# Z-SCORES

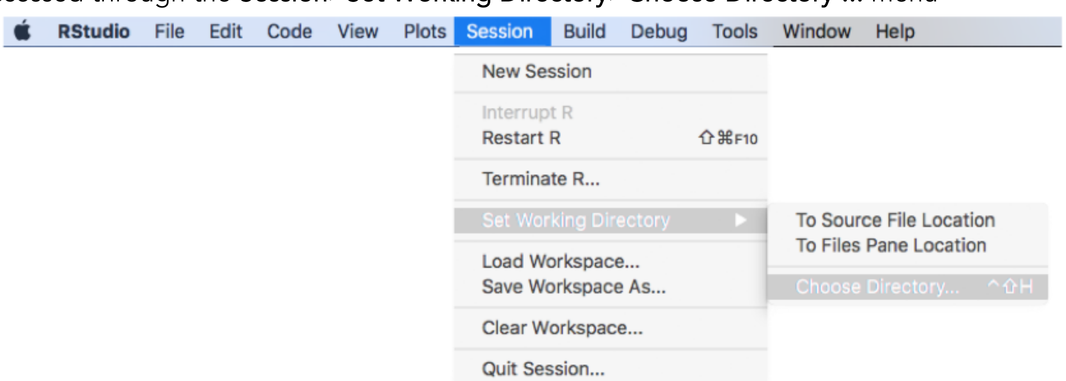
This handout is one of a series that accompanies *An Adventure in Statistics: The Reality Enigma* by me, Andy Field. These handouts are offered for free (although I hope you will buy the book).[1]

## Overview

In this handout we will look at how to do the procedures explained in Chapter 6 using R an open-source free statistics software. If you are not familiar with **R** there are many good websites and books that will get you started; for example, if you like *An Adventure In Statistics* you might consider looking at my book *Discovering Statistics Using R*.

### Some basic things to remember

- RStudio: I assume that you're working with RStudio because most sane people use this software instead of the native **R** interface. You should download and install both R and RStudio. A few minutes on Google will find you introductions to R Studio in the event that I don't write one, but these handouts don't particularly rely on R Studio except in setting the working directory (see below).
- Dataframe: A dataframe is a collection of columns and rows of data, a bit like a spreadsheet (but less pretty)
- Variables: variables in dataframes are referenced using the $ symbol, so *catData$fishesEaten* would refer to the variable called *fishesEaten* in the dataframe called *catData*
- Case sensitivity: **R** is case sensitive so it will think that the variable *fishesEaten* is completely different to the variable *fisheseaten*. If you get errors make sure you check for capitals or lower case letters where they shouldn't be.
- Working directory: you should place the data files that you need to use for this handout in a folder, then set it as the working directory by navigating to that folder when you execute the command accessed through the **Session>Set Working Directory>Choose Directory ...** menu




---

[1] This document is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, basically you can use it for teaching and non-profit activities but not meddle with it.

## Packages used in this chapter

We do not need to use any packages for this chapter.

### The data

In the book, Zach has met with Celia and is trying to get a job at JIG:SAW. Celia gives him three tests to measure his general intelligence (IQ), his emotional intelligence (EQ) and his creativity. She then compares Zach's scores to other workers at JIG:SAW. Therefore, we need to access the data that Celia used, and it is in a csv file on the companion website for the book. Execute the command below, which uses the *file.choose()* function to open a dialog box so that you can navigate to the file that you want to open, which in this case will be *Ais Ch 06 Celia's Recruitment Data.csv*. The rest of the command tells **R**to import this file into a dataframe called **recruit**.

```
recruit<-read.csv(file.choose())
```

We can use the *head()* function to look at the top of the data file. Executing the command below will display the top 10 lines of the **recruit** dataframe. If you want to look at a different number of lines then change the number 10, and if you want to see the whole data then execute the name of the dataframe (**recruit**) rather than using the *head()* function.

```
head(recruit, 10)
##     ID        Employee  Job_Type Footspeed Strength     Vision    Sex
## 1   29 JIGSAW Employee Code 1318      6.63     1931 0.01000000   Male
## 2   46 JIGSAW Employee Code 1318     19.68     2018 0.04000000   Male
## 3   48 JIGSAW Employee Code 1318     26.05     1894 0.17000000 Female
## 4   71 JIGSAW Employee Code 1318      1.19     1083 0.03517639 Female
## 5  103 JIGSAW Employee Code 1318     20.01     2540 0.04000000   Male
## 6  110 JIGSAW Employee Code 1318     19.98     2077 0.15000000   Male
## 7  117 JIGSAW Employee Code 1318      9.63     2008 0.12000000   Male
## 8  126 JIGSAW Employee Code 1318      8.86     2008 0.17000000   Male
## 9  127 JIGSAW Employee Code 1318      7.06     2004 0.07000000   Male
## 10 137 JIGSAW Employee Code 1318     27.44     1872 0.02000000 Female
##    Intelligence Creativity  EI
## 1           176         55 172
## 2           150         32 183
## 3           183          4 176
## 4            78         39 125
## 5           180         24 200
## 6           153          4 183
## 7           165         52 173
## 8           133         17 175
## 9           162         26 191
## 10           97         44 162
```

Looking at the top 10 cases you will see that the dataframe contains 7 variables:

- **ID**: identifies the participant.
- **Employee**: codes whether a participant worked for JIG:SAW or not (this variable is known as a factor).
- **Job_Type**: codes the type of job the participant had (all of the top 10 were scientists but if you look at the whole data you will see other job descriptions).
- **Footspeed**: this variable contains the participants' footspeed (mph).
- **Strength**: this variable contains the participants' maximal push force (N).
- **Vision**: this variable contains the participant's visual acuity scores.

- **Sex**: this variable codes the participant's biological sex as male or female.
- **Intelligence**: this variable contains the scores on the IQ test that Zach also completed.
- **Creativity**: this variable contains the scores on the creativity test that Zach also completed.
- **EI**: this variable contains the scores on the emotional intelligence test that Zach also completed.

## Standarizing a score

R can be used like a calculator. For example, the equation to convert a raw score to a *z*-score in a sample is:

$$z = \frac{X - \overline{X}}{s}$$

When The Head asks Zach to convert a score of 22 to a *z*-score using a mean of 18, and standard deviations of 10 and 5, we can do this in **R**by typing out the equation and executing it:

```
(22-18)/10
## [1] 0.4
(22-18)/5
## [1] 0.8
```

The brackets are important to make sure that **R**knows what is on the top half and bottom half of the fraction. Without the brackets, **R**will apply BODMAS and for the first line divide 18 by 10 and subtract the result from 22 (and we'd get a result of 20.2), which is not what we want.

We can also use the result of the *z*-score equation to create an object that we can use later on.

```
z1<-(22-18)/10
z2<-(22-18)/5

z1; z2
## [1] 0.4
## [1] 0.8
```

The first line creates an object called *z1* that is the result of the calculation of converting 22 to a *z*-score based on a mean of 18 and standard deviation of 10. The second line created an object *z2* that converts 22 to a *z*-score based on a mean of 18 and standard deviation of 5. To display the objects we execute their names, and this is what line 3 does (note we separate the names with a semi colon so that **R**treats them as separate commands).

## Using *z*-scores to compare scores

Zach scored 125 for IQ, 30 on creativity and 165 on EI. Celia wanted to compare his scores to the distribution of scores for worker's at JIG:SAW. We have these scores in the dataframe **recruit**. Let's start with Zach's IQ score of 125. To convert this score to *z* we need to subtract the mean of the JIG:SAW scores for IQ, which we can obtain with *mean(recruit$Intelligence)*, and then divide the result by the standard deviation of the JIG:SAW scores, which we can obtain with *sd(recruit$Intelligence).* Putting this together we can convert Zach's *z*-score by executing the following commands:[2]

---

[2] The book chapter uses values for the mean and standard deviation of the JIG:SAW scores that are rounded to whole numbers (to keep things simple). Therefore, the answer we get here will be slightly different because the functions for the mean and SD will return values that are not rounded. If you want

```
zachIQz<-(125 - round(mean(recruit$Intelligence)))/round(sd(recruit$Intelligence))
zachIQz
## [1] -0.2820513
```

The first line creates an object called *zachIQz* which contains Zach's IQ score converted to a *z*-score using the mean and standard deviation of the intelligence scores from Celia's JIG:SAW data. We can compute *z*s for Zach's creativity and EI scores in the same way, but we can also do all of the scores in one go using vectors. In R, the *c()* function collects objects together and we can use this function to first collect together Zach's three scores, then collect together the corresponding means from Celia's data, then collect the corresponding standard deviations. We can then use these collections to convert all of Zach's scores in one go. Let's see how this will work:

```
zachScores<- c(125, 30, 165)

jigsawMeans<-round(c(mean(recruit$Intelligence), mean(recruit$Creativity), mean(recruit$EI)))
jigsawMeans
## [1] 136  32 168
jigsawSDs<-round(c(sd(recruit$Intelligence), sd(recruit$Creativity), sd(recruit$EI)))
jigsawSDs
## [1] 39 19 32
zachZ<-(zachScores-jigsawMeans)/jigsawSDs
zachZ
## [1] -0.2820513 -0.1052632 -0.0937500
```

The first line collects Zach's three scores together in an object called *zachScores* in the order IQ, creativity, and EI. The second line creates an object called *jigsawMeans*, which uses the variables in the **recruit** dataframe that reflect the JIG:SAW intelligence, creativity and EI scores. For each variable we use the *mean()* function to obtain the mean for each variable, and then collect them together using *c()* in the same order as Zach's scores. I have then wrapped that within the *round* function to get whole numbers (as are used in the book). The third line displays this collection of mean scores. The fourth and fifth lines do much the same to extract the standard deviations of the JIG:SAW scores.

The final two lines use the *z*-score equation to convert Zach's three scores into *z*-scores by subtracting the means that we extracted, and the dividing the results by the standard deviations extracted. **R**does this calculation item-by-item for the collection of Zach's scores so it is important that the means and standard deviations from the JIG:SAW data are in the same order as Zach's scores (that is IQ, then creativity, then EI). The final line will display the three *z*-scores (which are now stored in an object called *zachZ*) so that you can compare them to the values in the book.

## Comparing distributions

Rather than converting individual scores we can convert distributions of scores to *z* so that we can compare variables that use different scales of measurement. In the chapter, creativity has a maximum score of about 60 whereas for IQ it is more like 200 (a score that is impossible on the creativity scale). By converting both variables to *z*-scores we can compare them because we change the units of measurement to be standard deviations (both variables will have a mean of 0 and standard deviation of

---

to replicate the values in the book we can change the command to *zachIQz<-(125 - round(mean(recruit$Intelligence)))/round(sd(recruit$Intelligence))*. All we have done is place the functions for the mean and sd into the *round()* function, which will round them to whole numbers.

1). This transformation is easy to do in **R**because we can, as before, use the *mean* and *sd()* functions to get the mean and standard deviation for a variable, and then rather than inputting a single value to be converted we put in the name of a variable instead: this will convert the entire set of scores for that variable. Let's have a go by executing this code:

```
recruit$zIQ <- (recruit$Intelligence-mean(recruit$Intelligence))/sd(recruit$Intelligence)
recruit$zCreate <- (recruit$Creativity-mean(recruit$Creativity))/sd(recruit$Creativity)
recruit$zEI <- (recruit$EI-mean(recruit$EI))/sd(recruit$EI)

head(recruit, 10)
##     ID        Employee  Job_Type Footspeed Strength     Vision    Sex
## 1   29 JIGSAW Employee Code 1318      6.63     1931 0.01000000   Male
## 2   46 JIGSAW Employee Code 1318     19.68     2018 0.04000000   Male
## 3   48 JIGSAW Employee Code 1318     26.05     1894 0.17000000 Female
## 4   71 JIGSAW Employee Code 1318      1.19     1083 0.03517639 Female
## 5  103 JIGSAW Employee Code 1318     20.01     2540 0.04000000   Male
## 6  110 JIGSAW Employee Code 1318     19.98     2077 0.15000000   Male
## 7  117 JIGSAW Employee Code 1318      9.63     2008 0.12000000   Male
## 8  126 JIGSAW Employee Code 1318      8.86     2008 0.17000000   Male
## 9  127 JIGSAW Employee Code 1318      7.06     2004 0.07000000   Male
## 10 137 JIGSAW Employee Code 1318     27.44     1872 0.02000000 Female
##    Intelligence Creativity  EI         zIQ     zCreate        zEI
## 1           176         55 172  1.03138653  1.22495925  0.1201691
## 2           150         32 183  0.36597587  0.01577201  0.4680269
## 3           183          4 176  1.21053556 -1.45628202  0.2466629
## 4            78         39 125 -1.47669982  0.38378551 -1.3661327
## 5           180         24 200  1.13375740 -0.40481486  1.0056255
## 6           153          4 183  0.44275402 -1.45628202  0.4680269
## 7           165         52 173  0.74986664  1.06723917  0.1517925
## 8           133         17 175 -0.06910034 -0.77282836  0.2150394
## 9           162         26 191  0.67308848 -0.29966814  0.7210145
## 10           97         44 162 -0.99043818  0.64665231 -0.1960653
```

The first line creates an object called *recruit$zIQ*. In fact, what is happening here is that **R**is creating an object called *zIQ* within the dataframe **recruit**. In other words it is creating a variable. So what we're doing here is adding a variable called **zIQ** to the **recruit** dataframe. The contents of that variable is determined by the right hand side of the command, and this tells **R**to take the scores in the variable **Intelligence**, subtract from each one the mean of intelligence scores and divide the result by the standard deviation of intelligence scores. The second and third lines do the same thing for the creativity and EI scores. The final line will display the first 10 cases of the dataframe **recruit** so that you can see that 3 variables have been added to the end called **zIQ**, **zCreate**, and **zEI** and that these contain the *z*-scores for IQ, creativity and EI respectively.

The above approach took three distributions and converted them all to *z*-scores (that is they all have a mean of 0 and standard deviation of 1). This transformation makes them comparable. An alternative is to rescale the distribution of one variable so that it has a mean and standard deviation that is the same as another variable. For example, in the book Celia shows Zach how to rescale the creativity scores so that they have the same mean and standard deviation as the IQ scores ($M = 136$, $SD = 39$). She did this by first converting the creativity scores to *z*-scores (which we have done above and these scores are in the variable **zCreate**) and then multiplying them by the standard deviation of IQ scores and then adding the mean of IQ scores. Essentially we're using the equation to convert to *z* but rearranged:

$$X = zs + \overline{X}$$

To do this in **R**we could execute this command:

```
recruit$Create2 <- recruit$zCreate*sd(recruit$Intelligence) + mean(recruit$Intelligence)

mean(recruit$Create2); sd(recruit$Create2)
## [1] 135.7
## [1] 39.07361
```

The first line creates a variable called **Create2** in the **recruit** dataframe. It creates this variable by taking the *z*-scores for creativity (*recruit$zCreate*), multiplying them by the standard deviation for the IQ score(*sd(recruit$Intelligence)*) and then adding the mean of IQ scores(*mean(recruit$Intelligence)*). The resulting variable is appended as a new column in the **recruit** dataframe. I have also included commands to display the mean and standard deviation of the variable that we just created so that you can verify that they are approximately 136 and 39, the same values as for IQ. If you want to see the values within the dataframe execute the name of the dataframe.

---

This handout is written to be used in conjunction with: Field, A. P. (2016). An adventure in statistics; the reality enigma. London: Sage.