HW_03

izd3

Use only commands & functions that are shown in the indicated chapter or prior chapters.

Problem #01 - Chapter 10 Exercise #02

```
# Show your work here

CharacterData001[10:15]
```

```
## [1] "freedom"   "and"       "security"  "go"        "together," "that"
```

```
CharacterData006[(length(CharacterData006)-5):64]
```

```
## [1] "as"   "gave" "the"  "to"   "that" "Good"
```

```
Filtering001[1:20]
```

```
##  [1]  0.003459579  2.584714042  9.146763128 -4.697780274  1.602831199
##  [6] -2.564479490 -3.511857823  9.424604238  8.287863084  3.901561676
## [11]  3.910133997  6.609050031 -6.447932529 -4.760478870  5.241202158
## [16]  3.315047692  3.986875410 -3.852335322 -8.803384653 -8.537535728
```

Problem #02 - Chapter 10 Exercise #04

```
# Show your work here
fill1<-which(Filtering001>=9.9)
Filtering001[fill1]
```

```
## [1] 9.911178 9.959709 9.954380 9.921894 9.986333 9.951565 9.984792
```

```
Filtering001[Filtering001>=0 & Filtering001<=0.025]
```

```
## [1] 0.003459579
```

```
CharacterData001[CharacterData001=='if' | CharacterData001=='this']
```

```
## [1] "if"   "if"   "if"   "this" "this" "this"
```

Problem #03 - Chapter 11 Exercise #02

```
# Show your work here

fruitMatrix[,5]
```

```
## [1] "date"        "dragonfruit" "durian"      "eggplant"    "elderberry" 
## [6] "feijoa"
```

Problem #04 - Chapter 11 Exercise #05

```
# Show your work here
str(fruitMatrix)
```

```
##  chr [1:6, 1:5] "apple" "apricot" "avocado" "banana" "bell pepper" ...
```

```
matrix(fruitMatrix[2:5,2:4],nrow = 4,ncol = 3,byrow = FALSE)
```

```
##      [,1]           [,2]           [,3]
## [1,] "blackcurrant" "cantaloupe"   "coconut"
## [2,] "blood orange" "cherimoya"    "cranberry"
## [3,] "blueberry"    "cherry"       "cucumber"
## [4,] "boysenberry"  "chili pepper" "currant"
```

Problem #05 - Chapter 11 Exercise #08

```
# Show your work here
matrix(numberMatrix[numberMatrix<=25],nrow = 4,ncol = 4,byrow = FALSE)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    8   20   16    2
## [2,]   18   10    6   15
## [3,]    5    4    7   11
## [4,]   25    9   12   19
```

Problem #06 - Chapter 12 Exercise #04

```r
# Show your work here
library('stringr')
```

```
## Warning: package ’stringr’ was built under R version 4.2.3
```

```r
any(tolower(substr(sentences, length(sentences), length(sentences))) == 'z')
```

```
## [1] FALSE
```

Problem #07 - Chapter 12 Exercise #05

```r
# Show your work here
all(length(words)<=11)
```

```
## [1] FALSE
```

Problem #08 - Chapter 13 Exercise #01

```
# Show your work here
length(PossiblyInfinite[PossiblyInfinite==Inf | PossiblyInfinite==-Inf])
```

```
## [1] 89
```

Problem #09 - Chapter 14 Exercise #02ab

```
# Show your work here
anyNA(firstNames)
```

```
## [1] TRUE
```

```
anyNA(lastNames)
```

```
## [1] FALSE
```

Problem #10 - Chapter 14 Exercise #03

```r
# Show your work here
which(is.na(MissingValues01))
```

```
## [1]  58  86 304
```

```r
which(is.na(MissingValues02))
```

```
## [1]  45 367 407 409
```

```r
which(is.na(MissingValues03))
```

```
## integer(0)
```

```r
which(is.na(firstNames))
```

```
## [1] 287 358 371
```

```r
which(is.na(lastNames))
```

```
## integer(0)
```

```r
which(is.na(idNumbers))
```

```
## [1]   7  33 136 301 308 378 460 466
```

Problem #11 - Chapter 14 Exercise #05a

```r
# Show your work here
casll<-which(is.na(lastNames))
firstNames[as.integer(casll)]
```

```
## character(0)
```

```r
#Since there is no NA in last names nothing
#is shown for firstrNames
```