IDEALS-GSE R Workshop

Scott Bailey

7/16/2019

## Installing R and RStudio

See instructions from one of my excellent colleagues, Claudia Engel: <https://cengel.github.io/R-intro/index.html#setup-instructions>.

## Data analysis with R

We’re going to take a very pragmatic approach in our workshop today. Instead of thinking about R as a programming language we’re trying to learn to code in, we’re going to approach R as just a program that we can use to do some fundamental data analysis. In that line of thinking, instead of learning a lot of base R, we’re going to start using the Tidyverse packages and approach right away. Briefly, Tidyverse is a set of opinionated packages with a coherent approach to how to work with data in R. For many of us, they make R easier to learn and to use.

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.1 ✔ purrr 0.3.2   
## ✔ tibble 2.1.1 ✔ dplyr 0.8.0.1  
## ✔ tidyr 0.8.3 ✔ stringr 1.4.0   
## ✔ readr 1.3.1 ✔ forcats 0.4.0

## ── Conflicts ────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

First, let’s read in some data that might be interesting for education research. Here’s a link to data on 2010 School Improvement Grants that I found on data.gov: <https://catalog.data.gov/dataset/school-improvement-2010-grants>.

data <- read\_csv("https://inventory.data.gov/dataset/ce23c458-9c25-4bcc-a7bd-0490641dec8e/resource/9665a448-6f2f-4872-99b9-bf12b5af84f4/download/userssharedsdfschoolimprovement2010grants.csv")

## Parsed with column specification:  
## cols(  
## `School Name` = col\_character(),  
## City = col\_character(),  
## State = col\_character(),  
## `District Name` = col\_character(),  
## `2010/11/Award Amount` = col\_character(),  
## `Model Selected` = col\_character(),  
## Location = col\_character()  
## )

head(data)

## # A tibble: 6 x 7  
## `School Name` City State `District Name` `2010/11/Award …  
## <chr> <chr> <chr> <chr> <chr>   
## 1 HOGARTH KING… SAVO… AK BERING STRAIT … $471014.00   
## 2 AKIACHAK SCH… AKIA… AK YUPIIT SCHOOL … $520579.00   
## 3 GAMBELL SCHO… GAMB… AK BERING STRAIT … $449592.00   
## 4 BURCHELL HIG… WASI… AK MATANUSKA-SUSI… $641184.00   
## 5 AKIAK SCHOOL AKIAK AK YUPIIT SCHOOL … $399686.00   
## 6 MIDVALLEY HI… WASI… AK MATANUSKA-SUSI… $697703.00   
## # … with 2 more variables: `Model Selected` <chr>, Location <chr>

Let’s use the glimpse function from the tidyverse just to see what the variables are, how many rows there are, the type of each variable, and the first few values for each variable.

glimpse(data)

## Observations: 831  
## Variables: 7  
## $ `School Name` <chr> "HOGARTH KINGEEKUK MEMORIAL SCHOOL", "AKI…  
## $ City <chr> "SAVOONGA", "AKIACHAK", "GAMBELL", "WASIL…  
## $ State <chr> "AK", "AK", "AK", "AK", "AK", "AK", "AK",…  
## $ `District Name` <chr> "BERING STRAIT SCHOOL DISTRICT", "YUPIIT …  
## $ `2010/11/Award Amount` <chr> "$471014.00", "$520579.00", "$449592.00",…  
## $ `Model Selected` <chr> "Transformation", "Transformation", "Tran…  
## $ Location <chr> "200 MAIN ST\nSAVOONGA, AK 99769\n(63.668…

We should notice that all of the variables here are being listed as the <chr> or character type. Let’s change a few of these. For some of the analysis that we’ll want to do, we want to treat the award amount as a number. We also want to treat variables like City, State, and Model as factors, or categoricals, which are variables with a limited number of possible values.

data$City <- as\_factor(data$City)  
data$State <- as\_factor(data$State)  
data$`Model Selected` <- as\_factor(data$`Model Selected`)  
glimpse(data)

## Observations: 831  
## Variables: 7  
## $ `School Name` <chr> "HOGARTH KINGEEKUK MEMORIAL SCHOOL", "AKI…  
## $ City <fct> SAVOONGA, AKIACHAK, GAMBELL, WASILLA, AKI…  
## $ State <fct> AK, AK, AK, AK, AK, AK, AK, AL, AL, AL, A…  
## $ `District Name` <chr> "BERING STRAIT SCHOOL DISTRICT", "YUPIIT …  
## $ `2010/11/Award Amount` <chr> "$471014.00", "$520579.00", "$449592.00",…  
## $ `Model Selected` <fct> Transformation, Transformation, Transform…  
## $ Location <chr> "200 MAIN ST\nSAVOONGA, AK 99769\n(63.668…

We can see that the three variables are now being treated as factors. Changing the type of the amout column is a bit more complicated. We need to remove the dollar sign, then we can change the value to a number.

data$`2010/11/Award Amount` <- as.numeric(gsub("\\$", "", data$`2010/11/Award Amount`))  
glimpse(data)

## Observations: 831  
## Variables: 7  
## $ `School Name` <chr> "HOGARTH KINGEEKUK MEMORIAL SCHOOL", "AKI…  
## $ City <fct> SAVOONGA, AKIACHAK, GAMBELL, WASILLA, AKI…  
## $ State <fct> AK, AK, AK, AK, AK, AK, AK, AL, AL, AL, A…  
## $ `District Name` <chr> "BERING STRAIT SCHOOL DISTRICT", "YUPIIT …  
## $ `2010/11/Award Amount` <dbl> 471014, 520579, 449592, 641184, 399686, 6…  
## $ `Model Selected` <fct> Transformation, Transformation, Transform…  
## $ Location <chr> "200 MAIN ST\nSAVOONGA, AK 99769\n(63.668…

Now that we have the data in the right types, let’s start getting a better sense of our data. First, we’ll use the summary function and see what it gives us.

summary(data)

## School Name City State District Name   
## Length:831 PHILADELPHIA: 29 CA : 92 Length:831   
## Class :character ST LOUIS : 18 FL : 71 Class :character   
## Mode :character MIAMI : 14 PA : 58 Mode :character   
## CLEVELAND : 12 TX : 48   
## JACKSONVILLE: 11 OH : 35   
## COLUMBUS : 11 MO : 32   
## (Other) :736 (Other):495   
## 2010/11/Award Amount Model Selected Location   
## Min. : 1 Transformation:608 Length:831   
## 1st Qu.:162541 Restart : 33 Class :character   
## Median :407794 Turnaround :168 Mode :character   
## Mean :420828 Closure : 16   
## 3rd Qu.:671142 NA's : 6   
## Max. :997852   
## NA's :74

We get some nice information here. We can see that R has actually counted the number of occurences of each city, state, and model, and seem to have provided the top most frequent of each of those. We’ll definitely check that. For our one numeric variable, the award amount, we get some basic descriptive statistics, such as min, max, mean, and a quartile breakdown. We can also see that there are 74 rows where there are NA values.

Let’s look a bit more at those factor variables. At the same, we’ll learn about the pipe operator, %>%.

data %>%   
 count(City)

## # A tibble: 424 x 2  
## City n  
## <fct> <int>  
## 1 SAVOONGA 1  
## 2 AKIACHAK 1  
## 3 GAMBELL 1  
## 4 WASILLA 2  
## 5 AKIAK 1  
## 6 TULUKSAK 1  
## 7 MONTGOMERY 4  
## 8 FORT DEPOSIT 1  
## 9 ROCKFORD 1  
## 10 HAYNEVILLE 2  
## # … with 414 more rows

By looking at the number of rows, we can see now that there are 424 different cities that received grants. Let’s order them to make sure we know which city received the most.

data %>%  
 count(City) %>%  
 arrange(desc(n))

## # A tibble: 424 x 2  
## City n  
## <fct> <int>  
## 1 PHILADELPHIA 29  
## 2 ST LOUIS 18  
## 3 MIAMI 14  
## 4 CLEVELAND 12  
## 5 JACKSONVILLE 11  
## 6 COLUMBUS 11  
## 7 KANSAS CITY 11  
## 8 MILWAUKEE 11  
## 9 SAN BERNARDINO 10  
## 10 SAN FRANCISCO 10  
## # … with 414 more rows

Let’s do the same thing with states and the model.

data %>%  
 count(State) %>%  
 arrange(desc(n))

## # A tibble: 50 x 2  
## State n  
## <fct> <int>  
## 1 CA 92  
## 2 FL 71  
## 3 PA 58  
## 4 TX 48  
## 5 OH 35  
## 6 MO 32  
## 7 MI 28  
## 8 GA 26  
## 9 NY 25  
## 10 NC 24  
## # … with 40 more rows

data %>%  
 count(`Model Selected`) %>%  
 arrange(desc(n))

## Warning: Factor `Model Selected` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`

## # A tibble: 5 x 2  
## `Model Selected` n  
## <fct> <int>  
## 1 Transformation 608  
## 2 Turnaround 168  
## 3 Restart 33  
## 4 Closure 16  
## 5 <NA> 6

R gives us the information we’re wanting here, but it also gives us a warning about having NA values that are simply missing. It could be better for us as we’re working with our data to have an explicit value that we know stands for missing or otherwise NA values.

data$`Model Selected` <- fct\_explicit\_na(data$`Model Selected`, na\_level = "Missing")  
levels(data$`Model Selected`)

## [1] "Transformation" "Restart" "Turnaround" "Closure"   
## [5] "Missing"

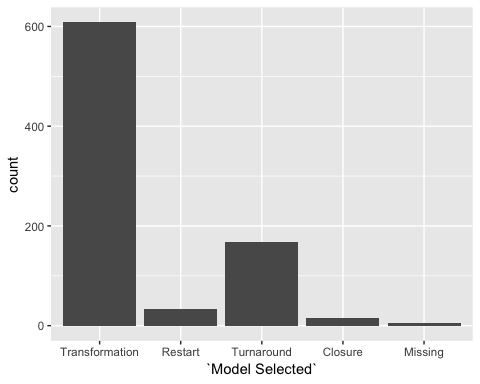
Let’s rerun our counting code for the model variable.

data %>%  
 count(`Model Selected`) %>%  
 arrange(desc(n))

## # A tibble: 5 x 2  
## `Model Selected` n  
## <fct> <int>  
## 1 Transformation 608  
## 2 Turnaround 168  
## 3 Restart 33  
## 4 Closure 16  
## 5 Missing 6

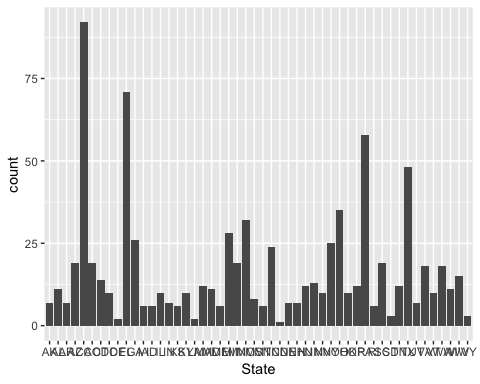
We’re no longer getting any warning or suggestion since we’ve made the NA values explicit. What if we wanted to quickly graph how many grants were given for each model type? We’ll use ggplot, which is the Tidyverse library for graphing that has become fairly standard in R.

ggplot(data, aes(`Model Selected`)) +  
 geom\_bar()



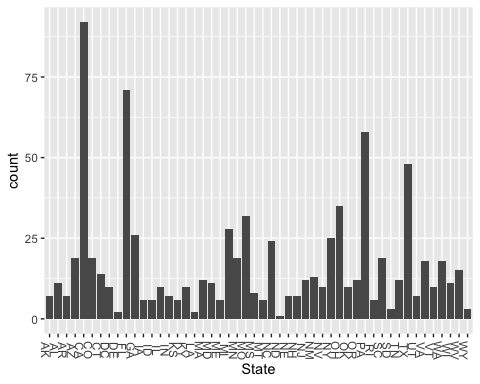
Visually representing this data increases the impact of the difference between model types. Many more schools went with the mildest type of change model than any other. We could do the same thing for the State variable.

ggplot(data, aes(State)) +  
 geom\_bar()



We can see that it automatically ordered the levels by alphabetical order, which is great. But, the state abbreviations are crammed together, and it’s hard to read them. Let’s fix that by changing the orientation of the labels.

ggplot(data, aes(State)) +  
 geom\_bar() +  
 theme(axis.text.x = element\_text(angle = -90))



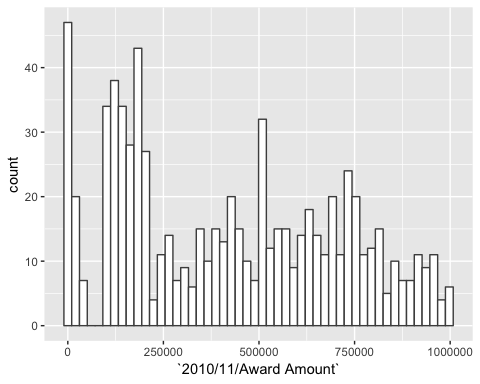
Now that we’ve gotten a bit of experience with categorical or factor variables, let’s switch over and take a look at our one numeric variable, the award amount.

summary(data$`2010/11/Award Amount`)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 1 162541 407794 420828 671142 997852 74

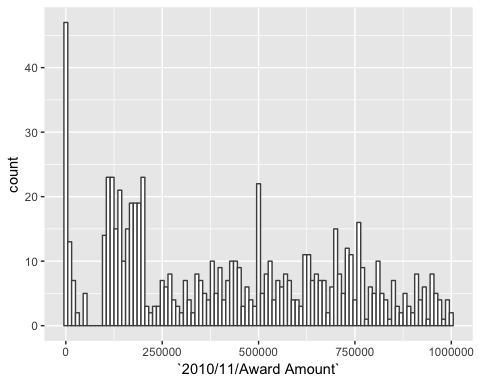
Let’s try to get a sense of the distribution of this data. Do more schools get smaller amounts of money? Do more schools get larger? We’ll do this with a histogram visualization.

ggplot(data, aes(`2010/11/Award Amount`)) +  
 geom\_histogram(bins = 50, na.rm = TRUE, fill = "white", color = "grey30")



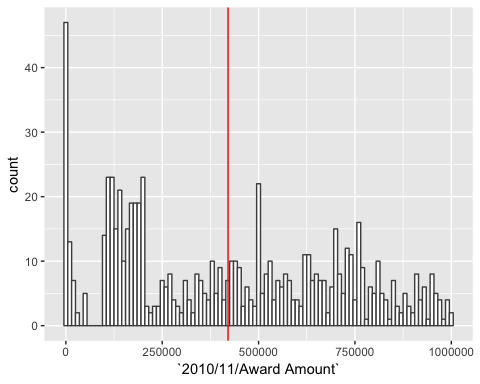
We can see that generally, higher numbers of schools received smaller awards, though there are a couple of other peaks in the distribution. We also see that there are some strange low values that we might want to look into it. Before we do that, let’s look at how our figure changes if we group not by a number of bins but by defining the width of each bin instead.

ggplot(data, aes(`2010/11/Award Amount`)) +  
 geom\_histogram(binwidth = 10000, na.rm = TRUE, fill = "white", color = "grey30")



Why don’t we add a line to mark the mean so that we can see roughly how much of the distribution falls above or below.

ggplot(data, aes(`2010/11/Award Amount`)) +  
 geom\_histogram(binwidth = 10000, na.rm = TRUE, fill = "white", color = "grey30") +  
 geom\_vline(aes(xintercept=mean(data$`2010/11/Award Amount`, na.rm = TRUE)), color = "red")



Before we move on, let’s see if we can figure out what’s going on with some of those low values.

data %>%   
 arrange(`2010/11/Award Amount`)

## # A tibble: 831 x 7  
## `School Name` City State `District Name` `2010/11/Award …  
## <chr> <fct> <fct> <chr> <dbl>  
## 1 EDISON HS/FA… PHIL… PA PHILADELPHIA C… 1  
## 2 PACIFIC HIGH SAN … CA SAN BERNARDINO… 2  
## 3 SAN GORGONIO… SAN … CA SAN BERNARDINO… 2  
## 4 BARTON ELEME… SAN … CA SAN BERNARDINO… 2  
## 5 SHANDIN HILL… SAN … CA SAN BERNARDINO… 2  
## 6 NORTH HIGH S… DES … IA DES MOINES IND… 2  
## 7 CROWNPOINT H… CROW… NM GALLUP-MCKINLE… 2  
## 8 ERNIE PYLE M… ALBU… NM ALBUQUERQUE PU… 2  
## 9 SOUTH PARK H… BUFF… NY BUFFALO CITY S… 2  
## 10 ROOSEVELT HI… YONK… NY YONKERS CITY S… 2  
## # … with 821 more rows, and 2 more variables: `Model Selected` <fct>,  
## # Location <chr>

We can see that some of the grants were for only one or two dollars. We can go back and look at the original csv file to verify that we didn’t alter that. To figure out why those grants were so small, or to figure out whether these are erroneous values in the original data, we would have to look into the study a bit more.

We’ve developed a bit of a sense of how to explore our data, especially with plots. Let’s start thinking about what types of questions we might ask our data and how we could find solutions with R.

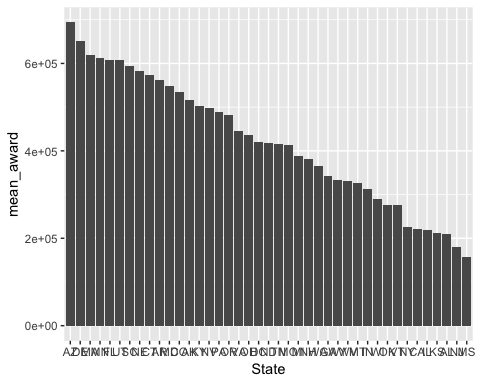
What if we wonder whether there is some set of states that might have received more money on average than other states?

This takes a few steps. We need to group all the observations together by state, average their award amounts, then sort the values. If we want to graph the data, which we will, in this descending order, we also have to do something a bit tricky to reorder the states as factors.

state\_award <- data %>%  
 group\_by(State) %>%  
 summarise(mean\_award = mean(`2010/11/Award Amount`)) %>%  
 arrange(desc(mean\_award)) %>%  
 mutate(State = factor(State, unique(State)))  
  
head(state\_award)

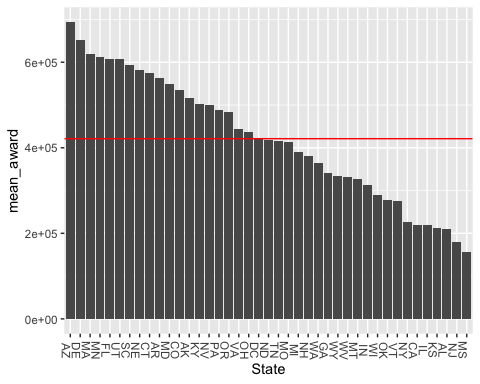
## # A tibble: 6 x 2  
## State mean\_award  
## <fct> <dbl>  
## 1 AZ 693755   
## 2 DE 651608   
## 3 MA 620106.  
## 4 MN 612284.  
## 5 FL 607590.  
## 6 UT 607375

state\_award %>%  
 drop\_na(mean\_award) %>%  
 ggplot(aes(x = State, y = mean\_award)) +  
 geom\_col()



And let’s add a mean line.

state\_award %>%  
 drop\_na(mean\_award) %>%  
 ggplot(aes(x = State, y = mean\_award)) +  
 geom\_col() +  
 geom\_hline(aes(yintercept=mean(state\_award$mean\_award, na.rm = TRUE)), color = "red") +  
 theme(axis.text.x = element\_text(angle = -90))



What if we want to just get a list of States that received on average more than the average across all states?

mean\_state <- mean(state\_award$mean\_award, na.rm = TRUE)  
  
state\_award %>%  
 filter(mean\_award >= mean\_state)

## # A tibble: 19 x 2  
## State mean\_award  
## <fct> <dbl>  
## 1 AZ 693755   
## 2 DE 651608   
## 3 MA 620106.  
## 4 MN 612284.  
## 5 FL 607590.  
## 6 UT 607375   
## 7 SC 593300.  
## 8 NE 582583.  
## 9 CT 574367.  
## 10 AR 562301.  
## 11 MD 548507   
## 12 CO 534644.  
## 13 AK 515345.  
## 14 KY 502594.  
## 15 NV 499157.  
## 16 PA 488245.  
## 17 OR 482669   
## 18 VA 444944.  
## 19 OH 436194.

What if we wanted to see whether awards were larger or smaller based on the model?

data %>%  
 group\_by(`Model Selected`) %>%  
 summarise(mean\_award = mean(`2010/11/Award Amount`, na.rm = TRUE)) %>%  
 arrange(desc(mean\_award))

## # A tibble: 5 x 2  
## `Model Selected` mean\_award  
## <fct> <dbl>  
## 1 Restart 503412.  
## 2 Transformation 434116.  
## 3 Turnaround 394101.  
## 4 Closure 75156.  
## 5 Missing NaN

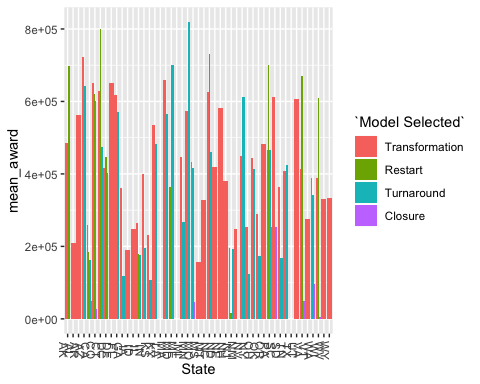
We can see, and should expect, that when you’re just closing the school and redistributing students, you receive less money on average. When you restart the school, or close it and reopen it, you get the largest award on average.

What if we want to do something a bit more complex, and check to see if this sort of average by model varies across states?

state\_model\_avgs <- data %>%  
 group\_by(State, `Model Selected`) %>%  
 summarise(mean\_award = mean(`2010/11/Award Amount`, na.rm = TRUE))  
  
glimpse(state\_model\_avgs)

## Observations: 98  
## Variables: 3  
## Groups: State [50]  
## $ State <fct> AK, AK, AL, AR, AZ, AZ, CA, CA, CA, CA, CO, CO,…  
## $ `Model Selected` <fct> Transformation, Restart, Transformation, Transf…  
## $ mean\_award <dbl> 484952.17, 697703.00, 209162.00, 562301.29, 723…

state\_model\_avgs %>%  
 filter(`Model Selected` != "Missing") %>%  
 ggplot(aes(x = State, y = mean\_award, fill = `Model Selected`)) +  
 geom\_col(position = "dodge", na.rm = TRUE) +  
 theme(axis.text.x = element\_text(angle = -90))



Let’s look at a single state: CO:

state\_model\_avgs %>%  
 filter(State == "CO")

## # A tibble: 4 x 3  
## # Groups: State [1]  
## State `Model Selected` mean\_award  
## <fct> <fct> <dbl>  
## 1 CO Transformation 651243.  
## 2 CO Restart 619609   
## 3 CO Turnaround 599881.  
## 4 CO Closure 26052.

Let’s find out which type of model in which state on average got the largest award.

state\_model\_avgs %>%  
 arrange(desc(mean\_award))

## # A tibble: 98 x 3  
## # Groups: State [50]  
## State `Model Selected` mean\_award  
## <fct> <fct> <dbl>  
## 1 MN Turnaround 818851.  
## 2 CT Restart 800000   
## 3 NC Restart 730689   
## 4 AZ Transformation 723803.  
## 5 MD Turnaround 701810.  
## 6 PA Restart 701497.  
## 7 AK Restart 697703   
## 8 VA Restart 669661   
## 9 MA Transformation 659791.  
## 10 DE Transformation 651608   
## # … with 88 more rows

What if instead of averages, just count the number of each type per state?

state\_model\_counts <- data %>%  
 group\_by(State, `Model Selected`) %>%  
 count()  
  
state\_model\_counts

## # A tibble: 98 x 3  
## # Groups: State, Model Selected [98]  
## State `Model Selected` n  
## <fct> <fct> <int>  
## 1 AK Transformation 6  
## 2 AK Restart 1  
## 3 AL Transformation 11  
## 4 AR Transformation 7  
## 5 AZ Transformation 12  
## 6 AZ Turnaround 7  
## 7 CA Transformation 56  
## 8 CA Restart 5  
## 9 CA Turnaround 29  
## 10 CA Closure 2  
## # … with 88 more rows

With this data we could then look at distributions for particular models across states.

state\_model\_counts %>%  
 filter(`Model Selected` == "Turnaround") %>%  
 arrange(desc(n))

## # A tibble: 29 x 3  
## # Groups: State, Model Selected [29]  
## State `Model Selected` n  
## <fct> <fct> <int>  
## 1 CA Turnaround 29  
## 2 FL Turnaround 17  
## 3 MO Turnaround 14  
## 4 MI Turnaround 9  
## 5 OH Turnaround 8  
## 6 AZ Turnaround 7  
## 7 CO Turnaround 6  
## 8 CT Turnaround 6  
## 9 KY Turnaround 6  
## 10 MD Turnaround 6  
## # … with 19 more rows

We can see that California received the largest number of turnaround grants, with Florida next, though a ways behind them in number.