Impact of Multicollinearity (DRAFT)

Katherine Daignault

In this module, we will visualize the impact on our ability to make inference with the estimated regression relationship when we have two or more predictors that are correlated to one another. To do this, we will simulate a dataset in which we know the true relationship between our response (Y) and our predictors (X_1, X_2, X_3), and we are able to decide how correlated we wish our predictors to be. When two or more predictors are correlated to each other, we say we have **multicollinearity** present in our model.

MCQ: Which statement is true regarding correlation between two measurements? - The correlation coefficient tells us whether these measurements are linearly related. - If the measurements are strongly linearly related, then a linear regression model can be used to represent this association. *** - The correlation coefficient can be used to describe the association between 3 or more measurements. - None of these options are true. - More than one of these options are true.

A Model Without Correlated Predictors

Creating the Data

We will start by simulating a dataset where we get to select the true values of the coefficients (β). We will create our dataset so that our predictors will be uncorrelated to each other (the ideal scenario). We will be trying to estimate a model using 3 predictors, where the true relationship is

$$Y = 1 + 2X_1 + 3X_2 + 0X_3 + \epsilon, \quad \epsilon \sim N(0, 2.25).$$

MCQ: Which of the below statements is correct regarding this true relationship? - X_3 has no linear association to Y. - In the presence of the other predictors, a one-unit increase in X_2 yields a 3-unit increase in Y. - On average, when other predictors are held fixed, a one-unit increase in X_1 increases Y by 2 units. *** - All of the statements are true. - None of the statements are true.

To begin, we will set all our fixed values for how we would like to generate our data. This includes the error variance in the population ($\sigma^2 = 2.25$), the true regression coefficients ($(\beta_0, \beta_1, \beta_2, \beta_3) = (1, 2, 3, 0)$), the sample size ($n = 50$), and the covariance matrix for the 3 predictors we will create. The covariance matrix takes the form

$$\Sigma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where the values on the main diagonal are the variances of each predictor in the population, and the off-diagonal elements are the covariances between any two predictors. Once these values are set up, we can "sample" our predictors from the population in which each predictor distribution has a mean of 0. When the predictors have been created, check the correlation between them to ensure we have nearly uncorrelated predictors.

```
# fill in the values from above:
n = 50
errorvar = 2.25
```

```
betas <- c(1,2,3,0)
covariance <- diag(c(1,1,1))

# generate/sample the predictors using a multivariate Normal distribution
X <- mvrnorm(n, mu = c(0,0,0), Sigma = covariance)

# check the correlation between the predictors
cor(X)
```

```
##             [,1]        [,2]        [,3]
## [1,] 1.00000000  0.13241305  0.06648685
## [2,] 0.13241305  1.00000000 -0.04373183
## [3,] 0.06648685 -0.04373183  1.00000000
```

MCQ: Why might it be possible to observe a sample correlation that is non-zero when we provided R with a covariance matrix with zero correlation? - The population has uncorrelated predictors, but the sample may not. *** - We did not set up our code correctly for the population. - The sampling process did not function correctly. - None of these options - More than one of these options

Lastly, we can use this information to generate our errors and our response variable to be able to fit linear regression models in the next section. To do this, we should use the true relationship that was given at the top of this section.

```
# generate the random errors for these data from a Normal(0, 2.25)
e <- rnorm(n, 0, sqrt(2.25))

# update your X vector to include an intercept column
Xmat <- cbind(1, X)

# use the above two components to generate a response variable
y <- Xmat %*% betas + e

# produce a summary of your response
summary(y)
```

```
##        V1
##  Min.   :-6.0651
##  1st Qu.:-0.5456
##  Median : 1.2908
##  Mean   : 1.5476
##  3rd Qu.: 4.1301
##  Max.   :11.0336
```

Fit Simple Linear Models with Each Predictor

Now that we have decided on our true population relationship between Y and predictors X_1, X_2, and X_3 and "collected" a sample of data from the population, we can begin to use this data to estimate the true relationship. To understand how each predictor individually relates to the response, we can fit three separate simple linear models, using each predictor one at a time.

Each time we fit a simple linear model, we will store that model for use later.

```
# Fit a simple linear model using X1 as the predictor
model1a <- lm(y ~ X[,1])
```

```r
# Fit a simple linaer model using X2 as the predictor
model2a <- lm(y ~ X[,2])

# Fit a simple linear model using X3 as the predictor
model3a <- lm(y ~ X[,3])

# view the summaries of each model
#summary(model1)
#summary(model2)
#summary(model3)
```

MCQ: Which statement is true regarding the three simple linear models? - The estimated slope in each model is not equal to the corresponding slope in the true relationship. *** - In only one of the true relationship and the simple linear model using X_3, X_3 has no linear association with Y. - The estimate of the error variance matches the true error variance in the population. - None of the options are true.

Visualizing the Significance of the Estimates

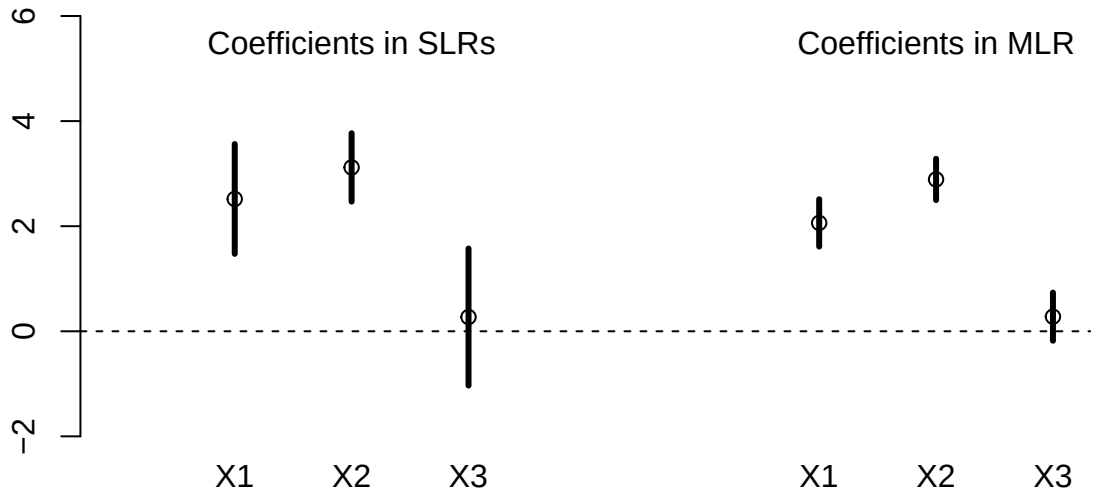
To visualize the estimated coefficient for each of X_1, X_2 and X_3 from the 3 separate simple linear models, we can make a plot that displays the estimated coefficients as points and places a confidence interval around each in the vertical direction. We can then compare the results from the separate SLRs to an estimated relationship from a multiple linear model. We will extract the 95% confidence interval of each slope using the confint() function. We will also extract the estimated coefficients using the coef() function.

```r
# Fit a multiple linear model which includes all 3 predictors
MLRa <- lm(y ~ X)

# Extract the slope coefficients from all 4 models
SLRcoefs <- c(coef(model1a)[2], coef(model2a)[2], coef(model3a)[2])
MLRcoefs <- coef(MLRa)[2:4]

# relabel to values more intuitively
names(SLRcoefs) <- c("X1", "X2", "X3")
names(MLRcoefs) <- c("X1", "X2", "X3")

# Extract confidence intervals (95%) from each simple linear model
x1confint <- confint(model1a)[2,]
x2confint <- confint(model2a)[2,]
x3confint <- confint(model3a)[2,]

# extract the confidence intervals of the 3 slopes from the MLR
MLRconfints <- confint(MLRa)[2:4,]

# create a combined coefficients vector and combined CI matrix
coefs <- c(SLRcoefs, MLRcoefs)
confints <- rbind(x1confint, x2confint, x3confint, MLRconfints)
```

Now we will plot the information we have extracted from the models. We will add the estimates as points and the bounds of confidence intervals as endpoints of a line. A horizontal line will also be added to represent the predictor having no effect on the response. By merging the coefficients and the confidence intervals from all models into a single storage element, it allows us to draw this plot more efficiently by using a for loop.

To draw this plot, you will need to add - the correct x-axis and y-axis elements to the plot function - the horizontal line at 0 as a dashed line - the vertical lines representing the confidence intervals for each slope.

```r
#plot the point estimates on a plot, where the position on the x-axis will be
# 1, 2, 3 for the SLR values and 6,7,8 for the MLR values
xposition <- c(1,2,3,6,7,8)
plot(coefs ~ xposition,
     type="p", ylim=c(-3, 6), xlim=c(0,8), bty="n", xaxt="n",
     ylab="", xlab="")

# add the horizontal line of no effect at 0
abline(h=0, lty="dashed")

# adds a label to distinguish the points
text(2, 5.5, "Coefficients in SLRs")
text(7, 5.5, "Coefficients in MLR")

# loop to add confidence intervals and labels to plot
for(i in 1:6){
    # get the x position to plot at
    x <- xposition[i]
    # adds a label to distinguish the points
    text(x, -2.75, names(coefs)[i])
    # plots a vertical lines for the CI of each point
    lines(c(x,x), confints[i,], lwd=3)
}
```



This plot allows us to visually investigate the relationship present between certain predictors and the response, as well as compare the estimates from the simple linear models to those from the multiple linear model. It

also allows us to view quickly the variation in each estimate as well as whether that slope is significantly different from 0.

MCQ: Why are the confidence intervals on the coefficient of each predictor narrower in the multiple linear model compared to the separate simple linear models? - The predictors we generated are strongly correlated. - The error variance is not estimated correctly in the simple linear models. - The true relationship is not estimated correctly in the simple linear models. - More than one option is correct. *** - None of the options are correct.

Impact of Correlation Between X_1 and X_2

In this section, we will now introduce a correlation between two of our predictors, X_1 and X_2. To begin, we will consider only moderate correlation (0.5) between them and investigate the impact of this on the estimates and confidence interval widths (i.e. length of the bar) from simple linear models and the multiple linear model.

First we need to generate predictors with this correlation included. Since we already created a covariance matrix without correlation, we simply need to replace the values in positions [1,2] and [2,1] with our correlation. Then, generate our predictors anew using this covariance matrix. Continue generating the rest of the data as before (no changes need to be made).

```
# replace elements [1,2] and [2,1] with 0.5 to incorporate the correlation
covariance <- diag(c(1,1,1))
covariance[1,2] <- covariance[2,1] <- 0.5
# re-generate our predictors with this new covariance matrix
X <- mvrnorm(n, mu = c(0,0,0), Sigma = covariance)
# generate new responses
Xmat <- cbind(1, X)
y <- Xmat %*% betas + e
```

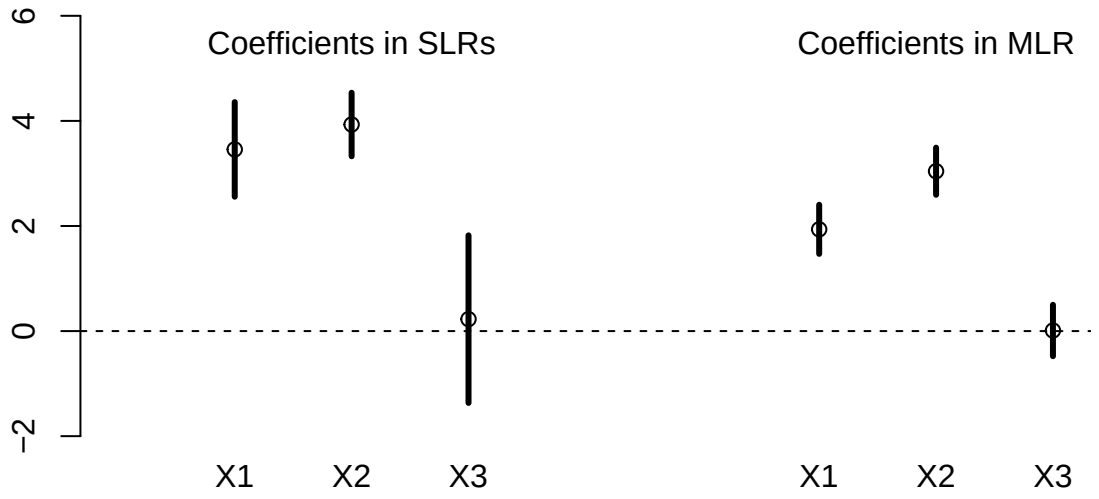
Now fit our models, like we did before (3 SLRs and 1 MLR), and extract the slope coefficients in all 4 models, as well as the 95% confidence intervals.

```
# fit the SLR and MLE models and extract coefficients
model1b <- lm(y ~ X[,1])
model2b <- lm(y ~ X[,2])
model3b <- lm(y ~ X[,3])
MLRb <- lm(y ~ X)

coefs <- c(coef(model1b)[2], coef(model2b)[2], coef(model3b)[2], coef(MLRb)[2:4])
names(coefs) <- c("X1", "X2", "X3", "X1", "X2", "X3")

# extract confidence intervals of slopes in all 4 models
confints <- rbind(confint(model1b)[2,], confint(model2b)[2,], confint(model3b)[2,],
                  confint(MLRb)[2:4,])
```

Finally, create the same type of plot as before, but using the information we extracted from models fit on the new data with moderately correlated X_1 and X_2.

```
#plot the point estimates on a plot, where the position on the x-axis will be
# 1, 2, 3 for the SLR values and 6,7,8 for the MLR values
xposition <- c(1,2,3,6,7,8)
plot(coefs ~ xposition,
     type="p", ylim=c(-3, 6), xlim=c(0,8), bty="n", xaxt="n",
     ylab="", xlab="")
```

```
# add the horizontal line of no effect at 0
abline(h=0, lty="dashed")

# adds a label to distinguish the points
text(2, 5.5, "Coefficients in SLRs")
text(7, 5.5, "Coefficients in MLR")

# loop to add confidence intervals and labels to plot
for(i in 1:6){
    # get the x position to plot at
    x <- xposition[i]
    # adds a label to distinguish the points
    text(x, -2.75, names(coefs)[i])
    # plots a vertical lines for the CI of each point
    lines(c(x,x), confints[i,], lwd=3)
}
```



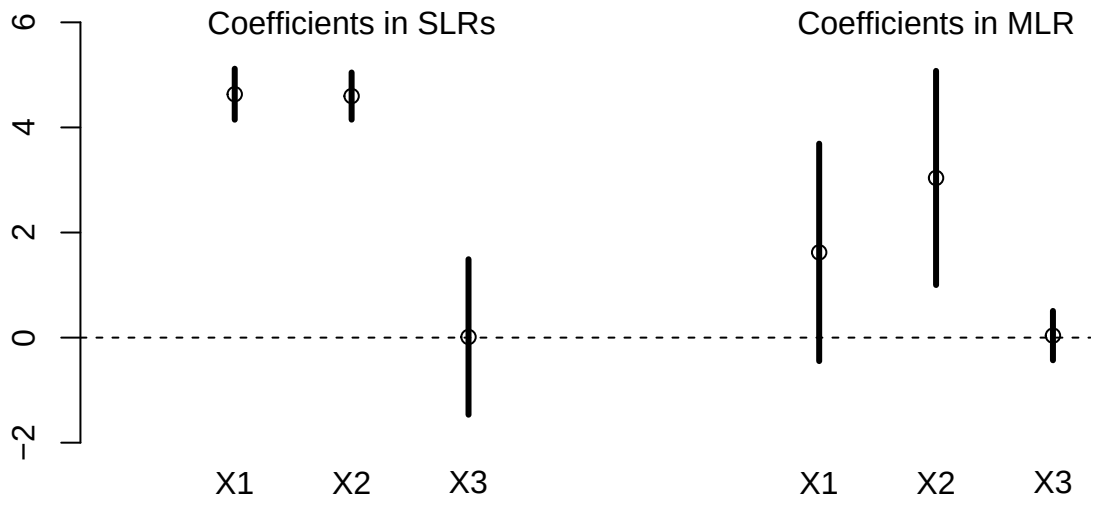
MCQ: Does there appear to be much difference in the widths of the confidence intervals compared to when there was no correlation present in the predictors? - Yes - No - Not totally sure ***

With moderate correlation, we generally don't see too much of a difference in the widths of the confidence intervals compared to when no correlation was present (although it generally depends on your data - sometimes it can be noticeable and other times it won't). We do still observe that the MLR confidence intervals are narrower than the SLR ones due to the fact we are specifying the correct model and therefore the error variance is being estimated more precisely (because we are explaining more variation in the response with

more predictors).

What if we had very strong correlation present between X_1 and X_2? Update the R code below to add a correlation of 0.98 between these two predictors.

```
covariance <- diag(c(1,1,1))
covariance[1,2] <- covariance[2,1] <- 0.98
# re-generate our predictors with this new covariance matrix
X <- mvrnorm(n, mu = c(0,0,0), Sigma = covariance)
# generate new responses
Xmat <- cbind(1, X)
y <- Xmat %*% betas + e

# fit the SLR and MLE models and extract coefficients
model1c <- lm(y ~ X[,1])
model2c <- lm(y ~ X[,2])
model3c <- lm(y ~ X[,3])
MLRc <- lm(y ~ X)

coefs <- c(coef(model1c)[2], coef(model2c)[2], coef(model3c)[2], coef(MLRc)[2:4])
names(coefs) <- c("X1", "X2", "X3", "X1", "X2", "X3")

# extract confidence intervals of slopes in all 4 models
confints <- rbind(confint(model1c)[2,], confint(model2c)[2,], confint(model3c)[2,],
                  confint(MLRc)[2:4,])


#plot the point estimates on a plot, where the position on the x-axis will be
# 1, 2, 3 for the SLR values and 6,7,8 for the MLR values
xposition <- c(1,2,3,6,7,8)
plot(coefs ~ xposition,
     type="p", ylim=c(-3, 6), xlim=c(0,8), bty="n", xaxt="n",
     ylab="", xlab="")

# add the horizontal line of no effect at 0
abline(h=0, lty="dashed")

# adds a label to distinguish the points
text(2, 6, "Coefficients in SLRs")
text(7, 6, "Coefficients in MLR")

# loop to add confidence intervals and labels to plot
for(i in 1:6){
    # get the x position to plot at
    x <- xposition[i]
    # adds a label to distinguish the points
    text(x, -2.75, names(coefs)[i])
    # plots a vertical lines for the CI of each point
    lines(c(x,x), confints[i,], lwd=3)
}
```



MCQ: What do you notice about the estimates of the slopes now? - Some slope estimates in the simple linear models are very different from the true values. - The widths of the confidence intervals in the multiple linear model are much wider. - X_1 no longer appears significantly different from 0 in the presence of X_2 and X_3. - X_1 and X_2 appear to have the same estimated slope in the simple linear models. - More than one of these are true. ***

Inflation of Standard Errors

We have seen some drastic changes in how our models use the data to estimate the true population relationship. It seems as if when two predictors are highly correlated, our confidence intervals get wider and this can lead us to potentially conclude that X_1 is not significantly related to the response when in reality the only predictor not related to the response is X_3. We also notice that in the simple linear models, when two predictors are highly correlated, the simple models estimate the same slope for both predictors, even though these are not the true population values.

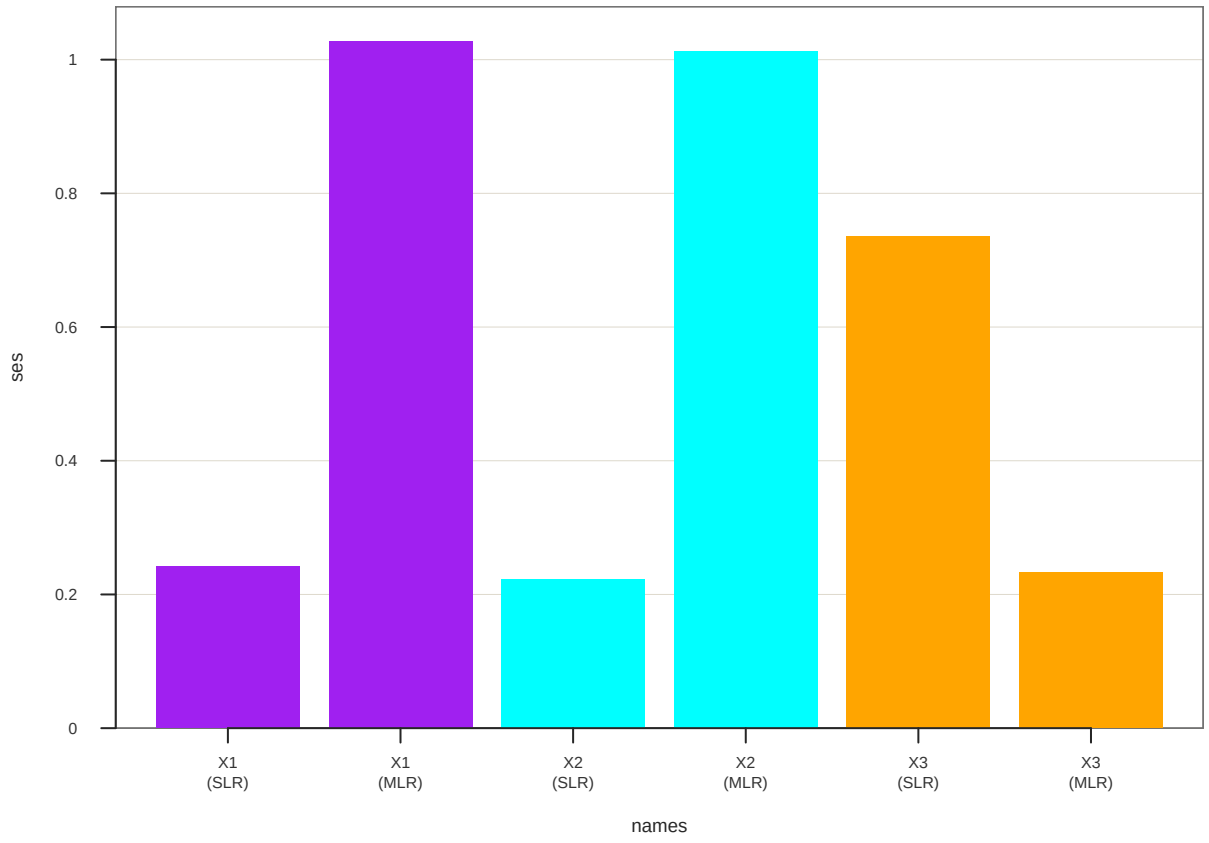
If we compare the standard errors we estimate from each model, we can see how the correlation changes the standard errors. This can be visualized with Bar Chart using the BarChart() function in the lessR package. We need to define the groups (each model's slope estimate) and the value we wish to plot (the standard errors of each model's slope estimate).

```
# extract standard errors of each slope from MLR model and SLR models
ses <- c(summary(model1c)$coef[2,2], summary(MLRc)$coef[2,2],
         summary(model2c)$coef[2,2], summary(MLRc)$coef[3,2],
         summary(model3c)$coef[2,2], summary(MLRc)$coef[4,2])

# create group names
```

```r
names <- c("X1 (SLR)", "X1 (MLR)","X2 (SLR)", "X2 (MLR)", "X3 (SLR)", "X3 (MLR)")

# create the Bar Chart
library(lessR)
```

```
##
## lessR 4.2.6                          feedback: gerbing@pdx.edu
## ----------------------------------------------------------------
## > d <- Read("")   Read text, Excel, SPSS, SAS, or R data file
##   d is default data frame, data= in analysis routines optional
##
## Learn about reading, writing, and manipulating data, graphics,
## testing means and proportions, regression, factor analysis,
## customization, and descriptive statistics from pivot tables.
##   Enter:  browseVignettes("lessR")
##
## View changes in this and recent versions of lessR.
##   Enter: news(package="lessR")
##
## **Newly Revised**: Interactive data analysis.
##   Enter: interact()
```

```r
BarChart(names, ses, data=data.frame(cbind(names, ses)), fill=c("purple", "purple", "cyan", "cyan", "ora
```




```
## >>> Suggestions
## Plot(ses, names) # lollipop plot
##
```

```
##  Plotted Values
##  --------------
##   X1 (SLR)  X1 (MLR)  X2 (SLR)  X2 (MLR)  X3 (SLR)  X3 (MLR)
##      0.241     1.028     0.223     1.012     0.736     0.233
```

MCQ: Approximately how many times higher are the bars for the slopes of X_1 and X_2 in the MLR model compared to the bars for the same slopes in the SLR model? - About 5 times higher *** - About 10 times higher - About 15 times higher - About 20 times higher

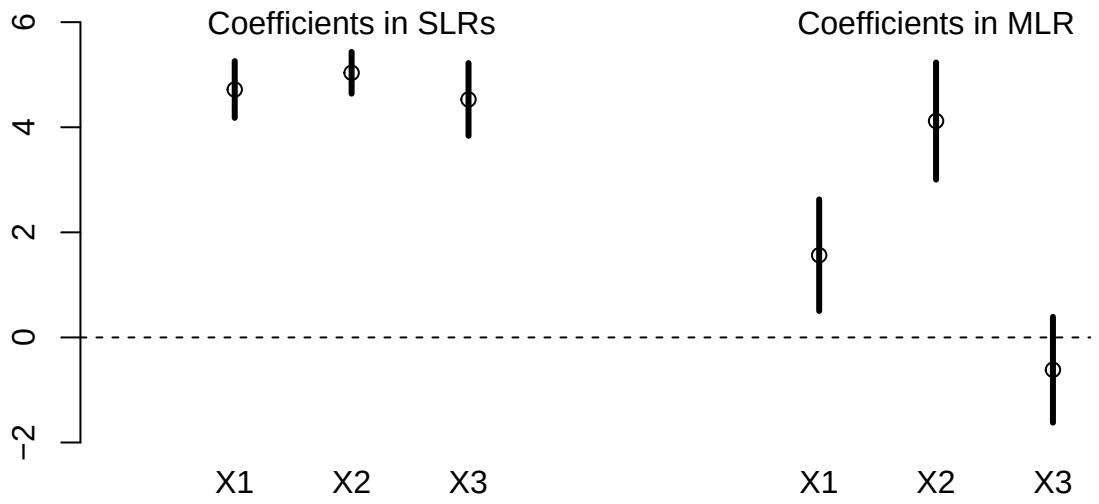
So when we have predictors in a model that are highly correlated, our model will incorrectly estimate the standard errors of those coefficients. Specifically we see that the standard errors are **inflated** by a certain amount relative to how correlated those predictors are.

Impact of All Predictors Being Correlated

To understand the impact of all predictors being correlated when fitting a multiple linear model, we can consider a similar approach to visualizing these. We can first start with generating data in which all 3 predictors have a correlation of 0.9 and then fitting separate simple linear models and a multiple linear model. We shall also extract the estimated slopes from all models as well as the confidence intervals for those slopes. Finally, we will make a plot that displays the confidence intervals from all models.

```
# first create a matrix of 0.9 everywhere, then change the diagonal to be 1
covariance <- matrix(rep(0.9, 9), nrow=3, ncol=3)
diag(covariance) <- c(1,1,1)
# re-generate our predictors with this new covariance matrix
X <- mvrnorm(n, mu = c(0,0,0), Sigma = covariance)
# generate new responses
Xmat <- cbind(1, X)
y <- Xmat %*% betas + e

# fit the SLR and MLE models and extract coefficients
model1d <- lm(y ~ X[,1])
model2d <- lm(y ~ X[,2])
model3d <- lm(y ~ X[,3])
MLRd <- lm(y ~ X)

coefs <- c(coef(model1d)[2], coef(model2d)[2], coef(model3d)[2], coef(MLRd)[2:4])
names(coefs) <- c("X1", "X2", "X3", "X1", "X2", "X3")

# extract confidence intervals of slopes in all 4 models
confints <- rbind(confint(model1d)[2,], confint(model2d)[2,], confint(model3d)[2,],
                  confint(MLRd)[2:4,])


#plot the point estimates on a plot, where the position on the x-axis will be
# 1, 2, 3 for the SLR values and 6,7,8 for the MLR values
xposition <- c(1,2,3,6,7,8)
plot(coefs ~ xposition,
     type="p", ylim=c(-3, 6), xlim=c(0,8), bty="n", xaxt="n",
     ylab="", xlab="")

# add the horizontal line of no effect at 0
abline(h=0, lty="dashed")
```

```r
# adds a label to distinguish the points
text(2, 6, "Coefficients in SLRs")
text(7, 6, "Coefficients in MLR")

# loop to add confidence intervals and labels to plot
for(i in 1:6){
    # get the x position to plot at
    x <- xposition[i]
    # adds a label to distinguish the points
    text(x, -2.75, names(coefs)[i])
    # plots a vertical lines for the CI of each point
    lines(c(x,x), confints[i,], lwd=3)
}
```



MCQ: Which model(s) seem to be estimating the true association of each predictor with the response? - The multiple linear model *** - Simple model with X_1 - Simple model with X_2 - Simple model with X_3 - More than one of these models - None of these models

---

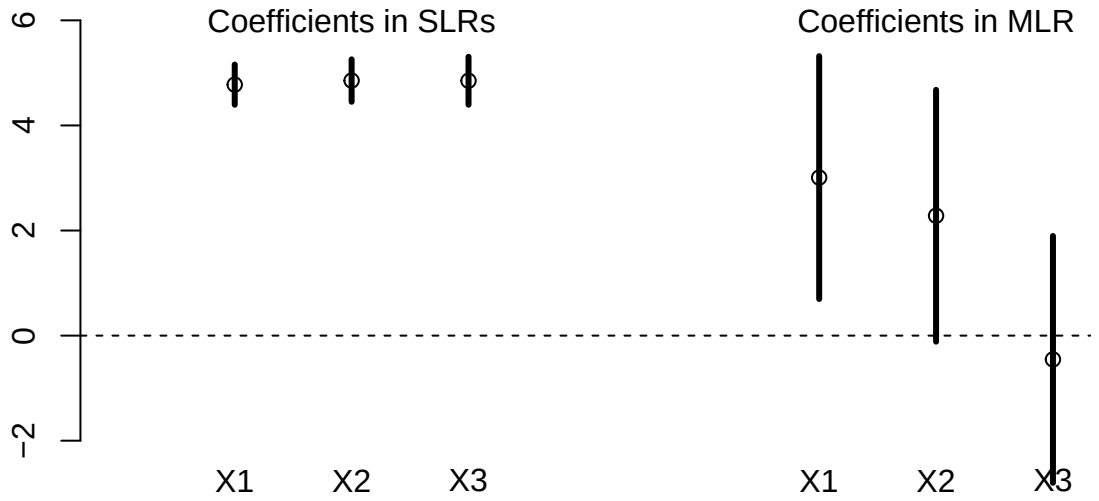
Worsening Correlation

It appears that even with a correlation of 0.9 among all three of the predictors, the multiple linear model still estimates the correct relationship (more or less) with slightly wider confidence intervals around the estimates. So we still see the impact of correlation on inflating the standard errors (in this case, by about a factor of 2).

Will this always be the case? If we increase the correlation to 0.98, will we still obtain estimates that appear

to reflect the true association? Repeat the previous part but now set the correlation to 0.98.

```
# first create a matrix of 0.9 everywhere, then change the diagonal to be 1
covariance <- matrix(rep(0.98, 9), nrow=3, ncol=3)
diag(covariance) <- c(1,1,1)
# re-generate our predictors with this new covariance matrix
X <- mvrnorm(n, mu = c(0,0,0), Sigma = covariance)
# generate new responses
Xmat <- cbind(1, X)
y <- Xmat %*% betas + e

# fit the SLR and MLE models and extract coefficients
model1d <- lm(y ~ X[,1])
model2d <- lm(y ~ X[,2])
model3d <- lm(y ~ X[,3])
MLRd <- lm(y ~ X)

coefs <- c(coef(model1d)[2], coef(model2d)[2], coef(model3d)[2], coef(MLRd)[2:4])
names(coefs) <- c("X1", "X2", "X3", "X1", "X2", "X3")

# extract confidence intervals of slopes in all 4 models
confints <- rbind(confint(model1d)[2,], confint(model2d)[2,], confint(model3d)[2,],
                  confint(MLRd)[2:4,])


#plot the point estimates on a plot, where the position on the x-axis will be
# 1, 2, 3 for the SLR values and 6,7,8 for the MLR values
xposition <- c(1,2,3,6,7,8)
plot(coefs ~ xposition,
     type="p", ylim=c(-3, 6), xlim=c(0,8), bty="n", xaxt="n",
     ylab="", xlab="")

# add the horizontal line of no effect at 0
abline(h=0, lty="dashed")

# adds a label to distinguish the points
text(2, 6, "Coefficients in SLRs")
text(7, 6, "Coefficients in MLR")

# loop to add confidence intervals and labels to plot
for(i in 1:6){
    # get the x position to plot at
    x <- xposition[i]
    # adds a label to distinguish the points
    text(x, -2.75, names(coefs)[i])
    # plots a vertical lines for the CI of each point
    lines(c(x,x), confints[i,], lwd=3)
}
```



MCQ: Do any of these models appear to estimate the correct association of each predictor to the response? - Yes - No ***

So when all predictors are correlated to one another, it appears as if, in addition to the inflated variances visible in the wide confidence intervals, we also struggle to capture the true relationships present in the population.

Impact of Correlation Between X_1 and X_3

We have seen the general impact of strongly correlated predictors on the standard errors of slopes in a multiple linear model. Now we can see what happens in a slightly different scenario: we will still have two strongly correlated predictors (X_1 and X_3) but here we will forget to include X_1 in our multiple linear model. By doing this, we are misspecifying our model (estimating the wrong population relationship) and therefore automatically introducing bias into our estimates. However, we will see something else interesting occurs.

Omitting a significant predictor

Recall that the true population relationship between the response and the three predictors is

$$Y = 1 + 2X_1 + 3X_2 + 0X_3 + \epsilon, \quad \epsilon \sim N(0, 2.25).$$

This tells us that only X_3 has no association with the response in the presence of the other two predictors.

For this section, we will set elements [1,3] and [3,1] in the covariance matrix to be 0.9, giving us predictors that are generate with a strong correlation between X_1 and X_3. We will still fit 3 separate simple linear models, but will now fit a multiple linear model using only X_2 and X_3. We will extract the slopes and

confidence intervals from each model again. At the end, print the summary of the simple linear model with X_2 and the simple linear model with X_3.

```
# edit the covariance matrix to introduce the correlation
covariance <- diag(c(1,1,1))
covariance[1,3] <- covariance[3,1] <- 0.9

# re-generate our predictors with this new covariance matrix
X <- mvrnorm(n, mu = c(0,0,0), Sigma = covariance)
# generate new responses
Xmat <- cbind(1, X)
y <- Xmat %*% betas + e

# fit the SLR and MLE models and extract coefficients
model1e <- lm(y ~ X[,1])
model2e <- lm(y ~ X[,2])
model3e <- lm(y ~ X[,3])
MLRe <- lm(y ~ X[,-1])

coefs <- c(coef(model1e)[2], coef(model2e)[2], coef(model3e)[2], coef(MLRe)[2:3])
names(coefs) <- c("X1", "X2", "X3", "X2", "X3")

# extract confidence intervals of slopes in all 4 models
confints <- rbind(confint(model1e)[2,], confint(model2e)[2,], confint(model3e)[2,],
                  confint(MLRe)[2:3,])

# print the model summaries for model1d and model3d
summary(model2e)
```

```
##
## Call:
## lm(formula = y ~ X[, 2])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0269 -1.8597  0.0577  1.3796  4.8159
##
## Coefficients:
##             Estimate Std. Error t value           Pr(>|t|)
## (Intercept)   1.0272     0.3518    2.92            0.00532 **
## X[, 2]        3.2758     0.3112   10.53 0.0000000000000462 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.396 on 48 degrees of freedom
## Multiple R-squared:  0.6977, Adjusted R-squared:  0.6914
## F-statistic: 110.8 on 1 and 48 DF,  p-value: 0.0000000000000462
```

```
summary(model3e)
```

```
##
## Call:
## lm(formula = y ~ X[, 3])
##
## Residuals:
```

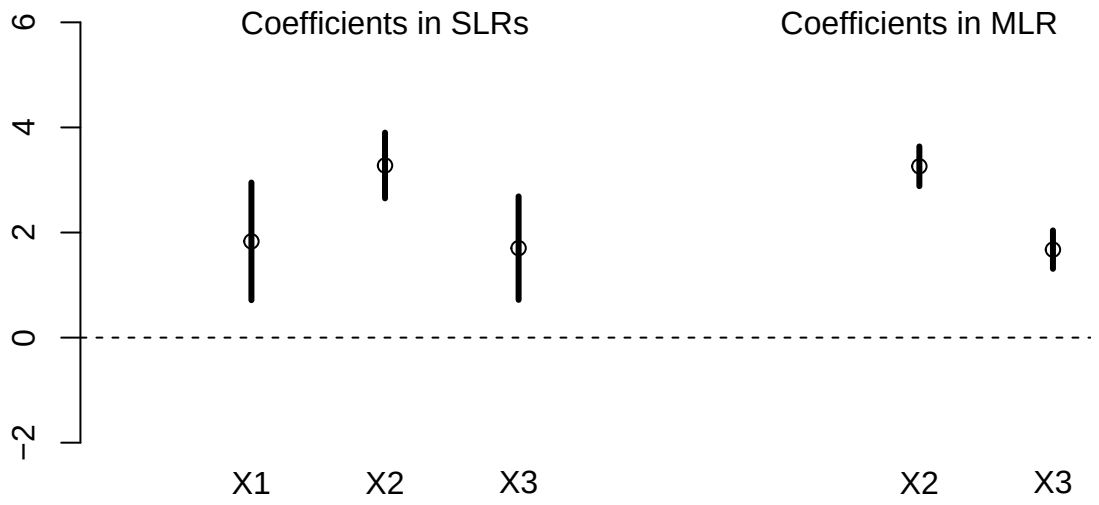
```
##    Min     1Q Median     3Q    Max
## -8.928 -2.635  0.123  2.721  6.804
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1808     0.5526   3.946 0.000258 ***
## X[, 3]        1.7028     0.4897   3.477 0.001086 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.894 on 48 degrees of freedom
## Multiple R-squared:  0.2012, Adjusted R-squared:  0.1846
## F-statistic: 12.09 on 1 and 48 DF,  p-value: 0.001086
```

MCQ: Which of the below is true regarding these two simple linear models? - The slope of X_3 is significantly different from 0. - The slopes X_3 seems to indicate the same association (more or less) with Y as X_1 should have. - The slope of X_2 seems to be an unbiased estimate while the slope of X_3 seems to be biased - More than one of these is true. ***

Unmeasured Confounders

To visualize the effect of this strong correlation between X_1 and X_3 in the situation when X_1 is not included in the multiple linear model, we can produce the same plot as before. Pay particular attention to the effect of this correlation paired with fitting the wrong model as a multiple linear model.

```r
#plot the point estimates on a plot, where the position on the x-axis will be
# 1, 2, 3 for the SLR values and 6,7 for the MLR values
xposition <- c(1,2,3,6,7)
plot(coefs ~ xposition,
     type="p", ylim=c(-3, 6), xlim=c(0,7), bty="n", xaxt="n",
     ylab="", xlab="")

# add the horizontal line of no effect at 0
abline(h=0, lty="dashed")

# adds a label to distinguish the points
text(2, 6, "Coefficients in SLRs")
text(6, 6, "Coefficients in MLR")

# loop to add confidence intervals and labels to plot
for(i in 1:5){
    # get the x position to plot at
    x <- xposition[i]
    # adds a label to distinguish the points
    text(x, -2.75, names(coefs)[i])
    # plots a vertical lines for the CI of each point
    lines(c(x,x), confints[i,], lwd=3)
}
```



MCQ: Which true association does the slope of X_3 from the multiple linear model represent? - The association between X_1 and Y. *** - The association between X_2 and Y. - The association between X_3 and Y. - None of the above.

What we have just seen is an effect called an **unmeasured confounder** effect. It occurs when there is a strong correlation between predictors, but the predictor that is truly associated with the response is not measured and therefore cannot be included in a model. Due to the correlation, the predictor that is correlated with this variable, when included in the model, will capture the effect of this variable. This is because the correlation between them implies that they represent something similar and thus by omitting one, the effect is still captured by the variable that is included.

This can be misleading because, contextually, it may not make sense for the variable that is used as a predictor to have this effect/association with Y. When this is observed in a model (e.g. an association that contextually does not make sense or agree with existing literature), it usually highlights that this variable is correlated with something that was unmeasured. This is a separate issue compared to **multicollinearity**, which only happens when predictors that are actually included in the model together happen to be highly correlated.