

Introduzione al pacchetto ggplot2

Creazione di grafici e diagrammi con il software R

dott. Luca Menghini Ph.D.

Assegnista di ricerca, Dipartimento di Psicologia,
Università degli Studi di Bologna

luca.menghini3@unibo.it

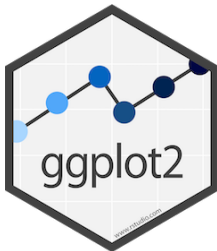
Analisi dei dati in ambito di comunità

Corso di laurea magistrale in Psicologia di comunità, della promozione
del benessere e del cambiamento sociale

Università degli Studi di Padova
Anno Accademico 2021 - 2022



Due mini-lezioni sul pacchetto ggplot2



1. **ggplot2**: La grammatica dei grafici
2. **Get started**: Installare e utilizzare *ggplot2*

ggplot2
●○○○○○○○○○

Esempi
○○○○○○○

Get started
○○○○○○○○○

Aesthetics
○○○○○○○○○

geom & stats
○○○○○○○○○○○

Facets
○○○○○○○

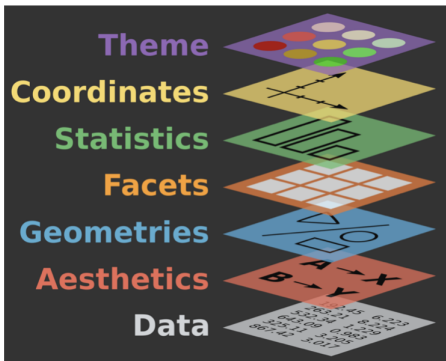
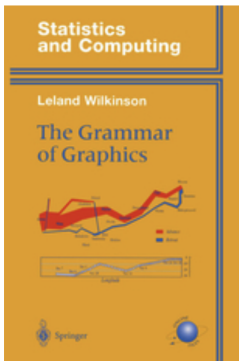
coord & theme
○○○○○○○○○○○○○

Extra
○○○

ggplot2

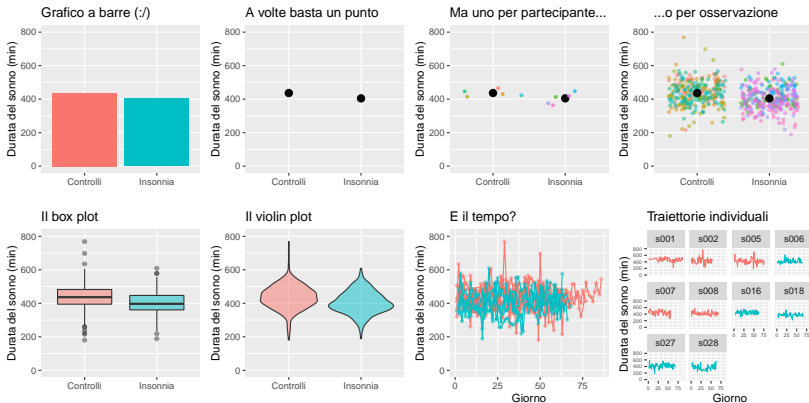
ggplot2: La grammatica dei grafici

ggplot2 è un pacchetto di R che implementa la **grammatica dei grafici** di Leland Wilkinson, semplificando fortemente le procedure necessarie per generare grafici complessi, scomponendoli in micro-elementi di base



ggplot2: Mille modi per visualizzare gli stessi dati

Con `ggplot2` si può partire da un grafico di base, potendone modificare qualsiasi aspetto, e aggiungere moltissimi elementi grafici con solo poche righe di codice.

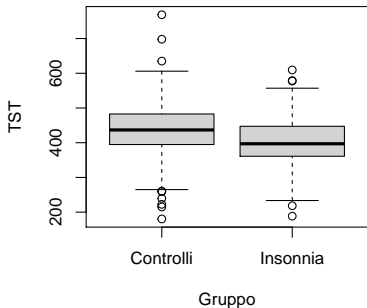


Dati da 10 adolescenti che hanno indossato un Fitbit per due mesi.

Da *graphics* a *ggplot2*: Box plot

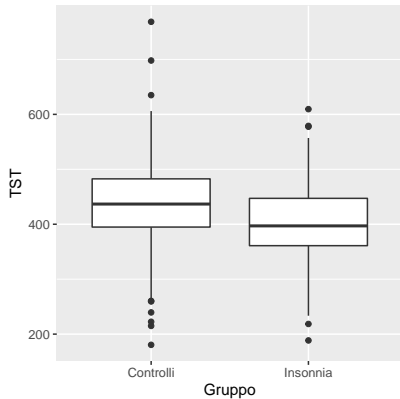
graphics (pacchetto base di R)

```
plot(TST ~ Gruppo, data = mydata)
# plot(x = mydata$Gruppo, y = mydata$TST)
```



ggplot2

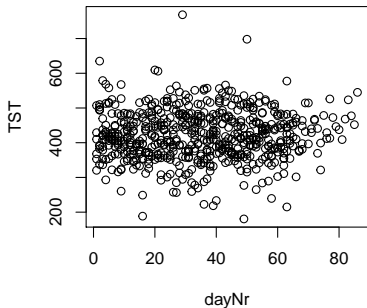
```
ggplot(data=mydata, aes(x=Gruppo, y=TST)) +
  geom_boxplot()
```



Da *graphics* a *ggplot2*: Scatter plot

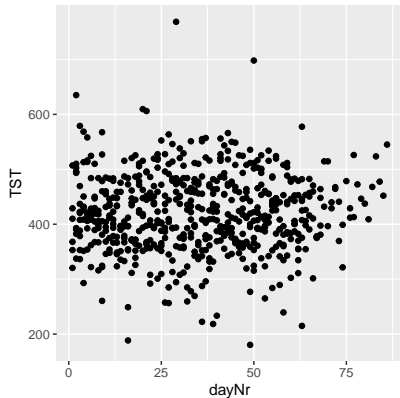
graphics

```
plot(TST ~ dayNr, data = mydata)  
# plot(x = mydata$dayNr, y = mydata$TST)
```



ggplot2

```
ggplot(data = mydata, aes(x = dayNr, y = TST)) +  
  geom_point()
```



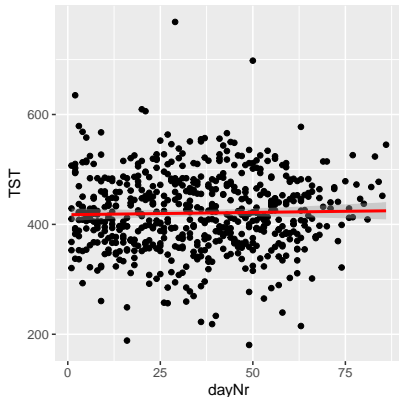
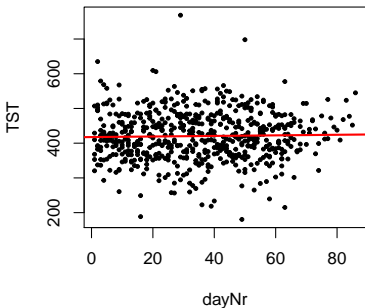
Da *graphics* a *ggplot2*: Regressione

graphics

```
plot(TST ~ dayNr, data = mydata, pch=19, cex=0.5)
abline(lm(TST ~ dayNr, data = mydata),
       col = "red", lwd = 2)
```

ggplot2

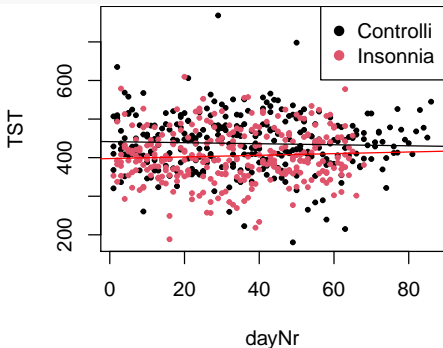
```
ggplot(data = mydata, aes(x = dayNr, y = TST)) +
  geom_point() +
  geom_smooth(method = lm, color = "red")
```



Da *graphics* a *ggplot2*: Distinzione per gruppo

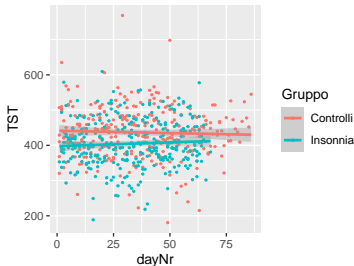
graphics

```
par(mar=c(5,4,0,2))
plot(TST-dayNr,data=mydata,pch=19,cex=0.5, col = Gruppo)
abline(lm(TST-dayNr,data=mydata[mydata$Gruppo=="Insonnia",]),
       col = "red")
abline(lm(TST-dayNr,data=mydata[mydata$Gruppo=="Controlli",]))
legend(x="topright", legend = levels(mydata$Gruppo),
       col = factor(levels(mydata$Gruppo)), pch=19)
```

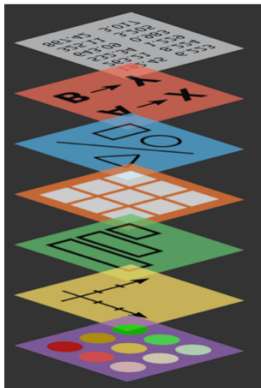


ggplot2

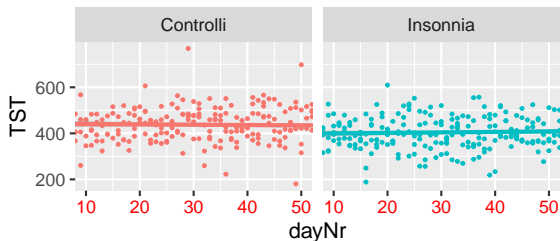
```
ggplot(data = mydata, aes(x=dayNr, y=TST,
                          color=Gruppo)) +
  geom_point(size = 0.5) +
  geom_smooth(method = lm)
```



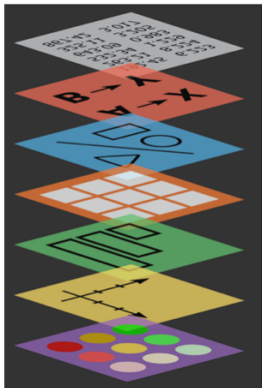
La grammatica dei grafici di ggplot2



```
ggplot(data = mydata, #.....Data
       mapping = aes(x = dayNr, #.....Aesthetics
                    y = TST,
                    color = Gruppo)) +
  geom_point(size = 0.5) + #.....Geometries
  facet_wrap(facets = "Gruppo") + # .....Facets
  geom_smooth(method = lm) + #.....Statistics
  coord_cartesian(xlim = c(10, 50)) + #..Coordinates
  theme(legend.position = "none", #.....Theme
        axis.text.x = element_text(color="red"))
```



qPlot: una via di mezzo



```
qplot(data = mydata, #.....Data

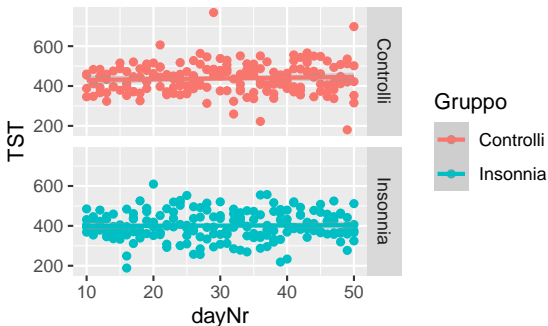
      x = dayNr, #.....Aesthetics

      y = TST, col = Gruppo,

      geom = c("point", #.....Geometries
              "smooth"), method = "lm", #..Statistics

      facets = "Gruppo", # .....Facets

      xlim = c(10, 50)) # .....Coordinates
```



Esempi

Students' sleep data

Dataset incluso nei pacchetti base di R, che mostra l'effetto di due droghe soporifere (group) su 10 pazienti (ID) in termini di ore di sonno aggiuntive (extra)

```
head(sleep) # prime 6 righe
```

	extra	group	ID
1	0.7	1	1
2	-1.6	1	2
3	-0.2	1	3
4	-1.2	1	4
5	-0.1	1	5
6	3.4	1	6

Lancia ?sleep per maggiori dettagli

```
p1 <- ggplot(data = sleep, #.....Data
             mapping = aes(x = group, #.....Aesthetics
                           y = extra,
                           color = group)) +

  geom_point(position = #.....Geometries
             position_jitter()) + # nuvola di punti

  geom_boxplot(aes(fill = group), #.....Statistics
              alpha = 0.5) +

  theme(legend.position = "none") + #.....Theme

  # elementi aggiuntivi

labs(x="Gruppo",y="Ore di sonno aggiuntivo") # titolo assi
```


Ability & Intelligence Tests

Covarianze tra i punteggi a sei test somministrati a 112 individui.

```
# da covarianze a correlazioni
ab <- cov2cor(ability.cov$cov)
```

```
# dati in forma lunga!
```

```
library(reshape2)
```

```
ab <- melt(ab)
```

```
head(ab,5) # prime 5 righe
```

	Var1	Var2	value
1	general	general	1.0000000
2	picture	general	0.4662649
3	blocks	general	0.5516632
4	maze	general	0.3403250
5	reading	general	0.5764799

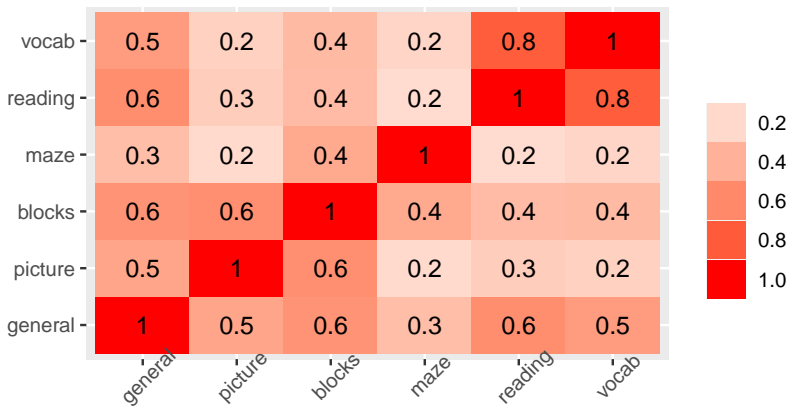
```
p2 <- ggplot(data = ab, #.....Data)
  mapping = aes(x = Var1, #.....Aesthetics
               y = Var2,
               fill = value)) +
  geom_tile() + #.....Geometries
  scale_fill_gradient2(low="white",high="red", # scala e legenda
                      name="", # senza nome
                      guide="legend") +
  geom_text(aes(label = round(value, 1))) +
  theme(axis.title = element_blank(), #.....Theme
        axis.text.x = element_text(angle = 45)) +
  ggtitle("Correlazioni tra tipi di abilità") # titolo
```

Lancia ability.cov per maggiori dettagli

Ability & Intelligence Tests

p2

Correlazioni tra tipi di abilità



Chatterjee–Price Attitude Data

Questionari somministrati ai dipendenti di un'org. finanziaria, aggregati per dipartimento. Ogni valore è la % di risposte favorevoli.

```
# dataset in forma lunga!
```

```
att <- melt(attitude)
```

```
head(att) # prime 6 righe
```

	variable	value
1	rating	43
2	rating	63
3	rating	71
4	rating	61
5	rating	81
6	rating	43

Lancia ?attitude per maggiori dettagli

```
p3 <- ggplot(data = att, #.....Data
             mapping = aes(x = variable, #.....Aesthetics
                           y = value,
                           color = variable,
                           fill = variable)) +

  geom_point(position = #.....Geometries
             position_jitter()) + # nuvola di punti

  geom_violin(alpha = 0.5) + #.....Statistics

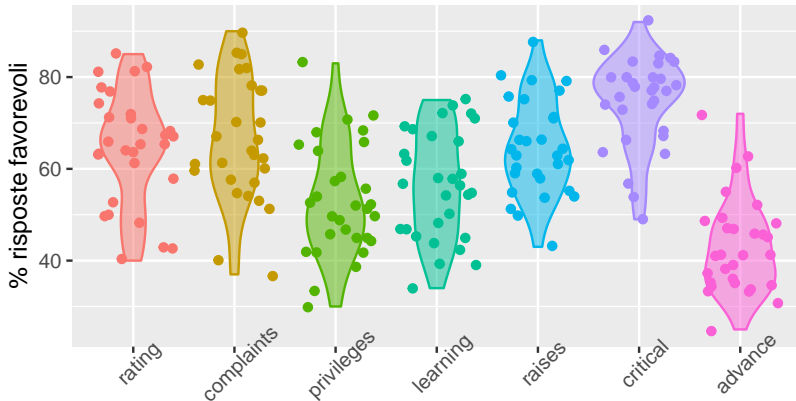
  theme(legend.position = "none", #.....Theme
        axis.text.x = element_text(angle=45)) +

  # elementi aggiuntivi

  labs(x="", y="% risposte favorevoli") # titolo assi
```

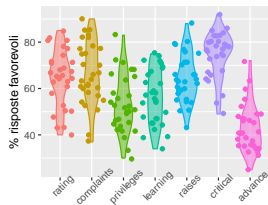
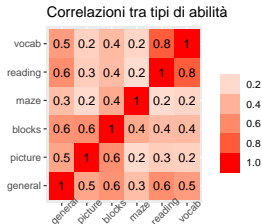
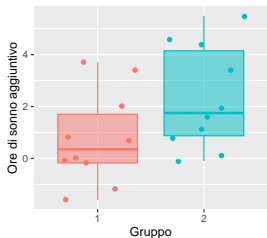
Chatterjee–Price Attitude Data

p3



gridExtra: pannelli di grafici

```
library(gridExtra)  
grid.arrange(p1, p2, p3, nrow = 1)
```



ggplot2
○○○○○○○○○○

Esempi
○○○○○○○○

Get started
●○○○○○○○○

Aesthetics
○○○○○○○○

geom & stats
○○○○○○○○○○

Facets
○○○○○○

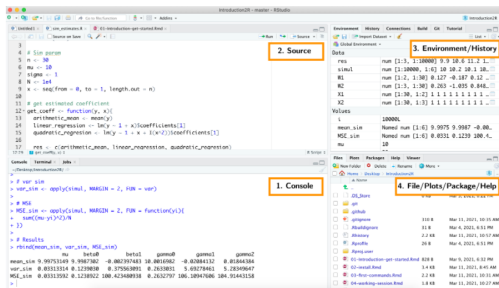
coord & theme
○○○○○○○○○○○○

Extra
○○

Get started

Installare R e RStudio

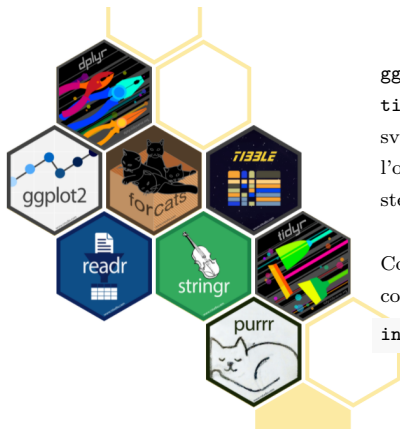
1. Scaricare R dal sito <https://www.r-project.org/> (Menu Download a sinistra > CRAN > primo mirror > selezionare il proprio sistema operativo)
2. Scaricare RStudio dal sito <https://rstudio.com/> (Cliccare su Download > versione FREE > selezionare il proprio sistema operativo)
3. Installare R e RStudio aprendo i due file scaricati



Installare ggplot2

```
# installazione pacchetto (richiede alcuni minuti)  
install.packages("ggplot2")  
  
# apertura pacchetto  
library(ggplot2)
```

tidyverse: l'universo ordinato di R



ggplot2 fa parte della collezione di pacchetti **tidyverse**, creata dalla stessa comunità di sviluppatori. I pacchetti **tidyverse** sono pensati per l'ottimizzazione dell'analisi dei dati e condividono la stessa filosofia, grammatica e struttura.

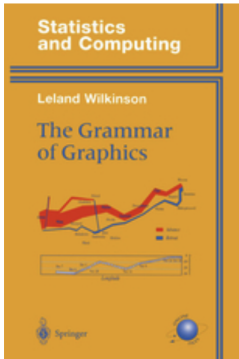
Con il seguente comando è possibile installare contemporaneamente tutti questi pacchetti:

```
install.packages("tidyverse")
```

Per maggiori informazioni: <https://www.tidyverse.org/packages/>

La grammatica dei grafici

ggplot2 è un pacchetto di R che implementa la **grammatica dei grafici** di Leland Wilkinson, semplificando fortemente le procedure necessarie per generare grafici complessi, scomponendoli in micro-elementi di base



Data: Preparare il dataset



I dati vanno organizzati **in forma lunga**, ovvero con una riga per ciascuna delle osservazioni che si vogliono visualizzare.

```
head(sleep, 10) # prime 10 righe
```

	extra	group	ID
1	0.7	1	1
2	-1.6	1	2
3	-0.2	1	3
4	-1.2	1	4
5	-0.1	1	5
6	3.4	1	6
7	3.7	1	7
8	0.8	1	8
9	0.0	1	9
10	2.0	1	10

Data: Preparare il dataset



I dati vanno organizzati **in forma lunga**, ovvero con una riga per ciascuna delle osservazioni che si vogliono visualizzare.

```
head(attitude,5) # prime 5 righe
```

	rating	complaints	privileges	learning	raises	critical	advance
1	43	51	30	39	61	92	45
2	63	64	51	54	63	73	47
3	71	70	68	69	76	86	48
4	61	63	45	47	54	84	35
5	81	78	56	66	71	83	47

```
library(reshape2)
att <- melt(attitude) # dataset in forma lunga
head(att, 5) # prime 5 righe
```

	variable	value
1	rating	43
2	rating	63
3	rating	71
4	rating	61
5	rating	81

Data: Preparare il dataset



I dati vanno organizzati in **forma lunga**, ovvero con una riga per ciascuna delle osservazioni che si vogliono visualizzare.

```
head(attitude,5) # prime 5 righe
```

```
  rating complaints privileges learning raises critical advance
1     43          51          30       39      61       92       45
2     63          64          51       54      63       73       47
3     71          70          68       69      76       86       48
4     61          63          45       47      54       84       35
5     81          78          56       66      71       83       47
```

```
library(tidyr) # soluzione alternativa usando il pacchetto tidyr
```

```
att <- pivot_longer(data = attitude, cols = colnames(attitude))
```

```
head(att, 5) # prime 5 righe
```

```
# A tibble: 5 x 2
```

```
  name      value
  <chr>    <dbl>
1 rating      43
2 complaints  51
3 privileges  30
4 learning    39
5 raises      61
```

Data: Preparare il dataset



I dati vanno organizzati in **forma lunga**, ovvero con una riga per ciascuna delle osservazioni che si vogliono visualizzare.

```
load("esm.RData") # carico dataset
head(esm, 10) # prime 10 righe
```

	ID	sex	day	within.day	TenseAr	Fatigue	WHAT	Workload
1	1	M	1	2	3.000000	5.666667	ANALYSIS	3.75
2	1	M	1	3	2.333333	5.333333	NETWORKING	3.50
3	1	M	1	4	2.000000	3.666667	NETWORKING	3.00
4	1	M	1	7	3.666667	6.333333	AUTHORING	4.50
5	1	M	3	3	3.000000	3.333333	NETWORKING	2.50
6	1	M	3	6	2.666667	6.000000	NETWORKING	1.50
7	1	M	3	7	2.666667	5.333333	OTHER	3.75
8	1	M	5	3	3.666667	3.000000	ACQUISITION	4.50
9	1	M	5	4	2.666667	3.333333	AUTHORING	4.25
10	1	M	5	6	2.333333	6.333333	ACQUISITION	5.25

ggplot2
○○○○○○○○○○

Esempi
○○○○○○○○

Get started
○○○○○○○○

Aesthetics
●○○○○○○○

geom & stats
○○○○○○○○○○

Facets
○○○○○○

coord & theme
○○○○○○○○○○○○○○

Extra
○○

Aesthetics

Aesthetics: mappatura tra dati ed elementi grafici

Aesthetics

x, y: variabili sugli assi x e y

colour: colore delle geometrie

fill: colore di riempimento

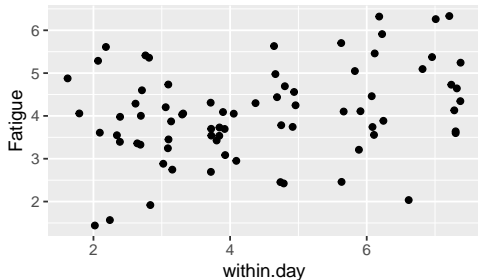
shape: forma dei punti

linetype: tipo di linea

size: dimensione di punti/linee

alpha: trasparenza

```
# seleziono i primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9]], #..Data
        mapping = aes(x = within.day, #.....Aesthetics
                      y = Fatigue)) +
geom_point(position = position_jitter()) #....Geometries
```



Aesthetics: mappatura tra dati ed elementi grafici

Aesthetics

x, y: variabili sugli assi x e y

colour: colore delle geometrie

fill: colore di riempimento

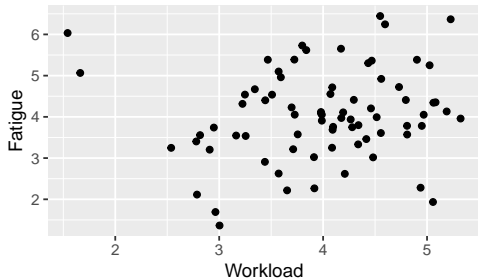
shape: forma dei punti

linetype: tipo di linea

size: dimensione di punti/linee

alpha: trasparenza

```
# seleziono i primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #..Data
        mapping = aes(x = Workload, #.....Aesthetics
                      y = Fatigue)) +
geom_point(position = position_jitter()) #....Geometries
```



Aesthetics: mappatura tra dati ed elementi grafici

Aesthetics

x, y: variabili sugli assi x e y

colour: colore delle geometrie

fill: colore di riempimento

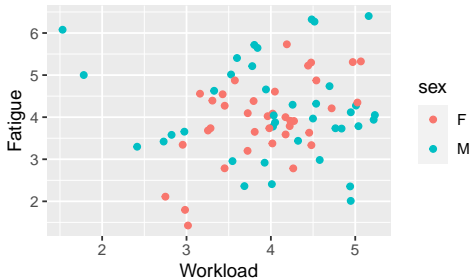
shape: forma dei punti

linetype: tipo di linea

size: dimensione di punti/linee

alpha: trasparenza

```
# seleziono i primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #..Data
        mapping = aes(x = Workload, #.....Aesthetics
                       y = Fatigue, color = sex)) +
geom_point(position = position_jitter()) #....Geometries
```

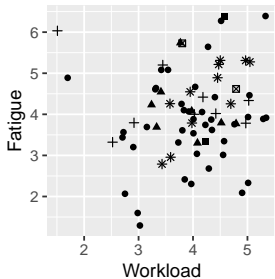


Aesthetics: mappatura tra dati ed elementi grafici

Aesthetics

- x, y: variabili sugli assi x e y
- colour: colore delle geometrie
- fill: colore di riempimento
- shape: forma dei punti
- linetype: tipo di linea
- size: dimensione di punti/linee
- alpha: trasparenza

```
# seleziono i primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #..Data
        mapping = aes(x = Workload, #.....Aesthetics
                       y = Fatigue, shape = WHAT)) +
geom_point(position = position_jitter()) #....Geometries
```



WHAT

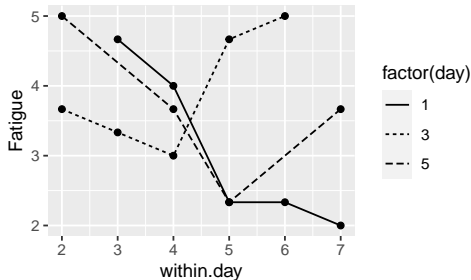
- ACQUISITION
- ▲ ANALYSIS
- AUTHORIZING
- ⊕ NETWORKING
- ⊠ DISSEMINATION
- * ADMINISTRATIVE
- OTHER

Aesthetics: mappatura tra dati ed elementi grafici

Aesthetics

- x, y: variabili sugli assi x e y
- colour: colore delle geometrie
- fill: colore di riempimento
- shape: forma dei punti
- linetype: tipo di linea
- size: dimensione di punti/linee
- alpha: trasparenza

```
# seleziono solo il partecipante 2
ggplot(data = esm[esm$ID==levels(esm$ID)[2],], #.....Data
       mapping = aes(x = within.day, #.....Aesthetics
                    y = Fatigue, linetype = factor(day))) +
geom_line() + geom_point() #.....Geometries
```

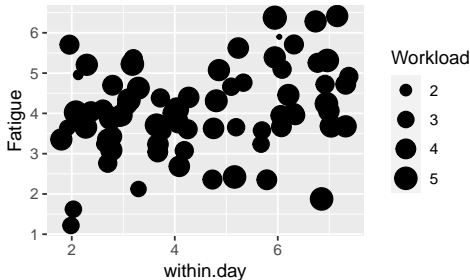


Aesthetics: mappatura tra dati ed elementi grafici

Aesthetics

- x, y: variabili sugli assi x e y
- colour: colore delle geometrie
- fill: colore di riempimento
- shape: forma dei punti
- linetype: tipo di linea
- size: dimensione di punti/linee
- alpha: trasparenza

```
# seleziono i primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #..Data
        mapping = aes(x = within.day, #.....Aesthetics
                       y = Fatigue, size = Workload)) +
geom_point(position = position_jitter()) #....Geometries
```

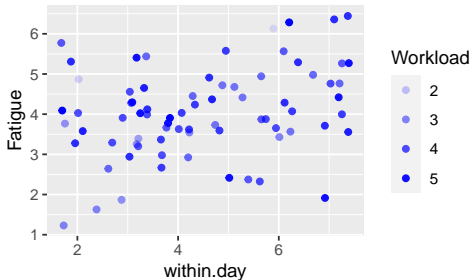


Aesthetics: mappatura tra dati ed elementi grafici

Aesthetics

- x, y: variabili sugli assi x e y
- colour: colore delle geometrie
- fill: colore di riempimento
- shape: forma dei punti
- linetype: tipo di linea
- size: dimensione di punti/linee
- alpha: **trasparenza**

```
# seleziono i primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9]], #..Data
       mapping = aes(x = within.day, #.....Aesthetics
                    y = Fatigue, alpha = Workload)) +
geom_point(position = position_jitter(), #....Geometries
           # stesso colore per tutti i punti
           color = "blue")
```



ggplot2
○○○○○○○○○○

Esempi
○○○○○○○○

Get started
○○○○○○○○

Aesthetics
○○○○○○○○

geom & stats
●○○○○○○○○○

Facets
○○○○○○

coord & theme
○○○○○○○○○○○○○○

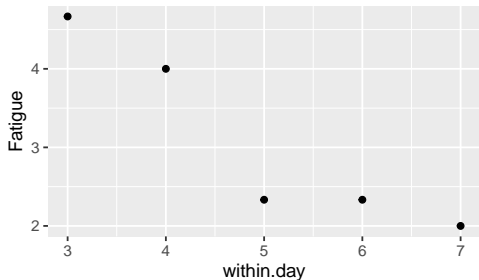
Extra
○○

geom & stats

Geometries: tipo di oggetti geometrici

Geometries

```
p1 <- # seleziono solo il partecipante 2, giorno 1
ggplot(data = esm[esm$ID==levels(esm$ID)[2] & #.....Data
        esm$day==1,],
        mapping = aes(x = within.day, #.....Aesthetics
                     y = Fatigue))
p1 + #.....Geometries
    geom_point()
```



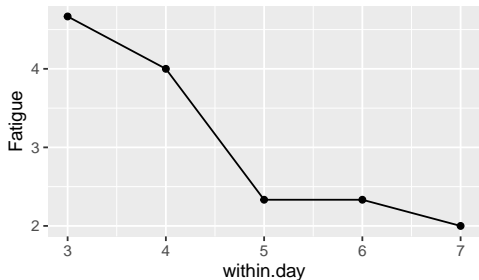
- geom_point(): grafico a punti
- geom_line(): linee tra punti
- geom_boxplot(): grafico a scatola
- geom_violin(): densità distribuzione
- geom_bar(): grafico a barre (categ.)
- geom_histogram(): istogramma
- geom_smooth(): linea di funzione

Geometries: tipo di oggetti geometrici

Geometries

- geom_point(): grafico a punti
- geom_line(): linee tra punti
- geom_boxplot(): grafico a scatola
- geom_violin(): densità distribuzione
- geom_smooth(): linea di funzione
- geom_bar(): grafico a barre (categ.)
- geom_histogram(): istogramma

```
p1 <- # seleziono solo il partecipante 2, giorno 1
ggplot(data = esm[esm$ID==levels(esm$ID)[2] & #.....Data
        esm$day==1,],
        mapping = aes(x = within.day, #.....Aesthetics
                      y = Fatigue))
p1 + # .....Geometries
    geom_point() + geom_line()
```

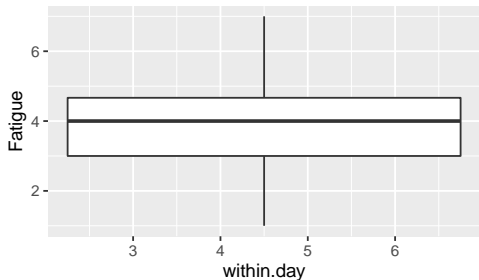


Geometries: tipo di oggetti geometrici

Geometries

- geom_point(): grafico a punti
- geom_line(): linee tra punti
- geom_boxplot(): **grafico a scatola**
- geom_violin(): densità distribuzione
- geom_smooth(): linea di funzione
- geom_bar(): grafico a barre (categ.)
- geom_histogram(): istogramma

```
p2 <- # selezione tutti i dati
      ggplot(data = esm, #.....Data
             mapping = aes(x = within.day, #.....Aesthetics
                           y = Fatigue))
p2 + #.....Geometries
     geom_boxplot()
```



Geometries: tipo di oggetti geometrici

Geometries

geom_point(): grafico a punti

geom_line(): linee tra punti

geom_boxplot(): **grafico a scatola**

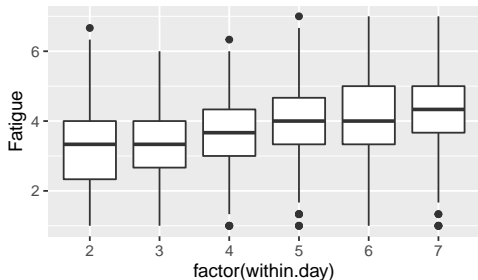
geom_violin(): densità distribuzione

geom_smooth(): linea di funzione

geom_bar(): grafico a barre (categ.)

geom_histogram(): istogramma

```
p2 <- # selezione tutti i dati
      ggplot(data = esm, #.....Data
            mapping = aes(x = factor(within.day), #..Aesthetics
                          y = Fatigue))
p2 + #.....Geometries
     geom_boxplot()
```



Geometries: tipo di oggetti geometrici

Geometries

geom_point(): grafico a punti

geom_line(): linee tra punti

geom_boxplot(): grafico a scatola

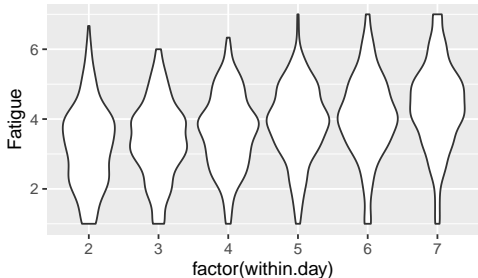
geom_violin(): **densità distribuzione**

geom_smooth(): linea di funzione

geom_bar(): grafico a barre (categ.)

geom_histogram(): istogramma

```
p2 <- # selezione tutti i dati
      ggplot(data = esm, #.....Data
            mapping = aes(x = factor(within.day), #..Aesthetics
                          y = Fatigue))
p2 + # .....Geometries
     geom_violin()
```

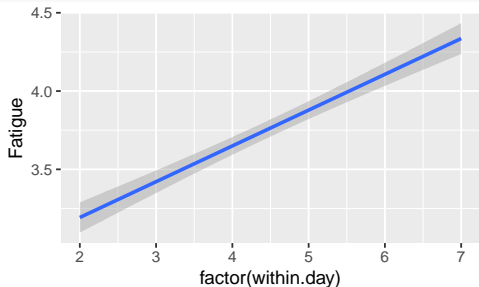


Geometries: tipo di oggetti geometrici

Geometries

- geom_point(): grafico a punti
- geom_line(): linee tra punti
- geom_boxplot(): grafico a scatola
- geom_violin(): densità distribuzione
- geom_smooth(): **linea di funzione**
- geom_bar(): grafico a barre (categ.)
- geom_histogram(): istogramma

```
p2 <- # selezione tutti i dati
      ggplot(data = esm, #.....Data
            mapping = aes(x = factor(within.day), #..Aesthetics
                          y = Fatigue))
p2 + # .....Geometries
      geom_smooth(aes(x = within.day),
                  method="lm")
```

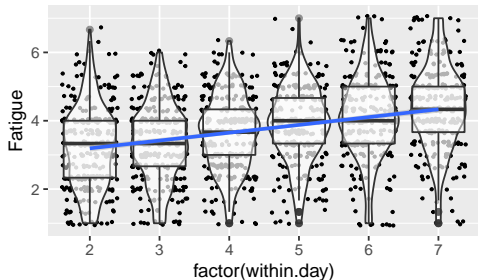


Geometries: tipo di oggetti geometrici

Geometries

- geom_point(): grafico a punti
- geom_line(): linee tra punti
- geom_boxplot(): grafico a scatola
- geom_violin(): densità distribuzione
- geom_smooth(): linea di funzione
- geom_bar(): grafico a barre (categ.)
- geom_histogram(): istogramma

```
p2 +
  geom_point(position = position_jitter(), size = 0.5) +
  geom_violin(alpha = 0.7) +
  geom_boxplot(alpha = 0.5) +
  geom_smooth(aes(x = within.day - 1), method="lm")
```



Geometries: tipo di oggetti geometrici

Geometries

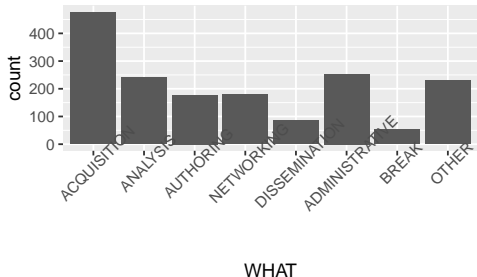
- geom_point(): grafico a punti
- geom_line(): linee tra punti
- geom_boxplot(): grafico a scatola
- geom_violin(): densità distribuzione
- geom_smooth(): linea di funzione
- geom_bar(): **grafico a barre (categ.)**
- geom_histogram(): istogramma

```
p3 <- # selezione tutti i dati

ggplot(data = esm, #.....Data
        mapping = aes(x = WHAT)) #.....Aesthetics

p3 + geom_bar() + #.....Geometries

theme(axis.text.x = element_text(angle = 45))
```

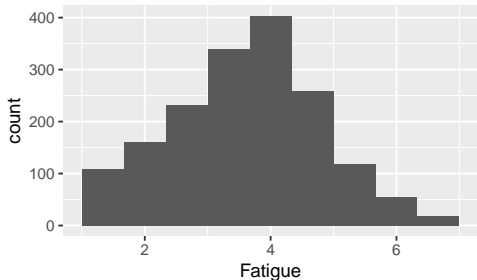


Geometries: tipo di oggetti geometrici

Geometries

- geom_point(): grafico a punti
- geom_line(): linee tra punti
- geom_boxplot(): grafico a scatola
- geom_violin(): densità distribuzione
- geom_smooth(): linea di funzione
- geom_bar(): grafico a barre (categ.)
- geom_histogram(): istogramma

```
p4 <- # selezione tutti i dati
ggplot(data = esm, #.....Data
       mapping = aes(x = Fatigue)) #.....Aesthetics
p4 + geom_histogram(bins=10) #.....Geometries
```



Statistics: trasformazioni statistiche dei dati



Gli elementi `geom` e `stat` sono come due facce della stessa medaglia: ogni `geom` ha un argomento `stat` di default e viceversa, non ha senso aggiungerli entrambi.

Gli elementi `stat` sintetizzano i dati in vari modi, ad esempio contando le osservazioni, visualizzando la retta di regressione, o aggiungendo gli intervalli di confidenza.

- `geom_point()` → `stat_identity()`
- `geom_bar()` → `stat_count()`
- `geom_boxplot()` → `stat_boxplot()`
- `geom_histogram()` → `stat_bin()`
- `geom_smooth()` → `stat_smooth()`

ggplot2
○○○○○○○○○○

Esempi
○○○○○○○○

Get started
○○○○○○○○

Aesthetics
○○○○○○○○

geom & stats
○○○○○○○○○○

Facets
●○○○○○

coord & theme
○○○○○○○○○○○○

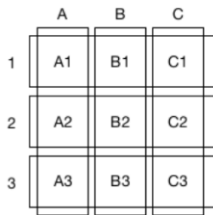
Extra
○○

Facets

Facets: sottodivisioni del grafico



- `facet_null()`: plot singolo (default)
- `facet_grid()`: griglia per 2 variabili
- `facet_wrap()`: griglia per 1 variabile



facet_grid



facet_wrap

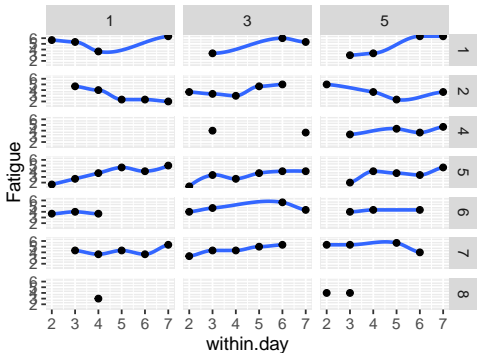
Facets: sottodivisioni del grafico

```
p5 <- # seleziono i primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #.Data
        mapping = aes(x=within.day, y=Fatigue)) + #....aes
geom_smooth(se=FALSE) + geom_point() #.....Geometries

p5 + #.....Facets
    facet_grid(rows = vars(ID), # righe = ID
              cols = vars(day)) # colonne = day
```



- facet_null(): plot singolo (default)
- facet_grid(): griglia per 2 variabili
- facet_wrap(): griglia per 1 variabile

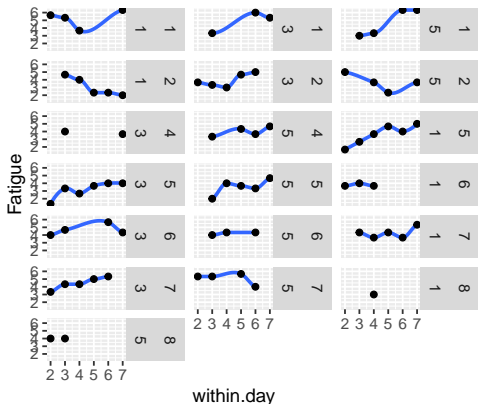


Facets: sottodivisioni del grafico

```
p5 + #.....Facets
facet_wrap(facets = c("ID", "day"), # variabili facets
           strip.position = "right", # posizione intest.
           ncol = 3) # numero di colonne
```



- facet_null(): plot singolo (default)
- facet_grid(): griglia per 2 variabili
- facet_wrap(): griglia per 1 variabile



Facets: sottodivisioni del grafico

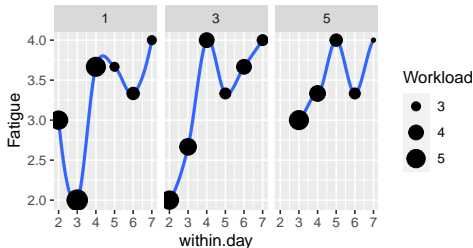


```
# seleziono solo il partecipante 2
p6 <- ggplot(data = esm[esm$ID==levels(esm$ID)[32],], #Data
             mapping = aes(x = within.day, #.....Aesthetics
                           y = Fatigue)) +
  geom_smooth(se=FALSE) + #.....Geometries
  geom_point(aes(size=Workload))

p6 + facet_wrap(facets="day")
```

- facet_null(): plot singolo (default)
- facet_grid(): griglia per 2 variabili
- facet_wrap(): griglia per 1 variabile

Buono per report individuali!



Facets: sottodivisioni del grafico

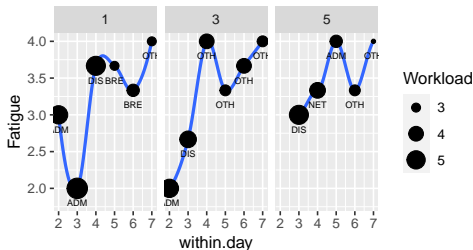


```
# seleziono solo il partecipante 32
p6 <- ggplot(data = esm[esm$ID==levels(esm$ID)[32],], #Data
             mapping = aes(x = within.day, #.....Aesthetics
                           y = Fatigue)) +
  geom_smooth(se=FALSE) + #.....Geometries
  geom_point(aes(size=Workload))

p6 + facet_wrap(facets="day") +
  geom_text(aes(label = substr(WHAT, 1, 3)), size = 2,
            position = position_nudge(y = -0.2))
```

- facet_null(): plot singolo (default)
- facet_grid(): griglia per 2 variabili
- facet_wrap(): griglia per 1 variabile

Buono per report individuali!



ggplot2
○○○○○○○○○○

Esempi
○○○○○○○○

Get started
○○○○○○○○

Aesthetics
○○○○○○○○

geom & stats
○○○○○○○○○○

Facets
○○○○○○

coord & theme
●○○○○○○○○○○○○

Extra
○○

coord & theme

Coordinates: combinazione di x e y

Coordinates

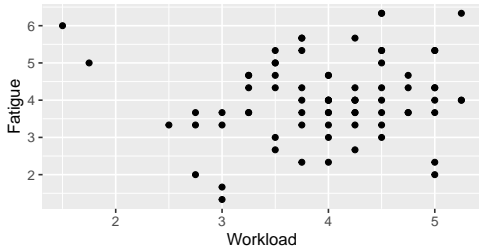
- coord_cartesian(): cartesiane
- coord_flip(): coordinate invertite
- coord_fixed(): coordinate fisse
- coord_polar(): plot circolari
- coord_map(): proiezioni su mappe

```
p7 <- # seleziono primi 9 partecipanti

ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #.Data
        mapping = aes(x = Workload, #.....Aesthetics
                      y = Fatigue)) +

geom_point() #.....Geometries

p7 + coord_cartesian() #.....Coordinates
```



Coordinates: combinazione di x e y

Coordinates

`coord_cartesian()`: cartesiane

`coord_flip()`: coordinate invertite

`coord_fixed()`: coordinate fisse

`coord_polar()`: plot circolari

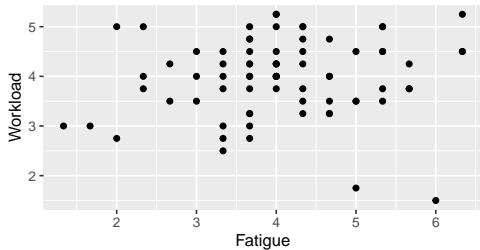
`coord_map()`: proiezioni su mappe

```
p7 <- # seleziono primi 9 partecipanti

ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #.Data
        mapping = aes(x = Workload, #.....Aesthetics
                      y = Fatigue)) +

geom_point() #.....Geometries

p7 + coord_flip() #.....Coordinates
```

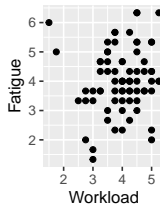


Coordinates: combinazione di x e y

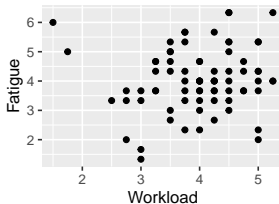
```
p7 + coord_fixed(ratio = 1) # rapporto = 1 (default)
```



- coord_cartesian(): cartesiane
- coord_flip(): coordinate invertite
- coord_fixed(): **coordinate fisse**
- coord_polar(): plot circolari
- coord_map(): proiezioni su mappe



```
p7 + coord_fixed(ratio = 0.5) # rapporto = 0.5
```



Coordinates: combinazione di x e y

```
p7 <- # seleziono primi 9 partecipanti  
  
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #.Data  
        mapping = aes(x = Workload, #.....Aesthetics  
                      y = Fatigue)) +  
  
geom_point() #.....Geometries  
  
p7 + coord_polar() #.....Coordinates
```

Coordinates

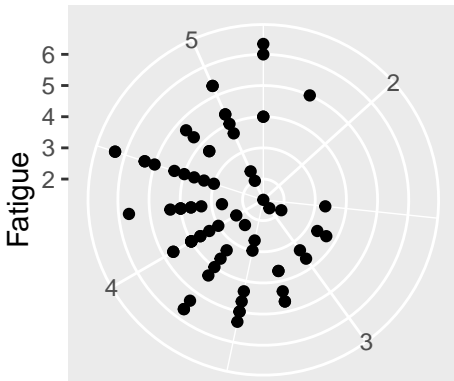
coord_cartesian(): cartesiane

coord_flip(): coordinate invertite

coord_fixed(): coordinate fisse

coord_polar(): plot circolari

coord_map(): proiezioni su mappe



Coordinates: combinazione di x e y

Coordinates



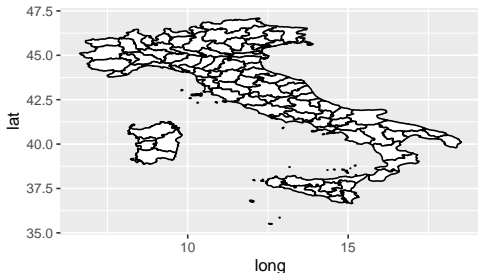
- coord_cartesian(): cartesiane
- coord_flip(): coordinate invertite
- coord_fixed(): coordinate fisse
- coord_polar(): plot circolari
- coord_map(): proiezioni su mappe

```
library(maps)

p8 <- # coordinate mappa Italia dal pacchetto maps

ggplot(data = map_data("italy"), #.....Data
       mapping = aes(x = long, y = lat, #.....Aesthetics
                    group = group)) +
  geom_polygon(fill = "white", # .....Geometries
             colour = "black")
```

```
p8 + coord_cartesian() #.....Coordinates
```



Coordinates: combinazione di x e y

Coordinates

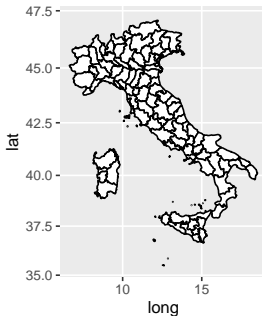
- coord_cartesian(): cartesiane
- coord_flip(): coordinate invertite
- coord_fixed(): coordinate fisse
- coord_polar(): plot circolari
- coord_map(): **proiezioni su mappe**

proietta porzione della Terra su piano
2D (default: proiezione di Mercatore)

```
library(maps)

p8 <- # coordinate mappa Italia dal pacchetto maps
      ggplot(data = map_data("italy"), #.....Data
            mapping = aes(x = long, y = lat, #.....Aesthetics
                          group = group)) +
      geom_polygon(fill = "white", # .....Geometries
                  colour = "black")

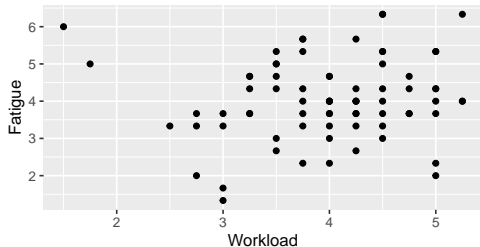
p8 + coord_map() #.....Coordinates
```



Theme: tutto ciò che non è legato ai dati



```
p7 <- # seleziono primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9]], #.Data
        mapping = aes(x = Workload, #.....Aesthetics
                      y = Fatigue)) +
geom_point() #.....Geometries
p7 + theme_gray() #.....Theme
```



Temi completi

`theme_gray()`: quello di `ggplot2`

`theme_bw()`: background white

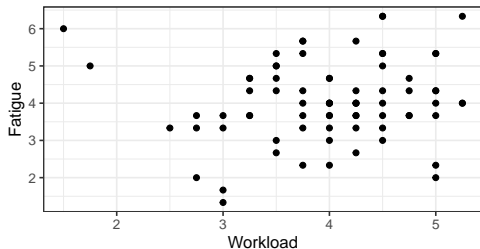
`theme_classic()`: classico cartesiano

`theme_minimal()`: senza background

Theme: tutto ciò che non è legato ai dati



```
p7 <- # seleziono primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #.Data
        mapping = aes(x = Workload, #.....Aesthetics
                      y = Fatigue)) +
geom_point() #.....Geometries
p7 + theme_bw() #.....Theme
```



Temi completi

`theme_gray()`: quello di ggplot2

`theme_bw()`: **background white**

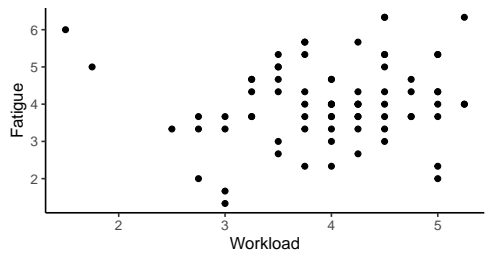
`theme_classic()`: classico cartesiano

`theme_minimal()`: senza background

Theme: tutto ciò che non è legato ai dati



```
p7 <- # seleziono primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9]], #.Data
        mapping = aes(x = Workload, #.....Aesthetics
                      y = Fatigue)) +
geom_point() #.....Geometries
p7 + theme_classic() #.....Theme
```



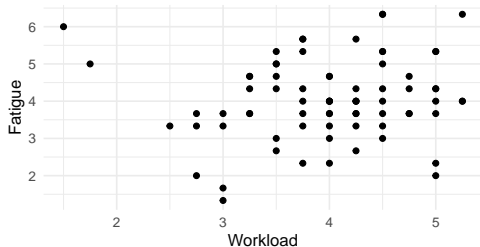
Temi completi

- theme_gray(): quello di ggplot2
- theme_bw(): background white
- theme_classic(): classico cartesiano
- theme_minimal(): senza background

Theme: tutto ciò che non è legato ai dati



```
p7 <- # seleziono primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #.Data
        mapping = aes(x = Workload, #.....Aesthetics
                      y = Fatigue)) +
geom_point() #.....Geometries
p7 + theme_minimal() #.....Theme
```



Temi completi

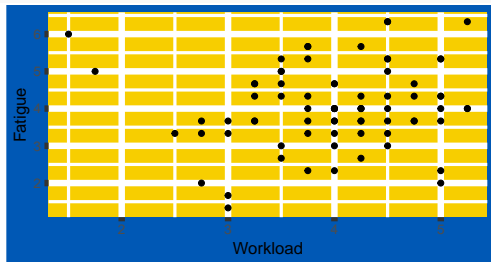
- theme_gray(): quello di ggplot2
- theme_bw(): background white
- theme_classic(): classico cartesiano
- theme_minimal(): senza background

Theme: tutto ciò che non è legato ai dati



```
p7 <- # seleziono primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9],], #.Data
        mapping = aes(x = Workload, #.....Aesthetics
                      y = Fatigue)) +
geom_point() #.....Geometries

p7 + theme(line = element_line(size = 2), #.....Theme
           panel.background = element_rect(fill="#F7CE00"),
           rect = element_rect(fill = "#0058B5",
                               linetype = 1))
```



theme(line, rect,
panel.background, text, title,
legend.position, ...): modifica
elementi del tema

line = element_line(): tutte le linee

rect = element_rect(): bordi

text = element_text(): tutto il testo

...

element_blank(): rimuove l'elemento

Theme: tutto ciò che non è legato ai dati



theme(line, rect,
panel.background, text, title,
legend.position, ...): modifica
elementi del tema

line = element_line(): tutte le linee

rect = element_rect(): bordi

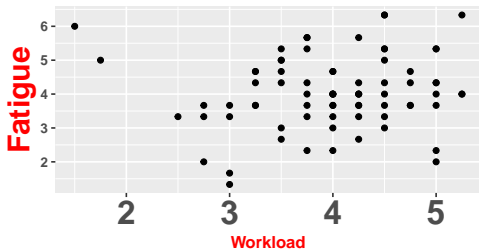
text = element_text(): tutto il testo

...

element_blank(): rimuove l'elemento

```
p7 <- # seleziono primi 9 partecipanti
ggplot(data = esm[esm$ID%in%levels(esm$ID)[1:9]], #.Data
        mapping = aes(x = Workload, #.....Aesthetics
                      y = Fatigue)) +
geom_point() #.....Geometries

p7 + theme(axis.title = element_text(colour = "red"),
           axis.title.y = element_text(size = 20),
           axis.text.x = element_text(size = 22),
           text = element_text(face = "bold"))
```



ggplot2
○○○○○○○○○○

Esempi
○○○○○○○○

Get started
○○○○○○○○

Aesthetics
○○○○○○○○

geom & stats
○○○○○○○○○○

Facets
○○○○○○

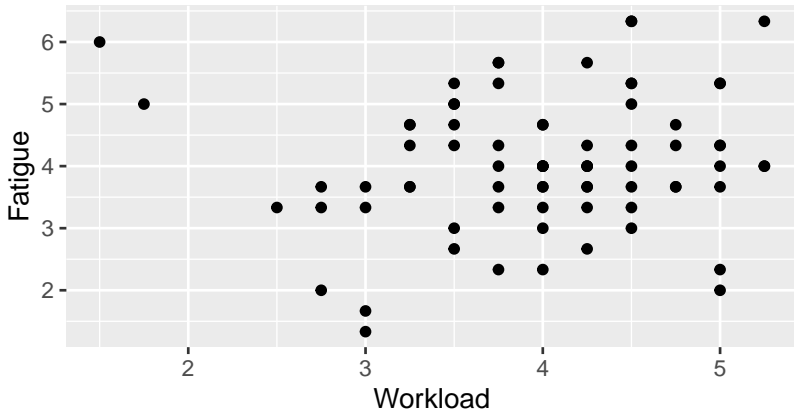
coord & theme
○○○○○○○○○○○○

Extra
●○○

Extra

Salvare un ggplot in alta risoluzione

p7



```
ggsave(filename = "myplot.tiff", plot = last_plot(), dpi = 300)
ggsave(filename = "myplot.tiff", plot = p7, dpi = 300)
```

Riferimenti

Wilkinson, L. (2012). The grammar of graphics. In Handbook of computational statistics (pp. 375-414). Springer, Berlin, Heidelberg.

Link utili

- Corso base di R: <https://psicostat.github.io/Introduction2R/>
- Schema riassuntivo di tutte le funzioni usate da *ggplot2*:
<https://github.com/rstudio/cheatsheets/blob/main/data-visualization-2.1.pdf>
- tidyVerse function reference: <https://ggplot2.tidyverse.org/reference/>

Lecture consigliate

- Pastore, M., Lionetti, F., Altoè, G. (2017). When One Shape Does Not Fit All: A Commentary Essay on the Use of Graphs in Psychological Research. *Frontiers in Psychology*, 8, 1666.