HW1

Lin Pin Tzu (Ruby)

2022-07-01

1) Write a function that will find the sum of the natural log, the common log and a log of base 2 for any given positive number. Use your function to find answers for the first five even integers. Show all details and structure of your function. You should get five answers. The first two answers are 1.994177 and 3.988354. (Your output should show the other three)

log = natural log , log10 = common log

```
evens <- function(e) subset(e, e %% 2 == 0)
evens(1:10)
```

```
## [1]  2  4  6  8 10
```

```
sumlog <- function(evens) {
  log(evens)+log10(evens)+log2(evens)
}
sumlog(c(2,4,6,8,10)) #five even integers
```

```
## [1] 1.994177 3.988354 5.154873 5.982532 6.624513
```

```
sumlog(c(6,8,10))
```

```
## [1] 5.154873 5.982532 6.624513
```

2) Use the if-else structure to print the statement "This is a big number" if the square of a value is greater than or equal to 100 and the following statement is printed if the square of the number is less than 100, "This is not a big number". Use and show values of assignment and if-else structures that will output both statements.

```
bignumber <- 20^2
if (bignumber > 100  & bignumber >= 100) {
  print('This is a big number')
} else if (bignumber < 100  & bignumber <= 100) {
  print('This is not a big number')
}
```

```
## [1] "This is a big number"
```

```
notbig <- 9^2
if (notbig > 100  & notbig >= 100) {
  print('This is a big number')
```

```
} else if (notbig < 100  & notbig <= 100) {
  print('This is not a big number')
}
```

```
## [1] "This is not a big number"
```

3) For the following if-else-if coding structure, make an adjustment so that it prints "Team A won"

```
team_A <- 4 # Number of goals scored by Team A
team_B <- 2# Number of goals scored by Team B
if (team_A > team_B){
  print ("Team A won")
} else if (team_A < team_B){
  print ("Team B won")
} else {
  "Team A & B tied"
}
```

```
## [1] "Team A won"
```

4) Write an if- else if- else sequence of commands that will output the following statements and appropriate output; If a value is divisible by 3 and 5 the output statement is "divisible by Three and Five", If a value is divisible by 3 and 4, the output statement is "divisible by Three and Four, If a value is a number that does not fall into either category, the output statement should be"neither". Use your function to show output statements for values 16, 45, and 24.

```
number <- c(16)
if (number %% 3==0 && number %% 5 == 0) {
  print('divisible by Three and Five')
} else if (number %% 3==0 && number %% 4 == 0) {
  print('divisible by Three and Four')
} else {
  print('neither')
}
```

```
## [1] "neither"
```

```
number <- c(45)
if (number %% 3==0 && number %% 5 == 0) {
  print('divisible by Three and Five')
} else if (number %% 3==0 && number %% 4 == 0) {
  print('divisible by Three and Four')
} else {
  print('neither')
}
```

```
## [1] "divisible by Three and Five"
```

```
number <- c(24)
if (number %% 3==0 && number %% 5 == 0) {
  print('divisible by Three and Five')
} else if (number %% 3==0 && number %% 4 == 0) {
  print('divisible by Three and Four')
} else {
  print('neither')
}
```

```
## [1] "divisible by Three and Four"
```

5) Use piping and a dplyr command show and use R code to produce the following modified mpg data table that contains only quantitative variables. Now use the "special loop coding chunk", illustrated in class, to produce the variance for all variables of the modified data table.

```
#piping
data(mpg)
loogmpg <- mpg %>%
  select(c(displ,cyl,cty,hwy))
head(loogmpg)
```

```
## # A tibble: 6 x 4
##   displ   cyl   cty   hwy
##   <dbl> <int> <int> <int>
## 1   1.8     4    18    29
## 2   1.8     4    21    29
## 3   2       4    20    31
## 4   2       4    21    30
## 5   2.8     6    16    26
## 6   2.8     6    18    26
```

```
# loop
loop <- data.frame(displ=mpg$displ,
                   cyl=mpg$cyl,
                   cty=mpg$cty,
                   hwy=mpg$hwy)
head(loop)
```

```
##   displ cyl cty hwy
## 1   1.8   4  18  29
## 2   1.8   4  21  29
## 3   2.0   4  20  31
## 4   2.0   4  21  30
## 5   2.8   6  16  26
## 6   2.8   6  18  26
```

```
mpgloop <- vector("double", ncol(loop))
for (i in seq_along(loop)) {
 mpgloop[[i]] <- var(loop[[i]])
}
```

```
mpgloop
```

```
## [1]  1.669158  2.597043 18.113074 35.457779
```

6) Construct a for loop (as illustrated in the notes) that will produce the difference between the cube and the square for each prime number between 10 and 30. (There are 6 answers. The first answer is 1210, your for loop coding should produce the other five answers)

```
Primes(10,30)
```

```
## [1] 11 13 17 19 23 29
```

```
primenumber<- data.frame(prime=c(11,13,17,19,23,29))
for (i in primenumber){
  print((i^3)-(i^2))
}
```

```
## [1]  1210  2028  4624  6498 11638 23548
```