

# STAT 33B Lec Workbook WK 8

Won Shil Park (3033452021)

Mar 7, 2021

This workbook is due **Mar 10, 2021 11:59pm**

Workbooks are graded for completeness, so as long as you make a clear effort to solve each problem, you'll get full credit. That said, make sure you understand the concepts here, because they're likely to reappear in homeworks, quizzes, and later lectures.

Last week, we wrote a function, called `traceMean()` that takes the mean of values above a certain cut-off. We added parameters, `threshold` and `dropNA` to the function in order to specify the threshold and whether or not to drop NAs in calculating the mean.

Below is the function that we wrote in lecture.

```
traceMean = function (x, traceAmt, dropNA)
{
  if (!is.numeric(x)) {
    return(NA)
  }
  mean(x[x > traceAmt], na.rm = dropNA)
}
```

```
#lecture notes
xtest = c(0, 5, 10, NA, 20, 30, NA, NA)
traceMean(xtest, 5, FALSE)
```

```
## [1] NA
```

```
traceMean(xtest, 5, TRUE)
```

```
## [1] 20
```

This week, we will modify the function in the following ways:

1. Provide a default value for `traceAmt` so that it doesn't need to be specified in the call. Copy the function above into the code chunk below and modify it.

```
traceMean = function (x, traceAmt = 0, dropNA)
{
  if (!is.numeric(x)) {
    return(NA)
  }
  mean(x[x > traceAmt], na.rm = dropNA)
}

traceMean
```

```
## function (x, traceAmt = 0, dropNA)
## {
##   if (!is.numeric(x)) {
##     return(NA)
##   }
##   mean(x[x > traceAmt], na.rm = dropNA)
## }
```

Test your function.

```
traceMean(xtest, dropNA = TRUE)
```

```
## [1] 16.25
```

```
traceMean(xtest, 5, dropNA = TRUE)
```

```
## [1] 20
```

```
# parameters doesn't have to be in order if you specify
traceMean(traceAmt = 5, xtest, dropNA = TRUE)
```

```
## [1] 20
```

2. Add a ... parameter to the function's signature and remove the dropNA parameter so that the user can simply pass the na.rm = TRUE or na.rm = FALSE in the function call. Copy the function from step 1 into the code chunk below and modify it.

```
traceMean = function (x, traceAmt = 0, ...)
{
  if (!is.numeric(x)) {
    return(NA)
  }
  mean(x[x > traceAmt], ...)
}
```

Test your function.

```
xtest = c(0, 5, 10, NA, 20, 30, NA, NA, 0, 5, 10, 0, 5, 10, 0, 5, 10, 1000, -2000)
```

```
traceMean(xtest, na.rm = TRUE, trim = 0.1)
```

```
## [1] 11.66667
```

```
# trim = takes vector 'xtest' and takes out 10% of each end since trim = 0.1
```

3. Allow the user to supply an alternative function to mean(). To do this, add a parameter called FUN and give it a default value of the mean function. Since the function has significantly changed, rename it to traceSummary() Copy the function from step 2 into the code chunk below and modify it.

```

traceSummary = function (x, traceAmt = 0, FUN = mean, ...)
{
  if (!is.numeric(x)) {
    return(NA)
  }
  FUN(x[x > traceAmt], ...)
}

```

Use the following code chunks to test your function.

- Use `traceSummary()` to compute the median of the values in `z` above 0 for and where NAs are removed. The answer should be 30.

```

z = c(-1, 0, NA, 0, 100, 50, 30, 10, 10, 17, 20, 110, 60, NA)

traceSummary(z, na.rm = TRUE)

```

```
## [1] 45.22222
```

```
traceSummary(z, 0, na.rm = TRUE, trim = 0.2)
```

```
## [1] 41
```

```
traceSummary(z, FUN = median, na.rm = TRUE)
```

```
## [1] 30
```

- Use `traceSummary()` to compute the 5% trimmed mean of the values in `z` that are greater than 10 and where NAs are removed. The answer should be about 55.3

```
traceSummary(z, 10, na.rm = TRUE, trim = 0.05)
```

```
## [1] 55.28571
```

- Use `traceSummary()` to compute quantile statistics of `z` values above 0. Pass into the function values for the `digits` and `na.rm` parameters.

```
traceSummary(z, FUN = quantile, na.rm = TRUE, digits = 2)
```

```
## 0% 25% 50% 75% 100%
## 10 17 30 60 110
```

```
traceSummary(z, FUN = quantile, na.rm = TRUE, probs = c(.3, .6))
```

```
## 30% 60%
## 18.2 46.0
```