Introduction to Linear Models

Levi Waldron, CUNY School of Public Health

July 11, 2016

Outline for Introduction to Linear Models

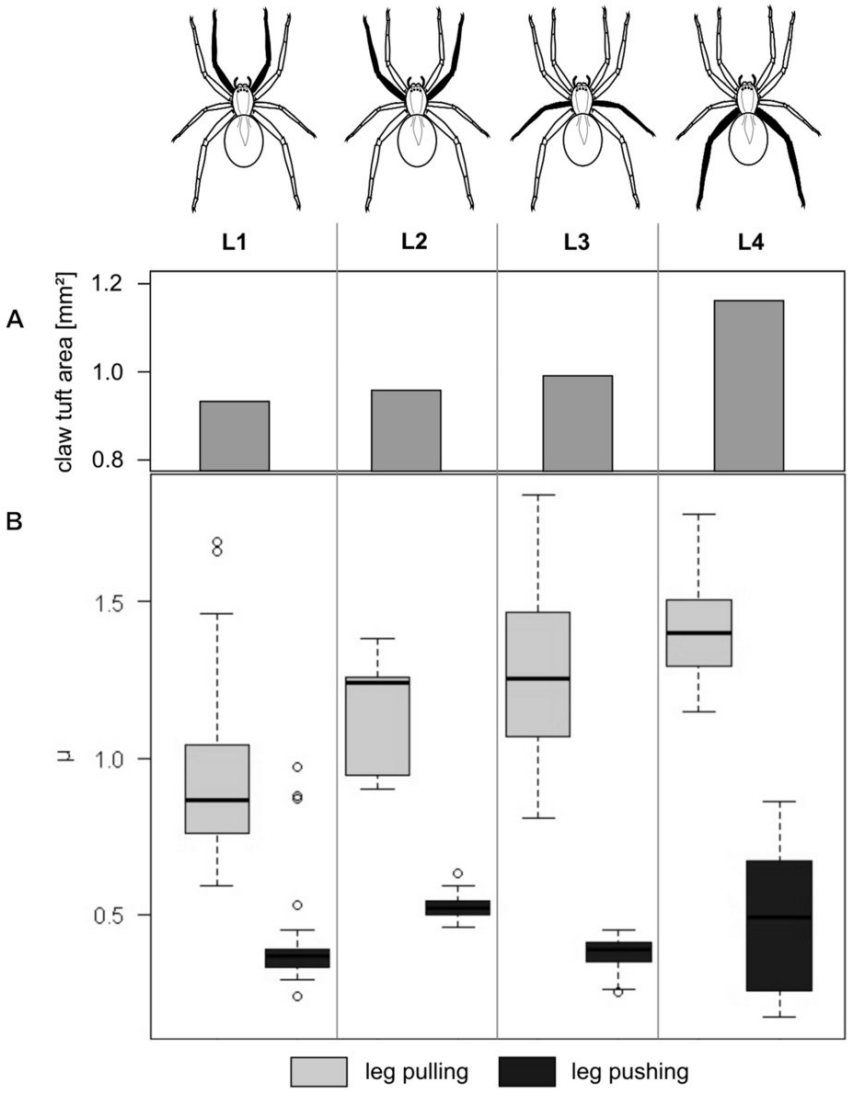
Based on Love and Irizarry, Data Analysis for the Life Sciences, Chapter 5

- Multiple linear regression
  - *Continuous and categorical predictors*
  - *Interactions*
- Model formulae
- Design matrix
- Analysis of Variance

Introduction to Linear Models

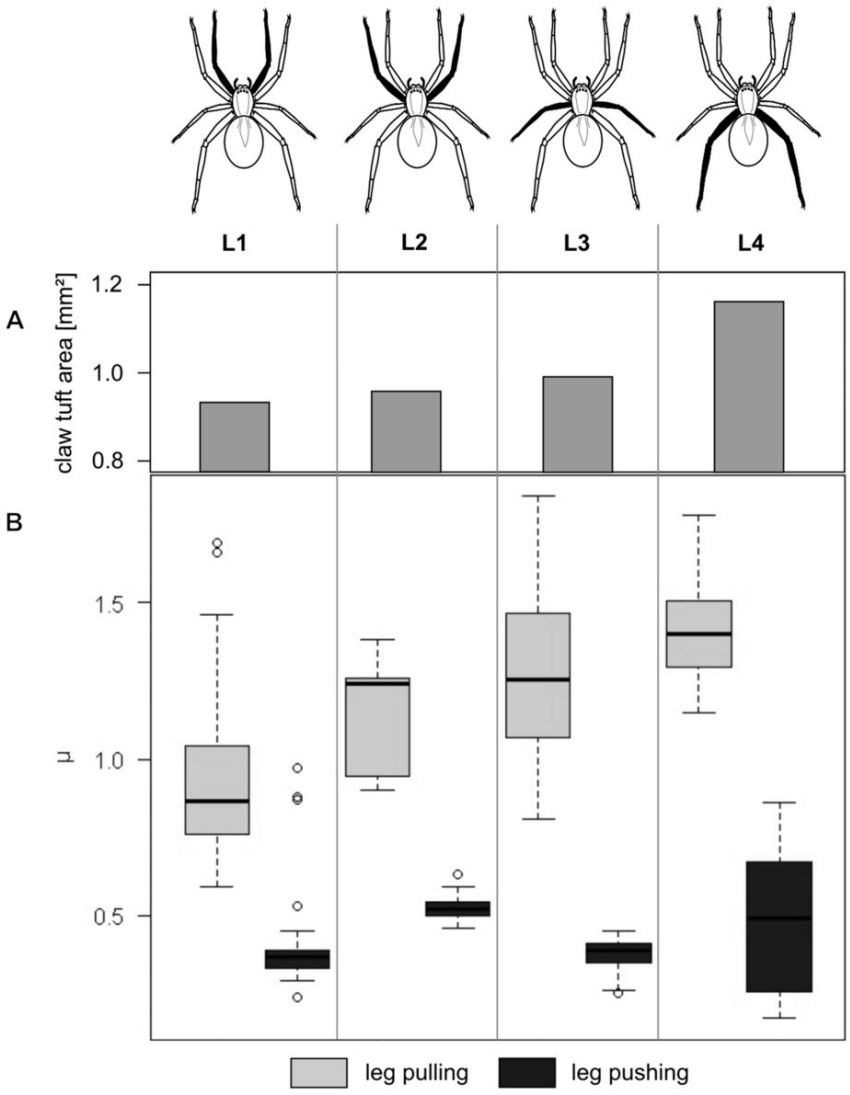
Example: friction of spider legs

- Wolff & Gorb, Radial arrangement of Janus-like setae permits friction control in spiders, *Sci. Rep.* 2013.



- **(A)** Barplot showing total claw tuft area of the corresponding legs.
- **(B)** Boxplot presenting friction coefficient data illustrating median, interquartile range and extreme values.

Example: friction of spider legs



- Are the pulling and pushing friction coefficients different?
- Are the friction coefficients different for the different leg pairs?
- Does the difference between pulling and pushing friction coefficients vary by leg pair?

Example: friction of spider legs

```
table(spider$leg,spider$type)
```

```
##
##       pull push
##    L1   34   34
##    L2   15   15
##    L3   52   52
##    L4   40   40
```

```
summary(spider)
```

```
##  leg        type        friction
##  L1: 68   pull:141   Min.   :0.1700
##  L2: 30   push:141   1st Qu.:0.3900
##  L3:104              Median :0.7600
##  L4: 80              Mean   :0.8217
##                      3rd Qu.:1.2400
##                      Max.   :1.8400
```

What are linear models?

- Linear models model a response variable Y_i as a linear combination of predictors, plus randomly distributed noise.
- Which of the following are examples of linear models?

1. $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
2. $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$
3. $y_i = \beta_0 + \beta_1 x_i + \times 2^{\beta_2 x_i} + \varepsilon_i$

Where: $i = 1, \ldots, N$

Assumption: $\varepsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$

What are linear models?

The following are examples of linear models:

1. $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ (simple linear regression)
2. $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$ (quadratic regression)

Multiple linear regression model

- Linear models can have any number of predictors
- Systematic part of model:

$$E[y|x] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

- $E[y|x]$ is the expected value of y given x
- y is the outcome, response, or dependent variable
- x is the vector of predictors / independent variables
- x_p are the individual predictors or independent variables
- β_p are the regression coefficients

Multiple linear regression model

Random part of model:

$$y_i = E[y_i|x_i] + \epsilon_i$$

Assumptions of linear models: $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$

- Normal distribution
- Mean zero at every value of predictors
- Constant variance at every value of predictors
- Values that are statistically independent

Continuous predictors

- **Coding:** as-is, or may be scaled to unit variance (which results in *adjusted* regression coefficients)
- **Interpretation for linear regression:** An increase of one unit of the predictor results in this much difference in the continuous outcome variable

Binary predictors (2 levels)

- **Coding:** indicator or dummy variable (0-1 coding)
- **Interpretation for linear regression:** the increase or decrease in average outcome levels in the group coded "1", compared to the reference category ("0")
- e.g. $E(y|x) = \beta_0 + \beta_1 x$
- where x={ 1 if push friction, 0 if pull friction }

Multilevel categorical predictors (ordinal or nominal)

- **Coding:** $K - 1$ dummy variables for K-level categorical variable
- Comparisons with respect to a reference category, *e.g.* `L1`:
  - *`L2`*={*1 if* 2^{nd} *leg pair, 0 otherwise*},
  - *`L3`*={*1 if* 3^{nd} *leg pair, 0 otherwise*},
  - *`L4`*={*1 if* 4^{th} *leg pair, 0 otherwise*}.
- R re-codes factors to dummy variables automatically.
- Note that factors can be *ordered* or *unordered*

Model formulae in R

Model formulae in R

Model formulae tutorial

- regression functions in R such as `aov()`, `lm()`, `glm()`, and `coxph()` use a "model formula" interface.
- The formula determines the model that will be built (and tested) by the R procedure. The basic format is:

```
> response variable ~ explanatory variables
```

- The tilde means "is modeled by" or "is modeled as a function of."

Regression with a single predictor

Model formula for simple linear regression:

```
> y ~ x
```

- where “x” is the explanatory (independent) variable
- “y” is the response (dependent) variable.

Return to the spider legs

Friction coefficient for leg type of first leg pair:

```
spider.sub <- spider[spider$leg=="L1", ]
fit <- lm(friction ~ type, data=spider.sub)
summary(fit)
```

```
##
## Call:
## lm(formula = friction ~ type, data = spider.sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33147 -0.10735 -0.04941 -0.00147  0.76853
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.92147    0.03827  24.078  < 2e-16 ***
## typepush    -0.51412    0.05412  -9.499  5.7e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2232 on 66 degrees of freedom
## Multiple R-squared:  0.5776, Adjusted R-squared:  0.5711
## F-statistic: 90.23 on 1 and 66 DF,  p-value: 5.698e-14
```

Regression on spider leg type

Regression coefficients for `friction ~ type` for first set of spider legs:

```
fit.table <- xtable::xtable(fit, label=NULL)
print(fit.table, type="html")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.9215	0.0383	24.08	0.0000
typepush	-0.5141	0.0541	-9.50	0.0000

- How to interpret this table?
  - *Coefficients for **(Intercept)** and **typepush***
  - *Coefficients are t-distributed when assumptions are correct*
  - *Standard Error is the sampling variance of the estimates*

Interpretation of coefficients

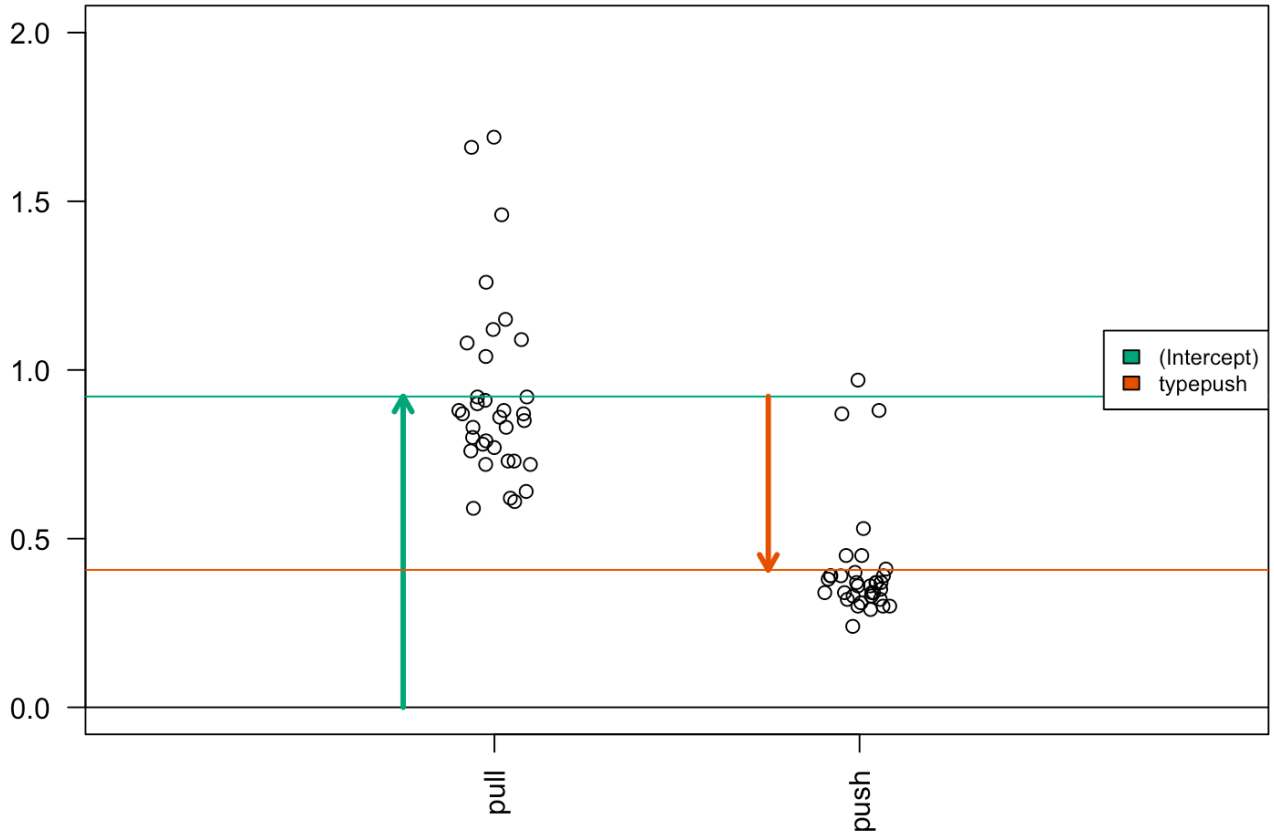


Diagram of the estimated coefficients in the linear model. The green arrow indicates the Intercept term, which goes from zero to the mean of the reference group (here the 'pull' samples). The orange arrow indicates the difference between the push group and the pull group, which is negative in this example. The circles show the individual samples, jittered horizontally to avoid overplotting.

Regression on spider leg position

Remember there are positions 1-4

```
fit <- lm(friction ~ leg, data=spider)
```

```
fit.table <- xtable::xtable(fit, label=NULL)
print(fit.table, type="html")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.6644	0.0538	12.34	0.0000
legL2	0.1719	0.0973	1.77	0.0784
legL3	0.1605	0.0693	2.32	0.0212
legL4	0.2813	0.0732	3.84	0.0002

- Interpretation of the dummy variables legL2, legL3, legL4 ?

Regression with multiple predictors

Additional explanatory variables can be added as follows:

```
> y ~ x + z
```

Note that "+" does not have its usual meaning, which would be achieved by:

```
> y ~ I(x + z)
```

Regression on spider leg type and position

Remember there are positions 1-4

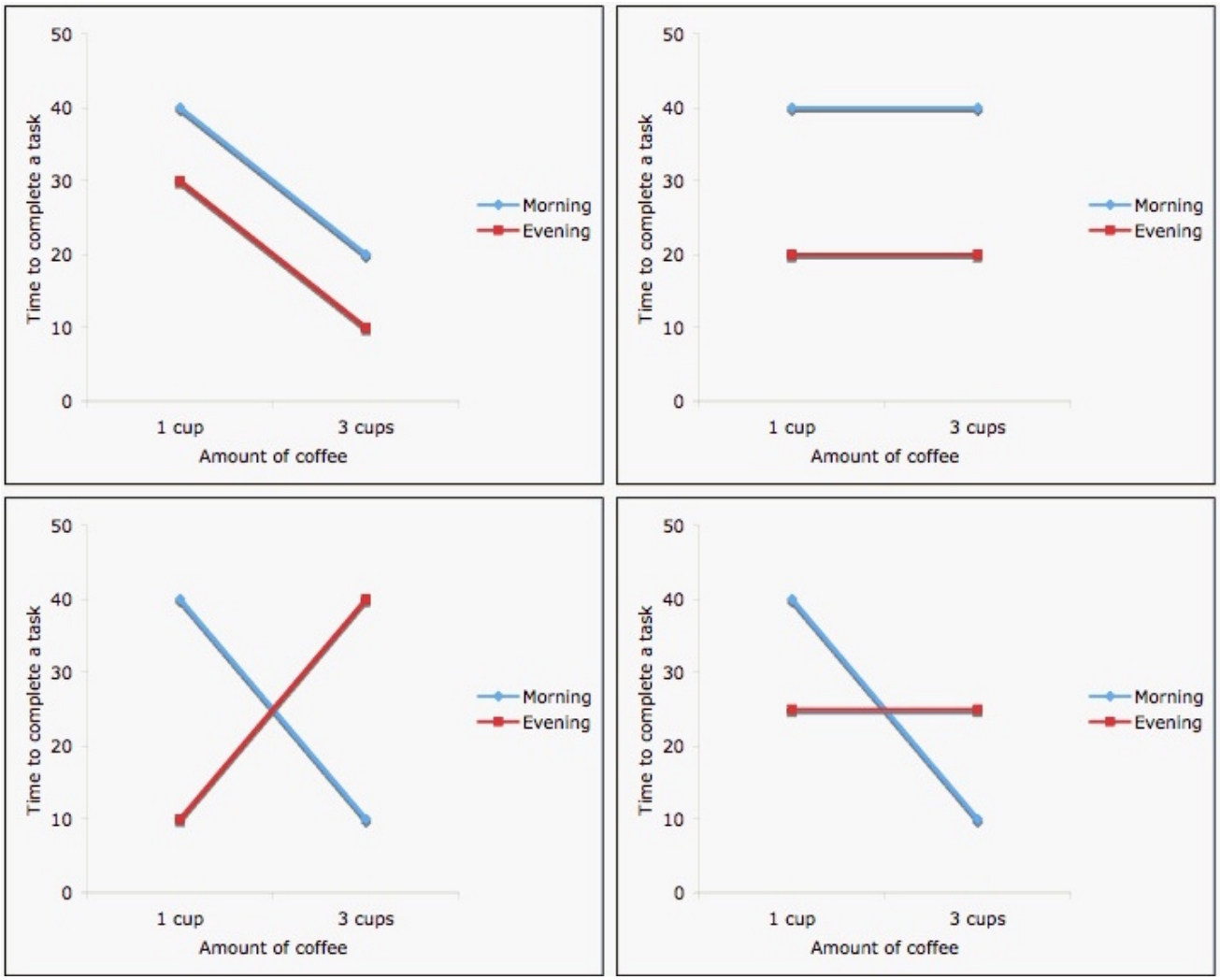
```
fit <- lm(friction ~ type + leg, data=spider)
```

```
fit.table <- xtable::xtable(fit, label=NULL)
print(fit.table, type="html")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.0539	0.0282	37.43	0.0000
typepush	-0.7790	0.0248	-31.38	0.0000
legL2	0.1719	0.0457	3.76	0.0002
legL3	0.1605	0.0325	4.94	0.0000
legL4	0.2813	0.0344	8.18	0.0000

- this model still doesn't represent how the friction differences between different leg positions are modified by whether it is pulling or pushing

Interaction (effect modification)



Interaction between coffee and time of day on performance

Image credit: http://personal.stevens.edu/~ysakamot/

Interaction (effect modification)

Interaction is modeled as the product of two covariates:

$$E[y|x] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 * x_2$$

Summary: model formulae

symbol	example	meaning
+	+ x	include this variable
-	- x	delete this variable
:	x : z	include the interaction
*	x * z	include these variables and their interactions
^	(u + v + w)^3	include these variables and all interactions up to three way
I	-I	intercept: delete the intercept

Summary: types of standard linear models

```
lm( y ~ u + v)
```

u and v factors: **ANOVA**
u and v numeric: **multiple regression**
one factor, one numeric: **ANCOVA**

- R does a lot for you based on your variable classes
  - *be **sure** you know the classes of your variables*
  - *be sure all rows of your regression output make sense*

The Design Matrix

The Design Matrix

Recall the multiple linear regression model:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_p x_{pi} + \epsilon_i$$

- x_{ji} is the value of predictor x_j for observation i

The Design Matrix

Matrix notation for the multiple linear regression model:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_N \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix}$$

or simply:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

- The design matrix is $\mathbf{X}$
  - *which the computer will take as a given when solving for $\boldsymbol{\beta}$ by minimizing the sum of squares of residuals $\boldsymbol{\varepsilon}$.*

Choice of design matrix

- there are multiple possible and reasonable design matrices for a given study design
- the model formula encodes a default model matrix, e.g.:

```
group <- factor( c(1, 1, 2, 2) )
model.matrix(~ group)
```

```
##   (Intercept) group2
## 1           1      0
## 2           1      0
## 3           1      1
## 4           1      1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

Choice of design matrix

What if we forgot to code group as a factor?

```
group <- c(1, 1, 2, 2)
model.matrix(~ group)
```

```
##   (Intercept) group
## 1           1     1
## 2           1     1
## 3           1     2
## 4           1     2
## attr(,"assign")
## [1] 0 1
```

More groups, still one variable

```
group <- factor(c(1,1,2,2,3,3))
model.matrix(~ group)
```

```
##   (Intercept) group2 group3
## 1           1      0      0
## 2           1      0      0
## 3           1      1      0
## 4           1      1      0
## 5           1      0      1
## 6           1      0      1
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

Changing the baseline group

```
group <- factor(c(1,1,2,2,3,3))
group <- relevel(x=group, ref=3)
model.matrix(~ group)
```

```
##   (Intercept) group1 group2
## 1           1      1      0
## 2           1      1      0
## 3           1      0      1
## 4           1      0      1
## 5           1      0      0
## 6           1      0      0
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

More than one variable

```
diet <- factor(c(1,1,1,1,2,2,2,2))
sex <- factor(c("f","f","m","m","f","f","m","m"))
model.matrix(~ diet + sex)
```

```
##   (Intercept) diet2 sexm
## 1           1     0    0
## 2           1     0    0
## 3           1     0    1
## 4           1     0    1
## 5           1     1    0
## 6           1     1    0
## 7           1     1    1
## 8           1     1    1
## attr(,"assign")
## [1] 0 1 2
## attr(,"contrasts")
## attr(,"contrasts")$diet
## [1] "contr.treatment"
##
## attr(,"contrasts")$sex
## [1] "contr.treatment"
```

With an interaction term

```
model.matrix(~ diet + sex + diet:sex)
```

```
##   (Intercept) diet2 sexm diet2:sexm
## 1           1     0    0          0
## 2           1     0    0          0
## 3           1     0    1          0
## 4           1     0    1          0
## 5           1     1    0          0
## 6           1     1    0          0
## 7           1     1    1          1
## 8           1     1    1          1
## attr(,"assign")
## [1] 0 1 2 3
## attr(,"contrasts")
## attr(,"contrasts")$diet
## [1] "contr.treatment"
##
## attr(,"contrasts")$sex
## [1] "contr.treatment"
```

Design matrix to contrast what we want

- Spider leg friction example:
  - *The question of whether push vs. pull difference is different in L2 compared to L1 is answered by the term `typepush:legL2` in a model with interaction terms:*

```
fitX <- lm(friction ~ type * leg, data=spider)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.9215	0.0327	28.21	0.0000
typepush	-0.5141	0.0462	-11.13	0.0000
legL2	0.2239	0.0590	3.79	0.0002
legL3	0.3524	0.0420	8.39	0.0000
legL4	0.4793	0.0444	10.79	0.0000
typepush:legL2	-0.1039	0.0835	-1.24	0.2144
typepush:legL3	-0.3838	0.0594	-6.46	0.0000
typepush:legL4	-0.3959	0.0628	-6.30	0.0000

**What if we want to ask this question for L3 vs L2?

Design matrix to contrast what we want

What if we want to contrast…

`typepush:legL3 - typepush:legL2`

There are many ways to construct this design, one is with `library(multcomp)`:

```
names(coef(fitX))
```

```
## [1] "(Intercept)"    "typepush"        "legL2"           "legL3"
## [5] "legL4"          "typepush:legL2" "typepush:legL3" "typepush:legL4"
```

```
C <- matrix(c(0,0,0,0,0,-1,1,0), 1)
L3vsL2interaction <- multcomp::glht(fitX, linfct=C)
```

Design matrix to contrast what we want

Is there a difference in pushing friction for L3 vs L2?

```
summary(L3vsL2interaction)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = friction ~ type * leg, data = spider)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0 -0.27988    0.07893  -3.546  0.00046 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

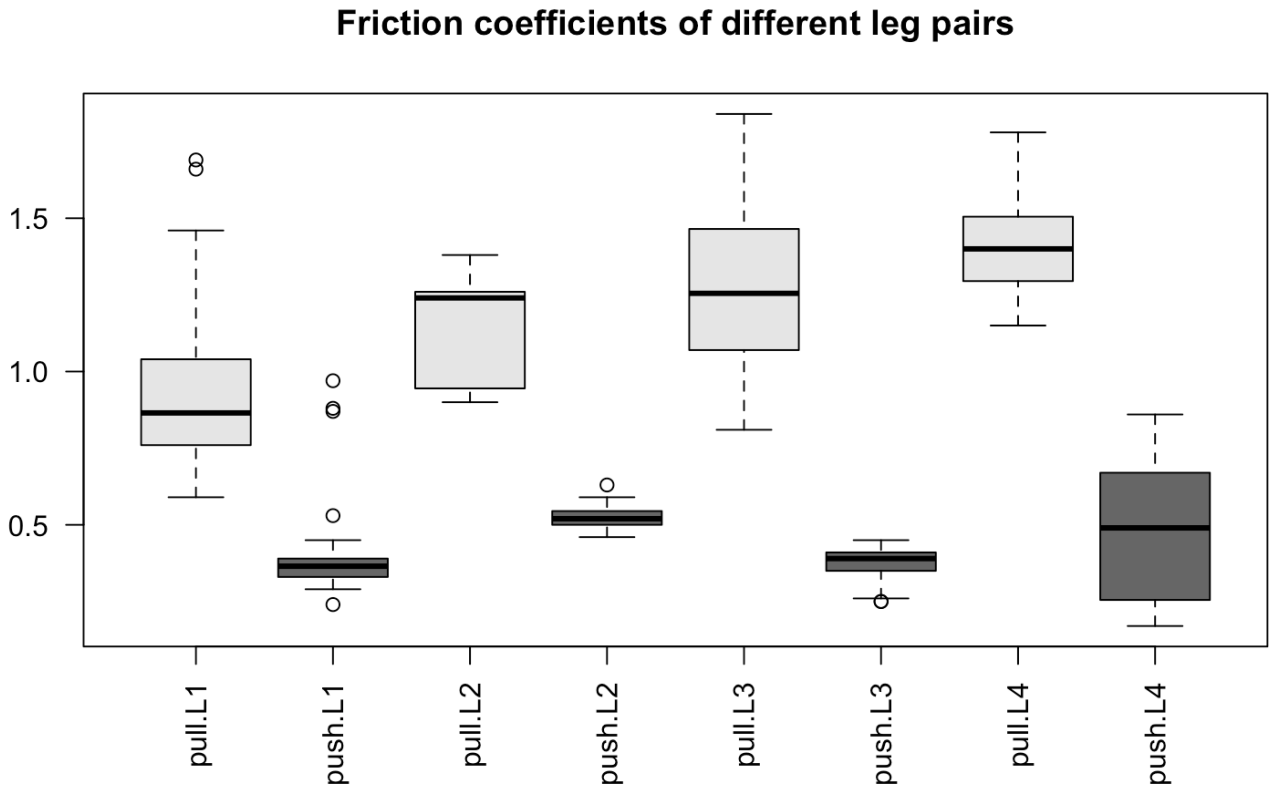
Summary: applications of model matrices

- Major differential expression packages recognize them:
  - *LIMMA (VOOM for RNA-seq)*
  - *DESeq2 for all kinds of count data*
  - *EdgeR*
- Can fit coefficients **directly** to your contrast of interest
  - *e.g.: what is the difference between push/pull friction for each spider-leg pair?*

Analysis of Variance

Why Analysis of Variance?

- Analysis of Variance allows inference on the inclusion of a categorical or continuous variable
  - *not just on re-coded "dummy" variables (e.g. for each spider leg pair)*



Compare ANOVA table to regression table

```
print(xtable::xtable(summary(fit)), type="html")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.0539	0.0282	37.43	0.0000
typepush	-0.7790	0.0248	-31.38	0.0000
legL2	0.1719	0.0457	3.76	0.0002
legL3	0.1605	0.0325	4.94	0.0000
legL4	0.2813	0.0344	8.18	0.0000

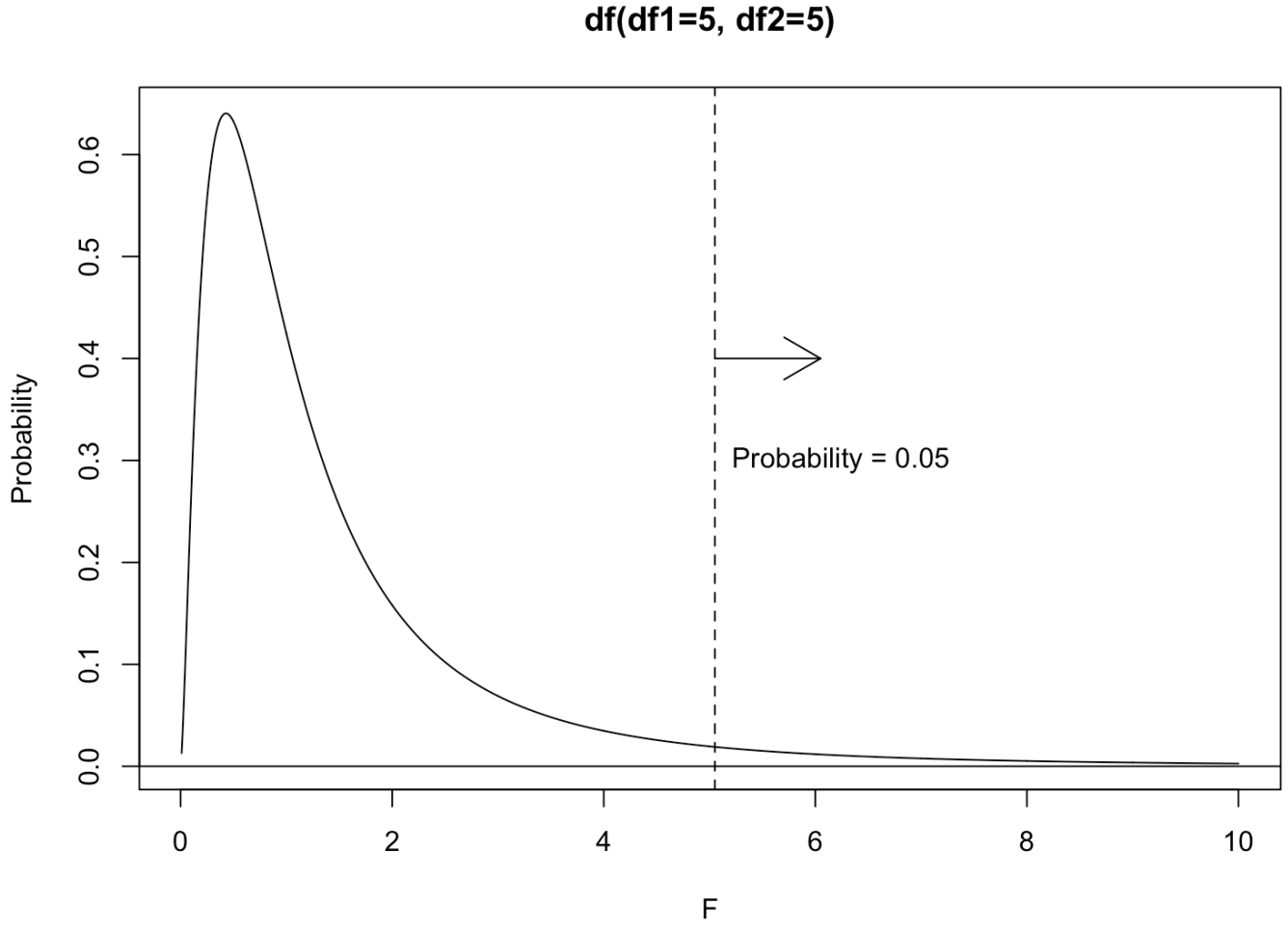
```
print(xtable::xtable(anova(fit)), type="html")
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
type	1	42.78	42.78	984.73	0.0000
leg	3	2.92	0.97	22.41	0.0000
Residuals	277	12.03	0.04		

$$F = \frac{\text{variability between groups}}{\text{variability within groups}} = \frac{\text{reduction in variance from adding variable}}{\text{variance of residuals}}$$

Analysis of Variance: F test

- Compares *between* group variance to *within* group variance
  - $F = \frac{\text{variability between groups}}{\text{variability within groups}} = \frac{\text{reduction in variance from adding variable}}{\text{variance of residuals}}$
- The F distribution depends on both *numerator* (df1) and *denominator* (df2) degrees of freedom
- Rejection region is in the right tail only:



Summary

- Linear models are the basis for identifying differential expression / differential abundance
  - *continuous Y; any kind of X variables*
- **Assumptions**:
  1. *normal, homoscedastic errors,*
  2. *a linear relationship, and*
  3. *independent observations.*
- Note that **t** and **F** tests are *robust* and *conservative* to violations of 1 and 2
  - *extremely so for $n > 30$*

Summary (cont'd)

- Know the model formula interface, but
  - *use model matrices to directly fit coefficients that you want to interpret*
- **Generalized Linear Models** extend these methods to:
  - *binary Y (logistic regression)*
  - *count Y (log-linear regression with e.g. Poisson or Negative Binomial link functions)*

Links

- A built html version of this lecture is available.
- The source R Markdown is also available from Github.