Foundations of Data Analysis

The University of Texas at Austin

R Tutorials: Week 5

Linear Models

In this R tutorial, we're going to learn how to run a linear regression model in R. And to do this, we're going to use the world records data set, so I need to import that first by selecting worldrecords.csv with the Import Dataset option in RStudio. Let's simplify the data frame name to just be WR.

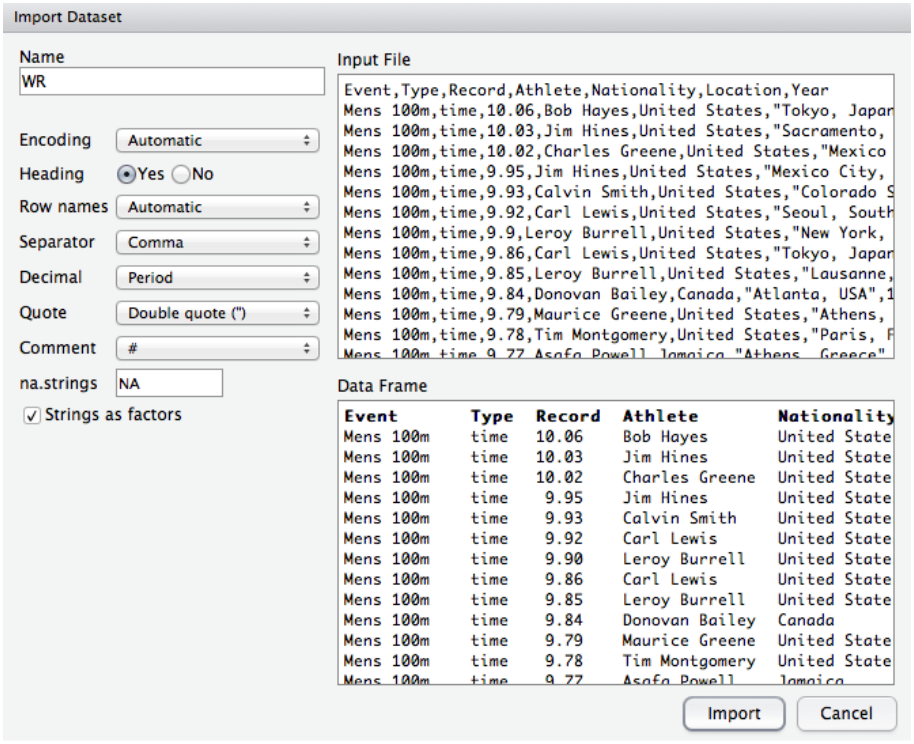


Figure 1: Importing Data in RStudio.

Alternatively, we can import the dataset with the "**read.csv()**" function:

```
WR <- read.csv("~/Desktop/SDSFoundations/WorldRecords.csv")
```

And we can see that this data set contains the world record time or distance of a various track and field events for both men and women. And for this exercise, we're going to be using the men's 800-meter event. So we're going to be looking at the world-record time, which is housed in this "record" variable, as well as the year that the world record was broken. And we're going to see if we can model the record time as a function of the year. So the first thing we're going to have to do is subset our data based off this event variable and only pick out the rows where the event is the men's 800-meter.

So let's make a new data frame called mens800 and pull out the cases from World Record based on a logical indexing statement. So we're going to say, World Record Event equals equals Men's 800 meter. And we want every column from our original data set, so we don't index the columns at all.

```
table(WR$Event)
```

```
##
##       Mens 100m       Mens 800m       Mens Mile  Mens Polevault
##              17              24              32              55
##    Mens Shotput Mens TripleJump     Womens 100m     Womens 800m
##              39              25              10              29
##     Womens Mile  Womens Shotput
##              13              41
```

```
mens800 <- WR[WR$Event == "Mens 800m", ]
```

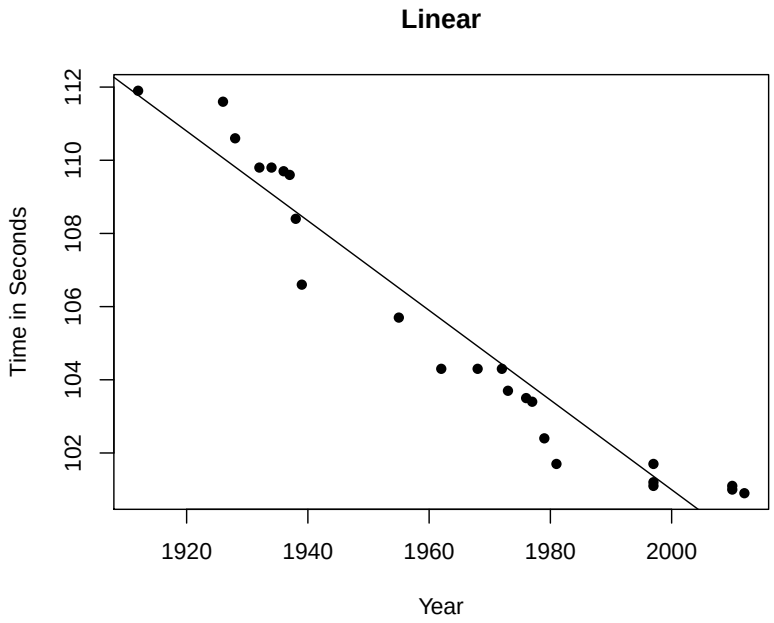
So here we have a subset of our original data set– 24 observations only.

To run a linear regression model in R, we can use the built-in function called "**linFit()**" from the **SDSFoundations** package available on the course website. To get access to the package **linFit()**, we'll need to call our SDSFoundations package with the **library()** function first.

```
library(SDSFoundations)
```

All we have to do is give **linFit()** two arguments. The first argument is going to be the vector of our independent variable, and the second will be the vector of our dependent variable. So we want to model the record time as a function of year. So we're going to give it the year from our Men's 800 data frame, and then also the record time will be our dependent variable.

```
linFit(mens800$Year, mens800$Record, xlab = "Year",
    ylab = "Time in Seconds")
```



Now if we've run this, we'll see a couple things happen. First, we see a nice scatter plot of our data. So we see year – our independent variable – along the horizontal axis, the record time along the vertical axis, and then along with all of our actual data points, we see the line of best fit plotted on the scatter plot.

Also in the console window, we see the estimates of our linear regression equation along with the model fit statistic, R-squared (R^2):

```
## Linear Fit
##  Intercept =  346.0325
##  Slope =  -0.12252
##  R-squared =  0.93554
```

So the intercept, or value of y when x is 0, is 346. And the slope, or change in y for every one unit increase in x, is negative 0.12. So we can see that the world-record time for the 800 meters decreases by about 0.12 seconds per year. We can also see that our model fits our data fairly well. Almost 94% of the variation in y can be explained by the variation in x.