Class 4 Lab*

Introduction to Data in R

Barry Grant

Version 221005

Instructions

Save this document to your computer and open it in a PDF viewer such as Preview (available on every mac) or Adobe Acrobat Reader (free for PC and Linux). Be sure to add your name and UC San Diego personal identification number (PID) and email below before answering all questions in the space provided.

Student Name	UCSD PID	UCSD Email
lisanne stouthart	A69036187	estouthart@ucsd.edu

Overview

This short lab supplement introduces basic tools for working with data.frames in R, as well as several commands for producing numerical and graphical summaries. We will build on these fundamentals next day when we delve into the powerful **ggplot2** package for making beautiful data visualizations.

Section 1: BRFSS as an example data.frame

The *Behavioral Risk Factor Surveillance System* (BRFSS) is an annual telephone survey of 350,000 people in the United States. The survey is designed to identify risk factors in the adult population and report emerging health trends.

For example, respondents are asked about their diet, weekly exercise, possible tobacco use, and healthcare coverage.

*http://thegrantlab.org/teaching/

Use the following command to download the dataset `cdc` from a URL. This dataset is a sample of 20,000 people from the survey conducted in 2000, and contains responses from a subset of the questions asked on the survey.

```
source("http://thegrantlab.org/misc/cdc.R")
```

Take a look at the Environment tab, where `cdc` should now be visible. Click the blue button next to the dataset name to view a summary of the 9 variables contained in the data matrix (see Figure 1).

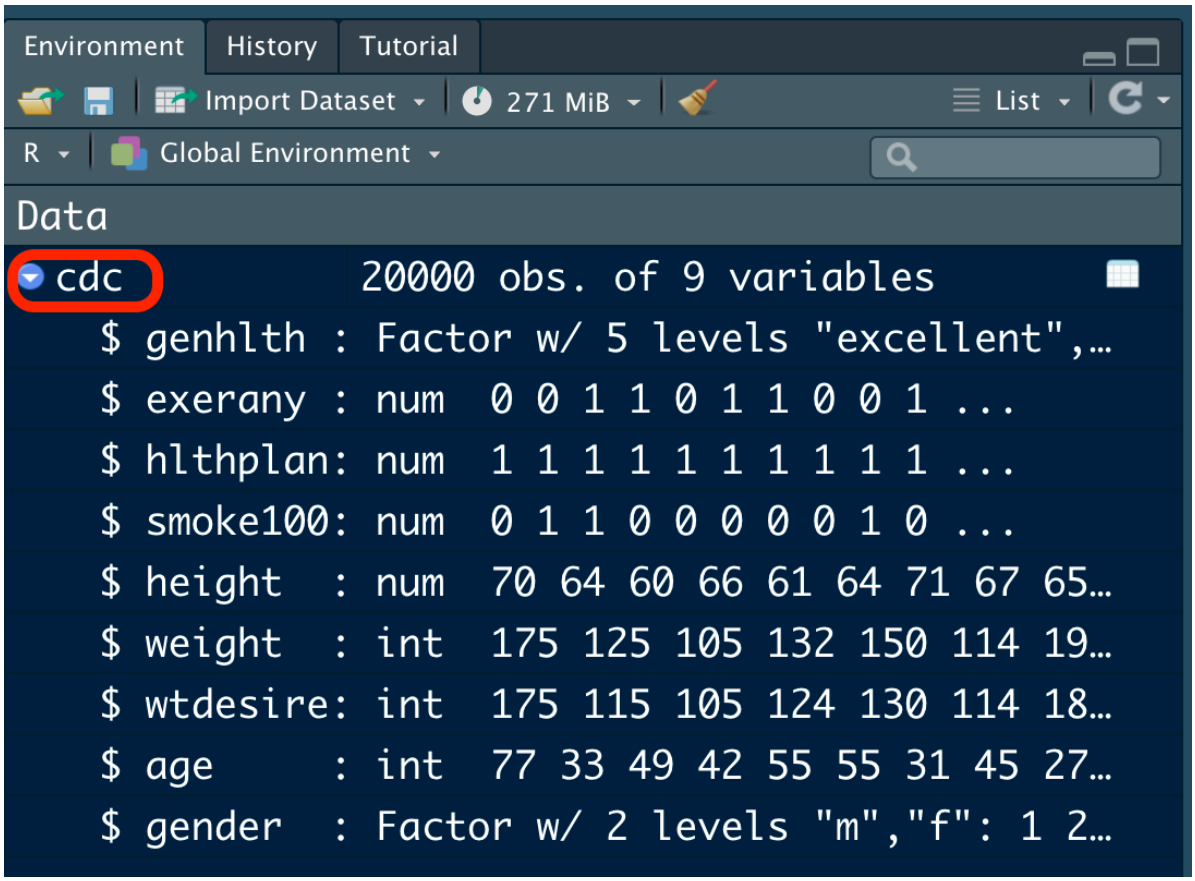

Figure 1: Available objects (e.g. varables and data) in your R session is listed in the **Environment** panel of RStudio. Click on these to see more details.

To view the dataset itself, click on the name of the dataset in the **Environment** panel; alternatively, enter the command:

```
View(cdc)
```

Each row of the **data.frame** represents a **case** and each column represents a **variable**. Each case is a person and each variable corresponds to a question that was asked in the survey.

- For `genhlth`, respondents were asked to evaluate their general health as either *"excellent"*, *"very good"*, *"good"*, *"fair"*, or *"poor"*.
- The variables `exerany`, `hlthplan`, and `smoke100` are binary variables, with responses recorded as either **0** for "no" and **1** for "yes": whether the respondent exercised in the past month, has health coverage, or has smoked at least 100 cigarettes in their lifetime.
- The other variables record the respondents' `height` in inches, `weight` in pounds, their desired weight (`wtdesire`), `age` in years, and `gender` (m for male and f for female).

Key-point: The $ operator in R is used to access variables (i.e. **columns**) within a data.frame; for example, `cdc$height` tells R to look in the `cdc` data.frame for the `height` variable:

```
head(cdc$height)
```

```
[1] 70 64 60 66 61 64
```

Note that the **`head()`** function prints out just the first 6 entries (rather than all 20,000). Guess what the **`tail()`** function does!

Q1 How would you "argue" with the **`tail()`** function to print out the last 20 `weight` values? Provide your code below:

```
last_20_weights <- tail(cdc$weight, 20)
print(last_20_weights)
```

Q2 Make a scatterplot of height vs weight using the `plot()` function. Add the code you used to generate this plot here:

> Tip
>
> If you already know how to use **ggplot** feel free to use it here (and elsewhere in this lab) in place of the "base R" `plot()` function.

```
plot(cdc$weight, cdc$height,
     main = "Scatterplot of Height vs Weight",
     xlab = "Height (inches)",
     ylab = "Weight (pounds)",
     pch = 1,
     col = "black")
```

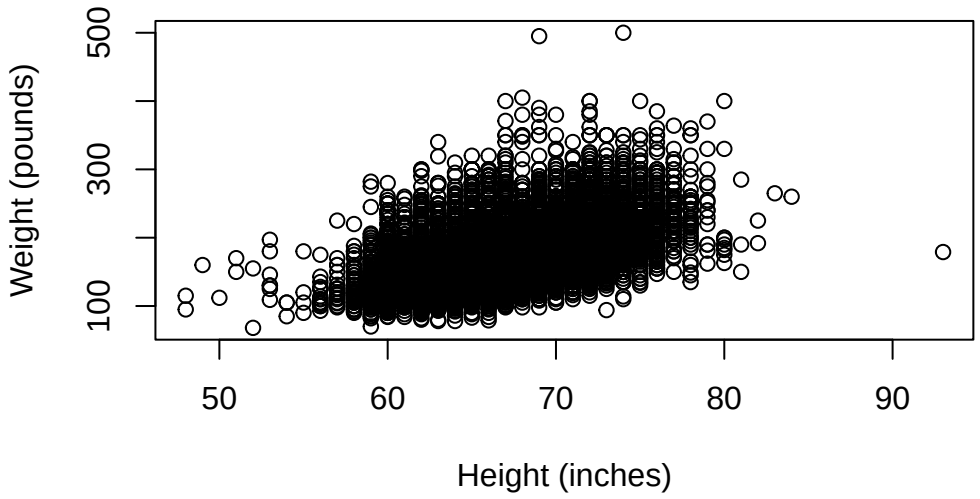

Figure 2: Scatterplot of height vs weight generated with the plot() function. Note custom axis labels generated with the xlab and ylab arguments. It is still rather basic and ugly - we will use ggplot to make beautiful versions next day ;-)

Q3 Do height and weight appear to be associated? If so are they positively associated or negatively associated?

```
# Calculate the correlation coefficient
correlation <- cor(cdc$weight, cdc$height, use = "complete.obs")
print(correlation)
# value = 0.5553222 -> so little to no association.
But if we look at the text below. 0.5 or larger is thought to represent a strong or large
correlation.
```

Side-Note: We can use the '**cor()** function to calculate the Pearson correlation of height and weight. A correlation coefficient of 0.1 is thought to represent a weak or small association; a correlation coefficient of 0.3 is considered a moderate correlation; and a correlation coefficient of 0.5 or larger is thought to represent a strong or large correlation.

Q4 What is the Pearson correlation value for height and weight?

> Tip
>
> Your code should look something like this: `cor(___$height, cdc$___)` with the `___` blanks filled in obviously.

```
# Calculate the correlation coefficient
correlation <- cor(cdc$weight, cdc$height, use = "complete.obs")
print(correlation)
# value = 0.5553222
```

Key-Point: When we use the $ notation we extract a vector from the data.frame

Many "base R" graphics functions work with vectors as input just like the `plot()` function. For example to make a histogram we can use:

```
hist(cdc$weight)
```

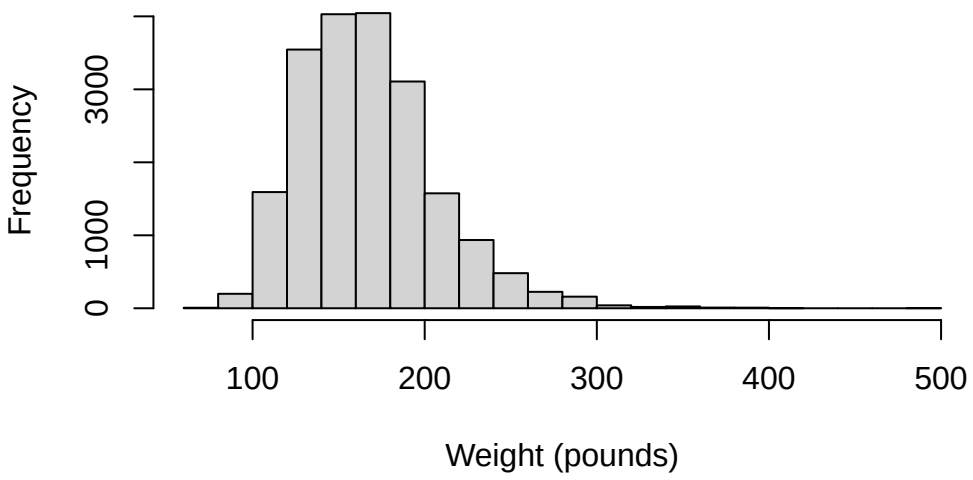

Figure 3: Distribution of weight values in the dataset

Now do the same for `height`.

```
hist(cdc$height)
```

> **Side-Note:** Do you notice a difference between the distributions for weight and height? Is one more scewed than the other? Why do you think this is?

Creating new vectors

The conversion from inches to meters is 1 in = .0254 m. The code below creates a new object `height_m` that records height in meters. Similarly, the conversion from pounds to kilograms is 1 lb = .454 kg.

```
# Create height.m
height_m <- cdc$height * 0.0254
```

Q5 Create a new object `weight_kg` that records weight in kilograms. Provide your code below:

```
weight_kg <- cdc$weight * .454
head(weight_kg)
```

Section 2: BMI

BMI is calculated as weight in kilograms divided by height in meters squared.

Q6 Create a new object `bmi` and make a plot of height vs BMI. Provide your code below and comment on whether height and BMI seem to be associated?

> Tip
>
> Your code should look something like this: `(weight_kg)/(height_m^2)` and then `plot(cdc$height, ___)` again with the blanks filled in.

```
bmi <- (weight_kg) / (height_m^2)

plot(cdc$height, bmi,
    main = "Scatterplot of Height vs BMI",
    xlab = "Height (inches)",
    vlab = "BMI".
```

Q7 Are height and BMI strongly associated? What are their correlation value?

> Tip
>
> Since height and BMI have a much weaker association, it is more useful to use BMI as a measure of obesity. Using BMI is one way to account for the fact that taller people tend to have more tissue and thus, weigh more than shorter people.

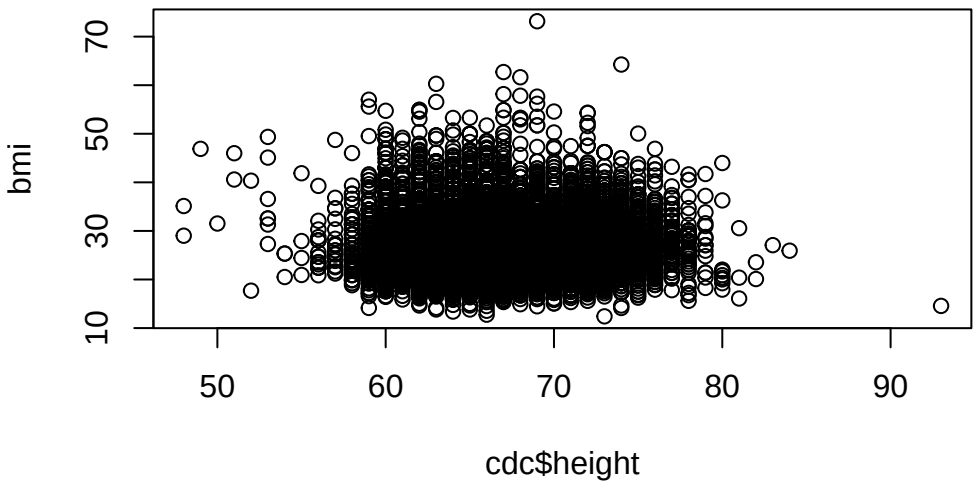

Figure 4: Height vs Body Mass Index (BMI)

```
correlation <- cor(cdc$height, bmi, use = "complete.obs")
print(correlation)
#Correlation = 0.03251694 -> so no correlation
```

Using logical vectors to count and subset

A BMI of 30 or above is considered obese. In R we can use code like the following to return a logical vector (i.e. TRUE and FALSE values) that can in turn help us find how many obese individuals are in our dataset.

```
# Note that I only have patience to print out the first 100 entries here
head(bmi >= 30, 100)
```

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

```
 [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
 [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
 [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
 [61] FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [73] FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [85] FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
 [97]  TRUE FALSE FALSE FALSE
```

A very useful trick that we will turn to over and over again is to treat logical vectors as equivalent to zero and one values. For example:

```
eg <- c(TRUE, TRUE, FALSE, FALSE)
sum(eg)
```

```
[1] 2
```

Q8 Can you use this summing of a logical vector approach to find out how many obese individuals there are in the dataset? Provide your code and answer below:

> Tip
>
> Try `sum(bmi >= 30)`

```
head(bmi >= 30, 100)
eg <- c(TRUE, TRUE, FALSE, FALSE)
sum(eg)

num_obese <- sum(bmi >= 30)
print(num_obese) #answer = 3897
```

To find the proportion of obese individuals we can use the following code:

```
sum(bmi >= 30)/length(bmi)
```

```
[1] 0.19485
```

Or to get percent value:

```
(sum(bmi >= 30)/length(bmi)) * 100
```

```
[1] 19.485
```

And to round this percent value to one significant figure:

```
round( (sum(bmi >= 30)/length(bmi)) * 100, 1)
```

```
[1] 19.5
```

Section 3: Accessing subsets of data using row and column indices

Row-and-column notation in combination with square brackets can be used to access a subset of the data. For example, to access the sixth variable (weight) of the 567th respondent, use the command:

```
cdc[567, 6]
```

```
[1] 160
```

To see the weight for the first ten respondents, use:

```
cdc[1:10, 6]
```

```
 [1] 175 125 105 132 150 114 194 170 150 180
```

If the column number is omitted, then all the columns will be returned for rows 1 through 10:

```
cdc[1:10, ]
```

```
     genhlth exerany hlthplan smoke100 height weight wtdesire age gender
1       good       0        1        0     70    175      175  77      m
2       good       0        1        1     64    125      115  33      f
3       good       1        1        1     60    105      105  49      f
4       good       1        1        0     66    132      124  42      f
5  very good       0        1        0     61    150      130  55      f
```

```
6  very good      1       1       0     64    114     114  55     f
7  very good      1       1       0     71    194     185  31     m
8  very good      0       1       0     67    170     160  45     m
9       good      0       1       1     65    150     130  27     f
10      good      1       1       0     70    180     170  44     m
```

Q9 Use bracket notation to make a scatterplot of height and weight for the first 100 respondents. Provide your code below.

> Tip
>
> There are multiple ways to do this—find one that works! For example `cdc[1:100, ]$height` or `cdc[1:100, "height"]` or `cdc[1:100, 5]` all return the first 100 height values. Which one do you prefer and why?

```
cdc[567, 6] #answer = 160
cdc[1:10, 6]
cdc[1:10, ]

plot(cdc$weight[1:100], cdc$height[1:100],
    main = "Scatterplot of Height vs Weight (First 100 Respondents)",
```

Section 4: Optional (for now)

Q10 How many obese individuals are male in the full dataset? Give your code and answer below:

> Tip
>
> Again there are multiple ways to do this. A very useful function in R is the `table()` function that will count up the number of different entries in a vector.
> For example `table( c("f","f","m"))`.
> To answer a question like this, first break it down into sub-steps like:
>
> - How do I first get at the data I need (i.e. the `gender` column)?
> - Then how do I subset this vector to have only `bmi >= 30` folks?
> - Then finally, how do I pass this to the `table()` function to count up the m and f values.

```
obese_logical <- bmi >= 30
unique(cdc$gender)
num_obese_males <- sum(obese_logical & (cdc$gender == "m"))
print(num_obese_males) #n = 1961
```

Discussion

With these codding skills there are lot's of other advanced questions we could ask and answer, for example:

- what portion of obese individuals judge their health to be "good"?
- Is this higher for male or female individuals?
- Is this difference significant?
- Do obese people smoke more, etc. etc.

Next class we will cover **ggplot** and in a couple of classes time we will cover the excellent **dplyr** package and the so-called **tidyverse** that together will make answering these types of questions much more tractable.

N.B. Note that it is exactly this sort of approach we will need to answer important bioinformatics questions like: "What proportion of my genes are up-regulated?"; "Of these, what proportion are from a certain key pathway?" etc.