Internship Summary

Alec McCallum

13/07/2020

Table of Contents

Week 1 4
Day 1 4
Morning 4
Course: Introduction to R 4
Afternoon 5
Key Takeaways 5
Day 2 5
Morning 5
Course: Introduction to R 5
Afternoon 6
Key Takeaways 6
Day 3 7
Morning 7
Course: Intro to Experimental Design 7
Key Takeaways 10
Day 4 11
Morning 11
Course: Intro to Experimental Design 11
Afternoon 14
Talk: Russel (Functions and Programming) 14
Key Takeaways 14
Day 5 14
Morning 14
Talk: Pete (Tidyverse) 14
Afternoon 15
Talk: Sam (Rmarkdown) 15
Euler Problem 1 15

Euler Problem 2 16
Key Takeaways 17
Week 2 17
Day 1 17
Morning 17
Meeting: Hotdesk 17
Afternoon 17
Talk: Pete (ggplot) 17
Euler Problem 4 19
Key Takeaways 20
Day 2 20
Morning 20
Meeting: Experiment Analysis 20
Afternoon 21
Talk: Wendy (Exact Permutations Test) 21
Euler Problem 5 21
Day 3 22
Morning 22
Talk: Mario (Bioinformatics) 22
Talk: Mexiuan (Honours) 22
Euler Problem 6 22
Day 4 23
Morning 23
Demonstration: Pete (Drones) 23
Afternoon 23
Talk: Enqi (Honours) 23
Day 5 23
Morning 23
Talk: Sam (CV and Website) 23
Afternoon 24
Week 3 24
Day 1 24
Morning 24
Talk: Beata (Genetic Association Analysis) 24

Afternoon 24
Talk: Paul (Experiences) 24
Euler Problem 9 25
Day 2 25
Morning 25
Afternoon 25
Euler Problem 8 26
Day 3 27
Morning 27
Afternoon 27
Meeting: Olena (Honours) 27
Day 4 27
Morning 27
Afternoon 27
Day 5 28
Morning 28
Afternoon 28
Presentation 28

Week 1

Day 1

Morning

Course: Introduction to R

Rstudio interface

Started off with an intro to the Rstudio interface

- R console
  - Where the script is executed
- Script window
  - For typing in the script/code
- Environment/history window
  - Environment shows all the variables and vectors that have been created in the session
  - History shows the previous code inputs
- Files, plots, packages, help window
  - Haven't really used this window except for the help menu and viewing plots that I generated

Then learnt about the basics of R

- Using R as a calculator
- Creating variables and vectors
- Manipulating the vectors
- Doing calculations and applying functions to the vectors

We also learnt to download packages (and learnt what packages are), how to set working directory and how to save and create new projects.

Functions and commands learnt

- 'gets' <-
- "#" is a comment, not script, used to explain the code and write notes
- mean -average
- var -variance
- max -maximum value
- min -minimum value
- length -number of values in the vector
- sum -add all numbers in the vector
- sort -sort in ascending order
- class -tells the classification of the vector, e.g. numeric, character

- which(x<2) -tells the position of values where x is less than 2
- rep -repeats a set of value
- seq -creates a sequence of values

Afternoon

Key Takeaways

- Data management and record keeping is very important. Keep original, unedited copies the data. Include text files that explain what was done to clean the data. Keep thorough notes in the code so you know what everything means later when you forget, other people should be able to know whats going on as well.
- Setting up the data to be in columns is also useful for better display and easier analysis in R
- Back up your files

Day 2

Morning

Course: Introduction to R

Advanced Functions

- Importing data
  - read.csv or read.xlsx
- Exporting tables
  - write.csv or write.xlsx
- create a data frame (df)
  - data.frame
  - essentially just a spreadsheet with everything in columns as variables, characters or factors
- head(df) gives the first 6 rows of the data frame
- summary(df) gives a short summary of the data
- str(df) gives information about the structure
  - there are levels within the factor
  - levels(df$factor)
  - these can be changed
- correlation matrix
  - cor(df)
- Linear regression model
  - Lm(formula=y-variable~x-variable, data=df)
  - ~ means 'by'
- t-test
  - t.test(sample1,sample2)

  - t.test(y-variable~x-variable, data=df)
- tapply
  - used to apply a function to a specific group or variable
  - tapply(X=variable,INDEX=factor,FUN=function)

Graphics

- Scatterplot
- Learnt how to make a basic scatterplot
- There are lots of characteristics of the plot that can be modified
  - col -colour
  - xlab/ylab -axis titles
  - main -main title
  - type -point/line/combo
  - pch -point type
  - lty -line type
- Can also add other attributes with lines of code
  - abline -adds a line
  - text -adds text
  - legend -adds a legend
  - points -adds points
- all these can be greatly modified
- Histogram
  - hist
  - also able to modify with code
- Boxplot
  - boxplot
  - able to modify

Afternoon

Key Takeaways

- Google and trial and error will get the answer eventually
- Code doesn't have to be pretty, just has to work
- Need to be accurate and precise with typing- capitals matter, punctuation etc., be exact

Day 3

Morning

Course: Intro to Experimental Design

Started off by learning the basic content of design

- Populations vs samples
- Treatments
- Replications and Psuedoreplications
- Experimental units vs Observational units
- Blocking
- Confounding
- Heterogeneity
- Factors and levels of factors
- Effects and Interactions

Designs

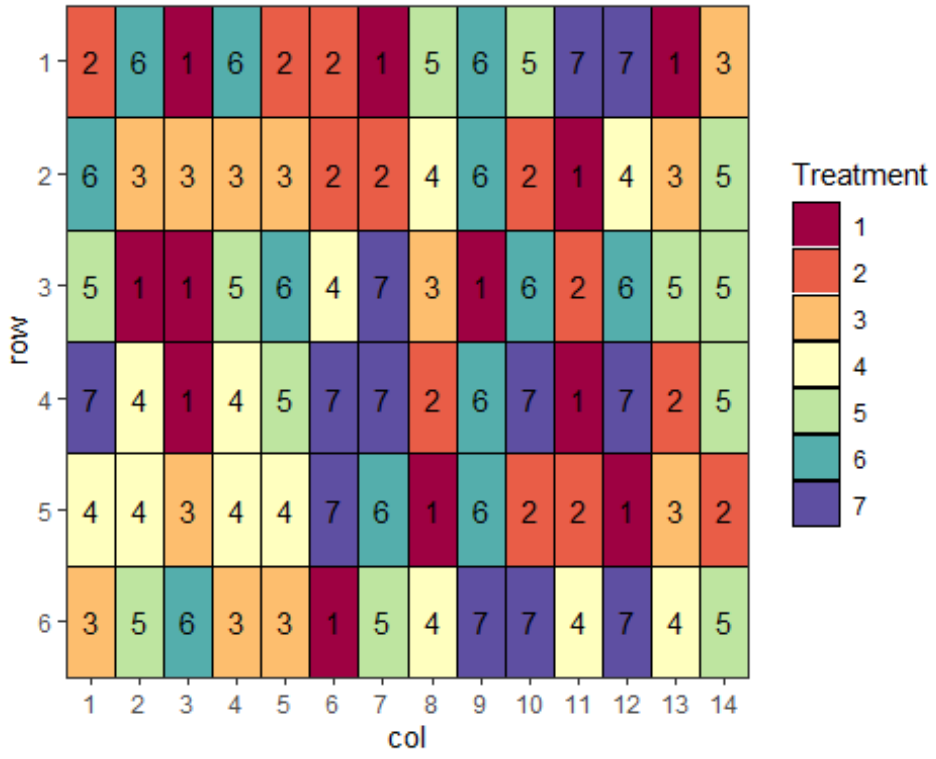
- Firstly write the summary of the experiment
- Decide on the treatment names and store this as a vector
- Decide on number of replicates

CRD

- Completely Randomised Design
  - To create the design
    - design.crd(treatment vector,r=no. of replicates)
  - To display the design
    - des.info(design.obj=design from above, nrows=, ncols=)

```
#exercise 1#

#Aim-effect of treatment on scab
#Obs- 84
#Arr- 6 rows, 14 colums, no blocking
#Trt- 7
#rep- 12
#des- crd

trt<-c(1:7)
rep<-12

outdesign<-design.crd(trt,r=rep)
des.out<-des.info(design.obj = outdesign,nrows=6,ncol=14)

Source of Variation                     df
 ============================================
```

```
trt                                         6
Residual                                    77
============================================
Total                                       83
```

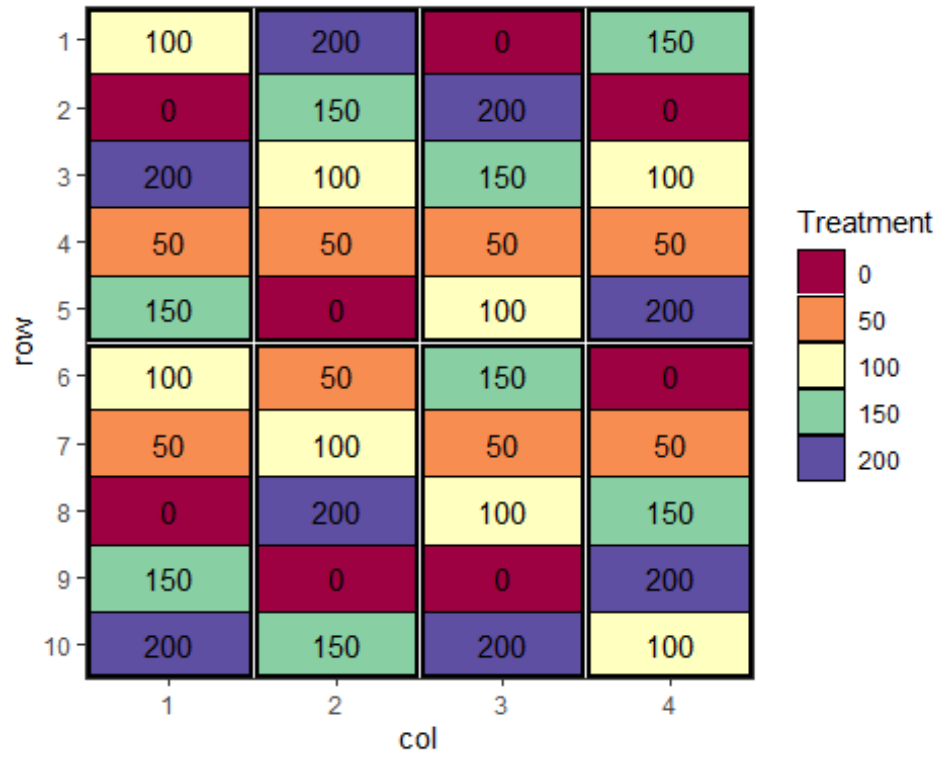


RCBD

- Randomised Complete Blocking Design
  - To create the design
    - design.rcbd(treatment,r=no.of replicates)
  - To display the design
    - des.info(design.obj=design from above,nrows=,ncols=,brows=,bcols=)

```
#Exercise 4

#aim- response to N fert
#obs- 40 plots
#arr- 10 rows, 4 col
#trt- 5
#rep- 8
#des- rcbd
#blk- 5 rows, 1 col

trt<-c(0,50,100,150,200)
rep<-8
```

```
outdesign<-design.rcbd(trt,rep)
des.out<-des.info(design.obj = outdesign,nrows=10,ncols=4,brows = 5,bcols=1)

Source of Variation                        df
 ============================================
 Block stratum                              7
 --------------------------------------------
 trt                                        4
 Residual                                  28
 ============================================
 Total                                     39
```



LSD

- Lattice Square Design
  - To create the design
    - design.lsd(treatment vector)
  - To display the design
    - des.info(design.obj=design from above,nrows=,ncols=)

```
#Exercise 6

#aim- effect of petal colour on pollen beetles in rapeseed
#obs- 25
#arr- 5x5
#trt- 5
#rep- 5
```

```
#des- lcd
#blk- 5x1+1x5

colour<-c("blue","yellow","green","red","pink")
outdesign<-design.lsd(colour)
des.out<-des.info(design.obj = outdesign,nrows = 5,ncols=5)

Source of Variation                        df
 ============================================
 Row                                        4
 Column                                     4
 colour                                     4
 Residual                                  12
 ============================================
 Total                                     24
```

P rep

Key Takeaways

- Always write down for each design:
  - Aim
  - Observations
  - Arrangement
  - Treatments
  - Replicates

  - Design
  - Block Arrangement
- Getting the right design is the most important thing

- Getting the treatments, replications etc. right is also important
- Bad design can invalidate results
- Determine the arrangement, treatments, replicates, sesign BEFORE

Day 4

Morning

Course: Intro to Experimental Design

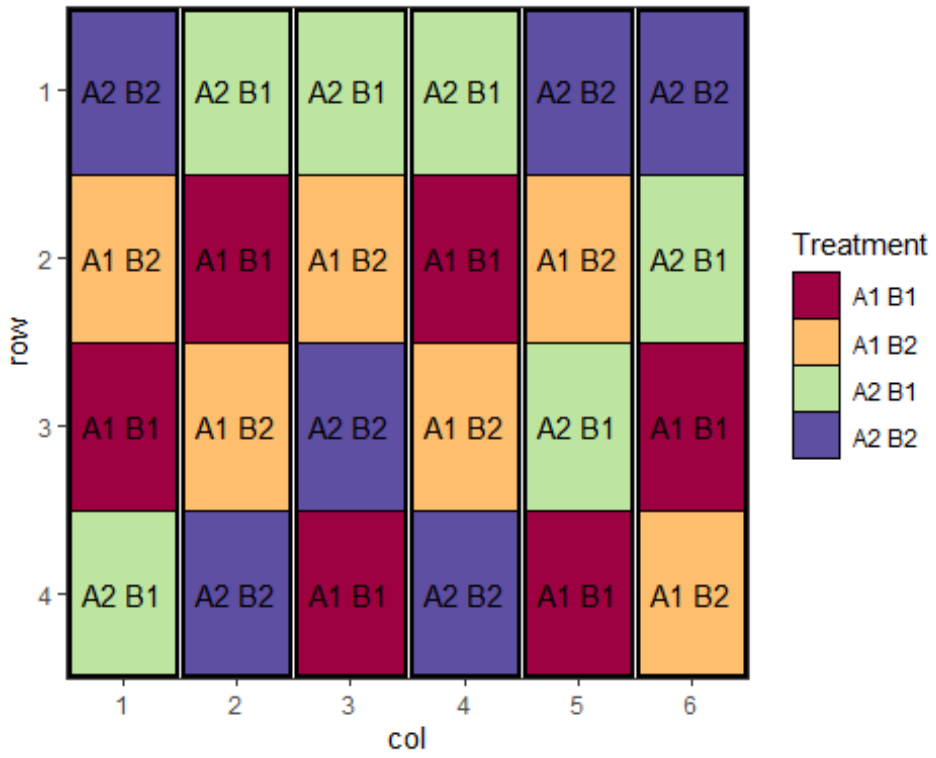
Continued learning different how to create different designs

Factorial Treatment Designs

- Factorial treatments
  - Used when there are 2 factors, i.e. two different treatments
  - Crossed- every combination of levels, e.g. variety + irrigation
  - Nested- one treatment is nested in the other, e.g. grain type + digestibility
  - Can use crd, rcbd, lsd
  - For crd, rcbd need to set the treatment as the levels in each factor
    - Trt<-c(3,2) for a 3x2 factorial
    - design.ab(trt,rep,design="crd/rcbd")
    - des.info is the same as crd/rcbd
    - for lsd use design.lsd

```
#aim-effect of genotype on plant extract
#obs- 24
#arr- 4x6
#rep- 6
#trt- 2x2 fact
#des- factorial rcbd
#blk- 4x1

trt<-c(2,2)
rep<-6
design<-design.ab(trt,rep,design="rcbd")
outdesign<-des.info(design.obj = design,
          nrows=4,ncols=6,brows=4,bcols=1)

Source of Variation                    df
 ============================================
 Block stratum                          5
 --------------------------------------------
 A                                      1
```

```
B                                         1
AB                                        1
Residual                                  15
==========================================
Total                                     23
```



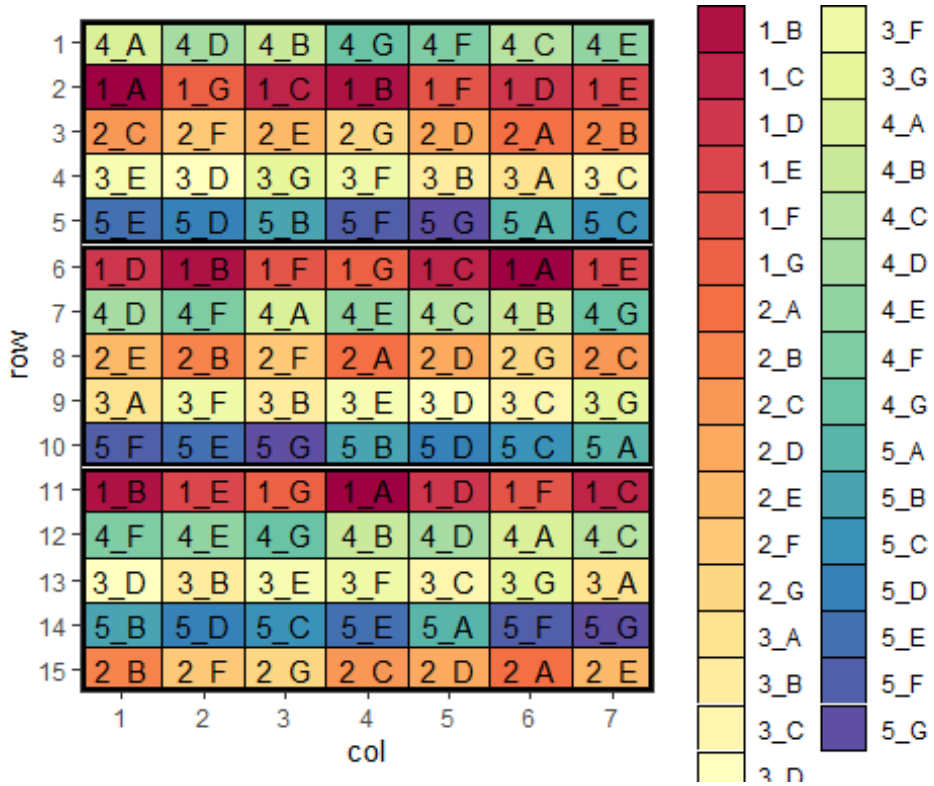
Split Plot

- Split Plot
  - First store two treatments
    - Trt1<-
    - Trt2<-
  - design.split(trt1,trt2,r=)
  - des.info same as rcbd

```
# Exercise 11

#aim- effect of variety and fertiliser on grain quality
#obs- 105
#arr- 15x7
#trt- 5x7 factorial
#rep- 3
#des- split plot
#blk- 5x7

#whole plot
fert<-c(1:5)
```

```
#subplot
var<-LETTERS[1:7]
rep<-3

design<-design.split(fert,var,r=rep)
designout<-des.info(design.obj = design,nrows=15,ncols=7,brows=5,bcols=7)
```

```
Source of Variation                          df
 ================================================
 Block stratum                                2
 ------------------------------------------------
 Whole plot stratum
          fert                               4
 Whole plot Residual                          8
 ================================================
 Subplot stratum
          var                                6
          fert:var                          24
          Subplot Residual                  60
 ================================================
 Total                                       104
```



Strip Plot

- Strip Plot
  - Similar to split plot but both treatments have whole plots that run perpendicular

- Degrees of Freedom
  - Different designs of the same experiment can influence the df
  - Increasing complexity decreases df
  - Therefore keep the design as simple as possible, accounting for sources of error

Afternoon

Talk: Russel (Functions and Programming)

- Writing functions in R
- Most important thing in coding
  - Managing Complexities
    - Write own function
    - Set out a skeleton to break the whole task into smaller steps
    - Break these steps into smaller steps
    - Write code for individual steps, ignoring the rest
    - Keep it Simple

Key Takeaways

- Need to have a good idea of what the layout of the trial will be before starting to code for the design, setting up the summary is important, makes coding easier
- Attention to detail in the summary and coding
- Look at degrees of freedom when choosing which design- should be >12
  - Crd>rcbd>lattice>split>strip in terms of ability to pick up on a difference between treatments
  - Keep in mind the situation of the trials to which design would be most practical and most accurate
- Break down complicated problems into simple steps

Day 5

Morning

Talk: Pete (Tidyverse)

We started today with a talk from Pete about tinyverse functions.

This included may very useful functions in data cleaning and manipulating that help to organise the data so that the analysis is much easier. Some of the functions are:

- tibble tables
  - a better way of putting data into a table compared to data frames
- cbind and rbind
  - used to put columns or rows together
- pipe

    - to apply functions in a string
- spread
    - sorts data from 1 column into 2
- separate
    - separates data from 1 element into a new column
- unite
    - opposite of separate
- missing values
    - use na.rm=TRUE to hide missing values
- complete
    - fill in missing data with NA
- filter
    - only shows the specific data that is being filtered by
- arrange
    - sorts the data
- select
    - choose specific columns
- mutate
    - adds new columns based on a function applied to other columns
- group_by
    - ability to group data by a factor
- for loops
    - This one is complicated and I still don't really understand it
    - used when there's repetition for different values

Afternoon

Talk: Sam (Rmarkdown)

We had a talk from Sam about Rmarkdown and briefly about Shiny. Rmarkdown seems interesting and easier to use for formatting tables and compared to Word and much easier for consistent formatting over the whole document. It can also allow for the easy inserting of R code into the document. Shiny seems to be a way of writing Rcode into a web app that anyone can easily use. It looks fairly complicated to code for but really easy to use the app.

Euler Problem 1

I also attemped one of the puzzles that Russel mentioned yesterday ended up getting the answer right. The code I wrote is here:

```
#Eulers Problem 1
#If we list all the natural numbers below 10 that are multiples of 3 or 5, we
get 3, 5, 6 and 9. The sum of these multiples is 23.
#Find the sum of all the multiples of 3 or 5 below 1000.
```

```r
#get numbers 1 to 999
numbers<-c(1:999)

# find all multiples of 3, 5 and 3 and 5

multiple_3<-numbers[(numbers%%3)==0]
multiple_5<-numbers[(numbers%%5)==0]

multiple_3and5<-multiple_5[(multiple_5%%3)==0]

# sum multiples together and minus the multiples of 3 and 5

sum_3<-sum(multiple_3)
sum_5<-sum(multiple_5)
sum_3and5<-sum(multiple_3and5)

total<-sum_3+sum_5-sum_3and5
print(total)
```

```
[1] 233168
```

Euler Problem 2

```r
# Problem 2- Even Fibonacci Numbers
#Each new term in the Fibonacci sequence is generated by adding the previous
two terms. By starting with 1 and 2, the first 10 terms will be:

#  1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

#By considering the terms in the Fibonacci sequence whose values do not excee
d four million, find the sum of the even-valued terms.

#create list of Fibonacci numbers
fib<-c(1,2,3)

for(i in 3:100){
  if(fib[length(fib)]<4e6){
   fib[i]<-fib[i-1]+fib[i-2]
}
}

#Keep even numbers
even_fib<-fib[(fib%%2)==0]

#sum numbers
total<-sum(even_fib)
print(total)
```

```
[1] 4613732
```

Key Takeaways

- One thing that Pete mentioned is that most of the mistakes and errors that we get are from spelling or punctuation typos in the code. This highlights the need to be extremely accurate with typing and read your code as your typing to pick up on mistakes.
- I don't need to remember how to use each function, I just need to remember there is a function for everything I want to do and just use Google to find it.
- Try using Rmarkdown for reports and assignments.

Week 2

Day 1

Morning

We started off the day trying to replicate our Genstat outputs from Workbook 10 in the Research Methodology course. This was fairly hard and required a lot of Googling and learning how to do ANOVAs of different comparisons and trying to create ggplots. This took a while and progress was slow but I worked through it slowly. It was pretty satisfying to finally get a plot to work after a lot of trying different codes heaps of times.

Meeting: Hotdesk

We then sat in on a "HotDesk" meeting between an Honours project student and some BiometryHub statisticians. The HotDesk is a new program created by the BiometryHub for students to consult with the biometricians to get an idea of how to design and run their project from a statistics point of view. This would be important for students so that they get an appropriate, accurate design that will allow for proper analysis.

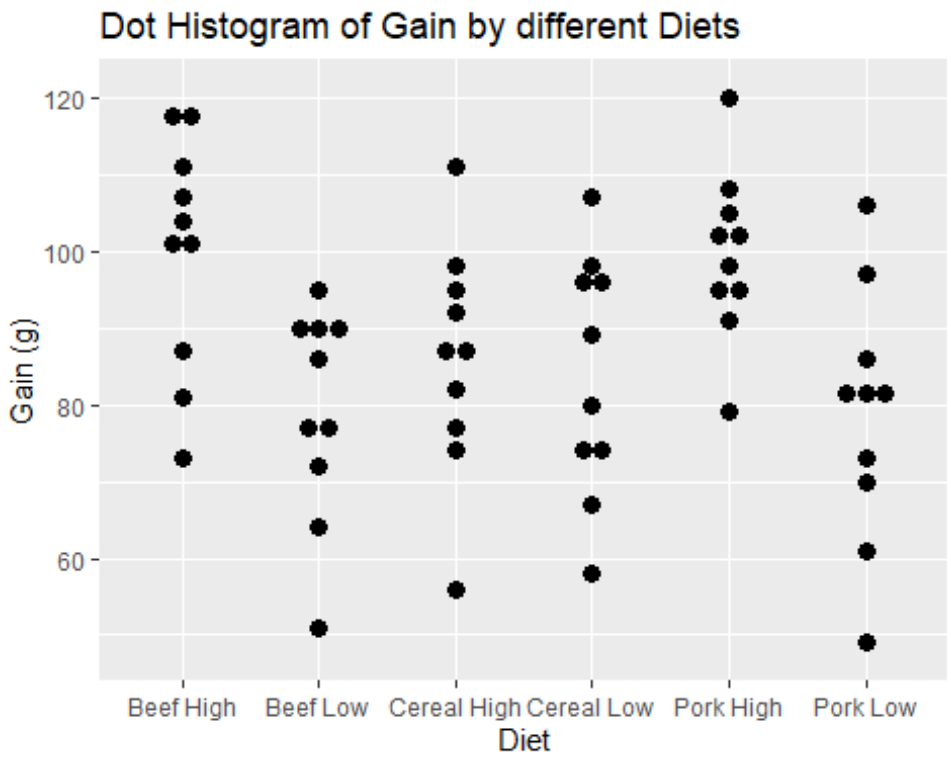
Afternoon

Talk: Pete (ggplot)

In the afternoon session, Pete gave a short talk on ggplot and its capabilities. It seems like an extremely useful tool with a diverse set of capabilities and sets everything out in a logical, step-wise order. I think it will take a bit of getting used to but with some practice I think it would be easier to use than Excel for many applications and is much more versatile.

Some of the graphs that I have plotted using ggplot using the data and instructions from the Week 10 Workbook from Research Methodology are shown below.

```
#create dot plot
ggplot(ratfactorial,aes(diet,Gain)) +
  geom_dotplot(binaxis="y",stackdir="center",binwidth = 2.5)+
  xlab("Diet")+ylab("Gain (g)")+
  ggtitle("Dot Histogram of Gain by different Diets")
```




```r
#Create means plot with standard errors
ggplot(means_table, aes(x=Diet, y=Gain)) +
  geom_errorbar(aes(ymin=Gain-se, ymax=Gain+se), width=.1) +
  geom_line() +
  geom_point()+
  ggtitle("Means Plot with Standard Errors")
```

Euler Problem 4

```
#Problem 4
#A palindromic number reads the same both ways.
#The largest palindrome made from the product of two
#2-digit numbers is 9009 = 91 x 99.
#Find the largest palindrome made from the product of two 3-digit numbers.

#get all possible combinations
a<-c(100:999)
b<-tcrossprod(a,a)

total<-as_tibble(as.integer((as.vector(t(b)))))
colnames(total)<-"A"

#reverse the numbers

total_char<-strsplit(as.character(total$A),split = "")
total_rev<-lapply(total_char, rev)

for(i in c(1:length(total$A))){
  total_rev[[i]]<-paste(total_rev[[i]],collapse="")
  }

# intersect the number
```

```
total_rev_int<-as_tibble(as.integer(total_rev))
colnames(total_rev_int)<-"A"
matching<-which(total$A == total_rev_int$A)

#find highest intersect

result<-total$A[matching]
print(max(result))

[1] 906609
```

Key Takeaways

- For the future projects: have a clear idea of what your project is and its limitations and constraints. You need to know what the treatments and how many replications you need before you design and run your experiment.
- Google everything you don't know

Day 2

Morning

Meeting: Experiment Analysis

The meeting was fairly indepth and Richard talked about how the analysis of an experiment conducted in 2003 may not be entirely accurate and possibly had some errors. The main takeaways that I got from the meeting was that you need to have a clear understanding of the structure of the experiment to properly analyse it and to account for all sources of error. It would also help to break down the overall experiment into smaller, easier to understand subsections.

Meeting Notes

- Block-treatment interactions
  - Block effect, treatment effect, no interaction- ideally
  - Any block-treatment is included in the error term
- Main plot x treatment interaction should be included
  - Cannot predict the effect of treatment in future main plots
  - If this is significant then treatment effect depends on which main plot it is in
- Two phase experiment
  - 8 patients given active and passive treatments ++ Patients grouped into expressive and unexpressive
  - The patients were rated for pain by 74 medics ++ Half were randomly assigned to a training course
  - Phase 1 ++ Analysing pain for each patient
  - Phase 2 ++ Raters analysed +++ Is training effective? Does effect depend on E/M?
  - Training effect may differ between patients

- Lessons
  - Understand the whole structure
  - Use simplified examples for better understanding
  - Analyse in different ways and compare outputs for differences or consistency
  - Look at subsets of the data, looking at phases separately
  - Use ANOVA to develop the model but use ASREML for final analysis

Afternoon

Talk: Wendy (Exact Permutations Test)

In the afternoon we had a talk from Wendy about her masters project in Random Permutations test. This is a pretty complicated topic that took a while to understand but I think I eventually got it. It involves taking every combination/arrangement/permutation of the responses to experimental units, calculating a test statistic for each permutation and then creating a distribution from these permutations. You then take all the permutations that have a test statistic equal to or more extreme than the observations and determine how likely the data results are.

Euler Problem 5

```
# Problem 5
#What is the smallest positive number that is evenly
#divisible by all of the numbers from 1 to 20?

#get list of all possibilities
n<-(300000000)

#list of all factors
# do not need any values below 11 because they are all factors of numbers 11:
20
start<-11
finish<-20

# function to get all factors
get_factors<- function(x){
  seq(from=0,to=n,by=x)
}

# intersect all factors
fact<-list()
fact[[1]]<-get_factors(start)

for(i in c(start:finish)){
  fact[[1]]<-intersect(fact[[1]],get_factors(i))
}

# print results
```

```
result<-min(fact[[1]][!fact[[1]] %in% 0])
print(result)

[1] 232792560
```

Day 3

Morning

Talk: Mario (Bioinformatics)

We had a talk from Mario on his work in bioinformatics and machine learning.

- Bioinformatics

Bioinformatics exists as a link between biology and statistics and a the bioinformatrician needs to have a good understanding of the biology behind the experiments being conducted as well as some understanding of the statistics. The aim of bioinformatics is to determine which genes have a statistically higher or lower gene expression in a treatment compared to a control. The function of these genes and any genes related to this gene also need to be determined. This information goes back to the biologist and they can then target specific genes to study that have the highest effect in response to the treatment.

- Machine learning

Using previous inputs and outputs to predict future outputs based on current inputs.

For Example:

- predicting the occurrence of spore showers based on weather inputs
- counting the number of weeds in a field based on an image input
- detecting cancer based on radiography images

Talk: Mexiuan (Honours)

We had a talk from Mexiuan who completed her honours project this year. Her project centered around modeling the emergence of Canola seedlings over time and she tried various different models to determine which one best described the data. She also mentioned that the hardest part about the project and the most important skill she learnt was about time management and the ability to stay focussed on the project to get all the work done on time.

Euler Problem 6

```
# The sum of the squares of the first ten natural numbers is,
# 1^2+2^2+...+10^2=385
# The square of the sum of the first ten natural numbers is,
# (1+2+...+10)^2=552=3025
#Hence the difference between the sum of the squares of the first ten natural
numbers and the square of the sum is 3025-385=2640.
# Find the difference between the sum of the squares of the first one hundred
```

```
natural numbers and the square of the sum.

#Calculate sum of square
number<-c(1:100)
sum_of_square<-(sum(number))^2

#Calculate square of sum
square_of_sum<-sum(number^2)

#calc difference
difference<-sum_of_square-square_of_sum
print(difference)

[1] 25164150
```

Day 4

Morning

Demonstration: Pete (Drones)

We went and had a look at Pete's work with drones and mapping seedling emergence of faba beans. This is super interesting work and has many applications for growers to be able to spatially map the emergence of seedlings and identify hot spots of good or bad emergence. It could also allow for researchers to get more accurate data collection using the drones instead of counting by hand.

Afternoon

Talk: Enqi (Honours)

We spoke to Enqi about his experiences doing Honours a couple of years ago. Since then he has worked for SARDI in Port Lincoln and now at a Chinese vegetables greenhouse that supplies to a lot of restaurants in Chinatown. We asked him a few questions about his experiences with Honours and getting a job later on. After this talk I definitely still want to do an Honours project because I think I would learn a lot and it would give a good taste of what a career in the research industry would bring.

Day 5

Morning

Talk: Sam (CV and Website)

Sam spoke to us about creating a CV and an online profile. He showed us what his CV looked like as well as his professional accounts on sites such as Linkedin and Github. He mentioned the importance of keeping the profiles professional and consistent. We also had a look at his website that he created and said that it would be very easy for us to make. Sam

gave us a link to a book on BlogDown that went through the step-by-step process of creating a website through Hugo and Netlify.

Afternoon

In the afternoon, I decided to try to create my own website, it was surprisingly easy to create and upload the website but took a long time to get it to be the way I wanted it and to get all the information right. I continued to work on it over the weekend and it did get a bit frustrating at times because a small error would screw the whole thing up and there are a lot of different files that I had to edit and create. Eventually after a lot of trial, error and debugging, I got the website up at which is available at:

alecmccallum.netlify.app

Week 3

Day 1

Morning

Talk: Beata (Genetic Association Analysis)

The talk from Beata about her work in genetic association analysis was really interesting and outlined how the phenotype and genotype are linked through markers or SNPs. This related really well to the content that we went through in the Plant Breeding course at Uni. Beata went through the whole process of how to determine which genes had the most significant effect on the target phenotype. There were two main ways of doing this analysis: Wgaim and Bayesian statistics. The Wgaim method is more widely used because it has been around for a long time and the Bayesian method is relatively new and still being researched to determine its accuracy. The statistics itself was fairly complicated but the results were easy to understand and had a very practical applications for the researchers and breeders.

Afternoon

Talk: Paul (Experiences)

We had a casual chat to Paul about his experiences in the workforce and his 20+ year career as a statistician working at the Waite Campus. He had some interesting points about what it was like to be working in research groups, mostly as a consultant. Some of the things he mentioned was to keep in contact with the Biometry Hub and statisticians over the whole project so that nothing goes wrong. This is especially important for the design because if the design is wrong the results are essentially useless. Paul also talked about having a diverse expertise because being too narrow could mean that if funding ends for that area then you don't really have anywhere else to go.

Euler Problem 9

```
# A Pythagorean triplet is a set of three natural numbers, a < b < c, for whi
ch,
# a^2 + b^2 = c^2
# For example, 32 + 42 = 9 + 16 = 25 = 52.
# There exists exactly one Pythagorean triplet for which a + b + c = 1000.
# Find the product abc.

# total and range of values
total<-1000
range<-c(1:total)


# function to find triplets
triplet<- function(A,B) {
    C<-sqrt(A^2+B^2)
    if((C%%1)==0){
    return(c(A,B,C))
    }
}

# find all triplets within range that satisfy conditions
for(i in range){
  for (j in range){
    triplet(i,j)
    if((i+j+(sqrt(i^2+j^2))==1000)& (i<j)){
      result<-triplet(i,j)
      }
  }
}

#get results
print(result)

[1] 200 375 425

print(prod(result))

[1] 31875000
```

Day 2

Morning

In the morning I continued to work in the Workbook 10 exercises

Afternoon

I started working on the presentation this afternoon which took a while to get started because I wanted to use Rmarkdown to make it. This meant that I had to download a

package to do that and it took a while to figure out how to work it because it was slighly different to the normal Rmarkdown.

Euler Problem 8

```
# Euler Problem 8
# The four adjacent digits in the 1000-digit number that have the greatest pr
oduct are 9 ? 9 ? 8 ? 9 = 5832.
options(scipen = 999)

# need to add separately otherwise it goes to infinity
# need to add as characters because of floating point numbers
value<-list()
value[[1]]<-c("731671765313306249192251196744265747423553491949349698352031277450632623957831801698480186947885184385861560789112949495459501737958331952853208805511125406987471585238630507156932909632952274430435576689664895044524452316173185640309871112172238311362229893423380308135336276614282806444486645238749")
value[[2]]<-c("622298934233803081353362766142828064444866452387493035890729629049156044077239071381051585930796086670172427121883998797908792274921901699720888093776657273330010533678812202354218097512545405947522435258490771167055601360483958644670632441572215539753697817977846174064955149290862569321978468622482")
value[[3]]<-c("536978179778461740649551492908625693219784686224828397224137565705605749026140797296865241453510047482166370484403199890008895243450658541227588666881164271714799244429282308634656748139191231628245861786645835912456652947654568284891288314260769004224219022671055626321111109370544217506941658")
value[[4]]<-c("178664583591245665294765456828489128831426076900422421902267105562632111110937054421750694165896040855626321111109370544217506941658960408071984038509624554443629812309878799272442849091888458015616609791913387549920052406368991256071760605886116467109405077541002256983155200055935729725716362695618826704282524836008232575304207529634")
# Find the thirteen adjacent digits in the 1000-digit number that have the gr
eatest product. What is the value of this product?

# Split the long number into individual numbers
value_split<-strsplit(as.character(value),split = "")
value_int<-list()
for(i in c(1:length(value_split))){
  value_int[[i]]<-as.integer(value_split[[i]])
}

# create lists of tibbles of 13 consecutive numbers in columns
y<-as_tibble(c(0:12))
z<-list()
for(j in c(1:4)){
for(i in c(1:300)){

  y[i]<-as_tibble(value_int[[j]][i:(i+12)],.rows=13)
```

```
  colnames(y)<-c(1:300)
  z[[j]]<-y
  }
}

# Find product of each column of 13
f<-list()
for(i in c(1:4)){
  f[[i]]<-colProds(as.matrix(z[[i]]),na.rm=TRUE)
}

# Print results
result<-max(unlist(lapply(f,max)))
print(result)
```

```
[1] 23514624000
```

Day 3

Morning

This morning I continued working on the presentation and finished off the last Workbook 10 exercise.

Afternoon

Meeting: Olena (Honours)

- Doing Honours will make me more prepared for a job and more competitive in the job market
- The agricultural industry wants and needs people to be trained in data management and analytics
- Continually learn and develop new skills

Day 4

Morning

Today was spent working on the presentation, it was slow progress because everything was a bit different to the normal Rmarkdown and it took a while to figure out.

Afternoon

Continued to work on the presentation.

Day 5

Morning

This morning we had a mock run through of our presentation which I think went really well. The presentation went for about 30 minutes which was what I was aiming for. There were, however, a number of mistakes that I had to fix up for the actual presentation.

Afternoon

Presentation

The presentation in the afternoon went really well, except for some technical issues in the beginning. We probably should have logged into the zoom early to make sure everything worked properly. Other than that I think I spoke for about 30 minutes which was a good time and I think I spoke well. There werent many hard to answer questions at the end but I think it was a good wrap up for the internship.