Introduction to tidyverse

By
Ravin Poudel and Garrett Lab
ICPP, Boston, MA

[*ravinpoudel.github.io*](ravinpoudel.github.io)
[*@raveenpoudel*](@raveenpoudel)

UF | IFAS UNIVERSITY *of* FLORIDA | Institute for Sustainable Food Systems

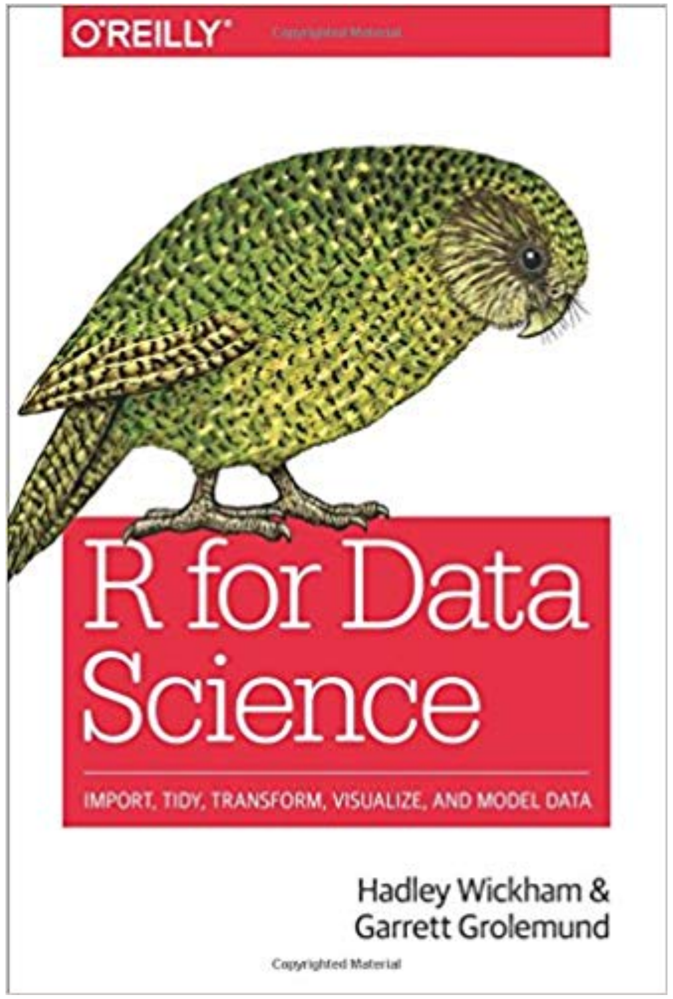
University of Florida Plant Pathology

Learning objectives

- Learn to setup R projects and tidy-work environment
- Learn data wrangling using dplyr
- Data visualization (ggplot2)

R for Data Science

http://r4ds.had.co.nz/



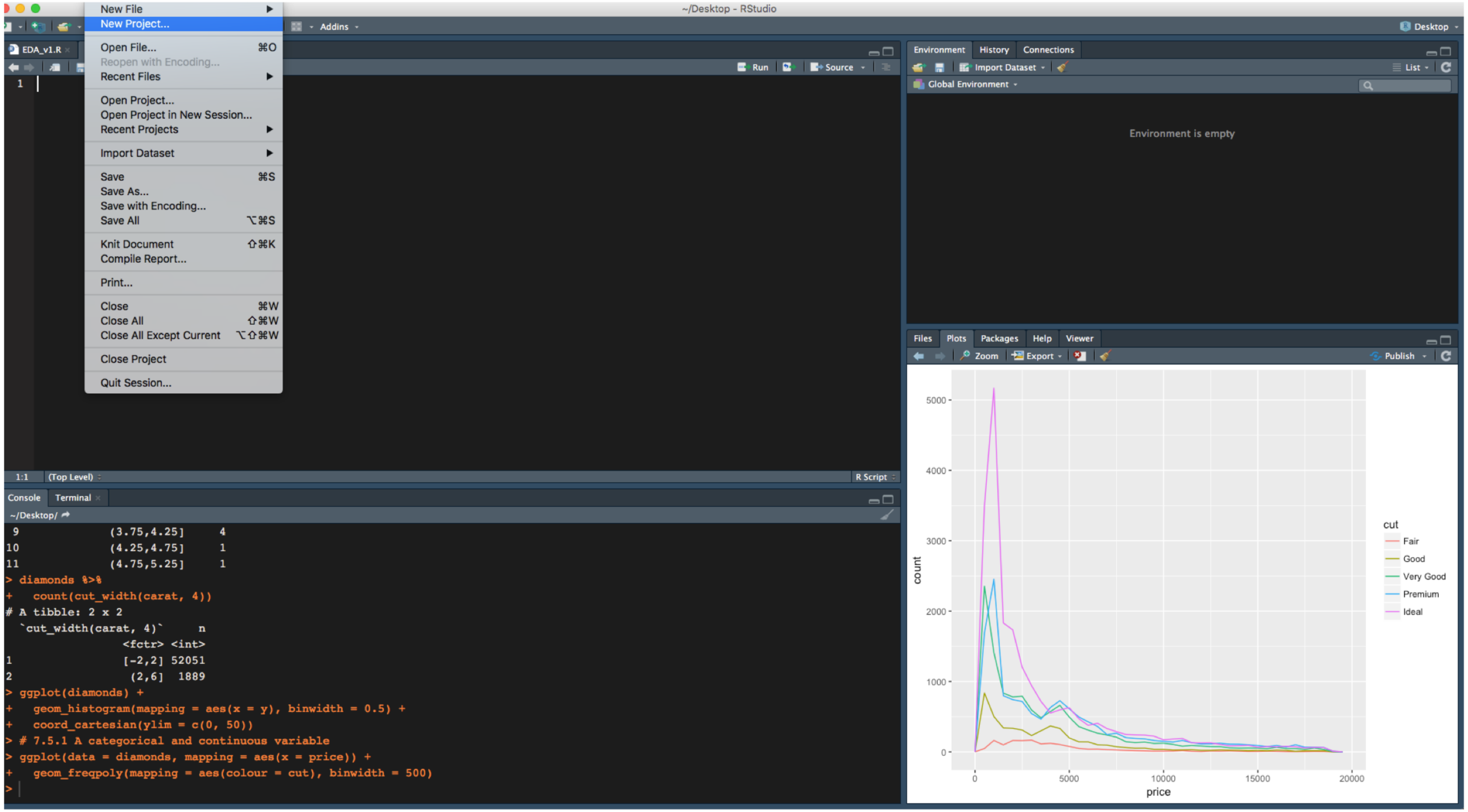
Why you need to setup R project? RStudio as your lab

Messy lab

Organized lab

Part One: R project

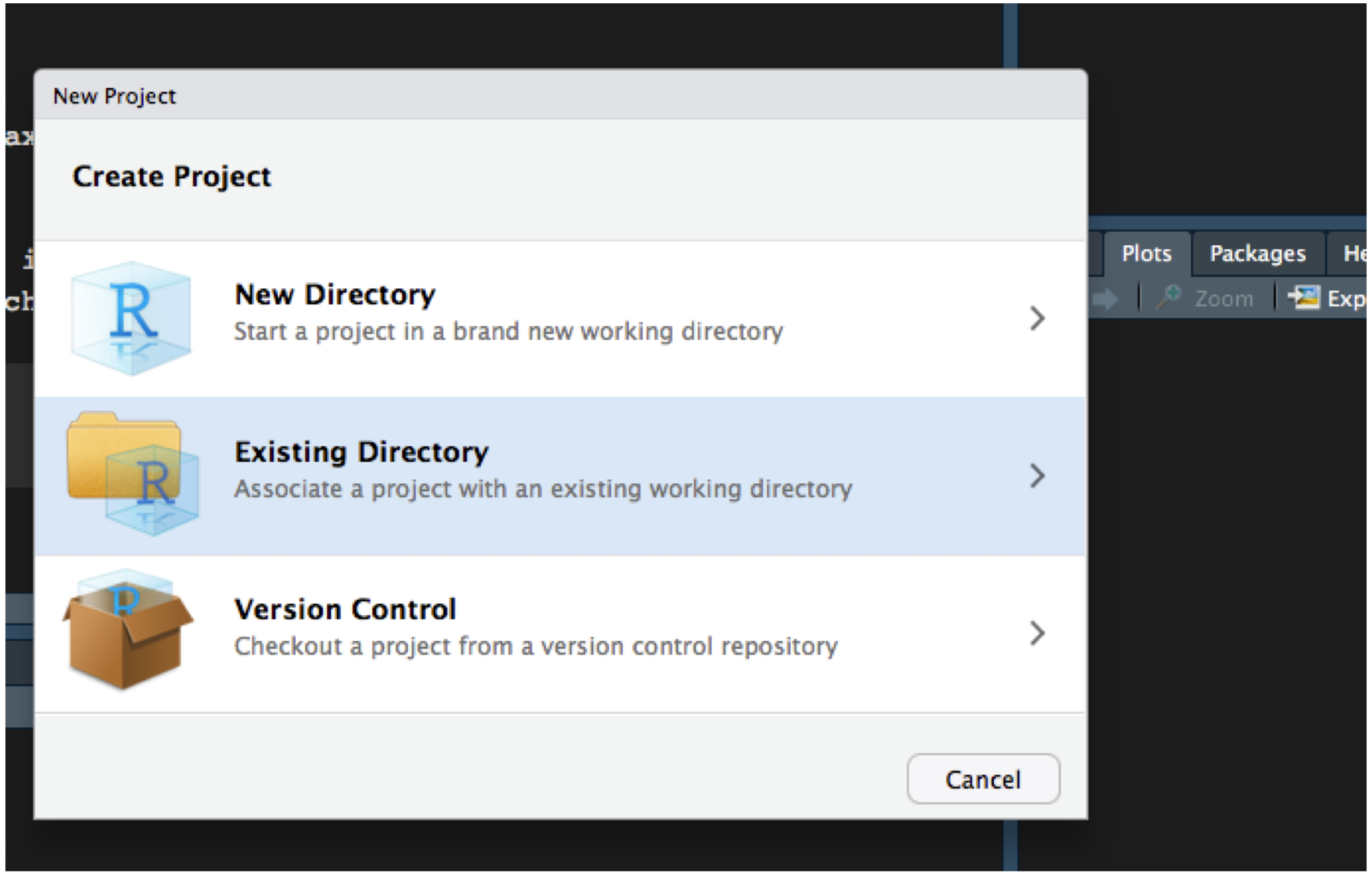
STEP: 1



```
9             (3.75,4.25]     4
10            (4.25,4.75]     1
11            (4.75,5.25]     1
> diamonds %>%
+   count(cut_width(carat, 4))
# A tibble: 2 x 2
  `cut_width(carat, 4)`     n
                 <fctr> <int>
1                [-2,2] 52051
2                 (2,6]  1889
> ggplot(diamonds) +
+   geom_histogram(mapping = aes(x = y), binwidth = 0.5) +
+   coord_cartesian(ylim = c(0, 50))
> # 7.5.1 A categorical and continuous variable
> ggplot(data = diamonds, mapping = aes(x = price)) +
+   geom_freqpoly(mapping = aes(colour = cut), binwidth = 500)
>
```

Part One: R project

STEP: 2



New Project

Create Project

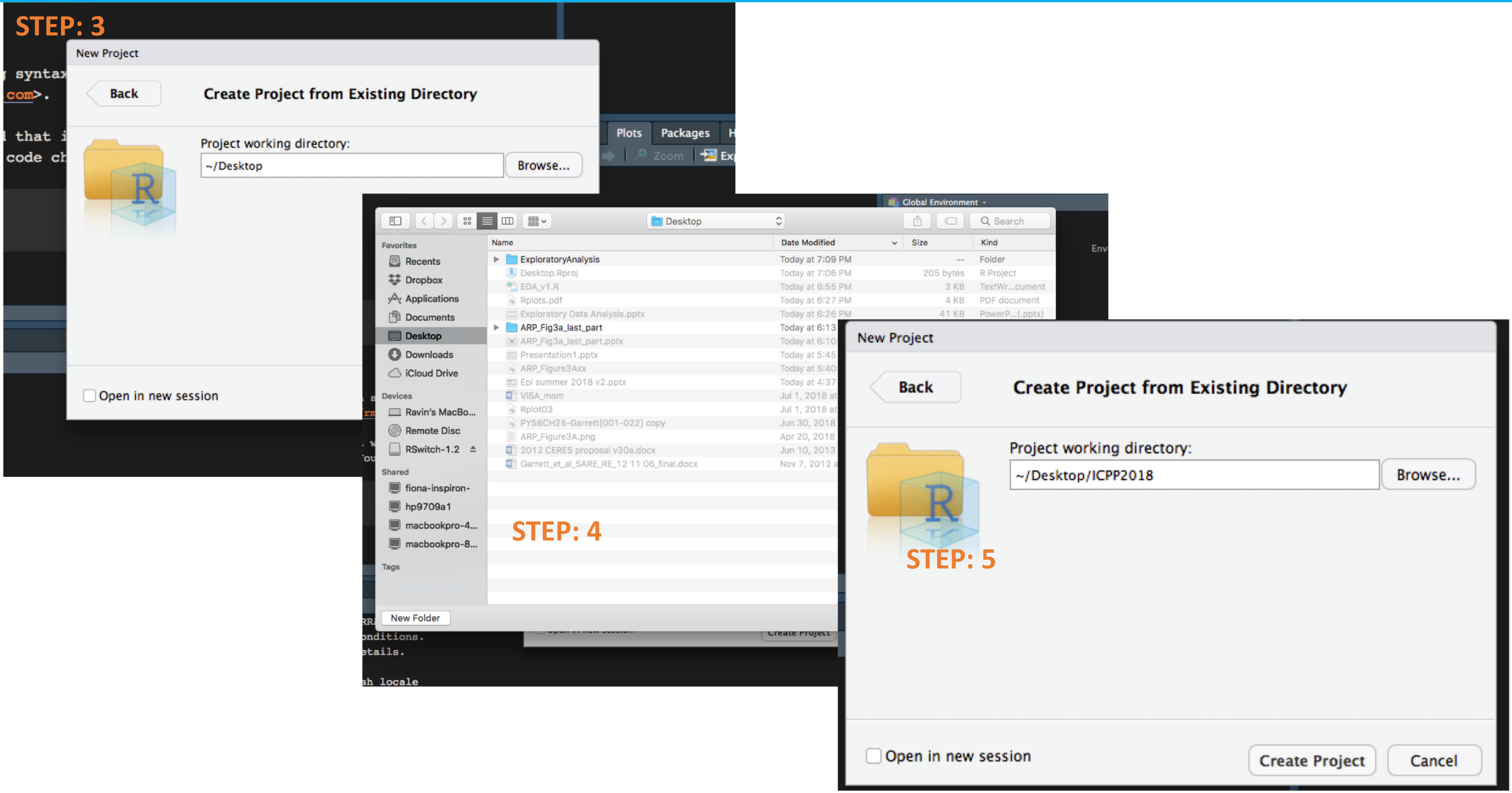
New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Cancel

R project



STEP: 3

New Project

Back

Create Project from Existing Directory

Project working directory:

~/Desktop

Browse...

Open in new session

STEP: 4

Desktop

Name	Date Modified	Size	Kind
ExploratoryAnalysis	Today at 7:09 PM	--	Folder
Desktop.Rproj	Today at 7:06 PM	205 bytes	R Project
EDA_v1.R	Today at 6:55 PM	3 KB	TextWr...cument
Rplots.pdf	Today at 6:27 PM	4 KB	PDF document
Exploratory Data Analysis.pptx	Today at 6:26 PM	41 KB	PowerP...(.pptx)
ARP_Fig3a_last_part	Today at 6:13		
ARP_Fig3a_last_part.pptx	Today at 6:10		
Presentation1.pptx	Today at 5:45		
ARP_Figure3Axx	Today at 5:40		
Epi summer 2018 v2.pptx	Today at 4:37		
VISA_mom	Jul 1, 2018 at		
Rplot03	Jul 1, 2018 at		
PY56CH25-Garrett[001-022] copy	Jun 30, 2018		
ARP_Figure3A.png	Apr 20, 2018		
2012 CERES proposal v30a.docx	Jun 10, 2013		
Garrett_et_al_SARE_RE_12 11 06_final.docx	Nov 7, 2012 a		

New Folder

STEP: 5

New Project

Back

Create Project from Existing Directory

Project working directory:

~/Desktop/ICPP2018

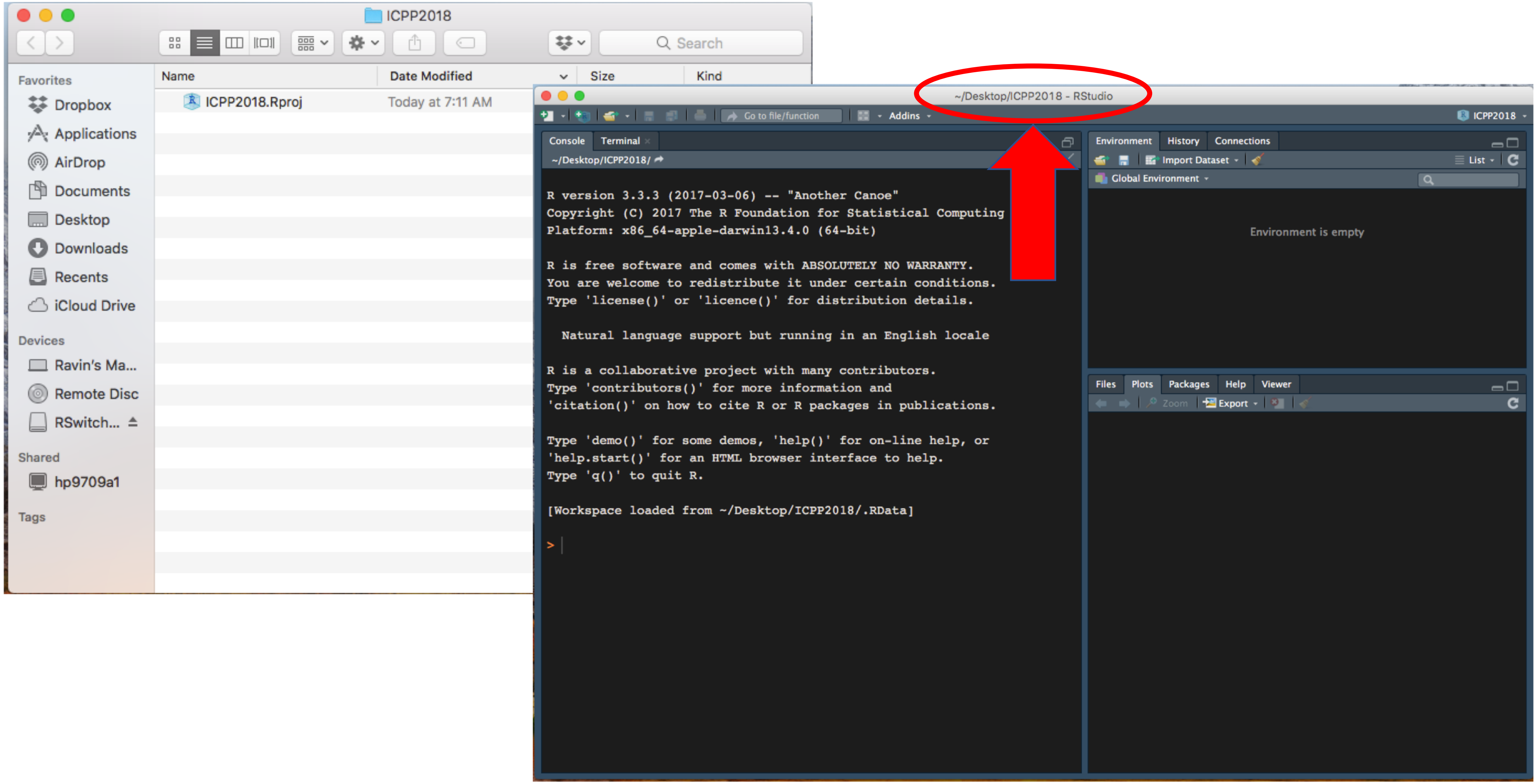
Browse...

Open in new session

Create Project

Cancel

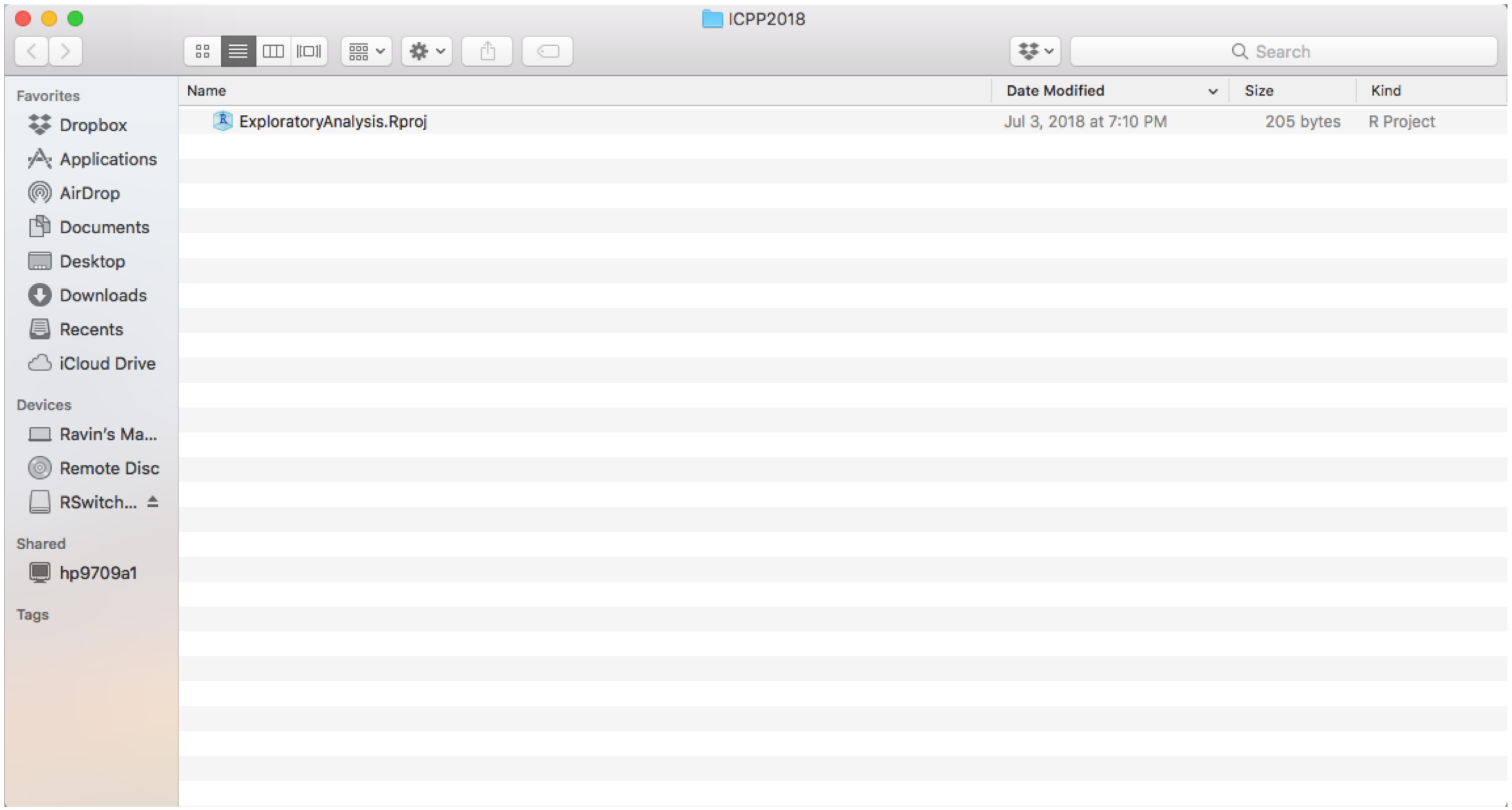
R project



```
R version 3.3.3 (2017-03-06) -- "Another Canoe"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/Desktop/ICPP2018/.RData]

>
```

R project

Everything you need is in one place



Name	Date Modified	Size	Kind
ExploratoryAnalysis.Rproj	Jul 3, 2018 at 7:10 PM	205 bytes	R Project

Advantages

- Allow to save all materials related to a single analysis in one working environment and sub-folders
- No need to worry about the file paths - less error
- Easy sharing and reproducible
- Saving working environment and output objects save time, especially if your input file is too large

Data wrangling and basic plots using tidyverse

Key dplyr functions in dplyr package for data manipulation

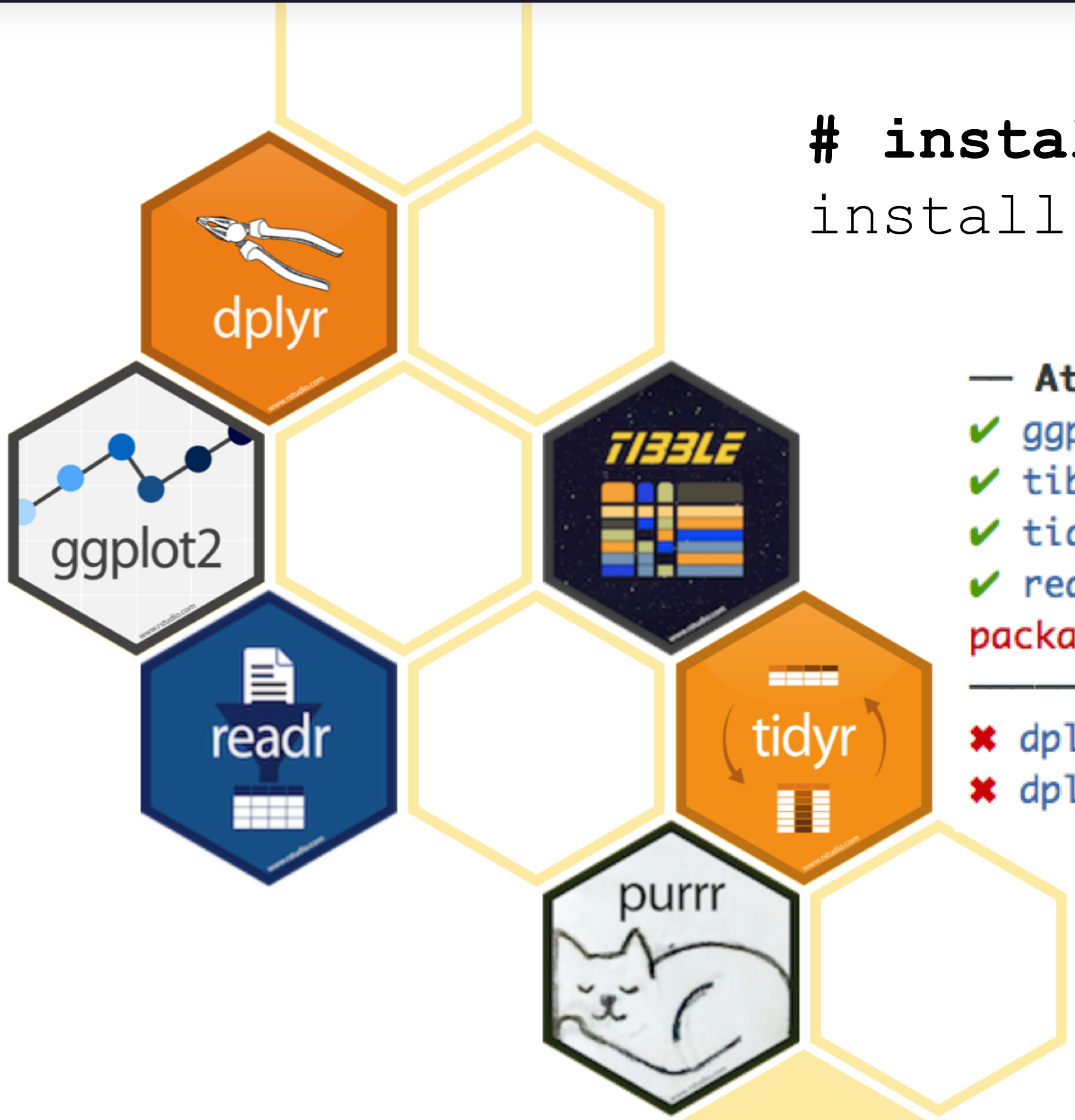
- filter()
- arrange()
- select()
- mutate()
- group_by()
- summarise()

Dataset used: Iris dataset, available in R.

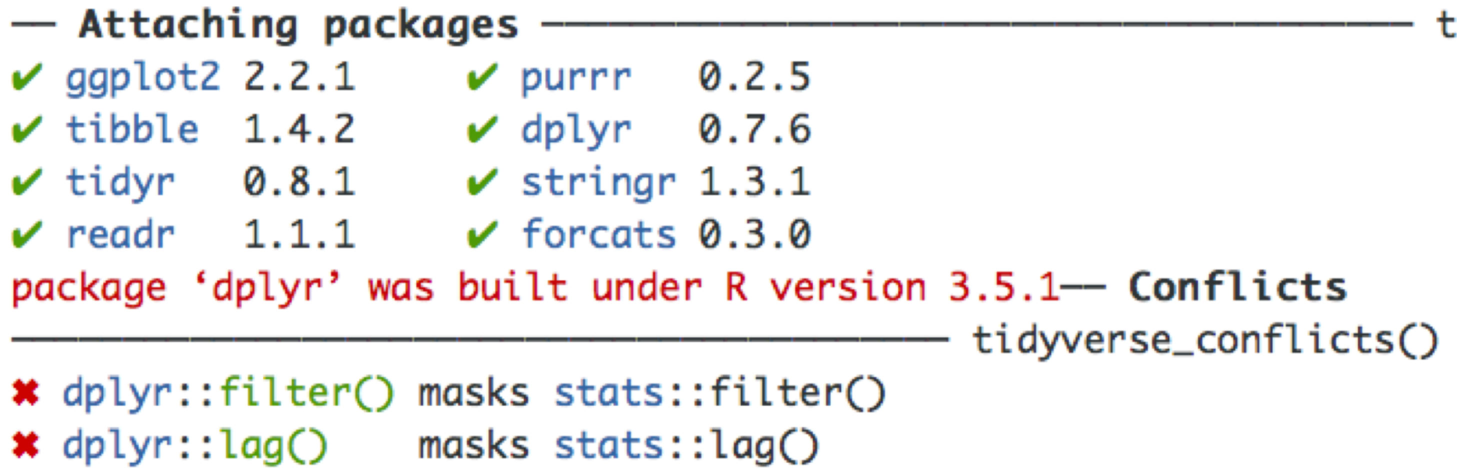
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa

str(iris)

## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1
1 1 1 1 ...
```

https://www.flickr.com/photos/33397993@N05

Tidyverse



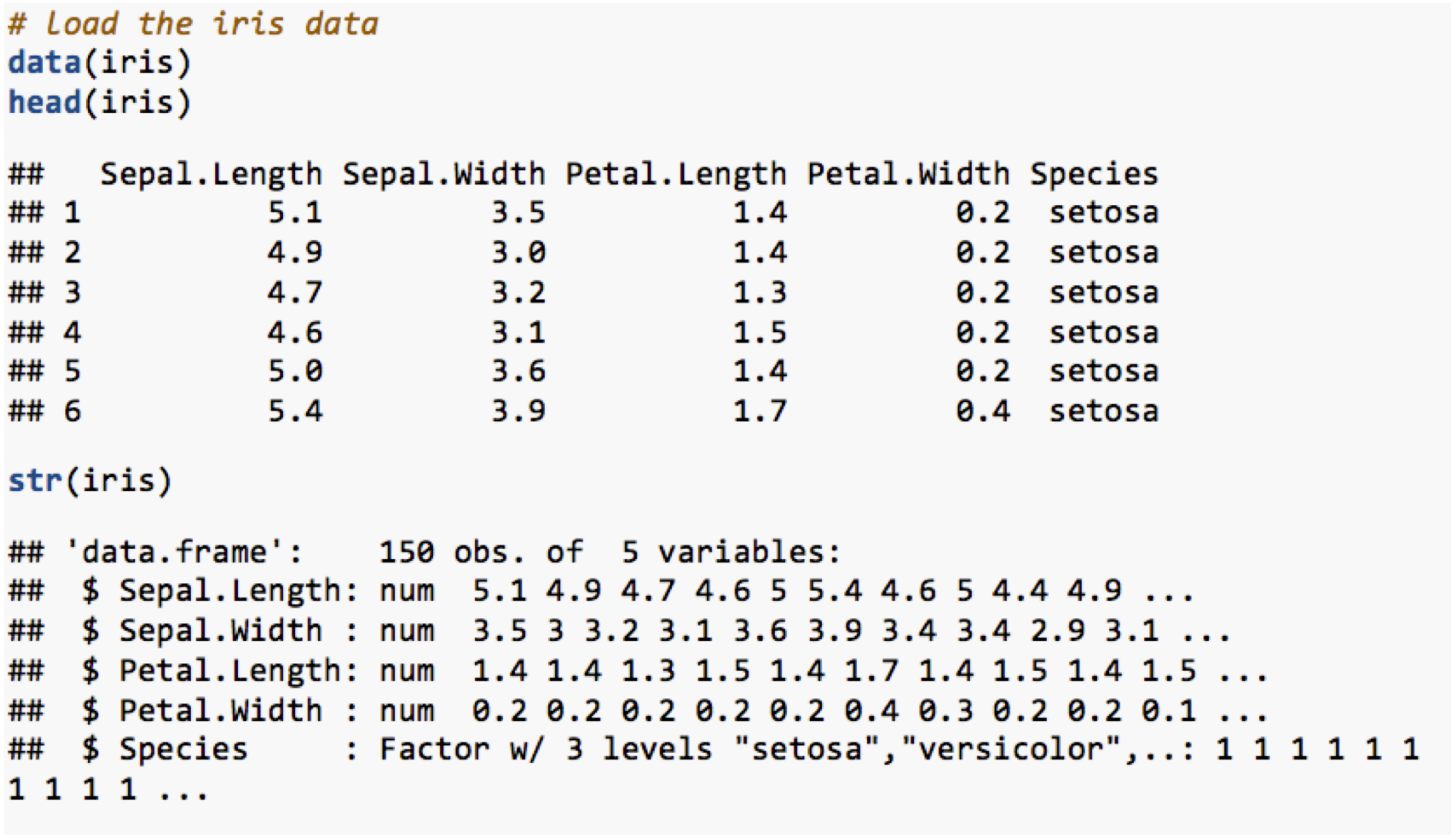
```
# install package
install.packages("tidyverse")
```



```
── Attaching packages ──────────────────────────────────────── t
✔ ggplot2 2.2.1     ✔ purrr   0.2.5
✔ tibble  1.4.2     ✔ dplyr   0.7.6
✔ tidyr   0.8.1     ✔ stringr 1.3.1
✔ readr   1.1.1     ✔ forcats 0.3.0
package ‘dplyr’ was built under R version 3.5.1── Conflicts
───────────────────────────────────────── tidyverse_conflicts()
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```

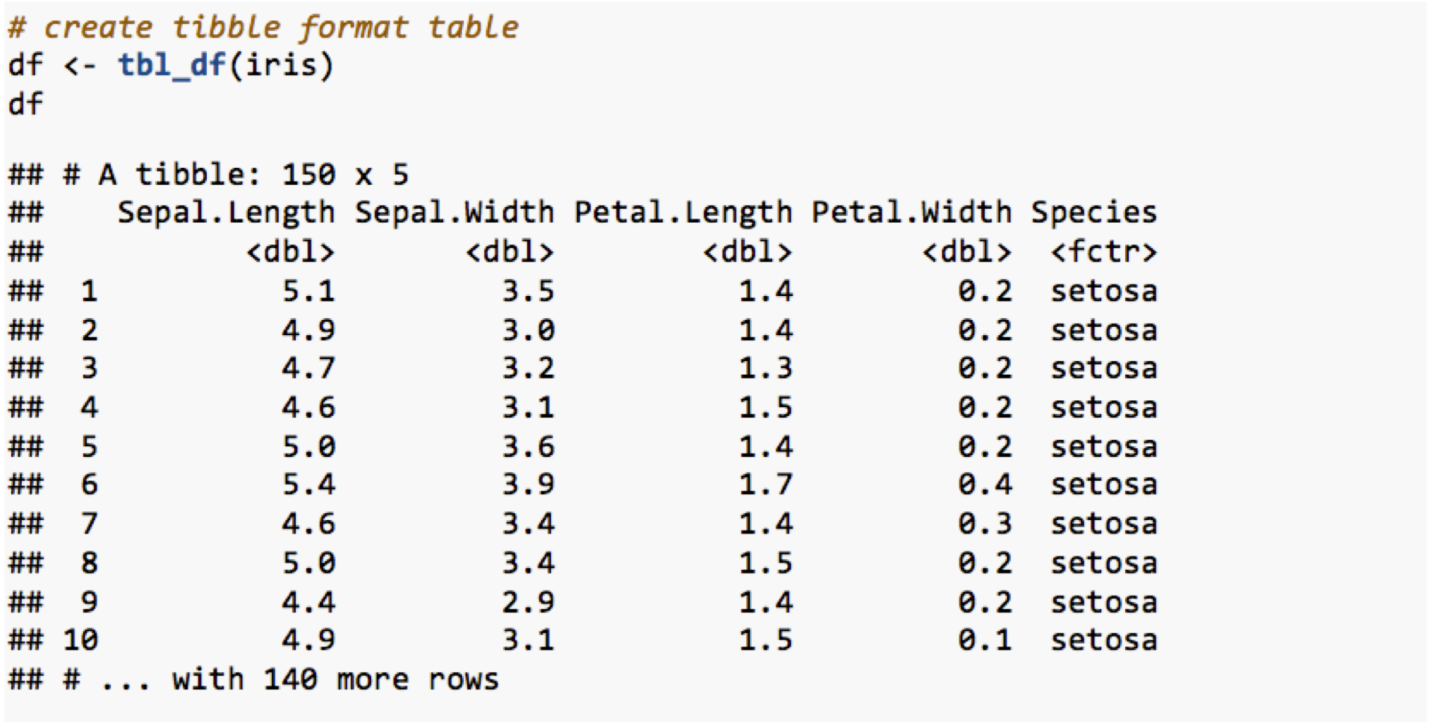
```
# load library
library(tidyverse)
```

```
# load iris data
data(iris)
head(iris)
str(iris)
```



```
# Load the iris data
data(iris)
head(iris)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa

str(iris)

## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1
1 1 1 1 ...
```

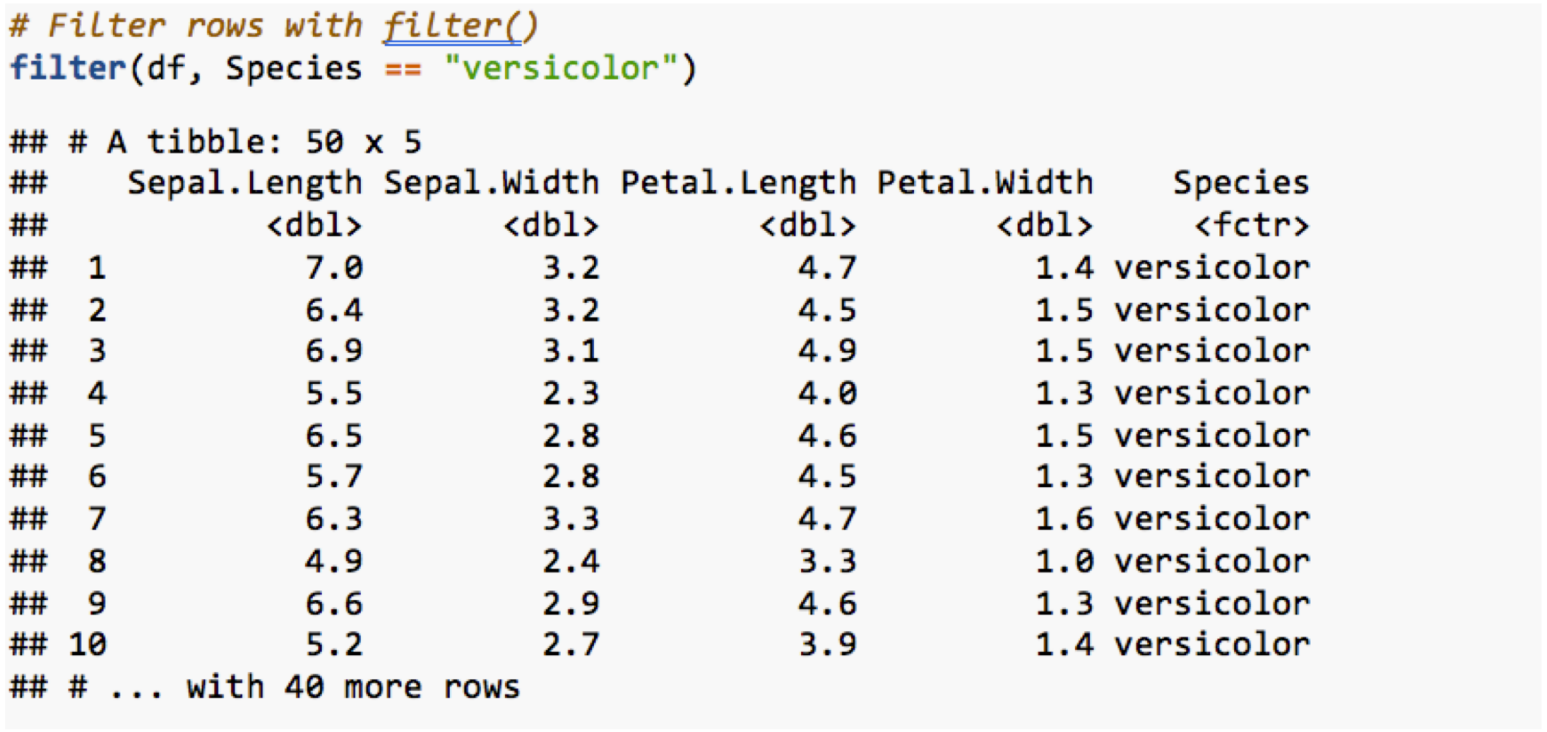
```
# create tibble format table
df <- tbl_df(iris)
df
```



```
# create tibble format table
df <- tbl_df(iris)
df

## # A tibble: 150 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           <dbl>       <dbl>        <dbl>       <dbl>  <fctr>
##  1          5.1         3.5          1.4         0.2  setosa
##  2          4.9         3.0          1.4         0.2  setosa
##  3          4.7         3.2          1.3         0.2  setosa
##  4          4.6         3.1          1.5         0.2  setosa
##  5          5.0         3.6          1.4         0.2  setosa
##  6          5.4         3.9          1.7         0.4  setosa
##  7          4.6         3.4          1.4         0.3  setosa
##  8          5.0         3.4          1.5         0.2  setosa
##  9          4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## # ... with 140 more rows
```

filter(): allows to subset observations based on their values

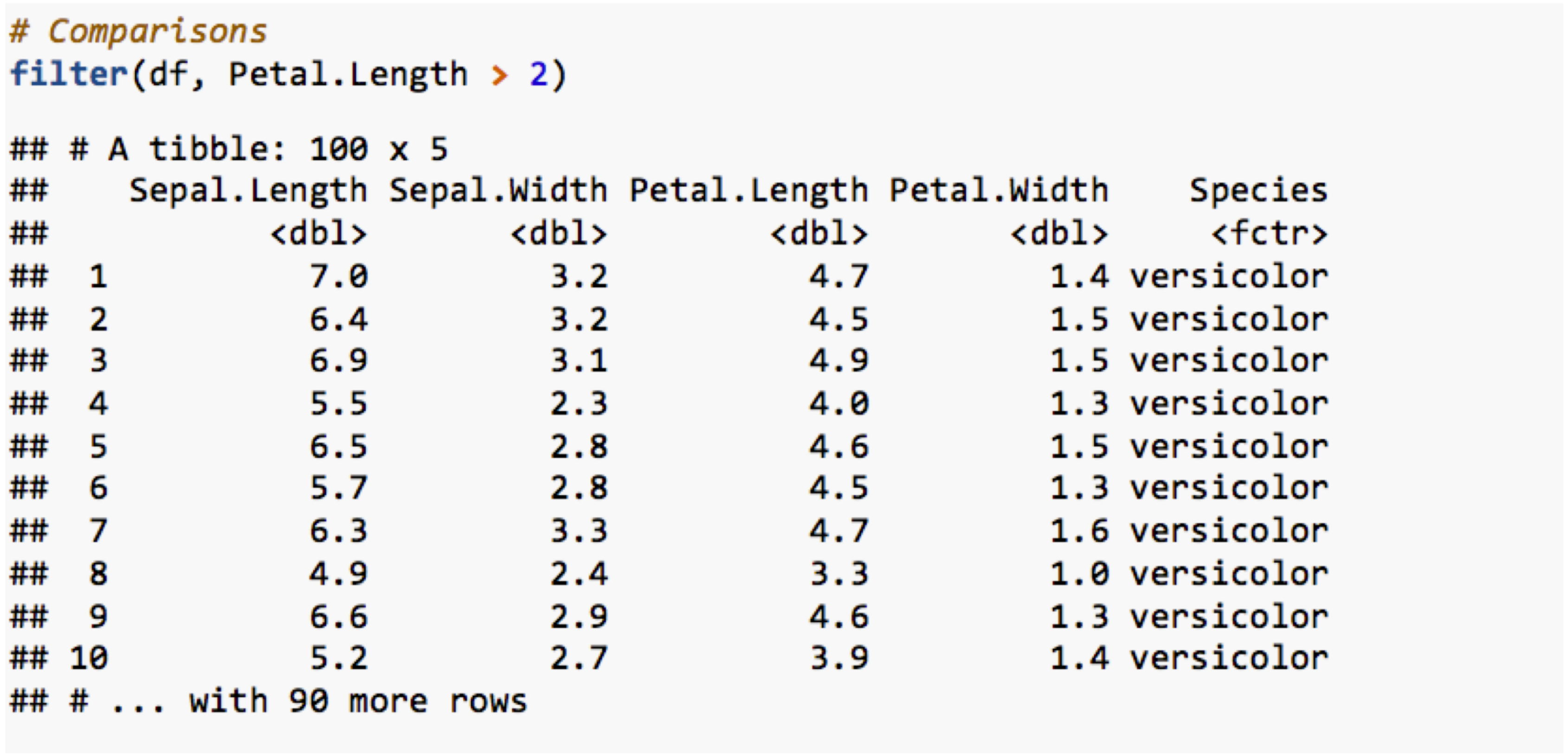
```
# Filter rows with filter()

filter(df, Species == "versicolor")
```



```
# Filter rows with filter()
filter(df, Species == "versicolor")

## # A tibble: 50 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
##           <dbl>       <dbl>        <dbl>       <dbl>     <fctr>
##  1          7.0         3.2          4.7         1.4 versicolor
##  2          6.4         3.2          4.5         1.5 versicolor
##  3          6.9         3.1          4.9         1.5 versicolor
##  4          5.5         2.3          4.0         1.3 versicolor
##  5          6.5         2.8          4.6         1.5 versicolor
##  6          5.7         2.8          4.5         1.3 versicolor
##  7          6.3         3.3          4.7         1.6 versicolor
##  8          4.9         2.4          3.3         1.0 versicolor
##  9          6.6         2.9          4.6         1.3 versicolor
## 10          5.2         2.7          3.9         1.4 versicolor
## # ... with 40 more rows
```

filter(): allows to subset observations based on their values

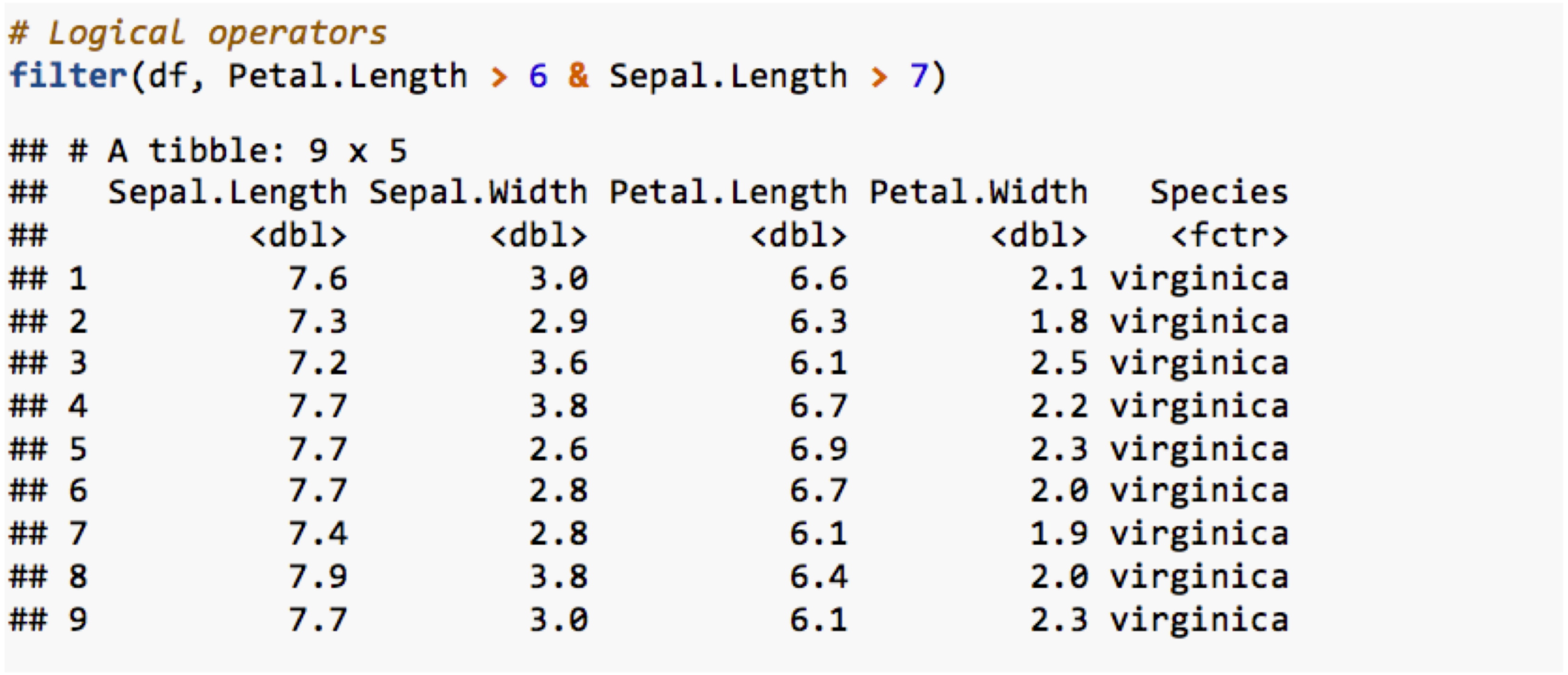
```
# Comparisons
filter(df, Petal.Length > 2)
```



```
# Comparisons
filter(df, Petal.Length > 2)

## # A tibble: 100 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
##           <dbl>       <dbl>        <dbl>       <dbl>     <fctr>
##  1          7.0         3.2          4.7         1.4 versicolor
##  2          6.4         3.2          4.5         1.5 versicolor
##  3          6.9         3.1          4.9         1.5 versicolor
##  4          5.5         2.3          4.0         1.3 versicolor
##  5          6.5         2.8          4.6         1.5 versicolor
##  6          5.7         2.8          4.5         1.3 versicolor
##  7          6.3         3.3          4.7         1.6 versicolor
##  8          4.9         2.4          3.3         1.0 versicolor
##  9          6.6         2.9          4.6         1.3 versicolor
## 10          5.2         2.7          3.9         1.4 versicolor
## # ... with 90 more rows
```

filter(): allows to subset observations based on their values

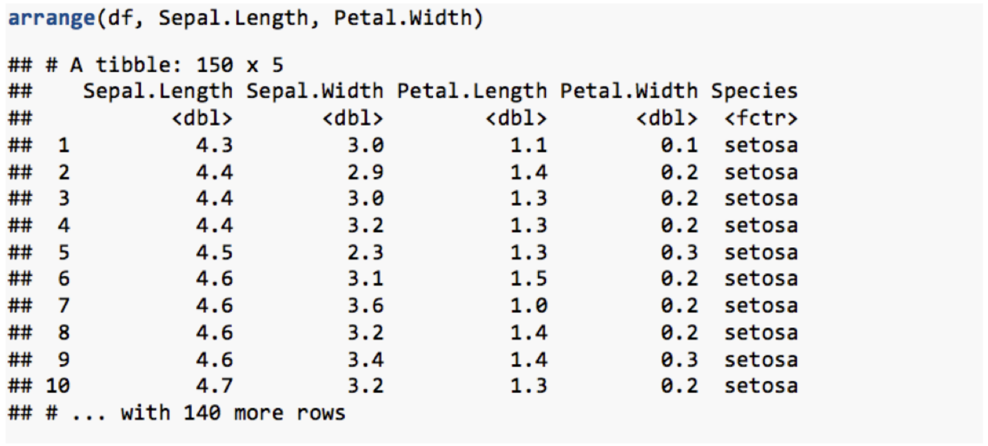
```
# Logical operators
filter(df, Petal.Length > 6 & Sepal.Length > 7)
```



```
# Logical operators
filter(df, Petal.Length > 6 & Sepal.Length > 7)

## # A tibble: 9 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
##          <dbl>       <dbl>        <dbl>       <dbl>    <fctr>
## 1          7.6         3.0          6.6         2.1 virginica
## 2          7.3         2.9          6.3         1.8 virginica
## 3          7.2         3.6          6.1         2.5 virginica
## 4          7.7         3.8          6.7         2.2 virginica
## 5          7.7         2.6          6.9         2.3 virginica
## 6          7.7         2.8          6.7         2.0 virginica
## 7          7.4         2.8          6.1         1.9 virginica
## 8          7.9         3.8          6.4         2.0 virginica
## 9          7.7         3.0          6.1         2.3 virginica
```

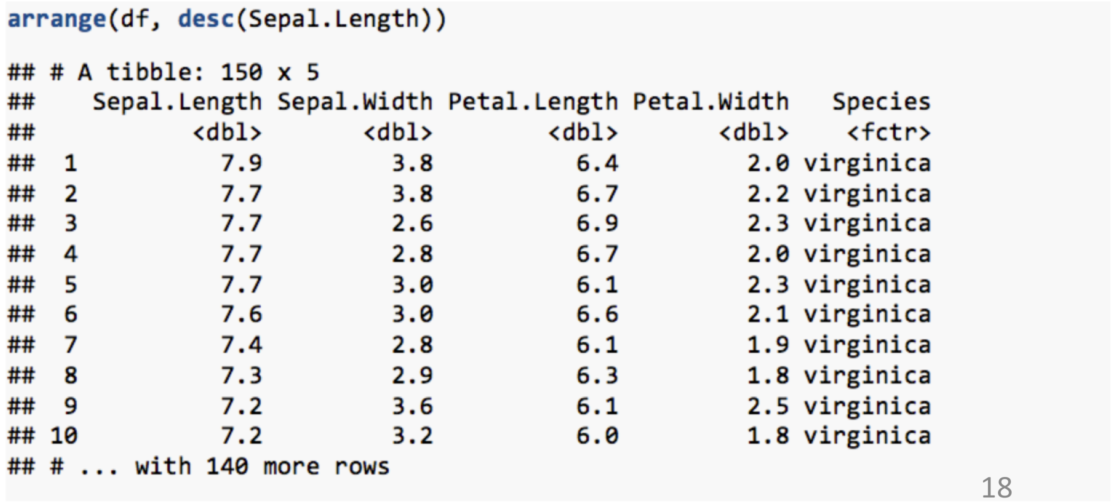
arrange(): works similarly to filter() except that instead of selecting rows, it changes their order.

```
# default is ascending order
arrange(df, Sepal.Length, Petal.Width)
```



```
arrange(df, Sepal.Length, Petal.Width)

## # A tibble: 150 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           <dbl>       <dbl>        <dbl>       <dbl>  <fctr>
##  1          4.3         3.0          1.1         0.1  setosa
##  2          4.4         2.9          1.4         0.2  setosa
##  3          4.4         3.0          1.3         0.2  setosa
##  4          4.4         3.2          1.3         0.2  setosa
##  5          4.5         2.3          1.3         0.3  setosa
##  6          4.6         3.1          1.5         0.2  setosa
##  7          4.6         3.6          1.0         0.2  setosa
##  8          4.6         3.2          1.4         0.2  setosa
##  9          4.6         3.4          1.4         0.3  setosa
## 10          4.7         3.2          1.3         0.2  setosa
## # ... with 140 more rows
```

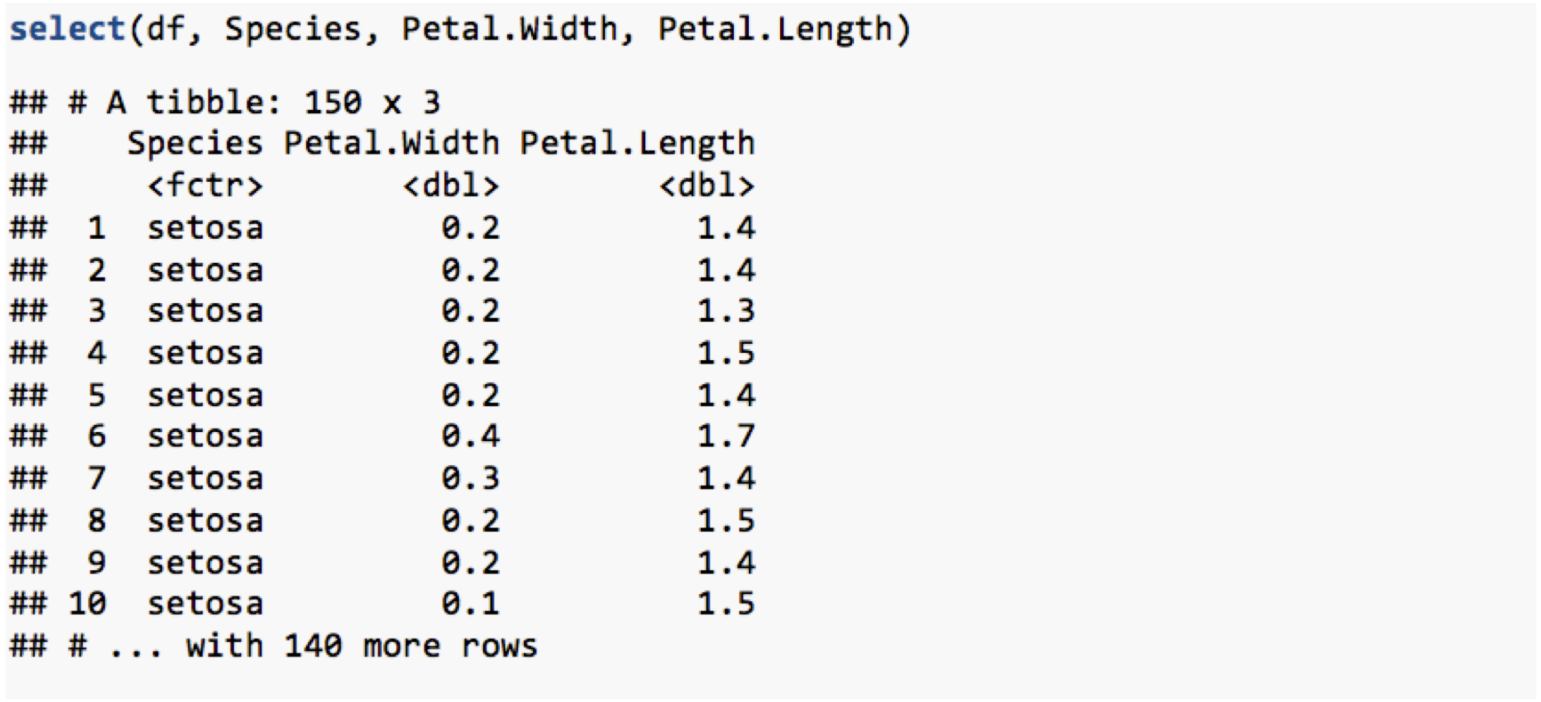
```
# to order in descending order
arrange(df, desc(Sepal.Length))
```



```
arrange(df, desc(Sepal.Length))

## # A tibble: 150 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
##           <dbl>       <dbl>        <dbl>       <dbl>    <fctr>
##  1          7.9         3.8          6.4         2.0 virginica
##  2          7.7         3.8          6.7         2.2 virginica
##  3          7.7         2.6          6.9         2.3 virginica
##  4          7.7         2.8          6.7         2.0 virginica
##  5          7.7         3.0          6.1         2.3 virginica
##  6          7.6         3.0          6.6         2.1 virginica
##  7          7.4         2.8          6.1         1.9 virginica
##  8          7.3         2.9          6.3         1.8 virginica
##  9          7.2         3.6          6.1         2.5 virginica
## 10          7.2         3.2          6.0         1.8 virginica
## # ... with 140 more rows
```

select(): select columns and allows to zoom in on a subset of data based on the names of the variables

```
# subsetting columns of interest
select(df, Species, Petal.Width, Petal.Length)
```

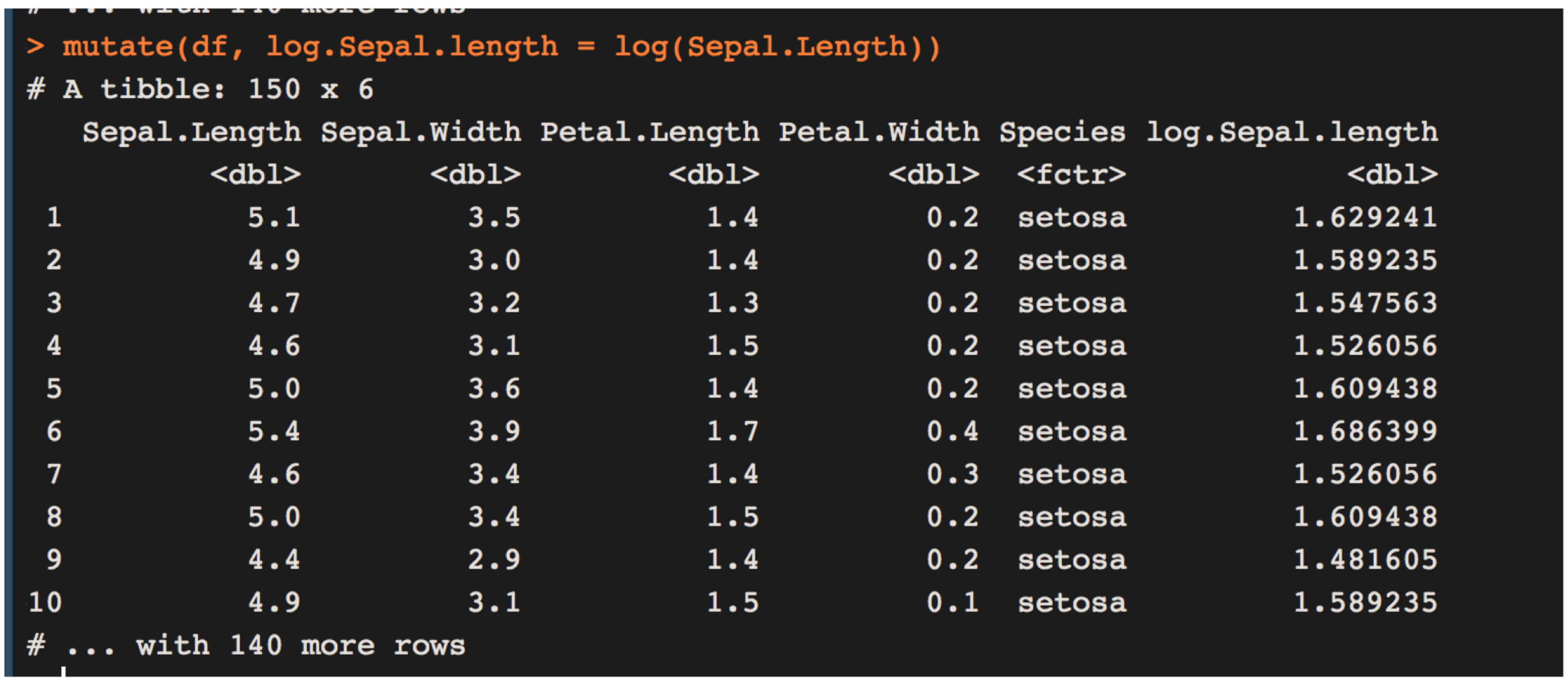


```
select(df, Species, Petal.Width, Petal.Length)

## # A tibble: 150 x 3
##    Species Petal.Width Petal.Length
##     <fctr>       <dbl>        <dbl>
##  1  setosa         0.2          1.4
##  2  setosa         0.2          1.4
##  3  setosa         0.2          1.3
##  4  setosa         0.2          1.5
##  5  setosa         0.2          1.4
##  6  setosa         0.4          1.7
##  7  setosa         0.3          1.4
##  8  setosa         0.2          1.5
##  9  setosa         0.2          1.4
## 10  setosa         0.1          1.5
## # ... with 140 more rows
```

mutate() : allows to add new column at the end of dataset

# Create a new column with additional information

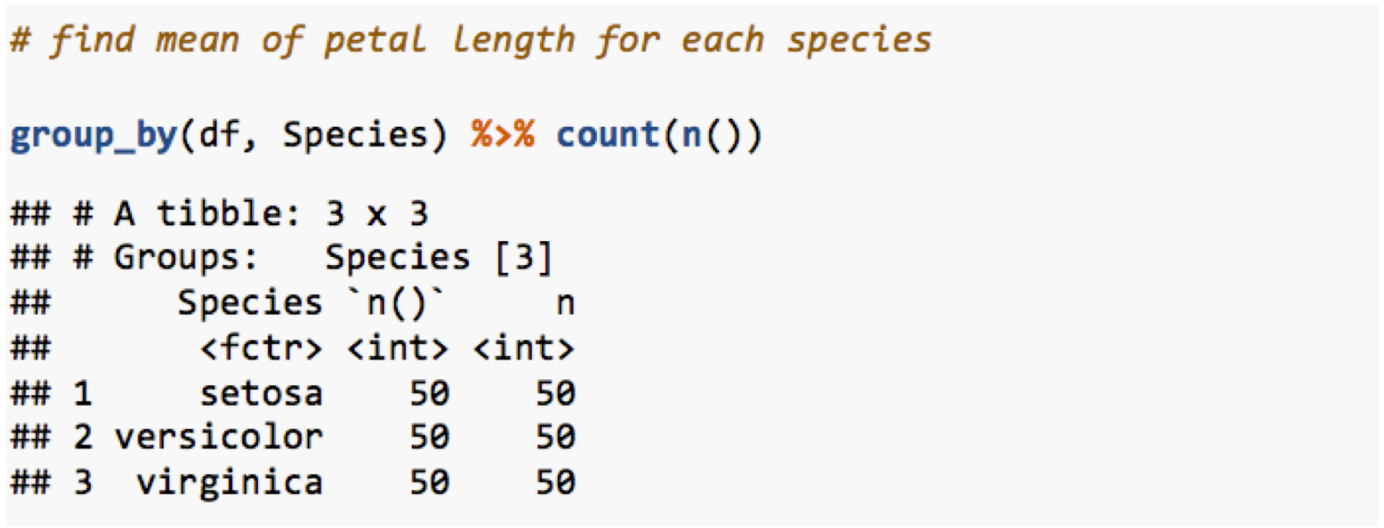
```
mutate(df, log.Sepal.length = log(Sepal.Length))
```



```
> mutate(df, log.Sepal.length = log(Sepal.Length))
# A tibble: 150 x 6
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species log.Sepal.length
          <dbl>       <dbl>        <dbl>       <dbl>  <fctr>            <dbl>
 1          5.1         3.5          1.4         0.2  setosa         1.629241
 2          4.9         3.0          1.4         0.2  setosa         1.589235
 3          4.7         3.2          1.3         0.2  setosa         1.547563
 4          4.6         3.1          1.5         0.2  setosa         1.526056
 5          5.0         3.6          1.4         0.2  setosa         1.609438
 6          5.4         3.9          1.7         0.4  setosa         1.686399
 7          4.6         3.4          1.4         0.3  setosa         1.526056
 8          5.0         3.4          1.5         0.2  setosa         1.609438
 9          4.4         2.9          1.4         0.2  setosa         1.481605
10          4.9         3.1          1.5         0.1  setosa         1.589235
# ... with 140 more rows
```

group_by(): allows to group / roll up dataset by multiple variables

```
# group dataset by species, and display the number of entries
# for each species of flower
group_by(df, Species) %>% count(n())
```

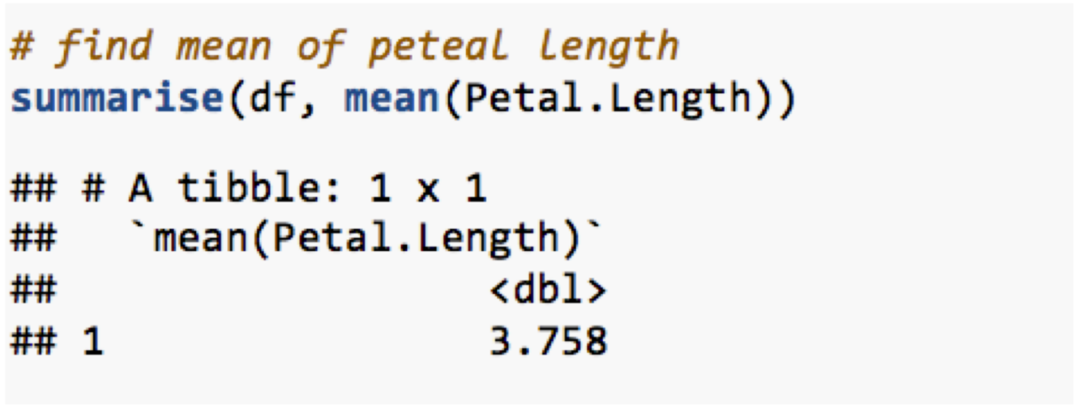


```
# find mean of petal length for each species

group_by(df, Species) %>% count(n())

## # A tibble: 3 x 3
## # Groups:   Species [3]
##      Species `n()`     n
##       <fctr> <int> <int>
## 1     setosa    50    50
## 2 versicolor    50    50
## 3  virginica    50    50
```

%>%: Pipe function in R- allows to pass the output from one operation as input to the next, without need to create object at each step

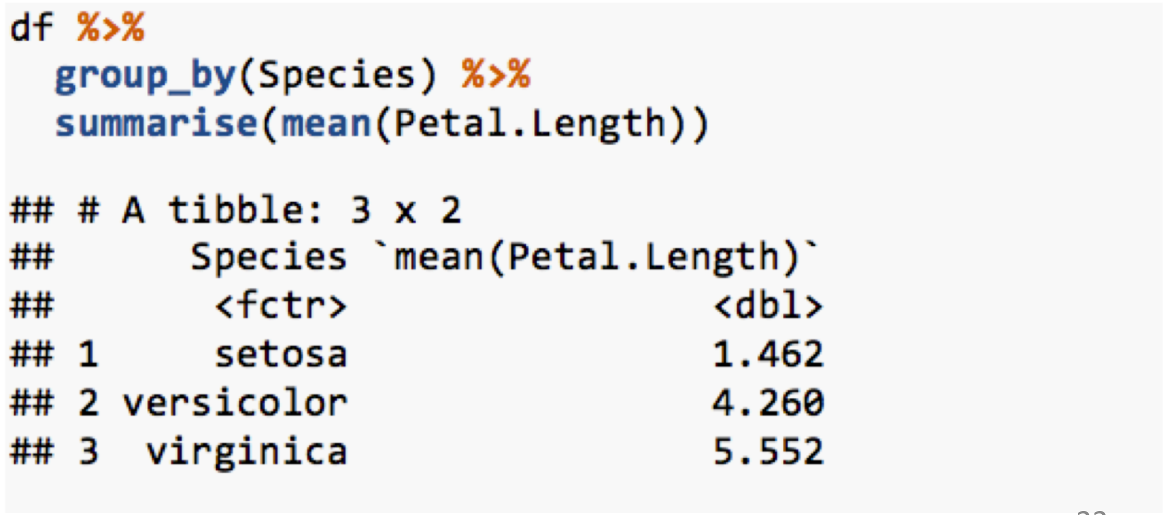
summarise() : provides summary information

```
# mean of petal length

summarise(df, mean(Petal.Length))
```



```
# find mean of peteal length
summarise(df, mean(Petal.Length))

## # A tibble: 1 x 1
##   `mean(Petal.Length)`
##                  <dbl>
## 1                3.758
```

```
# mean of petal length for each species

df %>%
  group_by(Species) %>%
  summarise(mean(Petal.Length))
```

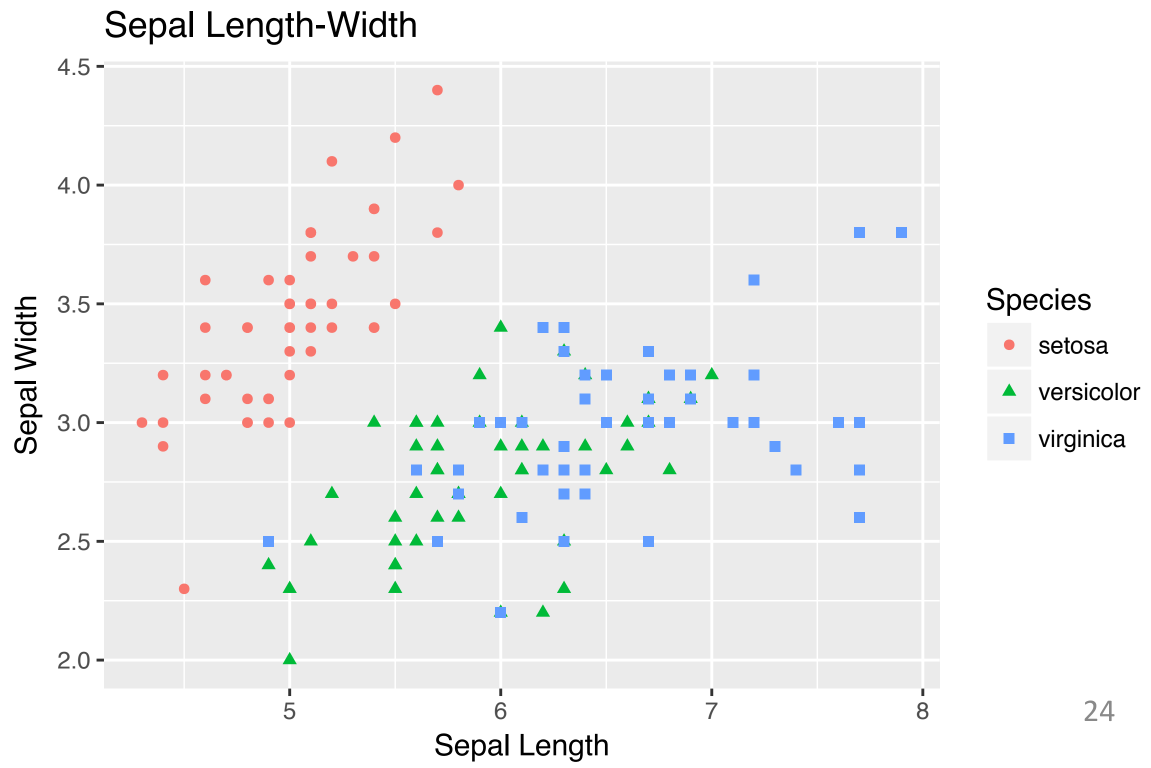


```
df %>%
  group_by(Species) %>%
  summarise(mean(Petal.Length))

## # A tibble: 3 x 2
##      Species `mean(Petal.Length)`
##       <fctr>                <dbl>
## 1     setosa                1.462
## 2 versicolor                4.260
## 3  virginica                5.552
## Question: create a summary table containing sample size,
mean petal length, and mean petal width, arranging the
output in descending order of mean petal length.
```

Frequently Used Plots

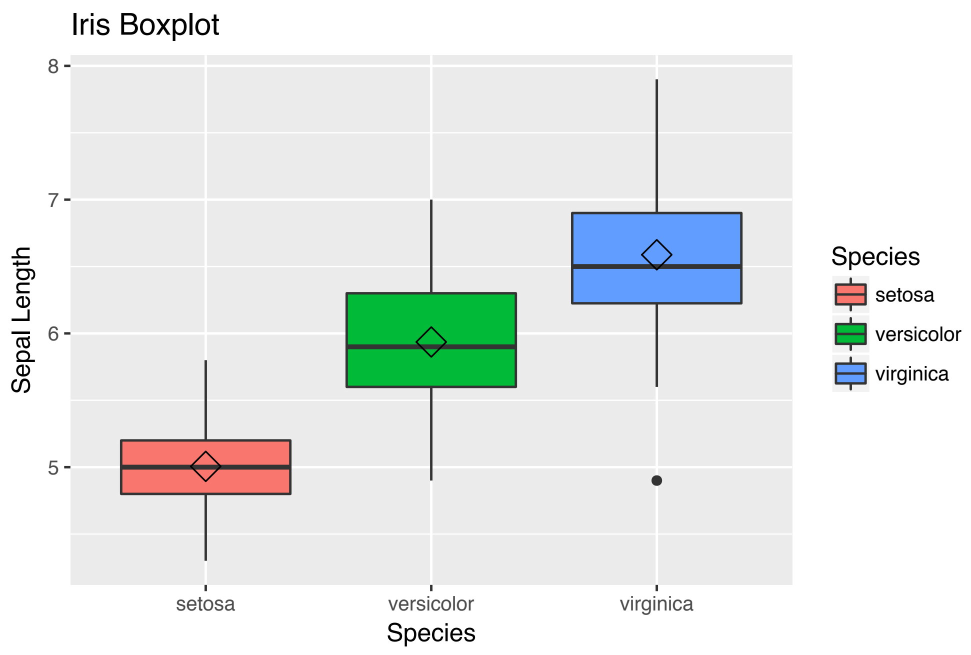
1) Scatterplot

```
ggplot(data=df, aes(x = Sepal.Length, y = Sepal.Width))+
  geom_point(aes(color=Species, shape=Species)) +
  xlab("Sepal Length") +
  ylab("Sepal Width") +
  ggtitle("Sepal Length-Width")
```



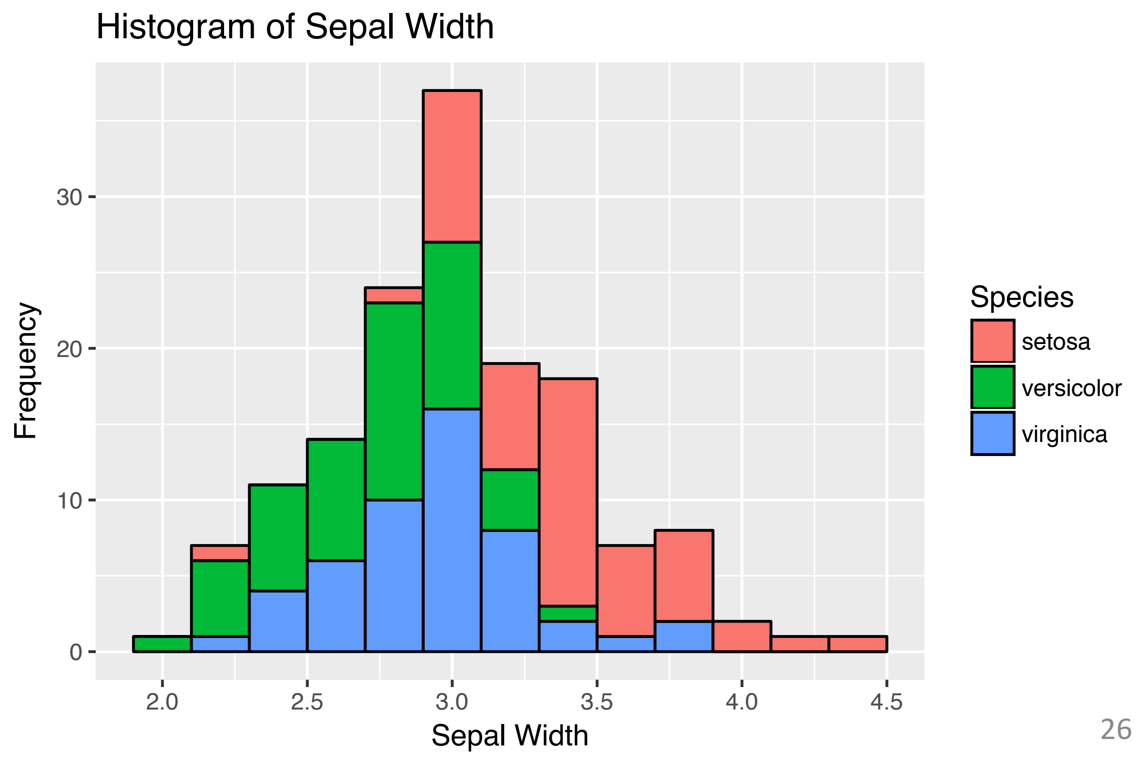
2) Box Plot

```
box <- ggplot(data=df, aes(x=Species,
y=Sepal.Length))

box +
  geom_boxplot(aes(fill=Species)) +
  ylab("Sepal Length") +
  ggtitle("Iris Boxplot") +
  stat_summary(fun.y=mean, geom="point",
shape=5, size=4)
```



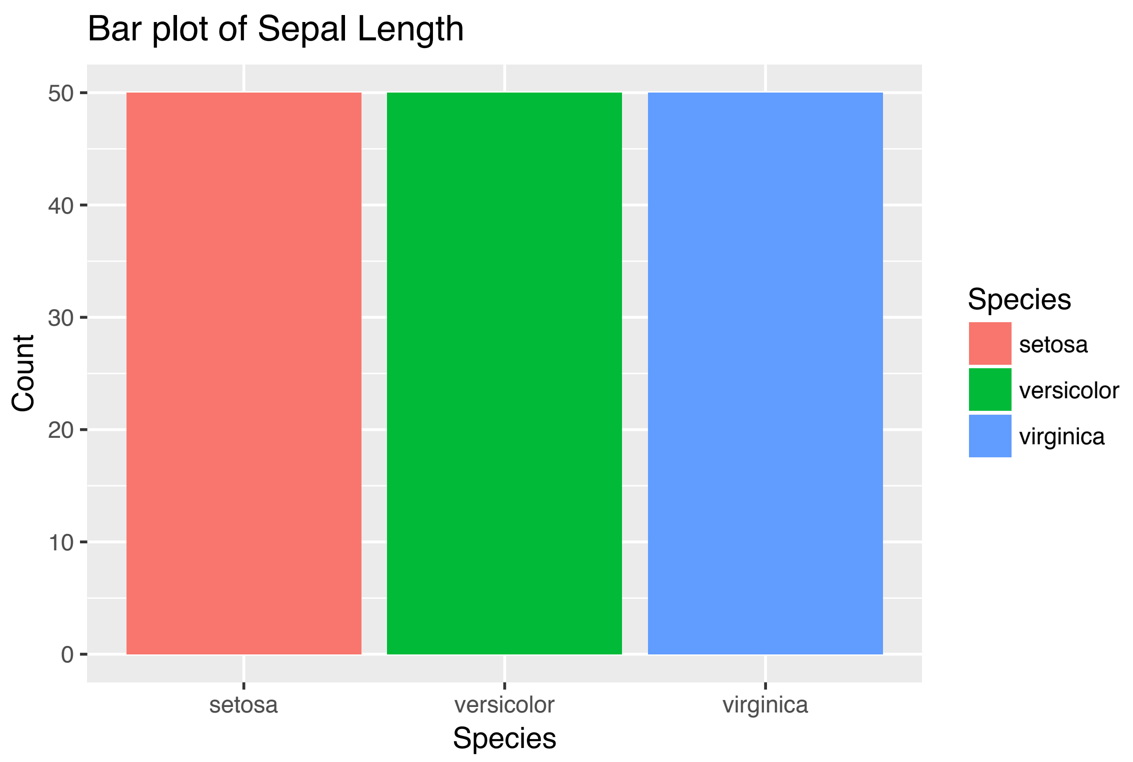
# 3) Histogram

```
histogram <- ggplot(data=df, aes(x=Sepal.Width))

histogram +
  geom_histogram(binwidth=0.2, color="black", aes(fill=Species)) +
  xlab("Sepal Width") +
  ylab("Frequency") +
  ggtitle("Histogram of Sepal Width")
```



# 4) bar plot

```
bar <- ggplot(data=df, aes(x=Species))

bar + geom_bar(aes(fill=Species)) +
xlab("Species") +
  ylab("Count") +
  ggtitle("Bar plot of Sepal Length")
```

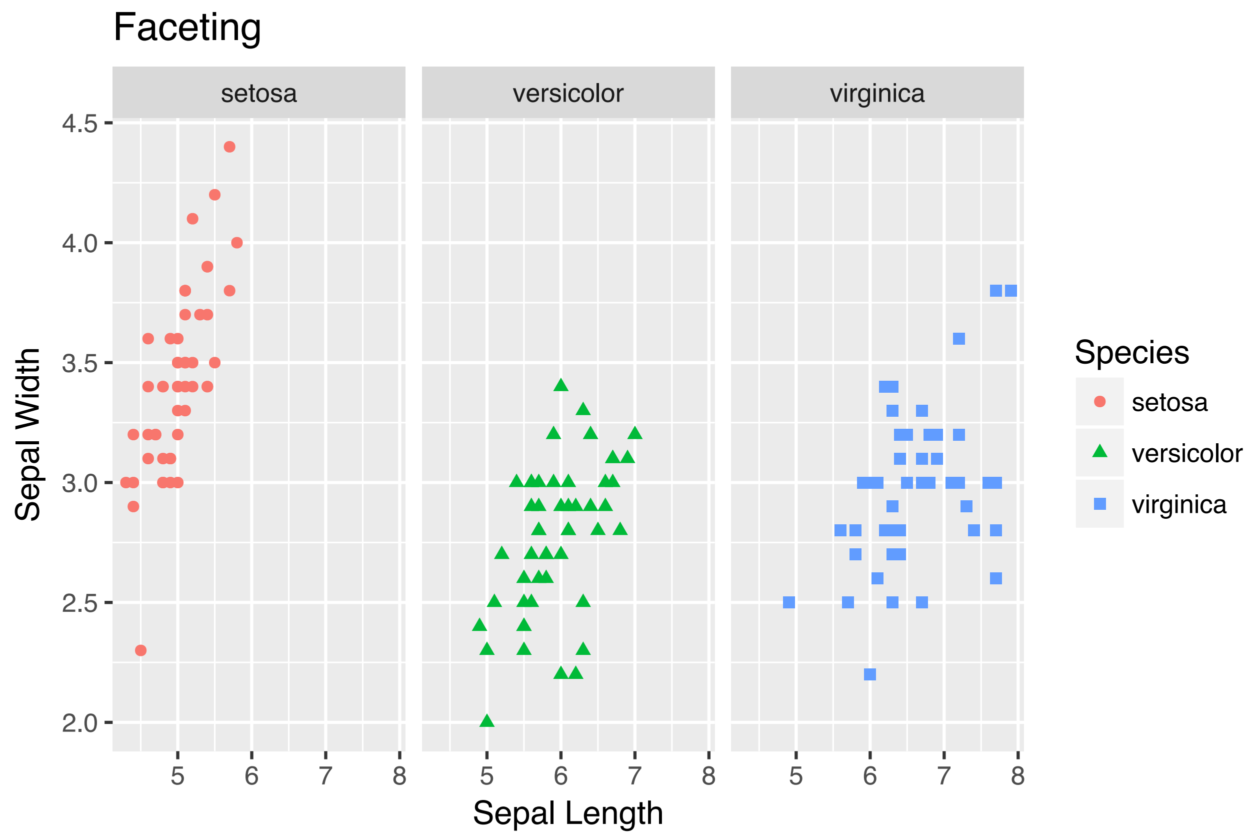


# 5) Faceting

```
facet <- ggplot(data=df, aes(Sepal.Length, y=Sepal.Width,
color=Species)) +
  geom_point(aes(shape=Species), size=1.5) +
  xlab("Sepal Length") +
  ylab("Sepal Width") +
  ggtitle("Faceting")
```

# Along columns

```
facet + facet_grid(. ~ Species)
```


Faceting
setosa
versicolor
virginica
Sepal Width
Sepal Length
Species
● setosa
▲ versicolor
■ virginica