

# MGLC Transferable Skills Workshop: R

Sahil Shah

sahil.shah@u.northwestern.edu

May 27, 2015

## 1 Introduction

## 2 UCI ML Repository: Wine Data Set

## 3 Introduction to R

## 4 Hands-On Practice

# Acknowledgments

- McCormick Graduate Leadership Council (MGLC)
- Eric Earley, Hayley Belli, Paula Straaton, Bruce Lindvall

# Schedule

- 1 Introduction & Lecture
- 2 Dinner
- 3 Hands-on Practice

# Icebreaker

Introduce yourself to your immediate neighbors and tell them:

- Your department and/or research interests
- Your favorite programming language
- Your level of experience with R

# Wine Data Set

The UCI ML Repository: Wine Data Set:

<https://archive.ics.uci.edu/ml/index.html>

- Chemical analysis of wine from three different cultivators
- 178 samples, 13 attributes (+ CLASS)

# Wine Data Set: V3 (Malic acid)

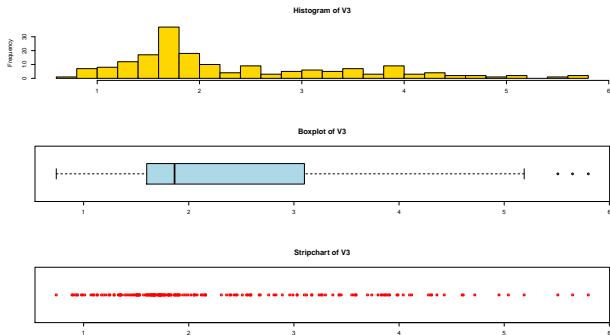
Outline

Introduction

Wine Data Set

Intro. to R

Hands-On  
Practice



# Wine Data Set: Scatterplot Matrix

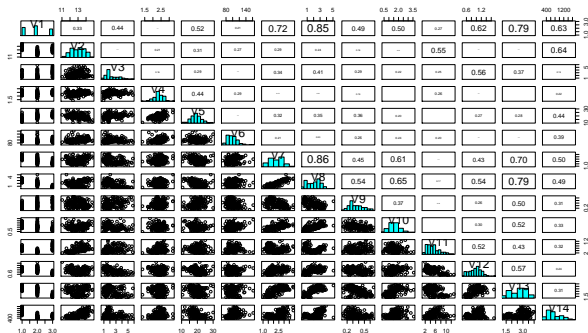
Outline

Introduction

Wine Data Set

Intro. to R

Hands-On  
Practice





# Wine Data Set: Total phenols vs Flavanoids

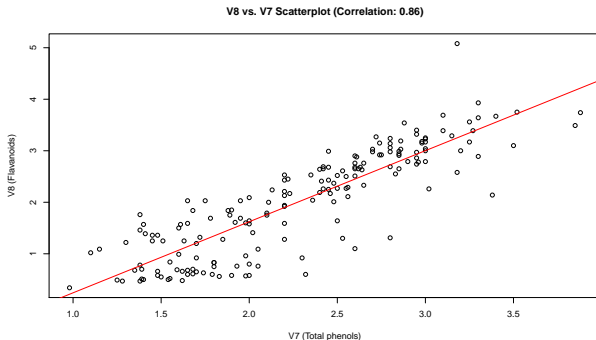
Outline

Introduction

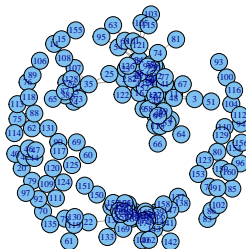
Wine Data Set

Intro. to R

Hands-On  
Practice



# Wine Data Set: Network



# Wine Data Set: Graph Components ( $N$ )

Outline

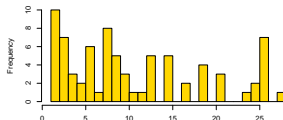
Introduction

Wine Data Set

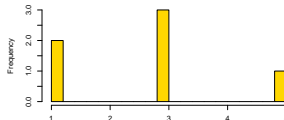
Intro. to R

Hands-On  
Practice

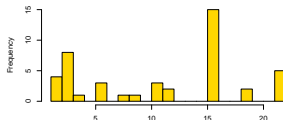
Degree Distribution (N=77)



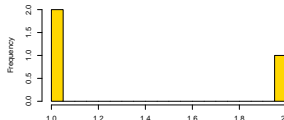
Degree Distribution (N=6)



Degree Distribution (N=45)



Degree Distribution (N=3)



# Packages

## Repositories

- CRAN (as of today) has **6690** packages for modeling, visualization, etc.
- Bioconductor (as of today) has **1024** packages for analysis of high-throughput genomic data

## Download and load packages with

- `install.packages()`
- `library()`

# Importing Data

Import data from a CSV file or install packages to **directly** import from Excel, Stata, etc.

- **CSV:** `read.table()` : NB. The default separator is a white space
- **Excel:** `read.xlsx()`
- **Stata:** `read.dta()`

# Sweave

Create dynamic reports (i.e. reproducible research) with Sweave

- Combine both LaTeX and R code in one document
- Part of every installation

`*.rnw`  $\rightarrow$  `*.tex`  $\rightarrow$  `*.pdf`

- Begin with a LaTeX document, but with the extensions `*.rnw`
- R commands start with `<<=>` and end with `@`
- Process the Sweave file in R with `Sweave()` and compile the resulting `*.tex` file

- R is installed on Quest
- Install packages by logging in and using `install.packages()`
- Submit a job by creating a submission script and \*.R script

## Working with R

- `?` : pop-out help for stated function
- `str()` : short description of any R data structure
- `getwd()`
- `setwd()`
- `ls()` : returns a vector of strings giving names of objects
- `rm()` : pass in either an object or (with the `list` argument) character vector naming the object

**How do you clear your workspace?**



# Data Manipulation: Data Types

Outline

Introduction

Wine Data Set

Intro. to R

Hands-On  
Practice

There are four types of data.

- 1 Logical
- 2 Integer
- 3 Double (aka Numeric)
- 4 Character

NB. Individual numbers or strings are actually vectors of length one

# Data Manipulation: Data Structures

## Homogenous data structures

- 1d: Atomic vector: `c()`
- 2d: Matrix: `matrix()`
- nd: Array

## Heterogeneous data structures

- 1d: List: `list()`
- 2d: Data frame: `data.frame()`

NB. Factors is a vector that can contain only predefined values  
(**cf.** Advanced R by Hadley Wickham)

# Data Manipulation: Lists

- Three properties: Type (e.g. Double), Length, Attributes (Arbitrary metadata)
- Attributes include: Names and Dimensions
- The elements of a list can have different types. NB. Lists can contain other lists.
- Under the hood, data frames are lists of equal length vectors
- Linear model objects (`lm()`) are lists

**If the columns of a data frame are of different types, can the elements of a column be of different types?**

# Data Manipulation: Subsetting

Outline

Introduction

Wine Data Set

Intro. to R

Hands-On  
Practice

With the subsetting operator `[]`, you can use

- Positive integers: return elements at specified positions
- Negative integers: omit elements at specified positions
- Logical vectors: select elements where the logical value is T
- Character vectors: return elements with matching names.

# Data Manipulation: Subsetting

Outline

Introduction

Wine Data Set

Intro. to R

Hands-On  
Practice

- Using `[` will always return a list, `[[` pulls out the elements of a list
- 'If list `x` is a train carrying objects, then `x[[5]]` is the object in car 5; `x[4:6]` is a train of cars 4-6'
- Include `drop=F` when subsetting matrices and data frames
- To modify selected values, values combine subsetting with `<-` (assignment)

**If `sample(x)` takes a sample from the elements of `x`, how can you use `sample()` to randomly permute the columns of a data frame?**

## 'Split-Apply-Combine'

R has standard control structures

- `for (var in seq){ expr }`
- `while (cond){ expr }`
- `if (cond) expr1 else expr2`

but it has built in 'functionals' that take in a function apply it to each component and then combine the results

- `lapply` applies a function to each element in a list and returns a list
- `lapply(x,function())`
- Other loop functionals use different types of input and output (**cf.** `dplyr`)

NB. You can loop over the elements, numeric indices, or the names.

# Functions

## Abstract code into small functions

```
myfunction <- function(arg1,arg2=3,...){  
  statements  
  return(object)}
```

- Arguments can have default values (e.g. arg2 above)
- It's not necessary a function return anything (e.g. plotting)
- Use the `source()` to load functions

## Plotting

R has strong graphical capabilities and can create for example

- Density plots: `hist()`
- Boxplots: `boxplot()`
- Scatter plots: `plot()`: Use `abline()` to add lines
- Combine plots using the `par` function and `mfrow` or `mfcol` arguments

The `lattice` and `ggplot2` are popular visualization packages  
(cf. Quick-R: Basic Graphs, Advanced Graphs)



- Use a consistent style
- Strive for names that are concise and meaningful
- Place spaces around all operators
- Use `<-` not `=` for assignment
- Comment with `#`
- Use commented lines to break up your file

# References and Resources

## References:

- <http://adv-r.had.co.nz/>
- <http://www.statmethods.net/>
- <http://www.statistik.tuwien.ac.at/public/filz/students/SweaveExa.pdf>
- <http://nicercode.github.io/>

## Resources:

- <http://google-styleguide.googlecode.com/svn/trunk/Rguide.xml>
- <http://stackoverflow.com/>
- <http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf>

## Hands-On Practice

With your neighbors,

- Write a function that returns the first  $n$  values of the Fibonacci sequence ( $F_1 = F_2 = 1, F_i = F_{i-1} + F_{i-2}$ )
- Import the Wine Data Set and create a histogram, boxplot, & stripchart of Malic acid. Combine the plots.
- Create a scatterplot matrix of the data set using `pairs()`. Read its documentation and put histograms on the diagonal and correlations on the upper triangle.
- Create a scatterplot of Total phenols vs Flavanoids. Calculate the Pearson correlation and include it in the title of the plot. Combine `abline` and `lm` to add a regression line on the plot. (cf. Quick-R)

## Hands-On Practice

With your neighbors,

- Calculate a  $178 \times 178$  **Spearman** correlation matrix of the wines. (exclude the CLASS variable (V1) in your calculation). Create an adjacency matrix by setting all entires of the correlation matrix  $\geq 0.99$  to 1 and all entries 0.99 to 0
- Install and load the `igraph` package. Use the above adjacency matrix to create an undirected, unweighted, graph without simple loops. Plot the network.
- Use `decompose.graph` and `lapply` to calculate the number of nodes and the degree distribution of each component (with at least three nodes ) of the above network. Plot the degree distribution of each of those components. Combine the plots. Include the number of nodes in the component in the title of the plot.