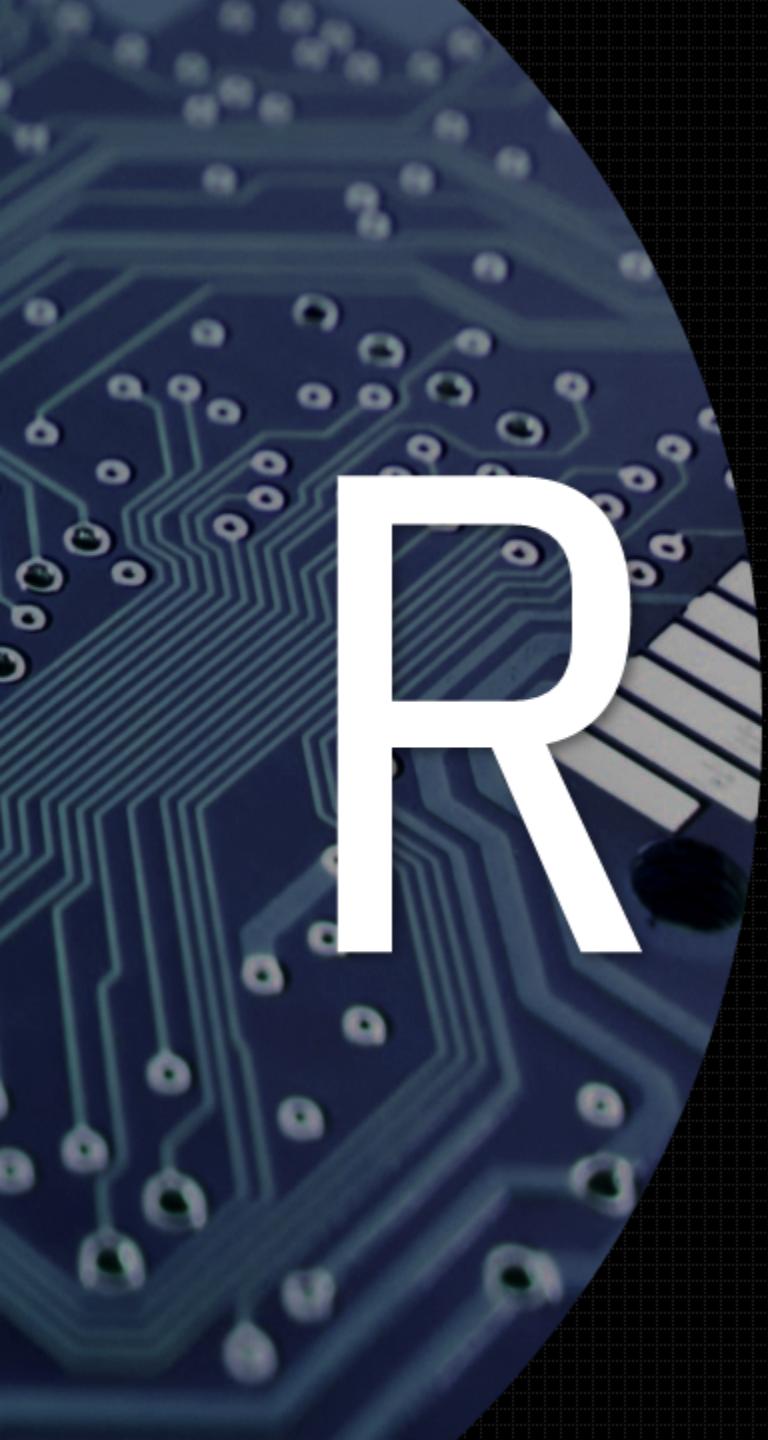


A brief introduction to

Losses Don
Member of team SC.

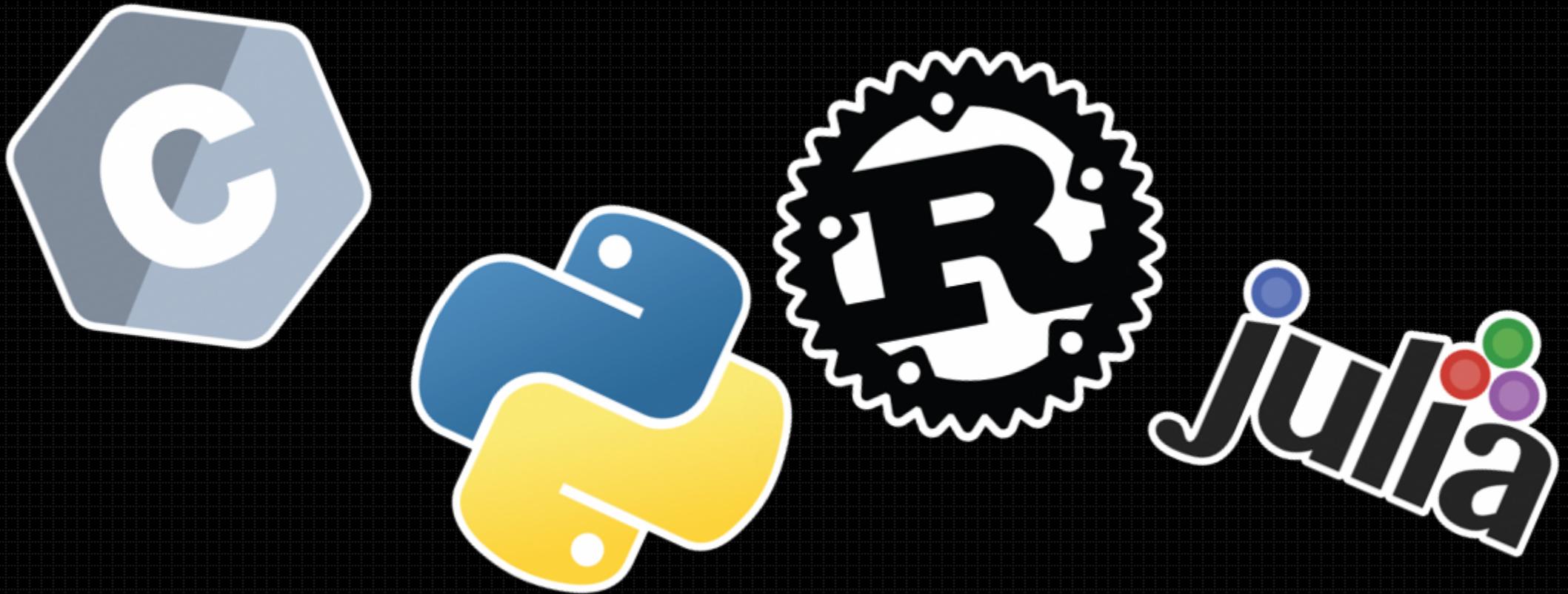
R

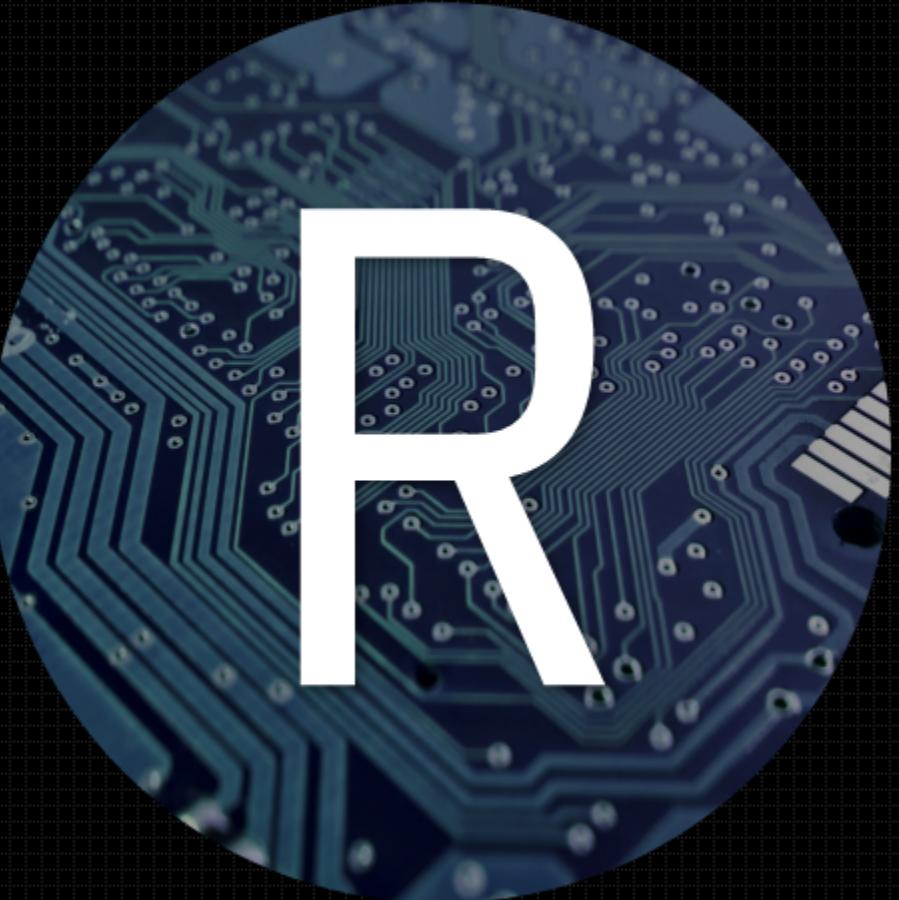




R

*is a programming language
for stat & analysis*



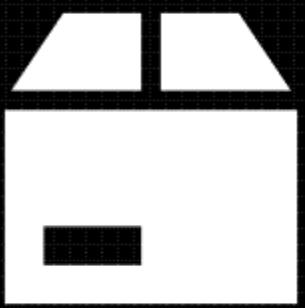




FREE



INTUITION



ECOSYSTEM



Say Hello to

R



C:/Users/Losses/OneDrive/work/Master/Homework/fNIRS/fNirs2018Spring/R - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Addins Go to file/function

signal.R 01-Data reading and single subject WT... NirsPlot.R readNirs.R wtc.R 002 - loop_S...

Source on Save Run Source

Environment History Connections Git List C

Global Environment

- nits.plo_ function (x, y, inten_)
- nits.plo_ function (nirs.data, _)
- read.nirs function (file_path)
- sample_r_ function (signal)
- scan_pai_ function (x)
- sunshine_ function (n)
- sunshine_ function (n)
- time_res_ function (signal)
- water_re_ function (n)
- water_sh_ function (n)

Files Plots Packages Help Viewer More C

New Folder Delete Rename More

Homework > Homework > fNIRS > fNirs2018Spring > R

Name Size

..

.git

.gitignore

.RData

.Rhistory

.Rproj.user

001 - loop_calc_wtc.R

002 - loop_stat_analysis.R

01-Data reading and single su

02-Non-parametric test.R

loop_calc_cortR

R.Rproj

R_modules

R_modules.zip

result

Plot Zoom

Signal

0 1000 2000 3000 4000

period

15.5

1.5

0.8

0.4

0.0

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64

1000 2000 3000 4000

index

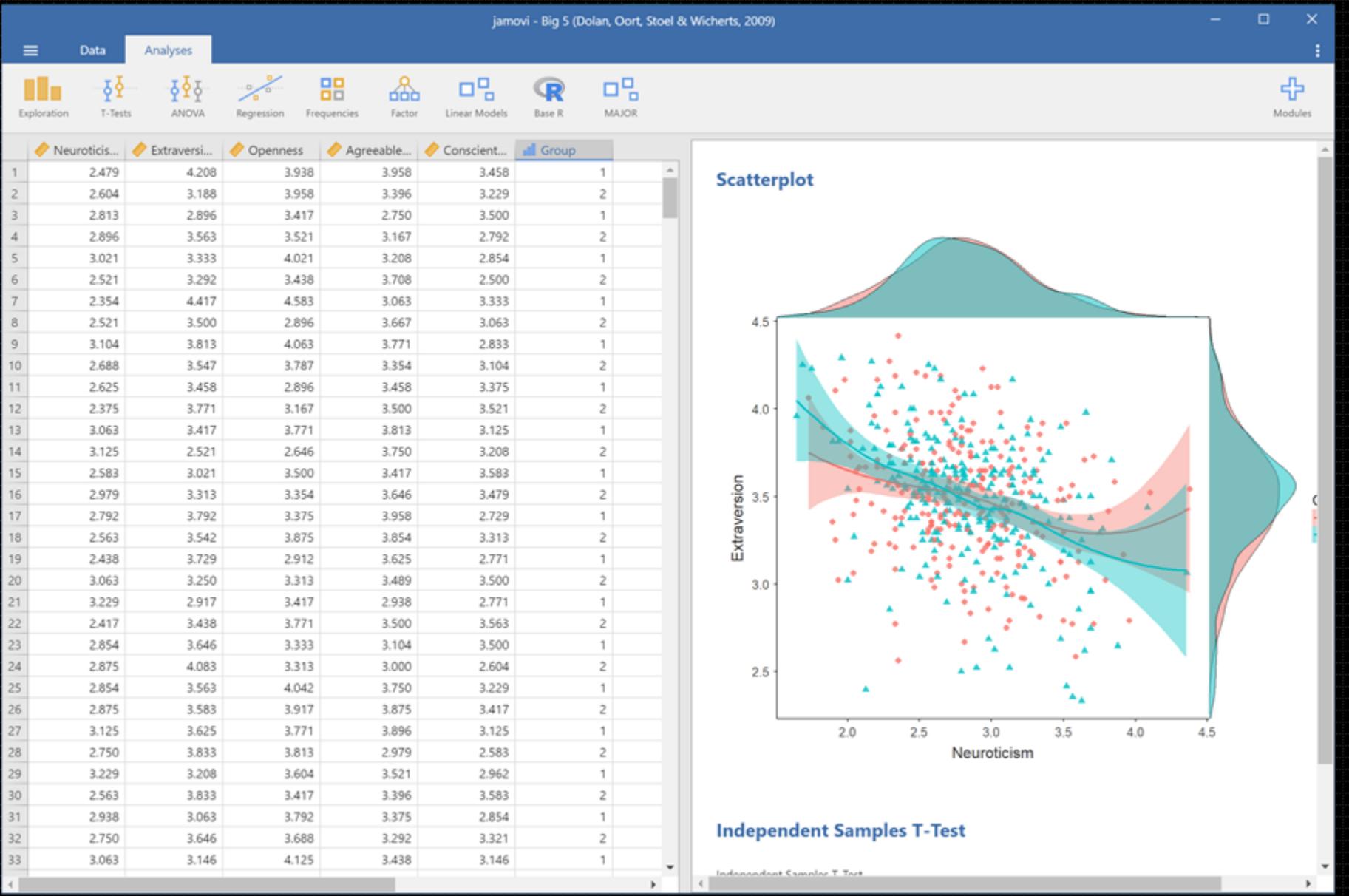
115:1 (Top Level) R Script

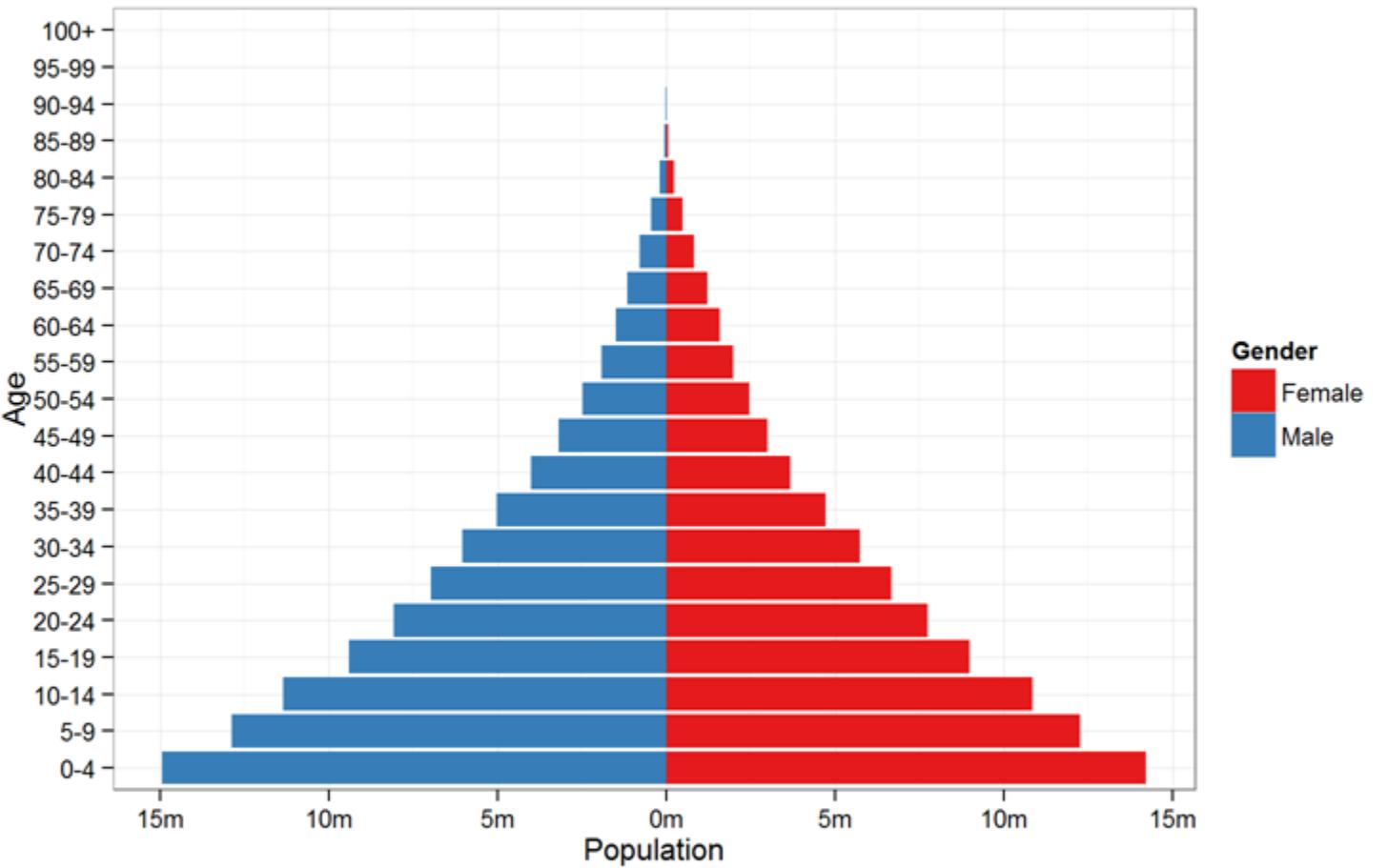
Console Terminal

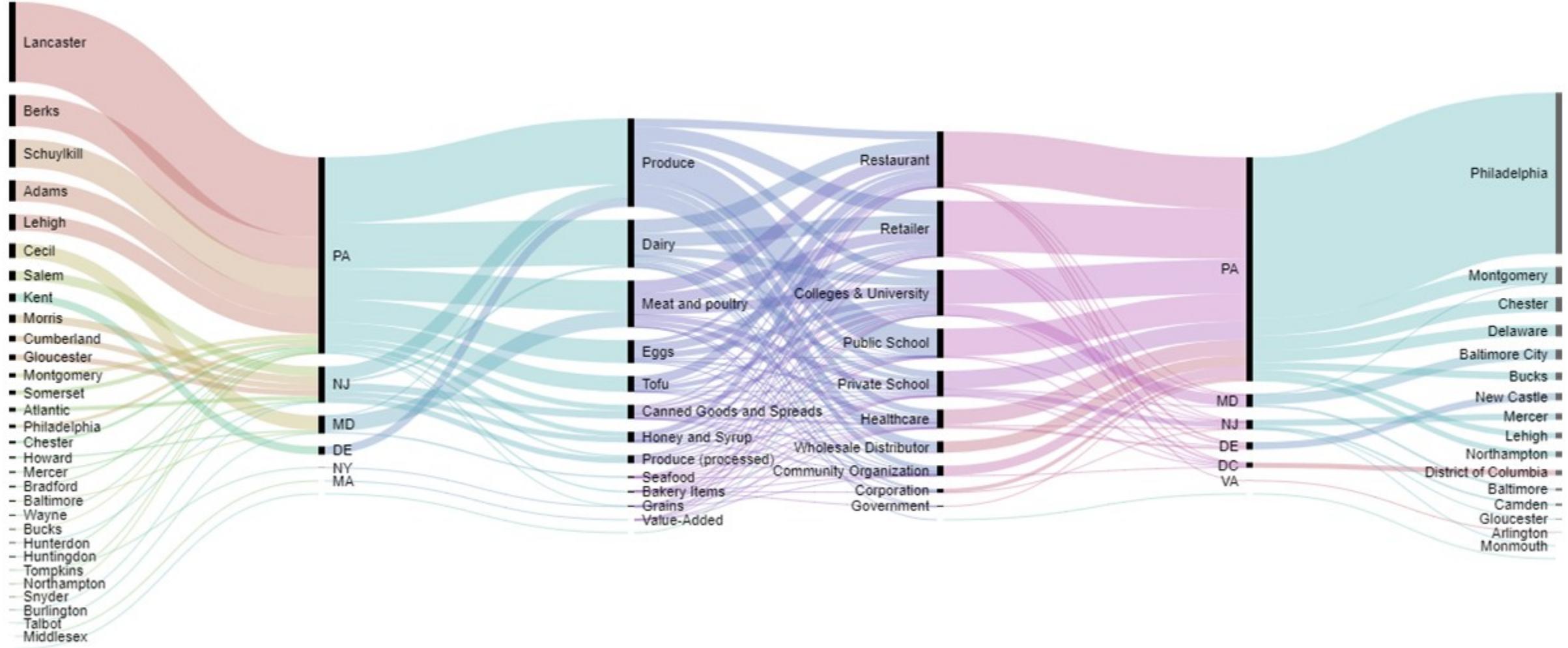
C:/Users/Losses/OneDrive/work/Master/Homework/fNIRS/fNirs2018Spring/R

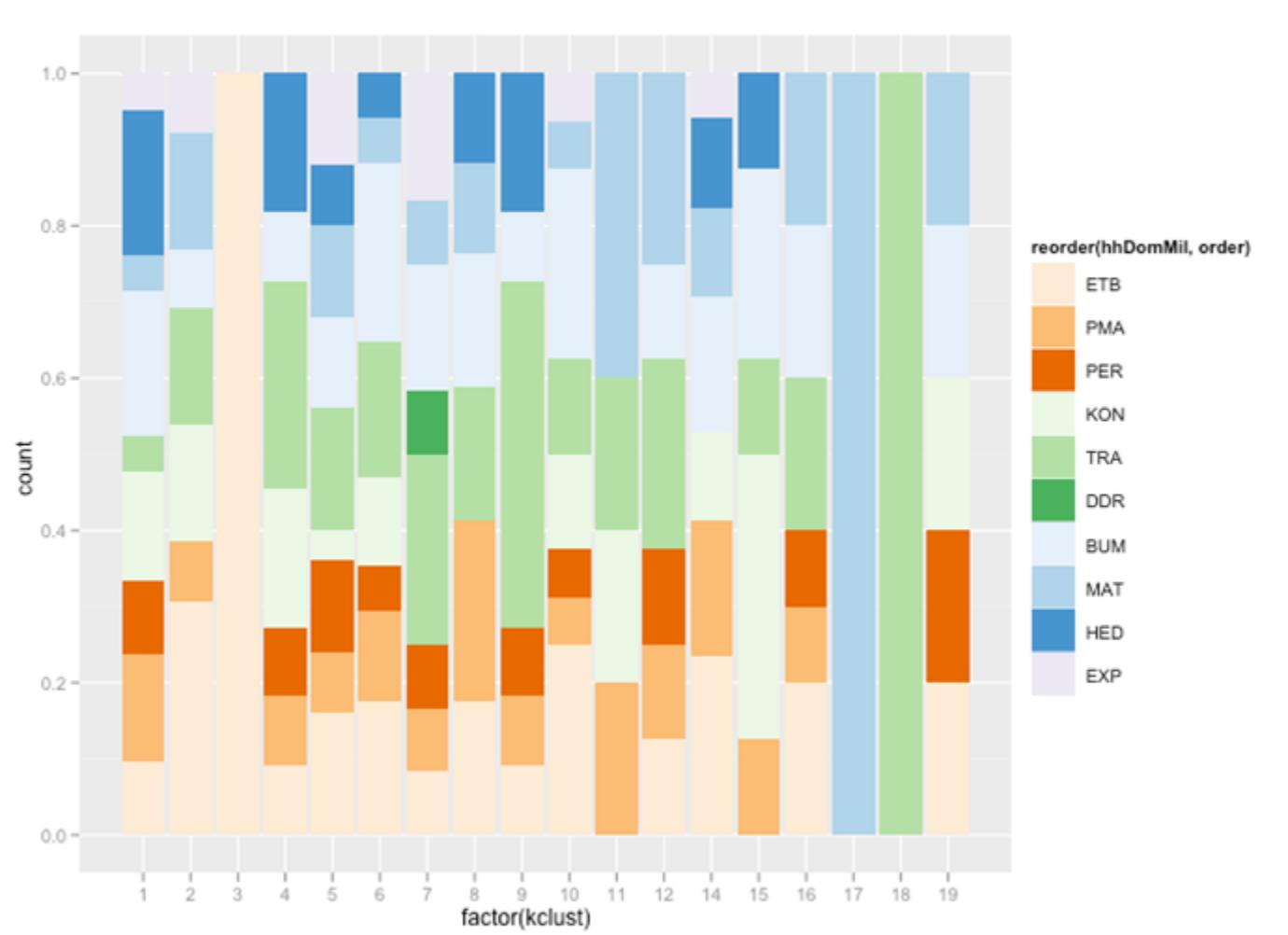
```
+ lines(x$original_data$B, type = 'S', col='#42B3D5')
+ legend(
+   'topright',
+   legend = c('A', 'B'),
+   col = c('#1A237E', '#42B3D5'),
+   lty = 1, cex = 0.6
+ )
+
+ par(fig = c(0, 1, 0, 0.72), new = TRUE)
+
+ wtc.image(x, color.palette = 'water_reverse_sheet(n.levels)')
+
> inspect_wtc(a)
> |
```

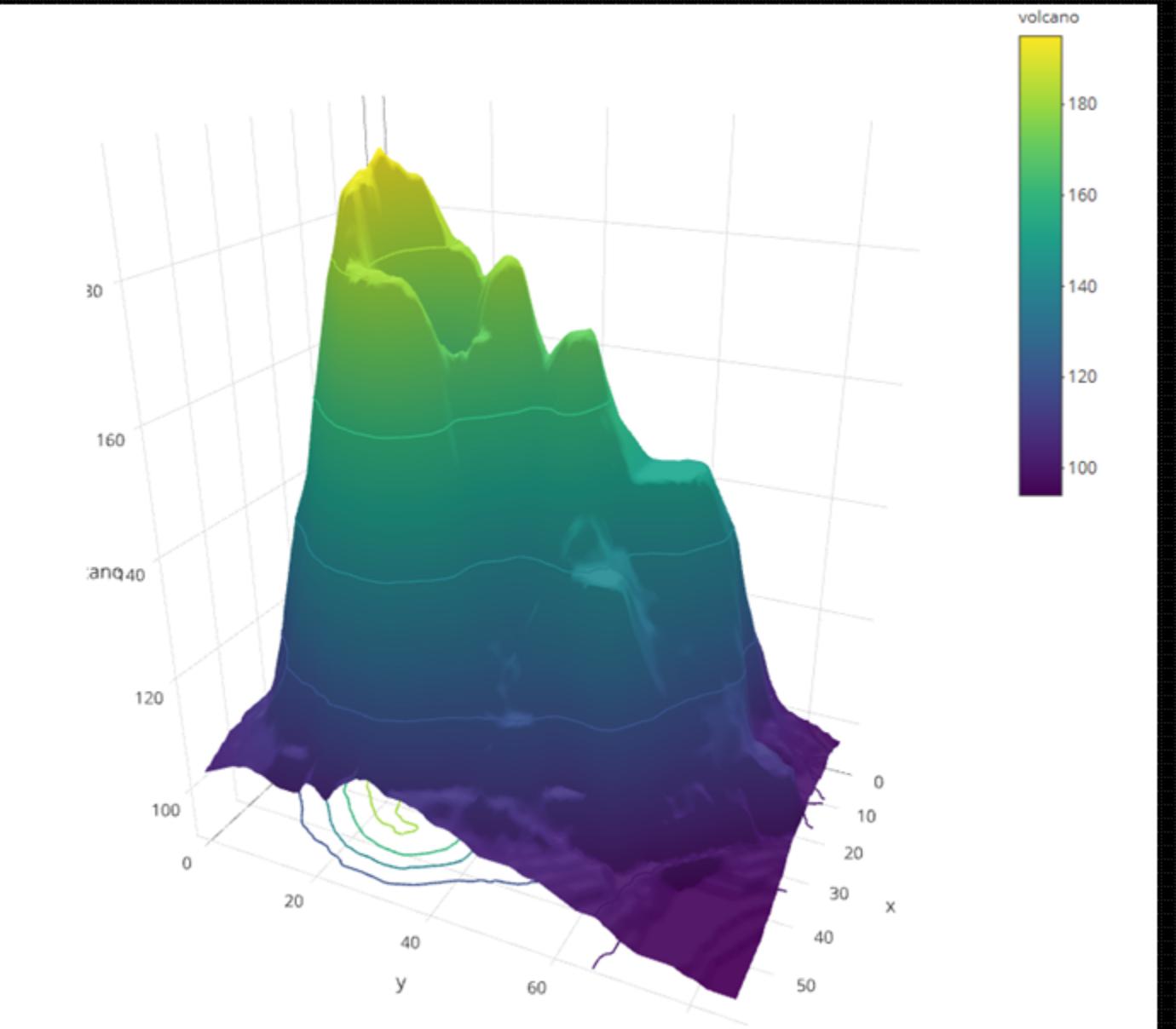


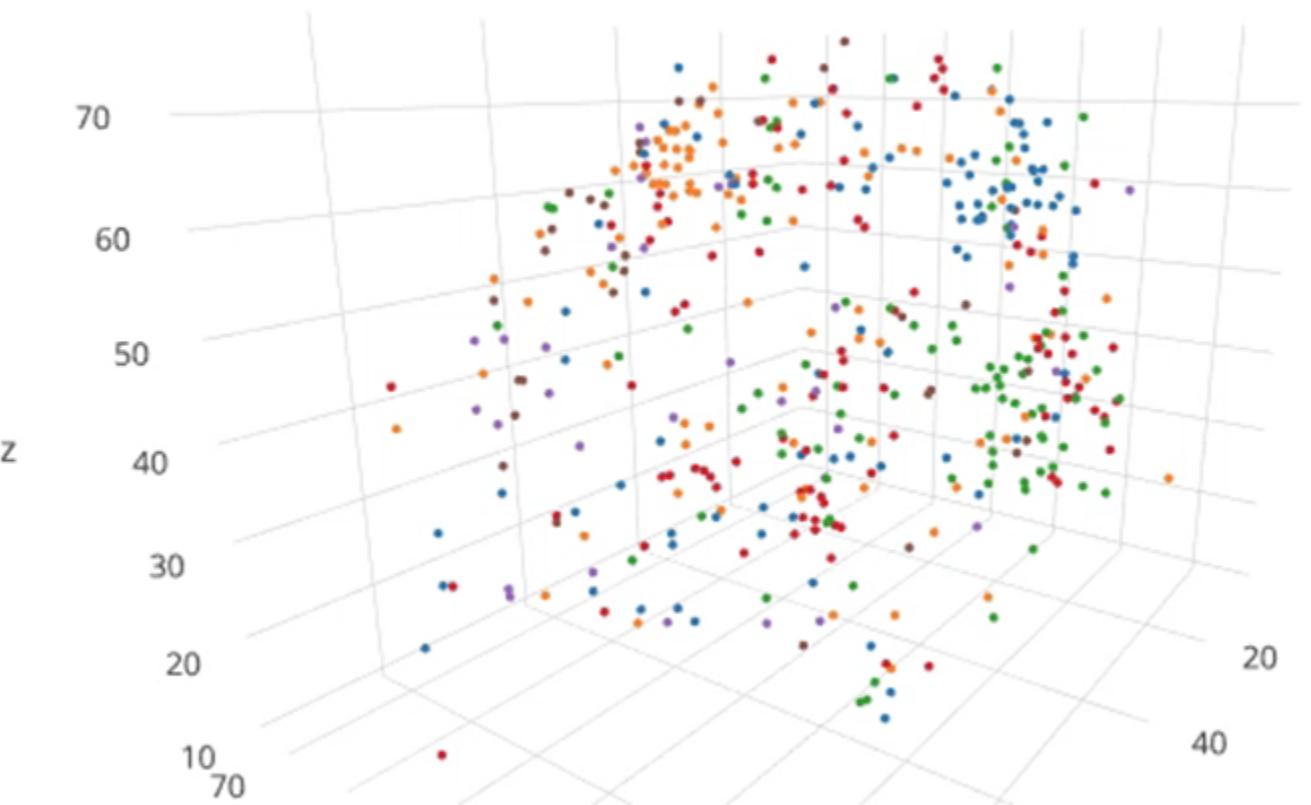




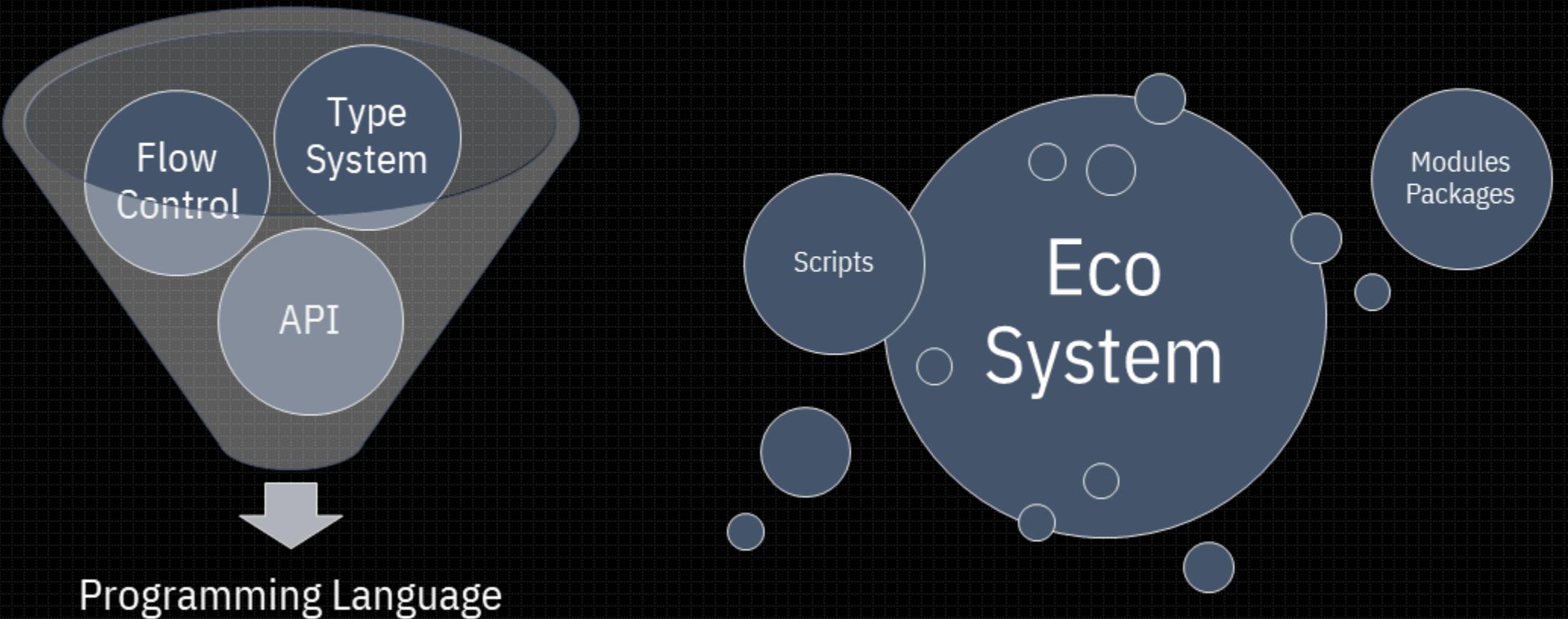








HOW TO LEARN A NEW PROGRAMMING LANG?



Type System

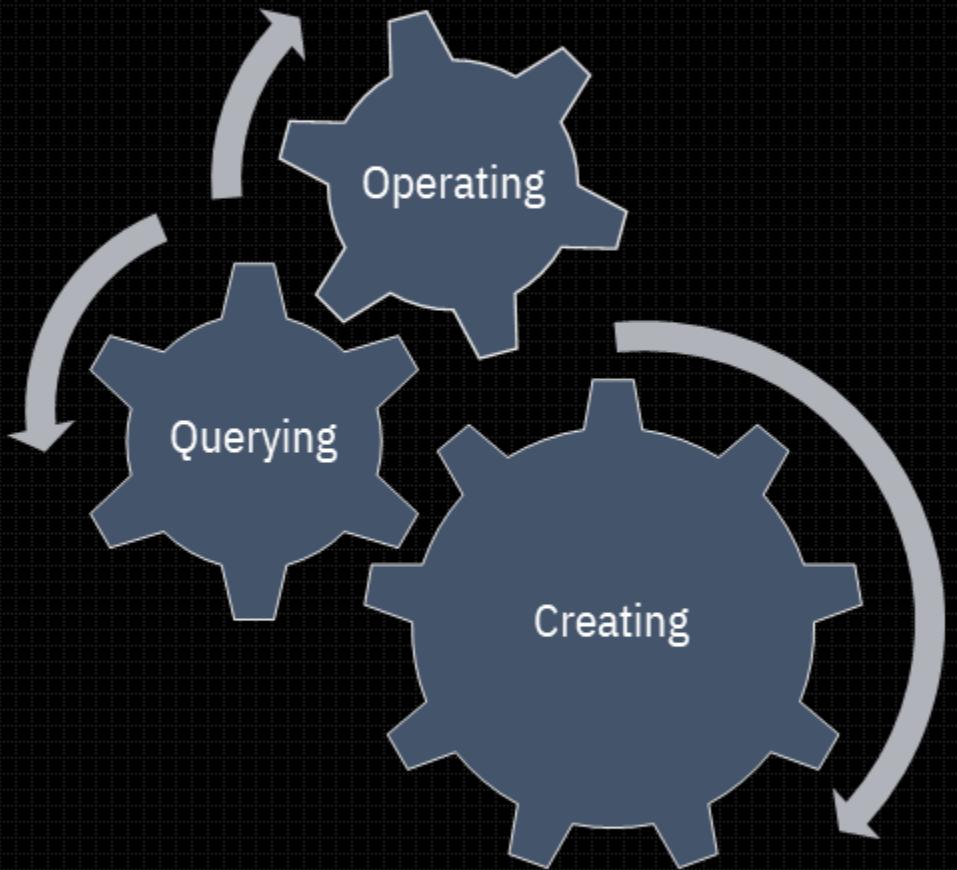


GIFSec.com

Vector, Special Vector

List, Special List

Vector, Special Vector
List, Special List



ATOMIC TYPES

VECTOR

```
> 1 # Numeric
```

```
> "a" # Character
```

```
> TRUE # Logical
```

```
> c(1, 2, 0, 4)
```

```
> c("apple", "brain")
```

```
> c(TRUE, FALSE, T, F)
```

VECTOR

CREATING VECTOR

```
> c("apple", "brain")  
[1] "a"
```

```
> 1:8  
[1] 1 2 3 4 5 6 7 8
```

```
> variable <- c('STG', 'TPJ', 'PCC')
```

QUERYING VECTOR

```
> variable[1]  
[1] "STG"
```

```
> variable[2:3]  
[1] "TPJ" "PCC"
```

```
> variable[c(T, F, T)]  
[1] "STG" "PCC"
```

VECTOR

OPERATING THE VECTORS

```
> 1 + 2  
[1] 3
```

```
> 2 / 0  
[1] Inf
```

```
> F + T + T  
[1] 2
```

BROADCASTING

```
> c(1, 2, 3) + 2  
[1] 3 4 5
```

```
> c(F, F, T) * 2  
[1] 0 0 2
```

```
> c(2, 3, 4) * c(1, 3, 5)  
[1] 2 9 20
```

FACTOR

```
> factor(c('a', 'c', 'd', 'a', 'd', 'b'))
```

```
[1] a c d a d b
```

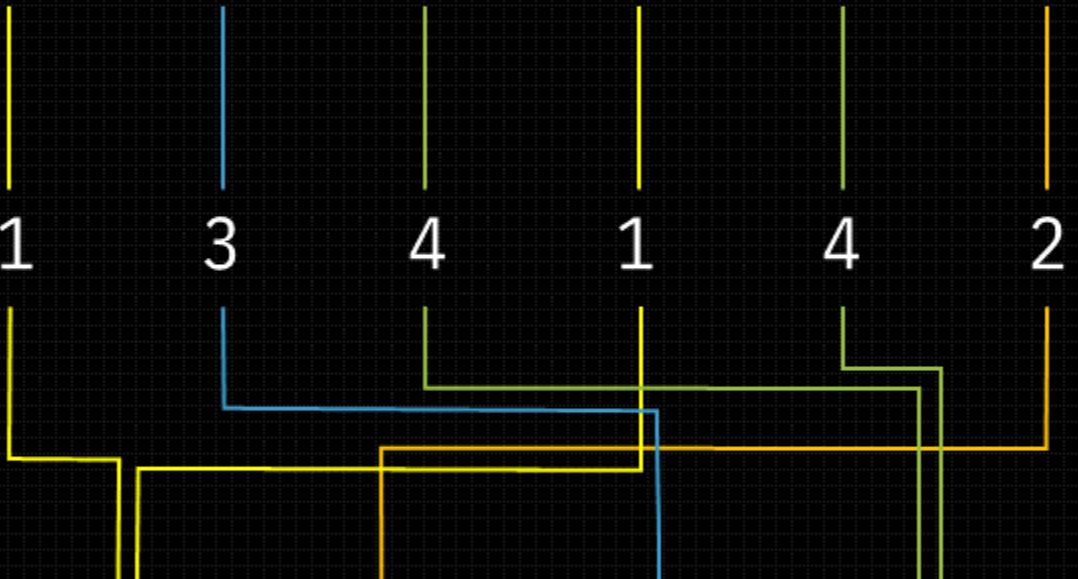
```
Levels: a b c d
```

'a', 'c', 'd', 'a', 'd', 'b'

Numeric Vector

1 3 4 1 4 2

Mapping a => 1 b => 2 c => 3 d => 4



FACTOR

```
> factor(c('a', 'c', 'd', 'a', 'd', 'b'))
```

```
[1] a c d a d b
```

```
Levels: a b c d
```

```
> as.numeric(factor(c('a', 'c', 'd', 'a', 'd', 'b')))
```

```
[1] 1 3 4 1 4 2
```

LIST

```
> c(2.2, 'Frontal Lobe', T, 1)
[1] "2.2"    "Frontal Lobe"    "TRUE"    "1"
```

ALL ATOMIC IN THE SPECIFIC VECTOR MUST BE CONSISTENT.

LIST

```
> list(2.2, 'Frontal Lobe', T, 1)
```

```
[[1]]
```

```
[1] 2.2
```

```
[[2]]
```

```
[1] "Frontal Lobe"
```

```
[[3]]
```

```
[1] TRUE
```

```
[[4]]
```

```
[1] 1
```

| <i>Name/Index</i> | |
|-------------------|--------------------|
| <i>Value</i> | |
| | [[1]] |
| | [1] 2.2 |
| | [[2]] |
| | [1] "Frontal Lobe" |

LIST

CREATING LIST

```
> list(T, "Insula")
```

```
[[1]]
```

```
[1] TRUE
```

```
[[2]]
```

```
[1] "Insula"
```

```
> list(broken = T, area = "Insula")
```

```
$`broken`
```

```
[1] TRUE
```

```
$area
```

```
[1] "Insula"
```

LIST

QUERYING LIST

```
> var <- list(broken = T, area = "Insula")
```

```
> var
```

```
$`broken`
```

```
[1] TRUE
```

```
$area
```

```
[1] "Insula"
```

```
> var[1]
```

```
$`broken`  
[1] TRUE
```

```
> var[1:2]
```

```
$`broken`  
[1] TRUE
```

```
$area
```

```
[1] "Insula"
```

LIST

QUERYING LIST

```
> var <- list(broken = T, area = "Insula")
```

```
> var
```

```
$`broken`
```

```
[1] TRUE
```

```
$area
```

```
[1] "Insula"
```

```
> var[c('area', 'broken')]
```

```
$`area`
```

```
[1] "Insula"
```

```
$broken
```

```
[1] TRUE
```

LIST

QUERYING LIST

```
> var <- list(broken = T, area = "Insula")          > var[['area']]  
[1] "Insula"  
  
> var  
$`broken`  
[1] TRUE  
  
$area  
[1] "Insula"
```

LIST

SINGLE BRACKET FOR SUBLIST

```
> var[c('area', 'broken')]  
$`area`  
[1] "Insula"  
  
$broken  
[1] TRUE
```

DOUBLE BRACKETS FOR CONTENT

```
> var[['area']]  
[1] "Insula"  
  
> var$area  
[1] "Insula"
```

DATA FRAME

| Case id | Bleeding | Surgery |
|---------|----------|---------|
| 10125 | Yes | No |
| 12357 | Yes | No |
| 17795 | No | Yes |

```
> research_cases <- data.frame(  
+   case_id = c(10125, 12357, 17795),  
+   bleeding = c(T, T, F),  
+   surgery = c(F, F, T)  
+ )
```

```
> research_cases  
case_id bleeding surgery  
1 10125 TRUE FALSE  
2 12357 TRUE FALSE  
3 17795 FALSE TRUE
```

DATA FRAME

```
> research_cases <- data.frame(  
+   case_id = c(10125, 12357, 17795),  
+   bleeding = c(T, T),  
+   surgery = c(F, F)  
+ )
```

- The length of each column must be consistent;

Error in data.frame(case_... :
arguments imply differing number of
rows: 3, 2

DATA FRAME

```
> research_cases <- data.frame(  
+   case_id = c(10, 12, 17, 19),  
+   bleeding = c(T, F),  
+   surgery = c(F, F)  
+ )
```

```
> research_cases  
case_id bleeding surgery  
1    10     TRUE FALSE  
2    12    FALSE FALSE  
3    17     TRUE FALSE  
4    19    FALSE FALSE
```

- The length of each column must be consistent;

DATA FRAME

```
> research_cases <- data.frame(  
+   case_id = c(10, 12, 17, 19),  
+   bleeding = c(T, F),  
+   surgery = c(F, F)  
+ )
```

```
> research_cases  
case_id bleeding surgery  
1    10     TRUE    FALSE  
2    12    FALSE    FALSE  
3    17     TRUE    FALSE  
4    19    FALSE    FALSE
```

- The length of each column must be consistent;
- If not consistent, R will try to broadcast the data;

DATA FRAME

```
as.list(research_cases)
```

```
$`case_id`
```

```
[1] 10 12 17 19
```

```
$bleeding
```

```
[1] TRUE FALSE TRUE FALSE
```

```
$surgery
```

```
[1] FALSE FALSE FALSE FALSE
```

- The length of each column must be consistent;
- If not consistent, R will try to broadcast the data;
- The essence of `data.frame` is a special kind of *list*;

DATA FRAME

```
> research_cases <- data.frame(  
+   case_id = c(10, 12, 17, 19),  
+   bleeding = c(T, F),  
+   surgery = c(F, F),  
+   name = c('ZhiHong', 'BinXing', 'Mei',  
'QingFeng')  
+ )
```

```
research_cases$name  
[1] ZhiHong BinXing Mei    QingFeng  
Levels: BinXing Mei QingFeng ZhiHong
```

- The length of each column must be consistent;
- If not consistent, R will try to broadcast the data;
- The essence of data.frame is a special kind of **list**;
- The **character** vector will be transformed to **factor** vector while creating the data.frame.

DATA FRAME

TRANSFORM FACTOR COLUMN IN DATA.FRAME

```
> research_cases$name <- as.character(research_cases$name)  
[1] 10 12 17 19
```

```
> research_cases$name  
[1] "ZhiHong" "BinXing" "Mei"    "QingFeng"
```

```
> class(research_cases$name)  
[1] "character"
```

DATA FRAME

QUERYING DATA FRAME

```
> research_cases[[1]]  
[1] 10 12 17 19
```

```
> research_cases[['case_id']]  
[1] 10 12 17 19
```

```
> research_cases$case_id  
[1] 10 12 17 19
```

DATA FRAME

QUERYING DATA FRAME

```
> research_cases[1]          > research_cases[1:2]
  case_id
  1  10
  2  12
  3  17
  4  19
                                case_id bleeding
                                1    10   TRUE
                                2    12  FALSE
                                3    17   TRUE
                                4    19  FALSE
```

DATA FRAME

QUERYING DATA FRAME

```
> research_cases[1]          > research_cases[1, 1:2]
  case_id                    case_id bleeding
  1   10                      1   10   TRUE
  2   12                      > research_cases[2:3, 1:2]
  3   17                      case_id bleeding
  4   19                      2   12   FALSE
                                3   17   TRUE
```

DATA FRAME

```
# QUERYING DATA FRAME
```

```
> research_cases
```

| case_id | bleeding | surgery |
|---------|----------|---------|
|---------|----------|---------|

| | | |
|---|----|------------|
| 1 | 10 | TRUE FALSE |
|---|----|------------|

| | | |
|---|----|-------------|
| 2 | 12 | FALSE FALSE |
|---|----|-------------|

| | | |
|---|----|------------|
| 3 | 17 | TRUE FALSE |
|---|----|------------|

| | | |
|---|----|-------------|
| 4 | 19 | FALSE FALSE |
|---|----|-------------|

```
> research_cases[4, 3]
```

```
[1] FALSE
```

```
> research_cases[1, 2]
```

```
[1] TRUE
```

```
> research_cases[1:2, 'bleeding']
```

```
[1] TRUE FALSE
```

DATA FRAME

OPERATING THE DATA.FRAME

```
> research_cases_1 <- data.frame(  
+   case_id = c(30, 35),  
+   bleeding = c(T, T),  
+   surgery = c(T, F),  
+   name = c('Hao', 'Si')  
)  
  
> research_cases_2 <- data.frame(  
+   case_id = c(10, 12),  
+   bleeding = c(T, T),  
+   surgery = c(T, F),  
+   name = c('Yang', 'Na')  
)
```

```
> rbind(  
+   research_cases_1,  
+   research_cases_2  
+ )
```

| | case_id | bleeding | surgery | name |
|---|---------|----------|---------|------|
| 1 | 30 | TRUE | TRUE | Hao |
| 2 | 35 | TRUE | FALSE | Si |
| 3 | 10 | TRUE | TRUE | Yang |
| 4 | 12 | TRUE | FALSE | Na |

DATA FRAME

```
# OPERATING THE DATA.FRAME
```

```
> research_cases_1 <- data.frame(  
+   case_id = c(10, 12, 16),  
+   name = c('Yang', 'Na', 'DanDan')  
)
```

```
> research_cases_2 <- data.frame(  
+   surgery = c(F, F, T),  
+   bleeding = c(T, T, F)  
)
```

```
> cbind(  
+   research_cases_1,  
+   research_cases_2  
)
```

| | case_id | name | surgery | bleeding |
|---|---------|--------|---------|----------|
| 1 | 10 | Yang | FALSE | TRUE |
| 2 | 12 | Na | FALSE | TRUE |
| 3 | 16 | DanDan | TRUE | FALSE |

DATA FRAME

OPERATING THE DATA.FRAME

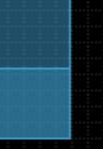
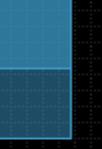
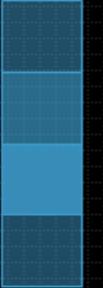
```
> research_cases_names <- data.frame(  
+   case_id = c(15, 16, 10, 12),  
+   name = c('MuDan', 'DanDan', 'Yang', 'Na')  
+ )  
> research_cases_detail <- data.frame(  
+   case_id = c(10, 12, 15, 16),  
+   surgery = c(F, F, F, T),  
+   bleeding = c(T, T, F, F)  
+ )
```

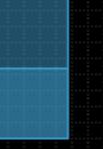
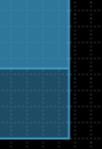
```
> merge(  
+   research_cases_names,  
+   research_cases_detail,  
+   by = 'case_id'  
+ )
```

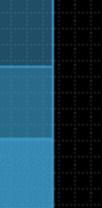
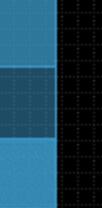
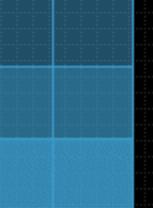
| | case_id | name | surgery | bleeding |
|---|---------|--------|---------|----------|
| 1 | 10 | Yang | FALSE | TRUE |
| 2 | 12 | Na | FALSE | TRUE |
| 3 | 15 | MuDan | FALSE | FALSE |
| 4 | 16 | DanDan | TRUE | FALSE |

DATA FRAME

OPERATING THE DATA.FRAME

`rbind( , ) => `

`cbind( , ) => `

`merge( , ) => `

ADVANCED DATA OPERATION



tidyR

TYPE SYSTEM OF R

| | VECTOR | LIST |
|-----------------|---------------|-------------|
| <i>GENERAL</i> | VECTOR | LIST |
| <i>COMPOUND</i> | FACTOR | DATA.FRAME |

Package System

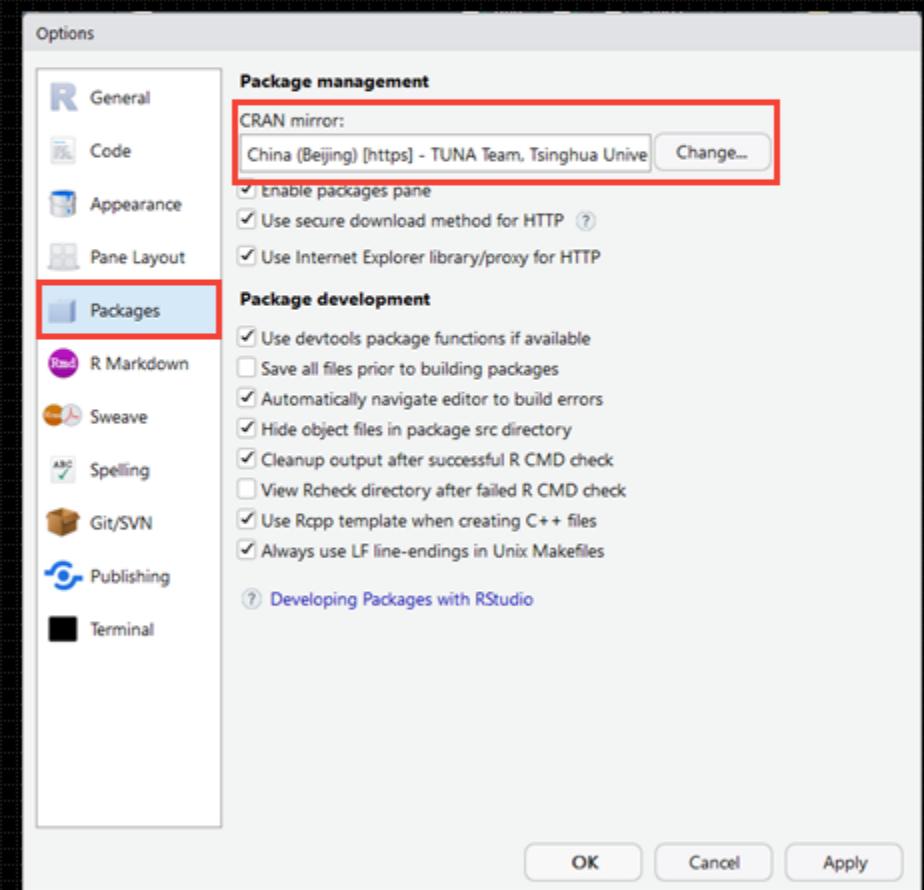
INSTALLING AND LOADING

INSTALLING

```
> install.packages(c("tidyverse", "WaveletComp"))
```

Installing packages into ‘~/Documents/R/win-library/3.5’

(as ‘lib’ is unspecified)



INSTALLING AND LOADING

LOADING

```
> library(tidyverse)
```

```
-- Attaching packages ----- tidyverse 1.2.1 --
```

```
v ggplot2 3.0.0   v purrr  0.2.5
```

```
v tibble  1.4.2   v dplyr   0.7.6
```

```
v tidyr   0.8.1   v stringr 1.3.1
```

```
v readr   1.1.1   vforcats 0.3.0
```

```
-- Conflicts ----- tidyverse_conflicts() --
```

```
x dplyr::filter() masks stats::filter()
```

```
x dplyr::lag()  masks stats::lag()
```

LOADING CUSTOM SCRIPT

SOURCE THE SCRIPT

```
> source('./R_modules/signal.R')
```

Attaching package: ‘signal’

The following object is masked from ‘package:dplyr’:

filter

The following objects are masked from ‘package:stats’:

filter, poly

Flow Control

CONDITION AND LOOP

CONDITION BLOCK

```
if( CONDITION ) {  
    CODE BLOCK 1  
} else {  
    CODE BLOCK 2  
}
```

LOOP BLOCK

```
for ( i in 1:5 ) {  
    CODE BLOCK 3  
}
```

CONDITION AND LOOP

CONDITION BLOCK

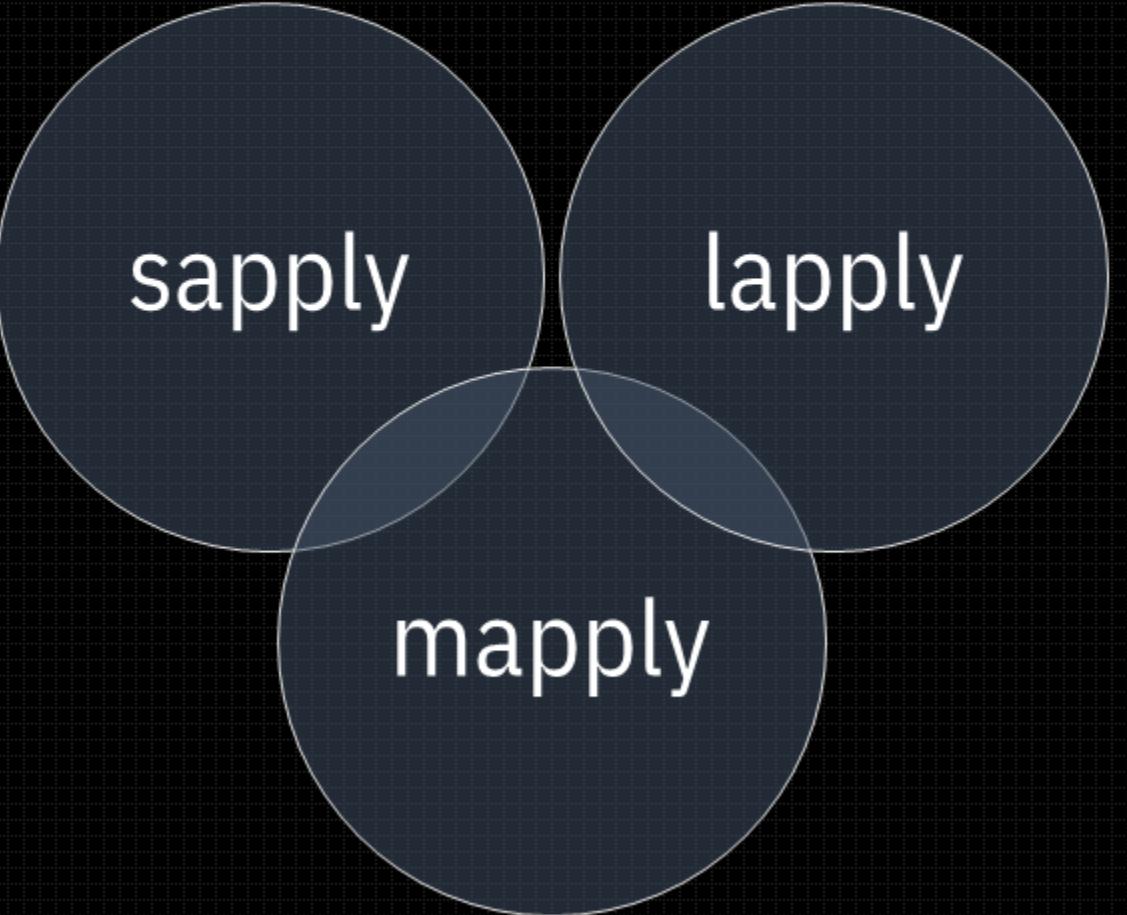
```
if( CONDITION ) {  
    CODE BLOCK 1  
} else {  
    CODE BLOCK 2  
}
```

LOOP BLOCK



```
r( 1:5 )  
CODE 10 .3  
}
```

ADVNCED FUNCTIONAL LOOP



sapply

lapply

mapply

DEFINE A FUNCTION

```
> say_hello <- function (x){  
+   paste0('Hello, ', x)  
+ }
```

```
> say_hello('Alice')
```

```
[1] "Hello, Alice"
```

LAPPLY

```
> names <- c('Hao', 'Si', 'Yang', 'Na')  
[1]  
[[1]]$`nchar`  
[1] 3  
  
> count_chars <- function (x){  
+   list(nchar = nchar(x), name = x)  
+ }  
[1] "Hao"  
  
> lapply(names, count_chars)  
[[1]]  
[[1]]$`nchar`  
[1] 3  
  
[[2]]  
[[2]]$name  
[1] "Hao"  
  
[[3]]  
[[3]]$name  
[1] "Si"  
[[4]]  
[[4]]$name  
[1] "Yang"  
[[5]]  
[[5]]$name  
[1] "Na"
```

SAPPLY

```
> names <- c('Hao', 'Si', 'Yang', 'Na')
> say_hello <- function (x){
+   paste0('Hello, ', x)
+ }
> sapply(names, say_hello)
```

| Hao | Si | Yang | Na |
|--------------|-------------|---------------|-------------|
| "Hello, Hao" | "Hello, Si" | "Hello, Yang" | "Hello, Na" |

MAPPLY

```
> research_cases <- data.frame(  
+   case_id = c(30, 35, 10, 12),  
+   bleeding = c(T, T, T, T),  
+   surgery = c(T, F, T, F),  
+   name = I(c('Hao', 'Si', 'Yang', 'Na'))  
+ )
```

MAPPLY

```
> query_patient_room <- function(name, bleeding, surgery) {  
+   room_type <- ifelse(bleeding && !surgery, 'ICU ward', 'General ward')  
+   paste0('Please meet ', name, ' in the ', room_type, '.')  
+ }  
  
> mapply(  
+   query_patient_room,  
+   research_cases$name,  
+   research_cases$bleeding,  
+   research_cases$surgery  
+ )
```

ADVANCED FLOW CONTROL



magrittr

PREPARE THE DATA

```
> library(mni2aal)  
> library(tidyverse)  
  
> coordinates = read.csv('./data/MNI_coordinates.csv')
```

PARENTHESES HELL & VARIABLE HELL

```
> cbind(  
+ coordinates,  
+ as.data.frame(  
+ t(  
+ mapply(  
+ mni_to_region_name,  
+ coordinates$x,  
+ coordinates$y,  
+ coordinates$z  
+ ))  
+ )  
+ )  
> aal_names_list <- mapply(  
+ mni_to_region_name,  
+ coordinates$x,  
+ coordinates$y,  
+ coordinates$z  
+ )  
> aal_names_long_matrix <- t(aal_names_list)  
> aal_names_df <- as.data.frame(aal_names_long_matrix)  
> aal_names <- cbind(coordinates, aal_names_df)
```

PIPELINE OPERATOR

```
> mapply(mni_to_region_name, coordinates$x, coordinates$y, coordinates$z) %>%  
+ t %>%  
+ as.data.frame %>%  
+ cbind(coordinates, .)
```



Data I/O and Plotting

READING CSV DATA

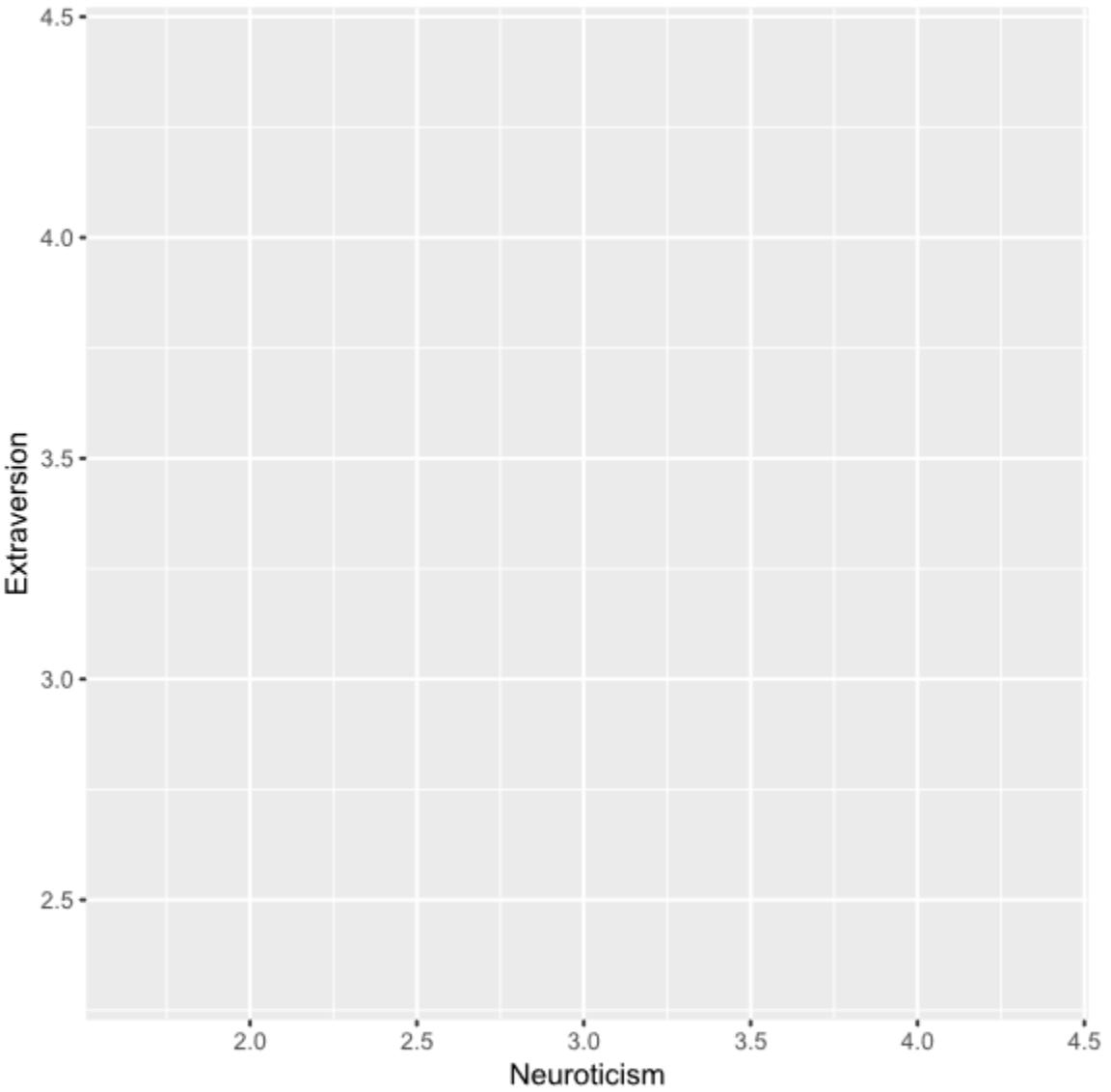
```
> bf <- read.csv('http://sc.bnu.edu.cn/learning-r/big5.csv')  
> bf
```

| | Neuroticism | Extraversion | Openness | Agreeableness | Conscientiousness | age | gender | home | drink |
|----|-------------|--------------|----------|---------------|-------------------|-----|--------|-------|-------|
| 1 | 2.479 | 4.208 | 3.938 | 3.958 | 3.458 | 27 | Male | South | Yes |
| 2 | 2.604 | 3.188 | 3.958 | 3.396 | 3.229 | 51 | Male | South | No |
| 3 | 2.813 | 2.896 | 3.417 | 2.750 | 3.500 | 45 | Male | South | No |
| 4 | 2.896 | 3.563 | 3.521 | 3.167 | 2.792 | 72 | Male | South | No |
| 5 | 3.021 | 3.333 | 4.021 | 3.208 | 2.854 | 26 | Male | North | No |
| 6 | 2.521 | 3.292 | 3.438 | 3.708 | 2.500 | 65 | Male | South | Yes |
| 7 | 2.354 | 4.417 | 4.583 | 3.063 | 3.333 | 48 | Male | South | No |
| 8 | 2.521 | 3.500 | 2.896 | 3.667 | 3.063 | 31 | Male | South | Yes |
| 9 | 3.104 | 3.813 | 4.063 | 3.771 | 2.833 | 51 | Male | North | Yes |
| 10 | 2.688 | 3.547 | 3.787 | 3.354 | 3.104 | 59 | Male | South | No |

PLOTTING

BASE

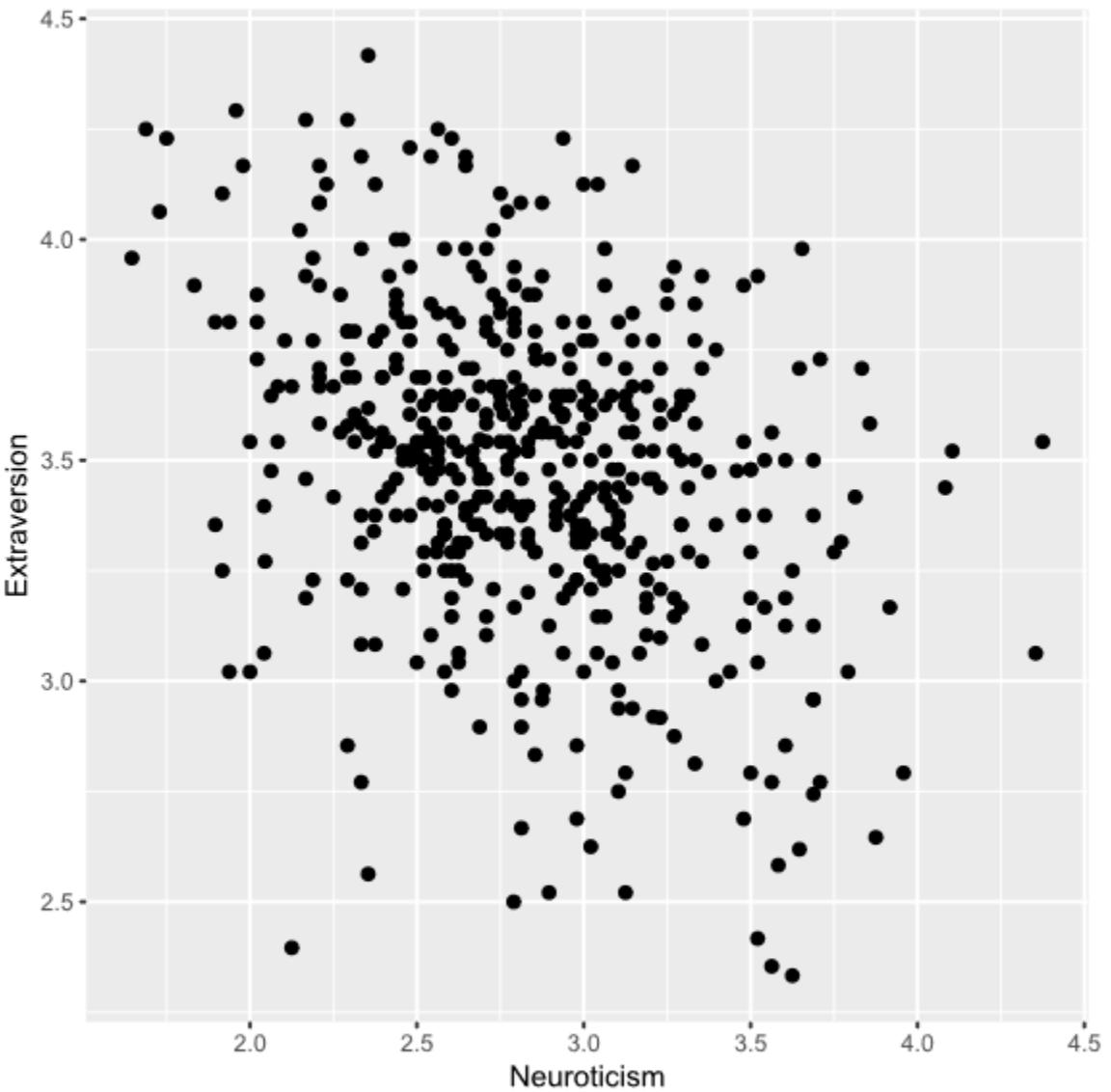
```
> library(tidyverse)  
> ggplot(bf, aes(Neuroticism, Extraversion))
```



PLOTTING

DOTPLOT

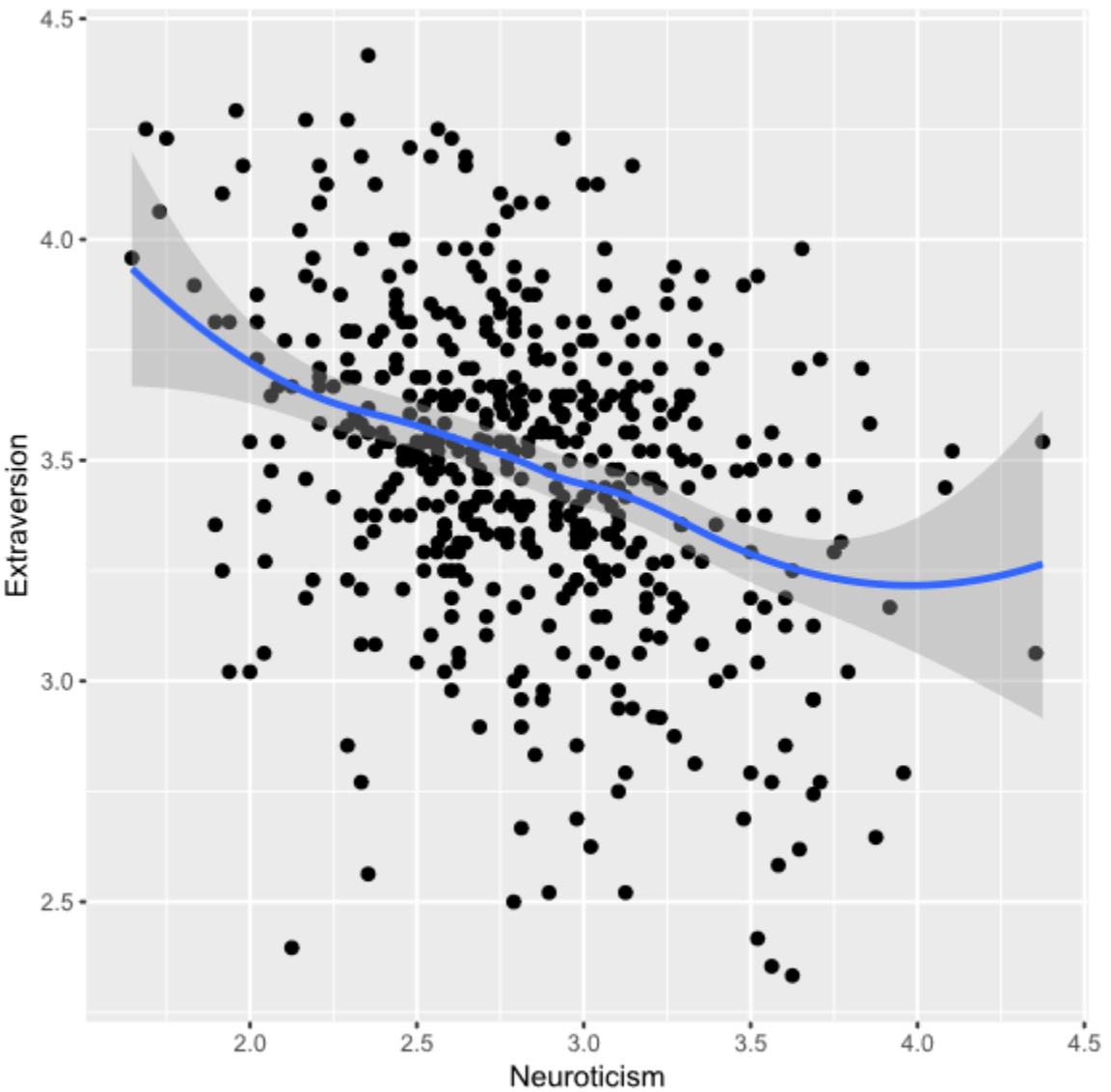
```
> library(tidyverse)  
> ggplot(bf, aes(Neuroticism, Extraversion)) +  
+ geom_point()
```



PLOTTING

DOTPLOT

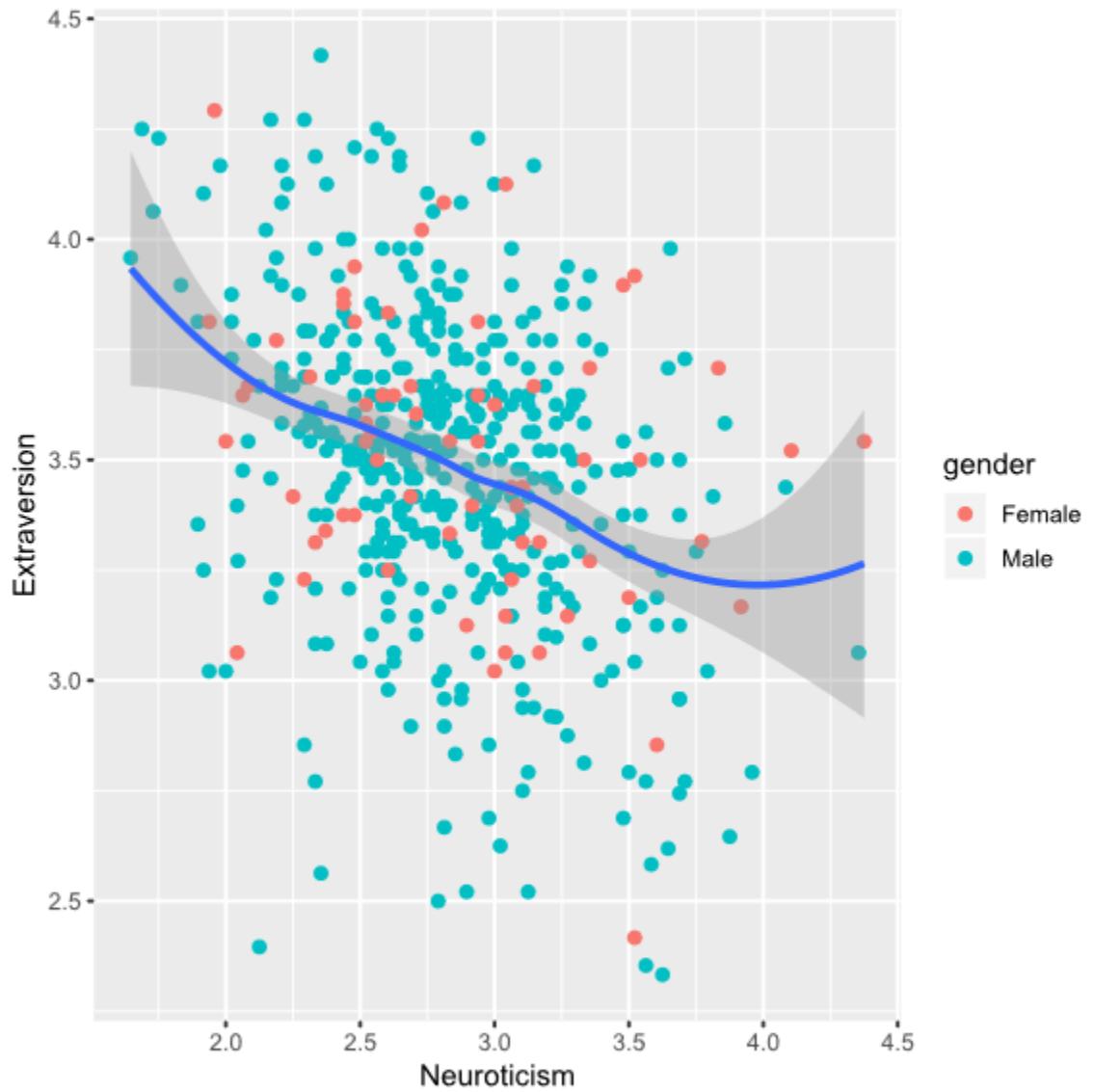
```
> library(tidyverse)  
> ggplot(bf, aes(Neuroticism, Extraversion)) +  
+ geom_point() +  
+ stat_smooth()
```



PLOTTING

DOTPLOT

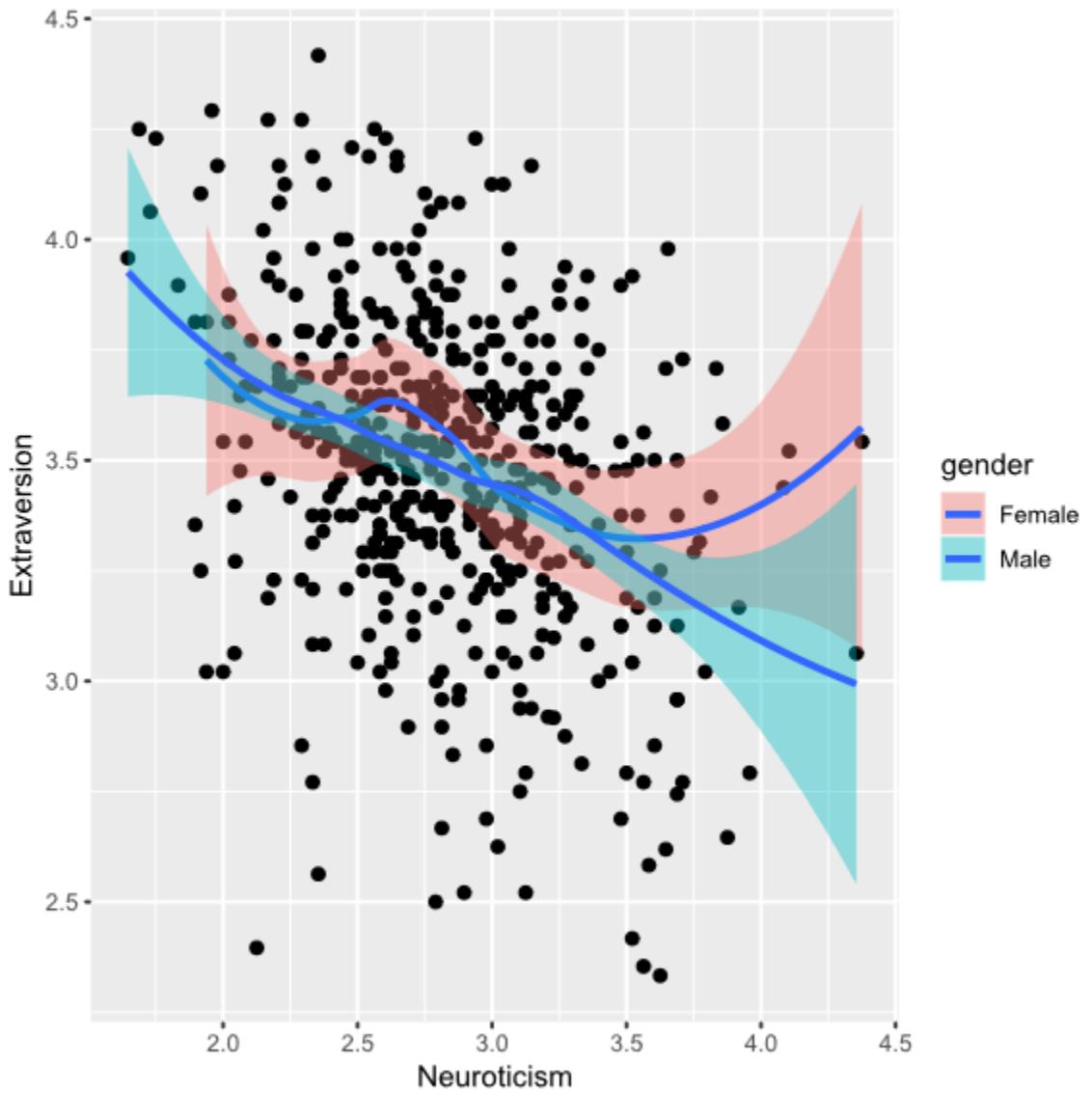
```
> library(tidyverse)  
> ggplot(bf, aes(Neuroticism, Extraversion)) +  
+ geom_point(aes(color = gender)) +  
+ stat_smooth()
```



PLOTTING

DOTPLOT

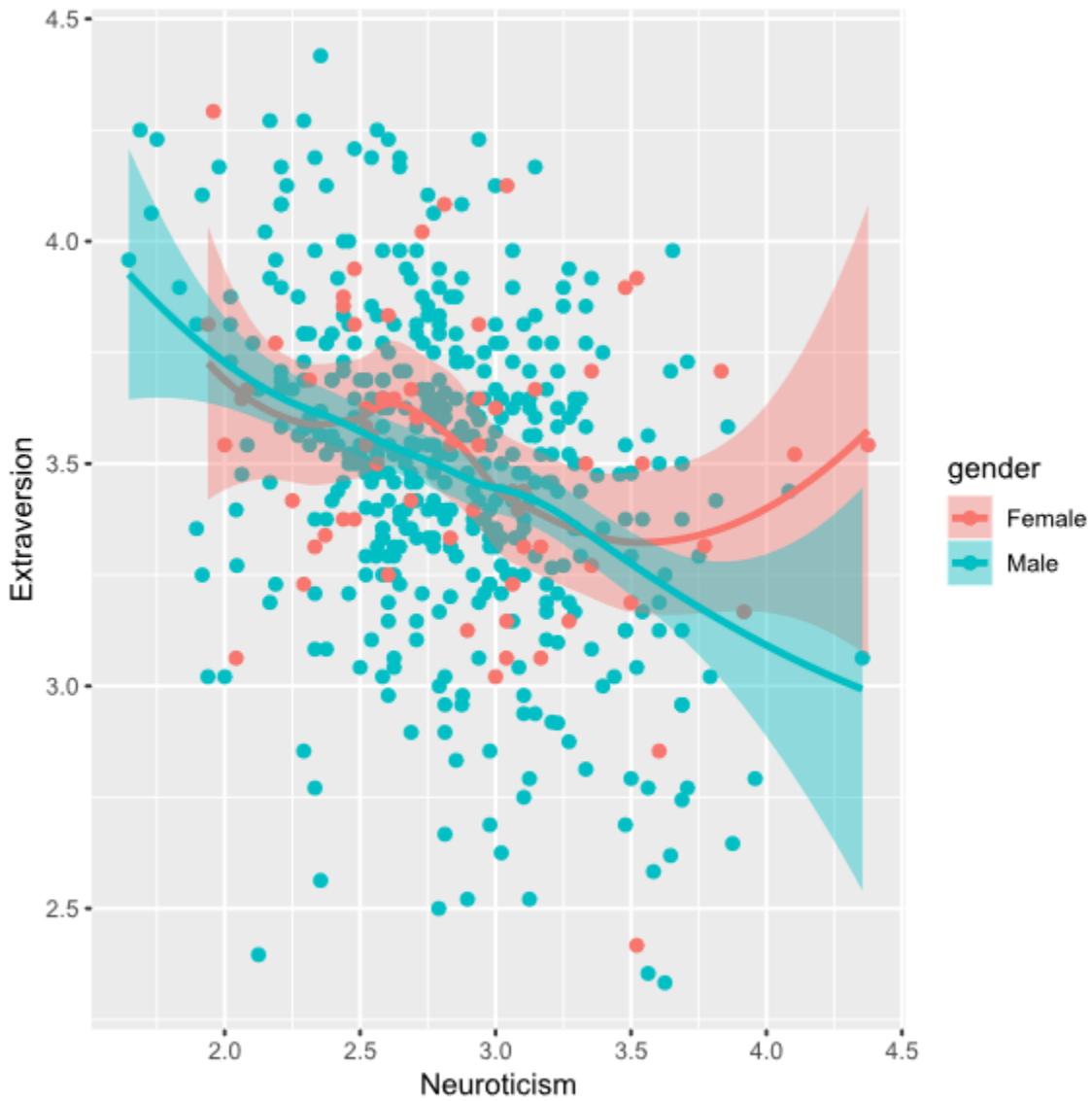
```
> library(tidyverse)  
> ggplot(bf, aes(Neuroticism, Extraversion)) +  
+ geom_point() +  
+ stat_smooth(aes(fill = gender))
```



PLOTTING

DOTPLOT

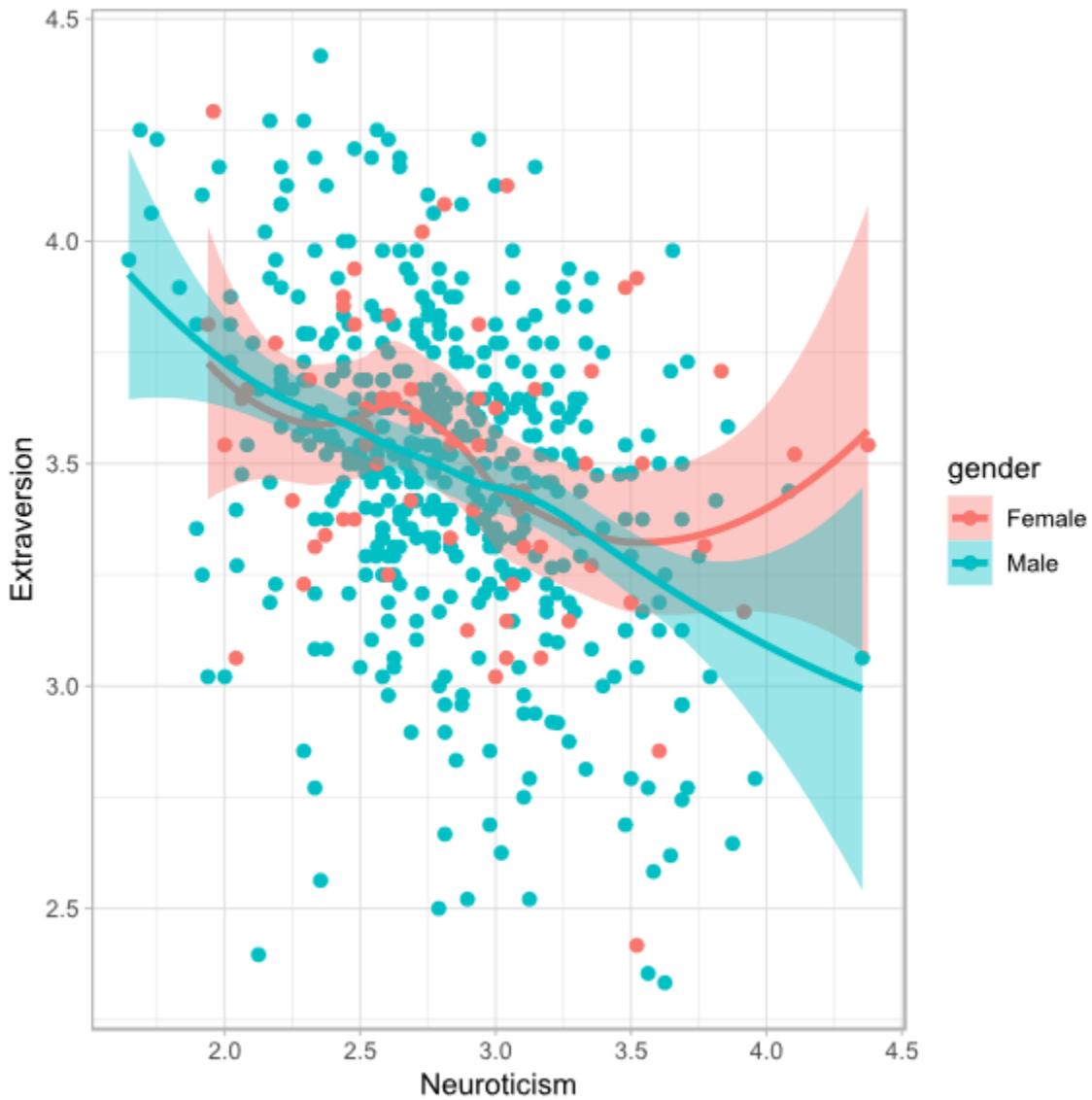
```
> library(tidyverse)
> ggplot(bf,
+   aes(
+     Neuroticism, Extraversion,
+     color = gender, fill = gender
+   )
+ ) +
+   geom_point() +
+   stat_smooth()
```



PLOTTING

DOTPLOT

```
> library(tidyverse)
> ggplot(bf,
+   aes(
+     Neuroticism, Extraversion
+     color = gender, fill = gender
+   )
+   +
+   geom_point() +
+   stat_smooth() +
+   theme_light()
```



PLOTTING

DOTPLOT

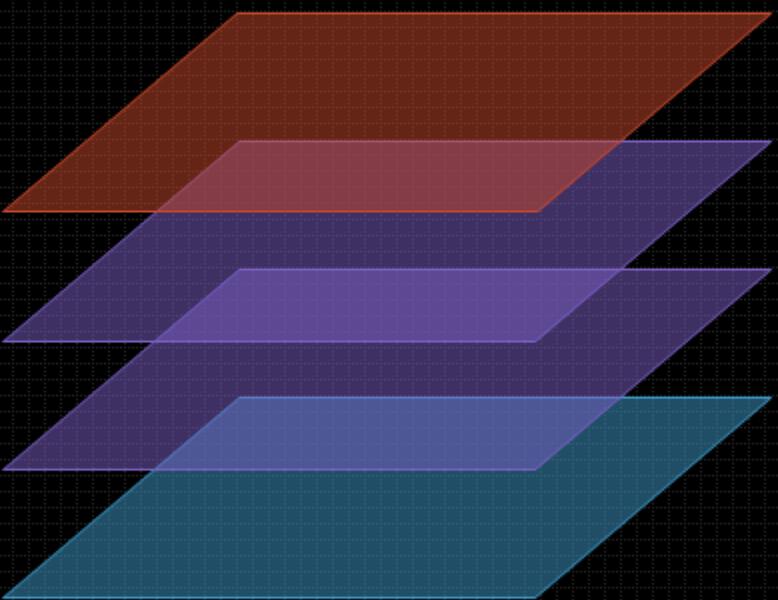
```
> library(tidyverse)
> ggplot(bf,
+   aes(
+     Neuroticism, Extraversion
+     color = gender, fill = gender
+   )
+ ) +
+   geom_point() +
+   stat_smooth() +
+   theme_light()
```

Theme

Smooth

Point

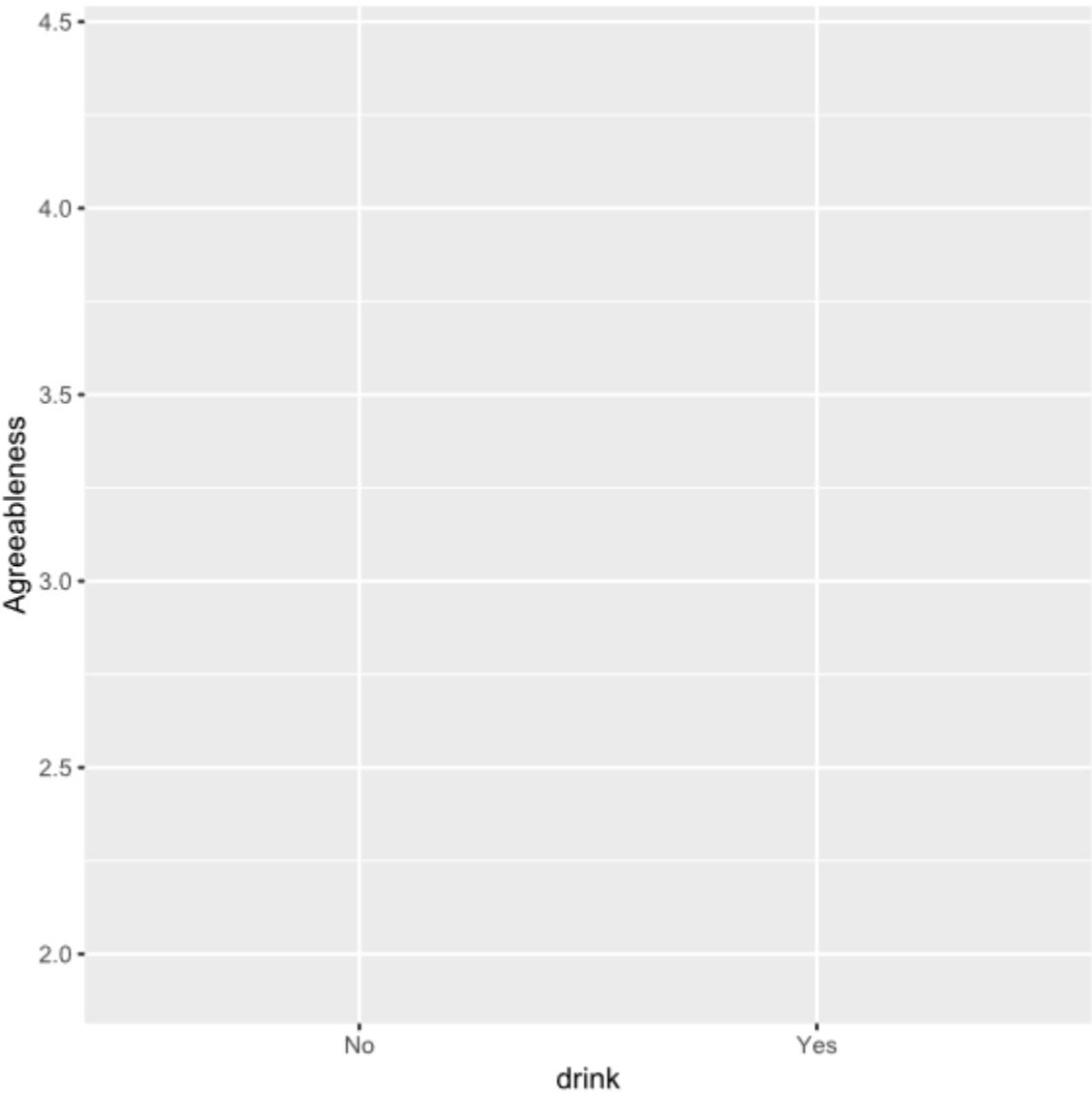
Base



PLOTTING

VIOLIN PLOT

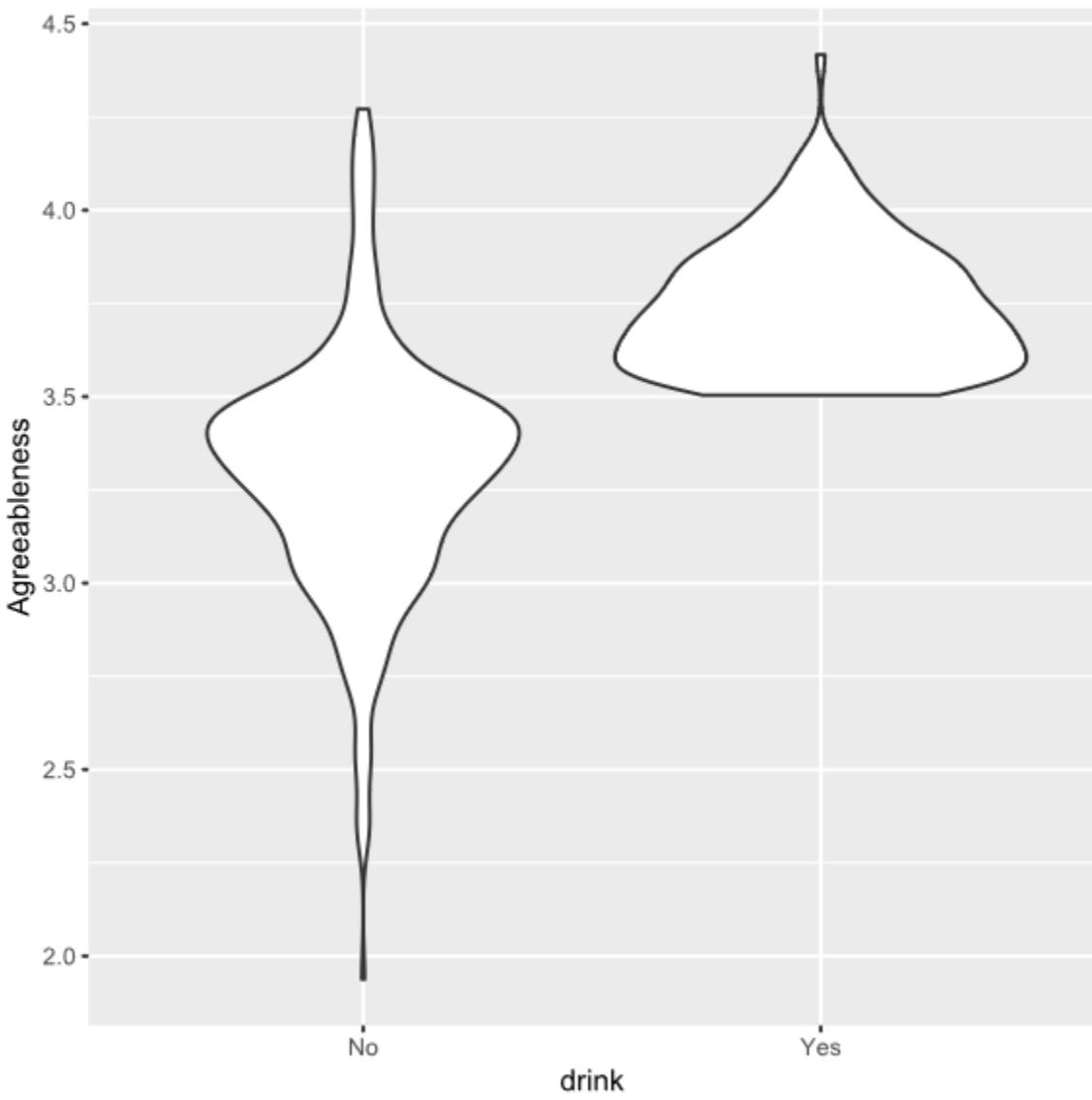
```
> ggplot(bf, aes(drink, Agreeableness))
```



PLOTTING

VIOLIN PLOT

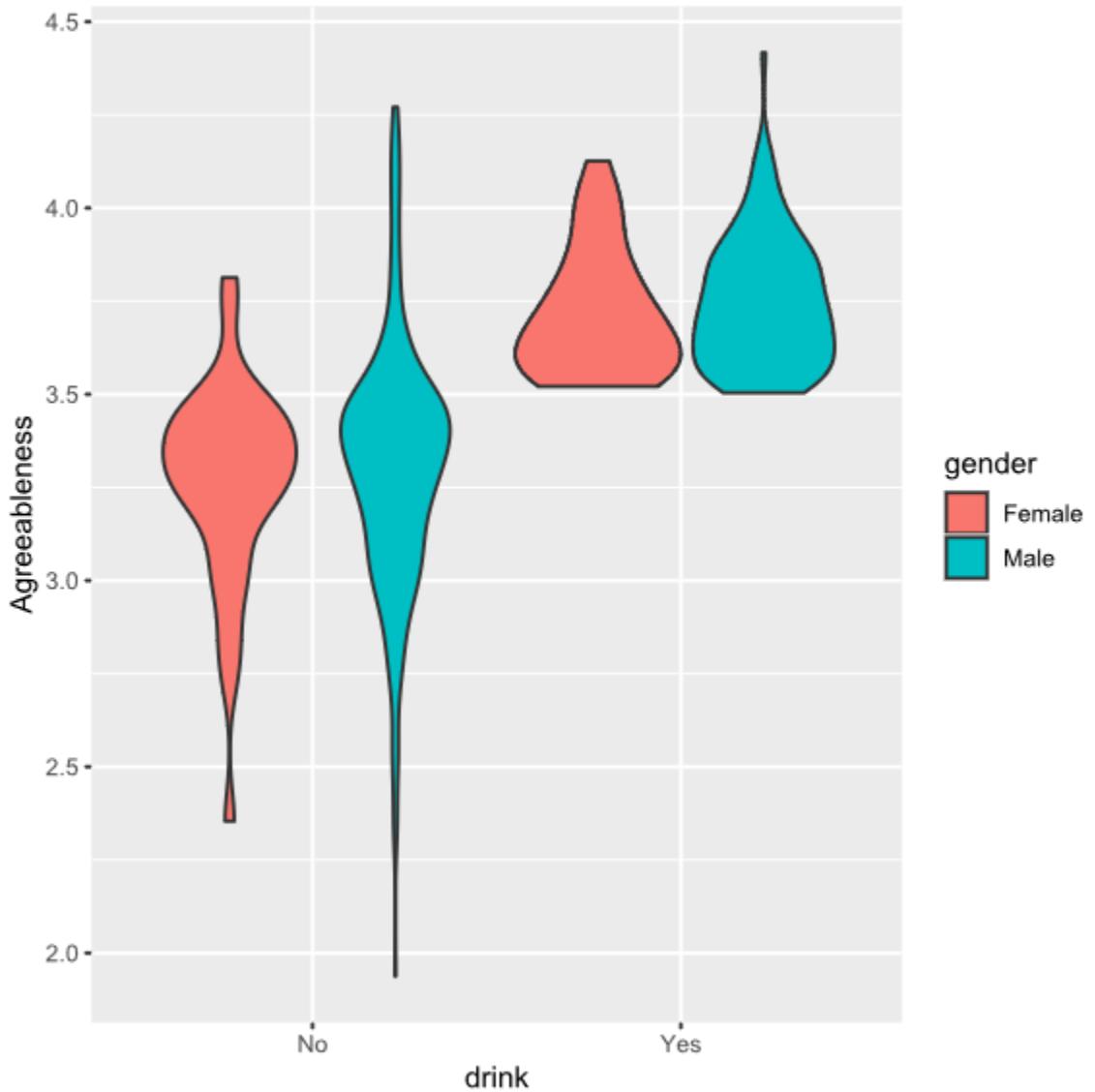
```
> ggplot(bf, aes(drink, Agreeableness)) +  
> geom_violin()
```



PLOTTING

VIOLIN PLOT

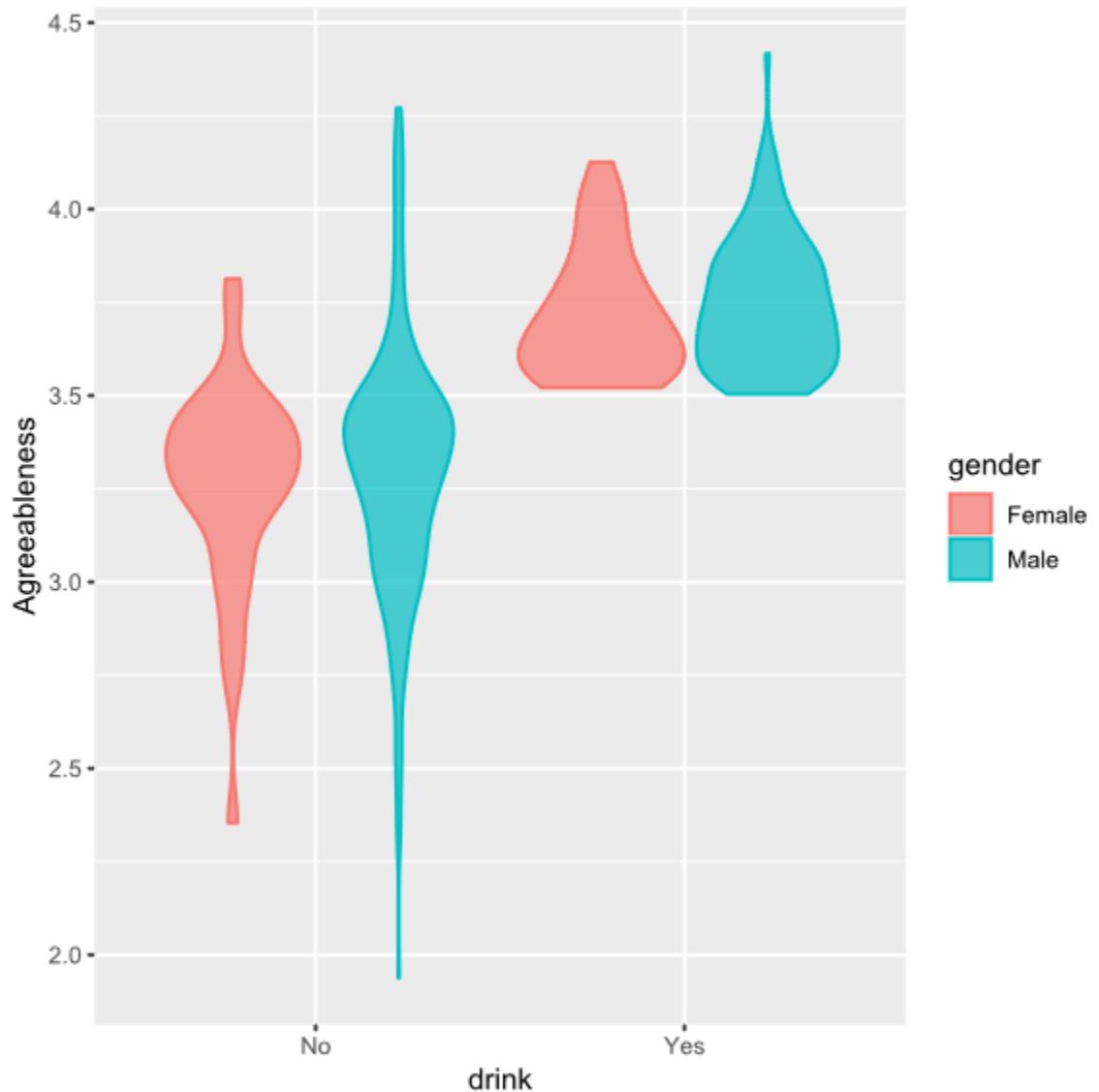
```
> ggplot(bf, aes(drink, Agreeableness)) +  
> geom_violin(aes(fill=gender))
```



PLOTTING

VIOLIN PLOT

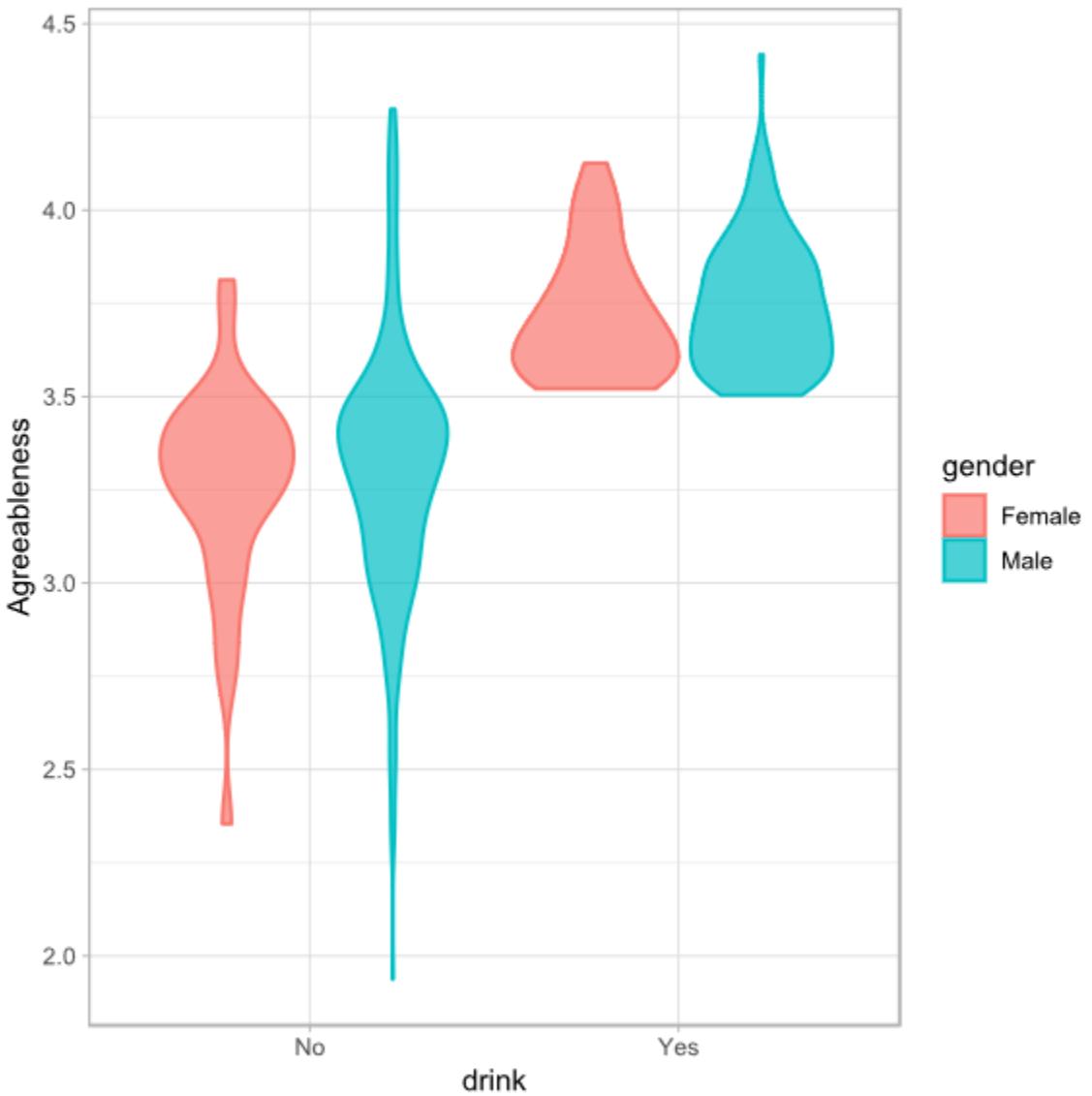
```
> ggplot(bf, aes(drink, Agreeableness)) +  
> geom_violin(  
+   alpha = 0.7,  
+   aes(  
+     fill = gender,  
+     color = gender  
+   )  
+ )
```



PLOTTING

VIOLIN PLOT

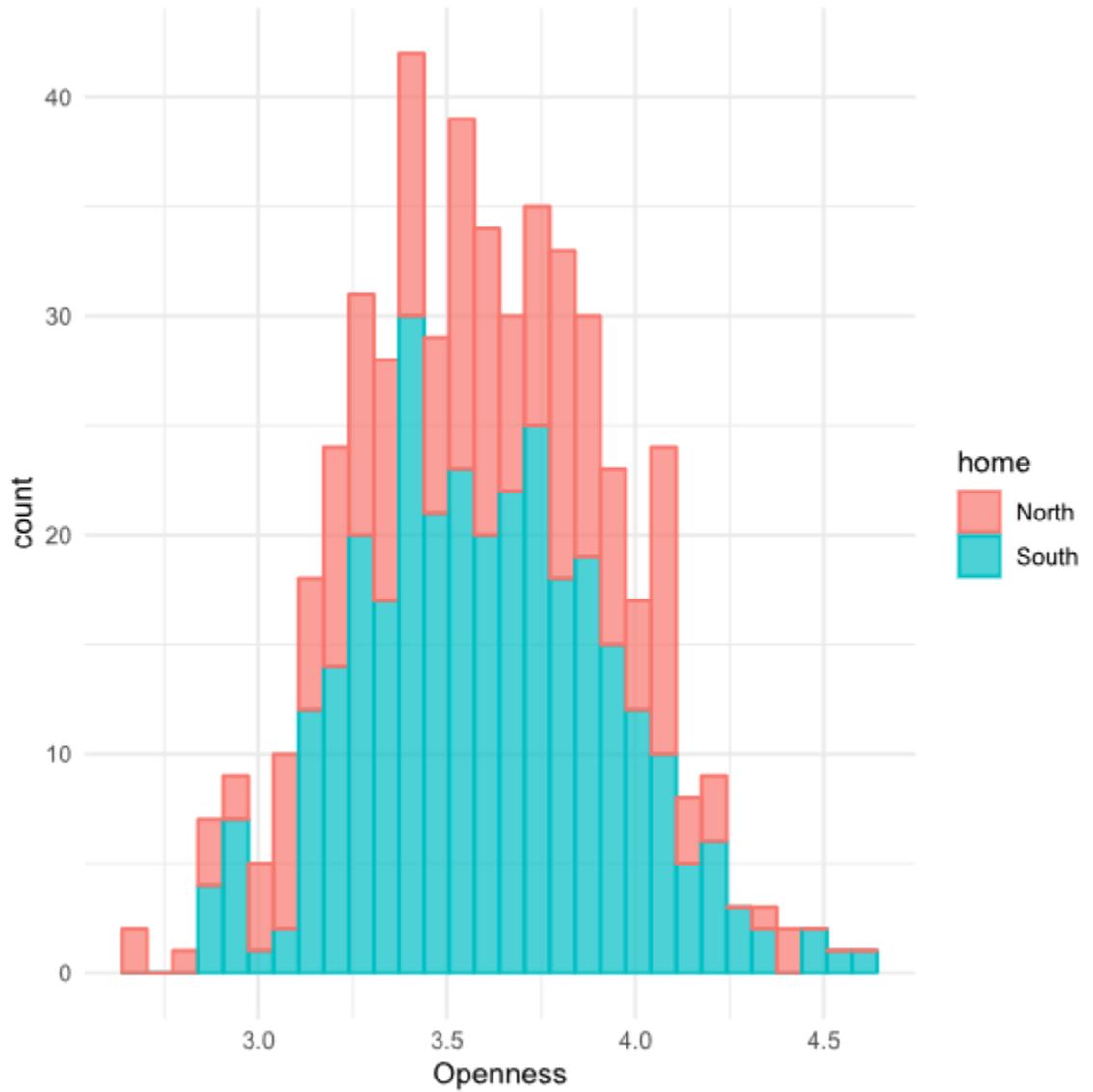
```
> ggplot(bf, aes(drink, Agreeableness)) +  
> geom_violin(  
+   alpha = 0.7,  
+   aes(  
+     fill = gender,  
+     color = gender  
+   )  
+ ) +  
+ theme_light()
```



PLOTTING

HISTOGRAM

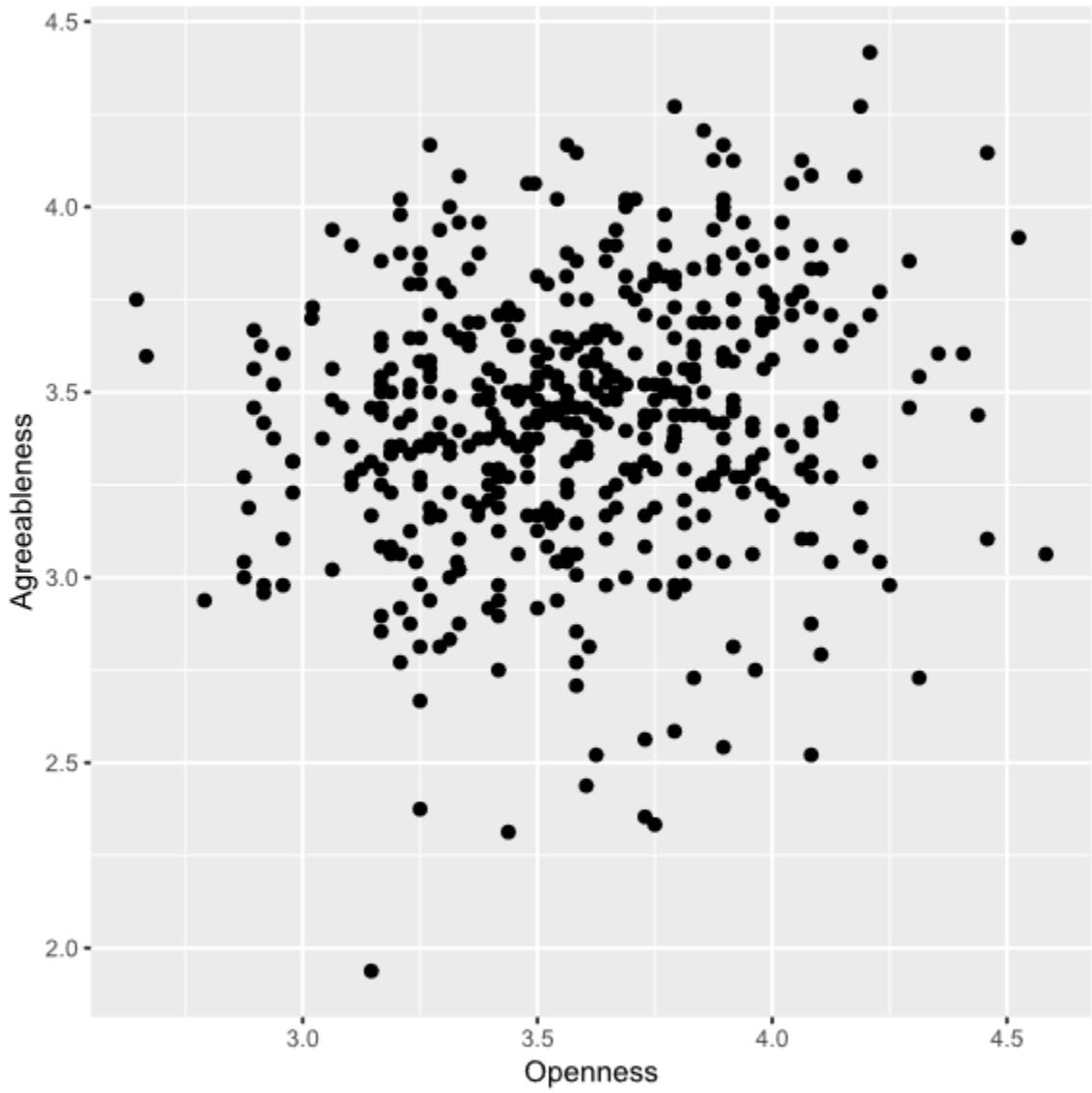
```
> ggplot(bf, aes(Openness, color = home)) +  
+   geom_histogram(  
+     alpha = 0.7,  
+     aes(fill = home)  
+   ) +  
+   theme_minimal()
```



PLOTTING

FACET

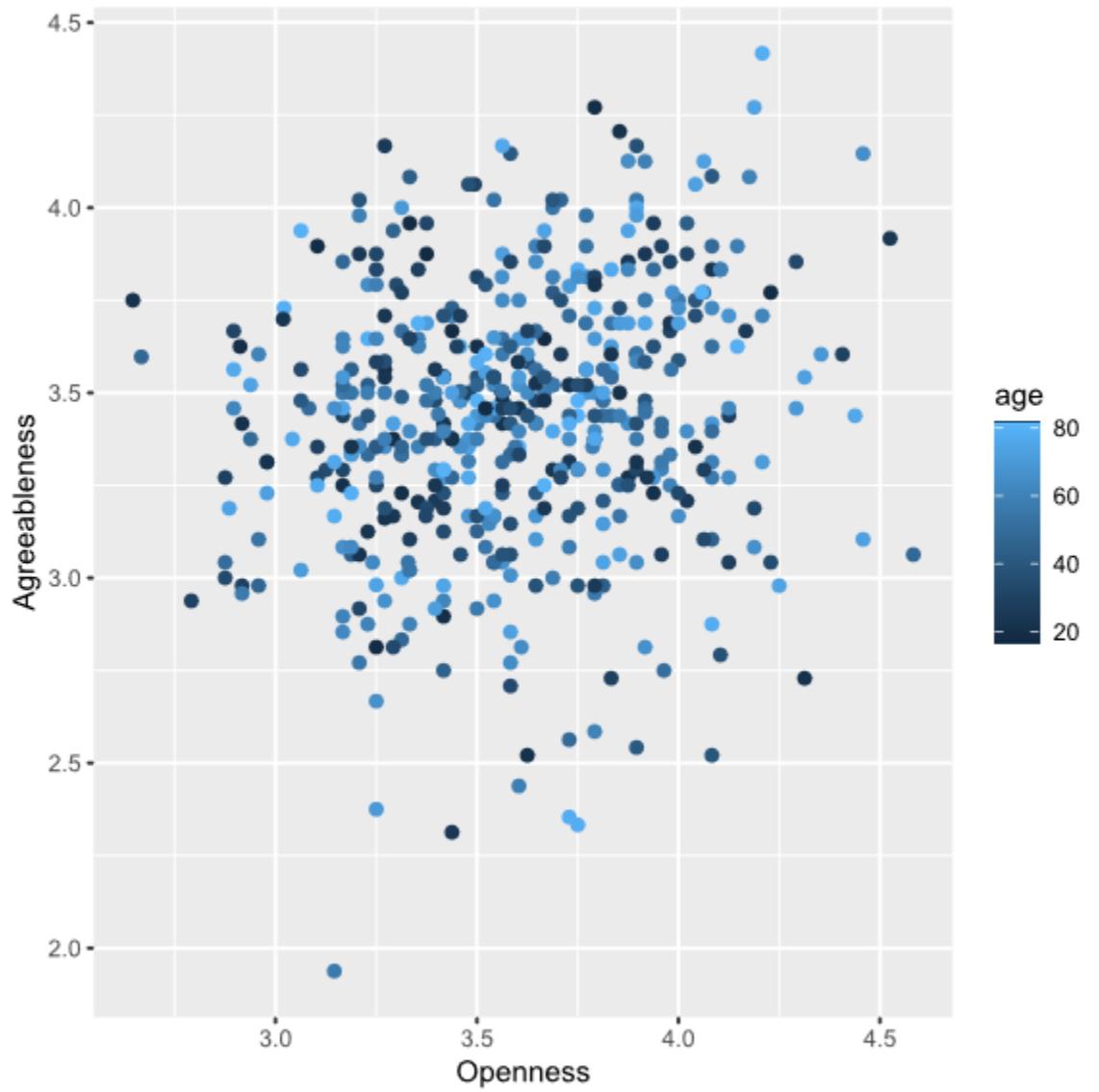
```
> ggplot(bf, aes(Openness, Agreeableness)) +  
+ geom_point()
```



PLOTTING

FACET

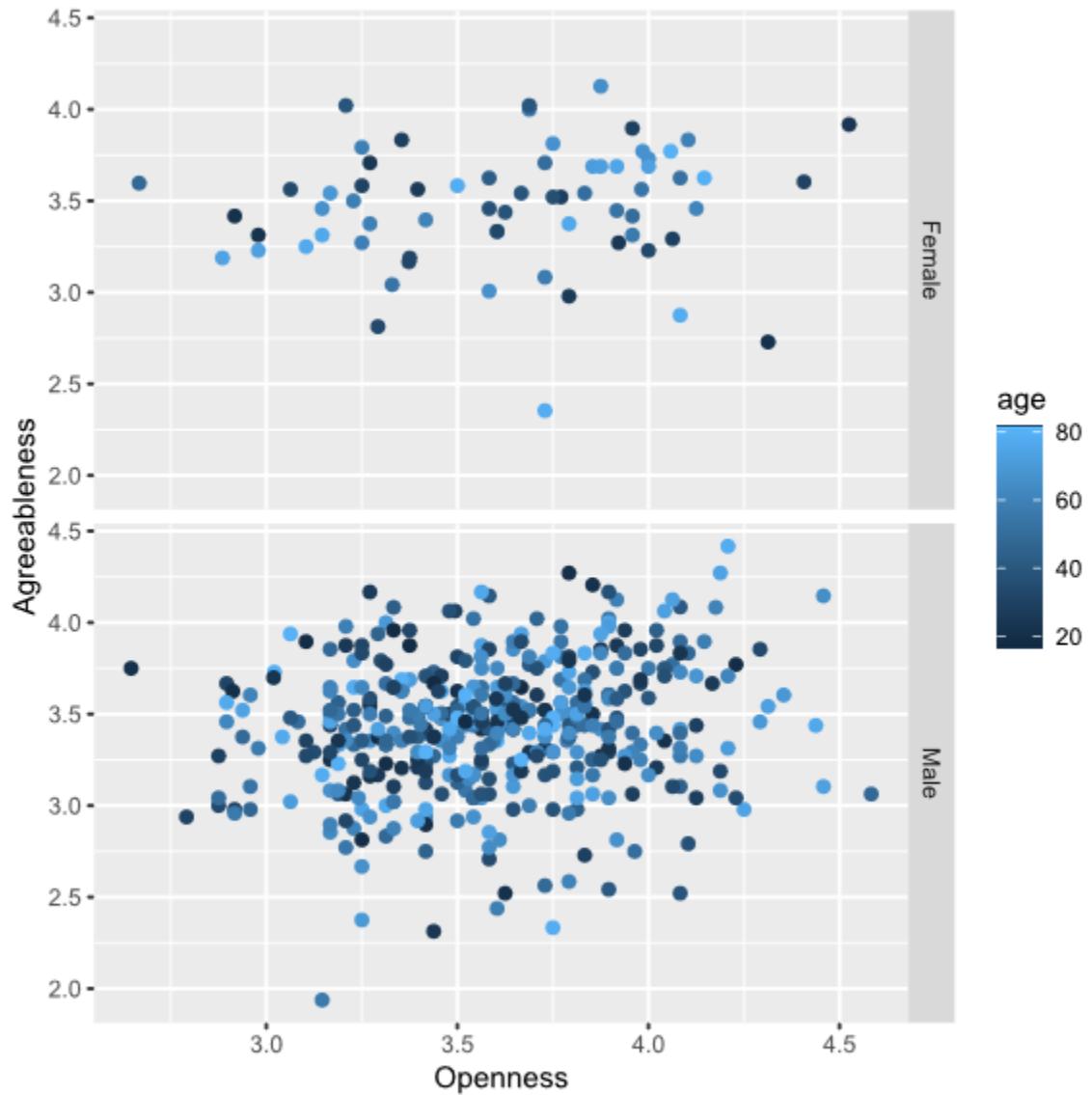
```
> ggplot(bf, aes(Openness, Agreeableness)) +  
+ geom_point(aes(color = age))
```



PLOTTING

FACET

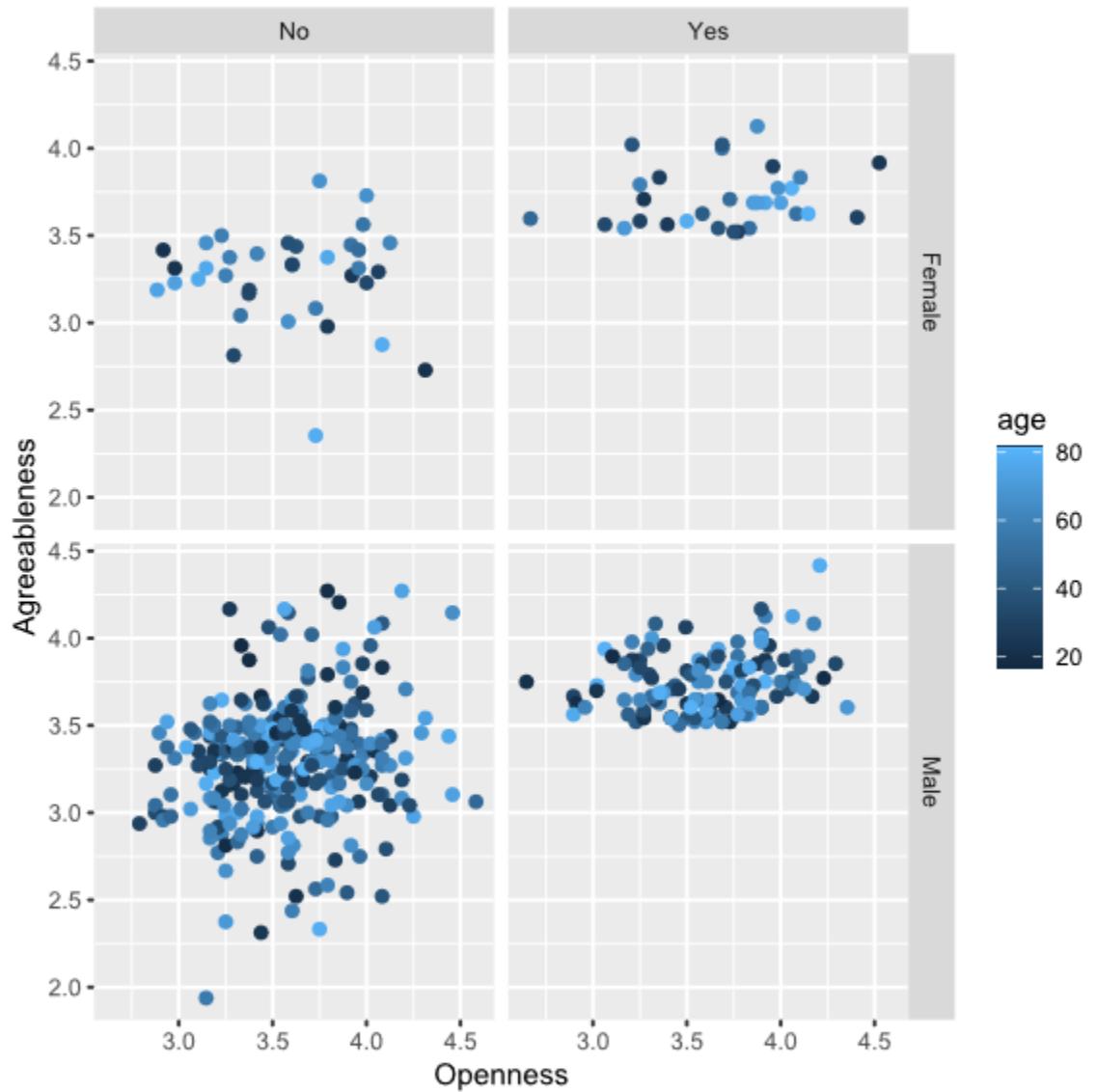
```
> ggplot(bf, aes(Openness, Agreeableness)) +  
+ geom_point(aes(color = age)) +  
+ facet_grid(rows = vars(gender))
```



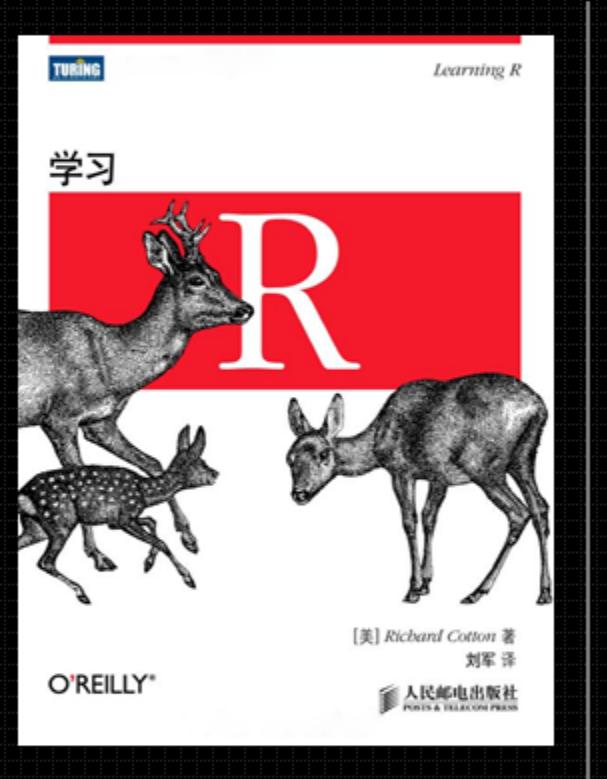
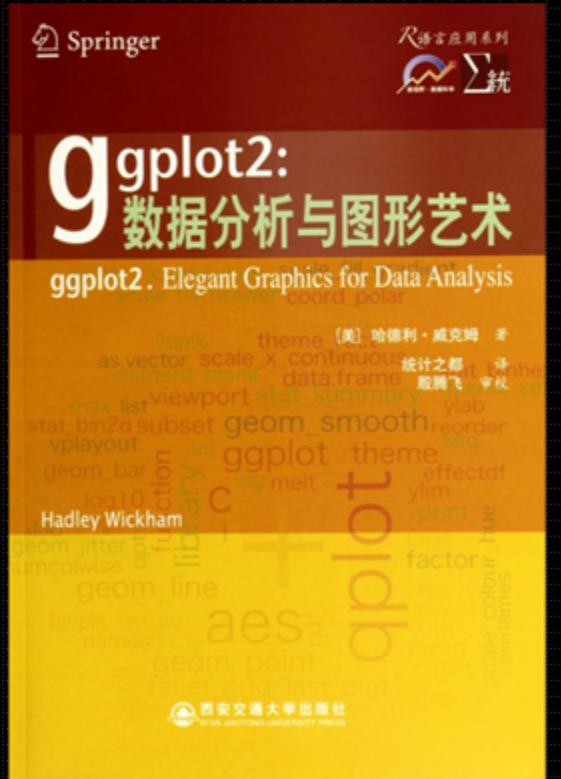
PLOTTING

FACET

```
> ggplot(bf, aes(Openness, Agreeableness)) +  
+ geom_point(aes(color = age)) +  
+ facet_grid(  
+   rows = vars(gender),  
+   cols = vars(drink)  
+ )
```



Self-study



GETTING SOME HELP

```
> ?`%>%`
```

```
> ?ggplot
```

R: magrittr forward-pipe operator

%>% {magrittr}

R Documentation

magrittr forward-pipe operator

Description

Pipe an object forward into a function or call expression.

Usage

```
lhs %>% rhs
```

Arguments

`lhs` A value or the `magrittr` placeholder.
`rhs` A function call using the `magrittr` semantics.

Details

Using %>% with unary function calls

When functions require only one argument, `x %>% f` is equivalent to `f(x)` (not exactly equivalent; see technical note below.)

Placing lhs as the first argument in rhs call

The default behavior of `%>%` when multiple arguments are required in the `rhs` call, is to place `lhs` as the first argument, i.e. `x %>% f(y)` is equivalent to `f(x, y)`.

Placing lhs elsewhere in rhs call

Often you will want `lhs` to the `rhs` call at another position than the first. For this purpose you can use the dot `(.)` as placeholder. For example, `y %>% f(x, .)` is equivalent to `f(x, y)` and `z %>% f(x, y, arg = .)` is equivalent to `f(x, y, arg = z)`.

Using the dot for secondary purposes

Often, some attribute or property of `lhs` is desired in the `rhs` call in addition to the value of `lhs` itself, e.g. the number of rows or columns. It is perfectly valid to use the dot placeholder several times in the `rhs` call, but by design the behavior is slightly different when using it inside nested function calls. In particular, if the placeholder is only used in a nested function call, `lhs` will also be placed as the first argument! The reason for this is that in most use-cases this produces the most readable code. For example, `iris %>% subset(1:nrow(.) %% 2 == 0)` is equivalent to `iris %>% subset(., 1:nrow(.) %% 2 == 0)` but slightly more compact. It is possible to overrule this behavior by enclosing the `rhs` in braces. For example, `1:10 %>% {c(min(.), max(.))}` is equivalent to `c(min(1:10), max(1:10))`.

Using %>% with call- or function-producing rhs

It is possible to force evaluation of `rhs` before the piping of `lhs` takes place. This is useful when `rhs` produces the relevant call or function. To evaluate `rhs` first, enclose it in parentheses, i.e. `a %>% (function(x) x^2)`, and `1:10 %>% (call("sum"))`. Another example where this is relevant is for reference class methods which are accessed using the `$` operator, where one would do `x %>% (rc$fn)`, and not `x %>% rc$fn`.

WEBSITES

- 统计之都: <https://cosx.org/>
- R-bloggers: <https://www.r-bloggers.com/>
- Tidyverse: <http://tidyverse.org/>
- Cheatsheets: <https://github.com/rstudio/cheatsheets>

That's All 
Thanks