PSTAT 10 Fall 25 — Homework 1

Tejas Patel

Due Date: Thursday, October 9, 2025, 11:59 PM

Homework policy
Some Homework problems or related questions may appear on the exams. Homework solutions will be posted after due date.
Submission format: Quarto PDF (Use the Quarto homework template in Canvas. Make sure you have installed the `tiny_tex()` package)

Question 1 (Displaying output).

```
greekLetters = c("alpha", "beta", "Gamma", "delta", "beta", "Kappa")
print(greekLetters);
```

```
[1] "alpha" "beta"  "Gamma" "delta" "beta"  "Kappa"
```

```
cat(greekLetters);
```

```
alpha beta Gamma delta beta Kappa
```

```
cat(greekLetters, sep = ",");
```

```
alpha,beta,Gamma,delta,beta,Kappa
```

The print() function prints out the vector with quotes around each element and a space separating the elements, cat() just prints them out with a space between each element, and adding a sep parameter lets them have a specified delimiter between the elements

```
print(length(greekLetters))
```

```
[1] 6
```

```
for (greekLetter in greekLetters) {
  print(nchar(greekLetter))
}
```

```
[1] 5
[1] 4
[1] 5
[1] 5
[1] 4
[1] 5
```

```
print(unique(greekLetters))
```

```
[1] "alpha" "beta"  "Gamma" "delta" "Kappa"
```

```
print(toupper(greekLetters))
```

```
[1] "ALPHA" "BETA"  "GAMMA" "DELTA" "BETA"  "KAPPA"
```

```
print(tolower(greekLetters))
```

```
[1] "alpha" "beta"  "gamma" "delta" "beta"  "kappa"
```

```
print(seq_along(greekLetters))
```

```
[1] 1 2 3 4 5 6
```

```
for(greekLetter in greekLetters) {
  print(greekLetter)
  print(which(greekLetters == greekLetter))
}
```

```
[1] "alpha"
[1] 1
[1] "beta"
[1] 2 5
[1] "Gamma"
[1] 3
[1] "delta"
[1] 4
[1] "beta"
[1] 2 5
[1] "Kappa"
[1] 6
```

```
paste(seq_along(greekLetters), ": ",  greekLetters)
```

```
[1] "1 :  alpha" "2 :  beta"  "3 :  Gamma" "4 :  delta" "5 :  beta"
[6] "6 :  Kappa"
```

```
cat(paste(seq_along(greekLetters), ": ", greekLetters), sep = ", ")
```

```
1 :  alpha, 2 :  beta, 3 :  Gamma, 4 :  delta, 5 :  beta, 6 :  Kappa
```

For paste, there are quotes separating the elements of the vector, where in cat the separation is done be a comma that we specified

Question 2 (Vector operations)

```
sign_ups = c(4, 9, 2, 15, 11, 5, 7, 0, 13, 3)
print(rev(sort(sign_ups))[1:3])
```

```
[1] 15 13 11
```

```
paste(mean(sign_ups), " sign ups per day")
```

```
[1] "6.9  sign ups per day"
```

```
print(sign_ups * 1.1)
```

```
 [1]  4.4  9.9  2.2 16.5 12.1  5.5  7.7  0.0 14.3  3.3
```

```
per_hour_sign_ups= sign_ups / 24
print(per_hour_sign_ups)
```

```
 [1] 0.16666667 0.37500000 0.08333333 0.62500000 0.45833333 0.20833333
 [7] 0.29166667 0.00000000 0.54166667 0.12500000
```

```
print(length(which(sign_ups == 10)))
```

```
[1] 0
```

```
print(length(which(sign_ups == 0)))
```

```
[1] 1
```

```
print(sum(sign_ups))
```

```
[1] 69
```

```
print(sd(sign_ups))
```

```
[1] 4.976612
```

```
print(range(sign_ups))
```

```
[1]  0 15
```

```
print(cumsum(sign_ups))
```

```
 [1]  4 13 15 30 41 46 53 53 66 69
```

```r
print(diff(sign_ups, 1, 1))
```

```
[1]   5  -7  13  -4  -6   2  -7  13 -10
```

```r
for (sign_up in sign_ups){
  print((sign_up - mean(sign_ups)) / sd(sign_ups))
}
```

```
[1] -0.5827258
[1] 0.4219738
[1] -0.9846056
[1] 1.627613
[1] 0.8238537
[1] -0.3817858
[1] 0.02009399
[1] -1.386485
[1] 1.225733
[1] -0.7836657
```

```r
print(length(c(11, 5, 7, 9) %in% sign_ups))
```

```
[1] 4
```

I would say the mean is representative of the data set, including the 0 and 15 sign up days. The standard deviation can be high but that doesn't necessarily mean the mean is not representative of the population

Question 3

```r
vector1=rep(c(1,2,3,4), times=4)
print(vector1)
```

```
 [1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
```

```r
vector2 = seq(0, 12, 0.25)
print(vector2)
```

```
 [1]  0.00  0.25  0.50  0.75  1.00  1.25  1.50  1.75  2.00  2.25  2.50  2.75
[13]  3.00  3.25  3.50  3.75  4.00  4.25  4.50  4.75  5.00  5.25  5.50  5.75
[25]  6.00  6.25  6.50  6.75  7.00  7.25  7.50  7.75  8.00  8.25  8.50  8.75
[37]  9.00  9.25  9.50  9.75 10.00 10.25 10.50 10.75 11.00 11.25 11.50 11.75
[49] 12.00
```

Question 4

```
print(sum(30:60))
```

```
[1] 1395
```

```
print(sum(80:100))
```

```
[1] 1890
```

Question 5 (Data cleaning)

```
amount_chr = c("12.50","8","","NA","19.2"," 3.0 ")
fever_chr = c("98.6","101.3","99.4","103.2")
likert_fac = factor(c("1","2","3","2","4","5","3"))
print(as.numeric(replace(trimws(amount_chr), trimws(amount_chr) == "", NA)))
```

```
[1] 12.5  8.0   NA   NA 19.2  3.0
```

```
fever_num = as.numeric(fever_chr)
print(fever_num >= 100)
```

```
[1] FALSE  TRUE FALSE  TRUE
```

```
print(as.integer(fever_num >= 100))
```

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

```
print(as.integer(as.character(likert_fac)))
```

```
[1] 1 2 3 2 4 5 3
```

```
cat("n_amount=", length(as.numeric(replace(trimws(amount_chr), trimws(amount_chr) == "", NA)))
```

```
n_amount=,6,, n_missing_amount=,2,, n_fever=,2
```

Question 6

```
stocks = c(8, 3, 14, 2, 11)
print(stocks[3])
```

```
[1] 14
```

```
print(stocks[-2])
```

```
[1]  8 14  2 11
```

```
print(stocks[-c(1,4)])
```

```
[1]  3 14 11
```

```
stocks[4] = 20
stocks[c(1,3)] = 5
print(stocks)
```

```
[1]  5  3  5 20 11
```

```
names(stocks) = c("A", "B", "C", "D", "E")
print(stocks)
```

```
 A  B  C  D  E
 5  3  5 20 11
```

```r
stocks[c("B","D")]
```

```
 B  D
 3 20
```