DaiglePS2.R

2011home

Sun Jan 28 16:15:02 2018

```
## Chris Daigle
## Homework 1

## Exercise 1
# Using the vector operation in R, obtain the slope coefficient
set.seed(2)
x <- rchisq(20, 5)

e <- rnorm(length(x), 0, 5)

y <- 3 + 2*x + e

b_hat <- (sum((x-mean(x))*(y-mean(y))))/(sum((x-mean(x))^2))
print(b_hat)
```

```
## [1] 2.155596
```

```
## Exercise 2
vector1 <- rbinom(100, 1, 0.5)
x2.3 <- seq(-1,1)
x2.2 <- 0
x2.1 <- rep(x2.2,100)
vector2 <- x2.1 + x2.3
```

```
## Warning in x2.1 + x2.3: longer object length is not a multiple of shorter
## object length
```

```
vector2
```

```
##   [1] -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0
##  [24]  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1
##  [47]  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1
##  [70] -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0  1 -1  0
##  [93]  1 -1  0  1 -1  0  1 -1
```

```
same <- which(vector1 == vector2)
print(same)
```

```
##  [1]  2  3  6 15 18 26 27 32 35 41 47 48 50 56 57 68 72 75 80 83 84 89 93
## [24] 98
```

```
## Exercise 3
poker_vector <- c(140, -50, 20, -120, 240)
roulette_vector <- c(-24, -50, 100, 350, 10)

days_vector <- c("Mon", "Tue", "Wed", "Thu", "Fri")
names(poker_vector) <- days_vector
names(roulette_vector) <- days_vector

poker_midweek <- poker_vector[c("Tue", "Thu")]

ave_poker_midweek <- mean(poker_midweek)

max_poker_vector <- max(poker_vector)
print(max_poker_vector)
```

```
## [1] 240
```

```
loc_max_poker_vector <- which.max(poker_vector)
print(loc_max_poker_vector)
```

```
## Fri
##   5
```

```
poker_rev_larger <- which(poker_vector > roulette_vector)
print(poker_rev_larger)
```

```
## Mon Fri
##   1   5
```

```
total <- poker_vector + roulette_vector
total_poker <- sum(poker_vector)
print(total_poker)
```

```
## [1] 230
```

```
total_roulette <- sum(roulette_vector)
print(total_roulette)
```

```
## [1] 386
```

```
total_poker > total_roulette
```

```
## [1] FALSE
```

```
## Exercise 4
set.seed(pi)
num <- sample(1:10, 1000, TRUE)
a <- sample(1:20, 20, TRUE)
num[num == a] <- NA
num
```

```
##    [1]  2  9  4  4  7 NA  2  3  6  7  6  6 NA NA  9  9  2  8  9  3  3  1  2
##   [24]  1  3  8  6 10  6  8  4  4  2  5  3  4  9  3  6  3  3  8  2  6  5  3
##   [47]  1  2  4  9  3  3  9 10  9 10  5  3  2  3  9  1  9  2  8  4  8  6  4
##   [70]  1  8  8 10 10  6  6  2  2  8  8  8  7  4  1 10  9  3  5  7 10  1  3
##   [93]  5  9  9  3  4  4  2  7  8  7 NA  8  7  4  1  5  1  4  4  1  2  2 10
##  [116]  8  9  7 NA  5  8  9  9  2  7  5  7  5  9  4  4  5  4  3  9  6  1  2
##  [139]  6  6  5 NA  7  9  9  3  7  9  6  2  8 NA  9  4  7  3  2  2  5  1 10
##  [162]  6  4  9  8  8  9  2  8  6  5 NA NA  9  2  6 NA  8  2  6  2 NA  2  1
##  [185]  3  4  9  2  4  4  5  9  4 NA  8  4  9 10  5  3  8  9  4  5  9  4  8
##  [208]  7  9  1  8  5 NA  3  1 10  3  2  7  8  4  9  4  6  4  8 10  1  3  4
##  [231]  4  5 NA  7  6  9  6  4  5  6 NA  8  7  1  7  4  4 10  8  8  3  3  7
##  [254]  7  3  6  1  5  9 10  8 10  6  7 10 NA  1  2  6  3  1  8 10 10  2  4
##  [277]  1  1  2  3  2  1  1  4  2  4  5 10  9  7  7  7  2  7  3  6 10  9  7
##  [300]  5 NA NA 10  8  7 NA  9  6  9 10  6  1 NA  7  2 10  1  9 10  7  3  3
##  [323]  9  3  9  6 10  9 10 10  8  6  9  2  2 10  8  1  9  2  6  3  4  5  6
##  [346]  5  3  1  4  9  9  3 10  9  1  4  1  8  2 10  7  2  2  9  7  2  6  5
##  [369]  4  3  3 NA  1  3  8 10 10  3  4  1 NA NA NA  4  8  8  8  5 10  4  1
##  [392]  9 NA 10  5  3  2  9 NA  5 NA  9  7  6  9  3  1  3  4  6 10  8  1 10
##  [415] 10  7  1  6  2  6  4 NA  2  9  9  6  4  2  8  6  3  1  7 NA 10  4 10
##  [438]  8  7  9  6  6 10  7  2  6  8  8  3  3 10  6  4  5  5  9  9  4  8  9
##  [461]  2  6  4  4  7  6  1  5  3  9  4 NA  2  1  2  7 NA  7  4  7  4  9 10
##  [484] NA  7  1  9 NA  9  3  8  4  4  5  7  9 NA  9 10  8  3  3 10  4  6  6
##  [507]  3  3 10  9  8  1  1 NA  3  9  5  6  1  6 10  8  8  8  4  2  4  5  7
##  [530]  2  2  4  3  9  6  7  6  2  9  9  7  3  5 NA  4  1  9  8  3  2  8 10
##  [553]  7  4  6  9  9  6  2  6  5  5  5  9  1  6  7  5 10  5  2  8 10 10  6
##  [576]  4 10  8  2 10  8 10 NA  9  3  3  7  8 10  2  7 10  4  1  9  9  4  9
##  [599]  8  9  2  7  8  9  3  2  4  2  1  6 10  8  2  8  6  1  6  9  2  4  7
##  [622] NA  8  7  8  1  1  3  2  1  5  1  5 NA  4  4 NA  4  6  5  2 NA  9  9
##  [645]  4  1  6  2 10  6  6  1  2  3  4  4  6  1  2  9 NA  9  4  8  8 10  2
##  [668]  8  8  6 10  5 NA  2 10  6 NA  5 NA  9  5  8 10  8  9  5  7  3  6  7
##  [691]  6  5  7 NA 10 10  8 10  6  7 NA 10  6  7  7  8  7  7  6  8  6  7  1
##  [714]  1 10  5 NA  1  7  1  8  6  7  3  1 10  2  1  9  9 10 10  4  9  7  2
##  [737] 10  8  5  2 10  6  4  9  3  1  5 NA  4  6  8  1  3  2  2  1 NA  4 10
##  [760]  5  4  6  2  1  3  4  4 NA  6  5  5  9  4  5  1  1  3  4  1  7  4  6
##  [783] NA  2  5 NA  5  3  9  3  2  9  3  1  2 10  5 10  7  4 10  8  9  3  7
##  [806] 10  6 NA  4  7  8  9 10  7  8  6 10  9 10  4  4  3  4  1  4 10  7  5
##  [829]  4  9  3  7 10  7  3  4  4  1 NA  4  8 10  6 NA  6  4  5  1  6  2  2
##  [852] NA  2  9  7  9 10 10  2  9  5 NA NA  3  9 NA 10 NA  2  4  8  8 NA  1
##  [875]  4  2 NA  4  4  4  6 NA  8  5  9  8  1  5  9  4  2  8  1  1  1  2  8
##  [898]  6 10  3  6  5  1  7  7  3  2  6  9 10  2  8  5 NA  7  3 10  6  6  8
##  [921]  8  9  8 NA  4 NA  3  2 10  2  8  7  4 NA  8  7 10  6  9  9  8  6  6
##  [944]  3  2  5  5  5 10  9  2  1  9  1  3  7  1 10  2  9  5  1 10  7  9 NA
##  [967]  8  1  9  6  4  8 10 NA  3  7  1  7  4  9 10  9  1  5  6  3 10  2  4
##  [990]  6  9 NA  1  1  2  3  2  6  5  6
```

```
which(is.na(num))
```

```
##  [1]   6  13  14 103 119 142 152 172 173 177 182 194 213 233 241 266 301
## [18] 302 306 313 372 381 382 383 393 399 401 422 434 472 477 484 488 497
## [35] 514 544 583 622 634 637 642 661 673 677 679 694 701 717 748 757 768
## [52] 783 786 808 839 844 852 862 863 866 868 873 877 882 914 924 926 934
## [69] 966 974 992
```