Chapter 2_Solutions

Jo Cao

1/23/2022

This is the first R markdown I created to record my solutions to the data exercises in Chapter 2.

1.Create a character vector called my_names that contains all your first, middle and last names as elements. Calculate the length of my_names.

```
my_names <- c("tianyu","joanne","cao")
length(my_names)
```

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

The length of my_names is 3.

2.Create a second numeric vector called which which corresponds to my_names. The entries should be the position of each name in the order of your full name. Verify that it has the same length as my_names.

```
which <- c(1,2,3)
length(which)
```

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

Which has the same length of 3 as my_names.

3.Create a dataframe called names, which consists of the two vectors my_names and which as columns. Calculate the dimensions of names.

```
names <- data.frame(my_names,which)
dim(names)
```

```
## [1] 3 2
```

The data frame has a dimension of 3, 2.

4.Create a new dataframe new_names with the which column converted to character type. Verify that your command worked using str().

```
new_names <-data.frame(my_names,as.character(which))
str(new_names)
```

```
## 'data.frame':    3 obs. of  2 variables:
##  $ my_names           : chr  "tianyu" "joanne" "cao"
##  $ as.character.which.: chr  "1" "2" "3"
```

5. Load the ugtests data set via the peopleanalyticsdata package or download it from the internet. Calculate the dimensions of ugtests and view the first three rows only.

```
library(peopleanalyticsdata)
dim(ugtests)
```

```
## [1] 975   4
```

Here is a preview of the first three rows.

```
head(ugtests, 3)
```

```
##   Yr1 Yr2 Yr3 Final
## 1  27  50  52    93
## 2  70 104 126   207
## 3  27  36 148   175
```

6. View a statistical summary of all of the columns of ugtests. Determine if there are any missing values.

Here is a statistical summary.

```
summary(ugtests)
```

```
##       Yr1             Yr2             Yr3            Final      
##  Min.   : 3.00   Min.   :  6.0   Min.   :  8.0   Min.   :  8  
##  1st Qu.:42.00   1st Qu.: 73.0   1st Qu.: 81.0   1st Qu.:118  
##  Median :53.00   Median : 94.0   Median :105.0   Median :147  
##  Mean   :52.15   Mean   : 92.4   Mean   :105.1   Mean   :149  
##  3rd Qu.:62.00   3rd Qu.:112.0   3rd Qu.:130.0   3rd Qu.:175  
##  Max.   :99.00   Max.   :188.0   Max.   :198.0   Max.   :295
```

There is 0 missing value.

```
sum(is.na(ugtests))
```

```
## [1] 0
```

7.View the subset of ugtests for values of Yr1 greater than 50.

```
head(subset(ugtests, subset = Yr1 > 50))
```

```
##    Yr1 Yr2 Yr3 Final
## 2   70 104 126   207
## 6   86 122 119   159
## 8   60  92  78    84
## 10  80 127  67    80
## 13  64 123 110   175
## 14  62  84 142   182
```

8.Install and load the package dplyr. Look up the help for the filter() function in this package and try to use it to repeat the task in the previous question.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
head(dplyr::filter(ugtests, Yr1 > 50))
```

```
##   Yr1 Yr2 Yr3 Final
## 1  70 104 126   207
## 2  86 122 119   159
## 3  60  92  78    84
## 4  80 127  67    80
## 5  64 123 110   175
## 6  62  84 142   182
```

9. Write code to find the mean of the Yr1 test scores for all those who achieved Yr3 test scores greater than 100. Round this mean to the nearest integer.

Load magrittr library to get the pipe operator.

```
library(magrittr)
```

And then find the mean score.

```
subset(ugtests$Yr1, subset = ugtests$Yr3 > 100) %>%
  mean() %>%
  round()
```

```
## [1] 52
```

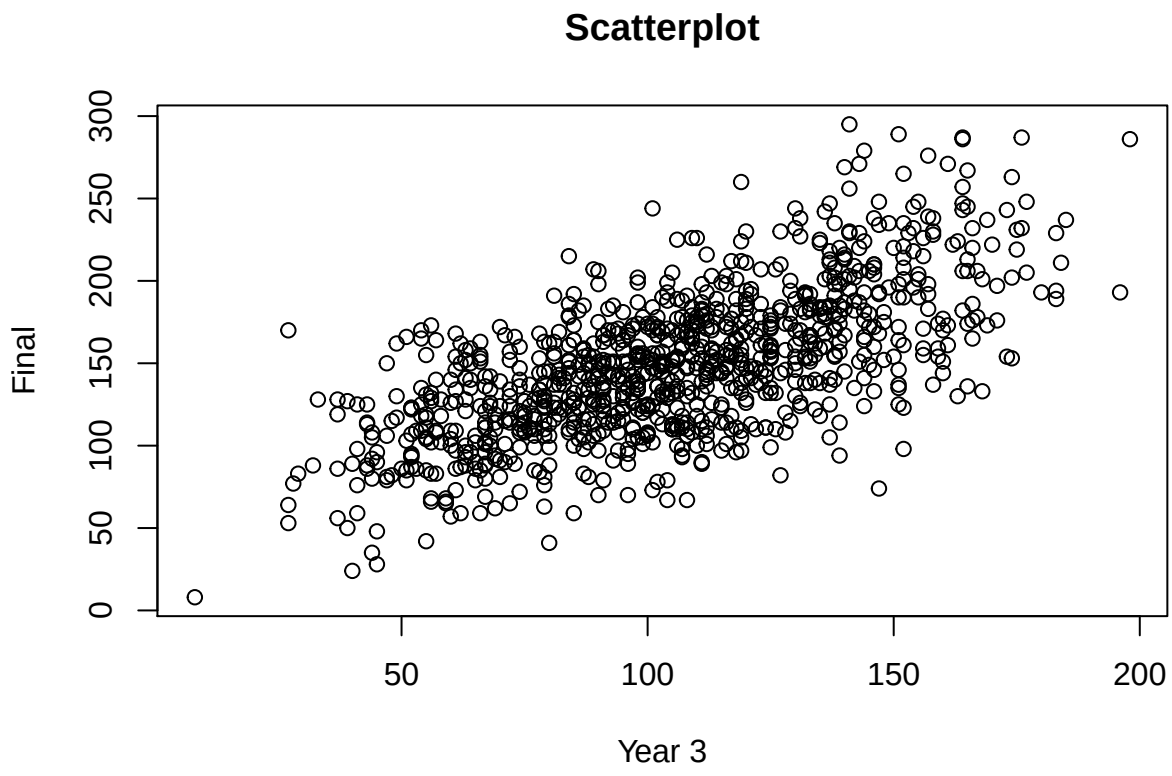
10. Familiarize yourself with the two functions filter() and pull() from dplyr. Use these functions to try to do the same calculation in the previous question using a single unbroken piped command. Be sure to namespace where necessary.

```
ugtests %>%
  dplyr::filter(Yr3 > 100) %>%
  dplyr::pull(Yr1) %>%
  mean() %>%
  round()
```

```
## [1] 52
```

11.Create a scatter plot using the ugtests data with Final scores on the y axis and Yr3 scores on the x axis.

```
plot(x = ugtests$Yr3, y = ugtests$Final, xlab = "Year 3", ylab = "Final", main = "Scatterplot")
```



12.Create your own 5-level grading logic and use it to create a new finalgrade column in the ugtests data set with grades 1–5 of increasing attainment based on the Final score in ugtests. Generate a histogram of this finalgrade column.

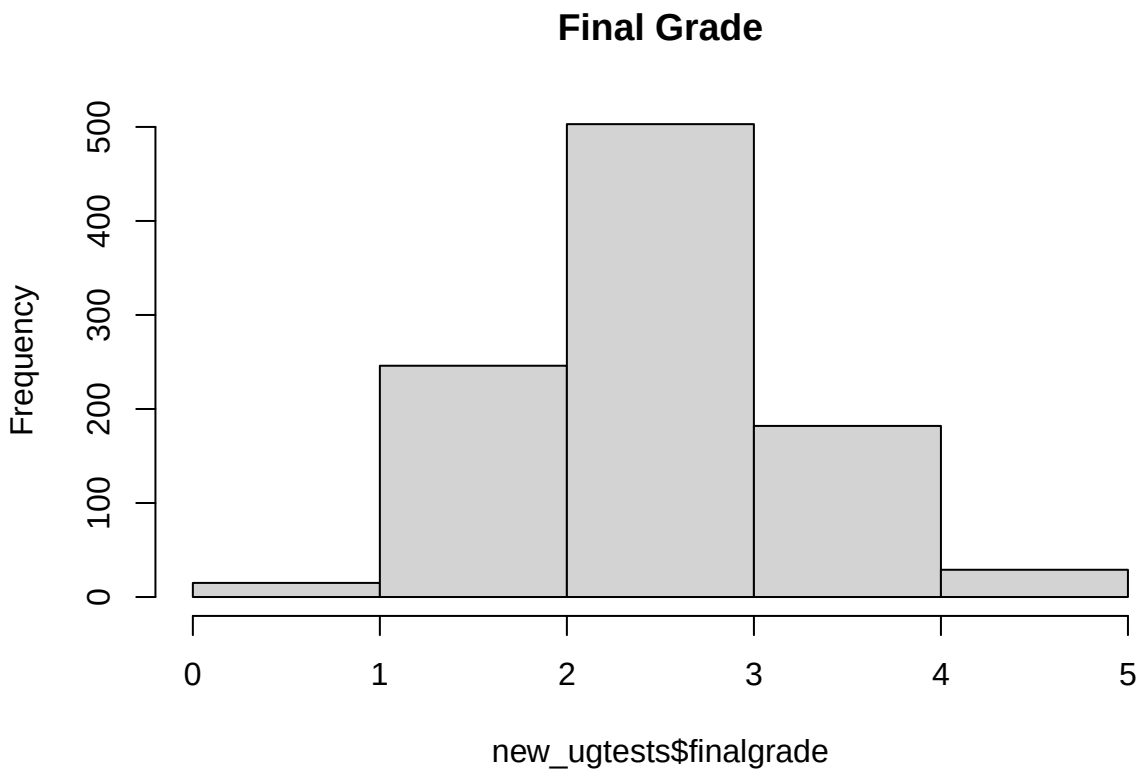
Add a new finalgrade column in the ugtests data to create a new data frame new_ugtests.

```
new_ugtests <- ugtests %>%
  mutate(finalgrade = case_when (ugtests$Final <= 60 ~ 1,
                                 between(ugtests$Final,60,120) ~2,
                                 between(ugtests$Final, 120,180) ~3,
                                 between(ugtests$Final, 180,240) ~4,
                                 between(ugtests$Final, 240,300) ~5))
str(new_ugtests)
```

```
## 'data.frame':    975 obs. of  5 variables:
##  $ Yr1       : int  27 70 27 26 46 86 40 60 49 80 ...
##  $ Yr2       : int  50 104 36 75 77 122 100 92 98 127 ...
##  $ Yr3       : int  52 126 148 115 75 119 125 78 119 67 ...
##  $ Final     : int  93 207 175 125 114 159 153 84 147 80 ...
##  $ finalgrade: num  2 4 3 3 2 3 3 2 3 2 ...
```

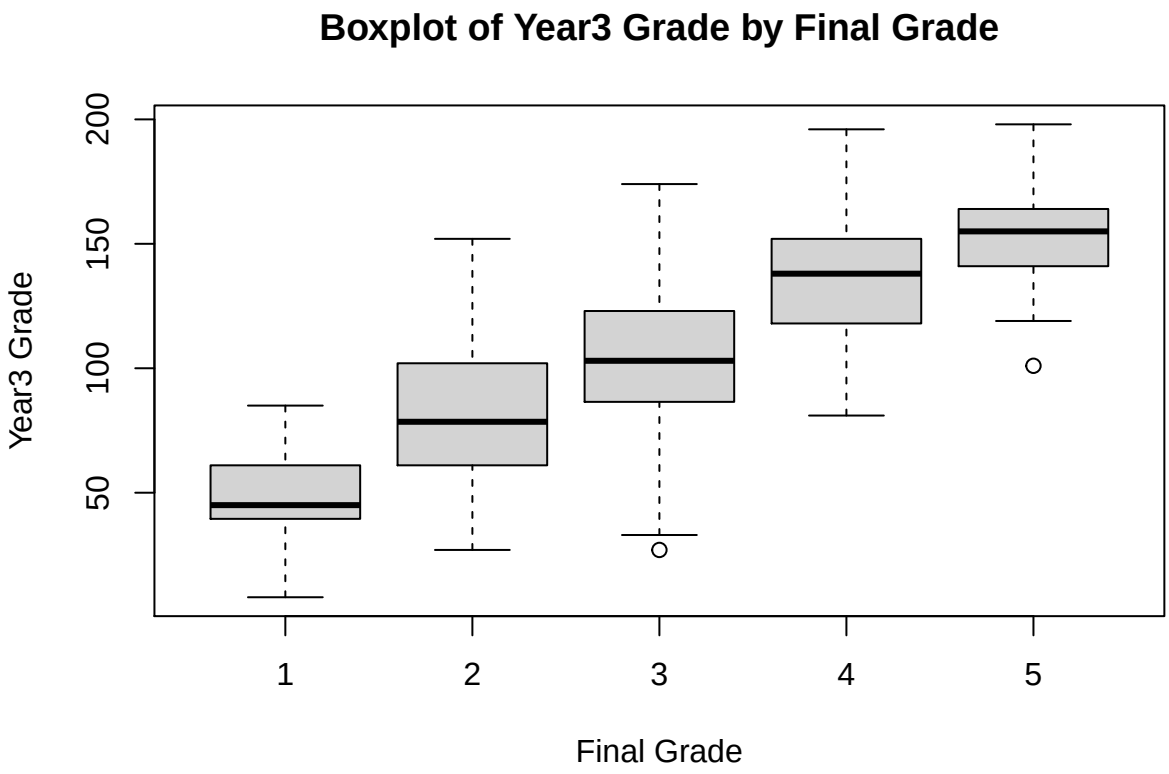
And then create a histogram

```
hist(new_ugtests$finalgrade, breaks=0:5, main="Final Grade")
```



13. Using your new ugtests data with the extra column from the previous exercise, create a box plot of Yr3 scores grouped by finalgrade.

```
boxplot(formula = Yr3 ~ finalgrade, data = new_ugtests, xlab="Final Grade", ylab="Year3 Grade", main =
```


Boxplot of Year3 Grade by Final Grade
Year3 Grade
50
100
150
200
1
2
3
4
5
Final Grade