STOR 320 Factors

Lecture 11

Yao Li
Department of Statistics and Operations Research
UNC Chapel Hill

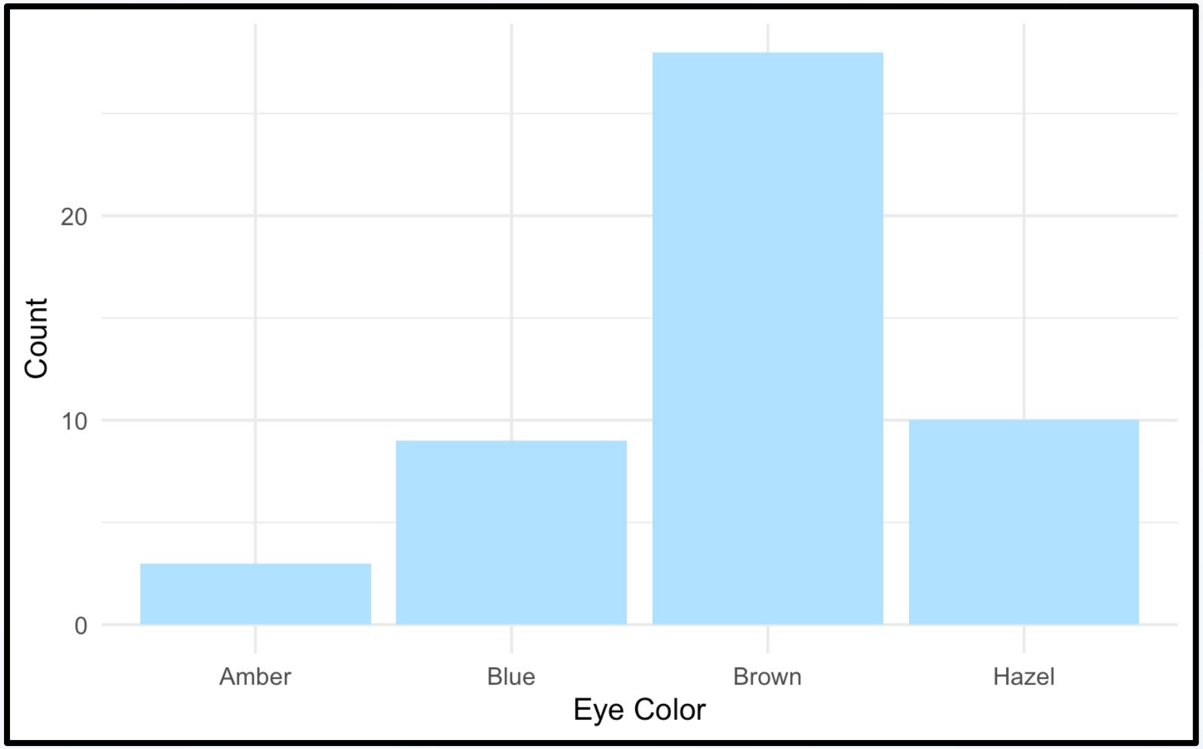
Introduction

- Read Chapter 15
- Additional Package

  -
    ```
    > library(forcats)
    ```
  - Part of the tidyverse
- For Variables with,
  - Fixed Set of Values
  - Known Set of Values
- Factors Are on a New Level

Motivation: Example 1

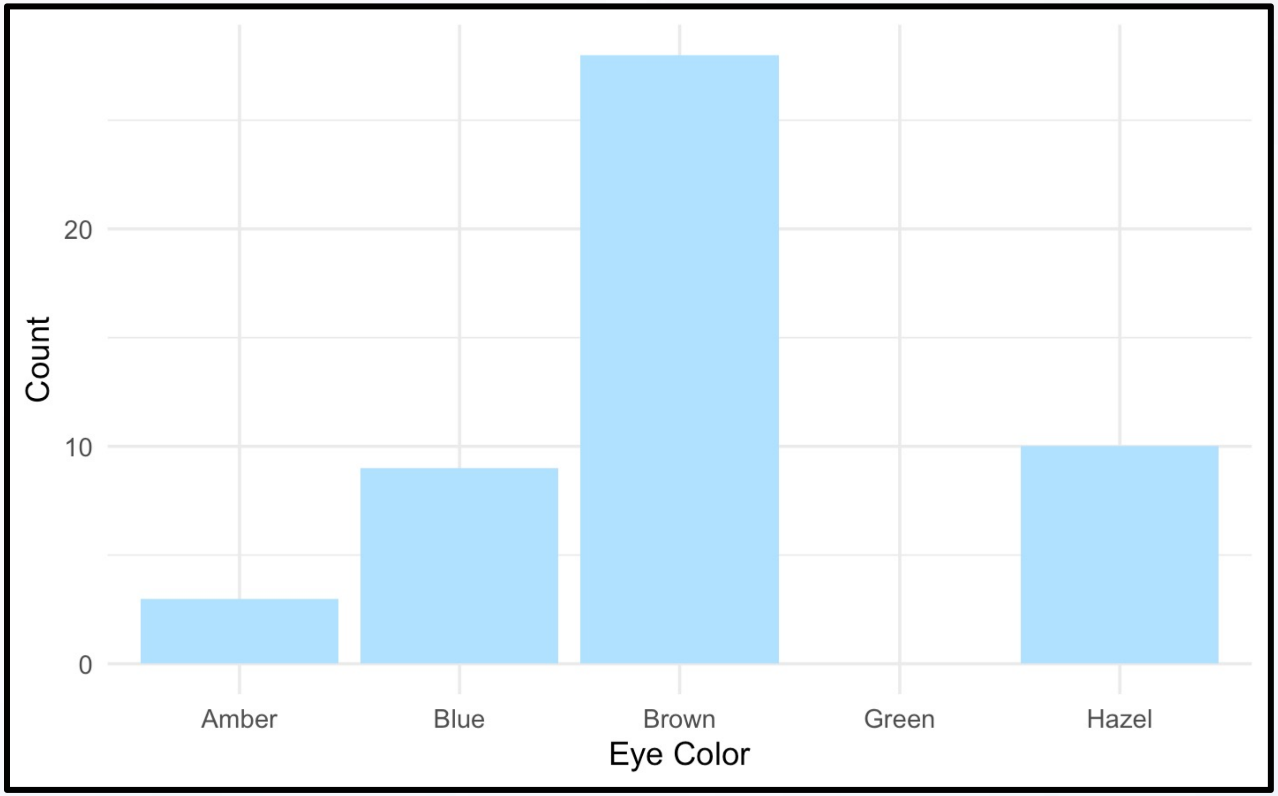
- Eye Color Distribution
  - Randomly Sample 50 People
  - Distribution via Bar Plot
  - How to Make More Informative?



Motivation: Example 1

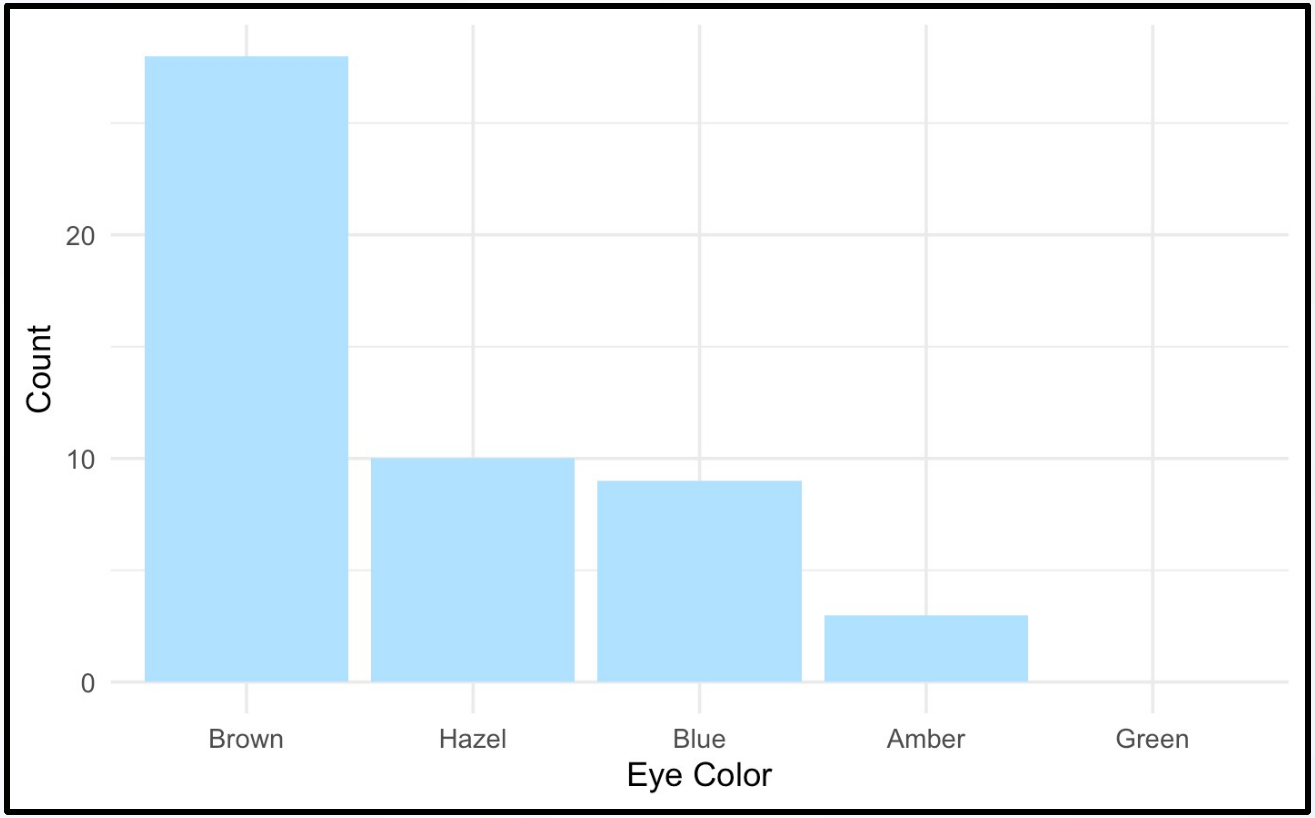
- Eye Color Distribution (Cont.)
  - Display Eye Colors Absent From Sample
  -

```
> scale_x_discrete(drop=F)
```



Motivation: Example 1

- Eye Color Distribution (Cont.)
  - Display in order



Motivation: Example 2

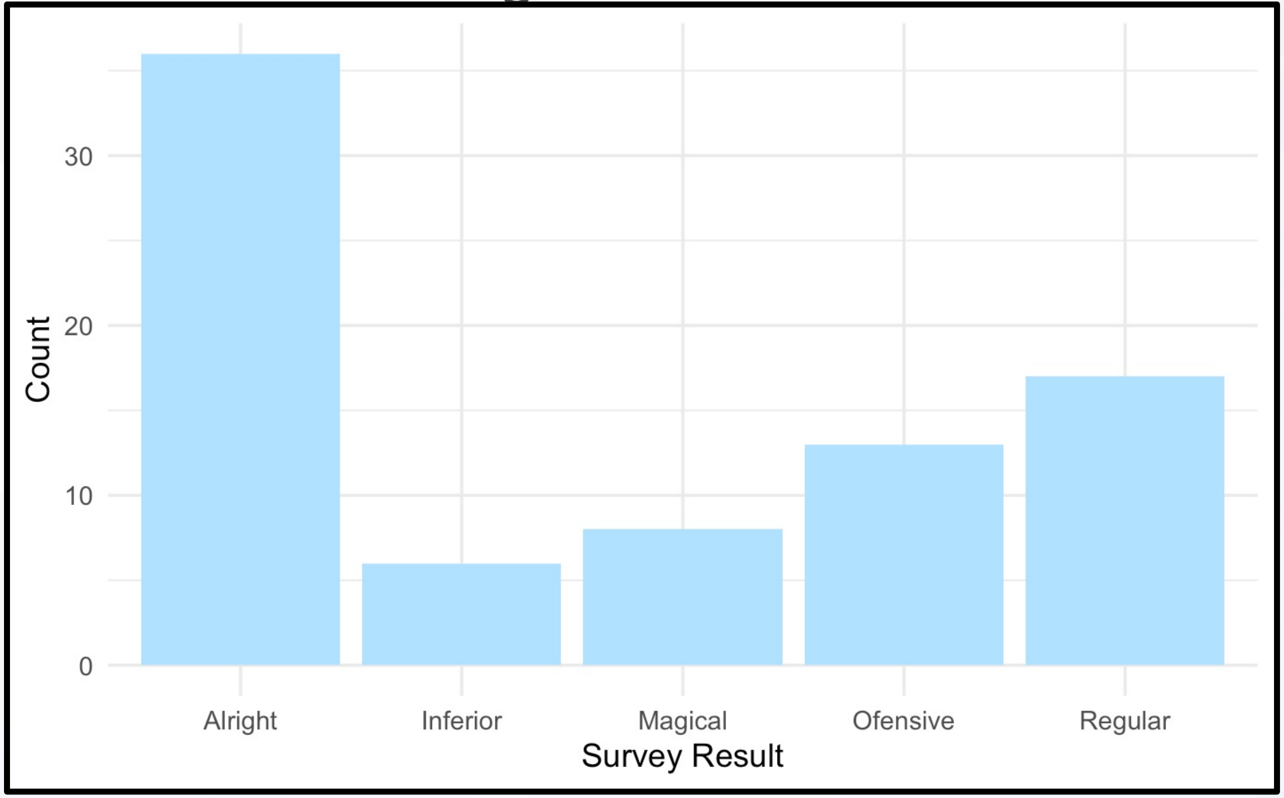
- Survey Results

  - How Would You Describe Dr. Example’s Teaching?
    - Magical
    - Alright
    - Regular
    - Inferior
    - Offensive

  - Class of 80 Students Answer End-of-the-Year Survey

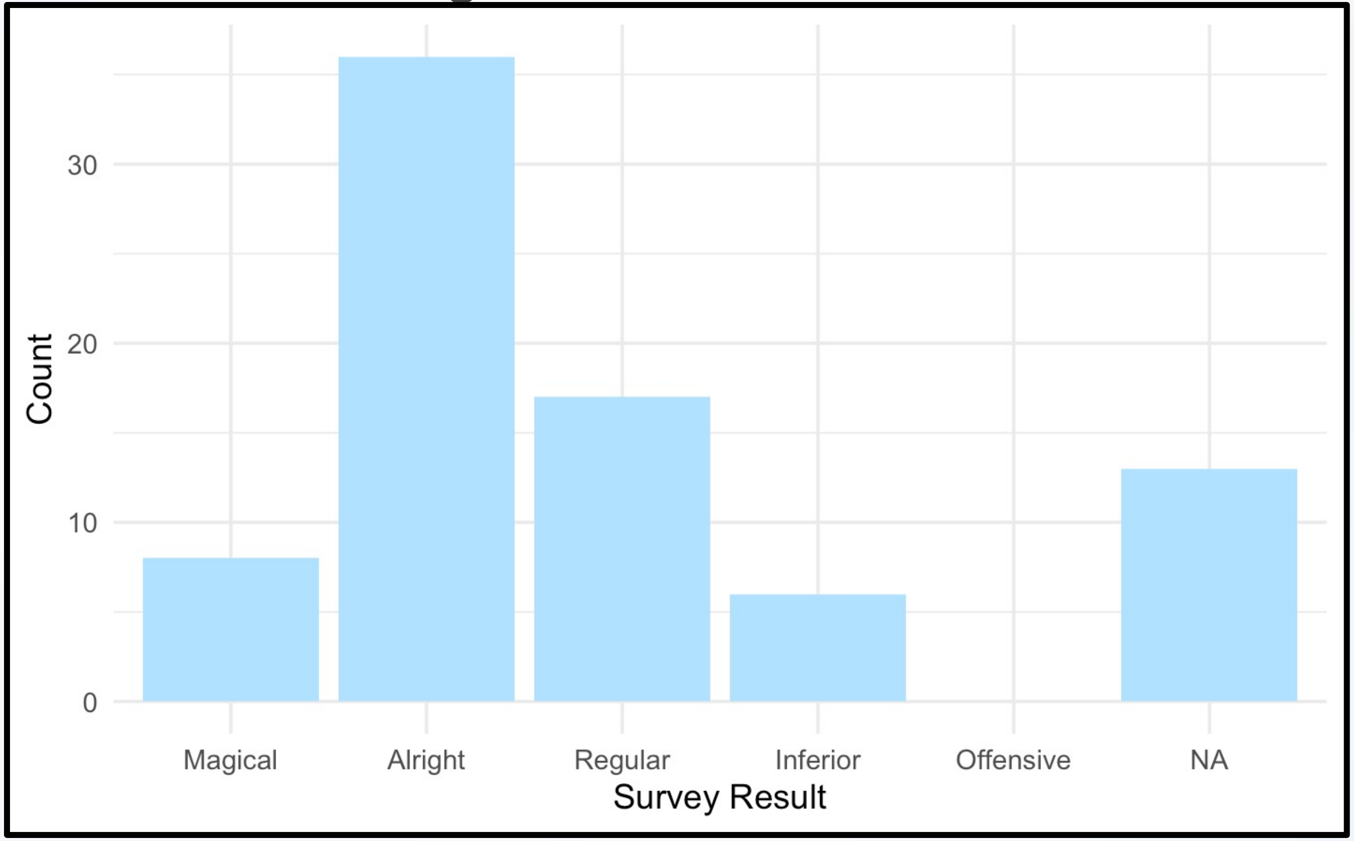
Motivation: Example 2

- Survey Results (Cont.)
  - Distribution of Results
  - What is Wrong?



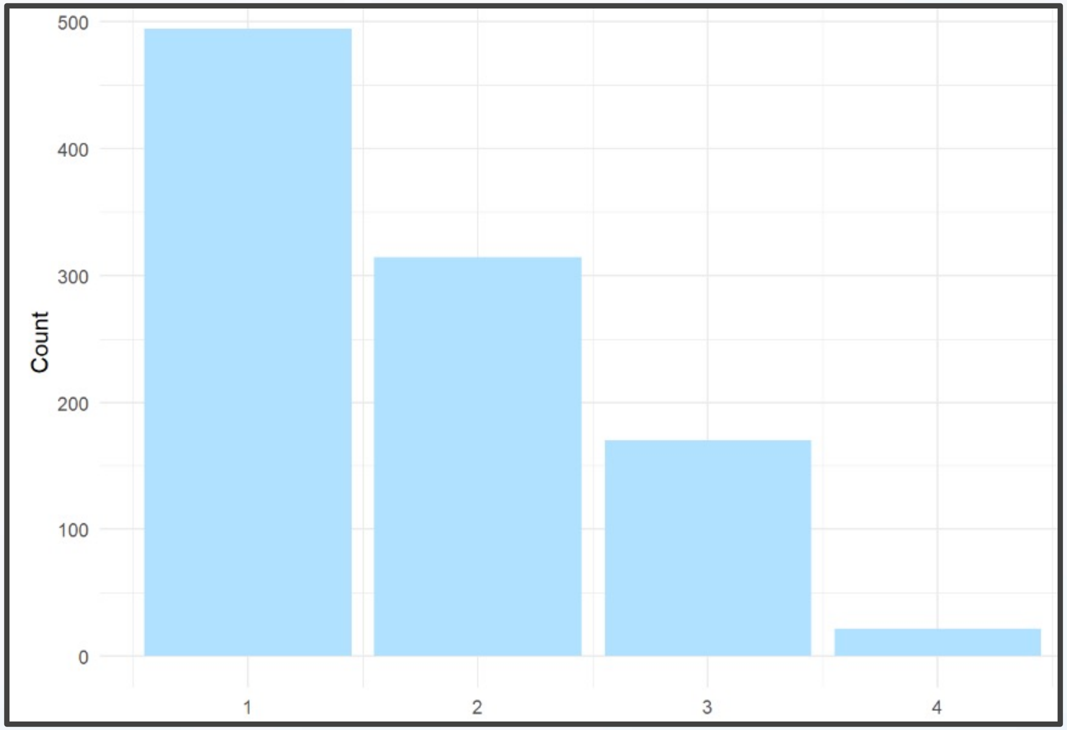
Motivation: Example 2

- Survey Results (Cont.)
  - Ordinal Categorical Variable



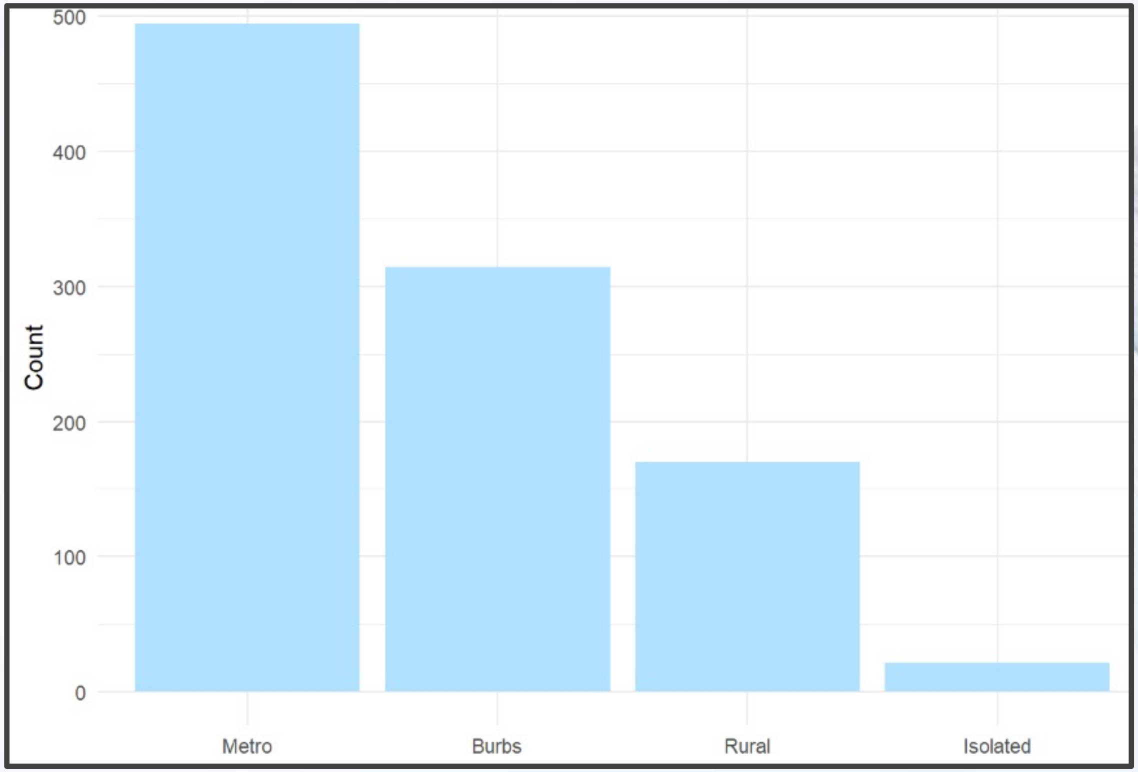
Motivation: Example 3

- Urbanicity
  - Classification {1,2,3,4}
  - Sample 1000 Households and Record Their Urbanicity
  - What Would Make this Better?



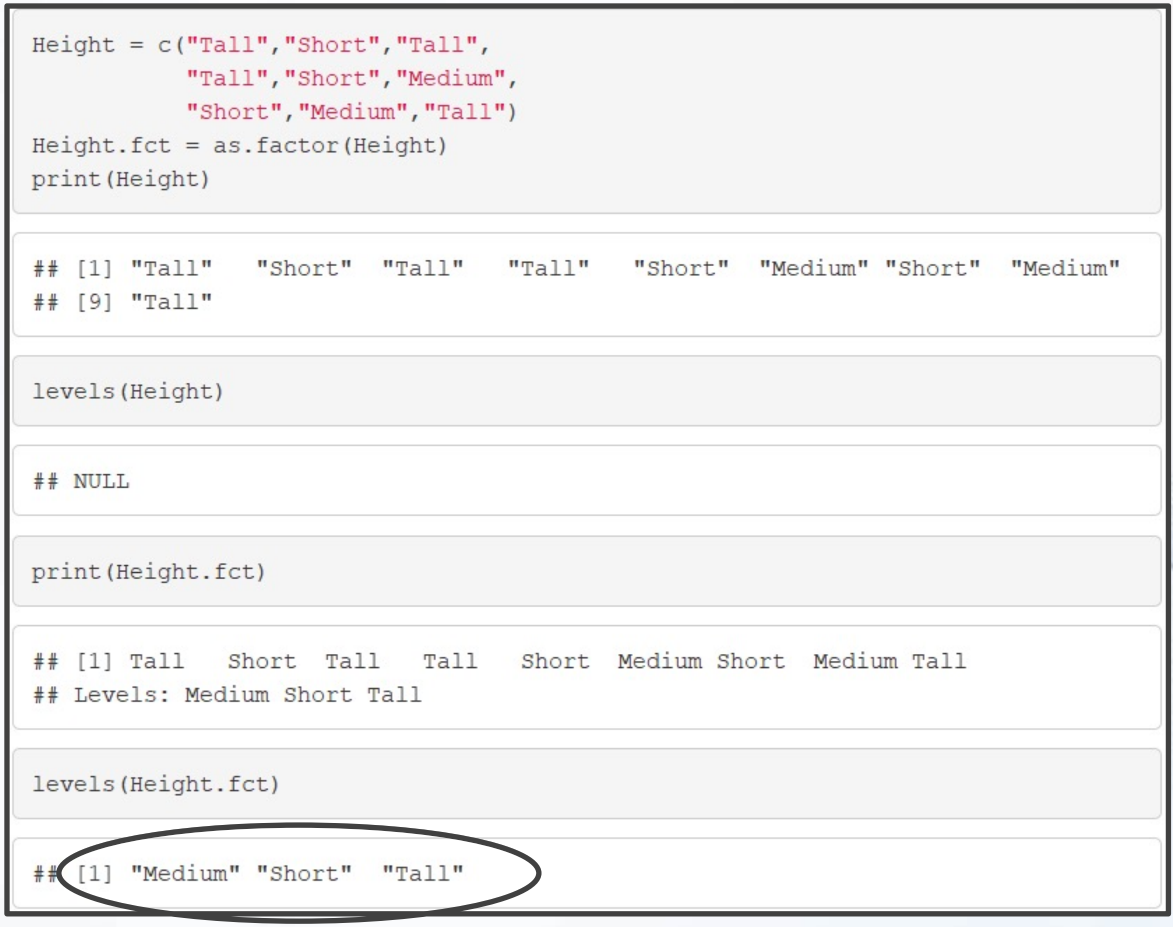
Motivation: Example 3

- Urbanicity
  - Data Dictionary
    - 1 = Metropolitan
    - 2 = Burbs
    - 3 = Rural
    - 4 = Isolated



Factor Variable Architecture

- Factor Variables Have Levels



```
Height = c("Tall","Short","Tall",
           "Tall","Short","Medium",
           "Short","Medium","Tall")
Height.fct = as.factor(Height)
print(Height)
```

```
## [1] "Tall"   "Short"  "Tall"   "Tall"   "Short"  "Medium" "Short"  "Medium"
## [9] "Tall"
```

```
levels(Height)
```

```
## NULL
```

```
print(Height.fct)
```

```
## [1] Tall   Short  Tall   Tall   Short  Medium Short  Medium Tall
## Levels: Medium Short Tall
```

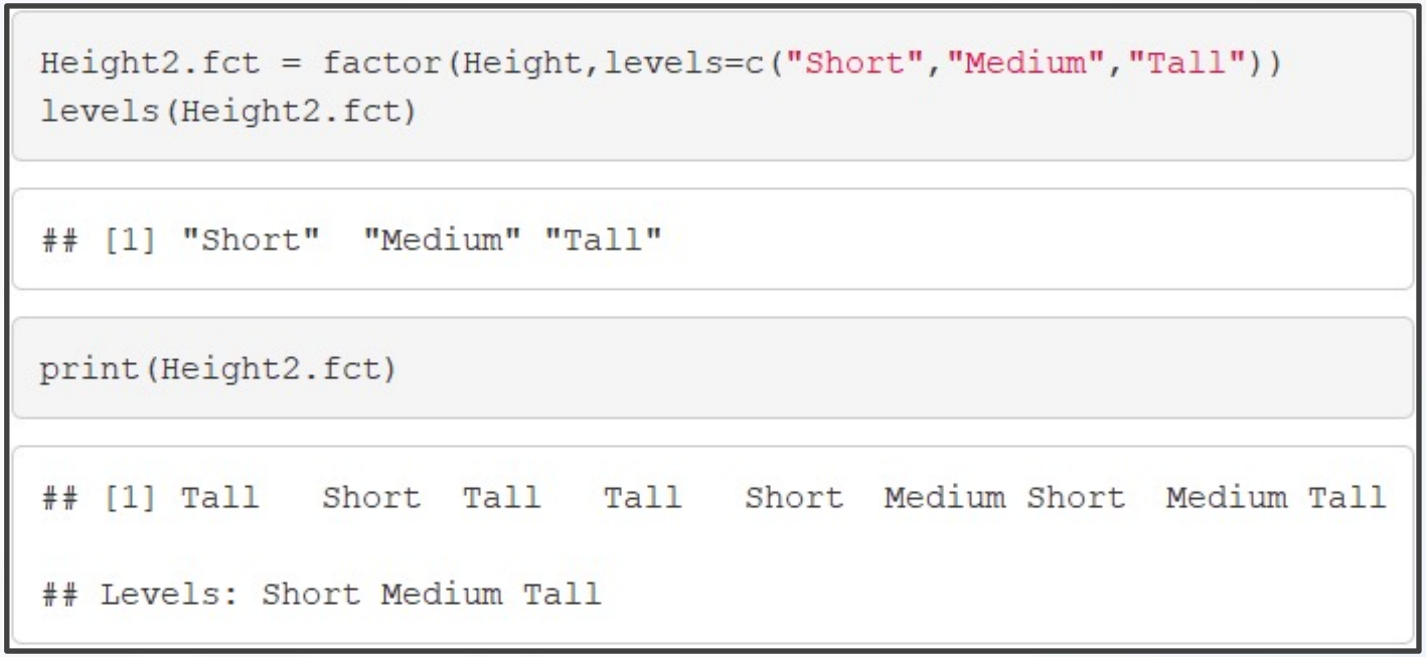
```
levels(Height.fct)
```

```
## [1] "Medium" "Short"  "Tall"
```

Default: Alphabetical

Factor: Level Order

- Level Order May Be Specified



```
Height2.fct = factor(Height,levels=c("Short","Medium","Tall"))
levels(Height2.fct)
```

```
## [1] "Short"  "Medium" "Tall"
```

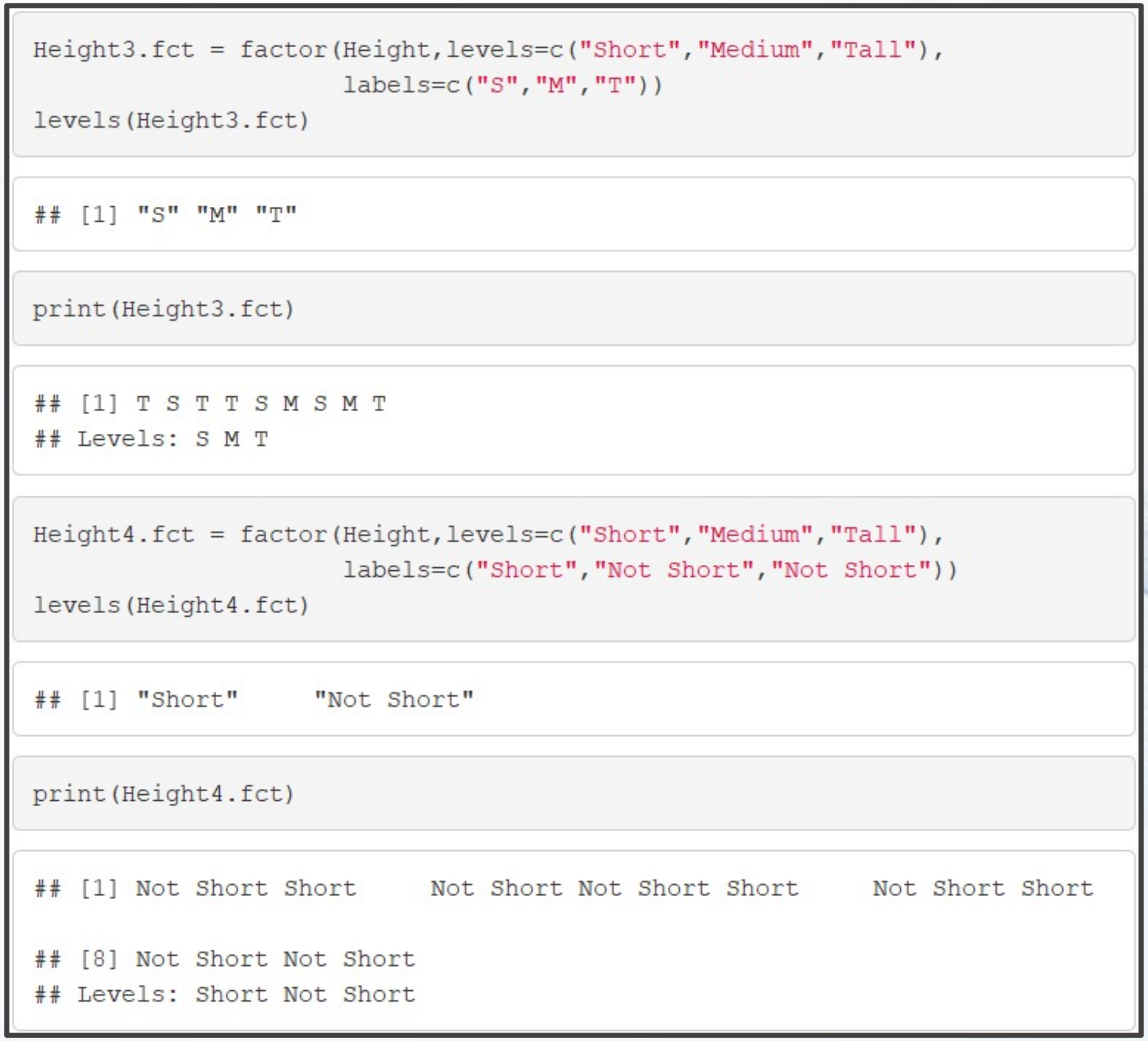
```
print(Height2.fct)
```

```
## [1] Tall   Short  Tall   Tall   Short  Medium Short  Medium Tall

## Levels: Short Medium Tall
```

Factor: Label

- Levels May Be Labeled



```
Height3.fct = factor(Height,levels=c("Short","Medium","Tall"),
                     labels=c("S","M","T"))
levels(Height3.fct)
```

```
## [1] "S" "M" "T"
```

```
print(Height3.fct)
```

```
## [1] T S T T S M S M T
## Levels: S M T
```

```
Height4.fct = factor(Height,levels=c("Short","Medium","Tall"),
                     labels=c("Short","Not Short","Not Short"))
levels(Height4.fct)
```

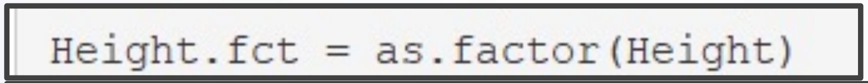
```
## [1] "Short"     "Not Short"
```

```
print(Height4.fct)
```

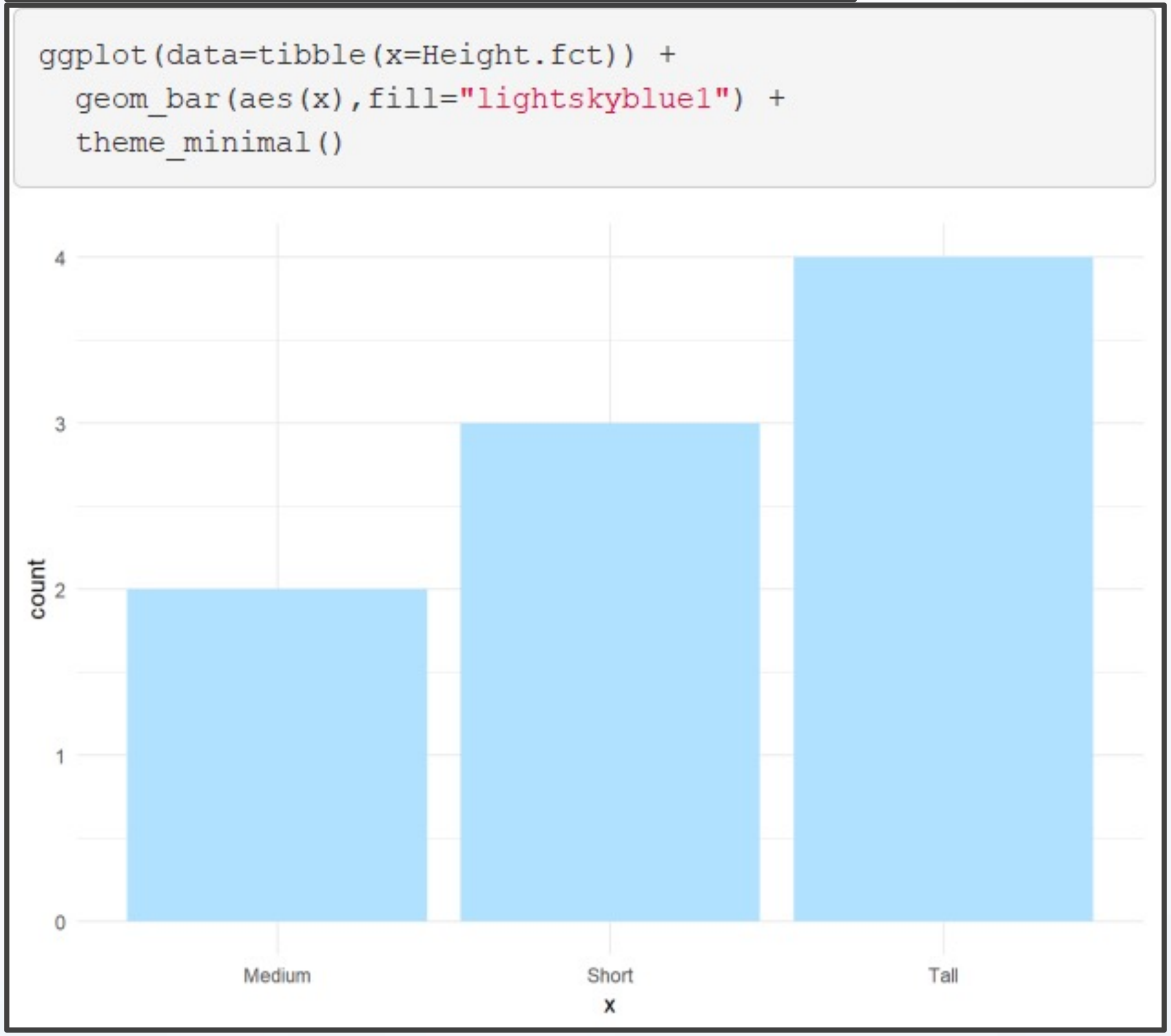
```
## [1] Not Short Short     Not Short Not Short Short     Not Short Short

## [8] Not Short Not Short
## Levels: Short Not Short
```

Graphic Comparison

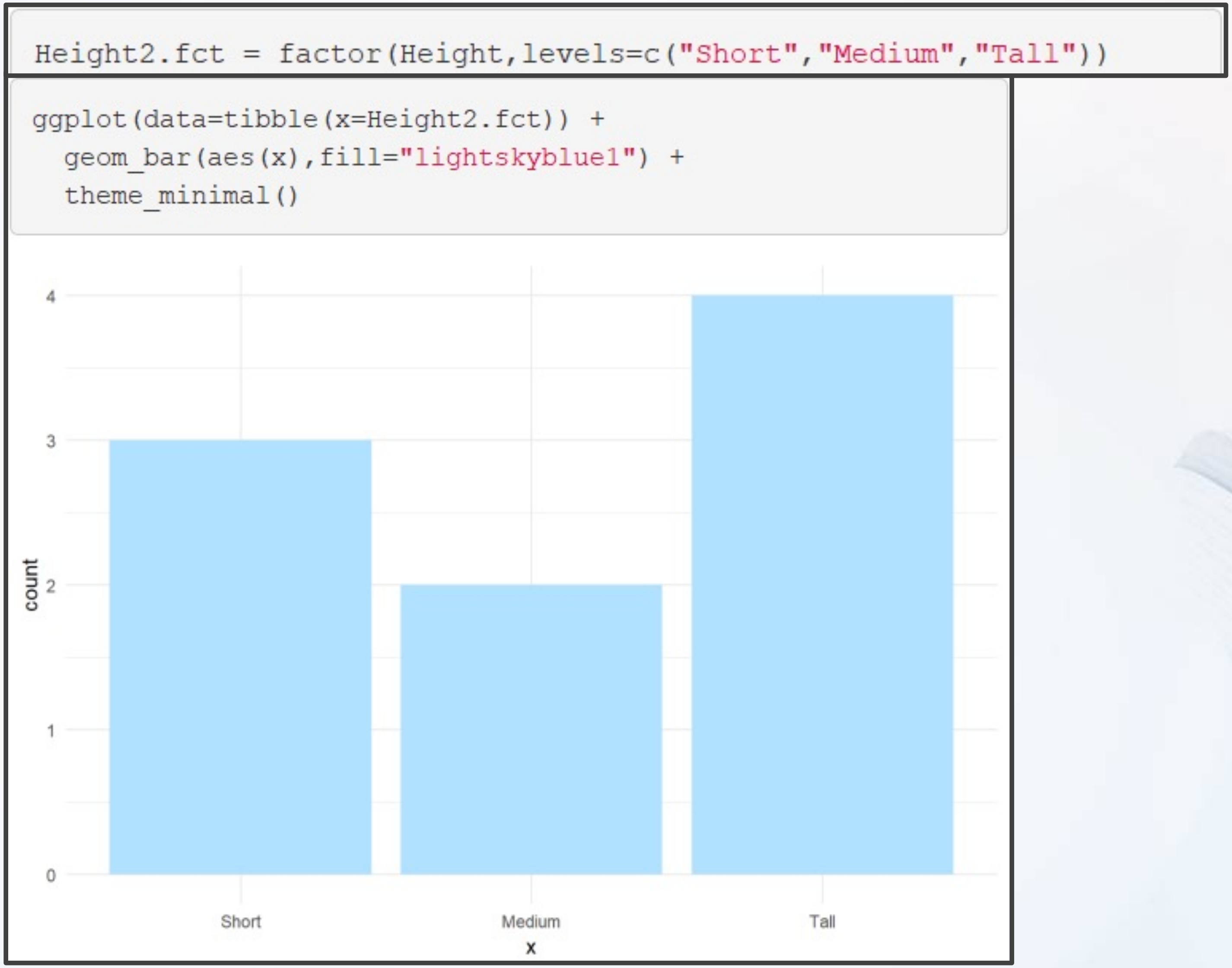


```
Height.fct = as.factor(Height)
```



```
ggplot(data=tibble(x=Height.fct)) +
  geom_bar(aes(x),fill="lightskyblue1") +
  theme_minimal()
```

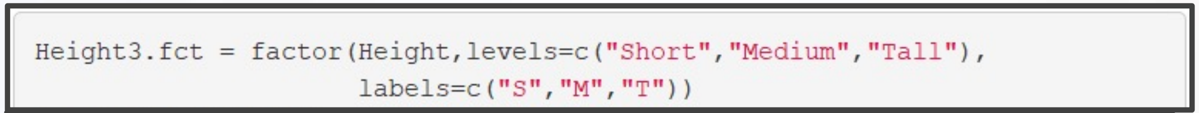
Graphic Comparison



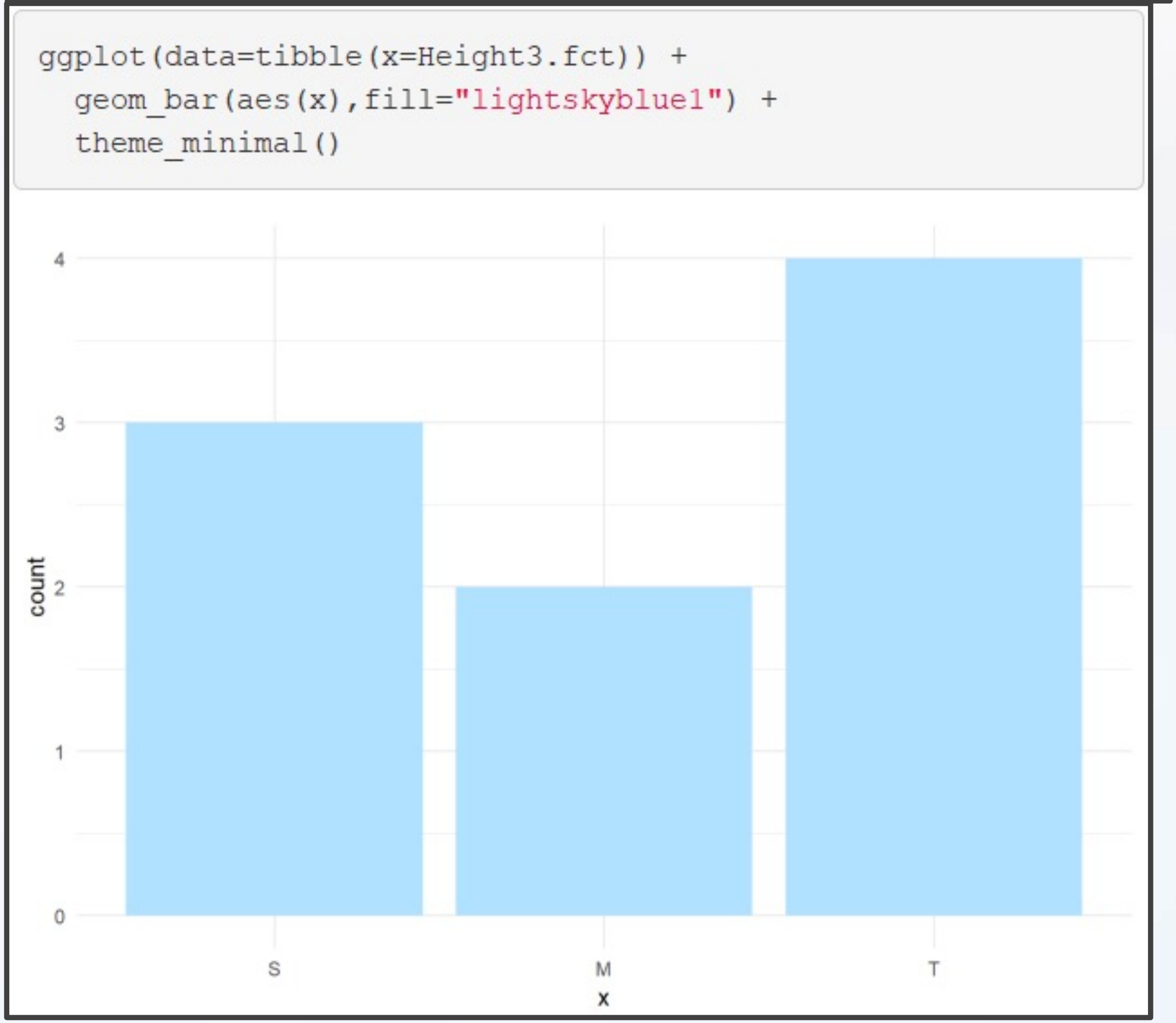
```
Height2.fct = factor(Height,levels=c("Short","Medium","Tall"))
```

```
ggplot(data=tibble(x=Height2.fct)) +
  geom_bar(aes(x),fill="lightskyblue1") +
  theme_minimal()
```

Graphic Comparison

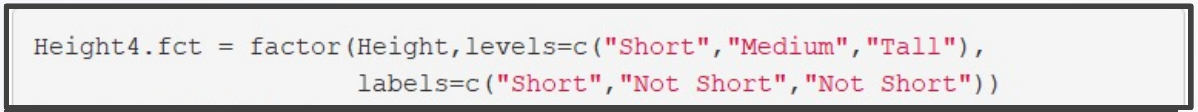


```
Height3.fct = factor(Height,levels=c("Short","Medium","Tall"),
                     labels=c("S","M","T"))
```

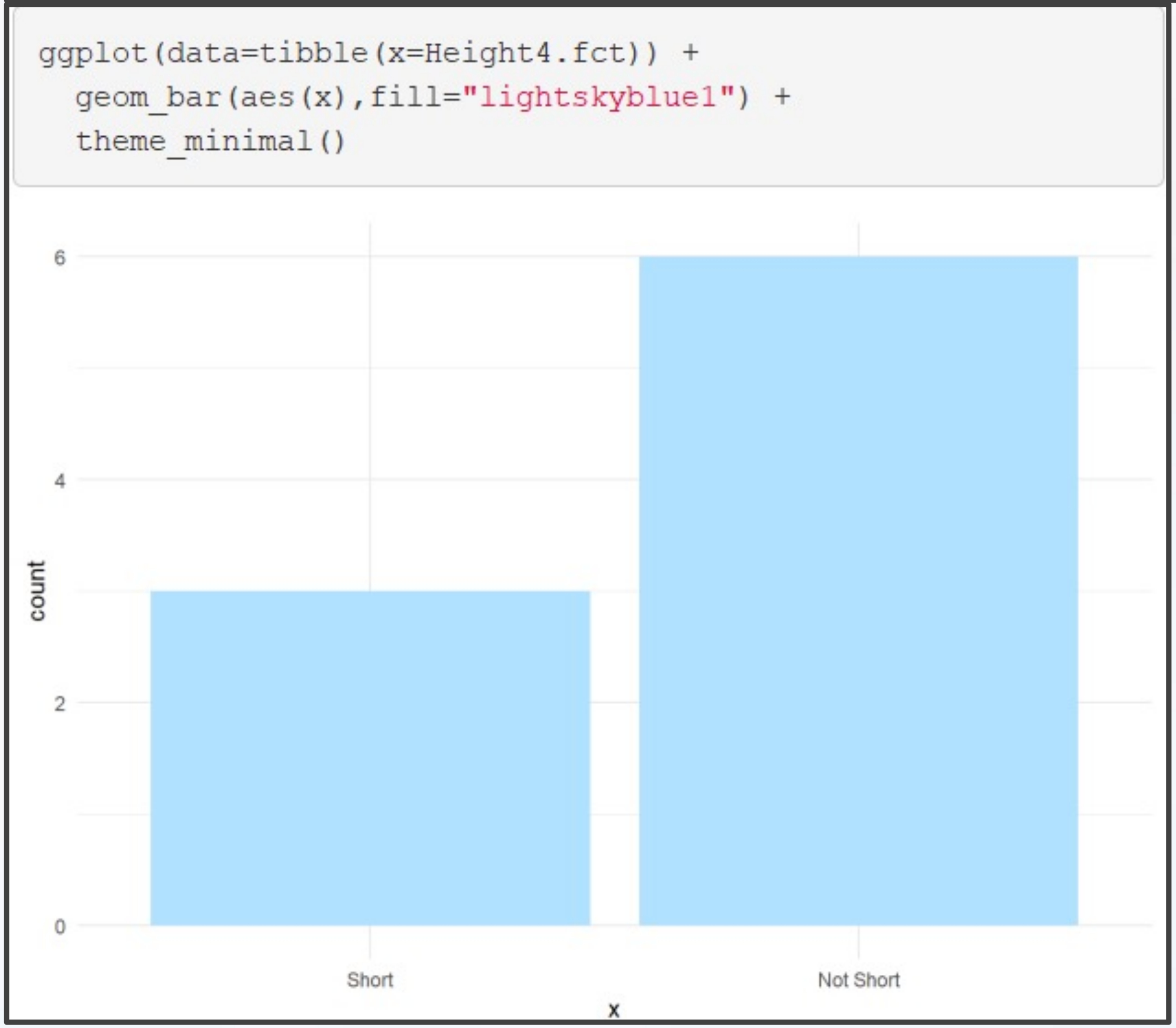


```
ggplot(data=tibble(x=Height3.fct)) +
  geom_bar(aes(x),fill="lightskyblue1") +
  theme_minimal()
```

Graphic Comparison



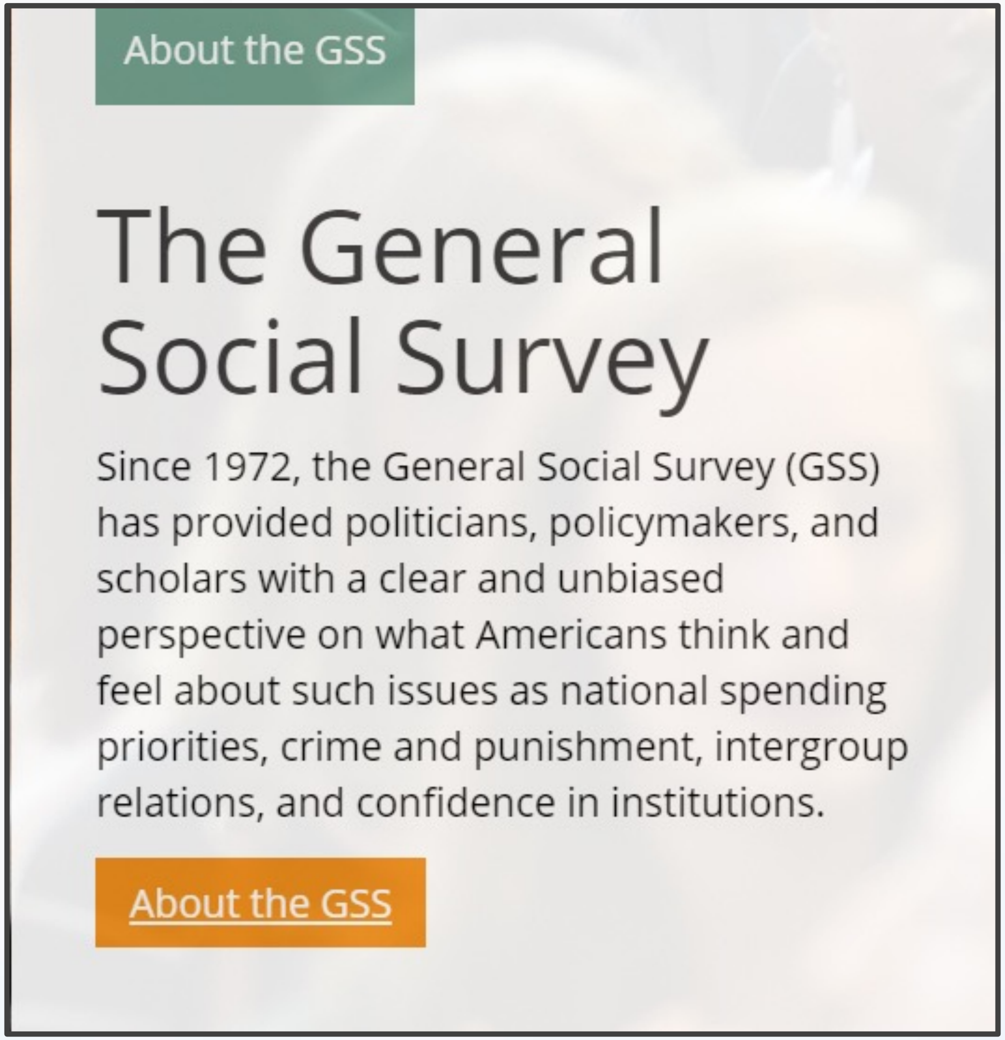
```
Height4.fct = factor(Height,levels=c("Short","Medium","Tall"),
                     labels=c("Short","Not Short","Not Short"))
```



```
ggplot(data=tibble(x=Height4.fct)) +
  geom_bar(aes(x),fill="lightskyblue1") +
  theme_minimal()
```

General Social Survey

- University of Chicago



About the GSS

The General Social Survey

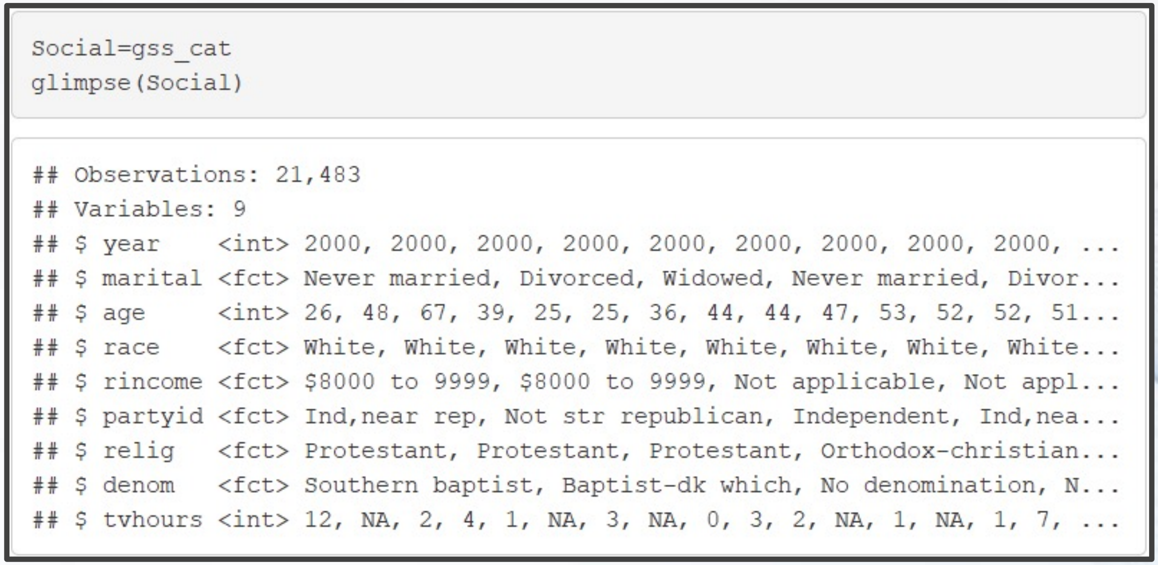
Since 1972, the General Social Survey (GSS) has provided politicians, policymakers, and scholars with a clear and unbiased perspective on what Americans think and feel about such issues as national spending priorities, crime and punishment, intergroup relations, and confidence in institutions.

About the GSS

General Social Survey

- Sample Provided in gss_cat

- Factor Variables Included
  - Marital
  - Race
  - Income Range
  - Political Party
  - Religion
  - Denomination

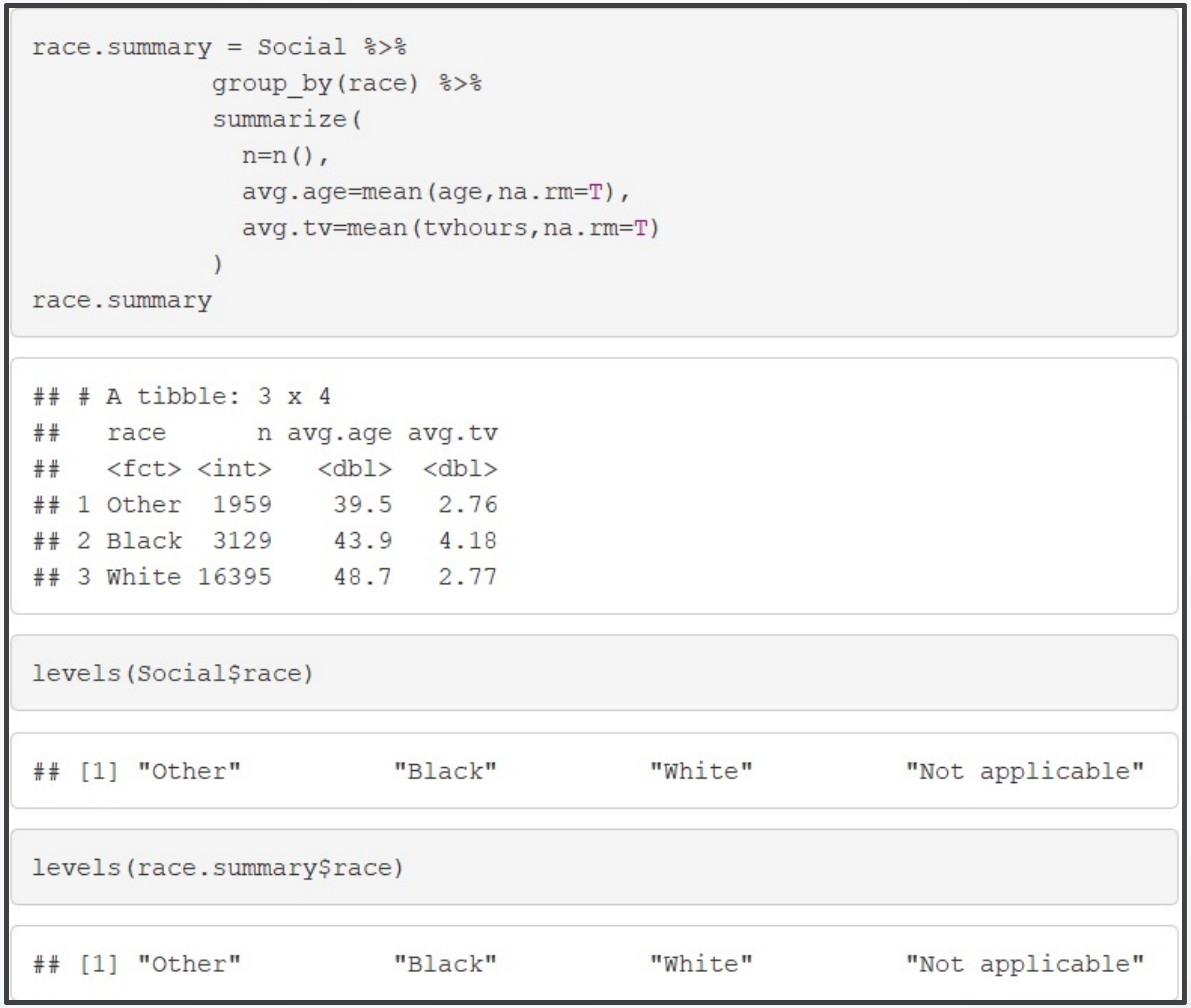


```
Social=gss_cat
glimpse(Social)
```

```
## Observations: 21,483
## Variables: 9
## $ year    <int> 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, ...
## $ marital <fct> Never married, Divorced, Widowed, Never married, Divor...
## $ age     <int> 26, 48, 67, 39, 25, 25, 36, 44, 44, 47, 53, 52, 52, 51...
## $ race    <fct> White, White, White, White, White, White, White, White...
## $ rincome <fct> $8000 to 9999, $8000 to 9999, Not applicable, Not appl...
## $ partyid <fct> Ind,near rep, Not str republican, Independent, Ind,nea...
## $ relig   <fct> Protestant, Protestant, Protestant, Orthodox-christian...
## $ denom   <fct> Southern baptist, Baptist-dk which, No denomination, N...
## $ tvhours <int> 12, NA, 2, 4, 1, NA, 3, NA, 0, 3, 2, NA, 1, NA, 1, 7, ...
```

Modifying Factor Order

- Summary by Race



```
race.summary = Social %>%
            group_by(race) %>%
            summarize(
              n=n(),
              avg.age=mean(age,na.rm=T),
              avg.tv=mean(tvhours,na.rm=T)
            )
race.summary
```

```
## # A tibble: 3 x 4
##   race       n avg.age avg.tv
##   <fct> <int>   <dbl>  <dbl>
## 1 Other  1959    39.5   2.76
## 2 Black  3129    43.9   4.18
## 3 White 16395    48.7   2.77
```

```
levels(Social$race)
```

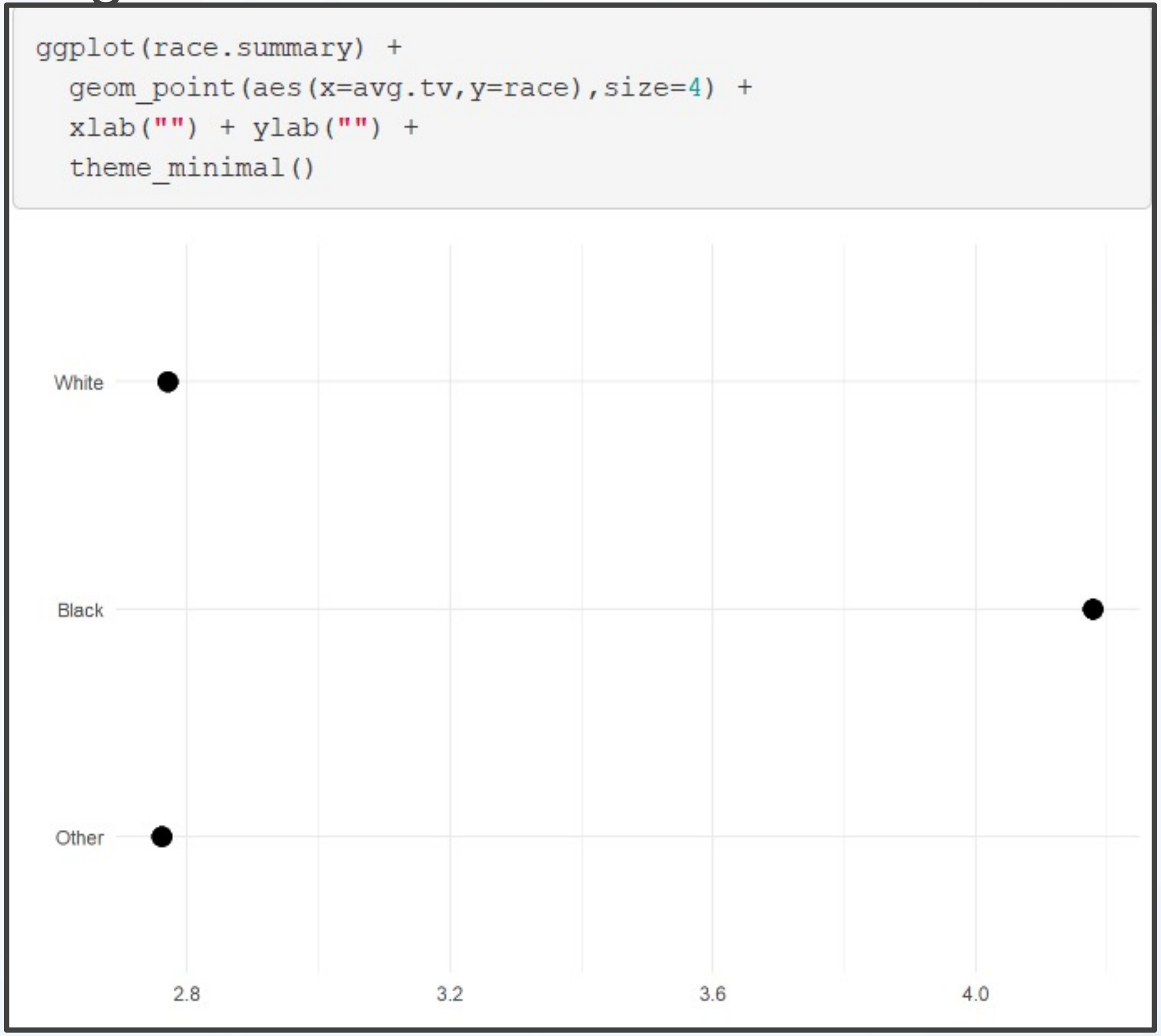
```
## [1] "Other"          "Black"          "White"          "Not applicable"
```

```
levels(race.summary$race)
```

```
## [1] "Other"          "Black"          "White"          "Not applicable"
```

Modifying Factor Order

- Comparing TV Hours



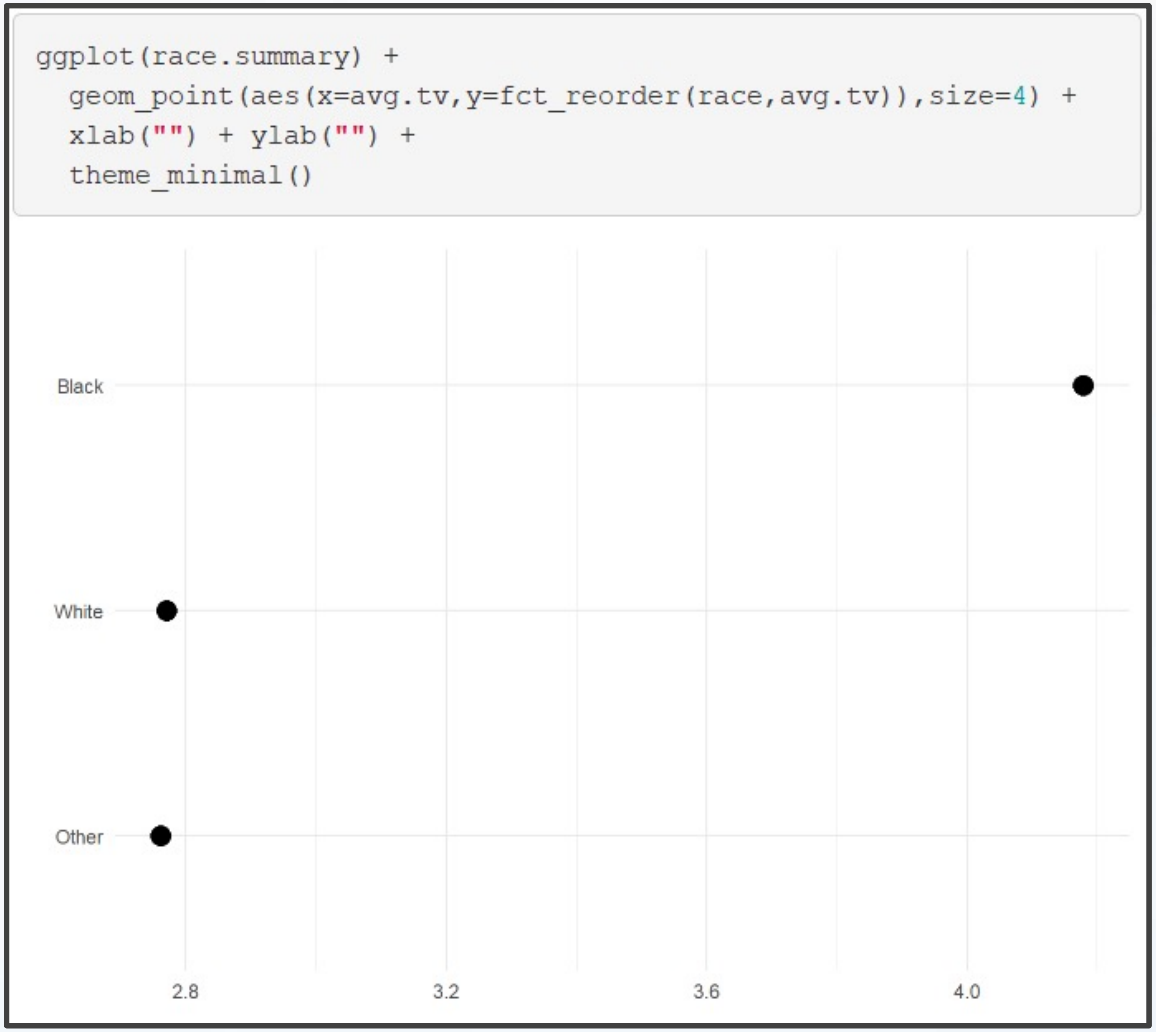
```
ggplot(race.summary) +
  geom_point(aes(x=avg.tv,y=race),size=4) +
  xlab("") + ylab("") +
  theme_minimal()
```

Modifying Factor Order

- fct_reorder()
    - f = Factor Variable
    - x = Numeric Vector
    - fun = Optional Function If Multiple Values of x for Each Value of f (Default: Median)

Modifying Factor Order

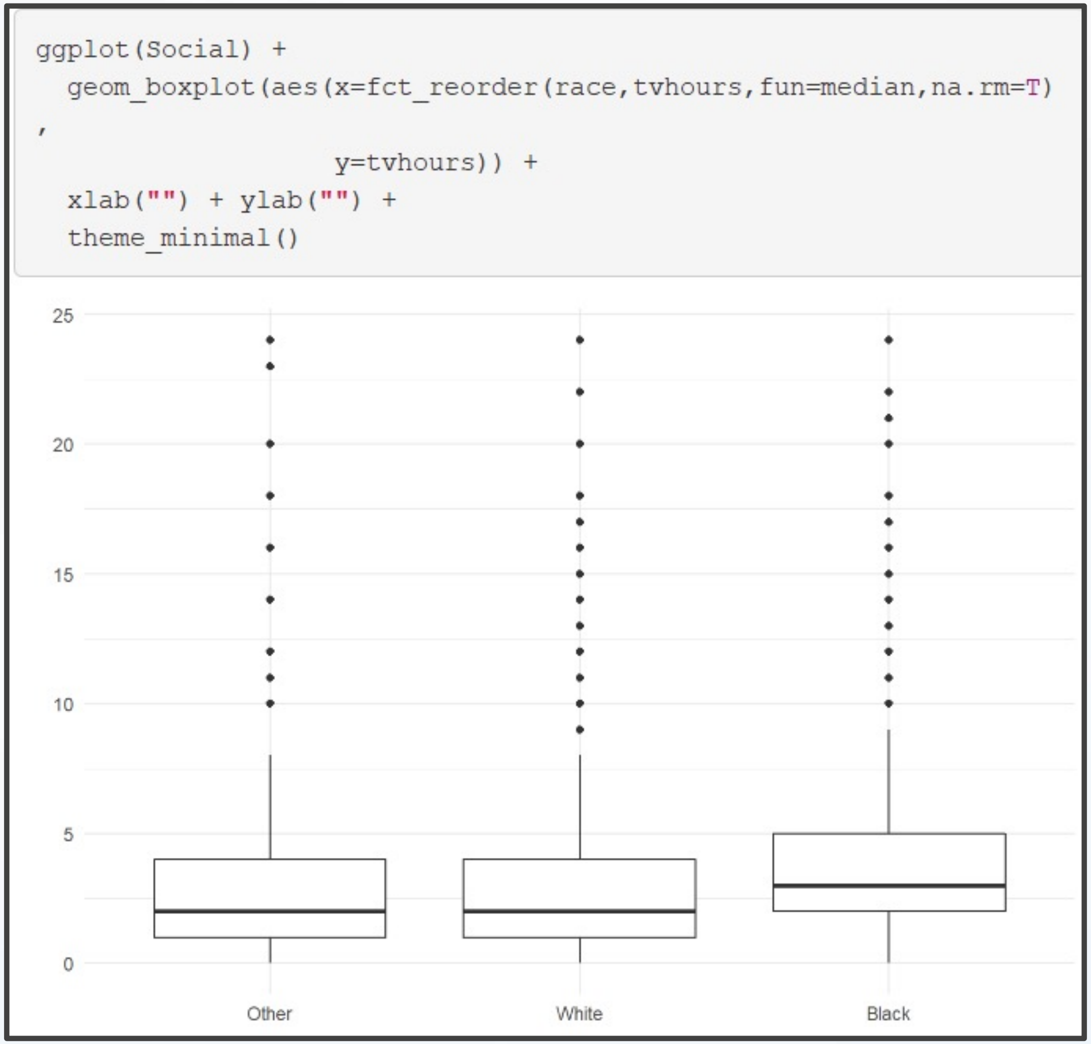
- Example 1: Reorder



```
ggplot(race.summary) +
  geom_point(aes(x=avg.tv,y=fct_reorder(race,avg.tv)),size=4) +
  xlab("") + ylab("") +
  theme_minimal()
```

Modifying Factor Order: Example 2

- Example 2: Reorder



```
ggplot(Social) +
  geom_boxplot(aes(x=fct_reorder(race,tvhours,fun=median,na.rm=T)
,
                  y=tvhours)) +
  xlab("") + ylab("") +
  theme_minimal()
```

Useful Functions

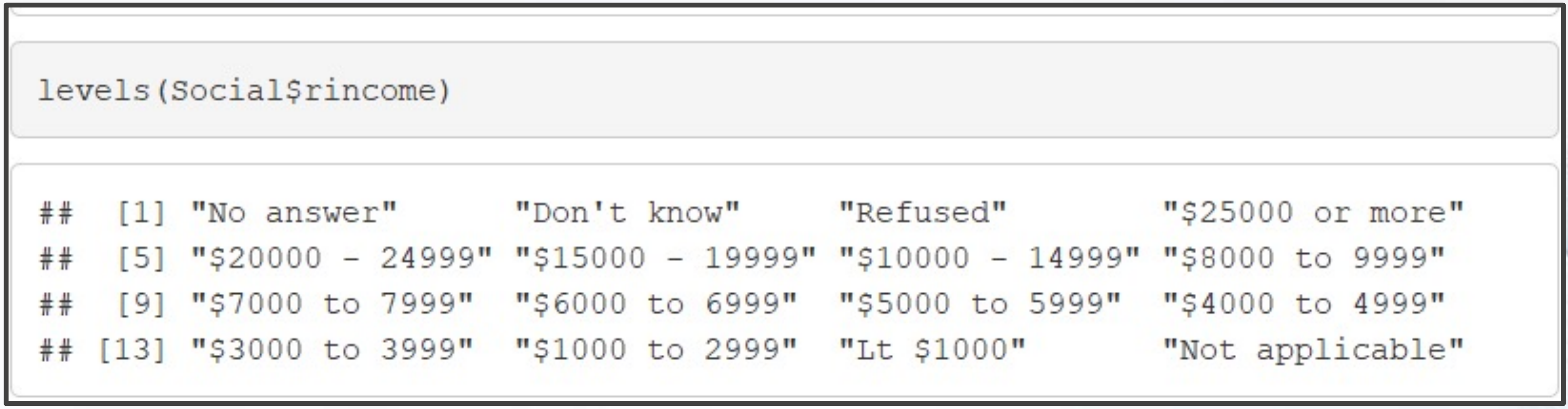
- Other Useful Functions
  - fct_relevel() = Specify Variable and the Specific Levels You Want in The Front
  - fct_rev() = Specify Variable and Reverses the Level Order
  - fct_infreq() = Order Levels Based on Increasing Frequency
- Combine Functions as Necessary

Types of Ordering

- Different Types of Ordering
  - Nominal = “Arbitrary”
  - Ordinal = “Principled”
- Example: Race vs Income
  - Race Levels are Arbitrary
  - Income Levels are Principled

Modifying Factor Order: Example 3

- Income Levels are Principled

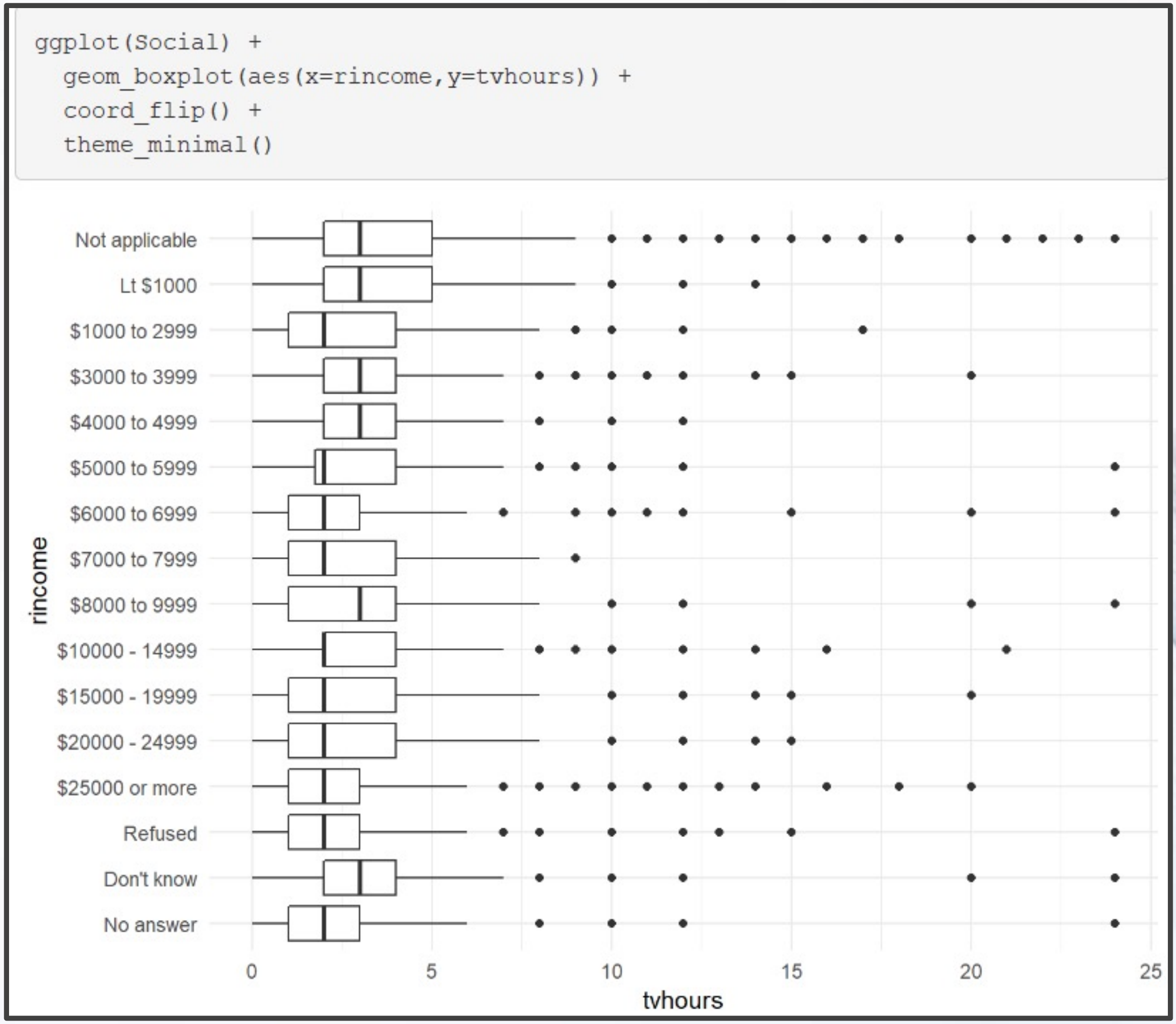


```
levels(Social$rincome)
```

```
##  [1] "No answer"      "Don't know"     "Refused"        "$25000 or more"
##  [5] "$20000 - 24999" "$15000 - 19999" "$10000 - 14999" "$8000 to 9999"
##  [9] "$7000 to 7999"  "$6000 to 6999"  "$5000 to 5999"  "$4000 to 4999"
## [13] "$3000 to 3999"  "$1000 to 2999"  "Lt $1000"       "Not applicable"
```

Modifying Factor Order: Example 3

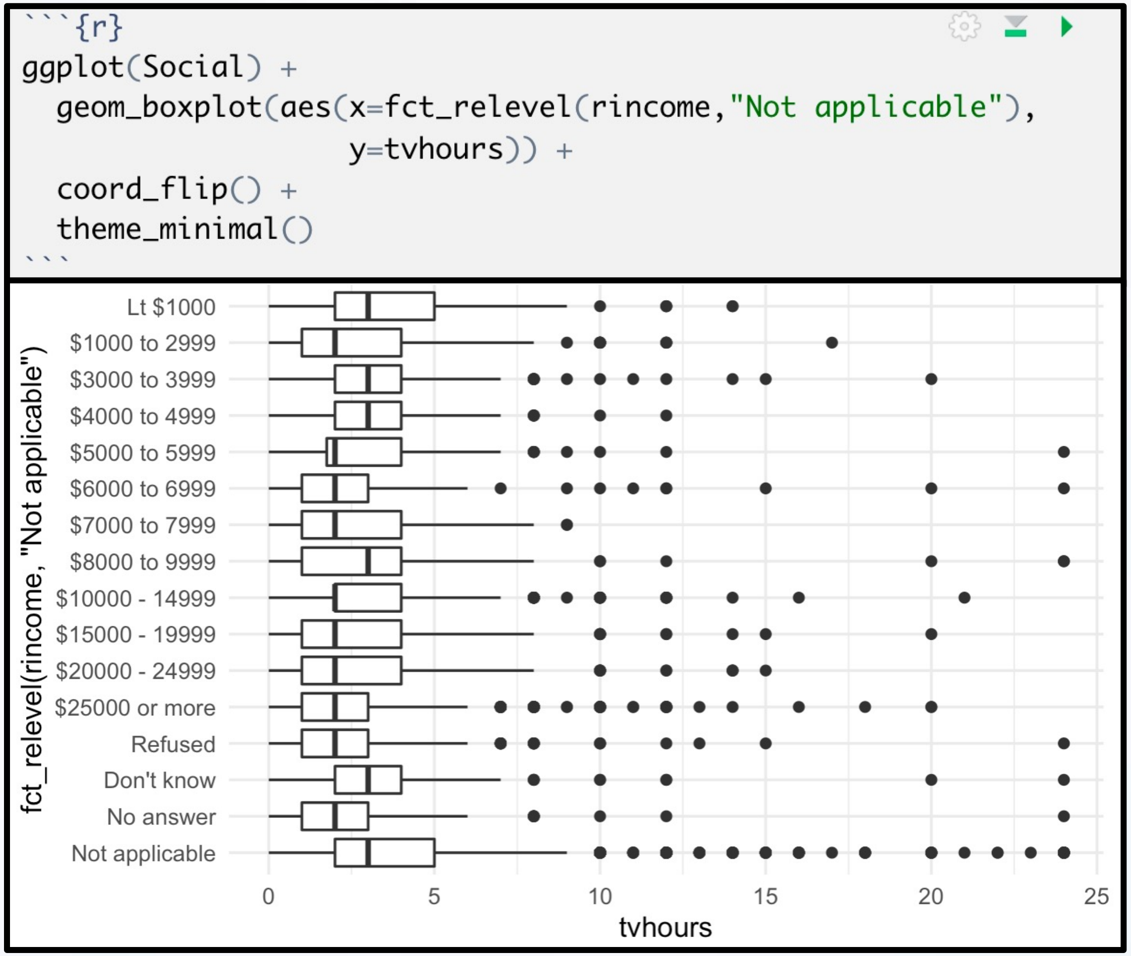
- Original Boxplot



```
ggplot(Social) +
  geom_boxplot(aes(x=rincome,y=tvhours)) +
  coord_flip() +
  theme_minimal()
```

Modifying Factor Order: Example 3

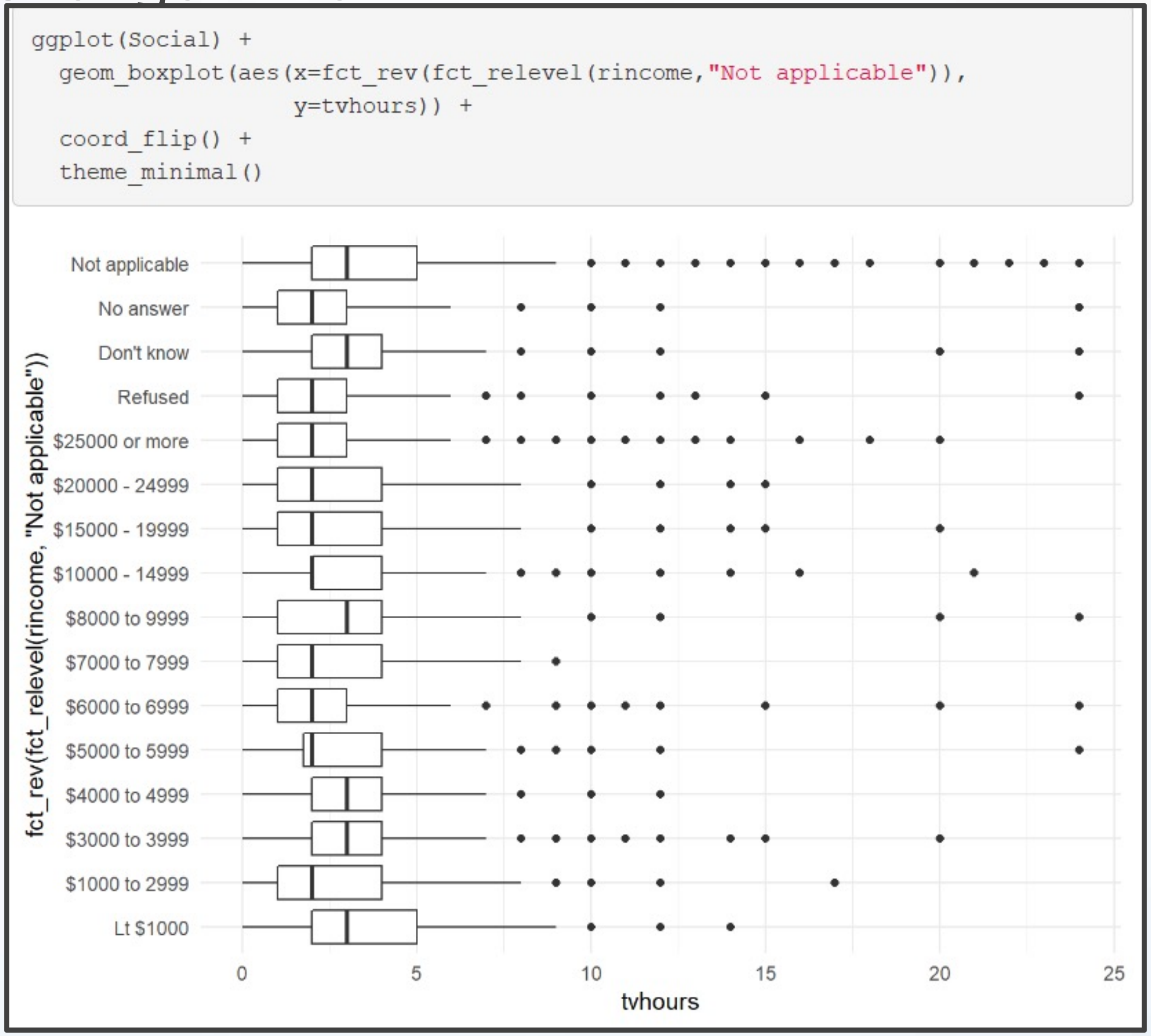
- Pull `Not applicable` to the front



~~~
```{r}
ggplot(Social) +
  geom_boxplot(aes(x=fct_relevel(rincome,"Not applicable"),
                   y=tvhours)) +
  coord_flip() +
  theme_minimal()
```
~~~

Modifying Factor Order: Example 3

- Level Change + Rev



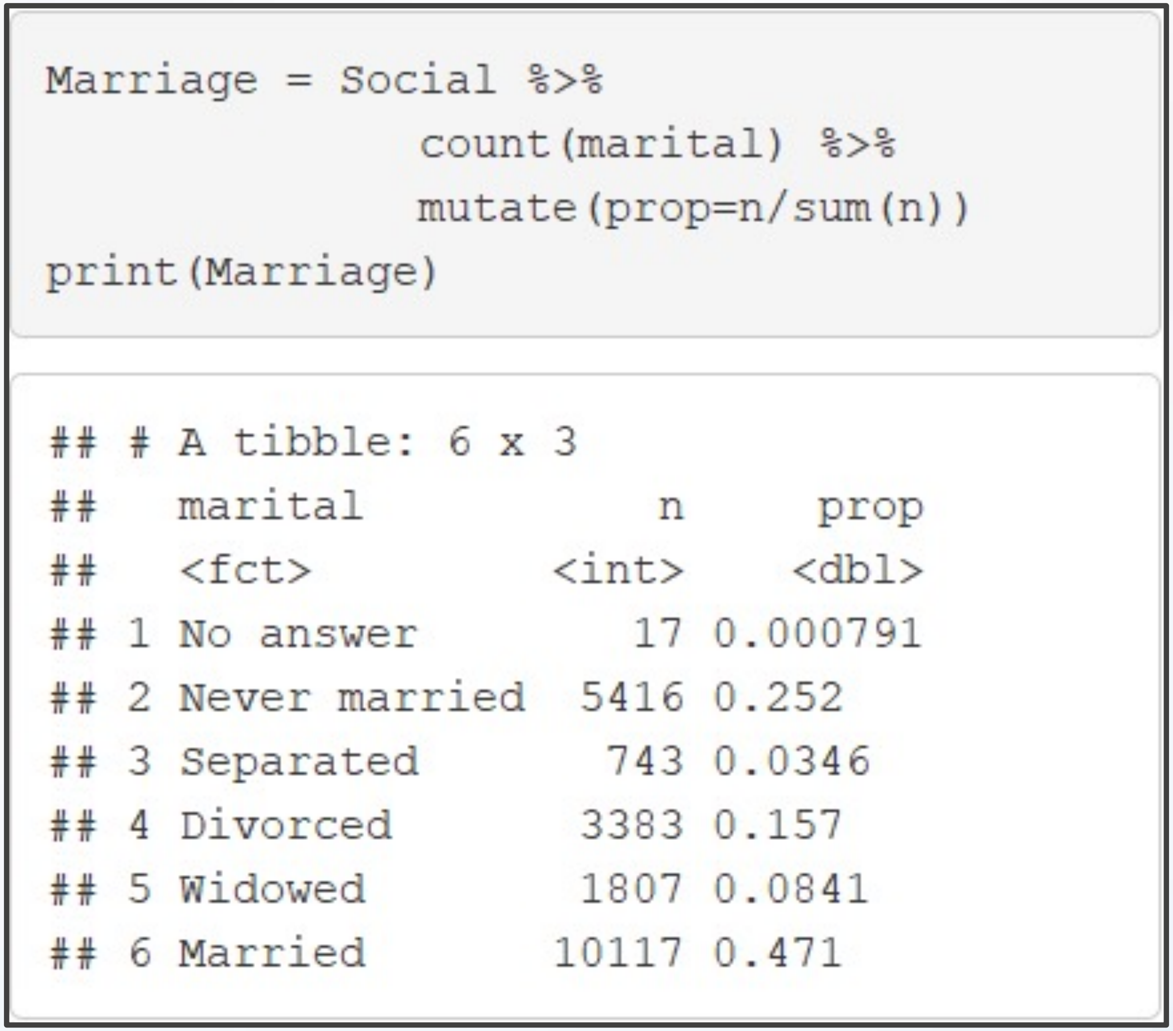
```
ggplot(Social) +
  geom_boxplot(aes(x=fct_rev(fct_relevel(rincome,"Not applicable")),
                   y=tvhours)) +
  coord_flip() +
  theme_minimal()
```

Modifying Factor Levels

- Purpose for Modifying Levels
  - Abbreviate or Better Names
  - Collapse Unimportant Levels
  - Group Categories
- Useful Functions
  - fct_recode() = Rename Levels
  - fct_collapse() = Collapse Levels
  - fct_lump() = Automatically Group Levels

Modifying Factor Levels

- Marital Counts

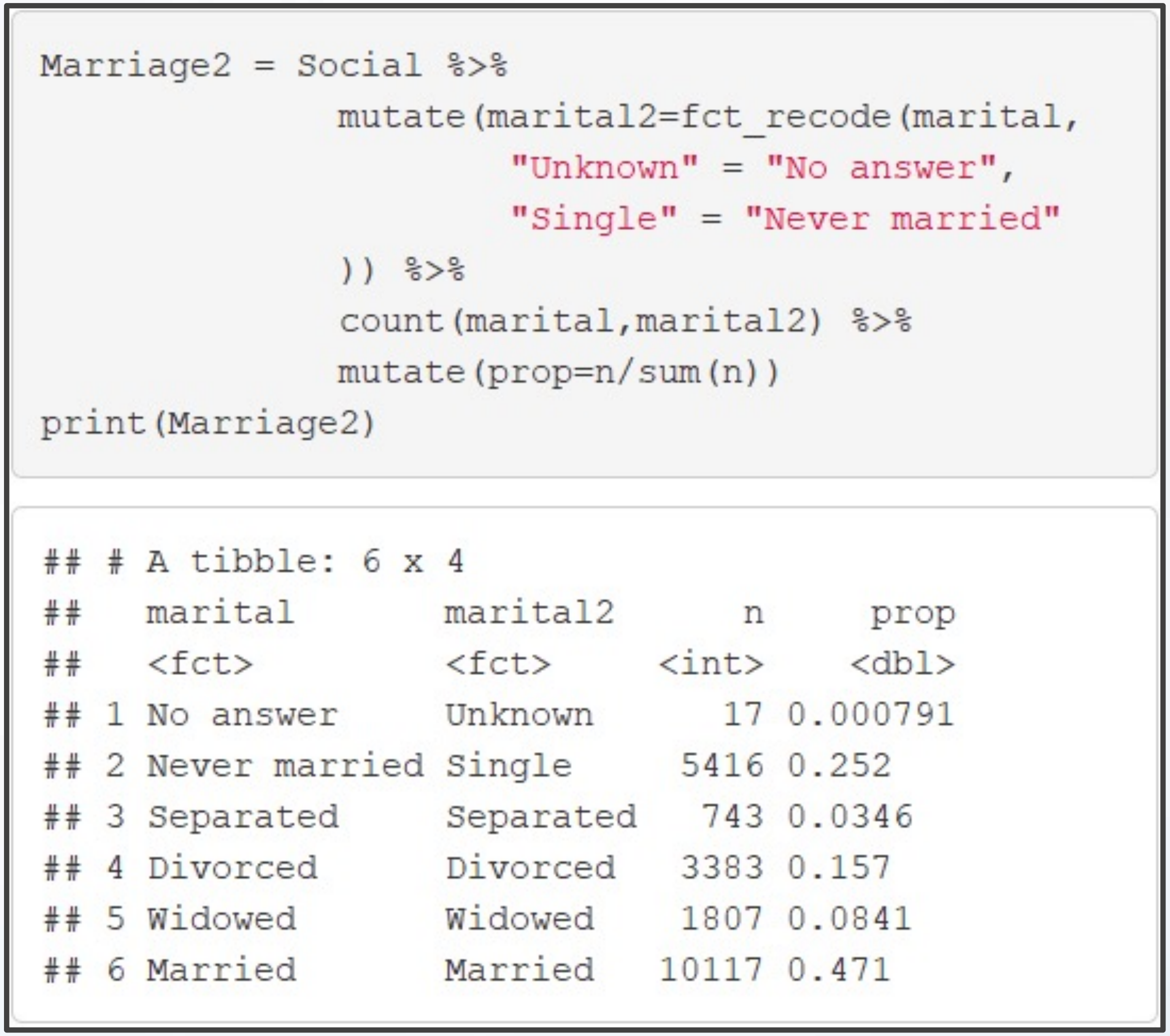


```
Marriage = Social %>%
              count(marital) %>%
              mutate(prop=n/sum(n))
print(Marriage)
```

```
## # A tibble: 6 x 3
##   marital                n     prop
##   <fct>             <int>    <dbl>
## 1 No answer            17 0.000791
## 2 Never married      5416 0.252
## 3 Separated           743 0.0346
## 4 Divorced           3383 0.157
## 5 Widowed            1807 0.0841
## 6 Married           10117 0.471
```

Recode Levels

- Example 1: Recode Levels

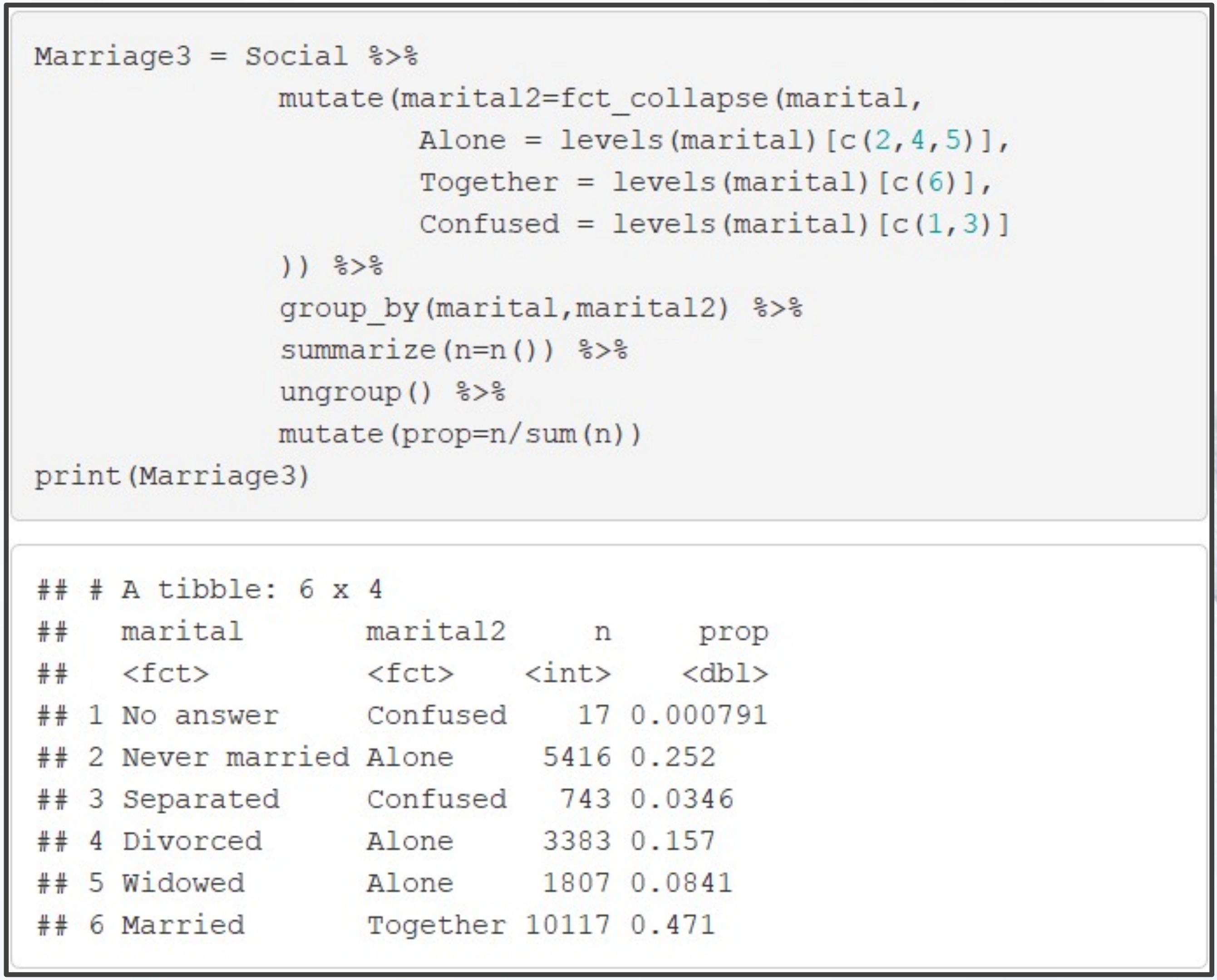


```
Marriage2 = Social %>%
              mutate(marital2=fct_recode(marital,
                      "Unknown" = "No answer",
                      "Single" = "Never married"
              )) %>%
              count(marital,marital2) %>%
              mutate(prop=n/sum(n))
print(Marriage2)
```

```
## # A tibble: 6 x 4
##   marital       marital2      n     prop
##   <fct>         <fct>     <int>    <dbl>
## 1 No answer     Unknown      17 0.000791
## 2 Never married Single     5416 0.252
## 3 Separated     Separated   743 0.0346
## 4 Divorced      Divorced   3383 0.157
## 5 Widowed       Widowed    1807 0.0841
## 6 Married       Married   10117 0.471
```

Collapse Levels

- Example 2: Collapse Levels

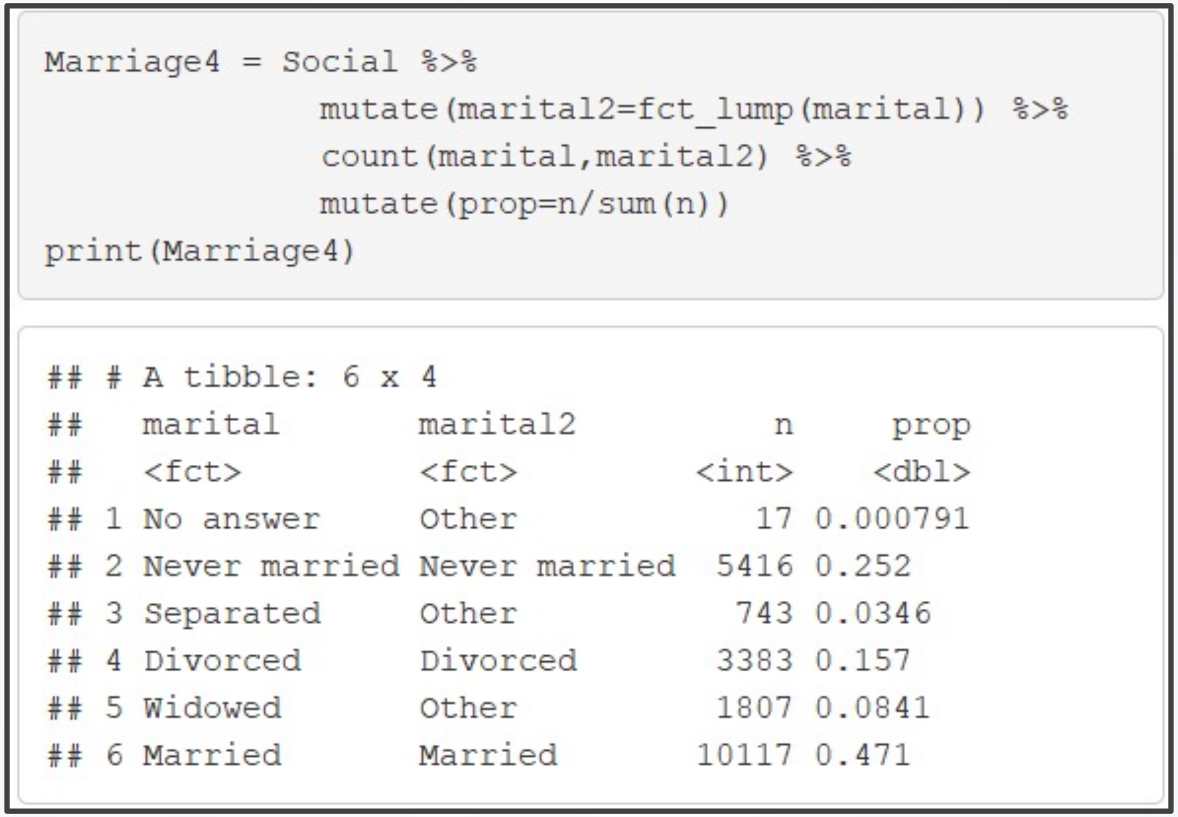


```
Marriage3 = Social %>%
              mutate(marital2=fct_collapse(marital,
                      Alone = levels(marital)[c(2,4,5)],
                      Together = levels(marital)[c(6)],
                      Confused = levels(marital)[c(1,3)]
              )) %>%
              group_by(marital,marital2) %>%
              summarize(n=n()) %>%
              ungroup() %>%
              mutate(prop=n/sum(n))
print(Marriage3)
```

```
## # A tibble: 6 x 4
##   marital       marital2     n     prop
##   <fct>         <fct>    <int>    <dbl>
## 1 No answer     Confused    17 0.000791
## 2 Never married Alone     5416 0.252
## 3 Separated     Confused   743 0.0346
## 4 Divorced      Alone     3383 0.157
## 5 Widowed       Alone     1807 0.0841
## 6 Married       Together 10117 0.471
```

Lumping Levels

- Example 3: Lumping Levels

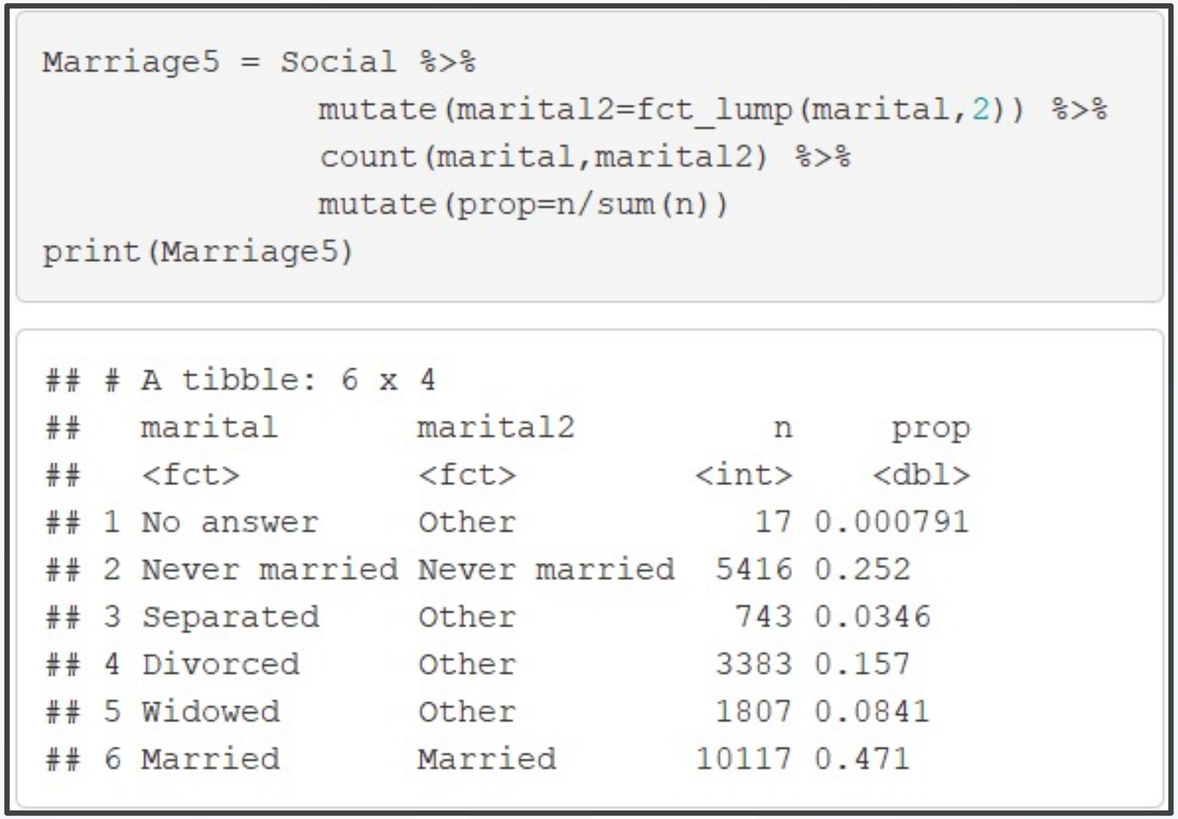


```
Marriage4 = Social %>%
              mutate(marital2=fct_lump(marital)) %>%
              count(marital,marital2) %>%
              mutate(prop=n/sum(n))
print(Marriage4)
```

```
## # A tibble: 6 x 4
##   marital       marital2          n     prop
##   <fct>         <fct>         <int>    <dbl>
## 1 No answer     Other            17 0.000791
## 2 Never married Never married  5416 0.252
## 3 Separated     Other           743 0.0346
## 4 Divorced      Divorced       3383 0.157
## 5 Widowed       Other          1807 0.0841
## 6 Married       Married       10117 0.471
```

Lumping Levels

- Example 3: Lumping Levels



```
Marriage5 = Social %>%
              mutate(marital2=fct_lump(marital,2)) %>%
              count(marital,marital2) %>%
              mutate(prop=n/sum(n))
print(Marriage5)
```

```
## # A tibble: 6 x 4
##   marital       marital2          n     prop
##   <fct>         <fct>         <int>    <dbl>
## 1 No answer     Other            17 0.000791
## 2 Never married Never married  5416 0.252
## 3 Separated     Other           743 0.0346
## 4 Divorced      Other          3383 0.157
## 5 Widowed       Other          1807 0.0841
## 6 Married       Married       10117 0.471
```