STAT-S675

Homework 12

John Koo

Link to assignment

```
# load stuff
import::from(magrittr, `%>%`)
dp <- loadNamespace('dplyr')
td <- loadNamespace('tidyr')
library(ggplot2)
source('http://pages.iu.edu/~mtrosset/Courses/675/knn.r')

# create data
set.seed(675)
s5.df <- phase2.createData.select(sigma = 5)
s10.df <- phase2.createData.select(sigma = 10)
s15.df <- phase2.createData.select(sigma = 15)
```

Exercise 1

$\sigma = 5$

```
s5.df %>%
  phase2.edm() %>%
  phase2.knn(k = 7)
```

```
$confusion
   pred
id   1  2  3
  1 22  3  0
  2  0 25  0
  3  0  0 25

$error
[1] 0.04
```

$\sigma = 10$

```
s10.df %>%
  phase2.edm() %>%
  phase2.knn(k = 7)
```

```
$confusion
   pred
id   1  2  3
  1 19  3  3
  2  2 19  4
  3  0  5 20

$error
[1] 0.2266667
```

$\sigma = 15$

```
s15.df %>%
  phase2.edm() %>%
  phase2.knn(k = 7)
```

```
$confusion
   pred
id   1  2  3
  1 13  7  5
  2  4 13  8
  3  4  4 17

$error
[1] 0.4266667
```

The error rate increases as σ increases. This is because the data are generated as MVN with standard deviations scaled by the `sigma` parameter that is fed to the function. The higher the standard deviation, the more mixing there is.

Exercise 2

```
cols.to.use <- paste0('Var', seq(5))
```

$\sigma = 5$

```
s5.df %>%
  dp$select(c('subject.id', 'group.id', cols.to.use)) %>%
  phase2.edm() %>%
  phase2.knn(k = 7)
```

```
$confusion
   pred
id   1  2  3
  1  3 11 11
  2  7 12  6
  3  2 11 12

$error
[1] 0.64
```

$\sigma = 10$

```r
s10.df %>%
  dp$select(c('subject.id', 'group.id', cols.to.use)) %>%
  phase2.edm() %>%
  phase2.knn(k = 7)
```

```
$confusion
   pred
id   1  2  3
  1  6  7 12
  2  5  5 15
  3 13  5  7

$error
[1] 0.76
```

$\sigma = 15$

```r
s15.df %>%
  dp$select(c('subject.id', 'group.id', cols.to.use)) %>%
  phase2.edm() %>%
  phase2.knn(k = 7)
```

```
$confusion
   pred
id   1  2  3
  1  6  6 13
  2  9  4 12
  3  7 11  7

$error
[1] 0.7733333
```

Exercise 3

```r
cols.to.use <- paste0('Var', seq(11, 15))
```

$\sigma = 5$

```r
s5.df %>%
  dp$select(c('subject.id', 'group.id', cols.to.use)) %>%
  phase2.edm() %>%
  phase2.knn(k = 7)
```

```
$confusion
   pred
id   1  2  3
  1 24  1  0
  2  0 25  0
  3  0  0 25

$error
[1] 0.01333333
```

$\sigma = 10$

```r
s10.df %>%
  dp$select(c('subject.id', 'group.id', cols.to.use)) %>%
  phase2.edm() %>%
  phase2.knn(k = 7)
```

```
$confusion
   pred
id   1  2  3
  1 19  3  3
  2  2 21  2
  3  0  6 19

$error
[1] 0.2133333
```

$\sigma = 15$

```r
s15.df %>%
  dp$select(c('subject.id', 'group.id', cols.to.use)) %>%
  phase2.edm() %>%
  phase2.knn(k = 7)
```

```
$confusion
   pred
id   1  2  3
  1 12  7  6
  2  4 13  8
  3  4  4 17

$error
[1] 0.44
```

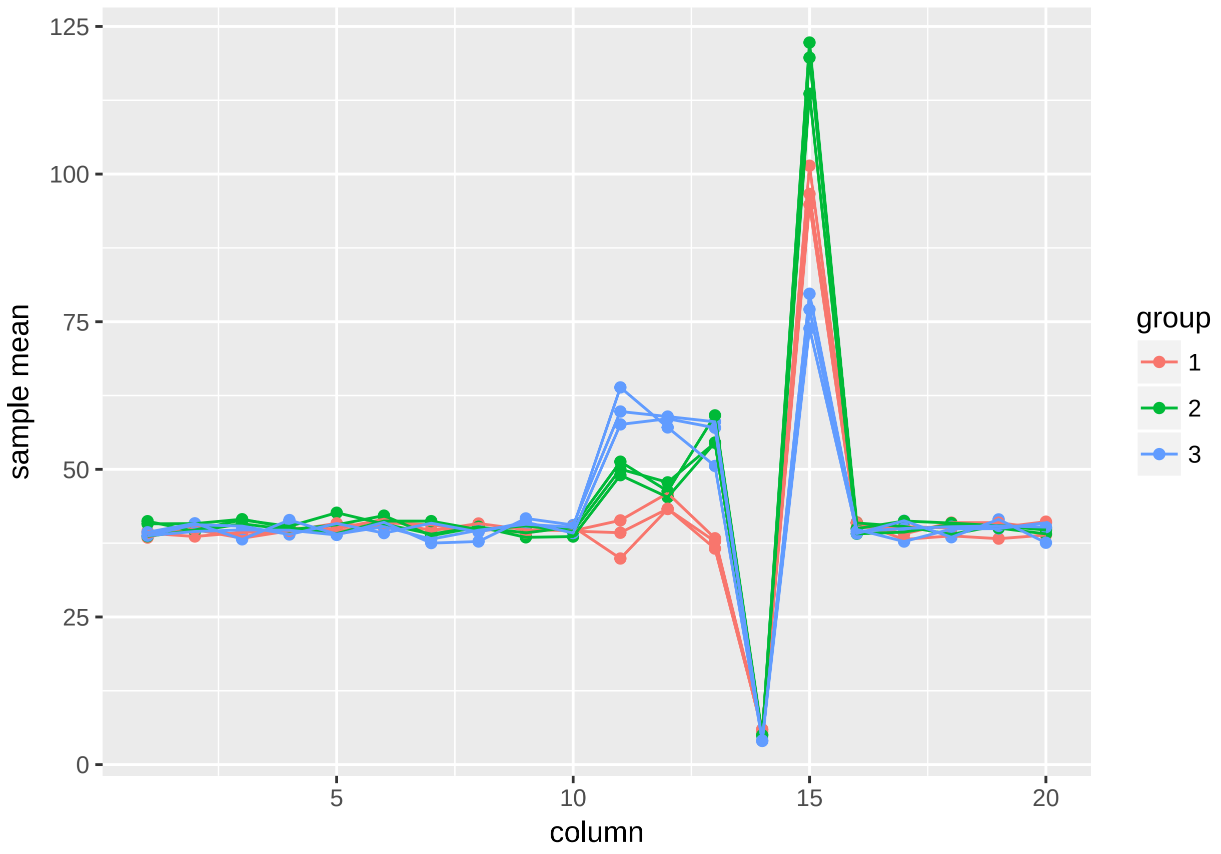
Exercise 4

Using just the first five columns resulted in poor performance across all values of σ (similar to random guessing) whereas using just columns 11 to 15 resulted in increased performance for $\sigma = 5$ and similar performance for $\sigma = 10$ and $\sigma = 15$. We can take a look at why by looking at the sample standard deviations of the columns:

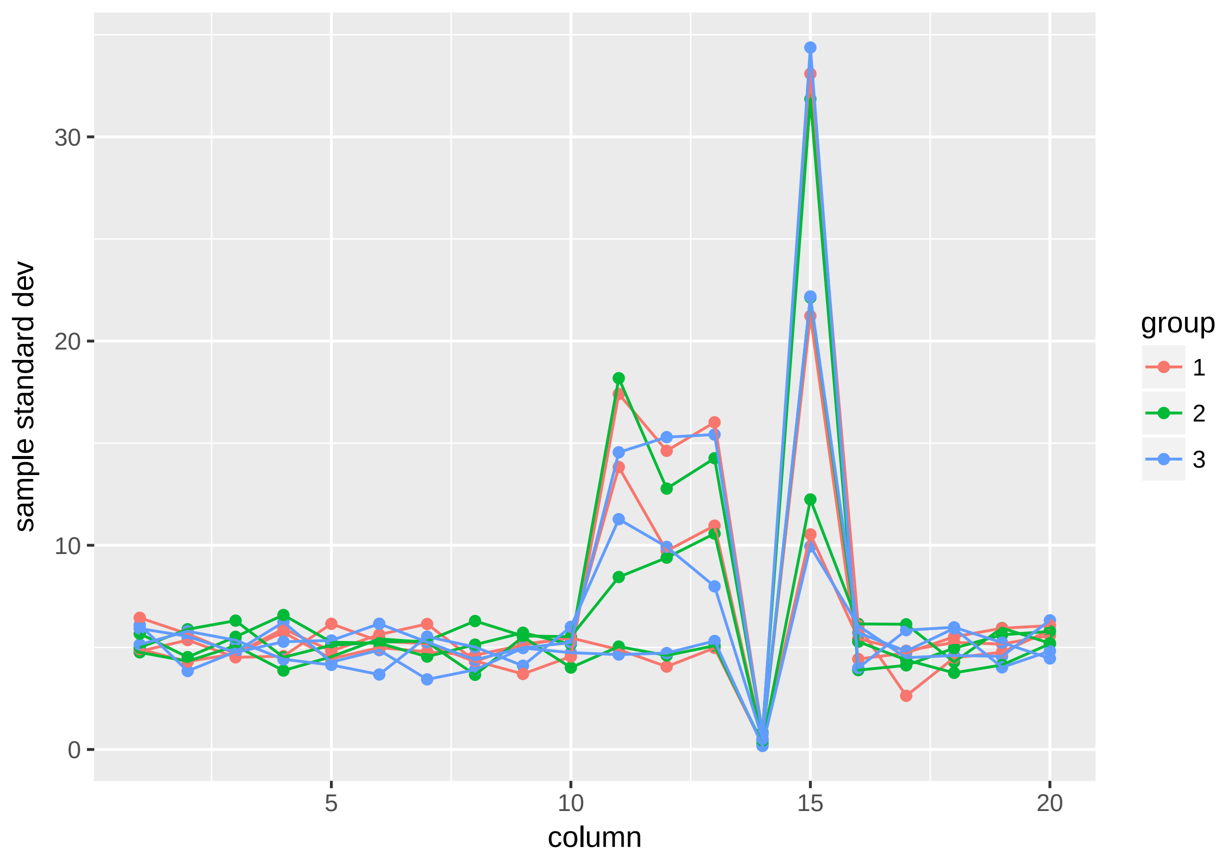
```r
.col.sample.mean <- function(input.df, input.sigma) {
  input.df %>%
    dp$select(-subject.id) %>%
    dp$group_by(group.id) %>%
    dp$summarise_all(mean) %>%
    dp$ungroup() %>%
    td$gather('column', 'm', seq(2, 21)) %>%
    dp$mutate(column = as.numeric(gsub('Var', '', column)),
              sigma = input.sigma,
              group.id = as.character(group.id))
}

.col.sample.sd <- function(input.df, input.sigma) {
  input.df %>%
    dp$select(-subject.id) %>%
    dp$group_by(group.id) %>%
    dp$summarise_all(sd) %>%
    dp$ungroup() %>%
    td$gather('column', 's', seq(2, 21)) %>%
    dp$mutate(column = as.numeric(gsub('Var', '', column)),
              sigma = input.sigma,
              group.id = as.character(group.id))
}

dp$bind_rows(
  .col.sample.mean(s5.df, '5'),
  .col.sample.mean(s10.df, '10'),
  .col.sample.mean(s15.df, '15')
) %>%
  dp$mutate(sigma = factor(sigma, levels = c('5', '10', '15'))) %>%
  ggplot() +
  geom_point(aes(x = column, y = m, colour = group.id)) +
  geom_line(aes(x = column, y = m,
                group = interaction(sigma, group.id), colour = group.id)) +
  labs(y = 'sample mean', colour = 'group')
```



```
dp$bind_rows(
  .col.sample.sd(s5.df, '5'),
  .col.sample.sd(s10.df, '10'),
  .col.sample.sd(s15.df, '15')
) %>%
  dp$mutate(sigma = factor(sigma, levels = c('5', '10', '15'))) %>%
  ggplot() +
  geom_point(aes(x = column, y = s, colour = group.id)) +
  geom_line(aes(x = column, y = s,
                group = interaction(sigma, group.id), colour = group.id)) +
  labs(y = 'sample standard dev', colour = 'group')
```



We can see why columns 1 to 5 did not help with classification—they all have the same mean regardless of group (and less importantly, the same standard deviation). On the other hand, columns 11-15 have different group-wise means (and less importantly, different standard deviations based on the `sigma` function argument).

One feature selection methid might be to, for each column, compute the group means and see how far apart they are. Then this can be compared to groupwise sample standard deviations—if the distances between the group means is large compared to the group standard deviations, then assuming that points for each group are clustered around their means, it is probably a good feature for classification.