Analisi descrittiva del dataset "heart.csv"

Renz Joshua Villanueva & Binod Comini

28 Febbraio 2021

1 Dataset

Questo dataset contiene dati riguardate il cuore. Per avere una prima visione delle variabili raccolte, abbiamo caricato il dataset "heart.csv" nella nuova variabile dataset con la funzione **read.csv(*dataset*)** e abbiamo, di conseguenza, analizzato la struttura dei dati raccolti.

	x	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1	1	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
2	2	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
3	3	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
4	4	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
5	5	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
6	6	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
7	7	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
8	8	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
9	9	52	1	2	51	199	1	1	162	0	0.5	2	0	3	1
10	10	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
11	11	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
12	12	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
13	14	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
14	14	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
15	15	58	0	3	150	283	1	0	162	0	1.0	2	0	2	1
16	16	50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
17	18	58	0	2	120	340	0	1	172	0	0.0	2	0	2	1
18	18	66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
19	19	43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
20	20	69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
21	21	59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
22	22	44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
23	23	42	1	0	140	226	0	1	178	0	0.0	2	0	2	1
24	24	61	1	2	150	243	1	1	137	1	1.0	1	0	2	1
25	25	40	1	3	140	199	0	1	178	1	1.4	2	0	3	1
26	26	71	0	1	160	302	0	1	162	0	0.4	2	2	2	1
27	27	59	1	2	150	212	1	1	157	0	1.6	2	0	2	1
28	28	51	1	2	110	175	0	1	123	0	0.6	2	0	2	1
29	29	65	0	2	140	417	1	0	157	0	0.8	2	1	2	1
30	30	53	1	2	130	197	1	0	152	0	1.2	0	0	2	1
31	31	41	0	1	105	198	0	1	168	0	0.0	2	1	2	1

Figure 1: Sceen dataset heart.csv

2 Correzione dei dati

2.1 valori NA

Osservando il dataset ci siamo accorti della presenza di valori NA dovuti, ad esempio, al data-entry manuale; quindi abbiamo creato un ciclo che controllasse la presenza di tali valori. Tutti i valori NA sono stati cancellati definitivamente dal dataset con la funzione ***dataset* <- na.omit (*dataset)***.

2.2 Colonne non necessarie

Abbiamo proseguito l'analisi del dataset controllando e rimuovendo le colonne non ritenute necessarie. Infatti, attraverso la funzione **view(*dataset*)**, l'abbiamo visualizzato e, attraverso la funzione **subset()**, abbiamo rimosso le colonne superflue; in questo specifico caso la colonna X perché ritenuta inutile.

2.3 Rinominazione delle colonne

Terminato quest'ultimo passaggio, abbiamo rinominato le colonne in maniera appropriata, descrivendone, di ciascuna, il tipo di attributo. Abbiamo stampato il dataset con la funzione 'str(dataset)' e, con la funzione
names(*dataset*)[names(*dataset*) == "*vecchio*"] <- "*nuovo*"
sono state rinominate le colonne in modo appropriato. Come ultima azione abbiamo assegnato ad ogni attributo il suo tipo.

2.4 Codice

```
# Codice con R tradizionale

dataset <- subset(dataset, select = - x)

names(dataset)[names(dataset) == "cp"] <- "chest_pain"
names(dataset)[names(dataset) == "trestbps"] <- "rest_bp"
names(dataset)[names(dataset) == "chol"] <- "cholesterol"
names(dataset)[names(dataset) == "thalach"] <- "max_hr"
names(dataset)[names(dataset) == "exang"] <- "exercise_angina"
names(dataset)[names(dataset) == "thal"] <- "thalassemia"
names(dataset)[names(dataset) == "target"] <- "heart_disease"
names(dataset)[names(dataset) == "ca"] <- "n_vessels"
names(dataset)[names(dataset) == "restecg"] <- "rest_ecg"

# Codice con Tidyverse

dataset <- dataset %>%
  select(-one_of("x")) %>%
  rename(
    chest_pain = cp,
```

```
rest_bp = trestbps,
cholesterol = chol,
max_hr= thalach,
exercise_angina = exang,
thalessemia = thal,
heart_disease = target,
n_vessels = ca,
rest_ecg = restecg
)
```

---

3 Tipi di dati

3.1 Tipo di ogni attributo

age int ORDINALE
sex chr NOMINALE
chest_pain int NOMINALE
rest_bp int DI RAPPORTO
cholesterol chr DI INTERVALLO
fbs int DI RAPPORTO
rest_ecg int NOMINALE
max_hr int DI INTERVALLO
exercise_angina int NOMINALE
oldpeak num ORDINALE
slope int NOMINALE
n_vessels int ORDINALE
thalassemia int NOMINALE
heart_disease int NOMINALE

3.2 Correzione dei tipi di dati

Successivamente, abbiamo eseguito un controllo per correggere la consistenza del tipo di dato per ogni variabile con la funzione
[***colonna del dataset***] **<- as.** [***tipo nel quale voglio cambiare i dati***] (***colonna del dataset***)
così come segue:
- Abbiamo trasformato, nella colonna sex, i semplici valori "0" e "1" in "F" per femmina e in "M" per maschio e poi abbiamo cambiato il tipo di dato da int a factor (quindi diviso in più livelli) per gli attributi "F" e "M".
- Abbiamo cambiato il tipo di dato per la colonna chest pain da int a factor quindi diviso in più livelli (0 - 1 - 2 - 3).
- Per la colonna cholesterol abbiamo trasformato in primo luogo tutti i valori "undefined" nella mediana dei valori di tutta la mia colonna; in secondo luogo abbiamo trasformato il tipo di dato da char a integer.
- Abbiamo cambiato il tipo di dato per la colonna fbs da int a factor quindi

diviso in più livelli (1 - 0).
- Abbiamo cambiato il tipo di dato per la colonna rest_ecg da int a factor quindi diviso in piu' livelli (0 - 1 - 2).

3.3 Codice

```
# Codice con R tradizionale

levels(dataset$chest_pain)[levels(dataset$chest_pain)== 0 ] <-
    "asymptomatic"
levels(dataset$chest_pain)[levels(dataset$chest_pain)== 1 ] <-
    "nontypical_angina"
levels(dataset$chest_pain)[levels(dataset$chest_pain)== 2 ] <-
    "nonanginal_pain"
levels(dataset$chest_pain)[levels(dataset$chest_pain)== 3 ] <-
    "typical_angina"

levels(dataset$fbs)[levels(dataset$fbs)== 0 ] <- "False"
levels(dataset$fbs)[levels(dataset$fbs)== 1 ] <- "True"

levels(dataset$rest_ecg)[levels(dataset$rest_ecg)== 0 ] <-
    "Ventricular_hypertrophy"
levels(dataset$rest_ecg)[levels(dataset$rest_ecg)== 1 ] <- "Normal"
levels(dataset$rest_ecg)[levels(dataset$rest_ecg)== 2 ] <- "Anomaly"

levels(dataset$exercise_angina)[levels(dataset$exercise_angina)== 0 ] <-
    "No"
levels(dataset$exercise_angina)[levels(dataset$exercise_angina)== 1 ] <-
    "Yes"

 --etc... con via con gli altri attributi

# Codice con Tidyverse

dataset <- dataset %>%
  mutate(
    age <- as.integer(age),
    sex = ifelse(sex == "1", "M", "F"),
    sex = as.factor(sex),
    chest_pain = as.factor(chest_pain),
    cholesterol =ifelse(cholesterol == "undefined",
        median(cholesterol), cholesterol),
    cholesterol = as.integer(cholesterol),
    fbs = as.factor(fbs),
    rest_ecg = as.factor(rest_ecg),
    exercise_angina = as.factor(exercise_angina),
```

```
  slope = as.factor(slope),
  thalassemia = as.factor(thalassemia),
  thalessemia = as.factor(thalessemia),
  heart_disease = as.factor(heart_disease)
)
```

3.4 Livelli dei fattori

Per vedere se le modifiche fossero avvenute con successo abbiamo stampato nuovamente il dataset, abbiamo rinominato i livelli dei fattori per ogni colonna, così da renderli più comprensibili.

```
Livelli per chest pain
0 = "asymptomatic"
1 = "nontypical_angina"
2 = "nonanginal_pain"
3 = "typical_angina"

Livelli per fbs
0 = "False"
1 = "True"


Livelli per rest_ecg
0 = "Ventricular_hypertrophy"
1 = "Normal"
2 = "Anomaly"

Livelli per exercise_angina
0 = "No"
1 = "Yes"

Livelli per slope
0 = "Descending"
1 = "Flat"
2 = "Ascending"

Livelli per thalessimia
0 = "non_existent"
1 = "defect_corrected"
2 = "normal_blood"
3 = "reversible_defect"

Livelli per heart disease
0 = "Yes"
1 = "No"
```

4 Consistenza dei dati

4.1 max_hr

Per far emergere gli outlier e le anomalie della frequenza cardiaca più alta, come prima cosa abbiamo estratto con la funzione

```
hist(dataset$max_hr)
```

un grafico a barre dei valori dei dati forniti in ingresso dal dataset. In ggplot la funzione diventa

```
dataset %>%
   ggplot(aes(x, fill = f)) +
   geom_histogram()
```

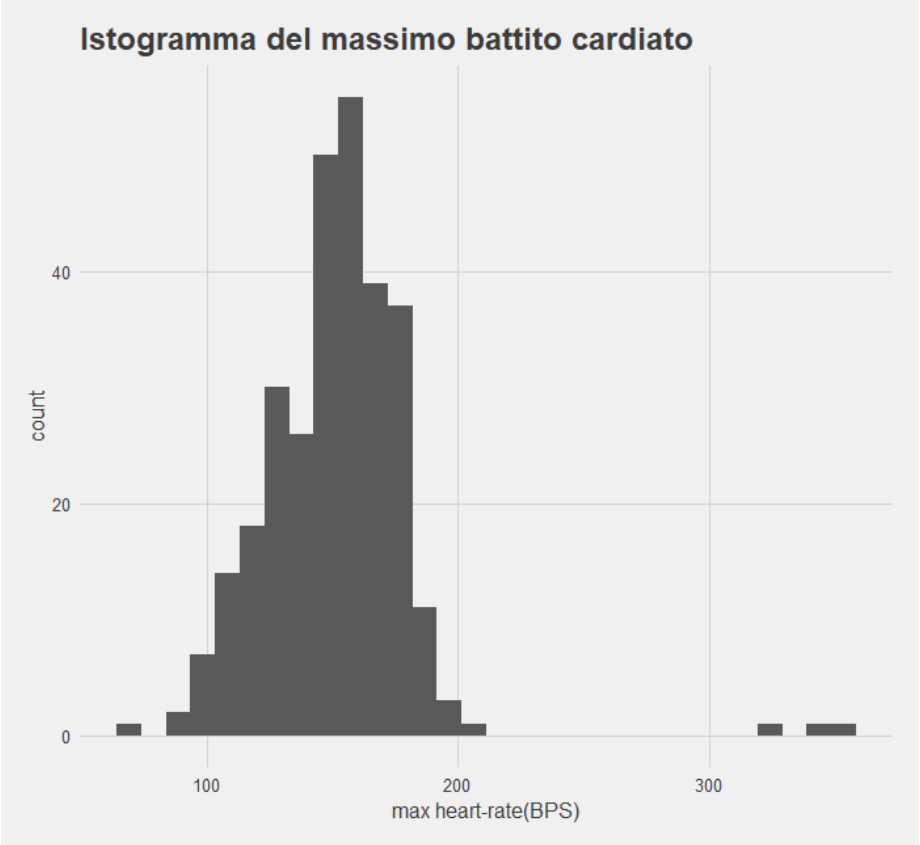

Figure 2: GRAFICO HIST max_hr con ggplot

Abbiamo deciso che il numero maggiore di battiti cardiaci non sia superiore

a 222 e che il numero minore di battiti cardiaci sia il valore medio della variabile.

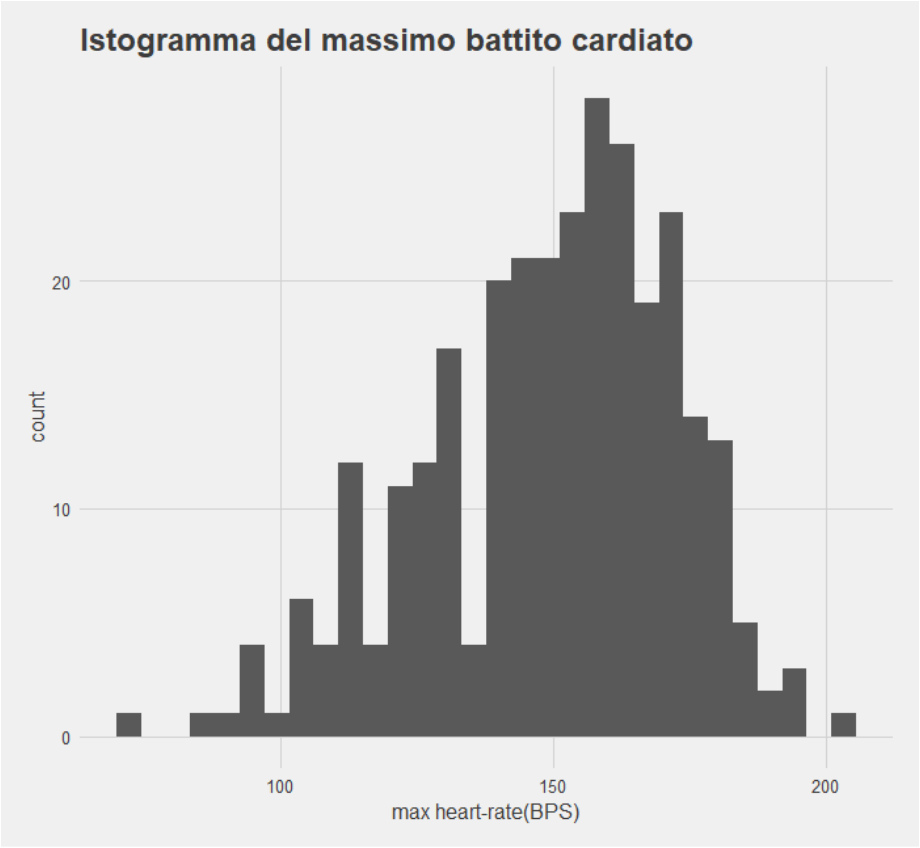

Figure 3: GRAFICO HIST max_hr con ggplot

4.2 rest_bp diviso per sesso

Per far emergere gli outlier e le anomalie della pressione sanguigna a riposo della persona, oltre ad un istogramma, abbiamo utilizzato un altro grafico più esplicativo: il boxplot.

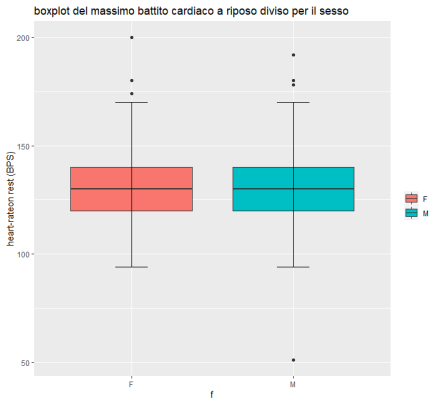

Figure 4: GRAFICO BOXPLOT max_hr con ggplot

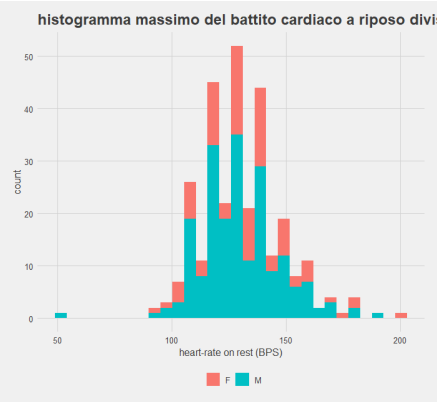

Figure 5: GRAFICO HIST max_hr con ggplot

Abbiamo rappresentato gli outliers relativi alla variabile rest_bp', abbiamo calcolato il terzo e il primo quantile e li abbiamo sostituiti nella formula **IQR<-(Q3-Q1)** per trovare lo scarto interquartile ed infine abbiamo rilevato il range della differenza interquartile per creare il nuovo boxplot.

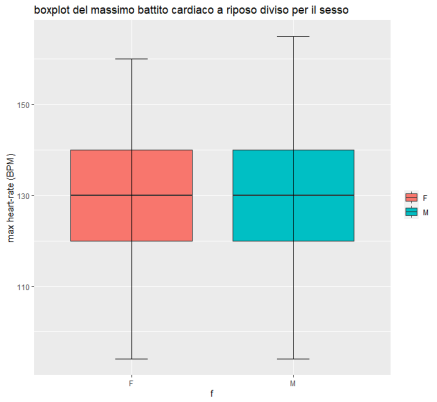

Figure 6: GRAFICO BOXPLOT max_hr con ggplot

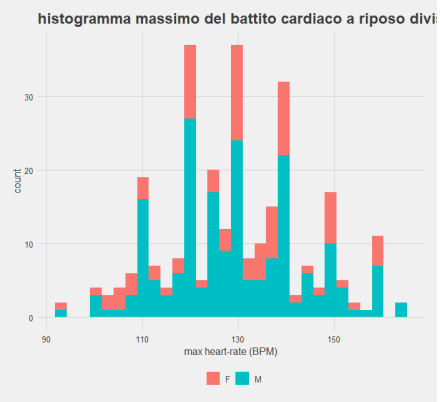

Figure 7: GRAFICO HIST max_hr con ggplot

4.3 Considerazioni finali

Ristampando a console il dataset, aggiornato e modificato, abbiamo notato che è più consistente rispetto alla prima volta che l'avevamo visualizzato; tuttavia ci sono ancora delle modifiche da apportare. Abbiamo quindi impostato sull'attributo age un controllo che non permettesse di inserire valori inferiori a 0 o superiori di 120; mentre sull'attributo rest_bp, abbiamo inserito un range che va da 70 a 150 per la pressione sanguigna a riposo. Al termine di queste modifiche abbiamo ristampato il dataset abbiamo stabilito che non ci fossero più modifiche da apportare poiché, per noi ritenuto consistente.

age	sex	chest_pain	rest_bp	cholesterol	fbs	rest_ecg	max_hr	exercise_angina	oldpeak	slope	n_vessels	thalessemia	heart_disease
63	M	typical_angina	145	233	True	Ventricular_hypertrophy	150.0000	False	2.3	Descending	0	defect_corrected	No
37	M	nonangial_pain	130	250	False	Normal	187.0000	False	3.5	Descending	0	normal_blood	No
41	F	nontypical_angina	130	204	False	Ventricular_hypertrophy	172.0000	False	1.4	Ascending	0	normal_blood	No
56	M	nontypical_angina	120	236	False	Normal	178.0000	False	0.8	Ascending	0	normal_blood	No
57	F	asymptomatic	120	354	False	Normal	163.0000	True	0.6	Ascending	0	normal_blood	No
57	M	asymptomatic	140	192	False	Normal	148.0000	False	0.4	Flat	0	defect_corrected	No
56	F	nontypical_angina	140	294	False	Ventricular_hypertrophy	153.0000	False	1.3	Flat	0	normal_blood	No
44	M	nontypical_angina	120	263	False	Normal	173.0000	False	0.0	Ascending	0	reversible_defect	No
52	M	nonangial_pain	51	199	True	Normal	162.0000	False	0.5	Ascending	0	reversible_defect	No
57	M	nonangial_pain	150	168	False	Normal	174.0000	False	1.6	Ascending	0	normal_blood	No
54	M	asymptomatic	140	239	False	Normal	160.0000	False	1.2	Ascending	0	normal_blood	No
48	F	nonangial_pain	130	275	False	Normal	139.0000	False	0.2	Ascending	0	normal_blood	No
49	M	nontypical_angina	130	266	False	Normal	171.0000	False	0.6	Ascending	0	normal_blood	No
64	M	typical_angina	110	211	False	Ventricular_hypertrophy	144.0000	True	1.8	Flat	0	normal_blood	No
58	F	typical_angina	150	283	True	Ventricular_hypertrophy	162.0000	False	1.0	Ascending	0	normal_blood	No
50	F	nonangial_pain	120	219	False	Normal	158.0000	False	1.6	Flat	0	normal_blood	No
58	F	nonangial_pain	120	340	False	Normal	172.0000	False	0.0	Ascending	0	normal_blood	No
66	F	typical_angina	150	226	False	Normal	114.0000	False	2.6	Descending	0	normal_blood	No
43	M	asymptomatic	150	247	False	Normal	171.0000	False	1.5	Ascending	0	normal_blood	No
69	F	typical_angina	140	239	False	Normal	151.0000	False	1.8	Ascending	2	normal_blood	No
59	M	asymptomatic	135	234	False	Normal	161.0000	False	0.5	Flat	0	reversible_defect	No
44	M	nonangial_pain	130	233	False	Normal	179.0000	True	0.4	Ascending	0	normal_blood	No
42	M	asymptomatic	140	226	False	Normal	178.0000	False	0.0	Ascending	0	normal_blood	No
61	M	nonangial_pain	150	243	True	Normal	137.0000	True	1.0	Flat	0	normal_blood	No
40	M	typical_angina	140	199	False	Normal	178.0000	True	1.4	Ascending	0	reversible_defect	No
71	F	nontypical_angina	160	302	False	Normal	162.0000	False	0.4	Ascending	2	normal_blood	No
59	M	nonangial_pain	150	212	True	Normal	157.0000	False	1.6	Ascending	0	normal_blood	No
51	M	nonangial_pain	110	175	False	Normal	123.0000	False	0.6	Ascending	0	normal_blood	No
65	F	nonangial_pain	140	417	True	Ventricular_hypertrophy	157.0000	False	0.8	Ascending	1	normal_blood	No
53	M	nonangial_pain	130	197	True	Ventricular_hypertrophy	152.0000	False	1.2	Descending	0	normal_blood	No
41	F	nontypical_angina	105	198	False	Normal	168.0000	False	0.0	Ascending	1	normal_blood	No

Figure 8: Sceen dataset heart.csv aggiornato

5 Relazioni tra i dati

5.1 Regressione linare

Procedendo con l'analisi del dataset ci siamo confrontati con la regressione lineare semplice e quindi abbiamo messo in relazione due variabili per vedere, tramite gli appositi grafici di regressione lineare, se ci fosse o meno una correlazione.

5.2 age & rest_bp

Per prima cosa abbiamo scelto i due attributi da mettere in relazione, nel nostro caso l'età(age) e la pressione sanguigna a riposo della persona (rest_pb), e abbiamo stampato a video con la funzione **summary(*dataset*)** il loro contenuti suddivisi in quantili. Abbiamo continuato disegnando la retta di regressione e abbiamo notato, dal grafico, che tra age e rest_bp non c'era un apparente correlazione.

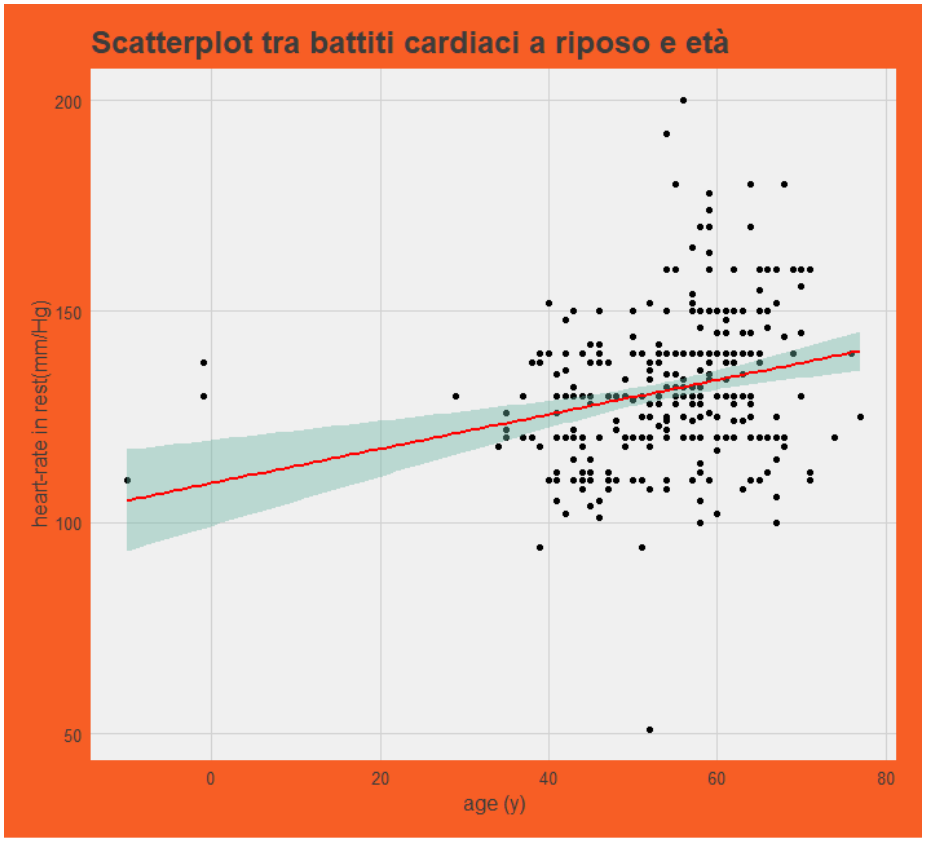

Figure 9: Relazione tra age e rest_bp

5.3 age & max_hr

Per poter continuare con il metodo della regressione abbiamo dovuto, perciò, cambiare attributi. Abbiamo scelto di mettere a confronto gli attributi age e max_hr; infatti, arrivati allo stesso punto di prima, abbiamo visto che c'era una forte correlazione tra i due. Infine, abbiamo potuto rappresentare il tutto tramite una rappresentazione grafica.

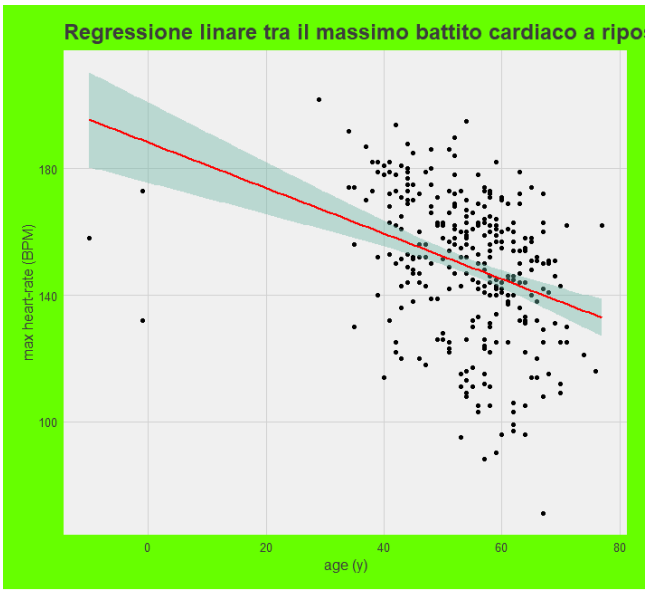

Figure 10: Relazione tra age e max_hr

Continuando la procedura, abbiamo calcolato il coefficiente di correlazione lineare e il coefficiente di determinazione per i due attributi (R^2). Per ultimare la fase di regressione lineare abbiamo analizzato i residui con apposito grafico e, con un altro grafico, abbiamo potuto confrontare la distribuzione in quantili rispetto ad una distribuzione normale standard. E possiamo vedere nella [Figure 12] che i valori sono equidistribuiti intorno alla retta.

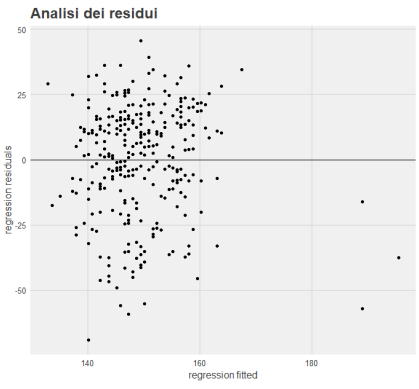

Figure 11: Grafico dei residui

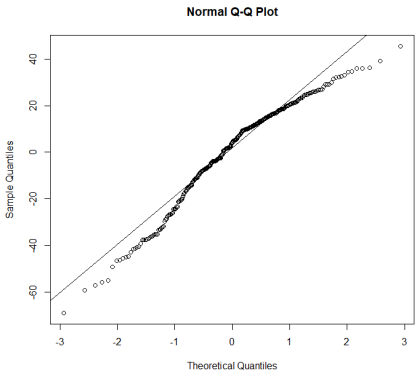

Figure 12: Distribuzione in quantili confrontabile con quella di una normale

6 Modelli di machine learning

6.1 Data frame per previsioni

Come ultima consegna ci è stato chiesto di creare un data frame contente 10 osservazioni, non presenti nel dataset e quindi di effettuare delle previsioni.

age	sex	chest_pain	rest_bp	cholesterol	fbs	rest_ecg	max_hr	exercise_angina	oldpeak	slope	n_vessels	thalassemia	heart_disease
91	1	0	142	201	0	1	130	1	3.703	2	0	3	1
76	1	0	117	192	0	0	90	1	2.731	1	2	2	0
84	0	0	126	245	1	1	140	1	3.380	2	4	0	0
31	0	2	135	246	1	1	148	0	3.008	0	2	3	1
67	1	0	118	245	0	2	179	1	1.443	0	2	0	1
46	0	0	106	165	0	1	173	1	5.764	0	4	2	0
101	0	2	131	153	1	1	116	0	3.262	0	0	3	0
48	1	1	95	243	0	2	200	1	5.307	2	1	2	0
105	1	1	112	132	1	0	71	0	4.335	2	4	2	1
112	0	2	131	230	1	1	104	0	6.069	0	0	2	1
43	1	2	95	178	0	0	100	0	3.996	2	2	2	0
87	0	3	123	206	0	1	104	0	1.962	0	2	2	1
40	0	2	119	146	0	1	85	0	3.830	1	1	3	1
88	0	0	136	221	1	1	89	1	5.694	0	4	3	0
116	0	0	135	211	1	1	150	0	0.577	2	3	0	0
75	0	1	115	130	0	0	73	1	4.188	0	3	1	1
29	0	2	116	162	0	2	185	0	3.274	0	2	2	0
69	1	0	130	136	0	2	130	0	2.716	0	0	3	0
50	0	1	104	139	0	2	189	0	1.810	2	2	2	1
81	1	2	149	171	0	0	82	1	1.103	0	4	0	0

Figure 13: Dataset osservazioni.csv

Tramite python abbiamo creato uno script che ci permettesse di creare 20 osservazioni casuali per il nostro nuovo file “osservazioni.csv”

```python
import random as r
def f():
    age_min = 0
    age_max = 120
    age = r.randint(age_min, age_max)


    sex = r.randint(0, 1)

    cp_min = 0
    cp_max = 3

    cp = r.randint(cp_min, cp_max)

    rbp_min = 94
    rbp_max = 150

    rbp = r.randint(rbp_min, rbp_max)

    chol_min = 126
    chol_max = 250

    chol = r.randint (chol_min, chol_max)

    fbs = r.randint(0, 1)

    recg = r.randint(0, 2)

    mhr = r.randint(71, 202)

    exang = r.randint(0, 1)

    oldpeak = round(r.uniform(0.000, 6.200), 3)

    slope = r.randint(0, 2)

    vessels = r.randint(0, 4)

    thal = r.randint(0, 3)

    hd = r.randint(0,1)

    print(f"{age}, {sex}, {cp}, {rbp}, {chol}, {fbs}, {recg}, {mhr}, {exang}, {oldpeak}, {slope}, {vessels}, {thal}, {hd}")

for i in range(20):
    f()
```

Figure 14: Script python

e grazie alla funzione **predict(*dataset*)** abbiamo potuto predire l’intervallo di confidenza per ogni mio attributo

6.2 Modelli di Machine Learning

Infine, abbiamo applicato tre modelli di machine learning (k-Nearest Neighbors, Multi-Layer Perceptron, Random Forest) per misurare l'accuratezza sul test set e, di conseguenza, è stato creato un **dotplot()** del risultato modelli utilizzati.

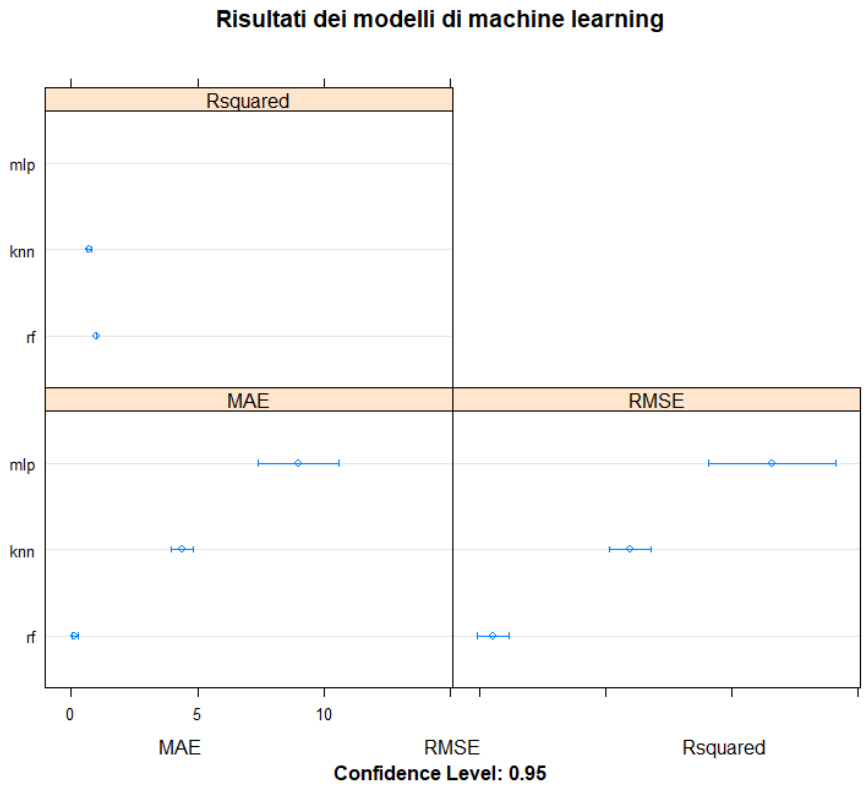

Figure 15: Grafico dei modelli di Machine Learning

6.3 K-Nearest Neighbors (KNN)

Il modello K-Nearest Neighbors (KNN) è uno degli altri algoritmi diffusi nel machine learning. Può essere utilizzato sia per problemi di classificazione che di regressione, anche se è più utilizzato nei primi. La forza di quest'algoritmo è che permette di memorizzare tutte le istanze disponibili e di classificarle valutando la distanza rispetto ai suoi vicini. L'istanza verrà assegnata alla classe che include il data point più vicino all'istanza stessa.

6.4 Perceptron Multistrato (MLP)

Il modello perceptron multistrato (MLP) è una rete neurale artificiale feedforward che genera un insieme di output da un insieme di input. Un MLP è caratterizzato da diversi livelli di nodi di input collegati come un grafico diretto tra i livelli di input e output. MLP utilizza la backpropogation per addestrare la rete. MLP è un metodo di apprendimento profondo.

6.5 Random Forest (RF)

Il modello Random Forest (RF)è un metodo versatile di machine learning, capace di affrontare sia compiti di classificazione che di regressione. Con le foreste casuali è anche possibile applicare metodi per la riduzione della dimensionalità, gestire dati mancanti, valori degli outlier ed altri passaggi essenziali di esplorazione dei dati, producendo buoni risultati.

7 Extra

Per facilitare la stesura del codice abbiamo utilizzato le librerie:
library(tidyverse)
library(caret)
library(ggthemes)
Per avere una condivisione più efficiente di tutti i file è stato usato GitHub.
Link di GitHub: `https://github.com/r-vil/heart_in_r`