

Esercizio 0 - HoloTC

per il modulo SM3 della SSSMT di Locarno

Dr Giorgio Pioda

Ultima compilazione il 25 agosto, 2023

This work is licensed under a [Creative Commons “Attribution-NonCommercial-ShareAlike 4.0 International”](https://creativecommons.org/licenses/by-nc-sa/4.0/) license.



Contents

1 HoloTC

2

1 HoloTC

Si svolga un'analisi di statistica descrittiva dei dati di 51 pazienti a cui è stata misurata la [holotranscobal-amina](#)¹. In particolare si proceda con le seguenti operazioni

a) Importazione dei dati e studio preliminare.

- usare l'apposito menu o in alternativa,
- `holoTC <- read.table("holoTC.csv", header=T, ...)` # oppure anche
- `holoTC <- read.csv("holoTC.csv", header=T)` # oppure anche direttamente dal web
- `holoTC <- read.csv("http://web.ticino.com/gfwp/stat/dataset/holoTC.csv", header=T)`
- `str(holoTC)` # Struttura dei dati
- `head(holoTC)` # Prime 6 righe
- `summary(holoTC)` # Sommario automatico
- `boxplot(..., horizontal=T, ...)`

b) Tracciare l'istogramma (saranno necessarie classi non omogenee).

- Sinossi funzione `seq(minimo, massimo, by=spaziatura)`
- `holobreaks <- c(seq(4.5, 40, by=5), seq(49.5, ..., by=10), seq(119.5, ..., ...))`
- `holobreaks` # Stampa i breaks
- `holodata <- hist(holoTC$holoTC, breaks=holobreaks, ...)` # Traccia e memorizza i dati dell'istogramma
- `holo.mean <- mean(...)`
- `holo.sd <- sd(...)`
- `curve(dnorm(x, mean=holo.mean, sd=holo.sd), ..., add=T, yaxt="n")` # aggiustare le opzioni `col`, `lwd`, e `lty`

c) Tracciare l'ogiva e stampare la tabella delle frequenze. Tracciare un ecdf da confrontare con l'ogiva

- `library(agricolae)` # chiaramente la libreria deve essere installata
- `tabella <- table.freq(holodata)` # memorizza la tabella in `tabella`
- `tabella` # Visualizza la tabella
- `t.o <- ogive.freq(...)` # Visualizza l'ogiva
- `plot.ecdf(...)`

d) Tracciare il kernel density e ecdf (empirical cumulative density function). Determinare moda, media e mediana e aggiungerle al grafico. Si guardi la precedente attività.

- `holo.d <- density(...)` # Calcola una kernel density
- `plot(holo.d, main="Titolo", xlab="unità misura")` # Ottimizzare titoli ed etichette
- `holo.max.i <- which.max(holo.d$y)` # Trova il massimo (posizione nei dati).
- `holo.d$х[holo.max.i]` # Stampa il valore x del massimo
- `abline(v=holo.d$х[holo.max.i], col="red", lty=2)` # Disegna il massimo sul grafico
- `abline(...)` # Stampare anche media e mediana, usare altri `col` e `lty`
- `legend("topright", legend=c("moda", "mediana", "media"), col=c(...), lty=c(...))`

e) Calcolo dei percentili classici, di quelli sulla kernel density e di quelli teorici normali.

- `quant.seq <- c(seq(0, 1, by=0.05))`
- `quantile(..., quant.seq)`
- `library(spatstat)` # libreria per il calcolo con la k. den.
- `quantile(holo.d, ...)`
- `qnorm(quant.seq, mean=holo.mean, sd=holo.sd)` # Trova il valore dall'area, funzione inversa

¹Dati lavoro di diploma SSMT (2014) di Michela Belvedere. Per gentile concessione.

f) Q-Q Plot con verifica della normalità.

- `library(car)`
- `t.q <- qqPlot(...)`

g) Q-Q plot manuale con simulazione sovrapposta (vedi funzione qui sotto)

```
sim.qq <- function(dlen,centro,disper,colore="lightblue"){
  if (is.numeric(dlen)) {}
  else {dlen <- length(dlen)}
  for (i in 1:20) {
    lines(qnorm(ppoints(dlen)),sort(rnorm(dlen,mean=centro,sd=disper)),col=colore)
  }
}
```

- `source("http://web.ticino.com/gfwp/stat/snippet/simqq.R")`
- `qqnorm(holoTC$HoloTC)` # Grafico iniziale oppure in alternativa
- `plot(qnorm(ppoints(holoTC$HoloTC)),sort(holoTC$HoloTC),ylab="Sample",xlab="Theoretical Quantiles", main="Q-Q plot")`
- `sim.qq(holoTC$HoloTC,centro=holo.mean,disper=holo.sd)` # Simulazione media e sd standard
- `qqline(holoTC$HoloTC)` # retta ideale robusta
- `points(qnorm(ppoints(holoTC$HoloTC)),sort(holoTC$HoloTC))` # Sovrascrive i punti
- Nota: `ppoints()` calcola il quantile empirico (tramite rango) di ciascun valore, `qnorm()` trova la posizione sulla x corrispondente al valore del quantile empirico.

h) determinare sulla ipotetica curva di Gauss:

- 1) la probabilità di avere valori più piccoli di 20 e più grandi di 100;
- 2) il valore che tiene a destra il 15% dei dati;
- 3) il valore che tiene alla sinistra il 5% dei dati.

- `pnorm(...)` # Calcola l'integrale a partire da $-\infty$
- `qnorm(...)` # Funzione inversa. Trova il valore che delimita una certa area da $-\infty$.
- Esempio: probabilità di valori più piccoli di 20: `pnorm(20,mean=holo.mean,sd=holo.sd)`

i) Calcolare l'indice di MAD "Median of Absolute Deviances" con la funzione `mad()` e aggiungere una nuova curva di gauss all'istogramma iniziale usando questo indicatore di dispersione e la mediana come indice di centralità

- ... (istogramma come nel punto b) # Poi si aggiungono
- `holo.mad <- mad(...)`
- `holo.median <- median(...)`
- `curve(dnorm(x,mean=...,sd=...),...,add=T,yaxt="n")` # aggiustare le opzioni `col`, `lwd`, e `lty`