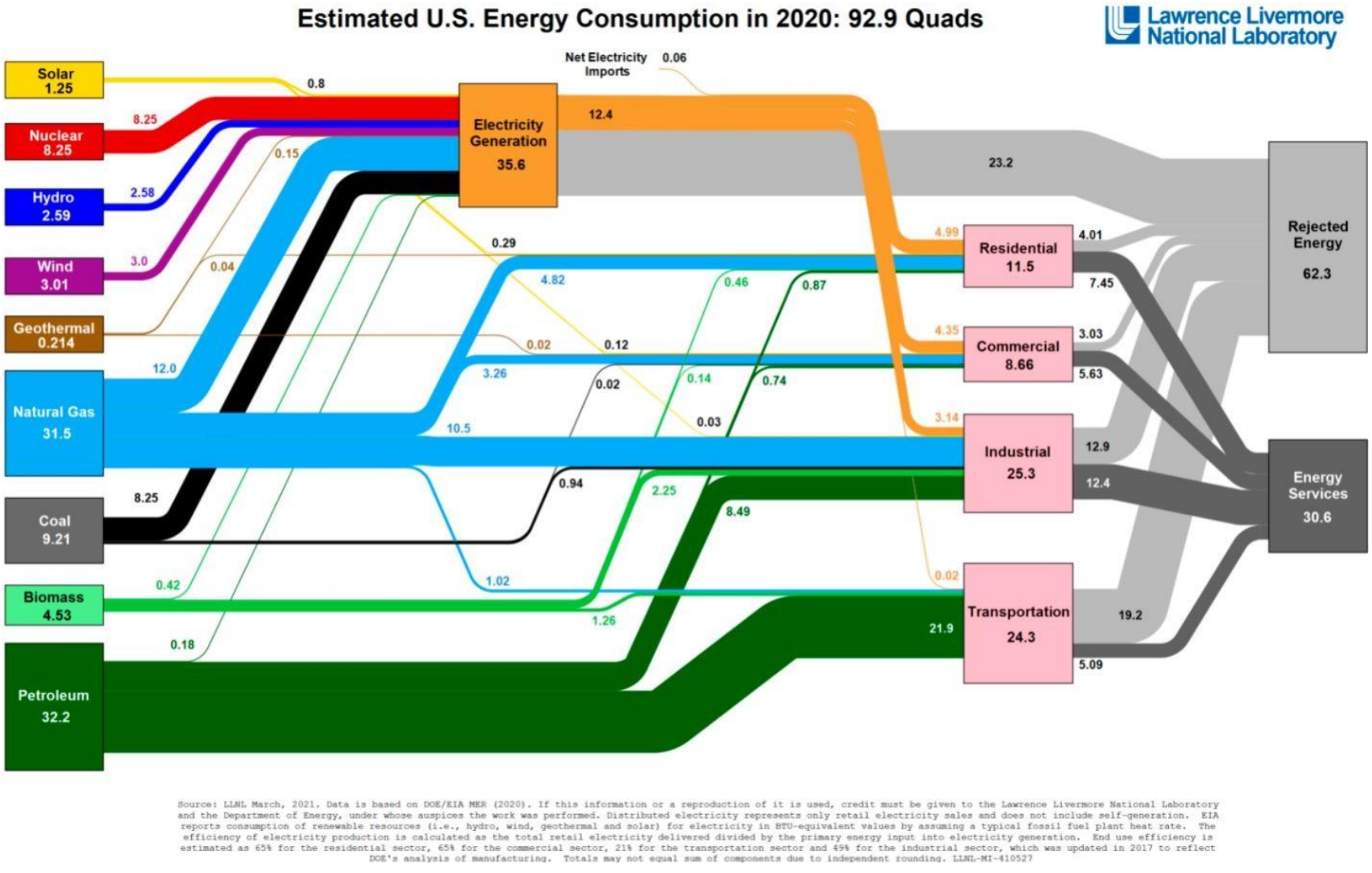


image source: Life in the Built Environment

Creating Sankeys

There are several packages that implement sankey diagrams/have sankey capabilities built on top of them. Let us start off with ggplot's implementation: ggsankey.

```
#install.packages("devtools")
#devtools::install_github("davidsjoberg/ggsankey")
library(ggsankey)
library(ggplot2)
library(dplyr)
head(mtcars[,c("gear", "cyl", "am", "carb")])
```

```
##                   gear cyl am carb
## Mazda RX4            4   6  1    4
## Mazda RX4 Wag        4   6  1    4
## Datsun 710           4   4  1    1
## Hornet 4 Drive       3   6  0    1
## Hornet Sportabout    3   8  0    2
## Valiant              3   6  0    1
```

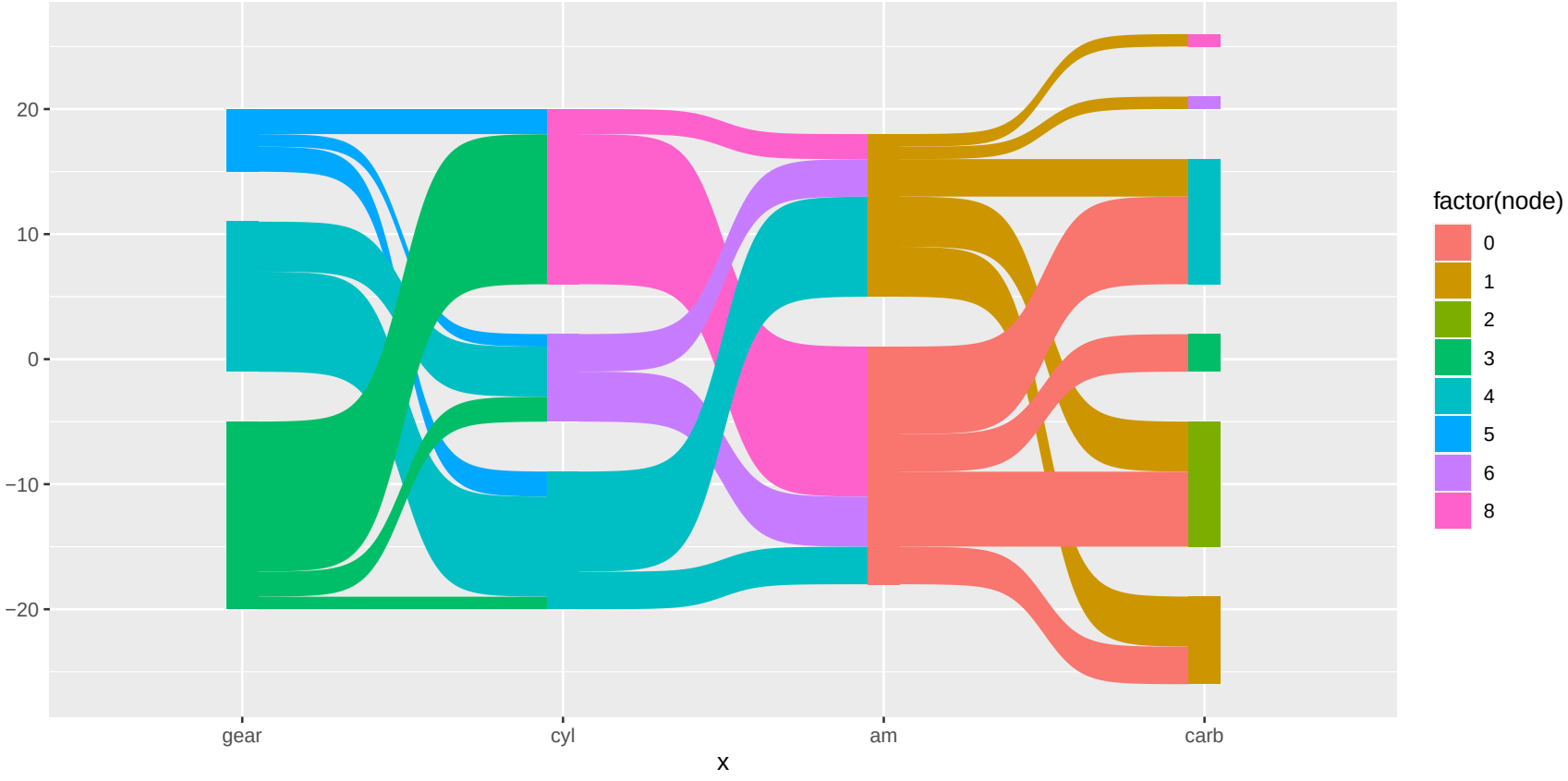
Baby Example

```
mt_sankey <- make_long(
    mtcars,
    gear,
    cyl,
    am,
    carb
)
head(mt_sankey)
```

```
## # A tibble: 6 x 4
##   x       node next_x next_node
##   <fct> <dbl> <fct>       <dbl>
## 1 gear      4 cyl             6
## 2 cyl       6 am              1
## 3 am        1 carb            4
## 4 carb      4 <NA>           NA
## 5 gear      4 cyl             6
## 6 cyl       6 am              1
```

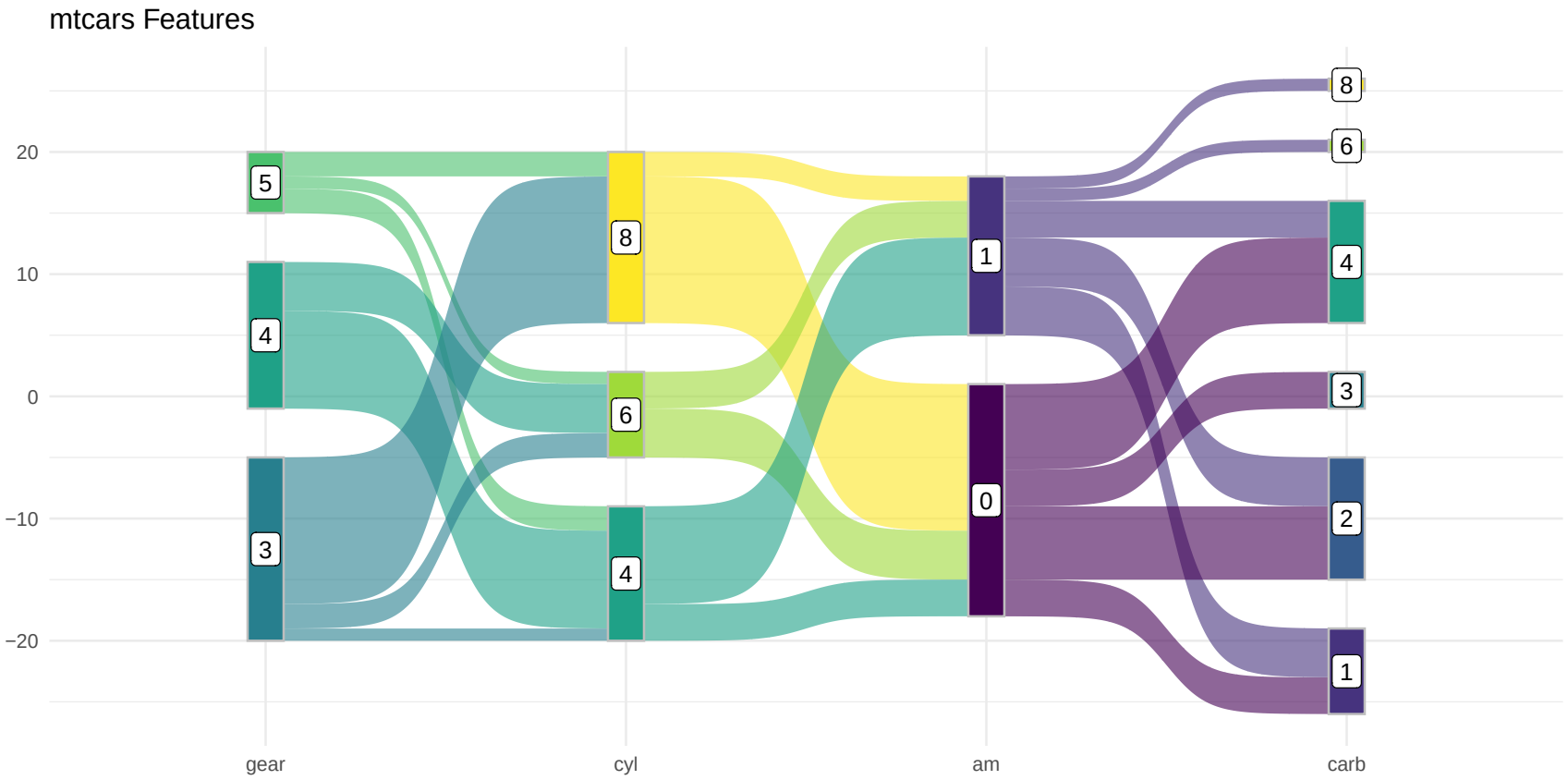
Plot

```
ggplot(mt_sankey, aes(x = x,
                      node = node,
                      next_node = next_node,
                      next_x = next_x,
                      fill = factor(node))) +
    geom_sankey()
```



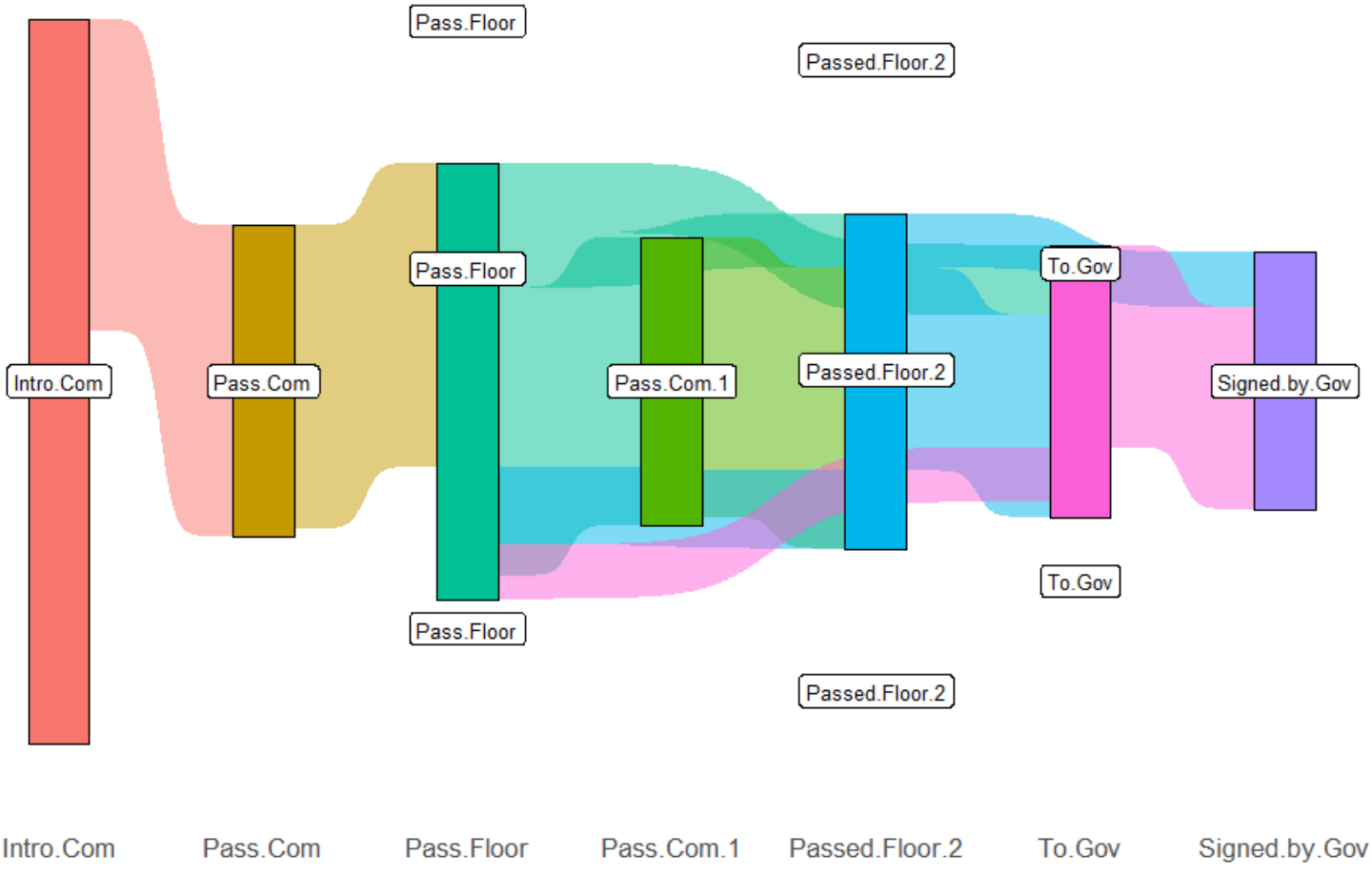
Fancier Plot

```
ggplot(mt_sankey, aes(x = x, node = node, next_node = next_node,
                      next_x = next_x, fill = factor(node))) +
    geom_sankey(flow.alpha = 0.6, node.color = "gray") +
    scale_fill_viridis_d() +
    geom_sankey_label(aes(label = node), fill = "white") +
    labs(x = NULL, title = "mtcars Features") +
    theme_minimal() + theme(legend.position = "none")
```



Some issues. . .

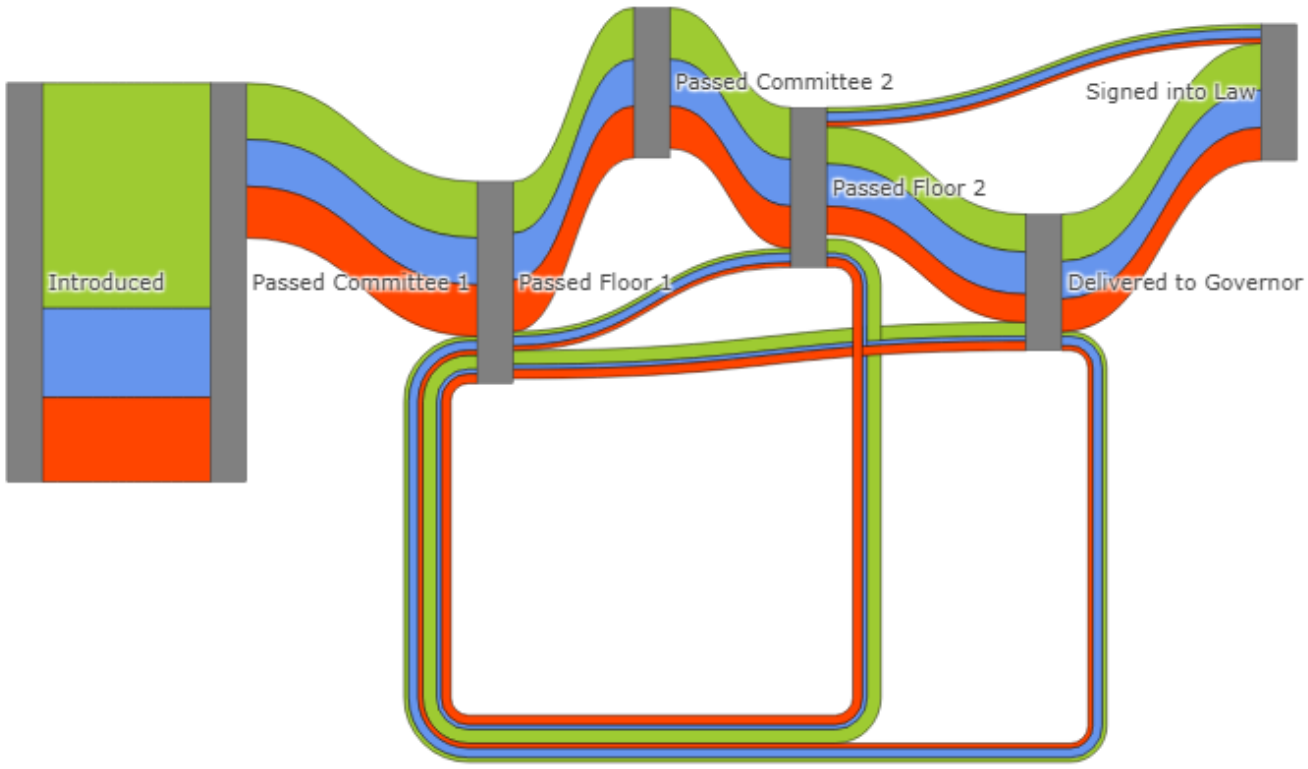
1. ggplot creates static images
2. Some. . . interesting results were generated



A New Challenger Approaches

```
#install.packages("plotly")
library(plotly)
```

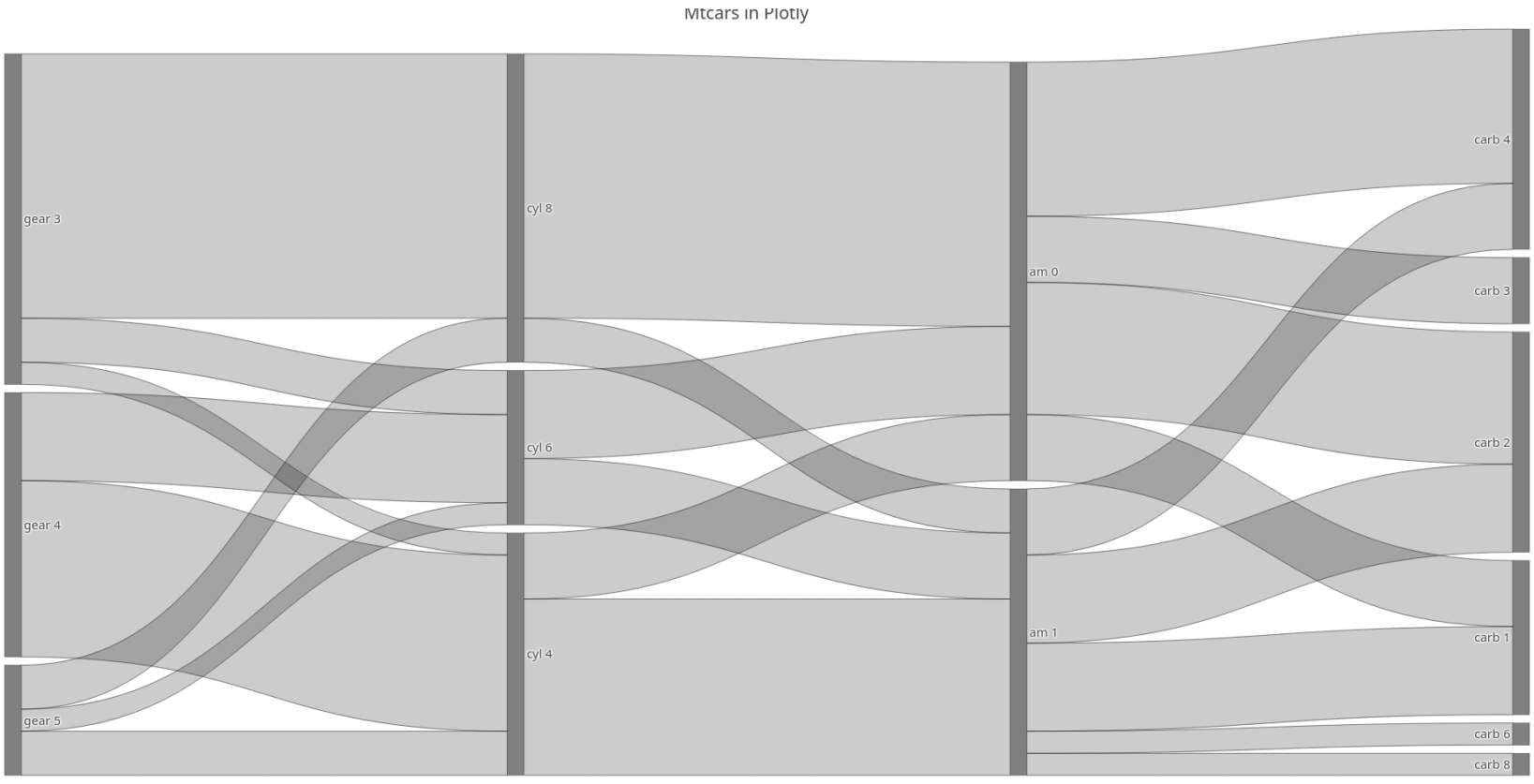
Plotly is a javascript based plotting software that can create several types of graphs, including Sankeys. It solves both of the issues mentioned above, making it the ideal choice for my research project.



Comparison in Input

```
mt_plotly <- mutate(mt_sankey, xnode = factor(paste(x, node))) %>%
    mutate(xnextnode = factor(paste(next_x, next_node),
                              levels = levels(xnode)))
levs <- (levels(mt_plotly$xnode))
mt_plotly <- filter(mt_plotly, !is.na(node), !is.na(next_x),
                    !is.na(next_node)) %>%
    group_by(xnode, xnextnode) %>% summarize(n = n())
plot_ly(
  type = "sankey", arrangement = "snap",
  node = list(color = "gray", label = levs, pad = 10),
  link = list(
    source = as.numeric(mt_plotly$xnode) - 1,
    target = as.numeric(mt_plotly$xnextnode) - 1,
    value = mt_plotly$n, line = list(color = "black", width = 0.5)
    ))%>%
  layout(title = "Mtcars in Plotly",
         xaxis = list(showgrid = F, zeroline = F),
         yaxis = list(showgrid = F, zeroline = F),
         font = list(size = 15),
         showlegend = T)
```

Output

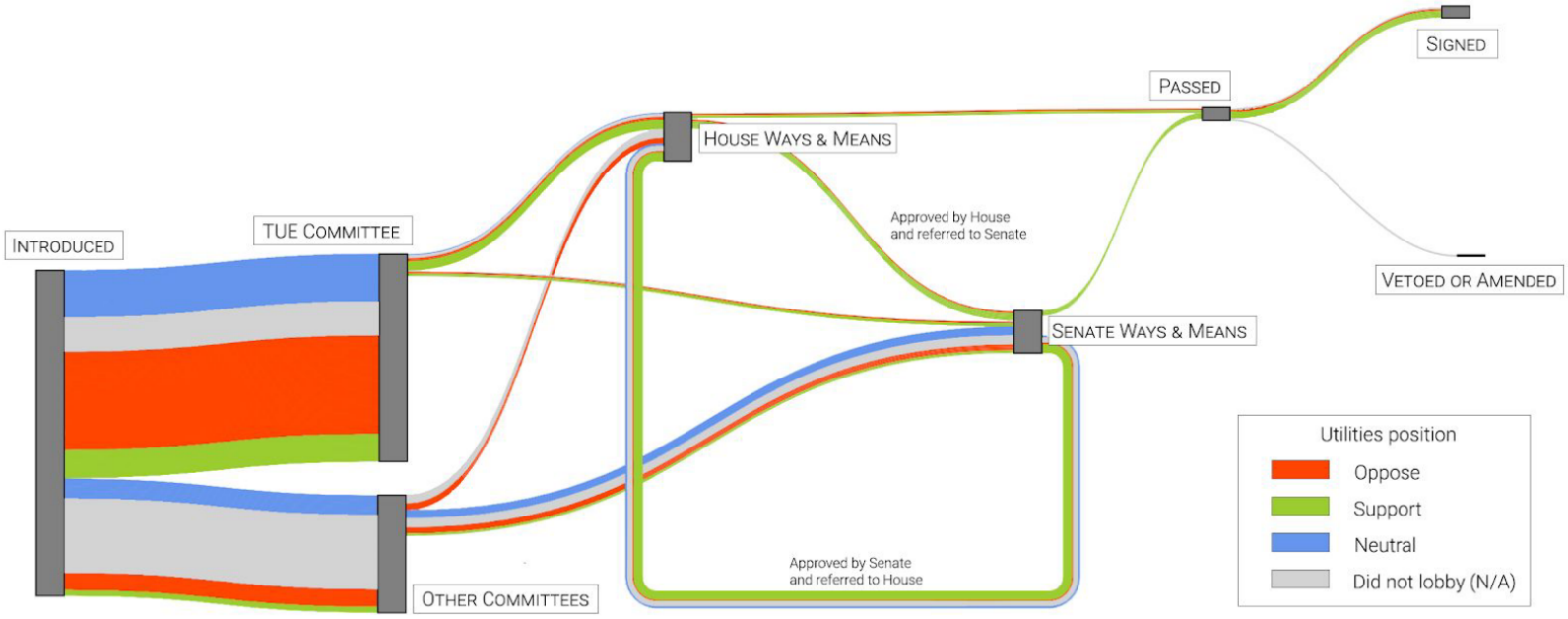

Mtcars in Plotly
gear 3
gear 4
gear 5
cyl 8
cyl 6
cyl 4
am 0
am 1
carb 4
carb 3
carb 2
carb 1
carb 6
carb 8

Research

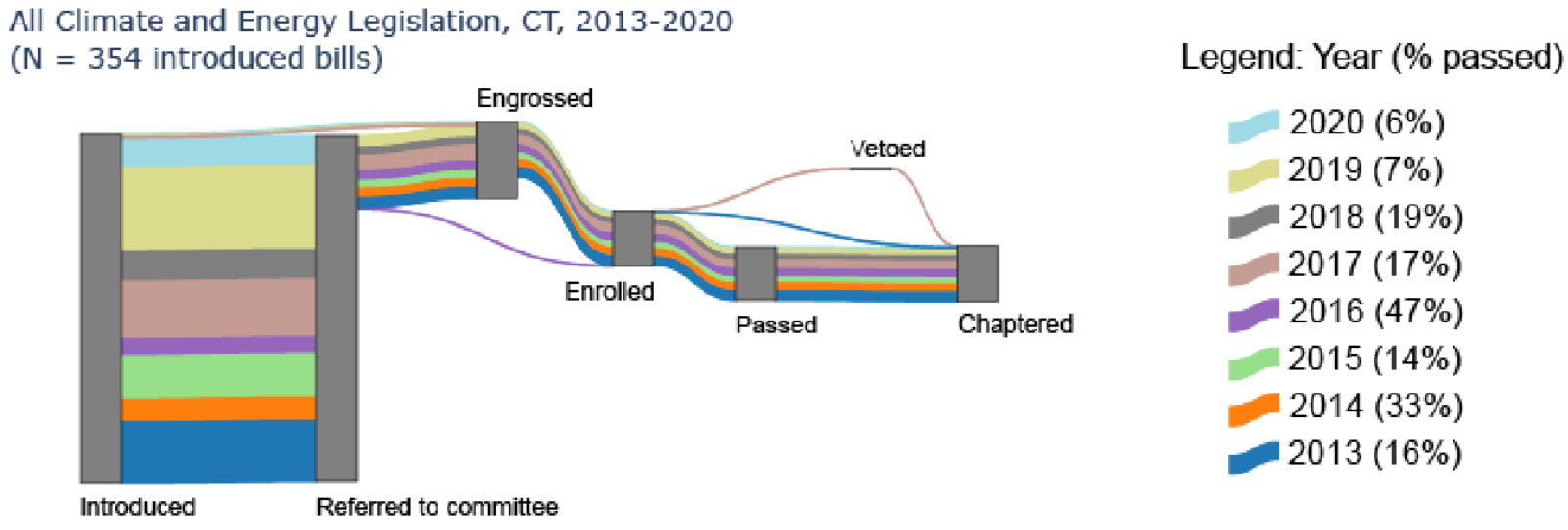
I have been working with Dr. Scerri from PSCI and the Climate Social Science Network since last semester to develop reports on the effects of lobbying on climate legislation. My role was to develop visualizations akin to what previous studies of the sort have been using.

We invited Dr. Lucero to join in for Spring semester as a research project + credit. With his guidance we developed a dashboard to allow people to look at the data for themselves.

Snippets from Other Projects



Connecticut Project



Issues with Virginia

1. Lobbyists do not have to disclose the position they are lobbying for
   - Ended up collaborating with Sierra Club to approximate climate friendliness of bills
2. There is no database to easily access climate data
   - Issue with most states

Final Product

Now for a demonstration