Live Session Assignment 04

Samira Zarandioon

2/11/2017

1. Basic Math

1. The log of a positive number (12):

   ```
   log(12)
   ```

   ```
   ## [1] 2.484907
   ```

2. The default base is Euler's number (e = 2.71828). To calculate the log of 12 with 2 as base, we have the following two options:

   1. Use the **base** parameter:

      ```
      log(12, base=2)
      ```

      ```
      ## [1] 3.584963
      ```

   2. Use math formula ($log_b^a/log_b^c = log_c^a$):

      ```
      log(12) / log(2)
      ```

      ```
      ## [1] 3.584963
      ```

3. Log of negative number (i.e. -12) is not defined, therefore it produces NaN (Not a Number) value and outputs a warning message:

   ```
   log(-12)
   ```

   ```
   ## Warning in log(-12): NaNs produced
   ```

   ```
   ## [1] NaN
   ```

4. Square root of a poitive number (16):

   ```
   sqrt(16)
   ```

   ```
   ## [1] 4
   ```

2. Random number generation.

1. 15 random samples drawn from normal distribuiotn with mean 0 and sd 1:

   ```
   v <- rnorm(15)
   mean(v)
   ```

   ```
   ## [1] -0.3970789
   ```

   ```
   sd(v)
   ```

   ```
   ## [1] 1.017002
   ```

2. 15 random samples drawn from normal distribuiotn with mean 10 and sd 2:

```r
v <- rnorm(15, mean=10, sd=2)
mean(v)
```

```
## [1] 9.872376
```

```r
sd(v)
```

```
## [1] 1.918909
```

3. They are not exactly the same because number of samples (15) is fairly small. The larger number of sample is, the closer the means and sd are to the normal distribution.

3.Vector Operation.

1. The weights
2. Their hights
3. In R:

   ```r
   weights <- c(60,72,57,90,95,72)
   heights <- c(1.80,1.85,1.72,1.90,1.74,1.91)
   weights
   ```

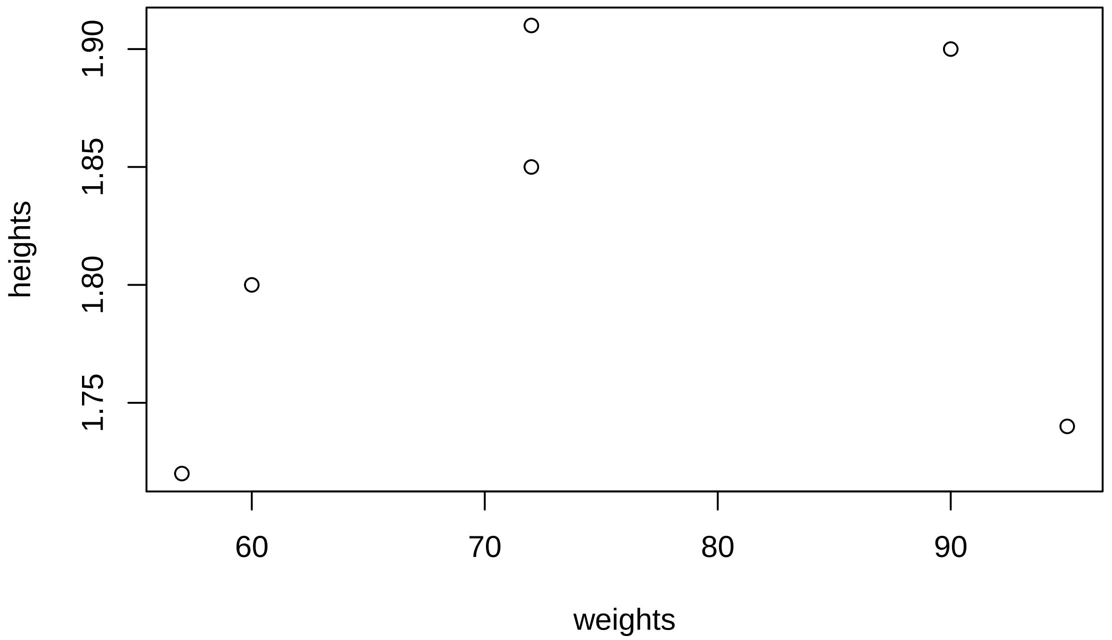
   ```
   ## [1] 60 72 57 90 95 72
   ```

   ```r
   heights
   ```

   ```
   ## [1] 1.80 1.85 1.72 1.90 1.74 1.91
   ```

4. Scatter Plot:

   ```r
   plot(weights,heights)
   ```



Aproximatly we can divide the data points into three categories.

5. BMI:

```
BMI <- weights/(heights*heights)
BMI
```

```
## [1] 18.51852 21.03725 19.26717 24.93075 31.37799 19.73630
```

6. Mean for Weights:

```
mean_weights <- mean(weights)
mean_weights
```

```
## [1] 74.33333
```

7. Deviation:

```
deviations <- weights - mean_weights
deviations
```

```
## [1] -14.333333  -2.333333 -17.333333  15.666667  20.666667  -2.333333
```

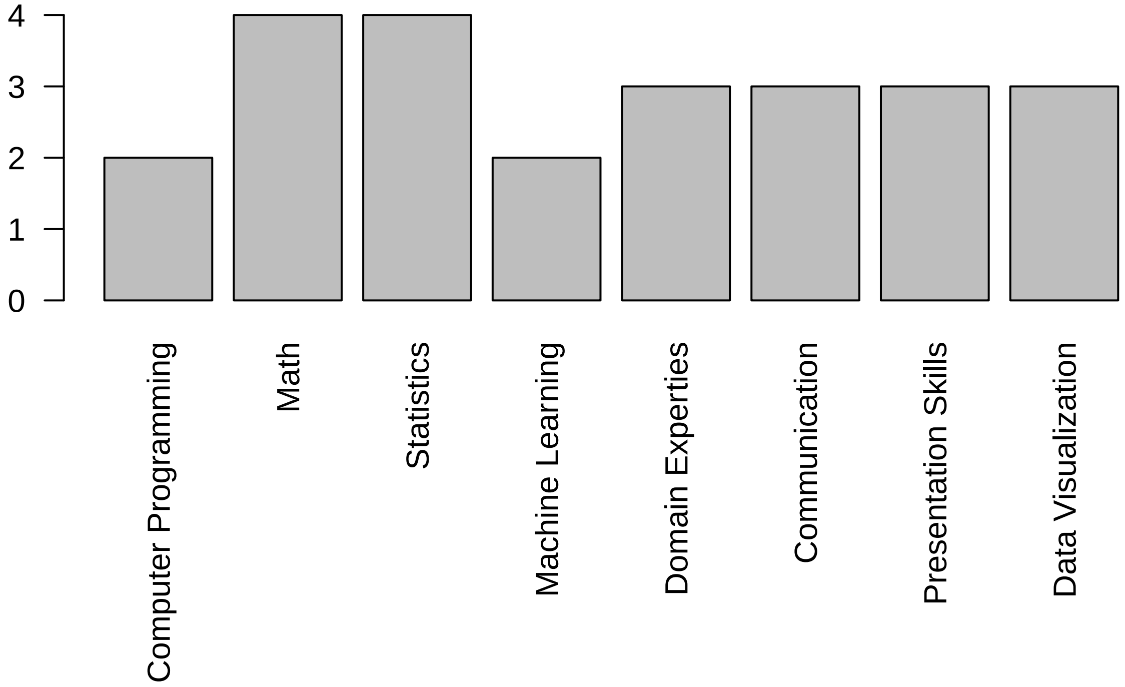
8. Sum of Deviations:

```
sum_deviations <- sum(deviations)
sum_deviations
```

```
## [1] 2.842171e-14
```

4. Data Science Profile

```
skills <- c("Computer Programming", "Math", "Statistics", "Machine Learning", "Domain Experties", "Commu
rank <- c(2, 4, 4, 2, 3, 3, 3, 3)
skills_rank <- data.frame(skills,rank)
par(mar=c(10,2,5,2)) # set margin so that labels fit, c(bottom, left, top, right)
par(las=2) # labels perpendicular to the axis
barplot(skills_rank$rank, names.arg=skills_rank$skills)
```


Computer Programming
Math
Statistics
Machine Learning
Domain Experties
Communication
Presentation Skills
Data Visualization