# exer1

Christopher Huong

2023-06-11

```
library(rio)
library(tidyverse)
library(caret)
library(ggplot2)
library(gridExtra)
library(ggpubr)
library(earth)
```

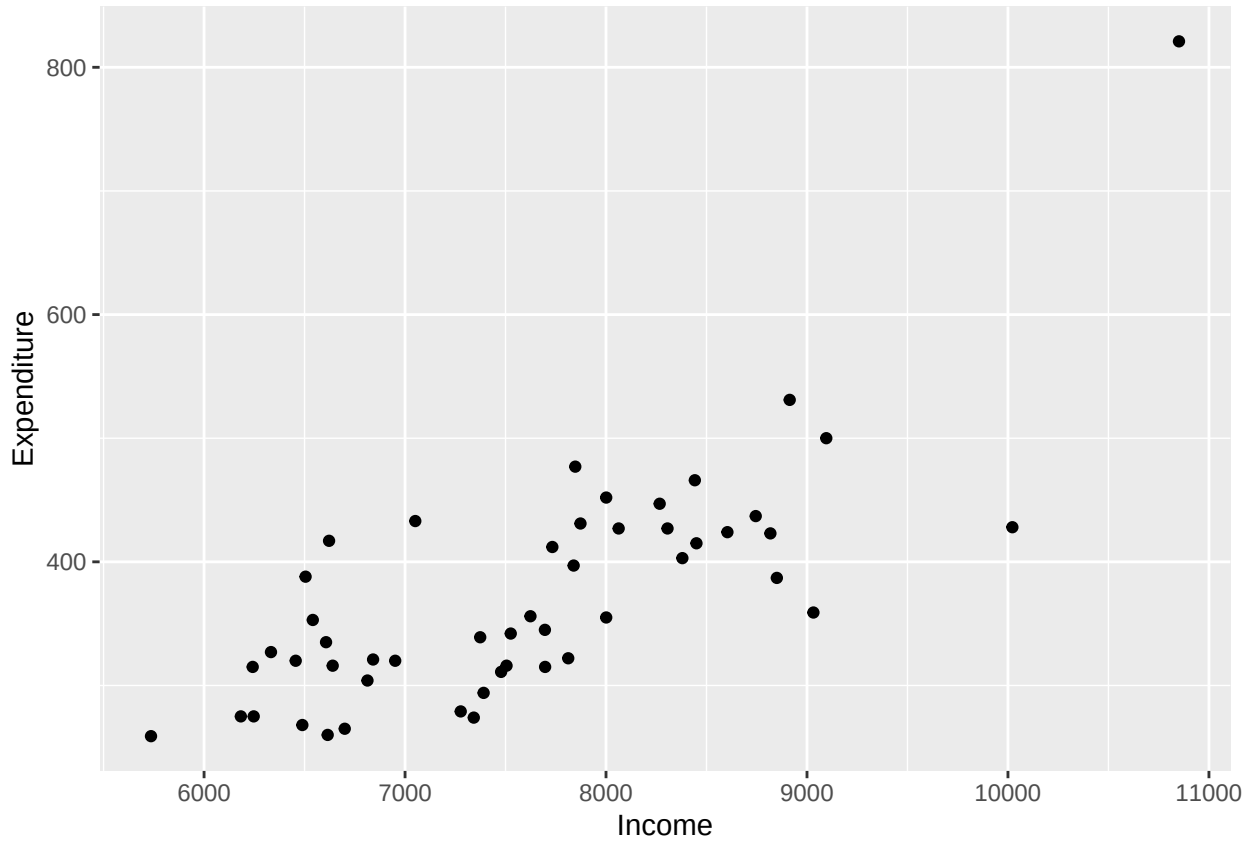
Due 11:59 PM CT 06/11/2023

Consider a well-known dataset on per capita income and per capita spending in public schools by state in the United States in 1979. (Available on blackboard). This dataset has been widely analyzed in various statistical. As in those previous analyses, we take per capita spending (Expenditure) as the dependent variable and per capita income as the predictor variable.

```
dat <- import('ex1.csv')



dat <- dat[order(dat$Income),]
```

a) Draw a scatter-plot to check the relationship between Income and Expenditure and interpret the relationship between Income and Expenditure.

```
ggplot(dat, aes(x=Income, y=Expenditure)) +
  geom_point()
```




# There is a positive linear relationship between state income and expenditure

b) Find and interpret the slope for the least squares regression line

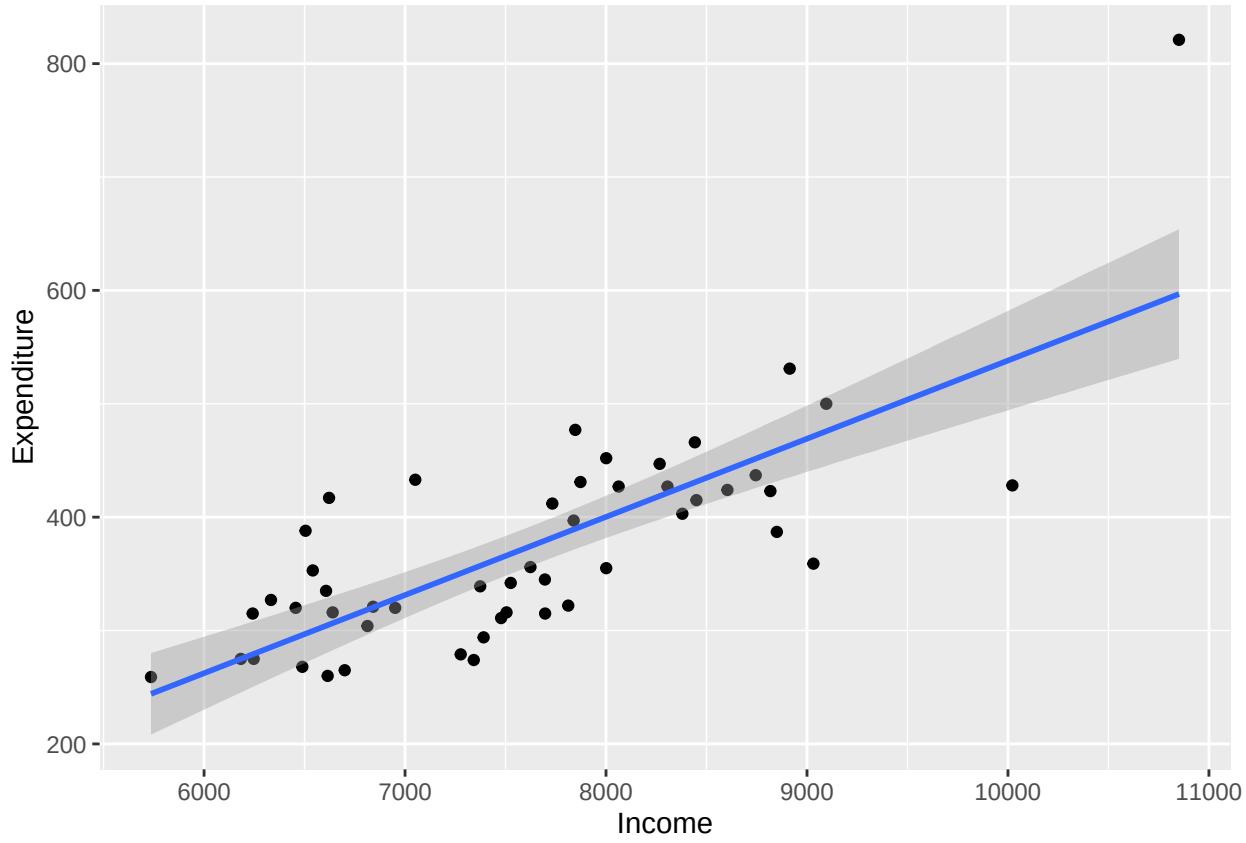
```
lm1 <- train(Expenditure ~ Income,
             data = dat,
             method = "lm",
             trControl = trainControl(method="cv"))
summary(lm1)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -112.390  -42.146   -6.162   30.630  224.210
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -151.26509   64.12183  -2.359   0.0224 *
## Income         0.06894    0.00835   8.256 9.05e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.41 on 48 degrees of freedom
```

```
## Multiple R-squared:  0.5868, Adjusted R-squared:  0.5782
## F-statistic: 68.16 on 1 and 48 DF,  p-value: 9.055e-11
```

```
ggplot(dat, aes(x=Income, y=Expenditure)) +
  geom_point()+
  geom_smooth(method=lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




**The slope for the least squares regression line is 0.069. This is interpreted as for every unit increase in income, you can expect a 0.069 unit increase in expenditure.**

c) Find and interpret y-intercept for the least squares regression line

```
summary(lm1)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -112.390  -42.146   -6.162   30.630  224.210
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -151.26509   64.12183  -2.359   0.0224 *
## Income         0.06894    0.00835   8.256 9.05e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.41 on 48 degrees of freedom
## Multiple R-squared:  0.5868, Adjusted R-squared:  0.5782
## F-statistic: 68.16 on 1 and 48 DF,  p-value: 9.055e-11
```

**The y-intercept for the least squares regression line is -151.265. This is interpreted as when income is at 0 units, expenditure is expected to be -151.265 units.**

d) Find the least square regression equation and circle the results from your outputs.

**y = -151.265 + x*0.0689 + e**

e) Find proportion of the variation that can be explained by the least squares regression line (i.e., R2).

```
lm1
```

```
## Linear Regression
##
## 50 samples
##  1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 46, 45, 43, 46, 46, 45, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   56.81128  0.6899715  46.53558
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

**R^2 = 0.6388, meaning 63.88% of the variance of expenditure can be explained by regressing expenditure on income**

f) Find the estimator of o2 (i.e., s2) and interpret the value of this estimator.

```r
sum((dat$Expenditure-mean(dat$Expenditure))^2)/(nrow(dat)-1)
```

```
## [1] 8940.319
```

```r
var(dat$Expenditure)
```

```
## [1] 8940.319
```

**The variance is the square of the standard deviation, interpreted as the square of the average distance each value of y is from it's mean.**

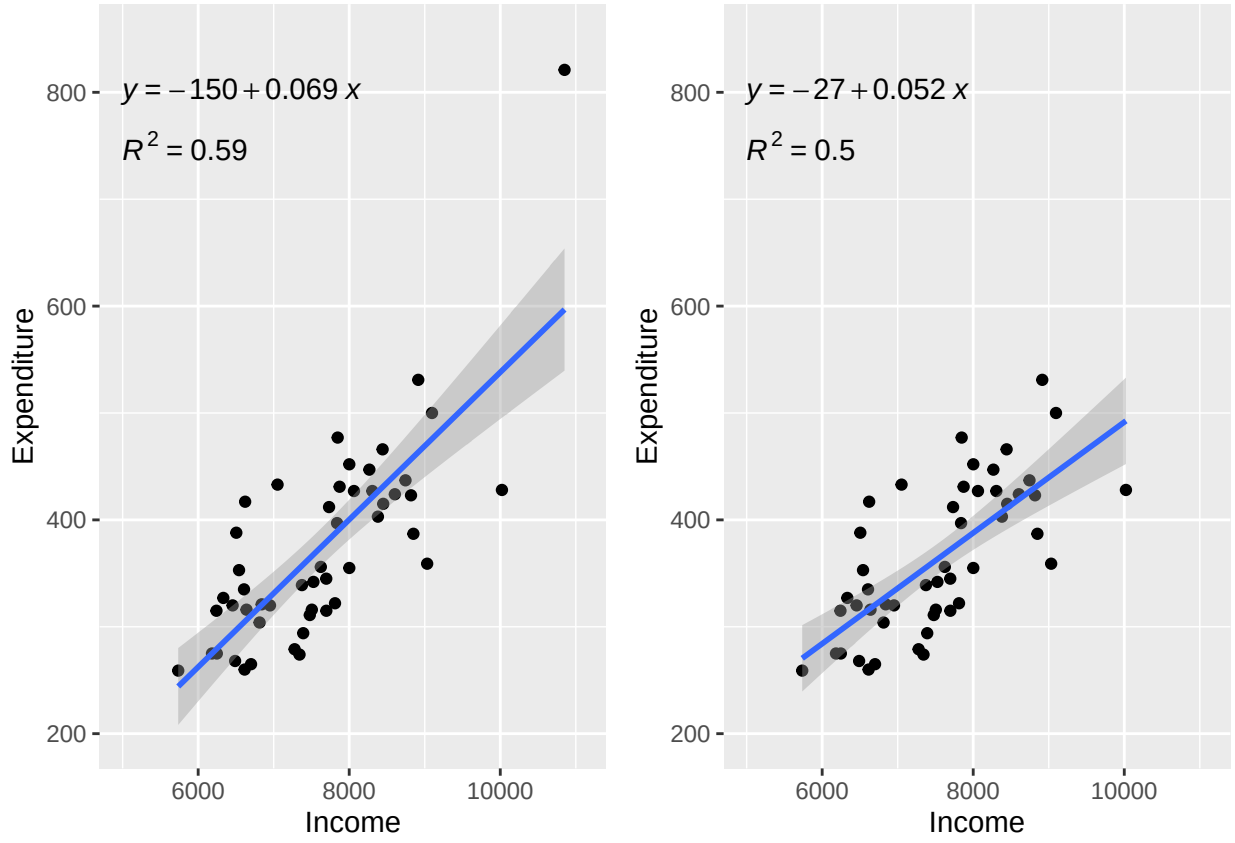
g) Check if the data contain any outlier or influential points?

**AK is an outlier with y=821 and x=10851. Removing AK from the data set yields:**

```r
a <- ggplot(dat, aes(x=Income, y=Expenditure)) +
       geom_point()+
       geom_smooth(method=lm)

b <-ggplot(filter(dat, State!='AK'), aes(x=Income, y=Expenditure)) +
       geom_point()+
       geom_smooth(method=lm)



a <- ggplot(dat,aes(x = Income, y = Expenditure)) +
  geom_point() +
  geom_smooth(method = "lm", se=T) +
  xlim(5000,11100) + ylim(200, 850) +
  stat_regline_equation(label.y = 800, aes(label = ..eq.label..)) +
  stat_regline_equation(label.y = 750, aes(label = ..rr.label..))

b <- ggplot(filter(dat, State!='AK'),aes(x = Income, y = Expenditure)) +
  geom_point() +
  geom_smooth(method = "lm", se=T) +
  xlim(5000,11100) + ylim(200, 850) +
  stat_regline_equation(label.y = 800, aes(label = ..eq.label..)) +
  stat_regline_equation(label.y = 750, aes(label = ..rr.label..))

grid.arrange(a, b, ncol=2)
```

```
## Warning: The dot-dot notation (`..eq.label..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(eq.label)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




# Removing AK reduces the slope by 100*(1-(0.052/0.069)) = 24.6%

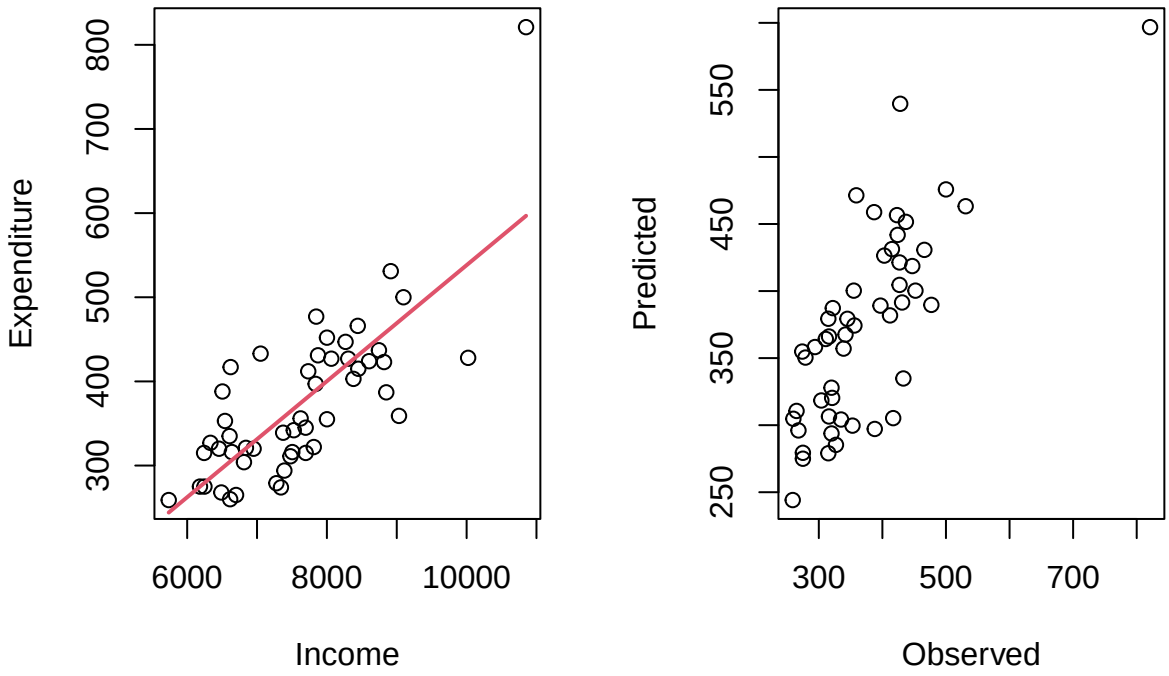
h) Fit a single linear model and conduct 10-fold CV to estimate the error. In addition, draw the scatter plot with the fitted line and the scatter plot between the observed and fitted values below.

```
lm <- train(Expenditure ~ Income,
            data = dat,
            method = "lm",
            trControl = trainControl(method = "cv"))

summary(lm)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -112.390  -42.146   -6.162   30.630  224.210
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -151.26509   64.12183  -2.359   0.0224 *
## Income         0.06894    0.00835   8.256 9.05e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.41 on 48 degrees of freedom
## Multiple R-squared:  0.5868, Adjusted R-squared:  0.5782
## F-statistic: 68.16 on 1 and 48 DF,  p-value: 9.055e-11
```

```
par(mfrow=c(1,2))
plot(dat$Income, dat$Expenditure, xlab = "Income", ylab= "Expenditure")
lines(dat$Income, fitted(lm), col=2, lwd=2)


Observed = dat$Expenditure
Predicted = fitted(lm)
plot(Observed, Predicted)
```



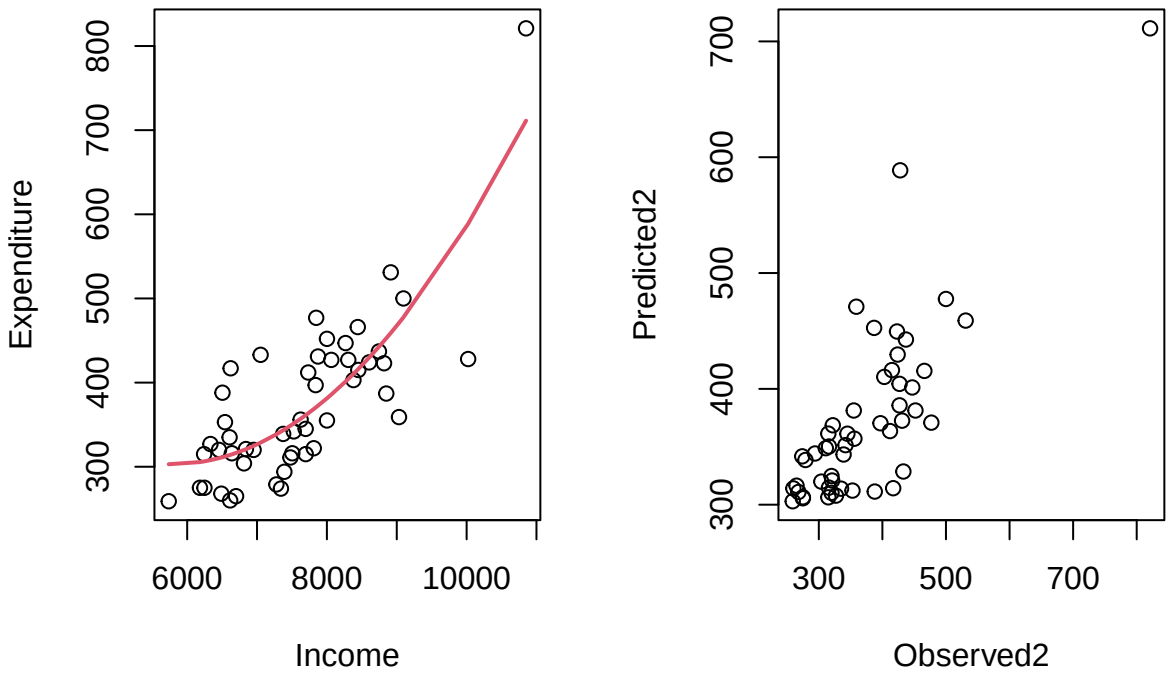

i) Fit a quadratic model and conduct 10-fold CV to estimate the error and draw the scatter plot with the fitted line and the scatter plot between the observed and fitted values.

```
dat$Income2 <- dat$Income^2


lm2 <- train(Expenditure ~ Income + Income2,
```

```
             data = dat,
             method = "lm",
             trControl = trainControl(method = "cv"))

summary(lm2)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -160.709  -36.896   -4.551   37.290  109.729
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.329e+02  3.273e+02   2.545  0.01428 *
## Income      -1.834e-01  8.290e-02  -2.213  0.03182 *
## Income2      1.587e-05  5.191e-06   3.057  0.00368 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56.68 on 47 degrees of freedom
## Multiple R-squared:  0.6553, Adjusted R-squared:  0.6407
## F-statistic: 44.68 on 2 and 47 DF,  p-value: 1.345e-11
```

```
par(mfrow=c(1,2))
plot(dat$Income, dat$Expenditure, xlab = "Income", ylab= "Expenditure")
lines(dat$Income, fitted(lm2), col=2, lwd=2)



Observed2 = dat$Expenditure
Predicted2 = fitted(lm2)
plot(Observed2, Predicted2)
```




j) Fit a mars model with optimal tuning parameters that you choose and conduct 10-fold CV to estimate the error and draw the scatter plot with the fitted line and the scatter plot between the observed and fitted values.

```
marsfit <- train(Expenditure ~ Income,
                 data = dat,
                 method = "earth",
                 tuneLength = 15,
                 trControl = trainControl(method="cv"))

marsfit
```

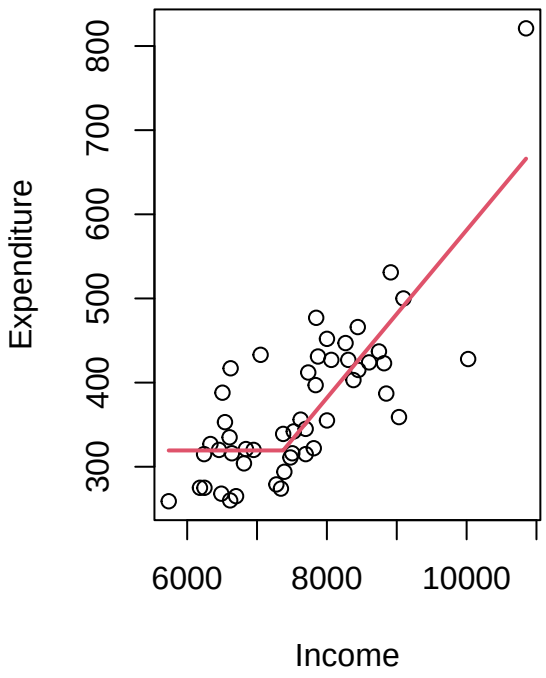
```
## Multivariate Adaptive Regression Spline
##
## 50 samples
##  1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 45, 45, 45, 46, 45, 45, ...
## Resampling results across tuning parameters:
##
##   nprune  RMSE      Rsquared   MAE
##   2       58.69657  0.6305127  45.89169
##   3       64.15931  0.5480277  49.21448
```

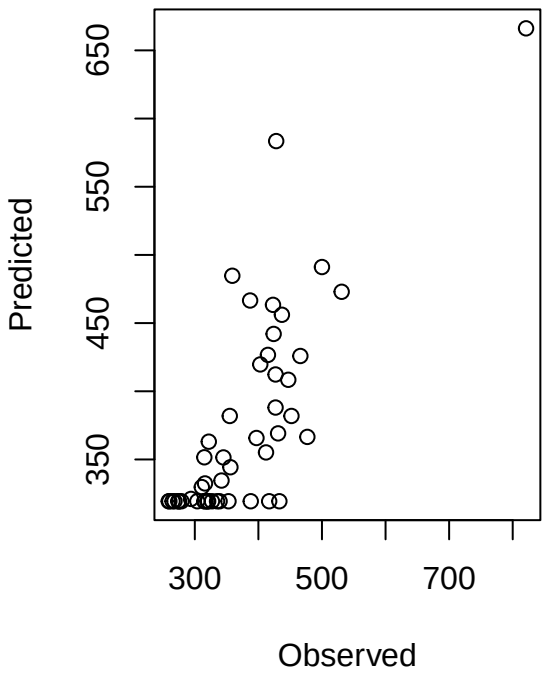
```
##   4       65.91297  0.5360764  50.72710
##   5       65.91297  0.5360764  50.72710
##   6       65.91297  0.5360764  50.72710
##
## Tuning parameter 'degree' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nprune = 2 and degree = 1.
```

```
summary(marsfit)
```

```
## Call: earth(x=c(5736,6183,624...), y=c(259,275,315,2...), keepxy=TRUE,
##             degree=1, nprune=2)
##
##                coefficients
## (Intercept)    319.37023087
## h(Income-7374)   0.09973307
##
## Selected 2 of 6 terms, and 1 of 1 predictors (nprune=2)
## Termination condition: RSq changed by less than 0.001 at 6 terms
## Importance: Income
## Number of terms at each degree of interaction: 1 1 (additive model)
## GCV 3653.606    RSS 161416.3    GRSq 0.5995071    RSq 0.6315332
```

```
par(mfrow=c(1,2))
plot(dat$Income, dat$Expenditure, xlab = "Income", ylab= "Expenditure")
lines(dat$Income, fitted(marsfit), col=2, lwd=2)



Observed = dat$Expenditure
Predicted = fitted(marsfit)
plot(Observed, Predicted)
```




k) Compare the three fitted models in terms of RMSE and R2, and then make a recommendation based on your criteria.

```
dat$lm1 <- predict(lm1, dat)
dat$lm2 <- predict(lm2, dat)
dat$marsfit <- predict(marsfit, dat)


postResample(pred = dat$lm1, obs = dat$Expenditure)
```

```
##       RMSE   Rsquared        MAE
## 60.1689650  0.5867946 45.5229231
```

```
postResample(pred = dat$lm2, obs = dat$Expenditure)
```

```
##       RMSE   Rsquared        MAE
## 54.9518887  0.6553437 42.3806732
```

```
postResample(pred = dat$marsfit, obs = dat$Expenditure)
```

```
##       RMSE   Rsquared        MAE
## 56.8183623  0.6315332 42.5209388
```

**Based on the best model fit (lowest RMSEA) and variance accounted for (highest R2), the quadratic model is recommended.**