STAT 40001/ STAT 50001 Statistical Computing Fall 2023

Lab-1

Calculate the following

a) 47+8-13×56+78

b) 46-87+98-57

c) 46-87+98-57

d) 4+(35mod 6)+9 ln(5)

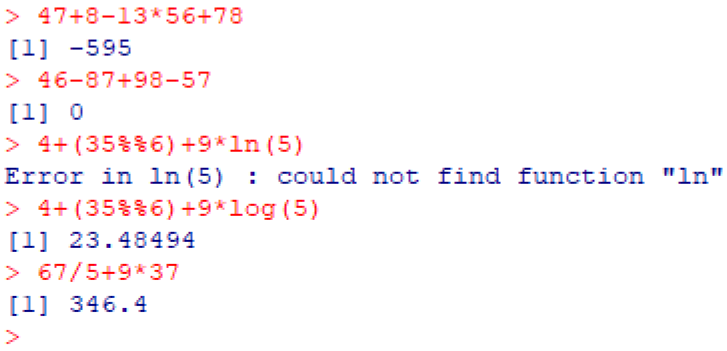
e) 67÷5+9×37

f) |-7|+|5|+log(10)

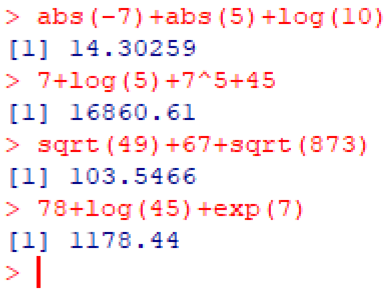
g) 7+log(5)+7^5+45

h) $\sqrt{49}+67+\sqrt{873}$

i) $78+\ln(45)+e^7$



```
> 47+8-13*56+78
[1] -595
> 46-87+98-57
[1] 0
> 4+(35%%6)+9*ln(5)
Error in ln(5) : could not find function "ln"
> 4+(35%%6)+9*log(5)
[1] 23.48494
> 67/5+9*37
[1] 346.4
>
```



```
> abs(-7)+abs(5)+log(10)
[1] 14.30259
> 7+log(5)+7^5+45
[1] 16860.61
> sqrt(49)+67+sqrt(873)
[1] 103.5466
> 78+log(45)+exp(7)
[1] 1178.44
> 
```

We have several way of rounding the numbers including

`ceiling` takes a single numeric argument `x` and returns a numeric vector containing the smallest integers not less than the corresponding elements of `x`.

`floor` takes a single numeric argument `x` and returns a numeric vector containing the largest integers not greater than the corresponding elements of `x`.

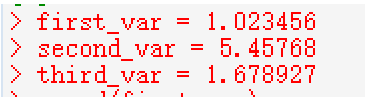
`trunc` takes a single numeric argument `x` and returns a numeric vector containing the integers formed by truncating the values in `x` toward `0`.

`round` rounds the values in its first argument to the specified number of decimal places (default 0).

`signif` rounds the values in its first argument to the specified number of significant digits.

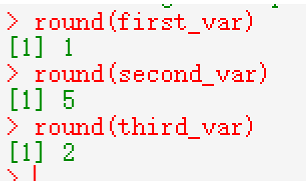
`zapsmall` determines a `digits` argument `dr` for calling `round(x, digits = dr)` such that values “close to zero” (compared with the maximal absolute value) are “zapped”, i.e., treated as `0`.

Given two numbers 1.023456, 5.45768, and 1.678927 use the following options



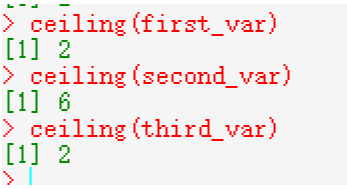
```
> first_var = 1.023456
> second_var = 5.45768
> third_var = 1.678927
```

i) round



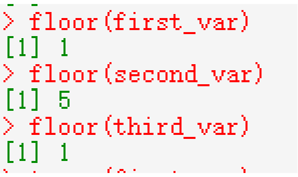
```
> round(first_var)
[1] 1
> round(second_var)
[1] 5
> round(third_var)
[1] 2
> 
```

ii) ceiling



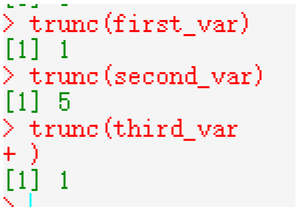
```
> ceiling(first_var)
[1] 2
> ceiling(second_var)
[1] 6
> ceiling(third_var)
[1] 2
> 
```

iii) floor



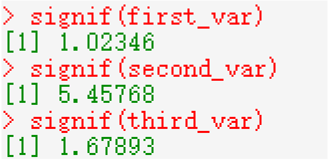
```
> floor(first_var)
[1] 1
> floor(second_var)
[1] 5
> floor(third_var)
[1] 1
```

iv) trunc



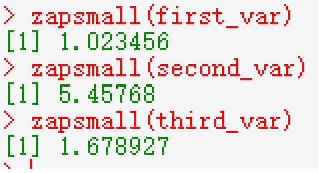
```
> trunc(first_var)
[1] 1
> trunc(second_var)
[1] 5
> trunc(third_var
+ )
[1] 1
> 
```

v) signif



```
> signif(first_var)
[1] 1.02346
> signif(second_var)
[1] 5.45768
> signif(third_var)
[1] 1.67893
```

vi) zapsmall



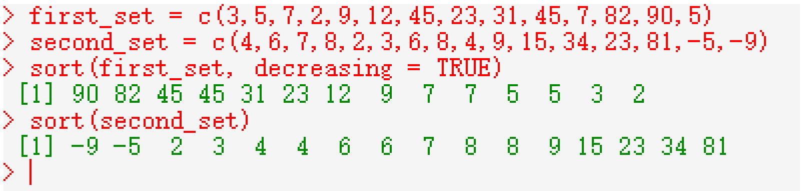
```
> zapsmall(first_var)
[1] 1.023456
> zapsmall(second_var)
[1] 5.45768
> zapsmall(third_var)
[1] 1.678927
```

j) Sort the data in decreasing order:

3, 5, 7, 2, 9, 12, 45, 23, 31, 45, 7, 82, 90, 5

k) Sort the data in increasing order

4,6,7,8,2,3,6,8,4,9,15,34,23,81,-5,-9



```
> first_set = c(3,5,7,2,9,12,45,23,31,45,7,82,90,5)
> second_set = c(4,6,7,8,2,3,6,8,4,9,15,34,23,81,-5,-9)
> sort(first_set, decreasing = TRUE)
 [1] 90 82 45 45 31 23 12  9  7  7  5  5  3  2
> sort(second_set)
 [1] -9 -5  2  3  4  4  6  6  7  8  8  9 15 23 34 81
> |
```