Problem Set 1

Nima Mohammadi
nimamo@vt.edu

February 11, 2020

Question 1: Omitting regressors under independence and dependence

Part A: Independence, Full Model

1. Generate two independent, normally distributed regressors (= explanatory variables), one with mean 2 and std 1, the other with mean 3 and std 1. Set the sample size to 1000 observations in each case. Call these variables x1 and x2

```
R> n <- 1000
R> x1mean <- 2
R> x1std <- 1
R> x1 <- matrix(rnorm(n, x1mean, x1std), n)
R> x2mean <- 3
R> x2std <- 1
R> x2 <- matrix(rnorm(n, x2mean, x2std), n)
```

2. Create a scatterplot to examine the relationship between x1 and x2.

```
R> plot(x1, x2)
```

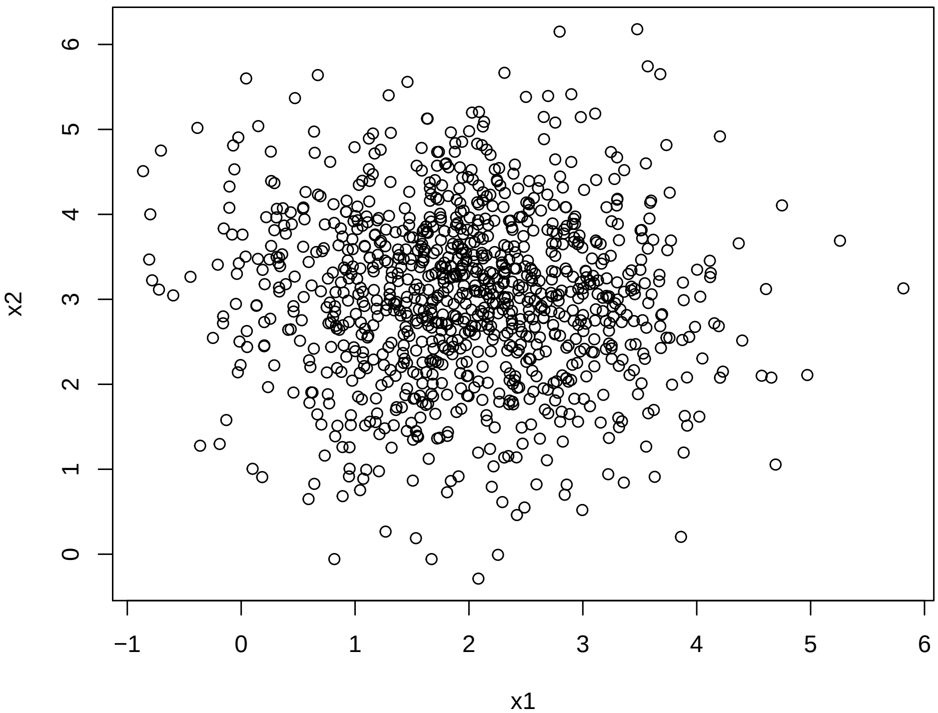


Figure 1: Scatterplot of x1 and x2

3. Create a table of sample statistics, including the correlation coefficient

```
R> df <- data.frame("var"=c("$x_1$", "$x_2$"),
+                    "mean"=c(mean(x1), mean(x2)),
+                    "std"=c(sd(x1), sd(x2)),
+                    "min"=c(min(x1), min(x2)),
+                    "max"=c(max(x1), max(x2)),
+                    "correlation"=c(cor(x1, x2), cor(x1, x2))
+                    )
```

Table 1: Sample statistics for x_1 and x_2

var	mean	std	min	max	correlation
x_1	1.9816	1.0166	-0.8613	5.8147	-0.0460
x_2	3.0349	1.0174	-0.2876	6.1790	-0.0460

4. Draw a normal(0,1) error term, define a vector of true parameters for the constant, x1, and x2 of $[1, 1, -1]$, and build your dependent variable.

```
R> eps <- rnorm(n)
R> X <- cbind(rep(1, n), x1, x2)
R> bvec <- c(1, 1, -1)
R> y <- X %*% bvec + eps
```

5. Run an OLS regression on the full model. Show the output table. Call this model "Independent, full"

```
R> bols <- solve(t(X) %*% X) %*% (t(X) %*% y)
R> e <- y - X %*% bols
R> k <- ncol(X)
R> s2 <- (t(e) %*% e) / (n-k)
R> Vb <- s2[1, 1] * solve(t(X) %*% X)
R> se <- sqrt(diag(Vb))
R> t <- bols / se
R> SSRindep <- t(e) %*% e

R> df2 <- data.frame(col1=c("constant", "$x_1$", "$x_2$"),
+                    col2=bvec,
+                    col3=bols,
+                    col4=se,
+                    col5=t
+                    )
R> colnames(df2) <- c("variable", "true value", "estimate", "s.e.", "t")
```

Table 2: OLS Estimation - Independent, Full

variable	true value	estimate	s.e.	t
constant	1.0000	0.8647	0.1197	7.2242
x_1	1.0000	1.0435	0.0312	33.4307
x_2	-1.0000	-0.9794	0.0312	-31.4050

Part B: Independence, Omitted

1. Next, drop the last column in X (your x2). Update your "k" value accordingly.

```
R> X <- X[, -k]
R> k <- ncol(X)
```

2. Re-run the regression and capture the output. Call this model "Independent, Omit".

```
R> bols <- solve(t(X) %*% X) %*% (t(X) %*% y)
R> e <- y - X %*% bols
R> s2 <- (t(e) %*% e) / (n-k)
R> Vb <- s2[1, 1] * solve(t(X) %*% X)
R> se <- sqrt(diag(Vb))
R> t <- bols / se
R> SSRindepOmit <- t(e) %*% e
```

```
R> bvec <- c(1, 1)
R> df3 <- data.frame(col1=c("constant", "$x_1$"),
+                    col2=bvec,
+                    col3=bols,
+                    col4=se,
+                    col5=t
+                    )
R> colnames(df3) <- c("variable", "true value", "estimate", "s.e.", "t")
```

Table 3: OLS Estimation - Independent, Omit

variable	true value	estimate	s.e.	t
constant	1.0000	-2.1972	0.0979	-22.4474
x_1	1.0000	1.0886	0.0440	24.7664

3. Comment on the estimated coefficient for x1 (with x2 omitted). Therefore, what can you conclude regarding the effects of an omitted variable that is independent from all included variables on the remaining coefficients?

   We can see that by omitting a relevant variable, namely x2, the estimated effects of the included variables change depending on the potential correlation that may exist between those variables and the ommited variable. Whereas the esimated value for coefficient of x1 is still very close to the true value, we can see that it has clearly impacted our estimation of the constant term. This results from the fact that x1 and x2 are independent.

Part C: Correlation, full model

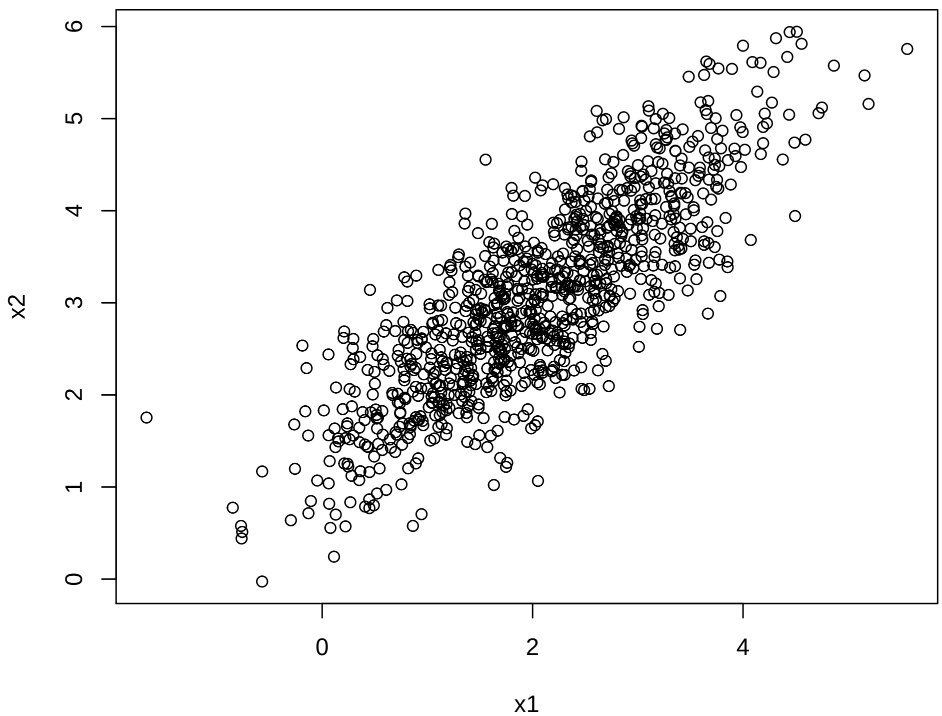
Continue with you original sweave file - **do NOT re-set the random number seed!**

1. Generate two correlated regressors (= explanatory variables), one with mean 2 and std 1, the other with mean 3 and std 1, and with covariance (correlation in this case) of 0.8. Set the sample size to 1000 as before. Use the "mvrnorm" function in the MASS package to obtain the correlated draws (some help with this is given below)

```
R> m <- c(2, 3)
R> V <- matrix(c(1, 0.8, 0.8, 1), nrow=2)
R> X <- mvrnorm(n=n, m, V)
R> x1 <- X[, 1]
R> x2 <- X[, 2]
```

2. Generate a scatter plot and a table with sample statistics, including correlation

```
R> plot(x1, x2)
```




```
R> df <- data.frame("var"=c("$x_1$", "$x_2$"),
+                   "mean"=c(mean(x1), mean(x2)),
+                   "std"=c(sd(x1), sd(x2)),
+                   "min"=c(min(x1), min(x2)),
+                   "max"=c(max(x1), max(x2)),
+                   "correlation"=c(cor(x1, x2), cor(x1, x2))
+                   )
```

Table 4: Sample statistics for x_1 and x_2

var	mean	std	min	max	correlation
x_1	2.0495	1.0598	-1.6692	5.5605	0.8221
x_2	3.0563	1.0516	-0.0266	5.9435	0.8221

3. **Use the same betas and error draws from before** and compute a new y variable. Run the full model. Call it "Correlated, full". Are there any noteworthy changes compared to the original model ("Independent, full")?

```
R> X <- cbind(rep(1, n), x1, x2)
R> bvec <- c(1, 1, -1)
R> y <- X %*% bvec + eps
R> k <- ncol(X)
```

```
R> bols <- solve(t(X) %*% X) %*% (t(X) %*% y)
R> e <- y - X %*% bols
R> s2 <- (t(e) %*% e) / (n-k)
R> Vb <- s2[1, 1] * solve(t(X) %*% X)
R> se <- sqrt(diag(Vb))
R> t <- bols / se
R> SSRcorr <- t(e) %*% e
```

```
R> df4 <- data.frame(col1=c("constant", "$x_1$", "$x_2$"),
+                    col2=bvec,
+                    col3=bols,
+                    col4=se,
+                    col5=t
+                    )
R> colnames(df4) <- c("variable", "true value", "estimate", "s.e.", "t")
```

Table 5: OLS Estimation - Correlated, Full

variable	true value	estimate	s.e.	t
constant	1.0000	0.7641	0.1004	7.6099
x_1	1.0000	0.9343	0.0524	17.8286
x_2	-1.0000	-0.8744	0.0528	-16.5576

We can observe that correlation within data makes the model less efficient. The standard error for the covariates have (negligibly?) increased. The t-value for x1 and x2 assumed almost half the prior values thereof, but remained relatively the same value for the constant term. Interestingly, for the "independent, full" model SSR value 1000.708 is obtained which is greater than the corresponding value of 995.849 for the "correlated, full" setting.

4. Omit x2, and estimate the model on the full sample. Call this model "Correlated, Omit"

```
R> X <- X[, -k]
R> k <- ncol(X)
R> bvec <- c(1, 1)
R> bols <- solve(t(X) %*% X) %*% (t(X) %*% y)
R> e <- y - X %*% bols
R> s2 <- (t(e) %*% e) / (n-k)
R> Vb <- s2[1, 1] * solve(t(X) %*% X)
R> se <- sqrt(diag(Vb))
R> t <- bols / se
R> SSRcorrOmit <- t(e) %*% e
```

```
R> df5 <- data.frame(col1=c("constant", "$x_1$"),
+                    col2=bvec,
+                    col3=bols,
+                    col4=se,
+                    col5=t
```

```
+                    )
R> colnames(df5) <- c("variable", "true value", "estimate", "s.e.", "t")
```

Table 6: OLS Estimation - Correlated, Omit

variable	true value	estimate	s.e.	t
constant	1.0000	-0.4464	0.0777	-5.7461
x_1	1.0000	0.2210	0.0337	6.5623

5. Comment on the estimated coefficient for x1 for each partial regression (with x2 omitted). Therefore, what can you conclude regarding the effects of an omitted variable that is correlated with some included variables on the remaining coefficients?

   Our results empirically shows that omitted variable can be tolerated only if they are not correlated with independent variables that are already included in the analysis. While we still calculated reliable coefficient estimates for the independent model when we omitted a variable, this is not the case for the situation with correlated variables. Omitting a variable in the correlated setting has caused our estimation for both the constant term and x1 to be far from the true values. Here our assumption of independence between the error term and the regressors is violated and our estimates are misleading.

Question 2: Omitting a variable in the wage regression

Continue with you original sweave file - **do NOT re-set the random number seed!**

Consider our wage regression from `mod1_2b`.

1. Load in the data and specify your dependent variable and your regression matrix. As before, drop "age".

```
R> data <- read.table("/Users/nima/AAEC5126/data/wage1000.txt",
+                     sep="\t", header=FALSE)
R> colnames(data) <- c("wage", "female", "nonwhite",
+                      "unionmember", "edu",
+                      "experience", "age")
R> data <- data[, -which(names(data) %in% c("age"))]
R> dftbl <- data.frame("var"=names(data), "means"=colMeans(data),
+                      "std"=apply(data, 2, sd), "min"=apply(data, 2, min),
+                      "max"=apply(data, 2, max))
```

Table 7: Sample statistics

var	means	std	min	max
wage	12.8167	8.2444	0.8400	64.0800
female	0.4910	0.5002	0.0000	1.0000
nonwhite	0.1460	0.3533	0.0000	1.0000
unionmember	0.1640	0.3705	0.0000	1.0000
edu	13.1830	2.8649	0.0000	20.0000
experience	19.2350	11.8294	0.0000	56.0000

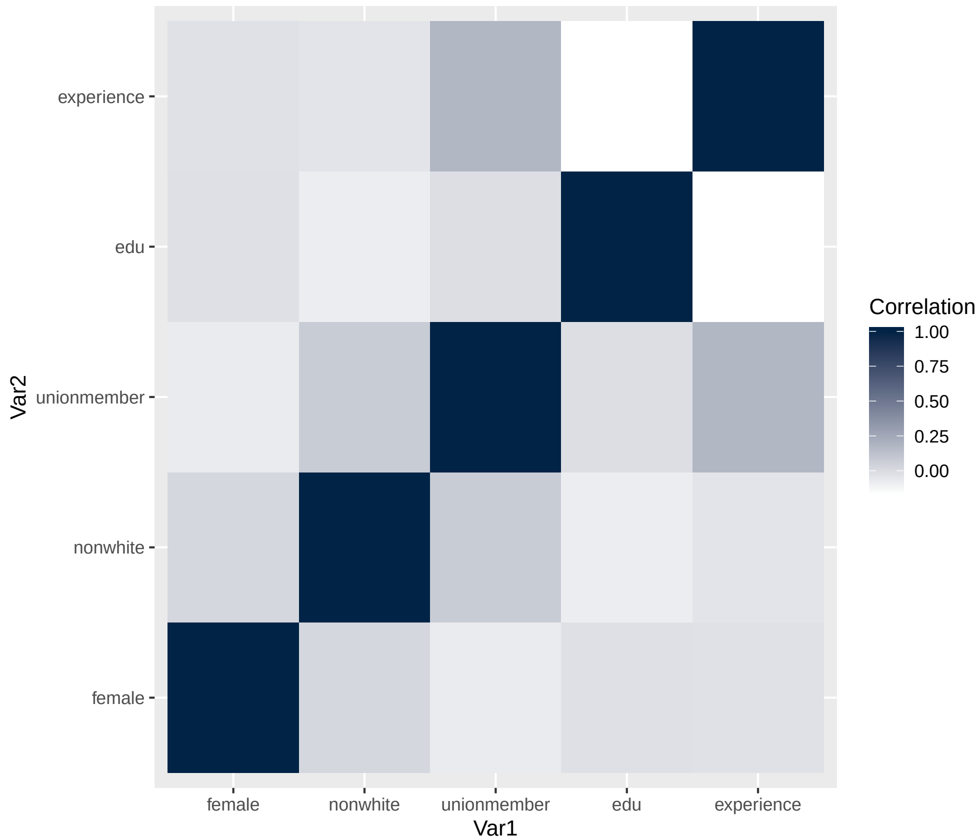
The regressand (dependent variable) is "wage", and the regressors (independent variables) are all the other covariates, namely "female", "nonwhite", "unionmember", "edu" and "experience". The regression matrix is the matrix $\boldsymbol{\beta}$ takes part in our regression $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$ where $\mathbf{y}$ and $\mathbf{X}$ are the dependent variable and independent variables, respectively.

2. Capture the sample correlation across regressors (without the constant term). Show the resulting correlation matrix in your output.

```
R> cormat <- cor(data[, -which(names(data) %in% c("wage"))])
R> dftbl2 <- data.frame(cormat)
```

Table 8: Correlation across regressors

	female	nonwhite	unionmember	edu	experience
female	1.000	0.024	-0.073	-0.022	-0.028
nonwhite	0.024	1.000	0.077	-0.082	-0.041
unionmember	-0.073	0.077	1.000	-0.009	0.178
edu	-0.022	-0.082	-0.009	1.000	-0.167
experience	-0.028	-0.041	0.178	-0.167	1.000



3. Run the full regression model and capture your output in a table.

```
R> regressors <- cbind(1, data[, -which(names(data) %in% c("wage"))])
R> colnames(regressors)[1] <- "constant"
R> X <- as.matrix(regressors)
R> k <- ncol(X)
R> n <- nrow(X)
R> y <- data[, "wage"]
R> bols <- solve(t(X) %*% X) %*% (t(X) %*% y)
R> e <- y - X %*% bols
R> SSR <- t(e) %*% e
R> s2 <- (t(e) %*% e) / (n-k)
R> Vb <- s2[1, 1] * solve(t(X) %*% X)
R> se <- sqrt(diag(Vb))
R> t <- bols / se

R> df5 <- data.frame(col1=names(regressors),
+                    col2=bols,
+                    col3=se,
+                    col4=t
+                    )
R> colnames(df5) <- c("variable", "estimate", "s.e.", "t")
```

Table 9: OLS Estimation - Wage data

variable	estimate	s.e.	t
constant	-8.5786	1.1611	-7.3884
female	-3.0985	0.4237	-7.3132
nonwhite	-1.6072	0.6032	-2.6644
unionmember	0.8212	0.5832	1.4082
edu	1.4983	0.0751	19.9483
experience	0.1697	0.0185	9.1973

We have calculated SSR = 44283.640 for this regression analysis.

4. Re-run the model without "experience" (and keep "age" out as well). How do the results change? What do your findings suggest regarding the correlation of "experience" with the remaining regressors? Is the correlation strong enough to induce noticeable omitted variable bias?

   Eliminating "experience" has not impacted most of the covariates considerably, with the exception of "unionmember" which has assumed a biased estimated value of higher magnitude. This suggest the existence of correlation between "unionmember" and the dropped variable "experience", which can align with an interpretation of the relation between the two variables that one may imagine. The correlation however is not dominating to an extent that causes unacceptable omitted variable bias.

```
R> regressors <- cbind(1, data[, -which(names(data) %in% c("wage", "experience"))])
R> colnames(regressors)[1] <- "constant"
R> X <- as.matrix(regressors)
R> k <- ncol(X)
R> n <- nrow(X)
R> y <- data[, "wage"]
R> bols <- solve(t(X) %*% X) %*% (t(X) %*% y)
R> e <- y - X %*% bols
R> SSR <- t(e) %*% e
R> s2 <- (t(e) %*% e) / (n-k)
R> Vb <- s2[1, 1] * solve(t(X) %*% X)
R> se <- sqrt(diag(Vb))
R> t <- bols / se

R> df6 <- data.frame(col1=names(regressors),
+                   col2=bols,
+                   col3=se,
+                   col4=t
+                   )
R> colnames(df6) <- c("variable", "estimate", "s.e.", "t")
```

   We have calculated SSR = 48052.202 for this regression analysis.

Table 10: OLS Estimation - Wage data

variable	estimate	s.e.	t
constant	-3.8082	1.0815	-3.5211
female	-3.1658	0.4411	-7.1776
nonwhite	-1.9937	0.6265	-3.1821
unionmember	1.8002	0.5970	3.0156
edu	1.3787	0.0770	17.9003

Question 3: Orthogonality and Projection

Consider the "residual maker matrix" $\mathbf{M}$ and the projection matrix $\mathbf{P}$. Show formally that the following hold (please type all Math in LaTeX):

1. $\mathbf{MX} = \mathbf{0}$ (Provide intuition).

$$\begin{aligned}\mathbf{MX} &= (I - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{X} \\ &= \mathbf{X} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} = \mathbf{X} - \mathbf{X} = 0\end{aligned}$$

   The orthogonality (and hence lack of correlation) between the residual maker $\mathbf{M}$ and regressors $\mathbf{X}$ results in transformation of $\mathbf{y}$ into "everything $\mathbf{X}$ could not explain"! One way of interpreting this result is that if $\mathbf{X}$ is regressed on $\mathbf{X}$, a perfect fit will result and the residuals will be zero.

2. $\mathbf{PX} = \mathbf{X}$ (Provide intuition).

$$\begin{aligned}\mathbf{PX} &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} \\ &= \mathbf{XI} = \mathbf{X}\end{aligned}$$

   As opposed to the case above, the projection matrix $\mathbf{P}$ transforms $\mathbf{y}$ into "everything that $\mathbf{X}$ is able to explain", that is the fitted values. In other words, $\mathbf{X}$ is invariant under $\mathbf{P}$.

3. $\mathbf{y} = \mathbf{Py} + \mathbf{M} * \mathbf{y}$ (Provide intuition)

$$\begin{aligned}\mathbf{Py} + \mathbf{M} * \mathbf{y} &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} + (\mathbf{y} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{y} \\ &= \mathbf{Iy} = \mathbf{y}\end{aligned}$$

   Obviously, $\mathbf{y}$ can be partitioned into two parts, one that can be explained via $\mathbf{X}$ and one that can not be explained via the regressors $\mathbf{X}$. Adding these two parts can "reconstruct" the original $\mathbf{y}$. In other words, summing up $\mathbf{X}$ transformed via these two complementary projections gives us the whole information that was to be captured from $\mathbf{y}$, reversing the decomposition.

4. $\mathbf{PM} = \mathbf{0}$

$$\mathbf{PM} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}X'(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' = 0$$

5. $\mathbf{e}'\mathbf{e} = \mathbf{e}'\mathbf{y}$

$$\begin{aligned}\mathbf{e}'\mathbf{e} &= (\mathbf{My})'(\mathbf{My}) = \mathbf{y}'\mathbf{My} = (\mathbf{Py} + \mathbf{My})'\mathbf{My} = \mathbf{y}'\mathbf{PMy} + \mathbf{y}'\mathbf{MMy} \\ &= (\mathbf{My})'\mathbf{y} = \mathbf{e}'\mathbf{y}\end{aligned}$$

6. $\mathbf{y}'\mathbf{y} = \hat{\mathbf{y}}'\hat{\mathbf{y}} + \mathbf{e}'\mathbf{e}$

$$\begin{aligned}
\mathbf{y}'\mathbf{y} &= (\mathbf{Py} + \mathbf{My})'(\mathbf{Py} + \mathbf{My}) \\
&= (\mathbf{Py})'(\mathbf{Py}) + (\mathbf{Py})'(\mathbf{My}) + (\mathbf{My})'(\mathbf{Py}) + (\mathbf{My})'(\mathbf{My}) \\
&= \hat{\mathbf{y}}'\hat{\mathbf{y}} + \hat{\mathbf{y}}'\mathbf{PMy} + \hat{\mathbf{y}}'\mathbf{MPy} + \mathbf{e}'\mathbf{e} \\
&= \hat{\mathbf{y}}'\hat{\mathbf{y}} + \mathbf{e}'\mathbf{e}
\end{aligned}$$

```
R> proc.time()-tic
```

```
   user  system elapsed 
  0.721   0.089   0.814 
```