Julie George
Professor Garcia-Rios
Assignment #3
May 2, 2018

Problem Set #3

A. Run mcls.r using its default settings. Make a note of the results. Rerun the program three times, setting the correlation of x_1 and x_2 to 0.5, 0.9, and 0.99, respectively.[3] Based on the results from these runs, what can you say about the effect of partial collinearity on least squares estimates? In particular, does raising the correlation of x_1 and x_2 add bias to our estimates of β_1, β_2, or β_3? Does raising the correlation of x_1 and x_2 affect the precision of estimates of β_1, β_2, or β_3?

MULTICOLLINEARITY

#These are the results of running mcls.r using its default settings below.

#I did not change the default code.

```
SigmaX <- c(1,  0,  0,
            0,  1,  0,
            0,  0,  1)
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
   1.003344    1.997949    2.997716    3.999585
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1401899   0.1480866   0.1409893   0.1399717
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1439255   0.1451582   0.1451158   0.1459178
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   7.133179   13.505615   21.278210   28.577204
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   6.971276   13.763948   20.657405   27.409846
```

#Setting the correlation of X1 and X2 to 0.5 (code and results below)

```
SigmaX <- c(1,  0.5,  0,
            0.5,   1,   0,
             0,   0,  1)
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.9921735   1.9963098   2.9997389   4.0012818
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1426864   0.1663376   0.1674496   0.1518163
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1442060   0.1685104   0.1678252   0.1459555
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   7.008375   12.023736   17.915838   26.347630
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   6.880253   11.846802   17.874188   27.414404
```

#Setting the correlation of X1 and X2 to 0.9 correlation

```
SigmaX <- c(1,  0.9,  0,
            0.9,   1,   0,
             0,   0,  1)
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
   1.006229    2.004127    2.997397    4.000993
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1487467   0.3311546   0.3333431   0.1499543
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1437390   0.3331409   0.3336535   0.1452494
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   6.722839    6.039475    8.999736   26.674801
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   7.000388    6.015855    8.983561   27.545678
```

#Setting the correlation of X1 and X2 to 0.99 correlation

```
SigmaX <- c(1,  0.99,  0,
            0.99,   1,   0,
            0,   0,  1)
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
   1.000749    2.012050    2.981383    3.999856
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1447374   1.0393176   1.0441279   0.1429929
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1437115   1.0303421   1.0297164   0.1457648
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   6.909064    1.924340    2.873211   27.973422
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   6.963598    1.952798    2.895344   27.440491
```

The effect of partial collinearity on least squares estimates is not a major concern. Raising the correlation of x1 and x2 does not bias our estimates of the regression coefficients, but it does decrease the precision of estimates of B1, B2, and B3 each time that we increase the correlation. The estimates of the coefficients are not biased as evidenced by the similar coefficients over each run to the true values (1, 2, 3, 4). Yet, this decrease of precision is most evident with the increasing standard errors for each time that I run the program with a higher correlation of x1 and x2, which gets farther away from the true values of the standard errors values estimates.

B. Set the correlation of x_1 and x_2 to 1, and rerun mcls.r. What has happened, and why? It will help to look at the summary of the regression results for the last run, using print(summary(res)).

PERFECT COLLINEARITY

```
SigmaX <- c(1,  1,  0,
            1,  1,  0,
            0,  0,  1)
```

#It will be help to look at the summary of the regression results for the last run

print(summary(res))

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
   1.002850    5.001229          NA    4.002091
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1415305   0.1413352          NA   0.1463254
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X3
  0.1425427   0.1443686   0.1440133
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   7.065613   14.150760          NA   27.336343
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   7.035438   34.642086          NA   28.076444
```

There is perfect collinearity when we set the correlation of x1 and x2 to 1. X2 becomes identified as "NA". This biases the coefficient of x1 (the model now adds the X2 value to X1, so that X1 is the addition of the values of ~2 and ~3 to get the overall value of ~5.00). However, the standard error value estimates of x1 and x3 as well as the intercept are near the true values of the standard error estimates.

C. Now open the program mcovb.r in your text editor. Note that this program is identical to mcls.r, with one exception. When this program runs lm(), it omits x_2 from the regression. Now run the program at its default settings, with the correlation of x_1 and x_2 set to 0. What effect does the omission of x_2 have on the bias and precision of the estimates of β_1 and β_3?

OMITTED VARIABLE BIAS

#I did not change the code for this problem. I kept the correlation of x1 and x2 as 0, as seen below.

```
SigmaX <- c(1,  0,  0,
            0,  1,  0,
            0,  0,  1)
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
     1.003468      1.996495      4.019567
[1] ""
[1] "True standard errors across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
    0.3225835     0.3417838     0.3446404
[1] "Average estimated standard errors across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
    0.3346237     0.3372902     0.3374947
[1] ""
[1] "True t-stat across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
     3.099972      5.851653     11.606300
[1] "Average estimated t-stat across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
     2.998797      5.919221     11.910016
```

The omission of x2 does not affects the bias the estimates of B1 and B3. We see that the regression coefficients have not changed much with the omission of x2 as the B1 coefficient is close to the true value of 2 and B3's coefficient is close to the true value of 4. However, the standard error estimates have become less precise (based on increased standard error estimates) from the true values of standard error.

D. Set the correlation of x_1 and x_2 to 0.9, and rerun mcovb.r. Now what effect does the omission of x_2 have on the bias and precision of the estimates of β_1 and β_3? Do our findings differ from those in part c? Why?

OMITTED VARIABLE BIAS

```
SigmaX <- c(1,  0.9,  0,
            0.9,  1,  0,
             0,   0,  1)
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
     1.004896      4.693686      3.994481
[1] ""
[1] "True standard errors across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
    0.1949991     0.1979362     0.1940485
[1] "Average estimated standard errors across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
    0.1950599     0.1970406     0.1974077
[1] ""
[1] "True t-stat across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
      5.12823      10.10426      20.61340
[1] "Average estimated t-stat across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
     5.151728     23.820903     20.234673
```

I set the correlation of x1 and x2 to 0.9. The omission of x2 does affect the bias and precision of the estimates of B1. The coefficient of X1 has increased (and is farther from the true value of 2), which is a biased estimate, and the standard error value estimates has increased, which results in less precision. The true value of x3 is 4, which is very close to what I got in my coefficient estimate, but the precision has increased as well based on the standard error (which is far from the true value of the standard error). Yes, these findings are different from part c due to the correlation of 0.9, but this model is more precise than part C (evidenced by the standard error estimates).

E. Finally, keep the correlation of x_1 and x_2 at 0.9, but rewrite mcovb.r to run the regression of y on x_1 and x_2, omitting x_3. What effect does the omission of x_2 have on the bias and precision of the estimates of β_1 and β_2?

OMITTED VARIABLE BIAS

#This is the line of code that I have kept the same from the previous question, as the correlation of X1 and X2 is 0.9

```
SigmaX <- c(1,  0.9,  0,
            0.9,   1,  0,
              0,   0,  1)
```

#This is the line of code in which I omit X3.

```
res <- lm(y~X[,c(1,2)])
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
  (Intercept) X[, c(1, 2)]1 X[, c(1, 2)]2
    0.9823384     2.0058306     2.9754004
[1] ""
[1] "True standard errors across 1000 simulation runs"
  (Intercept) X[, c(1, 2)]1 X[, c(1, 2)]2
    0.4462209     0.9955222     0.9974739
[1] "Average estimated standard errors across 1000 simulation runs"
  (Intercept) X[, c(1, 2)]1 X[, c(1, 2)]2
    0.4283557     0.9865692     0.9899784
[1] ""
[1] "True t-stat across 1000 simulation runs"
  (Intercept) X[, c(1, 2)]1 X[, c(1, 2)]2
     2.241043      2.008996      4.010130
[1] "Average estimated t-stat across 1000 simulation runs"
  (Intercept) X[, c(1, 2)]1 X[, c(1, 2)]2
     2.293277      2.033137      3.005520
```

I have kept the correlation of x1 and x2 at 0.9 (correlation), and rewrote mcovb.r to run the regression of y on x1 and x2, omitting x3. The estimate of X1 is close to the true value of 2 and estimate X2 is close to the true value of 3. However, the standard error values have increased from the true values of the standard error values, which is less precision in our model.

F. What explains the differences in your results across parts c, d, and e? Based on these results, and your findings in part a, how would you recommend users of least squares deal with highly correlated covariates?

I would recommend users of least squares deal with partially correlated covariates. The differences in the results deal with omitted variable bias (the removal of x2 or x3), which heavily bias estimates in some cases or lead to less precise estimates. This was caused by the correlation of 0.9 versus 0. However, if we have perfectly correlated variables in the model, we should remove one of the variables. If the variables are highly correlated, then I would recommend that he or she keeps them in the model as the coefficients would be unbiased – however, the cost of doing this would be less precision in the model. Multicollinearity is often a problem that many scholars face. This is the tradeoff as our model would not be biased, but we would have less precision. Last, a potential solution for some scholars for severe multicollinearity (the VIF for a factor is near or above 5) would be partial least squares regression.

G. Open the program mcselect.r in your text editor. Note that this program is identical to mcls.r, except now, all observations in which y is greater than its sample mean are deleted prior to running the regression. What effect does selection on y have on the bias and precision of the estimates of β_1, β_2, and β_3?

SELECTION ON THE DV

#I have kept the default code and made no changes. Below are the results.

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)    selectX1    selectX2    selectX3
  0.3061197   1.7888759   2.6902833   3.5975142
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)    selectX1    selectX2    selectX3
  0.3180597   0.2219794   0.2401474   0.2893459
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)    selectX1    selectX2    selectX3
  0.3122049   0.2185754   0.2415821   0.2696263
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)    selectX1    selectX2    selectX3
   3.144064    9.009844   12.492327   13.824284
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)    selectX1    selectX2    selectX3
  0.9805089   8.1842522  11.1361034  13.3425921
```

The selection on y does bias the estimates of the coefficients, as evidenced by the lower coefficients compared to the original regressions' coefficients of mcls.r (lower than the true values of 1, 2, 3, and 4). There is also less precision with this regression's standard error values compared to the original regression's true standard error values evidenced by the higher standard errors compared to the true standard error values.

H. Open the program mchet.r in your text editor. Note that this program is identical to mcls.r, except the structure of sigma has changed. Run mchet.r under its default setting, which sets $\gamma_0 = \log(2)$ and $\gamma_1 = 0$. Confirm that under these settings, y is still homoskedastic. Note the result. Now try adding heteroskedasticity by increasing γ_1 to 1. Confirm that changing this setting has made y heteroskedastic. What effect does this added heteroskedasticity have on our results?

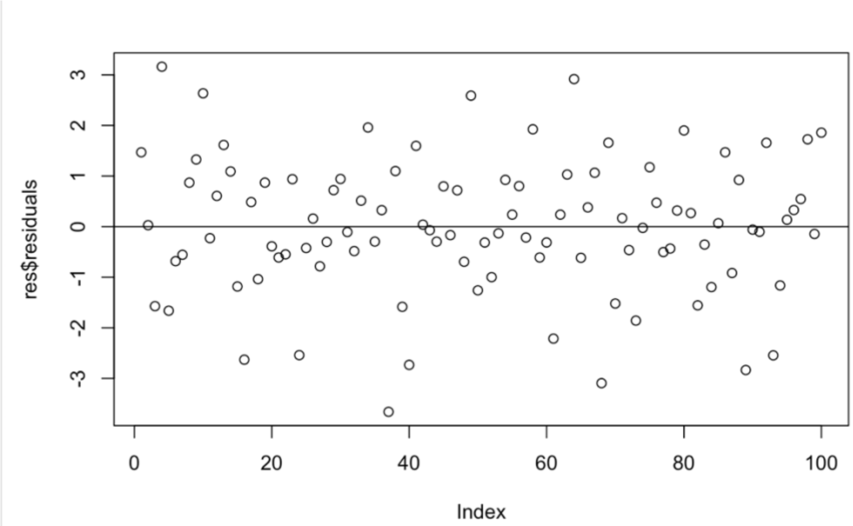
HETEROSKEDASTICITY

#Running the default code
g <- c(log(2),0)

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   1.005465    1.998396    3.002464    3.999388 
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3 
  0.1398657   0.1448236   0.1427664   0.1425964 
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3 
  0.1437665   0.1451698   0.1457582   0.1454846 
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   7.149717   13.809906   21.013345   28.051207 
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   6.993732   13.765920   20.598941   27.490115 
```

#Checking to see if it is homoscedastic (Yes, it is!)

plot(res$residuals)
abline(0,0)



Based on the results (when gamma is 0) and graph, this is homoscedastic. Homoskedasticity is also known as "same variance." It assumes that different samples have the similar variance, even if they come from different populations. This is evidenced by the very similar true standard error values and standard error estimates of X1, X2, and X3.

#Changing the code for gamma 1

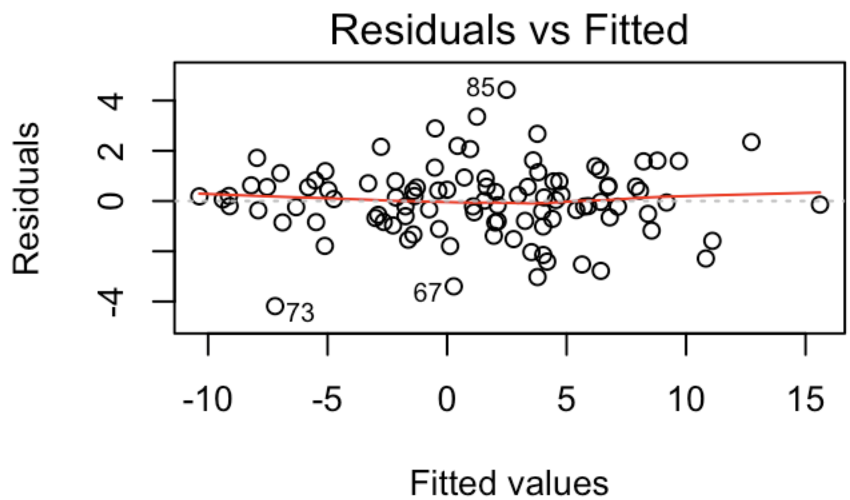
g <- c(log(2),1)

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   1.008003    2.017455    3.004190    3.998691 
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3 
  0.1813502   0.2573638   0.1815191   0.1864519 
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3 
  0.1841932   0.1849420   0.1863220   0.1862768 
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   5.514192    7.771100   16.527184   21.453255 
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   5.472529   10.908586   16.123646   21.466395 
```

#Checking residuals for heteroscedasticity (it is! Let's look at the standard error estimates and true values of the standard errors – they are different, especially X1!)

```
plot(res$residuals)
abline(0,0)

par(mfrow=c(2,2))
plot(res)
```



Heteroskedasticity is present when the size of the error term differs across values of an independent variable. When I add heteroskedasticity by increasing gamma1 to 1, the standard errors are farther away from the true values of standard errors. There is heteroskedasticity, which violates the homoscedasticity assumption! Of note, the coefficient estimates are still similar to the true values of the coefficient estimates.

The effect of heteroskedasticity here is that (1) The OLS estimators and regression predictions based on them remains unbiased and consistent. (i.e. true value and estimates are about the same) (2) The OLS estimators are no longer the BLUE (Best Linear Unbiased Estimators) because they are no longer efficient, so the regression predictions will be inefficient too. (i.e. in this case, larger variance) and (3) Because of the inconsistency of the covariance matrix of the estimated regression coefficients, the tests of hypotheses, (i.e. t-test) are no longer valid. (i.e. in this case, smaller t-score)

I. Open the program mcautocor.r in your text editor. Note that this program is identical to mcls.r, except for two differences. Run mcautocor.r under its default settings, with $\rho = 0$ and $\rho_{xk} = 0$ for all covariates k. Note the results. Rerun it twice: first set $\rho = 0.5$ and $\rho_{xk} = 0.5$ for all k; then set $\rho = 0.9$ and $\rho_{xk} = 0.9$ for all k. Based on the results from these runs, what can you say about the effect of serial correlation on least squares estimates? Experimenting further, what happens if you have serial correlation in y but not in X, or vice versa?

#Default code

rho <- 0
SigmaX <- c(1, 0, 0,
 0, 1, 0,
 0, 0, 1)

rhoX <- c(0, 0, 0)

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.9992603   1.9975657   2.9978329   3.9996159
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1441377   0.1518971   0.1409260   0.1465165
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1436390   0.1446550   0.1457161   0.1447732
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   6.937812   13.166810   21.287762   27.300676
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   6.956748   13.809176   20.573104   27.626773
```

The coefficient estimates are very close to the true values of 1, 2, 3 and 4. The standard errors estimates are also close to the true values of the standard error estimates.

#p is 0.5

rho <-0.5

rhoX <- c(0.5, 0.5, 0.5)

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
   1.010693    1.992458    2.997045    3.995078
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.2187694   0.1717553   0.1642446   0.1677842
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1606165   0.1447834   0.1446659   0.1448961
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   4.571022   11.644473   18.265435   23.840141
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   6.292589   13.761642   20.717012   27.572015
```

#The coefficient estimates are still close to the true values of 1, 2, 3, and 4 with p = 0.5. The standard error values have changed and are not similar to the true values of the standard errors!

#p is 0.9

rho <- 0.9

rhoX <- c(0.9, 0.9, 0.9)

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   1.006373    2.001214    2.998458    3.998257 
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3 
  0.2706776   0.1774575   0.1805526   0.1733315 
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3 
  0.1926477   0.1454030   0.1449074   0.1446126 
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   3.694432   11.270304   16.615659   23.077173 
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   5.223903   13.763221   20.692228   27.648046 
```

This is an issue of serial correlation. The coefficient estimates are still close to the true values 1, 2, 3, and 4, so there is no bias there. However, the standard errors have changed quite a bit each time I change the p value. Ultimately, while coefficient estimates are not biased, the standard errors have deviated from the true standard error values.

#This is to check on the serial correlation in x first, then I will do y.

# True effect of last period's error term on current period
rho <- 0

Serial correlation in X's

rhoX <- c(.9, .9, .9)

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
   1.004091    2.000591    2.994179    4.000893
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1491455   0.1142423   0.1087276   0.1124826
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1461012   0.1099875   0.1097526   0.1104749
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   6.704862   17.506650   27.591902   35.561052
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   6.872573   18.189266   27.281177   36.215391
```

VERSUS (serial correlation in Y)

True effect of last period's error term on current period

rho <- 0.9

Serial correlation in X's

rhoX <- c(0, 0, 0)

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
   1.008011    2.003297    3.001944    3.997969
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.2589231   0.1968284   0.1866372   0.1971361
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.1915724   0.1933369   0.1936765   0.1932871
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
    3.86215    10.16113    16.07397    20.29055
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   5.261775   10.361692   15.499787   20.684092
```

There is no bias of correlation estimates for serial correlation. But there are discrepancies with standard error value estimates, which have deviated from the true values of standard error estimates. When there is serial correlation in x, but not in y, the standard error estimates are smaller than the true standard error estimates. When there is correlation in Y but not in X, the standard error estimates are larger than the true standard error estimates.

J. Come up with a question about the properties of least squares to investigate using one or more of the provided programs, or modifications thereof. Illustrate the answer to your question by running the program(s) under different settings, and comparing results.

   An example question:

   Which of the problems identified in this homework can be mitigated by gathering more data (e.g., by setting n=1000, instead of n=100), and which problems will stay just as severe no matter how much data are collected?

#1: This problem deals with multicollinearity with the dataset mcls.r
#I will set the correlation of x1 and x2 to 0.9 and increase the N size to 1000

n <- 1000

```
SigmaX <- c(1,  0.9,  0,
            0.9,  1,  0,
              0,  0,  1)
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   0.999921    1.996198    3.003183    3.998989 
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3 
 0.04674177  0.09871021  0.10205302  0.04569401 
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3 
 0.04477028  0.10281553  0.10283470  0.04484357 
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   21.39414    20.26133    29.39648    87.53882 
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3 
   22.33448    19.41533    29.20398    89.17643 
```

When compared to the original results earlier (question a) (i.e. when n = 100, and the correlation of x1 and x2 is 0.9), we see that the standard error estimates have decreased with this larger N size, indicating improved precision. As a result, more data can mitigate the multicollinearity problem. Last, the regression coefficients are similar to the true values of the coefficients.

#2: This deals with omitted variable bias. I use the dataset mcovb.r where coefficient X2 is removed from the regression. I set the correlation of X1 and X2 to 0.9, and increase the N size to 1000.

```
n <- 1000

SigmaX <- c(1,  0.9,  0,
            0.9,   1,  0,
             0,   0,  1)
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
    0.9987112     4.6999999     4.0014872
[1] ""
[1] "True standard errors across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
   0.06003307    0.06024886    0.06288642
[1] "Average estimated standard errors across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
   0.06098482    0.06110335    0.06112435
[1] ""
[1] "True t-stat across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
     16.65749      33.19565      63.60674
[1] "Average estimated t-stat across 1000 simulation runs"
  (Intercept) X[, c(1, 3)]1 X[, c(1, 3)]2
     16.37639      76.91885      65.46470
```

When comparing the above results with the previous question related to omitted variable bias, question D, (i.e. when n = 100, x2 is omitted in regression, and the correlation of x1 and x2 is 0.9), we can see that there still remains bias of coefficient estimate X1. However, the larger N size makes the estimates more precise as evidenced by the decreased standard error estimates. As a result, more data cannot mitigate the omitted variable bias problem.

#3: This deals with selecting on the dependent variable. I use the dataset mcselect.r, where all observations in which y is greater than its sample mean are removed to regression.

```
n <- 1000
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)     selectX1     selectX2     selectX3
  0.2884792    1.7944873    2.6921739    3.5884506
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)     selectX1     selectX2     selectX3
 0.09722013   0.06871125   0.07424592   0.08555713
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)     selectX1     selectX2     selectX3
 0.09464067   0.06605228   0.07261806   0.08113904
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)     selectX1     selectX2     selectX3
   10.28594     29.10731     40.40626     46.75239
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)     selectX1     selectX2     selectX3
   3.048153    27.167682    37.073063    44.225943
```

When comparing the above result with the previous one in question g. (i.e. when n = 100, and selecting on the y is evident), I see that there is bias in the coefficient estimates. However, the larger N made the standard error estimates more precise as evident by the decreased standard error values. As a result, more data cannot mitigate the problem of selection bias on the y.

#4: This deals with heteroskedasticity. I used the dataset mchet.r, where gamma is set to 1, making the model heteroskedastic.

```
n <- 1000
g <- c(log(2),1)
```

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.9996974   1.9971016   3.0003889   3.9968980
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
 0.05877131  0.08028813  0.05612761  0.05959140
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
 0.05755227  0.05756069  0.05763852  0.05756720
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   17.01510    24.91028    53.44963    67.12378
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   17.37025    34.69558    52.05527    69.43013
```

When comparing the above results with the previous question H (i.e. when n = 100, and heteroskedasticity exists because gamma is equal to one), I see that for all of the variables, the "true" and estimated value of coefficient estimates are similar. However, the standard error estimates and true standard error estimates are different, especially with X1. There is still heteroskedasticity, even with a larger N. Therefore, more data cannot mitigate the heteroskedasticity problem.

#5: This deals with serial correlation. I use the dataset mcautocor.r.

n <- 1000
#Set Rho = 0.9 and Rho*Xk = 0.9 for all k
rho <- 0.9
rhoX <- c(0.9, 0.9, 0.9)

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)          X1          X2          X3
  0.9970006   1.9986872   3.0009990   3.9987558
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
 0.08385875  0.05532639  0.05484743  0.05580159
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)          X1          X2          X3
 0.06039552  0.04489394  0.04497351  0.04500325
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   11.92481    36.14912    54.69719    71.68254
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)          X1          X2          X3
   16.50786    44.52020    66.72814    88.85483
```

When comparing the above results with the previous question I (i.e. when n = 100, and autocorrelation exists in both y and x), I see that there is discrepancy between true and estimated standard error values is present – standard error estimates decrease. Overall, more data cannot mitigate autocorrelation problem.

K: Bonus Question:

Question: Using the dataset mcselect.r, how does changing the conditional mean (by setting MuX = 0 across the covariates to MuX = 0.5) affect the model in terms of bias?

Answer: One of the assumptions of the Gauss Markov Theorem is that the conditional mean should be zero. First, in the default code and repeated samples of size 100, the mean outcome of the estimate equals 0.

The mean of the error terms has an expected value of zero given values for the independent variables. In mathematical notation, this assumption is correctly written as E(U | X) = 0. Here, E is the expectation operator, U the matrix of error terms, and X the matrix of independent variables.

This assumption states the distribution each error term, u_i, is drawn from has a mean of zero and is independent of the x's.

By changing the mean of the model to 0.5, I have violated the assumption and have caused the regression coefficients to be biased. Thus, if we change the conditional mean to something other than 0 such as 0.5, then we are getting BIASED coefficient estimates. The regression coefficients are much lower than the true coefficient value parameters.

Below are my results to verify my conclusion.

True means of the covariates

muX <- c(0.5,0.5,0.5)

```
[1] "True parameters"
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[1] "Average LS estimate across 1000 simulation runs"
(Intercept)     selectX1     selectX2     selectX3
  0.7521497    1.7909756    2.6893509    3.5911235
[1] ""
[1] "True standard errors across 1000 simulation runs"
(Intercept)     selectX1     selectX2     selectX3
  0.2226564    0.2242533    0.2359738    0.2802586
[1] "Average estimated standard errors across 1000 simulation runs"
(Intercept)     selectX1     selectX2     selectX3
  0.2057561    0.2207055    0.2423226    0.2713888
[1] ""
[1] "True t-stat across 1000 simulation runs"
(Intercept)     selectX1     selectX2     selectX3
   4.491225     8.918488    12.713274    14.272532
[1] "Average estimated t-stat across 1000 simulation runs"
(Intercept)     selectX1     selectX2     selectX3
   3.655541     8.114775    11.098225    13.232393
```