

class06

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   :  2.00
##  1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##  Mean   :15.4    Mean   : 42.98
##  3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.   :25.0    Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
x <- 1:5
x
```

```
## [1] 1 2 3 4 5
```

```
x[x > 2] <- 100
```

```
#Example input vectors to start with
student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
mean
```

```
## function (x, ...)
## UseMethod("mean")
## <bytecode: 0x0000000016e24960>
## <environment: namespace:base>
```

```
#mean(student2, na.rm = TRUE)

student2[is.na(student2)] <- 0
mean(student2)
```

```
## [1] 79.625
```

```
#student2[which.min(student2)] <- NA  
#mean(student2, na.rm = TRUE)  
  
student2 <- student2[-which.min(student2)]  
mean(student2)
```

```
## [1] 91
```

```
#Experiment with student scores  
student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)  
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)  
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)  
  
#calculate mean (NA is not counted)  
mean(student1, na.rm = TRUE)
```

```
## [1] 98.75
```

```
mean(student2, na.rm = TRUE)
```

```
## [1] 91
```

```
mean(student3, na.rm = TRUE)
```

```
## [1] 90
```

```
#calculate mean (NA is zero grade)  
student1[is.na(student1)] <- 0  
student2[is.na(student2)] <- 0  
student3[is.na(student3)] <- 0  
  
mean(student1)
```

```
## [1] 98.75
```

```
mean(student2)
```

```
## [1] 79.625
```

```
mean(student3)
```

```
## [1] 11.25
```

```
#calculate mean with lowest score dropped  
student1[is.na(student1)] <- 0  
student2[is.na(student2)] <- 0  
student3[is.na(student3)] <- 0  
  
mean(student1[-which.min(student1)])
```

```
## [1] 100
```

```
mean(student2[-which.min(student2)])
```

```
## [1] 91
```

```
mean(student3[-which.min(student3)])
```

```
## [1] 12.85714
```

```
#Make general function for student scores
```

```
grade <- function(x) {  
  x[is.na(x)] <- 0  
  mean(x[-which.min(x)])  
}
```

```
#First, fix NA values
```

```
x[is.na(x)] <- 0
```

```
#Next remove lowest score and calculate avg
```

```
mean(x[-which.min(x)])
```

```
## [1] 75.5
```

```
#Import gradebook data
```

```
gradebook <- read.csv("https://tinyurl.com/gradeinput", row.names=1)  
head(gradebook)
```

```
##           hw1 hw2 hw3 hw4 hw5  
## student-1 100  73 100  88  79  
## student-2  85  64  78  89  78  
## student-3  83  69  77 100  77  
## student-4  88  NA  73 100  76  
## student-5  88 100  75  86  79  
## student-6  89  78 100  89  77
```

```
#Apply grade function over gradebook
```

```
student_scores <- apply(gradebook, 1, grade)  
student_scores
```

```
## student-1 student-2 student-3 student-4 student-5 student-6 student-7  
##      91.75      82.50      84.25      84.25      88.25      89.00      94.00  
## student-8 student-9 student-10 student-11 student-12 student-13 student-14  
##      93.75      87.75      79.00      86.00      91.75      92.25      87.75  
## student-15 student-16 student-17 student-18 student-19 student-20  
##      78.75      89.50      88.00      94.50      82.75      82.75
```

```
#For apply: apply(X, 1/2, FUN), where the 1 = apply to rows and 2 = apply to cols
```

```
#Top-scoring student  
which.max(student_scores)
```

```
## student-18  
##      18
```

```
#Lowest-scoring homework  
homework_scores <- apply(gradebook, 2, mean, na.rm = TRUE)  
homework_scores
```

```
##      hw1      hw2      hw3      hw4      hw5  
## 89.00000 80.88889 80.80000 89.63158 83.42105
```

```
which.min(homework_scores)
```

```
## hw3  
##    3
```

```
#Find which homework score was closest to student score  
gradebook2 <- gradebook  
gradebook2[is.na(gradebook2)] <- 0  
correlation <- apply(gradebook2, 2, cor, x = student_scores)  
which.max(correlation)
```

```
## hw5  
##    5
```