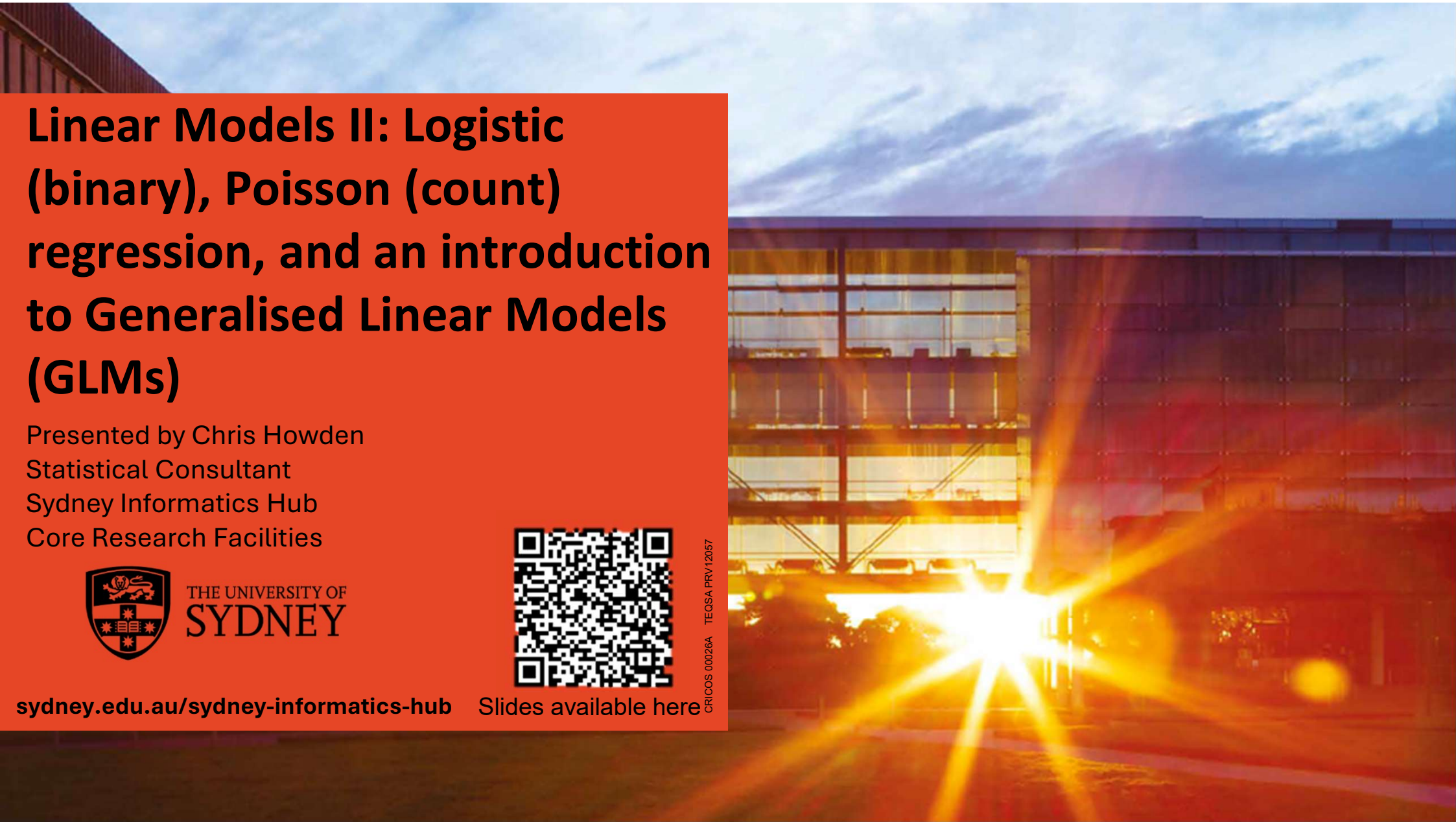


Linear Models II: Logistic (binary), Poisson (count) regression, and an introduction to Generalised Linear Models (GLMs)

Presented by Chris Howden
Statistical Consultant
Sydney Informatics Hub
Core Research Facilities

THE UNIVERSITY OF SYDNEY

sydney.edu.au/sydney-informatics-hub

Slides available here

Acknowledging SIH

- All University of Sydney resources are available to 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 for use of workshops and workflows:

- *"The authors acknowledge the Statistical workshops and workflows provided by the Sydney Informatics Hub, a Core Research Facility of the University of Sydney."*

What is a workflow?

- Every statistical analysis is different, but all follow similar paths. It can be useful to know what these paths are.

- We have developed practical, step-by-step instructions that we call ‘workflows’, that can you can follow and apply to your research.

- We have a **general research workflow** that you can follow from hypothesis generation to publication.

- And **statistical workflows** that focus on each major step along the way (e.g. experimental design, power calculation, model building, analysis using linear models/survival/multivariate/survey methods).

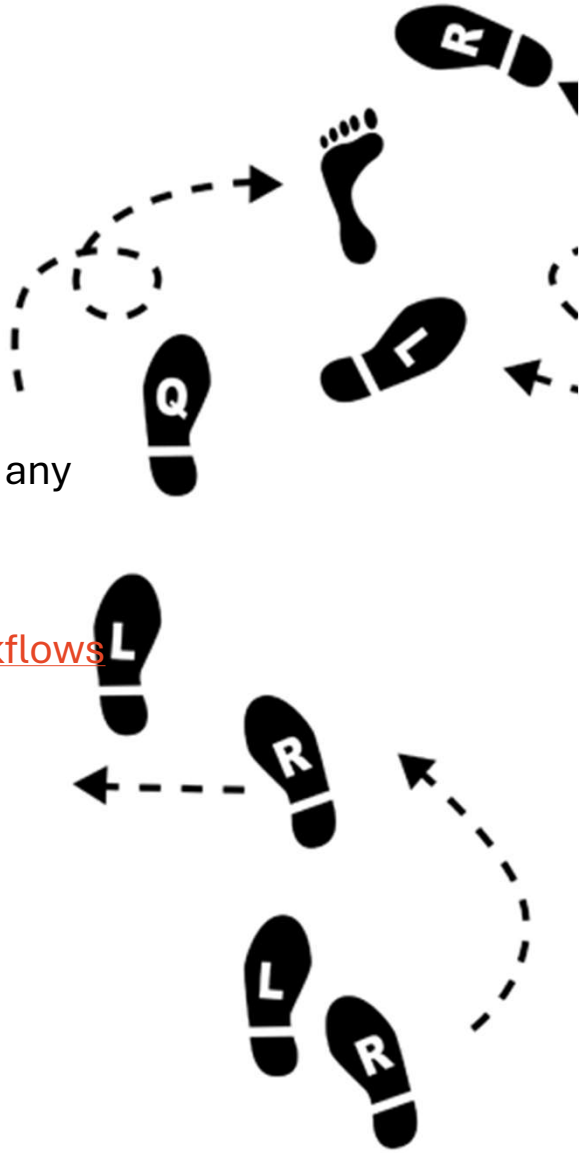
Statistical workflows

- Our **statistical workflows** can be found within our workshop slides.

- **Statistical workflows** are software agnostic, in that they can be applied using any statistical software.

- To access these statistical workflows and more, visit our Workshops and Workflows page.

- The broader aim of our workshop training is to increase understanding and application of statistical concepts, to facilitate higher impact research.



Software workflows

- There may also be accompanying **software workflows** that show you how to perform the statistical workflow using particular software packages (e.g. R or SPSS). We won't be going through these in detail during the workshop. If you are having trouble using them, we suggest you attend our monthly Hacky Hour where SIH staff can help you.

- Our software workflows contain:
    - R code and comments.
    - SPSS syntax as well as screenshots of the point and click procedures and written methods.
    - Screenshots of the point and click procedures and written methods for other bespoke software.

Using Artificial Intelligence (AI) responsibly

THE UNIVERSITY OF SYDNEY

- USyd supports the responsible use of AI in research, aiming to balance innovation with academic integrity. USyd publishes its AI guardrails and endorsed AI tools, and currently allows Microsoft Copilot and Cogniti to be used with public and protected data. Highly protected data must never be entered into these tools.

- The Statistical Consulting Unit (SCU) also supports AI use, but like with any tool, it should be used responsibly. The SCU recommends:
    - Being careful with handling sensitive information.
    - Verifying the accuracy of all AI-generated content.
    - Ensuring the output generated is not false, misleading, or misrepresenting findings.

- This aligns with the Statistical Society of Australia's statement on the use of generative AI in statistics and data science.

- For further AI training, check out the series of GenAI for Research workshops hosted by SIH's Data Science team.

Use AI to enhance, not replace statistical collaboration – it cannot match the nuance of a statistician who understands the research question, study design and constraints.

Using AI responsibly: Tips and tricks for statistical analysis

Ask detailed questions: Align each prompt to a specific step in a statistical workflow of your choice. E.g. for linear models, see below:

Be specific: State your outcome, predictors and how they are measured (numerically or categorically).

Protect your data: Avoid uploading data into AI tools. Instead, describe data structure, or upload simulated data or dummy tables.

Validate code: Check the code is statistically correct and using up to date packages and syntax. Understand what code does before running it.

Think critically: Always review, question and verify any content produced by AI.

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.

Understanding: Before relying on AI, make sure you understand the statistical foundations yourself. Or you won't know when it's wrong.

Check assumptions: AI rarely checks statistical assumptions unless explicitly asked. Check model diagnostic plots.

Statistical method should match study design: E.g. is your model accounting for clustering or repeated measures?

Why, not just what: Ask AI to explain why certain methods are appropriate for your data.

Using AI responsibly: Tips and tricks for R coding

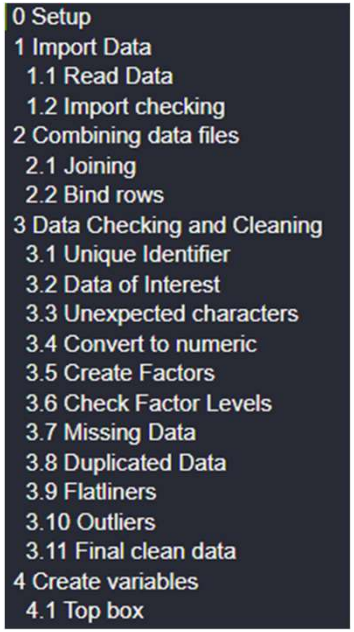
Tip 1: Use **headings and subheadings,** as well as the **document outline** in R Studio to break down the problem into a workflow and think systematically.

- The example below uses subheadings to outline a workflow for data cleaning and processing.
- Breaking the problem into smaller parts leads to simpler questions for AI, that are easier to check.

```
# ... ----
# 3 DATA CHECKING AND CLEANING ----
## 3.1 Unique Identifier ----
# Expecting a unique identifier in the data export.
# We should not have duplicate values as there is one response per respondent.

unique.id <- combined_all # creating a save point!
names(unique.id) # check for the correct variable name and update the code below

# CHECK: the unique identifier is unique
table(duplicated(unique.id$UID))
```



Tip 2: Under each heading/subheading write out the **purpose of the section**. Include AI prompts, as well as ideas for later (coding tends to be iterative).

Using AI responsibly: Tips and tricks for R coding

Tip 3: Every section has a **check**!

- This can be as simple as just having a look at the output, dimensions, or crosstabulations.
- Keep in mind you are responsible for your work and need to ensure your code does exactly what you want it to do. This is especially important when using AI as it can hallucinate!

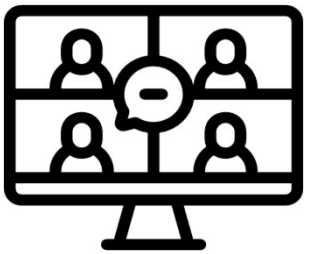
E.g. Should the number of rows change after joining a second data file (number of participants)?

```
> # CHECK: no additional rows
> nrow(raw_data_metro) # original n = 100
[1] 100
> nrow(combined_metro) # combined file should be the same!
[1] 100
```

E.g. Did all values get recoded correctly (were any NAs generated)?

```
> # CHECK: crosstab of original variable and new numeric variable
> # Looks good
> convert.numeric |>
+   count(Q3, Q3_numeric) |>
+   arrange(Q3_numeric)
                        Q3 Q3_numeric  n
1         Strongly Disagree          1 15
2                 Disagree          2 45
3 Neither Agree or Disagree          3 65
4                    Agree          4 42
5           Strongly Agree          5 27
```

How to use our workshops



Workshops developed by the Statistical Consulting Team within the Sydney Informatics Hub form an integrated modular framework. Researchers are encouraged to choose modules to **create custom programs tailored to their specific needs**. This is achieved through:

- Short 90-minute workshops, acknowledging researchers rarely have time for long multi day workshops.
- Providing statistical workflows appliable in any software, that give practical step by step instructions which researchers return to when analysing and interpreting their data or designing their study e.g. workflows for designing studies for strong causal inference, model diagnostics, interpretation and presentation of results.
- Each one focusing on a specific statistical method while also integrating and referencing the others to give a holistic understanding of how data can be transformed into knowledge from a statistical perspective from hypothesis generation to publication.

For other workshops that fit into this integrated framework, refer to our training link page under statistics, found at Workshops and training

During the workshop

- Ask short questions or clarifications during the workshop (either by Zoom chat or verbally). 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 questions will be encountered throughout the workshop.

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().

Generalised Linear Models Framework

Simple Linear Models (workshop 1) vs Generalised 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 on different outcomes and predictors types.

Meaning we only need to write up one set of methods based on GLM, saving time and reducing manuscript length.

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

Generalised Linear Models

The 3 elements of a GLM you need to know, and are the 1st topic of this workshop

When you ask software to do a GLM, it will ask you to specify the:

1. ***Deterministic part of the model*** i.e. relationship between Response (Y) and Predictors (X) ~ defined by the Design Matrix and Linear Predictor (Part 1 of a GLM)
2. ***Random/stochastic part of the model*** i.e. Responses distribution ~ e.g. is it normal (Part 2 of a GLM)
3. ***Link Function*** ~ which links the deterministic model with the random/stochastic model (Part 3 of a GLM)

We’re gonna need some Equations

DON’T FREAK OUT!!!

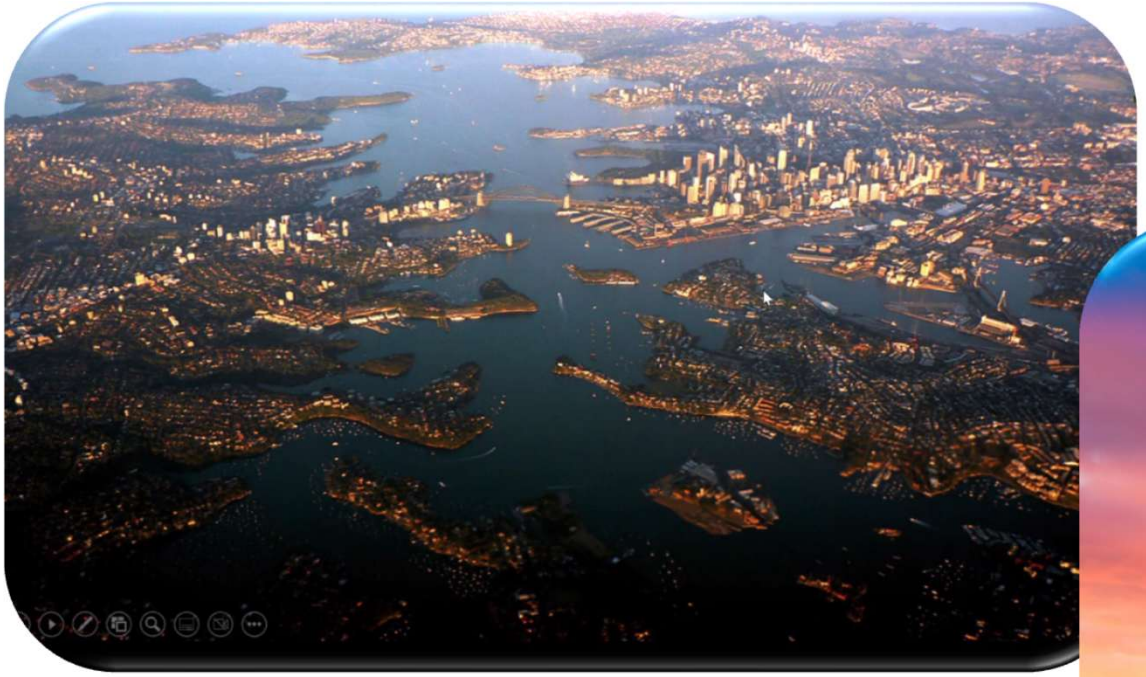
Couple tricks with equations:

- 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.

For example, this equation just means something called Y equals something called Beta Zero plus some Error.

$$Y_i = \beta_o + \varepsilon_i$$

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’re 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.

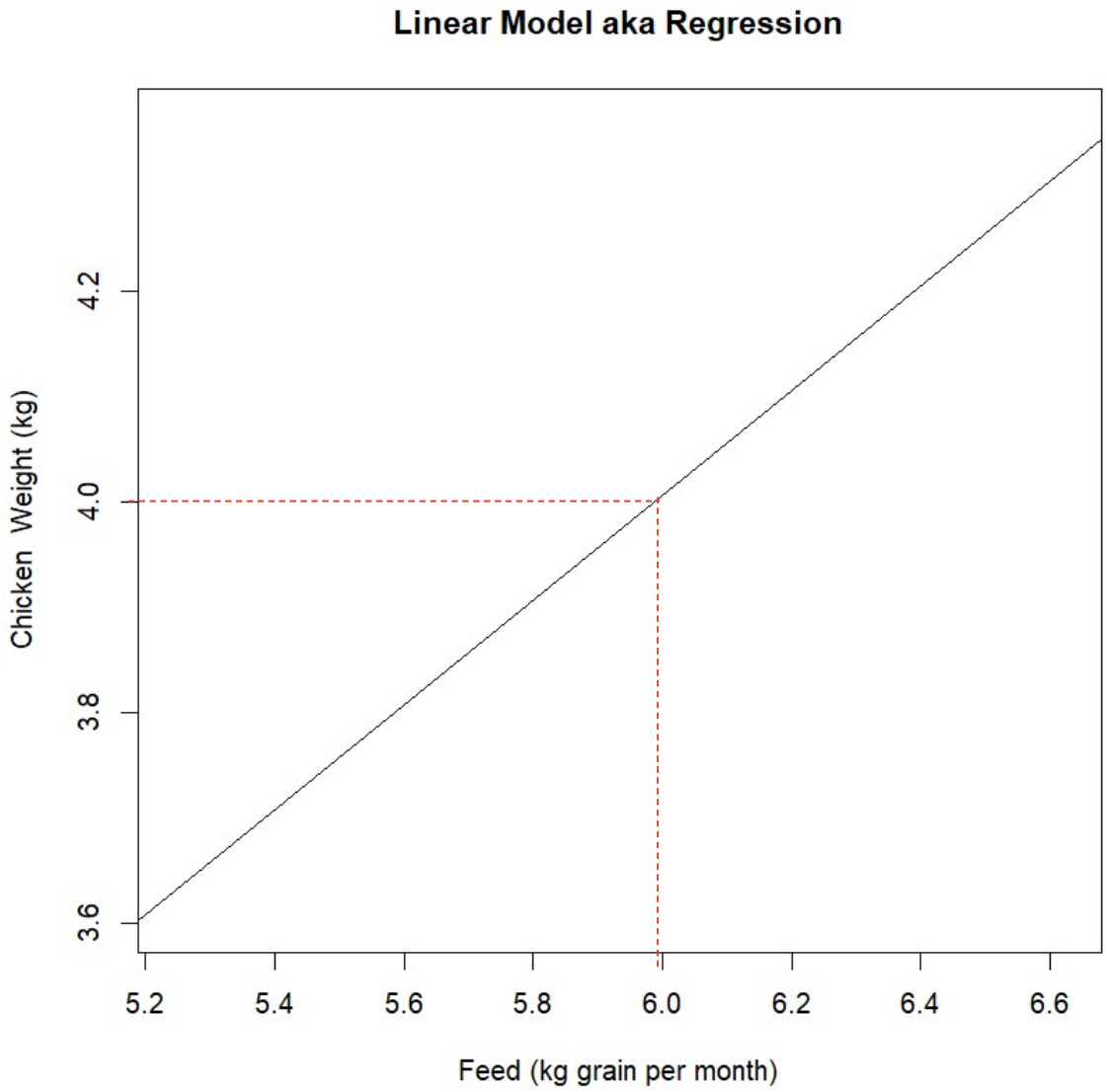
If you can just get the take homes in these red boxes today that’s a great start. The main concepts are:

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

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

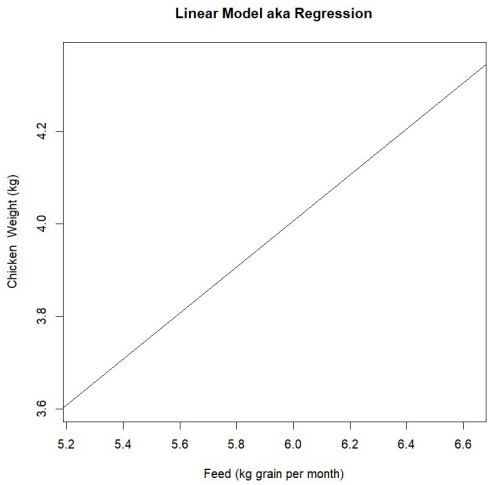
Y = slope (aka gradient) * X + Constant (aka Y intercept)

Y = mX + b

Statistical Equation for a line (puts the constant first)

$\widehat{Y}_i = \beta_o + \beta_1 X_i$

So we want to find β_1, which is the slope(gradient) of the line and repres
Feed has on Weight. (β_o 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!

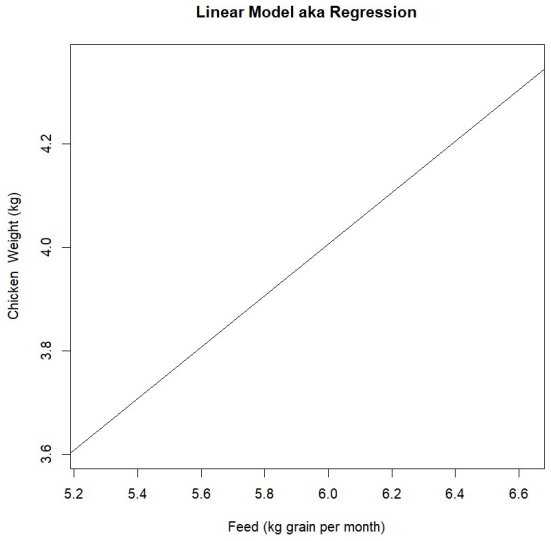
Another name for the natural variance is the ***error*** of the model. Which is why we usually represent it as an ε in the model. In reality, ***model error is usually a combination of natural variance and model error***. Including this natural variation is one difference between deterministic mathematical/physics models and statistical models.

$\hat{Y}$ ~ 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.

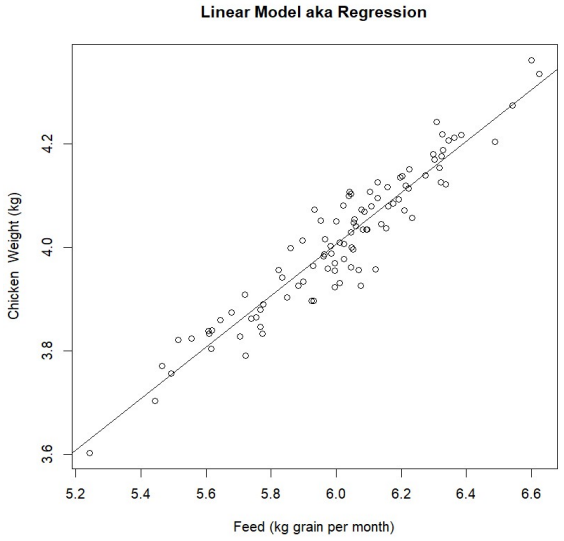
MODEL FOR A LINE

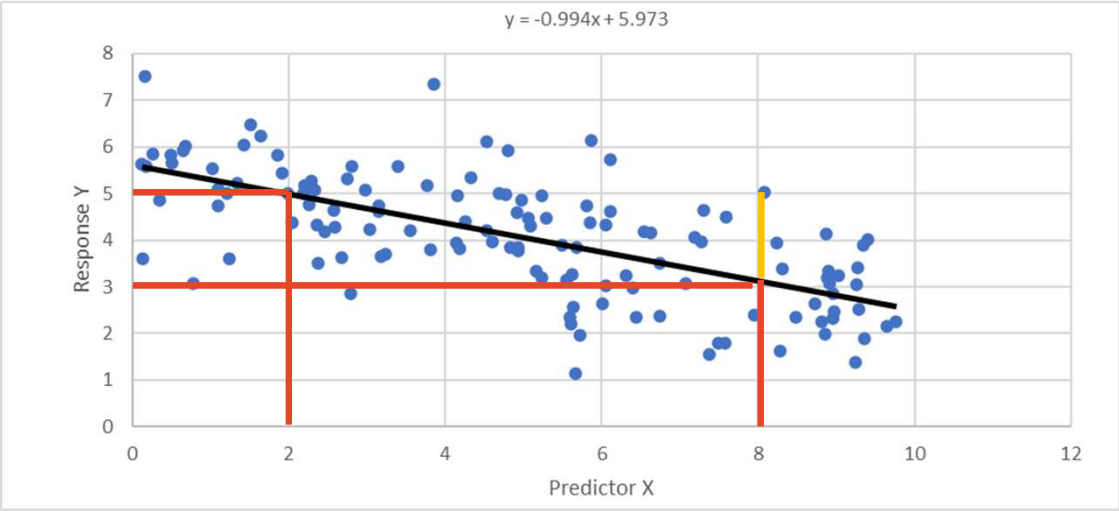
$$\hat{Y}_i = \beta_o + \beta_1 X_i$$



MODEL FOR OUR DATA

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





- The blue points are our data.
- The **black line** is the regression line we use to predict, it's our **model**
    - The red lines are some example predictions along the line. Notice that our prediction is ***conditional*** on what X is e.g. when X=2 our prediction is Y=5. When X=8 we predict Y=3. In other words the prediction of Y is **conditional** on X.
- 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

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

Actual Y value = Linear Prediction + Error/Natural Variation

Data			Design Matrix Parameters		Model Variables	
		Predictors				
Observation i	Response Yi	Continuous X1i	X0i	X1i	Prediction $\hat{Y}i$	Error εi
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 ~ **Response** of Observation i

X_{1i} ~ **Predictor X_1** of Observation i

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

X_{oi} ~ design parameter for parameter β_0 (Constant/Y intercept)

X_{1i} ~ design parameter for β_1 (parameter X_{1i})

Model Variables (variables the model calculates)

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

ε_i ~ **Error of** Observation i

β_0 ~ Constant/Y intercept parameter

$\beta_{1\,i}$ ~ parameter for predictor 1

Simple Regression – Numeric Statistical Model

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

Actual Y value = Linear Prediction + Error/Natural Variation

Data			Design Matrix Parameters		Model Variables	
		Predictors				
Observation i	Response Yi	Continuous X1i	X0i	X1i	Prediction $\hat{Y}i$	Error εi
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. You can fit ***fancy models*** by using a ***fancy design matrix***. Examples of parameters not in the data but are in the design matrix include:
    1. Intercept – so you can remove it to force the line through the origin e.g. calibrations.
    2. Polynomials e.g. add a quadratic term to fit a parabola curve.

Simple Regression – Numeric Statistical Model

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

Actual Y value = Linear Prediction + Error/Natural Variation

Data			Design Matrix Parameters		Model Variables	
		Predictors				
Observation i	Response Yi	Continuous X1i	X0i	X1i	Prediction $\hat{Y}i$	Error εi
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. You can fit ***fancy models*** by using a ***fancy design matrix***. Examples of parameters not in the data but are in the design matrix include:
    1. Intercept – so you can remove it to force the line through the origin e.g. calibrations.
    2. Polynomials e.g. add a quadratic term to fit a parabola curve.

Let’s add another continuous predictor variable

Yellow represents the changes required for this to happen

Multiple Regression

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

Actual Y value = Linear Prediction + 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	
		Predictors						
Observation i	Response Yi	Continuous X1i	Continuous X2i	X0i	X1i	X2i	Prediction $\hat{Y}i$	Error εi
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 X_1** of Observation i

X_{2i} ~ **Predictor X_2** of Observation i

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

X_{oi} ~ design parameter for parameter β_0 (Constant/Y intercept)

X_{1i} ~ design parameter for β_1 (parameter X_{1i})

X_{2i} ~ design parameter for β_2 (parameter X_{2i})

Model Variables (variables the model calculates)

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

ε_i ~ **Error of** Observation i

β_0 ~ Constant/Y intercept parameter

$\beta_{1\,i}$ ~ 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.

Just keep going!!

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

Actual Y value = Linear Prediction + Error/Natural Variation

Data				Design Matrix Parameters				Model Variables	
	Predictors								
Obs i	Response Yi	Continuous X1i	Continuous X2i	Continuous X3i	X0i	X1i	X2i	X3i	Prediction $\hat{Y}i$	Error εi
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 ~ **Response** of Observation i

X_{1i} ~ **Predictor X_1** of Observation i

X_{2i} ~ **Predictor X_2** of Observation i

X_{3i} ~ **Predictor X_3** of Observation i

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

X_{oi} ~ design parameter for parameter β_0 (Constant/Y intercept)

X_{1i} ~ design parameter for β_1 (parameter X_{1i})

X_{2i} ~ design parameter for β_2 (parameter X_{2i})

X_{3i} ~ design parameter for β_3 (parameter X_{3i})

Model Variables (variables the model calculates)

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

ε_i ~ **Error of** Observation i

β_{3i} ~ parameter for predictor 3

β_0 ~ Constant/Y intercept parameter

$\beta_{1\,i}$ ~ parameter for predictor 1

β_{2i} ~ parameter for predictor 2

Multiple Regression

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

Actual Y value = Linear Prediction + Error/Natural Variation

Data				Design Matrix Parameters				Model Variables	
	Predictors								
Obs i	Response Yi	Continuous X1i	Continuous X2i	Continuous X3i	X0i	X1i	X2i	X3i	Prediction $\hat{Y}i$	Error εi
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 ~ **Response** of Observation i

X_{1i} ~ **Predictor X_1** of Observation i

X_{2i} ~ **Predictor X_2** of Observation i

X_{3i} ~ **Predictor X_3** of Observation i

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

X_{oi} ~ design parameter for parameter β_0 (Constant/Y intercept)

X_{1i} ~ design parameter for β_1 (parameter X_{1i})

X_{2i} ~ design parameter for β_2 (parameter X_{2i})

X_{3i} ~ design parameter for β_3 (parameter X_{3i})

Model Variables (variables the model calculates)

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

ε_i ~ **Error of** Observation i

β_{3i} ~ parameter for predictor 3

β_0 ~ Constant/Y intercept parameter

$\beta_{1\,i}$ ~ parameter for predictor 1

β_{2i} ~ parameter for predictor 2

So how do we add Categorical Variables??

Yellow represents the changes required for this to happen

Adding Categorical Variables (e.g. ANOVA)

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

Actual Y value = Linear Prediction + Error/Natural Variation

Data				Design Matrix Parameters			Model Variables	
		Predictors						
Obs i	Response Yi	Continuous X1i	Categorical X2i	X0i	X1i	X2i	Prediction $\hat{Y}i$	Error εi
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 ~ **Response** of Observation i

X_{1i} ~ **Predictor X_1** of Observation i

X_{2i} ~ **Predictor X_2** of Observation i

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

X_{oi} ~ design parameter for parameter β_0 (Reference group = Non-Smoking)

X_{1i} ~ design parameter for β_1 (parameter X_{1i})

X_{2i} ~ design parameter for β_2 (parameter X_{2i} = Smoking)

Model Variables (variables the model calculates)

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

ε_i ~ **Error of** Observation i

β_0 ~ (Reference group = Non-Smoking)

$\beta_{1\,i}$ ~ parameter for predictor 1

β_{2i} ~ parameter for Smoking

Adding Categorical Variables (e.g. ANOVA)

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

Actual Y value = Linear Prediction + Error/Natural Variation

Data				Design Matrix Parameters			Model Variables	
		Predictors						
Obs i	Response Yi	Continuous X1i	Categorical X2i	X0i	X1i	X2i	Prediction $\widehat{Yi}$	Error εi
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 model categorical variables. The way I am showing you is called **Dummy** or **Treatment 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)

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

Actual Y value = Linear Prediction + Error/Natural Variation

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

Just keep going!!

Data				Design Matrix Parameters				Model Variables	
		Predictors							
Obs i	Response Yi	Continuous X1i	Categorical X2i	X0i	X1i	X2i	X3i	Prediction $\hat{Y}i$	Error εi
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 ~ **Response** of Observation i

X_{1i} ~ **Predictor X_1** of Observation i

X_{2i} ~ **Predictor X_2** of Observation i

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

X_{oi} ~ design parameter for parameter β_0 (Reference group = Never Smoked)

X_{1i} ~ design parameter for β_1 (parameter X_{1i})

X_{2i} ~ design parameter for β_2 (parameter X_{2i} = Smoking)

X_{3i} ~ design parameter for β_2 (parameter X_{3i} = Ex Smoker)

Model Variables (variables the model calculates)

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

ε_i ~ **Error of** Observation i

β_0 ~ (Reference group = Never Smoked)

$\beta_{1\,i}$ ~ parameter for predictor 1

β_{2i} ~ parameter for Smoking

β_{3i} ~ parameter for Never Smoked

Adding Categorical Variables (e.g. ANOVA)

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

Actual Y value = Linear Prediction + Error/Natural Variation

Data				Design Matrix Parameters				Model Variables	
		Predictors							
Obs i	Response Yi	Continuous X1i	Categorical X2i	X0i	X1i	X2i	X3i	Prediction $\hat{Y}i$	Error εi
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

Linear Models 3 goes into more detail by:

- explaining how Estimated Marginal Means (EMMs) are an alternative, more flexible, way to interpret the model parameters. They are usually easier with complex models, particularly if interactions are present. They can be invaluable to building an engaging research story.
- discussing ways other than dummy/treatment coding to model categorical variables, such as effects coding.
- having a worked example of how the deign matrix combines with the parameters to give the predictions.

What's the difference with multiple regression?

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

Actual Y value = Linear Prediction + Error/Natural Variation

Data					Design Matrix Parameters				Model Variables	
		Predictors								
Obs i	Response Yi	Continuous X1i	Continuous X2i	Continuous X3i	X0i	X1i	X2i	X3i	Prediction $\hat{Y}i$	Error εi
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	
		Predictors							
Obs i	Response Yi	Continuous X1i	Categorical X2i	X0i	X1i	X2i	X3i	Prediction $\hat{Y}i$	Error εi
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

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

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 categorical variable.
2. Interpretation of the parameters differs.
3. But they are both still ***linear models.***

Representing complex models in a single, simple, concise and generalisable way

Wouldn't it be great if we could represent any linear models study design e.g. ANOVA, regression, ANCOVA, BACI, etc.

Using the same notation?

That would give us a very easy framework to work within.

We wouldn't need to learn lots of different things, and could instead put lots of different analyses into the same 'compartment' in our brain!

We could remember less to understand more!!

The design matrix can represent any model!

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

$= \mathbf{X\beta} + \varepsilon_i$ ~ a shorter and simpler way to write any linear model

= linear/additive model

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

β = vector of parameters

β
β_o
β_1
β_2

ε = vector of errors

Error εi
-0.4
-0.5
0.7
-0.2
0.2
-0.6

The design matrix can represent any model!

$\hat{Y}_i = \beta_o X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + ...$

$= \mathbf{X\beta}$ ~ a shorter and simpler way to write any linear model

= linear/additive model

Let's remove the error to give us the predictive model. This is what the hat over the Y means i.e. it's the prediction of Y given (conditioned on) the X's i.e. it's the conditional expectation (average) of Y.

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

β = vector of parameters

β
β_o
β_1
β_2

The linear predictor (η) can represent any model!

$\hat{Y}_i = \beta_o X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + ...$

$= \mathbf{X\beta}$ ~ a shorter and simpler way to write any linear model

= linear/additive model

= η ~ the linear predictor (the symbol is called eta). The conditional expectation (average) of Y on the data X.

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

β = vector of parameters

β
β_o
β_1
β_2

The **Linear Predictor** (η_i)
Part 1 of the 3 required for a GLM

The linear predictor (η) can represent any model!

$\hat{Y}_i = \beta_o X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + ...$

$= \mathbf{X\beta}$ ~ a shorter and simpler way to write any linear model

= linear/additive model

= η ~ the linear predictor (the symbol is called eta). The conditional expectation (average) of Y on the data X.

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

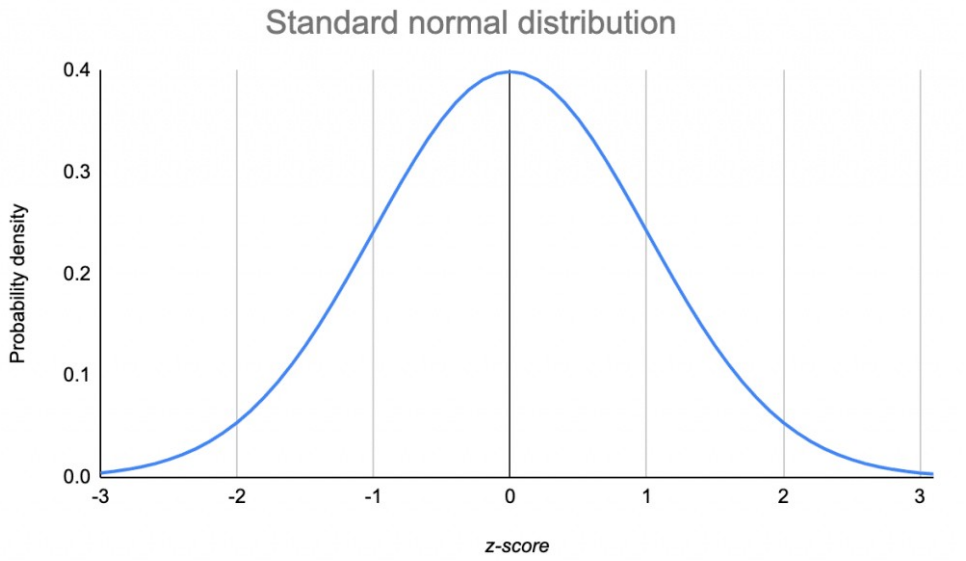
β = vector of parameters

β
β_o
β_1
β_2

The **Linear Predictor** (η_i) Part 1 of the 3 required for a GLM

So far we have assumed a Normal distribution for the Errors

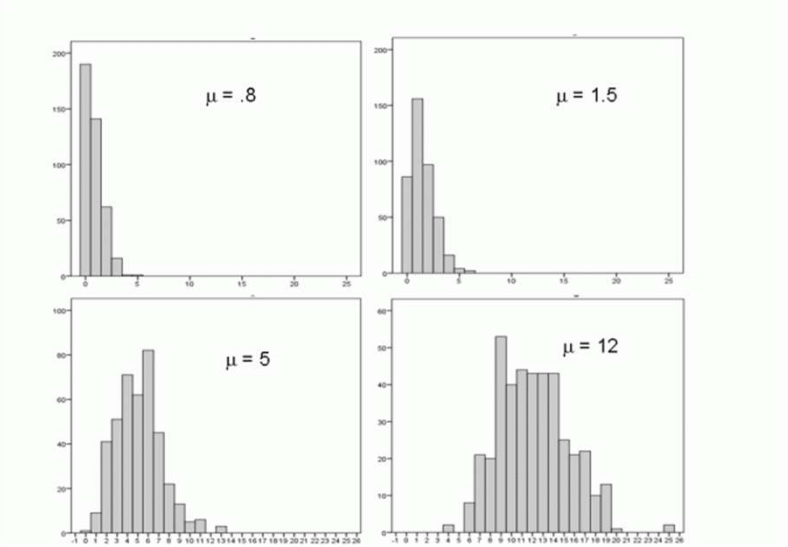
- Response is continuous
  - Ranges from –infinity to + infinity
- 2 parameters describes the curve
  - Mean = μ
  - Variance = σ^2
  - Variance independent of the mean i.e. different data sets with the same mean can have different variance.



BUT, what if it was different, say count data?

Could use the Poisson distribution for the errors instead

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



Different Error Distributions

The **Error Distribution** Part 2 of the 3 required for a GLM

Common Error Distributions

Normal for unbounded continuous data
Poisson for count, rate, positive integer and some log normal data
Binomial for binary data i.e. logistic regression

Different Error Distributions

The **Error Distribution** Part 2 of the 3 required for a GLM

Common Error Distributions

Normal for unbounded continuous data
Poisson for count, rate, positive integer and some log normal data
Binomial for binary data i.e. logistic regression

Let’s look at how we can ***Link*** the ***Linear Predictor*** with the ***Error Distribution*** to model a wide range of variables ***Data Distributions***

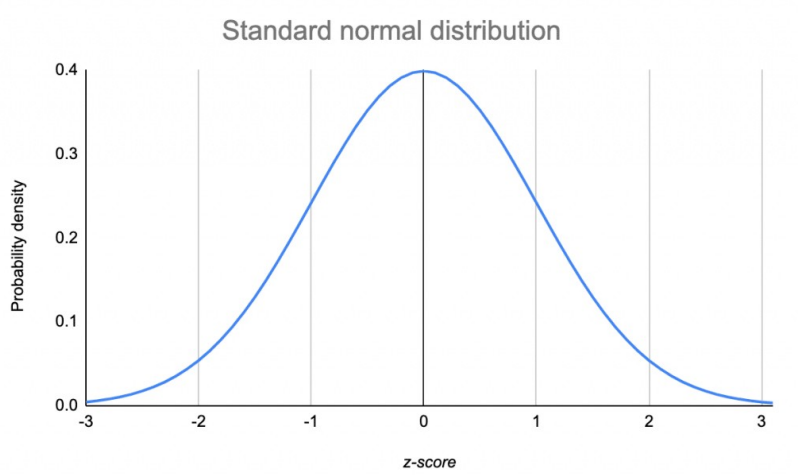
But first: some **new notation so we can succinctly represent** the responses Data Distribution (and **GLMs)**

If Y is normally distributed than we can represent it using this notation

$Y_i \sim N(\mu, \sigma^2)$, where:

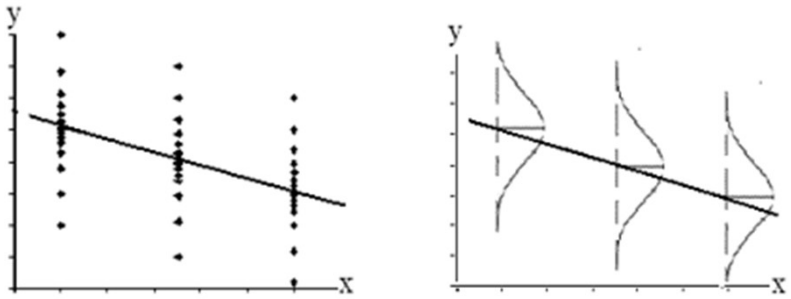
$\mu \sim$ average

$\sigma^2 \sim$ variance



Let's use this new notation to represent a simple linear regression $Y_i = \beta_o + \beta_1 X_{1i} + \varepsilon_i$

In a simple linear regression Y is predicted using a model which is a line, and the error about this line is normal, which looks like this.



So using this notation we can say that Y's Data Distribution is:

$Y_i \sim N(\mu, \sigma^2)$ where:

- **Y's average (μ)** comes from the model line/linear predictor so

$$\mu = \beta_o + \beta_1 X_{1i} = \eta$$

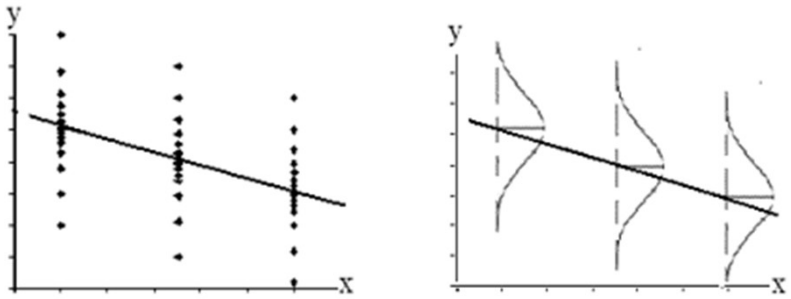
Meaning ***we've LINKED the deterministic relationship η with the random normal errors by setting the normal distributions average (μ) to be the linear predictor ($\eta = \beta_o + \beta_1 X_{1i}$)***

Making the ***Link function μ = η***

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

Let's use this new notation to represent a simple linear regression $Y_i = \beta_o + \beta_1 X_{1i} + \varepsilon_i$

In a simple linear regression Y is predicted using a model which is a line, and the error about this line is normal, which looks like this.



So using this notation we can say that Y's Data Distribution is:

$Y_i \sim N(\mu, \sigma^2)$ where:

- **Y's average (μ)** comes from the model line/linear predictor so

$$\mu = \beta_o + \beta_1 X_{1i} = \eta$$

This means

- That we predict Y's average (μ) for any combination of predictors (X) using the linear model η i.e. Y's average (μ) **is conditional on the predictors (X)**.
- The variance (σ^2) is constant i.e. is not conditional on the predictors (X)

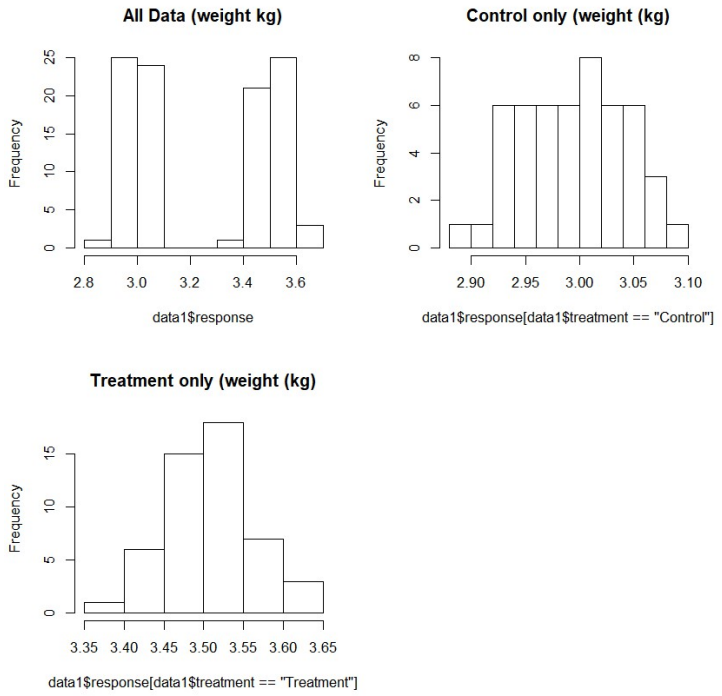
This does not mean the response is normal!!

The response is normal **conditional on what the predictors (X) are**

$$Y_i \sim N(\beta_o + \beta_1 X_{1i}, \sigma^2)$$

Remember the ANOVA from LM1, where the response wasn't normal. But the treatment and control were?

This is because the response has been conditioned on the predictor Treatment.

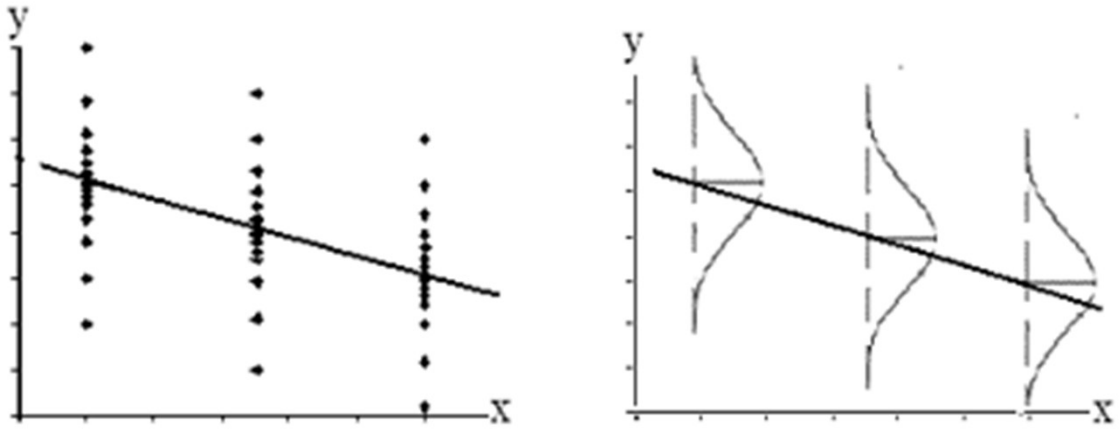


This does not mean the response is normal!!

The response is normal **conditional on the predictors (X)**

$$Y_i \sim N(\beta_o + \beta_1 X_{1i}, \sigma^2)$$

Similarly for the linear regression and ANCOVA examples the response is normal conditional on the predictors. But the 'raw' response is not.



Link functions also let us transform the responses, so they are normal

The previous example directly linked the data distribution with the linear predictor using an ***identify link*** i.e. **$E(Y|X) = \mu = \eta$**

Sometimes the relationship between the random data distributions average (**μ**) and the deterministic linear predictor (η) isn't normal.

Say the data had a big right skew. We might find the log of the response Y is normal, rather than the raw response itself. So we could try a log transformation and link them with a ***log link*.**
$\log[E(Y|X)] = \log(\mu) = \eta$

Log links also allow us to change the additive linear predictor into a multiplicative model

$$Y_i = \beta_o X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + ... + \varepsilon_i$$
$$= \eta_l + \varepsilon_i \text{ = linear predictor + } \varepsilon_i$$
$$= \text{linear/additive model}$$

$$Y_i = \beta_o X_{0i} \times \beta_1 X_{1i} \times \beta_2 X_{2i} \times \beta_3 X_{3i} ..... + \varepsilon_i$$
$$= \textbf{\textit{multiplicate model}}$$

More info and examples to come. For now, just take in that GLM's can use link functions, such as the log link, to 'convert' the linear predictor which is additive from an additive to a multiplicative model. This is how Poisson or Logistic regression become multiplicative, not additive, models.

Link functions also let us fit nonlinear relationships

I won’t explain how now, but you will see an example in the upcoming Logistic regression workflow.

Link functions can do the following

1. Link the data distribution with the linear predictor, to give us a GLM
2. Transform the response so its normal
3. Make the model multiplicative (using an additive linear model)
4. Fit a nonlinear relationship (using an additive linear model)

Link functions can do the following

1. Link the data distribution with the linear predictor, to give us a GLM
2. Transform the response so its normal
3. Make the model multiplicative (using an additive linear model)
4. Fit a nonlinear relationship (using an additive linear model)

GLM components

Part 1 of the 3 required for a GLM

The **Linear Predictor (η_l)** is a ***deterministic*** additive/linear equation of predictors (X) and parameters (β) that will be used to predict the response **($\hat{Y}$),** after linking with the data distribution. It tells us the expected value of the response Y conditional on the data X.

The parameters (β) are defined by the ***Design Matrix*** **(X)**

$$X\beta = \beta_o X_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + ...$$
$$= \eta_l \sim \text{linear predictor (additive/linear model)}$$

Part 2 of the 3 required for a GLM

Different Data Distributions add the ***random/stochastic*** element of the model **e.g. $Y \sim N(\mu, \sigma^2)$**

Part 3 of the 3 required for a GLM

The **Link Function** links Part 1 and 2 together by showing how the distributions average (Part 2) which is the model prediction can be predicted using a function of the Linear Predictor (Part 1) e.g. if the link function is $\mu = \eta = \beta_o + \beta_1 X$ then $Y_i \sim N(\beta_o + \beta_1 X_1, \sigma^2)$ **= Simple Linear Regression.** This also allows us to transform the response and make the model multiplicative.

So let's look at how these 3 things work together to let us model a wide range of different data types

1. **Linear Predictor**
2. **Data Distribution**
3. **Link Function**

We link the error distribution to the linear predictor

We want to predict E(Y|X) using a linear equation ($X\beta = \eta$) of our predictors (X) and some parameters (β). Meaning we need to find a way to **link** our linear equation ($X\beta = \eta$) with E(Y|X) = **μ.** To do that we will:

Use a **Linear Predictor** to model the ***deterministic*** relationship between Y and X

$$\eta = \beta_o + \beta_1 X_1 = X\beta$$

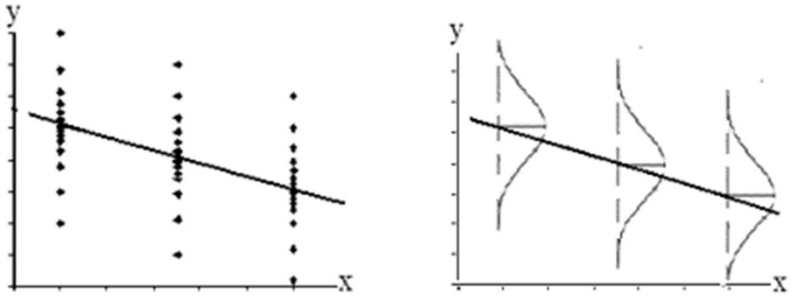
Set Y's ***random*** **Data Distribution** to be Normal (which also defines the errors)

$$Y_i \sim N(\mu, \sigma^2)$$

And then **Link** the deterministic with the random parts of the model by letting **$E(Y|X) = \mu = \eta$** to give us

$$\mathbf{Y_i \sim N(\beta_o + \beta_1 X_{1i}, \sigma^2)}$$

i.e. the **expectation of Y is conditional on X**



Challenge Q: What do we change if the Error was Poisson instead of Normal?

We will use a **Linear Predictor** to model the ***deterministic*** relationship between Y and X

$$\eta = \beta_o + \beta_1 X_1 = X\beta$$

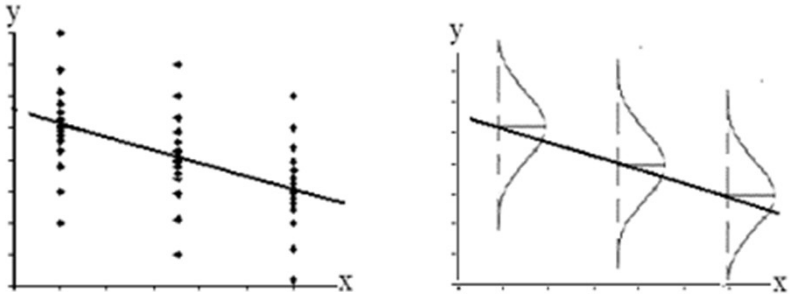
Set Y's ***random*** **Data Distribution** to be Normal

$$Y_i \sim N(\mu, \sigma^2)$$

And then **Link** the deterministic with the random parts of the model by letting **$E(Y|X) = \mu = \eta$** to give us

$Y_i \sim N(\beta_o + \beta_1 X_{1i}, \sigma^2)$

i.e. the **expectation of Y is conditional on X**



Challenge Q: What do we change if the Error was Poisson instead of Normal?

We will use a **Linear Predictor** to model the ***deterministic*** relationship between Y and X

$$\eta = \beta_o + \beta_1 X_1 = X\beta$$

Set Y's ***random*** **Data Distribution** to be Poisson

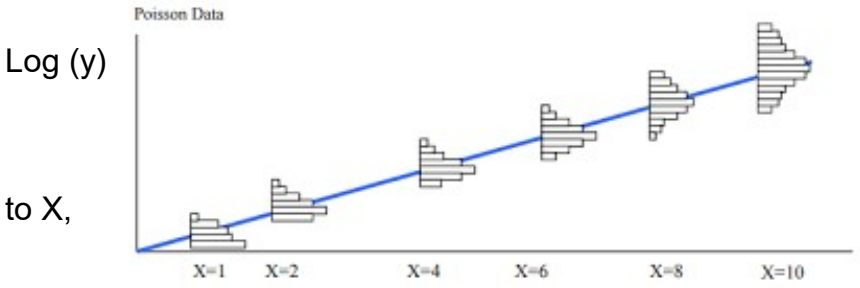
$$Y_i \sim P(\lambda)$$

And then **Link** the deterministic with the random parts of the model by letting $E(Y|X) = \lambda = e^{\eta}$, meaning $\log(\lambda) = \eta$, so we are using a log link function to give us

$$Y_i \sim P(e^{\beta_o + \beta_1 X_1})$$

i.e. the **expectation of Y is conditional on X**

Note that the log link means that Y isn't linearly related to X, Log(y) is.



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

So let’s take a breath and tie everything we learnt in LM1 and so far in LM2 together into a concise summary you can refer back to

Simple Linear Model ~ from LM1 workshops

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

= Deterministic model + Random model

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

~ Gives us a simple, single, unified way of fitting all types of continuous and categorical predictors so we can fit different models like simple linear 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 which are the topic of this workshop.

Data				Design Matrix Parameters				Model Variables	
		Predictors							
Obs i	Response Yi	Continuous X1i	Categorical X2i	X0i	X1i	X2i	X3i	Prediction $\hat{Y}i$	Error εi
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

Simple Linear Model ~ from LM1 workshops

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

= Deterministic model + Random model

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

~ Gives us a simple, single, unified way of fitting all types of continuous and categorical predictors so we can fit different models like simple linear 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 *General**ised*** Linear Models which are the topic of this workshop.

GENERALISED LINEAR MODEL (GLM) ~ todays workshop

Can fit ***all the same models*** as a Simple Linear Model since it uses the same design matrix within its Linear Predictor and can use a Normal distribution plus it:

1. ***Generalises*** the model so we can use **non normal errors/distributions** such as Poisson (for count data) and Binomial (for binary data).
2. **Adds** inbuilt response transformations via the **link** function.

The 3 most common GLM's

Simple Linear Models such as simple linear regression and ANOVA
Y_i ~ N(μ, σ^2) where E(Y|X) = μ, and an ***Identify Link of η = μ*** giving a mean function of E(Y|X) = μ = η hence our model is:

$$Y_i \sim N(X\beta, \sigma^2)$$

Poisson (count) Model ~ also used for rates and concentrations (refer to its example below)
Y_i ~ Poisson(λ) where E(Y|X) = λ, and a ***Log Link of η = log(λ)*** giving a mean function of E(Y|X) = λ = e^{η} hence our model is:

$$Y_i \sim P(e^{X\beta})$$

Logistic (binary) Model
Y_i ~ Binomial(p) where E(Y|X) = p, and a ***Logit (log odds) link of η =logit(p)***
$= ln\frac{p}{1-p}$ giving a mean function of E(Y|X) = p = $\frac{1}{1+e^{-\eta}}$ hence our model is:

$$Y_i \sim B(\frac{1}{1+e^{-X\beta}})$$

GLM components

Part 1 of the 3 required for a GLM

The **Linear Predictor (η_l)** is a ***deterministic*** additive/linear equation of predictors (X) and parameters (β) that will be used to predict the response **($\hat{Y}$),** after linking with the data distribution. It tells us the expected value of the response Y conditional on the data X.

The parameters (β) are defined by the ***Design Matrix*** **(X)**

$$X\beta = \beta_o X_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + ...$$
$$= \eta_l \sim \text{linear predictor (additive/linear model)}$$

Part 2 of the 3 required for a GLM

Different Data Distributions add the ***random/stochastic*** element of the model **e.g. $Y \sim N(\mu, \sigma^2)$**

Part 3 of the 3 required for a GLM

The **Link Function** links Part 1 and 2 together by showing how the distributions average (Part 2) which is the model prediction can be predicted using a function of the Linear Predictor (Part 1) e.g. if the link function is $\mu = \eta = \beta_o + \beta_1 X$ then $Y_i \sim N(\beta_o + \beta_1 X_1, \sigma^2)$ **= Simple Linear Regression.** This also allows us to transform the response and make the model multiplicative.


www.DesktopDress.com

Logistic Regression

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

Workflow Suitable for:

- Continuous predictor

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 often define the disease as the success since we want to know what influences getting it i.e. ***risk factors***. Conversely, we may want to look into ***protective factors***, so we would define those without the disease as the success.

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. ***Can handle data where the event happens for everyone i.e. everyone dies.***
2. 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).
3. Factors in patients lost to follow up (censoring)
4. Uses Hazard Ratios instead of Odds Ratio.
    - These are the ratio of 2 hazards. Hazards are the instantaneous rate of the event (e.g. death or failure) given an individual has survived up to that time (T), they are also the slope/tangent of the survival curve at time T. For a hazard ratio to be a consistent and hence good estimate of 2 hazards over a time interval they need to be proportional over this time period i.e. the slopes need to be parallel, which is why predator this assumption is often called the Parallel Lines assumption.
5. Naturally handles time varying covariates (since it naturally includes time to event while logistic regression does not).
    - 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 pack 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.

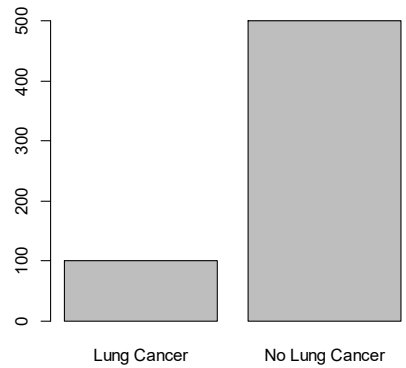
Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

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.

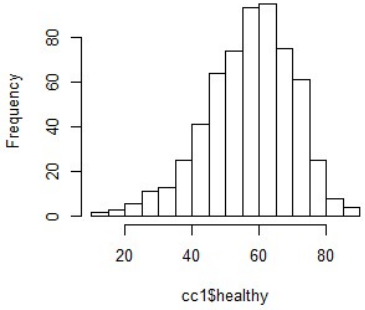
BUT it's a good contender for Logistic/Binary Regression.

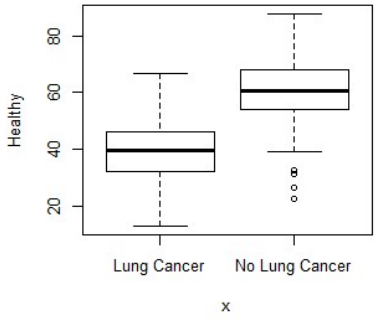


> plot(cc1$"lung cancer")

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

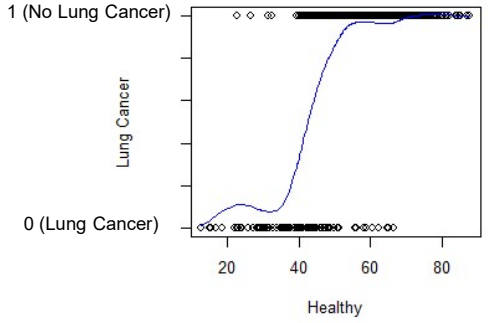
Now add the continuous predictor “healthy lifestyle” which is an index based on things like exercise, food, sleep, etc. It ranges from 0 = unhealthy to 100 = healthy. How might it be related to lung cancer?





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

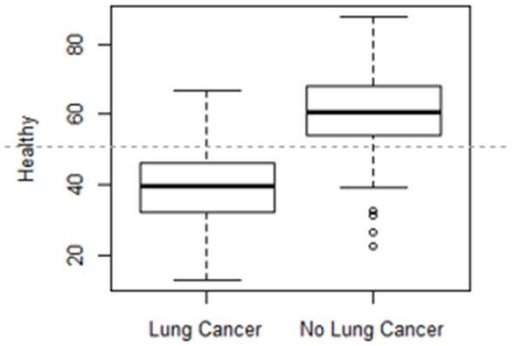
The boxplot and scatterplot show us there is a relationship between healthy and lung cancer.




```
> 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")
```

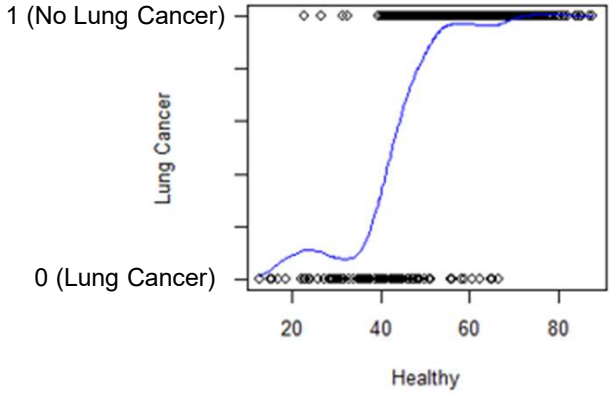
Vertical axis must be numeric for blue smoothing line to be created.

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



This plot show the health indices ***average difference*** between those with and without lung cancer. Its quantified using an **ANOVA** like we did in LM1.

It allows us to predict the health index score (continuous response) 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 (binary response) if we know their health score.

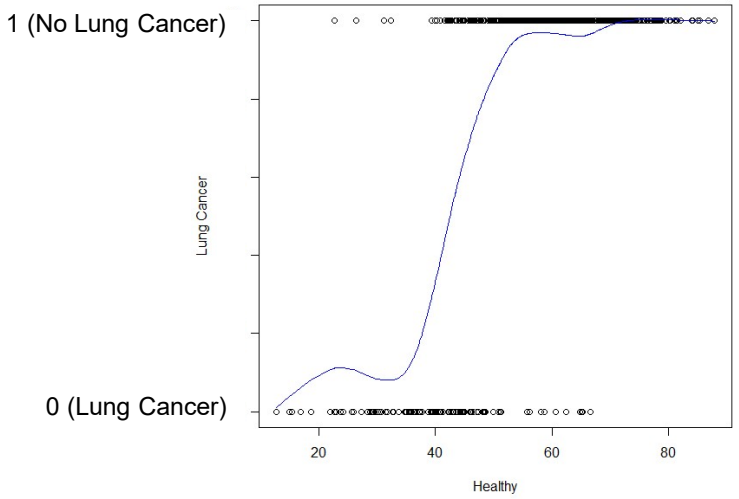
Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

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.



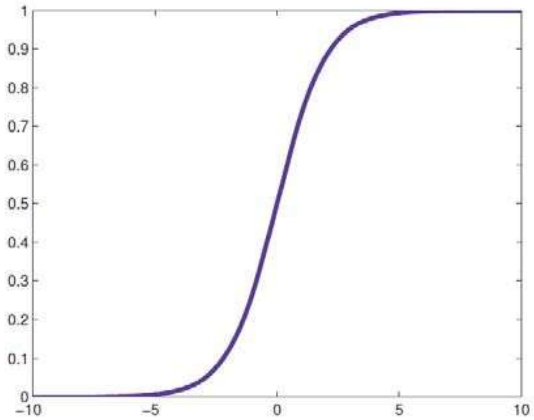
Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA).

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

$Y_i \sim$ Binomial(μ) where **logit**(μ)=$\ln \frac{p}{1-p} = X\beta$ (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 (log odds) link** function lets us fit this sigmoid function.
(And makes it multiplicative model when we back transform to odd's ratios).

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 successes 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 it's effectively trying to divide by 0).

Separation often causes error messages like "failed to converge", warning messages like "! glm.fit: fitted probabilities numerically 0 or 1 occurred" or high parameter SE's.

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

Even if we don't have complete separation, marginal separation can still cause problems such as loss of power caused by upward biased p-values due to elevated SE's and the Hauck-Donner effect.

	Estimate	**SE**
Constant	7.9	0.06
Smoker	1000	597000

Step 1) If we had categorical variables such as smoking

Solution to separation can be merging/collapsing categories. Common problems are:

Age categories that are too fine can have ***empty cells*** with no-one in them e.g. one would merge the last 3 columns to have a 75-85 category. One does need to be careful as it's a different age range, so some might merge them all into 10 year brackets.

	35-40	**40-45**	**45-50**	**50-55**	**55-60**	**60-65**	**65-70**	**70-75**	**75-80**	**80-85**
Success	100	45	35	25	26	31	27	15	8	0
Failure	10	64	46	24	24	28	32	13	6	4

Step 1) If we had categorical variables such as smoking

Modelling lots of interactions and high order interactions between lots of variables increases the chance of empty cells, so its important to check sample sizes of all interactions before modelling them. High order interactions can be evaluated using tables like the below rather than the more usual 2x2 contingency table.

For example: if you did a survey of skiers in Japan you might have plenty of people with red or black hair, brown or black eyes, and who are Scottish or Japanese. But it would be rare to find an ethnically Japanese person with red hair and green eyes!

Hair Colour	Eye colour	Ethnicity	Count
Red	Green	Scottish	15
Red	Green	Japanese	0
Red	Black	Scottish	12
Red	Black	Japanese	2
Black	Green	Scottish	98
Black	Green	Japanese	104
Black	Black	Scottish	74
Black	Black	Japanese	98

Step 2) Fit the Model

cc.model <- glm(lung.cancer ~ healthy, family=binomial(link= “logit”, data=cc2)

Linear Predictor is	***lung.cancer ~ healthy***
Data Distribution is	***family=binomial***(link= “logit”)
Link Function is	family=binomial(***link= “logit”***) ***i.e. log odds***

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

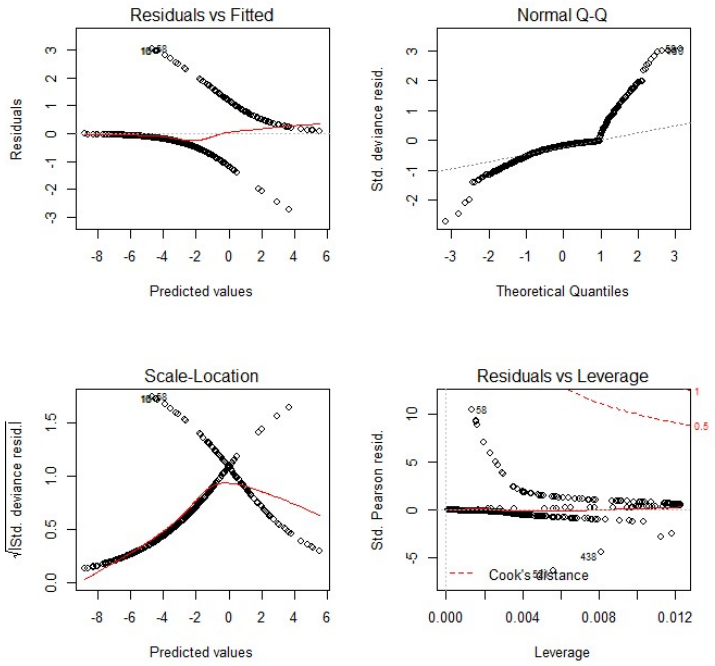
Step 3) Check Model Assumptions via Diagnostics: Residual Analysis

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.

Dharma residuals are more useful and are in the R workflow which can be downloaded from our online library.

```
> windows()
> par(mfrow=c(2,2))
> plot(cc.model)
```



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 the below does, notice the SE is an order of magnitude larger than the estimate i.e. times 10/add a zero. *Causing the Wald p-value to be inflated.* Often caused by separation, which we hopefully identified during the EDA. However marginal separation can be hard to identify during EDA, particularly if we are fitting a lot of 2 way interactions with a lot of cells.

```
Coeffects:
                          Estimate  Std. Error  z value    Pr(>|z|)
(Intercept)   7.9844     0.88840      8.987    <2e-16 ***
Healthy      -0.19048    18.560      -0.01026   0.99
```

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

As previously mentioned during the EDA stage (and copied below) a large SE can be a sign of separation.

Complete separation occurs when we have cells that are entirely successes 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 it's effectively trying to divide by 0).*

Separation often causes error messages like "failed to converge", warning messages like "! glm.fit: fitted probabilities numerically 0 or 1 occurred" or high parameter SE's.

Even if we don't have complete separation, marginal separation can still cause problems such as loss of power caused by upward biased p-values due to elevated SE's and the Hauck-Donner effect.

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

	Estimate	**SE**
Constant	7.9	0.06
Smoker	1000	597000

Using Likelihood Ratio Tests (LRT) instead of Wald z scores if there is marginal separation causing the Hauck-Donner

The p values and z scores most packages give you for each parameter come from Wald z scores. Marginal separation and very small/large parameter estimates with predictions close to 0/1 can cause the Hauck-Donner effect, which makes Wald z scores unreliable. (The relationship between the parameter estimate and Wald z score falls apart, it is no longer monotonic increasing – meaning a larger parameter estimate can lead to a smaller Wald Z score).

The solution is to use a Likelihood Ratio Test where each parameter is evaluated by adding them one at a time i.e. compare 2 models with and without the predictor in question using a LRT.

So consider using this LRT method and ignoring the Wald Z scores if there is marginal separation, very large parameter SE's or estimates, or predictions are close to 0/1 i.e. you get this warning "! glm.fit: fitted probabilities numerically 0 or 1 occurred.

```
Coeffects:
                    Estimate  Std. Error  WALD z value    Pr(>|z|)
(Intercept)  7.9844    0.88840      8.987                  <2e-16 ***
Healthy     -0.19048   18.560      -0.01026         0.99
```

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

Statistical models describes how we partition the data into deterministic vs random information.

- The **null** model is 1 extreme and represents pure random variation. It's the model without any predictors and only a constant/intercept e.g. the % of people with and without lung cancer.
- The **saturated** model is the other extreme and is completely deterministic. It is overfit with as many parameters as data, it tells us no more than the data and will usually be uninformative.
- The model we are fitting is a mix of deterministic (predictors) and random information. We want it to hit the sweet spot between the null and saturated model so we can infer and make predictions that work for the general population, not just this sample.

It's always worth comparing our model to the null model, to see if our predictors are doing a better job than random chance i.e. no information at all.

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

The test used is a Likelihood Ratio Test (LRT), if the models are nested and have the same data. One drawback is that the LRT makes the asymptotic assumption that the chi-square distribution approximates the null distribution of likelihoods. In other words, at small sample sizes it may not be particularly accurate. As such the F test (which is a specific type of LRT) might be better if the error is normal and sample sizes small - as it doesn't require the LRT asymptotic assumption since it's the actual ratio of 2 chi-squared variables.

```
> 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
```

REFERENCES

Low sample size LRT vs F Test

Difference between LRT and ANOVA

P values give us the weight of evidence in making a yes/no decision, they don't make it for us

Don't simply make no brain decisions like 'yes there is an effect' if the p-value is < 0.05.

If p = 0.05, this means you can be wrong as often as 1 in 20.

A p = 0.0000001 is much stronger evidence!! i.e. wrong 1 in a million.

A p=0.049 and p=0.051 is about the same evidence.

Use the p-value to understand your decision. Are you saying there is an effect with a lot of confidence since there is a very small chance you have made the wrong decision (p=0.0000001), or should you be a bit cautious in saying there is an effect since there is a high chance you have made the wrong decision (p=0.05)?

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 interpret the parameters.

BUT in logistic regression since we used a logit link these are hard to interpret as they are 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).

```
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
```

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

Step 5) Interpret Model Parameters and reach a conclusion - Using Odds Ratios (OR)

The parameters can be made more interpretable by taking their exponential since this turns them into ***odds ratios*** (which will be explained shortly).

Remember how the logit link used a log transform? Well, taking their exponential is the inverse of this, which puts them back into the original scale. And then some fancy math means we can also interpret them as odds ratios.

Taking the exponential is similar to taking something to the power10. 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!! You don't need to know why we use an exp, just accept and use it!

For 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 $\mathbf{2.718}^{-0.19048} = \mathbf{e}^{-0.19048} = \mathbf{exp}(-0.19048) = 0.83$.

Step 5) Interpret Model Parameters and reach a conclusion - Using Odds Ratios (OR)

We get the below OR=0.83 for the continuous variable Health, which 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 odds 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
```

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_o)	8.0	0.89	9.0	<2e-16			
Health index ($\beta 1$)	-0.19	0.019	-10	<2e-16	0.83	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 1point 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.

This is because we don't just care about statistical significance i.e. is the effect real. But also how big is the effect i.e. do I care? Also known as the difference between statistical significance and practical/clinical significance.

So what exactly is an Odds Ratio (OR)?

It's best described with an example.

Say the OR for smoking on whether you get lung cancer is 3. This means the odds of getting lung cancer if you smoke is 3 times the odds of getting it if you don't smoke. In other words, ***an odds ratio is the ratio of two odds***.

And what is an "odds"? The odds of something happening is related to its probability, but isn't the same.

Say the ***probability/chance/risk*** of getting lung cancer if you smoke is 75%. Then the corresponding ***odds*** are p/(1-p)=75/25 = 3:1 = 3. These are obviously different numbers with different interpretations, which is why odds ratios can be used to comment on the odds of something occurring, not its probability, chance or risk.

You would have seen it in horse racing too e.g. if Phar Lap tends to win 19 out of 20 races than the odds of Phar Lap winning are 19:1 = 19/1 = 19. On the other hand, the probability of Phar Lap winning is 19/20 = 95%.

Risks report the # of events in relation to the *# of trials* i.e. # events vs # trials.

Odds report the # of events in relation to the *# of nonevents* i.e. # events vs # nonevents.

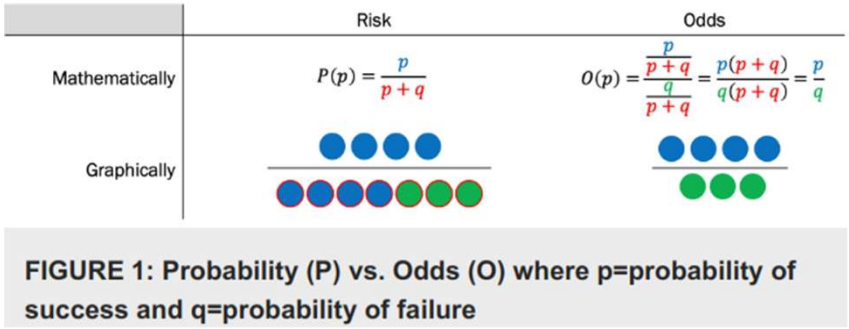


FIGURE 1: Probability (P) vs. Odds (O) where p=probability of success and q=probability of failure

Figure from George A, Stead TS, Ganti L. What's the Risk: Differentiating Risk Ratios, Odds Ratios, and Hazard Ratios? Cureus. 2020 Aug 26;12(8):e10047. doi: 10.7759/cureus.10047. PMID: 32983737; PMCID: PMC7515812.

Odd Ratios (OR) are different to the Relative Risk (RR)

Relative Risk (RR) is the ratio (relative difference) of probabilities. The Odds Ratio (OR) is the ratio (relative difference) of odds. Meaning they have different interpretations ***so be careful what language you use when communicating results.***

If the **OR** of smoking on getting lung cancer is 3, then you need to say the **odds** of getting lung cancer if you smoke is 3 times the ***odds*** of getting it if you don't smoke.

If the **RR** of smoking on getting lung cancer is 3, then you need to say the **chance** of getting lung cancer if you smoke is 3 times the ***chance*** of getting it if you don't smoke.

Incorrectly interpreting ORs as RRs can exaggerate the impact as ORs underestimate the RR when both are <1 and overestimate it when >1.

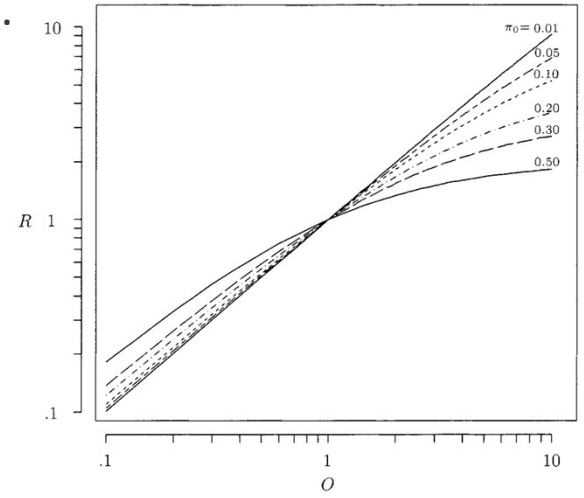


Fig. 6.1 Relationship between odds ratio O and relative risk R as a function of π_0, the background rate in the unexposed. Note that scale is logarithmic.

Gerald van Belle (2008) Statistical Rules of Thumb

Expert Trick 1) Interpreting Odds Ratios (OR) as Relative Risks (RR) using the rare disease assumption

The medical literature commonly interprets odds ratios from logistic regression as relative risks.

This is because **when an event is 'rare' odds ratios approximate relative risks.** The plot below shows that when the incidence is 1% the OR and RR closely follow the 1:1 equivalence line, but become different very quickly as one moves away from 1when the incidence is as low as 5% (plot is from Gerald van Belle (2008) Statistical Rules of Thumb).

So, although some authors say 10% is rare enough. I disagree and would suggest **1% is the maximum**. However, it is a subjective decision and if you are unsure then just report and interpret as an odds ratios.

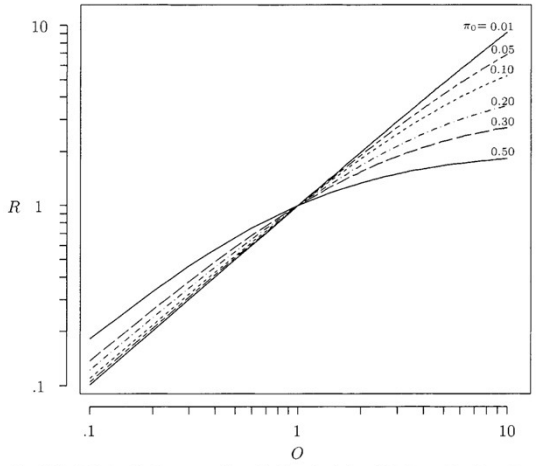


Fig. 6.1 Relationship between odds ratio O and relative risk R as a function of π_0, the background rate in the unexposed. Note that scale is logarithmic.

There are other complications as well e.g. this assumption usually can't be applied to case control studies meaning they always need to report odds ratios irrelevant to how small the incidence is. So before interpreting OR as RRs it's a good idea to read up on it, a good place to start is Gerald van Belle (2008) Statistical Rules of Thumb (which is where the plot on the left comes from).

Why OR underestimates the RR when both are <1 and overestimates it when >1

Because if we have p + q trials when we reduce p this means q has to increase. But this only impacts the numerator in the risk. While both the odds numerator and denominator are affected in opposite directions, so it falls faster. Similarly, if p increases the OR increases quicker.

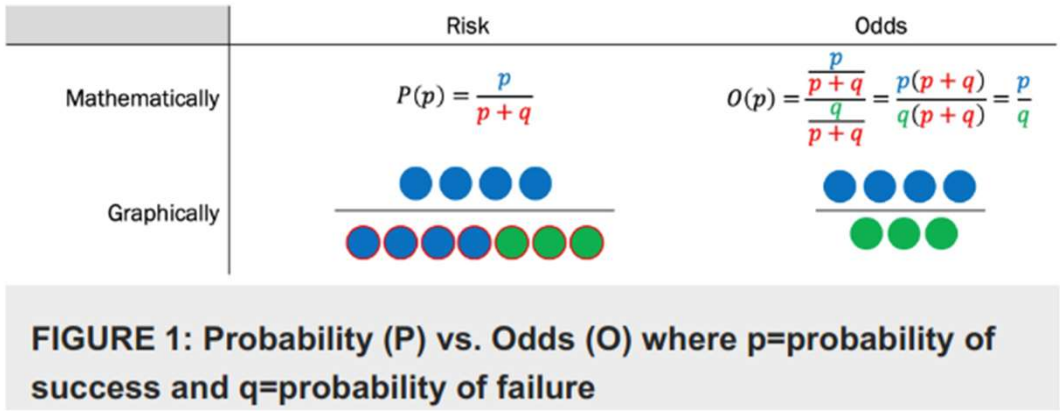


FIGURE 1: Probability (P) vs. Odds (O) where p=probability of success and q=probability of failure

George A, Stead TS, Ganti L. What's the Risk: Differentiating Risk Ratios, Odds Ratios, and Hazard Ratios? Cureus. 2020 Aug 26;12(8):e10047. doi: 10.7759/cureus.10047. PMID: 32983737; PMCID: PMC7515812.

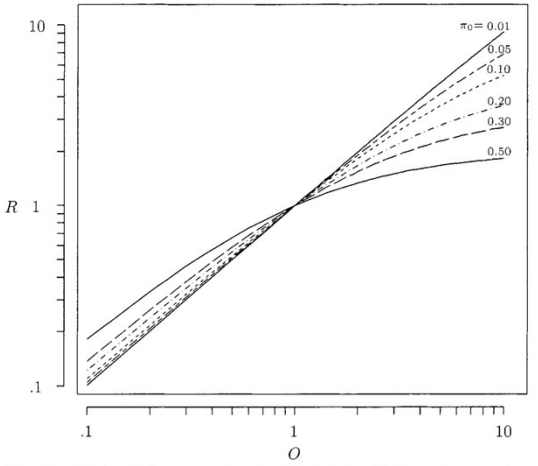


Fig. 6.1 Relationship between odds ratio O and relative risk R as a function of π_0, the background rate in the unexposed. Note that scale is logarithmic.

Gerald van Belle (2008) Statistical Rules of Thumb

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.

Expert Trick 2: Interpreting fractional OR and when swapping the response events success/fail definition is helpful

Let's continue the previous lung cancer example where smoking's OR was 3 – which means smokers have 3 times the odds of getting lung cancer than non-smokers.

If we changed the response reference category from having cancer to **not having cancer** than it makes sense for smoking's OR to now be the reciprocal of what it was before i.e. 1/3=0.33 - since this now means that smoker's have 1/3 = 33% the odds of **not** having lung cancer than non-smokers.

This is the same result, just expressed differently. Which makes sense since our conclusions shouldn't differ based on the arbitrary decision on what to make the reference response category. Mathematically it shouldn't affect the results.

The same thing occurs if one swaps the predictors event definitions around (but for slightly different reasons).

This is a handy trick to know since fractional odds ratios i.e. OR < 1, can be hard to interpret and communicate. So if you have a lot of hard to communicate OR < 1 just swap how you have defined the responses 'success' event and now they will all be greater than 1! (with the OR > 1 now less than 1).

You can also swap the response event, or predictor events, to make the interpretation easier. For example, double negatives when both are negative can be hard to interpret.

Interpreting fractional OR and when swapping the response events success/fail definition is helpful

Mathematically this happens for the response because the odds of an event happening is the reciprocal of the odds that it didn't happen i.e. if the odds(event A happening) = X than the odds(event A not happening) = 1/X.

For example, if we have a logistic regression where we define event A as the success, and this results in a predictors OR being 1/3=0.33

Then if we ***swap response events and make event A the failure*** each odds within the odds ratio is inverted within themselves making the OR it's reciprocal which is 1/0.33 = 3

On the other hand, if we swap the predictors then we are swapping the numerator and denominator odds in the odds ratio.

Expert Trick 3) Consider reporting absolute measures such as probabilities/risks, odds and % of sample too

ORs compare the ***relative*** change in the odds, but ignore the underlying ***absolute*** chance of an event happening. It's important to know both.

Example 1: Low absolute chance of event. Say eating avocados increases the odds of a rare cancer by 10 i.e. OR=10. ***But*** the baseline odds of getting the rare cancer is 0.00000000000000000000000000001. Increasing the odds by 10 has little practical impact on the chance of getting cancer, so keep eating avocados! (Especially considering their other health benefits).

Example 2: Different ORs can have very different baseline chances. 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 affected (I wonder why??).
- **However.** Far more people in Melbourne were affected 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 relative 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)

When reporting absolute measures such as probabilities/risks, % of sample and metrics based on them like RR can't be used, but odds and odds ratios are still OK

Before reporting absolute metrics such as probabilities/risks , sample %'s or metrics derived from them like relative risk we first need to decide if they are appropriate and useful metrics.

They ***may*** be useful if the study is an accurate representation of the overall population e.g. cross sectional studies.

They are ***not*** useful if the study is not an accurate representation of the overall population. In such cases odds and the odds ratio are still relevant, which is why logistic regression often focuses on odds ratios, since it's always applicable. For example:

- **Case-Control Studies:** are when we have a sample of cases e.g. a rare disease, and then collect a fixed number of controls e.g. those without the disease, to understand what the differences between the groups are and hence the risk factors for the disease. The # of controls collected is often fixed at 5 times the cases as this is optimal for minimising parameter standard errors. However, this means we can't estimate the chance of the disease since it's an artifact of the sampling (1/(1+5) = 1/6=0.17%) and not an accurate picture of its prevalence in the wider population. Meaning risks and relative risks can't be calculated, but odds and odds ratios can since they simply compare the difference between the cases and controls.

When 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

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).

EDA for interactions with 2 continuous variables

Interactions with 2 continuous variables are not straight forward to fit. There are a number of complications one needs to consider.

Just one is what type of surface is a suitable fit, for example is a plane suitable (i.e. a sheet of paper), or are there nonlinear relationships that need to also be fit e.g. maybe its more concave?

The first step in assessing this is (as always) EDA, and for a continuous response a suitable data visualization is either a 3D scatterplot plot, or some variant such as a contour plot or heatmap.

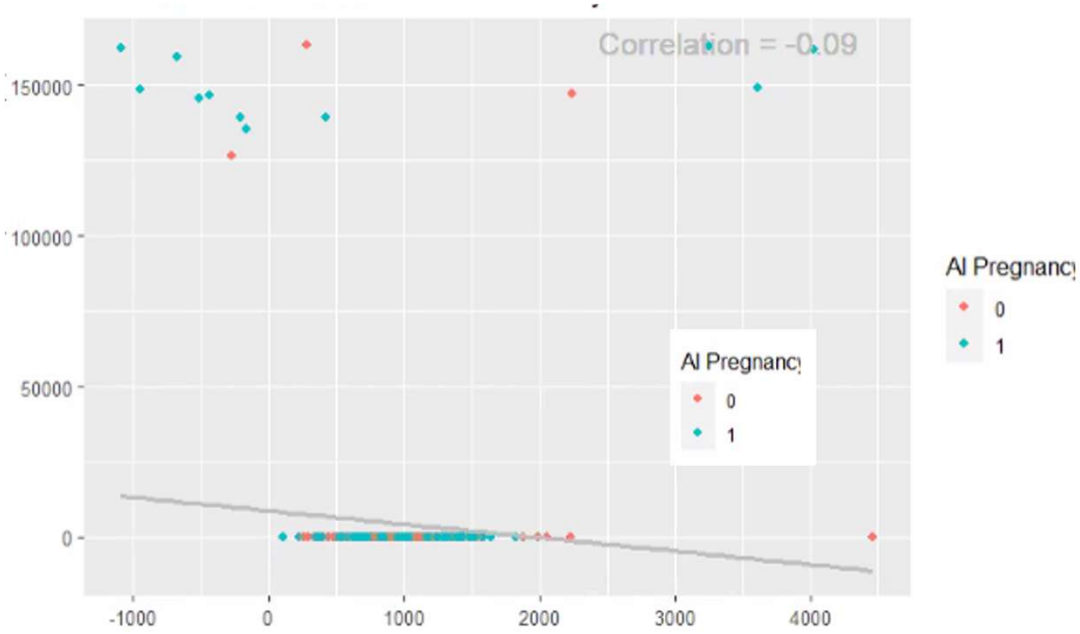
However, this doesn't work as well with a binary response.

EDA for interactions with 2 continuous variables

An alternative is to a 2D plot, with the response colour coded. Below is a real-world example. This came to us in a consult, with the problem being the model would not converge. So as usual we started diagnosing the problem with EDA, which showed us that:

- Although the horizontal x axes continuous predictor is continuous there is no strong pattern with neither the red (failure of Artificial Insemination) or blue (success) symbols being more to the left or right. Making it hard to fit a sigmoid curve and hence logistic regression.

- The more likely reason for the convergence problem though is the vertical axis. The large gap in the middle with no data makes it hard to fit a continuous interaction as a surface. This variable is actually more binary i.e. 0 vs very high.



Poisson (count) Regression

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

Workflow Suitable for:

- Positive Integers
- Counts
- Rates
- Some Log Normal data
- Before After Control Impact design (BACI)

Poisson (count) Regression

Uses the Poisson distribution which assumes the data is from the set of Natural Numbers i.e. the non-negative integers 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** variable to the model. This variable divides the count by something to turn it into a rate. For example:

- 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 i.e. cell concentration = $\frac{cell\ count}{volume}$.
- 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.

Changes to dingo diet caused by human interaction, and its implications on conservation.

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.***

Newsome T, Chris H, Wirsing A (2020) Restriction of anthropogenic foods alters a top predator diet and intraspecific interactions

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

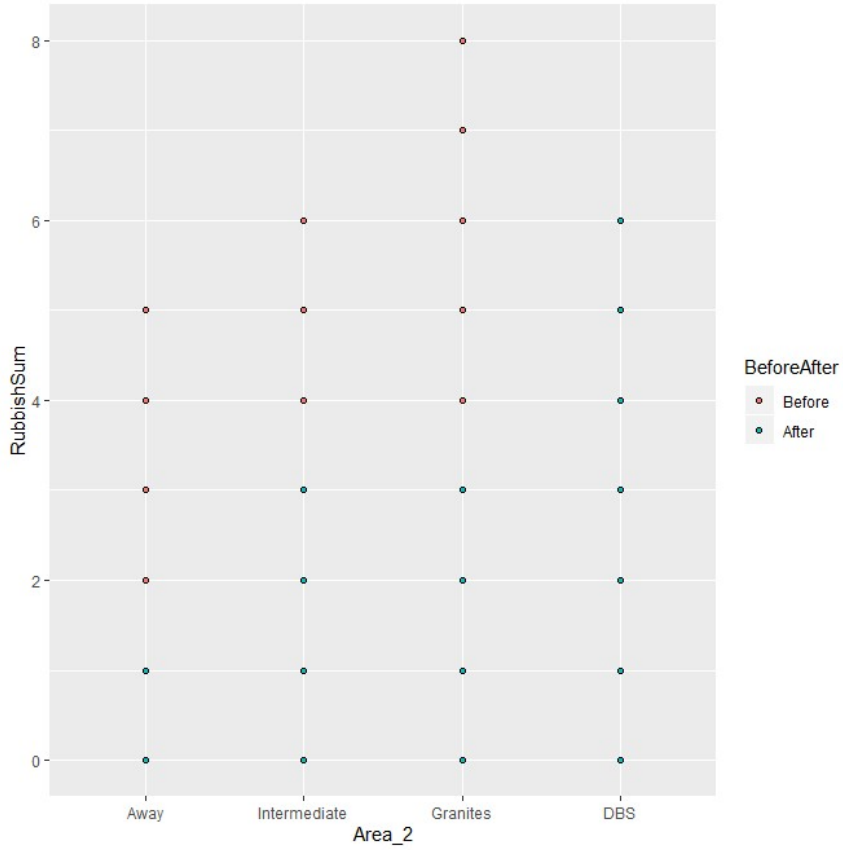
Step 6) Reporting

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 overlayed on each other.

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

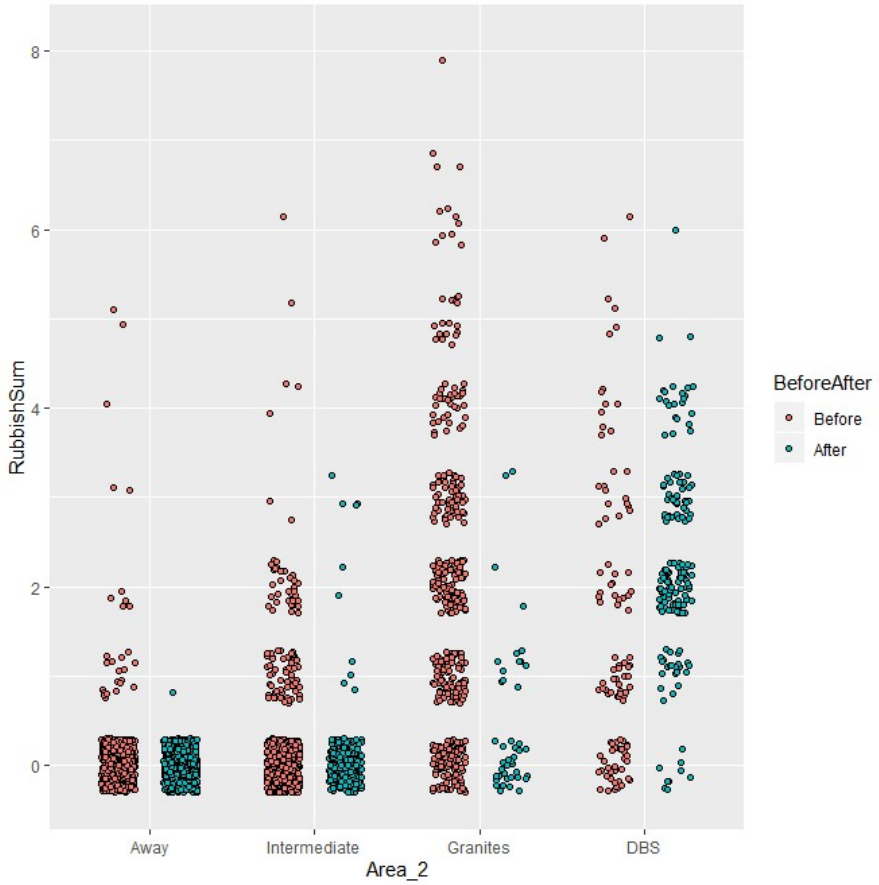


Step 1) Pick a suitable model to fit to the data via Exploratory Data Analysis (EDA)

To fix this I add a jitter - to the **plot only**, **not the data we model.**

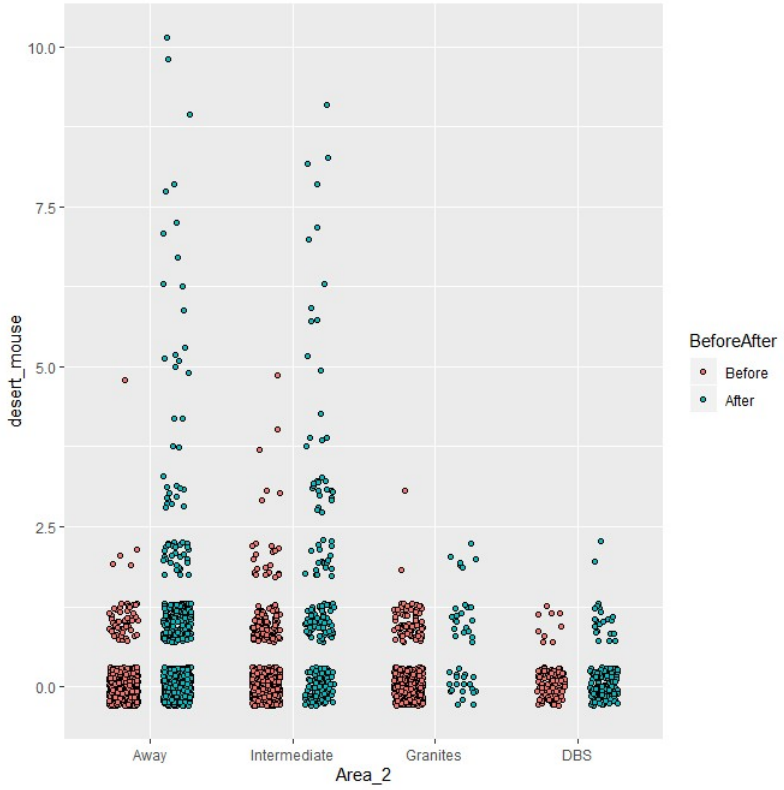
Now I can see that the number of scats with rubbish in them has dropped after the fences were installed. Except at DBS for some reason?

The reason was that they broke through the fence wasn't, so they all went over there!!



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



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

Poisson GLM might be a good fit, so let's try that meaning:

Y_i ~ Poisson(λ)
~ mean=variance= λ

We link the linear predictor ($X\beta$) to λ using the log link i.e. $\log(\lambda)=X\beta$ since that is the conventional model. (NB: this makes a multiplicative model when we back transform to rates).

Step 2) Fit the Model

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

Linear Predictor is	***desert_mouse~Area_2*BeforeAfter***
Data Distribution is	***family=poisson***(link="log")
Link Function is	family=poisson(***link="log"***)

Step 3) Check Model Assumptions via Diagnostics: Zero Inflation

Sometimes we get count data with far too many zeros 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 an older way around this (called 2 step/stage or hurdle models). These fit a binomial (logistic) model to whether the event occurs, and then a Poisson if it does. The modern approach is to use Zero Inflated Poisson (ZIP) and Zero Inflated Negative Binomial (ZINB) models that effectively combine these 2 models into a single model fit.

Step 3) Check Model Assumptions via Diagnostics: Zero Inflation

A rough test for this is to simulate the number of zeros 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

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 http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#overdispersion. There is ongoing research on this topic so more recent information and solutions may be available here.

This function tests whether the dispersion parameter is different to 1, which is what a Poisson distribution assumes. It tells us that although there is statistically significant overdispersion it is not very large at only 1.6, so not worth worrying about. What is considered too large is domain specific and subject to ongoing research, I have seen cutoffs from 1.10 – 5 used.

Common ways to deal with this are:

1. **Distributional Regression**. 2 distributions are commonly used:
    1. **Negative Binomial** distribution - fits a more suitable distribution with an extra dispersion parameter, there are a variety of R packages (including gamlss.dist) that fit this model and is usually available in other software such as SPSS. Very commonly used.
    2. **Generalised Poisson** distribution - fit in R using the gamlss.dist package and the GPO distribution, harder to fit in other software.
2. Fit an **individual level random effect using a GLMM** (this tricks the model into adding an extra variance parameter).
3. **Quasi-Poisson** can also be used. Given the above alternatives there is some debate on how useful it is due to the difficultly in applying inferential methods such as likelihood ratio test, AIC, etc. http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html

```
> is_overdispersed(desert_mouse.p1) #
       chisq        ratio          rdf            p
4.467379e+03 1.557663e+00 2.868000e+03 9.576844e-74
```

Step 3) Check Model Assumptions via Diagnostics: Residuals

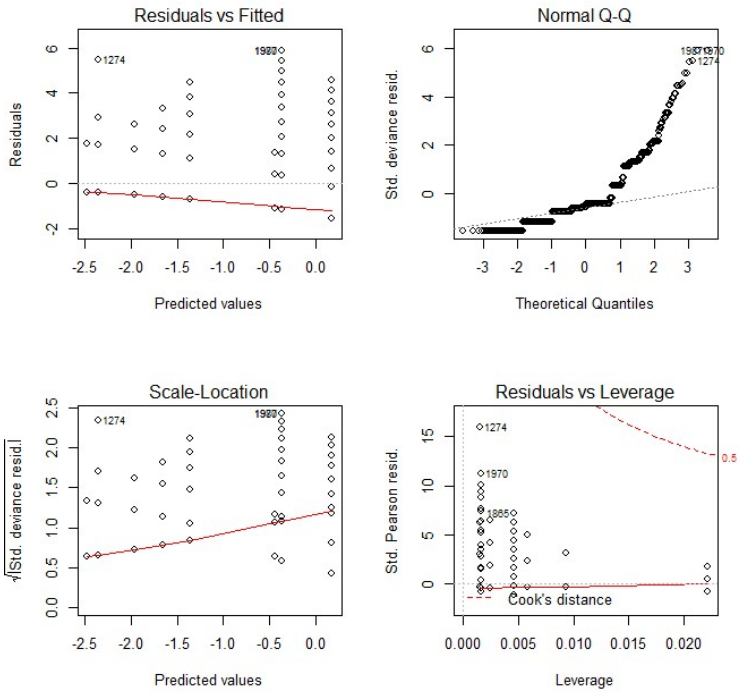
No obvious influential outliers

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!

Dharma residuals are more useful and are in the R workflow which can be downloaded from our online library.



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) – since p values are so small.

```
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
```

"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 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.

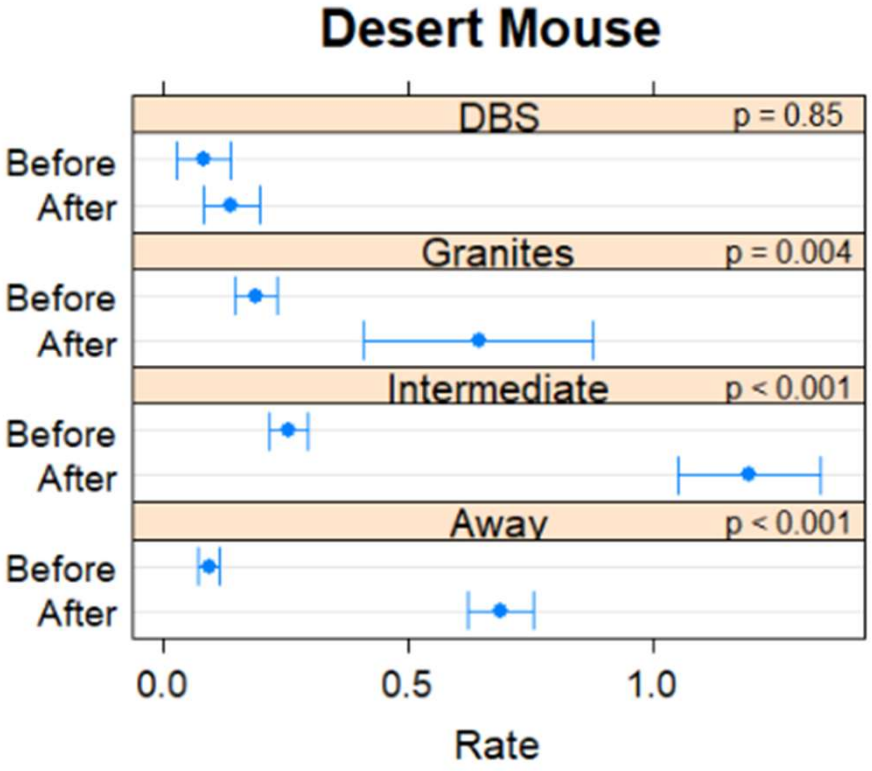
The model fit was a GLM with Poisson distribution and log link function. There was no evidence of over dispersion or zero inflation."

But as it's a complex design with a lot going on we will use a plot to report the patterns and effects.

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.

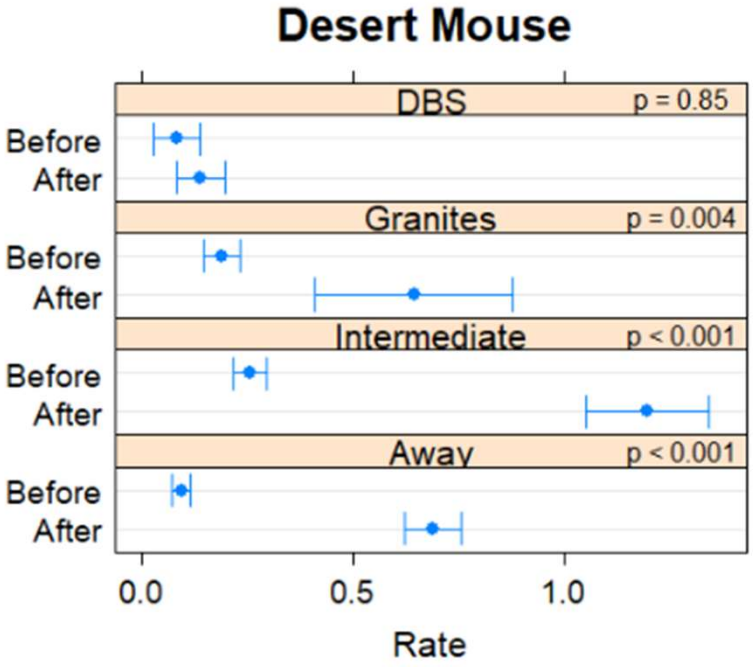


Step 6) Reporting: Overall Conclusion suitable for publication

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 ($p<0.001$).

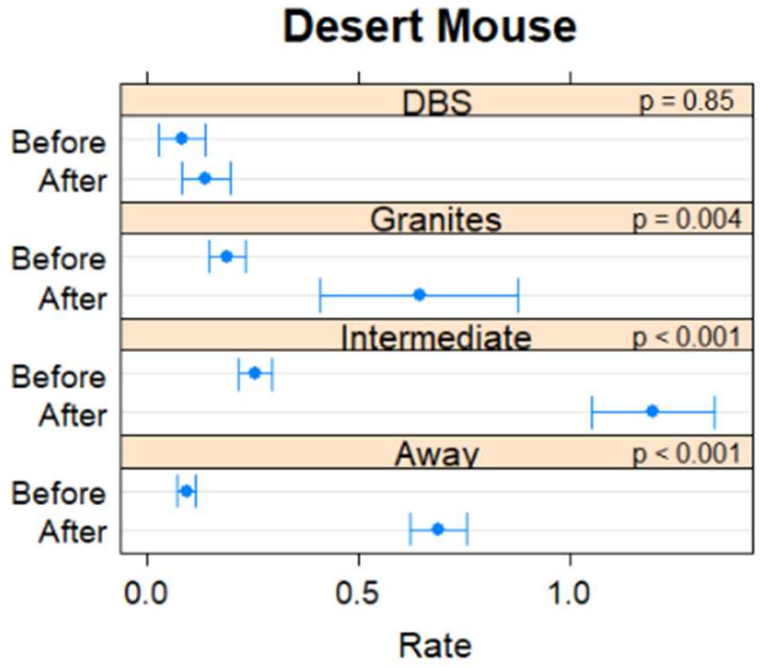


Step 6) Reporting: Overall Conclusion suitable for publication

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.



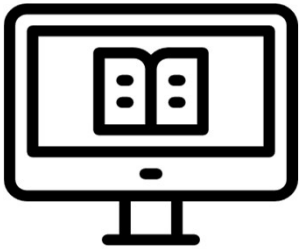
One Take Home?

Would people mind putting it in the chat.

Anyone here like to say what theirs is?

Other Resources

Other resources



Videos

- StatsQuest with Josh Starmer
  - Linear Models
  - What is a Statistical Model
- Zedstatistics, longer videos than StatsQuest.

Websites

- R GLMM FAQ, FAQ by Ben Bolker maintainer of 1 of the 2 main R packages. But good for all GLMM questions

Books

- Faraway, Julian James. (2016) Extending the Linear Model with R : Generalized Linear, Mixed Effects and Nonparametric Regression Models.
- Fox, John. (2016) Applied Regression Analysis and Generalized Linear Models.

Further Assistance

LOGISTIC REGRESSION

- *Harris Jenine K (2021) Primer on binary logistic regression. Family Medicine and Community Health.* ***Read this and use our workflow and you can't go wrong.***
- Meyers, L., Gamst, G., & Guarino, A. (2017). binary and multinomial logistic regression and roc analysis. In Binary and Multinomial Logistic Regression and ROC Analysis (Third ed., Vol. 0, pp. -). SAGE Publications, Inc. ***A good description of logistic regression. And the Receiving Operator Characteristic (ROC) Curve – a common way to assess the models predictive power when used as a classifier and to select predictor and/or probability cutoffs for classifying as a success. Note that the SPSS directions may be slightly out of date.***
  - *The next chapter has an acceptable intro on how to do it in SPSS, but is a bit out of date. Binary and Multinomial Logistic Regression and ROC Analysis Using IBM SPSS*

Further assistance at The University of Sydney

SIH

- Statistical Resources website: containing our workshop slides and our favourite external resources (including links for learning R and SPSS).
- Hacky Hour: an informal monthly meetup for getting help with coding or using statistics software.
- 1on1 Consults can be requested on our website or here (click on the big red ‘contact us’ link).
- Online library. Useful links and the most recent version of all our workshops.

SIH Workshops

- Create your own custom programs tailored to your research needs by attending more of our Statistical Consulting workshops. Look for the statistics workshops on our training page or on our Training calendar.
- Sign up to our mailing list to be notified of upcoming training.

Other

- LinkedIn Learning

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: Shows how to build interpretable models and analyse data to extract insightful & impactful patterns which enable you to make the impactful discoveries that expand our knowledge, and how to craft engaging research stories to communicate those discoveries.

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.

Linear Models 3: How to build interpretable models and analyse data to extract insightful & impactful patterns, and craft an engaging research story

Statistical analysis is more than just building the best predictive model, it should also enable you to make impactful discoveries that expand our knowledge. Constructing engaging narratives about your research is also invaluable as you look to connect with your field, the community and funding bodies. To do this you need to build interpretable models, test hypotheses, uncover insightful & impactful patterns, and present results in insightful, intuitive and memorable ways. In this workshop we explore tips and tricks to make your research do just that. Topics covered will be:

- ***Building impactful real-world recommendations and guideline****s* – i) why we need to understand both stated and model derived importance, ii) how Quadrant Analysis uses both variable performance and importance to develop impactful real-world recommendations and guidelines.
- ***Reporting tricks that extract insightful & impactful patterns and craft engaging stories*** – i) establishing the importance of a predictor/risk factor, ii) confidence vs prediction intervals, iii) applying and correcting for multiple comparisons, iv) testing different hypothesis using different model parameterisations of the design matrix, v) interpreting categorical predictors - dummy vs effects coding and estimated marginal means, plus other reporting and interpretation tricks.
- ***Building interpretable models*** – it's quite common for researchers to incorrectly use model parameters to establish variables 'impact' or 'importance' . We show how multi-collinearity prevents this interpretation, and how to assess and then fix it so parameters can be used to identify important predictor/risk factors and other insightful patterns.
- ***Mixed models*** – extend the Linear Model 1 intro to: i) better explain how mixed models work, ii) use them to test population wide hypotheses outside your sampled groups, and iii) use a random slope (with examples of the patterns it can explain and hypotheses it can test).
- ***Using data visualisation to report complex nonlinear models graphically and aid pattern extraction***

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 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
    - **Kabacoff, Robert (2015) R in Action: Data Analysis and Graphics with R**. 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. (Wickham, Hadley, and Garrett Grolemund (2017) R for Data Science Import, Tidy, Transform, Visualize, and Model Data)
- Finally I recommend **MASS (Modern Applied Statistics with S-Plus)** 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. (Venables, W. N, and B. D Ripley (2013) Modern Applied Statistics with S-Plus)

ONLINE

- Lots of short (and long) YouTube courses
    - EXPLORE, find a style you like and watch a little each day if too long.

Further R resources

- There is a large online community of R users contributing to free ‘packages’ with data analysis functions, which leads to many ways of coding your analysis in R. This can be confusing. We recommend using tidyverse packages and tidy-centric code.
- See our SIH helpful links for guides on using R and RStudio.
- LinkedIn Learning: R courses
    - Including Learning the R Tidyverse (2024), Complete Guide to R: Wrangling, Visualizing, and Modelling Data, and Cleaning Bad Data in R.
- RLadiesSydney: RYouWithMe
- Lme4 issues answers many questions about applying LME4.

A reminder: Acknowledging SIH

- All University of Sydney resources are available to 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 for use of workshops and workflows:

- *"The authors acknowledge the Statistical workshops and workflows provided by the Sydney Informatics Hub, a Core Research Facility of the University of Sydney."*

We value your feedback

- We want to hear about you and whether this workshop has helped you in your research. What worked and what didn't work.

- We actively use the feedback to improve our workshops.

- Completing this survey really does help us and we would appreciate your help! It only takes a few minutes to complete (promise!)

- You will receive a link to the anonymous survey by email.


Have
a
Break