Statistical Thinking

Week 10 Tutorial

Introduction

The purpose of the tutorial this week is to explore aspects of multiple linear regression (MLR) analysis, from the frequentist perspective using some generated data. Part A details the simulated data to be used in Part B, Part B examines various strategies used under a frequentist approach, including those helpful for the selection of the explanatory variables.

We will need (at least) the following **R** package libraries:

```
library(tidyverse)
library(broom)
library(GGally)
library(car)
library(meifly)
library(gridExtra)
library(MCMCpack)
library(coda)
```

Part A: Simulation

We will simulate $n = 50$ observations of each of the following variables:

- $z_{j,i} \overset{i.i.d}{\sim} N(0,1)$ for $j = 1, 2, 3$ and $i = 1, 2, \ldots, n$. This produces ***three*** independent sequences of n *i.i.d.* standard normal random variables.
- $x_{1,i} \overset{i.i.d.}{\sim} Uniform(min = 5,\ max = 15)$, for $i = 1, 2, \ldots, n$.
- $x_{2,i} \overset{i.i.d.}{\sim} Student - t$ with $\nu = 4$ degrees of freedom, for $i = 1, 2, \ldots, n$.
- $x_{3,i} \overset{i.i.d.}{\sim} Gamma(shape = 3,\ scale = 5)$, for $i = 1, 2, \ldots, n$.
- $x_{4,i} = 3 + 2\, z_{1,i}$, for $i = 1, 2, \ldots, n$.
- $x_{5,1} = -1 - 0.96 z_{1,i} + 0.28 * z_{2,i}$, for $i = 1, 2, \ldots, n$.

Now we can simulate n observations for the response variable y according to:

$$y_i = \beta_0 + \beta_1\, x_{1,i} + \beta_2\, x_{2,i} + \beta_3\, x_{3,i} + \beta_4\, x_{4,i} + \beta_5\, x_{5,i} + \sigma\, z_i,$$

where $\beta_0 = 12,\ \beta_1 = 2.5,\ \beta_2 = 4,\ \beta_3 = 1,\ \beta_4 = 9,\ \beta_5 = 5$ and $\sigma = 20$.

Simulation of these variables are completed in the code chunk below.[1]

[1] Note that as **R** does not use a zero value to index an element of a vector, the indices of the individual β coefficients is increased by one (*i.e.* β_0 is coded as tb[1] and so on).

```
n <- 50
set.seed(345820)

z1 <- rnorm(n)
z2 <- rnorm(n)
z3 <- rnorm(n)

x1 <- runif(n, min = 5, max = 15)
x2 <- rt(n, df = 4)
x3 <- rgamma(n, shape = 3, scale = 5)
x4 <- 3 + 2 * z1
x5 <- -1 - 0.96 * z1 + 0.28 * z2

tb <- c(12, 2.5, 4, 1, 9, 5)  #tb for 'true beta'
ts <- c(20)  #ts for 'true sigma'

y <- tb[1] + tb[2] * x1 + tb[3] * x2 + tb[4] * x3 + tb[5] * x4 +
    tb[6] * x5 + ts[1] * z3
```

We will put all variables in a *tibble* named *simd*.

```
simd <- tibble(x1 = x1, x2 = x2, x3 = x3, x4 = x4, x5 = x5, y = y)
```

Have a quick look at *simd*.

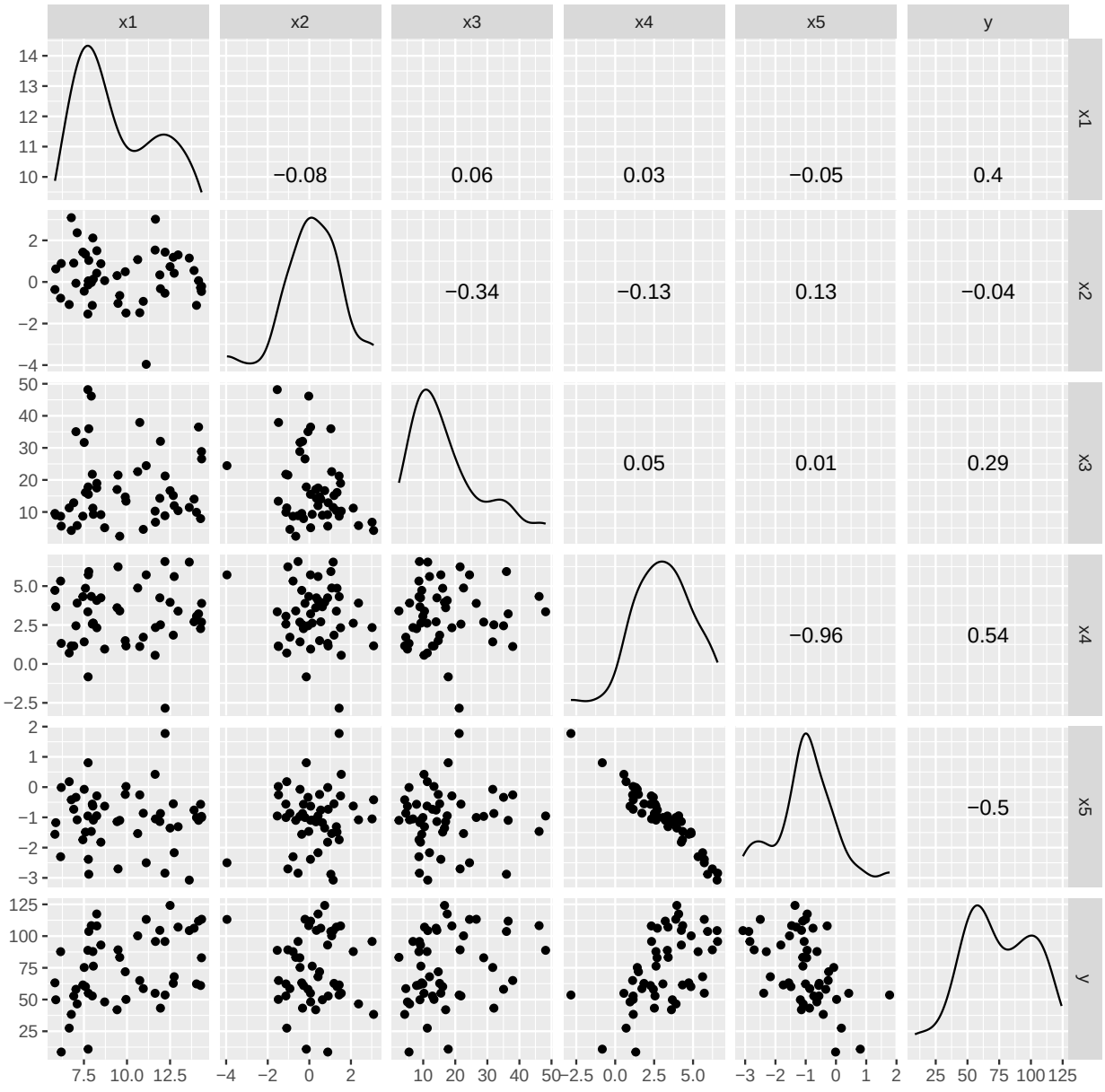
We want to check out our data. In particular, we want to see if there is any redundant variables (that is, we want to see if there could be multicollinearity). So we will use scatterplots.

We can produce a scatterplot matrix of all columns of *simd* (including *y*) using the **GGally**::*ggscatmat*() function.

This will show correlations between the regressors (x's) and also correlations of y for all of the x's.

The lower diagonal are the scatterplots, the upper diagonal is the estimated correlation, and the diagonals are the kernel densities for the variables.

```
ggscatmat(simd)
```



Q1. Which pairs of x variables in the scatterplot matrix appear to show a relationship between each other?[2] Do the correlations between the x and y variables seem to align with how the data were simulated?

Answer Q1. Most of these variables are not linearly correlated. The thresholds for the absolute value of the sampled correlation coefficient is $2/sqrtn = 0.2828$. $x2$ and $x3$ have a "significant" correlation coefficient,

[2]Note that a simple test of H_0:$\rho = 0$ vs. H_1:$\rho \neq 0$ will reject H_0 at the 5% level if $|r| > 1.96/\sqrt{n}$ (often rounded to $2/\sqrt{n}$), where r is the relevant sample correlation coefficient. However also be aware that correlation is only a measure of *linear* dependence. If the relationship is not linear, this correlation coefficient may be affected by skewness in one or the other variable.

but the relationship does not look strong in the plot and may be impacted by a few observations in the right tail of x_3.

However, x_5 and x_4 are almost perfectly negatively correlated. The scatterplot looks very linear and the correlation is -0.96. This may indicate a problem with multicollinearity.

The correlation between x_2 and y is negative, even though $\beta_2 = 2.5$. The correlation is low however, so it probably means that β_2 is insignificant.

The variable x_5 is negatively correlated with y even though it had a positive coefficient. This is probably due to the multicollinearity between x_4 and x_5. However we do need to remember that this can sometimes happen. The scatterplots are simple correlations (so just look at the relationship between the two variables ignoring all of the other variables), while the regression coefficients are essentially partial correlations. Once the other variables have been accounted for, the relationship between x and y can be different to the simple correlation between them.

Part B: Frequentist MLR

Next we'll pretend we didn't simulate the data, and instead try to identify a suitable multiple linear regression relationship between the response variable y and the five different regressors, including an intercept term. We'll start with fitting the largest model (the one we used to generate y), using OLS. We'll call this model "Model 1".

```
# not given to students

M1 <- lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = simd)  # Model 1
summary(M1)
#
# Call:
# lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = simd)
#
# Residuals:
#    Min     1Q Median     3Q    Max
# -43.46 -11.50   0.09  14.50  39.37
#
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -4.7923    13.6710  -0.351 0.727600
# x1            4.0573     1.1032   3.678 0.000637 ***
# x2            3.6595     2.5017   1.463 0.150634
# x3            0.7351     0.2863   2.568 0.013698 *
# x4            9.3386     5.2839   1.767 0.084101 .
# x5            3.2502    10.7491   0.302 0.763797
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 20.46 on 44 degrees of freedom
# Multiple R-squared:  0.5265,  Adjusted R-squared:  0.4727
# F-statistic: 9.785 on 5 and 44 DF,  p-value: 2.457e-06
```

Consider the output from the fit of Model 1.

Q2. According to the model output, which regression coefficients are (individually) significant at the 5% level? Do the estimated coefficients seem "close" to the true values used to simulate the data?

Answer Q2. Look at the p-values first. The $\hat{\beta}_0$, $\hat{\beta}_2$, $\hat{\beta}_4$ and $\hat{\beta}_5$ all have p-values greater than 0.05. The estimated values are also a bit off, but β_2, β_3 and β_4 seem "close" to the true values (although this is subjective.)

Q3. What can you say about the estimated standard errors of the estimated regression coefficients?

Answer Q3. The size of the standard errors are pretty large! Of course, the sample size is quite small for this size of a model. We also have the issue (to be discovered below if not already noticed) that there is multi-collinearity present.

Q4. What are the R^2 and the adjusted R^2 values for the fitted Model 1?

Answer Q4. Look at the summary to see the result: $R^2 = 0.5265$ or 52.65% and adjusted $R^2 = 0.4727$ (note that we would just interpret the R^2 and use the adjusted R^2 to compare models). The adjusted R^2 penalises (reduces) the R^2 because there are many regressors in the model, some of which are not statistically significant.

Next, calculate the variance inflation factors (VIF) associated with the regressors in Model 1.

```
# not given to students
vif(M1)
#        x1        x2        x3        x4        x5
#  1.014920  1.159435  1.180894 12.260319 12.313398
```

Q5. Find all variables associated with an excessively large VIF. How should the VIFs be interpreted? (Should you remove *all* of these variables from your model? Discuss possible strategies to decide on the variables to exclude from your model due to the large VIF values.)

Answer Q5. x4 and x5 have VIFs over 10 (the threshold). Values over 10 mean than if we fit the auxiliary linear regression like $x_4 = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_5$, we'll get an R^2 (R_4^2) of over 90%. (Since $VIF = \frac{1}{1-R_4^2}$ where the R_4^2 is for this special regression for x4. Solve for R_4^2 when VIF=12.26 and you'll get $R^2 = 91.84$.)

We can run the regression and see the R^2:

```
# not given to students
r4 <- lm(x4 ~ x1 + x2 + x3 + x5, simd)

summary(r4)
#
# Call:
# lm(formula = x4 ~ x1 + x2 + x3 + x5, data = simd)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -1.3716 -0.3383  0.0343  0.3216  1.1101
#
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  1.040502   0.353130   2.947  0.00508 **
# x1          -0.013978   0.031053  -0.450  0.65477
# x2           0.020610   0.070513   0.292  0.77141
# x3           0.010702   0.007917   1.352  0.18319
# x5          -1.948006   0.087402 -22.288  < 2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.5772 on 45 degrees of freedom
# Multiple R-squared:  0.9184,  Adjusted R-squared:  0.9112
# F-statistic: 126.7 on 4 and 45 DF,  p-value: < 2.2e-16
```

It makes sense to remove at least one variable since at least one must be close to redundant given the other regressors. (i.e. there is multi-collinearity present). Given there are two values with nearly the same large VIF, it makes sense to take only one out at a time. (We also know this makes sense since we simulated the data. . . but we can always check the VIFs after one regressor is removed and if the VIF for the other variable is still large we can then take out the second variable.)

The code chunk below fits a *modified* version of Model 1 (dropping x_5) and saves the result in an object named *MM1*.

```
MM1 <- lm(formula = y ~ x1 + x2 + x3 + x4, data = simd)  # 'Modified' Model 1
tidy.MM1 <- tidy(MM1)
glance.MM1 <- glance(MM1)
```

Look at the output from the Modified Model 1.

```
tidy.MM1  # not included for students
# # A tibble: 5 x 5
#   term        estimate std.error statistic    p.value
#   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
# 1 (Intercept)   -3.47     12.8      -0.271 0.788
# 2 x1             4.03      1.09      3.70  0.000579
# 3 x2             3.72      2.47      1.51  0.139
# 4 x3             0.753     0.277     2.71  0.00937
# 5 x4             7.81      1.51      5.18  0.00000504
```

```
glance.MM1  # not included for students
# # A tibble: 1 x 12
#   r.squared adj.r.squared sigma statistic     p.value    df logLik   AIC   BIC
#       <dbl>         <dbl> <dbl>     <dbl>       <dbl> <dbl>  <dbl> <dbl> <dbl>
# 1     0.526         0.483  20.2      12.5 0.000000665     4  -219.  449.  461.
# # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Q6. What is the value of p associated with Modified Model 1? *HINT:* It is not the **p-value**!

Answer Q6. p is the number of regression coefficients (number of regressors), including the intercept (including a constant regressor). So here $p = 5$.

Q7. According to the Modified Model 1 output, which regression coefficients are significant at the 5% level? Do the estimated coefficients seem "close" to the true values used to simulate the data?

Answer Q7. The estimated intercept and coefficient on x_2 are not (individually) significant. The estimated values are not spot on, but they seem closer to the corresponding true values than under Model 1.

Q8. Note the size of the standard errors associated with each estimated coefficient. How do these compare with the coefficients that were not significant in Model 1? (See Q2.) Given the way the data were simulated, do you have any ideas about why there might still be insignificant coefficients, even for the Modified Model 1?

Answer Q8. See the *glance.MM1* output. The standard errors for the remaining coefficients haven't changed all that much, but we have removed the coefficient that had the very large standard error (x5). Multi-collinearity will tend to inflate the standard errors.

Why is there still relatively large standard errors? We still have a small sample size here. It seems that the "signal" for the intercept, and for x2, are too small to be detected amid all of the noise in the data.

Q9. What are the values of: R^2 and the Adjusted R^2 values for the fitted Modified Model 1? Compare these values to those from the original Model 1 (see Q4.) Do you think removing regressor x_5 has improved or diminished the model?

Answer Q9. Need to look at *glance.MM1* to see the result: r.squared = 0.5255 or 52.55% and adjusted r.squared = 0.4833. We really can only compare the adjusted R^2 since the ordinary R^2 doesn't account for the number of regressors. We can see that the adjusted R^2 improves from 0.473 with the removal of x_5, so the model seems better without x_5 in it (on this measure, at least).

Next, calculate the variance inflation factors (VIF) associated with the regressors in Modified Model 1.

```
# not given to students
vif(MM1)
#       x1       x2       x3       x4
# 1.008953 1.151655 1.131160 1.018386
```

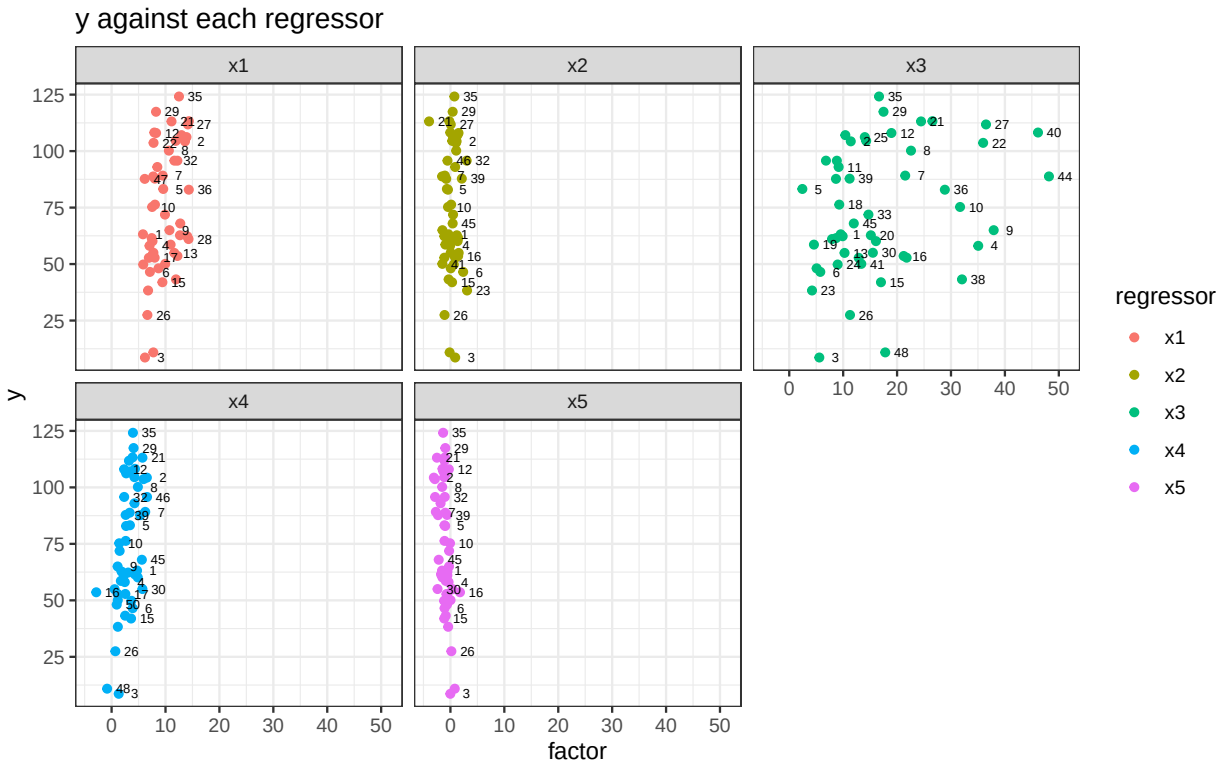
Q10. Given the VIF information, and the OLS results for the Modified Model 1, do you think the variable x_4 should be removed from the model? Provide a justification for your answer.

Answer Q10. The VIFs are all well below 10 now and the coefficient of x_4 is statistically significant. This suggests leaving x_4 in the model (for now, at least) as there are no longer any issues with multi-collinearity.

Change the value of p in the code chunk below to match the correct value for Modified Model 1. Then run the code chunk to extract the *leverage* and *Cook's D* measures associated with the fit of Modified Model 1 that exceed the "rule of thumb" threshold of $2 * p/n$. We will use the plots to try and identify any potentially influential observations for each regressor.

```
simd <- simd %>%
    mutate(rowid = 1:n)
aug.MM1 <- augment(MM1)
aug.MM1 <- aug.MM1 %>%
    mutate(rowid = 1:n)

p <- 1  # change this once you know the correct value of p

threshold <- 2 * p/n  # threshold for .hat and .cooksd

exceed_hat <- aug.MM1 %>%
    arrange(desc(.hat)) %>%
    dplyr::select(rowid, .hat) %>%
    filter(.hat > threshold)

exceed_cooksd <- aug.MM1 %>%
    arrange(desc(.cooksd)) %>%
    dplyr::select(rowid, .cooksd) %>%
    filter(.cooksd > threshold)

simd_longer <- simd %>%
    pivot_longer(-c(y, rowid), names_to = "regressor", values_to = "factor")

simd_longer %>%
    ggplot(aes(x = factor, y = y, colour = regressor)) + geom_point() +
    facet_wrap(~regressor, nrow = 2) + theme_bw() + geom_text(label = simd_longer$rowid,
    nudge_x = 3, colour = "black", size = 2, check_overlap = T) +
    ggtitle("y against each regressor")
```

Code chunk below has the correct value of p (not shown to students)



Q11. How many points exceed the leverage threshold? Find these points using their *rowid* value which is also shown on the faceted plot. (Some will be easier to find than others! Try filtering the data points to see the values of the regressors to make it easier to find the points.)

Answer Q11. Need to view the *exceed_hat* object. Then look at the plot produced above.

Code chunk below not shown to students

```
exceed_hat
# # A tibble: 3 x 2
#   rowid  .hat
#   <int> <dbl>
# 1    21 0.272
# 2    16 0.251
# 3    44 0.205

simd %>%
    dplyr::filter(rowid == 21 | rowid == 16 | rowid == 44)
# # A tibble: 3 x 7
#      x1    x2    x3    x4     x5     y rowid
#   <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <int>
# 1 12.2   1.43  21.2 -2.84  1.77   53.6    16
# 2 11.1  -3.96  24.4  5.72 -2.50  113.     21
# 3  7.73 -1.54  48.2  3.35 -0.954  88.8    44
```

Three observations have high threshold. Obs 44 is very close, and the others aren't too far away either, so this doesn't seem to be a major issue. We should still investigate these just to be sure.

Q12a. How many points exceed the Cook's D threshold?

Answer Q12. Need to view the *exceed_cooksd* object. No points are flagged as influential according to this measure.

Code chunk below not shown to students

```
exceed_cooksd
# # A tibble: 0 x 2
# # i 2 variables: rowid <int>, .cooksd <dbl>
```

No observations have a high Cook's D. So while those observations have high leverage (on the horizontal axis), they do not have a large deviation on the vertical axis. So overall, we probably don't need to be too concerned with the high leverage observations.

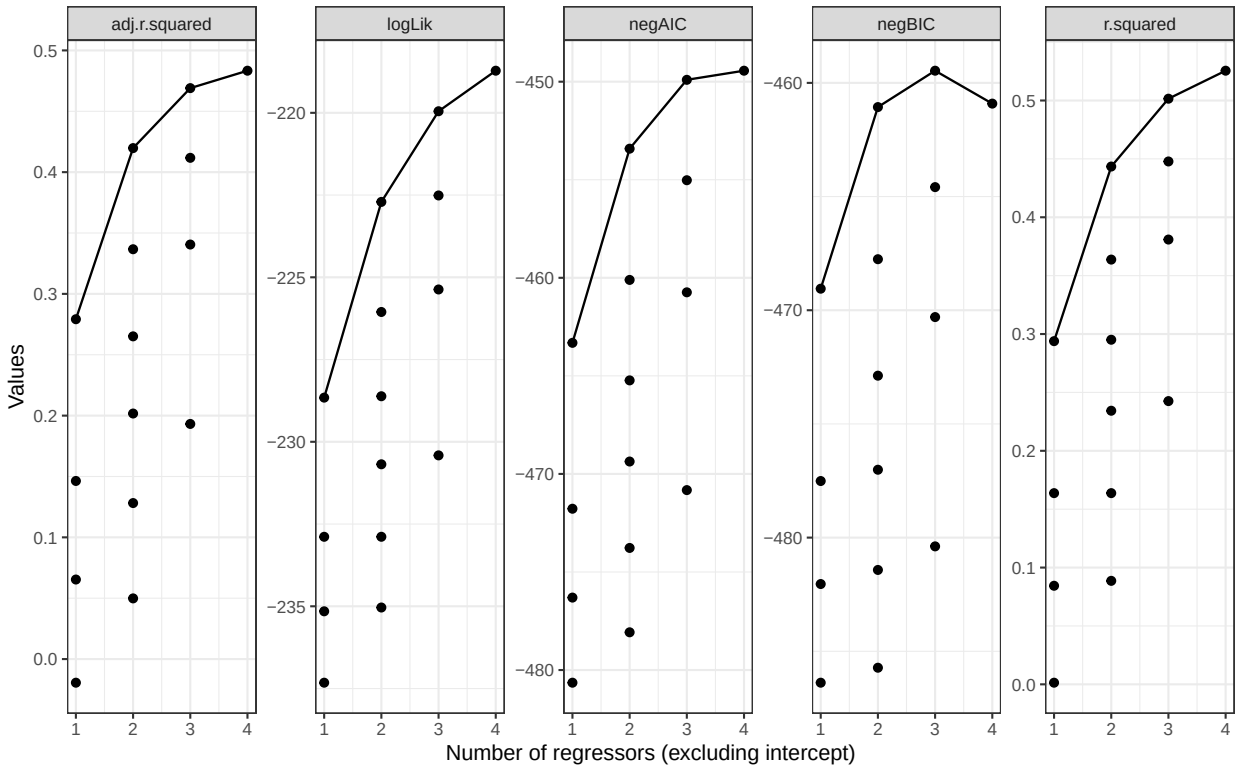
Use the code chunk below to fit all possible linear regression models that contains an intercept term and at least one of regressors used in Modified Model 1.

```
quiet <- function(x) {
    sink(tempfile())
    on.exit(sink())
    invisible(force(x))
}

all_mod <- quiet(fitall(y = simd$y, x = simd[, c(1:4)], method = "lm"))
summary(all_mod)
#     df      logL       AIC       BIC          R2       adjR2  n model
# m1   3 -232.8874 -471.7749 -477.5109 0.163739837  0.14631775 50     1
# m2   3 -237.3229 -480.6457 -486.3818 0.001396933 -0.01940730 50     2
# m3   4 -232.8871 -473.7741 -481.4222 0.163751860  0.12816683 50     3
# m4   3 -235.1528 -476.3056 -482.0417 0.084423086  0.06534857 50     4
# m5   4 -230.6836 -469.3672 -477.0153 0.234302921  0.20172007 50     5
# m6   4 -235.0376 -478.0752 -485.7233 0.088632789  0.04985121 50     6
# m7   5 -230.4126 -470.8252 -480.3853 0.242558914  0.19316058 50     7
# m8   3 -228.6577 -463.3154 -469.0514 0.293904760  0.27919444 50     8
# m9   4 -222.7080 -453.4160 -461.0640 0.443447357  0.41976427 50     9
# m10  4 -228.6150 -465.2299 -472.8780 0.295110286  0.26511498 50    10
# m11  5 -222.5120 -455.0240 -464.5841 0.447793355  0.41177988 50    11
# m12  4 -226.0536 -460.1072 -467.7553 0.363752825  0.33667848 50    12
# m13  5 -219.9522 -449.9043 -459.4644 0.501537125  0.46902868 50    13
# m14  5 -225.3699 -460.7398 -470.2999 0.380917282  0.34054232 50    14
# m15  6 -218.7199 -449.4398 -460.9119 0.525511081  0.48333429 50    15
nmod <- nrow(summary(all_mod))
```

Run the code below to produce a panel of visualisations of the model performance measures for all fitted models (using `fitall` from the `meifly` package). In each panel, a single point is shown indicating the

relevant performance measure for each model. While not explicitly reporting the model with the highest measure, the plots shown do give information about the similarity of these values across the ensemble of models, and give an impression of the improvements that can be achieved through the inclusion (or removal) of an additional regressor.

```
all_mod_s <- all_mod %>%
    map_df(glance) %>%
    mutate(model = nmod) %>%
    mutate(negBIC = -1 * BIC, negAIC = -1 * AIC)

label <- NULL
for (i in nmod) {
    l <- as.character(summary(all_mod[[i]])$call)[2]
    label <- c(label, substr(l, 5, str_length(l)))
}

all_mod_s_long <- all_mod_s %>%
    gather(fit_stat, val, adj.r.squared, negAIC, negBIC, logLik,
        r.squared) %>%
    group_by(fit_stat, df) %>%
    mutate(rank = min_rank(desc(val)))

p1 <- ggplot(all_mod_s_long, aes(df, val)) + geom_point() + geom_line(data = filter(all_mod_s_long,
    rank == 1)) + facet_wrap(~fit_stat, ncol = 5, scales = "free_y") +
    xlab("Number of regressors (excluding intercept)") + ylab("Values") +
    theme_bw(base_size = 10)
p1
```



Q13. Given the information in the panel of plots above, is there one single model that appears "best"

according to all five performance measures?

Answer Q13. First, we need to ignore the r.squared and logLik graphs. We cannot use these to select a model.
No, there is not a single "best" model. The best BIC measure with three regressors is higher than the BIC measure for Modified Model 1 (with 4 regressors). For all other criterion, specifically adj-Rsquared and AIC, Modified Model 1 is reported to be "best".

Q14. Why do we consider "Negative" BIC (NegBIC) and "Negative" AIC (NegAIC), rather than BIC and AIC, respectively?

Answer Q14. So that the measures are all positively oriented.

Next, run the code chunk below to extract the model associated with the maximum value of the Adjusted R-squared, Negative AIC and Negative BIC measures.

```
print("Adjusted R-squared")
# [1] "Adjusted R-squared"
indexadjRsq <- c(1:nmod)[all_mod_s$adj.r.squared == max(all_mod_s$adj.r.squared)]
indexadjRsq
# [1] 15
max_adjRsq <- all_mod[[indexadjRsq]]
max_adjRsq
#
# Call:
# lm(formula = y ~ x1 + x2 + x3 + x4, data = data, model = FALSE)
#
# Coefficients:
# (Intercept)           x1           x2           x3           x4
#     -3.4686       4.0318       3.7214       0.7529       7.8087

print("Negative AIC")
# [1] "Negative AIC"
indexAIC <- c(1:nmod)[all_mod_s$negAIC == max(all_mod_s$negAIC)]
indexAIC
# [1] 15
max_AIC <- all_mod[[indexAIC]]
max_AIC
#
# Call:
# lm(formula = y ~ x1 + x2 + x3 + x4, data = data, model = FALSE)
#
# Coefficients:
# (Intercept)           x1           x2           x3           x4
```

```
#     -3.4686       4.0318       3.7214       0.7529       7.8087

print("Negative BIC")
# [1] "Negative BIC"
indexBIC <- c(1:nmod)[all_mod_s$negBIC == max(all_mod_s$negBIC)]
indexBIC
# [1] 13
max_BIC <- all_mod[[indexBIC]]
max_BIC
#
# Call:
# lm(formula = y ~ x1 + x3 + x4, data = data, model = FALSE)
#
# Coefficients:
# (Intercept)           x1           x3           x4
#      1.7089       3.9278       0.6139       7.5324
```

Obtain the full summary, tidy and glance output from OLS fit of the best BIC model, which we'll denote by "MM1BIC".

```
MM1BIC <- lm(formula = y ~ x1 + x3 + x4, data = simd)
summary(MM1BIC)
#
# Call:
# lm(formula = y ~ x1 + x3 + x4, data = simd)
#
# Residuals:
#    Min     1Q Median     3Q    Max
# -44.02 -12.43  -1.60  14.27  44.77
#
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)   1.7089    12.5211   0.136  0.89203
# x1            3.9278     1.1015   3.566  0.00086 ***
# x3            0.6139     0.2651   2.315  0.02511 *
# x4            7.5324     1.5168   4.966 9.85e-06 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 20.53 on 46 degrees of freedom
# Multiple R-squared:  0.5015,  Adjusted R-squared:  0.469
# F-statistic: 15.43 on 3 and 46 DF,  p-value: 4.413e-07
tidy(MM1BIC)
# # A tibble: 4 x 5
#   term        estimate std.error statistic    p.value
#   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
# 1 (Intercept)    1.71     12.5       0.136 0.892
# 2 x1             3.93      1.10      3.57  0.000860
# 3 x3             0.614     0.265     2.32  0.0251
# 4 x4             7.53      1.52      4.97  0.00000985
glance(MM1BIC)
# # A tibble: 1 x 12
#   r.squared adj.r.squared sigma statistic     p.value    df logLik   AIC   BIC
```

```
#       <dbl>         <dbl> <dbl>     <dbl>       <dbl> <dbl>  <dbl> <dbl> <dbl>
# 1     0.502         0.469  20.5      15.4 0.000000441     3  -220.  450.  459.
# # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Q15. Compare these with the corresponding output from MM1, which is the output for the best AIC and Adj-Rsquared model. Do you see any major differences between MM1 and MM1BIC when comparing the model fit summaries?

Code chunk below not shown to students

```
tidy.MM1
# # A tibble: 5 x 5
#   term        estimate std.error statistic    p.value
#   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
# 1 (Intercept)   -3.47     12.8      -0.271 0.788
# 2 x1             4.03      1.09      3.70  0.000579
# 3 x2             3.72      2.47      1.51  0.139
# 4 x3             0.753     0.277     2.71  0.00937
# 5 x4             7.81      1.51      5.18  0.00000504
glance.MM1
# # A tibble: 1 x 12
#   r.squared adj.r.squared sigma statistic     p.value    df logLik   AIC   BIC
#       <dbl>         <dbl> <dbl>     <dbl>       <dbl> <dbl>  <dbl> <dbl> <dbl>
# 1     0.526         0.483  20.2      12.5 0.000000665     4  -219.  449.  461.
# # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

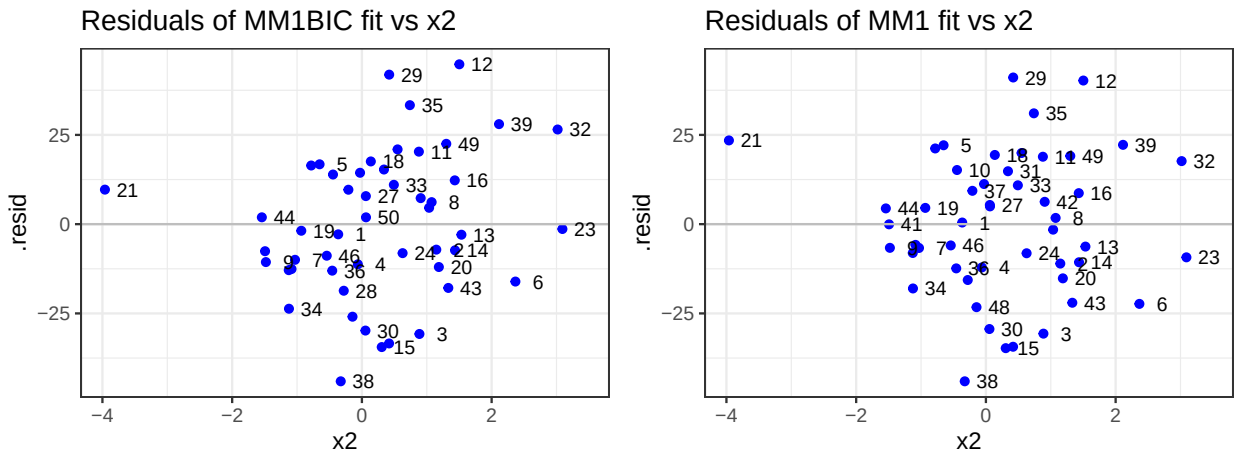
Answer Q15. Need to view the tidy and glance output from M1 to compare, so have printed again here.

It seems the coefficient on x_2 is not significant in the MM1 model - perhaps this would justify excluding this regressor and using the MM1BIC model. But we will do a little more checking below.

One additional point we could check, before making the ultimate conclusion, is to see if the residuals corresponding to the points with the large .hat or .cooksd values changed much from the MM1BIC fit to the MM1 fit. For example, if they change (for the better) this would give some justification for including the x_2 regressor. This is done in the code chunk below.

```
aug.MM1BIC <- augment(MM1BIC)
aug.MM1BIC <- aug.MM1BIC %>%
    mutate(x2 = simd$x2)

yr <- range(aug.MM1BIC$.resid, aug.MM1$.resid)

px2.MM1BIC <- aug.MM1BIC %>%
    ggplot(aes(x = x2, y = .resid)) + geom_point(colour = "blue") +
    geom_hline(yintercept = 0, colour = "grey") + ggtitle("Residuals of MM1BIC fit vs x2") +
    theme(plot.title = element_text(size = 8)) + theme_bw() +
    ylim(yr) + geom_text(label = simd$rowid, nudge_x = 0.35,
    colour = "black", size = 3, check_overlap = T)
```

```
px2.MM1 <- aug.MM1 %>%
    ggplot(aes(x = x2, y = .resid)) + geom_point(colour = "blue") +
    geom_hline(yintercept = 0, colour = "grey") + ggtitle("Residuals of MM1 fit vs x2") +
    theme(plot.title = element_text(size = 8)) + theme_bw() +
    ylim(yr) + geom_text(label = simd$rowid, nudge_x = 0.35,
    colour = "black", size = 3, check_overlap = T)

grid.arrange(px2.MM1BIC, px2.MM1, ncol = 2)
```



Q16. In which row(s) of *simd* (corresponding to the identifier *rowid* on residual plots above) are there observations that are better "explained" by the MM1 model than the MM1BIC model?

Answer Q16. We might look first to see if there are some changes to residuals relating to those observations with large .hat values, rowid= 16, 21 and 44. Surprisingly the residual for rowid 21 gets bigger with the inclusion of x_2. The other two residuals corresponding to rowid=16, or 44 moved slightly but not by very much. At least judging visually, this doesn't seem to be a driving reason for the increase in NegAIC.

Q17. Based on the information available, which model would you ultimately choose, and why?

Answer Q17. Personally, I would choose BIC. The extra regressor doesn't seem to do very much.

But it is quite interesting here - even when we know the truth (because we simulated it!) we end up choosing a different model. First of all, we had to drop one of the highly correlated regressors (x_5) so already we don't end up with the original model. But even after that, we end up with an insignificant intercept term and some doubt over whether or not to include x_2.

Notice that the error standard deviation parameter, σ was quite large when we simulated the data. This explains why it was hard to estimate the intercept term - the noise was larger compared to the "signal". If you were to repeat this exercise with a smaller value of sigma, or if a larger sample size n were used, we might have been able to recover the original model (except without x_5.)

It would be interesting to re-do this exercise with a different value of the "true" σ^2, or with a much larger sample size (though the plots may be more difficult to see with a bigger n).

Q18. Compare the LOOCV values for MM1 and MM1BIC- does this support your answer to Q17?

Answer Q18.

```
# not given to students

loocv1 <- mean((aug.MM1$.resid/(1 - aug.MM1$.hat))^2)
loocvbic <- mean((aug.MM1BIC$.resid/(1 - aug.MM1BIC$.hat))^2)
```

The LOOCV value for the MM1 model is lower than the BIC model, so this suggests that MM1 is preferred, contrary to my answer in Q17

Additional thoughts
Here we have selected just based upon fit. We should also check that our residuals have desirable properties, and perhaps double check that the leverage and Cook's D are still OK (in this example they will be, but in practice, they may not).

We saw that the LOOCV suggested a different model. So model selection is not always clear.

If we find other problems with the "best" model, we may need to start over and think of new variables, data transformations, different model specifications and so on.

We would also need to keep in mind the purpose of our analysis when deciding what to do.
