Tutorial Problems

Question 1

For any matrix $\mathbf{C}$ of dimension $m \times m$, the trace of $\mathbf{C}$ is the sum of all the diagonal elements of $\mathbf{C}$, i.e $\text{trace}(\mathbf{C}) = \sum_{i=1}^{m} c_{ii}$.

(a) Using matrix multiplication rules, prove that for any matrix $\mathbf{A}$ and $\mathbf{B}$ of dimension $n \times p$ and $p \times n$ respectively, the two matrices $\mathbf{AB}$ and $\mathbf{BA}$ always have the same trace, i.e

$$\text{trace}(\mathbf{AB}) = \text{trace}(\mathbf{BA}).$$

Hint: denote elements of $\mathbf{A}$ as $a_{ij},\ i = 1, \ldots, n,\ j = 1, \ldots, p$ and elements of $\mathbf{B}$ as $b_{ij}\ i = 1, \ldots, p,\ j = 1, \ldots, n$, find the formula for each diagonal element of $\mathbf{AB}$ and $\mathbf{BA}$ and sum them up.

(b) The hat matrix that is used extensively in multiple regression model is defined to be $\mathbf{H} = \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top$, where $\mathbf{X}$ is the design (full-ranked) matrix with dimension $n \times p$ and $n > p$. Using the result in part (a), prove that $\text{trace}(\mathbf{H}) = p$.

Computer problem

In this tutorial, we will examine how to specify arguments (inputs) for the **lm()** function in R, especially if we want to fit multiple linear models with a large number of covariates. First, you can always access the R documentation for this command by using

```
?lm
help(lm)
```

As you can see, the **lm()** command has a number of arguments, but the only **required** input is a *formula* object. However, especially when the result from this command is used as input for another function, for example, *predict()*, it is the best to specify the *data* input as well. Following we will demonstrate it with the mtcars dataset.

```
data(mtcars)
```

First, let's assume we want to fit the linear model of fuel consumptions (mpg) on three covariates (disp, hp, drat). The first way to input the lm command is the following:

```
fit1 <- lm(formula = mtcars$mpg ~ mtcars$disp + mtcars$hp + mtcars$drat)
summary(fit1)
```

```
##
## Call:
## lm(formula = mtcars$mpg ~ mtcars$disp + mtcars$hp + mtcars$drat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```



```
## -5.1225 -1.8454 -0.4456  1.1342  6.4958
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.344293   6.370882   3.036  0.00513 **
## mtcars$disp -0.019232   0.009371  -2.052  0.04960 *
## mtcars$hp   -0.031229   0.013345  -2.340  0.02663 *
## mtcars$drat  2.714975   1.487366   1.825  0.07863 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.008 on 28 degrees of freedom
## Multiple R-squared:  0.775,  Adjusted R-squared:  0.7509
## F-statistic: 32.15 on 3 and 28 DF,  p-value: 3.28e-09
```

The *formula* object is "a symbolic description of the model to be fitted.'' Essentially, a formula is really just a character vector that *contains the tilde sign (~) to separate the outcome (on the left) from the covariates (right).* To see it clearer, I can create a formula first, and then simply paste it to the lm command as in the following:

```
cars_formula <- paste("mtcars$mpg", "mtcars$disp + mtcars$hp + mtcars$drat", sep = "~")
cars_formula
```

```
## [1] "mtcars$mpg~mtcars$disp + mtcars$hp + mtcars$drat"
```

```
fit2 <- lm(formula = cars_formula)
summary(fit2)
```

```
##
## Call:
## lm(formula = cars_formula)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1225 -1.8454 -0.4456  1.1342  6.4958
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.344293   6.370882   3.036  0.00513 **
## mtcars$disp -0.019232   0.009371  -2.052  0.04960 *
## mtcars$hp   -0.031229   0.013345  -2.340  0.02663 *
## mtcars$drat  2.714975   1.487366   1.825  0.07863 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.008 on 28 degrees of freedom
## Multiple R-squared:  0.775,  Adjusted R-squared:  0.7509
## F-statistic: 32.15 on 3 and 28 DF,  p-value: 3.28e-09
```

In the two previous specifications, we only specify the formula; again this is the only required argument (input) for the **lm()** function. The main two problems with that is: (1) we have to type

the name of the dataset *mtcars()* for both the response and all the covariates, and (2) if we want to fit a model with many more covariates (for example, we want to fit the model of the outcome on all the other variables in the dataset), it would takes lot of time to type.

To avoid these problems, we can specify the *data* argument in the **lm** function. As specified in the R documentation, **this argument must be as a data frame object** (i.e not a matrix object), so we want to check the class of our dataset first.

```
class(mtcars)
```

```
## [1] "data.frame"
```

```
### If not data.frame, we can convert it to the dataframe by using as.data.frame
# as.data.frame(mtcars)
```

When it is data.frame, we can fit the same model by the following

```
fit3 <- lm(mpg ~ disp + hp + drat, data = mtcars)
summary(fit3)
```

```
##
## Call:
## lm(formula = mpg ~ disp + hp + drat, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1225 -1.8454 -0.4456  1.1342  6.4958
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.344293   6.370882   3.036  0.00513 **
## disp        -0.019232   0.009371  -2.052  0.04960 *
## hp          -0.031229   0.013345  -2.340  0.02663 *
## drat         2.714975   1.487366   1.825  0.07863 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.008 on 28 degrees of freedom
## Multiple R-squared:  0.775,  Adjusted R-squared:  0.7509
## F-statistic: 32.15 on 3 and 28 DF,  p-value: 3.28e-09
```

or if you want to create a separate formula first, then you can do

```
cars_formula_df <- paste("mpg", "disp + hp + drat", sep = "~")
fit4 <- lm(cars_formula_df, data = mtcars)
summary(fit4)
```

```
##
## Call:
## lm(formula = cars_formula_df, data = mtcars)
##
```



```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1225 -1.8454 -0.4456  1.1342  6.4958
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.344293   6.370882   3.036  0.00513 **
## disp        -0.019232   0.009371  -2.052  0.04960 *
## hp          -0.031229   0.013345  -2.340  0.02663 *
## drat         2.714975   1.487366   1.825  0.07863 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.008 on 28 degrees of freedom
## Multiple R-squared:  0.775,  Adjusted R-squared:  0.7509
## F-statistic: 32.15 on 3 and 28 DF,  p-value: 3.28e-09
```

Next, if you want to fit the model of all the response on all the covariates in the dataset, then you can use the following special formulation of R

```
fit_all_covariates <- lm(mpg ~ ., data = mtcars)
summary(fit_all_covariates)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4506 -1.6044 -0.1196  1.2193  4.6271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.30337   18.71788   0.657   0.5181
## cyl         -0.11144    1.04502  -0.107   0.9161
## disp         0.01334    0.01786   0.747   0.4635
## hp          -0.02148    0.02177  -0.987   0.3350
## drat         0.78711    1.63537   0.481   0.6353
## wt          -3.71530    1.89441  -1.961   0.0633 .
## qsec         0.82104    0.73084   1.123   0.2739
## vs           0.31776    2.10451   0.151   0.8814
## am           2.52023    2.05665   1.225   0.2340
## gear         0.65541    1.49326   0.439   0.6652
## carb        -0.19942    0.82875  -0.241   0.8122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.65 on 21 degrees of freedom
## Multiple R-squared:  0.869,  Adjusted R-squared:  0.8066
## F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07
```


Note that we use the dot (.) on the right hand side of the formula to indicate that we want the covariates to be all other variables in the dataset. And finally, if you want to carry out the regression of the outcome on **all the other variables except some of them**, then we can do

```
fit_all_but_one_covariate <- lm(mpg ~ . - disp, data = mtcars)
fit_all_but_two_covariates <- lm(mpg ~ . - disp - gear, data = mtcars)
summary(fit_all_but_one_covariate)
```

```
##
## Call:
## lm(formula = mpg ~ . - disp, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7863 -1.4055 -0.2635  1.2029  4.4753
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.55052   18.52585   0.677   0.5052
## cyl          0.09627    0.99715   0.097   0.9240
## hp          -0.01295    0.01834  -0.706   0.4876
## drat         0.92864    1.60794   0.578   0.5694
## wt          -2.62694    1.19800  -2.193   0.0392 *
## qsec         0.66523    0.69335   0.959   0.3478
## vs           0.16035    2.07277   0.077   0.9390
## am           2.47882    2.03513   1.218   0.2361
## gear         0.74300    1.47360   0.504   0.6191
## carb        -0.61686    0.60566  -1.018   0.3195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.623 on 22 degrees of freedom
## Multiple R-squared:  0.8655, Adjusted R-squared:  0.8105
## F-statistic: 15.73 on 9 and 22 DF,  p-value: 1.183e-07
```

```
summary(fit_all_but_two_covariates)
```

```
##
## Call:
## lm(formula = mpg ~ . - disp - gear, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8812 -1.3079 -0.2288  1.1706  4.3880
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.37158   16.62855   0.985   0.3351
## cyl         -0.07725    0.92058  -0.084   0.9338
## hp          -0.01153    0.01783  -0.647   0.5243
```



```
## drat         0.99743    1.57595   0.633   0.5330
## wt          -2.74163    1.15698  -2.370   0.0266 *
## qsec         0.62741    0.67801   0.925   0.3644
## vs           0.19796    2.03756   0.097   0.9234
## am           2.79970    1.90147   1.472   0.1545
## carb        -0.46277    0.51435  -0.900   0.3776
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.581 on 23 degrees of freedom
## Multiple R-squared:  0.864,  Adjusted R-squared:  0.8167
## F-statistic: 18.26 on 8 and 23 DF,  p-value: 2.97e-08
```

In other words, if we want to exclude any variable, we can add the "-" sign before its name in the data frame.

