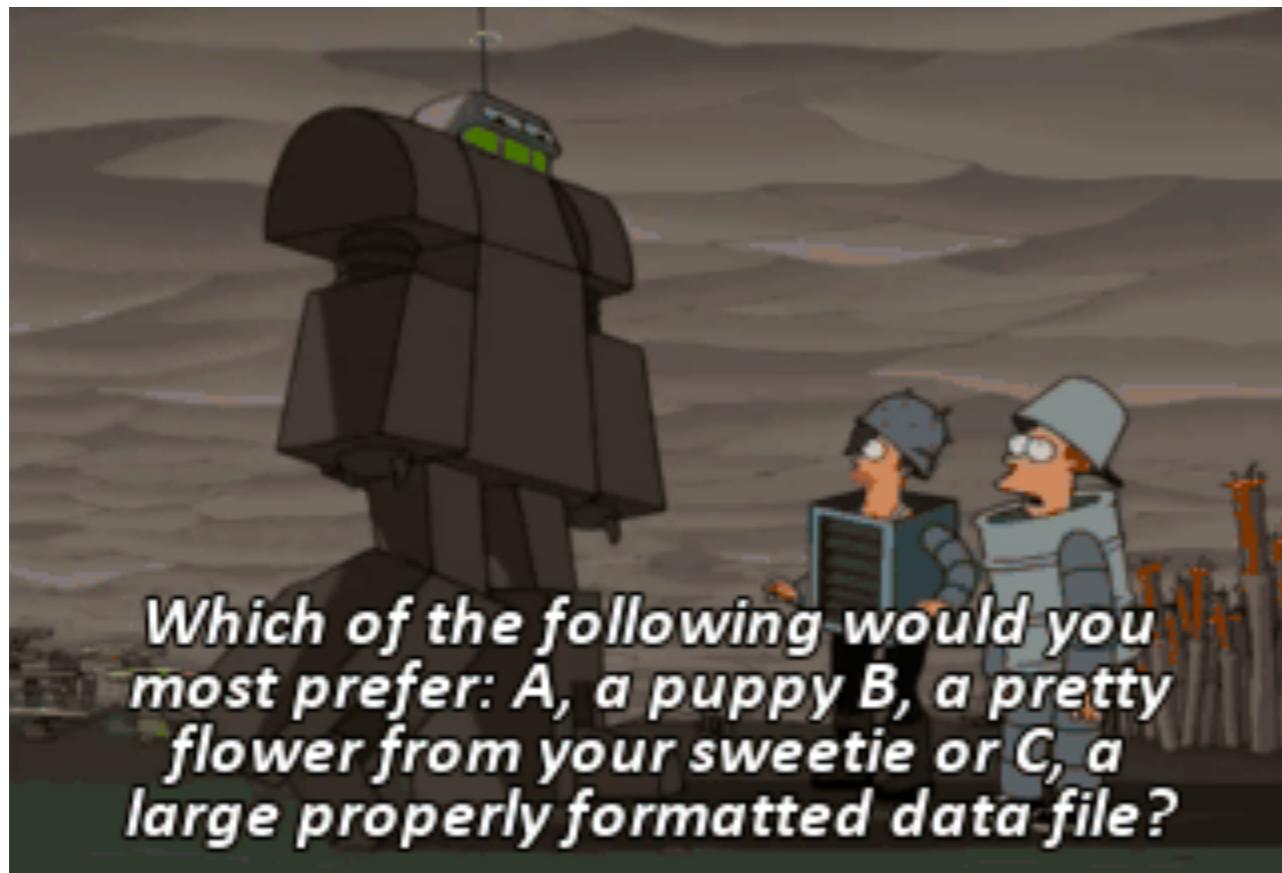


Basics of R: Data manipulation

Research Methods for Human Inquiry
Andrew Perfors

Still don't know what to get Shadow...

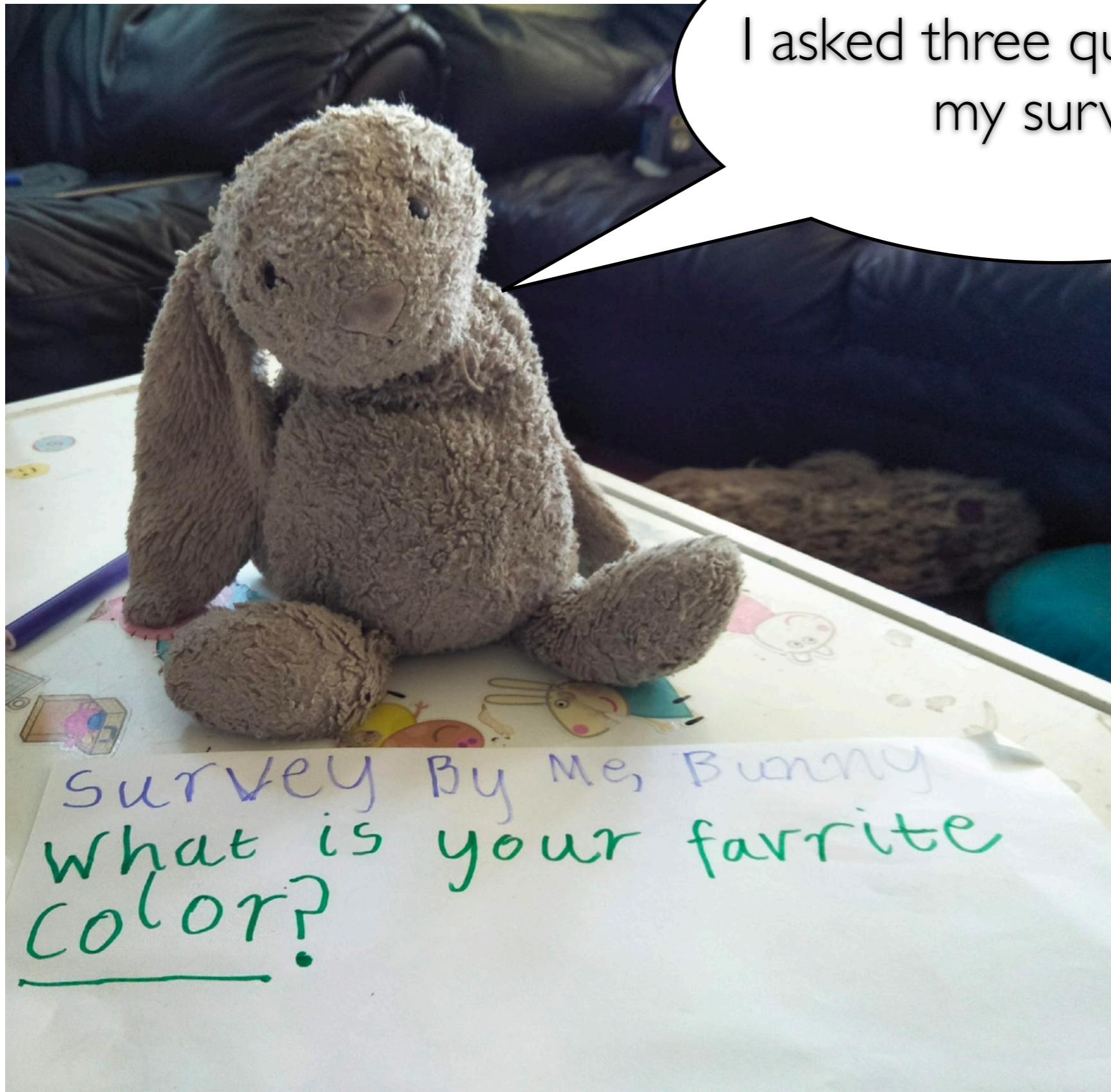


A data file! A data file!
Oh please....





I know! I'll get Shadow a dataset. I can ask all sorts of questions to our friends about what they really think and how they think in general.



I asked three questions on my survey.

1. What is your favourite colour?
2. How tall are you in cm?
3. Rank the following from best to worst:
(a) bunnies; (b) bears; (c) doggies

Let's take a look at Bunny's survey

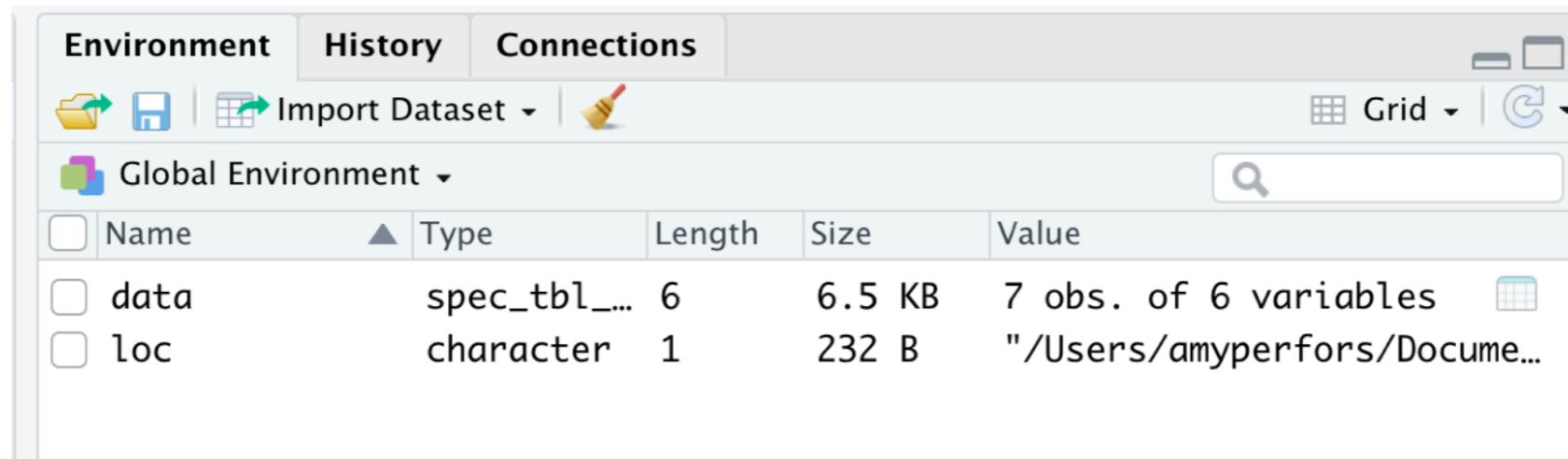


FLOPSY
Gladys
Shadow
LFB
doggy
cuddly PAWS
Bunny

1

Reading the file

You should have successfully loaded the file in the last video.



The screenshot shows the RStudio interface with the 'Environment' tab selected. The Global Environment pane displays two variables:

Name	Type	Length	Size	Value
data	spec_tbl(...)	6	6.5 KB	7 obs. of 6 variables
loc	character	1	232 B	"/Users/amyperfors/Docume..."

```
> data <- read_csv(file=loc)
```

```
— Column specification —
```

```
cols(  
  name = col_character(),  
  colour = col_character(),  
  height = col_double(),  
  bunnyrank = col_double(),  
  bearrank = col_double(),  
  doggyrank = col_double()  
)
```

Viewing the data

Type **data** at the prompt and you should see:

```
> data  
# A tibble: 7 x 6  
  name      colour height bunnyrank bearrank doggyrank  
  <chr>     <chr>   <dbl>    <dbl>     <dbl>     <dbl>  
1 bunny     grey     20        1         3         2  
2 gladly   purple    18        3         1         2  
3 flopsy   black    20        1         2         3  
4 shadow   red      20        1         3         2  
5 lfb      purple    24        1         2         3  
6 cuddly paws NA       24        NA        NA        NA  
7 doggie   blue     17        2         3         1
```

data is a special kind of variable called a tibble, which is basically a dataset.

Viewing the data

Type `data` at the prompt and you should see:

	name	colour	height	bunnyrank	bearrank	doggyrank
1	bunny	grey	20	1	3	2
2	gladly	purple	18	3	1	2
3	flopsy	black	20	1	2	3
4	shadow	red	20	1	3	2
5	lfb	purple	24	1	2	3
6	cuddly paws	NA	24	NA	NA	NA
7	doggie	blue	17	2	3	1

These are the seven vectors. I created each vector individually and put them all together using the following command:

```
> data <- tibble(name, colour, height, bunnyrank, bearrank, doggyrank)
```

Viewing the data

```
> name <- c("bunny", "gladly", "flopsy", "shadow", "lfb",  
"cuddly paws", "doggie")
```



Viewing the data

```
> colour <- c("grey", "purple", "black", "red", "purple", NA, "blue")
```

1. What is your
favourite colour?



Viewing the data

```
> height <- c(20,18,20,20,24,24,17)
```

2. How tall are you
in cm?



Viewing the data

```
> bunnyrank <- c(1,3,1,1,1,NA,2)
> bearrank <- c(3,1,2,3,2,NA,3)
> doggyrank <- c(2,2,3,2,3,NA,1)
```



3. Rank the following from best to worst:
(a) bunnies; (b) bears; (c) doggies

```
> data
# A tibble: 7 x 6
  name      colour height bunnyrank bearrank doggyrank
  <chr>    <chr>   <dbl>     <dbl>     <dbl>     <dbl>
1 bunny    grey      20        1         3         2
2 gladly  purple    18        3         1         2
3 flopsy  black     20        1         2         3
4 shadow  red       20        1         3         2
5 lfb     purple    24        1         2         3
6 cuddly paws NA      24        NA        NA        NA
7 doggie  blue      17        2         3         1
```

By being put into a tibble these vectors are given a special relationship to each other....

But they are still ordinary vectors, which I can pick out using the `$` operator

```
> data$height
[1] 20 18 20 20 24 24 17
```

`data$height` tells R to look for a vector called `height` stored in a tibble called `data`.

```
> data
# A tibble: 7 x 6
  name      colour height bunnyrank bearrank doggyrank
  <chr>    <chr>   <dbl>     <dbl>     <dbl>     <dbl>
1 bunny    grey      20        1         3         2
2 gladly   purple    18        3         1         2
3 flopsy   black     20        1         2         3
4 shadow   red       20        1         3         2
5 lfb      purple    24        1         2         3
6 cuddly paws NA       24        NA        NA        NA
7 doggie   blue      17        2         3         1
```

```
> data$colour
[1] "grey"   "purple" "black"  "red"    "purple" NA       "blue"
```

```
> data
# A tibble: 7 x 6
  name      colour height bunnyrank bearrank doggyrank
  <chr>    <chr>   <dbl>     <dbl>     <dbl>     <dbl>
1 bunny    grey     20        1         3         2
2 gladly   purple   18        3         1         2
3 flopsy   black    20        1         2         3
4 shadow   red      20        1         3         2
5 lfb      purple   24        1         2         3
6 cuddly paws NA      24        NA        NA        NA
7 doggie   blue     17        2         3         1
```

```
> data$bunnyrank
[1]  1  3  1  1  1 NA  2
```

```
> bunnyrank
Error: object 'bunnyrank' not found
```

Yep, this **\$** trick works
for all of them!

But you need the **\$**...

```
> data
```

```
# A tibble: 7 x 6
```

	name	colour	height	bunnyrank	bearrank	doggyrank
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	bunny	grey	20	1	3	2
2	gladly	purple	18	3	1	2
3	flopsy	black	20	1	2	3
4	shadow	red	20	1	3	2
5	lfb	purple	24	1	2	3
6	cuddly paws	NA	24	NA	NA	NA
7	doggie	blue	17	2	3	1

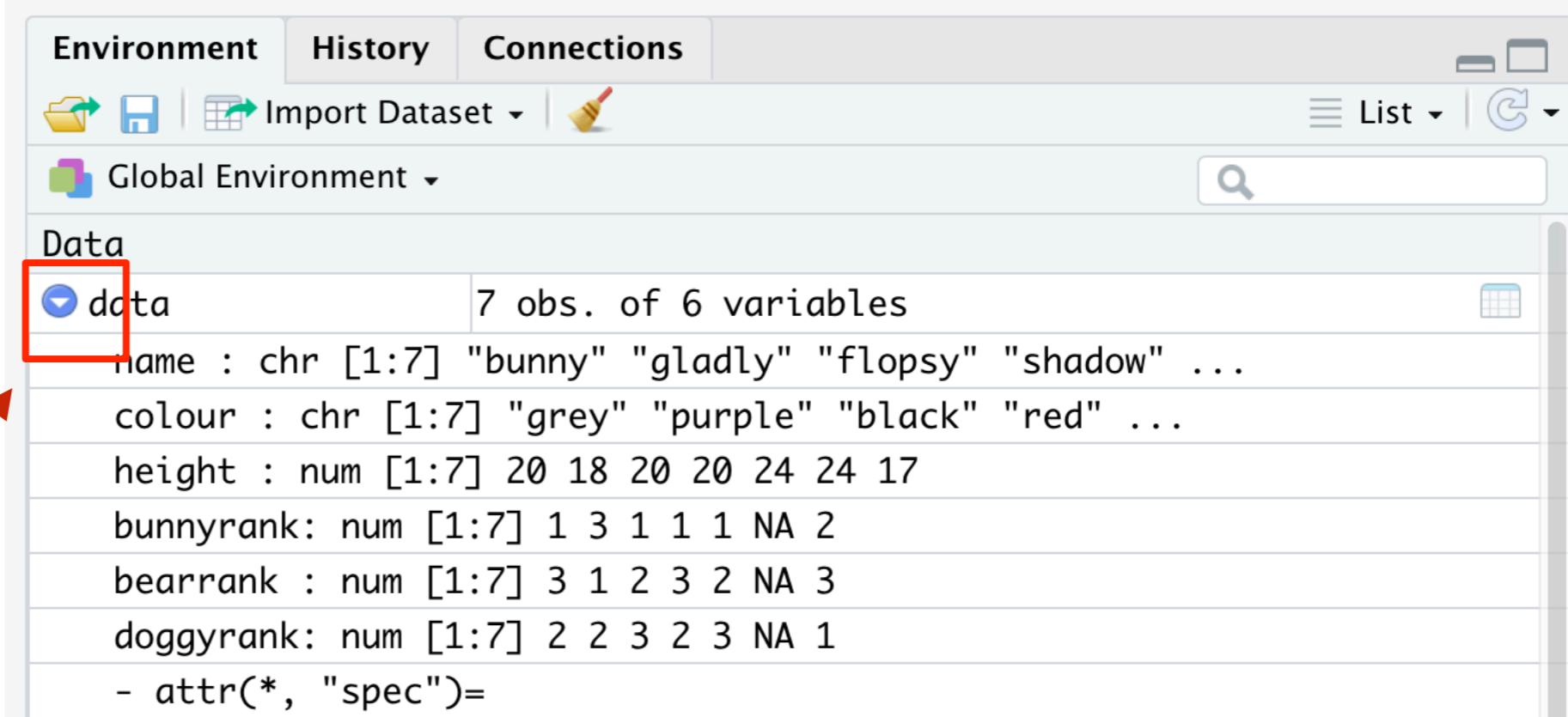
Hey
did you notice
that most people
like bunnies more?
Makes you think
huh?



Yeah, makes you
think you asked a
bunch of bunnies!



You can also view the dataset using RStudio



Environment History Connections

Import Dataset |

Global Environment |

Data

data	7 obs. of 6 variables
name : chr [1:7] "bunny" "gladly" "flopsy" "shadow" ...	
colour : chr [1:7] "grey" "purple" "black" "red" ...	
height : num [1:7] 20 18 20 20 24 24 17	
bunnyrank: num [1:7] 1 3 1 1 1 NA 2	
bearrank : num [1:7] 3 1 2 3 2 NA 3	
doggyrank: num [1:7] 2 2 3 2 3 NA 1	
- attr(*, "spec")=	

Clicking here (in List view) lets you see the entire dataset

You can also view the dataset
using RStudio

Clicking it again shows you the dataset in another panel.

The figure shows two panels of the RStudio interface. The left panel displays a data frame named 'data' as a grid of 7 rows and 6 columns. The columns are labeled: name, colour, height, bunnyrank, bearrank, and doggyrank. The right panel shows the 'Environment' tab of the global environment, listing the 'data' object along with its structure and contents.

	name	colour	height	bunnyrank	bearrank	doggyrank
1	bunny	grey	20	1	3	2
2	gladly	purple	18	3	1	2
3	flopsy	black	20	1	2	3
4	shadow	red	20	1	3	2
5	lfb	purple	24	1	2	3
6	cuddly paws	NA	24	NA	NA	NA
7	doggie	blue	17	2	3	1

Environment | History | Connections | Build | Git | List | C |

Global Environment |

Data

data 7 obs. of 6 variables

name : Factor w/ 7 levels "bunny", "cuddly paws", ... : 1 5 4 7 6 2 3
colour : Factor w/ 5 levels "black", "blue", ... : 3 4 1 5 4 NA 2
height : num 20 18 20 20 24 24 17
bunnyrank: num 1 3 1 1 1 NA 2
bearrank : num 3 1 2 3 2 NA 3
doggyrank: num 2 2 3 2 3 NA 1

Variables inside tibbles behave the same way as any other variable

```
> data$height  
[1] 20 18 20 20 24 24 17
```

```
> data$height + 100  
[1] 120 118 120 120 124 124 117
```

```
> data$height[1]  
[1] 20
```

You can change the values of variables in a tibble in the usual way...

```
> data$height[1] <- 99
> data
# A tibble: 7 x 6
  name      colour height bunnyrank bearrank doggyrank
  <chr>     <chr>   <dbl>    <dbl>     <dbl>     <dbl>
1 bunny     grey      99        1         3         2
2 gladly   purple     18        3         1         2
3 flopsy   black     20        1         2         3
4 shadow   red       20        1         3         2
5 lfb      purple     24        1         2         3
6 cuddly paws NA       24        NA        NA        NA
7 doggie   blue      17        2         3         1
```

Remember to change it back!

```
> data$height[1] <- 20
```

You can add variables to a tibble...

```
> data$tall <- data$height > 19
> data
# A tibble: 7 x 7
  name    colour height bunnyrank bearrank doggyrank tall
  <chr>   <chr>   <dbl>     <dbl>     <dbl>     <dbl>   <lgl>
1 bunny   grey      99        1          3        2      TRUE
2 gladly  purple    18        3          1        2      FALSE
3 flopsy  black     20        1          2        3      TRUE
4 shadow  red       20        1          3        2      TRUE
5 lfb     purple    24        1          2        3      TRUE
6 cuddly  paws     NA        NA         NA       NA      TRUE
7 doggie blue     17        2          3        1      FALSE
```

Removing them is even easier...

```
> data$tall <- NULL  
> data  
# A tibble: 7 x 6  
  name      colour height bunnyrank bearrank doggyrank  
  <chr>     <chr>   <dbl>    <dbl>    <dbl>    <dbl>  
1 bunny     grey     99        1        3        2  
2 gladly    purple   18        3        1        2  
3 flopsy    black    20        1        2        3  
4 shadow    red      20        1        3        2  
5 lfb       purple   24        1        2        3  
6 cuddly paws NA      24        NA       NA       NA  
7 doggie    blue     17        2        3        1
```

NULL is a special “value” in R that means “this variable does not exist” or “it has no value”. It is different to **NA**, which means “the variable exists (and in principle has a value), but the value is missing/unknown”

Selecting elements from a tibble

```
> data$height[1]  
[1] 20
```

`data$height` is a vector, and we're requesting the 1st element of it

```
> data[1, 3]  
# A tibble: 1 x 1  
height  
<dbl>  
1      20
```

`data` is a tibble, and we're requesting part of the tibble found in the 1st row, and the 3rd column

```
> data[1,"height"]  
# A tibble: 1 x 1  
height  
<dbl>  
1      20
```

`data` is a tibble, and we're requesting the part of the tibble found in the 1st row, and the column named “`height`”

Selecting a whole row

```
> data[3,]
# A tibble: 1 x 7
  name   colour height bunnyrank bearrank doggyrank tall
  <chr>  <chr>    <dbl>      <dbl>      <dbl>      <dbl> <lgl>
1 flopsy black     20         1          2         3 TRUE
```

Note the comma and then no number for columns, just the square bracket]. This empty bit after the comma indicates that we want to select all columns. If we don't specify that it assumes we're selecting by column:

```
> data[3]
# A tibble: 7 x 1
  height
  <dbl>
1     20
2     18
3     20
4     20
5     24
6     24
7     17
```

Selecting multiple rows

```
> data[c(1,2,4),]  
# A tibble: 3 x 7  
  name   colour height bunnyrank bearrank doggyrank tall  
  <chr>  <chr>   <dbl>     <dbl>     <dbl>     <dbl>     <lgl>  
1 bunny  grey      20        1         3        2 TRUE  
2 gladly purple    18        3         1        2 FALSE  
3 shadow red       20        1         3        2 TRUE
```

```
> data[1:4,]  
# A tibble: 4 x 7  
  name   colour height bunnyrank bearrank doggyrank tall  
  <chr>  <chr>   <dbl>     <dbl>     <dbl>     <dbl>     <lgl>  
1 bunny  grey      20        1         3        2 TRUE  
2 gladly purple    18        3         1        2 FALSE  
3 flopsy black     20        1         2        3 TRUE  
4 shadow red       20        1         3        2 TRUE
```

Selecting rows and columns?

```
> data[c(1,2,4),c("name","colour")]
# A tibble: 3 x 2
  name   colour
  <chr>  <chr>
1 bunny  grey
2 gladly purple
3 shadow red
```

```
> data[1:4,1:3]
# A tibble: 4 x 3
  name   colour height
  <chr>  <chr>   <dbl>
1 bunny  grey     20
2 gladly purple   18
3 flopsy black    20
4 shadow red      20
```

Selecting rows that match a criterion?

```
> small0nes <- data$height < 20
> data[small0nes,]
# A tibble: 2 x 7
  name   colour height bunnyrank bearrank doggyrank tall
  <chr>  <chr>    <dbl>      <dbl>      <dbl>      <dbl>     <lgl>
1 gladly purple     18         3          1          2 FALSE
2 doggie blue       17         2          3          1 FALSE
```

```
> data[data$height < 20,]
# A tibble: 2 x 7
  name   colour height bunnyrank bearrank doggyrank tall
  <chr>  <chr>    <dbl>      <dbl>      <dbl>      <dbl>     <lgl>
1 gladly purple     18         3          1          2 FALSE
2 doggie blue       17         2          3          1 FALSE
```

Exercises

1. Load the `bunnysurvey.csv` dataset. Make a new dataframe called `d` which is just a copy of `data`.
2. In `d`, add 1 to every entry for `height`. Then subtract it again.
3. Create a new variable in `d` called `dislikesDogs` which is `TRUE` if that person ranked dogs as #3, and `FALSE` otherwise.
4. Create a new variable in `d` called `inches` which gives the height in inches (hint: inches is cm divided by 2.54).
5. Select the first three rows (with all columns) out of `d`.
6. Select only the rows of `d` that contain an `NA` for the colour variable. (hint: use the `is.na()` function). For an extra challenge, try to select only the rows of `d` that do *not* contain an `NA` for the colour variable. (Remember our logical operators from last week).