

HW3__Porter__Erica

Erica Porter

9/13/2017

Problem 3

Takeaways from the style guides include:

- Consistency is important; it is best to develop a sufficiently neat programming style and use a similar format throughout multiple files/projects.
- Spacing is essential for writing and reading code; be wary of using tabs.
- Identifiers and variable names should not have excessive special characters and should follow neat, standard naming conventions
- Try to neatly indent code/functions/processes with different types of commands (e.g. when using tidy, each "verb" should be on its own line).

I will try to improve my own code by:

- Weave my code and text a more effectively for R Markdown documents requiring code, text, and explanations.
- Format, label tables and printed results better (e.g. explore grid functions, stargazer, graph packages).
- Keeping track of variable and data frame names better.
- Including effective spacing and code chunks to improve the appearance and progression of .Rmd files, rather than solely creating a neat PDF.

Problem 4

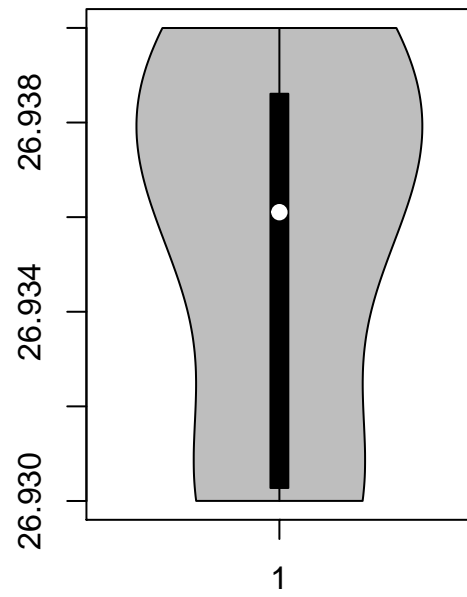
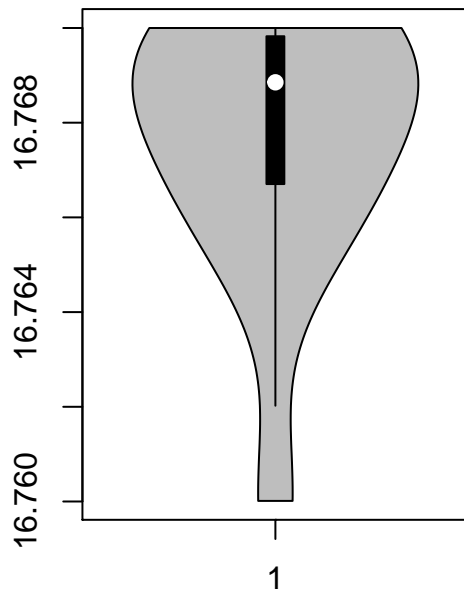
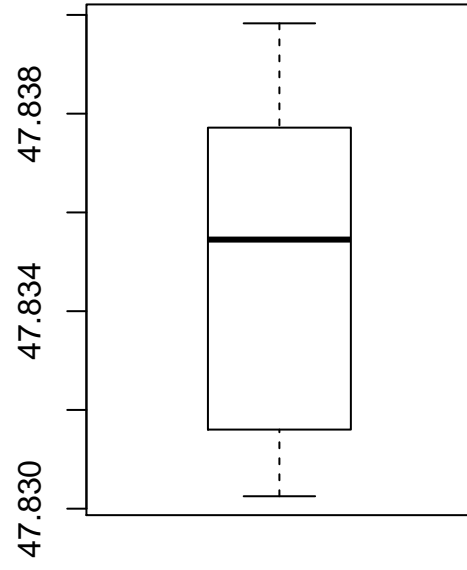
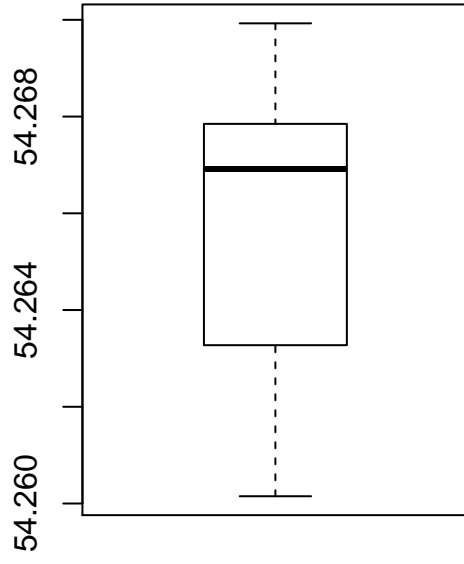
Some suggestions for stylistic improvements included:

- Inserting spaces after commas and around operators
- Format functions better (e.g. opening and closing curly braces on their own line)
- Limit lines to <80 characters
- Use <- for assignment rather than =
- Consistently use double-quotes rather than single-quotes
- Avoid using absolute paths if possible
- Variable and function names should be lowercase
- Avoid using absolute paths

Problem 5

Table 1: Compare dev means

| mean1 | mean2 | sd1 | sd2 | corr |
|----------|----------|----------|----------|------------|
| 54.26610 | 47.83472 | 16.76983 | 26.93974 | -0.0641284 |
| 54.26873 | 47.83082 | 16.76924 | 26.93573 | -0.0685864 |
| 54.26732 | 47.83772 | 16.76001 | 26.93004 | -0.0683434 |
| 54.26327 | 47.83225 | 16.76514 | 26.93540 | -0.0644719 |
| 54.26030 | 47.83983 | 16.76774 | 26.93019 | -0.0603414 |
| 54.26144 | 47.83025 | 16.76590 | 26.93988 | -0.0617148 |
| 54.26881 | 47.83545 | 16.76670 | 26.94000 | -0.0685042 |
| 54.26785 | 47.83590 | 16.76676 | 26.93610 | -0.0689797 |
| 54.26588 | 47.83150 | 16.76885 | 26.93861 | -0.0686092 |
| 54.26734 | 47.83955 | 16.76896 | 26.93027 | -0.0629611 |
| 54.26993 | 47.83699 | 16.76996 | 26.93768 | -0.0694456 |
| 54.26692 | 47.83160 | 16.77000 | 26.93790 | -0.0665752 |
| 54.26015 | 47.83972 | 16.76996 | 26.93000 | -0.0655833 |



Problem 6

The Blood Pressure data from Wu and Humada needs to be reformatted/tidied because the measurements for blood pressure span six different columns, the devices measurements are not grouped together, and doctor measurements are not all grouped together. I will use the `gather`, `separate`, and `mutate` commands in `tidyr` to create a single column for day (the original data has an extraneous column for day) and columns for measuring entity, associated measure number, and the measurement value. Below I have printed the first 5 rows of the tidy data set and a summary table describing the data. See the Appendix for full R code.

Table 2: First 5 observations for Blood Pressure

| Day | method | replicate | value |
|-----|--------|-----------|--------|
| 1 | Dev | 1 | 133.34 |
| 2 | Dev | 1 | 110.94 |
| 3 | Dev | 1 | 118.54 |
| 4 | Dev | 1 | 137.94 |
| 5 | Dev | 1 | 139.52 |

Table 3: Summary of Blood Pressure data

| Day | method | replicate | value |
|------------------|------------------|------------------|---------------|
| Length:90 | Length:90 | Length:90 | Min. :110.8 |
| Class :character | Class :character | Class :character | 1st Qu.:125.5 |
| Mode :character | Mode :character | Mode :character | Median :130.4 |
| NA | NA | NA | Mean :129.0 |
| NA | NA | NA | 3rd Qu.:134.3 |
| NA | NA | NA | Max. :139.6 |

Problem 7

```
## $solution
## [1] -9.162986
##
## $`number iterations`
## [1] -9.109611 -9.162556 -9.162986 -9.162986
##
Read 4.8% of 1048575 rows
Read 70.6% of 1048575 rows
Read 1048575 rows and 13 (of 13) columns from 0.093 GB file in 00:00:04
```

Table 4: Number of Unique Makes and Models

| unique_make | unique_model |
|-------------|--------------|
| 38 | 405 |

Table 5: Top Defects, their makes, and models

| top_defects | top_defect_makes | top_defect_models |
|-------------|------------------|---------------------|
| K04 | VOLKSWAGEN | GOLF PLUS |
| AC1 | VOLKSWAGEN | FOX |
| G05 | PEUGEOT | 307; SW 2.0HDI 66KW |
| K05 | VOLKSWAGEN | GOLF |
| J03 | OPEL | AGILA; Z1.2XE |

Table 6: Description of linear fit by Make

| <i>Dependent variable:</i> | |
|--|----------------------|
| n | |
| Make | 0.593** (0.282) |
| Constant | 4.788 (6.301) |
| Observations | 38 |
| R ² | 0.109 |
| Adjusted R ² | 0.085 |
| Residual Std. Error | 19.040 (df = 36) |
| F Statistic | 4.425** (df = 1; 36) |
| <i>Note:</i> *p<0.1; **p<0.05; ***p<0.01 | |

Table 7: ANOVA of linear fit by Make

| Statistic | N | Mean | St. Dev. | Min | Max |
|-----------|---|-----------|-----------|-----------|------------|
| Df | 2 | 18.500 | 24.749 | 1 | 36 |
| Sum Sq | 2 | 7,327.276 | 8,093.601 | 1,604.236 | 13,050.320 |
| Mean Sq | 2 | 983.372 | 878.034 | 362.509 | 1,604.236 |
| F value | 1 | 4.425 | | 4.425 | 4.425 |
| Pr(>F) | 1 | 0.042 | | 0.042 | 0.042 |

Table 8: Description of linear fit by Model

| <i>Dependent variable:</i> | |
|--|---------------------|
| n | |
| Model | 0.0001 (0.001) |
| Constant | 1.518*** (0.121) |
| Observations | 405 |
| R ² | 0.0001 |
| Adjusted R ² | -0.002 |
| Residual Std. Error | 1.213 (df = 403) |
| F Statistic | 0.022 (df = 1; 403) |
| <i>Note:</i> *p<0.1; **p<0.05; ***p<0.01 | |

There appears to be a relationship between car make and number of defects.

Table 9: ANOVA of linear fit by Model

| Statistic | N | Mean | St. Dev. | Min | Max |
|-----------|---|---------|----------|-------|---------|
| Df | 2 | 202.000 | 284.257 | 1 | 403 |
| Sum Sq | 2 | 296.400 | 419.126 | 0.033 | 592.767 |
| Mean Sq | 2 | 0.752 | 1.017 | 0.033 | 1.471 |
| F value | 1 | 0.022 | | 0.022 | 0.022 |
| Pr(>F) | 1 | 0.881 | | 0.881 | 0.881 |

This workflow could certainly be improved; I had trouble finding the most common makes and models for each defects, so I ended up creating many separate data sets and using `subset` and `count` multiple times. This does not seem very efficient to me and I would have liked to use the `subset` and `count` functions to make the table of common values in one step (or at least fewer).

Appendix: R code

```
## Problem 4 ##
## Get stylistic comments about homework 2 ##

lint(filename = ("02_data_munging_summarizing_R_git/HW2_Porter_Erica.Rmd"))
```

```
## Problem 5 ## Write a function to determine summary
## statistics for device measurements ##

dev_data <- readRDS("./HW3_data.rds")

# Initialize empty vectors to receive each of the stats for
# the observations #
mean1 <- c()
mean2 <- c()
sd1 <- c()
sd2 <- c()
corr <- c()

# Loop through the data to evaluate mean, sd, and correlation
# for each of 13 observers #
for (i in 1:13) {
  mean1[i] <- mean(subset(dev_data, Observer == i)$dev1)
  mean2[i] <- mean(subset(dev_data, Observer == i)$dev2)
  sd1[i] <- sd(subset(dev_data, Observer == i)$dev1)
  sd2[i] <- sd(subset(dev_data, Observer == i)$dev2)
  corr[i] <- cor(subset(dev_data, Observer == i)$dev1, subset(dev_data,
    Observer == i)$dev2)
}

summary_stats <- cbind(mean1, mean2, sd1, sd2, corr)
summary_stats <- as.data.frame(summary_stats)
```

```
## Problem 5 ## Paste the summary statistics into a data frame
## to print ##
knitr::kable(summary_stats, caption = "Compare dev means")
```

```
## Problem 5 ## Create boxplots to compare dev1 and dev2 means
## ##
par(mfrow = c(1, 2))
boxplot(mean1, data = summary_stats)
boxplot(mean2, data = summary_stats)
vioplot(summary_stats$sd1, col = "gray")
vioplot(summary_stats$sd2, col = "gray")
```

```
## Problem 6 ## Tidy the Blood Pressure data set ##
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BloodPressure.dat"
blood_pressure <- read.table(url, header = F, skip = 1, fill = T,
  stringsAsFactors = F)
```

```

blood_pressure_tidy <- blood_pressure[-1, -5]
colnames(blood_pressure_tidy) <- c("Day", "Dev_1", "Dev_2", "Dev_3",
  "Doc_1", "Doc_2", "Doc_3")
blood_pressure_tidy <- blood_pressure_tidy %>% gather(measure_num,
  value, Dev_1:Doc_3) %>% separate(measure_num, into = c("method",
  "replicate"), sep = "_") %>% mutate(value = as.numeric(value))

# Print first 5 observations and a summary table
knitr::kable(head(blood_pressure_tidy, n = 5), caption = "First 5 observations for Blood Pressure")
knitr::kable(summary(blood_pressure_tidy), caption = "Summary of Blood Pressure data")

## Another method I tried for tidying Blood Pressure Data that
## I wanted to keep for reference ##
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BloodPressure.dat"
blood_pressure <- read.table(url, header = F, skip = 1, fill = T,
  stringsAsFactors = F)
blood_pressure_tidy <- blood_pressure[-1, ]
colnames(blood_pressure_tidy) <- c("Day_1", "Dev_1", "Dev_2",
  "Dev_3", "Day_2", "Doc_1", "Doc_2", "Doc_3")
devices <- blood_pressure_tidy[, c("Dev_1", "Dev_2", "Dev_3")]
doctors <- blood_pressure_tidy[, c("Doc_1", "Doc_2", "Doc_3")]
devices <- devices %>% gather(Device, device_read, Dev_1:Dev_3) %>%
  mutate(Device = gsub("Dev_", "", Device))
doctors <- doctors %>% gather(Doctor, doc_read, Doc_1:Doc_3) %>%
  mutate(Doctor = gsub("Doc_", "", Doctor))
Day <- rep(1:15, 3)
blood_pressure_tidy <- cbind(Day, devices, doctors)
blood_pressure_tidy <- as.numeric(c(blood_pressure_tidy$device_read,
  blood_pressure_tidy$doc_read))

```

```

## Problem 7 ##

# This is my first attempt at a function for Newton's method
# # This generates a solution without displaying intermediate
# iterations # Begin with a starting value for x Obtain an
# estimate  $x - f(x)/f'(x)$  Repeat while the tolerance
# conditions are satisfied

fun <- function(x) {
  3^x - sin(x) + cos(5 * x)
}
der <- function(x) {
  log(3) * 3^x - 5 * sin(5 * x) - cos(x)
}
newton <- function(fun, der, a, t = 0.01) {
  b <- a - fun(a)/der(a)
  while ((abs(a - b) > t) & (abs(a - b)/(abs(a) + abs(b))) >
    t) {
    c <- a - fun(a)/der(a)
    b <- a
    a <- c
  }
  result <- a
}

```



```

        return(result)
    }
}

## Problem 7 ## This is my second attempt at Newton's method
## with iterations #

fun <- function(x) {
  3^x - sin(x) + cos(5 * x)
}
der <- function(x) {
  log(3) * 3^x - 5 * sin(5 * x) - cos(x)
}

newton2 <- function(f, a, b, t, n = 1000) {
  x_0 <- a
  k <- n
  for (i in 1:n) {
    deriv <- der(x_0)
    c <- x_0 - (fun(x_0)/deriv)
    k[i] <- c
    if (abs(c - x_0) < t) {
      estimate <- tail(k, n = 1)
      to_print <- list(solution = estimate, `number iterations` = k)
      return(to_print)
    }
  }
  x_0 <- c
}

newton2(fun, -10, 10, 1e-04, n = 100)

```

```

## Problem 8 ## Merge and analyze car data ##

# Read in data using fread since files are large
Car_Geb <- fread(input = "Open_Data_RDW__Gebreken.csv", header = T) #dat2
Car_Gec <- fread(input = "Open_Data_RDW__Geconstateerde_Gebreken.csv",
  header = T) #dat3
Car_Person <- fread(input = "Personenauto_basisdata.csv", header = T) #dat1

# Merge the 3 data sets by common columns
plates <- merge(Car_Person, Car_Gec, by = "Kenteken")
defects <- merge(plates, Car_Geb, by = "Gebrek identificatie")

# Select only columns of interest, rename columns in English
defects_small <- defects[, c(1, 2, 4, 5, 16, 24)]
colnames(defects_small) <- c("Defect Code", "License", "Make",
  "Model", "Inspection Date", "Defect Description")

# Check for NA values in the data set we will be working with
check_na <- sum(is.na(defects_small$`Defect Code`))

```

```

# Subset all rows from year 2017 and count unique Makes and
# Models
defects17 <- defects_small[grepl("2017", defects_small$`Inspection Date`),
]
unique_make <- length(unique(defects17$Make))
unique_model <- length(unique(defects17$Model))
knitr::kable(as.data.frame(cbind(unique_make, unique_model)),
  caption = "Number of Unique Makes and Models")

# Find top 5 defect codes
common_defects <- as.data.frame(defects17 %>% count(`Defect Code`,
  sort = TRUE))
top_defects <- common_defects$`Defect Code`[1:5]

# Find most common make for each of above defects
defect_make1 <- (as.data.frame(subset(defects17, `Defect Code` ==
  top_defects[1]) %>% count(Make, sort = TRUE)))[1, 1]
defect_make2 <- (as.data.frame(subset(defects17, `Defect Code` ==
  top_defects[2]) %>% count(Make, sort = TRUE)))[1, 1]
defect_make3 <- (as.data.frame(subset(defects17, `Defect Code` ==
  top_defects[3]) %>% count(Make, sort = TRUE)))[1, 1]
defect_make4 <- (as.data.frame(subset(defects17, `Defect Code` ==
  top_defects[4]) %>% count(Make, sort = TRUE)))[1, 1]
defect_make5 <- (as.data.frame(subset(defects17, `Defect Code` ==
  top_defects[5]) %>% count(Make, sort = TRUE)))[1, 1]
top_defect_makes <- c(defect_make1, defect_make2, defect_make3,
  defect_make4, defect_make5)

# Find most common model for each of above makes
def1 <- subset(defects17, `Defect Code` == top_defects[1] & Make ==
  top_defect_makes[1])
def2 <- subset(defects17, `Defect Code` == top_defects[2] & Make ==
  top_defect_makes[2])
def3 <- subset(defects17, `Defect Code` == top_defects[3] & Make ==
  top_defect_makes[3])
def4 <- subset(defects17, `Defect Code` == top_defects[4] & Make ==
  top_defect_makes[4])
def5 <- subset(defects17, `Defect Code` == top_defects[5] & Make ==
  top_defect_makes[5])

defect_model1 <- (as.data.frame(def1 %>% count(Model, sort = TRUE)))[1,
  1]
defect_model2 <- (as.data.frame(def2 %>% count(Model, sort = TRUE)))[1,
  1]
defect_model3 <- (as.data.frame(def3 %>% count(Model, sort = TRUE)))[1,
  1]
defect_model4 <- (as.data.frame(def4 %>% count(Model, sort = TRUE)))[1,
  1]
defect_model5 <- (as.data.frame(def5 %>% count(Model, sort = TRUE)))[1,
  1]
top_defect_models <- c(defect_model1, defect_model2, defect_model3,
  defect_model4, defect_model5)

```

```

# Table of common defect codes, their makes, and models
defect_table <- as.data.frame(cbind(top_defects, top_defect_makes,
  top_defect_models))
knitr::kable(defect_table, caption = "Top Defects, their makes, and models")

# Check for relationship b/w number of defects and make, then
# by model
carmake <- as.data.frame(defects17 %>% count(Make, sort = TRUE))
carmake$Make <- as.numeric(factor(carmake$Make))
colnames(carmake) <- c("Make", "n")
carmodel <- as.data.frame(defects17 %>% count(Model, sort = TRUE))
carmodel$Model <- as.numeric(factor(carmodel$Model))

## Problem 8 ##
fit = lm(n ~ Make, data = carmake)
stargazer(fit, title = "Description of linear fit by Make", header = F,
  no.space = T, single.row = T)
stargazer(anova(fit), title = "ANOVA of linear fit by Make",
  header = F, no.space = T, single.row = T)

fit1 = lm(n ~ Model, data = carmodel)
stargazer(fit1, title = "Description of linear fit by Model",
  header = F, no.space = T, single.row = T)
stargazer(anova(fit1), title = "ANOVA of linear fit by Model",
  header = F, no.space = T, single.row = T)

```