Lab 7

Jia Yu Lin

11:59PM April 22, 2021

#Rcpp

We will get some experience with speeding up R code using C++ via the `Rcpp` package.

First, clear the workspace and load the `Rcpp` package.

```
pacman::p_load(Rcpp)
```

Create a variable `n` to be 10 and a vaiable `Nvec` to be 100 initially. Create a random vector via `rnorm` `Nvec` times and load it into a `Nvec` x `n` dimensional matrix.

```
n <- 10
Nvec <- 100
X = matrix(data=rnorm(Nvec*n), nrow=Nvec)

head(X)
```

```
##             [,1]       [,2]       [,3]       [,4]       [,5]        [,6]
## [1,]  0.07011642 -0.7615992 -0.4075162  1.1010911  0.5946187 -0.14417587
## [2,]  0.57131881 -2.2347250 -0.1902574 -1.1799675 -1.7423134 -0.06285873
## [3,] -1.19266932  0.5169610  2.5136603  2.1952230 -1.3829669 -0.78323740
## [4,] -1.77388418 -0.7521812  1.2908121  0.5063234 -1.0328083  1.64529916
## [5,] -0.05192583  0.5186841 -1.7979108  0.1422486 -1.7246030  0.01476811
## [6,] -0.85060590 -1.0533052 -0.2810875 -1.2102295 -1.1815109 -0.46641048
##             [,7]       [,8]        [,9]       [,10]
## [1,] -1.07566187  0.2895300 -0.32454720 -0.87200519
## [2,] -0.06727345  0.9827661  0.79433797  0.05334742
## [3,]  2.03380800 -1.4137344  0.29070414  0.52567555
## [4,]  0.58010197 -1.8517176 -0.14704366  1.21674842
## [5,] -0.48148370 -0.6668875 -0.02057173  0.60918812
## [6,] -0.93718464  0.8273659  0.93629624 -0.83259206
```

Write a function `all_angles` that measures the angle between each of the pairs of vectors. You should measure the vector on a scale of 0 to 180 degrees with negative angles coerced to be positive.

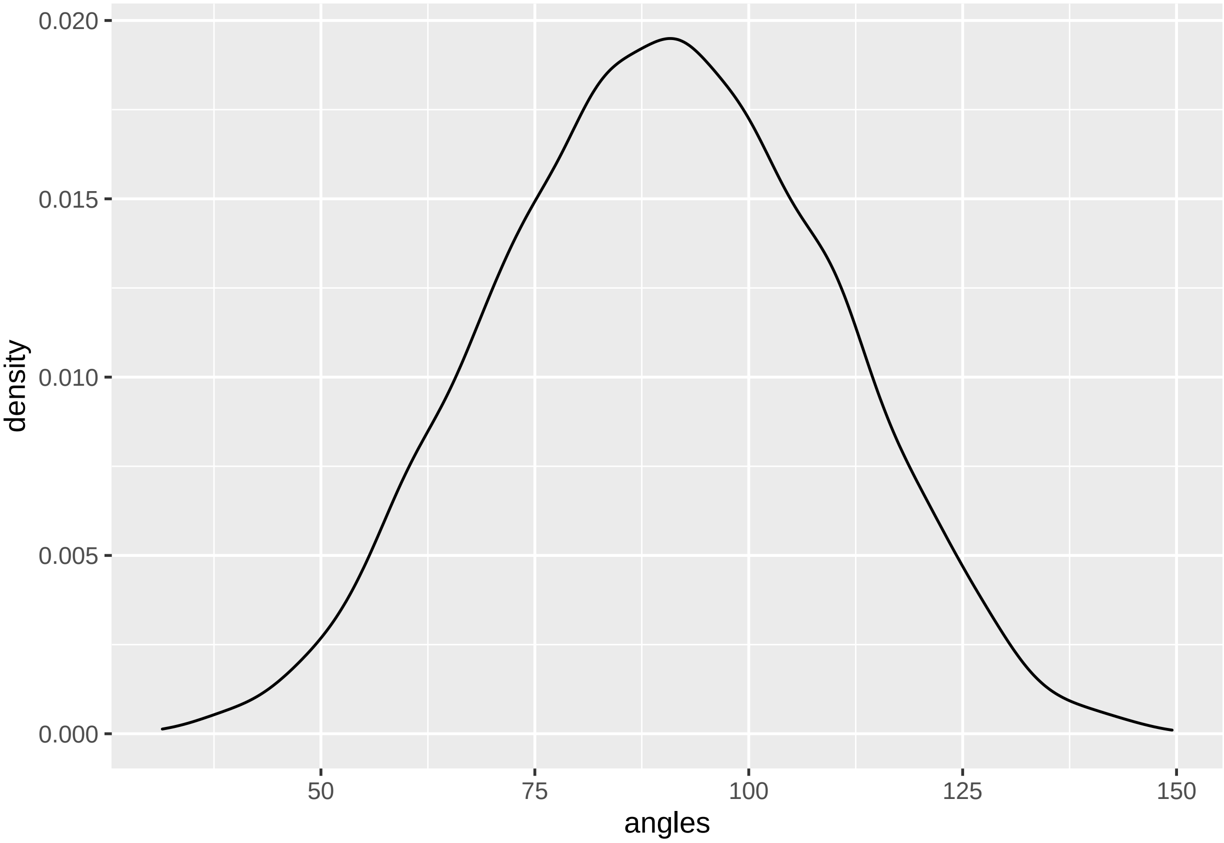
```
angle <- function(u,v){
  (acos(sum(u*v)/sqrt(sum(u^2)*sum(v^2)))) * (180/pi)
}

all_angles <- function(X){
  A <- matrix(NA, nrow=nrow(X), ncol=nrow(X))
  for( i in 1:nrow(X)-1){
    for(j in (i+1):nrow(X)){
      A[i,j] = angle(X[i,],X[j,])
    }

  }
  A
}

all_angles(X)
```

```
##       [,1]     [,2]     [,3]      [,4]     [,5]      [,6]     [,7]      [,8]
##  [1,]   NA 93.65858 107.3811 108.36645 97.30783  84.66151 84.42648  92.01642
##  [2,]   NA       NA 103.6140  91.72940 81.42575  43.19842 59.10322  57.51608
##  [3,]   NA       NA       NA  53.63657 95.47720 108.08359 92.49230  72.25870
##  [4,]   NA       NA       NA        NA 84.77435  97.62098 80.56597 100.76146
##  [5,]   NA       NA       NA        NA       NA  80.92600 65.92819  83.07987
##  [6,]   NA       NA       NA        NA       NA        NA 72.06045  68.18116
##  [7,]   NA       NA       NA        NA       NA        NA       NA  73.11877
##  [8,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [9,]   NA       NA       NA        NA       NA        NA       NA        NA
## [10,]   NA       NA       NA        NA       NA        NA       NA        NA
## [11,]   NA       NA       NA        NA       NA        NA       NA        NA
## [12,]   NA       NA       NA        NA       NA        NA       NA        NA
## [13,]   NA       NA       NA        NA       NA        NA       NA        NA
## [14,]   NA       NA       NA        NA       NA        NA       NA        NA
## [15,]   NA       NA       NA        NA       NA        NA       NA        NA
## [16,]   NA       NA       NA        NA       NA        NA       NA        NA
## [17,]   NA       NA       NA        NA       NA        NA       NA        NA
## [18,]   NA       NA       NA        NA       NA        NA       NA        NA
## [19,]   NA       NA       NA        NA       NA        NA       NA        NA
## [20,]   NA       NA       NA        NA       NA        NA       NA        NA
## [21,]   NA       NA       NA        NA       NA        NA       NA        NA
## [22,]   NA       NA       NA        NA       NA        NA       NA        NA
## [23,]   NA       NA       NA        NA       NA        NA       NA        NA
## [24,]   NA       NA       NA        NA       NA        NA       NA        NA
## [25,]   NA       NA       NA        NA       NA        NA       NA        NA
## [26,]   NA       NA       NA        NA       NA        NA       NA        NA
## [27,]   NA       NA       NA        NA       NA        NA       NA        NA
## [28,]   NA       NA       NA        NA       NA        NA       NA        NA
## [29,]   NA       NA       NA        NA       NA        NA       NA        NA
## [30,]   NA       NA       NA        NA       NA        NA       NA        NA
## [31,]   NA       NA       NA        NA       NA        NA       NA        NA
## [32,]   NA       NA       NA        NA       NA        NA       NA        NA
## [33,]   NA       NA       NA        NA       NA        NA       NA        NA
## [34,]   NA       NA       NA        NA       NA        NA       NA        NA
## [35,]   NA       NA       NA        NA       NA        NA       NA        NA
## [36,]   NA       NA       NA        NA       NA        NA       NA        NA
## [37,]   NA       NA       NA        NA       NA        NA       NA        NA
## [38,]   NA       NA       NA        NA       NA        NA       NA        NA
## [39,]   NA       NA       NA        NA       NA        NA       NA        NA
## [40,]   NA       NA       NA        NA       NA        NA       NA        NA
## [41,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [42,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [43,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [44,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [45,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [46,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [47,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [48,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [49,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [50,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [51,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [52,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [53,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [54,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [55,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [56,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [57,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [58,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [59,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [60,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [61,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [62,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [63,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [64,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [65,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [66,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [67,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [68,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [69,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [70,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [71,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [72,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [73,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [74,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [75,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [76,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [77,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [78,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [79,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [80,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [81,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [82,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [83,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [84,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [85,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [86,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [87,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [88,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [89,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [90,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [91,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [92,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [93,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [94,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [95,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [96,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [97,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [98,]   NA       NA       NA        NA       NA        NA       NA        NA
##  [99,]   NA       NA       NA        NA       NA        NA       NA        NA
## [100,]   NA       NA       NA        NA       NA        NA       NA        NA
##              [,9]     [,10]     [,11]     [,12]     [,13]     [,14]     [,15]
##   [1,]  89.48132  70.52331  76.02873  94.92291 127.09267  88.09188  49.96688
##   [2,] 102.18250  86.17797 112.08170  86.43145  60.40163  97.97960  73.65424
##   [3,]  83.24701  91.73882  71.23097  76.04145  73.49683 110.34944 100.99095
##   [4,]  79.84601 116.40945 108.12980  75.56546  89.35925 126.22409  97.90012
##   [5,]  59.11443 114.55487 115.41563  43.82456  83.31839 138.15229  86.41656
##   [6,] 109.39416  78.16935 119.12323  71.43661  71.53492  90.48187  68.66136
##   [7,]  64.55377  81.70064 115.22381  81.42565  78.25902 123.18962  61.26725
##   [8,]  90.01521  58.97682  81.55226  81.07649  62.44158  93.11711  79.24370
##   [9,]        NA 103.35838  82.61839  77.36775 105.31227 122.47914  71.97800
##  [10,]        NA        NA  83.46319 110.40966  91.27779  58.68947  75.68685
##  [11,]        NA        NA        NA 108.70414 100.46156  73.29460  88.68574
##  [12,]        NA        NA        NA        NA  70.72520 128.80252  75.52478
##  [13,]        NA        NA        NA        NA        NA  93.39751  95.51546
##  [14,]        NA        NA        NA        NA        NA        NA  98.32084
##  [15,]        NA        NA        NA        NA        NA        NA        NA
##  [16,]        NA        NA        NA        NA        NA        NA        NA
##  [17,]        NA        NA        NA        NA        NA        NA        NA
##  [18,]        NA        NA        NA        NA        NA        NA        NA
##  [19,]        NA        NA        NA        NA        NA        NA        NA
##  [20,]        NA        NA        NA        NA        NA        NA        NA
##  [21,]        NA        NA        NA        NA        NA        NA        NA
##  [22,]        NA        NA        NA        NA        NA        NA        NA
##  [23,]        NA        NA        NA        NA        NA        NA        NA
##  [24,]        NA        NA        NA        NA        NA        NA        NA
##  [25,]        NA        NA        NA        NA        NA        NA        NA
##  [26,]        NA        NA        NA        NA        NA        NA        NA
##  [27,]        NA        NA        NA        NA        NA        NA        NA
##  [28,]        NA        NA        NA        NA        NA        NA        NA
##  [29,]        NA        NA        NA        NA        NA        NA        NA
##  [30,]        NA        NA        NA        NA        NA        NA        NA
##  [31,]        NA        NA        NA        NA        NA        NA        NA
##  [32,]        NA        NA        NA        NA        NA        NA        NA
##  [33,]        NA        NA        NA        NA        NA        NA        NA
##  [34,]        NA        NA        NA        NA        NA        NA        NA
##  [35,]        NA        NA        NA        NA        NA        NA        NA
##  [36,]        NA        NA        NA        NA        NA        NA        NA
##  [37,]        NA        NA        NA        NA        NA        NA        NA
##  [38,]        NA        NA        NA        NA        NA        NA        NA
##  [39,]        NA        NA        NA        NA        NA        NA        NA
##  [40,]        NA        NA        NA        NA        NA        NA        NA
##  [41,]        NA        NA        NA        NA        NA        NA        NA
##  [42,]        NA        NA        NA        NA        NA        NA        NA
##  [43,]        NA        NA        NA        NA        NA        NA        NA
##  [44,]        NA        NA        NA        NA        NA        NA        NA
##  [45,]        NA        NA        NA        NA        NA        NA        NA
##  [46,]        NA        NA        NA        NA        NA        NA        NA
##  [47,]        NA        NA        NA        NA        NA        NA        NA
##  [48,]        NA        NA        NA        NA        NA        NA        NA
##  [49,]        NA        NA        NA        NA        NA        NA        NA
##  [50,]        NA        NA        NA        NA        NA        NA        NA
##  [51,]        NA        NA        NA        NA        NA        NA        NA
##  [52,]        NA        NA        NA        NA        NA        NA        NA
##  [53,]        NA        NA        NA        NA        NA        NA        NA
##  [54,]        NA        NA        NA        NA        NA        NA        NA
##  [55,]        NA        NA        NA        NA        NA        NA        NA
##  [56,]        NA        NA        NA        NA        NA        NA        NA
##  [57,]        NA        NA        NA        NA        NA        NA        NA
##  [58,]        NA        NA        NA        NA        NA        NA        NA
##  [59,]        NA        NA        NA        NA        NA        NA        NA
##  [60,]        NA        NA        NA        NA        NA        NA        NA
##  [61,]        NA        NA        NA        NA        NA        NA        NA
##  [62,]        NA        NA        NA        NA        NA        NA        NA
##  [63,]        NA        NA        NA        NA        NA        NA        NA
##  [64,]        NA        NA        NA        NA        NA        NA        NA
##  [65,]        NA        NA        NA        NA        NA        NA        NA
##  [66,]        NA        NA        NA        NA        NA        NA        NA
##  [67,]        NA        NA        NA        NA        NA        NA        NA
##  [68,]        NA        NA        NA        NA        NA        NA        NA
##  [69,]        NA        NA        NA        NA        NA        NA        NA
##  [70,]        NA        NA        NA        NA        NA        NA        NA
##  [71,]        NA        NA        NA        NA        NA        NA        NA
##  [72,]        NA        NA        NA        NA        NA        NA        NA
##  [73,]        NA        NA        NA        NA        NA        NA        NA
##  [74,]        NA        NA        NA        NA        NA        NA        NA
##  [75,]        NA        NA        NA        NA        NA        NA        NA
##  [76,]        NA        NA        NA        NA        NA        NA        NA
##  [77,]        NA        NA        NA        NA        NA        NA        NA
##  [78,]        NA        NA        NA        NA        NA        NA        NA
##  [79,]        NA        NA        NA        NA        NA        NA        NA
##  [80,]        NA        NA        NA        NA        NA        NA        NA
##  [81,]        NA        NA        NA        NA        NA        NA        NA
##  [82,]        NA        NA        NA        NA        NA        NA        NA
##  [83,]        NA        NA        NA        NA        NA        NA        NA
##  [84,]        NA        NA        NA        NA        NA        NA        NA
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,16]     [,17]     [,18]     [,19]     [,20]     [,21]     [,22]
##   [1,]  83.26991 127.55330  99.95639  79.43161  58.38297  99.90997 102.23447
##   [2,]  61.47423 110.26191 111.09815  49.43176  92.74155  66.94950  61.49380
##   [3,] 126.75671  86.32421  68.87026 115.72578  90.43109  92.96535  83.14114
##   [4,] 108.69652  87.71088  74.84367 104.01774 124.75252  93.11184  53.73929
##   [5,]  72.49043  90.01009 101.61309  69.95166  87.83180  87.69917  86.91363
##   [6,]  71.84892 103.96513 118.11844  44.03729 102.85865  61.63221  73.34281
##   [7,]  90.66373 102.39148 105.77402  61.49491  77.69984  93.02071  72.92370
##   [8,]  81.35989 115.35790 104.12075  75.13861  65.32258  87.82232  80.73651
##   [9,]  85.51129 107.67027  99.31281  98.18774  67.50141 129.36712  99.23060
##  [10,] 102.09967 102.49798  98.79438  83.44107  68.46139  98.98345  93.14370
##  [11,]  92.90592 104.91241  86.80733 122.03874  48.67365 109.98120 121.08667
##  [12,]  97.37288  92.00391  79.27808  62.38285  97.71203  64.60223  97.81829
##  [13,] 103.44502  71.16707  73.18413  63.97079  97.61166  45.65243  91.45240
##  [14,]  90.33031  82.94851  86.80311  96.53855  89.44777  88.33557 107.89575
##  [15,]  90.47748 135.70609  91.71431  52.18299  64.80277  87.75265 106.11694
##  [16,]        NA 113.97377 132.00470  82.07082  83.44443 101.07295  74.93429
##  [17,]        NA        NA  73.94848 103.04072 117.19787  71.72169  94.55515
##  [18,]        NA        NA        NA  89.47768 100.32970  66.65840 110.14962
##  [19,]        NA        NA        NA        NA  90.40050  50.52792  94.43137
##  [20,]        NA        NA        NA        NA        NA 109.73786 120.49426
##  [21,]        NA        NA        NA        NA        NA        NA  96.17535
##  [22,]        NA        NA        NA        NA        NA        NA        NA
##  [23,]        NA        NA        NA        NA        NA        NA        NA
##  [24,]        NA        NA        NA        NA        NA        NA        NA
##  [25,]        NA        NA        NA        NA        NA        NA        NA
##  [26,]        NA        NA        NA        NA        NA        NA        NA
##  [27,]        NA        NA        NA        NA        NA        NA        NA
##  [28,]        NA        NA        NA        NA        NA        NA        NA
##  [29,]        NA        NA        NA        NA        NA        NA        NA
##  [30,]        NA        NA        NA        NA        NA        NA        NA
##  [31,]        NA        NA        NA        NA        NA        NA        NA
##  [32,]        NA        NA        NA        NA        NA        NA        NA
##  [33,]        NA        NA        NA        NA        NA        NA        NA
##  [34,]        NA        NA        NA        NA        NA        NA        NA
##  [35,]        NA        NA        NA        NA        NA        NA        NA
##  [36,]        NA        NA        NA        NA        NA        NA        NA
##  [37,]        NA        NA        NA        NA        NA        NA        NA
##  [38,]        NA        NA        NA        NA        NA        NA        NA
##  [39,]        NA        NA        NA        NA        NA        NA        NA
##  [40,]        NA        NA        NA        NA        NA        NA        NA
##  [41,]        NA        NA        NA        NA        NA        NA        NA
##  [42,]        NA        NA        NA        NA        NA        NA        NA
##  [43,]        NA        NA        NA        NA        NA        NA        NA
##  [44,]        NA        NA        NA        NA        NA        NA        NA
##  [45,]        NA        NA        NA        NA        NA        NA        NA
##  [46,]        NA        NA        NA        NA        NA        NA        NA
##  [47,]        NA        NA        NA        NA        NA        NA        NA
##  [48,]        NA        NA        NA        NA        NA        NA        NA
##  [49,]        NA        NA        NA        NA        NA        NA        NA
##  [50,]        NA        NA        NA        NA        NA        NA        NA
##  [51,]        NA        NA        NA        NA        NA        NA        NA
##  [52,]        NA        NA        NA        NA        NA        NA        NA
##  [53,]        NA        NA        NA        NA        NA        NA        NA
##  [54,]        NA        NA        NA        NA        NA        NA        NA
##  [55,]        NA        NA        NA        NA        NA        NA        NA
##  [56,]        NA        NA        NA        NA        NA        NA        NA
##  [57,]        NA        NA        NA        NA        NA        NA        NA
##  [58,]        NA        NA        NA        NA        NA        NA        NA
##  [59,]        NA        NA        NA        NA        NA        NA        NA
##  [60,]        NA        NA        NA        NA        NA        NA        NA
##  [61,]        NA        NA        NA        NA        NA        NA        NA
##  [62,]        NA        NA        NA        NA        NA        NA        NA
##  [63,]        NA        NA        NA        NA        NA        NA        NA
##  [64,]        NA        NA        NA        NA        NA        NA        NA
##  [65,]        NA        NA        NA        NA        NA        NA        NA
##  [66,]        NA        NA        NA        NA        NA        NA        NA
##  [67,]        NA        NA        NA        NA        NA        NA        NA
##  [68,]        NA        NA        NA        NA        NA        NA        NA
##  [69,]        NA        NA        NA        NA        NA        NA        NA
##  [70,]        NA        NA        NA        NA        NA        NA        NA
##  [71,]        NA        NA        NA        NA        NA        NA        NA
##  [72,]        NA        NA        NA        NA        NA        NA        NA
##  [73,]        NA        NA        NA        NA        NA        NA        NA
##  [74,]        NA        NA        NA        NA        NA        NA        NA
##  [75,]        NA        NA        NA        NA        NA        NA        NA
##  [76,]        NA        NA        NA        NA        NA        NA        NA
##  [77,]        NA        NA        NA        NA        NA        NA        NA
##  [78,]        NA        NA        NA        NA        NA        NA        NA
##  [79,]        NA        NA        NA        NA        NA        NA        NA
##  [80,]        NA        NA        NA        NA        NA        NA        NA
##  [81,]        NA        NA        NA        NA        NA        NA        NA
##  [82,]        NA        NA        NA        NA        NA        NA        NA
##  [83,]        NA        NA        NA        NA        NA        NA        NA
##  [84,]        NA        NA        NA        NA        NA        NA        NA
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,23]     [,24]     [,25]     [,26]     [,27]     [,28]     [,29]
##   [1,]  77.95286  53.63374 100.73886  90.52462  77.23876  94.64575  54.57178
##   [2,] 113.29747  85.68240 104.51622  98.06135  62.90933  79.15888 100.78523
##   [3,]  60.52447 111.59563  86.66668  39.69180  88.78791  85.71894 119.41923
##   [4,]  61.64297 110.71779  79.27568  68.04097  91.03678  93.49675 118.14756
##   [5,] 107.97643 119.80229  98.23160 111.39275  67.75493 117.69642 111.99696
##   [6,] 122.20456  77.14014 103.42903  85.61377  90.79324  86.18361  80.81675
##   [7,]  83.31903  79.16695 122.28953  88.01865  63.86151  77.05977 105.82616
##   [8,] 112.31996  87.21606 119.14746  72.95090  62.11600  69.60658  93.99462
##   [9,]  82.27424 102.54919  90.28775  95.12413  69.93152  75.77364  96.72261
##  [10,]  99.85646  41.08606 130.57065  70.53257  94.18535  59.91391  56.85741
##  [11,]  71.63875  93.20295  75.87413  73.88137  86.17812  74.60511  80.67375
##  [12,] 103.56825 111.69944  78.19769  87.52292  74.43339 122.66429 115.53844
##  [13,] 105.18374 104.09222  86.27140  83.30975  76.65985  92.25550 134.11475
##  [14,] 107.18233  58.78930  85.76871  94.20440 114.66774  68.76483  54.24026
##  [15,]  91.20064  54.35941  88.74222  89.14190  59.70665  77.90534  77.58835
##  [16,] 121.90332 100.78296  94.81291 125.72487  75.91015  87.64390  72.79882
##  [17,]  83.55178 110.18978  86.71518  90.97265 125.67667 110.73618 112.12777
##  [18,]  74.69430  87.97447  58.34536  90.60392  79.83858 110.07821 116.67260
##  [19,] 119.73576  69.55082  93.60244 108.52094  63.71313 100.50779  96.08795
##  [20,]  83.92108  77.89927 105.63692  89.82491  60.71325  77.73040  79.94747
##  [21,] 102.71983  92.13823  73.43125  93.89389  85.54771 121.03678 116.31558
##  [22,]  87.50985  98.19403 114.04127  81.63069  92.27008  82.54890  94.96215
##  [23,]        NA  92.61264  86.02251  62.19402  95.66159  92.05713 104.11371
##  [24,]        NA        NA 107.00594  87.89787  89.54479  67.85139  51.60198
##  [25,]        NA        NA        NA 100.58720  90.37601 103.68173 104.21212
##  [26,]        NA        NA        NA        NA 109.31064  71.02828  92.33819
##  [27,]        NA        NA        NA        NA        NA  93.73927 110.64038
##  [28,]        NA        NA        NA        NA        NA        NA  69.92727
##  [29,]        NA        NA        NA        NA        NA        NA        NA
##  [30,]        NA        NA        NA        NA        NA        NA        NA
##  [31,]        NA        NA        NA        NA        NA        NA        NA
##  [32,]        NA        NA        NA        NA        NA        NA        NA
##  [33,]        NA        NA        NA        NA        NA        NA        NA
##  [34,]        NA        NA        NA        NA        NA        NA        NA
##  [35,]        NA        NA        NA        NA        NA        NA        NA
##  [36,]        NA        NA        NA        NA        NA        NA        NA
##  [37,]        NA        NA        NA        NA        NA        NA        NA
##  [38,]        NA        NA        NA        NA        NA        NA        NA
##  [39,]        NA        NA        NA        NA        NA        NA        NA
##  [40,]        NA        NA        NA        NA        NA        NA        NA
##  [41,]        NA        NA        NA        NA        NA        NA        NA
##  [42,]        NA        NA        NA        NA        NA        NA        NA
##  [43,]        NA        NA        NA        NA        NA        NA        NA
##  [44,]        NA        NA        NA        NA        NA        NA        NA
##  [45,]        NA        NA        NA        NA        NA        NA        NA
##  [46,]        NA        NA        NA        NA        NA        NA        NA
##  [47,]        NA        NA        NA        NA        NA        NA        NA
##  [48,]        NA        NA        NA        NA        NA        NA        NA
##  [49,]        NA        NA        NA        NA        NA        NA        NA
##  [50,]        NA        NA        NA        NA        NA        NA        NA
##  [51,]        NA        NA        NA        NA        NA        NA        NA
##  [52,]        NA        NA        NA        NA        NA        NA        NA
##  [53,]        NA        NA        NA        NA        NA        NA        NA
##  [54,]        NA        NA        NA        NA        NA        NA        NA
##  [55,]        NA        NA        NA        NA        NA        NA        NA
##  [56,]        NA        NA        NA        NA        NA        NA        NA
##  [57,]        NA        NA        NA        NA        NA        NA        NA
##  [58,]        NA        NA        NA        NA        NA        NA        NA
##  [59,]        NA        NA        NA        NA        NA        NA        NA
##  [60,]        NA        NA        NA        NA        NA        NA        NA
##  [61,]        NA        NA        NA        NA        NA        NA        NA
##  [62,]        NA        NA        NA        NA        NA        NA        NA
##  [63,]        NA        NA        NA        NA        NA        NA        NA
##  [64,]        NA        NA        NA        NA        NA        NA        NA
##  [65,]        NA        NA        NA        NA        NA        NA        NA
##  [66,]        NA        NA        NA        NA        NA        NA        NA
##  [67,]        NA        NA        NA        NA        NA        NA        NA
##  [68,]        NA        NA        NA        NA        NA        NA        NA
##  [69,]        NA        NA        NA        NA        NA        NA        NA
##  [70,]        NA        NA        NA        NA        NA        NA        NA
##  [71,]        NA        NA        NA        NA        NA        NA        NA
##  [72,]        NA        NA        NA        NA        NA        NA        NA
##  [73,]        NA        NA        NA        NA        NA        NA        NA
##  [74,]        NA        NA        NA        NA        NA        NA        NA
##  [75,]        NA        NA        NA        NA        NA        NA        NA
##  [76,]        NA        NA        NA        NA        NA        NA        NA
##  [77,]        NA        NA        NA        NA        NA        NA        NA
##  [78,]        NA        NA        NA        NA        NA        NA        NA
##  [79,]        NA        NA        NA        NA        NA        NA        NA
##  [80,]        NA        NA        NA        NA        NA        NA        NA
##  [81,]        NA        NA        NA        NA        NA        NA        NA
##  [82,]        NA        NA        NA        NA        NA        NA        NA
##  [83,]        NA        NA        NA        NA        NA        NA        NA
##  [84,]        NA        NA        NA        NA        NA        NA        NA
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,30]     [,31]     [,32]     [,33]     [,34]     [,35]     [,36]
##   [1,]  64.23768  61.55511  89.95374  69.90742  87.46733  83.20237  64.35921
##   [2,]  82.72041  85.45745 103.39424  80.59053  96.34648  95.13718 117.14473
##   [3,] 113.68731 126.62398  51.21630  79.20577  45.49378 103.30344  88.62928
##   [4,]  94.51768  88.57433  94.76103 115.41530  83.62946  92.01297 104.50199
##   [5,]  98.11041  75.66231 113.31866  98.64478  91.33683  97.40615 105.47800
##   [6,]  99.23096  65.84634  91.07203  90.40092 109.56512  94.19753 103.57418
##   [7,]  89.05813  89.41852 115.33592  75.76185  98.52564  77.49437 108.21464
##   [8,]  88.92115 112.94648  65.81778  59.33308  64.82605  86.79320  81.09065
##   [9,]  85.95967  85.29540  98.80484  97.58603  97.21350  80.84845  75.58986
##  [10,]  78.42625 101.96715  73.05401  61.03804  88.42054  54.77862  58.37036
##  [11,]  89.61789 114.57786  47.61152  66.95441  60.73092 108.94105  62.17800
##  [12,] 120.59066  75.81523  83.46415  88.59815  76.97248 105.19773 105.47532
##  [13,] 127.49250 123.03725  77.67949  66.12075  71.31549 106.13724 130.74700
##  [14,]  81.04619  96.26617  75.16796  84.64027 103.55168  73.51134  68.79888
##  [15,]  83.55481  63.98354  88.06901  66.02800  96.26638  77.57234  80.57578
##  [16,]  58.06031  65.17551 111.94260 108.92620 110.56425  97.34916  86.42044
##  [17,] 123.12924 113.11269  97.62346 100.18024  92.42798 100.57666 118.30839
##  [18,] 102.86989 105.75447  81.25689  73.72965  63.55258  84.77735 102.00021
##  [19,]  96.75628  68.07316 105.54873  70.84351 101.01580  84.26990 114.15034
##  [20,]  80.50640 105.14585  74.68539  45.99937  67.94498  92.55330  67.29955
##  [21,] 121.20928  91.47736  86.60984  72.09390  78.26222 112.39574 135.20495
##  [22,]  69.06679  84.63584 111.18395 116.62013  98.30271  82.76786 101.00921
##  [23,]  94.72393 105.61491  85.03746  82.42254  69.75841 103.13837  94.87095
##  [24,]  68.67090  75.46308  92.16678  69.34518 106.19741  48.82596  67.43769
##  [25,] 104.97899  80.17832  78.77434 102.39405  88.72297 112.24954 100.87667
##  [26,] 115.05188 112.49686  44.98185  79.12244  67.90198  98.97391  79.28054
##  [27,]  71.96502  92.01456  98.88982  60.30970  66.83102  87.63749  97.81509
##  [28,]  77.62304 102.34524  76.54567  89.42386 107.45533  66.20321  64.69073
##  [29,]  59.08160  61.09144  86.94931  99.80451 117.53759  63.74712  39.60364
##  [30,]        NA  70.11086 115.92115  99.52626 101.57645  58.68220  63.14819
##  [31,]        NA        NA 114.84852 119.19093 127.54925  79.64241  81.22994
##  [32,]        NA        NA        NA  68.12295  57.06282 107.80735  68.75420
##  [33,]        NA        NA        NA        NA  51.36241  94.36266  92.94225
##  [34,]        NA        NA        NA        NA        NA 112.62978  92.97229
##  [35,]        NA        NA        NA        NA        NA        NA  61.38309
##  [36,]        NA        NA        NA        NA        NA        NA        NA
##  [37,]        NA        NA        NA        NA        NA        NA        NA
##  [38,]        NA        NA        NA        NA        NA        NA        NA
##  [39,]        NA        NA        NA        NA        NA        NA        NA
##  [40,]        NA        NA        NA        NA        NA        NA        NA
##  [41,]        NA        NA        NA        NA        NA        NA        NA
##  [42,]        NA        NA        NA        NA        NA        NA        NA
##  [43,]        NA        NA        NA        NA        NA        NA        NA
##  [44,]        NA        NA        NA        NA        NA        NA        NA
##  [45,]        NA        NA        NA        NA        NA        NA        NA
##  [46,]        NA        NA        NA        NA        NA        NA        NA
##  [47,]        NA        NA        NA        NA        NA        NA        NA
##  [48,]        NA        NA        NA        NA        NA        NA        NA
##  [49,]        NA        NA        NA        NA        NA        NA        NA
##  [50,]        NA        NA        NA        NA        NA        NA        NA
##  [51,]        NA        NA        NA        NA        NA        NA        NA
##  [52,]        NA        NA        NA        NA        NA        NA        NA
##  [53,]        NA        NA        NA        NA        NA        NA        NA
##  [54,]        NA        NA        NA        NA        NA        NA        NA
##  [55,]        NA        NA        NA        NA        NA        NA        NA
##  [56,]        NA        NA        NA        NA        NA        NA        NA
##  [57,]        NA        NA        NA        NA        NA        NA        NA
##  [58,]        NA        NA        NA        NA        NA        NA        NA
##  [59,]        NA        NA        NA        NA        NA        NA        NA
##  [60,]        NA        NA        NA        NA        NA        NA        NA
##  [61,]        NA        NA        NA        NA        NA        NA        NA
##  [62,]        NA        NA        NA        NA        NA        NA        NA
##  [63,]        NA        NA        NA        NA        NA        NA        NA
##  [64,]        NA        NA        NA        NA        NA        NA        NA
##  [65,]        NA        NA        NA        NA        NA        NA        NA
##  [66,]        NA        NA        NA        NA        NA        NA        NA
##  [67,]        NA        NA        NA        NA        NA        NA        NA
##  [68,]        NA        NA        NA        NA        NA        NA        NA
##  [69,]        NA        NA        NA        NA        NA        NA        NA
##  [70,]        NA        NA        NA        NA        NA        NA        NA
##  [71,]        NA        NA        NA        NA        NA        NA        NA
##  [72,]        NA        NA        NA        NA        NA        NA        NA
##  [73,]        NA        NA        NA        NA        NA        NA        NA
##  [74,]        NA        NA        NA        NA        NA        NA        NA
##  [75,]        NA        NA        NA        NA        NA        NA        NA
##  [76,]        NA        NA        NA        NA        NA        NA        NA
##  [77,]        NA        NA        NA        NA        NA        NA        NA
##  [78,]        NA        NA        NA        NA        NA        NA        NA
##  [79,]        NA        NA        NA        NA        NA        NA        NA
##  [80,]        NA        NA        NA        NA        NA        NA        NA
##  [81,]        NA        NA        NA        NA        NA        NA        NA
##  [82,]        NA        NA        NA        NA        NA        NA        NA
##  [83,]        NA        NA        NA        NA        NA        NA        NA
##  [84,]        NA        NA        NA        NA        NA        NA        NA
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,37]     [,38]     [,39]     [,40]     [,41]     [,42]     [,43]
##   [1,]  73.66523  60.20443 116.55563 107.08235  80.50280  81.52119  87.49389
##   [2,]  73.43367 101.93388  57.03770  82.43877 100.62227 124.84090 112.72587
##   [3,]  75.92319 117.61461  96.06081  71.64094  59.73733  99.41028  69.48436
##   [4,]  96.96923 112.78645  73.62400  85.62520  47.68141 105.44421 100.15747
##   [5,]  81.15224 129.09545  71.09362  99.87190 119.92587 101.40364 116.15532
##   [6,]  58.22404  89.64973  64.06676 102.59743  99.41507  98.72991  91.82769
##   [7,]  79.02911 117.56314  88.12465  79.23632  92.33609 106.15672 126.95176
##   [8,]  54.10493 114.99123  89.08980  80.16906 101.59304 112.04875  91.71266
##   [9,]  77.44898 105.77537 101.98924  76.53707 104.17125  69.79558 111.34962
##  [10,]  77.90002  81.60810 123.74822  97.03449  85.40712  85.38859  85.12171
##  [11,]  72.78655  70.71536 113.40417  63.08508  89.89403  68.51483  53.43927
##  [12,]  64.72695 113.60829  81.26317 115.66148 104.32882 101.38633  94.32353
##  [13,]  82.46836 115.64462  74.71410  82.13223 106.91039 121.42958  91.47918
##  [14,] 101.55373  47.74701 108.32479  96.01633  95.62342  71.06283  70.26119
##  [15,]  56.61461  69.71795 114.29215 106.24535  95.40546  85.45236 105.59907
##  [16,]  83.58240  86.25507  56.45657  83.34482 117.02854  91.72038 104.09832
##  [17,] 125.25254 104.84706  80.35780  85.68868  91.78305  89.20504  77.75794
##  [18,] 113.74562  84.70480 115.56384 112.23200  79.31333 108.19900  93.85192
##  [19,]  74.56381  88.65009  85.36947 119.01910 114.72705 112.49979 116.41169
##  [20,]  66.17588  86.18787 120.66455  73.63072 109.50213  81.42011  85.41940
##  [21,]  89.48764  90.13471  74.17495 110.15108  98.01645 121.13883  85.38777
##  [22,]  98.72650 115.70504  52.51868  80.40152  59.76961 117.14938 108.51805
##  [23,]  99.94619  90.45726 103.71571  66.55066  45.11737  87.34263  77.05786
##  [24,]  88.78914  55.06741 128.58175 113.06736  80.84105  82.76358  99.26364
##  [25,]  94.27579  63.21713  87.14722  97.49004  93.83152  80.41669  78.55431
##  [26,]  61.44479  99.21382  98.77165  71.67350  51.42736  81.47903  53.82442
##  [27,]  79.01200 103.19238  95.44269  92.91065 106.66523 125.25978 125.69696
##  [28,]  74.40619  80.42758 101.71871  62.15922  85.04508  68.98934  90.72228
##  [29,]  82.57146  50.19906 108.91251 105.27781  87.96096  55.32174  79.26698
##  [30,] 106.79556  75.68679  95.52944  97.15398  82.24308  97.57600 118.71160
##  [31,]  84.28907  64.60855  82.17275 125.53960  92.36064  76.66281 105.45611
##  [32,]  53.27814  82.32963 106.56199  80.09064  81.78764  75.58260  39.04116
##  [33,]  73.42034  90.58329 120.44827  87.07371  98.09994 108.49712  84.42273
##  [34,]  80.23844 109.80852 100.90561  81.83035  78.63329 118.22551  72.15248
##  [35,] 106.76100  83.55513 121.26877 115.70125  86.04368  86.53083 123.95951
##  [36,]  75.72731  69.14861 124.56075  97.95762  84.83505  55.97969  77.75128
##  [37,]        NA  91.12420  89.91187  82.77336  98.18765  77.57363  69.67618
##  [38,]        NA        NA 111.34247 105.46443  87.45540  61.32826  74.48826
##  [39,]        NA        NA        NA  73.56389  93.76269 108.13934  92.00073
##  [40,]        NA        NA        NA        NA  85.37309  83.01576  74.76695
##  [41,]        NA        NA        NA        NA        NA  96.35416  78.35660
##  [42,]        NA        NA        NA        NA        NA        NA  63.61395
##  [43,]        NA        NA        NA        NA        NA        NA        NA
##  [44,]        NA        NA        NA        NA        NA        NA        NA
##  [45,]        NA        NA        NA        NA        NA        NA        NA
##  [46,]        NA        NA        NA        NA        NA        NA        NA
##  [47,]        NA        NA        NA        NA        NA        NA        NA
##  [48,]        NA        NA        NA        NA        NA        NA        NA
##  [49,]        NA        NA        NA        NA        NA        NA        NA
##  [50,]        NA        NA        NA        NA        NA        NA        NA
##  [51,]        NA        NA        NA        NA        NA        NA        NA
##  [52,]        NA        NA        NA        NA        NA        NA        NA
##  [53,]        NA        NA        NA        NA        NA        NA        NA
##  [54,]        NA        NA        NA        NA        NA        NA        NA
##  [55,]        NA        NA        NA        NA        NA        NA        NA
##  [56,]        NA        NA        NA        NA        NA        NA        NA
##  [57,]        NA        NA        NA        NA        NA        NA        NA
##  [58,]        NA        NA        NA        NA        NA        NA        NA
##  [59,]        NA        NA        NA        NA        NA        NA        NA
##  [60,]        NA        NA        NA        NA        NA        NA        NA
##  [61,]        NA        NA        NA        NA        NA        NA        NA
##  [62,]        NA        NA        NA        NA        NA        NA        NA
##  [63,]        NA        NA        NA        NA        NA        NA        NA
##  [64,]        NA        NA        NA        NA        NA        NA        NA
##  [65,]        NA        NA        NA        NA        NA        NA        NA
##  [66,]        NA        NA        NA        NA        NA        NA        NA
##  [67,]        NA        NA        NA        NA        NA        NA        NA
##  [68,]        NA        NA        NA        NA        NA        NA        NA
##  [69,]        NA        NA        NA        NA        NA        NA        NA
##  [70,]        NA        NA        NA        NA        NA        NA        NA
##  [71,]        NA        NA        NA        NA        NA        NA        NA
##  [72,]        NA        NA        NA        NA        NA        NA        NA
##  [73,]        NA        NA        NA        NA        NA        NA        NA
##  [74,]        NA        NA        NA        NA        NA        NA        NA
##  [75,]        NA        NA        NA        NA        NA        NA        NA
##  [76,]        NA        NA        NA        NA        NA        NA        NA
##  [77,]        NA        NA        NA        NA        NA        NA        NA
##  [78,]        NA        NA        NA        NA        NA        NA        NA
##  [79,]        NA        NA        NA        NA        NA        NA        NA
##  [80,]        NA        NA        NA        NA        NA        NA        NA
##  [81,]        NA        NA        NA        NA        NA        NA        NA
##  [82,]        NA        NA        NA        NA        NA        NA        NA
##  [83,]        NA        NA        NA        NA        NA        NA        NA
##  [84,]        NA        NA        NA        NA        NA        NA        NA
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,44]     [,45]     [,46]     [,47]     [,48]     [,49]     [,50]
##   [1,] 105.65559  74.34012  49.32716 104.46985  74.18430  81.12574 124.00943
##   [2,]  77.06381  53.82806  88.98885  81.94013 102.64388 139.81366  94.45314
##   [3,]  94.59014 123.13579  77.62143  74.37479 100.56976  85.15404  89.22135
##   [4,]  83.18753 120.95038  86.21333  69.83434  73.15273  90.22741 104.82703
##   [5,]  92.12665  66.38157  91.53565  59.32655 108.29566 117.70202  98.32797
##   [6,]  59.95141  64.51606  92.29105  65.18067  94.07935 127.57451 116.74742
##   [7,] 104.04694  65.49939  72.10867  65.80846 108.38305 108.68788  96.92159
##   [8,]  96.92148  71.33086  66.70625  83.36803 128.79060 121.78061  84.98615
##   [9,]  93.63700  92.43850  67.97219  91.10515 120.61489  83.44189  94.15479
##  [10,] 112.31038  84.10986  66.46704  93.37341  99.40296  81.11583  84.36657
##  [11,]  93.33077 105.34762  69.89729 129.43788 105.33114  73.56492  84.56472
##  [12,]  73.90907  74.31547  97.43206  40.44060 105.91587 103.99819 107.13210
##  [13,]  79.24709  69.91208 121.53944  60.16511 120.21285 110.03642  61.33777
##  [14,]  85.29788  93.95379 104.37392 120.42907  84.31648  67.41605  67.89223
##  [15,]  82.92641  59.86576  63.92547  84.32126 106.53445  84.06257 113.22115
##  [16,]  77.78921  68.95252  82.36949 116.77776  92.02783 129.12490 101.06660
##  [17,]  94.62524 105.58097 138.62964  71.06038  77.83989  80.24642  62.59938
##  [18,] 100.21432  91.72974 112.82153  79.31468  85.87970  52.56554  61.65070
##  [19,]  79.08343  31.45780 100.86920  60.56662 103.67750 108.49833  94.88896
##  [20,] 114.77775  69.77294  55.30137 111.46946 121.07470  87.93294  86.43174
##  [21,]  73.91148  62.64678 128.44444  58.27583  86.18894 102.92878  83.66097
##  [22,]  92.29297 102.68791  78.29888  83.51832  67.43450 122.77395 105.27039
##  [23,] 108.76595 119.78532  73.00404  94.50454  68.65768  67.89474 100.15125
##  [24,] 102.65150  75.09542  73.21531  97.67585  82.58456  65.81145  93.35520
##  [25,]  54.08473  97.79798 118.34962  98.84190  87.02976  69.24505  85.97643
##  [26,]  85.26320 125.69558  69.03695  74.79495  91.84231  85.86681 108.93471
##  [27,] 105.81661  48.34563  71.71544  91.62275 116.71387 105.11703  81.95744
##  [28,]  81.81112 103.70640  69.43494 112.63648 113.25746  84.58060  83.89936
##  [29,]  89.04100  95.57472  66.56388 118.39650  74.85554  76.93879 110.20814
##  [30,] 112.88599  83.43677  59.95138 129.09933  72.35354  90.54316  93.25381
##  [31,]  68.58125  78.25993  83.89269  88.17672  66.96615  91.88392 130.68246
##  [32,]  75.38575 110.52135  79.92756  92.07946 108.98036  81.19768  93.55182
##  [33,] 115.36947  60.33761  77.79719  86.47471 114.22610  86.75787  74.57191
##  [34,] 113.24187  91.14599  77.44415  85.65192  99.43017  92.65654  80.69308
##  [35,] 111.76329  87.21397  77.91354  92.36818  90.09151  65.79856  77.05055
##  [36,]  99.02192 108.88579  52.02650 117.89470  93.54225  67.48738 100.70133
##  [37,]  64.90924  82.57572  62.39398  79.57540 121.08252 110.60220 122.61984
##  [38,]  72.19861  90.26788  91.69590 122.88988  72.13999  58.81033  98.34795
##  [39,]  63.14258  89.56356 107.92540  82.20579  80.31519 142.71126 102.15615
##  [40,]  89.14062 109.09951  80.52061 110.40547 105.76677 105.83208  83.04289
##  [41,]  98.78129 129.64750  69.83873  89.26285  50.57869  76.55772 111.58851
##  [42,]  70.28393 116.14062  84.97192 106.98183  94.06557  62.67985 103.47097
##  [43,]  72.91935 120.28067  91.29010  97.43046  83.85811  83.42334 102.72738
##  [44,]        NA  96.65983 110.67839  85.05274  96.46515 100.24860 108.84468
##  [45,]        NA        NA  95.39345  80.78318 109.60206 110.55690  81.64802
##  [46,]        NA        NA        NA 105.18710  92.78006  94.66370 120.79761
##  [47,]        NA        NA        NA        NA  96.10195 100.77513  98.90455
##  [48,]        NA        NA        NA        NA        NA  81.05817 110.93251
##  [49,]        NA        NA        NA        NA        NA        NA  75.19403
##  [50,]        NA        NA        NA        NA        NA        NA        NA
##  [51,]        NA        NA        NA        NA        NA        NA        NA
##  [52,]        NA        NA        NA        NA        NA        NA        NA
##  [53,]        NA        NA        NA        NA        NA        NA        NA
##  [54,]        NA        NA        NA        NA        NA        NA        NA
##  [55,]        NA        NA        NA        NA        NA        NA        NA
##  [56,]        NA        NA        NA        NA        NA        NA        NA
##  [57,]        NA        NA        NA        NA        NA        NA        NA
##  [58,]        NA        NA        NA        NA        NA        NA        NA
##  [59,]        NA        NA        NA        NA        NA        NA        NA
##  [60,]        NA        NA        NA        NA        NA        NA        NA
##  [61,]        NA        NA        NA        NA        NA        NA        NA
##  [62,]        NA        NA        NA        NA        NA        NA        NA
##  [63,]        NA        NA        NA        NA        NA        NA        NA
##  [64,]        NA        NA        NA        NA        NA        NA        NA
##  [65,]        NA        NA        NA        NA        NA        NA        NA
##  [66,]        NA        NA        NA        NA        NA        NA        NA
##  [67,]        NA        NA        NA        NA        NA        NA        NA
##  [68,]        NA        NA        NA        NA        NA        NA        NA
##  [69,]        NA        NA        NA        NA        NA        NA        NA
##  [70,]        NA        NA        NA        NA        NA        NA        NA
##  [71,]        NA        NA        NA        NA        NA        NA        NA
##  [72,]        NA        NA        NA        NA        NA        NA        NA
##  [73,]        NA        NA        NA        NA        NA        NA        NA
##  [74,]        NA        NA        NA        NA        NA        NA        NA
##  [75,]        NA        NA        NA        NA        NA        NA        NA
##  [76,]        NA        NA        NA        NA        NA        NA        NA
##  [77,]        NA        NA        NA        NA        NA        NA        NA
##  [78,]        NA        NA        NA        NA        NA        NA        NA
##  [79,]        NA        NA        NA        NA        NA        NA        NA
##  [80,]        NA        NA        NA        NA        NA        NA        NA
##  [81,]        NA        NA        NA        NA        NA        NA        NA
##  [82,]        NA        NA        NA        NA        NA        NA        NA
##  [83,]        NA        NA        NA        NA        NA        NA        NA
##  [84,]        NA        NA        NA        NA        NA        NA        NA
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,51]     [,52]     [,53]     [,54]     [,55]     [,56]     [,57]
##   [1,]  83.84547  71.02282  80.31713  63.91642  73.77349  95.67086  74.82346
##   [2,] 104.14682 104.82454  64.89086  86.63038  63.50283  82.42119  92.91822
##   [3,]  76.20174 110.09013  57.48122  85.07831  97.46076  78.10145  67.83700
##   [4,]  83.63742 138.58967  75.10157  91.59741  93.18519  88.01459  79.55670
##   [5,]  87.53995 104.24467  92.43363  86.07185  87.45196 103.43640  75.12388
##   [6,] 120.80342  98.12339  68.05277 105.69638  72.65372  92.99257 113.88208
##   [7,]  83.60772 119.75456  62.00427  83.05402  78.11601 106.18647  80.55156
##   [8,]  91.78368  87.04630  56.22079  69.35437  59.06290  83.98827  86.14141
##   [9,]  82.11102 108.87083 102.52235  68.32750  84.96192  84.91605  73.77172
##  [10,]  88.41506  67.34280  72.27822  81.23797  71.99051 110.12569 111.78907
##  [11,]  86.93986  69.90363  91.38565  71.52701  87.89425  51.17715  68.25842
##  [12,]  80.15217  95.79804  74.98198  86.79216 106.45294 103.53880  72.69791
##  [13,]  80.89842  91.84055  63.47198 100.84109 109.84029  91.80853  84.88818
##  [14,]  98.49451  49.71864 110.58164  98.08549  93.20115  88.93180 124.45155
##  [15,]  77.03324  83.09355  72.66496  58.62412  86.07361  97.20391  76.61051
##  [16,] 121.16082  91.01548 112.01826  82.93462  48.23953  66.60425  97.94561
##  [17,]  89.70135  92.20984 100.63458 138.01709 128.29372 105.35920 102.92939
##  [18,]  33.96236  70.04818  87.87831  74.56875 138.01101 110.29943  66.70722
##  [19,]  81.80537  80.10215  73.11624  83.23609  95.12280 114.56519  90.06815
##  [20,]  77.10659  67.98365  80.80239  57.37650  76.76404  79.30190  61.74938
##  [21,]  80.72348  78.31753  68.50343 105.59849 119.69740 102.19456  84.07875
##  [22,] 109.72652 133.59580  74.93980  98.30237  54.00294  88.08372 103.20882
##  [23,]  74.46165 111.96098  71.41835  90.49859 102.89920  81.06748  59.84308
##  [24,]  79.23678  65.41425  83.32871  75.17155  84.73524 116.47652 107.23616
##  [25,]  80.45515  81.23690 111.49185  87.40561 124.87066  69.39807  76.87087
##  [26,]  99.33658 110.50679  48.87244 101.65437  87.94476  78.40126  90.88884
##  [27,]  56.73279  84.16508  74.48122  35.59919  78.93707  90.79589  47.43776
##  [28,] 106.67238 100.32876  89.16576  83.92451  65.81657  71.88418 114.51976
##  [29,] 112.92479  67.60209 110.18176  86.44261  63.51478  92.05610 123.43130
##  [30,]  86.73680  82.64991 107.60559  55.76986  49.09691  91.70511  92.41640
##  [31,] 104.47331  90.77225 109.09811  86.35636  79.77131  99.59129 103.36781
##  [32,]  93.69218  76.58030  69.34669  87.36060  95.00451  66.08192  85.37421
##  [33,]  59.09778  62.64206  53.90971  66.91956  98.32243  97.31390  59.70592
##  [34,]  60.48930  80.05084  56.25360  68.16144  96.13649  82.12891  43.54651
##  [35,]  73.03342  78.85153 103.10563  70.31937  80.05285 129.69424 115.42636
##  [36,]  94.91879  69.87571 105.83786  66.57793  65.55383  87.39050 104.04057
##  [37,] 109.62852  95.18572  61.83522  81.40466  71.89335  67.92149  84.95858
##  [38,]  94.08102  58.56677 112.00003  84.79456  97.73197  81.55062 102.65071
##  [39,] 127.23485 123.90683  89.28213 117.95178  73.55062  66.61892  98.50725
##  [40,] 110.75349 119.33945  80.97366 102.24454  75.87864  44.88816  81.62664
##  [41,]  87.49263 114.93876  66.15200  93.14788  84.90362  88.74890  86.84243
##  [42,] 114.78796  85.23288 119.84443 105.51463  94.91550  76.54037 113.96530
##  [43,] 113.10866  80.92961  82.12238 115.11175  98.65624  61.51862  94.45504
##  [44,] 123.84115 101.85145  98.81746 110.83441  96.31508  61.66727 110.16528
##  [45,]  74.42995  67.20401  82.68631  70.16084  88.40627 107.94941  76.79045
##  [46,]  91.41545  98.93435  69.62151  56.29964  46.57443  80.91148  75.15282
##  [47,]  81.93454 107.09835  61.30322 108.21447 111.74798 123.00806  89.79688
##  [48,]  96.16704  95.36188  96.11811 106.51592  90.15417  98.93304  97.62593
##  [49,]  59.71679  66.88440 109.85671  81.46786 125.15647 107.34675  90.57202
##  [50,]  62.57099  67.72496 105.57149  86.01249 111.85861 100.61586  90.16604
##  [51,]        NA  68.74967  82.69276  52.98730 118.90635 120.07100  52.82035
##  [52,]        NA        NA 106.48074  71.59821 103.90133 104.78151  89.88358
##  [53,]        NA        NA        NA  86.51871  83.65935  91.83381  69.27010
##  [54,]        NA        NA        NA        NA  74.70570  92.07973  57.38147
##  [55,]        NA        NA        NA        NA        NA  72.45053 100.45614
##  [56,]        NA        NA        NA        NA        NA        NA  80.82894
##  [57,]        NA        NA        NA        NA        NA        NA        NA
##  [58,]        NA        NA        NA        NA        NA        NA        NA
##  [59,]        NA        NA        NA        NA        NA        NA        NA
##  [60,]        NA        NA        NA        NA        NA        NA        NA
##  [61,]        NA        NA        NA        NA        NA        NA        NA
##  [62,]        NA        NA        NA        NA        NA        NA        NA
##  [63,]        NA        NA        NA        NA        NA        NA        NA
##  [64,]        NA        NA        NA        NA        NA        NA        NA
##  [65,]        NA        NA        NA        NA        NA        NA        NA
##  [66,]        NA        NA        NA        NA        NA        NA        NA
##  [67,]        NA        NA        NA        NA        NA        NA        NA
##  [68,]        NA        NA        NA        NA        NA        NA        NA
##  [69,]        NA        NA        NA        NA        NA        NA        NA
##  [70,]        NA        NA        NA        NA        NA        NA        NA
##  [71,]        NA        NA        NA        NA        NA        NA        NA
##  [72,]        NA        NA        NA        NA        NA        NA        NA
##  [73,]        NA        NA        NA        NA        NA        NA        NA
##  [74,]        NA        NA        NA        NA        NA        NA        NA
##  [75,]        NA        NA        NA        NA        NA        NA        NA
##  [76,]        NA        NA        NA        NA        NA        NA        NA
##  [77,]        NA        NA        NA        NA        NA        NA        NA
##  [78,]        NA        NA        NA        NA        NA        NA        NA
##  [79,]        NA        NA        NA        NA        NA        NA        NA
##  [80,]        NA        NA        NA        NA        NA        NA        NA
##  [81,]        NA        NA        NA        NA        NA        NA        NA
##  [82,]        NA        NA        NA        NA        NA        NA        NA
##  [83,]        NA        NA        NA        NA        NA        NA        NA
##  [84,]        NA        NA        NA        NA        NA        NA        NA
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,58]     [,59]     [,60]     [,61]     [,62]     [,63]     [,64]
##   [1,]  82.39178 104.61224  84.00134 139.26635  95.59526 106.69540  91.59968
##   [2,] 100.11379  71.70435 115.15938  89.98776  77.98940  73.36352  78.30900
##   [3,]  80.81242  77.34852  86.01360  93.64850 117.51788  76.41576  92.77087
##   [4,] 129.01893 104.06808  71.92897  68.56303  99.24395  95.85259  55.69600
##   [5,] 101.38199  79.88510  87.84890  70.72398 112.70375  56.94208 110.52401
##   [6,] 102.61471  94.34422 108.13655  81.26430  92.31477  78.52570  76.55013
##   [7,]  97.98973  57.22515  82.90784  95.12722  81.78422  79.09060  85.24090
##   [8,]  69.33602  48.58368 132.69516 110.65215 108.69376  55.31087 107.73335
##   [9,]  90.13235  68.90291  67.20844  87.17251  94.44670  94.83237  98.91539
##  [10,]  69.08077  70.80948 117.02429 122.00284  89.38852  87.65444  94.00601
##  [11,]  47.15923  77.92734  89.39594 125.90185  97.99548 103.10679 116.25246
##  [12,]  91.64106  96.36360  77.05067  69.37834 115.65226  50.50535  95.58844
##  [13,]  70.73514  61.28520  98.49949  72.03183  80.88191  44.44416  95.52969
##  [14,]  70.54066  97.06410 106.64695  97.77735  64.82994 107.95754  86.13540
##  [15,]  79.19164  87.28132  70.19490 111.99425  75.83768  92.47277  75.24499
##  [16,] 105.90541  90.25200 118.66335  90.87030  92.25502  96.65508  98.59008
##  [17,]  90.13342  92.53774  81.48179  57.93384  86.39363  80.95781 100.64824
##  [18,]  75.24651  99.64810  64.31988  83.76514  72.69654  73.99327  77.18364
##  [19,]  87.70556  86.43037  89.14834  84.77996  72.20001  61.87185  79.70606
##  [20,]  43.54110  54.66454  94.16942 141.27990  96.21137  84.15322 129.59222
##  [21,]  81.97838  99.89190  87.39928  75.21257  80.04064  57.20459  84.14534
##  [22,] 142.04991  93.47125 114.90812  80.47926 100.29822  95.69215  64.56626
##  [23,]  90.33808  93.47415  57.21376 109.43928  92.70011 113.18401  84.70867
##  [24,]  81.84619  93.92929  89.73334 117.06653  63.41131 105.92831  68.56708
##  [25,]  86.26748 118.53259  56.45255  68.19704  72.08227  99.77594  73.68064
##  [26,]  82.75767  84.59362  89.16832 102.19536 115.44230  94.32982  85.82583
##  [27,]  77.09039  64.16494  92.89045 110.53403  83.46009  62.41196  95.94551
##  [28,]  84.71982  64.08318 100.68857  99.19085  71.92460 114.96073  75.88773
##  [29,]  93.20922 109.49459 104.96739 110.62802  90.64207 124.05757  84.89803
##  [30,] 109.61277  95.83074 110.56808 114.08068  81.07148 111.99623  77.53410
##  [31,] 123.72811 133.17279  77.66207  80.22864  87.24096 110.30600  64.19566
##  [32,]  54.13181  84.56857  96.24630 106.08463 112.02672  83.83036 102.99168
##  [33,]  36.13376  57.34589  93.50125 132.23562  86.16172  62.03848 113.80638
##  [34,]  59.31686  73.76924  97.84219 118.00549 116.57717  58.40048 115.68122
##  [35,] 100.97365  86.71911  96.70323  91.76062  70.99519  97.16337  67.54111
##  [36,]  81.35036  94.35416 102.72834 115.54648 105.62618 111.62951  95.17576
##  [37,]  72.68867  78.05552  95.99458 105.24211 115.01103  81.01869 100.27900
##  [38,]  81.10972 121.97899  75.07105 103.31256  61.36921 128.57646  70.61325
##  [39,] 123.57689  96.63572 107.55943  60.61708 100.17583  85.39190  85.93983
##  [40,]  80.76398  56.06927  94.59873  98.66369  91.84863 104.07405 106.99189
##  [41,] 114.05914 111.63392  81.93629 101.68723  98.88488 115.29001  58.53041
##  [42,]  82.61823 101.29762  70.73248  86.27714  90.56128 128.03865  94.84061
##  [43,]  67.08198 101.94863  91.92503 100.18272 114.47786 100.52691 108.44343
##  [44,] 100.72197 111.12395  79.78319  56.98939  83.58192 101.78537  68.70064
##  [45,]  74.49389  73.49431  97.78130  99.70184  72.78172  58.23094 100.81350
##  [46,]  89.99349  79.29590 100.48040 137.19011 114.32518 103.46561  94.46631
##  [47,]  98.80574  89.34696  78.81831  64.27027 104.91975  53.27746  85.57139
##  [48,] 125.60074 140.65930  84.69162  90.27500  91.56874 118.31152  66.13204
##  [49,]  75.46278 105.50618  54.20655  93.60967  68.46962 112.35528  77.02875
##  [50,]  64.88634  60.72052  97.56276  80.83200  61.75650  70.23348 102.49725
##  [51,]  70.31972  81.51598  69.45879 102.34508  75.22965  67.34472  89.00609
##  [52,]  50.84168  94.49947  99.00493 109.79852  79.32971  78.36737 108.27612
##  [53,]  80.49450  71.85068  96.50291 113.04232 107.04510  67.05553  89.66910
##  [54,]  76.43463  78.52048  92.32822 122.18331  86.75321  81.67091  89.13048
##  [55,] 108.02333  79.14515 133.79167 112.92307 109.35524 100.35903  92.11862
##  [56,]  83.67193  85.59170  95.23595  97.02813  97.15461 110.75766  97.84333
##  [57,]  70.56887  78.56041  70.71759 115.01920  99.31760  72.08509 107.72774
##  [58,]        NA  59.92157  89.72858 118.15084  87.00078  70.44619 127.83189
##  [59,]        NA        NA 106.52024 104.34585  87.64759  64.99397 119.40671
##  [60,]        NA        NA        NA  77.90041  70.73319 107.17890  73.78851
##  [61,]        NA        NA        NA        NA  81.06588  84.32955  71.48172
##  [62,]        NA        NA        NA        NA        NA 106.35232  61.34907
##  [63,]        NA        NA        NA        NA        NA        NA 115.20091
##  [64,]        NA        NA        NA        NA        NA        NA        NA
##  [65,]        NA        NA        NA        NA        NA        NA        NA
##  [66,]        NA        NA        NA        NA        NA        NA        NA
##  [67,]        NA        NA        NA        NA        NA        NA        NA
##  [68,]        NA        NA        NA        NA        NA        NA        NA
##  [69,]        NA        NA        NA        NA        NA        NA        NA
##  [70,]        NA        NA        NA        NA        NA        NA        NA
##  [71,]        NA        NA        NA        NA        NA        NA        NA
##  [72,]        NA        NA        NA        NA        NA        NA        NA
##  [73,]        NA        NA        NA        NA        NA        NA        NA
##  [74,]        NA        NA        NA        NA        NA        NA        NA
##  [75,]        NA        NA        NA        NA        NA        NA        NA
##  [76,]        NA        NA        NA        NA        NA        NA        NA
##  [77,]        NA        NA        NA        NA        NA        NA        NA
##  [78,]        NA        NA        NA        NA        NA        NA        NA
##  [79,]        NA        NA        NA        NA        NA        NA        NA
##  [80,]        NA        NA        NA        NA        NA        NA        NA
##  [81,]        NA        NA        NA        NA        NA        NA        NA
##  [82,]        NA        NA        NA        NA        NA        NA        NA
##  [83,]        NA        NA        NA        NA        NA        NA        NA
##  [84,]        NA        NA        NA        NA        NA        NA        NA
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,65]     [,66]     [,67]     [,68]     [,69]     [,70]     [,71]
##   [1,]  65.55513  73.82191 119.24927  74.91407  86.17527  48.64634  71.70862
##   [2,]  96.90889 127.93635 109.25737 106.50572  94.92558  68.74103  99.47971
##   [3,] 109.32019  99.81865  67.50358  96.34229  88.45031 120.71068  93.13409
##   [4,]  86.60477  95.39516  50.86946  86.22881  99.56371 110.07035  66.28617
##   [5,]  66.85279 108.89900  93.03704 116.50942  70.08520  81.37050 117.14489
##   [6,]  90.03467  99.45435 125.17532 100.98457  57.47099  78.28560 101.54174
##   [7,] 100.19901 101.70791  91.56591 101.76499  88.03048  77.54035  95.29547
##   [8,] 110.92921 115.06081 103.40034  97.17896  79.76032  73.56177 119.22363
##   [9,]  87.22755  81.36252  82.47084  94.83007  85.05413  81.93753 110.79298
##  [10,] 115.73220  69.58808  97.75790  61.33058  79.13110  74.81500  87.21434
##  [11,]  96.95697  85.71986 104.59139  94.86759  99.13630  90.74862 104.89392
##  [12,]  81.79904 109.93638 101.26579 109.63804  53.54574  93.60504 108.94496
##  [13,] 129.64051 134.90277  94.25765 117.10114  86.37963 111.72469 113.36498
##  [14,] 110.32567  68.16481  98.31235  63.12773  95.55487  89.17557  82.78714
##  [15,]  96.45445  88.87186 121.85963  80.70510  76.91971  54.69418  88.18013
##  [16,]  57.06860 101.02320 110.21723 101.52701  96.19953  54.69262 107.12716
##  [17,]  99.32646  87.36368  68.66468 104.82414  89.08364 142.03080  90.37927
##  [18,] 111.46030 103.18917  66.38772  73.42953 108.88960 103.93010  65.76920
##  [19,]  97.58038 113.81382 120.05918  96.20812  73.84983  65.82054  95.29281
##  [20,]  94.09356  93.12833 116.91407  98.96547  91.55843  64.12618 113.59197
##  [21,] 100.90279 125.38754 108.13193 108.99278  82.42550 101.46985  88.58306
##  [22,]  76.81166  97.86554  64.01083  85.74525 103.61054  87.67740  71.01067
##  [23,]  87.98471  81.94237  72.28555  93.48470 110.68628 110.07487  63.82819
##  [24,] 106.39554  64.91378 101.51206  48.15270  89.64420  62.37397  63.17272
##  [25,]  90.22608  98.27022  92.90981  93.13984  97.83423 102.95868  85.87631
##  [26,] 110.29019  77.89337  87.63738  91.77603  70.19433 118.81368  90.29455
##  [27,]  91.97092 132.37856  97.01526  95.42989 110.06061  50.76490  96.80824
##  [28,] 123.69518  70.89321  87.93545  72.65969  93.68450  86.25158  98.21243
##  [29,]  75.36508  42.52667 106.87003  55.48526  77.02307  62.83191  79.75620
##  [30,]  67.23538  81.42680  78.31704  55.10961 122.97944  39.67362  63.66685
##  [31,]  52.05209  71.53300 106.65072  74.62289  73.73139  60.12189  73.69463
##  [32,] 112.81375  87.46495 105.88042  93.43530  69.19346 107.90886 108.63924
##  [33,] 116.69421 112.38436 112.54791  98.19814  93.42300  78.96208  98.88156
##  [34,]  96.73733 121.19111  87.12427 103.51631 100.25721  96.07410  94.55458
##  [35,] 104.64101  62.55024  65.20888  32.09554  95.10658  67.39640  67.46497
##  [36,]  84.96643  48.03232  90.53979  52.41542  79.26119  68.86419  89.12193
##  [37,]  94.76074  93.71038 132.36535 107.79577  49.84856  80.86774 124.08486
##  [38,]  87.91849  63.93530 112.71064  64.16009  94.32978  74.69607  69.31716
##  [39,]  68.63045 116.12185  92.19662 124.27038  90.69357 100.22789 102.15040
##  [40,] 101.25244  98.15658  88.35174 121.89553 106.22240 110.59251 112.41683
##  [41,]  86.71490  75.26432  63.28806  69.47829 104.01346 103.37548  43.68068
##  [42,]  88.03011  35.88700 102.83167  83.02814  64.22769 101.14907 103.14589
##  [43,]  91.17522  75.51859 110.08227 104.70246  68.71828 120.52394 102.48830
##  [44,]  92.68837  89.94799 112.64627 103.40783  65.77691 103.62421 106.50358
##  [45,]  90.52722 122.16837 121.21936 101.23768  89.81862  52.54514 102.46205
##  [46,]  76.41614  76.92002  95.87929  76.98978  89.23372  58.14530  85.85397
##  [47,]  98.80316 101.43804  87.61641 104.30682  55.59138 110.41143  95.80717
##  [48,]  56.29509  72.62284  75.12757  72.52743 104.84675  92.50602  36.81556
##  [49,] 106.59521  54.90019  72.10869  53.30449  97.44040 100.16927  64.02311
##  [50,] 126.11590 106.02786  67.03953  85.65454 117.75965 101.86714  97.18316
##  [51,] 105.99239 106.24196  69.78826  72.68590 113.47964  81.52807  70.10196
##  [52,]  96.27718  86.54292 109.10061  70.96335  89.92836  70.31586  90.57969
##  [53,] 105.91715 113.61274 100.28521 105.26010  81.67050  94.17428  88.98507
##  [54,]  89.70564 103.41913  88.64276  67.34973 110.75504  43.24975  82.83647
##  [55,]  70.21202  88.00624  95.94360  84.03118  93.50695  55.33455  95.22170
##  [56,]  84.49207  99.92125 106.42271 116.19381  98.89994  96.92699 111.45928
##  [57,]  81.18614 124.13376  92.72653 110.95317 110.25667  81.67183  89.96247
##  [58,] 119.25513  99.22507 115.59894 101.71770  85.17672  92.06333 117.32383
##  [59,] 122.85534 110.86619  90.60390 109.28973  96.14363  89.31351 125.23085
##  [60,]  92.29083  78.33164  87.29598  92.70945  88.94258 106.60382  76.54290
##  [61,]  93.86521  91.26042  71.13219  97.58051  77.30422 120.36979  97.40363
##  [62,] 117.13874  91.53762  87.07159  74.21226 118.56711  84.00557  72.59919
##  [63,] 102.55598 132.33091  93.69326 109.99857  77.41596  90.55101 114.66514
##  [64,]  97.58133  80.30213  72.12761  57.31215 100.92635  87.32053  47.92245
##  [65,]        NA  84.40035  96.48681  96.85473  88.70831  70.31428  80.49030
##  [66,]        NA        NA  83.88826  57.59073  71.55760  94.99463  76.10020
##  [67,]        NA        NA        NA  65.92387 118.15528 108.21330  63.10303
##  [68,]        NA        NA        NA        NA 100.82696  69.71386  49.94801
##  [69,]        NA        NA        NA        NA        NA 100.12578 118.64790
##  [70,]        NA        NA        NA        NA        NA        NA  82.53261
##  [71,]        NA        NA        NA        NA        NA        NA        NA
##  [72,]        NA        NA        NA        NA        NA        NA        NA
##  [73,]        NA        NA        NA        NA        NA        NA        NA
##  [74,]        NA        NA        NA        NA        NA        NA        NA
##  [75,]        NA        NA        NA        NA        NA        NA        NA
##  [76,]        NA        NA        NA        NA        NA        NA        NA
##  [77,]        NA        NA        NA        NA        NA        NA        NA
##  [78,]        NA        NA        NA        NA        NA        NA        NA
##  [79,]        NA        NA        NA        NA        NA        NA        NA
##  [80,]        NA        NA        NA        NA        NA        NA        NA
##  [81,]        NA        NA        NA        NA        NA        NA        NA
##  [82,]        NA        NA        NA        NA        NA        NA        NA
##  [83,]        NA        NA        NA        NA        NA        NA        NA
##  [84,]        NA        NA        NA        NA        NA        NA        NA
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,72]     [,73]     [,74]     [,75]     [,76]     [,77]     [,78]
##   [1,]  95.56091  89.22288  75.88155  78.50433  64.38150  71.74554  91.01204
##   [2,]  89.34858  87.62318  94.97257  94.44387 105.64566  95.80720 119.16297
##   [3,]  85.42378  93.95515  68.87566  66.96676 105.93392  76.49173  95.41299
##   [4,]  54.73548  62.07815  97.99041  72.91110 118.53026  86.16144 102.95148
##   [5,]  53.11570  99.88848 110.61419  55.94699 100.10129  96.08019 114.75934
##   [6,]  86.54398 101.47998  86.69812  79.85322  88.58529  93.73893 113.67207
##   [7,]  89.52165  88.78028 118.53107  82.98801  97.45012  79.78137 133.72370
##   [8,] 107.72035 113.28357  69.13062  74.26902 109.67776  86.67875 101.68603
##   [9,]  73.54677  86.42822 107.71562  65.73385  95.62742 111.58210 116.21183
##  [10,] 141.00008 108.52678  77.34071  99.99890  86.74970  70.06815  75.19132
##  [11,] 112.20764 103.79956  50.55076  86.71965  75.65142  94.19602  79.21610
##  [12,]  44.50046  91.59265  98.42823  51.49684  81.85187  97.90218 121.78598
##  [13,]  92.34737  95.05549  99.68043 102.14218  89.78684  95.42872 113.68915
##  [14,] 132.81906  93.25663  78.23228 131.64553  73.90912  98.01041  53.91272
##  [15,]  83.82530  73.60223 100.30206  81.72940  60.57116 101.78358 131.22368
##  [16,]  82.95226  97.38416  82.39299  83.34747 108.83037 110.11437  88.05343
##  [17,]  96.00687 102.08209 104.72661 109.88590  87.00229  75.74498  71.44728
##  [18,]  77.50271  54.40798 108.49944 111.42692  71.20903  91.17334  87.52056
##  [19,]  78.51190  80.51751 116.12814  95.20668  69.54372  98.62766 126.34609
##  [20,] 114.95406 111.13186  74.54984  81.64515  72.64087  84.46176  98.40184
##  [21,]  78.18543  84.28621  98.28560 103.41326  67.91685  86.17776 105.30207
##  [22,]  81.26234  83.49611  88.11725  81.68424 146.45940  74.23494  86.69637
##  [23,]  91.01460  80.43947  83.89444  86.94523  84.68042  60.52779  92.84311
##  [24,] 119.44461  77.52897  97.74779 113.96824  67.16540  81.53835  86.15998
##  [25,]  59.42738  58.07254  94.62731  98.93240  66.52592 127.96082  97.77760
##  [26,] 104.04456 107.81426  57.27392  67.92318  94.72496  67.48340  94.56145
##  [27,]  75.33840  71.20119 102.88457  85.10734  92.50894  99.32446 120.20041
##  [28,] 123.93800  91.35230  83.72304 101.42895 103.17315 107.39428  94.42846
##  [29,] 113.72150  99.47547  71.49210  91.43722  82.16351  90.81991  62.12944
##  [30,]  95.52087  70.44934  90.34106  99.13910 111.95289  89.61972  68.66273
##  [31,]  58.67571  71.28572 101.43100  76.84668  79.76009 105.54557  97.61280
##  [32,] 105.37647 110.45979  44.12644  74.24516  78.40666  91.24958  86.90149
##  [33,] 116.54920 100.82095  82.44535  98.14657  63.72193  70.54386 103.59633
##  [34,]  91.08322  98.87365  63.42684  75.59889  91.09781  65.97390  88.41002
##  [35,] 104.25462  70.59253 116.14898 107.60260  99.39173  92.87770  76.68885
##  [36,] 108.51580  99.17743  65.00770  76.38280  92.35301  94.50187  63.66842
##  [37,]  88.51835 113.85913  60.53205  49.97085  81.36779  99.53580 121.67385
##  [38,] 101.16861  71.63268  82.93798 114.41740  52.34204 106.99893  77.57881
##  [39,]  67.74079  97.31717  85.31746  79.56568 120.33179  95.96260  97.57576
##  [40,] 111.13045 110.56975  72.07848  89.32419 108.09110  87.44191  97.22704
##  [41,]  89.54430  73.85243  74.48887  85.22560 104.58181  58.83504  78.81802
##  [42,] 101.73146 105.06700  80.34375  83.26071  69.55735 108.47584  81.77922
##  [43,] 107.19605 122.67630  38.37672  78.17032  73.36726  79.11920  72.31166
##  [44,]  69.13021  84.81910  83.83377  82.81614  81.31813 131.90081 110.28087
##  [45,]  87.26029  86.96147 112.11105  99.99443  71.49826  96.42048 114.89191
##  [46,]  98.74481  98.17755  64.72247  58.73636 103.08041  72.88213  93.77313
##  [47,]  67.57990  95.25619 109.77980  70.49531  86.82569  76.95858 117.09244
##  [48,]  81.42112  74.73276  85.25282  97.55381  92.87463  62.35907  60.52480
##  [49,]  98.78741  66.57103 103.16895 111.95148  60.36269  94.25854  74.01896
##  [50,] 111.64922  83.78721 111.58756 133.61034  91.95384  99.74089  75.53413
##  [51,]  82.46150  59.17255 115.89897 104.81437  74.64042  83.31478  95.27797
##  [52,] 109.58062  92.27662  84.78094 112.07898  56.11380  94.84197  65.86210
##  [53,]  96.07536 100.55275  74.14182  73.00749  92.06364  56.38738 115.69323
##  [54,]  82.72965  65.25195  91.56898  84.74966  90.95651 100.30565  98.27786
##  [55,] 100.86356 105.02862  64.79842  70.42814 129.07383  84.46144  82.91151
##  [56,]  91.18144  99.96961  53.08804  79.25084  98.73989 109.00592  95.02530
##  [57,]  70.44440  79.47712  87.88367  75.59771  79.84348  79.17307 111.28692
##  [58,] 120.27505 110.79806  75.80299  99.21327  55.82335  90.40360  92.65019
##  [59,] 118.07212 111.83882  94.21979  93.13447 102.16037  87.90287 107.42086
##  [60,]  66.11286  62.56224 117.21636  92.12399  54.58369 100.34278 115.92361
##  [61,]  60.54696  78.46806 119.15695  93.91452 100.42447 118.31185  96.74609
##  [62,] 100.87644  52.46281 128.45183 149.48335  70.77333 113.87495  99.40818
##  [63,]  80.93908 103.32695  97.35993  79.78603  92.84385  83.35913 103.72286
##  [64,]  73.22503  39.68046 109.56857 105.65452  95.55216 103.77022  98.99661
##  [65,]  61.39387  94.05601  78.85873  62.19883  98.17353  79.54218  77.38125
##  [66,] 106.50737  96.49123  85.85039  88.58261  81.19078  87.39445  65.80324
##  [67,]  83.16363  66.80030 109.37548 100.65731 125.70150  82.31539  69.19958
##  [68,]  99.28647  60.30083  98.36977 107.86902  93.17177  92.73171  63.50508
##  [69,]  84.15694 122.37735  79.50885  54.92059  74.76459  93.02418 103.29597
##  [70,]  87.94620  77.05787  92.54312  87.41072  90.16377  96.73668  93.91056
##  [71,]  84.84112  48.93798 101.21308 110.69768  90.14027  69.28958  71.92409
##  [72,]        NA  62.86691 106.66242  62.92351  94.39398 108.30836 113.03281
##  [73,]        NA        NA 122.75231 112.79741  86.99867 110.89738 100.35614
##  [74,]        NA        NA        NA  65.57422  96.82805  75.11532  69.75123
##  [75,]        NA        NA        NA        NA 102.87025  83.35855 105.33578
##  [76,]        NA        NA        NA        NA        NA  95.02959 102.71837
##  [77,]        NA        NA        NA        NA        NA        NA  73.23997
##  [78,]        NA        NA        NA        NA        NA        NA        NA
##  [79,]        NA        NA        NA        NA        NA        NA        NA
##  [80,]        NA        NA        NA        NA        NA        NA        NA
##  [81,]        NA        NA        NA        NA        NA        NA        NA
##  [82,]        NA        NA        NA        NA        NA        NA        NA
##  [83,]        NA        NA        NA        NA        NA        NA        NA
##  [84,]        NA        NA        NA        NA        NA        NA        NA
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,79]     [,80]     [,81]     [,82]     [,83]     [,84]     [,85]
##    [1,]  82.15629 112.40009 109.83561  73.40207 100.91326  83.53958 103.07805
##    [2,] 100.89549 106.12958  72.15529  75.57994  83.98135  94.48649  77.02549
##    [3,] 111.85852  56.19630 100.49493  93.98859  80.95608 113.33867  97.51780
##    [4,]  92.28951  82.54736  78.47266  85.96507  91.60481  78.91708  59.77471
##    [5,]  65.30241 113.75866  71.13056 122.86709  45.63472  81.61402  78.63356
##    [6,]  72.62132  92.61110  81.33026  93.54628  96.97667  83.92162  85.62254
##    [7,] 108.80542  92.81453  88.91936  83.58051  79.01826 105.66088  81.76314
##    [8,] 107.12320  76.84821  98.17514  97.91205  77.86952 137.01041 123.50407
##    [9,]  99.72423  91.41879  89.09554 103.38374  82.61819  95.36259 102.10625
##   [10,] 100.45924  61.04242 121.67795  87.59847 112.61552 135.47970 135.21738
##  [11,] 117.91492  80.84773 114.57800  82.49809  93.68909 105.95474 123.29375
##  [12,]  63.75059  96.40531  61.92587 108.28520  62.93617  73.88469  80.87959
##  [13,] 110.17220  74.93725  63.89210  83.73702  69.34058 103.75212  81.42656
##  [14,]  94.03596  73.04269 100.42854  77.67715 126.34975  98.04143 114.19271
##  [15,]  97.80706  99.41027  80.06795  62.51346 106.92668  84.29230 101.27003
##  [16,]  74.85125 130.36431  85.04130 101.91802  82.15648  78.87339  88.63965
##  [17,]  83.64290  73.19590  89.12854 106.25510  74.35636  86.25378  68.25829
##  [18,]  99.69701  84.11829  60.62122  61.32080  89.06288  80.70566  78.24147
##  [19,]  80.78105 108.10958  56.45720  74.39367  84.95532  78.26222  79.30174
##  [20,] 115.18300  95.69973 113.11790  85.84665  77.37781 118.14103 127.80154
##  [21,]  82.50479  97.54221  56.74411  73.33421  75.70188  70.71945  62.30121
##  [22,]  85.89454  93.75685  94.56354  95.48569  90.82631  95.18492  67.77726
##  [23,] 115.74043  83.03083 111.75512  70.42433  90.12458  89.31513  74.37201
##  [24,]  95.79193  82.06604 100.42867  60.53671 130.63447  99.71810 109.87910
##  [25,]  88.51842  99.64281  52.61326  69.59705  99.22771  46.32436  71.42238
##  [26,] 106.84823  42.05600 122.15686  93.92019 102.78958 115.25040 106.20770
##  [27,] 109.18485 119.87910  64.82737  67.96785  67.09564  96.86454  90.93436
##  [28,] 125.02615  57.43231 108.91489  81.34815 128.98675 122.31539 122.16039
##  [29,]  70.23630  88.45749 118.84269  95.70053 129.09300  91.82137 121.79798
##  [30,]  87.58721 118.27085  96.74106  77.66266 104.57566  92.74108  96.93579
##  [31,]  46.65087 121.17411  74.96050  90.20438 108.27492  45.70753  76.33327
##  [32,] 103.74154  53.65306 107.65075  92.97324  99.36266 110.43470 125.45127
##  [33,] 121.43605  85.26936  96.86964  66.56448  75.74597 118.76883 110.36877
##  [34,] 108.96193  86.48676  96.41332  84.38637  58.98259 111.26445  98.58632
##  [35,]  87.21581  77.45584  91.00832  85.02537 119.81173 107.94158 110.42310
##  [36,]  82.87848  75.06726 121.51738 103.00285 118.43385 109.55653 144.26437
##  [37,]  92.75512  78.80966 100.19218  99.62318  92.02229 101.24769 117.83217
##  [38,]  86.60321  96.68914  89.52420  59.83581 134.91098  64.88303  97.82880
##  [39,]  76.22712 108.83554  76.49940 106.11456  70.73200  71.49779  54.56190
##  [40,] 133.03444  77.12866 114.91981  92.43313  81.38013 112.63910  92.25841
##  [41,]  97.59217  74.58941 110.56567  73.48570 110.81245  90.74899  76.89888
##  [42,]  82.26111  69.80161 113.55684 107.39534 119.33847  84.10669 115.18936
##  [43,]  88.97510  65.03271 120.43859 102.05021  96.17053  92.81876 106.29464
##  [44,]  78.84371  84.87899  67.66393  91.34670 110.36506  59.42772  80.89078
##  [45,]  88.17062 123.61897  63.00377  75.55246  69.96609  86.28475  86.51354
##  [46,]  99.71486  91.45218 125.27835  90.44781  96.79367 114.11455 118.43162
##  [47,]  72.93240  76.74053  73.96319 104.93004  71.07028  87.57859  73.26415
##  [48,]  65.01919 106.26828  98.72630  81.30174 102.20072  60.58019  58.15708
##  [49,]  96.84940  70.58237  92.26394  71.18946 119.82412  84.74082 101.47631
##  [50,] 115.41895  78.78341  74.73344  81.52927  80.69780 110.94848  95.90015
##  [51,] 105.72640  95.65545  68.53742  62.05660  78.18104  94.31613  87.60427
##  [52,]  82.00739  96.38120  85.81115  80.46448  94.53070  92.13947 116.96826
##  [53,] 110.40756  76.35492 101.11748  77.49604  78.24220 112.78602  88.86138
##  [54,] 104.17670 108.95831  79.00022  67.46323  89.69982 100.08949 110.14339
##  [55,]  88.49248 100.73869 114.93777 102.80927  94.28829 111.22375 108.46949
##  [56,] 110.59660  92.54515  99.67114  88.29086  92.07031  86.92471  93.50769
##  [57,] 109.79320 114.46835  80.81917  71.15827  54.65662  86.40696  78.77778
##  [58,] 117.01966  75.81176  98.85838  81.65067  80.29014 114.99675 124.66151
##  [59,] 132.27504  74.62596 100.15082  94.29193  68.72376 141.66215 113.20034
##  [60,]  95.55263  91.11471  73.63934  70.00988  97.68466  57.38584  67.69345
##  [61,]  68.76460  82.91036  55.63572 108.40674  89.60341  64.83154  62.54881
##  [62,] 113.91668  94.60759  62.87146  44.60976 116.51718  77.40033  76.06265
##  [63,]  85.97986  91.51027  68.94331 103.03793  45.93382 106.60705  92.15273
##  [64,]  88.93169  86.45983  67.58761  58.58495 133.43093  66.87098  67.00452
##  [65,]  48.56149 135.25005  96.18070 109.12990  73.42564  58.57291  71.58382
##  [66,]  74.00144  66.79865 122.42051 104.28268 129.83652  90.07011 111.69223
##  [67,]  95.65115  76.10551  86.84678  92.87661  91.30972  99.20047  77.21517
##  [68,]  82.89430  79.43485  91.86992  74.54726 134.05750  94.58411 107.93629
##  [69,]  59.63373  69.57840  98.60995 125.96843  91.23412  89.78467 108.92724
##  [70,]  82.75410 127.77231  84.34264  75.83780  94.51524  89.25470 103.73957
##  [71,]  85.36504  98.60257  87.75293  56.82011 114.90059  72.17850  66.15579
##  [72,]  62.15912 118.07606  50.10065  94.43949  74.10313  45.79440  55.42076
##  [73,]  96.38048 106.28263  49.00360  45.71299 111.52751  61.08599  63.46337
##  [74,]  90.23845  79.53398 127.45538 102.84678  91.79298 105.30639 115.62470
##  [75,]  68.51962  91.69410 104.01274 125.68689  70.43890  91.03440  99.42659
##  [76,]  89.40782  92.97567  82.42295  68.37071  98.65674  69.15590  93.03230
##  [77,]  89.57328  86.06556 126.62312  93.86062  72.76531 109.11294  84.50098
##  [78,]  73.40596  84.27421 114.61830 104.54011  98.95084  98.45854 103.34142
##  [79,]        NA 107.89330  82.92510 122.54064  85.75853  59.79099  81.37105
##  [80,]        NA        NA 112.15278 100.52456 112.58796 124.44606 117.89057
##  [81,]        NA        NA        NA  70.69994  83.63153  56.88949  60.62170
##  [82,]        NA        NA        NA        NA 111.74036  78.62720  74.93558
##  [83,]        NA        NA        NA        NA        NA  94.50924  75.33899
##  [84,]        NA        NA        NA        NA        NA        NA  47.74442
##  [85,]        NA        NA        NA        NA        NA        NA        NA
##  [86,]        NA        NA        NA        NA        NA        NA        NA
##  [87,]        NA        NA        NA        NA        NA        NA        NA
##  [88,]        NA        NA        NA        NA        NA        NA        NA
##  [89,]        NA        NA        NA        NA        NA        NA        NA
##  [90,]        NA        NA        NA        NA        NA        NA        NA
##  [91,]        NA        NA        NA        NA        NA        NA        NA
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,86]     [,87]     [,88]     [,89]     [,90]     [,91]     [,92]
##   [1,]  67.36217 106.24689 113.21095 113.75571 105.24102  87.95734 104.68508
##   [2,] 102.55576  49.62210  62.94470 105.08183 108.37453  81.24917  84.64672
##   [3,] 110.26841  86.48648  67.01362  81.18083  67.03001  59.67033  77.37160
##   [4,]  77.85379  99.40157  80.78196  80.73826  72.95758  53.02708 109.46169
##   [5,] 100.22625  61.12445  88.33293 120.24364  88.48160 116.57738 105.88571
##   [6,]  86.20932  53.20511  58.64445 113.13020  94.38261 104.74385  69.62258
##   [7,]  91.88037  75.23657  91.00870 125.88207  91.74883  73.21601 109.15730
##   [8,] 121.97560  38.17186  64.13575 120.36030 102.13723  84.66176  65.17957
##   [9,]  80.13792  87.21437 125.29750 117.84366  98.01460  92.26876 101.65027
##  [10,]  90.77781  85.86753  90.53618 115.57473  96.39120  84.05682  76.21577
##  [11,] 104.47471  97.05117 101.06948  75.37640  94.28292  85.24567  65.15955
##  [12,]  92.86368  66.12196  69.90150 113.08050  89.20762 111.32184  85.85313
##  [13,] 127.48370  58.39685  43.26778  83.46271  90.18107  88.78386  65.45864
##  [14,]  83.25177 103.68972  96.33185  71.92941 102.96232  99.48987  65.62079
##  [15,]  62.69246  88.91973 106.50229 130.29760 124.82945  86.14592  90.98972
##  [16,]  91.53540  62.32916  97.79516  99.71413 109.88870 110.59846  91.95444
##  [17,] 109.12683 103.04001  73.15437  54.44063  51.04882 101.67970  89.92881
##  [18,]  86.94636 119.10152  90.09114  70.40720 101.78202  73.70227 100.57071
##  [19,]  84.28811  68.16073  76.62259 120.25258 117.85200 103.92155  90.31098
##  [20,] 107.40477  82.56730 108.79685 112.82643 105.97532  92.79157  83.60976
##  [21,] 104.21479  79.15391  46.71962  76.76884  91.81119  98.12358  78.84875
##  [22,]  89.73517  77.03217  72.44718  90.75752  75.78352  60.93529 108.96231
##  [23,]  87.92201 128.52847  97.94566  68.10197  63.46447  48.86974 109.92330
##  [24,]  59.61473 109.79761 109.38031 111.50556 113.98016  79.90234  94.64969
##  [25,]  74.08492 107.81396  95.18115  58.79172 105.76444  95.96609  80.25839
##  [26,]  97.52736  87.85332  64.53361  87.91016  57.64568  65.03141  64.09101
##  [27,] 100.34050  72.03090  96.38355 116.41583 131.78971  76.65560 106.60129
##  [28,]  81.34520  83.67128 102.46556 100.20595 101.20811  71.21358  68.07684
##  [29,]  58.58614 101.04463 116.98471 105.56509 100.71944 104.96272  84.00614
##  [30,]  69.56649 100.56763 124.65581 102.82272 119.47795  74.44942 122.39613
##  [31,]  39.44817  97.13640 110.83312 110.35727 107.32725 109.41419 107.15392
##  [32,] 105.33954  79.76827  68.57746  83.47704  82.87567  88.82518  38.53984
##  [33,] 117.09715  83.61257  79.28952 101.84499 103.05980  79.87389  80.00677
##  [34,] 128.44514  83.28683  67.03610  82.29079  84.07303  70.94636  84.90881
##  [35,]  59.69154 104.10246 120.95859 118.77101 112.82560  81.35212 109.90880
##  [36,]  69.30854  98.12922 121.92004 111.02690 101.47158  95.18432  81.46344
##  [37,]  94.71903  53.63607  73.62165 120.51884  94.06406  99.97552  51.55156
##  [38,]  52.85802 121.49490 115.11001  77.61446 114.32896  96.52422  81.06773
##  [39,] 106.08626  59.73735  57.97092  74.26659  73.74665  96.21263  86.73551
##  [40,] 120.23883  79.62863  83.02563  70.17819  68.28079  70.97235  75.19239
##  [41,]  74.46733 119.04272  84.16793  73.70974  66.77768  40.15749 105.32757
##  [42,]  65.22336 106.24355 119.21685  89.96684  80.52855 114.49293  69.04282
##  [43,] 104.26961  92.34073  68.77525  63.12328  59.29756 100.17977  47.54212
##  [44,]  74.18550  74.51541  74.79427  80.16316  94.00178 108.41158  53.86100
##  [45,]  98.45821  67.73282  87.94107 117.76238 126.26175 107.21328  97.28620
##  [46,]  82.29422  86.66502 107.36841 123.28261  95.05838  68.30126  98.94514
##  [47,]  95.44128  74.21810  58.49705 107.97733  70.11611  97.01570  90.59981
##  [48,]  67.24092 126.81024  92.53532  64.50272  71.33507  77.14173 119.60330
##  [49,]  60.20159 146.01337 125.36184  78.48944  96.33849  83.02391  97.44177
##  [50,] 114.65585  93.09429  92.38655  72.16447 102.47195  88.48084  88.27305
##  [51,]  90.04732 112.99481 104.25499  94.44046 112.12915  70.82515 117.05940
##  [52,]  90.77106  98.97726 102.00471  91.33929 118.91863 114.61884  79.69842
##  [53,] 111.54606  72.22209  50.77212 102.31055  76.52172  60.30794  82.36631
##  [54,]  81.71499  90.67665 116.11134 116.92975 138.40304  72.88767 106.67192
##  [55,]  92.06694  62.92568  92.50595 114.40457  96.89507  81.41914  91.87950
##  [56,] 103.67444  76.52775  83.02577  66.43493  86.86995  84.50703  61.72812
##  [57,] 108.17103  94.78197  91.64762  88.57383  97.11483  71.56933 108.37611
##  [58,] 117.99549  85.06741  87.82712  90.24994 100.44949 100.59118  60.26637
##  [59,] 129.95792  61.04100  84.30926 107.24650  93.34479  81.78096  79.05516
##  [60,]  62.27832 127.93112 112.07943  80.92761  87.48340  85.25414 104.17792
##  [61,]  81.57308  83.65809  79.19250  77.00573  79.87012 108.92612  86.22456
##  [62,]  70.07387 112.43879 109.75490  77.44924 121.05707  79.39182  97.83256
##  [63,] 128.41880  53.71801  57.13481 107.07105  94.14115 102.94677  83.81032
##  [64,]  43.73148 110.32914  94.96157  85.74507 104.19471  61.95402 102.75453
##  [65,]  78.74711  94.20110 101.79874  91.97566  81.66511 105.71868 117.30495
##  [66,]  52.35221 120.44461 122.51408  95.33625  76.34960  98.97878  89.70452
##  [67,]  86.02046 111.79232  99.46094  75.23361  77.98664  60.56365 119.81275
##  [68,]  48.85279 119.46807 119.94744 100.56677 113.58479  75.01194 105.75726
##  [69,]  85.33097  66.75481  74.37552 117.40953  73.70909 128.16470  59.33466
##  [70,]  72.97386  82.19657 115.58092 127.40859 138.09903  91.62646 109.52930
##  [71,]  56.32182 138.00430 103.84074  75.99080  93.10709  55.11910 128.71131
##  [72,]  72.74418  87.76486  89.32755  93.70545  95.48103  96.08377 109.62256
##  [73,]  53.00517 119.83849 111.10740  82.46414 123.32756  62.72339 119.44588
##  [74,] 108.44216  76.50924  70.16905  80.09631  73.63194  88.86956  57.96689
##  [75,]  93.50450  65.54833  80.76887 116.90608  72.21957  99.93150  84.87760
##  [76,]  76.73377 111.90090 100.26101  87.82573 100.97626 109.88366  78.97477
##  [77,] 108.43597 100.80166  73.61781  87.81795  54.00829  70.62247 109.13663
##  [78,]  92.10121 109.69555  97.99608  66.78254  78.20553  95.18647  93.82640
##  [79,]  70.57838  87.17968  89.46124 100.77330  82.91436 129.39526  98.49947
##  [80,]  95.22917  89.88712  76.26309  87.04640  67.04426  80.61500  56.63885
##  [81,]  79.92172  84.57191  86.26890  87.74659 123.07905  96.94063  97.86655
##  [82,]  74.14254 116.17458  98.09054  77.00855 120.87416  53.74030 101.12624
##  [83,] 136.92275  66.12010  70.41881  91.50406  75.23851 102.13603 101.65598
##  [84,]  58.60188 109.94964  98.22335  72.48162  96.42583 105.26835 104.60814
##  [85,]  85.09661 103.23763  77.57777  63.67909  77.82161  78.70859 119.84885
##  [86,]        NA 122.40816 126.59672 101.53851 108.96888  87.97723 107.34230
##  [87,]        NA        NA  56.01109 114.46063  93.66850 108.08919  61.59181
##  [88,]        NA        NA        NA  77.84189  66.14021  88.14247  62.50575
##  [89,]        NA        NA        NA        NA  69.51441  79.42389  84.33594
##  [90,]        NA        NA        NA        NA        NA  88.87780  85.91527
##  [91,]        NA        NA        NA        NA        NA        NA 109.24105
##  [92,]        NA        NA        NA        NA        NA        NA        NA
##  [93,]        NA        NA        NA        NA        NA        NA        NA
##  [94,]        NA        NA        NA        NA        NA        NA        NA
##  [95,]        NA        NA        NA        NA        NA        NA        NA
##  [96,]        NA        NA        NA        NA        NA        NA        NA
##  [97,]        NA        NA        NA        NA        NA        NA        NA
##  [98,]        NA        NA        NA        NA        NA        NA        NA
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##            [,93]      [,94]      [,95]      [,96]      [,97]      [,98]      [,99]
##   [1,]  83.57487  83.11076  99.98068 125.21884  94.81987  81.42386  82.53554
##   [2,]  66.21635 110.59481  82.88926  88.17225 128.75849  91.12727 103.68146
##   [3,]  75.48195 103.43209 106.24086  57.03405  92.32903 104.01303  93.89541
##   [4,]  57.10825  84.90060  78.40685  89.92633  95.99568 114.88480  93.09386
##   [5,] 100.78406 103.62940  83.01494  91.35123 111.48976  83.66008  79.40806
##   [6,]  68.62637 123.20965 106.05169  83.87588 128.10180 103.53491  84.04930
##   [7,]  77.34901  98.88121  67.54432  96.49275 123.78028  64.85447 106.25504
##   [8,]  84.60448 129.96524 106.64637  69.44914 116.86936  91.04548 118.95533
##   [9,] 100.27942  91.58824  89.05816  91.78258 110.78627  76.60979  95.89583
##  [10,]  92.47717 104.88685  95.92878  93.74029  84.25038  81.08762 124.36857
##  [11,] 105.77887  85.17434 122.60388  71.74269  86.77635  80.61331  83.62708
##  [12,]  73.56052 129.86779 106.47083  76.20565 108.92719 114.12673  65.14258
##  [13,]  75.80962 125.98341  89.64957  48.69335 100.02852  95.59884  91.13951
##  [14,] 101.19366  86.16541  97.61823  90.07919  61.33049  93.29925 101.35351
##  [15,]  58.92917 116.41476 101.44067 104.94769 117.85709  95.34044  89.03910
##  [16,] 105.65588  81.94370  90.67085 105.99083 113.39810  83.60982  92.60897
##  [17,] 113.11724  73.63926  72.83374  70.42832  59.05114  77.22509  74.69776
##  [18,]  67.23567  95.98297  76.22876  89.77479  51.11324 112.69401  88.29477
##  [19,]  63.57102 125.86060  84.57267  97.45869 110.83320  97.73167  86.12603
##  [20,] 109.64527  94.56640 104.05186  88.64912 101.35475  56.62573  95.07206
##  [21,]  65.29968 114.44437  90.76391  74.91888  88.41692 107.21050  65.92341
##  [22,]  73.94138  77.69627  68.77224 102.52203 104.42004  97.28155 114.45531
##  [23,]  82.47178  58.44054  82.57504  91.89031  82.43501  73.86352  79.19400
##  [24,]  73.54224  94.83565  84.04439 118.33278  80.65533  89.00827 111.47690
##  [25,]  72.93217  92.68743 103.51545  80.76145  80.75701 121.85916  58.81042
##  [26,]  74.86775 104.14698 120.42792  58.64536 102.74239  98.10188  89.51801
##  [27,]  71.69627 107.71449  77.78403 102.26269 107.19524  88.57623 105.88558
##  [28,]  87.05491  96.54361  98.26313  80.16657 107.48579  84.99582 126.32168
##  [29,] 102.65069  81.16353 105.48820 116.27417  84.30620  90.51778  99.17681
##  [30,]  90.94073  64.47311  66.50524 143.91468  81.83217  86.95719 124.99052
##  [31,]  75.78943  88.41226  92.38627 128.05452 101.48666 109.55871  74.31854
##  [32,]  84.91349 118.67710 145.56289  48.25968  94.44466 108.21154  81.65366
##  [33,]  84.38629 110.60555  95.98040  79.00497  90.57862  72.12341  94.58872
##  [34,]  84.72149 100.56067 101.42450  71.03246  82.87118  91.68449  88.00327
##  [35,]  84.80667  91.64540  64.13100 123.08022  71.00069  95.14405 139.23549
##  [36,] 104.00583  90.10507 112.68603 103.64340  81.90653  96.09397 110.01718
##  [37,]  78.01947 132.95610 144.32344  63.98256 140.72393  99.40817  78.62326
##  [38,]  82.06400  80.95130 104.47691 108.36464  75.79712 101.83881  77.35711
##  [39,]  89.22549  85.06579  87.58835  78.45411 115.55658  95.00255  76.73731
##  [40,] 107.86772  72.65469  95.11970  60.29490 111.59509  57.62362  92.81579
##  [41,]  62.85679  69.95622  85.45687 100.32742  81.05852 103.19057  96.43192
##  [42,] 114.46473  82.38229 118.80707  82.76397  90.90715  83.64703  73.80933
##  [43,] 103.02619  90.98386 139.43129  54.76708  87.00520  93.28474  59.06400
##  [44,]  72.07216 112.34169 123.53610  65.69451 117.05331 122.64619  63.28934
##  [45,]  83.93577 111.57100  77.64340 103.35467 103.26703  79.67864  91.00350
##  [46,]  86.03454  87.93489 102.32476 107.52648 111.49511  80.80894 109.30001
##  [47,]  71.56433 121.29611  88.32130  74.47483 101.13342 101.45838  77.82257
##  [48,]  82.82241  49.34032  73.26949 123.27831  64.57920  97.41318  77.89815
##  [49,]  88.86716  78.20479  85.74858 101.16491  48.45003  95.93256  90.59120
##  [50,] 106.26471  88.62811  63.70507  78.51689  57.36555  78.72531 113.94225
##  [51,]  74.30329  94.45145  64.74448 104.63008  60.87361  93.08616 101.82902
##  [52,] 102.69805  99.37347  98.20057  99.03065  57.31364  93.17621  89.54138
##  [53,]  60.20819 113.44710  99.52579  72.73329 114.22977  89.29001  92.16367
##  [54,]  73.55510 100.84156  85.93991 114.43199  90.65411  99.14183 115.21126
##  [55,]  95.05858  85.72357  94.56609 105.61129 117.57463  83.67056 118.93532
##  [56,]  94.86471  83.31762 121.15724  64.41931 116.18204  89.61321  75.81301
##  [57,]  80.48272  87.68223  87.08073  90.70952  94.16634  81.00160  75.87166
##  [58,] 105.05569 109.39479 112.34636  63.11114  83.54514  73.29392  84.42407
##  [59,] 105.65796 104.33649  83.57986  66.23810 107.55018  55.99052 117.66778
##  [60,]  75.58896  81.99147  86.33754  92.77632  86.17169  90.80663  59.54494
##  [61,]  87.37497  97.25488  81.10212  75.68394  88.66169 111.74362  79.42442
##  [62,]  75.41139  82.61091  61.83573 103.25345  77.98418  86.74474 101.50998
##  [63,]  88.06379 127.86203  88.05726  70.62732  92.85609  94.06640  92.79646
##  [64,]  39.74295  90.02831  76.01296 110.30034  89.96183 125.98442 101.72938
##  [65,] 103.44547  62.28519  89.18579 121.23546  93.01875  87.62901  67.07990
##  [66,] 106.36953  72.49900  99.04535 104.56555  75.16032  86.97578  91.88713
##  [67,]  88.61558  67.63032  52.49983 100.55030  59.27100  96.72993 119.88762
##  [68,]  75.16159  85.59661  76.92405 126.20717  59.25279 113.02438 125.08663
##  [69,]  94.50467 126.11048 129.96211  68.00618 112.07366 100.38343  67.56468
##  [70,]  82.20591  93.92369  82.85740 133.67435 100.50290  89.98181 110.06024
##  [71,]  62.31575  61.78448  60.44838 132.78785  59.33782 104.22195  99.81965
##  [72,]  67.08980  97.05665  88.09386  99.13597 101.37405 120.92916  66.08649
##  [73,]  48.23470  84.01441  62.61220 119.55338  76.14488 118.68053 100.09357
##  [74,]  98.72180  91.67351 136.24437  70.51859  99.17756  95.02673  81.00044
##  [75,]  90.71246 106.15662 121.56140  80.49671 122.73877  98.25318  73.87137
##  [76,]  86.73533  99.29536 106.23969  88.09179  81.42562  86.07550  54.40003
##  [77,]  96.32841  70.55236  77.65376  98.51600  77.25757  65.87733  90.98889
##  [78,] 119.34779  60.09599  84.19652 103.60833  42.08969  89.54452 102.38011
##  [79,]  97.90894  91.39197  96.39709 109.01059  82.31818 110.49205  69.59792
##  [80,]  89.98817 107.99891 108.19330  53.56679  85.50741  95.79505 104.45956
##  [81,]  61.33172 110.45814  74.81407  93.53633  89.66056 120.48164  83.46297
##  [82,]  50.43258  86.10457  74.82361 105.86813  83.35738  96.91350  95.81583
##  [83,] 110.27956  90.32211  80.65764  78.14015  95.42433  68.19706  75.48002
##  [84,]  74.95999  78.01502  88.41353 107.81256  87.53741 111.15486  48.04207
##  [85,]  71.43404  68.57960  59.79127 100.96232  88.81058  93.53622  67.13854
##  [86,]  66.98815  83.51355  86.64658 126.80906  85.41975 114.04203  89.90086
##  [87,]  93.14558 128.76313 109.52177  62.28896 131.53164  92.64437  96.73961
##  [88,]  75.87468 115.62017 105.49900  51.63015 104.83133 102.78468  78.71102
##  [89,]  95.73180  59.31460  85.74085  74.85540  62.98302  91.66147  70.51790
##  [90,] 109.83692  71.35911  95.49541  66.97721  88.30948  72.76292  70.05090
##  [91,]  57.43747  76.14037  69.80967  98.17472  88.84218  91.90780 116.63198
##  [92,]  93.87476 125.06127 143.55437  36.80696 107.61621 102.22643  79.21741
##  [93,]        NA 113.80995  89.87064  96.29646 103.54579 130.47017  93.96910
##  [94,]        NA        NA  60.69432 117.46869  65.49962  61.70059  87.64873
##  [95,]        NA        NA        NA 123.63216  66.83386  72.14413 115.39448
##  [96,]        NA        NA        NA        NA 106.80496  91.44877  74.94007
##  [97,]        NA        NA        NA        NA        NA  92.74388  96.85984
##  [98,]        NA        NA        NA        NA        NA        NA  91.25875
##  [99,]        NA        NA        NA        NA        NA        NA        NA
## [100,]        NA        NA        NA        NA        NA        NA        NA
##           [,100]
##    [1,]  88.13120
##    [2,]  50.38906
##    [3,] 131.72676
##    [4,] 103.09504
##    [5,]  63.24204
##    [6,]  55.90199
##    [7,]  60.36343
##    [8,]  81.81044
##    [9,]  78.91581
##   [10,]  91.96956
##   [11,] 124.43170
##   [12,]  76.68294
##   [13,]  79.93407
##   [14,]  95.84557
##   [15,]  60.74356
##   [16,]  59.08495
##   [17,] 108.53906
##   [18,] 104.43899
##   [19,]  37.30364
##   [20,]  92.37859
##   [21,]  78.61347
##   [22,]  85.71972
##   [23,] 130.35379
##   [24,]  76.52183
##   [25,]  93.77589
##   [26,] 128.99188
##   [27,]  63.49280
##   [28,]  85.96402
##   [29,]  87.42652
##   [30,]  76.88309
##   [31,]  59.03378
##  [32,] 124.92614
##  [33,]  94.31942
##  [34,] 123.95579
##  [35,]  72.05671
##  [36,] 103.62132
##  [37,]  84.77943
##  [38,]  88.72438
##  [39,]  77.53427
##  [40,] 108.99936
##  [41,] 125.58491
##  [42,] 100.63196
##  [43,] 136.10067
##  [44,]  76.35303
##  [45,]  40.90304
##  [46,]  98.33707
##  [47,]  80.46786
##  [48,] 107.44370
##  [49,] 110.32021
##  [50,]  89.63288
##  [51,]  92.46245
##  [52,]  88.72796
##  [53,]  99.83069
##  [54,]  79.36613
##  [55,]  80.16705
##  [56,] 105.46070
##  [57,] 101.50195
##  [58,] 102.64809
##  [59,]  83.16174
##  [60,]  93.08946
##  [61,]  74.26837
##  [62,]  63.73757
##  [63,]  78.63378
##  [64,]  76.23217
##  [65,]  89.62043
##  [66,] 104.08197
##  [67,] 107.16515
##  [68,]  89.61737
##  [69,]  84.69250
##  [70,]  56.05418
##  [71,]  99.67168
##  [72,]  75.89632
##  [73,]  74.46536
##  [74,] 127.14159
##  [75,]  98.45572
##  [76,]  87.60124
##  [77,] 119.03254
##  [78,] 119.55153
##  [79,]  79.89984
##  [80,] 115.93714
##  [81,]  54.43487
##  [82,]  83.26185
##  [83,]  91.21020
##  [84,]  76.58716
##  [85,]  81.38402
##  [86,]  74.52426
##  [87,]  67.57332
##  [88,]  98.71847
##  [89,] 122.89009
##  [90,] 128.07622
##  [91,] 107.93237
##  [92,]  98.75656
##  [93,]  78.03359
##  [94,] 109.90068
##  [95,]  73.63532
##  [96,] 108.48049
##  [97,] 113.44228
##  [98,]  92.48617
##  [99,]  99.88902
## [100,]        NA
```

Plot the density of these angles.

```
pacman::p_load(ggplot2)
ggplot(data.frame(angles=c(all_angles(X)))) +
  aes(x = angles) +
  geom_density()
```

```
## Warning: Removed 5050 rows containing non-finite values (stat_density).
```



Write an Rcpp function `all_angles_cpp` that does the same thing. Use an IDE if you want, but write it below in-line.

```
pacman::p_load(Rcpp)

cppFunction(
  "
   NumericMatrix all_angles_cpp(NumericMatrix X) {
    int n = X.nrow();
    int p = X.ncol();
    NumericMatrix A(n, n);
    std::fill(A.begin(), A.end(), NA_REAL);

    for (int i_1 = 0; i_1 < (n - 1); i_1++) {
      for (int i_2 = i_1 + 1; i_2 < n; i_2++) {
        double sum_sqd_u = 0;
        double sum_sqd_v = 0;
        double sum_u_times_v = 0;
        for (int j = 0; j < p; j++) {
          sum_sqd_u += pow(X(i_1, j), 2);
          sum_sqd_v += pow(X(i_2, j), 2);
          sum_u_times_v += X(i_1, j) * X(i_2, j);
        }
        A(i_1, i_2) = acos(sum_u_times_v / sqrt(sum_sqd_u * sum_sqd_v)) * (180/M_PI);
      }
    }
    return A;
  }
  "
)
```

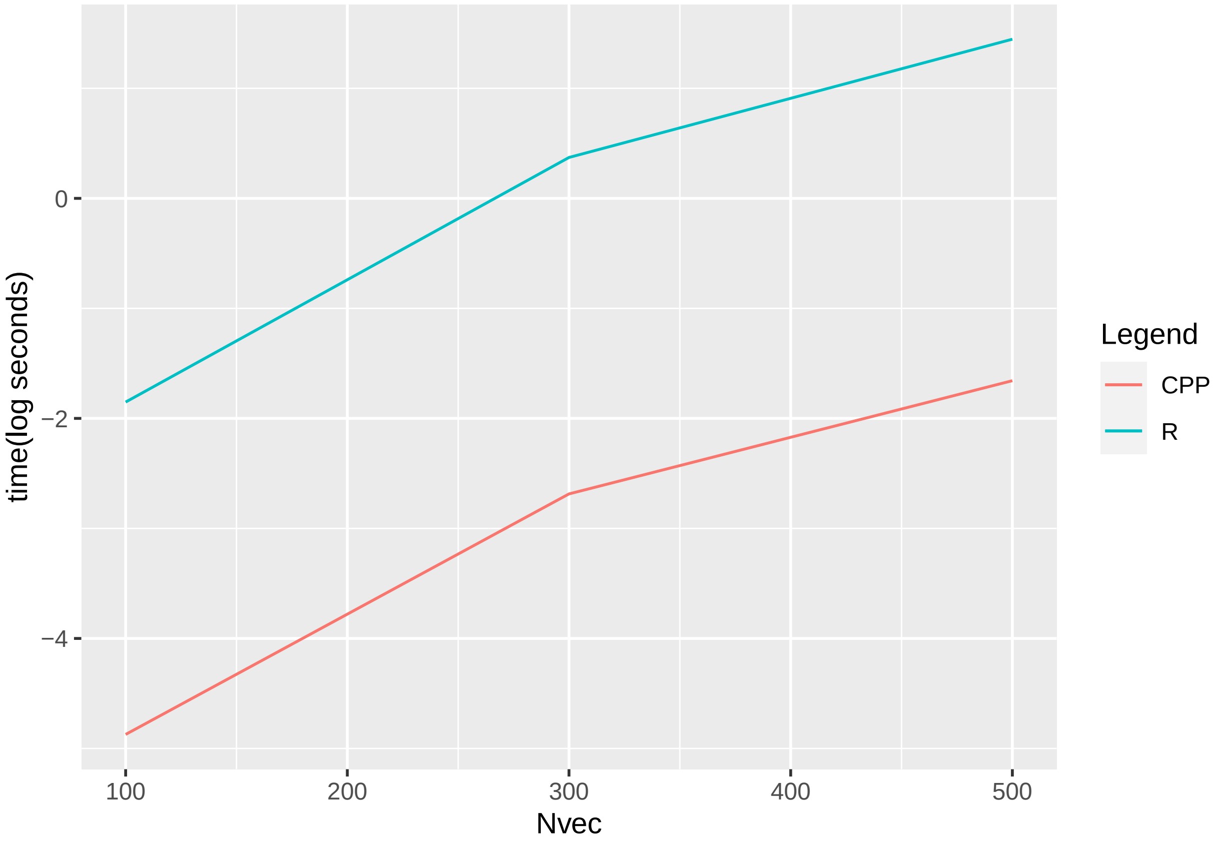
Test the time difference between these functions for `n = 1000` and `Nvec = 100, 500, 1000, 5000` using the package `microbenchmark`. Store the results in a matrix with rows representing `Nvec` and two columns for base R and Rcpp.

```
pacman::p_load(microbenchmark)
n <- 1000
Nvec <- c(100, 300, 500)
benchmarkMatrix = matrix(data = NA, nrow = length(Nvec), ncol = 2)

for( i in 1:length(Nvec)){
  X = matrix(data = rnorm(Nvec[i] * n), nrow = Nvec[i])
  stored = summary(microbenchmark(all_angles(X), all_angles_cpp(X), times = 10, unit= "s"))
  benchmarkMatrix[i,1] = stored[1,2]
  benchmarkMatrix[i,2] = stored[2,2]
}
benchmarkMatrix
```

```
##            [,1]        [,2]
## [1,] 0.1570091 0.007654829
## [2,] 1.4501711 0.068141938
## [3,] 4.2487208 0.190652625
```

Plot the divergence of performance (in log seconds) over n using a line geometry. Use two different colors for the R and CPP functions. Make sure there's a color legend on your plot. We wil see later how to create "long" matrices that make such plots easier.

```
colors <- c("R" = "blue", "CPP" = "red")
ggplot(data.frame(log(benchmarkMatrix)))+
  aes(x = Nvec)+
  geom_line(aes(y = log(benchmarkMatrix[,1]), color = "R")) +
  geom_line(aes(y = log(benchmarkMatrix[,2]), color = "CPP")) +
   labs(x = "Nvec",
        y = "time(log seconds)",
        color = "Legend")
```



```
  scale_color_manual(values = colors)
```

```
## <ggproto object: Class ScaleDiscrete, Scale, gg>
##     aesthetics: colour
##     axis_order: function
##     break_info: function
##     break_positions: function
##     breaks: waiver
##     call: call
##     clone: function
##     dimension: function
##     drop: TRUE
##     expand: waiver
##     get_breaks: function
##     get_breaks_minor: function
##     get_labels: function
##     get_limits: function
##     guide: legend
##     is_discrete: function
##     is_empty: function
##     labels: waiver
##     limits: NULL
##     make_sec_title: function
##     make_title: function
##     map: function
##     map_df: function
##     n.breaks.cache: NULL
##     na.translate: TRUE
##     na.value: NA
##     name: waiver
##     palette: function
##     palette.cache: NULL
##     position: left
##     range: <ggproto object: Class RangeDiscrete, Range, gg>
##         range: NULL
##         reset: function
##         train: function
##         super:  <ggproto object: Class RangeDiscrete, Range, gg>
##     rescale: function
##     reset: function
##     scale_name: manual
##     train: function
##     train_df: function
##     transform: function
##     transform_df: function
##     super:  <ggproto object: Class ScaleDiscrete, Scale, gg>
```

Let `Nvec = 10000` and vary `n` to be 10, 100, 1000. Plot the density of angles for all three values of `n` on one plot using color to signify `n`. Make sure you have a color legend. This is not easy.

```
# #TO-DO --> to fix n =  10, 100, 1000
# # fix, -- do density, no benchmark
#
# n <- c(10, 100)
# Nvec <- 10000
# #density_n = matrix(data = NA, nrow = length(n), ncol = 2)
# #density_r = list()
#
# density_n = matrix(data = NA, nrow = 1000*1000, ncol =3 )
#
# for( i in 1:length(n)){
#   X = matrix(data = rnorm(n[i] * Nvec), nrow = n[i])
#   #density_r[i] <- list(all_angles(X))
#   print(c(all_angles(X)))
#   density_n[,i] <- c(all_angles(X))
#   #density_r <- list(density_r,c(all_angles(X)))
#   #density_n[i,1] = c(all_angles(X))
#   #density_n[i,2] = list(c(all_angles_cpp(X)))
#   # stored = summary(microbenchmark(all_angles(X), all_angles_cpp(X), times = 10, unit= "s"))
#   # benchmarkMatrix_n[i,1] = stored[1,2]
#   # benchmarkMatrix_n[i,2] = stored[2,2]
# }
# #c(density_r[[1]])
# #c(density_r[[2]])
# #density_n
# #
# # df <- data.frame(matrix(unlist(density_r), ncol = max(lengths(density_r)), byrow = TRUE))
# # names(df) <- names(density_r[[which(lengths(density_r)>0)[1]]])
# # df
# # ggplot(data.frame(density = c(density_r[[1]]))) +
# #   aes(x = density) +
# #   geom_density()
#
# # ggplot(data.frame(density = density_n)) +
# #   aes(x = density,
# #       color = variable,
# #          fill = variable) +
# #   geom_density()
#
# ggplot(data.frame(density = density_n), aes(x = density)) +
#    geom_line(aes(y = 1, color = "Red")) +
#    geom_line(aes(y = 2, color = "Blue")) +
#    geom_line(aes(y = 3, color = "Green"))
```

Write an R function `nth_fibonnaci` that finds the nth Fibonnaci number via recursion but allows you to specify the starting number. For instance, if the sequency started at 1, you get the familiar 1, 1, 2, 3, 5, etc. But if it started at 0.01, you would get 0.01, 0.01, 0.02, 0.03, 0.05, etc.

```
nth_fibonnaci <- function(nth, starting_num){
  if( nth <= 2){
    return(starting_num)
  }
  return(nth_fibonnaci(nth-1,starting_num)+ nth_fibonnaci(nth-2,starting_num))
}

x = nth_fibonnaci(5, 0.01)
x
```

```
## [1] 0.05
```

Write an Rcpp function `nth_fibonnaci_cpp` that does the same thing. Use an IDE if ou want, but write it below in-line.

```
cppFunction(
  "
   double nth_fibonnaci_cpp(int nth, double starting_num) {
    if( nth <= 2){
      return starting_num;
    }
    return nth_fibonnaci_cpp(nth-1, starting_num)+ nth_fibonnaci_cpp(nth-2,starting_num);
   }
  "
)
```

```
x =  nth_fibonnaci_cpp(5, 0.01)
x
```

```
## [1] 0.05
```

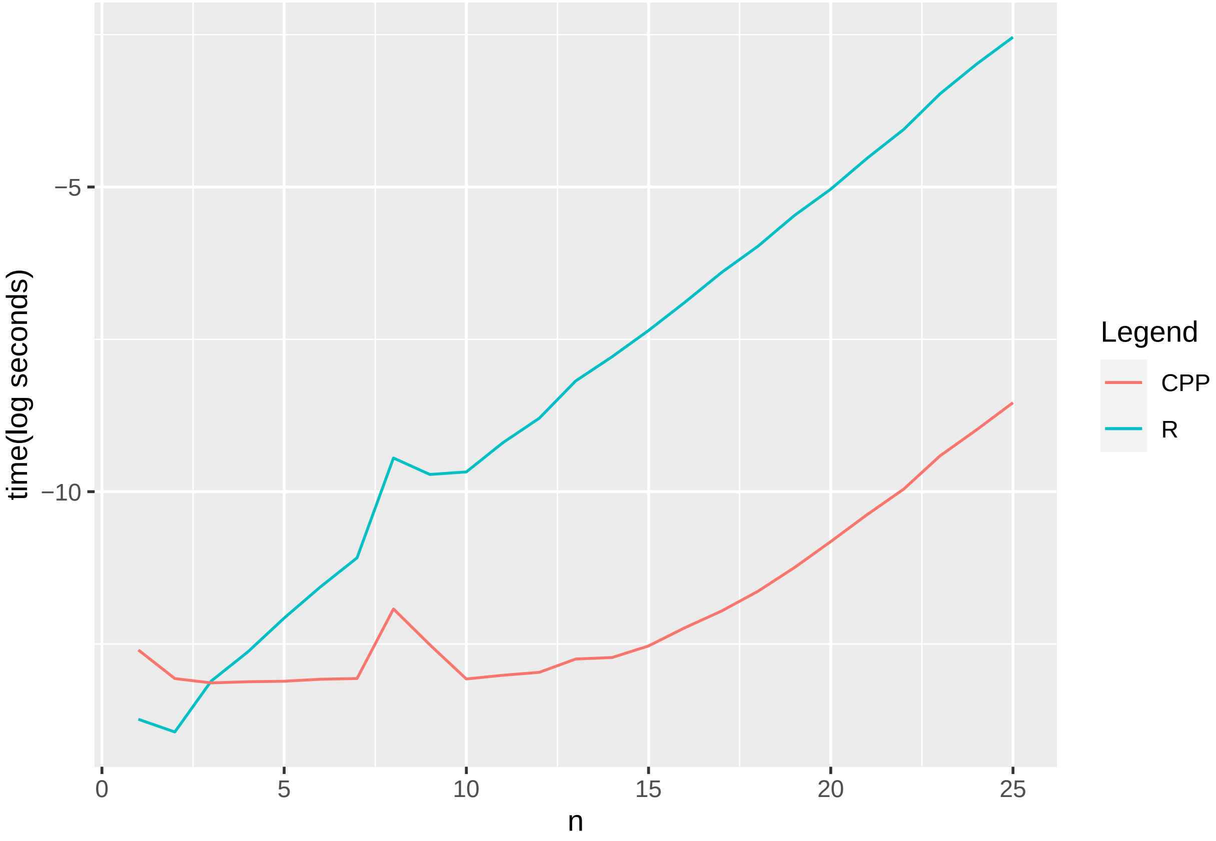
Time the difference in these functions for n = 100, 200, ..., 1500 while starting the sequence at the smallest possible floating point value in R. Store the results in a matrix.

```
n <- seq(1, 25, by = 1)
benchmarkMatrix_nth = matrix(data = NA, nrow = length(n), ncol = 2)

for( i in 1:length(n)){
  stored = summary(microbenchmark(nth_fibonnaci(n[i], .Machine$double.xmin), nth_fibonnaci_cpp(n[i], .Ma
  benchmarkMatrix_nth[i,1] = stored[1,2]
  benchmarkMatrix_nth[i,2] = stored[2,2]
}
benchmarkMatrix_nth
```

```
##              [,1]        [,2]
##  [1,] 0.000001084 0.000003380
##  [2,] 0.000000880 0.000002114
##  [3,] 0.000002032 0.000001970
##  [4,] 0.000003270 0.000002007
##  [5,] 0.000005704 0.000002024
##  [6,] 0.000009539 0.000002091
##  [7,] 0.000015329 0.000002115
##  [8,] 0.000078879 0.000006625
##  [9,] 0.000060201 0.000003677
## [10,] 0.000062794 0.000002100
## [11,] 0.000101299 0.000002232
## [12,] 0.000151564 0.000002341
## [13,] 0.000279397 0.000002913
## [14,] 0.000415819 0.000002987
## [15,] 0.000639672 0.000003611
## [16,] 0.001017762 0.000004873
## [17,] 0.001654220 0.000006398
## [18,] 0.002545345 0.000008849
## [19,] 0.004218166 0.000013051
## [20,] 0.006502226 0.000020024
## [21,] 0.010813058 0.000031173
## [22,] 0.017316870 0.000047310
## [23,] 0.031113107 0.000081803
## [24,] 0.050613148 0.000125227
## [25,] 0.078787818 0.000195587
```

Plot the divergence of performance (in log seconds) over n using a line geometry. Use two different colors for the R and CPP functions. Make sure there's a color legend on your plot.

```
ggplot(data.frame(log(benchmarkMatrix_nth)))+
  aes(x = n)+
  geom_line(aes(y = log(benchmarkMatrix_nth[,1]), color = "R")) +
  geom_line(aes(y = log(benchmarkMatrix_nth[,2]), color = "CPP")) +
   labs(x = "n",
        y = "time(log seconds)",
        color = "Legend")
```



```
  scale_color_manual(values = colors)
```

```
## <ggproto object: Class ScaleDiscrete, Scale, gg>
##     aesthetics: colour
##     axis_order: function
##     break_info: function
##     break_positions: function
##     breaks: waiver
##     call: call
##     clone: function
##     dimension: function
##     drop: TRUE
##     expand: waiver
##     get_breaks: function
##     get_breaks_minor: function
##     get_labels: function
##     get_limits: function
##     guide: legend
##     is_discrete: function
##     is_empty: function
##     labels: waiver
##     limits: NULL
##     make_sec_title: function
##     make_title: function
##     map: function
##     map_df: function
##     n.breaks.cache: NULL
##     na.translate: TRUE
##     na.value: NA
##     name: waiver
##     palette: function
##     palette.cache: NULL
##     position: left
##     range: <ggproto object: Class RangeDiscrete, Range, gg>
##         range: NULL
##         reset: function
##         train: function
##         super:  <ggproto object: Class RangeDiscrete, Range, gg>
##     rescale: function
##     reset: function
##     scale_name: manual
##     train: function
##     train_df: function
##     transform: function
##     transform_df: function
##     super:  <ggproto object: Class ScaleDiscrete, Scale, gg>
```

Data Wrangling / Munging / Carpentry

Throughout this assignment you can use either the `tidyverse` package suite or `data.table` to answer but not base R. You can mix `data.table` with `magrittr` piping if you wish but don't go back and forth between `tbl_df`'s and `data.table` objects.

```
pacman::p_load(dplyr, magrittr, data.table)
```

Load the `storms` dataset from the `dplyr` package and investigate it using `str` and `summary` and `head`. Which two columns should be converted to type factor? Do so below.

```
data(storms)
str(storms)
```

```
## tibble [10,010 x 13] (S3: tbl_df/tbl/data.frame)
##  $ name       : chr [1:10010] "Amy" "Amy" "Amy" "Amy" ...
##  $ year       : num [1:10010] 1975 1975 1975 1975 1975 ...
##  $ month      : num [1:10010] 6 6 6 6 6 6 6 6 6 6 ...
##  $ day        : int [1:10010] 27 27 27 27 28 28 28 28 29 29 ...
##  $ hour       : num [1:10010] 0 6 12 18 0 6 12 18 0 6 ...
##  $ lat        : num [1:10010] 27.5 28.5 29.5 30.5 31.5 32.4 33.3 34 34.4 34 ...
##  $ long       : num [1:10010] -79 -79 -79 -79 -78.8 -78.7 -78 -77 -75.8 -74.8 ...
##  $ status     : chr [1:10010] "tropical depression" "tropical depression" "tropical depression" "trop
##  $ category   : Ord.factor w/ 7 levels "-1"<"0"<"1"<"2"<..: 1 1 1 1 1 1 1 1 2 2 ...
##  $ wind       : int [1:10010] 25 25 25 25 25 25 25 30 35 40 ...
##  $ pressure   : int [1:10010] 1013 1013 1013 1013 1012 1012 1011 1006 1004 1002 ...
##  $ ts_diameter: num [1:10010] NA NA NA NA NA NA NA NA NA NA ...
##  $ hu_diameter: num [1:10010] NA NA NA NA NA NA NA NA NA NA ...
```

```
head(storms)
```

```
## # A tibble: 6 x 13
##   name   year month   day  hour   lat  long status       category  wind pressure
##   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>        <ord>    <int>    <int>
## 1 Amy    1975     6    27     0  27.5 -79   tropical de~ -1          25     1013
## 2 Amy    1975     6    27     6  28.5 -79   tropical de~ -1          25     1013
## 3 Amy    1975     6    27    12  29.5 -79   tropical de~ -1          25     1013
## 4 Amy    1975     6    27    18  30.5 -79   tropical de~ -1          25     1013
## 5 Amy    1975     6    28     0  31.5 -78.8 tropical de~ -1          25     1012
## 6 Amy    1975     6    28     6  32.4 -78.7 tropical de~ -1          25     1012
## # ... with 2 more variables: ts_diameter <dbl>, hu_diameter <dbl>
```

Reorder the columns so name is first, status is second, category is third and the rest are the same.

```
storms%>%
  select(name,status,category,everything())
```

```
## # A tibble: 10,010 x 13
##    name  status       category  year month   day  hour   lat  long  wind pressure
##    <chr> <chr>        <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl> <int>    <int>
##  1 Amy   tropical d~ -1         1975     6    27     0  27.5 -79      25     1013
##  2 Amy   tropical d~ -1         1975     6    27     6  28.5 -79      25     1013
##  3 Amy   tropical d~ -1         1975     6    27    12  29.5 -79      25     1013
##  4 Amy   tropical d~ -1         1975     6    27    18  30.5 -79      25     1013
##  5 Amy   tropical d~ -1         1975     6    28     0  31.5 -78.8    25     1012
##  6 Amy   tropical d~ -1         1975     6    28     6  32.4 -78.7    25     1012
##  7 Amy   tropical d~ -1         1975     6    28    12  33.3 -78      25     1011
##  8 Amy   tropical d~ -1         1975     6    28    18  34   -77      30     1006
##  9 Amy   tropical s~ 0          1975     6    29     0  34.4 -75.8    35     1004
## 10 Amy   tropical s~ 0          1975     6    29     6  34   -74.8    40     1002
## # ... with 10,000 more rows, and 2 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>
```

Find a subset of the data of storms only in the 1970's.

```
storms %>%
  filter(year >= 1970 & year <= 1979)
```

```
## # A tibble: 546 x 13
##    name   year month   day  hour   lat  long status      category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical d~ -1          25     1013
##  2 Amy    1975     6    27     6  28.5 -79   tropical d~ -1          25     1013
##  3 Amy    1975     6    27    12  29.5 -79   tropical d~ -1          25     1013
##  4 Amy    1975     6    27    18  30.5 -79   tropical d~ -1          25     1013
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical d~ -1          25     1012
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical d~ -1          25     1012
##  7 Amy    1975     6    28    12  33.3 -78   tropical d~ -1          25     1011
##  8 Amy    1975     6    28    18  34   -77   tropical d~ -1          30     1006
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical s~ 0           35     1004
## 10 Amy    1975     6    29     6  34   -74.8 tropical s~ 0           40     1002
## # ... with 536 more rows, and 2 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>
```

Find a subset of the data of storm observations only with category 4 and above and wind speed 100MPH and above.

```
storms%>%
  filter(category >= 4 & wind >= 100)
```

```
## # A tibble: 416 x 13
##    name   year month   day  hour   lat  long status    category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>     <ord>    <int>    <int>
##  1 Anita  1977     9     2     0  24.6 -96.2 hurricane 5          140      931
##  2 Anita  1977     9     2     6  24.2 -97.1 hurricane 5          150      926
##  3 Anita  1977     9     2    12  23.7 -98   hurricane 4          120      940
##  4 David  1979     8    28     0  12.2 -52.9 hurricane 4          115      947
##  5 David  1979     8    28     6  12.5 -54.4 hurricane 4          125      941
##  6 David  1979     8    28    12  12.8 -55.7 hurricane 4          130      938
##  7 David  1979     8    28    18  13.2 -56.9 hurricane 4          125      941
##  8 David  1979     8    29     0  13.7 -58   hurricane 4          120      944
##  9 David  1979     8    29     6  14.2 -59.2 hurricane 4          120      942
## 10 David  1979     8    29    12  14.8 -60.3 hurricane 4          125      938
## # ... with 406 more rows, and 2 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>
```

Create a new feature `wind_speed_per_unit_pressure`.

```
storms %>%
  mutate(wind_speed_per_unit_pressure = wind / pressure)
```

```
## # A tibble: 10,010 x 14
##    name   year month   day  hour   lat  long status      category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical d~ -1          25     1013
##  2 Amy    1975     6    27     6  28.5 -79   tropical d~ -1          25     1013
##  3 Amy    1975     6    27    12  29.5 -79   tropical d~ -1          25     1013
##  4 Amy    1975     6    27    18  30.5 -79   tropical d~ -1          25     1013
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical d~ -1          25     1012
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical d~ -1          25     1012
##  7 Amy    1975     6    28    12  33.3 -78   tropical d~ -1          25     1011
##  8 Amy    1975     6    28    18  34   -77   tropical d~ -1          30     1006
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical s~ 0           35     1004
## 10 Amy    1975     6    29     6  34   -74.8 tropical s~ 0           40     1002
## # ... with 10,000 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, wind_speed_per_unit_pressure <dbl>
```

Create a new feature: `average_diameter` which averages the two diameter metrics. If one is missing, then use the value of the one that is present. If both are missing, leave missing.

```
storms %>%
  rowwise() %>%
  arrange(desc(year)) %>%
  mutate(average_diameter = if_else(!is.na(ts_diameter) & !is.na(hu_diameter),mean(c(ts_diameter,hu_diam
```

```
## # A tibble: 10,010 x 14
## # Rowwise:
##    name   year month   day  hour   lat  long status      category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Ana    2015     5     9     6  32.2 -77.5 tropical s~ 0           50      998
##  2 Ana    2015     5     9    12  32.5 -77.8 tropical s~ 0           50     1001
##  3 Ana    2015     5     9    18  32.7 -78   tropical s~ 0           45     1001
##  4 Ana    2015     5    10     0  33.1 -78.3 tropical s~ 0           45     1001
##  5 Ana    2015     5    10     6  33.5 -78.6 tropical s~ 0           40     1002
##  6 Ana    2015     5    10    10  33.8 -78.8 tropical s~ 0           40     1002
##  7 Ana    2015     5    10    12  33.9 -78.8 tropical s~ 0           35     1002
##  8 Ana    2015     5    10    18  34.3 -78.7 tropical d~ -1          30     1006
##  9 Ana    2015     5    11     0  34.7 -78.5 tropical d~ -1          30     1009
## 10 Ana    2015     5    11     6  35.5 -78   tropical d~ -1          30     1010
## # ... with 10,000 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, average_diameter <dbl>
```

For each storm, summarize the maximum wind speed. “Summarize” means create a new dataframe with only the summary metrics you care about.

```
storms %>%
  group_by(name) %>%
  summarise(max_wind_speed = max(wind, na.rm = TRUE))
```

```
## # A tibble: 198 x 2
##    name     max_wind_speed
##  * <chr>             <int>
##  1 AL011993             30
##  2 AL012000             25
##  3 AL021992             30
##  4 AL021994             30
##  5 AL021999             30
##  6 AL022000             30
##  7 AL022001             25
##  8 AL022003             30
##  9 AL022006             45
## 10 AL031987             40
## # ... with 188 more rows
```

Order your dataset by maximum wind speed storm but within the rows of storm show the observations in time order from early to late.

```
storms %>%
  group_by(name) %>%
  mutate(max_wind_by_storm = max(wind, na.rm =TRUE)) %>%
  select(name, max_wind_by_storm, everything()) %>%
  arrange(desc(max_wind_by_storm), year, month, day, hour)
```

```
## # A tibble: 10,010 x 14
## # Groups:   name [198]
##    name   max_wind_by_sto~  year month   day  hour   lat  long status   category
##    <chr>             <int> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>    <ord>
##  1 Gilbe~              160  1988     9     8    18  12   -54   tropica~ -1
##  2 Gilbe~              160  1988     9     9     0  12.7 -55.6 tropica~ -1
##  3 Gilbe~              160  1988     9     9     6  13.3 -57.1 tropica~ -1
##  4 Gilbe~              160  1988     9     9    12  14   -58.6 tropica~ -1
##  5 Gilbe~              160  1988     9     9    18  14.5 -60.1 tropica~ 0
##  6 Gilbe~              160  1988     9    10     0  14.8 -61.5 tropica~ 0
##  7 Gilbe~              160  1988     9    10     6  15   -62.8 tropica~ 0
##  8 Gilbe~              160  1988     9    10    12  15.3 -64.1 tropica~ 0
##  9 Gilbe~              160  1988     9    10    18  15.7 -65.4 tropica~ 0
## 10 Gilbe~              160  1988     9    11     0  15.9 -66.8 hurrica~ 1
## # ... with 10,000 more rows, and 4 more variables: wind <int>, pressure <int>,
## #   ts_diameter <dbl>, hu_diameter <dbl>
```

Find the strongest storm by wind speed per year.

```
storms %>%
  group_by(year) %>%
  arrange(year, desc(wind)) %>%
  slice(1) %>%
  select(name, year)
```

```
## # A tibble: 41 x 2
## # Groups:   year [41]
##    name      year
##    <chr>    <dbl>
##  1 Caroline  1975
##  2 Belle     1976
##  3 Anita     1977
##  4 Cora      1978
##  5 David     1979
##  6 Ivan      1980
##  7 Harvey    1981
##  8 Debby     1982
##  9 Alicia    1983
## 10 Diana     1984
## # ... with 31 more rows
```

For each named storm, find its maximum category, wind speed, pressure and diameters. Do not allow the max to be NA (unless all the measurements for that storm were NA).

```
storms %>%
  group_by(name) %>%
  summarise(max_wind_speed = max(wind, na.rm = TRUE), max_category = max(category, na.rm =TRUE), max_pr
```

```
## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(ts_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf
```

```
## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## Warning in max(hu_diameter, na.rm = TRUE): no non-missing arguments to max;
## returning -Inf

## # A tibble: 198 x 6
##    name     max_wind_speed max_category max_pressure max_diameter_ts
##  * <chr>             <int> <ord>               <int>           <dbl>
##  1 AL011993             30 -1                   1003          -Inf
##  2 AL012000             25 -1                   1010          -Inf
##  3 AL021992             30 -1                   1009          -Inf
##  4 AL021994             30 -1                   1017          -Inf
##  5 AL021999             30 -1                   1006          -Inf
##  6 AL022000             30 -1                   1010          -Inf
##  7 AL022001             25 -1                   1012          -Inf
##  8 AL022003             30 -1                   1010          -Inf
##  9 AL022006             45 0                    1008            69.0
## 10 AL031987             40 0                    1015          -Inf
## # ... with 188 more rows, and 1 more variable: max_diameter_hu <dbl>
```

For each year in the dataset, tally the number of storms. “Tally” is a fancy word for “count the number of”. Plot the number of storms by year. Any pattern?

```
storms %>%
  group_by(year) %>%
  distinct(name) %>%
  count(year)
```

```
## # A tibble: 41 x 2
## # Groups:   year [41]
##     year     n
##    <dbl> <int>
##  1  1975     3
##  2  1976     2
##  3  1977     3
##  4  1978     4
##  5  1979     7
##  6  1980     8
##  7  1981     5
##  8  1982     5
##  9  1983     4
## 10  1984    10
## # ... with 31 more rows
```

For each year in the dataset, tally the storms by category.

```
storms %>%
  group_by(year,category) %>%
  distinct(name) %>%
  count(category)
```

```
## # A tibble: 233 x 3
## # Groups:   year, category [233]
##     year category     n
##    <dbl> <ord>    <int>
##  1  1975 -1           2
##  2  1975 0            3
##  3  1975 1            2
##  4  1975 2            2
##  5  1975 3            1
##  6  1976 -1           2
##  7  1976 0            2
##  8  1976 1            2
##  9  1976 2            2
## 10  1976 3            1
## # ... with 223 more rows
```

For each year in the dataset, find the maximum wind speed per status level.

```
storms %>%
  group_by(year, status) %>%
  summarise(max_wind_speed = max(wind, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.

## # A tibble: 123 x 3
## # Groups:   year [41]
##     year status              max_wind_speed
##    <dbl> <chr>                        <int>
##  1  1975 hurricane                      100
##  2  1975 tropical depression             30
##  3  1975 tropical storm                  60
##  4  1976 hurricane                      105
##  5  1976 tropical depression             30
##  6  1976 tropical storm                  60
##  7  1977 hurricane                      150
##  8  1977 tropical depression             30
##  9  1977 tropical storm                  60
## 10  1978 hurricane                       80
## # ... with 113 more rows
```

For each storm, summarize its average location in latitude / longitude coordinates.

```
storms %>%
  rowwise() %>%
  group_by(year, name) %>%
  summarise(average_lat = mean(lat, na.rm = TRUE), average_long = mean(long, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

```
## # A tibble: 426 x 4
## # Groups:   year [41]
##     year name     average_lat average_long
##    <dbl> <chr>          <dbl>        <dbl>
##  1  1975 Amy             35.4        -69.8
##  2  1975 Caroline        22.5        -87.6
##  3  1975 Doris           36.5        -45.3
##  4  1976 Belle           30.6        -74.0
##  5  1976 Gloria          31.9        -53.9
##  6  1977 Anita           25.4        -93.8
##  7  1977 Clara           34.2        -69.8
##  8  1977 Evelyn          36.7        -62.8
##  9  1978 Amelia          27.5        -98.0
## 10  1978 Bess            23.0        -94.4
## # ... with 416 more rows
```

For each storm, summarize its duration in number of hours (to the nearest 6hr increment).

```
storms %>%
  group_by(name) %>%
  count() %>%
  summarise(duration = n*6)
```

```
## # A tibble: 198 x 2
##    name     duration
##  * <chr>       <dbl>
##  1 AL011993       48
##  2 AL012000       24
##  3 AL021992       30
##  4 AL021994       36
##  5 AL021999       24
##  6 AL022000       72
##  7 AL022001       30
##  8 AL022003       24
##  9 AL022006       30
## 10 AL031987      192
## # ... with 188 more rows
```

For storm in a category, create a variable `storm_number` that enumerates the storms 1, 2, ... (in date order).

```
storms %>%
  group_by(category) %>%
  arrange(year, month, day, hour) %>%
  summarise(storm_number = row_number(category))
```

```
## `summarise()` has grouped output by 'category'. You can override using the `.groups` argument.
```

```
## # A tibble: 10,010 x 2
## # Groups:   category [7]
##    category storm_number
##    <ord>           <int>
##  1 -1                  1
##  2 -1                  2
##  3 -1                  3
##  4 -1                  4
##  5 -1                  5
##  6 -1                  6
##  7 -1                  7
##  8 -1                  8
##  9 -1                  9
## 10 -1                 10
## # ... with 10,000 more rows
```

Convert year, month, day, hour into the variable `timestamp` using the `lubridate` package. Although the new package `clock` just came out, `lubridate` still seems to be standard. Next year I'll probably switch the class to be using `clock`.

```
pacman::p_load("lubridate")

storms %>%
  mutate(timestamp = ymd_h(paste(year,month,day,hour, sep ="-")))
```

```
## # A tibble: 10,010 x 14
##    name   year month   day  hour   lat  long status      category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical d~ -1          25     1013
##  2 Amy    1975     6    27     6  28.5 -79   tropical d~ -1          25     1013
##  3 Amy    1975     6    27    12  29.5 -79   tropical d~ -1          25     1013
##  4 Amy    1975     6    27    18  30.5 -79   tropical d~ -1          25     1013
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical d~ -1          25     1012
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical d~ -1          25     1012
##  7 Amy    1975     6    28    12  33.3 -78   tropical d~ -1          25     1011
##  8 Amy    1975     6    28    18  34   -77   tropical d~ -1          30     1006
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical s~ 0           35     1004
## 10 Amy    1975     6    29     6  34   -74.8 tropical s~ 0           40     1002
## # ... with 10,000 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, timestamp <dttm>
```

Using the `lubridate` package, create new variables `day_of_week` which is a factor with levels "Sunday", "Monday", ... "Saturday" and `week_of_year` which is integer 1, 2, ..., 52.

```
storms %>%
  mutate(day_of_week = weekdays(ymd_h(paste(year, month, day, hour, sep='.'))))
```

```
## # A tibble: 10,010 x 14
##    name   year month   day  hour   lat  long status      category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical d~ -1          25     1013
##  2 Amy    1975     6    27     6  28.5 -79   tropical d~ -1          25     1013
##  3 Amy    1975     6    27    12  29.5 -79   tropical d~ -1          25     1013
##  4 Amy    1975     6    27    18  30.5 -79   tropical d~ -1          25     1013
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical d~ -1          25     1012
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical d~ -1           25     1012
##  7 Amy    1975     6    28    12  33.3 -78   tropical d~ -1           25     1011
##  8 Amy    1975     6    28    18  34   -77   tropical d~ -1           30     1006
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical s~ 0            35     1004
## 10 Amy    1975     6    29     6  34   -74.8 tropical s~ 0            40     1002
## # ... with 10,000 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, day_of_week <chr>
```

For each storm, summarize the day in which is started in the following format "Friday, June 27, 1975".

```
storms %>%
  group_by(name) %>%
  mutate(timestamp= ymd_h(paste(year, month, day, hour, sep ='.'))) %>%
  arrange(timestamp) %>%
  slice(1) %>%
  mutate(timestamp = paste(weekdays(timestamp), ', ', months(timestamp), ' ', day(timestamp), ', ', yea
```

```
## # A tibble: 198 x 14
## # Groups:   name [198]
##    name    year month   day  hour   lat  long status     category  wind pressure
##    <chr>  <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>      <ord>    <int>    <int>
##  1 AL011~  1993     5    31    12  21.5 -84   tropical ~ -1          25     1003
##  2 AL012~  2000     6     7    18  21   -93   tropical ~ -1          25     1008
##  3 AL021~  1992     6    25    12  24.5 -85.5 tropical ~ -1          25     1009
##  4 AL021~  1994     7    20     6  32.2 -78.9 tropical ~ -1          25     1017
##  5 AL021~  1999     7     2    18  20.2 -95   tropical ~ -1          30     1006
##  6 AL022~  2000     6    23     0   9.5 -19.8 tropical ~ -1          25     1010
##  7 AL022~  2001     7    11    18  10.9 -42.1 tropical ~ -1          25     1011
##  8 AL022~  2003     6    11     0   9.5 -40.8 tropical ~ -1          30     1009
##  9 AL022~  2006     7    17     6  39.1 -66.4 tropical ~ -1          30     1008
## 10 AL031~  1987     8     9    12  26.3 -93.6 tropical ~ -1          30     1010
## # ... with 188 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, timestamp <chr>
```

Create a new factor variable `decile_windspeed` by binning wind speed into 10 bins.

```
storms %>%
  mutate(decile_windspeed = ntile(wind, 10))
```

```
## # A tibble: 10,010 x 14
##    name    year month   day  hour   lat  long status      category  wind pressure
##    <chr>  <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical d~ -1           25     1013
##  2 Amy    1975     6    27     6  28.5 -79   tropical d~ -1           25     1013
##  3 Amy    1975     6    27    12  29.5 -79   tropical d~ -1           25     1013
##  4 Amy    1975     6    27    18  30.5 -79   tropical d~ -1           25     1013
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical d~ -1           25     1012
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical d~ -1           25     1012
##  7 Amy    1975     6    28    12  33.3 -78   tropical d~ -1           25     1011
##  8 Amy    1975     6    28    18  34   -77   tropical d~ -1           30     1006
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical s~ 0            35     1004
## 10 Amy    1975     6    29     6  34   -74.8 tropical s~ 0            40     1002
## # ... with 10,000 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, decile_windspeed <int>
```

Create a new data frame `serious_storms` which are category 3 and above hurricanes.

```
serious_storms = storms %>% filter(category >= 3)
serious_storms
```

```
## # A tibble: 779 x 13
##    name      year month   day  hour   lat  long status   category  wind pressure
##    <chr>    <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>    <ord>    <int>    <int>
##  1 Caroline  1975     8    31     0  24   -97   hurrica~ 3          100      973
##  2 Caroline  1975     8    31     6  24.1 -97.5 hurrica~ 3          100      963
##  3 Belle     1976     8     8    18  29.5 -75.3 hurrica~ 3          100      958
##  4 Belle     1976     8     9     0  30.9 -75.3 hurrica~ 3          105      957
##  5 Belle     1976     8     9     6  32.5 -75.2 hurrica~ 3          105      959
##  6 Anita     1977     9     1    18  25.2 -95.5 hurrica~ 3          110      945
##  7 Anita     1977     9     2     0  24.6 -96.2 hurrica~ 5          140      931
##  8 Anita     1977     9     2     6  24.2 -97.1 hurrica~ 5          150      926
##  9 Anita     1977     9     2    12  23.7 -98   hurrica~ 4          120      940
## 10 David     1979     8    28     0  12.2 -52.9 hurrica~ 4          115      947
## # ... with 769 more rows, and 2 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>
```

In `serious_storms`, merge the variables lat and long together into `lat_long` with values `lat / long` as a string.

```
serious_storms$lat_long = paste(serious_storms$lat, serious_storms$long, sep = "/")

serious_storms$lat <- NULL
serious_storms$long <- NULL

serious_storms
```

```
## # A tibble: 779 x 12
##    name      year month   day  hour status   category  wind pressure ts_diameter
##    <chr>    <dbl> <dbl> <int> <dbl> <chr>    <ord>    <int>    <int>       <dbl>
##  1 Caroline  1975     8    31     0 hurrica~ 3          100      973          NA
##  2 Caroline  1975     8    31     6 hurrica~ 3          100      963          NA
##  3 Belle     1976     8     8    18 hurrica~ 3          100      958          NA
##  4 Belle     1976     8     9     0 hurrica~ 3          105      957          NA
##  5 Belle     1976     8     9     6 hurrica~ 3          105      959          NA
##  6 Anita     1977     9     1    18 hurrica~ 3          110      945          NA
##  7 Anita     1977     9     2     0 hurrica~ 5          140      931          NA
##  8 Anita     1977     9     2     6 hurrica~ 5          150      926          NA
##  9 Anita     1977     9     2    12 hurrica~ 4          120      940          NA
## 10 David     1979     8    28     0 hurrica~ 4          115      947          NA
## # ... with 769 more rows, and 2 more variables: hu_diameter <dbl>,
## #   lat_long <chr>
```

Let’s return now to the original storms data frame. For each category, find the average wind speed, pressure and diameters (do not count the NA’s in your averaging).

```
storms %>%
  group_by(category) %>%
  summarise(average_wind_speed = mean(wind, na.rm=TRUE), average_pressure = mean(pressure, na.rm=TRUE),
## # A tibble: 7 x 4
##   category average_wind_speed average_pressure average_diameters
## * <ord>                 <dbl>            <dbl>             <dbl>
## 1 -1                     27.3            1008.               0
## 2 0                      45.8             999.              79.8
## 3 1                      70.9             982.             168.
## 4 2                      89.4             967.             180.
## 5 3                     105.              954.             199.
## 6 4                     122.              940.             209.
## 7 5                     145.              916.             219.
```

For each named storm, find its maximum category, wind speed, pressure and diameters (do not allow the max to be NA) and the number of readings (i.e. observations).

```
storms %>%
  group_by(name) %>%
  summarise(max_category = max(category, na.rm=TRUE), max_wind_speed = max(wind, na.rm=TRUE), max_pressu
```

```
## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf
## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf
## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf
## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf
## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf

## Warning in max(c(ts_diameter, hu_diameter), na.rm = TRUE): no non-missing
## arguments to max; returning -Inf
## `summarise()` has grouped output by 'name'. You can override using the `.groups` argument.
```

```
## # A tibble: 39,204 x 6
## # Groups:   name [198]
##    name     max_category max_wind_speed max_pressure max_diameters num_reading
##    <chr>    <ord>                 <int>        <int>         <dbl>       <int>
##  1 AL011993 -1                       30         1003          -Inf           8
##  2 AL011993 -1                       30         1003          -Inf           4
##  3 AL011993 -1                       30         1003          -Inf           5
##  4 AL011993 -1                       30         1003          -Inf           6
##  5 AL011993 -1                       30         1003          -Inf           4
##  6 AL011993 -1                       30         1003          -Inf          12
##  7 AL011993 -1                       30         1003          -Inf           5
##  8 AL011993 -1                       30         1003          -Inf           4
##  9 AL011993 -1                       30         1003          -Inf           5
## 10 AL011993 -1                       30         1003          -Inf          32
## # ... with 39,194 more rows
```

Calculate the distance from each storm observation to Miami in a new variable `distance_to_miami`. This is very challenging. You will need a function that computes distances from two sets of latitude / longitude coordinates.

```
MIAMI_LAT_LONG_COORDS = c(25.7617, -80.1918)
 getDistanceFromLatLonInKm <- function (lat1,lon1,lat2,lon2) {
     R = 6371 # Radius of the earth in km
     dLat = deg2rad(lat2-lat1)  # deg2rad below
     dLon = deg2rad(lon2-lon1)
     a = sin(dLat/2) * sin(dLat/2) +
     cos(deg2rad(lat1)) * cos(deg2rad(lat2)) *
     sin(dLon/2) * sin(dLon/2)
     c = 2 * atan2(sqrt(a), sqrt(1-a))
     d = R * c # Distance in km
     d
  }

deg2rad <- function (deg) {
 deg * (pi/180)
}

storms %>%
  mutate(distance_to_miami = getDistanceFromLatLonInKm(MIAMI_LAT_LONG_COORDS[1],MIAMI_LAT_LONG_COORDS[2
```

```
## # A tibble: 10,010 x 14
##    name   year month   day  hour   lat  long status      category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical d~ -1          25     1013
##  2 Amy    1975     6    27     6  28.5 -79   tropical d~ -1          25     1013
##  3 Amy    1975     6    27    12  29.5 -79   tropical d~ -1          25     1013
##  4 Amy    1975     6    27    18  30.5 -79   tropical d~ -1          25     1013
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical d~ -1          25     1012
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical d~ -1          25     1012
##  7 Amy    1975     6    28    12  33.3 -78   tropical d~ -1          25     1011
##  8 Amy    1975     6    28    18  34   -77   tropical d~ -1          30     1006
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical s~ 0           35     1004
## 10 Amy    1975     6    29     6  34   -74.8 tropical s~ 0           40     1002
## # ... with 10,000 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, distance_to_miami <dbl>
```

For each storm observation, use the function from the previous question to calculate the distance it moved since the previous observation.

```
storms %>%
  group_by(name) %>%
   mutate(distance_moved = getDistanceFromLatLonInKm(last(lat),last(long), lat, long))
```

```
## # A tibble: 10,010 x 14
## # Groups:   name [198]
##    name   year month   day  hour   lat  long status      category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical d~ -1          25     1013
##  2 Amy    1975     6    27     6  28.5 -79   tropical d~ -1          25     1013
##  3 Amy    1975     6    27    12  29.5 -79   tropical d~ -1          25     1013
##  4 Amy    1975     6    27    18  30.5 -79   tropical d~ -1          25     1013
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical d~ -1          25     1012
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical d~ -1          25     1012
##  7 Amy    1975     6    28    12  33.3 -78   tropical d~ -1          25     1011
##  8 Amy    1975     6    28    18  34   -77   tropical d~ -1          30     1006
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical s~ 0           35     1004
## 10 Amy    1975     6    29     6  34   -74.8 tropical s~ 0           40     1002
## # ... with 10,000 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, distance_moved <dbl>
```

For each storm, find the total distance it moved over its observations and its total displacement. “Distance” is a scalar quantity that refers to “how much ground an object has covered” during its motion. “Displacement” is a vector quantity that refers to “how far out of place an object is”; it is the object’s overall change in position.

```
storms %>%
  group_by(name) %>%
  mutate(total_distance = getDistanceFromLatLonInKm(first(lat),first(long), last(lat), last(long))) %>%
  mutate(total_displacement = paste( last(lat) - first(lat), last(long) - first(long),sep ="/"))
```

```
## # A tibble: 10,010 x 15
## # Groups:   name [198]
##    name   year month   day  hour   lat  long status      category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical d~ -1          25     1013
##  2 Amy    1975     6    27     6  28.5 -79   tropical d~ -1          25     1013
##  3 Amy    1975     6    27    12  29.5 -79   tropical d~ -1          25     1013
##  4 Amy    1975     6    27    18  30.5 -79   tropical d~ -1          25     1013
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical d~ -1          25     1012
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical d~ -1          25     1012
##  7 Amy    1975     6    28    12  33.3 -78   tropical d~ -1          25     1011
##  8 Amy    1975     6    28    18  34   -77   tropical d~ -1          30     1006
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical s~ 0           35     1004
## 10 Amy    1975     6    29     6  34   -74.8 tropical s~ 0           40     1002
## # ... with 10,000 more rows, and 4 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, total_distance <dbl>, total_displacement <chr>
```

For each storm observation, calculate the average speed the storm moved in location.

```
storms %>%
  group_by(name) %>%
  mutate(average_speed_moved = getDistanceFromLatLonInKm(last(lat),last(long), lat, long)/6)
```

```
## # A tibble: 10,010 x 14
## # Groups:   name [198]
##    name   year month   day  hour   lat  long status      category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical d~ -1          25     1013
##  2 Amy    1975     6    27     6  28.5 -79   tropical d~ -1          25     1013
##  3 Amy    1975     6    27    12  29.5 -79   tropical d~ -1          25     1013
##  4 Amy    1975     6    27    18  30.5 -79   tropical d~ -1          25     1013
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical d~ -1          25     1012
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical d~ -1          25     1012
##  7 Amy    1975     6    28    12  33.3 -78   tropical d~ -1          25     1011
##  8 Amy    1975     6    28    18  34   -77   tropical d~ -1          30     1006
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical s~ 0           35     1004
## 10 Amy    1975     6    29     6  34   -74.8 tropical s~ 0           40     1002
## # ... with 10,000 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, average_speed_moved <dbl>
```

For each storm, calculate its average ground speed (how fast its eye is moving which is different from windspeed around the eye).

```
ground_speed <- storms %>%
  group_by(name) %>%
  mutate(avg_ground_speed = getDistanceFromLatLonInKm(first(lat),first(long), last(lat), last(long))/6)

ground_speed
```

```
## # A tibble: 10,010 x 14
## # Groups:   name [198]
##    name   year month   day  hour   lat  long status      category  wind pressure
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>       <ord>    <int>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical d~ -1          25     1013
##  2 Amy    1975     6    27     6  28.5 -79   tropical d~ -1          25     1013
##  3 Amy    1975     6    27    12  29.5 -79   tropical d~ -1          25     1013
##  4 Amy    1975     6    27    18  30.5 -79   tropical d~ -1          25     1013
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical d~ -1          25     1012
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical d~ -1          25     1012
##  7 Amy    1975     6    28    12  33.3 -78   tropical d~ -1          25     1011
##  8 Amy    1975     6    28    18  34   -77   tropical d~ -1          30     1006
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical s~ 0           35     1004
## 10 Amy    1975     6    29     6  34   -74.8 tropical s~ 0           40     1002
## # ... with 10,000 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, avg_ground_speed <dbl>
```

Is there a relationship between average ground speed and maximum category attained? Use a dataframe summary (not a regression).

```
summary(data.frame(ground_speed))
```

```
##      name                year          month             day
##  Length:10010       Min.   :1975   Min.   : 1.000   Min.   : 1.00
##  Class :character   1st Qu.:1990   1st Qu.: 8.000   1st Qu.: 8.00
##  Mode  :character   Median :1999   Median : 9.000   Median :16.00
##                     Mean   :1998   Mean   : 8.779   Mean   :15.86
##                     3rd Qu.:2006   3rd Qu.: 9.000   3rd Qu.:24.00
##                     Max.   :2015   Max.   :12.000   Max.   :31.00
##
##       hour             lat             long            status
##  Min.   : 0.000   Min.   : 7.20   Min.   :-109.30   Length:10010
##  1st Qu.: 6.000   1st Qu.:17.50   1st Qu.: -80.70   Class :character
##  Median :12.000   Median :24.40   Median : -64.50   Mode  :character
##  Mean   : 9.114   Mean   :24.76   Mean   : -64.23
##  3rd Qu.:18.000   3rd Qu.:31.30   3rd Qu.: -48.60
##  Max.   :23.000   Max.   :51.90   Max.   :  -6.00
##
##  category       wind           pressure       ts_diameter       hu_diameter
##  -1:2545   Min.   : 10.00   Min.   : 882.0   Min.   :   0.00   Min.   :  0.00
##  0 :4373   1st Qu.: 30.00   1st Qu.: 985.0   1st Qu.:  69.05   1st Qu.:  0.00
##  1 :1685   Median : 45.00   Median : 999.0   Median : 138.09   Median :  0.00
##  2 : 628   Mean   : 53.49   Mean   : 992.1   Mean   : 166.76   Mean   : 21.41
##  3 : 363   3rd Qu.: 65.00   3rd Qu.:1006.0   3rd Qu.: 241.66   3rd Qu.: 28.77
##  4 : 348   Max.   :160.00   Max.   :1022.0   Max.   :1001.18   Max.   :345.23
##  5 :  68                                     NA's   :6528      NA's   :6528
##  avg_ground_speed
##  Min.   :   7.624
##  1st Qu.: 317.493
##  Median : 478.746
##  Mean   : 531.635
##  3rd Qu.: 723.792
##  Max.   :1358.166
##
```

```
#Yes, the higher the category, the higher the average ground speed.
```

Now we want to transition to building real design matrices for prediction. This is more in tune with what happens in the real world. Large data dump and you convert it into X and y how you see fit.

Suppose we wish to predict the following: given the first three readings of a storm, can you predict its maximum wind speed? Identify the `y` and identify which features you need $x_1, ... x_p$ and build that matrix with `dplyr` functions. This is not easy, but it is what it's all about. Feel free to "featurize" as creatively as you would like. You aren't going to overfit if you only build a few features relative to the total 198 storms.

```
#TO-DO

# K = 5
# test_prop = 1 / K
# train_indices = sample(1 : nrow(adult), round((1 - test_prop) * nrow(adult)))
# adult_train = adult[train_indices, ]
# y_train = adult_train$income
# X_train = adult_train
# X_train$income = NULL
# test_indices = setdiff(1 : nrow(adult), train_indices)
# adult_test = adult[test_indices, ]
# y_test = adult_test$income
# X_test = adult_test
# X_test$income = NULL

data_original = storms %>%
  group_by(name)%>%
  mutate(max_wind_speed = max(wind, na.rm = TRUE))
data = data_original %>% filter(n()>=3) %>% slice(1:3)

K = 5
test_prop = 1 / K
train_indices = sample(1 : nrow(data), round((1 - test_prop) * nrow(data)))
storm_train = data %>% slice(train_indices)
y_train = (storm_train %>% select(max_wind_speed))$max_wind_speed
```

```
## Adding missing grouping variables: `name`
```

```
X_train = storm_train %>% select(-max_wind_speed)
test_indices = setdiff(1 : nrow(data), train_indices)
storm_test = data %>% slice(test_indices)
y_test = (storm_test %>% select(max_wind_speed))$max_wind_speed
```

```
## Adding missing grouping variables: `name`
```

```
X_test = storm_test %>% select(-max_wind_speed)
```

Fit your model. Validate it.

```
#TO-DO
class(X_train)
```

```
## [1] "grouped_df" "tbl_df"     "tbl"        "data.frame"
```

```
X_train
```

```
## # A tibble: 591 x 13
## # Groups:   name [197]
##    name    year month   day  hour   lat  long status     category  wind pressure
##    <chr>  <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>      <ord>    <int>    <int>
##  1 AL011~  1993     5    31    12  21.5 -84   tropical ~ -1          25     1003
##  2 AL011~  1993     6     1     0  23.2 -80.3 tropical ~ -1          25     1000
##  3 AL011~  1993     5    31    18  22.3 -82   tropical ~ -1          25     1002
##  4 AL012~  2000     6     7    18  21    -93   tropical ~ -1          25     1008
##  5 AL012~  2000     6     8     6  20.7 -93.1 tropical ~ -1          25     1010
##  6 AL012~  2000     6     8     0  20.9 -92.8 tropical ~ -1          25     1009
##  7 AL021~  1992     6    25    12  24.5 -85.5 tropical ~ -1          25     1009
##  8 AL021~  1992     6    26     0  27   -84.5 tropical ~ -1          30     1007
##  9 AL021~  1992     6    25    18  25.7 -85.5 tropical ~ -1          30     1007
## 10 AL021~  1994     7    20     6  32.2 -78.9 tropical ~ -1          25     1017
## # ... with 581 more rows, and 2 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>
```

```
OLS = lm(y_train ~ ., data.frame(X_train))
y_pred = predict(OLS, X_test)
```

```
## Warning in predict.lm(OLS, X_test): prediction from a rank-deficient fit may be
## misleading
```

```
y_pred
```

```
## numeric(0)
```

Assess your level of success at this endeavor.

#TO-DO

The Forward Stepwise Procedure for Probability Estimation Models

Set a seed and load the adult dataset and remove missingness and randomize the order.

```
set.seed(1)
pacman::p_load_gh("coatless/ucidata")
data(adult)
adult = na.omit(adult)
adult = adult[sample(1 : nrow(adult)), ]
```

Copy from the previous lab all cleanups you did to this dataset.

```
#Cast income to binary where 1 is the `>50K` level.
adult$income = ifelse(adult$income == ">50K", 1 , 0)
table(adult$income)
```

```
## 
##     0     1 
## 22653  7508
```

```
#Merge martial status/Education
table(adult$marital_status)
```

```
## 
##              Divorced     Married-AF-spouse    Married-civ-spouse 
##                  4214                    21                 14065
## Married-spouse-absent         Never-married             Separated
##                   370                  9725                   939
##               Widowed
##                   827
```

```
adult$marital_status = as.character(adult$marital_status)
adult$marital_status = ifelse(adult$marital_status == "Married-AF-spouse" | adult$marital_status ==  "Ma
adult$marital_status = as.factor(adult$marital_status)
table(adult$marital_status)
```

```
##
##              Divorced               married Married-spouse-absent
##                  4214                 14086                   370
##         Never-married             Separated               Widowed
##                  9725                   939                   827
```

```
table(adult$education)
```

```
##
##         10th         11th         12th      1st-4th      5th-6th      7th-8th
##          820         1048          377          151          288          557
##          9th   Assoc-acdm    Assoc-voc    Bachelors    Doctorate      HS-grad
##          455         1008         1307         5043          375         9840
##      Masters    Preschool  Prof-school Some-college
##         1627           45          542         6678
```

```
adult$education = as.character(adult$education)
adult$education = ifelse(adult$education == "1st-4th" | adult$education ==  "Preschool", "<=4th", adult$
adult$education = as.factor(adult$education)
table(adult$education)
```

```
##
##        <=4th         10th         11th         12th      5th-6th      7th-8th
##          196          820         1048          377          288          557
##          9th   Assoc-acdm    Assoc-voc    Bachelors    Doctorate      HS-grad
##          455         1008         1307         5043          375         9840
##      Masters  Prof-school Some-college
##         1627          542         6678
```

```
#Merge native countries
tab = sort(table(adult$native_country))
tab
```

```
##
##         Holand-Netherlands                   Scotland
##                          1                         11
##                   Honduras                    Hungary
##                         12                         13
## Outlying-US(Guam-USVI-etc)                 Yugoslavia
##                         14                         16
##                       Laos                   Thailand
##                         17                         17
##                   Cambodia            Trinadad&Tobago
##                         18                         18
##                       Hong                    Ireland
##                         19                         24
##                    Ecuador                     France
##                         27                         27
##                     Greece                       Peru
##                         29                         30
##                  Nicaragua                   Portugal
##                         33                         34
##                      Haiti                       Iran
##                         42                         42
##                     Taiwan                   Columbia
##                         42                         56
##                     Poland                      Japan
##                         56                         59
##                  Guatemala                    Vietnam
##                         63                         64
##         Dominican-Republic                      China
##                         67                         68
##                      Italy                      South
##                         68                         71
##                    Jamaica                    England
##                         80                         86
##                       Cuba                El-Salvador
##                         92                        100
##                      India                     Canada
##                        100                        107
##                Puerto-Rico                    Germany
##                        109                        128
##                Philippines                     Mexico
##                        188                        610
##              United-States
##                      27503
```

```
adult$native_country = as.character(adult$native_country)
adult$native_country = ifelse(adult$native_country %in% names(tab[tab < 100]), "Other", adult$native_cou
adult$native_country = as.factor(adult$native_country)
table(adult$native_country)
```

```
##
##        Canada   El-Salvador       Germany         India        Mexico
##           107           100           128           100           610
##         Other   Philippines   Puerto-Rico United-States
##          1316           188           109         27503
```

```
#Merge workclass and occupation
adult$worktype = paste(adult$occupation, adult$workclass, sep = ":")
adult$workclass <- NULL
adult$occupation <- NULL
tabulate = sort(table(adult$worktype))
tabulate
```

```
##
##           Craft-repair:Without-pay      Handlers-cleaners:Without-pay
##                                  1                                  1
##      Machine-op-inspct:Without-pay          Other-service:Without-pay
##                                  1                                  1
##       Transport-moving:Without-pay     Handlers-cleaners:Self-emp-inc
##                                  1                                  2
##           Adm-clerical:Without-pay          Tech-support:Self-emp-inc
##                                  3                                  3
##       Protective-serv:Self-emp-inc        Farming-fishing:Without-pay
##                                  5                                  6
##   Protective-serv:Self-emp-not-inc                    Sales:Local-gov
##                                  6                                  7
##        Farming-fishing:Federal-gov           Armed-Forces:Federal-gov
##                                  8                                  9
##        Handlers-cleaners:State-gov     Machine-op-inspct:Self-emp-inc
##                                  9                                 10
##        Machine-op-inspct:Local-gov                    Sales:State-gov
##                                 11                                 11
##        Machine-op-inspct:State-gov      Machine-op-inspct:Federal-gov
##                                 13                                 14
##                  Sales:Federal-gov          Farming-fishing:State-gov
##                                 14                                 15
## Handlers-cleaners:Self-emp-not-inc      Handlers-cleaners:Federal-gov
##                                 15                                 22
##       Transport-moving:Federal-gov      Tech-support:Self-emp-not-inc
##                                 24                                 26
##      Transport-moving:Self-emp-inc         Other-service:Self-emp-inc
##                                 26                                 27
##        Protective-serv:Federal-gov          Adm-clerical:Self-emp-inc
##                                 27                                 28
##          Farming-fishing:Local-gov          Other-service:Federal-gov
##                                 29                                 34
## Machine-op-inspct:Self-emp-not-inc             Tech-support:Local-gov
##                                 35                                 38
##         Transport-moving:State-gov        Handlers-cleaners:Local-gov
##                                 41                                 46
##      Adm-clerical:Self-emp-not-inc       Farming-fishing:Self-emp-inc
##                                 49                                 51
##             Craft-repair:State-gov             Tech-support:State-gov
##                                 55                                 56
##           Craft-repair:Federal-gov           Tech-support:Federal-gov
##                                 63                                 66
##          Craft-repair:Self-emp-inc         Transport-moving:Local-gov
##                                 99                                115
##          Protective-serv:State-gov  Transport-moving:Self-emp-not-inc
##                                116                                118
##            Other-service:State-gov             Craft-repair:Local-gov
##                                123                                143
##            Priv-house-serv:Private        Prof-specialty:Self-emp-inc
##                                143                                157
##         Prof-specialty:Federal-gov     Other-service:Self-emp-not-inc
##                                167                                173
##        Exec-managerial:Federal-gov          Exec-managerial:State-gov
##                                179                                186
##            Protective-serv:Private            Other-service:Local-gov
##                                186                                189
##          Exec-managerial:Local-gov             Adm-clerical:State-gov
##                                212                                250
##             Adm-clerical:Local-gov                 Sales:Self-emp-inc
##                                281                                281
##          Protective-serv:Local-gov           Adm-clerical:Federal-gov
##                                304                                316
##    Prof-specialty:Self-emp-not-inc             Sales:Self-emp-not-inc
##                                365                                376
##   Exec-managerial:Self-emp-not-inc       Exec-managerial:Self-emp-inc
##                                383                                385
##           Prof-specialty:State-gov   Farming-fishing:Self-emp-not-inc
##                                403                                430
##            Farming-fishing:Private      Craft-repair:Self-emp-not-inc
##                                450                                523
##           Prof-specialty:Local-gov               Tech-support:Private
##                                692                                723
##           Transport-moving:Private          Handlers-cleaners:Private
##                               1247                               1255
##          Machine-op-inspct:Private             Prof-specialty:Private
##                               1882                               2254
##            Exec-managerial:Private              Other-service:Private
##                               2647                               2665
##               Adm-clerical:Private                      Sales:Private
##                               2793                               2895
##               Craft-repair:Private
##                               3146
```

```
#collapse levels of worktype
adult$worktype = as.character(adult$worktype)
adult$worktype = ifelse(adult$worktype %in% names(tabulate[tabulate < 100]), "Other", adult$worktype)
adult$worktype = as.factor(adult$worktype)
sort(table(adult$worktype))
```

```
##
##        Transport-moving:Local-gov         Protective-serv:State-gov
##                               115                               116
## Transport-moving:Self-emp-not-inc           Other-service:State-gov
##                               118                               123
##            Craft-repair:Local-gov           Priv-house-serv:Private
##                               143                               143
##       Prof-specialty:Self-emp-inc        Prof-specialty:Federal-gov
##                               157                               167
##    Other-service:Self-emp-not-inc       Exec-managerial:Federal-gov
##                               173                               179
##         Exec-managerial:State-gov           Protective-serv:Private
##                               186                               186
##           Other-service:Local-gov         Exec-managerial:Local-gov
##                               189                               212
##            Adm-clerical:State-gov            Adm-clerical:Local-gov
##                               250                               281
##                Sales:Self-emp-inc         Protective-serv:Local-gov
##                               281                               304
##          Adm-clerical:Federal-gov   Prof-specialty:Self-emp-not-inc
##                               316                               365
##            Sales:Self-emp-not-inc  Exec-managerial:Self-emp-not-inc
##                               376                               383
##      Exec-managerial:Self-emp-inc          Prof-specialty:State-gov
##                               385                               403
##  Farming-fishing:Self-emp-not-inc           Farming-fishing:Private
##                               430                               450
##     Craft-repair:Self-emp-not-inc          Prof-specialty:Local-gov
##                               523                               692
##              Tech-support:Private                             Other
##                               723                              1008
##          Transport-moving:Private         Handlers-cleaners:Private
##                              1247                              1255
##         Machine-op-inspct:Private            Prof-specialty:Private
##                              1882                              2254
##           Exec-managerial:Private             Other-service:Private
##                              2647                              2665
##              Adm-clerical:Private                     Sales:Private
##                              2793                              2895
##              Craft-repair:Private
##                              3146
```

```
#merge relationship
adult$relationship_status = paste(adult$relationship, adult$marital_status, sep = ":")
adult$relationship <- NULL
adult$marital_status <- NULL
tabulate = sort(table(adult$relationship_status))
tabulate
```

```
##
##                    Own-child:Widowed                 Not-in-family:married
##                                   12                                    14
## Other-relative:Married-spouse-absent                Other-relative:Widowed
##                                   26                                    40
##      Own-child:Married-spouse-absent              Other-relative:Separated
##                                   43                                    53
##                    Own-child:married                   Own-child:Separated
##                                   84                                    90
##              Other-relative:Divorced                Other-relative:married
##                                  103                                   119
##      Unmarried:Married-spouse-absent  Not-in-family:Married-spouse-absent
##                                  120                                   181
##                   Own-child:Divorced                     Unmarried:Widowed
##                                  308                                   343
##              Not-in-family:Separated                   Unmarried:Separated
##                                  383                                   413
##                Not-in-family:Widowed          Other-relative:Never-married
##                                  432                                   548
##              Unmarried:Never-married                          Wife:married
##                                  801                                  1406
##                   Unmarried:Divorced                Not-in-family:Divorced
##                                 1535                                  2268
##              Own-child:Never-married          Not-in-family:Never-married
##                                 3929                                 4447
##                      Husband:married
##                                12463
```

```
adult$relationship_status = as.character(adult$relationship_status)
adult$relationship_status = ifelse(adult$relationship_status %in% names(tabulate[tabulate < 100]), "Oth
adult$relationship_status = as.factor(adult$relationship_status)
```

We will be doing model selection. We will split the dataset into 3 distinct subsets. Set the size of our splits here. For simplicitiy, all three splits will be identically sized. We are making it small so the stepwise algorithm can compute quickly. If you have a faster machine, feel free to increase this.

```
Nsplitsize = 1000
```

Now create the following variables: `Xtrain`, `ytrain`, `Xselect`, `yselect`, `Xtest`, `ytest` with `Nsplitsize` observations. Binarize the y values.

```
Xtrain = adult[1 : Nsplitsize, ]
Xtrain$income = NULL
ytrain = ifelse(adult[1 : Nsplitsize, "income"] == ">50K", 1, 0)
Xselect = adult[(Nsplitsize + 1) : (2 * Nsplitsize), ]
Xselect$income = NULL
yselect = ifelse(adult[(Nsplitsize + 1) : (2 * Nsplitsize), "income"] ==">50K", 1, 0)
Xtest = adult[(2 * Nsplitsize + 1) : (3 * Nsplitsize), ]
Xtest$income = NULL
ytest = ifelse(adult[(2 * Nsplitsize + 1) : (3 * Nsplitsize), "income"] == ">50K", 1, 0)
```

Fit a vanilla logistic regression on the training set.

```
logistic_mod = glm(ytrain ~ ., Xtrain, family = "binomial", maxit = 1000)
```

and report the log scoring rule, the Brier scoring rule.

```
brier_score = function(prob_est_mod, X, y){
  phat=predict(prob_est_mod, X, type="response")
  mean(-(y-phat)^2)
}

brier_score(logistic_mod, Xtrain, ytrain)
```

```
## [1] -2.085639e-26
```

```
brier_score(logistic_mod, Xtest, ytest)
```

```
## [1] -2.085639e-26
```

```
brier_score(logistic_mod, Xselect, yselect)
```

```
## [1] -2.085639e-26
```

We will be doing model selection using a basis of linear features consisting of all first-order interactions of the 14 raw features (this will include square terms as squares are interactions with oneself).

Create a model matrix from the training data containing all these features. Make sure it has an intercept column too (the one vector is usually an important feature). Cast it as a data frame so we can use it more easily for modeling later on. We're going to need those model matrices (as data frames) for both the select and test sets. So make them here too (copy-paste). Make sure their dimensions are sensible.

```
#TO-DO
Xmm_train = data.frame(model.matrix(~ . * . +0,Xtrain))
Xmm_select = data.frame(model.matrix(~ . * . +0, Xselect))
Xmm_test = data.frame(model.matrix(~ . * . +0, Xtest))
dim(Xmm_train)
```

```
## [1] 1000 2953
```

```
dim(Xmm_select)
```

```
## [1] 1000 2953
```

```
dim(Xmm_test)
```

```
## [1] 1000 2953
```

Write code that will fit a model stepwise. You can refer to the chunk in the practice lecture. Use the negative Brier score to do the selection. The negative of the Brier score is always positive and lower means better making this metric kind of like s_e so the picture will be the same as the canonical U-shape for oos performance.

Run the code and hit "stop" when you begin to the see the Brier score degrade appreciably oos. Be patient as it will wobble.

```
pacman::p_load(Matrix)
p_plus_one = ncol(Xmm_train)
predictor_by_iteration = c() #keep a growing list of predictors by iteration
in_sample_brier_by_iteration = c() #keep a growing list of briers by iteration
oos_brier_by_iteration = c() #keep a growing list of briers by iteration
i = 1
repeat {
  #TO-DO
  #wrap glm and predict calls with use suppressWarnings() so the console is clean during run

  if (i > Nsplitsize || i > p_plus_one){
    break
  }
}
```

Plot the in-sample and oos (select set) Brier score by p. Does this look like what's expected?

```
#TO-DO
```