Anastashia Pelletier
MATH 494
Assignment 4

1.

Part A:

R Code:

```
library(readr)

library(tidyverse)

library(ggpubr)


#Part 1.A: Importing File and Creating a Subset with Needed
Variables

Teams <- read_csv("MATH 494/Teams.csv")


teamW <- select(Teams, c(yearID, teamID, W, L, R, RA))
```

Part B:

R Code:

```
#Part 1.B: Calculate Winning Percentage

winF <- function(df){

  wPer <- with(df, (W/(W+L))*100)

  df <- cbind(df, wPer)

}


teamW <- winF(teamW)
```

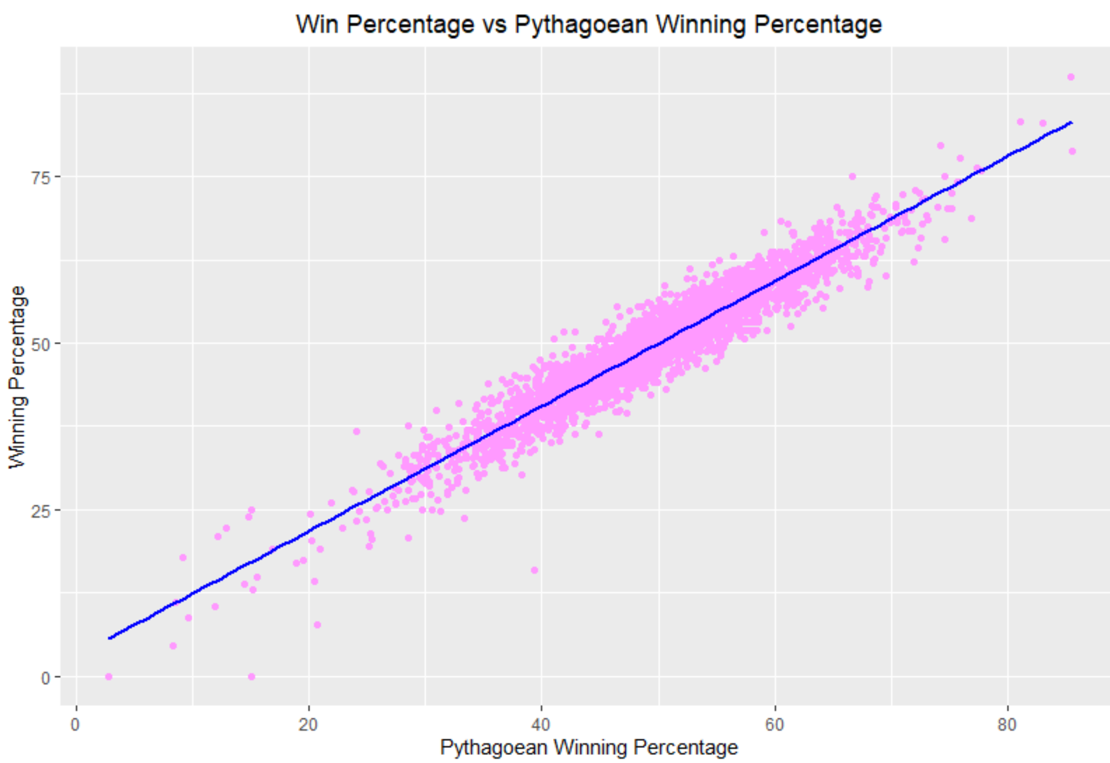
Part C:

R Code:

```
#Part 1.C: Calculate Pythagorean Winning Percentage
winPythF <- function(df,n){
  wPythD <- with(df, (R^n)+(RA^n))
  wPythN <- with(df, R^n)
  wPyth <- (wPythN/wPythD)*100
  df <- cbind(df, wPyth)
}
##Function will be used for another part of the assignment

teamW <- winPythF(teamW, 2)
```

Part D:

Scatterplot:



R Code:

```
#Part 1.D: Create Win Per vs Pyth. Win Per Scatterplot
WinP_Pyth <- ggplot(teamW, aes(wPyth, wPer)) +
  geom_point(color="#ff99ff") +
  geom_smooth(method = "lm", se=FALSE, color="blue") +
  labs(title="Win Percentage vs Pythagoean Winning
  Percentage",
       x="Pythagoean Winning Percentage",
       y="Winning Percentage") +
  theme(plot.title = element_text(hjust = 0.5))
WinP_Pyth
```

Part E:

Linear Model:

```
y = 2.998 + 0.939x
R = 0.955
R-Squared = 0.912
```

R Code:

```
#Part 1.E: Create Linear Model
model1 <- lm(wPer ~ wPyth, data=teamW)
summary(model1)

slope1 <- round(coef(model1)[2], 3)
int1 <- round(coef(model1)[1], 3)
r2_1 <- round(summary(model1)$r.squared, 3)
r_1 <- round(sqrt(r2_1), 3)

##Display Linear Model
glue::glue("y = {int1} + {slope1}x", "
           R = {r_1}", "
           R-Squared = {r2_1}")
```

2.

Part A:

To show that the Wins to Losses is approximately close to the square of Runs to Runs Allowed, we can use what we have learned from part 1, which is:

$$Winning\ Percent\ \approx Pythagorean\ Winning\ Percentage$$

$$\frac{Wins}{Wins + Losses} \approx \frac{Runs^2}{Runs^2 + Runs\ Allowed^2}\,.$$

If we treat the approximation as we would an equal, then we can cross multiple both sides to get:

$$Wins(Runs^2 + Runs\ Allowed^2) \approx\ Runs^2(Wins + Losses)$$

$$\cancel{Wins(Runs^2)} + Wins(Runs\ Allowed^2) \approx \cancel{Wins(Runs^2)} +\ Losses(Runs^2)$$

$$Wins(Runs\ Allowed^2) \approx\ Losses(Runs^2)\,.$$

Once more, we can divide each side by Runs Allowed ^{2} and Losses to get:

$$\frac{Wins}{Losses} \approx \frac{Runs^2}{Runs\ Allowed^2} = \left(\frac{Runs}{Runs\ Allowed}\right)^2.$$

Part B:

If we use the equation from part a, but replace 2 with P, we will have:

$$\frac{Wins}{Losses} \approx \left(\frac{Runs}{Runs\ Allowed}\right)^p.$$

Again, let's treat the approximation as we would an equal and log both sides:

$$\log\frac{Wins}{Losses} \approx \log\left(\frac{Runs}{Runs\ Allowed}\right)^p.$$

Using logarithmic property for exponentials, we can simplify the approximation to:

$$\log\frac{Wins}{Losses} \approx P\log\frac{Runs}{Runs\ Allowed}\,.$$

Since we do not want this to just be an approximation, we can add in an Error Term (ϵ) to make an equation.

$$\log\frac{Wins}{Losses} = P\log\frac{Runs}{Runs\ Allowed} + \epsilon\,.$$

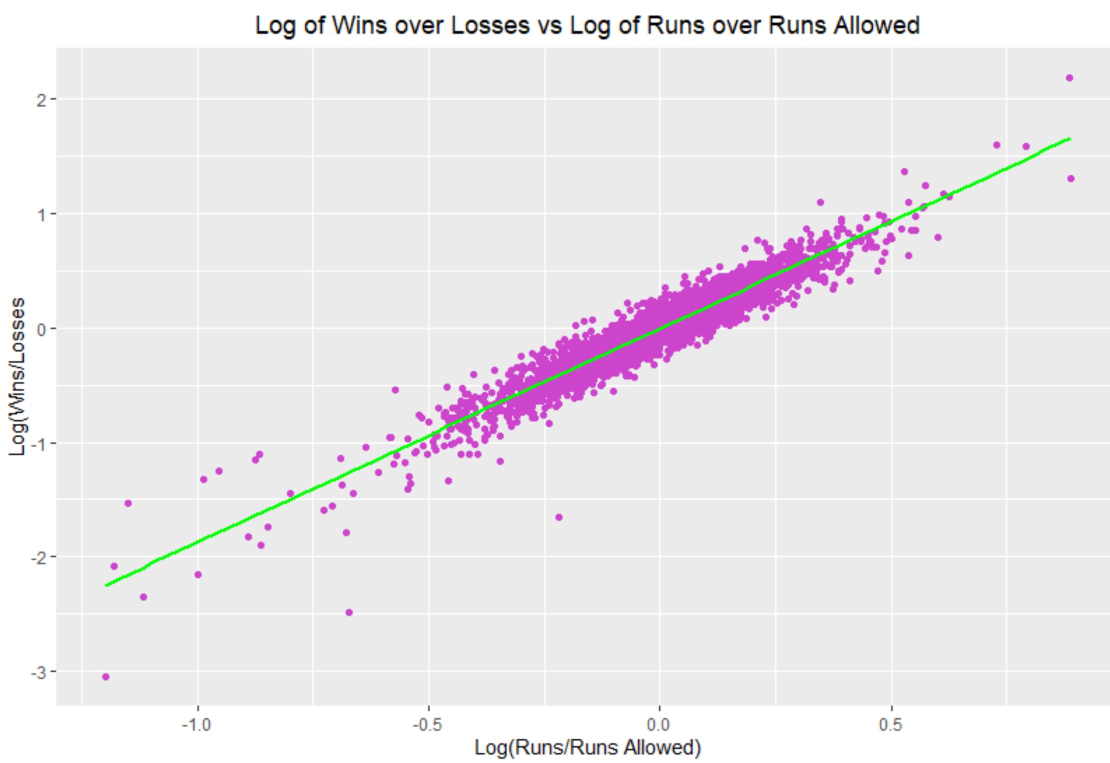
Part C:

R Code:

```
#Part 2.C: Calculating the Logs of Win/Losses and R/RA
lteamW <- filter(teamW, W > 0)

winLogF <- function(df){
  lWL <- with(df, log(W/L))
  lRRA <- with(df, log(R/RA))
  df <- cbind(df, lWL, lRRA)
}

lteamW <- winLogF(lteamW)
```

Part D:

Scatterplot:



R Code:

```
#Part 2.D: Create a Scatterplot for log(w/l) vs log(r/ra)
lWL_lRRA <- ggplot(lteamW, aes(lRRA, lWL)) +
  geom_point(color="#CC44CC") +
  geom_smooth(method = "lm", se=FALSE, color="green") +
  labs(title="Log of Wins over Losses vs Log of Runs over
       Runs Allowed",
       x="Log(Runs/Runs Allowed)",
       y="Log(Wins/Losses") +
  theme(plot.title = element_text(hjust = 0.5))
lWL_lRRA
```

Part E:

Linear Model:

```
y = -0.003 + 1.873x
R = 0.952
R-Squared = 0.906
```

R Code:

```
#Part 2.E: Create Linear Model
model2 <- lm(lWL ~ lRRA, data=lteamW)
summary(model2)


P <- round(coef(model2)[2], 3)
Eps <- round(coef(model2)[1], 3)
r2_2 <- round(summary(model2)$r.squared, 3)
r_2 <- round(sqrt(r2_2), 3)


##Display Linear Model
glue::glue("y = {Eps} + {P}x", "
           R = {r_2}", "
           R-Squared = {r2_2}")
```

3.

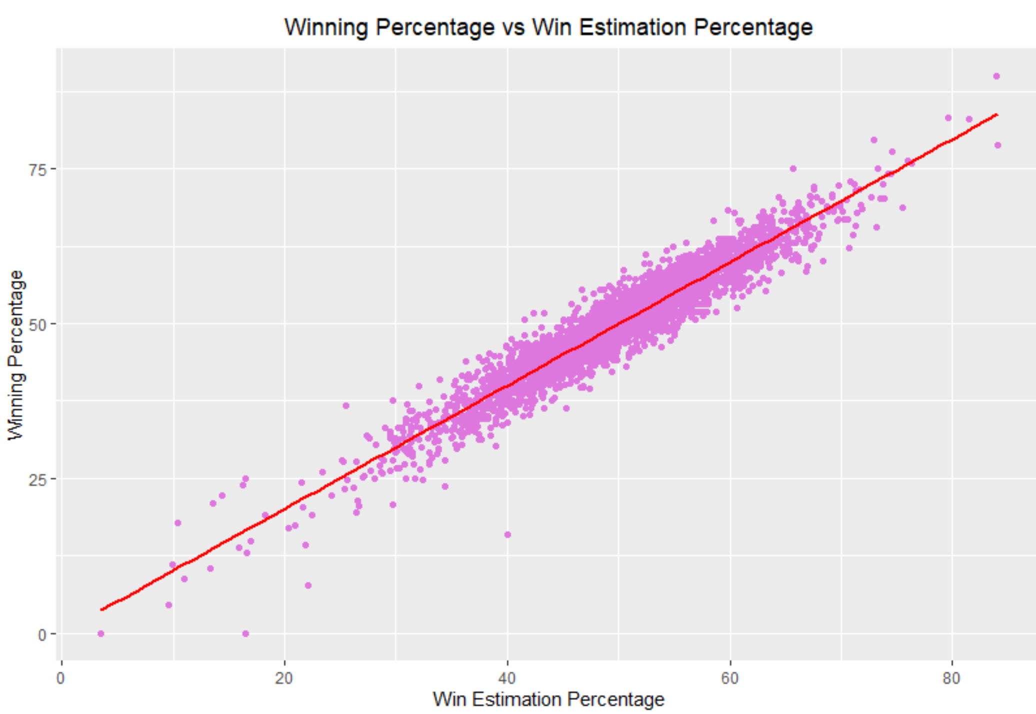
Part A and B:

R Code:

```
#Part 3.A and 3.B: Calculate the Razakean Winning Percentage

## NOTE: Since I used a subset to calculate P, and P is a
static variable, I am going to apply it to a copy of the first
subset


copyTW <- select(teamW, c(yearID, teamID, W, L, R, RA, wPer))


##Using the previous function with the new variable

copyTW <- winPythF(copyTW, P)
```

Part C:

Scatterplot:



R Code:

```r
#Part 3.C: Create Scatterplot of Winning % vs Win Estimation %
WP_WEP <- ggplot(copyTW, aes(wPyth, wPer)) +
  geom_point(color="#DD77DD") +
  geom_smooth(method = "lm", se=FALSE, color="red") +
  labs(title="Winning Percentage vs Win Estimation
       Percentage",
       x="Win Estimation Percentage",
       y="Winning Percentage") +
  theme(plot.title = element_text(hjust = 0.5))
WP_WEP
```

Part D:

Linear Model:

```
y = 0.153 + 0.996x
R = 0.955
R-Squared = 0.912
```

Question:

Like the linear model from part 1, the slope is extremely close to 1. However, unlike the intercept from part 1, the intercept is very close to 0. Since the intercept is closer to 0, this model is closer to $y = x$.

R Code:

```r
#Part 3.D: Create Linear Model
model3 <- lm(wPer ~ wPyth, data=copyTW)
summary(model3)


Pslope <- round(coef(model3)[2], 3)
inter2 <- round(coef(model3)[1], 3)
r2_3 <- round(summary(model3)$r.squared, 3)
r_3 <- round(sqrt(r2_3), 3)


glue::glue("y = {inter2} + {Pslope}x", "
           R = {r_3}", "
           R-Squared = {r2_3}")
```