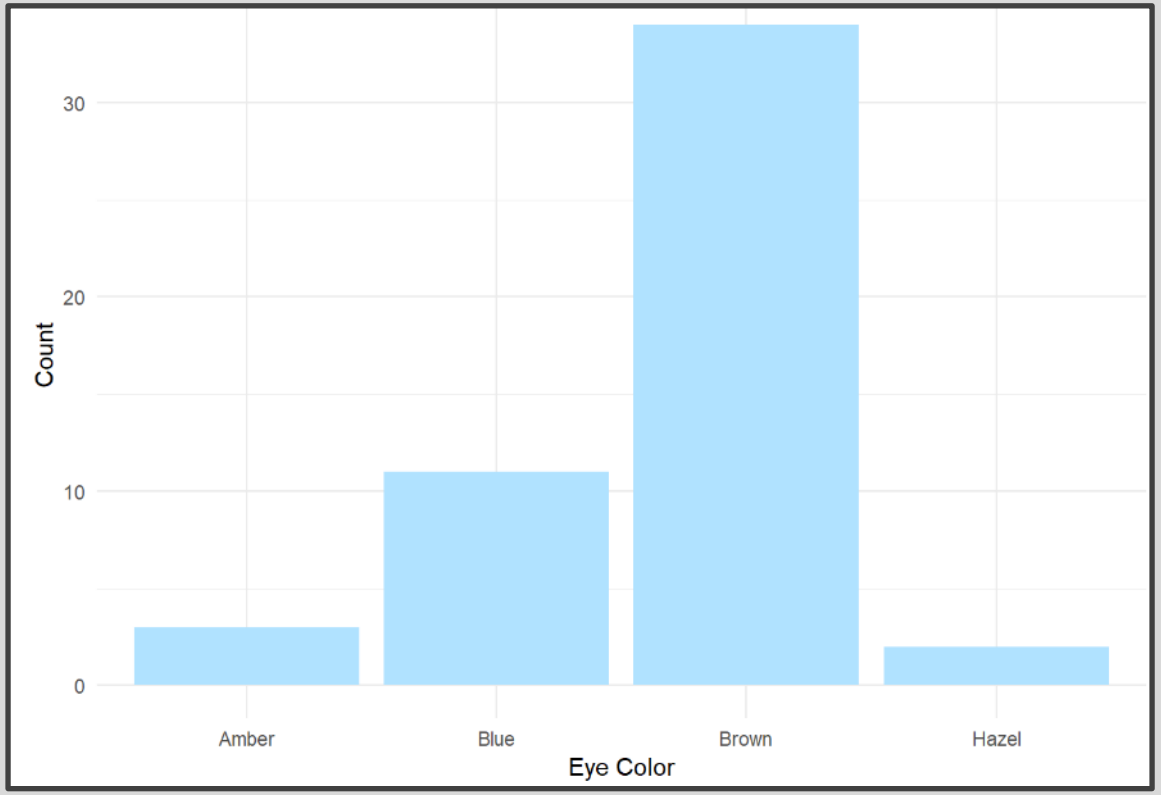
Factors

Introduction

- Joyfully Read Chapter 12 (R4DS) and Chapter 16 (R4DS2)
- Additional Package
  - `> library(forcats)`
  - Not Part of the tidyverse
- For Variables with,
  - Fixed Set of Values
  - Known Set of Values
- Sophisticated Character Vector
- Factors Are on a New Level

Level 1: Motivation

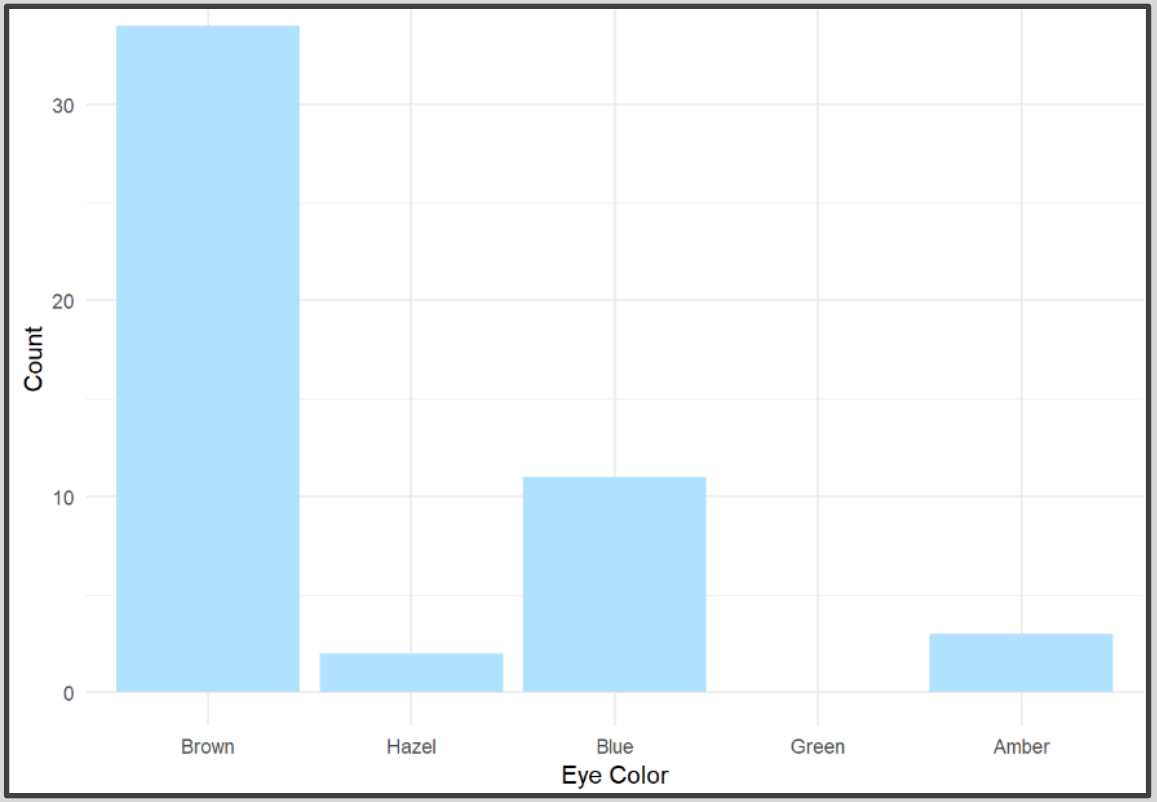
- Eye Color Distribution
  - Randomly Sample 50 People
  - Distribution via Bar Plot



- How to Make More Informative?

Level 1: Motivation

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

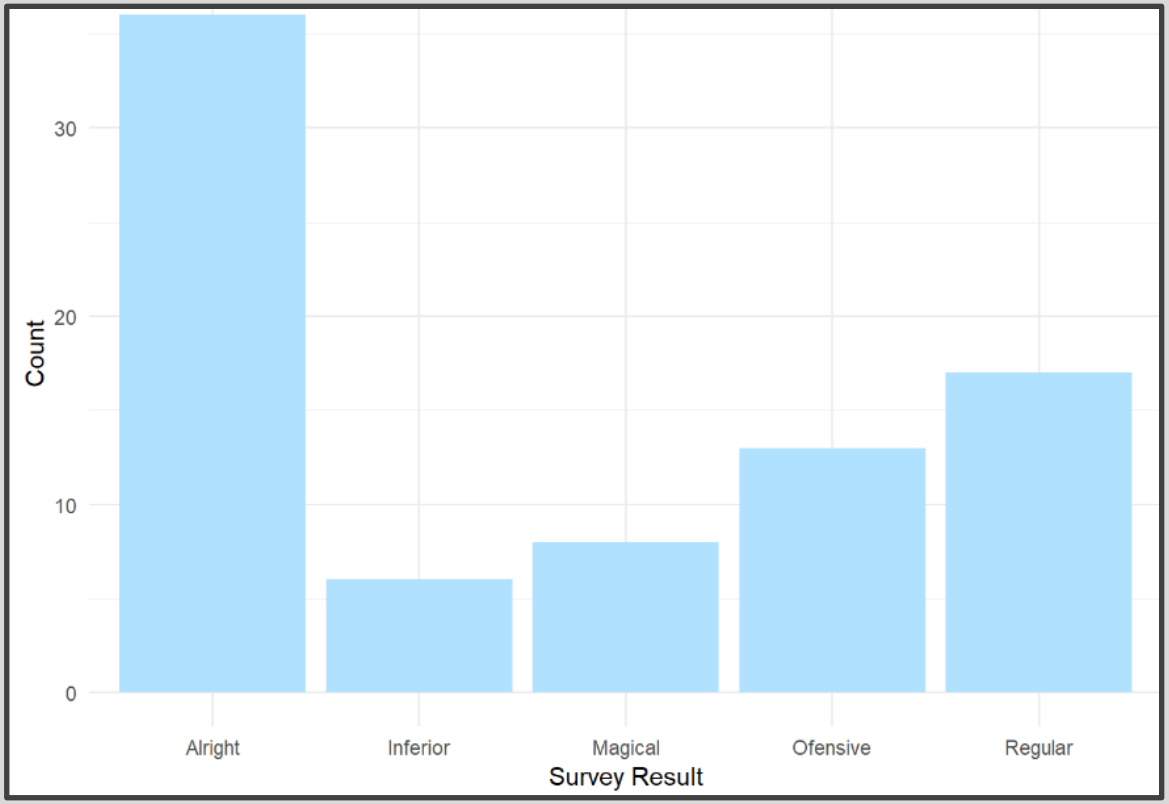


Level 1: Motivation

- Survey Results
  - How Would You Describe Dr. Mario's Teaching?
    - Magical
    - Alright
    - Regular
    - Inferior
    - Offensive
  - Class of 80 Students Answer End-of-the-Year Survey

Level 1:
Motivation

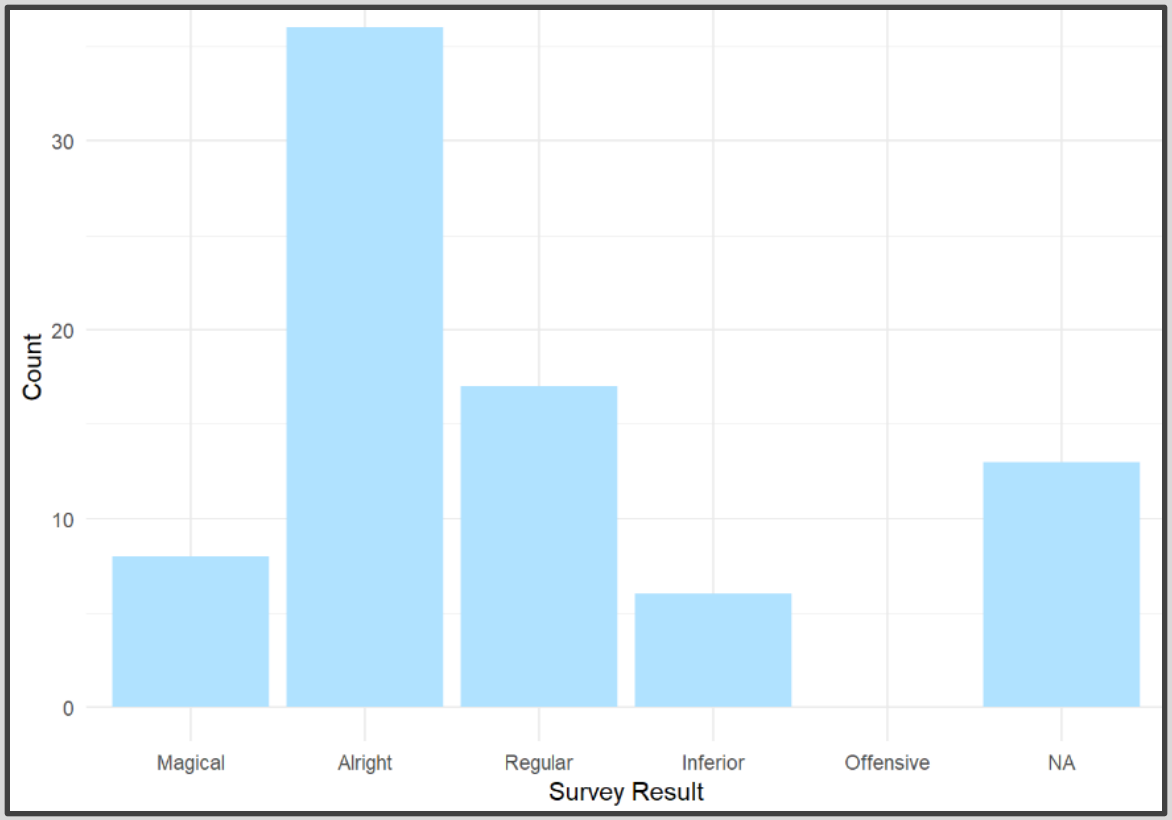
- Survey Results (Cont.)
  - Distribution of Results



  - What is Wrong?

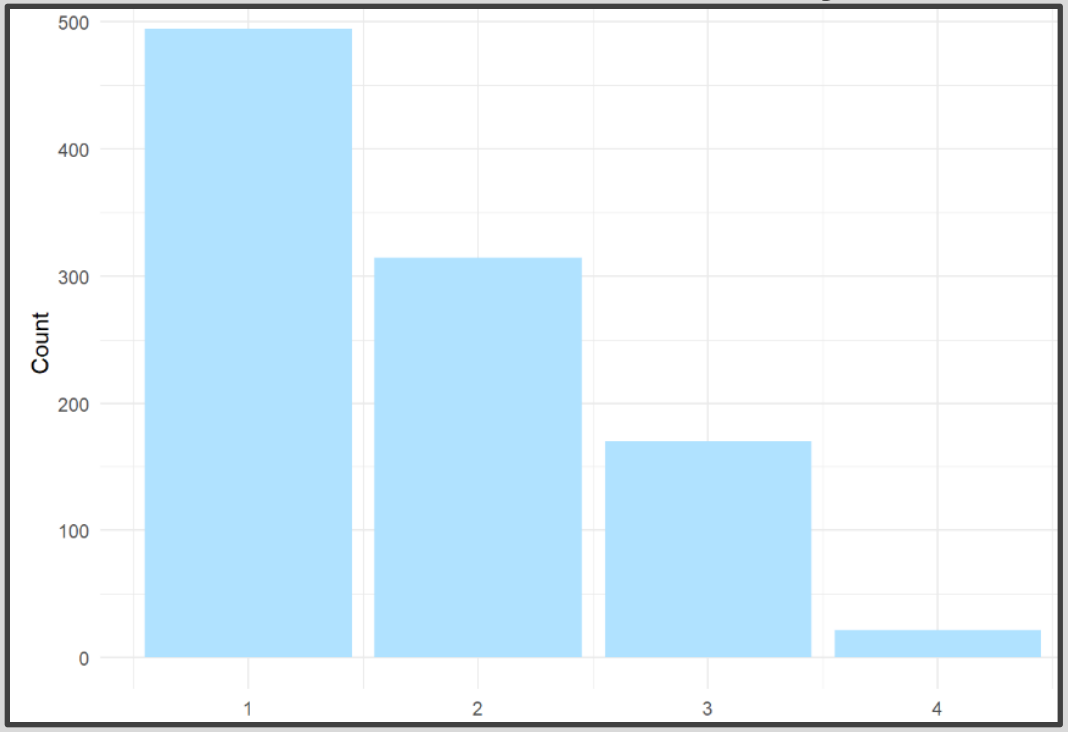
Level 1: Motivation

- Survey Results (Cont.)
  - Misspelling “Offensive” is Offensive
  - Ordinal Categorical Variable



Level 1: Motivation

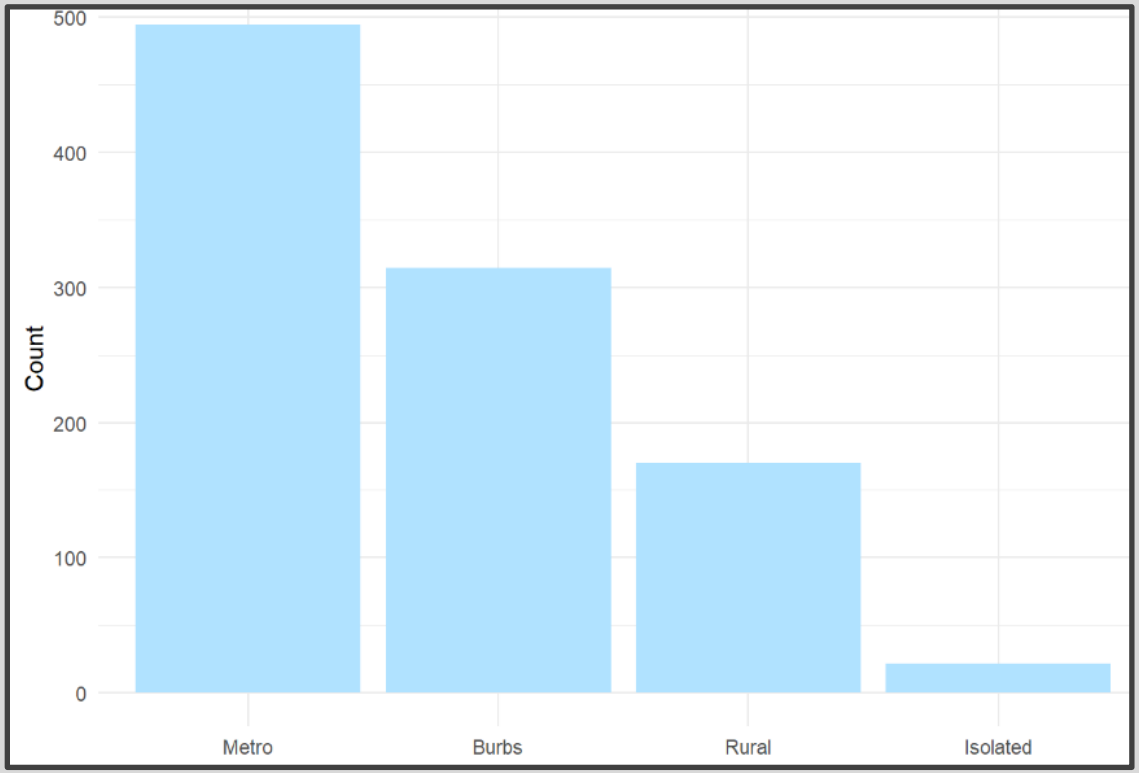
- Urbanicity
  - Classification {1,2,3,4}
  - Sample 1000 Households and Record Their Urbanicity



  - What Would Make this Better?

Level 1: Motivation

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



Level 2: 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

Level 2: Factor Variable Architecture

- 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
```

Level 2: Factor Variable Architecture

- 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)
```

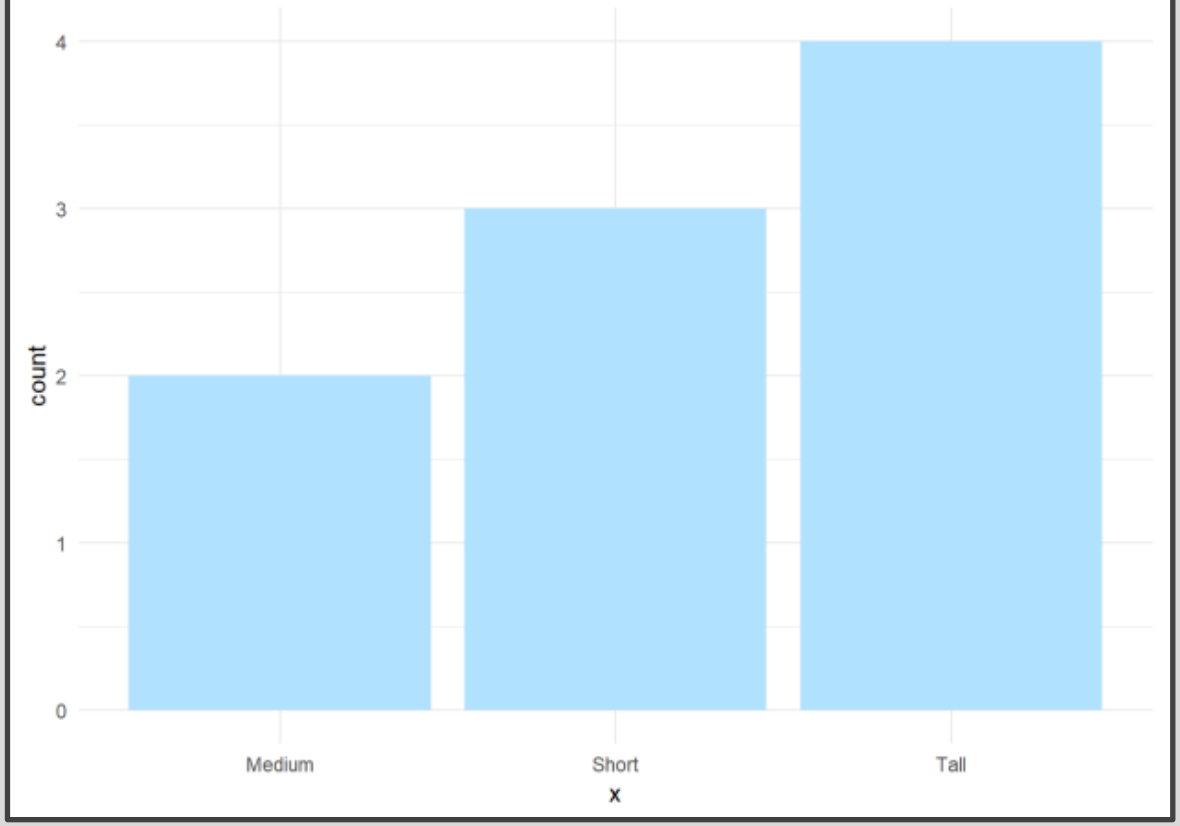
```
## [1] Not Short Short     Not Short Not Short Short     Not Short Short

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

Level 2: Factor Variable Architecture

- Graphic Comparison

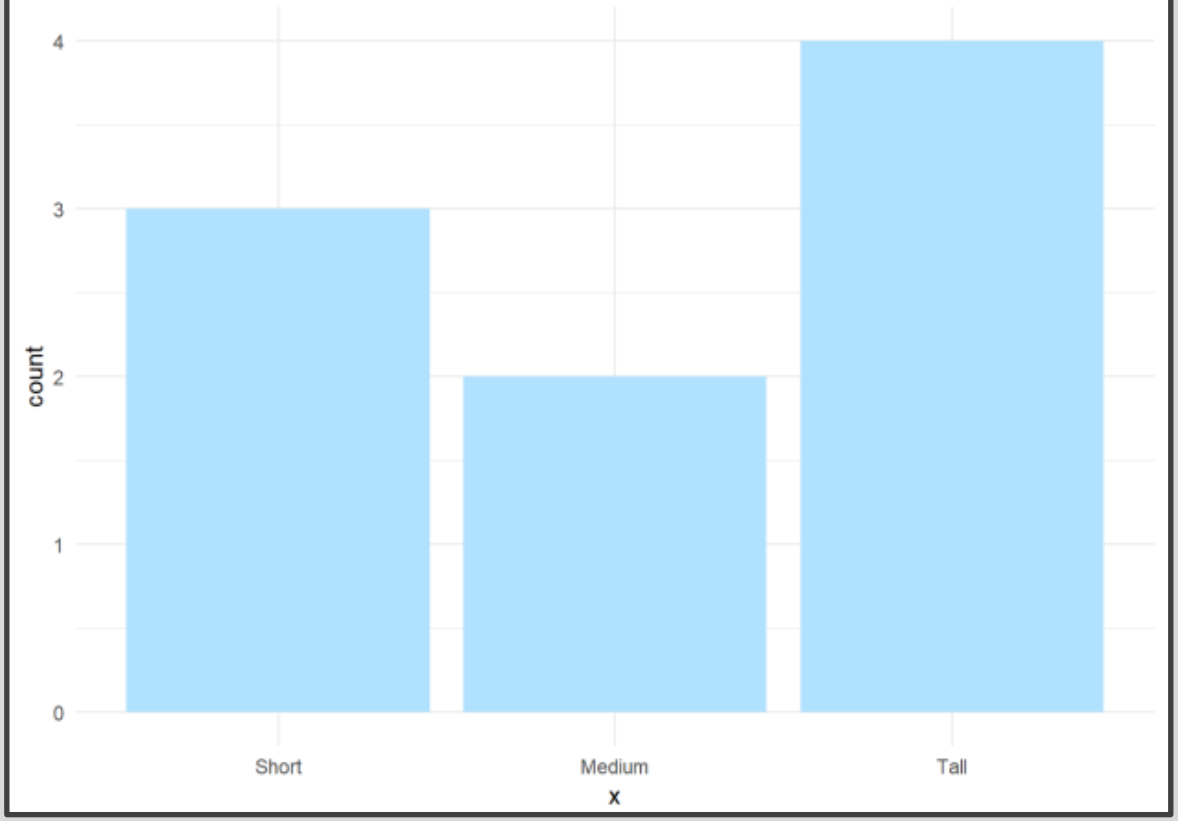
```
ggplot(data=tibble(x=Height.fct)) +
  geom_bar(aes(x),fill="lightskyblue1") +
  theme_minimal()
```



Level 2: Factor Variable Architecture

- Graphic Comparison

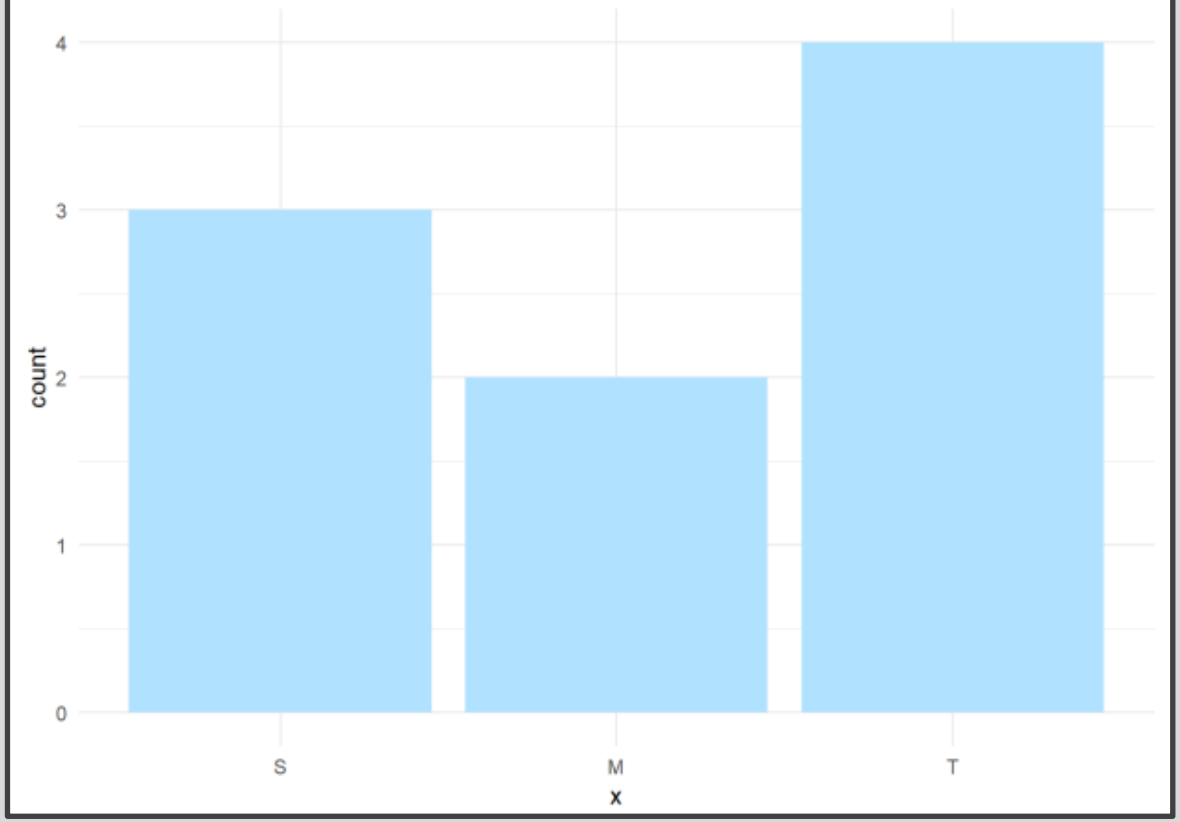
```
ggplot(data=tibble(x=Height2.fct)) +
  geom_bar(aes(x),fill="lightskyblue1") +
  theme_minimal()
```



Level 2: Factor Variable Architecture

- Graphic Comparison

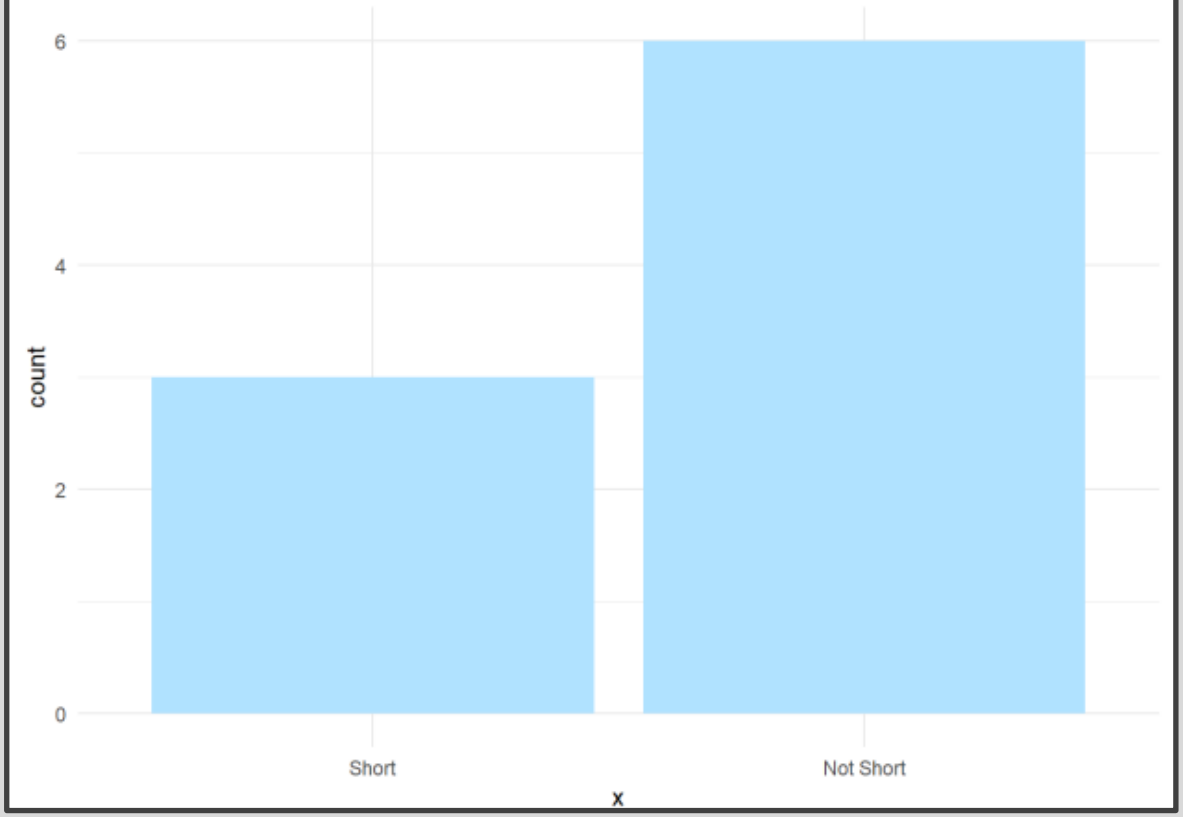
```
ggplot(data=tibble(x=Height3.fct)) +
  geom_bar(aes(x),fill="lightskyblue1") +
  theme_minimal()
```



Level 2: Factor Variable Architecture

- Graphic Comparison

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



Level 3:
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

Level 3: General Social Survey

- Sample Provided in forcats

```
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, ...
```

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

Level 4: 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"
```

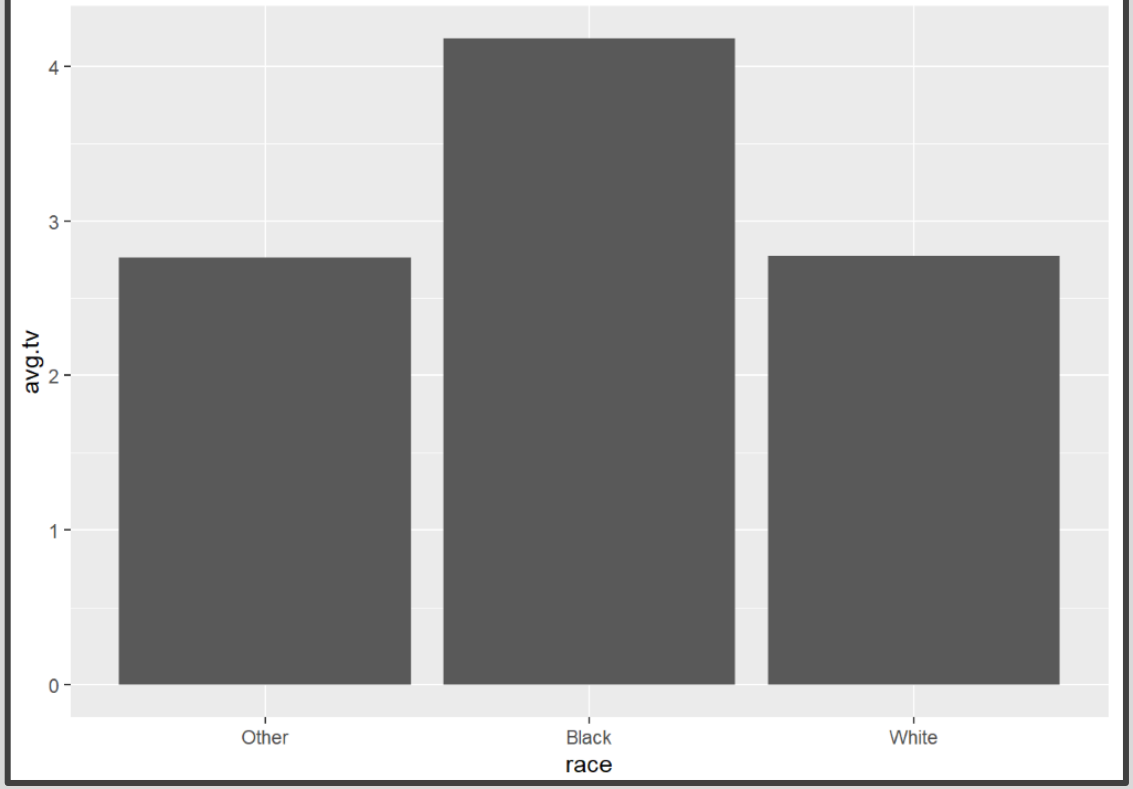
```
levels(race.summary$race)
```

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

Level 4: Modifying Factor Order

- Comparing TV Hours

```
ggplot(race.summary) +
  geom_bar(aes(y=avg.tv,x=race),stat="identity",size=4)
```



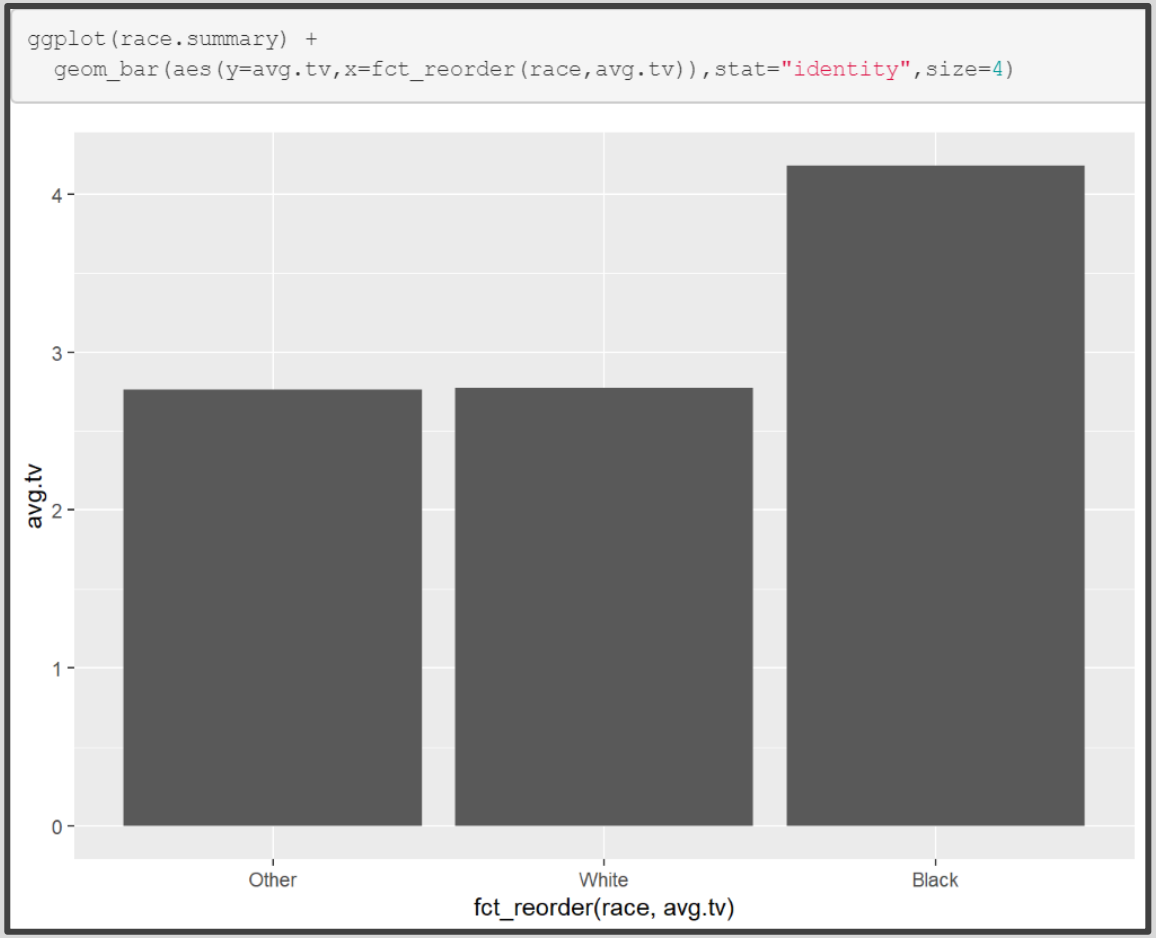
Level 4: Modifying Factor Order

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

Level 4: Modifying Factor Order

- Example 1: Reorder

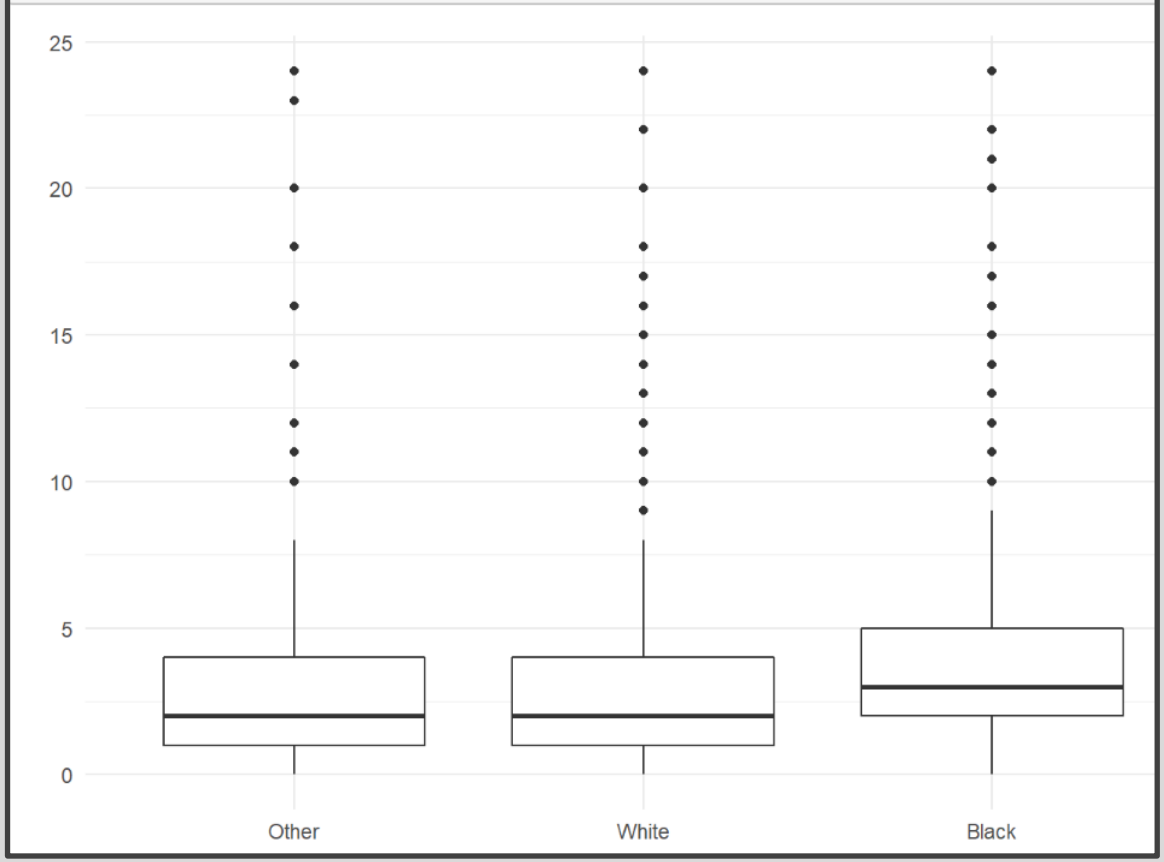
```
ggplot(race.summary) +
  geom_bar(aes(y=avg.tv,x=fct_reorder(race,avg.tv)),stat="identity",size=4)
```



Level 4: Modifying Factor Order

- Example 2: Reorder

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



Level 4: Modifying Factor Order

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

Level 4: Modifying Factor Order

```
head(Social[,c("race","rincome")])
```

```
## # A tibble: 6 x 2
##   race  rincome
##   <fct> <fct>
## 1 White $8000 to 9999
## 2 White $8000 to 9999
## 3 White Not applicable
## 4 White Not applicable
## 5 White Not applicable
## 6 White $20000 - 24999
```

```
str(Social[,c("race","rincome")])
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    21483 obs. of  2 variables:
##  $ race   : Factor w/ 4 levels "Other","Black",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ rincome: Factor w/ 16 levels "No answer","Don't know",..: 8 8 16 16 16 5
 4 9 4 4 ...
```

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

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

```
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"
```

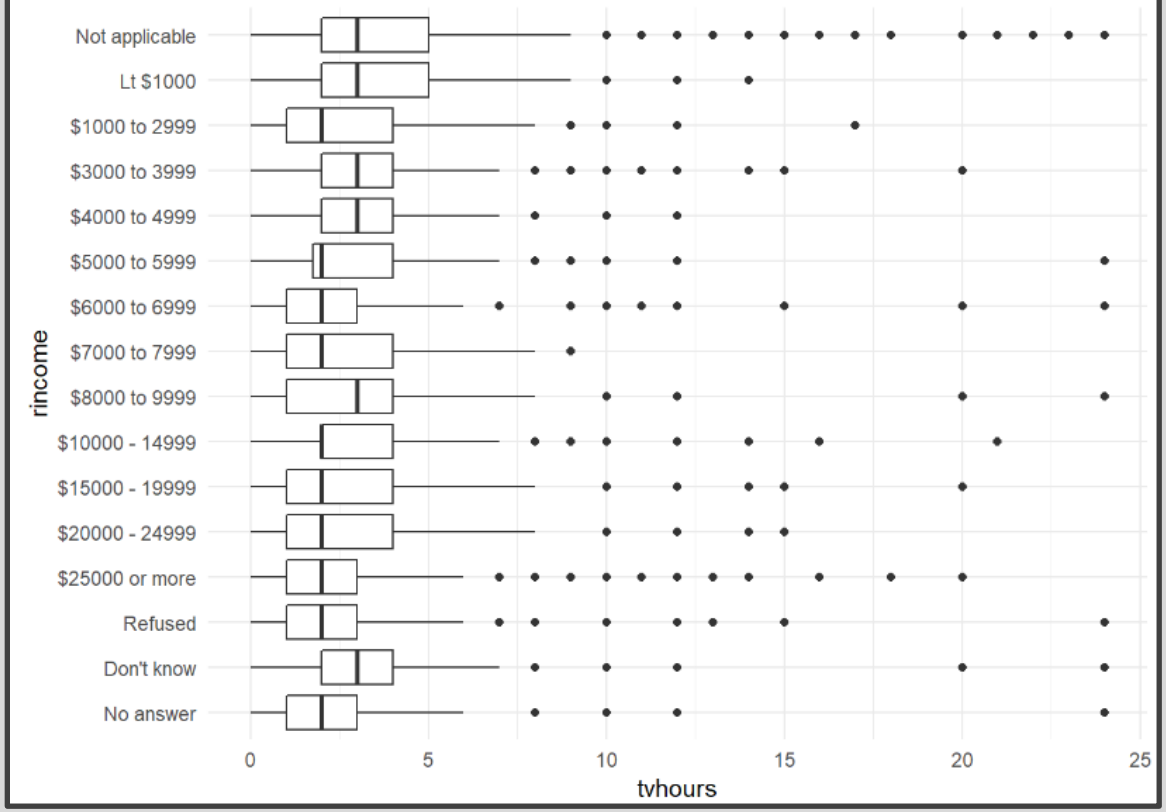
Level 4: Modifying Factor Order

- 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

Level 4: Modifying Factor Order

- Original Boxplot

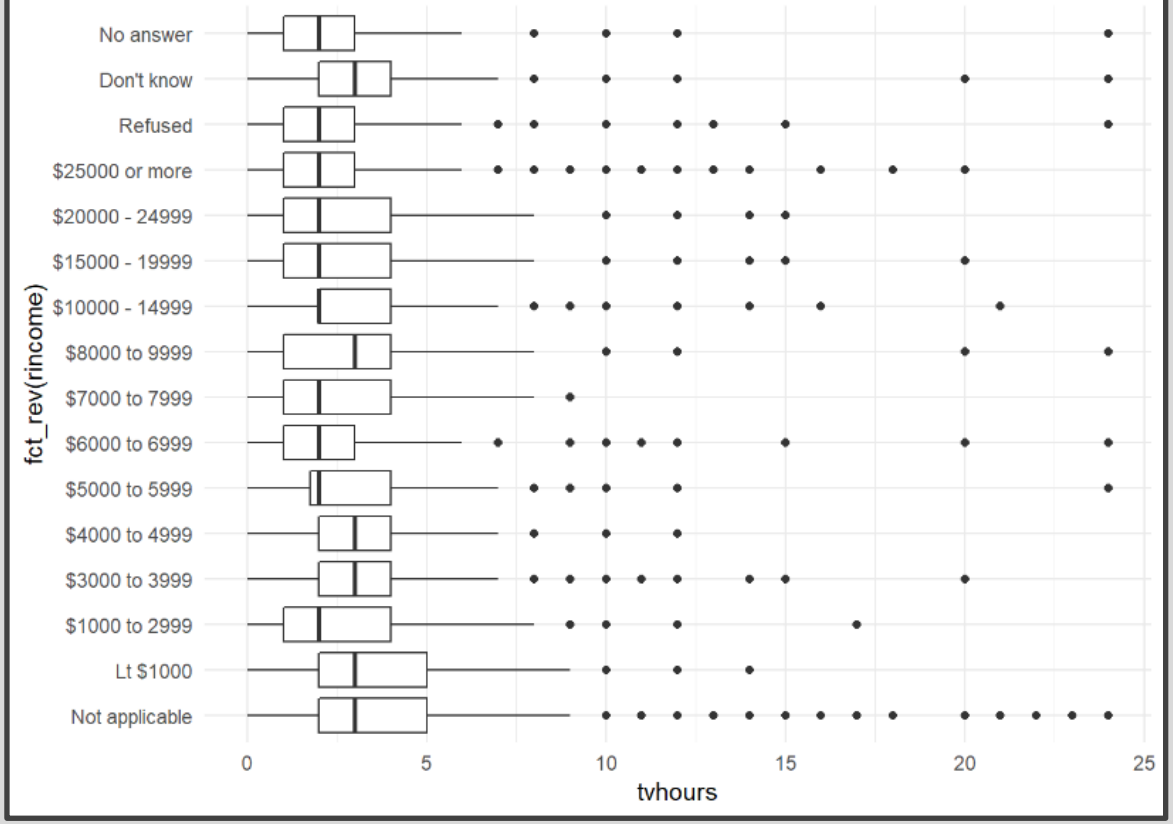
```
ggplot(Social) +
  geom_boxplot(aes(x=rincome,y=tvhours)) +
  coord_flip() +
  theme_minimal()
```



Level 4: Modifying Factor Order

- Example 1: Reverse Income

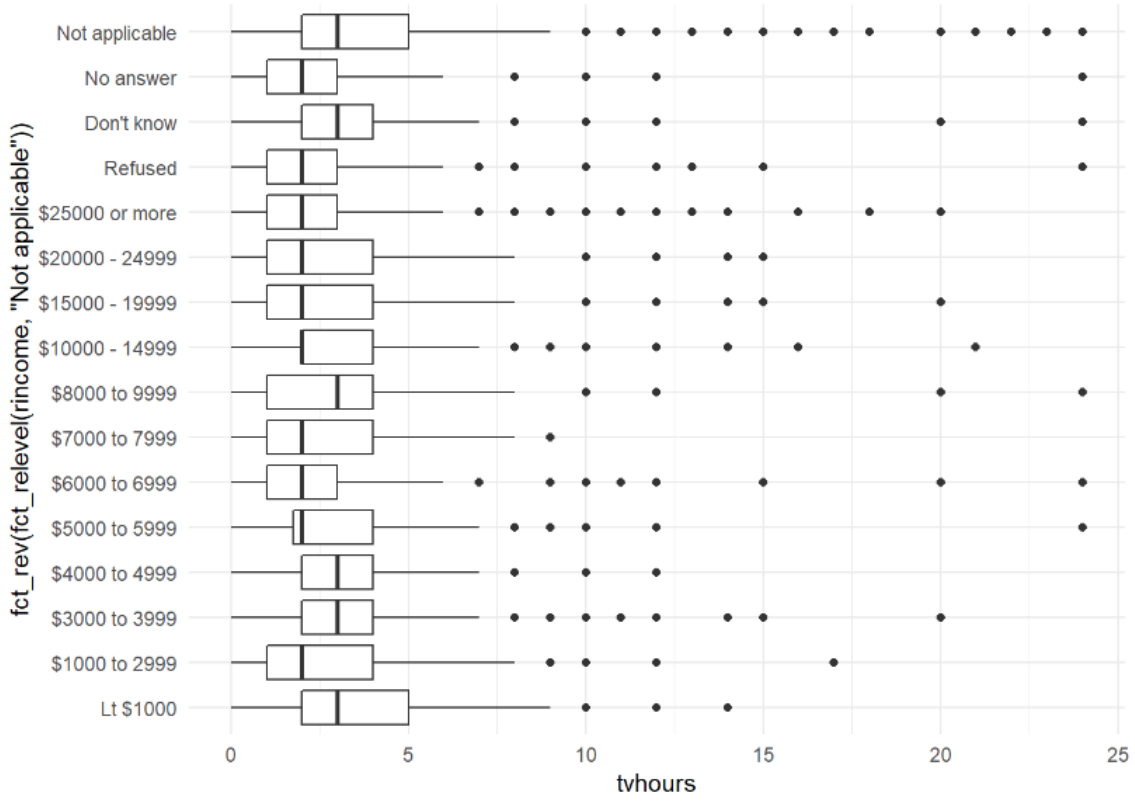
```
ggplot(Social) +
  geom_boxplot(aes(x=fct_rev(rincome),y=tvhours)) +
  coord_flip() +
  theme_minimal()
```



Level 4: Modifying Factor Order

- Example 2: Level Change + Rev

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



Level 5: 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() = Create Subgroups

Level 5: 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
```

Level 5: Modifying Factor 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
```

Level 5: Modifying Factor Levels

- Example 2: Collapse Levels

```
levels(Social$marital)
```

```
## [1] "No answer"     "Never married" "Separated"     "Divorced"

## [5] "Widowed"       "Married"
```

```
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
```

Level 5: Modifying Factor 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
```

Level 5: Modifying Factor 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
```

Level 6: Numeric to Factor

- Cut Function
  - Convert Numeric to Factor
  - Syntax

    > cut(VARIABLE, # of Breaks)

  - Useful In Visuals and Summary

- Example 1: New Age Variable

```
NewAge = Social %>%
            mutate(new.age=cut(age,3))

str(NewAge)
```

```
tibble [21,483 x 10] (S3: tbl_df/tbl/data.frame)
 $ year   : int [1:21483] 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
 $ marital: Factor w/ 6 levels "No answer","Never married",..: 2 4 5 2 4 6 2 4 6 6 ...
 $ age    : int [1:21483] 26 48 67 39 25 25 36 44 44 47 ...
 $ race   : Factor w/ 4 levels "Other","Black",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ rincome: Factor w/ 16 levels "No answer","Don't know",..: 8 8 16 16 16 5 4 9 4 4 ...
 $ partyid: Factor w/ 10 levels "No answer","Don't know",..: 6 5 7 6 9 10 5 8 9 4 ...
 $ relig  : Factor w/ 16 levels "No answer","Don't know",..: 15 15 15 6 12 15 5 15 15 15 ...
 $ denom  : Factor w/ 30 levels "No answer","Don't know",..: 25 23 3 30 30 25 30 15 4 25 ...
 $ tvhours: int [1:21483] 12 NA 2 4 1 NA 3 NA 0 3 ...
 $ new.age: Factor w/ 3 levels "(17.9,41.7]",..: 1 2 3 1 1 1 1 2 2 2 ...
[1] "(17.9,41.7]" "(41.7,65.3]" "(65.3,89.1]"
```

```
levels(NewAge$new.age)
```

```
[1] "(17.9,41.7]" "(41.7,65.3]" "(65.3,89.1]"
```

Level 6: Numeric to Factor

- Example 2: Make It Pretty

```
NewAge = Social %>%
            mutate(new.age=cut(age,pretty(age,3)))

levels(NewAge$new.age)
```

```
[1] "(0,20]"   "(20,40]"  "(40,60]"  "(60,80]"  "(80,100]"
```

What Happened?

- Example 3: Label Levels

```
NewAge = Social %>%
            mutate(new.age=cut(age,3,
                               labels=c("Young","Middle","Old")))

levels(NewAge$new.age)
```

```
[1] "Young"  "Middle" "Old"
```

Level 6: Numeric to Factor

- Example 4: Using Percentiles
  - Goal: Cut on the Quartiles
  - Use Quantile Function

```
NewAge = Social %>%
            mutate(new.age=cut(age,
                       quantile(Social$age,c(0,0.25,0.5,0.75,1),na.rm=T)))

levels(NewAge$new.age)
```

```
[1] "(18,33]" "(33,46]" "(46,59]" "(59,89]"
```

- Helpful Package **> library(expss)**

```
NewAge=select(NewAge,age,marital) %>%
        apply_labels(age="Age",marital="Marital Status")

cro_cases(NewAge$age,list(NewAge$marital,total()))
```

	Marital Status						#Total
	No answer	Never married	Separated	Divorced	Widowed	Married	
Age							
(17.9,35.8]	1	3490	197	413	26	2241	6368
(35.8,53.5]	5	1364	350	1546	156	4175	7596
(53.5,71.2]	3	462	163	1169	578	2817	5192
(71.2,89.1]	1	86	31	236	1040	857	2251
#Total cases	10	5402	741	3364	1800	10090	21407

Closing

Disperse and Make Reasonable Decisions