Homework 2

Ashutosh Ekade

Question 1: Covariance

1.a) Show that the correlation between random variables X and Y (a feature of their joint distribution) is equivalent when de-meaning one or both variables. Namely, using the notation that $\mathbb{E}[X] = \mu_X$ and $\mathbb{E}[Y] = \mu_Y$, show that cov$(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] = \mathbb{E}[(X - \mu_X)Y]$.

$cov(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] = E[XY - \mu_X Y - \mu_Y X - \mu_X \mu_Y] = E[XY] - \mu_X E[Y] - \mu_Y E[X] + \mu_X \mu_Y = E[XY] - \mu_X \mu_Y - \mu_X \mu_Y + \mu_X \mu_Y = E[XY] - \mu_X \mu_Y$

Derive $E[(X - \mu_X)Y]$

$E[(X - \mu_X)Y] = E[XY - \mu_X Y] = E[XY] - \mu_X E[Y] = E[XY] - \mu_X \mu_Y = cov(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$

Similarly, derive $E[(Y - \mu_Y)X]$

$E[(X - \mu_Y)X] = E[XY - \mu_Y X] = E[XY] - \mu_Y E[X] = E[XY] - \mu_Y \mu_X = cov(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$

So,$cov(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] = E[(X - \mu_X)Y] = E[(Y - \mu_Y)X$

1.b) Show that the sample correlation between vectors X and Y is equivalent when de-meaning one or both variables. Namely, using the notation that $n\bar{X} = \sum_{i=1}^{n} X_i$ and $n\bar{Y} = \sum_{i=1}^{n} Y_i$, show that $(n-1)$côv$(X, Y) = \sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}) = \sum_{i=1}^{n}(X_i - \bar{X})Y_i$.

$(n - 1)\hat{\text{cov}}(X, Y) = \sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}) = \sum_{i=1}^{n}(X_i Y_i) - \sum_{i=1}^{n}(\bar{X} Y_i) - \sum_{i=1}^{n}(\bar{Y} X_i) + \sum_{i=1}^{n}(\bar{X}\bar{Y}) = \sum_{i=1}^{n}(X_i Y_i) - \bar{X} * n * \bar{Y} - n * \bar{X}\bar{Y} + n * \bar{X} * \bar{Y} = \sum_{i=1}^{n}(X_i Y_i) - n * \bar{X} * \bar{Y}$

Derive $\sum_{i=1}^{n}(X_i - \bar{X}) * Y_i$

$\sum_{i=1}^{n}(X_i - \bar{X}) * Y_i = \sum_{i=1}^{n}(X_i Y_i) - \sum_{i=1}^{n}(\bar{X} Y_i) = \sum_{i=1}^{n}(X_i Y_i) - \bar{X} \sum_{i=1}^{n}(Y_i) = \sum_{i=1}^{n}(X_i Y_i) - \bar{X} * n * \bar{Y} = \sum_{i=1}^{n}(X_i Y_i) - n\bar{X}\bar{Y}$

Similary, derive $\sum_{i=1}^{n}(Y_i - \bar{Y}) * X_i$

$\sum_{i=1}^{n}(Y_i-\bar{Y})*X_i = \sum_{i=1}^{n}(X_iY_i) - \sum_{i=1}^{n}(\bar{Y}X_i) = \sum_{i=1}^{n}(X_iY_i) - \bar{Y}\sum_{i=1}^{n}(X_i) = \sum_{i=1}^{n}(X_iY_i) - \bar{Y}*n*\bar{X} = \sum_{i=1}^{n}(X_iY_i) - n\bar{X}\bar{Y}$

So,$(n-1)\hat{\text{cov}}(X,Y) = \sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y}) = \sum_{i=1}^{n}(X_i-\bar{X})*Y_i = \sum_{i=1}^{n}(Y_i-\bar{Y})*X_i$

Question 2: Simpson's Paradox and the FWL Theorem

2.a) Read 3.16 and 3.18 of BHE. Then, load the `multi` dataset which has the quantity sold (`Sales`) and prices (`p1`) for a product at 100 stores. The dataset also contains and prices (`p2`) of a competing product at those 100 stores. Create a scatterplot of `Sales` on the vertical axis and `p1` on the horizontal axis. Regress `Sales` on `p1` (ie, use least squares to find the estimated coefficients for the model $Y = \beta_0 + \beta_1 X + e$ where Y is `Sales` and `X` is `p1`). Briefly summarize the relationship you have discovered between `Sales` and `p1`. What is unusual about your finding?

```
# ...add answer here...

setwd('./')
load('./multi.RData')
ls()
```

```
[1] "has_annotations" "multi"
```

```
head(multi)
```

```
        p1        p2     Sales
1 5.135670  5.204186 144.48788
2 3.495460  8.059732 637.24524
3 7.275341 11.675979 620.78693
4 4.662816  8.364421 549.00714
5 3.584537  2.150292  20.42542
6 5.167917 10.153037 713.00665
```

```
plot(Sales ~ p1, data=multi, xlab="Price", ylab="Sales", col="black", main = "Price vs Sal
```


```
integer(0)
```

```
summary(lm(Sales ~ p1, data=multi))
```

```
Call:
lm(formula = Sales ~ p1, data = multi)

Residuals:
    Min      1Q  Median      3Q     Max
-513.91 -157.69   -1.42  155.20  650.20

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   211.16      66.49   3.176    0.002 **
p1             63.71      13.04   4.886 4.01e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 223.4 on 98 degrees of freedom
Multiple R-squared:  0.1959,    Adjusted R-squared:  0.1877
F-statistic: 23.87 on 1 and 98 DF,  p-value: 4.015e-06
```

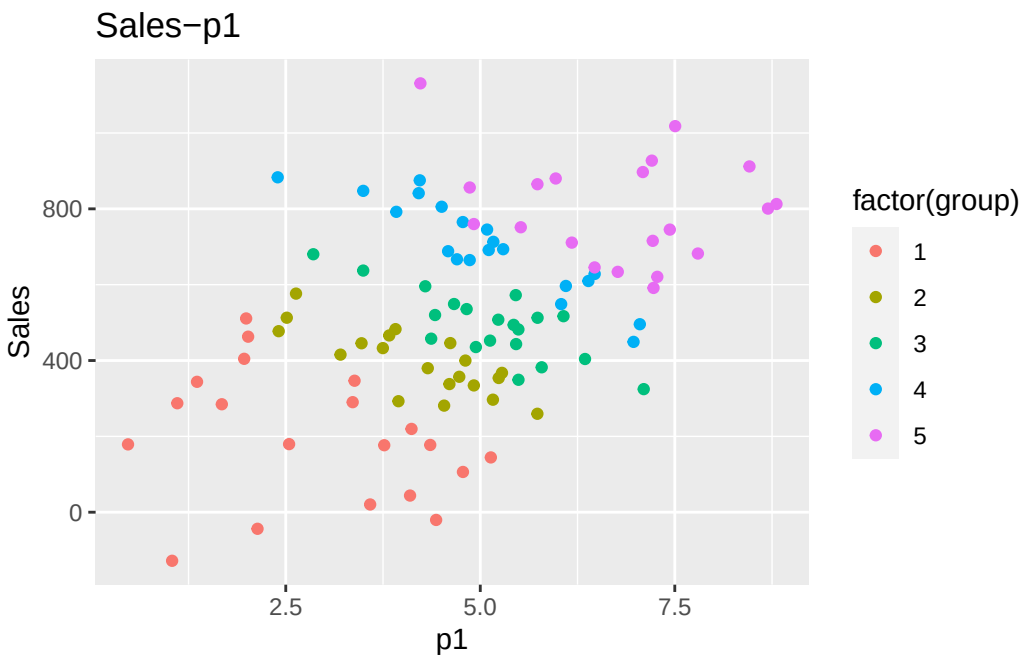
When price goes up, quantitiy sold INCREASES due to positive coefficient. This is unusual as normally quantity sold decreases with increase in price. This could be a Veblen good.

2.b) Sort the dataset by `p2`. Then assign colors to the points in groups of 20 (eg, the first 20 data points are red, the second 20 are blue, etc.). Recreate your scatterplot from 1a above, but now color the points. Regress `Sales` on `p1` and `p2`. What does this plot and the estimated regression coefficients tell you about the relationship between `Sales`, `p1`, and `p2`. See BEH 2.14 for a reminder.

```
multi_sorted <- multi[order(multi$p2),]

library(ggplot2)
multi_sorted$group <- rep(1:ceiling(nrow(multi_sorted)/20), each = 20)
colors = rainbow(max(multi_sorted$group))

ggplot(multi_sorted, aes(x=p1, y=Sales, color = factor(group))) +
    geom_point() +
    ggtitle("Sales-p1") +
    xlab("p1") +
    ylab("Sales")
```



```r
summary(lm(Sales ~ p1+p2, data=multi))
```

```
Call:
lm(formula = Sales ~ p1 + p2, data = multi)

Residuals:
    Min      1Q  Median      3Q     Max
-66.916 -15.663  -0.509  18.904  63.302

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  115.717      8.548   13.54   <2e-16 ***
p1           -97.657      2.669  -36.59   <2e-16 ***
p2           108.800      1.409   77.20   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 28.42 on 97 degrees of freedom
Multiple R-squared:  0.9871,    Adjusted R-squared:  0.9869
F-statistic:  3717 on 2 and 97 DF,  p-value: < 2.2e-16
```

We can see the coefficient of Price P1 is negative, whereas that of Price P2 is positive. This means that with increase in price, quantity sold for P2 goes up whereas that of P1 goes down.

2.c) Regress `p1` on `p2`. From this regression and your observations in 2b above, state some sort of economic theory or business decision that might explain the relationship between `p1` and `p2`.

```r
summary(lm(p1 ~ p2, data=multi))
```

```
Call:
lm(formula = p1 ~ p2, data = multi)

Residuals:
    Min      1Q  Median      3Q     Max
-2.9469 -0.7205  0.1294  0.7971  2.1617

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.49261    0.28628   5.214 1.03e-06 ***
p2           0.41371    0.03316  12.475  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.076 on 98 degrees of freedom
Multiple R-squared:  0.6136,    Adjusted R-squared:  0.6097
F-statistic: 155.6 on 1 and 98 DF,  p-value: < 2.2e-16
```

The coefficient of P2 is positive when regressed against P1. This highlights that when P1 increases, price of its competitor also increases.

2.d) Regress `Sales` on the residuals from the regression you ran in 2c above. Compare the estimated slope coefficient from this regression to your results in 2b above. Explain what is going on here.

```r
summary(lm(Sales ~ residuals(lm(p1 ~ p2, data=multi)), data=multi))
```

```
Call:
lm(formula = Sales ~ residuals(lm(p1 ~ p2, data = multi)), data = multi)

Residuals:
    Min      1Q  Median      3Q     Max
-638.66 -136.88   11.99  150.50  486.84

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                           517.13      22.59  22.893  < 2e-16 ***
residuals(lm(p1 ~ p2, data = multi))  -97.66      21.21  -4.604 1.25e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 225.9 on 98 degrees of freedom
Multiple R-squared:  0.1778,    Adjusted R-squared:  0.1694
F-statistic: 21.19 on 1 and 98 DF,  p-value: 1.246e-05
```

The coeffienct of the residuals is the same as in regression2. This is because reg3 represents the relationship between price1 and price2, and its residuals contain parts of price1 that price2 cannot explain. Doing a direct regression of sales on the residuals yields the direct relationship

between price1 and sales, stripping out the effect of pice1 on sales via price2, and so yields the net effect of price1 in reg2.

Question 3: Standard Errors

3.a) Write a function that takes 5 inputs:

1. `mu` a length-two vector of means for X_1 and X_2,
2. `sd` a length-two vector of standard deviations for X_1 and X_2,
3. `rho` the correlation between X_1 and X_2,
4. `n` the sample size, and
5. `beta` a length-three vector of coefficients.

The function should draw `n` values of X_1 and `n` values of X_2 from a multivariate normal distribution MVN(μ, Σ) where μ is the length-two vector of means and Σ is the 2×2 variance-covariance matrix that must be constructed from `sd` and `rho` (you may find the function `mvrnorm()` from the `MASS` package to be helpful once you've calculated Σ from `sd` and `rho`).

The function should then use those n draws of X_1 and X_2 along with `beta` to compute $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + e$ where the length-n vector e is drawn from a N(0,2^2) distribution.

The function should return an $n \times 3$ data.frame where the first column is Y and the next two columns are X_1 and X_2.

```
library(MASS)
my_reg_func<- function(mu,sd,rho,n,beta) {
  cov_matrix=matrix(0,2,2)
  for(i in 1:length(sd)){
    for(j in 1:length(sd)){
      if(i==j){
        cov_matrix[i,j]<-sd[i]*sd[i]
      }else{
        cov_matrix[i,j]<-sd[i]*sd[j]*rho
      }
    }
  }

  Sigma=cov_matrix
  X_matrix=mvrnorm(n, mu, Sigma, tol = 1e-6)
  e <- rnorm(n, mean = 0, sd = 2)
  Y_list<-list()
  for(k in 1:n){
  Y_list[k]<- beta[1]+beta[2]*X_matrix[k,1]+beta[3]*X_matrix[k,2]+e[k]
  }
```

```
  Y_list<-unlist(Y_list)
  df <- data.frame("Y"=Y_list,"X1" = X_matrix[,1],"X2" = X_matrix[,2])
  return(df)
}

mu=c(3,7)
sd=c(2,3)
rho=0.7
n=1000
beta=c(0,1,1)
df2=my_reg_func(mu,sd,rho,n,beta)
```

3.b) Use your function from 3a above to generate a dataset with `mu=c(3,7)`, `sd=c(2,3)`, `rho=0.7`, `n=1000`, and `beta=c(0,1,1)`. Regress Y on X_1 and X_2 using only the first 10 observations and store the value of the standard error of $\hat{\beta}_1$ (call this $s^{(1)}_{\hat{\beta}_1}$). Then regress Y on X_1 and X_2 using only the first 20 observations and store the value of the standard error of $\hat{\beta}_1$ (call this $s^{(2)}_{\hat{\beta}_1}$). Repeat this process until you fit your 100th regression, which uses all 1000 observations. Plot n on the horizontal axis versus $s^{(n)}_{\hat{\beta}_1}$ on the vertical axis for the 100 stored values of $s^{(n)}_{\hat{\beta}_1}$. What does this exercise demonstrate?

```
y<-df2[1:n,1]
x1<-df2[1:n,2]
x2<-df2[1:n,3]
lm_model<-lm(y~x1+x2)
beta_hat <- coef(lm_model)
beta1_hat_list=list()
std_error_list=list()
for(i in 1:100){
  n=i*10
  y<-df2[1:n,1]
  x1<-df2[1:n,2]
  x2<-df2[1:n,3]
  lm_model<-lm(y~x1+x2)
  beta_hat <- summary(lm_model)$coefficients
  beta1_hat_list[i]=beta_hat[2]
  se_beta <- summary(lm_model)$coefficients[2, "Std. Error"]
  std_error_list[i]=se_beta
}
plot(x=1:100,
```

```
      y=std_error_list,
      col = "blue",
      pch = 19,
      cex = 0.5,
      xlab = "n",
      ylab = "standard error of predicted beta1",
      main = "standard error of 100 trials"
)
```

...add answer here...

3.c) Write a loop. Each time through the loop:

1. **start by setting the seed to 567 (ie, `set.seed(567)`)**
2. **create a dataset using your function from 3a above where `mu=c(3,7)`, `sd=c(2,3)`, `n=1000`, and `beta=c(0,1,1)`**
3. **regress Y on X_1 and X_2**
4. **store the value of the standard error of $\hat{\beta}_1$ (call this $s^{(n)}_{\hat{\beta}_1}$)**

Each time through the loop, you will use a different value for `rho`, starting with 0.50, ending with 0.99, and going in increments of 0.01.

Plot `rho` the correlation of X_1 and X_2 (which ranges from 0.5 to 0.99) on the horizontal axis versus $s_{\hat{\beta}_1}^{(n)}$ on the vertical axis for the 50 stored values of $s_{\hat{\beta}_1}^{(n)}$. What does this exercise demonstrate?

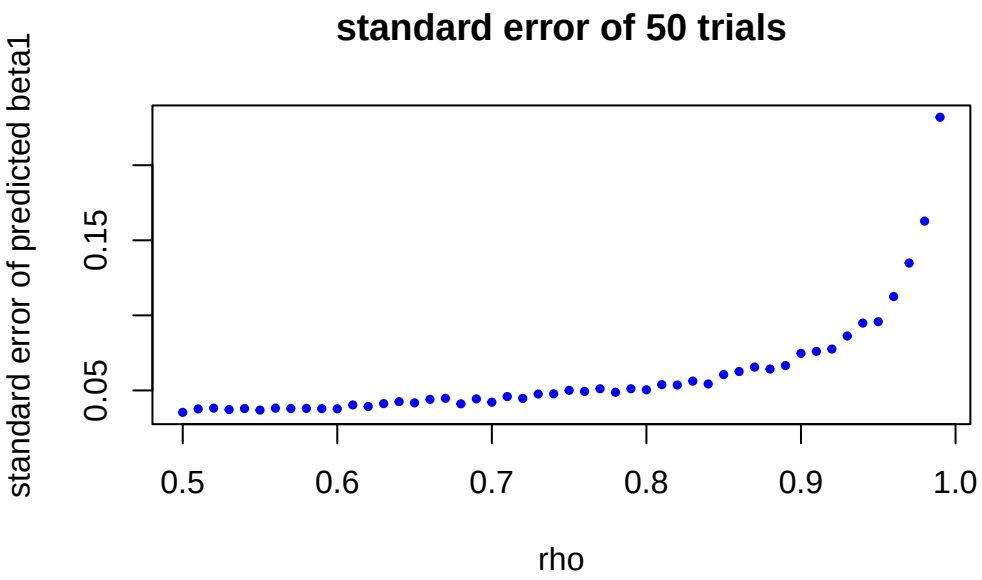
```
set.seed(567)
rho=0.5
rho_list=list()
std_error_list=list()
for(i in 0:49){

  rho=0.5+0.01*i
  rho_list[i+1]=rho
  mu=c(3,7)
  sd=c(2,3)
  n=1000
  beta=c(0,1,1)
  df2=my_reg_func(mu,sd,rho,n,beta)

  y<-df2[,1]
  x1<-df2[,2]
  x2<-df2[,3]
  lm_model<-lm(y~x1+x2)

  beta_hat <- summary(lm_model)$coefficients
  se_beta <- summary(lm_model)$coefficients[2, "Std. Error"]
  std_error_list[i+1]=se_beta
}

plot(x=rho_list,
     y=std_error_list,
     col = "blue",
     pch = 19,
     cex = 0.5,
     xlab = "rho",
     ylab = "standard error of predicted beta1",
     main = "standard error of 50 trials"
)
```


standard error of 50 trials
standard error of predicted beta1
0.05
0.15
0.5
0.6
0.7
0.8
0.9
1.0
rho

Question 4: Standard Errors under Homoskedasticity and Heteroskedasticity

Load the `Hitters` dataset from the `ISLR` package. Drop any rows where Salary is `NA`. Assume the model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + e$ where Y denotes `Salary`, X_1 denotes `Hits`, and X_2 denotes `Years`. See `?ISLR::Hitters` for definitions of these variables.

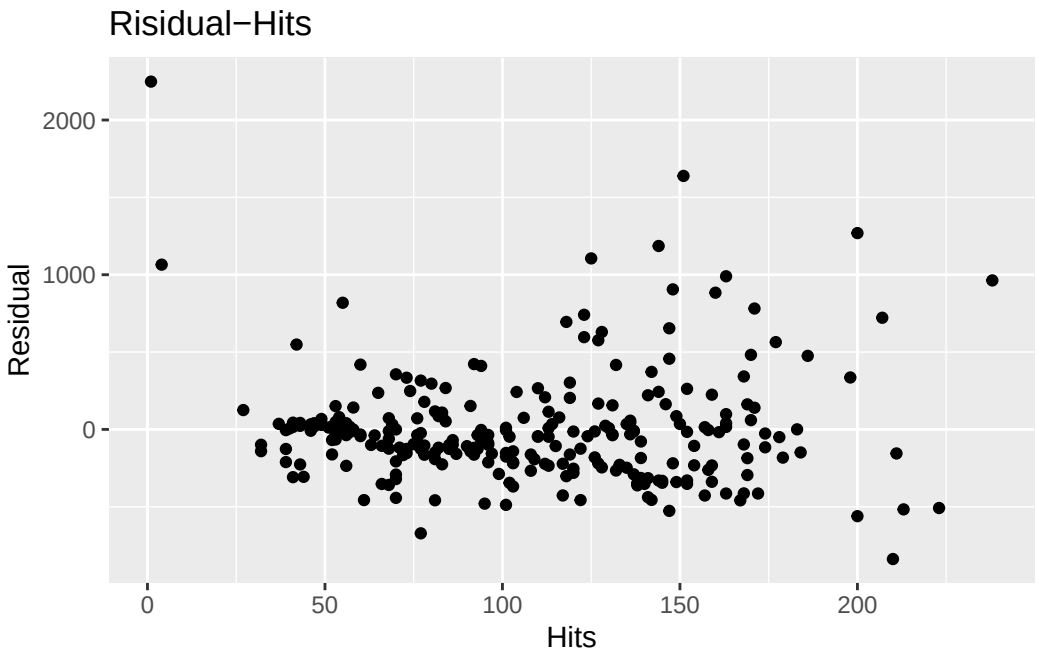
4.a) By hand (ie, without using `lm()` or `summary()`), calculate the OLS estimates of the 3 beta cofficients.

```
library(ISLR)
data(Hitters)
Hitters$isna<-is.na(Hitters$Salary)
df<-Hitters[Hitters$isna==FALSE, ]
y<-matrix(df$Salary,ncol=1)
x<-cbind(1, df$Hits, df$Years)
xx_cross<-solve(crossprod(x))
xy_cross<-crossprod(x, y)
beta_hat<-xx_cross%*%xy_cross
beta_hat
```

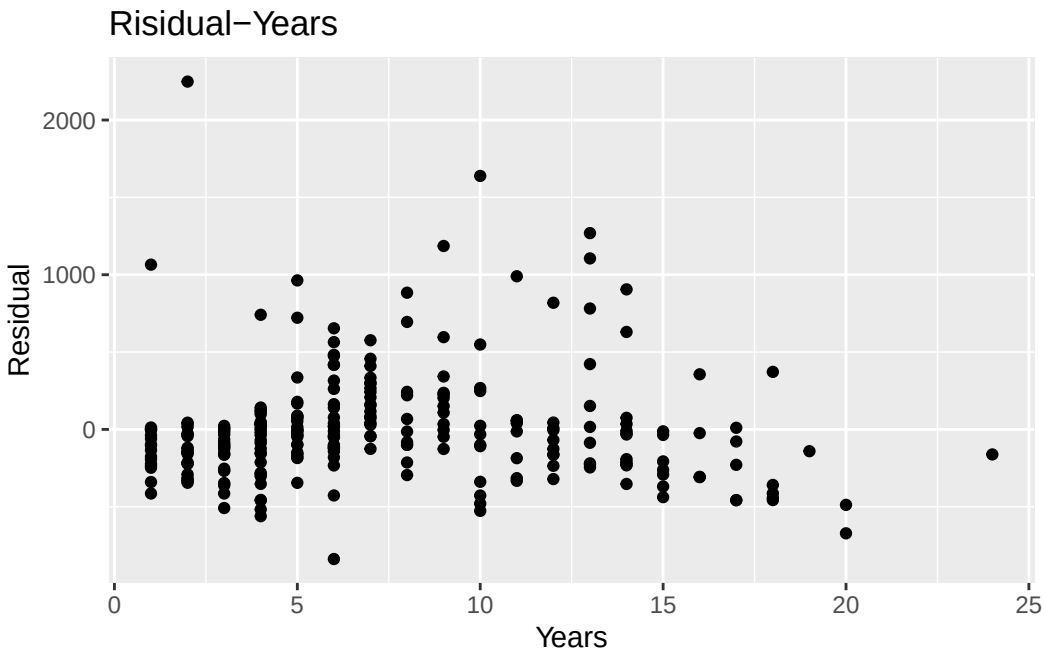
```
            [,1]
[1,] -199.250976
[2,]    4.312438
[3,]   36.950116
```

4.b) Make the following two plots (1) a scatterplot of residuals on the vertical axis and Hits (X_1) on the horizontal axis, (2) the same scatterplot but with Years (X_2) on the horizontal axis instead of Hits (X_1). What do these plots suggest about an assumption of homoskedasticity or heteroskedasticity?

```
library(ggplot2)
y_hat=x %*% beta_hat
e_hat<-y-y_hat
ggplot(df)+
  geom_point(aes(x=Hits,y=e_hat))+
  ggtitle("Risidual-Hits") +
  xlab("Hits") +
  ylab("Residual")
```




```
library(ggplot2)
y_hat=x %*% beta_hat
e_hat<-y-y_hat
ggplot(df)+
  geom_point(aes(x=Years,y=e_hat))+
  ggtitle("Risidual-Years") +
  xlab("Years") +
  ylab("Residual")
```



...add answer here...

4.c) By hand, calculate the standard errors of the OLS estimates under the assumption of homoskedasticity.

```
serr<-(1/(nrow(y)-ncol(x)))*t(e_hat)%*% e_hat
serr<-as.vector(serr)
serrho <-serr*xx_cross
sqrt(diag(serrho))
```

```
[1] 67.4689750  0.5012647  4.7187203
```

4.d) By hand, calculate the standard errors of the OLS estimates under the assumption of heteroskedasticity (use the HC1 estimated variance-covariance matrix).

```
u<-x*(e_hat%*%matrix(1,ncol=ncol(x)))
VHC0<-xx_cross%*%(t(u)%*%u)%*%xx_cross
VHC1<-(nrow(y)/(nrow(y)-ncol(x)))*VHC0
sqrt(diag(VHC1))
```

```
[1] 96.7030610  0.7548996  4.9022273
```

4.e) By hand, calculate R^2 and $\bar{R}^2$ (the adjusted R-squared).

```
ybar=mean(y)
TSS<-sum((y-ybar)^2)
SSE<-t(e_hat)%*%e_hat
r2<-1-SSE/TSS
r2adj<-1-serr/var(y)
r2
```

```
          [,1]
[1,] 0.3465439
```

```
r2adj
```

```
          [,1]
[1,] 0.3415173
```