Main1

2024-01-19

Analisi Esplorativa del Mercato Immobiliare del Texas

Parte 1

installazione pacchetti:

```
#install.packages("moments")
#install.packages("psych")
#install.packages("dplyr")
source("Utils.R")
#tinytex::install_tinytex()
```

1) Importa il dataset "Real Estate Texas.csv.

```
RealEstateTexax_Dataframe = read.csv("Real Estate Texas.csv")
```

per riferirmi alle colonne del dataframe senza usare la notazione: RealEstateTexax_Dataframe$nome_colonna

```
attach(RealEstateTexax_Dataframe)
```

2) Indica il tipo di variabili contenute nel dataset.

```
class(city) #qualitativa nominale
```

```
## [1] "character"
```

```
class(year) #qualitativa ordinale
```

```
## [1] "integer"
```

```
class(month) #qualitativa ordinale
```

```
## [1] "integer"
```

```
class(sales) #quantitativa discreta, scala rapporti
```

```
## [1] "integer"
```

```r
class(volume) #quantitativa continua, scala rapporti
```

```
## [1] "numeric"
```

```r
class(median_price) #quantitativa continua, scala rapporti
```

```
## [1] "numeric"
```

```r
class(listings) #quantitativa discreta, scala rapporti
```

```
## [1] "integer"
```

```r
class(months_inventory) #quantitavia discreta, scala rapporti
```

```
## [1] "numeric"
```

3) Calcola Indici di posizione, variabilità e forma per tutte le variabili per le quali ha senso farlo, per le altre crea una distribuzione di frequenza.

INDICI DI POSIZIONE:

Per city year e month essendo variabili qualitative ha senso calcolare solo la moda.

Variabile city

```r
table(city)
```

```
## city
##              Beaumont Bryan-College Station                 Tyler
##                    60                    60                    60
##         Wichita Falls
##                    60
```

```r
max(table(city))
```

```
## [1] 60
```

distribuzione equimodale. Valore della fequenza: 60.

Variabile year

```r
table(year)
```

```
## year
## 2010 2011 2012 2013 2014
##   48   48   48   48   48
```

```r
max(table(year))
```

```
## [1] 48
```

distribuzione equimodale. Valore della fequenza: 48.

Variabile month

```r
table(month)
```

```
## month
##  1  2  3  4  5  6  7  8  9 10 11 12
## 20 20 20 20 20 20 20 20 20 20 20 20
```

```r
max(table(month))
```

```
## [1] 20
```

distribuzione equimodale. Valore della fequenza:20.

Variabile sales

divisione in classi di sales:

```r
sales_div_classi = cut(sales, seq(min(sales), max(sales), (max(sales)-min(sales))/10 ))
```

distribuzione in frequenza di sales:

```r
ni = table(sales_div_classi)
fi = table(sales_div_classi) / length(sales)
Ni = cumsum(table(sales_div_classi))
Fi = cumsum(table(sales_div_classi)) / length(sales)
sales_distr_freq = as.data.frame( cbind( ni, fi, Ni, Fi))
```

calcolo della moda, vengono presentati due metodi:

```r
table(sales_div_classi)
```

```
## sales_div_classi
##  (79,113] (113,148] (148,182] (182,217] (217,251] (251,285] (285,320] (320,354]
##        38        44        47        30        20        21        18         9
## (354,389] (389,423]
##         8         3
```

```r
max(table(sales_div_classi))  #1° metodo
```

```
## [1] 47
```

```r
max(sales_distr_freq["ni"])  #2° metodo
```

```
## [1] 47
```

moda: classe (148,182], con valore della frequenza uguale a 47.

calcolo della mediana:

```r
median(sort(sales))   #1° metodo
```

```
## [1] 175.5
```

```r
x = sales_distr_freq["Fi"][[1]]     #2° metodo
index = which(x >= 0.50)[1]
rownames(sales_distr_freq)[index]
```

```
## [1] "(148,182]"
```

mediana: 175.5. Il secondo metodo ritorna la classe "(148,182]" ovvero che la mediana è contenuta all'interno del range di tale classe. Il risultato è coerente con il valore 175.5.

quantili:

```r
quantile(sort(sales))[c(1, 5)]
```

```
##   0% 100%
##   79  423
```

minimo: 79, massimo: 423.

media:

ARITMETICA

```r
mean(sales)
```

```
## [1] 192.2917
```

media aritmetica: 192.2917.

PONDERATA

```r
valori = seq(min(sales), max(sales), (max(sales)-min(sales))/10 ) + (((max(sales)-min(sales))/10) /2)
valori = valori[-11]
pesi = sales_distr_freq["ni"][[1]]
weighted.mean(valori, pesi)
```

```
## [1] 193.763
```

media ponderata: 193.763. NOTA: anche con i dati sintetizzati si ha avuto un'ottima precisione.

Variabile volume

divisione in classi di volume:

```
volume_div_classi = cut(volume, seq(min(volume), max(volume), (max(volume)-min(volume))/10 ))
```

distribuzione in frequenza di volume:

```
ni = table(volume_div_classi)
fi = table(volume_div_classi) / length(volume)
Ni = cumsum(table(volume_div_classi))
Fi = cumsum(table(volume_div_classi)) / length(volume)

volume_distr_freq = as.data.frame( cbind( ni, fi, Ni, Fi))
```

calcolo della moda:

```
table(volume_div_classi)
```

```
## volume_div_classi
## (8.17,15.7] (15.7,23.2] (23.2,30.8] (30.8,38.3] (38.3,45.9] (45.9,53.4]
##          46          46          45          34          21          22
## (53.4,60.9] (60.9,68.5]   (68.5,76]   (76,83.5]
##           8           9           4           4
```

```
max(table(volume_div_classi))  #1° metodo
```

```
## [1] 46
```

```
max(volume_distr_freq["ni"])  #2° metodo
```

```
## [1] 46
```

moda: distribuzione bimodale con modalità: (8.17,15.7] e (15.7,23.2]. Valore Frequenza: 46

calcolo della mediana:

```
median(sort(volume))   #1° metodo
```

```
## [1] 27.0625
```

```
x = volume_distr_freq["Fi"][[1]]     #2° metodo
index = which(x >= 0.50)[1]
rownames(volume_distr_freq)[index]
```

```
## [1] "(23.2,30.8]"
```

mediana: 27.0625 Il secondo metodo ritorna la classe "(23.2,30.8]" ovvero che la mediana è contenuta all'interno del range di tale classe. Il risultato è coerente con il valore 27.0625.

quantili:

```r
quantile(sort(volume))[c(1, 5)]
```

```
##     0%   100%
##  8.166 83.547
```

minimo: 8.166, massimo: 83.547

media:

ARITMETICA

```r
mean(volume)
```

```
## [1] 31.00519
```

media aritmetica: 31.00519

PONDERATA

```r
valori = seq(min(volume), max(volume), (max(volume)-min(volume))/10 ) + (((max(volume)-min(volume))/10)
valori = valori[-11]
pesi = volume_distr_freq["ni"][[1]]
weighted.mean(valori, pesi)
```

```
## [1] 31.20609
```

media ponderata: 31.20609.

Variabile median_price

divisione in classi di median_price:

```r
median_price_div_classi = cut(median_price, seq(min(median_price), max(median_price), (max(median_price)
```

distribuzione in frequenza di median_price:

```r
ni = table(median_price_div_classi)
fi = table(median_price_div_classi) / length(median_price)
Ni = cumsum(table(median_price_div_classi))
Fi = cumsum(table(median_price_div_classi)) / length(median_price)

median_price_distr_freq = as.data.frame( cbind( ni, fi, Ni, Fi))
```

calcolo della moda:

```r
table(median_price_div_classi)
```

```
## median_price_div_classi
## (7.38e+04,8.44e+04]  (8.44e+04,9.5e+04]  (9.5e+04,1.06e+05] (1.06e+05,1.16e+05]
##                   1                   16                   23                   17
## (1.16e+05,1.27e+05] (1.27e+05,1.38e+05] (1.38e+05,1.48e+05] (1.48e+05,1.59e+05]
##                  25                   48                   38                   46
## (1.59e+05,1.69e+05]  (1.69e+05,1.8e+05]
##                  16                    9
```

```r
max(table(median_price_div_classi))  #1° metodo
```

```
## [1] 48
```

```r
max(median_price_distr_freq["ni"])  #2° metodo
```

```
## [1] 48
```

moda: classe (1.27e+05,1.38e+05], con valore della frequenza uguale a 48.

calcolo della mediana:

```r
median(sort(median_price))   #1° metodo
```

```
## [1] 134500
```

```r
x = median_price_distr_freq["Fi"][[1]]     #2° metodo
index = which(x >= 0.50)[1]
rownames(median_price_distr_freq)[index]
```

```
## [1] "(1.27e+05,1.38e+05]"
```

mediana: 134500. secondo metodo: classe "(1.27e+05,1.38e+05]".

quantili:

```r
quantile(sort(median_price))[c(1, 5)]
```

```
##     0%   100%
##  73800 180000
```

minimo: 73800, massimo: 180000

media:

ARITMETICA

```r
mean(median_price)
```

```
## [1] 132665.4
```

media aritmetica: 132665.4.

PONDERATA

```r
valori = seq(min(median_price), max(median_price), (max(median_price)-min(median_price))/10 ) + (((max(m
valori = valori[-11]
pesi = median_price_distr_freq["ni"][[1]]
media_ponderata = weighted.mean(valori, pesi)
```

media ponderata: 132965.4.

Nota: in questo caso si può vedere la differenza di risultato dovuta ai due metodi. Nel secondo a causa della sintetizzazione dei dati si ha un valore approssimato.

Variabile listings

divisione in classi di listings:

```
listings_div_classi = cut(listings, seq(min(listings), max(listings), (max(listings)-min(listings))/10 )
```

distribuzione in frequenza di listings:

```
ni = table(listings_div_classi)
fi = table(listings_div_classi) / length(listings)
Ni = cumsum(table(listings_div_classi))
Fi = cumsum(table(listings_div_classi)) / length(listings)

listings_distr_freq = as.data.frame( cbind( ni, fi, Ni, Fi))
```

calcolo della moda:

```
table(listings_div_classi)
```

```
## listings_div_classi
##           (743,998]      (998,1.25e+03] (1.25e+03,1.51e+03] (1.51e+03,1.76e+03]
##                  53                  20                  20                  67
## (1.76e+03,2.02e+03] (2.02e+03,2.27e+03] (2.27e+03,2.53e+03] (2.53e+03,2.79e+03]
##                  19                   1                   2                  15
## (2.79e+03,3.04e+03]  (3.04e+03,3.3e+03]
##                  24                  18
```

```
max(table(listings_div_classi))  #1° metodo
```

```
## [1] 67
```

```
max(listings_distr_freq["ni"])  #2° metodo
```

```
## [1] 67
```

moda: classe (1.51e+03,1.76e+03], con valore della frequenza uguale a 67.

calcolo della mediana

```
median(sort(listings))   #1° metodo
```

```
## [1] 1618.5
```

```
x = listings_distr_freq["Fi"][[1]]     #2° metodo
index = which(x >= 0.50)[1]
rownames(listings_distr_freq)[index]
```

```
## [1] "(1.51e+03,1.76e+03]"
```

mediana: 1618.5. secondo metodo: classe “(1.51e+03,1.76e+03]”

quantili:

```r
quantile(sort(listings))[c(1, 5)]
```

```
##   0% 100%
##  743 3296
```

minimo: 743, massimo: 3296

media

ARITMETICA

```r
mean(listings)
```

```
## [1] 1738.021
```

media aritmetica: 1738.021.

PONDERATA

```r
valori = seq(min(listings), max(listings), (max(listings)-min(listings))/10 ) + (((max(listings)-min(lis
valori = valori[-11]
pesi = listings_distr_freq["ni"][[1]]
media_ponderata = weighted.mean(valori, pesi)
```

media ponderata: 1739.097.

Variabile month_inventory

divisione in classi di months_inventory:

```r
months_inventory_div_classi = cut(months_inventory, seq(min(months_inventory), max(months_inventory), (
```

distribuzione in frequenza di months_inventory:

```r
ni = table(months_inventory_div_classi)
fi = table(months_inventory_div_classi) / length(months_inventory)
Ni = cumsum(table(months_inventory_div_classi))
Fi = cumsum(table(months_inventory_div_classi)) / length(months_inventory)

months_inventory_distr_freq = as.data.frame( cbind( ni, fi, Ni, Fi))
```

calcolo della moda:

```r
table(months_inventory_div_classi)
```

```
## months_inventory_div_classi
##  (3.4,4.55]  (4.55,5.7]  (5.7,6.85]    (6.85,8]    (8,9.15] (9.15,10.3]
##           7           8           9          54          54          28
## (10.3,11.4] (11.4,12.6] (12.6,13.8] (13.8,14.9]
##          34          29           9           7
```

```r
max(table(months_inventory_div_classi))  #1° metodo
```

```
## [1] 54
```

```r
max(months_inventory_distr_freq["ni"])  #2° metodo
```

```
## [1] 54
```

moda: distribuzione bimodale delle classi (6.85,8] e (8,9.15], con valore della frequenza uguale a 54.

calcolo della mediana:

```r
median(sort(months_inventory))   #1° metodo
```

```
## [1] 8.95
```

```r
x = months_inventory_distr_freq["Fi"][[1]]     #2° metodo
index = which(x >= 0.50)[1]
rownames(months_inventory_distr_freq)[index]
```

```
## [1] "(8,9.15]"
```

mediana: 8.95 secondo metodo: classe "(8,9.15]".

quantili:

```r
quantile(sort(months_inventory))[c(1, 5)]
```

```
##   0% 100%
##  3.4 14.9
```

minimo: 3.4, massimo: 14.9

media:

ARITMETICA

```r
mean(months_inventory)
```

```
## [1] 9.1925
```

media aritmetica: 9.1925.

PONDERATA

```r
valori = seq(min(months_inventory), max(months_inventory), (max(months_inventory)-min(months_inventory)
valori = valori[-11]
pesi = months_inventory_distr_freq["ni"][[1]]
media_ponderata = weighted.mean(valori, pesi)
```

media ponderata: 9.200523.

INDICI DI VARIABILITA'

Per le variabili qualitative è stato calcolato solo l'indice di Gini con la funzione "indice_gini" che si trova nel file Utils.R. Per le altre variabili oltre all'indice di Gini sono stati calcolati anche tutti gli altri indici di variabilità tramite la funzione "indici_di_variabilità", sempre del file Utils.R. Come la moda, anche per il calcolo dell'indice di Gini è stata prima calcolata la distribuzione in frequenze quando la variabile è quantitativa.

Variabile city

```
gini_city = indice_gini(city, tipo_variabile="qualitativa")
gini_city
```

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

gini: 1. Le classi hanno il massimo livello di omogeneità, infatti la distribuzione è equimodale.

Variabile year

```
gini_year = indice_gini(year, tipo_variabile="qualitativa")
gini_year
```

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

gini: 1. Le classi hanno il massimo livello di omogeneità, infatti la distribuzione è equimodale.

Variabile month

```
gini_month = indice_gini(month, tipo_variabile="qualitativa")
gini_month
```

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

gini: 1. Le classi hanno il massimo livello di omogeneità, infatti la distribuzione è equimodale.

Variabile sales

```
indici_di_variabilità("sales", sales)
```

```
## [1] "**RANGE**"
## [1] "il range per la variabile 'sales' è: 344 "
##
## [1] "**RANGE INTERQUARTILE**"
## [1] "il range interquartile per la variabile 'sales' è: 120 "
##
## [1] "**VARIANZA**"
## [1] "la variazna per la variabile 'sales' è: 6344.29951185495 "
##
```

```
## [1] "**DEVIAZIONE STANDARD**"
## [1] "la deviazione standard per la variabile 'sales' è: 79.6511111777793 "
##
## [1] "**COEFFICIENTE DI VARIAZIONE**"
## [1] "il coefficiente di variazione per la variabile 'sales' è: 41.4220296482492 "
##
## [1] "**INDICE DI GINI**"
## [1] "l' indice di GINI per la variabile 'sales' è: 0.960493827160494 "
```

Variabile volume

```
indici_di_variabilità("volume", volume)
```

```
## [1] "**RANGE**"
## [1] "il range per la variabile 'volume' è: 75.381 "
##
## [1] "**RANGE INTERQUARTILE**"
## [1] "il range interquartile per la variabile 'volume' è: 23.2335 "
##
## [1] "**VARIANZA**"
## [1] "la variazna per la variabile 'volume' è: 277.270692404027 "
##
## [1] "**DEVIAZIONE STANDARD**"
## [1] "la deviazione standard per la variabile 'volume' è: 16.6514471564494 "
##
## [1] "**COEFFICIENTE DI VARIAZIONE**"
## [1] "il coefficiente di variazione per la variabile 'volume' è: 53.7053586805415 "
##
## [1] "**INDICE DI GINI**"
## [1] "l' indice di GINI per la variabile 'volume' è: 0.946855709876543 "
```

Variabile median_price

```
indici_di_variabilità("median_price", median_price)
```

```
## [1] "**RANGE**"
## [1] "il range per la variabile 'median_price' è: 106200 "
##
## [1] "**RANGE INTERQUARTILE**"
## [1] "il range interquartile per la variabile 'median_price' è: 32750 "
##
## [1] "**VARIANZA**"
## [1] "la variazna per la variabile 'median_price' è: 513572983.089261 "
##
## [1] "**DEVIAZIONE STANDARD**"
## [1] "la deviazione standard per la variabile 'median_price' è: 22662.148686505 "
##
## [1] "**COEFFICIENTE DI VARIAZIONE**"
## [1] "il coefficiente di variazione per la variabile 'median_price' è: 17.0821825732064 "
##
## [1] "**INDICE DI GINI**"
## [1] "l' indice di GINI per la variabile 'median_price' è: 0.958699845679012 "
```

Variabile listings

```
indici_di_variabilità("listings", listings)
```

```
## [1] "**RANGE**"
## [1] "il range per la variabile 'listings' è: 2553 "
##
## [1] "**RANGE INTERQUARTILE**"
## [1] "il range interquartile per la variabile 'listings' è: 1029.5 "
##
## [1] "**VARIANZA**"
## [1] "la variazna per la variabile 'listings' è: 566568.966091353 "
##
## [1] "**DEVIAZIONE STANDARD**"
## [1] "la deviazione standard per la variabile 'listings' è: 752.707756098841 "
##
## [1] "**COEFFICIENTE DI VARIAZIONE**"
## [1] "il coefficiente di variazione per la variabile 'listings' è: 43.3083275909432 "
##
## [1] "**INDICE DI GINI**"
## [1] "l' indice di GINI per la variabile 'listings' è: 0.926138117283951 "
```

Variabile month_inventory

```
indici_di_variabilità("months_inventory", months_inventory)
```

```
## [1] "**RANGE**"
## [1] "il range per la variabile 'months_inventory' è: 11.5 "
##
## [1] "**RANGE INTERQUARTILE**"
## [1] "il range interquartile per la variabile 'months_inventory' è: 3.15 "
##
## [1] "**VARIANZA**"
## [1] "la variazna per la variabile 'months_inventory' è: 5.30688912133891 "
##
## [1] "**DEVIAZIONE STANDARD**"
## [1] "la deviazione standard per la variabile 'months_inventory' è: 2.30366862229334 "
##
## [1] "**COEFFICIENTE DI VARIAZIONE**"
## [1] "il coefficiente di variazione per la variabile 'months_inventory' è: 25.0603059264982 "
##
## [1] "**INDICE DI GINI**"
## [1] "l' indice di GINI per la variabile 'months_inventory' è: 0.938715277777778 "
```

INDICI DI FORMA

Gli indici di forma non sono stati calcolati per le variabili qualitative. Per le restanti è stata usata la funzione "indici_di_forma" che consente di calcolare sia l'asimmetria che la curtosi di una distribuzione. Anch'essa è contenuta nel file Utils.R.

```r
library(moments)
```

Variabile sales

```r
indici_di_forma("sales", sales)
```

```
## [1] "**ASIMMETRIA**"
## [1] "l'indice di asimmetria per la variabile sales è: 0.713620553137559 "
##
## [1] "**CURTOSI**"
## [1] "l'indice di curtosi per la variabile sales è: -0.335519959419668 "
```

prova correttezza funzioni "indice asimmetria" e "indice curtosi" del file Utils.R tramite le funzioni del pacchetto "moments":

```r
skewness(sales)
```

```
## [1] 0.718104
```

```r
kurtosis(sales) - 3
```

```
## [1] -0.3131764
```

ok piccola differenza dovuta probabilmente ad arrotondamenti.

Variabile volume

```r
indici_di_forma("volume", volume)
```

```
## [1] "**ASIMMETRIA**"
## [1] "l'indice di asimmetria per la variabile volume è: 0.879218152706982 "
##
## [1] "**CURTOSI**"
## [1] "l'indice di curtosi per la variabile volume è: 0.150567261471582 "
```

Variabile median_price

```r
indici_di_forma("median_price", median_price)
```

```
## [1] "**ASIMMETRIA**"
## [1] "l'indice di asimmetria per la variabile median_price è: -0.362276797730666 "
##
## [1] "**CURTOSI**"
## [1] "l'indice di curtosi per la variabile median_price è: -0.642729204225302 "
```

Variabile listings

```r
indici_di_forma("listings", listings)
```

```
## [1] "**ASIMMETRIA**"
## [1] "l'indice di asimmetria per la variabile listings è: 0.645443093804895 "
##
## [1] "**CURTOSI**"
## [1] "l'indice di curtosi per la variabile listings è: -0.810153446076231 "
```

Variabile month_inventory

```r
indici_di_forma("months_inventory", months_inventory)
```

```
## [1] "**ASIMMETRIA**"
## [1] "l'indice di asimmetria per la variabile months_inventory è: 0.0407194374109849 "
##
## [1] "**CURTOSI**"
## [1] "l'indice di curtosi per la variabile months_inventory è: -0.197944757394652 "
```

4) Qual è la variabile con variabilità più elevata? Come ci sei arrivato? E quale quella più asimmetrica?

La variabile con variabilità più elevata è il volume, conclusione fatta osservando il coefficiente di variazione delle diverse variabili. Ho utilizzato questo indice perchè permette di confrontare variabili provenienti da distribuzioni diverse. La variabile più asimmetrica è ancora il volume. Per giungere a questa conclusione ho confrontato il valore assoluto dei vari indici di asimmetria delle variabili e prendendo quello più grande, ricordando che un indice di asimmetria pari a zero corrisponde a una simmetria perfetta, come quella della distribuzione normale.

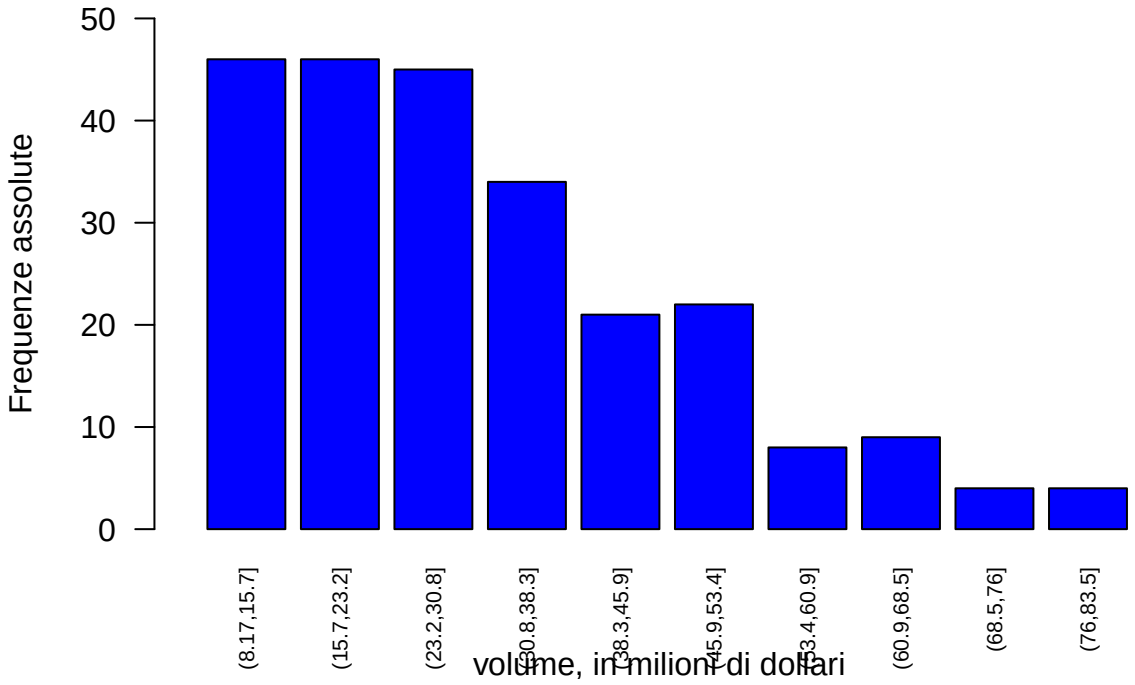
5) Dividi una delle variabili quantitative in classi, scegli tu quale e come, costruisci la distribuzione di frequenze, il grafico a barre corrispondente e infine calcola l'indice di Gini.

distribuzione in frequenze:

```r
distribuzione_frequenze_volume = get_distribuzione_frequenze(volume)
```

grafico a barre:

```r
barplot(distribuzione_frequenze_volume$ni,
        xlab = "volume, in milioni di dollari",
        ylab = "Frequenze assolute",
        names.arg = rownames(distribuzione_frequenze_volume),
        # cex.axis = 0.8,    -> riduzione font per labelasse y
        # cex.lab = 0.8,     #-> riduzione font per per label asse x
        cex.names = 0.7,  # -> riduzione font per per descrizione classi
        xlim = c(0,12),
        ylim = c(0,50),
        las=2,
        col="blue")
```



Nota: per una visione corretta del grafico

Per l'indice di Gini vedere il punto 3.

6) Indovina l'indice di gini per la variabile city.

```
gini_city = indice_gini(city, tipo_variabile="qualitativa")
gini_city
```

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

l'indice di Gini è uguale a 1, infatti le classi hanno il massimo livello di omogeneità ovvero la distribuzione è equimodale.

7) Qual è la probabilità che presa una riga a caso di questo dataset essa riporti la città "Beaumont"? E la probabilità che riporti il mese di Luglio? E la probabilità che riporti il mese di dicembre 2012?

Essendo i dati noti si può usare l'approccio classico ovvero numero di casi favorevoli diviso il totale dei casi.

```
prob_beaumont = table(city)["Beaumont"] / length(city)
prob_beaumont
```

```
## Beaumont
##     0.25
```

La probabilità che esca la città di Beaumont è 0.25, infatti ho 4 valori possibili equiprobabili.

```
prob_luglio = table(month)[7] / length(month)
prob_luglio
```

```
##          7
## 0.08333333
```

La probabilità che esca il mese di luglio è 0.8333, infatti ho 12 valori possibili equiprobabili.

```
prob_dic_2012 = sum( RealEstateTexax_Dataframe[["month"]] == "12" & RealEstateTexax_Dataframe[["year"]]
prob_dic_2012  # 0.01666667
```

```
## [1] 0.01666667
```

La probabilità che esca il mese di dicembre 2012 è 0.0166.

8) Esiste una colonna col prezzo mediano, creane una che indica invece il prezzo medio, utilizzando le altre variabili che hai a disposizione

```
RealEstateTexax_Dataframe["mean_price"] = volume/sales*1000000
```

9) Prova a creare un'altra colonna che dia un'idea di "efficacia" degli annunci di vendita. Riesci a fare qualche considerazione?

```
RealEstateTexax_Dataframe["conversion_coefficient"] = sales/listings*100

coefficiente_conversione = RealEstateTexax_Dataframe$conversion_coefficient

mean_coefficiente_conversione = mean(coefficiente_conversione)


std_coefficiente_conversione = sd(coefficiente_conversione)
```

L'indicatore creato è stato chiamato "Coefficiente di Conversione" e indica la percentuale di annunci che si sono convertiti in vendite.

con media:

```
mean_coefficiente_conversione
```

```
## [1] 11.87449
```

e deviazione standard:

```
std_coefficiente_conversione
```

```
## [1] 4.6899
```

Inoltre sono state fatte le seguenti osservazioni:

1 - Si è riscontrato che la maggior efficacia degli annunci la si ha nel periodo estivo:

```r
library(dplyr)

dati_raggruppati1 <- RealEstateTexax_Dataframe[ c("month", "conversion_coefficient")] %>%
  group_by(month) %>%
  summarise(media_conversione1 = mean(conversion_coefficient, na.rm = TRUE))
```

2 - Si è riscontrato un trend positivo nell'efficacia degli annunci in base al tempo:

```r
dati_raggruppati2 <- RealEstateTexax_Dataframe[ c("year", "conversion_coefficient")] %>%
  group_by(year) %>%
  summarise(media_conversione2 = mean(conversion_coefficient, na.rm = TRUE))
```

ulteriori considerazioni:

```r
model <- lm(sales ~ coefficiente_conversione, data = RealEstateTexax_Dataframe)
summary(model)
```

```
##
## Call:
## lm(formula = sales ~ coefficiente_conversione, data = RealEstateTexax_Dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.926  -67.248   -8.434   49.235  209.651
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              107.2895    12.7420    8.42 3.56e-15 ***
## coefficiente_conversione   7.1584     0.9983    7.17 9.37e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 72.38 on 238 degrees of freedom
## Multiple R-squared:  0.1777, Adjusted R-squared:  0.1742
## F-statistic: 51.42 on 1 and 238 DF,  p-value: 9.374e-12
```

1. -> il p-value proposto permette di non confutare il legame che sussiste tra le due variabili, quindi effettivamente pubblicare annunci ha un effetto sulle vendite.
2. -> d altra parte considerando Multiple R-squared si può notare che solo il 17% circa della variabilità dei dati poò essere spiegata dal modello, di consegunza gli annunci hanno un'influenza relativamente bassa sulle vendite.

10) Prova a creare dei summary.

Oltre a quelli fatti nel punto 9:

```r
totale_venduto_città <- RealEstateTexax_Dataframe %>%
  group_by(city) %>%
  summarise(media_fatturato = mean(volume, na.rm = TRUE))

totale_venduto_città
```

```
## # A tibble: 4 x 2
##   city                  media_fatturato
##   <chr>                           <dbl>
## 1 Beaumont                         26.1
## 2 Bryan-College Station            38.2
## 3 Tyler                            45.8
## 4 Wichita Falls                    13.9
```

La città con maggiore fatturato è Tyler.