HBC
Harvard Chan Bioinformatics Core

Introduction to differential gene expression (DGE) analysis

https://tinyurl.com/hbc-dge

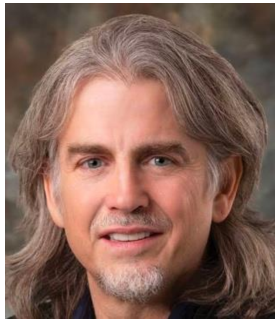
Harvard Chan Bioinformatics Core

Introductions!

Shannan Ho Sui
Director

Meeta Mistry
Associate Director

Lorena Pantano
Director of Bioinformatics Platform



John Quackenbush
Faculty Advisor

Upen Bhattarai

Heather Wick

Will Gammerdinger

Noor Sohail

Alex Bartlett

Elizabeth Partan

Emma Berdan

James Billingsley

Zhu Zhuo

Maria Simoneau


Shannan Ho Sui
Director


Meeta Mistry
Associate Director


Lorena Pantano
Director of Bioinformatics Platform


John Quackenbush
Faculty Advisor


Upen Bhattarai


Heather Wick


Will Gammerdinger


Noor Sohail


Alex Bartlett


Elizabeth Partan


Emma Berdan

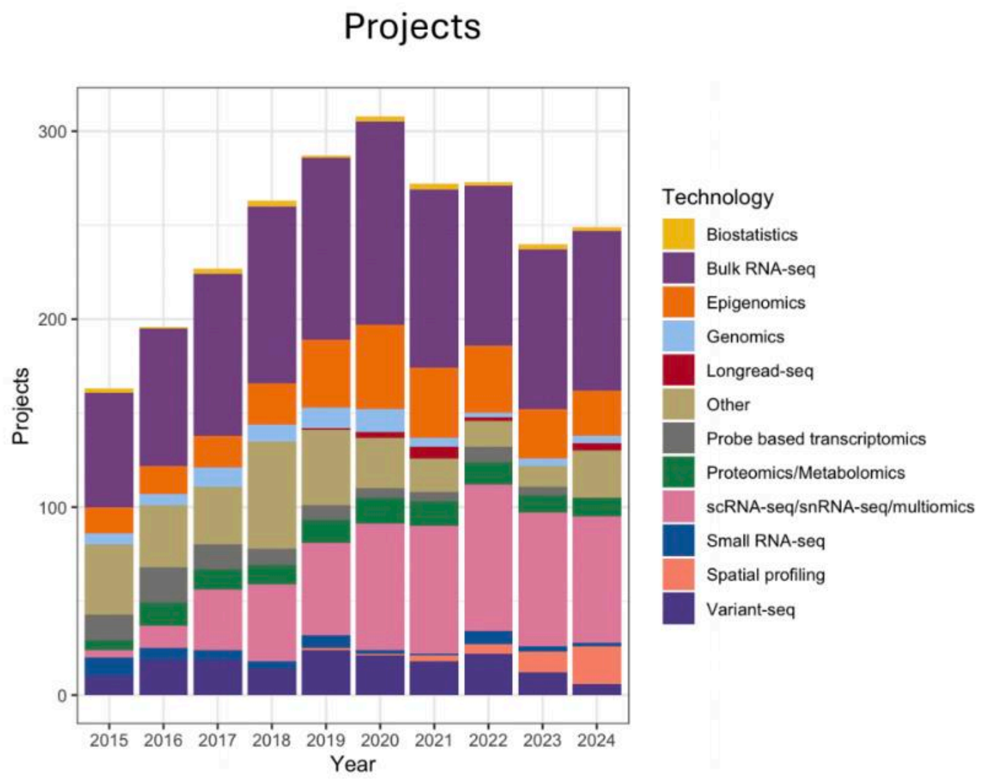

James Billingsley


Zhu Zhuo


Maria Simoneau

Consulting

- Transcriptomics: Bulk, single cell, small RNA
- Epigenomics: ChIP-seq, CUT&RUN, ATAC-seq, DNA methylation
- Variant discovery: WGS, resequencing, exome-seq and CNV
- Multiomics integration
- Spatial biology
- Experimental design and grant support



https://bioinformatics.sph.harvard.edu/services

Consulting

- Transcriptomics: Bulk, single cell, small RNA
- Epigenomics: ChIP-seq, CUT&RUN, ATAC-seq, DNA methylation
- Variant discovery: WGS, resequencing, exome-seq and CNV
- Multiomics integration
- Spatial biology
- Experimental design and grant support

CFAR
HARVARD UNIVERSITY
CENTER FOR AIDS RESEARCH

HARVARD
T.H. CHAN
SCHOOL OF PUBLIC HEALTH

NIEHS

HARVARD
CATALYST
THE HARVARD CLINICAL AND TRANSLATIONAL SCIENCE CENTER

HARVARD
MEDICAL SCHOOL

Training

- Hands-on workshops design to reflect best practices, reproducibility and an emphasis on experimental design
  - Basic Data Skills
    - Shell
    - R
  - Advanced Topics: Analysis of high-throughput sequencing data
    - Chromatin Biology
    - Bulk RNA-seq
    - Differential Gene Expression
    - scRNA-seq
    - Variant Calling
  - Current Topics in Bioinformatics

https://bioinformatics.sph.harvard.edu/training

Training

- Hands-on workshops design to reflect best practices, reproducibility and an emphasis on experimental design
  - Basic Data Skills
    - Shell
    - R
  - Advanced Topics: Analysis of high-throughput sequencing data
    - Chromatin Biology
    - Bulk RNA-seq
    - Differential Gene Expression
    - scRNA-seq
    - Variant Calling
  - Current Topics in Bioinformatics

https://bioinformatics.sph.harvard.edu/training

CFAR
HARVARD UNIVERSITY
CENTER FOR AIDS RESEARCH

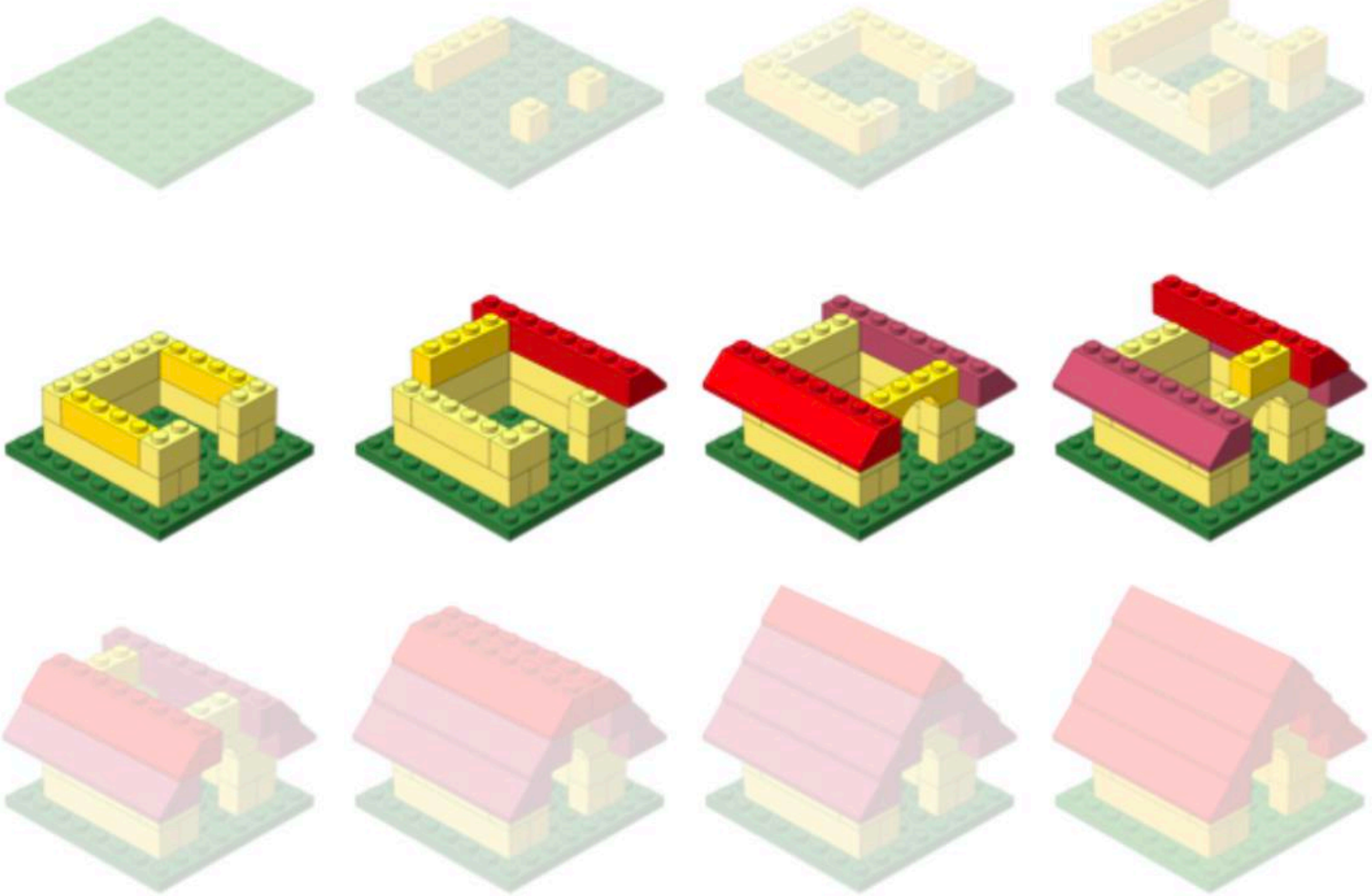
HARVARD
T.H. CHAN
SCHOOL OF PUBLIC HEALTH

DF/HCC
DANA-FARBER / HARVARD CANCER CENTER

HARVARD
CATALYST
THE HARVARD CLINICAL
AND TRANSLATIONAL
SCIENCE CENTER

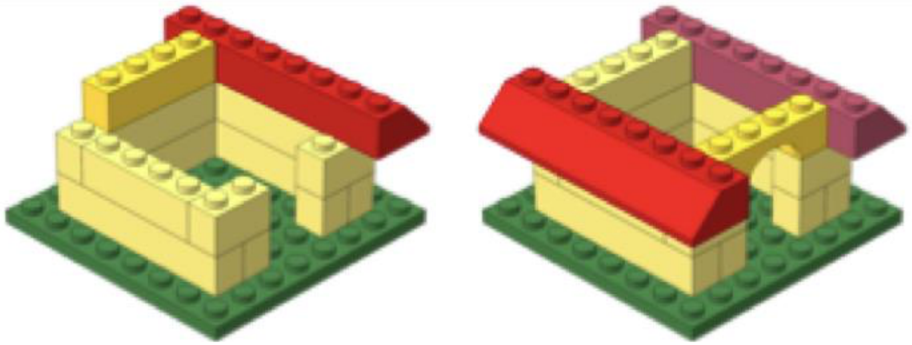
HARVARD
MEDICAL SCHOOL

Workshop scope



Bioinformatics Data Analysis

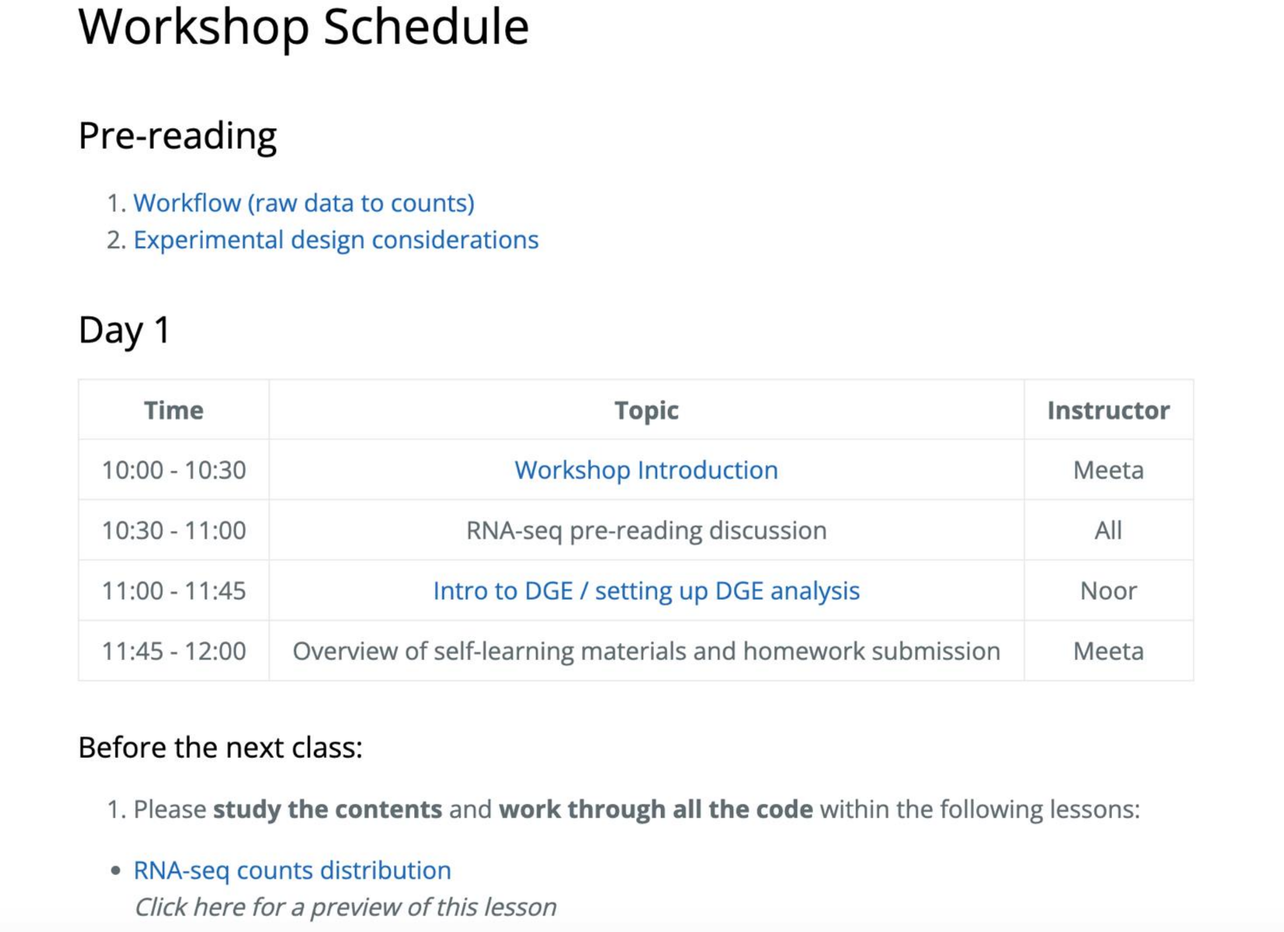
Workshop Scope



- Understand the considerations for performing statistical analysis on RNA-seq data
- Start with gene counts (after alignment and counting)
- Perform QC on count data
- Use DESeq2 to perform differential expression analysis on the count data and obtain a list of significantly different genes
- Visualize results of the analysis
- Perform functional analysis on the lists of differentially expressed genes

Logistics

Course schedule



Workshop Schedule

Pre-reading

1. Workflow (raw data to counts)
2. Experimental design considerations

Day 1

Time	Topic	Instructor
10:00 - 10:30	Workshop Introduction	Meeta
10:30 - 11:00	RNA-seq pre-reading discussion	All
11:00 - 11:45	Intro to DGE / setting up DGE analysis	Noor
11:45 - 12:00	Overview of self-learning materials and homework submission	Meeta

Before the next class:

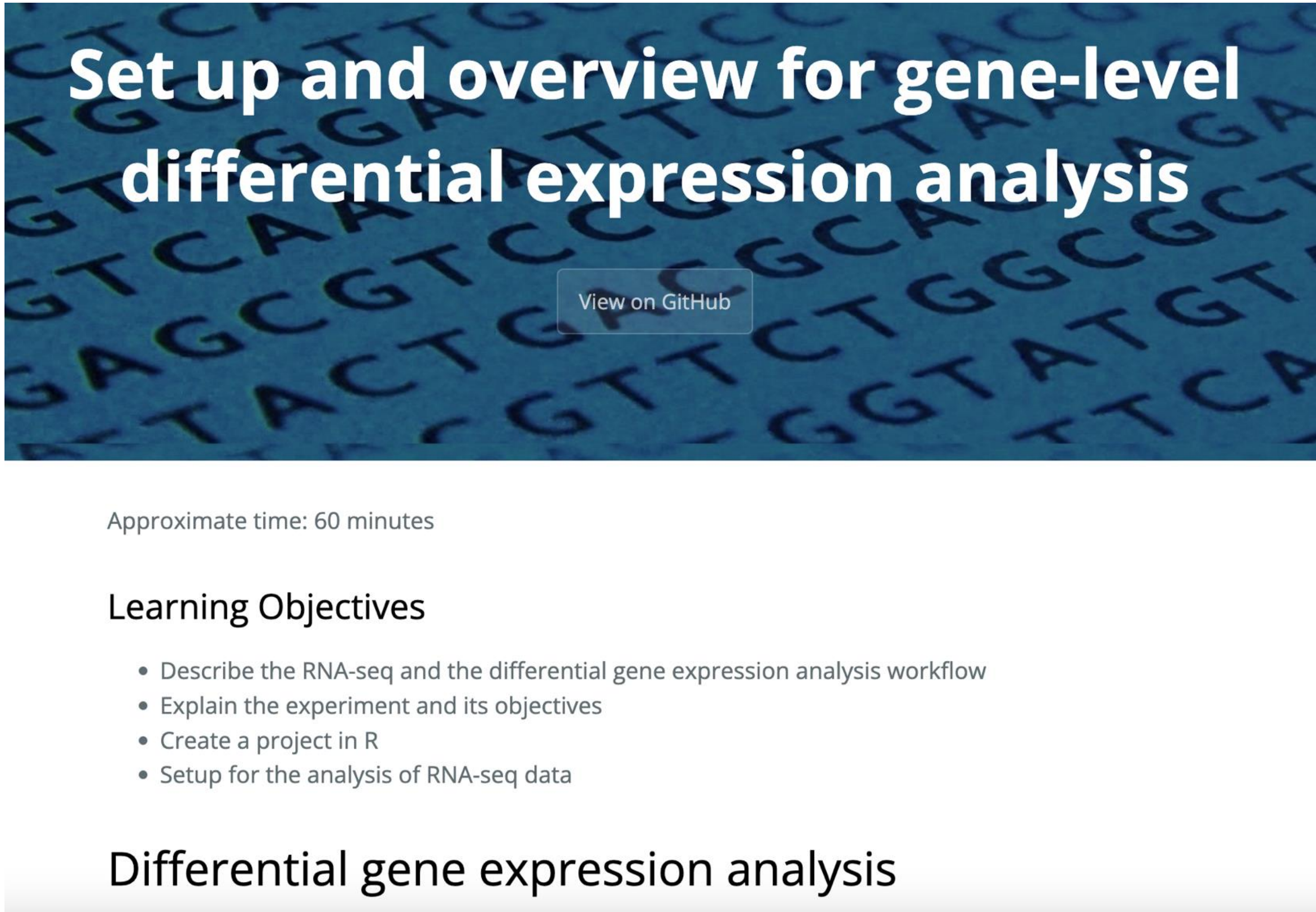
1. Please **study the contents** and **work through all the code** within the following lessons:

- RNA-seq counts distribution
  Click here for a preview of this lesson

https://tinyurl.com/hbc-dge

Course materials

- We continuously update our materials to reflect changes in the field/software



Set up and overview for gene-level differential expression analysis

Approximate time: 60 minutes

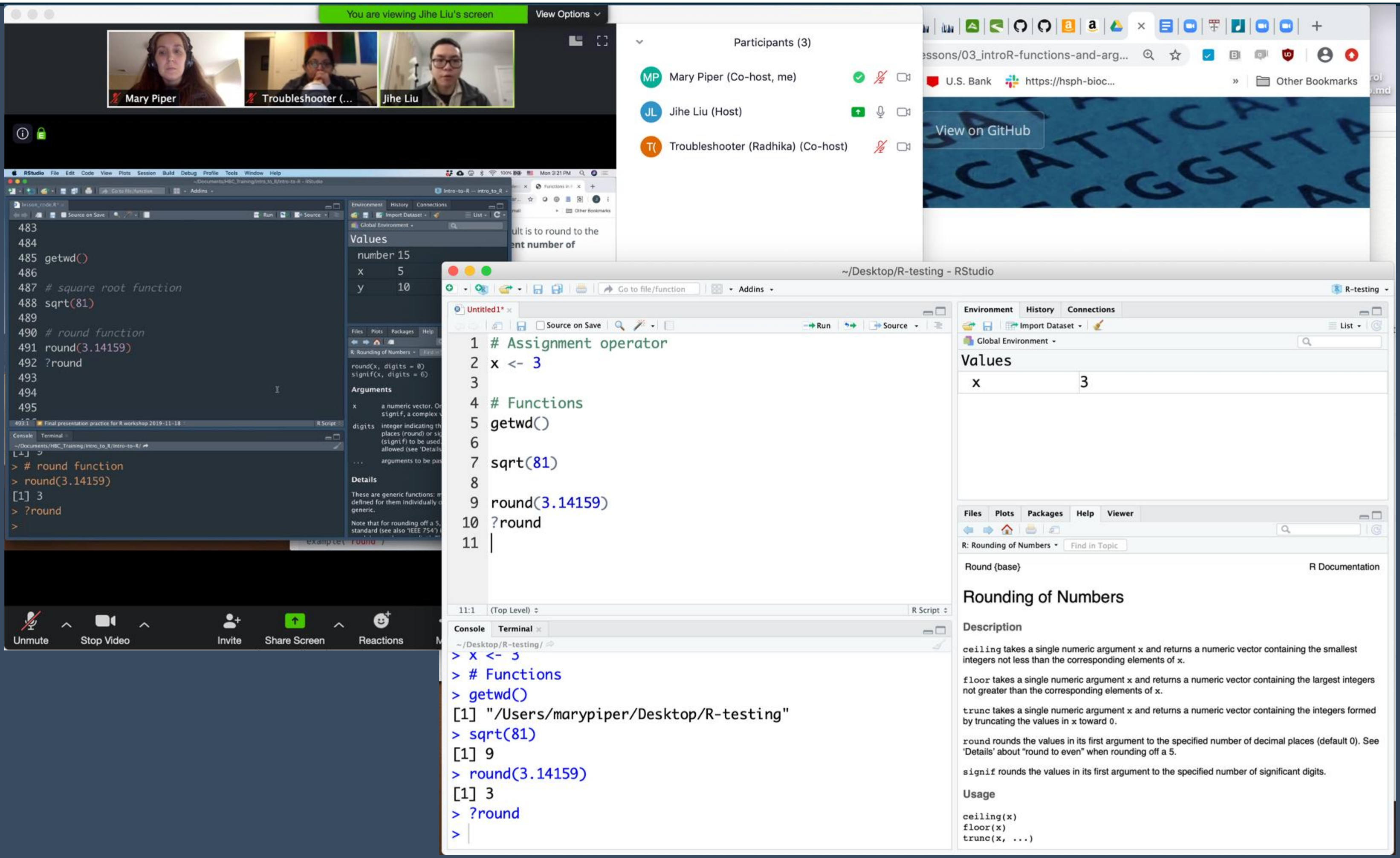
Learning Objectives

- Describe the RNA-seq and the differential gene expression analysis workflow
- Explain the experiment and its objectives
- Create a project in R
- Setup for the analysis of RNA-seq data

Differential gene expression analysis

https://tinyurl.com/hbc-dge

Single Screen & 3 Windows



```
getwd()

# square root function
sqrt(81)

# round function
round(3.14159)
?round
```

```
> # round function
> round(3.14159)
[1] 3
> ?round
>
```

```
# Assignment operator
x <- 3

# Functions
getwd()

sqrt(81)

round(3.14159)
?round
```

Values

x	3

```
> x <- 3
> # Functions
> getwd()
[1] "/Users/marypiper/Desktop/R-testing"
> sqrt(81)
[1] 9
> round(3.14159)
[1] 3
> ?round
>
```

Round {base} R Documentation

Rounding of Numbers

Description

`ceiling` takes a single numeric argument `x` and returns a numeric vector containing the smallest integers not less than the corresponding elements of `x`.

`floor` takes a single numeric argument `x` and returns a numeric vector containing the largest integers not greater than the corresponding elements of `x`.

`trunc` takes a single numeric argument `x` and returns a numeric vector containing the integers formed by truncating the values in `x` toward `0`.

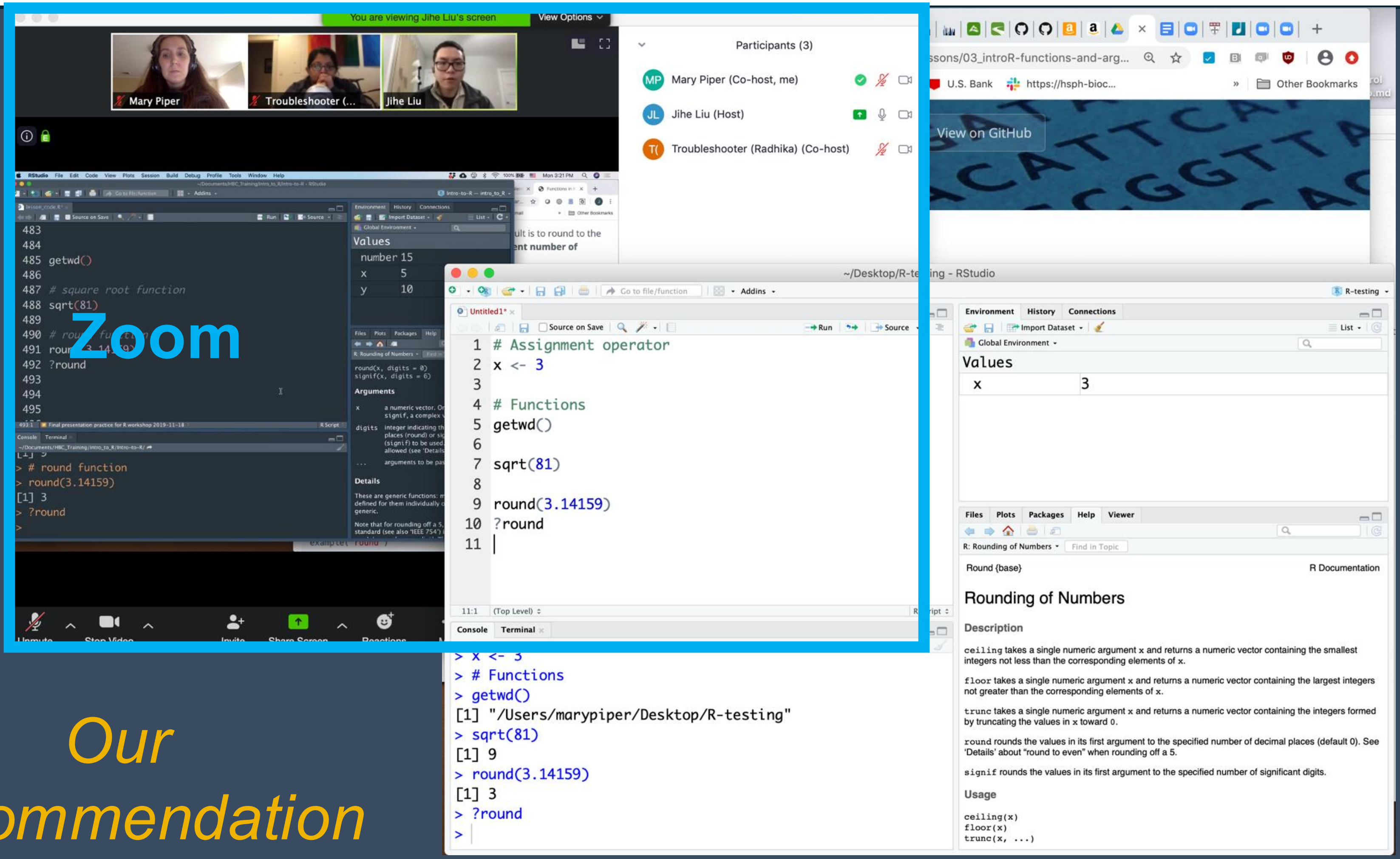
`round` rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

`signif` rounds the values in its first argument to the specified number of significant digits.

Usage

```
ceiling(x)
floor(x)
trunc(x, ...)
```

Single Screen & 3 Windows



```
# Assignment operator
x <- 3

# Functions
getwd()

sqrt(81)

round(3.14159)
?round
```

Values

x 3

```
> x <- 3
> # Functions
> getwd()
[1] "/Users/marypiper/Desktop/R-testing"
> sqrt(81)
[1] 9
> round(3.14159)
[1] 3
> ?round
>
```

Round {base} R Documentation

Rounding of Numbers

Description

`ceiling` takes a single numeric argument `x` and returns a numeric vector containing the smallest integers not less than the corresponding elements of `x`.

`floor` takes a single numeric argument `x` and returns a numeric vector containing the largest integers not greater than the corresponding elements of `x`.

`trunc` takes a single numeric argument `x` and returns a numeric vector containing the integers formed by truncating the values in `x` toward `0`.

`round` rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

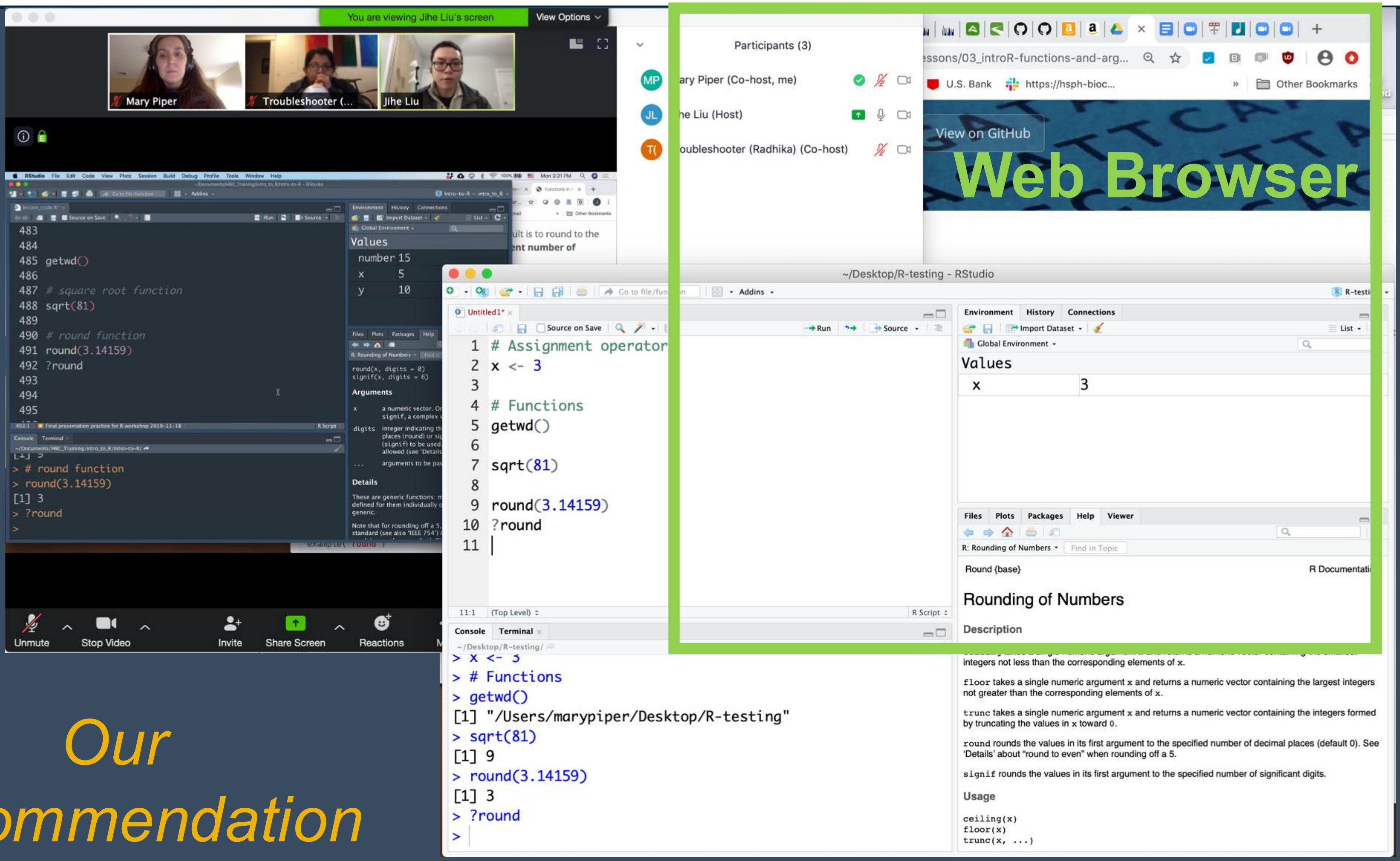
`signif` rounds the values in its first argument to the specified number of significant digits.

Usage

```
ceiling(x)
floor(x)
trunc(x, ...)
```

Our Recommendation

Single Screen & 3 Windows



```
getwd()

# square root function
sqrt(81)

# round function
round(3.14159)
?round
```

```
> # round function
> round(3.14159)
[1] 3
> ?round
>
```

Values
number 15
x 5
y 10

```
# Assignment operator
x <- 3

# Functions
getwd()

sqrt(81)

round(3.14159)
?round
```

```
> x <- 3
> # Functions
> getwd()
[1] "/Users/marypiper/Desktop/R-testing"
> sqrt(81)
[1] 9
> round(3.14159)
[1] 3
> ?round
>
```

Values
x 3

Round {base} R Documentation

Rounding of Numbers

Description

integers not less than the corresponding elements of x.

floor takes a single numeric argument x and returns a numeric vector containing the largest integers not greater than the corresponding elements of x.

trunc takes a single numeric argument x and returns a numeric vector containing the integers formed by truncating the values in x toward 0.

round rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

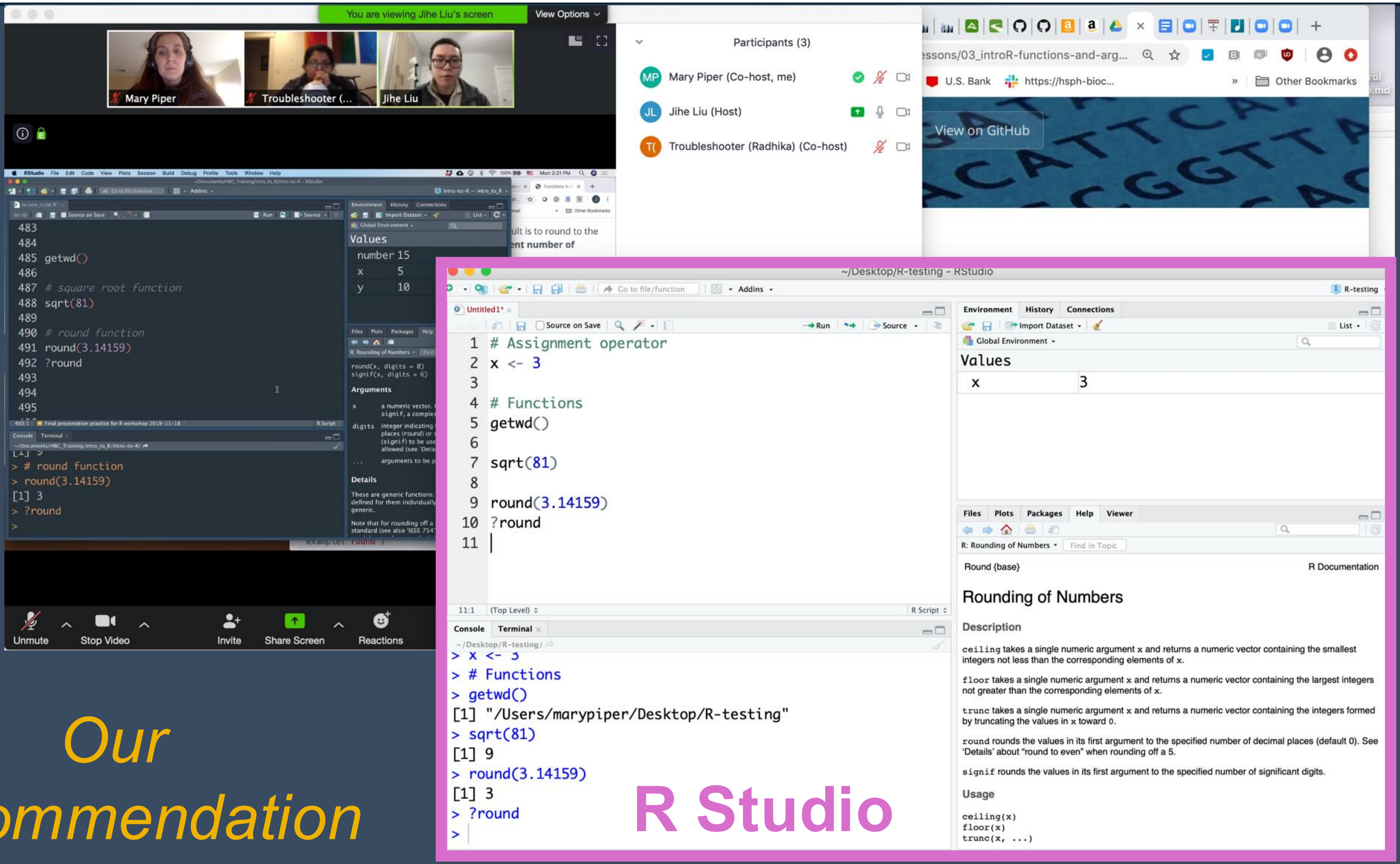
signif rounds the values in its first argument to the specified number of significant digits.

Usage

```
ceiling(x)
floor(x)
trunc(x, ...)
```

Our Recommendation

Single Screen & 3 Windows



```
getwd()

# square root function
sqrt(81)

# round function
round(3.14159)
?round
```

```
> # round function
> round(3.14159)
[1] 3
> ?round
>
```

```
# Assignment operator
x <- 3

# Functions
getwd()

sqrt(81)

round(3.14159)
?round
```

Values

x	3

```
> x <- 3
> # Functions
> getwd()
[1] "/Users/marypiper/Desktop/R-testing"
> sqrt(81)
[1] 9
> round(3.14159)
[1] 3
> ?round
>
```

Round {base} R Documentation

Rounding of Numbers

Description

`ceiling` takes a single numeric argument `x` and returns a numeric vector containing the smallest integers not less than the corresponding elements of `x`.

`floor` takes a single numeric argument `x` and returns a numeric vector containing the largest integers not greater than the corresponding elements of `x`.

`trunc` takes a single numeric argument `x` and returns a numeric vector containing the integers formed by truncating the values in `x` toward `0`.

`round` rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

`signif` rounds the values in its first argument to the specified number of significant digits.

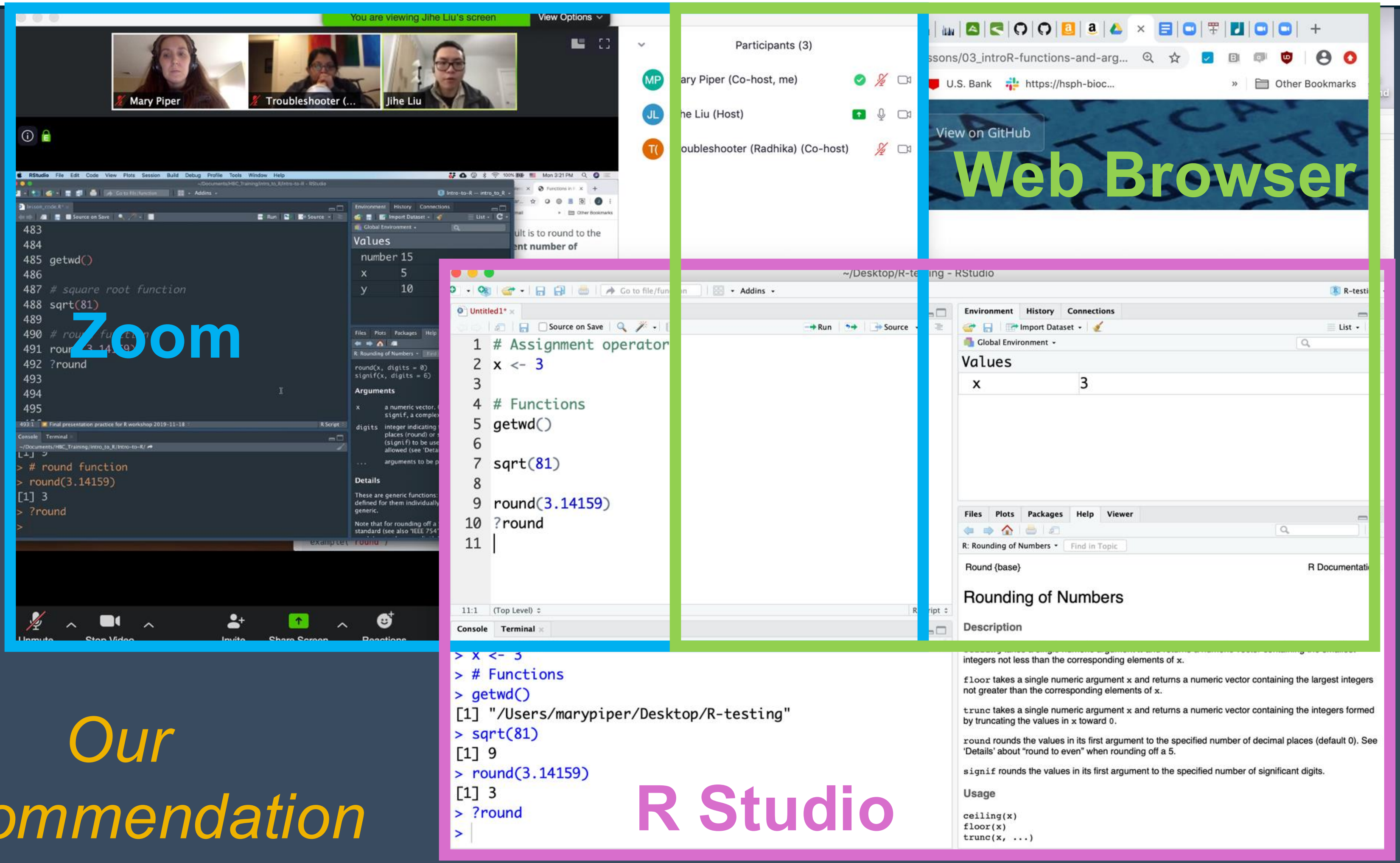
Usage

```
ceiling(x)
floor(x)
trunc(x, ...)
```

R Studio

Our Recommendation

Single Screen & 3 Windows



```
# Assignment operator
x <- 3

# Functions
getwd()

sqrt(81)

round(3.14159)
?round

```

```
> x <- 3
> # Functions
> getwd()
[1] "/Users/marypiper/Desktop/R-testing"
> sqrt(81)
[1] 9
> round(3.14159)
[1] 3
> ?round
>
```

Round {base} R Documentati

Rounding of Numbers

Description

integers not less than the corresponding elements of x.

floor takes a single numeric argument x and returns a numeric vector containing the largest integers not greater than the corresponding elements of x.

trunc takes a single numeric argument x and returns a numeric vector containing the integers formed by truncating the values in x toward 0.

round rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

signif rounds the values in its first argument to the specified number of significant digits.

Usage

```
ceiling(x)
floor(x)
trunc(x, ...)
```

Our Recommendation

Course participation

- Mandatory review of self-learning lessons and assignments
- Attendance required for all classes
- Your questions and active participation drive learning
- **We look forward to all of your questions!**

Course participation

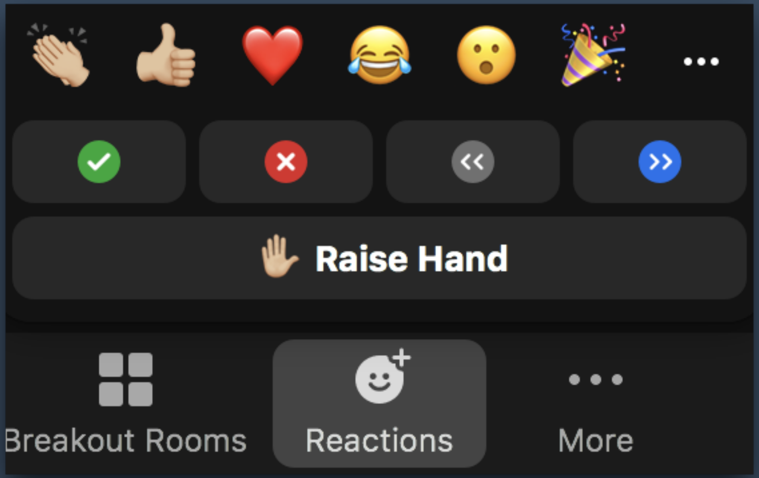
- At-home lessons and exercises after each session
- Cover material not previously discussed
- Provides us feedback to help pace the course appropriately
- 3-5 hours to complete
- Homework load is heavier in the beginning of this workshop series and tapers off

Using AI for Assignments

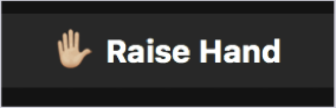
- Do
  - Try to resolve error messages with it
  - Test code written by AI on a dataset where you have expected results
  - Take the time to review the generated code line-by-line
- Don’t
  - Implement it in replacement to learning
  - Write code that you don’t understand
  - Assume the output from an AI process is correct

Odds & Ends

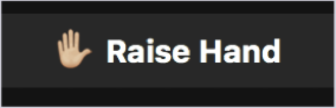
- Quit/minimize all applications that are not required for class
- Are you all set?
  - ✓ = "agree", "I'm all set"
  - ✗ = "disagree", "I need help"



Odds & Ends

- Questions for the presenter?
  - Post the question in the Chat window OR
  -  when the presenter asks for questions
  - Let the Moderator know

Odds & Ends

- Questions for the presenter?
    - Post the question in the Chat window OR
    -  Raise Hand when the presenter asks for questions
    - Let the Moderator know
- Technical difficulties with software?
    - Start a private chat with the Troubleshooter with a description of the problem

Contact Us

- *HBC training team:* hbctraining@hsph.harvard.edu
- *HBC consulting:* bioinformatics@hsph.harvard.edu