STAT408_HW5

2022-11-21

Question 1

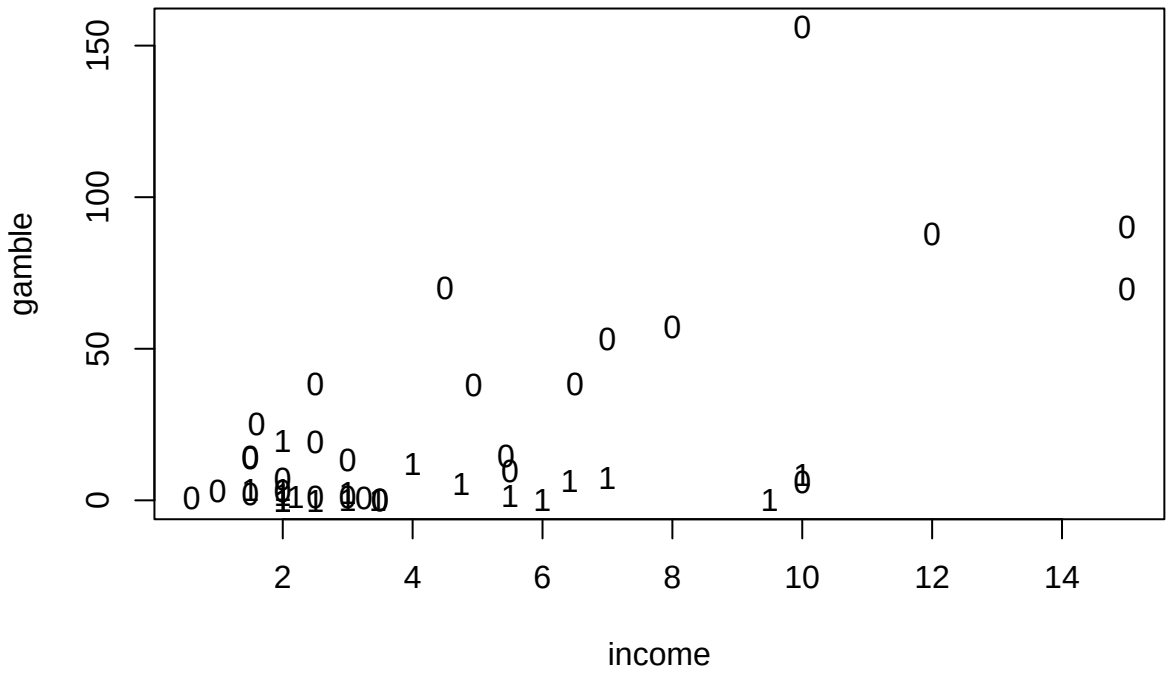
(15 points) Let's revisit the teengamb dataset in this question.

```
teengamb <- read.csv("teengamb.csv")
```

part (a)

Make a plot of gamble on income using a different plotting symbol depending on the sex.

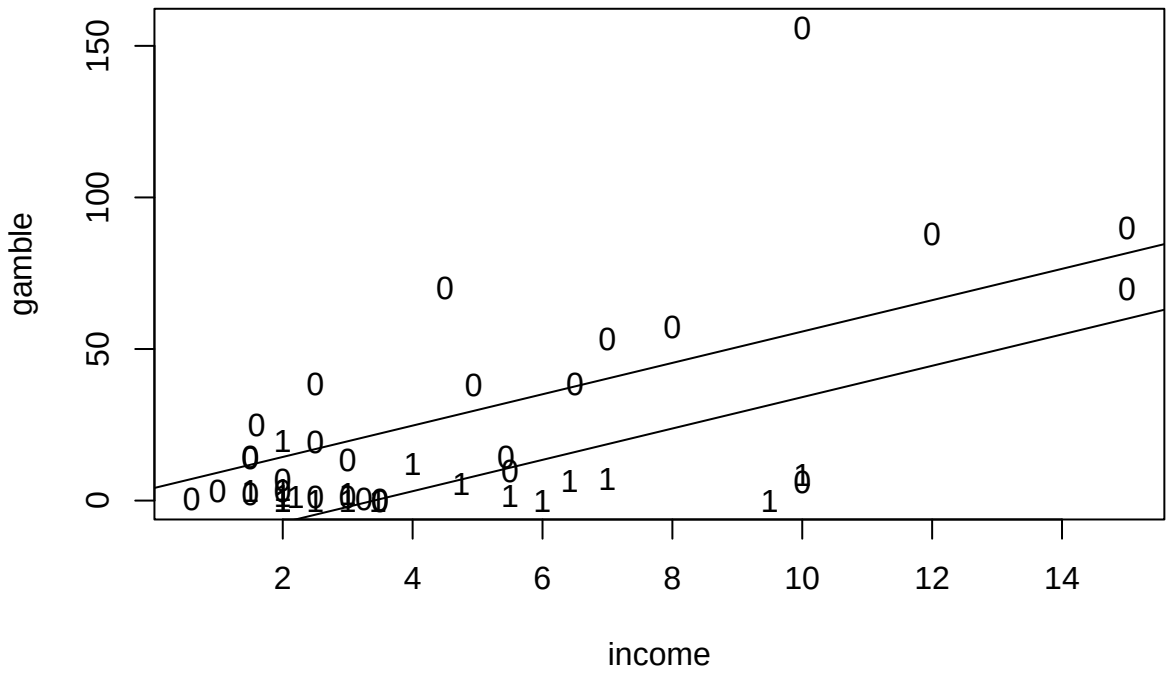
```
plot(gamble~income,pch=as.character(sex),data = teengamb)
```



part (b)

Fit a regression model with gamble as the response and income and sex as predictors. Display the regression fit with sex = 0 and sex = 1 separately on the plot. (Hint: use abline function)

```
fit <- lm(gamble ~ income + sex, data = teengamb)
plot(gamble~income,pch=as.character(sex),data = teengamb)
abline(fit$coef[1] + fit$coef[3], fit$coef[2])
abline(fit$coef[1], fit$coef[2])
```



part (c)

Use the Matching package to find matches on sex by treating income as the confounder. Use the same parameters as in the lecture slides. How many matched pairs were found? How many cases were not matched?

```
library(Matching)
```

```
## Loading required package: MASS
```

```
## ##
## ##  Matching (Version 4.10-8, Build Date: 2022-11-03)
## ##  See http://sekhon.berkeley.edu/matching for additional documentation.
## ##  Please cite software as:
```

```
## ##   Jasjeet S. Sekhon. 2011. ‘‘Multivariate and Propensity Score Matching
## ##   Software with Automated Balance Optimization: The Matching package for R.’’
## ##   Journal of Statistical Software, 42(7): 1-52.
## ##
```

```
mm <- GenMatch(teengamb$sex, teengamb$income, ties=FALSE)
```

```
## Loading required namespace: rgenoud
```

```
## Warning in GenMatch(teengamb$sex, teengamb$income, ties = FALSE): The key
## tuning parameters for optimization were are all left at their default values.
## The 'pop.size' option in particular should probably be increased for optimal
## results. For details please see the help page and http://sekhon.berkeley.edu/
## papers/MatchingJSS.pdf
```

```
##
##
## Tue Nov 22 16:01:33 2022
## Domains:
##  0.000000e+00   <=  X1   <=    1.000000e+03
##
## Data Type: Floating Point
## Operators (code number, name, population)
##  (1) Cloning.....................  15
##  (2) Uniform Mutation............  12
##  (3) Boundary Mutation...........  12
##  (4) Non-Uniform Mutation........  12
##  (5) Polytope Crossover..........  12
##  (6) Simple Crossover............  12
##  (7) Whole Non-Uniform Mutation..  12
##  (8) Heuristic Crossover.........  12
##  (9) Local-Minimum Crossover.....  0
##
## SOFT Maximum Number of Generations: 100
## Maximum Nonchanging Generations: 4
## Population size       : 100
## Convergence Tolerance: 1.000000e-03
##
## Not Using the BFGS Derivative Based Optimizer on the Best Individual Each Generation.
## Not Checking Gradients before Stopping.
## Using Out of Bounds Individuals.
##
## Maximization Problem.
## GENERATION: 0 (initializing the population)
## Lexical Fit..... 6.648968e-01  1.000000e+00
## #unique..... 100, #Total UniqueCount: 100
## var 1:
## best..... 8.280656e+01
## mean..... 4.598687e+02
## variance..... 8.169399e+04
##
## GENERATION: 1
## Lexical Fit..... 6.648968e-01  1.000000e+00
```

```
## #unique......... 57, #Total UniqueCount: 157
## var 1:
## best............ 8.280656e+01
## mean............ 3.889773e+02
## variance........ 7.585482e+04
##
## GENERATION: 2
## Lexical Fit..... 6.648968e-01  1.000000e+00
## #unique......... 58, #Total UniqueCount: 215
## var 1:
## best............ 8.280656e+01
## mean............ 4.159265e+02
## variance........ 7.460439e+04
##
## GENERATION: 3
## Lexical Fit..... 6.648968e-01  1.000000e+00
## #unique......... 57, #Total UniqueCount: 272
## var 1:
## best............ 8.280656e+01
## mean............ 3.524477e+02
## variance........ 7.396039e+04
##
## GENERATION: 4
## Lexical Fit..... 6.648968e-01  1.000000e+00
## #unique......... 54, #Total UniqueCount: 326
## var 1:
## best............ 8.280656e+01
## mean............ 3.521424e+02
## variance........ 7.837408e+04
##
## GENERATION: 5
## Lexical Fit..... 6.648968e-01  1.000000e+00
## #unique......... 56, #Total UniqueCount: 382
## var 1:
## best............ 8.280656e+01
## mean............ 3.616704e+02
## variance........ 7.968338e+04
##
## 'wait.generations' limit reached.
## No significant improvement in 4 generations.
##
## Solution Lexical Fitness Value:
## 6.648968e-01  1.000000e+00
##
## Parameters at the Solution:
##
##  X[ 1] : 8.280656e+01
##
## Solution Found Generation 1
## Number of Generations Run 5
##
## Tue Nov 22 16:01:33 2022
## Total run time : 0 hours 0 minutes and 0 seconds
```

```
match <- mm$matches[,1:2]
match
```

```
##       [,1] [,2]
##  [1,]    1   41
##  [2,]    2   37
##  [3,]    3   34
##  [4,]    4   32
##  [5,]    5   41
##  [6,]    6   23
##  [7,]    7   30
##  [8,]    8   25
##  [9,]    9   34
## [10,]   10   30
## [11,]   11   21
## [12,]   12   45
## [13,]   13   41
## [14,]   14   34
## [15,]   15   43
## [16,]   16   35
## [17,]   17   24
## [18,]   18   39
## [19,]   19   20
```

19 matched pairs were found while 28 cases were not matched.

part (d)

Compute the differences in gamble for the matched pairs. Is there a significant non-zero difference using one-sample t-test?

```
pdiff <- teengamb$gamble[match[,1]] - teengamb$gamble[match[,2]]
t.test(pdiff)
```

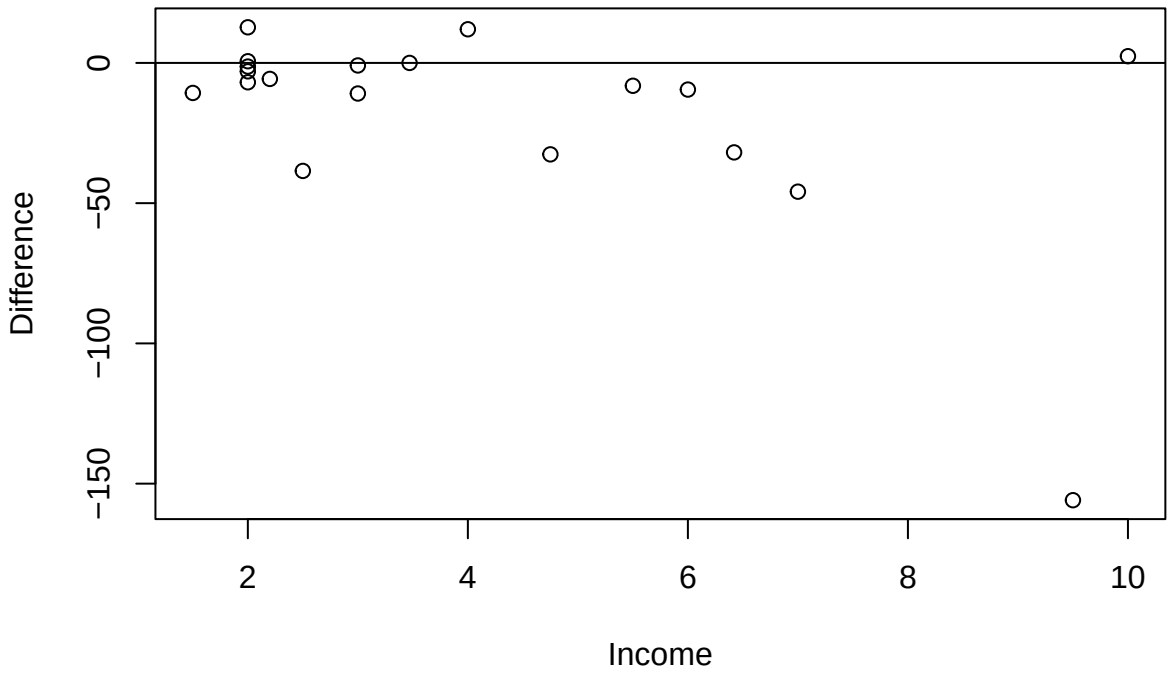
```
##
##  One Sample t-test
##
## data:  pdiff
## t = -2.0615, df = 18, p-value = 0.054
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -35.5099940   0.3363098
## sample estimates:
## mean of x
## -17.58684
```

There appears to be a significant difference in gambling among the matched pairs, because the p-value of 0.01644 is less than a significance level of $alpha = 0.05$ and the 95 percent confidence interval includes only negative values and does not include zero.

part (e)

Plot the difference in gamble against income. In what proportion of pairs did the female gamble more than the male?

```
plot(pdiff ~ teengamb$income[match[,1]], xlab="Income", ylab="Difference")
abline(h=0)
```



Only a small proportion of pairs had the female gambling more than the male as many of the points fall below zero.

part (f)

Do the conclusions from the linear model and the matched pair approach agree? Give you interpretation and insight.

Yes, both the linear model and the matched pair approach show that males tend to gamble more than females.

Question 2

(15 points) The infmort dataset records the infant mortality of 105 countries with their income, region, and oil export information. The infant mortality in regions of the world may be related to per capita income and whether oil is exported.

```
library(faraway)
```

```
##
## Attaching package: 'faraway'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     teengamb
```

```
data(infmort)
```

part (a)

Which variables are continuous? Which are categorical variables? How many levels the categorical variable have?

```
head(infmort)
```

```
##                       region income mortality           oil
## Australia               Asia   3426      26.7 no oil exports
## Austria               Europe   3350      23.7 no oil exports
## Belgium               Europe   3346      17.0 no oil exports
## Canada              Americas   4751      16.8 no oil exports
## Denmark               Europe   5029      13.5 no oil exports
## Finland               Europe   3312      10.1 no oil exports
```

```
unique(infmort$region)
```

```
## [1] Asia     Europe   Americas Africa
## Levels: Africa Europe Asia Americas
```

```
unique(infmort$oil)
```

```
## [1] no oil exports oil exports
## Levels: oil exports no oil exports
```

Region and oil export information are both categorical variables, with region having four levels and oil exports having two level. Income and mortality, on the other hand, are continuous variables.

part (b)

Regress mortality on all other variables. Interpret the model output and the meaning of estimated parameters.

```
fit2 <- lm(mortality ~ ., data = infmort)
summary(fit2)
```

```
##
## Call:
## lm(formula = mortality ~ ., data = infmort)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -156.00  -32.20   -4.44   13.65  488.82
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.152e+02  2.974e+01   7.234 1.19e-10 ***
## regionEurope      -1.015e+02  3.073e+01  -3.303 0.001351 **
## regionAsia        -4.589e+01  2.014e+01  -2.278 0.024977 *
## regionAmericas    -8.365e+01  2.180e+01  -3.837 0.000224 ***
## income            -5.290e-03  7.404e-03  -0.714 0.476685
## oilno oil exports -7.834e+01  2.891e+01  -2.710 0.007992 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 77.36 on 95 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.3105, Adjusted R-squared:  0.2742
## F-statistic: 8.556 on 5 and 95 DF,  p-value: 1.015e-06
```

This model shows that region and oil are significant in predicting mortality at the level $alpha = 0.05$ while income is not. Mortality is expected to decrease by approximately 7.834e+01 in regions with no oil exports compared to those with oil exports. Additionally, mortality is lowest in the Americas (decreases by approximately 8.365e+01) and highest in Europe (decreases by only 1.015e+02).

part (c)

Regress mortality on income, region, oil, the interaction between income and region, and the interaction between income and oil. Compare this model with the one in (b). Interpret the estimated parameters

```
fit3 <- lm(mortality ~ income + region + oil + income*region + income*oil, data = infmort)
summary(fit3)
```

```
##
## Call:
## lm(formula = mortality ~ income + region + oil + income * region +
##     income * oil, data = infmort)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -218.172  -25.264   -3.993   14.988  304.041
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                77.46084   35.35672   2.191 0.031018 *
## income                      0.08484    0.02616   3.243 0.001656 **
## regionEurope             -135.53952   39.94761  -3.393 0.001025 **
## regionAsia                -72.88297   20.98476  -3.473 0.000789 ***
```

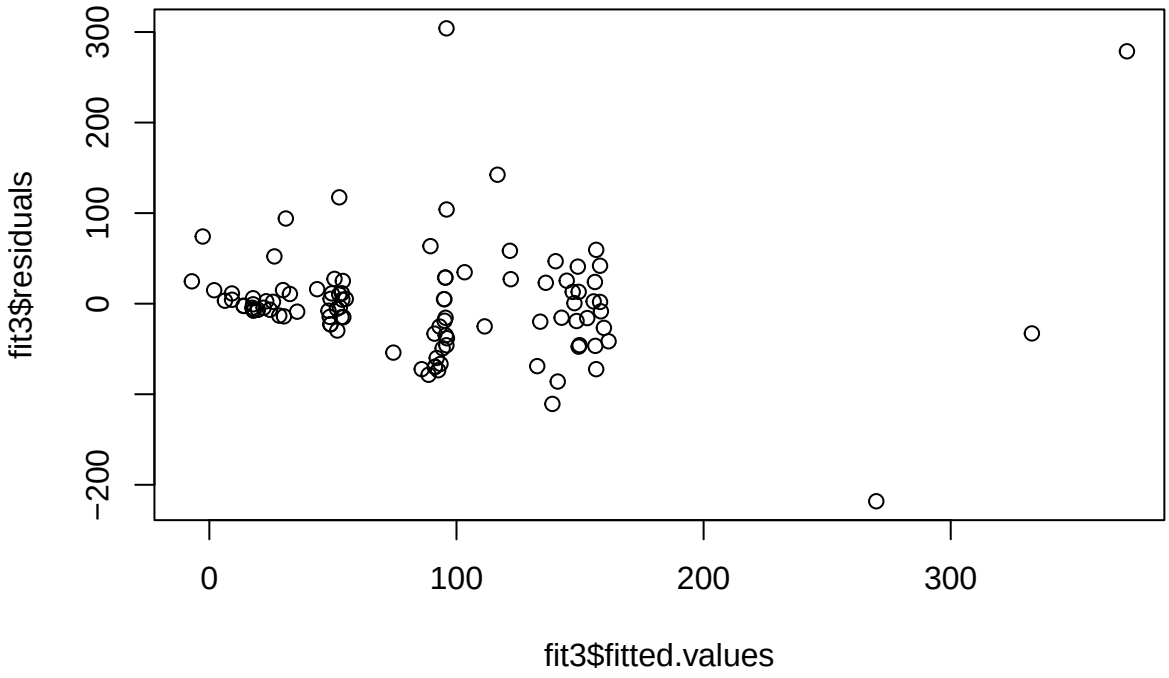
```
## regionAmericas           -112.65044   22.73665  -4.955 3.33e-06 ***
## oilno oil exports          92.73318   37.12260   2.498 0.014285 *
## income:regionEurope         0.16781    0.04090   4.103 8.89e-05 ***
## income:regionAsia           0.15485    0.04009   3.862 0.000210 ***
## income:regionAmericas       0.16117    0.04044   3.986 0.000136 ***
## income:oilno oil exports   -0.25772    0.04185  -6.158 1.97e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.09 on 91 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.5179, Adjusted R-squared:  0.4702
## F-statistic: 10.86 on 9 and 91 DF,  p-value: 2.664e-11
```

This model shows that income appears to be significant at the $alpha = 0.05$ level, even though that was not the case for the model in part (b). Additionally, all of the included parameters appear to be significant. Mortality is expected to increase by 0.08484 for each one unit increase in income. Mortality is expected to decrease by 135.53952 in Europe, 72.88297 in Asia, 112.65044 in the Americas, and it is expected to increase by 92.73318 when there are no oil exports. Mortality increase by an additional 0.16781 for each one unit increase in income when in Europe, 0.15485 for each one unit increase in income when in Asia, 0.16117 for each one unit increase in income when in the Americas, and decrease by 0.25772 for every one unit increase in income when there are no oil exports.

part (d)

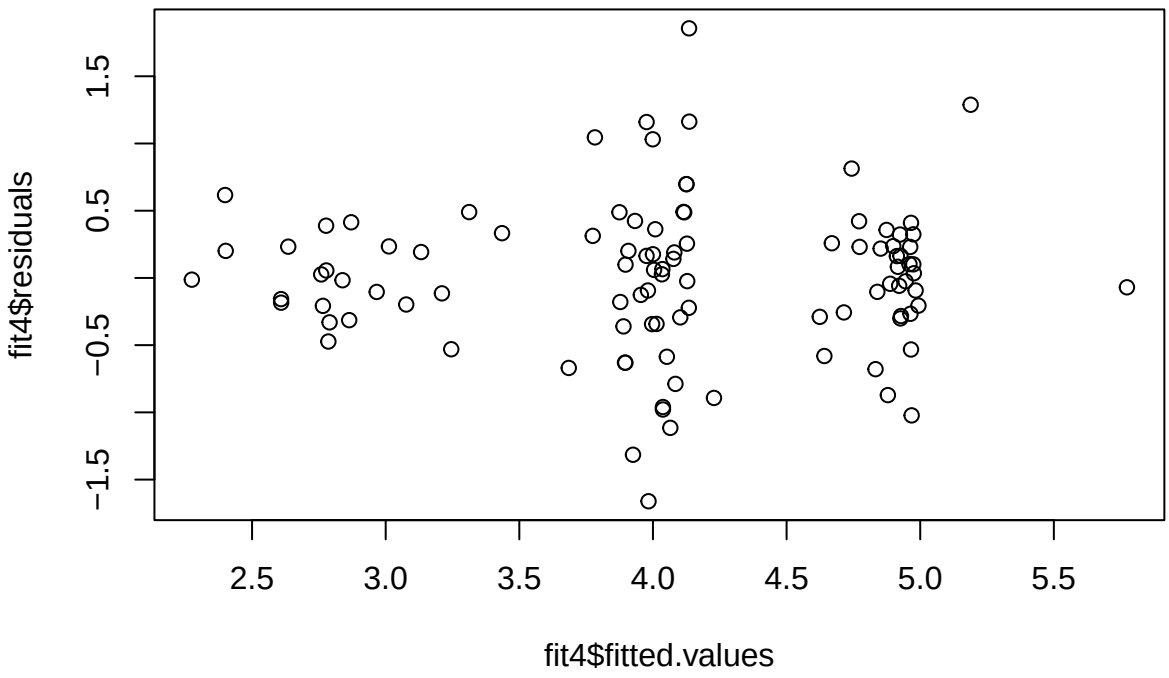
Does the model in (c) satisfy the constant variance assumption? If not, give a transformation and refit the model. Check if the transformation solves the issue.

```
plot(fit3$fitted.values, fit3$residuals)
```



This model does not appear to meet the assumption of constant variance because the plot of the residuals versus fitted values shows a bit of a megaphone shape.

```
fit4 <- lm(log(mortality) ~ income + region + oil + income*region + income*oil, data = infmort)
plot(fit4$fitted.values, fit4$residuals)
```



The log transformation of mortality does appear to solve the issue of the violation of the contant variance assumption as the plot no longer shows a megaphone shape.

part (e)

Interpret the estimated parameters in (d) for region and oil variables.

```
summary(fit4)
```

```
##
## Call:
## lm(formula = log(mortality) ~ income + region + oil + income *
##     region + income * oil, data = infmort)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66101 -0.28951  0.02687  0.25788  1.85641
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               4.5520271  0.3104348  14.663  < 2e-16 ***
## income                    0.0004058  0.0002297   1.767 0.080648 .
## regionEurope             -1.5154005  0.3507433  -4.321 3.96e-05 ***
## regionAsia               -0.8781276  0.1842478  -4.766 7.09e-06 ***
## regionAmericas           -0.9534513  0.1996296  -4.776 6.81e-06 ***
```

```
## oilno oil exports         0.4894459  0.3259394   1.502 0.136650
## income:regionEurope       0.0007379  0.0003591   2.055 0.042792 *
## income:regionAsia         0.0005842  0.0003520   1.660 0.100416
## income:regionAmericas     0.0006985  0.0003551   1.967 0.052208 .
## income:oilno oil exports -0.0013673  0.0003675  -3.721 0.000343 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5803 on 91 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.6732, Adjusted R-squared:  0.6409
## F-statistic: 20.83 on 9 and 91 DF,  p-value: < 2.2e-16
```

Mortality is expected to decrease by exp(1.5154005) in Europe, exp(0.8781276) in Asia, exp(0.9534513) in the Americas, and increase by exp(0.4894459) if there are no oil exports.

Question 4

(10 points) In this question, you will use all predictors in births dataset to predict the baby's birth weight.

```
births <- read.csv("births.csv")
```

part (a)

Randomly split the whole dataset into 80% training and 20% test set. Train a linear model with all predictors using training set. Use this model to predict the weight in the test set. Calculate the prediction MSE, RMSE, and NRMSE on the test set. Use random seed 2022 before you split the data. Interpret the meaning of NRMSE.

```
# random split the data into 80% training and 20% test
set.seed(2022)
index.train <- sample(1:dim(births)[1], 0.8 * dim(births)[1])
data.train <- births[index.train,]
data.test <- births[-index.train,]

# fit a linear model on the training set
lm.model <- lm(weight ~ .,
               data=data.train)

# predict on the test set
yhat.test <- predict(lm.model, data.test)

# calculate test MSE
y.test <- data.test$weight
MSE.test <- mean((y.test - yhat.test)^2)
MSE.test
```

```
## [1] 278.7375
```

```r
# root MSE
RMSE.test <- sqrt(MSE.test)
RMSE.test
```

```
## [1] 16.69543
```

```r
# normalized root MSE
NRMSE.test <- RMSE.test / mean(y.test)
NRMSE.test
```

```
## [1] 0.1421661
```

The linear model gives a 14.2% error for birth weight prediction

part (b)

Repeat the data split and model training in (a), but this time predict on the training set. Calculate the MSE, RMSE, and NRMSE on the training set. Compare with test MSE, RMSE, and RMSE. What did you find? What do you think why you have a such result?

```r
# random split the data into 80% training and 20% test
set.seed(2022)
index.train <- sample(1:dim(births)[1], 0.8 * dim(births)[1])
data.train <- births[index.train,]
data.test <- births[-index.train,]

# fit a linear model on the training set
lm.model <- lm(weight ~ .,
               data=data.train)

# predict on the test set
yhat.train <- predict(lm.model, data.train)

# calculate test MSE
y.train<- data.train$weight
MSE.train <- mean((y.train - yhat.train)^2)
MSE.train
```

```
## [1] 250.2563
```

```r
# root MSE
RMSE.train <- sqrt(MSE.train)
RMSE.train
```

```
## [1] 15.81949
```

```r
# normalized root MSE
NRMSE.train <- RMSE.train / mean(y.train)
NRMSE.train
```

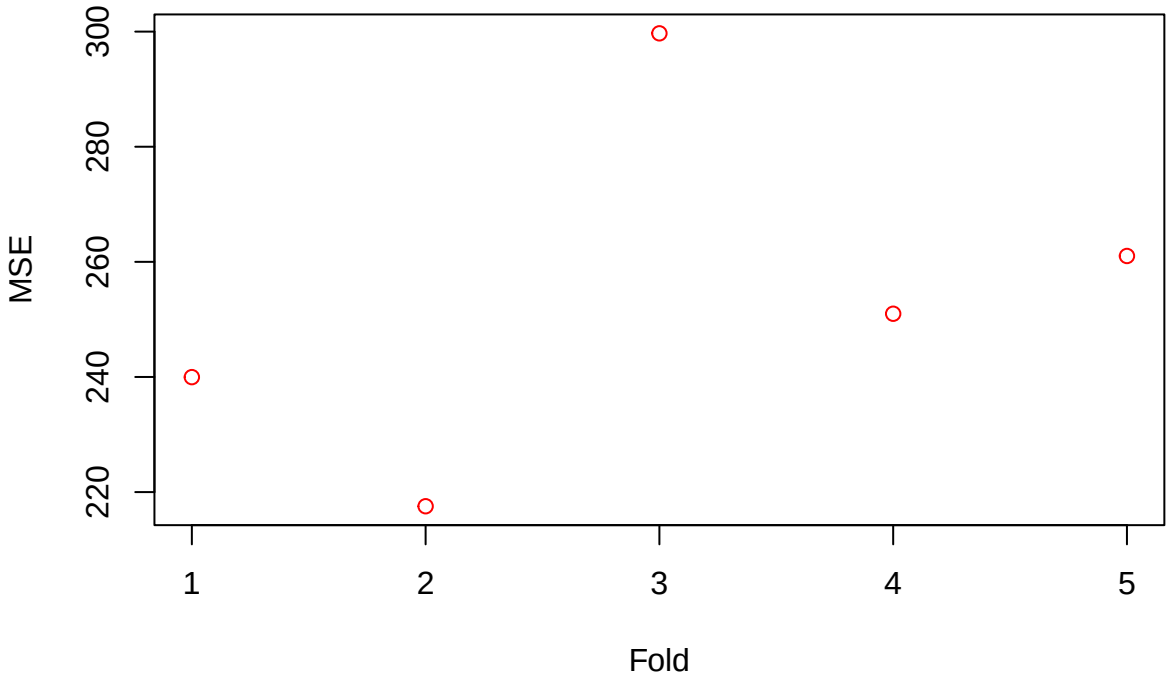
```
## [1] 0.1367234
```

This linear model gives a 13.7% error for birth weight prediction. This model provides better results than the model in part (a) with a lower MSE and RMSE as well as a lower prediction error. This is due to the fact that the model was trained on the training data, so it had already learned the patterns present in the data whereas it had not been trained on the test set so it was predicting on completely new data in the previous part.

part (c)

Conduct a 5-fold cross-validation to predict weight. Plot the test MSE for each fold. Show the average test MSE obtained from the cross-validation. Again, use 2022 as the random seed.

```
set.seed(2022)

# randomly shuffle the index
index.random <- sample(1:dim(births)[1])

# split the data (index) into 5 folds
groups <- cut(1:1992, 5, labels = FALSE)
index.fold <- split(index.random, groups)

# an empty vector to save individual MSE
MSEs <- c()

# 5-fold cross-validation
for(index.test in index.fold){

  # creat training and test set
  data.test <- births[index.test,]
  data.train <- births[-index.test,]

  # fit a linear model on the training set
  lm.model <- lm(weight ~ .,
                 data=births)

  # predict on the test set
  yhat.test <- predict(lm.model, data.test)

  # calculate test MSE
  y.test <- data.test$weight
  MSE.test <- mean((y.test - yhat.test)^2)
  MSEs <- c(MSEs, MSE.test)
}

# plot 5 MSEs
plot(1:5, MSEs, col='red', xlab='Fold', ylab='MSE')
```




```
# Average 5 MSEs
mean(MSEs)
```

```
## [1] 253.8412
```