Format:

In order to receive full credit for the practice problems, you need to submit an R script file. All the non-R codes need to be placed after the # signs. You also need to write your name on top of the page. Please make sure to include your name as part of the file name.

To complete this assignment, you need to use what you've learned from this course.

You can only ask questions for clarification purposes on the Discussion Board. That is to say, you can't ask questions about how to answer a specific question or help you to debug your program.

Problems 1

The following code runs regression by using `mpg` as the outcome and `wt` as the X variable for each type of cylinder (`cyl`).

```
> library(tidyverse)
> models <- mtcars %>%
+     split(.$cyl) %>%
+     map(function(df) lm(mpg ~ wt, data = df))
> models

$`4`

Call:
lm(formula = mpg ~ wt, data = df)

Coefficients:
(Intercept)           wt
     39.571       -5.647


$`6`

Call:
lm(formula = mpg ~ wt, data = df)

Coefficients:
(Intercept)           wt
      28.41        -2.78


$`8`

Call:
lm(formula = mpg ~ wt, data = df)
```

```
Coefficients:
(Intercept)           wt
     23.868       -2.192
```

Based on the result above, extracts the coefficients of each model by using a series of `map` functions. Here's the expected result:

```
        4         6         8
-5.647025 -2.780106 -2.192438
```

Hint:

```
> models [[1]]

Call:
lm(formula = mpg ~ wt, data = df)

Coefficients:
(Intercept)           wt
     39.571       -5.647

> summary(models [[1]])

Call:
lm(formula = mpg ~ wt, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-4.1513 -1.9795 -0.6272  1.9299  5.2523

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   39.571      4.347   9.104 7.77e-06 ***
wt            -5.647      1.850  -3.052   0.0137 *
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 3.332 on 9 degrees of freedom
Multiple R-squared:  0.5086,        Adjusted R-squared:  0.454
F-statistic: 9.316 on 1 and 9 DF,  p-value: 0.01374

> summary(models [[1]])$coefficients

             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 39.571196   4.346582  9.103980 7.771511e-06
wt          -5.647025   1.850119 -3.052251 1.374278e-02

> summary(models [[1]])$coefficients[2,1]

[1] -5.647025
```

Problems 2

Create a function to calculate the skewness of a given vector based on the formula below.

$$\text{Skew}(x) = \frac{\frac{1}{n-2}\left(\sum_{i=1}^{n}(x_i - \bar{x})^3\right)}{\text{Var}(x)^{3/2}}.$$

Here's the expected output:

```
> skewness(c(1, 2, 5, 100))

[1] 1.494554
```

Problems 3

Create a function, called `miss_count`, which counts the number of missing and non-missing values of each variable from a given data. The result should be a matrix or a data frame with first columns contains the number of missing values, and the second column contains number of non-missing values. You will use `patient_tib` to test this function. Here's the expected result.

```
> load("patient_tib.RData")
> patient_tib

# A tibble: 10 x 8
   ID     GLUC   TGL   HDL   LDL HRT   MAMM  SMOKE
   <fct> <int> <int> <int> <int> <fct> <fct> <fct>
 1 A        88    NA    32    99 Y     <NA>  ever
 2 B        NA   150    60    NA <NA>  no    never
 3 C       110    NA    NA   120 N     <NA>  <NA>
 4 D        NA   200    65   165 <NA>  yes   never
 5 E        90   210    NA   150 Y     <NA>  never
 6 F        88    NA    32   210 <NA>  yes   ever
 7 G       120   164    NA    NA Y     yes   <NA>
 8 H       110   170    70   188 <NA>  <NA>  ever
 9 I        NA   190    NA   190 N     no    <NA>
10 J        90    NA    75    NA <NA>  yes   never

> miss_count(patient_tib)

      N_Miss N_non_missing
ID         0            10
GLUC       3             7
TGL        4             6
HDL        4             6
LDL        3             7
HRT        5             5
MAMM       4             6
SMOKE      3             7
```

Problems 4

For this problem, you need to create a function, called `continuous_stats`. This function takes only two arguments:

`...` : one or more variable names in `dat` (the second argument of this function) that you would like to compute descriptive statistics

`dat` : the name of the data frame

The function returns a matrix that contains the descriptive statistics, including mean, median, standard deviation (SD), number of non missing values (N_non_missing), and number of missing values (N_missing), for each of the variable. Here are the expected results by calling this function.

```
> continuous_stats("LDL", dat=patient_tib)

                    LDL
Mean          160.28571
Median        165.00000
SD             40.06126
N_non_missing   7.00000
N_missing       3.00000

> continuous_stats("TGL", "LDL", dat=patient_tib)

                   TGL       LDL
Mean          180.6667 160.28571
Median        180.0000 165.00000
SD             23.0362  40.06126
N_non_missing   6.0000   7.00000
N_missing       4.0000   3.00000

> continuous_stats("TGL", "HDL", "LDL", dat=patient_tib)

                   TGL      HDL       LDL
Mean          180.6667 55.66667 160.28571
Median        180.0000 62.50000 165.00000
SD             23.0362 19.00175  40.06126
N_non_missing   6.0000  6.00000   7.00000
N_missing       4.0000  4.00000   3.00000
```