Lab3

Nick Gembs

2/6/2023

Exercises

1. Use apply to compute column sums of the matrix in the first section.

```
M <- matrix( 1:12, 4, 3 )
colSums(M)
```

```
## [1] 10 26 42
```

```
apply(X = M, MARGIN = 2, FUN = sum)
```

```
## [1] 10 26 42
```

2. Read in the airline data and use one of the apply functions to figure out how many missing values there are in each column of the airline data. Make sure the output is a named vector.

```
dat =
read.csv("C:/Users/Nick/Documents/GitHub/statcomp2023/datasets/airline_2019-07-01.csv")
sapply(dat, function(x) sum(is.na(x)))
```

```
##                             Year                          Quarter
##                                0                                0
##                            Month                       DayofMonth
##                                0                                0
##                        DayOfWeek                       FlightDate
##                                0                                0
##                Reporting_Airline         DOT_ID_Reporting_Airline
##                                0                                0
##     IATA_CODE_Reporting_Airline                      Tail_Number
##                                0                                0
## Flight_Number_Reporting_Airline                 OriginAirportID
##                                0                                0
##               OriginAirportSeqID              OriginCityMarketID
##                                0                                0
##                           Origin                  OriginCityName
##                                0                                0
##                      OriginState                 OriginStateFips
##                                0                                0
##                  OriginStateName                       OriginWac
##                                0                                0
##                    DestAirportID                 DestAirportSeqID
```

```
##                             0                             0
##                DestCityMarketID                          Dest
##                             0                             0
##                    DestCityName                     DestState
##                             0                             0
##                   DestStateFips                  DestStateName
##                             0                             0
##                         DestWac                   CRSDepTime
##                             0                             0
##                         DepTime                      DepDelay
##                             275                           275
##                DepDelayMinutes                      DepDel15
##                             275                           275
##           DepartureDelayGroups                      DepTimeBlk
##                             275                             0
##                         TaxiOut                     WheelsOff
##                             277                           277
##                        WheelsOn                        TaxiIn
##                             283                           283
##                      CRSArrTime                       ArrTime
##                             0                             283
##                        ArrDelay                  ArrDelayMinutes
##                             306                           306
##                        ArrDel15                ArrivalDelayGroups
##                             306                           306
##                      ArrTimeBlk                     Cancelled
##                             0                             0
##                  CancellationCode                      Diverted
##                             0                             0
##                 CRSElapsedTime                ActualElapsedTime
##                             0                             306
##                         AirTime                       Flights
##                             306                             0
##                        Distance                   DistanceGroup
##                             0                             0
##                     CarrierDelay                    WeatherDelay
##                            17059                         17059
##                        NASDelay                   SecurityDelay
##                            17059                         17059
##               LateAircraftDelay                    FirstDepTime
##                            17059                         20457
##                   TotalAddGTime                  LongestAddGTime
##                            20457                         20457
##              DivAirportLandings                   DivReachedDest
##                             0                            20550
##             DivActualElapsedTime                     DivArrDelay
##                            20554                         20554
##                      DivDistance
##                            20550
```

3. Use tapply to compute a matrix holding the distances between every pair of airports. You'll have to read the documentation for tapply to see how to deal with multiple factors. Print out the rows and columns for the 10 airports with the most flights

```
mat = matrix(data = 0, nrow = length(unique(dat$Origin)), ncol =
length(unique(dat$Origin)), dimnames =
list(unique(dat$Origin),unique(dat$Origin)))

mat1 = tapply(dat$Distance , list(dat$Origin, dat$Dest) , mean )

flights = dat %>% group_by(Flights, Origin) %>% mutate(TotalFlights = n())

flights = flights[!duplicated(flights[,15]),]

flights = head(flights[order(flights$TotalFlights, decreasing=TRUE),], 10)

flights[,c("Origin", "TotalFlights")]
```

```
## # A tibble: 10 × 2
## # Groups:   Origin [10]
##    Origin TotalFlights
##    <chr>         <int>
##  1 ATL            1013
##  2 ORD             993
##  3 DFW             826
##  4 DEN             753
##  5 CLT             657
##  6 LAX             631
##  7 SFO             492
##  8 IAH             491
##  9 PHX             474
## 10 LAS             467
```

```
indices = c('ATL', 'ORD','DFW','DEN','CLT','LAX','SFO','IAH','PHX','LAS')

mat1[indices,indices]
```

```
##      ATL  ORD  DFW  DEN  CLT  LAX  SFO  IAH  PHX  LAS
## ATL   NA  606  731 1199  226 1947 2139  689 1587 1747
## ORD  606   NA  801  888  599 1744 1846  925 1440 1514
## DFW  731  801   NA  641  936 1235 1464  224  868 1055
## DEN 1199  888  641   NA 1337  862  967  862  602  628
## CLT  226  599  936 1337   NA 2125 2296  912 1773 1916
## LAX 1947 1744 1235  862 2125   NA  337 1379  370  236
## SFO 2139 1846 1464  967 2296  337   NA 1635  651  414
## IAH  689  925  224  862  912 1379 1635   NA 1009 1222
## PHX 1587 1440  868  602 1773  370  651 1009   NA  255
## LAS 1747 1514 1055  628 1916  236  414 1222  255   NA
```