StemNav_R

2025-08-12

Yale STEM Navigators - Crash Course in R

Below are a couple practice problems to review the topics covered in today's slides. There are also a couple new tidbits of information and tips specific to RStudio!

Hint: to run code chunks below, you can either press the green triangle at the top-right of the chunk, use `Cmd + Enter` on Mac, or `Ctrl + Enter` on Windows.

The answer key can be found **here**!

Problem 1 (R Markdown in RStudio)

Below is a code chunk. This whole file is something called an "R-Markdown" file, which is a mix of different text formats alongside executable code blocks. The shortcut to insert a code block is `Cmd + Option + I` on Mac and `Ctrl + Alt + I` on Windows.

Highlight the italicized text below and input that shortcut to replace it with a code block. Add a comment inside that code block for good measure.

```
# This is a comment! It's good practice to include these in your code as a form of docum
entation for future reference or for people collaborating with you.
# That said, go ahead and replace the italicized text below with a code block
```

```
# Answer to Problem 1
```

Problem 2 (Creating Objects)

Insert another code block below, then create and display the following:

1. An object called *fifteen* which contains the number 15.
2. A vector of length 3 called *pi* containing the digits 3, 1, and 4.
3. A vector called *big_vec* containing all numbers from 1 to 100
    - **Hint:** a consecutive sequence of numbers can be input via `x:y` , where `x` is the first number in the sequence, and `y` is the last number (e.g. `1:5 = 1, 2, 3, 4, 5`)
4. A list called *stuff* containing the following elements:
    - The character string `"python could never"`
    - A vector of length 3 containing the logical values `TRUE, FALSE, TRUE`
    - The number `1.2345`
5. A 2x2 matrix called *nums* containing the values `1, 2, 3, 4`

```r
# 1
fifteen <- 15
fifteen
```

```
## [1] 15
```

```r
# 2
pi <- c(3, 1, 4)
pi
```

```
## [1] 3 1 4
```

```r
# 3
big_vec <- c(1:100)
big_vec
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100
```

```r
# 4
stuff <- list("python could never", c(TRUE, FALSE, TRUE), 1.2345)
stuff
```

```
## [[1]]
## [1] "python could never"
##
## [[2]]
## [1]  TRUE FALSE  TRUE
##
## [[3]]
## [1] 1.2345
```

```r
# 5
nums <- matrix(c(1, 2, 3, 4), nrow = 2)
nums
```

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

Problem 3 (Importing/Exploring a Data Frame)

Make a code block, and then create a data frame called *y_admits* using the CSV file linked here (https://raw.githubusercontent.com/SebastianReyes2005/StemNav_R/refs/heads/main/Yale_Admit_Rates.csv). Display the first 5 rows, and then find the summary statistics (min, Q1, median, Q3, max) of the *Matriculations* variable.

Tip: remember to include quotation marks around the link.

```
y_admits <- read.csv("https://raw.githubusercontent.com/SebastianReyes2005/StemNav_R/refs/heads/main/Yale_Admit_Rates.csv")

head(y_admits)
```

```
##   Application.Year Class.of Applications Admissions  Rate Matriculations
## 1             1976     1980         9387       2481 0.264           1300
## 2             1977     1981         9785       2423 0.248           1330
## 3             1978     1982        10015       2464 0.246           1372
## 4             1979     1983        10275       2204 0.215           1276
## 5             1980     1984        10304       2130 0.207           1257
## 6             1981     1985        10937       2186 0.200           1296
##   Percent.Matric
## 1      0.5239823
## 2      0.5489063
## 3      0.5568182
## 4      0.5789474
## 5      0.5901408
## 6      0.5928637
```

```
summary(y_admits)
```

```
##  Application.Year    Class.of     Applications     Admissions
##  Min.   :1976     Min.   :1980   Min.   : 9387   Min.   :1878
##  1st Qu.:1988     1st Qu.:1992   1st Qu.:11922   1st Qu.:2014
##  Median :2000     Median :2004   Median :13270   Median :2178
##  Mean   :2000     Mean   :2004   Mean   :20749   Mean   :2180
##  3rd Qu.:2012     3rd Qu.:2016   3rd Qu.:28974   3rd Qu.:2310
##  Max.   :2024     Max.   :2028   Max.   :57465   Max.   :2521
##       Rate         Matriculations Percent.Matric
##  Min.   :0.03734   Min.   :1255   Min.   :0.5240
##  1st Qu.:0.06910   1st Qu.:1299   1st Qu.:0.5690
##  Median :0.16100   Median :1321   Median :0.6359
##  Mean   :0.13998   Mean   :1362   Mean   :0.6297
##  3rd Qu.:0.19700   3rd Qu.:1364   3rd Qu.:0.6866
##  Max.   :0.26400   Max.   :1789   Max.   :0.8290
```

Problem 4 (Data Subsetting)

Create two new vectors called *yield_rate* and *year* containing the *Percent.Matric* and *Application.Year* columns of *y_admits*, respectively.

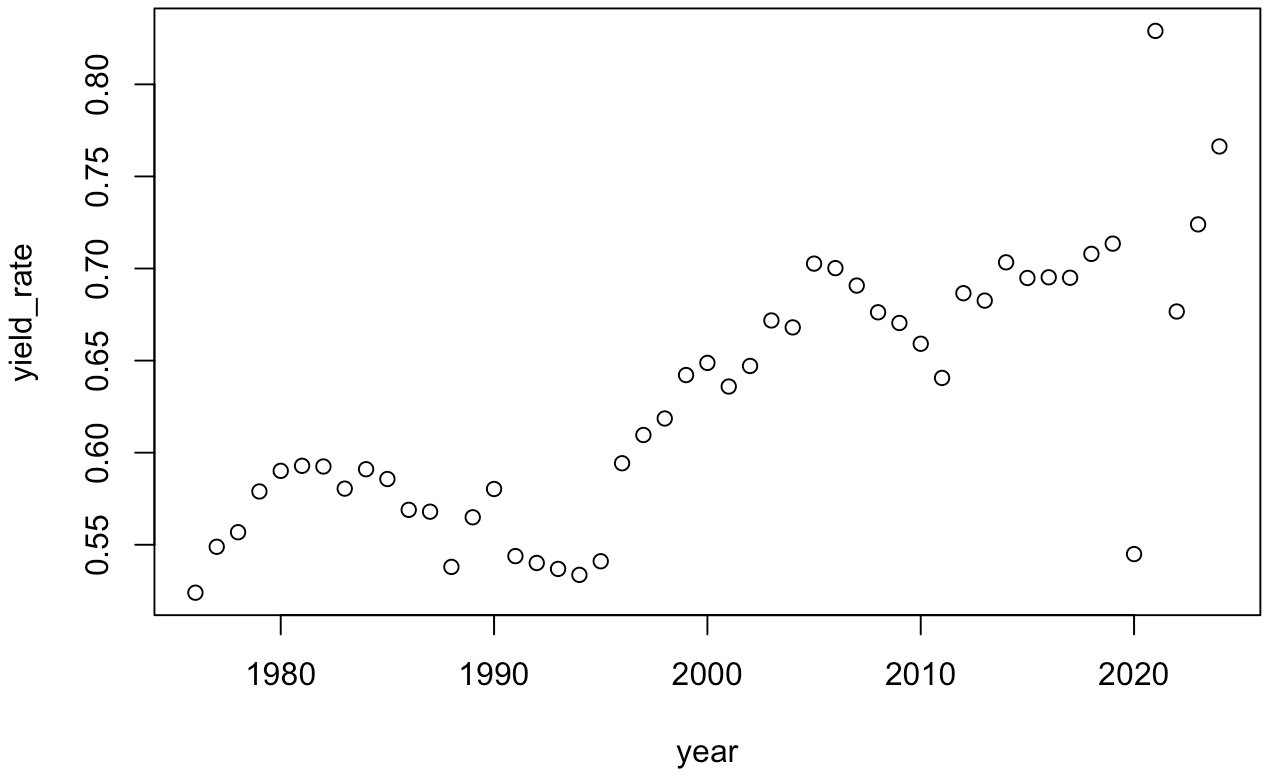
```
yield_rate <- y_admits$Percent.Matric

year <- y_admits$Application.Year
```

Problem 5 (Data Visualization)

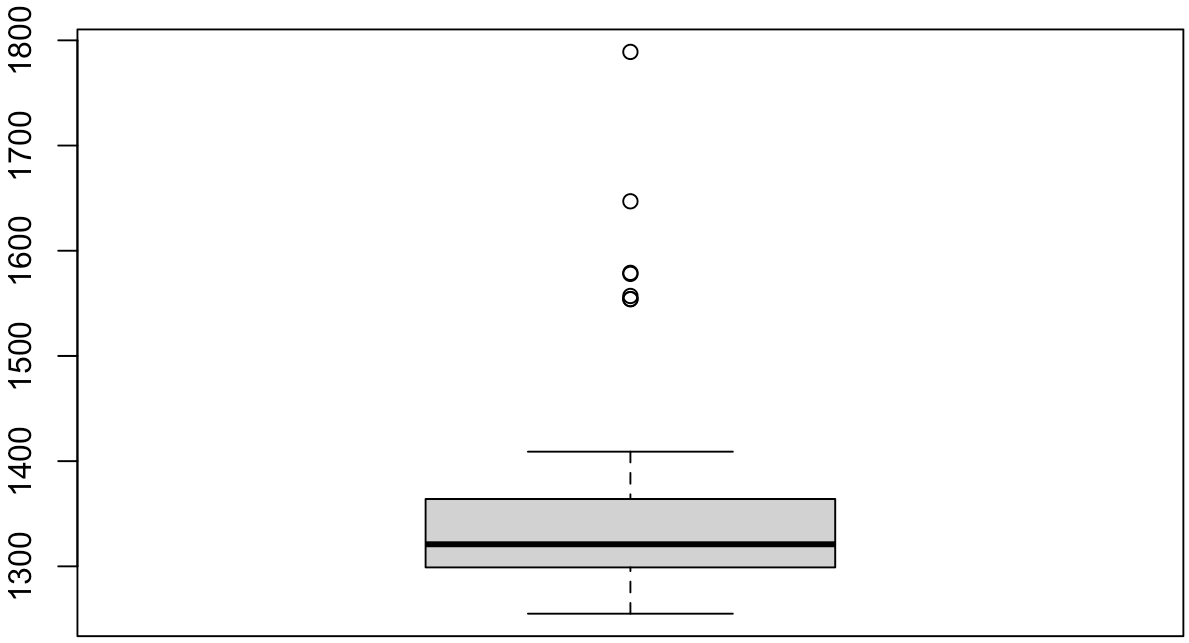
Create:

- A scatterplot with *year* on the x-axis and *yield_rate* on the y-axis
- A box plot using the *Matriculations* column of *y_admits* (does this match the summary statistics you found in problem 3?)

```
plot(year, yield_rate)
```




```
boxplot(y_admits$Matriculations)
```



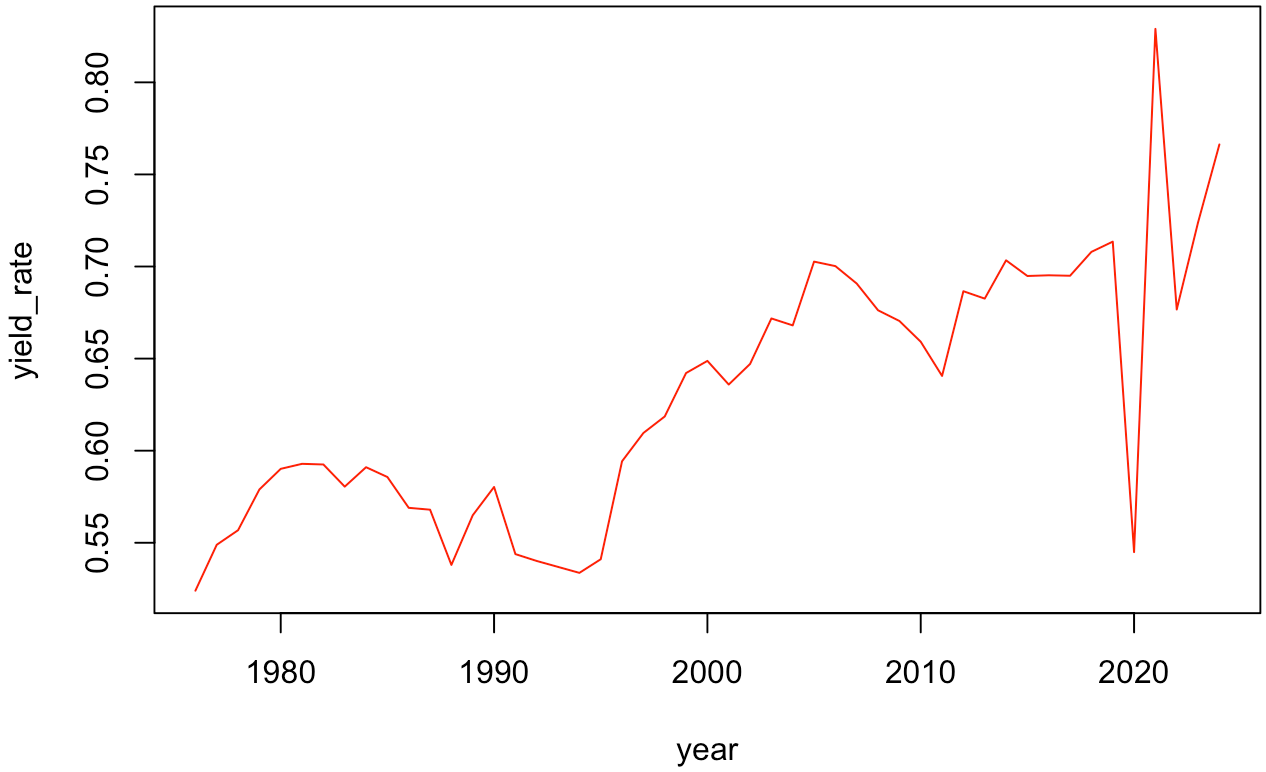
Problem 6 (CHALLENGE)

Run the below code block. Note that it pulls up a documentation page for the `plot()` function! This can also be done for other functions (e.g. `?boxplot`). Go through the documentation for `plot()`, and use the information to re-create the scatterplot from question 5 using a line. Additionally, make the line red.

Hint: for color, try setting the appropriate parameter as `___ = "red"`.

```
?plot.default
```

```
plot(year, yield_rate, type = "l", col = "red")
```


yield_rate
0.55
0.60
0.65
0.70
0.75
0.80
1980
1990
2000
2010
2020
year