SCHOOL OF PSYCHOLOGY
THE UNIVERSITY of ADELAIDE

Describing Data in R
To provide examples of calculating descriptive statistics.

Lauren Kennedy

School of Psychology, University of Adelaide

2016-Version 1

1 Assumed knowledge

This guide is to help you calculate the basic descriptions of various elements in your data frame. We will cover how to summarise the how many individuals in each group and how to find the mean and standard deviation of continuous variables.

If you're doing a practical, you'll need to be able to load your own data into R. If you're not sure how to do that, you can look up how to do this with the help guide on Loading your Practical Data in R. We will also assume you are relatively familiar with R (if you're not, refer to the guide "Getting started with R"), and familiar with using functions in R (if you are not , please refer to "Functions in R").

I strongly, strongly advise that you use R Studio for all of your statistical analyses.

2 What type of data do you have?

One of the first questions to ask when you would like to describe you data is what type of data you have. In psychology at Adelaide there are two common types. The first is a continuous or integer variable like your age or height, or how much you sleep you had last night. This is generally stored in R as a *numeric variable* (interger variables do exist, but for the most part we don't use them in your practical data). The second is a grouping variable like gender or hair colour or a intervention in a randomised control trial. This is generally stored in R as a *factor*.

We're going to calculate the descriptive statistics for some data in R. This data is already on every installation of R. It's hidden away, so you can't see it in the environment panel in Rstudio, but don't worry, it's there. To do your practicals you will need to load the practical data.

First off we are going to look at a data frame called sleep. If you type:

```
View(sleep) #Note View has a captial V
```

You can see what is in the data frame. This data frame has 3 columns and 20 rows. Each of the 10 participants received both intervention 1 and then intervention 2 (or vice versa). The first column, extra, is the post-pre difference in sleep duration (measured in hours). The second column, group, is the allocated intervention, coded as 1 or 2. The third column is the amount of sleep the participant gained in that particular intervention. So the first row says the first participant received treatment 1 and had an increase of .7hours.

To see what the type of data each column is stored is, we need to find out what structure each is saved as. To do this we use the **str()** function.

```
str(sleep)
```

This uses a function. For help with functions see the help guide "Functions in R". First we need to say **str** as it is the name of the *function*. We need *sleep* as we need to specify the **dataframe**. We're telling the function to look in the **dataframe** *sleep* and find the structure of each column. You should get the following output:

```
  'data.frame': 20 obs. of  3 variables:
 $ extra: num  0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0 2 ...
 $ group: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ ID   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

First we get a line that confirms that *sleep* is, in fact, a data frame. Then we are told that this dataframe has three variables (columns) with 20 observations (rows) of each variable. Then it goes through and lists each column of variable name stored in this dataframe and tells us what type of variable it as stored as. For example, we can *extra* is a numeric variable because it says **num** after it. *group* and *ID* are factor level variables. We can see that *group* has two different options, either 1 or 2. This corresponds to the different levels of group.

3 Describing Continuous variables

Our first step in this guide will be to describe the continuous variable *extra* in the data frame sleep. We'll start by taking the mean and standard deviation for *extra* variable as a whole, and then we will find the mean and standard deviation of the *extra* variable for each of the intervention groups *1* and *2*.

The **mean** and **sd** function can be used to find the mean and standard deviation of the *extra* function as follows:

```
mean(sleep$extra)
```

```
sd(sleep$extra)
```

Here the **mean** or the **sd** function tells R we want to calculate the mean or the standard deviation respectively. The **sleep$extra** says that we want R to find the **sleep** dataframe and select the **extra** column. You should have found the mean is 1.54 and the sd is 2.02.

Now we can use those same functions to calculate the mean and standard deviation of each group. To do this we need to be able to tell R to find the data frame *sleep* and find the column *sleep* but only return the elements of sleep that correspond to the column *group* equalling "1" or "2". The first line of code below does this for the first group followed by a line to do this for the second group. The square brackets are used to tell R that we will be specifying which of the **sleep$extra** numbers to keep, and bit inside the square brackets **sleep$group=="1"** tells R to look for where the *group* variable is equal to 1.

```
sleep$extra[sleep$group=="1"]
```

```
sleep$extra[sleep$group=="2"]
```

Now we can combine those above commands to find the mean and the standard deviation for the *extra* variable where group equals "1" and "2" respectively.

```
mean(sleep$extra[sleep$group=="1"])
```

```
mean(sleep$extra[sleep$group=="2"])
```

```
sd(sleep$extra[sleep$group=="1"])
```

```
sd(sleep$extra[sleep$group=="2"])
```

In this section we found the mean and standard deviation of a continuous variable, and then the mean and variances of a grouping variable broken into two different conditions.

4 Describing Grouping Variables

In the previous section we looked at how we might summarise a continuous variable by taking its mean and standard deviation. In this section we will look at how we count the number in each level of a factor variable like *group* in the *sleep* dataframe. To do this we will use the **summary** function, and tell R we would the like to take the summary of the *group* column in the *sleep* data frame as follows:

```
summary(sleep$group)
```

This command produces the following output. It tells us that there were ten instances of intervention "1" and ten instances of "2".

```
 1  2
10 10
```

5 Conclusion

In this short guide we looked how how we might chain commands in R to calculate descriptive statistics of our dataframe sleep.