trial

Laura

2025-03-04

Solutions to the lab session

```r
# Exercise 1:
# Create a vector with name x with a sequence of 50 equally-spaced values between 0 and 1.
x<-seq(0,1,length.out = 50)

length = length(x)

# a. Put names to the elements of x
names(x)<-paste("x_",1:length)
names(x)
```

```
##  [1] "x_ 1"  "x_ 2"  "x_ 3"  "x_ 4"  "x_ 5"  "x_ 6"  "x_ 7"  "x_ 8"  "x_ 9"
## [10] "x_ 10" "x_ 11" "x_ 12" "x_ 13" "x_ 14" "x_ 15" "x_ 16" "x_ 17" "x_ 18"
## [19] "x_ 19" "x_ 20" "x_ 21" "x_ 22" "x_ 23" "x_ 24" "x_ 25" "x_ 26" "x_ 27"
## [28] "x_ 28" "x_ 29" "x_ 30" "x_ 31" "x_ 32" "x_ 33" "x_ 34" "x_ 35" "x_ 36"
## [37] "x_ 37" "x_ 38" "x_ 39" "x_ 40" "x_ 41" "x_ 42" "x_ 43" "x_ 44" "x_ 45"
## [46] "x_ 46" "x_ 47" "x_ 48" "x_ 49" "x_ 50"
```

```r
# b. Mean and standard deviation of x
mx<- mean(x)
sx<-sd(x)

# c. Find how many elements of x are above mx in more than two standard deviations

elements<-x[x>mx+2*sx]
print(elements)
```

```
## named numeric(0)
```

```r
# Replace them with NA
x[x > mx + 2 * sx] <- NA

# d. Compute again
new_mx <- mean(x, na.rm = TRUE)
new_varx <- var(x, na.rm = TRUE)

print(new_mx)
```

```
## [1] 0.5
```

```r
print(new_varx)
```

```
## [1] 0.08850479
```

```r
#Exercise 2
a<- -2
b<- 2
n<- 1000
h<- (b-a)/n

values<- seq(a,b,length.out = n+1)

f_x<-(x<(-1))*(1) + (-1<=x & x<0)*(log(x^2)) + (0<=x & x<1)*(log(x^2 + 1)) + (x>=1)*(2)
f_x
```

```
##          x_ 1         x_ 2         x_ 3         x_ 4         x_ 5         x_ 6
##           NaN 0.0004164064 0.0016645863 0.0037414303 0.0066417845 0.0103584933
##          x_ 7         x_ 8         x_ 9        x_ 10        x_ 11        x_ 12
## 0.0148824574 0.0202027073 0.0263064903 0.0331793695 0.0408053347 0.0491669203
##         x_ 13        x_ 14        x_ 15        x_ 16        x_ 17        x_ 18
## 0.0582453328 0.0680205817 0.0784716154 0.0895764586 0.1013123495 0.1136558764
##         x_ 19        x_ 20        x_ 21        x_ 22        x_ 23        x_ 24
## 0.1265831109 0.1400697377 0.1540911790 0.1686227124 0.1836395827 0.1991171058
##         x_ 25        x_ 26        x_ 27        x_ 28        x_ 29        x_ 30
## 0.2150307648 0.2313562981 0.2480697791 0.2651476873 0.2825669718 0.3003051061
##         x_ 31        x_ 32        x_ 33        x_ 34        x_ 35        x_ 36
## 0.3183401356 0.3366507177 0.3552161545 0.3740164190 0.3930321757 0.4122447951
##         x_ 37        x_ 38        x_ 39        x_ 40        x_ 41        x_ 42
## 0.4316363627 0.4511896842 0.4708882857 0.4907164105 0.5106590126 0.5307017471
##         x_ 43        x_ 44        x_ 45        x_ 46        x_ 47        x_ 48
## 0.5508309584 0.5710336657 0.5912975476 0.6116109244 0.6319627394 0.6523425398
##         x_ 49        x_ 50
## 0.6727404558 2.0000000000
```

```r
f_x[is.nan(f_x)]<-0
f_x
```

```
##          x_ 1         x_ 2         x_ 3         x_ 4         x_ 5         x_ 6
## 0.0000000000 0.0004164064 0.0016645863 0.0037414303 0.0066417845 0.0103584933
##          x_ 7         x_ 8         x_ 9        x_ 10        x_ 11        x_ 12
## 0.0148824574 0.0202027073 0.0263064903 0.0331793695 0.0408053347 0.0491669203
##         x_ 13        x_ 14        x_ 15        x_ 16        x_ 17        x_ 18
## 0.0582453328 0.0680205817 0.0784716154 0.0895764586 0.1013123495 0.1136558764
##         x_ 19        x_ 20        x_ 21        x_ 22        x_ 23        x_ 24
## 0.1265831109 0.1400697377 0.1540911790 0.1686227124 0.1836395827 0.1991171058
##         x_ 25        x_ 26        x_ 27        x_ 28        x_ 29        x_ 30
## 0.2150307648 0.2313562981 0.2480697791 0.2651476873 0.2825669718 0.3003051061
##         x_ 31        x_ 32        x_ 33        x_ 34        x_ 35        x_ 36
## 0.3183401356 0.3366507177 0.3552161545 0.3740164190 0.3930321757 0.4122447951
##         x_ 37        x_ 38        x_ 39        x_ 40        x_ 41        x_ 42
## 0.4316363627 0.4511896842 0.4708882857 0.4907164105 0.5106590126 0.5307017471
##         x_ 43        x_ 44        x_ 45        x_ 46        x_ 47        x_ 48
## 0.5508309584 0.5710336657 0.5912975476 0.6116109244 0.6319627394 0.6523425398
##         x_ 49        x_ 50
## 0.6727404558 2.0000000000
```

```r
# Exercise 3

set.seed(1)
```

```r
x<-rnorm(100, mean=160, sd=10)

# a. Compute the first and last quartiles using the function quantile.
?quantile
quartiles <- quantile(x, probs = c(0.25, 0.75))
quartil_1 <- quartiles[1]
quartil_3 <- quartiles[2]

# b. Create a factor with 3 levels, encoding the values in x
x_factor <- cut(x, breaks = c(-Inf, quartil_1, quartil_3, Inf), labels = c("low", "medium", "high"))

# c. Generate a second vector y
y<-x-100+rnorm(100, mean = 0, sd = 1)
y
```

```
##   [1] 53.11510 61.87855 50.73279 76.11084 62.64049 53.56260 65.59100 68.29342
##   [9] 66.14200 58.62829 74.48208 63.43679 55.21988 37.20230 71.04193 59.15786
##  [17] 59.51810 69.15925 68.70640 65.76168 68.68382 69.16440 60.53107 39.92693
##  [25] 66.09807 60.15138 58.36848 45.25484 54.53684 63.85515 73.64696 58.38323
##  [33] 64.40821 57.94356 46.53596 54.31360 55.75612 58.87859 70.34816 67.57486
##  [41] 56.44040 58.64297 65.30466 65.10310 51.99652 52.17423 65.73299 67.70272
##  [49] 57.59024 67.17047 64.43125 53.86118 63.09313 47.77701 72.84278 78.72881
##  [57] 57.32781 48.93739 64.31277 60.51874 84.44128 59.36895 67.95588 61.16644
##  [65] 51.94802 64.09403 41.69539 73.23105 61.38813 81.93366 67.06307 53.00634
##  [73] 66.56426 50.58187 47.12967 62.87974 56.35472 62.08630 61.77081 55.31270
##  [81] 53.08199 59.63211 72.00079 43.29708 66.46048 63.17075 72.09559 56.19208
##  [89] 63.26998 61.74488 54.39770 72.48069 70.87228 67.83251 74.66025 64.53688
##  [97] 48.67524 53.25150 48.16585 54.88492
```

```r
# d. Compute a summary of y for each level of the factor
summary_by_factor <- tapply(y, x_factor, summary)
print(summary_by_factor)
```

```
## $low
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   37.20   47.13   50.73   49.47   53.12   55.31
##
## $medium
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   54.31   58.63   61.76   61.40   64.38   67.96
##
## $high
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   65.30   68.68   71.04   71.94   73.65   84.44
```

```r
# Exercise 4
# a. Compute the sum of the elements in the vector using sum
n <- 30
a1 <- 1
d <- 1.3
an <- a1 + (n - 1) * d
prog_arit <- seq(a1, an, by = d)
sum <- sum(prog_arit)
sum_formula <- n * (a1 + an) / 2
```

```r
# b. Compute the std using sd.
std <- sd(prog_arit)
std_formula <- abs(d) * sqrt(n * (n + 1) / 12)

# c. Compute the product of the elements in the vector using the function prod.
product <-prod(prog_arit)
prod_formula <- d^n * gamma(a1/d + (n-1))/gamma(a1/d)
print(product)
```

```
## [1] 2.632626e+35
```

```r
print(prod_formula)
```

```
## [1] 8.843447e+33
```