Week 3: Summarizing Data II

Wenhao Jiang

Department of Sociology
New York University

September 23, 2022

Data

Data Structures

- R also has a number of basic data *structures*.
- A data structure is either
  - homogeneous (all elements are of the same data type)
  - heterogeneous (elements can be of more than one data type).

Dimension	**Homogeneous**	**Heterogeneous**
1	Vector	List
2	Matrix	Data Frame
3+	Array	

Summarize Data (Basics)

Read Data

- A **data frame** is the **most** common way that we store and interact with data

```
## set working directory
setwd("~/Dropbox/Teaching/SOCUA-302/Week 2")

## read the file
gss <- read.csv("GSS_SOCUA_W2.csv")
```

Combine data subsetting and summarizing (base R)

- Some values in the variable `sibs` are not real observations
- Negative values are missingness
- How do we calculate the mean value of `sibs` without negative `sibs` values?

```r
mean(gss[gss$sibs>=0,"sibs"])
```

```
## [1] 3.863355
```

Exercise

- How do we calculate the variance of `sibs` in year 2018 only?

Combine data subsetting and summarizing (dplyr)

- We can also use `dplyr` to subset and summarize data
    - Remember `filter()` filters rows and `select()` select columns
- How do we calculate the mean value of `sibs` without negative `sibs` values?

```
library(dplyr)
gss %>% filter(sibs>0) %>% summarize(sibs_mean = mean(sibs))
```

```
##   sibs_mean
## 1  4.066925
```

- `summarize()` follows the format: `summarize(your_summarize_name = function(variable))`

Combine data subsetting and summarizing (dplyr)

- How do we calculate the mean value of `sibs` without negative `sibs` values?
- Why bothering using `dplyr` and `summarize`?
  - Because we can summarize data using different functions at the same time

```
gss %>% filter(sibs>0) %>% summarize(sibs_mean = mean(sibs),
                                     sibs_var = var(sibs))
```

```
##   sibs_mean sibs_var
## 1  4.066925 9.620029
```

Export the data

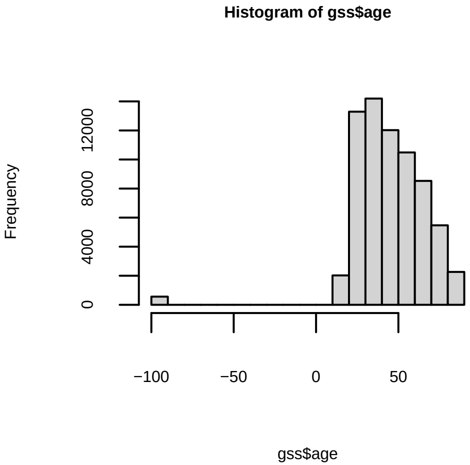
- `write.csv` save (or export) the dataframe in `.csv` format.

```
write.csv(gss, "cleaned_gss.csv")
```

Histogram plot

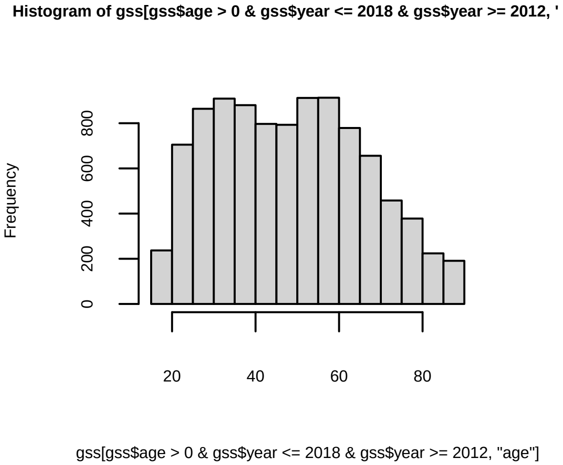
- Histograms describe the distribution of **numeric** data

```
hist(gss$age)
```



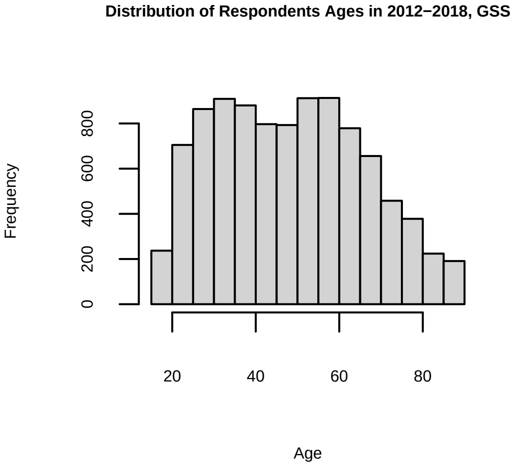
- Again, we want to drop negative values of age
- We may also want to see the distribution for some years but not others

```
hist(gss[gss$age>0 & gss$year<=2018 & gss$year>=2012,"age"])
```



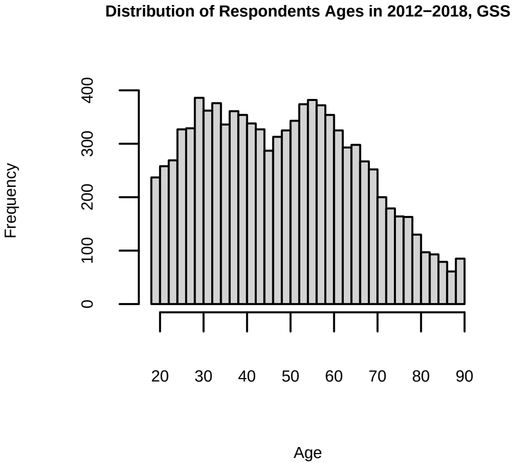
- To change plot titles and x-axis label

```
hist(gss[gss$age>0 & gss$year<=2018 & gss$year>=2012,"age"],
     main = "Distribution of Respondents Ages in 2012-2018, GSS",
     xlab = "Age")
```



- To control the number of bins

```
hist(gss[gss$age>0 & gss$year<=2018 & gss$year>=2012,"age"],
     main = "Distribution of Respondents Ages in 2012-2018, GSS",
     xlab = "Age",
     breaks = 30)
```



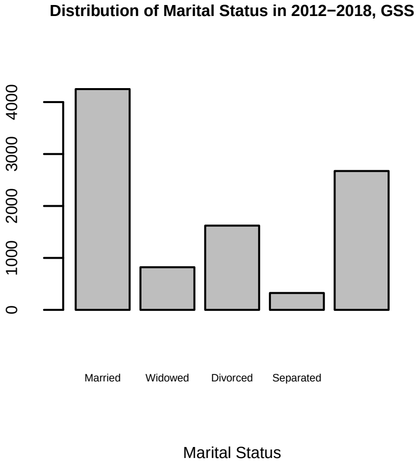
Barplot

- Barplots describe the distribution of **categorical** data

```
## create a count summary in each category by function `table`
counts <- table(gss[gss$age>0 & gss$year<=2018 &
           gss$year>=2012 & gss$marital>=0, "marital"])
counts
```

```
## 
##    1    2    3    4    5 
## 4250  821 1622  325 2673
```

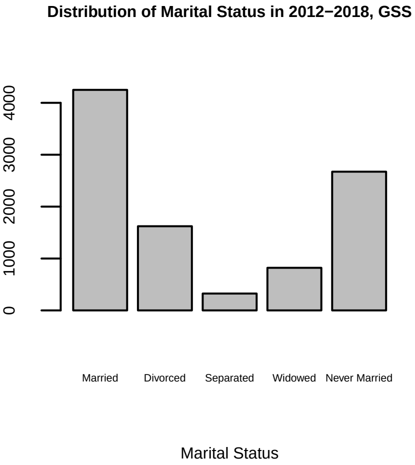
```
barplot(counts, main="Distribution of Marital Status in 2012-2018, GSS",
   xlab="Marital Status",
   names.arg=c("Married", "Widowed", "Divorced", "Separated",
               "Never Married"),
   cex.lab=0.5, cex.axis=0.5, cex.main=0.5, cex.sub=0.5, cex.names=0.32)
```



- We can also change the order of the categories by first `factorizing` the variable with manual `levels`

```r
gss$marital <- factor(gss$marital,levels = c(1,3,4,2,5))
counts <- table(gss[gss$age>0 & gss$year<=2018 &
           gss$year>=2012, "marital"])
```

```
barplot(counts, main="Distribution of Marital Status in 2012-2018, GSS",
   xlab="Marital Status",
   names.arg=c("Married", "Divorced", "Separated", "Widowed", "Never Married"),
   cex.lab=0.5, cex.axis=0.5, cex.main=0.5, cex.sub=0.5, cex.names=0.35)
```



Summarize Data (Advanced)

Summarize data by group

- In many cases, we not only want data summaries for the total sample, but summaries for some sub-samples
- For example, we may want to summarize religious preference by gender (the column `relig`)
- In the `relig` column, 4 means no religious belief, and other non-zero numbers mean some religious belief
- It is a non-ordered categorical variable

```
unique(gss$relig)
```

```
##  [1]   3   2   1   5   4 -99 -98  13  11   9   6  10  12   7   8 -97
```

Summarize data by group

- As there are many religious categories, we may only want two groups categorization for now: religious and non-religious individuals

```r
gss[gss$relig!=4 & gss$relig>0,
    "relig"] <- 1
gss[gss$relig==4,
    "relig"] <- 0
unique(gss$relig)
```

```
## [1]   1   0 -99 -98 -97
```

Summarize data by group

- We will use the `group_by()` function in `dplyr` to summarize data by the group we specify
- It can be nested in a `%>%` pipeline

```
gss %>%
  filter(relig>=0&sex>0) %>%
  group_by(sex) %>%
  summarize(religious = mean(relig))
```

```
## # A tibble: 2 x 2
##     sex religious
##   <int>     <dbl>
## 1     1     0.835
## 2     2     0.897
```

Exercise

- The calculation above includes all individuals surveyed since 1972 until 2021
- How to get the same statistic, but only in 2021?

Exercise

- The variable `relig16` asks which religion was the respondent raised in
- How to get the proportion of religious people (at the time of being interviewed, the column `relig`) by whether they were raised in a religious family?
- Do not forget to drop missing values first!

Exercise

```
gss[gss$relig16!=4&gss$relig16>=0,"relig16"] <- 1
gss[gss$relig16==4,"relig16"] <- 0
gss %>%
  filter(relig>=0&sex>0&relig16>=0) %>%
  group_by(relig16) %>%
  summarize(religious = mean(relig))
```

```
## # A tibble: 2 x 2
##   relig16 religious
##     <dbl>     <dbl>
## 1       0     0.450
## 2       1     0.904
```

Exercise

- The variable `relig16` asks which religion was the respondent raised in
- How to get the proportion of religious people (at the time of being interviewed, the column `relig`) by whether they were raised in a religious family?
- How to add another column summarizing the proportion of non-religious people by whether they were raised in a religious family?

Exercise

```
gss[gss$relig16!=4&gss$relig16>=0,"relig16"] <- 1
gss[gss$relig16==4,"relig16"] <- 0
gss %>%
  filter(relig>=0&sex>0&relig16>=0) %>%
  group_by(relig16) %>%
  summarize(religious = mean(relig),
            nonreligious=mean(1-relig))
```

```
## # A tibble: 1 x 3
##   relig16 religious nonreligious
##     <dbl>     <dbl>        <dbl>
## 1       1     0.879        0.121
```

Viasualizing the trend

- We have already calculated the proportion of religious men and women in all and one specific year(s)
- Another common exploration of such data is to analyze the temporal trend of the proportion of religious people by gender over the years

```
gss %>%
  filter(relig>=0&sex>0) %>%
  group_by(year, sex) %>%
  summarize(religious = mean(relig)) %>%
  head()
```

```
## # A tibble: 6 x 3
## # Groups:   year [3]
##     year   sex religious
##    <int> <int>     <dbl>
## 1  1972     1     0.932
## 2  1972     2     0.965
## 3  1973     1     0.911
## 4  1973     2     0.958
## 5  1974     1     0.903
## 6  1974     2     0.957
```

Viasualizing the trend (ggplot2)

- While base R can handle such visualizations, we will introduce another powerful visualization package, ggplot2
- ggplot2 can also be nested in the %>% pipeline
- It is highly flexible in changing colors, setting fonts, customing legends, etc.

Viasualizing the trend (ggplot2)

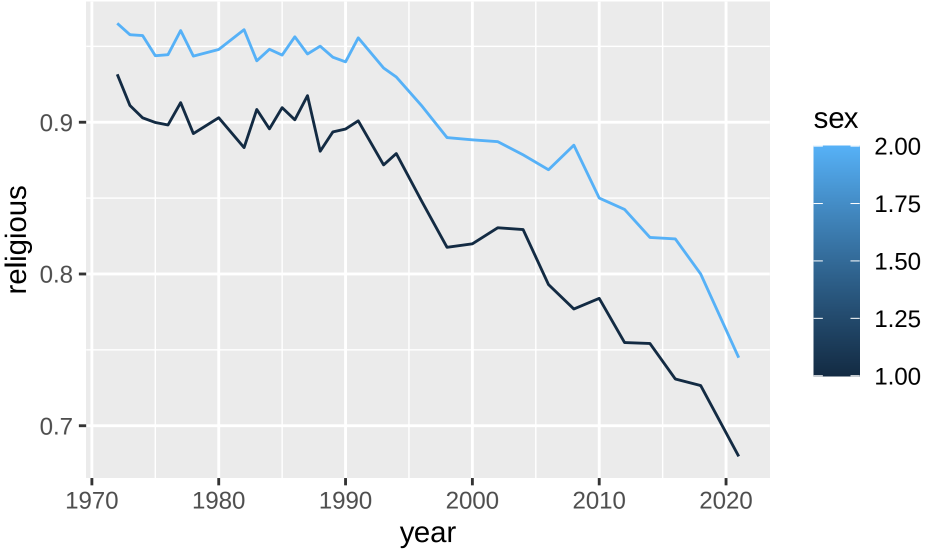
- The basic structure of ggplot2 is

```
## create a ggplot object
ggplot(data,
       aes(x=variable1,
           y=variable2,
           group=variable3,
           ...)) +
  geom_line() +
  geom_point() +
  ...
```

- The basic **parameters** such as x-axis, y-axis, groups and colors, are controlled by `aes()` that stands for aesthetics
- To create a trend plot, we need a **line** plot controlled by `geom_line()`
- There are many other options, including `geom_point()` that creates scatter plot, `geom_histogram()` that creates histograms, etc.
- Check this website for much more details of what `ggplot2` can do!
- https://ggplot2.tidyverse.org/reference/

Viasualizing the trend (ggplot2)

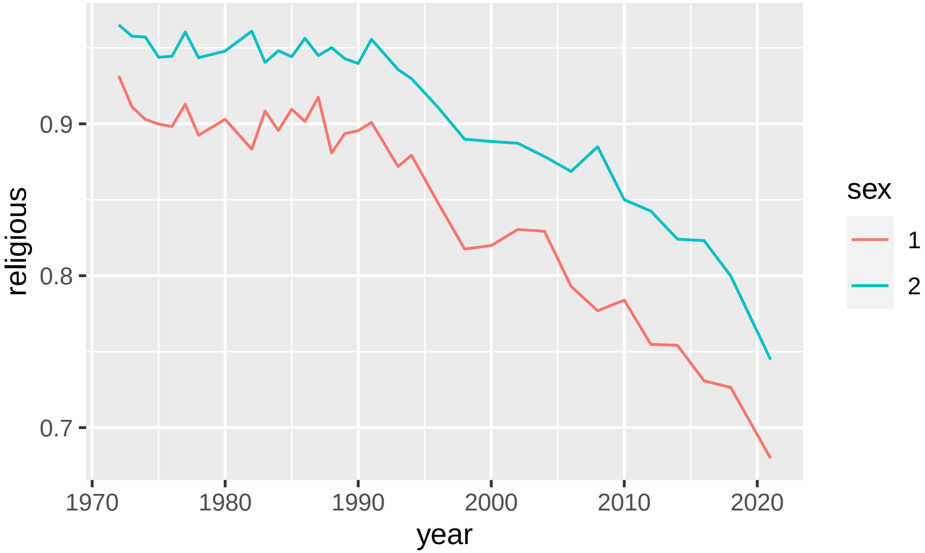
```
library(ggplot2)
gss %>%
  filter(relig>=0&sex>0) %>%
  group_by(year, sex) %>%
  summarize(religious = mean(relig)) %>%
  ggplot(aes(x=year,y=religious,group=sex,color=sex)) +
  geom_line()
```


religious
0.9
0.8
0.7
1970
1980
1990
2000
2010
2020
year
sex
2.00
1.75
1.50
1.25
1.00

- It looks fine, but need some refinements!

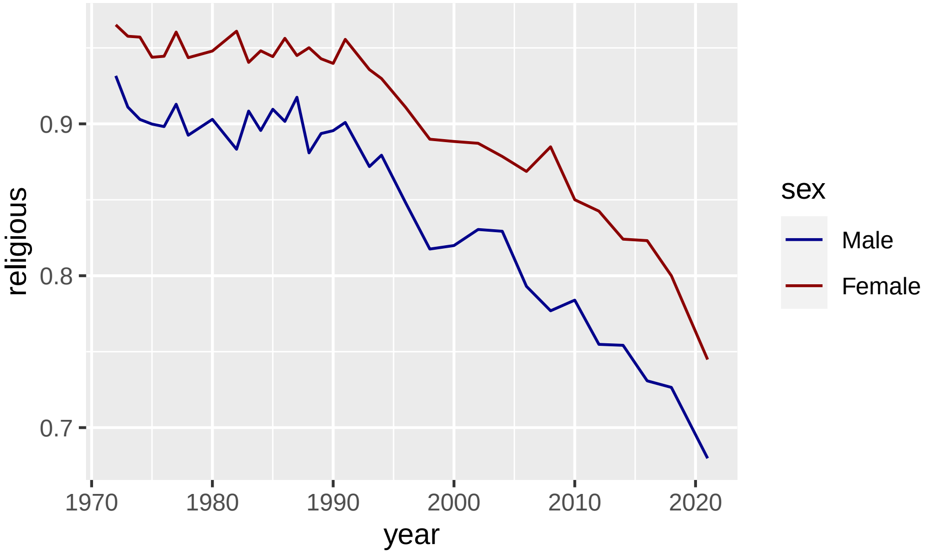
- It looks fine, but need some refinements!
- R identifies the variable `sex` as a continuous variable, we need a categorical variable
  - We can change the data type of `sex` into characters
  - We use function `mutate(new_variable = ...)` in `dplyr` to create new variable(s)
  - The new variable name can be the same as the original one (i.e., overwriting)

```
gss %>%
  filter(relig>=0 & sex>=0) %>%
  mutate(sex = as.character(sex)) %>%
  group_by(year, sex) %>%
  summarize(religious = mean(relig)) %>%
  ggplot(aes(x=year,y=religious,color=sex)) +
  geom_line()
```


religious
0.9
0.8
0.7
1970
1980
1990
2000
2010
2020
year
sex
1
2

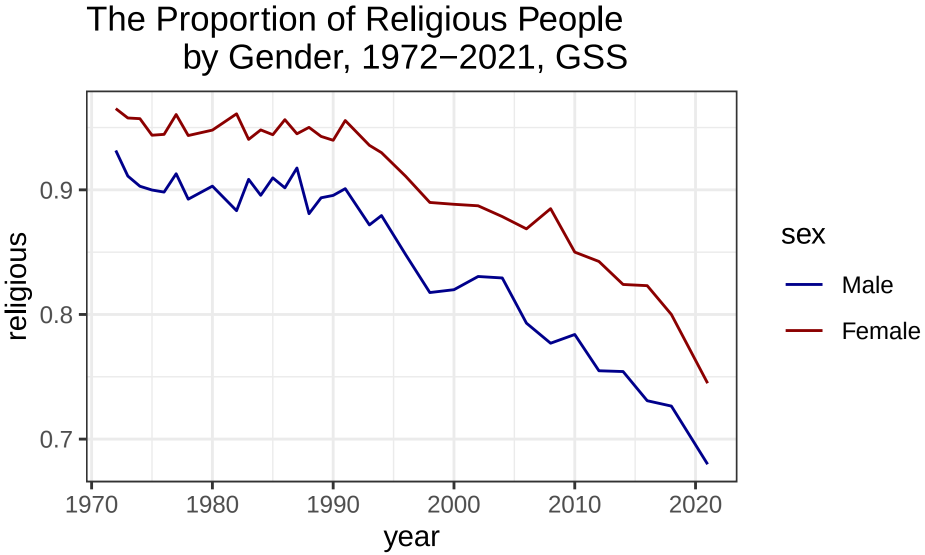
- Readers do not know what 1 and 2 in sex means! We need `male` and `female` labels
- We may also want to change the line color

```
gss %>%
  filter(relig>=0 & sex>=0) %>%
  mutate(sex = as.character(sex)) %>%
  group_by(year, sex) %>%
  summarize(religious = mean(relig)) %>%
  ggplot(aes(x=year,y=religious,color=sex)) +
  scale_color_manual(labels = c("Male", "Female"),
                     values = c("darkblue", "darkred"))
  geom_line()
```


religious
0.9
0.8
0.7
1970
1980
1990
2000
2010
2020
year
sex
Male
Female

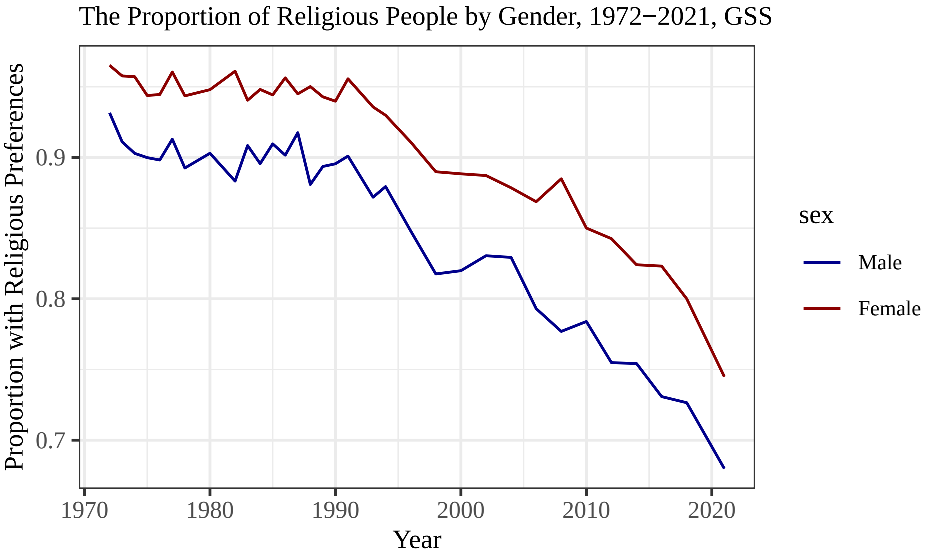
- The background looks a bit pale. . .
- And we need a title

```
gss %>%
  filter(relig>=0 & sex>=0) %>%
  mutate(sex = as.character(sex)) %>%
  group_by(year, sex) %>%
  summarize(religious = mean(relig)) %>%
  ggplot(aes(x=year,y=religious,color=sex)) +
  scale_color_manual(labels = c("Male", "Female"),
                     values = c("darkblue", "darkred")) +
  geom_line() +
  theme_bw() +
  ggtitle("The Proportion of Religious People \n
          by Gender, 1972-2021, GSS")
```


The Proportion of Religious People
by Gender, 1972−2021, GSS
religious
0.9
0.8
0.7
1970
1980
1990
2000
2010
2020
year
sex
Male
Female

With more tweaks, we can make the figure publishable

```r
gss %>%
  filter(relig>=0 & sex>=0) %>%
  mutate(sex = as.character(sex)) %>%
  group_by(year, sex) %>%
  summarize(religious = mean(relig)) %>%
  ggplot(aes(x=year,y=religious,color=sex)) +
  geom_line() +
  scale_color_manual(labels = c("Male", "Female"),
                     values = c("darkblue", "darkred")) +
  theme_bw() +
  ggtitle("The Proportion of Religious People by Gender, 1972-2021, GSS") +
  xlab("Year") +
  ylab("Proportion with Religious Preferences") +
  theme(plot.title=element_text(size=10),
        text=element_text(family="Times"),
        axis.title.x=element_text(size=8),
        axis.title.y=element_text(size=8),
        legend.title = element_text(size=8),
        legend.text = element_text(size=6))
```


The Proportion of Religious People by Gender, 1972−2021, GSS
Proportion with Religious Preferences
0.9
0.8
0.7
1970
1980
1990
2000
2010
2020
Year
sex
Male
Female