Linear models II

Etienne Low-Décarie

2017-12-20

General Linear Models in R: Part 2

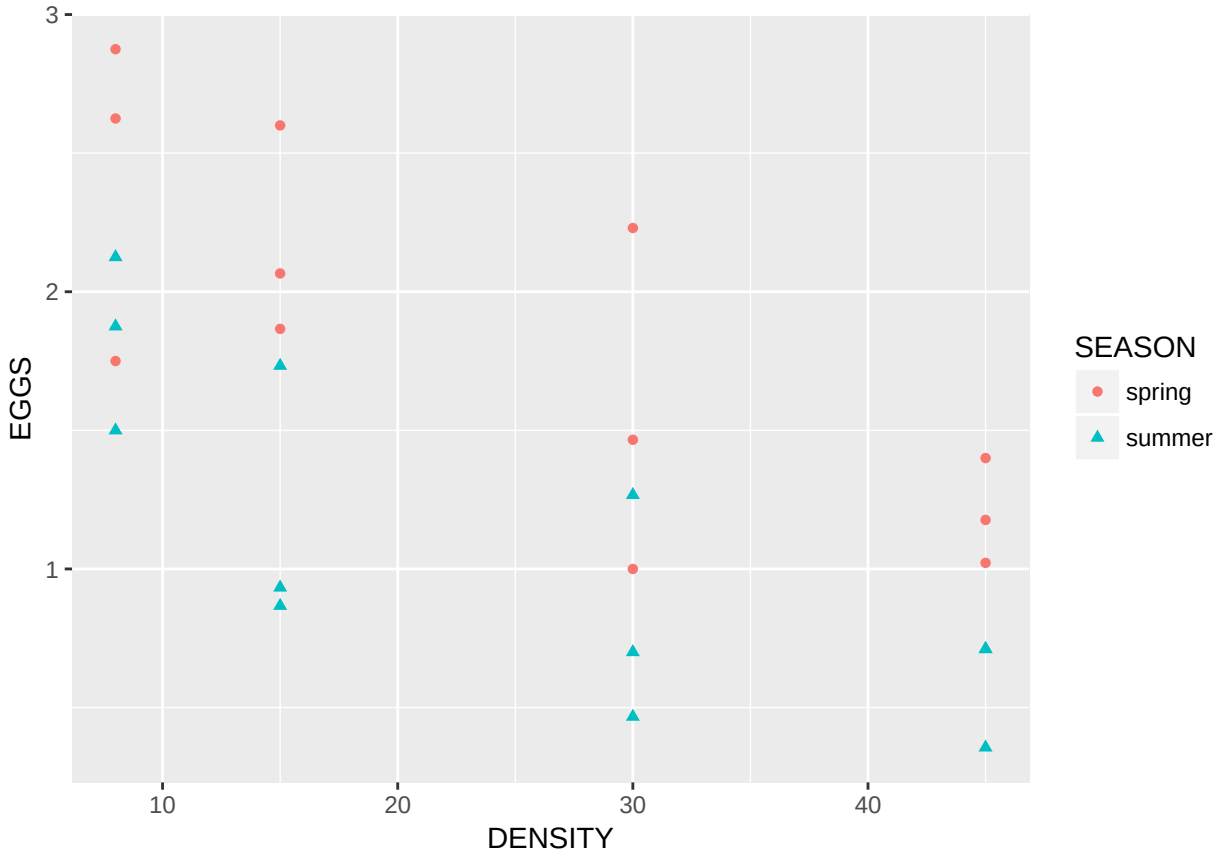
A 2-Way Factorial Analysis of Variance of Limpets

This dataset is taken from the Quinn and Keough book. It is a study looking at density dependent fecundity in limpets in two different seasons. The null hypothesis is that there is no effect of density or season on the production of eggs. Alternatives include there being a density effect that differs between seasons (interaction between density and season) or an additive effect of season and density (no interaction – e.g. slope of density dependence is the same, but means are lower or higher between seasons). > Step 1: Import the limpet data into R and attach the data sheet. Identify the names of the columns in R and do a panel plot of eggs vs. density, split by season

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
p <- qplot(data=limp,
           x=DENSITY,
           y=EGGS,
           colour=SEASON,
           shape=SEASON)
print(p)
```



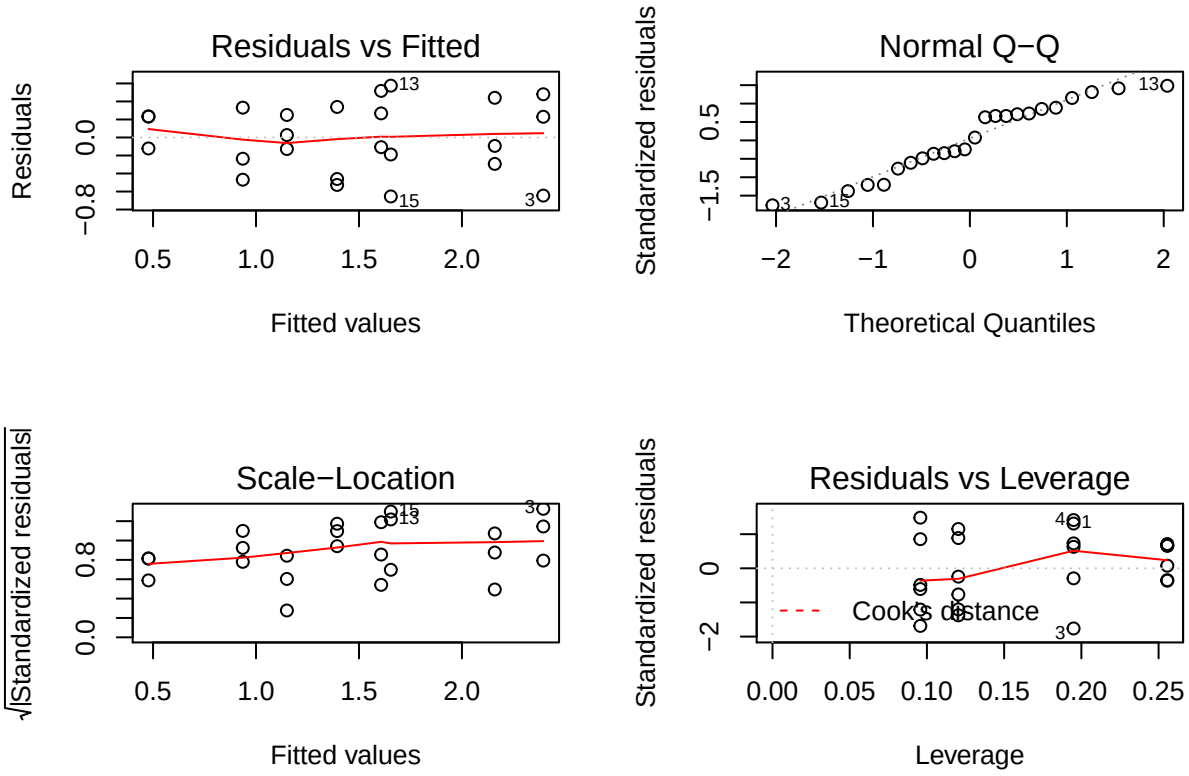
Ask yourself now – what kind of analysis are we doing? ANOVA, ANCOVA, Multiple Regression? Which variables are categorical and which are continuous? Can you find out?

```
str(limp)
```

```
## 'data.frame':    24 obs. of  3 variables:
##  $ DENSITY: int  8 8 8 8 8 8 15 15 15 15 ...
##  $ SEASON : Factor w/ 2 levels "spring","summer": 1 1 1 2 2 2 1 1 1 2 ...
##  $ EGGS   : num  2.88 2.62 1.75 2.12 1.5 ...
```

Based on the graph, what do I expect my 2-way analysis to show – the null hypothesis, the alternative of an interaction, or the alternative of no interaction? To do this analysis, we use the workhorse, lm(), follow this immediately by a check of assumptions, then an examination of the anova table and the coefficients table. lm performs general linear models – these models assume (near) normally distributed errors, but are robust to deviations in sample sizes (unbalanced designs) and robust to normality with very large sample sizes.

```
m1<-lm(EGGS~DENSITY*SEASON,
       data=limp)
par(mfrow=c(2,2))
plot(m1)
```




```
anova(m1)
```

```
## Analysis of Variance Table
##
## Response: EGGS
##                Df Sum Sq Mean Sq F value    Pr(>F)
## DENSITY         1 5.0241  5.0241 30.1971 2.226e-05 ***
## SEASON          1 3.2502  3.2502 19.5350 0.0002637 ***
## DENSITY:SEASON  1 0.0118  0.0118  0.0711 0.7925333
## Residuals      20 3.3275  0.1664
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m1)
```

```
##
## Call:
## lm(formula = EGGS ~ DENSITY * SEASON, data = limp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65468 -0.25021 -0.03318  0.28335  0.57532
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.664166   0.234118  11.380 3.45e-10 ***
## DENSITY               -0.033650   0.008259  -4.074 0.000591 ***
## SEASONsummer          -0.812282   0.331092  -2.453 0.023450 *
## DENSITY:SEASONsummer   0.003114   0.011680   0.267 0.792533
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4079 on 20 degrees of freedom
## Multiple R-squared:  0.7135, Adjusted R-squared:  0.6705
## F-statistic:  16.6 on 3 and 20 DF,  p-value: 1.186e-05
```

before we go over the details: * why do we assign the model a name `m1`? * What does `DENSITY*SEASON` mean in the model? * What does `par(mfrow=c(2,2))` do? * What should plot(m1) produce?

Some Interpretations

The "Call" in `summary()` specifies your model – is this what you wanted? The Residuals gives the quartiles of the residuals – are they evenly distributed? The coefficients are as in any statistical package – the estimates for each term are the coefficients that help describe change in the dependent variable as a function of the independent variable(s).

The Residual Standard Error, Multiple R2 and Adjusted R2 tell you the standard things about variance explanation. Finally, the F-test, degrees of freedom and overall significance of the model is presented. Do the diagnostic plots show anything bad? The anova table shows which if any terms are significant? Is this table a sequential sums of squares? HINT - google is your friend. Which terms are marginal to the interaction? What does this mean? HINT - google is your friend. Which Hypothesis does this analysis support? Remove the interaction term (e.g. reanlyse as m2<-lm(EGGS~DENSITY+SEASON)). Are the p-values the same as the ANOVA table with the interaction term?

Let's examine the fitted values and make a plot of our predictions. Note how we can use `augment` and `dplyr:::summarise` to get the fitted values (look at the help for them!). By using augment instead of predict we get the fitted.values, which are the same as predictions (see above), already tabulated and sorted with their original treatments.

```
#install.packages("broom")
require(broom)
```

```
## Loading required package: broom
```

```
m1_augmented <- augment(m1)

require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
season_means <- m1_augmented %>%
  group_by(SEASON) %>%
  summarise(mean_fitted=mean(.fitted))

print(season_means)
```

```
## # A tibble: 2 x 2
##   SEASON mean_fitted
##   <fctr>       <dbl>
## 1 spring     1.83975
## 2 summer     1.10375
```

```
density_means <- m1_augmented %>%
  group_by(DENSITY) %>%
  summarise(mean_fitted=mean(.fitted))

print(density_means)
```

```
## # A tibble: 4 x 2
##   DENSITY mean_fitted
##     <int>       <dbl>
## 1       8   2.0012823
## 2      15   1.7766322
## 3      30   1.2952392
## 4      45   0.8138462
```

```
density_season_means <- m1_augmented %>%
  group_by(DENSITY,SEASON) %>%
  summarise(mean_fitted=mean(.fitted))

print(density_season_means)
```

```
## # A tibble: 8 x 3
## # Groups:   DENSITY [?]
##   DENSITY SEASON mean_fitted
##     <int> <fctr>       <dbl>
## 1       8 spring   2.3949692
## 2       8 summer   1.6075953
## 3      15 spring   2.1594217
## 4      15 summer   1.3938428
## 5      30 spring   1.6546769
## 6      30 summer   0.9358016
## 7      45 spring   1.1499321
## 8      45 summer   0.4777604
```

```
require(tidyr)
```

```
## Loading required package: tidyr
```

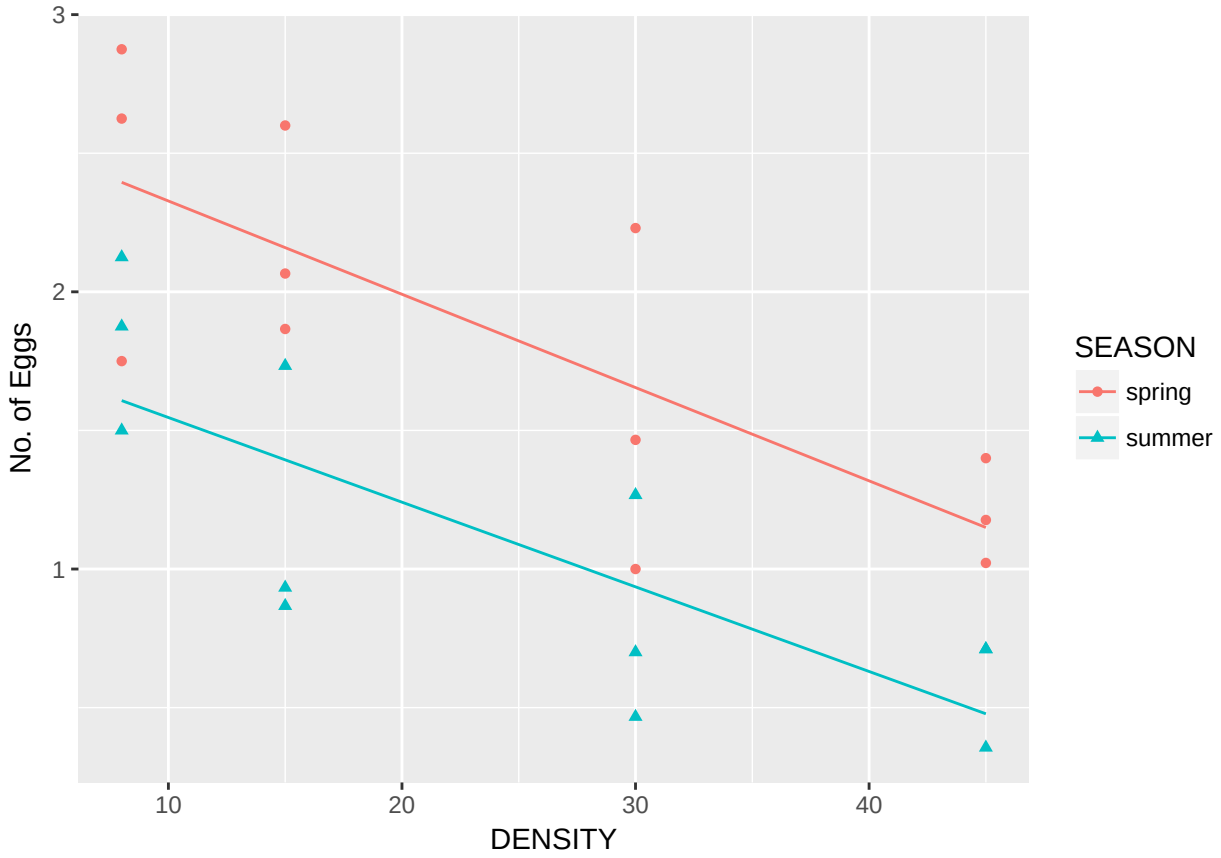
```
spread(density_season_means,
       key=SEASON,
       value=mean_fitted)
```

```
## # A tibble: 4 x 3
## # Groups:   DENSITY [4]
##   DENSITY   spring    summer
## *   <int>    <dbl>     <dbl>
## 1       8 2.394969 1.6075953
## 2      15 2.159422 1.3938428
## 3      30 1.654677 0.9358016
## 4      45 1.149932 0.4777604
```

What does `spread` do?

We can now make a final plot of the predicted/fitted values from the linear model and compare this to the original raw data. Sometimes useful for presenting your results in papers and presentations and checking model fit by eye.

```
p <- qplot(data=m1_augmented,
           x=DENSITY,
           y=.fitted,
           ylab="No. of Eggs",
           colour=SEASON,
           shape=SEASON,
           geom=c("line"))+
  geom_point(aes(y=EGGS))

print(p)
```



How does this plot help you interpret your data and its associated "best" model? Ask yourself - Is the density dependence in fecundity the same in each season?

Now do the same to compare between two models - one with and one without an interaction between DENSITY and SEASON HINT - build two models (m1 and m2) - use ggplot to show the raw data and model fits (as above) - use cowplot to place the plot for each model into a set of panels/grids

Soay Sheep Data Exploration

Import the file soay2.csv. Call it sheep.

Use `str` to explore your data. The details of the data will be put up on the board.

```
str(sheep)
```

```
## 'data.frame':    501 obs. of  9 variables:
##  $ NAME        : Factor w/ 466 levels "NG020","NG039",..: 1 2 3 4 5 6 6 7 8 9 ...
##  $ YEARf       : int  93 93 93 93 93 93 93 93 93 93 ...
##  $ AGE         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ SEX         : Factor w/ 2 levels "F","M": 1 1 2 2 2 2 2 1 1 1 ...
##  $ WEIGHT      : num  19 15 22 18.4 18.8 25.4 25.4 15.6 20.4 17.2 ...
##  $ Testosterone: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ ODIN        : num  1.715 0.594 1.75 1.054 1.957 ...
##  $ STR         : int  0 100 300 300 1200 1000 1000 0 0 300 ...
##  $ SURV1       : int  1 1 1 1 1 1 1 1 1 1 ...
```

Produce a histogram of WEIGHT, Testosterone, ODIN, STR and SURV1 in a graph with 6 panels.

Part of the a plot produced by ggpairs is a density plot akin to a histogram.

```
if(!require(GGally)){install.packages("GGally")}
```

```
## Loading required package: GGally
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
require(GGally)
ggpairs(sheep[,names(sheep) %in% c("WEIGHT", "Testosterone", "ODIN", "STR" ,"SURV1")])
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 230 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 22 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 81 rows containing missing values
```

```
## Warning: Removed 230 rows containing missing values (geom_point).
```

```
## Warning: Removed 230 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 250 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 230 rows containing missing values
```

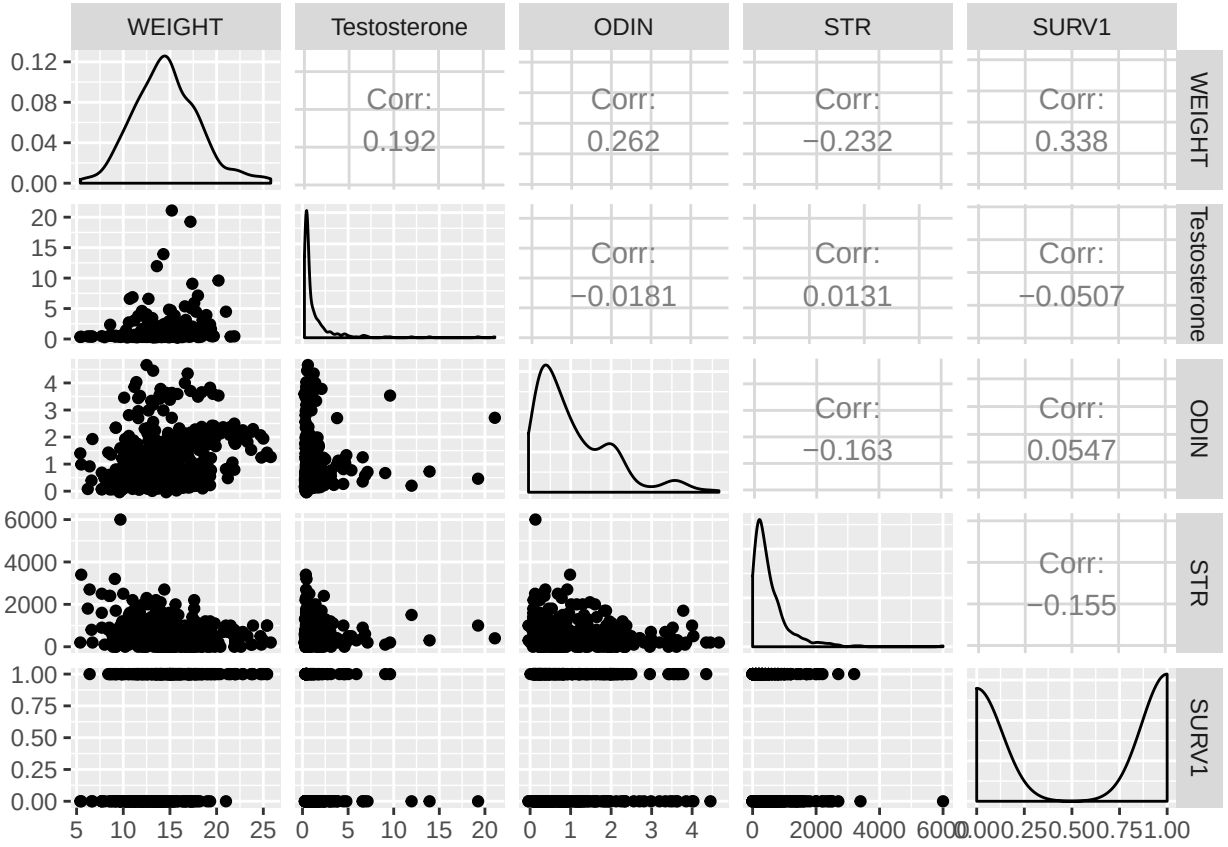
```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 263 rows containing missing values
```

```
## Warning: Removed 22 rows containing missing values (geom_point).
```

```
## Warning: Removed 250 rows containing missing values (geom_point).
```

```
## Warning: Removed 21 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 21 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 98 rows containing missing values

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 230 rows containing missing values (geom_point).

## Warning: Removed 21 rows containing missing values (geom_point).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 80 rows containing missing values

## Warning: Removed 81 rows containing missing values (geom_point).

## Warning: Removed 263 rows containing missing values (geom_point).

## Warning: Removed 98 rows containing missing values (geom_point).

## Warning: Removed 80 rows containing missing values (geom_point).

## Warning: Removed 80 rows containing non-finite values (stat_density).
```

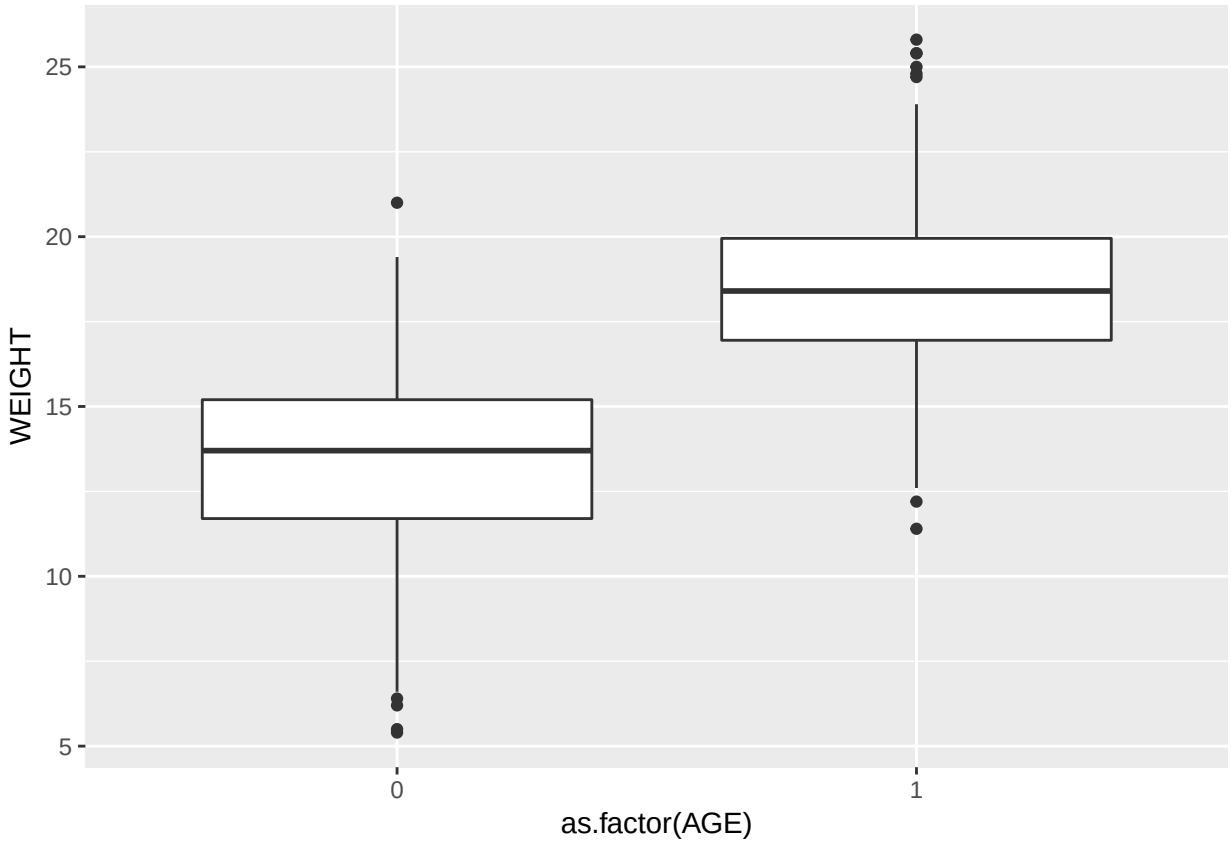


Is WEIGHT approximately normally distributed? you can use base plot to explore the raw data HINT - par(mfrow=c(2,3)) #opens a plot window with 2 rows and 3 columns then plot each data e.g. hist(sheep$WEIGHT)... and so on until you have 6 plots

Linear regression Now let's do an exploratory linear regression analysis. The question we are asking is "What factors influence body weight in Soay sheep?" Our goal is to explore, using linear regression, whether sex age and parasite load influence body weight. Start with plots of the data again Is AGE a factor or a Covariate?

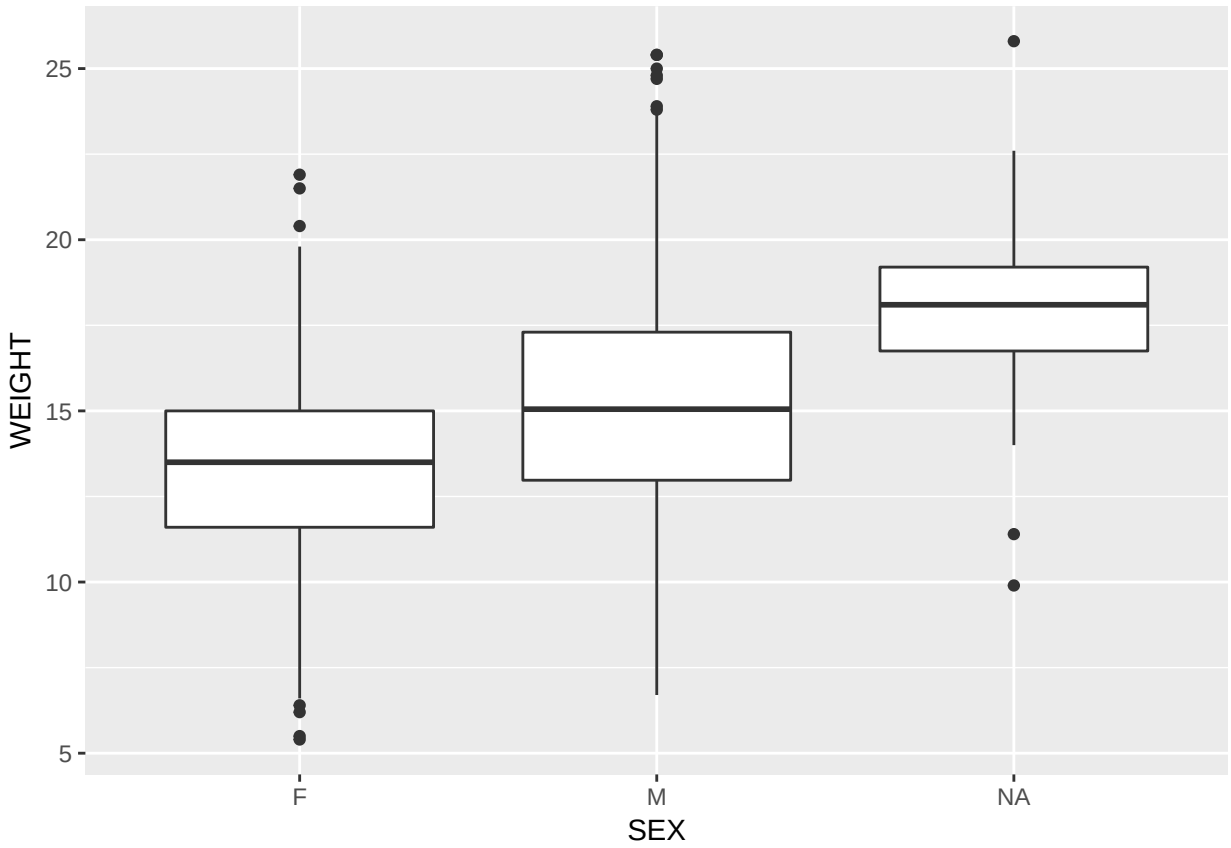
```
p <- qplot(data=sheep,
           y=WEIGHT,
           x=as.factor(AGE),
           geom="boxplot")
print(p)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```



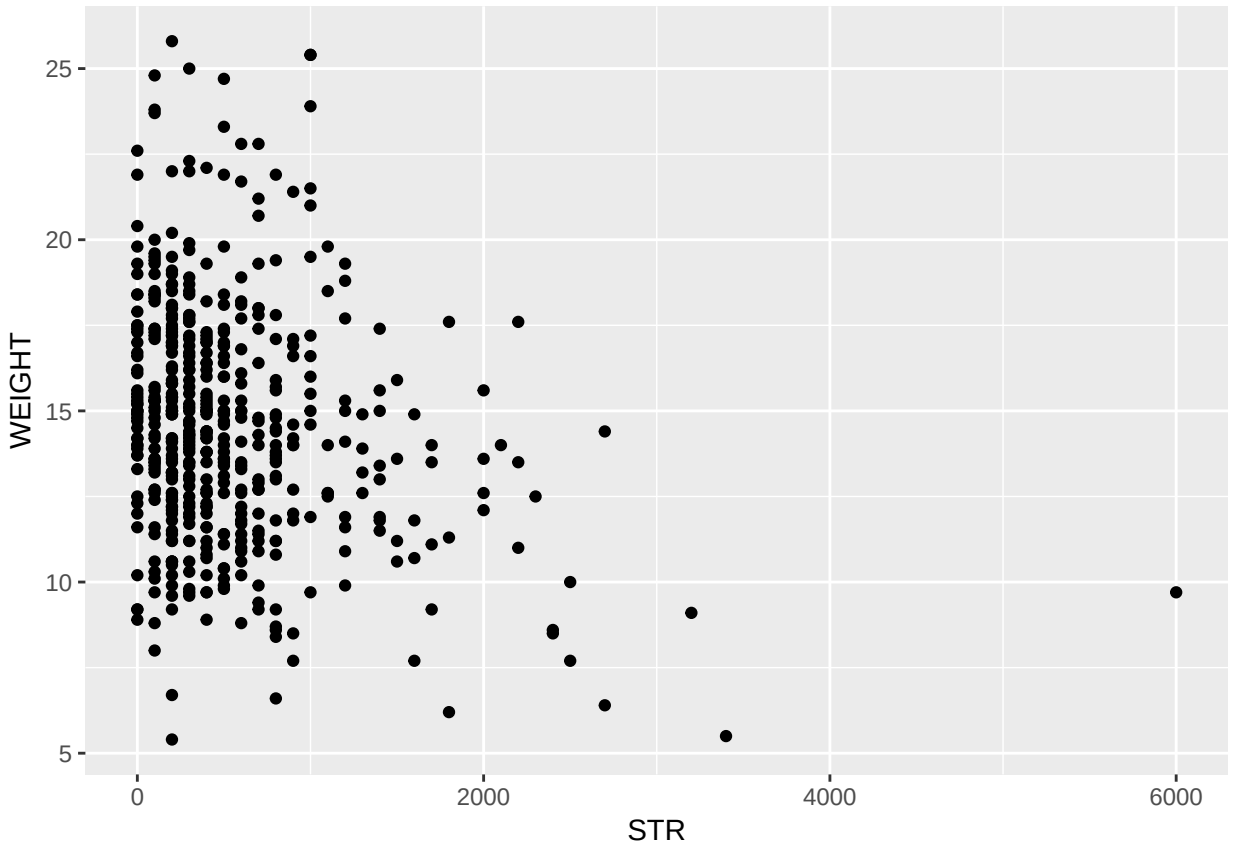

```
p <- qplot(data=sheep,
           y=WEIGHT,
           x=SEX,
           geom="boxplot")
print(p)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```




```
p <- qplot(data=sheep,
           y=WEIGHT,
           x=STR)
print(p)
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



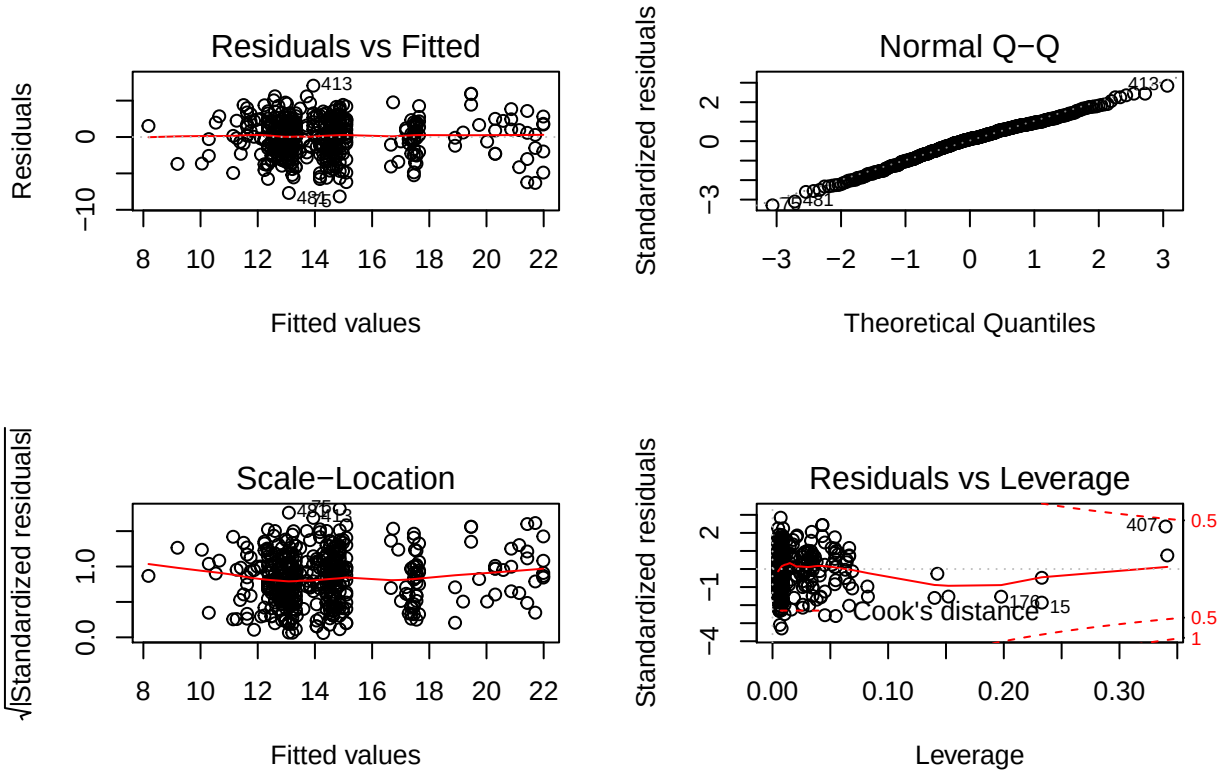

```
m1 <- lm(WEIGHT~factor(AGE)*SEX*STR,
         data=sheep)
```

Note how `names(m1)` produced a list of component pieces to the object, defined by my lm. We can use these objects to reproduce some of the diagnostic plots. There are a number of ways to do this below but in this instance baseR plot(modelname) is probably the easiest

```
names(m1)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "na.action"     "contrasts"     "xlevels"       "call"
## [13] "terms"         "model"
```

```
par(mfrow=c(2,2))
plot(m1)
```




```
if(!require(ggfortify)){install.packages("ggfortify")}
```

```
## Loading required package: ggfortify
```

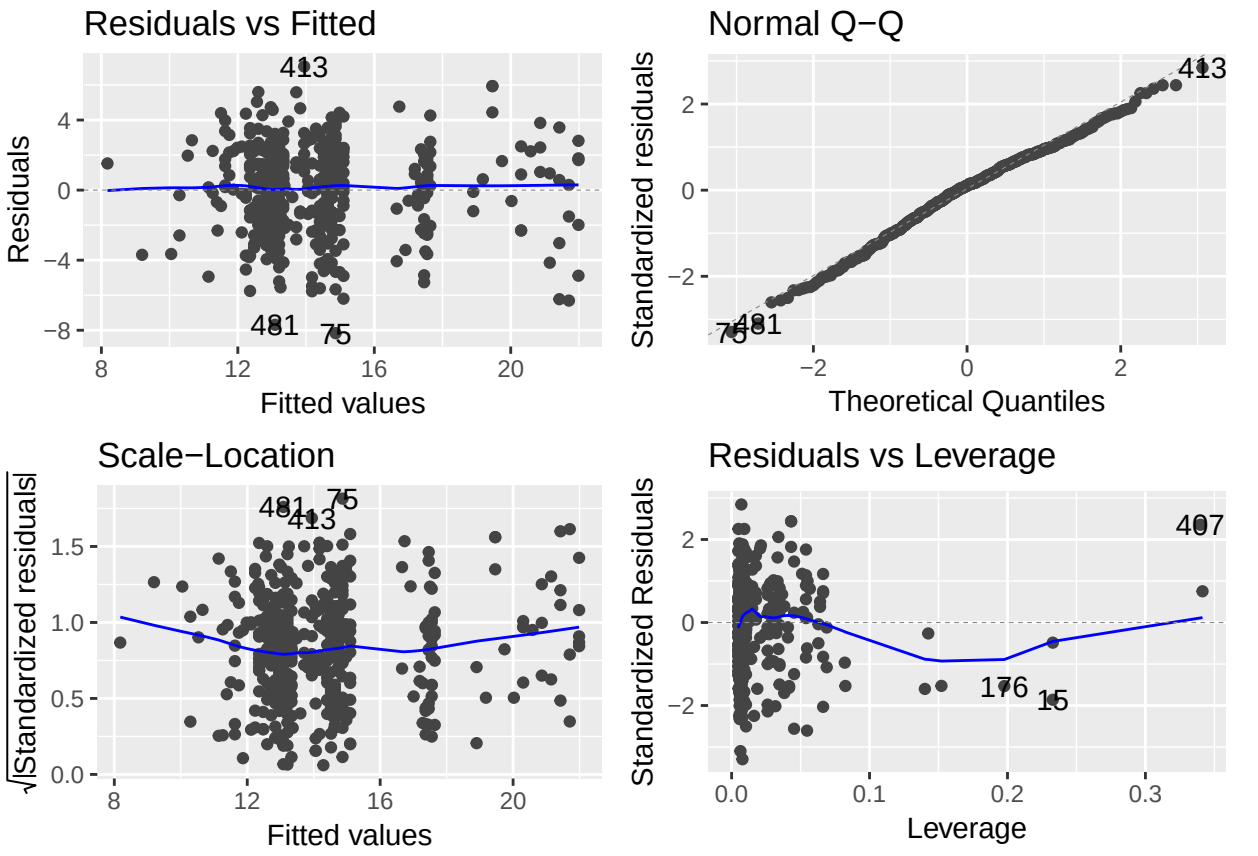
```
## Warning: namespace 'DBI' is not available and has been replaced
## by .GlobalEnv when processing object 'collapse'
```

```
## Warning: namespace 'DBI' is not available and has been replaced
## by .GlobalEnv when processing object 'collapse'
```

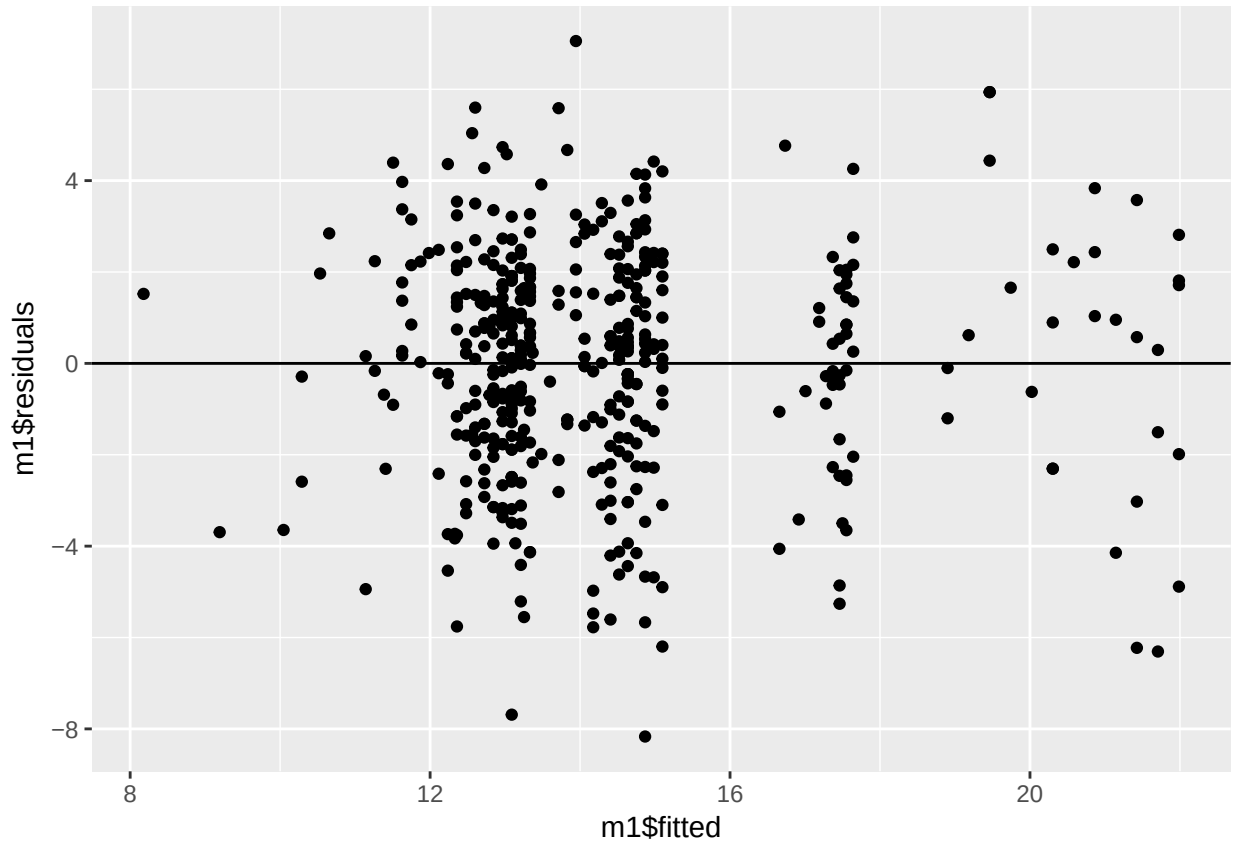
```
require(ggfortify)
if(!require(bindrcpp)){install.packages("bindrcpp")}
```

```
## Loading required package: bindrcpp
```

```
require(bindrcpp)
autoplot(m1)
```




```
p <- qplot(y=m1$residuals,
           x=m1$fitted)+
  geom_hline(yintercept = 0)
print(p)
```



As above, we use anova and summary to explore this model

```
anova(m1)
```

```
## Analysis of Variance Table
##
## Response: WEIGHT
##                      Df  Sum Sq Mean Sq  F value    Pr(>F)
## factor(AGE)           1 1719.05 1719.05 277.2729 < 2.2e-16 ***
## SEX                   1  391.41  391.41  63.1326 1.565e-14 ***
## STR                   1  236.49  236.49  38.1447 1.470e-09 ***
## factor(AGE):SEX       1   43.57   43.57   7.0279  0.008308 **
## factor(AGE):STR       1   16.61   16.61   2.6793  0.102356
## SEX:STR               1    0.01    0.01   0.0018  0.965796
## factor(AGE):SEX:STR   1    5.71    5.71   0.9215  0.337588
## Residuals           450 2789.93    6.20
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m1)
```

```
##
## Call:
## lm(formula = WEIGHT ~ factor(AGE) * SEX * STR, data = sheep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1674 -1.6051  0.2602  1.7329  7.0551
##
```

```
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           13.3321950  0.2370810  56.235  < 2e-16 ***
## factor(AGE)1           4.3106971  0.6249103   6.898 1.79e-11 ***
## SEXM                   1.7658136  0.3477047   5.078 5.58e-07 ***
## STR                   -0.0012170  0.0003170  -3.839 0.000141 ***
## factor(AGE)1:SEXM      2.8583150  0.9761075   2.928 0.003581 **
## factor(AGE)1:STR       0.0003090  0.0018420   0.168 0.866867
## SEXM:STR               0.0000639  0.0004160   0.154 0.877998
## factor(AGE)1:SEXM:STR -0.0019600  0.0020417  -0.960 0.337588
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.49 on 450 degrees of freedom
##   (43 observations deleted due to missingness)
## Multiple R-squared:  0.4638, Adjusted R-squared:  0.4554
## F-statistic:  55.6 on 7 and 450 DF,  p-value: < 2.2e-16
```

Now, we are faced with some interesting decisions. As we are exploring this data, one might argue that we are not necessarily looking for the best predictive model (adding terms to the model will invariably increase the R2 (explanatory power) and make it "better" at predicting). Instead, we use the philosophy of the "minimum adequate model" to seek out the indivisual variables that explain "significant" amounts of variance.

To do this, we begin by looking at the highest order terms in the model – in this case the 3-way interaction. Because 1) our anova table is sequential and we can only trust the p-values on the highest order terms, and 2) because everything above this in the table is marginal, we ask the very simple question – is the 3way significant? If the answer is no, than removing it from the model makes no significant change in our explanation of variance. Remember this principle.

We can use a trick in R to update our model and make sure that our interpretation is correct.

```
m2 <- update(m1, ~.-factor(AGE):SEX:STR)
anova(m2)
```

```
## Analysis of Variance Table
##
## Response: WEIGHT
##                   Df  Sum Sq Mean Sq  F value    Pr(>F)
## factor(AGE)        1 1719.05 1719.05 277.3212 < 2.2e-16 ***
## SEX                1  391.41  391.41  63.1436 1.551e-14 ***
## STR                1  236.49  236.49  38.1513 1.463e-09 ***
## factor(AGE):SEX    1   43.57   43.57   7.0291  0.008302 **
## factor(AGE):STR    1   16.61   16.61   2.6798  0.102325
## SEX:STR            1    0.01    0.01   0.0018  0.965793
## Residuals        451 2795.65    6.20
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m1,m2)
```

```
## Analysis of Variance Table
##
## Model 1: WEIGHT ~ factor(AGE) * SEX * STR
## Model 2: WEIGHT ~ factor(AGE) + SEX + STR + factor(AGE):SEX + factor(AGE):STR +
##     SEX:STR
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    450 2789.9
```

```
## 2    451 2795.7 -1   -5.7134 0.9215 0.3376
```

m2 is now a model formed, using the command update(), without the 3 way interaction. We look at the new anova table to confirm we lost it. We can then use anova(m1,m2) to compare the model, using an F-test, to determine whether one explains more variation than the other. It tests the change in the sums of squares against the F-distribution. Compare this p-value to the one in the anova table for model 1 above. Is it the same? Now, we are stuck with a rather difficult prospect. Model 2 has three 2-way interactions in it. Each of these is in the highest order category now (2-way). Moreover, as the table is sequential, the only p-value that we can trust is the last one, for SEX:STR. How do we cope?

We could, if we had the time and inclination, rewrite our model three times, each time, placing one of the 2-way terms at the end of our model description. Or, as we saw above, we could create 3 models, each missing one of the 2-way variables, and use anova() to compare the two. A significant p-value on any of the comparisons would indicate that indeed, the term is significant and important. These are called single-term-deletion tests. Not surprisingly there is an easier way: dropterm() from the MASS library. This function implements the single-term-deletions. You therefore need to load MASS.

```
require(MASS)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
dropterm(m2, test="F")
```

```
## Single term deletions
##
## Model:
## WEIGHT ~ factor(AGE) + SEX + STR + factor(AGE):SEX + factor(AGE):STR +
##     SEX:STR
##                 Df Sum of Sq    RSS    AIC F Value    Pr(F)
## <none>                       2795.7 842.50
## factor(AGE):SEX  1    58.250 2853.9 849.94  9.3970 0.002304 **
## factor(AGE):STR  1    16.246 2811.9 843.15  2.6208 0.106171
## SEX:STR          1     0.011 2795.7 840.50  0.0018 0.965793
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Q:Why does dropterm() only report on the higher order interactions? Q:Note the use of AIC - do you know what it is and how to read it? Q:Which terms can we consider dropping?

Let's begin with the most insignificant term which also has the lowest AIC, and work our way down the chain of significance and order of interactions

```
m3 <- update(m2, ~.-SEX:STR)
dropterm(m3,test="F")
```

```
## Single term deletions
##
## Model:
## WEIGHT ~ factor(AGE) + SEX + STR + factor(AGE):SEX + factor(AGE):STR
##                 Df Sum of Sq    RSS    AIC F Value    Pr(F)
## <none>                       2795.7 840.50
## factor(AGE):SEX  1    59.743 2855.4 848.18  9.6592 0.002003 **
```

```
## factor(AGE):STR  1    16.612 2812.3 841.21  2.6857 0.101946
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m4 <- update(m3, ~.-factor(AGE):STR)
dropterm(m4,test="F")
```

```
## Single term deletions
##
## Model:
## WEIGHT ~ factor(AGE) + SEX + STR + factor(AGE):SEX
##                 Df Sum of Sq    RSS    AIC F Value     Pr(F)
## <none>                       2812.3 841.21
## STR              1   253.528 3065.8 878.75  40.838 4.113e-10 ***
## factor(AGE):SEX  1    43.572 2855.8 846.26   7.019  0.008349 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice now that we have the minimum explanatory model. Our deletions of the 2-way terms left us with only one significant higher order term: age*sex. This left parasite load behind, and as it is not involved in an interaction, the main effect is the highest order term for parasite load. We have detected significant effects of parasite load on Weight and of age and sex on weight. Using summary(m4) we can identify that increasing parasite load, controlling for sex and age, causes decreases in weight.

```
summary(m4)
```

```
##
## Call:
## lm(formula = WEIGHT ~ factor(AGE) + SEX + STR + factor(AGE):SEX,
##     data = sheep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2181 -1.6295  0.2323  1.7511  7.0959
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       13.3579455  0.2006654  66.568  < 2e-16 ***
## factor(AGE)1       4.3678730  0.4384794   9.961  < 2e-16 ***
## SEXM               1.8136110  0.2556621   7.094 5.05e-12 ***
## STR               -0.0012674  0.0001983  -6.390 4.11e-10 ***
## factor(AGE)1:SEXM  1.7077476  0.6446152   2.649  0.00835 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.492 on 453 degrees of freedom
##   (43 observations deleted due to missingness)
## Multiple R-squared:  0.4595, Adjusted R-squared:  0.4547
## F-statistic: 96.27 on 4 and 453 DF,  p-value: < 2.2e-16
```

Finally we can use predicted values from the model to explore the sex*age interaction! First, we must declare a parasite load at which to make the prediction. Then we build a data frame for prediction. Then we use this data frame.

```
newdat <- with(sheep,
  expand.grid(AGE=levels(factor(AGE)),
              SEX=levels(SEX),
```

```
              STR=mean(STR)))
```

Note how expand.grid makes a minimal dataset of values on which to predict.

```
pd <- predict(m4, newdat, se.fit=T)
pd
```

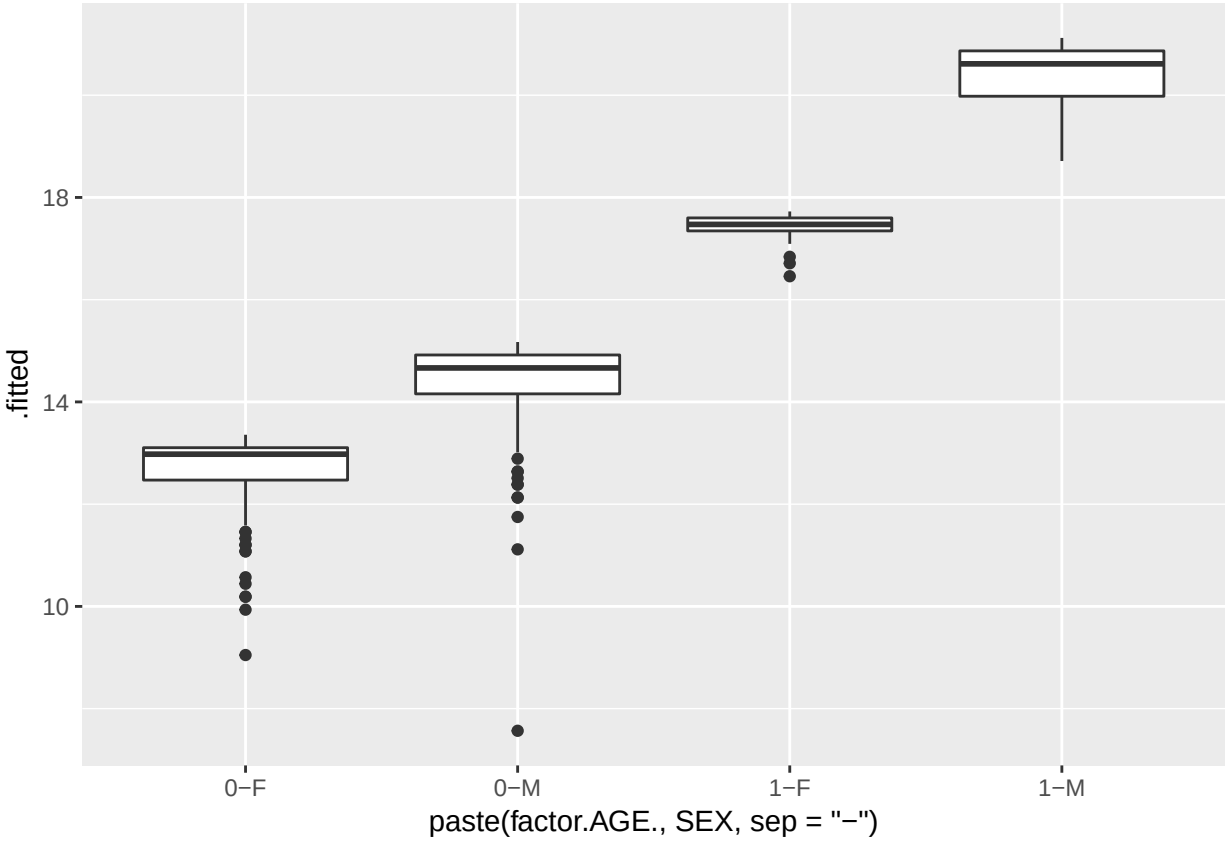
```
## $fit
##        1        2        3        4
## 12.67743 17.04530 14.49104 20.56666
##
## $se.fit
##         1         2         3         4
## 0.1732542 0.4035703 0.1874287 0.4344602
##
## $df
## [1] 453
##
## $residual.scale
## [1] 2.491605
```

Now we make a dataset for plotting. This can be done with values from `predict` or using `augment` in `broom`.

```
require(broom)
augment_m4 <- augment(m4)
```

We then plot this augmented data

```
p <- qplot(data=augment_m4,
           x=paste(factor.AGE.,SEX, sep="-"),
           y=.fitted,
           geom="boxplot")
print(p)
```



Why dont you try to generate a two panel plot (cowplot), each panel using a different parasite load.

Some people like the idea of Type III sums of squares (as you'd get in Minitab) – so that you can look at the terms' significance without doing the dropterm() or drop1(). See http://www.stats.ox.ac.uk/pub/MASS3/Exegeses.pdf for a discussion why Type IIIs are often not used. If you insist, you can use the command:

```
require(car)
```

```
## Loading required package: car
```

```
## Warning: package 'car' was built under R version 3.4.3
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
Anova(m1,type="III")
```

```
## Anova Table (Type III tests)
##
## Response: WEIGHT
##                       Sum Sq  Df   F value    Pr(>F)
## (Intercept)         19606.1   1 3162.3487 < 2.2e-16 ***
## factor(AGE)           295.0   1   47.5839 1.795e-11 ***
## SEX                   159.9   1   25.7910 5.583e-07 ***
## STR                    91.4   1   14.7372 0.0001413 ***
## factor(AGE):SEX        53.2   1    8.5748 0.0035814 **
## factor(AGE):STR         0.2   1    0.0281 0.8668668
```

```
## SEX:STR                 0.1   1    0.0236 0.8779976
## factor(AGE):SEX:STR     5.7   1    0.9215 0.3375885
## Residuals            2789.9 450
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```