Contents

#— #title: “Useful R syntax” #author: “Dr Chris McNeil” #site: bookdown::bookdown_site #documentclass: book #output: # bookdown::gitbook: default # bookdown::pdf_book: default #—

1 Introduction

This document is a collection of useful code for Rmarkdown 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 packaged
- 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 it 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
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     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
```

```r
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

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

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

```r
detach(package:clipr) #unload
```

3.3 Import using Janitor

```r
# 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
```

```r
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

```r
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 concatination

```
# 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: 0x00000180a6b02150>
## <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(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(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> <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 specfic 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 replace NA with a specified value across specified columns

```
#df_dement <- df_dement %>%
  #mutate(across(3:8, ~ifelse(is.na(.), 0, .)))
```

4.9 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
```

4.10 Change specific datapoint

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

4.11 Work with dates

4.11.1 Add a date column

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

4.12 Extract the last x num of a string

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

```
mtcars <- mtcars %>%rownames_to_column("car_name")

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

4.13 Add an order (e.g. order of birth)

Calculate where in order of siblings particpant was born

create dedicated df for dob calcs

add an order no

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

4.15 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()
```

4.16 Delete duplicate merged columns

4.17 Impute missing values

4.17.1 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:

4.18 check for any na's in a df

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

```
## integer(0)
```

4.19 Keep rows based on a unique value.

e.g. prescription code

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

Reference

4.20 Delete rows on a variable value

```r
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
```

4.21 Use if else to calculate on values

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

4.22 Find the rowwise maximum and make a new variable

4.23 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.

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

```
# 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(valves=cyl*4)
mtcars_extradata <- mtcars_extradata %>% select(-cyl)

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
```

4.24 Merge data frames (individuals)

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

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

Reference

4.25 Create a new factor from existing

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

4.26 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))
```

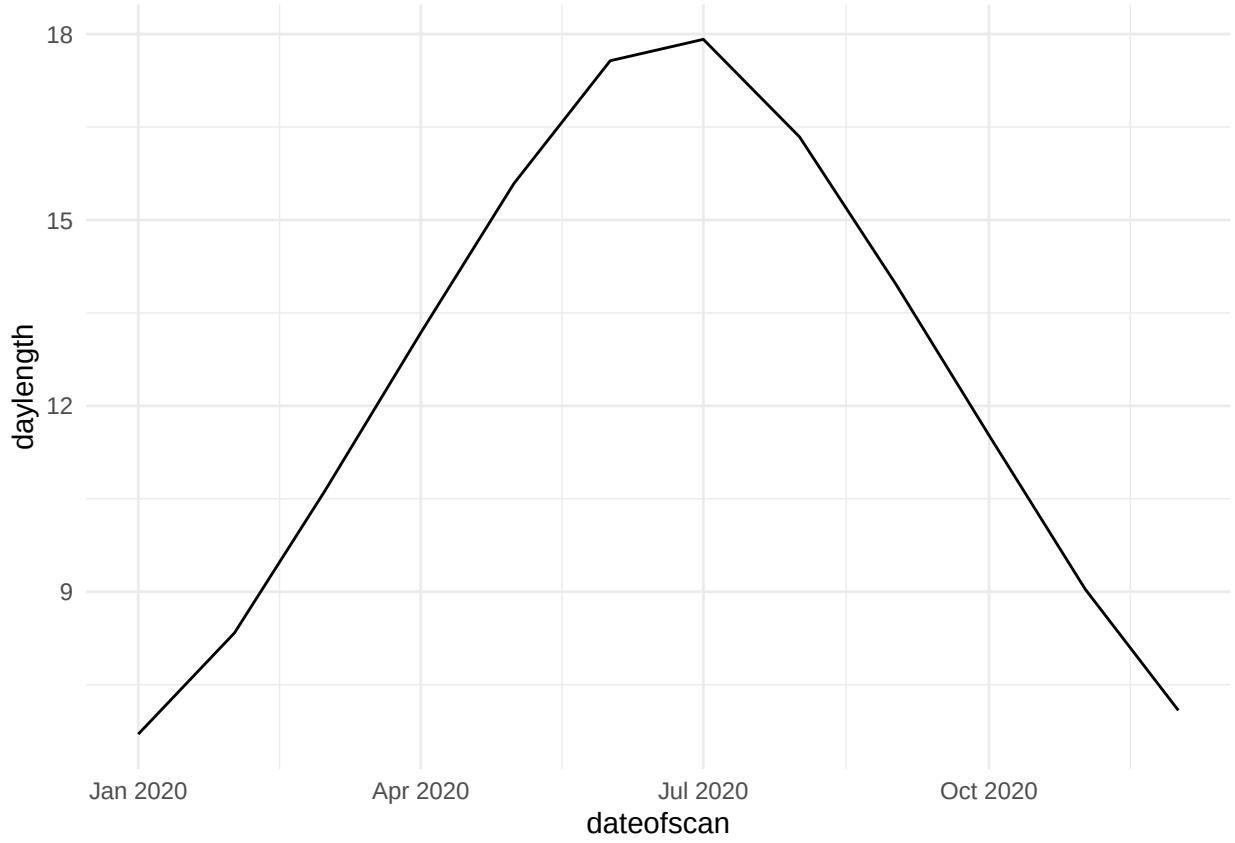
4.27 calculate dates and photoperiod

(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()
```



4.28 Batch reading csvs

4.28.1 Method 1 This function is needed to read_csv for multiple files - with the id option to add the source file to the dataframe as a new column

```
# # Your custom function to read CSV files with options
# read_csv_with_options <- function(file_path) {
#   read_csv(data_files,id='filename',skip = 5)  # Replace with your desired options
# }
# # use read_csv to allow use of id= function, not available with read_table
```

4.28.2 Option 2

This would import multiple csv files into seperate objects each with a name based on the source csv file import each lm file a df with its id as new column

```
# # Install and load tidyverse package
#
# # Get a list of CSV files in the directory
# files <- list.files(path = "data_in/pollution_data", pattern = "\\.csv$", full.names = TRUE)
#
# # Read each CSV file into a separate data frame, skipping the first 5 rows
# data_frames <- map(files, ~ read_csv(.x, skip = 5))
```

```
#
# # Extract file names without extension and use them as unique names for the data frames
# file_names <- tools::file_path_sans_ext(basename(files))
# names(data_frames) <- make.unique(file_names)
#
# # Now, data_frames is a list of data frames with unique names based on their source
# list2env(data_frames, envir = .GlobalEnv)
```

4.28.3 merge all obj based on wildcards (i.e. the equivalent of doing muliple left_joins) all objects must have the same rows

```
# list_obj<- lapply(ls(pattern="\\map"),get)
# benzene <- reduce(list_obj, dplyr::left_join)
```

4.29 add prefix to defined column names

```
mtcars %>% rename_with(.cols = hp:wt, function(x){paste0("cars.", x)})
```

```
##              car_name  mpg cyl  disp cars.hp cars.drat cars.wt  qsec vs am
## 1           Mazda RX4 21.0   6 160.0     110      3.90   2.620 16.46  0  1
## 2       Mazda RX4 Wag 21.0   6 160.0     110      3.90   2.875 17.02  0  1
## 3      Hornet 4 Drive 21.4   6 258.0     110      3.08   3.215 19.44  1  0
## 4   Hornet Sportabout 18.7   8 360.0     175      3.15   3.440 17.02  0  0
## 5             Valiant 18.1   6 225.0     105      2.76   3.460 20.22  1  0
## 6          Duster 360 14.3   8 360.0     245      3.21   3.570 15.84  0  0
## 7           Merc 280 19.2   6 167.6     123      3.92   3.440 18.30  1  0
## 8           Merc 280C 17.8   6 167.6     123      3.92   3.440 18.90  1  0
## 9          Merc 450SE 16.4   8 275.8     180      3.07   4.070 17.40  0  0
## 10         Merc 450SL 17.3   8 275.8     180      3.07   3.730 17.60  0  0
## 11        Merc 450SLC 15.2   8 275.8     180      3.07   3.780 18.00  0  0
## 12 Cadillac Fleetwood 10.4   8 472.0     205      2.93   5.250 17.98  0  0
## 13 Lincoln Continental 10.4   8 460.0     215      3.00   5.424 17.82  0  0
## 14  Chrysler Imperial 14.7   8 440.0     230      3.23   5.345 17.42  0  0
## 15   Dodge Challenger 15.5   8 318.0     150      2.76   3.520 16.87  0  0
## 16        AMC Javelin 15.2   8 304.0     150      3.15   3.435 17.30  0  0
## 17         Camaro Z28 13.3   8 350.0     245      3.73   3.840 15.41  0  0
## 18   Pontiac Firebird 19.2   8 400.0     175      3.08   3.845 17.05  0  0
## 19     Ford Pantera L 15.8   8 351.0     264      4.22   3.170 14.50  0  1
## 20       Ferrari Dino 19.7   6 145.0     175      3.62   2.770 15.50  0  1
## 21      Maserati Bora 15.0   8 301.0     335      3.54   3.570 14.60  0  1
##    gear carb  hp_cat    loghp    sum   loghp1       date last3letters carnames
## 1     4    4 high hp 2.041393 273.90 2.041393 2001-05-24          RX4        1
## 2     4    4 high hp 2.041393 273.90 2.041393 2001-05-24          Wag        2
## 3     3    1 high hp 2.041393 371.08 2.041393 2001-05-24          ive        3
## 4     3    2 high hp 2.243038 538.15 2.243038 2001-05-24          out        4
## 5     3    1 high hp 2.021189 332.76 2.021189 2001-05-24          ant        5
## 6     3    4 high hp 2.389166 608.21 2.389166 2001-05-24          360        6
## 7     4    4 high hp 2.089905 294.52 2.089905 2001-05-24          280        7
## 8     4    4 high hp 2.089905 294.52 2.089905 2001-05-24          80C        8
```

```
## 9     3    3 high hp 2.255273 458.87 2.255273 2001-05-24          0SE        9
## 10    3    3 high hp 2.255273 458.87 2.255273 2001-05-24          0SL       10
## 11    3    3 high hp 2.255273 458.87 2.255273 2001-05-24          SLC       11
## 12    3    4 high hp 2.311754 679.93 2.311754 2001-05-24          ood       12
## 13    3    4 high hp 2.332438 678.00 2.332438 2001-05-24          tal       13
## 14    3    4 high hp 2.361728 673.23 2.361728 2001-05-24          ial       14
## 15    3    2 high hp 2.176091 470.76 2.176091 2001-05-24          ger       15
## 16    3    2 high hp 2.176091 457.15 2.176091 2001-05-24          lin       16
## 17    3    4 high hp 2.389166 598.73 2.389166 2001-05-24          Z28       17
## 18    3    2 high hp 2.243038 578.08 2.243038 2001-05-24          ird       18
## 19    5    4 high hp 2.421604 619.22 2.421604 2001-05-24          a L       19
## 20    5    6 high hp 2.243038 323.62 2.243038 2001-05-24          ino       20
## 21    5    8 high hp 2.525045 639.54 2.525045 2001-05-24          ora       21
##    valves cyc_carb
## 1      24      6-4
## 2      24      6-4
## 3      24      6-1
## 4      32      8-2
## 5      24      6-1
## 6      32      8-4
## 7      24      6-4
## 8      24      6-4
## 9      32      8-3
## 10     32      8-3
## 11     32      8-3
## 12     32      8-4
## 13     32      8-4
## 14     32      8-4
## 15     32      8-2
## 16     32      8-2
## 17     32      8-4
## 18     32      8-2
## 19     32      8-4
## 20     24      6-6
## 21     32      8-8
```

4.30 standarise certain columns

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()
```

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

gear	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	n
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 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.

# 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 ha
#summarise PCA
summary(pcamodel.pca)
```

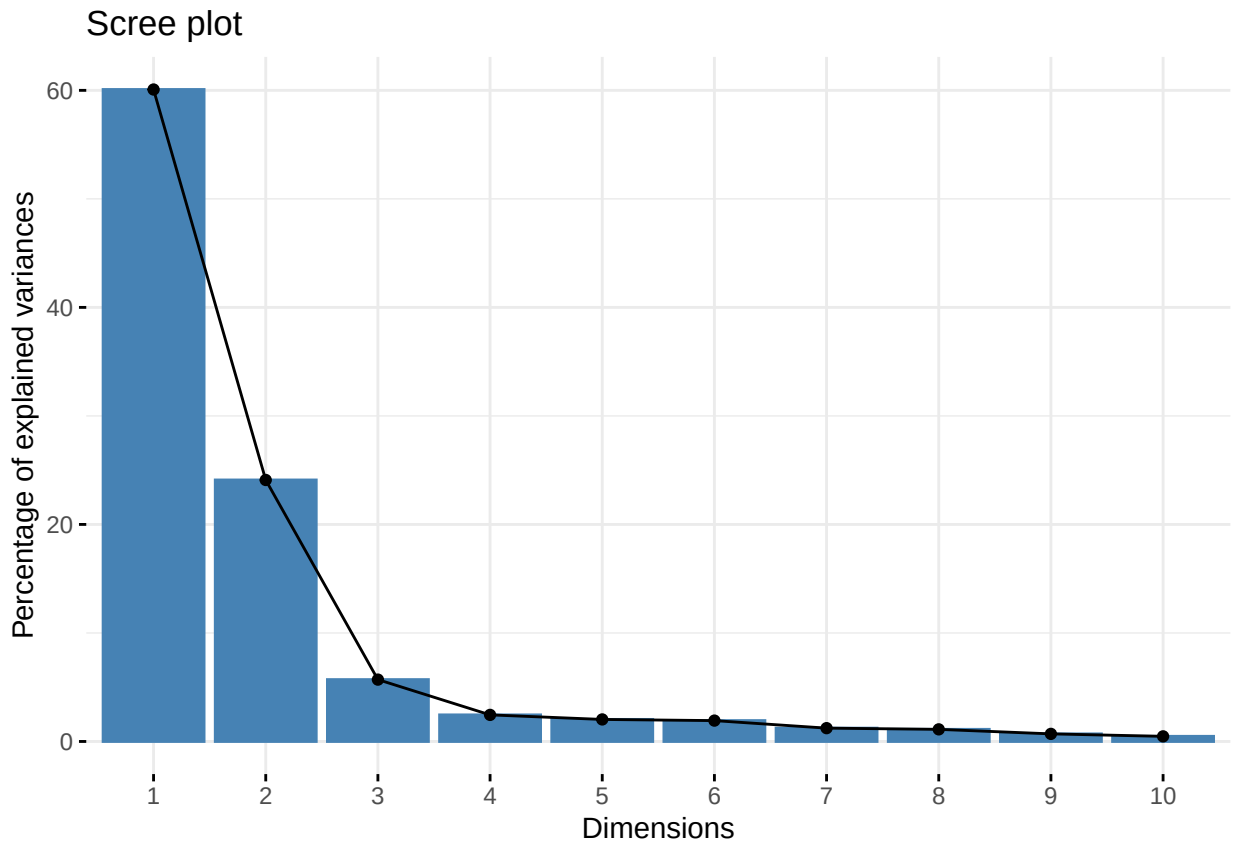
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6    PC7
## Standard deviation     2.5707 1.6280 0.79196 0.51923 0.47271 0.46000 0.3678
## Proportion of Variance 0.6008 0.2409 0.05702 0.02451 0.02031 0.01924 0.0123
## Cumulative Proportion  0.6008 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

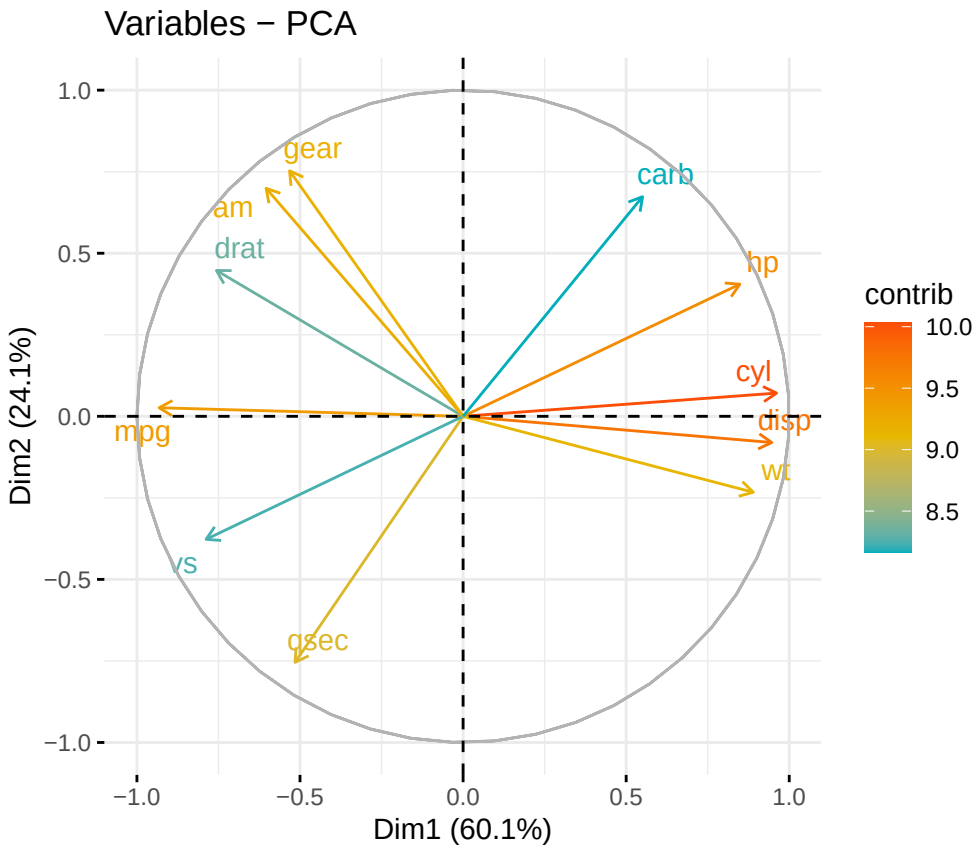
```
#creates scree plot
fviz_eig(pcamodel.pca)
```



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
```

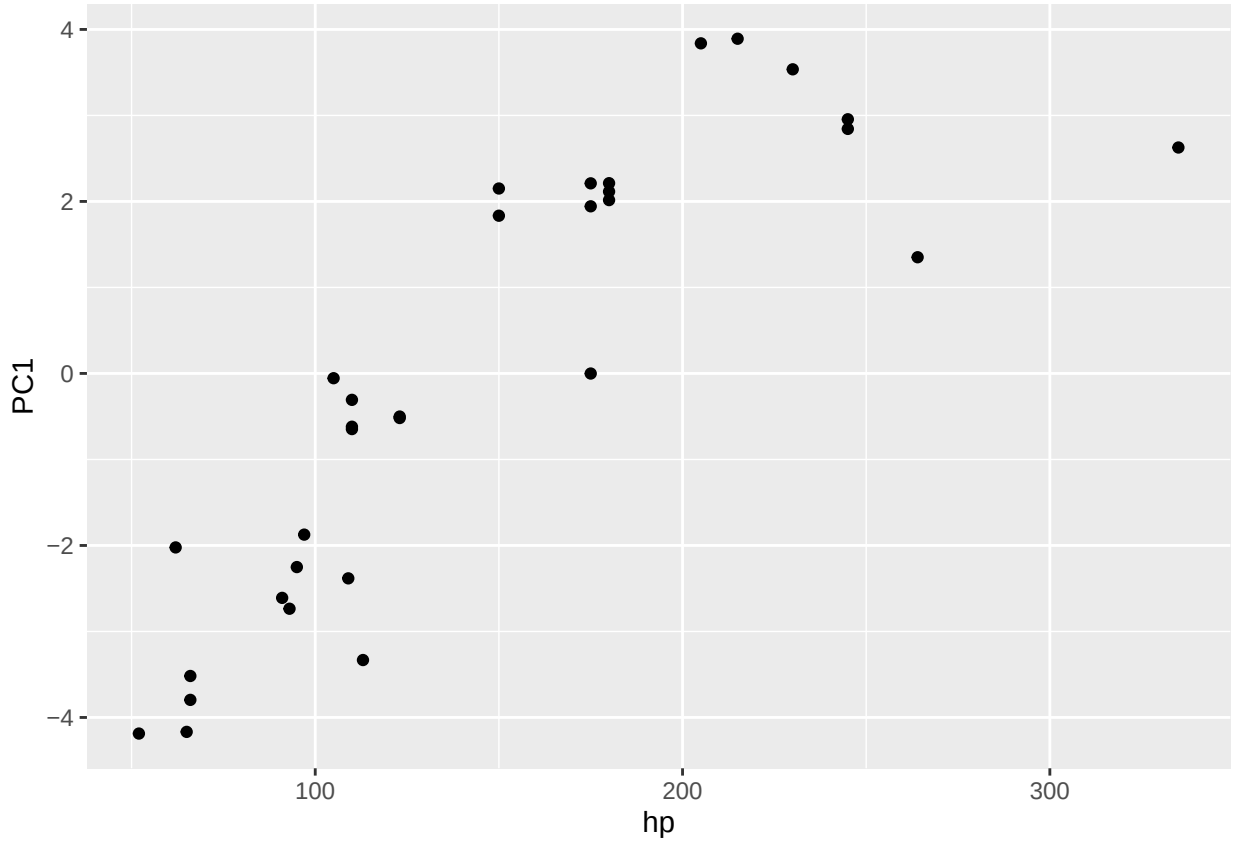
- these are the seven principal components of the first six people. we are only interested really in PC1.

```r
# 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

```r
#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)
```

```
##
## Attaching package: 'UpSetR'
```

```
## The following object is masked from 'package:lattice':
##
##     histogram
```

```
#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 ****
```

5.10 Machine Learning

5.10.1 Simple example

```
data("iris")
iris <- iris %>% clean_names()
```

5.10.1.1 Load example data

```
#make this example reproducible
set.seed(1)

#create ID column
iris$id <- 1:nrow(iris)

#use 70% of dataset as training set and 30% as test set
train_iris <- iris %>% dplyr::sample_frac(0.70)
test_iris  <- dplyr::anti_join(iris, train_iris, by = 'id')


train_iris <- train_iris %>% select(species, sepal_length:petal_width)
test_iris <- test_iris %>% select(species, sepal_length:petal_width)
```

5.10.1.2 Creating random test and train subsets

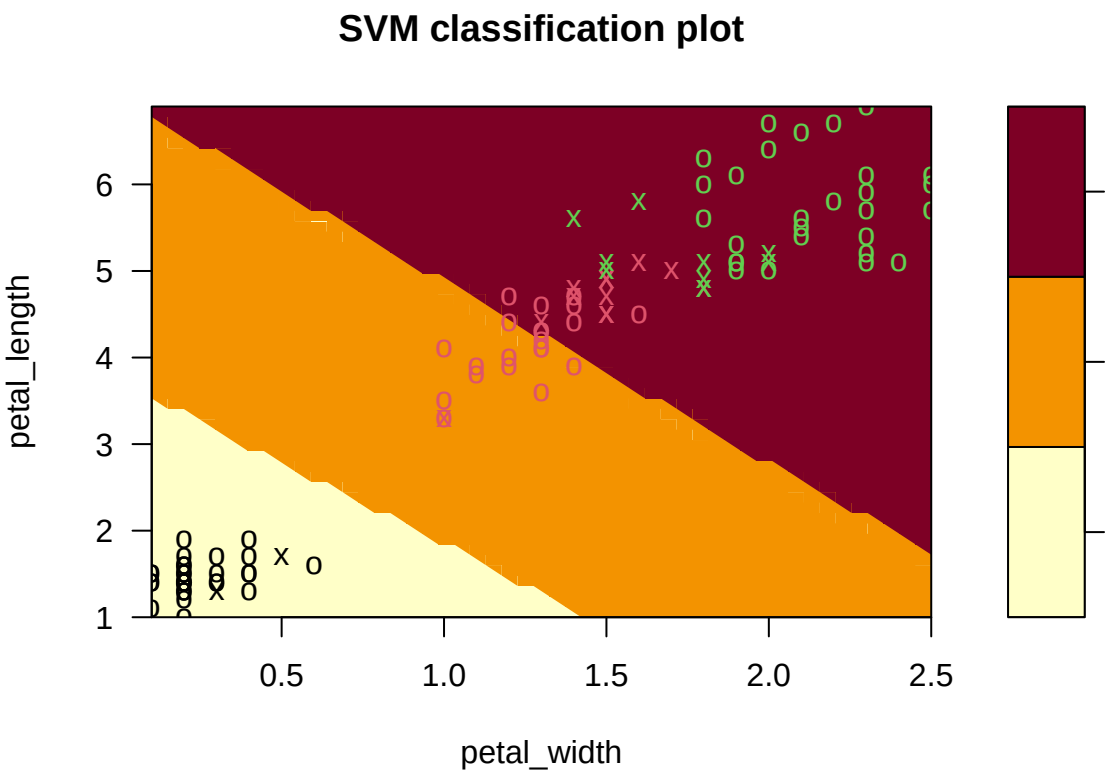
```
iris_svm_model <- svm(species~., data = train_iris, kernel = "linear")
print(iris_svm_model)
```

5.10.1.3 create svm iris model

```
##
## Call:
## svm(formula = species ~ ., data = train_iris, kernel = "linear")
##
##
## Parameters:
```

```
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  23
```

```
plot(iris_svm_model,petal_length~petal_width, data = train_iris)
```



5.10.1.4 Plot results

Test SVM model on test set

```
test_iris <- test_iris %>%
  mutate(pred = predict(iris_svm_model, newdata = ., type = "class"))
```

```
confusionMatrix(test_iris$pred, test_iris$species)
```

5.10.1.5 Percent accuracy

```
## Confusion Matrix and Statistics
##
##             Reference
```

```
## Prediction   setosa versicolor virginica
##   setosa         15          0         0
##   versicolor      0         17         0
##   virginica       0          0        13
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9213, 1)
##     No Information Rate : 0.3778
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            1.0000           1.0000
## Specificity                 1.0000            1.0000           1.0000
## Pos Pred Value              1.0000            1.0000           1.0000
## Neg Pred Value              1.0000            1.0000           1.0000
## Prevalence                  0.3333            0.3778           0.2889
## Detection Rate              0.3333            0.3778           0.2889
## Detection Prevalence        0.3333            0.3778           0.2889
## Balanced Accuracy           1.0000            1.0000           1.0000
```

5.10.2 Clear environment

```
rm(list=setdiff(ls(), "clean environment"))
data("iris")
iris <- as_tibble(iris)
iris <- iris %>% clean_names()
```

5.10.3 caret to choose machine learning method

```
set.seed(123)  # for reproducibility
train_index_iris <- createDataPartition(iris$species, p = 0.8, list = FALSE)
train_data <- iris[train_index_iris, ]
test_data <- iris[-train_index_iris, ]
```

```
# prepare training scheme
control <- trainControl(method="repeatedcv", number=10, repeats=3)
```

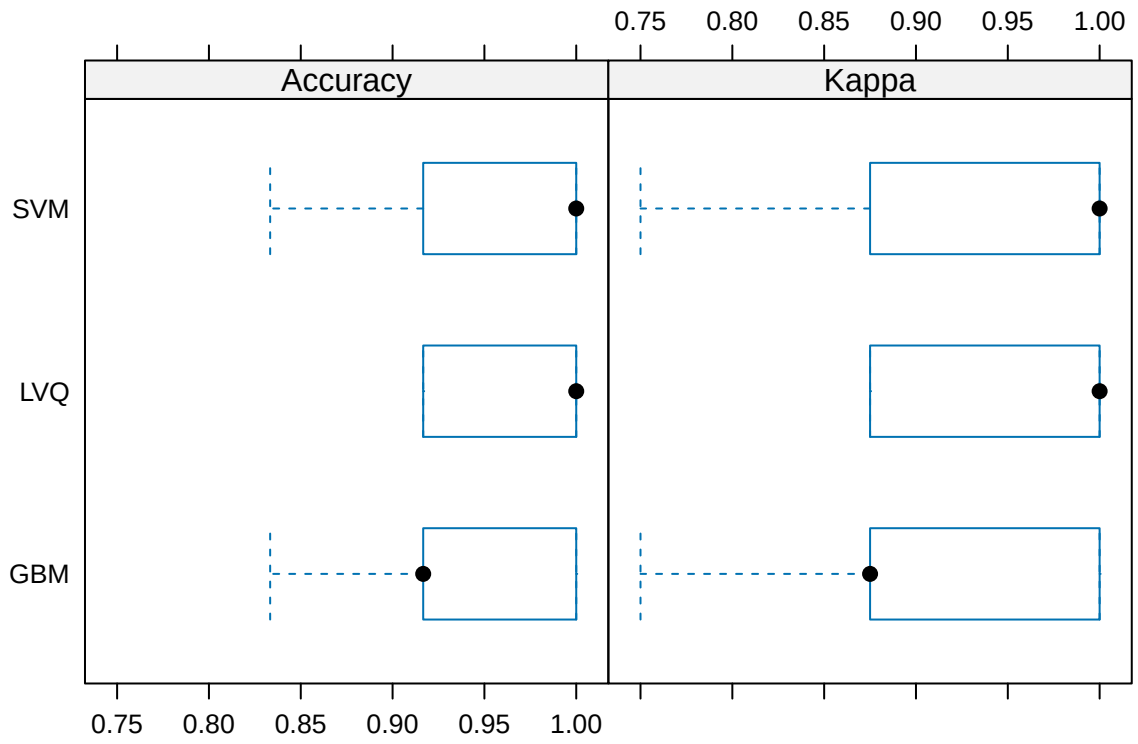
```
# train the LVQ model
set.seed(7)
modelLvq <- train(species~., data=train_data, method="lvq", trControl=control)
# train the GBM model
set.seed(7)
```

```r
modelGbm <- train(species~., data=train_data, method="gbm", trControl=control, verbose=FALSE)
# train the SVM model
set.seed(7)
modelSvm <- train(species~., data=train_data, method="svmRadial", trControl=control)
```

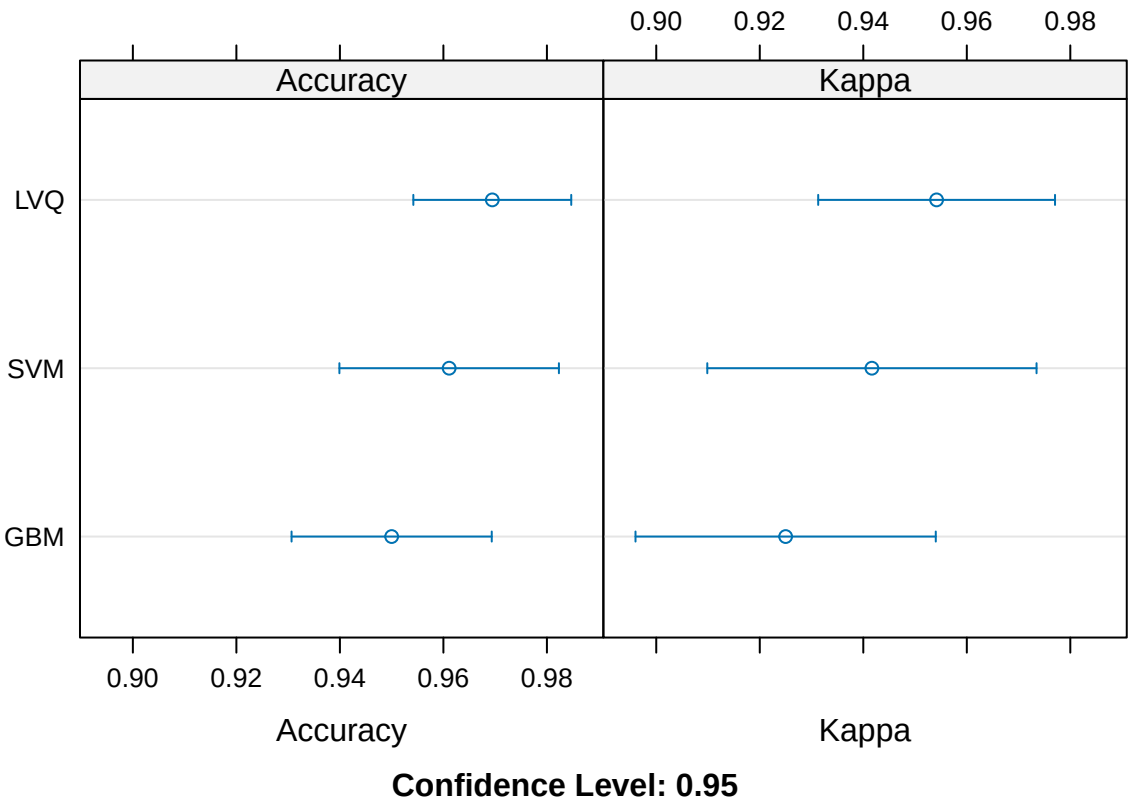
```r
# collect resamples
results <- resamples(list(LVQ=modelLvq, GBM=modelGbm, SVM=modelSvm))
# summarize the distributions
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: LVQ, GBM, SVM
## Number of resamples: 30
##
## Accuracy
##          Min.   1st Qu.    Median      Mean 3rd Qu. Max. NA's
## LVQ 0.9166667 0.9166667 1.0000000 0.9694444       1    1    0
## GBM 0.8333333 0.9166667 0.9166667 0.9500000       1    1    0
## SVM 0.8333333 0.9166667 1.0000000 0.9611111       1    1    0
##
## Kappa
##      Min. 1st Qu. Median      Mean 3rd Qu. Max. NA's
## LVQ 0.875   0.875  1.000 0.9541667       1    1    0
## GBM 0.750   0.875  0.875 0.9250000       1    1    0
## SVM 0.750   0.875  1.000 0.9416667       1    1    0
```

```r
# boxplots of results
bwplot(results)
```




```
# dot plots of results
dotplot(results)
```



5.10.4 Use caret to tune the parameters of machine learning

```
rm(list=setdiff(ls(), "clean environment"))
data("iris")
iris <- as_tibble(iris)
iris <- iris %>% clean_names()
```

5.10.4.1 Clear environment

5.10.5 caret to choose machine learning method

```
set.seed(123)  # for reproducibility
train_index_iris <- createDataPartition(iris$species, p = 0.8, list = FALSE)
train_data <- iris[train_index_iris, ]
test_data <- iris[-train_index_iris, ]
```

```
# ensure results are repeatable
set.seed(7)
```

```r
# prepare training scheme
control <- trainControl(method="repeatedcv", number=10, repeats=3)
```

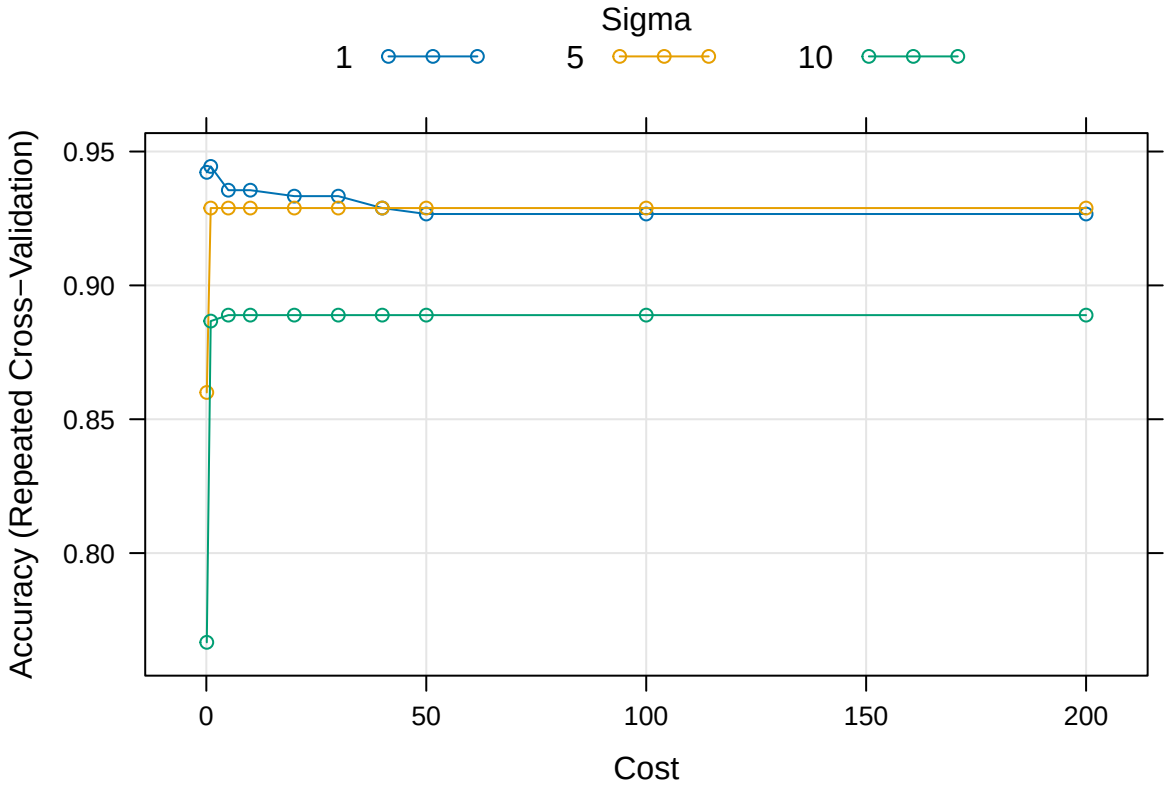
```r
# design the parameter tuning grid
grid <- expand.grid(C=c(0.1,1,5,10,20,30,40,50,100,200), sigma=c(1,5,10))
# train the model
model <- train(species~., data=iris, method="svmRadial", trControl=control, tuneGrid=grid)
# summarize the model
print(model)
```

5.10.5.1 set the training scheme for a ML method (e.g. svm)

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 150 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
## Resampling results across tuning parameters:
##
##   C      sigma  Accuracy   Kappa
##     0.1   1     0.9422222  0.9133333
##     0.1   5     0.8600000  0.7900000
##     0.1  10     0.7666667  0.6500000
##     1.0   1     0.9444444  0.9166667
##     1.0   5     0.9288889  0.8933333
##     1.0  10     0.8866667  0.8300000
##     5.0   1     0.9355556  0.9033333
##     5.0   5     0.9288889  0.8933333
##     5.0  10     0.8888889  0.8333333
##    10.0   1     0.9355556  0.9033333
##    10.0   5     0.9288889  0.8933333
##    10.0  10     0.8888889  0.8333333
##    20.0   1     0.9333333  0.9000000
##    20.0   5     0.9288889  0.8933333
##    20.0  10     0.8888889  0.8333333
##    30.0   1     0.9333333  0.9000000
##    30.0   5     0.9288889  0.8933333
##    30.0  10     0.8888889  0.8333333
##    40.0   1     0.9288889  0.8933333
##    40.0   5     0.9288889  0.8933333
##    40.0  10     0.8888889  0.8333333
##    50.0   1     0.9266667  0.8900000
##    50.0   5     0.9288889  0.8933333
##    50.0  10     0.8888889  0.8333333
##   100.0   1     0.9266667  0.8900000
##   100.0   5     0.9288889  0.8933333
##   100.0  10     0.8888889  0.8333333
```

```
##   200.0   1     0.9266667  0.8900000
##   200.0   5     0.9288889  0.8933333
##   200.0  10     0.8888889  0.8333333
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 1 and C = 1.
```

```
plot(model)
```




```
print(model$finalModel)
```

5.10.5.2 Use Caret to show and use the best algorithm

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  1
##
## Number of Support Vectors : 63
```

```
##
## Objective Function Value : -5.1016 -5.7584 -19.8262
## Training error : 0.013333
```

```
predictions <- predict.train(model, newdata = test_data)
```

```
confusionMatrix(test_data$species, predictions)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0         10         0
##   virginica       0          1         9
##
## Overall Statistics
##
##                Accuracy : 0.9667
##                  95% CI : (0.8278, 0.9992)
##     No Information Rate : 0.3667
##     P-Value [Acc > NIR] : 4.476e-12
##
##                   Kappa : 0.95
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            0.9091           1.0000
## Specificity                 1.0000            1.0000           0.9524
## Pos Pred Value              1.0000            1.0000           0.9000
## Neg Pred Value              1.0000            0.9500           1.0000
## Prevalence                  0.3333            0.3667           0.3000
## Detection Rate              0.3333            0.3333           0.3000
## Detection Prevalence        0.3333            0.3333           0.3333
## Balanced Accuracy           1.0000            0.9545           0.9762
```

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
```

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

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

```
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.githubusercontent.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)))
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
```

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

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
##       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")
```

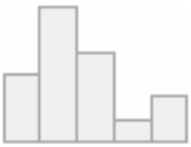
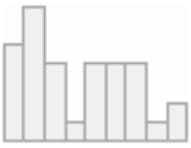
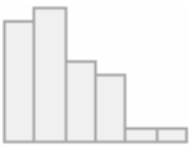
```
## temporary images written to 'C:\tmp'
```

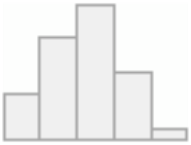
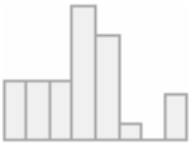
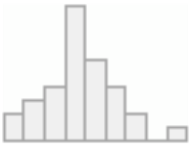
	Mean	Std.Dev	N.Valid
am	0.406	0.499	32
carb	2.812	1.615	32
cyl	6.188	1.786	32
disp	230.722	123.939	32
drat	3.597	0.535	32
gear	3.688	0.738	32
hp	146.688	68.563	32
mpg	20.091	6.027	32
qsec	17.849	1.787	32
vs	0.438	0.504	32
wt	3.217	0.978	32

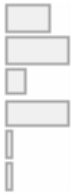
6.2.6 Data Frame Summary

6.2.6.1 mtcars **Dimensions:** 32 x 11

Duplicates: 0

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

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
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%)
9	am [numeric]	Min : 0 Mean : 0.4 Max : 1	0 : 19 (59.4%) 1 : 13 (40.6%)		0 (0.0%)
10	gear [numeric]	Mean (sd) : 3.7 (0.7) 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%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
11	carb [numeric]	Mean (sd) : 2.8 (1.6) 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

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

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

P <- corrp$p
M <- corr1$r
#M
#colnames(M) <- c("mpg", "HP", "Axle Ratio", "Weight (kPounds)", "Quarter Mile (s)")
#rownames(M) <-  c("mpg", "HP", "Axle 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

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`.
```

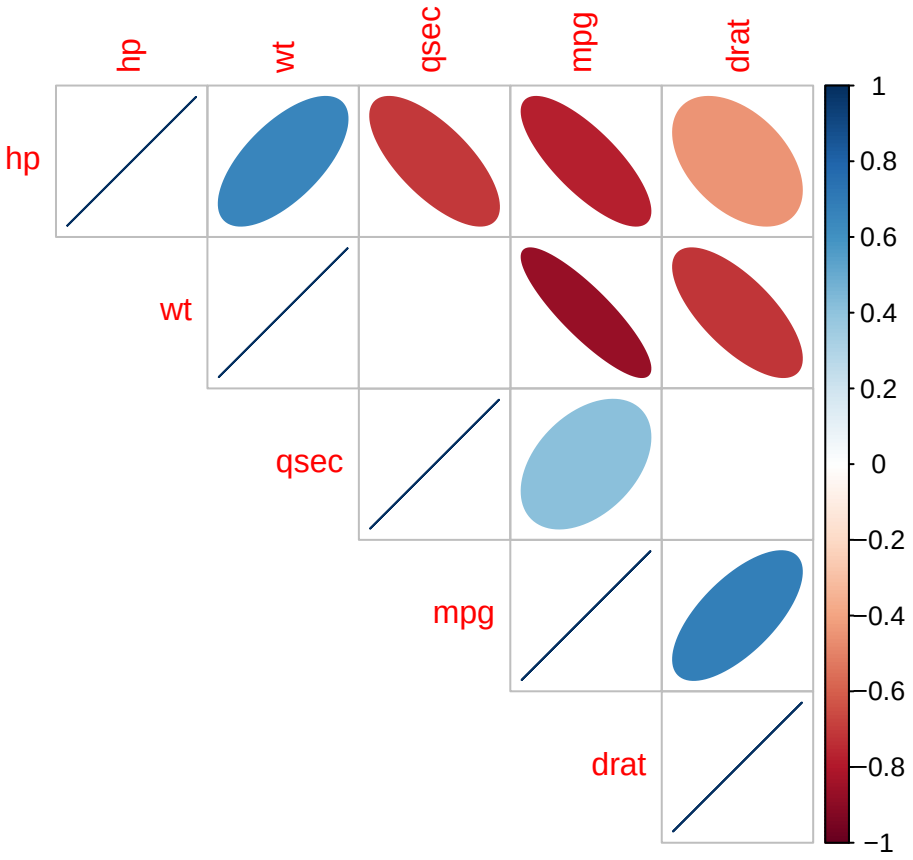
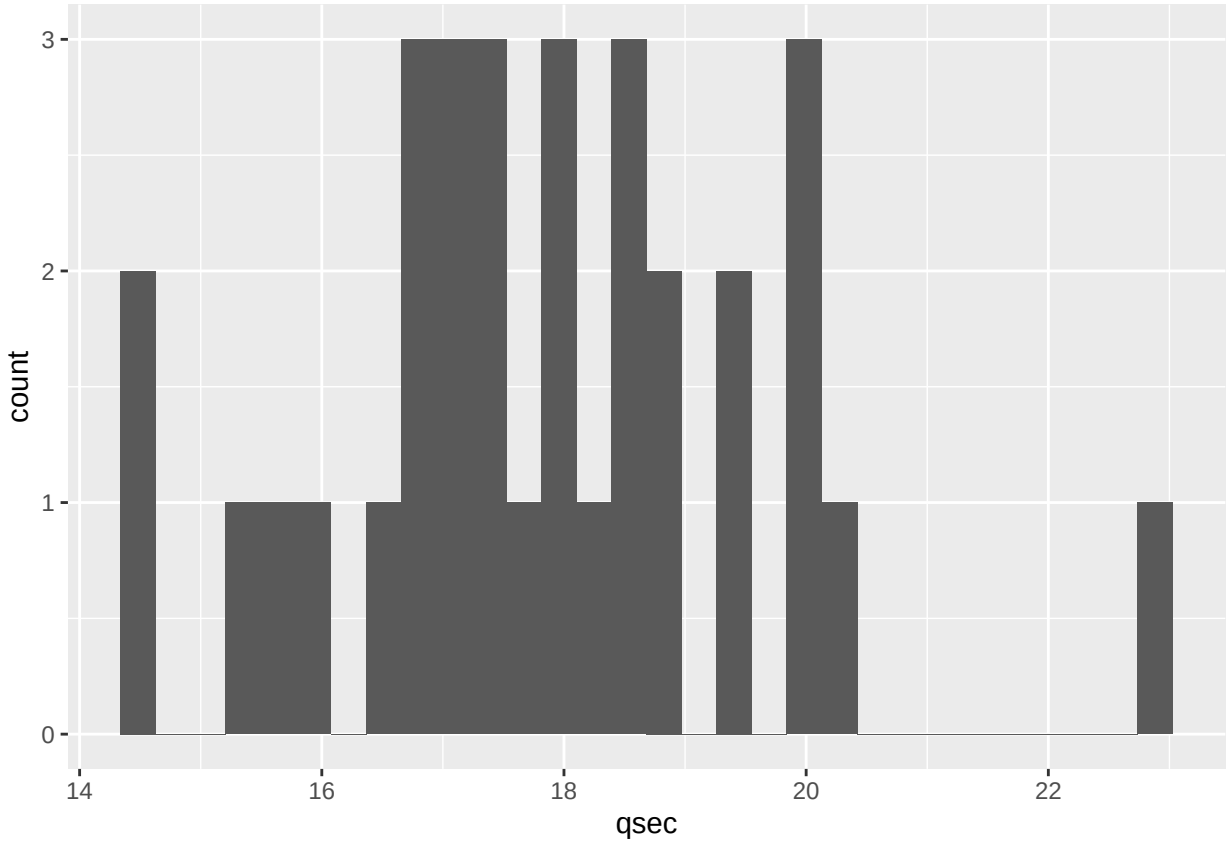


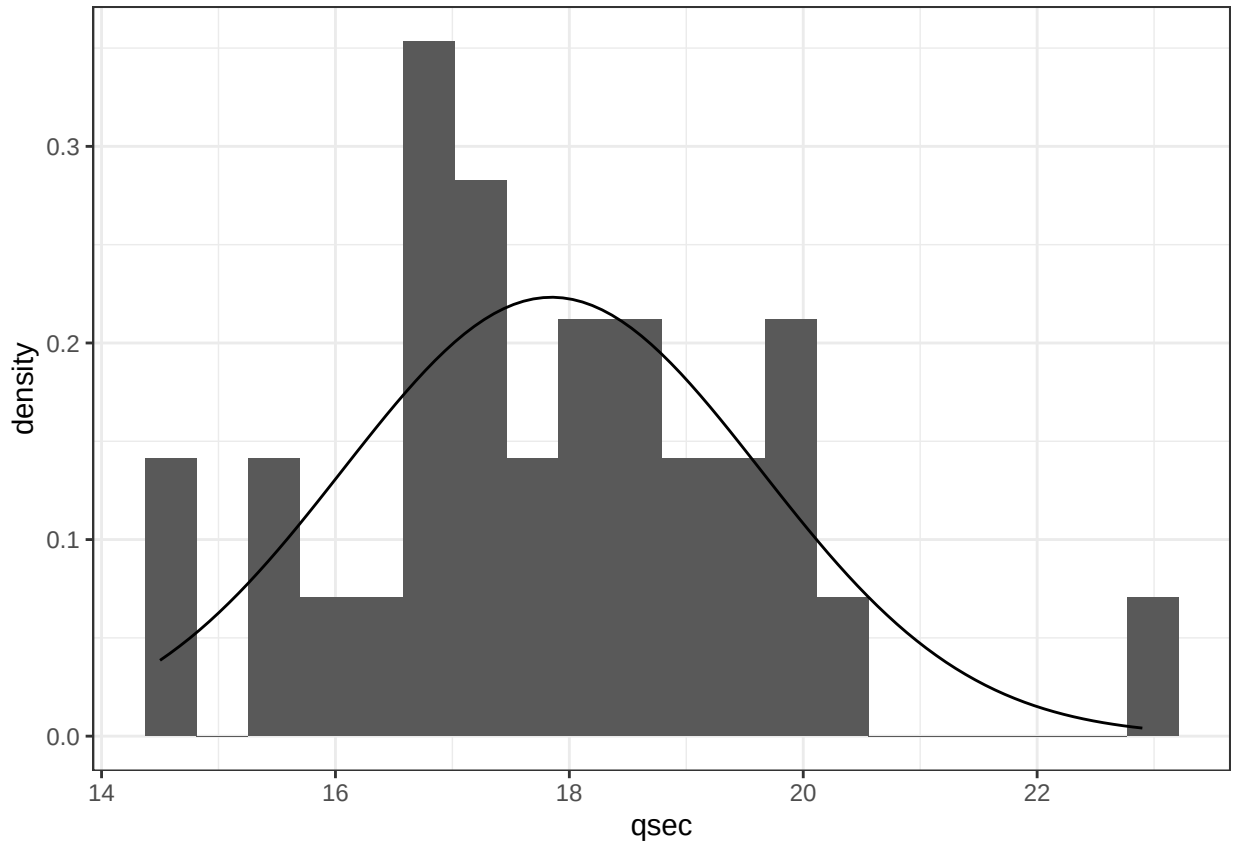
Figure 2: Correlation Plot



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()
```

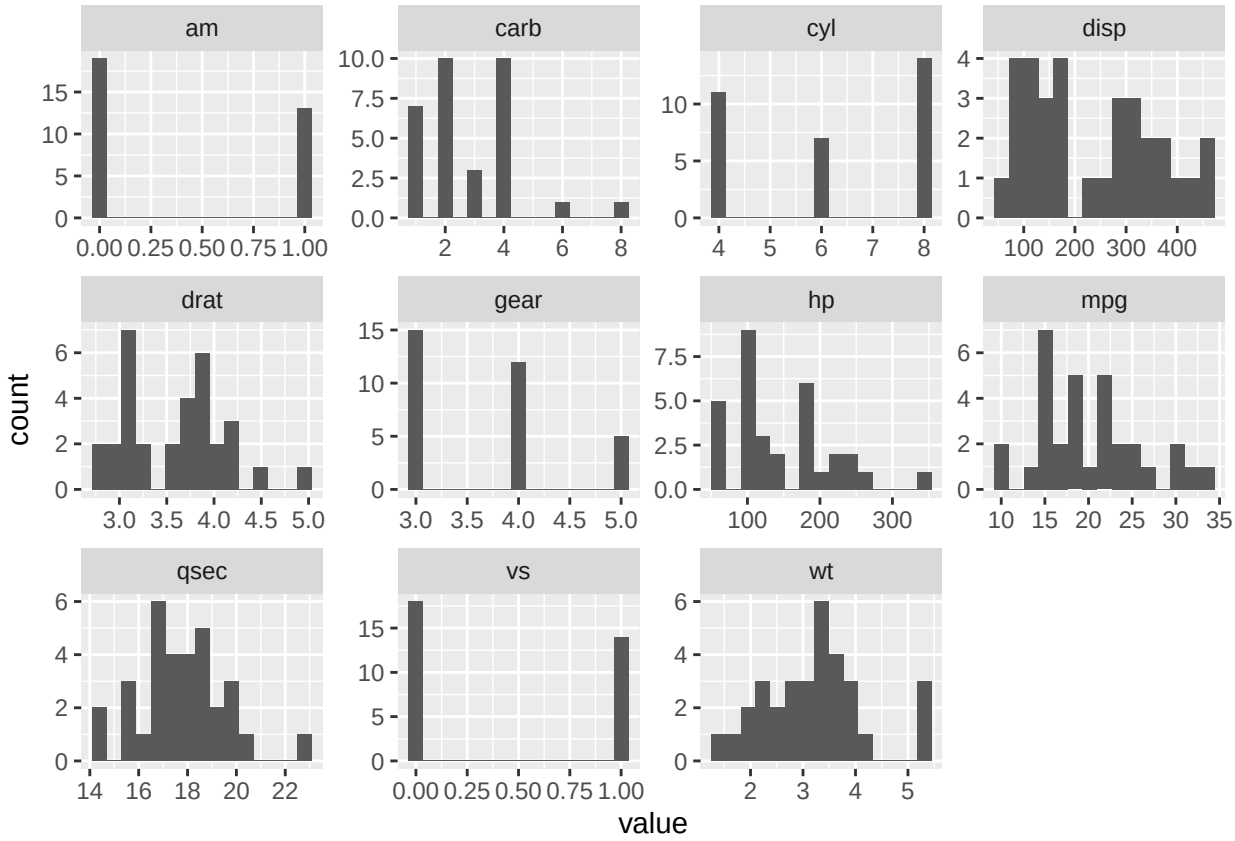
```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




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

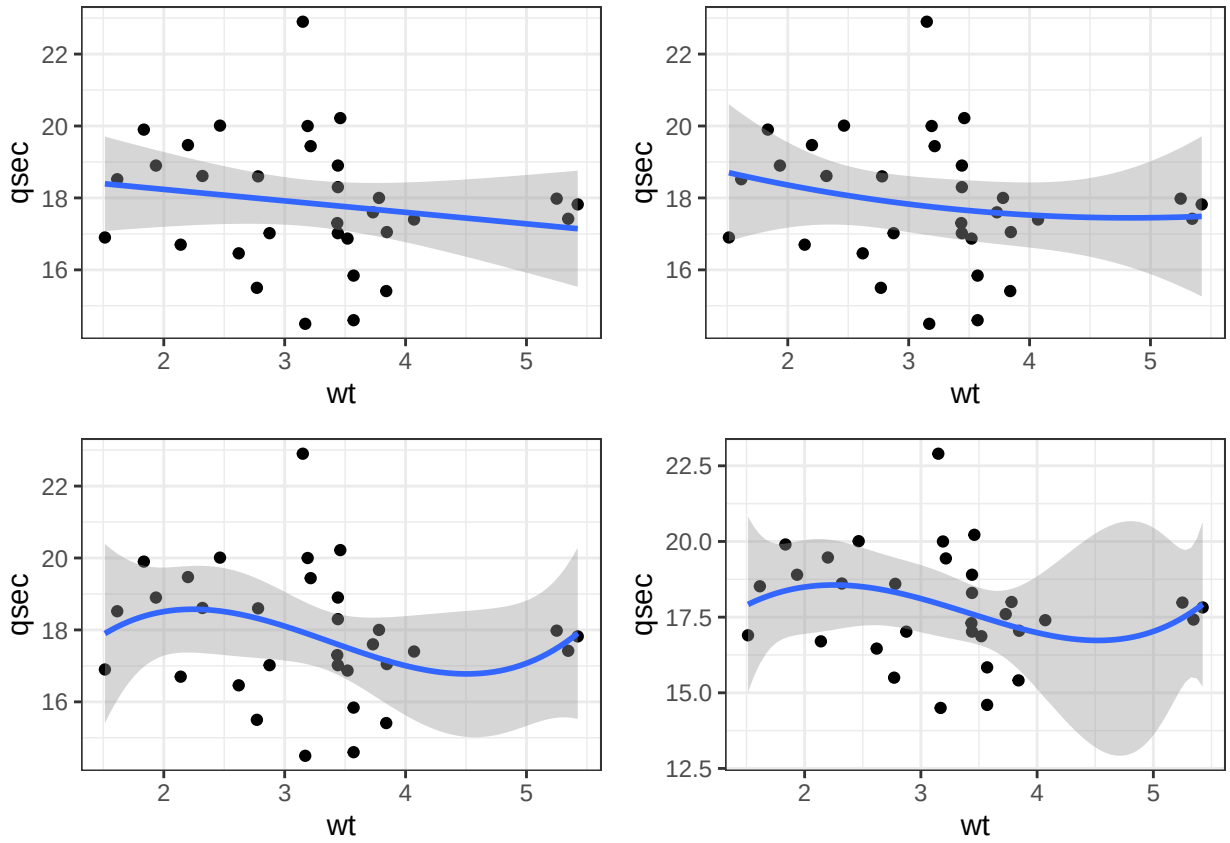
6.4.2 multiple plot of all distributions

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



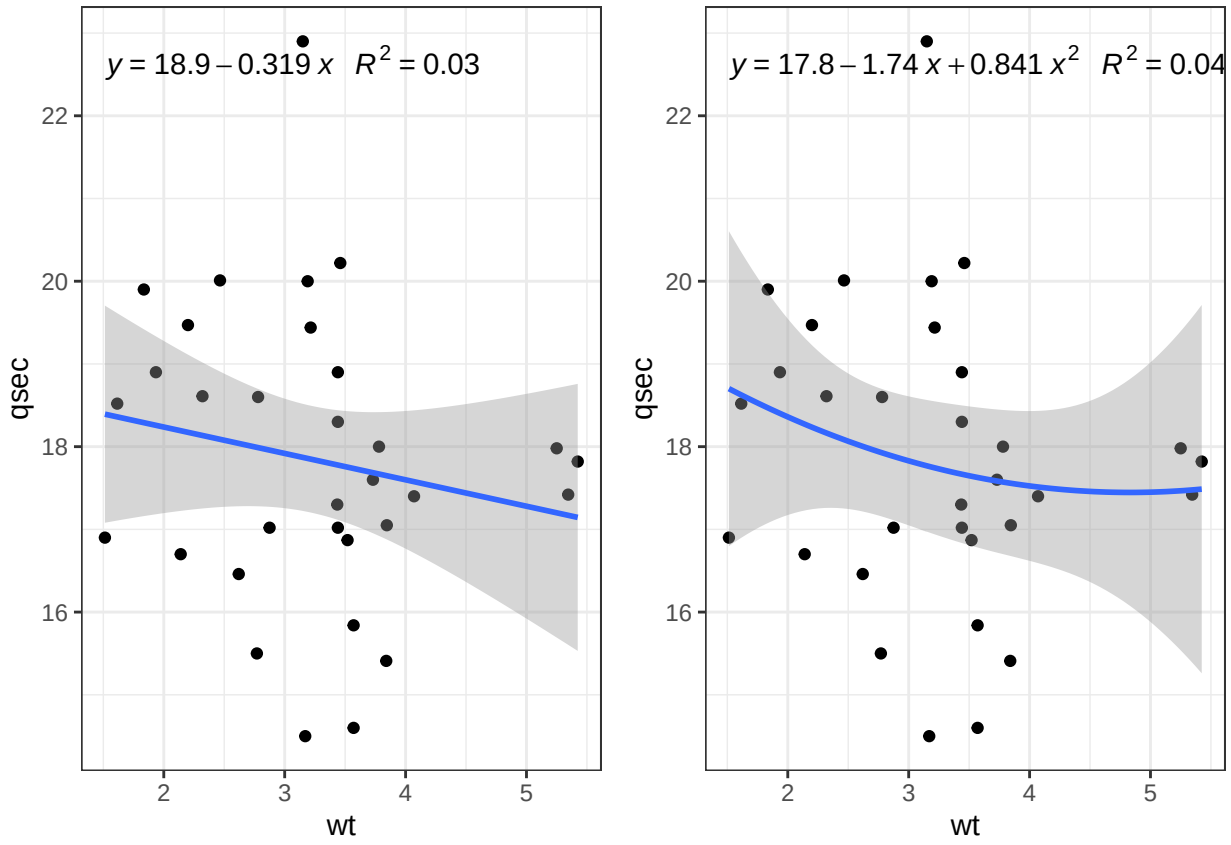
6.4.3 x*y scatterplot with linear or polynomial regression

```
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

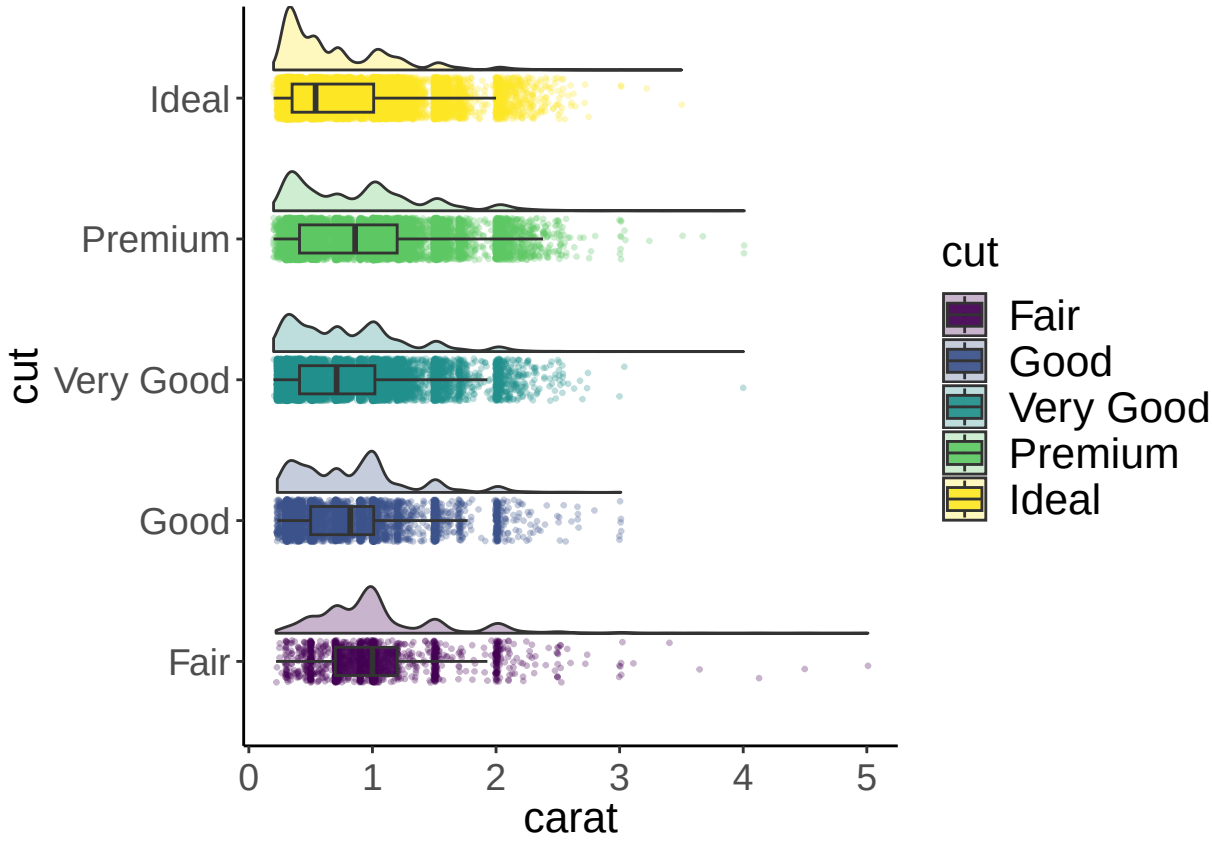
6.4.6 raincloud plot (diamonds)

```
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),

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.
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
```

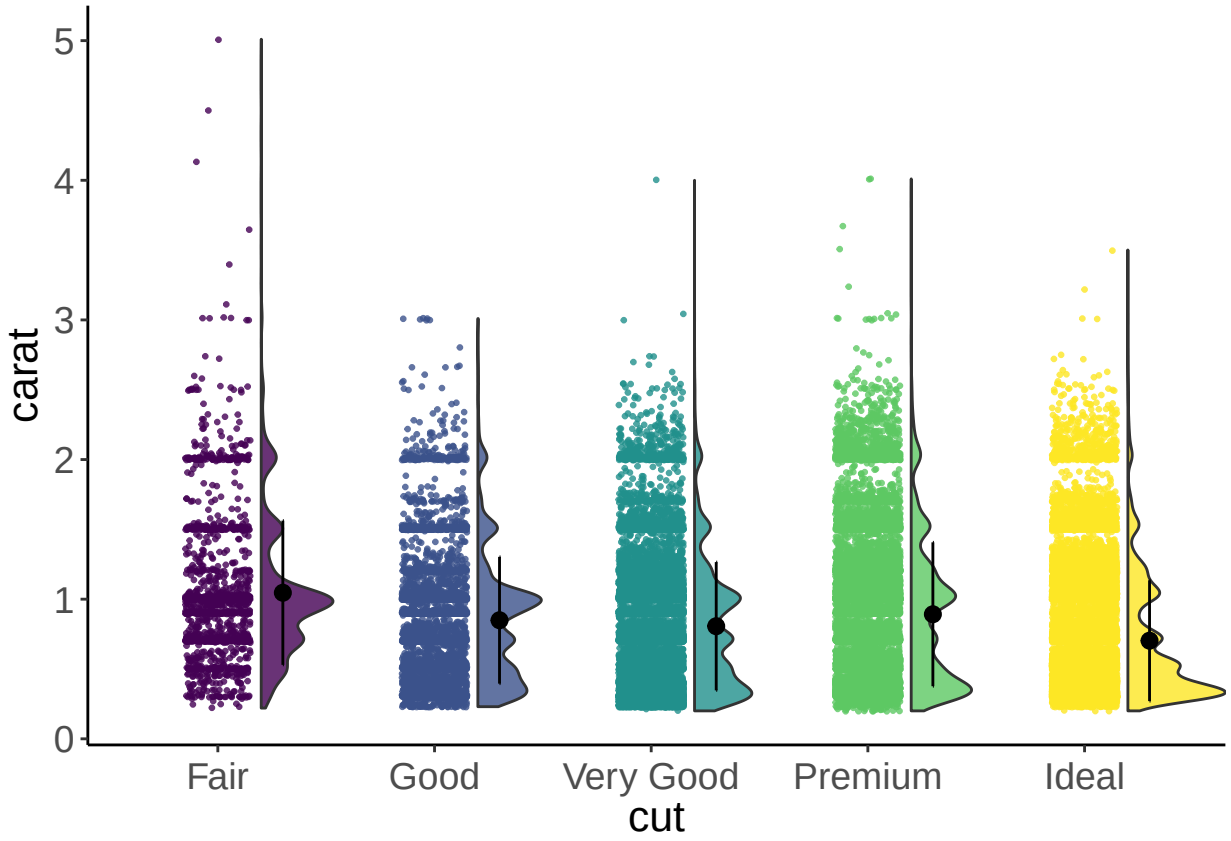


Figure 3: Raincoud plot of means

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')
```

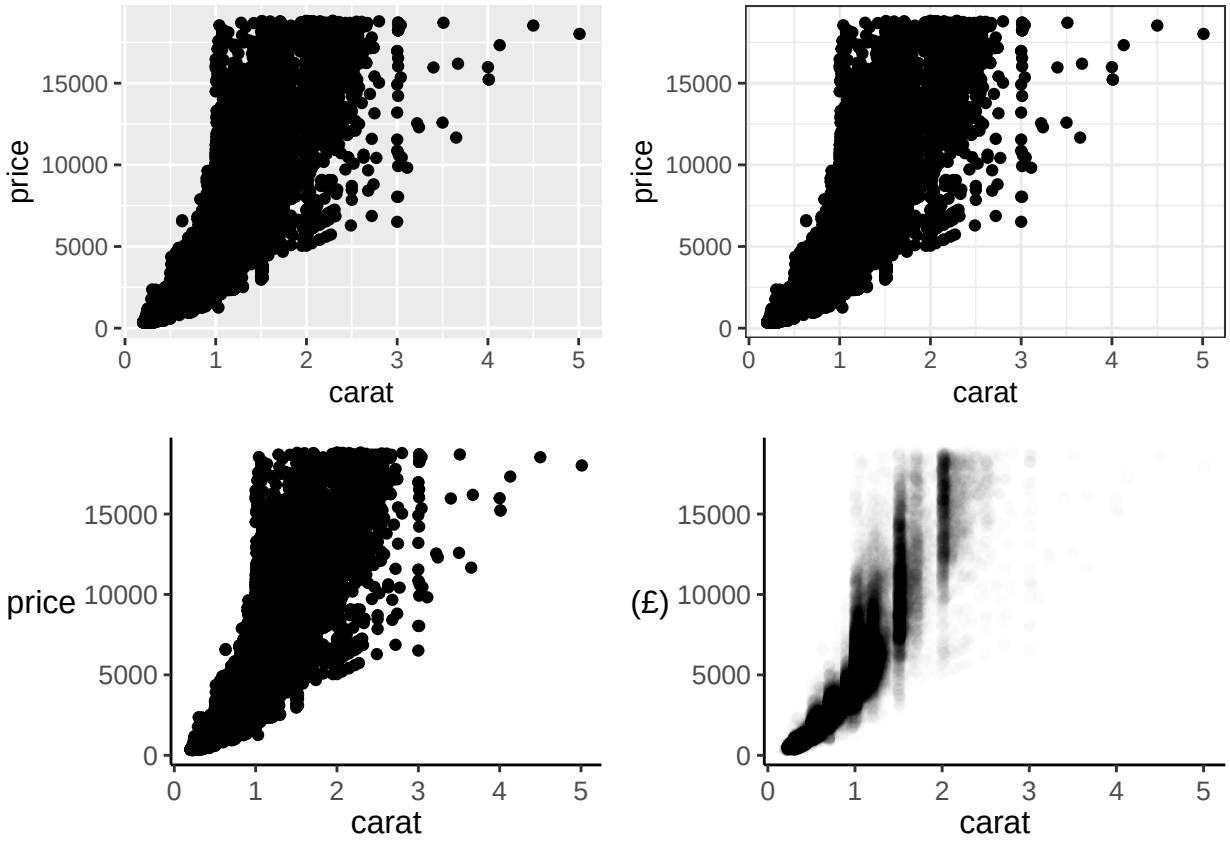
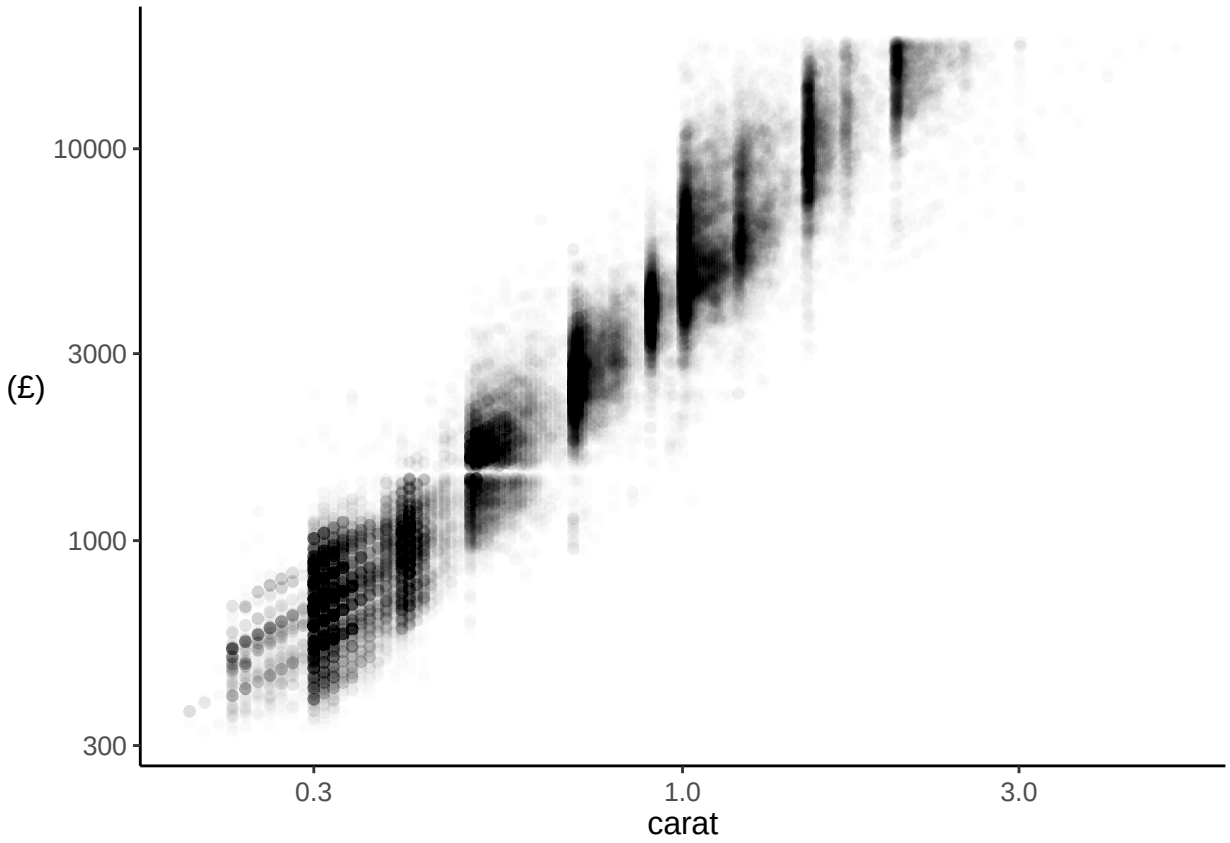


Figure 4: Scatterplots



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()`).
```

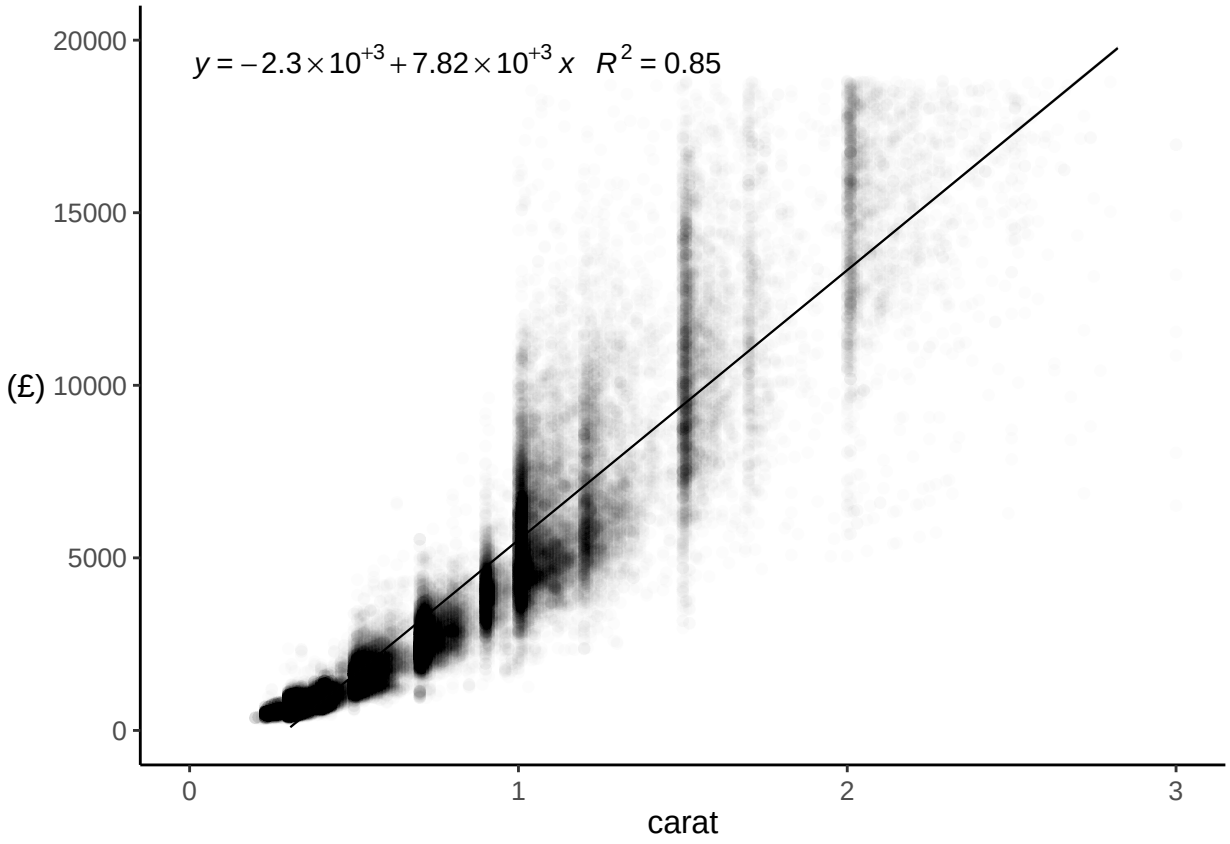


Figure 5: linear Trendline

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()`).
```

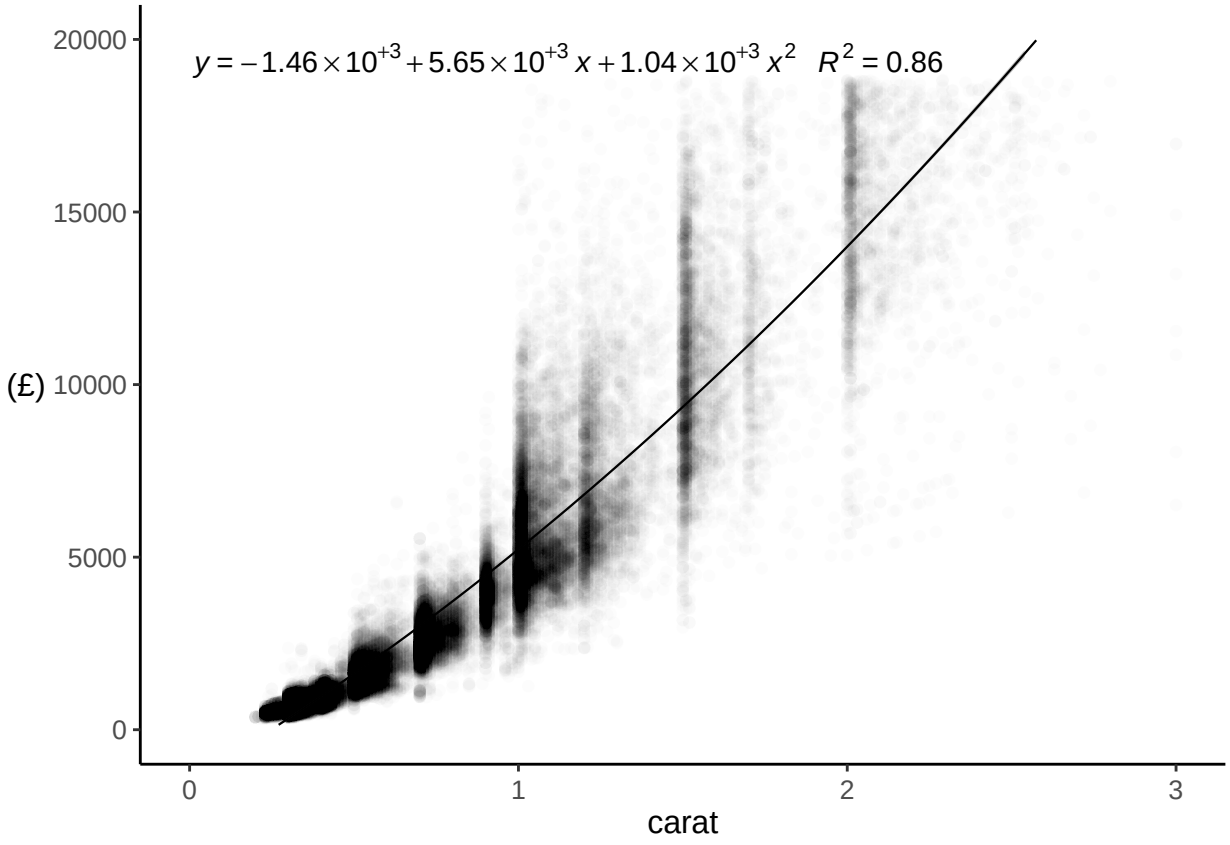
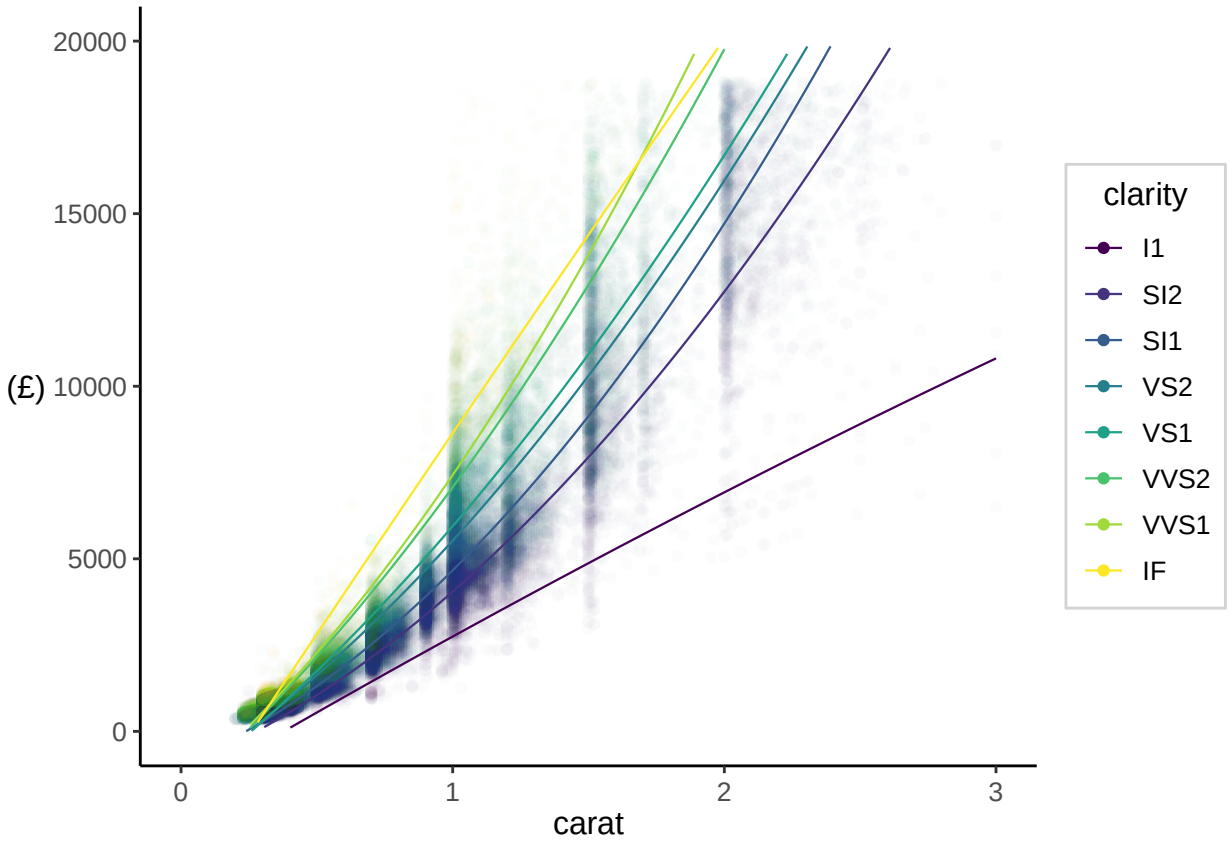


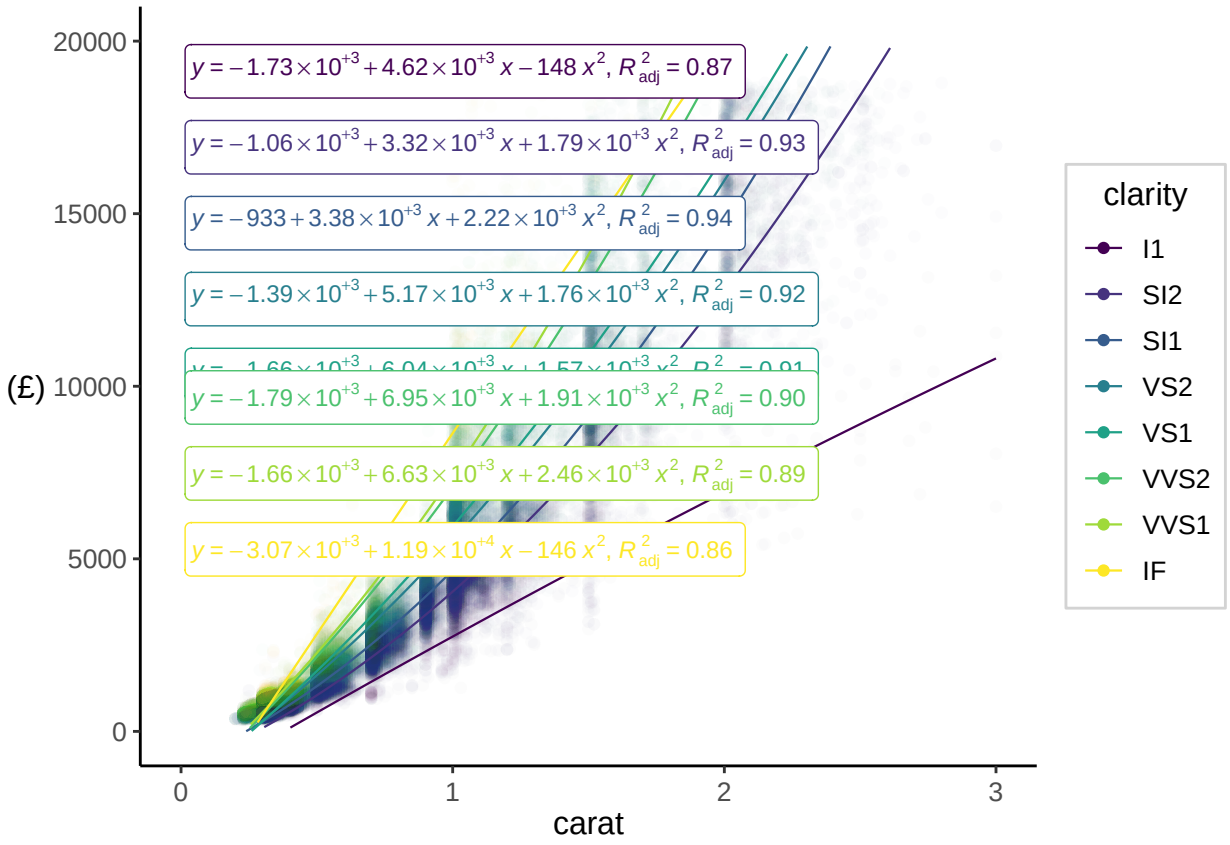
Figure 6: Polynomial trendline

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,s
#   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=
  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?
```

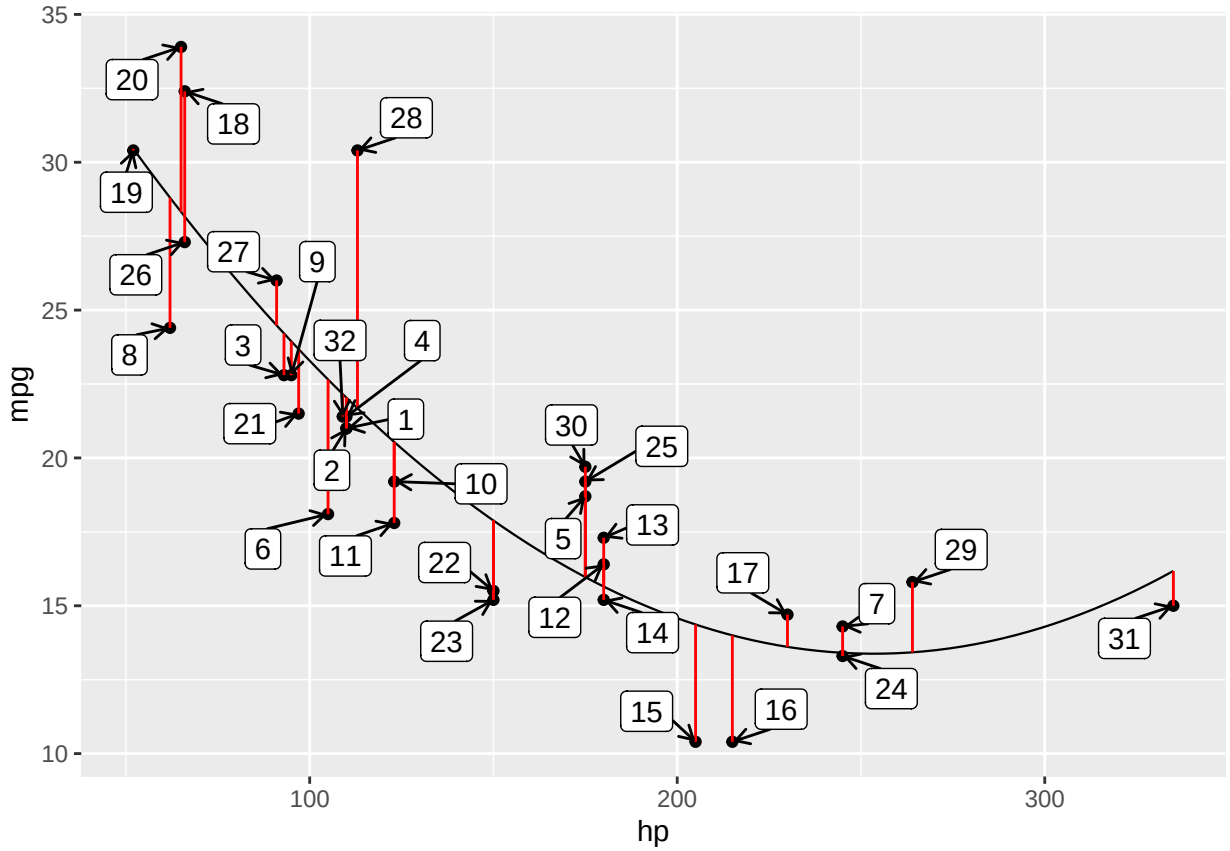


Figure 7: Deviation from predection

```
#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

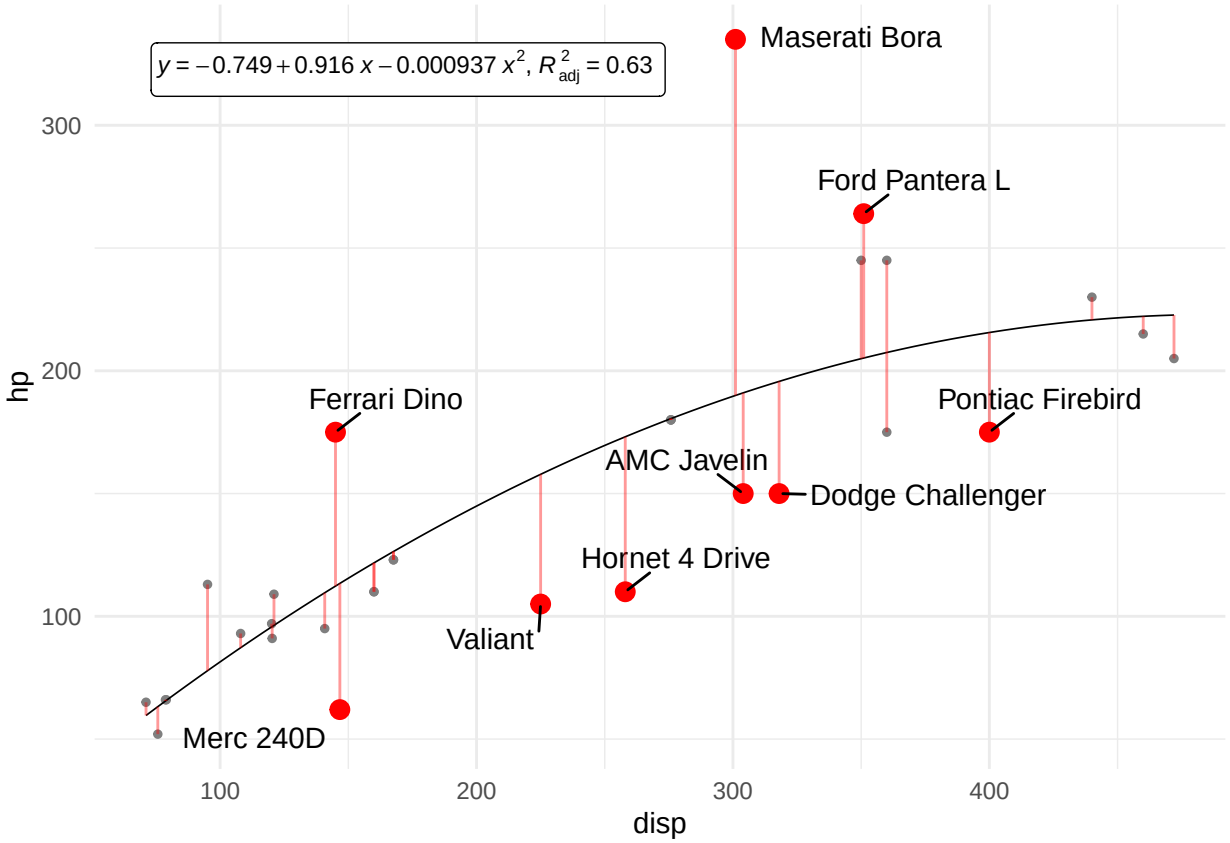
```
kable(head(mtcars[-c(6:12)]))%>%
  kable_styling(full_width = FALSE) %>% kable_minimal()
```

```
p2 <- hpvmpg+geom_point()+geom_smooth(method='lm',formula =formula,colour='black', size=.4,alpha=.6,se=
  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

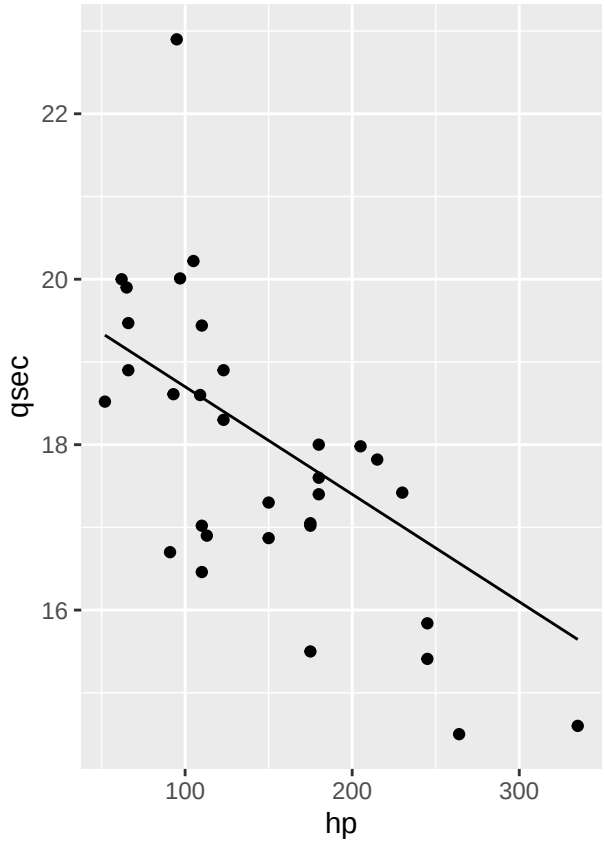
```
formula <- y ~ poly(x, 2,raw=TRUE) # calc formula for display
dat2 <- mtcars
dat2$car_name <- ""
ix_label <- which(mtcars$sds == 1)
dat2$car_name[ix_label] <- mtcars$car_name[ix_label]
hpvmpg <- dat2 %>% ggplot(aes(x=disp,y=hp,label=car_name))

hpvmpg+geom_point(color = ifelse(dat2$car_name == "", "grey50", "red"),size = ifelse(dat2$car_name == "
    box.padding = .55)+geom_smooth(method='lm',formula =formula,colour='black', size=.3,alpha=.6,se=F)+
  stat_fit_deviations(formula = formula, colour = "red",size=.5,alpha=.4)+
  stat_poly_eq(aes(label =  paste(stat(eq.label),
                                  stat(adj.rr.label), sep = "*\", \"*")),
               formula = formula, parse = TRUE,size=3,  geom = "label_npc")+theme_minimal()
```

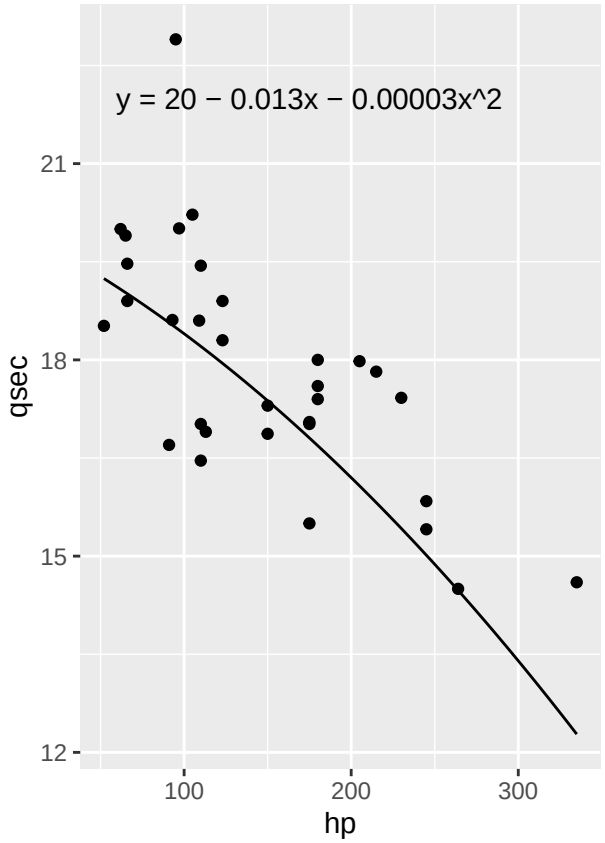
```
## Warning: The following aesthetics were dropped during statistical transformation: label
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
##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


22
20
18
16
qsec
100
200
300
hp


y = 20 − 0.013x − 0.00003x^2
21
18
15
12
qsec
100
200
300
hp