Lab2

Lin Pin Tzu (Ruby)

2022-06-30

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
V <- c("Bears", "Lions", "Dolphins", "Eagles", "Bengals")
V
```

```
## [1] "Bears"    "Lions"    "Dolphins" "Eagles"   "Bengals"
```

```
str(V)
```

```
##  chr [1:5] "Bears" "Lions" "Dolphins" "Eagles" "Bengals"
```

1) Why is the vector shown above an atomic vector? (Explain using two or three sentences)

Atomic vector show logical, integer , numeric , complex , character, or raw and can have any attributes except a dimension attribute, and atomic vector only shows the same type of object.

2) Use and show R code that will extract “Dolphins” from the vector shown above.

```
V[3]
```

```
## [1] "Dolphins"
```

3) Use and show Rcode that will extract “Bears” , “Dolphins” and “Bengals” from the vector shown above.

```
V[c(1,3,5)]
```

```
## [1] "Bears"    "Dolphins" "Bengals"
```

4) Use and show two Rcoding methods that will show all objects of the vector given above except "Bears".

```
V[-1]
```

```
## [1] "Lions"    "Dolphins" "Eagles"   "Bengals"
```

```
V[2:5]
```

```
## [1] "Lions"    "Dolphins" "Eagles"   "Bengals"
```

K <- list(x = 3:7, "never", 43, y = list(10,20,30))

5) Why is the vector given above called a list? (Explain in two or three sentences) If the vector is a list, identify the type of each object in the list.

List shows logical, integer , numeric , complex , character, or raw and can have any attributes except a dimension attribute but list can show different types of objects.

```
K <- list( x = 3:7, "never", 43, y = list(10,20,30))
str(K)
```

```
## List of 4
##  $ x: int [1:5] 3 4 5 6 7
##  $  : chr "never"
##  $  : num 43
##  $ y:List of 3
##   ..$ : num 10
##   ..$ : num 20
##   ..$ : num 30
```

6) Use and show R code that will give the length of the vector shown above.

```
length(K)
```

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

7) Use and show R code that will output the fourth object in the vector shown above.

```
K[4]
```

```
## $y
## $y[[1]]
## [1] 10
##
## $y[[2]]
## [1] 20
##
```

```
## $y[[3]]
## [1] 30
```

8) Use and show R code that will show all objects in the vector (list) given above.

```
K[]
```

```
## $x
## [1] 3 4 5 6 7
##
## [[2]]
## [1] "never"
##
## [[3]]
## [1] 43
##
## $y
## $y[[1]]
## [1] 10
##
## $y[[2]]
## [1] 20
##
## $y[[3]]
## [1] 30
```

```
str(K)
```

```
## List of 4
##  $ x: int [1:5] 3 4 5 6 7
##  $  : chr "never"
##  $  : num 43
##  $ y:List of 3
##   ..$ : num 10
##   ..$ : num 20
##   ..$ : num 30
```

9) Copy paste and run the tribble given below.

```
tribble( ~x,    ~y,    ~w,    ~z,
              210,   300,   220,   180,
              102,   100,   119,   187,
              176,   175,   188,   173,
              87,    95,   91,     94,
              202,   210,  234,    218,
              110,   122,  131,    128,
) -> dt

dt
```

```
## # A tibble: 6 x 4
##       x     y     w     z
```

```
##   <dbl> <dbl> <dbl> <dbl>
## 1   210   300   220   180
## 2   102   100   119   187
## 3   176   175   188   173
## 4    87    95    91    94
## 5   202   210   234   218
## 6   110   122   131   128
```

9a) Use and show a map function to find the mean of each column of the dt data table

```
map_dbl(dt,mean)
```

```
##        x        y        w        z
## 147.8333 167.0000 163.8333 163.3333
```

9b) Use and show a map function to find the standard deviation of each column of the dt data table.

```
map_dbl(dt,sd)
```

```
##        x        y        w        z
## 54.45151 79.12016 58.40348 44.66617
```

9c) Use and show a map function that will calculate the square root of each value of each column of the data table dt.

```
dt %>%
  map(~.^(.5)) -> d
d
```

```
## $x
## [1] 14.491377 10.099505 13.266499  9.327379 14.212670 10.488088
##
## $y
## [1] 17.320508 10.000000 13.228757  9.746794 14.491377 11.045361
##
## $w
## [1] 14.832397 10.908712 13.711309  9.539392 15.297059 11.445523
##
## $z
## [1] 13.41641 13.67479 13.15295  9.69536 14.76482 11.31371
```

```
sqrt(dt)
```

```
## # A tibble: 6 x 4
##       x     y     w     z
##   <dbl> <dbl> <dbl> <dbl>
## 1 14.5  17.3  14.8  13.4
## 2 10.1  10    10.9  13.7
## 3 13.3  13.2  13.7  13.2
```

```
## 4  9.33  9.75  9.54  9.70
## 5 14.2  14.5  15.3  14.8
## 6 10.5  11.0  11.4  11.3
```

9d) Use R code to find the mean, max, 1st Quartile, 2nd Quartile,Median, and Mean for each column of the dt data table. (Hint: You do not have to use a map function)

```
summary(dt)
```

```
##        x               y               w               z
##  Min.   : 87.0   Min.   : 95.0   Min.   : 91.0   Min.   : 94.0
##  1st Qu.:104.0   1st Qu.:105.5   1st Qu.:122.0   1st Qu.:139.2
##  Median :143.0   Median :148.5   Median :159.5   Median :176.5
##  Mean   :147.8   Mean   :167.0   Mean   :163.8   Mean   :163.3
##  3rd Qu.:195.5   3rd Qu.:201.2   3rd Qu.:212.0   3rd Qu.:185.2
##  Max.   :210.0   Max.   :300.0   Max.   :234.0   Max.   :218.0
```