Advanced Methods Homework I

Bohao Tang

1 Inference and estimation in linear models

1.

(a) $Y_i \sim N(\xi+\mu, \tau^2+\sigma^2)$ according to the additivity of normal distribution

(b) $E[Y_iY_j] = E[U^2+UZ_i+UZ_j+Z_iZ_j]$

$= \xi^2+\tau^2+\xi\mu+\mu\xi+\mu^2$.

$\Rightarrow \operatorname{cov}(Y_i, Y_j) = E[Y_iY_j]-E[Y_i]E[Y_j]$

$= \xi^2+\tau^2+\mu^2+2\xi\mu-\xi^2-\mu^2-2\xi\mu$.

$= \tau^2$

(c) $$\vec{Y} = \begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 1 & 0 \cdots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ 1 & 0 & \cdots & & 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{pmatrix} = [J_n \ I_n] \begin{pmatrix} U \\ Z_1 \\ \vdots \\ Z_n \end{pmatrix}$$

where $$\begin{pmatrix} U \\ Z_1 \\ \vdots \\ Z_n \end{pmatrix} \sim N\left[\begin{pmatrix} \xi \\ \mu \\ \vdots \\ \mu \\ \vdots \\ \mu \end{pmatrix}, \begin{pmatrix} \tau^2 & & & 0 \\ & \sigma^2 & & \\ & & \sigma^2 & \\ 0 & & & \ddots \ \sigma^2 \end{pmatrix}\right]$$

$$\Rightarrow \vec{Y} \sim N\left([J_n \ I_n]\begin{pmatrix} \xi \\ \mu \\ \vdots \\ \mu \end{pmatrix}, [J_n \ I_n]\begin{pmatrix} \tau^2 & & 0 \\ & \sigma^2 & \\ 0 & & \ddots \ \sigma^2 \end{pmatrix}\begin{pmatrix} J_n' \\ I_n \end{pmatrix}\right)$$

$$= N\left[\begin{pmatrix} \xi+\mu \\ \xi+\mu \\ \vdots \\ \xi+\mu \end{pmatrix}, \begin{pmatrix} \tau^2+\sigma^2 & \tau^2 & \cdots & \tau^2 \\ \tau^2 & \tau^2+\sigma^2 & \tau^2 \cdots & \tau^2 \\ \tau^2 & \tau^2 & \ddots & \tau^2 \\ \vdots & \vdots & & \vdots \\ \tau^2 & \tau^2 & \cdots & \tau^2+\sigma^2 \end{pmatrix}\right]$$

(d) $\cdot\ E[\bar{Y}] = \frac{1}{n}\sum_{i=1}^{n} EY_i = \xi + \mu = E[Y_1]$

so $\bar{Y}$ is a unbiased estimactor for $E[Y_1]$

(e) $(n-1)S^2 = (\vec{Y} - J_n(J_n'J_n)^{-1}J_n'\vec{Y})'(\vec{Y} - J_n(J_n'J_n)^{-1}J_n'\vec{Y})$

$= \vec{Y}'[I - J_n(J_n'J_n)^{-1}J_n']\vec{Y}$

where $I - J_n(J_n'J_n)^{-1}J_n'$ is a projection matrix

(f) $I - J_n(J_n'J_n)^{-1}J_n' = \begin{pmatrix} 1-\frac{1}{n}, & -\frac{1}{n}, & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n}, & 1-\frac{1}{n}, & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & \ddots & -\frac{1}{n} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \\ -\frac{1}{n} & \cdots & & & 1-\frac{1}{n} \end{pmatrix}$

So $(I - J_n(J_n'J_n)^{-1}J_n')\cdot var[\vec{Y}]$

$= \begin{pmatrix} (1-\frac{1}{n})\sigma^2, & -\frac{1}{n}\sigma^2, & -\frac{1}{n}\sigma^2 & \cdots \\ -\frac{1}{n}\sigma^2 & (1-\frac{1}{n})\sigma^2, & -\frac{1}{n}\sigma^2 & \cdots \\ & \ddots & \ddots & \\ & & \ddots & \ddots \end{pmatrix} = \sigma^2[I - J_n(J_n'J_n)^{-1}J_n']$

So $\frac{n-1}{\sigma^2}S^2 = \vec{Y}'\frac{P}{\sigma^2}\vec{Y}$ where $\frac{P}{\sigma^2}var(Y)$ is idempotent of rank $n-1$

therefore $\frac{n-1}{\sigma^2}S^2 \sim \chi^2(n-1, \frac{1}{2\sigma^2}E[\vec{Y}]'[I - J_n(J_n'J_n)^{-1}J_n']E[\vec{Y}])$

$= \chi^2(n-1)$

$S^2 \sim \frac{\sigma^2}{n-1}\chi^2(n-1)$

(g) $ES^2 = \frac{\sigma^2}{n-1}E\chi^2(n-1) = \sigma^2 \neq \sigma^2 + \tau^2 = var[Y_1]$

(h) $V = var(\vec{Y})$ then $V = \begin{pmatrix} \tau^2+\sigma^2 & \tau^2 & \tau^2 & \cdots & \tau^2 \\ \tau^2 & \tau^2+\sigma^2 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \tau^2 \\ \tau^2 & \cdots & \cdots & \tau^2 & \tau^2+\sigma^2 \end{pmatrix} = \tau^2(J_nJ_n') + \sigma^2 I_n$

$$\det(V) = \det\begin{pmatrix} n\tau^2+\sigma^2, & n\tau^2+\sigma^2 & \cdots & n\tau^2+\sigma^2 \\ \tau^2, & \tau^2+\sigma^2, & \tau^2 \cdots & \tau^2 \\ \vdots & & \ddots & \tau^2 \\ \tau^2 & \cdots & \tau^2 & \tau^2+\sigma^2 \end{pmatrix} \quad \text{(plus row } 2,3\cdots n \text{ to row } 1)$$

$$= \cancel{n\tau^2+\sigma^2} \, (n\tau^2+\sigma^2) \det\begin{pmatrix} 1, & 1, & 1 & \cdots & 1 \\ \tau^2, & \tau^2+\sigma^2, & \tau^2 & \cdots & \tau^2 \\ \vdots & & \ddots & & \vdots \\ \tau^2 & \cdots & & \tau^2 & \tau^2+\sigma^2 \end{pmatrix}$$

$$= (n\tau^2+\sigma^2) \det\begin{pmatrix} 1, & 1, & 1 & \cdots & 1 \\ 0 & \sigma^2 & & & 0 \\ 0 & & \sigma^2 & & \\ \vdots & & & \ddots & \\ 0 & \cdots & & 0 & \sigma^2 \end{pmatrix} \quad \text{(row } 2,3\cdots n \text{ subtract } \tau^2 \text{ times row } 1)$$

$$= (\sigma^2)^{n-1}(n\tau^2+\sigma^2) = n\tau^2\sigma^{2n-2} + \sigma^{2n}.$$

$$V^{-1} = \left(\frac{1}{\sigma^2} I_n - \frac{\tau^2}{\sigma^2(n\tau^2+\sigma^2)} J_nJ_n'\right)$$

(where we find that $\left(\frac{1}{\sigma^2} I_n - \frac{\tau^2}{\sigma^2(n\tau^2+\sigma^2)} J_nJ_n'\right) \cdot (\tau^2 J_nJ_n' + \sigma^2 I_n)$

$$= I_n - \frac{\tau^2}{n\tau^2+\sigma^2} J_nJ_n' + \frac{\tau^2}{\sigma^2} J_nJ_n' - \frac{n\tau^4}{\sigma^2(n\tau^2+\sigma^2)} J_nJ_n'$$

$= I_n$)

2.
(a) Within every group, S_i is unbiased estimator of σ^2

so $E[S_p^2] = \frac{1}{J_1+J_2-2}\left[(J_1-1)\sigma^2 + (J_2-1)\sigma^2\right] = \sigma^2$, so S_p^2 unbiased.

also for here $\vec{Y} = \begin{pmatrix} J_{J_1} & 0 \\ 0 & J_{J_2} \end{pmatrix}\begin{pmatrix}\mu_1 \\ \mu_2\end{pmatrix} + \vec{\varepsilon} = X\begin{pmatrix}\mu_1 \\ \mu_2\end{pmatrix} + \vec{\varepsilon}$

We ~~find~~ found out in last homework that $S_p^2 = \frac{1}{n-2}(\vec{Y} - X\hat{\beta})'(\vec{Y} - X\hat{\beta})$
where $n = J_1+J_2$

therefore it is independent with $\hat{\beta}$

We then have that :

$$\frac{(1,-1)\begin{pmatrix}\hat{\mu}_1\\ \hat{\mu}_2\end{pmatrix} - (1,-1)\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}}{\sqrt{S_p^2}\sqrt{(1,-1)(X'X)^{-1}\begin{pmatrix}1\\ -1\end{pmatrix}}} \sim t_{n-2} = t_{J_1+J_2-2}.$$

$$\Rightarrow \quad \frac{(\hat{\mu}_1 - \hat{\mu}_2) - (\mu_1 - \mu_2)}{S_p\sqrt{\frac{J_1+J_2}{J_1J_2}}} \sim t_{J_1+J_2-2}.$$

$\Rightarrow$ a $100(1-\alpha)\%$ confidence interval for $\mu_1-\mu_2$ is

$$(\hat{\mu}_1 - \hat{\mu}_2) \pm t_{J_1+J_2-2,\,1-\frac{\alpha}{2}} \cdot S_p\sqrt{\frac{J_1+J_2}{J_1J_2}},$$

where $\hat{\mu}_1 = \frac{1}{J_1}\sum_{j=1}^{J_1} Y_{1j}$ $\quad \hat{\mu}_2 = \frac{1}{J_2}\sum_{j=1}^{J_2} Y_{2j}$

To test $H_0: \mu_1 = \mu_2$, compute $T = \hat{\mu}_1 - \hat{\mu}_2$

if $|t| > t_{J_1+J_2-2,\,1-\frac{\alpha}{2}} \cdot S_p\sqrt{\frac{J_1+J_2}{J_1J_2}}$

then reject the null hypothesis.

this test is of level α.

(b) Here $\vec{Y} = \begin{pmatrix} J_{J_1} & 0 & \cdots & 0 \\ 0 & J_{J_2} & & \vdots \\ 0 & 0 & \ddots & J_{J_I} \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_I \end{pmatrix} + \vec{\varepsilon}$

$= X \vec{\mu} + \vec{\varepsilon}$

To test $H_0: \mu_1 = \mu_2 = \cdots = \mu_I$.

it's equivalent to test $\cancel{(I_I - J_I(J_I'J_I)^{-1}J_I')\vec{\mu} = \vec{0}}$

$\Leftrightarrow K\vec{\mu} = \vec{0}$ set $K = \begin{pmatrix} 1, & -1 & & 0 \\ 1, & 0, & -1 & \\ \vdots & \vdots & & \ddots \\ 1, & 0 & \cdots & 0 & -1 \end{pmatrix} = (\vec{J}_{I-1}, -I_{I-1})$.

We have that
$$\frac{(K\hat{\mu})'\{K(X'X)^{-1}K'\}^{-1}(K\hat{\mu})}{(I-1)\cdot \dfrac{(\vec{Y}-X\hat{\mu})'(\vec{Y}-X\hat{\mu})}{n-I}} \sim F_{I-1,\, n-I}(\lambda)$$

where $n = J_1 + J_2 + \cdots J_I$

$\lambda = (K\vec{\mu})'\{K(X'X)^{-1}K'\}^{-1}K\vec{\mu}/2\sigma^2$

Where H_0 holds $\lambda = 0$, and compute the left side out we get.

$$\frac{(\vec{Y}-X\hat{\mu})'(\vec{Y}-X\hat{\mu})}{n-I} = \frac{1}{n-I}\sum_{i=1}^{I}(J_i-1)S_i^2 \triangleq S^2_{within}$$

Where S_i^2 is the standard variance estimate within group i

And $K(X'X)^{-1}K' = \begin{pmatrix} J_2^{-1} & & 0 \\ & J_3^{-1} & \\ 0 & & \ddots & J_I^{-1} \end{pmatrix} + J_1^{-1}\vec{J}_{I-1}\vec{J}_{I-1}'$ where $\vec{J}_{I-1}' = \underbrace{(1,1,\cdots 1)}_{I-1}$

then $\{K(X'X)^{-1}K'\}^{-1} = \begin{pmatrix} J_2 & & 0 \\ & J_3 & \\ 0 & & \ddots & J_I \end{pmatrix} - \frac{1}{n}\begin{pmatrix} J_2 \\ J_3 \\ \vdots \\ J_I \end{pmatrix}(J_2, J_3, \cdots J_I)$

notify the the sample mean of all samples $\bar{\mu} = \frac{1}{n} \sum J_i \hat{\mu}_i$

After simplification we have that:

$$\frac{(K\hat{\mu})' \{K(X'X)^{-1}K'\}^{-1}(K\hat{\mu})}{I-1} = \frac{\sum_{i=1}^{I} J_i (\hat{\mu}_i - \bar{\mu})^2}{I-1} \triangleq S^2_{between}$$

Then we are testing the statistics $\frac{S^2_{between}}{S^2_{within}}$ which is the ratio of variation between and within groups.

3: $$\hat{\beta}_1 = \frac{n\sum X_i Y_i - \sum X_i \sum Y_i}{n\sum X_i^2 - (\sum X_i)^2}$$

$Y_i \sim N(\beta_0 + \beta_1 X_i, \sigma^2)$ and Y_i are independent.

$$= \frac{\frac{\sum X_i Y_i}{n} - \frac{\sum X_i}{n} \cdot \frac{\sum Y_i}{n}}{\frac{\sum X_i^2}{n} - \left(\frac{\sum X_i}{n}\right)^2}$$

$$var(\hat{\beta}_1) = \left(\frac{\sum X_i^2}{n} - \left(\frac{\sum X_i}{n}\right)^2\right)^{-2} \cdot \left(\frac{\sum X_i^2 \cdot \sigma^2}{n^2} - \left(\frac{\sum X_i}{n}\right)^2 \cdot \frac{\sigma^2}{n}\right)$$

$$= \left[\frac{\sum X_i^2}{n} - \left(\frac{\sum X_i}{n}\right)^2\right]^{-1} \frac{\sigma^2}{n} = \left[var(X) \cdot \frac{n-1}{n}\right]^{-1} \frac{\sigma^2}{n} = \frac{\sigma^2}{n-1} \cdot [var(X)]^{-1}$$

So when $var(X)$ reachs its maximum, $var(\hat{\beta}_1)$ reaches its minimum.

If $\{X_i\}$ is from a large range, i.e. $X_{(n)} - X_{(1)}$ is large.
and n, the number of X_i is large, the $var(X)$ tend to be large
then we can get lower variance estimate
when $X_{(n)} - X_{(1)}$ and n are given, if X_i concertrate on the two side $X_{(1)}, X_{(n)}$
then we get the lowest variance estimate.

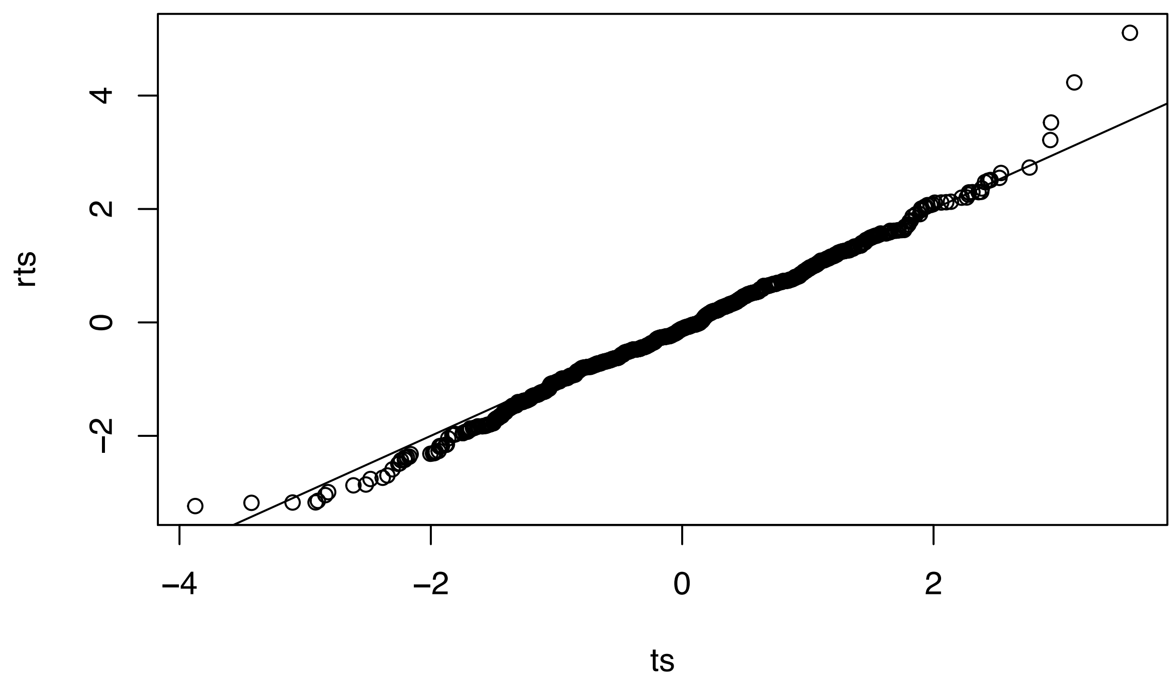
Coding and data analysis exercises

1.

```
samples = rnorm(1000*20, 5, sqrt(2))
samples = matrix(samples,1000,20)
ms = apply(samples, 1, mean)
stds = sqrt(apply(samples, 1, var))
ts = sqrt(19) * (ms - 5)  / stds

rts = rt(1000, 19)

qqplot(ts, rts)
abline(0,1)
```



As shown in the graph, the quantiles agree well, since every t statistics is computed from 20 i.i.d normal samples, it will follow a t distribution with 19 df. So the quantiles will agree.

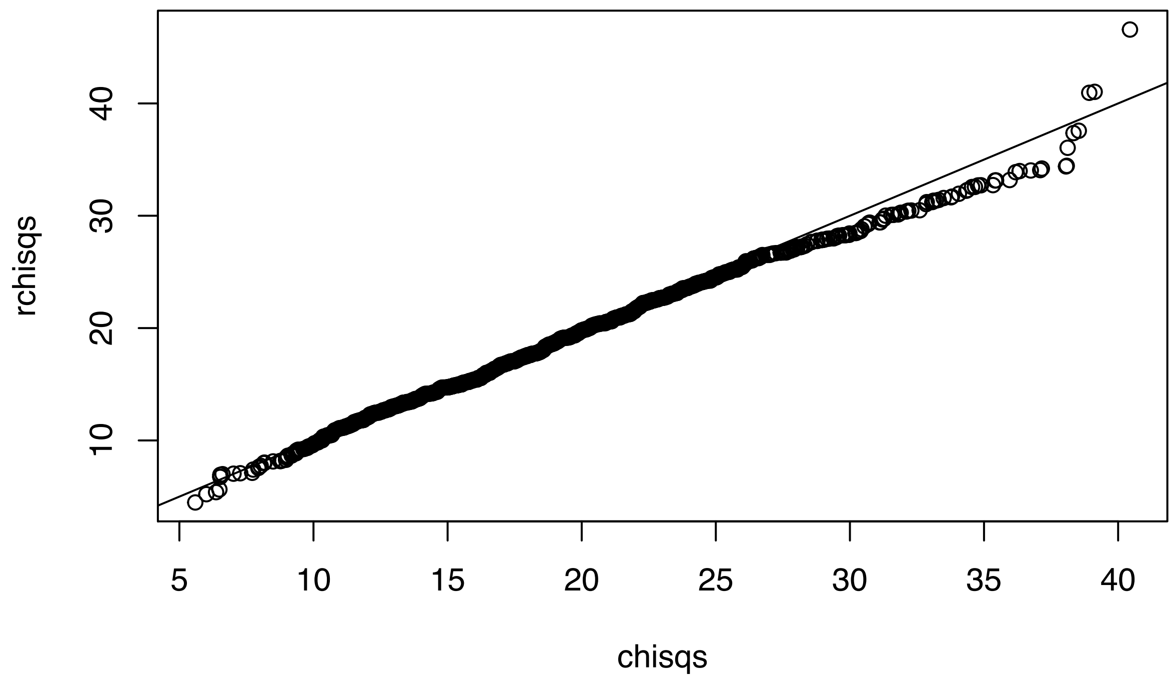
2.

```
samples = rnorm(1000*20, 5, sqrt(2))
samples = matrix(samples,1000,20)
vs = apply(samples, 1, var)
```

```
chisqs = 19 * vs / 2

rchisqs = rchisq(1000, 19)

qqplot(chisqs, rchisqs)
abline(0,1)
```



As shown in the graph, the quantiles agree well, also because they are just from the same distribution.

3.

```
require(stats)

mylm <- function(Y,X){

    Y = as.matrix(Y); Xnew = as.matrix(X)

    # Check if numeric
    if(!is.numeric(Y) | !is.numeric(Xnew))
        stop("Y or X is not numeric!\n")

    # Check for dimensions
    dy = dim(Y) ; dx = dim(Xnew)
    if(dy[2] != 1 | dy[1] != dx[1])
        stop("Y or X has wrong dimensions\n")
```

```r
# Check for ill conditioned elements
# we can use is.finite to response only to finite real numbers
if(FALSE %in% is.finite(Y) | FALSE %in% is.finite(Xnew))
    warning("Y or X is ill conditioned\n")

# Check if of full rank
D = cbind(1,Xnew)
DtD = t(D) %*% D
if(det(DtD) == 0)
    stop("Design matrix is not full rank\n")

# Regressing
DtD.inv = solve(DtD)
hat.matrix = D %*% DtD.inv %*% t(D)
beta = DtD.inv %*% t(D) %*% Y
fitted = D %*% beta
residuals = Y - fitted

SS.tot = sum((Y - mean(Y))^2)
if(SS.tot == 0)
    warning("Y is constant!\n")
SS.res = sum((Y - fitted)^2)
SS.reg = SS.tot - SS.res
R2 = SS.reg / SS.tot

df = dim(D)[1] - dim(D)[2]
df2 = dim(D)[2] - 1

s2 = SS.res / df
std_error = sqrt(s2 * diag(DtD.inv))
t_value = beta / std_error
P_value = 1 - pt(abs(t_value), df) + pt(-abs(t_value), df)

K = cbind(rep(0, df2), diag(df2))
Kbeta = K %*% beta
Fstat = t(Kbeta) %*% solve(K %*% DtD.inv %*% t(K)) %*% Kbeta
Fstat = Fstat / (df2 * s2)
P_value_F = 1 - pf(Fstat, df2, df)

beta_names = c("(Interception)")
for(i in 1:(dim(D)[2] - 1)){
    beta_names = c(beta_names, sprintf("beta%d",i))
}
t_summary = data.frame("Estimate"=beta, "Std.Error"=std_error,
                       "t.value"=t_value, "Pvalue"=P_value)
row.names(t_summary) = beta_names

summary <- function(){
    cat("T table:\n")
    print(t_summary)
    cat("\nOverall F test:\n")
    cat(sprintf("F-statistics: %f on %d and %d DF,  p-value: %f",
                Fstat, df2, df, P_value_F))
```

```r
    }

    # Return result
    result = list(beta = beta,
                  fitted = fitted,
                  residuals = residuals,
                  R2 = R2,
                  hatdiag = diag(hat.matrix),
                  summary = summary)
    return(result)
}

test.X = cbind(sample(1:100),sample(1:100),sample(1:100))
beta = c(5,-1,0.01,2)
test.y = cbind(1,test.X) %*% beta + rnorm(100,0,5)
model = lm(test.y ~ test.X)
summary(model)
```

```
##
## Call:
## lm(formula = test.y ~ test.X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8314  -3.3723  -0.1633   3.3739  12.5185
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.973250   1.657447   3.001  0.00343 **
## test.X1     -0.987051   0.017115 -57.671  < 2e-16 ***
## test.X2      0.005103   0.017119   0.298  0.76630
## test.X3      2.019717   0.016818 120.090  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.832 on 96 degrees of freedom
## Multiple R-squared:  0.9944, Adjusted R-squared:  0.9942
## F-statistic:  5666 on 3 and 96 DF,  p-value: < 2.2e-16
```

```r
mymodel = mylm(test.y, test.X)
mymodel$summary()
```

```
## T table:
##                    Estimate  Std.Error     t.value        Pvalue
## (Interception)  4.973249946 1.65744736   3.0005477  3.433648e-03
## beta1          -0.987051101 0.01711515 -57.6711825  1.313666e-76
## beta2           0.005102669 0.01711925   0.2980661  7.662969e-01
## beta3           2.019716782 0.01681843 120.0895022 9.733797e-107
##
## Overall F test:
## F-statistics: 5666.304005 on 3 and 96 DF,  p-value: 0.000000
```