

Esame scritto - Corso base R con soluzioni

Guido Ceccherini

2025-05-05

Istruzioni

1. Leggi attentamente le consegne per ogni blocco.
 2. Puoi usare qualsiasi funzione di base o del tidyverse.
-

Tier 1 – Livello Base

Esercizio 1: Caricamento ed esplorazione

Carica il file `aggregationComuniErg5_DAILY_PREC.csv` e salvalo in un oggetto chiamato `pioggia`.

```
pioggia <- read_csv("Data/aggregationComuniErg5_DAILY_PREC.csv")

## Rows: 8878 Columns: 331
## -- Column specification -----
## Delimiter: ","
## dbl   (330): 1_AVG_D, 2_AVG_D, 3_AVG_D, 4_AVG_D, 5_AVG_D, 6_AVG_D, 7_AVG_D, 8...
## date   (1): Date
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

Esercizio 2: Mostra le prime 5 righe del dataset `pioggia`.

```
head(pioggia, 5)

## # A tibble: 5 x 331
##   Date      '1_AVG_D' '2_AVG_D' '3_AVG_D' '4_AVG_D' '5_AVG_D' '6_AVG_D'
##   <date>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 2001-01-01      0.1      0      0      0      0      0
## 2 2001-01-02      6.8      9.7      7.5     12.8     15.2      6
## 3 2001-01-03      0.7      0.1      0      0.7      2      0.5
## 4 2001-01-04      9.2      7.5      8.7      5      8.6      9.2
## 5 2001-01-05      1.3      1.3      1.3      1.5      4.4      0.9
```

```
## # i 324 more variables: '7_AVG_D' <dbl>, '8_AVG_D' <dbl>, '9_AVG_D' <dbl>,
## #   '10_AVG_D' <dbl>, '11_AVG_D' <dbl>, '12_AVG_D' <dbl>, '13_AVG_D' <dbl>,
## #   '14_AVG_D' <dbl>, '15_AVG_D' <dbl>, '16_AVG_D' <dbl>, '17_AVG_D' <dbl>,
## #   '18_AVG_D' <dbl>, '19_AVG_D' <dbl>, '20_AVG_D' <dbl>, '21_AVG_D' <dbl>,
## #   '22_AVG_D' <dbl>, '23_AVG_D' <dbl>, '24_AVG_D' <dbl>, '25_AVG_D' <dbl>,
## #   '26_AVG_D' <dbl>, '27_AVG_D' <dbl>, '28_AVG_D' <dbl>, '29_AVG_D' <dbl>,
## #   '30_AVG_D' <dbl>, '31_AVG_D' <dbl>, '32_AVG_D' <dbl>, '33_AVG_D' <dbl>, ...
```

Esercizio 3: Qual è la dimensione del dataset pioggia? (righe e colonne?)

```
dim(pioggia)
```

```
## [1] 8878 331
```

Esercizio 4: Calcola la pioggia media per la stazione 1_AVG_D nel periodo coperto.

Suggerimenti: per accedere alla colonna 1_AVG_D la sintassi è : pioggia\$1_AVG_D

```
mean(pioggia$`1_AVG_D`, na.rm = TRUE)
```

```
## [1] 2.10793
```

Esercizio 5: Quanti giorni hanno registrato più di 10 mm nella stazione 5_AVG_D?

Suggerimenti: sum(pioggia\$`5_AVG_D`)

Alternativa: usa filter e summarise

pioggia |> filter(.....) |> summarise(n_days = n())

```
sum(pioggia$`5_AVG_D` > 10, na.rm = TRUE)
```

```
## [1] 662
```

```
# alternativa
pioggia |>
  filter(`5_AVG_D` > 10) |>
  summarise(n_days = n())
```

```
## # A tibble: 1 x 1
##   n_days
##   <int>
## 1     662
```

Esercizio 6: Crea una nuova variabile totale_mensile che somma le piogge della stazione '1_AVG_D' per mese.

Suggerimento-> usa la sintassi per creare nuove variabili mese (month) e anno (year): mutate(year = year(Date), month = month(Date)) %>% successivamente usa group_by

Scheletro:

```
pioggia_monthly <- pioggia %>% mutate(year = year(Date), month = month(Date)) %>% group_by(year, month) %>% summarise(.....)
```

```
pioggia_monthly <- pioggia %>%  
  mutate(year = year(Date), month = month(Date)) %>%  
  group_by(year, month) %>%  
  summarise(sum_monthly = sum(`1_AVG_D`, na.rm = TRUE)) |>  
  ungroup()
```

```
## 'summarise()' has grouped output by 'year'. You can override using the  
## '.groups' argument.
```

```
pioggia_monthly
```

```
## # A tibble: 292 x 3  
##   year month sum_monthly  
##   <dbl> <dbl>         <dbl>  
## 1  2001     1          56.6  
## 2  2001     2          13.6  
## 3  2001     3          88.8  
## 4  2001     4          42.8  
## 5  2001     5          75.3  
## 6  2001     6          91.5  
## 7  2001     7          97.2  
## 8  2001     8           50  
## 9  2001     9          94.6  
## 10 2001    10          61.3  
## # i 282 more rows
```

Esercizio 7: Qual è il mese più piovoso?

Suggerimento-> Scheletro:

```
pioggia_monthly %>%  
  filter(max_monthly == ....) %>%  
  select(.....)
```

```
pioggia_monthly %>%  
  filter(sum_monthly == max(sum_monthly)) %>%  
  select(month, sum_monthly)
```

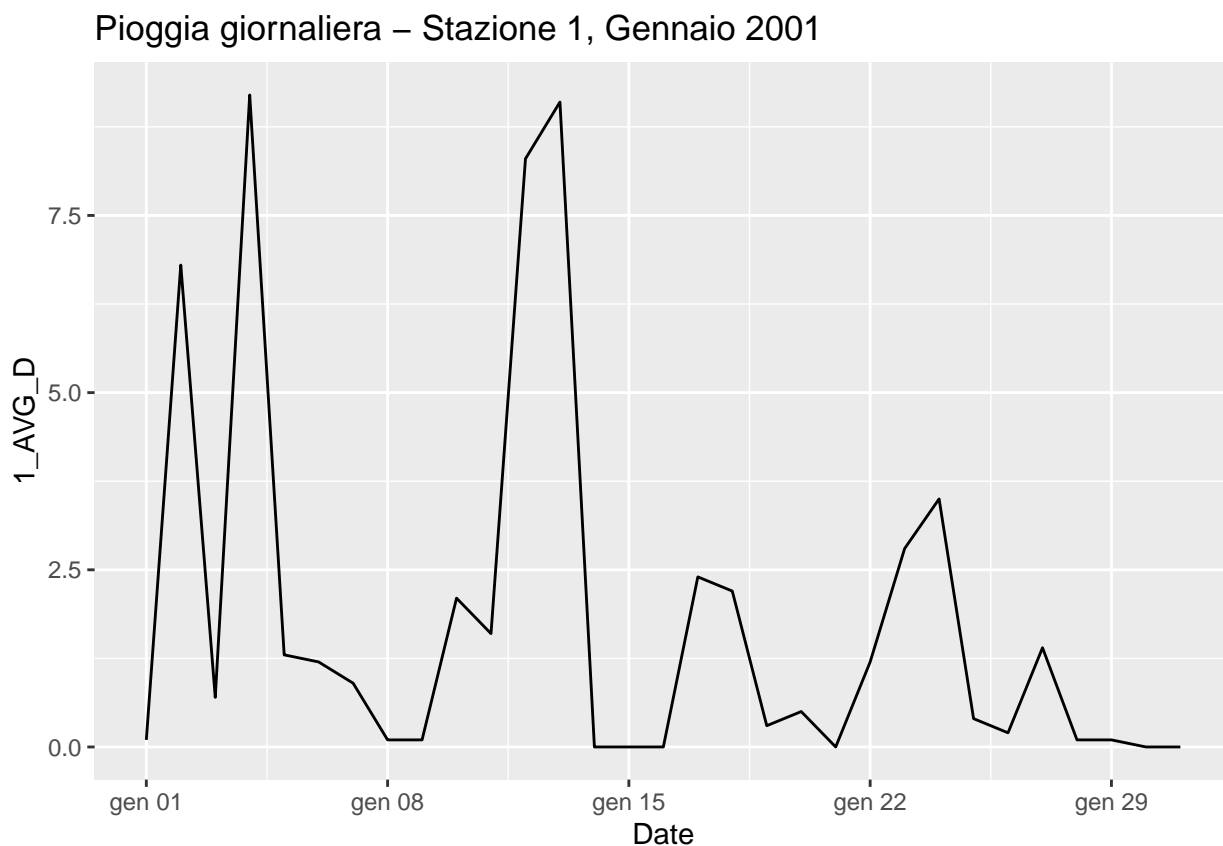
```
## # A tibble: 1 x 2  
##   month sum_monthly  
##   <dbl>         <dbl>  
## 1    11          281.
```

Esercizio 8: Crea un grafico della pioggia giornaliera della stazione '1_AVG_D' nel primo mese del dataset.

Suggerimenti: per accedere al primo mese del dataset la sintassi è :
`filter(month(Date) == 1, year(Date) == 2001) %>%`

Scheletro: `pioggia %>% filter(month(Date) == 1, year(Date) == 2001) %>% ggplot(aes(.....)) + geom_line() #oppure geom_bar()`

```
pioggia %>%  
  filter(month(Date) == 1, year(Date) == 2001) %>%  
  ggplot(aes(x = Date, y = `1_AVG_D`)) +  
  geom_line() +  
  labs(title = "Pioggia giornaliera - Stazione 1, Gennaio 2001")
```



Tier 2 – Livello Avanzato

Esercizio 1: Dati in formato long (BONUS)

```
pioggia_long <- pioggia %>%  
  pivot_longer(cols = -Date, names_to = "Stazione", values_to = "Pioggia")
```

```
pioggia_long %>%
  group_by(Stazione) %>%
  summarise(media = mean(Pioggia, na.rm = TRUE))
```

```
## # A tibble: 330 x 2
##   Stazione media
##   <chr>      <dbl>
## 1 100_AVG_D  2.32
## 2 101_AVG_D  2.06
## 3 102_AVG_D  2.14
## 4 103_AVG_D  2.16
## 5 104_AVG_D  1.91
## 6 105_AVG_D  2.48
## 7 106_AVG_D  2.15
## 8 107_AVG_D  2.29
## 9 108_AVG_D  1.99
## 10 109_AVG_D  1.86
## # i 320 more rows
```

Esercizio 2: Filtraggio e sintesi

```
pioggia_2002 <- pioggia %>%
  mutate(year = year(Date), month = month(Date)) %>%
  filter(year(Date) == 2002) %>%
  group_by(year, month) %>%
  summarise(sum_monthly = sum(`1_AVG_D`, na.rm = TRUE)) |>
  ungroup()
```

'summarise()' has grouped output by 'year'. You can override using the
'.groups' argument.

```
pioggia_2002 %>%
  filter(sum_monthly == max(sum_monthly))
```

```
## # A tibble: 1 x 3
##   year month sum_monthly
##   <dbl> <dbl>      <dbl>
## 1  2002    11         192.
```

Esercizio 3: Grafico con ggplot2

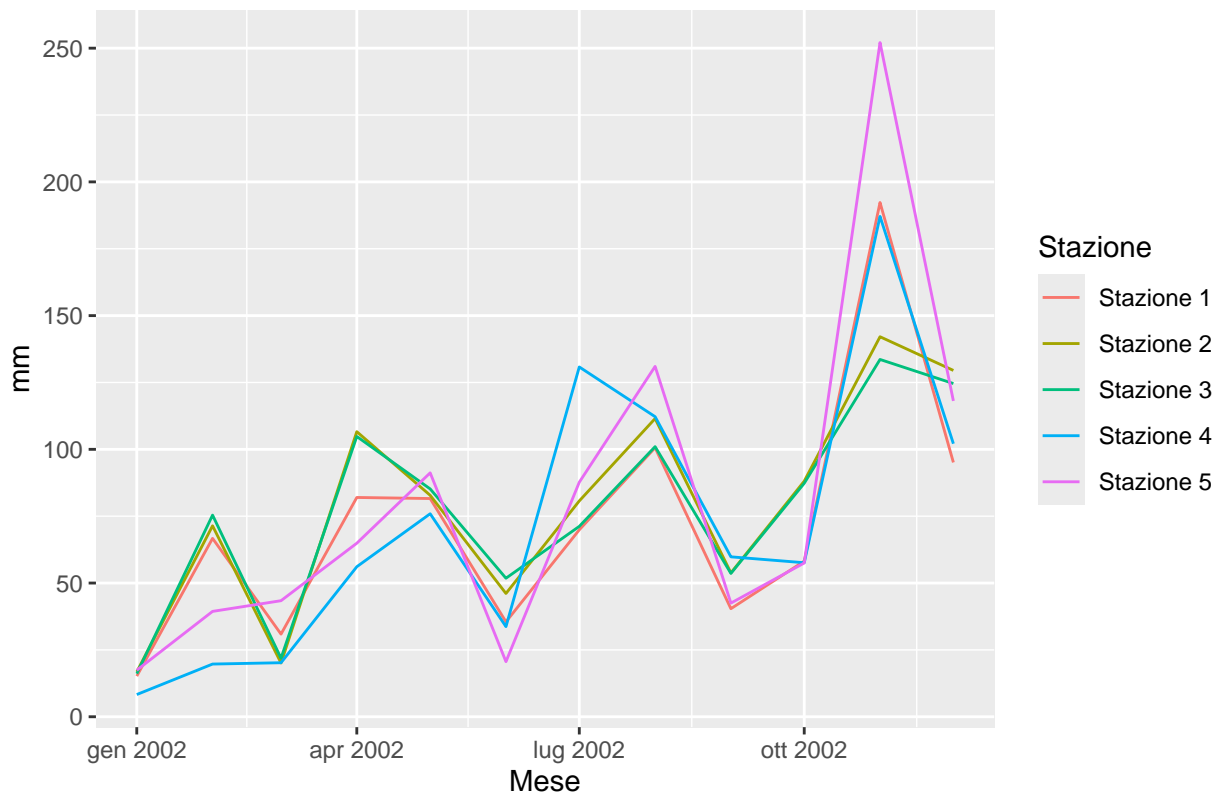
```

pioggia_mensile <- pioggia %>%
  select(Date, `1_AVG_D`:`5_AVG_D`) %>%
  filter(year(Date) == 2002) %>%
  mutate(Mese = floor_date(Date, "month")) %>%
  group_by(Mese) %>%
  summarise(
    Stazione_1 = sum(`1_AVG_D`, na.rm = TRUE),
    Stazione_2 = sum(`2_AVG_D`, na.rm = TRUE),
    Stazione_3 = sum(`3_AVG_D`, na.rm = TRUE),
    Stazione_4 = sum(`4_AVG_D`, na.rm = TRUE),
    Stazione_5 = sum(`5_AVG_D`, na.rm = TRUE),
    .groups = "drop"
  )

# Ricostruisce il grafico usando il formato largo
ggplot(pioggia_mensile, aes(x = Mese)) +
  geom_line(aes(y = Stazione_1, color = "Stazione 1")) +
  geom_line(aes(y = Stazione_2, color = "Stazione 2")) +
  geom_line(aes(y = Stazione_3, color = "Stazione 3")) +
  geom_line(aes(y = Stazione_4, color = "Stazione 4")) +
  geom_line(aes(y = Stazione_5, color = "Stazione 5")) +
  labs(title = "Pioggia cumulata mensile - Prime 5 stazioni (2002)", y = "mm", color = "Stazione")

```

Pioggia cumulata mensile – Prime 5 stazioni (2002)



```

# Alternativa con pivot_longer

```

```

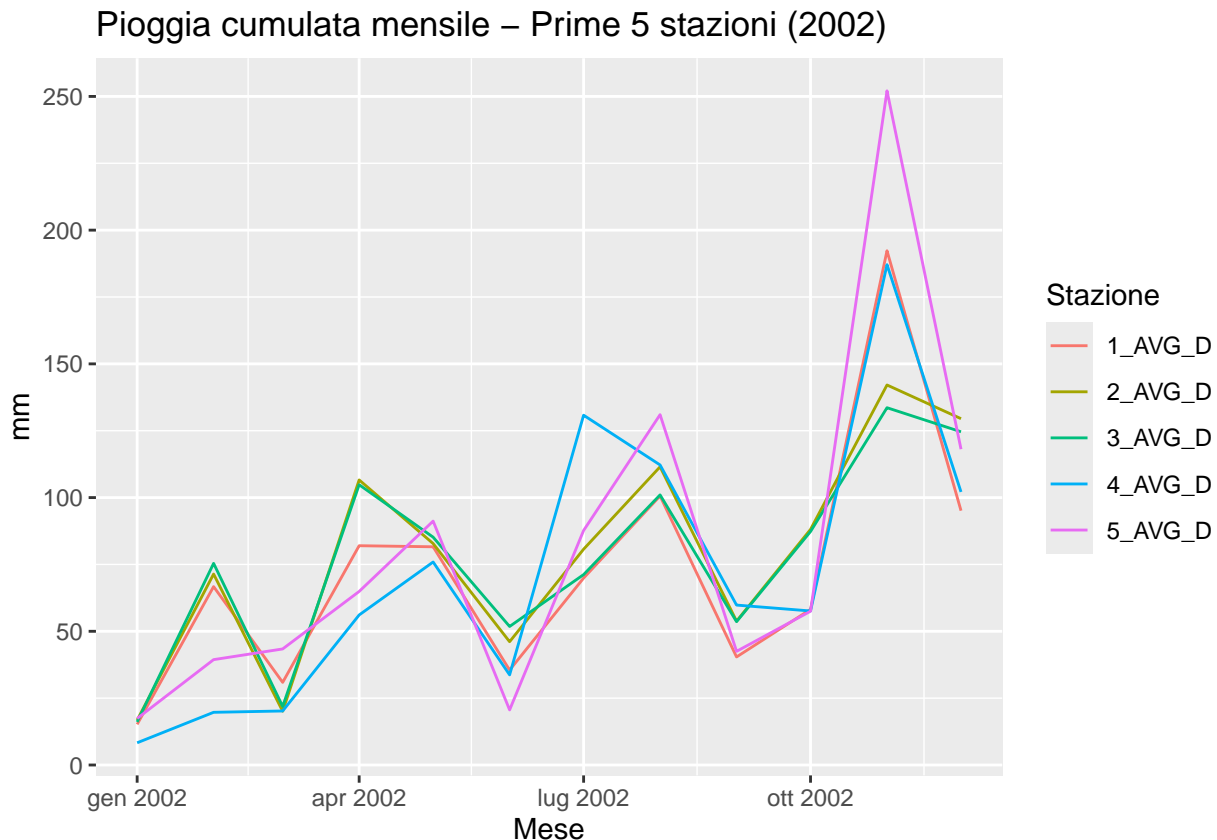
pioggia %>%

```

```

select(Date, `1_AVG_D`:`5_AVG_D`) %>%
filter(year(Date) == 2002) %>%
mutate(Mese = floor_date(Date, "month")) %>%
pivot_longer(cols = -c(Date, Mese), names_to = "Stazione", values_to = "Pioggia") %>%
group_by(Mese, Stazione) %>%
summarise(Pioggia_cumulata = sum(Pioggia, na.rm = TRUE), .groups = "drop") %>%
ggplot(aes(x = Mese, y = Pioggia_cumulata, color = Stazione)) +
geom_line() +
labs(title = "Pioggia cumulata mensile - Prime 5 stazioni (2002)", y = "mm")

```

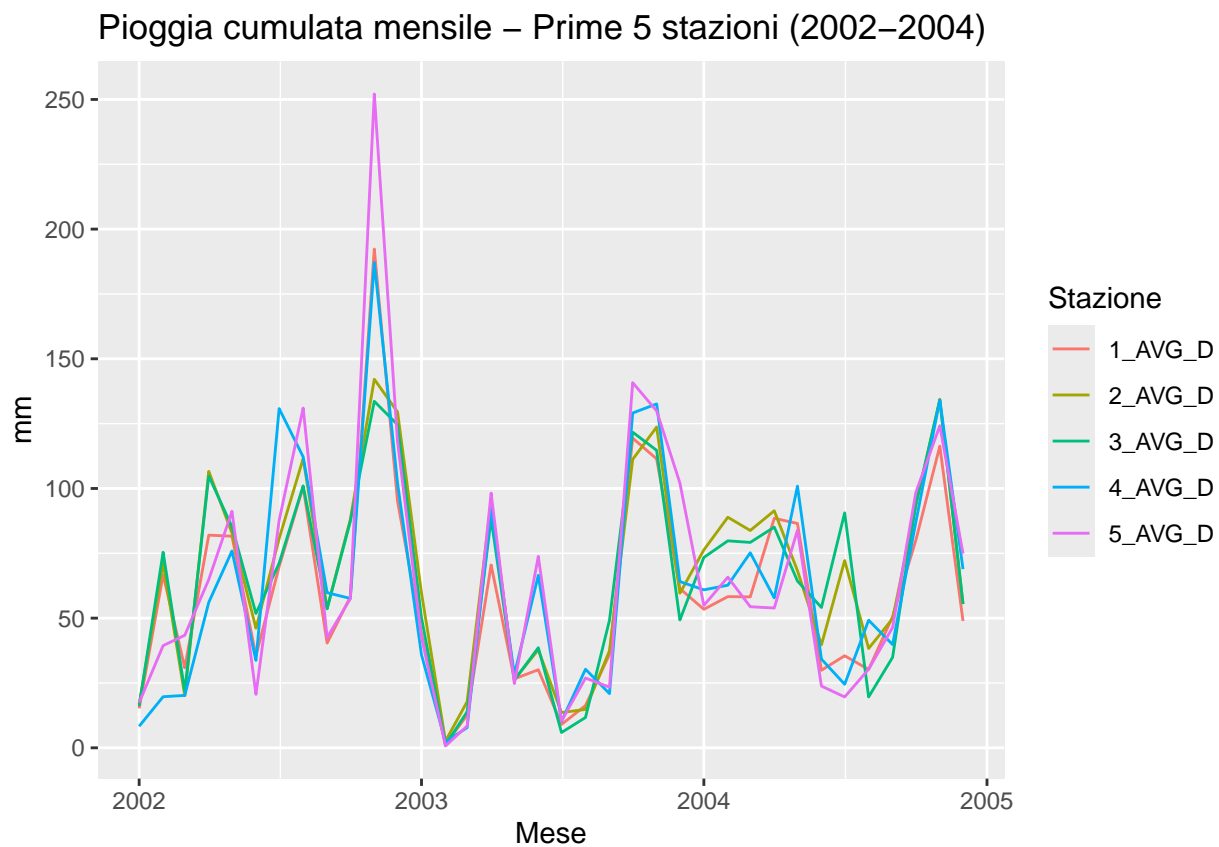


Alternativa con pivot_longer e molti anni

```

pioggia %>%
select(Date, `1_AVG_D`:`5_AVG_D`) %>%
filter(year(Date) %in% c(2002, 2003, 2004)) %>%
pivot_longer(cols = `1_AVG_D`:`5_AVG_D`, names_to = "Stazione", values_to = "Pioggia") %>%
mutate(Mese = floor_date(Date, "month")) %>%
group_by(Mese, Stazione) %>%
summarise(Pioggia_cumulata = sum(Pioggia, na.rm = TRUE), .groups = "drop") %>%
ggplot(aes(x = Mese, y = Pioggia_cumulata, color = Stazione)) +
geom_line() +
labs(title = "Pioggia cumulata mensile - Prime 5 stazioni (2002-2004)", y = "mm")

```



Note finali

- Ricordati di salvare lo script.
- L'uso di commenti nel codice è apprezzato.
- Buon lavoro!