# Midterm - Practice

*Chiaoya Chang*

*2/7/2020*

```r
setwd("~/Desktop/2020Winter/IMT574/midterm")
```

## Problem 1 [45 points]

```r
dermatology <- read.csv("dermatology.csv", header = TRUE, sep = "\t")
str(dermatology)
```

```
## 'data.frame':    366 obs. of  35 variables:
##  $ Erythema         : int  2 3 2 2 2 2 2 2 2 2 ...
##  $ Scathing         : int  2 3 1 2 3 3 1 2 2 2 ...
##  $ Definite.Borders: int  0 3 2 2 2 2 0 3 1 1 ...
##  $ Itching          : int  3 2 3 0 2 0 2 3 0 0 ...
##  $ Koebner          : int  0 1 1 0 2 0 0 3 2 1 ...
##  $ Polygonal        : int  0 0 3 0 2 0 0 3 0 0 ...
##  $ Follicular       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Oral             : int  0 0 3 0 2 0 0 2 0 0 ...
##  $ Knee             : int  1 1 0 3 0 0 0 0 0 0 ...
##  $ Scalp            : int  0 1 0 2 0 0 0 0 0 0 ...
##  $ Family.Hostory   : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Melanin          : int  0 0 1 0 1 0 0 2 0 0 ...
##  $ Eosinophils      : int  0 0 0 0 0 2 0 0 0 0 ...
##  $ PNL              : int  0 1 0 3 0 1 0 0 0 0 ...
##  $ Fibrosis         : int  0 0 0 0 0 0 3 0 0 0 ...
##  $ Exocytosis       : int  3 1 1 0 1 2 1 2 2 3 ...
##  $ Acanothosis      : int  2 2 2 2 2 2 3 3 1 2 ...
##  $ Hyperkeratosis   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Parakeratosis    : int  0 2 2 3 0 2 0 0 1 2 ...
##  $ Clubbing         : int  0 2 0 2 0 0 0 0 0 0 ...
##  $ Elongation       : int  0 2 0 2 0 0 2 0 0 0 ...
##  $ Thinning         : int  0 2 0 2 0 0 0 0 0 0 ...
##  $ Spongiform       : int  0 2 0 2 0 1 0 0 0 0 ...
##  $ Munro            : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Focal            : int  0 0 2 0 2 0 0 0 0 0 ...
##  $ Disapperance     : int  0 0 0 3 2 0 0 2 0 0 ...
##  $ Vacuolisation    : int  0 0 2 0 3 0 0 2 0 0 ...
##  $ Spongiosis       : int  3 0 3 0 2 2 0 3 2 2 ...
##  $ Retes            : int  0 0 2 0 3 0 0 2 0 0 ...
##  $ Follicular.1     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Perifollicular   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Inflamatory      : int  1 1 2 3 2 1 2 3 2 2 ...
##  $ Band.like        : int  0 0 3 0 3 0 0 3 0 0 ...
##  $ Age              : Factor w/ 61 levels "?","0","10","12",..: 45 60 17 31 36 32 9 47 13 21 ...
##  $ Disease          : int  2 1 3 1 3 2 5 3 4 4 ...
```

```r
dermatology$Age <- as.integer(dermatology$Age)
```

1. Let's try determining the type of disease based on the patient's Age. Use gradient descent (GD) to

build your regression model (model1). Start by writing the GD algorithm and then implement it using a programming language of your choice. [10 points]

```r
# Build a linear model
model1 = lm(Disease~Age, data=dermatology)
summary(model1)
```

```
##
## Call:
## lm(formula = Disease ~ Age, data = dermatology)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9698 -1.6992  0.1264  1.1806  3.3910
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.975805   0.177946  16.723   <2e-16 ***
## Age         -0.006014   0.005478  -1.098    0.273
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.597 on 364 degrees of freedom
## Multiple R-squared:  0.003301,   Adjusted R-squared:  0.0005624
## F-statistic: 1.205 on 1 and 364 DF,  p-value: 0.273
```

```r
# Define "X", and "Y" for the gradient descent algorithm
x <- as.matrix(dermatology[,34])
y <- as.matrix(dermatology[,35])

# Define the gradient descent function
gradD <- function(x, y, alpha, epsilon){
  iter <- 0
  i <- 0
  x <- cbind(rep(1,nrow(x)),x)
  theta <- matrix(c(1,1),ncol(x),1)
  cost <- t(x %*% theta - y) %*% (x %*% theta - y)
  # Can also multiply with constant (1/(2*nrow(x)))
  delta <- 1
  while(delta > epsilon){
    i <- i + 1
    theta <- theta - alpha*(t(x) %*% (x %*% theta - y))
    cval <- t(x %*% theta - y) %*% (x %*% theta - y)
    cost <- append(cost, cval)
    delta <- abs(cost[i+1] - cost[i])
    if((cost[i+1] - cost[i]) > 0){
      print("The cost is increasing. Try reducing alpha.")
      return()
    }
    iter <- append(iter, i)
  }
  print(sprintf("Completed in %i iterations.", i))
  return(theta)
}
```

```r
# Using the gradient descent function in a scaled data
# stheta <- gradD(scale(x), y, alpha = 0.00000009, epsilon = 10^-10)
stheta <- gradD(scale(x), y, alpha = 0.0000005, epsilon = 10^-10)
```

```
## [1] "Completed in 61509 iterations."
```

```r
stheta
```

```
##             [,1]
## [1,]  2.80325540
## [2,] -0.09178036
```

2. Use random forest on the clinical as well as histopathological attributes to classify the disease type (model2). [5 points]

```r
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```r
model2 = randomForest(Disease ~., data =dermatology)
model2
```

```
##
## Call:
##  randomForest(formula = Disease ~ ., data = dermatology)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 11
##
##           Mean of squared residuals: 0.1574269
##                     % Var explained: 93.82
```

3. Use kNN on the clinical attributes and histopathological attributes to classify the disease type and report your accuracy (model3). [5 points]

```r
sample = sample(2, nrow(dermatology), replace=TRUE, prob=c(0.7,0.3))
dermatology.training = dermatology[sample==1, 1:34]
dermatology.testing = dermatology[sample==2, 1:34]

dermatology.trainingLabels = dermatology[sample==1,35]
dermatology.testingLabels = dermatology[sample==2,35]

library(class)
dermatology_pred = knn(train=dermatology.training, test=dermatology.testing, cl=dermatology.trainingLabe
summary(dermatology_pred)
```

```
##  1  2  3  4  5  6
## 33 14 26 20 17  6
```

```r
#accuracy
library(gmodels)
CrossTable(x=dermatology_pred, y=dermatology.testingLabels, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
```

```
## |                         N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  116
##
##
##                  | dermatology.testingLabels
## dermatology_pred |           1 |           2 |           3 |           4 |           5 |           6 |  Row Total
## -----------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------
##                1 |          33 |           0 |           0 |           0 |           0 |           0 |         33
##                  |       1.000 |       0.000 |       0.000 |       0.000 |       0.000 |       0.000 |      0.284
##                  |       0.971 |       0.000 |       0.000 |       0.000 |       0.000 |       0.000 |
##                  |       0.284 |       0.000 |       0.000 |       0.000 |       0.000 |       0.000 |
## -----------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------
##                2 |           1 |           8 |           0 |           4 |           1 |           0 |         14
##                  |       0.071 |       0.571 |       0.000 |       0.286 |       0.071 |       0.000 |      0.121
##                  |       0.029 |       0.471 |       0.000 |       0.250 |       0.056 |       0.000 |
##                  |       0.009 |       0.069 |       0.000 |       0.034 |       0.009 |       0.000 |
## -----------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------
##                3 |           0 |           0 |          26 |           0 |           0 |           0 |         26
##                  |       0.000 |       0.000 |       1.000 |       0.000 |       0.000 |       0.000 |      0.224
##                  |       0.000 |       0.000 |       1.000 |       0.000 |       0.000 |       0.000 |
##                  |       0.000 |       0.000 |       0.224 |       0.000 |       0.000 |       0.000 |
## -----------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------
##                4 |           0 |           8 |           0 |          11 |           1 |           0 |         20
##                  |       0.000 |       0.400 |       0.000 |       0.550 |       0.050 |       0.000 |      0.172
##                  |       0.000 |       0.471 |       0.000 |       0.688 |       0.056 |       0.000 |
##                  |       0.000 |       0.069 |       0.000 |       0.095 |       0.009 |       0.000 |
## -----------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------
##                5 |           0 |           0 |           0 |           1 |          16 |           0 |         17
##                  |       0.000 |       0.000 |       0.000 |       0.059 |       0.941 |       0.000 |      0.147
##                  |       0.000 |       0.000 |       0.000 |       0.062 |       0.889 |       0.000 |
##                  |       0.000 |       0.000 |       0.000 |       0.009 |       0.138 |       0.000 |
## -----------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------
##                6 |           0 |           1 |           0 |           0 |           0 |           5 |          6
##                  |       0.000 |       0.167 |       0.000 |       0.000 |       0.000 |       0.833 |      0.052
##                  |       0.000 |       0.059 |       0.000 |       0.000 |       0.000 |       1.000 |
##                  |       0.000 |       0.009 |       0.000 |       0.000 |       0.000 |       0.043 |
## -----------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------
##     Column Total |          34 |          17 |          26 |          16 |          18 |           5 |        116
##                  |       0.293 |       0.147 |       0.224 |       0.138 |       0.155 |       0.043 |
## -----------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------
##
##
```

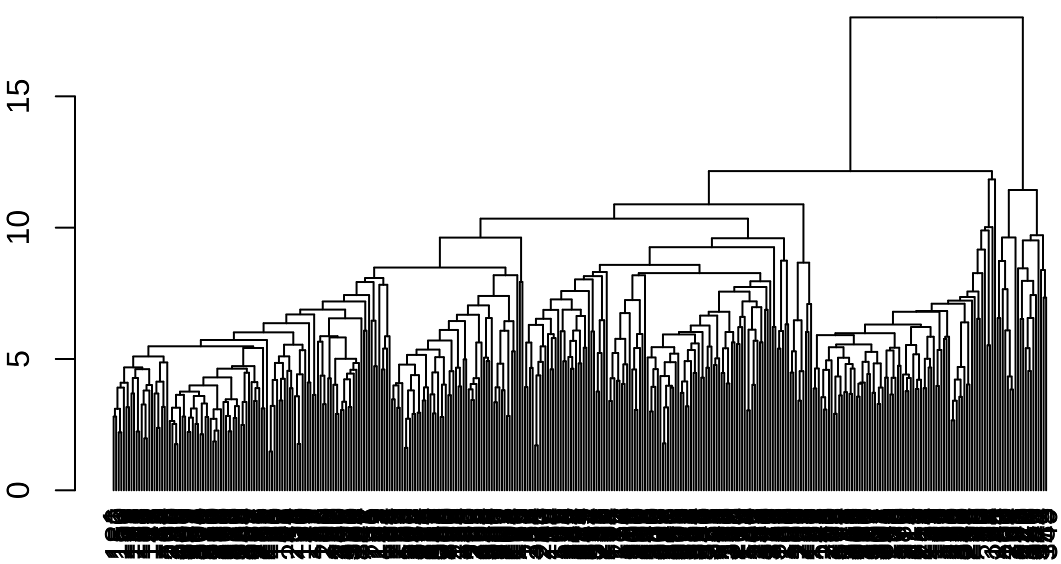
4. Finally, use two different clustering algorithms and see how well these attributes can determine the disease type (model4 and model5). [10 points]

```r
library(cluster)
#agglomerative
model4 = agnes(x=dermatology[,1:34], diss=FALSE, stand=TRUE, method="average")
```

```r
model4
```

```
## Call:     agnes(x = dermatology[, 1:34], diss = FALSE, stand = TRUE, method = "average")
## Agglomerative coefficient:  0.7579789
## Order of objects:
##   [1]   1 186 150 157 151  71 159  12 187 178 188  83  97 117 141 126  13  76
##  [19]  84  17  65 165   9 280 322 346 169 307 309 149 161 257 304 308 220 222
##  [37] 271 333  10 260 282 278 262 160 219 221 326 270 363 286  44 362 329 332
##  [55] 162 272 287 285 259 261 121  16  68  90  52  42 107  74  49  59 281  35
##  [73]  96 130  75  47 119 137  77  82  32 198 199 200  57  92  60 147 230 101
##  [91] 361 196 323 360 258 104 347  41  22  38  36  91 189 279 142 154 182 184
## [109] 183   7 223  63 129 155  98 102 113 342 354 227 228 224 225 229  25 343
## [127] 355 335 296 299 300  80 226  85 134 135 339  45  55  20 263 265 266 264
## [145]  28 148 334 338 298  93 201 297 203 205 202 204 116 122  69  23  29   2
## [163] 166 234 324  11  64 273  40 206  14  34  62 106 152  18 194 195 242 244
## [181] 245 293 274 275 249 311  94 352 357 236 238 237 239  54   4 176 172 177
## [199] 211 174 190 181 173 207 209 210 212 208 143  26 292 310 318 306  67 247
## [217] 248 341 359 366 351 305 276 312 283 337 284 319 321 295 246 320  31 336
## [235] 353  43  56  53 111 167 330 358 277 294  70 153  99 124  81 140 103 132
## [253] 235 243 325 110 125 331  33 356  89 136 131 108 120   6 138 105 231 233
## [271] 232  78  86  87   3 145 171  46 156 288 290  88  95 314 112 128 115 328
## [289] 315 365 316 109 291 289 170 191 317 197 302 301 313 303 146 168 158 175
## [307] 163 179 180   5 213 214 216   8 215 217 218 114 251 133 139 118  15  66
## [325] 192  51 193  58 250 252 253 327 340 254 255 256  39  19  50  73  30  37
## [343]  24 364  79 144  21 269  61 100 164 267 268  48  27 123 240 127 348 350
## [361] 345 241 344  72 185 349
## Height (summary):
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.478   3.805   4.771   5.120   6.083  18.004 
## 
## Available components:
## [1] "order"  "height" "ac"     "merge"  "diss"   "call"   "method" "data"
```

```r
dendcluster = as.dendrogram(model4)
plot(dendcluster)
```



```r
#divisive
model5 = kmeans(dermatology[,1:34], 6, nstart=12)
model5
```

```
## K-means clustering with 6 clusters of sizes 63, 56, 76, 67, 30, 74
##
## Cluster means:
##   Erythema Scathing Definite.Borders  Itching   Koebner Polygonal Follicular
## 1 1.968254 1.777778         1.571429 1.460317 0.6507937 0.6349206 0.09523810
## 2 2.053571 1.660714         1.107143 1.357143 0.3928571 0.1250000 0.42857143
## 3 2.118421 1.828947         1.644737 1.236842 0.6710526 0.3552632 0.03947368
## 4 2.149254 1.955224         1.626866 1.477612 0.8955224 0.6567164 0.04477612
## 5 1.933333 1.800000         1.533333 1.100000 0.4000000 0.1000000 0.70000000
## 6 2.094595 1.729730         1.702703 1.432432 0.6216216 0.5810811 0.05405405
##         Oral      Knee     Scalp Family.Hostory    Melanin Eosinophils
## 1 0.44444444 0.6190476 0.3968254     0.04761905 0.49206349  0.14285714
## 2 0.10714286 0.6607143 0.3571429     0.21428571 0.10714286  0.05357143
## 3 0.32894737 0.5921053 0.6184211     0.06578947 0.34210526  0.17105263
## 4 0.56716418 0.3880597 0.4925373     0.10447761 0.56716418  0.22388060
## 5 0.06666667 1.1000000 0.7000000     0.33333333 0.06666667  0.03333333
## 6 0.52702703 0.6081081 0.5945946     0.12162162 0.60810811  0.13513514
##         PNL  Fibrosis Exocytosis Acanothosis Hyperkeratosis Parakeratosis
## 1 0.4603175 0.2698413   1.587302    1.984127      0.4603175      1.174603
## 2 0.6071429 0.5535714   1.214286    1.892857      0.6428571      1.250000
## 3 0.6184211 0.2631579   1.355263    1.960526      0.4868421      1.328947
## 4 0.4029851 0.2686567   1.417910    2.014925      0.4776119      1.313433
## 5 0.5333333 0.4666667   1.100000    1.733333      0.7666667      1.333333
## 6 0.6351351 0.3108108   1.378378    2.013514      0.4864865      1.337838
##    Clubbing Elongation  Thinning Spongiform     Munro      Focal Disapperance
## 1 0.5873016  0.8095238 0.5238095  0.2063492 0.2698413 0.50793651    0.3968254
## 2 0.5000000  1.1071429 0.4642857  0.3035714 0.2500000 0.10714286    0.2321429
## 3 0.7500000  1.0000000 0.7631579  0.2500000 0.4473684 0.30263158    0.5921053
## 4 0.5671642  0.7910448 0.5671642  0.2985075 0.4179104 0.61194030    0.4029851
## 5 0.8000000  1.1666667 0.6666667  0.4333333 0.4666667 0.06666667    0.4666667
## 6 0.7972973  1.1621622 0.7702703  0.3513514 0.3513514 0.54054054    0.6216216
##   Vacuolisation Spongiosis     Retes Follicular.1 Perifollicular Inflamatory
## 1     0.5873016  1.0634921 0.6031746   0.03174603     0.03174603    2.031746
## 2     0.1250000  0.9642857 0.1250000   0.25000000     0.32142857    1.553571
## 3     0.3421053  1.0657895 0.3815789   0.00000000     0.00000000    1.815789
## 4     0.6716418  0.9850746 0.6567164   0.02985075     0.01492537    2.044776
## 5     0.1000000  0.7333333 0.1000000   0.63333333     0.70000000    1.800000
## 6     0.6621622  0.7972973 0.6081081   0.01351351     0.00000000    1.878378
##   Band.like       Age
## 1 0.7301587 15.888889
## 2 0.2142857  6.589286
## 3 0.4210526 24.644737
## 4 0.7611940 34.059701
## 5 0.1000000 56.366667
## 6 0.7972973 44.378378
##
## Clustering vector:
##   [1] 6 5 1 4 4 4 2 6 1 3 2 1 1 2 5 4 3 3 1 2 5 6 4 4 1 3 2 2 4 6 6 2 3 2 2 2 2
##  [38] 2 1 4 6 5 2 3 3 4 4 2 6 6 3 3 4 4 6 2 1 1 1 3 5 4 3 4 3 6 3 1 4 1 3 2 6 4
##  [75] 2 1 3 5 6 1 4 3 1 2 6 3 4 4 1 3 6 4 3 4 1 2 3 6 1 2 2 5 4 2 3 4 6 3 4 3 4
```

```
## [112] 2 4 4 4 2 1 6 1 2 3 4 5 3 3 1 5 4 2 2 6 4 3 1 3 1 5 3 4 3 2 2 6 1 3 4 6 1
## [149] 3 4 3 1 4 6 1 4 6 3 3 4 3 6 1 2 2 6 5 6 2 1 4 6 6 2 4 3 3 4 4 3 1 6 6 2 5
## [186] 3 2 6 6 1 6 4 1 2 3 1 6 6 3 4 2 1 6 3 4 6 2 6 3 1 6 5 1 3 6 1 6 4 4 3 3 4
## [223] 2 1 3 4 3 2 6 4 3 4 6 6 5 4 3 6 6 5 5 1 6 6 3 1 6 5 2 6 4 6 3 1 4 6 3 1 6
## [260] 1 3 3 2 2 2 2 2 2 5 3 5 4 3 6 5 3 2 1 2 6 6 1 6 1 5 1 4 4 1 3 1 4 1 6 3 5
## [297] 2 3 5 6 1 3 6 3 2 1 1 6 2 4 6 3 1 2 3 6 6 4 6 6 1 3 1 5 6 4 3 6 2 4 3 1 4
## [334] 3 5 6 6 4 3 6 3 5 6 5 2 3 4 1 2 5 1 5 6 4 6 3 5 4 1 4 1 1 3 1 6 3
##
## Within cluster sum of squares by cluster:
## [1] 1857.905 1769.214 2277.750 2140.090 1023.733 2597.324
##  (between_SS / total_SS =  87.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Make sure to report your actual model for each of the above. Now, compare and contrast the five models you built. Having done both classification and clustering on the same dataset, what can you say about this data and/or the techniques you used? Write your thoughts in 2-3 paragraphs. [10 points]

```r
library('dplyr')
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
dermatology %>% group_by(Disease) %>% summarise_all(funs(mean))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.

## # A tibble: 6 x 35
##   Disease Erythema Scathing Definite.Borders Itching Koebner Polygonal
##     <int>    <dbl>    <dbl>            <dbl>   <dbl>   <dbl>     <dbl>
## 1       1     2.29     2.20            2.10    0.946  0.670        0
## 2       2     2.28     2.07            0.951   1.62   0.0328       0
```

```
## 3       3     2.08     1.62            2.10    2.28   1.35         2.28
## 4       4     1.90     1.51            1.18    0.469  1.18         0
## 5       5     1.5      1.13            0.846   1.88   0            0
## 6       6     2.05     1.75            1.05    0.5    0            0
## # ... with 28 more variables: Follicular <dbl>, Oral <dbl>, Knee <dbl>,
## #   Scalp <dbl>, Family.Hostory <dbl>, Melanin <dbl>, Eosinophils <dbl>,
## #   PNL <dbl>, Fibrosis <dbl>, Exocytosis <dbl>, Acanothosis <dbl>,
## #   Hyperkeratosis <dbl>, Parakeratosis <dbl>, Clubbing <dbl>,
## #   Elongation <dbl>, Thinning <dbl>, Spongiform <dbl>, Munro <dbl>,
## #   Focal <dbl>, Disapperance <dbl>, Vacuolisation <dbl>, Spongiosis <dbl>,
## #   Retes <dbl>, Follicular.1 <dbl>, Perifollicular <dbl>, Inflamatory <dbl>,
## #   Band.like <dbl>, Age <dbl>
```

Overall presentation [5 points]

## Problem 2 [25 points]

```
hatecrime <- read.csv("hatecrime.csv", header = TRUE, sep = ",")
str(hatecrime)
```

```
## 'data.frame':    51 obs. of  12 variables:
##  $ state                                   : Factor w/ 51 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7
##  $ median_household_income                 : int  42278 67629 49254 44922 60487 60940 70161 57522 682
##  $ share_unemployed_seasonal               : num  0.06 0.064 0.063 0.052 0.059 0.04 0.052 0.049 0.067
##  $ share_population_in_metro_areas         : num  0.64 0.63 0.9 0.69 0.97 0.8 0.94 0.9 1 0.96 ...
##  $ share_population_with_high_school_degree: num  0.821 0.914 0.842 0.824 0.806 0.893 0.886 0.874 0.8
##  $ share_non_citizen                       : num  0.02 0.04 0.1 0.04 0.13 0.06 0.06 0.05 0.11 0.09 .
##  $ share_white_poverty                     : num  0.12 0.06 0.09 0.12 0.09 0.07 0.06 0.08 0.04 0.11
##  $ gini_index                              : num  0.472 0.422 0.455 0.458 0.471 0.457 0.486 0.44 0.53
##  $ share_non_white                         : num  0.35 0.42 0.49 0.26 0.61 0.31 0.3 0.37 0.63 0.46 .
##  $ share_voters_voted_trump                : num  0.63 0.53 0.5 0.6 0.33 0.44 0.41 0.42 0.04 0.49 ..
##  $ hate_crimes_per_100k_splc               : num  0.1258 0.1437 0.2253 0.0691 0.2558 ...
##  $ avg_hatecrimes_per_100k_fbi             : num  1.806 1.657 3.414 0.869 2.398 ...
```

1. How does income inequality relate to the number of hate crimes and hate incidents? [5 points]

```
#regression
library(tidyr)
new_df <- hatecrime %>% drop_na(hate_crimes_per_100k_splc)
new_df_1 <- new_df %>% drop_na(gini_index, median_household_income)

# Build a linear model
model2_1_1 = lm(hate_crimes_per_100k_splc~gini_index, data=new_df_1)
summary(model2_1_1)
```

```
##
## Call:
## lm(formula = hate_crimes_per_100k_splc ~ gini_index, data = new_df_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28669 -0.14565 -0.04991  0.07356  0.91085
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.5275     0.7833  -1.950   0.0574 .
```

```
## gini_index    4.0205     1.7177   2.341   0.0237 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2412 on 45 degrees of freedom
## Multiple R-squared:  0.1085, Adjusted R-squared:  0.08872
## F-statistic: 5.478 on 1 and 45 DF,  p-value: 0.02374
```

```
# Define "X", and "Y" for the gradient descent algorithm
x <- as.matrix(new_df_1[,8])
y <- as.matrix(new_df_1[,11])

# Using the gradient descent function in a scaled data
stheta <- gradD(scale(x), y, alpha =  0.00005 , epsilon = 10^-10)
```

```
## [1] "Completed in 4695 iterations."
```

```
stheta
```

```
##            [,1]
## [1,] 0.30410407
## [2,] 0.08327066
```

2. How can we predict the number of hate crimes and hate incidents from race/nature of the population? [5 points]

```
new_df$new <- new_df$hate_crimes_per_100k_splc + (new_df$avg_hatecrimes_per_100k_fbi)*10/365*6
new_df_2 <- new_df %>% drop_na(share_non_citizen)

# regression
model2_2_1 = lm(hate_crimes_per_100k_splc ~ share_non_white, data=new_df_2)
summary(model2_2_1)
```

```
##
## Call:
## lm(formula = hate_crimes_per_100k_splc ~ share_non_white, data = new_df_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24715 -0.14189 -0.09149  0.05348  1.16111
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.24271    0.08973   2.705  0.00975 **
## share_non_white  0.18807    0.25640   0.734  0.46723
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2529 on 43 degrees of freedom
## Multiple R-squared:  0.01236,    Adjusted R-squared:  -0.01061
## F-statistic: 0.538 on 1 and 43 DF,  p-value: 0.4672
```

```
model2_2_2 = lm(new ~ share_non_white, data=new_df_2)
summary(model2_2_2)
```

```
##
## Call:
## lm(formula = new ~ share_non_white, data = new_df_2)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57656 -0.28426 -0.10761  0.08071  2.49162
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.5518     0.1784   3.093  0.00348 **
## share_non_white   0.4436     0.5098   0.870  0.38907
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5029 on 43 degrees of freedom
## Multiple R-squared:  0.0173, Adjusted R-squared:  -0.005551
## F-statistic: 0.7571 on 1 and 43 DF,  p-value: 0.3891
```

```
# Define "X", and "Y" for the gradient descent algorithm
# Maine, and    Mississippi have null values in share_non_citizen
x <- as.matrix(new_df_2[,9])
y <- as.matrix(new_df_2[,13])

# Using the gradient descent function in a scaled data
stheta <- gradD(scale(x), y, alpha =  0.00005 , epsilon = 10^-10)
```

```
## [1] "Completed in 4840 iterations."
```

```
stheta
```

```
##           [,1]
## [1,] 0.6926513
## [2,] 0.0659863
```

3. How does the number of hate crimes vary across states? Is there any similarity in number of hate incidents (per 100,000 people) between some states than in others — both according to the SPLC after the election and the FBI before it? [10 points]

```
# cluster
# divisive

# hate_crimes_per_100k_splc + avg_hatecrimes_per_100k_fbi *10/365*6
model2_3_1 = kmeans(new_df[,13], 3, nstart=25)
model2_3_1
```

```
## K-means clustering with 3 clusters of sizes 33, 13, 1
##
## Cluster means:
##        [,1]
## 1 0.4694958
## 2 1.0438446
## 3 3.3228737
##
## Clustering vector:
##  [1] 1 1 2 1 1 2 2 1 3 1 1 1 1 1 1 1 2 1 2 1 2 2 2 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1
## [39] 1 1 1 1 1 1 2 1 1
##
## Within cluster sum of squares by cluster:
## [1] 0.7180603 0.5817005 0.0000000
```

```
##  (between_SS / total_SS =  88.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
# hate_crimes_per_100k_splc, avg_hatecrimes_per_100k_fbi
model2_3_3 = kmeans(new_df[,c(11,12)], 3, nstart=25)
model2_3_3
```

```
## K-means clustering with 3 clusters of sizes 1, 30, 16
##
## Cluster means:
##   hate_crimes_per_100k_splc avg_hatecrimes_per_100k_fbi
## 1                 1.5223017                   10.953480
## 2                 0.2077298                    1.464373
## 3                 0.4086358                    3.449140
##
## Clustering vector:
##  1  2  3  4  5  6  7  8  9 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
##  2  2  3  2  2  3  3  2  1  2  2  2  2  2  2  2  3  2  3  2  3  3  3  2  2  3
## 28 29 30 31 32 33 34 36 37 38 39 40 41 43 44 45 46 47 48 49 50
##  3  2  2  3  2  3  2  3  2  3  2  2  2  3  2  2  2  2  3  2  2
##
## Within cluster sum of squares by cluster:
## [1]  0.000000 10.606706  6.560275
##  (between_SS / total_SS =  87.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```