STATS 3001 / STATS 4104 / STATS 7054
Statistical Modelling III
Workshop 1 - beyond linear models

John Maclean

Week 1

Load packages

```
pacman::p_load(tidyverse, gglm, janitor)
```

I'm going to make a theme for my figures.

```
#apa theme
apatheme=theme_bw()+
    theme(panel.grid.major=element_blank(),
          panel.grid.minor=element_blank(),
          panel.border=element_blank(),
          axis.line=element_line(),
          text=element_text(family='Times'),
          legend.title=element_blank(),
          axis.text.y=element_text(size = 12),
          axis.text.x=element_text(size = 12))
theme_set(apatheme) #change as you like
```

Data

Get the data

```
clams <- read_delim('./ZuurData/Clams.txt') #read_delim will infer the delimitor
clams
```

```
## # A tibble: 398 x 5
##    MONTH LENGTH   AFD LNLENGTH LNAFD
##    <dbl>  <dbl> <dbl>    <dbl> <dbl>
##  1    11   28.4 0.248     3.35 -1.39
##  2    11   16.6 0.052     2.81 -2.96
##  3    11   13.7 0.028     2.62 -3.57
##  4    11   17.4 0.07      2.86 -2.66
##  5    11   11.8 0.022     2.47 -3.83
##  6    11   28.1 0.187     3.34 -1.68
##  7    11   32.6 0.361     3.48 -1.02
##  8    11   15.7 0.05      2.75 -3.01
##  9    11   18.7 0.087     2.93 -2.44
## 10    11   21.5 0.128     3.07 -2.06
## # ... with 388 more rows
```

Clean up the names:

```
clams <- janitor::clean_names(clams)
clams
```

```
## # A tibble: 398 x 5
##    month length   afd lnlength lnafd
##    <dbl>  <dbl> <dbl>    <dbl> <dbl>
##  1    11   28.4 0.248     3.35 -1.39
##  2    11   16.6 0.052     2.81 -2.96
##  3    11   13.7 0.028     2.62 -3.57
##  4    11   17.4 0.07      2.86 -2.66
##  5    11   11.8 0.022     2.47 -3.83
##  6    11   28.1 0.187     3.34 -1.68
##  7    11   32.6 0.361     3.48 -1.02
##  8    11   15.7 0.05      2.75 -3.01
##  9    11   18.7 0.087     2.93 -2.44
## 10    11   21.5 0.128     3.07 -2.06
## # ... with 388 more rows
```

Read about the dataset and clearly name the weight measurements and the log-transforms:

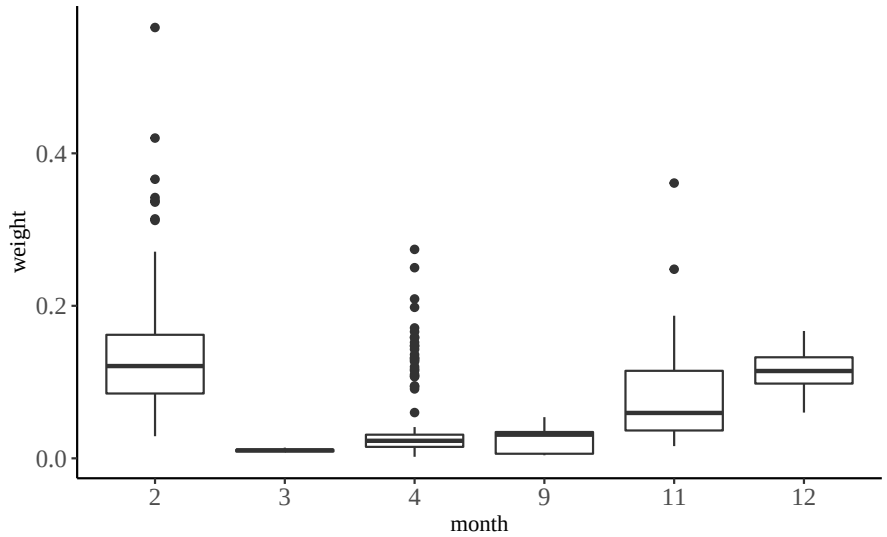
```
clams <- transmute(clams,
                   month = as_factor(month), length, #keep these columns
                weight = afd, #rename the following columns
                log_length = lnlength,
                log_weight = lnafd
                )
clams
```

```
## # A tibble: 398 x 5
##    month length weight log_length log_weight
##    <fct>  <dbl>  <dbl>      <dbl>      <dbl>
##  1 11      28.4  0.248       3.35      -1.39
##  2 11      16.6  0.052       2.81      -2.96
##  3 11      13.7  0.028       2.62      -3.57
##  4 11      17.4  0.07        2.86      -2.66
##  5 11      11.8  0.022       2.47      -3.83
##  6 11      28.1  0.187       3.34      -1.68
##  7 11      32.6  0.361       3.48      -1.02
##  8 11      15.7  0.05        2.75      -3.01
##  9 11      18.7  0.087       2.93      -2.44
## 10 11      21.5  0.128       3.07      -2.06
## # ... with 388 more rows
```

EDA

Have a look at the data. We want to predict weight. Let's look at the relationship with month:

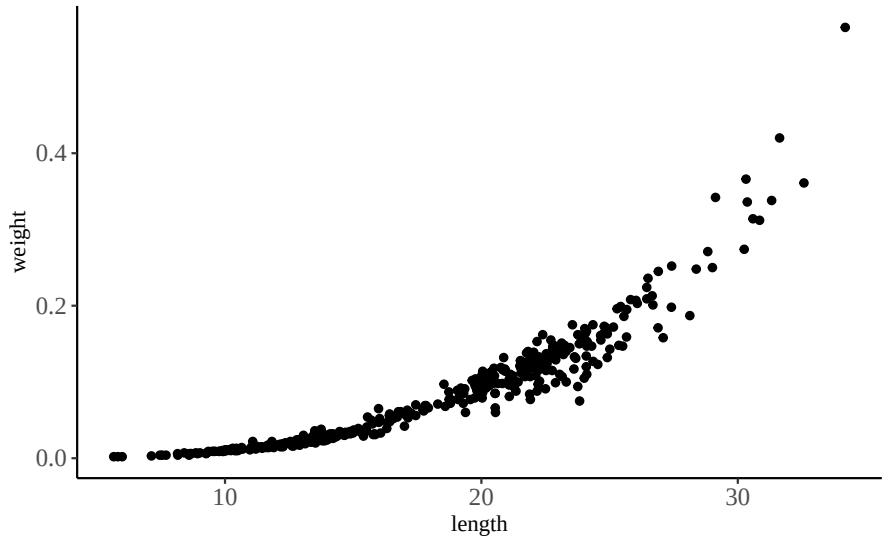
```
#This is also a quick ggplot tutorial/reminder
ggplot(clams, #choose the dataset
       aes(month, weight) #choose what variables to plot
       ) +
  geom_boxplot()  #choose what to plot
```



Worrying observation one: the spread of weights is quite different for different months. Also the month entries are not every month, but at the following months: 2, 3, 4, 9, 11, 12

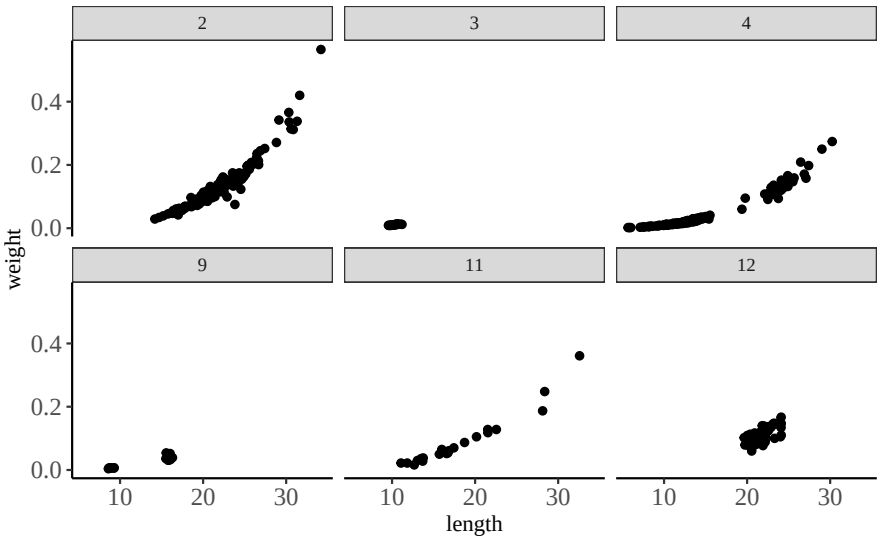
Now look at length:

```
ggplot(clams, #choose the dataset
       aes(length, weight) #choose what variables to plot
       ) +
  geom_point()  #choose what to plot
```



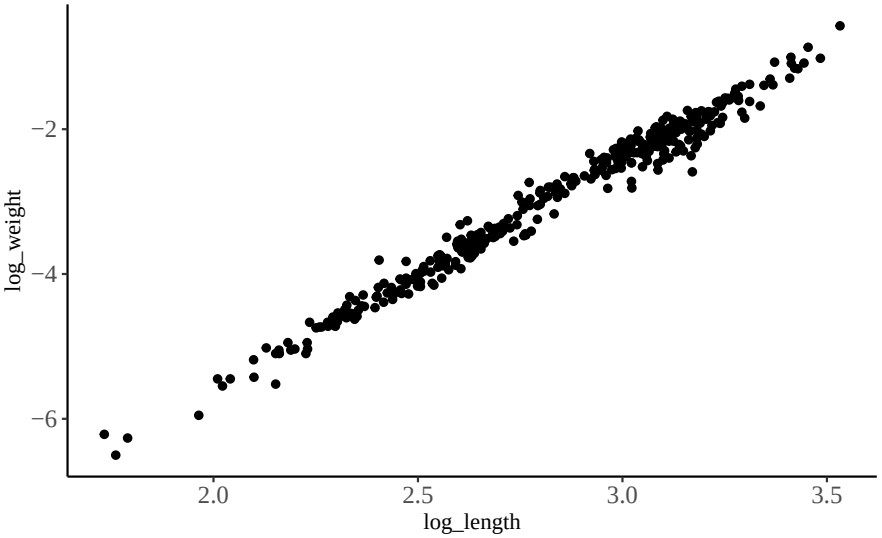
What about both?

```
ggplot(clams, #choose the dataset
       aes(length, weight) #choose what variables to plot
       ) +
  geom_point() + #choose what to plot
  facet_wrap(~month)
```



At this point it's clear why the log transforms are included in the dataset:

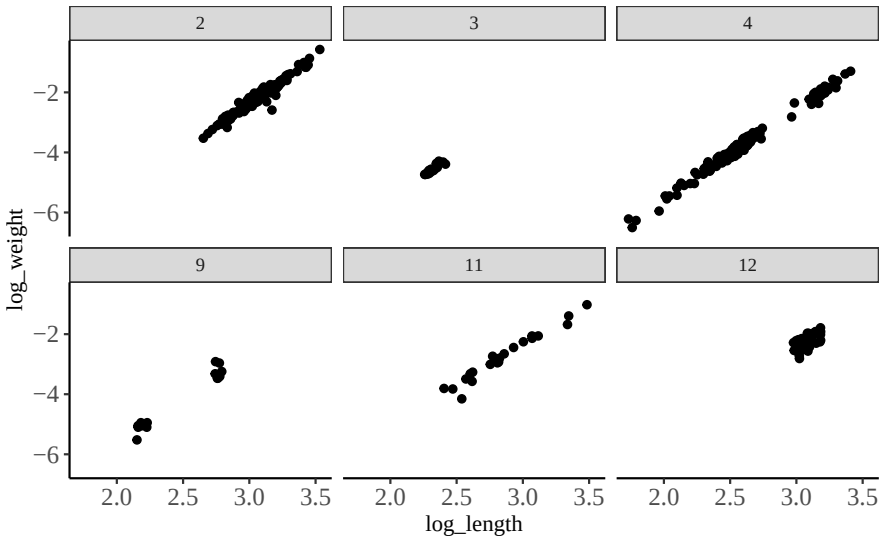
```
ggplot(clams, #choose the dataset
       aes(log_length, log_weight) #choose what variables to plot
       ) +
  geom_point()  #choose what to plot
```



Three connected questions: why did we plot `log_length` vs `log_weight`, why does the log-log data fit neatly around a line with slope 3, and what transformation could we have guessed for `weight` that would have led to an almost-linear relationship with `length`? Hint: all clams are spherical until proven otherwise.

Moving on. At this point we could write down a reasonable model for (log) weight using (log) length as a predictor. Unfortunately, the ecologists insist on you including `month`. Consider that again:

```
ggplot(clams, #choose the dataset
       aes(log_length, log_weight) #choose what variables to plot
       ) +
  geom_point() + #choose what to plot
  facet_wrap(~month)
```



Key discussion

This figure is hopefully deeply frustrating to you. We can clearly see that all clams have a similar relationship between length and weight, regardless of the month. **However**, knowing the month can also help narrow down the likely weights (for example, in month 3 or 12).

The key question that motivates the rest of the workshops: can we include the information from `month` in a linear model without breaking it? (Answer: no. But you just WAIT)

Model

At this point, please consider if any tool in your arsenal can be applied to model the relationship between length, month and weight. If you think so, give it a go. Make sure to verify assumptions afterward.

Here's mine: perhaps we can include an *interaction term* between the two predictors. Let's try it:

```
model_interact <- lm(log_weight ~ log_length * month, data = clams)
summary(model_interact)
```

```
##
## Call:
## lm(formula = log_weight ~ log_length * month, data = clams)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.73740 -0.06472  0.01014  0.08337  0.42577
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -11.47192    0.20194 -56.809  < 2e-16 ***
## log_length           3.03384    0.06555  46.286  < 2e-16 ***
## month3               0.17483    1.69538   0.103  0.91792
## month4              -0.03489    0.21853  -0.160  0.87324
## month9              -0.50343    0.35282  -1.427  0.15442
## month11              0.93373    0.34311   2.721  0.00680 **
## month12              1.63580    0.97881   1.671  0.09549 .
## log_length:month3   -0.13145    0.72773  -0.181  0.85675
```

```
## log_length:month4   -0.04575    0.07284  -0.628  0.53035
## log_length:month9    0.11172    0.13256   0.843  0.39984
## log_length:month11  -0.30471    0.11680  -2.609  0.00944 **
## log_length:month12  -0.55618    0.31739  -1.752  0.08051 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1293 on 386 degrees of freedom
## Multiple R-squared:  0.9868, Adjusted R-squared:  0.9865
## F-statistic:  2629 on 11 and 386 DF,  p-value: < 2.2e-16
```

Use `drop1` to see if the interaction is significant:

```
drop1(model_interact, test = "F") #compare to model log_weight ~ log_length + month
```

```
## Single term deletions
##
## Model:
## log_weight ~ log_length * month
##                  Df Sum of Sq    RSS     AIC F value  Pr(>F)
## <none>                        6.4490 -1616.8
## log_length:month  5   0.20328 6.6523 -1614.4  2.4334 0.03444 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

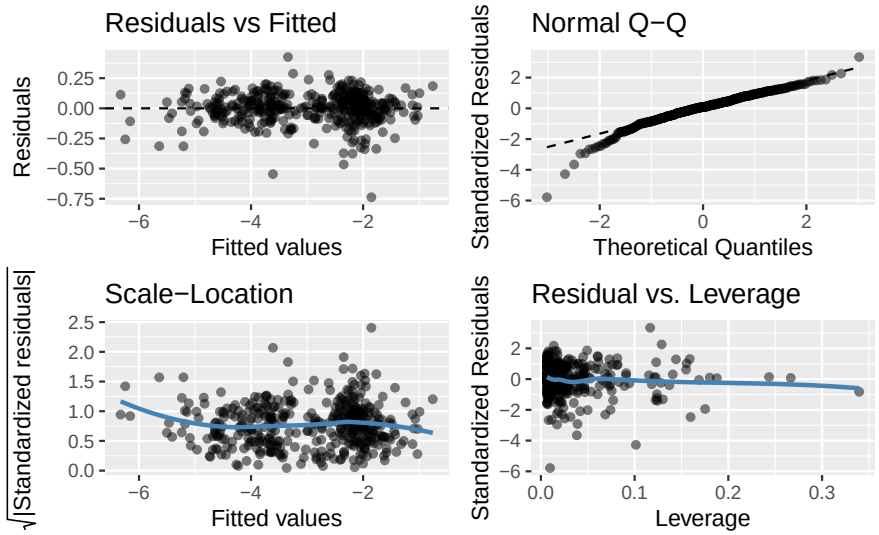
```
# drop1(model_interact, test = "F", scope = "month") #compare to model log_weight ~ log_length
```

The line marked shows output from the full model with the interaction term. The next line drops the interaction term. The F-statistic suggests that the interaction is significant (`p=0.034`).

Do we trust this? What should we do?

Model Verification

```
gglm(model_interact)
```



Some non-normality.

Worse: spread in residuals is not the same for different fitted values - heteroscedasticity. This fact was first visible back when we plotted `month` vs `log_weight`, but it's not become a problem in our model.

Outlook

The model will be a bad analysis in some areas because a linear model fits a single parameter to describe the variance of the residuals. But, as shown by the heteroscedasticity, the variance of the residuals depends on the month.

What we need here is really very simple: we need a different parameter, for each month, that describes the variance of the data. If we had that - then all would be well. The overall issue is the *heteroscedasticity* of the data. We'll deal with that using `gls` models later in the workshops.

So you'd better keep coming!