# STAT151A-HW6

*Xuanpei Ouyang*

*March 30, 2017*

Using R performs matrix computations, work with regression on the States Data. Take the States Data. Use the teacherpay as the Response variable and the Sat Math and Percentage as the Explanatory variable.

### (a) Compute the least-squares regression coefficients, $b = (X'X)^{-1}X'y$

```
States = read.table("~/Desktop/STAT 151A/STAT-151A/hw/hw6/States.txt")
summary(States)
```

```
##      region     population      satVerbal        satMath
##  SA     : 9   Min.   :  481   Min.   :480.0   Min.   :473.0
##  MTN    : 8   1st Qu.: 1423   1st Qu.:502.5   1st Qu.:500.0
##  WNC    : 7   Median : 3699   Median :525.0   Median :521.0
##  NE     : 6   Mean   : 5202   Mean   :531.9   Mean   :529.3
##  ENC    : 5   3rd Qu.: 5966   3rd Qu.:564.5   3rd Qu.:557.0
##  PAC    : 5   Max.   :31878   Max.   :596.0   Max.   :600.0
##  (Other):11
##  percentTaking   percentNoHS      teacherPay
##  Min.   : 4.00   Min.   :13.40   Min.   :26.30
##  1st Qu.: 9.00   1st Qu.:19.90   1st Qu.:31.55
##  Median :30.00   Median :23.30   Median :35.00
##  Mean   :35.49   Mean   :23.78   Mean   :35.89
##  3rd Qu.:61.00   3rd Qu.:26.50   3rd Qu.:40.05
##  Max.   :80.00   Max.   :35.70   Max.   :50.30
##
```

```
States <- as.data.frame(unclass(States))
attach(States)

X = as.matrix(cbind(1, States$satMath, States$percentTaking))
Y = as.matrix(States$teacherPay)
head(X)
```

```
##      [,1] [,2] [,3]
## [1,]    1  558    8
## [2,]    1  513   47
## [3,]    1  521   28
## [4,]    1  550    6
## [5,]    1  511   45
## [6,]    1  538   30
```

```
head(Y)
```

```
##      [,1]
## [1,] 31.3
## [2,] 49.6
## [3,] 32.5
## [4,] 29.3
## [5,] 43.1
## [6,] 35.4
```

```r
## Estimated Slope Coefficients matrix b = (X'X)^{-1}X'y
beta_hat = solve(t(X)%*%X) %*% t(X) %*% Y
beta_hat_coefficient = as.data.frame(cbind(
  c("Intercept","satMath","PercentageTaking"), beta_hat))
names(beta_hat_coefficient) = c("Slope Coefficient","Estimates")
beta_hat_coefficient
```

```
##   Slope Coefficient          Estimates
## 1         Intercept  -15.1576165055178
## 2           satMath 0.0806665874796603
## 3  PercentageTaking  0.235362014333382
```

Therefore,

$$b_0 = -15.1576165055178, b_1 = 0.0806665874796603, b_2 = 0.235362014333382`$$

**(b)Calculate the estimated error variance, $s_e^2 = \frac{e'e}{(n-k-1)}$(where $e = y - Xb$), and the estimated covariance matrix of the coefficients, $V(b) = s_e^2(X'X)^{-1}$**

```r
# first calculate the residuals
residuals = as.matrix(States$teacherPay - beta_hat[1] -
                        beta_hat[2]*States$satMath -
                        beta_hat[3]*States$percentTaking)
head(residuals)
```

```
##             [,1]
## [1,] -0.43723542
## [2,] 12.31364245
## [3,] -0.95981197
## [4,] -1.32117869
## [5,]  6.44569966
## [6,]  0.09813201
```

```r
# then we can calculate the estimated covariance matrix of the coeffients
n = nrow(States)  # number of data points
k = ncol(X)  # number of parameters

# calculate the estimated error variance
SE_variance = (t(residuals)%*%residuals) / (n-k-1)
SE_variance
```

```
##          [,1]
## [1,] 23.95391
```

```r
# calculate the variance-covariance matrix
VCV_matrix = as.numeric(SE_variance) * solve(t(X)%*%X)
VCV_matrix
```

```
##             [,1]         [,2]         [,3]
## [1,] 495.8684743 -0.867961344 -1.014617531
## [2,]  -0.8679613  0.001523393  0.001737607
## [3,]  -1.0146175  0.001737607  0.002675280
```

Therefore, the estimated error variance is

$$23.95391$$

and the variance-covariance matrix is shown above.

**(c) Calculate the coefficient Standard Error for this model**

```
# take the square root of variance-covariance matrix to find standard
# errors of the estimated coeffients
StdErr = sqrt(diag(VCV_matrix))
StdErr
```

```
## [1] 22.26810442  0.03903067  0.05172311
```

The coefficient standard error are

$$[22.26810442, 0.03903067, 0.05172311]$$

**(d) Verify that the lm() model provides us with the same t and pvalues as the matrix formulation.**

```
# conduct the individual hypothesis for the esimated coeffients
# we calculate the t values
t_value = rbind(beta_hat[1]/StdErr[1],
          beta_hat[2]/StdErr[2],
          beta_hat[3]/StdErr[3])
t_value
```

```
##            [,1]
## [1,] -0.6806873
## [2,]  2.0667486
## [3,]  4.5504228
```

```
# calculate the p-value for t test for determining coefficient significance
p_value = rbind(2*pt(abs(beta_hat[1]/StdErr[1]), df=n-k, lower.tail= FALSE),
                2*pt(abs(beta_hat[2]/StdErr[2]), df=n-k, lower.tail= FALSE),
                2*pt(abs(beta_hat[3]/StdErr[3]), df=n-k, lower.tail= FALSE))

p_value
```

```
##              [,1]
## [1,] 4.993399e-01
## [2,] 4.417476e-02
## [3,] 3.656745e-05
```

```
# create a table summary of the matrix formulation and t test
matrix_summary = data.frame(
  "Slope Estimate" = beta_hat,
  "Standard Errors" = StdErr,
  "t value" = t_value,
  "p value" = p_value
)
matrix_summary
```

```
##   Slope.Estimate Standard.Errors    t.value      p.value
## 1   -15.15761651     22.26810442 -0.6806873 4.993399e-01
## 2     0.08066659      0.03903067  2.0667486 4.417476e-02
## 3     0.23536201      0.05172311  4.5504228 3.656745e-05
```

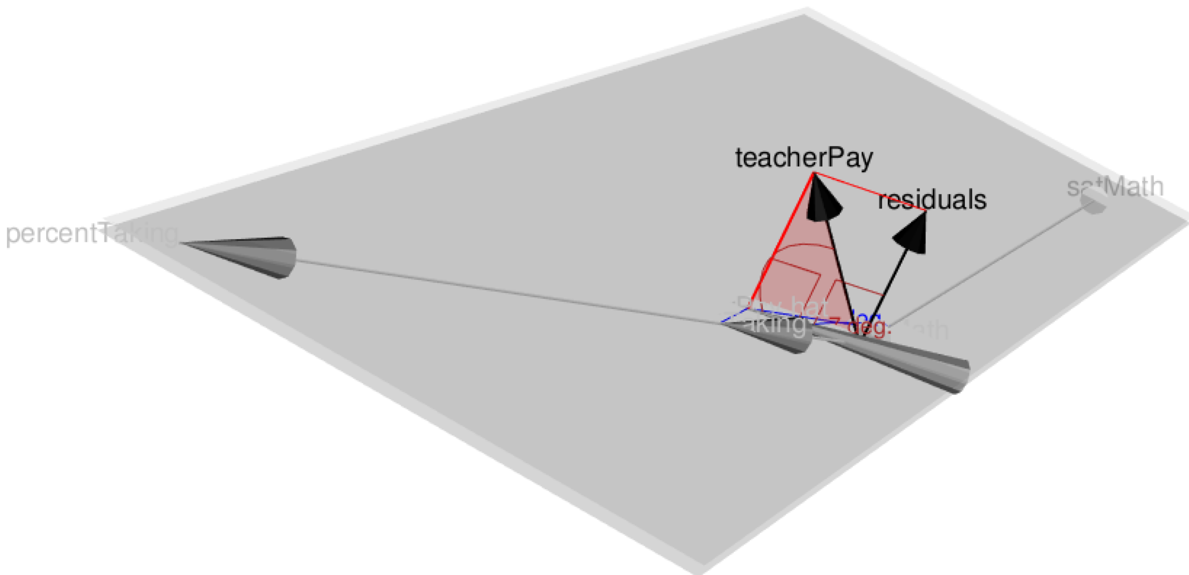
```
estimated_model<-lm(teacherPay~satMath+percentTaking,data=States)
estimated_model_summary = summary(estimated_model)
lm_result = estimated_model_summary$coefficients
lm_result
```

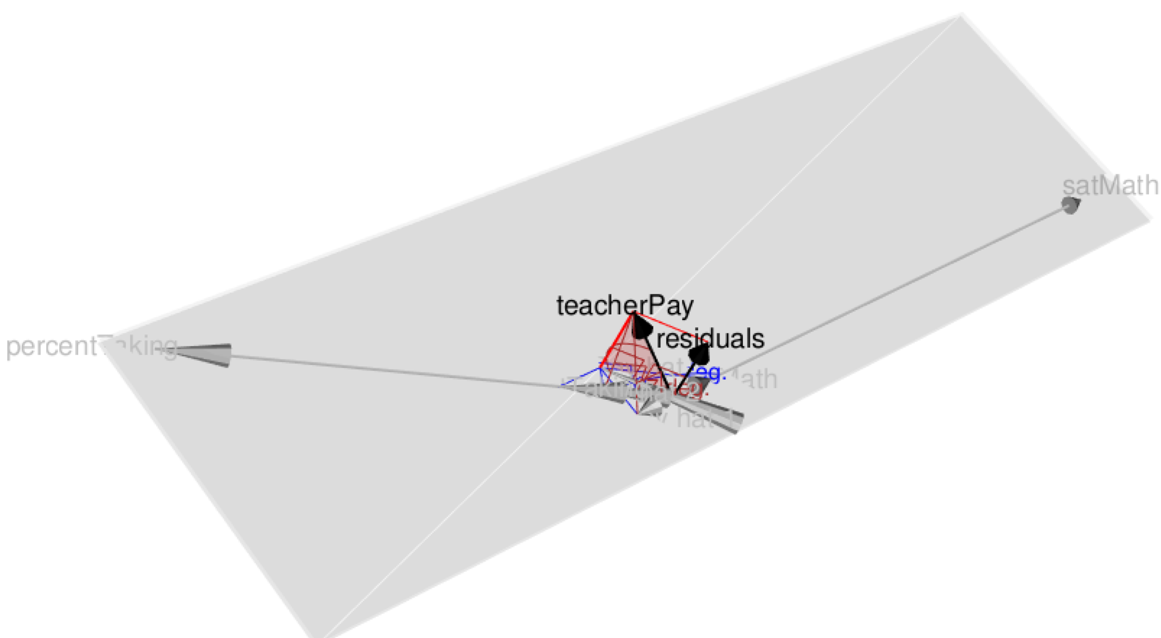
```
##                  Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)   -15.15761651 22.03492412 -0.6878906 4.948326e-01
## satMath         0.08066659  0.03862196  2.0886195 4.206963e-02
## percentTaking   0.23536201  0.05118149  4.5985768 3.116289e-05
```

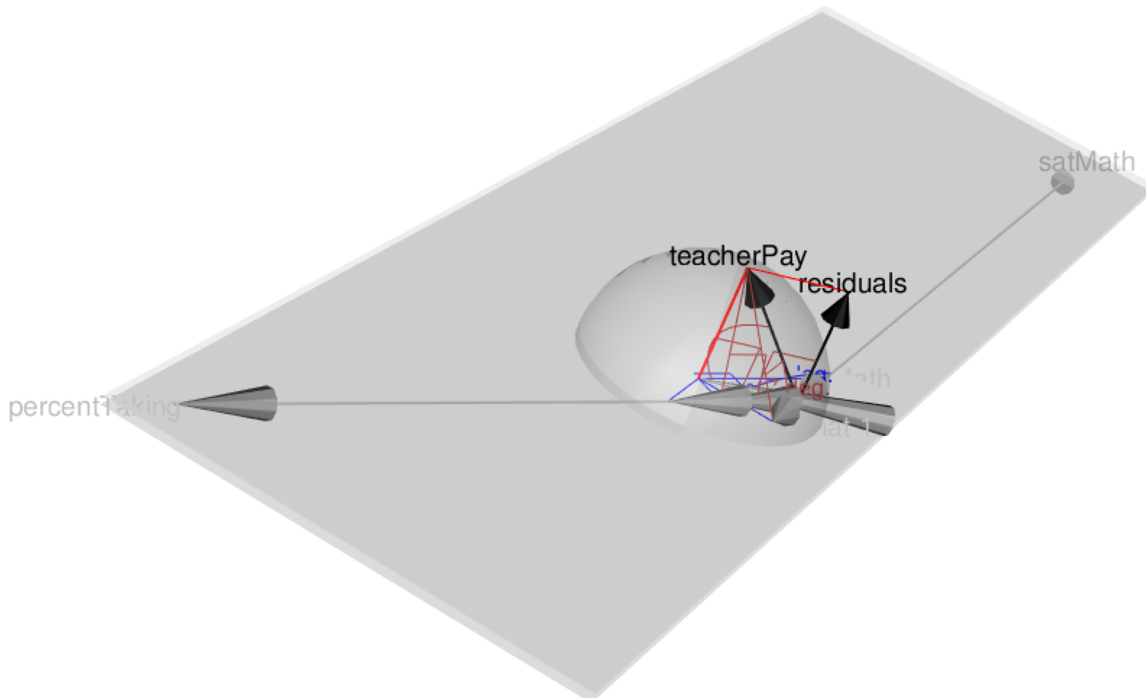
Therefore, by conducting individual t-test for estimated coeffients, we can see that the t values and p values p otained from lm() function and individual t-test are approximately equal (when we around the results to one decimal place).

### (e) Create a 3d vector geometric representation for this data.

```r
states_vecs = regvec3d(teacherPay ~ satMath + percentTaking, data=States)

# plot 3D vector geometric representation for the regression model and data
plot(states_vecs)

# with the marginal regression
plot(states_vecs, show.marginal = TRUE)

# show the 3D projection of the error hypersphere, scaled so that its
# projections on the x axes show confidence intervals for the standardized
# regression coeffients
plot(states_vecs, show.marginal = TRUE, error.sphere="y.hat")
```


teacherPay
residuals
satMath
percentTaking


satMath
teacherPay
residuals
percentTaking


satMath
teacherPay
residuals
percentTaking