HW4_ahcooper

Andrew Cooper

10/11/2020

```
library(tidyverse)
library(knitr)
library(microbenchmark)
library(parallel)
```

Problem 2

```
grad_descent <- function(X, h, theta0, theta1, alpha = 1e-7, tol = 1e-9){
  m <- nrow(X)
  theta0_old <- theta0
  theta1_old <- theta1
  diff0 <- diff1 <- 1
  num_iter <- 0
  while(diff0 > tol && diff1 > tol && num_iter < 5e6){
    h0 <- X %*% c(theta0_old, theta1_old)
    theta0_new <- theta0_old - alpha*(1/m)*sum(h0 - h)
    theta1_new <- theta1_old - alpha*(1/m)*sum((h0 - h)*X[, 2])
    diff0 <- abs(theta0_new - theta0_old)
    diff1 <- abs(theta1_new - theta1_old)
    theta0_old <- theta0_new
    theta1_old <- theta1_new
    num_iter <- num_iter + 1
  }
  return(list(
    "theta" = c(theta0_new, theta1_new),
    "num_iter" = num_iter))
}
```

```
set.seed(1256)
theta <- as.matrix(c(1, 2), nrow = 2)
X <- cbind(1, rep(1:10, 10))
h <- X %*% theta + rnorm(100, 0, 0.2)
grad_descent(X, h, 1, 4, 0.01)
```

```
## $theta
## [1] 0.9695673 2.0015635
##
## $num_iter
## [1] 5347
```

```r
lm(h ~ 0 + X)
```

```
##
## Call:
## lm(formula = h ~ 0 + X)
##
## Coefficients:
##     X1      X2
## 0.9696  2.0016
```

The results are about the same.

Problem 3

Part a

```r
n <- 15
starting_vals <- seq(1 - n, 1 + n)
grid_inds <- expand.grid(seq(1, n, 1), seq(1, n, 1))
```

```r
cl <- makeCluster(10)
clusterExport(cl, "grad_descent")
clusterExport(cl, "X")
clusterExport(cl, "h")
clusterExport(cl, "starting_vals")
```

```r
starting_value_analysis <- parApply(cl, grid_inds, 1, function(row){
  i <- row[1]
  j <- row[2]
  return(est <- grad_descent(X, h, starting_vals[i], starting_vals[j], 0.01))
})
```

```r
beta_estimates <- sapply(1:length(starting_value_analysis), function(i){starting_value_analysis[[i]][[1]
iterations <-  sapply(1:length(starting_value_analysis), function(i){starting_value_analysis[[i]][[2]]}
```

The minimum number of iterations was 4415 when our starting value for θ_0 was 15 and our starting values for θ_1 was 10.

The maximum number of iterations was 7272 when our starting value for θ_0 was 1 and our starting values for θ_1 was 15.

The mean beta estimate was 0.9695677, 2.0015634. The standard deviation was 1.5456037×10^{-6}, 2.2201136×10^{-7}.

Part b

No this would not be a good idea. The purpose of these algorithms is to search for and approximate unknown values. In this case we can actually calculate the exact solution, but in most others cases where the function is not convex we can't find an exact formula for the solution and need to use an algorithm like this to find it.

Part c

It is nice that it is such a simple approach, however this can be really slow and take too many iterations, especially toward the end. A more advanced/complex algorithm would likely find approximations to the minimum of the objective function with fewer iterations.

Problem 4

This is the exact solution for the values of $[\theta_0, \theta_2]$ that minimize the sum of squared-errors, $(Y - X\hat{\beta})^T(Y - X\hat{\beta})$. We can derive an equation for the minimum of this objective function since it is quadratic and therefore convex. We derive this solution by taking the derivative of the objective function with respect to $\hat{\beta}$, then equating it to 0 and solving.

To calculate this matrix we need to invert the square of the data matrix X, which we know is invertible due to its symmetry. We can compute the exact inverse using a program like R, although it is an $O(n^3)$ operation which is relatively slow. We could instead calculate a pseudo-inverse, which adds some error to the calculation but is usually significantly quicker.

Problem 5

```
set.seed(12456)

G <- matrix(sample(c(0, 0.5, 1), size = 16000, replace = T), ncol = 10)
R <- cor(G)
C <- kronecker(R, diag(1600))
id <- sample(1:1600, size = 932, replace = F)
q <- sample(c(0, 0.5, 1), size = 15068, replace = T)
A <- C[id, -id]
B <- C[-id, -id]
p <- runif(932, 0, 1)
r <- runif(15068, 0, 1)
C <- NULL
```

Part a

A is 1.1234722×10^8. B is 1.8163572×10^9.

```
microbenchmark({
  p + A %*% solve(B) %*% t(t(q - r))
}, times = 1)
```

Part b

We would likely want to break apart the inverse operation from the rest of the

Part c

Problem 3 (2)

Part a

```
success_rate <- function(v, s){
  return(length(which(v == s)) / length(v))
}
```

Part b

```
set.seed(12345)
P4b_data <- matrix(rbinom(10, 1, prob = (31:40)/100), nrow = 10, ncol = 10, byrow = F)
```

Part c

```
apply(P4b_data, 2, function(x){success_rate(x, 1)})
```

```
##  [1] 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6
```

The issue with this is the creation of the matrix "P4b_data". Instead of simulating the results of ten tosses for each of the different probabilities, we only simulated one series of 10 flips and made each column that same series.

Part d

```
calc_prob <- function(p){
  return(rbinom(10, 1, p))
}
```

```
P4b_data <- sapply((31:40)/100, calc_prob)
```

```
apply(P4b_data, 2, function(x){success_rate(x, 1)})
```

```
##  [1] 0.2 0.3 0.4 0.3 0.4 0.6 0.3 0.3 0.5 0.6
```

These are the correct estimates of the marginal success rates.

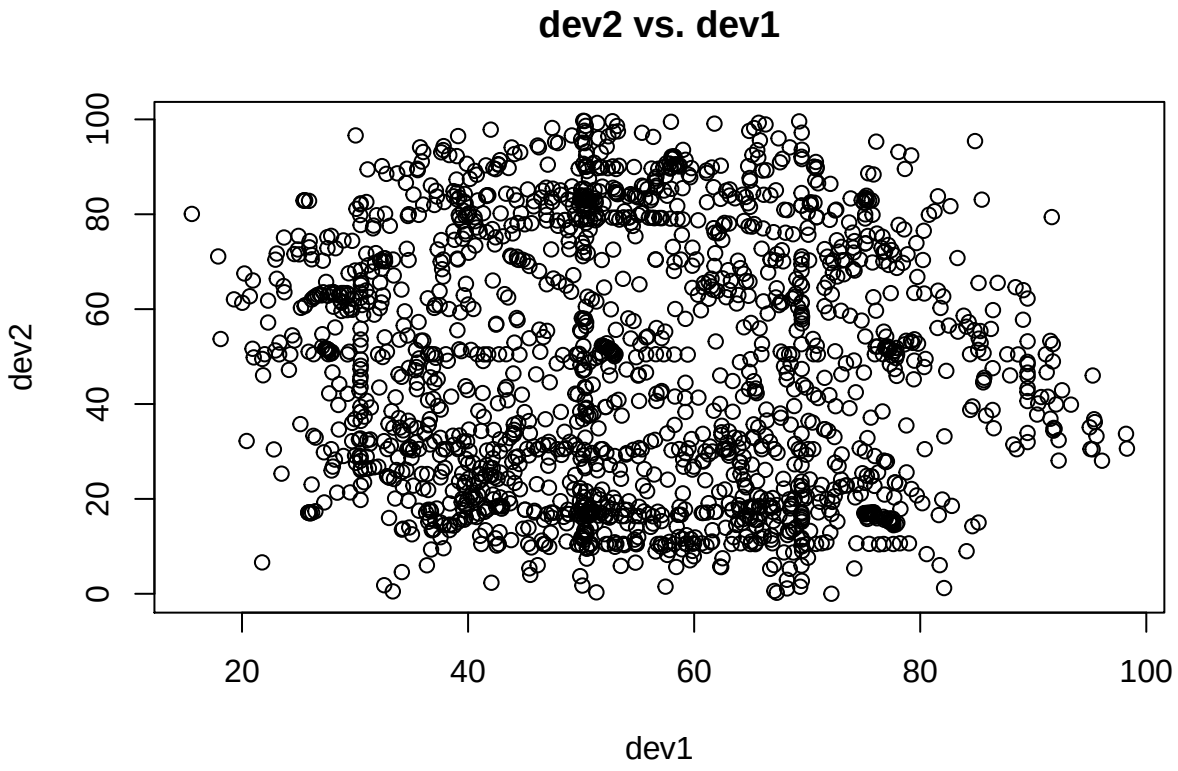
Problem 4 (2)

```r
df <- readRDS("HW3_data.rds")
```

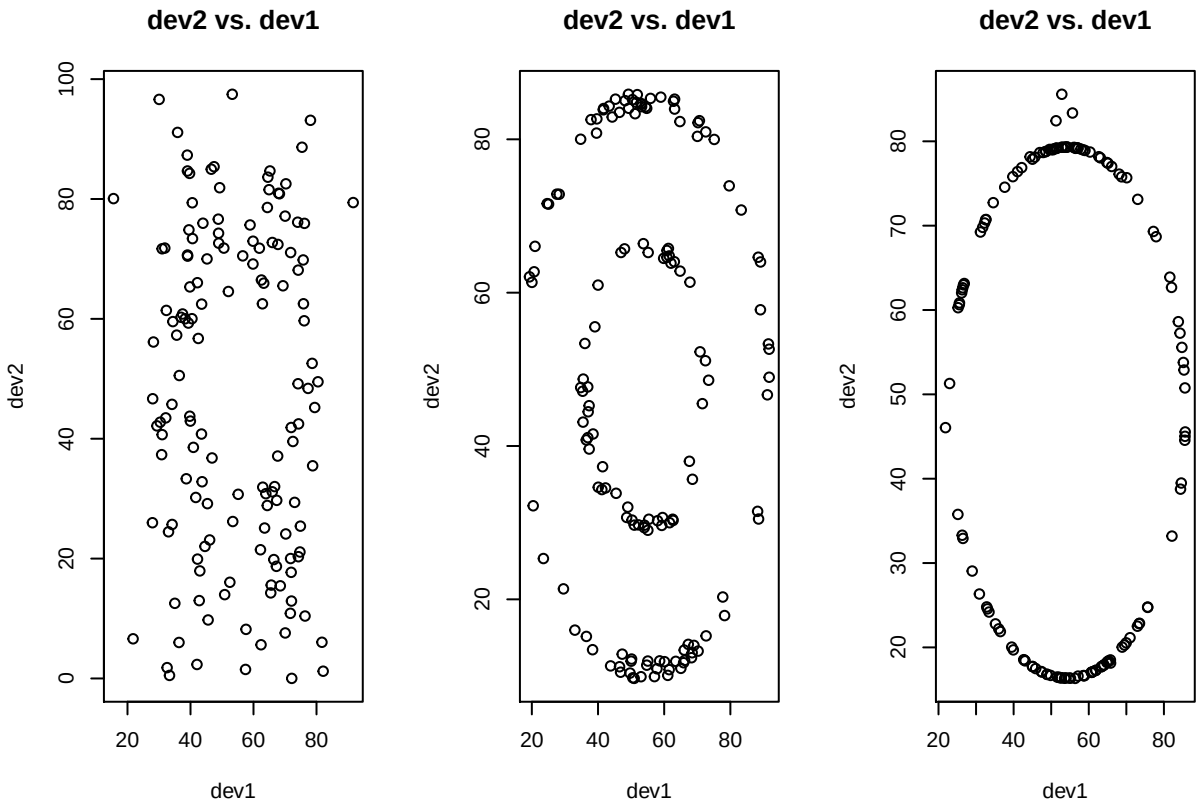
```r
df <- df %>%
  rename(x = dev1,
         y = dev2)
```

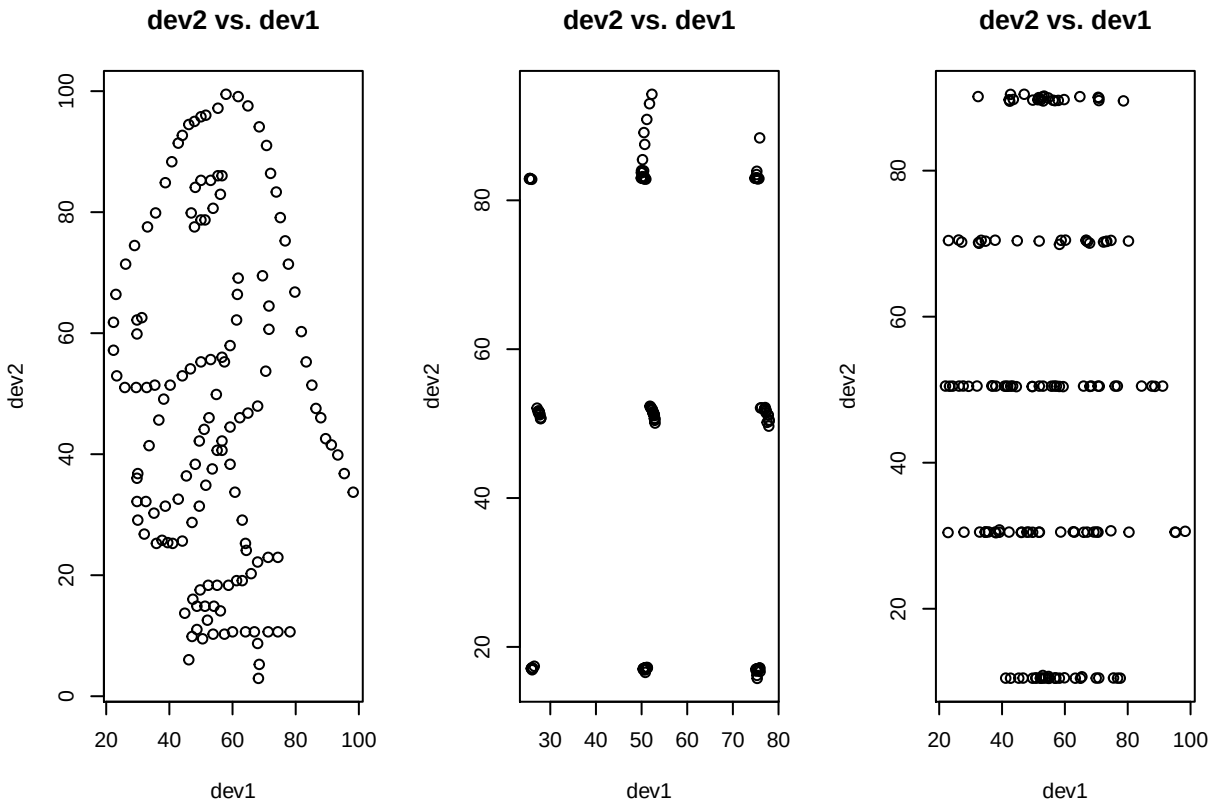
```r
create_scatter <- function(df, title, label){
  df %>%
    plot(main = title, xlab = label[1], ylab = label[2])
}
```

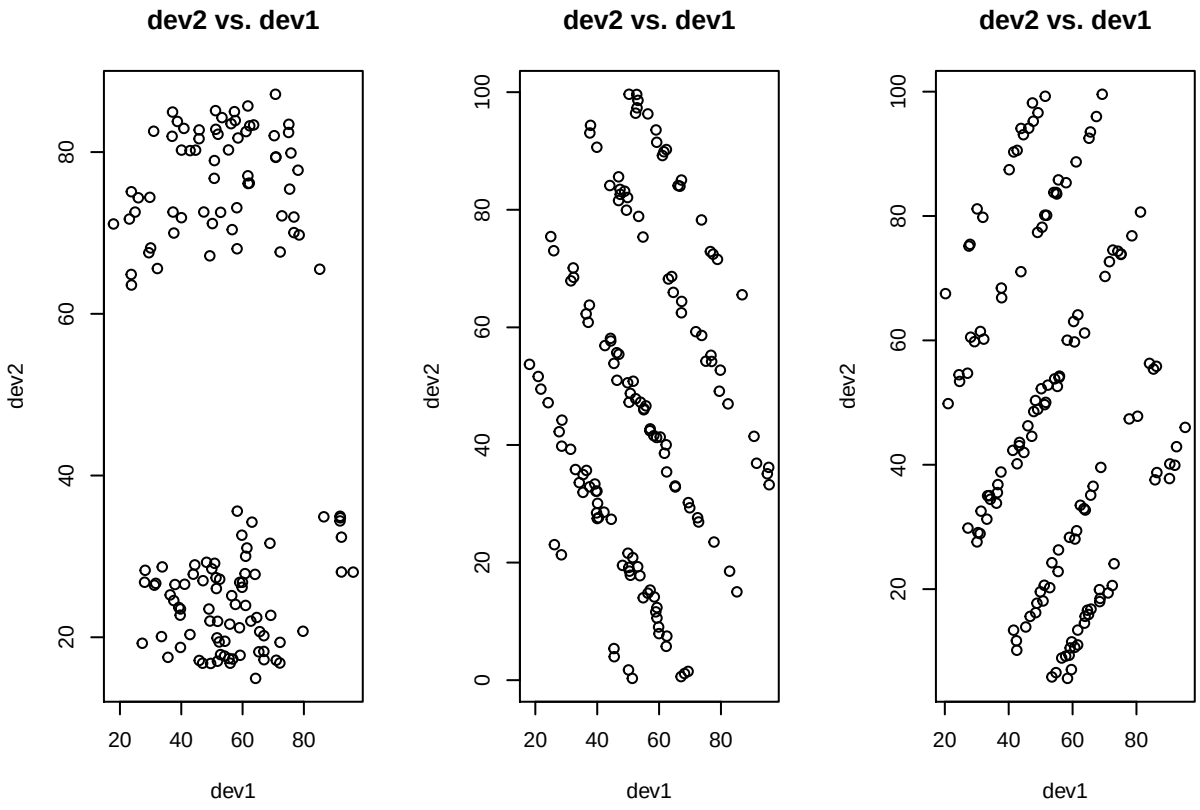
```r
create_scatter(df[c(2,3)], "dev2 vs. dev1", c("dev1", "dev2"))
```

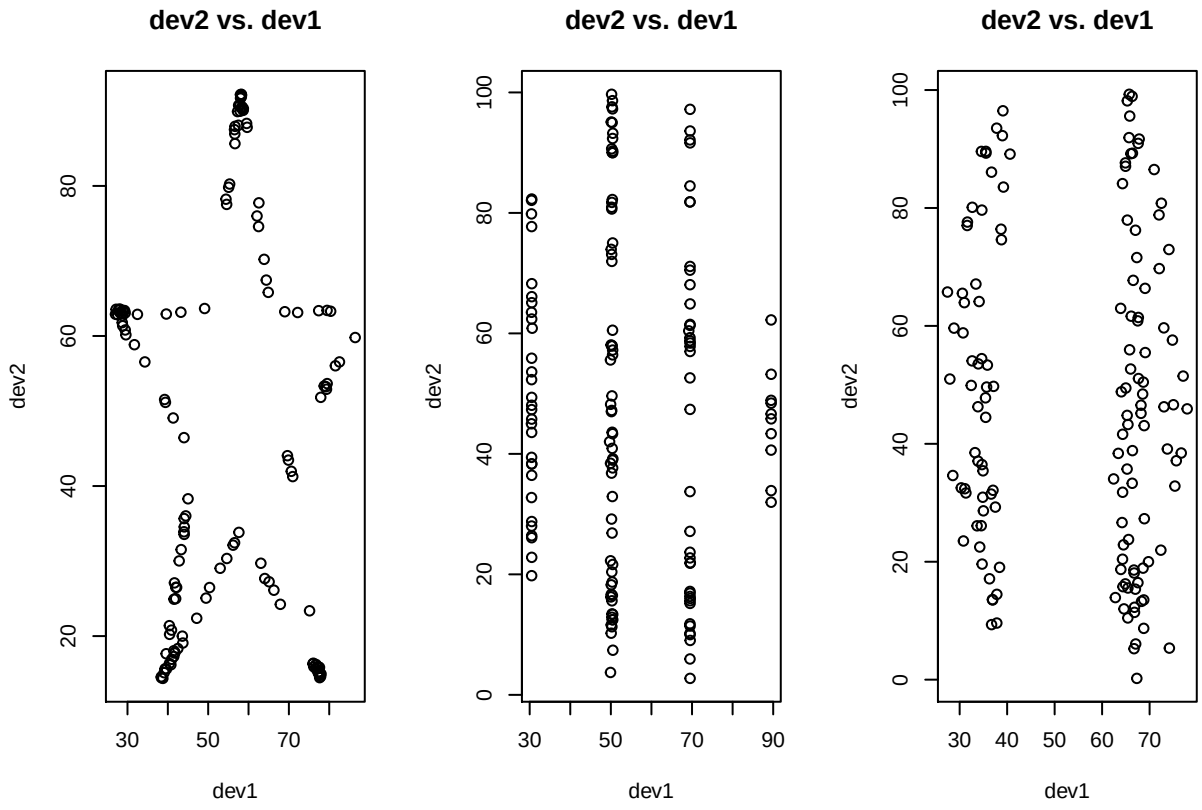



```r
par(mfrow = c(1, 3))
sapply(seq(1, 13, 1), function(i){
  df_filtered <- df %>% filter(Observer == i)
  create_scatter(df_filtered[c(2, 3)], "dev2 vs. dev1", c("dev1", "dev2"))
})
```


dev2 vs. dev1
dev2
dev1
0
20
40
60
80
100
20
40
60
80
dev2 vs. dev1
dev2
dev1
20
40
60
80
20
40
60
80
dev2 vs. dev1
dev2
dev1
20
30
40
50
60
70
80
20
40
60
80

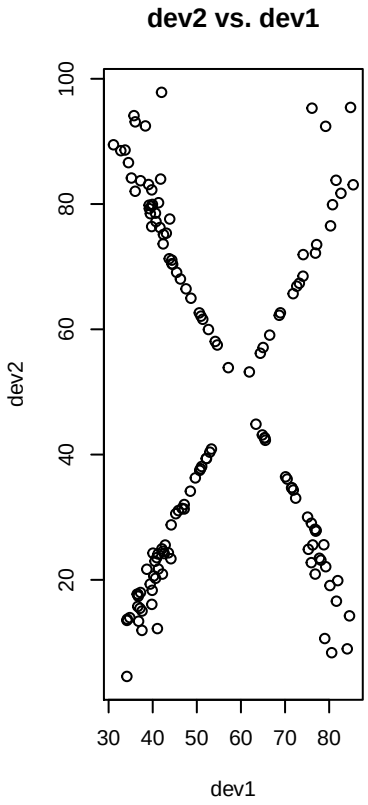

dev2 vs. dev1
dev2
dev1
0
20
40
60
80
100
dev2 vs. dev1
dev2
dev1
20
40
60
80
30
40
50
60
70
80
dev2 vs. dev1
dev2
dev1
20
40
60
80
100


dev2 vs. dev1
dev2
dev1
20
40
60
80
dev2 vs. dev1
dev2
dev1
0
20
40
60
80
100
dev2 vs. dev1
dev2
dev1
20
40
60
80
100




```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
```

```
##
## [[10]]
## NULL
##
## [[11]]
## NULL
##
## [[12]]
## NULL
##
## [[13]]
## NULL
```



Problem 5 (2)

Part a

```
library(downloader)
download("http://www.farinspace.com/wp-content/uploads/us_cities_and_states.zip",dest="us_cities_states
unzip("us_cities_states.zip")

#read in data, looks like sql dump, blah
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
states <- fread(input = "./us_cities_and_states/states.sql",skip = 23,sep = "'", sep2 = ",", header = F
### YOU do the CITIES
### I suggest the cities_extended.sql may have everything you need
### can you figure out how to limit this to the 50?
cities <- fread(input = "./us_cities_and_states/cities_extended.sql", sep = "'", sep2 = ",", header = F
```

Part b

```
cities %>%
  rename(State = V4) %>%
  unique() %>%
  group_by(State) %>%
  summarise("Number of Cities" = n()) %>%
  kable()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

State	Number of Cities
AK	229
AL	579
AR	605
AZ	264
CA	1239
CO	400
CT	269
DC	3
DE	57
FL	524
GA	629
HI	92
IA	937
ID	266
IL	1287
IN	738
KS	634
KY	803
LA	479
MA	511

State	Number of Cities
MD	430
ME	461
MI	885
MN	810
MO	942
MS	440
MT	360
NC	762
ND	373
NE	528
NH	255
NJ	579
NM	346
NV	99
NY	1612
OH	1069
OK	585
OR	379
PA	1802
PR	99
RI	70
SC	377
SD	364
TN	548
TX	1466
UT	250
VA	839
VT	288
WA	493
WI	753
WV	753
WY	176

Part c

```
count_frequency <- function(letter, state_name){
  v <- strsplit(state_name, "") %>% unlist()
  return(sum(tolower(v) == tolower(letter)))
}
```

```
letter_count <- data.frame(matrix(NA, nrow = 50, ncol = 26))
getCount <- function(state){
  return(sapply(letters, function(x){count_frequency(x, state)}))
}
```

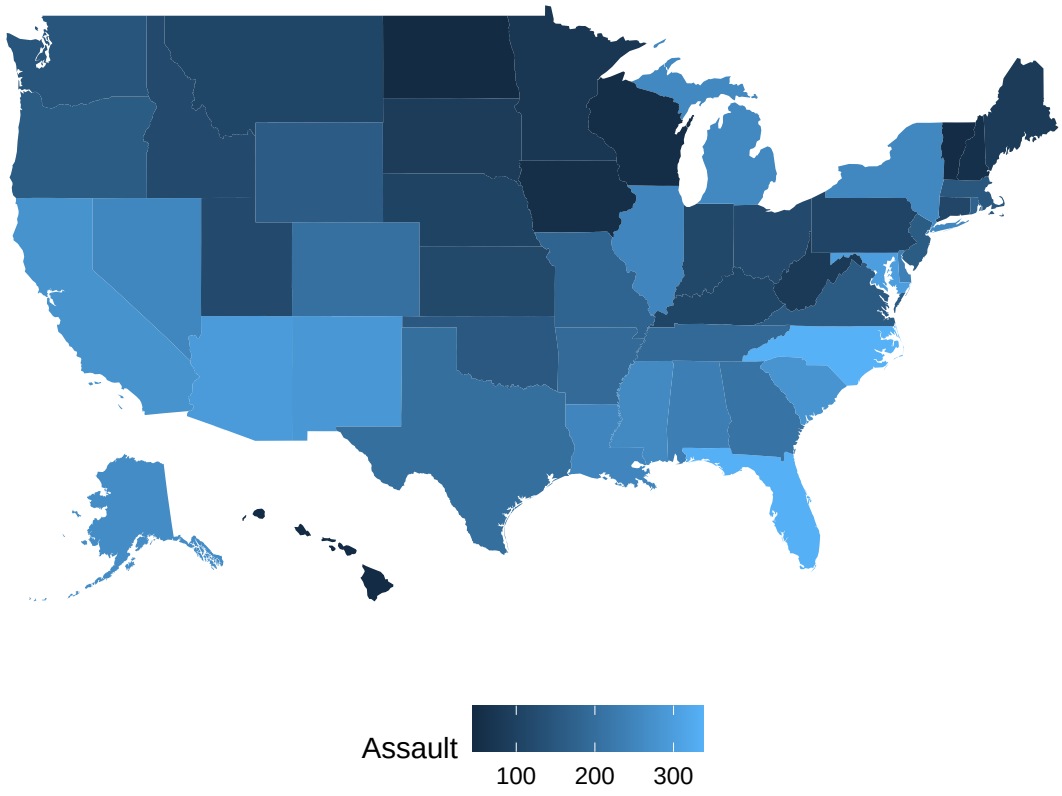
```
letter_count <- sapply(states$V2, function(x){getCount(x)}) %>% t()
```

Part d

```
library(fiftystater)

data("fifty_states") # this line is optional due to lazy data loading
crimes <- data.frame(state = tolower(rownames(USArrests)), USArrests)
# map_id creates the aesthetic mapping to the state name column in your data
p <- ggplot(crimes, aes(map_id = state)) +
  # map points to the fifty_states shape data
  geom_map(aes(fill = Assault), map = fifty_states) +
  expand_limits(x = fifty_states$long, y = fifty_states$lat) +
  coord_map() +
  scale_x_continuous(breaks = NULL) +
  scale_y_continuous(breaks = NULL) +
  labs(x = "", y = "") +
  theme(legend.position = "bottom",
        panel.background = element_blank())

p
```




```
#ggsave(plot = p, file = "HW6_Problem6_Plot_Settlage.pdf")
```

Problem 2 (3)

Part a

The issue with the code is in the "lm" statement in the for-loop. The formula uses existing data frames, not the names of variables in df08. The lm function just ignores the data argument in this case, which is why the result is just the same as the original regression. To fix this you need to change the variable names to those in df08.

Part b

```
Sensory_raw <- read_table2("https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat",
     col_types = cols(Item = col_character()), col_names = F, skip = 2)
```

```
# Remove NA's, align columns
Sensory_values <- apply(Sensory_raw, 1, function(x){
  if(any(is.na(x))){
    return(x[!is.na(x)])
  }else{
    return(x[-1])
  }
})
# Add column for item
Sensory <- data.frame(cbind(rep(seq(1, 10, 1), each = 3), t(Sensory_values)))
colnames(Sensory) <- c("Item", seq(1, 5, 1))
Sensory <- as_data_frame(Sensory)
```

```
Sensory <- Sensory %>%
  gather("Operator", "Y", -Item)
```

```
bootstrap_size <- 30
bootstrap_sd <- matrix(nrow = 100, ncol = 5)
for(i in 1:100){
  bootstrap_sample <- purrr::map(1:5, function(x){
    df_filtered <- Sensory[which(Sensory$Operator == x), ]
    bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T)
    return(df_filtered[bootstrap_inds, ])
  }) %>% combine()
  bootstrap_sd[i, ] <- lm(Y ~ Operator, bootstrap_sample) %>% coef()
}
```

```
## Warning: `combine()` is deprecated as of dplyr 1.0.0.
## Please use `vctrs::vec_c()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
apply(bootstrap_sd, 2, sd)
```

```
## [1] 0.4190709 0.5529760 0.6166114 0.5798449 0.5565354
```

```
bootstrap_nonparallel_results <- microbenchmark({
  bootstrap_size <- 30
  bootstrap_sd <- matrix(nrow = 100, ncol = 5)
  for(i in 1:100){
    bootstrap_sample <- purrr::map(1:5, function(x){
      df_filtered <- Sensory[which(Sensory$Operator == x), ]
      bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T)
      return(df_filtered[bootstrap_inds, ])
    }) %>% combine()
    bootstrap_sd[i, ] <- lm(Y ~ Operator, bootstrap_sample) %>% coef()
  }
})
```

Part c

We can run the bootstraps in parallel since the bootstraps don't depend on one another, i.e. the results of one bootstrap don't affect the results of another.

```
kable(bootstrap_nonparallel_results)
```

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 686858300\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	903518200
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 870343100\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	789639700
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 875242700\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	950297200

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 743107900\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	722397300
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 897392200\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	761560800
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } \| 702874300\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	672598000
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } \| 993165100\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	928355000
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } \| 911369700\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	751940700

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 694621200\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	815750000
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 770288300\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	759109800
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 673348700\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	843074000
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 755721400\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	809991600
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 733920500\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	684535700

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 630502300| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	743190500
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 788976800| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	740318400
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 701879300| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	744284600
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 704446900| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	693468300
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 763852000| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	685013000

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 664795000| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	769718400
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 675124300| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	614872300
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 636837600| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	570251200
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 558007700| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	594809600
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 559007000| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	583983700

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 580768500\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	604143100
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 580687500\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	578268000
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 633764800\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	606769500
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 596027600\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	576703100
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 616280400\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	593573100

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 577333400| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	685444100
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 565149600| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	651018800
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 582232800| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	650744800
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 758929900| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	646630300
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 548330600| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	621393500

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 621814300\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	584638300
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 689447400\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	549744400
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 639788800\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	590024600
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 545363600\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	669587100
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 612097500\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	631233400

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 664563600| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	558794100
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 582400500| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	635575300
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 594568300| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	584839400
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 546521500| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	661918500
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 576664200| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	581240600

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 674577800| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	551444500
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 626892700| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	653474800
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 571545900| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	579278400
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 602221700| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	669223100
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } | 617017300| |{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	582549900

expr	time
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 979819700\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	614797700
{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } } \| 574588800\| \|{ bootstrap_size <- 30 bootstrap_sd <- matrix(nrow = 100, ncol = 5) for (i in 1:100) { bootstrap_sample <- purrr::map(1:5, function(x) { df_filtered <- Sensory[which(Sensory$Operator == x),] bootstrap_inds <- sample(1:nrow(df_filtered), bootstrap_size, replace = T) return(df_filtered[bootstrap_inds,]) }) %>% combine() bootstrap_sd[i,] <- lm(Y ~ Operator, bootstrap_sample) %>% coef() } }	629636400

Problem 3 (3)

```
Newton <- function(f, fp, x0, tol = 1e-6){
  # Returns a root of the function f with derivative fp at starting point x0 under error tolerance tol
  x <- x0
  x_vals <- c(x)
  f_vals <- c(f(x))
  fp_vals <- c(fp(x))
  diff <- abs(f(x))
  diff_vals <- c(diff)
  # Iterate until y-value at x is below tolerance
  while(diff > tol){
    x <- x - f(x)/fp(x)
    diff <- abs(f(x))
    diff_vals <- c(diff_vals, diff)
    x_vals <- c(x_vals, x)
    f_vals <- c(f_vals, f(x))
    fp_vals <- c(fp_vals, fp(x))
  }
  return(list(
    "root" = x,
    "iterations" = diff_vals,
    "x_vals"     = x_vals,
    "f_vals"     = f_vals,
    "fp_vals"    = fp_vals
  ))
}
```

Part a

```
f <- function(x){3^x - sin(x) + cos(5*x)}
fp <- function(x){3^x*log(3) - cos(x) - 5*sin(5*x)}
```

```
x_grid <- seq(-10, 10, 1)
```

```
sapply(x_grid, function(x){Newton(f, fp, x)$root})
```

```
##  [1]  -9.162986  -9.162986  -8.115867  -7.068483  -6.021155  -4.971508
##  [7]  -3.930114  -2.887058  -5.107437  -2.887058 -20.682151  -8.115867
## [13]  -3.930114  -3.528723  -2.887058  -2.887058 -11.257371  -2.887058
## [19]  -2.887058 -29.059732 -15.446164
```

Part b

```
cl <- makeCluster(8)
clusterExport(cl, "Newton")
clusterExport(cl, "f")
clusterExport(cl, "fp")
parSapply(cl, x_grid, function(x){Newton(f, fp, x)$root})
```

```
##  [1]  -9.162986  -9.162986  -8.115867  -7.068483  -6.021155  -4.971508
##  [7]  -3.930114  -2.887058  -5.107437  -2.887058 -20.682151  -8.115867
## [13]  -3.930114  -3.528723  -2.887058  -2.887058 -11.257371  -2.887058
## [19]  -2.887058 -29.059732 -15.446164
```