

lm

general linear models in R

Natalie Cooper
Natural History Museum London
natalie.cooper@nhm.ac.uk
@nhcooper123

Preamble

- Who has used R before?
- Has everyone downloaded R, the packages etc?
- Don't be afraid to ask questions. Interrupt us! Chat to us over coffee/lunch! We are here to help (and we really love R!)
- Help each other!

Plan for today

- Quick reminder about scripts in R.
- General linear models
- Lunch
- *Generalised linear models (GLMs)* - David Orme

Why use a script?

- We can do everything in the console, why bother with a script?
 - Record of what you did and WHY (R ignores comments after #).
 - Allows you to quickly repeat the analysis and make changes.
 - The code in the console will not be saved so that you can read it. But you can save the script.

Ideal workflow...

1. Type code into the SCRIPT.
2. Add notes after # so you know what you were trying to achieve with that bit of code.
3. Run the code in the R CONSOLE (**Select it then Ctrl R or Cmd enter for Mac**).
4. If it doesn't work, edit the SCRIPT and then try to run it again.
5. Keep repeating until it works.
6. SAVE the SCRIPT. This results in a messy CONSOLE (who cares!) and a tidy perfect SCRIPT (yay!)

```

#-----#
# Create citations through time graphs
#-----#

# Clear R's brain
rm(list = ls())

# Load required libraries
library(ggplot2)
library(reshape)

# Input data
ds <- read.csv("Data/CitesThruTime.csv")
type <- read.csv("Data/PaperContents.csv")

# Reshape the data so it's in long format
data_long <- melt(ds, id = "Year")]

# Merge in data explaining what each paper contains
ds.all <- merge(data_long, type, by.x = "variable", by.y = "Paper")

# Remove Felsenstein 1985 & Harvey&Pagel 1991
ds.all <- subset(ds.all, variable != "Felsenstein1985"
  & variable != "HarveyPagel1991")

# Summarise citations for each type of assumption
dsBrownian <- with(ds.all[which(ds.all$Brownian == "Yes"), ],
  aggregate(value, by = list(Year),
    FUN = sum, na.rm = TRUE))

```

R4All Philosophy

Experimental Design	Bad Data is Bad
PLOT YOUR DATA	Theory Drives Expectation of Pattern
Make A Model	Model and Experiment Match
Evaluate Assumptions	BEFORE YOU Check the p-value
Evaluate Model Conclusions	NOW have a look at F and p
Add Interpretation to Figure	Figures are Better than Tables

General Linear Models

- One Response Variable
- Covariates and Factors
 - *continuous or discrete independent variables*
- Balanced and unbalanced designs
- *Statistically* linear: $Y \sim mX + c$
- i.e. not always a straight line!

General Linear Models

- ANOVA
- ANCOVA
- Regression
- Multiple Regression
- Non-linear via polynomials & transformations

Regression, ANOVA or ANCOVA?

- Nitrogen concentration versus phosphorus concentration in a lake?
- Growth rates of two different species of fish in response to temperature?
- Seagrass biomass on two rocky shores with different exposures?

Regression, ANOVA or ANCOVA?

- Nitrogen concentration versus phosphorus concentration in a lake? **REGRESSION**
- Growth rates of two different species of fish in response to temperature? **ANCOVA**
- Seagrass biomass on two rocky shores with different exposures? **ANOVA**

Regression, ANOVA or ANCOVA?

- Nitrogen concentration versus phosphorus concentration in a lake? **REGRESSION.**
SCATTER PLOT
- Growth rates of two different species of fish in response to temperature? **ANCOVA.** **SCATTER PLOT WITH TWO LINES**
- Seagrass biomass on two rocky shores with different exposures? **ANOVA.** **BOXPLOT/ BARCHART**

Regression

Simple Linear Regression

- equation for a line
 - $y = b + m * x$
- coefficients
 - b & m
 - intercept and slope/gradient
- In “R-speak”: $y \sim x$

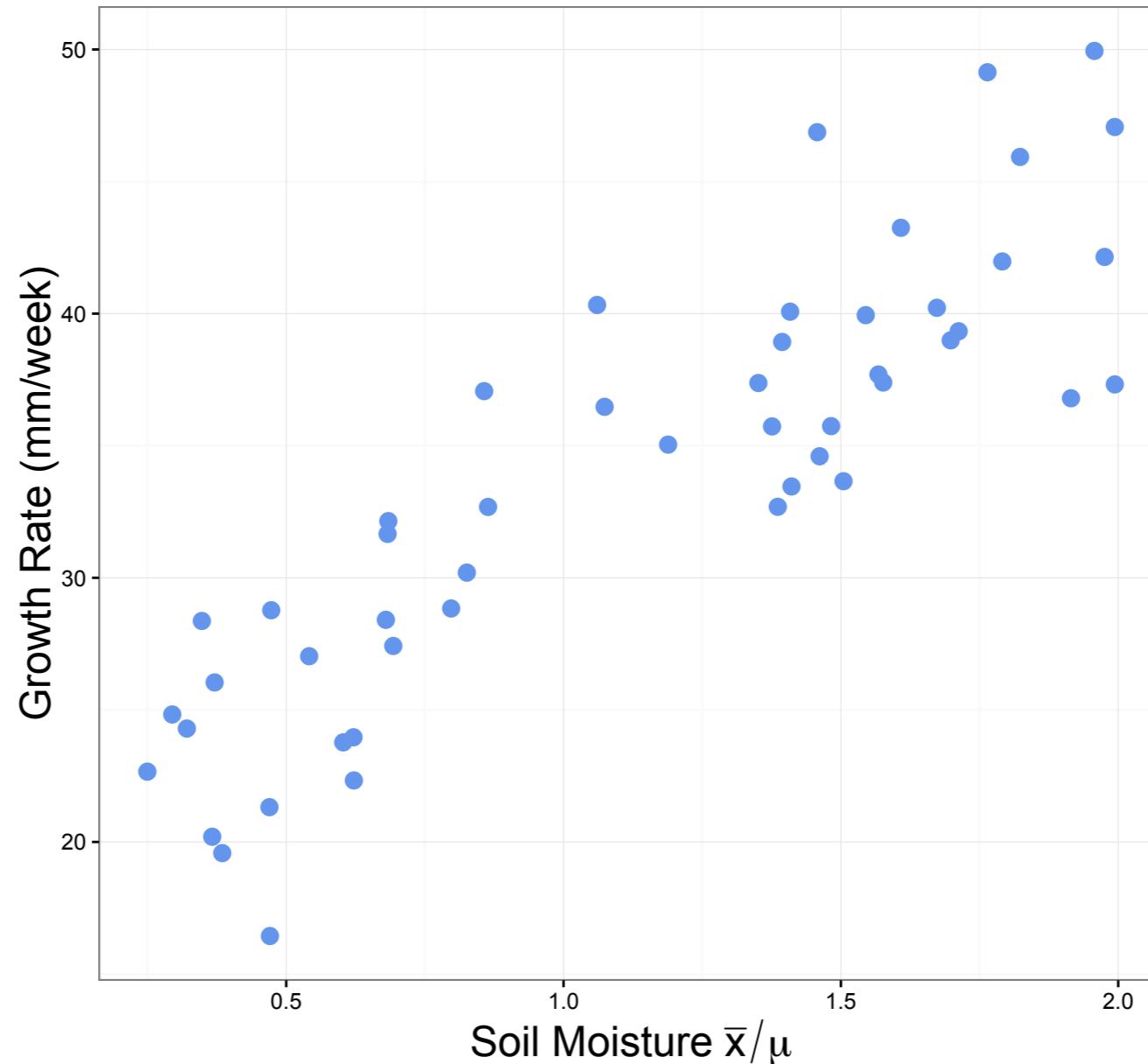
Plant growth rate data



R4All Philosophy

Experimental Design	Bad Data is Bad
PLOT YOUR DATA	Theory Drives Expectation of Pattern
Make A Model	Model and Experiment Match
Evaluate Assumptions	BEFORE YOU Check the p-value
Evaluate Model Conclusions	NOW have a look at F and p
Add Interpretation to Figure	Figures are Better than Tables

1. Import the data in R (plant.growth.rate.csv)
2. Make a scatter plot
3. Guess what the slope and intercept might be



Fitting the model using lm()

```
model1 <- lm(plant.growth.rate ~  
soil.moisture, data = mydata)
```

R4All Philosophy

Experimental Design	Bad Data is Bad
PLOT YOUR DATA	Theory Drives Expectation of Pattern
Make A Model	Model and Experiment Match
Evaluate Assumptions	BEFORE YOU Check the p-value
Evaluate Model Conclusions	NOW have a look at F and p
Add Interpretation to Figure	Figures are Better than Tables

Diagnostics

plot.lm {stats}

R Documentation

Plot Diagnostics for an lm Object

Description

Six plots (selectable by `which`) are currently available: a plot of residuals against fitted values, a Scale-Location plot of $\sqrt{|residuals|}$ against fitted values, a Normal Q-Q plot, a plot of Cook's distances versus row labels, a plot of residuals against leverages, and a plot of Cook's distances against leverage/(1-leverage). By default, the first three and 5 are provided.

Usage

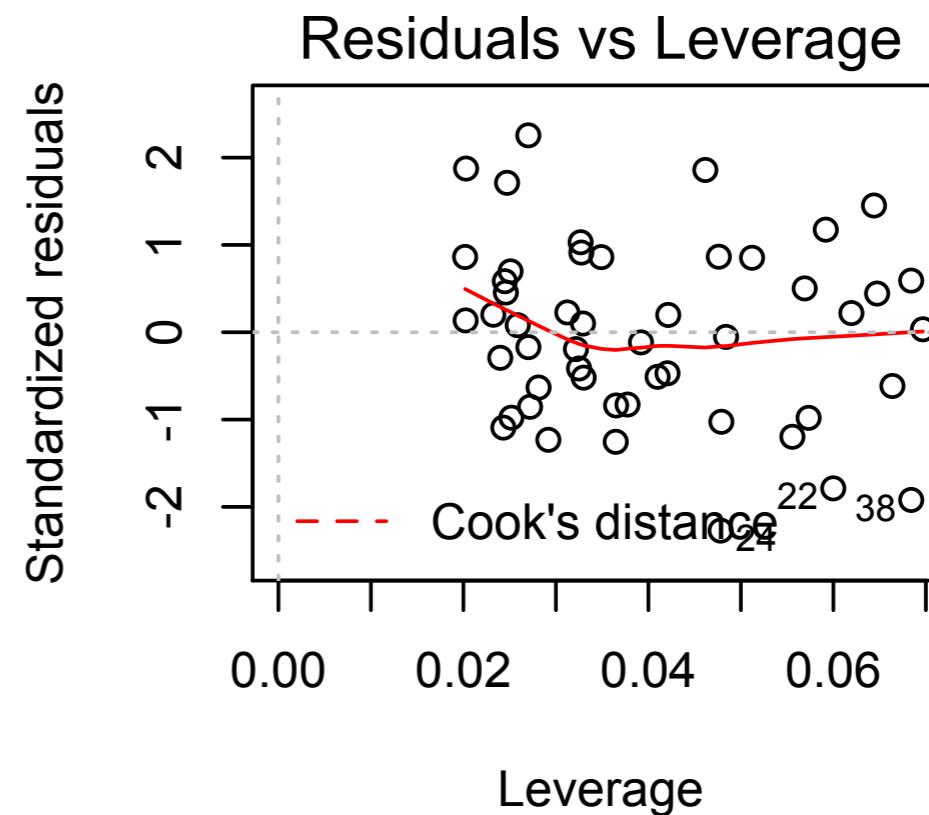
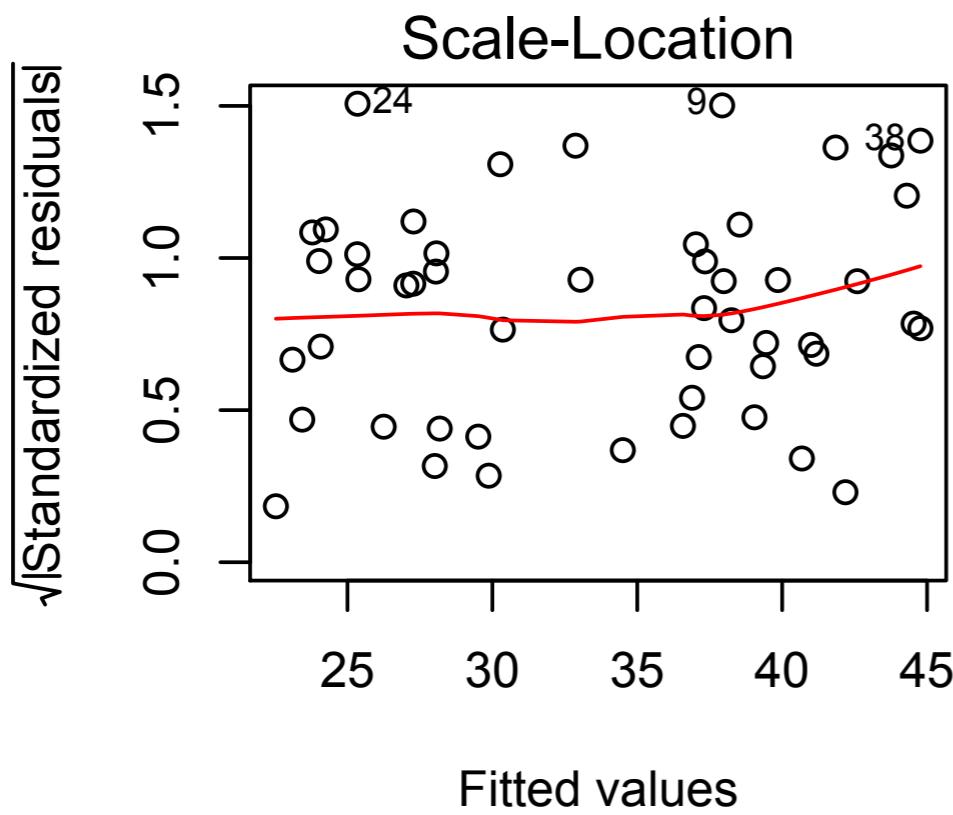
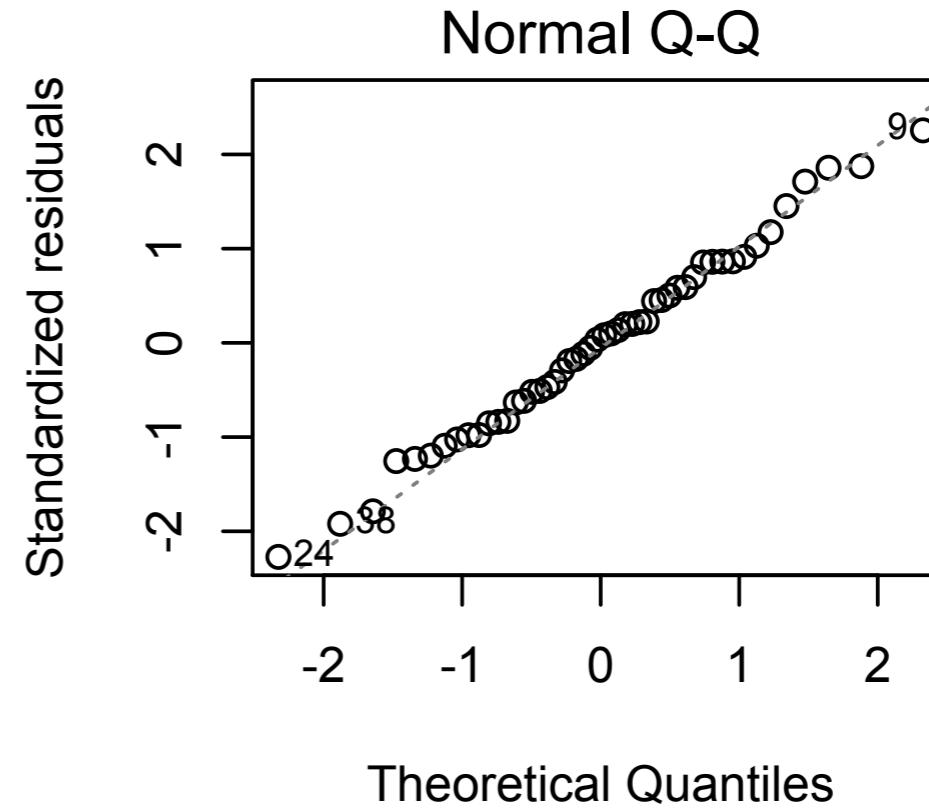
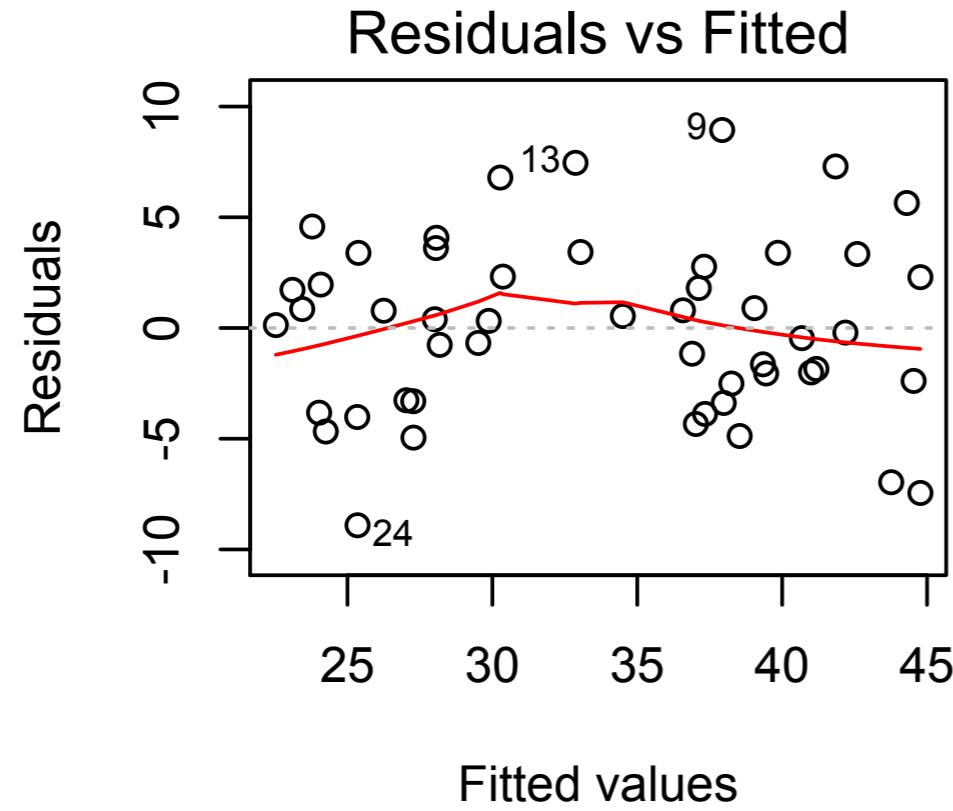
```
## S3 method for class 'lm':
plot(x, which = c(1:3,5),
      caption = list("Residuals vs Fitted", "Normal Q-Q",
                     "Scale-Location", "Cook's distance",
                     "Residuals vs Leverage",
                     expression("Cook's dist vs Leverage   " * h[ii] / (1 - h[ii]))),
      panel = if(add.smooth) panel.smooth else points,
      sub.caption = NULL, main = "",
      ask = prod(par("mfcol")) < length(which) && dev.interactive(),
      ...,
      id.n = 3, labels.id = names(residuals(x)), cex.id = 0.75,
      qqline = TRUE, cook.levels = c(0.5, 1.0),
      add.smooth = getOption("add.smooth"), label.pos = c(4,2),
      cex.caption = 1)
```

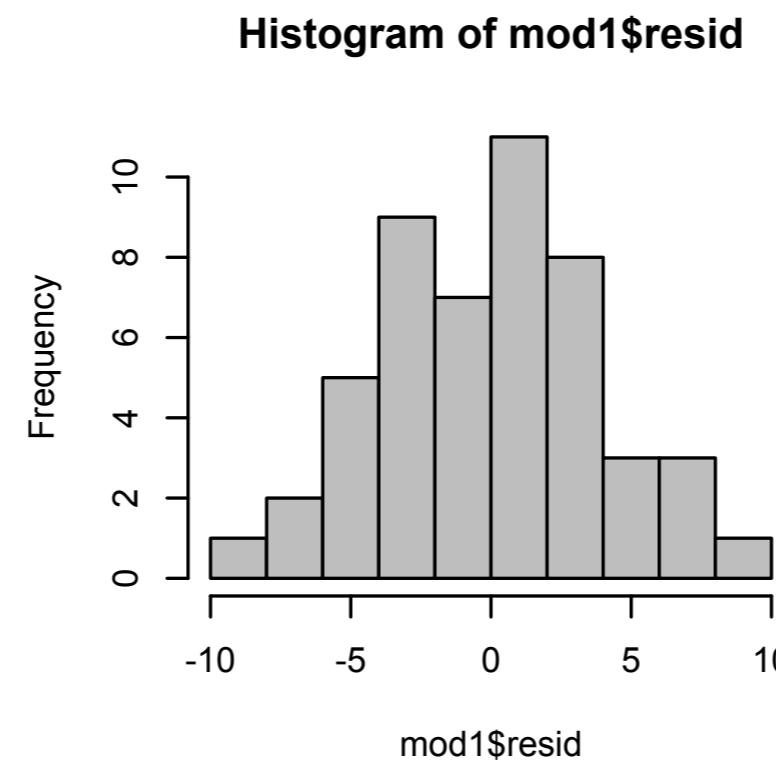
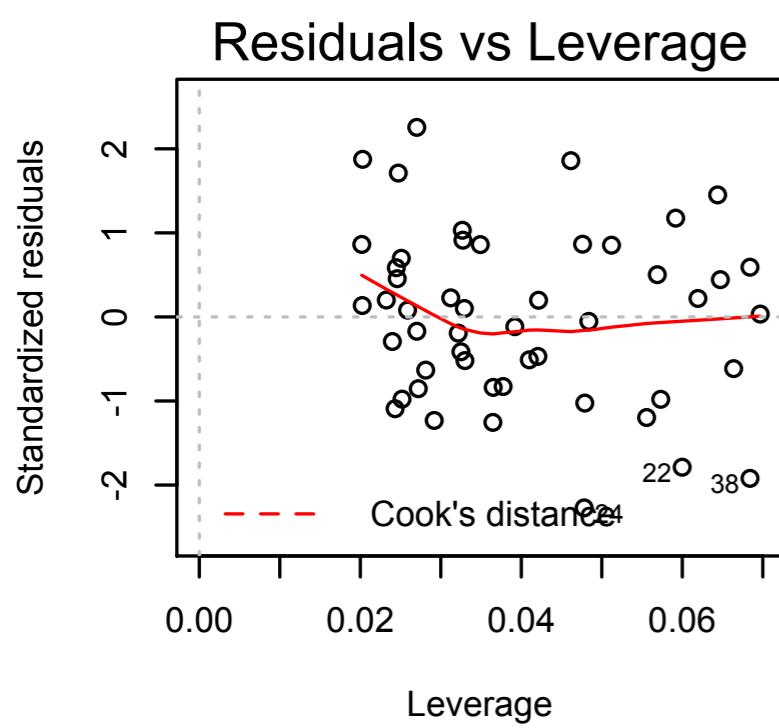
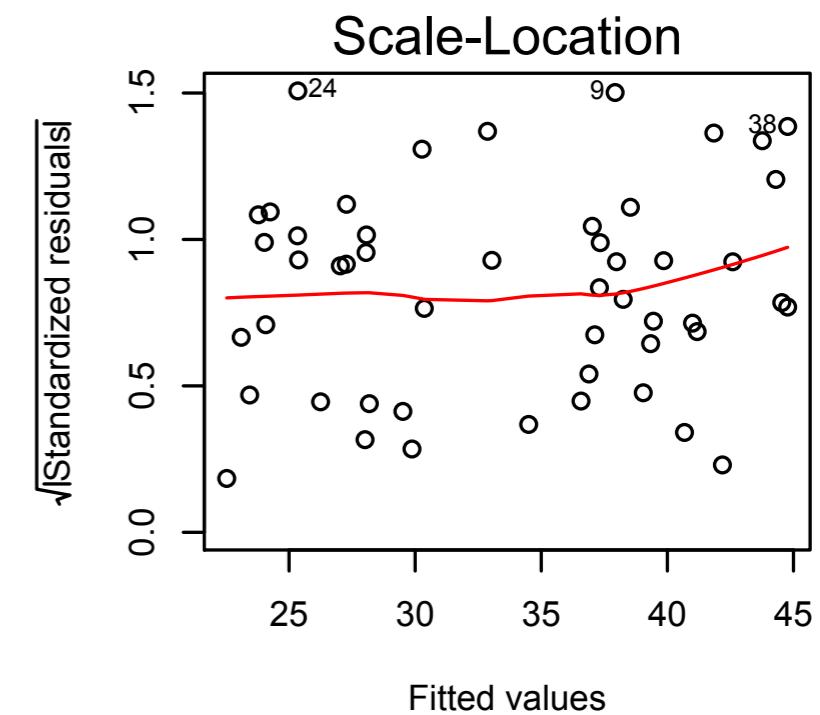
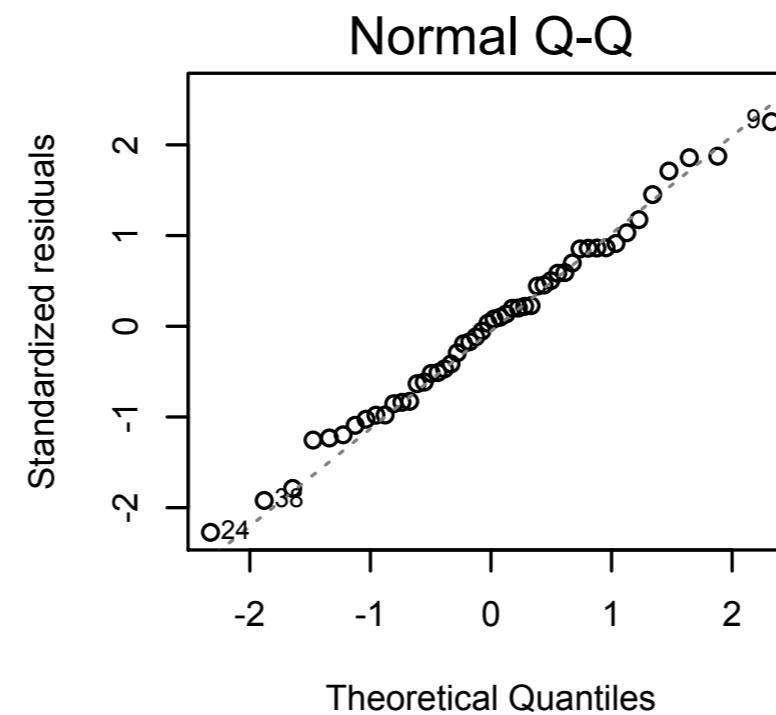
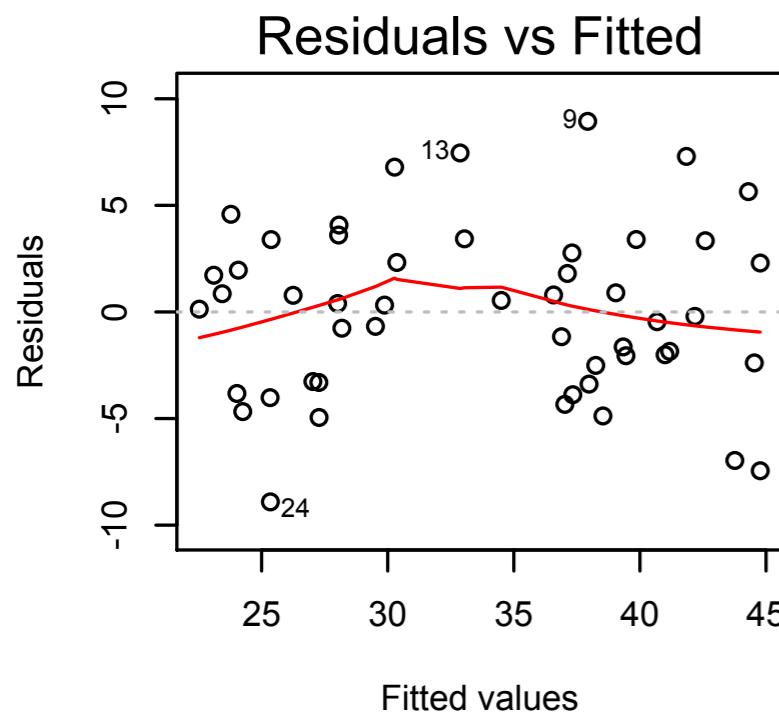
Arguments

- `x` `lm` object, typically result of [lm](#) or [glm](#).
- `which` if a subset of the plots is required, specify a subset of the numbers 1:6.
- `caption` captions to appear above the plots; [character](#) vector or [list](#) of valid graphics annotations, see [as.graphicsAnnot](#). Can be set to "" or NA to suppress all captions.
- `panel` panel function. The useful alternative to [points](#), [panel.smooth](#) can be chosen by `add.smooth = TRUE`.
- `sub.caption` common title—above the figures if there are more than one; used as `sub(s.title)` otherwise. If `NULL`, as by default, a possible abbreviated version of `deparse(x$call)` is used.
- `main` title to each plot—in addition to `caption`.
- `ask` logical; if `TRUE`, the user is *asked* before each plot, see [par\(ask=..\)](#).
- `...` other parameters to be passed through to plotting functions.

Evaluating the model assumptions...

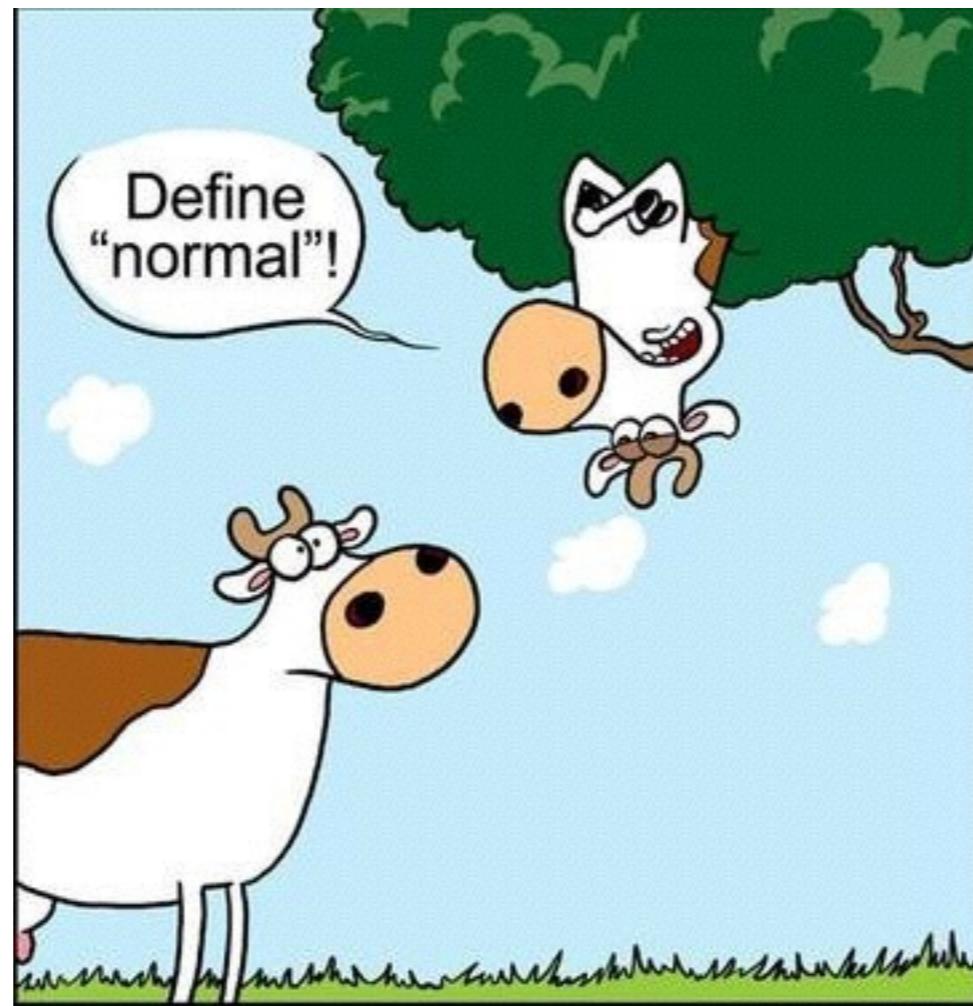
```
plot(model1)
```





Linear models - Know what matters

- Distributional assumptions (e.g. normality & equal variance) are NOT so important
 - UNLESS, your focus is prediction
- Getting the model structure is important:
 - Is the additivity assumption met?
 - Have you captured the interactions?
- The MOST IMPORTANT thing you need to do is to make sure the model addresses the research question



```

## function adapted from
## http://www.nate-miller.org/1/post/2013/03/how-normal-is-normal-a-q-q-plot-approach.html

qqfunc <- function(model, num.reps) {

  N <- length(resid(model))
  sigma <- summary(model)$sigma

  x <- rnorm(N, 0, sigma)
  xx <- qnorm(x, plot.it=F)
  xx$y <- xx$y[order(xx$x)]
  xx$x <- xx$x[order(xx$x)]
  plot(xx$x, xx$y, pch=19, col="#00000011", type="l")
  ##qqline(x)

  for(i in 2:num.reps) {

    x <- rnorm(N, 0, sigma)
    xx <- qnorm(x, plot.it=F)
    xx$y <- xx$y[order(xx$x)]
    xx$x <- xx$x[order(xx$x)]
    points(xx$x, xx$y, pch=19, col="#00000011", type="l")

  }

  xx <- qnorm(m1$residuals, plot.it=F)
  points(xx$x, xx$y, col="red", pch=19)

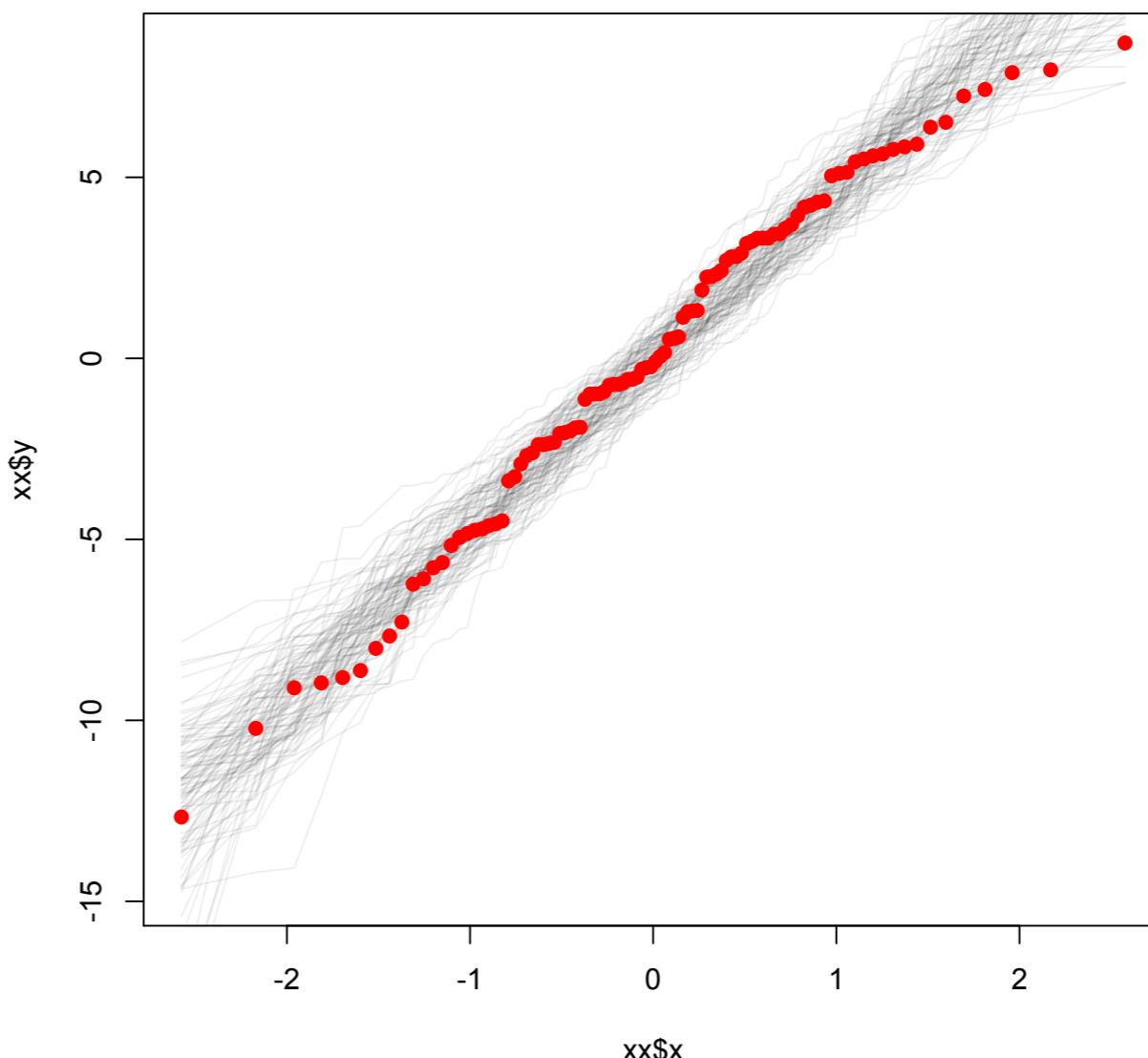
}

n <- 100
x <- seq(1, 10, length=n)
y <- 2 + 3*x + rnorm(length(x), 0, 2)
##y <- 2 + 3*x + rlnorm(length(x), 0, 2)
##y <- 2 + 3*x + 2*runif(length(x))
##y <- rpois(length(x), x)
m1 <- lm(y ~ x)
xx <- qqline(m1$residuals)

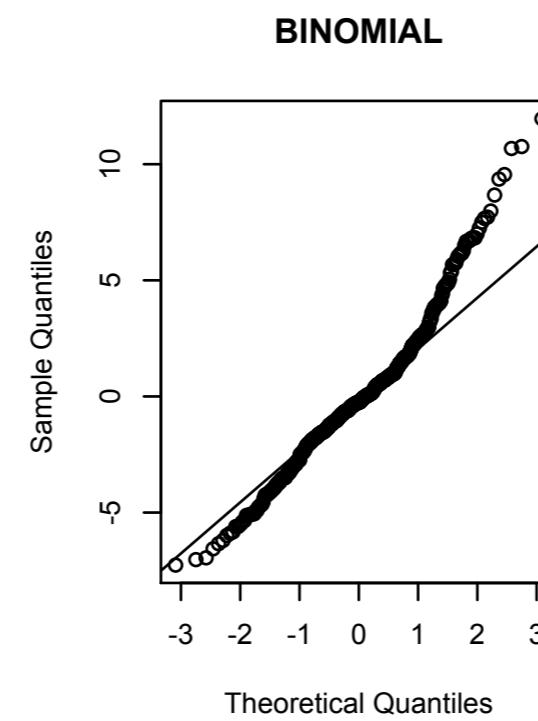
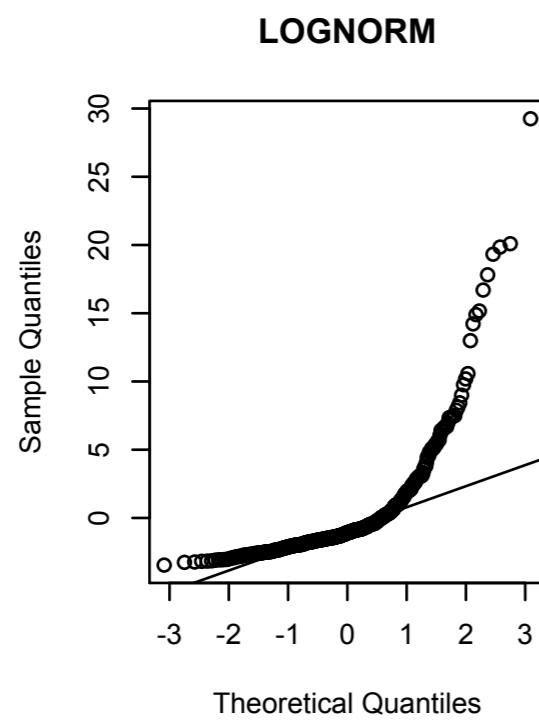
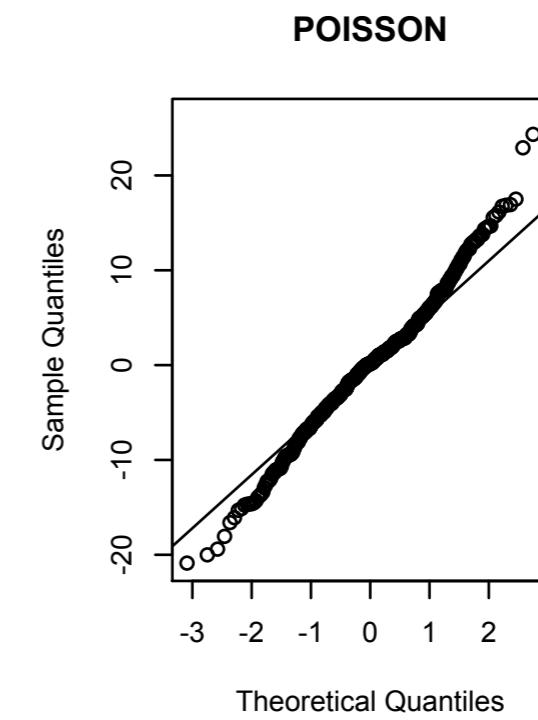
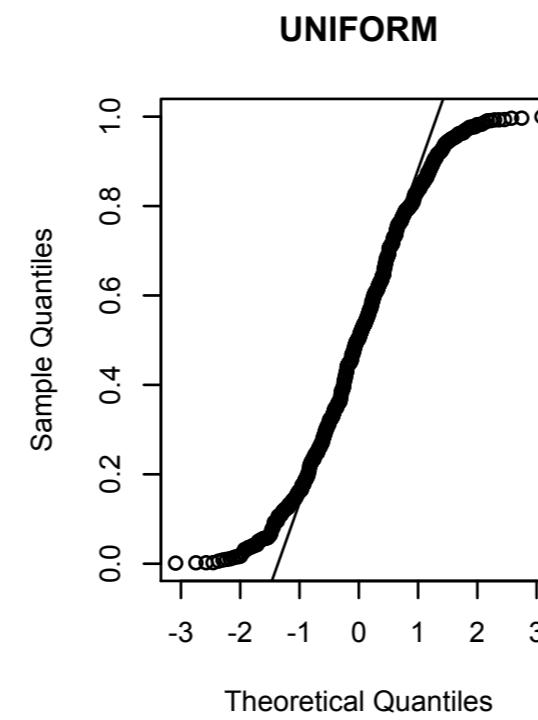
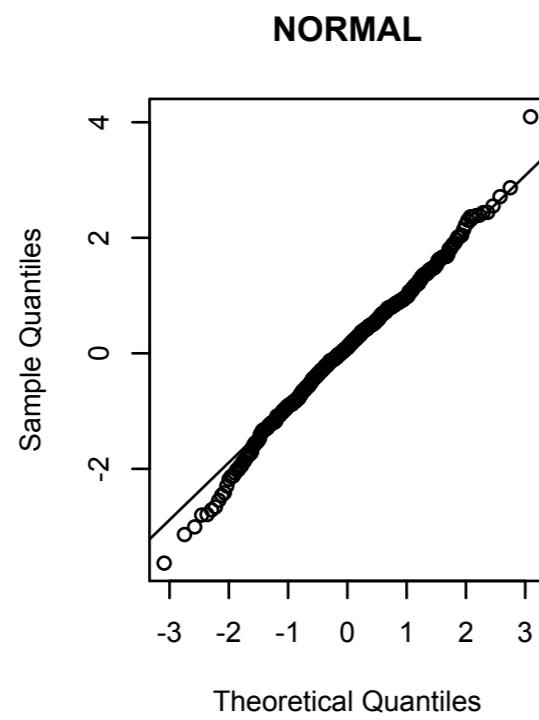
qqfunc(m1, 100)

```

A qq-plot of 100 samples of the random normal distribution



qqnorm diagnostics



R4All Philosophy

Experimental Design	Bad Data is Bad
PLOT YOUR DATA	Theory Drives Expectation of Pattern
Make A Model	Model and Experiment Match
Evaluate Assumptions	BEFORE YOU Check the p-value
Evaluate Model Conclusions	NOW have a look at F and p
Add Interpretation to Figure	Figures are Better than Tables

the workhorses

- **anova**(model.name)
 - *overall significance of main, 2-way, 3-way etc.*
 - *sequential sums of squares*
- **summary**(model.name)
 - coefficients, significance, with respect to contrasts

apply anova and
summary to your
model

R4All Philosophy

Experimental Design	Bad Data is Bad
PLOT YOUR DATA	Theory Drives Expectation of Pattern
Make A Model	Model and Experiment Match
Evaluate Assumptions	BEFORE YOU Check the p-value
Evaluate Model Conclusions	NOW have a look at F and p
Add Interpretation to Figure	Figures are Better than Tables

predictions workflow

$$y = 20 + 13 * x$$

$$\text{PGR} = 20 + 13 * \text{Soil Moisture}$$

- Make "New X"
- Make "New Y"
- Housekeeping
- Add lines to plot

NewX	NewY
0	
0.1	
0.2	
...	
2	

$$\text{biomass} = 118 - 3.9 * \text{days}$$

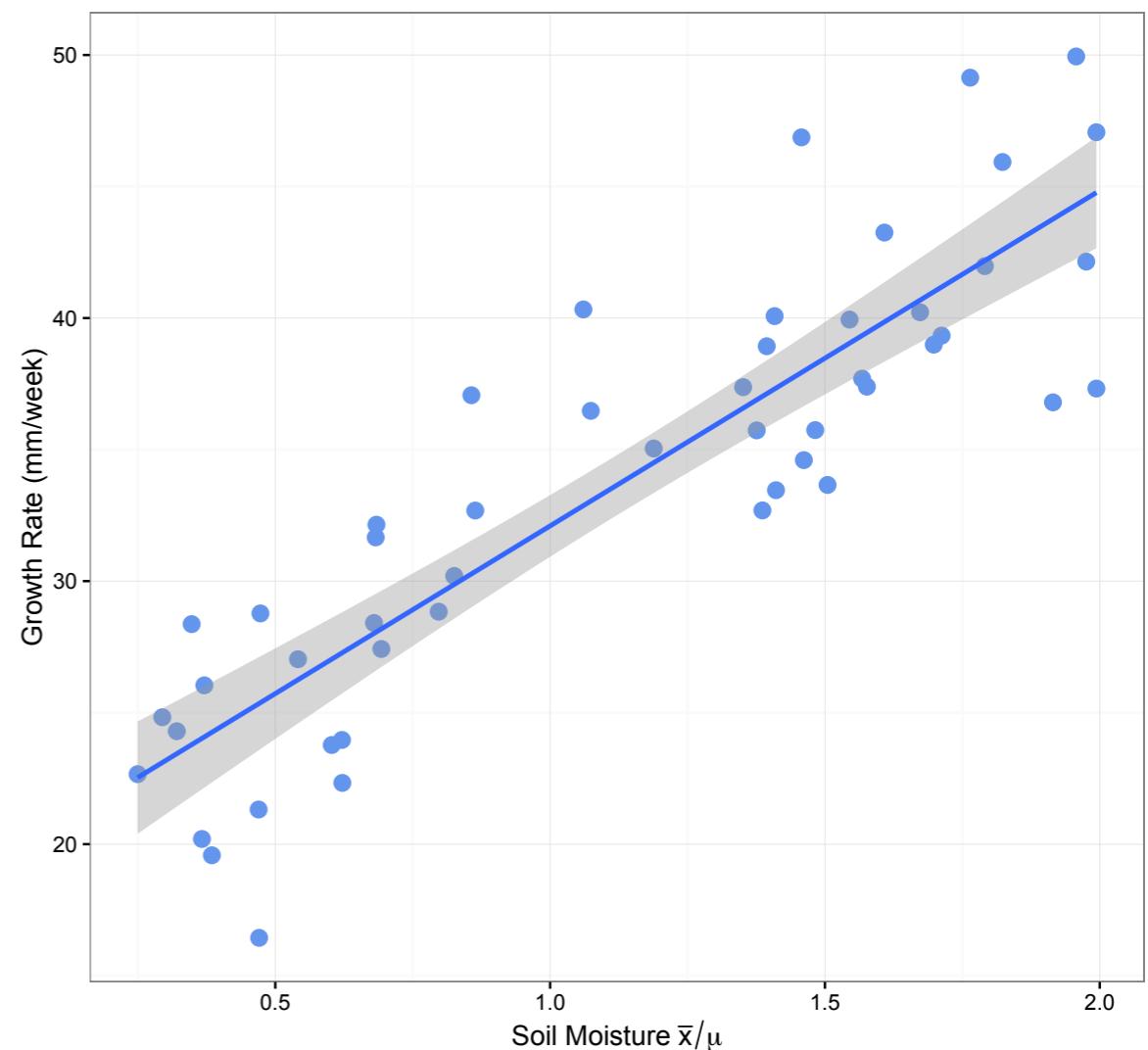
plot model workflow

- Make "New X"
 - `newX <- expand.grid()`
- Make "New Y"
 - `newY <- predict(model, newdata=newX, interval="confidence")`
- Houskeeping
 - `addThese <- data.frame(newX, newY)`
- Add lines (using lines)

NewX	NewY
0	
1	
2	
...	
30	

Review

- Name Your Dataframe
- Plot the data
- Estimate what you expect from the graph
 - Slope and Intercept
- Fit the model
- Check the Assumptions
- `anova()` and `summary()` for inference
- add line and CIs



```
plot(plant.growth.rate ~ soil.moisture.content, data = plant.growth,  
    xlab = "Soil moisture content", ylab = "Plant growth rate", cex = 2, pch = 21, bg =  
    "cornflowerblue")
```

ADDING fitted values to your figure

```
# make new X
```

```
newX <- expand.grid(soil.moisture.content = seq(from = 0.2, to = 2, length = 20))
```

```
# make new Y's (predictions)
```

```
newY <- predict(model1, newdata = newX, interval = "confidence")
```

```
# housekeeping
```

```
addThese <- data.frame(newX, newY)
```

```
# Add the lines - 3 of them - FROM THE addThese data frame
```

```
lines(fit ~ soil.moisture.content, data = addThese)
```

```
lines(lwr ~ soil.moisture.content, data = addThese, lty = 2, col = 'deeppink')
```

```
lines(upr ~ soil.moisture.content, data = addThese, lty = 2, col = 'deeppink')
```

ANOVA

Daphnia growth dataset

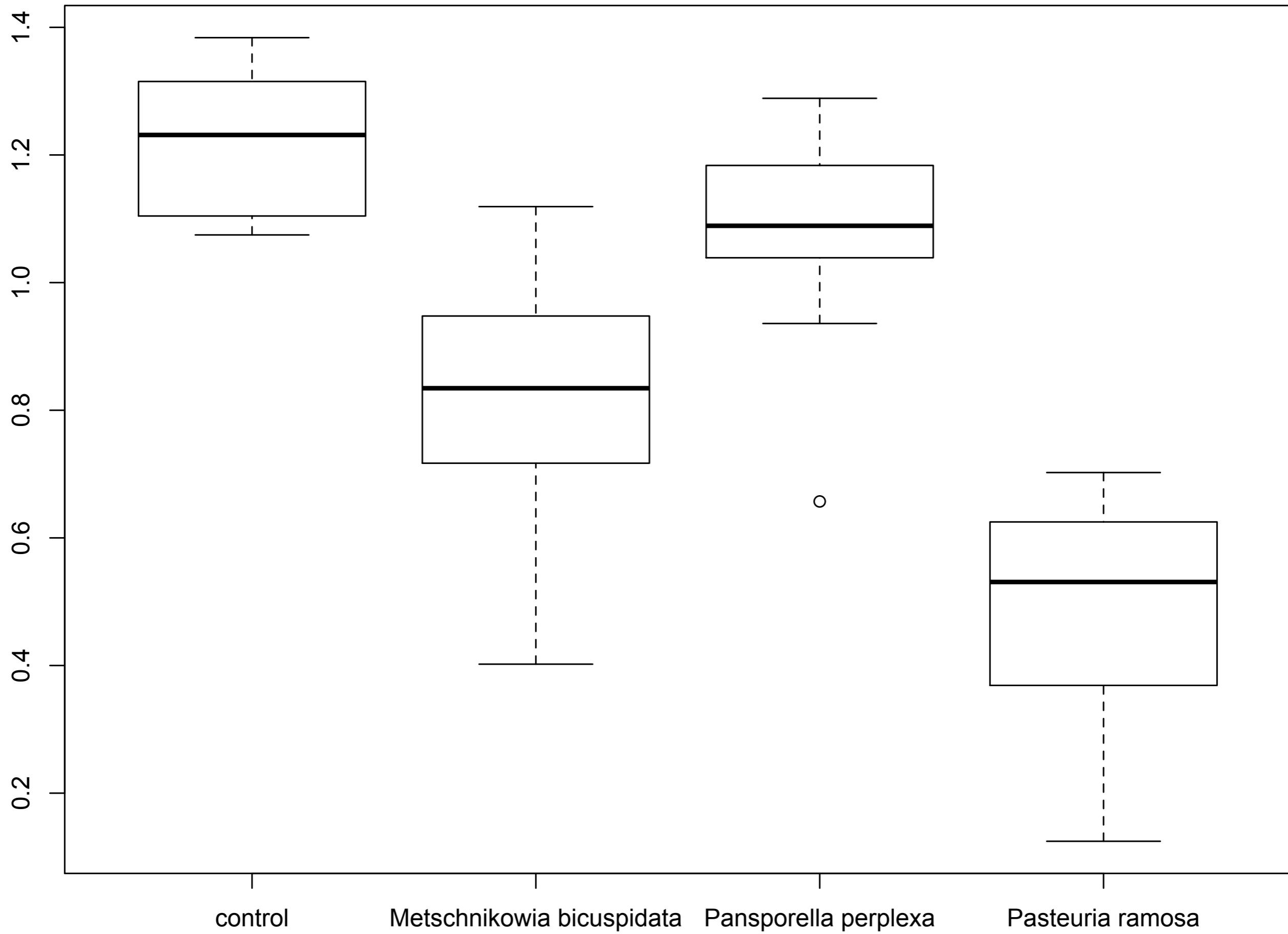


1. Import the data in R (Daphniagrowth.csv)
2. Make a box plot
3. Interpret the graph

parasite	rep	growth.rate
control	1	1.07470924
control	2	1.26590155
control	3	1.31515627
control	4	1.07575188
control	5	1.19676195
control	6	1.38379547
control	7	1.32396515
control	8	1.10436999
control	9	1.27753432
control	10	1.12114201
Metschnikowia bicuspidata	1	1.11905616
Metschnikowia bicuspidata	2	0.94766494
Metschnikowia bicuspidata	3	0.87796865
Metschnikowia bicuspidata	4	0.79101328
Metschnikowia bicuspidata	5	0.91878026
Metschnikowia bicuspidata	6	0.40212966
Metschnikowia bicuspidata	7	0.50584952
Metschnikowia bicuspidata	8	0.77944246



Photo: Functional Genomics Thickens the Biological Plot. Gewin V, PLoS Biology Vol. 3/6/2005, e219. doi:10.1371/journal.pbio.0030219



Fitting the model using lm()

```
model2 <- lm(growth.rate ~  
parasite, data = daphnia)
```

Why lm not aov?

```
model2 <- lm(growth.rate ~  
parasite, data = daphnia)
```

Balanced vs unbalanced

Balanced
(= orthogonal design)

	High Food	Low Food
Wet	9	9
Control	9	9
Dry	9	9

Unbalanced

	High Food	Low Food
Wet	10	8
Control	6	10
Dry	10	5

Order of testing does
not matter

Order of testing does
matter

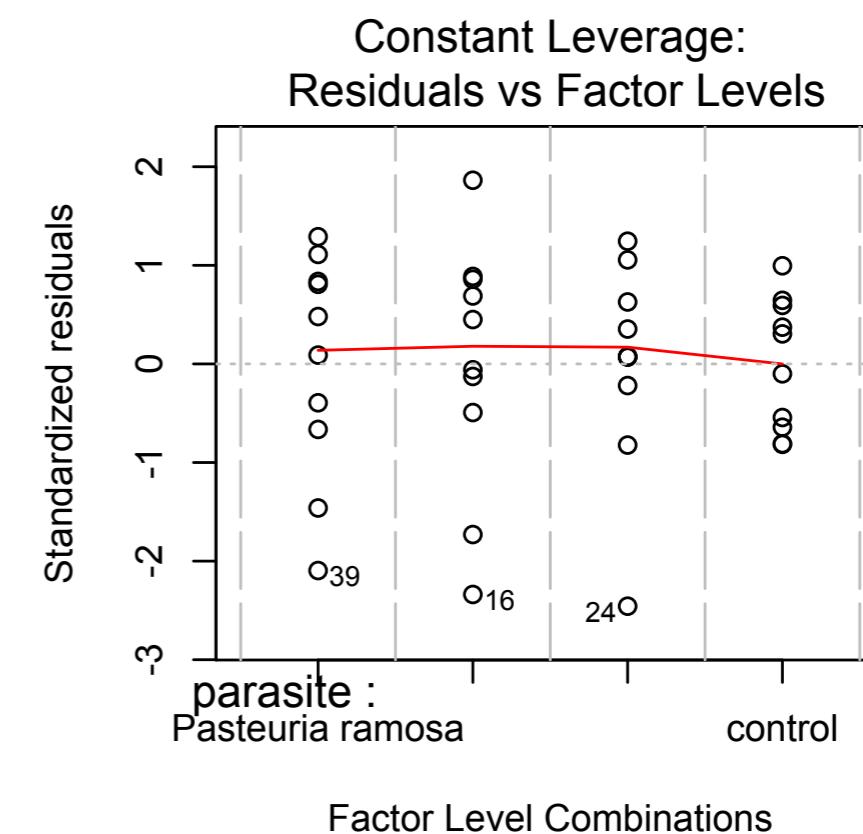
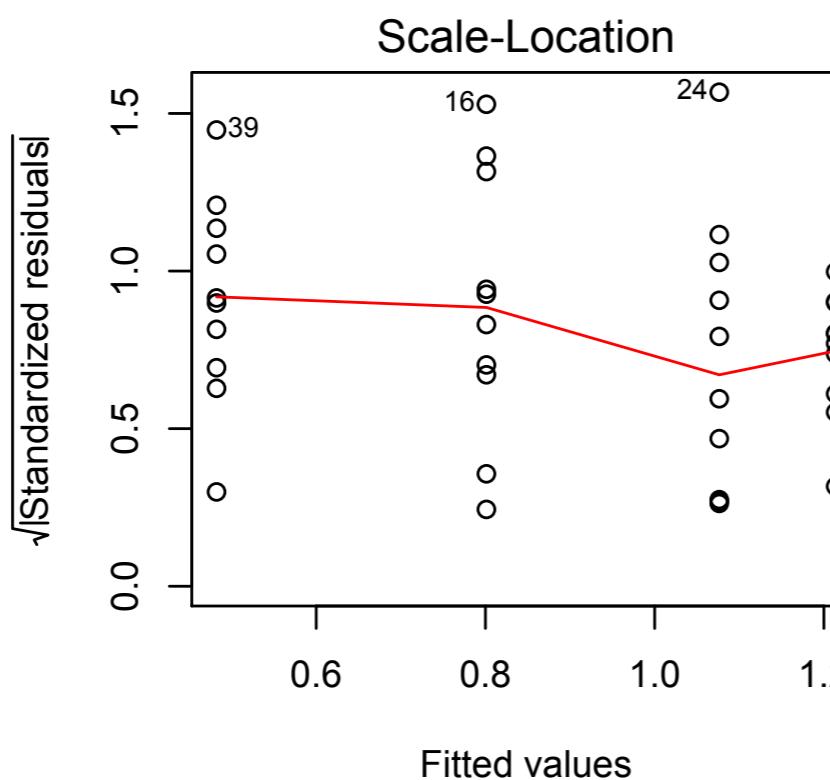
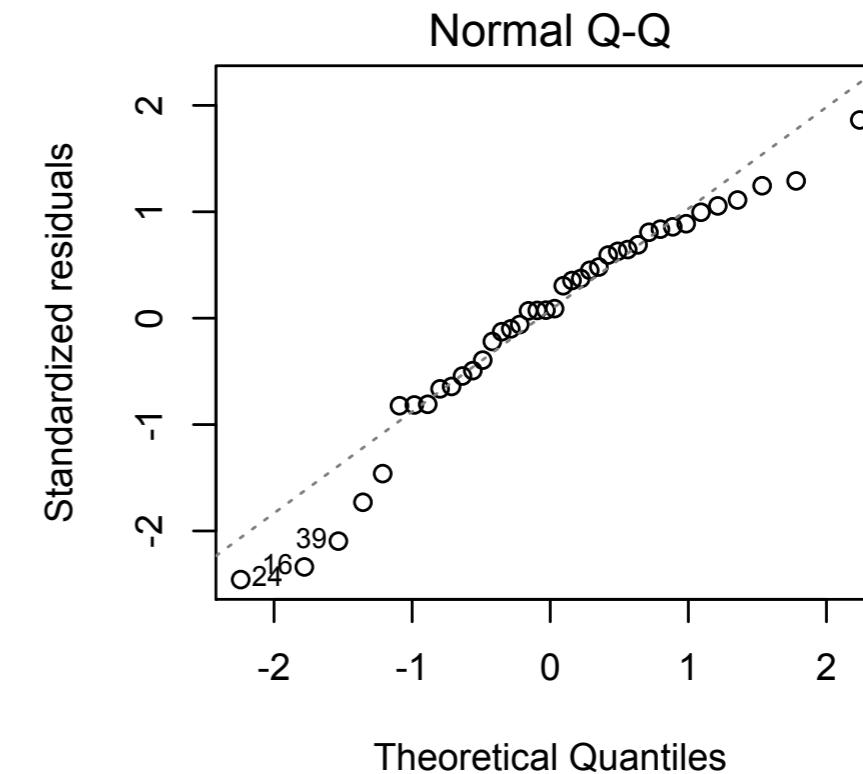
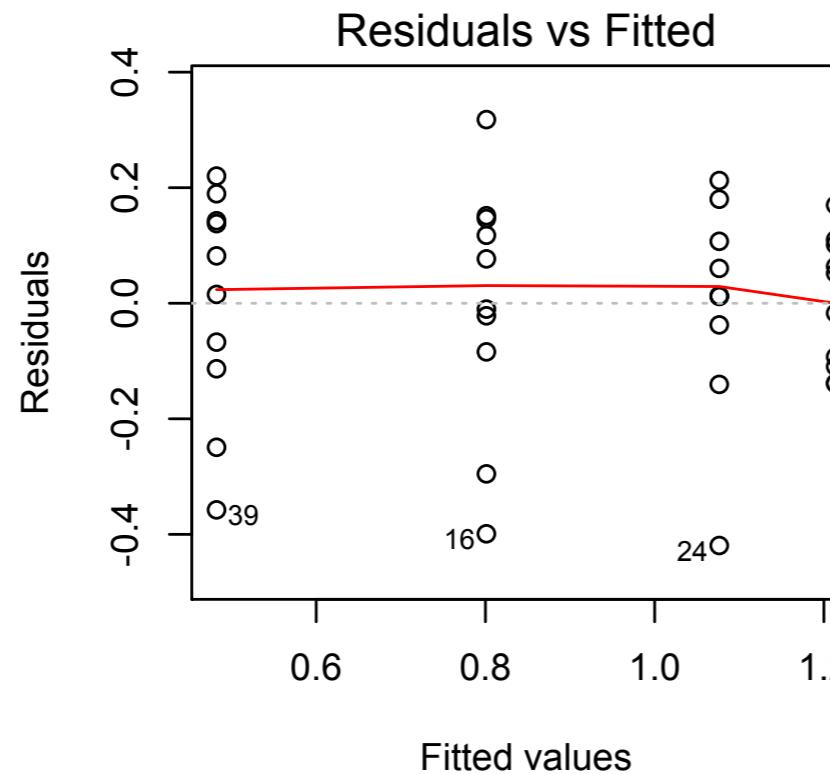
Evaluating the model assumptions...

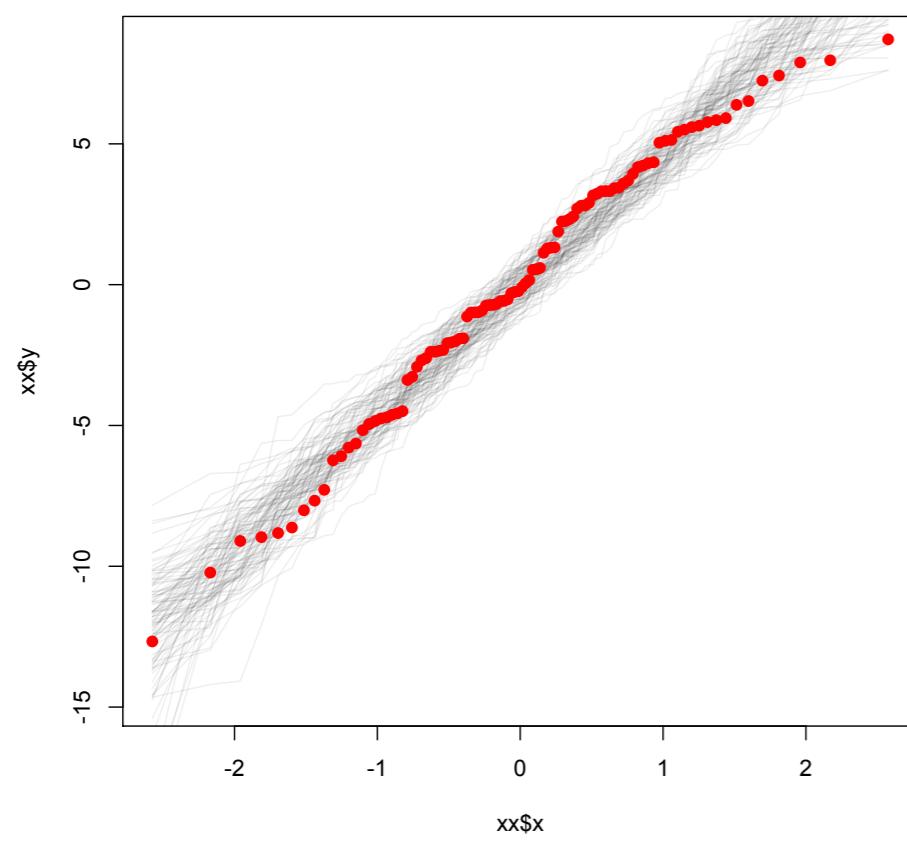
plot(model2)

```
model2 <- lm(growth.rate ~ parasite, daphnia)
```

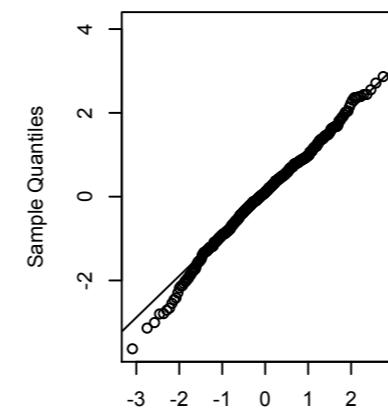
```
par(mfrow=c(2, 2))
```

```
plot(mm)
```

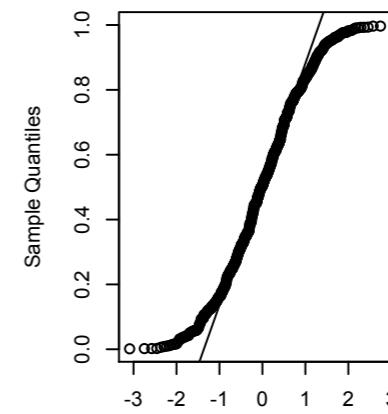




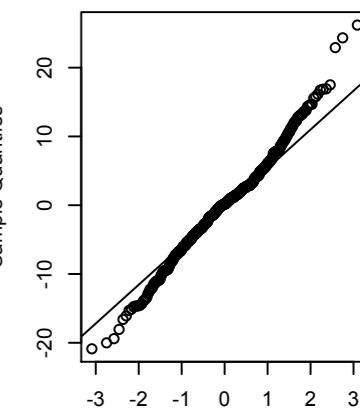
NORMAL



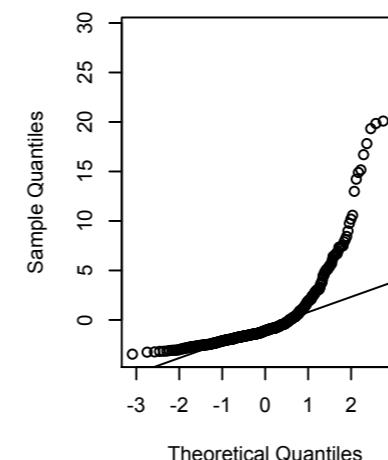
UNIFORM



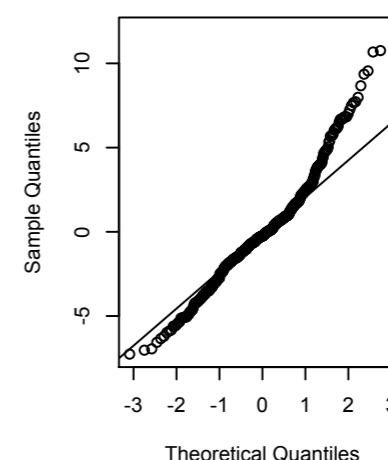
POISSON



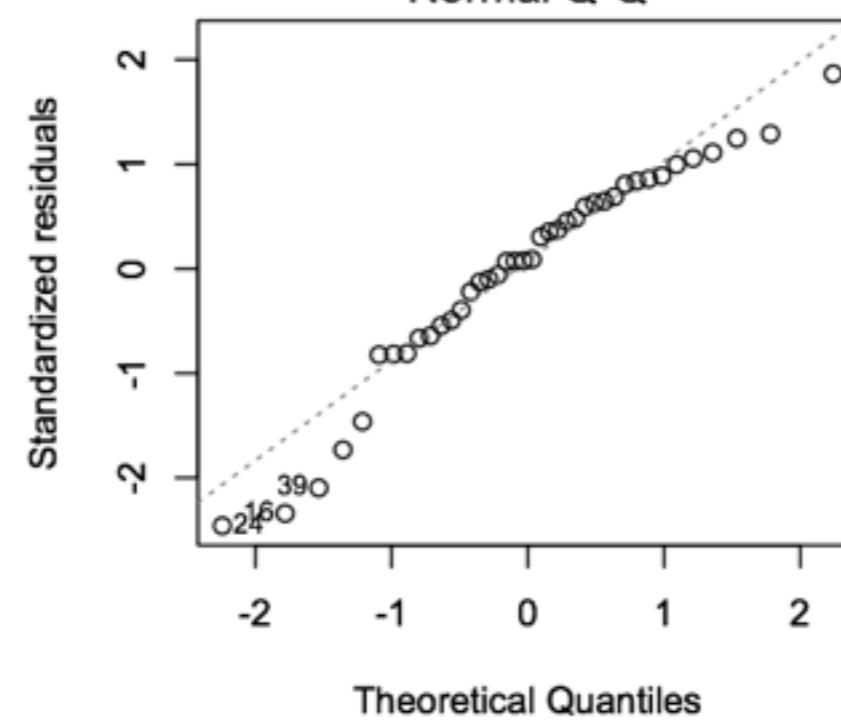
LOGNORM



BINOMIAL



Normal Q-Q

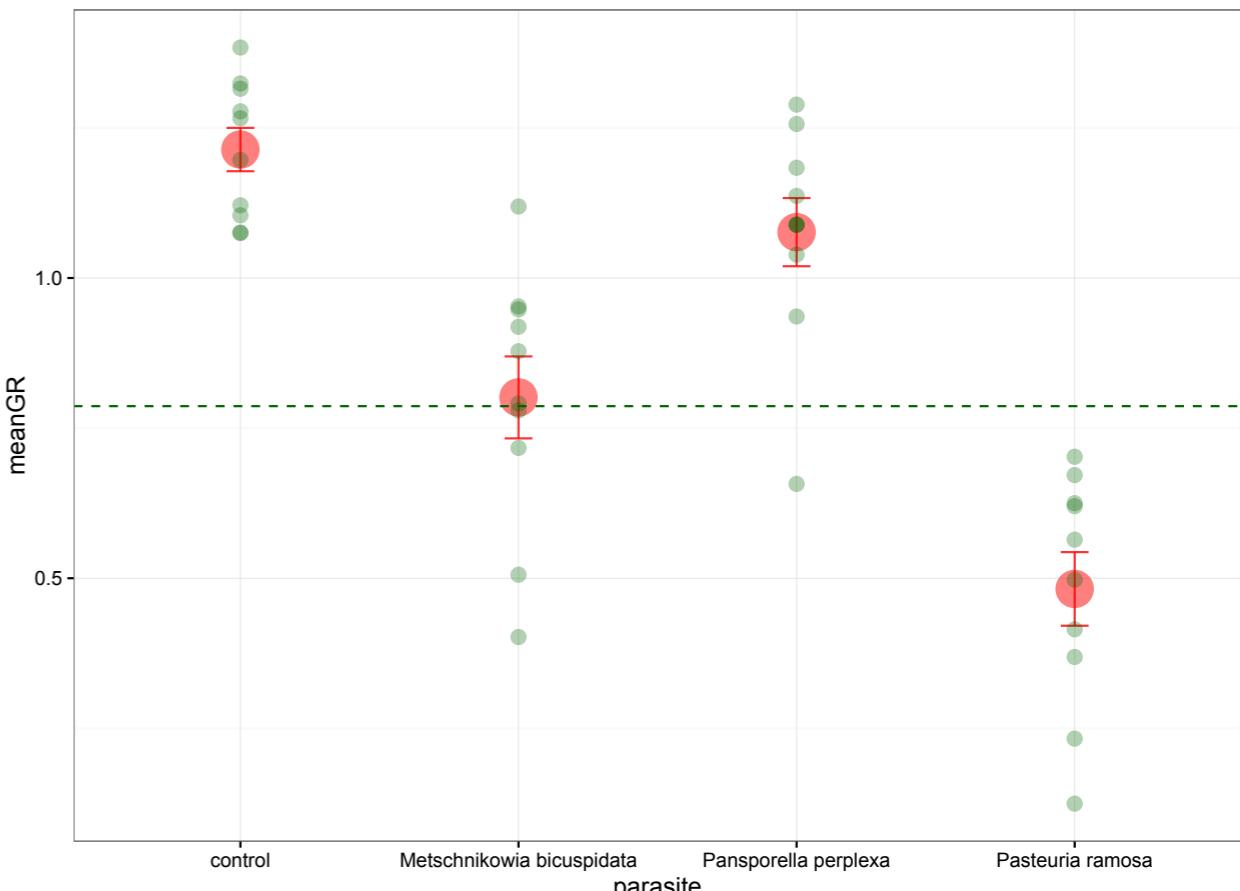


the workhorses

- **anova**(model.name)
 - *overall significance of main, 2-way, 3-way etc.*
 - *sequential sums of squares*
- **summary**(model.name)
 - coefficients, significance, with respect to contrasts

apply anova and
summary to your
model

ANOVA table



```
> anova(mm)
```

Analysis of Variance Table

Response: growth.rate

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
parasite	3	3.1379	1.04597	32.325	2.571e-10 ***
Residuals	36	1.1649	0.03236		

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

> mm <- lm(growth.rate ~ parasite, dd)
> summary(mm)

```

Call:
`lm(formula = growth.rate ~ parasite, data = dd)`

Residuals:

Min	1Q	Median	3Q	Max
-0.41930	-0.09696	0.01408	0.12267	0.31790

Coefficients:

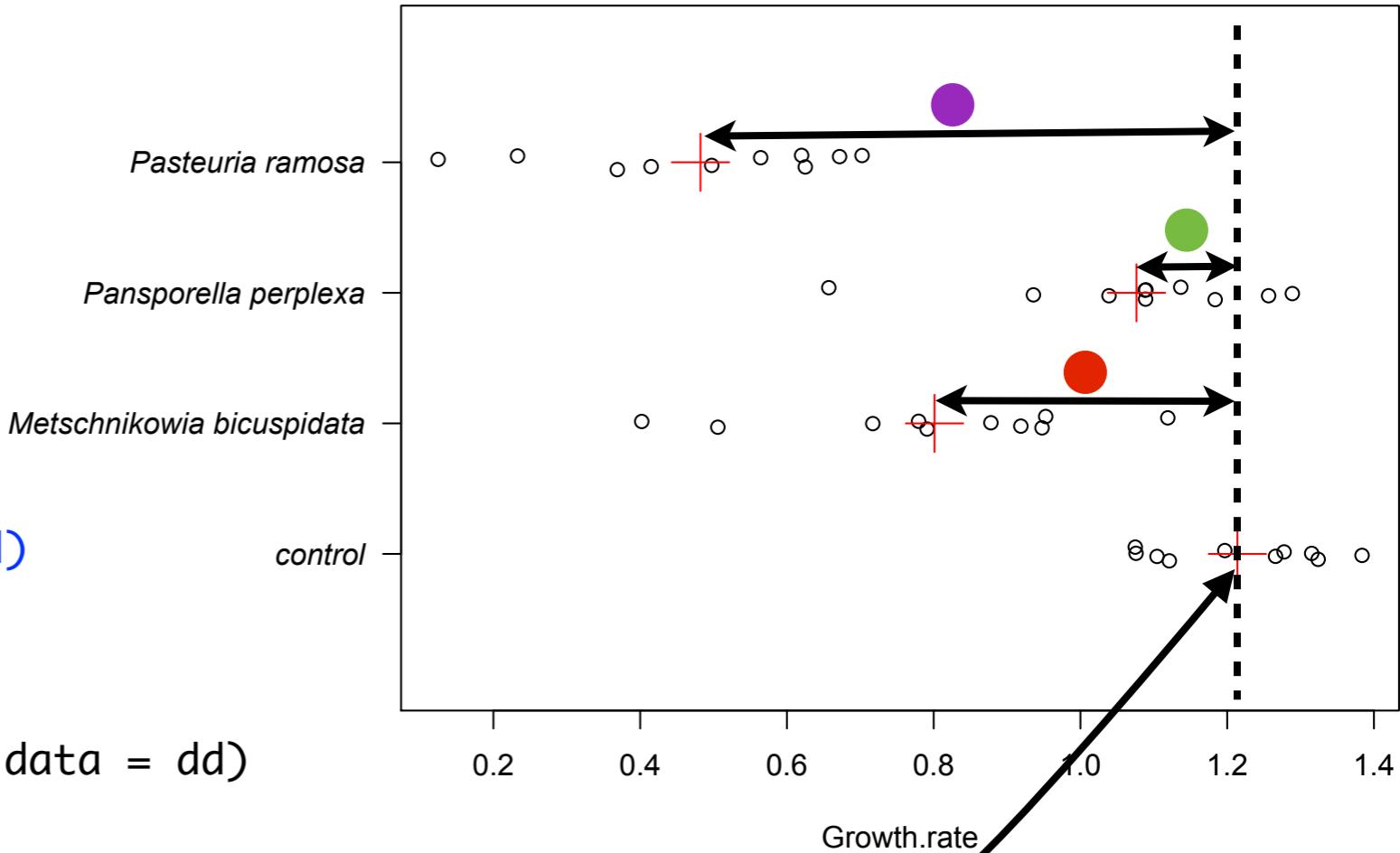
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.21391	0.05688	21.340	< 2e-16 ***
parasiteMetschnikowia bicuspidata	-0.41275	0.08045	-5.131	1.01e-05 ***
parasitePansporella perplexa	-0.13755	0.08045	-1.710	0.0959 .
parasitePasteuria ramosa	-0.73171	0.08045	-9.096	7.34e-11 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1799 on 36 degrees of freedom

Multiple R-squared: 0.7293, Adjusted R-squared: 0.7067

F-statistic: 32.33 on 3 and 36 DF, p-value: 2.571e-10



Interpretation

- Treatment contrasts
- R behaves alphabetically and numerically

ABC.....123.....ABC 123....

Learn your Contrasts

Package	Contrast
R	treatment
SPLUS	helmert
SAS	SAS
Minitab	sum

```
options(contrasts=contr.sum,contr.poly)
```

Tukey Test

```
tukey<-glht(mod.aov,  
linfct=mcp(parasite = "Tukey"))
```

glht	general linear hypothesis tests
linfct	linear function
mcp	multiple comparison

```
c("Dunnett", "Tukey", "Sequen", "AVE",  
"Changepoint", "Williams", "Marcus",  
"McDermott", "UmbrellaWilliams",  
"GrandMean")
```

ANCOVA

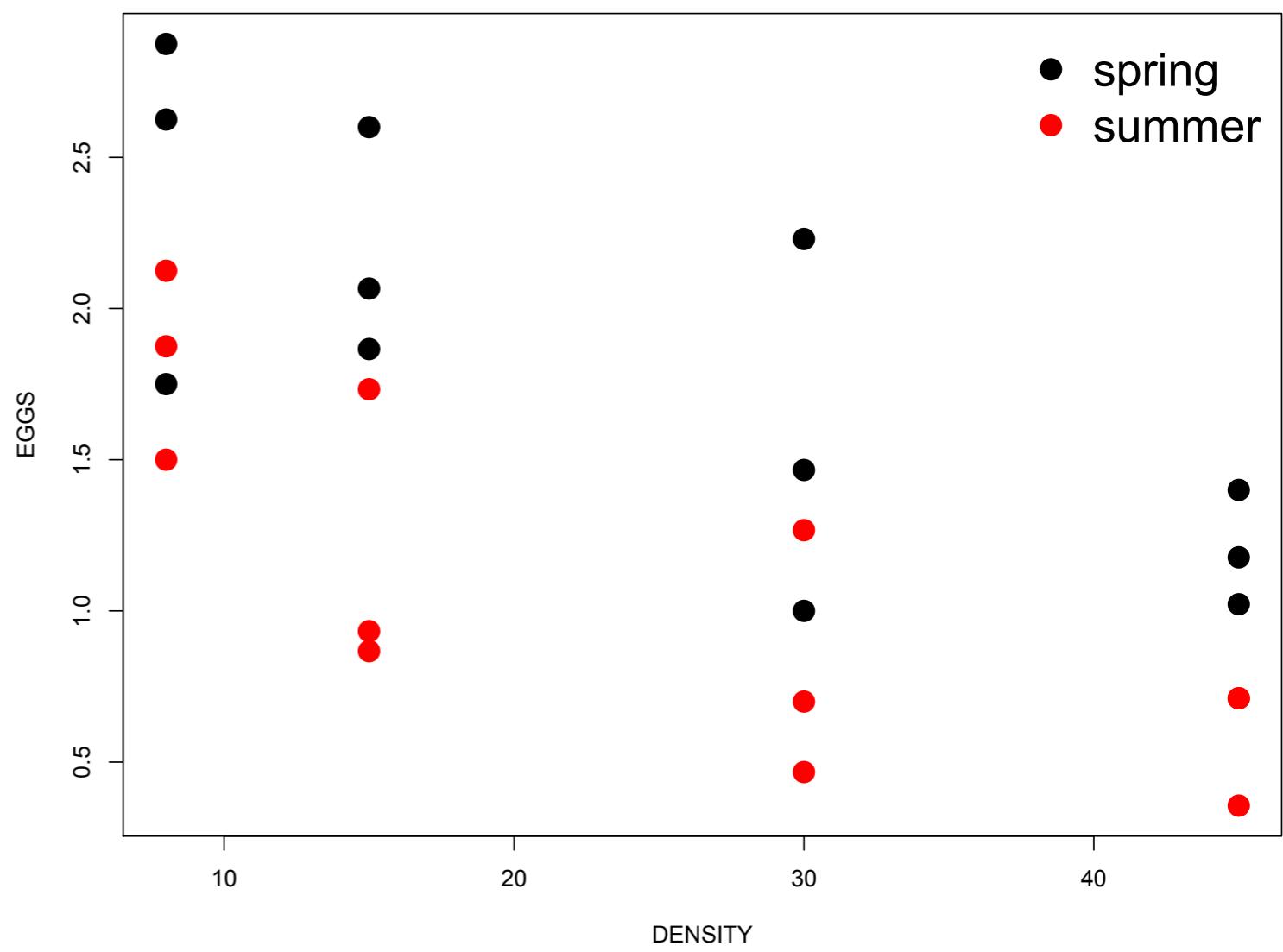
Limpet dataset



Planned Experiment

- Egg numbers vs. Density
 - Density Dependence
 - 4 levels; covariate
- Seasonal Differences
 - 2 levels
- 3 Reps

1. Import the data in R (limpet.csv)
2. Make a scatterplot
3. Interpret the graph



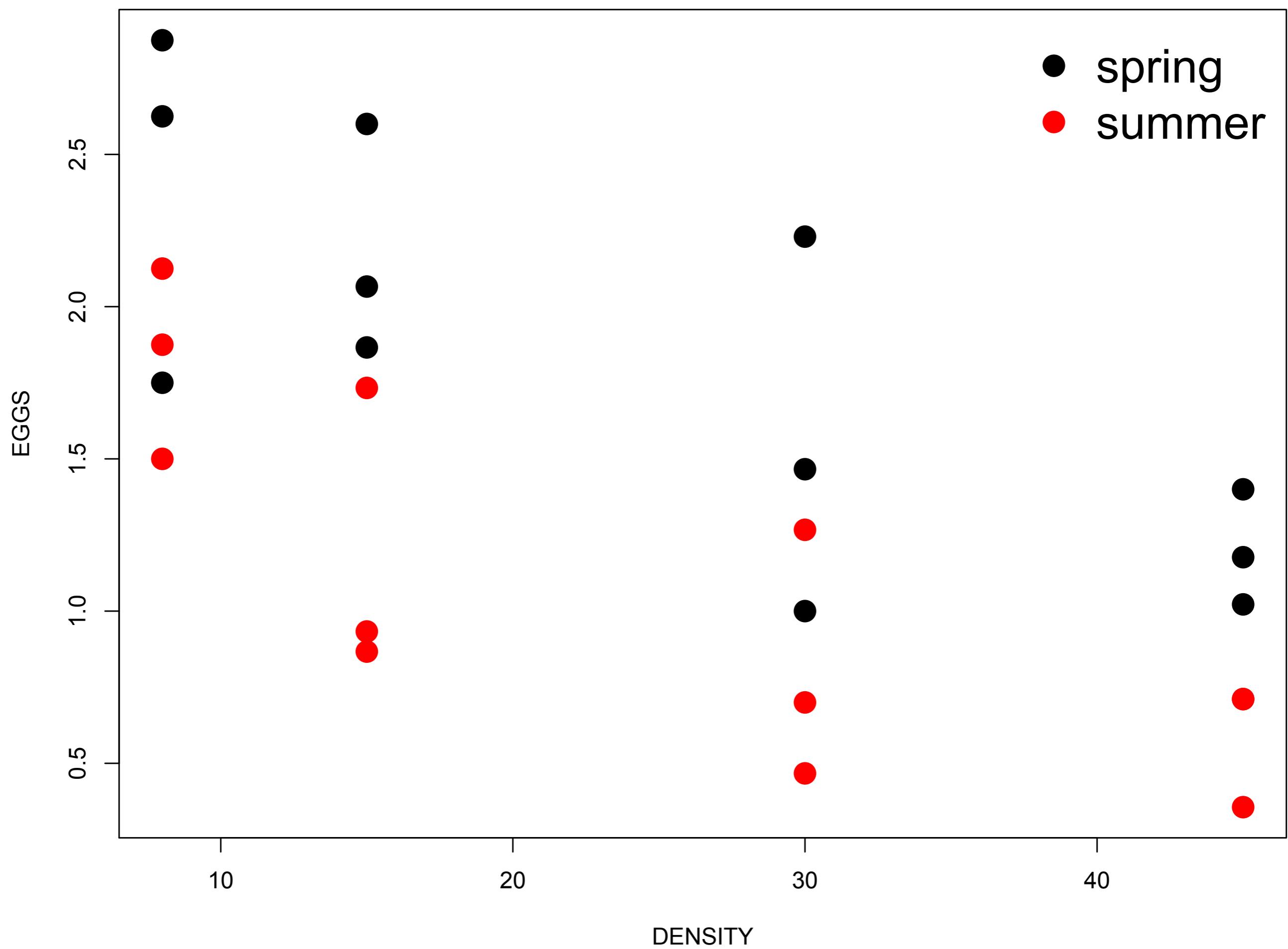
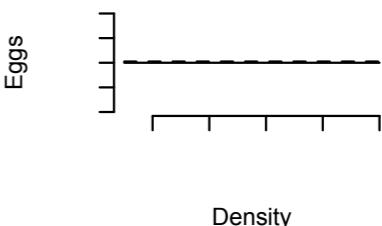
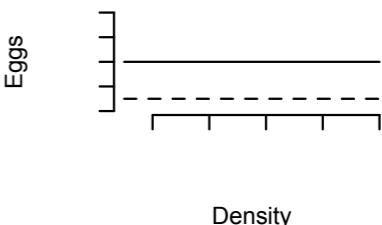
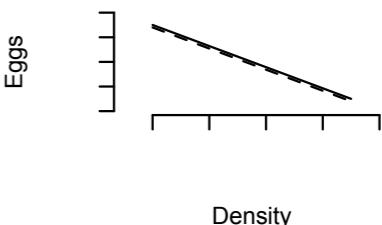
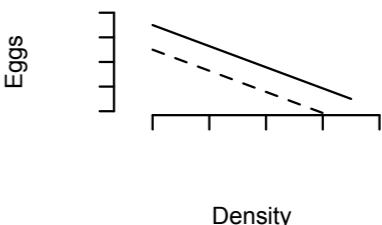
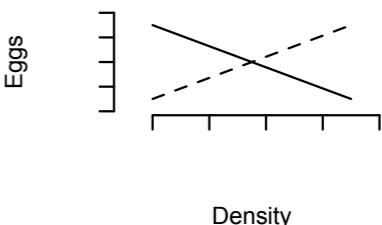


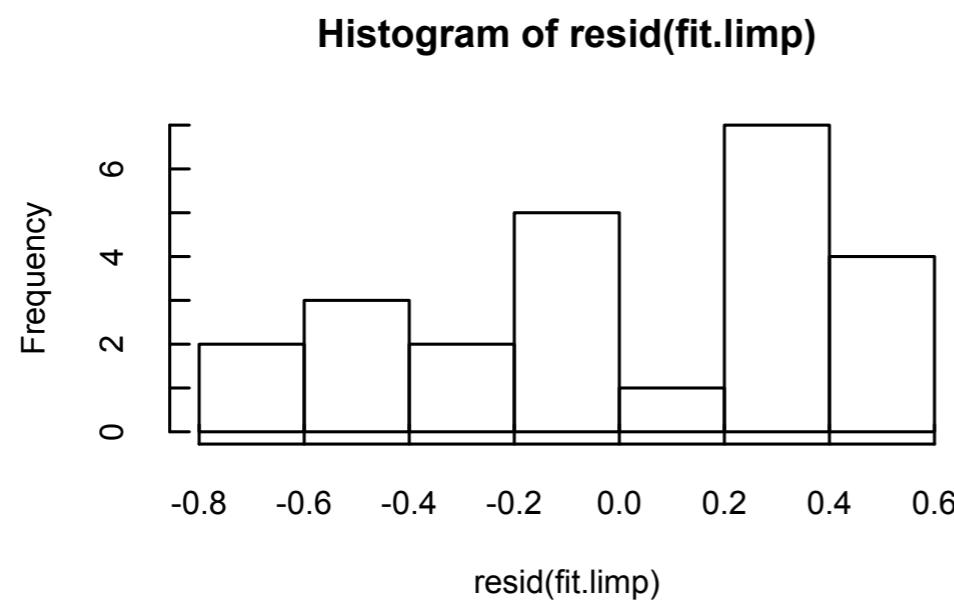
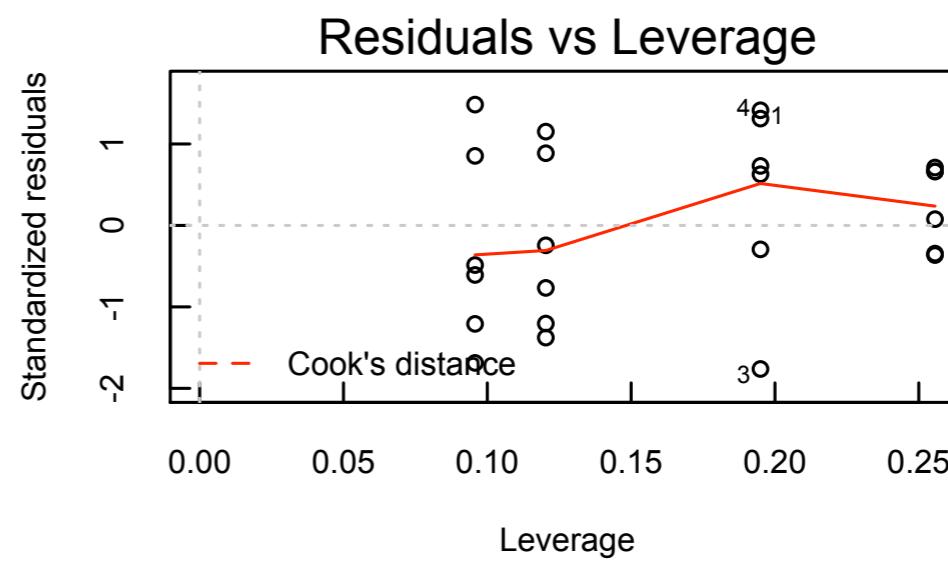
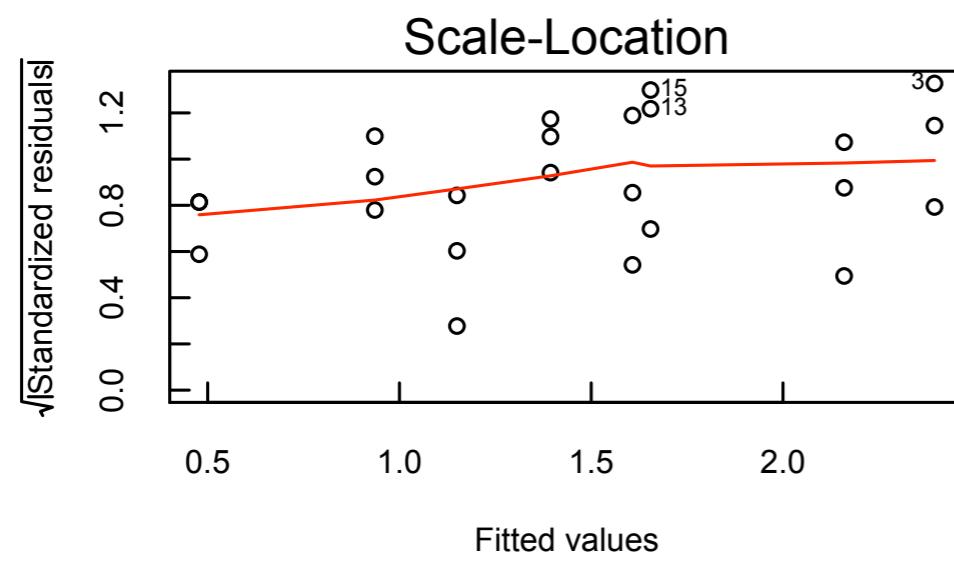
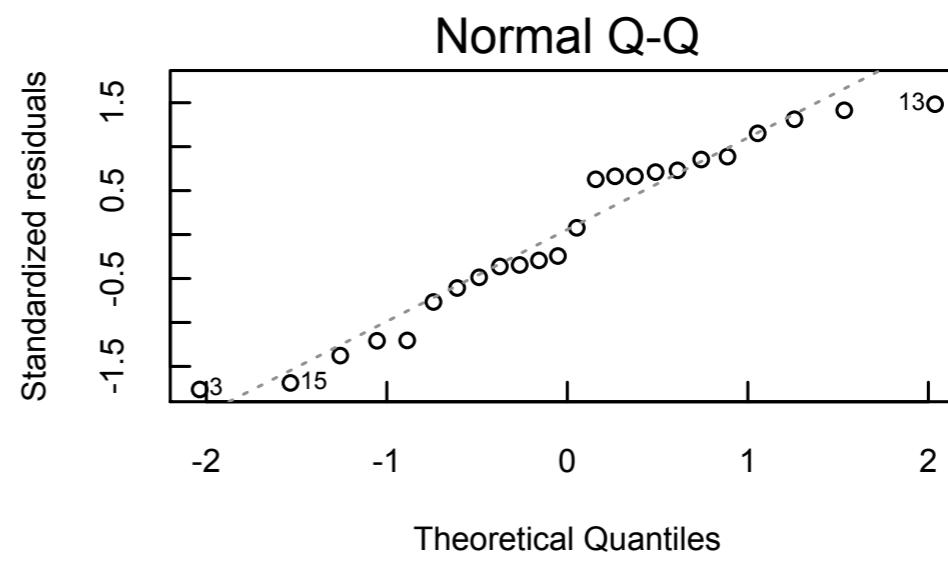
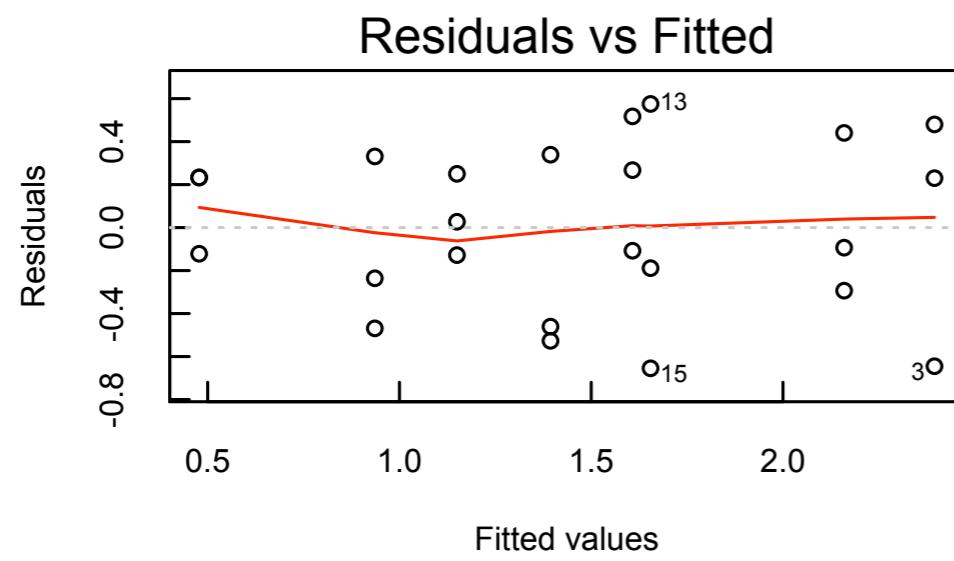
FIGURE	VERBAL HYPOTHESIS	INTERCEPT/ SLOPE	MODEL IN R	GRAPHICAL
A	There is an average number of eggs produced by limpets that does not vary with density or season	Common Intercept Zero Slope(s)	<code>lm(Eggs~1, data=limp)</code>	Eggs  Density
B	There is an average number of eggs produced by limpets that does not vary with density but is reduced in the summer season	Different Intercepts Zero Slope(s) Parallel Horizontal Line	<code>lm(Eggs~Season, data=limp)</code>	Eggs  Density
C	The number of eggs produced declines with density, but the maximum number of eggs (intercept) and the rate (slope) does not vary with season	Same Intercept Same Slope Same Lines	<code>lm(Eggs~Density, data=limp)</code>	Eggs  Density
D	The number of eggs produced declines with density and the maximum number of eggs (intercept) differ between seasons but the rate (slope) does not vary with season	Different Intercepts Same (negative) Slope Parallel lines	<code>lm(Eggs~Density+Season, data=limp)</code>	Eggs  Density
E	The maximum number of eggs (intercept) and the rate (slope) does vary with season	Different Intercepts Different Slopes "Crossing" Lines	<code>lm(Eggs~Density*Season, data=limp)</code>	Eggs  Density

Fitting the model using lm()

```
model3 <- lm(EGGS ~ DENSITY *  
SEASON, data = limpet)
```

Evaluating the model assumptions...

plot(model3)



apply anova and
summary to your
model

> **anova(model3)**

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(model3)**

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

```
> summary(model3) # coefficients
```

Call:

```
lm(formula = EGGS ~ DENSITY * SEASON, data = limp)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.65468	-0.25021	-0.03318	0.28335	0.57532

Spring Intercept & Slope

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

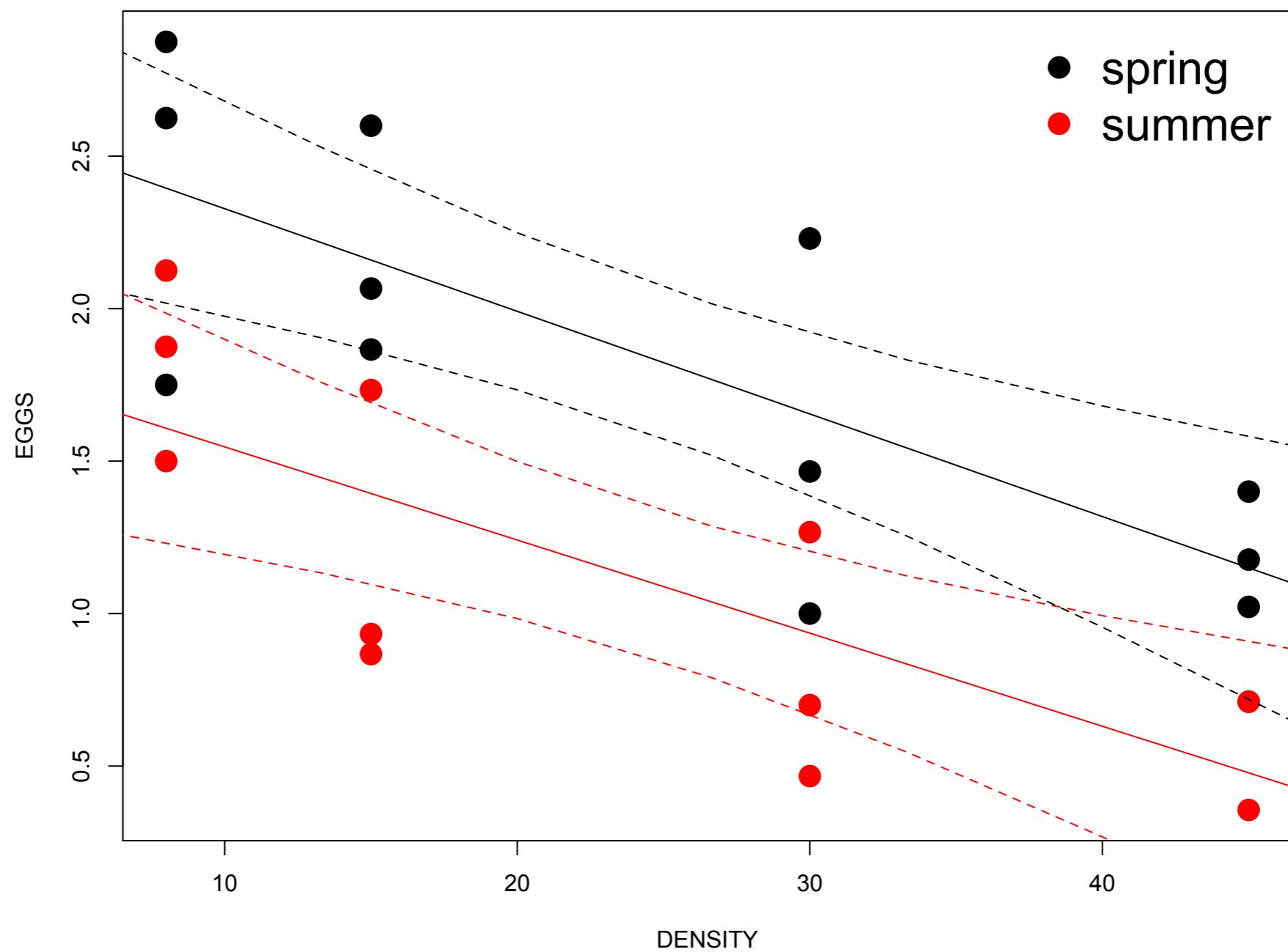
Δ Intercept
Sp to Sum

Δ Slope
Sp to Sum

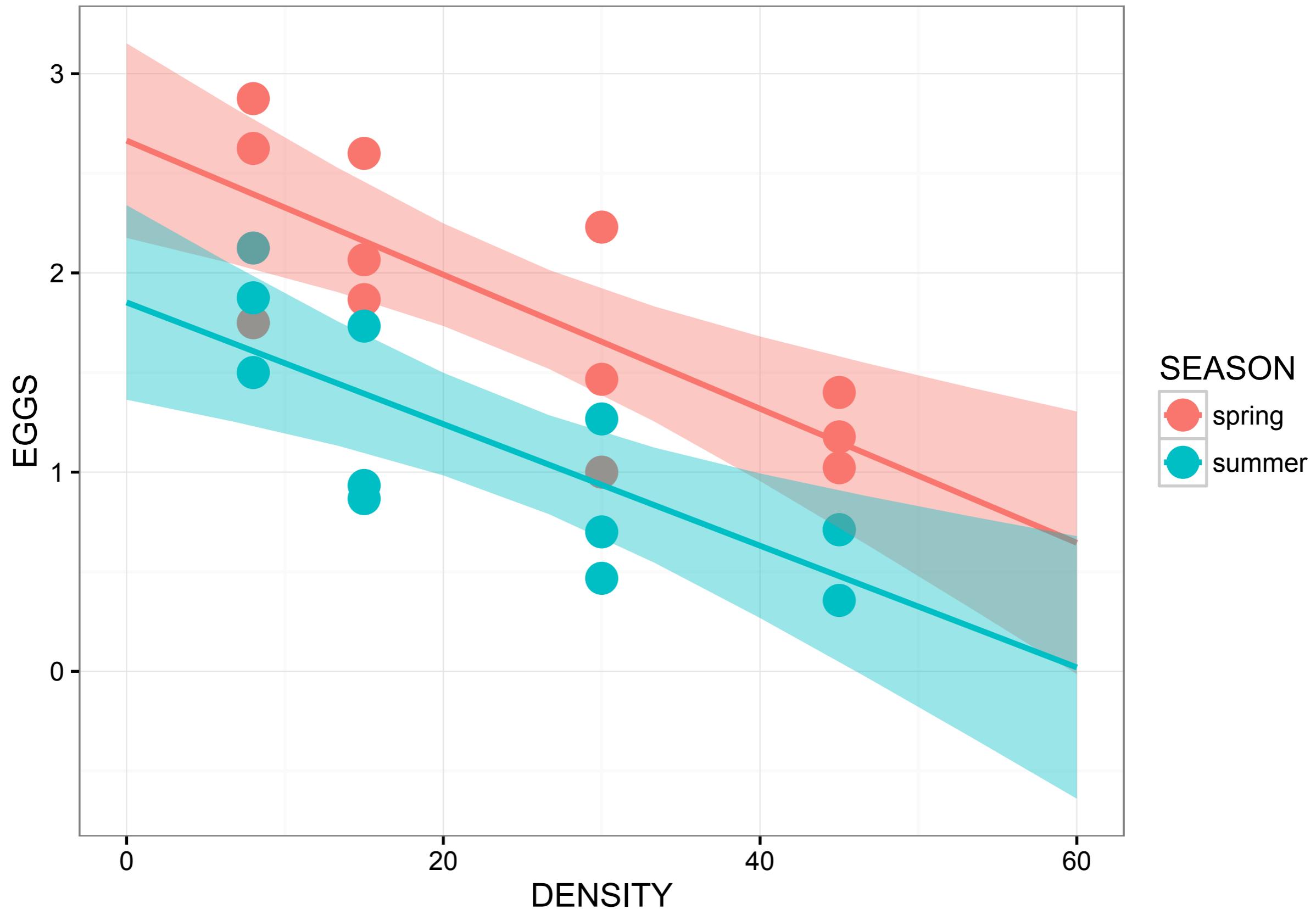
plot model workflow

- Make "New X"
 - `newX <- expand.grid()`
- Make "New Y"
 - `newY <- predict(model, newdata=newX, interval="confidence")`
- Houskeeping
 - `addThese <- data.frame(newX, newY)`
- Add lines using `lines()`

NewX	NewY
0	
1	
2	
...	
30	

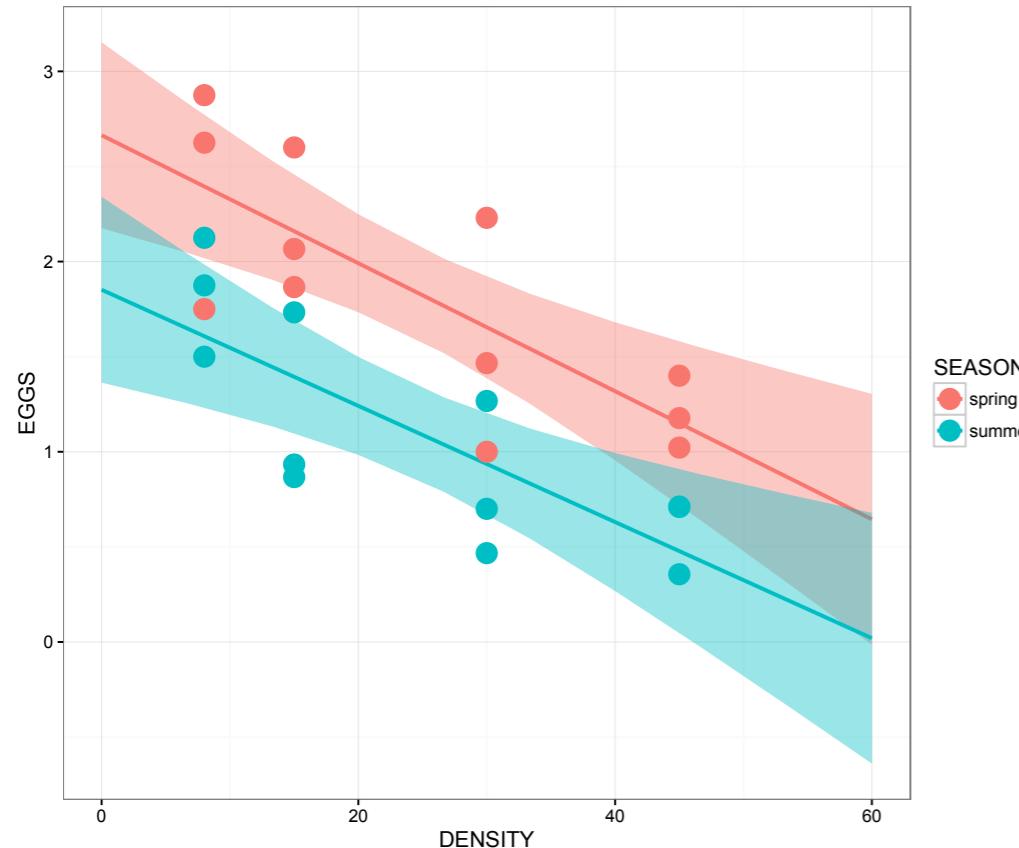


Prettier with ggplot...



the workhorses

- **anova(model.name)**
 - overall significance of main, 2-way, 3-way etc.
 - sequential sums of squares
- **summary(model.name)**
 - coefficients, significance, with respect to contrasts



plot model workflow

- Make "New X"
 - `newX <- expand.grid()`
- Make "New Y"
 - `newY <- predict(model, newdata=newX, interval="confidence")`
- Houskeeping
 - `addThese <- data.frame(newX, newY)`
- use `rename()`

NewX	NewY
0	
1	
2	
...	
30	

anova()
summary()
predict()
all work with
regression, anova and ancova

Two way ANOVA

Two way ANOVA

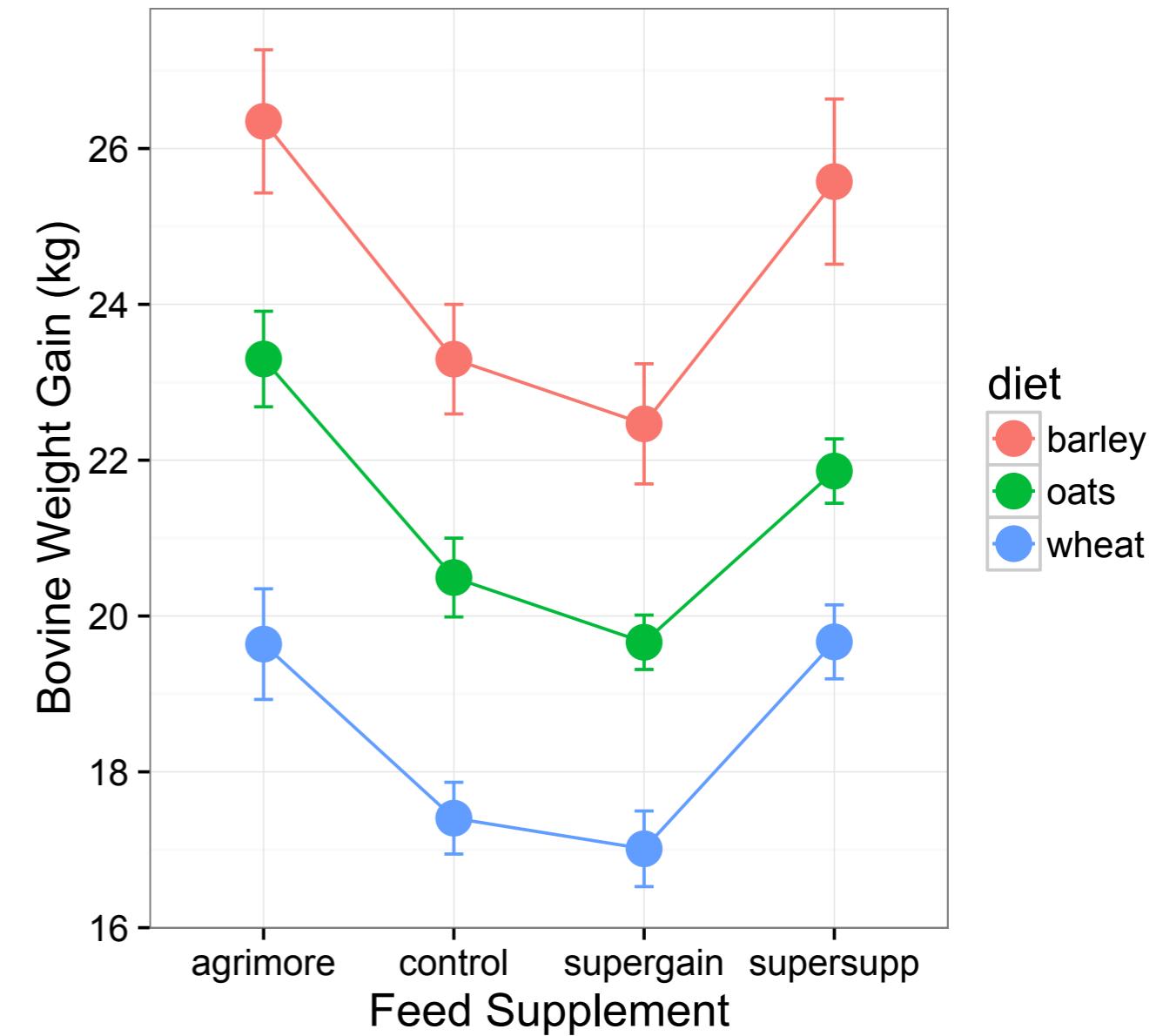
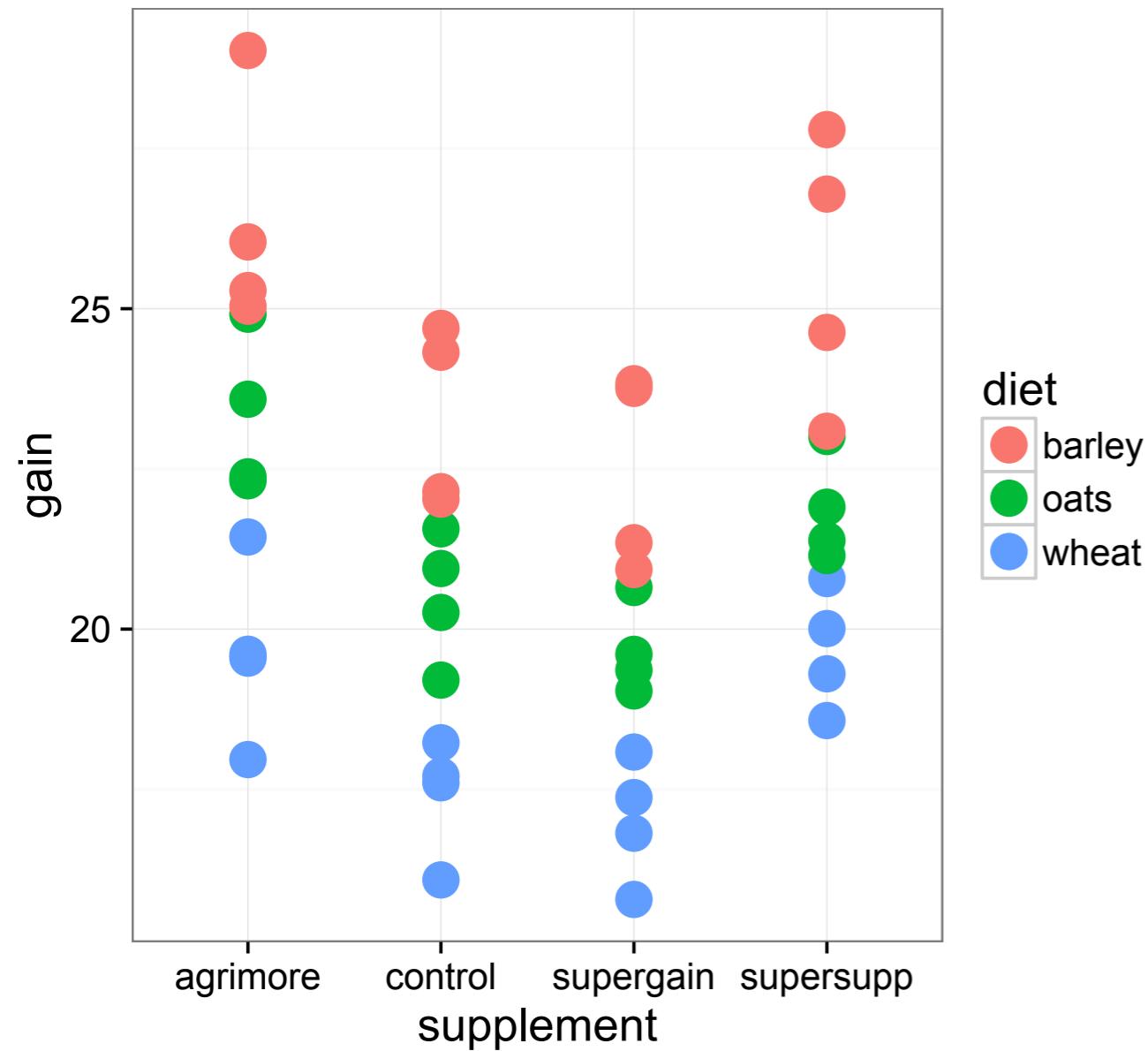
1. Import the data in R (growth.csv)
2. Get the plots from yesterday....

supplement	diet	gain
supergain	wheat	17.3712511
supergain	wheat	16.814889
supergain	wheat	18.0818374
supergain	wheat	15.7817483
control	wheat	17.7065646
control	wheat	18.2271693
control	wheat	16.0865029
control	wheat	17.6018361



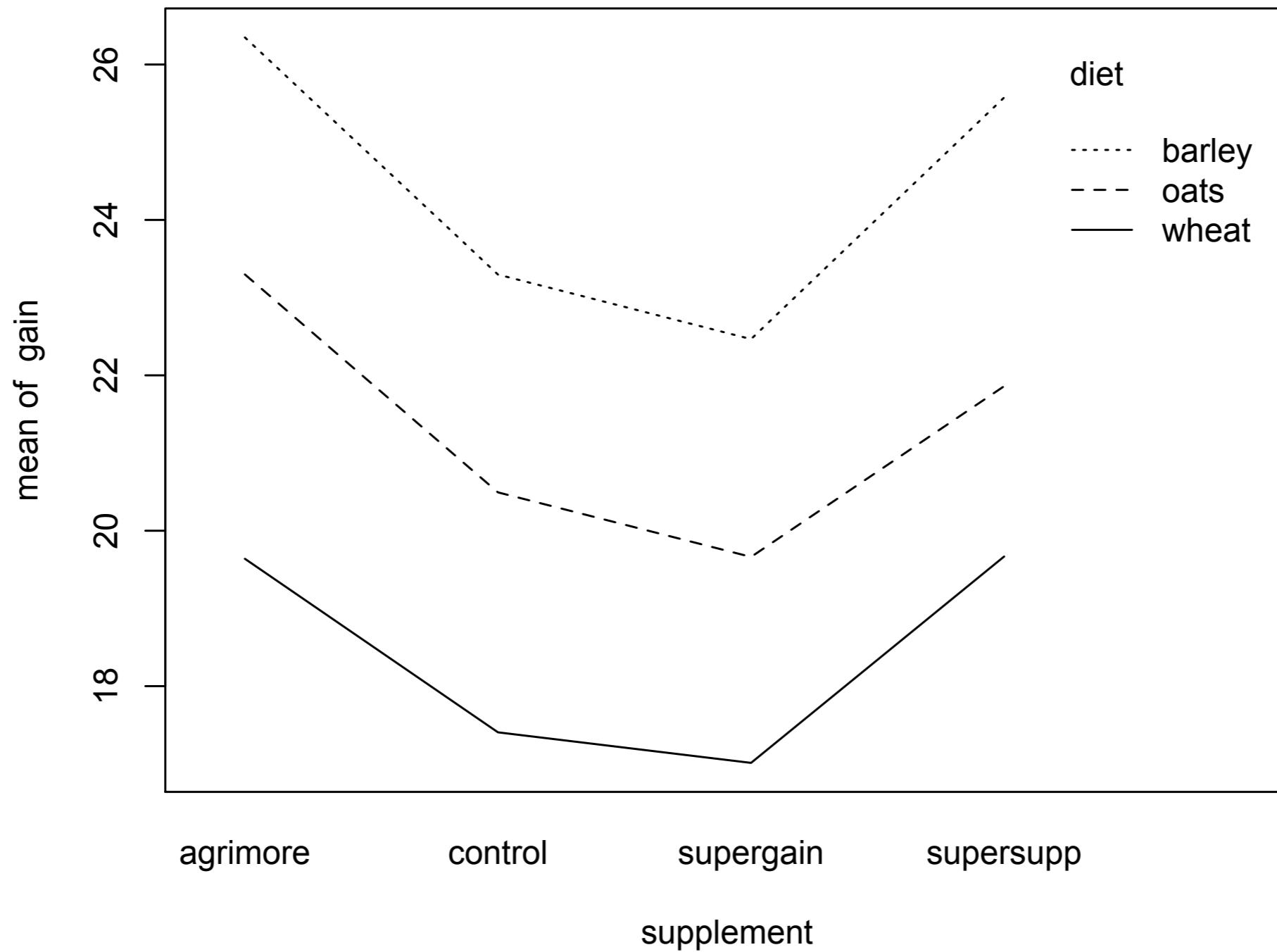
[Daniel Schwen](#)

```
> str(dd)
'data.frame': 48 obs. of 3 variables:
$ supplement: Factor w/ 4 levels "agrimore","control",...: 3 3 3 3 2 2 2 2 4
$ diet       : Factor w/ 3 levels "barley","oats",...: 3 3 3 3 3 3 3 3 3 ...
$ gain       : num  17.4 16.8 18.1 15.8 17.7 ...
```

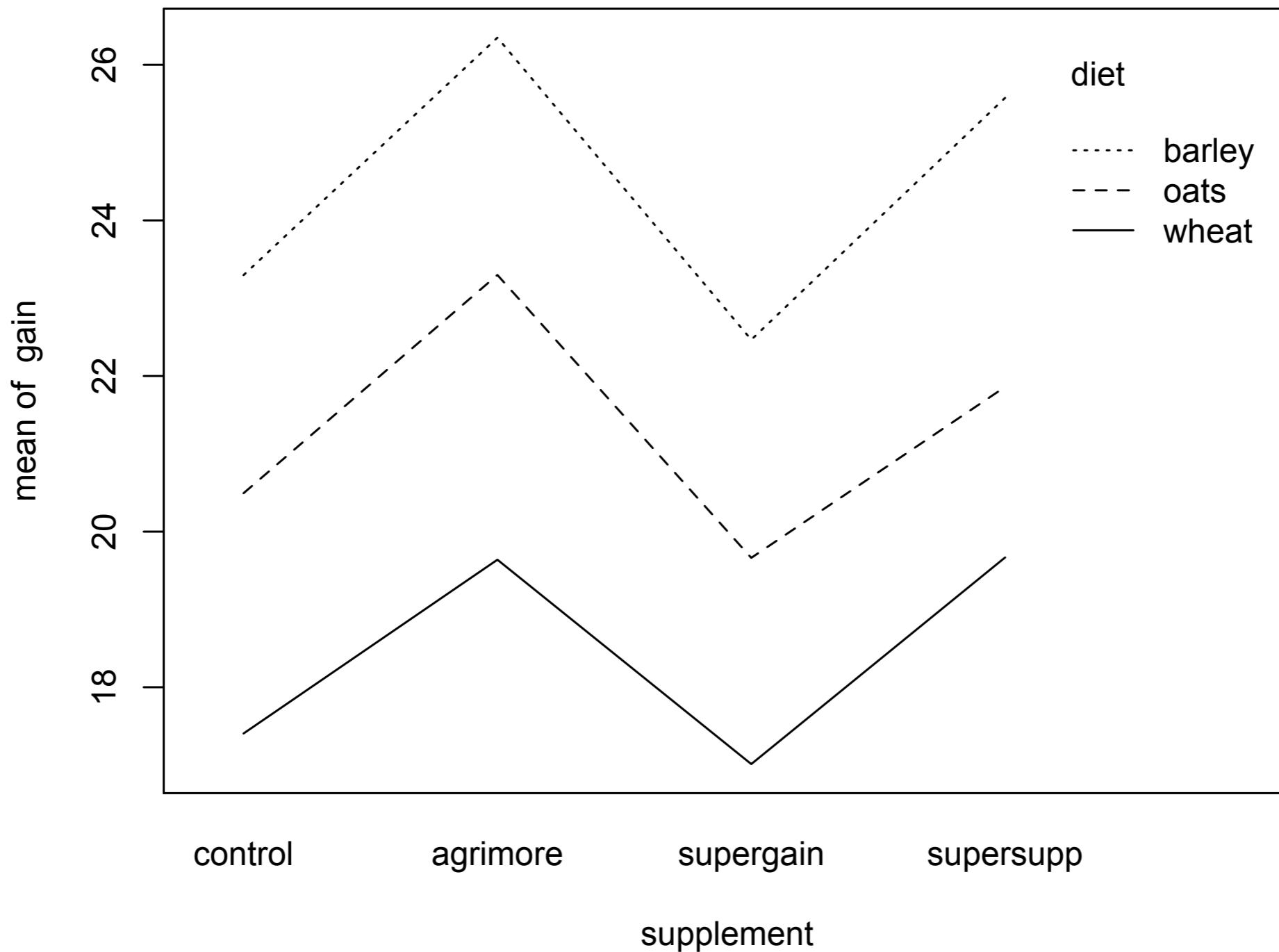


interaction plots

```
with(mydata, interaction.plot(supplement, diet, gain))
```



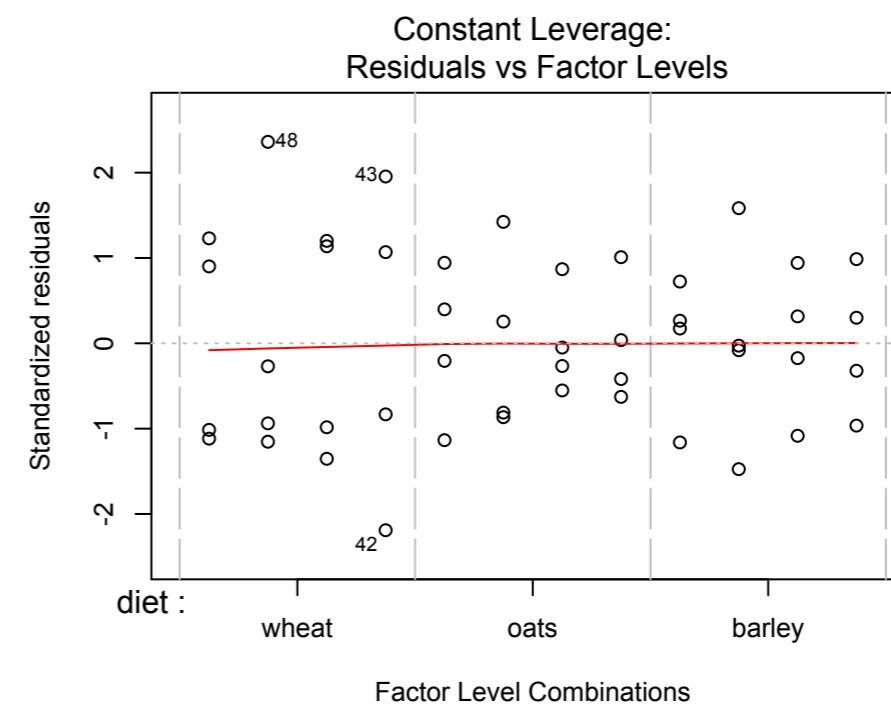
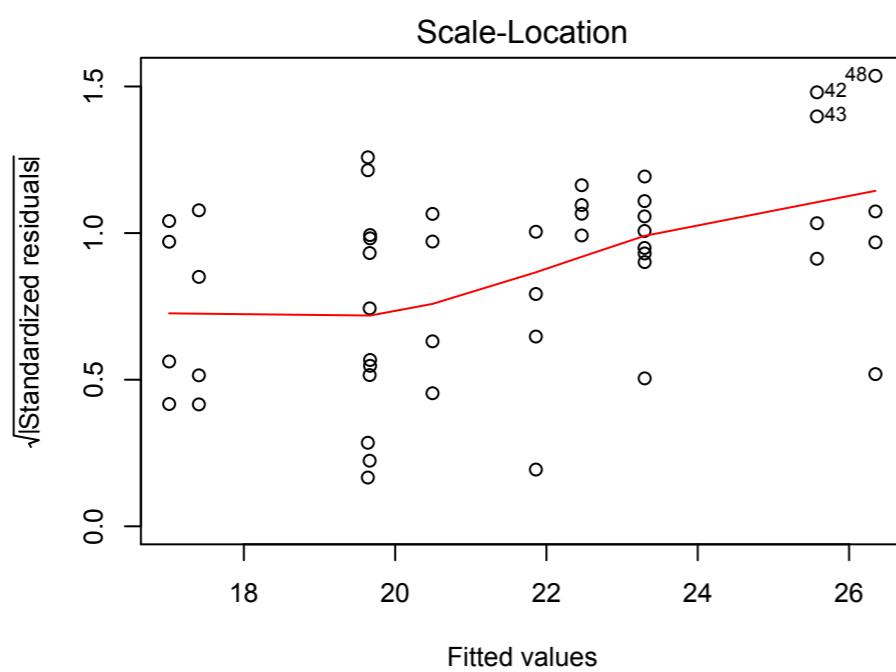
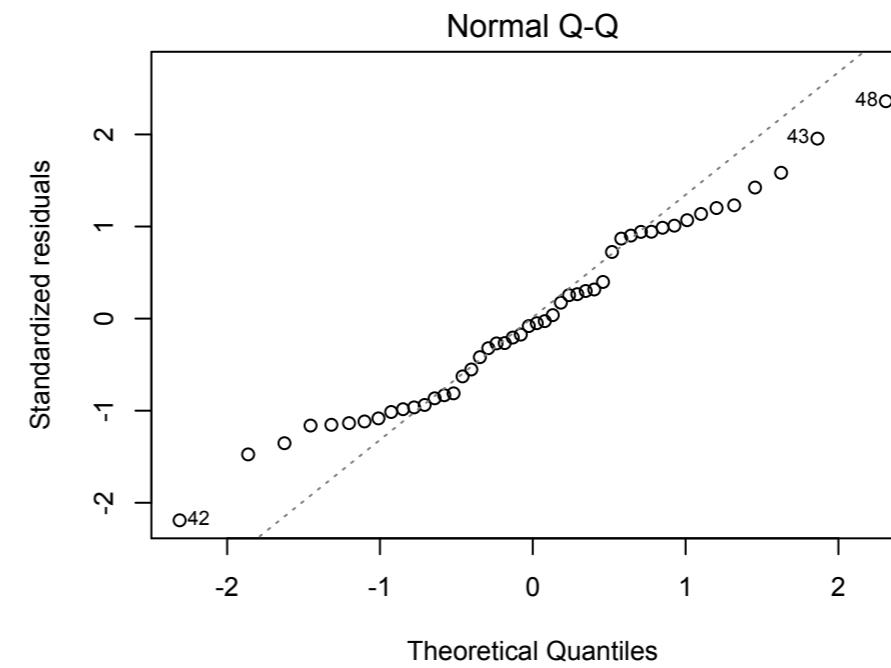
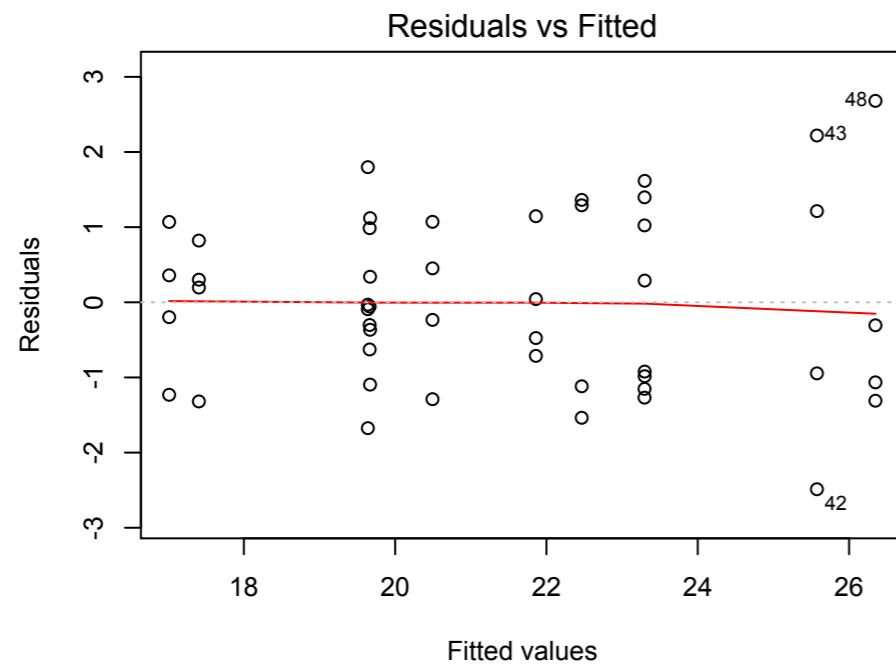
```
dd <- mutate(dd, supplement = relevel(supplement, ref="control"))
with(dd, interaction.plot(supplement, diet, gain))
```



```

mm <- lm(gain ~ diet*supplement, data = dd)
par(mfrow=c(2, 2))
plot(mm)

```



ANOVA table

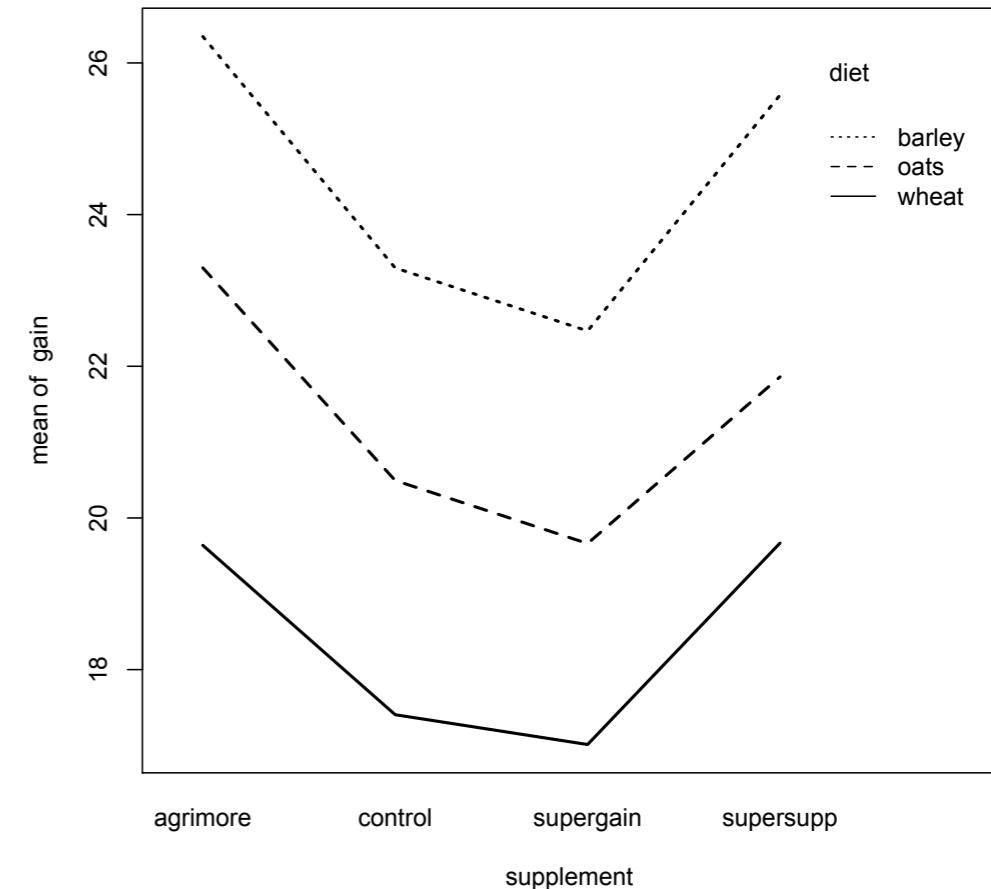
```
> anova(mm)
```

Analysis of Variance Table

Response: gain

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
diet	2	287.171	143.586	83.5201	2.999e-14	***
supplement	3	91.881	30.627	17.8150	2.952e-07	***
diet:supplement	6	3.406	0.568	0.3302	0.9166	
Residuals	36	61.890	1.719			

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1						



```
> summary(mm)
```

Call:
lm(formula = gain ~ diet * supplement, data = dd)

Residuals:
Min 1Q Median 3Q Max
-2.48756 -1.00368 -0.07452 1.03496 2.68069

Coefficients:

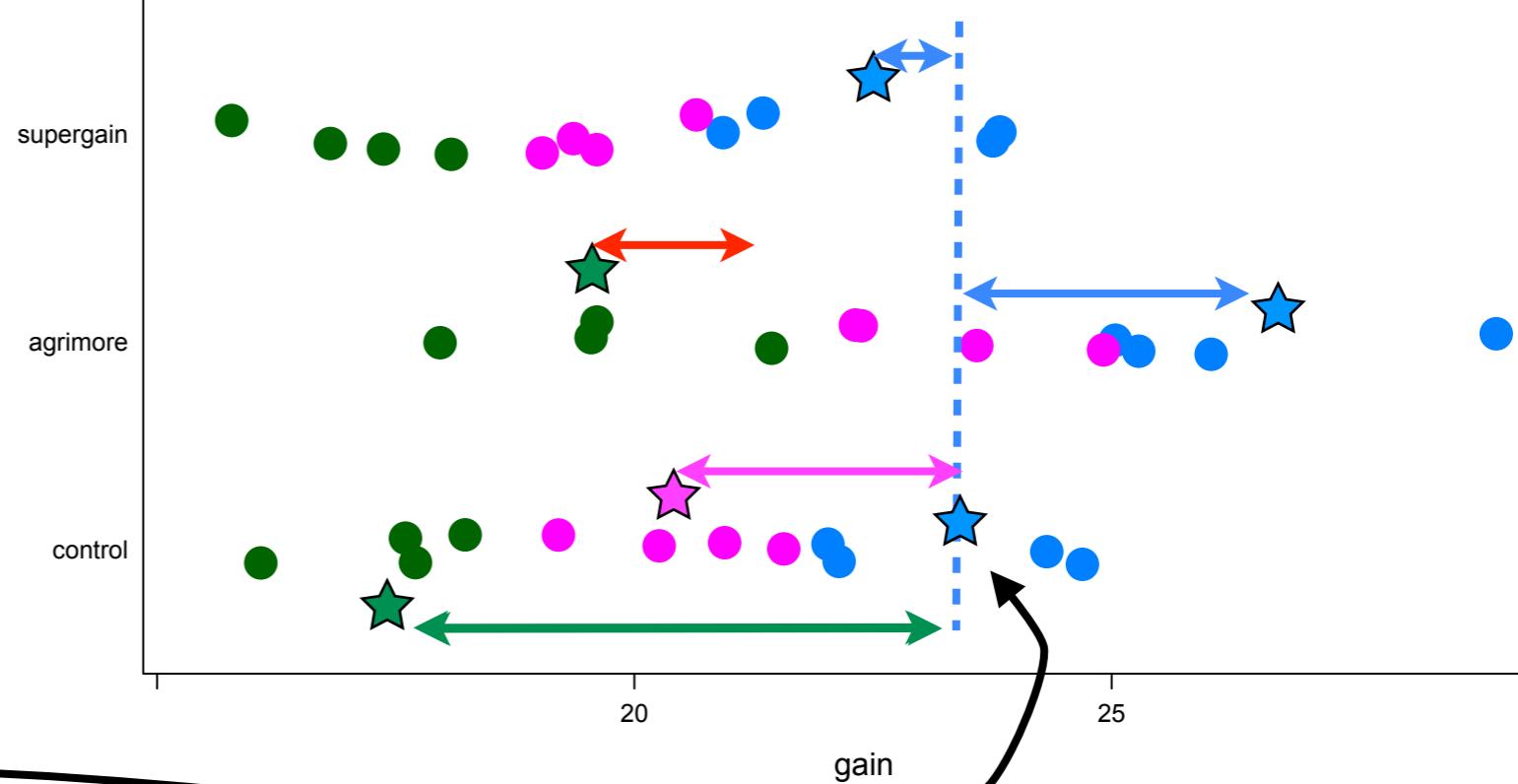
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	23.2966499	0.6555863	35.536	< 2e-16	***
dietoats	-2.8029851	0.9271390	-3.023	0.00459	**
dietwheat	-5.8911317	0.9271390	-6.354	2.34e-07	***
supplementagrimore	3.0518277	0.9271390	3.292	0.00224	**
supplementsupergain	-0.8305263	0.9271390	-0.896	0.37631	
supplementsupersupp	2.2786527	0.9271390	2.458	0.01893	*
dietoats:supplementagrimore	-0.2471088	1.3111726	-0.188	0.85157	
dietwheat:supplementagrimore	-0.8182729	1.3111726	-0.624	0.53651	
dietoats:supplementsupergain	-0.0001351	1.3111726	0.000	0.99992	
dietwheat:supplementsupergain	0.4374395	1.3111726	0.334	0.74060	
dietoats:supplementsupersupp	-0.9120830	1.3111726	-0.696	0.49113	
dietwheat:supplementsupersupp	-0.0158299	1.3111726	-0.012	0.99043	

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.311 on 36 degrees of freedom

Multiple R-squared: 0.8607, Adjusted R-squared: 0.8182

F-statistic: 20.22 on 11 and 36 DF, p-value: 3.295e-12



barley blue
oats magenta
wheat green