Getting Started in R

Aug. 2019 STAT 6000

What is R?

For now, we will use R as a fancy calculator. There are a lot of formulas in this class that would be tedious to do by hand or with a simple calculator. For instance, calculating the sample mean of a list of 100 numbers would take a while to do by hand, but seconds in R. Another convenient feature of R is its plotting capabilities. Using R one can produce clean looking plots very efficiently. Later, we will learn how to use various built in functions (such as *lm*) in R to perform data analysis.

Installation

- Installing R:
  1. Go to r-project.org.
  2. On the left side of the page, click the "CRAN" link.
  3. Scroll down to the USA section, and click a link (say, Berkeley).
  4. Click the download link for your operating system (Windows, Mac, or Linux).
  5. Follow the instructions to download the latest version of R.
- Installing RStudio (An integrated development environment for R): Go to www.rstudio.com and follow the instructions.

Using R

Assignment and Arithmetic

Assignment of variables means that in R you can create your own variables, vectors, matrices, etc. R can perform basic arithmetic operations with numbers or declared variables. Here are some examples:

```
#define variables x and y
> x = 4
> y = 3

#perform arithmetic on variables as well as numbers
```

```
> x * y - 4

#output
[1] 8
```

R can perform arithmetic on numbers, but it is really designed to work with ***vectors***. Vectors are just lists of numbers, and in R we can define and perform operations on them. To create a vector, we use the concatenate function c():

```
#use concatenate function to make a vector called 'myvec'
> myvec = c(4,5,6)

#display myvec
> myvec
[1] 4 5 6
```

Once we have a vector, we can perform arithmetic on the entire vector at once. We can also add two vectors together.

```
#perform arithmetic on entire vectors at once
> myvec + 3
[1] 7 8 9
> myvec * 2
[1]  8 10 12

#create another vector called 'myvec2'
> myvec2 = c(1,2,1)

#add my vectors together
> myvec + myvec2
[1] 5 7 7
```

There are built in functions that we can perform on a vector as well.

```
#sum the elements of a vector
> sum(myvec)
[1] 15

#take the product of all elements of a vector
> prod(myvec)
[1] 120
```

```
#calculate the sample mean of a vector
> mean(myvec)
[1] 5

#calculate the median of a vector
> median(myvec)
[1] 5

#calculate the sample standard deviation of a vector
> sd(myvec)
[1] 1
```

As an example, consider a simple regression case.

```
#define vectors X and Y
> X = c(1.86,.22,3.55,3.29,1.25)
> Y = c(3.34,1.79,5.66,5.83,4.74)

#Xbar is the sample mean of X
> Xbar = mean(X)
> Xbar
[1] 2.034

#similarly for Y
> Ybar = mean(Y)
> Ybar
[1] 4.272

#more advanced calculations (^ means exponent)
> sum((X - Xbar)^2)
[1] 7.81132
> sum((X - Xbar)*(Y - Ybar))
[1] 8.35866
```

Here we have everything calculated, so we can easily obtain

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(X_i - \overline{X})^2} = \frac{8.35866}{7.81132} = 1.07,$$
$$\hat{\beta}_0 = \overline{Y} - \hat{\beta}_1\overline{X} = 4.272 - 1.07(2.034) = 2.09.$$

```
> beta1 = 8.35866 / 7.81132
> beta0 = Ybar - beta1*Xbar
```

We can now use this to find the fitted values $\hat{Y}_i$ and residuals e_i:

```
### fitted values
> Yhat = beta0 + beta1*X
> Yhat
[1] 4.085808 2.330893 5.894226 5.616008 3.433065

## residuals
> e=Y-Yhat
> e
[1] -0.7458078 -0.5408928 -0.2342263  0.2139920  1.3069350


### check properties of residuals:
> sum(e)
[1] 3.108624e-15
> sum(X*e)
[1] 5.925815e-15

> sum(Yhat*e)
[1] 1.24345e-14
```

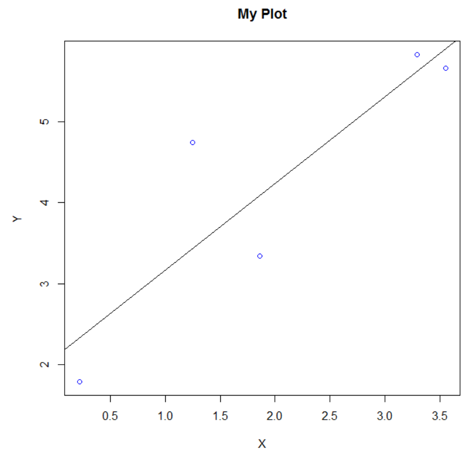
Plotting

Now that we have our data, and the fitted regression line, we would like to plot them both to see what we have done. The plot function in R is fairly simple. There are several options one can use when plotting, but we start with the basic ones:

```
#main is the title of the plot, col is the color of the points
> plot(X,Y,main='My Plot',col='blue')

#use abline to meake a line
> abline(beta0,beta1)
```

Here, the function abline() creates a line with a given slope and intercept. Here is the output for this simple example:



Example: Inference

Suppose a sequence of observations $x_1, \dots, x_{50}$ are drawn from a normal distribution $N(\mu, \sigma^2)$. We'd like to create a 90% confidence interval for the mean parameter μ. We use the formula given in our preliminary statistics courses:

$$\bar{x} \pm z_{(1-\alpha/2)} \frac{s}{\sqrt{n}}.$$

Since $\alpha = 1 - .9 = .1$ and $n = 50$, here

$$\bar{x} \pm z_{(.95)} \frac{s}{\sqrt{50}}.$$

What we need to do now is to calculate $\bar{x}$, s and the appropriate z-quantile. This can be done in R:

```
#define vector x

#Compute xbar: the sample mean of x
xbar = mean(x)
xbar
[1] 0.6345216

#Compute s: the sample standard deviation of x
s = sd(x)
s
```

```
[1] 2.131412

#Compute the confidence interval
#lower bound
xbar - qnorm(.95)*s/sqrt(50)
[1] 0.138718

#upper bound
xbar + qnorm(.95)*s/sqrt(50)
[1] 1.130325
```

This gives our 90% confidence interval: (.139,1.130).

Interpretation of 90% confidence interval: If repeated samples are taken from the same population and the 90% confidence interval are computed for each sample, 90% of the intervals would contain the true population parameter.

Note that the function qnorm() gives the percentiles (a.k.a. quantiles) of the standard normal distribution if you provide the percentile you want. Sometimes we would like to find the percentiles for a t distribution or a Chisquare distribution or an F distribution, these are by the functions qt(), qchisq(), qf(), respectively.

```
##95% percentile of Chisquare distribution with 3 degrees of freedom
> qchisq(0.95, 3)
[1] 7.814728

### 95% percentile of t-distribution with 3 degrees of freedom
> qt(0.95, 3)
[1] 2.353363

## 95% percentile of F-distribution with (1,3) degrees of freedom
> qf(0.95, 1,3)
[1] 10.12796
```

We would also like to test whether the mean parameter μ is 0. We'd like to test

$$H_0 : \mu = 0 \text{ vs } H_1 : \mu \neq 0.$$

Z-statistic:

$$Z = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} = \frac{\bar{x}}{s/\sqrt{n}}.$$

Given the significance level α, we reject H_0 if $|Z| > z_{(1-\alpha/2)}$, or equivalently, if pvalue:$= P(|z| > |Z|) < \alpha$ (z is a standard normal random variable).
Take $\alpha = .05$, here, we reject H_0 if $|Z| > z_{(.975)}$, or equivalently, if pvalue:$= P(|z| > |Z|) < .05$.
In R, we can do the hypothesis testing efficiently:

```
#Compute Z statistic
Z = xbar/(s/sqrt(50))
Z
[1] 2.105058

#Compute the critical value: 97.5% percentile of standard normal distribution
c_val = qnorm(.975)
c_val
[1] 1.959964

#Compute the p-value
p_val = 2*(1-pnorm(Z))
p_val
[1] 0.0352863
```

Exercises: Calculate the 0.975 percentile for $N(0,1)$ – standard normal, $t_{(10)}$ – t-distribution with 10 degrees of freedom, $\chi^2_{(10)}$ – Chisquare distribution with 10 degrees of freedom, $F_{1,10}$ – F-distribution with $(1,10)$ degrees of freedom.