

Linear Models 2: Logistic and Poisson (count) regression-an introduction to Generalised Linear Models (GLM)

Presented by
Chris Howden
Sydney Informatics Hub
Core Research Facilities
The University of Sydney



THE UNIVERSITY OF
SYDNEY

Acknowledging SIH



All University of Sydney resources are available to Sydney researchers **free of charge**. The use of the SIH services including the Artemis HPC and associated support and training warrants acknowledgement in any publications, conference proceedings or posters describing work facilitated by these services.

The continued acknowledgment of the use of SIH facilities ensures the sustainability of our services.

Suggested wording:

General acknowledgement:

"The authors acknowledge the technical assistance provided by the Sydney Informatics Hub, a Core Research Facility of the University of Sydney."

Acknowledging specific staff:

"The authors acknowledge the technical assistance of (name of staff) of the Sydney Informatics Hub, a Core Research Facility of the University of Sydney."

For further information about acknowledging the Sydney Informatics Hub, please contact us at sih.info@sydney.edu.au.

We value your feedback



- We aim to help HDR students and researchers in a wide range of fields across different faculties
- We want to hear about **you** and whether this workshop has helped you in your research.
- Later in this workshop there will be a link to a survey
- It only takes a few minutes to complete (*really!*)
- Completing this survey will help us create workshops that best meet the needs of researchers like you

During the workshop

- Ask short questions or clarifications during the workshop. There will be breaks during the workshop for longer questions.



- Slides with this blackboard icon are mainly for your reference, and the material will not be discussed during the workshop.

Challenge Question

- A wild boar is coming towards you at 200mph. Do you:?
 - A. Ask it directions
 - B. Wave a red flag
 - C. Wave a white flag
 - D. Begin preparing a trap



After the workshop

These slides should be used after the workshop as **Workflows** and reference material.

- Today's workshop gives you the **statistical workflow**, which is software agnostic in that they can be applied in any software.
- There are also accompanying **software workflows** that show you how to do it. We won't be going through these in detail. But if you have problems we have a monthly hacky hour where people can help you.

1 on 1 assistance

- You can email us about the material in these workshops at any time
- Or request a consultation for more in-depth discussion of the material as it relates to your specific project. Consults can be requested via our Webpage (link is at the end of this presentation)

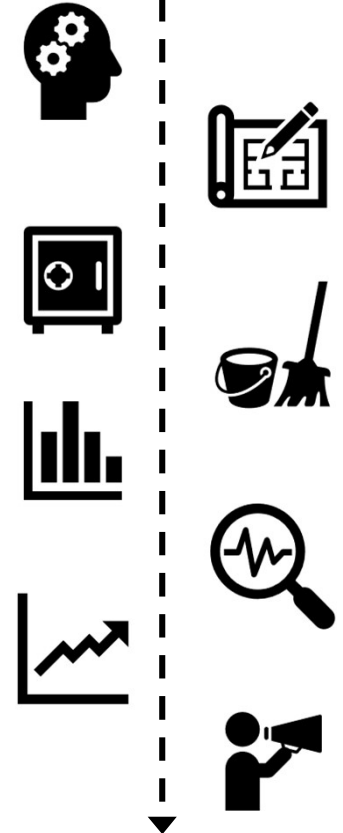
Research Workflow

- Why do we use a research workflow?
 - As researchers we are motivated to find answers *quickly*
 - This drive can cause problems if we don't think systematically
 - ... and we need to in order to:
 - Find the right method
 - Use it correctly
 - Interpret and report our results accurately
 - The payoff is huge, we can avoid mistakes that would affect the quality of our work *and* get to the answers sooner
- So... what is a workflow?
 - The process of doing a statistical analysis follows the same general “shape”.
 - We provide a general research workflow, and a specific workflow for each major step in your research
(currently **experimental design, power calculation, analysis using linear models/survival/multivariate/survey methods**)
 - You will need to tweak them to your needs



General Research Workflow

1. **Hypothesis Generation** (Research/Desktop Review)
2. **Experimental and Analytical Design** (sampling, power, ethics approval)
3. **Collect/Store Data**
4. **Data cleaning**
5. **Exploratory Data Analysis (EDA)**
6. **Data Analysis aka inferential analysis**
7. **Predictive modelling**
8. **Publication**



CONTENTS: Generalised Linear Models II

First we will explain the Generalised Linear Model Framework and how it is just an extension of the Simple Linear Framework introduced in Workshop I.

Statistical Workflows for:

- Logistic (binary) regression
- Poisson (count) regression

These workflows are software agnostic but also have accompanying R code if you wish to do it in R. Plots are done using a combination of default plotting functions and ggplot functions. You will know the difference since ggplot functions start with ggplot().

Generalized Linear Models Framework

Simple Linear Models (workshop 1) vs Generalized Linear Models (workshop 2+)

Introducing the concepts of:

- Design Matrix
- Linear Predictor
- Data Distribution
- Link Function



What are Generalised Linear Models?

ANOVA

Linear Regression

ANCOVA

Logistic (Binary) regression

Before After Control
Impact (BACI) Studies

Count (Poisson) regression

Repeated measures

Randomised Control
Trials (RCT's)

Plus Many More!!

A single unifying Theory

In Linear Models I we showed that although Regression and ANOVA are often taught as different things, they aren't. Instead it's much easier to understand them using a single unifying Linear Models theory.

This allows us to apply them using the same workflow.

In this workshops we extend this theory to allow non normal (gaussian) errors and responses. This extended theory is called:

Generalized Linear Models

We're gonna need some Equations

DON'T FREAK OUT!!!



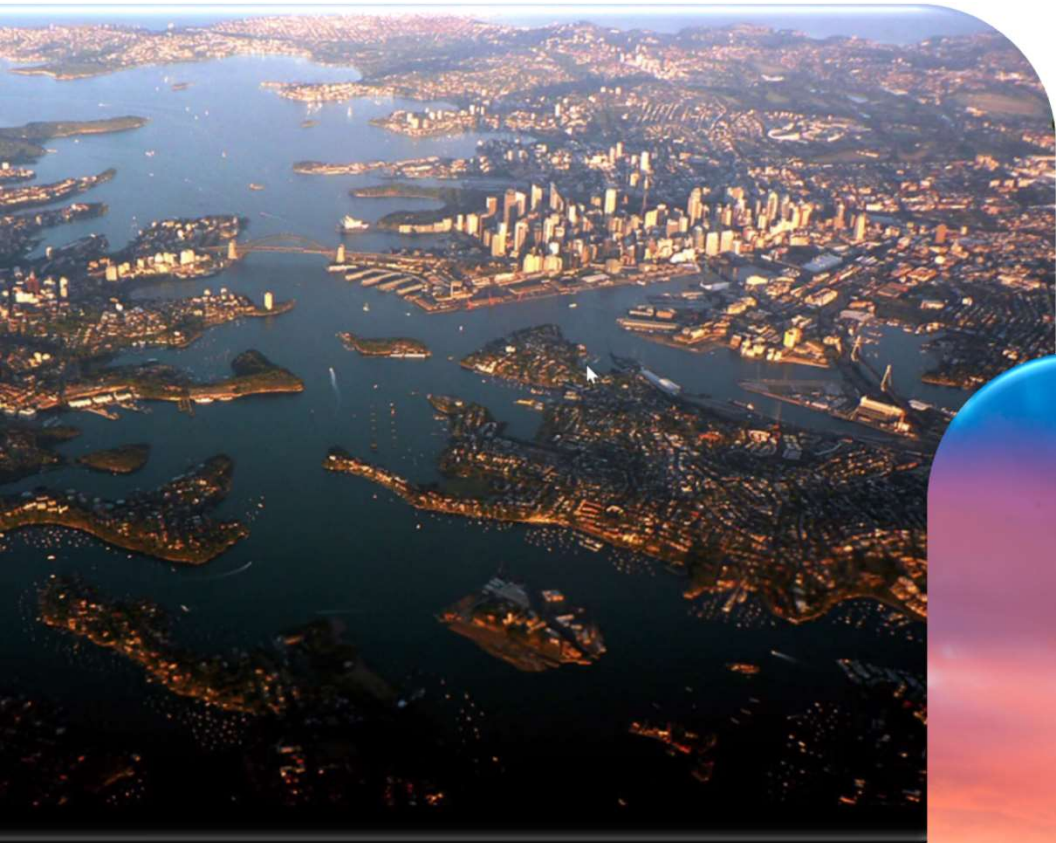
Couple tricks with equations:

- They are a language.
 - Each symbol represents a concept, so learn the concept to learn the equation.
 - Then write the equation out in your native tongue
- If you don't get the concept that's fine. Just work on it a little bit each day. Like any language.

EG: $Y_i = \beta_0 + \varepsilon_i$

- Means something called Y equals something called Beta zero plus some Error.

Don't get lost in the detail. Get the lay of the land at a **Conceptual** - big scale. And then come back and zoom into the detail when you have time.



We are covering a lot in this first section, so don't worry if you get a little lost.

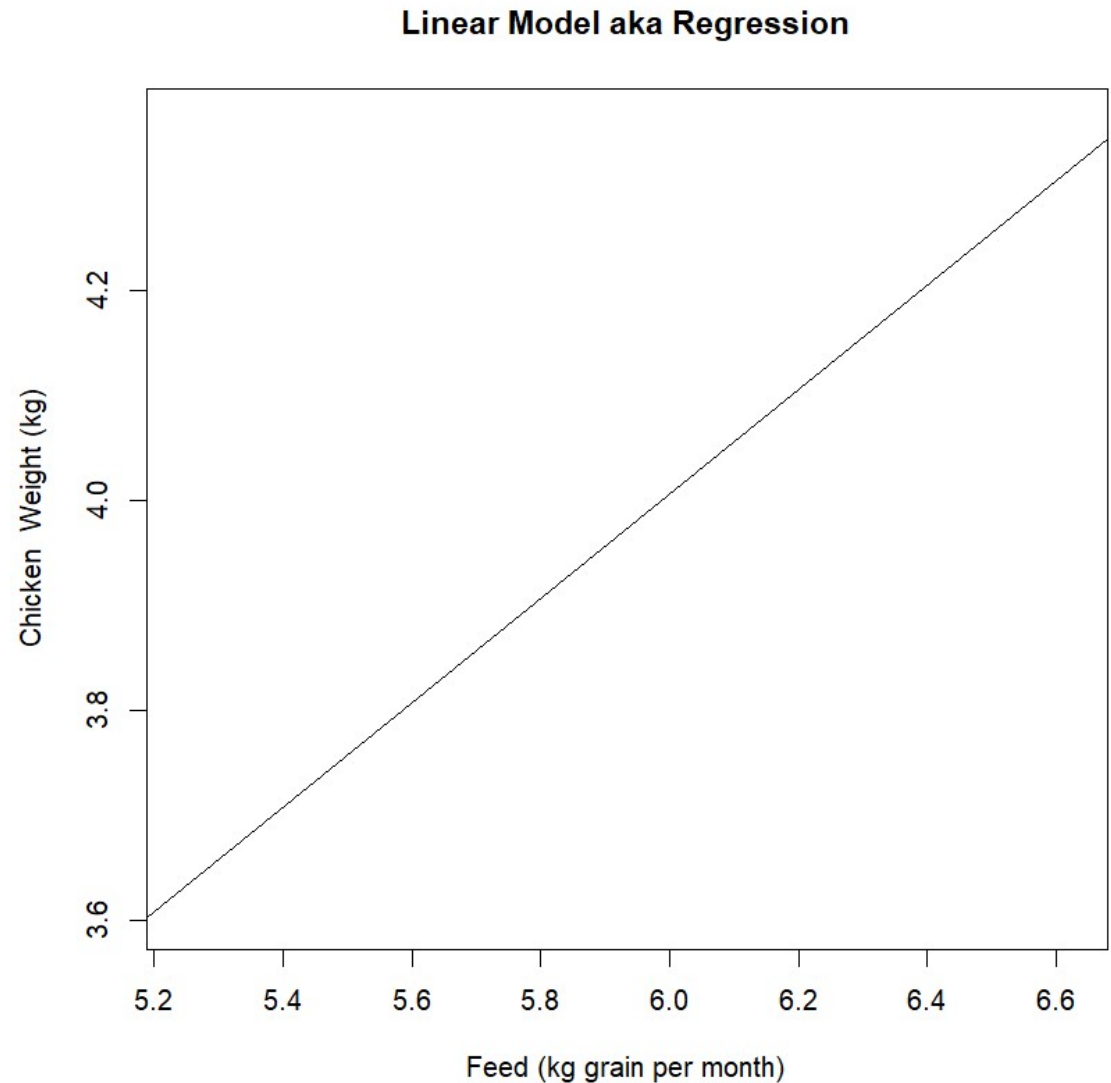
Just get the big picture, remember where you get lost and then come back and learn a little more each day.

If you can just get the take homes in these red boxes today that's a great start.

Simple Linear Model

Your Turn: Draw a linear model for the weight of chicken compared to the amount of feed it eats in its first month.

So in this example a chicken that eats 6 kg of Feed will weigh about 4kg



So we know it's linear. Is that all we need to know?

NO! We want to know exactly how our Predictor (feed) affects our Response (weight).

And for that we need to fit an equation to the pictorial model you just drew so we can pull out the parameter that represents the Predictors affect on our Response.

High School Equation for a line

$Y = \text{slope (aka gradient)} * X + \text{Constant (aka Y intercept)}$

$$Y = mx + b$$

Statistical Equation for a line (puts the constant first)

$$\hat{Y}_i = \beta_0 + \beta_1 X_i$$

So we want to find β_1 , which is the slope (gradient) of the line and represents the effect Feed has on Weight. (β_0 is the constant)

But we're still missing something?

THE DATA!!!!

Each datum has its **own natural variance** from the line since each chicken is a bit different!

Another name for the Natural Variance is the “Error” of the model. Which is why we usually represent it as an ε in the model.

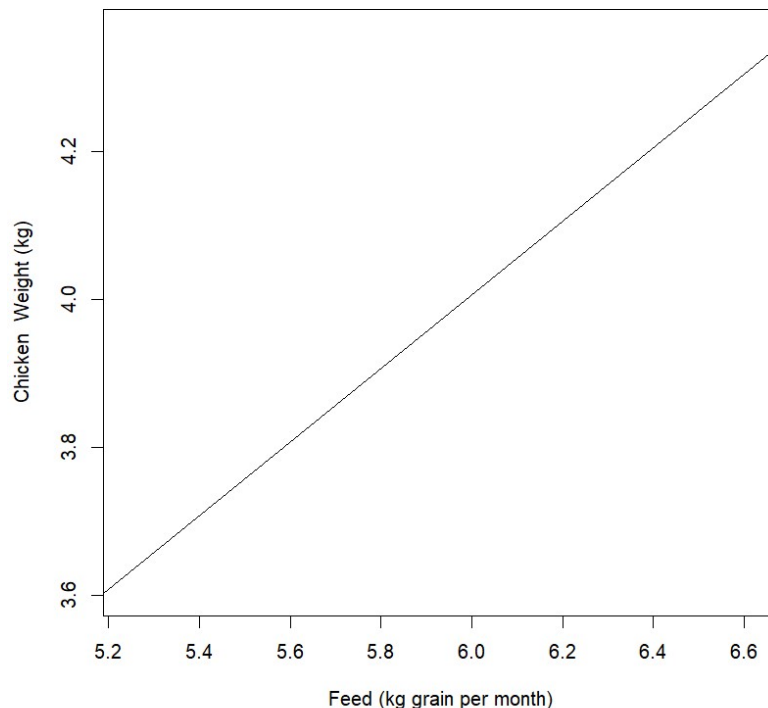
\hat{Y} ~ The “hat” over the \hat{Y} tells us that it's a **prediction** of Y for those specific predictor values for X .

Y ~ Is the **actual value** of Y , so it's the prediction + error.

MODEL FOR A LINE

$$\hat{Y}_i = \beta_0 + \beta_1 X_i$$

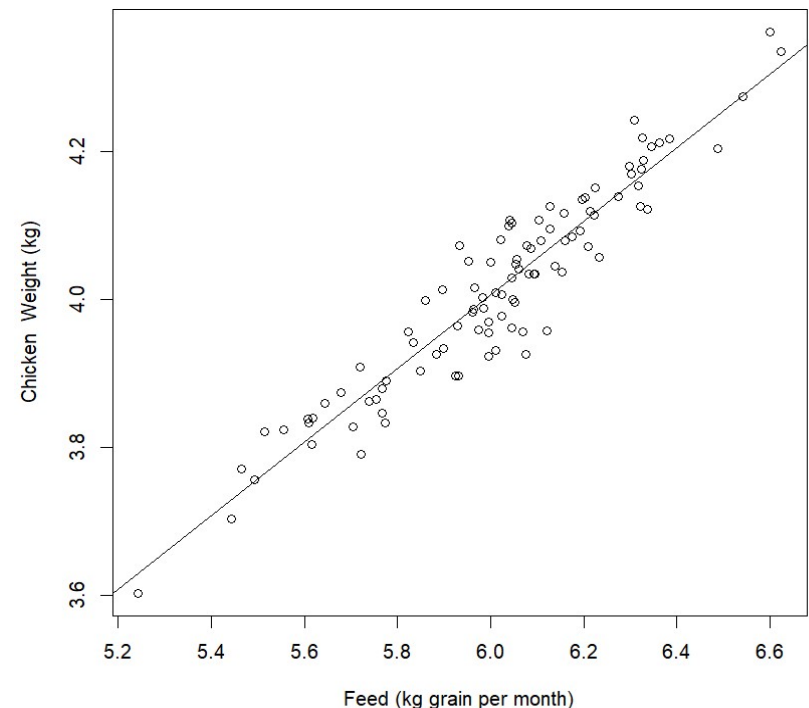
Linear Model aka Regression



MODEL FOR OUR DATA

$$Y_i = \hat{Y}_i + \varepsilon_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Linear Model aka Regression



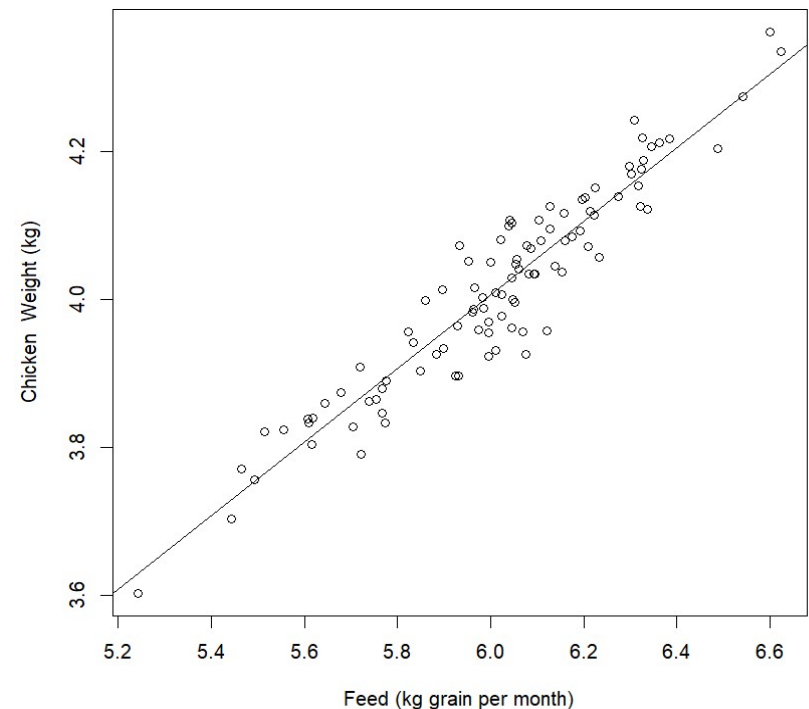
So let's look at all the different components of this equation so we can **generalise it to more complex models**. Such as:

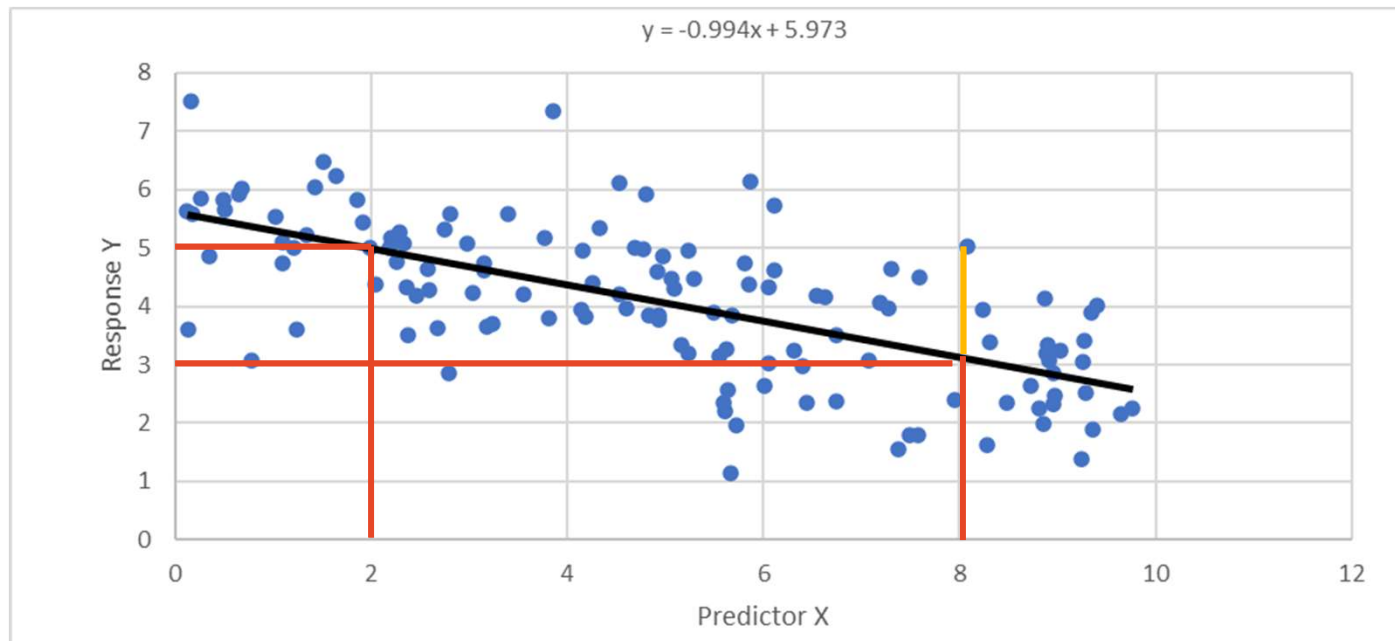
- More than 1 continuous variable
- Categorical Variables
- Non normal error

MODEL FOR OUR DATA

$$Y_i = \hat{Y}_i + \varepsilon_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Linear Model aka Regression





- The **blue points** are our data.
- The **black line** is the regression line we use to predict.
 - The prediction of Y is **conditional** on X .
 - The **red lines** are some example predictions along the line. So our prediction when $X=2$ is that $Y=5$. When $X=8$ we predict $Y=3$.
- The **orange line** is the error for the specific blue point $X=8, Y=5$. So although we predict $Y=3$, this particular point has $Y=5$. So an error of 2 above the line i.e. $Y = \hat{Y} + \varepsilon$ so $\varepsilon = Y - \hat{Y} = 5 - 3 = 2$.

Simple Regression – Numeric Statistical Model

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \varepsilon_i$$

Prediction = Linear Predictor + Error/Natural Variation

Data			Design Matrix Parameters		Model Variables	
Observation	Response	Predictors			Prediction	Error
i	Y _i	Continuous X _{1i}	X _{0i}	X _{1i}	\hat{Y}_i	ε_i
1	4	4	1	4	4.6	-0.6
2	4	8	1	8	4.7	-0.7
3	6	1	1	1	5.1	0.9
4	3	9	1	9	2.1	0.9
5	2	1	1	1	2.9	-0.9
6	2	7	1	7	2.5	-0.5

Data (the actual data you collect)

$Y_i \sim$ **Response** of Observation i

$X_{1i} \sim$ **Predictor** X_1 of Observation i

Design Matrix Parameters (the parameters in your model i.e. the actual data you model)

$X_{0i} \sim$ design parameter for parameter β_0 (Constant/Y intercept)

$X_{1i} \sim$ design parameter for β_1 (parameter X_{1i})

Model Variables (variables the model calculates)

$\hat{Y}_i \sim$ **Prediction** for Observation i

$\varepsilon_i \sim$ **Error** of Observation i

$\beta_0 \sim$ Constant/Y intercept parameter

$\beta_{1i} \sim$ parameter for predictor 1

Simple Regression – Numeric Statistical Model

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \varepsilon_i$$

Prediction = Linear Predictor + Error/Natural Variation

Data			Design Matrix Parameters		Model Variables	
Observation	Response	Predictors			Prediction	Error
i	Y _i	Continuous	X _{0i}	X _{1i}	\hat{Y}_i	ε_i
1	4	4	1	4	4.6	-0.6
2	4	8	1	8	4.7	-0.7
3	6	1	1	1	5.1	0.9
4	3	9	1	9	2.1	0.9
5	2	1	1	1	2.9	-0.9
6	2	7	1	7	2.5	-0.5

Take Home

1. We only indirectly model the data. What we actually model is the design matrix, this is usually created in the background by the software.
2. Examples of parameters not in the data but are in the design matrix include:
 1. Intercept
 2. Quadratic terms (to fit a parabola curve)

Simple Regression – Numeric Statistical Model

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \varepsilon_i$$

Prediction = Linear Predictor + Error/Natural Variability



Data			Design Matrix Parameters		Model Variables	
Observation	Response	Predictors			Prediction	Error
i	Y _i	Continuous	X _{0i}	X _{1i}	\hat{Y}_i	ε_i
1	4	4	1	4	4.6	-0.6
2	4	8	1	8	4.7	-0.7
3	6	1	1	1	5.1	0.9
4	3	9	1	9	2.1	0.9
5	2	1	1	1	2.9	-0.9
6	2	7	1	7	2.5	-0.5

Take Home

1. We only indirectly model the data. What we actually model is the design matrix, this is usually created in the background by the software.
2. Examples of parameters not in the data but are in the design matrix include:
 1. Intercept
 2. Quadratic terms (to fit a parabola curve)

Let's add another continuous predictor variable

Yellow represents the changes required for this to happen

Multiple Regression

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \varepsilon_i$$

Prediction = Linear Predictor + Error/Natural Variation

Notice the predictions have changed and the errors are **overall** smaller (although some are individually larger) As expected when we add new parameters.

Data				Design Matrix Parameters			Model Variables	
Observation i	Response Y _i	Predictors		X _{0i}	X _{1i}	X _{2i}	Prediction Ŷ _i	Error ε _i
		Continuous	Continuous					
		X _{1i}	X _{2i}					
1	4	4	12	1	4	12	4.4	-0.4
2	4	8	54	1	8	54	4.5	-0.5
3	6	1	87	1	1	87	5.3	0.7
4	3	9	96	1	9	96	3.2	-0.2
5	2	1	41	1	1	41	1.8	0.2
6	2	7	47	1	7	47	2.6	-0.6

Data (the actual data you collect)

Y_i ~ **Response** of Observation i

X_{1i} ~ **Predictor X₁** of Observation i

X_{2i} ~ **Predictor X₂** of Observation i

Design Matrix Parameters (the parameters in your model i.e. the actual data you model)

X_{0i} ~ design parameter for parameter β₀ (Constant/Y intercept)

X_{1i} ~ design parameter for β₁ (parameter X_{1i})

X_{2i} ~ design parameter for β₂ (parameter X_{2i})

Model Variables (variables the model calculates)

Ŷ_i ~ **Prediction** for Observation i

ε_i ~ **Error of** Observation i

β₀ ~ Constant/Y intercept parameter

β_{1i} ~ parameter for predictor 1

β_{2i} ~ parameter for predictor 2

Multiple Regression

A new design matrix predictor is simply added for any new continuous predictors you want.

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \varepsilon_i$$

Prediction = Linear Predictor + Error/Natural Variation

Just keep going!!

Data					Design Matrix Parameters				Model Variables	
Obs i	Predictors								Prediction \hat{Y}_i	Error ε_i
	Response Y_i	Continuous X_{1i}	Continuous X_{2i}	Continuous X_{3i}	X_{0i}	X_{1i}	X_{2i}	X_{3i}		
1	4	4	12	12	1	4	12	12	4.2	-0.2
2	4	8	54	54	1	8	54	54	4.3	-0.3
3	6	1	87	87	1	1	87	87	5.3	0.7
4	3	9	96	96	1	9	96	96	2.9	0.1
5	2	1	41	41	1	1	41	41	1.8	0.2
6	2	7	47	47	1	7	47	47	2.4	-0.4

Data (the actual data you collect)

$Y_i \sim$ Response of Observation i

$X_{1i} \sim$ Predictor X_1 of Observation i

$X_{2i} \sim$ Predictor X_2 of Observation i

$X_{3i} \sim$ Predictor X_3 of Observation i

Design Matrix Parameters (the parameters in your model i.e. the actual data you model)

$X_{0i} \sim$ design parameter for parameter β_0 (Constant/Y intercept)

$X_{1i} \sim$ design parameter for β_1 (parameter X_{1i})

$X_{2i} \sim$ design parameter for β_2 (parameter X_{2i})

$X_{3i} \sim$ design parameter for β_3 (parameter X_{3i})

Model Variables (variables the model calculates)

$\hat{Y}_i \sim$ Prediction for Observation i

$\varepsilon_i \sim$ Error of Observation i

$\beta_0 \sim$ Constant/Y intercept parameter

$\beta_{1i} \sim$ parameter for predictor 1

$\beta_{2i} \sim$ parameter for predictor 2

$\beta_{3i} \sim$ parameter for predictor 3

Multiple Regression

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \epsilon_i$$



Prediction = Linear Predictor + Error/Natural Variation

Data					Design Matrix Parameters				Model Variables	
Obs i	Predictors								Prediction \hat{Y}_i	Error ϵ_i
	Response Y_i	Continuous X_{1i}	Continuous X_{2i}	Continuous X_{3i}	X_{0i}	X_{1i}	X_{2i}	X_{3i}		
1	4	4	12	12	1	4	12	12	4.2	-0.2
2	4	8	54	54	1	8	54	54	4.3	-0.3
3	6	1	87	87	1	1	87	87	5.3	0.7
4	3	9	96	96	1	9	96	96	2.9	0.1
5	2	1	41	41	1	1	41	41	1.8	0.2
6	2	7	47	47	1	7	47	47	2.4	-0.4

Data (the actual data you collect)

$Y_i \sim$ Response of Observation i

$X_{1i} \sim$ Predictor X_1 of Observation i

$X_{2i} \sim$ Predictor X_2 of Observation i

$X_{3i} \sim$ Predictor X_3 of Observation i

Design Matrix Parameters (the parameters in your model i.e. the actual data you model)

$X_{0i} \sim$ design parameter for parameter β_0 (Constant/Y intercept)

$X_{1i} \sim$ design parameter for β_1 (parameter X_{1i})

$X_{2i} \sim$ design parameter for β_2 (parameter X_{2i})

$X_{3i} \sim$ design parameter for β_3 (parameter X_{3i})

Model Variables (variables the model calculates)

$\hat{Y}_i \sim$ Prediction for Observation i

$\epsilon_i \sim$ Error of Observation i

$\beta_0 \sim$ Constant/Y intercept parameter

$\beta_{1i} \sim$ parameter for predictor 1

$\beta_{2i} \sim$ parameter for predictor 2

$\beta_{3i} \sim$ parameter for predictor 3

So how do we add Categorical variables??

Yellow represents the changes required for this to happen

Adding Categorical Variables (e.g. ANOVA)

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \varepsilon_i$$

Prediction = Linear Predictor + Error/Natural Variation

Data				Design Matrix Parameters				Model Variables	
Obs i	Response Y _i	Predictors Continuous X _{1i}	Categorical X _{2i}		X _{0i}	X _{1i}	X _{2i}	Prediction \hat{Y}_i	Error ε_i
1	4	4	Non Smoking		1	4	0	4.6	-0.6
2	4	8	Smoking		1	8	1	4.2	-0.2
3	6	1	Non Smoking		1	1	0	5.1	0.9
4	3	9	Smoking		1	9	1	3.4	-0.4
5	2	1	Non Smoking		1	1	0	1.4	0.6
6	2	7	Non Smoking		1	7	0	2.2	-0.2

Data (the actual data you collect)

$Y_i \sim$ Response of Observation i

$X_{1i} \sim$ Predictor X_1 of Observation i

$X_{2i} \sim$ Predictor X_2 of Observation i

Design Matrix Parameters (the parameters in your model i.e. the actual data you model)

$X_{0i} \sim$ design parameter for parameter β_0 (Reference group = Non-Smoking)

$X_{1i} \sim$ design parameter for β_1 (parameter X_{1i})

$X_{2i} \sim$ design parameter for β_2 (parameter X_{2i} = smoking)

Model Variables (variables the model calculates)

$\hat{Y}_i \sim$ Prediction for Observation i

$\beta_0 \sim$ (Reference group = Non-Smoking)

$\beta_{2i} \sim$ parameter for smoking

$\varepsilon_i \sim$ Error of Observation i

$\beta_{1i} \sim$ parameter for predictor 1

Adding Categorical Variables (e.g. ANOVA)

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \varepsilon_i$$

Prediction = Linear Predictor + Error/Natural Variation

Data				Design Matrix Parameters				Model Variables	
Obs	Response	Predictors						Prediction	Error
i	Y _i	Continuous	Categorical					\hat{Y}_i	ε_i
		X _{1i}	X _{2i}		X _{0i}	X _{1i}	X _{2i}		
1	4	4	Non Smoking		1	4	0	4.6	-0.6
2	4	8	Smoking		1	8	1	4.2	-0.2
3	6	1	Non Smoking		1	1	0	5.1	0.9
4	3	9	Smoking		1	9	1	3.4	-0.4
5	2	1	Non Smoking		1	1	0	1.4	0.6
6	2	7	Non Smoking		1	7	0	2.2	-0.2

There are many different **parameterisations** (ways) to add categorical variables. The way I am showing you is called **Dummy** or **Treatment Coding**. Linear Models 3 discusses other ways such as effects coding.

Dummy coding works by picking 1 category as the **reference category**, this category is captured in the **constant/intercept parameter** and is always 'on'. We then adjust it when a different category is present by adding their specific parameter into the prediction equation/model.

This means that every other category other than the reference category has it's own design parameter which functions as an 'indicator variable' since:

- When $X_2 = 1$ it “turns on” β_2 since $\beta_2 X_{2i} = \beta_2 * 1 = \beta_2$
 - β_2 only comes into the model when $X_2 = 1$, i.e. when people smoke i.e. it is the extra effect of smoking compared to the baseline reference level of not smoking.
- When $X_2 = 0$ it “turns off” β_2 since $\beta_2 X_{2i} = \beta_2 * 0 = 0$
 - We only have β_0 when people don't smoke i.e. $X_2 = 0$, i.e. it is the baseline prediction when people don't smoke i.e. it's the reference level.

Adding Categorical Variables (e.g. ANOVA)

A new design matrix predictor is simply added for any new categorical levels you want.

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \varepsilon_i$$

Prediction = Linear Predictor + Error/Natural Variation

Just keep going!!

Data				Design Matrix Parameters				Model Variables	
Obs i	Response	Predictors		X0i	X1i	X2i	X3i	Prediction \hat{Y}_i	Error ε_i
	Yi	Continuous X1i	Categorical X2i						
1	4	4	Never Smoked	1	4	0	0	4.6	-0.6
2	4	8	Smoking	1	8	1	0	4.2	-0.2
3	6	1	Ex smoker	1	1	0	1	5.1	0.9
4	3	9	Smoking	1	9	1	0	3.4	-0.4
5	2	1	Never Smoked	1	1	0	0	1.4	0.6
6	2	7	Never Smoked	1	7	0	0	2.2	-0.2

Data (the actual data you collect)

$Y_i \sim$ Response of Observation i

$X_{1i} \sim$ Predictor X_1 of Observation i

$X_{2i} \sim$ Predictor X_2 of Observation i

Design Matrix Parameters (the parameters in your model i.e. the actual data you model)

$X_{0i} \sim$ design parameter for parameter β_0 (Reference group = Never Smoked)

$X_{1i} \sim$ design parameter for β_1 (parameter X_{1i})

$X_{2i} \sim$ design parameter for β_2 (parameter X_{2i} = Smoking)

$X_{3i} \sim$ design parameter for β_3 (parameter X_{3i} = Ex Smoker)

Model Variables (variables the model calculates)

$\hat{Y}_i \sim$ Prediction for Observation i

$\beta_0 \sim$ (Reference group = Non-Smoking)

$\beta_{2i} \sim$ parameter for smoking

$\varepsilon_i \sim$ Error of Observation i

$\beta_{1i} \sim$ parameter for predictor 1

$\beta_{3i} \sim$ parameter for Ex Smoker

What's the Difference with Multiple Regression?

Virtually none. The underlying model is exactly the same!! The only changes are in the data:

1. The X predictor is continuous when adding a continuous variable aka multiple regression, while it's an indicator variable if adding a discrete variable.
2. Interpretation of the parameters differs
3. But they are both still *linear models*

Take Home
Categorical ANOVA style models are the same as continuous style regression models. The only difference is in the design matrix.

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \varepsilon_i$$

Prediction = Linear Predictor + Error/Natural Variation

Data					Design Matrix Parameters				Model Variables	
Obs i	Response	Predictors	Continuous X2i	Continuous X3i	X0i	X1i	X2i	X3i	Prediction \hat{Y}_i	Error ε_i
	Yi	Continuous X1i								
1	4	4	12	12	1	4	12	12	4.4	-0.4
2	4	8	54	54	1	8	54	54	4.5	-0.5
3	6	1	87	87	1	1	87	87	5.3	0.7
4	3	9	96	96	1	9	96	96	3.2	-0.2
5	2	1	41	41	1	1	41	41	1.8	0.2
6	2	7	47	47	1	7	47	47	2.6	-0.6

Data					Design Matrix Parameters				Model Variables	
Obs i	Response	Predictors	Categorical X2i		X0i	X1i	X2i	X3i	Prediction \hat{Y}_i	Error ε_i
	Yi	Continuous X1i								
1	4	4	Non Smoking		1	4	0	0	4.5	-0.5
2	4	8	Smoking		1	8	1	0	4.1	-0.1
3	6	1	Ex smoker		1	1	0	1	4.9	1.1
4	3	9	Smoking		1	9	1	0	3.4	-0.4
5	2	1	Non Smoking		1	1	0	0	1.2	0.8
6	2	7	Non Smoking		1	7	0	0	1.8	0.2

What's the Difference with Multiple Regression

Virtually none. The underlying model is exactly the same!! The only changes are in the design matrix

1. The X predictor is continuous when adding a continuous variable aka multiple regression, it's an indicator variable if adding a discrete variable.
2. Interpretation of the parameters differs
3. But they are both still *linear models*



Take Home
Categorical ANOVA style models are the same as continuous style regression models. The only difference is in the design matrix

$$Y_i = \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \epsilon_i$$

Prediction = Linear Predictor + Error/Natural Variation

Data					Design Matrix Parameters				Model Variables	
Obs i	Response	Predictors	Continuous X2i	Continuous X3i	X0i	X1i	X2i	X3i	Prediction \hat{Y}_i	Error ϵ_i
	Yi	Continuous X1i								
1	4	4	12	12	1	4	12	12	4.4	-0.4
2	4	8	54	54	1	8	54	54	4.5	-0.5
3	6	1	87	87	1	1	87	87	5.3	0.7
4	3	9	96	96	1	9	96	96	3.2	-0.2
5	2	1	41	41	1	1	41	41	1.8	0.2
6	2	7	47	47	1	7	47	47	2.6	-0.6

Data					Design Matrix Parameters				Model Variables	
Obs i	Response	Predictors	Categorical X2i		X0i	X1i	X2i	X3i	Prediction \hat{Y}_i	Error ϵ_i
	Yi	Continuous X1i								
1	4	4	Non Smoking		1	4	0	0	4.5	-0.5
2	4	8	Smoking		1	8	1	0	4.1	-0.1
3	6	1	Ex smoker		1	1	0	1	4.9	1.1
4	3	9	Smoking		1	9	1	0	3.4	-0.4
5	2	1	Non Smoking		1	1	0	0	1.2	0.8
6	2	7	Non Smoking		1	7	0	0	1.8	0.2

Using the design matrix to represent complex models in a more simple, concise and generalisable way

$$\begin{aligned} Y_i &= \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \dots + \varepsilon_i \\ &= \mathbf{X}\boldsymbol{\beta} + \varepsilon_i \sim \text{a shorter and simpler way to write any linear model} \\ &= \text{linear/additive model} \end{aligned}$$

(As opposed to a multiplicate model which is

$$Y_i = \beta_0 X_{0i} \times \beta_1 X_{1i} \times \beta_2 X_{2i} \times \beta_3 X_{3i} \dots + \varepsilon_i)$$

Using the design matrix to represent complex models in a more simple, concise and generalisable way

$$\begin{aligned} Y_i &= \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \dots + \varepsilon_i \\ &= \mathbf{X}\boldsymbol{\beta} + \varepsilon_i \sim \text{a shorter and simpler way to write any linear model} \\ &= \text{linear/additive model} \end{aligned}$$

\mathbf{X} = design matrix
~ the actual data modelled

Design Matrix Parameters

X0i	X1i	X2i
1	4	12
1	8	54
1	1	87
1	9	96
1	1	41
1	7	47

$\boldsymbol{\beta}$ = vector of
parameters

$\boldsymbol{\beta}$
β_0
β_1
β_2

ε = vector of
Errors

Error ε_i
-0.4
-0.5
0.7
-0.2
0.2
-0.6

The linear predictor is even more simple, concise and generalisable than the design matrix!!

$$\begin{aligned}\hat{Y}_i &= \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} \\ &= \mathbf{X}\boldsymbol{\beta}\end{aligned}$$

= linear/additive model = η_i = **linear predictor**.

The Linear Predictor
Part 1 of the 3
required for a GLM

Notice that we removed the error, which means now we have a prediction, which is what the hat over the y means.

\mathbf{X} = design matrix

~ the actual data modelled

$\boldsymbol{\beta}$ = vector of
parameters

Design Matrix Parameters

X0i	X1i	X2i
1	4	12
1	8	54
1	1	87
1	9	96
1	1	41
1	7	47

$\boldsymbol{\beta}$

β_0

β_1

β_2

Adding Transformations

So we have established that $Y_i = \eta_i + \varepsilon_i$ can be used to efficiently represent all types of linear models.

But we often want to transform the response e.g. a very common transformation is to take its log so we now have

$$\text{Log}(Y_i) = \eta_i + \varepsilon_i$$

This is called the **link function** in a GLM

The Link Function
Part 2 of the 3
required for a GLM

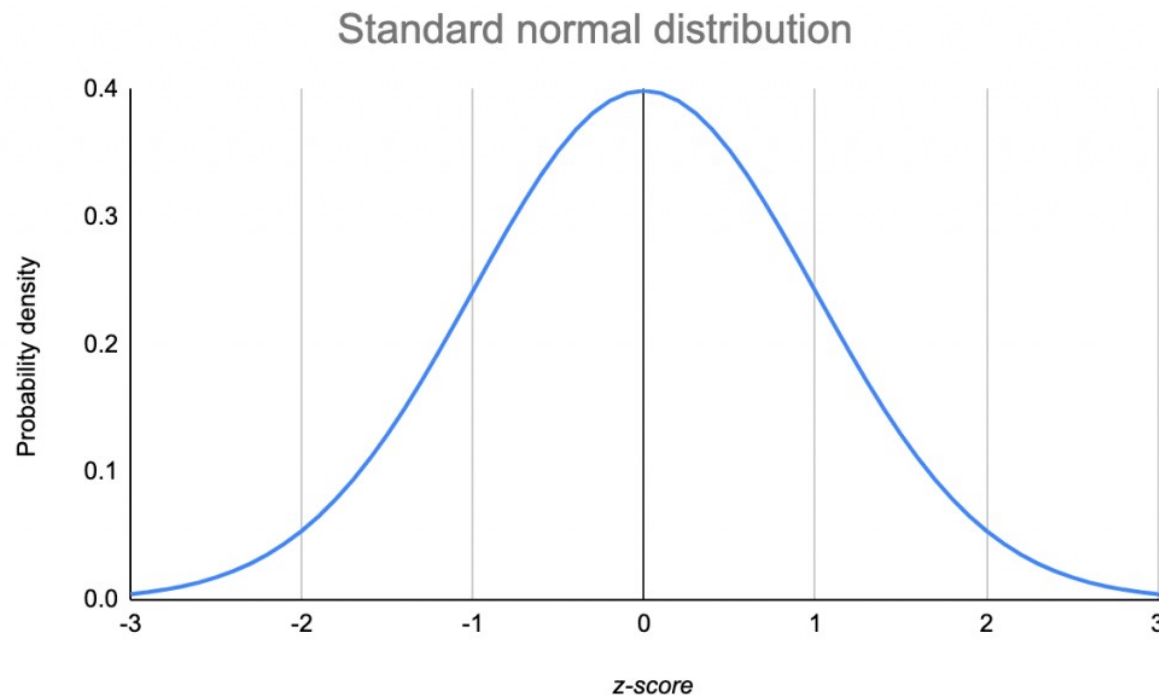
(A more formal way to represent it is $E(Y | X) = \mu = g^{-1}(\eta)$ where g is the link function.)

TAKE HOME

Link function allows us to effectively transform the response

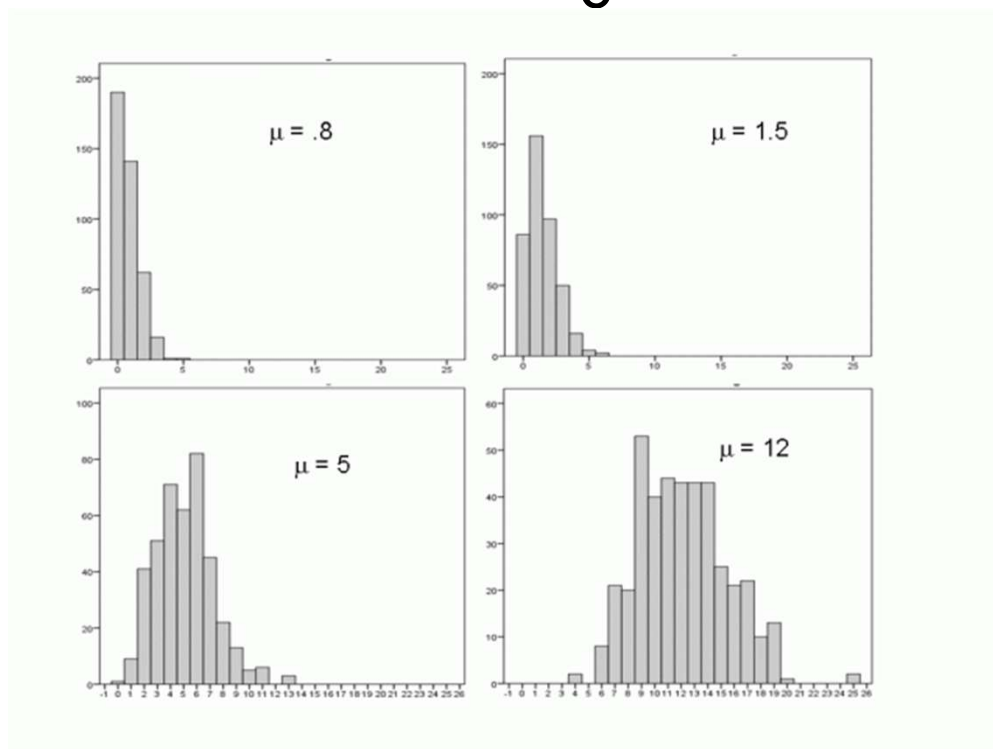
So far we have assumed a Normal distribution

- 2 parameters describes the curve
- Response is continuous
 - Ranges from $-\infty$ to $+\infty$
- Mean = μ
- Variance = σ^2



BUT, what if it was different, say a Poisson Distribution

- 1 parameter describes the curve
- Response is discrete
 - Often used for counts
 - Ranges from 0 to + infinity
- Mean = variance = λ (lambda)
- Variance gets bigger as mean does. Which makes sense since larger counts can have larger variance



Different Data Distributions

The Data Distribution
Part 3 of the 3
required for a GLM

Common Distributions

Normal Distribution for unbounded continuous data

Poisson for count data

Binomial for binary data i.e. logistic regression

GLM components

$$\begin{aligned}\hat{Y}_i &= \beta_0 X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} \\ &= \mathbf{X}\beta \\ &= \text{linear/additive model} = \eta_i = \text{linear predictor.}\end{aligned}$$

Part 1 of the 3 required for a GLM

The **Linear Predictor** is the equation of predictors and parameters that predicts the response

The parameters are defined by the **Design Matrix**.

Part 2 of the 3 required for a GLM

The **Link Function** is a way to add transformations

Part 3 of the 3 required for a GLM

Different Data Distributions

So let's take a breath and tie all that together into a concise definition of a Linear Model

Simple Linear Model – Tying it all together

$$Y_i = X\beta + \varepsilon_i$$

= Deterministic model + Random model

Can be used to fit any of these models

- 1 continuous predictor (simple linear regression)
- Multiple continuous predictors (multiple regression)
- Discrete predictors (ANOVA, RCT, BACI, Control/Treatment)
- Discrete + continuous predictors e.g. ANCOVA is a special case

The only difference between them is the Design Matrix X from $X\beta$!

Observation <i>i</i>	Response <i>Y</i>	Design Matrix		Prediction	Error
		Continuous <i>X1</i>	Categorical <i>X2</i>		
1	4.0	4	0	4.6	0.6
2	4.0	8	0	4.7	0.7
3	6.0	1	0	5.1	-0.9
3	3.0	9	1	2.1	-0.9
4	2.0	1	0	2.9	0.9
5	2.0	7	1	2.5	0.5

Simple Regression – some new notation that succinctly and simply explains these concepts

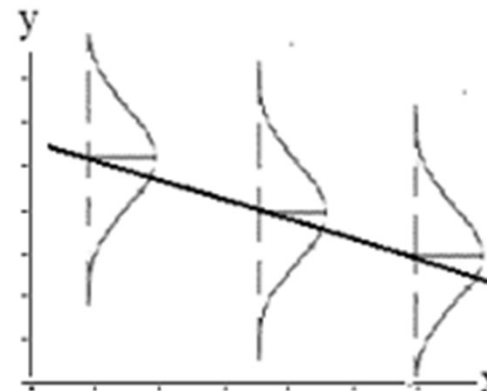
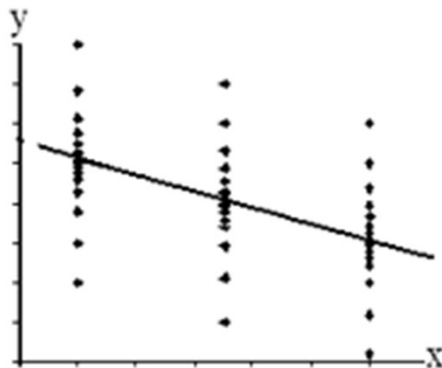
$$Y_i = \beta_0 + \beta_1 X_{1i} + \varepsilon_i$$

$Y_i \sim$ Is normally distributed with an

- average conditional on the predictors i.e. $\mu = \beta_0 + \beta_1 X_{1i}$
- and variance σ^2

Or more concisely

$$Y_i \sim N(\beta_0 + \beta_1 X_{1i}, \sigma^2) \text{ since } N(\mu, \sigma^2)$$



Simple Linear Model – Tying it all together

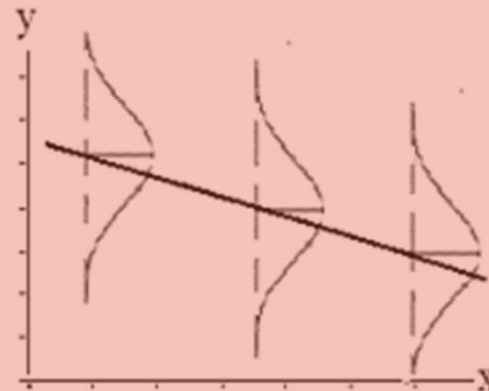
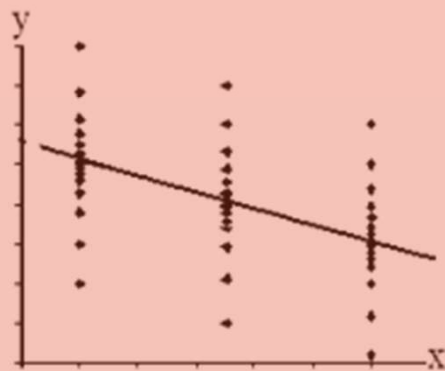
$$Y_i = \mathbf{X}\beta + \varepsilon_i$$

= Deterministic model + Random model

$Y_i \sim$ Is normally distributed with an:

- average conditional on the predictors i.e. $\mu = \beta_0 + \beta_1 X_{1i}$
- and variance σ^2

$Y_i \sim N(\mathbf{X}\beta, \sigma^2)$ since $N(\mu, \sigma^2)$ i.e. the conditional expectation = $\mu = \mathbf{X}\beta$



Challenge Q: What do we change if the Error isn't normal??



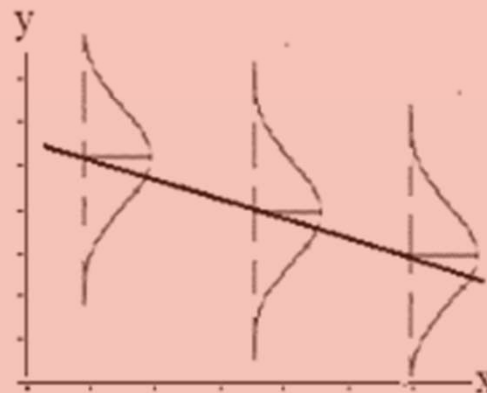
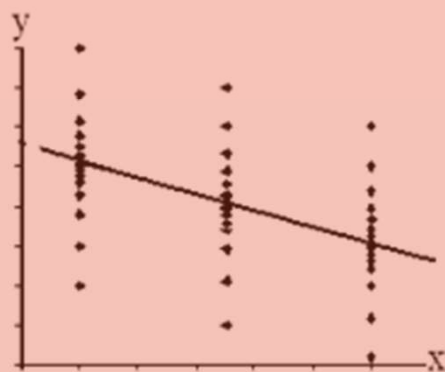
$$Y_i = \mathbf{X}\beta + \varepsilon_i$$

= Deterministic model + Random model

$Y_i \sim$ Is normally distributed with an:

- average conditional on the predictors i.e. $\mu = \beta_0 + \beta_1 X_{1i}$
- and variance σ^2

$Y_i \sim N(\mathbf{X}\beta, \sigma^2)$ since $N(\mu, \sigma^2)$ i.e. the conditional expectation = $\mu = \mathbf{X}\beta$



Challenge Q: What do we change if the Error isn't normal??



$$Y_i = \mathbf{X}\beta + \varepsilon_i$$

= Deterministic model + Random model

$Y_i \sim$ Is **Poisson** distributed with an:

- average conditional on the predictors i.e. $\mu = \beta_0 + \beta_1 X_{1i}$

If it was Poisson we might try changing the distribution from normal to Poisson

$Y_i \sim \text{Poisson}(\mathbf{X}\beta)$ since the conditional expectation = $\mu = \lambda = \mathbf{X}\beta$

Challenge Q: What do we change if the Error isn't normal??



BUT we have a problem. In Poisson the conditional expectation μ doesn't usually equal $\mathbf{X}\beta$. Instead $\log(\mu)$ usually does. So instead we add a link function. That links the linear predictor to the conditional expectation.

If it was Poisson we might try changing the distribution from normal to Poisson

$Y_i \sim \text{Poisson}(\mu)$ since the conditional expectation μ is $\log(\mu) = \mathbf{X}\beta = \eta$
~ i.e. log is the link function between the linear predictor ($\mathbf{X}\beta = \eta$) and the conditional expectation μ

**Congratulations. You just developed Generalised
Linear Models from 1st principles!**

Let's tie everything we just saw together into a single summary you can come back to

Simple Linear Model

SIMPLE LINEAR MODEL

$$Y_i = X\beta + \varepsilon_i$$

= Deterministic model ($X\beta$) + Random model (ε_i)

$\sim N(\mu, \sigma^2)$ where $\mu = X\beta$ i.e. assumes a Normal error

~ Gives us a simple, single, unified way of fitting all types of continuous and discrete predictors so we can fit different models like regression, ANOVA, ANCOVA, BACI, RCT, Control/Treatment, etc. It does this by using a **design matrix X** with different design variables.

Observation	Response	Design Matrix		Prediction	Error
		Continuous	Categorical		
i	Y	X1	X2		
1	4.0	4	0	4.6	0.6
2	4.0	8	0	4.7	0.7
3	6.0	1	0	5.1	-0.9
3	3.0	9	1	2.1	-0.9
4	2.0	1	0	2.9	0.9
5	2.0	7	1	2.5	0.5

GLM's are a simple extension of Simple Linear models

SIMPLE LINEAR MODEL

$$Y_i = X\beta + \varepsilon_i$$

= Deterministic model ($X\beta$) + Random model (ε_i)

~ $N(\mu, \sigma^2)$ where $\mu = X\beta$ i.e. assumes a Normal error

~ Gives us a simple, single, unified way of fitting all types of continuous and discrete predictors so we can fit different models like regression, ANOVA, ANCOVA, BACI, RCT, Control/Treatment, etc. It does this by using a **design matrix X** with different design variables.

~ (also known as General Linear Models – as opposed to Generalised Linear Models)

GENERALISED LINEAR MODEL (GLM)

- Can fit *all the same models* as a Simple Linear Model PLUS it:
 - **Generalises** the model so we can use non normal errors
 - Adds inbuilt response transformations via the link function

GLM's are a simple extension of Simple Linear models with 3 parts

1. $Y_i \sim N(\mu, \sigma)$ or $\text{Poisson}(\mu)$ or $\text{Binomial}(\mu)$ or etc
 - **Random model:** which is the **distribution** of the data conditional on the expectation e.g. my response is normally distributed with average μ .
2. $X\beta = \eta$
 - **Deterministic model:** which is the **linear predictor** that relates the predictors to the response.
 - Notice the **Design Matrix X**, is the same as in the simple linear model. This lets us fit all the models we are used e.g. ANOVA, BACI, RCT, etc, **but with a different error**.
3. $\mu = g(\eta) = g(X\beta)$
 - The **link function (g)** which links the linear predictor $\eta = X\beta$ with the response via it's distribution and conditional expectation.

Here are the 3 most common GLM's

Simple Linear Models

$$Y_i \sim N(\mu, \sigma^2) \text{ where } \mu = X\beta$$

Poisson (count) Model

$$Y_i \sim \text{Poisson}(\mu) \text{ where } \log(\mu) = X\beta$$

Logistic (binary) Model

$$Y_i \sim \text{Binomial}(\mu) \text{ where } \text{logit}(\mu) = \ln \frac{p}{1-p} = X\beta \text{ (since the probability, } p, \text{ is just the mean of the } Y \text{ values, assuming 0,1 coding, which is often expressed as } \mu)$$



Logistic Regression example 1

Binary Response e.g. yes/no, success/failure, 0/1

Workflow Suitable for:

- Continuous predictor



THE UNIVERSITY OF
SYDNEY

Logistic/Binary Regression

Used when we have a categorical response than can be 1 of 2 categories. We usually code them as:

1 = Success

0 = Failure

Tells us which predictors are positively and negatively correlated with more Successes. To make the output easy to understand the trick is defining the 'success' group.

Medical: We usually define the disease as the 'success' since we want to know what influences getting it.

Churn: Could be either the people who left or stayed, depending on who we want to focus on.

Loan Defaults: Defaulters would usually be the 'success' group since we want to know why people default.

Similar to Survival Analysis

When deciding which to use consider the data available and Research Question:

Logistic Regression models the probability (chance) of an event occurring

Survival Analysis models the probability (chance) of an event occurring *and the time to that event*

The main differences are that Survival Analysis:

1. Factors in time to the Event/Success and gives you survival curves. There is an important distinction between living for 6 months vs 6 years after diagnosis! Logistic treats them the same (unless time to death is explicitly added).
2. Can handle data where the event happens for everyone i.e. everyone dies.
3. Factors in patients lost to follow up (censoring)
4. Tends to report the Hazard Ratio, rather than the Odds Ratio.
5. Naturally handles time varying covariates (since it naturally includes time to event while logistic regression does not).
 1. Logistic regression factors in time as an additional predictor. A categorical predictor gives us different parameters/logit curves e.g. event occurred at 6 months vs 6 years, or continuous e.g. covariate adjustment parameter of Beta. Covariates that then vary by time can be added as interactions to the time predictor.

Refer to our Survival Analysis workshop for more information.

Model Fitting Workflow

Step 0) Clean and check data.

Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

Step 2) Fit the Model

Step 3) Check Model Assumptions via Diagnostics: Residual Analysis

Step 4) Goodness of Fit: Plots and Statistics

Step 5) Interpret Model Parameters and reach a conclusion

Step 6) Reporting

Linear Models 3 and Model Building Workshops have more detail on many of these steps.

Step 0) Clean and check data

- Is covered in “Research Essentials”, not this workshop.
- Is very important, so ensure you do it!
- Get in the habit of checking the data every time you open it by looking at the **corners** i.e. start at the top left corner, then scroll to the far right corner, scroll down to the bottom right corner, scroll left to the bottom left corner, then finish by scrolling back up to the beginning top left corner.
 - Weird things can happen. New versions, a stray cosmic ray. I have literally opened data to find it corrupted, and then reopened it and it's fine. Similarly I have seen weird results only to rerun them to find them OK.

Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

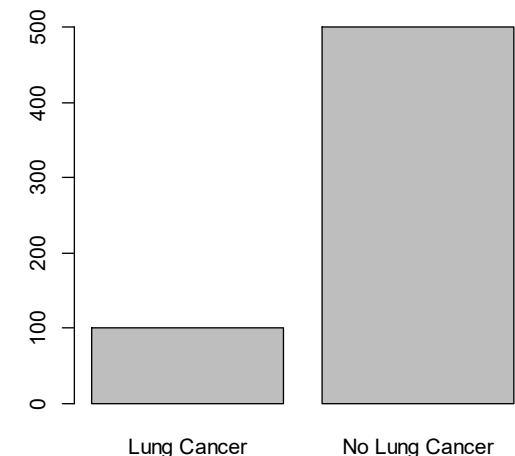


Challenge Question: We have done a case control study. We got 100 people with lung cancer and 500 people without. How would you plot the response variable?

Our response has 2 options. There are no outliers or NA's.

So it's not appropriate for a Simple Linear Regression with a Normal error. No way the error will be normal with only 2 responses.

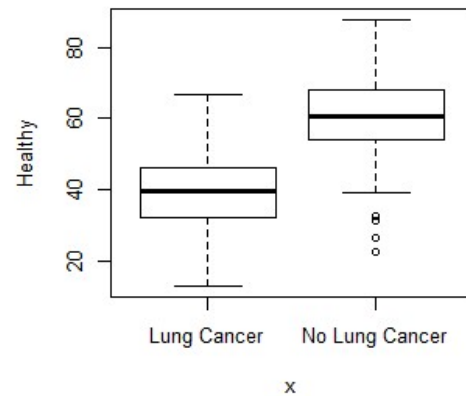
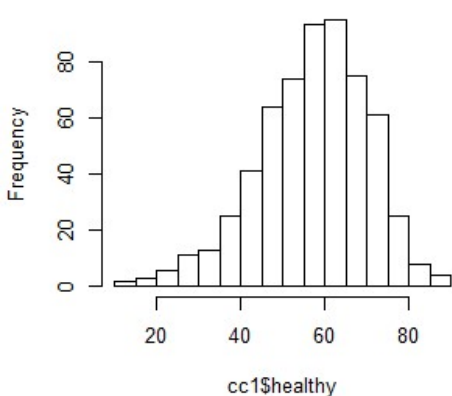
BUT it's a good contender for Logistic/Binary Regression.



```
> plot(cc1$"lung cancer")
```

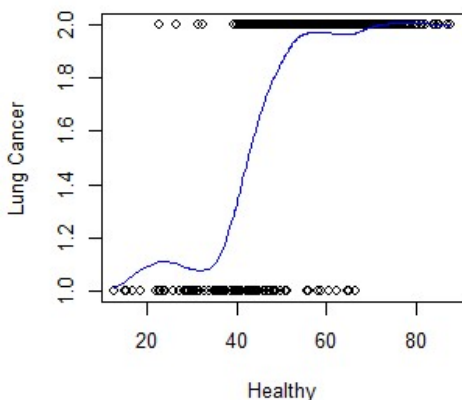
Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

Now add the continuous predictor “healthy lifestyle” which is an index based on things like exercise, food, sleep, etc. It ranges from 0 = unhealthy to 100 = healthy. How might it be related to lung cancer?



All 3 plots tells us there are no outliers or other data problems with “Healthy”.

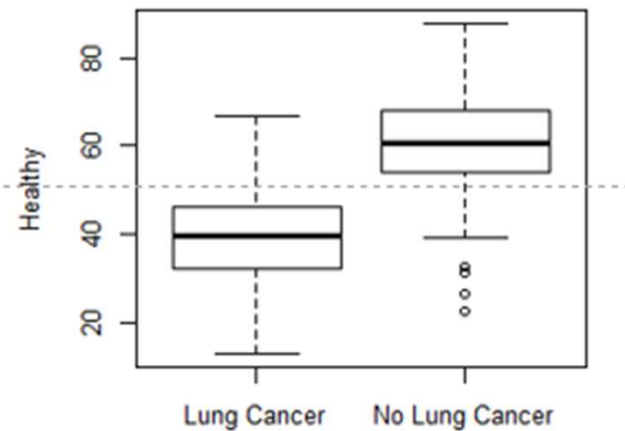
The boxplot and scatterplot show us there is a relationship between healthy and lung cancer.



```
> windows()
> par(mfrow=c(2,2))
> hist(cc1$healthy, main="")
> plot(cc1$`lung cancer`, cc1$healthy, ylab="Healthy")
> plot(cc1$healthy, as.numeric(cc1$`lung cancer`), ylab="Lung Cancer", xlab="Healthy")
> lines(smooth.spline(cc1$healthy, as.numeric(cc1$`lung cancer`))), col="blue", ylab="Lung Cancer",
xlab="Healthy")
```

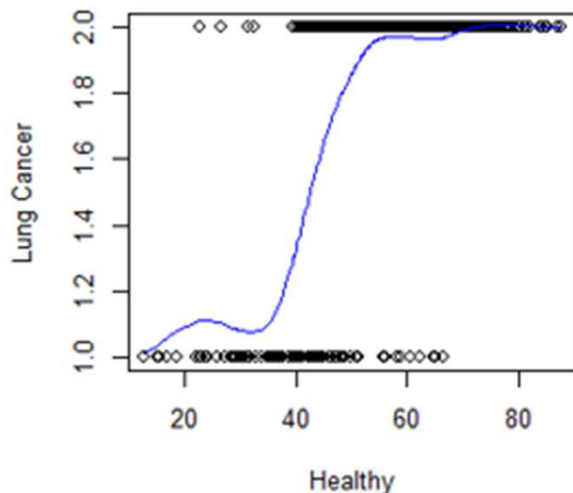
In vertical axis: 1 = Lung Cancer, 2 = No Lung Cancer. Had to convert to numbers and not label with the text in order to get the smoothed blue line.

Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).



This plot shows the Health Indices **average difference** between those with and without lung cancer. It is quantified using an ANOVA like we did in LM1.

It allows us to predict the health index score knowing if someone has lung cancer.



This plot shows the relationship between the health index and getting lung cancer. It is quantified using logistic regression.

It allows us to predict the chance of having lung cancer if we know their health score.

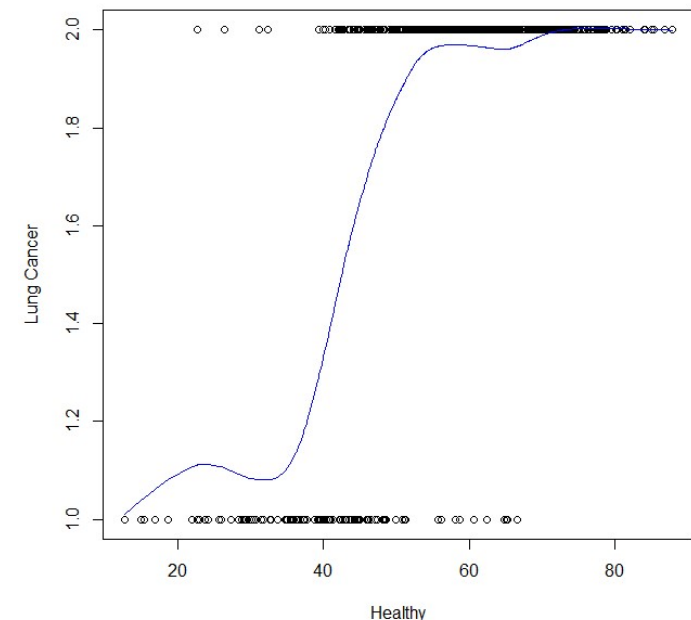
Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

Notice that the relationship between healthy and Lung Cancer **isn't linear**. It's more of an S shape.

This relationship is called a ***sigmoid*** function, and is what logistic regression fits.

But how do we fit this using a linear model?

The trick is the link function in a GLM.
Which lets us fit non linear models.



Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

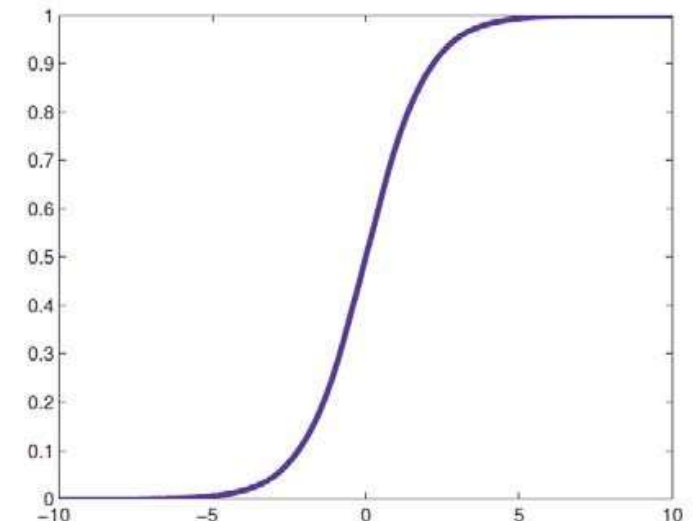
Logistic GLM might be a good fit, so lets try that

$Y_i \sim \text{Binomial}(\mu)$ where $\text{logit}(\mu) = \ln \frac{p}{1-p} = X\beta$ (since the probability of having lung cancer, p , is just the mean of the Y values, assuming 0,1 coding, which is often expressed as μ)

The **logit link** function lets us fit this sigmoid function.

(And makes it multiplicative model when we back transform to Odd's Ratio's).

SIGMOID FUNCTION



Step 1) If we had categorical variables such as smoking

We also need to look for **Separation**.

Complete Separation occurs when we have cells that are entirely success or failures e.g. if we had included smoking perhaps all the smokers got lung cancer. This is an example of where smoking has *separated* the response. The model can not fit when this happens and is one common reason for logistic models not converging (since its effectively trying to divide by 0).

Separation often causes error messages like “failed to converge” or high parameter SE’s.

Even if we don’t have complete separation, marginal separation can still cause problems.

	Lung Cancer	No Lung Cancer
Smoker	100	0
Non Smoker	10	800

	Estimate	SE
Constant	7.9	0.06
Smoker	1000	597000

Step 2) Fit the Model

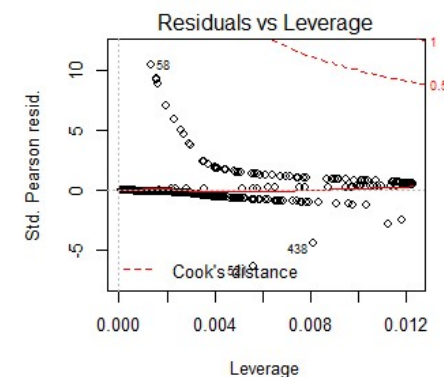
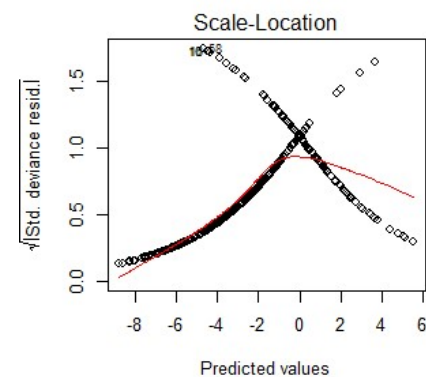
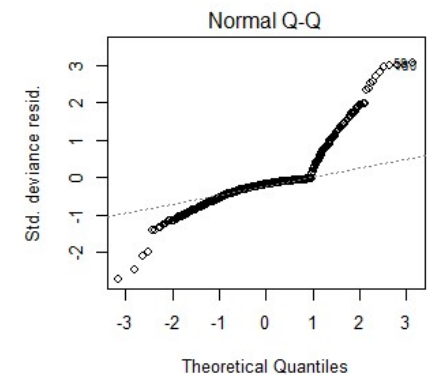
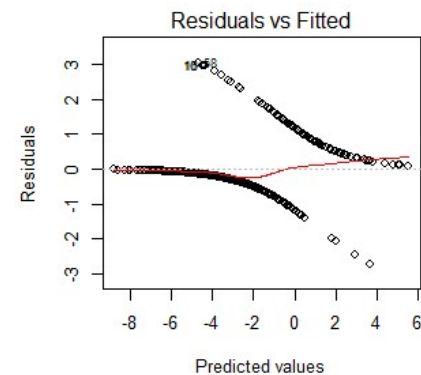
```
cc.model <- glm(lung.cancer ~ healthy, data=cc2,  
family=binomial)
```

“Success” = Having Lung Cancer, meaning the parameters tell us what risk factors there are for getting cancer.

Step 3) Check Model Assumptions via Diagnostics: Residual Analysis

The standard residual plots don't help much here since we don't expect normal residuals and as we only have 2 responses we get these 2 lines in the residual plots.

However they can be used to look for
Outliers.



Step 3) Check Model Assumptions via Diagnostics: Is there any Over Dispersion?

One of the problems we have is that the Binomial Distribution has no separate variance parameter.

The Normal distribution has 2 parameters. The mean (μ) and the variance (σ).

However the Binomial Distribution only has 1 parameter: p ~ the probability of an event occurring. Its average and variance are both functions of this single parameter. But sometimes we have more variance than the distribution can handle.

There are some complications on how we handle this for logistic regression which are beyond the scope of this workshop. However we mention it here so you are aware.

Step 4) Goodness of Fit: Are any parameter SE's too high?

It's always a good idea to look at the parameter SE's to see if any are a lot higher than the others. This can be a sign of a variety of problems. At the very least they suggest the estimate for this parameter is very unstable. The below is for our model and doesn't suggest any problems.

```
# Some of the R output available from  
> summary(cc.model)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	7.98444	0.88840	8.987	<2e-16 ***
healthy	-0.19048	0.01856	-10.265	<2e-16 ***

BUT this might, notice SE is an order of magnitude larger than estimate i.e. times 10/add a zero. Often caused by Separation, which we hopefully identified during the EDA. However marginal separation can be hard to identify, particularly if we are fitting a lot of 2 way interactions with a lot of cells.

Coeffects:

	Estimate	Std. Error
(Intercept)	7.9844	88.840
Healthy	-0.19048	1.856

Step 4) Goodness of Fit: Are any parameter SE's too high?

As previously mentioned (and copied below) a large SE can be a sign of **Separation**.

Complete Separation occurs when we have cells that are entirely success or failures e.g. if we had included smoking perhaps all the smokers got lung cancer. This is an example of where smoking has *separated* the response. The model can not fit when this happens and is one common reason for logistic models not converging (since its effectively trying to divide by 0)

Even if we don't have complete separation, marginal separation can still cause problems such as very high SE's.

Separation often causes error messages like "failed to converge"

	Lung Cancer	No Lung Cancer
Smoker	100	0
Non Smoker	10	800

	Estimate	SE
Constant	7.9	0.06
Smoker	1000	597000

Step 4) Goodness of Fit: Compare it to the NULL model

It's always worth comparing any model to the NULL model, which is the model without any predictors and only a constant/intercept.

In this case we have strong evidence that our model is outperforming the NULL model ($P < 2.2e-16$)

```
> null <- glm(lung.cancer ~ 1, data=cc2, family=binomial)
> anova(null, cc.model, test = "Chisq")
Analysis of Deviance Table

Model 1: lung.cancer ~ 1
Model 2: lung.cancer ~ healthy
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         599      540.67
2         598      292.26  1    248.41 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Step 4) Goodness of Fit: What is it's (Pseudo) R-Squared?

Technically there is no R-Squared for a GLM, however there is an equivalent based on the % Deviance explained. This is one type of Pseudo R-Squared.

Which in this case is acceptable, at 45%

```
> # GOODNESS OF FIT: R-squared equivalent % Deviance explained  
> (deviance.explained <- ((deviance(null)-deviance(cc.model))/deviance(null))*100)  
[1] 45.94528
```


Step 5) Interpret Model Parameters and reach a conclusion

For Simple Linear models we can simply look at the parameter estimate summary and CI's. BUT in logistic regression these are hard to interpret as they are still on the logit scale.

The only really useful part of this 'raw' output is the p-value associated with the parameters. Which in this case shows strong evidence of being associated with healthy ($p < 2e-16$)

```
# Some of the R output available from  
> summary(cc.model)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	7.98444	0.88840	8.987	<2e-16	***
healthy	-0.19048	0.01856	-10.265	<2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
> confint(cc.model) # 95% CI for the coefficients
```

waiting for profiling to be done...

	2.5 %	97.5 %
(Intercept)	6.3469356	9.8414159
healthy	-0.2294994	-0.1564947

Step 5) Interpret Model Parameters and reach a conclusion

The parameters can be made more interpretable by taking the exponential since this turns them into Odds Ratio's.

Taking the exponential is similar to taking something to the power 10. But instead of 10 we use the constant $e = \exp = 2.718$, which is the inverse of the natural logarithm function (\ln) we used in the link function.

Don't overthink it! As an example, as our coefficient is -0.19 if we took it to the power 10 we would get $10^{-0.19048} = 0.65$, but instead we do $2.718^{-0.19048} = e^{-0.19048} = \exp(-0.19048) = 0.83$.

Step 5) Interpret Model Parameters and reach a conclusion

This tells us that for each 1 point increase on the Health index the odds of getting lung cancer are 0.8 compared to the lower score (95%CI = 0.79-0.86).

So being healthy lowers the chance of getting lung cancer!

```
> exp(coef(cc.model)) # exponentiated coefficients
(Intercept)      healthy
2934.9224129      0.8265662
> exp(confint(cc.model)) # 95% CI for exponentiated coefficients
Waiting for profiling to be done...
              2.5 %      97.5 %
(Intercept) 570.7410272 1.879631e+04
healthy      0.7949314  8.551361e-01
```

It's a multiplicative model, not an additive one



Given the odds of getting lung cancer drop by 0.8 for a 1 point increase in health. What impact does a 2 point increase in health Have?

Would it be $0.8 + 0.8 = 1.6$ (additive)?

- Can't be this, since it goes from dropping the odds of lung cancer (<1) to increasing them (>1)!

Or $0.8 * 0.8 = 0.64$ (multiplicative)?

- This makes more sense as a 2 point increase in health leads to a lower chance of lung cancer than a 1 point increase.
- This is what the log link (transformation) does. It turns the additive linear predictor which is an additive model without a log link, into a multiplicative model when it has one.
- So to calculate the Odds Ratio for k intervals of difference in the health predictor it's 0.8^k for this example or β^k in general e.g. if we wanted the odds ratio for a continuous predictor that moved from 5 to 10 it would be β^5 .
 - Notice that this is for any difference in the predictor. The impact is the same if its 5 vs 10 or 100 vs 105, since both are a 5 interval difference.

Step 5) Interpret Model Parameters and reach a conclusion

Parameter	Estimate (raw)	SE (raw)	T score (raw)	P value (raw)	95% Confidence Interval Exp(β) i.e. Odds Ratio		
					Estimate	Lower Bound	Upper Bound
Constant / Control (β_0)	8.0	0.89	9.0	<2e-16			
Health index (β_1)	-0.19	0.019	-10	<2e-16	0.82	0.79	0.86

Step 6) Reporting: Overall Conclusion suitable for publication

“There is strong evidence to show that being healthy is associated with lower odds of Lung Cancer ($p < 2e-16$). For each 1 point increase on the Health index the Odds of getting lung Cancer are 0.8 compared to the lower score (95%CI Odds Ratio = 0.79-0.86). This effect on lung cancer has been estimated very accurately [as 95% CI is quite narrow].

The model is an acceptable fit to the data with a pseudo $R^2 = 45\%$.
There were no outliers or unexplained structure.

The model fit was a GLM with binomial distribution and logit link function”

When giving a p-value always give an estimate of the effect size as well i.e. the 95% CI.

Reporting and Interpreting Logistic Regression

Interpreting Odds Ratio's as Relative Risks using the rare disease assumption

It's quite common to see the Odds Ratio's (OR) from Logistic Regression interpreted as Relative Risks (RR).

The difference is that Odds Ratios refers to the relative difference in Odds, while Relative Risk the difference in probabilities.

This is because mathematically when an event is 'rare' Odds Ratio's approximate Relative Risks.

Some authors say 10% is rare enough. I disagree and would suggest 1% is the minimum. However it is a subjective decision and if you are unsure then just report and interpret as Odds Ratio's.

There are some other complications as well e.g. some authors say it can't be applied to case control studies. So before interpreting OR's as RR's it's a good idea to read up on it. A good reference is Gerald van Belle (2008) Statistical Rules of Thumb.

Reporting and Interpreting Logistic Regression

Consider reporting the probabilities (or odds) too

OR's compare the relative change in the odds, but ignore the underlying chance of a success happening.

For example; the below might show the % of people who got a dash of cabin fever during the COVID lockdowns of 2020. From 2 studies, one done in Melbourne (which had strict lockdowns) and 1 in Cairns Qld (who were largely unaffected)

- As you can see the Odds Ratio is the same, people with kids were more likely to be effected (I wonder why??).
- However.** Far more people in Melbourne were effected than in Cairns, as expected.

Location	% of people who got Cabin Fever who had Children	% of people who got Cabin Fever who had no children	Odds(Children)	Odds(None)	OR Children vs None	p
Melbourne	75%	50%	3	1	3	0.003
Cairns	25%	10%	0.33	0.11	3	0.007

So, when reporting it can be useful to report both the underlying absolute %'s and their OR i.e.

- People with kids were more likely to exhibit signs of Cabin Fever than people with no kids (Melb-75% vs 50%; OR=3, p=0.003: Cairns-25% vs 10%; OR=3, p=0.007)

Reporting and Interpreting Logistic Regression

Reporting more than 2 Categories

One has to be careful that the wording makes it clear what the reference category is. This is because the p value refers to the comparison to the reference category i.e. the category captured in the intercept, not comparisons between the other groups.

So assuming people with Kids were the reference category we might say: “Compared to people with no kids those with kids were more likely to get Cabin Fever (5+ kids-90% vs 50%; OR=9, $p=0.003$: 1-5 kids: 73% vs 50%; OR=2.7, $p=0.007$)”.

So in this example all the p-values are for comparing to the “No Kids” group. The 2 groups with kids are not directly compared.

% of people who got Cabin Fever who have 5+ kids Children	% of people who got Cabin Fever who have 1-5 kids Children	% of people who got Cabin Fever who had no children	Odds (5+)	Odds (1-5)	Odds none	OR 5+ vs none	p	OR 1-5 vs none	p
90%	73%	50%	9	3	1.00	9.0	0.003	2.7	0.007



Reporting sample %, probabilities and relative risks are not always appropriate

Before reporting sample %'s or probabilities, or metrics that are based on them like relative risks, we first need to decide if they are appropriate and useful.

- They *may* be useful if the study is an accurate representation of the overall population e.g. cross sectional studies.
- They are usually *not* useful if they are not an accurate representation of the overall population e.g. Case-Control Studies.
 - These are where we have a sample of cases e.g. a rare disease, and then compare them to some controls e.g. those without the disease, to understand what the differences are. In such cases the Odds Ratio is still relevant, which is why so much of logistic regression focuses on the Odds Ratio, since it is always useful.

Sample Size: Rule of 10

A common Rule of Thumb is that for stable results one needs 10 observations for each parameter.

This is modified for logistic regression.

Instead of 10 observations/parameter we need 10 events/parameter (or 10 non events if that is less common). E.g.

- A sample of 500 with 20 successes can have a model with 2 parameters
- A sample of 500 with 480 successes can still only have a model with 2 parameters (since we only have 20 failures).



Poisson (count) Regression

Discrete Positive Integer Response e.g. 0, 1, 2, 3, 4.

Workflow Suitable for:

- Counts
- Before After Control Impact design (BACI)



Poisson (count) Regression

Uses the Poisson distribution which assumes the data is a positive integer i.e. 0, 1, 2, 3, 4, etc. So it's a good distribution for counts.

Can also be used to model rates. This is done by adding an offset to the model. This basically divides the count by something to turn it into a rate. Examples

- Cell **concentrations** are actually cell counts divided by volume of blood/plasma/etc. So rather than model the concentration assuming a normal error which often fails we can instead model the counts as a Poisson using the volume as the offset.
- We might have the count of fish caught, and want to divide it by the size of the net so it has no impact on the analysis (otherwise big nets would simply have higher counts which is obvious and not helpful). This is done by adding the net size in m^2 as an offset so we convert the count of fish caught to the amount of fish caught/ m^2 of net.

Newsome T, Chris H, Wirsing A (2020) Restriction of anthropogenic foods alters a top predator's diet and intraspecific interactions

Dingos are an important predator in Australian Landscapes. The meso-predator theory states that increasing them decreases cat/fox numbers and reduces pressure on small natives currently under threat of extinction.

A mine in the Tanami desert had 2 garbage tips which they fenced off. This gave us the opportunity to investigate how this affects dingo feeding behaviour.

4 sites were selected: the 2 mine sites, 1 site that was a long way away from the tips and one that was an intermediate distance away. Scats were collected Before and After the tips were fenced and the # of different types of animals and rubbish found in them were counted.

This gave us a Before, After, Control, Impact (BACI) design. Which has good causal interpretation.



Model Fitting Workflow

Step 0) Clean and check data.

Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

Step 2) Fit the Model

Step 3) Check Model Assumptions via Diagnostics: Residual Analysis

Step 4) Goodness of Fit: Plots and Statistics

Step 5) Interpret Model Parameters and reach a conclusion

Step 6) Reporting

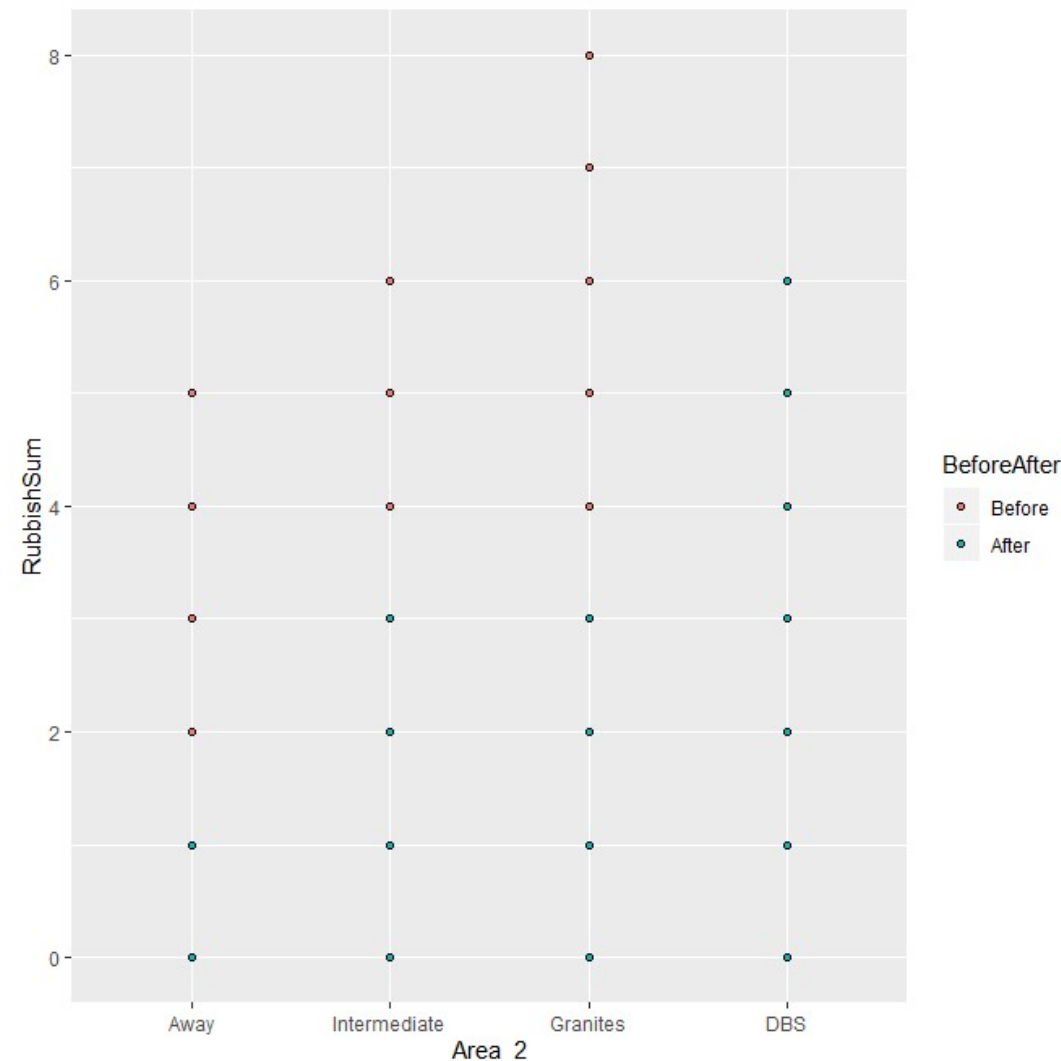
Linear Models 3 and Model Building Workshops have more detail on many of these steps.

Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

So here is a plot for each of the 4 sites. But it's not very good since all the scats are overlaid on each other.

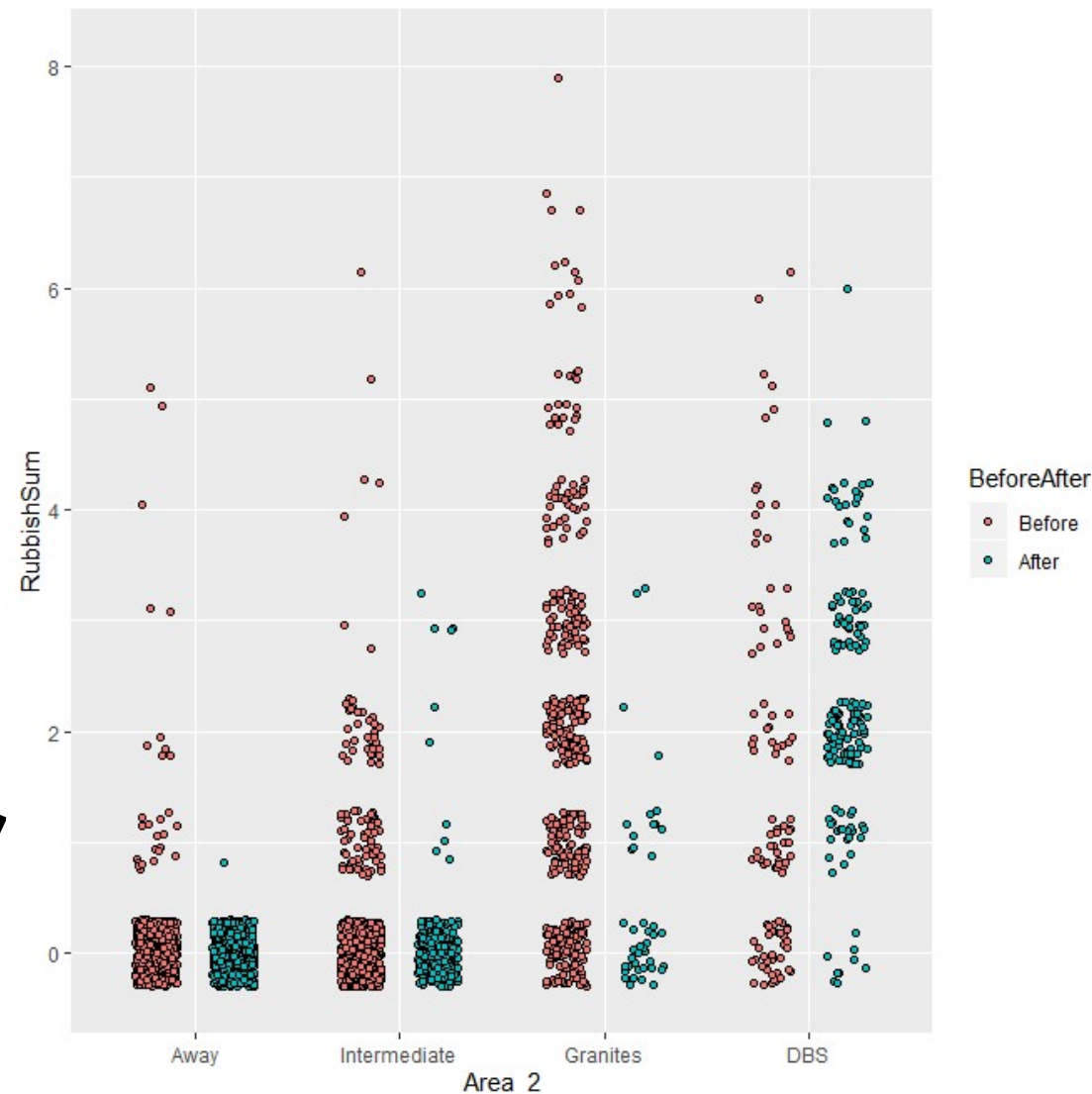
EG: all the Away Scats that had 1 piece of rubbish in them are being plotted at the same point.

```
> windows()  
> ggplot(data = data, aes(x=Area_2, y=RubbishSum,  
  fill=BeforeAfter)) + geom_point(pch=21)
```



Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

- To fix this I add a jitter (to the **plot only**, not the data we model).
- Now I can see that the number of scats with rubbish in them has dropped After the fences were installed. Except at DBS for some reason?
- (The reason was that the fence wasn't as dingo proof, so they all went over there!!)



```
> windows()
> ggplot(data = data, aes(x=Area_2, y=RubbishSum, fill=BeforeAfter))
+ geom_point(pch=21,
position=position_jitterdodge(jitter.width=0.4, jitter.height=0.3))
```

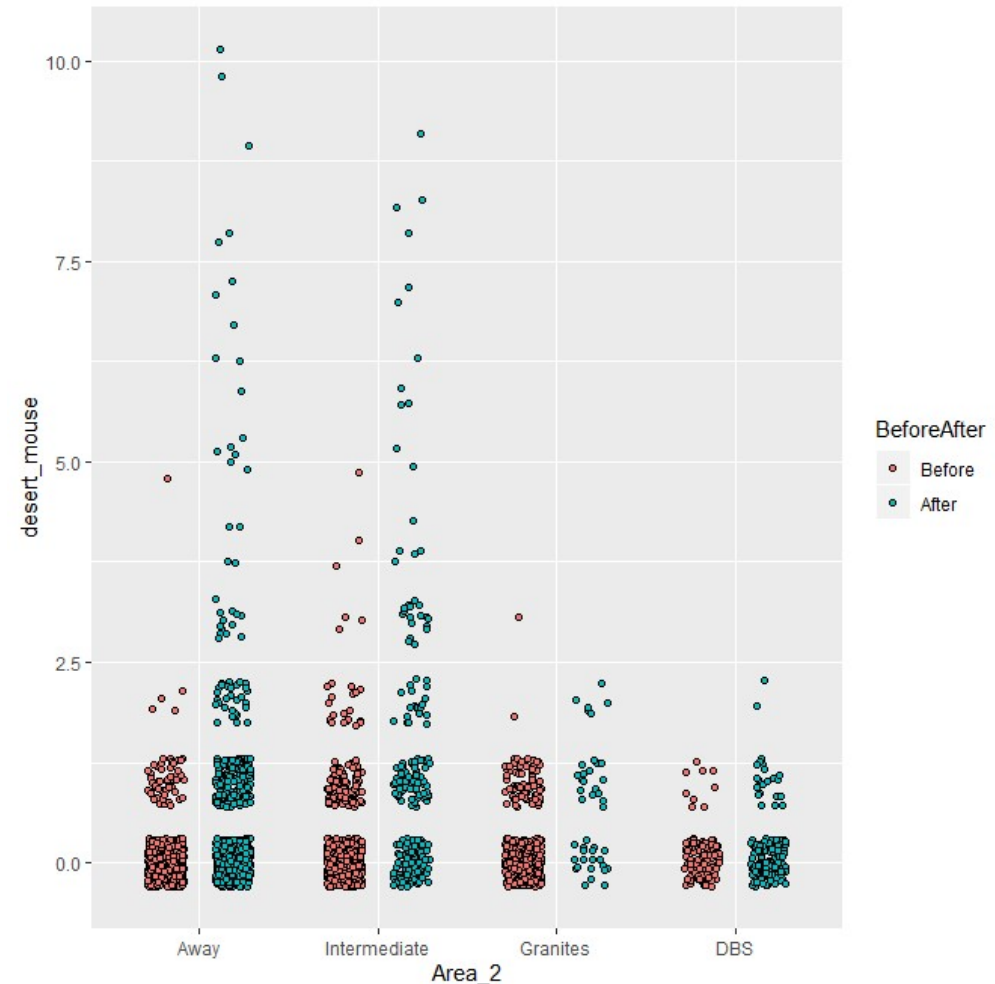


“Graphs allow us to view complex mathematical models fitted to data, and they allow us to assess the validity of such (statistical) models”
(Cleveland 1994, author of “*The elements of graphing data*” and “*Visualising data*”).

Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

- The model I will show you is for the Desert mouse

```
> windows()
> ggplot(data = data, aes(x=Area_2, y=desert_mouse,
  fill=BeforeAfter)) + geom_point(pch=21,
  position=position_jitterdodge(jitter.width=0.4, jitter.height=0.3))
```



Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

Poisson GLM might be a good fit, so lets try that

$$Y_i \sim \text{Poisson}(\mu) \text{ where } \log(\mu) = X\beta$$

We will use a log link since that is the conventional model.
(NB: makes a multiplicative model when we back transform to rates).

Step 2) Fit the Model

```
desert_mouse.p1 <- glm(desert_mouse~Area_2*BeforeAfter,  
data=data, family="poisson")
```

Step 3) Check Model Assumptions via Diagnostics: Zero Inflation

Sometimes we get count data with far too many zero's for the Poisson distribution to handle. This is called Zero Inflation.

It often happens if there are effectively 2 processes occurring:

1. Whether the event occurs
2. If it does occur, how often it does

Simplistically fitting 2 models is often a way around this. We fit a binomial (logistic) model to whether the event occurs, and then a Poisson if it does. There are also special Zero Inflated Poisson (ZIP) models that account for this, and the negative binomial model is often used too.

Step 3) Check Model Assumptions via Diagnostics:

Zero Inflation

A rough test for this is to simulate the number of zero's we expect based on the overall average and then compare it to what we have. If it is very different we may need some type of ZIP model.

Below shows we may have more zero's than the theoretical distribution. But I have seen much worse and this is only rough since it's actually the conditional theoretical distribution we should be comparing to. So it isn't bad enough to be overly worried about.

Theoretical Distribution

0	1	2	3	4
69.25	25.13	4.96	0.60	0.06

Actual Distribution

0	1	2	3	4	5	6	7	8	9	10
75.87	18.60	2.89	1.18	0.38	0.35	0.24	0.17	0.17	0.07	0.07

```
> mean(data$desert_mouse)
> test.0i.theory <- rpois(mean(data$desert_mouse), n=10000)
# better to use proportion with large N since it will be stable.
count of 0's at low n will not be.
> prop.table(table(test.0i.theory))*100
> round(prop.table(table(data$desert_mouse))*100,2)
```


Step 3) Check Model Assumptions via Diagnostics: Overdispersion

For the same reasons explained in Logistic regression Poisson Distributions can be over dispersed i.e. there is too much variance for the single parameter in the Poisson distribution to handle.

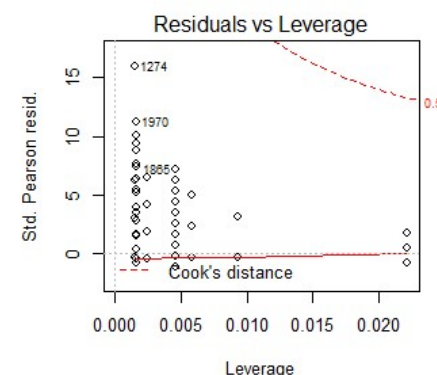
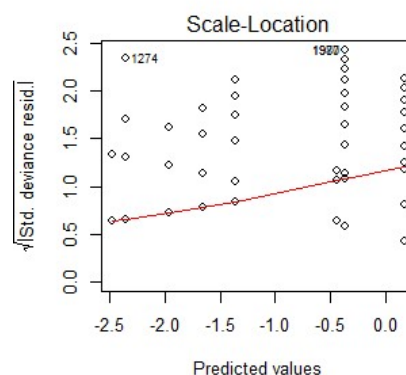
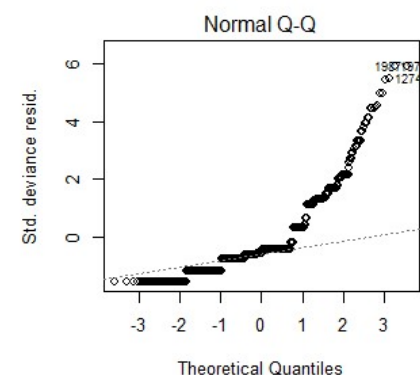
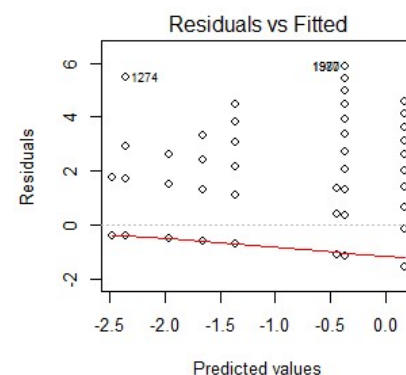
We test this using a function from <https://github.com/lme4/lme4/issues/220>. There is ongoing research on this topic so more recent information may be available here.

It tests whether the dispersion parameter is different to 1, which is what a poison distribution assumes. It tells us that although there is statistical significant overdispersion it is not very large at only 1.6, so not worth worrying about.

```
> is_overdispersed(desert_mouse.pl) #  
      chisq      ratio      rdf      p  
4.467379e+03 1.557663e+00 2.868000e+03 9.576844e-74
```

Step 3) Check Model Assumptions via Diagnostics: Residuals

- No obvious influential outliers
- No systematic patterns we need to account for
 - The discrete lines are caused by the 8 combinations of treatments i.e. 4 sites before and after = 8
- Residuals aren't normal, but nor do we expect them to be. They're Poisson!



```
# Standard plots  
> windows()  
> par(mfrow=c(2,2))  
> plot(rubbish.p1)
```

Step 4) Goodness of Fit: Compare to NULL model

It's a much better fit than the NULL model.

```
> anova(null, desert_mouse.p1, test = "chisq")
Analysis of Deviance Table

Model 1: desert_mouse ~ 1
Model 2: desert_mouse ~ Area_2 * BeforeAfter
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      2875      3477.5
2      2868      2743.8  7    733.69 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Step 4) Goodness of Fit: What is it's Pseudo R-Squared?

Technically there is no Pseudo R-Squared for a GLM, however there is an equivalent based on the % Deviance explained.

Which in this case is acceptable, at 57%

```
> (deviance.explained <- ((deviance(null)-deviance(rubbish.p1))/deviance(null))*100)
[1] 57.37869
```

Step 5) Interpret Model Parameters and reach a conclusion

For Simple Linear models we can simply look at the parameter estimate summary and CI's. BUT in Poisson regression these are hard to interpret as they are still on the log scale (which was our link function).

The only really useful part of this 'raw' output is the p-value associated with the parameters. Which in this case shows strong evidence of Intermediate and Granites being different from Away (Intercept), Before/After and the interactions (which means the Before/After effect differs between sites).

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-2.3638	0.1240	-19.057	< 2e-16	***
Area_2Intermediate	1.0033	0.1475	6.802	1.03e-11	***
Area_2Granites	0.7043	0.1679	4.194	2.74e-05	***
Area_2DBS	-0.1119	0.3557	-0.314	0.753150	
BeforeAfterAfter	1.9915	0.1331	14.960	< 2e-16	***
Area_2Intermediate:BeforeAfterAfter	-0.4518	0.1671	-2.704	0.006842	**
Area_2Granites:BeforeAfterAfter	-0.7715	0.2550	-3.025	0.002483	**
Area_2DBS:BeforeAfterAfter	-1.4796	0.4129	-3.583	0.000339	***

signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step 6) Reporting: Overall Conclusion suitable for publication

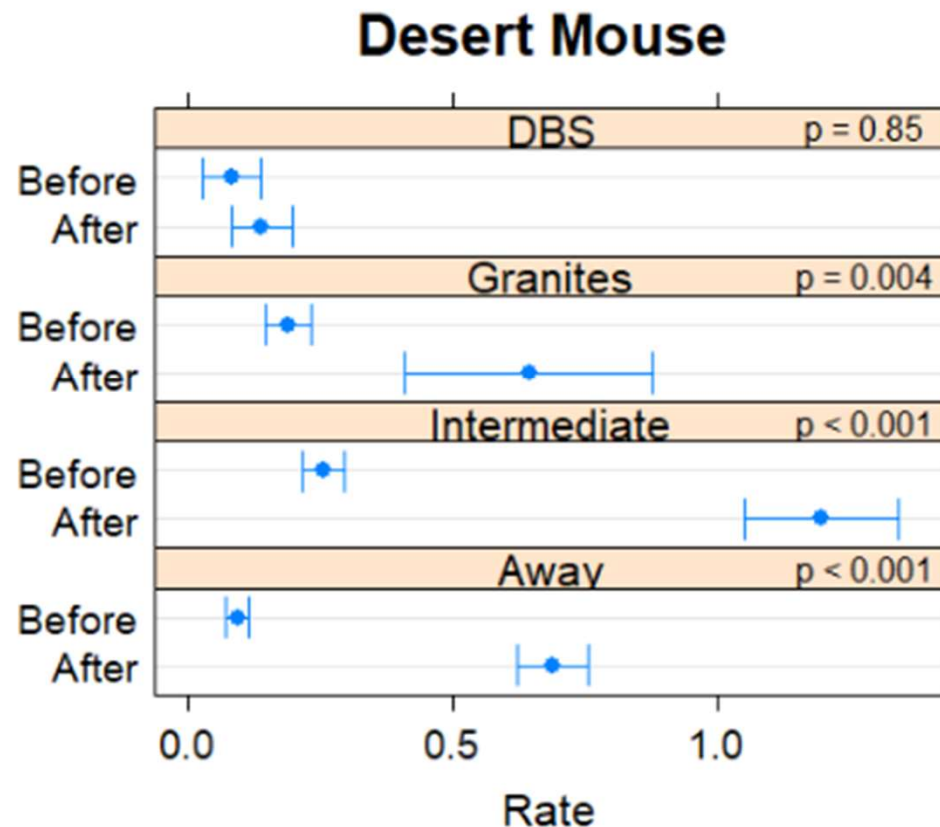
The model is a good fit to the data with a pseudo $R^2=57\%$. There were no outliers or unexplained structure.

The model fit was a GLM with Poisson distribution and log link function. There was no evidence of over dispersion or zero inflation”

Step 6) Reporting: Overall Conclusion suitable for publication

So far our examples have had few predictors and easy interpretation, so the words I've been giving you have been sufficient.

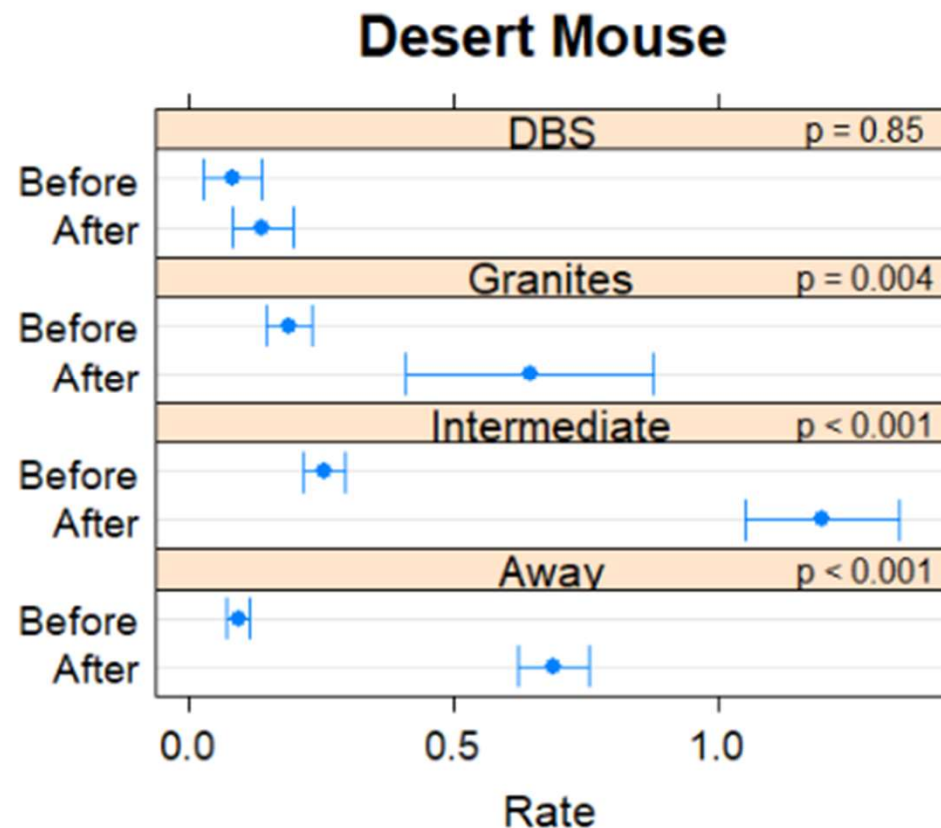
More complex designs with more predictors often require novel reporting methods. And charts are a great way to do that.



Step 6) Reporting: Overall Conclusion suitable for publication

We actually used this chart. Where the p-value at the top right is the specific t-test comparing Before vs After for each site, adjusted for multiple comparisons using Tukeys. The response has been adjusted to the response scale. The interpretation is:

- DBS, where dingos could still access garbage, is the only site where there is no evidence of dingos eating more Desert Mouse after the tips were fenced. This provides strong evidence that anthropocentric food availability can effect dingos diet and the wider Tanami Ecology.
- Interestingly, even at the sites far Away there is very strong evidence of a difference after the tips were fenced with scats having Desert Mouse in them increasing to a rate of [95%CI: 0.6-0.8] from [95% CI: 0.07-0.12] before the tip was fenced. There is strong evidence these rates have changed ($p < 0.001$).

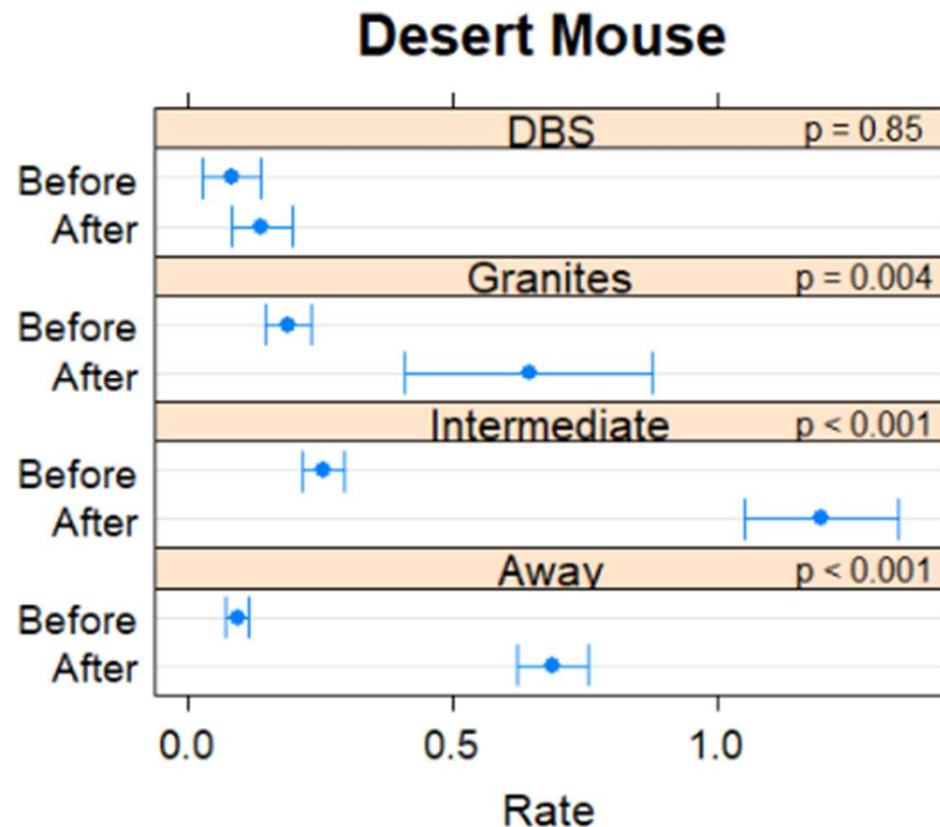


Step 6) Reporting: Overall Conclusion suitable for publication

This type of chart can be used for any GLM.

Not just Poisson.

This is the power of GLM's, similar charts work for all of them. So what you learn for one type of data you can easily apply to other types.



Step 6) Reporting R code



```
?pmmeans
```

```
(desert_mouse.p1.mm1 <- pmmeans(desert_mouse.p1, ~BeforeAfter | Area_2,  
transform="response"))
```

```
# Chart
```

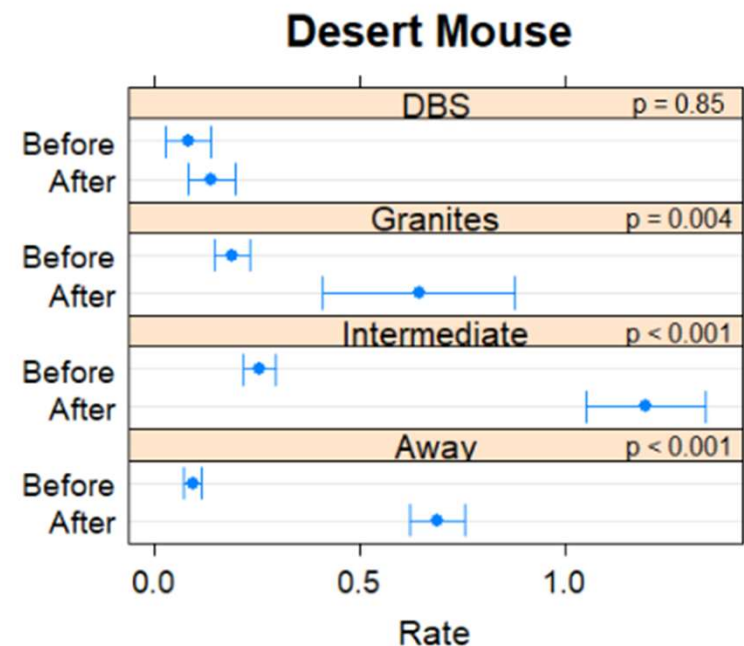
```
windows()
```

```
plot(desert_mouse.p1.mm1, main="desert_mouse")
```

```
# P-values
```

```
(desert_mouse.p1.mm2 <- pmmeans(desert_mouse.p1, specs=c( "BeforeAfter", "Area_2"),  
transform="response"))
```

```
(desert_mouse.pw <- summary(pairs(desert_mouse.p1.mm2)))
```





Other Resources



THE UNIVERSITY OF
SYDNEY



Further Assistance: Sydney University

SIH

- **1on1 Consults** can be requested on our website:
www.sydney.edu.au/research/facilities/sydney-informatics-hub.html OR Google “Sydney Informatics Hub” with the “I’m feeling lucky” button
- **Training** Sign up to our mailing list to be notified of upcoming training:
<https://signup.e2ma.net/signup/1945889/1928048/>
 - Research Essentials
 - Experimental Design
 - Power Analysis
- **Online library.** Useful links and the most recent version of all our workshops.
 - <https://sydney-informatics-hub.github.io/stats-resources/>
- **Hacky Hour**
www.sydney.edu.au/research/facilities/sydney-informatics-hub/workshops-and-training/hacky-hour.html OR Google “Sydney Hacky Hour”

OTHER

- **Open Learning Environment (OLE) courses**
 - **Science:** OLET5608 Linear Modelling: Exploratory data analysis, sampling, simple linear regression, t-tests and confidence intervals. Ability to perform data analytics with coding, basic linear algebra.
 - **Business:** BSTA5007 Linear Models
 - Many others, and constantly changing, so have a look at what is available by getting the list and searching for key words such as linear, regression, GLM, ANOVA, etc.
- **Linkedin Learning:** <https://linkedin.com/learning/>
 - **SPSS** <https://www.linkedin.com/learning/machine-learning-ai-foundations-linear-regression/welcome?u=2196204>

Other SIH workshops

Linear Models 1: Basic intro to *Linear models* with a normal (gaussian) error. Example workflows for Simple Linear Regression, ANOVA, ANCOVA, mixed models.

Linear Models 2: Extends the Linear Model framework introduced in LM1 to *Generalised Linear Models* which allow non normal errors and responses. Example workflows for Poisson (Count) and Logistic (Binary) regression.

Linear Models 3: *Tricks of the Trade* including Interpretation, Reporting and different ways to code categorical data (parametrising the data)

Model Building: LM workshops use simple 1 or 2 predictor examples. More than this requires additional Workflow steps and possibly different Methods to account for things like Multi-Collinearity. These additional topics are covered in this workshop.



Further Assistance

VIDEOS

- StatsQuest with Josh Starmer
 - Linear Models: <https://www.youtube.com/playlist?list=PLblh5JKOoLUlzaEkCLIUxQFjPIIapw8nU>
 - What is a Statistical Model https://www.youtube.com/watch?v=yQhTtdq_y9M
 - Logistic Regression: <https://www.youtube.com/watch?v=yIYKR4sgzl8>
- Zedstatistics, longer videos than StatsQuest. <https://www.youtube.com/c/zedstatistics>

WEBSITES

- R GLMM FAQ <https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html>

BOOKS AND PAPERS

- Julian J Faraway (2006) Extending the Linear Model with R. Chapman & Hall.
- John Fox (2008) Applied Regression Analysis and Generalized Linear Models. Sage.

Tricks to learning – R, linear models, SPSS, etc

- The trick is doing a little bit everyday and getting really good at it so by the time you get to actually needing R you are comfortable in it.
- When working on an actual problem let yourself ‘process’ problems overnight. I’ve lost count of the time times I have battled for hours only to wake up the next day and nail it.
- As tempting as it is. Don’t just google stuff, if you get to know your books and references it will give you a broader understanding, which will help you in the long run.
- Create an R script with your ‘training code’. So as you read the book jump into R and try stuff out. Get used to creating sample data to test stuff out.
- And I’ll leave you with a paraphrased quote from one of the R guru’s Hadley Wickham “Frustration is good, it means you’re at the edges of your understanding and are learning!!”

R: Where to start

BOOKS

- Find an intro R book
 - Read it a little bit everyday, try and get a routine going such as a little at breakfast, before bed, whatever.
- I like this one for a good intro that includes a lot of statistical methods
 - R in Action by Robert I Kabacoff
 - It also has a great web page resource which is a good first port of call too
 - <https://www.statmethods.net/>
 - Buy through Web site for a discount
- Only downside is that it doesn't use Hadley Wickhams packages, so I would also recommend one of his. In particular R for Data Science gives a great intro to data wrangling and visualisation using his packages.
- Finally I recommend MASS (Modern Applied Statistics in S) by Veneables and Ripley. The 'Yellow Bible'. It has at least a little bit on pretty much any statistical method you can think of. I tend to start here to get an intro on what R can do and then research outwards.

ONLINE

- Lots of short (and long) YouTube courses
 - A series of short videos on **Logistic Regression**
<https://www.youtube.com/playlist?list=PLblh5JKOoLUKxzEP5HA2d-Li7IJkHfXSe>

Acknowledging SIH



All University of Sydney resources are available to Sydney researchers **free of charge**. The use of the SIH services including the Artemis HPC and associated support and training warrants acknowledgement in any publications, conference proceedings or posters describing work facilitated by these services.

The continued acknowledgment of the use of SIH facilities ensures the sustainability of our services.

Suggested wording:

General acknowledgement:

"The authors acknowledge the technical assistance provided by the Sydney Informatics Hub, a Core Research Facility of the University of Sydney."

Acknowledging specific staff:

"The authors acknowledge the technical assistance of (name of staff) of the Sydney Informatics Hub, a Core Research Facility of the University of Sydney."

For further information about acknowledging the Sydney Informatics Hub, please contact us at sih.info@sydney.edu.au.

We value your feedback



- We will email you a link to the survey shortly
- It only takes a few minutes to complete (*really!*)
- Completing this survey is another way to help us keep providing these workshop resources free of charge

