CS-E-106: Data Modeling

Assignment 2

Instructor: Hakan Gogtas
Submitted by: *Saurabh Kulkarni*

Due Date: *09/30/2019*

Solution 1:

Regression Model: $Y_i = \beta_1 * X_i + \epsilon_i$

$f(y_i) = f_i = \frac{1}{\sqrt{2*pi}*\sigma} \exp(-\frac{1}{2}(\frac{y_i - \beta_1 * X_i}{\sigma})^2)$

Likelihood Function: $L(\beta_1, \sigma^2) = \prod_{i=1}^{n} f_i = (2\pi)^{\frac{-n}{2}} \sigma^{-n} \exp(\frac{-1}{2} \sum_{i=1}^{n} (\frac{y_i - \beta_1 * X_i}{\sigma})^2)$

Goal: Choose values $\hat{\beta_1}$, $\hat{\sigma^2}$ that maximize L (or l = ln(L))

$l = \frac{-n}{2} ln(2\pi) - \frac{n}{2} ln(\sigma^2) - \frac{1}{2} \sum_{i=1}^{n} (\frac{y_i - \beta_1 * X_i}{\sigma})^2$

Calculating optimal β_1*:*

$\frac{\partial l}{\partial \beta_1} = 2 \sum_{i=1}^{n} (\frac{y_i - \beta_1 * X_i}{\sigma})(-X_i) =^{set} 0$

$\implies \sum_{i=1}^{n} X_i y_i = \beta_1 \sum_{i=1}^{n} X_i^2$

$\implies \beta_1 = \frac{\sum_{i=1}^{n} X_i y_i}{\sum_{i=1}^{n} X_i^2}$

Calculating optimal $\hat{\sigma_2}$*:*

$\frac{\partial l}{\partial \sigma^2} = -\frac{n}{2}(\frac{1}{\sigma^2}) - (-1)\frac{1}{2} \sum_{i=1}^{n} (\frac{y_i - \beta_1 * X_i}{\sigma})^2 =^{set} 0$

$\implies \hat{\sigma^2} = \frac{\sum_{i=1}^{n} (y_i - \beta_1 \hat{X}_i)^2}{n}$

Solution 2:

(a)

```
gpa = read.csv("GPA.csv")
lm_gpa = lm(GPA~ACT, data=gpa)
print(summary(lm_gpa))
```

```
##
## Call:
## lm(formula = GPA ~ ACT, data = gpa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.74004 -0.33827  0.04062  0.44064  1.22737
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.11405    0.32089   6.588  1.3e-09 ***
## ACT          0.03883    0.01277   3.040  0.00292 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6231 on 118 degrees of freedom
```

```
## Multiple R-squared:  0.07262,    Adjusted R-squared:  0.06476
## F-statistic:  9.24 on 1 and 118 DF,  p-value: 0.002917
```

Thus, the estimated regression function is:

$GPA = 2.11 + 0.039 * ACT$

(b)

```
confint(lm_gpa, level=0.99)
```

```
##                   0.5 %     99.5 %
## (Intercept) 1.273902675 2.95419590
## ACT         0.005385614 0.07226864
```

We can see that the confidence interval of β_1 does not contain 0. Therefore, there is 99% chance that the β_1 is not actually 0.

(c)

$t = \frac{b_1}{s(b_1)}$

```
t = 0.03883/0.01277
print(t)
```

```
## [1] 3.04072
```

```
pr_t = 1-pt(t, 118)
print(pr_t)
```

```
## [1] 0.001454075
```

Based on the t-test, the probability that $beta_1$ is greater than 0 is 0.001454. This means that there exists a linear relationship between GPA and ACT.

Solution 3:

(a)

```
Xh<-data.frame(ACT=28)
predict(lm_gpa,Xh,se.fit=TRUE,interval="confidence",level=0.95)
```

```
## $fit
##        fit      lwr      upr
## 1 3.201209 3.061384 3.341033
##
## $se.fit
## [1] 0.07060873
##
## $df
## [1] 118
##
## $residual.scale
## [1] 0.623125
```

Interpretation:

- This means that we can predict with 95% confidence that β_1 is within the range: (3.061384, 3.341033)
- Narrower C.I. means that we are more confident about our fit for $X_h = 28$
- Also, since we are using the entire data set for development of our model, it might give us a better fit on the observation that is within the sample.

(b)

```r
Xh<-data.frame(ACT=28)
predict(lm_gpa,Xh, se.fit=TRUE, interval="prediction",level=0.95)
```

```
## $fit
##        fit      lwr      upr
## 1 3.201209 1.959355 4.443063
##
## $se.fit
## [1] 0.07060873
##
## $df
## [1] 118
##
## $residual.scale
## [1] 0.623125
```

Interpretation:

- This means that we can predict with 95% confidence that β_1 is within the range: (1.959355, 4.443063)
- This range is considerably larger than that in part a, as we are now calculating the C.I. for an out of sample observation.

(c)

Yes, the prediction interval in part (b) is wider than the confidence interval in part (a). This makes sense and is expected, since an observation outside the development model will have wider distribution and more variation.

(d)

We will use the formula for Working-Hotelling $1-\alpha$ confidence band for a regression line:

$\hat{Y_h} +_- W * s(\hat{Y_h})$

where, $W = 2F(1-\alpha; 2, n-2)$

```r
Xh<-data.frame(ACT=28)
W <- sqrt( 2 * qf(0.95,2,118))
CI<-predict(lm_gpa, Xh, se.fit=TRUE,interval="confidence",level=0.95)
print(CI)
```

```
## $fit
##        fit      lwr      upr
## 1 3.201209 3.061384 3.341033
##
## $se.fit
## [1] 0.07060873
##
## $df
## [1] 118
##
## $residual.scale
## [1] 0.623125
```

```r
cbind(CI$fit[,1]-W*CI$se.fit, CI$fit[,1] + W*CI$se.fit )
```

```
##          [,1]     [,2]
## [1,] 3.026159 3.376258
```

We see that the band on the regression line at $X_h = 28$ is wider than that in part (a). This is expected, because in part (a) we use t multiple to calculate the confidence interval, whereas, in part (d) we use W multiple. And W multiple is larger than t since it encompasses the entire regression line whereas C.I. for $E[\hat{Y}_h]$ at $X_h = 28$ applies only to that particular observation.

Solution 4:

```
set.seed(1234)
train_ind = sample(1:nrow(gpa), 0.7 * nrow(gpa))
test_ind = setdiff(1:nrow(gpa), train_ind)
train_gpa = gpa[train_ind,]
test_gpa = gpa[test_ind,]
```

```
lm_gpa_tr = lm(GPA~ACT, data=train_gpa)
print(summary(lm_gpa_tr))
```

```
##
## Call:
## lm(formula = GPA ~ ACT, data = train_gpa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.65607 -0.36415  0.01958  0.46192  1.14472
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.35221    0.40813   5.763  1.4e-07 ***
## ACT          0.02772    0.01630   1.701   0.0927 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6375 on 82 degrees of freedom
## Multiple R-squared:  0.03408,    Adjusted R-squared:  0.0223
## F-statistic: 2.893 on 1 and 82 DF,  p-value: 0.09273
```

Thus, the regression function based on the development sample becomes:

$GPA = 2.35221 + 0.02772 * ACT$

(a)

```
Xh<-data.frame(ACT=28)
predict(lm_gpa_tr, Xh, se.fit=TRUE, interval="confidence",level=0.95)
```

```
## $fit
##        fit      lwr      upr
## 1 3.128354 2.953027 3.303681
##
## $se.fit
## [1] 0.08813404
##
## $df
## [1] 82
##
## $residual.scale
## [1] 0.6374881
```

Interpretation:

- This means that we can predict with 95% confidence that β_1 is within the range: (2.953027, 3.303681)
- Now that we are using the only the training data set for development of our model, we get slightly higher variation in our prediction than in Q.3(a).

(b)

```
Xh<-data.frame(ACT=28)
predict(lm_gpa_tr, Xh, se.fit=TRUE, interval="prediction",level=0.95)
```

```
## $fit
##        fit      lwr      upr
## 1 3.128354 1.848125 4.408583
##
## $se.fit
## [1] 0.08813404
##
## $df
## [1] 82
##
## $residual.scale
## [1] 0.6374881
```

Interpretation:

- This means that we can predict with 95% confidence that β_1 is within the range: (1.848125, 4.408583)
- Now that we are using the only the training data set for development of our model, we get slightly higher variation in our prediction than in Q.3(b).

(c)

Yes, the prediction interval in part (b) is wider than the confidence interval in part (a). This makes sense and is expected, since an observation outside the development model will have wider distribution and more variation.

(d)

```
Xh = data.frame(ACT=28)
W = sqrt( 2 * qf(0.95,2,82))
CI = predict(lm_gpa_tr, Xh, se.fit=TRUE,interval="confidence",level=0.95)
print(CI)
```

```
## $fit
##        fit      lwr      upr
## 1 3.128354 2.953027 3.303681
##
## $se.fit
## [1] 0.08813404
##
## $df
## [1] 82
##
## $residual.scale
## [1] 0.6374881
```

```
cbind(CI$fit[,1]-W*CI$se.fit, CI$fit[,1] + W*CI$se.fit )
```

```
##          [,1]     [,2]
## [1,] 2.908623 3.348085
```

We see that the band on the regression line at $X_h = 28$ is wider than that in part (a). This is expected, because in part (a) we use t multiple to calculate the confidence interval, whereas, in part (d) we use W multiple. And W multiple is larger than t since it encompasses the entire regression line whereas C.I. for $E[\hat{Y}_h]$ at $X_h = 28$ applies only to that particular observation.

Again, the band is wider than Q.3(d) since we are only using training set for development of our model.

Calculating MSE for the hold-out set:

```
Yhat = predict(lm_gpa_tr, test_gpa)
resids = (test_gpa$GPA-Yhat)
SSE = sum(resids^2)
df_resids = (nrow(test_gpa)-2)
MSE_te = (SSE/df_resids)
print(MSE_te)
```

```
## [1] 0.3807896
```

Solution 5:

(a)

$\epsilon_i \; N(0, 25)$

$\implies \sigma = \sqrt{25} = 5$

```
get_xy_data <- function(){
  X = array(c(4,8,12,16,20))
  ei = rnorm(5, 0, 5)

  Y = (20+4*X+ei)
  xy_data = data.frame(cbind(X,Y))
  xy_data
}

set.seed(1234)
xy_data = get_xy_data()
print(xy_data)
```

```
##    X         Y
## 1  4  29.96467
## 2  8  53.38715
## 3 12  73.42221
## 4 16  72.27151
## 5 20 102.14562
```

```
lm_xy = lm(Y~X, data=xy_data)
summary(lm_xy)
```

```
##
## Call:
## lm(formula = Y ~ X, data = xy_data)
##
## Residuals:
##       1       2       3       4       5
##  -3.624   3.474   7.184 -10.291   3.258
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  17.2644     8.4197   2.050  0.13270
## X             4.0812     0.6347   6.431  0.00762 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.028 on 3 degrees of freedom
## Multiple R-squared:  0.9324, Adjusted R-squared:  0.9098
## F-statistic: 41.35 on 1 and 3 DF,  p-value: 0.007624
```

Thus, we have, $Y_i = b_0 + b_1 * X_i$ where $b_0 = 17.2644$ and $b_1 = 4.0812$

```
Xh = data.frame(X=10)
predict(lm_xy, Xh, se.fit=TRUE, interval="confidence",level=0.95)
```

```
## $fit
##        fit      lwr      upr
## 1 58.07592 45.95738 70.19445
##
## $se.fit
## [1] 3.807931
##
## $df
## [1] 3
##
## $residual.scale
## [1] 8.027824
```

(b)

```
for(i in 1:200){
  data = get_xy_data()
  lm = lm(Y~X, data=data)
  Xh = data.frame(X=10)
  CI = data.frame(predict(lm, Xh, se.fit=TRUE, interval="confidence",level=0.95)$fit, row.names=NULL)
  b1 = data.frame(b1 = as.numeric(lm$coefficients[2]))
  EY = mean(predict(lm, data))
  if(i==1){
    df_results = cbind(CI,b1,EY)
  }
  else{
    df_results = rbind(df_results, cbind(CI,b1,EY))
  }

}

df_results
```

```
##          fit      lwr      upr       b1       EY
## 1   58.62567 55.99191 61.25943 3.652263 65.93019
## 2   57.78807 53.65183 61.92430 4.492027 66.77212
## 3   58.86465 48.73191 68.99739 4.590760 68.04617
## 4   59.14506 55.05052 63.23960 3.911832 66.96872
## 5   57.04315 49.86885 64.21745 4.054327 65.15181
## 6   59.15902 55.12275 63.19530 3.313944 65.78691
## 7   53.72886 48.95920 58.49852 4.411149 62.55116
## 8   59.27544 52.04986 66.50102 3.487543 66.25053
```

```
## 9   55.26980 53.14433 57.39527 4.190852 63.65150
## 10  54.61209 50.96072 58.26346 4.356820 63.32573
## 11  62.75653 52.42106 73.09200 3.564545 69.88562
## 12  63.63033 54.84214 72.41853 3.431624 70.49358
## 13  63.77560 53.56314 73.98806 3.948369 71.67234
## 14  60.59583 55.10493 66.08673 4.653814 69.90346
## 15  57.34311 52.20319 62.48304 4.165205 65.67352
## 16  58.44330 52.19209 64.69450 4.256531 66.95636
## 17  60.59022 58.17170 63.00874 3.434267 67.45875
## 18  62.44856 54.82617 70.07096 3.982684 70.41393
## 19  61.78352 55.01571 68.55132 4.704832 71.19318
## 20  59.30452 56.70081 61.90824 3.686400 66.67732
## 21  59.53851 55.38437 63.69265 4.101631 67.74177
## 22  58.43313 52.58012 64.28615 4.241867 66.91687
## 23  60.35211 53.21165 67.49258 4.025328 68.40277
## 24  62.26867 54.71938 69.81796 4.695451 71.65957
## 25  57.45362 51.41831 63.48892 4.035974 65.52556
## 26  60.34101 52.71159 67.97044 3.386048 67.11311
## 27  57.48644 48.63779 66.33508 4.090267 65.66697
## 28  62.05511 56.12670 67.98353 4.024002 70.10312
## 29  63.50685 58.77035 68.24336 3.842497 71.19185
## 30  60.59910 52.39372 68.80448 4.003098 68.60529
## 31  59.84914 53.78909 65.90918 3.482540 66.81422
## 32  58.05500 56.36296 59.74703 4.323300 66.70160
## 33  63.46089 55.69241 71.22937 3.961712 71.38431
## 34  61.61685 49.66133 73.57236 3.693548 69.00394
## 35  60.54233 46.71045 74.37420 4.370929 69.28418
## 36  57.51953 49.61911 65.41994 4.975343 67.47021
## 37  64.35717 58.54390 70.17045 3.396347 71.14987
## 38  56.28343 45.72217 66.84469 5.094472 66.47237
## 39  59.64327 53.26751 66.01904 3.298073 66.23942
## 40  62.46571 60.52204 64.40937 3.957109 70.37992
## 41  64.02341 57.89868 70.14814 3.305047 70.63350
## 42  60.20710 51.02222 69.39198 3.640692 67.48848
## 43  59.92342 55.45633 64.39050 4.032759 67.98894
## 44  65.46549 58.97317 71.95780 3.912295 73.29008
## 45  62.91669 50.31401 75.51937 3.761133 70.43896
## 46  58.48374 54.98302 61.98446 4.209832 66.90340
## 47  54.40665 45.09950 63.71379 4.825788 64.05822
## 48  60.40051 54.22086 66.58015 4.202160 68.80483
## 49  62.81083 58.90768 66.71398 3.867484 70.54580
## 50  62.36195 56.92087 67.80304 3.557058 69.47607
## 51  60.46261 51.77255 69.15268 4.041512 68.54564
## 52  60.03309 51.85941 68.20677 3.663901 67.36089
## 53  58.13607 53.35188 62.92026 4.411234 66.95854
## 54  64.23108 57.88644 70.57571 4.478957 73.18899
## 55  61.17558 55.66041 66.69076 4.140693 69.45697
## 56  58.57355 51.42154 65.72556 4.093037 66.75963
## 57  58.07475 51.57775 64.57176 3.863967 65.80269
## 58  60.30362 48.62312 71.98411 4.067177 68.43797
## 59  61.30401 51.92783 70.68018 3.818420 68.94085
## 60  58.52821 53.75407 63.30234 4.396282 67.32077
## 61  60.05675 52.36832 67.74518 4.360228 68.77721
## 62  62.06430 58.54164 65.58695 3.448542 68.96138
```

```
## 63  62.45985 55.52662 69.39308 3.479784 69.41942
## 64  58.56947 52.84026 64.29868 3.771035 66.11154
## 65  59.84605 50.83303 68.85908 4.022367 67.89079
## 66  61.41518 56.85329 65.97707 3.861856 69.13890
## 67  59.00988 49.74659 68.27317 4.086351 67.18258
## 68  62.19520 57.85673 66.53366 4.140364 70.47593
## 69  57.83533 47.83603 67.83463 3.616464 65.06826
## 70  55.48217 45.70437 65.25997 4.083668 63.64950
## 71  63.19718 52.44650 73.94786 4.099824 71.39683
## 72  63.41780 58.23953 68.59608 3.380973 70.17975
## 73  61.94306 50.71736 73.16875 4.359257 70.66157
## 74  56.98467 51.17295 62.79639 4.125932 65.23654
## 75  59.38507 55.42112 63.34902 3.981057 67.34718
## 76  54.86198 47.21583 62.50812 5.181518 65.22501
## 77  62.33896 55.40963 69.26830 3.436433 69.21183
## 78  58.46063 41.33071 75.59055 4.295195 67.05102
## 79  61.84704 59.37815 64.31592 3.518222 68.88348
## 80  57.31777 52.47213 62.16342 4.294124 65.90602
## 81  60.51195 52.83933 68.18457 3.695268 67.90249
## 82  55.07161 48.08207 62.06116 4.048122 63.16786
## 83  60.50053 54.17372 66.82735 4.097238 68.69501
## 84  58.95987 54.93138 62.98836 4.447714 67.85529
## 85  61.69838 52.90489 70.49187 3.802081 69.30254
## 86  54.09483 50.85560 57.33407 4.300626 62.69608
## 87  64.23693 57.71859 70.75527 3.240417 70.71776
## 88  59.37874 56.79189 61.96559 3.455344 66.28942
## 89  63.31266 54.90584 71.71948 4.170916 71.65449
## 90  63.75169 57.30068 70.20270 3.872646 71.49698
## 91  56.76645 49.40024 64.13266 4.777683 66.32182
## 92  59.25992 52.40047 66.11937 3.837599 66.93512
## 93  53.90987 50.08298 57.73677 4.322462 62.55480
## 94  63.05383 59.95476 66.15290 3.455514 69.96486
## 95  61.49028 53.11556 69.86501 3.771657 69.03360
## 96  59.13237 49.49814 68.76660 4.146724 67.42582
## 97  61.88364 41.99141 81.77587 4.045070 69.97378
## 98  59.83900 46.06098 73.61702 3.470712 66.78042
## 99  64.63543 57.07094 72.19992 4.255906 73.14724
## 100 61.46536 52.21151 70.71922 4.338897 70.14316
## 101 58.47485 52.25473 64.69497 4.291656 67.05816
## 102 57.95026 52.33161 63.56891 4.407651 66.76556
## 103 56.78192 47.81191 65.75194 4.936603 66.65513
## 104 62.98256 57.06877 68.89634 3.482263 69.94708
## 105 58.60140 56.54822 60.65458 4.302211 67.20582
## 106 62.36384 56.48044 68.24724 4.174781 70.71340
## 107 58.24397 53.92775 62.56020 4.367832 66.97964
## 108 61.33274 56.03231 66.63317 3.584433 68.50160
## 109 56.96437 51.72826 62.20048 4.747890 66.46015
## 110 54.93035 52.74250 57.11821 4.445173 63.82070
## 111 58.03160 44.44324 71.61995 4.226307 66.48421
## 112 57.74931 55.22507 60.27354 4.503739 66.75678
## 113 61.86472 52.68850 71.04095 3.700023 69.26477
## 114 56.79558 50.06285 63.52831 3.965741 64.72706
## 115 56.39720 48.34805 64.44636 3.832813 64.06283
## 116 59.69036 55.39142 63.98931 4.302202 68.29477
```

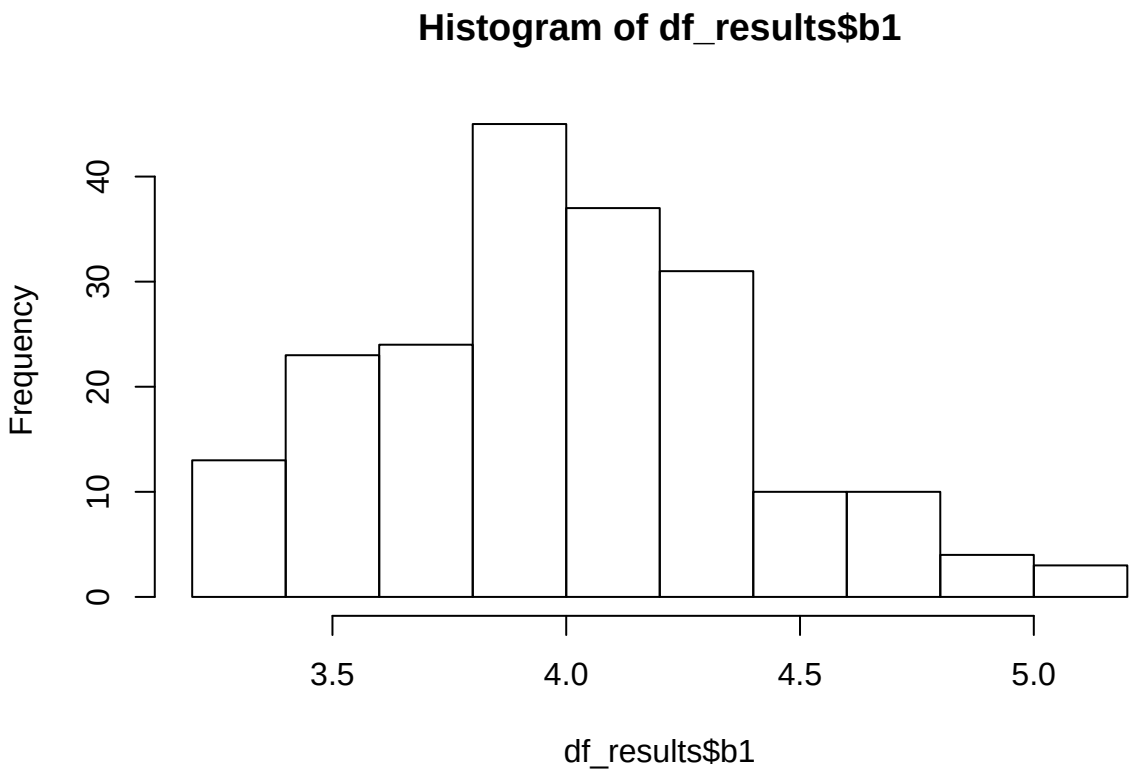
```
## 117 57.36658 48.01275 66.72041 4.283330 65.93324
## 118 62.33830 59.03278 65.64381 3.992321 70.32294
## 119 58.57717 52.72115 64.43319 3.819066 66.21530
## 120 60.13921 51.50001 68.77842 3.946140 68.03149
## 121 62.23648 52.73394 71.73903 4.016433 70.26935
## 122 59.06672 52.65249 65.48096 3.792403 66.65153
## 123 60.38294 52.76739 67.99848 3.665682 67.71430
## 124 61.42480 55.27143 67.57818 3.842299 69.10940
## 125 61.61922 52.60896 70.62948 3.689314 68.99785
## 126 61.44380 52.62618 70.26143 3.202732 67.84927
## 127 58.55530 54.45967 62.65092 4.669352 67.89400
## 128 59.30086 55.78770 62.81402 3.929061 67.15898
## 129 57.73117 53.70941 61.75293 4.223998 66.17917
## 130 58.28286 55.33325 61.23248 4.252714 66.78829
## 131 61.39122 57.07105 65.71139 3.945935 69.28309
## 132 56.52004 49.76530 63.27477 3.662306 63.84465
## 133 56.90073 46.53456 67.26690 4.225395 65.35152
## 134 60.62951 56.24349 65.01553 3.503513 67.63654
## 135 59.37487 52.07287 66.67687 3.418016 66.21090
## 136 57.36015 48.23702 66.48329 3.883278 65.12671
## 137 57.62996 49.06158 66.19835 3.863389 65.35674
## 138 62.59743 57.64705 67.54781 4.137390 70.87221
## 139 62.40829 52.95580 71.86079 4.239778 70.88785
## 140 61.15128 54.85285 67.44972 3.464021 68.07933
## 141 58.58454 51.71693 65.45215 3.889935 66.36441
## 142 63.91465 59.19925 68.63005 3.304029 70.52271
## 143 55.63397 47.76172 63.50622 5.040570 65.71511
## 144 57.37680 50.85891 63.89469 4.654874 66.68655
## 145 58.88042 55.56905 62.19179 3.999797 66.88001
## 146 64.01698 59.87107 68.16289 3.448386 70.91375
## 147 61.15636 55.05126 67.26146 4.023303 69.20297
## 148 61.61979 50.48661 72.75298 3.391330 68.40245
## 149 62.83940 56.26277 69.41604 3.998010 70.83543
## 150 60.88386 55.00376 66.76396 3.835450 68.55476
## 151 57.12719 50.95057 63.30382 3.747138 64.62147
## 152 61.08357 53.16926 68.99787 4.048515 69.18060
## 153 62.45654 55.99480 68.91828 3.962906 70.38235
## 154 63.56069 51.76234 75.35904 3.673051 70.90679
## 155 57.63230 53.62469 61.63991 4.166155 65.96461
## 156 60.58124 50.90031 70.26216 3.688748 67.95873
## 157 57.08930 45.49364 68.68495 3.864070 64.81744
## 158 59.86569 54.45686 65.27452 3.932736 67.73116
## 159 59.49238 54.15287 64.83189 3.351643 66.19566
## 160 57.98878 49.71374 66.26382 4.630932 67.25065
## 161 59.81306 53.26577 66.36035 3.925537 67.66413
## 162 62.01266 51.17193 72.85339 4.209888 70.43244
## 163 58.77526 56.81635 60.73418 3.744540 66.26434
## 164 59.65670 52.07033 67.24307 4.604773 68.86625
## 165 60.20545 54.60048 65.81043 3.588397 67.38225
## 166 60.07129 56.42205 63.72054 4.295789 68.66287
## 167 64.15359 57.52505 70.78214 3.358536 70.87067
## 168 57.46495 52.03887 62.89104 4.374289 66.21353
## 169 56.37908 46.38911 66.36906 3.995459 64.37000
## 170 59.44667 49.55225 69.34108 3.931769 67.31020
```

```
## 171 58.53523 53.65145 63.41901 4.079007 66.69324
## 172 58.57681 53.84484 63.30878 3.924428 66.42567
## 173 60.48300 49.52125 71.44475 3.926737 68.33648
## 174 58.09517 52.39592 63.79442 3.771054 65.63728
## 175 63.05783 57.54161 68.57405 3.778589 70.61501
## 176 59.26285 54.74112 63.78458 4.813106 68.88906
## 177 61.01298 50.94104 71.08492 4.280528 69.57404
## 178 61.08401 53.98576 68.18227 4.068149 69.22031
## 179 62.83449 57.66100 68.00798 4.149771 71.13403
## 180 53.88865 45.81517 61.96212 4.734778 63.35820
## 181 61.71432 60.80575 62.62289 3.615240 68.94480
## 182 58.91917 52.88979 64.94855 3.853399 66.62597
## 183 59.88782 48.67638 71.09925 3.830234 67.54828
## 184 58.89178 54.78624 62.99732 4.045199 66.98218
## 185 60.38244 55.33074 65.43414 3.477819 67.33808
## 186 59.99194 52.57260 67.41127 3.992120 67.97618
## 187 61.64215 48.14283 75.14148 3.876967 69.39609
## 188 60.12625 48.69232 71.56017 3.440883 67.00801
## 189 56.64040 46.67169 66.60911 3.678251 63.99690
## 190 58.90371 53.68891 64.11850 4.266657 67.43702
## 191 59.66340 54.32537 65.00144 3.994179 67.65176
## 192 61.26722 49.23197 73.30246 4.070085 69.40739
## 193 56.18894 45.48376 66.89412 4.092535 64.37401
## 194 61.18822 49.50259 72.87385 3.935757 69.05974
## 195 60.20521 53.14204 67.26838 3.254556 66.71432
## 196 65.02401 62.45921 67.58881 3.491996 72.00800
## 197 61.94122 58.33633 65.54611 3.903465 69.74815
## 198 60.98308 48.49765 73.46850 3.769463 68.52200
## 199 61.19338 56.62815 65.75862 3.969039 69.13146
## 200 57.85769 50.50897 65.20641 4.518809 66.89531
```

(c)

Histogram of b1

```
hist(df_results$b1)
```



Mean (b1):

```
mean(df_results$b1)
```

```
## [1] 3.994485
```

Standard Deviation (b1):

```
sd(df_results$b1)
```

```
## [1] 0.3998482
```

With mean of $\hat{\beta}_1 = 3.97236$, the results seem pretty consistent with our theoretical expectations, since $E[\hat{\beta}_1] = 4$ (based on the regression given function).

(d)

```
df_results$inc_EY = ifelse(df_results$EY>=df_results$lwr&df_results$EY<=df_results$upr, 1, 0)
print(paste("% of E{Yh} within Confidence Interval:",sum(df_results$inc_EY)*100/200, "%"))
```

```
## [1] "% of E{Yh} within Confidence Interval: 33 %"
```

Here we are calculating the 95% confidence interval at $X_h = 10$ and checking how many times that includes the mean of the fitted values from the 200 experiments. Looking at the regression function, $Y = 20 + 4 * X$ the $E[Y]$ should be the mean of this equation.

$$\implies E[Y] = \frac{\sum_{i=1}^{n} Y_i}{n} = \frac{\sum_{i=1}^{n} (b_0 + b_1 * X_i)}{n} = b_0 + b_1 * \bar{X} = 20 + 4 * 12 = 68$$

```
mu = mean(xy_data$X)
sigma = 5
Z = (10-mu)/5

print(Z)
```

```
## [1] -0.4
```

```
pnorm(10, mu, sigma)
```

```
## [1] 0.3445783
```

Thus, the probability that the X-value is less than 10 is 0.34 which is more or less consistent with the result we got: 39.5%.