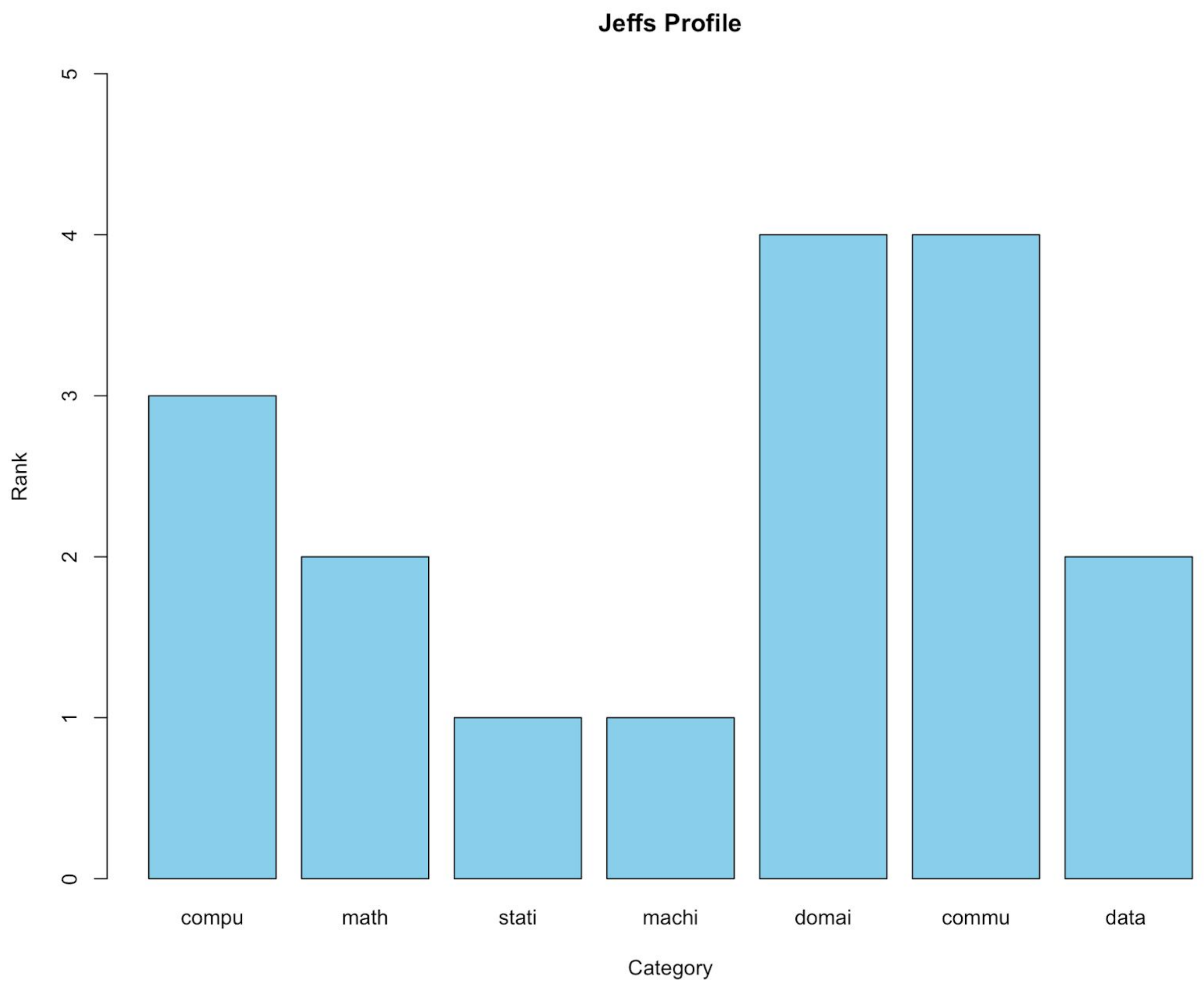
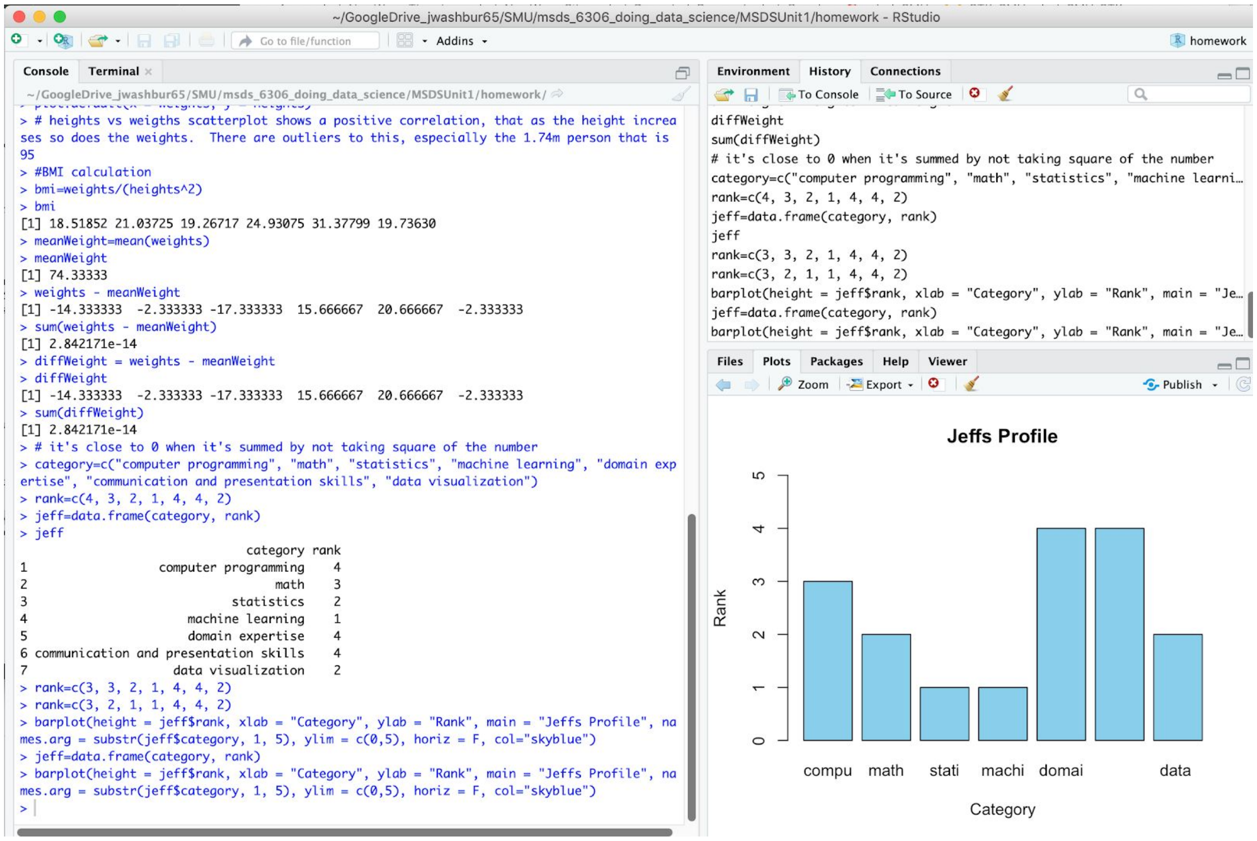
```
> ###################
> ## Basic Math
> #1
> log(18)
[1] 2.890372
> #2
> #Default base is e (2.71828)
> log(18,3)
[1] 2.63093
> #3
> log(-18)
[1] NaN
Warning message:
In log(-18) : NaNs produced
> #"e is the base rate of growth shared by all continually growing processes"
> #from https://betterexplained.com/articles/an-intuitive-guide-to-exponential-functions-e/
> #"Natural Log (ln) is the amount of time needed to reach a certain level of continuous growth"
> #from https://betterexplained.com/articles/demystifying-the-natural-logarithm-ln/
> # negative growth is impossible, you can't start with something and end with a negative
amount, the best you can do is start with something and end up nothing, 0
> #4
> sqrt(18)
[1] 4.242641

> ###################
> ## Random number generation
> #1
> myVect=rnorm(15)
> myVect
 [1]  0.6703234  0.2634273  0.8137644  0.6908441 -0.0561715 -2.2614468 -0.1140799
 [8]  0.2051591  0.7671953 -1.7469238  1.4157998 -0.8171081 -2.6838893  1.3551259
[15] -0.4875251
> mean(myVect)
[1] -0.132367
> sd(myVect)
[1] 1.258048
> #2
> myVect2=rnorm(15,mean = 10, sd = 2)
> myVect2
 [1]  9.701839 12.792797 11.363508 11.861090  9.840437 10.560990 11.120608  9.304913
 [9]  7.316593 10.607950  7.002428 11.197805 11.014657  6.317545 10.805918
> mean(myVect2)
[1] 10.05394
```

```
> sd(myVect2)
[1] 1.860814
> #3
> # The means and SD are not the same as specified because rnorm generates random
numbers that are close to the requested values.  Samples of data are taken that using only 15
numbers isn't a very large sample.  Using larger n would get closer to the desired mean and SD


> ###################
> # Vector operations
> weights=c(60,72,57,90,95,72)
> heights=c(1.80,1.85,1.72,1.90,1.74,1.91)
> plot.default(x = weights, y = heights)
> # heights vs weigths scatterplot shows a positive correlation, that as the height increases so
does the weights.  There are outliers to this, especially the 1.74m person that is 95
> #BMI calculation
> bmi=weights/(heights^2)
> bmi
[1] 18.51852 21.03725 19.26717 24.93075 31.37799 19.73630
> meanWeight=mean(weights)
> meanWeight
[1] 74.33333
> weights - meanWeight
[1] -14.333333  -2.333333 -17.333333  15.666667  20.666667  -2.333333
> sum(weights - meanWeight)
[1] 2.842171e-14
> diffWeight = weights - meanWeight
> diffWeight
[1] -14.333333  -2.333333 -17.333333  15.666667  20.666667  -2.333333
> sum(diffWeight)
[1] 2.842171e-14
> # it's close to 0 when it's summed by not taking square of the number
> category=c("computer programming", "math", "statistics", "machine learning", "domain
expertise", "communication and presentation skills", "data visualization")
> rank=c(4, 3, 2, 1, 4, 4, 2)
> jeff=data.frame(category, rank)
> jeff
                           category rank
1              computer programming    4
2                              math    3
3                        statistics    2
4                  machine learning    1
5                  domain expertise    4
```

```
6 communication and presentation skills    4
7                  data visualization    2
> rank=c(3, 3, 2, 1, 4, 4, 2)
> rank=c(3, 2, 1, 1, 4, 4, 2)
> barplot(height = jeff$rank, xlab = "Category", ylab = "Rank", main = "Jeffs Profile", names.arg
= substr(jeff$category, 1, 5), ylim = c(0,5), horiz = F, col="skyblue")
> jeff=data.frame(category, rank)
> barplot(height = jeff$rank, xlab = "Category", ylab = "Rank", main = "Jeffs Profile", names.arg
= substr(jeff$category, 1, 5), ylim = c(0,5), horiz = F, col="skyblue")
>
```





```
> # heights vs weigths scatterplot shows a positive correlation, that as the height increa
ses so does the weights.  There are outliers to this, especially the 1.74m person that is
95
> #BMI calculation
> bmi=weights/(heights^2)
> bmi
[1] 18.51852 21.03725 19.26717 24.93075 31.37799 19.73630
> meanWeight=mean(weights)
> meanWeight
[1] 74.33333
> weights - meanWeight
[1] -14.333333  -2.333333 -17.333333  15.666667  20.666667  -2.333333
> sum(weights - meanWeight)
[1] 2.842171e-14
> diffWeight = weights - meanWeight
> diffWeight
[1] -14.333333  -2.333333 -17.333333  15.666667  20.666667  -2.333333
> sum(diffWeight)
[1] 2.842171e-14
> # it's close to 0 when it's summed by not taking square of the number
> category=c("computer programming", "math", "statistics", "machine learning", "domain exp
ertise", "communication and presentation skills", "data visualization")
> rank=c(4, 3, 2, 1, 4, 4, 2)
> jeff=data.frame(category, rank)
> jeff
                               category rank
1                  computer programming    4
2                                  math    3
3                            statistics    2
4                      machine learning    1
5                      domain expertise    4
6 communication and presentation skills    4
7                    data visualization    2
> rank=c(3, 3, 2, 1, 4, 4, 2)
> rank=c(3, 2, 1, 1, 4, 4, 2)
> barplot(height = jeff$rank, xlab = "Category", ylab = "Rank", main = "Jeffs Profile", na
mes.arg = substr(jeff$category, 1, 5), ylim = c(0,5), horiz = F, col="skyblue")
> jeff=data.frame(category, rank)
> barplot(height = jeff$rank, xlab = "Category", ylab = "Rank", main = "Jeffs Profile", na
mes.arg = substr(jeff$category, 1, 5), ylim = c(0,5), horiz = F, col="skyblue")
>
```

```
diffWeight
sum(diffWeight)
# it's close to 0 when it's summed by not taking square of the number
category=c("computer programming", "math", "statistics", "machine learni…
rank=c(4, 3, 2, 1, 4, 4, 2)
jeff=data.frame(category, rank)
jeff
rank=c(3, 3, 2, 1, 4, 4, 2)
rank=c(3, 2, 1, 1, 4, 4, 2)
barplot(height = jeff$rank, xlab = "Category", ylab = "Rank", main = "Je…
jeff=data.frame(category, rank)
barplot(height = jeff$rank, xlab = "Category", ylab = "Rank", main = "Je…
```