Progetto - Data Mining and Organization

Ubaldo Puocci

3/17/2020

Introduzione

Il progetto da me scelto comprende il dataset "Coorti 2010- 2016 studenti di tre CdS Scuola SMFN - produttivita I anno + esame di matematica" e prevede l'applicazione di algoritmi di clustering per analizzare i dati proposti.

Il dataset si presenta in questo modo:

```
dataset = read.csv("./Data/dataset.csv")
summary(dataset)
```

```
##    CdS          Coorte     Genere    Voto_test     Crediti_convoto
##  CdS1:435   Min.   :2010   F:380   Min.   :-1.00   Min.   : 0.00
##  CdS2:458   1st Qu.:2011   M:852   1st Qu.:12.00   1st Qu.:15.00
##  CdS3:339   Median :2013           Median :16.62   Median :33.00
##             Mean   :2013           Mean   :16.09   Mean   :30.75
##             3rd Qu.:2015           3rd Qu.:21.00   3rd Qu.:48.00
##             Max.   :2016           Max.   :25.00   Max.   :69.00
##
##  Crediti_totali    Voto_medio   Scuola_provenienza        Esame_Matematica
##  Min.   :  3.0   Min.   : 0.0   LS     :698                      :261
##  1st Qu.: 18.0   1st Qu.:22.0   IT     :246        EsameMatematica:970
##  Median : 36.0   Median :25.0   LC     : 91        MATEMATICA I   :  1
##  Mean   : 34.1   Mean   :22.1   TC     : 67
##  3rd Qu.: 51.0   3rd Qu.:27.0   XX     : 42
##  Max.   :123.0   Max.   :31.0   AL     : 32
##                                 (Other): 56
##  Voto_Matematica Crediti_Matematica
##  Min.   : 0.00   Min.   : 1.0
##  1st Qu.:23.00   1st Qu.:12.0
##  Median :25.00   Median :12.0
##  Mean   :25.04   Mean   :12.8
##  3rd Qu.:28.00   3rd Qu.:15.0
##  Max.   :31.00   Max.   :15.0
##  NA's   :261     NA's   :261
```

Per ogni studente abbiamo quindi le seguenti informazioni:

- Corso di Laurea, con 3 possibili opzioni
- Coorte di iscrizione, dal 2010 al 2016 compresi
- Il genere
- Il voto del test d'ingresso obbligatorio per gli studenti iscritto alla scuola di SMFN.
- Crediti che corrispondono ad esami con attribuzione di voto
- Crediti che corrispondono ad esami con o senza attribuzione di voto
- Il voto medio che lo studente ha ottenuto negli esami da esso superati

- La scuola di provenienza prima dell'iscrizione all'Università
- Se lo studente ha superato o meno l'esame di Analisi I o Matematica I al primo anno
- Il voto conseguito al suddetto esame
- Il numero di crediti conseguiti con il superamento del medesimo esame

Preprocessing con R

Prima di poter applicare i classici algoritmi di clustering, è necessario preparare i dati per modificarne alcune caratteristiche senza alterare od eliminare alcuna informazione contenuta nel dataset.

La colonna `Scuola_provenienza` presenta valori ricondicibili alla seguente legenda:

- LS = Liceo Scientifico
- LC = Liceo Classico
- IT = Istituto Tecnico Industriale
- TC = Istituto Tecnico Commerciale
- IP = Istituto Professionale
- AL, IA, IPC, LL, XX, o cella vuota = Altro

ed è quindi necessario modificare il dato per far sì che questo sia rappresentato nel dataset:

```
summary(dataset$Scuola_provenienza)
```

```
##      AL  IA  IP IPC  IT  LC  LL  LS  TC  XX
##  23  32   4  12   1 246  91  16 698  67  42
```

```
dataset$Scuola_provenienza = as.character(dataset$Scuola_provenienza)
dataset$Scuola_provenienza = with(dataset,
                                  ifelse(
                                    Scuola_provenienza %in%
                                    c('AL', 'IA', 'IPC', 'LL', 'XX', ''),
                                    'Altro',
                                    Scuola_provenienza
                                  ))

dataset$Scuola_provenienza = as.factor(dataset$Scuola_provenienza)
summary(dataset$Scuola_provenienza)
```

```
## Altro    IP    IT    LC    LS    TC
##   118    12   246    91   698    67
```

La prossima colonna da analizzare è `Esame_matematica`.

```
summary(dataset$Esame_Matematica)
```

```
##                 EsameMatematica    MATEMATICA I
##             261             970               1
```

Questa colonna ci da un'informazione molto importante: se lo studente ha superato o meno l'esame di profitto di Analisi I o Matematica I al primo anno. Una cella vuota sta a significare che lo studente non ha superato l'esame. Dobbiamo quindi modificare il dato per meglio spiegare questo fenomeno, ignorando il nome dell'esame poiché non è di nostro interesse al momento.

```
dataset$Esame_Matematica = as.character(dataset$Esame_Matematica)
dataset$Esame_Matematica = with(dataset,
                                ifelse(Esame_Matematica %in% (''),
                                'Non superato', Esame_Matematica))
dataset$Esame_Matematica = with(dataset,
                                ifelse(Esame_Matematica %in% ('MATEMATICA I'),
```

```
                                'EsameMatematica', Esame_Matematica))
dataset$Esame_Matematica = as.factor(dataset$Esame_Matematica)
summary(dataset$Esame_Matematica)
```

```
## EsameMatematica    Non superato
##             971              261
```

Un attributo direttamente legato al precedente è `Voto_Matematica`.

```
summary(dataset$Voto_Matematica)
```

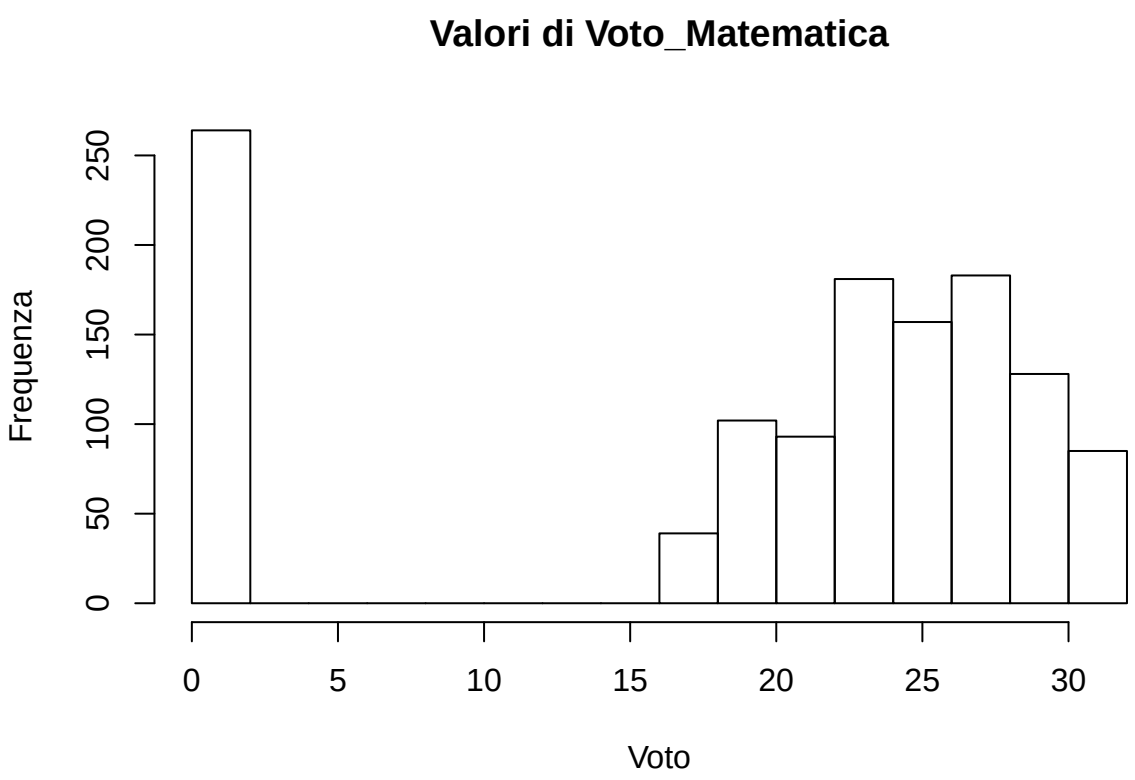
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   23.00   25.00   25.04   28.00   31.00     261
```

Come mostrato, questo attributo presenta valori pari a zero e valori nulli. I valori pari a zero sono interpretabili come informazione non presente nel dataset, mentre i valori nulli corrispondono agli studenti che non hanno superato l'esame di matematica. L'informazione mancante non può essere esclusa, considereremo quindi la media dei voti dello studente come valore attendibile per `Voto_Matematica`.

```
dataset$Voto_Matematica = with(dataset, ifelse(Voto_Matematica %in% (0),
                                               Voto_medio, Voto_Matematica))
dataset$Voto_Matematica = with(dataset, ifelse(Esame_Matematica %in% ('Non superato'),
                                               0, Voto_Matematica))
summary(dataset$Voto_Matematica)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   19.00   24.00   19.85   28.00   31.00
```

In questo modo abbiamo mantenuto le informazioni intatte all'interno del nostro dataset, in qualche modo inferendo quelle mancanti, e modificato il significato di un valore dell'attributo `Voto_Matematica`: adesso il lo zero corrisponde agli studenti che non hanno superato l'esame di matematica.



Uno degli attributi più importanti di questo dataset è `Voto_test` che indica il voto conseguito da uno studente per il test di ingresso al Corso di Laurea a cui si è iscritto. L'attribuzione del voto è stata modificata negli anni, in particolare: negli anni 2010-2015, il test era costituito da un questionario con 25 domande e ogni risposta corretta era valutata 1, ogni risposta sbagliata o non data era valutata 0; il test risultava superato con un punteggio >=12. Dal 2016, invece, il test e' costituito da un questionario con 20 domande: ogni risposta corretta viene valutata 1, ogni risposta sbagliata viene valutata -0.25 e ogni risposta non data 0; il test risulta superato con punteggio >=8. E' quindi presente, a seconda dell'anno preso in considerazione, un diverso range di valori con attributi diversi che dovrebbero in realtà avere lo stesso significato, come per esempio 8 e 12 per il superamento del test. Possiamo dunque applicare una tecnica di standardizzazione che riconduce un qualunque attributo v con media μ e varianza σ^2 ad una variabile v' con media $\mu = 0$ e varianza $\sigma^2 = 1$, ossia con distribuzione standard. Definendo $\mu_0, \mu_1 \ldots, \mu_6$ come la media dei valori dell'attributo e $\sigma_0, \sigma_1 \ldots, \sigma_6$ la sua deviazione standard rispettivamente per gli anni $2010, 2011, \ldots, 2016$, il nuovo valore è cacolato come:

$v' = \frac{v - \mu_i}{\sigma_i}$

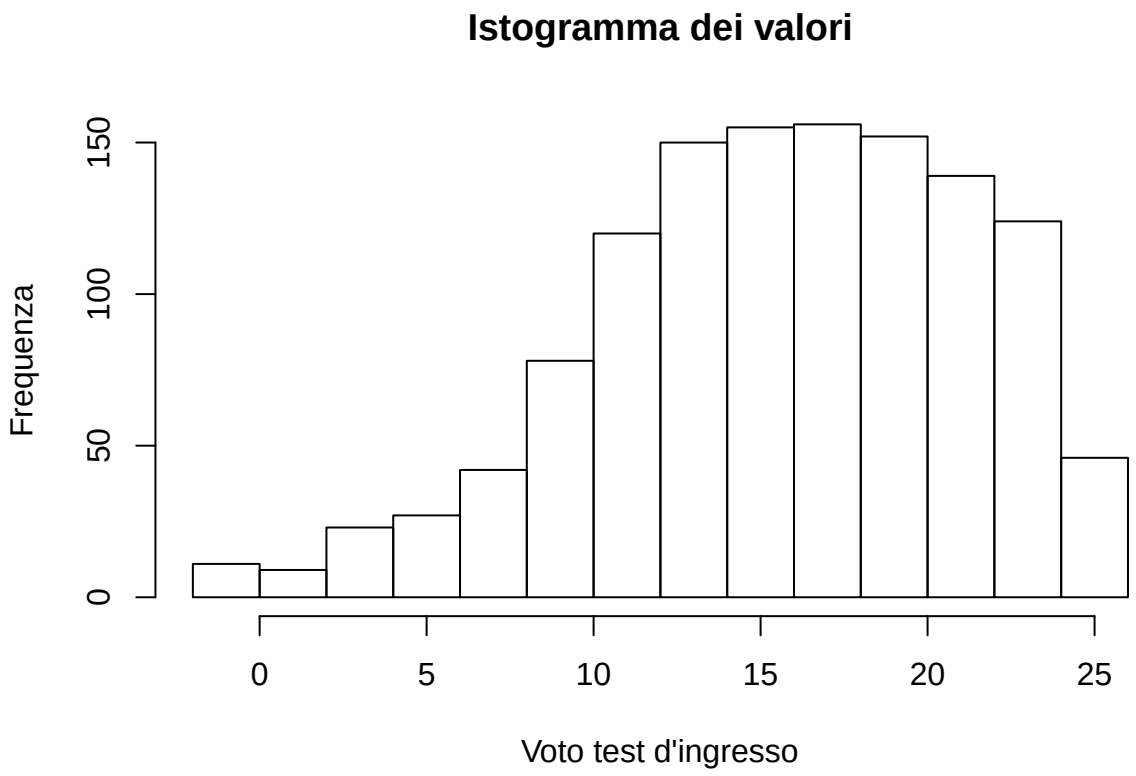
attraverso la funzione `scale`.

```
summary(dataset$Voto_test)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -1.00   12.00   16.62   16.09   21.00   25.00
```

```
sd(dataset$Voto_test)
```

```
## [1] 5.626392
```



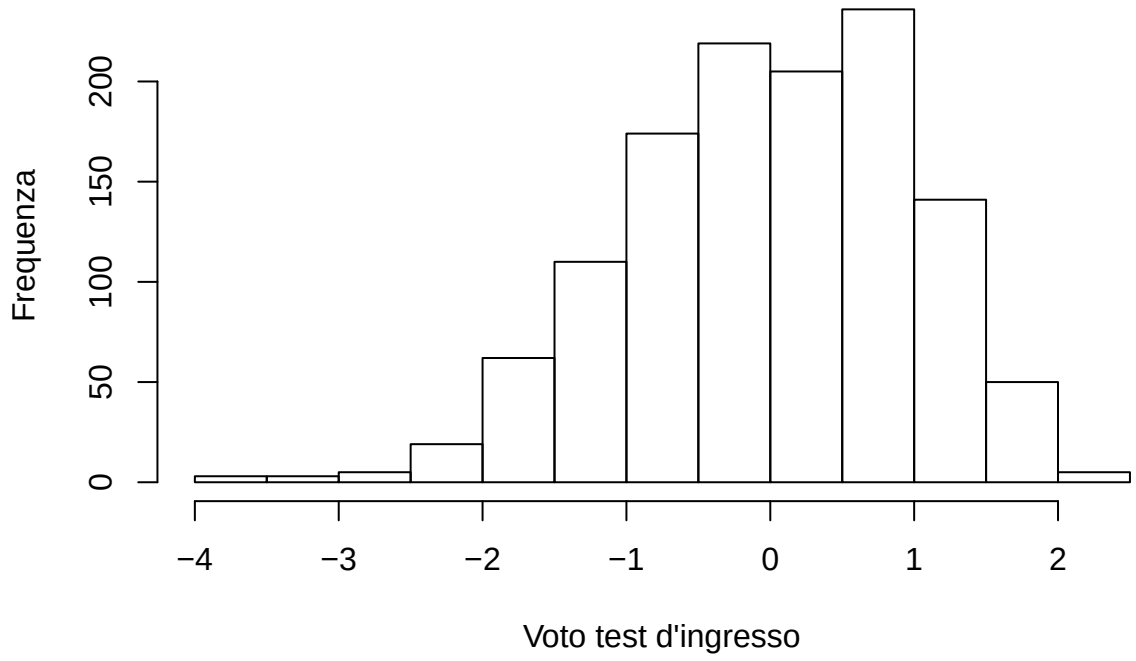

```
rescale_to_01 <- function(dataset, anno) {
  subset_data = subset(dataset, dataset$Coorte == anno)
  subset_data$Voto_test = scale(subset_data$Voto_test)
  return(subset_data)
}
```

```
summary(dataset$Voto_test)
```

```
##        V1
##  Min.   :-3.99885
##  1st Qu.:-0.74231
##  Median : 0.06651
##  Mean   : 0.00000
##  3rd Qu.: 0.78585
##  Max.   : 2.37405
```

```
sd(subset(dataset, dataset$Coorte == 2010)$Voto_test)
```

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



Abbiamo in tal modo terminato la parte di preprocessing del nostro dataset. Il risultato ottenuto è quindi il seguente:

```
dataset$Coorte = as.factor(dataset$Coorte)
summary(dataset)
```

```
##    CdS       Coorte    Genere     Voto_test.V1     Crediti_convoto
##  CdS1:435   2010:163   F:380   Min.   :-3.998851   Min.   : 0.00
##  CdS2:458   2011:161   M:852   1st Qu.:-0.742312   1st Qu.:15.00
##  CdS3:339   2012:161           Median : 0.066509   Median :33.00
##             2013:197           Mean   : 0.000000   Mean   :30.75
##             2014:193           3rd Qu.: 0.785851   3rd Qu.:48.00
##             2015:169           Max.   : 2.374047   Max.   :69.00
##             2016:188
##  Crediti_totali    Voto_medio   Scuola_provenienza        Esame_Matematica
##  Min.   :  3.0   Min.   : 0.0   Altro:118          EsameMatematica:971
##  1st Qu.: 18.0   1st Qu.:22.0   IP   : 12          Non superato   :261
##  Median : 36.0   Median :25.0   IT   :246
##  Mean   : 34.1   Mean   :22.1   LC   : 91
##  3rd Qu.: 51.0   3rd Qu.:27.0   LS   :698
##  Max.   :123.0   Max.   :31.0   TC   : 67
##
##  Voto_Matematica Crediti_Matematica
##  Min.   : 0.00   Min.   : 1.0
##  1st Qu.:19.00   1st Qu.:12.0
##  Median :24.00   Median :12.0
##  Mean   :19.85   Mean   :12.8
```

```
##  3rd Qu.:28.00   3rd Qu.:15.0
##  Max.   :31.00   Max.   :15.0
##                  NA's   :261
```

Studio del dataset

Prima di passare alla fase di applicazione delle tecniche di clustering, possiamo cercare nel dataset delle relazioni fra i dati. La prima relazione interessante da approfondire è sicuramente quella fra il voto medio di ogni studente e sia il suo voto medio durante l'anno, sia il suo voto al test d'ingresso.

Possiamo calcolare la correlazione di Pearson per le tre variabili. Questo valore r è tale che $r \in [-1, 1]$, dove:

- $r = -1$ indica una perfetta correlazione negativa
- $r = 0$ indica che non c'è correlazione fra le variabili, sono indipendenti
- $r = 1$ indica una perfetta correlazione negativa

La bontà di questo valore può essere a sua valta giudicata dal valore del p-value.

Valore della relazione fra il voto medio ed il voto al test d'ingresso:

```
## r:  0.3767487
```

```
## p-value:  7.8023e-43
```

Valore della relazione fra il voto all'esame di matematica ed il voto al test d'ingresso:

```
## r:  0.4302622
```

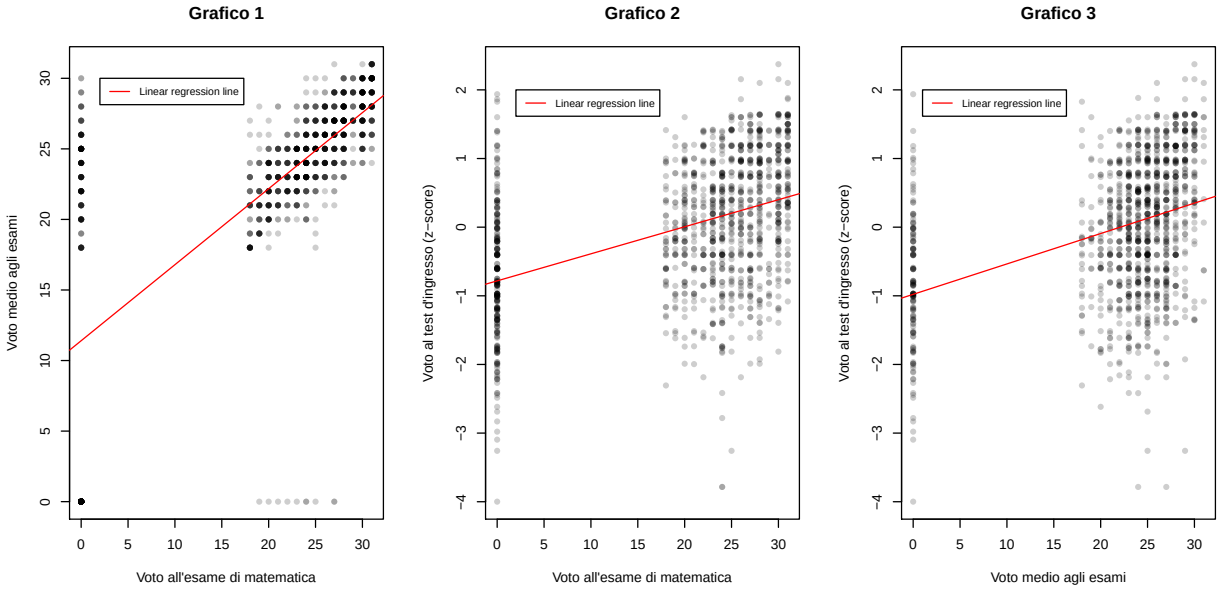
```
## p-value:  1.102598e-56
```

Valore della relazione fra il voto medio ed il voto all'esame di matematica:

```
## r:  0.6915192
```

```
## p-value:  6.057936e-176
```

Dunque, tra l'ultima coppia di variabili c'è la correlazione più forte, di tipo positivo. In conclusione, tutte e tre le variabili sono correlate positivamente ed i valori di r trovati sono attendibili in quanto abbiamo un p-value molto molto basso.



Nel primo grafico riusciamo a distinguere tre cluster:

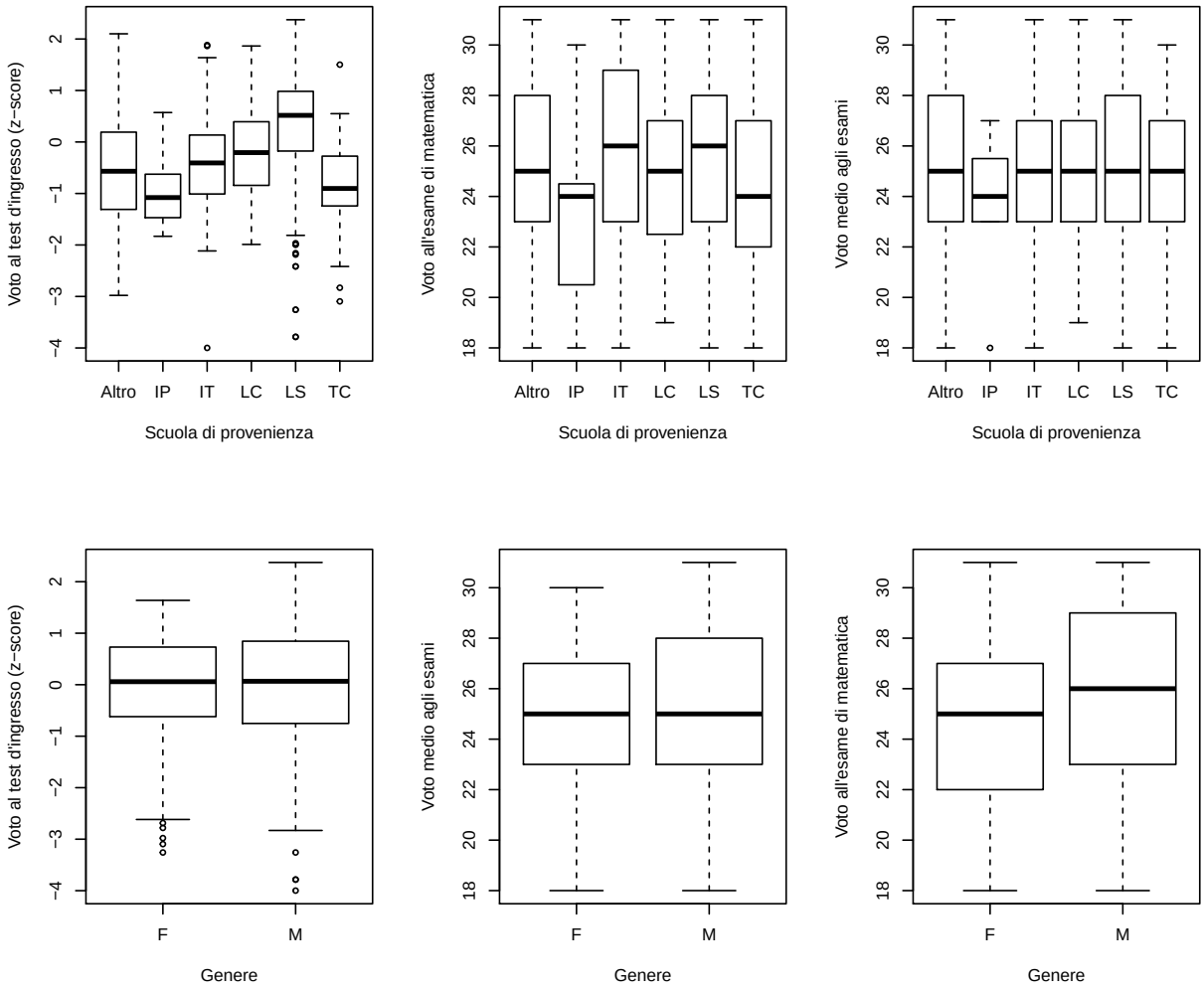
- Studenti che non hanno superato nessun esame, come mostrato dal mark in $(0, 0)$
- Studenti che hanno superato sia l'esame di matematica che altri esami
- Studenti che hanno superato solo altri esami

Empiricamente, non sono visibili outlier. Nei grafici 2 e 3, invece, possiamo fare distinzione fra due macrogruppi interessanti di studenti:

- Quelli che non hanno superato l'esame di matematica
- Quelli che non hanno superato nessun esame

La distribuzione del voto di matematica e del voto medio di tutti gli esami è molto più ampia in relazione al voto del test d'ingresso, e sono anche visibili degli outlier.

E' sicuramente interessante cercare una relazione tra la scuola di provenienza ed il genere degli studenti ed il loro voto medio agli esami, a quello di matematica e a quello del testo d'ingresso. Filtrando gli studenti con voti ≤ 18 per quanto riguarda gli esami, il dataset mostra chiaramente che chi proviene da un Liceo Scientifico e da un Liceo Tecnico riesce ad affrontare con più facilità il primo anno dei tre Corsi di Laurea. Per quanto riguarda il test d'ingresso, invece, l'aver frequentato un Liceo Scientifico in qualche modo garantisce un certo vantaggio.



Confrontando il genere degli studenti con i loro risultati accademici si evince che le femmine hanno valutazioni superiori ai maschi. Se consideriamo però il loro genere con il numero di studenti che non hanno superato l'esame di matematica o alcun esame possiamo fare delle considerazioni importanti. Considerando che il numero di iscritti diviso per genere è:

```
length(with(subset(dataset, Genere=='F'), Genere))
```

```
## [1] 380
```

```
length(with(subset(dataset, Genere=='M'), Genere))
```

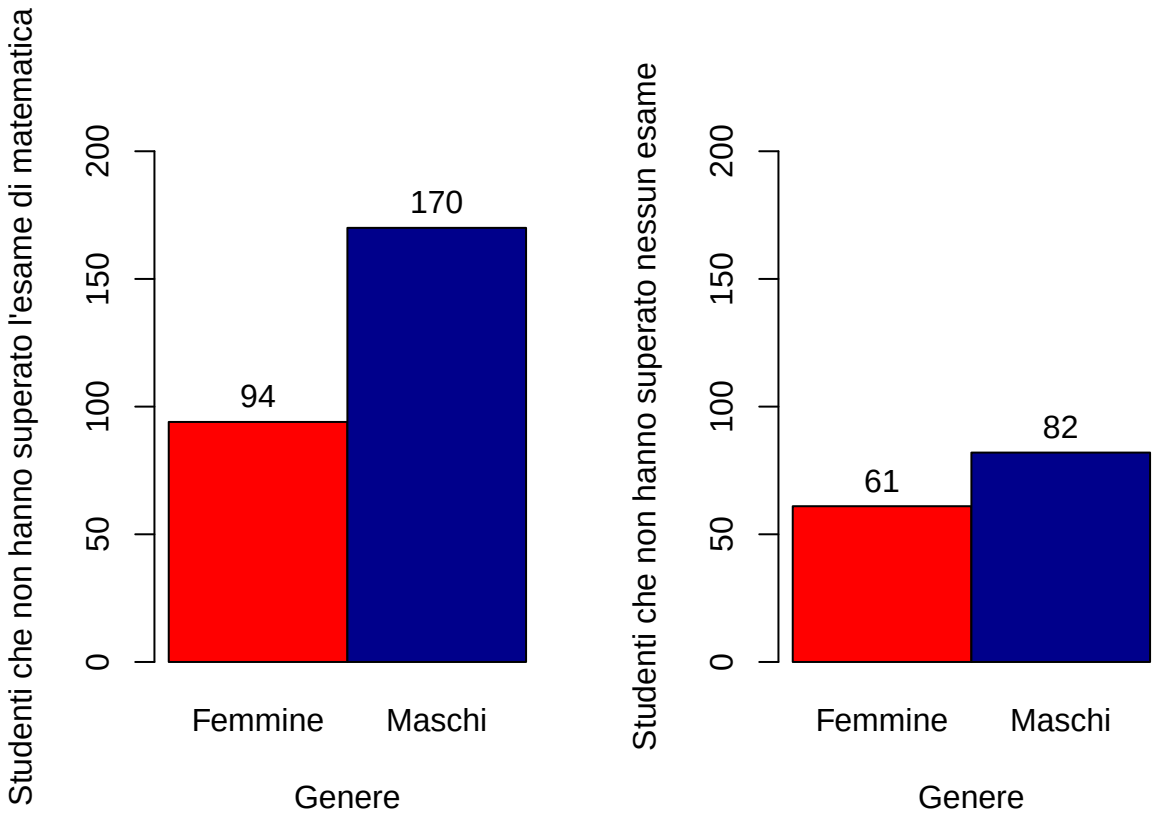
```
## [1] 852
```

allora i valori ottenuti per l'esame di matematica equivalgono al:

- $\frac{94}{380} \times 100 \approx 24.74\%$ del totale delle femmine
- $\frac{170}{852} \times 100 \approx 19.95\%$ del totale dei maschi

mentre quelli relativi a tutti gli esami:

- $\frac{61}{380} \times 100 \approx 16.05\%$ del totale delle femmine
- $\frac{82}{852} \times 100 \approx 9.62\%$ del totale dei maschi



In conclusione, possiamo dire che le ragazze ottengono risultati migliori durante il loro primo anno accademico, ma una grande percentuale di loro non riesce ad affrontare gli esami.

Clustering

L'algoritmo di clustering che è stato usato per questo dataset è k-means, un algoritmo partizionale basato su prototipi. In questo tipo di algoritmi di clustering non sovrapposti, un cluster è un'insieme di oggetti i quali

sono più vicini al prototipo che definisce il cluster che a quelli che definiscono gli altri cluster. Il concetto di vicinanza può variare, come vedremo più avanti.

Nel nostro caso il prototipo sarà un centroide, cioè la media di tutti i punti del custer: questo è possibile poiché vengono considerato solo attributi continui, se avessimo considerato anche attributi categorici allora avremmo dovuto scegliere un medoide, cioè il punto più rappresentativo del cluster.

L'algoritmo

- Seleziona k punti come centroidi iniziali
- **ripeti**
  - Forma k cluster assegnando ogni punto al centroide più vicino
  - Ricalcola il centroide di ogni cluster
- **finché** I centroidi non cambiano (o vengono modificati in valori entro un certo range definito)

La prima operazione da fare è quindi quella di scegliere k, cioè il numero di cluster che vogliamo in output dal nostro algoritmo. Tipicamente l'obbiettivo di un clustering è espresso da una funzione obbiettivo, nel nostro caso la somma dell'errore quadratico medio, o **SSE**. Per calcolarla possiamo considerare l'errore (la distanza) di ogni punto rispetto al suo centroide.

$$\text{SSE} = \sum_{i=1}^{k} \sum_{x_i \in C_k} dist(x_i, c_i)^2 = \sum_{i=1}^{k} \sum_{x_i \in C_k} (x_i - c_i)^2$$

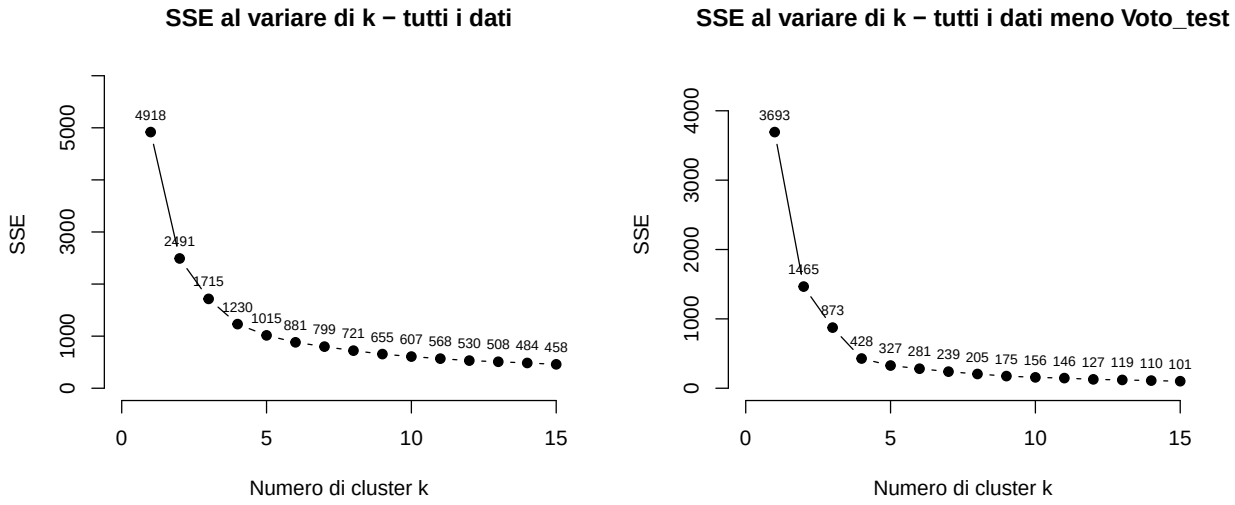
Fatte queste premesse, possiamo considerare come migliore un clustering che minimizza l'SSE, poiché mostra dei centroidi che meglio rappresentano i punti del dataset.

La modalità con cui è stato scelto k è stata quella di graficare l'SSE risultante da diverse esecuzioni dell'algoritmo, per $k = 1, 2, \ldots, 15$.

```
dataset <- subset(dataset, select = -c(CdS,
                                       Coorte,
                                       Genere,
                                       Crediti_totali,
                                       Scuola_provenienza,
                                       Esame_Matematica,
                                       Crediti_Matematica))
dataset <- na.omit(dataset)
dataset[c(2: 4)] <- lapply(dataset[c(2: 4)], function(x) c(scale(x)))
head(dataset)
```

```
##    Voto_test Crediti_convoto Voto_medio Voto_Matematica
## 1  0.9699223      -0.2014972  0.2238823      0.01348544
## 2 -0.1593444      -1.1693122  0.2238823      0.01348544
## 3 -1.0627577      -0.5241022 -0.2474086      0.47243755
## 4  0.2923623       0.4437128  0.4595278      0.93138966
## 5 -0.1593444       0.7663178  0.1060596      0.38064713
## 6  0.9699223       1.4115277  0.6951733      0.65601840
```

```
k.max <- 15
plot_kmeans <- function(dataset, title){
  wss <- sapply(1:k.max,
                function(k){kmeans(dataset, k, nstart=50,iter.max = 15 )$tot.withinss})
  xx = plot(1:k.max, wss,
       type="b", pch = 19, frame = FALSE,
       xlab="Numero di cluster k",
       ylab="SSE", xlim = c(0,k.max), ylim = c(0,max(wss)*1.2), main = title)
  text(wss~c(1:15), labels = as.integer(wss), pos = 3, cex=0.7)
}
```



Dal grafico si evince chiaramente che per $k \geq 4$ abbiamo un SSE che decresce molto più lentamente che per $k < 4$. Possiamo quindi dedurre che $k = 4$ è con tutta probabilità il valore migliore che possiamo dare in input all'algoritmo k-means per questo dataset.