

Домашняя работа №5 (от 14.11.2019)

Выполнил: студент группы 18ПИ-1 Саратовцев Артем

Теория:

Стандартная ошибка среднего — параметр в статистике, который характеризует выборочное распределение, в частности стандартное отклонение выборочного среднего, рассчитанное по выборке размера n из генеральной совокупности. Она рассчитывается как отношение стандартного отклонения к квадратному корню из объема выборки:

$$s_{\bar{x}} = \frac{s}{\sqrt{n}}$$

Ассиметрия — коэффициент, характеризующий асимметричность распределения признака в совокупности. Находится по формуле:

$$K_A = \frac{\bar{x} - M_o}{\sigma}$$

Эксцесс — отклонение вершины эмпирического распределения вверх или вниз от вершины кривой нормального распределения. Находится по формуле:

$$E_x = \frac{M_4}{\sigma^4} - 3$$

Экспоненциальное распределение — распределение, которое описывает интервалы времени между независимыми событиями, происходящими со средней интенсивностью. Количество наступлений такого события за некоторый отрезок времени описывается дискретным распределением Пуассона. Формула Пуассона:

$$P_n(m) \approx \frac{\lambda^m}{m!} e^{-\lambda}$$

Функции:

na.omit(x) — убирает значения NA из объекта X

length(x) — находит число элементов в X

sd(x, na.rm = FALSE) — находит стандартное отклонение значений из выборки X

na.rm – аргумент, служащий для игнорирования значений NA в выборке

rexp(N, rate =) - экспоненциальное распределение

N – кол-во элементов

rate - параметр λ

mean(x) – находит средний элемент объекта X

cbind() - нужна для объединения векторов

str(x) – служит для получения информации о внутренней структуре объекта X

aggregate(x, by, FUN) – разбивает таблицу данных X на отдельные наборы данных, применяет к этим наборам определенную функцию FUN и возвращает результат в удобном для чтения формате.

Код 1 задания:

```
# task 1
# Задание 1. Написать функцию в R для вычисления среднего квадратичного отклонения
# (стандартной ошибки) среднего. Учесть возможность существования в выборке NA.

findSE <- function(arr) {
  # MQD - mean square deviation
  # SE - Standard error
  res <- sd(arr, na.rm = TRUE)/sqrt(length(na.omit(arr)))
  return(res)
}

data(mtcars); mtcars
mtcars$mpg
a <- c(mtcars$mpg, NA, NA, NA);a
res1 <- findSE(mtcars$mpg); res1
res2 <- findSE(a); res2
```

Результат выполнения:

```
> res1 <- findSE(mtcars$mpg); res1
[1] 1.065424
> res2 <- findSE(a); res2
[1] 1.065424
```

Код 2 задания:

```
# task 2
# 2. Написать функции для вычисления асимметрии и эксцесса и вычислить эти
# величины для выборки, объемом n = 100, взятой генеральной совокупности с
# экспоненциальной функцией распределения с параметром  $\lambda = 3$ . При написании функций
# учесть возможность существования в выборке NA.

findCS <- function(arr){
  # CS - coefficient of skewness
  X <- mean(arr, na.rm = TRUE); X
  M <- median(arr, na.rm = TRUE); M
  sigma <- findSE(arr); sigma
  res <- (X - M) / sigma
  return(res)
}

findCA <- function(arr) {
  # coefficient of asymmetry
  m4 <- mean(arr^4); m4
  sigma <- findSE(arr); sigma
  res <- m4 / sigma^4
  return(res)
}

N <- 100
lambda <- 3
arr <- rexp(N, rate = lambda); arr
res <- findCS(arr); res
res1 <- findCA(arr); res1
```

Результат выполнения:

```
> res <- findCS(arr); res
[1] 2.874581
> res1 <- findCA(arr); res1
[1] 218829.2
```

Код 3 задания:

```
# task 3
# 3. Для всех нефакторных переменных из таблице данных mtcars вычислить
# одновременно median и mean с группировкой по переменным am и vs, включив в одну из
# нефакторных переменных 2 NA.

meanAndMedian <- function(a) {
  na.omit(a)
  return(c(median = median(a), mean = mean(a)))
}

?mtcars
mtcars
str(mtcars)
df <- cbind(mtcars[1:7],mtcars[10:11]); df

df$mpg[3] <- NA; df$cyl[3] <- NA; df$disp[3] <- NA; df$hp[3] <- NA; df$drat[3] <- NA; df$wt[3] <- NA; df$qsec[3] <- NA;

df$mpg; df$cyl; df$disp; df$hp; df$drat; df$wt; df$qsec;

#aggregate(df,by=list(mtcars$am, mtcars$vs), FUN = (median),na.rm=TRUE)
#aggregate(df,by=list(mtcars$am, mtcars$vs), FUN = (mean),na.rm=TRUE)

aggregate(df,by=list(mtcars$am, mtcars$vs), FUN = (meanAndMedian))
```

Результат выполнения:

```
> df$mpg[3] <- NA; df$cyl[3] <- NA; df$disp[3] <- NA; df$hp[3] <- NA; df$drat[3] <- NA; df$wt[3] <- NA; df$qsec[3] <- NA;
> df$mpg; df$cyl; df$disp; df$hp; df$drat; df$wt; df$qsec;
[1] 21.0 21.0 NA 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7 15.0 21.4
[1] 6 6 NA 6 8 6 8 4 4 6 6 8 8 8 8 8 8 4 4 4 8 8 8 8 4 4 4 8 6 8 4
[1] 160.0 160.0 NA 258.0 360.0 225.0 360.0 146.7 140.8 167.6 167.6 275.8 275.8 275.8 472.0 460.0 440.0 78.7 75.7 71.1 120.1 318.0 304.0 350.0 400.0 79.0 120.3 95.1 351.0
[30] 145.0 301.0 121.0
[1] 110 110 NA 110 175 105 245 62 95 123 123 180 180 180 205 215 230 66 52 65 97 150 150 245 175 66 91 113 264 175 335 109
[1] 3.90 3.90 NA 3.08 3.15 2.76 3.21 3.69 3.92 3.92 3.92 3.07 3.07 3.07 2.93 3.00 3.23 4.08 4.93 4.22 3.70 2.76 3.15 3.73 3.08 4.08 4.43 3.77 4.22 3.62 3.54 4.11
[1] 2.620 2.875 NA 3.215 3.440 3.460 3.570 3.190 3.150 3.440 3.440 4.070 3.730 3.780 5.250 5.424 5.345 2.200 1.615 1.835 2.465 3.520 3.435 3.840 3.845 1.935 2.140 1.513 3.170
[30] 2.770 3.570 2.780
[1] 16.46 17.02 NA 19.44 17.02 20.22 15.84 20.00 22.90 18.30 18.90 17.40 17.60 18.00 17.98 17.82 17.42 19.47 18.52 19.90 20.01 16.87 17.30 15.41 17.05 18.90 16.70 16.90 14.50
[30] 15.50 14.60 18.60
> aggregate(df,by=list(mtcars$am, mtcars$vs), FUN = (meanAndMedian))
  Group.1 Group.2 mpg.median mpg.mean cyl.median cyl.mean disp.median disp.mean hp.median hp.mean drat.median drat.mean wt.median wt.mean qsec.median qsec.mean gear.median
1 0 0 15.20000 15.05000 8.000000 8.000000 355.0000 357.6167 180.0000 194.1667 3.075000 3.120833 3.810000 4.104083 17.35000 17.14250 3.000000
2 1 0 20.35000 19.75000 6.000000 6.333333 160.0000 206.2167 142.5000 180.8333 3.900000 3.935000 2.822500 2.857500 15.98000 15.79667 5.000000
3 0 1 21.40000 20.74286 6.000000 5.142857 167.6000 175.1143 105.0000 102.1429 3.700000 3.570000 3.215000 3.194286 20.00000 19.96714 4.000000
4 1 1 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA 4.000000
 gear.mean carb.mean carb.median
1 3.000000 3.000000 3.083333
2 4.666667 4.000000 4.666667
3 3.571429 2.000000 2.142857
4 4.142857 1.000000 1.428571
```