

# STAT 8051 HW 4

Jingxiang Li

October 4, 2014

## Problem 7.8

Sue fits a wls regression with all weights equal to 2. Joe fits a wls regression to the same data with all weights equal to 1. What are the differences in estimates of coefficients, standard errors,  $\sigma^2$  F-tests between Sues and Joes analyses?

### Solution

Let  $m_1$  be the regression model fitted by Sue with weights all equal to 2;  $m_2$  be the regression model fitted by Joe, where all weights equal to 1. Since they both use constant weights, the estimates of coefficients and the F-tests should be the same, but the standard errors and  $\sigma^2$  are different between these two models, but it's only a scale difference.

Claim that  $W_1$  and  $W_2$  are the weighting matrix respectively for  $m_1$  and  $m_2$ , so that we have  $W_1 = 2I$ ,  $W_2 = I$ , where  $I$  is the  $n \times n$  identity matrix. For the coefficient estimator, we have

$$\hat{\beta} = (X'WX)^{-1}X'WY$$

Let  $\hat{\beta}_1$  be the estimator obtained from  $m_1$ ,  $\hat{\beta}_2$  be the estimator obtained from  $m_2$ , so that

$$\hat{\beta}_1 = (X' \times 2I \times X)^{-1}X' \times 2I \times Y = (X'X)^{-1}X'Y = \hat{\beta}_2$$

so that the estimates of coefficients between two models are the same.

Let  $e_i$  be the residual vectors from model  $m_i$ ,  $RSS_i$  be the  $RSS$  derived from model  $m_i$ ,  $\sigma_i^2$  be the  $\sigma^2$  from model  $m_i$ , where  $i = 1, 2$ . and let  $df$  be the degree of freedom. we have

$$RSS_1 = e_1'W_1e_1 = 2e_1'e_1$$

$$RSS_2 = e_2'W_2e_2 = e_1'e_1$$

Note that the estimates of coefficients between two models are the same, so that  $e_1 = e_2$ , and  $RSS_1 = 2RSS_2$ , and

$$\hat{\sigma}_1^2 = \frac{RSS_1}{df} = \frac{2RSS_2}{df} = 2\hat{\sigma}_2^2$$

then we have

$$\hat{\sigma}_1 = \sqrt{2}\hat{\sigma}_2$$

For  $\sigma^2$ , based on the error assumption of WLS, we have  $Var(e_{1i}) = \sigma_1^2/2$ ,  $Var(e_{2i}) = \sigma_2^2$ , and note that  $e_{1i} = e_{2i}$ ,  $\forall i$ , so that

$$\sigma_1^2 = 2\sigma_2^2$$

Let  $F_i$  be the  $F$  statistic for model  $m_i$ , where  $i = 1, 2$ , and let  $n$  be the sample size.

$$F_1 = \frac{(\sum 2(y_i - \bar{y})^2 - RSS_1)/(n - 1 - df)}{RSS_1/df}$$

$$F_1 = \frac{(\sum 2(y_i - \bar{y})^2 - 2RSS_2)/(n - 1 - df)}{2RSS_2/df}$$

$$F_1 = \frac{(\sum (y_i - \bar{y})^2 - RSS_2)/(n - 1 - df)}{RSS_2/df}$$

$$F_1 = F_2$$

In summary, we have

$$\hat{\beta}_1 = \hat{\beta}_2$$

$$\hat{\sigma}_1 = \sqrt{2}\hat{\sigma}_2$$

$$\sigma_1^2 = 2\sigma_2^2$$

$$F_1 = F_2$$

Q.E.D.

## Problem 7.7

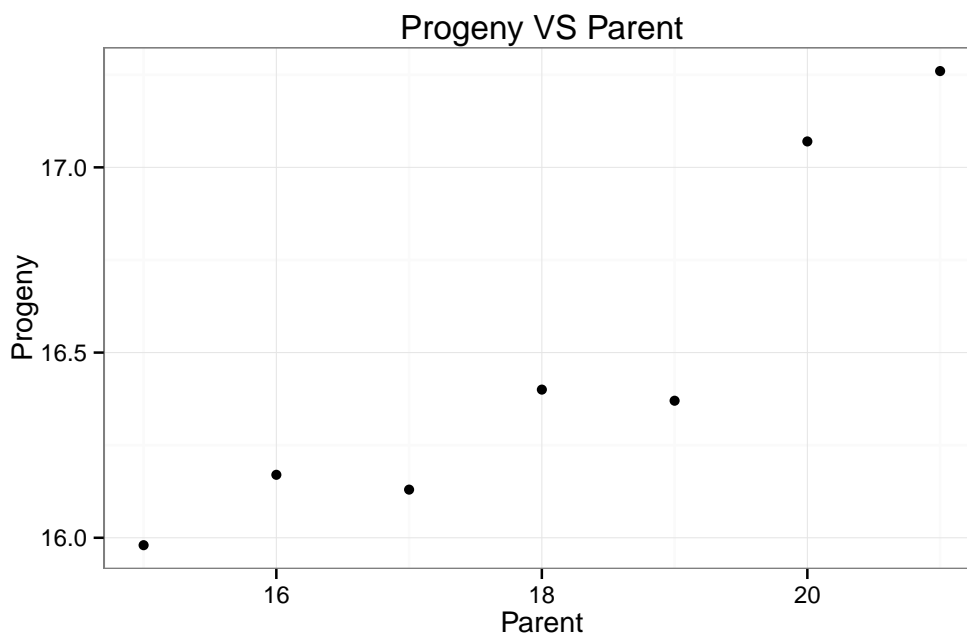
**Galtons sweet peas** (Data file: galtonpeas) Many of the ideas of regression first appeared in the work of Sir Francis Galton (1822-1911) on the inheritance of characteristics from one generation to the next. In Galton (1877), he discussed experiments on sweet peas. By comparing the sweet peas produced by parent plants to those produced by offspring plants, he could observe inheritance from one generation to the next. Galton categorized parent plants according to the typical diameter of the peas they produced. For seven size classes from 0.15 to 0.21 inches, he arranged for each of nine of his friends to grow 10 plants from seed in each size class; however, two of the crops were total failures. A summary of Galton's data were later published in Pearson (1930). The data file includes Parent diameter, Progeny diameter, and SD the standard deviation of the progeny diameters. Sample sizes are unknown but are probably large.

### Problem 7.7.1

Draw the scatterplot of Progeny versus Parent.

### Solution

```
require(alr4)
require(ggplot2)
data <- galtonpeas
p <- ggplot(data = data, aes(x = Parent, y = Progeny))
p <- p + geom_point()
p <- p + theme_bw() + theme(text = element_text(size = 12))
p <- p + ggtitle("Progeny VS Parent")
p
```



### Problem 7.7.2

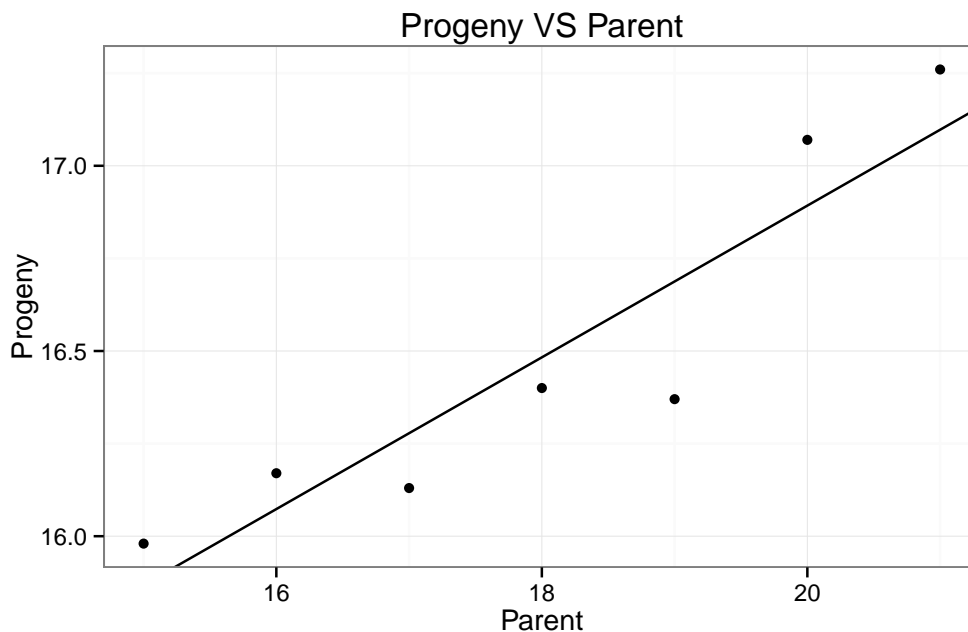
Assuming that the standard deviations given are population values, compute the weighted regression of Progeny on Parent. Draw the fitted mean function on your scatterplot.

### Solution

```
model <- lm(Progeny ~ Parent, data = data, weights = 1 / data$SD^2)
coef <- model$coefficients
```

```
p <- p + geom_abline(intercept = coef[1], slope = coef[2])
```

```
p
```



### Problem 7.7.3

Galton took the average size of all peas produced by a plant to determine the size class of the parental plant. Yet for seeds to represent that plant and produce offspring, Galton chose seeds that were as close to the overall average size as possible. Thus, for a small plant, the exceptional large seed was chosen as a representative, while larger, more robust plants were represented by relatively smaller seeds. What effects would you expect these experimental biases to have on (1) estimation of the intercept and slope and (2) estimates of error?

### Solution

Following the way Galton determine the seeds that represent the parent plant, the estimation of the slope will be lower than the true nature, and thus the intercept term will increase, and the variance will increase too.

Due to the fact that larger seeds are chosen for small plant, smaller seeds are chosen for large plant, the offspring's size will be centralized. It means that for each level of Parent, Progeny are more closer to each other compared with the true nature, so that the slope coefficient will decrease. Then, since the central point for this regression model won't change too much and the slope will decrease, the intercept will increase. Then for the variance of model, since Galton choose exceptional large or small seeds to represent parent plant, the variance of Progeny's size for each level of Parent will definitely increase.

## Problem 7.8

**Jevonss gold coins** (Data file: jevons) The data in this example are deduced from a diagram in Jevons (1868) and provided by Stephen M. Stigler. In a study of coinage, Jevons weighed 274 gold sovereigns that he had collected from circulation in Manchester, England. For each coin, he recorded the weight after cleaning to the nearest 0.001 g, and the date of issue. The data file includes Age, the age of the coin in decades, n, the number of coins in the age class, Weight, the average weight of the coins in the age class, SD, the standard deviation of the weights. The minimum Min and maximum Max of the weights are also given. The standard weight of a gold sovereign was 7.9876 g; the minimum legal weight was 7.9379 g.

### Problem 7.8.1

Draw a scatterplot of Weight versus Age, and comment on the applicability of the usual assumptions of the linear regression model. Also draw a scatterplot of SD versus Age, and summarize the information in this plot.

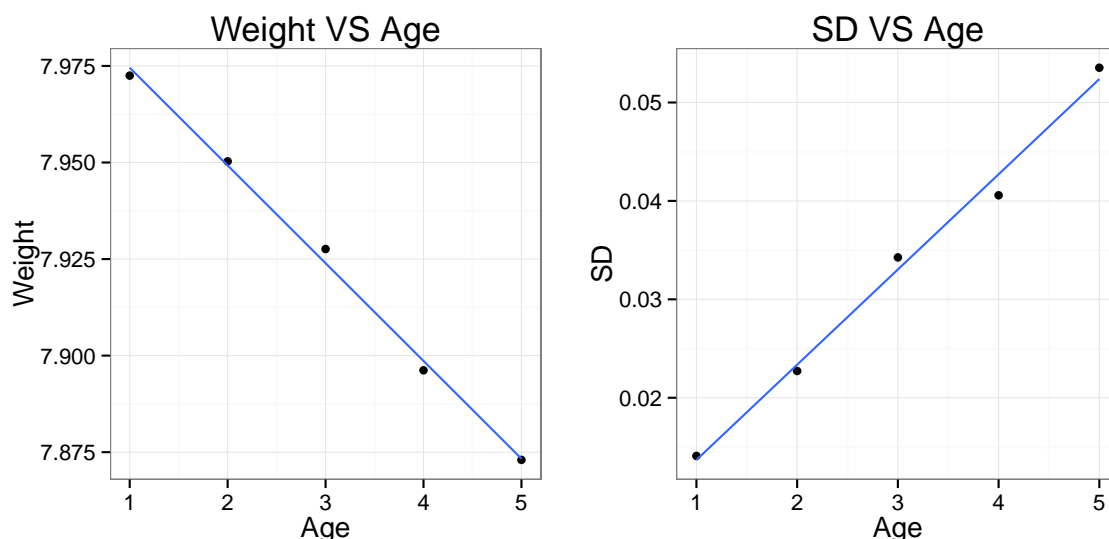
### Solution

```
require(gridExtra)
data <- jevons

p1 <- ggplot(data = data, aes(x = Age, y = Weight))
p1 <- p1 + geom_point()
p1 <- p1 + geom_smooth(formula = y ~ x, method = "lm", se = FALSE)
p1 <- p1 + theme_bw() + theme(text = element_text(size = 14))
p1 <- p1 + ggtitle("Weight VS Age")

p2 <- ggplot(data = data, aes(x = Age, y = SD))
p2 <- p2 + geom_point()
p2 <- p2 + geom_smooth(formula = y ~ x, method = "lm", se = FALSE)
p2 <- p2 + theme_bw() + theme(text = element_text(size = 14))
p2 <- p2 + ggtitle("SD VS Age")

grid.arrange(p1, p2, ncol=2)
```



For the regression model Weight versus Age, following the information provided by the scatterplot, the linear form is appropriate and there is no specific information shows that the variance of each case in this dataset is different. However, as the standard deviation for each point is given, we should consider WLS as our primary choice.

For the SD versus Age scatterplot, we can see that as Age goes on, the standard deviation of the weights increase linearly against Age.

## Problem 7.8.2

To fit a simple linear regression model with Weight as the response, wls should be used with variance function  $Var(\text{Weight}|\text{Age}) = n\sigma^2/SD^2$ . Sample sizes are large enough to assume the SD are population values. Fit the wls model.

## Solution

```
w <- data$SD^2 / data$n
m <- lm(Weight ~ Age, data = data, weight = w)
summary(m)

##
## Call:
## lm(formula = Weight ~ Age, data = data, weights = w)
##
## Weighted Residuals:
##      1      2      3      4      5
## -5.21e-06 -5.23e-07  1.94e-05 -2.07e-05  8.66e-06
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.00270    0.00559  1431.3  7.5e-10 ***
```



```
## Age          -0.02610    0.00129   -20.2  0.00027 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.74e-05 on 3 degrees of freedom
## Multiple R-squared:  0.993, Adjusted R-squared:  0.99
## F-statistic: 408 on 1 and 3 DF, p-value: 0.000265
```

### Problem 7.8.3

Is the fitted regression consistent with the known standard weight for a new coin?

### Solution

To obtain the average weight for a new coin, we only need refer to the intercept term.

```
coef(m)
```

```
## (Intercept)      Age
##      8.0027    -0.0261
```

Note that the fitted intercept is 8.0027, which is larger than the standard weight 7.9876, suggesting that it is consistent with the known standard weight for a new coin.

### Problem 7.8.4

For previously unsampled coins of Age = 1, 2, 3, 4, 5, estimate the probability that the weight of the coin is less than the legal minimum.

(Hints: The standard error of prediction is the square root of the sum of two terms, the assumed known variance of an unsampled coin of known Age, which is different for each age, and the estimated variance of the fitted value for that Age; the latter is computed from the formula for the variance of a fitted value. You should use the normal distribution rather than a  $t$  to get the probabilities.)

### Solution

```
m_var <- vcov(m)
m_var <- m_var[1,1] + data$Age^2 * m_var[2,2] + 2 * data$Age * m_var[1,2]
m_sd <- sqrt(m_var + data$SD^2)
y_hat <- predict(object = m, newdata = data.frame(Age = 1:5))
pnorm((7.9379 - y_hat) / m_sd)

##      1      2      3      4      5
## 0.004325 0.291282 0.652945 0.835362 0.890044
```

### Problem 7.8.5

Determine the Age at which the predicted weight of coins is equal to the legal minimum, and use the delta method to get a standard error for the estimated age. This problem is called inverse regression, and is discussed by Brown (1993).

### Solution

To solve this problem, we construct estimator

$$g(\hat{\beta}) = \frac{7.9379 - \hat{\beta}_0}{\hat{\beta}_1}$$

So that the estimated Age should be

$$g^*(\hat{\beta}) = \frac{7.9379 - 8.0027}{-0.0261} = 2.4828$$

Using delta method, we can get a standard error for the estimated age.

$$g(\hat{\beta})' = \left(-\frac{1}{\hat{\beta}_1}, -\frac{7.9379 - \hat{\beta}_0}{\hat{\beta}_1^2}\right) = (38.3141, 95.1249)$$

So that

$$Var[g(\hat{\beta})] = g(\hat{\beta})' Var(\hat{\beta}) (g(\hat{\beta}))^T$$

```
grad <- as.matrix(c(38.3141, 95.1249))
sqrt(t(grad) %*% vcov(m) %*% grad)
```

```
##           [,1]
## [1,] 0.09657
```

So that the standard error for the estimated age is 0.09657.

## Problem 7.10

(Data file: fuel2001)

### Problem 7.10.1

Use the bootstrap to estimate confidence intervals for the coefficients in the fuel data, and compare the results with the usual large sample ols estimates.

### Solution

```
data <- fuel2001
data$Dlic <- data$Drivers / data$Pop
data$Fuel <- 1000 * data$FuelC / data$Pop
m <- lm(Fuel ~ Tax + Dlic + Income + log(Miles), data = data)
data_ori <- data

library(boot)
confint_bootstrap <- function(data, indices) {
  d <- data[indices, ] # allows boot to select sample
  data_tmp <- data_ori
  data_tmp$Fuel <- data_tmp$Fuel + d
  fit <- lm(Fuel ~ Tax + Dlic + Income + log(Miles), data_tmp)
  return(coef(fit))
}

set.seed(123)
results <- boot(data = as.matrix(m$residuals),
               statistic = confint_bootstrap, R = 500)
result <- results$t
colnames(result) <- names(coef(m))

## Bootstrap confidence interval
apply(X = result, MARGIN = 2, FUN = quantile, probs = c(0.025, 0.975))

##      (Intercept)      Tax  Dlic      Income log(Miles)
## 2.5%      -219.1  -7.870 231.1 -0.010151      8.587
## 97.5%       560.6 -0.405 705.6 -0.002059     44.601

## OLS confidence interval
t(confint(m))

##      (Intercept)      Tax  Dlic      Income log(Miles)
## 2.5 %      -238.1 -8.3144 213.2 -0.01055      7.96
## 97.5 %       546.5 -0.1416 730.6 -0.00172     45.55
```

In this problem, since the linear form of model is appropriate, we apply residual bootstrap to obtain the bootstrap confidence interval of coefficients. As the result displayed in the code chunk above, we see that the bootstrap confidence interval is more or less different from the interval derived from usual OLS, especially for the coefficient of Tax.

## Problem 7.10.2

Examine the histograms of the bootstrap replications for each of the coefficients. Are the histograms symmetric or skewed? Do they look like normally distributed data, as they would if the large sample normal theory applied to these data? Do the histograms support or refute the differences between the bootstrap and large sample confidence intervals found in Problem 7.10.1?

## Solution

```
colnames(result)[1] <- "Intercept"
colnames(result)[5] <- "log_Miles"

p1 <- ggplot(data = as.data.frame(result), aes(x = Intercept)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
  geom_density(size = 1) + theme_bw() + theme(text = element_text(size = 14)) +
  ggtitle("Intercept")

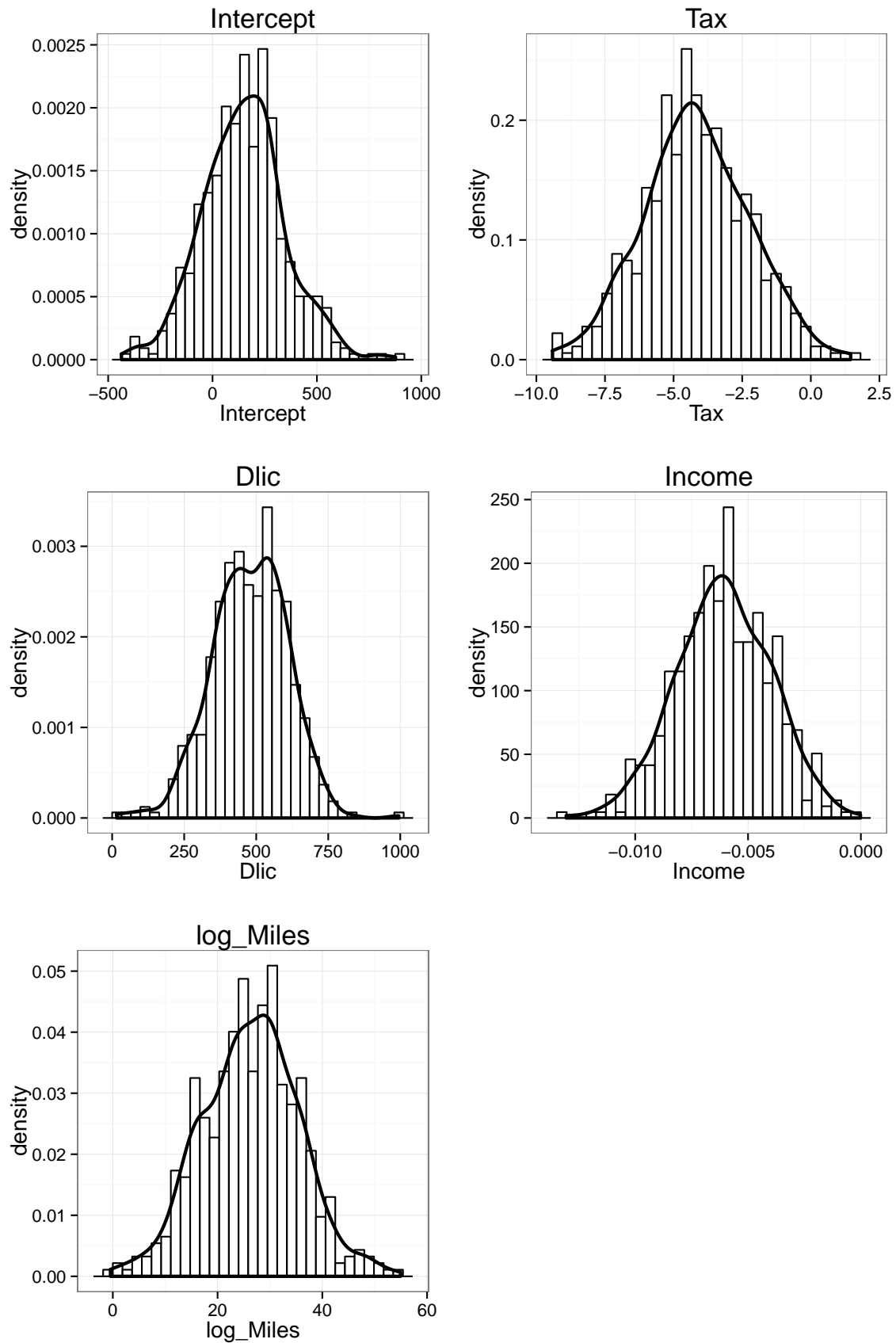
p2 <- ggplot(data = as.data.frame(result), aes(x = Tax)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
  geom_density(size = 1) + theme_bw() + theme(text = element_text(size = 14)) +
  ggtitle("Tax")

p3 <- ggplot(data = as.data.frame(result), aes(x = Dlic)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
  geom_density(size = 1) + theme_bw() + theme(text = element_text(size = 14)) +
  ggtitle("Dlic")

p4 <- ggplot(data = as.data.frame(result), aes(x = Income)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
  geom_density(size = 1) + theme_bw() + theme(text = element_text(size = 14)) +
  ggtitle("Income")

p5 <- ggplot(data = as.data.frame(result), aes(x = log_Miles)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
  geom_density(size = 1) + theme_bw() + theme(text = element_text(size = 14)) +
  ggtitle("log_Miles")

grid.arrange(p1, p2, p3, p4, p5, ncol=2)
```



According to the histograms above, we can say that the histograms are not all symmetric, and not all of them look like normally distributed data. For example, the coefficient of the intercept seems like skewed, and the distribution of coefficient of Dlic seems has two peaks, suggesting that they may not be normally distributed, so that the large sample

normal theory may not be appropriate to these data.

However, it's kind of hard to say that whether the histograms support or refuse the differences between the bootstrap and large sample confidence intervals, because the shape of a histogram is also largely affected by the binwidth we choose, and it's risky to make decision based on histograms. Maybe we need to do some non-parametric test to determine whether they are normally distributed, and then we can say that if the differences are supported or not.

## Problem 10.2

(Data file: Highway)

### Problem 10.2.1

For the highway accident data, use your software to verify the forward selection and backward elimination subsets that are given in Section 10.2.2.

### Solution

```
data <- Highway
data$sigs1 <- (data$sigs * data$len + 1) / data$len
m <- lm(log(rate) ~ log(len) + log(adt) + log(trks) + log(shld) + log(sigs1) +
        lane + slim + shld + lwid + acpt + itg + htype, data = data)
m_min <- lm(log(rate) ~ shld + log(len), data = data)
m_forward <- step(m_min, scope = formula(m), direction = "forward",
                 trace = FALSE)
m_backward <- step(m, scope = list(lower = formula(m_min)),
                  direction = "backward", trace = FALSE)

## forward selection
summary(m_forward)

##
## Call:
## lm(formula = log(rate) ~ shld + log(len) + acpt + log(sigs1),
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.4394 -0.1844 -0.0352  0.2384  0.4661
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.16330    0.29354   7.37 1.5e-08 ***
## shld          -0.04491    0.01671  -2.69  0.0111 *
## log(len)      -0.26795    0.09068  -2.95  0.0056 **
## acpt           0.01562    0.00598   2.61  0.0133 *
## log(sigs1)     0.09986    0.04467   2.24  0.0321 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.273 on 34 degrees of freedom
## Multiple R-squared:  0.69, Adjusted R-squared:  0.653
```

```
## F-statistic: 18.9 on 4 and 34 DF,  p-value: 2.92e-08

## backward selection
summary(m_backward)

##
## Call:
## lm(formula = log(rate) ~ log(len) + log(adl) + log(sigs1) + slim +
##      shld + htype, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.5842 -0.1179 -0.0252  0.0960  0.4561
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.42937    0.62740    7.06 7.5e-08 ***
## log(len)     -0.25768    0.08445   -3.05  0.0047 **
## log(adl)     -0.13099    0.08534   -1.53  0.1353
## log(sigs1)    0.20838    0.05930    3.51  0.0014 **
## slim         -0.03190    0.01245   -2.56  0.0156 *
## shld          0.00681    0.02398    0.28  0.7784
## htypefai      0.11833    0.34202    0.35  0.7318
## htypepa      -0.38224    0.23619   -1.62  0.1161
## htypema      -0.14106    0.20511   -0.69  0.4969
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.247 on 30 degrees of freedom
## Multiple R-squared:  0.776, Adjusted R-squared:  0.716
## F-statistic: 13 on 8 and 30 DF,  p-value: 7.29e-08
```

Note that the forward selection and backward selection subsets we have from the code chunk above are exactly the same as what are given in Section 10.2.2.

## Problem 10.2.2

Use as response  $\log(\text{rate} \times \text{len})$  and treat `lwid` as the focal regressor. Use both forward selection and backward elimination to assess the importance of `lwid`. Summarize your results.

## Solution



```

data$log_rate_len <- log(data$rate * data$len)
m_max <- lm(log_rate_len ~ log(len) + log(adt) + log(trks) + log(shld) + log(sigs1) +
            lane + slim + shld + lwid + acpt + itg + htype, data = data)
m_min <- lm(log_rate_len ~ lwid, data = data)

m_forward <- step(m_min, scope = formula(m_max), direction = "forward",
                 trace = FALSE)
m_backward <- step(m_max, scope = list(lower = formula(m_min)),
                  direction = "backward", trace = FALSE)

summary(m_forward)

##
## Call:
## lm(formula = log_rate_len ~ lwid + log(len) + slim + acpt + log(trks),
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.5152 -0.1617  0.0397  0.1733  0.5563
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.80121    1.46241   3.28  0.0024 **
## lwid          -0.05235    0.10359  -0.51  0.6167
## log(len)       0.75168    0.08938   8.41 1e-09 ***
## slim          -0.03117    0.01046  -2.98  0.0054 **
## acpt           0.01086    0.00675   1.61  0.1171
## log(trks)     -0.33565    0.21626  -1.55  0.1302
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.273 on 33 degrees of freedom
## Multiple R-squared:  0.75, Adjusted R-squared:  0.713
## F-statistic: 19.8 on 5 and 33 DF, p-value: 4.28e-09

summary(m_backward)

##
## Call:
## lm(formula = log_rate_len ~ log(len) + log(adt) + log(sigs1) +
##     slim + lwid + htype, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max

```

```
## -0.4707 -0.1212 -0.0209  0.1019  0.4535
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.65585     1.32782   2.75  0.0099 **
## log(len)     0.75333     0.08690   8.67 1.1e-09 ***
## log(adt)    -0.13616     0.08530  -1.60  0.1209
## log(sigs1)   0.21702     0.06093   3.56  0.0013 **
## slim        -0.02971     0.00979  -3.04  0.0049 **
## lwid         0.05676     0.09832   0.58  0.5680
## htypefai     0.15770     0.34508   0.46  0.6510
## htypepa     -0.36249     0.23278  -1.56  0.1299
## htypepema    -0.12339     0.20689  -0.60  0.5554
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.246 on 30 degrees of freedom
## Multiple R-squared:  0.816, Adjusted R-squared:  0.767
## F-statistic: 16.6 on 8 and 30 DF,  p-value: 4.34e-09
```

As the result we see that the forward model end up with 5 regressors, `lwid`, `log(len)`, `slim`, `acpt` and `log(trks)`. The p-value of our focal regressor `lwid` is very large, suggesting that the `lwid` does not have significant effect on the response term.

Then, 6 regressors included in the backward selection model, they are `log(len)`, `log(adt)`, `log(sigs1)`, `slim`, `lwid` and `htype`. Note that the p-value of focal regressor `lwid` is still very large, which means that the main effect of `lwid` over the response is negligible.

### Problem 10.2.3

Using the identity  $\log(\text{rate} \times \text{len}) = \log(\text{rate}) + \log(\text{len})$ , we can write

$$E(\log(\text{rate} \times \text{len})|X = x) = \beta_0 + \beta'x$$

$$E(\log(\text{rate}) + \log(\text{len})|X = x) = \beta_0 + \beta'x$$

$$E(\log(\text{rate})|X = x) = \beta_0 + \beta'x - \log(\text{len})$$

In this last equation, the variable  $\log(\text{len})$  is on the right side of the equation with an implied known regression coefficient equal to 1. A regressor with a known regression coefficient is called an *offset*, and most modern regression software allows you to include offsets in fitting a model. The difference between an offset and a regressor is that no coefficient will be estimated for the offset.

Repeat Problem 10.2.2, but use  $\log(\text{rate})$  as the response and  $\log(\text{len})$  as an offset. Is the analysis the same or different? Explain.

### Solution

```

m_max <- lm(log(rate) ~ offset(-log(len)) + log(adt) + log(trks) + log(shld) +
            log(sigs1) + lane + slim + shld + lwid + acpt + itg + htype,
            data = data)
m_min <- lm(log(rate) ~ lwid + offset(-log(len)), data = data)
m_forward <- step(m_min, scope = formula(m_max), direction = "forward",
                 trace = FALSE)
m_backward <- step(m_max, scope = list(lower = formula(m_min)),
                  direction = "backward", trace = FALSE)

summary(m_forward)

##
## Call:
## lm(formula = log(rate) ~ lwid + shld + log(shld) + offset(-log(len)),
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.8838 -0.3298 -0.0145  0.1978  0.7301
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.5330     1.9484   3.87 0.00046 ***
## lwid         -0.3071     0.1553  -1.98 0.05591 .
## shld         -0.2222     0.0897  -2.48 0.01825 *
## log(shld)     0.7344     0.4564   1.61 0.11657
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.414 on 35 degrees of freedom
## Multiple R-squared:  0.688, Adjusted R-squared:  0.661
## F-statistic: 25.7 on 3 and 35 DF, p-value: 5.77e-09

summary(m_backward)

##
## Call:
## lm(formula = log(rate) ~ log(shld) + shld + lwid + offset(-log(len)),
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.8838 -0.3298 -0.0145  0.1978  0.7301
##
## Coefficients:

```

```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.5330     1.9484   3.87 0.00046 ***
## log(shld)     0.7344     0.4564   1.61 0.11657
## shld         -0.2222     0.0897  -2.48 0.01825 *
## lwid         -0.3071     0.1553  -1.98 0.05591 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.414 on 35 degrees of freedom
## Multiple R-squared:  0.688, Adjusted R-squared:  0.661
## F-statistic: 25.7 on 3 and 35 DF,  p-value: 5.77e-09
```

Since we view  $-\log(\text{len})$  as an offset, the coefficient of  $\log(\text{len})$  will be set as -1, so that the other predictors' estimates will not be affected by  $\log(\text{len})$ . In this way, we see that both two models select the same three regressors, which are *lwid*, *shld* and *log(shld)*. Then the p-value of focal predictor *lwid* is very close to 0.05, suggesting that the main effect of *lwid* is worthy of further attention and study, and we need more samples to certify its significance.