Data Structures and Data Wrangling

TRR259 seminar Session 1

JS

29.10.2025

Goal: Empower you to perform your own data analysis

1. Develop skills in data handling and visualisation
2. Speed up data processing and analysis of repeated projects
3. Dive into Sequencing data and RNA Seq analysis
4. Underline domain-specific statistics

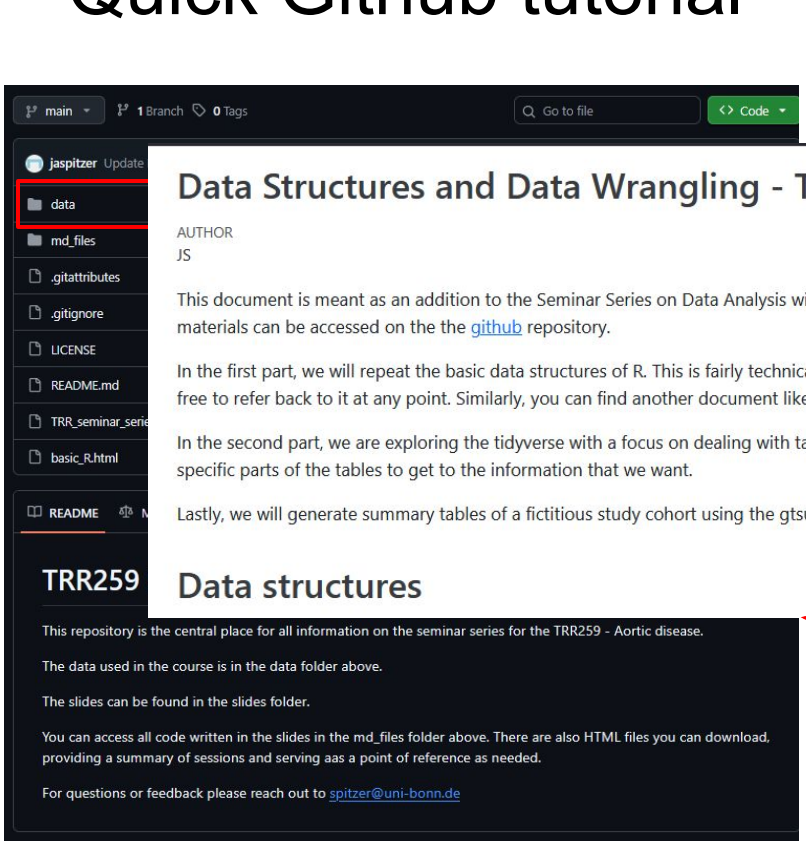
Outline for the first Session

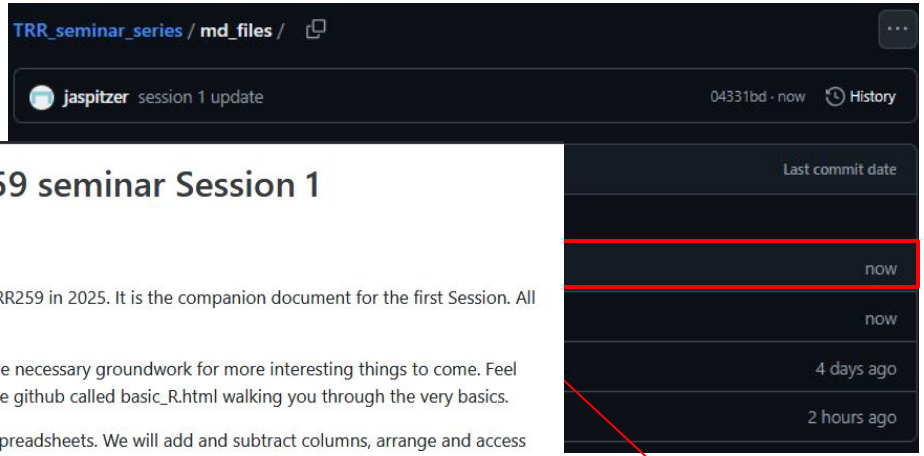
1. General introduction, expectations, goals
2. Data structures in R
3. Data wrangling using the tidyverse
4. Summary of survey data using gtsummary
5. Practice Problems

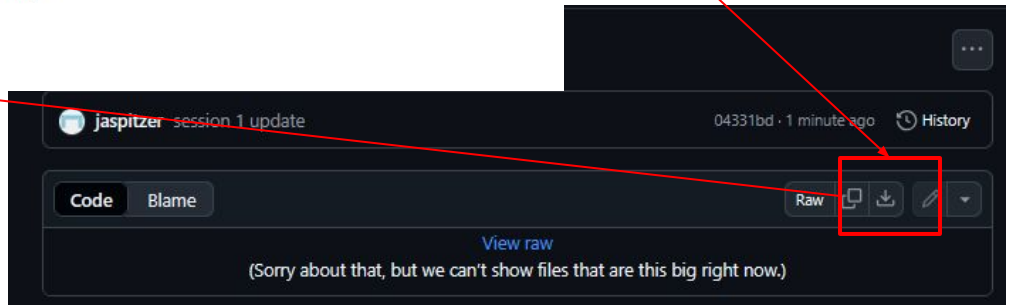
Course structure and materials

- 4 sessions with one session every two weeks, each session 2 to 2.5 h
- 1-1.5 h will be lecture-style presentation, the rest will be practical
- Material will be deposited on the github page:
  - Slides
  - All code written here, compiled with an in-depth explanation
  - questions and problems
- for questions, suggestions and complaints, feel free to contact me under spitzer@uni-bonn.de

Quick Github tutorial







Go with the flow!

- you are learning a new coding language
- you are learning how to think differently about your data
- you are learning new statistics
- you are going to learn new biology

This stuff is not easy, understanding takes time

Data structures in R

- vector
- factor
- matrix
- data frame
- list

Data structures in R - Vector

`c(120,360,600,840)`	Creates a vector
`ages <- c(120,360,600,840)`	Saves the vector with the name “ages”
`length(ages)`	How many values are in “ages”?
`ages[3]`	Third value in ages
`ages[3:4] | ages[c(3,4)]`	First two values of the vector
`ages[-3]`	Everything but the third value
`ages + 5`	Adds 5 to every age in ages
`ages / 60`	Divides all ages by 60
`ages > 480`	Is an age larger than 480?
`sum(ages > 480)`	How many ages are over 480?
`mean(ages)`	What is the mean age?

Data structures in R - Vector

```
c(120,360,600,840)

ages <- c(120,360,600,840)

length(ages)

ages[3]

ages[3:4] | ages[c(3,4)]

ages[-3]

ages + 5

ages / 60

ages > 480

sum(ages > 480)

mean(ages)
```

```
[1] 120 360 600 840
[1] 4
[1] 600
[1] 600 840
[1] 600 840
[1] 120 360 840
[1] 125 365 605 845
[1]  2  6 10 14
[1] FALSE FALSE  TRUE  TRUE
[1] 2
[1] 480
```

Data structures in R - Factor

```
f <- factor(c(“C”,”C”,”C”,
              “A”,“A”,“A”),
       levels = c(“A”, “C”))

sort(f)

f <- relevel(f, “C”)

sort(f)

levels(f)
```

Creates a factor with 2 levels, A and C
By default, levels are alphabetical

sort *f*

Relevel *f* with C as the default

sort *f*

Access the levels of *f*

Data structures in R - Factor

```
f <- factor(c(“C”,”C”,”C”,
              “A”,“A”,“A”),
       levels = c(“A”, “C”))

sort(f)

f <- relevel(f, “C”)

sort(f)

levels(f)
```

```
[1] A A A C C C
Levels: A C
[1] C C C A A A
Levels: C A
[1] "C" "A"
```

Data structures in R - two dimensional data

- most commonly used data format
- we’ll use both matrix and dataframe, but mostly dataframes
- both forms are organised in rows and columns

Data structures in R - Matrix

```
m <- matrix(1:20, nrow = 5)
m[1, ]
m[ ,1]
colnames(m) <- c("a", "b", "c", "d")
rownames(m) <- c("v", "w", "x", "y", “z”)
m["x", "a"]
```

Creates a matrix, here a simple 4 by 5 matrix

retrieves the first row of the *m*

retrieves the first column of the *m*

Sets column names for *m*

sets row names for *m*

Accesses a single value using the names

Data structures in R - Matrix

```
m <- matrix(1:20, nrow = 5)

m[1, ]

m[ ,1]

colnames(m) <- c("a", "b", "c", "d")

rownames(m) <- c("v", "w", "x", "y", “z”)

m["x", "a"]
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20
[1]  1  6 11 16
[1] 1 2 3 4 5
  a  b  c  d
v 1  6 11 16
w 2  7 12 17
x 3  8 13 18
y 4  9 14 19
z 5 10 15 20
[1] 3
```

Data structures in R - Data Frame

```
df <- data.frame(diagnosis = rep(c("CTRL",
                                   "AAA"),
                                 each = 5),
                 diameter = c(rnorm(5, 5, 0.5),
                              rnorm(5, 2, 0.1)),
                 age = round(runif(10, 45, 80)))

df

df[ ,1]

df$diameter

summary(df)

df$diagnosis <- as.factor(df$diagnosis)

summary(df)
```

Generates a data frame *df* with 3 variables

diagnosis - CTRL or AAA

diameter, here randomly generated

age, randomly generated

returns the data frame *df*

Returns the first column of *df*

Returns the column of *df* called diameter

Quick statistical overview for *df*

Replace diagnosis with a factor version

Results from *summary()* changes now

Data structures in R - Data Frame

```
df <- data.frame(diagnosis = rep(c("CTRL",
                                   "AAA"),
                                 each = 5),
                 diameter = c(rnorm(5, 5, 0.5),
                              rnorm(5, 2, 0.1)),
                 age = round(runif(10, 45, 80)))

df

df[ ,1]

df$diameter

summary(df)

df$diagnosis <- as.factor(df$diagnosis)

summary(df)
```

diagnosis <chr>	diameter <dbl>	age <dbl>
CTRL	5.348054	64
CTRL	4.206326	70
CTRL	4.765926	55
CTRL	4.408508	69
CTRL	4.438945	59
AAA	2.154448	59
AAA	2.146144	73
AAA	2.187224	67
AAA	2.210580	51
AAA	2.000905	49

1-10 of 10 rows

```
 [1] "CTRL" "CTRL" "CTRL" "CTRL" "CTRL" "AAA"  "AAA"  "AAA"  "AAA"  "AAA"
 [1] 5.348054 4.206326 4.765926 4.408508 4.438945 2.154448 2.146144 2.187224 2.210580
2.000905
  diagnosis           diameter          age
 Length:10          Min.   :2.001   Min.   :49.0
 Class :character   1st Qu.:2.163   1st Qu.:56.0
 Mode  :character   Median :3.208   Median :61.5
                    Mean   :3.387   Mean   :61.6
                    3rd Qu.:4.431   3rd Qu.:68.5
                    Max.   :5.348   Max.   :73.0
 diagnosis    diameter          age
 AAA :5    Min.   :2.001   Min.   :49.0
 CTRL:5    1st Qu.:2.163   1st Qu.:56.0
           Median :3.208   Median :61.5
           Mean   :3.387   Mean   :61.6
           3rd Qu.:4.431   3rd Qu.:68.5
           Max.   :5.348   Max.   :73.0
```

Data structures in R - list

`examples <- list("a", c("c", "example"), df))`	generates a *list* and stores it in *examples*
`examples[[1]]`	accesses the first element of *examples*, note the double brackets
`examples[1:2]`	accesses the first elements of *examples*, but returns a list
`length(examples)`	How many elements are in *examples*?
`str(examples)`	Gives a summary of *examples* and its elements

Data structures in R - list

```
examples <- list(“a”, c(“c”, “example”), df))

examples[[1]]


examples[1:2]


length(examples)

str(examples)
```

```
[1] "a"
[[1]]
[1] "a"

[[2]]
[1] "c"       "example"

[1] 3
List of 3
 $ : chr "a"
 $ : chr [1:2] "c" "example"
 $ :'data.frame':       10 obs. of  3 variables:
  ..$ diagnosis: Factor w/ 2 levels "AAA","CTRL": 2 2 2 2 2 1 1 1 1 1
  ..$ diameter : num [1:10] 4.71 5.2 5.36 4.51 4.65 ...
  ..$ age      : num [1:10] 73 77 53 47 64 66 80 73 47 64
```

Data structures in R - named vectors and lists

```
ducks <- list(number_of_ducks = 3,
              duck_names = c(“Tick”, “Trick”,
                             “Track”),
              duck_info = df)

ducks[[1]]

ducks[[“duck_info”]]

ducks$duck_names
```

generates a *list* and stores it in *ducks*
examples has 3 elements, all about ducks

elements can be accessed by position

elements can be accessed by name

elements can be accessed by name using “$”

Data structures in R - named vectors and lists

```
ducks <- list(number_of_ducks = 3,
              duck_names = c(“Tick”, “Trick”,
                             “Track”),
              duck_info = df)

ducks[[1]]

ducks[[“duck_info”]]

ducks$duck_names
```

```
[1] 3
[1] "Tick"  "Trick" "Track"
```

diagnosis <fctr>	diameter <dbl>	age <dbl>
CTRL	4.709629	73
CTRL	5.204867	77
CTRL	5.355331	53
CTRL	4.513321	47
CTRL	4.650572	64
AAA	2.034768	66
AAA	2.019956	80
AAA	2.052112	73
AAA	2.159436	47
AAA	1.974188	64

1-10 of 10 rows

Data wrangling in the *tidyverse*

- group of packages designed for data science
- most of the work will take place in data frames or the tidyverse version, *tibble*
- assumed structure: each row is an observation, each column is a variable

Individual packages in this family have specialised purpose

- *dplyr* and *tidyr* for data cleaning and wrangling
- *stringr* and *forcats* for character and factor manipulation

Data used throughout the course - aneurysm data

- generated by me (not real data!)
- contains 900 patients, their aortic diameters as well as some additional information on comorbidities, age, etc.
- this data set is meant as an exercise tool

A tibble: 900 × 9

diagnosis <chr>	abdominal_... <dbl>	thoracic_dia... <dbl>	age <dbl>	male <dbl>	hypertension <dbl>	cad <dbl>	artherioscle... <dbl>	bmi <dbl>
CTRL	1.903918	1.918242	64	0	0	0	0	25.20
CTRL	2.050726	2.033109	49	1	0	0	1	22.95
CTRL	1.780361	1.868136	55	0	0	0	1	23.34
CTRL	1.799354	1.641936	52	1	0	1	0	22.26
CTRL	1.950962	1.588081	52	1	1	0	0	23.07
CTRL	2.081565	2.117073	69	1	0	0	1	23.57
CTRL	1.783003	1.670469	74	1	1	0	0	24.23
CTRL	2.144597	2.048529	47	0	1	0	1	22.75
CTRL	1.975882	1.762458	52	0	0	0	1	22.75
CTRL	2.075961	2.249491	76	0	0	0	1	27.66

1-10 of 900 rows

Previous 1 2 3 4 5 6 ... 90 Next

Data used throughout the course - airway data

- data from the airway R package
- split into expression and annotation

A tibble: 53,563 × 9

gene <chr>	SRR1039508 <dbl>	SRR1039509 <dbl>	SRR1039512 <dbl>	SRR1039513 <dbl>	SRR1039516 <dbl>	SRR1039517 <dbl>	SRR1039520 <dbl>	SRR10
TSPAN6	679	448	873	408	1138	1047	770	
TNMD	0	0	0	0	0	0	0	0
DPM1	467	515	621	365	587	799	417	508
SCYL3	260	211	263	164	245	331	233	229
C1orf112	60	55	40	35	78	63	76	60
FGR	0	0	2	0	1	0	0	0
CFH	3251	3679	6177	4				
FUCA2	1433	1062	1733					
GCLC	519	380	595					
NFYA	394	236	464					

1-10 of 53,563 rows

A tibble: 8 × 9

SampleName <chr>	cell <chr>	dex <chr>	albut <chr>	Run <chr>	avgLength <dbl>	Experiment <chr>	Sample <chr>
GSM1275862	N61311	untrt	untrt	SRR1039508	126	SRX384345	SRS508568
GSM1275863	N61311	trt	untrt	SRR1039509	126	SRX384346	SRS508567
GSM1275866	N052611	untrt	untrt	SRR1039512	126	SRX384349	SRS508571
GSM1275867	N052611	trt	untrt	SRR1039513	87	SRX384350	SRS508572
GSM1275870	N080611	untrt	untrt	SRR1039516	120	SRX384353	SRS508575
GSM1275871	N080611	trt	untrt	SRR1039517	126	SRX384354	SRS508576
GSM1275874	N061011	untrt	untrt	SRR1039520	101	SRX384357	SRS508579
GSM1275875	N061011	trt	untrt	SRR1039521	98	SRX384358	SRS508580

8 rows

› PLoS One. 2014 Jun 13;9(6):e99625. doi: 10.1371/journal.pone.0099625. eCollection 2014.

RNA-Seq transcriptome profiling identifies CRISPLD2 as a glucocorticoid responsive gene that modulates cytokine function in airway smooth muscle cells

Blanca E Himes [1], Xiaofeng Jiang [2], Peter Wagner [2], Ruoxi Hu [2], Qiyu Wang [2], Barbara Klanderman [3], Reid M Whitaker [4], Qingling Duan [4], Jessica Lasky-Su [4], Christina Nikolos [5], William Jester [5], Martin Johnson [5], Reynold A Panettieri Jr [5], Kelan G Tantisira [4], Scott T Weiss [6], Quan Lu [2]

Affiliations + expand

PMID: 24926665 PMCID: PMC4057123 DOI: 10.1371/journal.pone.0099625

readr and readxl


```
library(tidyverse)

df <- read_tsv(“example_data.txt”)

df <- read_csv(“example_data.csv”)

df <- read_csv2(“example_data2.csv”)


readxl::read_xlsx(“example_data.xlsx”, sheet = 1)
```

activating the *tidyverse* packages

reading in a tab separated .txt file, saving it in *df*

read in a comma separated .csv file, saving it in df

read in the same file as above, using a totally not annoying german convention of .csv files, where the separator is “;”, and the decimal point is “,”

read in the same date from an excel file
readxl is a little discouraged, so you need to specify its use or un *library(readxl)*

readr and readxl

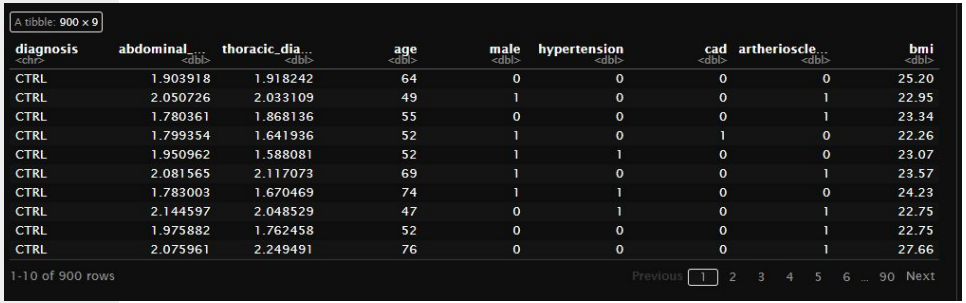
```
library(tidyverse)

df <- read_tsv("example_data.txt")

df <- read_csv("example_data.csv")

df <- read_csv2("example_data2.csv")


readxl::read_xlsx("example_data.xlsx", sheet = 1)
```



A tibble: 900 × 9

diagnosis <chr>	abdominal_... <dbl>	thoracic_dia... <dbl>	age <dbl>	male <dbl>	hypertension <dbl>	cad <dbl>	artherioscle... <dbl>	bmi <dbl>
CTRL	1.903918	1.918242	64	0	0	0	0	25.20
CTRL	2.050726	2.033109	49	1	0	0	1	22.95
CTRL	1.780361	1.868136	55	0	0	0	1	23.34
CTRL	1.799354	1.641936	52	1	0	1	0	22.26
CTRL	1.950962	1.588081	52	1	1	0	0	23.07
CTRL	2.081565	2.117073	69	1	0	0	1	23.57
CTRL	1.783003	1.670469	74	1	1	0	0	24.23
CTRL	2.144597	2.048529	47	0	1	0	1	22.75
CTRL	1.975882	1.762458	52	0	0	0	1	22.75
CTRL	2.075961	2.249491	76	0	0	0	1	27.66

1-10 of 900 rows

Previous 1 2 3 4 5 6 ... 90 Next

dplyr - grammar for data manipulation

- mutate - creating a new column
- filter - filter the observation based on a TRUE/FALSE criterium
- select - select (or de-select) columns by name, position, class
- summarise - perform summary calculations like mean, median, sum, …

- group_by - group observations by a variable and apply following calculations group-wise

mutate() examples


```
df <- aa_data

df <- mutate(df,
          patient = paste("Pat", 1:900, sep="_"))

df <- mutate(df, age_months = age * 12)

df <- mutate(df,severe_AAA = ifelse(
                        abdominal_diameter > 5,
                        "severe",
                        "non_severe"))

df <- mutate(df, diameter_ratio =
          abdominal_diameter / thoracic_diameter)

df <- mutate(df, diameter_ratio =
          round(diameter_ratio, 2))
```

save the aneurysm data in *df*

adds a new column called patient to *df*

adds a new column, multiplying *age* by 12

returns a classification of severe AAA based on *diameter*

Calculates the ratio of diameters

Round the previous ratio

mutate() examples


```
df <- aa_data

df <- mutate(df,
          patient = paste("Pat", 1:900, sep="_"))

df <- mutate(df, age_months = age * 12)

df <- mutate(df,severe_AAA = ifelse(diameter > 5,
                                    "severe",
                                    "non_severe"))

df <- mutate(df, diameter_ratio =
          abdominal_diameter / thoracic_diameter)

df <- mutate(df, diameter_ratio =
          thoracic_diameter / abdominal_diameter)

df <- mutate(df, diameter_ratio =
          round(diameter_ratio, 2))
```

```
Rows: 900
Columns: 13
$ diagnosis          <chr> "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL",
"CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CT…
$ abdominal_diameter <dbl> 1.903918, 2.050726, 1.780361, 1.799354, 1.950962, 2.081565, 1.783003, 2.144597, 1.975882,
2.075961, 2.197370, 1.883413, 1.864002, 2…
$ thoracic_diameter  <dbl> 1.918242, 2.033109, 1.868136, 1.641936, 1.588081, 2.117073, 1.670469, 2.048529, 1.762458,
2.249491, 2.157571, 1.946155, 1.762742, 1…
$ age                <dbl> 64, 49, 55, 52, 52, 69, 74, 47, 52, 76, 48, 71, 61, 46, 60, 76, 65, 72, 59, 58, 72, 72, 50,
61, 77, 74, 69, 63, 66, 75, 63, 57, 63,…
$ male               <dbl> 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0,
0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0,…
$ hypertension       <dbl> 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0,…
$ cad                <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,…
$ arteriosclerosis   <dbl> 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1,
1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0,…
$ bmi                <dbl> 25.20, 22.95, 23.34, 22.26, 23.07, 23.57, 24.23, 22.75, 22.75, 27.66, 20.97, 22.94, 26.49,
21.74, 25.72, 24.90, 22.32, 25.91, 25.94…
$ patient            <chr> "Pat_1", "Pat_2", "Pat_3", "Pat_4", "Pat_5", "Pat_6", "Pat_7", "Pat_8", "Pat_9", "Pat_10",
"Pat_11", "Pat_12", "Pat_13", "Pat_14", …
$ age_months         <dbl> 768, 588, 660, 624, 624, 828, 888, 564, 624, 912, 576, 852, 732, 552, 720, 912, 780, 864,
708, 696, 864, 864, 600, 732, 924, 888, 8…
$ severe_AAA         <chr> "non_severe", "non_severe", "non_severe", "non_severe", "non_severe", "non_severe",
"non_severe", "non_severe", "non_severe", "non_…
$ diameter_ratio     <dbl> 1.01, 0.99, 1.05, 0.91, 0.81, 1.02, 0.94, 0.96, 0.89, 1.08, 0.98, 1.03, 0.95, 0.80, 0.81,
0.83, 1.00, 1.01, 1.02, 0.90, 1.09, 1.14,…
```

mutate() examples


```
Rows: 900
Columns: 13
$ diagnosis          <chr> "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CTRL",
"CTRL", "CTRL", "CTRL", "CTRL", "CTRL", "CT…
$ abdominal_diameter <dbl> 1.903918, 2.050726, 1.780361, 1.799354, 1.950962, 2.081565, 1.783003, 2.144597, 1.975882,
2.075961, 2.197370, 1.883413, 1.864002, 2…
$ thoracic_diameter  <dbl> 1.918242, 2.033109, 1.868136, 1.641936, 1.588081, 2.117073, 1.670469, 2.048529, 1.762458,
2.249491, 2.157571, 1.946155, 1.762742, 1…
$ age                <dbl> 64, 49, 55, 52, 52, 69, 74, 47, 52, 76, 48, 71, 61, 46, 60, 76, 65, 72, 59, 58, 72, 72, 50,
61, 77, 74, 69, 63, 66, 75, 63, 57, 63,…
$ male               <dbl> 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0,
0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0,…
$ hypertension       <dbl> 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0,…
$ cad                <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,…
$ arteriosclerosis   <dbl> 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1,
1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,…
$ bmi                <dbl> 25.20, 22.95, 23.34, 22.26, 23.07, 23.57, 24.23, 22.75, 22.75, 27.66, 20.97, 22.94, 26.49,
21.74, 25.72, 24.90, 22.32, 25.91, 25.94…
$ patient            <chr> "Pat_1", "Pat_2", "Pat_3", "Pat_4", "Pat_5", "Pat_6", "Pat_7", "Pat_8", "Pat_9", "Pat_10",
"Pat_11", "Pat_12", "Pat_13", "Pat_14", …
$ age_months         <dbl> 768, 588, 660, 624, 624, 828, 888, 564, 624, 912, 576, 852, 732, 552, 720, 912, 780, 864,
708, 696, 864, 864, 600, 732, 924, 888, 8…
$ severe_AAA         <chr> "non_severe", "non_severe", "non_severe", "non_severe", "non_severe", "non_severe",
"non_severe", "non_severe", "non_severe", "non_…
$ diameter_ratio     <dbl> 1.01, 0.99, 1.05, 0.91, 0.81, 1.02, 0.94, 0.96, 0.89, 1.08, 0.98, 1.03, 0.95, 0.80, 0.81,
0.83, 1.00, 1.01, 1.02, 0.90, 1.09, 1.14,…
```

filter() examples


```
df <- aa_data
```
save the aneurysm data in *df*

```
filter(df, diagnosis == “AAA”)
```
filters *df* for only the AAA *diagnosis*

```
filter(df, diameter > 4.5)
```
filters *df* for a diameter > 4.5

```
filter(df, age < 75, abdominal_diameter > 4.5)
```
filters for both *age* and the diameter

```
filter(df, abdominal_diameter > 4 |
           thoracic_diameter > 4)
```
filter if either abdominal or thoracic diameter > 4. The | indicates the or

```
df <- mutate(df,
          patient = paste(“Pat”, 1:900, sep=“_”))
```
the same *patient* column as before

```
filter(df, patient %in% c(“Pat_2”, “Pat_6”))
```
filters *df* for two patients using their ID

filter() examples


```
df <- aa_data

filter(df, diagnosis == “AAA”)

filter(df, diameter > 4.5)

filter(df, age < 75, abdominal_diameter > 4.5)

filter(df, abdominal_diameter > 4 |
           thoracic_diameter > 4)

df <- mutate(df,
          patient = paste(“Pat”, 1:900, sep=“_”))

filter(df, patient %in% c(“Pat_2”, “Pat_6”))
```

```
Rows: 300
Columns: 9
$ diagnosis           <chr> "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA"…
$ abdominal_diameter  <dbl> 3.535668, 3.897484, 4.831148, 4.427527, 3.206244, 4.247534, 3.215302, 5.101495, 3.304103, 3.598273, 5.712208, 2.412300, 4.894991, 3…
$ thoracic_diameter   <dbl> 2.096804, 1.998180, 2.024948, 2.080301, 2.125109, 2.092148, 2.118127, 2.123469, 2.071918, 1.926679, 2.109896, 2.126546, 1.928095, 1…
$ age                 <dbl> 74, 80, 55, 68, 56, 59, 61, 51, 46, 53, 78, 54, 61, 74, 65, 53, 56, 55, 56, 72, 71, 51, 64, 69, 63, 62, 48, 71, 56, 60, 58, 75, 51,…
$ male                <dbl> 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,…
$ hypertension        <dbl> 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,…
$ cad                 <dbl> 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,…
$ artheriosclerosis   <dbl> 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1,…
$ bmi                 <dbl> 21.63, 32.45, 29.95, 28.60, 19.83, 28.38, 25.00, 30.40, 23.38, 22.86, 35.38, 20.45, 20.21, 24.26, 37.82, 29.88, 44.94, 30.29, 21.80…
```

```
Rows: 294
Columns: 9
$ diagnosis           <chr> "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA", "AAA"…
$ abdominal_diameter  <dbl> 4.831148, 4.427527, 4.247534, 5.101495, 5.712208, 4.894991, 4.099050, 4.011038, 4.380367, 4.718541, 4.298386, 4.108062, 4.697727, 4…
$ thoracic_diameter   <dbl> 2.024948, 2.080301, 2.092148, 2.123469, 2.109896, 1.928095, 1.988699, 2.372107, 1.973466, 2.019829, 2.099083, 2.122954, 1.872085, 1…
$ age                 <dbl> 55, 68, 59, 51, 78, 61, 53, 55, 56, 72, 71, 51, 64, 69, 63, 56, 73, 52, 75, 72, 79, 62, 59, 52, 55, 74, 79, 60, 79, 74, 67, 75, 61,…
$ male                <dbl> 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0,…
$ hypertension        <dbl> 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1,…
$ cad                 <dbl> 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,…
$ artheriosclerosis   <dbl> 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,…
$ bmi                 <dbl> 29.95, 28.60, 28.38, 30.40, 35.38, 20.21, 29.88, 30.29, 21.80, 37.81, 37.24, 31.09, 29.49, 29.37, 23.91, 41.06, 22.62, 37.47, 31.70…
```

filter() & count()

```
df <- aa_data

df <- filter(df, male == 1,
                 hypertension!= 1,
                 cad == 1)

count(df, diagnosis)
```

save the aneurysm data in *df*

filter only male patients
without hypertensions
with coronary artery disease

counts the patients with different diagnosis filtered from the original data drame

A tibble: 3 × 2

diagnosis <chr>	n <int>
AAA	20
CTRL	6
TAA	16

3 rows

select() examples

`df <- aa_data`	save the aneurysm data in *df*
`select(df, diagnosis, age, abdomninal_diameter)`	select 3 columns from *df*
`select(df, -age)`	select all but one column from *df*
`select(df, 1:3)`	selects the first three columns
`select(df, -c(1:2))`	de-selects the first and second column from *df*
`select(df, where(is.character))`	only selects character columns from *df*
`select(df, where(is.numeric))`	only selects numeric columns from *df*
`select(df, diagnosis, where(is.numeric))`	selects a specific column and all numeric columns

select() examples

```
df <- aa_data

select(df, diagnosis, age, abdominal_diameter)

select(df, -age)

select(df, 1:3)

select(df, -c(1:2))

select(df, where(is.character))

select(df, where(is.numeric))

select(df, diagnosis, where(is.numeric))
```

A tibble: 900 × 3

diagnosis <chr>	age <dbl>	abdominal_diameter <dbl>
CTRL	64	1.903918
CTRL	49	2.050726
CTRL	55	1.780361
CTRL	52	1.799354
CTRL	52	1.950962
CTRL	69	2.081565
CTRL	74	1.783003
CTRL	47	2.144597
CTRL	52	1.975882
CTRL	76	2.075961

1-10 of 900 rows

A tibble: 900 × 8

abdominal_di... <dbl>	thoracic_dia... <dbl>	age <dbl>	male <dbl>	hypertension <dbl>	cad <dbl>	artherioscler... <dbl>	bmi <dbl>
1.903918	1.918242	64	0	0	0	0	25.20
2.050726	2.033109	49	1	0	0	1	22.95
1.780361	1.868136	55	0	0	0	1	23.34
1.799354	1.641936	52	1	0	1	0	22.26
1.950962	1.588081	52	1	1	0	0	23.07
2.081565	2.117073	69	1	0	0	1	23.57
1.783003	1.670469	74	1	1	0	0	24.23
2.144597	2.048529	47	0	1	0	1	22.75
1.975882	1.762458	52	0	0	0	1	22.75
2.075961	2.249491	76	0	0	0	1	27.66

1-10 of 900 rows

Previous 1 2 3 4 5 6 ... 90 Next

The Pipe: %>% and |>

Some R code can be rather wonky and illegible

```
df %>%
    filter(diagnosis == “AAA”) %>%
    mutate(operation = ifelse(diameter > 5,
                              “operation”,
                              “monitor”)) %>%
    filter(operation == “operation”, age < 75)
```

this code performs exactly the same operations, but it is formatted a different way. The pipe ‘%>%’ passes on the output to the next operation.

summarise() examples


```
df <- aa_data

summarise(df, mean_diameter = mean(diameter))

summarise(df, median_age = median(age),
              mean_age = mean(age),
              sd_age = sd(age))

df %>%
     group_by(diagnosis) %>%
     summarise(median_age = mean(age),
               sd_age = sd(age))

df %>%
     filter(diagnosis == “AAA”) %>%
     group_by(abdominal_diameter > 4.5) %>%
     summarise(mean_bmi = mean(bmi),
               sd_bmi = sd(bmi))
```

saving the data in *df*

get the overall mean of *diameter*

get the median *age* of the data
and the mean
and the standard deviation

pipes *df*
groups the *df* by *diagnosis*
summarise (per group), generate mean *age*
generate standard deviation of *age*

pipes *df*
filter on only AAA patients
group them by their abdominal diameter
calculate summary statistics

summarise() examples

```
df <- aa_data

summarise(df, mean_diameter = mean(diameter))

summarise(df, median_age = median(age),
              mean_age = mean(age),
              sd_age = sd(age))

df %>%
     group_by(diagnosis) %>%
     summarise(median_age = mean(age),
               sd_age = sd(age))

df %>%
     filter(diagnosis == “AAA”) %>%
     group_by(abdominal_diameter > 4.5) %>%
     summarise(mean_bmi = mean(bmi),
               sd_bmi = sd(bmi))
```

A tibble: 1 × 1

mean_abdominal_diameter <dbl>
2.660884

1 row

A tibble: 1 × 3

median_age <dbl>	mean_age <dbl>	sd_age <dbl>
62	62.11111	10.05529

1 row

A tibble: 3 × 3

diagnosis <chr>	median_age <dbl>	sd_age <dbl>
AAA	62	10.16246
CTRL	62	10.01787
TAA	62	10.01206

3 rows

A tibble: 2 × 3

abdominal_diameter > 4.5 <lgl>	mean_bmi <dbl>	sd_bmi <dbl>
FALSE	27.71643	5.793834
TRUE	27.37563	5.992346

2 rows

arrange(), *slice_min()* and *slice_max()*


```
df <- aa_data

arrange(df, bmi)

arrange(df, -bmi)

slice_max(df, abdominal_diameter, n = 3)

slice_min(df, bmi, n = 10)
```

saving the data in *df*

arrange the *data frame* by the *bmi* column, ascending

arrange the *data frame* by the *bmi* column, descending

get the top 3 observations by abdominal diameter

get the bottom 10 observations by *bmi*

diagnosis <chr>	abdominal_... <dbl>	thoracic_di... <dbl>	age <dbl>	male <dbl>	hypertension <dbl>	cad <dbl>	artherioscl... <dbl>	bmi <dbl>
TAA	1.417867	3.313813	63	0	1	0	0	20.13
CTRL	1.434809	1.857863	77	0	0	0	0	21.95
TAA	1.472359	4.125879	76	1	0	0	0	25.27
CTRL	1.474740	2.082274	67	0	0	0	0	24.93
TAA	1.498853	3.795084	75	0	0	0	0	24.25
CTRL	1.503340	1.643513	76	0	0	0	0	24.19
TAA	1.518932	3.893461	68	1	0	0	0	25.84
TAA	1.532775	4.091640	68	1	0	0	0	22.50
TAA	1.538799	3.861082	72	1	0	0	0	18.99
TAA	1.561262	3.621747	52	1	1	0	0	23.84

1-10 of 10 rows

A tibble: 3 × 9

diagnosis <chr>	abdominal_... <dbl>	thoracic_di... <dbl>	age <dbl>	male <dbl>	hypertension <dbl>	cad <dbl>	artherioscl... <dbl>	bmi <dbl>
AAA	6.025393	2.169723	46	1	0	0	0	36.33
AAA	5.740262	1.952531	62	1	0	0	1	25.34
AAA	5.712208	2.109896	78	1	1	0	0	35.38

3 rows

dplyr - grammar for data manipulation

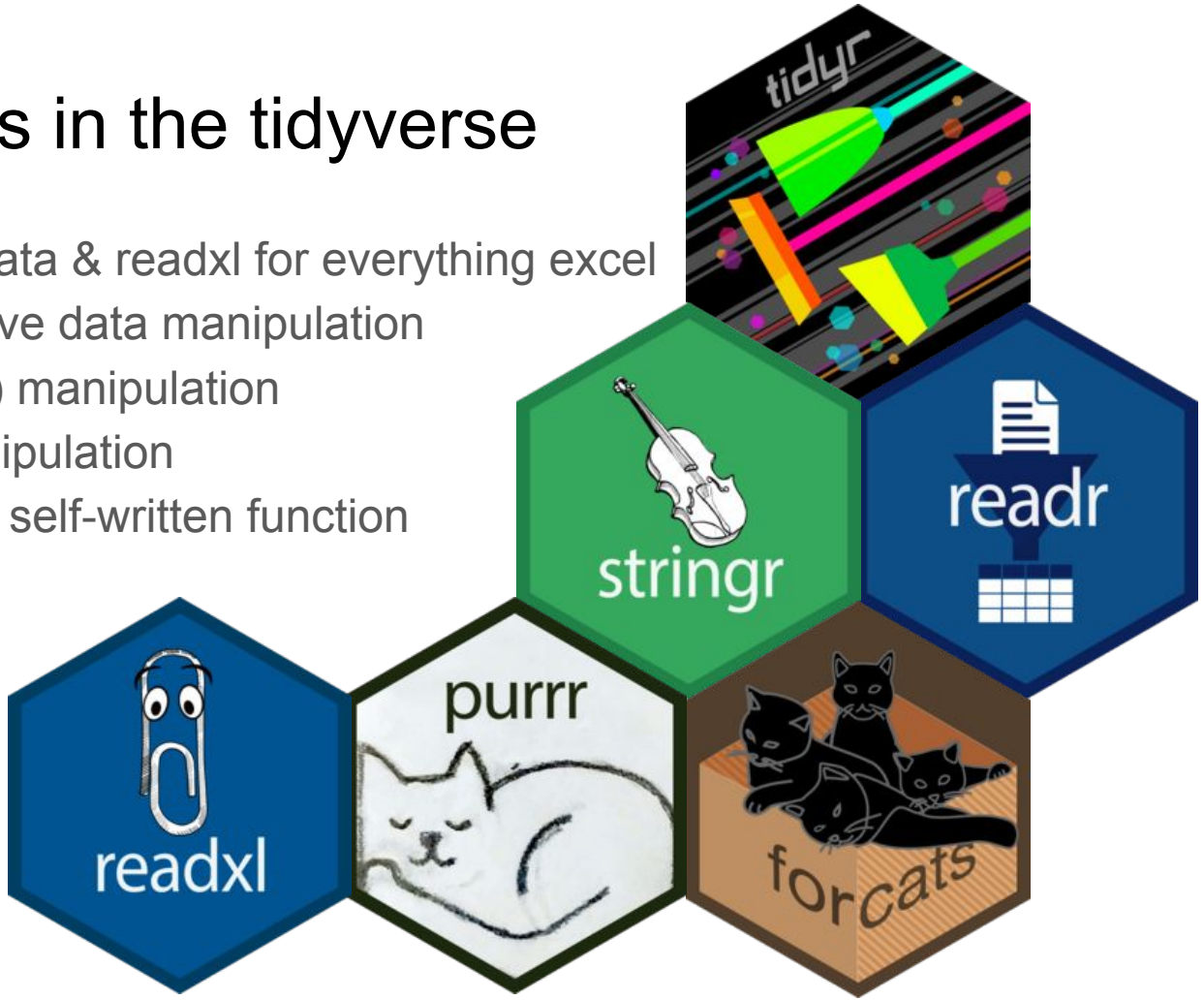
- mutate - creating a new column
- filter - filter the observation based on a TRUE/FALSE criterium
- select - select (or de-select) columns by name, position, class
- summarise - perform summary calculations like mean, median, sum, …

- group_by - group observations by a variable and apply following calculations group-wise

- chain together operations using the pipe

Further packages in the tidyverse

- readr for reading in data & readxl for everything excel
- tidyr for more extensive data manipulation
- stringr for string (text) manipulation
- forcats for factor manipulation
- purrr for working with self-written function



Some practise for you after the break

Use the Github to get to the html doc for this session: practise on the bottom:

Easy

E1: Use the code below to generate a vector called `words`. Access the third value of `words`.

```
words <- c("This", "is", "so", "so", "great")
```

E2: Use the `length()` function to get to the length of the vector. What is its length?

E3: What class is `words`?

Refer to the github and download the data from the data folder. As a help you can check out the the basic_R.html file, download and open it.

E3: Read in the aneurysm survey data. How many rows does it have? How many columns? Use the `colnames()` function to receive the column names.

E4: Add a column to the data frame containing a specific label for each patient. Which patient is the oldest? Which patient has the smallest thoracic diameter? Does that patient also have hypertension?

E5: Using the same data, how many patients have a thoracic diameter larger than 3.5 cm? Hint: use the `filter()` function

E6: We do not need that much precision in the diameters we have. Round them to two decimal points.

Data used throughout the course - airway data

- data from the airway R package
- split into expression and annotation

A tibble: 53,563 × 9

gene <chr>	SRR1039508 <dbl>	SRR1039509 <dbl>	SRR1039512 <dbl>	SRR1039513 <dbl>	SRR1039516 <dbl>	SRR1039517 <dbl>	SRR1039520 <dbl>	SRR10
TSPAN6	679	448	873	408	1138	1047	770	
TNMD	0	0	0	0	0	0	0	0
DPM1	467	515	621	365	587	799	417	508
SCYL3	260	211	263	164	245	331	233	229
C1orf112	60	55	40	35	78	63	76	60
FGR	0	0	2	0	1	0	0	0
CFH	3251	3679	6177	4				
FUCA2	1433	1062	1733					
GCLC	519	380	595					
NFYA	394	236	464					

1-10 of 53,563 rows

A tibble: 8 × 9

SampleName <chr>	cell <chr>	dex <chr>	albut <chr>	Run <chr>	avgLength <dbl>	Experiment <chr>	Sample <chr>
GSM1275862	N61311	untrt	untrt	SRR1039508	126	SRX384345	SRS508568
GSM1275863	N61311	trt	untrt	SRR1039509	126	SRX384346	SRS508567
GSM1275866	N052611	untrt	untrt	SRR1039512	126	SRX384349	SRS508571
GSM1275867	N052611	trt	untrt	SRR1039513	87	SRX384350	SRS508572
GSM1275870	N080611	untrt	untrt	SRR1039516	120	SRX384353	SRS508575
GSM1275871	N080611	trt	untrt	SRR1039517	126	SRX384354	SRS508576
GSM1275874	N061011	untrt	untrt	SRR1039520	101	SRX384357	SRS508579
GSM1275875	N061011	trt	untrt	SRR1039521	98	SRX384358	SRS508580

8 rows

› PLoS One. 2014 Jun 13;9(6):e99625. doi: 10.1371/journal.pone.0099625. eCollection 2014.

RNA-Seq transcriptome profiling identifies CRISPLD2 as a glucocorticoid responsive gene that modulates cytokine function in airway smooth muscle cells

Blanca E Himes [1], Xiaofeng Jiang [2], Peter Wagner [2], Ruoxi Hu [2], Qiyu Wang [2], Barbara Klanderman [3], Reid M Whitaker [4], Qingling Duan [4], Jessica Lasky-Su [4], Christina Nikolos [5], William Jester [5], Martin Johnson [5], Reynold A Panettieri Jr [5], Kelan G Tantisira [4], Scott T Weiss [6], Quan Lu [2]

Affiliations + expand

PMID: 24926665 PMCID: PMC4057123 DOI: 10.1371/journal.pone.0099625

tidyr for extensive data manipulation


```
library(tidyverse)

df <- read_tsv("data/expression_data/counts.txt")



df_long <- df %>%
     pivot_longer(cols = -gene,
                  names_to = "sample",
                  values_to = "count")
```

activating the *tidyverse* packages

reading in a tab separated .txt file, saving it in *df*

piping *df*
select the cols to pivot longer, here de-selecting *gene*
put the column names to *sample*
put the values in a new column named *count*

A tibble: 53,563 × 9

gene <chr>	SRR1039508 <dbl>	SRR1039509 <dbl>	SRR1039512 <dbl>	SRR1039513 <dbl>	SRR1039516 <dbl>	SRR1039517 <dbl>	SRR1039520 <dbl>	SRR1039521 <dbl>
TSPAN6	679	448	873	408	1138	1047	770	572
TNMD	0	0	0	0	0	0	0	0
DPM1	467	515	621	365	587	799	417	508
SCYL3	260	211	263	164	245	331	233	229
C1orf112	60	55	40	35	78	63	76	60
FGR	0	0	2	0	1	0	0	0
CFH	3251	3679	6177	4252	6721	11027	5176	7995
FUCA2	1433	1062	1733	881	1424	1439	1359	1109
GCLC	519	380	595	493	820	714	696	704
NFYA	394	236	464	175	658	584	360	269

1-10 of 53,563 rows Previous 1 2 3 4 5 6 ... 100 Next

gene <chr>	sample <chr>	count <dbl>
TSPAN6	SRR1039508	679
TSPAN6	SRR1039509	448
TSPAN6	SRR1039512	873
TSPAN6	SRR1039513	408
TSPAN6	SRR1039516	1138
TSPAN6	SRR1039517	1047

tidyr for extensive data manipulation


```
library(tidyverse)

anno <- read_tsv(“annotation.txt”)

anno_filtered <- anno %>%
     select(Run, dex, cell)

df_long <- df_long %>%
     inner_join(anno_filtered,
                by = join_by(sample == Run))


df_long %>%
     filter(gene == “IL6”) %>%
     group_by(dex) %>%
     summarise(mean_expression = mean(count))
```

activating the *tidyverse* packages

reading in a tab separated .txt file, saving it in *anno*

piping *anno*
select only columns of interest, here *Run*, *dex*, and *cell*

takes the *df_long* from the previous slides
joins in the annotation created above
observations are “sorted”, where *sample* in *df_long* is equivalent to *Run* from *anno_filtered*

take *df_long*
filter on only *IL6*
group by dexa treatment
get the mean expression per group

tidyr for extensive data manipulation

```
library(tidyverse)

anno <- read_tsv("annotation.txt")

anno_filtered <- anno %>%
     select(Run, dex, cell)

df_long <- df_long %>%
     inner_join(anno_filtered,
                by = join_by(sample == Run))


df_long %>%
     filter(gene == "IL6") %>%
     group_by(dex) %>%
     summarise(mean_expression = mean(count))
```

A tibble: 8 × 3

Run <chr>	dex <chr>	cell <chr>
SRR1039508	untrt	N61311
SRR1039509	trt	N61311
SRR1039512	untrt	N052611
SRR1039513	trt	N052611
SRR1039516	untrt	N080611
SRR1039517	trt	N080611
SRR1039520	untrt	N061011
SRR1039521	trt	N061011

8 rows

A tibble: 6 × 5

gene <chr>	sample <chr>	count <dbl>	dex <chr>	cell <chr>
TSPAN6	SRR1039508	679	untrt	N61311
TSPAN6	SRR1039509	448	trt	N61311
TSPAN6	SRR1039512	873	untrt	N052611
TSPAN6	SRR1039513	408	trt	N052611
TSPAN6	SRR1039516	1138	untrt	N080611
TSPAN6	SRR1039517	1047	trt	N080611

6 rows

A tibble: 2 × 2

dex <chr>	mean_expression <dbl>
trt	98.5
untrt	146.0

2 rows

tidyr for extensive data manipulation

```
df_wide <- df_long %>%
     pivot_wider(
          names_from = “gene”,
          values_from = “count”)
```

take the data from the previous slide in long format
pivot wider
taking the new column names from the *gene* column
taking the values for the new columns from the *count* column

A tibble: 428,504 × 5

gene <chr>	sample <chr>	count <dbl>	dex <chr>	cell <chr>
TSPAN6	SRR1039508	679	untrt	N61311
TSPAN6	SRR1039509	448	trt	N61311
TSPAN6	SRR1039512	873	untrt	N052611
TSPAN6	SRR1039513	408	trt	N052611
TSPAN6	SRR1039516	1138	untrt	N080611
TSPAN6	SRR1039517	1047	trt	N080611
TSPAN6	SRR1039520	770	untrt	N061011
TSPAN6	SRR1039521	572	trt	N061011
TNMD	SRR1039508	0	untrt	N61311
TNMD				

A tibble: 8 × 53,566

sample <chr>	dex <chr>	cell <chr>	TSPAN6 <dbl>	TNMD <dbl>	DPM1 <dbl>	SCYL3 <dbl>	C1orf112 <dbl>	FGR <dbl>	CFH <dbl>
SRR1039508	untrt	N61311	679	0	467	260	60	0	3251
SRR1039509	trt	N61311	448	0	515	211	55	0	3679
SRR1039512	untrt	N052611	873	0	621	263	40	2	6177
SRR1039513	trt	N052611	408	0	365	164	35	0	4252
SRR1039516	untrt	N080611	1138	0	587	245	78	1	6721
SRR1039517	trt	N080611	1047	0	799	331	63	0	11027
SRR1039520	untrt	N061011	770	0	417	233	76	0	5176
SRR1039521	trt	N061011	572	0	508	229	60	0	7995

stringr - text manipulation

- package designed for manipulation of strings (text)

- we'll be mainly using 3 different functions:
  - str_detect(string, pattern) → detect a certain pattern in a string
  - str_replace(string, pattern, replacement) → replaces a certain pattern in a string
  - str_remove(string, pattern) → removes a certain pattern in a string

- this becomes a lot more powerful when working with regular expressions, *regex*

stringr - str_detect()


```
test_string <- “Hello, this is some writing”

str_detect(test_string, “Hello”)

str_detect(test_string, “Howdy”)
```

an example text bit

Detecting whether or not “Hello” is present in the string

Detecting whether or not “Howdy” is present in the string

```
df_long %>%
     filter(str_detect(gene, “MT-”)) %>%
     group_by(gene, dex) %>%
     summarise(mean_expression = mean(count))

df_long %>% filter(!str_detect(gene, “-AS”))
```

starting with the long expression data from before
filter to only genes containing “MT-” → mitochondrial
group by gene and treatment
generate a mean per group and gene

filter expression data to exclude antisense RNAs (“-AS”)

stringr - str_detect()


```
test_string <- “Hello, this is some writing”

str_detect(test_string, “Hello”)

str_detect(test_string, “Howdy”)
```

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

```
df_long %>%
     filter(str_detect(gene, “MT-”)) %>%
     group_by(gene, dex) %>%
     summarise(mean_expression = mean(count))

df_long %>% filter(str_detect(gene, “IL”))

df_long %>% filter(!str_detect(gene, “-AS”))
```

gene <chr>	dex <chr>	mean_count <dbl>
MT-CO3	trt	52225.25
MT-CO3	untrt	53137.25
MT-CYB	trt	61816.75
MT-CYB	untrt	56403.25
MT-ND1	trt	31134.75
MT-ND1	untrt	30542.50
MT-ND2	trt	32005.25
MT-ND2	untrt	31839.25
MT-ND3	trt	5399.75
MT-ND3	untrt	5685.00

11-20 of 76 rows

stringr - str_replace()


```
df <- aa_data

df %>%
     mutate(diagnosis = str_replace(diagnosis,
                                    “AAA”, “BAA”)

df <- anno_filtered

mutate(df, dex = str_replace(dex, “trt”, “treated”)
```

the aneurysma survey data

take the data
replace the AAA in diagnosis with BAA

expression data annotation

replace the “trt” in the *dex* column with “treated

A tibble: 8 × 3

Run <chr>	dex <chr>	cell <chr>
SRR1039508	untrt	N61311
SRR1039509	trt	N61311
SRR1039512	untrt	N052611
SRR1039513	trt	N052611
SRR1039516	untrt	N080611
SRR1039517	trt	N080611
SRR1039520	untrt	N061011
SRR1039521	trt	N061011

A tibble: 8 × 3

Run <chr>	dex <chr>	cell <chr>
SRR1039508	untreated	N61311
SRR1039509	treated	N61311
SRR1039512	untreated	N052611
SRR1039513	treated	N052611
SRR1039516	untreated	N080611
SRR1039517	treated	N080611
SRR1039520	untreated	N061011
SRR1039521	treated	N061011

gtsummary - a tool for summary tables

- package designed for summary tables
- high customizability
- ideal for a supplementary figure / table to give you an overview over your study cohort

Characteristic	CTRL N = 300[1]	AAA N = 300[1]	TAA N = 300[1]	p-value[2]
Abdominal diameter	2.00 (1.88, 2.11)	3.95 (3.44, 4.59)	1.98 (1.83, 2.11)	<0.001
Thoracic diameter	2.00 (1.84, 2.12)	2.02 (1.94, 2.09)	4.02 (3.79, 4.18)	<0.001
Age	62 (52, 71)	62 (53, 71)	62 (54, 71)	0.8
Male	148 (49%)	229 (76%)	231 (77%)	<0.001
Hypertension	67 (22%)	150 (50%)	129 (43%)	<0.001
Coronary artery disease	25 (8.3%)	53 (18%)	31 (10%)	0.001
Artheriosclerosis	157 (52%)	91 (30%)	112 (37%)	<0.001

[1] Median (Q1, Q3); n (%)

[2] Kruskal-Wallis rank sum test; Pearson's Chi-squared test

tbl_summary()

```
library(gtsummary)

aa_data %>%
  select(-bmi) %>%
  tbl_summary()
```

Characteristic	N = 900[1]
diagnosis	
AAA	300 (33%)
CTRL	300 (33%)
TAA	300 (33%)
abdominal_diameter	2.11 (1.94, 3.44)
thoracic_diameter	2.10 (1.96, 3.79)
age	62 (53, 71)
male	608 (68%)
hypertension	346 (38%)
cad	109 (12%)
artheriosclerosis	360 (40%)

[1] n (%); Median (Q1, Q3)

tbl_summary()

```
aa_data %>%
  select(-bmi) %>%
  tbl_summary(label = list(
     abdominal_diameter ~ "Abdominal diameter",
     thoracic_diameter ~ "Thoracic diameter",
     age ~ "Age",
     male ~ "Male",
     hypertension ~ "Hypertension",
     cad ~ "Coronary artery disease",
     artheriosclerosis ~ "Artheriosclerosis"))
```

Characteristic	N = 900[1]
diagnosis	
AAA	300 (33%)
CTRL	300 (33%)
TAA	300 (33%)
Abdominal diameter	2.11 (1.94, 3.44)
Thoracic diameter	2.10 (1.96, 3.79)
Age	62 (53, 71)
Male	608 (68%)
Hypertension	346 (38%)
Coronary artery disease	109 (12%)
Artheriosclerosis	360 (40%)

[1] n (%); Median (Q1, Q3)

columns

tbl_summary()


```
aa_data %>%
  select(-bmi) %>%
  tbl_summary(by = diagnosis,
     label = list(
     abdominal_diameter ~ "Abdominal diameter",
     thoracic_diameter ~ "Thoracic diameter",
     age ~ "Age",
     male ~ "Male",
     hypertension ~ "Hypertension",
     cad ~ "Coronary artery disease",
     artheriosclerosis ~ "Artheriosclerosis"))
```

pipe the aneurysm survey data
deselect bmi
get the table & split by diagnosis
supply custom labels for the columns

Characteristic	AAA N = 300[1]	CTRL N = 300[1]	TAA N = 300[1]
Abdominal diameter	3.95 (3.44, 4.59)	2.00 (1.88, 2.11)	1.98 (1.83, 2.11)
Thoracic diameter	2.02 (1.94, 2.09)	2.00 (1.84, 2.12)	4.02 (3.79, 4.18)
Age	62 (53, 71)	62 (52, 71)	62 (54, 71)
Male	229 (76%)	148 (49%)	231 (77%)
Hypertension	150 (50%)	67 (22%)	129 (43%)
Coronary artery disease	53 (18%)	25 (8.3%)	31 (10%)
Artheriosclerosis	91 (30%)	157 (52%)	112 (37%)

[1] Median (Q1, Q3); n (%)

tbl_summary()

```
aa_data %>%
  select(-bmi) %>%
     mutate(diagnosis =
          fct_relevel(diagnosis, "CTRL")) %>%
  tbl_summary(by = diagnosis,
     label = list(
     abdominal_diameter ~ "Abdominal diameter",
     thoracic_diameter ~ "Thoracic diameter",
     age ~ "Age",
     male ~ "Male",
     hypertension ~ "Hypertension",
     cad ~ "Coronary artery disease",
     artheriosclerosis ~ "Artheriosclerosis"))
```

pipe the aneurysm survey data
deselect bmi
relevel using the forcats package (focussed on next session)
get the table & split by diagnosis
supply custom labels for the columns

Characteristic	CTRL N = 300[1]	AAA N = 300[1]	TAA N = 300[1]
Abdominal diameter	2.00 (1.88, 2.11)	3.95 (3.44, 4.59)	1.98 (1.83, 2.11)
Thoracic diameter	2.00 (1.84, 2.12)	2.02 (1.94, 2.09)	4.02 (3.79, 4.18)
Age	62 (52, 71)	62 (53, 71)	62 (54, 71)
Male	148 (49%)	229 (76%)	231 (77%)
Hypertension	67 (22%)	150 (50%)	129 (43%)
Coronary artery disease	25 (8.3%)	53 (18%)	31 (10%)
Artheriosclerosis	157 (52%)	91 (30%)	112 (37%)

[1] Median (Q1, Q3); n (%)

tbl_summary()


```
aa_data %>%
  select(-bmi) %>%
     mutate(diagnosis =
          fct_relevel(diagnosis, "CTRL")) %>%
  tbl_summary(by = diagnosis,
     label = list(
     abdominal_diameter ~ "Abdominal diameter",
     thoracic_diameter ~ "Thoracic diameter",
     age ~ "Age",
     male ~ "Male",
     hypertension ~ "Hypertension",
     cad ~ "Coronary artery disease",
     artheriosclerosis ~ "Artheriosclerosis"))%>%
  italisize_labels() %>%
  add_p()
```

pipe the aneurysm survey data
deselect bmi
relevel using the forcats package (focussed on next session)
get the table & split by diagnosis
supply custom labels for the columns

Characteristic	CTRL N = 300[1]	AAA N = 300[1]	TAA N = 300[1]	p-value[2]
Abdominal diameter	2.00 (1.88, 2.11)	3.95 (3.44, 4.59)	1.98 (1.83, 2.11)	<0.001
Thoracic diameter	2.00 (1.84, 2.12)	2.02 (1.94, 2.09)	4.02 (3.79, 4.18)	<0.001
Age	62 (52, 71)	62 (53, 71)	62 (54, 71)	0.8
Male	148 (49%)	229 (76%)	231 (77%)	<0.001
Hypertension	67 (22%)	150 (50%)	129 (43%)	<0.001
Coronary artery disease	25 (8.3%)	53 (18%)	31 (10%)	0.001
Artheriosclerosis	157 (52%)	91 (30%)	112 (37%)	<0.001

[1] Median (Q1, Q3); n (%)

[2] Kruskal-Wallis rank sum test; Pearson's Chi-squared test

Summary

- Reading in data with readr
- data wrangling and manipulation with dplyr
- grouping and piping
- summary calculations
- string manipulation with stringr
- building summary tables using gtsummary

Practise Practise Practise

For practise purposes, I have compiled some questions in the github.

Please work through them and ask any questions you might encounter!

I’ll be happy to answer questions and receive feedback!

More practise questions!

Medium

M1: What is the minimal aortic diameter in the AAA condition?

M2: What is the mean abdominal diameter in the data set?

M3: What is the mean abdominal diameter by diagnosis? Considering only patients older than 60, how does this change?

M4: If you consider the median instead of mean and thoracic diameter instead of abdominal, what are the results?

Hard

H1: Refer back to the github and read in the expression data set. Merge them in the same fashion as above.

H2: Consider the gene *CXCL8*. What is the mean expression in the data set? What is the mean expression per treatment? What is the mean expression by treatment and cell line? Which sample shows the highest expression of *CXCL8*?

H2a: *CXCL8* is the gene name coding for the protein IL8. For your table, you want to replace the gene name with the protein name. Hint: use `str_replace()` and `mutate()`

H3: Generate an overview table of the aneurysm survey data using gtsummary.

H4: Split the table by diagnosis.

H5: Subset your study data to only include patients over the age of 50 and with a BMI larger than 22.