Econometric Methods Homework 3

Chinying Lin

October 15, 2024

Set Up

```
library(sandwich)

Data <- read.csv("Equity_Premium.csv")
Matrix <- as.matrix(Data)
Y <- Matrix[, 2, drop = FALSE]
X1 <- matrix(1, nrow = nrow(Y), ncol = 1)
X_dfy <- Matrix[, 3, drop = FALSE]
X_infl <- Matrix[, 4, drop = FALSE]
X_svar <- Matrix[, 5, drop = FALSE]
X_tms <- Matrix[, 6, drop = FALSE]
X_tbl <- Matrix[, 7, drop = FALSE]
X_dfr <- Matrix[, 8, drop = FALSE]
X_db <- Matrix[, 9, drop = FALSE]
X_ltr <- Matrix[, 10, drop = FALSE]
X <- cbind(X1, X_dfy, X_infl, X_svar, X_tms, X_tbl, X_dfr, X_db, X_ltr)

UpdateName <- function(coef){
  new_names <- names(coef)
  new_names[1] <- "constant"
  new_names <- sub("^Xx_", "x_", new_names)
  new_names <- sub("^X", "", new_names)
  names(coef) <- new_names

  return(coef)
}

model <- lm(Y ~ X-1)
```

Problem 1

β

```
beta <- model$coefficients
```

```
> print(UpdateName(beta))
   constant       x_dfy      x_infl      x_svar       x_tms       x_tbl       x_dfr        x_dp       x_ltr 
 0.24602183 -0.81627730 -0.25872644 -0.19365928 -0.24649239 -0.25348610  0.27006370  0.05099741  0.13180038 
```

$S(\hat{\beta}_j)$

```
s <- summary(model)$sigma
S_beta <- s * sqrt(diag(solve(t(X) %*% X)))
```

```
> print(UpdateName(S_beta))
  constant      x_dfy     x_infl     x_svar      x_tms      x_tbl      x_dfr       x_dp      x_ltr 
0.04265661 0.59660216 0.63135837 0.38033409 0.18030425 0.09817880 0.14728129 0.00912120 0.07367261 
```

$S^W(\hat{\beta}_j)$

```
Vw <- vcovHC(model, type = "HC")
SW_beta <- sqrt(diag(Vw))
```

```
> print(UpdateName(SW_beta))
   constant       x_dfy      x_infl      x_svar       x_tms       x_tbl       x_dfr        x_dp       x_ltr 
0.045786860 0.820993132 0.657849477 0.810762836 0.188770481 0.107648965 0.240869076 0.009811133 0.098061658 
```

Problem 2

```
betaJ <- summary(model)$coefficients[, "Estimate"]
sBetaJ <- summary(model)$coefficients[, "Std.␣Error"]
t_statistic <- abs(betaJ / sBetaJ)

df <- length(model$residuals) - length(t_statistic)
alpha_1 <- qt(1 - 0.01 / 2, df)
alpha_5 <- qt(1 - 0.05 / 2, df)
alpha_10 <- qt(1 - 0.10 / 2, df)

result_df <- data.frame(
  t_statistic = t_statistic,
  alpha_1_significant = t_statistic > alpha_1,
  alpha_5_significant = t_statistic > alpha_5,
  alpha_10_significant = t_statistic > alpha_10
)
```

```
> print(result_df)
        t_statistic alpha_1_significant alpha_5_significant alpha_10_significant
X         5.7674958                TRUE                TRUE                 TRUE
Xx_dfy    1.3682104               FALSE               FALSE                FALSE
Xx_infl   0.4097933               FALSE               FALSE                FALSE
Xx_svar   0.5091820               FALSE               FALSE                FALSE
Xx_tms    1.3670914               FALSE               FALSE                FALSE
Xx_tbl    2.5818824               FALSE                TRUE                 TRUE
Xx_dfr    1.8336592               FALSE               FALSE                 TRUE
Xx_dp     5.5910859                TRUE                TRUE                 TRUE
Xx_ltr    1.7890011               FALSE               FALSE                 TRUE
```

Problem 3

3 - 1

JB

```
variance_hat <- sum(resid(model)^2) / length(resid(model))
std_hat <- sqrt(variance_hat)
sk_hat <- sum((resid(model) / std_hat)^3) / length(resid(model))
kr_hat <- sum((resid(model) / std_hat)^4) / length(resid(model))
JB <- length(resid(model)) * ((sk_hat^2 / 6) + ((kr_hat - 3)^2 / 24))
```

```
> print(JB)
[1] 74.89123
```

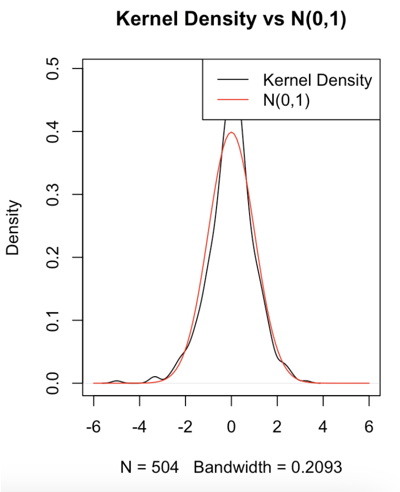
Compare with different α

```
chi_1 <- qchisq(1 - 0.01, 2)   # alpha = 1%
chi_5 <- qchisq(1 - 0.05, 2)   # alpha = 5%
chi_10 <- qchisq(1 - 0.10, 2)   # alpha = 10%
```

```
> print(paste("Significant at 1% level:", JB > chi_1))
[1] "Significant at 1% level: TRUE"
> print(paste("Significant at 5% level:", JB > chi_5))
[1] "Significant at 5% level: TRUE"
> print(paste("Significant at 10% level:", JB > chi_10))
[1] "Significant at 10% level: TRUE"
```

3 - 2

```
standardized_residuals <- resid(model) / std_hat
density_res <- density(standardized_residuals)
plot(density_res, main = "Kernel␣Density␣vs␣N(0,1)", xlim = c(-6, 6))
curve(dnorm(x), col = "red", add = TRUE)
legend("topright", legend = c("Kernel␣Density", "N(0,1)"), col = c("black", "red"), lty = 1)
```



GitHub Link

EconometricMethods-homework5-b10901069