R CHEATSHEET

Packages

```
install.packages('tree')
installed.packages()
```

To use functions in your R code

```
library(tree)
```

Conditionals/Loops

```
if (x > 10) {
  print("x is greater than 10")
} else {
  print("x is less than or equal to 10")
}

repeat { if (x>10) break else
{ print(x); x <- x+1 } }
```

You must use a break to get out of a repeat

```
while (x<10) { print(x); x <- x+1 }

for (x in 1:10) print(x)
```

Charting

Output to PDF

```
pdf('some_file.pdf')
# charting
dev.off()
```

Scatterplot

```
win_or_draw <- league_table$home_wins +
league_table$home_draws
plot(win_or_draw)
text(win_or_draw, teams, pos=4)
```

Stacked bar chart

```
win_or_draw <-
league_table[c('home_wins',
'home_draws')]
data <- t(as.matrix(win_or_draw))
par(bg="gray")
barplot(data,  names.arg=league_table
$teams, legend=c('wins', 'draws'))
```

Using ggplot2

```
library("ggplot2")
```

Note: it comes with many data tables, such as “mpg”

Using qplot

```
qplot(displ, hwy, data=mpg)
qplot(displ, hwy, data=mpg,
shape=factor(year))
```

Building with layers

```
p <- ggplot()
p + geom_point(data=mpg, aes(displ, hwy,
shape=factor(year)))
```

Aesthetics

```
aes(x=displ, y=hwy, shape=factor(year))
```

Statistical transformations

```
p <- ggplot(movies)
p +
stat_bin(aes(year, ..count..),binwidth=1
)
```

Geometric objects

```
# geom_point x,y, color, size, shape
# geom_histogram default stat is
stat_bin
p <- ggplot(movies)
p + geom_histogram(aes(year))
```

Adjustments

```
p <- ggplot(mpg)
p + geom_point(aes(cty, hwy),
position="jitter")
```

https://github.com/narath/r_cheatsheet

Version 0.1

Data Structures

Vectors

```
vector <- c(1,2,3)
vector <- 1:10
vector <- seq(0,100,by=10)
```

Vector index starts a 1 !

```
# vector = [1] 0 25 50 75 100
vector[1] # 0
vector[2] # 25
vector[2:4] # 25 50 75
vector[c(2,4)] # 25 70
```

To select a *single* value from a vector, use [[]]

```
vector[[3]] # 50
```

Lists

```
peter <- list(name="Peter", age=30, glasses=TRUE)
peter[1] # $name [1] "Peter"
peter$name # Peter
peter[["name"]] # Peter
peter[["na"]] # NULL
peter[["na", exact=FALSE]] # "Peter"
```

Matrix

```
m <- matrix(1:20, nrow=5, ncol=4) # by default populates by column
m <- matrix(1:20, nrow=5, ncol=4, byrow=TRUE)
```

Selected elements in a matrix

```
m[n_th_item_in_running_sequence_COLUMN_wise]
m[row, colum]
m[row1:rowX, column1:columnX]
```

Giving column and row names

```
dimnames(m) <- list(vector_row_names, vector_col_names)
```

Factors

```
colors <- factor(c('green', 'red', 'blue'))
factor(c("poor","average","good"), order=TRUE, levels=results)
```

Data Frames

Remember they are constructed by columns

```
teams <- c("Manchester", "Totenham", "Arsenal")
home_wins <- c(14, 10, 10)
home_draws <- c(0, 1, 2)
league_table <- data.frame(teams, home_wins, home_draws)
league_table

league_table$team
```

To select a subset of columns

```
league_table[c("team","home_wins")]
```

To select elements from a column

```
league_table$team[league_table$home_wins>8]
league_table$team[league_table$away_wins > league_table$home_wins]
# because vector comparisons, compare element by element

with(league_table, team[away_wins > home_wins])
```

To add another column of data

```
points <- c(66, 61, 53)
points_table <- data.frame(teams, points)
league_table <- merge(league_table, points_table, by="teams")
```

To add another row of data

```
row <- list(teams="Norwich", home_wins=2, home_draws=1, points=10)
row_table <- data.frame(list)
league_table <- rbind(league_table, row_table)
```

To sort

```
with(league_table, league_table[order(-points), ])
```

Importing

Import from csv

```
epl <- read.csv('league_data.csv')
```

Import from DB

```
install.packages(c("DBI", "RMySQL"))
con <- dbConnect(MySQL(), host="localhost", dbname="epl",
user="root", password="pwd")
league <- dbGetQuery(con, "select * from league")
dbDisconnect(con)
```

Import from DB using a config file for credentials

Create $HOME/.my.cnf

```
[epl]
user=root
password=pwd
database=epl
host=localhost
```

Then use this to connect in R

```
con <- dbConnect(MySQL(), groups="epl")
```