Iteration

Xzm

2018年2月18日

```
library("tidyverse")
library("magrittr")
```

```
df <- tibble(
  a = rnorm(10),
  b = rnorm(10),
  c = rnorm(10),
  d = rnorm(10)
)
```

compute the median of each column

```
median(df$a)
#> [1] -0.5103426
median(df$b)
#> [1] 0.2124157
median(df$c)
#> [1] 0.2462326
median(df$d)
#> [1] -0.3760143
```

never copy and paste more than twice

```
output <- vector('double', ncol(df)) # 1. output
for (i in seq_along(df)) {           # 2. sequence
  output[[i]] <- median(df[[i]])     # 3. body
}
output
#> [1] -0.5103426  0.2124157  0.2462326 -0.3760143
```

21.2.1 Exercises

1. Write for loops to:

- Compute the mean of every column in *mtcars*

```
mt.mean <- vector()
for (var in names(mtcars)) {
  mt.mean[[paste0(var, '.mean')]] <- mean(mtcars[[var]], na.rm = TRUE)
}
mt.mean
#>   mpg.mean   cyl.mean  disp.mean    hp.mean  drat.mean    wt.mean
#>  20.090625   6.187500 230.721875 146.687500   3.596563   3.217250
#>  qsec.mean    vs.mean    am.mean  gear.mean  carb.mean
#>  17.848750   0.437500   0.406250   3.687500   2.812500
```

- Determine the type of each column in *nycflights13::flights*

```
flt <- nycflights13::flights
flt.type <- vector()
for (xvar in names(flt)) {
  flt.type[[xvar]] <- typeof(flt[[xvar]])
}
flt.type
#>          year         month           day      dep_time sched_dep_time
#>     "integer"     "integer"     "integer"     "integer"     "integer"
#>     dep_delay      arr_time sched_arr_time     arr_delay       carrier
#>      "double"     "integer"     "integer"      "double"   "character"
#>        flight       tailnum        origin          dest      air_time
#>     "integer"   "character"   "character"   "character"      "double"
#>      distance          hour        minute     time_hour
#>      "double"      "double"      "double"      "double"
```

- Compute the number of unique values in each column of *iris*

```
num_unique_col <- vector()
for (xvar in names(iris)) {
  num_unique_col[[xvar]] <- unique(iris[[xvar]]) %>% length()
}
num_unique_col
#> Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
#>           35           23           43           22            3
```

- Generate 10 random normals for each of $\mu = -10, 0, 10, and 100$

```
mus <- c(-10, 0, 10, 100)
l <- vector('list')
for (mu in mus) {
  l[[paste0('mu = ', mu)]] <- rnorm(10, mean = mu)
}
l
#> $`mu = -10`
#>  [1]  -9.745002 -10.553238  -8.594891 -10.795461 -11.566514 -11.040579
#>  [7]  -8.980066 -10.702082  -9.026684 -10.076818
#>
#> $`mu = 0`
#>  [1]  0.8929249245 -0.7775030885  0.4367971056  0.4134439348  0.9763417720
#>  [6]  1.1465004990  1.2172716875  0.0004800131  0.7551250562  0.3424035105
#>
#> $`mu = 10`
#>  [1] 10.168473 11.397067  9.320905 10.737629  9.139276 10.421230 11.450543
#>  [8] 10.194392  9.308795 11.339860
#>
#> $`mu = 100`
#>  [1] 102.73611  99.05590  98.21894  99.28394 100.91108  99.22781  99.21792
#>  [8]  99.56780  99.33244 101.38951
```

column summary

```
col_summary <- function(df, fun) {
#  namefun <- fun
  funname <- enquo(fun)
  out <- vector()
  for (i in seq_along(df)) {
    out[[i]] <- fun(df[[i]])
  }

  names(out) <- paste0(names(df),'.', quo_name(funname))
  out
}
col_summary(mtcars, mean)
#>   mpg.mean   cyl.mean  disp.mean    hp.mean  drat.mean    wt.mean
#>  20.090625   6.187500 230.721875 146.687500   3.596563   3.217250
#>  qsec.mean    vs.mean    am.mean  gear.mean  carb.mean
#>  17.848750   0.437500   0.406250   3.687500   2.812500
```