Answer 4

Set up:

```
library(FNN)
library(MASS)
library(flextable)
library(dplyr)

train_data <- read.csv("h1q4-train-data.csv")
test_data <- read.csv("h1q4-test-data.csv")

X_train <- train_data[,2:5]
X_test <- test_data[,2:5]

y_train <- train_data["y"]
y_test <- test_data["y"]


k = c(1,5,25)
```

Fitting 10 nearest neighbors models:

```
# Scaled Model
for (i in c(1:3)){
  Model_name <- paste("ScaledModel_", i, sep = "")
  assign(Model_name, knn.reg(train = scale(X_train), test =scale( X_test),
                             y = y_train, k = k[i]))
}

# Unscaled Model

for (i in c(1:3)){
  Model_name <- paste("UnscaledModel_", i, sep = "")
  assign(Model_name, knn.reg(train = X_train, test = X_test,
                             y = y_train, k = k[i]))
}
```

Hence we have 6 nearest neighbors models. Now we need to calculate test RMSE and train RMSE for each:

```
rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))
}

pred = function (training,predicting,k){
  knn.reg(train = training, test = predicting, y = y_train, k = k)$pred
}
```

```
# test RMSE (Unscaled)
Unscaled_model_1_train_RMSE = rmse ((pred(training= X_train,
                                          predicting = X_test,
                                          k = 1)), actual = y_test)
Unscaled_model_2_train_RMSE = rmse ((pred(training= X_train,
                                          predicting = X_test, k = 5)),
                                    actual = y_test)
Unscaled_model_3_train_RMSE = rmse ((pred(training= X_train,
                                          predicting = X_test, k = 25)),
                                    actual = y_test)

# test RMSE (scaled)
scaled_One = rmse (predicted= pred(training= scale(X_train),
                                         predicting = scale(X_test), k = 1),
                                   actual = y_test)
scaled_One
```

```
## [1] NA
```

```
scaled_model_2_train_RMSE = rmse ((pred(training= scale(X_train),
                                         predicting = scale(X_test),
                                         k = 5)), actual = y_test)
scaled_model_3_train_RMSE = rmse ((pred(training= scale(X_train),
                                         predicting = scale(X_test), k = 25)),
                                   actual = y_test)
```

Now to tabulate our findings and conclusion:

```
library(flextable)
table <- data.frame(
  kValue = c(1,1,5,5,25,25),
  model_test_RMSE = c(Unscaled_model_1_train_RMSE,scaled_One,
                      Unscaled_model_2_train_RMSE,scaled_model_2_train_RMSE,
                      Unscaled_model_3_train_RMSE,scaled_model_3_train_RMSE),
  scaleCheck = c ("No","Yes","No","Yes","No","Yes"))



table
```

```
##   kValue model_test_RMSE scaleCheck
## 1      1              NA         No
## 2      1              NA        Yes
## 3      5              NA         No
## 4      5              NA        Yes
## 5     25              NA         No
## 6     25              NA        Yes
```

```
t<-delete_part(flextable(table), part = "header")
t<- add_header(t,top=T,kValue="k",model_test_RMSE= 'test RMSE',
               scaleCheck="Scaled?")



m<- colformat_num(t,j=c(1),digits = 0)
```

```r
j<- colformat_num(m,j=c(2),digits = 2)


#autofit(align(j,align = "center", part = "all"))
```