Econ 106: Data Analyis for Economics

Lecture 8

slides adapted from: https://r4ds.had.co.nz/tidy-data.html

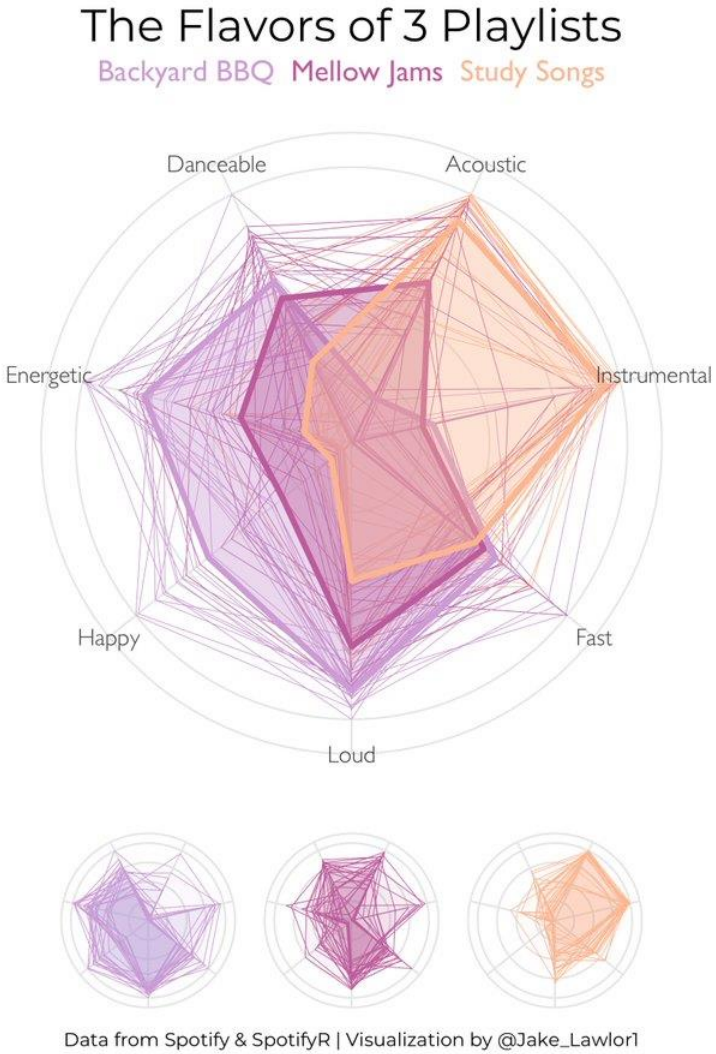
Reminders

- Lab 2 is due Sunday, 11:59pm
- Please plan on finalizing your data selection for your research project by this weekend (reach out to me if you have any questions)
- I will review your proposed datasets if you post it here by Friday 5pm

https://pollev.com/vsovero

#tidytuesday

- time for some study music (midterms are coming up)

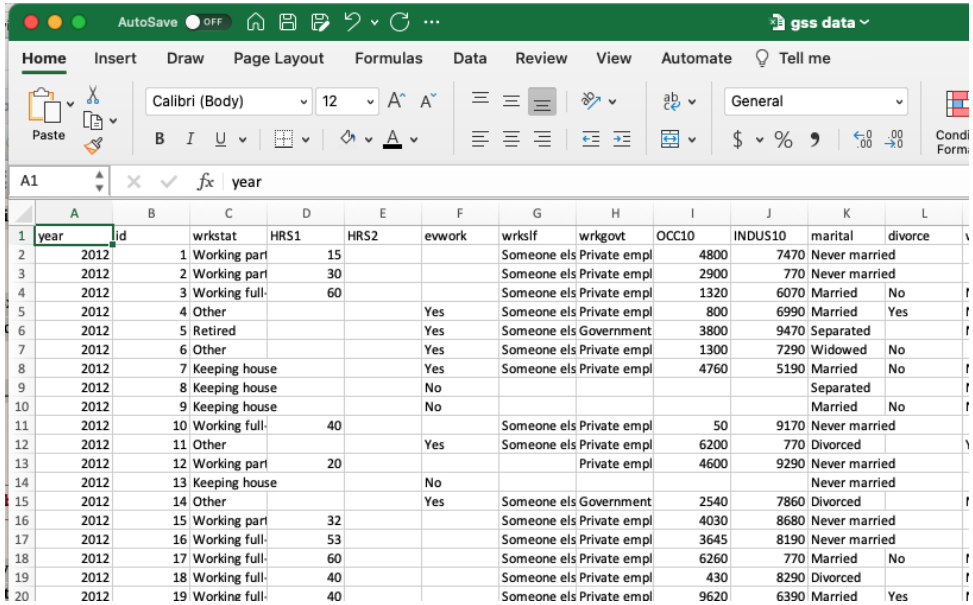


Outline for Today

- reading in “real” data
- cleaning categorical variables:
  - converting type from character to factor
  - collapsing levels

Loading Different Types of Data

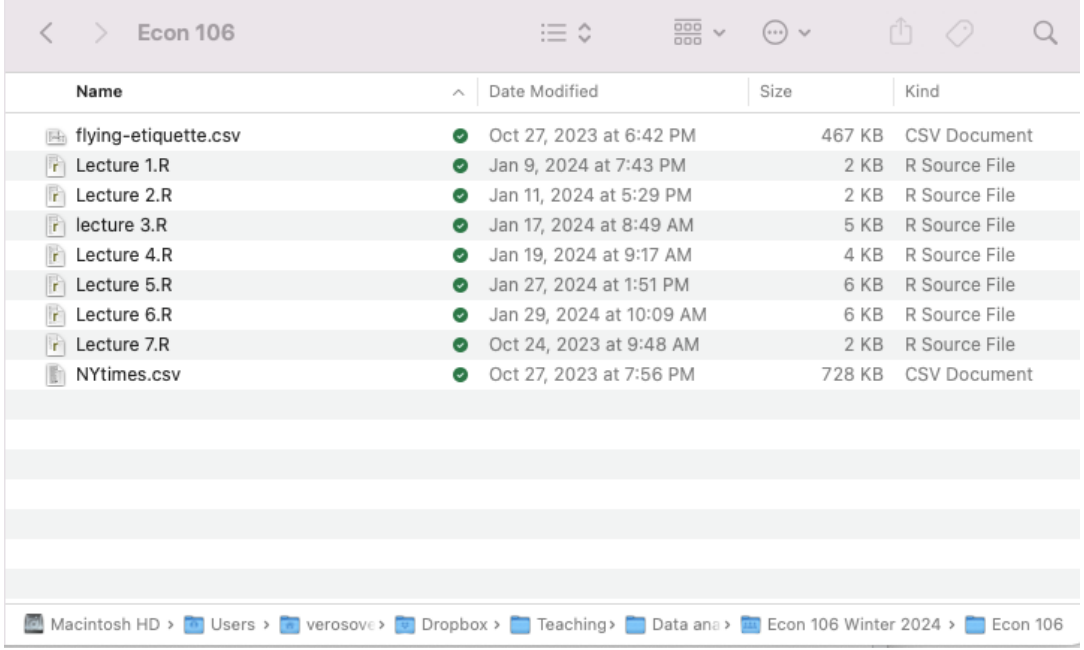
- Real data isn't going to come pre-loaded into a package
- It will typically come from excel, csv files
- We will need to import it into Rstudio



year	id	wrkstat	HRS1	HRS2	evwork	wrkslf	wrkgovt	OCC10	INDUS10	marital	divorce
2012	1	Working part	15			Someone els	Private empl	4800	7470	Never married	
2012	2	Working part	30			Someone els	Private empl	2900	770	Never married	
2012	3	Working full-	60			Someone els	Private empl	1320	6070	Married	No
2012	4	Other			Yes	Someone els	Private empl	800	6990	Married	Yes
2012	5	Retired			Yes	Someone els	Government	3800	9470	Separated	
2012	6	Other			Yes	Someone els	Private empl	1300	7290	Widowed	No
2012	7	Keeping house			Yes	Someone els	Private empl	4760	5190	Married	No
2012	8	Keeping house			No					Separated	
2012	9	Keeping house			No					Married	No
2012	10	Working full-	40			Someone els	Private empl	50	9170	Never married	
2012	11	Other			Yes	Someone els	Private empl	6200	770	Divorced	
2012	12	Working part	20				Private empl	4600	9290	Never married	
2012	13	Keeping house			No					Never married	
2012	14	Other			Yes	Someone els	Government	2540	7860	Divorced	
2012	15	Working part	32			Someone els	Private empl	4030	8680	Never married	
2012	16	Working full-	53			Someone els	Private empl	3645	8190	Never married	
2012	17	Working full-	60			Someone els	Private empl	6260	770	Married	No
2012	18	Working full-	40			Someone els	Private empl	430	8290	Divorced	
2012	19	Working full-	40			Someone els	Private empl	9620	6390	Married	Yes

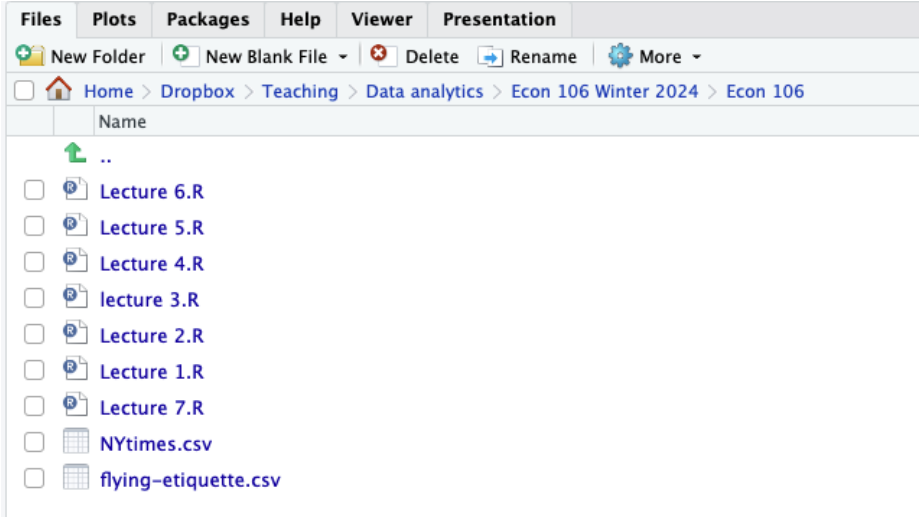
Step 1: Put your csv file in your Econ 106 folder

- Create an Econ 106 folder on your computer if you haven't already done so
- Then move the csv file to the folder



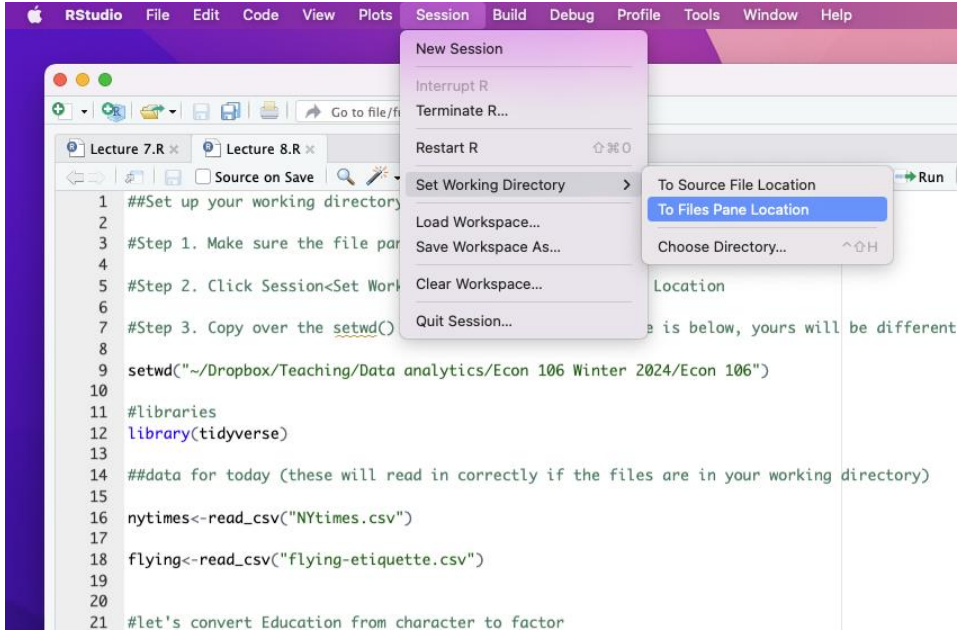
Step 2: Set your file pane to your 106 folder

- In R studio, navigate to your 106 folder in the file pane window



Step 3: Set Working Directory

- Set the working directory to the location of the data file:
  - click Session
  - click Set Working Directory



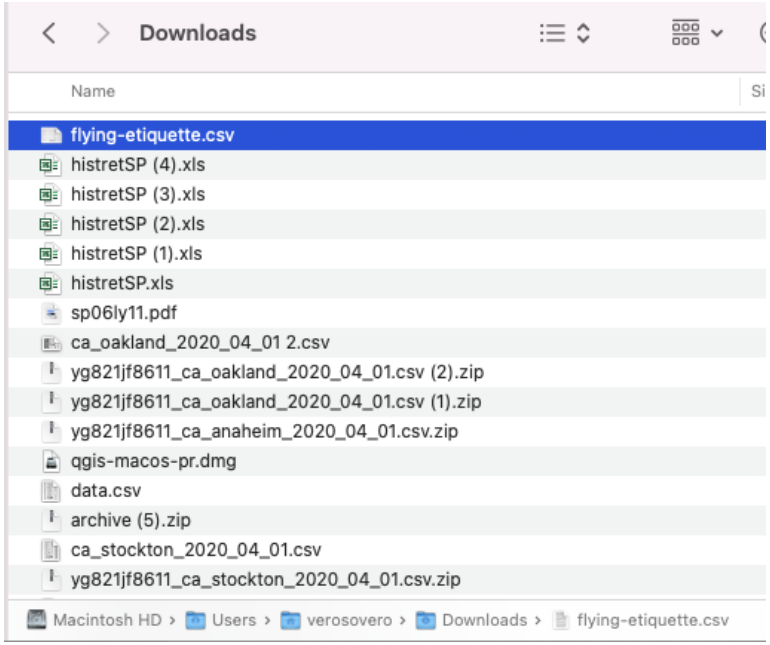
Step 4: Use read_csv() function

- Important: this only works if you placed the file in your working directory
- Otherwise you will have to figure out and write the whole path name (next slide)

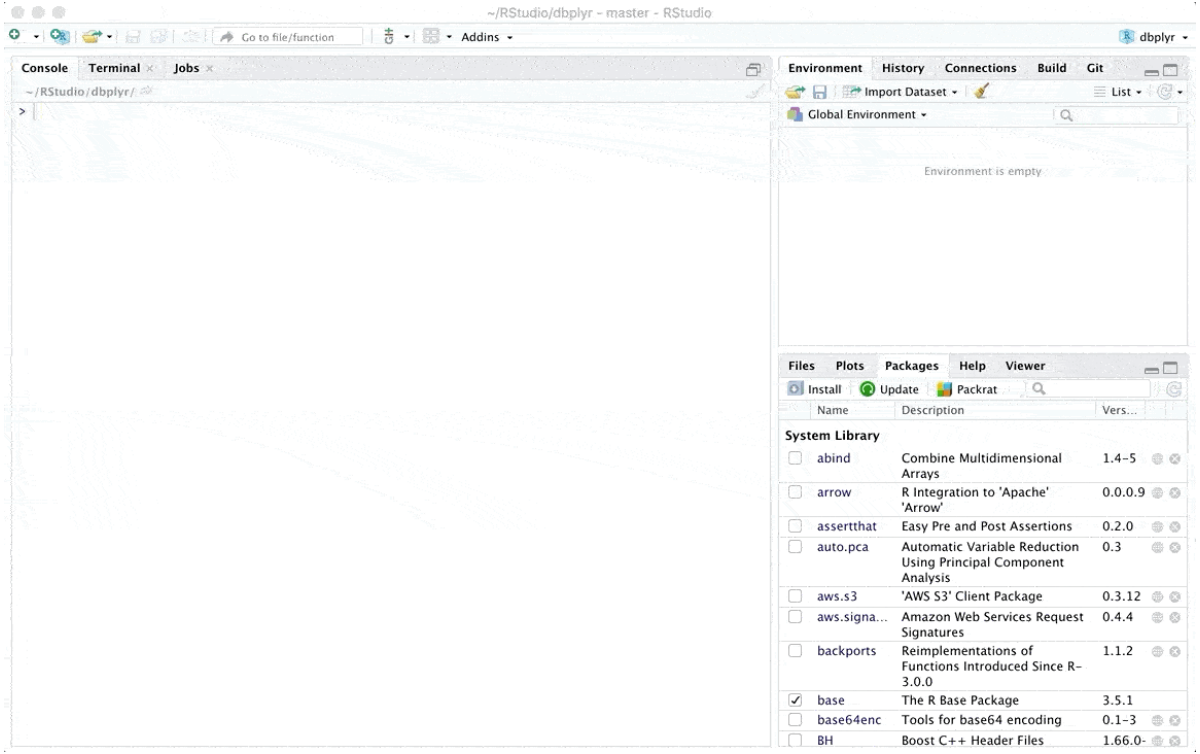
```
flying<-read_csv("flying-etiquette.csv”)
```

The other (not recommended) way

- somehow your csv file ends up in another folder
- There is a way to get it loaded into R using some point and click options



Point and click Alternative



https://support.posit.co/hc/en-us/articles/218611977-Importing-Data-with-the-RStudio-IDE

Converting Vector Type

- as.numeric(): your values should be numbers
- as.character(): your values can be numbers or characters
- factor(): your values are categorical (finite set of values)

Categorical Variables as Factor Variables

- Currently, Education is stored as character type
- If we convert it to a factor variable, we will be able to:
  - change the order of the levels
  - change the values of the levels

```
$ Gender                    : chr [1:1040] NA "Male" "Male" "Male…
$ Age                       : chr [1:1040] NA "30-44" "30-44" "30…
$ Household Income          : chr [1:1040] NA NA "$100,000 - $149…
$ Education                 : chr [1:1040] NA "Graduate degree" "…
$ Location (Census Region)  : chr [1:1040] NA "Pacific" "Pacific"…
```

factor()

Arguments:

- the variable you are converting to factor

```
flying_factor<-flying%>%
        mutate(Education_f=factor(Education))
```

Check the Current Levels

- When Education is a factor variable, we have some Base R options for examining the levels

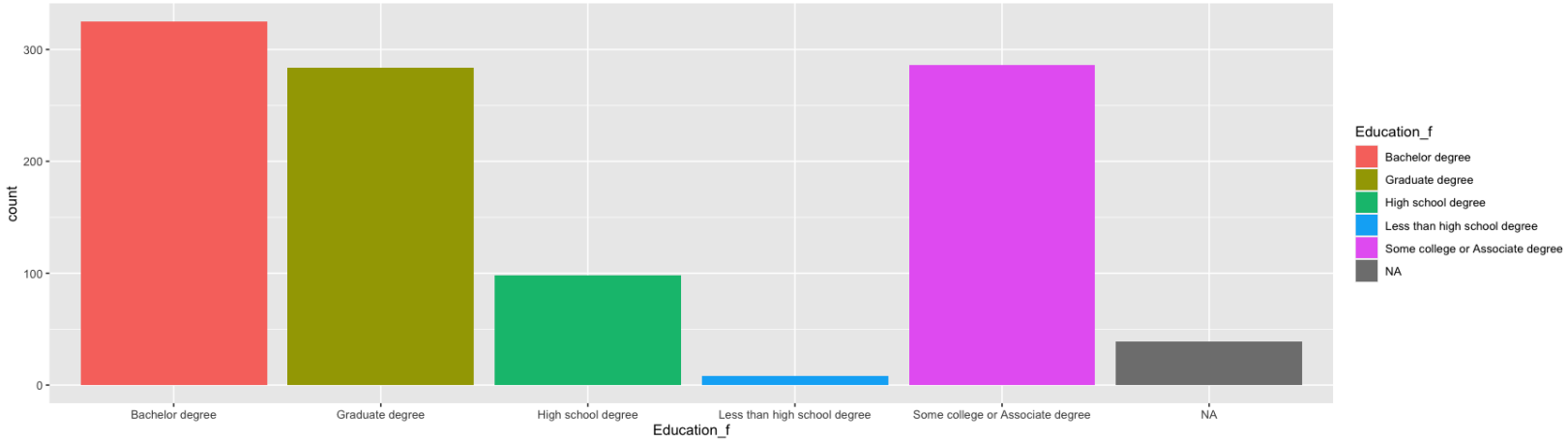
```
levels(flying_factor$Education_f)

summary(flying_factor$Education_f)

nlevels(flying_factor$Education_f)
```

https://pollev.com/vsovero

Order of levels

https://pollev.com/vsovero

- levels will be sorted alphabetically
- This doesn’t look great when you are plotting your data (order should go from most to least educational attainment)



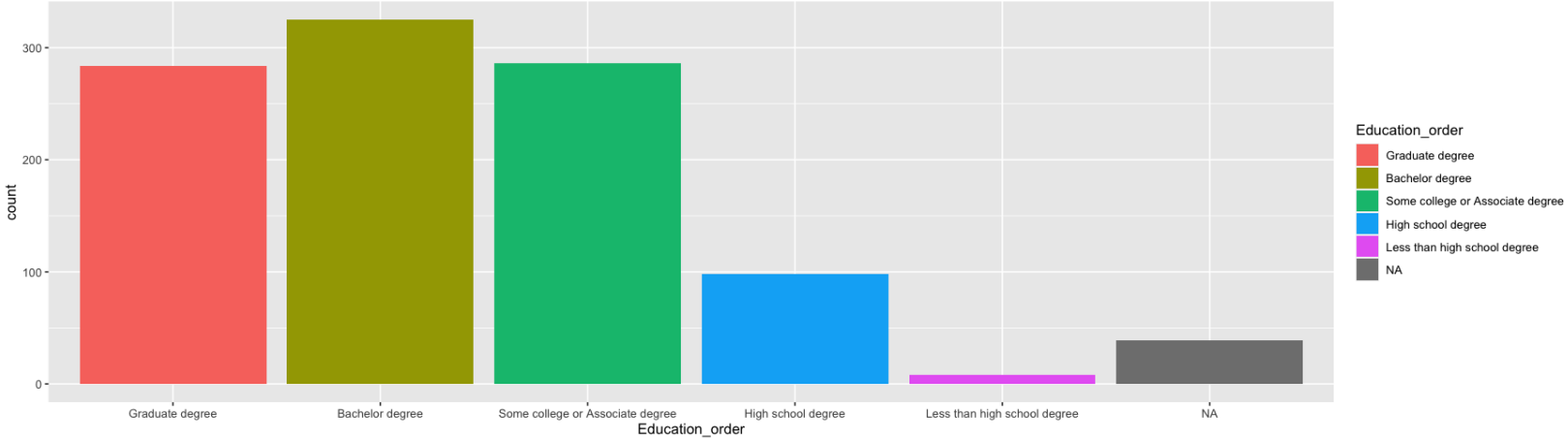
Ordering levels

- you can set the order of the levels yourself using the levels argument

```
flying_factor_ordered<-flying%>%
        mutate(Education_order=factor(Education,
                levels=c("Graduate degree", "Bachelor degree", "Some college or Associate degree",
                        "High school degree", "Less than high school degree" )))
```

Ordered Levels

- Now our visualizations make more sense (order goes from most to least educational attainment)



Exercise

- convert `Do you ever recline your seat when you fly?` into a factor variable
- Order the levels from Always to Never
- Make a bar graph of the frequencies

What’s in a (sur)Name?

- Why do women choose to retain their surname at marriage?
- Claudia Goldin (Nobel Laureate) has a thing or two to say about it:
  - Women’s educational attainment has been increasing over time
  - Women are waiting longer to marry
  - They may have “made a name for themselves” in the profession by the time they marry
  - Less likely to change surname to not lose the professional reputation they have built for themselves

Making a Name: Women's Surnames at Marriage and Beyond (Goldin and Shim 2004)

https://scholar.harvard.edu/goldin/files/making_a_name_womens_surnames_at_marriage_and_beyond.pdf

- Used Nytimes wedding announcements to track trends in surname retention over time

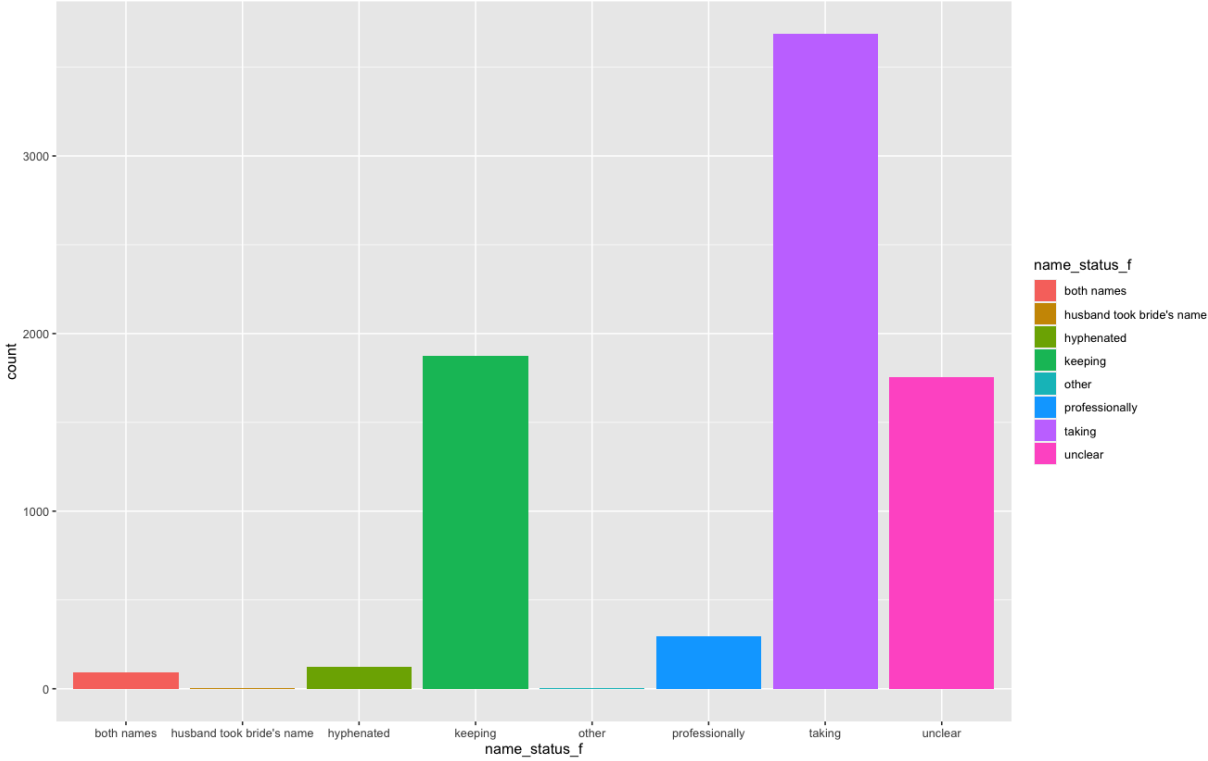
Nytimes Example

```
nytimes_factor<-nytimes%>%
        mutate(name_status_f=factor(name_status))
```

```
nytime_factor                7835 obs. of 7 variables
    $ url          : chr [1:7835] "http://www.nytimes.com/1985/12/01/style/myra-cohen-becomes-bride
    $ name_status  : chr [1:7835] "keeping" "taking" "taking" "keeping" ...
    $ bride_age    : num [1:7835] NA NA NA NA NA NA NA NA NA NA ...
    $ groom_age    : num [1:7835] NA NA NA NA NA NA NA NA NA NA ...
    $ date         : Date[1:7835], format: "1985-12-01" "1985-12-22" "1985-11-03" "1985-11-10" ...
    $ year         : num [1:7835] 1985 1985 1985 1985 1985 ...
    $ name_status_f: Factor w/ 8 levels "both names","husband took bride's name",..: 4 7 7 4 4 7 8
```

Nytimes Example

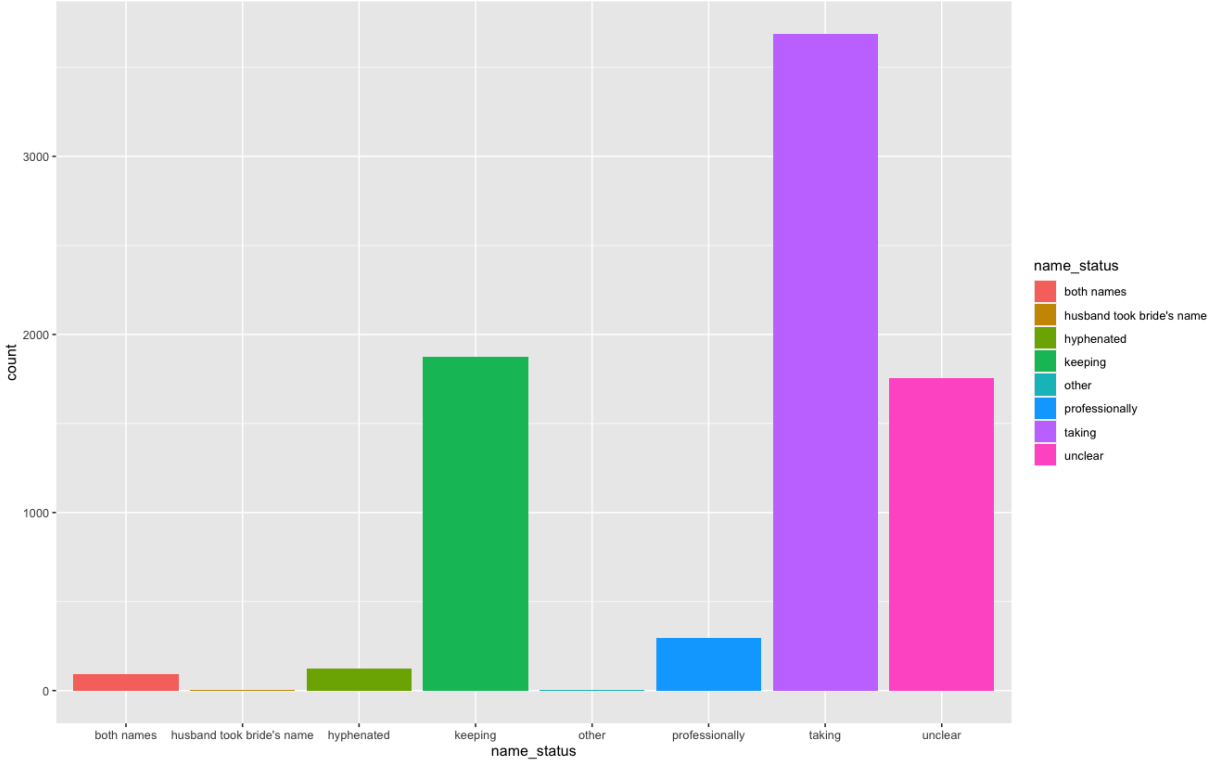
```
ggplot(data=nytimes_factor,
      mapping=aes(x=name_status_f))
  geom_bar(aes(fill=name_status_f))
```



Name Change Status

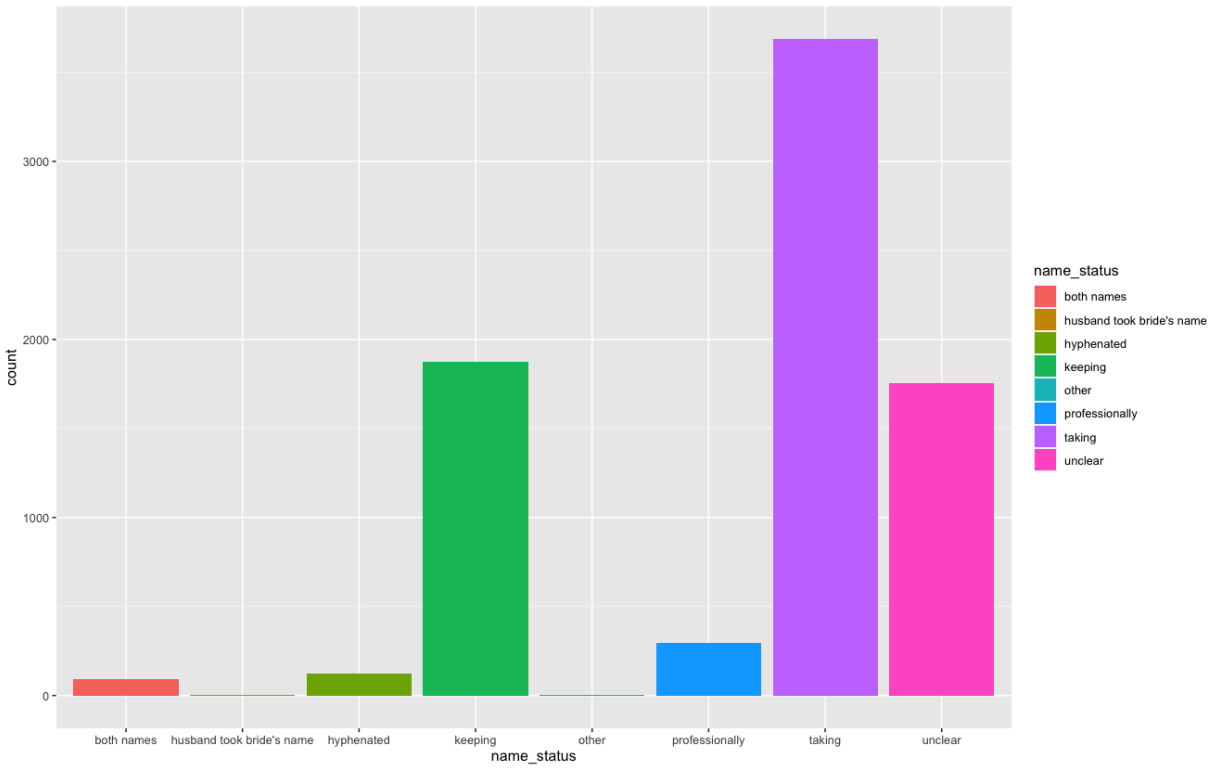
Goldin and Shim (2004):

- brides are coded as “keepers” if they stated they would retain their surnames socially and/or professionally
- All others are deemed “changers”—those taking the groom’s surname, those hyphenating their names and those for whom no information is given



Nytimes wedding announcements

How do we recategorize name_status to match the approach taken by Goldin and Shim (2004)?



Changing the Levels of a Factor Variable

Current Levels:

- "both names"
- "husband took bride's name" "hyphenated "
- "keeping"
- "other"
- "professionally"
- "taking "
- "unclear"

Desired Levels:

- "Keepers"
- "Takers"

Mapping the Current Levels to New Levels

Current Levels:

- "both names"
- "husband took bride's name"
- "hyphenated"
- "keeping"
- "other"
- "professionally"
- "taking"
- "unclear"

Desired Levels:

- "Keeper"
- "Changer"

Mapping the Current Levels to New Levels

Current Levels:

- "keeping"
- "husband took bride's name"
- "professionally"
- "both names"
- "hyphenated"
- "other"
- "taking"
- "unclear"

Desired Levels:

- " Keeper"
- " Changer "

fct_recode()

From the forcats package inside the tidyverse

Arguments:

1. the factor variable you are recoding
2. the names of the new levels
3. the old levels you want to assign to the new levels

```
nytimes_recode<-nytimes_factor%>%
        mutate(name_status_recode=fct_recode(name_status_f,
                "Keeper" ="husband took bride's name",
                "Keeper" ="keeping",
                "Keeper" ="professionally",
               "Changer" ="both names",
                "Changer" ="hyphenated",
                "Changer" ="other",
                "Changer" ="taking",
                "Changer" ="unclear"))
```

fct_recode()

```
nytimes_recode<-nytimes_factor%>%
        mutate(name_status_recode=fct_recode(name_status_f,
                "Keeper" ="husband took bride's name",
                "Keeper" ="keeping",
                "Keeper" ="professionally",
               "Changer" ="both names",
                "Changer" ="hyphenated",
                "Changer" ="other",
                "Changer" ="taking",
                "Changer" ="unclear"))
```

name_status_f	name_status_recode
keeping	Keeper
taking	Changer
taking	Changer
keeping	Keeper
keeping	Keeper
taking	Changer
unclear	Changer
keeping	Keeper
keeping	Keeper
taking	Changer
taking	Changer
keeping	Keeper
keeping	Keeper
unclear	Changer

fct_collapse()

- useful when you want to combine (collapse) a lot of levels together
- You list all the levels that will go into “Keeper”, “Changer”
- Any thing not listed in the collapse will remain as its own level

```
nytimes_collapse<-nytimes_factor%>%
        mutate(surname=fct_collapse(name_status_f,
                "Keeper" =c("husband took bride's name", "keeping","professionally"),
                "Changer" =c("both names","hyphenated","other","taking", "unclear")))
```

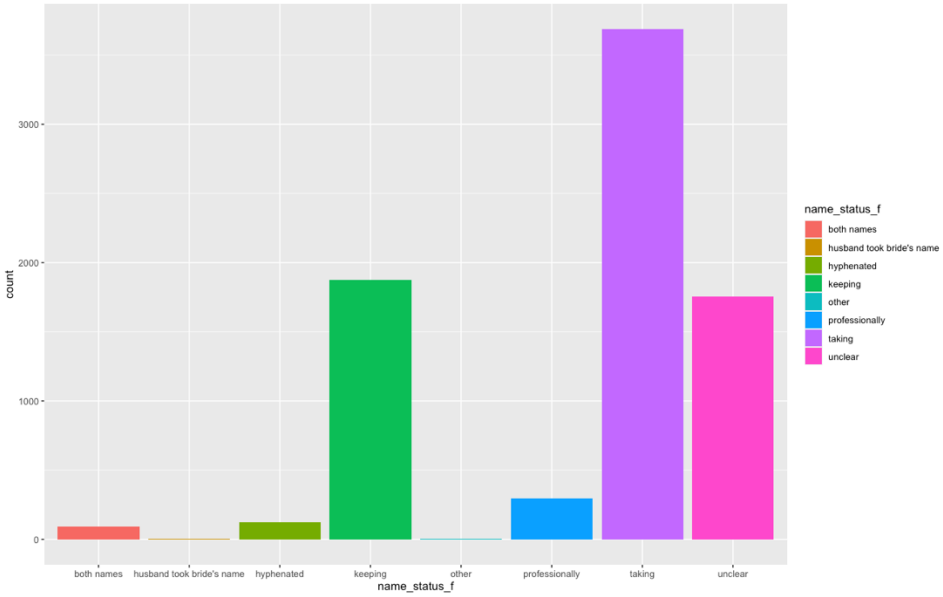
Exercise

- collapse levels of `How often do you travel by plane?` that are once a month or more frequent
- Call the new level “monthly”
- Order the levels from highest to lowest frequency of travel

https://pollev.com/vsovero

fct_lump()

- Have a lot of levels that you don't really care about and want to lump together? This is the function for you
- lumps all of the smaller levels together into a single "other" level



fct_lump()

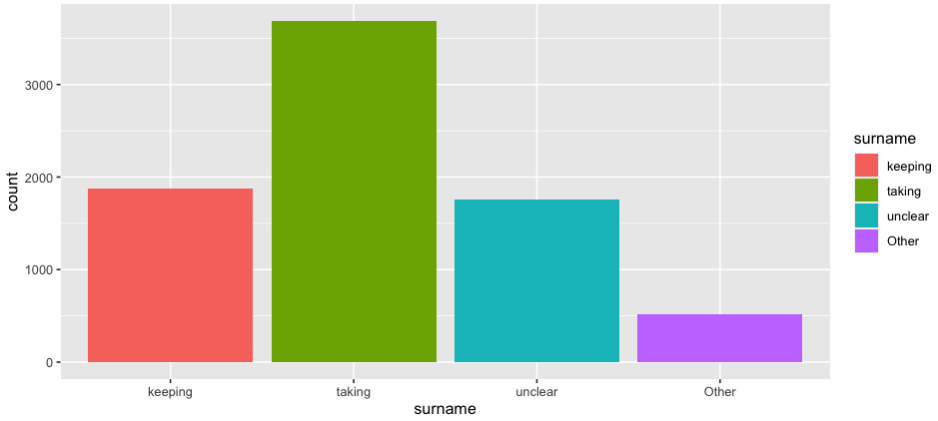
Arguments:

1. the factor variable you are recoding
2. the number of groups (excluding other) that you want to keep

```
nytimes_lump<-nytimes_factor%>%
    mutate(surname=fct_lump(name_status_f, n=3))
```

fct_lump()

```
nytimes_lump<-nytimes_factor%>%
        mutate(surname=fct_lump(name_status_f,
n=3))

ggplot(data=nytimes_lump,
    mapping=aes(x=surname)) +
  geom_bar(aes(fill=surname))
```



https://pollev.com/vsovero

Exercise

- create a new factor variable that groups everything but the two most frequent levels of `How often do you travel by plane?`
- What level is now in “Other”?

https://pollev.com/vsovero

Creating Factors from Numeric Variables

- We can also convert numeric variables into Factors
- we break up the numeric variable into intervals

bride_age	groom_age	bride_age_f
25	27	[20,30)
29	32	[20,30)
31	38	[30,40)
25	27	[20,30)
27	26	[20,30)
30	29	[30,40)
31	29	[30,40)
28	31	[20,30)
32	34	[30,40)
25	27	[20,30)

https://bookdown.org/rwnahhas/IntroToR/convert-numeric-to-categorical-by-binning.html

cut() : choose your own ranges

Arguments:

- the name of the numeric variable
- breaks: the end points of the interval ranges
- include.lowest : whether you want to include the low value (strict inequality)
- right: whether you want to include the high value (strict inequality)

```
nytime_age_cut<-nytimes%>%
mutate(bride_age_f = cut(bride_age, breaks = c(20, 30, 40, 50, 60, 70, 80, 90),
                         include.lowest = T,
                         right = F))
```

How do you know where to cut?

- I always use summary() to first check the summary statistics (min, max)

```
summary(nytimes$bride_age)
```

cut_interval() : equally sized intervals

Arguments:

- the name of the numeric variable
- length: the size of the interval
- right: whether you want to include the high value (strict inequality)

```
nytime_age_interval<-nytimes%>%
mutate(bride_age_f = cut_interval(bride_age,
                                  length = 10,
                                  right = F))
```