14.1

For part 1, I first found which column had the missing values and where the missing values were (column 7). Then, I found the mean of that column for all values that weren't missing. Then i plugged the mean back in for all the values that were missing

```
# Identify missing values
missing_values <- which(data == "?")
num_missing <- colSums(data == "?")
col_with_mv <- names(which(num_missing>0))
print(col_with_mv)

data[,7]

# Calculate mean/mode for column with missing values
replacement_value <- mean(as.numeric(data[,7][data[,7]!="?"], na.rm = TRUE))
# Or use mode function instead of mean function for categorical data

# Replace missing values with the mean/mode value
data_no_mv <- data
data_no_mv[data_no_mv =="?"] <- replacement_value
which(data_no_mv == "?")
```

14.2

I first split the data into the rows with complete values and the rows with missing values. I trained a regression model on the rows without missing values. Then, I used that model to predict the values for the dataset with missing values. The result was that all of the predicted values were the same (3.544656). This is likely because the regression model had a poor fit and the coefficient for all of the variables was close to zero. The intercept ended up being the mean of the column with missing values.

```r
# Split the dataset into two groups: one with missing values and another without missing values
data_with_missing_values <- data[which(rowSums(data == "?") > 0), ]
data_without_missing_values <- data[which(rowSums(data == "?") ==  0), ]

#train a regression model usining non-missing data to predict missing values
model <- lm(V7 ~ V2 + V3 + V4 + V5 + V6 + V8 + V9 + V10 + V11, data = data_without_missing_values

# Use the trained model to predict missing values
predicted_values <- predict(model, newdata = data_with_missing_values[,-7])
predicted_values

summary(model)

```

38:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.2.2 · ~/

```
> predicted_values
      24       41      140      146      159      165      236      250      276      293 
3.544656 3.544656 3.544656 3.544656 3.544656 3.544656 3.544656 3.544656 3.544656 3.544656 
     295      298      316      322      412      618 
3.544656 3.544656 3.544656 3.544656 3.544656 3.544656 
> summary(model)

Call:
lm.default(formula = V7 ~ V2 + V3 + V4 + V5 + V6 + V8 + V9 + 
    V10 + V11, data = data_without_missing_values)

Residuals:
       Min         1Q     Median         3Q        Max 
-9.330e-15 -3.250e-15 -1.750e-15 -8.000e-17  1.077e-12 

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)    
(Intercept)  3.545e+00  6.446e-15  5.499e+14   <2e-16 ***
V2           8.951e-16  8.244e-16  1.086e+00    0.278    
V3          -2.165e-16  1.401e-15 -1.550e-01    0.877    
V4          -4.064e-16  1.363e-15 -2.980e-01    0.766    
V5          -1.319e-16  8.583e-16 -1.540e-01    0.878    
V6           4.034e-17  1.149e-15  3.500e-02    0.972    
V8           8.387e-16  1.109e-15  7.560e-01    0.450    
V9          -1.188e-16  8.259e-16 -1.440e-01    0.886    
V10          2.160e-17  1.086e-15  2.000e-02    0.984    
V11         -2.654e-15  3.706e-15 -7.160e-01    0.474    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.162e-14 on 673 degrees of freedom
Multiple R-squared:    0.5,	Adjusted R-squared:  0.4933 
F-statistic: 74.77 on 9 and 673 DF,  p-value: < 2.2e-16
```

14.3

For this problem, I first created the 10 perturbations and then wrote a variable that added perturbations to the non-missing data. Then, I used the trained model to predict missing values for each perturbed data set. Finally I calculated the mean, which ended up being different from the previous two steps (2.2).

```r
# Create multiple perturbations of the non-missing data by randomly adding noise to the data
num_perturbations <- 10
perturbations <- lapply(1:num_perturbations, function(i) {
  perturbed_data <- apply(data_without_missing_values, 2, as.numeric)
  perturbed_data[, -7] <- jitter(perturbed_data[, -7])
  return(perturbed_data)
})

# Use the trained model to predict missing values for each perturbed dataset
predicted_values <- lapply(perturbations, function(perturbed_data) {
  predict(model, newdata = data_with_missing_values)
})

# Calculate the mean of the predicted missing values from the perturbed datasets and use it to replace the
mean_predicted_values <- apply(predicted_values[1], 1, mean)
data[missing_values] <- mean_predicted_values

pv <- data.frame(predicted_values)
pv <- apply(pv,2,as.numeric)

mean_predicted_values <- apply(pv,1,mean)
mean_predicted_values
```

65:1 (Top Level) R Scr

Console Terminal Background Jobs

R 4.2.2 · ~/

```
 [5,]                                                                  1.271663
 [6,]                                                                  1.444743
 [7,]                                                                  1.960806
 [8,]                                                                  1.407689
 [9,]                                                                  1.625150
[10,]                                                                  6.343076
[11,]                                                                  1.219350
[12,]                                                                  1.000995
[13,]                                                                  2.005965
[14,]                                                                  1.407689
[15,]                                                                  1.200127
[16,]                                                                  1.048844
> pv <- apply(pv,2,as.numeric)
> mean_predicted_values <- apply(pv,1,mean)
> mean_predicted_values
 [1] 7.201509 3.412194 1.200127 1.588095 1.271663 1.444743 1.960806 1.407689 1.625150 6.343076
[11] 1.219350 1.000995 2.005965 1.407689 1.200127 1.048844
> mean(mean_predicted_values)
[1] 2.208626
```

15.1

In my every day life, I believe optimization would be appropriate to understand the optimal time to do yoga with highly rated yoga instructors, considering i want to do 4-5 classes per week. Some data I would need:

- Instructor rating
- My schedule and obligations

- The location of my work
- The location of my home
- Instructor class schedule
- Location of class

I would want to minimize commute time to get to the desired exercise class while still ensuring I take classes with highly rated instructors and do a minimum of 4-5 classes per week.