Versione documento: 22 febbraio 2022, h16:25

Università degli Studi di Salerno
Laurea in Statistica per i Big Data (L41)

ANALISI E VISUALIZZAZIONE DEI DATI (PT I)

Pietro Coretto

ESERCIZI

Indicazioni generali

- Questi esercizi sono essenziali e complementari allo svolgimento delle lezioni in aula/laboratorio. Lo studente che saprà risolvere questi esercizi in autonomia non avrà nessun problema negli esami di valutazione.
- In generale non ci sarà una sistematica verifica/correzione di questi esercizi. È interesse dello studente accertarsi che la sua preparazione sia adeguata al livello di difficoltà proposto in questo documento. Tuttavia, alcuni di questi problemi possono essere usati durante le lezioni come casi didattici.
- È innegabile che alcuni di questi esercizi sono ispirati a libri e/o risorse in rete. Prima di cercare l'aiuto del docente, oppure quello di libri e *web*, è importante aver fatto ogni possibile sforzo per produrre una soluzione ragionevole.
- Lo schema degli esercizi segue il syllabus del corso, pertanto i problemi presentati vanno risolti nell'ordine proposto.
- La notazione segue quella delle lezioni in assenza di altre indicazioni.
- Con R si indica l'ambiente di sviluppo "R". Con `foo()` si intente una funzione di R chiamata foo.
- Si consiglia di svolgere ogni esercizio in un file separato cercando di seguire un qualche ordine/convenzione nell'organizzazione dei files. Ordine e sistematica conformità ad uno standard, sono requisiti essenziali per essere produttivi. Esempio: si crea *sistematicamente* il file `ex_x_y.R` per risolvere l'"Esercizio x.y".
- Gli esercizi contrassegnati con (*) saranno svolti in classe da uno studente scelto generalmente a caso.
- "*Errare humanum est, perseverare autem diabolicum*": errori, omissioni ed ambiguità vanno segnalati a e-mail: pcoretto@unisa.it.

Indice

1 R: strutture dati e loro manipolazioni elementari ed operazioni

Esercizio 1.1 (*) Creare in R il vettore `x` contenente i valori (nell'ordine)

{2, 7, 21, 54, -100, 0, 0, 1}

Risolvere i seguenti punti.

1. Calcolare la somma degli elementi del vettore `x`.
2. Calcolare la somma dei quadrati degli elementi di `x`.
3. Trovare gli indici degli elementi di `x` che sono negativi.
4. Trovare gli indici degli elementi di `x` che sono positivi.
5. Creare il vettore `x2` ottenuto eliminando i primi tre elementi di `x`.
6. Creare il vettore `x3` formato dagli elementi positivi di `x`.
7. Contare gli elementi di `x` che sono maggiori 20 in valore assoluto.
8. Determinare la posizione dell'elemento massimo di `x`.
9. Determinare la posizione dell'elemento minimo di `x`.
10. Calcolare il massimo valore contenuto in `x`.
11. Calcolare il minimo valore contenuto in `x`.

* * *

Esercizio 1.2 Considera le seguenti matrici

$$\boldsymbol{X} = \begin{pmatrix} 6 & 4 & 19 & 20 \\ 8 & 14 & 1 & 10 \\ 11 & 15 & 3 & 5 \\ 16 & 9 & 2 & 7 \end{pmatrix}, \qquad \boldsymbol{Y} = \begin{pmatrix} 5 & 10 & 6 \\ 9 & -7 & -6 \\ -3 & -1 & -10 \\ 3 & -9 & 2 \end{pmatrix}. \tag{1.2.1}$$

1. In R costruisci le due matrici assegnandole ad `X` e `Y`.
2. Selezionare l'elemento di posizione $(1,2)$ nella matrice `X`, e l'elemento di posizione $(4,2)$ nella matrice `Y`.
3. Selezionare la prima colonna di `X` e la seconda riga di `Y`.
4. Calcolare le matrici: (a) `XX` contente i quadrati degli elementi di `X`; (b) `logX` contente i logaritmi naturali degli elementi di `X`; (c) `sqrtX` contente la radice quadrata degli elementi di `X`.
5. Calcolare: (a) il prodotto matriciale $\boldsymbol{XY}$; (b) la matrice $\boldsymbol{X}^\top\boldsymbol{X}$; (c) $\det(\boldsymbol{X})$ e $\boldsymbol{X}^{-1}$; (d) $\det(\boldsymbol{X}^\top\boldsymbol{X})$ e $(\boldsymbol{X}^\top\boldsymbol{X})^{-1}$.
6. Dalla matrice $\boldsymbol{Y}$ estrarre la sotto-matrice quadrata ottenuta eliminando l'ultima riga.

7. Estrarre la diagonale principale della matrice $\boldsymbol{X}$.

8. Sostituire la diagonale principale della matrice $\boldsymbol{X}$ con un il vettore nullo $(0, 0, 0, 0)^\mathsf{T}$.

9. Nella matrice $\boldsymbol{Y}$ sostituire tutti i numeri negativi con 0.

Esercizio 1.3 (*) Il giorno 25/12/2018 il team della "DATABOH Srl" si reca presso il campus di UniSA per effettuare una rilevazione sullo stato delle carriere accademiche. In un campione scelto a caso si rilevano i seguenti dati.

Id	**Sex**	**Ofa**	**Mes**	**Nes**
AX12	M	0	24.3	4
BZ55	F	1	na	na
CW32	F	0	28.5	1
DQ55	M	1	na	0

Dove:

Id: identificativo univoco dello studente;

Sex: =M se maschio; =F se femmina;

Ofa: =1 se lo studente è stato immatricolato con debiti formativi; =0 altrimenti;

Mes: media degli esami superati;

Nes: numero di esami superati.

1. Chi sono le unità statistiche? Quali e quante sono le variabili rilevate?

2. Classifica le variabili rispetto al loro *espressione*.

3. Classifica questo data set rispetto al *ruolo delle unità statistiche*.

4. In R costruisci un `data.frame()` chiamato `stud` contenente il data set. Usa la risposta al punto precedente ed assicurati che le variabili siano adeguatamente formattate (tipo numerico, metadati, etc.).

5. Rispetto alla variabile **Nes**, estrai gli indici delle osservazioni (anche detti *sample points*) per le quali è stato registrato un valore mancante (na).

6. Rispetto alla variabile **Mes**, estrai gli indici delle osservazioni che non presentano una risposta mancante.

7. Estrai i codici identificativi (**Id**) degli studenti che non hanno risposto alla domanda sulla media degli esami superati.

* * *

Esercizio 1.4 La rappresentazione digitale *bitmap* (anche detta *raster*) di un immagine avviene solitamente dividendo l'area dell'immagine in una *tabella* di piccoli quadrati/rettangoli chiamati *pixels*. Un'immagine bitmap $w \times h$ divide l'area da rappresentare in una griglia formata da w pixels orizontali (width) e h pixels verticali (height).

Vi sono vari sistemi per codificare i colori, il più comune è la codifica RGB. In pratica il colore di ogni pixels viene rappresentato dalla combinazione delle espressioni dei canali R=rosso, V=verde e B=blue. Il colore espresso in ognuno dei tre canali viene rappresentato con 8-bit, ovvero un numero intero tra 0 e 255 (256 livelli possibili). Poiché il colore di ogni pixel è formato dalla mescolanza dei canali R, G, e B, ogni pixel richiede 24-bit = 3-bytes di informazione (8-bit $\times$ 3 canali = 24-bit = 3-bytes).

Supponiamo di avere una macchina fotografica con una risoluzione di $2 \times 2 = 4$ pixels. Ad ogni scatto la macchina dovrà misurare (campinare) i canali R,G e B in una griglia di 2 pixels orizzontali $\times$ 2 pixels verticali. Supponiamo di scattere una foto, e rappresentiamo i pixels campionati nel seguente array 3-dimensionale che chiamiamo `U`

```
> U
, , 1

     [,1] [,2]
[1,]   67  146
[2,]   95  232

, , 2

     [,1] [,2]
[1,]   51  241
[2,]  229  169

, , 3

     [,1] [,2]
[1,]  161   52
[2,]   15   45
```

I pixel sono organizzati in una griglia fatta di righe e colonne. La numerazione delle righe avviene dall'alto verso il basso, mentre la numerazione delle colonne è da sinistra verso destra. Quindi il pixel di coordinata $(i, j) = (1, 1)$ è il primo pixel in alto a sinistra. Dato il pixel di riga i e colonna j, `U[i,j,1]` è l'espressione del suo canale R=red, `U[i,j,2]` è l'espressione del suo canale G=green, `U[i,j,3]` è l'espressione del suo canale B=blue.

1. Costruire l'array `U` in R. Assicurarsi che i livelli di espressione dei pixels siano di tipo intero.
2. Estrarre i tre canali separatamente assegnandoli a tre arrays di dimensione 2×2 chiamati `Rchannel`, `Gchannel` e `Bchannel`.
3. Converti `Rchannel`, `Gchannel` e `Bchannel` in matrici.
4. Ottieni un vettore chiamato `u12` contenente i valori dei canali R, G, e B per il pixel che si trova sulla prima riga/seconda colonna.
5. Da `U` elimina tutti i valori dei pixels della prima riga
6. Usando opportunamente la funzione `dimnames(U)` ottieni la sequente formattazione dell'oggetto `U`

```
> U
, , Rchannel

      Col1 Col2
Row1    67  146
Row2    95  232

, , Gchannel

      Col1 Col2
Row1    51  241
Row2   229  169

, , Bchannel

      Col1 Col2
Row1   161   52
Row2    15   45
```

7. La rappresentazione bitmap/raster delle immagini e dei colori è un più sofisticata di come è stata presentata, tuttavia la logica di base è ben rappresentata da questo esempio. Sei curioso di vedere la rappresentazione a schermo della tua foto da 4 pixels di risoluzione? La foto è mostrata in Figura 1.

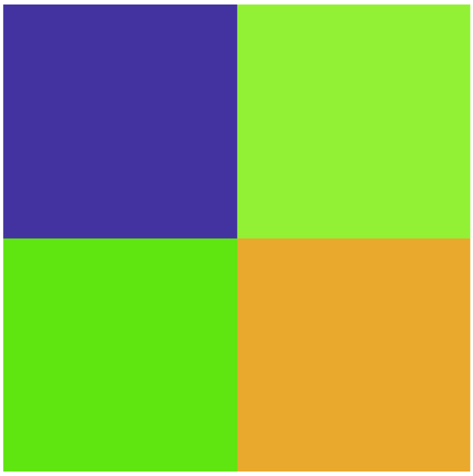

Figura 1: Rappresentazione dei pixels campionati nell'Esercizio 1.4.

Adesso prova questo semplice esperimento: accedi a `http://www.google.com` ed inserisci nel campo di ricerca la chiave `rgb(146, 241 ,52)`. Confronta il risultato della ricerca con il punto 4 dell'esercizio, e con la Figura 1. Cosa noti?

* * *

Esercizio 1.5 Considera la matrice $\boldsymbol{X}$ dell'Esercizio 1.2, in aggiunta considera i vettori

$$\boldsymbol{y} = \begin{pmatrix} 98 \\ 35 \\ 36 \\ 18 \end{pmatrix}, \qquad \boldsymbol{e} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

Sia $\boldsymbol{z}$ un vettore dell'insieme $\mathbb{R}^4$, si consideri il sistema di equazioni lineari (in forma matriciale)

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{z} + \boldsymbol{e},$$

dove $\boldsymbol{X}$ è la matrice dei coefficienti, e $\boldsymbol{z}$ è l'incognita.

1. Usare le condizioni del teorema di Rouché-Capelli per controllare l'esistenza e l'unicità della soluzione del sistema. Si noti che in R il rango di una matrice `A` si ottiene con `qr(A)$rank`.
2. Controllare che la matrice dei coefficienti $\boldsymbol{X}$ sia invertibile.
3. Calcolare la soluzione del sistema.

* * *

Esercizio 1.6 (*) Cos truire la seguente lista chiamata `LaPrimaLista`:

```
> LaPrimaLista
$id
[1] "AA A"  "BBB"  "CCC"   "AAA"   "AAA"   "CCC"   "BBB"   "ZZZ"

$x
[1] -0.6  0.2 -0.8  1.6  0.3 -0.8  0.5

$mat
     [,1] [,2] [,3] [,4]
[1,]   16    7    8    1
[2,]   10   12    2   19
[3,]   13   14   18    6
[4,]   17    3    4   11
```

dove l'elemento `LaPrimaLista$mat` è un 2D-array. Nota che nella prima posizione del primo elemento della lista va inserito intenzionalmente ''`AA A`'' e non ''`AAA`'' (vedi ultimi due punti sotto).

1. Estrai dalla lista il suo primo elemento usando prima `$` e poi il selettore `[[ ]]`.
2. Estrai il terzo elemento della lista eliminando l'ultima riga e la seconda colonna .
3. Nel primo elemento della lista sostituisci ''`ZZZ`'' con ''`AAA`''.
4. Nel primo elemento della lista conta quante volte compare ''`AAA`''.
5. Nel primo elemento della lista elimina tutti gli spazi tra i caratteri.
6. Adesso conta nuovamente quante volte compare ''`AAA`''.

2 R: data input/output

Esercizio 2.1 Considera il data set:

- `https://pietro-coretto.github.io/datasets/crime2005/readme.txt`
- `https://pietro-coretto.github.io/datasets/crime2005/crime2005.csv`

1. Scarica il data set in una directory locale del tuo computer. Leggi la documentazione.
2. Il data set è un file a codifica testuale con formato "*comma-separated-value*". Leggi il data set in R importandolo in un `data.frame`
3. Verifica la codifica delle variabili. Formatta adeguatamente il tipo numerico di tutte le variabili in funzione della loro natura statistica. Presta particolare attenzione alle variabili di tipo non numerico.
4. Esporta il tuo `data.frame` opportunamente formattato in un file `.RData`. Apri una nuova sessione di R e prova a leggere il file appena creato.
5. Esporta il dati in un file a codifica testuale con formato "*tab-separated-values*". Salva il file con estensione `.tsv`. Apri una nuova sessione di R e prova a leggere il file appena creato.
6. Conta quanti stati hanno un tasso criminale (MU) contenuto tra 5 e 10.
7. Estrai la lista degli stati in cui il tasso criminale è contenuto tra 5 e 10.
8. Conta in quanti stati la percentuale di "white people" (WH) è inferiore al 70%.
9. Estrai la lista degli stati con percentuale di "white people" inferiore al 70%.
10. Conta quanti stati hanno un tasso criminale contenuto in [5 , 10] e percentuale di "white people" inferiore al 70%.
11. Estrai la lista degli stati che hanno un tasso criminale contenuto in [5 , 10] e percentuale di "white people" inferiore al 70%.

* * *

Esercizio 2.2 (*) Considera il data set:

- `https://pietro-coretto.github.io/datasets/apple/readme.txt`
- `https://pietro-coretto.github.io/datasets/apple/apple.dat`

1. Scarica il data set in una directory locale del tuo computer. Leggi la documentazione.
2. Il file contiene un `data.frame` chiamato `dat`. Ispeziona la struttura del `data.frame` ed accertati che tutte le variabili siano formattate correttamente.

3. Esporta il data set in un file a codifica testuale con formato "*comma-separated-value*". Per verificare la corretta scrittura dei dati importali nuovamente in un `data.frame` chiamato `dat2`.

4. Esporta il data set in un file a codifica testuale con formato "*tab-separated-value*". Per verificare la corretta scrittura dei dati importali nuovamente in un `data.frame` chiamato `dat3`.

5. Conta il numero di righe (imprese produttrici di mele). Conta il numero di variabili/features misurate.

6. Verifica la presenza di `NA` nei dati. Fai l'attachment del `data.frame`.

7. Estrai gli indici delle imprese (righe del data set) per le quali la quantità di mele prodotte (`qApples`) è maggiore o uguale a 3.

8. Conta il numero di imprese che hanno un costo del lavoro (`vLab`) compreso tra 50000 e 100000. Quale è la percentuale di queste imprese?

9. Seleziona i costi del capitale (`vCap`) per tutte le imprese la cui quantità prodotta (`qApples`) è maggiore o uguale a 3.

10. Calcola la percentuale di imprese che hanno un costo del capitale tra 50000 e 100000, ed un costo del lavoro tra 140000 e 260000.

11. La funzione `download.file()` consente di scaricare file remoti. Prova il seguente esempio.

```r
## Verifichiamo la directory di lavoro
getwd()

## Verifichiamo il contenuto della directory di lavoro
dir()


## Scarichiamo il file remoto, salvandolo come file locale chiamato
## "xxx.Rdata"
download.file(url="http://www.decg.it/pcoretto/datasets/apple_data.RData",
              dest = "xxx.RData")

## Verifichiamo che il file "xxx.RData" sia stato salvato correttamente
## nella directory di lavoro
dir()
```

3 R: uso elementare del device grafico

Esercizio 3.1 Costruire il vettore

```r
v <- c(9, 1 , -1,8 , 10 , 14,  17 , -5,  -10, 12)
```

1. Visualizza i valori di v contro l'indice della loro posizione nel vettore. Usa la funzione `plot()` con t=''p''.

2. Ripeti il punto precedente facendo variare i parametri `pch, col, main, xalb, ylab, xlim, ylim`.
3. Calcola il vettore `u = v**2`. Rappresenta `v` sull'asse orizzontale e `u` sull'asse orizzontale usando `t = ''p''`. Fai qualche esperimento variando i parametri `cex, pch, col`.
4. Ripeti il punto precedente. Questa volta ogni punto deve essere rappresentato dal simbolo `''O''` usando un colore dove il canale rosso è espresso all'85.8%, il canale verde è espresso al 24.7%, ed il canale blue è espresso al 19.7% (questo è esattamente il rosso della "O" e della "e" di Google!)
5. Ripeti il punto precedente con `t = ''l''` producendo una linea di spessore 3pt. Ha senso questo tipo grafico? Individua quale è la causa della strana visualizzazione ottenuta.

* * *

Esercizio 3.2 (*) Si consideri la funzione $y = \sqrt{x}$

1. Genera una sequenza di valori per x nell'intervallo $[0, 10]$. Per i valori generati calcola y in in un vettore `y`.
2. Produci il grafico della funzione limitando entrambi gli assi (verticale e orizzontale) all'intervallo [0,1]. Assicurati che l'aspect ratio del device grafico rispecchi la scala delle due variabili. Nomina opportunamente gli assi e ed il titolo del grafico.
3. Aggiungi al grafico una retta orizzontale tratteggiata di colore grigio passante per il punto $y = 0.5$. Inoltre, aggiungi una retta verticale di colore grigio passante per il punto $x = 0.5$.
4. Aggiungi al grafico una retta di colore rosso di equazione $y = x$ (cosiddetta retta a 45 gradi).
5. Aggiungi sullo stesso device il grafico della funzioni $y = x^2$.
6. Esporta il grafico in formato bitmap/raster. Ottimizza il file ottenuto per la visualizzazione su display elettronico.
7. Esporta il grafico in formato bitmap/raster. Ottimizza il file per la stampa.
8. Esporta il grafico in formato vettoriale. Ottimizza il file per la stampa.

4 R: strutture di controllo e programmazione

Esercizio 4.1 Si consideri la funzione

$$f(n) = \sum_{i=1}^{n} \exp(i),$$

dove i e n sono i numeri interi, e *exp* la funzione esponenziale, ovvero $\exp(i) = e^i$ Risolvi i seguenti problemi senza usare funzioni (come `sum()`, `length()`, etc.)

1. Calcola $f(10)$ usando un loop `while()`.
2. Calcola $f(100)$ usando un loop `for()`.
3. Ripeti i due punti precedenti per $n = 1000$. Dai una spiegazione per lo strano risultato ottenuto.
4. Ripeti i due punti precedenti aggiungendo nel loop un controllo tale che, se nel corso del loop $f(n) \geq$ `.Machine$double.xmax`, l'iterazione si arresta e viene visualizzato il seguente messaggio di errore:

```
Spiacente:
  f(1000) assume un valore troppo grande per essere rappresentato su
  questo computer
```

* * *

Esercizio 4.2 (*) Considera il data set:

- https://pietro-coretto.github.io/datasets/balancesheet/readme.txt
- https://pietro-coretto.github.io/datasets/balancesheet/balancesheet.RData

Il file `RData` contiene un `data.frame` chiamato `dat`. I seguenti punti possono essere svolti usando funzioni di alto livello. Tuttavia, per scopi didattici in questo esercizio dovrai lavorare con le le sole strutture di controllo iterativo (for/while). Le iterazioni `for` e `while` sono talvolta interscambiabili, scegli tu lo strumento più opportuno se questo non espressamente indicato nella domanda.

1. Importa il data set in una sessione di R. Esplora la struttura dei dati.
2. Con un opportuno loop costruisci il vettore `NumCol`, dove `NumCol[j]` =1 se la la j-esima colonna del data set è di tipo *numeric*, `NumCol[j]` =0 altrimenti. Assegna ad ogni elemento di `NumCol` il nome della corrispondente colonna di `dat`.
3. Con un opportuno loop `for` ottieni il vettore chiamato `SumQuadCol`, dove `SumQuadCol[j]` = somma dei quadrati dei valori contenuti nella colonna j-esima di `dat` se questa contiene un tipo *numeric*, `SumQuadCol[j]=NA` altrimenti.
4. Ogni unità statistica (riga) è un'impresa del settore alimentare. Le imprese campionate sono $n = 5000$. Considera la variabile `roe`, questo è un importantissimo indice di bilancio che misura la redditività del capitale proprio (vedi https://it.wikipedia.org/wiki/Return_on_equity). Costruisci la matrice `RoeDiff`, dove `RoeDiff[h,k]` contiene la differenza in valore assoluto tra il `roe` dell'impresa h e quello dell'impresa k, con $h, k = 1, 2, \ldots, 5000$. Ovvero `RoeDiff[h,k] = abs(dat$roe[h] - dat$roe[k])`. (Attenzione: nota che in `roe$dat` sono presenti molti `NA, NaN`)
5. Controlla se la matrice `RoeDiff` è simmetrica.
6. `RoeDiff[h,k]` può essere considerata una misura della *diversità* nella redditività di due imprese. Infatti, `RoeDiff[h,k]` =0 se le imprese h e k hanno lo stesso ROE, `RoeDiff[h,k]` sarà tanto più grande quanto più ampia è la differenza tra i ROE delle due imprese. Usa un doppio loop `for` su righe e colonne di `RoeDiff` per trovare la coppia di imprese più dissimili.

7. Confronta il risultato precedente con l'output dei seguenti comandi

```r
which.max(RoeDiff)

max(RoeDiff)

which(RoeDiff == max(RoeDiff , na.rm = TRUE), arr.ind = TRUE)
```

* * *

Esercizio 4.3 Si consideri la seguente funzione

$$g(x) = \ln\left(\frac{x}{1-x}\right) \qquad \text{per} \qquad x \in (0, 1). \tag{4.3.1}$$

La funzione $g(x)$ prende il nome di funzione logit, uno strumento molto importante nella modellistica.

1. Genera la griglia `x <- seq(0.001, 0.999, length=100)`, calcola la funzione $g(x)$, e produci il grafico.
2. Programma la funzione `logitfunc`. La funzione prende in input il valore di x e restituisce $g(x)$.
3. Verifica che la tua funzione è in grado di funzionare in modo *vettoriale*. Verifica il seguente comando

```r
a <- c(0.001, 0.5, 0.99)
logitfunc(x = a)
```

4. Ripeti il punto 1 usando la funzione sviluppata nel punto 2.
5. Calcola `logitfunc(0)`, `logitfunc(-1)`. Ottieni un errore? Se la risposta è si vai al punto successivo, altrimenti vai direttamente al punto 7.
6. Nota che il dominio di $g(x)$ è ristretto all'intervallo aperto $(0, 1)$. Una funzione ben programmata dovrebbe prevedere i casi più comuni di utilizzo improprio, e dovrebbe indicare chiaramente la causa dell'errore. Ripeti il punto 2 inserendo opportuni controlli su x. Ad esempio, all'interno della funzione potresti usare `stop()` oppure `break` nel caso di valori x non ammessi. Correda la tua funzione con un opportuno messaggio di errore (ad esempio: "*Non sono ammessi valori x<=0 oppure x>=1*").
7. Ora che la tua funzione `logitfunc` è ben programmata, prova il seguente comando

```r
curve(logitfunc)
```

Prova anche i seguenti comandi eseguendoli uno alla volta

```r
curve(sin)
curve(sin(x), xlim = c(-2*pi , 2*pi))
curve(x^2 + x +1, xlim=c(-10,10))
```

* * *

Esercizio 4.4 (*) Si consideri la seguente funzione

$$H(x;\delta,\lambda) = \begin{cases} \frac{(x+\delta)^\lambda - 1}{\lambda} & \text{se} \quad \lambda \neq 0 \\ \ln(x+\delta) & \text{se} \quad \lambda = 0 \end{cases} \qquad \text{dove} \quad x > -\delta. \tag{4.4.1}$$

La funzione H prende il nome di "*trasformata di Box-Cox*". Anche questa funzione trova largo utilizzo nell'ambito della metodologia statistica. Nota che H è funzione di x, mentre δ e λ sono due parametri che controllando le sue proprietà. Solitamente fissati i parametri (δ, λ), calcoliamo la funzione su x. Nota inoltre che il dominio della funzione è ristretto ai valori di $x > -\delta$.

1. Fissa $\lambda = 0$ e $\delta = 0$. Calcola $H(10; 0, 0)$

2. Fissa $\lambda = 2$ e $\delta = 0$. Genera una griglia di valori `x <- seq(1, 10, length=100)`. Calcola il valore di $H(x; 0, 2)$ e visualizza il grafico.

3. Programma la funzione `bc(x, delta=0, lambda=0)`. I parametri `delta` e `lambda` hanno i valori di default: $\delta = 0$ e $\lambda = 0$. La funzione restituisce $H(x;\delta,\lambda)$. Nota che il dominio di x soddisfa la restrizione $x > -\delta$. Controlla opportunamente la correttezza degli input della funzione come hai fatto al punto 6 dell'Esercizio 4.3.

4. Verifica il funzionamento della tua funzione con i seguenti comandi

```
dati <- c(-2.1, 1, 10, 100)
bc(x = dati, delta = 3, lambda = 0  )
bc(x = dati, delta = 3, lambda = 1.5)
bc(x = dati, delta = 2, lambda = 2  )
```

   Se la funzione da un errore non previsto, migliora la programmazione.

5 Distribuzioni univariate, ECDF, quantili e rappresentazioni grafiche

Esercizio 5.1 Considera il data set:

- https://pietro-coretto.github.io/datasets/balancesheet/readme.txt
- https://pietro-coretto.github.io/datasets/balancesheet/balancesheet.RData

Il file `RData` contiene un `data.frame` chiamato `dat`.

1. Considera la variabile `dat$status`. Quale è il tipo di `status` (continua, discreta, ordinale, etc)? Quali e quanti sono i livelli di espressione distinti di `status`? Filtra i valori mancanti (`NA`), e senza usare la funzione `table()` calcola le frequenze assolute e relative (percentuali), rappresentale in forma tabellare (distribuzione di frequenze).

2. Ripeti il punto precedente usando la funzione `table()`.

3. Considera la variabile `dat$empl`. Quale è il tipo di `empl` (continua, discreta, ordinale, etc)? Quanti, e quali sono i suoi livelli di espressione distinti? Ha senso trattarla come una variabile discreta? Perché? Da questo punto in poi l'esercizio riguarda esclusivamente `empl`.

4. Calcola il numero K di classi ottimali secondo la regola di Sturges. Secondo te ha senso usare questo K? Perché?

5. Supponi di voler trasformare `dat$empl` in classi di livelli (windowing) con $K = 5$. Ottieni gli estremi inferiori e superiori degli intervalli senza usare la funzione `cut()`. Osserva con attenzione gli intervalli ottenuti. Calcola il range e visualizza lo stripchart. Secondo te queste classi hanno senso? Cosa succede?

6. Adesso fissiamo $K = 4$, e fissiamo le classi di livelli in modo discrezionale come segue:

$$[0, 10] \quad (10, 50] \quad (50, 250] \quad (250, \infty)$$

Intuitivamente, ritieni che queste classi siano più appropriate di quelle ottenute al punto 4 e 5? La scelta è discrezionale, ma trova fondamento nella nelle classificazioni standard (vedi: `https://it.wikipedia.org/wiki/Piccola_e_media_impresa`). Guarda attentamente gli esempi/esercizi delle lezioni. Usa la funzione `cut()` controllando l'argomento `breaks` in modo adeguato per ottenere la suddivisione di `dat$empl` nelle quattro classi indicate sopra. Usa la funzione `table()` ed ottieni la corrispondente distribuzione di frequenze assolute e relative.

* * *

Esercizio 5.2 Considera i dati dell'Esercizio 5.1.

1. Per la variabile `dat$status` calcola le frequenze assolute, le frequenze relative, e le frequenze relative percentuali.

2. Rappresenta i risultati precedenti in un'unica struttura tabellare, ad esempio in un `data.frame`. Esporta la tabella in un file testuale di tipo *tab-separated-values* (TSV).

3. Rappresenta la distribuzione di frequenze calcolata nei punti precedenti con tutte le rappresentazioni grafiche che ritieni appropriate.

4. Considera la variabile `dat$curr`. Rappresenta la sua sua distribuzione con i grafici che ritieni opportuni. Esporta ognuno di questi grafici in un file .pdf immaginando che questo file sia destinato alla stampa di qualità.

5. Usando la soluzione al punto 6 dell'Esercizio 5.1, ottieni le frequenze assolute e relative percentuali per le 4 classi. Rappresenta le classi e le frequenze in un'unica struttura tabellare.

* * *

Esercizio 5.3 Si osserva la variabile X su $n = 10$ unità statistiche:

-33	-30.2	26.2	-17.7	-11.5	28.2	-20.3	-1.2	38	1.6

1. Elenca quali rappresentazioni grafiche sono appropriate per la distribuzione di X. Visualizza lo stripchart.

2. Fai un windowing dei dati con i seguenti intervalli

$$[-33, 0] \quad (0, 15] \quad (15, 38].$$

   Per ciascun intervallo calcola le frequenze relative e le densità, in altri termini calcola l'istogramma dei dati. Nota: questo punto va svolto senza usare la funzione `hist()`.

3. Ripeti il punto precedente usando la funzione `hist()` e controlla il risultato. Visualizza il grafico dell'istogramma.

4. Fai un windowing dei dati con $K = 3$ classi uniformi e ripeti il punto 2. In questo caso avrebbe senso riscalare le densità in termini di frequenze assolute? Perchè?

5. Ripeti il punto precedente usando la funzione `hist()` e controlla i risultati.

6. Considera il punto 2. Calcola le frequenze relative cumulate. Usando le classi e le frequenze cumulate applica la definizione di quantile empirico e approssima i quantili ai livelli $\alpha = 0.3, 0.5, 0.9$. Nota: una volta calcolate le frequenze cumulate applica la definizione senza usare R.

* * *

Esercizio 5.4 Si osservano le ore lavorate settimanali (H) su $n = 9$ dipendenti scelti a caso. I dati rilevati sono

40	0	40	24	8	36	8	46.58	26

1. Costruisci l'ECDF per i dati osservati. Calcola il suo valore su tutti i punti osservati.

2. Rappresenta graficamente l'ECDF nominando gli assi opportunamente. Presta attenzione alle unità di misura.

3. Osserva il grafico, quanto vale approssimativamente $\mathbb{F}(20)$? Fornisci un'interpretazione di questo numero. $H = 16$ significa circa 2 giorni di lavoro (considerando 8h/giorno). Quanto vale approssimativamente $\mathbb{F}(16)$? Fornisci un'interpretazione di questo numero.

4. Ordina i dati in senso non-decrescente usando la funzione `sort()`.

5. Ripeti il punto precedente usando la funzione `order()`.

6. Estrai dal vettore delle osservazioni il valore che andrebbe posizionato al quarto posto nella lista ordinata. Estrai valore che andrebbe posizionato al sesto posto nella lista ordinata.

7. Usa i risultati del punto 4 per calcolare i quantili di H usando l'approssimazione basata sullo smoothing dei dati ordinati. Calcola i quantili al livello $\alpha = 0.33, 0.67$. Interpreta il risultato. Nota: svolgi questo punto applicando la definizione e senza usare la funzione `quantile()`.

8. Ripeti il punto precedente usando la funzione `quantile()`
9. Calcola i quantili ai livelli $\alpha = 0.1, 0.5, 0.9$ usando la funzione `quantile()` per produrre entrambe le approssimazioni introdotte a lezione.
10. Considera i quantili approssimati sulla basse dello smoothing dei dati ordinati, ed interpreta il risultato.

* * *

Esercizio 5.5 (*) Considera il data set:

- https://pietro-coretto.github.io/datasets/balancesheet/readme.txt
- https://pietro-coretto.github.io/datasets/balancesheet/balancesheet.RData

Sia Y = `dat$prma` = "*profit margin*". Y è misura in scala percentuale, e misura il margine di profitto dell'impresa, ovvero

$$Y = \frac{\text{Ricavi - Costi}}{\text{Ricavi}} \times 100.$$

In altri termini Y è la percentuale di prezzo di vendita che l'impresa riesce a trasformare in profitto.

1. Chi sono le unità statistiche in questo data set?
2. Rappresenta graficamente l'ECDF e l'istogramma di Y in un'unica finestra grafica.
3. Focalizza l'attenzione sulla sola ECDF. Guardando il grafico, "*ad occhio*", quanto vale approssimativamente $\mathbb{F}(0)$? Interpreta il risultato.
4. Guardando il grafico dell'ECDF, quanto vale approssimativamente il quantile empirico al livello $\alpha = 0.2$, ovvero $Y_{0.2}$?
5. Adesso calcola $Y_{0.2}$ usando la funzione `quantile()` e selezionato l'opportuno `type`. La tua "*approssimazione ad occhio*" del punto precedente era giusta? Interpreta il risultato.
6. Usa la funzione `quantile()` selezionando il `type = 7` (default). Calcola i quantili di Y per i livelli $\alpha = 5\%, 95\%$. Interpreta il risultato. A quale delle due approssimazioni corrisponde il quantile che hai calcolato in questo punto?

* * *

Esercizio 5.6 Considera il data set:

- https://pietro-coretto.github.io/datasets/balancesheet/readme.txt
- https://pietro-coretto.github.io/datasets/balancesheet/balancesheet.RData

Il file `RData` contiene un `data.frame` chiamato `dat`.

1. Prima di tutto controlliamo il comportamento delle funzioni `sort()`, `order()`, e `rank()` quando il vettore in input presenta valori speciali. Prova il seguente codice. Cosa osservi?

```
u <- c(1,NA,-1,0, Inf, NaN)
length(u)
## Funzione sort()
su <- sort(u)
su
length(su)
## Funzione order()
order(u)
u[ order(u) ]
## Funzione rank()
rank(u)
```

2. Data la variabile `dat$roe`:
   - ordina i dati in senso crescente e poi in senso decrescente
   - quale è il terzo valore nel vettore ordinato in senso non-decrescente?
   - quale è la posizione del quinto valore in `dat$roe` nell'ordinamento non-decrescente?

3. Assicurati che `dat$nuts1` sia codificata come un fattore ordinato come segue

```
ITH-Northeast <  ITC-Northwest < ITI-Centre < ITF-South < ITG-Insular Italy
```

4. Ordina le unità statistiche (righe di `dat`) in base a `dat$nuts1`

5. Aggiungi al data set un nuova colonna denominata `dat$status2`. Dove `dat$status2` è un fattore ordinato con due livelli. In particolare

   `dat$status2 = Active` se `dat$status = Active`
   `dat$status2 = NotActive` altrimenti

   L'ordinamento di `dat$status2` è

```
Active < NotActive
```

6. Ordina le unità statistiche in base a `dat$status2`

7. Ordina le unità statistiche in base al seguente criterio gerarchico: `dat$empl`, a parità di `dat$empl` ordina in base a `dat$status2`, a parità di `dat$status2` ordina in base `dat$nuts1`.

8. Considera il data set ordinato in base al criterio gerarchico del punto precedente. Estrai i codici identificativi (`dat$id`) delle prime 100 imprese. Estrai i codici identificativi delle ultime 100 imprese.

6 Posizione, dispersione, simmetria e code di una distribuzione

Esercizio 6.1 Usa il data set `mtcars` disponibile in `R base package`:

```
data("mtcars")
str(mtcars)
?mtcars
```

1. Calcola la media di `mtcars$mpg` senza usare la funzione `mean()`. Controlla il risultato usando la funzione `mean()`
2. Calcola la media di tutte le variabili usando la funzione `colMeans()`. Ripeti i calcoli usando la funzione `apply()`
3. Calcola la mediana di `mtcars$mpg` usando `median()` e poi `quantile()`.
4. Calcola la mediana di tutte le variabili usando la funzione `apply()`
5. Calcola la mediana di `mtcars$mpg` usando l'ordinamento dei dati (non usare la funzione `median()` e/o `quantile()`)
6. Produci un istogramma di `mtcars$mpg` con 5 classi uniformi. Usa l'istogramma per approssimare la Moda dei dati.
7. Tabula le frequenza di `mtcars$gear`. Calcola la moda.
8. Usa la funzione `tapply()` per calcola la media di `mtcars$mpg` condizionatamente a `mtcars$vs=0` e `mtcars$vs=1`. Ripeti i calcoli usando la mediana al posto della media. Leggi la documentazione dei dati e tenta di interpretare i risultati
9. Calcola la media di `mtcars$wt` condizionatamente a `mtcars$vs` e `mtcars$am`. Ripeti i calcoli usando la mediana al posto della media. Leggi la documentazione dei dati e tenta di interpretare i risultati
10. Ripeti il punto precedente usando la mediana al posto della media

* * *

Esercizio 6.2 (*) Usa il data set `USArrests` disponibile in `R base package`:

```
data("USArrests")
str(USArrests)
?USArrests
```

1. Calcola la media di `USArrests$Murder`. Estrai i nomi degli stati per cui `Murder` è superiore alla media (nota: in questo caso il nome delle unità campionarie è codificato come `rownames()`)
2. Calcola la mediana di `USArrests$UrbanPop` e `USArrests$Murder`. Estrai i nomi degli stati per cui `Murder` e `UrbanPop` sono maggiori delle rispettive mediane. Pensando alla geografia degli USA, il risultato ti suggerisce qualche interpretazione?
3. Quale è lo stato più vicino alla media di `rape`? Quale è lo stato più vicino alla mediana di `rape`?
4. Calcola media di tutte le variabili. Per ogni variabile trova lo stato più vicino alla media.
5. Calcola la mediana di tutte le variabili. Per ogni variabile trova lo stato più vicino alla mediana.
6. In base ai due punti precedenti, valutando i risultati globalmente, esiste uno stato che è maggiormente vicino alla tendenza centrale delle variabili?

* * *

Esercizio 6.3 (*) Usa il data set `PlantGrowth` disponibile in `R base package`:

```
data("PlantGrowth")
str(PlantGrowth)
?PlantGrowth
```

1. Calcola i quartili di `PlantGrowth$weight`.
2. Calcola la media e la moda di `PlantGrowth$weight`. Per la moda usa l'approssimazione basata sull'istogramma fissando il numero di classi uniformi che ritieni più opportuno.
3. Visualizza uno `stripchart()` di `PlantGrowth$weight`. Sullo stesso grafico, usando dei segmenti verticali di colore diverso, indica la posizione di media, mediana, moda. Riporta anche una legenda per la corretta interpretazione dei colori.
4. Visualizza l'istogramma del punto 2. Sullo stesso grafico riporta lo stripchart dei punti usando `rug()`. Con segmenti verticali di colore diverso, indica la posizione di mediana e quartili. Riporta una legenda per la corretta interpretazione dei colori.
5. Visualizza l'ecdf `PlantGrowth$weight`. Analogamente ai punti 3 e 4, sul grafico visualizza media, mediana e quartili.
6. Calcola i quartili di `PlantGrowth$weight` condizionatamene alle categorie di `PlantGrowth$group`
7. Calcola i decili di `PlantGrowth$weight`.
8. Calcola i percentili `PlantGrowth$weight`.
9. Rappresenta la curva dei percentili di `PlantGrowth$weight`. Sull'asse orizzontale rappresenta il livello del percentile ($\alpha = 1\%, 2\%, \ldots$), sull'asse verticale rappresenta i percentili calcolati al punto precedente.
10. Calcola i percentili di `PlantGrowth$weight` separatamente per i tre gruppi sperimentali `{ctrl, trt1, trt2}` indicati nel fattore `PlantGrowth$group`
11. Rappresenta le curve dei percentili calcolate al punto precedente in un unico grafico (come nel punto 9) usando tre colori diversi ed una legenda per spiegare i colori. Interpreta il grafico ottenuto.

* * *

Esercizio 6.4 Usa il data set `PlantGrowth` disponibile in `R base package`:

```
data("PlantGrowth")
str(PlantGrowth)
?PlantGrowth
```

Sia $X =$ `PlantGrowth$weight`

1. Calcola le deviazioni assolute e quadratiche da $a = 4.5$

2. Calcola le deviazioni assolute e quadratiche totali $a = 5$

3. Costruisci la griglia di valori

```r
gra <- seq(0, 10, length.out = 100)
```

   Per ogni valore a della griglia `gra` calcola le deviazioni quadratiche totali ed assolute da a

4. Costruisci il grafico delle deviazioni totali assolute e quadratiche in funzione dei valori della griglia

5. Verifica graficamente che le deviazioni quadriatiche totali sono minimizzate in corrispondenza di $a = \bar{x}$, mentre quelle assolute sono minimizzate in corrispondenza di $a = \mathrm{Med}(X)$

6. Calcola la varianza, deviazione standard e IQR di X senza usare le funzioni `var()`, `sd()`, e `IQR`. Controlla il risultato usando `var()`, `sd()`, e `IQR`.

7. Calcola varianza, deviazione standard e IQR di X condizionatamente (separatamente) per i tre gruppi sperimentali `{ctrl, trt1, trt2}` indicati nel fattore `PlantGrowth$group`.

8. L'unità di misura è l'oncia. Quale trasformazione lineare consente di trasformare i dati in grammi? Applica la trasformazione e controlla che siano soddisfatte tutte le proprietà sulle trasformazioni lineari viste durante il corso.

* * *

Esercizio 6.5 Considera il data set:

- https://pietro-coretto.github.io/datasets/balancesheet/readme.txt
- https://pietro-coretto.github.io/datasets/balancesheet/balancesheet.RData

I dati sono contenuti in un data.frame chiamato `dat`.

1. Costruisci un nuovo `data.frame`, chiamato `X`, che contiene solo le variabili numeriche di `balancesheet.RData`.

2. Senza usare la funzione `scale()` standardizza tutte le colonne di `X`.

3. Ripeti l'operazione precedente applicando la funzione `scale()` su ogni colonna di `X`.

4. Ripeti il punto precedente mediante il comando: `scale(X)`. Il risultato di questo punto coincide con quello dei due punti precedenti?

* * *

Esercizio 6.6 In relazione alla curva di densità

$$\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2},$$

del modello normale standard risolvi i seguenti punti.

1. Calcola $\phi(-2)$ e $\phi(2)$.
2. Su una griglia di valori (piuttosto fitta) $z \in [-10, 10]$, calcola il corrispondente valore $\phi(z)$
3. Programma una funzione `phi(z)` che calcola il valore di ϕ nel punto z.
4. Una la risposta al punto precedente. Verifica mediante l'ottimizzatore generico `nlm` che ϕ ha un massimo in corrispondenza di $z = 0$
5. Ripeti il punto 2 usando la funzione `phi()`. Visualizza anche il grafico di ϕ sull'intervallo $[-4, 4]$

* * *

Esercizio 6.7 (*) Usa i dati ed i risultati dell'esercizio Esercizio 6.1. Sia $X =$ `dat$mpg`.

1. Visualizza l'istogramma e l'ECDF di X. La distribuzione appare unimodale e simmetrica?
2. Calcola le misure di simmetria γ_1 e B_1. Cosa concludi?
3. Applica la trasformazione $Y = X^{\frac{3}{10}}$ ai dati. Calcola media e mediana di Y, queste seguono approssimativamente la relazione $\bar{y} = \bar{x}^{\frac{3}{10}}$ e $\text{Med}_Y = \text{Med}_X^{\frac{3}{10}}$?
4. Visualizza l'istogramma e l'ECDF di Y. La distribuzione appare unimodale e simmetrica?
5. Calcola le misure di simmetria γ_1 e B_1 su Y. Cosa concludi?
6. Visualizza il Normal QQ-plot di X e di Y. Quale è la tua conclusione?

7 Metodi robusti e smoothing

Esercizio 7.1 Considera il data set:

- https://pietro-coretto.github.io/datasets/nls80/readme.txt
- https://pietro-coretto.github.io/datasets/nls80/nls80.RData

I dati sono contenuti nel data.frame `NLS80`.

1. Rappresenta l'istogramma di `NLS80$wage` riportando sotto lo stripchart. Rappresenta il Normal QQ-plot per la stessa variabile. Commenta il risultato.
2. Calcola media, mediana e varianza di `NLS80$wage`. Visualizza la media e la mediana sull'istogramma.

3. Lavorando sempre su `NLS80$wage` calcola: trimean, media troncata al 10%, media troncata al 25%, e MAD. Confronta i risultati con il punto precedente, e cerca una spiegazione delle differenze osservate.

* * *

Esercizio 7.2 (*) Considera il data set:

- https://pietro-coretto.github.io/datasets/nls80/readme.txt
- https://pietro-coretto.github.io/datasets/nls80/nls80.RData

I dati sono contenuti nel data.frame `NLS80`.

1. Calcola i quartili e l'IQR di `NLS80$wage`.
2. Usa i risultati del punto precedente per calcolare i *Tukey's fences* con $k = 1.5$ e $k = 3$.
3. Calcola il numero di outliers e gross-outliers. Trova gli indici delle unità statistiche classificate come outliers e gross-outliers.
4. Costruisci una variabile categoriale chiamata `U` con i tre livelli ordinati: `reg < ol < gol`. Per l'unità i-ma: `U[i] = reg` se i è una unità regolare, `U[i] = ol` se i è un outlier, `U[i] = gol` se i è un gross-outlier.
5. Visualizza con un barplot della distribuzione di `U`.
6. Estrai i dati su `NLS80$age` corrispondenti agli outliers trovati rispetto a `NLS80$wage`.
7. Usando i risultati dei punti precedenti visualizza la distribuzione di `NLS80$educ` per le unità segnalate come regolari. Ripeti la visualizzazione per le unità segnalate come potenziali outliers.
8. Ora supponi che alcuni punti campionari siano stai registrati in modo inesatto, ad esempio aggiungendo uno zero finale.

```
NLS80$iq[24] <- 1450
NLS80$iq[25] <- 1140
```

   Standardizza la i dati contaminati su `NLS80$iq` usando al solito la media e la varianza. Visualizza l'istogramma e l'ECDF dei dati appena standardizzati.

9. Ripeti la standardizzazione usando misure di centralità e dispersione robuste, ovvero la media troncata al 10%, e il MAD. Visualizza l'istogramma e l'ECDF dei dati trasformati. Confronta i risultati con il punto precedente.

* * *

Esercizio 7.3 (*) Usa il data set `iris` (*Edgar Anderson's Iris Data*) disponibile in `R base package`:

```r
data("iris")
str(iris)
?iris
```

1. Calcola la media e varianza per tutte le variabili numeriche contenute nel data set.
2. Calcola la media troncata al 2% per tutte le variabili numeriche contenute nel data set.
3. Calcola IQR e MAD per tutte le variabili numeriche nel data set. Confronta e commenta i risultati dei punti precedenti.
4. Rappresenta il boxplot per ognuna delle variabili numeriche del data set. Commenta i boxplots (posizione, asimmetria, code, potenziali outliers).
5. Visualizza in un unico grafico i boxplot paralleli per tutte le variabili numeriche contenute nel data set.
6. Calcola Trimean e MAD per `iris$Sepal.Width` condizionatamente alla specie di appartenenza. La specie di appartenenza è contenuta in `iris$Species`.
7. Visualizza il boxplot `iris$Sepal.Width` condizionatamente alla specie di appartenenza.
8. Costruisci una finestra grafica divisa in due righe e due colonne. Visualizza i boxplot di ciascuna variabile numerica di `iris` condizionatamente alla specie di appartenenza. Commenta il risultato.

* * *

Esercizio 7.4 Usa il data set `faithful` (*Old Faithful Geyser Data*) disponibile in `R base package`:

```r
data("faithful")
str(faithful)
?faithful
```

1. Rappresenta ECDF, istrogramma e boxplot di `faithful$eruptions`.
2. Visualizza l'istrogramma usando il bandwidth di "Freedman-Diaconis". Ripeti il punto provando a variare il numero di classi uniformi dell'istogramma con $K = 5, 10, 20, 50$. Commenta i risultati.
3. Visualizza l'approssimazione kernel della densità di `faithful$eruptions` usando i parametri di default di `R`.
4. Ripeti il punto precedente tenendo fissa la funzione kernel di default usata da `R`, e variando il parametro di bandwidth per i valori `bw=0.01, 0.1, 0.25, 0.5, 1, 2, 10`. Commenta i risultati.
5. Ripeti il punto precedente usando la funzione kernel di Epanechnikov e commenta i risultati.
6. Produci e visualizza l'approssimazione della densità usando la funzione kernel di Epanechnikov, e la larghezza di banda di Sheather e Jones.
7. Per la variabile `faithful$waiting` visualizza in un unica finestra grafica le seguenti quattro rappresentazioni della sua distribuzione: stripchart, ECDF, boxplot e densità kernel.