

# Final Report Exercise

N Hartley

## Problem 1

Create a vector of five numbers of your choice between 0 and 10, save that vector to an object, and use the `sum()` function to calculate the sum of the numbers.

```
# Create a vector of five numbers between 0 and 10
my_vector <- c(3, 7, 2, 9, 4)

# Calculate the sum of the numbers in the vector
total_sum <- sum(my_vector)

# Print the sum
print(total_sum)
```

```
## [1] 25
```

## Problem 2

Create a data frame that includes two columns. One column must have the numbers 1 through 5, and the other column must have the numbers 6 through 10. The first column must be named “alpha” and the second column must be named “beta”. Name the object “my\_dat”. Display the data.

Put your code and solution here:

```
# Create a data frame with two columns
my_dat <- data.frame(alpha = 1:5, beta = 6:10)

# Display the data frame
print(my_dat)
```

```
##   alpha beta
## 1     1    6
## 2     2    7
## 3     3    8
## 4     4    9
## 5     5   10
```

## Problem 3

Using the data frame created in Problem 2, use the `summary()` command to create a five-number summary for the column named “beta”.

Put your code and solution here:

```
# Create a data frame with two columns
my_dat <- data.frame(alpha = 1:5, beta = 6:10)

# Use summary() to create a five-number summary for the column named "beta"
summary(my_dat$beta)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##         6         7         8         8         9        10
```

## Problem 4

There is code for importing the example survey data that will run automatically in the setup chunk for this report (Line 13). Using that data, make a boxplot of the Family Income column using the Base R function (not a figure drawn using qplot). Include your name in the title for the plot. Your name should be in the title. Relabel that x-axis as “Family Income”.

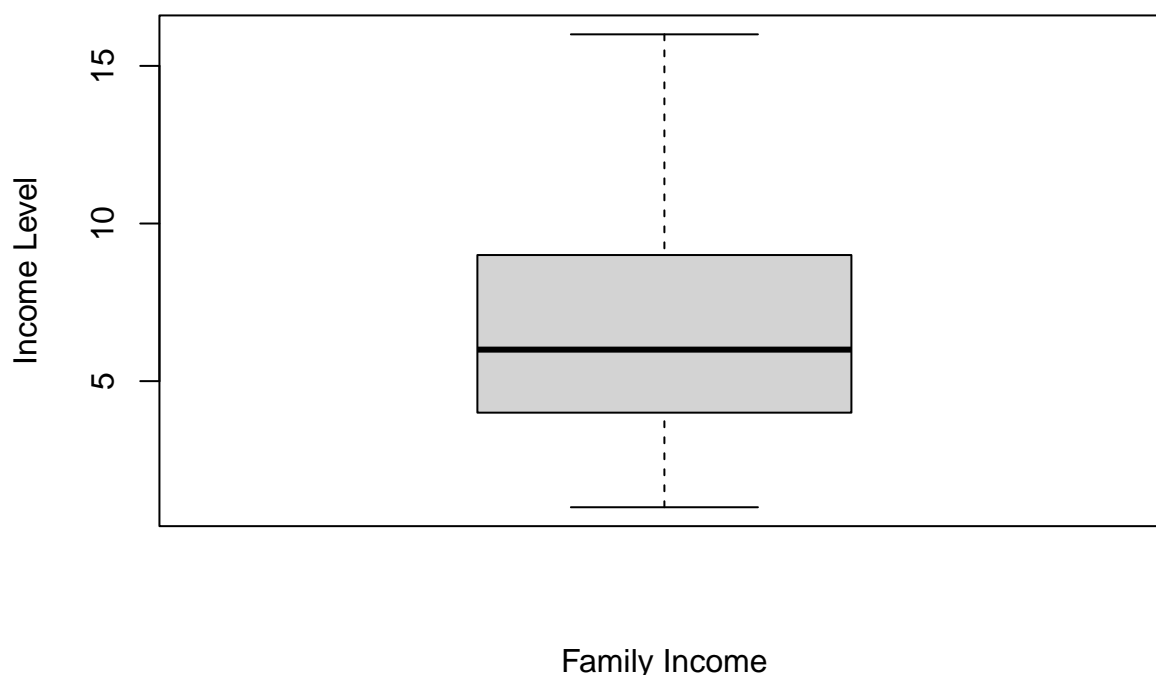
Hint: consult the codebook to identify the correct column name for the family income question.

Put your code and solution here:

```
# Adjust margins
par(mar=c(5, 4, 4, 2) + 0.1)

# Plotting
boxplot(dat$faminc_new,
        main = "Boxplot of Family Income_N Hartley",
        xlab = "Family Income",
        ylab = "Income Level")
```

## Boxplot of Family Income\_N Hartley



## Problem 5

Using the survey data, filter to subset the survey data so you only have male survey respondents who live in the northwest or midwest of the United States, are married, and identify as being interested in the news most of the time.

Use the `str()` function to provide information about the resulting dataset.

Put your code and solution here:

```
subset_dat <- dat[dat$gender == 1 & (dat$region == 1 | dat$region == 2) &
                  dat$marstat == 1 & dat$newsint == 1, ]
```

```
str(subset_dat)
```

```
## 'data.frame':    75 obs. of  25 variables:
##  $ caseid      : int  420208067 412948037 411855595 414480371 416806180 414729651 412021973 41234883
##  $ region      : int  2 1 1 1 2 2 1 2 1 2 ...
##  $ gender      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ educ        : int  3 5 6 5 3 2 6 5 5 5 ...
##  $ edloan      : int  2 1 2 1 1 2 2 2 2 2 ...
##  $ race        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hispanic    : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ employ      : int  1 1 1 1 1 5 5 1 1 1 ...
##  $ marstat     : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
## $ pid7      : int  4 1 4 4 6 2 1 1 1 7 ...
## $ ideo5     : int  5 1 3 3 3 1 2 3 1 5 ...
## $ pew_religimp: int  4 4 4 1 2 4 4 2 4 1 ...
## $ newsint   : int  1 1 1 1 1 1 1 1 1 1 ...
## $ faminc_new : int 10 11 10 9 10 2 13 8 7 11 ...
## $ union     : int  1 2 3 3 1 3 2 2 2 3 ...
## $ investor  : int  2 1 1 1 1 2 1 2 1 2 ...
## $ CC18_308a : int  4 4 4 1 2 3 4 4 4 1 ...
## $ CC18_310a : int  2 3 3 3 2 2 3 2 3 2 ...
## $ CC18_310b : int  2 3 3 3 2 3 3 5 3 3 ...
## $ CC18_310c : int  3 3 3 3 3 3 2 5 3 3 ...
## $ CC18_310d : int  2 5 3 3 3 2 2 2 3 3 ...
## $ CC18_325a : int  1 2 1 1 1 2 2 2 2 1 ...
## $ CC18_325b : int  2 2 1 1 1 1 2 1 2 2 ...
## $ CC18_325c : int  1 2 2 1 1 2 2 1 2 2 ...
## $ CC18_325d : int  1 1 1 1 1 1 1 1 2 1 ...
```

```
head(subset_dat)
```

```
##      caseid region gender educ edloan race hispanic employ marstat pid7 ideo5
## 5  420208067      2      1    3      2    1        2      1      1      4      5
## 8  412948037      1      1    5      1    1        2      1      1      1      1
## 19 411855595      1      1    6      2    1        2      1      1      4      3
## 25 414480371      1      1    5      1    1        2      1      1      4      3
## 41 416806180      2      1    3      1    1        2      1      1      6      3
## 56 414729651      2      1    2      2    1        2      5      1      2      1
##      pew_religimp newsint faminc_new union investor CC18_308a CC18_310a CC18_310b
## 5           4      1          10      1        2      4      2      2
## 8           4      1          11      2        1      4      3      3
## 19          4      1          10      3        1      4      3      3
## 25          1      1           9      3        1      1      3      3
## 41          2      1          10      1        1      2      2      2
## 56          4      1           2      3        2      3      2      3
##      CC18_310c CC18_310d CC18_325a CC18_325b CC18_325c CC18_325d
## 5           3      2      1      2      1      1
## 8           3      5      2      2      2      1
## 19          3      3      1      1      2      1
## 25          3      3      1      1      1      1
## 41          3      3      1      1      1      1
## 56          3      2      2      1      2      1
```

## Problem 6

Filter the data the same as in Problem 5. Use a R function to create a frequency table for the responses for the question asking whether these survey respondents are invested in the stock market.

Put your code and solution here:

```
# Ensure the 'investor' column is a factor with all expected levels
subset_dat$investor <- factor(subset_dat$investor, levels = c(1, 2, 3))

# create the frequency table
```

```
freq_table <- table(subset_dat$investor)

names(freq_table) <- c("Yes, currently a member", "Formerly was a member", "Not now nor have been a member")

# Print the named frequency table
print(freq_table)
```

```
##           Yes, currently a member           Formerly was a member
##                               57                               18
## Not now nor have been a member
##                               0
```

## Problem 7

Going back to using all rows in the dataset, create a new column in the data using mutate that is equal to either 0, 1, or 2, to reflect whether the respondent supports increasing the standard deduction from 12,000 to 25,000, supports cutting the corporate income tax rate from 39 to 21 percent, or both (so, support for neither policy equals 0, one of the two policies equals 1, and both policies equals two). Name the column “tax\_scale”. Hint: you’ll need to use recode() as well.

Display the first twenty elements of the new column you create.

Put your code and solution here:

```
library(dplyr)

# Assuming dat is your dataset
dat <- dat %>%
  mutate(tax_scale = recode(
    (CC18_325a == 1) + (CC18_325d == 1), # This creates a logical sum, where supporting both policies equals 2
    `0` = 0, `1` = 1, `2` = 2
  ))

# Display the first twenty elements of the new column "tax_scale"
print(dat$tax_scale[1:20])
```

```
## [1] 2 1 1 1 2 2 2 1 0 1 0 2 0 1 0 1 1 1 2 1
```

## Problem 8

Use a frequency table command to show how many 0s, 1s, and 2s are in the column you created in Problem 7.

Put your code and solution here:

```
# Create a frequency table for the 'tax_scale' column
tax_scale_freq <- table(dat$tax_scale)

# Print the frequency table
print(tax_scale_freq)
```

```
##
##    0    1    2
## 130 408 331
```

## Problem 9

Again using all rows in the original dataset, use `summarise` and `group_by` to calculate the average (mean) job of approval for President Trump in each of the four regions listed in the “region” column.

Put your code and solution here:

```
library(dplyr)

# Calculating the average job approval for President Trump by region
average_approval <- dat %>%
  group_by(region) %>%
  summarise(average_approval = mean(CC18_308a, na.rm = TRUE))

# Display the results
print(average_approval)
```

```
## # A tibble: 4 x 2
##   region average_approval
##   <int>         <dbl>
## 1     1           2.77
## 2     2           2.76
## 3     3           2.71
## 4     4           3.03
```

## Problem 10

Again start with all rows in the original dataset, use `summarise()` to create a summary table for survey respondents who are not investors and who have an annual family income of between \$40,000 and \$119,999 per year. The table should have the mean, median and standard deviations for the importance of religion column.

Put your code and solution here:

```
library(dplyr)

# use dat data frame
summary_table <- dat %>%
  filter(investor != 1, faminc_new >= 5, faminc_new <= 10) %>%
  summarise(
    mean_religion = mean(pew_religimp, na.rm = TRUE),
    median_religion = median(pew_religimp, na.rm = TRUE),
    sd_religion = sd(pew_religimp, na.rm = TRUE)
  )

# Display the summary table
print(summary_table)
```

```
##   mean_religion median_religion sd_religion
## 1      2.325893           2      1.188906
```

## Problem 11

Use `kable()` and the `summarise()` function to create a table with one row and three columns that provides the mean, median, and standard deviation for the column named `faminc_new` in the survey data.

Put your code and solution here:

```
# Create summary table
summary_table <- dat %>%
  summarise(
    Mean = mean(faminc_new, na.rm = TRUE),
    Median = median(faminc_new, na.rm = TRUE),
    SD = sd(faminc_new, na.rm = TRUE)
  ) %>%
  kable()

# Print table
print(summary_table)
```

```
##
##
## |      Mean| Median|      SD|
## |-----:|-----:|-----:|
## | 6.581128|      6| 3.247035|
```

## Problem 12

With the survey data, use `qplot()` to make a histogram of the column named `pid7`. Change the x-axis label to “Seven Point Party ID” and the y-axis label to “Count”.

Note: you can ignore the “`stat_bin()`” message that R generates when you draw this. The setup for the code chunk will suppress the message.

Put your code and solution here:

```
# Load necessary libraries
library(ggplot2)

# Create a histogram
ggplot(dat, aes(x = as.factor(pid7))) + # Ensure pid7 is treated as a factor
  geom_bar(fill = "blue", color = "black") +
  labs(x = "Seven Point Party ID", y = "Count", title = "Histogram of Seven Point Party ID") +
  theme_minimal()
```

