

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Structure of collection . . . . .	4
<b>2</b>	<b>RMarkdown</b>	<b>4</b>
2.1	Formatting basics . . . . .	4
<b>3</b>	<b>Rstudio</b>	<b>5</b>
3.1	Useful packages . . . . .	5
3.2	Remove a package . . . . .	6
3.3	Import using Janitor . . . . .	6
3.4	Remove dataframe . . . . .	6
3.5	New vars by concatenation . . . . .	7
3.6	Save dataframe (CSV or Rdata) . . . . .	7
3.7	Save a diagram or plot . . . . .	7
3.8	Recode a text variable . . . . .	8
3.9	Alter variable names: . . . . .	8
<b>4</b>	<b>Data Wrangling and manipulation</b>	<b>9</b>
4.1	Bin variable ( e.g. Low/Medium/High) . . . . .	10
4.2	Conditional function . . . . .	10
4.3	Sum across rows . . . . .	10
4.4	Standardise variable . . . . .	10
4.5	Conditional Replacement . . . . .	11
4.6	replace a specific value . . . . .	11
4.7	Filter na's or retain complete cases . . . . .	11
4.8	Delete specified columns . . . . .	11
4.9	Change specific datapoint . . . . .	11
4.10	Work with dates . . . . .	11
4.11	Extract the last x num of a string . . . . .	11
4.12	Add an order (e.g. order of birth) . . . . .	12
4.13	Create a df with the names (and labels) of a dataframe . . . . .	12
4.14	Find duplicate rows . . . . .	12
4.15	Delete duplicate merged columns . . . . .	12
4.16	Impute missing values . . . . .	12
4.17	check for any na's in a df . . . . .	12
4.18	Keep rows based on a unique value. . . . .	13

4.19	Delete rows on a variable value . . . . .	13
4.20	Use if else to calculate on values . . . . .	13
4.21	Find the rowwise maximum and make a new variable . . . . .	13
4.22	Merge data frames (variables) . . . . .	13
4.23	Merge data frames (individuals) . . . . .	14
4.24	Create a new factor from existing . . . . .	15
4.25	change data types . . . . .	15
4.26	calculate dates and photoperiod . . . . .	15
<b>5</b>	<b>Statistical Analysis</b>	<b>16</b>
5.1	Regression . . . . .	16
5.2	Linear Regression . . . . .	16
5.3	Logistic Regression . . . . .	18
5.4	Principle Component Analysis . . . . .	18
5.5	Survival Analysis and Visualisation . . . . .	22
5.6	Receiver Operated Curves (ROC) . . . . .	22
5.7	Missing Values . . . . .	22
5.8	Pivot data longer . . . . .	22
5.9	Do analysis on multiple groups (e.g. dependent variable are many shape PCs) rstatix . . . . .	22
<b>6</b>	<b>Data Visualisation</b>	<b>23</b>
6.1	Packages needed . . . . .	23
6.2	Summary Tables . . . . .	23
6.3	Correlation matrix . . . . .	27
6.4	Graphing . . . . .	29
6.5	line defined by equation to scatterplot . . . . .	47
#—	#title: “Useful R syntax” #author: “Dr Chris McNeil” #site: bookdown::bookdown_site #document-class: book #output: # bookdown::gitbook: default # bookdown::pdf_book: default #—	

## 1 Introduction

This document is a collection of useful code for Rmarkdowm and R

I have used the mtcars dataset if possible

I have used the Tidyverse and the pipe ( %>% ) if possible

I recommend that the code is checked for warnings that is is not depreciated



Figure 1: Don't Panic

## 1.1 Structure of collection

- 1. Rmarkdown
  - formatting basics
- 2. Rstudio
  - load/unload packages
  - print figures to files
  - Libraries/packages
  - essential/useful packages
- 3. Data wrangling
  - load dataset
  - clean environment
  - check for duplicates
  - Merging datasheets
  - Merging datasets
  - Reshaping
  - recode factors
  - dealing with missing data
  - Data reduction with PCA
  - Data standardisation
- 4. Statistical analysis
- 5. Data Visualisation
  - Tables
  - Plots

## 2 RMarkdown

This chapter contains syntax for the non-code rmarkdown sections.

### 2.1 Formatting basics

```
*** on its own, for a horizontal line
**text** for bold
*text* for italics
1. Item 1
2. Item 2
3. Item 3
  + Item 3a
  + Item 3b for ordered lists

[linked phrase](http://example.com) for links
![alt text](figures/img.png) for images

### R chunk basics
message=FALSE, warning=FALSE, include=FALSE, ECHO=FALSE (show output),
```

```
To set document default knitr::opts_chunk$set(echo=FALSE)
```

```
To render this book:
```

```
bookdown::render_book(output_format = 'all')
```

## 3 Rstudio

This chapter contains syntax for manipulating data and packages within the R studio environment.

### 3.1 Useful packages

Load all libraries

```
library(tidyverse) # data handling and viz
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## vforcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyrr    1.3.0
## v purrr    1.0.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(janitor) #dataframe import cleaning
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(knitr) #nice html tables
library(kableExtra) # nicer knitr tables
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(broom)
library(readr) # load csv stored data
library(geosphere) # for calc daylength
```

```
## The legacy packages maptools, rgdal, and rgeos, underpinning this package
## will retire shortly. Please refer to R-spatial evolution reports on
## https://r-spatial.org/r/2023/05/15/evolution4.html for details.
## This package is now running under evolution status 0
```

### 3.2 Remove a package

```
#Unload a module:
library(clipr) #load

## Welcome to clipr. See ?write_clip for advisories on writing to the clipboard in R.

detach(package:clipr) #unload
```

### 3.3 Import using Janitor

```
# Create a data.frame with dirty names
test_df <- as.data.frame(matrix(ncol = 6))
names(test_df) <- c("firstName", "ábc@!*", "% successful (2009)",
                     "REPEAT VALUE", "REPEAT VALUE", "")
head(test_df)

##   firstName ábc@!* % successful (2009) REPEAT VALUE REPEAT VALUE
## 1          NA          NA             NA          NA          NA NA

test_df <- test_df %>%
  clean_names()
head(test_df)

##   first_name abc_percent_successful_2009 repeat_value repeat_value_2  x
## 1          NA      NA                 NA          NA          NA NA
```

Reference

### 3.4 Remove dataframe

```
data("mtcars")
data("band_instruments")
data("band_instruments2") # Load example datasets
```

```

rm(list=ls()[! ls() %in% c("band_instruments", "band_instruments2")])
# Everything except Band instruments
rm(list=setdiff(ls(), "band_instruments")) # Everything except "bandinstruments"
rm(list=ls()) # Remove everything

```

Reference:Stackoverflow

### 3.5 New vars by concatenation

```
# USE rbind to add new variables to dataset by combination
```

### 3.6 Save dataframe (CSV or Rdata)

*make date string*

```
datenow <- format(Sys.time(), "%Y_%m_%d")
date
```

```
## function (x)
## {
##   UseMethod("date")
## }
## <bytecode: 0x000001ce93b332f8>
## <environment: namespace:lubridate>
```

```
data(mtcars)
```

*Write file names*

```
#create data directory
dir.create("data_out")
```

```
## Warning in dir.create("data_out"): 'data_out' already exists
```

```
filenamecsv <- paste("data_out/mtcsvdata", datenow, ".csv", sep="")
filenamerda <- paste("data_out/mtrdadata", datenow, ".rda", sep="")
```

*Save the files as CSV files or as R data files*

```
save(mtcars, file=filenamerda)
write.csv(mtcars, file=filenamecsv)
```

### 3.7 Save a diagram or plot

name	band
Mick	Stones
John	Beatles
Paul	Beatles

name	band
m	Stones
John	Beatles
Paul	Beatles

```
plot1 <- mtcars %>% ggplot(aes(hp,qsec)) + geom_point()
#plot1 #print plot if required
pdf("plot.pdf")
plot1
dev.off()
```

```
## pdf
## 2
```

```
pdf('device' off.
```

### 3.8 Recode a text variable

```
data("band_members")
kable(head(band_members)) %>% kable_minimal(full_width = F)
```

```
band_members <- band_members %>% mutate(name=recode(name, "Mick"= "m"))
kable(head((band_members))) %>% kable_minimal(full_width = F)
```

```
rm(list=ls()) # Remove everything
```

Reference: Kable Extra

### 3.9 Alter variable names:

*Remove underscores*

```
data("mtcars")
mtcars <- mtcars %>% dplyr::rename(hp_new=hp)
kable(head((mtcars))) %>% kable_minimal(full_width = F)
```

```
mtcars <- mtcars %>% rename_with(.fn = ~str_replace(., "_", ""))
kable(head((mtcars))) %>% kable_minimal(full_width = F)
```

	mpg	cyl	disp	hp_new	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

	mpg	cyl	disp	hpnew	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

### 3.9.1 list datasets available

```
#data() # list all available datasets
data("diamonds")
```

### 3.9.2 Render book

## 4 Data Wrangling and manipulation

```
library(Hmisc) #impute values

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
## 
##     src, summarize

## The following objects are masked from 'package:base':
## 
##     format.pval, units

library(nanar) # deal with NAs
library(geosphere)
library(tidyverse) # data handling and viz
library(janitor) #dataframe import cleaning
library(knitr) #nice html tables
library(kableExtra) # nicer knitr tables
library(broom)
library(readr) # load csv stored data
library(geosphere) # for calc daylength
library(lubridate)
```

## 4.1 Bin variable ( e.g. Low/Medium/High)

```
data(mtcars)
mtcars <- mtcars %>% mutate(hp_cat=cut(hp, breaks=c(-Inf, 100, Inf),
                                             labels=c("low hp","high hp")))
```

## 4.2 Conditional function

```
mtcars <- mtcars %>% mutate(loghp=ifelse(cyl>4,log10(hp),NA))
# Nonsensical example, but log transformed all horse powers of cars with more
# than four cylinders
```

## 4.3 Sum across rows

```
mtcars <- mtcars %>% mutate(sum = select(., disp:drat) %>%
apply(1, sum, na.rm=TRUE))
# apply() takes Data frame or matrix as an input and gives output in vector
#(i.e.many columns to one list)
# the '1' sets the dataframe to use (already selected here)
```

Reference

## 4.4 Standardise variable

```
dat2 <- mtcars %>%
  as_tibble() %>%
  mutate(across(where(is.numeric), scale))

funcs <- list(mean = ~mean(.x,na.rm = TRUE),
  sd = ~sd(.x,na.rm = TRUE)
)
dat2 %>% dplyr::summarise(across(where(is.numeric), funcs))

## # A tibble: 1 x 26
##   mpg_mean mpg_sd cyl_mean cyl_sd disp_mean disp_sd hp_mean hp_sd drat_mean
##   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 7.11e-17    1 -1.47e-17    1 -9.08e-17    1 1.04e-17    1 -2.92e-16
## # i 17 more variables: drat_sd <dbl>, wt_mean <dbl>, wt_sd <dbl>,
## #   qsec_mean <dbl>, qsec_sd <dbl>, vs_mean <dbl>, vs_sd <dbl>, am_mean <dbl>,
## #   am_sd <dbl>, gear_mean <dbl>, gear_sd <dbl>, carb_mean <dbl>,
## #   carb_sd <dbl>, loghp_mean <dbl>, loghp_sd <dbl>, sum_mean <dbl>,
## #   sum_sd <dbl>
```

## 4.5 Conditional Replacement

Replace all 'NA's in a specified variable with 0.

```
mtcars <- mtcars %>% mutate(loghp1 = coalesce(loghp, 0))
```

## 4.6 replace a specific value

```
#ageandheight [row number, column number] = [new value]
```

## 4.7 Filter na's or retain complete cases

```
mtcars <- mtcars %>% filter(!is.na(hp)) # no missing values found  
mtcars <- mtcars %>% filter(complete.cases(.)) # no missing values found
```

## 4.8 Delete specified columns

```
mtcars1 <- mtcars %>% select(-(drat)) # single column  
mtcars2 <- mtcars %>% select(-c(drat, hp, vs, gear)) # multiple columns  
  
rm(list=setdiff(ls(), "mtcars")) # clean environment
```

## 4.9 Change specific datapoint

```
mtcarsmissingvalues <- mtcars %>% mutate(gear=ifelse(gear==5, "missing", gear))
```

## 4.10 Work with dates

Add a date column

```
mtcars <- mtcars %>% mutate(date=ymd("2001-05-24"))
```

## 4.11 Extract the last x num of a string

```
mtcars <- mtcars %>% rownames_to_column("car_name")  
  
mtcars <- mtcars %>% mutate(last3letters=str_sub(car_name, -3))
```

	Var1	Freq
1	10.4	2
6	15.2	2
14	19.2	2
16	21	2

## 4.12 Add an order (e.g. order of birth)

Calculate where in order of siblings participant was born

create dedicated df for dob calcs

add an order no

## 4.13 Create a df with the names (and labels) of a dataframe

## 4.14 Find duplicate rows

```
# specify which variable to check for duplication
n_occur1 <- data.frame(table(mtcars$mpg))
kable(n_occur1[n_occur1$Freq > 1,]) %>% kable_styling(full_width = F) %>%
  kable_minimal()
```

## 4.15 Delete duplicate merged columns

## 4.16 Impute missing values

### 4.16.1 Imputing missing values using the mean:

```
#create missing values
#mtcarsmissingvalues <- mtcars %>% mutate(gear=ifelse(gear==5, "", gear))

mtcarsmissingvalues <- mtcars %>% replace_with_na(replace = list(gear = 5))
mtcarsmissingvalues$gear <- impute(mtcarsmissingvalues$gear, mean) # replace with mean
mtcarsmissingvalues$gear <- impute(mtcarsmissingvalues$gear, median) # median
mtcarsmissingvalues$gear <- impute(mtcarsmissingvalues$gear, 4) # replace specific number
```

Reference:

## 4.17 check for any na's in a df

```
which(complete.cases(mtcars) == FALSE)

## integer(0)
```

## 4.18 Keep rows based on a unique value.

e.g. prescription code

```
mtcarsdistinct <- mtcars %>%     distinct(cyl, .keep_all= TRUE)
```

Reference

## 4.19 Delete rows on a variable value

```
mtcars1<-mtcars %>% filter(!(cyl==6))
mtcars2<-mtcars %>% filter(!(cyl==6 | hp==180)) # / is the 'or' operator
mtcars3<-mtcars %>% filter(!(cyl==8 & hp==215)) # & is the 'and' operator
# remove the ! To select the individuals with the specified conditions
```

## 4.20 Use if else to calculate on values

```
# no NA's so all values unchanged.
mtcars <- mtcars %>% mutate(vs=ifelse(is.na(vs),(carb-am)/365.25,vs))
```

## 4.21 Find the rowwise maximum and make a new variable

### 4.22 Merge data frames (variables)

\*left\_join(x, y): returns all rows from x, and all columns from x and y. Rows in x with no match in y will have NA values in the new columns. If there are multiple matches between x and y, all combinations of the matches are returned.

\*inner\_join(x, y): returns all rows from x where there are matching values in y, and all columns from x and y. If there are multiple matches between x and y, all combinations of the matches are returned.

\*full\_join(x, y): returns all rows and all columns from both x and y. Where there are not matching values, the function returns NA for the one missing

- inner: only rows with matching keys in both x and y
- left: all rows in x, adding matching columns from y
- right: all rows in y, adding matching columns from x
- full: all rows in x with matching columns in y, then the rows of y that don't match x.

```
# prepare new dataset
# make the rownames into a 'joinable' column
mtcars <- mtcars %>% mutate(carnames=rownames(mtcars))
mtcars_extradata <- mtcars %>% select(cyl)
# make the rownames into a 'joinable' column
mtcars_extradata <- mtcars_extradata %>%
  mutate(carnames=rownames(mtcars_extradata))
mtcars_extradata <- mtcars_extradata %>% mutate(values=cyl*4)
mtcars_extradata <- mtcars_extradata %>% select(-cyl)
```

	carnames	valves
1	24	
2	24	
3	24	
4	32	
5	24	
6	32	

carb	hp_cat	loghp	sum	loghp1	date	last3letters	carnames	valves
4	high hp	2.041393	273.90	2.041393	2001-05-24	RX4	1	24
4	high hp	2.041393	273.90	2.041393	2001-05-24	Wag	2	24
1	high hp	2.041393	371.08	2.041393	2001-05-24	ive	3	24
2	high hp	2.243038	538.15	2.243038	2001-05-24	out	4	32
1	high hp	2.021189	332.76	2.021189	2001-05-24	ant	5	24
4	high hp	2.389166	608.21	2.389166	2001-05-24	360	6	32

```
kable(glimpse(mtcars_extradata %>% slice(1:6))) %>%
  kable_styling(full_width = F) %>%
  kable_minimal()
```

```
## Rows: 6
## Columns: 2
## $ carnames <chr> "1", "2", "3", "4", "5", "6"
## $ valves    <dbl> 24, 24, 24, 32, 24, 32

mtcars <- left_join(mtcars, mtcars_extradata, by = 'carnames')

kable(glimpse(mtcars %>% select(carb:valves) %>% slice(1:6))) %>%
  kable_styling(full_width = F) %>%
  kable_minimal()

## Rows: 6
## Columns: 9
## $ carb        <dbl> 4, 4, 1, 2, 1, 4
## $ hp_cat     <fct> high hp, high hp, high hp, high hp, high hp, high hp
## $ loghp       <dbl> 2.041393, 2.041393, 2.041393, 2.243038, 2.021189, 2.389166
## $ sum         <dbl> 273.90, 273.90, 371.08, 538.15, 332.76, 608.21
## $ loghp1      <dbl> 2.041393, 2.041393, 2.041393, 2.243038, 2.021189, 2.389166
## $ date        <date> 2001-05-24, 2001-05-24, 2001-05-24, 2001-05-24, 2001-05-2-
## $ last3letters <chr> "RX4", "Wag", "ive", "out", "ant", "360"
## $ carnames    <chr> "1", "2", "3", "4", "5", "6"
## $ valves       <dbl> 24, 24, 24, 32, 24, 32
```

## 4.23 Merge data frames (individuals)

```
mtcarsmerged <- bind_rows(mtcars2, mtcars3)
rm(list=setdiff(ls(), "mtcars")) # clean environment
```

Reference

## 4.24 Create a new factor from existing

```
mtcars <- mtcars %>% mutate(cyc_carb = paste(cyl,carb,sep="-"))
```

## 4.25 change data types

(merging fails if data types are different)

```
# adni_demog<-adni_demog %>% mutate(age_scan=as.numeric(age_scan))
# ukbb<-ukbb %>% mutate(scan_no=as.numeric(scan_no))
```

## 4.26 calculate dates and photoperiod

(using geosphere library)

```
#import sample dataset
dateslat <- read_csv("dateslat.csv")

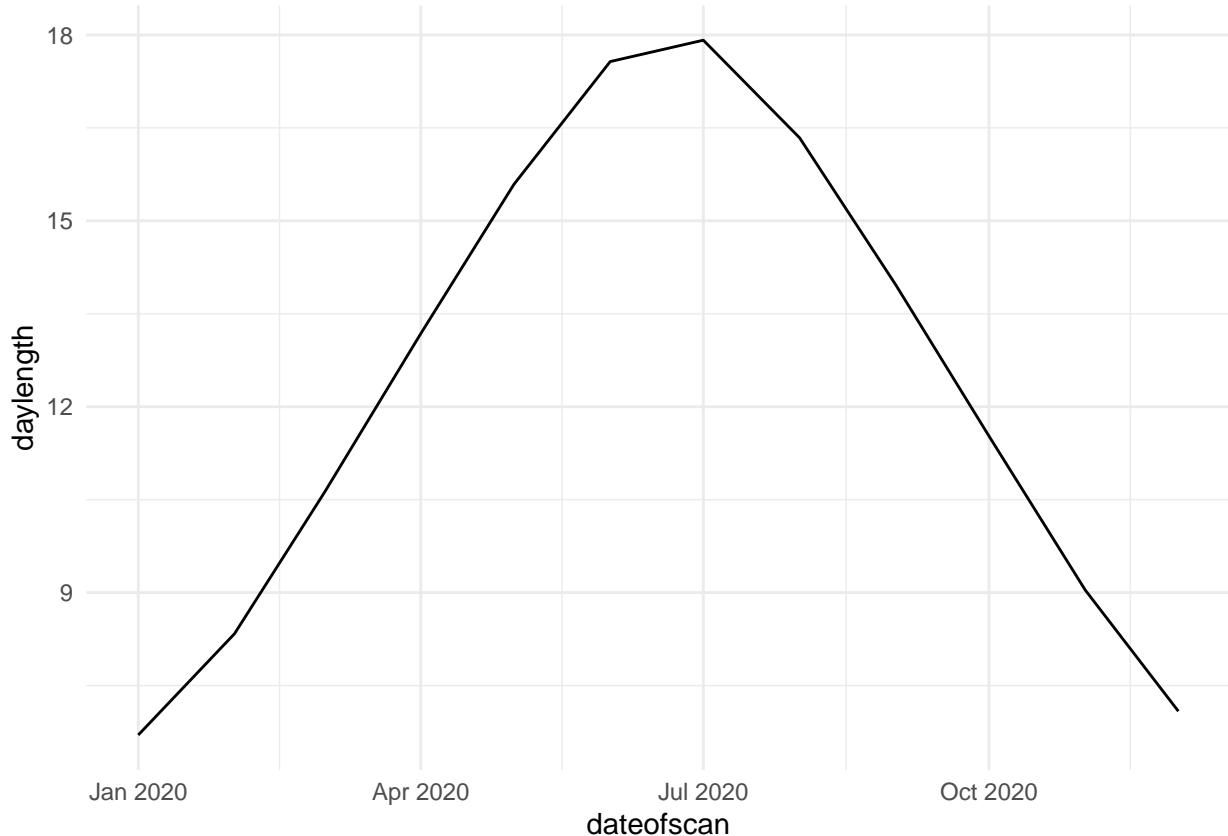
## Rows: 12 Columns: 3
## -- Column specification -----
## Delimiter: ","
## chr (1): date (dmy)
## dbl (2): ID's, latitude
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

dateslat <- dateslat %>%
  clean_names()

dateslat <- dateslat %>% mutate(dateofscan=(as.Date(date_dmy,format="%d/%m/%Y")))
dateslat <- dateslat %>% mutate(daylength=daylength(latitude,dateofscan))

dateslat %>% ggplot(aes(x=dateofscan,y=daylength)) +geom_line() +theme_minimal()
```

as.factor(gear)	mean	sd
3	17.52643	1.2327244
4	17.67000	1.1249296
5	14.86667	0.5507571



## 5 Statistical Analysis

### 5.1 Regression

#### 5.1.1 Linear regression on groups

### 5.2 Linear Regression

```
kable(mtcars %>% group_by(as.factor(gear)) %>%
  summarise(mean = mean(qsec), sd = sd(qsec))) %>%
  kable_styling(full_width = F) %>%
  kable_minimal()
```

```
#Run the same linear regression model by group levels?
#Instead of running #summary(lm(y~x)) for the number of levels
#you have, you can use the R package "broom" along with dplyr.
```

gear	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	no
3	0.58	0.54	29.74	16.40	0.00	1	-66.28	138.57	140.48	10616.60	12	
4	0.91	0.87	2.74	20.54	0.05	1	-8.32	22.64	20.80	14.99	2	
5	0.73	0.45	59.23	2.66	0.35	1	-14.85	35.71	33.00	3507.70	1	

```
# Run the same regression model for gears ##
kable(mtcars %>% group_by(gear) %>%
  do(fitgear = glance(lm(hp~qsec, data = .))) %>%
  unnest(fitgear), digits=2) %>% kable_styling(full_width = F) %>%
  kable_minimal()
```

Reference

```
fit <- lm(qsec ~ wt + hp+disp+factor(cyl), data = mtcars)
summary(fit)

##
## Call:
## lm(formula = qsec ~ wt + hp + disp + factor(cyl), data = mtcars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.1891 -0.4605 -0.1190  0.4892  1.5292 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16.861038  1.060566 15.898 3.18e-11 ***
## wt          1.200726  0.400340  2.999  0.00849 **  
## hp         -0.019943  0.004141 -4.817  0.00019 *** 
## disp        -0.001025  0.004350 -0.236  0.81664    
## factor(cyl)8 -0.356400  0.732189 -0.487  0.63303    
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7975 on 16 degrees of freedom
## Multiple R-squared:  0.7625, Adjusted R-squared:  0.7032 
## F-statistic: 12.84 on 4 and 16 DF,  p-value: 7.179e-05

effectsize(fit)

## # Standardization method: refit
##
## Parameter           | Std. Coef. |      95% CI
## -----
## (Intercept)          |     0.16 | [-0.59,  0.91]
## wt                  |     0.63 | [ 0.19,  1.08]
## hp                 -0.82 | [-1.18, -0.46]
## disp                -0.07 | [-0.71,  0.57]
## factor(cyl)0.690065559342355 | -0.24 | [-1.30,  0.82]
```

```

anova_table <- anova(fit)
effectsize(anova_table)

## # Effect Size for ANOVA (Type I)
##
## Parameter | Eta2 (partial) |      95% CI
## -----
## wt        |          0.15 | [0.00, 1.00]
## hp        |          0.75 | [0.53, 1.00]
## disp      |          0.03 | [0.00, 1.00]
## factor(cyl) |          0.01 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].

```

## 5.3 Logistic Regression

### 5.3.1 Create the LogR model

## 5.4 Principle Component Analysis

*complete dataset needed for following - impute values if required ##### make the PCA model*

```

data("mtcars")
mtcars <- mtcars %>% rownames_to_column("car_name") # if needed for df
mtcars <- rowid_to_column(mtcars, "row_num")
#make the PCA model
pcamodel.pca <- prcomp(mtcars[,c(3:13)], center = TRUE,scale. = TRUE) # I prefer naming columns, but have to use .scale = TRUE
#summarise PCA
summary(pcamodel.pca)

## Importance of components:
##              PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation   2.5707 1.6280 0.79196 0.51923 0.47271 0.46000 0.3678
## Proportion of Variance 0.6008 0.2409 0.05702 0.02451 0.02031 0.01924 0.0123
## Cumulative Proportion 0.6008 0.8417 0.89873 0.92324 0.94356 0.96279 0.9751
##              PC8     PC9     PC10    PC11
## Standard deviation   0.35057 0.2776 0.22811 0.1485
## Proportion of Variance 0.01117 0.0070 0.00473 0.0020
## Cumulative Proportion 0.98626 0.9933 0.99800 1.0000

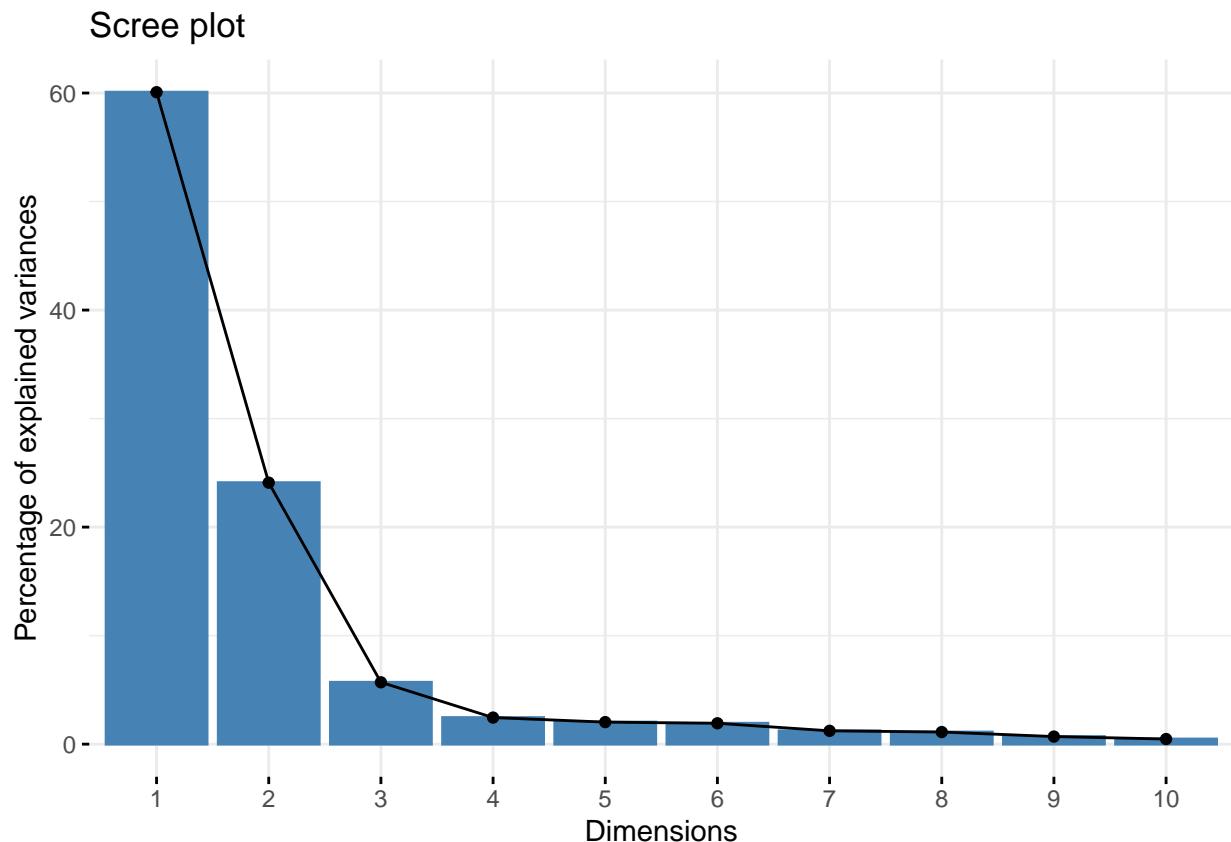
head(mtcars[,c(3:13)]) #check columns are corrrect

##      mpg cyl disp  hp drat    wt  qsec vs am gear carb
## 1 21.0   6 160 110 3.90 2.620 16.46  0  1    4    4
## 2 21.0   6 160 110 3.90 2.875 17.02  0  1    4    4
## 3 22.8   4 108  93 3.85 2.320 18.61  1  1    4    1
## 4 21.4   6 258 110 3.08 3.215 19.44  1  0    3    1
## 5 18.7   8 360 175 3.15 3.440 17.02  0  0    3    2
## 6 18.1   6 225 105 2.76 3.460 20.22  1  0    3    1

```

#### 5.4.1 make a scree plot of the PCA model

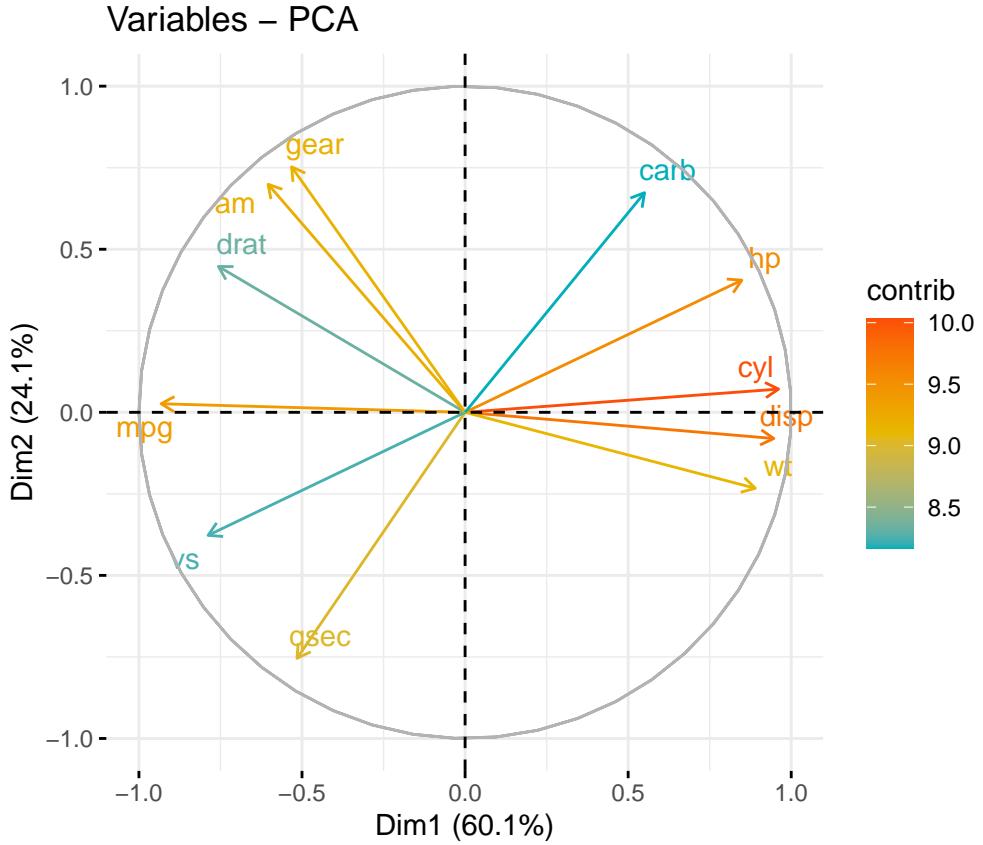
```
#creates scree plot  
fviz_eig(pcamodel.pca)
```



scree plot shows how much variance can be summarised with one variable (components)

#### 5.4.2 Graph of variables. Positive correlated variables point to the same side of the plot. Negative correlated variables point to opposite sides of the graph.

```
fviz_pca_var(pcamodel.pca,  
             col.var = "contrib", # Color by contributions to the PC  
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
             repel = TRUE )
```



```
# Extract the Principal components For each individual
```

this is the key stage of getting the first principal component out per rows score. essentially doing the principal component creates a method for then taking an rows scores, and giving them a value for '1 PC'. then reattach them to the original data frame spreadsheet.

```
extractedpcas <- predict(pcamodel.pca, newdata = mtcars)
head(extractedpcas)
```

```
##          PC1        PC2        PC3        PC4        PC5        PC6
## [1,] -0.64686274  1.7081142 -0.5917309  0.11370221  0.9455234 -0.01698737
## [2,] -0.61948315  1.5256219 -0.3763013  0.19912121  1.0166807 -0.24172464
## [3,] -2.73562427 -0.1441501 -0.2374391 -0.24521545 -0.3987623 -0.34876781
## [4,] -0.30686063 -2.3258038 -0.1336213 -0.50380035 -0.5492089  0.01929700
## [5,]  1.94339268 -0.7425211 -1.1165366  0.07446196 -0.2075157  0.14919276
## [6,] -0.05525342 -2.7421229  0.1612456 -0.97516743 -0.2116654 -0.24383585
##          PC7        PC8        PC9        PC10       PC11
## [1,] -0.42648652  0.009631217 -0.14642303  0.06670350  0.17969357
## [2,] -0.41620046  0.084520213 -0.07452829  0.12692766  0.08864426
## [3,] -0.60884146 -0.585255765  0.13122859 -0.04573787 -0.09463291
## [4,] -0.04036075  0.049583029 -0.22021812  0.06039981  0.14761127
## [5,]  0.38350816  0.160297757  0.02117623  0.05983003  0.14640690
## [6,] -0.29464160 -0.256612420  0.03222907  0.20165466  0.01954506
```

- these are the seven principal components of the first six people. we are only interested really in PC1.

```

# make the Principal component matrix into a data frame
extractedpcasdf <- as.data.frame(extractedpcas)
extractedpcasdf <- rowid_to_column(extractedpcasdf , "row_num")
#add the principal component values for the individuals to masterPCA file
mtcars <- left_join(mtcars,extractedpcasdf,by="row_num")

```

this was to add a row number that corresponds to the row number in the cognition data file above. \* doing this then allows us to add the principal component data to the cognition data and match the stradl IDs.

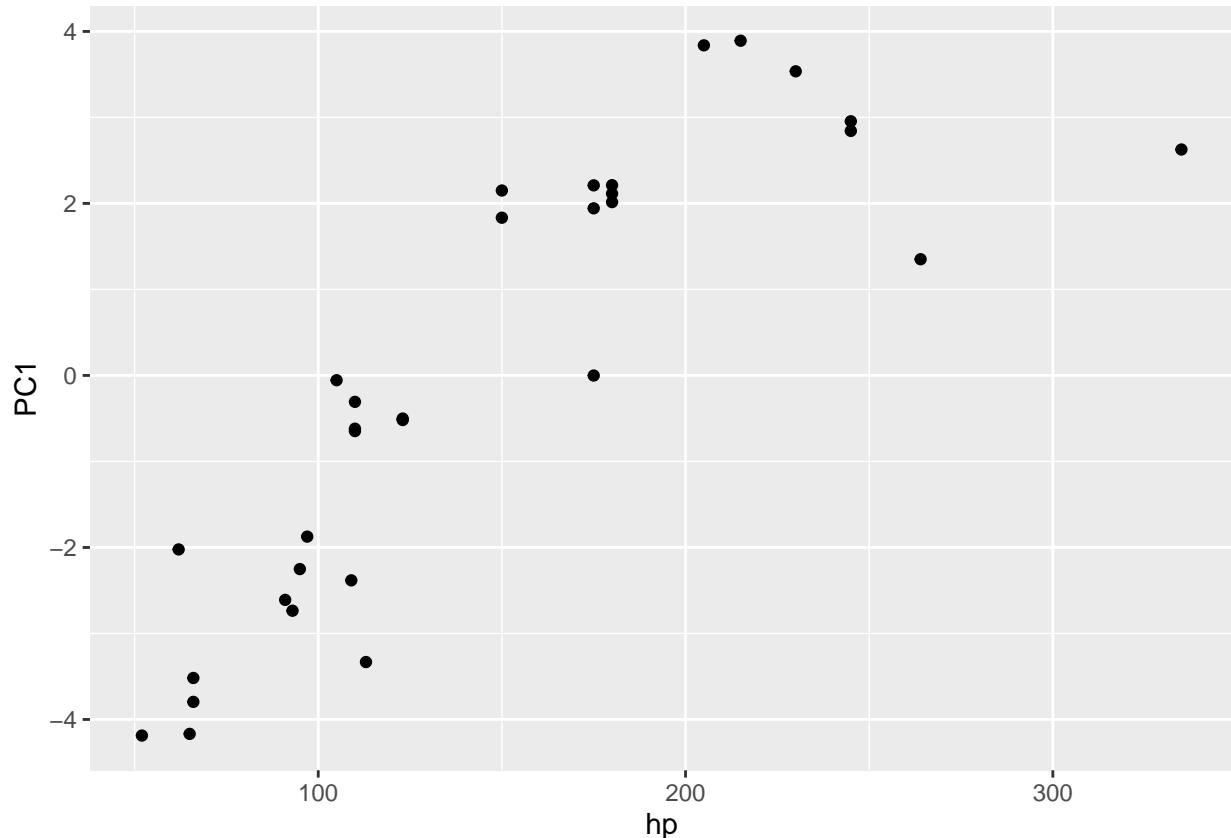
create a final spreadsheet with cog data and PCA

```

#delete row_num

mtcars<- mtcars%>% select(-c(row_num, PC3:PC11))
mtcars %>% ggplot(aes(x=hp,y=PC1))+geom_point()

```



## 5.5 Survival Analysis and Visualisation

### 5.5.1 To be completed

## 5.6 Receiver Operated Curves (ROC)

### 5.6.1 To be completed

## 5.7 Missing Values

```
library(naniar)
library(UpSetR)
#df_cog_w3w4$block3 <- with(df_cog_w3w4, impute(block3, mean))
#df_cog_w3w4$o3tco <- with(df_cog_w3w4, impute(o3tco, mean))
```

### 5.7.1 replaces all missing values with mean of column (across all variable columns)

```
# Assuming 'my_data' is your data frame with missing values
# Calculate column means
means <- colMeans(abc36, na.rm = TRUE)

# Replace missing values in each column with the corresponding column mean
#for (col in names(abc36)) {
#  abc36[is.na(abc36[, col]), col] <- means[col]
#}
```

## 5.8 Pivot data longer

```
mtcars_long <- mtcars %>% pivot_longer(PC1:PC2, names_to = 'PC', values_to = 'value')
```

## 5.9 Do analysis on multiple groups (e.g. dependent variable are many shape PCs) rstatix

```
stat.test <- mtcars_long %>%
  group_by(PC) %>%
  anova_test(value ~ hp+drat) %>%
  adjust_pvalue(method = "BH") %>%
  add_significance()
stat.test

## # A tibble: 4 x 10
##   PC    Effect  DFn    DFd      F     p `p<.05` ges   p.adj p.adj.signif
##   <chr> <chr>  <dbl> <dbl> <dbl> <dbl> <chr>  <dbl> <dbl> <chr>
## 1 PC1    hp      1     29  90.8 1.94e-10 *    0.758 7.76e-10 ****
## 2 PC1    drat    1     29  49.4 9.95e- 8 *    0.63   1.99e- 7 ****
## 3 PC2    hp      1     29  39.1 7.95e- 7 *    0.574 7.95e- 7 ****
## 4 PC2    drat    1     29  42.2 4.14e- 7 *    0.593 5.52e- 7 ****
```

cyl	Ave	StDev
4	82.63636	20.93453
6	122.28571	24.26049
8	209.21429	50.97689

## 6 Data Visualisation

Tables and graphs, survival plots, missing values.

### 6.1 Packages needed

```
library(arsenal) # for summary tables
library(summarytools) # for summary tables
library(gridExtra) # print multiple plots as grid
library(ggpmisc) # add formulas and p values to scatterplots
library(corrplot) #plotting correlations
library(Hmisc) #impute values
library(naniar) # deal with NAs
library(geosphere)
library(tidyverse) # data handling and viz
library(janitor) #dataframe import cleaning
library(knitr) #nice html tables
library(kableExtra) # nicer knitr tables
library(broom)
library(readr) # load csv stored data
library(geosphere) # for calc daylength
#library(RColorBrewer)
library(viridis)
library(reshape2)
library(ggrepel) # label points on ggplot
source("https://gist.github.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97121f7f9ce947837e")
```

### 6.2 Summary Tables

#### 6.2.1 Summarise by group

```
data(mtcars)
kable(mtcars %>% group_by(cyl) %>% summarise(Ave=mean(hp), StDev=sd(hp))) %>%
  kable_styling(full_width = FALSE) %>% kable_minimal()
```

#### 6.2.2 Multiple functions, variables

```
# make sure brackets are correct

df.sum <- mtcars %>% select(mpg,cyl,hp) %>%
  dplyr::summarise(across(everything(),list(mean=mean,sd=sd)))
```

mpg_mean	mpg_sd	cyl_mean	cyl_sd	hp_mean	hp_sd
20.09	6.03	6.19	1.79	146.69	68.56

Attribute	mean	sd
mpg	20.09	6.03
cyl	6.19	1.79
hp	146.69	68.56

```
kable(df.sum,digits=2) %>% kable_styling(full_width = FALSE) %>%
  kable_minimal() # perform the analysis
```

```
df.longer <- df.sum%>% pivot_longer(col=everything(),
names_to = c("Attribute",".value"),
names_sep = "_")
kable(df.longer,digits=2) %>%
  kable_styling(full_width = FALSE) %>%
  kable_minimal() # pivot longer the analysis to make it readable
```

### 6.2.3 ‘Arsenal’ summary table

```
tab1 <- tableby(cyl~gear+hp+wt,data=mtcars)
summary(tab1, text=TRUE, digits=2, digits.p=2, digits.pct=1)
```

	4 (N=11)	6 (N=7)	8 (N=14)	Total (N=32)	p value
gear					0.01
- Mean (SD)	4.09 (0.54)	3.86 (0.69)	3.29 (0.73)	3.69 (0.74)	
- Range	3.00 - 5.00	3.00 - 5.00	3.00 - 5.00	3.00 - 5.00	
hp					< 0.01
- Mean (SD)	82.64 (20.93)	122.29 (24.26)	209.21 (50.98)	146.69 (68.56)	
- Range	52.00 - 113.00	105.00 - 175.00	150.00 - 335.00	52.00 - 335.00	
wt					< 0.01
- Mean (SD)	2.29 (0.57)	3.12 (0.36)	4.00 (0.76)	3.22 (0.98)	
- Range	1.51 - 3.19	2.62 - 3.46	3.17 - 5.42	1.51 - 5.42	

### 6.2.4 Summarytools tables

```
descr(mtcars, stats = c("mean", "sd"), transpose = TRUE, headings = FALSE)
```

```
##
##          Mean   Std.Dev
## -----
##      am     0.41    0.50
##      carb    2.81    1.62
##      cyl     6.19    1.79
##      disp   230.72   123.94
```

	Mean	Std.Dev	N.Valid
am	0.406	0.499	32
carb	2.812	1.615	32
cyl	6.188	1.786	32
disp	230.722	123.939	32
drat	3.597	0.535	32
gear	3.688	0.738	32
hp	146.688	68.563	32
mpg	20.091	6.027	32
qsec	17.849	1.787	32
vs	0.438	0.504	32
wt	3.217	0.978	32

```
##      drat     3.60    0.53
##      gear     3.69    0.74
##      hp    146.69   68.56
##      mpg    20.09    6.03
##      qsec    17.85    1.79
##      vs      0.44    0.50
##      wt      3.22    0.98

kable(descr(mtcars, stats = c("mean", "sd", "n.valid")), transpose = TRUE,
      headings = FALSE), digits = 3) %>%
  kable_styling(full_width = FALSE)%>% kable_minimal()
```

### 6.2.5 Visual summary of data

Options are for markdown

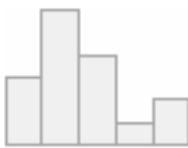
```
dfSummary(mtcars, plain.ascii = FALSE, style = "grid",
          graph.magnif = 0.5, valid.col = FALSE, tmp.img.dir = "/tmp")
```

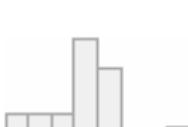
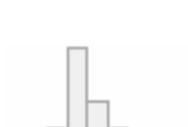
## temporary images written to 'C:\tmp'

### 6.2.6 Data Frame Summary

#### 6.2.6.1 mtcars Dimensions: 32 x 11

Duplicates: 0

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
1	mpg	Mean (sd) : 20.1 (6) [numeric] min < med < max: 10.4 < 19.2 < 33.9 IQR (CV) : 7.4 (0.3)	25 distinct values		0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
2	cyl [numeric]	Mean (sd) : 6.2 (1.8) min < med < max: $4 < 6 < 8$ IQR (CV) : 4 (0.3)	4 : 11 (34.4%) 6 : 7 (21.9%) 8 : 14 (43.8%)		0 (0.0%)
3	disp [numeric]	Mean (sd) : 230.7 (123.9) min < med < max: $71.1 < 196.3 < 472$ IQR (CV) : 205.2 (0.5)	27 distinct values		0 (0.0%)
4	hp [numeric]	Mean (sd) : 146.7 (68.6) min < med < max: $52 < 123 < 335$ IQR (CV) : 83.5 (0.5)	22 distinct values		0 (0.0%)
5	drat [numeric]	Mean (sd) : 3.6 (0.5) min < med < max: $2.8 < 3.7 < 4.9$ IQR (CV) : 0.8 (0.1)	22 distinct values		0 (0.0%)
6	wt [numeric]	Mean (sd) : 3.2 (1) min < med < max: $1.5 < 3.3 < 5.4$ IQR (CV) : 1 (0.3)	29 distinct values		0 (0.0%)
7	qsec [numeric]	Mean (sd) : 17.8 (1.8) min < med < max: $14.5 < 17.7 < 22.9$ IQR (CV) : 2 (0.1)	30 distinct values		0 (0.0%)
8	vs [numeric]	Min : 0 Mean : 0.4 Max : 1	0 : 18 (56.2%) 1 : 14 (43.8%)		0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
9	am	Min : 0 [numeric] Mean : 0.4 Max : 1	0 : 19 (59.4%) 1 : 13 (40.6%)		0 (0.0%)
10	gear	Mean (sd) : 3.7 (0.7) [numeric] min < med < max: 3 < 4 < 5 IQR (CV) : 1 (0.2)	3 : 15 (46.9%) 4 : 12 (37.5%) 5 : 5 (15.6%)		0 (0.0%)
11	carb	Mean (sd) : 2.8 (1.6) [numeric] min < med < max: 1 < 2 < 8 IQR (CV) : 2 (0.6)	1 : 7 (21.9%) 2 : 10 (31.2%) 3 : 3 ( 9.4%) 4 : 10 (31.2%) 6 : 1 ( 3.1%) 8 : 1 ( 3.1%)		0 (0.0%)

## 6.3 Correlation matrix

### 6.3.1 Ellipse style

```

corrdatal <- mtcars %>% select(-c(cyl,disp,vs,am,gear,carb))
corr1 <- Hmisc::rcorr(as.matrix(corrdatal))

corrp = cor.mtest(corrdatal, conf.level = 0.95)

P <- corrp$p
M <- corr1$r
#M
#colnames(M) <- c("mpg", "HP", "Axe Ratio", "Weight (kPounds)", "Quarter Mile (s)")
#rownames(M) <- c("mpg", "HP", "Axe Ratio", "Weight (kPounds)", "Quarter Mile (s)")
p_mat <- corrp$p
corr <- corrplot(M, type = "upper", method="ellipse", order = "hclust",
                  p.mat = p_mat, sig.level = 0.05, insig = "blank")

```

- Red is -ve correlation
- Blue is + ve correlation
- Blank is no correlation

Reference

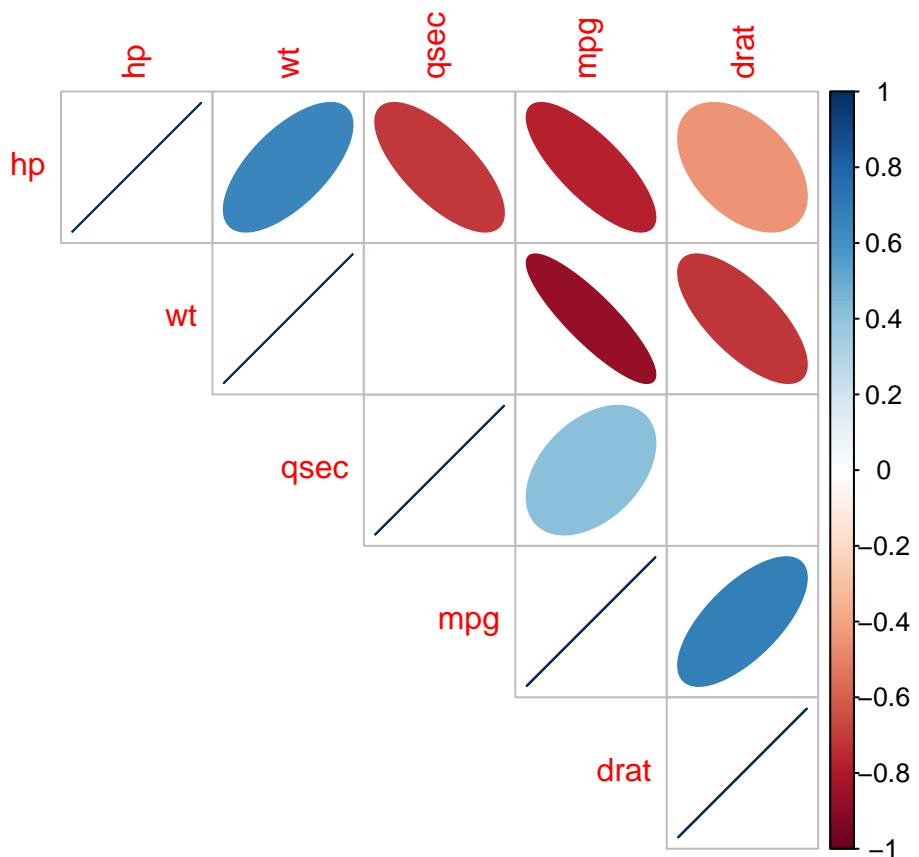
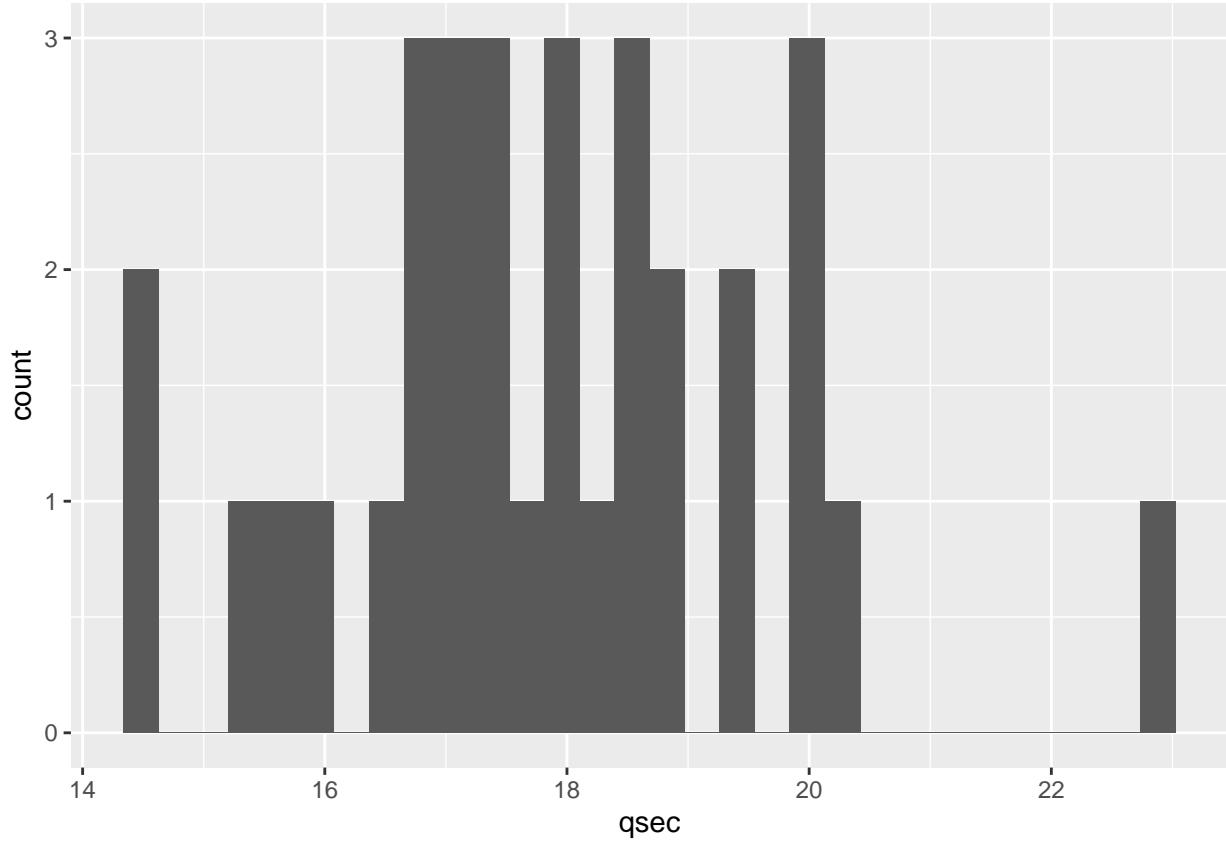


Figure 2: Correlation Plot

## 6.4 Graphing

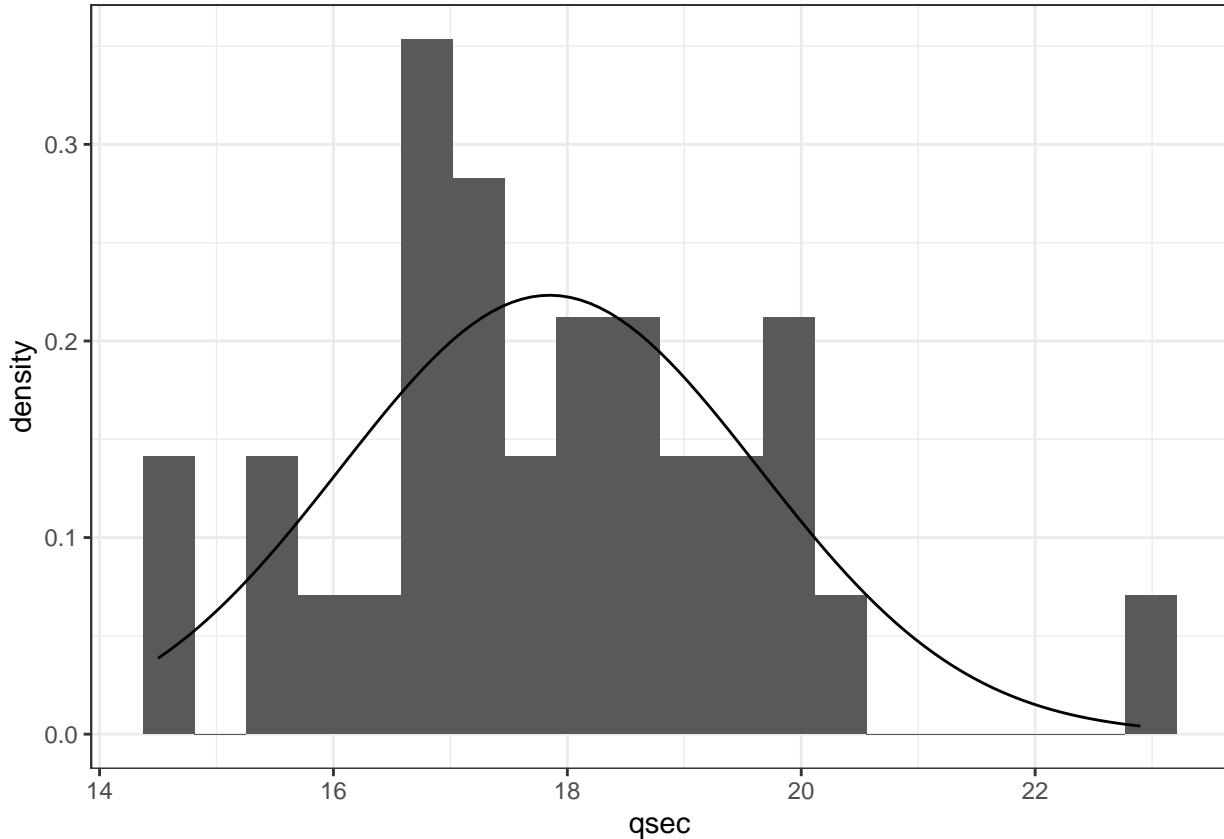
### 6.4.1 Basic distribution Histogram

```
plot1 <- mtcars %>% ggplot(aes(qsec)) + geom_histogram()  
plot1  
  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
#### Distribution + Normal line
```

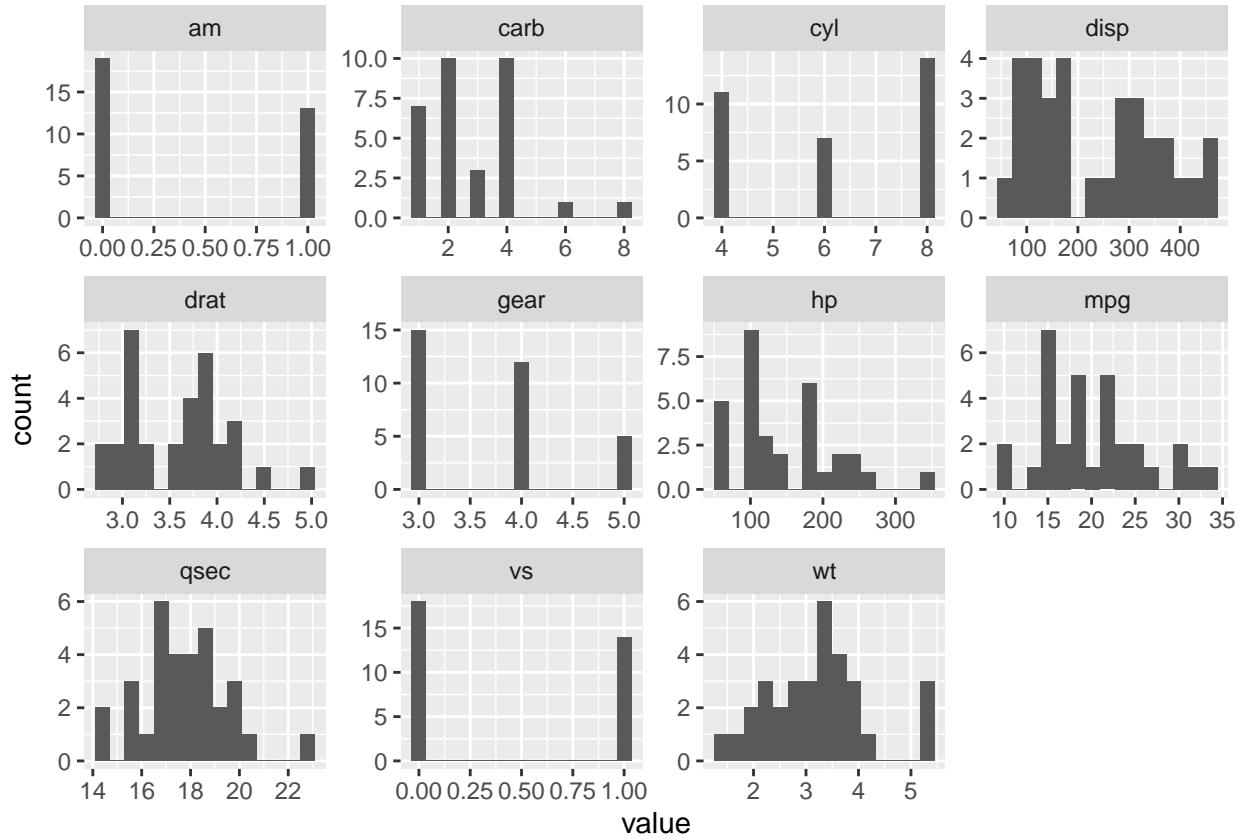
```
plot1 <- mtcars %>% ggplot(aes(qsec))  
# plot1+geom_histogram()  
# add normal plot  
plot1 + geom_histogram(aes( y=..density..),bins = 20)+  
  stat_function(fun = dnorm, args = list(mean =mean(mtcars$qsec), sd=sd(mtcars$qsec))) +  
  theme_bw()  
  
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.  
## i Please use `after_stat(density)` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was  
## generated.
```



```
# ..density.. changes y axis to density, not count. stat function defines normal
# line based on data provided.
```

#### 6.4.2 multiple plot of all distributions

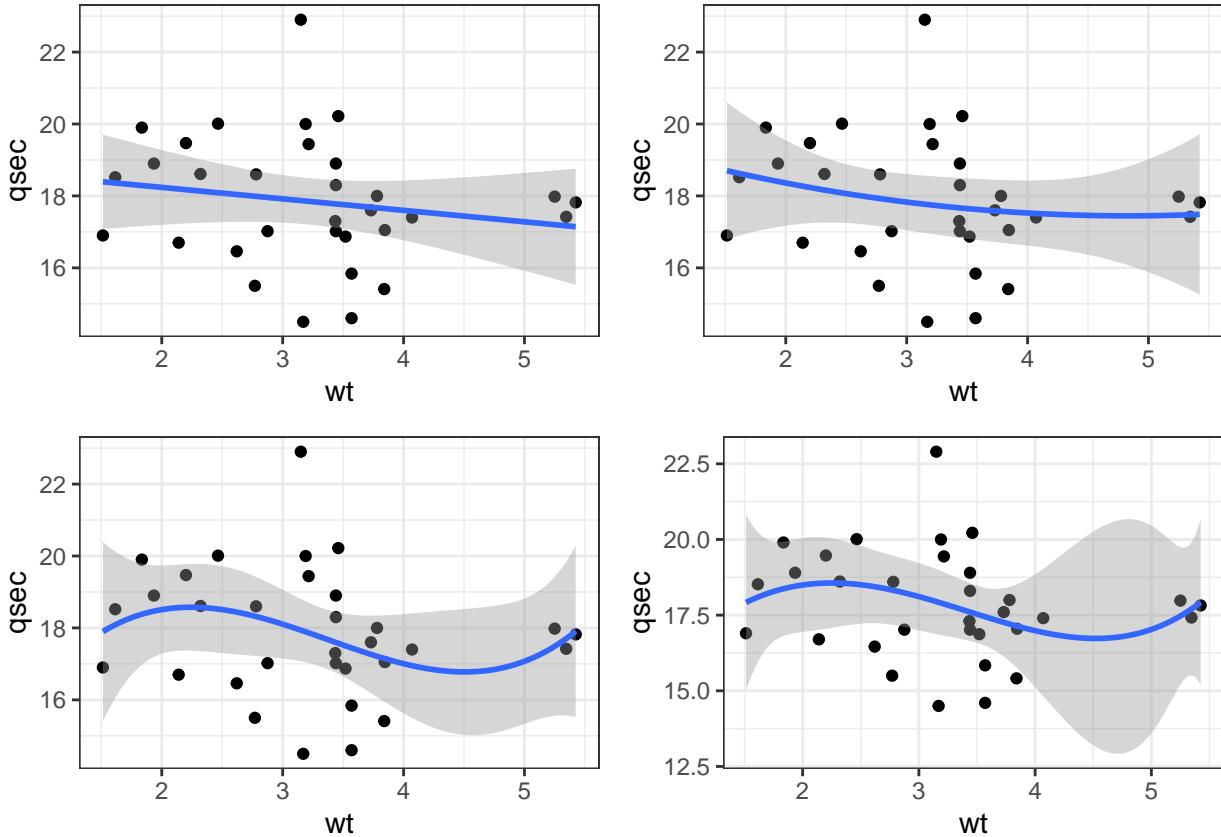
```
mtcars %>% keep(is.numeric) %>% gather() %>% ggplot(aes(value)) +
  facet_wrap(~ key, scales = "free") + geom_histogram(bins = 15)
```



#### 6.4.3 x\*y scatterplot with linear or polynomial regression

```
plot2 <- mtcars %>% ggplot(aes(x=wt,y=qsec))
plot2a <- plot2 +geom_point() +stat_smooth(method='lm',formula=y~x) + theme_bw()
plot2b <- plot2 +geom_point() +stat_smooth(method='lm',formula = y ~ poly(x, 2)) + theme_bw()
plot2c <- plot2 +geom_point() +stat_smooth(method='lm',formula = y ~ poly(x, 3)) + theme_bw()
plot2d <- plot2 +geom_point() +stat_smooth(method='lm',formula = y ~ poly(x, 4)) + theme_bw()

grid.arrange(plot2a,plot2b,plot2c,plot2d,nrow=2,ncol=2)
```



#### 6.4.4 Add formula to plot.

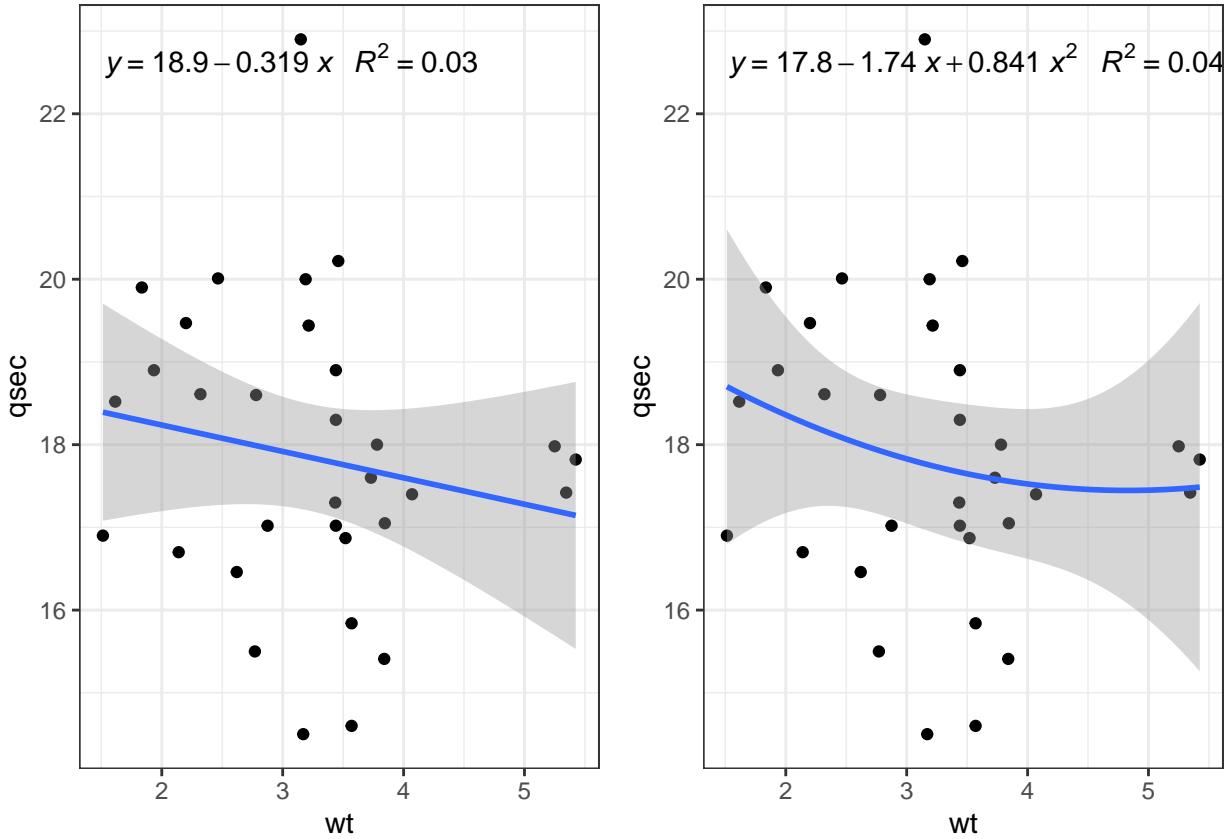
```

my.formula <- y ~ x
a <- plot2 +geom_point() +geom_smooth(method='lm', formula=my.formula) +
  stat_poly_eq(formula = my.formula, aes(label = paste(..eq.label.., ..rr.label..,
                                                sep = "~~~")), parse = TRUE) +
  theme_bw()

my.formula2 <- y ~ poly(x, 2)
b <- plot2 +geom_point() +geom_smooth(method='lm', formula=my.formula2) +
  stat_poly_eq(formula = my.formula2, aes(label = paste(..eq.label.., ..rr.label..,
                                                sep = "~~~")), parse = TRUE) +
  theme_bw()

grid.arrange(a,b,nrow=1)

```



#### 6.4.5 Raincloud plots (ggplot)

```
library(plyr)

## -----
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## -----
## 
## Attaching package: 'plyr'

## The following objects are masked from 'package:rstatix':
## 
##     desc, mutate

## The following objects are masked from 'package:Hmisc':
## 
##     is.discrete, summarize
```

```

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:purrr':
##
##     compact

library(dplyr)

```

```

theme_rain = theme(
text = element_text(size = 10),
axis.title.x = element_text(size = 16),
axis.title.y = element_text(size = 16),
axis.text = element_text(size = 14),
axis.text.x = element_text(angle = 0, vjust = 0.5),
legend.title=element_text(size=16),
legend.text=element_text(size=16),
legend.position = "right",
plot.title = element_text(lineheight=.8, face="bold", size = 16),
panel.border = element_blank(),
panel.grid.minor = element_blank(),
panel.grid.major = element_blank(),
axis.line.x = element_line(colour = 'black', size=0.5, linetype='solid'),
axis.line.y = element_line(colour = 'black', size=0.5, linetype='solid'))

```

#### 6.4.5.1 custom theme creation

```

## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

```

lb <- function(x) mean(x) - sd(x)
ub <- function(x) mean(x) + sd(x)

```

#### 6.4.5.2 make summary functions

```

mtcars <- tibble::rownames_to_column(mtcars, "car_name")
mtcars <- mtcars %>% mutate(cyl=as_factor(cyl))

```

#### 6.4.5.3 row names as real column

cut	mean	median	lower	upper
Fair	1.0461366	1.00	0.5297323	1.562541
Good	0.8491847	0.82	0.3951303	1.303239
Very Good	0.8063814	0.71	0.3469460	1.265817
Premium	0.8919549	0.86	0.3766933	1.407217
Ideal	0.7028370	0.54	0.2699607	1.135713

```
data("diamonds")
sumld<- ddply(diamonds, ~cut, summarise, mean = mean(carat), median = median(carat),
               lower = lb(carat), upper = ub(carat))
kable(head(sumld)) %>% kable_minimal()
```

#### 6.4.5.4 calc summary data

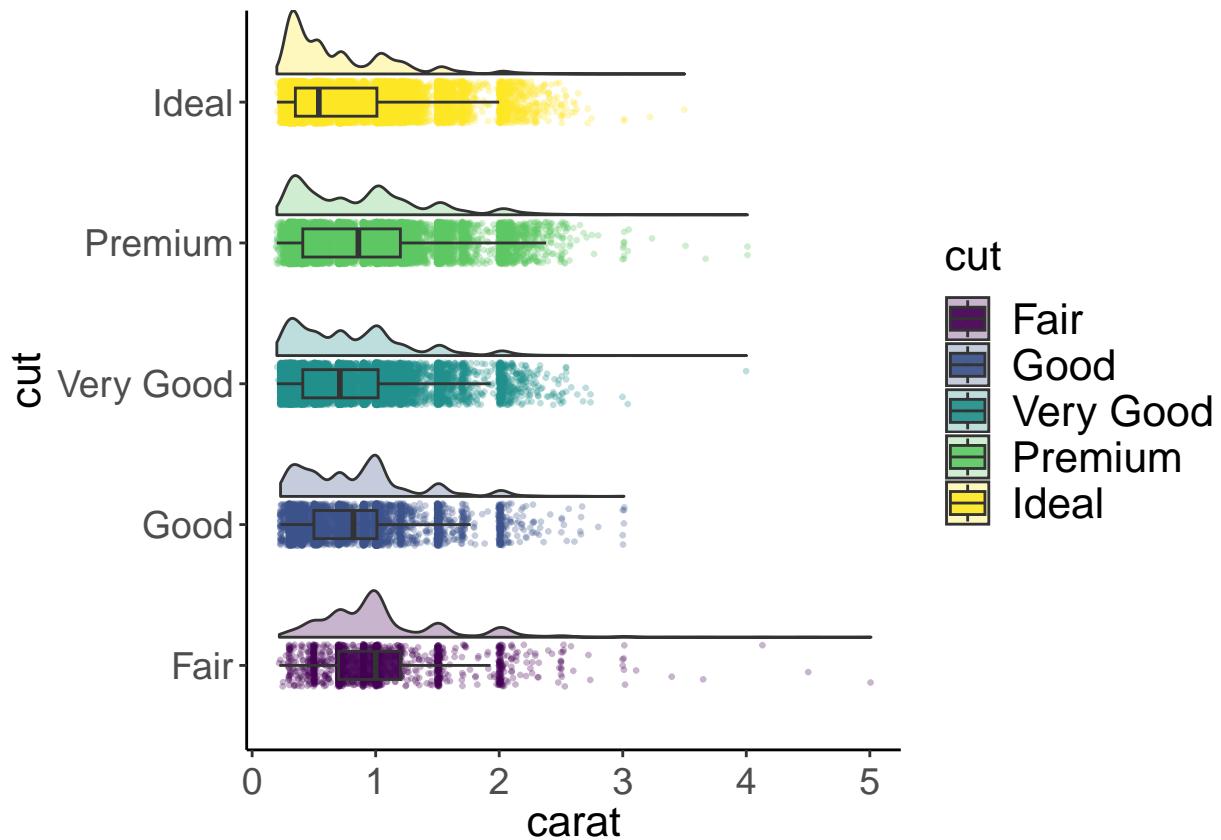
#### 6.4.6 raincloud plot (diamonds)

```
g <- ggplot(data = diamonds, aes(y = carat, x = cut, fill = cut)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .3) +
  geom_point(aes(y = carat, color = cut), position = position_jitter(width = .15), size = .5, alpha = 0.3) +
  geom_boxplot(width = .2, guides = FALSE, outlier.shape = NA, alpha = 0.9) +
  expand_limits(x = 5.25) +
  scale_color_viridis_d() +
  scale_fill_viridis_d() +
  coord_flip() +
  theme_bw() +
  theme_rain

## Warning in geom_boxplot(width = 0.2, guides = FALSE, outlier.shape = NA, :
## Ignoring unknown parameters: `guides`  

g

## Warning: Using the `size` aesthetic with geom_polygon was deprecated in ggplot2 3.4.0.
## i Please use the `linewidth` aesthetic instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



### Alternative raincloud

```
#calculations needed
sumld<- ddply(diamonds, ~cut, summarise, mean = mean(carat), median = median(carat), lower = lb(carat), upper = ub(carat))

g <- ggplot(data = diamonds, aes(y = carat, x = cut, fill = cut)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8) +
  geom_point(aes(y = carat, color = cut), position = position_jitter(width = .15), size = .5, alpha = 0.8) +
  geom_point(data = sumld, aes(x = cut, y = mean), position = position_nudge(x = 0.3), size = 2.5) +
  geom_errorbar(data = sumld, aes(ymin = lower, ymax = upper, y = mean), position = position_nudge(x = 0.3), width = 0.2) +
  expand_limits(x = 5.25) +
  guides(fill = FALSE) +
  guides(color = FALSE) +
  scale_color_viridis_d() +
  scale_fill_viridis_d() +
  theme_bw() +
  theme_rain

## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

g
```

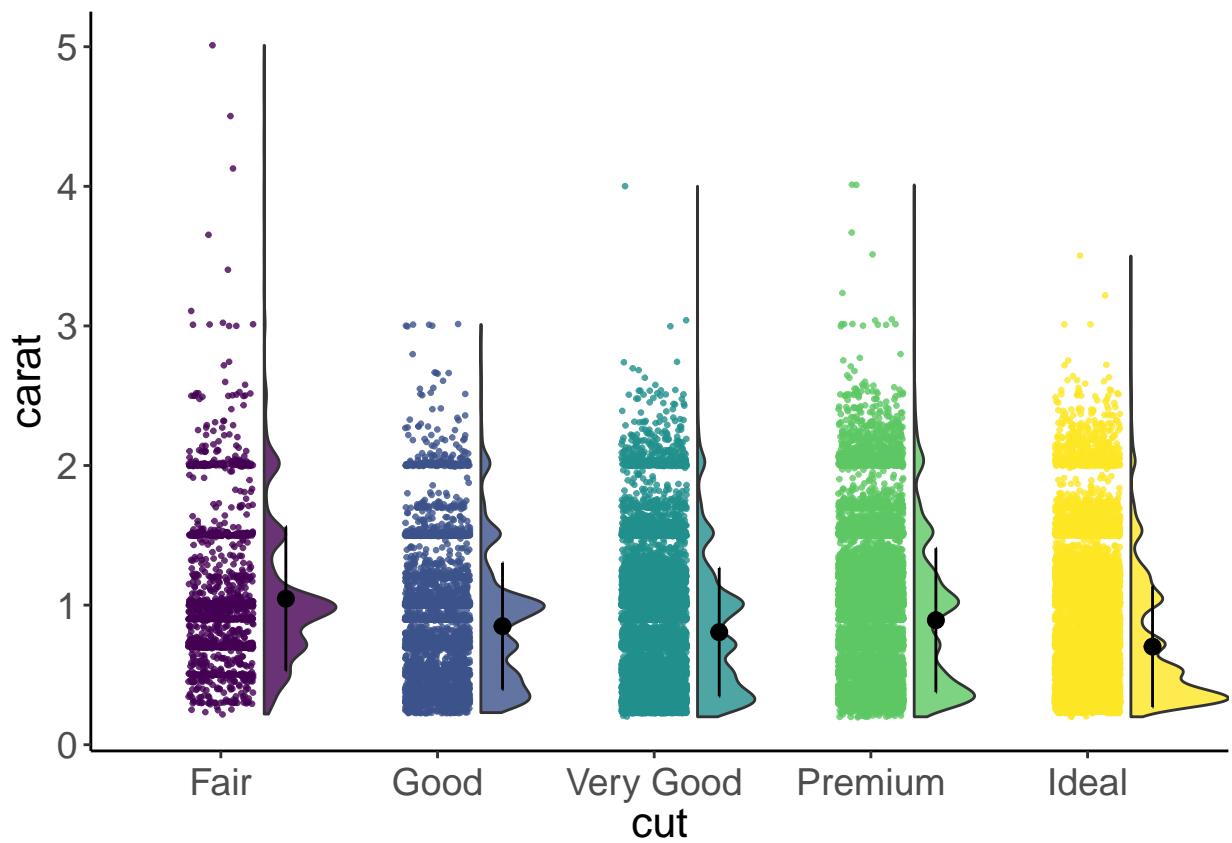


Figure 3: Raincoud plot of means

#### 6.4.7 Scatterplot theme

```
theme_scatter = theme(
text = element_text(size = 10),
axis.title.x = element_text(size = 12),
axis.title.y = element_text(size = 12, angle = 0, vjust = .5),
axis.text = element_text(size = 10),
axis.text.x = element_text(angle = 0, vjust = 0.5),
legend.title=element_text(size=12,hjust = .5),
legend.text=element_text(size=10),
#legend.position = "right",
legend.background = element_rect(colour='light grey'),
plot.title = element_text(lineheight=.8, face="bold", size = 16),
panel.border = element_blank(),
panel.grid.minor = element_blank(),
panel.grid.major = element_blank(),
axis.line.x = element_line(colour = 'black', size=0.5, linetype='solid'),
axis.line.y = element_line(colour = 'black', size=0.5, linetype='solid'))
```

#### 6.4.8 Scatterplots

```
sp <- diamonds %>% ggplot(aes(x=carat,y=price))
sp1 <- sp+geom_point()
sp2 <- sp+geom_point() +theme_bw()
sp3 <- sp+geom_point() +theme_bw() +theme_scatter
sp4 <- sp+geom_point(alpha=.01)+ylab('£') +theme_bw() +theme_scatter

grid.arrange(sp1,sp2,sp3,sp4,nrow=2,ncol=2)
```

#### 6.4.9 make axis logarithmic

```
sp5 <- sp+geom_point(alpha=.01)+ylab('£') +theme_bw() +theme_scatter
sp5+ scale_x_continuous(trans='log10') +
  scale_y_continuous(trans='log10')
```

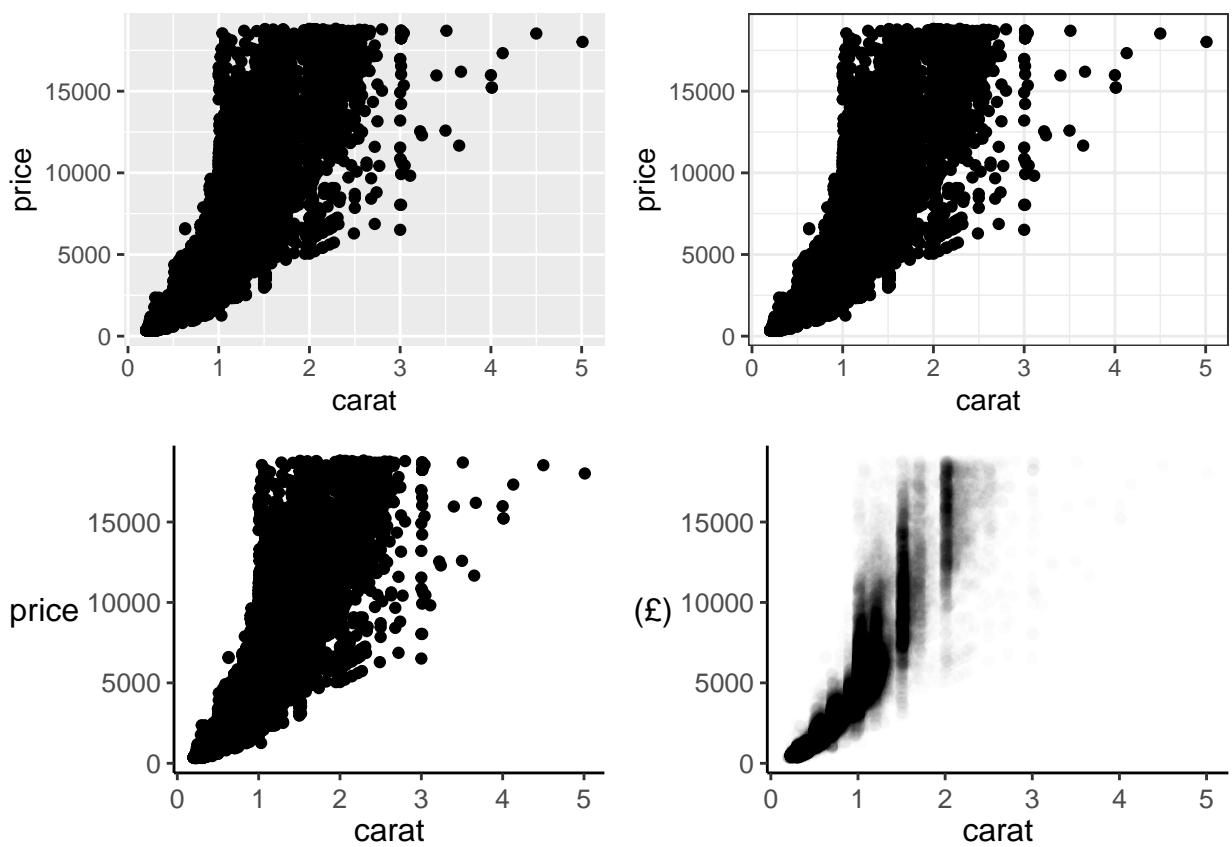
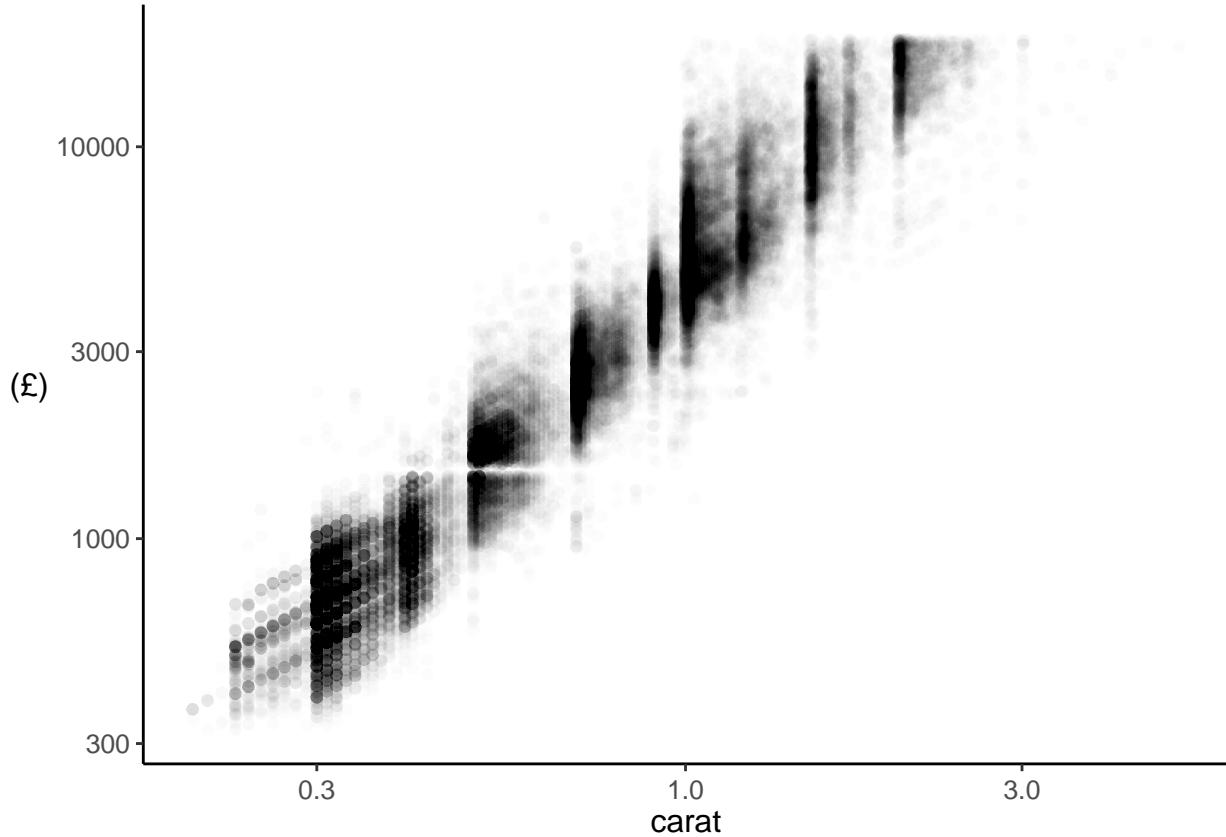


Figure 4: Scatterplots



#### 6.4.10 add a trendline

```

my.formula <- y ~ x # calc formula for display

sp5+ylim(0,20000)+xlim(0,3)+geom_smooth(method='lm',formula =my.formula,
                                             colour='black', size=.4,alpha=.6)+

  stat_poly_eq(formula = my.formula,
               aes(label = paste(..eq.label.., ..rr.label..,
                                 sep = "~~~")), parse = TRUE)

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning: Removed 32 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 32 rows containing non-finite values (`stat_poly_eq()`).

## Warning: Removed 32 rows containing missing values (`geom_point()`).

## Warning: Removed 8 rows containing missing values (`geom_smooth()`).

```

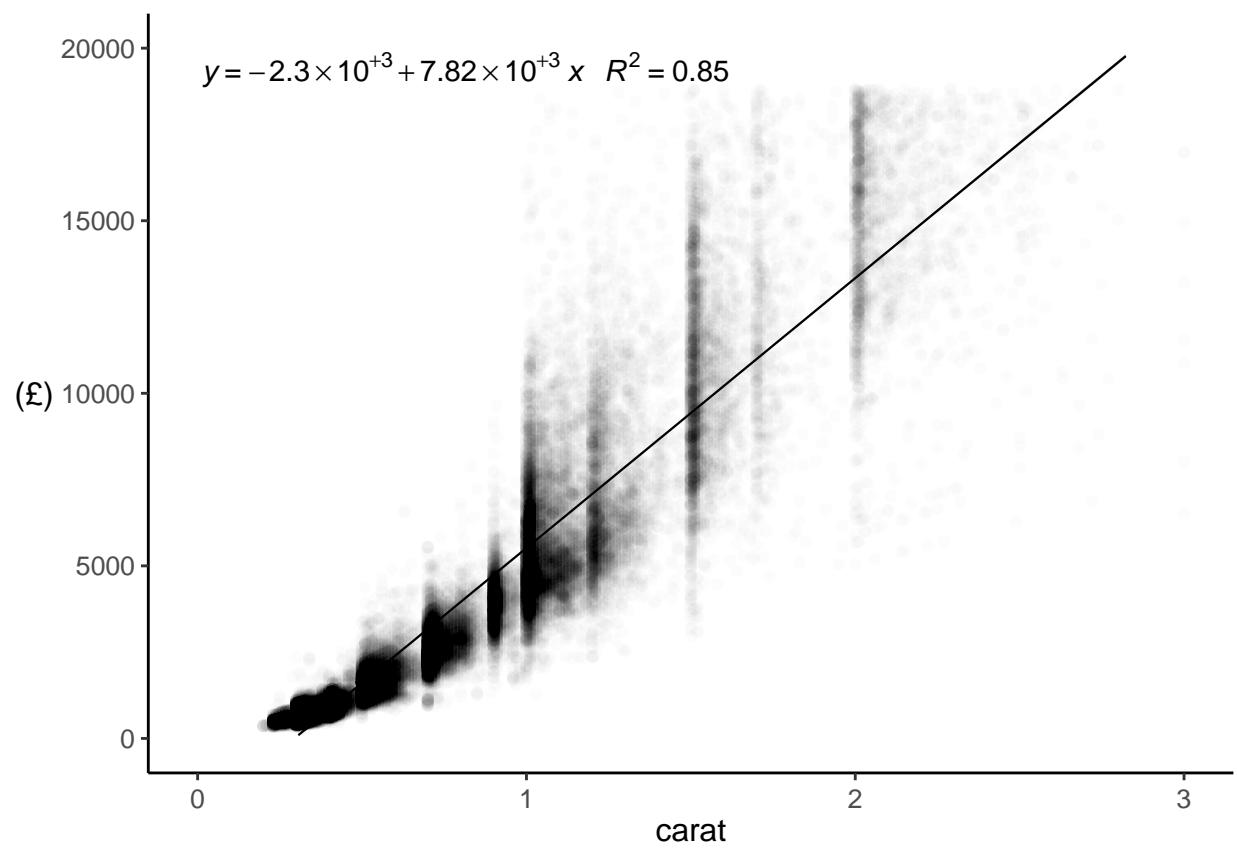


Figure 5: linear Trendline

#### 6.4.11 add a trendline

```
formula <- y ~ poly(x, 2, raw=TRUE) # calc formula for display

sp5+ylim(0,20000)+xlim(0,3)+geom_smooth(method='lm', formula = formula,
                                         colour='black', size=.4, alpha=.6)+
  stat_poly_eq(formula = formula,
               aes(label = paste(..eq.label.., ..rr.label..,
                                 sep = "~~~")), parse = TRUE)

## Warning: Removed 32 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 32 rows containing non-finite values (`stat_poly_eq()`).
## Warning: Removed 32 rows containing missing values (`geom_point()`).
## Warning: Removed 14 rows containing missing values (`geom_smooth()`).
```

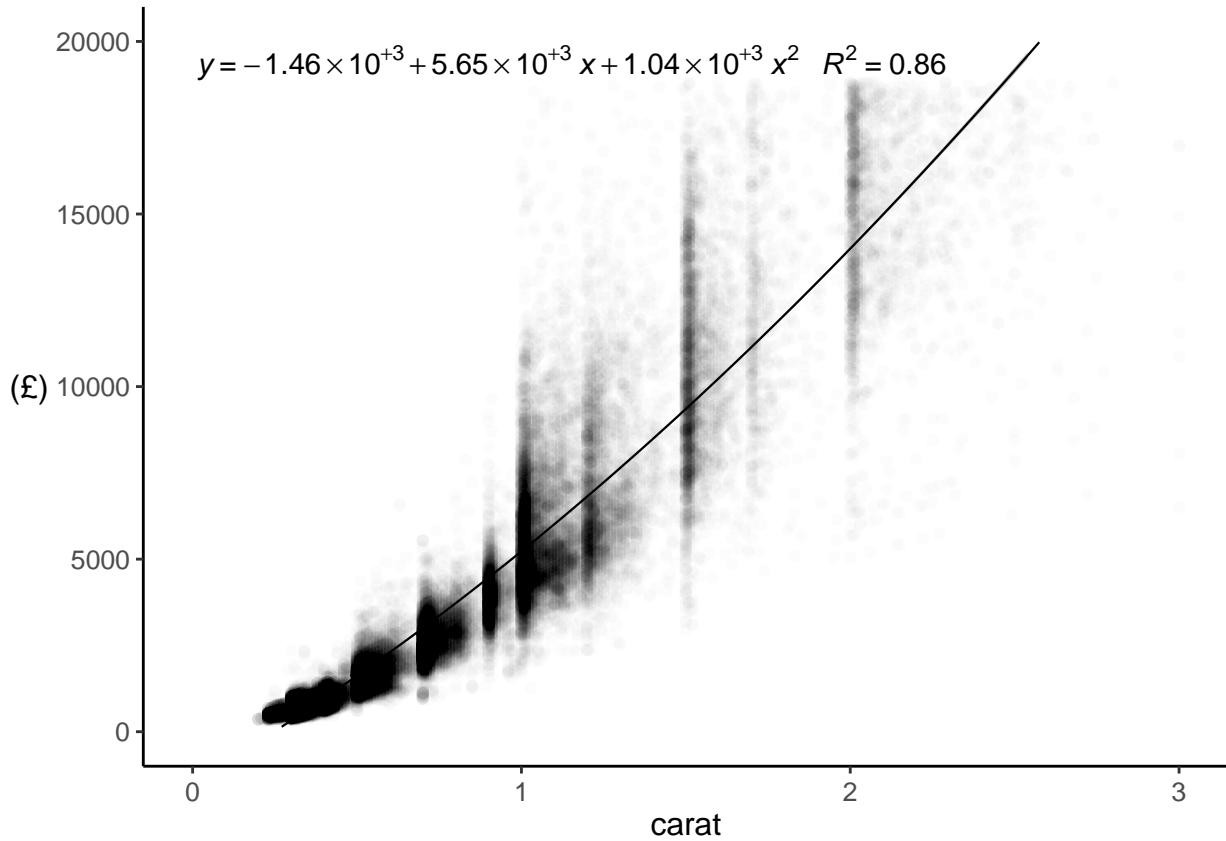


Figure 6: Polynomial trendline

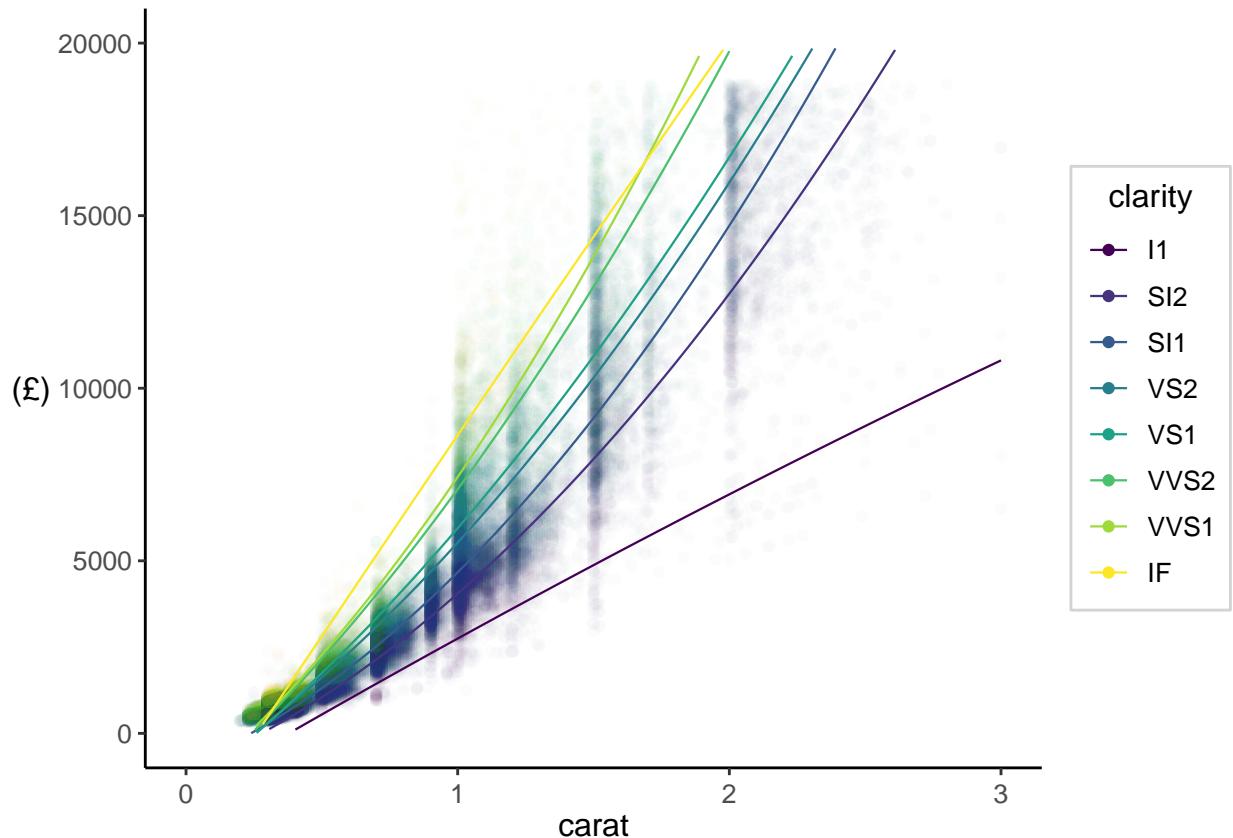
#### 6.4.12 add multiple trendlines

```

sp <- diamonds %>% ggplot(aes(x=carat,y=price,colour=clarity))
sp5 <- sp+geom_point(alpha=.01)+ylab(' (£)') +theme_bw() +theme_scatter
sp6 <- sp5+ylim(0,20000)+xlim(0,3)+ guides(colour = guide_legend(override.aes = list(alpha = 1)))
my.formula4 <- y ~ poly(x, 2,raw = TRUE) # calc formula for display

sp7 <- sp6+geom_smooth(aes(colour=clarity),method='lm',formula = my.formula4,se=F, size=.4,alpha=.6)
sp7

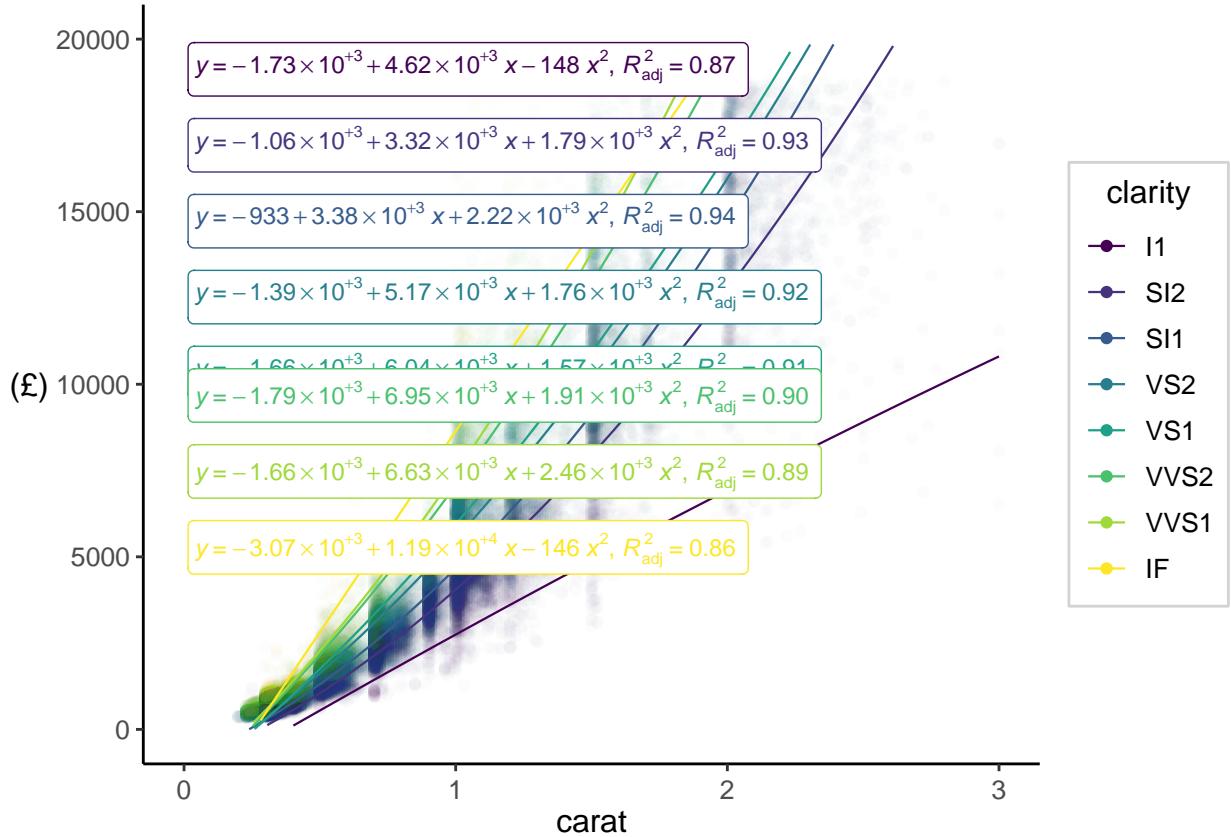
```



```

sp8 <- sp7+
  stat_poly_eq(aes(label =  paste(stat(eq.label),
                                stat(adj.rr.label), sep = "*\\", "\\*")),
                formula = my.formula4, parse = TRUE, size=3, geom = "label_npc")
sp8

```



```
#### add deviation from regression
```

```
data(mtcars)
mtcars <- tibble::rownames_to_column(mtcars, "car_name")

formula <- y ~ poly(x, 2, raw=TRUE) # calc formula for display

hpvmpg <- mtcars %>% ggplot(aes(x=hp, y=mpg, label=rownames(mtcars)))
# p1 <- hpvmpg+geom_point()+geom_smooth(method='lm', formula =formula, colour='black', size=.4, alpha=.6, se=TRUE)
# stat_fit_deviations(formula = formula, colour = "red") + geom_label_repel(aes(label =rownames(mtcars)),
#   box.padding = 0.1,
#   point.padding = 0.3,
#   segment.color = 'grey50')

p2 <- hpvmpg+geom_point()+geom_smooth(method='lm', formula =formula, colour='black', size=.4, alpha=.6, se=TRUE)
  stat_fit_deviations(formula = formula, colour = "red") + geom_label_repel(aes(label =rownames(mtcars)),
    arrow = arrow(length = unit(0.02, "npc")),
    box.padding = .5, min.segment.length = 0, max.overlaps = Inf)
```

```
p2
```

```
## Warning: The following aesthetics were dropped during statistical transformation: label
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```

## The following aesthetics were dropped during statistical transformation: label
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?

```

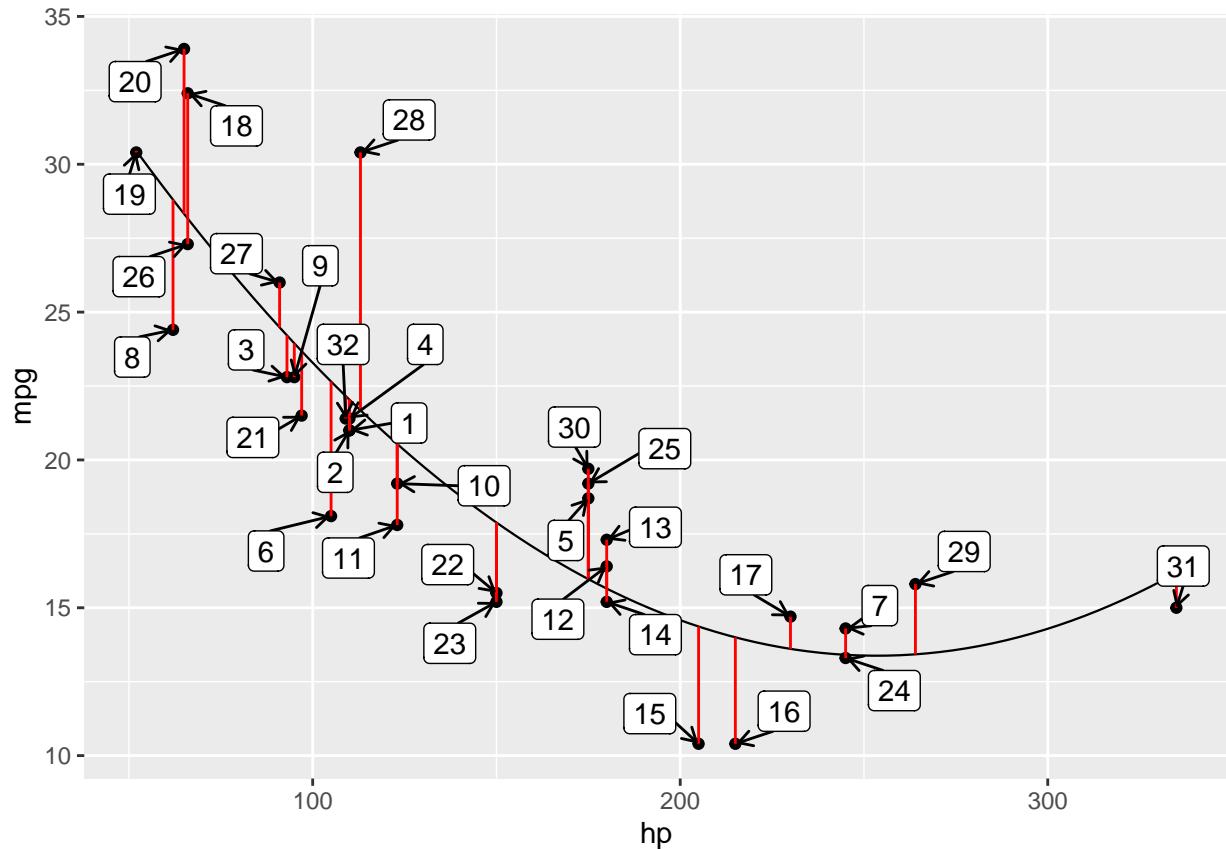


Figure 7: Deviation from prediction

```
#grid.arrange(p1,p2)
```

#### 6.4.13 residuals (ID those >1SD from \$\bar{X}\$)

```

lm <- lm(hp ~ poly(disp, 2, raw=TRUE), data=mtcars) # make lin model
resids <- resid(lm) # extract resids as vector
mtcars <- mtcars %>% mutate(lmresids=resids) # add to df

low <- mtcars %>% summarise(low=mean(lmresids)-sd(lmresids))#calc low limit
# assign as variable
high <- mtcars %>% summarise(high=mean(lmresids)+sd(lmresids))
low <- dplyr::pull(low)
high <- dplyr::pull(high)
mtcars <- mtcars %>% mutate(sds=ifelse(lmresids>low & lmresids<high, 0, 1)) #create new var

```

car_name	mpg	cyl	disp	hp	lmresids	sds
Mazda RX4	21.0	6	160	110	-11.763042	0
Mazda RX4 Wag	21.0	6	160	110	-11.763042	0
Datsun 710	22.8	4	108	93	5.792055	0
Hornet 4 Drive	21.4	6	258	110	-63.114644	1
Hornet Sportabout	18.7	8	360	175	-32.450810	0
Valiant	18.1	6	225	105	-52.832288	1

```
kable(head(mtcars[-c(6:12)]))%>%
  kable_styling(full_width = FALSE) %>% kable_minimal()
```

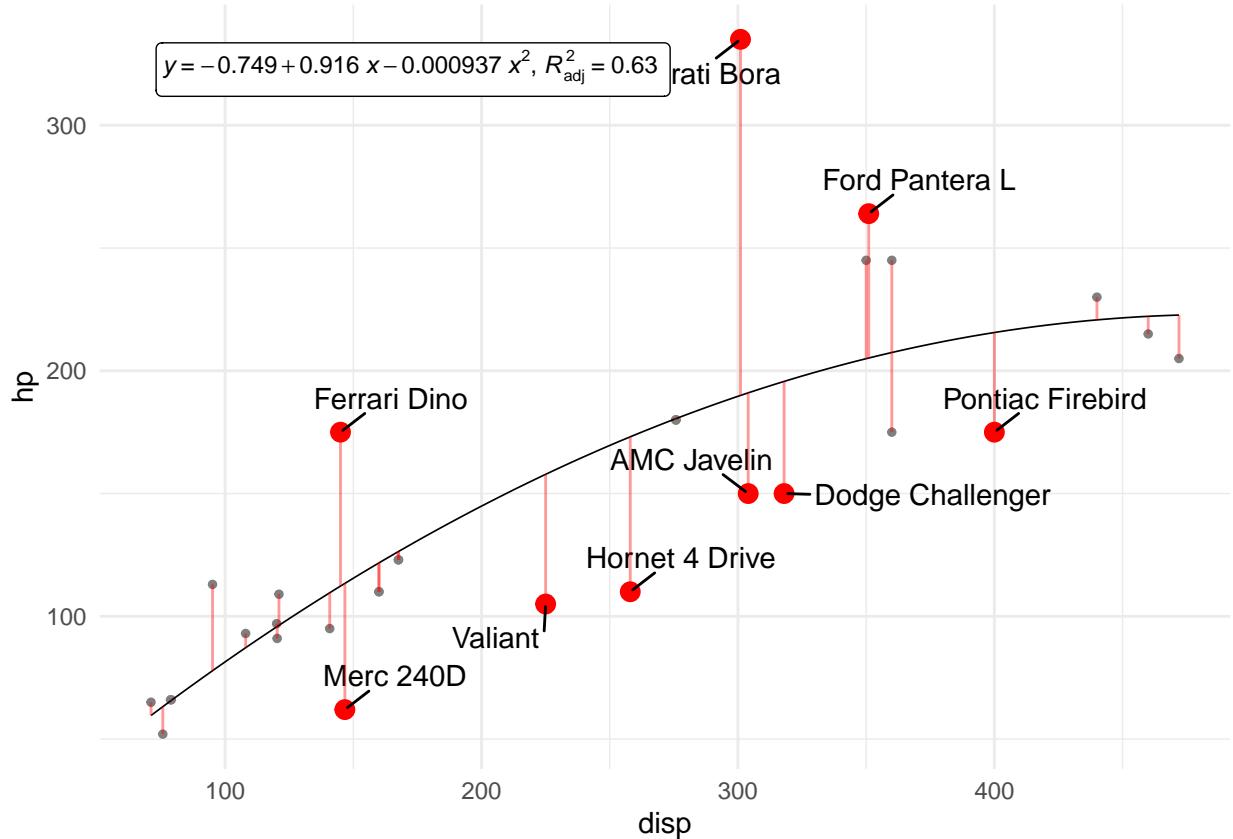
```
p2 <- hpvmpg+geom_point()+geom_smooth(method='lm',formula = formula,colour='black', size=.4,alpha=.6,se=F)+stat_fit_deviations(formula = formula, colour = "red")+geom_label_repel(aes(label = car_name),arrow = arrow(length = unit(0.02, "npc")), box.padding = .5,min.segment.length = 0,max.overlaps = Inf)
```

#### 6.4.14 only label extreme residuals

```
formula <- y ~ poly(x, 2,raw=TRUE) # calc formula for display
dat2 <- mtcars
dat2$car_name <- ""
ix_label <- which(mtcars$sds == 1)
dat2$car_name[ix_label] <- mtcars$car_name[ix_label]
hpvmpg <- dat2 %>% ggplot(aes(x=disp,y=hp,label=car_name))

hpvmpg+geom_point(color = ifelse(dat2$car_name == "", "grey50", "red"),size = ifelse(dat2$car_name == "", 1, 4),shape=19)+stat_poly_eq(aes(label = paste(stat(eq.label),
                                             stat(adj.rr.label), sep = "*\\", "\\*")))+stat_fit_deviations(formula = formula, colour = "red",size=.5,alpha=.4)+stat_label_npc(aes(label = car_name))
formula = formula, parse = TRUE,size=3, geom = "label_npc")+theme_minimal()

## Warning: The following aesthetics were dropped during statistical transformation: label
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: label
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



## 6.5 line defined by equation to scatterplot

```
plot2<- mtcars %>% ggplot(aes(x=hp,y=qsec))
q1 <- plot2+geom_point()+
  stat_function(fun = function(x) 20-(.013*x)) # linear function
fun = 'y = 20 - 0.013x - 0.00003x^2'
q2 <- plot2+geom_point()+
  stat_function(fun = function(x) 20-(.013*x+.00003*x^2))# poly function
# poly function
q2 <- q2+ annotate("text", x = 175, y = 22, label = fun,size=4)
grid.arrange(q1,q2,nrow=1)
```

