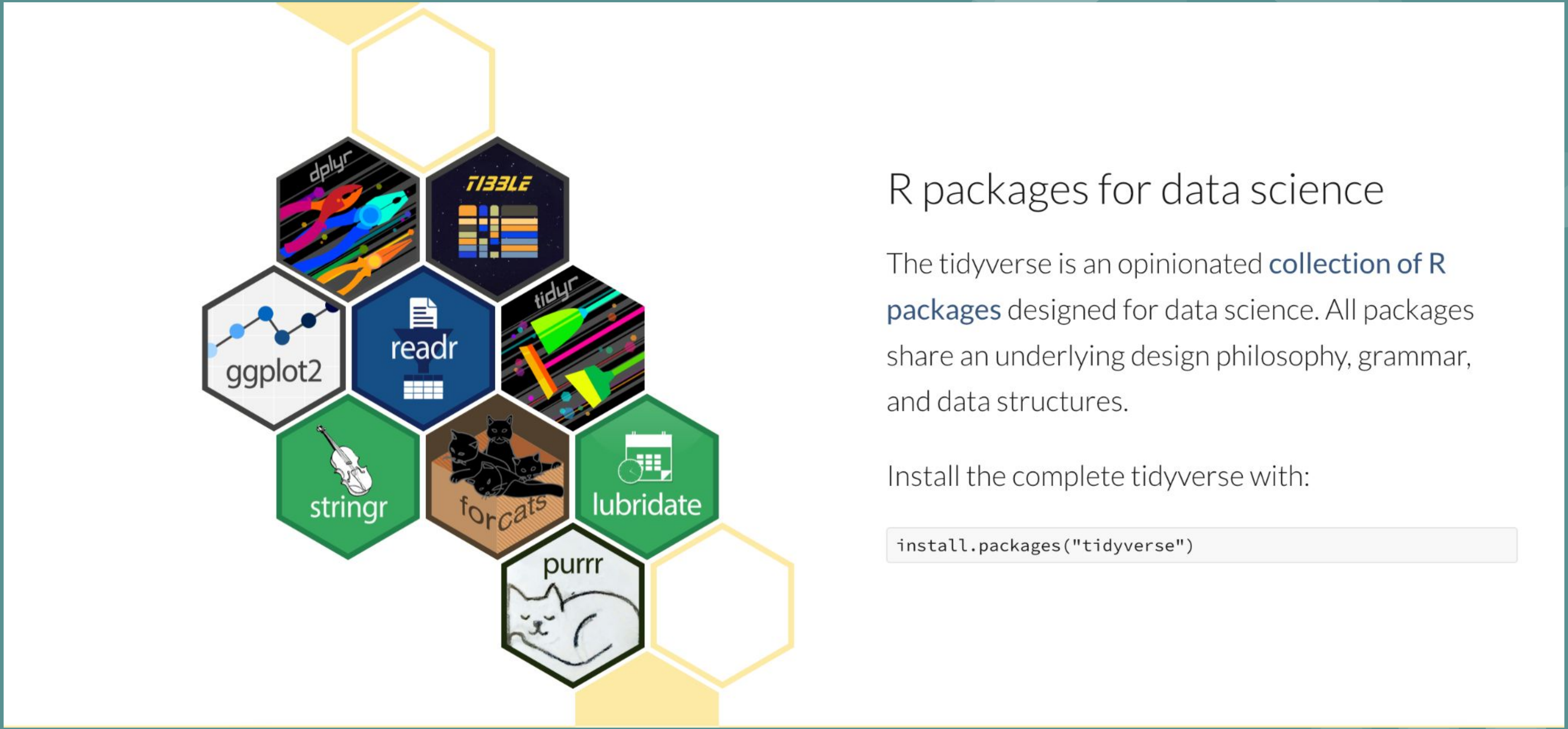
Intro to R for biologists Day 2

Working with data using Tidyverse tools
Data transformations and computations
Hypothesis testing with statistics

Brandon Ely, Doctoral Candidate
PhD program in Biology (Molecular, Cellular, Developmental)
CUNY Graduate Center

*Adapted from Dr. Weigang Qiu's "R Tutorials for biologists"

The Tidyverse R package



R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

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

Uploading a data table

R has several functions designed to open data tabled stored in various file formats

```
df <- read.table(file, header, sep, ...)
```

file: path to the file (or just file name if its in working directory)

header: TRUE or FALSE (specifies if first row are column names)

sep: specifies the delimiter for the file (comma = “,” or tab = “\t”)

Use `df$ColumnName` to reference specific col, use [ ] for index

Upload the “compensation” dataset

```
df <-
read.table('http://wiki.genometracker.org/~weigang/datas
ets-master/compensation.csv',
sep = ',',
header = T)
```

Looking at your data

Below are functions to just check out your data. What does each do or tell you?

dim(df)

nrow(df)

ncol(df)

names(df)

head(df) *or* tail(df)

glimpse(df)

Subsetting and sorting your data

select function → select or deselect columns from your dataframe

`select(df, col_name)` , `select(df, -col_name)`, `select(df, 2:5))`

slice functions → subset a select set of rows

`slice(df, n)` **n can be a single row or vector of rows like 5:20*

Other slice functions:

slice_head(), slice_tail(), slice_sample(), slice_max(), slice_min()

arrange function → sorts rows by a specified variable
`arrange(df, variable)` or try `arrange(df, desc(variable))`

Filtering your dataframe

filter subsets rows from your dataframe based on a specified condition

`filter(df, condition)`

For conditions, use operators:

== (equivalent)	`filter(df, Treatment == “drug”)`
> (greater than)	`filter(df, height > 65)`
< (less than)	
& (and)	`filter(df, height > 75 & weight == 185)`
| (or)	`filter(df, Treatment == “control” | sleep < 1)`

Piping functions for speed and readability

Functions within tidyverse can be linked together and executed in order

Use `%>%` or `|>` to link functions

**you don’t enter the dataframe as an arg in the tidy functions*

```
new_df <- df %>% filter(treatment == “drug”) %>% arrange(height)
```

You can hit enter after each pipe to help readability:

```
new_df <- df %>%
                filter(treatment == “drug”) %>%
                arrange(height)
```

Computation within the dataframe

Columns in your dataframe are vectors, so any operation you can perform on a vector (or set of vectors) you can do on columns in your dataframe

mean(df$height) or median(df$height): both return single numeric value

log2(df$height) or df$height / 2: returns a vector of numeric values

Create a new column in your dataframe with a computation:
`mutate(df, log_height = log(height))` *computation on single col

`mutate(df, combined_score = test_1 + test_2)` *comp on multi cols

Statistics

summary function → provides summary statistics for your data

```
summary(df)
```

summarise function → outputs df with columns of computations

```
summarise(df, mean_height = mean(height))
```

group_by function → groups data based on specified categorical variable; VERY HELPFUL in combination with summarise function

```
df %>%
    group_by(‘treatment’) %>%
    summarise(mean_height = mean(height), sd_height = sd(height)
```

Practice: The Iris dataset

Use command data(iris) to load dataset into environment

1. Find the dimensions of the dataset
2. List the variables and their data types (Hint: run `glimpse("iris")`)
3. Summarize the variables (Hint: run `summary("iris")`)
4. Get the last 10 observations of the dataset
5. Select only the first four columns (remove the "`species`" column)
6. Filter rows by species, retain only rows from one species (e.g., "`setosa`")
7. Filter rows by a cutoff value (e.g., "`Sepal.Length >= 4`")
8. Add a column by taking the log10 of "`Sepal.Lengh`"
9. What are the medians of the variable "`Sepal.Length`" for each species?
10. Count how many samples for each species
11. Convert the table to long format

Hypothesis testing

What are some hypotheses you can test with the compensation dataset?