

Linear Regression

Fall 2022, MATH8050: Homework 4
Your Name, Section XXX

Due September 28, 12:00 PM

General instructions for homeworks: Please follow the uploading file instructions according to the syllabus. Each answer must be supported by written statements as well as any code used. Your code must be completely reproducible and must compile. For writing mathematical expressions in R Markdown, refer to the [homework template](#) posted on Canvas, a [30-minute tutorial](#), or [LaTeX/Mathematics](#).

Advice: Start early on the homeworks and it is advised that you not wait until the last day. While the professor and the TA's check emails, they will be answered in the order they are received and last minute help will not be given.

No late homeworks will be accepted.

R Working Environment

Please load all the packages used in the following R chunk before the function `sessionInfo()`

```
# load packages
library(tidyverse)
library(patchwork)
library(mlbench)

sessionInfo()

## R version 4.1.3 (2022-03-10)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.3.1
##
## Matrix products: default
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] mlbench_2.1-3   scico_1.3.1     patchwork_1.1.2 forcats_0.5.1
## [5] stringr_1.4.0   dplyr_1.0.8     purrr_0.3.4    readr_2.1.2
## [9] tidyr_1.2.0     tibble_3.1.6    ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
## [1] tidyselect_1.1.2 xfun_0.30       haven_2.5.0     colorspace_2.0-3
## [5] vctrs_0.4.1      generics_0.1.2  htmltools_0.5.2 yaml_2.3.5
## [9] utf8_1.2.2       rlang_1.0.2     pillar_1.7.0    glue_1.6.2
## [13] withr_2.5.0      DBI_1.1.2       dbplyr_2.1.1    modelr_0.1.8
```

```
## [17] readxl_1.4.0      lifecycle_1.0.1  munsell_0.5.0    gtable_0.3.0
## [21] cellranger_1.1.0  rvest_1.0.2      evaluate_0.15     knitr_1.38
## [25] tzdb_0.3.0        fastmap_1.1.0    fansi_1.0.3       broom_0.8.0
## [29] backports_1.4.1   scales_1.2.0     jsonlite_1.8.0    fs_1.5.2
## [33] hms_1.1.1         digest_0.6.29    stringi_1.7.6     grid_4.1.3
## [37] cli_3.2.0         tools_4.1.3      magrittr_2.0.3    crayon_1.5.1
## [41] pkgconfig_2.0.3   ellipsis_0.3.2   xml2_1.3.3        reprex_2.0.1
## [45] lubridate_1.8.0   assertthat_0.2.1 rmarkdown_2.13    httr_1.4.2
## [49] rstudioapi_0.13  R6_2.5.1         compiler_4.1.3
```

Total points on assignment: 10 (reproducibility) + 40 (Q1) + 25 (Q2) + 25 (Q3)

Reproducibility component: 10 points.

1. (40pts total, equally weighted) The normal distribution $\mathcal{N}(\theta, \sigma^2)$ with mean $\theta \in \mathbb{R}$ and variance $\sigma^2 > 0$ - (standard deviation $\sigma = \sqrt{\sigma^2}$) has p.d.f.

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

for $x \in \mathbb{R}$. Assume that the observations y_1, \dots, y_n are drawn from $\mathcal{N}(\theta, \lambda^{-1})$, i.e.,

$$y_1, \dots, y_n | \theta, \lambda \stackrel{iid}{\sim} \mathcal{N}(\theta, \lambda^{-1}),$$

where the inverse variance $\lambda = 1/\sigma^2$ is called *precision*. Assume that λ is known and fixed. Answer the following questions.

- a. (Normal-Uniform Model) Assume the uniform prior $p(\theta) \propto 1$ on θ over the entire real line. Derive the posterior distribution of θ given $y_{1:n} := (y_1, \dots, y_n)^\top$.

Solution

$$\begin{aligned} p(\theta | y_{1:n}) &\propto \prod_{i=1}^n \mathcal{N}(y_i | \theta, \lambda^{-1}) \times 1 \\ &\propto \left(\frac{\lambda}{2\pi}\right)^{n/2} \exp\left\{-\frac{1}{2}\lambda \sum_i (y_i - \theta)^2\right\} \\ &\propto \exp\left\{-\frac{1}{2}\lambda \sum_i (y_i - \bar{y} + \bar{y} - \theta)^2\right\} \\ &\propto \exp\left\{-\frac{1}{2}\lambda \sum_i (y_i - \bar{y})^2\right\} \exp\left\{-\frac{1}{2}\lambda \sum_i (\bar{y} - \theta)^2\right\} \\ &\propto \exp\left\{-\frac{1}{2}\lambda \sum_i (\bar{y} - \theta)^2\right\} \\ &\propto \exp\left\{-\frac{n\lambda}{2}(\theta - \bar{y})^2\right\} \end{aligned}$$

This implies that

$$\theta | y_{1:n} \sim \mathcal{N}(\bar{y}, (n\lambda)^{-1})$$

- b. (Normal-Normal Model) Assume the normal distribution prior $\theta \sim \mathcal{N}(\mu_0, \lambda_0^{-1})$ on θ , where μ_0, λ_0 are fixed and known quantities. Derive the posterior distribution of θ given $y_{1:n}$.

Solution

For any x and λ ,

$$\begin{aligned}\mathcal{N}(x | \theta, \lambda^{-1}) &= \sqrt{\frac{\lambda}{2\pi}} \exp \left\{ -\frac{1}{2} \lambda (x - \theta)^2 \right\} \\ &\propto_{\theta} \exp \left\{ -\frac{1}{2} \lambda (x^2 - 2x\theta + \theta^2) \right\} \\ &\propto_{\theta} \exp \left\{ \lambda x \theta - \frac{1}{2} \lambda \theta^2 \right\}.\end{aligned}\tag{1}$$

Direct calculation yields that

$$\begin{aligned}p(\theta | y_{1:n}) &\propto p(\theta) p(y_{1:n} | \theta) \\ &\propto \mathcal{N}(\theta | \mu_0, \lambda_0^{-1}) \prod_{i=1}^n \mathcal{N}(y_i | \theta, \lambda^{-1}) \\ &\stackrel{(a)}{\propto} \exp \left\{ \lambda_0 \mu_0 \theta - \frac{1}{2} \lambda_0 \theta^2 \right\} \exp \left\{ \lambda \left(\sum_{i=1}^n y_i \right) \theta - \frac{1}{2} n \lambda \theta^2 \right\} \\ &\propto \exp \left\{ (\lambda_0 \mu_0 + \lambda \sum_i y_i) \theta - \frac{1}{2} (\lambda_0 + n \lambda) \theta^2 \right\} \\ &= \mathcal{N} \left(\theta | M, L^{-1} \right),\end{aligned}$$

where $L = \lambda_0 + n \lambda$ and $M = \frac{\lambda_0 \mu_0 + n \lambda \bar{y}}{L}$. Step (a) uses (1).

The normal distribution is, itself, a conjugate prior for the mean of a normal distribution with known precision.

- c. Simulate $n = 100$ data points from $\mathcal{N}(0, 9)$. Then treat these data points as your observations and assume that these observations are iid samples from the model $\mathcal{N}(\theta, \lambda^{-1})$. Find the maximum likelihood function of θ and λ , and plot a 2d contour plot (see `?geom_contour()` in R) of the log-likelihood as a function of θ and λ . Make sure that you label the point $(\theta_{mle}, \lambda_{mle})$ at which you obtain the mle of μ and λ . Please also set your seed number as `set.seed(123)` when simulating the observations to ensure your results are numerically reproducible. See `?set.seed()` in R for help.

Solution

The likelihood of $y_{1:n}$ given θ is

$$\begin{aligned}L(\theta; y_{1:n}) &= p(y_{1:n} | \theta) = \prod_{i=1}^n \mathcal{N}(y_i | \theta, \lambda^{-1}) \\ &= \left(\frac{\lambda}{2\pi} \right)^{n/2} \exp \left\{ -\frac{\lambda}{2} \sum_i (y_i - \theta)^2 \right\}.\end{aligned}$$

The log-likelihood is

$$\log L(\theta; y_{1:n}) = \frac{n}{2} \log(p 2\pi \lambda) - \frac{1}{2} \lambda \sum_i (y_i - \theta)^2.$$

Taking the partial derivatives w.r.t. θ, λ and setting them to zero will give

$$\theta_{mle} = \bar{y}$$

$$\lambda_{mle} = \frac{n}{\sum_{i=1}^n (y_i - \theta_{mle})^2}.$$

The second derivative of $\log L(\theta; y_{1:n})$ w.r.t. θ is $-n\lambda < 0$, indicating that θ_{mle} is the unique MLE, and so is λ_{mle} . (Take 2 points off if there is no justification on the uniqueness of the MLE.)

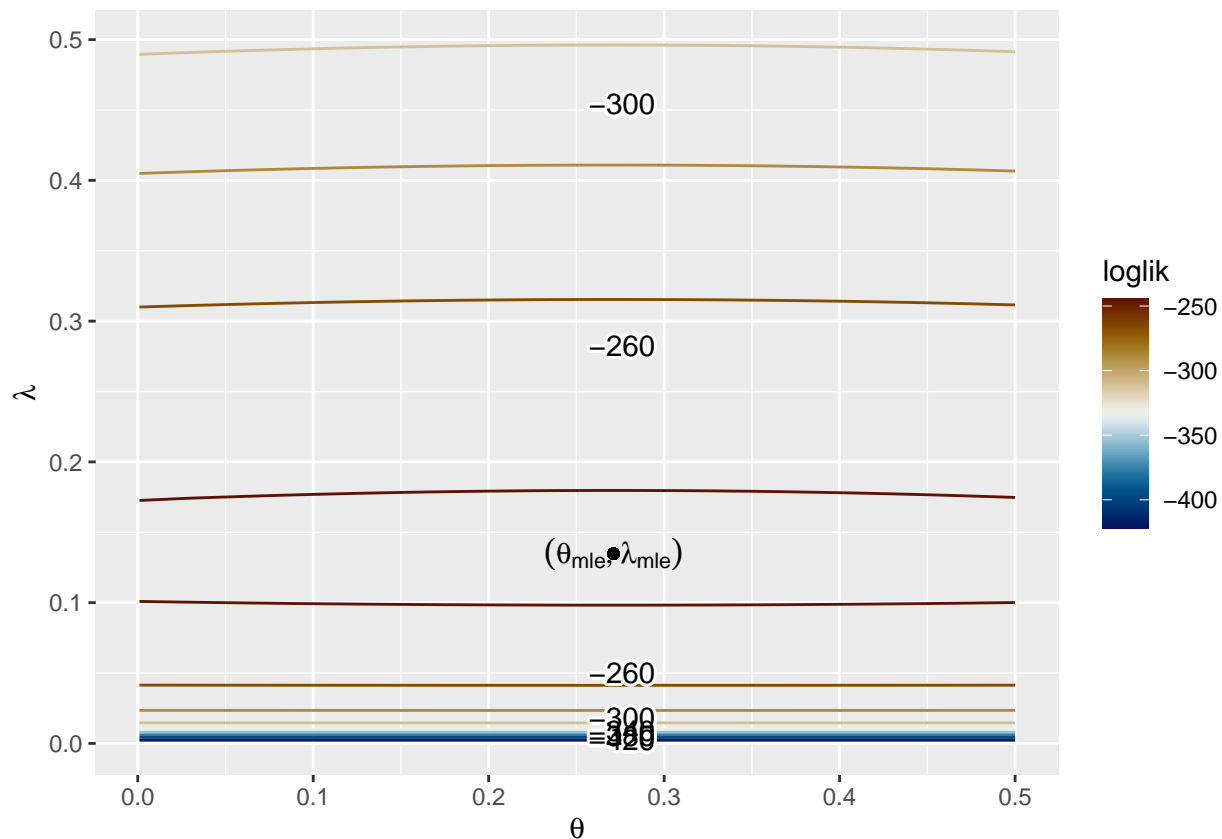
```
# on different R version and operating systems, the random
# numbers maybe different. But the MLE should be a unique solution in the contour plot
set.seed(123)
n = 100
y = rnorm(n, 0, sd=3)
lambda = 1/9
theta.mle = mean(y)
lambda.mle = n / sum((y-theta.mle)^2)

loglik.norm = function(theta,lambda){
  result = 0.5*n*log(lambda) - 0.5*lambda*sum((y-theta)^2) -0.5*n*log(2*pi)
  return(result)
}

theta.x = seq(1e-3,.5,length.out=50)
lambda.y = seq(1e-3,.5,length.out=50)
df = expand_grid(theta=theta.x, lambda=lambda.y)
df$z = rep(NA, nrow(df))
for(i in 1:nrow(df)){
  df$z[i] = loglik.norm(theta=df$theta[i], lambda=df$lambda[i])
}

label.mle = expression( group( "(",list(theta[mle] , lambda[mle]),")" ) )

df %>%
  ggplot(aes(x=theta, y=lambda)) +
  geom_contour(aes(z=z, color=stat(level)), bins = 10) +
  metR::geom_text_contour(aes(z=z), stroke=0.15)+
  scico::scale_color_scico(palette="vik",name="loglik") +
  geom_point(aes(x=theta.mle, y=lambda.mle)) +
  annotate(geom="text", x=theta.mle, y=lambda.mle, label=label.mle) +
  labs(x=expression(theta), y=expression(lambda))
```



the MLE should maximize the (log)-likelihood function globally, and any figure does not respect this fact is wrong. At least 5 points should be taken off.

- d. Continue from (c) above. After generating the observations as in (c), assume that these observations are iid samples from the model $\mathcal{N}(\theta, 9)$, where we pretend that we do not know θ . Plot the likelihood, the posterior density of θ in Part 1.a and 1.b, where $\mu_0 = 0$ and λ_0 takes values in $\{0.1, 100\}$. Make sure that you use different linetypes and/or colors to differentiate these four curves.

```
theta = seq(1e-3, 0.5, length.out=1000)
log.likelihood = 0.5*n*log(lambda) - 0.5*lambda*sum((y-theta)^2) - 0.5*n*log(2*pi)
ybar = mean(y)
log.post1 = 0.5*log(n*lambda/(2*pi)) - 0.5*n*lambda*(theta-ybar)^2
mu0 = 0
lambda0 = 0.1
L = lambda0 + n*lambda
M = (lambda0*mu0 + n*lambda*ybar)/L
log.post2 = 0.5*log(L/(2*pi)) - 0.5*L*(theta-M)^2
mu0 = 0
lambda0 = 100
L = lambda0 + n*lambda
M = (lambda0*mu0 + n*lambda*ybar)/L
log.post3 = 0.5*log(L/(2*pi)) - 0.5*L*(theta-M)^2

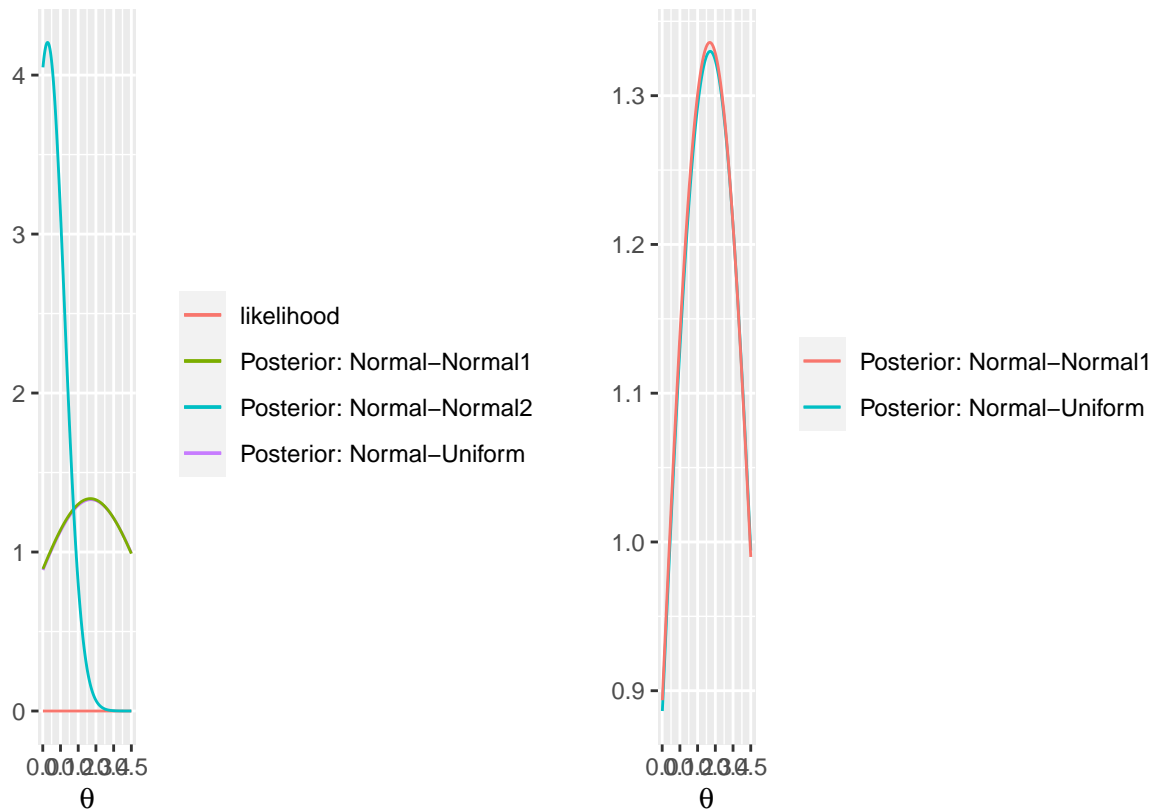
dat = data.frame(theta=theta,
                  likelihood=exp(log.likelihood),
                  posterior1=exp(log.post1),
                  posterior2=exp(log.post2),
```

```

    posterior3=exp(log.post3)
  )
g1 = dat %>% ggplot() +
  geom_line(aes(x=theta,y=likelihood, color="likelihood")) +
  geom_line(aes(x=theta,y=posterior1, color="Posterior: Normal-Uniform")) +
  geom_line(aes(x=theta,y=posterior2, color="Posterior: Normal-Normal1")) +
  geom_line(aes(x=theta,y=posterior3, color="Posterior: Normal-Normal2")) +
  labs(x=expression(theta), y="") +
  theme(legend.title=element_blank())

g2 = dat %>% ggplot() +
  #geom_line(aes(x=theta,y=likelihood, color="likelihood")) +
  geom_line(aes(x=theta,y=posterior1, color="Posterior: Normal-Uniform")) +
  geom_line(aes(x=theta,y=posterior2, color="Posterior: Normal-Normal1")) +
  #geom_line(aes(x=theta,y=posterior3, color="Posterior: Normal-Normal2")) +
  labs(x=expression(theta), y="") +
  theme(legend.title=element_blank())
g1 + g2

```



$\lambda_0 = 100$

indicates a strong prior, thus the resulting posterior in the Normal-Normal model is quite informative. In contrast, $\lambda_0 = 0.1$ indicates a non-informative prior, and the resulting posterior in the Normal-Normal model is almost the same as the posterior in the Normal-Uniform model as shown in the right panel.

- (20pts, equally weighted) **Review of basic linear regression:** Ozone is a component of smog that can injure sensitive plants even at low levels. In 1979 a federal ozone standard of 0.12 ppm was set. It is thought that the ozone level in air currents over New England exceeds this level. To verify this contention, air samples are obtained from 30 monitoring stations set up across the region.

- a. Set up the appropriate null and alternative hypotheses for verifying the contention.

Solution:

Let u be the population. In this case, the null hypothesis and alternative hypothesis are

$$H_0 : u = 0.12 \quad \text{versus} \quad H_1 : u > 0.12.$$

- b. What is the rejection region for an $\alpha = 0.01$ level test based on a sample of size 30?

Solution:

The test statistic is

$$T = \frac{\bar{Y} - 0.12}{S/\sqrt{30}}$$

and the rejection region for testing $H_0 : \mu = 0.12$ versus H_1 is given by:

$$RR = \{T > t_{29,0.01}\} = \{T > 2.462\}.$$

- c. When the data are analyzed, a sample mean of 0.135 and a sample standard deviation of 0.03 are obtained. Use these data to test H_0 . Can H_0 be rejected at the $\alpha = 0.01$ level using the rejection region approach? What does this mean in a practical sense?

Solution:

The observed T is $T_0 = \frac{0.135 - 0.12}{0.03/\sqrt{30}} = 2.738613$, so $T_0 \in RR = \{T > 2.462\}$, then we reject H_0 , which means that the ozone level in air currents over New England exceeds the federal ozone standard 0.12 ppm set in 1979 with the significance level of 0.01.

- d. Find the p -value for the test. Is the conclusion based on the p -value consistent with that obtained from part (c)?

Solution:

$$p\text{-value} = P(T > T_0) = P(T > 2.738613) = 1 - F_{t_{29}}(2.738613) = 1 - 0.9948 = 0.0052.$$

Yes. The conclusion based on p -value is consistent with which was obtained from part (c).

- e. What assumptions are you making concerning the distribution of the random variable X , ozone level in the air?

Solution:

We assume that the data are normally distributed or the sample size of the data is large (Here $n=30$).

3. (30pts total) This problem involves the Boston housing dataset `BostonHousing` in the R package `mlbench`. The goal is to predict per capita crime rate using the other variables in this dataset. In other words, per capital crime rate is the response, and the other variables are the predictors.

- a. (4pts) Fit a multiple linear regression model to predict the response using all of the predictors. Describe and back up your results. For which predictors can we reject the null hypothesis $H_0 : \beta_j = 0$ at significance level $\alpha = 0.05$? You should use `lm()` in R to fit the model.

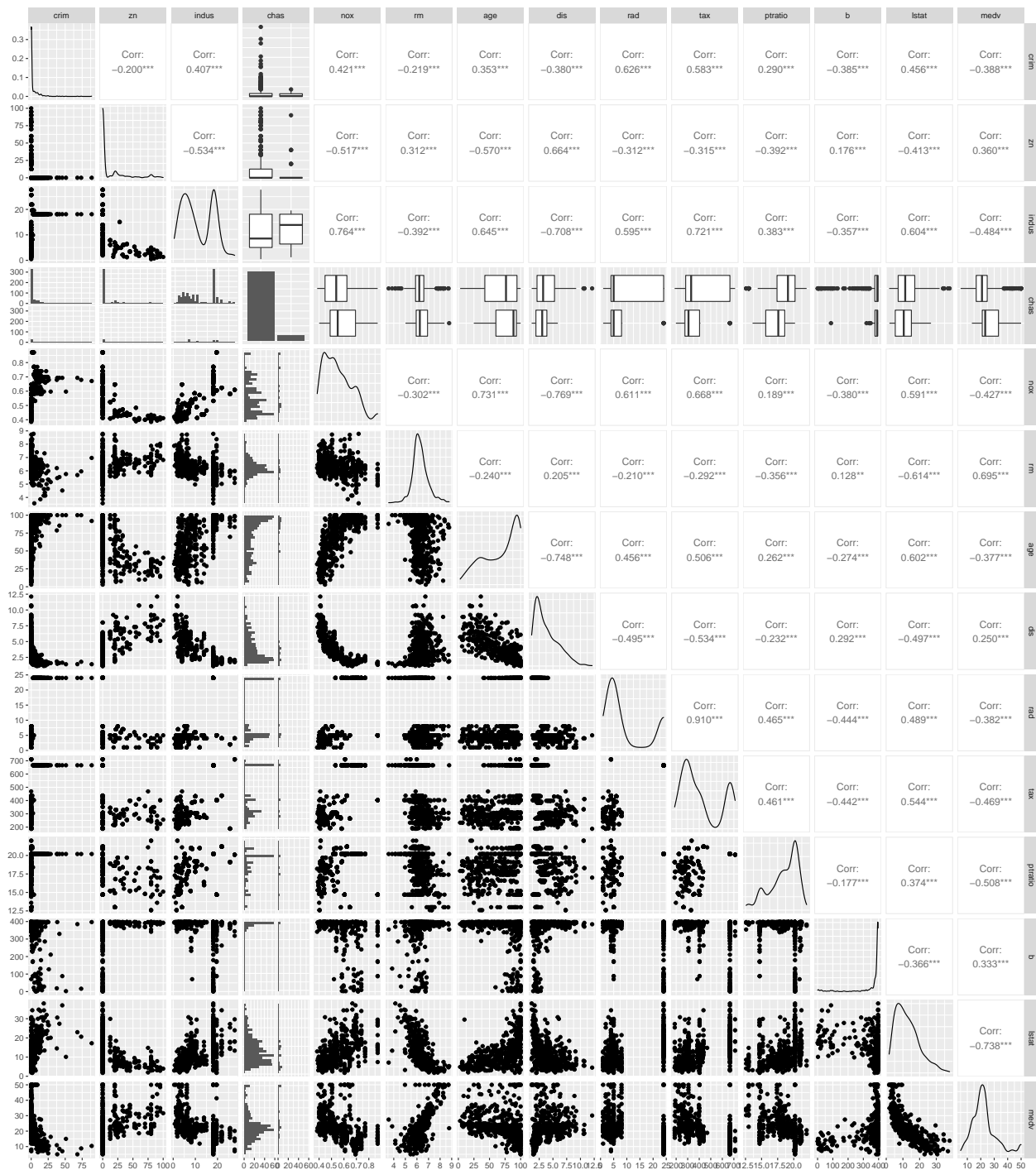
```
data("BostonHousing")
head(BostonHousing)
```

```
##      crim zn  indus chas   nox   rm  age   dis rad tax ptratio    b lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

```
# plot the data first before using any statistical models
GGally::ggpairs(BostonHousing)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Then we run the multiple linear regression model using the R code below.

```
mod = lm(crim~., data=BostonHousing)
summary(mod)

##
## Call:
## lm(formula = crim ~ ., data = BostonHousing)
##
## 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
```

From the output above, it is clearly that we can reject the null hypotheses for predictors `indus`, `chas1`, `nox`, `rm`, `age`, `tax`, `ptratio`, `lstat` at significance level $\alpha = 0.05$.

- b. (8pts) In part (a), you have used the built-in R function `lm()` to get the parameter estimates. Now you need to write your own R function called `my.lm()` to get the parameter estimates. In your function, you need to compute the parameter estimates of the regression coefficients β using least squares methods and compute the mean squared error (MSE) as an unbiased estimator of σ^2 . Verify your results (i.e., parameter estimates) with the output from `lm()`, and you do not need to replicate the output from `lm()` with your function `my.lm()`.

Define the `my.lm()` function as follows

```
my.lm = function(formula=~1, response, predictor){

  df = as.data.frame(predictor)
  X = model.matrix(formula, df)
  beta.hat = solve(t(X)%*%X, t(X)%*%response)
  colnames(beta.hat) = "Estimate"

  p = ncol(X)
  n = nrow(X)
  MSE = (sum(response*response) - t(response)%*%(X%*%beta.hat))/(n-p)

  out = list(coef=beta.hat, MSE=drop(MSE))
  return(out)
}
```

Verify solutions

```

X = BostonHousing[, -1]
y = BostonHousing$crim
mod2 = my.lm(y~., response=y, predictor=X)
# verify estimates for regression coefficients
sum((c(mod2$coef)-mod$coef)^2)

## [1] 6.758536e-23

# verify estimates for variance parameter, i.e., MSE
sqrt(mod2$MSE)

## Estimate
## 6.439198

```

- c. (9pts) With your written function from Part (b), write your own function called `my.predict()` to generate predictive mean and predictive intervals. Suppose now we held out the first 10 observations for predictive checking and use the remaining observations to train the model to get parameter estimates using your written function from Part (b). You then compute the root mean squared prediction error (RMSE) over these 10 held-out observations, where the RMSE is defined as

$$RMSE = \sqrt{\frac{1}{n_*} \sum_{i=1}^{n_*} (y_i - \hat{y}_i)^2},$$

where n_* is the number of testing data points, y_i is the i th testing response corresponding to predictor vector \mathbf{x}_i , \hat{y}_i is the corresponding predictive mean for the i th data point.

Define the prediction function:

```

my.predict = function(formula, response, predictor, xnew, level=0.95){

  df = as.data.frame(predictor)
  X = model.matrix(formula, df)
  beta.hat = solve(t(X)%*%X, t(X)%*%response)
  colnames(beta.hat) = "Estimate"
  p = ncol(X)
  n = nrow(X)
  MSE = c((sum(response*response) - t(response)%*%(X%*%beta.hat))/(n-p))

  if(is.vector(xnew)){
    xnew = matrix(xnew, 1,p)
  }else if(!is.matrix(xnew)){
    xnew = as.matrix(xnew)
  }

  yhat = drop(xnew%*%beta.hat)
  quad = xnew%*%solve(t(X)%*%X)%*%t(xnew)
  quad.vec = c(diag(quad))

  pred.lower = yhat - qt(level, df=n-p) * sqrt(MSE*(1+quad.vec))
  pred.upper = yhat + qt(level, df=n-p) * sqrt(MSE*(1+quad.vec))

  pred.response = cbind(yhat, pred.lower, pred.upper)
  colnames(pred.response) = c("fit", "lwr", "upr")
}

```

```

    return(pred=pred.response)
}

```

Compute RMSE for testing or held-out data

```

df.new = BostonHousing[1:10, ]
df.obs = BostonHousing[-c(1:10), ]
predictor = df.obs[,-1]
response = df.obs[,1]
xnew = df.new[,-1]
x.df = xnew
xnew$chas = as.numeric(xnew$chas)
xnew = as.matrix(xnew)
xnew = cbind(1, xnew)

pred.mod = my.predict(response~., response,
                      predictor, xnew, level=0.95)

RMSE = sqrt(mean((pred.mod[,1]-df.new[,1])^2))
print(RMSE)

```

```
## [1] 3.283333
```

- d. (4pts) Use R to perform F test to test $H_0 : \beta_0 = \dots = \beta_{p-1}$ for $p = 14$. Which predictors are excluded at $\alpha = 0.05$ level of significance.

```
anova(lm(crim~., data=BostonHousing))
```

```
## Analysis of Variance Table
##
## Response: crim
##          Df Sum Sq Mean Sq F value    Pr(>F)
## zn         1  1501.5   1501.5   36.2140 3.457e-09 ***
## indus       1  4689.3   4689.3  113.0959 < 2.2e-16 ***
## chas        1   247.8    247.8    5.9761 0.0148509 *
## nox         1  1270.8   1270.8   30.6476 5.041e-08 ***
## rm          1   138.5    138.5    3.3406 0.0681959 .
## age         1   165.5    165.5    3.9916 0.0462773 *
## dis         1   300.1    300.1    7.2372 0.0073834 **
## rad         1  7238.3   7238.3  174.5724 < 2.2e-16 ***
## tax         1     3.3     3.3    0.0798 0.7776270
## ptratio     1     7.3     7.3    0.1756 0.6753646
## b           1   455.3   455.3   10.9798 0.0009890 ***
## lstat       1   497.7   497.7   12.0042 0.0005772 ***
## medv        1   447.9   447.9   10.8012 0.0010868 **
## Residuals 492 20399.9    41.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The excluded predictors at $\alpha = 0.05$ are `rm`, `tax`, `ptratio`.

- e. (5pts) Use R to perform partial F test for the null hypothesis without predictors `indus`, `chas1`, `nox`, `rm`, `age`, `tax`, `ptratio`, `lstat` versus the alternative hypothesis with the full model at the significance level $\alpha = 0.05$.

```

model1 = lm(crim~zn+dis+rad+b+medv, data=BostonHousing)
model2 = lm(crim~., data=BostonHousing)
anova(model1, model2)

```

```

## Analysis of Variance Table
##
## Model 1: crim ~ zn + dis + rad + b + medv
## Model 2: crim ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
##          ptratio + b + lstat + medv
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      500 20950
## 2      492 20400   8    550.61 1.6599 0.1057

```

It is clearly from the R output that the predictors `indus`, `chas1`, `nox`, `rm`, `age`, `tax`, `ptratio`, `lstat` do not contribute to the linear relationship between response `crim` and all the predictors at the significance level $\alpha = 0.05$.