S2208 MATH8050 Data Analysis - Section 001: Homework 4 Due on 09/28/22

Adithya Ravi, C09059838

2022-09-28

Solutions

Question1

1a

$$N(x|\theta, l^{-1}) = \sqrt{\frac{l}{2\pi}} exp(-\frac{1}{2})l(x-\theta)^2$$

$$\propto exp(-\frac{1}{2}l(x^2 - 2x\theta + \theta^2))$$

$$\propto exp(lx\theta - \frac{1}{2}l\theta^2)$$

Due to symmetry of the normal p.d.f.,

$$N(\theta|\mu_0, \lambda_0^{-1}) = N(\mu_0|\theta, \lambda_0^{-1}) \propto exp(\lambda_0\mu_0\theta - \frac{1}{2}\lambda_0\theta^2)$$

by $exp(lx\theta - \frac{1}{2}l\theta^2)$ with $x = \mu_0$ and $l = \lambda_0$. Therefore, defining L and M as above,

$$p(\theta|x_{1:n}) \propto 1p(x_{1:n}|\theta)$$

$$\prod_{i=1}^{n} N(x_i|\theta, \lambda^{-1})$$

$$\propto exp(\lambda)(\sum(x_i)\theta - \frac{1}{2}n\lambda\theta^2)$$

1b

$$N(x|\theta, l^{-1}) = \sqrt{\frac{l}{2\pi}} exp(-\frac{1}{2})l(x-\theta)^2$$

$$\propto exp(-\frac{1}{2}l(x^2 - 2x\theta + \theta^2))$$

$$\propto exp(lx\theta - \frac{1}{2}l\theta^2)$$

Due to symmetry of the normal p.d.f.,

$$N(\theta|\mu_0, \lambda_0^{-1}) = N(\mu_0|\theta, \lambda_0^{-1}) \propto exp(\lambda_0\mu_0\theta - \frac{1}{2}\lambda_0\theta^2)$$

by $exp(lx\theta - \frac{1}{2}l\theta^2)$ with $x = \mu_0$ and $l = \lambda_0$. Therefore, defining L and M as above,

$$p(\theta|x_{1:n}) \propto p(\theta)p(x_{1:n}|\theta)$$

$$= N(\theta|\mu_0, \lambda_0^{-1})\prod_{i=1}^{n} N(x_i|\theta, \lambda^{-1})$$

$$\propto exp(\lambda_0\mu_0\theta - \frac{1}{2}\lambda_0\theta^2)exp(\lambda)(\sum(x_i)\theta - \frac{1}{2}n\lambda\theta^2)$$

$$= exp((\lambda_0\mu_0 + \lambda\sum x_i)\theta - \frac{1}{2}(\lambda_0 + n\lambda)\theta^2)$$

$$= exp(LM\theta) - \frac{1}{2}L\theta^2$$

$$\propto N(M|\theta, L^{-1}) = N(\theta|M, L^{-1})$$

where

$$L = \lambda_0 + n\lambda$$

and

$$M = \frac{\lambda_0\mu_0 + \lambda\Sigma_{i=1}^{n}x_i}{\lambda_0 + n\lambda}$$

1c

MLE for μ:

$$\frac{\partial}{\partial\mu}loglikelihood = -\frac{1}{2}\lambda\sum_{i=1}^{n}2(x_i - \mu)(-1)$$

$$= \lambda\sum_{i=1}^{n}(x_i) - \mu = 0$$

$$= \sum_{i=1}^{n}(x_i - \mu) = 0$$

$$= \sum_{i=1}^{n}x_i - n\mu = 0$$

$$=> \mu = \frac{\sum_{i=1}^{n}x_i}{n}$$

MLE for λ:

$$\frac{\partial}{\partial\lambda}(\frac{n}{2\pi}ln(\frac{\lambda}{2\pi}) - \frac{1}{2}\lambda(x_i - \mu)^2) = 0$$

$$\frac{n}{2}\frac{\frac{1}{2\pi}}{\frac{\lambda}{2\pi}} = \frac{1}{2}\sum(x_i - \mu)^2$$

$$\lambda = \frac{n}{\sum(x_i - \mu)^2}$$

$$\lambda = \frac{1}{\frac{\sum(x_i-\mu)^2}{n}}$$

```
set.seed(123)
rnd <- rnorm(100, mean=0, sd=3)
lambda_max <- 1/var(rnd)
meanvalue_max <- mean(rnd)

lambda_max
```

```
## [1] 0.1333494
```

```
meanvalue_max
```

```
## [1] 0.2712177
```

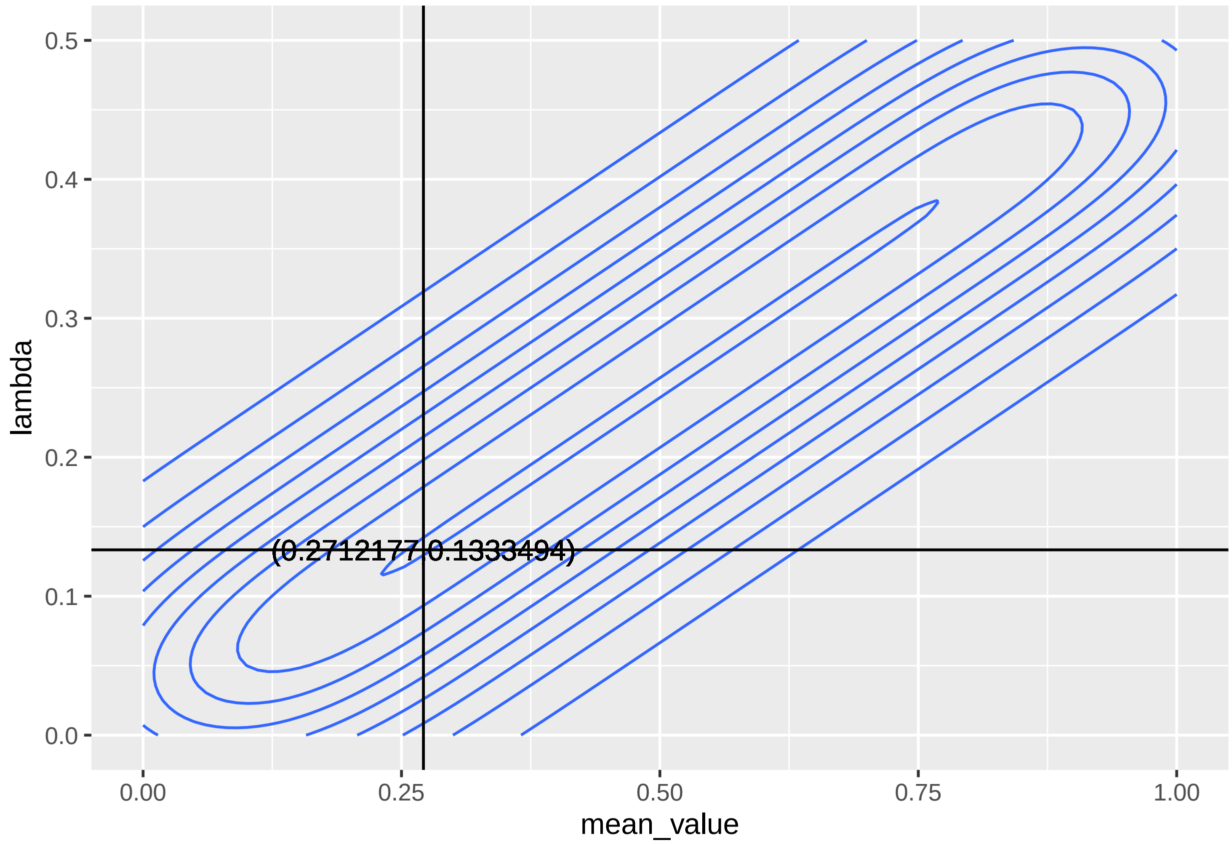
```
my_function<-function(mean_value,lambda){
final <- 0
for (v in rnd) {
  c<-(v-mean_value)**2
  final<-final+c
}

log.likli.hood<-(50)*log(lambda/(2*3.14))-((1/2)*(lambda)*final)

return(log.likli.hood)
}

lambda<-seq(0,0.5,length=100)
mean_value<-seq(0,1,length=100)

z<-my_function(mean_value,lambda)
total<-data.frame(lambda = lambda,mean_value = mean_value, z = z)
total
```

```
##          lambda mean_value         z
## 1   0.000000000 0.00000000      -Inf
## 2   0.005050505 0.01010101 -358.1738
## 3   0.010101010 0.02020202 -325.4059
## 4   0.015151515 0.03030303 -307.0195
## 5   0.020202020 0.04040404 -294.5200
## 6   0.025252525 0.05050505 -285.2453
## 7   0.030303030 0.06060606 -278.0097
## 8   0.035353535 0.07070707 -272.1808
## 9   0.040404040 0.08080808 -267.3812
## 10  0.045454545 0.09090909 -263.3675
## 11  0.050505051 0.10101010 -259.9735
```

```
## 12  0.055555556 0.11111111 -257.0808
## 13  0.060606061 0.12121212 -254.6020
## 14  0.065656566 0.13131313 -252.4707
## 15  0.070707071 0.14141414 -250.6354
## 16  0.075757576 0.15151515 -249.0552
## 17  0.080808081 0.16161616 -247.6973
## 18  0.085858586 0.17171717 -246.5348
## 19  0.090909091 0.18181818 -245.5455
## 20  0.095959596 0.19191919 -244.7108
## 21  0.101010101 0.20202020 -244.0149
## 22  0.106060606 0.21212121 -243.4445
## 23  0.111111111 0.22222222 -242.9881
## 24  0.116161616 0.23232323 -242.6357
## 25  0.121212121 0.24242424 -242.3788
## 26  0.126262626 0.25252525 -242.2096
## 27  0.131313131 0.26262626 -242.1216
## 28  0.136363636 0.27272727 -242.1089
## 29  0.141414141 0.28282828 -242.1662
## 30  0.146464646 0.29292929 -242.2890
## 31  0.151515152 0.30303030 -242.4729
## 32  0.156565657 0.31313131 -242.7142
## 33  0.161616162 0.32323232 -243.0097
## 34  0.166666667 0.33333333 -243.3562
## 35  0.171717172 0.34343434 -243.7509
## 36  0.176767677 0.35353535 -244.1914
## 37  0.181818182 0.36363636 -244.6754
## 38  0.186868687 0.37373737 -245.2008
## 39  0.191919192 0.38383838 -245.7657
## 40  0.196969697 0.39393939 -246.3683
## 41  0.202020202 0.40404040 -247.0070
## 42  0.207070707 0.41414141 -247.6805
## 43  0.212121212 0.42424242 -248.3872
## 44  0.217171717 0.43434343 -249.1261
## 45  0.222222222 0.44444444 -249.8958
## 46  0.227272727 0.45454545 -250.6955
## 47  0.232323232 0.46464646 -251.5240
## 48  0.237373737 0.47474747 -252.3805
## 49  0.242424242 0.48484848 -253.2641
## 50  0.247474747 0.49494949 -254.1741
## 51  0.252525253 0.50505051 -255.1098
## 52  0.257575758 0.51515152 -256.0704
## 53  0.262626263 0.52525253 -257.0553
## 54  0.267676768 0.53535354 -258.0640
## 55  0.272727273 0.54545455 -259.0960
## 56  0.277777778 0.55555556 -260.1506
## 57  0.282828283 0.56565657 -261.2276
## 58  0.287878788 0.57575758 -262.3264
## 59  0.292929293 0.58585859 -263.4466
## 60  0.297979798 0.59595960 -264.5878
## 61  0.303030303 0.60606061 -265.7498
## 62  0.308080808 0.61616162 -266.9322
## 63  0.313131313 0.62626263 -268.1347
## 64  0.318181818 0.63636364 -269.3571
## 65  0.323232323 0.64646465 -270.5989
```

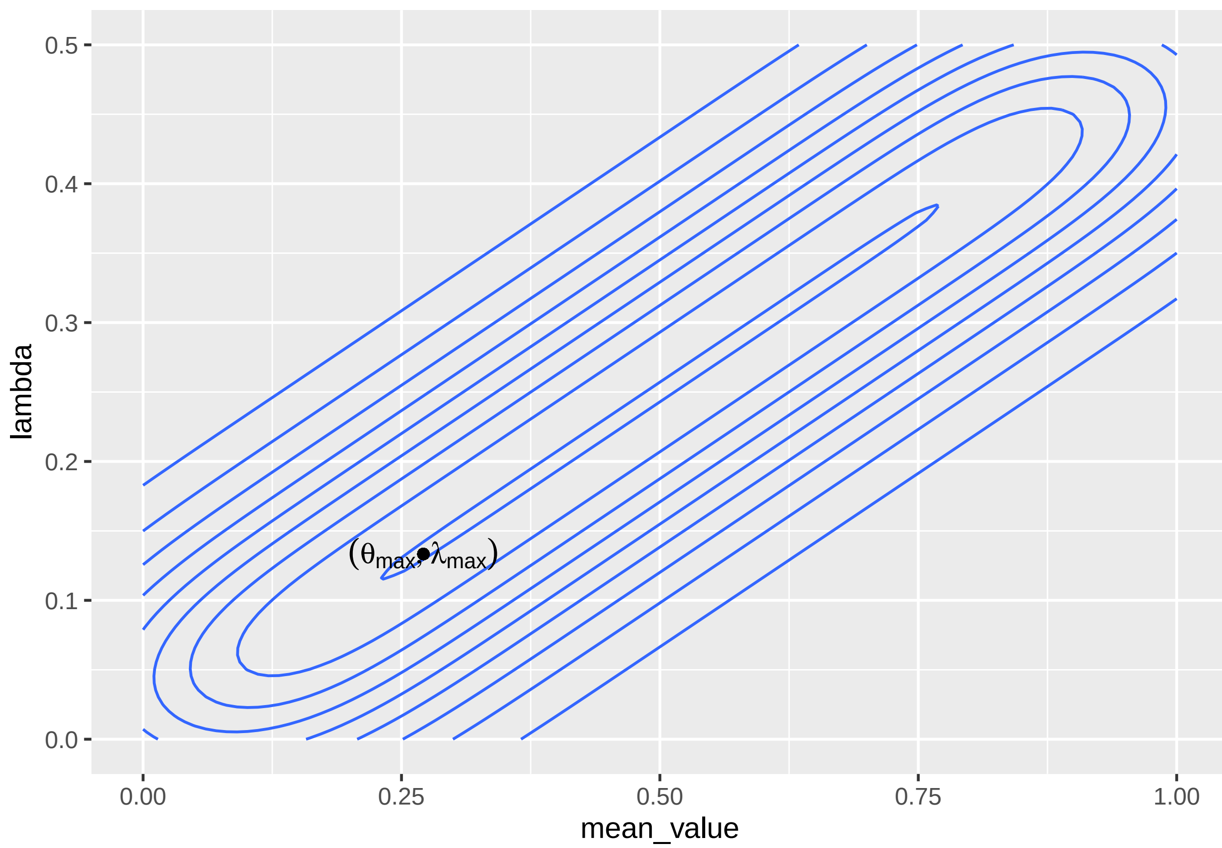
```
## 66  0.328282828 0.65656566 -271.8602
## 67  0.333333333 0.66666667 -273.1405
## 68  0.338383838 0.67676768 -274.4398
## 69  0.343434343 0.68686869 -275.7578
## 70  0.348484848 0.69696970 -277.0943
## 71  0.353535354 0.70707071 -278.4493
## 72  0.358585859 0.71717172 -279.8225
## 73  0.363636364 0.72727273 -281.2138
## 74  0.368686869 0.73737374 -282.6232
## 75  0.373737374 0.74747475 -284.0504
## 76  0.378787879 0.75757576 -285.4955
## 77  0.383838384 0.76767677 -286.9583
## 78  0.388888889 0.77777778 -288.4387
## 79  0.393939394 0.78787879 -289.9367
## 80  0.398989899 0.79797980 -291.4521
## 81  0.404040404 0.80808081 -292.9851
## 82  0.409090909 0.81818182 -294.5355
## 83  0.414141414 0.82828283 -296.1032
## 84  0.419191919 0.83838384 -297.6883
## 85  0.424242424 0.84848485 -299.2907
## 86  0.429292929 0.85858586 -300.9104
## 87  0.434343434 0.86868687 -302.5474
## 88  0.439393939 0.87878788 -304.2017
## 89  0.444444444 0.88888889 -305.8733
## 90  0.449494949 0.89898990 -307.5622
## 91  0.454545455 0.90909091 -309.2683
## 92  0.459595960 0.91919192 -310.9918
## 93  0.464646465 0.92929293 -312.7327
## 94  0.469696970 0.93939394 -314.4909
## 95  0.474747475 0.94949495 -316.2665
## 96  0.479797980 0.95959596 -318.0595
## 97  0.484848485 0.96969697 -319.8700
## 98  0.489898990 0.97979798 -321.6979
## 99  0.494949495 0.98989899 -323.5435
## 100 0.500000000 1.00000000 -325.4066
```

```r
ggplot(total,aes(mean_value,lambda,z=z)) +
  geom_density_2d()  +
  geom_vline(xintercept = meanvalue_max) +
  geom_hline(yintercept = lambda_max) +
  geom_text(label = "(0.2712177,0.1333494)", x=0.2712177, y=0.1333494)
```




```r
plot1 <- ggplot(total,aes(mean_value,lambda,z=z)) +
  geom_density_2d() +
  geom_point(x = meanvalue_max, y = lambda_max)

plot1 + annotate("text", x = meanvalue_max, y = lambda_max,
           label = expression( group("(",list(theta[max] , lambda[max]),")")))
```

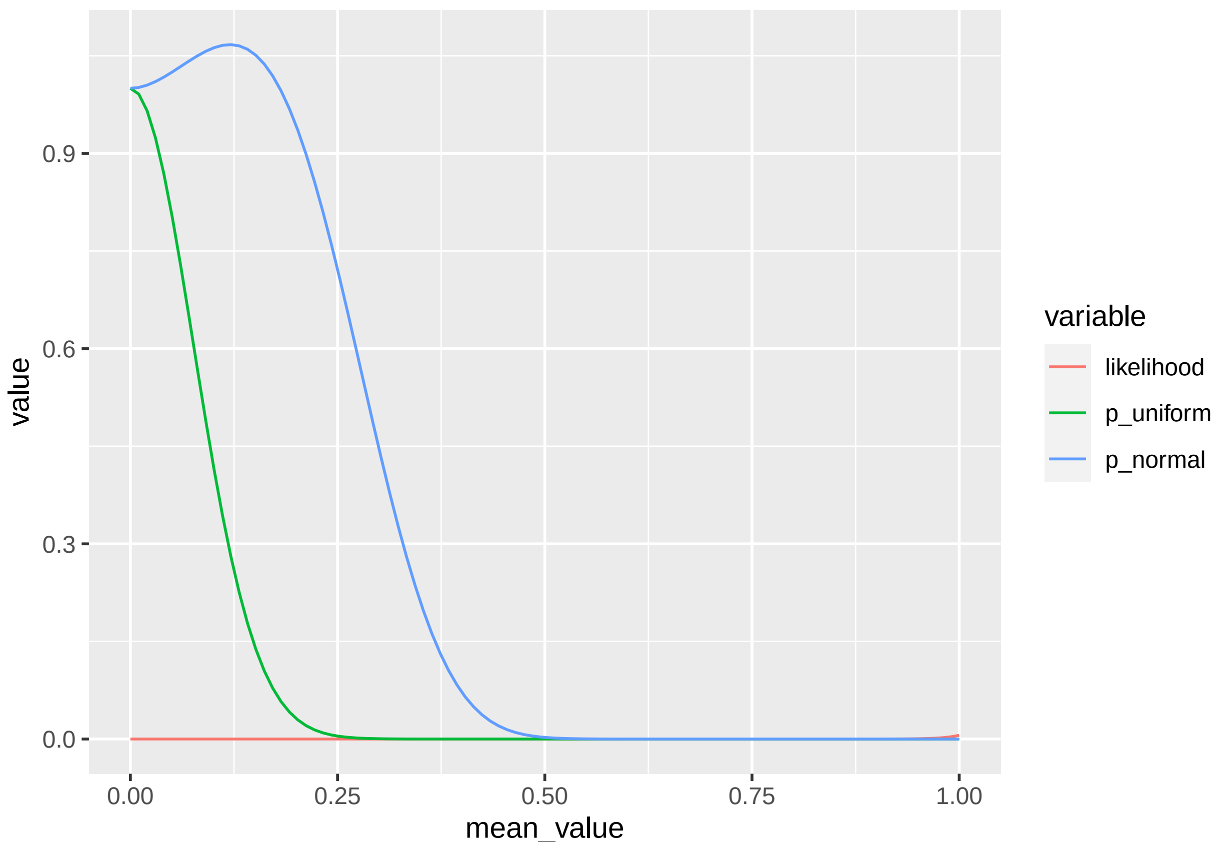
```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```



1d

```
p_uni <- function(mean_value,lambda){
p_uniform <- exp (lambda*sum(rnd)*mean_value-length(rnd)*
                           ((mean_value)**2))
return(p_uniform)
}

p_normal <- function(mean_value,lambda,lambda_0){
  mean_0=0
  L <- lambda_0+length(rnd)*lambda
  M <- (lambda_0*mean_0+lambda*sum(rnd))/(L)
  p_normal <- exp(L*M*mean_value - 0.5*L*(mean_value**2))
}

my_function2<-function(mean_value,lambda){
  final <- 0
  for (variable in rnd) {
    c <- (variable-mean_value)**2
    final <- final+c
  }

  likelihood <- ((sqrt(lambda/2*3.14))**length(rnd))*exp(0.5*lambda)*final
  return(likelihood)
```

```
}

lambda_0 <- seq(0.1,100,length=100)


lvt <- my_function2(mean_value,lambda)
pn <- p_normal(mean_value,lambda,lambda_0 = lambda_0)
pu <- p_uni(mean_value = mean_value,lambda = lambda)
id <- 0:99
final_data <- data.frame(likelihood = lvt, p_uniform = pu, p_normal = pn,
                         mean_value = mean_value)

df_1 <- melt(final_data, id.vars="mean_value")


ggplot(data=df_1, aes(x=mean_value,y=value,col=variable)) +
  geom_line()
```



Question2

2a

Hypotheses: Null Hypothesis,

$$H_0 : \mu = 0.12$$

Alternate Hypothesis,

$$H_1 : \mu > 0.12$$

2b

Reject H_0 if $z >= z_\alpha$ Since $\alpha = 0.01$, from the z table we can get \$z_{0.01} = 2.33 \$ Therefore rejection region is: Reject H_0 if $z >= 2.33$

2c

Hypothesis testing:

Hypotheses: Null Hypothesis, $H_0 : \mu = 0.12$ Alternate Hypothesis, $H_1 : \mu > 0.12$

Test Statistic:

$$z_{obs} = \frac{\overline{y} - \mu_0}{s/\sqrt{n}}$$

In the problem, we have

$$\overline{y} = 0.135$$

$$s = 0.03$$

$$\mu_0 = 0.12$$

$$n = 30$$

Substituting this on the above formule we get the test statistic

$$z_{obs} = \frac{0.135 - 0.12}{0.03/\sqrt{30}} = 2.74$$

Rejection Region:

$$z_{obs} = 2.74 > z_{\alpha=0.01} = 2.33$$

Therefore we reject H_0.

This means that there is sufficient evidence to conclude the alternate hypothesis that mean ozone levels in air currents over New England exceeds the federal ozone standard of 0.12 ppm.

2d

p-value $= p(z >= z_{obs}) = p(z >= 2.74) = 1 - p(z < 2.74) = 1 - 0.9969 = 0.0031$

Because p-value $p = 0.0031 < \alpha = 0.01$ we reject the null hypothesis H_0. This is consistent with our result in part c.

2e

Assumptions concerning the distribution of the random variable X, ozone level in the air:

1. The data is continuous and not discrete
2. The data is a simple random sample
3. The data in the population is normally distributed
4. The population standard deviation is known

Question3

3a

```
data("BostonHousing")
df_2 = BostonHousing

bh = lm(crim ~ .,data = df_2)
lm.betas <- bh$coefficients
summary(bh)
```

```
##
## Call:
## lm(formula = crim ~ ., data = df_2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas1        -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## b            -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

We can see from the t-value that the model is significant. These are predictors can be rejected for $H_0 : \beta_j = 0$ at $\alpha = 0.05$
(ii)zn
(iii)dis
(iv)rad
(v)b
(vi)medv

Null hypothesis can be rejected for features whose Pr(>|t|) < 0.05. So from the above table we can reject null hypothesis for zn, dis, rad, b, medv

3b

```r
y <- df_2$crim
X <- as.matrix(df_2[-1])
int <- rep(1, length(y))
X <- cbind(int, X)
X <- matrix(as.numeric(X),ncol = ncol(X))
my.lm <- function(y,X){
betas <- solve(t(X) %*% X) %*% t(X) %*% y
return (betas)
}
my.lm(y,X)
```

```
##                [,1]
##  [1,]  17.033227523
##  [2,]   0.044855215
##  [3,]  -0.063854824
##  [4,]  -0.749133611
##  [5,] -10.313534912
##  [6,]   0.430130506
##  [7,]   0.001451643
##  [8,]  -0.987175726
##  [9,]   0.588208591
## [10,]  -0.003780016
## [11,]  -0.271080558
## [12,]  -0.007537505
## [13,]   0.126211376
## [14,]  -0.198886821
```

```r
#Comparision
results <- data.frame(our.results=my.lm(y,X), lm.results=lm.betas)
print(results)
```

```
##               our.results    lm.results
## (Intercept)  17.033227523  17.033227523
## zn            0.044855215   0.044855215
## indus        -0.063854824  -0.063854824
## chas1        -0.749133611  -0.749133611
## nox         -10.313534912 -10.313534912
## rm            0.430130506   0.430130506
## age           0.001451643   0.001451643
## dis          -0.987175726  -0.987175726
## rad           0.588208591   0.588208591
## tax          -0.003780016  -0.003780016
## ptratio      -0.271080558  -0.271080558
## b            -0.007537505  -0.007537505
## lstat         0.126211376   0.126211376
## medv         -0.198886821  -0.198886821
```

```r
#MSE
beta = my.lm(y,X)
```

```r
int <- rep(1, length(y))
#Z = cbind(int,X)
#Z <- matrix(as.numeric(Z),ncol = ncol(Z))
pred   = X %*% beta
MSE_own = mean((y - pred)^2)
MSE_lm = mean(bh$residuals^2)
results_mse <- data.frame(our.result=MSE_own, lm.result=MSE_lm)
print(results_mse)
```

```
##   our.result lm.result
## 1   40.31607  40.31607
```

3c

```r
train = tail(df_2,-10)
test = head(df_2,10)
ytrain = train$crim
ytest = test$crim


Xtest =  as.matrix(test[-1])
int2 = rep(1, length(ytest))
Xtest = cbind(int2,Xtest)
Xtest <- matrix(as.numeric(Xtest),ncol = ncol(Xtest))
xtest = head(Xtest, 1)

Xtrain =  as.matrix(train[-1])
int2 = rep(1, length(ytrain))
Xtrain = cbind(int2,Xtrain)
Xtrain = matrix(as.numeric(Xtrain),ncol = ncol(Xtrain))

my.predict <- function(Xtrain, ytrain, Xtest){
  n = length(ytrain)
  #lm.model <- lm(y ~ x)
  p = ncol(Xtest)
 # y.fitted <- lm.model$fitted.values # Extract the fitted values of y
  beta = my.lm(ytrain,Xtrain)
  y.fitted =  Xtrain %*% beta
#pred.y <- b1 * pred.x + b0

  pred.y =  Xtest%*%beta
  return(pred.y)
}
predTest = my.predict(Xtrain, ytrain,Xtest)
RMSE = sqrt((1/10)*sum((predTest - ytest)^2))
print(RMSE)
```

```
## [1] 2.896944
```

3d

```r
summary(bh)
```

```
##
## Call:
## lm(formula = crim ~ ., data = df_2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas1        -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## b            -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

Pvalue is therefore lower than significance value (0.05). Thus, the null hypothesis that a model with no independent variables would adequately describe the data can be discarded. We can make the conclusion that independent variables help models fit better.

3e

```r
bh2 = lm(crim ~ zn+dis+rad+b+medv,data = df_2)
summary(bh2)
```

```
##
## Call:
## lm(formula = crim ~ zn + dis + rad + b + medv, data = df_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -10.553  -1.869  -0.358   0.839  75.744
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.919933   1.778986   4.452 1.05e-05 ***
## zn           0.051799   0.017329   2.989 0.002935 **
## dis         -0.672189   0.202939  -3.312 0.000992 ***
## rad          0.472306   0.042102  11.218  < 2e-16 ***
## b           -0.008211   0.003615  -2.271 0.023562 *
## medv        -0.174219   0.036295  -4.800 2.10e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.473 on 500 degrees of freedom
## Multiple R-squared:  0.4393, Adjusted R-squared:  0.4337
## F-statistic: 78.34 on 5 and 500 DF,  p-value: < 2.2e-16
```

```
anova(bh,bh2)
```

```
## Analysis of Variance Table
##
## Model 1: crim ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##     ptratio + b + lstat + medv
## Model 2: crim ~ zn + dis + rad + b + medv
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    492 20400
## 2    500 20951 -8   -550.61 1.6599 0.1057
```

The F statistic is 1.6599 and the pvalue is 0.1507. pvalue is greater that significance level (0.05) so we need to accept the Null hypothesis for the partial F test that coefficients of the features of reduced model are 0.