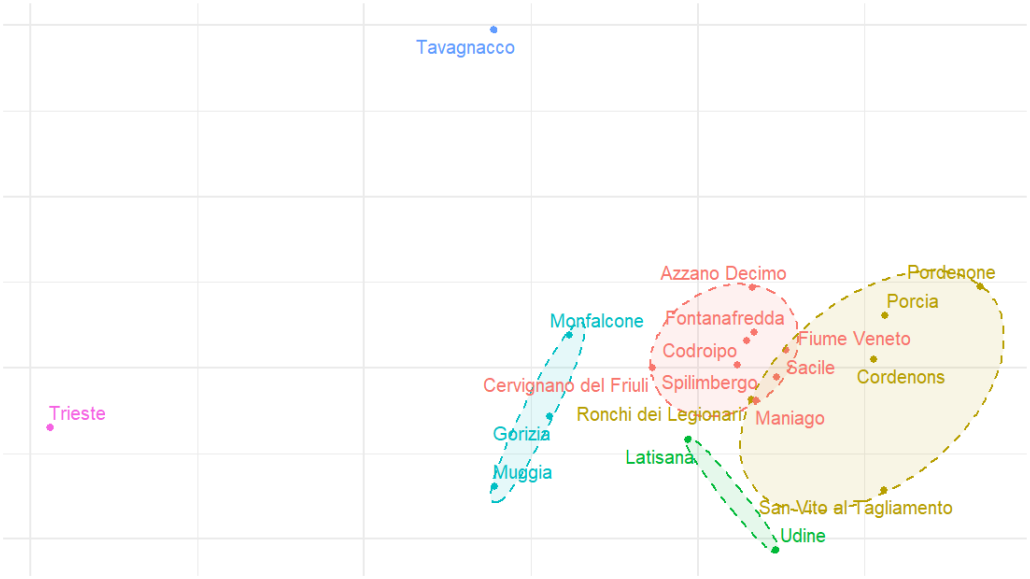
UNITS

Analisi sulla Produzione dei Rifiuti nel Friuli-Venezia Giulia

Progetto dell'Esame di Statistica Computazionale e Modelli Multivariati



Erik De Luca

A.A. 2024/2025

Indice

Consegna

Si consideri l'insieme di dati contenuti nel file RU_FVG_22_semestre-2 utilizzando il linguaggio SAS dopo aver importato il file scrivendo gli eventuali problemi e indicando come sono stati risolti si scrivano i comandi necessari e si commentino i risultati per:

a. Calcolare le statistiche univariate e le opportune rappresentazioni grafiche per le variabili e commentare i risultati.

b. Calcolare le frequenze assolute e le tabelle a doppia entrata (si utilizzi l'opzione \chisq) ricodificando preventivamente le variabili ove necessario e commentare i risultati.

c. Calcolare la matrice di correlazione commentando i risultati.

d. Nell'ipotesi in cui si ritenga necessario standardizzare le variabili (giustificare).

e. Effettuare un'analisi dei cluster gerarchica (eventualmente su un campione di dati) utilizzando i diversi metodi (legame singolo, completo, etc…), individuare il metodo migliore spiegare perché lo si è scelto. Commentare i risultati indicando quali comuni appartengono ai cluster scelti.

f. Dare come input della successiva cluster non gerarchica il numero di cluster ottenuti con la cluster gerarchica. Commentar l'output e i risultati. Confrontare i risultati delle due analisi dei clusters.

Capitolo 1

Preparazione dati in R

1.1 Importazione librerie e dati

I dati, contenuti in un foglio Excel, sono stati dapprima manipolati in Excel e successivamente in R. Grazie al pacchetto `{tidyverse}` si è gestito tutto il processo riguardante il preprocessamento dati. Ovviamente, ciò sarebbe stato possibile anche attraverso il software SAS.

Dal foglio iniziale in Excel sono state estratte le informazioni essenziali per l'analisi e divise in 3 diversi fogli per una gestione separata dei dati. Il primo foglio contiene i dati del comune, il secondo contiene l'analisi dettagliata dei rifiuti e la terza raccoglie degli indicatori basati sui dati del primo e secondo foglio. L'output seguente è un esempio del contenuto del foglio Excel sugli indicatori.

```
library(tidyverse)
library(readxl)
library(visdat)

# Importazione dati da fogli excel che iniziano con R_
info <- read_excel(
  here::here("data/RU_FVG_22_semestre-2.xlsx"),
  sheet = "R_info"
  )

rifiuti <- read_excel(
  here::here("data/RU_FVG_22_semestre-2.xlsx"),
  sheet = "R_rifiuti"
  )

indicatori <- read_excel(
  here::here("data/RU_FVG_22_semestre-2.xlsx"),
  sheet = "R_indicatori"
  )

print_df <- function(data, nrow = 5, ncol = 5, accuracy = .01)
{
  decimals <- -log10(accuracy)
  data |>
    slice_head(n = nrow) |>
    select(1:ncol) |>
    mutate(across(where(is.numeric), \(x) round(x, decimals)))
```

```
}

print_df(indicatori)
```

Istat	Totale RU (t)	Indifferenziati (t)	Differenziati (t)	RD (%)
031001	370.23	106.12	264.10	0.71
031002	1715.81	533.74	1182.07	0.69
031003	290.08	62.50	227.59	0.78
031004	89.51	24.68	64.84	0.72
031005	303.64	81.71	221.93	0.73

1.2 Rinominazione colonne

Osservando i nomi delle colonne ci sono casi di omonimia.

```
rifiuti |>
  names() |>
  tibble(rifiuti = _) |>
  filter(if_any(rifiuti, \(x) str_detect(x, "\\.\\.\\.\\.."))) |>
  slice_sample(n = 5) |>
  bind_cols(indicatori |>
              names() |>
              tibble(indicatori = _) |>
              filter(if_any(indicatori, \(x) str_detect(x, "\\("))) |>
              slice_sample(n = 5))
```

rifiuti	indicatori
Raee...8	Indifferenziati (t)
Altri tipi di imballaggi...52	Plastica (Kg)
Cartucce e toner per stampa...57	Totale RU (t)
Metalli...22	Vetro (Kg)
Altri oli grassi ed emulsioni...45	RAEE (Kg)

Unisco le colonne con lo stesso nome e faccio una somma dei valori. Inoltre per le colonne di indicatori elimino il contenuto tra parentesi per una lettura più semplice.

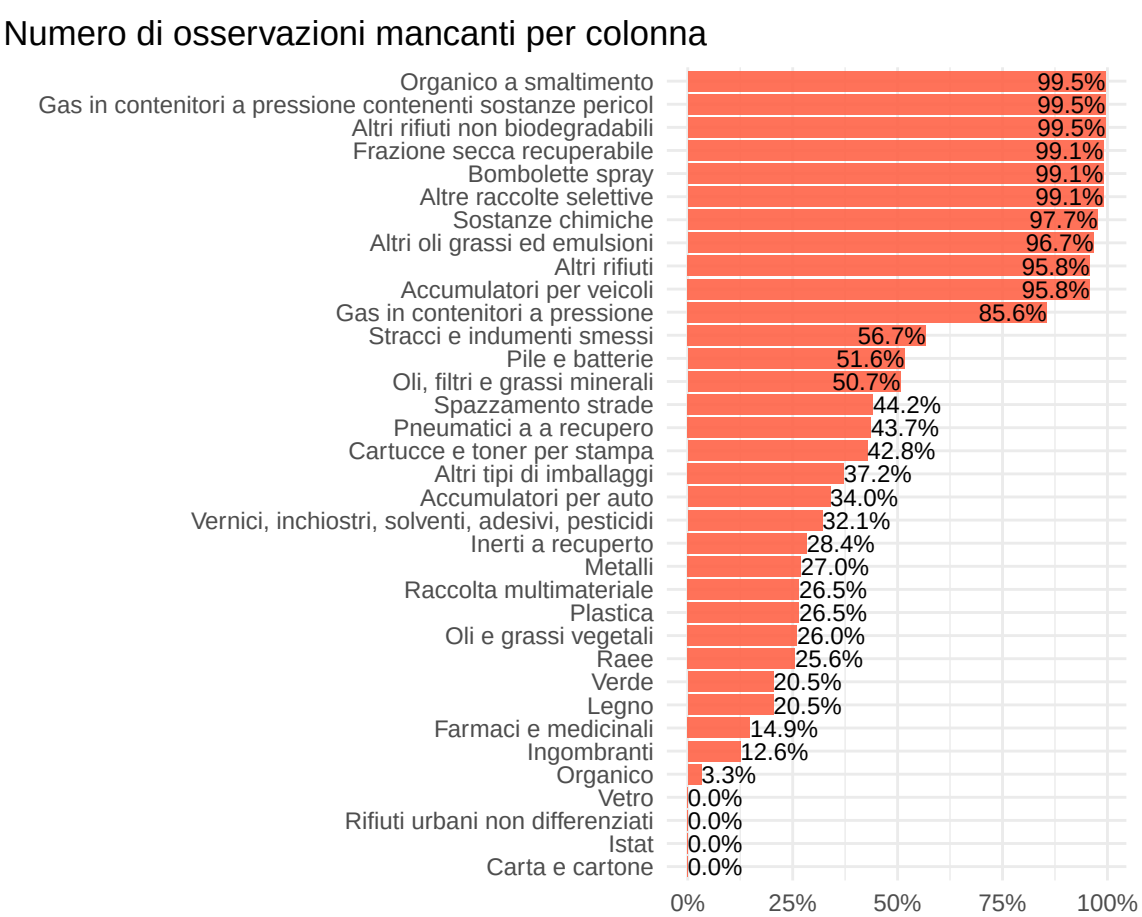
```
# elimino le parentesi e il loro contenuto
indicatori |>
  rename_with(~str_remove(.x, "[:space:]\\(.*\\)"), -1) -> indicatori

# raggruppo colonne con lo stesso nome effettuando la somma
rifiuti |>
  pivot_longer(cols = -Istat, names_to = "group", values_to = "value") |>
  mutate(group = str_remove(group, "\\.\\.\\..*")) |>
  pivot_wider(
    names_from = group,
    values_from = value,
    values_fn = \(x) sum(x, na.rm = T)
    ) |>
  # mutate 0 in NA
  mutate(across(where(is.numeric), ~na_if(.x, 0))) -> rifiuti
```

1.3 Dati mancanti

Come si può vedere ci sono alcune colonne con un alto tasso di dati mancanti e alcune righe con molti dati mancanti.

Potrei valutare di eliminare i comuni che raccolgono pochi dati ed eliminare i tipi di rifiuti che i coumni normalemente non raccolgono o non classificano.

```
rifiuti |>
  summarise(across(everything(), ~sum(is.na(.x))/n())) |>
  pivot_longer(cols = everything(), names_to = "col", values_to = "miss") |>
  mutate(col = fct_reorder(col, miss, .desc = F,
                           .na_rm = F, .fun = ~ sum(., na.rm = T))) |>
  ggplot(aes(miss, col, label = scales::percent(miss, accuracy = .1))) +
  geom_col(fill = "tomato", alpha = .9) +
  geom_text(hjust = "inward", size = 3) +
  scale_x_continuous(labels = scales::percent) +
  theme_minimal() +
  labs(
    x = "",
    y = "",
    title = "Numero di osservazioni mancanti per colonna"
  ) +
  theme(plot.title.position = "plot")
```



Elimino le colonne con oltre l'80% di dati mancanti. Infatti queste colonne sarebbero difficili da trattare. In un'analisi più approfondita potrebbero essere considerate come: cura di dettaglio della raccolta dati che a sua volta può essere indice di una migliore raccolta differenziata da parte del comune.

```
rifiuti |>
  summarise(across(everything(), ~sum(is.na(.x))/n())) |>
  pivot_longer(cols = everything(), names_to = "col", values_to = "miss") |>
  filter(miss > 0.8) |>
  pull(col) -> drop_cols

rifiuti |>
  select(-all_of(drop_cols)) -> rifiuti

print_df(tibble("Colonne Eliminate" = drop_cols), ncol = 1)
```

Colonne Eliminate
Organico a smaltimento
Altri rifiuti non biodegradabili
Bombolette spray
Frazione secca recuperabile
Accumulatori per veicoli

I comuni con oltre il 50% di dati mancanti sono 42. Le opzioni sono due:

1. Eliminare i comuni con molti dati mancanti
2. Imputare i dati tramite media o mediana, o tramite un modello di regressione che stima i dati mancanti considerando la provincia e la popolazione

```
rifiuti |>
  rowwise() |>
  transmute(
    Istat,
    miss = round(sum(is.na(c_across(where(is.numeric)))) / ncol(rifiuti), 2)
    ) |>
  ungroup() |>
  filter(miss > 0.5) |>
  arrange(desc(miss)) |>
  left_join(info, by = "Istat") |>
  print_df(nrow = 10)
```

Istat	miss	Provincia	Comune	Popolazione
030003	0.75	Udine	Ampezzo	920
030007	0.75	Udine	Attimis	1683
030021	0.75	Udine	Cavazzo Carnico	937
030022	0.75	Udine	Cercivento	652
030029	0.75	Udine	Comeglians	442
030036	0.75	Udine	Faedis	2775
030047	0.75	Udine	Lauco	663
030081	0.75	Udine	Prato Carnico	853
030084	0.75	Udine	Preone	252
030088	0.75	Udine	Ravascletto	494

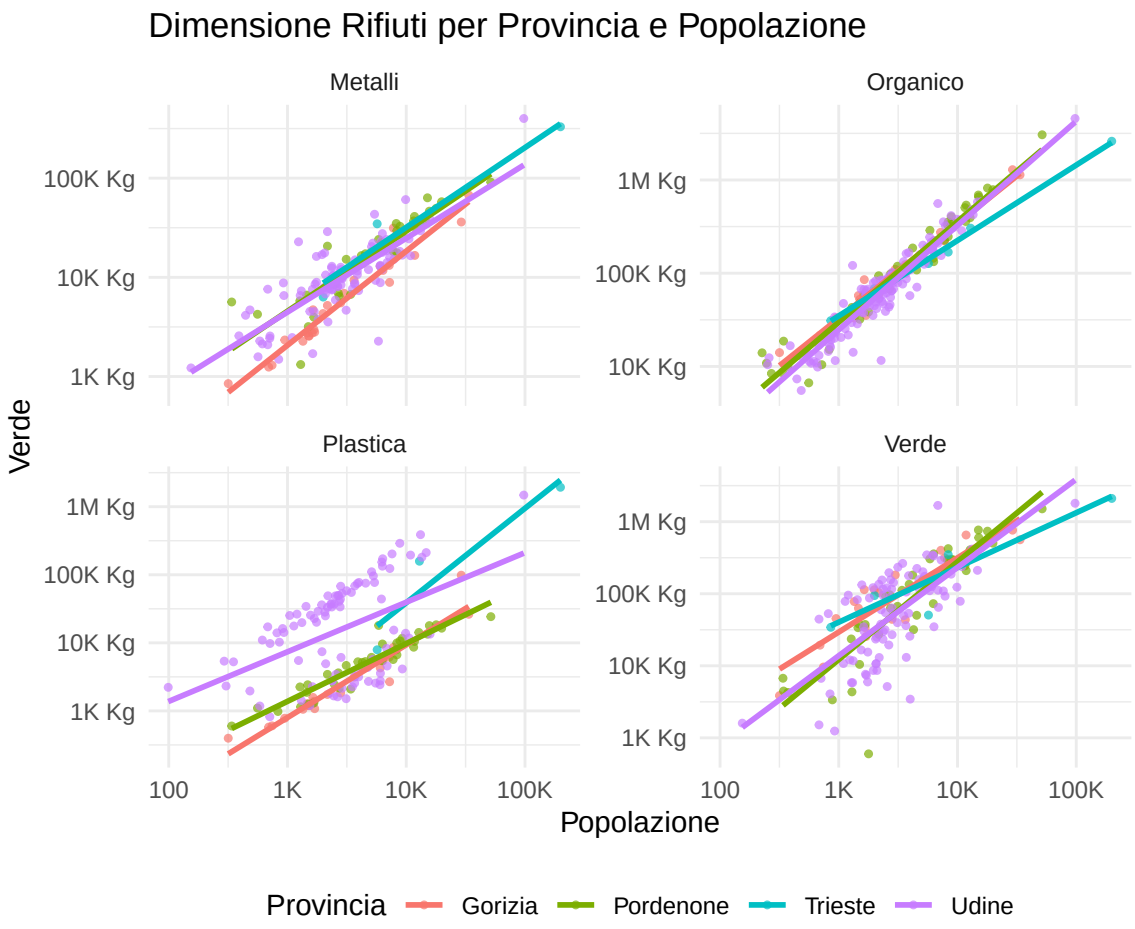
1.3.1 Imputazione dati mancanti

Nel grafico viene rappresentato un esempio di imputazione dei dati mancanti tramite regressione lineare per i rifiuti:

- Verde
- Metalli
- Plastica
- Organico

Un modello più complesso potrebbe essere considerare le variabili disponibili sugli altri rifiuti e sugli indicatori ma per semplicità considero solo la popolazione e la provincia.

```
rifiuti |>
  left_join(info, by = "Istat") |>
  pivot_longer(
    cols = -c(Istat, Provincia, Popolazione, Comune),
    names_to = "rifiuto",
    values_to = "peso"
    ) |>
  filter(rifiuto %in% c("Verde", "Plastica", "Metalli", "Organico")) |>
  # filter(is.na(peso))
  ggplot(aes(x = Popolazione, y = peso, color = Provincia)) +
  geom_point(alpha = .7, size = .9) +
  facet_wrap(~rifiuto, ncol = 2, scales = "free_y") +
  geom_smooth(method = "lm", se = F) +
  scale_x_log10(
    labels = scales::label_number_si(accuracy = 1)
  ) +
  scale_y_log10(
    labels = scales::label_number_si(accuracy = 1, suffix = " Kg")
  ) +
  labs(title = "Dimensione Rifiuti per Provincia e Popolazione",
       x = "Popolazione",
       y = "Verde") +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Per semplificare l'imputazione creo una funzione che crea un modello per ciascuna variabile contenente NAs e procede a riempire il dataset con i nuovi dati generati

```
# Imputazione attraverso regressione lineare
imputa_dati <- function(data, var_mancante, var_predictors) {
  # Rimuoviamo i casi NA per il fitting del modello
  data_non_na <- data |> filter(!is.na(!!sym(var_mancante)))

  # Adattiamo il modello di regressione
  formula <- as.formula(paste0("`", var_mancante, "` ~ ",
                               paste(var_predictors, collapse = " + ")))
  modello <- lm(formula, data = data_non_na)

  # Prevediamo i valori mancanti
  data |>
    mutate(
      !!sym(var_mancante) := ifelse(is.na(!!sym(var_mancante)),
                                    predict(modello, newdata = data) |> round(0),
                                    !!sym(var_mancante))
    ) -> data
  return(data)
}
```

```
reduce(
  colnames(rifiuti)[colSums(is.na(rifiuti)) > 0],
  ~imputa_dati(.x, .y, c("Provincia", "Popolazione")),
  .init = rifiuti |> left_join(info, by = "Istat")
) |>
  select(-c(Popolazione, Provincia, Comune)) -> rifiuti
```

1.4 Trasformazione Variabili

Alcuni indici sono per persona mentre altri sono globali. Per una migliore analisi li trasformo tutti per persona (pro capite).

```
indicatori |>
  left_join(
    info |> select(Istat, Popolazione),
    by = "Istat"
  ) |>
  mutate(
    across(c(Differenziati, Indifferenziati), \(x) x / Popolazione * 1E3)
  ) |>
  select(-Popolazione, -`Totale RU`) -> indicatori

indicatori |>
  print_df(ncol = 7)
```

Istat	Indifferenziati	Differenziati	RD	Rsecco	RU	Carta
031001	65.63	163.33	0.71	59.07	228.96	27.68
031002	74.15	164.22	0.69	64.43	238.37	31.69
031003	46.29	168.58	0.78	41.52	214.88	22.57
031004	77.84	204.53	0.72	70.22	282.37	31.03
031005	48.90	132.81	0.73	39.95	181.71	22.31

1.5 Aggiunta Variabili Categoriche

Per la consegna di un esercizio bisogna creare una tabella a doppia entrata. Per far ciò devo usare due variabili qualitative, le otterrò categorizzando le variabili numeriche degli indicatori. Creo, quindi, un nuovo dataset chiamato *indicatori_categorici* che conterrà tutti gli indicatori in forma di variabili qualitative con 3 categorie:

1. Basso
2. Medio
3. Alto

L'assegnazione della categoria è basata sui quantili della variabile.

```
# dividi in 3 categorie
indicatori |>
  mutate(
    across(
      where(is.numeric),
            ~ cut(
```

```
            .x,
            breaks = quantile(.x,
                              probs = seq(0, 1, by = 1/3), # c(0.33, .66, 1)
                              na.rm = TRUE),
            labels = c("Basso", "Medio", "Alto"),
            include.lowest = TRUE)
    )
  ) -> indicatori_categorici
```

1.6 Esportazione dati

Ora che i dati sono puliti e pronti li *impacchetto* con dei join e li salvo in formato *csv*, un formato leggero e facilmente importabile.

Nel creare un unico file con tutti i dati uniti, devo rinominare gli indicatori per evitare casi di omonimia (e.g. Vetro).

```
# solo rifiuti
info |>
  right_join(rifiuti, by = "Istat") |>
  write_csv(here::here("data/rifiuti.csv"))

# solo indicatori
info |>
  right_join(indicatori, by = "Istat") |>
  write_csv(here::here("data/indicatori.csv"))

# indicatori categorici
info |>
  right_join(indicatori_categorici, by = "Istat") |>
  write_csv(here::here("data/indicatori_categorici.csv"))

# rifiuti e indicatori
info |>
  right_join(rifiuti, by = "Istat") |>
  # rename indicatori con I_ per evitare conflitti
  right_join(indicatori |> rename_with(~str_c("I_", .x), -1), by = "Istat") |>
  write_csv(here::here("data/rifiuti_indicatori.csv"))
```

Capitolo 2

Analisi dati in SAS

2.1 Importazione dei dati

Carico in SAS i diversi file *csv* e li salvo nelle rispettive nuove variabili.

- `datafile = "/home/u64115021/sasuser.v94/.*"`: Percorso della posizione del file.
- `out`: Nome della variabile salvata in SAS.
- `dbms=csv` (*Data Base Management System*): Serve a indicare il tipo di formato dei dati. In questo caso in formato csv.
- `replace`: Se la variabile è già presente, allora andrà sostituita.
- `getnames=yes`: Usa la prima riga come nomi delle variabili del dataset.

```
proc import datafile="/home/u64115021/sasuser.v94/rifiuti.csv"
    out=rifiuti
    dbms=csv
    replace;
    getnames=yes;
run;

proc import datafile="/home/u64115021/sasuser.v94/indicatori.csv"
    out=indicatori
    dbms=csv
    replace;
    getnames=yes;
run;

proc import datafile="/home/u64115021/sasuser.v94/rifiuti_indicatori.csv"
    out=rifiuti_indicatori
    dbms=csv
    replace;
    getnames=yes;
run;

proc import datafile="/home/u64115021/sasuser.v94/indicatori_categorici.csv"
    out=indicatori_categorici
    dbms=csv
    replace;
```

```
    getnames=yes;
run;
```

2.2 Statistiche univariate e rappresentazioni grafiche

Calcolo delle statistiche univariate e rappresentazione grafica. Per semplicità l'analisi è stata eseguita solo sulla variabili *RU* (rifiuto urbano) presente nel dataset *indicatori*.

- `proc univariate`: Calcola le statistiche univariate per un singolo indice del dataset.
- `data=indicatori`: Specifica il dataset su cui calcolare le statistiche.
- `var RU`: Indica la variabile su cui calcolare le statistiche, in questo caso *RU*.

```
/* Calcolare statistiche univariate solo su un indice per semplicità */
proc univariate data=indicatori;
    var RU;
run;
```

Momenti			
N	215	Somma dei pesi	215
Media	215.605891	Somma delle osservazioni	46355.2666
Deviazione std	77.4711776	Varianza	6001.78336
Skewness	5.22192642	Curtosi	50.0138743
SS non corretta	11278850.2	SS corretta	1284381.64
Coeff var	35.9318464	Errore std media	5.28349046

(a) Momenti

Misure statistiche di base			
Posizione		Variabilità	
Media	215.6059	Deviazione std	77.47118
Mediana	205.8548	Varianza	6002
Moda	.	Range	914.03503
		Range interquartile	67.60181

Test di posizione: Mu0=0				
Test	Statistica		P-value	
T di Student	t	40.80747	Pr > |t|	<.0001
Segno	M	107.5	Pr >= |M|	<.0001
Rango con segno	S	11610	Pr >= |S|	<.0001

(b) Statistiche di base

Quantili (Definizione 5)	
Livello	Quantile
100% Max	1005.0626
99%	422.3496
95%	305.6099
90%	282.3691
75% Q3	243.5948
50% Mediana	205.8548
25% Q1	175.9929
10%	147.9234
5%	132.1351
1%	108.0797
0% Min	91.0276

(c) Quantili

Figura 2.1: Variabile RU (rifiuto urbano)

Produco un istogramma per vedere come si distribuisce la variabile RU.

- `proc sgplot`: Crea rappresentazioni grafiche del dataset.
- `histogram RU`: Genera un istogramma per la variabile *RU*.

```
/* Rappresentazioni grafiche */
proc sgplot data=indicatori;
```

```
    histogram 'Totale RU'n;
run;
```

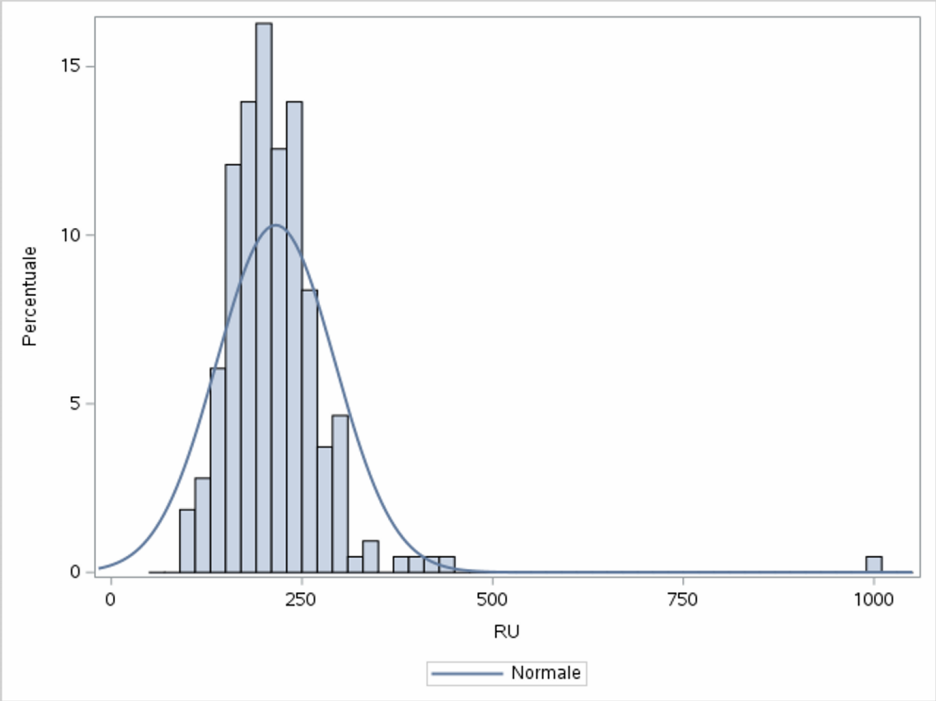


Figura 2.2: Istogramma dei rifiuti urbani per capita nei comuni della regione FVG

```
proc sgplot data=indicatori;
    scatter y=Differenziati x=Carta;
    yaxis label="Differenziati";
    xaxis label="Carta";
run;
```

Osservando l'istogramma si nota che i dati potrebbero essere approssimati da una normale. C'è la presenza di un principale outlier: Lignano Sabbiadoro. Lignano, come gli altri comuni con un indice di rifiuti urbani per capita maggiore, sono località turistiche. La popolazione residente è bassa ma c'è un grande afflusso di gente e di produzione di rifiuti durante le stagioni turistiche.

```
indicatori |>
  left_join(info, by = "Istat") |>
  select(Comune, Popolazione, RU) |>
  arrange(-RU) |>
  print_df(ncol = 3, nrow = 5)
```

Comune	Popolazione	RU
Lignano Sabbiadoro	6833	1005.06
Grado	7789	433.35
Barcis	226	422.35
Sappada	1308	391.83
Villesse	1640	382.21

Nello scatterplot, ogni punto rappresenta un comune, illustrando la relazione tra i rifiuti differenziati e carta. La distribuzione dei punti suggerisce una correlazione positiva.

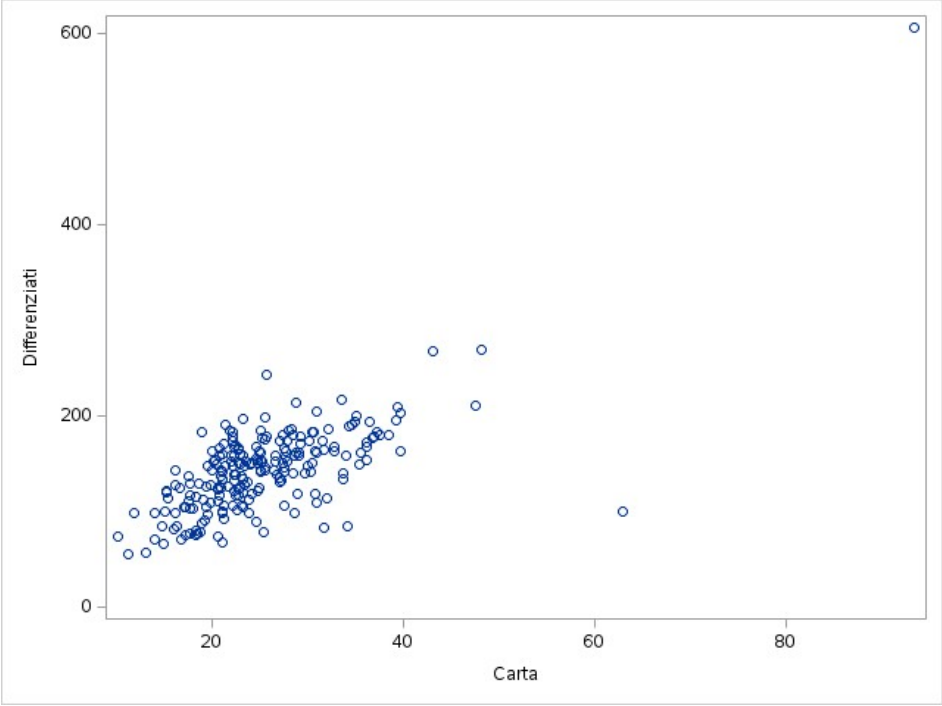


Figura 2.3: Scatterplot

2.3 Tabelle a singola e doppia entrata

- `proc freq`: Calcola le frequenze delle variabili nel dataset.
- `data=indicatori_categorici`: Specifica il dataset che contiene le variabili d'interesse.
- `tables Provincia / nocum`: Crea una tabella di frequenze assolute per la variabile 'Provincia' senza visualizzare i totali cumulativi in quanto non è una variabile categorica ordinata.

```
/* Frequenze assolute */
proc freq data=indicatori_categorici;
    tables Provincia / nocum;
run;
```

Provincia	Frequenza	Percentuale
Gorizia	25	11.63
Pordeno	50	23.26
Trieste	6	2.79
Udine	134	62.33

Figura 2.4: Numero di comuni per provincia

- `proc freq`: Utilizzato ancora per calcolare le frequenze, ma questa volta per due variabili.
- `tables RD*Provincia / chisq`: Crea una tabella a doppia entrata per le variabili *RD* (rifiuti differenziati) e *Provincia*, includendo il test chi-quadrato per valutare l'associazione tra le due variabili.

```
/* Tabelle a doppia entrata*/
proc freq data=indicatori_categorici;
    tables RD*Provincia / chisq;
run;
```

Frequenza
Percentuale
Pct riga
Pct col

Tabella di RD rispetto a Provincia

RD	Provincia				
	Gorizia	Pordeno	Trieste	Udine	Totale
Alto	15 6.98 20.83 60.00	25 11.63 34.72 50.00	0 0.00 0.00 0.00	32 14.88 44.44 23.88	72 33.49
Basso	4 1.86 5.56 16.00	9 4.19 12.50 18.00	4 1.86 5.56 66.67	55 25.58 76.39 41.04	72 33.49
Medio	6 2.79 8.45 24.00	16 7.44 22.54 32.00	2 0.93 2.82 33.33	47 21.86 66.20 35.07	71 33.02
Totale	25 11.63	50 23.26	6 2.79	134 62.33	215 100.00

Figura 2.5: Tabella di frequenze a doppia entrata

Il valore della statistica Chi-quadro suggerisce che c'è dipendenza tra la percentuale di rifiuti che vengono differenziati e la provincia d'appartenenza del comune in questione. Infatti i comuni delle provincie di Gorizia e Pordenone tendono a fare maggior raccolta differenziata, i comuni della provincia di Trieste sarebbero, invece, i meno virtuosi.

Statistiche per la tabella di RD rispetto a Provincia

Statistica	DF	Valore	Prob
Chi-quadrato	6	25.9457	0.0002
Chi-quadrato rapp verosim	6	27.5659	0.0001
Chi-quadrato MH	1	9.5464	0.0020
Coefficiente Phi		0.3474	
Coefficiente di contingenza		0.3282	
V di Cramer		0.2456	
WARNING: 25% delle celle ha conteggi previsti minori di 5. Il chi-quadrato potrebbe non essere un test valido.			

Figura 2.6: Statistiche d'indipendenza della variabile RD rispetto a Provincia

2.4 Matrice di Correlazione

Calcolo della matrice di correlazione per tutte le variabili numeriche nel dataset indicatori, escludendo 'istat' trasformato in carattere. Ecco come:

Trasformazione della variabile *istat*:

- `data indicatori`: Crea un nuovo dataset con lo stesso nome per elaborare i dati.
- `set indicatori`: Importa i dati originali.

- `istat_char = put(istat, 8.)`: Converte la variabile *istat* in formato carattere.
- `drop istat`: Elimina la variabile originale *istat*.
- `rename istat_char = istat`: Rinomina la variabile convertita per mantenere la coerenza nel dataset.

```
/* Matrice di correlazione */
/* Prima trasformo la variabile Istat in carattere in modo da non considerarla
nell'analisi*/
data indicatori;
    set indicatori;
    istat_char = put(istat, 8.); /* Conversione in carattere */
    drop istat; /* Rimuovo l'originale */
    rename istat_char = istat;
run;
```

Calcolo della matrice di correlazione:

- `proc corr`: Calcola la matrice di correlazione per le variabili numeriche.
- `var _numeric_`: Specifica che l'analisi deve essere effettuata su tutte le variabili numeriche del dataset.

```
proc corr data=indicatori;
    var _numeric_;
run;
```

Coefficienti di correlazione di Pearson, N = 215
Prob > |r| sotto H0: Rho=0

	Popolazione	Indifferenziati	Differenziati	RD	Rsecco	RU	Carta	Legno	Metallo	Organico	Plastica	RAEE	Verde	Vetro
Popolazione	1.00000	0.08248 0.2285	0.04386 0.5224	-0.06566 0.3380	0.09598 0.1608	0.07123 0.2985	0.08222 0.2299	0.07206 0.2929	-0.01264 0.8538	0.06513 0.3419	0.05547 0.4184	-0.01228 0.8580	0.02901 0.6723	-0.22502 0.0009
Indifferenziati	0.08248 0.2285	1.00000	0.47901 <.0001	-0.67463 <.0001	0.93630 <.0001	0.83005 <.0001	0.52241 <.0001	0.20596 0.0024	-0.03431 0.6168	0.06693 0.3287	0.28689 <.0001	0.08277 0.2268	0.41820 <.0001	0.57599 <.0001
Differenziati	0.04386 0.5224	0.47901 <.0001	1.00000	0.28336 <.0001	0.47115 <.0001	0.88715 <.0001	0.72870 <.0001	0.64649 <.0001	0.25246 0.0002	0.60896 <.0001	0.04361 0.5247	0.17381 0.0107	0.76296 <.0001	0.46542 <.0001
RD	-0.06566 0.3380	-0.67463 <.0001	0.28336 <.0001	1.00000	-0.64505 <.0001	-0.17463 0.0103	0.01669 0.8078	0.28712 <.0001	0.21241 0.0017	0.45684 <.0001	-0.32386 <.0001	0.03848 0.5747	0.15412 0.0238	-0.23324 0.0006
Rsecco	0.09598 0.1608	0.93630 <.0001	0.47115 <.0001	-0.64505 <.0001	1.00000	0.79156 <.0001	0.52343 <.0001	0.17997 0.0082	-0.10258 0.1338	0.00855 0.9008	0.33940 <.0001	0.03741 0.5854	0.48584 <.0001	0.53503 <.0001
RU	0.07123 0.2985	0.83005 <.0001	0.88715 <.0001	-0.17463 0.0103	0.79156 <.0001	1.00000	0.73760 <.0001	0.51901 <.0001	0.14236 0.0370	0.42208 <.0001	0.17853 0.0087	0.15394 0.0240	0.70459 <.0001	0.59850 <.0001
Carta	0.08222 0.2299	0.52241 <.0001	0.72870 <.0001	0.01669 0.8078	0.52343 <.0001	0.73760 <.0001	1.00000	0.40810 <.0001	0.08862 0.1955	0.40185 <.0001	0.12823 0.0605	0.00170 0.9803	0.43798 <.0001	0.48223 <.0001
Legno	0.07206 0.2929	0.20596 0.0024	0.64649 <.0001	0.28712 <.0001	0.17997 0.0082	0.51901 <.0001	0.40810 <.0001	1.00000	0.67515 <.0001	0.31138 <.0001	-0.05639 0.4107	0.02641 0.7002	0.30597 <.0001	0.19246 0.0046
Metallo	-0.01264 0.8538	-0.03431 0.6168	0.25246 0.0002	0.21241 0.0017	-0.10258 0.1338	0.14236 0.0370	0.08862 0.1955	0.67515 <.0001	1.00000	0.05800 0.3974	-0.14204 0.0374	0.04258 0.5347	-0.05779 0.3991	0.00705 0.9182
Organico	0.06513 0.3419	0.06693 0.3287	0.60896 <.0001	0.45684 <.0001	0.00855 0.9008	0.42208 <.0001	0.40185 <.0001	0.31138 <.0001	0.05800 0.3974	1.00000	-0.26973 <.0001	0.00498 0.9421	0.23205 0.0006	0.29861 <.0001
Plastica	0.05547 0.4184	0.28689 <.0001	0.04361 0.5247	-0.32386 <.0001	0.33940 <.0001	0.17853 0.0087	0.12823 0.0605	-0.05639 0.4107	-0.14204 0.0374	-0.26973 <.0001	1.00000	-0.08528 0.2130	0.20861 0.0021	0.05561 0.4172
RAEE	-0.01228 0.8580	0.08277 0.2268	0.17381 0.0107	0.03848 0.5747	0.03741 0.5854	0.15394 0.0240	0.00170 0.9803	0.02641 0.7002	0.04258 0.5347	0.00498 0.9421	-0.08528 0.2130	1.00000	-0.02764 0.6870	0.06831 0.3188
Verde	0.02901 0.6723	0.41820 <.0001	0.76296 <.0001	0.15412 0.0238	0.48584 <.0001	0.70459 <.0001	0.43798 <.0001	0.30597 <.0001	-0.05779 0.3991	0.23205 0.0006	0.20861 0.0021	-0.02764 0.6870	1.00000	0.16938 0.0129
Vetro	-0.22502 0.0009	0.57599 <.0001	0.46542 <.0001	-0.23324 0.0006	0.53503 <.0001	0.59850 <.0001	0.48223 <.0001	0.19246 0.0046	0.00705 0.9182	0.29861 <.0001	0.05561 0.4172	0.06831 0.3188	0.16938 0.0129	1.00000

Figura 2.7: Matrice di correlazione

2.5 Clustering Gerarchico

Effettuiamo un'analisi di clustering gerarchico sui 20 comuni più popolosi utilizzando diversi metodi.

Selezione dei comuni più popolosi:

- `proc sql outobs=20;`: Utilizza SQL per selezionare i primi 20 comuni per popolazione, `outbus` sostituisce il comando `LIMIT` in SAS.
- `create table top20_comuni as`: Crea una nuova tabella con questi comuni.
- `order by Popolazione desc;`: Ordina i comuni in base alla popolazione, in ordine decrescente.

```
/* Eseguo l'analisi solo sui 20 comuni più popolosi */
/* In SAS non c'è il comando LIMIT per SQL, quindi bisogna usare outobs*/
proc sql outobs=20;
   create table top20_comuni as
   select *
   from rifiuti_indicatori
   order by Popolazione desc;
quit;
```

Standardizzazione delle variabili:

- `proc standard`: Standardizza le variabili numeriche per avere media 0 e deviazione standard 1.

```
/* Standardizza le variabili */
proc standard data=top20_comuni
    mean=0
    std=1
    out=top20_comuni_std;
  var _NUMERIC_;
run;
```

> **Nota**
>
> In SAS, a differenza di R, non è necessario calcolare le distanze prima di applicare il clustering gerarchico. In questo progetto, per i metodi utilizzati, verrà sempre impiegata la distanza Euclidea.

2.5.1 Metodo del legame singolo

Unisce i cluster sulla base della distanza minima tra i punti di due cluster. Può creare cluster allungati e meno compatti. Infatti, osservando il dendogramma, si nota che a ogni iterazione dell'algoritmo viene creato un nuovo cluster quasi sempre solo con un comune.

```
/* Metodo del legame singolo*/
proc cluster data=top20_comuni_std method=single outtree=tree_single;
   var _numeric_; /* Sostituisci con le tue variabili */
   id Comune;
run;
```

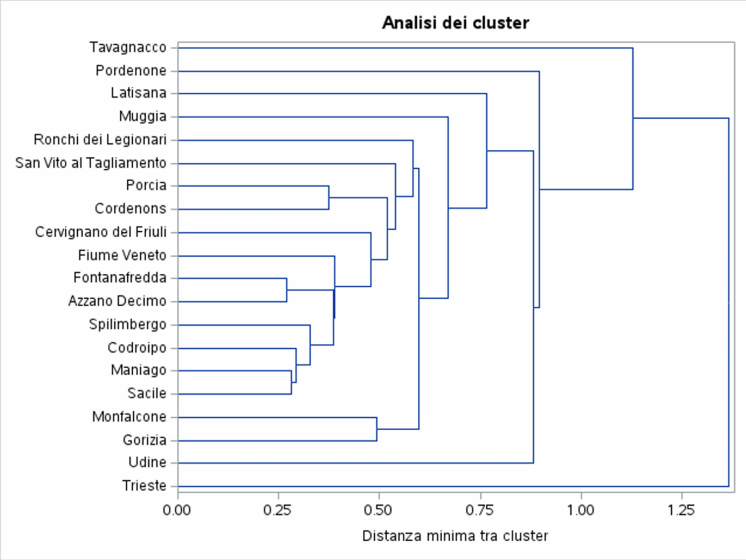


Figura 2.8: Dendogramma del cluster gerarchico con il metodo del legame singolo

2.5.2 Metodo del legame completo

Unisce i cluster considerando la distanza massima tra i punti di due cluster.

```
/* Metodo del legame completo*/
proc cluster data=top20_comuni_std method=complete outtree=tree_complete;
   var _numeric_;
   id Comune;
run;
```

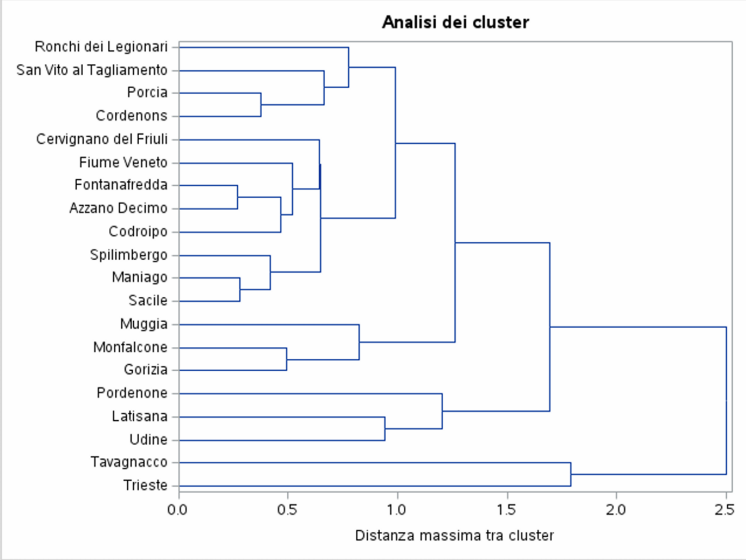


Figura 2.9: Dendogramma del cluster gerarchico con il metodo del legame completo

2.5.3 Metodo del legame medio

Utilizza la distanza media tra tutti i punti di due cluster per unirli, bilanciando tra legame singolo e completo.

```
/* Metodo del legame medio*/
proc cluster data=top20_comuni_std method=average outtree=tree_average;
   var _numeric_;
   id Comune;
run;
```

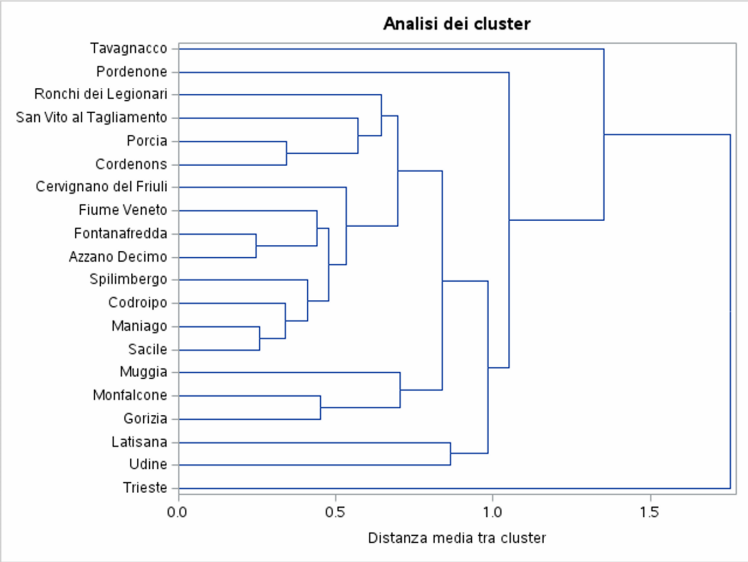


Figura 2.10: Dendogramma del cluster gerarchico con il metodo del legame medio

2.5.4 Metodo di Ward

Minimizza la varianza totale all'interno dei cluster. Produce cluster di dimensioni simili e ben separati.

- `ccc`: Il Cubic Clustering Criterion (CCC) fornisce una misura per valutare l'adeguatezza del numero di cluster scelto. Rimossa in quanto dava valori nulli dal quarto cluster in poi.
- `pseudo`: L'opzione pseudo genera statistiche pseudo-F e pseudo-t^2, che sono utili per identificare i cluster significativi e valutare la qualità della suddivisione dei dati.

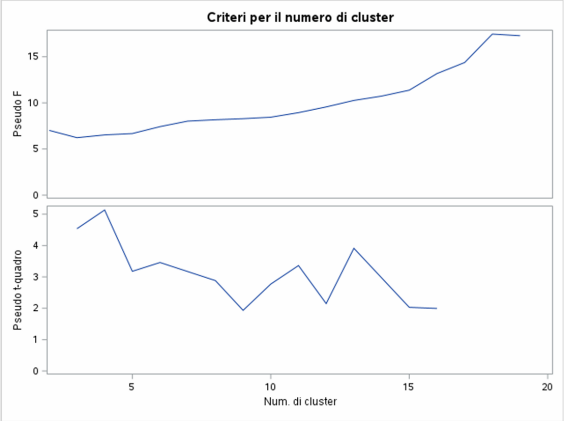
```
/* Metodo di Ward*/
proc cluster data=top20_comuni_std
  method=ward
  outtree=tree_ward
  /* ccc */
  pseudo;
  var _NUMERIC_;
  id Comune;
run;
```

La statistica pseudo-t^2 suggerisce un numero di cluster ottimale di quattro. Tuttavia, due dei quattro cluster sarebbero composti solo da una singola unità. Quindi, ad eccezione dei due outliers si otterrebbero solo due cluster con più unità. Per un analisi più rappresentativa risulta più opportuno scegliere un numero di cluster maggiore. Non si cerca più un massimo globale bensì un massimo locale, in questo caso il massimo locale scelto corrisponde a un numero di sei cluster.

> **Nota**
>
> Si sarebbe potuto anche scegliere il cluster di tredici che con il valore della statistica pseudo-t^2 maggiore al cluster di sei. In questo caso però si avrebbe affrontato il problema opposto dei quattro cluster. Infatti, in questo caso la maggior parte di cluster sarebbero stati composti da una o due unità.

Autovalori della matrice di covarianza				
	Autovalore	Differenza	Proporzione	Cumulativa
1	5.73988156	3.39882993	0.4415	0.4415
2	2.34105163	0.48267458	0.1801	0.6216
3	1.85837705	0.69198777	0.1430	0.7646
4	1.16638928	0.34508212	0.0897	0.8543
5	0.82130715	0.36091652	0.0632	0.9175
6	0.46039063	0.17339660	0.0354	0.9529
7	0.28699403	0.10330790	0.0221	0.9750
8	0.18368613	0.05252006	0.0141	0.9891
9	0.13116607	0.12331809	0.0101	0.9992
10	0.00784798	0.00509616	0.0006	0.9998
11	0.00275183	0.00259518	0.0002	1.0000
12	0.00015665	0.00015665	0.0000	1.0000
13	-.00000000		-0.0000	1.0000

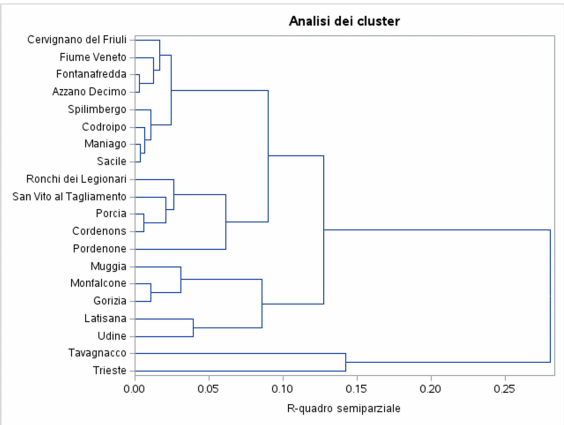
(a) Primi 10 autovalori



(c) Statistiche utili all'identificazione del numero di cluster

Cronologia dei cluster								
Numero di cluster	Cluster uniti		Freq	R-quadro semiparziale	R-quadro	Statistica pseudo F	Pseudo t-quadro	Legame
19	Azzano Decimo	Fontanafredda	2	0.0032	.997	17.3	.	
18	Sacile	Maniago	2	0.0035	.993	17.5	.	
17	Cordenons	Porcia	2	0.0062	.987	14.4	.	
16	CL18	Codroipo	3	0.0070	.980	13.2	2.0	
15	CL16	Spilimbergo	4	0.0106	.970	11.4	2.0	
14	Gorizia	Monfalcone	2	0.0108	.959	10.7	.	
13	CL19	Fiume Veneto	3	0.0125	.946	10.3	3.9	
12	CL13	Cervignano del Friuli	4	0.0169	.929	9.6	2.1	
11	CL17	San Vito al Tagliamento	3	0.0208	.909	8.9	3.4	
10	CL15	CL12	8	0.0248	.884	8.4	2.8	
9	CL11	Ronchi dei Legionari	4	0.0261	.858	8.3	1.9	
8	CL14	Muggia	3	0.0311	.827	8.2	2.9	
7	Udine	Latisana	2	0.0392	.787	8.0	.	
6	Pordenone	CL9	5	0.0611	.726	7.4	3.5	
5	CL7	CL8	5	0.0859	.640	6.7	3.2	
4	CL6	CL10	13	0.0899	.550	6.5	5.1	
3	CL5	CL4	18	0.1275	.423	6.2	4.5	
2	Trieste	Tavagnacco	2	0.1424	.280	7.0	.	
1	CL2	CL3	20	0.2804	.000	.	7.0	

(b) Cronologia dei cluster



(d) Dendogramma del cluster gerarchico

Figura 2.11: Cluster gerarchica usando il metodo di Ward

2.6 Cluster Non Gerarchica

Nel caso della cluster non gerarchica il numero di cluster è fissato a priori, nel nostro caso 6. La cluster gerarchica è utile nei casi di dataset elevati dove l'obiettivo è quello di evidenziare le caratteristiche non di singole unità ma dei gruppi. Per questo motivo sono state prodotte due cluster non gerarchiche: una utilizzando gli stessi dati usati per le altre cluster, e un'altra con i dati di tutti i comuni del FVG.

2.6.1 20 comuni più popolosi

- `maxclusters=6`: fissa il numero massimo di cluster a sei, scelta basasta sulla cluster gerarchica del capitolo precedente.

- `maxiter=100`: definisce il numero massimo di iterazioni per l'algoritmo, garantendo che il processo di clustering giunga a una conclusione stabile entro un numero ragionevole di passi.
- `converge=0.001`: stabilisce la soglia di convergenza, con l'algoritmo che termina se il miglioramento tra iterazioni successive è inferiore a 0.001. Nonostante ci sia già un parametro che definisca l'uscita dal ciclo (`maxiter`), quest'altro garantisce una procedura più veloce, con meno passi, nei casi in cui la soglia di convergenza sia raggiunta.
- `replace=full`: indica che i cluster verranno ricalcolati completamente in ogni iterazione, fornendo una maggiore accuratezza nel posizionamento dei centroidi. Trattando pochi dati ci si può permettere una maggior accuratezza a discapito di un maggior tempo macchina.
- `radius=0`: specifica che non vi è alcuna limitazione sul raggio dei cluster, permettendo ai cluster di adattarsi liberamente alla distribuzione dei dati.

```
proc fastclus data=top20_comuni_std
  out=clus_out
  maxclusters=6
  maxiter=100
  converge=0.001
  replace=full
  radius=0;
  var _numeric_;
run;
```

La procedura FASTCLUS
Replace=FULL Radius=0 Maxclusters=6 Maxiter=100 Converge=0.001

Semi iniziali

Cluster	Indifferenziati	Differenziati	RD	Rsecco	RU	Carta	Legno	Metallo	Organico	Plastica	RAEE	Verde	Vetro
1	3.118943519	-2.618619553	-3.136843029	3.046577639	0.381140985	-1.328459512	-0.714432679	-0.844659980	-2.563131329	0.555522906	-0.020179351	-1.530024168	-3.948952418
2	-0.743973395	0.024725156	0.606239275	-0.825786064	-0.904384185	1.271291276	-0.096805990	1.034074724	-0.751161335	-0.562219896	0.164487951	-0.454675349	0.255357872
3	-0.958858232	1.785046708	1.200110618	-0.922307876	1.211708927	2.408084701	-0.408481352	-0.734604805	2.220124747	-0.615849217	-1.050504398	0.079241815	0.923346402
4	-0.051494129	0.796305828	0.279847705	0.042018061	1.014872169	-0.416765095	-0.210918529	-1.096751259	-0.366266984	-0.675488910	0.756091798	2.335847983	-0.090561213
5	1.125073263	-0.884978958	-1.130103119	1.099317432	0.218325598	-1.375116171	-0.186753170	0.301046625	-1.478686926	0.879632007	1.944515325	0.189413025	0.094306983
6	0.192306920	-1.567440027	-0.806953149	0.545510504	-1.883009500	0.605511914	-2.939055107	-2.532074521	0.176478625	1.151112606	-2.939979304	-1.203627219	0.029043247

Figura 2.12: Semi iniziali assegnati dalla procedura FASTCLUS del software SAS

Distanza minima tra semi iniziali =	4.531424

Cronologia delle iterazioni

		Cambiamento relativo nei semi dei cluster					
Iterazione	Criterio	1	2	3	4	5	6
1	0.7439	0	0.3905	0	0.4102	0.4691	0
2	0.5687	0	0	0	0.2175	0.1525	0
3	0.5522	0	0	0	0	0	0

Il criterio di convergenza è soddisfatto.

Criterio basato su semi finali =	0.5522

Figura 2.13: Cronologia delle iterazioni e del cambio dei semi nei cluster

Statistiche per variabili				
Variabile	STD totale	Entro STD	R-quadro	RSQ/(1-RSQ)
Indifferenziati	1.00000	0.48398	0.827401	4.793789
Differenziati	1.00000	0.51391	0.805400	4.138745
RD	1.00000	0.40674	0.878098	7.203303
Rsecco	1.00000	0.47766	0.831886	4.948344
RU	1.00000	0.91330	0.385389	0.627044
Carta	1.00000	0.77521	0.557198	1.258345
Legno	1.00000	0.76292	0.571123	1.331670
Metallo	1.00000	0.64168	0.696602	2.296000
Organico	1.00000	0.67642	0.662858	1.966113
Plastica	1.00000	0.95696	0.325224	0.481974
RAEE	1.00000	0.74126	0.595129	1.469925
Verde	1.00000	0.58589	0.747063	2.953549
Vetro	1.00000	0.27511	0.944233	16.931685
OVER-ALL	1.00000	0.65998	0.679046	2.115716

Figura 2.14: Statistiche per variabili

Osservando la tabella si nota come nel primo cluster ci siano i comuni che producono una quantità maggiore di rifiuti secchi, indifferenziati e plastici, a discapito di rifiuti differenziati, organici e vetro. Nel cluster 6, invece, si potrebbe ipotizzare che sono stati raggruppati i comuni senza un centro di smaltimento dei rifiuti in quanto hanno dei valori molto bassi per legno, metallo e RAEE (rifiuti di apparecchi elettrici ed elettronici).

Medie dei cluster													
Cluster	Indifferenziati	Differenziati	RD	Rsecco	RU	Carta	Legno	Metallo	Organico	Plastica	RAEE	Verde	Vetro
1	3.118943519	-2.618619553	-3.136843029	3.046577639	0.381140985	-1.328459512	-0.714432679	-0.844659980	-2.563131329	0.555522906	-0.020179351	-1.530024168	-3.948952418
2	-0.336817812	0.111302088	0.304108864	-0.375777601	-0.273693571	0.213529381	0.388385476	0.674782162	0.022993661	-0.231455781	0.155439062	-0.206917378	0.371531202
3	-0.958858232	1.785046708	1.200110618	-0.922307876	1.211708927	2.408084701	-0.408481352	-0.734604805	2.220124747	-0.615849217	-1.050504398	0.079241815	0.923346402
4	-0.700555790	0.865915005	0.809981321	-0.694756514	0.291003871	-0.694030245	-0.040821573	-0.308153484	0.440005952	-0.631892504	0.056017576	1.811426672	-0.016897587
5	0.779363320	-0.327438255	-0.681836761	0.793066321	0.538520921	-0.434585044	0.075144773	-0.428005465	-0.345856627	0.779862256	0.572054927	-0.177674165	-0.167014122
6	0.192306920	-1.567440027	-0.806953149	0.545510504	-1.883009500	0.605511914	-2.939055107	-2.532074521	0.176478625	1.151112606	-2.939979304	-1.203627219	0.029043247

Figura 2.15: Medie dei cluster

2.6.2 Tutti i comuni

Nel caso in cui vengano inclusi tutti i comuni la procedura `FASTCLUS` produce risultati diversi. Dal riepilogo dei cluster si può notare come ci sia solo un cluster contenente un unico comune e un altro cluster con 3 comuni.

Guardando le medie dei cluster si capisce subito come il cluster con un unico comune contenga Lignano Sabbiadoro. Infatti il rifiuto urbano totale è di 1005. Il cluster con i 3 comuni potrebbe essere composto dai comuni meno efficienti nella differenzazione dei rifiuti (RD = 40%).

```
/* Rimuovo Istat e popolazione */
data cluster_indicatori;
    set indicatori;
    drop Istat Popolazione;
run;

/* su tutti i comuni*/
proc fastclus data=cluster_indicatori
```

```
  out=clus_out
  maxclusters=6
  maxiter=100
  converge=0.001
  replace=full
  radius=0;
  var _numeric_;
run;
```

Riepilogo dei cluster

Cluster	Frequenza	Deviazione std RMS	Distanza massima da seme a osservazione	Raggio superato	Cluster più vicino	Distanza tra centroidi dei cluster
1	29	15.7498	97.5683		3	83.9574
2	3	25.9112	87.3476		4	179.8
3	86	11.1544	74.0819		1	83.9574
4	29	19.1134	148.2		3	104.1
5	67	10.7095	84.6523		3	90.2051
6	1	.	0		2	837.6

Figura 2.16: Riepilogo dei cluster

Medie dei cluster

Cluster	Indifferenziati	Differenziati	RD	Rsecco	RU	Carta	Legno	Metallo	Organico	Plastica	RAEE	Verde	Vetro
1	103.340782	112.068054	0.519078	89.592925	215.408836	24.669202	4.222144	1.653852	16.493486	10.580500	3.441792	16.562284	23.999544
2	234.834920	165.786890	0.407347	147.812655	400.621810	35.975477	6.623443	1.342492	38.699116	6.309552	1.818151	13.478838	41.621195
3	58.590750	161.941992	0.735896	45.907839	220.532743	26.652429	11.637814	3.269574	34.854752	4.321285	3.188222	30.144507	22.662680
4	106.596655	187.802938	0.637674	92.523338	294.399594	31.895939	14.849100	2.773797	32.057393	11.559909	5.585131	39.879003	28.295402
5	48.148679	107.046518	0.686420	40.682615	155.195197	19.799322	6.640316	2.387149	26.899070	2.030522	1.589513	6.200887	21.665149
6	399.622421	605.440217	0.602391	378.984341	1005.062637	93.452364	46.667642	3.516757	81.843992	0.000000	6.706425	246.246158	76.964730

Figura 2.17: Medie dei cluster

Capitolo 3

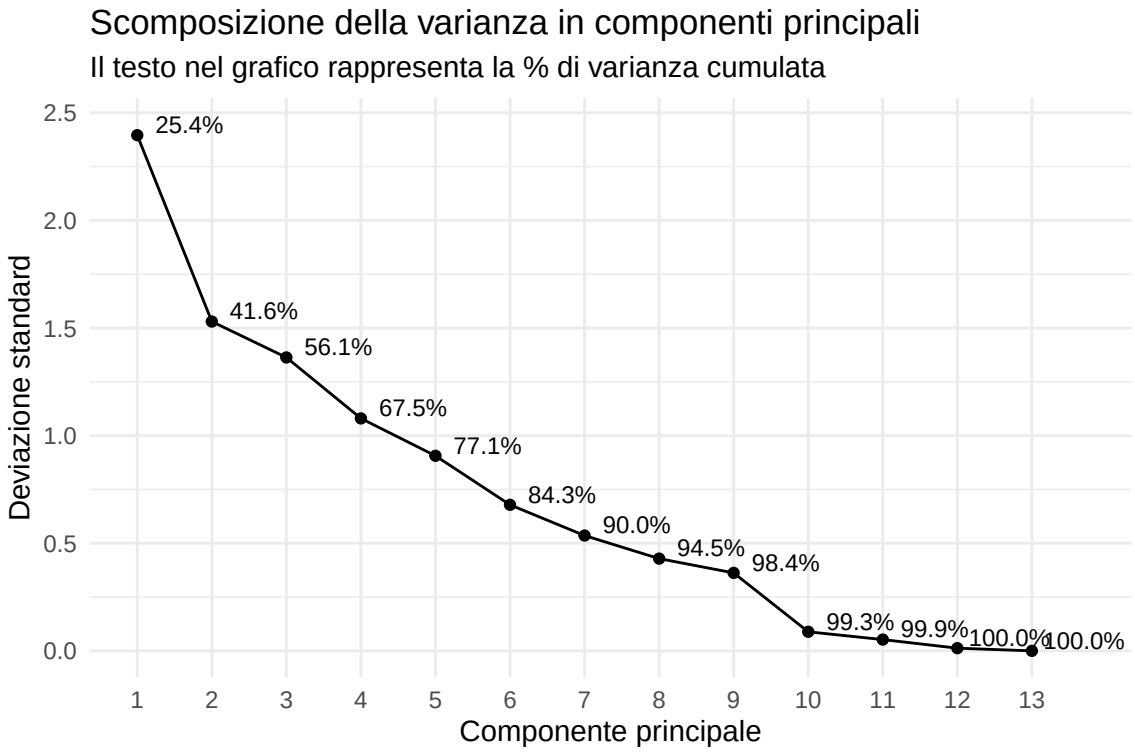
Elaborazioni aggiuntive

3.1 PCA

Per una visualizzazione migliore del clustering eseguito in SAS, si è deciso di rappresentare i cluster in un biplot, ovvero un grafico dove sugli assi cartesiani sono presenti le prime due componenti principali. Considerando le prime due componenti principali, catturerò il 41,6% della varianza totale.

```
# creo il df con i 20 comuni più popolosi
info |>
  # right_join(rifiuti, by = "Istat") |>
  # rename indicatori con I_ per evitare conflitti
  right_join(
    indicatori,
    # indicatori |> rename_with(~str_c("I_", .x), -1),
    by = "Istat"
    ) |>
  arrange(-Popolazione) |>
  slice_head(n = 20) |>
  select(-Popolazione, -Istat) -> data4pca

# calcolo le componenti principali
data4pca |>
  select(where(is.numeric)) |>
  scale() |>
  prcomp() -> pca

# stampo il grafico
tibble(
  sd = pca$sdev,
  componente = 1:length(pca$sdev)
) |>
  mutate(perc = scales::percent(cumsum(sd) / sum(sd), accuracy = .1)) |>
  ggplot(aes(componente, sd, label = perc)) +
  geom_point() +
  geom_line() +
  geom_text(
    nudge_x = .7,
    nudge_y = .05,
```

```
  size = 3
) +
theme_minimal() +
scale_x_continuous(
  # labels = scales::number_format(suffix = "°"),
  breaks = seq(1, length(pca$sdev), by = 1)
  ) +
theme(
  panel.grid.minor.x = element_line(linetype = 0)
) +
labs(
  x = "Componente principale",
  y = "Deviazione standard",
  title = "Scomposizione della varianza in componenti principali",
  subtitle = "Il testo nel grafico rappresenta la % di varianza cumulata"
)
```



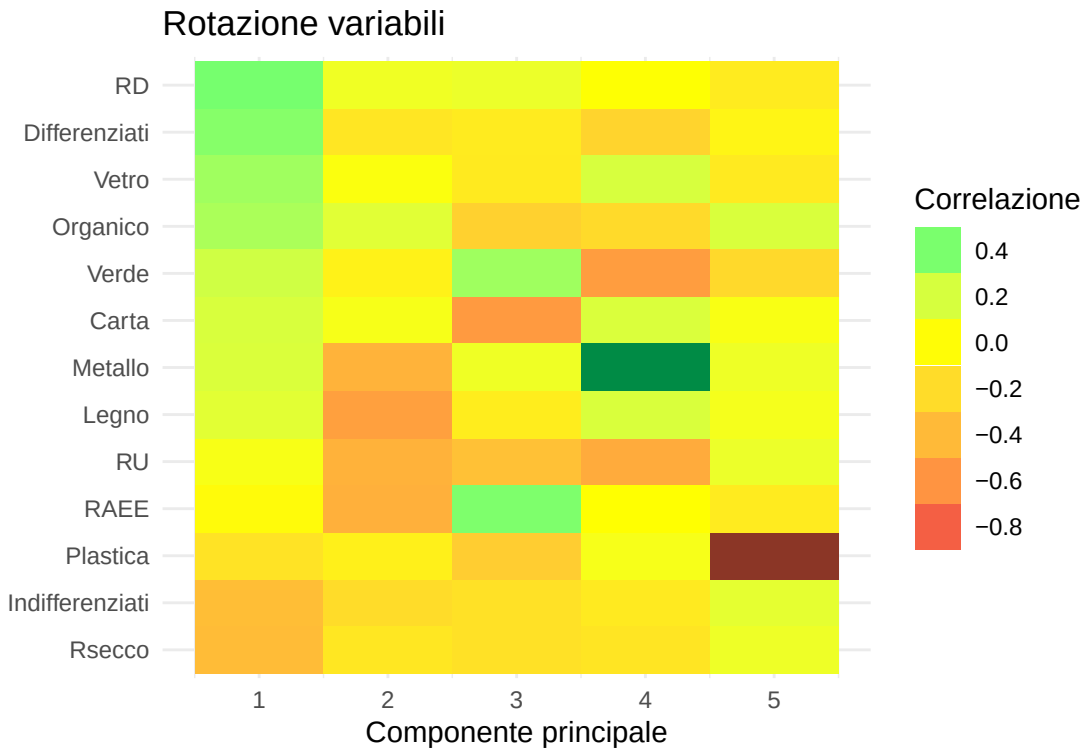
Di seguito un grafico intuitivo per comprendere meglio come le variabili sono state ruotate nelle prime 6 componenti principali. Il correlogramma mostra la correlazione tra le variabili originali e le componenti principali.

Nella prima componente principale si può vedere come i rifiuti differenziati e le loro componenti (*e.g.* vetro, organico, …) sono contrapposti al rifiuto secco e indifferenziato.

> Dettagli di Implementazione
>
> Viene usata la funzione `continuous_scale`, anziché le funzioni standard di `{ggplot2}` in quanto essa offre una maggior flessibilità e personalizzazione. In questo caso ho potuto dar in pasto una palette

con 5 colori personalizzati come input e assegnarli manualmente i valori in modo tale che il colore centrale, il giallo, corrisponda alla correlazione nulla. Per far ciò sono stati simulati i valori da una distribuzione beta con parametri α e β. I parametri sono stati selezionati manualmente, altrimenti per una maggior precisione si sarebbe potuto optare per il pacchetto `{fitdistrplus}` che permette di stimare i parametri di una distribuzione dato un campione in base a diversi metodi (*e.g.* MLE, MSE, ...).

```
pca$rotation |>
  data.frame() |>
  rownames_to_column("rifiuti") |>
  pivot_longer(-rifiuti, names_to = "PC") |>
  mutate(
    across(PC, \(x) str_extract(x, "[:digit:]+") |> as.numeric()),
    # Ordina i rifiuti per i valori della prima colonna
    combined_order = if_else(PC == 1, value, 0),
    across(
      rifiuti,
      \(x) fct_reorder(x, combined_order, .fun = "mean")
    )
    ) |>
  filter(PC < 6) |>
  ggplot(aes(PC, rifiuti, fill = value)) +
  geom_tile() +
  scale_x_continuous(
    breaks = 1:10
  ) +
  continuous_scale(
    aesthetics = "fill",
    scale_name = "palette mia",
    name = "Correlazione",
    palette = scales::gradient_n_pal(
      colours = c("tomato4", "tomato", "yellow", "springgreen", "springgreen4"),
      values = pbeta(seq(0, 1, by = .1), shape1 = 5, shape2 = 6.5)
      ),
    breaks = seq(1, -1, by = -.2)
  ) +
  theme(
    # legend.position = "bottom"
    panel.grid.major = element_line(linetype = 0),
    panel.grid.minor = element_line(linetype = 0),
  ) +
  theme_minimal() +
  labs(
    x = "Componente principale",
    y = "",
    title = "Rotazione variabili"
  )
```



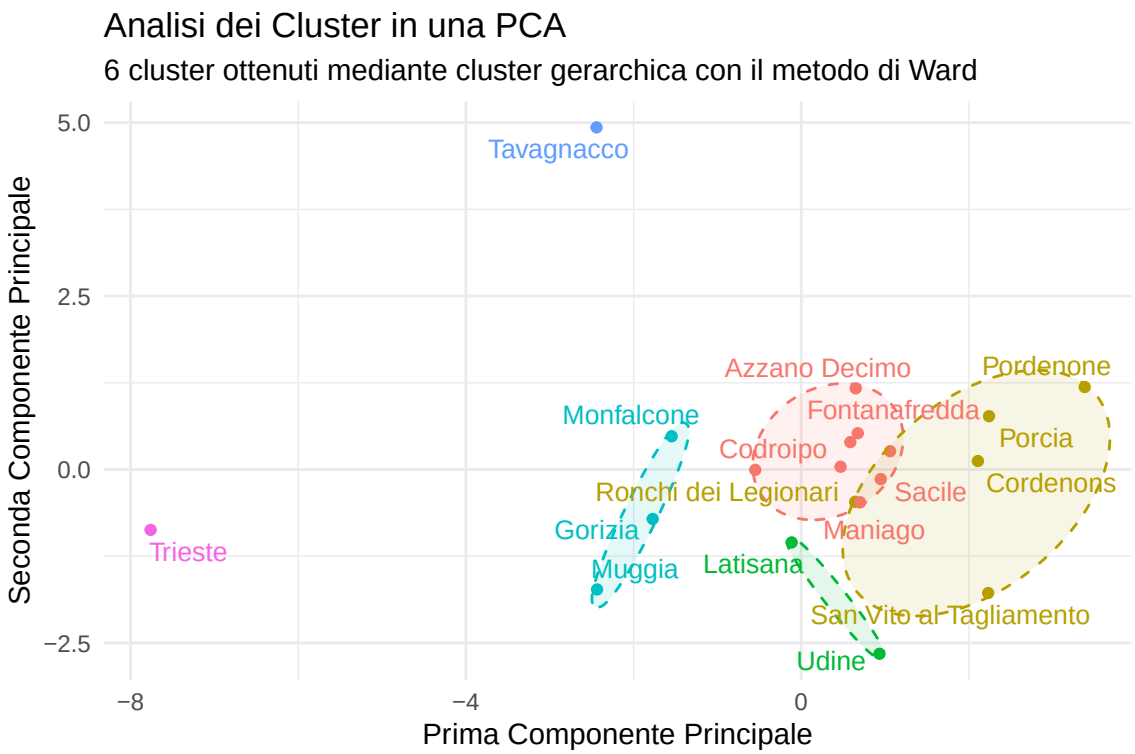
Sono stati ottenuti i 6 cluster, in SAS, dal metodo di Ward. I dati sono stati quindi esportati e importati in R.

```
/* Esporto i dati per visualizzarli in R */
/* Primo passo: Taglio l'albero per ottenere 6 cluster */
proc tree data=tree_ward nclusters=6 out=clusters;
   id Comune;
run;

/* Secondo passo: Visualizzo i comuni e i loro cluster */
proc print data=clusters;
   var Comune cluster;
run;

/* Terzo passo: Esporto i dati */
proc export data=clusters
   outfile="/home/u64115021/sasuser.v94/cluster_top20comuni.csv"
   dbms=csv
   replace;
run;
```

Infine, inserisco i comuni nel grafico e li ragruppo per cluster. Gli insiemi singoletti (Trieste e Tavagnacco) non sono stati cerchiati. Si può osservare una buona separazione tra i cluster.

```
# dati originali
data4pca |>
  # dati post rotazione
  bind_cols(
    pca$x |>
      data.frame()
```

```
) |>
# Unisco i cluster di SAS
left_join(
  read.csv(here::here("data/cluster_top20comuni.csv")),
  by = "Comune"
) |>
ggplot(aes(PC1, PC2, color = CLUSNAME, fill = CLUSNAME)) +
geom_point() +
ggforce::geom_mark_ellipse(
  expand = unit(0.5,"mm"), # riduce gli elissi per una migliore rappresentazione
  alpha = .1,
  size = .5,
  linetype = "dashed"
  ) +
ggrepel::geom_text_repel(
  aes(label = Comune), # se lo metto in aes() principale poi compare anche negli elissi
  size = 3.5,
  # nudge_x = -1.1,
  segment.size = 0.5,  # Riduce lo spessore della linea
  segment.linetype = "dotted"  # Rende la linea tratteggiata con puntini
          ) +
theme_minimal() +
labs(
  x = "Prima Componente Principale",
  y = "Seconda Componente Principale",
  title = "Analisi dei Cluster in una PCA",
  subtitle = "6 cluster ottenuti mediante cluster gerarchica con il metodo di Ward"
) +
theme(legend.position = "none")
```



Osservando il grafico soprastante e il correlogramma con le componenti principali si possono trarre molte conclusioni. Di seguito alcune con una tabella con i dati per confermare ciò:

1. Trieste ha la prima componente principale molto negativa, e d'accordo con il correlogramma, ciò è dovuto da un alto valore di indifferenziati, secco e bassi valori, invece, per la raccolta differenziata.
2. Tavagnacco ha un valore della seconda componente principale estremamente alto. Infatti, non ricicla RAEE, legno e metallo che sono correlati negativamente con la seconda componente principale. Probabilmente è l'unico dei 20 comuni più popolosi del FVG senza un centro di smaltimento dei rifiuti.
3. Pordenone è il comune più attivo nella raccolta differenziata, infatti ha un tasso di raccolta differenziata del 82,1% confronto il 70,4% della media dei 20 comuni più popolosi del FVG. Mentre, Trieste è il comune con il valore più basso. Motivo per la quale Trieste e Pordenone sono i comuni con la distanza massima della prima componente principale.
4. Udine è il comune con il valore della seconda componente principale minore. Un'ipotesi potrebbe essere che tutti i comuni più piccoli si rivolgano alla città per lo smaltimento di rifiuti speciali come legno, metalli e RAEE. Per il legno, il valore è quasi il doppio della media.

```
data4pca |>
  summarise(
    Comune = "Media top 20 FVG",
    across(where(is.numeric), mean),
  ) |>
  bind_rows(
    data4pca |>
      filter(Comune %in% c("Trieste", "Pordenone", "Tavagnacco", "Udine"))
  ) |>
  mutate(
    RD =  scales::percent(RD, accuracy = .1)
  ) |>
```

```
  select(Comune, Indifferenziati, "Tasso Raccolta Differenziata" = RD,
         Legno, Organico) |>
  print_df()
```

Comune	Indifferenziati	Tasso Raccolta Differenziata	Legno	Organico
Media top 20 FVG	66.77	70.4%	10.62	37.79
Trieste	139.64	39.9%	8.04	13.09
Udine	82.62	68.4%	19.28	41.33
Pordenone	44.37	82.1%	9.14	59.19
Tavagnacco	71.27	62.6%	0.00	39.49