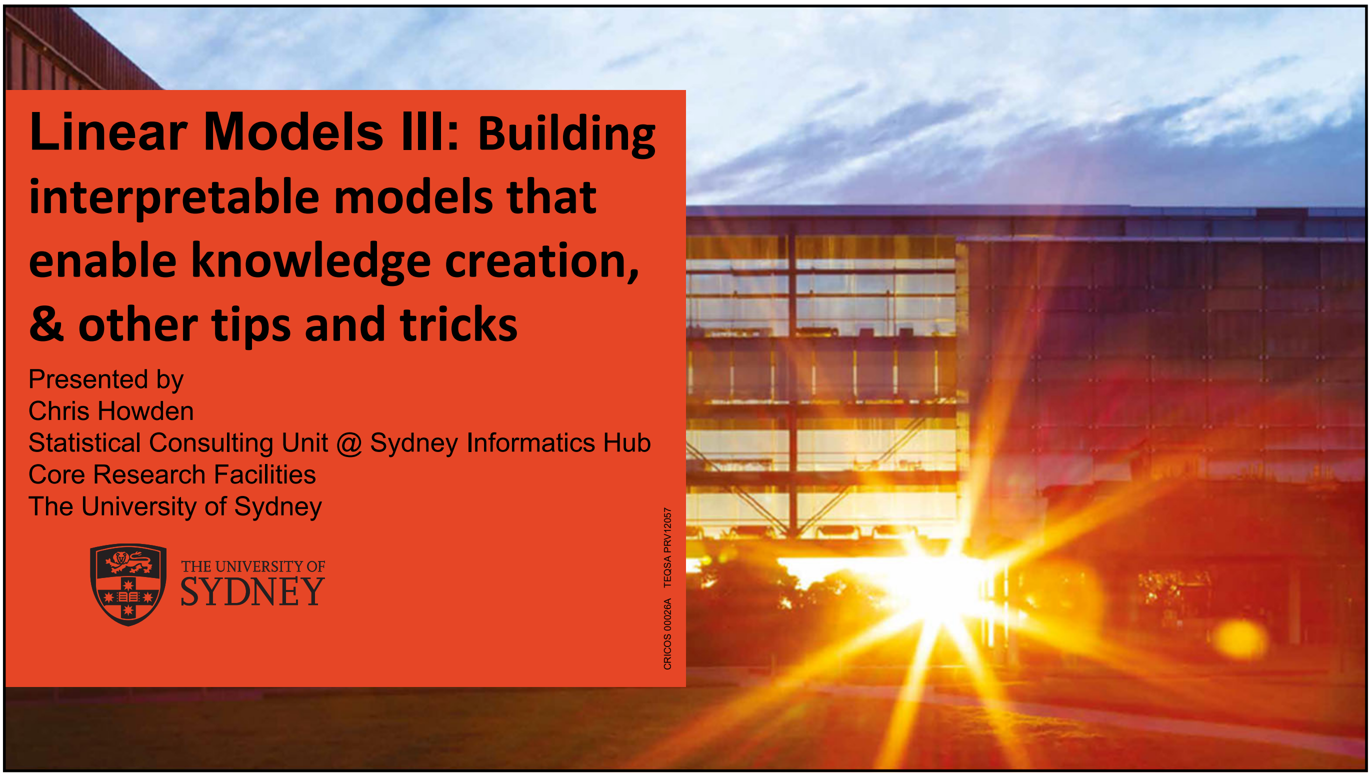
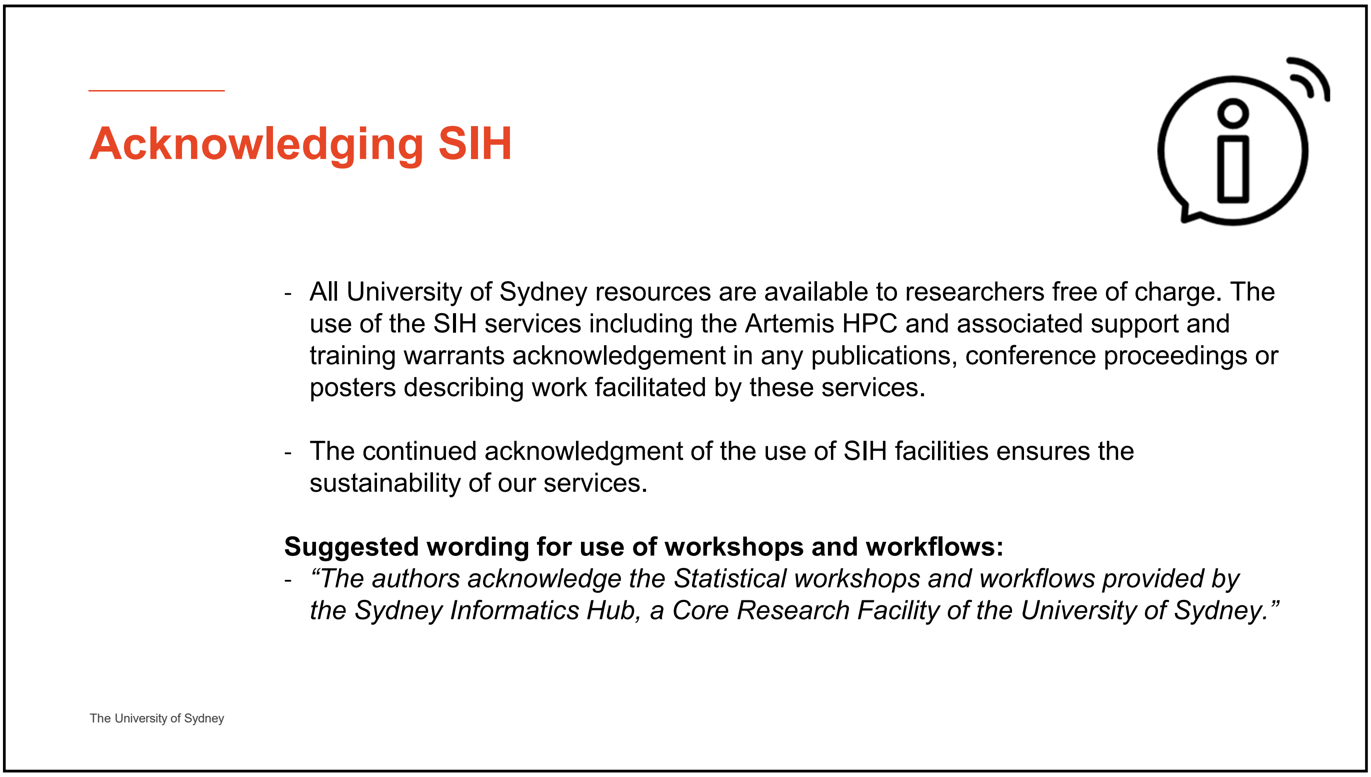


Linear Models III: Building interpretable models that enable knowledge creation, & other tips and tricks

Presented by
Chris Howden
Statistical Consulting Unit @ Sydney Informatics Hub
Core Research Facilities
The University of Sydney

1



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.”*

The University of Sydney

2

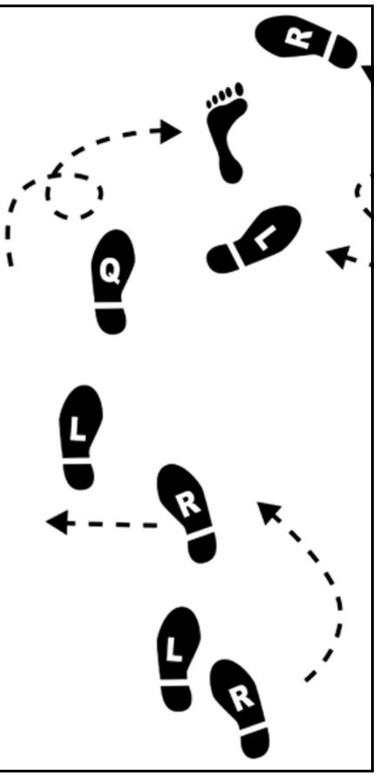
What is a workflow?

- 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).

3

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.



4

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.

5

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

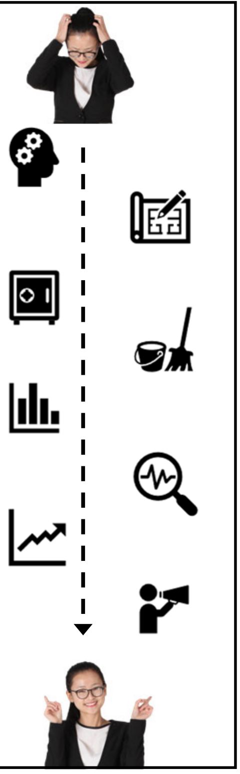
6

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**



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
- Parametrising the model
- Mixed Models

Step 3) Check Model Assumptions via Diagnostics: Residual Analysis

Step 4) Goodness of Fit: Plots and Statistics

Step 5) Interpret and Report Model Parameters to reach a conclusion and build Knowledge
- Estimated Marginal Means vs Parameter contrasts, Confidence and Prediction Intervals, Multiple Comparisons

Step 6) Reporting

Linear Models 1 and 2 and Model Building Workshops have more detail on many of these steps.

A Conversation is better than a Presentation

So please speak up and ask questions!

People think differently. So I may need to explain things in 2 or 3 different ways!

Experimental Design Tips

11

What are Linear Models?

Linear Regression

ANOVA

ANCOVA

Logistic (binary) regression

Count (Posson) regression

Before After Control Impact (BACI) Studies

Randomised Control Trials (RCT's)

Repeated measures

Plus Many More!!

12

Predictors don’t need to be normally distributed

Remember, it’s the model error we assume to be normally distributed. Not the response or the predictors.

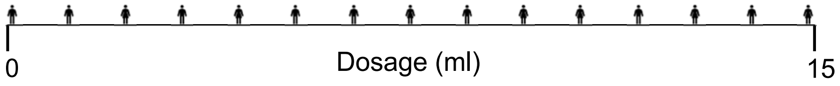
It’s ***usually better if the predictors aren’t normally distributed.***

Some common design methods follow.

Equally/Uniformly spaced between the minimum and maximum

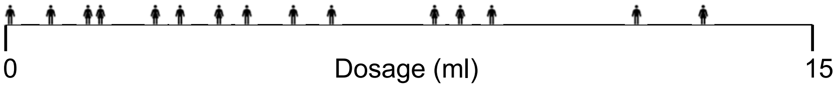
This might seem the best way since it seems to capture the data efficiently e.g. i) over a 6 month period collect data equally from the start to the end at T=0,2,4,6 months; or ii) administer medicine from the lowest (0ml) to the highest high (15ml) dosages at 0,3,6,9,12,15ml. *A*lthough commonly used there are problems such as:

1. **There may be structure between the points which is missed and/or it may introduce bias** e.g. the change might occur at month 1 or at 10-11ml. This is of particular concern if the data is spatial or temporal e.g. if researching public transport always sampling during the week would not give an accurate picture of weekend usage.
2. **Observational studies often find it hard to collect data like this.**, as they need to take what they can get.
3. **Lacks randomisation**, which removes bias by balancing out unknown confounders (refer to our experiential design workshop if this isn’t well understand as it’s an important part of causal analysis).



Randomly spaced between the minimum and maximum

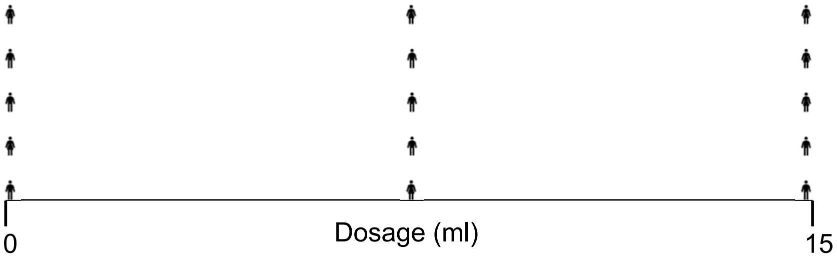
Can be better than uniformly spaced as it also efficiently samples the space, and as random avoids bias due to unknown structures and gives a well-structured variance. However, it may miss focal points or lead to clusters of points which is not ideal e.g. i) for each patient 3 random samples over a 6 month period might miss the focal before and end of treatment points; or ii) a new medicine using a random dosage between 0-15ml might randomly end up with more data at the low end.



15

Equally spaced categories between the minimum and maximum

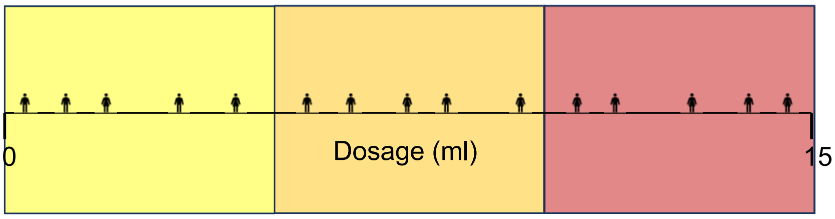
A type of ***categorical stratification*** that collects data at set points e.g. i) over a 6 month period collect data at the start, end and midpoint; or ii) administer medicine at fixed low (1ml), medium (8ml) & high (15ml) dosages. *A*lthough common not always ideal as structure of interest may sit between the equally spaced points e.g. the change might occur at 2 months or at 10-12ml.



16

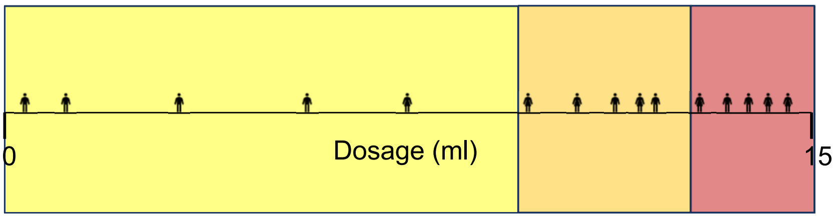
Equally spaced intervals/bins that are randomly sampled

This combines the above two methods and avoids the problems of both. We first create equally spaced intervals/bins (rather than points) along the predictors range and then randomly sample within those bins. This ensures each bin has the same # of points, reduces clustering and introduces some randomness within each bin so we don't accidently miss patterns that don't align with the regularly spaced points e.g. i) randomly sample within the months 1-2, 3-4 & 5-6; or ii) within 3 dosages defined as low (0-5ml), medium (>5-10ml) and high (>10-15ml). May still miss areas of focus though.



Intervals/Bins designed to focus on areas of interest that are randomly sampled (if possible)

Combines all of the above 3 to avoid their problems e.g. i) for the 6 month time period we may need the 0 and 6 month times and then randomly sample between the rest; or ii) for dosage the effect might not be linear and if we expected the impact to kick in within the 10-12 range we'd want more data in this area to model the turning point, so we could define low as 0-9ml, medium as >9-13ml and high as >13-15ml. Note that we would usually also have a control with 0ml dosage, and might also want a treatment with the max dosage of 15ml.



Predictors don’t need to be normally distributed

Other considerations

- **Equally Spaced Categories usually gives categorical data**, while the others continuous which often gives more interesting information e.g. one can fit a curve to continuous data rather than simply compare categories.
- **Continuous data can be binned into categorical data if an ANOVA style method** is preferred. But it’s much harder/impossible to turn categorical data into continuous data. Meaning continuous data gives us more options.
- When sampling in a continuous fashion we **need enough sample for the range to be well sampled**, if not then it may be better to sample within specific categories such as min, average/midpoint, max.
- **What is the data your analytical method requires** e.g. formal timeseries methods assume equally spaced data, ANOVA requires categorical data, curve fitting continuous, etc.
- **Random sampling allows for much stronger causal inference**, since it removes bias. (refer to our experiential design workshop if this isn’t well understand as it’s an important part of causal analysis).

More on Experimental Design and Sample Size

Experimental Design and Research Essentials Workshops

- Far too many researchers think they know all they need to in this area, when they don’t. We commonly see designs that could be **substantially improved for stronger causal inference and results -** leading to **publication in higher impact journals** (amongst other benefits).
- **If you don’t thoroughly understand the things I have been talking about then you could benefit from these workshops** e.g. randomisation leads to stronger causal inference, the same data but different ED has different causal inference, what is causal inference!!
- Even if you have already collected your data it is well worth attending since it may improve your write up and analysis e.g. we had a client who didn’t realise they had a Before/After Control/Impact (BACI) design which allowed them to make **much stronger causal inferences** than the simple observational study they thought they had.

Sample and Power Workshop

- Shows the steps and decisions researchers need to make when designing experiments to **ensure sufficient sample** e.g. power, minimum required to fit the necessary model, stability, etc.
- And **how much power the study has** i.e. does it have sufficient power to detect the effects you expect to see, or is your study a complete waste of time and resources.

21

Building Interpretable Models

How statistics uses causal models to build knowledge, while predictive models do not

Case Study: Why identifying and accounting for multicollinearity is the key that unlocks interpretable models

Workflows for interpretable models, and accounting for multicollinearity

22

Our **Goal** is to **Build Models that answer our Research Questions and expand our Knowledge** by showing **Causality** or Correlation if causal inference is not possible (this is what statisticians focus on)

Not just build the best predictive model

(which is what machine learning methods usually focus on)

23

Very different processes are used for those 2 goals

If all you want is the ‘best predictive model’ then model building is rather straightforward.

1. Pick a fit metric and method to maximise it, usually penalised for complexity e.g. cross validation on the correct answer
2. Try out all models with lots of different variable combinations to find the best fit
3. Maybe do some model averaging at the end

24

The problem with these methods is that their models are rarely interpretable

For a few reasons:

1) The **model and model parameters may not be easily extracted e.g. Neural Networks.**
2) Even if the model gives model parameters they often **can't be easily interpreted due to multicollinearity**. Which they usually sweep under the carpet and ignore, rather than explicitly deal with.
3) The modelling process and models created **don't test specific research questions and scientific hypotheses**.

A statistical workflow for answering specific research questions is covered in our Model Building Workshop. It also covers ways to handle multicollinearity as does our Multivariate Workshop.

25

Case Study: How multicollinearity effects model building, interpretation and reporting

Let's assume there is a segment of coffee drinkers who only care about 2 things: **coffee taste** and **sweetness**. And we give them some coffee with honey in it.

We measure the following variables on a 100 point continuous scale (which is not ideal as explained in Surveys 1: but makes this example easier to explain)

- Response
  - Overall Liking
- Predictors
  - Coffee Taste
  - Sugar (measures sweetness)
  - Honey (also measures sweetness)

Notice that we **don't measure the underlying sweetness latent variable directly, instead we use Sugar and Honey.**

26

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

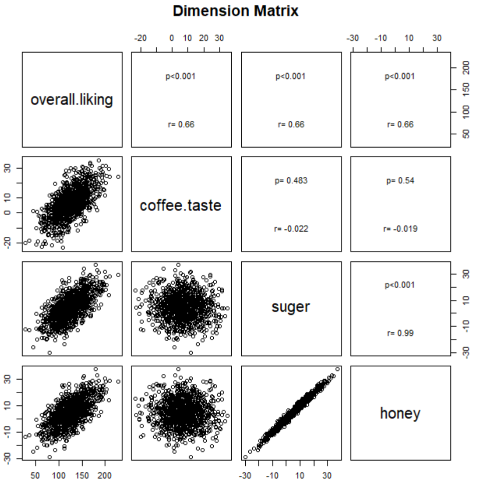
Step 6) Reporting

Linear Models 1 and 2 and Model Building Workshops have more detail on many of these steps.

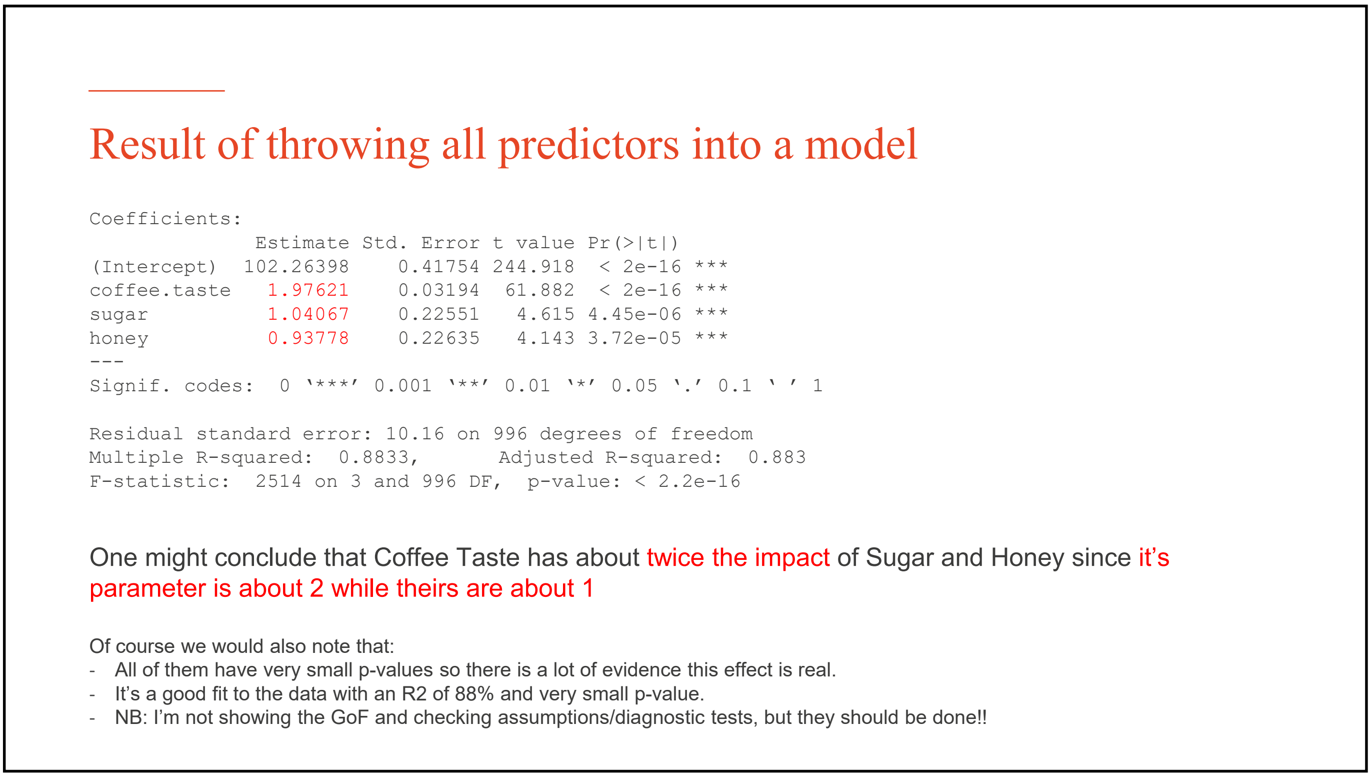
27

EDA (Exploratory Data Analysis) – We notice that

- Overall Liking is correlated to all of them.
- Coffee Taste and Sugar/Honey aren’t correlatec
- Sugar and Honey are strongly correlated.



28



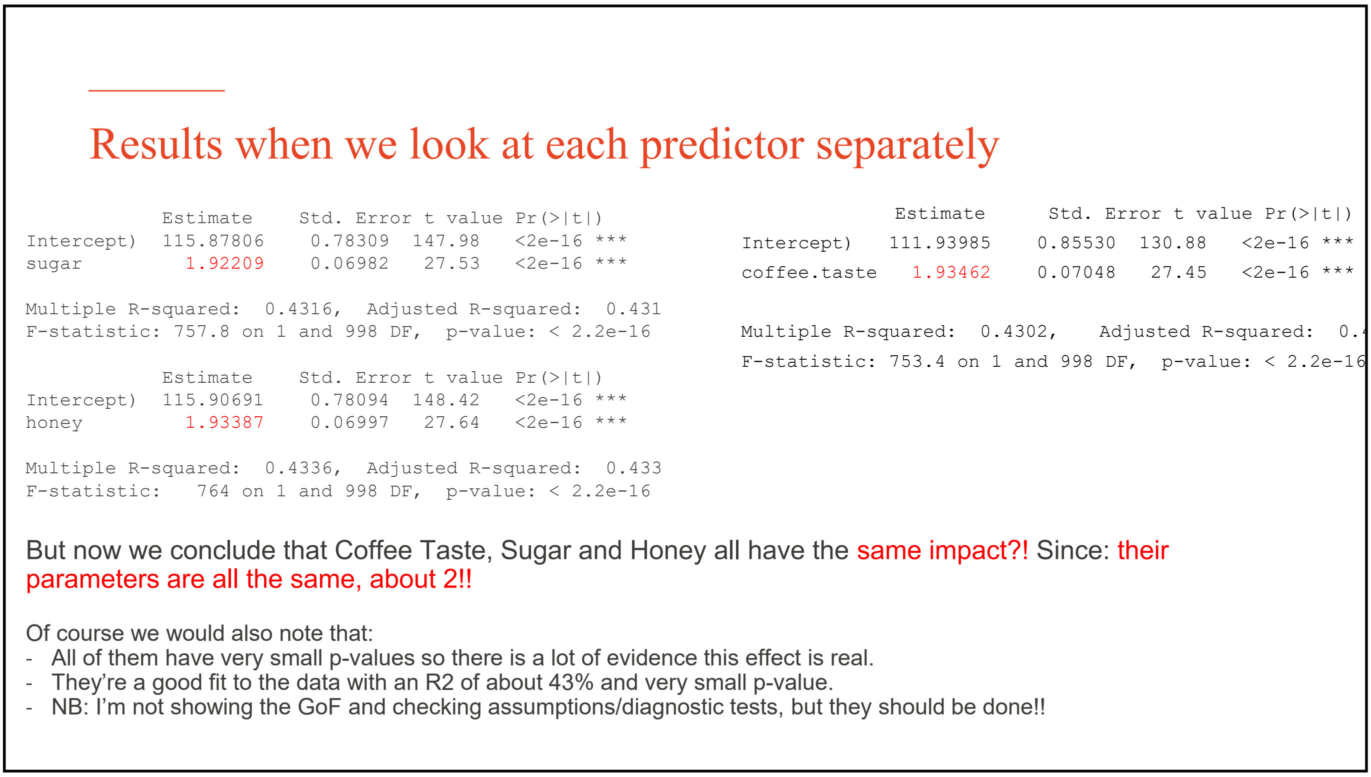
Result of throwing all predictors into a model

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 102.26398    0.41754 244.918  < 2e-16 ***
coffee.taste   1.97621    0.03194  61.882  < 2e-16 ***
sugar          1.04067    0.22551   4.615 4.45e-06 ***
honey          0.93778    0.22635   4.143 3.72e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.16 on 996 degrees of freedom
Multiple R-squared:  0.8833,	Adjusted R-squared:  0.883 
F-statistic:  2514 on 3 and 996 DF,  p-value: < 2.2e-16
```

One might conclude that Coffee Taste has about twice the impact of Sugar and Honey since it’s parameter is about 2 while theirs are about 1

Of course we would also note that:
- All of them have very small p-values so there is a lot of evidence this effect is real.
- It’s a good fit to the data with an R2 of 88% and very small p-value.
- NB: I’m not showing the GoF and checking assumptions/diagnostic tests, but they should be done!!

29



Results when we look at each predictor separately

```
            Estimate    Std. Error t value Pr(>|t|)
Intercept)  115.87806    0.78309  147.98   <2e-16 ***
sugar         1.92209    0.06982   27.53   <2e-16 ***

Multiple R-squared:  0.4316,  Adjusted R-squared:  0.431
F-statistic: 757.8 on 1 and 998 DF,  p-value: < 2.2e-16

            Estimate    Std. Error t value Pr(>|t|)
Intercept)  115.90691    0.78094  148.42   <2e-16 ***
honey         1.93387    0.06997   27.64   <2e-16 ***

Multiple R-squared:  0.4336,  Adjusted R-squared:  0.433
F-statistic:   764 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
                 Estimate     Std. Error t value Pr(>|t|)
Intercept)      111.93985     0.85530   130.88   <2e-16 ***
coffee.taste      1.93462     0.07048    27.45   <2e-16 ***

Multiple R-squared:  0.4302,    Adjusted R-squared:  0.4
F-statistic: 753.4 on 1 and 998 DF,  p-value: < 2.2e-16
```

But now we conclude that Coffee Taste, Sugar and Honey all have the same impact?! Since: their parameters are all the same, about 2!!

Of course we would also note that:
- All of them have very small p-values so there is a lot of evidence this effect is real.
- They’re a good fit to the data with an R2 of about 43% and very small p-value.
- NB: I’m not showing the GoF and checking assumptions/diagnostic tests, but they should be done!!

30

So which model is right? What’s happened?

Because we simulated the data we know that the underlying Sweetness and Coffee Taste dimensions have the same impact, which in both cases is a gradient (slope) of 2.

So the information we want to get from this analysis is that:
1. There is an underlying Sweetness dimension that we are capturing twice. Once with Honey and once with Sugar.
2. That Sweetness and Coffee Taste have the same impact on Overall Liking.

The **statistical workflow we just used tells us this** by using:
- EDA (Exploratory Data Analysis) to show that Sugar and Honey are highly correlated and measuring the same underlying variable. A bit of thought would suggest this is sweetness, and that if we want to understand the unique effect of this we should have only 1 of them in a model. So we need to decide which of them to use as a ***proxy*** for sweetness.
- Individual models show that the marginal/independent effect of each of them is about the same. Which is the ***correct interpretation if we want knowledge.***
- We might also look at the models with Coffee Taste and Honey or Sugar to understand their combined effect. This confirms that they have the same effect as Coffee Taste, and tells us they are operating independently.

```
             Estimate      Std. Error t value  Pr(>|t|)
(Intercept)  102.26709     0.42091   242.96    <2e-16 ***
coffee.taste   1.97865     0.03219    61.47    <2e-16 ***
sugar          1.96569     0.03193    61.57    <2e-16 ***
```

```
             Estimate   Std. Error t value  Pr(>|t|)
(Intercept)  102.36156    0.42123   243.01    <2e-16 ***
coffee.taste   1.97306    0.03225    61.18    <2e-16 ***
honey          1.97199    0.03211    61.41    <2e-16 ***
```

31

So which model is right? What’s happened?

Simply going for a model fit with all 3 variables needs to ‘share’ the effect of sweetness between Sugar and Honey, which is why their parameters are halved and suggest that Honey and Sugar have halve the effect of Coffee Taste.

This highlights the problem with interpreting models with more than 1 predictor. They ***need to be interpreted in the context (at the same time) as all the others***, ***which is very difficult when there is multicollinearity and more than 2 or 3 predictors.***

Which is why it is impossible for anyone to interpret machine learning “best fit” model parameters independently in the presence of multicollinearity. One shouldn’t even try unless multicollinearity has been assessed, and ideally found to be negligible.

32

Reproducibility Crisis: Instrument and Design Bias

Imagine for a moment that we have 2 researchers, both doing the preceding experiment. But they use the below 2 different designs and instruments, which differ on how they measure the sweetness latent variable:

- Researcher 1: captures both Honey and Sugar
- Researcher 2: captures only Honey

If they used our statistical workflow both researchers would come to the same conclusion i.e. there is a sweetness latent variable which has about the same impact as Coffee Taste.

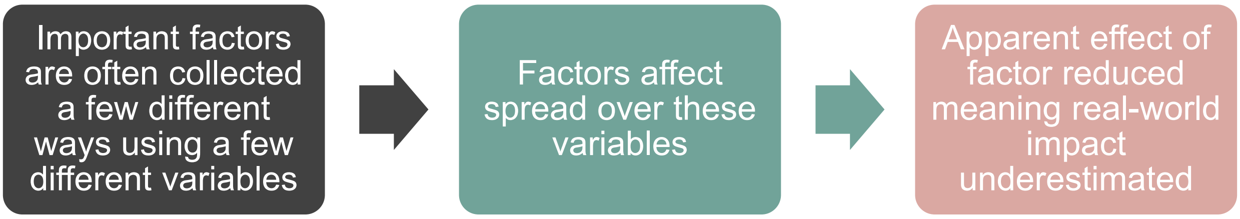
But if they simply fitted a machine learning ‘best model’ they would disagree i.e.

- Researcher 1 would incorrectly conclude that Honey and Sugar have half the impact of Coffee Taste
- Researcher 2 would correctly conclude that Honey has the same impact as Coffee Taste

I feel this is one of the biggest problems and mistakes researchers make.

  - And is 1 reason for the Reproducibility Crisis.

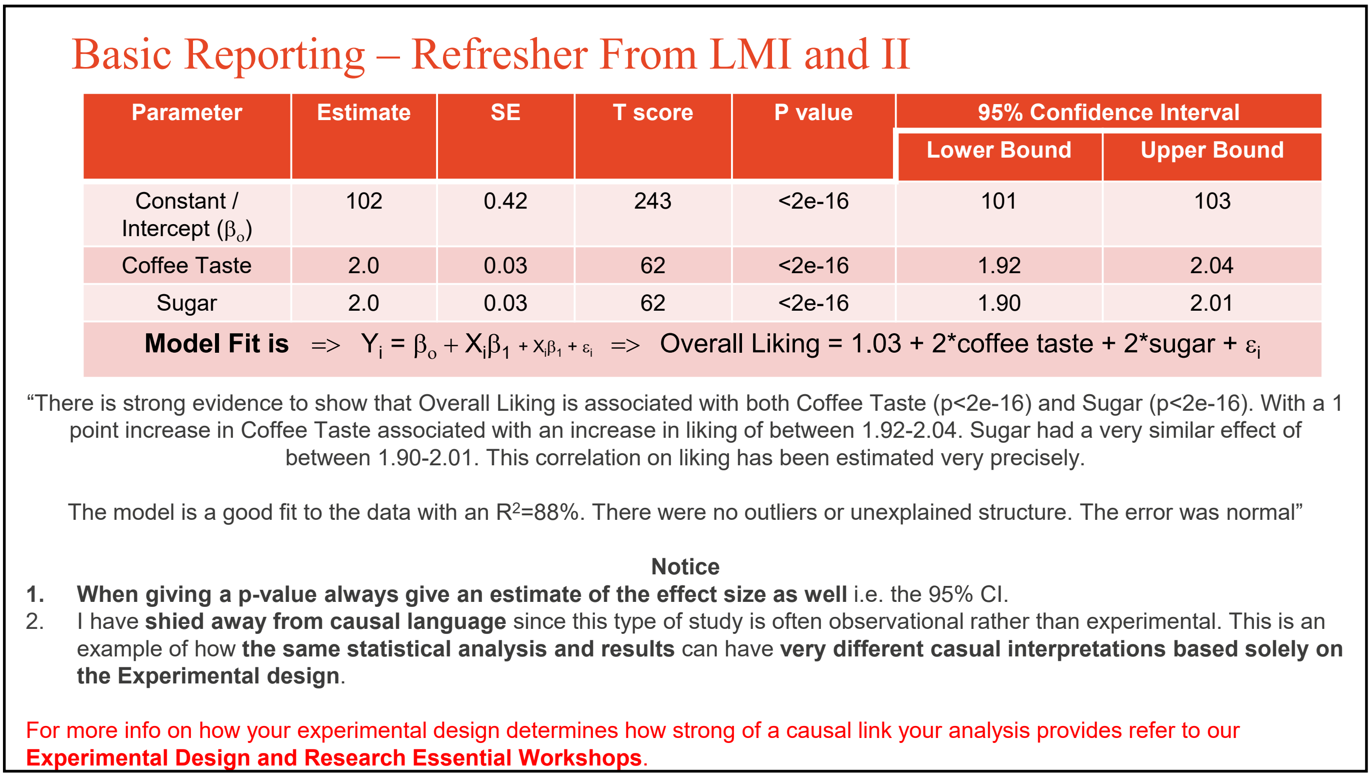
Important factors can be biased down



Important factors are often collected a few different ways using a few different variables

Factors affect spread over these variables

Apparent effect of factor reduced meaning real-world impact underestimated



Basic Reporting – Refresher From LMI and II

Parameter	Estimate	SE	T score	P value	95% Confidence Interval	
					Lower Bound	Upper Bound
Constant / Intercept (β_o)	102	0.42	243	<2e-16	101	103
Coffee Taste	2.0	0.03	62	<2e-16	1.92	2.04
Sugar	2.0	0.03	62	<2e-16	1.90	2.01

Model Fit is => $Y_i = \beta_o + X_i\beta_1 + x_i\beta_1 + \varepsilon_i$ => Overall Liking = 1.03 + 2*coffee taste + 2*sugar + ε_i

“There is strong evidence to show that Overall Liking is associated with both Coffee Taste (p<2e-16) and Sugar (p<2e-16). With a 1 point increase in Coffee Taste associated with an increase in liking of between 1.92-2.04. Sugar had a very similar effect of between 1.90-2.01. This correlation on liking has been estimated very precisely.

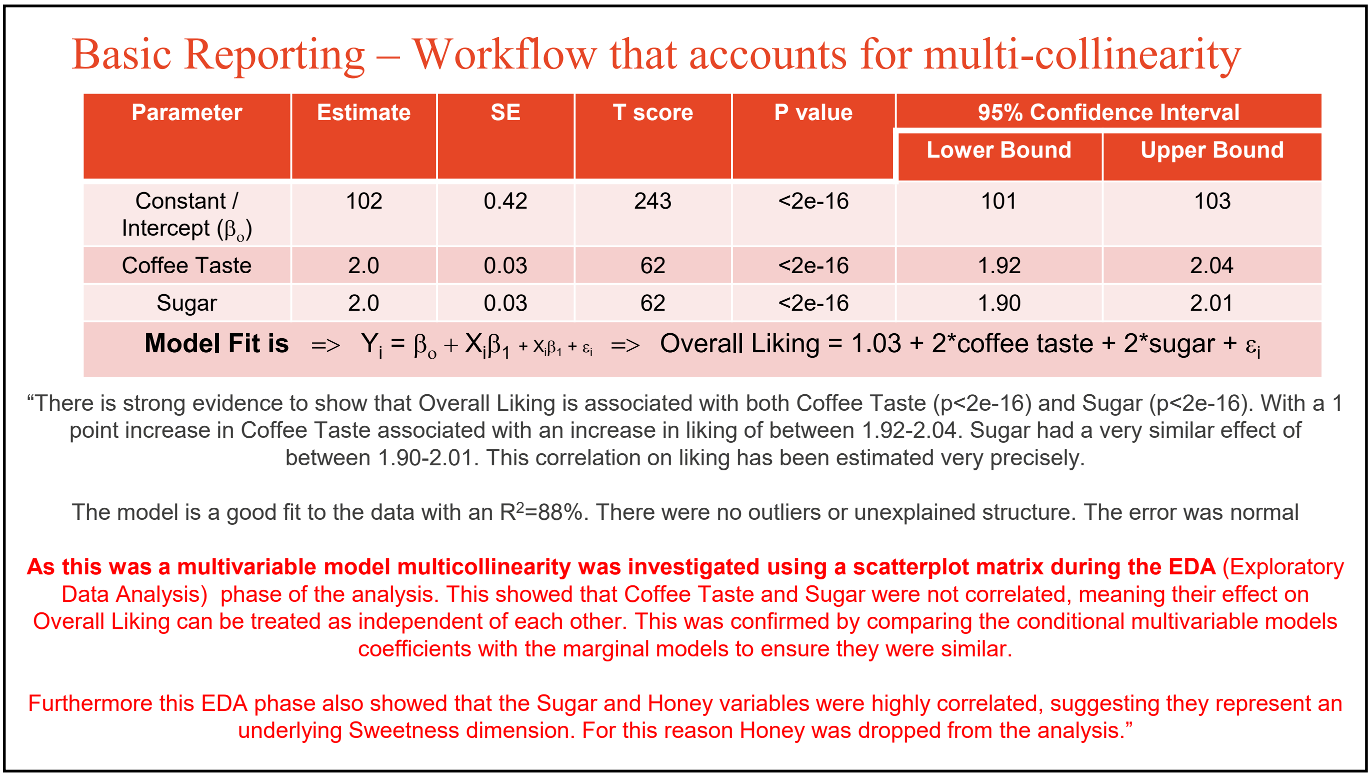
The model is a good fit to the data with an R^2=88%. There were no outliers or unexplained structure. The error was normal”

Notice

1. **When giving a p-value always give an estimate of the effect size as well** i.e. the 95% CI.
2. I have **shied away from causal language** since this type of study is often observational rather than experimental. This is an example of how **the same statistical analysis and results** can have **very different casual interpretations based solely on the Experimental design**.

For more info on how your experimental design determines how strong of a causal link your analysis provides refer to our **Experimental Design and Research Essential Workshops**.

35



Basic Reporting – Workflow that accounts for multi-collinearity

Parameter	Estimate	SE	T score	P value	95% Confidence Interval	
					Lower Bound	Upper Bound
Constant / Intercept (β_o)	102	0.42	243	<2e-16	101	103
Coffee Taste	2.0	0.03	62	<2e-16	1.92	2.04
Sugar	2.0	0.03	62	<2e-16	1.90	2.01

Model Fit is => $Y_i = \beta_o + X_i\beta_1 + x_i\beta_1 + \varepsilon_i$ => Overall Liking = 1.03 + 2*coffee taste + 2*sugar + ε_i

“There is strong evidence to show that Overall Liking is associated with both Coffee Taste (p<2e-16) and Sugar (p<2e-16). With a 1 point increase in Coffee Taste associated with an increase in liking of between 1.92-2.04. Sugar had a very similar effect of between 1.90-2.01. This correlation on liking has been estimated very precisely.

The model is a good fit to the data with an R^2=88%. There were no outliers or unexplained structure. The error was normal

As this was a multivariable model multicollinearity was investigated using a scatterplot matrix during the EDA (Exploratory Data Analysis) phase of the analysis. This showed that Coffee Taste and Sugar were not correlated, meaning their effect on Overall Liking can be treated as independent of each other. This was confirmed by comparing the conditional multivariable models coefficients with the marginal models to ensure they were similar.

Furthermore this EDA phase also showed that the Sugar and Honey variables were highly correlated, suggesting they represent an underlying Sweetness dimension. For this reason Honey was dropped from the analysis.”

36

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)?

37

So Remember

Predictors can only be interpreted **independently** if they are **independent**.

If they are **dependant** (correlated) they need to be interpreted **dependant** on each other (in the context of each other).

*Another way of saying this is that when there is **multicollinearity** predictors need to be interpreted in the **context** of each other.*

To do this one needs to:

1. Determine how dependant/correlated they are
2. Where they are dependant/correlated

38

Statistical Workflow for **interpretable models**
ALWAYS start with EDA that identifies multicollinearity

This workflow gives an interpretable model by accounting for multicollinearity. Which is why it should always be used.

Always start by assessing if there is a multicollinearity problem using EDA (Exploratory Data Analysis) plots such as pairwise scatterplots and correlations.
- ***A correlation with r>0.7-0.8 is often considered high enough multicollinearity to warrant intervention.*** This is domain specific so find appropriate references to support your decision.
- This is another reason good EDA is so important. Since it removes multicollinearity right from the beginning and allows for simple interpretation at the end.

If multicollinearity is found then account for it, the following slides show some possible workflows.

The EDA methods shown here tend to only pick up pairwise correlation, other methods are required for higher dimensionality multicollinearity.
- The Model Building workshop covers other model assessment methods to assess multicollinearity such as Variance Inflation Factors (VIF's).
- Comparing ANOVA Sum of Squares I (sequential) to Sum of Squares III (partial) can also help, with differences suggesting multicollinearity/confounding.

Statistical Workflow for **interpretable models**:
Accounting for Multicollinearity
Method 1) Use only uncorrelated variables

These methods account for multicollinearity by using iterative models to identify and then remove it by only using a subset of variables that aren't highly correlated.

Start Simple and Get Complex
- If multicollinearity is a problem you can ***Start Simple and Get Complex.*** I usually use the following method:
  - ***Step 1) Identify which parameters can't be interpreted independently in the complex model.*** Done by fitting all the simple pairwise models between response and all predictor/covariate pairs to establish a benchmark, which subsequent more complex models can be compared to. Any large differences require interpretation e.g. maybe they are they effect after accounting for the other variables?
  - ***Step 2) Identify subgroups with different behaviours.*** Done by fitting 2 way predictor*covariate interactions to look for subgroups e.g. blood pressure and gender.
- This is the method we used in the preceding Coffee Taste example. The EDA showed that Honey and Sugar represent the same underlying sweetness dimension which we need to account for to make the model parameters interpretable. We then chose to only include one of them. We used the following iterative models when doing this:
  1. Showing that if one replaces Sugar with Honey in all models their parameters are approximately the same (or vice versa).
  2. Including them in the same model changes their parameters from those that only have 1 or the other.
  3. That adding Honey to a model with Sugar (or vice versa) does not improve the model fit using appropriate methods such as no increase in R2 or the Likelihood Ratio Test.

Statistical Workflow for **interpretable models**: Accounting for Multicollinearity Method 1) Use only uncorrelated variables

These methods account for multicollinearity by using iterative models to identify and then remove it by only using a subset of variables that aren't highly correlated.

Start Complex – Then Simplify

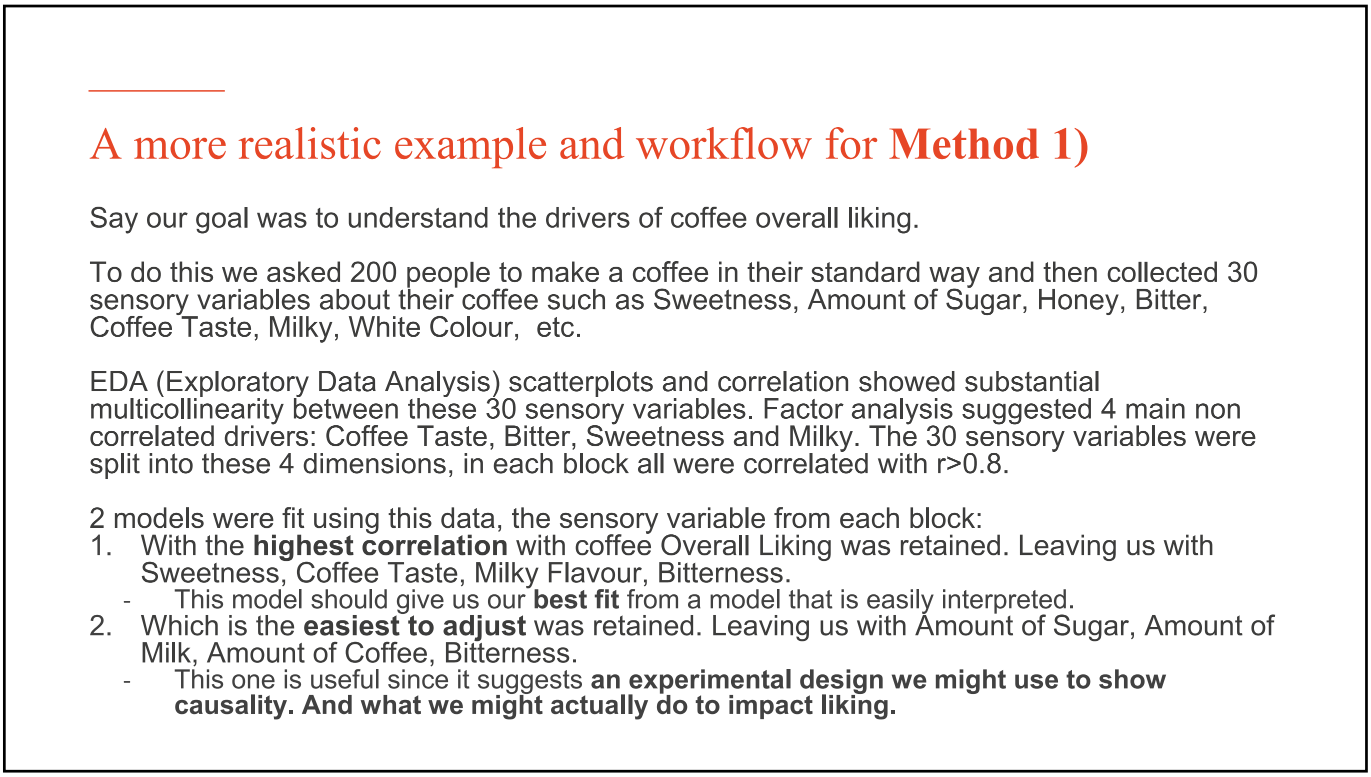
- You can also ***Start Complex – Then Simplify***. Meaning you fit a complex model and then drop each predictor one at a time to see how it impacts the other predictors coefficients and interpretations.
- 1 problem with this is that dropping one at a time may not lead to much of a difference if the correlation is spread across lots of variables.
  - VIFs can alleviate some of this problem as they measure such correlation and can be used to identify which variables may be worth dropping, even if when doing so one at a time the parameters change very little.
  - This problem is one reason I prefer to Start Simple and Get Complex to build up an understanding of the data. I can then move on this method if building more complex models, and use the simpler models as benchmarks.

Statistical Workflow for **interpretable models**: Accounting for Multicollinearity Method 2) Model the underlying latent variable

These methods account for multicollinearity by using models that incorporate it by directly modelling the underlying latent variable. **They do not require a subset of uncorrelated variables**. There are 2 common methods.

1. Use **Principle Components (PCA) or Factor Analysis (FA) to model the underlying dimension.** This is a 2 step process where we:
   i. Create a Sweetness factor from Sugar and Honey using Multivariate methods such as Principle Components or Factor Analysis and then,
   ii. Use that in the model instead of either of them (refer to Multivariate Workshop for how to do this).

2. **Model the dimension and its relationship to the response** using path/network models such as Partial Least Square Regression (PLS) or Structural Equation Modelling SEM (refer to Surveys 2 for a brief example).



A more realistic example and workflow for **Method 1)**

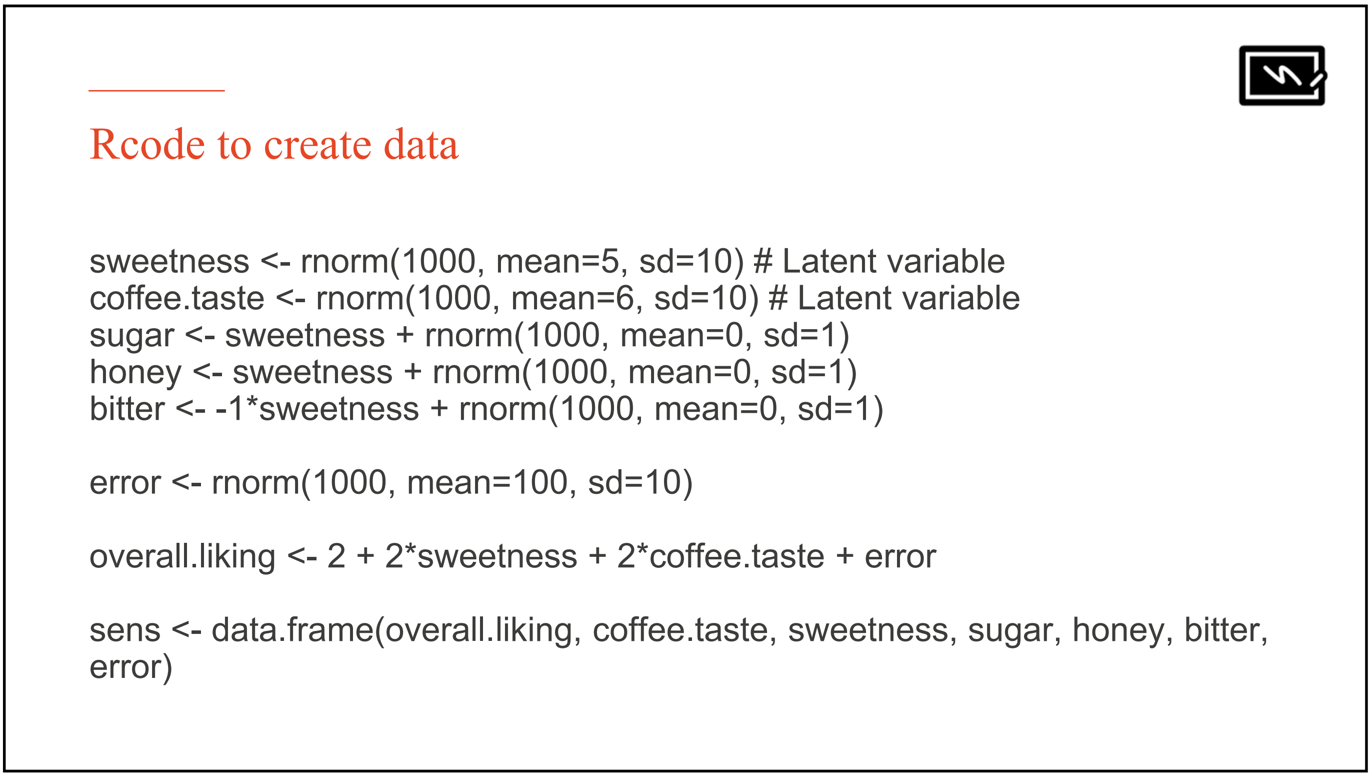
Say our goal was to understand the drivers of coffee overall liking.

To do this we asked 200 people to make a coffee in their standard way and then collected 30 sensory variables about their coffee such as Sweetness, Amount of Sugar, Honey, Bitter, Coffee Taste, Milky, White Colour, etc.

EDA (Exploratory Data Analysis) scatterplots and correlation showed substantial multicollinearity between these 30 sensory variables. Factor analysis suggested 4 main non correlated drivers: Coffee Taste, Bitter, Sweetness and Milky. The 30 sensory variables were split into these 4 dimensions, in each block all were correlated with r>0.8.

2 models were fit using this data, the sensory variable from each block:

1. With the **highest correlation** with coffee Overall Liking was retained. Leaving us with Sweetness, Coffee Taste, Milky Flavour, Bitterness.
   - This model should give us our **best fit** from a model that is easily interpreted.
2. Which is the **easiest to adjust** was retained. Leaving us with Amount of Sugar, Amount of Milk, Amount of Coffee, Bitterness.
   - This one is useful since it suggests **an experimental design we might use to show causality. And what we might actually do to impact liking.**



Rcode to create data

```
sweetness <- rnorm(1000, mean=5, sd=10) # Latent variable
coffee.taste <- rnorm(1000, mean=6, sd=10) # Latent variable
sugar <- sweetness + rnorm(1000, mean=0, sd=1)
honey <- sweetness + rnorm(1000, mean=0, sd=1)
bitter <- -1*sweetness + rnorm(1000, mean=0, sd=1)

error <- rnorm(1000, mean=100, sd=10)

overall.liking <- 2 + 2*sweetness + 2*coffee.taste + error

sens <- data.frame(overall.liking, coffee.taste, sweetness, sugar, honey, bitter,
error)
```

Rcode for EDA (Exploratory Data Analysis) and Analysis

EDA (Exploratory Data Analysis)

```
panel.cor <- function(x, y, digits=2, cex.cor)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- cor(x, y)
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  test <- cor.test(x,y)
  Signif <- ifelse(round(test$p.value,3)<0.001,"p<0.001",paste("p=",round(test$p.value,3)))
  text(0.5, 0.25, paste("r=",txt))
  text(.5, .75, Signif)
}

windows()
pairs(sens[c(1,2,4,5)], main="Dimension Matrix", upper.panel=panel.cor)
graphics.off()
```

Analysis

```
lm.coffee <- lm(overall.liking~coffee.taste, data=sens)
(s.lm.coffee <- summary(lm.coffee))

lm.sugar <- lm(overall.liking~sugar, data=sens)
(s.lm.sugar <- summary(lm.sugar))

lm.honey <- lm(overall.liking~honey, data=sens)
(s.lm.honey <- summary(lm.honey))

lm.coffee.sugar <- lm(overall.liking~coffee.taste+sugar, data=sens)
(s.lm.coffee.sugar <- summary(lm.coffee.sugar))

lm.coffee.honey <- lm(overall.liking~coffee.taste+honey, data=sens)
(s.lm.coffee.honey <- summary(lm.coffee.honey))

lm.sugar.honey <- lm(overall.liking~sugar+honey, data=sens)
(s.lm.sugar.honey <- summary(lm.sugar.honey))

lm.coffee.sugar.honey <- lm(overall.liking~coffee.taste+sugar+honey, data=sens)
(s.lm.coffee.sugar.honey <- summary(lm.coffee.sugar.honey)
```

45

46

Reporting variable importance and using it to make actionable real-world recommendations

The impact of multicollinearity on variable importance metrics like % importance and the Shapley Value

Quadrant Analysis: using performance as well as importance when making recommendations

Other tips e.g. Stated Importance, ROI, etc

Reporting the “Importance” of Predictors

People often want to calculate the “importance” of predictors. There are many ways to do this. 2 common ways use the regression coefficients and the R-squared (R^2) from a linear regression. They often give similar results.

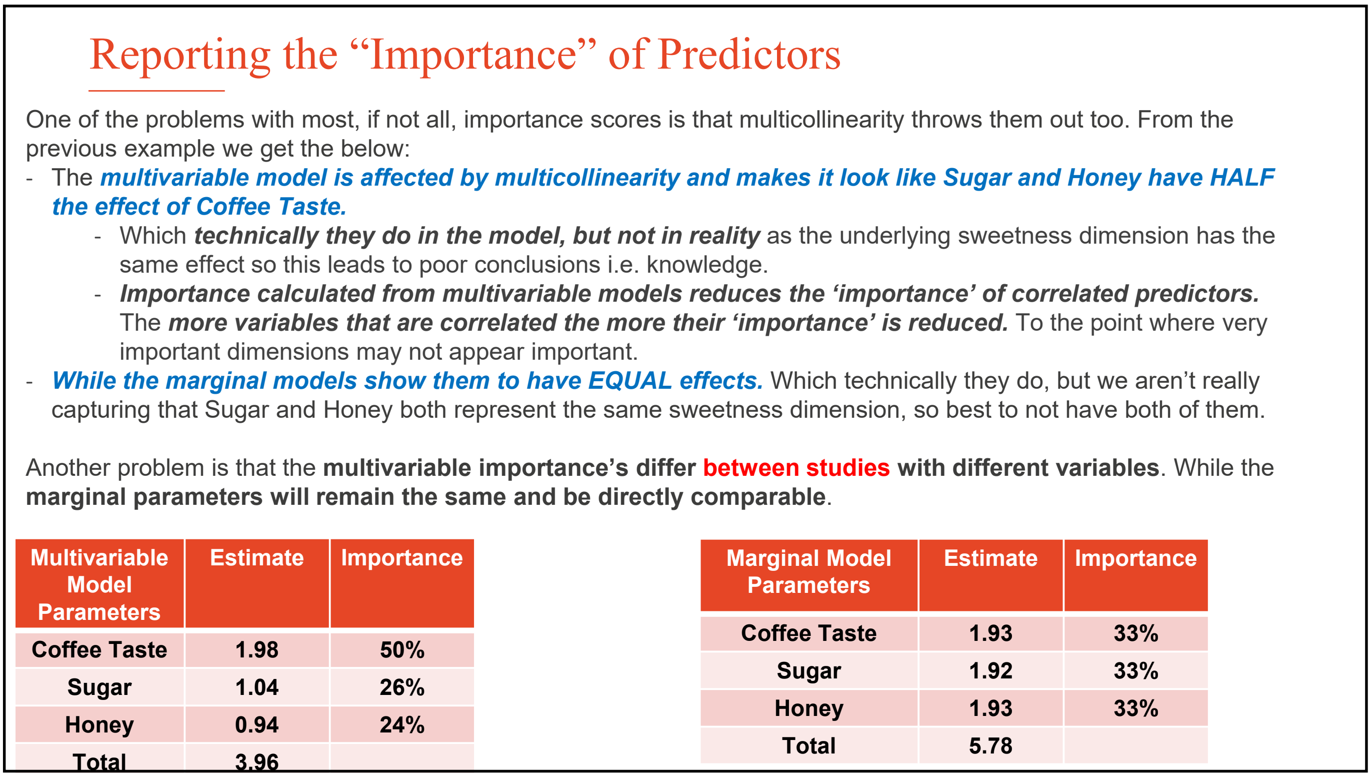
The regression coefficient method simply divides each regression parameter by their sum and then multiples by 100. To give a % importance score.

Both can be misleading so use with care, **neither are recommended.** ***Better to talk about them in terms of the relative difference in their parameters i.e. relative importance*** e.g.

- Example 1: Coffee Taste and Sugar have a similar association with Liking
- Example 2: Coffee Taste has 3 times the association as Sugar

Example 1 Parameter	Estimate	Importance
Coffee Taste	1.93	50%
Sugar	1.92	50%
Total	3.85	

Example 2 Parameter	Estimate	Importance
Coffee Taste	6.0	75%
Sugar	2.0	25%
Total	8	



Reporting the “Importance” of Predictors

One of the problems with most, if not all, importance scores is that multicollinearity throws them out too. From the previous example we get the below:

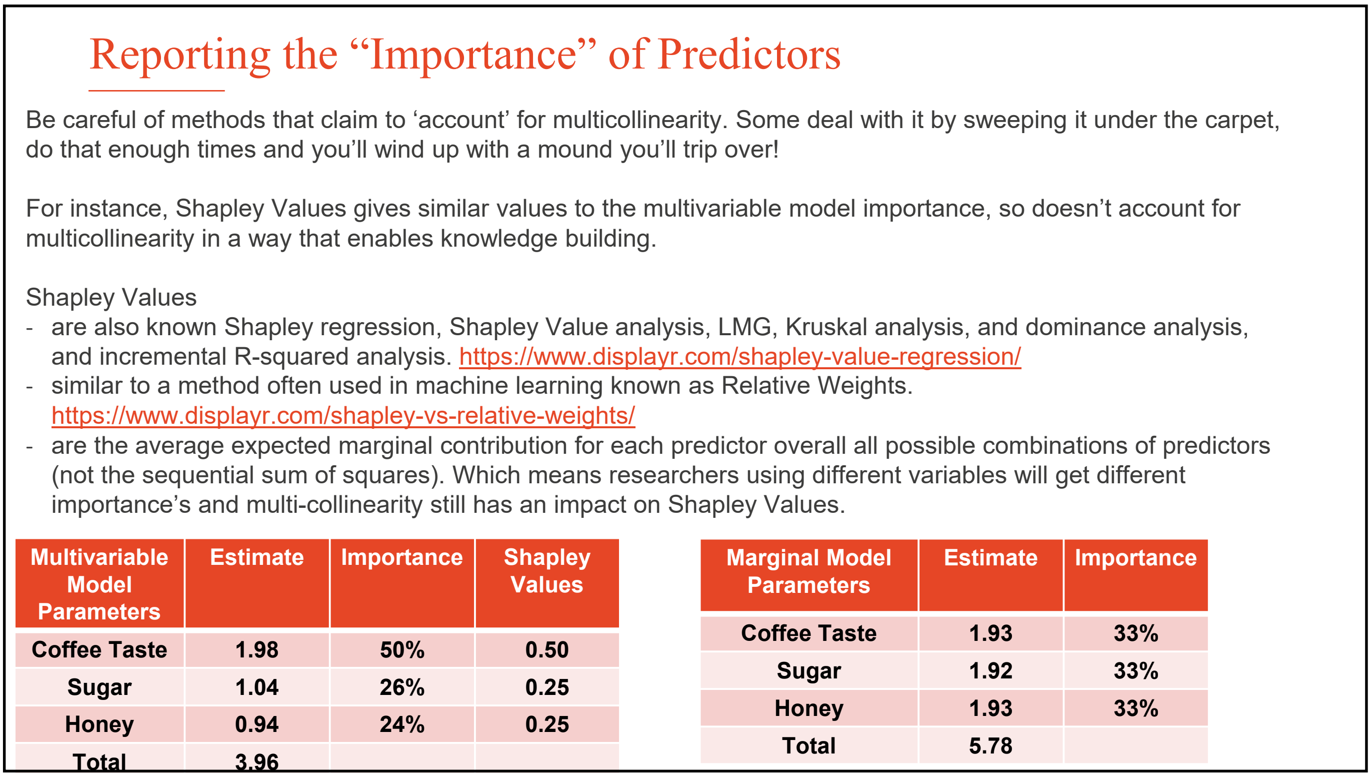
- The ***multivariable model is affected by multicollinearity and makes it look like Sugar and Honey have HALF the effect of Coffee Taste.***
  - Which ***technically they do in the model, but not in reality*** as the underlying sweetness dimension has the same effect so this leads to poor conclusions i.e. knowledge.
  - ***Importance calculated from multivariable models reduces the ‘importance’ of correlated predictors.*** The ***more variables that are correlated the more their ‘importance’ is reduced.*** To the point where very important dimensions may not appear important.
- ***While the marginal models show them to have EQUAL effects.*** Which technically they do, but we aren’t really capturing that Sugar and Honey both represent the same sweetness dimension, so best to not have both of them.

Another problem is that the **multivariable importance’s differ between studies with different variables**. While the **marginal parameters will remain the same and be directly comparable**.

Multivariable Model Parameters	Estimate	Importance
Coffee Taste	1.98	50%
Sugar	1.04	26%
Honey	0.94	24%
Total	3.96	

Marginal Model Parameters	Estimate	Importance
Coffee Taste	1.93	33%
Sugar	1.92	33%
Honey	1.93	33%
Total	5.78	

49



Reporting the “Importance” of Predictors

Be careful of methods that claim to ‘account’ for multicollinearity. Some deal with it by sweeping it under the carpet, do that enough times and you’ll wind up with a mound you’ll trip over!

For instance, Shapley Values gives similar values to the multivariable model importance, so doesn’t account for multicollinearity in a way that enables knowledge building.

Shapley Values

- are also known Shapley regression, Shapley Value analysis, LMG, Kruskal analysis, and dominance analysis, and incremental R-squared analysis. https://www.displayr.com/shapley-value-regression/
- similar to a method often used in machine learning known as Relative Weights. https://www.displayr.com/shapley-vs-relative-weights/
- are the average expected marginal contribution for each predictor overall all possible combinations of predictors (not the sequential sum of squares). Which means researchers using different variables will get different importance’s and multi-collinearity still has an impact on Shapley Values.

Multivariable Model Parameters	Estimate	Importance	Shapley Values
Coffee Taste	1.98	50%	0.50
Sugar	1.04	26%	0.25
Honey	0.94	24%	0.25
Total	3.96		

Marginal Model Parameters	Estimate	Importance
Coffee Taste	1.93	33%
Sugar	1.92	33%
Honey	1.93	33%
Total	5.78	

50

Derived/Modelled "importance" can miss Cost of Entry and other highly important predictors

The term "importance' when talking about model parameters is a poor descriptor and open to misinterpretation.

Importance's obtained from statistical analysis are ***Derived Importance's, and only identify predictors correlated with the response.*** Meaning they need both high and low scores and there is Room To Improve them. It allows the user to identify factors which they might reasonably change and possibly impact the response.

As such it will ***miss Cost of Entry drivers that are optimized*** i.e. everyone scores them high, or other variables that are always required. This is why **Stated Importance** is also important. We can get stated importance by asking respondents directly what is important to them, and also stating what researchers already know to be important.

Highlights the Importance of DAGs (Directed Acyclic Graphs) to show the data generating mechanism. They highlight the Cost of Entry drivers analysis won't detect as important, but actually are. Necessary for correctly interpreting the models real-world **meaning** in terms of what predictors are correlated/causally related to the response.

Meaning we can't assume variables with low derived importance aren't important e.g.

1. If looking at what's important to people when buying a parachute 'opening' would be an important stated driver, however likely not a derived one as all parachutes open (one would hope!) However 'colour' might not be an important stated driver but could be an important derived one - if different parachute models deliver this feature differently and it impacts the model bought.
2. If looking at what resources impact an animals distribution oxygen is pretty important! But in many studies would not come up as important as all sites collected would have enough oxygen. This may seem to be a silly example, but:
   - Might be very important to consider when researching new, novel or extraterrestrial life.
3. Shows why interpretable models require researchers to think through their research questions, sampling design and underlying theory. Automated artificial intelligence and machine learning models cannot yet do that and if left to themselves, without human level interpretation and intervention, often make mistakes.

51

Identifying which variables should be used to improve the response

Derived Importance's are often used to identify which variables should be used to improve the response.

A common mistake it to simply say everything with high derived importance should be improved.

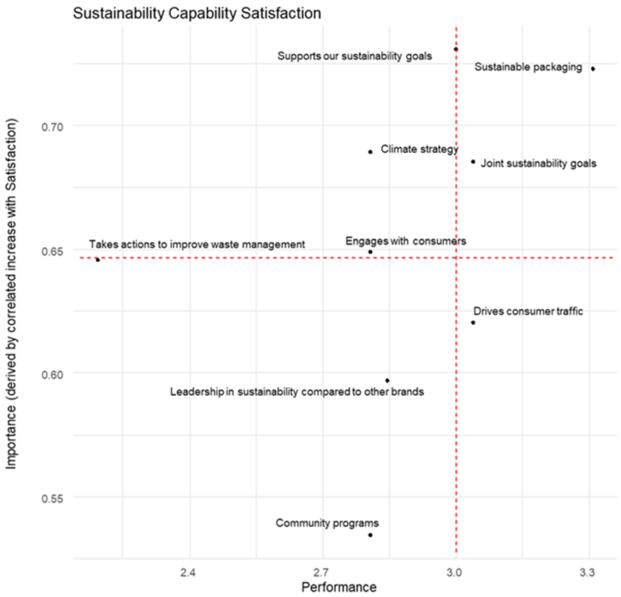
It's wrong as it ignores individual's actual performance. Which can lead to incorrect conclusions and sub optimal resource allocation.

For example. A worldwide health study on certain birth defects would likely find that a lack of folate is strongly corelated.

- However, stating that all countries should increase folate consumption as a remedy is a naive recommendation, since Australia already fortifies bread with folate.
- Instead, one would first need to determine which countries i) have a high incidence of these birth defects, and ii) if lack of folate is a likely cause given what public policy they currently have.

52

Identifying which variables should be used to improve the response: Quadrant maps



Case Study: What drives peoples purchase decisions for product X.

- Data collected is product satisfaction and performance on a range of attributes. Derived importance over lots of products tells us which factors are correlated with overall satisfaction and have Room to Improve.

By including performance for product X we now have a **Quadrant analysis** which identifies which factors should be:

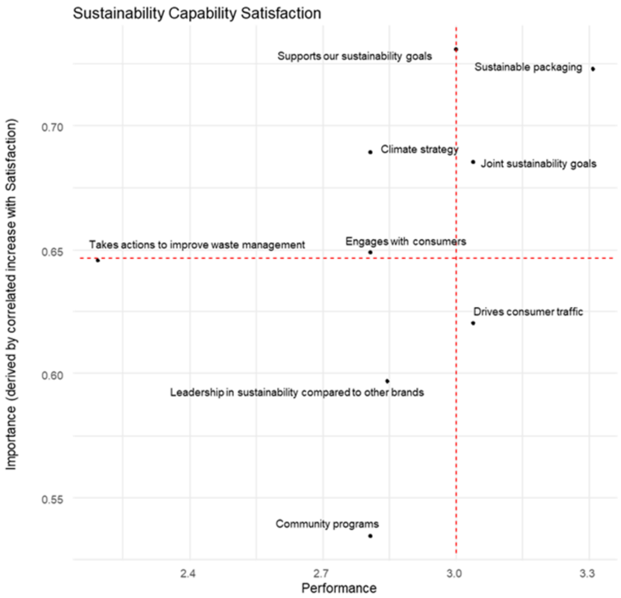
- **Improved:** important and underperforming (upper left)
- **Maintained:** important and over performing (upper right)

And after considering their stated importance we can also consider if factors could be:

- **Over Resourced,** and have their resources allocated elsewhere: not important and underperforming (lower left)
- **Maintained:** not important and underperforming (lower right)

53

Identifying which variables should be used to improve the response: Quadrant maps



- A large drink brand wanted to know how satisfied their retailer partners were, their capability for sustainability and where to focus their efforts if any improvement was necessary.
- From Driver analysis and this Quadrant map we now know:
  - (Top left quad) Climate Strategy is underperforming and important so improving would likely increase Satisfaction.
  - (Bottom right quad) Conversely, one of the highest performers was Drives Consumer Traffic but possibly not worth focussing on as its derived importance wasn't that high.
  - Meaning they can **optimise resources** by possibly shifting priorities and resources from things that drive Consumer Traffic to those that drive Climate Strategy. ***If Consumer Traffic does not have high stated importance.***

54

Because it's so important a reminder!

Stated Importance needs to be considered too

As ***derived importance may not flag important cost of entry dimensions which are optimised*** (as they have no Room to Improve) care should be taken using this information to reduce resourcing without the corresponding stated importance quadrant analysis.

For example, if analysing parachutes then opening would be an important stated driver, however not a derived one as all parachutes open (one would hope!).

Meaning opening might come out as something over-resourced and could maybe be cut. Which is obviously wrong.

Considering stated importance ensures we don't come to this incorrect conclusion for less obvious examples which are quite common when doing research.

55

Other considerations when using derived importance to optimise

Return on Investment (ROI)
Practically it can be better and more efficient to take easy wins by trying to improve something that is under performing, but isn't as important than something else which may be more important but is also performing much better. This is because it often takes more effort and costs more money to improve things that are already over benchmark.

Improving highly performing predictors may have the reverse effect
Improving highly performing variables may take them outside the sampling space i.e. predictors range. Realistically you can go a bit out of the data range collected but too far and the impact might be the reverse of what is expected.
For example: Sensory attributes like Sugar are often modelled as linear even though we know they are usually unimodal. Practically this is often necessary due to small samples and as products tend to be optimised we often sample either side of the optimal point as products don't go past it – meaning we only capture the linear part. However, if we push to far past it can turn from good to bad i.e. too sweet.

56

57

Reporting Categorical Predictors

ANOVA vs Parameter Tables

Confidence vs Prediction Intervals

Multiple Comparison

Estimated Marginal Means vs Parameters

58

Categorical predictor tests and p-values

There are 2 common outputs:

1. ANOVA table
2. Parameter estimates

Consider hair colour as a predictor for # of freckles.

The **ANOVA TABLE** tells us that there is an overall association between hair colour and freckles (p<2.2e-16). In general, we look at this one first to determine if there is an overall/familywise/global effect so we can report as such e.g. hair colour is associated with the # of freckles someone has.

Parameter	Degrees of Freedom (DF)	Sum of Squares (SS)	Mean (MS)	F value	P value
Hair Colour	3	84943183	28314394	286624	<2.2e-16
Residuals/Error	396	39119	99		

59

Categorical predictor tests and p-values

BUT it doesn't describe exactly how the category *Hair Colour* is correlated with the response *Freckles*, which is what the **PARAMETER TABLE** does. This tells us that:

- Our **reference category of Black Hair has about 8 freckles** (p=2.08e-15), so in general most black haired people have between 6-10 freckles (95% CI).
- And that compared to our black haired reference level:
  - **Blondes have on average 91** more freckles (p<2.2e-16), which is more precisely estimated as being between 88-94 (95% CI)
  - There is **no evidence that brown haired** folk have a different # of freckles since p>0.05. Although **one might say there is some weak evidence** of about 3 more since p=0.06.
  - **Redheads tend to have just over 1000 more freckles!!** (since its estimate is 1092, p=2.22e-16, 95% CI=[1089, 1094])

Parameter	Estimate	SE	T score	P value	95% Confidence Interval	
					Lower Bound	Upper Bound
Constant (Black)	8	0.99	8.3	2.08e-15	6	10
Blonde	91	1.4	64.6	<2.2e-16	88	94
Brown	3	1.4	1.9	0.0627	-0.1	5
Red	1092	1.4	776	<2.2e-16	1089	1094

60

Categorical predictor tests and p-values

But there is a bit of a problem with using parameters to describe and report the associations. Can anyone see what it is? Hint: what if we wanted to focus on redheads?

It only describes the difference from the black haired folk it doesn't (effectively):

- **Tell us the overall # of freckles we expect each hair type to have overall.** (This can be done by ***predicting*** the number each should have and putting a ***confidence or prediction interval*** around it.)
- **Compare to other hair types** e.g. Redheads to all other hair types. This is done using ***Multiple Pair Wise Comparisons*** or changing the ***Parametrisation*** so a different hair colour is used as the reference level.

Parameter	Estimate	SE	T score	P value	95% Confidence Interval	
					Lower Bound	Upper Bound
Constant (Black)	8	0.99	8.3	2.08e-15	6	10
Blonde	91	1.4	64.6	<2.2e-16	88	94
Brown	3	1.4	1.9	0.0627	-0.1	5
Red	1092	1.4	776	<2.2e-16	1089	1094

Predicting the # of freckles we expect each hair type to have

There are 2 common ways to do this:

1. **Confidence Intervals** estimate the number of freckles **all the people** in a hair type have e.g. the average number of freckles all redheads have is between 1098-1102.
2. **Prediction Intervals** estimate the number of freckles **an individual** can expect to have e.g. the number of freckles we can expect an individual redhead to have is between 1081-1120.
   - They are wider than confidence intervals since we expect an individual to be more variable than the average of lots of individuals.

Hair Colour	Point Estimate	95% Confidence Interval		95% Prediction Interval	
		Lower Bound	Upper Bound	Lower Bound	Upper Bound
Black	8	6	10	-11	28
Blonde	99	97	101	79	119
Brown	11	9	13	-9	30
Red	1100	1098	1102	1081	1120

Multiple Comparisons

The **parametrisation** we've used makes black haired people the **reference group**. So the parameters tell us the difference between other levels compared to this reference group.

But what if we want to compare other groups e.g. red with blond?

This is where we do **multiple comparisons** which compare all possible pairwise comparisons of the levels.

Notice that we made 6 different comparisons.

```
contrast        estimate    SE  df t.ratio  p.value
1 black - blonde   -90.850 1.406 396  -64.634 <.0001
2 black - brown     -2.624 1.406 396   -1.867 0.2441
3 black - red    -1092.028 1.406 396 -776.911 <.0001
4 blonde - brown    88.226 1.406 396   62.767 <.0001
5 blonde - red   -1001.178 1.406 396 -712.277 <.0001
6 brown - red    -1089.404 1.406 396 -775.045 <.0001

P value adjustment: tukey method for comparing a family of 4 estimates
```

Say we made 20 such (unadjusted) multiple comparisons and they all had p=0.05. If we concluded that all of them showed a real difference in the population how many would we expect to be wrong (on average)?

A) None
B) 1 Correct
C) 5
D) Can't tell

A p-value of 5% means we make the wrong decision to reject the null hypothesis of no effect and accept there is one 5% or 1 in 20 times. Since we made 20 comparisons with a p-value of 5% we expect to come to the wrong conclusion 1 in 20 times (on average).

And we don't know which one is likely wrong either!

So how do we fix this?

Correcting for Multiple Comparisons

We effectively **adjust the p-value cut off to keep the family wide error rate of all comparisons at 5%.**

The simplest method is called **Bonferroni** and simply divides the family wide p-value we want by the # of comparison we make.

New Bonferroni p-value = $\frac{\text{family wide p-value}}{\#\text{ of comparisons}}$

E.G.: New Bonferroni p-value = $\frac{0.05}{20}$ = 0.0025

65

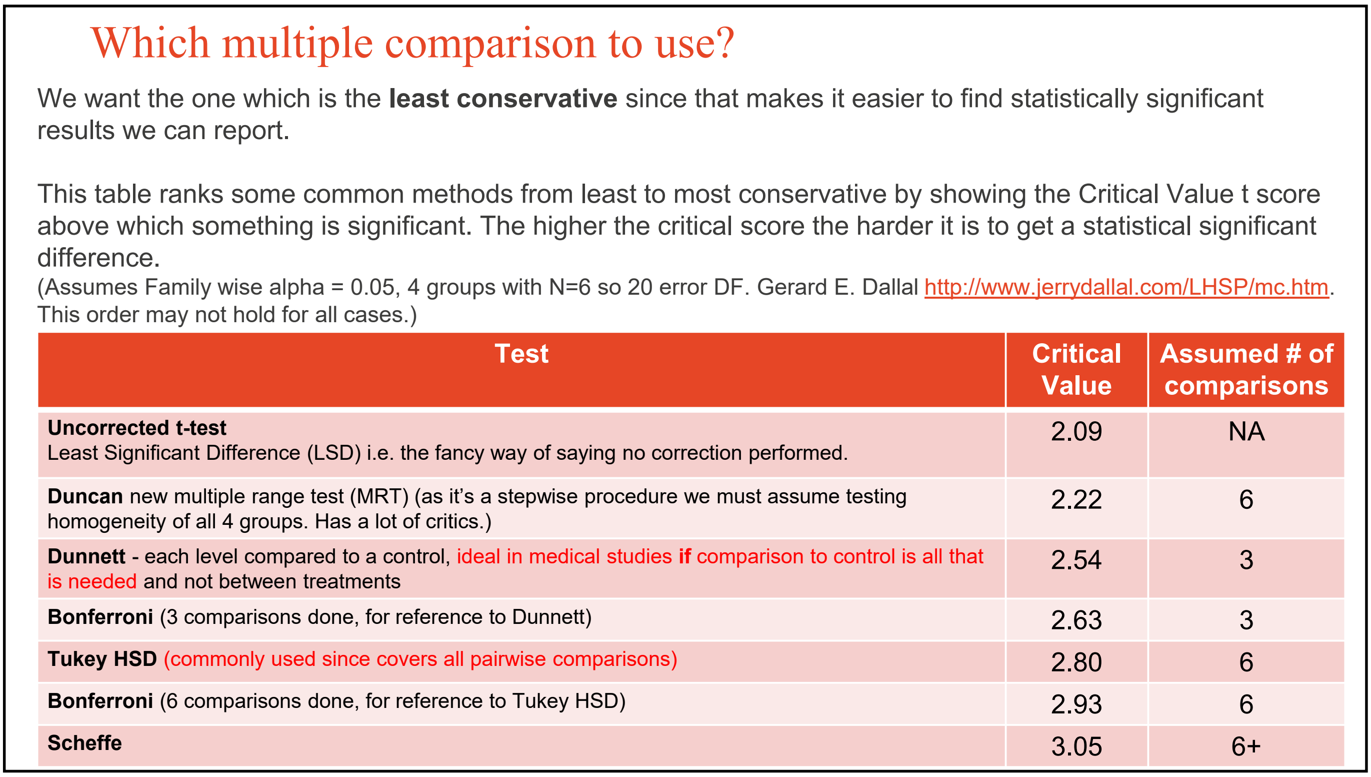
Correcting for Multiple Comparisons

Unfortunately, Bonferroni is overly conservative i.e. it makes the adjusted p-value unnecessarily small, making it harder to find statistically significant results worth reporting.

To fix this there are other multiple comparisons that are less conservative, the one we used is Tukey's which assumes we are comparing all possible means.

```
contrast        estimate    SE  df t.ratio  p.value
1 black - blonde   -90.850 1.406 396  -64.634 <.0001
2 black - brown     -2.624 1.406 396   -1.867 0.2441
3 black - red    -1092.028 1.406 396 -776.911 <.0001
4 blonde - brown    88.226 1.406 396   62.767 <.0001
5 blonde - red   -1001.178 1.406 396 -712.277 <.0001
6 brown - red    -1089.404 1.406 396 -775.045 <.0001

P value adjustment: tukey method for comparing a family of 4
estimates
```

66



Which multiple comparison to use?

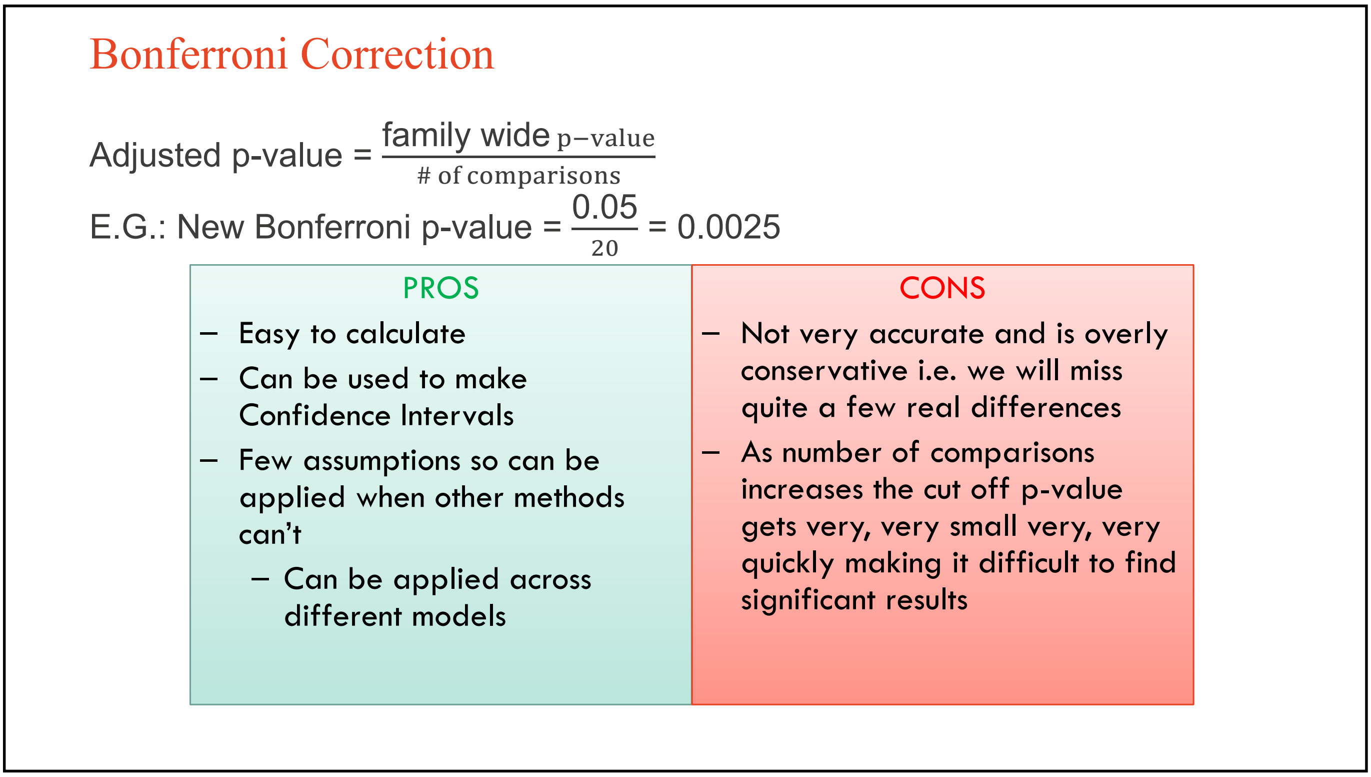
We want the one which is the **least conservative** since that makes it easier to find statistically significant results we can report.

This table ranks some common methods from least to most conservative by showing the Critical Value t score above which something is significant. The higher the critical score the harder it is to get a statistical significant difference.

(Assumes Family wise alpha = 0.05, 4 groups with N=6 so 20 error DF. Gerard E. Dallal http://www.jerrydallal.com/LHSP/mc.htm. This order may not hold for all cases.)

Test	Critical Value	Assumed # of comparisons
Uncorrected t-test Least Significant Difference (LSD) i.e. the fancy way of saying no correction performed.	2.09	NA
Duncan new multiple range test (MRT) (as it's a stepwise procedure we must assume testing homogeneity of all 4 groups. Has a lot of critics.)	2.22	6
Dunnett - each level compared to a control, ideal in medical studies **if** comparison to control is all that is needed and not between treatments	2.54	3
Bonferroni (3 comparisons done, for reference to Dunnett)	2.63	3
Tukey HSD (commonly used since covers all pairwise comparisons)	2.80	6
Bonferroni (6 comparisons done, for reference to Tukey HSD)	2.93	6
Scheffe	3.05	6+

67



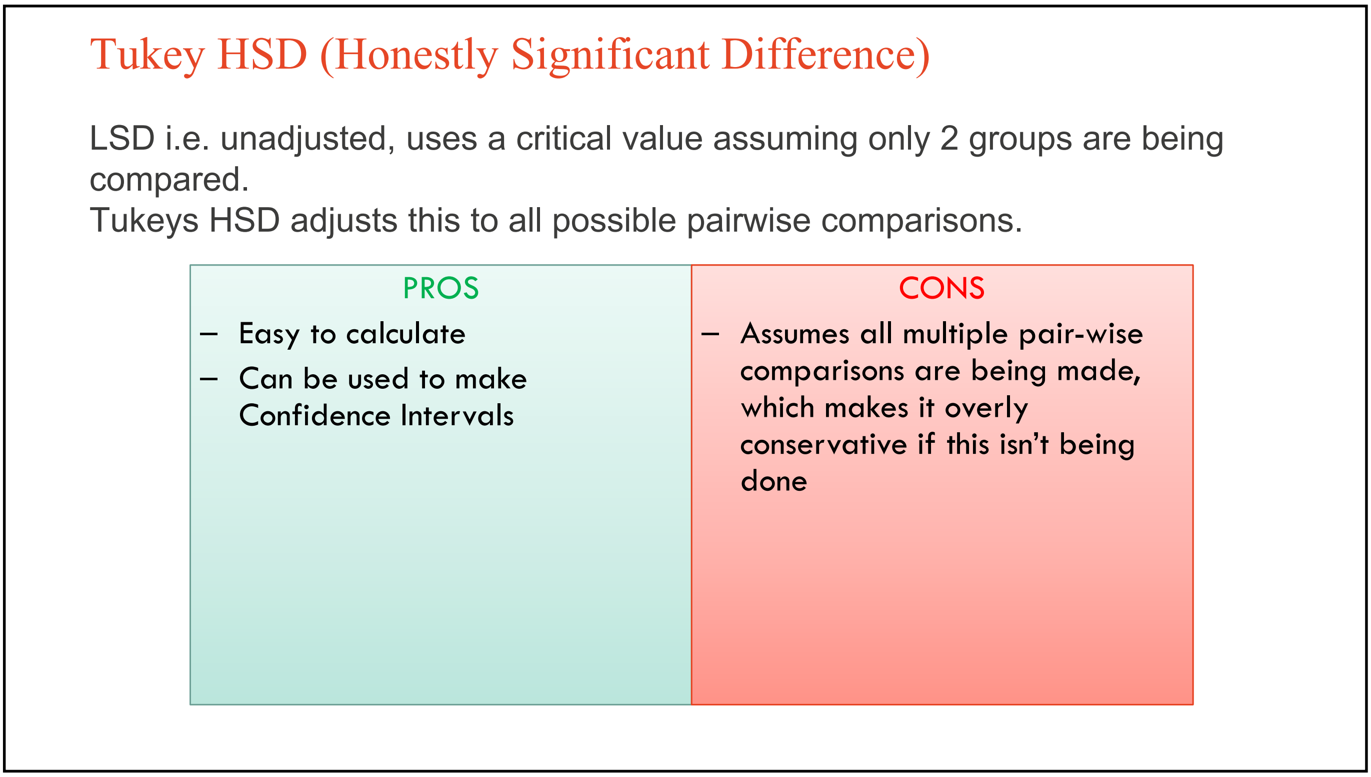
Bonferroni Correction

Adjusted p-value = $\frac{\text{family wide p−value}}{\text{\# of comparisons}}$

E.G.: New Bonferroni p-value = $\frac{0.05}{20}$ = 0.0025

PROS	CONS
– Easy to calculate – Can be used to make Confidence Intervals – Few assumptions so can be applied when other methods can't – Can be applied across different models	– Not very accurate and is overly conservative i.e. we will miss quite a few real differences – As number of comparisons increases the cut off p-value gets very, very small very, very quickly making it difficult to find significant results

68

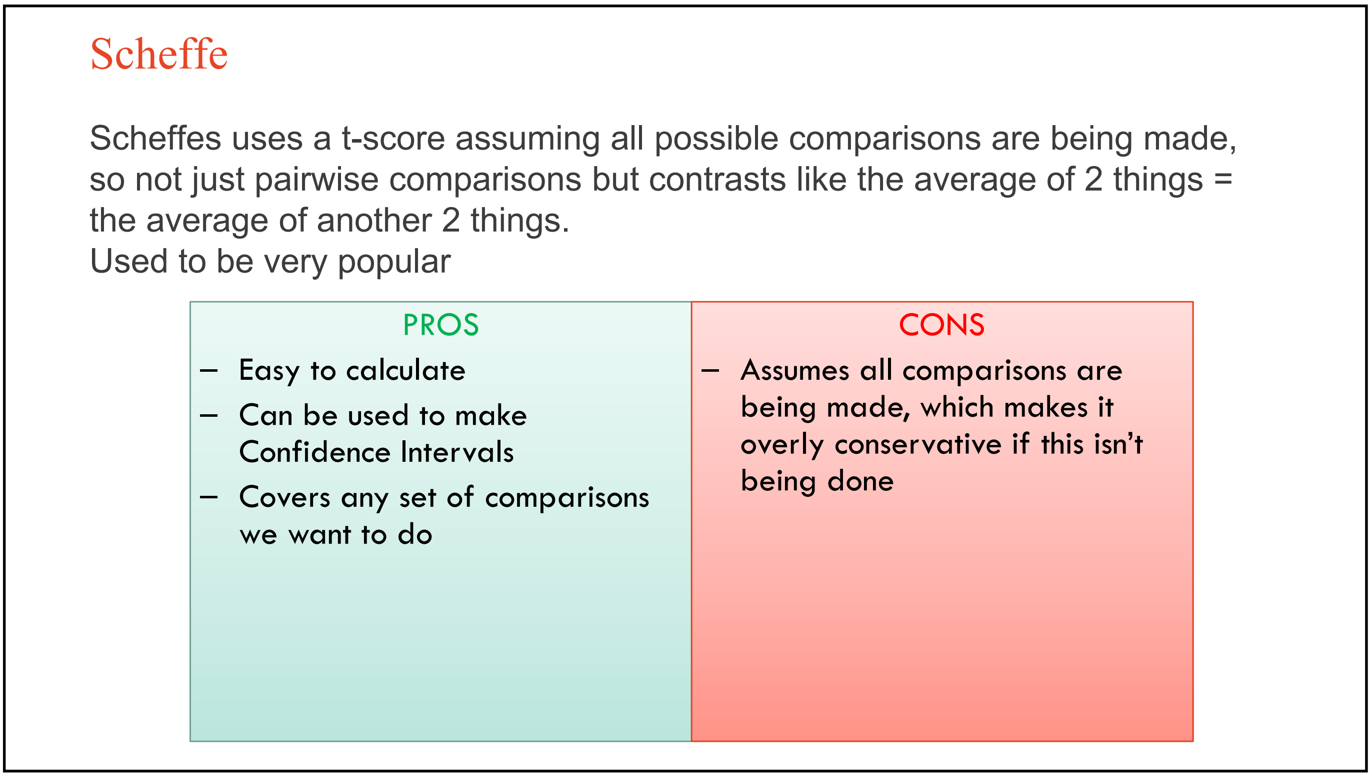


Tukey HSD (Honestly Significant Difference)

LSD i.e. unadjusted, uses a critical value assuming only 2 groups are being compared.
Tukeys HSD adjusts this to all possible pairwise comparisons.

PROS	CONS
– Easy to calculate – Can be used to make Confidence Intervals	– Assumes all multiple pair-wise comparisons are being made, which makes it overly conservative if this isn’t being done

69



Scheffe

Scheffes uses a t-score assuming all possible comparisons are being made, so not just pairwise comparisons but contrasts like the average of 2 things = the average of another 2 things.
Used to be very popular

PROS	CONS
– Easy to calculate – Can be used to make Confidence Intervals – Covers any set of comparisons we want to do	– Assumes all comparisons are being made, which makes it overly conservative if this isn’t being done

70

Dunnet

Uses a t-score assuming groups are being compared to a single control.

PROS	CONS
– Easy to calculate – Can be used to make Confidence Intervals – Accurate when applicable	

71

Duncans new multiple range test (MRT)

Uses a completely different approach to the previous methods. Finds **homogenous groups**, rather than looking for differences.

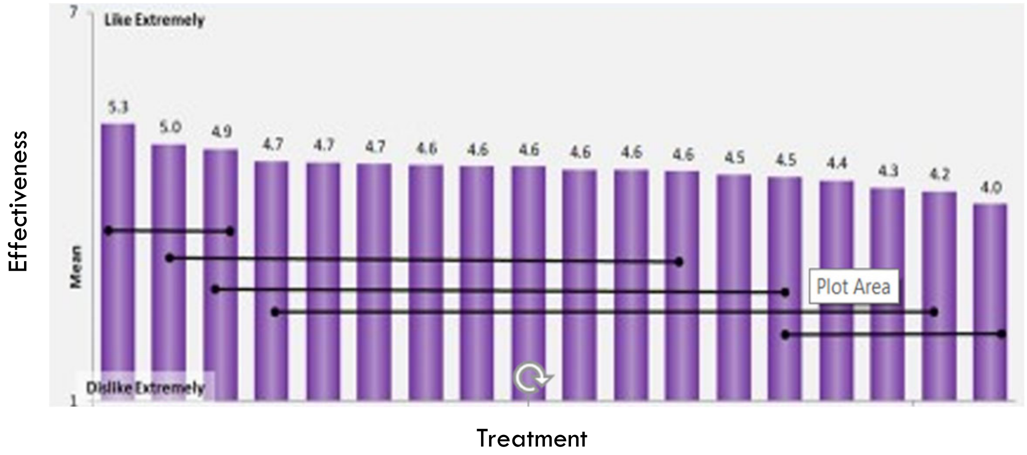
Based on the [Student]-Newman-Keuls Procedure but with greater power. This is a complex algorithmic procedure.

PROS	CONS
– Least conservative i.e. more significant differences so more to talk about. But some say too liberal. – Useful if we want to find homogenous groups. – A quick way of doing fuzzy clustering. – An efficient way to summarise lots of groups in 1 slide.	– Can't be used to make Confidence Intervals that match the test results – Has some (a lot?) of critics so may get criticized at the review stage if not used in your domain

72

Homogenous Subset Example

- Bars linked with a black line form a homogenous group i.e. there is no significant difference.
- Duncan's Multiple Range Test (MRT) is one way to get these.



Hypothesis testing vs Screening/Exploratory analysis

There is considerable debate about when Multiple Comparisons should be used, preferences can be quite domain specific.

One generally **always tests 'within model and/or factor' comparisons, but rarely between model comparisons** i.e. also known as correcting for multiple ***testing*** to distinguish it from multiple ***comparisons.*** For example: if we had a single model for freckles with 2 predictors: hair colour (4 options) and eye colour (4 options) we would generally correct each predictor for multiple comparisons independently i.e. assume 6 comparisons were being done for each. We wouldn't sum up the total comparison and correct for 12. Similarly if we ran 2 different models each with a different predictor we would correct each one independently.

1 useful distinction I often make is the difference between Hypothesis Testing vs Screening/Exploratory Analysis.

Hypothesis testing

- Requires corrections for Multiple Comparisons, e.g. Bonferroni, Tukey, Holmes, False Discovery Rate.
- Is when we are testing apriori theories developed from previous research or modelling and are the focus of the paper. Usually only a few are made.
- Often used to make important decisions with minimal or no supporting evidence.
- EXAMPLE: Randomised clinical trials to evaluate 3 vaccines, Comparing a new formulation to the existing product, Land management Trials.

Screening/Exploratory Analysis i.e. Screening lots of tests for possibly interesting pattern.

- Often doesn't correct for all multiple comparisons being done.
- Is when we do lots of tests looking for unknown associations or interesting patterns.
- Often used to suggest future research.
- If used to make decisions must be in conjunction with other information e.g. other studies, qualitative work, prior expert knowledge.
- EXAMPLE: Pharmacological study on 1000's of off the shelf medications impact on covid to identify those worth moving into better randomised clinical trials , analysing a survey with lots of questions and splits, driver analysis between numerous sensory/hedonistic variables and liking, data mining.

Surveys

Surveys often consider analyses of each question a different test, so we don’t correct for multiple testing.

We also consider different splits of the same variable as different tests e.g. if comparing different medical treatments between genders, age and BMI we don’t correct for all of them at the same time. **Instead of using strict hypothesis testing we take the view that these p-values are used to screen** all the different comparisons being done to see what might be worthwhile incorporating into the story and to generate hypotheses to be tested in future research.

We do however often correct for comparisons between different categories within a single variable e.g. if we had 4 age groups that’s 6 different pairwise comparisons which we would usually correct for. Sometimes though we can have so many different categories to make even this problematic as correcting for multiple comparisons in the normal ways usually means nothing is worthwhile reporting.

As such we can also report both. For instance, if one was comparing some statements to a benchmark one can use colour, font and/or asterisk’s to signify whether something has a p-value <0.05 **with and without correcting for multiple comparisons (MC)**.

The basic idea is that **as we are more sure of those corrected for multiple comparisons we bring more attention to them**.

Method	P<0.05 No MC correction	P<0.05 MC correction
Colour	Light Red	Dark Red
Asterisk	*	**
Bold or not	Not Bold	Bold

75

Significance testing, colour coding and screening

Example 1 - Colour

Importance of Animal Welfare on purchase decisions	% who agree
Australian Average (Benchmark)	**50%**
Vegetarian	**90%**
Byron Bay	60%
Low Socio Economic Band	**20%**
Sydney	53%

Example 2 - No colour so can be used in more journals

Importance of Animal Welfare on purchase decisions	% who agree
AUSTRALIAN AVERAGE (BENCHMARK)	**50%**
Vegetarian	**90% ** **
Byron Bay	60% *
Low Socio Economic Band	**20% ** **
Sydney	53% *

76

77

Reporting more than 1 categorical predictor: Estimated Marginal Means (EMMs)

Reporting more than 1 categorical predictor presents some challenges.

Let's extend our example to include the factor SUN with 2 levels

1. Bronzed Bondi Beach Bathers (BBBB)
2. Goths

78

Reporting more than 1 categorical predictor: Estimated Marginal Means (EMMs)

The first table we look at is below, this tells us that we don’t need the interaction (interactions are explained in more detail in the LM1 and 2 workshops). So let’s rerun it without.

```
             Df   Sum Sq  Mean Sq    F value Pr(>F)    
hair          3 41701428 13900476 1.3958e+05 <2e-16 ***
sun           1 53843550 53843550 5.4065e+05 <2e-16 ***
hair:sun      3       48       16 1.5910e-01 0.9238    
Residuals   392    39039      100                      
```

Main Effects Model ANOVA table. Shows there is strong evidence that both predictors are associated with # of freckles since p<2.2e-16

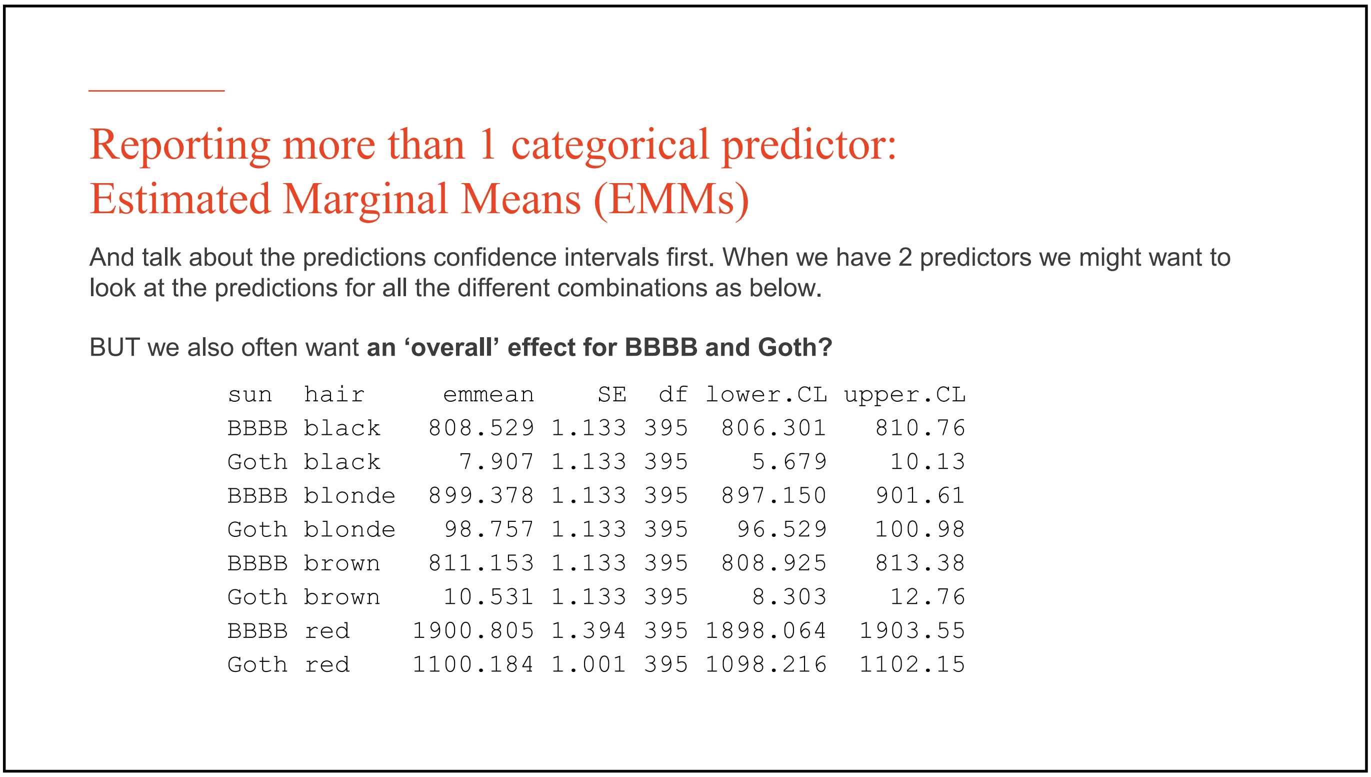
```
           Df   Sum Sq  Mean Sq F value    Pr(>F)    
hair        3 41701428 13900476  140474 < 2.2e-16 ***
sun         1 53843550 53843550  544129 < 2.2e-16 ***
Residuals 395    39087       99                      
```

Reporting more than 1 categorical predictor: Estimated Marginal Means (EMMs)

So let’s look at the parameters. And now we may run into a bit of a problem interpreting them.

Things are a little more complicated now…. So let’s come back to that.

```
             Estimate Std. Error  t value Pr(>|t|)    
(Intercept)  808.529      1.133  713.519   <2e-16 ***
hairblonde    90.850      1.407   64.579   <2e-16 ***
hairbrown      2.624      1.407    1.865   0.0629 .  
hairred     1092.276      1.472  741.903   <2e-16 ***
sunGoth     -800.621      1.085 -737.651   <2e-16 ***
```



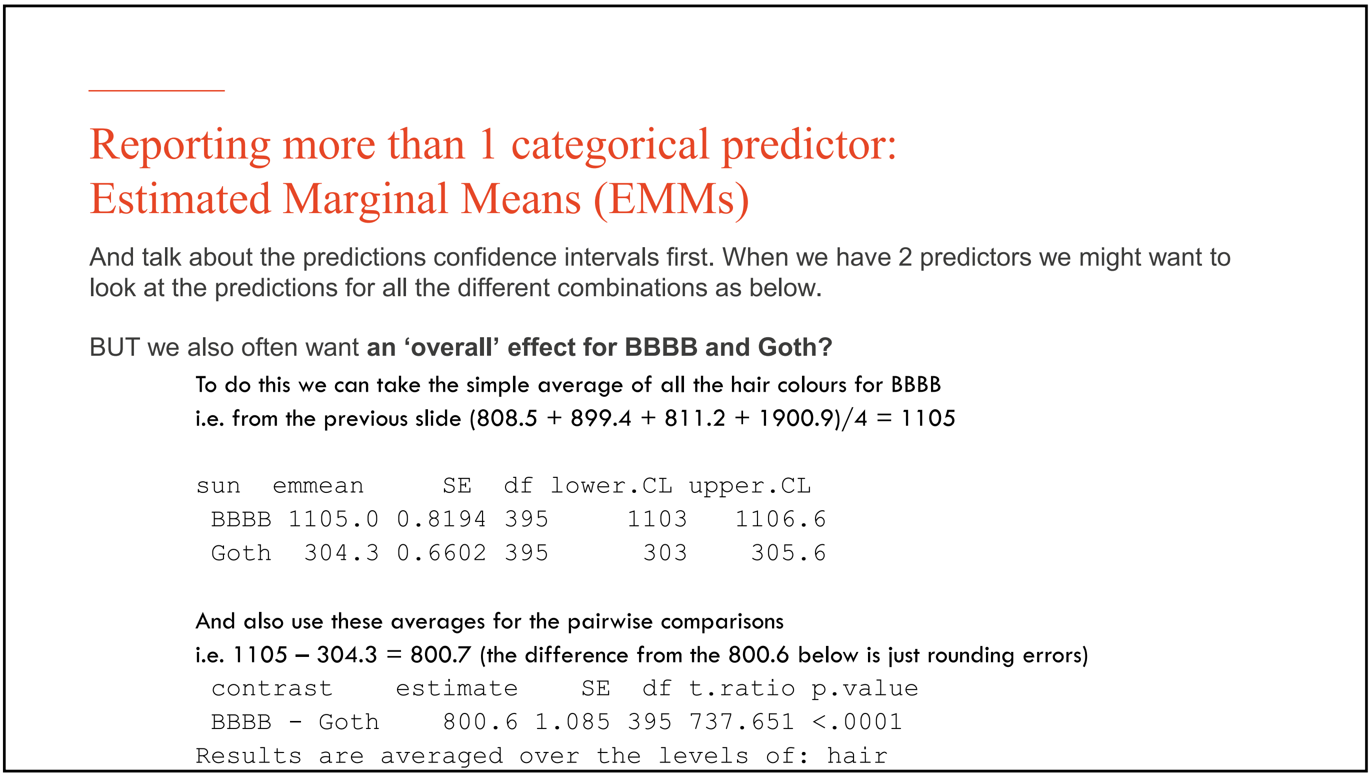
Reporting more than 1 categorical predictor: Estimated Marginal Means (EMMs)

And talk about the predictions confidence intervals first. When we have 2 predictors we might want to look at the predictions for all the different combinations as below.

BUT we also often want **an 'overall' effect for BBBB and Goth?**

```
sun  hair      emmean    SE  df lower.CL upper.CL
BBBB black   808.529 1.133 395  806.301   810.76
Goth black     7.907 1.133 395    5.679    10.13
BBBB blonde  899.378 1.133 395  897.150   901.61
Goth blonde   98.757 1.133 395   96.529   100.98
BBBB brown   811.153 1.133 395  808.925   813.38
Goth brown    10.531 1.133 395    8.303    12.76
BBBB red    1900.805 1.394 395 1898.064  1903.55
Goth red    1100.184 1.001 395 1098.216  1102.15
```

81



Reporting more than 1 categorical predictor: Estimated Marginal Means (EMMs)

And talk about the predictions confidence intervals first. When we have 2 predictors we might want to look at the predictions for all the different combinations as below.

BUT we also often want **an 'overall' effect for BBBB and Goth?**

To do this we can take the simple average of all the hair colours for BBBB
i.e. from the previous slide (808.5 + 899.4 + 811.2 + 1900.9)/4 = 1105

```
sun  emmean     SE  df lower.CL upper.CL
 BBBB 1105.0 0.8194 395     1103   1106.6
 Goth  304.3 0.6602 395      303    305.6
```

And also use these averages for the pairwise comparisons
i.e. 1105 – 304.3 = 800.7 (the difference from the 800.6 below is just rounding errors)

```
 contrast    estimate    SE  df t.ratio p.value
 BBBB - Goth    800.6 1.085 395 737.651 <.0001
Results are averaged over the levels of: hair
```

82

Reporting more than 1 categorical predictor: Estimated Marginal Means (EMMs)

We calculate the overall effect of hair colours in a similar way, we just average over BBBB and Goth.

```
 hair    emmean     SE  df lower.CL upper.CL
 black    408.2 0.9948 395    406.3    410.2
 blonde   499.1 0.9948 395    497.1    501.0
 brown    410.8 0.9948 395    408.9    412.8
 red     1500.5 1.0854 395   1498.4   1502.6
```

And use these averages for the pairwise comparisons

```
  contrast        estimate    SE  df t.ratio  p.value
 black - blonde   -90.850 1.407 395  -64.579 <.0001
 black - brown     -2.624 1.407 395   -1.865 0.2448
 black - red    -1092.276 1.472 395 -741.903 <.0001
 blonde - brown    88.226 1.407 395   62.714 <.0001
 blonde - red   -1001.427 1.472 395 -680.196 <.0001
 brown - red    -1089.652 1.472 395 -740.121 <.0001
Results are averaged over the levels of: sun
P value adjustment: tukey method for comparing a family of 4 estimates
```

But the EMM is different to the data's mean. And that's why we use model averages not data averages.

One might expect a good model to replicate the data, right? A naïve person might think the best estimate for the # of freckles a redhead has is to average the number of freckles from our sample.

So why then does the EMM for red hair differ so much from the data average??

It's because the sample size is skewed towards Goths, if we take the average EMM for Red-Goth and Red-BBBB and weight it by the sample size we get the Data Average = 1100 * 0.9 + 1901*0.1 = 1180.

So an EMM let's us **remove the effect of our sample and get a clean read assuming all categories had equal sample size.**

Average Freckles	EMM	Data Average
Black	408	408
Blonde	499	499
Brown	411	411
Red	**1501**	**1180**

Sample Size	BBB	Goths
Black	50	50
Blonde	50	50
Brown	50	50
Red	**10**	**90**

But the EMM is different to the data's mean. And that's why we use model averages not data averages.

One might expect a good model to replicate the data, right? A naïve person might think the best estimate for the # of freckles a redhead has is to average the number of freckles from our sample.

So why then does the EMM for red hair differ so much from the data average??

It's because the sample size is skewed towards Goths, if we take the average EMM for Red-Goth and Red-BBBB and weight it by the sample size we get the Data Average = 1100 * 0.9 + 1901*0.1 = 1180.

So an EMM let's us **remove the effect of our sample and get a clean read assuming all categories had equal sample size.**

Which is a **good thing if our sample is not a good representation of the overall population.** In this instance it would have made it look like being a red head didn't have as much impact on freckles as it does.

But a bad thing if our sample does represent the population. Which is why EMMs can be weighted using different inputs.

85

EMMs can also incorporate continuous variables

There are a number of ways to do but we usually include it's contribution to the prediction at a single point, often it's average Other options are:

- A different value for each contributing category (often the average for that category). e.g. if we added age to our example we might use a different age for each hair*sun combination (its specific average) rather than the overall average.

86

Examples of when EMMs are better than data averages

We want to estimate the impact of a predictor such as a new medical treatment, **after removing the effects of other covariates**. Particularly useful if the covariate distribution in our sample data doesn't match the population. (We did this when estimating the average # of freckles by Hair Colour, after correcting for Sun.)

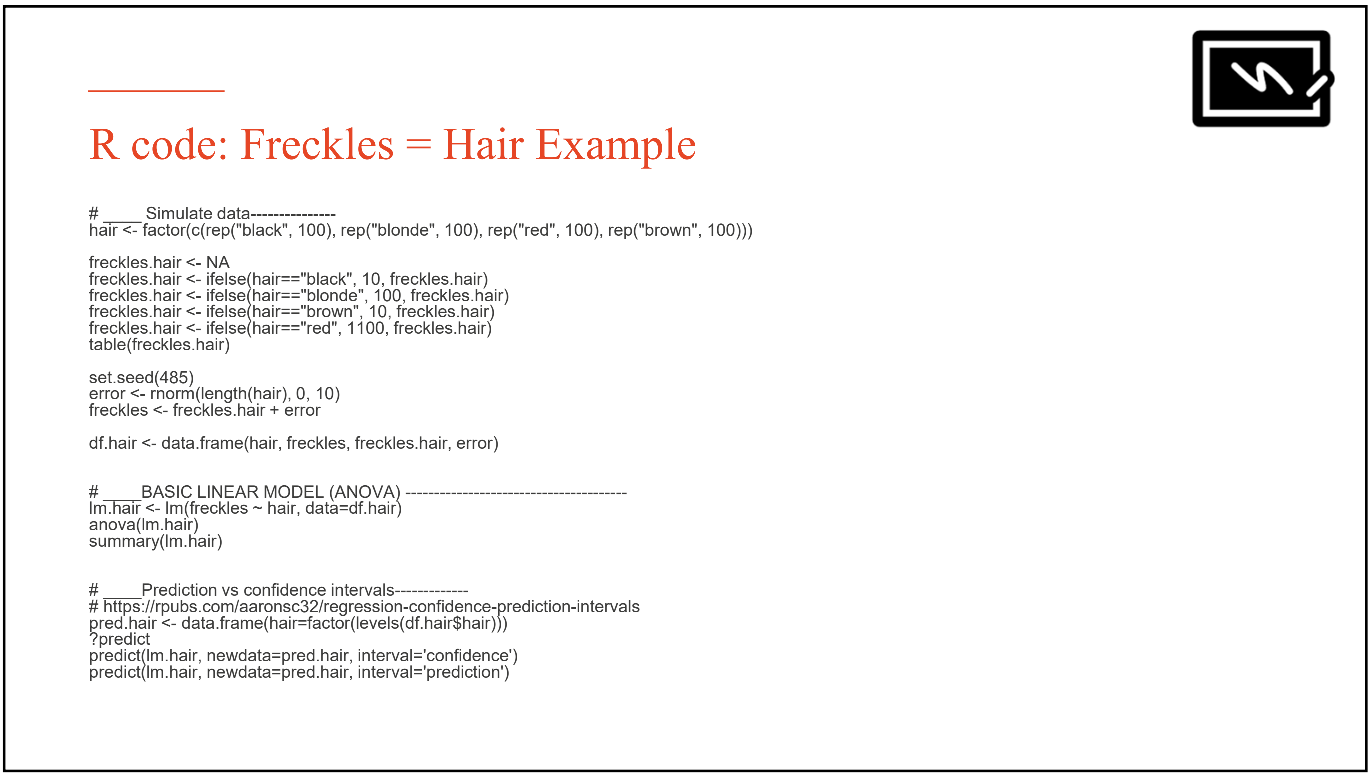
We want to estimate the response **in the general population** by weighting/setting predictors to their expected population proportions.

We want to ***account for an unbalanced data set***.

We want to ***account for continuous covariates.***

References

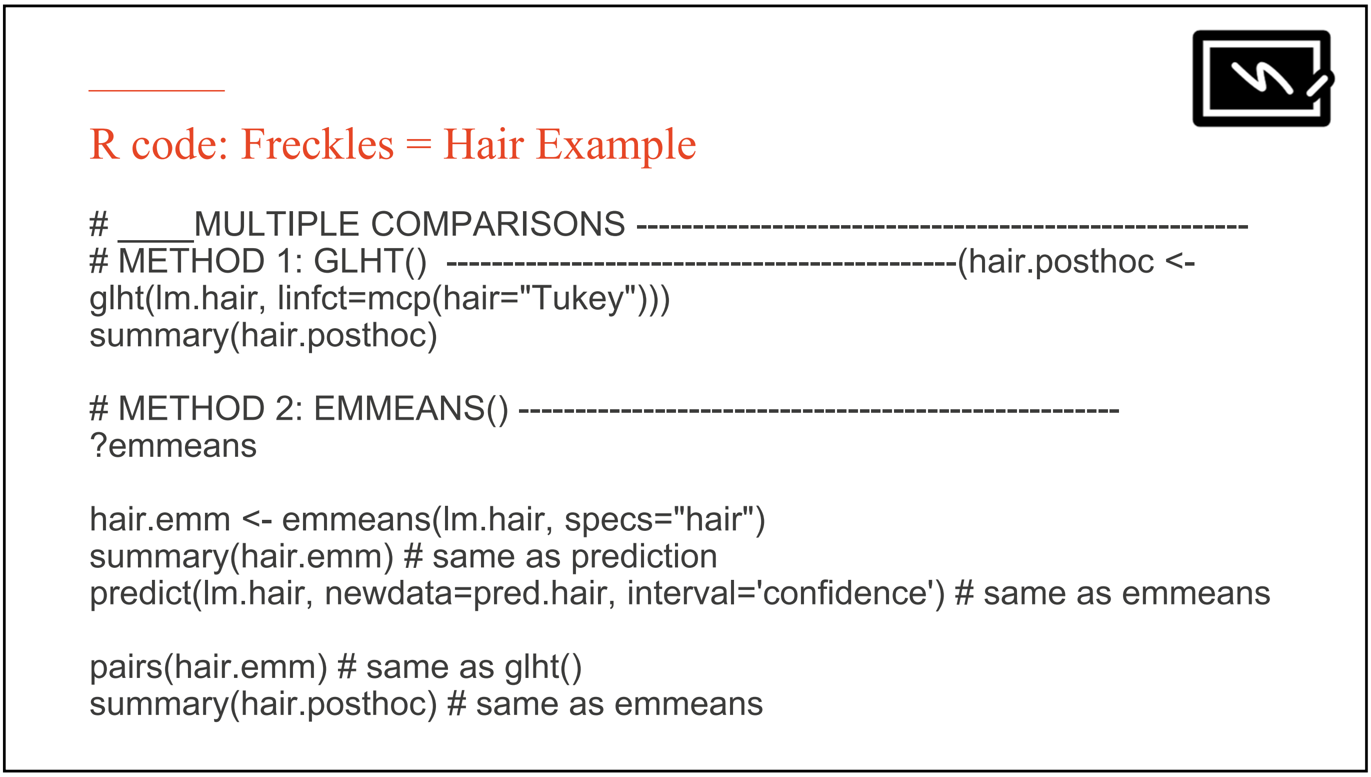
- Vignettes from the R package emmeans https://cran.r-project.org/web/packages/emmeans/index.html



R code: Freckles = Hair Example

```
# ____ Simulate data---------------
hair <- factor(c(rep("black", 100), rep("blonde", 100), rep("red", 100), rep("brown", 100)))

freckles.hair <- NA
freckles.hair <- ifelse(hair=="black", 10, freckles.hair)
freckles.hair <- ifelse(hair=="blonde", 100, freckles.hair)
freckles.hair <- ifelse(hair=="brown", 10, freckles.hair)
freckles.hair <- ifelse(hair=="red", 1100, freckles.hair)
table(freckles.hair)

set.seed(485)
error <- rnorm(length(hair), 0, 10)
freckles <- freckles.hair + error

df.hair <- data.frame(hair, freckles, freckles.hair, error)


# ____BASIC LINEAR MODEL (ANOVA) ---------------------------------------
lm.hair <- lm(freckles ~ hair, data=df.hair)
anova(lm.hair)
summary(lm.hair)


# ____Prediction vs confidence intervals-------------
# https://rpubs.com/aaronsc32/regression-confidence-prediction-intervals
pred.hair <- data.frame(hair=factor(levels(df.hair$hair)))
?predict
predict(lm.hair, newdata=pred.hair, interval='confidence')
predict(lm.hair, newdata=pred.hair, interval='prediction')
```

89



R code: Freckles = Hair Example

```
# ____MULTIPLE COMPARISONS -----------------------------------------------------
# METHOD 1: GLHT()  --------------------------------------------(hair.posthoc <-
glht(lm.hair, linfct=mcp(hair="Tukey")))
summary(hair.posthoc)

# METHOD 2: EMMEANS() ------------------------------------------------------
?emmeans

hair.emm <- emmeans(lm.hair, specs="hair")
summary(hair.emm) # same as prediction
predict(lm.hair, newdata=pred.hair, interval='confidence') # same as emmeans

pairs(hair.emm) # same as glht()
summary(hair.posthoc) # same as emmeans
```

90

91

Parametrising the Model

92

What does Parametrising the Model mean?

All **linear models have a set of parameters that need to be defined** for the software to estimate our model and **give us the knowledge that we seek** e.g. fixed effects parameters in the design matrix, random part of the model if there is one, distribution (normal, poisson, binomial, etc)

1 of the most basic are the parameters in the equation and design matrix. There is often **more than 1 way to define and calculate these parameters**. How we do so determines how we interpret the parameters we get at the end.

Which influences how we interpret and report our results.

And the knowledge we get from our analysis.

93

Simple Regression – Numeric Statistical Model

Quick Refresher from Linear Models 2

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

Prediction = Linear Predictor + 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
3	3	9	1	9	2.1	0.9
4	2	1	1	1	2.9	-0.9
5	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

B_o ~ Constant/Y intercept parameter

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

94

The Design Matrix is an important part of our model Parametrisation

It defines the **fixed effects** part of our model Parametrisation

And is directly used in the software's calculations

Design Matrix Parameters	
X0i	X1i
1	4
1	8
1	1
1	9
1	1
1	7

Multiple Regression Parametrisation

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

Prediction = Linear Predictor + Error/Natural Variation

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

Just keep going!!

Data					Design Matrix Parameters				Model Variables	
	Predictors									
Obs	Response	Continuous	Continuous	Continuous					Prediction	Error
i	Yi	X1i	X2i	X3i	X0i	X1i	X2i	X3i	$\hat{Y}i$	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
3	3	9	96	96	1	9	96	96	2.9	0.1
4	2	1	41	41	1	1	41	41	1.8	0.2
5	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

B_o ~ Constant/Y intercept parameter

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

β_{2i} ~ parameter for predictor 2

β_{3i} ~ parameter for predictor 3

No Intercept Parametrisation

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

Prediction = Linear Predictor + Error/Natural Variation

Forces the line through the origin. Can be useful if you know this should happen.

Data						Design Matrix Parameters				Model Variables	
		Predictors									
Obs	Response	Continuous	Continuous	Continuous						Prediction	Error
i	Yi	X1i	X2i	X3i			X1i	X2i	X3i	$\hat{Y}i$	εi
1	4	4	12	12			4	12	12	4.2	-0.2
2	4	8	54	54			8	54	54	4.3	-0.3
3	6	1	87	87			1	87	87	5.3	0.7
3	3	9	96	96			9	96	96	2.9	0.1
4	2	1	41	41			1	41	41	1.8	0.2
5	2	7	47	47			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} ~ Removed

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

B_o ~ Removed

β_{1i} ~ parameter for predictor 1

β_{2i} ~ parameter for predictor 2

β_{3i} ~ parameter for

Other important parts of Model Parametrisation

Equation

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

Is usually defined in the software e.g.

```
R> lm(response ~ predictor, data=data)
```

Note that the `~ predictor` defines the design matrix

Other important parts of Model Parametrisation

Transformations on the response and predictors e.g.

$\text{Log}(Y_i) = \beta_o X_{0i} + \beta_1 X_{1i} + \varepsilon_i$

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

There are generally 2 ways to do this:

1. Use the raw variable and include the transformation in the model equation e.g.
```
R> lm(log(response)~predictor, data=data)
```
   - Usually the **preferred option** since doing it within the equation modelled means the software knows the response has been transformed and can pass this on to other functions, such as emmeans() in R.
2. Transform the variable and include it in the model equation e.g.
```
R> log.response <- log(response)
R> lm(log.response ~predictor, data=data)
```

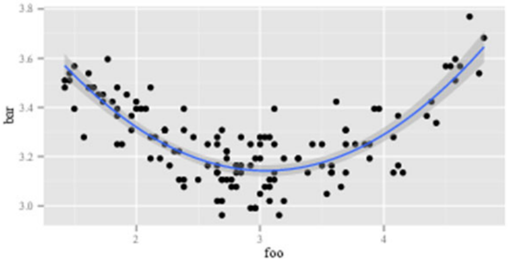
Other important parts of Model Parametrisation

Quadratic and other functions e.g.

```
lm(log.response~predictor+I(predictor^2), data=data)
```

This above uses the raw variable and tells the equation to square it in the design matrix. Which as mentioned in the previous slide is generally preferred over calculating it and entering the squared variable beforehand i.e.

```
R> predictor.sq <- predictor^2
R> lm(response ~ p                                    =data)
```



Model Building has more info on this

Other important parts of Model Parametrisation

General Linear Mixed Models

- The **link** function (this is where we would usually transform the response rather than log it beforehand)
- The **distribution** e.g. normal, poisson, binomial, etc

Mixed Models

Need to define the random effects parameters e.g. this example defines a nested design with id nested in class. And a fixed effect design matrix of 2 parameters: intercept and time. The response is score.

```
R>lme(data=mixed.int3, fixed=score~time, random= ~ 1|class/id)
```

101

Categorical Predictor Interpretation

Is particularly influenced by the type of parametrisation used.

Recall our freckles = Hair + Sun model

Below are the results we get which I said we'd come back to.

In order to interpret it we need to recognise that it used **Dummy Coding parametrisation with Black haired BBBB's as the reference.**

```
                                                Estimate Std. Error
t value Pr(>|t|)
(Intercept)  808.529      1.133  713.519   <2e-16 ***
hairblonde    90.850      1.407   64.579   <2e-16 ***
hairbrown      2.624      1.407    1.865   0.0629 .
hairred     1092.276      1.472  741.903   <2e-16 ***
sunGoth     -800.621      1.085 -737.651   <2e-16 ***
```

102

Categorical Variables (e.g. ANOVA)

Quick Refresher from Linear Models 2

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

Prediction = Linear Predictor + Error/Natural Variation

Data				Design Matrix Parameters			Model Variables	
	Predictors							
Obs	Response	Continuous	Categorical				Prediction	Error
i	Yi	X1i	X2i	X0i	X1i	X2i	$\hat{Y}i$	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
3	3	9	Smoking	1	9	1	3.4	-0.4
4	2	1	Non Smoking	1	1	0	1.4	0.6
5	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

B_o ~ (Reference group = Non-Smoking)

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

β_{2i} ~ parameter for smoking

103

Categorical Variables (e.g. ANOVA)

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

Prediction = Linear Predictor + Error/Natural Variation

Data				Design Matrix Parameters			Model Variables	
	Predictors							
Obs	Response	Continuous	Categorical				Prediction	Error
i	Yi	X1i	X2i	X0i	X1i	X2i	$\hat{Y}i$	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
3	3	9	Smoking	1	9	1	3.4	-0.4
4	2	1	Non Smoking	1	1	0	1.4	0.6
5	2	7	Non Smoking	1	7	0	2.2	-0.2

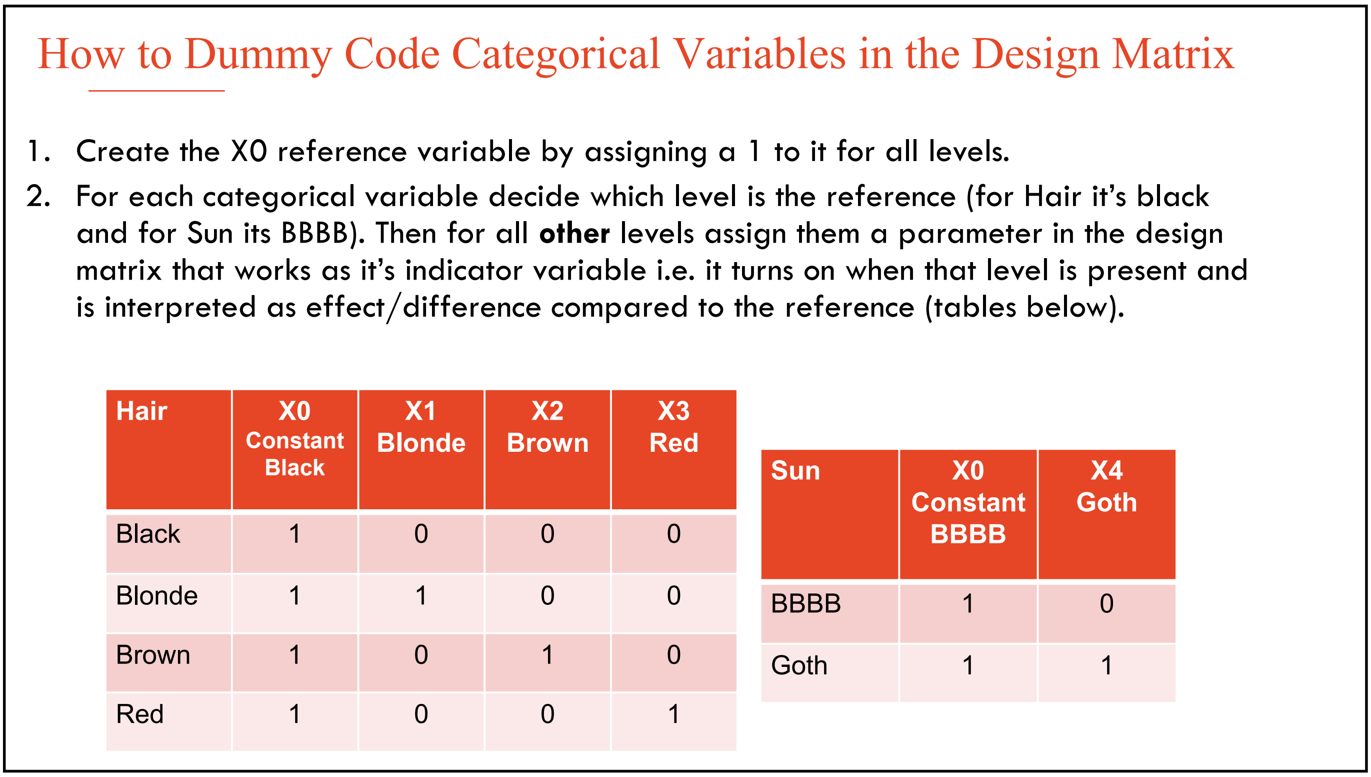
There are many different **parameterisations** (ways) to add categorical variables. The way I am showing you is called **Dummy** or **Treatment Coding.** Linear Models 3 discusses other ways such as effects coding.

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

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

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

104



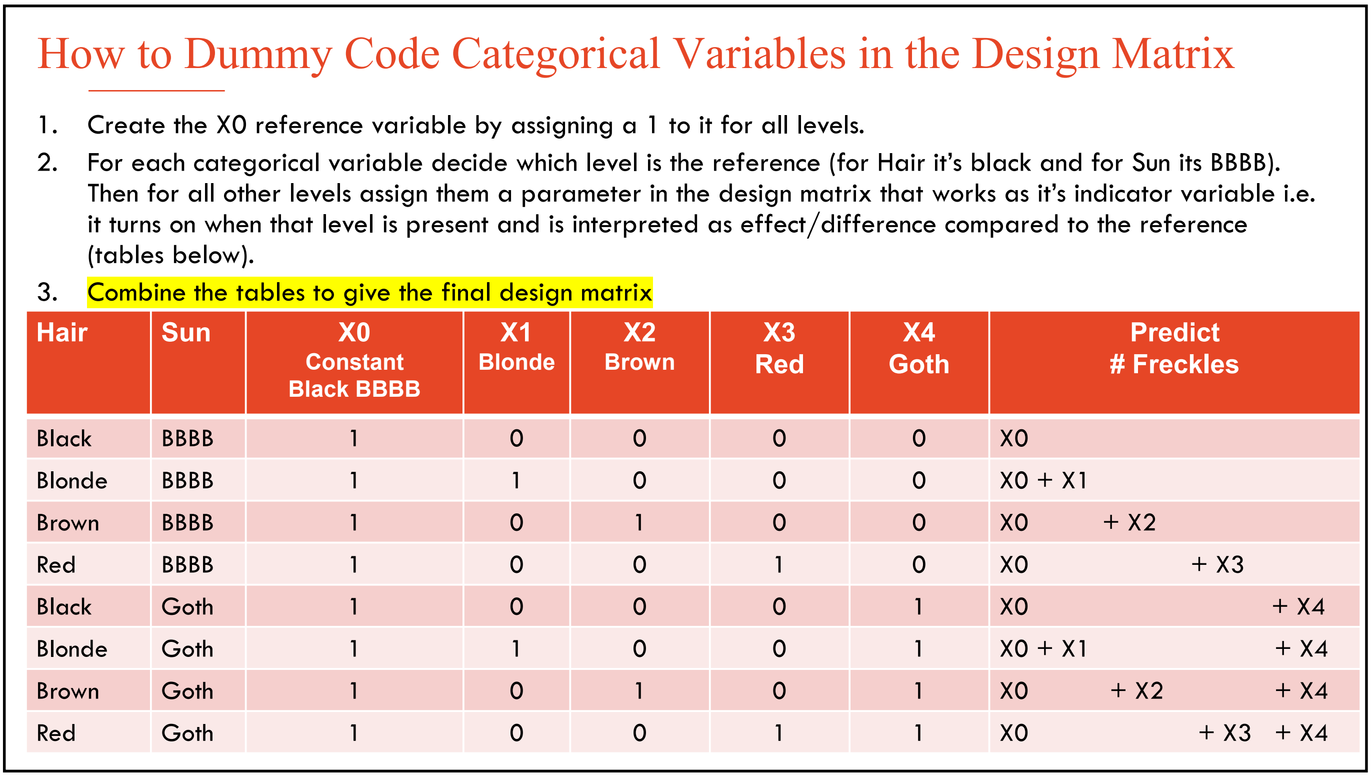
How to Dummy Code Categorical Variables in the Design Matrix

1. Create the X0 reference variable by assigning a 1 to it for all levels.
2. For each categorical variable decide which level is the reference (for Hair it's black and for Sun its BBBB). Then for all **other** levels assign them a parameter in the design matrix that works as it's indicator variable i.e. it turns on when that level is present and is interpreted as effect/difference compared to the reference (tables below).

Hair	X0 Constant Black	X1 Blonde	X2 Brown	X3 Red
Black	1	0	0	0
Blonde	1	1	0	0
Brown	1	0	1	0
Red	1	0	0	1

Sun	X0 Constant BBBB	X4 Goth
BBBB	1	0
Goth	1	1

105

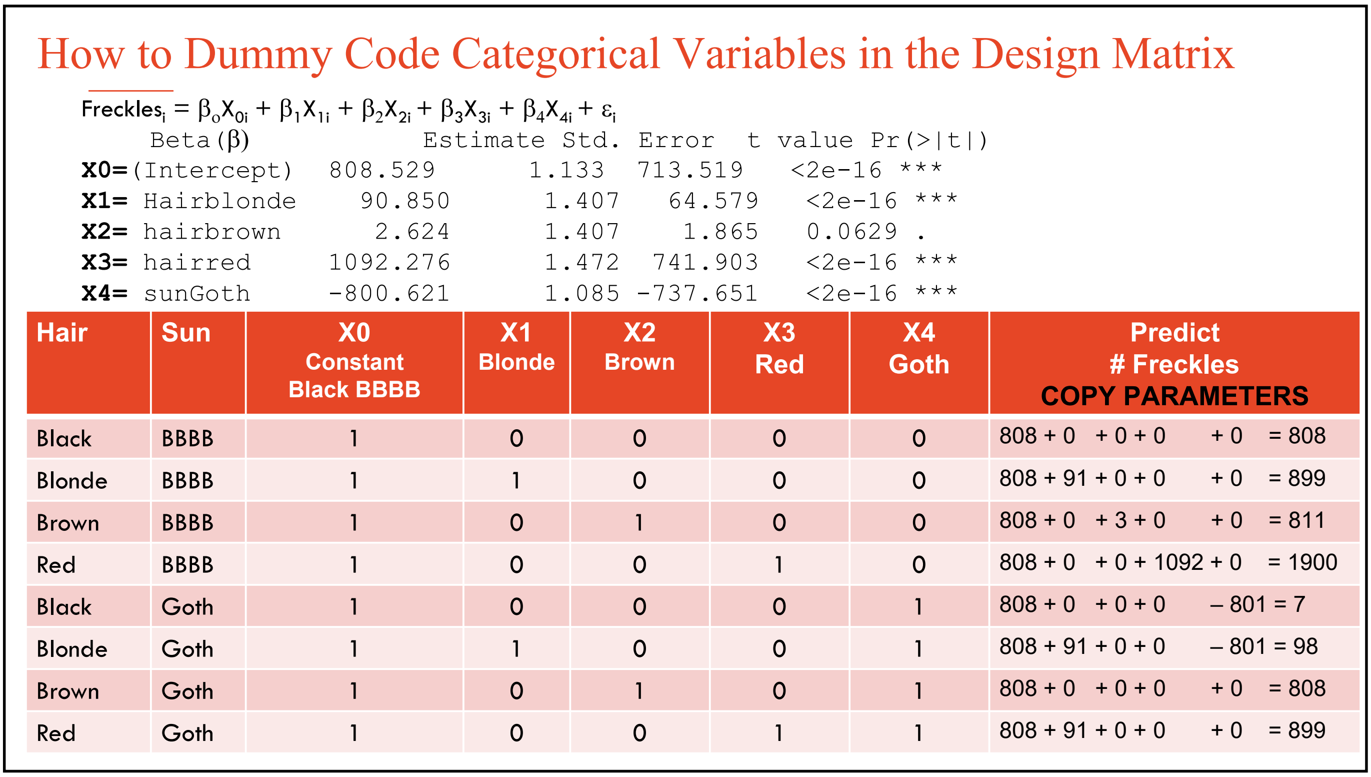


How to Dummy Code Categorical Variables in the Design Matrix

1. Create the X0 reference variable by assigning a 1 to it for all levels.
2. For each categorical variable decide which level is the reference (for Hair it's black and for Sun its BBBB). Then for all other levels assign them a parameter in the design matrix that works as it's indicator variable i.e. it turns on when that level is present and is interpreted as effect/difference compared to the reference (tables below).
3. Combine the tables to give the final design matrix

Hair	Sun	X0 Constant Black BBBB	X1 Blonde	X2 Brown	X3 Red	X4 Goth	Predict # Freckles
Black	BBBB	1	0	0	0	0	X0
Blonde	BBBB	1	1	0	0	0	X0 + X1
Brown	BBBB	1	0	1	0	0	X0 + X2
Red	BBBB	1	0	0	1	0	X0 + X3
Black	Goth	1	0	0	0	1	X0 + X4
Blonde	Goth	1	1	0	0	1	X0 + X1 + X4
Brown	Goth	1	0	1	0	1	X0 + X2 + X4
Red	Goth	1	0	0	1	1	X0 + X3 + X4

106

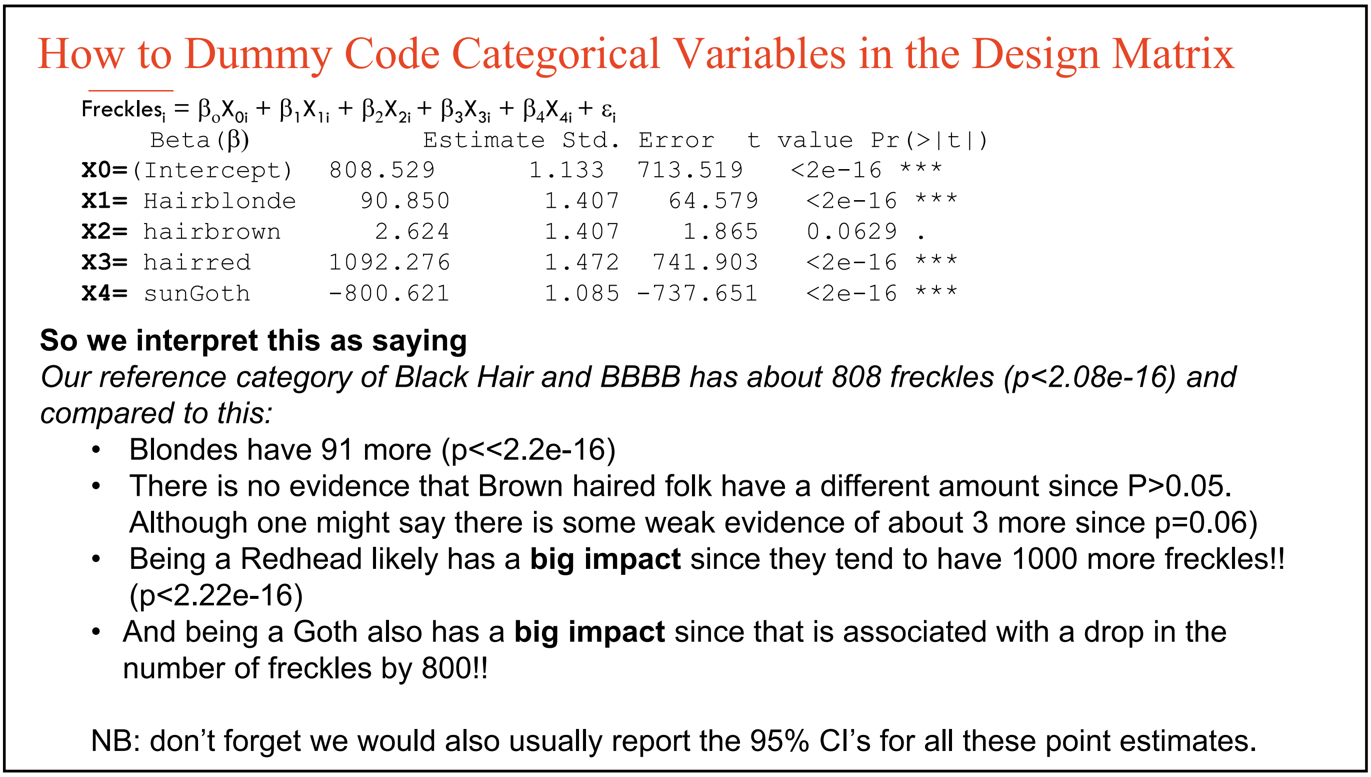


How to Dummy Code Categorical Variables in the Design Matrix

$Freckles_i = \beta_o X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \varepsilon_i$

```
         Beta(β)              Estimate Std. Error  t value Pr(>|t|)
X0=(Intercept)  808.529      1.133  713.519   <2e-16 ***
X1= Hairblonde    90.850      1.407   64.579   <2e-16 ***
X2= hairbrown      2.624      1.407    1.865   0.0629 .
X3= hairred     1092.276      1.472  741.903   <2e-16 ***
X4= sunGoth     -800.621      1.085 -737.651   <2e-16 ***
```

Hair	Sun	X0 Constant Black BBBB	X1 Blonde	X2 Brown	X3 Red	X4 Goth	Predict # Freckles COPY PARAMETERS
Black	BBBB	1	0	0	0	0	808 + 0 + 0 + 0 + 0 = 808
Blonde	BBBB	1	1	0	0	0	808 + 91 + 0 + 0 + 0 = 899
Brown	BBBB	1	0	1	0	0	808 + 0 + 3 + 0 + 0 = 811
Red	BBBB	1	0	0	1	0	808 + 0 + 0 + 1092 + 0 = 1900
Black	Goth	1	0	0	0	1	808 + 0 + 0 + 0 – 801 = 7
Blonde	Goth	1	1	0	0	1	808 + 91 + 0 + 0 – 801 = 98
Brown	Goth	1	0	1	0	1	808 + 0 + 0 + 0 + 0 = 808
Red	Goth	1	0	0	1	1	808 + 91 + 0 + 0 + 0 = 899



How to Dummy Code Categorical Variables in the Design Matrix

$Freckles_i = \beta_o X_{0i} + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \varepsilon_i$

```
         Beta(β)              Estimate Std. Error  t value Pr(>|t|)
X0=(Intercept)  808.529      1.133  713.519   <2e-16 ***
X1= Hairblonde    90.850      1.407   64.579   <2e-16 ***
X2= hairbrown      2.624      1.407    1.865   0.0629 .
X3= hairred     1092.276      1.472  741.903   <2e-16 ***
X4= sunGoth     -800.621      1.085 -737.651   <2e-16 ***
```

So we interpret this as saying

Our reference category of Black Hair and BBBB has about 808 freckles (p<2.08e-16) and compared to this:

- Blondes have 91 more (p<<2.2e-16)
- There is no evidence that Brown haired folk have a different amount since P>0.05. Although one might say there is some weak evidence of about 3 more since p=0.06)
- Being a Redhead likely has a **big impact** since they tend to have 1000 more freckles!! (p<2.22e-16)
- And being a Goth also has a **big impact** since that is associated with a drop in the number of freckles by 800!!

NB: don’t forget we would also usually report the 95% CI’s for all these point estimates.

Parameters and their interpretation are framed against an arbitrary reference level. Let’s look at the impact of changing Sun’s reference level from BBBB to Goth

Changing the reference level changes the way we look at the data. It doesn’t change the overall interpretation but it does change our focus which makes answering specific Research Questions easier or harder.

Reference is Hair:Black, Sun:BBBB.
This parametrisation suggests that **Goths reduce** the # of freckles by 800

```
            Beta(β)      Estimate Std. Error  t value Pr(>|t|)
Intercept)  808.529       1.133  713.519   <2e-16 ***
Hairblonde    90.850       1.407   64.579   <2e-16 ***
hairbrown      2.624       1.407    1.865   0.0629 .
hairred     1092.276       1.472  741.903   <2e-16 ***
sunGoth     -800.621       1.085 -737.651   <2e-16 ***
```

Reference is Hair:Black, Sun:Goth
This parametrisation suggests that **BBBB’s increase** the # of freckles by 800.
The **overall effect is the same**, but we are just looking at it from a different angle. And maybe one that is **more relevant to our research question?**

```
            Beta(β)      Estimate Std. Error  t value Pr(>|t|)
(Intercept)    7.907       1.133   6.978 1.27e-11 ***
hairblonde    90.850       1.407  64.579  < 2e-16 ***
hairbrown      2.624       1.407   1.865   0.0629 .
hairred     1092.276       1.472 741.903  < 2e-16 ***
sunBBBB      800.621       1.085 737.651  < 2e-16 ***
```

109

Parameters and their interpretation are framed against an arbitrary reference level. Let’s look at the impact of changing Hairs reference level from Black to Red

If we wanted to focus on the difference compared to redheads then let’s make them the reference level. BUT notice how this changes our interpretation!

Reference is Hair:Black, Sun:Goth

```
            Beta(β)      Estimate Std. Error  t value Pr(>|t|)
Intercept)  808.529       1.133  713.519   <2e-16 ***
Hairblonde    90.850       1.407   64.579   <2e-16 ***
hairbrown      2.624       1.407    1.865   0.0629 .
hairred     1092.276       1.472  741.903   <2e-16 ***
sunGoth     -800.621       1.085 -737.651   <2e-16 ***
```

Its common to make the control the reference level. Since then its easy to understand how treatments differ from it.

Reference is Hair:Redhead, Sun:Goth
By focusing on redheads we see some changes. All the hair parameters:
- are now strongly significant <2e-16
- and have negative effects

The **overall effect is the same**, but we are just looking at it from a different angle. And maybe one that is **more relevant to our research question?**

Estimate Std. Error t value Pr(>|t|)

```
            Beta(β)      Estimate Std. Error  t value Pr(>|t|)
(Intercept)  1900.805      1.394   1363.4    <2e-16 ***
hairblack   -1092.276      1.472   -741.9    <2e-16 ***
hairblonde  -1001.427      1.472   -680.2    <2e-16 ***
hairbrown   -1089.652      1.472   -740.1    <2e-16 ***
sunGoth      -800.621      1.085   -737.7    <2e-16 ***
```

110

The family wise ANOVA table never changes though!

Since the model is the same, we've just changed how the categorical variable is parametrised

```
> # Reference level is HAIR:Black & SUN:BBBB
> anova(lm.hair.sun)
Df    Sum Sq  Mean Sq F value    Pr(>F)
hair        3 41701428 13900476  140474 < 2.2e-16 ***
sun         1 53843550 53843550  544129 < 2.2e-16 ***
Residuals 395    39087       99

> # Reference level is HAIR:Black & SUN:Goth
> anova(lm.hair.sun3.0)
Df    Sum Sq  Mean Sq F value    Pr(>F)
hair        3 41701428 13900476  140474 < 2.2e-16 ***
sun         1 53843550 53843550  544129 < 2.2e-16 ***
Residuals 395    39087       99

> # Reference level is HAIR:red & SUN:BBBB
> anova(lm.hair.sun4.0)
Df    Sum Sq  Mean Sq F value    Pr(>F)
hair        3 41701428 13900476  140474 < 2.2e-16 ***
sun         1 53843550 53843550  544129 < 2.2e-16 ***
Residuals 395    39087       99
```

Common ways to Parametrise Categorical Variables

Dummy Coding/Treatment coding

- Useful when we have a control or some natural reference group we want to compare other treatment levels to since each parameter is interpreted as the difference from this control/reference group.
- Most common.
- Constant by itself represents the base reference level for all factors.
- Can't calculate an effect for each level since the reference level for each factor is confounded with all the other reference levels.

Effects Coding

- Useful if there is no natural reference group since we can calculate the effect of each level. So likely better for our freckles example.
- Constant by itself represents the 'grand mean' which is the average effect overall factor levels.
- Each parameter is that levels change from the 'grand mean'. The missing level can be calculated from the other levels.

Estimated Marginal Means

- Have to some extent made different parametrisations obsolete.
- That said there are still situations where a specific form of parametrisation is useful e.g. when you want to use them to test a specific hypothesis.

113

Reporting complex nonlinear effects

114

Reporting complex nonlinear effects

These comments refer to the plot on the next slide:

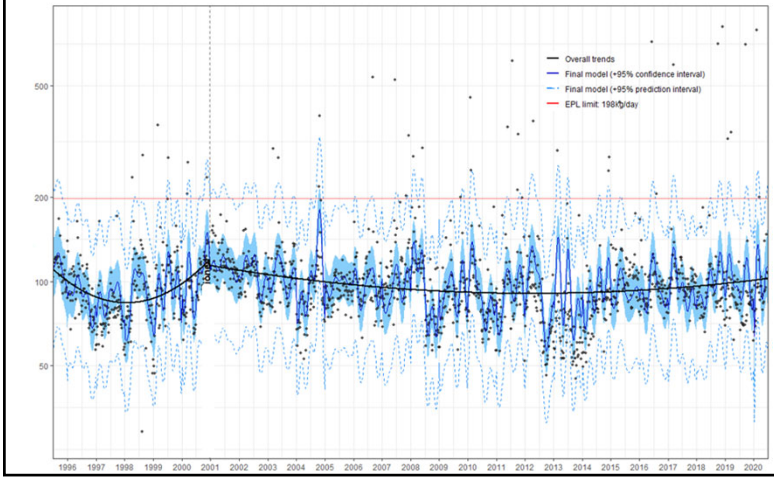
- This is a model I built to capture the effect of a water treatment plant upgrade on a analyte of interest (such as nitrogen, phosphorus, dissolved oxygen, etc). The analyte is not shown as the analysis was confidential.
- The horizontal dotted line is the plant upgrade.
- The green line captures the overall trend.
- The blue line factors in seasonal trends.
- Notice the difference between the confidence and prediction intervals.
  - **Confidence interval** is where we expect the modelled average i.e. the blue line, to be after considering sample variance and model uncertainty.
  - **Prediction interval** is where we expect the individual samples to be i.e. the grey points. It's 95% so we do expect some to be outside of it as we have more than 100 samples. They can be used when we want to predict an actual point in the future to give us the range we expect such a prediction to be in, rather than the average. And can also tell us if an individual sample is within expected tolerances, or possibly behaving outside the model expectations. They are wider since they factor in the extra variance associated with a single observation, rather than the average of observations.

115

Reporting complex nonlinear effects

When reporting simple linear effects like ANOVA or regression a table representing the effect and it's CI is often sufficient. But when reporting non linear effects or complex models other options such as those below are easier:

- Plots (as shown below)
- Estimated Marginal Means (as the previous freckles/hair example shows)



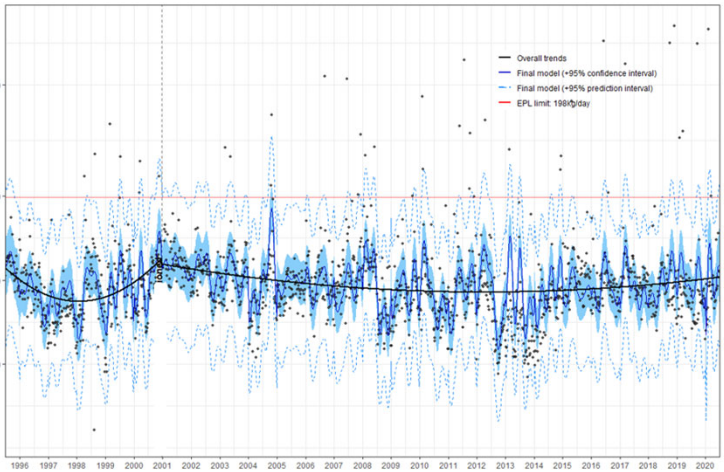
Notice how the 1st model predicts that without the plant upgrade there would have been exceedance problems in about 2 years.

116

Confidence Intervals vs Prediction Intervals

Confidence interval shows us the range of where we expect the overall model blue line to be.

Prediction interval is where we expect the individual samples to be. It’s 95% so we do expect some to be outside of it as we have more than 100 samples.



Notice how the 1st model predicts that without the plant upgrade there would have been exceedance problems in about 2 years.

117

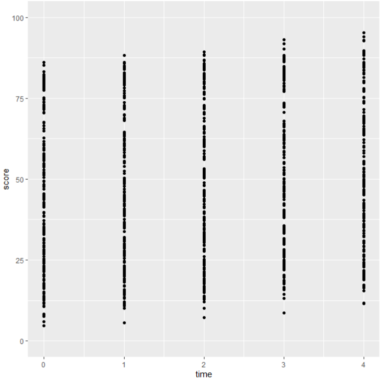
More on Mixed Models

118

Random Intercepts

Let’s say we wanted to understand the effect of teaching on some skill. And we had 5 classes with 40 people in each and 5 time points.

Here’s the data a normal regression would model.



Random Intercepts

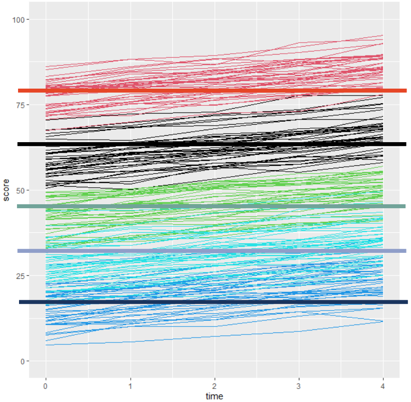
And here’s the data a mixed effects model that nests student in class would model.

Notice how there is more structure in this model. How it groups:

- Each students data together via the **lines**
- Each classes data together via the **colours**

A random intercept model factors this information in by

- **Adjusting** each **classes** intercept from the base B0 y intercept
  - Notice the 5 different lines, 1 for each class.
  - It then captures this adjustment by calculating the variance for these 5 points.
- **Adjusting** each **person’s** intercept from their classes y intercept
  - It then captures this adjustment by calculating the variance for each individuals adjustment.



“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*).

121

Including Random Effects: gives us more precise and sensitive models

Because they **increase the signal to noise ratio, by reducing the noise.** Which allows us to **detect smaller signals, with greater precision.**

They do this by partitioning out different types of variance/error/noise by adjusting the intercept for different parameters e.g. class and ID. We then capture this adjustment as a variance and remove it from the model.

This can often be the difference between finding publishable results or not. As the example in our LM1 workshop showed i.e. the fixed effect model did not detect a difference between treatments, while the mixed effect model did since by giving each patient a random intercept it removed the between patient noise/variance.

This is another reason why understanding and **developing a great Experimental Design is so important.** It allows us to identify and remove noise leading to better results. (Refer to our Experimental Design for more info).

NB: they are not always more accurate, in that the parameter estimates often stay the same. They are usually more ***precise*** though as their SE’s are usually reduced, leading to smaller p-values and narrower CI’s.

122

Including Random Effects: Understanding the relative source of variance/noise/error

Another benefit is that we get estimates of the different sources of variance. So in our example we can tell that **class accounted for about 5 times more difference** in the scores than individuals or the individuals change over time.

If we wanted to improve results this might prompt us to investigate why the classes are so different.

While if this were a quality control exercise such estimates are used to design better processes by determining which elements introduce the most difference from batch to batch.

	Variance Point Estimate	95% CI Lower Bound	95% CI Upper Bound
Difference **between Classes**	25	12	50
Difference **between Individual**	5	4.5	5.4
Error/noise/change/difference **within each Individual**	1	1.0	1.1

123

Including Random Effects: Answering Population Level Research Questions

If we had included **Class as a fixed effect** we can only answer the question if these 5 specific classes differ. It **tells us nothing about the wider population.**

But by including as a **random effect** we instead ask the **RQ: do all classes differ in the entire population**

And our answer is **yes, there is evidence they do.**

This is an often overlooked advantage of Random Effects.

	Variance Point Estimate	95% CI Lower Bound	95% CI Upper Bound
Difference **between Classes**	25	12	50
Difference **between Individual**	5	4.5	5.4
Error/noise/change/difference **within each Individual**	1	1.0	1.1

124

Including Random Effects: Answering Population Level Research Questions

It also tells us that the **variance between Classes has been poorly estimated** since the CI is so wide 12-50.

So **future studies that want to measure this effect more accurately should increase the number of classes.**

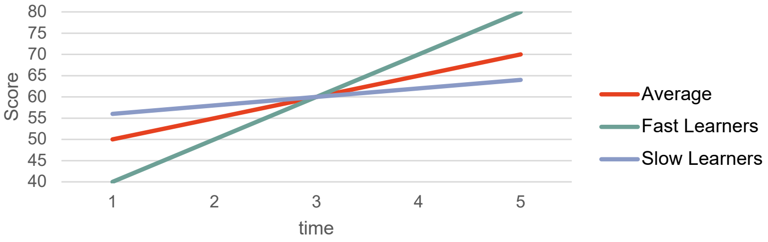
	Variance Point Estimate	95% CI Lower Bound	95% CI Upper Bound
Difference **between Classes**	25	12	50
Difference **between Individual**	5	4.5	5.4
Error/noise/change/difference **within each Individual**	1	1.0	1.1

125

Random Slopes

Are similar to random intercepts, except they allow the slope to differ for each individual.

Which is useful when we want to understand the overall 'average' trend over time, after accounting for the different learning abilities of students. Another way of putting this is that peoples learning differs not just in their error but systematically i.e. their slope differs from an underlying average slope.



126

Random Slopes: Answering Population Level Research Questions

Adding a Random Slope lets us test the **Population Level Question**

> There is little variation from the average trend **so most students learn at similar rates.**
>
> Vs
>
> There is a lot of variation from the average trend **which suggests students learn at quite different rates.** And perhaps this is worthy of further study to understand why, so we can apply these learnings to all students?

127

Using the same variable as a fixed and random effect

In general, you ***can't fit the exact same effect*** as both fixed and random. For example, you wouldn't fit *Person ID* as both a fixed and random effect since then you're fitting the difference from the overall mean for each person twice.

- Meaning the effect of each person has to be 'shared' over both their fixed effect parameter and their random effect BLUP (Best Linear Unbiased Predictor).
- Making neither an accurate estimate of each person difference from the overall mean and both being unstable (if the model converges at all, which it often won't).

But you can use the same variable in the fixed and random parts of the model to tell the software what model you want to fit, if each use represents a different effect. Be careful when doing this. Ensure you know exactly why you are doing it and what the model is you are fitting. **DO NOT just throw everything into a model and see what you get!**

128

Using the same variable as a fixed and random effect

A good example is from Karen Grace-Martin of the Analysis Factor https://www.theanalysisfactor.com/mixed-models-predictor-both-fixed-random/.

In her example she fits a population level regression line for the increase of jobs over time in a sample of different counties i.e. to test if the *average* effect of Time across all counties = 0. And also fits a ***random slope for each county*** to evaluate if they differ, or if they are all the same i.e. is the *individual* effect of Time for each County the same or is the variance of their slopes = 0.

County is the random variable, not Time. But we need Time in both the fixed and random parts of the syntax to correctly specify the model.

By fitting a line for each County it accounts for the expected correlation of jobs within a county e.g. big counties have more jobs than small ones. And makes the fixed effect the population effect of all these lines, rather than all the individual points.

To do this she has to include the variable Time in both the fixed and random part of the syntax so it fits the model as per the below R code:

```
model<-lme(JobsK~rural*time, random=~time|County,data=countylong, na.action=na.omit)
```

129

Further Reading: Linear Models I covers

- Introduction to random effects and mixed models
- Random intercepts

130

References

GLMM FAQ by Ben Bolker (can’t recommend this highly enough!! Just start here with any question before you even google it) https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html

131

1:2
5

132

Other Resources

133

Other resources

VIDEOS

- StatsQuest with Josh Starmer
- Zedstatistics, longer videos than StatsQuest

WEBSITES

- R GLMM FAQ, a great first place to look for any GLMM problems, even if not in R.

BOOKS AND PAPERS

- 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.

134

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

- Open Learning Environment (OLE) courses
- Linkedin Learning
  - SPSS Linear Models workflow

135

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!!”

136

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 **R in Action by Robert I Kabacoff** for a good intro that includes a lot of statistical methods
  - It also has a great web page resource which is a good first port of call.
  - Buy through website for a discount
- Only downside is that it doesn't use Hadley Wickhams Tidyverse packages, so I would also recommend one of his. In particular **R for Data Science by Hadley Wickham** gives a great intro to data wrangling and visualisation using his Tidyverse packages.
- Finally, I recommend **MASS (Modern Applied Statistics in S) by Veneables and Ripley**. The 'Yellow Bible'. It has at least a little bit on pretty much any statistical method you can think of. I tend to start here to get an intro on what R can do and then research outwards.

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

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.