Assignment Submission for Session 11-15 – Task 1 to 5 dated 18 Sept 2019

Task 1:

1. Use the given link below and locate the bank marketing dataset. Data Set Link

Perform the below operations:

a. Is there any association between Job and default?

b. Is there any significant difference in duration of last call between people having housing loan or not?

c. Is there any association between consumer price index and consumer?

d. Is the employment variation rate consistent across job types?

e. Is the employment variation rate same across education?

f. Which group is more confident?

Solution:

a. Is there any association between Job and default?

The R-script for the given problem is as follows:

 # Import Bank Marketing Data

```
> # Import BankMArketing Data
> View(bank_additional)
> dim(bank_additional)
[1] 4119   21
> str(bank_additional)
Classes ‘spec_tbl_df’, ‘tbl_df’, ‘tbl’ and 'data.frame':     4119 obs. of  21
variables:
 $ age           : num  30 39 25 38 47 32 32 41 31 35 ...
 $ job           : chr  "blue-collar" "services" "services" "services" ...
 $ marital       : chr  "married" "single" "married" "married" ...
 $ education     : chr  "basic.9y" "high.school" "high.school" "basic.9y" ...
 $ default       : chr  "no" "no" "no" "no" ...
 $ housing       : chr  "yes" "no" "yes" "unknown" ...
 $ loan          : chr  "no" "no" "no" "unknown" ...
 $ contact       : chr  "cellular" "telephone" "telephone" "telephone" ...
 $ month         : chr  "may" "may" "jun" "jun" ...
 $ day_of_week   : chr  "fri" "fri" "wed" "fri" ...
 $ duration      : num  487 346 227 17 58 128 290 44 68 170 ...
 $ campaign      : num  2 4 1 3 1 3 4 2 1 1 ...
 $ pdays         : num  999 999 999 999 999 999 999 999 999 999 ...
 $ previous      : num  0 0 0 0 0 2 0 0 1 0 ...
```

```
 $ poutcome      : chr  "nonexistent" "nonexistent" "nonexistent" "nonexisten
t" ...
 $ emp.var.rate  : num  -1.8 1.1 1.4 1.4 -0.1 -1.1 -1.1 -0.1 -0.1 1.1 ...
 $ cons.price.idx: num  92.9 94 94.5 94.5 93.2 ...
 $ cons.conf.idx : num  -46.2 -36.4 -41.8 -41.8 -42 -37.5 -37.5 -42 -42 -36.4
...
 $ euribor3m     : num  1.31 4.86 4.96 4.96 4.19 ...
 $ nr.employed   : num  5099 5191 5228 5228 5196 ...
 $ y             : chr  "no" "no" "no" "no" ...
 - attr(*, "spec")=
  .. cols(
  ..   age = col_double(),
  ..   job = col_character(),
  ..   marital = col_character(),
  ..   education = col_character(),
  ..   default = col_character(),
  ..   housing = col_character(),
  ..   loan = col_character(),
  ..   contact = col_character(),
  ..   month = col_character(),
  ..   day_of_week = col_character(),
  ..   duration = col_double(),
  ..   campaign = col_double(),
  ..   pdays = col_double(),
  ..   previous = col_double(),
  ..   poutcome = col_character(),
  ..   emp.var.rate = col_double(),
  ..   cons.price.idx = col_double(),
  ..   cons.conf.idx = col_double(),
  ..   euribor3m = col_double(),
  ..   nr.employed = col_double(),
  ..   y = col_character()
  .. )
> with(bank_additional,table(job,default))
               default
job              no unknown yes
  admin.        889     123   0
  blue-collar   599     285   0
  entrepreneur  113      35   0
  housemaid      79      31   0
  management    280      44   0
  retired       126      40   0
  self-employed 134      25   0
  services      306      87   0
  student        70      12   0
  technician    606      85   0
  unemployed     92      18   1
  unknown        21      18   0
```

Conclusion/Interpretation: There is NO association between Job and default.

b. Is there any significant difference in duration of last call between people having housing loan or not?

```
> #Is there any significant difference in duration of last call between peopl
e having housing loan or not?
> with(bank_additional,table(duration,housing))
        housing
duration no unknown yes
     0    0       0   1
     4    0       0   1
     5    3       0   1
     6    2       0   3
     7    2       0   2
     8    0       0   6
     9    6       0   3
     10   4       0   6
     11   3       0   5
     12   4       0   2
     13   2       0   4
     14   3       0   3
     15   3       0   3
     16   4       0   7
     17   5       1   4
     18   1       0   3
     19   3       0   6
     20   5       0   2
     21   4       0   3
     22   5       0   4
     23   2       1   5
     24   4       0   3
     25   1       0   4
     26   3       0   6
     27   4       0   5
     28   1       0   2
     29   2       0   2
     30   1       1   2
     31   3       0   5
     32   0       0   6
     33   0       0   3
     34   3       0   3
     35   3       2   4
     36   3       0   6
     37   2       1   3
     38   2       0   6
     39   3       0   4
     40   4       0   2
     41   3       0   2
     42   5       0   5
     43   1       0   8
     44   2       0   6
     45   1       0   3
```

46	2	0	1
47	1	0	5
48	1	0	4
49	9	0	1
50	6	0	2
51	7	1	4
52	2	0	3
53	2	0	5
54	7	0	5
55	6	0	6
56	4	1	3
57	4	0	9
58	3	0	7
59	6	1	7
60	5	0	2
61	3	1	6
62	4	0	6
63	4	0	8
64	7	0	6
65	6	0	2
66	6	0	3
67	6	1	7
68	8	0	6
69	9	0	8
70	8	0	5
71	7	0	6
72	6	1	6
73	10	0	12
74	7	0	6
75	5	0	8
76	8	0	5
77	8	1	15
78	6	0	7
79	8	0	2
80	7	0	5
81	10	2	9
82	4	0	12
83	11	0	9
84	10	0	3
85	8	1	7
86	5	0	6
87	7	1	10
88	6	0	13
89	8	0	5
90	8	1	11
91	12	0	2
92	7	0	5
93	6	1	7
94	7	0	8
95	9	2	6
96	8	0	7
97	5	0	8
98	5	0	7
99	6	0	7
100	5	1	6
101	6	1	9
102	8	0	9

103	11	0	7
104	3	1	9
105	2	0	8
106	3	0	8
107	5	2	11
108	3	1	4
109	5	0	7
110	2	0	4
111	7	0	10
112	9	0	14
113	5	0	15
114	10	0	9
115	6	0	7
116	3	1	5
117	5	0	5
118	3	0	7
119	8	0	5
120	3	0	5
121	8	0	8
122	7	1	12
123	5	2	7
124	9	0	4
125	7	0	6
126	10	1	5
127	8	1	5
128	8	0	8
129	5	0	6
130	8	0	9
131	9	1	9
132	6	0	5
133	4	1	6
134	7	0	5
135	7	2	10
136	8	1	7
137	4	0	9
138	3	0	4
139	8	0	9
140	5	0	6
141	7	1	5
142	6	0	8
143	3	1	9
144	6	1	5
145	6	0	14
146	8	1	7
147	7	3	7
148	6	0	7
149	5	0	6
150	1	0	10
151	9	0	4
152	4	0	7
153	4	2	5
154	5	1	6
155	8	1	9
156	2	1	9
157	9	0	6
158	4	0	7
159	6	0	12

160	5	0	12
161	8	0	9
162	6	0	4
163	5	0	7
164	7	1	8
165	6	0	7
166	7	0	9
167	2	0	9
168	6	0	9
169	2	3	4
170	5	0	6
171	7	1	5
172	4	0	6
173	3	1	9
174	4	0	3
175	6	1	7
176	2	0	6
177	5	0	4
178	6	0	5
179	4	0	2
180	9	0	8
181	6	0	10
182	4	0	7
183	6	0	7
184	4	0	11
185	5	0	3
186	3	0	4
187	6	0	5
188	5	0	7
189	3	0	3
190	4	0	3
191	5	0	5
192	4	0	4
193	5	0	8
194	2	0	3
195	3	0	6
196	2	0	5
197	6	0	2
198	5	1	5
199	4	1	3
200	7	0	7
201	6	0	8
202	4	1	5
203	5	0	5
204	9	1	7
205	0	0	2
206	8	0	6
207	4	0	9
208	3	0	5
209	4	0	4
210	0	0	7
211	5	0	8
212	4	0	8
213	5	0	3
214	5	0	3
215	3	1	9
216	2	0	2

217	7	0	2
218	7	0	4
219	5	0	10
220	2	0	1
221	4	1	5
222	4	1	4
223	6	0	3
224	4	0	6
225	4	0	8
226	2	0	9
227	2	1	4
228	5	1	5
229	0	1	3
230	4	0	5
231	3	0	8
232	5	0	7
233	1	0	6
234	4	1	5
235	4	0	0
236	2	0	5
237	1	0	2
238	4	0	3
239	4	0	5
240	2	0	4
241	4	0	3
242	2	0	2
243	3	0	4
244	5	0	5
245	6	1	7
246	5	1	5
247	4	0	9
248	2	0	3
249	1	0	6
250	2	0	6
251	3	0	2
252	4	0	9
253	4	0	2
254	0	0	3
255	1	0	4
256	3	0	3
257	7	0	3
258	8	0	5
259	8	0	5
260	2	0	3
261	4	0	3
262	4	1	3
263	4	0	4
264	4	0	4
265	3	0	4
266	2	0	6
267	1	0	5
268	4	0	3
269	1	0	3
270	3	1	2
271	2	1	3
272	4	1	2
273	2	0	3

274	1	0	4
275	3	0	3
276	3	1	0
277	0	0	2
278	4	0	1
279	2	0	2
280	3	0	3
281	4	0	7
282	1	0	1
283	1	0	0
284	3	0	2
285	3	0	0
286	5	0	5
287	4	0	1
288	2	0	5
289	4	0	1
290	2	0	2
291	2	0	3
292	3	0	2
293	3	0	4
294	2	0	3
295	5	0	1
296	5	0	1
297	2	0	3
298	2	0	2
299	3	0	0
300	2	0	4
301	4	0	2
302	2	0	1
303	0	0	2
304	2	0	4
305	1	0	4
306	1	0	0
307	2	0	2
308	1	0	3
309	1	1	4
310	2	0	2
311	1	1	2
312	1	0	2
313	4	1	3
314	2	0	5
315	3	0	0
316	5	0	3
317	4	0	0
318	2	1	3
319	2	1	0
320	0	0	7
321	2	0	2
322	3	0	7
323	1	0	1
324	0	0	2
325	0	0	1
326	1	0	5
327	2	0	1
328	2	0	2
329	5	0	2
330	1	0	2

```
    331   2       1   2
    332   4       0   1
    333   2       1   1
    334   1       0   3
    335   2       0   1
 [ reached getOption("max.print") -- omitted 495 rows ]
```

c. Is there any association between consumer price index and consumer?

```
> #c. Is there any association between consumer price index and consumer?
> with(bank_additional,table(cons.price.idx,cons.conf.idx))
              cons.conf.idx
cons.price.idx -50.8 -50 -49.5 -47.1 -46.2 -45.9 -42.7 -42 -41.8 -40.8 -40.4 -40.3 -40 -39.8 -38.3
        92.201     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        92.379     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        92.431     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        92.469     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        92.649     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        92.713     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        92.756     0   0     0     0     0     1     0   0     0     0     0     0   0     0     0
        92.843     0  25     0     0     0     0     0   0     0     0     0     0   0     0     0
        92.893     0   0     0     0   597     0     0   0     0     0     0     0   0     0     0
        92.963     0   0     0     0     0     0     0   0     0    75     0     0   0     0     0
        93.075     0   0     0   201     0     0     0   0     0     0     0     0   0     0     0
        93.2       0   0     0     0     0     0     0 386     0     0     0     0   0     0     0
        93.369     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        93.444     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        93.749     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        93.798     0   0     0     0     0     0     0   0     0     0     6     0   0     0     0
        93.876     0   0     0     0     0     0     0   0     0     0     0     0  23     0     0
        93.918     0   0     0     0     0     0   667   0     0     0     0     0   0     0     0
        93.994     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        94.027     0   0     0     0     0     0     0   0     0     0     0     0   0     0    33
        94.055     0   0     0     0     0     0     0   0     0     0     0     0   0    24     0
        94.199     0   0     0     0     0     0     0   0     0     0     0     0   0     0     0
        94.215     0   0     0     0     0     0     0   0     0     0     0    30   0     0     0
        94.465     0   0     0     0     0     0     0   0   431     0     0     0   0     0     0
        94.601     0   0    20     0     0     0     0   0     0     0     0     0   0     0     0
        94.767    24   0     0     0     0     0     0   0     0     0     0     0   0     0     0
              cons.conf.idx
cons.price.idx -37.5 -36.4 -36.1 -34.8 -34.6 -33.6 -33 -31.4 -30.1 -29.8 -26.9
        92.201     0     0     0     0     0     0   0    75     0     0     0
        92.379     0     0     0     0     0     0   0     0     0    25     0
        92.431     0     0     0     0     0     0   0     0     0     0    43
        92.469     0     0     0     0     0    14   0     0     0     0     0
        92.649     0     0     0     0     0     0   0     0    36     0     0
        92.713     0     0     0     0     0     0  21     0     0     0     0
        92.756     0     0     0     0     0     0   0     0     0     0     0
        92.843     0     0     0     0     0     0   0     0     0     0     0
        92.893     0     0     0     0     0     0   0     0     0     0     0
        92.963     0     0     0     0     0     0   0     0     0     0     0
        93.075     0     0     0     0     0     0   0     0     0     0     0
        93.2       0     0     0     0     0     0   0     0     0     0     0
        93.369     0     0     0    23     0     0   0     0     0     0     0
        93.444     0     0   528     0     0     0   0     0     0     0     0
        93.749     0     0     0     0    14     0   0     0     0     0     0
        93.798     0     0     0     0     0     0   0     0     0     0     0
        93.876     0     0     0     0     0     0   0     0     0     0     0
        93.918     0     0     0     0     0     0   0     0     0     0     0
        93.994     0   758     0     0     0     0   0     0     0     0     0
        94.027     0     0     0     0     0     0   0     0     0     0     0
```

```
  94.055      0     0     0     0     0     0   0     0     0     0     0
  94.199     39     0     0     0     0     0   0     0     0     0     0
  94.215      0     0     0     0     0     0   0     0     0     0     0
  94.465      0     0     0     0     0     0   0     0     0     0     0
  94.601      0     0     0     0     0     0   0     0     0     0     0
  94.767      0     0     0     0     0     0   0     0     0     0     0

>
```

d. Is the employment variation rate consistent across job types?

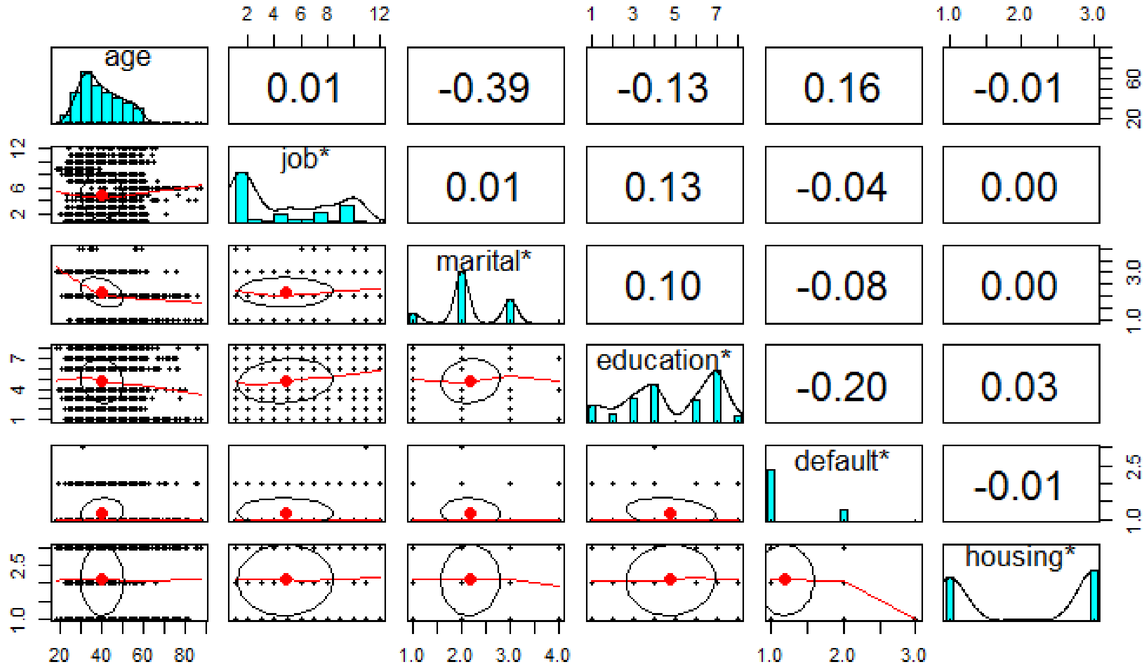
```
> #Is the employment variation rate consistent across job types?
> with(bank_additional,table(job,emp.var.rate))
               emp.var.rate
job              -3.4  -3 -2.9 -1.8 -1.7 -1.1 -0.2 -0.1 1.1 1.4
  admin.           33   4   52  199   24   23    0   92 161 424
  blue-collar       8   1    3  246    5    8    1   59 203 350
  entrepreneur      2   0    2   26    1    1    0   34  34  48
  housemaid         4   1    5    9    1    4    0   10  17  59
  management        6   3   15   71    5    5    0   62  50 107
  retired          14   3   18   28   11   10    0   11  19  52
  self-employed     4   2    6   30    4    2    0   21  34  56
  services          1   1   14  112    6    7    0   23  84 145
  student           8   1   12   18   12    6    0    4   8  13
  technician       18   1   27  122   13   13    0   59 123 315
  unemployed        5   3    6   19    4    4    0   17  13  40
  unknown           1   1    4    3    1    0    0    0  12  17
```

e. Is the employment variation rate same across education?

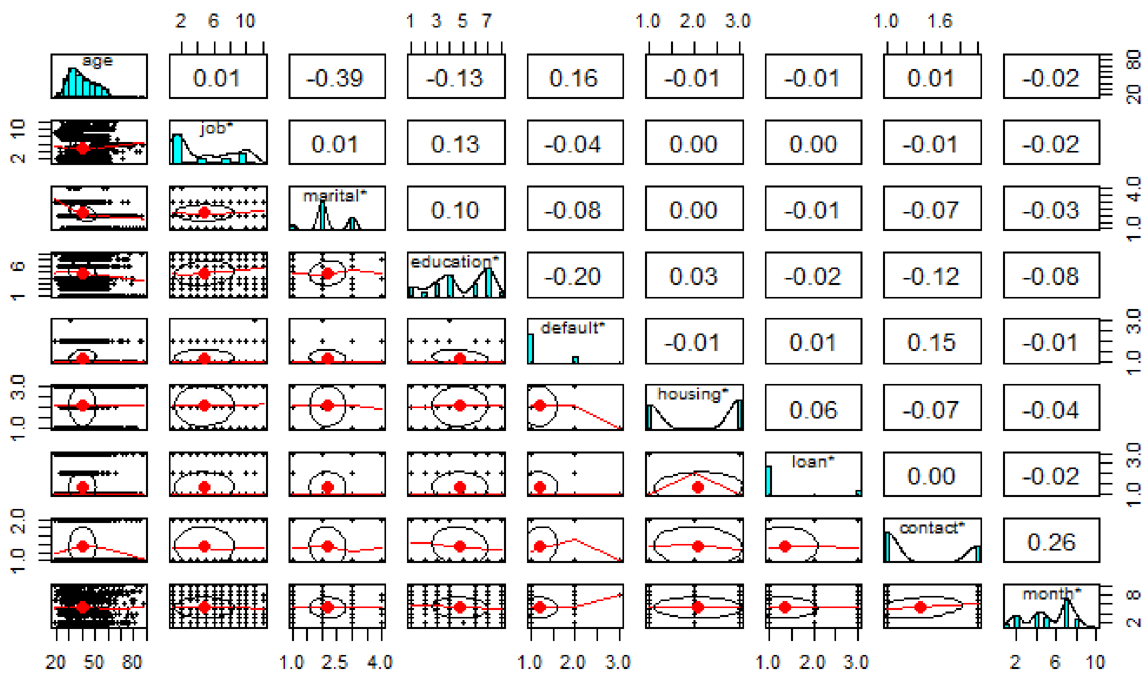
```
> #Is the employment variation rate same across education?
> with(bank_additional,table(education,emp.var.rate))
                     emp.var.rate
education             -3.4  -3 -2.9 -1.8 -1.7 -1.1 -0.2 -0.1 1.1 1.4
  basic.4y              13   2    7   83    6    8    0   28  93 189
  basic.6y               1   0    2   59    1    2    0   20  57  86
  basic.9y               8   2    4  152    5    4    0   56 127 216
  high.school           23   4   34  231   19   18    1   83 161 347
  illiterate             0   0    1    0    0    0    0    0   0   0
  professional.course   15   2   22   97   12   15    0   46 106 220
  university.degree     40   9   80  230   37   31    0  150 177 510
  unknown                4   2   14   31    7    5    0    9  37  58
```

f. Which group is more confident?

```
> pairs.panels(bank_additional[,1:6])
```



```
> pairs.panels(bank_additional[,1:9])
```



```
> summary(bank_additional)
```

```
      age             job                marital            education            d
efault
 Min.   :18.00   Length:4119        Length:4119        Length:4119        Len
gth:4119
 1st Qu.:32.00   Class :character   Class :character   Class :character   Cla
ss :character
 Median :38.00   Mode  :character   Mode  :character   Mode  :character   Mod
e  :character
 Mean   :40.11
 3rd Qu.:47.00
 Max.   :88.00
   housing               loan              contact              month
day_of_week
 Length:4119        Length:4119        Length:4119        Length:4119
Length:4119
 Class :character   Class :character   Class :character   Class :character
Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
Mode  :character



    duration         campaign           pdays           previous         poutcom
e
 Min.   :   0.0   Min.   : 1.000   Min.   :  0.0   Min.   :0.0000   Length:41
19
 1st Qu.: 103.0   1st Qu.: 1.000   1st Qu.:999.0   1st Qu.:0.0000   Class :ch
aracter
 Median : 181.0   Median : 2.000   Median :999.0   Median :0.0000   Mode  :ch
aracter
 Mean   : 256.8   Mean   : 2.537   Mean   :960.4   Mean   :0.1903
 3rd Qu.: 317.0   3rd Qu.: 3.000   3rd Qu.:999.0   3rd Qu.:0.0000
 Max.   :3643.0   Max.   :35.000   Max.   :999.0   Max.   :6.0000
  emp.var.rate      cons.price.idx  cons.conf.idx     euribor3m      nr.emplo
yed
 Min.   :-3.40000   Min.   :92.20   Min.   :-50.8   Min.   :0.635   Min.   :4
964
 1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7   1st Qu.:1.334   1st Qu.:5
099
 Median : 1.10000   Median :93.75   Median :-41.8   Median :4.857   Median :5
191
 Mean   : 0.08497   Mean   :93.58   Mean   :-40.5   Mean   :3.621   Mean   :5
166
 3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4   3rd Qu.:4.961   3rd Qu.:5
228
 Max.   : 1.40000   Max.   :94.77   Max.   :-26.9   Max.   :5.045   Max.   :5
228
      y
 Length:4119
 Class :character
 Mode  :character
```

Task 2:

1. Use the given link: Data Set.

Answer the below questions:

a. What are the assumptions of ANOVA, test it out?

b. Why the ANOVA test? Is there any other way to answer the above question?

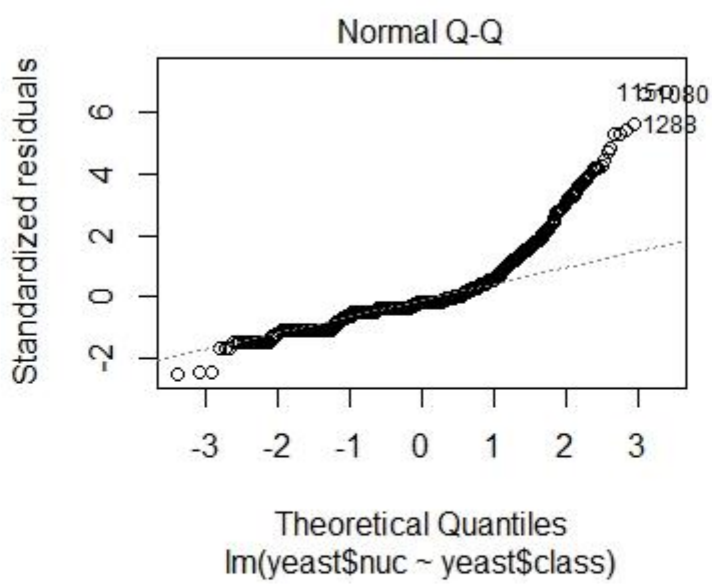
<u>Solution</u>

a)

Assumptions of ANOVA

Assumption #1. The data are Quantitative in nature and are Normally Distributed.

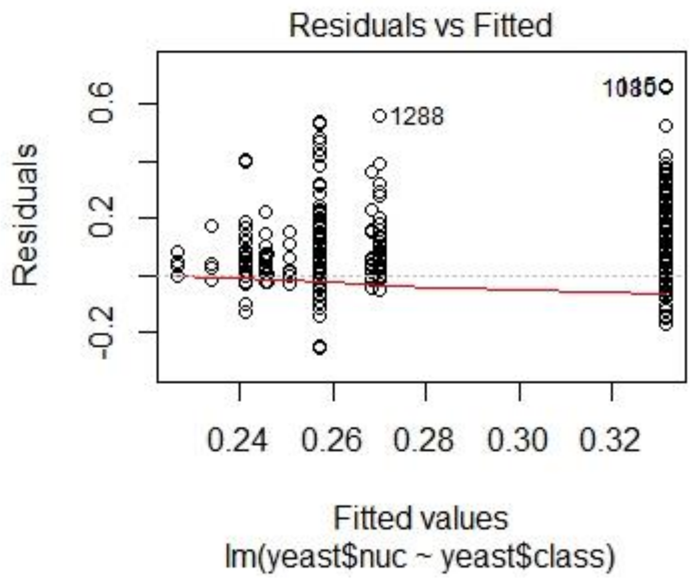
plot(data,2) # the data is not normally distibuted, its a right skewed



Assumption #2. samples are drawn from the population randomly and independently.

Assumption #3.homogeneity of variance, variances of the population

from which samples have been drawn are equal



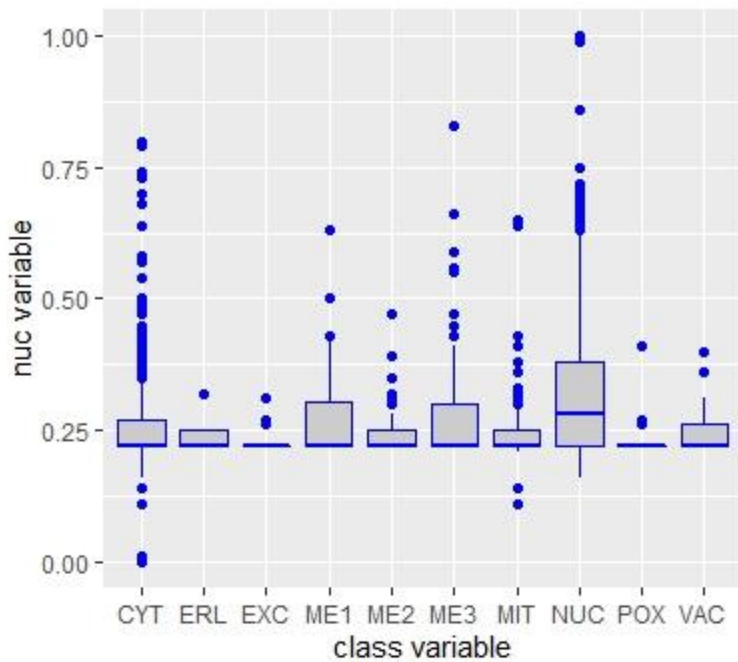

```
library(ggplot2)

ggplot(yeast, aes(x =yeast$class, y = yeast$nuc)) +

  geom_boxplot(fill = "grey80", colour = "blue") +

  scale_x_discrete() + xlab("class variable") +

  ylab("nuc variable")
```



#b. Why ANOVA test? Is there any other way to answer the above question?

to compare several population means at the same time.

kruskal.test(yeast$class, yeast$nuc) # when the data is not homoscadasticity ,

then we use non parametric test of Kruskal_Wallis test

Console:

```
> l <- l[grep('\\d\\..*:', l)]
> names(yeast) <- make.names(c(sub('.*\\d\\.\\s+(.*):.*', '\\1', l), 'class')
)
> View(yeast)
> #1. The data are Quantitative in nature and  are Normally Distributed.
> plot(data,2) # the data is not normally distibuted, its a right skewed
> # Extract the residuals
> aov_residuals <- residuals(object = data)
> # Run Shapiro-Wilk test
```

```
> shapiro.test(x = aov_residuals ) # p valu is less than 0.5

	Shapiro-Wilk normality test

data:  aov_residuals
W = 0.7959, p-value < 2.2e-16

> data:  aov_residuals
Error in data:aov_residuals : NA/NaN argument
In addition: Warning messages:
1: In data:aov_residuals :
  numerical expression has 13 elements: only the first used
2: In data:aov_residuals :
  numerical expression has 1484 elements: only the first used
> #2. samples are drawn from the population randomly and independently.
> #2. samples are drawn from the population randomly and independently.
> library(car)
> Anova(data,type = "III") # two way anaova uing car package
Anova Table (Type III tests)

Response: yeast$nuc
             Sum Sq   Df  F value    Pr(>F)
(Intercept) 30.6367    1 3046.127 < 2.2e-16 ***
yeast$class  1.9927    9   22.014 < 2.2e-16 ***
Residuals   14.8249 1474
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> #3.homogeneity of variance, variances of the population
> # from which samples have been drawn are equal
> plot(data,1) # points 1288, 115 and 1080 are outliers
> library(ggplot2)
> ggplot(yeast, aes(x =yeast$class, y = yeast$nuc)) +
+   geom_boxplot(fill = "grey80", colour = "blue") +
+   scale_x_discrete() + xlab("class variable") +
+   ylab("nuc variable")
> data<-lm(nuc~class, data=yeast)
> data<-lm(yeast$nuc~yeast$class, data=yeast)
> summary(data)

Call:
lm(formula = yeast$nuc ~ yeast$class, data = yeast)

Residuals:
     Min       1Q   Median       3Q      Max
-0.25724 -0.04818 -0.02098  0.02276  0.66832

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.257235   0.004661  55.192   <2e-16 ***
yeast$classERL -0.011235   0.045092  -0.249   0.8033
yeast$classEXC -0.030664   0.017581  -1.744   0.0813 .
yeast$classME1  0.010946   0.015821   0.692   0.4891
yeast$classME2 -0.011745   0.014796  -0.794   0.4274
yeast$classME3  0.012765   0.009134   1.398   0.1625
yeast$classMIT -0.016252   0.007934  -2.048   0.0407 *
yeast$classNUC  0.074443   0.006721  11.077   <2e-16 ***
yeast$classPOX -0.023235   0.022904  -1.014   0.3105
```

```
yeast$classVAC -0.006569   0.018894  -0.348   0.7281
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1003 on 1474 degrees of freedom
Multiple R-squared:  0.1185,   Adjusted R-squared:  0.1131
F-statistic: 22.01 on 9 and 1474 DF,  p-value: < 2.2e-16

> t.test(yeast$nuc,yeast$yeastclass)

        One Sample t-test

data:  yeast$nuc
t = 99.915, df = 1483, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.2707770 0.2816219
sample estimates:
mean of x
0.2761995

> anova(data)
Analysis of Variance Table

Response: yeast$nuc
              Df  Sum Sq  Mean Sq F value    Pr(>F)
yeast$class    9  1.9927 0.221406  22.014 < 2.2e-16 ***
Residuals   1474 14.8249 0.010058
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> res.ano <- aov(nuc~class, data= yeast)
> summary(res.ano)
              Df Sum Sq Mean Sq F value Pr(>F)
class          9  1.993 0.22141   22.01 <2e-16 ***
Residuals   1474 14.825 0.01006
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> kruskal.test(yeast$class, yeast$nuc) # when the data is not homoscadasticit
y ,

        Kruskal-Wallis rank sum test

data:  yeast$class and yeast$nuc
Kruskal-Wallis chi-squared = 151.44, df = 67, p-value = 1.771e-08

> pairwise.wilcox.test(yeast$nuc,yeast$class,p.adjust.method = "BH")

        Pairwise comparisons using Wilcoxon rank sum test

data:  yeast$nuc and yeast$class

    CYT     ERL     EXC     ME1     ME2     ME3     MIT     NUC     POX
ERL 0.96898 -       -       -       -       -       -       -       -
EXC 0.06093 0.20891 -       -       -       -       -       -       -
ME1 0.79331 0.96898 0.04065 -       -       -       -       -       -
ME2 0.63477 0.79331 0.15500 0.48277 -       -       -       -       -
ME3 0.16785 0.84965 0.00208 0.74838 0.15500 -       -       -       -
```

```
MIT 0.10389 0.74838 0.17667 0.21914 0.81416 0.00105 -       -       -
NUC < 2e-16 0.19820 1.6e-08 0.00081 4.2e-07 1.1e-08 < 2e-16 -       -
POX 0.25218 0.45796 0.81416 0.19513 0.44478 0.06152 0.48277 0.00013 -
VAC 0.89519 0.96898 0.03971 0.96898 0.52306 0.66483 0.31255 0.00091 0.19075

P value adjustment method: BH
Warning messages:
1: In wilcox.test.default(xi, xj, paired = paired, ...) :
  cannot compute exact p-value with ties
```

Alternative method

```
> # Problem 12
> yeast <- read.table(url("http://archive.ics.uci.edu/ml/machine-learning-dat
abases/yeast/yeast.data"), header = FALSE)
> names(yeast)<- c("SequenceName", "mcg", "gvh", "alm", "mit", "erl", "pox",
"vac", "nuc", "LocalizationSite")
> pca <- princomp(yeast[, 2:9], cor=T) # principal components analysis using
correlation matrix
> pc.comp <- pca$scores
> PrincipalComponent1 <- -1*pc.comp[,1] # principal component 1 scores (negat
ed for convenience)
> PrincipalComponent2 <- -1*pc.comp[,2] # principal component 2 scores (negat
ed for convenience)
> clustering.data <- cbind(PrincipalComponent1, PrincipalComponent2)
> # K-Mean Clustering
> set.seed(100)
> km <- kmeans(clustering.data, 8, iter.max = 30, nstart=30)
> km
K-means clustering with 8 clusters of sizes 199, 399, 192, 191, 130, 260, 3,
110

Cluster means:
  PrincipalComponent1 PrincipalComponent2
1           1.6051406         -0.17329450
2           0.2882285          0.01334731
3           1.0686111          1.39771690
4          -0.9430200         -1.09237604
5          -2.8601651          0.09471085
6          -0.7214210          0.76632446
7           3.6562743         -8.49636811
8           0.8085402         -1.96932237

Clustering vector:
   [1] 6 6 6 2 8 2 4 3 8 3 1 1 2 4 4 2 3 2 1 4 2 2 5 2 6 4 6 2 4 5 6 4 3 2 5
4 5 3 4 4 5 5 5 6 1 1 4 2
  [49] 6 3 2 3 2 2 4 4 6 2 2 4 2 1 2 3 6 6 3 2 2 5 2 6 4 4 8 8 4 4 8 4 4 4 4
8 5 6 3 5 2 2 1 2 1 3 6 2
  [97] 4 5 5 4 2 3 8 1 1 6 3 2 6 2 2 6 1 2 3 8 4 8 8 1 5 4 4 4 6 4 1 4 2 1 2
2 8 4 2 3 3 2 1 1 1 4 4 2
 [145] 2 3 2 2 3 2 2 3 2 2 6 3 1 4 2 5 1 1 2 3 6 3 2 2 3 3 1 6 2 1 4 2 1 2 1
8 1 2 2 4 6 2 2 3 6 3 5 5
 [193] 2 4 4 4 4 5 5 2 6 1 2 4 6 5 8 4 8 8 4 4 4 2 6 2 5 2 5 5 6 6 2 8 8 6 8
1 5 8 2 1 3 5 5 1 1 1 2 2
```

```
 [241] 2 2 2 1 3 2 6 1 6 3 2 6 1 2 6 8 1 1 1 2 1 1 6 6 1 4 1 2 6 6 1 2 6 2 1
2 5 2 2 1 6 5 3 2 2 3 6 5
 [289] 6 4 6 6 5 6 6 6 2 1 3 6 6 6 6 3 5 5 3 2 6 6 1 6 6 6 3 6 3 3 5 6 6 1 5
3 6 5 5 2 2 8 3 4 6 5 2 1
 [337] 6 3 6 8 6 2 2 4 6 2 5 1 2 3 3 1 4 3 3 2 2 1 2 2 3 2 2 5 1 2 6 3 3 3 2
2 1 2 2 2 3 6 6 1 2 1 2 3
 [385] 3 2 2 6 2 2 4 1 6 8 1 6 3 2 3 4 1 1 8 6 2 3 1 1 8 8 6 3 6 2 6 5 1 2 1
2 2 2 3 3 5 8 6 1 4 4 2 6
 [433] 4 8 6 2 2 1 1 4 1 2 6 6 3 6 2 6 6 3 2 4 4 2 4 4 2 4 4 4 2 4 2 8 2 2 1
2 6 3 5 4 6 6 6 8 3 6 2 2
 [481] 3 2 5 5 6 6 1 1 5 1 6 5 5 4 4 5 5 3 1 6 4 8 1 4 2 5 5 6 4 2 2 5 5 5 5
1 6 8 3 1 4 4 3 3 6 3 2 2
 [529] 4 2 2 3 2 3 2 4 4 3 6 2 1 3 1 1 3 1 2 4 4 4 3 5 2 8 1 2 6 6 6 4 4 6 2
1 2 4 4 6 5 6 2 1 6 2 2 6
 [577] 6 1 8 4 5 8 4 2 2 3 1 2 6 4 6 2 2 5 5 5 4 6 4 6 3 4 8 8 2 8 2 8 8 4 1
4 2 8 2 8 8 4 4 4 8 2 8 4
 [625] 4 8 4 4 2 4 4 4 6 6 5 2 1 4 3 6 6 4 4 1 3 2 6 2 4 4 2 4 8 2 2 2 1 1 2
6 2 8 6 4 5 6 2 4 1 2 4 1
 [673] 8 1 2 4 5 6 6 4 4 3 3 1 1 6 6 2 3 2 1 3 3 1 1 4 3 1 5 3 2 4 3 4 4 6 1
5 5 3 5 6 4 2 8 2 3 3 3 5
 [721] 1 1 2 6 2 2 2 4 2 2 6 4 4 2 2 2 2 6 2 5 3 4 3 1 4 5 2 2 5 6 1 2 2 2 4
4 2 1 2 2 4 2 2 2 6 8 1 2
 [769] 3 2 3 8 5 5 2 5 1 5 6 6 3 1 3 8 4 3 4 4 4 4 2 5 6 3 3 6 5 5 3 6 3 6 2
2 1 6 3 1 6 2 6 8 2 6 4 2
 [817] 1 6 6 5 4 4 4 6 2 2 8 1 8 2 5 2 1 2 1 1 2 2 3 1 1 3 2 3 2 2 3 2 2 4 4
2 5 2 1 3 8 1 2 4 2 2 6 2
 [865] 2 4 6 2 2 6 1 4 3 2 1 1 2 3 2 2 2 3 3 2 2 1 2 6 2 2 3 3 6 4 2 2 2 4 2
2 1 4 1 1 6 6 3 1 1 2 2 2
 [913] 2 6 1 4 4 3 8 2 4 2 6 4 2 2 8 8 1 1 2 8 8 8 8 8 8 1 6 1 2 2 2 2 3 2 1
1 1 2 2 8 8 2 1 3 1 1 3 3
 [961] 3 1 8 8 8 8 6 6 2 2 2 2 2 8 8 8 8 4 4 4 1 1 2 1 8 8 8 8 7 7 7 2 2 8 1
2 6 6 6 6
 [ reached getOption("max.print") -- omitted 484 entries ]

Within cluster sum of squares by cluster:
[1] 114.078257 126.152899 145.595268 144.310502 149.922267 127.815144   3.998
783 113.647111
 (between_SS / total_SS =  79.8 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss
" "betweenss"
[7] "size"         "iter"         "ifault"
> km$cluster
   [1] 6 6 6 2 8 2 4 3 8 3 1 1 2 4 4 2 3 2 1 4 2 2 5 2 6 4 6 2 4 5 6 4 3 2 5
4 5 3 4 4 5 5 5 6 1 1 4 2
  [49] 6 3 2 3 2 2 4 4 6 2 2 4 2 1 2 3 6 6 3 2 2 5 2 6 4 4 8 8 4 4 8 4 4 4 4
8 5 6 3 5 2 2 1 2 1 3 6 2
  [97] 4 5 5 4 2 3 8 1 1 6 3 2 6 2 2 6 1 2 3 8 4 8 8 1 5 4 4 4 6 4 1 4 2 1 2
2 8 4 2 3 3 2 1 1 1 4 4 2
 [145] 2 3 2 2 3 2 2 3 2 2 6 3 1 4 2 5 1 1 2 3 6 3 2 2 3 3 1 6 2 1 4 2 1 2 1
8 1 2 2 4 6 2 2 3 6 3 5 5
 [193] 2 4 4 4 4 5 5 2 6 1 2 4 6 5 8 4 8 8 4 4 4 2 6 2 5 2 5 5 6 6 2 8 8 6 8
1 5 8 2 1 3 5 5 1 1 1 2 2
 [241] 2 2 2 1 3 2 6 1 6 3 2 6 1 2 6 8 1 1 1 2 1 1 6 6 1 4 1 2 6 6 1 2 6 2 1
2 5 2 2 1 6 5 3 2 2 3 6 5
```

```
 [289] 6 4 6 6 5 6 6 6 2 1 3 6 6 6 6 3 5 5 3 2 6 6 1 6 6 6 3 6 3 3 5 6 6 1 5
3 6 5 5 2 2 8 3 4 6 5 2 1
 [337] 6 3 6 8 6 2 2 4 6 2 5 1 2 3 3 1 4 3 3 2 2 1 2 2 3 2 2 5 1 2 6 3 3 3 2
2 1 2 2 2 3 6 6 1 2 1 2 3
 [385] 3 2 2 6 2 2 4 1 6 8 1 6 3 2 3 4 1 1 8 6 2 3 1 1 8 8 6 3 6 2 6 5 1 2 1
2 2 2 3 3 5 8 6 1 4 4 2 6
 [433] 4 8 6 2 2 1 1 4 1 2 6 6 3 6 2 6 6 3 2 4 4 2 4 4 2 4 4 4 2 4 2 8 2 2 1
2 6 3 5 4 6 6 6 8 3 6 2 2
 [481] 3 2 5 5 6 6 1 1 5 1 6 5 5 4 4 5 5 3 1 6 4 8 1 4 2 5 5 6 4 2 2 5 5 5 5
1 6 8 3 1 4 4 3 3 6 3 2 2
 [529] 4 2 2 3 2 3 2 4 4 3 6 2 1 3 1 1 3 1 2 4 4 4 3 5 2 8 1 2 6 6 6 4 4 6 2
1 2 4 4 6 5 6 2 1 6 2 2 6
 [577] 6 1 8 4 5 8 4 2 2 3 1 2 6 4 6 2 2 5 5 5 4 6 4 6 3 4 8 8 2 8 2 8 8 4 1
4 2 8 2 8 8 4 4 4 8 2 8 4
 [625] 4 8 4 4 2 4 4 4 6 6 5 2 1 4 3 6 6 4 4 1 3 2 6 2 4 4 2 4 8 2 2 2 1 1 2
6 2 8 6 4 5 6 2 4 1 2 4 1
 [673] 8 1 2 4 5 6 6 4 4 3 3 1 1 6 6 2 3 2 1 3 3 1 1 4 3 1 5 3 2 4 3 4 4 6 1
5 5 3 5 6 4 2 8 2 3 3 3 5
 [721] 1 1 2 6 2 2 2 4 2 2 6 4 4 2 2 2 2 6 2 5 3 4 3 1 4 5 2 2 5 6 1 2 2 2 4
4 2 1 2 2 4 2 2 2 6 8 1 2
 [769] 3 2 3 8 5 5 2 5 1 5 6 6 3 1 3 8 4 3 4 4 4 4 2 5 6 3 3 6 5 5 3 6 3 6 2
2 1 6 3 1 6 2 6 8 2 6 4 2
 [817] 1 6 6 5 4 4 4 6 2 2 8 1 8 2 5 2 1 2 1 1 2 2 3 1 1 3 2 3 2 2 3 2 2 4 4
2 5 2 1 3 8 1 2 4 2 2 6 2
 [865] 2 4 6 2 2 6 1 4 3 2 1 1 2 3 2 2 2 3 3 2 2 1 2 6 2 2 3 3 6 4 2 2 2 4 2
2 1 4 1 1 6 6 3 1 1 2 2 2
 [913] 2 6 1 4 4 3 8 2 4 2 6 4 2 2 8 8 1 1 2 8 8 8 8 8 8 1 6 1 2 2 2 2 3 2 1
1 1 2 2 8 8 2 1 3 1 1 3 3
 [961] 3 1 8 8 8 8 6 6 2 2 2 2 2 8 8 8 8 4 4 4 1 1 2 1 8 8 8 8 7 7 7 2 2 8 1
2 6 6 6 6
 [ reached getOption("max.print") -- omitted 484 entries ]
> plot(PrincipalComponent1, PrincipalComponent2, col=km$cluster)
> points(km$centers, pch=16)
> aggregate(yeast[, 2:9],by=list(km$cluster),mean)
  Group.1       mcg       gvh       alm       mit       erl         pox
vac         nuc
1       1 0.3757286 0.3686935 0.5618593 0.2151759 0.5000000 0.004170854 0.481
8090 0.276532663
2       2 0.4792231 0.4787719 0.5196992 0.2337343 0.5000000 0.012080201 0.505
5138 0.259548872
3       3 0.3833333 0.4115104 0.4686458 0.1800000 0.5052083 0.000000000 0.527
2396 0.408750000
4       4 0.5817277 0.5768063 0.5130366 0.4321466 0.5026178 0.004345550 0.485
3927 0.240471204
5       5 0.7648462 0.7179231 0.4101538 0.3045385 0.5230769 0.006384615 0.519
6923 0.247153846
6       6 0.5357692 0.5591154 0.4424231 0.2018462 0.5096154 0.012769231 0.530
3462 0.273076923
7       7 0.3766667 0.2133333 0.9300000 0.7966667 0.5000000 0.000000000 0.160
0000 0.006666667
8       8 0.4693636 0.4452727 0.5797273 0.3632727 0.5000000 0.004545455 0.403
4545 0.215727273
> table(km$cluster, yeast$LocalizationSite)

    CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC
  1  76   0   0   0   0   3  11 105   1   3
  2 179   0   1   0   0  25  48 130  10   6
```

```
  3  45   0   0   0   4  49   3  88   0   3
  4  36   0  12   1   3   2 113  21   1   2
  5   3   5  20  43  30   6  14   2   2   5
  6  73   0   0   0  14  78  23  57   5  10
  7   3   0   0   0   0   0   0   0   0   0
  8  48   0   2   0   0   0  32  26   1   1
> #Spectral Clustering
> library(kknn)
> cl   <- specClust(clustering.data, centers=8, nn=50, iter.max=100)
> cl
K-means clustering with 8 clusters of sizes 219, 186, 172, 160, 161, 195, 235
, 156

Cluster means:
        [,1]         [,2]         [,3]        [,4]        [,5]        [,6]
[,7]        [,8]
1 -0.3859207 -0.356402209 -0.001120503 -0.18897403 -0.15872031  0.42663259 -0
.1959341 -0.34792688
2 -0.3808008 -0.010307326  0.335345170 -0.34971429  0.12569821  0.12122808 -0
.2889526  0.51087588
3 -0.3261405  0.473393062 -0.201756081  0.43978983 -0.28173634  0.29245070 -0
.1178271  0.14336792
4 -0.3253228 -0.481735595 -0.380420920  0.30376846  0.43922822  0.00971019  0
.1457893  0.17068547
5 -0.3113686 -0.308406853  0.445304695  0.27223211 -0.38520253 -0.27321926  0
.3357884 -0.03074654
6 -0.3490415  0.263465580  0.365421550  0.18599020  0.39027381 -0.23528531 -0
.1942592 -0.29581023
7 -0.3971609  0.303283449 -0.099096132 -0.38195357  0.03938998 -0.02858538  0
.5215953 -0.04244720
8 -0.3706620  0.009057016 -0.499097988 -0.08118804 -0.25164079 -0.49508121 -0
.2549182 -0.02314869

Clustering vector:
   [1] 1 1 8 7 5 2 1 7 5 3 7 6 2 1 1 7 3 7 6 1 7 7 4 7 8 1 8 2 1 4 8 5 3 2 4
4 4 7 1 1 4 4 4 8 6 6 5 2
  [49] 4 3 7 3 2 2 1 1 1 1 2 5 2 6 7 7 8 8 3 7 7 4 7 8 5 5 5 5 1 1 5 5 2 1 1
5 4 3 3 4 2 2 6 7 6 3 1 2
  [97] 4 4 4 5 1 7 5 6 2 8 7 2 8 7 2 7 3 2 3 5 5 5 5 6 4 1 1 1 8 1 6 1 2 6 1
2 5 1 7 7 3 7 6 6 6 1 1 2
 [145] 7 3 2 7 3 2 7 3 7 7 8 6 6 4 2 4 6 6 7 3 8 3 7 7 7 3 6 1 1 6 1 2 6 2 6
5 6 2 2 1 1 7 7 3 1 3 4 4
 [193] 7 1 1 5 5 4 4 7 1 6 1 5 8 4 5 5 5 5 5 1 4 1 8 7 4 2 4 4 8 8 1 5 5 8 5
6 4 5 7 2 3 4 4 2 6 6 2 2
 [241] 7 2 2 6 3 2 1 6 8 3 2 8 6 2 8 5 6 6 6 7 6 6 1 8 6 1 6 7 8 8 6 2 3 1 6
2 4 7 1 6 3 4 3 1 1 3 4 4
 [289] 8 1 8 8 4 8 4 8 7 6 7 8 8 1 3 3 4 4 3 7 1 8 6 1 1 1 3 1 3 3 4 1 8 6 4
3 8 4 4 7 2 5 3 4 8 4 1 6
 [337] 1 3 8 5 8 7 2 1 8 2 4 6 2 3 7 6 1 3 3 7 7 6 2 2 3 2 2 4 6 7 8 3 3 7 7
7 6 7 7 7 3 8 8 6 1 6 2 3
 [385] 3 2 2 7 7 2 1 6 8 5 6 8 3 7 3 1 6 6 5 8 7 3 6 6 5 5 8 3 8 7 8 4 6 2 6
2 7 7 3 3 4 5 8 6 1 1 7 8
 [433] 1 5 8 2 2 6 6 1 6 2 7 7 3 8 7 8 8 3 2 1 5 2 1 1 7 1 5 5 2 1 7 5 7 2 6
2 8 3 4 1 8 8 8 5 3 8 2 2
 [481] 3 7 4 4 4 8 6 6 4 6 8 4 4 5 1 4 4 7 6 7 1 5 6 5 6 4 4 8 5 7 2 4 4 4 4
6 7 5 3 6 5 1 6 7 8 3 1 7
```

```
 [529] 5 7 1 3 7 3 7 5 1 3 8 2 6 3 6 6 3 6 2 5 1 4 7 4 7 5 6 7 1 8 1 5 1 4 7
6 1 1 1 8 4 8 1 2 8 2 7 1
 [577] 1 6 5 4 4 5 1 2 7 3 6 7 8 1 8 2 2 4 4 4 1 7 5 1 3 1 5 5 2 5 2 5 5 1 6
5 2 5 2 5 5 1 1 1 5 2 5 5
 [625] 5 5 1 5 2 1 1 1 1 1 4 1 6 1 3 1 1 5 1 6 7 7 1 2 5 1 2 1 5 2 7 7 6 6 7
8 7 5 8 1 4 8 2 5 6 2 1 7
 [673] 2 6 7 1 4 8 1 1 1 3 3 6 6 3 8 7 3 7 6 3 3 6 6 4 3 6 4 3 7 4 7 1 5 1 6
4 4 3 4 8 5 2 5 7 3 3 3 4
 [721] 6 6 2 1 1 2 2 5 2 7 8 1 5 7 2 7 2 8 7 4 3 1 3 6 1 4 7 7 4 7 6 1 2 1 4
5 2 6 2 1 5 2 7 1 3 5 6 2
 [769] 8 7 3 5 4 4 7 4 6 4 8 8 3 6 3 5 1 3 5 5 5 5 1 4 8 3 3 7 4 4 3 3 7 8 7
7 6 7 7 6 8 2 8 5 2 4 1 7
 [817] 6 1 4 4 1 1 1 1 2 2 5 6 5 7 4 7 6 2 6 6 7 2 7 6 6 3 2 3 2 7 7 2 1 1 1
2 4 2 6 3 5 6 7 1 1 7 8 2
 [865] 7 1 1 2 1 1 6 1 3 2 6 6 7 3 2 2 2 3 3 2 1 6 2 1 2 2 3 3 3 1 7 7 2 1 2
2 6 1 6 6 7 8 3 6 6 2 7 2
 [913] 2 8 6 1 1 3 5 2 1 2 3 1 2 2 5 6 6 6 2 5 5 5 5 5 5 2 8 6 7 7 2 7 7 1 6
6 6 2 2 5 5 7 6 3 6 6 7 3
 [961] 3 6 5 5 5 5 8 8 2 7 7 2 2 5 5 5 5 5 1 1 6 6 1 2 5 5 5 5 5 5 5 2 2 5 6
2 8 8 8 4
 [ reached getOption("max.print") -- omitted 484 entries ]

Within cluster sum of squares by cluster:
[1] 70.44780 45.59679 40.60411 29.07148 32.00669 60.00491 74.81030 36.33080
 (between_SS / total_SS =  69.9 %)

Available components:

 [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withins
s" "betweenss"
 [7] "size"         "iter"         "ifault"       "eigenvalue"   "eigenvector
"  "data"
[13] "indAll"       "indUnique"    "L"            "archetype"    "call"
> plot(PrincipalComponent1, PrincipalComponent2, col=cl$cluster)
> table(cl$cluster, yeast$LocalizationSite)

    CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC
  1  72   0   2   0   2  11  93  33   2   4
  2  71   0   1   0   0   3  31  70   9   1
  3  35   0   0   0   2  50   2  80   0   3
  4   7   5  25  43  33  12  20   6   2   7
  5  52   0   7   1   2   0  67  29   2   1
  6  74   0   0   0   1   3  11 102   1   3
  7 110   0   0   0   1  29  12  75   2   6
  8  42   0   0   0  10  55   8  34   2   5
> aggregate(yeast[, 2:9],by=list(cl$cluster),mean)
  Group.1       mcg       gvh       alm       mit       erl         pox
vac       nuc
1       1 0.5599087 0.5620548 0.5038813 0.3309132 0.5022831 0.013652968 0.505
9817 0.2422831
2       2 0.4755914 0.4774731 0.5415054 0.2611290 0.5000000 0.018763441 0.492
7957 0.2497312
3       3 0.3816860 0.4130233 0.4589535 0.1778488 0.5087209 0.000000000 0.527
9651 0.4241860
4       4 0.7473125 0.7039375 0.4175625 0.3013750 0.5218750 0.005187500 0.519
2500 0.2461250
```

```
5       5 0.5042236 0.4885714 0.5668323 0.4272671 0.5000000 0.003105590 0.411
4907 0.2188199
6       6 0.3738462 0.3676923 0.5625641 0.2140000 0.5000000 0.004256410 0.481
9487 0.2754359
7       7 0.4680426 0.4667660 0.4985957 0.2000000 0.5000000 0.007063830 0.517
4043 0.2804681
8       8 0.5244231 0.5530769 0.4301282 0.1937821 0.5096154 0.005320513 0.536
2179 0.2767949
> #Hierarchical Clustering
> d_yeast<- dist(clustering.data)
> hclusters <- hclust(d_yeast, method = "average")
> clusterCut <- cutree(hclusters, 8)
> clusterCut
   [1] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 3 2 1 1 2
2 2 1 1 1 2 2 2 1 1 1 4 1
  [49] 2 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1
1 2 3 1 2 1 1 1 1 1 3 1 1
  [97] 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 3 1 2 1 5 1 2 1 2 1 2 1 1 2 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1
 [145] 1 3 1 1 3 1 1 3 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 2 2
 [193] 1 1 2 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1 1 2 1 2 2 2 1 1 1 1 1 1
1 2 1 1 1 3 2 2 1 1 1 1 1
 [241] 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1
1 2 1 1 1 1 2 1 1 1 1 2 2
 [289] 1 1 1 1 2 1 2 1 1 1 1 1 2 1 1 1 2 2 3 1 1 2 1 1 1 1 1 1 1 3 2 1 1 1 2
3 1 2 2 1 1 1 1 2 1 2 1 1
 [337] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 3 1 1 1 1 1 3 1 1 2 1 1 1 1 1 1 1
1 1 1 1 1 3 1 1 1 1 1 1 3
 [385] 1 1 1 1 1 1 2 1 1 5 1 2 3 1 1 2 1 1 1 1 1 1 1 1 5 5 1 1 1 1 2 2 1 1 1
1 1 1 1 1 2 1 1 1 1 1 1 1
 [433] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 2 1 1 1 1 5 1 2 1 1
 [481] 1 1 2 2 2 1 1 1 2 1 1 2 2 1 1 2 2 1 1 1 1 1 1 4 1 2 2 1 4 1 1 2 2 2 2
1 1 1 1 1 1 2 1 1 1 1 1 1
 [529] 1 1 1 3 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 5 1 1 1 2 1 1 1 2 1
1 1 1 1 2 2 1 1 1 1 1 1 1
 [577] 1 1 1 4 2 1 1 1 1 3 1 1 2 2 1 1 1 2 2 2 1 1 1 1 3 2 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1
 [625] 2 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1
1 1 1 1 1 2 1 1 1 1 1 1 1
 [673] 1 1 1 2 2 1 1 1 1 3 1 1 1 1 1 1 3 1 1 3 3 1 1 2 1 1 2 1 1 2 1 2 4 1 1
2 2 1 2 2 2 1 1 1 3 3 3 2
 [721] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 3 1 1 1 1 2 1 1 2 1 1 1 1 1 2
1 1 1 1 1 2 1 1 1 3 1 1 1
 [769] 1 1 3 1 2 2 1 2 1 2 2 1 1 1 3 1 2 1 1 4 4 4 1 2 1 3 1 1 2 2 3 1 1 1 1
1 1 1 1 1 1 1 1 5 1 2 1 1
 [817] 1 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 3 1 3 1 1 1 1 1 1 1
1 2 1 1 3 1 1 1 2 1 1 2 1
 [865] 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1
 [913] 1 1 1 1 1 3 1 1 1 1 1 2 1 1 5 1 1 1 1 1 1 1 1 5 5 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 3
 [961] 3 1 5 5 1 1 1 1 1 1 1 1 1 5 1 1 1 1 2 2 1 1 1 1 1 1 5 5 6 6 6 1 1 1 1
1 1 2 2 2
 [ reached getOption("max.print") -- omitted 484 entries ]
> table(clusterCut, yeast$LocalizationSite)
```

```
clusterCut CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC
         1 411   0   4   0  14 130 194 358  17  21
         2  16   5  25  43  36  19  46  10   3   9
         3  16   0   0   0   0  14   2  49   0   0
         4   0   0   6   1   1   0   0   2   0   0
         5  17   0   0   0   0   0   1   8   0   0
         6   3   0   0   0   0   0   0   0   0   0
         7   0   0   0   0   0   0   1   1   0   0
         8   0   0   0   0   0   0   0   1   0   0
> aggregate(yeast[, 2:9],by=list(clusterCut),mean)
  Group.1       mcg       gvh       alm       mit       erl         pox
vac         nuc
1       1 0.4715405 0.4762228 0.5104178 0.2522715 0.5013055 0.008964317 0.500
7659 0.268398607
2       2 0.7120283 0.6790566 0.4296698 0.3211792 0.5188679 0.003915094 0.520
5189 0.248867925
3       3 0.3545679 0.3871605 0.4807407 0.1697531 0.5185185 0.000000000 0.523
7037 0.497530864
4       4 0.7750000 0.7390000 0.5210000 0.4280000 0.5000000 0.000000000 0.366
0000 0.241000000
5       5 0.4115385 0.4076923 0.5992308 0.3080769 0.5000000 0.000000000 0.321
9231 0.200384615
6       6 0.3766667 0.2133333 0.9300000 0.7966667 0.5000000 0.000000000 0.160
0000 0.006666667
7       7 0.2350000 0.1700000 0.7000000 0.3100000 0.5000000 0.000000000 0.490
0000 0.230000000
8       8 0.6600000 0.4300000 0.5700000 0.6000000 0.5000000 0.000000000 0.190
0000 0.330000000
> plot(PrincipalComponent1, PrincipalComponent2, col=clusterCut)
> ### main anova
> yeast <- read.table('https://archive.ics.uci.edu/ml/machine-learning-databa
ses/yeast/yeast.data', stringsAsFactors = FALSE)
> char <- readLines('https://archive.ics.uci.edu/ml/machine-learning-database
s/yeast/yeast.names')
> char<-char[(grep('^7', char) + 1):(grep('^8', char) - 1)]
> char <- char[grep('\\d\\..*:', char)]
> names(yeast) <- make.names(c(sub('.*\\d\\.\\s+(.*):.*', '\\1', char), 'clas
s'))
> View(yeast)
> library(ggplot2)
> ggplot(yeast, aes(x =yeast$class, y = yeast$nuc)) +
+   geom_boxplot(fill = "grey80", colour = "blue") +
+   scale_x_discrete() + xlab("class variable") +
+   ylab("nuc variable")
> data<-lm(nuc~class, data=yeast)
> data<-lm(yeast$nuc~yeast$class, data=yeast)
> summary(data)

Call:
lm(formula = yeast$nuc ~ yeast$class, data = yeast)

Residuals:
     Min       1Q   Median       3Q      Max
-0.25724 -0.04818 -0.02098  0.02276  0.66832

Coefficients:
```

```
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)      0.257235   0.004661  55.192   <2e-16 ***
yeast$classERL -0.011235   0.045092  -0.249   0.8033    
yeast$classEXC -0.030664   0.017581  -1.744   0.0813 .  
yeast$classME1  0.010946   0.015821   0.692   0.4891    
yeast$classME2 -0.011745   0.014796  -0.794   0.4274    
yeast$classME3  0.012765   0.009134   1.398   0.1625    
yeast$classMIT -0.016252   0.007934  -2.048   0.0407 *  
yeast$classNUC  0.074443   0.006721  11.077   <2e-16 ***
yeast$classPOX -0.023235   0.022904  -1.014   0.3105    
yeast$classVAC -0.006569   0.018894  -0.348   0.7281    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1003 on 1474 degrees of freedom
Multiple R-squared:  0.1185,	Adjusted R-squared:  0.1131 
F-statistic: 22.01 on 9 and 1474 DF,  p-value: < 2.2e-16

> anova(data) # one way anova, single variable
Analysis of Variance Table

Response: yeast$nuc
              Df  Sum Sq  Mean Sq F value    Pr(>F)    
yeast$class    9  1.9927 0.221406  22.014 < 2.2e-16 ***
Residuals   1474 14.8249 0.010058                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> library(car)
> Anova(data,type = "III") # two way anaova uing car package
Anova Table (Type III tests)

Response: yeast$nuc
             Sum Sq   Df  F value    Pr(>F)    
(Intercept) 30.6367    1 3046.127 < 2.2e-16 ***
yeast$class  1.9927    9   22.014 < 2.2e-16 ***
Residuals   14.8249 1474                       
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> t.test(yeast$nuc,yeast$yeastclass) # no need of t test here

	One Sample t-test

data:  yeast$nuc
t = 99.915, df = 1483, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.2707770 0.2816219
sample estimates:
mean of x 
0.2761995 

> confint(data) # confidence interval test at 95%
                      2.5 %       97.5 %
(Intercept)     0.248092991  0.2663778510
yeast$classERL -0.099685772  0.0772149300
yeast$classEXC -0.065149950  0.0038219647
yeast$classME1 -0.020087712  0.0419805058
```

```
yeast$classME2 -0.040769281  0.0172788313
yeast$classME3 -0.005151997  0.0306811542
yeast$classMIT -0.031814209 -0.0006894197
yeast$classNUC  0.061259865  0.0876259359
yeast$classPOX -0.068163744  0.0216929015
yeast$classVAC -0.043630381  0.0304928715
> require(ggplot2)
> mod<-data.frame(Fitted = fitted(data),
+                 Residuals = resid(data), Treatment = yeast$class)
> ggplot(mod, aes(Fitted, Residuals, colour = Treatment)) + geom_point()
>
```

yeast <- read.table(url("http://archive.ics.uci.edu/ml/machine-learning-databases/yeast/yeast.data"), header = FALSE)

names(yeast)<- c("SequenceName", "mcg", "gvh", "alm", "mit", "erl", "pox", "vac", "nuc", "LocalizationSite")

pca <- princomp(yeast[, 2:9], cor=T) # principal components analysis using correlation matrix

pc.comp <- pca$scores

PrincipalComponent1 <- -1*pc.comp[,1] # principal component 1 scores (negated for convenience)

PrincipalComponent2 <- -1*pc.comp[,2] # principal component 2 scores (negated for convenience)

clustering.data <- cbind(PrincipalComponent1, PrincipalComponent2)

K-Mean Clustering

set.seed(100)

km <- kmeans(clustering.data, 8, iter.max = 30, nstart=30)

km

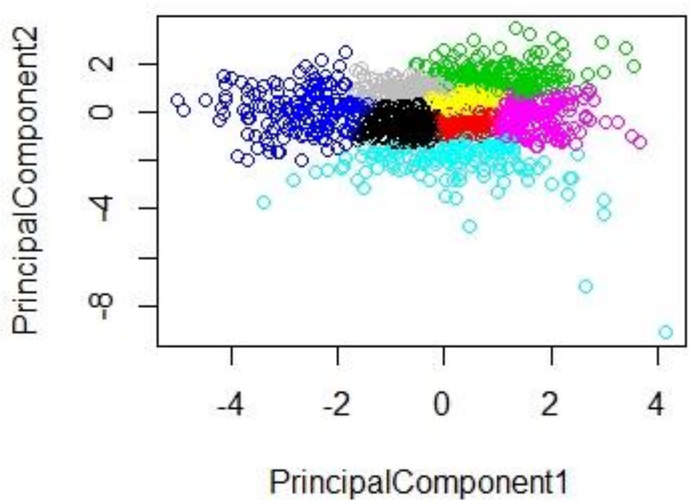
km$cluster

plot(PrincipalComponent1, PrincipalComponent2, col=km$cluster)

points(km$centers, pch=16)

aggregate(yeast[, 2:9],by=list(km$cluster),mean)

table(km$cluster, yeast$LocalizationSite)



#Spectral Clustering

library(kknn)

cl   <- specClust(clustering.data, centers=8, nn=50, iter.max=100)

cl

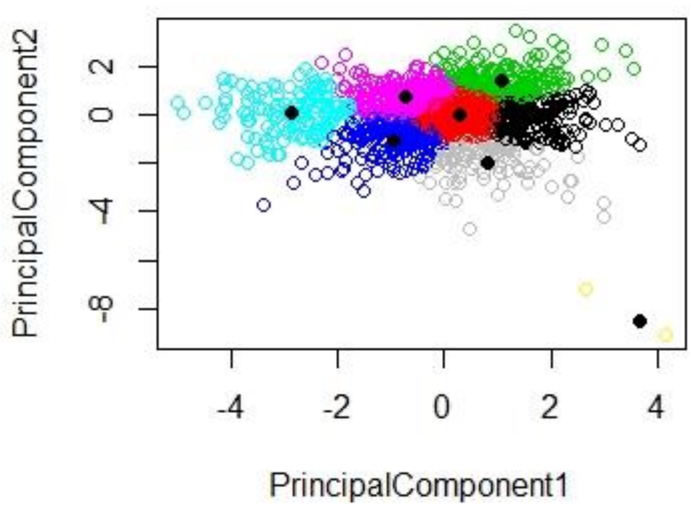
plot(PrincipalComponent1, PrincipalComponent2, col=cl$cluster)

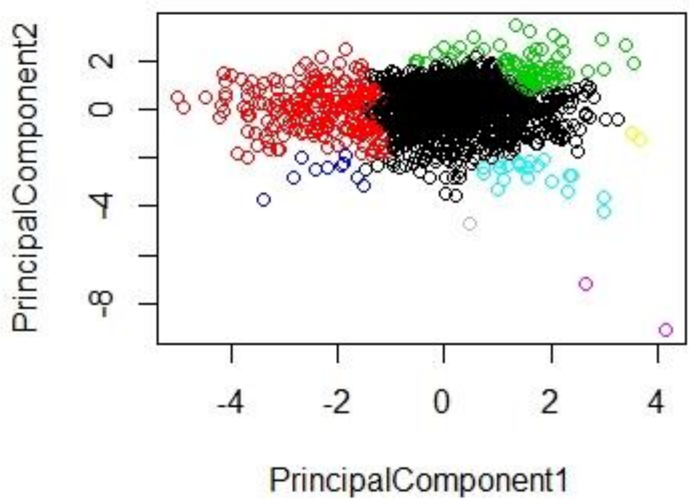
```
table(cl$cluster, yeast$LocalizationSite)

aggregate(yeast[, 2:9],by=list(cl$cluster),mean)

#Hierarchical Clustering
d_yeast<- dist(clustering.data)
hclusters <- hclust(d_yeast, method = "average")
clusterCut <- cutree(hclusters, 8)
clusterCut
table(clusterCut, yeast$LocalizationSite)
aggregate(yeast[, 2:9],by=list(clusterCut),mean)

plot(PrincipalComponent1, PrincipalComponent2, col=clusterCut)
```





main anova

yeast <- read.table('https://archive.ics.uci.edu/ml/machine-learning-databases/yeast/yeast.data', stringsAsFactors = FALSE)

char <- readLines('https://archive.ics.uci.edu/ml/machine-learning-databases/yeast/yeast.names')

char<-char[(grep('^7', char) + 1):(grep('^8', char) - 1)]

char <- char[grep('\\d\\..*:', char)]

names(yeast) <- make.names(c(sub('.*\\d\\.\\s+(.*):.*', '\\1', char), 'class'))

View(yeast)

#a. Perform ANOVA test on the discriminant analysis scores of nuclear localization signals of both nuclear

#and non-nuclear proteins by class variables (Target).

```
library(ggplot2)

ggplot(yeast, aes(x =yeast$class, y = yeast$nuc)) +
  geom_boxplot(fill = "grey80", colour = "blue") +
  scale_x_discrete() + xlab("class variable") +
  ylab("nuc variable")



data<-lm(nuc~class, data=yeast)
data<-lm(yeast$nuc~yeast$class, data=yeast)
summary(data)


anova(data) # one way anova, single variable


library(car)
Anova(data,type = "III") # two way anaova uing car package


t.test(yeast$nuc,yeast$yeastclass) # no need of t test here
```

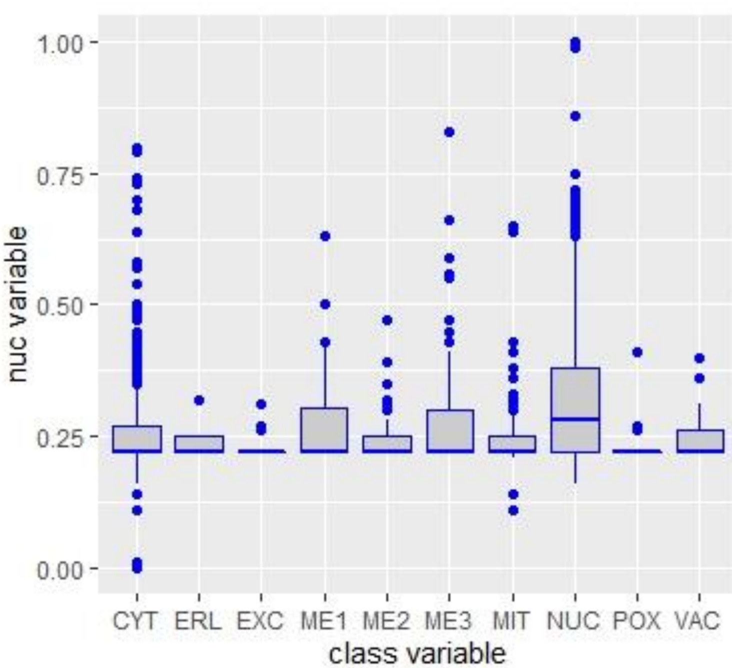
#b. Which class is significantly different from others?

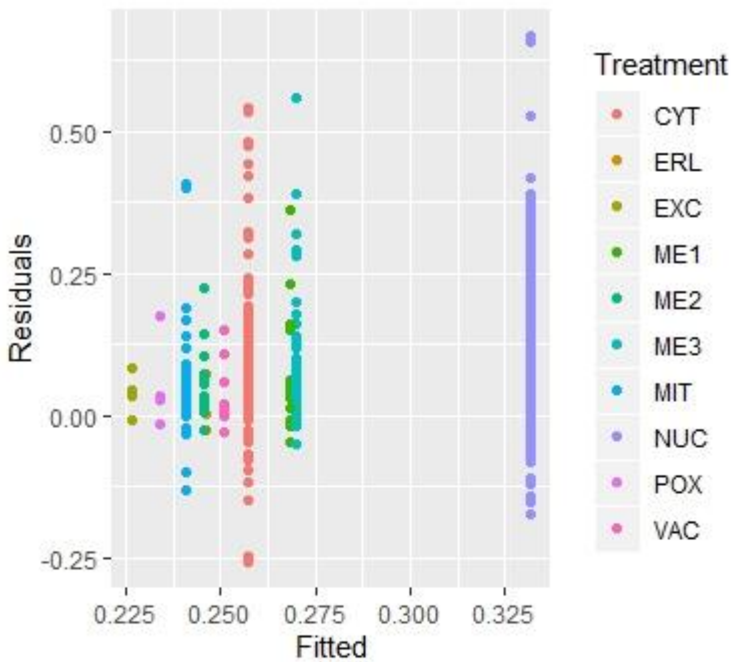
```
confint(data) # confidence interval test at 95%

require(ggplot2)
mod<-data.frame(Fitted = fitted(data),
        Residuals = resid(data), Treatment = yeast$class)

ggplot(mod, aes(Fitted, Residuals, colour = Treatment)) + geom_point()
```

class NUC is significantly different from others





Task 3:

1. Use the given link: Data Set.

Remarks – Solution (wrong link of data set- questions pertain to crime data set while the link is leading to Yeast data set) however, attempt has been made to work on crime data set.

Answer the below questions:

a. Visualize the correlation between all variables in a meaningful and clear way of representing.

Find out top 3 reasons for having more crime in a city.

b. What is the difference between covariance and correlation? Take an example from this

dataset and show the differences if any?

Solution:

library(readr)

Crimes<-read.csv("C:/Users/user/Desktop/Data Analytics/Assignments/session 11 to 15/Communities data.csv")

View(Crimes)

names(Crimes) <- c("Case", "Number", "Date", "Block", "IUCR", "Primary Type",

"Description", "Location Desc", "Arrest", "Domestic", "Beat", "District", "Ward",

"Community Area","FBI Code", "X Coordinate", "Y Coordinate", "Year", "Updated On",

"Latitude", "Longitude", "Location")

head(Crimes)

str(Crimes)

#a. Visualize the correlation between all variables in a meaningful and clear way of representing.

library(dplyr)

Crimes <- na.omit(Crimes)

names(Crimes)

c <- cor(Crimes[c(11,12,13,14,18,20,21)])

c

psych::cor.plot(c)

 # a.Find out top 3 reasons for having more crime in a city.

x <- as.data.frame(table(Crimes$Description))

x[order(x$Freq, decreasing = T)[1:3],]

b. What is the difference between covariance and correlation, take an example from this dataset and show the differences if any?

```
correlation <- cor(Crimes[c(11,12,13,14,18,20,21)])

correlation

psych::cor.plot(correlation)

covariance <- cov(Crimes[c(11,12,13,14,18,20,21)])

covariance

psych::cor.plot(covariance)
```

Conclusion/Interpretation:

Co-Variance is a systematic relationship between a pair of random variables wherein a change in one variable reciprocated by an equivalent change in another variable. Measure of correlation, Lie between $-\infty$ and $+\infty$. Change in scale affects covariance

Correlation is statistical measure that indicates how strongly two variables are related. Scaled version of covariance, Lie between -1 and +1, Change in scale does not affect the correlation.

Unit free measure

Correlation is a special case of covariance which can be obtained when the data is standardized.

Task 4:

Problem- prediction of the number of comments in the upcoming 24 hours on those blogs, the train data was generated from different base times that may temporally overlap. Therefore, if you simply split the train into disjoint partitions, the underlying time intervals may overlap. Therefore, the you should use the provided, temporally disjoint train and test splits to ensure that the evaluation is fair.

a) Read the dataset and identify the right features.

b) Clean dataset, impute missing values and perform exploratory data analysis.

c) Visualize the dataset and make inferences from that.

d) Perform any 3 hypothesis tests using columns of your choice, make conclusions.

e. Create a linear regression model to predict the number of comments in the next 24 hours

(relative to basetime)

f. Fine tune the model and represent important features

g. Interpret the summary of the linear model

h. Report the test accuracy vs. the training accuracy

i. Interpret the final model coefficients

j. Plot the model result and compare it with assumptions of the model

Solutions:

library(data.table)

library(foreach)

library(dplyr)
library(readr)

blogData_train <- read_csv("blogData_train.csv")

View(blogData_train)

> # retrieve filenames of test sets

> test_filenames = list.files(pattern = "blogData_test")

>

> # load and combine dataset

> train = fread("blogData_train.csv")

> fbtest = foreach(i = 1:length(test_filenames), .combine = rbind) %do% {

+ temp = fread(test_filenames[i], header = F)

+ }

> # Assign variable names to the train and test data set

> colnames(blogData_train) <-
c("plikes","checkin","talking","category","d5","d6","d7","d8","d9","d10","d11
","d12",
+
"d13","d14","d15","d16","d17","d18","d19","d20","d21","d22","d23","d24","d25"
,"d26",
+

```
"d27","d28","d29","cc1","cc2","cc3","cc4","cc5","basetime","postlength","post
shre",
+
"postpromo","Hhrs","sun","mon","tue","wed","thu","fri","sat","basesun","basem
on",
+
"basetue","basewed","basethu","basefri","basesat","target")
> dim(blogData_train)
 > dim(fbtest)
> View(blogData_train)
 > View(fbtest)
> str(blogData_train)
 > str(fbtest)
> train <- blogData_train; test <- fbtest
> head(train); head(test)
> # making the data tidy by constructing single collumn for post publish day
> train$pubday<- ifelse(train$sun ==1, 1, ifelse(train$mon ==1, 2,
ifelse(train$tue ==1, 3,
+
ifelse(train$wed ==1, 4, ifelse(train$thu ==1, 5, ifelse(train$fri ==1, 6,
+
ifelse(train$sat ==1, 7, NA)))))))
> # making the data tidy by constructing single collumn for base day
> train$baseday<- ifelse(train$basesun ==1, 1, ifelse(train$basemon ==1, 2,
ifelse(train$basetue ==1, 3,
+
ifelse(train$basewed ==1, 4, ifelse(train$basethu ==1, 5,
+
ifelse(train$basefri ==1, 6, ifelse(train$basesat ==1, 7, NA)))))))
```

b. Clean dataset, impute missing values and perform exploratory data analysis.

```
distinct(train) # removing overlapping observations if any
dim(train)
sapply(train, function(x) sum(is.na(x))) # no missing values
correlation <- cor(train,y = NULL, use = "everything",
 method = c("pearson", "kendall", "spearman"))
corr <- as.data.frame(reshape::melt(correlation))
corr <- corr%>%filter(X1 == "target" & value != 1 & value > 0.32 & value > -0.32)
corr # good corelations with target variable
library(corrplot)
corrplot.mixed(cor(train[,c(30:32)]))
```

# Total comments are strongly correlated to correlated with cc3(comments in last 48 to last 24 hours relative to base date/time)
df <- train
melt_df <- melt(df)
library(ggplot2)
# Distribution of all the Variables - Histogram
ggplot(melt_df, aes(x=value, fill = variable))+
 geom_histogram(bins=10, color = "Blue")+
 facet_wrap(~variable, scales = 'free_x')
df <- log(train[1:39])
par(mfrow=c(1,1))
Conclusion/Interpretation:
There is a good corelations with target variable
Total comments are strongly correlated to correlated cc3(comments in last 48 to last 24 hours relative to base date/time)

c. Visualize the dataset and make inferences from that.

barplot(table(train$target, train$pubday), col = heat.colors(7),
 xlab = "Weekday", ylab = "Number of comments",
 main = "Number of comments Vs. Weekday")

library(car)
# number of comments vs Post Likes
scatterplot(train$plikes, train$target , col = "Blue",
 xlab = "Page Likes", ylab = "Number of comments",
 main = "Number of comments Vs. Pagelikes",
 xlim = c(0,10000000), ylim = c(0,400))
abline(lm(plikes~target, data = train), col = "red")

Number of comments Vs Post length
scatterplot(train$postlength, train$target , col = "Red",
 xlab = "Post Length", ylab = "Number of comments",
 main = "Number of comments Vs. Psot Length",
 ylim = c(0,400), xlim = c(0,5000))
abline(lm(postlength~target, data = train), col= "blue")
hist(train$target, breaks = 1000, xlim = c(0,10))

d. Perform any 3 hypothesis tests using columns of your choice, make conclusions.

1. **# Ho: Mean difference bet comments across the publish day is not significant**
day <- aov(target~pubday, data = train)

summary(day)

2. **# Ho: Difference between Mean comments within cc2 and cc4 is not significant**

cc2 <- t.test(x=train$cc2, y=train$cc4, paired = FALSE, alternative = "two.sided", mu=0)
cc2

3. **# Ho: Difference between Mean comments within cc1 and cc3 is not significant**

cc3 <- t.test(x=train$cc1, y=train$cc3, paired = FALSE, alternative = "two.sided", mu=0)
cc3

f. Fine tune the model and represent important features

```
final_model <- lm(target ~ talking + d5 + d7 + d8 + d10 + d11 +
 d12 + d13 + d16 + d17 + d19 + d20 + d22 + d23 +
 cc1 + cc2 + cc3 + cc4 + basetime + postshre + Hhrs, data = train)
summary(final_model)
prediction <- predict(final_model, test)
predicted <- data.frame(cbind(actuals = test$target, prediction = prediction))
predicted$prediction <- ifelse(prediction<0, 0, round(prediction,0))
cor(predicted)
View(predicted)
```

g. Interpret the summary of the linear model.

summary(final_model)

h. Report the test accuracy vs. the training accuracy

```
# test accuracy
round(accuracy(predicted$prediction,predicted$actuals),3)
prediction <- predict(final_model, test)
predicted <- data.frame(cbind(actuals = test$target, prediction = prediction))
predicted$prediction <- ifelse(prediction<0, 0, round(prediction,0))
min_max_accuracy <- mean(apply(predicted, 1, min) / apply(predicted, 1, max))
# training accuracy
round(accuracy(predicted$prediction,predicted$actuals),3)
prediction <- predict(final_model, train)
predicted <- data.frame(cbind(actuals = train$target, prediction = prediction))
predicted$prediction <- ifelse(prediction<0, 0, round(prediction, 0))
min_max_accuracy <- mean(apply(predicted, 1, min) / apply(predicted, 1, max))
```

Task 5 :

This is based on SLR data and will be updated on the basis of UCI data later .

```
#Assignment Session 11-15 task 5

#Problem
#a. Predict the no of comments in next H hrs
#b. Use regression technique
#c. Report the training accuracy and test accuracy


#Answers
#a) & b)
#reading the dataset and viewing
slr
slr1<- slr
View(slr1)

#features
dim(slr1)
str(slr1)

library(psych)
describe(slr1)
summary(slr1)

#visualization
hist(slr1$Advt ,xlab = "advt", ylab = "Frequency",main="Histogram of advt",col="red")
hist(slr1$Sales ,xlab = "sales", ylab = "Frequency",main="Histogram of sales",col="blue")

plot(slr1$Advt,slr1$Sales)

#***NOTE***
#using linear regression model technique
#using slr1 dataset
#linear regression model
model<- lm(slr1$Advt~slr1$Sales)
model

#Important features

#multiple r squared value
#p value of slope test
#F stats
```

```
#predicting
Pred<- predict(lm(slr1$Sales~slr1$Advt))
Pred

pred<- predict(model,newdata= slr1,type = "response")
table(slr1$Advt,pred>= 0.5)

conf<- table(slr1$Advt,pred)
conf

predict(model)
Pred=predict(model)
slr1$predicted =NA
slr1$predicted =Pred

slr1$error =model$residuals

#verfify residuals
error<- residuals(lm(slr1$Sales~slr1$Advt))
error

summary(error)

#check and interpreting the summary
summary(model)

#**NOTE**
#Interpreting
#thus by multiple r squared value we see our model is good
#also our p value of slope test is <0.05 so good for our model
#adjusted r squared value is also good 0.891
#f stats value of 90.93 suggest our model is good and also its p value is <0.05
#our model accuracy is 0.9009 which is good

#result of all of our models
summary(model)
summary(model1)
summary(model2)

#model coefficients
model
model1
```

```
model2

slr1$coefficients<- NA
slr1$coefficients<- model$coefficients
slr1$coefficients


#c)
#test and training accuracy
#dataset slr1

set.seed(1)
split<- sample.split(slr1$Advt,SplitRatio = 0.70)
slr1Train <- subset(slr1,split == TRUE)
slr1Test<- subset(slr1, split == FALSE)

#train
model1<- lm(slr1Train$Advt~slr1Train$Sales)
model1

summary(model1)
#accuracy is 0.926

#test
model2<- lm(slr1Test$Advt~slr1Test$Sales)
model2

summary(model2)
#accuracy is 0.871
```