

HW(1)

Tuseng Vang

2/6/2022

#Run this code first

If you don't know the answer, leave it blank. If you are caught cheating, you will be given minus 50 points.

Q1. Replace the author name with your name in YAML part above.

Q2. Store five values 82.0, 31.2, 98.2, 19.4, 72.6 into the `scores` variable.

```
scores <- c(82.0, 31.2, 98.2, 19.4, 72.6)
```

Q3. Write a code that finds the minimum value of `scores` that you have created in Q2.

```
min(scores)
```

```
## [1] 19.4
```

Q4. Assign the value of 4 raised to 2 to a variable `generation`. Then, print out the value of `generation`.

```
generation <- 4^2  
print(generation)
```

```
## [1] 16
```

Q5. Assign the value of square root 81 to a variable `nine`, and print out `nine`.

```
nine <- sqrt(81)
```

Q6. Store a text `mozart` into the variable `piano`.

```
piano <- "mozart"
```

Q7. What are three components for a single plot of `ggplot2` package?

```
data
```

```
## function (... , list = character(), package = NULL, lib.loc = NULL,  
##     verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)  
## {  
##     fileExt <- function(x) {  
##         db <- grepl("\\.[^.]+\\.(gz|bz2|xz)$", x)  
##         ans <- sub(".*\\.\\.", "", x)  
##         ans[db] <- sub(".*\\.([^.]+\\.)(gz|bz2|xz)$", "\\1\\2",  
##             x[db])  
##         ans  
##     }  
## }
```

```

## my_read_table <- function(...) {
##   lcc <- Sys.getlocale("LC_COLLATE")
##   on.exit(Sys.setlocale("LC_COLLATE", lcc))
##   Sys.setlocale("LC_COLLATE", "C")
##   read.table(...)
## }
## names <- c(as.character(substitute(list(...))[-1L]), list)
## if (!is.null(package)) {
##   if (!is.character(package))
##     stop("'package' must be a character string or NULL")
##   if (FALSE) {
##     if (any(package %in% "base"))
##       warning("datasets have been moved from package 'base' to package 'datasets'")
##     if (any(package %in% "stats"))
##       warning("datasets have been moved from package 'stats' to package 'datasets'")
##     package[package %in% c("base", "stats")] <- "datasets"
##   }
## }
## paths <- find.package(package, lib.loc, verbose = verbose)
## if (is.null(lib.loc))
##   paths <- c(path.package(package, TRUE), if (!length(package)) getwd(),
##             paths)
## paths <- unique(normalizePath(paths[file.exists(paths)]))
## paths <- paths[dir.exists(file.path(paths, "data"))]
## dataExts <- tools:::.make_file_exts("data")
## if (length(names) == 0L) {
##   db <- matrix(character(), nrow = 0L, ncol = 4L)
##   for (path in paths) {
##     entries <- NULL
##     packageName <- if (file_test("-f", file.path(path,
##             "DESCRIPTION")))
##       basename(path)
##     else "."
##     if (file_test("-f", INDEX <- file.path(path, "Meta",
##             "data.rds"))) {
##       entries <- readRDS(INDEX)
##     }
##     else {
##       dataDir <- file.path(path, "data")
##       entries <- tools::list_files_with_type(dataDir,
##             "data")
##       if (length(entries)) {
##         entries <- unique(tools::file_path_sans_ext(basename(entries)))
##         entries <- cbind(entries, "")
##       }
##     }
##   }
##   if (NROW(entries)) {
##     if (is.matrix(entries) && ncol(entries) == 2L)
##       db <- rbind(db, cbind(packageName, dirname(path),
##             entries))
##     else warning(gettextf("data index for package %s is invalid and will be ignored",
##             sQuote(packageName)), domain = NA, call. = FALSE)
##   }
## }
## }

```

```

##      colnames(db) <- c("Package", "LibPath", "Item", "Title")
##      footer <- if (missing(package))
##        paste0("Use ", sQuote(paste("data(package = ", ".packages(all.available = TRUE)))"),
##              "\n", "to list the data sets in all *available* packages.")
##      else NULL
##      y <- list(title = "Data sets", header = NULL, results = db,
##              footer = footer)
##      class(y) <- "packageIQR"
##      return(y)
##    }
##    paths <- file.path(paths, "data")
##    for (name in names) {
##      found <- FALSE
##      for (p in paths) {
##        tmp_env <- if (overwrite)
##          envir
##        else new.env()
##        if (file_test("-f", file.path(p, "Rdata.rds"))) {
##          rds <- readRDS(file.path(p, "Rdata.rds"))
##          if (name %in% names(rds)) {
##            found <- TRUE
##            if (verbose)
##              message(sprintf("name=%s:\t found in Rdata.rds",
##                              name), domain = NA)
##            thispkg <- sub(".*(?:[/]*)/data$", "\\1", p)
##            thispkg <- sub("_.*$", "", thispkg)
##            thispkg <- paste0("package:", thispkg)
##            objs <- rds[[name]]
##            lazyLoad(file.path(p, "Rdata"), envir = tmp_env,
##                    filter = function(x) x %in% objs)
##            break
##          }
##        else if (verbose)
##          message(sprintf("name=%s:\t NOT found in names() of Rdata.rds, i.e.,\n\t%s\n",
##                          name, paste(names(rds), collapse = ",")),
##                  domain = NA)
##      }
##      if (file_test("-f", file.path(p, "Rdata.zip"))) {
##        warning("zipped data found for package ", sQuote(basename(dirname(p))),
##              "\nThat is defunct, so please re-install the package.",
##              domain = NA)
##        if (file_test("-f", fp <- file.path(p, "filelist")))
##          files <- file.path(p, scan(fp, what = "", quiet = TRUE))
##        else {
##          warning(gettextf("file 'filelist' is missing for directory %s",
##                          sQuote(p)), domain = NA)
##          next
##        }
##      }
##      else {
##        files <- list.files(p, full.names = TRUE)
##      }
##      files <- files[grepl(name, files, fixed = TRUE)]
##      if (length(files) > 1L) {

```

```

##           o <- match(fileExt(files), dataExts, nomatch = 100L)
##           paths0 <- dirname(files)
##           paths0 <- factor(paths0, levels = unique(paths0))
##           files <- files[order(paths0, o)]
##       }
##   if (length(files)) {
##       for (file in files) {
##           if (verbose)
##               message("name=", name, ":\t file= ...", .Platform$file.sep,
##                       basename(file), ":\t", appendLF = FALSE,
##                       domain = NA)
##           ext <- fileExt(file)
##           if (basename(file) != paste0(name, ".", ext))
##               found <- FALSE
##           else {
##               found <- TRUE
##               zfile <- file
##               zipname <- file.path(dirname(file), "Rdata.zip")
##               if (file.exists(zipname)) {
##                   Rdatadir <- tempfile("Rdata")
##                   dir.create(Rdatadir, showWarnings = FALSE)
##                   topic <- basename(file)
##                   rc <- .External(C_unzip, zipname, topic,
##                                   Rdatadir, FALSE, TRUE, FALSE, FALSE)
##                   if (rc == 0L)
##                       zfile <- file.path(Rdatadir, topic)
##               }
##               if (zfile != file)
##                   on.exit(unlink(zfile))
##               switch(ext, R = , r = {
##                   library("utils")
##                   sys.source(zfile, chdir = TRUE, envir = tmp_env)
##               }, RData = , rdata = , rda = load(zfile,
##                   envir = tmp_env), TXT = , txt = , tab = ,
##                   tab.gz = , tab.bz2 = , tab.xz = , txt.gz = ,
##                   txt.bz2 = , txt.xz = assign(name, my_read_table(zfile,
##                       header = TRUE, as.is = FALSE), envir = tmp_env),
##                   CSV = , csv = , csv.gz = , csv.bz2 = ,
##                   csv.xz = assign(name, my_read_table(zfile,
##                       header = TRUE, sep = ";", as.is = FALSE),
##                       envir = tmp_env), found <- FALSE)
##               }
##               if (found)
##                   break
##           }
##           if (verbose)
##               message(if (!found)
##                       "*NOT* ", "found", domain = NA)
##       }
##   if (found)
##       break
## }
## if (!found) {
##     warning(gettextf("data set %s not found", sQuote(name)),

```

```
##             domain = NA)
##         }
##         else if (!overwrite) {
##             for (o in ls(envir = tmp_env, all.names = TRUE)) {
##                 if (exists(o, envir = envir, inherits = FALSE))
##                     warning(gettextf("an object named %s already exists and will not be overwritten",
##                                     sQuote(o)))
##                 else assign(o, get(o, envir = tmp_env, inherits = FALSE),
##                             envir = envir)
##             }
##             rm(tmp_env)
##         }
##     }
##     invisible(names)
## }
## <bytecode: 0x000000001d29d008>
## <environment: namespace:utils>
geom
```

```
## Error in eval(expr, envir, enclos): object 'geom' not found
```

```
mapping
```

```
## Error in eval(expr, envir, enclos): object 'mapping' not found
```

Q8. A line of code that shows presidential data as a table

```
data("presidential")
```

Q9. Create a matrix with 4 rows that contain the numbers 1 up to 12

```
matrix( 1:12, byrow = TRUE, nrow = 4)
```

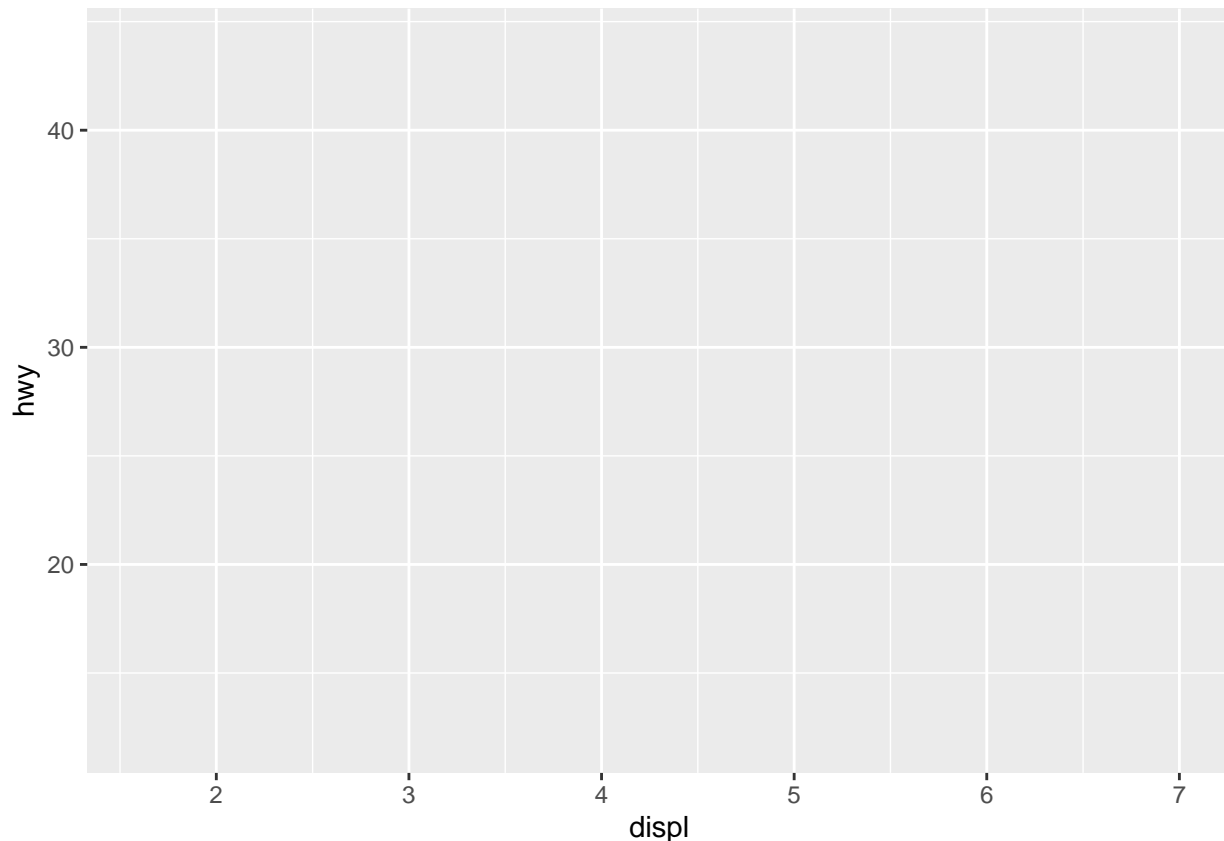
```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
## [4,]   10   11   12
```

Q10. A line of code that assigns displ column as x-axis and hwy column as y-axis of mpg data to a variable mpg_plot using ggplot2 package

```
mpg_plot <- ggplot(data = mpg, aes(x = displ, y = hwy))
```

Q11. Two lines of code that create a scatter plot of a variable mpg_plot that you have made in Q10

```
ggplot(data = mpg, aes(x = displ, y = hwy))
```



```
geom_bar()
```

```
## geom_bar: width = NULL, na.rm = FALSE, orientation = NA
## stat_count: width = NULL, na.rm = FALSE, orientation = NA
## position_stack
```

Q12. Three lines of code that create subplots (four rows) by `class` column, using two lines of code for Q10.

Q13. A line of code that returns dimension information of `presidential` data

Q14. What are the unique values of `party` column of `presidential` data?

```
unique(presidential$party)
```

```
## [1] "Republican" "Democratic"
```

Q15. Two lines of code that will directly create a simple stacked bar plot that shows the count by `class` column of `mpg` data with filling color by `trans` column

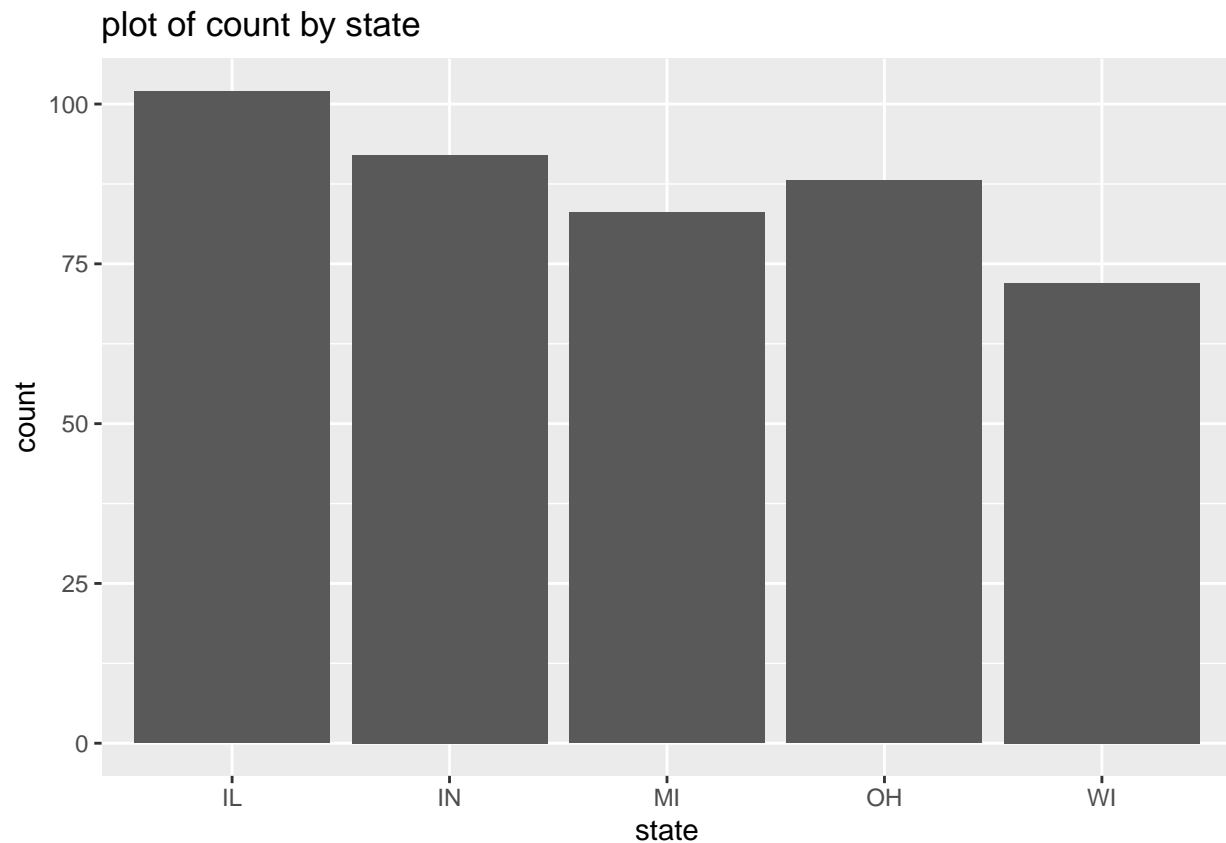
```
mpg_plot <- ggplot(data = mpg, aes (x = class, y = trans)) +
  geom_bar()
```

Q16. A line of code that assigns `state` column as x position of `midwest` data to a variable `midwest_plot` using `ggplot2` package

```
midwest_plot <- ggplot(data = midwest, aes (x = state))
```

Q17. Five lines of code that will return a bar plot of the `midwest_plot` variable with a title `Plot of count by state`. X-axis is labeled as `state` and y-axis as `count`.

```
midwest_plot +
  geom_bar() +
  ggtitle("plot of count by state") +
  xlab("state")
```



```
ylab("count")
```

```
## $y
## [1] "count"
##
## attr(,"class")
## [1] "labels"
```

Q18. What is the name of 7th column of **diamonds** dataset?

```
colnames(diamonds)[7]
```

```
## [1] "price"
```

Q19. How many columns and rows does **midwest** data have?

```
dim(midwest)
```

```
## [1] 437 28
```

Q20. Two different commands for a quick overview of **mpg** data that we have learned in our class

```
summary(mpg)
```

```
## manufacturer      model      displ      year
```

```
## Length:234      Length:234      Min.   :1.600   Min.   :1999
## Class :character Class :character 1st Qu.:2.400   1st Qu.:1999
## Mode  :character Mode  :character Median :3.300   Median :2004
##                                     Mean  :3.472   Mean  :2004
##                                     3rd Qu.:4.600   3rd Qu.:2008
##                                     Max.   :7.000   Max.   :2008
##      cyl      trans      drv      cty
## Min.   :4.000   Length:234   Length:234   Min.   : 9.00
## 1st Qu.:4.000   Class :character Class :character 1st Qu.:14.00
## Median :6.000   Mode  :character Mode  :character Median :17.00
## Mean    :5.889                                     Mean  :16.86
## 3rd Qu.:8.000                                     3rd Qu.:19.00
## Max.    :8.000                                     Max.   :35.00
##      hwy      fl      class
## Min.   :12.00   Length:234   Length:234
## 1st Qu.:18.00   Class :character Class :character
## Median :24.00   Mode  :character Mode  :character
## Mean    :23.44
## 3rd Qu.:27.00
## Max.    :44.00
```

```
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
## $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
## $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
## $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
## $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
## $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
## $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
## $ drv         : chr [1:234] "f" "f" "f" "f" ...
## $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
## $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
## $ fl          : chr [1:234] "p" "p" "p" "p" ...
## $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

End of document