R Masterclass Slides

Dr Emma Howard

2023-06-01

Plan for Masterclass

1. Introduction to R/RStudio
2. Introduction to R Markdown
3. Loading and Formatting the Data
4. *Open session and Tea Break*
5. Manipulation of Data
6. Data Summaries
7. *Open session and Tea Break*
8. Graphical analysis
9. Bonus segment on cool things in RStudio
10. *Open session and Feedback*

Section 1

Introduction to R/RStudio

About R

- R is a language and environment for statistical computing and graphics: https://www.r-project.org/
- R was created by Ross Ihaka and Robert Gentlemen in the early 90s at the University of Auckland New Zealand.
- R was developed from another statistical language S.
- The R Core Team was formed in 1997 to further develop the language.
- Currently associated with R, there are 18,498 packages, 36,817 datasets, and 14,981 articles by 9,241 maintainers and 18,697 contributors.
- Version 4.3.0 (Already Tomorrow) will be released on 21st of April 2023.

Why use R (part 1)

- R is free and open-source
- R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible.
- It is available for Windows, Mac and Linux
- R can be used online with RStudio Cloud https://rstudio.cloud/
- It is considered better for statistical analysis than SAS, Stata, SPSS and Python

Why use R (part 2)

- For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.
- Extra functions can be imported through the use of packages i.e., R is augmented through packages
- Did I mention it is free?

RStudio

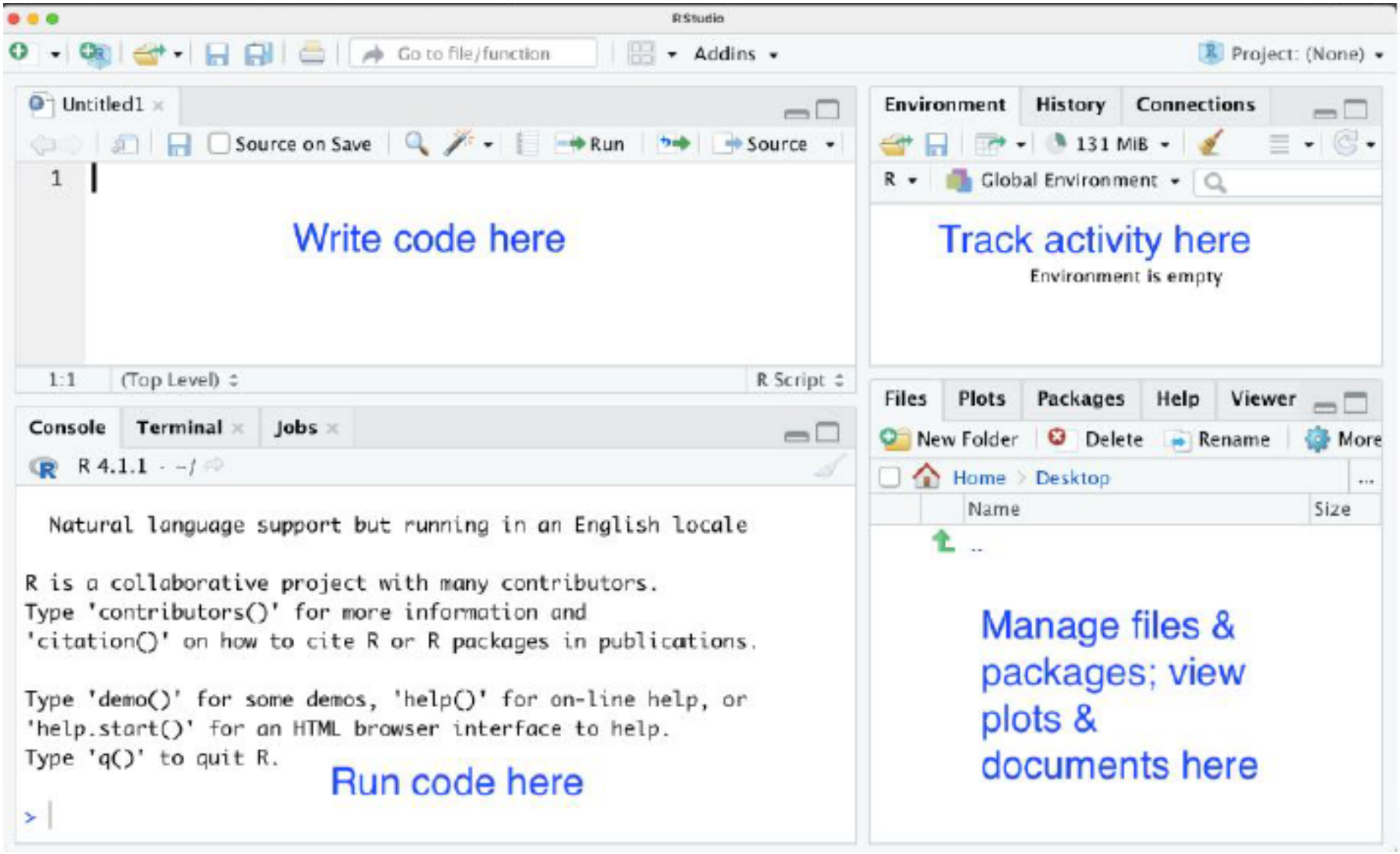
It is easier to use R through an Integrated Development Environment (IDE). The most popular IDE for R is RStudio.

Features provided by RStudio include:

- syntax highlighting, code completion, smart indentation
- interactively send code chunks from editor to R
- organise multiple scripts, help files, plots
- search code and help files

RStudio provides a few shortcuts to help write code in the R console go to Help - Keyboard Shortcuts Help

RStudio


RStudio
Go to file/function
Addins
Project: (None)
Untitled1
Source on Save
Run
Source
1
Write code here
1:1
(Top Level)
R Script
Console
Terminal
Jobs
R 4.1.1 · ~/
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
Run code here
>
Environment
History
Connections
131 MiB
R
Global Environment
Track activity here
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Desktop
Name
Size
..
Manage files & packages; view plots & documents here

Section 2

Introduction to R Markdown

Creating a R Markdown Documents

1. To create a new R Markdown document: go to File > New File > R Markdown

   - The first time you use R Markdown on your machine you may be asked to install some R packages; if so, press Yes.

2. Select the type of output document you want to create.

   - You are able to produce HTML output on any computer.
   - To produce a Word document Microsoft Word needs to be installed.
   - To produce pdf output you need to have LaTeX installed.

If and only if you do not have LaTeX installed (check by trying to compile a R Markdown pdf document), run the following lines to install LaTeX:

install.packages("tinytex")

tinytex::install_tinytex()

Help with R Markdown Documents

1. An extended guide with tutorials is available on the RMarkdown website: https://rmarkdown.rstudio.com/lesson-1.html
2. R Markdown Cheatsheet https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf
3. Free online book: R Markdown for Scientists
4. Free online book: R Markdown Cookbook

R Markdown Basics

- Headings
- Italics
- Bold
- Bullet Points
- Numbered Points

Next steps in R Markdown: TOC and Images

- Table of Contents

```
---
title: "Maths Education Dataset Analysis"
author: "Dr Emma Howard"
output:
  pdf_document:
      toc: true
      toc_depth: 2
      number_sections: true
  html_document:
    df_print: paged
---
```

Figure 1: Table of Contents Code for YAML

- Inserting Images

Next steps in R Markdown: Formulas and LaTeX

The style for formulas follows TeX syntax (proficiency with TeX syntax is not required). In general, you can use TeX syntax anywhere in the R Markdown documents. TeX is widely used as it makes it easy to include mathematical notations and formulas in documents.

For example, an inline formula could be $ax^3 + bx + c = 0$. Otherwise, an equation would look like:

$$\bar{X}_{ij} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \mathcal{L}_{\mathbf{x}}(\theta) = \prod_{i=1}^{n} f(x_i|\theta)$$

Next steps in R Markdown: Colour

How to specify color depends on the format of your output

html output:

Roses are red, and violets are blue. Colour is good to highlight a point.

pdf output:

Roses are red, and violets are blue. Colour is good to highlight a point.

html and pdf output:

Roses are red, and violets are blue. Colour is good to highlight a point.

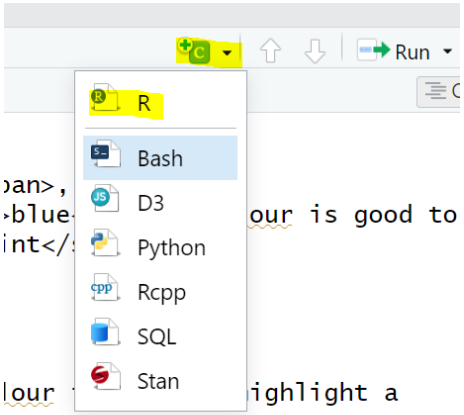
Embedding R Code (part 1)

To embed code, on my keyboard, the special dash that we use is next to the "1" key:

```
```{r}

```
```

Or we can insert a code block using:



Embedding R Code (part 2)

These initial lines are used to set up global options:

```
knitr::opts_chunk$set(echo = TRUE)
```

In this case, we intend to show both the code and the corresponding output, for every chunk of code. Note the option wheel on the right hand side of the text editor, which allows a more user friendly setup.

Commonly used options are listed in the R Markdown Cheat Sheet.

Embedding R Code (part 3)

I have included here a chunk of code that will not appear in the document (however the output of the code does).

```
## [1] 5
```

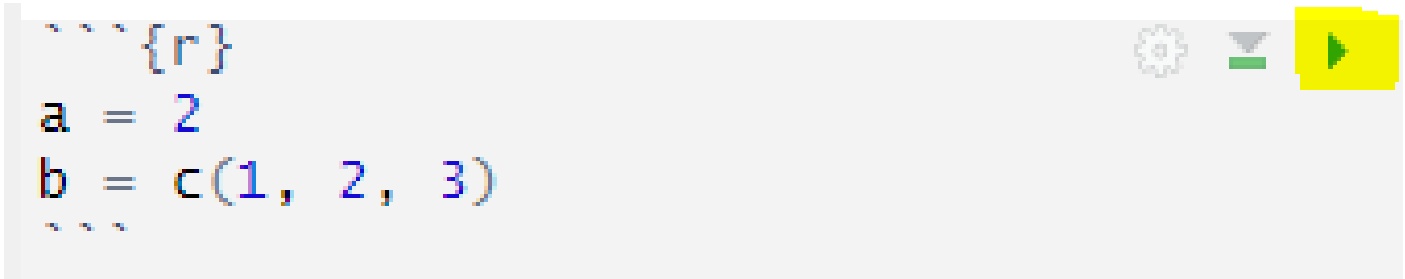
Note the option (echo=FALSE) that guarantees that the code will not appear. When you type in the comma after r, RStudio will prompt a lengthy list of options that we could choose from. For any given chunk you can temporary redefine any global option (i.e., local options override global options).

Basic operations (part 1)

Let's define some variables:

```
a = 2
b = c(1, 2, 3)
```

The scalar *a* and the vector *b* are now defined and can be used in any of future code. If I want *a* and *b* to be accessible in the R environment (outside of this R Markdown document), I should click the highlighted button:



Basic operations (part 2)

Example of using *a* and *b*:

```
vec = a * b
vec
```

```
## [1] 2 4 6
```

The code can be added inline with the same dash symbol as the dash symbol used to define code blocks. For it to be recognised as R code, I need to write ‘ r (the code) ‘ and it will be evaluated. For example, ‘sum(vec)‘ is equal to 12.

Section 3

Loading and Formatting the Data

Reading in the Data (part 1)

How we read in a dataset depends on:

- the format we want to work with, and
- the type of file we are importing.

For the Masterclass, we will work with dataframes and the Maths Ed dataset[1].

[1] Alcock, L., Attridge, N., Kenny, S., & Inglis, M. (2013). Achievement and Behaviour in Undergraduate Mathematics: Personality is a Better Predictor than Gender (Version 1). Loughborough University. https://doi.org/10.6084/m9.figshare.865640.v1

Reading in the Data (part 2)

```
# Reading in a .csv file
MathsTotal = read.csv("Maths_Ed.csv", header = TRUE)
Maths = MathsTotal[,1:4] # work with first 4 columns only

# Reading in a .txt file
# name = read.table("name.txt", header = TRUE)

# Reading in an excel file
# install.packages("readxl")
# require(readxl)
# school = read_excel("Schools.xlsx") # tibble format
```

Check the Data (part 1)

```
# Check dimensions - number of rows and columns
dim(Maths) # or use nrow(Maths) and ncol(Maths)
```

```
## [1] 89  4
```

```
# See a few rows of the data
head(Maths) # Look at the first few rows of the data
```

```
##   Gender_OF_1M MLSC_Use VLE_Use Maths_Achievement
## 1            0       10     897           88.40000
## 2            0        0     853           70.25000
## 3            0       22     832           58.25000
## 4            0       22     558           68.90000
## 5            0        8     540           78.16667
## 6            0       38     520           68.66667
```

Check the Data (part 2)

```
# Check the current structure of the data
str(Maths)
```

```
## 'data.frame':    89 obs. of  4 variables:
##  $ Gender_0F_1M      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MLSC_Use          : int  10 0 22 22 8 38 18 15 3 5 ...
##  $ VLE_Use           : int  897 853 832 558 540 520 518 476
##  $ Maths_Achievement: num  88.4 70.2 58.2 68.9 78.2 ...
```

Renaming the Columns

```
# Renaming columns
colnames(Maths) = c("Gender", "MSC_Use",
                    "VLE_Use", "Mark")

# Always check any changes made!!!
str(Maths)
```

```
## 'data.frame':    89 obs. of  4 variables:
##  $ Gender : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MSC_Use: int  10 0 22 22 8 38 18 15 3 5 ...
##  $ VLE_Use: int  897 853 832 558 540 520 518 476 457 429 ...
##  $ Mark   : num  88.4 70.2 58.2 68.9 78.2 ...
```

Check the Data (part 3)

```
# Provides a quick summary of the data
summary(Maths[,1:3])
```

```
##      Gender          MSC_Use          VLE_Use
##  Min.   :0.000   Min.   :  0.000   Min.   : 81.0
##  1st Qu.:0.000   1st Qu.:  0.000   1st Qu.:250.2
##  Median :1.000   Median :  1.000   Median :319.5
##  Mean   :0.618   Mean   :  7.191   Mean   :349.8
##  3rd Qu.:1.000   3rd Qu.:  8.000   3rd Qu.:427.2
##  Max.   :1.000   Max.   :100.000   Max.   :897.0
##                                    NA's   :3
```

Formatting the Data

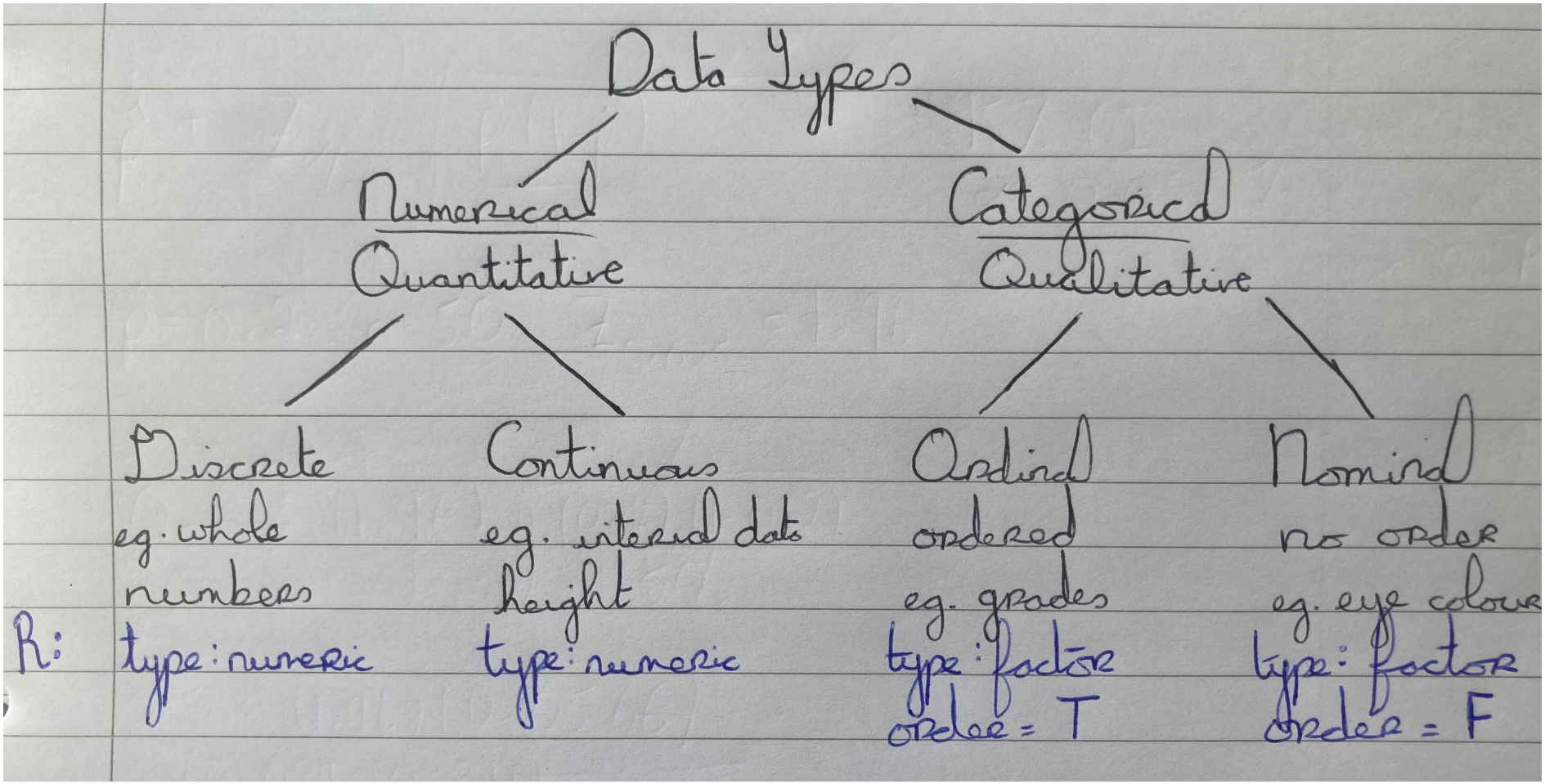


Figure 2: Data types and corresponding R Types

Formatting the Data

```
# Make 'Gender' a nominal factor
Maths$Gender = factor(Maths$Gender,
                      levels=c(0,1),
                      labels=c("Female", "Male"))

# Make 'MSC_Use', 'VLE_Use' and 'Mark' numeric variables
Maths$MSC_Use = as.numeric(Maths$MSC_Use)
Maths$VLE_Use = as.numeric(Maths$VLE_Use)
Maths$Mark = as.numeric(Maths$Mark)

# Round 'Mark' to two decimal places
Maths$Mark = round(Maths$Mark, 2)
```

Formatting the Data

```
# ... and check the structure!!!
str(Maths)
```

```
## 'data.frame':    89 obs. of  4 variables:
##  $ Gender : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1
##  $ MSC_Use: num  10 0 22 22 8 38 18 15 3 5 ...
##  $ VLE_Use: num  897 853 832 558 540 520 518 476 457 429 ...
##  $ Mark   : num  88.4 70.2 58.2 68.9 78.2 ...
```

Section 4

Manipulation of Data

Creating new variables (part 1)

```
# Not realistic but...
# create a new variable for 'Total Engagement'
# Maths$Total_engagement = Maths$VLE_Use + Maths$MSC_Use

# or maybe....
# Maths$Total_engagement =
        # Maths$VLE_Use + (10*Maths$MSC_Use)
```

As the variable on the LHS 'Total Engagement' does not exist in the dataset, R will create a new variable/column called 'Total_Engagement' according to our formula on the RHS. This approach can also be used to overwrite variables if they already exist.

Creating new variables (part 2)

```
#Create a new variable called Grade
Maths$Grade = cut(Maths$Mark,
    breaks = c(0, 40, 50, 60, 70, 100),
    labels = c("F", "D", "C", "B", "A"),
    right = FALSE)

# Check the dataset
head(Maths, n=3)
```

```
##   Gender MSC_Use VLE_Use  Mark Grade
## 1 Female      10     897 88.40     A
## 2 Female       0     853 70.25     A
## 3 Female      22     832 58.25     C
```

Creating new variables (part 2)

```
#Create a new variable called Grade
Maths$Grade = factor(Maths$Grade,
    labels = c("F", "D", "C", "B", "A"),
    levels = c("F", "D", "C", "B", "A"),
    ordered = TRUE)
```

Creating new variables (part 2)

```
# Check the data
str(Maths)
```

```
## 'data.frame':    89 obs. of  5 variables:
##  $ Gender : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1
##  $ MSC_Use: num  10 0 22 22 8 38 18 15 3 5 ...
##  $ VLE_Use: num  897 853 832 558 540 520 518 476 457 429 ...
##  $ Mark   : num  88.4 70.2 58.2 68.9 78.2 ...
##  $ Grade  : Ord.factor w/ 5 levels "F"<"D"<"C"<"B"<..: 5 5
```

Missing Data

```
# Are there NAs?
apply(Maths, 2, function(x) table(is.na(x)))
```

```
## $Gender
##
## FALSE
##    89
##
## $MSC_Use
##
## FALSE
##    89
##
## $VLE_Use
##
## FALSE  TRUE
##    86     3
```

Missing Data

Nearly all statistical techniques assume (or require) complete data. If cases with missing values are systematically different from cases without missing values, the results can be misleading.

There are three types of missing data:

- Missing Completely at Random (MCAR)
- Missing At Random (MAR)
- Missing Not At Random (MNAR)

Missing Data

To test for MCAR versus MAR (cannot test for MNAR):

- Create a dummy code for missing/non-missing in the dataset
- Conduct a chi-square or independent samples t-test to determine if the missingness on that variable, relates to the outcome of interest.
- If the results are significant, then the data are MAR (you have a variable in your dataset that relates to missingness)

Missing Data

```
Maths$VLE_Missing = ifelse(is.na(Maths$VLE_Use), 1, 0)
t.test(Mark~VLE_Missing, data=Maths)
```

```
##
##  Welch Two Sample t-test
##
## data:  Mark by VLE_Missing
## t = -0.93487, df = 2.2743, p-value = 0.4381
## alternative hypothesis: true difference in means between gr
## 95 percent confidence interval:
##  -22.79908  13.87458
## sample estimates:
## mean in group 0 mean in group 1
##        65.40442        69.86667
```

Handling Missing Data

1. Listwise deletion (data should be MCAR)
2. Pairwise deletion (data should be MCAR)
3. Last value carried forward (conservative)
4. Mean/median imputation (no statistician will recommend)
5. Multiple imputation
6. Maximum likelihood approach

Handling Missing Data - Listwise Deletion

```
# Remove rows with missing data
Maths1 = Maths
Maths1 = Maths1[complete.cases(Maths1), ]
dim(Maths1)
```

```
## [1] 86  6
```

Handling Missing Data - Mean Imputation

```
# A really bad approach...
Maths2 = Maths
Maths2$VLE_Use[is.na(Maths2$VLE_Use)] =
  mean(Maths2$VLE_Use, na.rm = TRUE)
```

Handling Missing Data - Multiple Imputation Example

```
require(missForest)
Maths3 = Maths
Maths3b = missForest(Maths3)$ximp
```

Handling Missing Data - Comparison

```
summary(Maths1$VLE_Use) # listwise deletion
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    81.0   250.2   319.5   349.8   427.2   897.0
```

```
summary(Maths2$VLE_Use) # mean imputation
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    81.0   251.0   324.0   349.8   422.0   897.0
```

```
summary(Maths3b$VLE_Use) # imputation
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    81.0   251.0   324.0   349.8   422.0   897.0
```

Section 5

Data Summaries

Descriptive Statistics

```
mean(Maths$MSC_Use) # mean
median(Maths$MSC_Use) # median
sd(Maths$MSC_Use) # standard deviation
var(Maths$MSC_Use) # variance
max(Maths$MSC_Use) # maximum
min(Maths$MSC_Use) # minimum
IQR(Maths$MSC_Use) # interquartile range
```

Descriptive Statistics by Group

```
tapply(Maths$MSC_Use, Maths$Grade, mean)
```

```
##         F         D         C         B         A
##  7.000000  0.500000  5.818182  4.843750 11.600000
```

```
tapply(Maths$MSC_Use, Maths$Gender, summary)
```

```
## $Female
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   4.500   9.706  16.500  42.000
##
## $Male
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   5.636   5.500 100.000
```

Tables (counts)

```
table(Maths$Gender)
```

```
##
## Female   Male
##     34     55
```

```
table(Maths$Gender, Maths$Grade)
```

```
##
##             F  D  C  B  A
##   Female    0  0  7 12 15
##   Male      1  4 15 20 15
```

Tables (Proportion)

```
prop.table(table(Maths$Gender, Maths$Grade))
```

```
##
##                   F           D           C           B
##   Female 0.00000000 0.00000000 0.07865169 0.13483146 0.1685
##   Male   0.01123596 0.04494382 0.16853933 0.22471910 0.1685
```

```
round(prop.table(table(Maths$Gender, Maths$Grade)),2)
```

```
##
##                F    D    C    B    A
##   Female 0.00 0.00 0.08 0.13 0.17
##   Male   0.01 0.04 0.17 0.22 0.17
```

Tables (Proportion)

```
require(magrittr)

table(Maths$Gender, Maths$Grade) %>%
  prop.table() %>% round(2)
```

```
##
##                F    D    C    B    A
##   Female 0.00 0.00 0.08 0.13 0.17
##   Male   0.01 0.04 0.17 0.22 0.17
```

Tables (Proportion)

```
table(Maths$Gender, Maths$Grade) %>%
  prop.table(1) %>% round(2)
```

```
##
##                F    D    C    B    A
##   Female 0.00 0.00 0.21 0.35 0.44
##   Male   0.02 0.07 0.27 0.36 0.27
```

Fancy Tables

```
# Create summaries
Maths_F = subset(Maths, Gender=="Female")
MSC_USE_F = summary(Maths_F$MSC_Use)
MSC_USE_F = as.numeric(MSC_USE_F)

Maths_M = subset(Maths, Gender=="Male")
MSC_USE_M = summary(Maths_M$MSC_Use)
MSC_USE_M = as.numeric(MSC_USE_M)

table_summary = rbind.data.frame(MSC_USE_F, MSC_USE_M)
table_summary = round(table_summary,1)

# Rename columns
colnames(table_summary) = c("Min", "1st Quartile", "Median",
                            "Mean", "3rd Quartile", "Max")
```

Fancy Tables

```
# Rename rows
rownames(table_summary) = c("Female MSC Visits",
                            "Male MSC Visits")

# Create Nice Table
kable(table_summary, align = 'c', format = "simple")
```

	Min	1st Quartile	Median	Mean	3rd Quartile	M
Female MSC Visits	0	1	4.5	9.7	16.5	4
Male MSC Visits	0	0	1.0	5.6	5.5	1(

Fancy Tables

```
kable(table_summary, align = 'c',
      caption = "Summary of MSC Visits by Gender",
      booktabs=TRUE) %>%
  kable_styling(font_size = 7)
```

Table 2: Summary of MSC Visits by Gender

	Min	1st Quartile	Median	Mean	3rd Quartile	Max
Female MSC Visits	0	1	4.5	9.7	16.5	42
Male MSC Visits	0	0	1.0	5.6	5.5	100

Fancy Tables

Table 3: Summary of Resource Usage by Gender

	Min	1st Quartile	Median	Mean	3rd Quartile	Max	NA
MSC Engagement							
Female MSC Visits	0	1	4.5	9.7	16.5	42	0
Male MSC Visits	0	0	1.0	5.6	5.5	100	0
VLE Engagement							
Female VLE Accesses	171	282	345.0	387.3	457.0	897	1
Male VLE Accesses	81	246	306.0	326.4	391.0	744	2

More on Tables

Helpful Resources:

- https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html
- https://bookdown.org/yihui/rmarkdown-cookbook/kableextra.html

Quick summary tables:

```
# install.packages("skimr")
# require(skimr)
# skim(Maths)
```

Section 6

Graphical Analysis - ggplot2

Graphical Analysis - ggplot2

- Two approaches to plotting: Base R and ggplot2

ggplot2 is a package that produces plots based on the grammar of graphics (Leland Wilkinson), the idea is that you can split the graph into components:

- a data set
- scale
- a co-ordinate system
- geoms (points, lines, bars, etc.)
- annotation

Graphical Analysis - ggplot2

ggplot2 is fully customisable

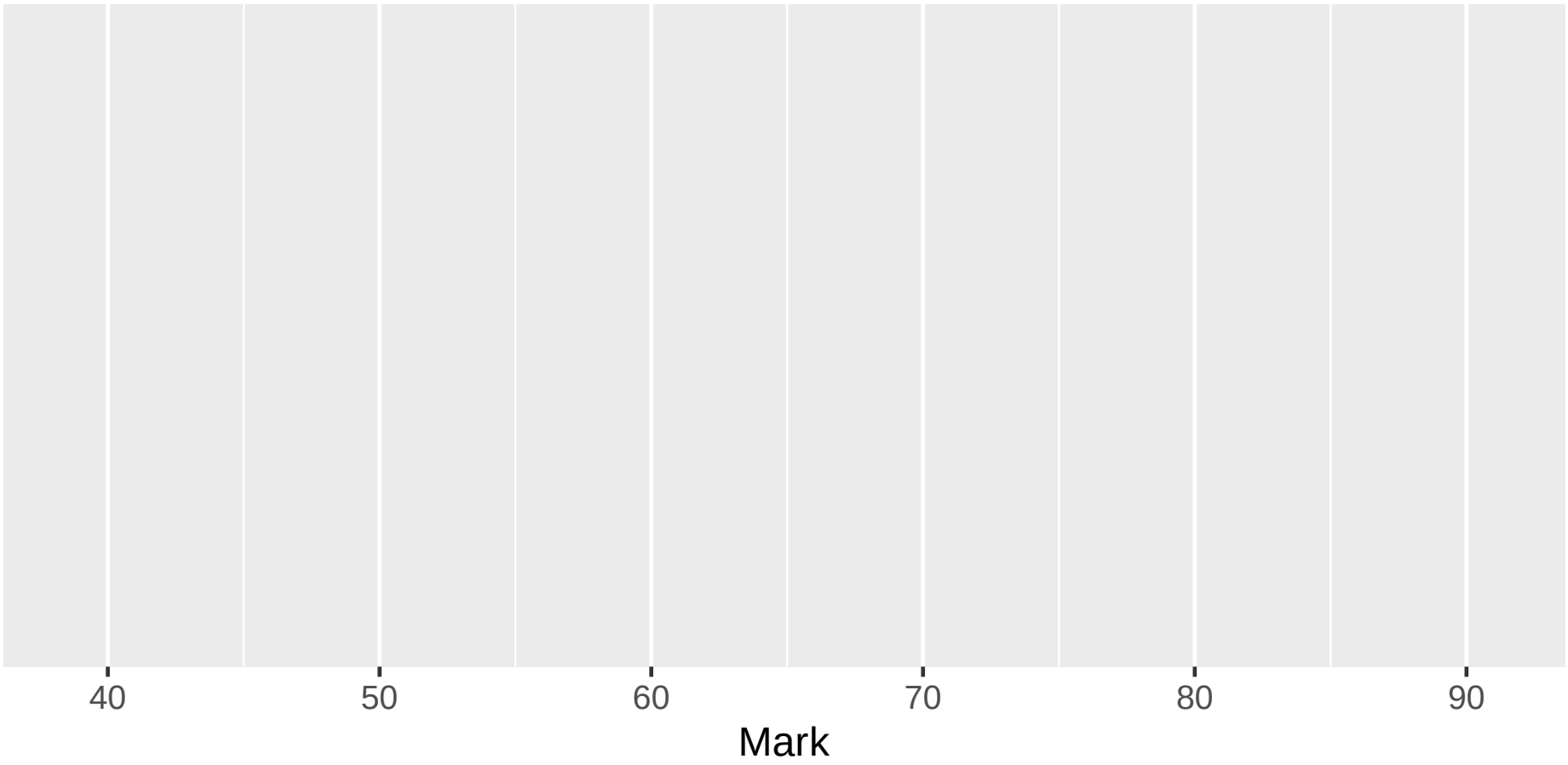
You can:

- colour geoms
- colour based on a factor
- manipulate all aspects of a legend
- adjust format of text (size, colour, style)

For more on ggplot2, see the ggplot2 cheatsheet

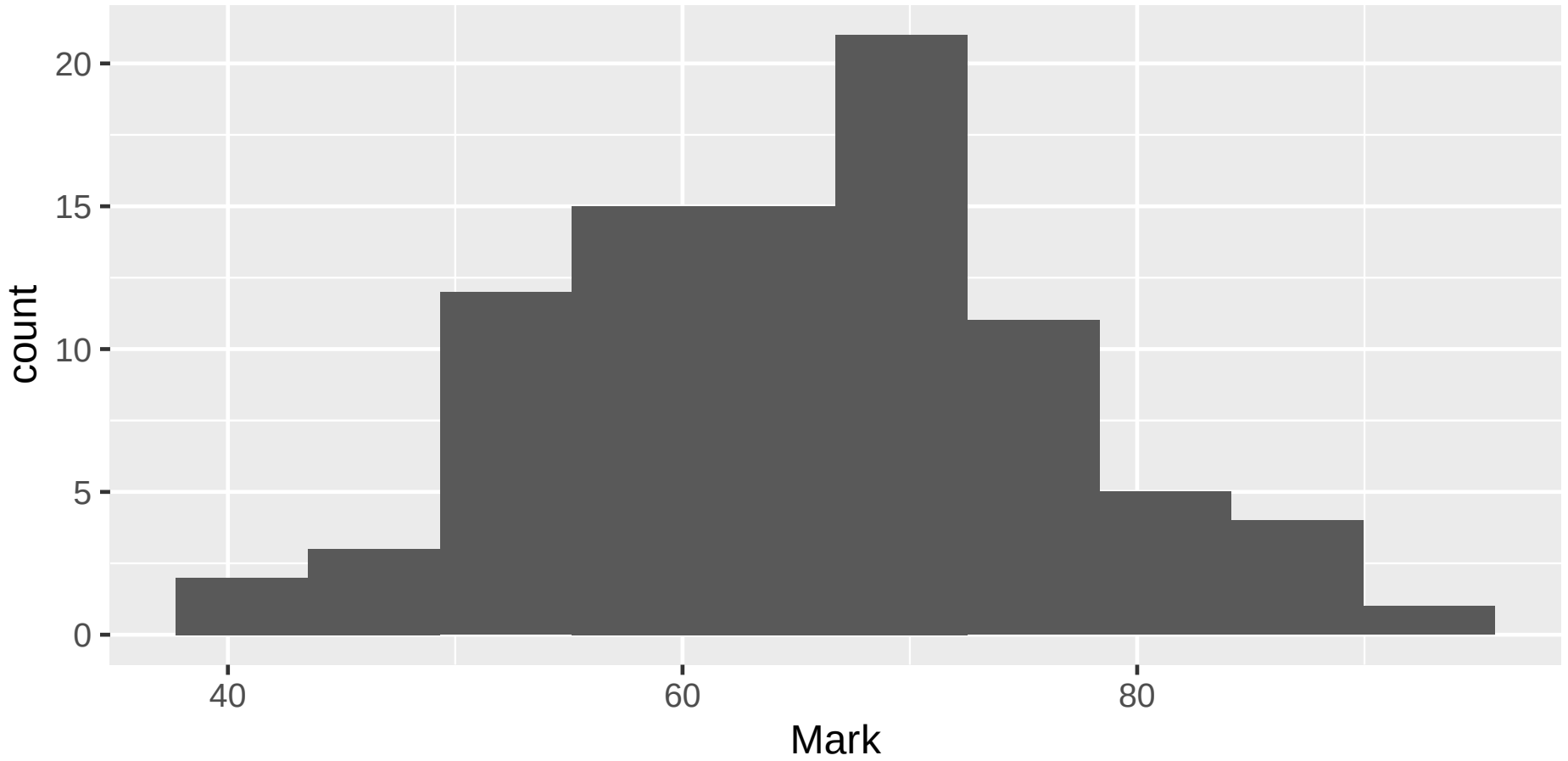
Histogram of Marks - create co-orinate system for data

```
ggplot(Maths, aes(Mark))
```



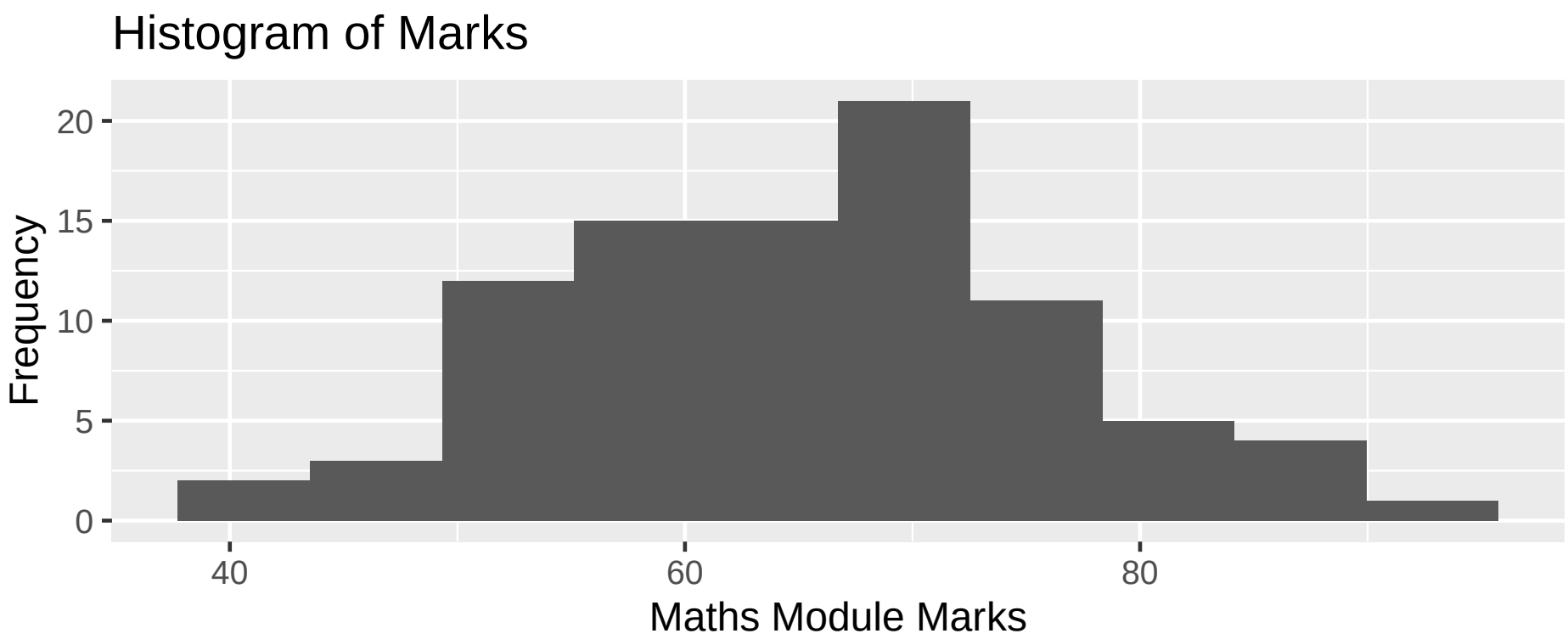
Histogram of Marks - add geom

```
ggplot(Maths, aes(Mark)) +
  geom_histogram(bins=10)
```



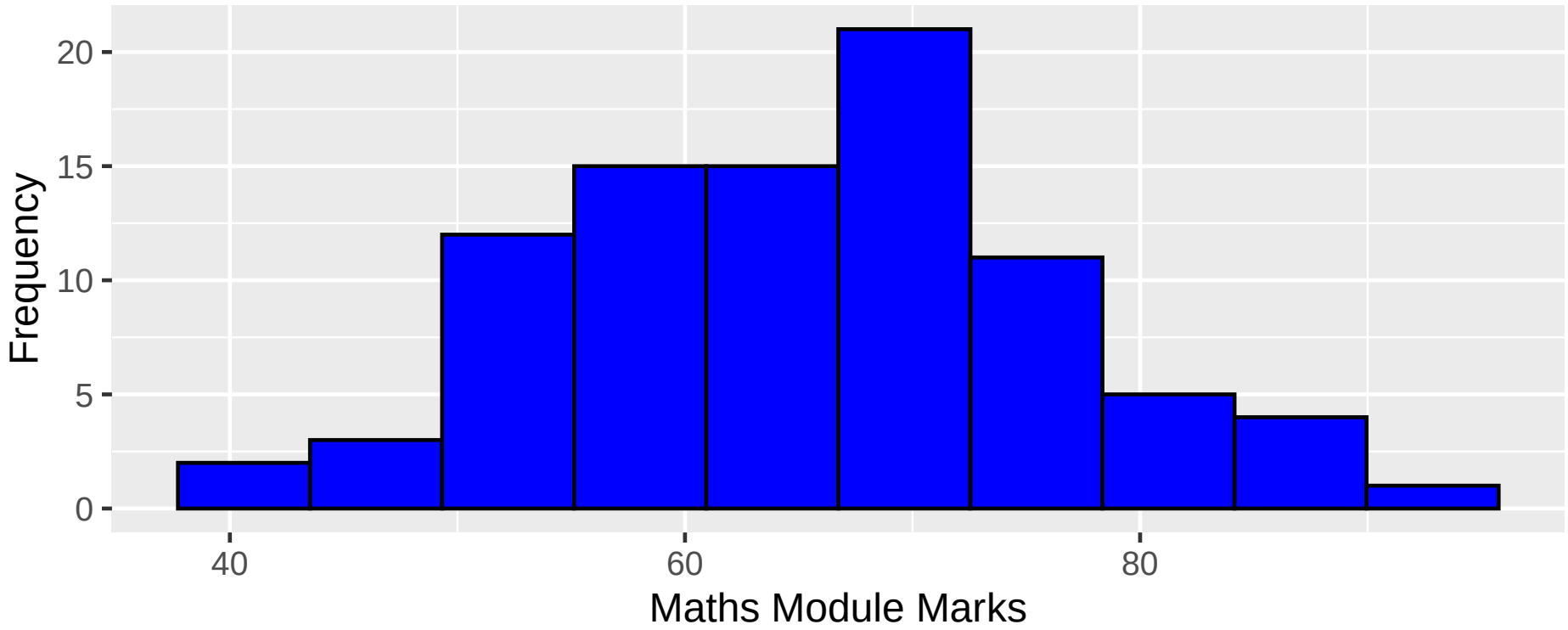
Histogram of Marks - add/change axis labels

```
ggplot(Maths, aes(Mark)) +
  geom_histogram(bins=10) +
  labs(x="Maths Module Marks", y="Frequency",
       title="Histogram of Marks")
```



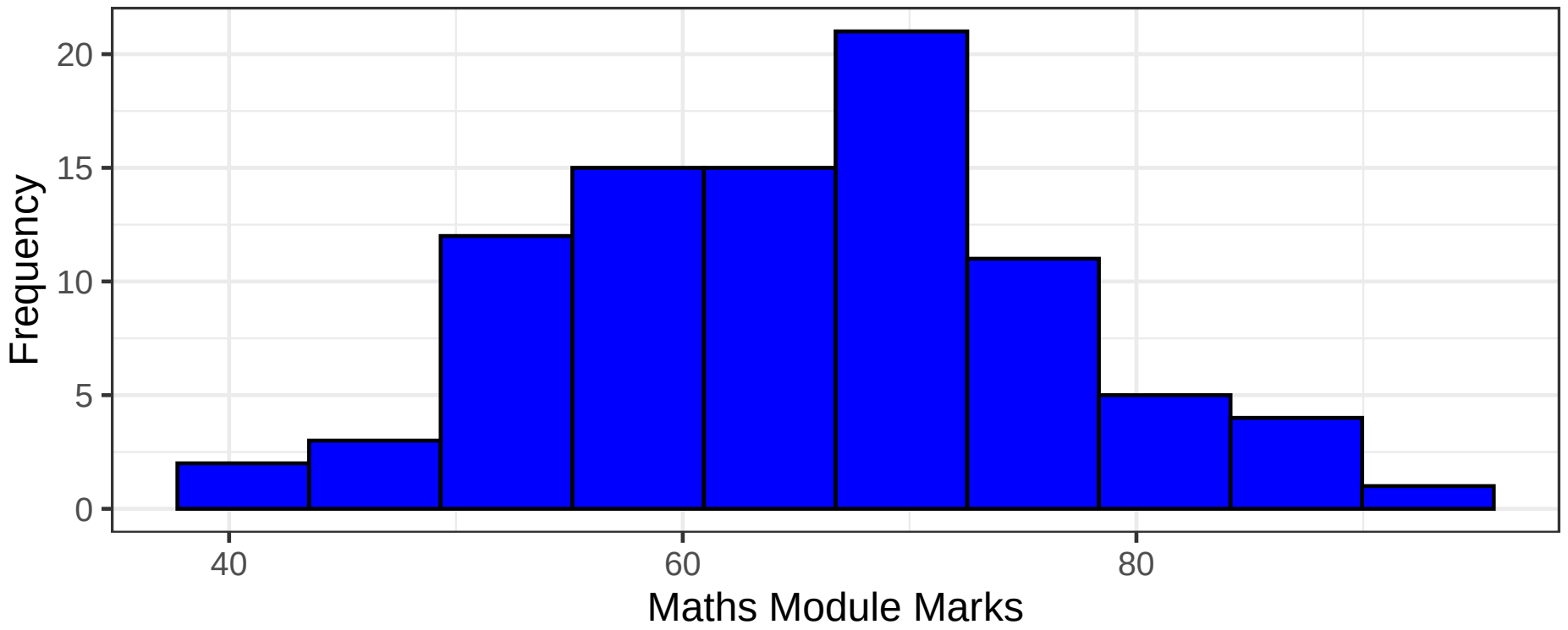
Histogram of Marks - add colour

```
ggplot(Maths, aes(Mark)) +
  geom_histogram(bins=10, fill="blue", color = "black") +
  labs(x="Maths Module Marks", y="Frequency")
```



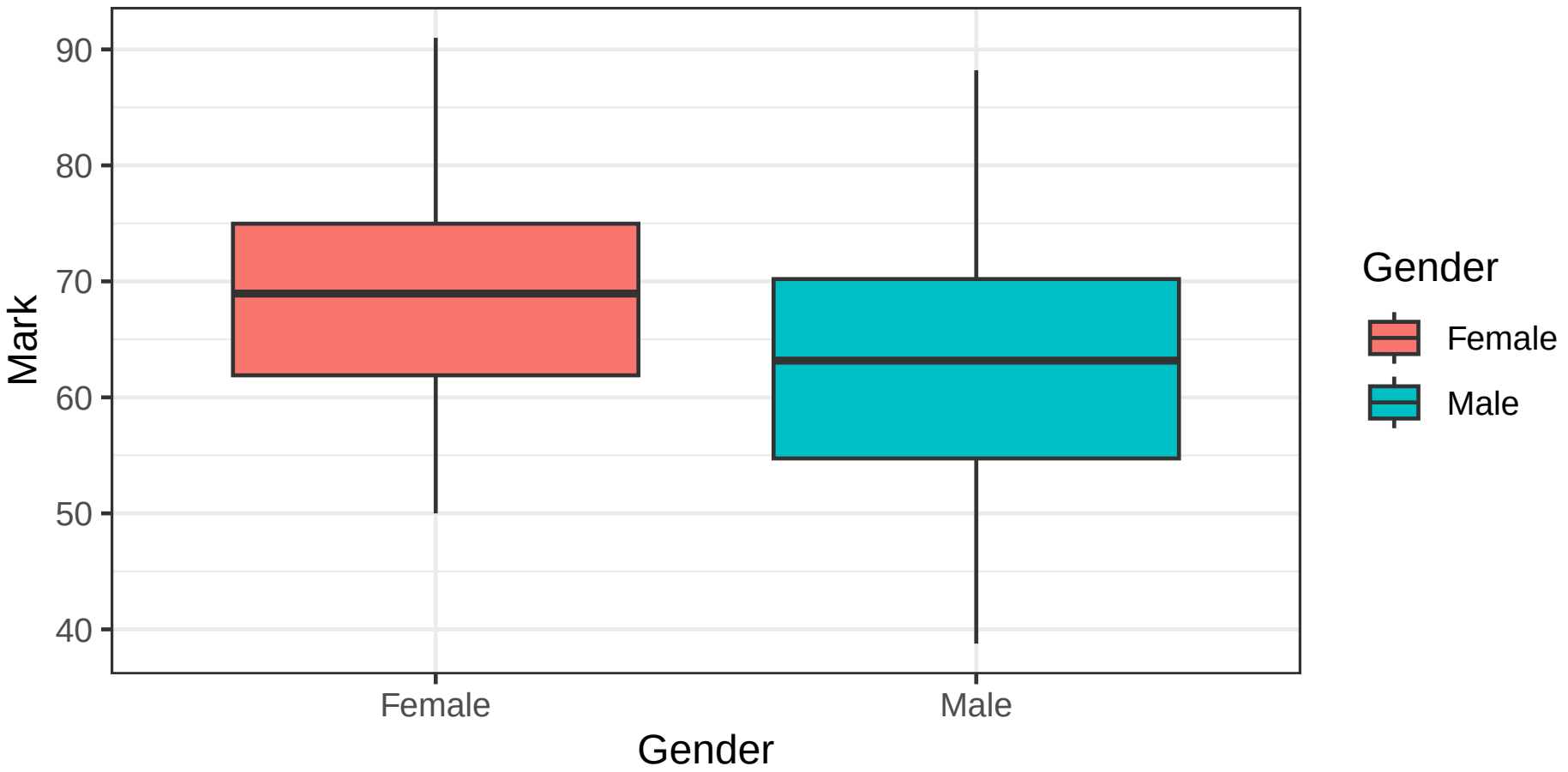
Histogram of Marks - change background

```
ggplot(Maths, aes(Mark)) +
    geom_histogram(bins=10, fill="blue", color = "black") +
  labs(x="Maths Module Marks", y="Frequency") +
  theme_bw()
```



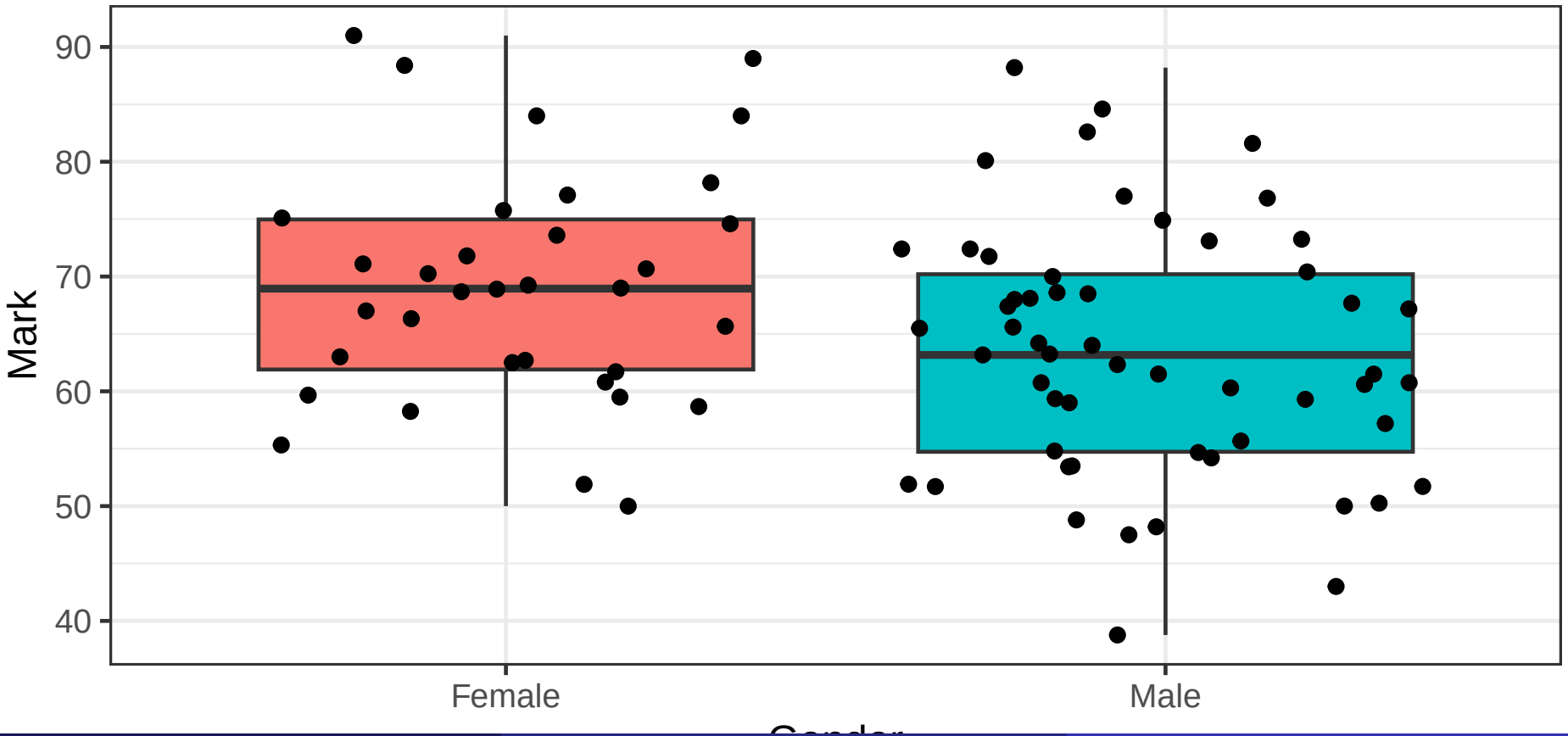
Boxplot of Marks ~ Gender

```
ggplot(Maths, aes(x=Gender, y=Mark, fill=Gender)) +
  geom_boxplot() + theme_bw()
```



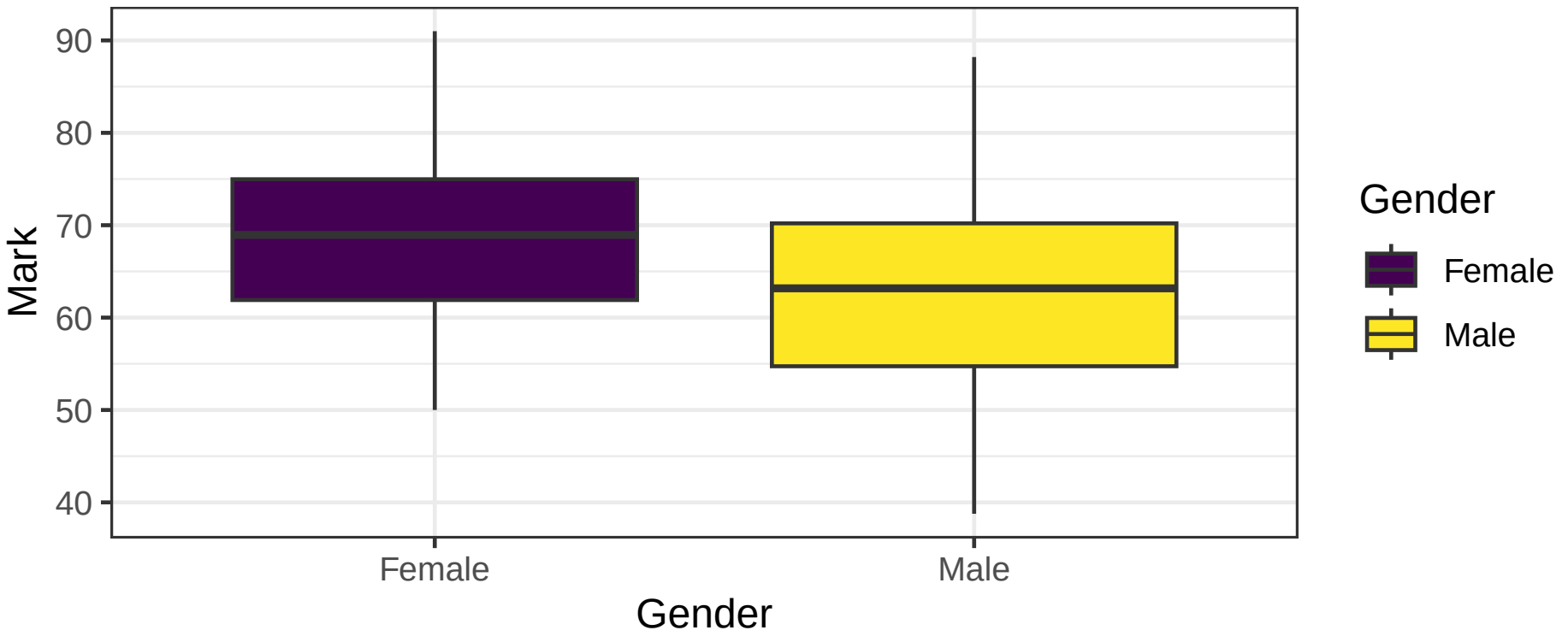
Boxplot of Marks ~ Gender (no legend)

```
ggplot(Maths, aes(x=Gender, y=Mark, fill=Gender)) +
  geom_boxplot() + theme_bw() +
  theme(legend.position = 'None') + geom_jitter()
```



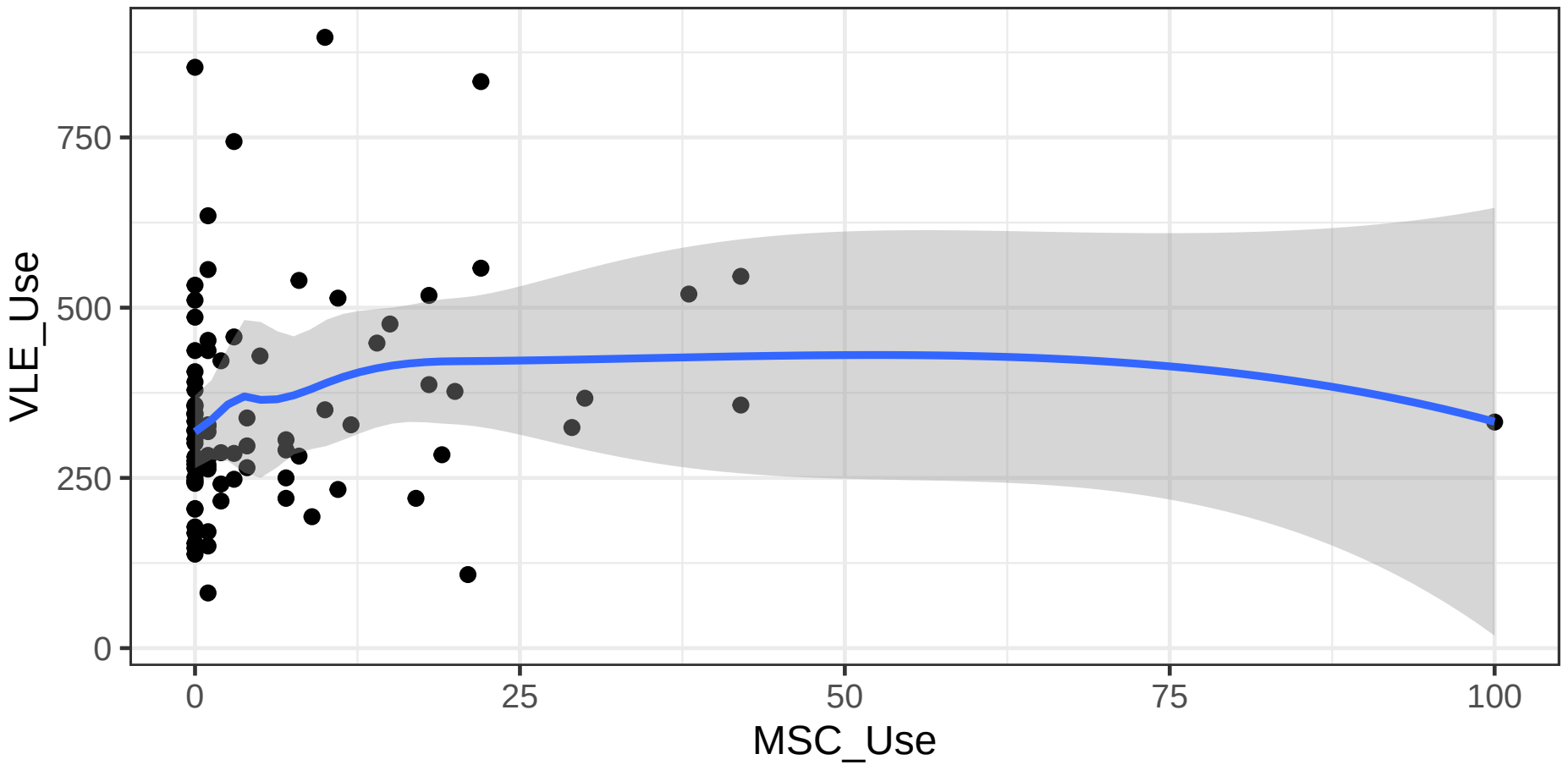
Boxplot of Marks ~ Gender (change colour palette)

```
require(viridis)
ggplot(Maths, aes(x=Gender, y=Mark, fill=Gender)) +
  geom_boxplot() + theme_bw() +
  scale_fill_viridis(discrete=TRUE, option = 'D')
```



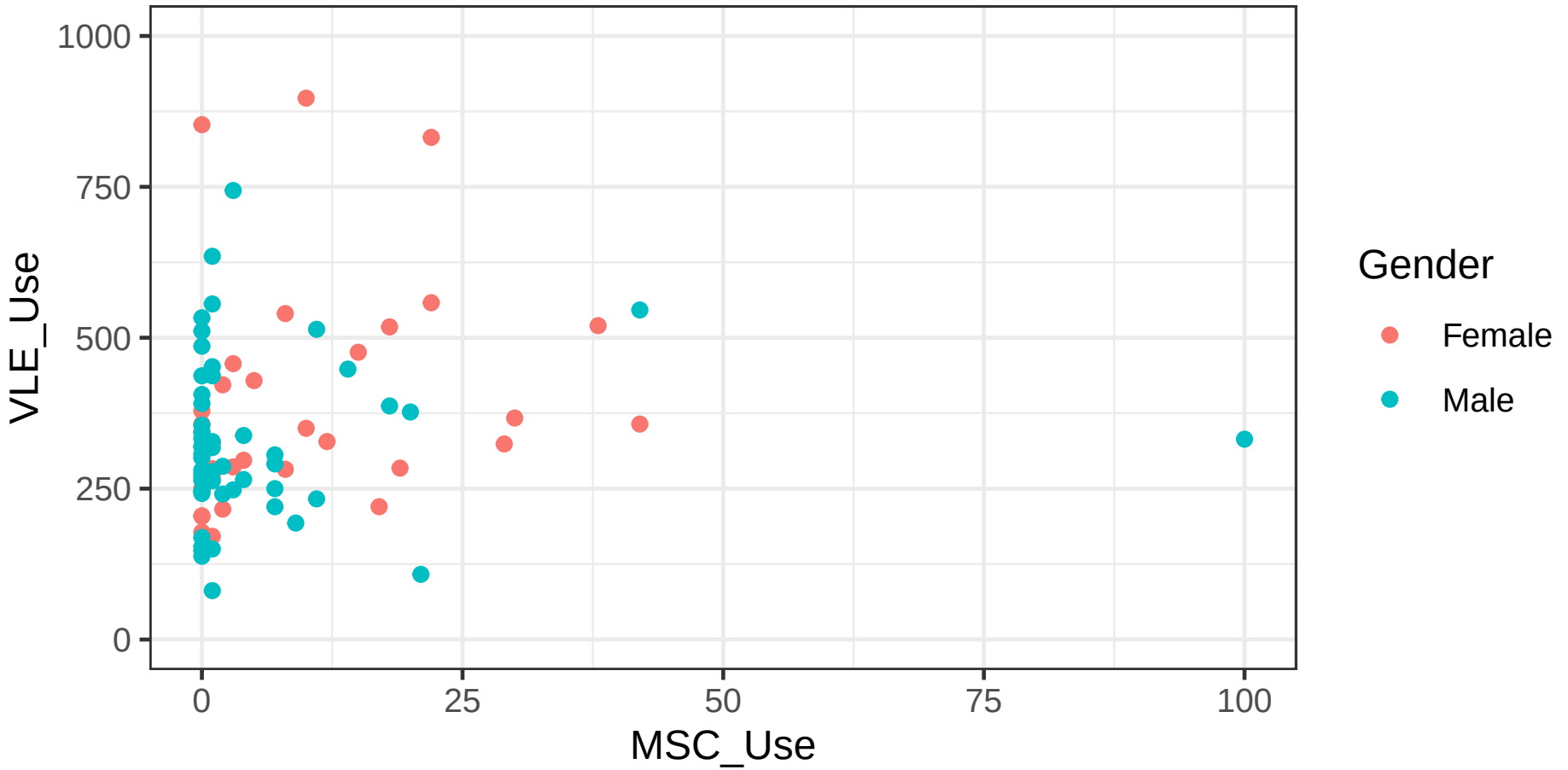
Scatterplot of VLE Use versus MSC Use

```
ggplot(Maths, aes(x=MSC_Use, y=VLE_Use))+
        geom_point() + theme_bw() + stat_smooth()
```



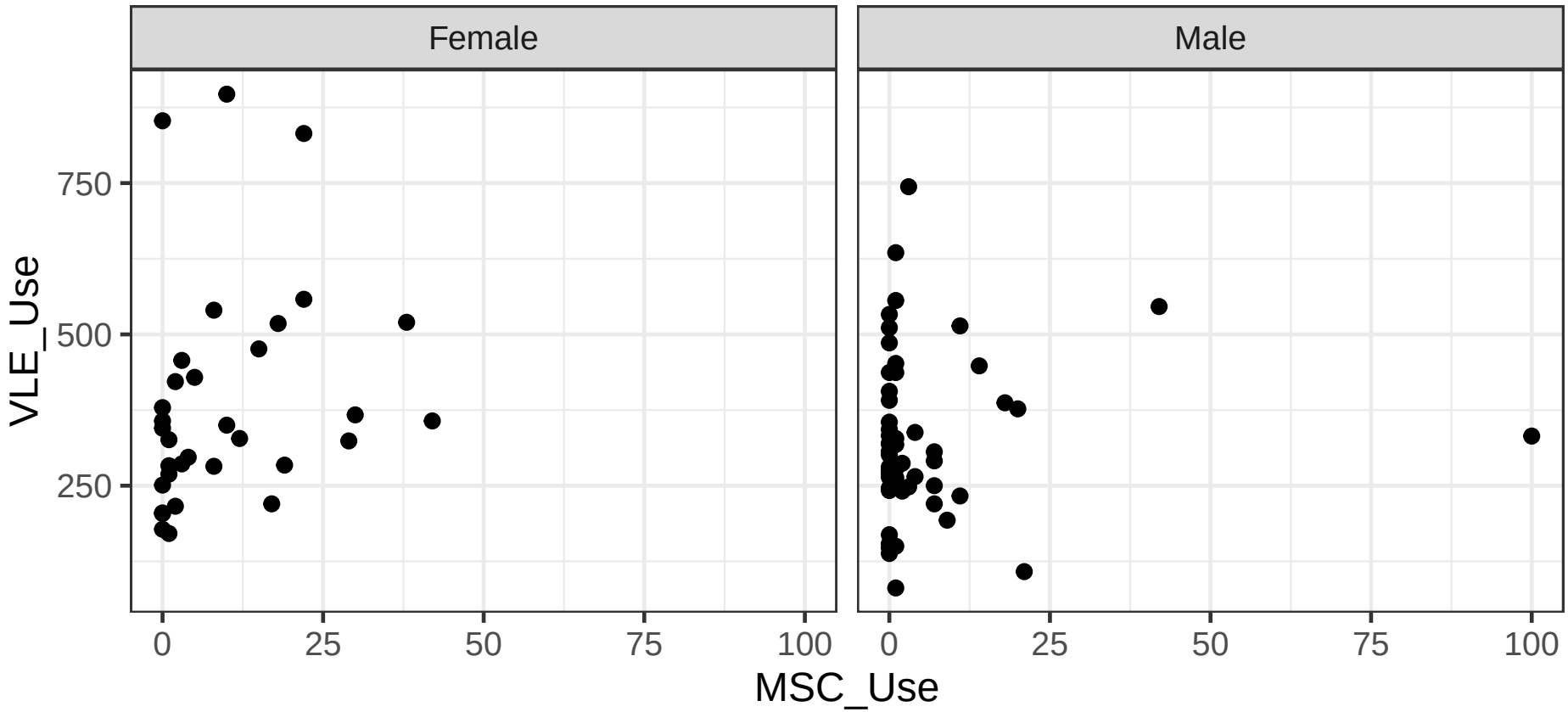
Scatterplot of VLE Use versus MSC Use

```
ggplot(Maths, aes(x=MSC_Use, y=VLE_Use, color=Gender)) +
        geom_point() + theme_bw() + ylim(c(0, 1000))
```



Scatterplot of VLE Use versus MSC Use

```
p <- ggplot(Maths, aes(x=MSC_Use, y=VLE_Use))+
        geom_point() + theme_bw()
p +facet_wrap(~`Gender`)
```



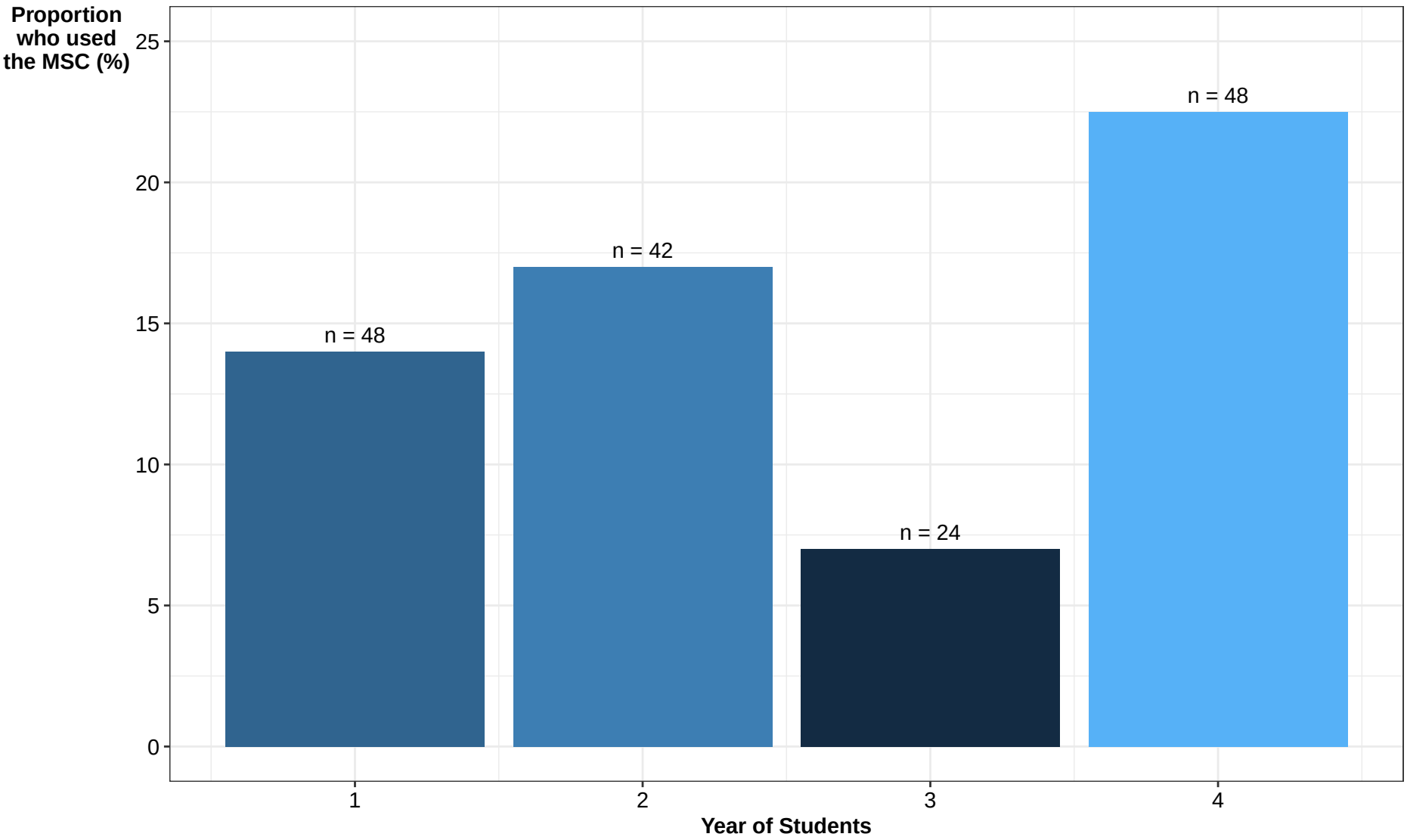
Interactive Scatterplot of VLE Use versus MSC Use

```
#install.packages("plotly")
require(plotly)

p<- ggplot(Maths, aes(x=MSC_Use, y=VLE_Use, color=Gender))+
        geom_point() + theme_bw()

ggplotly(p) # mkes it interactive
```

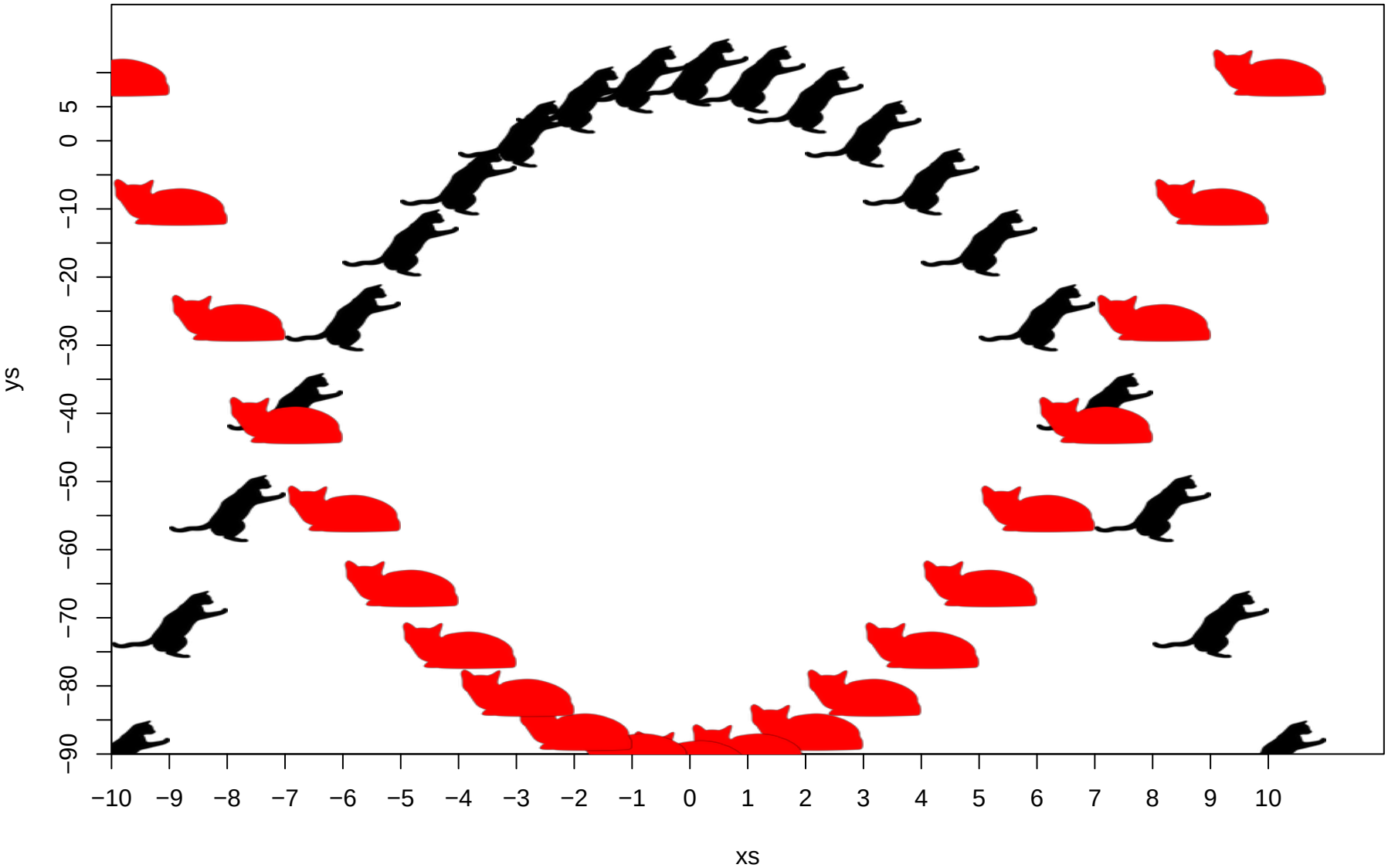
Barplot


Proportion who used the MSC (%)
25
20
15
10
5
0
n = 48
n = 42
n = 24
n = 48
1
2
3
4
Year of Students

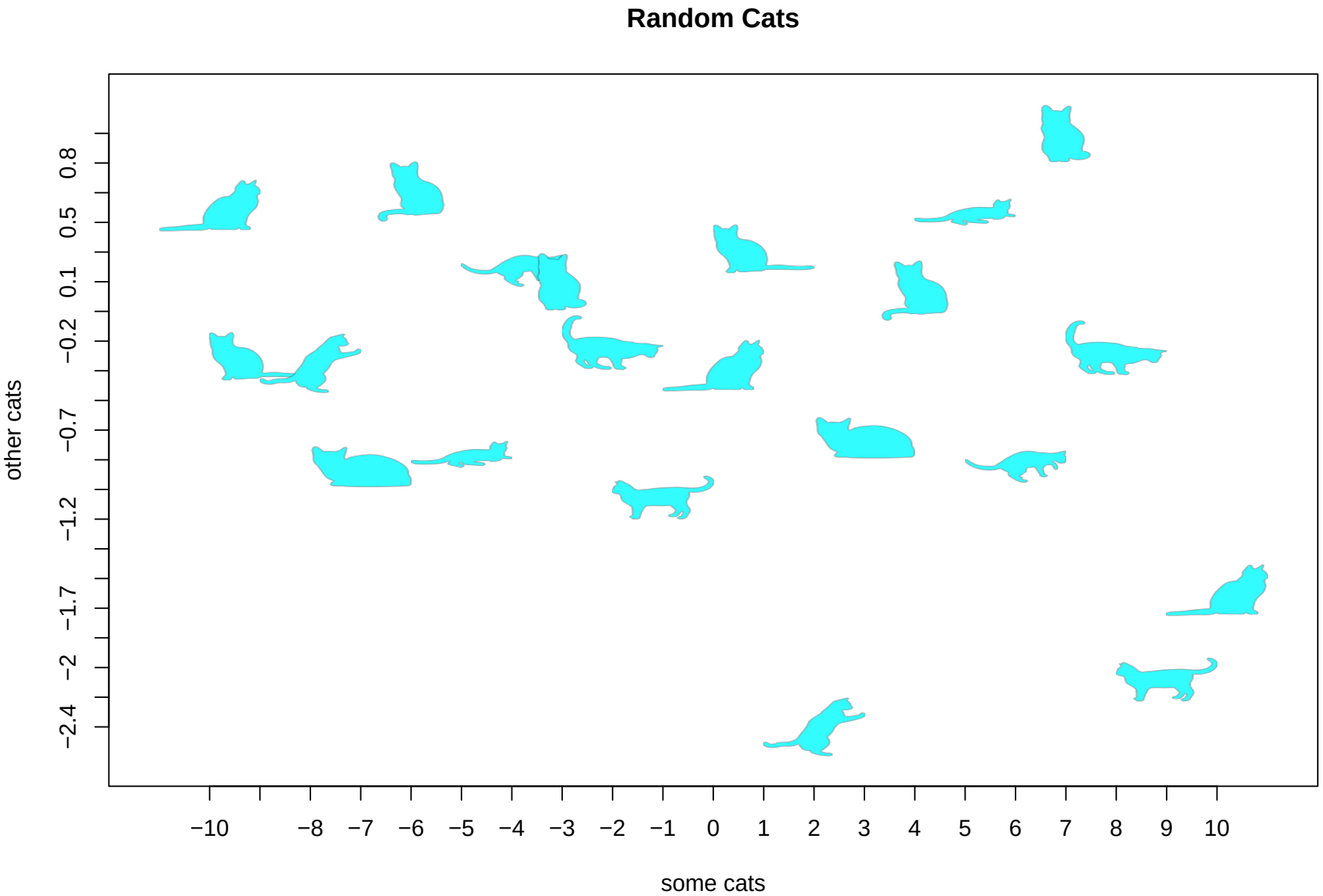
Section 7

Random and Cool Stuff

Catterplots



Catterplots



Beep Beep

```
#install.packages("beepr")
#library(beepr)
#beep(3)
#beep(4)
#beep(5)
#beep(0)
```

Wellbeing

```
#install.packages("praise")
require(praise)
praise()
```

```
## [1] "You are fantastic!"
```

```
praise()
```

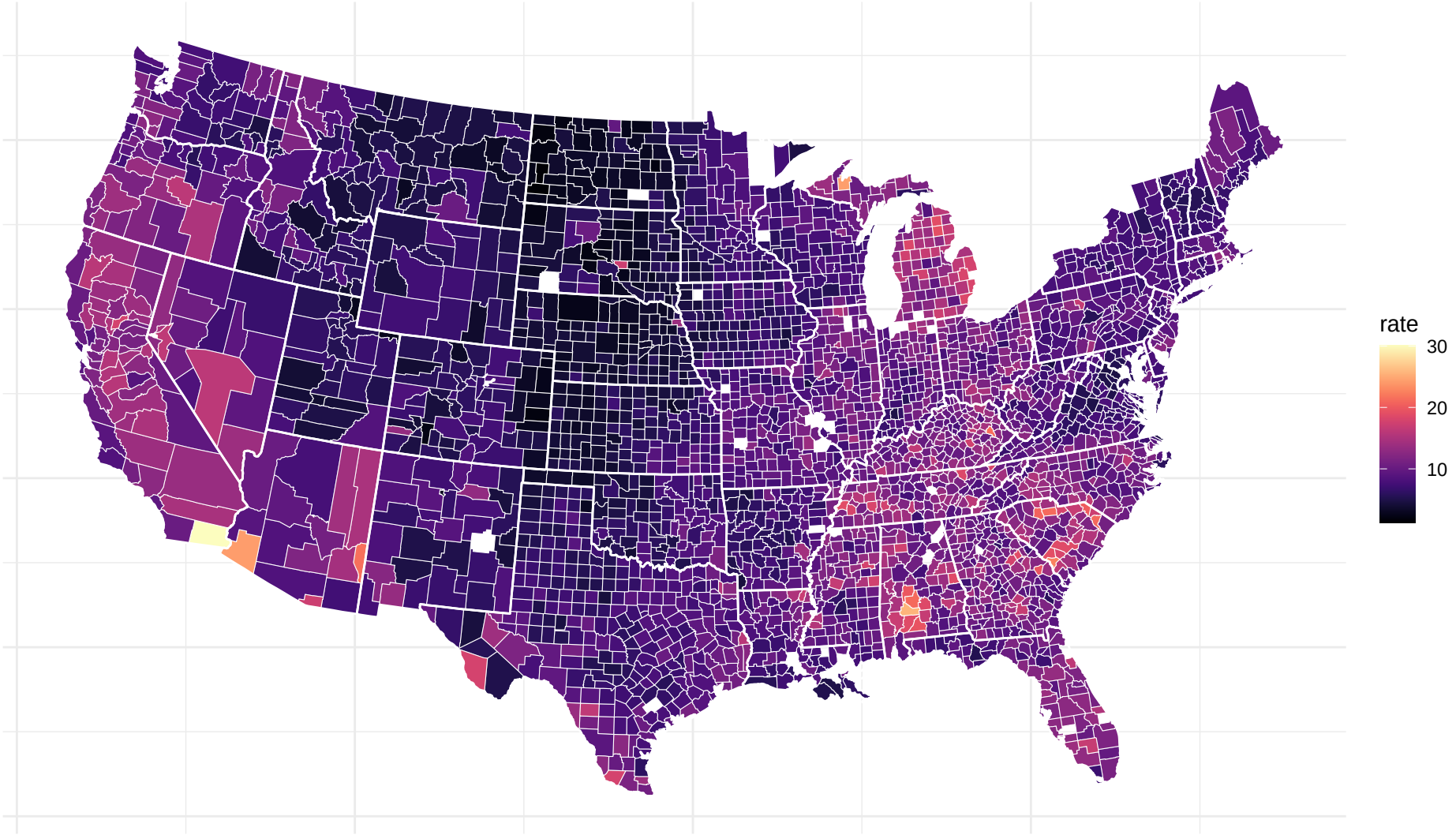
```
## [1] "You are priceless!"
```

```
praise()
```

```
## [1] "You are kryptonian!"
```

Maps

US unemployment rate by county



Interactive plots

```
library(gganimate)
require(gapminder)

# Taken from the gganimate examples: https://gganimate.com/
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour =
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  facet_wrap(~continent) +
  # Here comes the gganimate specific bits
  labs(title = 'Year: {frame_time}', x = 'GDP per capita', y =
  transition_time(year) +
  ease_aes('linear')
```

Galleries

Example of plots: https://r-graph-gallery.com/

Extensions to ggplot2: https://exts.ggplot2.tidyverse.org/gallery/

Examples of RShiny apps: https://shiny.rstudio.com/gallery/

Conclusion

Thank you for attending :)