Useful R Syntax

Dr Chris McNeil

2021-05-12

Contents

Chapter 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

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

Figure 1.1: Don’t Panic

    - recode factors
    - dealing with missing data
    - Data reduction with PCA
    - Data standardisation

- 4. Statistical analysis

- 5. Data Visualisation

    - Tables
    - Plots

Chapter 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)
```

Chapter 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
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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

```
## Warning: package 'janitor' was built under R version 4.0.5
```

```
##
## Attaching package: 'janitor'
```

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

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

```
## Warning: package 'knitr' was built under R version 4.0.5
```

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

```
## Warning: package 'kableExtra' was built under R version 4.0.5
```

```
##
## Attaching package: 'kableExtra'
```

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

```
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.0.5
```

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

```
## Warning: package 'geosphere' was built under R version 4.0.5
```

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 ins
rm(list=setdiff(ls(), "band_instruments")) # Everything except "bandinstruments"
rm(list=ls()) # Remove everything
```

Reference:Stackoverflow

3.5 Save dataframe (CSV or Rdata)

make date string

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

```
## function ()
## .Internal(date())
## <bytecode: 0x000000000ee84790>
## <environment: namespace:base>
```

```
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

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

3.6 Save a diagram or plot

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

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

pdf ‘device’ off.

name	band
Mick	Stones
John	Beatles
Paul	Beatles

name	band
m	Stones
John	Beatles
Paul	Beatles

3.7 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

3.8 Alter variable names:

Remove underscores

```
data("mtcars")
mtcars <- mtcars %>% rename(hp_new=hp)
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

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

Chapter 4

Data Wrangling and manipulation

```r
library(Hmisc) #impute values
```

```
## Warning: package 'Hmisc' was built under R version 4.0.5

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

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

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

```r
library(naniar) # deal with NAs
```

```
## Warning: package 'naniar' was built under R version 4.0.5
```

```
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
```

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 %>% 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
## # ... with 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))
#or
mtcars <- mtcars %>% mutate(loghp = replace_na(loghp, "missing"))
```

4.6 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
```

4.7 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
```

	Var1	Freq
1	10.4	2
6	15.2	2
14	19.2	2
16	21	2
17	21.4	2
19	22.8	2
23	30.4	2

##Change specific datapoint

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

4.8 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
```

4.9 Impute missing values

4.9.1 To be completedImputing 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 mea
mtcarsmissingvalues$gear <- impute(mtcarsmissingvalues$gear, median) # median
mtcarsmissingvalues$gear <- impute(mtcarsmissingvalues$gear, 4) # replace specific num
```

Reference:

4.10 Keep rows based on a unique value.

e.g. prescription code

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

Reference

4.11 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
```

4.12 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))
```

4.13 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
Mazda RX4	Mazda RX4	24
Mazda RX4 Wag	Mazda RX4 Wag	24
Datsun 710	Datsun 710	16
Hornet 4 Drive	Hornet 4 Drive	24
Hornet Sportabout	Hornet Sportabout	32
Valiant	Valiant	24

```
# 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> "Mazda RX4", "Mazda RX4 Wag", "Datsun 710", "Hornet 4 Drive",~
## $ valves   <dbl> 24, 24, 16, 24, 32, 24
```

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

kable(glimpse(mtcars %>%select(carb:valves) %>%  slice(1:6))) %>% kable_styling(full_w
  kable_minimal()
```

```
## Rows: 6
## Columns: 7
## $ carb     <dbl> 4, 4, 1, 1, 2, 1
## $ hp_cat   <fct> high hp, high hp, low hp, high hp, high hp, high hp
## $ loghp    <chr> "2.04139268515822", "2.04139268515822", "missing", "2.0413926~
## $ sum      <dbl> 273.90, 273.90, 204.85, 371.08, 538.15, 332.76
## $ loghp1   <dbl> 2.041393, 2.041393, 0.000000, 2.041393, 2.243038, 2.021189
## $ carnames <chr> "Mazda RX4", "Mazda RX4 Wag", "Datsun 710", "Hornet 4 Drive",~
## $ valves   <dbl> 24, 24, 16, 24, 32, 24
```

carb	hp_cat	loghp	sum	loghp1	carnames	valves
4	high hp	2.04139268515822	273.90	2.041393	Mazda RX4	24
4	high hp	2.04139268515822	273.90	2.041393	Mazda RX4 Wag	24
1	low hp	missing	204.85	0.000000	Datsun 710	16
1	high hp	2.04139268515822	371.08	2.041393	Hornet 4 Drive	24
2	high hp	2.24303804868629	538.15	2.243038	Hornet Sportabout	32
1	high hp	2.02118929906994	332.76	2.021189	Valiant	24

4.14 Merge data frames (individuals)

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

Reference

4.15 Create a new factor from existing

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

4.16 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))
```

4.17 calculate dates and photoperiod

(using geosphere library)

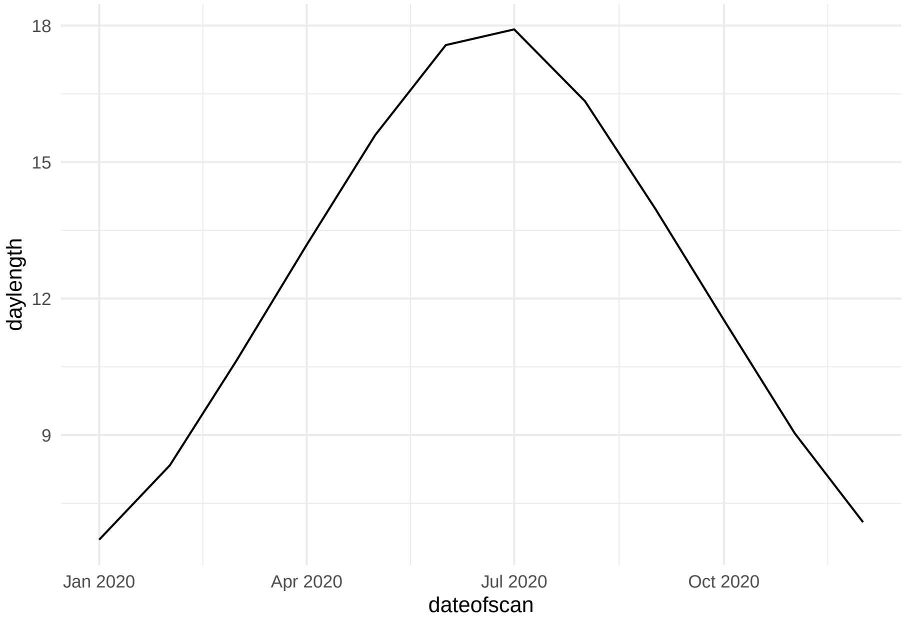
```
#import sample dataset
dateslat <- read_csv("dateslat.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   `ID's` = col_double(),
##   `date (dmy)` = col_character(),
##   latitude = col_double()
## )
```

```
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()
```



4.18 Reduce variables using PCA

4.18.1 To be completed

Chapter 5

Statistical Analysis

5.1 Regression

5.1.1 Linear regression on groups

```
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.

# 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

as.factor(gear)	mean	sd
3	17.692	1.349916
4	18.965	1.613880
5	15.640	1.130487

gear	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviar
3	0.66	0.63	28.87	25.19	0.00	1	-70.65	147.31	149.43	10837
4	0.10	0.01	25.72	1.15	0.31	1	-54.90	115.80	117.26	6616
5	0.88	0.83	41.95	21.03	0.02	1	-24.50	55.00	53.83	5279

5.2 Logistic Regression

5.2.1 to be completed

5.3 Survival Analysis and Visualisation

5.3.1 To be completed

5.4 Receiver Operated Curves (ROC)

5.4.1 To be completed

Chapter 6

Data Visualisation

Tables and graphs, survival plots, missing values. ## Summary Tables

6.0.1 Packages needed

6.0.2 Summarise by group

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

6.0.3 Summary Table - Multiple functions, variables

```
# make sure brackets are correct

df.sum <- mtcars %>%  select(mpg,cyl,hp) %>%
  summarise(across(everything(),list(mean=mean,sd=sd)))
kable(df.sum,digits=2) %>% kable_styling(full_width = FALSE) %>%
  kable_minimal() # perform the analysis
```

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

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

```r
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
```

6.1 Specific package for summary tables:

6.1.1 Arsenal package

```r
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	

6.1.2 Summary tools package

```r
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
```

	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

```
##       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),digit
```

```
## Warning in if (grepl(re1, str, perl = TRUE)) {: the condition has length > 1 and
## only the first element will be used

## Warning in if (grepl(re2, str, perl = TRUE)) {: the condition has length > 1 and
## only the first element will be used

## Warning in if (grepl(re3, str, perl = TRUE)) {: the condition has length > 1 and
## only the first element will be used

## Error in pryr::where(obj_name) : length(name) == 1 is not TRUE
```

6.2 Visual summary of data

Options are for markdown

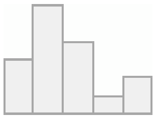
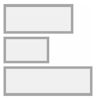
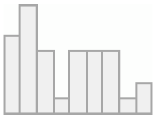
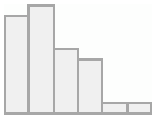
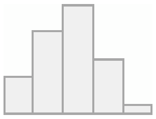
```
dfSummary(mtcars, plain.ascii = FALSE, style = "grid",
          graph.magnif = 0.5, valid.col = FALSE, tmp.img.dir = "/tmp")
```

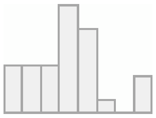
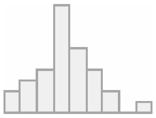
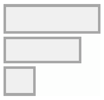
```
## temporary images written to 'C:\tmp'
```

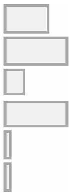
6.2.1 Data Frame Summary

6.2.1.1 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%)
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%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
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))
M <- corr1$r
M
```

```
##             mpg         hp        drat         wt        qsec
## mpg   1.0000000 -0.7761684  0.68117191 -0.8676594  0.41868403
## hp   -0.7761684  1.0000000 -0.44875912  0.6587479 -0.70822339
## drat  0.6811719 -0.4487591  1.00000000 -0.7124406  0.09120476
## wt   -0.8676594  0.6587479 -0.71244065  1.0000000 -0.17471588
## qsec  0.4186840 -0.7082234  0.09120476 -0.1747159  1.00000000
```

```
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 <- corr1$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

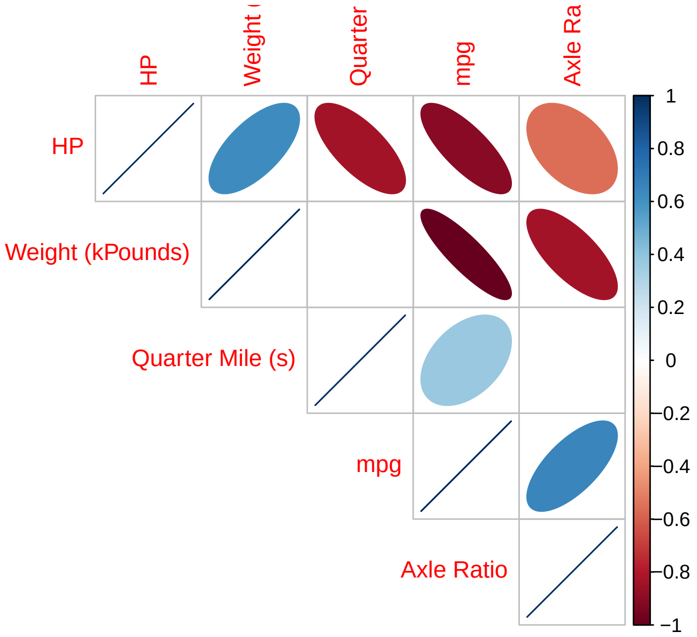


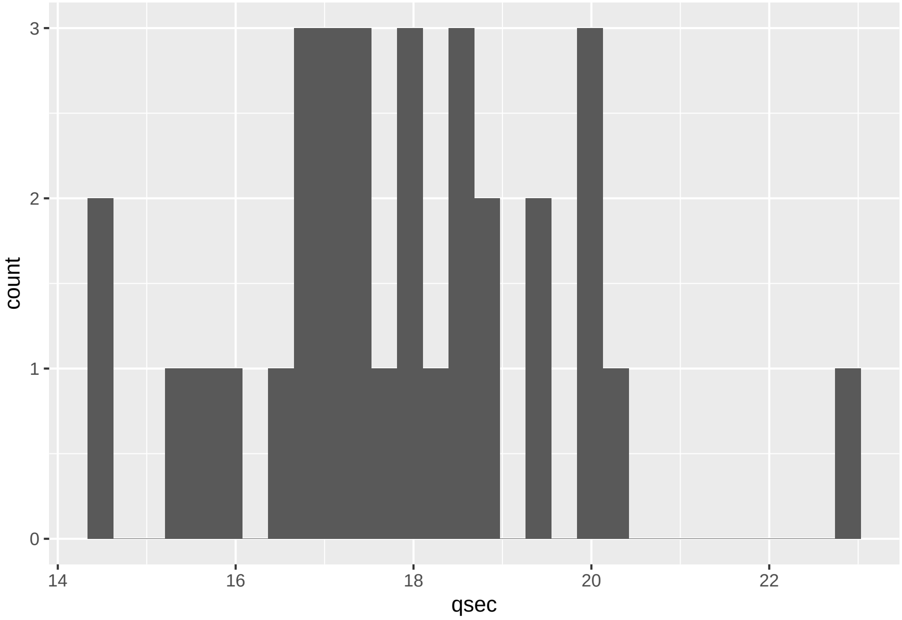
Figure 6.1: Correlation Plot

6.4 Graphing

6.4.1 Frequency Histogram - basic

```
plot1 <- mtcars %>% ggplot(aes(qsec))+geom_histogram()
plot1
```

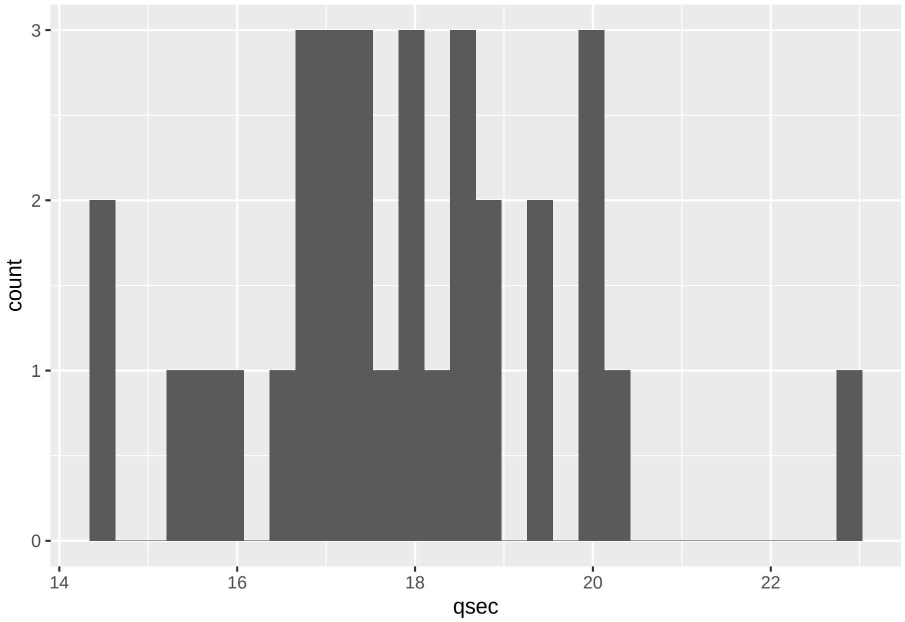
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



### Frequency Histogram + normal distribution

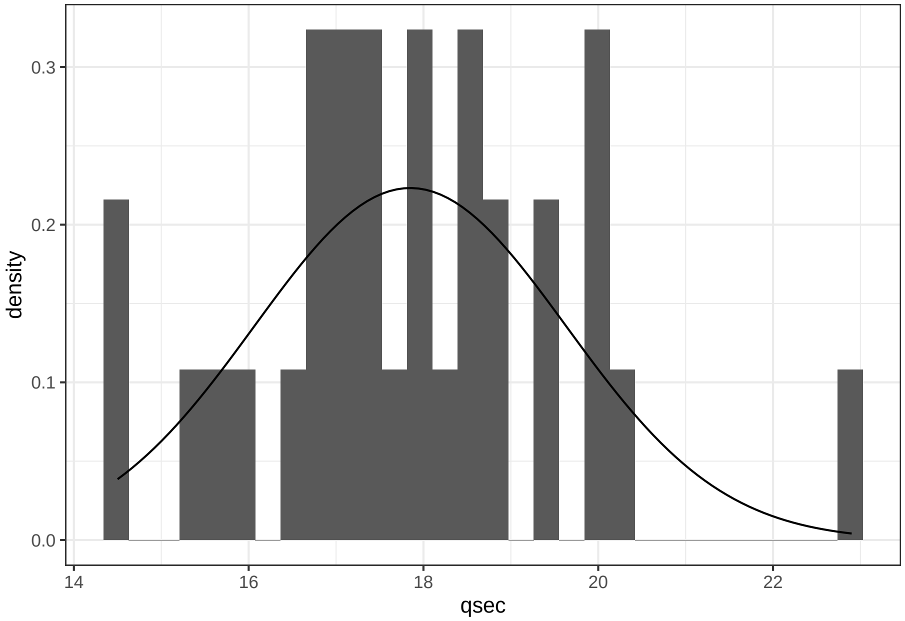
```
plot1 <- mtcars %>% ggplot(aes(qsec))
plot1+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# add normal plot
plot1 + geom_histogram(aes( y=..density..))+stat_function(fun = dnorm, args = list(mean =mean(mtc
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

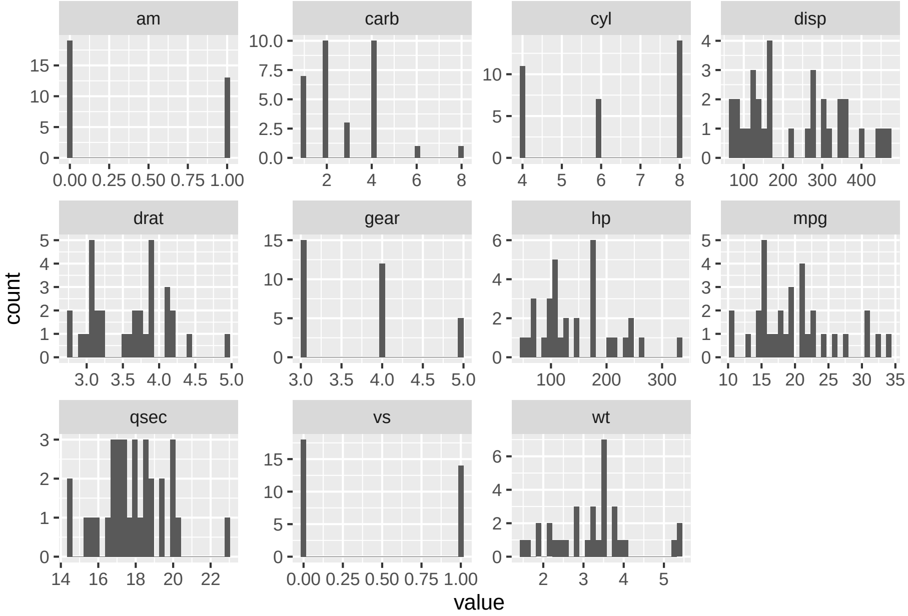


```
# ..density.. changes y axis to density, not count. stat function defines normal line
```

6.4.2 multiple plot of all distributions

```
mtcars %>% keep(is.numeric) %>% gather() %>% ggplot(aes(value)) + facet_wrap(~ key, sca
```

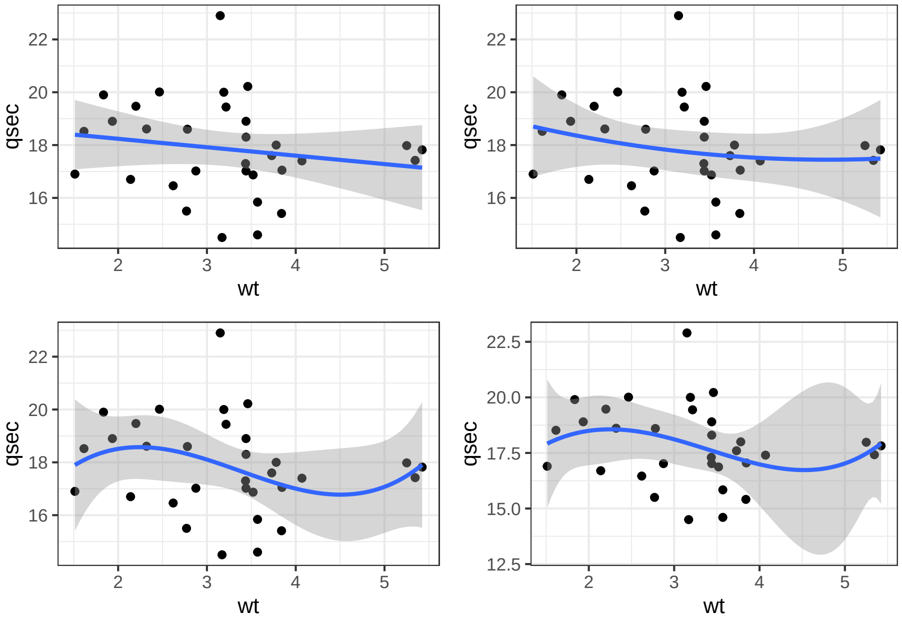
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



6.4.3 x*y scatterplot with linear regression 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
plot2c <- plot2 +geom_point()+stat_smooth(method='lm',formula = y ~ poly(x, 3)) + theme
plot2d <- plot2 +geom_point()+stat_smooth(method='lm',formula = y ~ poly(x, 4)) + theme

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

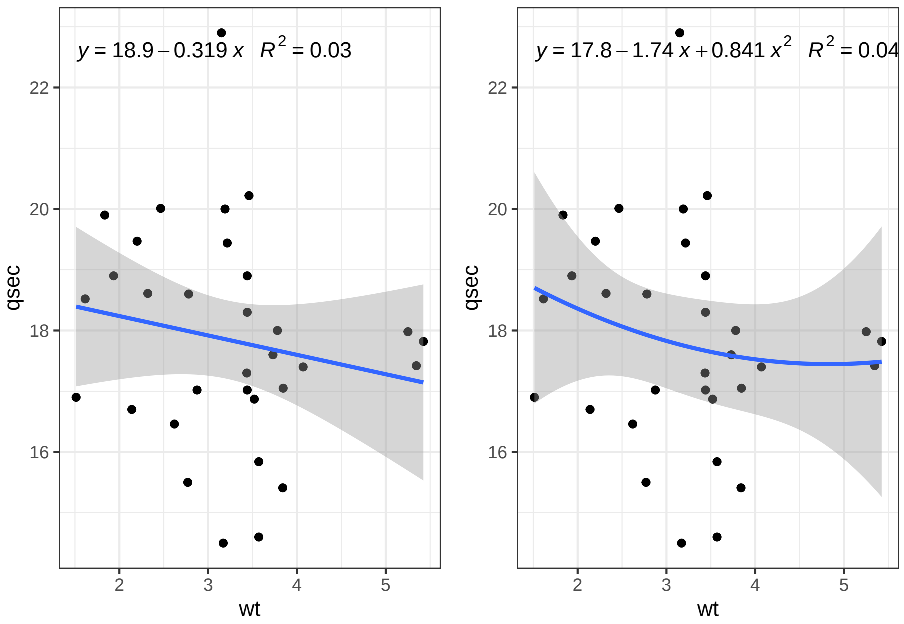


6.4.4 Add formula to plot.

```
my.formula <- y ~ x
a <- plot2 +geom_point()+geom_smooth(method='lm',formula=my.formula)+stat_poly_eq(formula = my.fo

my.formula2 <- y ~ poly(x, 2)
b <- plot2 +geom_point()+geom_smooth(method='lm',formula=my.formula2)+stat_poly_eq(formula = my.f

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


$y = 18.9 - 0.319\,x \quad R^2 = 0.03$
$y = 17.8 - 1.74\,x + 0.841\,x^2 \quad R^2 = 0.04$
qsec
22
20
18
16
wt
2
3
4
5