Code for QSS tidyverse Chapter 2: Causality

Kosuke Imai and Nora Webb Williams

First Printing

Causality

Racial Discrimination in the Labor Market

```
## Load required packages
library(tidyverse)
library(qss)

## Load in the data from QSS package
data(resume, package = "qss")
```

```
## Or read in a saved CSV
resume <- read_csv("resume.csv")
```

```
##
## -- Column specification ------------------------------------
## cols(
##   firstname = col_character(),
##   sex = col_character(),
##   race = col_character(),
##   call = col_double()
## )
```

```
resume <- read_csv("resume.csv",
                   col_types = cols(
                     firstname = col_character(),
                     sex = col_character(),
                     race = col_character(),
                     call = col_number()))
```

```
dim(resume)
```

```
## [1] 4870    4
```

```
head(resume)
```

```
## # A tibble: 6 x 4
##   firstname sex    race   call
##   <chr>     <chr>  <chr> <dbl>
## 1 Allison   female white     0
## 2 Kristen   female white     0
## 3 Lakisha   female black     0
## 4 Latonya   female black     0
## 5 Carrie    female white     0
## 6 Jay       male   white     0
```

```
tail(resume)
```

```
## # A tibble: 6 x 4
##   firstname sex    race   call
##   <chr>     <chr>  <chr> <dbl>
## 1 Lakisha   female black     0
## 2 Tamika    female black     0
## 3 Ebony     female black     0
## 4 Jay       male   white     0
## 5 Latonya   female black     0
## 6 Laurie    female white     0
```

```
glimpse(resume)
```

```
## Rows: 4,870
## Columns: 4
## $ firstname <chr> "Allison", "Kristen", "Lakisha", "Latony~
## $ sex       <chr> "female", "female", "female", "female", ~
## $ race      <chr> "white", "white", "black", "black", "whi~
## $ call      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
summary(resume)
```

```
##   firstname             sex                race          
##  Length:4870        Length:4870        Length:4870       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##                                                          
##                                                          
##                                                          
##       call        
##  Min.   :0.00000  
##  1st Qu.:0.00000  
##  Median :0.00000  
##  Mean   :0.08049  
##  3rd Qu.:0.00000  
##  Max.   :1.00000  
```

```
race.call.summary <- resume %>%
  group_by(race, call) %>% # create for each race and callback status
  count()

race.call.summary
```

```
## # A tibble: 4 x 3
## # Groups:   race, call [4]
##   race   call     n
##   <chr> <dbl> <int>
## 1 black     0  2278
## 2 black     1   157
## 3 white     0  2200
## 4 white     1   235
```

```
race.call.tab <- race.call.summary %>%
  pivot_wider(names_from = call, # reshape the data
              values_from = n)

race.call.tab
```

```
## # A tibble: 2 x 3
## # Groups:   race [2]
##   race    ‘0‘   ‘1‘
##   <chr> <int> <int>
## 1 black  2278   157
## 2 white  2200   235
```

```
race.call.tab.names <- race.call.tab %>%
    rename(no_callback = "0",
         callback = "1")

race.call.tab.names
```

```
## # A tibble: 2 x 3
## # Groups:   race [2]
##   race  no_callback callback
##   <chr>       <int>    <int>
## 1 black        2278      157
## 2 white        2200      235
```

```
race.call.tab.names <- race.call.tab.names %>%
  mutate(total_resumes = no_callback + callback,
         callback_prop = callback / total_resumes)

race.call.tab.names
```

```
## # A tibble: 2 x 5
## # Groups:   race [2]
##   race  no_callback callback total_resumes callback_prop
##   <chr>       <int>    <int>         <int>         <dbl>
## 1 black        2278      157          2435        0.0645
## 2 white        2200      235          2435        0.0965
```

```
overall_callback <- resume %>%
  summarize(total_callback_rate = sum(call) / n())

overall_callback
```

```
## # A tibble: 1 x 1
##   total_callback_rate
##                 <dbl>
## 1              0.0805
```

```
overall_callback <- resume %>%
  summarize(total_callback_rate = mean(call))

overall_callback
```

```
## # A tibble: 1 x 1
##   total_callback_rate
##                 <dbl>
## 1              0.0805
```

```
callback_by_race <- resume %>%
  group_by(race) %>%
  summarize(callback_rate = mean(call))

callback_by_race
```

```
## # A tibble: 2 x 2
##   race  callback_rate
##   <chr>         <dbl>
## 1 black        0.0645
## 2 white        0.0965
```

Subsetting Data in R

Logical Values and Operators

```
class(TRUE)
```

```
## [1] "logical"
```

```
as.integer(TRUE)
```

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

```
as.integer(FALSE)
```

```
## [1] 0
```

```
x <- c(TRUE, FALSE, TRUE) # a vector with logical values
mean(x) # proportion of TRUEs
```

```
## [1] 0.6666667
```

```r
sum(x) # number of TRUEs
```

```
## [1] 2
```

```r
FALSE & TRUE
```

```
## [1] FALSE
```

```r
TRUE & TRUE
```

```
## [1] TRUE
```

```r
TRUE | FALSE
```

```
## [1] TRUE
```

```r
FALSE | FALSE
```

```
## [1] FALSE
```

```r
TRUE & FALSE & TRUE
```

```
## [1] FALSE
```

```r
(TRUE | FALSE) & FALSE # the parentheses evaluate to TRUE
```

```
## [1] FALSE
```

```r
TRUE | (FALSE & FALSE) # the parentheses evaluate to FALSE
```

```
## [1] TRUE
```

```r
TF1 <- c(TRUE, FALSE, FALSE)
TF2 <- c(TRUE, FALSE, TRUE)
TF1 | TF2
```

```
## [1]  TRUE FALSE  TRUE
```

```r
TF1 & TF2
```

```
## [1]  TRUE FALSE FALSE
```

Relational Operators

```r
4 > 3
```

```
## [1] TRUE
```

```r
"Hello" == "hello"  # R is case-sensitive
```

```
## [1] FALSE
```

```r
"Hello" != "hello"
```

```
## [1] TRUE
```

```r
x <- c(3, 2, 1, -2, -1)
x >= 2
```

```
## [1]  TRUE  TRUE FALSE FALSE FALSE
```

```r
x != 1
```

```
## [1]  TRUE  TRUE FALSE  TRUE  TRUE
```

```r
## logical conjunction of two vectors with logical values
(x > 0) & (x <= 2)
```

```
## [1] FALSE  TRUE  TRUE FALSE FALSE
```

```r
## logical disjunction of two vectors with logical values
(x > 2) | (x <= -1)
```

```
## [1]  TRUE FALSE FALSE  TRUE  TRUE
```

```r
x.int <- (x > 0) & (x <= 2) # logical vector
x.int
```

```
## [1] FALSE  TRUE  TRUE FALSE FALSE
```

```r
mean(x.int) # proportion of TRUEs
```

```
## [1] 0.4
```

```r
sum(x.int)  # number of TRUEs
```

```
## [1] 2
```

Subsetting

```
## callback rate for black-sounding names
resume %>%
  filter(race == "black") %>% # only keep observations where "race" is black
  summarize(mean(call)) # take the average of call
```

```
## # A tibble: 1 x 1
##   ‘mean(call)‘
##          <dbl>
## 1       0.0645
```

```
## Subset the data with black names
resumeB <- filter(resume, race == "black")
## Calculate the mean callback rate
summarize(resumeB, mean(call))
```

```
## # A tibble: 1 x 1
##   ‘mean(call)‘
##          <dbl>
## 1       0.0645
```

```
## with $ operator to run mean() on the call column
mean(resumeB$call)
```

```
## [1] 0.06447639
```

```
resumeBf <- filter(resume, race == "black" & sex == "female")
```

```
## callback rate for black female names
Bf_callback <- filter(resume, race == "black" & sex == "female") %>%
  summarize(callback_rate = mean(call)) %>%
  pull()

## print the value to the console
print(Bf_callback)
```

```
## [1] 0.06627784
```

```
## callback rate for black male names
Bm_callback <- filter(resume, race == "black" & sex == "male") %>%
  summarize(callback_rate = mean(call)) %>%
  pull()

## print the value to the console
print(Bm_callback)
```

```
## [1] 0.0582878
## callback rate for white female names
Wf_callback <- filter(resume, race == "white" & sex == "female") %>%
  summarize(callback_rate = mean(call)) %>%
  pull()

## print the value to the console
print(Wf_callback)
```

```
## [1] 0.09892473
```

```
## callback rate for white male names
Wm_callback <- filter(resume, race == "white" & sex == "male") %>%
  summarize(callback_rate = mean(call)) %>%
  pull()

## print the value to the console
print(Wm_callback)
```

```
## [1] 0.08869565
```

```
## difference between white women and black women
Wf_callback - Bf_callback
```

```
## [1] 0.03264689
```

```
## difference between white men and black men
Wm_callback - Bm_callback
```

```
## [1] 0.03040786
```

```
racial_gaps_by_sex <- resume %>%
  group_by(race, sex) %>% # using two variables to group the data
  summarize(callback = mean(call)) %>% # the callback rate for each group
  pivot_wider(names_from = race, # reshaping the data
              values_from = callback) %>%
  mutate(race_gap = white - black)

print(racial_gaps_by_sex)
```

```
## # A tibble: 2 x 4
##   sex     black  white race_gap
##   <chr>   <dbl>  <dbl>    <dbl>
## 1 female 0.0663 0.0989   0.0326
## 2 male   0.0583 0.0887   0.0304
```

```
## what happens in this portion of the code?
resume %>%
  group_by(race, sex) %>%
  summarize(callback = mean(call))
```

```
## # A tibble: 4 x 3
## # Groups:   race [2]
##   race  sex    callback
##   <chr> <chr>     <dbl>
## 1 black female   0.0663
## 2 black male     0.0583
## 3 white female   0.0989
## 4 white male     0.0887
```

```
## What happens after we add the pivot_wider()?
resume %>%
  group_by(race, sex) %>%
  summarize(callback = mean(call)) %>%
  pivot_wider(names_from = race,
              values_from = callback)
```

```
## # A tibble: 2 x 3
##   sex     black  white
##   <chr>   <dbl>  <dbl>
## 1 female 0.0663 0.0989
## 2 male   0.0583 0.0887
```

```
## And so on
```

Simple Conditional Statements

```
resume <- resume %>%
  ## create a new variable that is 1 if the resume is black and female,
  ## 0 otherwise
  mutate(BlackFemale = if_else(race == "black" &
                               sex == "female", 1, 0))
```

```
# Rows in the resumeBf data
nrow(resumeBf)
```

```
## [1] 1886
```

```
# Is that equal to sum of BlackFemale?
nrow(resumeBf) == resume %>% summarize(bf = sum(BlackFemale))
```

```
##        bf
## [1,] TRUE
```

Factor Variables

```
resume_fact <- resume %>%
  mutate(type = if_else(race == "black" & sex == "female", "BlackFemale", ""),
         type = if_else(race == "black" & sex == "male", "BlackMale", type),
         type = if_else(race == "white" & sex == "female", "WhiteFemale", type),
         type = if_else(race == "white" & sex == "male", "WhiteMemale", type))

head(resume_fact)
```

```
## # A tibble: 6 x 6
##   firstname sex    race   call BlackFemale type
##   <chr>     <chr>  <chr> <dbl>       <dbl> <chr>
## 1 Allison   female white     0           0 WhiteFemale
## 2 Kristen   female white     0           0 WhiteFemale
## 3 Lakisha   female black     0           1 BlackFemale
## 4 Latonya   female black     0           1 BlackFemale
## 5 Carrie    female white     0           0 WhiteFemale
## 6 Jay       male   white     0           0 WhiteMemale
```

```
resume <- resume %>%
  ## add a categorical variable for race/gender type
  mutate(type = case_when(race == "black" & sex == "female" ~ "BlackFemale",
                          race == "white" & sex == "female" ~ "WhiteFemale",
                          race == "black" & sex == "male" ~ "BlackMale",
                          race == "white" & sex == "male" ~ "WhiteMale",
                          TRUE ~ "other"
))

head(resume)
```

```
## # A tibble: 6 x 6
##   firstname sex    race   call BlackFemale type
##   <chr>     <chr>  <chr> <dbl>       <dbl> <chr>
## 1 Allison   female white     0           0 WhiteFemale
## 2 Kristen   female white     0           0 WhiteFemale
## 3 Lakisha   female black     0           1 BlackFemale
## 4 Latonya   female black     0           1 BlackFemale
## 5 Carrie    female white     0           0 WhiteFemale
## 6 Jay       male   white     0           0 WhiteMale
```

```
## Did any observations receive the "other" value for type?
filter(resume, type == "other")
```

```
## # A tibble: 0 x 6
## # ... with 6 variables: firstname <chr>, sex <chr>,
## #   race <chr>, call <dbl>, BlackFemale <dbl>, type <chr>
```

```
## check object class
class(resume$type)
```

```
## [1] "character"
```

```
## coerce the character variable into a factor variable
resume <- resume %>%
  mutate(type = as.factor(type))
```

```
## look at the levels of the factor
levels(resume$type)
```

```
## [1] "BlackFemale" "BlackMale"   "WhiteFemale" "WhiteMale"
```

```
firstname_callback <- resume %>%
  group_by(firstname) %>%
  select(firstname, call) %>%
  summarize(callback = mean(call))

head(firstname_callback)
```

```
## # A tibble: 6 x 2
##   firstname callback
##   <chr>        <dbl>
## 1 Aisha       0.0222
## 2 Allison     0.0948
## 3 Anne        0.0826
## 4 Brad        0.159
## 5 Brendan     0.0769
## 6 Brett       0.0678
```

Causal Effects and the Counterfactual

```
slice(resume, 1)
```

```
## # A tibble: 1 x 6
##   firstname sex    race   call BlackFemale type
##   <chr>     <chr>  <chr> <dbl>       <dbl> <fct>
## 1 Allison   female white     0           0 WhiteFemale
```

Randomized Controlled Trials

The Role of Randomization

Social Pressure and Voter Turnout

```
## Load in the data from QSS package
data(resume, package = "qss")

## Or from a csv
social <- read_csv("social.csv",
                   col_types = cols(sex = col_character(),
                                    yearofbirth = col_double(),
                                    primary2004 = col_double(),
                                    messages = col_character(),
                                    primary2006 = col_double(),
                                    hhsize = col_double()))
summary(social) # summarize the data
```

```
##      sex             yearofbirth    primary2004
##  Length:305866      Min.   :1900   Min.   :0.0000
##  Class :character   1st Qu.:1947   1st Qu.:0.0000
##  Mode  :character   Median :1956   Median :0.0000
##                     Mean   :1956   Mean   :0.4014
##                     3rd Qu.:1965   3rd Qu.:1.0000
##                     Max.   :1986   Max.   :1.0000
##    messages          primary2006        hhsize
##  Length:305866      Min.   :0.0000   Min.   :1.000
##  Class :character   1st Qu.:0.0000   1st Qu.:2.000
##  Mode  :character   Median :0.0000   Median :2.000
##                     Mean   :0.3122   Mean   :2.184
##                     3rd Qu.:1.0000   3rd Qu.:2.000
##                     Max.   :1.0000   Max.   :8.000
```

```
## Average turnout by treatment message
turnout_by_message <- social %>%
  group_by(messages) %>%
  summarize(turnout = mean(primary2006))

turnout_by_message
```

```
## # A tibble: 4 x 2
##   messages   turnout
##   <chr>        <dbl>
## 1 Civic Duty   0.315
## 2 Control      0.297
## 3 Hawthorne    0.322
## 4 Neighbors    0.378
```

```
## Differences between treatment(s) and control means
turnout_diffs <- turnout_by_message %>%
  pivot_wider(names_from = messages,
              values_from = turnout) %>%
  mutate(diff_Civic_Duty = `Civic Duty` - Control,
         diff_Hawthorne = Hawthorne - Control,
         diff_Neighbors = Neighbors - Control) %>%
  select(diff_Civic_Duty, diff_Hawthorne, diff_Neighbors)

turnout_diffs
```

```
## # A tibble: 1 x 3
##   diff_Civic_Duty diff_Hawthorne diff_Neighbors
##             <dbl>          <dbl>          <dbl>
## 1          0.0179         0.0257         0.0813
```

```
social %>%
  mutate(age = 2006 - yearofbirth) %>%
  group_by(messages) %>%
  summarize(age_avg = mean(age),
            primary2004_avg = mean(primary2004),
            hhsize_avg = mean(hhsize))
```

```
## # A tibble: 4 x 4
##   messages   age_avg primary2004_avg hhsize_avg
##   <chr>        <dbl>           <dbl>      <dbl>
## 1 Civic Duty    49.7           0.399       2.19
## 2 Control       49.8           0.400       2.18
## 3 Hawthorne     49.7           0.403       2.18
## 4 Neighbors     49.9           0.407       2.19
```

Observational Studies

Minimum Wage and Unemployment

```
minwage <- read_csv("minwage.csv") # load the data
## or
data(minwage, package = "qss")
dim(minwage) # dimension of data
```

```
## [1] 358   8
```

```
glimpse(minwage)
```

```
## Rows: 358
## Columns: 8
## $ chain      <chr> "wendys", "wendys", "burgerking", "burg~
## $ location   <chr> "PA", "PA", "PA", "PA", "PA", "PA", "PA~
## $ wageBefore <dbl> 5.00, 5.50, 5.00, 5.00, 5.25, 5.00, 5.0~
## $ wageAfter  <dbl> 5.25, 4.75, 4.75, 5.00, 5.00, 5.00, 4.7~
## $ fullBefore <dbl> 20.0, 6.0, 50.0, 10.0, 2.0, 2.0, 2.5, 4~
## $ fullAfter  <dbl> 0, 28, 15, 26, 3, 2, 1, 9, 7, 18, 5, 15~
## $ partBefore <dbl> 20.0, 26.0, 35.0, 17.0, 8.0, 10.0, 20.0~
## $ partAfter  <dbl> 36, 3, 18, 9, 12, 9, 25, 32, 39, 10, 20~
```

```
summary(minwage) # summary of data
```

```
##     chain             location           wageBefore   
##  Length:358         Length:358         Min.   :4.250  
##  Class :character   Class :character   1st Qu.:4.250  
##  Mode  :character   Mode  :character   Median :4.500  
##                                        Mean   :4.618  
##                                        3rd Qu.:4.987  
##                                        Max.   :5.750  
##    wageAfter       fullBefore       fullAfter     
##  Min.   :4.250   Min.   : 0.000   Min.   : 0.000  
##  1st Qu.:5.050   1st Qu.: 2.125   1st Qu.: 2.000  
##  Median :5.050   Median : 6.000   Median : 6.000  
##  Mean   :4.994   Mean   : 8.475   Mean   : 8.362  
##  3rd Qu.:5.050   3rd Qu.:12.000   3rd Qu.:12.000  
##  Max.   :6.250   Max.   :60.000   Max.   :40.000  
##    partBefore      partAfter    
##  Min.   : 0.00   Min.   : 0.00  
##  1st Qu.:11.00   1st Qu.:11.00
##  Median :16.25   Median :17.00
##  Mean   :18.75   Mean   :18.69
##  3rd Qu.:25.00   3rd Qu.:25.00
##  Max.   :60.00   Max.   :60.00
```

```
## Add a 'state' variable
minwage <- minwage %>%
  mutate(state = if_else(location == "PA", "PA", "NJ"))

## Create the 'new_wage' object
new_wage <- 5.05

## Calculate the proportions above and below the new wage by state
state_props <- minwage %>%
  mutate(above_min_before = if_else(wageBefore >= new_wage, 1, 0),
         above_min_after = if_else(wageAfter >= new_wage, 1, 0)) %>%
  group_by(state) %>%
  summarize(prop_before = mean(above_min_before),
            prop_after = mean(above_min_after))

state_props
```

```
## # A tibble: 2 x 3
##   state prop_before prop_after
##   <chr>       <dbl>      <dbl>
## 1 NJ         0.0893     0.997
## 2 PA         0.0597     0.0448
```

```
## First create new variables to calculate the
## proportion of full-time employees
minwage <- minwage %>%
  mutate(totalAfter = fullAfter + partAfter,
        fullPropAfter = fullAfter / totalAfter)

## Then calculate the average proportion of full-time workers by state
full_prop_by_state <- minwage %>%
  group_by(state) %>%
  summarize(fullPropAfter = mean(fullPropAfter))

## To calculate the difference between states, we use pivot_wider()
## and mutate()
pivot_wider(full_prop_by_state,
            names_from = state, values_from = fullPropAfter) %>%
  mutate(diff = NJ - PA)
```

```
## # A tibble: 1 x 3
##      NJ    PA   diff
##   <dbl> <dbl>  <dbl>
## 1 0.320 0.272 0.0481
```

Confounding Bias

```
chains_by_state <- minwage %>%
  group_by(state) %>%
  count(chain) %>%
  mutate(prop = n / sum(n)) %>%
  pivot_wider(-n, # this drops the 'n' variable prior to pivoting
              names_from = state,
              values_from = prop)

chains_by_state
```

```
## # A tibble: 4 x 3
##   chain         NJ    PA
##   <chr>      <dbl> <dbl>
## 1 burgerking 0.405 0.463
## 2 kfc        0.223 0.149
## 3 roys       0.251 0.224
## 4 wendys     0.120 0.164
```

```
full_prop_by_state_chain <- minwage %>%
  group_by(state, chain) %>%
  summarize(fullPropAfter = mean(fullPropAfter)) %>%
  pivot_wider(names_from = state,
              values_from = fullPropAfter) %>%
  mutate(diff = NJ - PA)

full_prop_by_state_chain
```

```
## # A tibble: 4 x 4
##   chain         NJ    PA   diff
##   <chr>      <dbl> <dbl>  <dbl>
## 1 burgerking 0.358 0.321 0.0364
## 2 kfc        0.328 0.236 0.0918
## 3 roys       0.283 0.213 0.0697
## 4 wendys     0.260 0.248 0.0117
```

```
prop_by_state_chain_location_subset <- minwage %>%
  filter(!location %in% c("shoreNJ", "centralNJ")) %>%
  group_by(state, chain) %>%
  summarize(fullPropAfter = mean(fullPropAfter)) %>%
  pivot_wider(names_from = state,
              values_from = fullPropAfter) %>%
  mutate(diff = NJ - PA)

prop_by_state_chain_location_subset
```

```
## # A tibble: 4 x 4
##   chain         NJ    PA    diff
##   <chr>      <dbl> <dbl>   <dbl>
## 1 burgerking 0.353 0.321 0.0315
## 2 kfc        0.385 0.236 0.149
## 3 roys       0.289 0.213 0.0761
## 4 wendys     0.253 0.248 0.00510
```

Before-and-After and Difference-in-Differences Designs

```
## First, create a variable for the full-time
## proportion prior to the change
minwage <- minwage %>%
  mutate(totalBefore = fullBefore + partBefore,
         fullPropBefore = fullBefore / totalBefore)

## Then look at the differences in average proportion of full-time
## before and after (in NJ only)
minwage %>%
  filter(state == "NJ") %>%
  summarize(diff = mean(fullPropAfter) - mean(fullPropBefore))
```

```
##         diff
## 1 0.02387474
```

```
## DiD estimate
minwage %>%
  group_by(state) %>%
  ## difference before and after
  summarize(diff = mean(fullPropAfter) - mean(fullPropBefore)) %>%
  pivot_wider(names_from = state, values_from = diff) %>%
  ## difference in difference between states
  mutate(diff_in_diff = NJ - PA)
```

```
## # A tibble: 1 x 3
##       NJ      PA diff_in_diff
##    <dbl>   <dbl>        <dbl>
## 1 0.0239 -0.0377       0.0616
```

Descriptive Statistics for a Single Variable

Quantiles

```
## median difference-in-differences
minwage %>%
  group_by(state) %>%
  summarize(diff = median(fullPropAfter) - median(fullPropBefore)) %>%
  pivot_wider(names_from = state, values_from = diff) %>%
  mutate(diff_in_diff = NJ - PA)
```

```
## # A tibble: 1 x 3
##       NJ      PA diff_in_diff
##    <dbl>   <dbl>        <dbl>
## 1 0.0250 -0.0120       0.0370
## summary() shows quartiles for the two wages variables
## as well as minimum, maximum, and mean
minwage %>%
  filter(state == "NJ") %>% # just look at NJ
  select(wageBefore, wageAfter) %>%
  summary()
```

```
##    wageBefore     wageAfter
##  Min.   :4.25   Min.   :5.000
##  1st Qu.:4.25   1st Qu.:5.050
##  Median :4.50   Median :5.050
##  Mean   :4.61   Mean   :5.081
##  3rd Qu.:4.87   3rd Qu.:5.050
##  Max.   :5.75   Max.   :5.750
```

```
## The interquartile range
minwage %>%
  filter(state == "NJ") %>%
  select(wageBefore, wageAfter) %>%
  summarize(wageBeforeIQR = IQR(wageBefore),
            wageAfterIQR = IQR(wageAfter))
```

```
##   wageBeforeIQR wageAfterIQR
## 1          0.62            0
```

```
## Create an object for the quantiles we want (deciles)
decile_probs <- seq(from = 0, to = 1, by = 0.1)
## Save deciles as characters
decile_names <- as.character(decile_probs)

## Generate the deciles for wage before and after
minwage %>%
  filter(state == "NJ") %>%
  select(wageBefore, wageAfter) %>%
  summarize(wageBeforeDecile = quantile(wageBefore, probs = decile_probs),
            wageAfterDecile = quantile(wageAfter, probs = decile_probs),
            decile = decile_names)
```

```
##    wageBeforeDecile wageAfterDecile decile
## 1              4.25            5.00      0
## 2              4.25            5.05    0.1
## 3              4.25            5.05    0.2
## 4              4.25            5.05    0.3
## 5              4.50            5.05    0.4
## 6              4.50            5.05    0.5
## 7              4.65            5.05    0.6
## 8              4.75            5.05    0.7
## 9              5.00            5.05    0.8
## 10             5.00            5.15    0.9
## 11             5.75            5.75      1
```

Standard Deviation

```r
## Calculate the RMS of the change in full-time
## employment proportion in NJ
minwageNJ %>%
  mutate(fullPropChange = fullPropAfter - fullPropBefore,
         sqfullPropChange = fullPropChange^2) %>%
  summarize(rms = sqrt(mean(sqfullPropChange)))
```

```
##         rms
## 1 0.3014669
```

```r
## Compare to the mean
minwageNJ %>%
  mutate(fullPropChange = fullPropAfter - fullPropBefore) %>%
  summarize(mean = mean(fullPropChange))
```

```
##         mean
## 1 0.02387474
```

```r
minwage %>%
  group_by(state) %>%
  summarize_at(vars(fullPropBefore, fullPropAfter),
               .funs = list(sd, var))
```

```
## # A tibble: 2 x 5
##   state fullPropBefore_f~ fullPropAfter_fn1 fullPropBefore_~
##   <chr>             <dbl>             <dbl>            <dbl>
## 1 NJ                0.230             0.251           0.0531
## 2 PA                0.240             0.247           0.0575
## # ... with 1 more variable: fullPropAfter_fn2 <dbl>
```

```r
minwage %>%
  group_by(state) %>%
  summarize_at(vars(fullPropBefore, fullPropAfter),
               .funs = list(stdv = sd,
                            variance = var))
```

```
## # A tibble: 2 x 5
##   state fullPropBefore_s~ fullPropAfter_s~ fullPropBefore_v~
##   <chr>             <dbl>            <dbl>             <dbl>
## 1 NJ                0.230            0.251            0.0531
## 2 PA                0.240            0.247            0.0575
## # ... with 1 more variable: fullPropAfter_variance <dbl>
```