Final Exam Review Key

Coding and graphical questions

Problem 1

Describe a way or sketch out R code to find the mean of the cost vector below. Note that `mean(cost)` will give an error.

```
cost <- c("$1100", "$250.12", "$675")
```

General steps:

1. Remove `$` character
2. Convert to numeric
3. Take mean

Here is one possible solution:

```
library(stringr)
cost <- str_remove(cost, "\\$")
cost
```

```
## [1] "1100"   "250.12" "675"
```

```
cost <- as.numeric(cost)
cost
```

```
## [1] 1100.00  250.12  675.00
```

```
mean(cost)
```

```
## [1] 675.04
```

Problem 2

Consider the following data frame:

```
msu.football <- data.frame(opponent = c("Washington State", "South Dakota State",
                                         "North Dakota", "Weber State", "Univ of Montana"),
                           points = c(0, 27, 49, 17, 134),
                           outcome = c("Loss", "Loss", "Win", "Loss", "Win"))
```

For each part below, explain what each line of code is doing (how each line of code helps produce the output). Then write the R output from the code below. Exactly one part will produce an error.

a.

```
for (i in 1:nrow(msu.football)){
  print(msu.football[i, 2])
}
```

```
## [1] 0
## [1] 27
## [1] 49
## [1] 17
## [1] 134
```

b.

```
msu.football %>% filter(outcome == "Loss") %>%
  summarize(MaxPoints = max(points))
```

```
##   MaxPoints
## 1        27
```

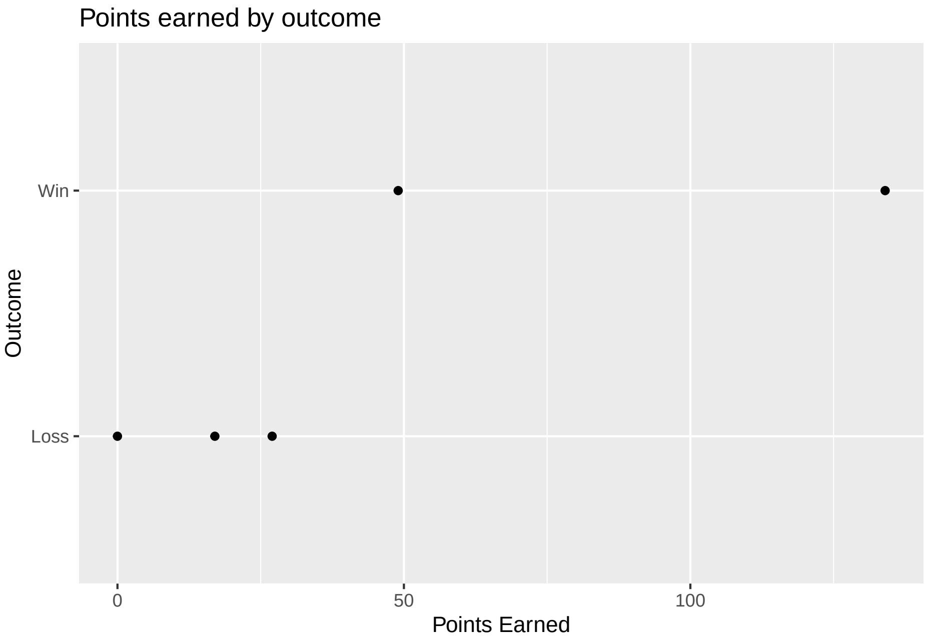
c.

```
msu.football %>% select(points) %>%
  group_by(outcome) %>%
  count()
```

```
## Error in `group_by()`:
## ! Must group by variables found in `.data`.
## x Column `outcome` is not found.
```

d.

```
msu.football %>% ggplot(aes(x = points, y = outcome)) +
  geom_point() +
  labs(x = "Points Earned",
   y = "Outcome") +
  ggtitle("Points earned by outcome")
```



e.
```
msu.football %>% select(opponent) %>%
  mutate(CatGriz = case_when(
           opponent == "Univ of Montana" ~ "Yes",
           opponent != "Univ of Montana" ~ "No"
           )
         )
```

```
##                opponent CatGriz
## 1   Washington State      No
## 2 South Dakota State      No
## 3       North Dakota      No
## 4        Weber State      No
## 5    Univ of Montana     Yes
```

Problem 3

Describe a strategy to merge the two data frames defined below without losing any information (i.e., keep all rows from `df1` and all rows from `df1`), then write the output you'd expect to see.

```
df1 <- data.frame(school = c("MSU", "VT", "Mines", "Luther"),
                  state = c("MT", "VA", "CO", "IA"))
df2 <- data.frame(college = c("Mines", "MSU", "VT"),
                  enrollment = c(5794, 15688, 30598))
```

General steps:

1. Noticing that `school` and `college` serve as the variable on which to merge, rename one of the variables so that the two names match.
2. Perform a full join by `school/college`.

Here is one possible solution:

```
df2 <- df2 %>% rename(school = college)
full_join(df1, df2)
```

```
## Joining, by = "school"
```

```
##   school state enrollment
## 1    MSU    MT      15688
## 2     VT    VA      30598
## 3  Mines    CO       5794
## 4 Luther    IA         NA
```

Problem 4

You would like to write a function that will take our original sample comprised of a single quantitative variable and create a bootstrap confidence interval for the true population mean, with a user-specified confidence level. Fill in the function below by adding R code or pseudocode wherever you see `<***>`.

One possible solution:

```r
bootstrap_means <- function(dat, conf.level = 0.95, reps = 1000) {
  # Function to generate a bootstrap distribution of means,
  # and calculate and report a confidence interval for the mean.
  # ARGS:
  #   dat = sample data (numerical vector)
  #   conf.level = confidence level as a decimal (defaults to 0.95)
  #   reps = number of bootstrap samples to generate (defaults to 1000)
  # RETURNS: a histogram plotting `reps` simulated means of size `sample_size`

  # Set up vector to store bootstrapped means
  means <- vector("numeric", length = reps)

  # Create bootstrap distribution of means
  for(i in 1:reps) {
    # Generate bootstrap sample
    ind <- sample(1:length(dat), length(dat), replace = TRUE)
    boot_samp <- dat[ind]

    # Calculate and store mean of bootstrap sample
    means[i] <- mean(boot_samp)
  }

  # Use distribution of bootstrapped means to calculate confidence interval
  quantile(means, c((1 - conf.level)/2, conf.level + (1 - conf.level)/2))
}

bootstrap_means(rnorm(50))
```

```
##       2.5%      97.5%
## -0.3259990  0.1502774
```