ENVX2001 LAB
07 - REGRESSION MODEL DEVELOPMENT

Table of contents

Exercise 1: Bird abundance . . . 1
- Histograms . . . 2

`hist()` . . . 2

`hist.data.frame()` from `Hmisc` . . . 9

`ggplot()` . . . 9

`ggplot()` with `dplyr` . . . 10
- Correlation matrix . . . 11
- Plotting correlation . . . 12

Scatterplot matrix . . . 12

Correlation matrix . . . 13
- Transformations . . . 14

Exercise 2: Modelling bird abundance . . . 15
- Best single predictor? . . . 16
- Assumptions and interpretation . . . 16

At your own time: California streamflow . . . 18
- Partial F-Tests . . . 19

> **Tip**
>
> Please work on this exercise by creating your own R Markdown file.

Exercise 1: Bird abundance

This is the *same* dataset used in the lecture.

Fragmentation of forest habitat has an impact of wildlife abundance. This study looked at the relationship between bird abundance (bird ha^{-1}) and the characteristics of forest patches at 56 locations in SE Victoria.

The predictor variables are:

- `ALT` Altitude (m)
- `YR.ISOL` Year when the patch was isolated (years)
- `GRAZE` Grazing (coded 1-5 which is light to heavy)

- `AREA` Patch area (ha)
- `DIST` Distance to nearest patch (km)
- `LDIST` Distance to largest patch (km)

Import the data from the "Loyn" tab in the MS Excel file.

```
library(readxl)
loyn <- read_xlsx("mlr.xlsx", "Loyn")
```

Often, the first step in model development is to examine the data. This is a good way to get a feel for the data and to identify any issues that may need to be addressed. In this case, we will examine the data using histograms and a correlation matrix.

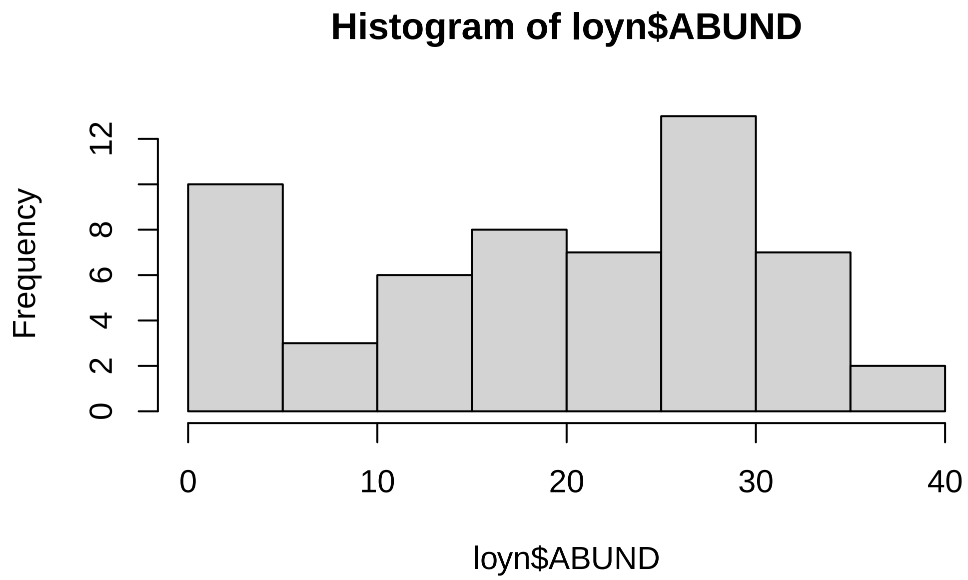
Histograms

There are a breadth of ways to create histograms in R. In each tab below you will find some different ways to create the same plot outputs.

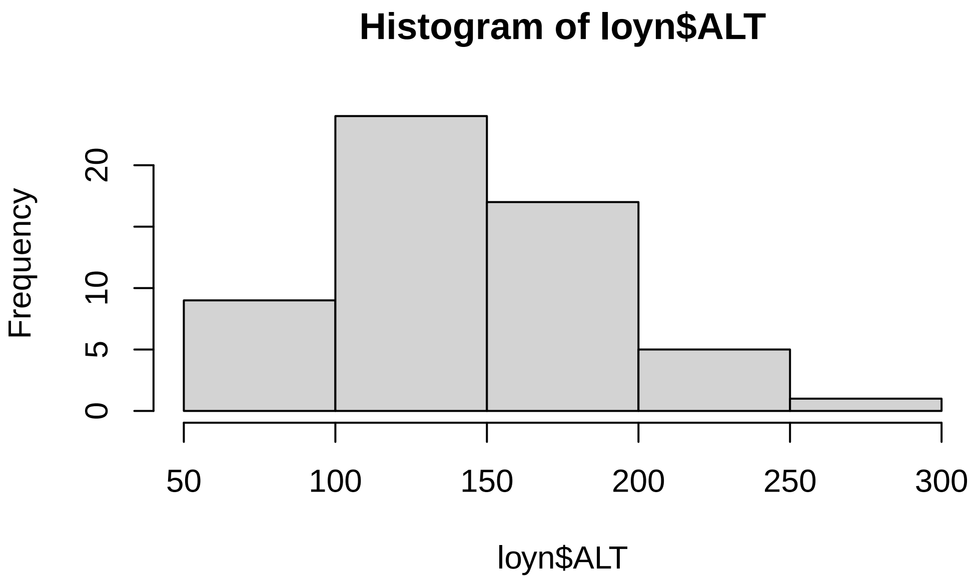
`hist()`

This is a straightforward way to create multiple histograms with `hist()`. The `par()` function is used to arrange the plots on the page. The `mfrow` argument specifies the number of rows and columns of plots.

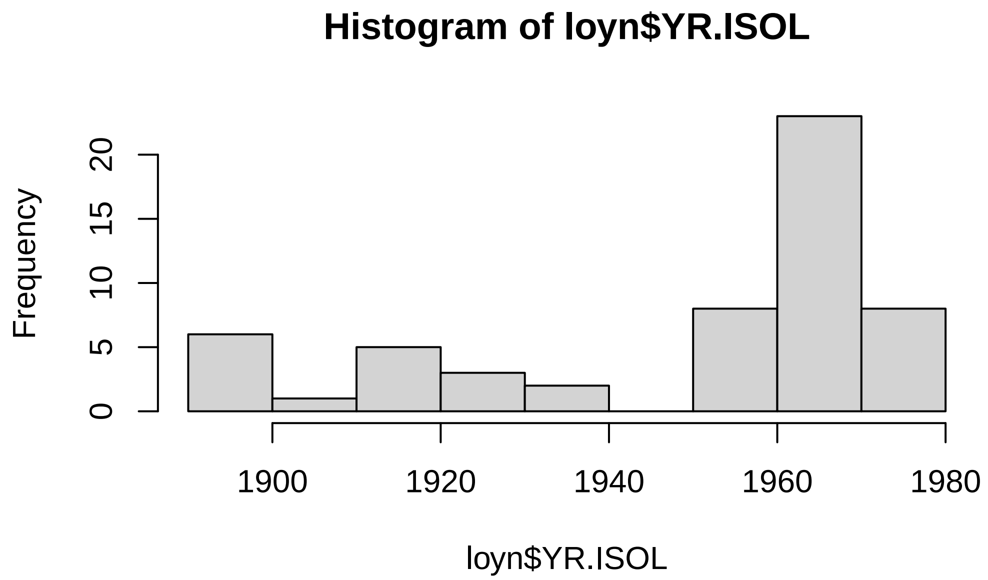
```
#par(mfrow=c(3,3))
hist(loyn$ABUND)
```



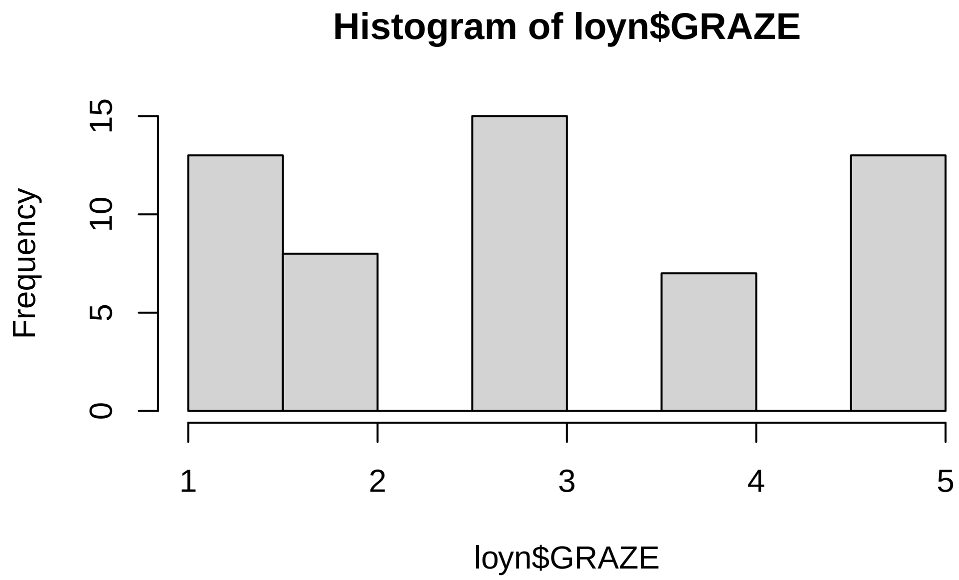

```
hist(loyn$ALT)
```



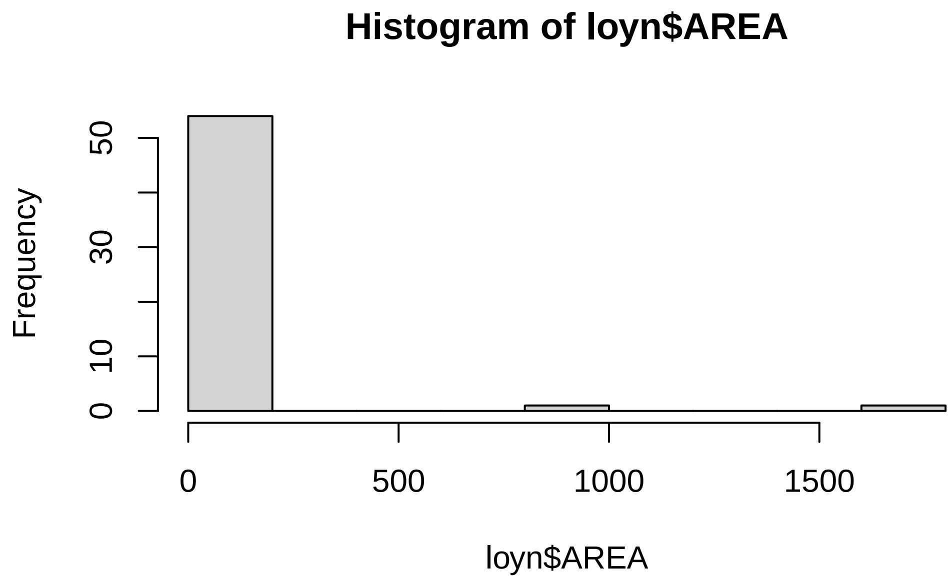

```r
hist(loyn$YR.ISOL)
```



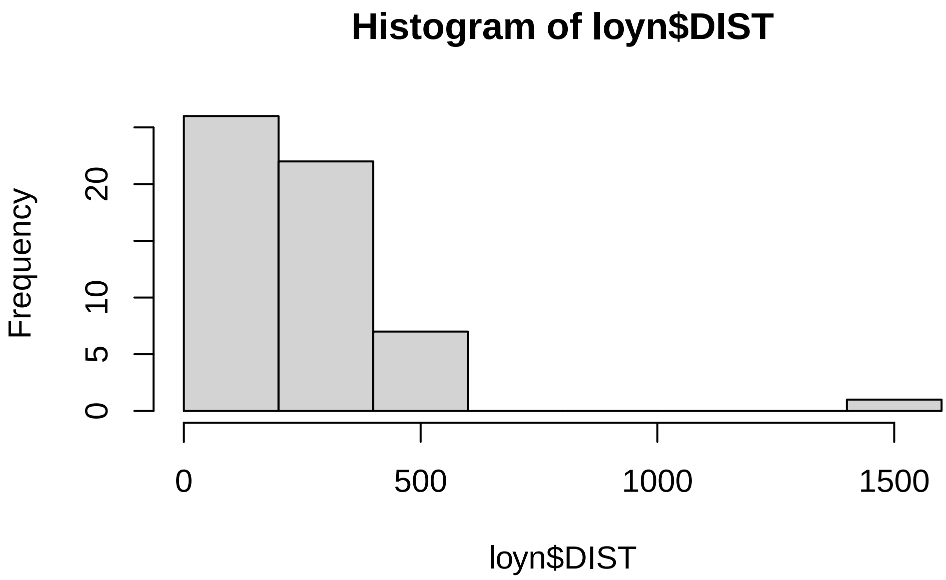

```
hist(loyn$GRAZE)
```



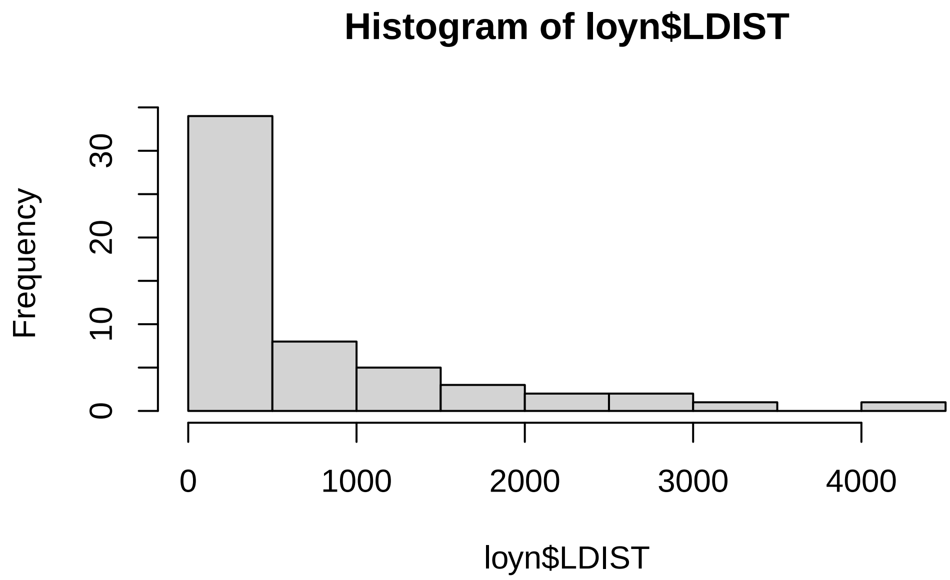

```
hist(loyn$AREA)
```




```
hist(loyn$DIST)
```




```
hist(loyn$LDIST)
```




```
#par(mfrow=c(1,1))
```

`hist.data.frame()` from `Hmisc`

The `Hmisc` package provides a function `hist.data.frame()` that can be used to create multiple histograms, which can be called by simply using `hist()`. You may need to tweak the `nclass` argument to get the desired number of bins, as the default may not look appropriate.

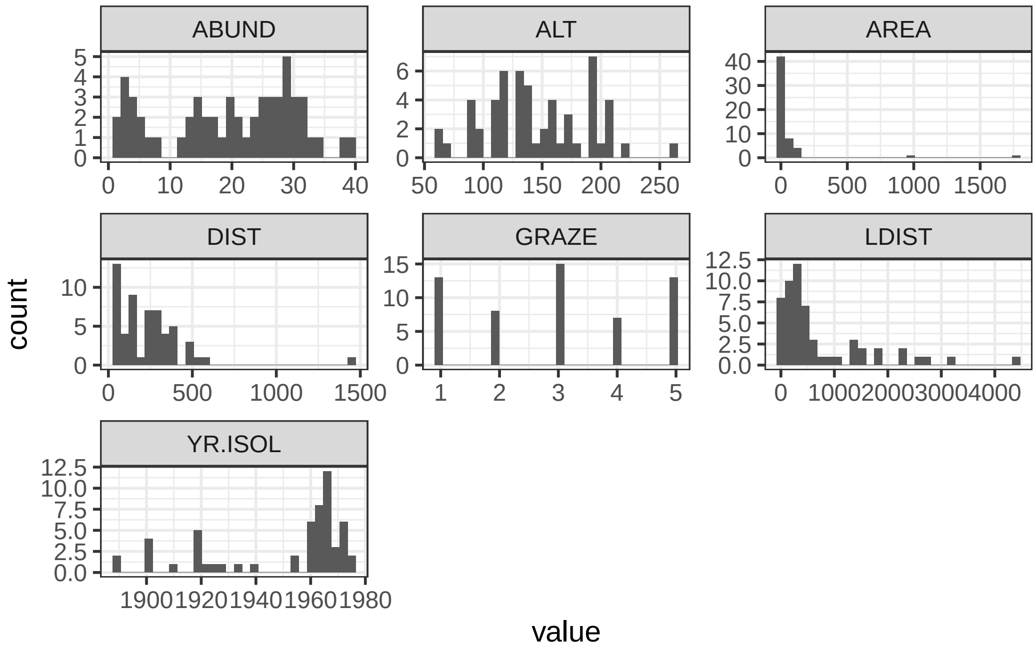
```
# install.packages("Hmisc")
library(Hmisc)
hist(loyn, nclass = 50)
```

`ggplot()`

A more modern approach is to use `ggplot()` with `facet_wrap()` to arrange multiple plots on a single page. To do this, the `pivot_longer()` function from the `tidyr` package is used to reshape the data into a tidy format.

```r
# tidy the data
loyn_tidy <- pivot_longer(loyn, cols = everything())

# plot
ggplot(loyn_tidy, aes(x = value)) +
  geom_histogram() +
  facet_wrap(~name, scales = "free") +
  theme_bw()
```

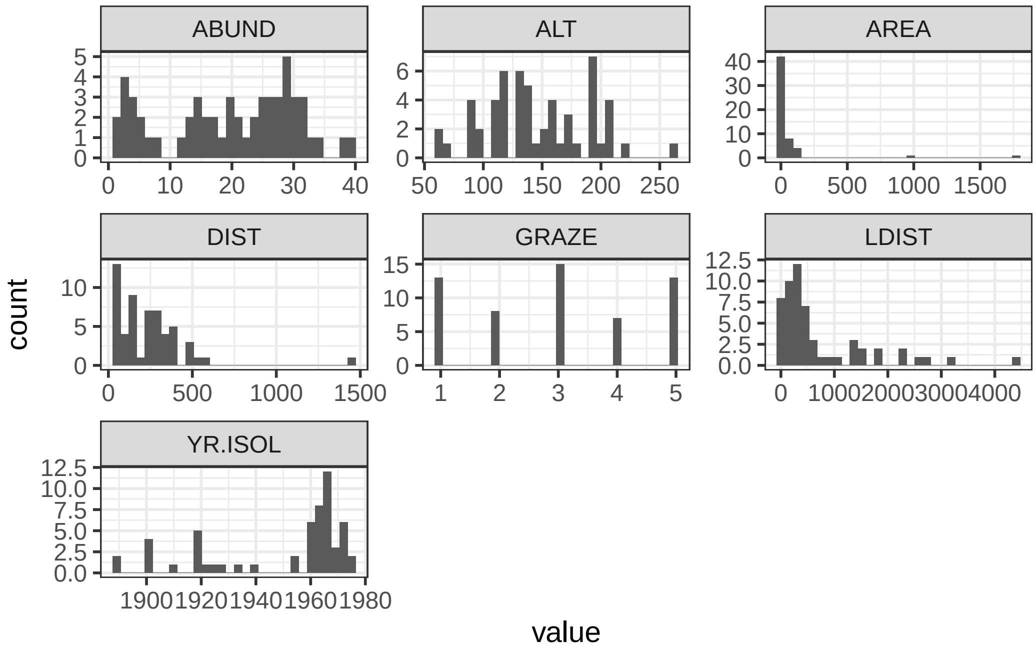


`ggplot()` with `dplyr`

Here we use the pipe operator `%>%` from `dplyr` to chain together a series of commands. The pipe operator takes the output of the command on the left and passes it to the command on the right (or below) the pipe. This means that we can create a series of commands that are executed in order.

```r
loyn %>%
  pivot_longer(cols = everything()) %>%
```

```r
ggplot(aes(x = value)) +
geom_histogram() +
facet_wrap(~name, scales = "free") +
theme_bw()
```



> ⚠ Question 1
>
> Comment on the histograms in terms of leverage. *Hint: what is the relationship between leverage and skewness?*

Correlation matrix

Calculate the correlation matrix using `cor(Loyn)`.

```r
cor(loyn)
```

```
              ABUND          AREA     YR.ISOL        DIST       LDIST
```

```
ABUND    1.00000000  0.255970206  0.503357741  0.2361125  0.08715258
AREA     0.25597021  1.000000000 -0.001494192  0.1083429  0.03458035
YR.ISOL  0.50335774 -0.001494192  1.000000000  0.1132175 -0.08331686
DIST     0.23611248  0.108342870  0.113217524  1.0000000  0.31717234
LDIST    0.08715258  0.034580346 -0.083316857  0.3171723  1.00000000
GRAZE   -0.68251138 -0.310402417 -0.635567104 -0.2558418 -0.02800944
ALT      0.38583617  0.387753885  0.232715406 -0.1101125 -0.30602220
              GRAZE         ALT
ABUND   -0.68251138  0.3858362
AREA    -0.31040242  0.3877539
YR.ISOL -0.63556710  0.2327154
DIST    -0.25584182 -0.1101125
LDIST   -0.02800944 -0.3060222
GRAZE    1.00000000 -0.4071671
ALT     -0.40716705  1.0000000
```

Question 2

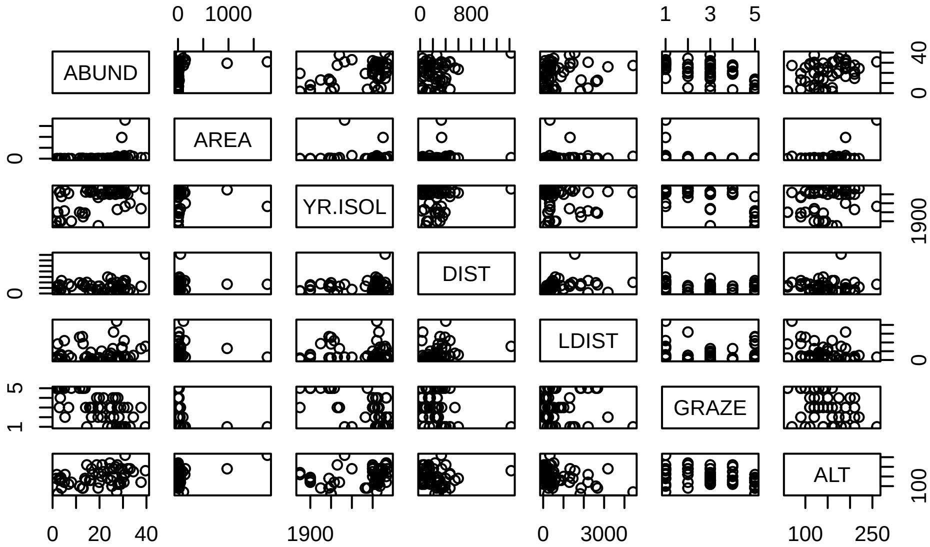
Which independent variables are useful for predicting the dependent variable abundance? Is there evidence for multi-collinearity?

Plotting correlation

Examine correlations visually using `pairs()` or `corrplot()` from the `corrplot` package.

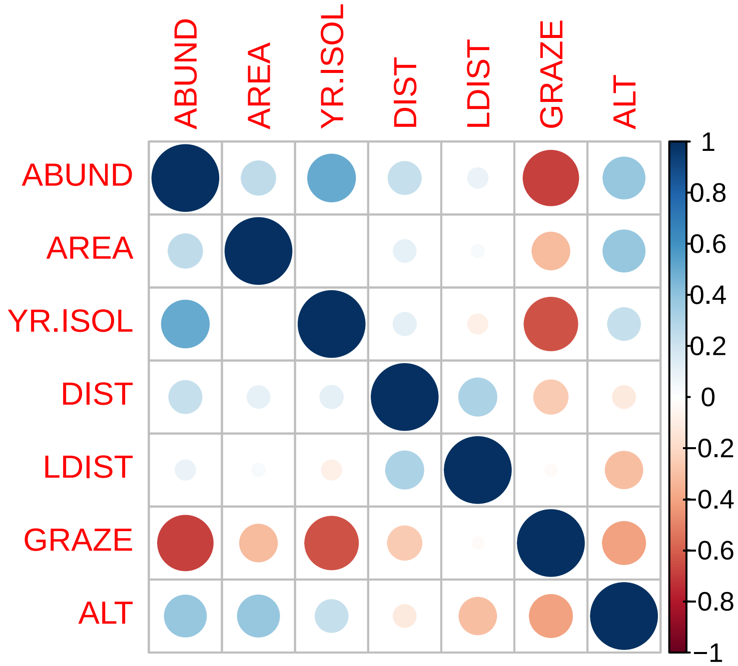
Scatterplot matrix

```r
pairs(loyn)
```



Correlation matrix

```r
library(corrplot)
corrplot(cor(loyn))
```



> **Question 3**
>
> Are there any trends visible from the plots?

> **Tip**
>
> We can also bring in variance inflation factors (VIF) to help us identify multi-collinearity, but that is done only after we have selected a model.

Transformations

The AREA predictor has a small number of observations with very large values. Apply a $\log_{10}$ transformation and label the new variable `Loyn$L10AREA`.

```r
loyn$L10AREA <- log10(loyn$AREA)
```

> ⚠ Question 4
>
> Why are we transforming AREA?

> ⚠ Question 5
>
> Re-run `pairs(Loyn)` and create a histogram using the transformed value of AREA, how do the plots look?

```
hist(loyn$L10AREA)
pairs(loyn)
```

> ⚠ Question 6
>
> In preparation for modelling, transform the remaining skewed variables, DIST and LDIST the same way you did for AREA and examine the histogram and pairs plots using these new variables.

Make sure you end up with two new variables labelled `loyn$L10DIST` and `loyn$L10LDIST`.

Exercise 2: Modelling bird abundance

We will now use the transformed data in `loyn` for this exercise. If you have not already figured out how to perform the transformation, or if something is wrong, you may use the `loyn` tab in the `mlr.xlsx` MS Excel document. Alternatively, the code to convert the data is below.

```
# reset the data import just in case it has been modified
loyn <- read_xlsx("mlr.xlsx", "Loyn")
# make transformations

loyn <- loyn %>%
  mutate(L10AREA = log10(AREA),
    L10DIST = log10(DIST),
    L10LDIST = log10(LDIST))

# check
```

```r
glimpse(loyn)
```

```
Rows: 56
Columns: 10
$ ABUND    <dbl> 5.3, 2.0, 1.5, 17.1, 13.8, 14.1, 3.8, 2.2, 3.3, 3.0, 27.6, 1.~
$ AREA     <dbl> 0.1, 0.5, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 2.0, 2~
$ YR.ISOL  <dbl> 1968, 1920, 1900, 1966, 1918, 1965, 1955, 1920, 1965, 1900, 1~
$ DIST     <dbl> 39, 234, 104, 66, 246, 234, 467, 284, 156, 311, 66, 93, 39, 4~
$ LDIST    <dbl> 39, 234, 311, 66, 246, 285, 467, 1829, 156, 571, 332, 93, 39,~
$ GRAZE    <dbl> 2, 5, 5, 3, 5, 3, 5, 5, 4, 5, 3, 5, 2, 1, 5, 5, 3, 3, 3, 2, 2~
$ ALT      <dbl> 160, 60, 140, 160, 140, 130, 90, 60, 130, 130, 210, 160, 210,~
$ L10AREA  <dbl> -1.0000000, -0.3010300, -0.3010300, 0.0000000, 0.0000000, 0.0~
$ L10DIST  <dbl> 1.591065, 2.369216, 2.017033, 1.819544, 2.390935, 2.369216, 2~
$ L10LDIST <dbl> 1.591065, 2.369216, 2.492760, 1.819544, 2.390935, 2.454845, 2~
```

Best single predictor?

> **Question 1**
>
> Obtain the correlation between ABUND and all of the predictor variables using `cor()`. Based on these, what would you expect to be the best single predictor of ABUND?

```r
cor(loyn)
```

Assumptions and interpretation

> **Question 2**
>
> Use multiple linear regression to see whether ABUND can be predicted from L10AREA and GRAZE. Are the assumptions met? Is there a significant relationship? *Note: we are using these 2 predictors as they have the largest absolute correlations. Use `lm()` and specify the model as `ABUND ~ L10AREA + GRAZE`.*

```r
lm.mod1 <- lm(ABUND~GRAZE + L10AREA, data=loyn)

par(mfrow=c(2,2))
plot(lm.mod1)
par(mfrow=c(1,1))

summary(lm.mod1)
```

Question 3

How good is the model based on the (i) r^2 (ii) adjusted r^2? Use `summary()`.

```r
summary(lm.mod1)$r.squared
summary(lm.mod1)$adj.r.squared
```

Question 4

Which variable(s) has the most significant effect(s)? *(Refer specifically to the t probabilities in the table of predictors and their estimated parameters or coefficients in the output of* `summary()`*).* Interpret the p-values in terms of dropping predictor variables.

Question 5

Repeat the multiple regression, but this time include YRS.ISOL as a predictor variable (it has the 3rd largest absolute correlation). This will allow you to assess the effect of YRS.ISOL with the other variables taken into account.

Question 6

Check assumptions, do the residuals look ok? If you are happy with the assumptions, you can proceed to interpret the model output.

Question 7

Compare the r^2 and adjusted r^2 values with those you calculated for the 2 predictor model, Which is the better model? Why?

```
summary(lm.mod2)
```

At your own time: California streamflow

> **Note**
>
> This additional exercise can be done at your own time. Most of the code are provided. You will need to run the code and interpret the results.

The following dataset contains 43 years of annual precipitation measurements (in mm) taken at (originally) 6 sites in the Owens Valley in California. I have reduced this to three variables labelled `L10APSAB` (Lake Sabrina), `L10OBPC` (Big Pine Creek), `L10OPRC` (Rock Creek), and the dependent variable stream runoff volume (measured in ML/year) at a site near Bishop, California (labelled `L10BSAAM`). There is also a variable `Year` but you can ignore this.

Note the variables have already been log-transformed to increase normality of the residuals in the regressions.

Start with a full model and manually remove the variables one at a time, checking every time whether removal of a variable actually improves the model.

```
# read in the data
s.data <- read_xlsx("mlr.xlsx", "California_streamflow")
names(s.data)
```

```
[1] "L10APSAB" "L10OBPC"  "L10OPRC"  "L10BSAAM"
```

```
s.mod_full <-lm(L10BSAAM~L10APSAB + L10OBPC + L10OPRC, data=s.data)
s.mod_full <-lm(L10BSAAM~., data=s.data) ## you can also use the . to indicate use all
 ↪  variables
summary(s.mod_full)
```

```
Call:
lm(formula = L10BSAAM ~ ., data = s.data)

Residuals:
```

```
     Min       1Q   Median       3Q      Max
-0.09885 -0.03331  0.01025  0.03359  0.09495

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.25716    0.12360  26.352  < 2e-16 ***
L10APSAB     0.05631    0.03756   1.499  0.14185
L10OBPC      0.21085    0.06756   3.121  0.00339 **
L10OPRC      0.43838    0.08798   4.983 1.32e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04861 on 39 degrees of freedom
Multiple R-squared:  0.8817,    Adjusted R-squared:  0.8726
F-statistic: 96.88 on 3 and 39 DF,  p-value: < 2.2e-16
```

Partial F-Tests

The above analysis tells us that both `L10OBPC` & `L10OPRC` are significant, according to the t-test, in the model and `L10APSAB` is not? This involves performing Partial F-Tests as discussed in the lecture.

This can be done in **R** by using `anova()` on two model objects. To be able to compare the models and run the anova, you need to make objects of all the possible model combinations you want to compare.

```r
s.mod_reduced <- lm(L10BSAAM ~ L10OPRC + L10OBPC, data=s.data)
anova(s.mod_reduced, s.mod_full)
```

The last row gives the results of the partial F-test.

> ⚠ Question 1
>
> Should we remove `L10APSAB` from the model?

> ⚠ Question 2
>
> Is the p-value for the f-test the same as for the t-test?

Question 3

Write out the hypotheses you are testing.

Perform a Partial F-Test to work out if the removal of `L10APSAB` and `L10OBPC` improves upon the full model.

```
s.mod_reduced2  <- lm(L10BSAAM ~ L10APSAB + L10OBPC,data=s.data)
anova(s.mod_reduced2, s.mod_full)
```

```
Analysis of Variance Table

Model 1: L10BSAAM ~ L10APSAB + L10OBPC
Model 2: L10BSAAM ~ L10APSAB + L10OBPC + L10OPRC
  Res.Df      RSS Df Sum of Sq     F    Pr(>F)
1     40 0.150845
2     39 0.092166  1   0.05868 24.83 1.321e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 4

Which variable should be added to the model containing L10OPRC?

Question 5

Could things be even simpler? Perform a partial F-Test to see if a model containing L10OPRC alone could be suitable.

```
s.mod_reduced3  <- lm(L10BSAAM ~ L10OPRC,data=s.data)
anova(s.mod_reduced3, s.mod_full)
```

Question 6

What is your optimal model?

That's it for today! Great work fitting simple and multiple linear regression! Next week we jump into stepwise selection and predictive modelling!