PG06_A3

Lachlan Moody ID27809951

Abhishek Sinha ID31322743

Sen Wang ID30382106

Yiwen Zhang ID31203019

04/11/2020

```
knitr::opts_chunk$set(echo = TRUE, eval=TRUE,   warning = FALSE, message = FALSE,
                      error = FALSE, tidy.opts = list(width.cutoff=60), tidy=TRUE, fig.align = 'center')
options(digits = 3)
```

```
library(tidyverse)
library(bayess)
library(broom)
library(car)
library(GGally)
library(meifly)
library(patchwork)
library(kableExtra)
library(boot)
```

```
data(caterpillar)
cat <- as_tibble(caterpillar)
data_desc <- tibble("variable" = c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "y"),
       "description" = c("altitude (in meters)", "slope (in degrees)", "number of pine trees in the area", "hei
ght (in meters) of the tree sampled at the center of the area", "orientation of the area (from 1 if southbound
 to 2 otherwise)", "height (in meters) of the dominant tree", "number of vegetation strata", "mix settlement in
dex (from 1 if not mixed to 2 if mixed)", "logarithmic transform of the average number of nests of caterpillars
per tree"))
```

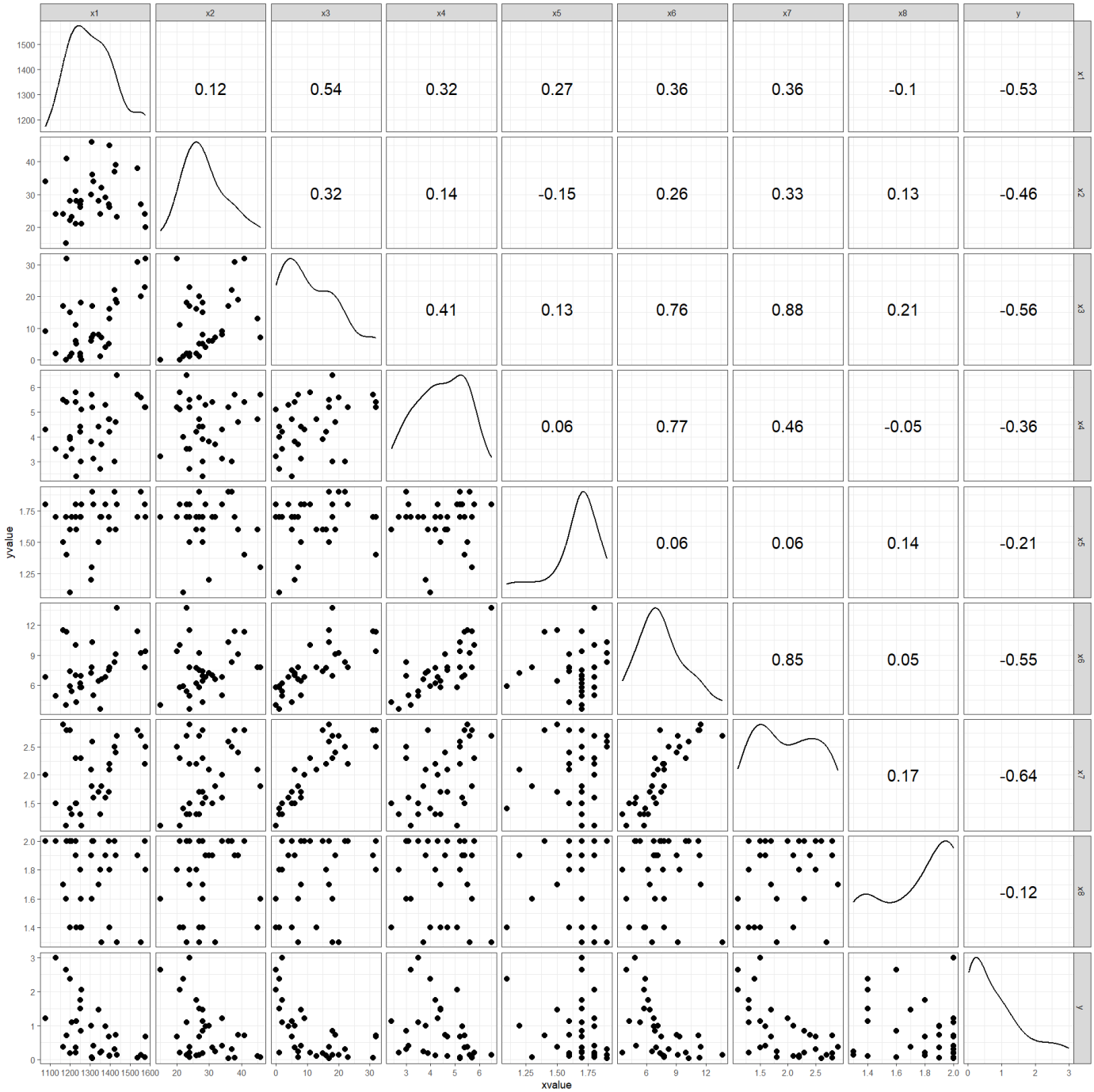
Part B: Multiple Linear Regression

```
data_desc %>%
  kable(caption = "Description of variables in `caterpillar` dataset") %>%
  kable_styling(bootstrap_options = c("striped", "border"))
```

Description of variables in `caterpillar` dataset

variable	description
x1	altitude (in meters)
x2	slope (in degrees)
x3	number of pine trees in the area
x4	height (in meters) of the tree sampled at the center of the area
x5	orientation of the area (from 1 if southbound to 2 otherwise)
x6	height (in meters) of the dominant tree
x7	number of vegetation strata
x8	mix settlement index (from 1 if not mixed to 2 if mixed)
y	logarithmic transform of the average number of nests of caterpillars per tree

```
ggscatmat(cat, columns=c(1:9)) +
  theme(geom.text.size = 7, strip.text.x = element_text(size = 12)) +
  theme_bw(base_size = 7)
```



Q10. [10 marks]

The *ggscatmat()* function used above comes from the **GGally** package and is used to generate a scatterplot matrix for the caterpillar data set. This data is based on a 1973 study of pine processionary caterpillars and contains a response variable y which relates to the log transform of the number of nests per unit, and 8 potential explanatory variables.

This matrix is divided into three distinct parts and from this we can gather information about the complete set of regressors (x1 through x8) and the response variable y.

A description of the data from the helpfile and also been included in the table above to provide context for these observations.

i) The lower triangle of the table provides a scatter plot for each variable plotted against all others. The variables in the top margin are plotted on the x-axis and those on the left are plotted on the y-axis. This can used to quickly visualise the relationship between any two variables in the data set and see if any are suitable for further analysis.

Looking at this for the caterpillar data, while there doesn't appear to be much association for the first two variables, x3 shows somewhat of an a relationship with x6 and a stronger one with x7. This makes sense considering considering x3 relates to the number of pine trees in the area being analysed. As this increases it is probable that both the height of the dominant tree in the area (x6) also increases as there is a larger sample, also the amount of vegetation strata (x7) would also increase as a higher number of trees may suggest an area is more fertile and suitable for growing more plants.

Moving to the right, x4 (height of the centre tree in the area) has a similarly strong relationship with x6 (height of the tallest tree in the area). Again, this makes sense to be highly associated as areas that can produce taller trees are also more likely to be able to produce tall trees in general, thus increasing the chance that the tree in the centre would be taller.

Next across, x6 (height of the tallest tree) and x7 (number of vegetation strata) also display a high degree of association which is reasonable given that an area that produces overly tall trees should have high quality soil that increases the amount of vegetation strata in the area.

Finally, along the bottom row, we see what appears to be a negative association for all the x variables with y along the logarithmic scale that it has been transformed by.

ii) The top right triangle in this matrix display the Pearson correlation coefficient for the relationships between different variables. First examining just the different x variables, almost all are correlated positively, indicating that as one increases so does the others. The only ones that do not are x5 with x2, x8 with x1, and x8 with x4. However, these values are all small and do not indicate any strong linear relationship.

Looking at the realtionships discussed previously, the correlation for x3 (number of pine trees) with x6 (height of the tallest tree) and x7 (number of vegetation strata) are 0.76 and 0.88 respectively. This supports that these two values are highly positively correlated. Similar results are seen for x4 (height of the centre tree in the area) and x6 (height of the tallest tree in the area) with a correlation of 0.77 and also for x6 (height of the tallest tree) and x7 (number of vegetation strata) with a correlation of 0.85. This provides evidence that there may be some collinearity between the variables discussed.

Interestingly, all x variables share a negative correlation with the y response variable, with x7(number of vegetation strata) being the strongest. With y relating to the number of caterpillar nest per tree it makes sense to be negatively correlated with x3 (the number of trees) as this gives the caterpillars area to spread out, which is supported by it having the second strongest correlation at -0.56. At discussed previously, this variable is strongly positively correlated with x6 and x7 making it reasonable for them to also have strong negative correlations with y. Also it may be likely that a high altitude (measured by x1) and a steep slope (measured by x2), are not ideal caterpillar habitats and may cause the somewhat strong negative correlation observed.

iii) Finally, the main diagonal displays denisty plots for each variable. Variables x1, x2, and x6 are all approximately normal with varying amounts of positive skew while x4 and x5 are also normal they are instead negatively skewed. Meanwhile x7 and x8 display multi-modality which is reasonable considering there is a fixed number of vegetable strata, and the mix settlement index has a binary outcome of only 1 or 2. Meanwhile examining the response variable, y, it appears that many of the observed values were quite low with quite a long tail to the right. This suggests that most habitats support a small number of caterpillar nests.

Q11. [5 marks]

```
modelf <- lm(data = cat, formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8)
car::vif(modelf)
```

```
##    x1    x2    x3    x4    x5    x6    x7    x8
##  1.85  1.18  6.15  4.09  1.17 11.80  9.09  1.22
```

Q12. [10 marks]

The VIF, or variable inflation factor, measures the degree of collinearity between the explanatory variables within a model. This is similar to the visual inference conducted above in the scatter plot matrix but instead provides an automated way to make this comparison. This involves calculating the r-squared value for each variable on all other variables. In general, values for VIF greater than 10 are considered to high.

This can be used to identify redundant variables where the information is already provided by other variables in the model. It is desirable to identify and remove such regressors as if there is multicollinearity, there tends to be high standard errors on the coefficients for that regressor - that is the weight applied to each of the associated regressors becomes arbitrary.

Looking at the output for modelf above, variable x6 is the only one with a VIF above the cuttoff value of 10. This variable relates to the height (in meters) of the dominant tree, and as discussed previously, is highly positively correlated with x3, x4, and x7 - or the number of pine trees, height of the centre tree in the area, and number of vegetation strata respectively. As mentioned these associations are reasonable given that more trees (x3) would increase the sample from which the highest tree can be drawn from, a higher center tree would indicate there are taller trees in the area, and an increased about of vegetation would indicate an

environment condusive to growing large trees. As such the predictive power of x6 is already captured by these variables and the rest of the data. That means, according to the VIF, this variable should be removed from consideration. This is also desirable as it reduced the total number of regressors for the model without losing its predictive power, leading to the best simplest model.

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

all_mod <- quiet(fitall(y=cat$y,x=cat[,-c(6,9)], method="lm"))
summary(all_mod)
```

```
##      df  logL   AIC   BIC     R2    adjR2  n model
## m1    3 -33.8 -73.5 -78.0 0.2811  0.25794 33     1
## m2    3 -35.4 -76.7 -81.2 0.2074  0.18188 33     2
## m3    4 -29.7 -67.5 -73.5 0.4367  0.39915 33     3
## m4    3 -32.9 -71.8 -76.3 0.3180  0.29598 33     4
## m5    4 -31.0 -70.1 -76.1 0.3905  0.34988 33     5
## m6    4 -30.7 -69.5 -75.5 0.4017  0.36180 33     6
## m7    5 -28.3 -66.5 -74.0 0.4852  0.43194 33     7
## m8    3 -36.9 -79.9 -84.4 0.1281  0.09997 33     8
## m9    4 -32.8 -73.7 -79.7 0.3204  0.27509 33     9
## m10   4 -33.4 -74.8 -80.8 0.2965  0.24961 33    10
## m11   5 -29.0 -68.0 -75.5 0.4616  0.40595 33    11
## m12   4 -32.4 -72.9 -78.9 0.3366  0.29238 33    12
## m13   5 -30.8 -71.5 -79.0 0.4011  0.33913 33    13
## m14   5 -30.2 -70.4 -77.9 0.4200  0.36002 33    14
## m15   6 -27.9 -67.9 -76.9 0.4950  0.42286 33    15
## m16   3 -38.5 -82.9 -87.4 0.0448  0.01403 33    16
## m17   4 -33.6 -75.3 -81.3 0.2863  0.23874 33    17
## m18   4 -33.6 -75.2 -81.2 0.2883  0.24088 33    18
## m19   5 -29.1 -68.1 -75.6 0.4595  0.40360 33    19
## m20   4 -32.4 -72.8 -78.8 0.3377  0.29356 33    20
## m21   5 -30.9 -71.7 -79.2 0.3967  0.33426 33    21
## m22   5 -29.5 -69.1 -76.6 0.4435  0.38595 33    22
## m23   6 -27.6 -67.2 -76.2 0.5053  0.43462 33    23
## m24   4 -36.2 -80.5 -86.5 0.1647  0.10899 33    24
## m25   5 -32.7 -75.4 -82.9 0.3265  0.25684 33    25
## m26   5 -31.7 -73.4 -80.9 0.3647  0.29902 33    26
## m27   6 -28.3 -68.5 -77.5 0.4850  0.41140 33    27
## m28   5 -31.9 -73.9 -81.4 0.3561  0.28953 33    28
## m29   6 -30.6 -73.1 -82.1 0.4077  0.32307 33    29
## m30   6 -29.0 -70.0 -79.0 0.4615  0.38457 33    30
## m31   7 -27.2 -68.5 -79.0 0.5158  0.42613 33    31
## m32   3 -30.7 -67.3 -71.8 0.4043  0.38512 33    32
## m33   4 -27.5 -63.1 -69.0 0.5073  0.47448 33    33
## m34   4 -28.6 -65.3 -71.2 0.4733  0.43823 33    34
## m35   5 -25.1 -60.1 -67.6 0.5759  0.53207 33    35
## m36   4 -30.7 -69.3 -75.3 0.4045  0.36481 33    36
## m37   5 -26.6 -63.2 -70.7 0.5343  0.48614 33    37
## m38   5 -28.6 -67.3 -74.7 0.4734  0.41893 33    38
## m39   6 -23.6 -59.2 -68.2 0.6116  0.55607 33    39
## m40   4 -30.5 -69.0 -75.0 0.4098  0.37045 33    40
## m41   5 -27.5 -65.0 -72.5 0.5075  0.45659 33    41
## m42   5 -28.4 -66.9 -74.4 0.4794  0.42557 33    42
## m43   6 -25.0 -62.1 -71.1 0.5763  0.51575 33    43
## m44   5 -30.5 -71.0 -78.5 0.4099  0.34888 33    44
## m45   6 -26.6 -65.2 -74.2 0.5343  0.46779 33    45
## m46   6 -28.4 -68.9 -77.8 0.4795  0.40517 33    46
## m47   7 -23.6 -61.2 -71.7 0.6116  0.53966 33    47
## m48   4 -29.8 -67.6 -73.6 0.4339  0.39613 33    48
## m49   5 -27.3 -64.5 -72.0 0.5155  0.46536 33    49
## m50   5 -27.0 -64.0 -71.4 0.5234  0.47411 33    50
## m51   6 -24.2 -60.5 -69.5 0.5963  0.53868 33    51
## m52   5 -29.8 -69.6 -77.1 0.4340  0.37550 33    52
## m53   6 -26.3 -64.6 -73.5 0.5433  0.47808 33    53
## m54   6 -26.9 -65.8 -74.8 0.5256  0.45781 33    54
## m55   7 -22.6 -59.2 -69.7 0.6346  0.56690 33    55
## m56   5 -29.7 -69.4 -76.9 0.4385  0.38046 33    56
## m57   6 -27.2 -66.5 -75.5 0.5157  0.44656 33    57
## m58   6 -26.8 -65.6 -74.6 0.5285  0.46113 33    58
## m59   7 -24.2 -62.4 -72.9 0.5968  0.52215 33    59
## m60   6 -29.7 -71.4 -80.3 0.4388  0.35858 33    60
## m61   7 -26.3 -66.6 -77.0 0.5433  0.45877 33    61
## m62   7 -26.7 -67.4 -77.9 0.5308  0.44393 33    62
## m63   8 -22.6 -61.2 -73.2 0.6346  0.55031 33    63
## m64   3 -39.0 -83.9 -88.4 0.0153 -0.01642 33    64
## m65   4 -33.0 -74.0 -80.0 0.3127  0.26690 33    65
## m66   4 -35.3 -78.6 -84.6 0.2117  0.15918 33    66
## m67   5 -29.3 -68.6 -76.1 0.4517  0.39500 33    67
```

```
## m68   4 -32.9 -73.8 -79.8 0.3180  0.27258 33    68
## m69   5 -30.9 -71.7 -79.2 0.3968  0.33438 33    69
## m70   5 -30.7 -71.5 -78.9 0.4018  0.33995 33    70
## m71   6 -28.1 -68.3 -77.2 0.4892  0.41618 33    71
## m72   4 -36.6 -81.1 -87.1 0.1478  0.09095 33    72
## m73   5 -32.0 -74.1 -81.5 0.3530  0.28607 33    73
## m74   5 -33.2 -76.5 -84.0 0.3037  0.23168 33    74
## m75   6 -28.5 -69.0 -78.0 0.4778  0.40324 33    75
## m76   5 -32.4 -74.8 -82.3 0.3374  0.26885 33    76
## m77   6 -30.5 -73.0 -82.0 0.4096  0.32522 33    77
## m78   6 -30.2 -72.4 -81.4 0.4201  0.33724 33    78
## m79   7 -27.7 -69.5 -80.0 0.5007  0.40822 33    79
## m80   4 -38.3 -84.6 -90.6 0.0540 -0.00911 33    80
## m81   5 -33.0 -76.0 -83.4 0.3145  0.24357 33    81
## m82   5 -33.6 -77.2 -84.7 0.2887  0.21514 33    82
## m83   6 -28.8 -69.6 -78.6 0.4682  0.39228 33    83
## m84   5 -32.4 -74.8 -82.3 0.3378  0.26928 33    84
## m85   6 -30.8 -73.5 -82.5 0.4010  0.31541 33    85
## m86   6 -29.5 -71.0 -80.0 0.4451  0.36578 33    86
## m87   7 -27.6 -69.1 -79.6 0.5065  0.41513 33    87
## m88   5 -36.0 -82.0 -89.4 0.1778  0.09271 33    88
## m89   6 -32.0 -75.9 -84.9 0.3553  0.26319 33    89
## m90   6 -31.7 -75.4 -84.3 0.3665  0.27597 33    90
## m91   7 -28.0 -69.9 -80.4 0.4946  0.40097 33    91
## m92   6 -31.9 -75.9 -84.9 0.3563  0.26431 33    92
## m93   7 -30.4 -74.8 -85.3 0.4138  0.30527 33    93
## m94   7 -29.0 -72.0 -82.4 0.4619  0.36222 33    94
## m95   8 -27.2 -70.3 -82.3 0.5180  0.40682 33    95
## m96   4 -30.7 -69.3 -75.3 0.4045  0.36480 33    96
## m97   5 -27.4 -64.7 -72.2 0.5124  0.46198 33    97
## m98   5 -28.6 -67.3 -74.7 0.4734  0.41892 33    98
## m99   6 -25.0 -61.9 -70.9 0.5785  0.51829 33    99
## m100  5 -30.7 -71.3 -78.8 0.4046  0.34306 33   100
## m101  6 -26.1 -64.3 -73.3 0.5470  0.48227 33   101
## m102  6 -28.6 -69.3 -78.2 0.4734  0.39822 33   102
## m103  7 -23.2 -60.4 -70.9 0.6208  0.55054 33   103
## m104  5 -30.5 -71.0 -78.5 0.4104  0.34938 33   104
## m105  6 -27.3 -66.7 -75.7 0.5129  0.44336 33   105
## m106  6 -28.4 -68.9 -77.9 0.4794  0.40507 33   106
## m107  7 -24.9 -63.9 -74.3 0.5791  0.50117 33   107
## m108  6 -30.5 -73.0 -82.0 0.4104  0.32622 33   108
## m109  7 -26.1 -66.3 -76.8 0.5472  0.46332 33   109
## m110  7 -28.4 -70.9 -81.3 0.4795  0.38317 33   110
## m111  8 -23.2 -62.4 -74.4 0.6210  0.53351 33   111
## m112  5 -29.8 -69.6 -77.1 0.4340  0.37540 33   112
## m113  6 -27.1 -66.3 -75.3 0.5186  0.44979 33   113
## m114  6 -26.9 -65.8 -74.8 0.5251  0.45722 33   114
## m115  7 -24.2 -62.4 -72.9 0.5968  0.52216 33   115
## m116  6 -29.8 -71.6 -80.6 0.4341  0.35327 33   116
## m117  7 -25.9 -65.9 -76.3 0.5527  0.46984 33   117
## m118  7 -26.9 -67.7 -78.2 0.5269  0.43934 33   118
## m119  8 -22.4 -60.8 -72.8 0.6390  0.55575 33   119
## m120  6 -29.7 -71.4 -80.3 0.4385  0.35834 33   120
## m121  7 -27.1 -68.3 -78.7 0.5191  0.43002 33   121
## m122  7 -26.8 -67.5 -78.0 0.5294  0.44226 33   122
## m123  8 -24.2 -64.4 -76.4 0.5974  0.50451 33   123
## m124  7 -29.7 -73.4 -83.8 0.4388  0.33483 33   124
## m125  8 -25.9 -67.9 -79.8 0.5529  0.44967 33   125
## m126  8 -26.7 -69.4 -81.4 0.5315  0.42339 33   126
## m127  9 -22.4 -62.8 -76.2 0.6392  0.53823 33   127
```

Q13. [5 marks]

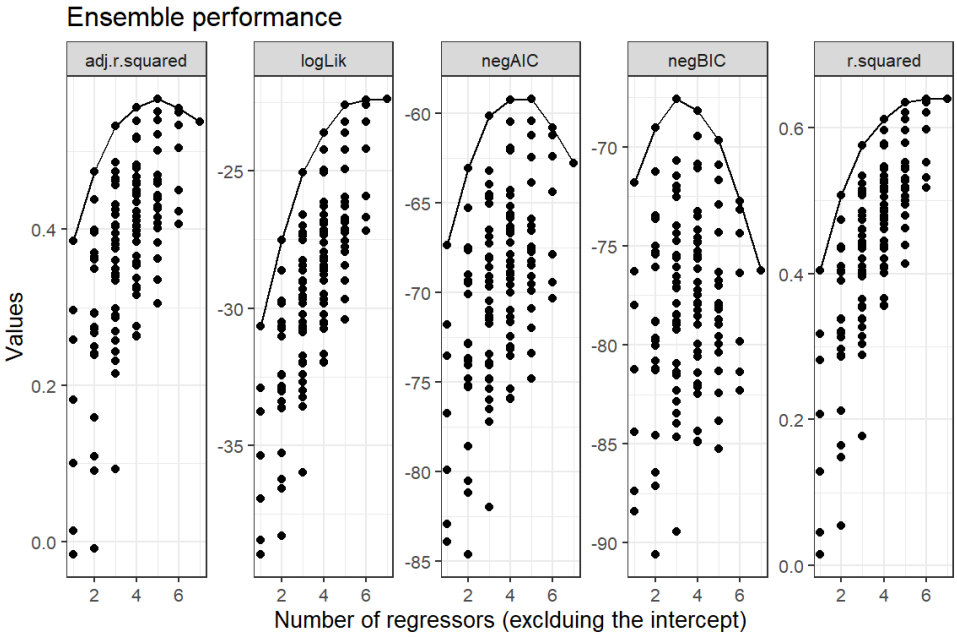
```
nrow(summary(all_mod))
```

```
## [1] 127
```

```
nmod <- nrow(summary(all_mod))
```

The *all_mod* object contains an ensemble of 127 models. This number was caluclated by counting the number of rows within the data summary. This can also be confirmed by examing the model column in the data summary which has the last model as 127.

```
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 (exclduing the intercept)") +
  ylab("Values") +
  theme_bw(base_size = 10)

p1 +
  ggtitle("Ensemble performance")
```




```
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] 55
```

```
max_adjRsq <- all_mod[[indexadjRsq]]
max_adjRsq
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x5 + x7, data = data, model = FALSE)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x5           x7
##      8.1696      -0.0026      -0.0361       0.0396      -0.6923      -1.1078
```

```
print("log-Likelihood")
```

```
## [1] "log-Likelihood"
```

```
indexlogLik<-c(1:nmod)[all_mod_s$logLik==max(all_mod_s$logLik)]
indexlogLik
```

```
## [1] 127
```

```
max_logLik <- all_mod[[indexlogLik]]
max_logLik
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x7 + x8, data = data,
##     model = FALSE)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4           x5
##     8.68330     -0.00275     -0.03530      0.04252     -0.01268     -0.62963
##          x7           x8
##    -1.11429     -0.23308
```

```
print("Negative AIC")
```

```
## [1] "Negative AIC"
```

```
indexAIC<-c(1:nmod)[all_mod_s$negAIC==max(all_mod_s$negAIC)]
indexAIC
```

```
## [1] 55
```

```
max_AIC <- all_mod[[indexAIC]]
max_AIC
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x5 + x7, data = data, model = FALSE)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x5           x7
##      8.1696      -0.0026      -0.0361       0.0396      -0.6923      -1.1078
```

```
print("Negative BIC")
```

```
## [1] "Negative BIC"
```

```
indexBIC<-c(1:nmod)[all_mod_s$negBIC==max(all_mod_s$negBIC)]
indexBIC
```

```
## [1] 35
```

```
max_BIC <- all_mod[[indexBIC]]
max_BIC
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x7, data = data, model = FALSE)
##
## Coefficients:
## (Intercept)           x1           x2           x7
##     5.71117     -0.00215     -0.03058     -0.59857
```

```
print("R-squared")
```

```
## [1] "R-squared"
```

```
indexRsq<-c(1:nmod)[all_mod_s$r.squared==max(all_mod_s$r.squared)]
indexRsq
```

```
## [1] 127
```

```
max_Rsq <- all_mod[[indexRsq]]
max_Rsq
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x7 + x8, data = data,
##     model = FALSE)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4           x5
##     8.68330     -0.00275     -0.03530      0.04252     -0.01268     -0.62963
##          x7           x8
##    -1.11429     -0.23308
```

Q14. [20 marks]

Based on the ensemble of models produced, five potentially “best” models are selected based on those with the highest adjusted r-squared, log-likelihood, negative AIC, negative BIC and r-squared values. However, as r-squared and log-likelihood will never decrease when new regressors are added they should not be used to compare models with different numbers of regressors as in this scenario. The other three measures, adjusted r-squared, negative AIC and negative BIC introduce penalty terms for the number of regressors thus making them more useful for comparison in this scenario.

Based on this, there are only two models for consideration. Model 55 as the one with highest adjusted r-squared and negative AIC and model 35 as the one with the highest BIC. The formula for each of these models has been saved as `mod55` and `mod35` below. Additionally, a summary of each models performance has been output. Note that the table shows the AIC and BIC values rather than their negatives. So while model 35’s BIC is lower, between these two it is better considering the aim is to maximise the negative value.

```
mod55 <- lm(formula = y ~ x1 + x2 + x3 + x5 + x7, data = cat)
mod35 <- lm(formula = y ~ x1 + x2 + x7, data = cat)

bind_rows(glance(mod55),
          glance(mod35), .id = "model") %>%
  mutate(model = c("model 55", "model 35")) %>%
  kable(caption = "Model summary statistics") %>%
  kable_styling(bootstrap_options = c("striped", "bordered", "hover"))
```

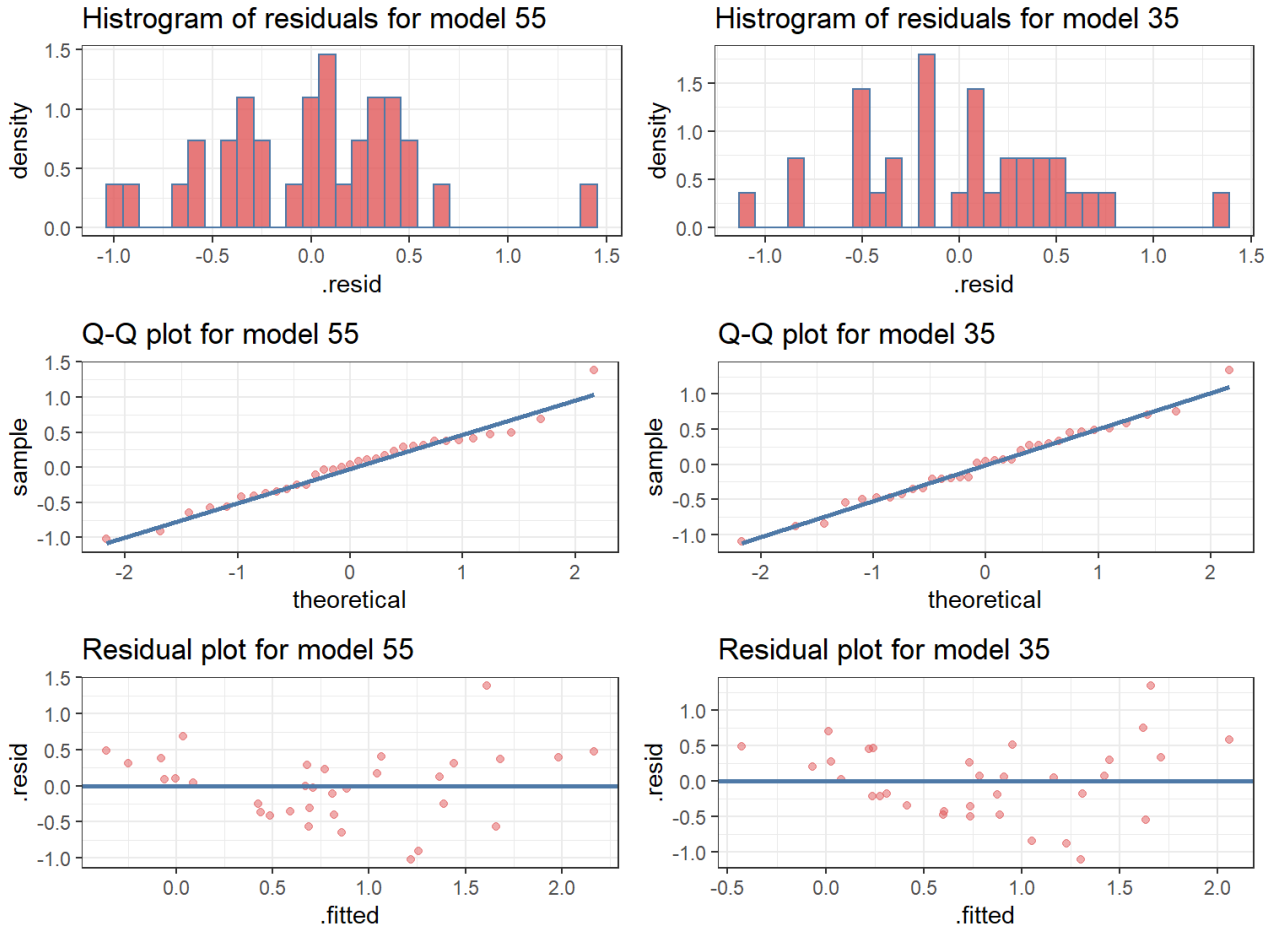
Model summary statistics

model	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
model 55	0.635	0.567	0.531	9.38	0	5	-22.6	59.2	69.7	7.60	27	33
model 35	0.576	0.532	0.552	13.13	0	3	-25.1	60.1	67.6	8.82	29	33

Firstly regarding the number of regressors in each model, looking back to the table titled *'Ensemble Performance'*, we can see that both adjusted r-squared and negative AIC both peak at 5 regressors each while negative BIC peaks at 3. This matches the form of model 55 and 35 respectively. The difference in model preference for the negative AIC and BIC values is not unexpected as BIC penalises model complexity more heavily than AIC or adjusted r-squared. Thus it produces a higher value for model 35 which only has three regressors plus an intercept over model 55 which has 5 regressors plus an intercept.

So as both models have some level of validity the next step is to examine the residual values of each model. This can be visualised in three ways - plotting a histogram of the residuals, producing a normal probability (or quantile-quantile) plot, and ploting a residuals versus fitted plot. This can be seen in the output below.

```
model_hist <- function(model, title){
  augment(model) %>%
    ggplot(aes(x= .resid, y = ..density..)) +
    geom_histogram(fill = "#e15759", colour = "#4e79a7", alpha = 0.8) +
    theme_bw() +
    ggtitle(paste("Histrogram of residuals for", title))
}

model_qq <- function(model, title){
    augment(model) %>%
      ggplot(aes(sample = .resid)) +
      geom_qq(color = "#e15759",
              alpha = 0.5) +
      geom_qq_line(color = "#4e79a7",
                   size = 1) +
      theme_bw() +
      ggtitle(paste("Q-Q plot for", title))
}

model_resid <- function(model, title){
    augment(model) %>%
      ggplot(aes(x = .fitted, y = .resid)) +
      geom_point(color = "#e15759",
              alpha = 0.5) +
      geom_hline(yintercept = 0,
                 color = "#4e79a7",
                 size = 1) +
      theme_bw() +
      ggtitle(paste("Residual plot for", title))
}

p1 <- model_hist(mod55, "model 55")
p2 <- model_qq(mod55, "model 55")
p3 <- model_resid(mod55, "model 55")

p4 <- model_hist(mod35, "model 35")
p5 <- model_qq(mod35, "model 35")
p6 <- model_resid(mod35, "model 35")

(p1 | p4) /
(p2 | p5) /
(p3 | p6)
```



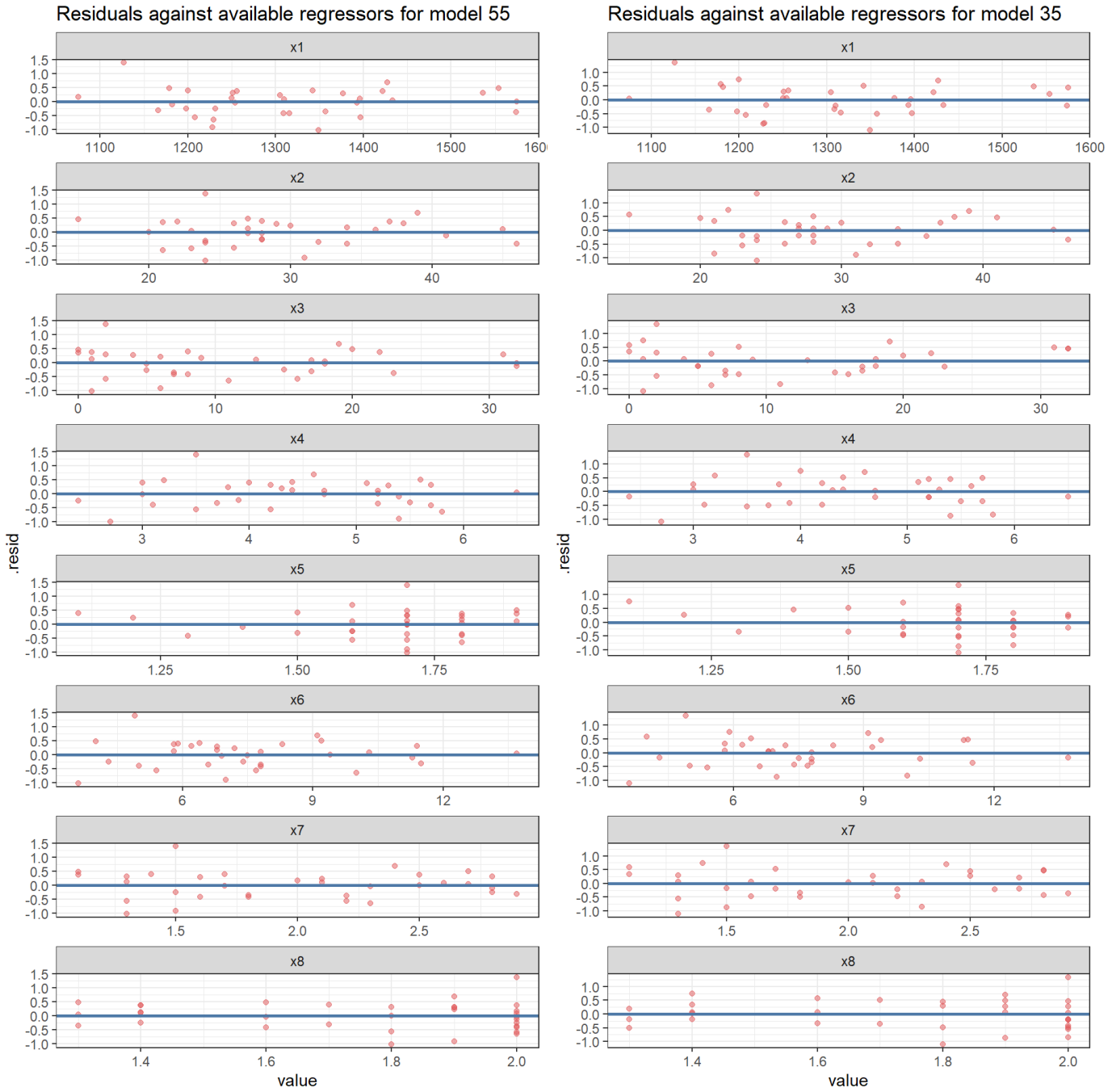
Starting from the top row, the residuals for model 55 appear to be more normally distributed than model 35 as it appears to have some slight positive skew in its distribution. This indicates that the normality assumption is more likely to be true for model 55 as the variance is normally distributed.

In the second row, the normal probability plot for each model is shown. This plots the quantiles sampled from the data against the standard normal distribution. If the points fall along the line, this indicates that the residuals come from a normal distribution. As both models fit this line quite well they both pass this check.

The third plot shows the residuals of each model against its fitted value. The plot is used to detect non-linearity, unequal error variances, and outliers which can be detected by the appearance of any type of pattern. For model 55 there appears to be no visible pattern in the data which strengthens the evidence for this models use. In comparison for model 35 there appears to be somewhat of an upward trend towards the end of the plot indicating that there still may be some relationship not captured by the model.

An additional method that can be used is to plot the residuals against all available regressors in the data, even those not included in either model. This plot is displayed below. A well-fitted model should likewise not show any obvious pattern. Comparing the models across rows, most of the plots look similar with random variation around 0 except for x2 under model 35. Towards the end of the data there is a clearly identifiable curve indicating that there is an underlying relationship that mass been missed. This provides more strength of evidence for model 55 being the preferred model.

```
residregress <- function(model, title){
cat %>% left_join(augment(model)) %>%
  pivot_longer(x1:x8) %>%
    ggplot(aes(x = value, y = .resid)) +
    geom_point(color = "#e15759",
            alpha = 0.5) +
    facet_wrap(~name, scales ="free", ncol = 1) +
    ggtitle(paste("Residuals against available regressors for", title)) +
    geom_hline(yintercept = 0,
                color = "#4e79a7",
                size = 1) +
    theme_bw()
}

residregress(mod55, "model 55") + residregress(mod35, "model 35")
```

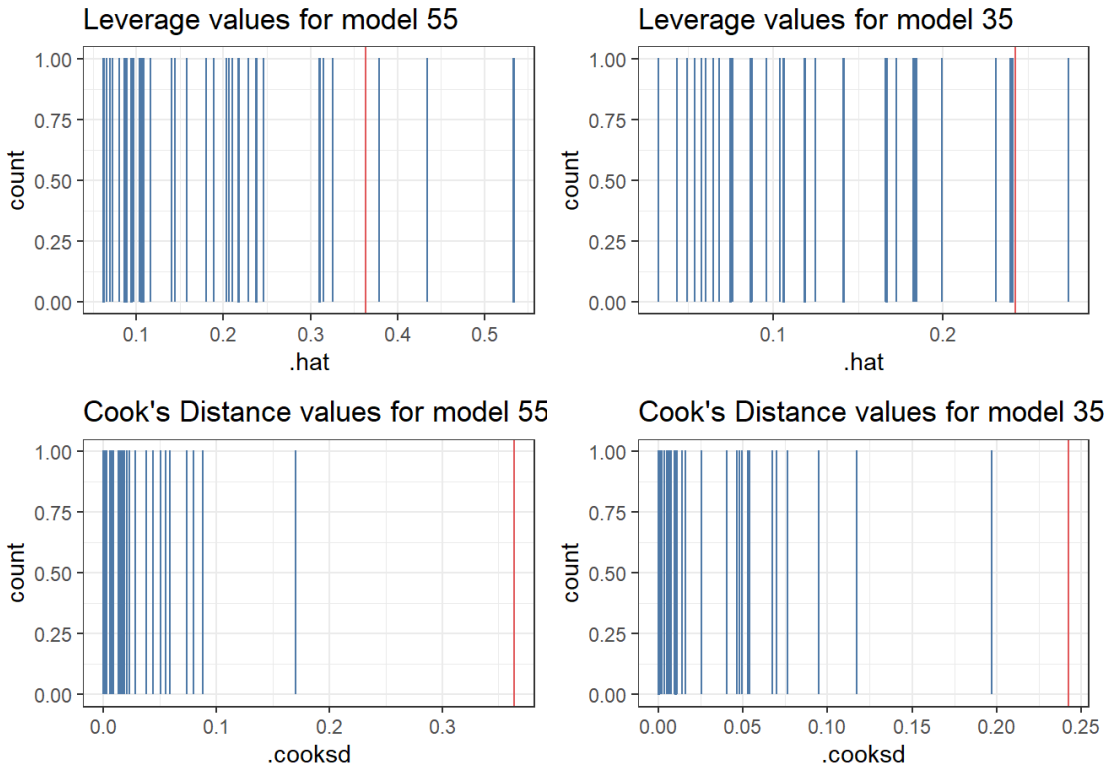


As a final check, the Leverage and Cook's Distance values for each model can be compared relative to their cuttoff value which is calculated as 2p/n where p is the number of regressors in the model and n in the number of observations. Rather than being a strict cuttoff value like VIF, here it is only values far away from the threshold that may be of concern.

These values deal with individual obsevations in different ways. Leverage is a measure of how distant an observed independent variable is from other observations of the same variable. Values with a high leverage has a greater influence on the fitted regression line. For model 55 there appears to be two observations with noticeably high leverage compared to one for model 35.

We can examine this further using Cook's Distance which measures the effect of deleting a given observation using a combination of each observations leverage and residual values. Here neither model displays any obersvations that require further analysis. As neither have values that fall outside this threshold it would appear both models don't have any overly influential observations.

```
n = nrow(cat)

p55 = nrow(tidy(mod55))
t55 = 2*p55/n

p35 = nrow(tidy(mod35))
t35 = 2*p35/n

leverage_plot <- function(model, t, title){
  augment(model) %>%
  ggplot(aes(x = .hat)) +
  geom_bar(colour = "#4e79a7", fill = "#4e79a7") +
  geom_vline(xintercept = t, colour = "#e15759") +
  theme_bw() +
  ggtitle(paste("Leverage values for", title))
}

cooks_plot <- function(model, t, title){
  augment(model) %>%
    ggplot(aes(x = .cooksd)) +
    geom_bar(colour = "#4e79a7", fill = "blue") +
    geom_vline(xintercept = t, colour = "#e15759") +
    theme_bw() +
    ggtitle(paste("Cook's Distance values for", title))
}

p7 <- leverage_plot(mod55, t55, "model 55")
p8 <- cooks_plot(mod55, t55, "model 55")

p9 <- leverage_plot(mod35, t35, "model 35")
p10 <- cooks_plot(mod35, t35, "model 35")


(p7 | p9) /
(p8 | p10)
```



Based on the output and discussion conducted above, model 55 has been chosen as the preferred model as it recorded higher values in regards to both adjusted r-squared and negative AIC, it's residual plots displayed less evidence of breaking the model assumptions, and it had no influential observations when using Cook's Distance and only one more than model 35 when using leverage.

The form of model 55 has been saved in the output below as modelp.

```
modelp <- lm(formula = y ~ x1 + x2 + x3 + x5 + x7, data = cat)
```

Q15. [5 marks]

```
modelp
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x5 + x7, data = cat)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x5           x7
##      8.1696      -0.0026      -0.0361       0.0396      -0.6923      -1.1078
```

Based on the above analysis, the preferred model is model 55 which is saved as modelp. The form of this model can be seen in the above output and can be written as:

$$y = 8.167 - 0.0026 \times x1 - 0.0361 \times x2 + 0.0396 \times x3 - 0.692 \times x5 - 1.108 \times x7$$

Or with abbreviated variable names as:

$$y = 8.167 - 0.0026 \times altitude - 0.0361 \times slope + 0.0396 \times trees - 0.692 \times orientation - 1.108 \times vegetation$$

CLT-based confidence levels can also be constructed for the regression coefficients using the *confit* function and are provided in the table below:

```
clt <- confint(modelp) %>%
  as_tibble() %>%
  mutate(coefficient = c("Incercept", "x1", "x2", "x3", "x5", "x7")) %>%
  dplyr::select(coefficient, `2.5 %`, `97.5 %`)

clt %>% kable(caption = "CLT-based confidence intervals") %>%
  kable_styling(bootstrap_options = c("striped", "bordered"))
```

CLT-based confidence intervals

coefficient	**2.5 %**	**97.5 %**
Incercept	4.992	11.347
x1	-0.004	-0.001
x2	-0.065	-0.008
x3	-0.009	0.088
x5	-1.782	0.397
x7	-1.843	-0.372

An additional statistic of interest is the estimated standard deviation of the residuals or, also named, the residual standard deviation. This can be extracted using the *sigma* function is shown in the output below to be .531. This value tells us the standard deviation of the residual values which is the difference between the obseved and model fitted values.

```
sigma(modelp)
```

```
## [1] 0.531
```

Q16. [10 marks]

```
modelp <- lm(y ~ x1 + x2 + x3 + x5 + x7, data = cat)
tidyp <- tidy(modelp)

R <- 1000
n <- nrow(cat)

R_coeffs <- tibble(b0 = rep(0,R), b1 = rep(0, R), b2 = rep(0, R), b3=rep(0,R), b4=rep(0,R), b5 = rep(0, R))

set.seed(2020)
for(j in (1:R)){
  temp <- cat %>% slice_sample(n=n, replace=TRUE)
  tempf <- lm(y ~ x1 + x2 + x3 + x5 + x7, data = temp)
  tidyf <- tidy(tempf)
  R_coeffs[j,] <- t(tidyf$estimate)
}

beta_coeff <- function(b){
  R_coeffs %>%
  pull({{b}}) %>%
  quantile(c(0.025, 0.975))
}

boot_interval <- bind_rows(
  beta_coeff(b0),
  beta_coeff(b1),
  beta_coeff(b2),
  beta_coeff(b3),
  beta_coeff(b4),
  beta_coeff(b5))



boot_interval <- boot_interval %>%
  mutate(coefficient = c("b0", "b1", "b2", "b3", "b4", "b5")) %>%
  dplyr::select(coefficient, `2.5%`, `97.5%`)

boot_interval %>%
  kable(caption = "Bootstrap-based confidence intervals") %>%
  kable_styling(bootstrap_options = c("striped", "bordered", "hover"))
```

Bootstrap-based confidence intervals

coefficient	2.5%	97.5%
b0	5.106	11.673
b1	-0.005	-0.001
b2	-0.063	-0.007
b3	0.003	0.086
b4	-1.527	0.525
b5	-1.734	-0.391

```
modelp
```

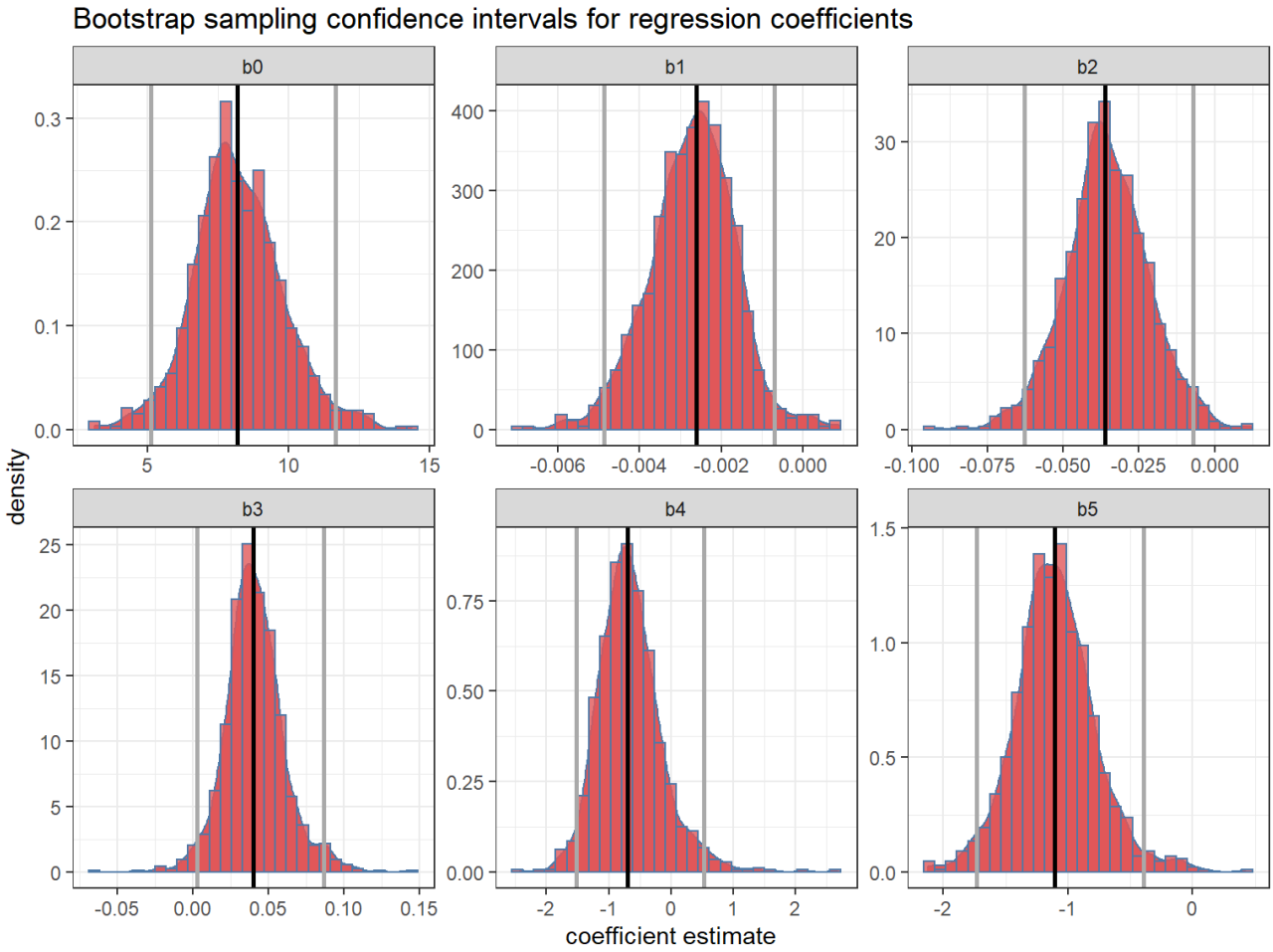
```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x5 + x7, data = cat)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x5           x7
##      8.1696      -0.0026      -0.0361       0.0396      -0.6923      -1.1078
```

The code above has been used to generate 1000 values from the bootstrap sampling distribution of the regression coefficients from the preferred model, modelp. This has been sampled from the original caterpillar data with replacement. The model of form modelp has then been refit for each sample and the regression coefficients saved for each of the 1000 replications. From this bootstrap sample, a 95% confidence interval was produced for each of the regression coefficients within this model and is shown in the table outputted above.

This interval helps provide an insight into the variability present in each estimate and, although not formally, helps identify which are statistically different from 0 and are therefore having an effect on the response variable - the logarithmic transform of the average number of nests of caterpillars per tree. The lower 2.5% and 97.5% quantiles of the data shown are the end points of this interval and represent the degree of uncertainty surrounding these estimated coefficients (which are shown in the model output above) in relation to the ‘true’ population values. For example, the intercept term b0 has an observed value of 8.167, however this only comes from a sample from the entire population. Looking at the produced confidence interval this can be expanded to say that we are 95% confident the ‘true’ value of b0 falls between 5.106 and 11.673 using the bootstrap method. The same method can be applied to all other estimates shown.

This can also be visualised by producing a histogram of the bootstrap distribution for each regression coefficients, which is shown in the output below. On this plot, the observed value of eat beta is shown by a black line whearas the two grey lines correspond to the lower and upper bounds of the confidence interval. This shows what was discussed above where the grey bars capture 95% of the simulated values for each estimate.

```
boot_plot <- R_coeffs %>%
  pivot_longer(names_to = "coefficient", cols = b0:b5) %>%
  left_join(boot_interval) %>%
  left_join(tidyp %>% mutate(coefficient = c("b0", "b1", "b2", "b3", "b4", "b5")) %>%
  dplyr::select(coefficient, estimate))

boot_plot %>% ggplot(aes(x = value, y = ..density..)) +
  geom_density(fill = "#e15759", colour = "#4e79a7") +
  geom_histogram(fill = "#e15759", colour = "#4e79a7", alpha = 0.8) +
  geom_vline(aes(xintercept = `2.5%`), size = 1, colour = "dark grey") +
  geom_vline(aes(xintercept = `97.5%`), size = 1, colour = "dark grey") +
  geom_vline(aes(xintercept = estimate), size = 1) +
  facet_wrap(~coefficient, scales = "free") +
  theme_bw() +
  ggtitle("Bootstrap sampling confidence intervals for regression coefficients") +
  xlab("coefficient estimate")
```



Part C: Additional Questions for ETC5242 Groups

Q17. [20 marks]

The code displayed below is used to conduct a two sided permutation test for b1 for the preferred model, designated as modelp. This test can be used to conduct the following hypothesis test at a 5% significance level.

$$H_0 : \beta_1 = 0$$

$$H_1 : \beta_1 \neq 0$$

Similar to the bootstrap process described earlier, here the data is sampled from the empirical data but this time without replacement. Additionally, as we are only testing the value of β_1 we only shuffle the values of the associated explanatory variable, in this case x1. Once this is done we re-fit modelp to this new sample and extract the estimate of the coefficient for β_1. As with the bootstrap sample, we repeat this process 1,000 times.

This process is done to break any existing association between the regressor and the response variable, which is y, in our collected data. The resulting distribution from these permutations represents the hypothetical distribution of β_1 under H_0. In regards to this model, this would mean that x1 (altitude in m) is not having a statistically significant effect on the number of caterpillar nests per tree (y variable). The estimate calculated for our model from the observed data can then be compared to this null distribution in order to calculate the p-value for the permutation test.

```
set.seed(2020)

cat2 <- cat
n <- nrow(cat2)
R <- 1000


Rcat2 <- cat2

for (r in 1:R) {
  Rcat2 <- Rcat2 %>% mutate(x1 = sample(cat2$x1, n, replace = FALSE))
  tempf <- lm(y ~ x1 + x2 + x3 + x5 + x7, data = Rcat2)
  tidyf <- tidy(tempf)
  R_coeffs[j,] <- t(tidyf$estimate)
}
```

The p-value calculated from this will be used to determine the outcome of the hypothesis tests shown earlier. As we are using a 5% confidence level we requiire a p-value of 0.05 or smaller for there to be enough evidence to reject the null hypothesis in favour of the alternative. To calculate this p-value, the proportion of randomised samples that are as or more extreme than the observed value are required which is designated as `b1obs` and has a value of -0.0026. We use this value to filter the permuted samples with estimates more extreme than this value. As this is a two-sided test we use the absolute value of both as we are interested in any difference regardless of direction.

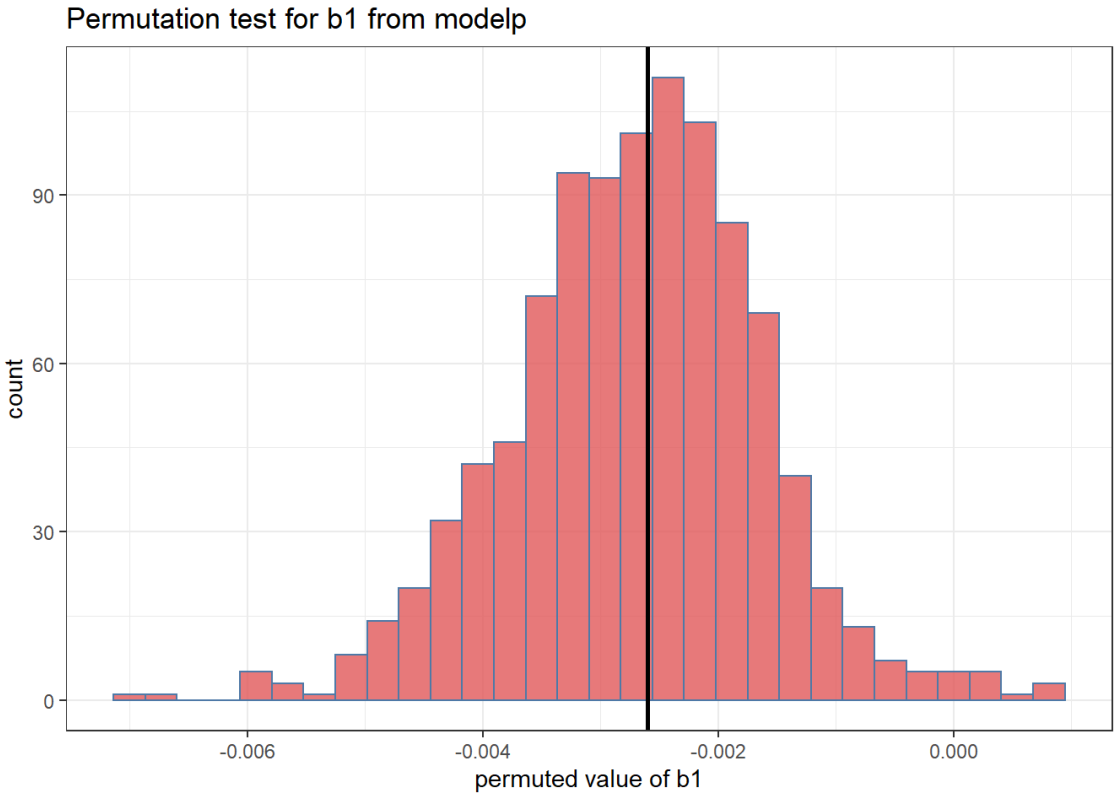
The code below performs this calculation and is saved as `pval` , and calling this value we see that it is 0.52. This means that 52% of our data sets sampled from the null distribution produced a β_1 more extreme than that observed from the 'real' data. Since this value is above 0.05 we cannot reject H_0 as there is not enough strength of evidence to do so. Thus it appears the value of β_1 is not significantly different from 0.

```
b1obs <- tidyp %>% filter(term == "x1") %>%
  pull(estimate)

pval <- R_coeffs %>%
  filter(abs(b1) >= abs(b1obs)) %>%
  nrow()/R

pval
```

```
## [1] 0.52
```

This can be viusalised, as with the bootstrap, by producing a histogram of the permuted samples. Here the black line relates to the observed value of β_1 from the modelp. From this we can see that roughly half of the values produced were more extreme than what was modeled. This is further evidence for the failure to reject H_0.

```
R_coeffs %>%
  dplyr::select(b1) %>%
  mutate(b1obs = b1obs) %>%
  ggplot(aes(x = b1)) +
  geom_histogram(fill = "#e15759", colour = "#4e79a7", alpha = 0.8) +
  geom_vline(xintercept = b1obs, size = 1) +
  theme_bw() +
  ggtitle("Permutation test for b1 from modelp") +
  xlab("permuted value of b1")
```



Q18. [15 marks]

An additional method for model validation is leave one out cross validation, denoted as LOOCV. This method works by leaving out one data point and building the model on the remaining data. The produced model is then used to predict the value of the removed observation. This process is repeated for all of the observations. Once complete the overall prediction error can be calculated by taking the average of the test errors associated with each of the predictions. The function created below calculates this value for each model by taking the mean of the residuals over the leverage squared. The output is provided in the table.

```
loocv <- function(model){
  h = lm.influence(model)$h
  mean((residuals(model)/(1-h))^2)
}

tibble("model" = c("modelf", "modelp")) %>%
  mutate("loocv" = c(loocv(modelf), loocv(modelp))) %>%
  kable(caption = "LOOCV values for modelf and modelp") %>%
  kable_styling(bootstrap_options = c("striped", "bordered"))
```

LOOCV values for modelf and modelp

model	loocv
modelf	0.372
modelp	0.315

From this table we see that modelp has a lower LOOCV value and therefore a lower overall error - even with fewer regressors. This means that the preferred model performs better than the full model at predicting the response variable y - the logarithmic transform of the average number of nests of caterpillars per tree. This then supports the process undertaken to determine the best model, and the removal of x4, x6, and x8 from the linear model. Respectively, this means that it is unlikely that the height of the center tree, the height of the dominant tree and the settlement index are associated with the number of caterpillar nests per tree.