Assignment2

```r
#Code snippets given
salary <-read.csv("http://www.math.mcgill.ca/yyang/regression/data/salary.csv")
x1<-salary$SPENDING/1000
y<-salary$SALARY

fit.salary <- lm(y ~ x1)
summary(fit.salary)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3848.0 -1844.6  -217.5  1660.0  5529.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12129.4     1197.4   10.13 1.31e-13 ***
## x1            3307.6      311.7   10.61 2.71e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2325 on 49 degrees of freedom
## Multiple R-squared:  0.6968, Adjusted R-squared:  0.6906
## F-statistic: 112.6 on 1 and 49 DF,  p-value: 2.707e-14
```

(A) Write R code to verify the calculation of the entries in the Estimate column, and show that your code produces the correct results.

```r
n <- length(x1)
#Creating (n*2) matrix X
X <- cbind(rep(1,n), x1)
XtX <- t(X) %*% X
Xty <- t(X) %*% y
#Finding estimates
beta.hat <- solve(XtX, Xty)
beta.hat
```

```
##         [,1]
##    12129.371
## x1  3307.585
```

(B) Write R code to compute the value of the omitted entry for the Residual standard error on line 20.

since

$$\mathrm{E}[e_i] = 0$$

then the residual standard error

$$\widehat{\sigma} = \sqrt{\sigma^2} = \sqrt{MS_{res}}$$

and

$$MS_{res} = \frac{Y^T(I_n - H)Y}{n - p}$$

with $p = 2$ for Simple Linear Regression.

```
#Defining Hat matrix H
XtXinv <-solve(XtX)
H <- X%*%(XtXinv%*% t(X))
I_n <- diag(n)

MS_res<- (t(y) %*%((I_n-H)%*% y))/(n-2)
MS_res
```

```
##         [,1]
## [1,] 5404597
```

```
RSE <- sqrt(MS_res)
RSE
```

```
##          [,1]
## [1,] 2324.779
```

```
Resid <- y- X%*%beta.hat
t(Resid) %*% Resid/(n-2)
```

```
##         [,1]
## [1,] 5404597
```

```
sqrt(t(Resid) %*% Resid/(n-2))
```

```
##          [,1]
## [1,] 2324.779
```

(C) Compute the value of the entry in the Std. Error column on line 15 first using entries already given in the table, and then using the data directly.

We know our observed student t-value is

$$t_j = \frac{\widehat{\beta}_j}{ese(\widehat{\beta}_j)}; j = 0, 1$$

If we solve this for estimated standard error (Std. Error in R output),

$$ese(\widehat{\beta}_j) = \frac{\widehat{\beta}_j}{t_j}$$

```r
#using the values
tbl_computed_ese_b0 <- 12129.4/10.13
tbl_computed_ese_b1 <- 3307.6/10.61
c(tbl_computed_ese_b0, tbl_computed_ese_b1)
```

```
## [1] 1197.3741  311.7436
```

```r
#Using the data directly
Var.beta <- MS_res[1][1]*(XtXinv)
Var.beta
```

```
##                        x1
##    1433648.9 -359160.75
## x1 -359160.8   97159.55
```

```r
raw_ese_b0 <- sqrt(Var.beta[1])
raw_ese_b1 <- sqrt(Var.beta[4])
c(raw_ese_b0,raw_ese_b1)
```

```
## [1] 1197.3508  311.7043
```

They seem to agree to 1 digit after the decimal point (there lies some rounding error).

(D) The entry for Multiple R-squared on the table is computed using the formula

`$$R^2 = SS_R/SS_T$$`

where SS_R is the 'regression sum-of-squares' and SS_T is the 'total sum of squares' as defined in lectures. Write R code to verify the calculation of R^2.

From the lecture, we know

$$SS_T = Y^T(I_n - H_1)Y$$

and

$$SS_R = Y^T(H - H_1)Y$$

Where

$$H_1 = 1_n(1_n^T 1_n)^{-1} 1_n^T$$

Thus, we will compute SS_T and

$$SS_R$$

using linear algebra on R.

```r
  #Creating H_1 matrix
one_n <- rep(1,n)
# 1/n = [t(one_n) %*% one_n]inv
H_1 <- one_n %*% t(one_n)/n
SS_T <- t(y) %*% ((I_n-H_1)%*%y)
SS_R <- t(y) %*% ((H-H_1)%*%y)
R_sq <- SS_R/SS_T
SS_res <- SS_T - SS_R
c(SS_T, SS_res, SS_R, R_sq)
```

```
## [1] 8.733803e+08 2.648252e+08 6.085550e+08 6.967813e-01
```

6.967813e-01 = 0.69678… so it agrees with the value on the table.

(E) Prove for a simple linear regression that, in the notation from lectures,

$$SS_R = \widehat{\beta}_1 Sxy$$

and show this result holds numerically for the salary data.

$$SS_R = \sum_{i=1}^{n} (\widehat{y}_i - \bar{y})^2$$

Since

$$1_n^T * e = \sum_{i=1}^{n} e_i = \sum_{i=1}^{n} (y_i - \widehat{y}_i) = 0$$

we have

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \widehat{y}_i => \bar{y} = \widehat{\beta}_0 + \widehat{\beta}_1 \bar{x_1}$$

$$= \sum_{i=1}^{n} (\widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} - (\widehat{\beta}_0 + \widehat{\beta}_1 \bar{x_1}))^2 = \sum_{i=1}^{n} (\widehat{\beta}_1 (x_{i1} - \bar{x}))^2 = \widehat{\beta}_1^2 \sum_{i=1}^{n} (x_{i1} - \bar{x})^2$$

$$= \widehat{\beta}_1^2 S_{xx} = \widehat{\beta}_1 \frac{(S_{xy})}{(S_{xx})} * S_{xx} = \widehat{\beta}_1 S_{xy} Q.E.D.$$

N.B. We also know from the lectures

$$S_{xy} = \sum_{i=1}^{n} (y_i - \bar{y})(x_{i1} - \bar{x_1}) = [\sum_{i=1}^{n} x_{i1} y_i] - n\bar{x_1}\bar{y}$$

```
S_xy <- t(x1) %*% y - n*mean(x1)*mean(y)
SS_R_tilde <- beta.hat[2]*S_xy
c(SS_R, SS_R_tilde)
```

```
## [1] 608555015 608555015
```

so the results numerically agree.

(F)The F-statistic on the table is computed suing the sums-of-squares decompotion

The F-statistic on the table is computed using the sums-of-squares decomposition

$$SS_T = SS_{res} + SS_R$$

$$F = \frac{SS_R/(p-1)}{SS_{res}/(n-p)}$$

having p = 2 for Simple Linear Regression. Write R code to compute the omitted value for F.

From previous questions we also know

$$\frac{SS_{res}}{n-p} = MS_{res}$$

So

```r
#Calculating SS_res using sums-of-squares decomposition

F_value <-  (SS_R/(2-1))/(SS_res/(n-2))
#
F_value_verified <- SS_R/MS_res
c(F_value, F_value_verified)
```

```
## [1] 112.5995 112.5995
```

So our numerical verification agrees with our F-value on the table after rounding.

(G) In the notation from lectures, we have that the sums-of-squares decomposition can be written

$$y^T(I_n - H_1)y = y^T(I_n - H)y + y^T(H - H_1)y$$

Show, mathematically and numerically, that

$$trace(I_n - H_1) = n - 1$$
$$trace(H - H_1) = p - 1$$

with p=2

Remark that trace is a linear map, so

$$trace(I_n - H_1) = trace(I_n) + trace(-H_1) = trace(I_n) - trace(H_1)$$

and we know

$$trace(I_n) = 1 * n = n$$

and

$$trace(H_1) = trace(\begin{bmatrix} 1/n & 1/n & ... & 1/n \\ 1/n & 1/n & ... & 1/n \\ 1/n & 1/n & ... & 1/n \\ 1/n & 1/n & 1/n & 1/n \end{bmatrix}_{nxn}) = 1/n * n = 1$$

so

$$trace(I_n - H_1) = trace(I_n) - trace(H_1) = n - 1$$

Likewise,

$$trace(H - H_1) = trace(H) - trace(H_1) = trace(H) - 1$$
$$trace(H) = trace(X(X^TX)^{-1}X^T) = trace(X^TX(X^TX)^{-1}) = trace(I_p) = p$$

Thus,

$$trace(H - H_1) = trace(H) - 1 = p - 1$$

(ii) Numerical verification

```
#We have hat matrix H, Identity matrix I_n, and hat_1 matrix H_1 ready
#from the answers to the previous questions

#The theory suggests that
#trace(I_n-H_1) should return n-1 = 51-1 = 50
sum(diag(I_n-H_1))
```

```
## [1] 50
```

```
#The theory also suggests that trace(H)-1 = 2-1 =1
sum(diag(H-H_1))
```

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

(H)Using residual plots, assess the validity of the assumptions underlying the least squares analysis. Verify numerically the orthogonality results concerning the residuals, that is, in vector form

$$1_n^T * e = 0$$
$$X^T * e = 0_p$$
$$\widehat{y}^T e = 0$$

```
#Residuals
y_hat <- X %*% beta.hat
computed_residuals <- y-y_hat

#Checking orthogonality
t(one_n) %*% computed_residuals
```

```
##               [,1]
## [1,] -1.67347e-10
```

```
t(one_n)%*% residuals(fit.salary)
```

```
##               [,1]
## [1,] 5.684342e-12
```

```
t(X)%*% computed_residuals
```

```
##              [,1]
##    -1.673470e-10
## x1  1.587978e-09
```

```
t(X)%*% residuals(fit.salary)
```

```
##            [,1]
##    5.684342e-12
## x1 3.637979e-12
```

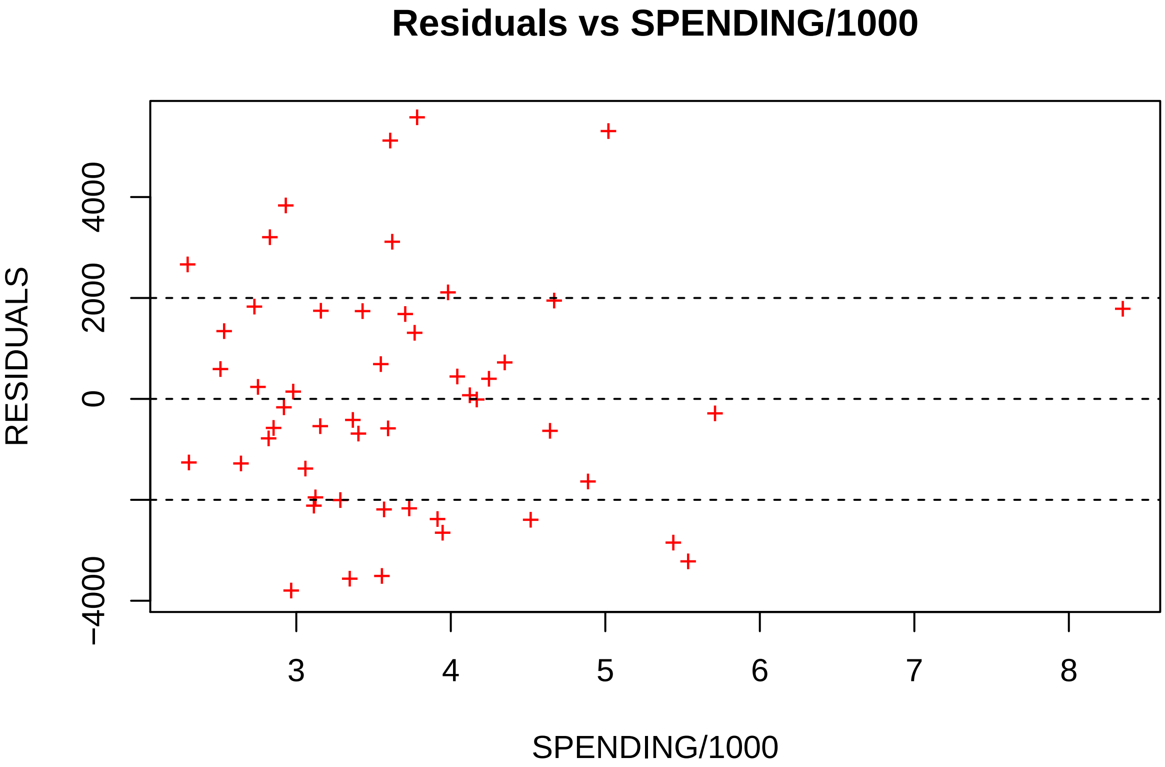
```
t(y_hat) %*% computed_residuals
```

```
##               [,1]
## [1,] 3.311783e-06
```

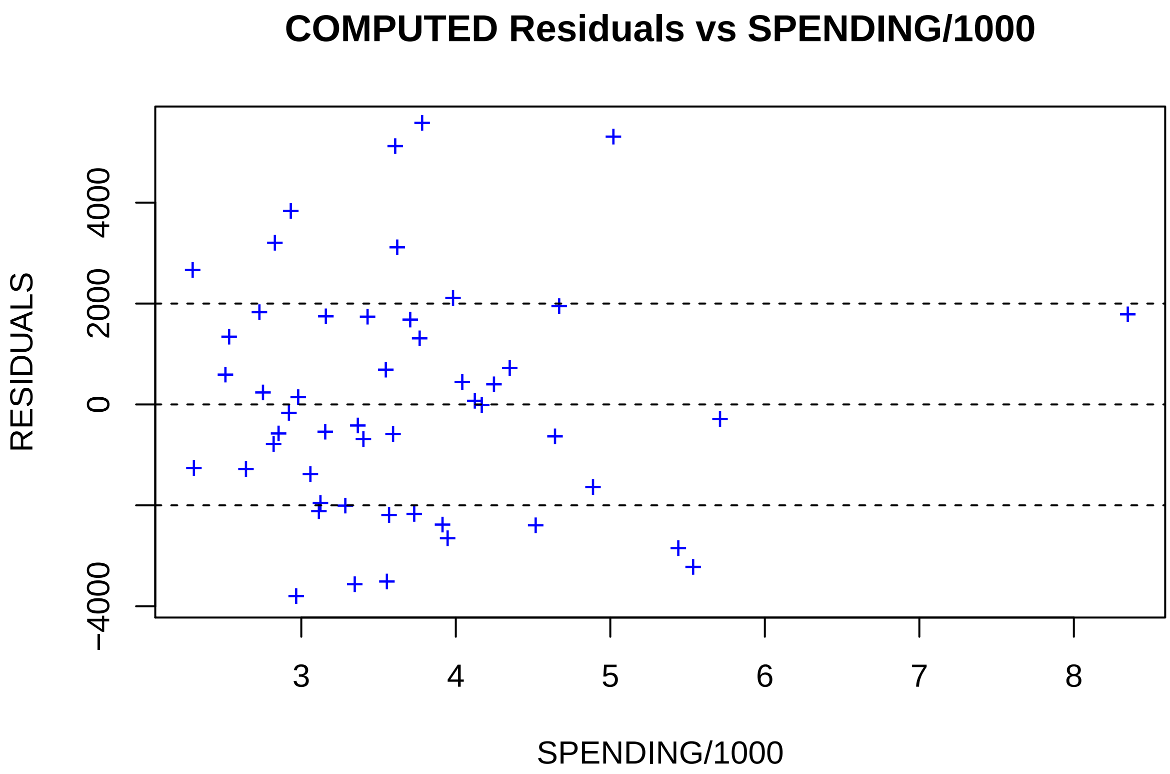
```
t(fitted(fit.salary)) %*% computed_residuals
```

```
##                [,1]
## [1,] 3.352761e-06
```

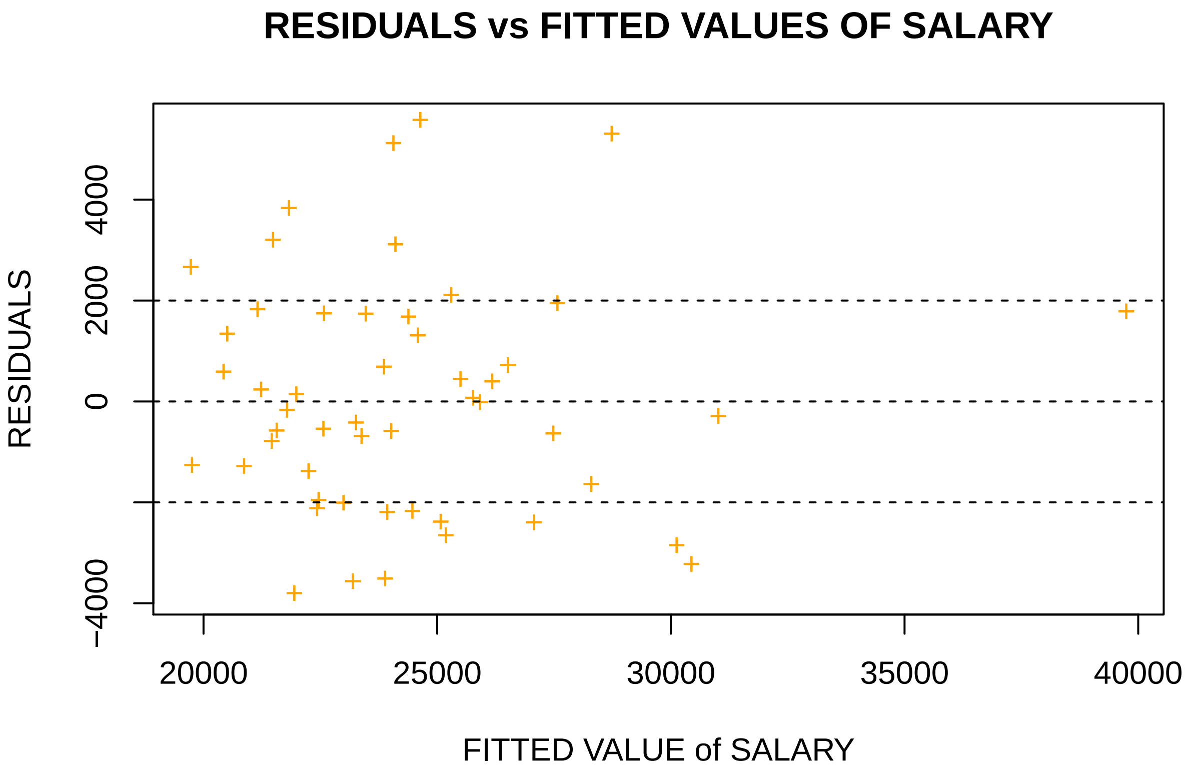
```
s.resids<-residuals(fit.salary)
s.fitted<-fitted(fit.salary)



#Plotting residuals - Resid v SPENDING using residuals()
#for comparison with the hand-calculated data
plot(x1, s.resids,xlab='SPENDING/1000',ylab='RESIDUALS',pch="+",
     col = 'red')
abline(h=0,lty=2); abline(h=2000, lty=2);
abline(h=-2000, lty=2)
title('Residuals vs SPENDING/1000')
```



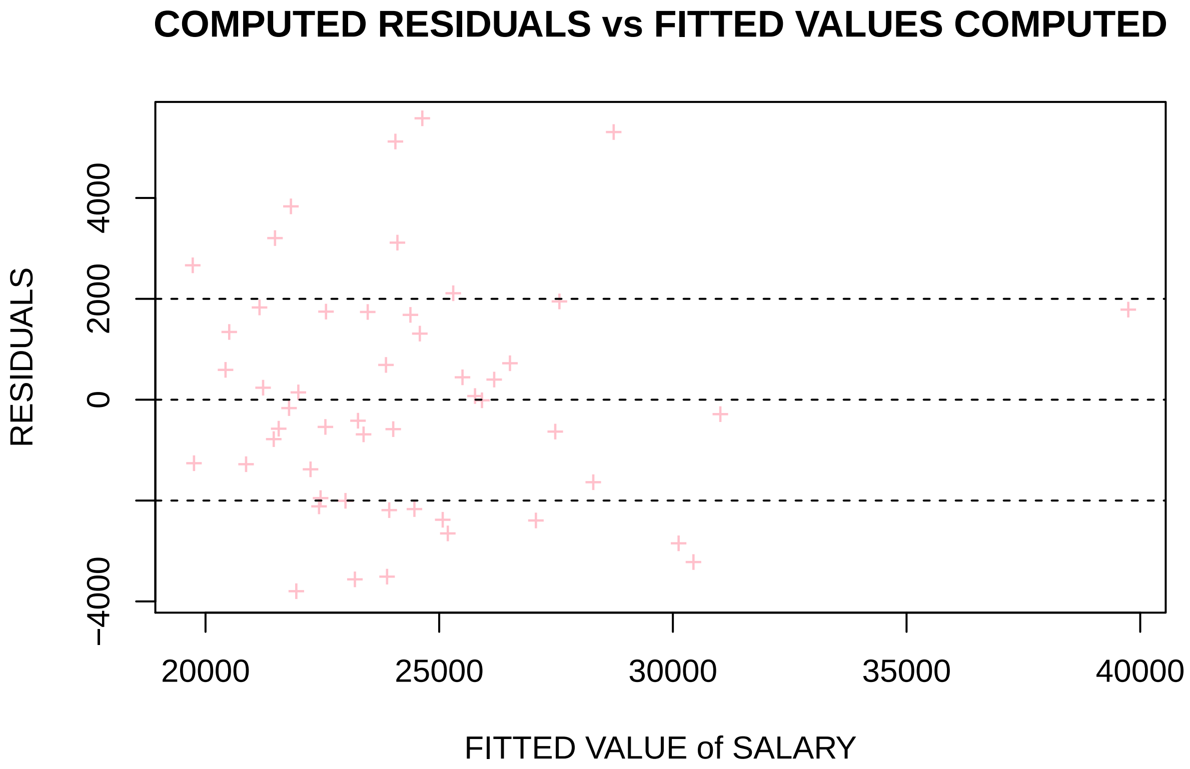
```
#Plotting residuals - Resid v SPENDING using hand-calculated data
plot(x1, computed_residuals,xlab='SPENDING/1000',ylab='RESIDUALS',pch="+",
     col = 'blue')
abline(h=0,lty=2); abline(h=2000, lty=2);
abline(h=-2000, lty=2)
title('COMPUTED Residuals vs SPENDING/1000')
```



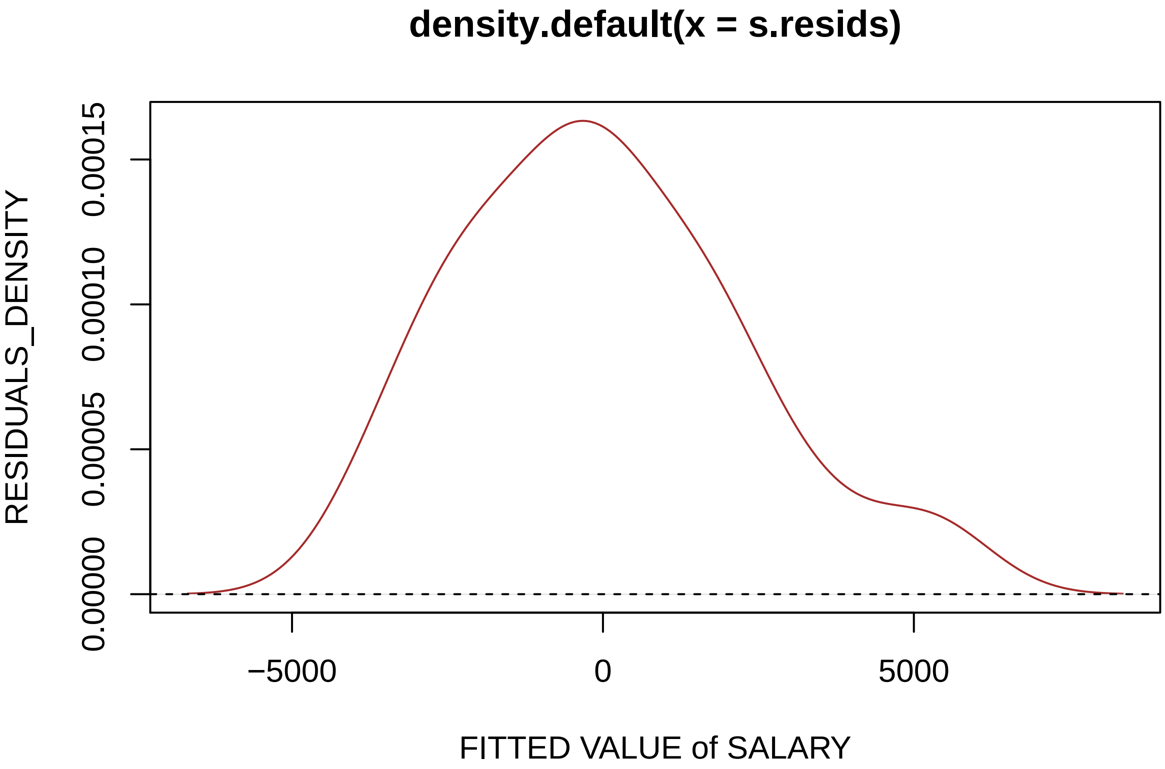
```
#Plotting residuals - Resid v FITTED VALUE using residuals()
#for comparison with the hand-calculated data
plot(s.fitted, s.resids , xlab='FITTED VALUE of SALARY',ylab='RESIDUALS', pch="+",
     col = 'orange')
abline(h=0,lty=2)
abline(h=2000, lty=2)
abline(h=-2000, lty=2)
title('RESIDUALS vs FITTED VALUES OF SALARY')
```



```
#Plotting residuals - Resid v FITTED VALUE using hand-calculated data
plot(y_hat, computed_residuals, xlab='FITTED VALUE of SALARY',ylab='RESIDUALS',
     pch="+", col = 'pink')
abline(h=0,lty=2)
abline(h=2000, lty=2)
abline(h=-2000, lty=2)
title('COMPUTED RESIDUALS vs FITTED VALUES COMPUTED')
```



```
#Plotting residuals - Residual density v. X
plot(density(s.resids), xlab='FITTED VALUE of SALARY',ylab='RESIDUALS_DENSITY',
     pch="+", col = 'brown')
abline(h=0,lty=2)
abline(h=2000, lty=2)
abline(h=-2000, lty=2)
```



It seems that our normality assumptions are roughly kept. It is centered around 0 and there is no visual pattern recognizable.

(I) Using the fitted model, predict what the average public teacher annual salary would be in a state where the spending per pupil is $4800.

```
y_new <- c(1, 4800/1000) %*% beta.hat
y_new
```

```
##          [,1]
## [1,] 28005.78
```

(J)

The prediction at an arbitrary new x value x^1_{new} can be wrriten in terms of the estimates $\widehat{\beta}$ as

$$\widehat{y}^{new} = x_1^{new}\widehat{\beta} = [1, x_1^{new}]\widehat{\beta} = \widehat{\beta}_0 + \widehat{\beta}_1 x_1^{new}$$

with $\widehat{\beta}$ the least squares estimate Compute the estimated standard prediction error for $\widehat{y}^{new}$, that is, the square root of the estimated variance.

From the class we know,

$$\mathrm{Var}_{\mathrm{Y|X,x^*}}[\widehat{\mathrm{Y}}_0^*|\mathrm{X},\mathrm{x}^*] = \mathrm{Var}_{\mathrm{Y|X,x^*}}[\widehat{\mathrm{Y}}^*|\mathrm{X},\mathrm{x}^*] + \mathrm{Var}_{\mathrm{Y|X,x^*}}[\epsilon|\mathrm{X},\mathrm{x}^*]$$

$$= x^*[\mathrm{Var}_{\mathrm{Y|X,x^*}}[\widehat{\beta}|\mathrm{X},\mathrm{x}^*]](x^*)^T + \sigma^2 = x^*[\mathrm{Var}_{\mathrm{Y|X,x^*}}[\widehat{\beta}|\mathrm{X}]](x^*)^T + \sigma^2 = x^*[\sigma^2(X^TX)^{-1}](x^*)^T + \sigma^2$$
$$= \sigma^2 h^* + \sigma^2 = \sigma^2(h^* + 1)$$

where

$$h^* = \frac{1}{n} + \frac{(x_1^* - \bar{x}_1)^2}{S_{xx}}$$

so Estimated Standard Prediction Error would be =

$$\sqrt{\mathrm{Var}_{\mathrm{Y|X,x^*}}\widehat{[\widehat{\mathrm{Y}}_0^*}|\mathrm{X},\mathrm{x}^*]} = \sqrt{\widehat{\sigma}^2(1 + \frac{1}{n} + \frac{(x_1^* - \bar{x}_1)^2)}{S_{xx}}}$$

```r
#We have sigma^2 = MS_res ready due to answers to previous questions
S_xx <- sum((x1-mean(x1))^2)
Prediction_var_est <- MS_res*(1/n+(4800/1000-mean(x1))^2/S_xx)
st.prediction_error <- sqrt(Prediction_var_est)
st.prediction_error
```

```
##          [,1]
## [1,] 473.5628
```