



Master in Data Science – Profession AI

Statistica descrittiva per Data Scientist

Progetto finale:

***Analisi esplorativa del mercato
immobiliare del Texas***

Studente: Virginio Cocciaglia Anno 2023 - 2024

DESCRIZIONE DEL PROGETTO

L'azienda **Texas Realty Insights** desidera analizzare le tendenze del mercato immobiliare nello stato del Texas, sfruttando i dati storici relativi alle vendite di immobili. L'obiettivo è fornire insight statistici e visivi che supportino le decisioni strategiche di vendita e ottimizzazione delle inserzioni immobiliari.

OBIETTIVI DEL PROGETTO

- Identificare e interpretare i trend storici delle vendite immobiliari nel Texas.
- Valutare l'efficacia delle strategie di marketing delle inserzioni immobiliari.
- Offrire una rappresentazione grafica dei dati che evidenzia la distribuzione dei prezzi e delle vendite tra città, mesi e anni.

VALORE AGGIUNTO

L'analisi statistica proposta permetterà a **Texas Realty Insights** di ottimizzare le loro strategie di mercato, identificando città con opportunità di crescita e valutando l'efficacia delle inserzioni immobiliari nel tempo. Grazie a una visione chiara e strutturata dei dati, l'azienda potrà prendere decisioni basate su informazioni concrete, migliorando la gestione delle vendite immobiliari in Texas.

ANALISI ESPLORATIVA DEL MERCATO IMMOBILIARE DEL TEXAS

PARTE 1

Quesito 1

Scarica il dataset “Real Estate Texas.csv” e importalo con R, questo contiene dei dati riguardanti le vendite di immobili in Texas. Le variabili del dataset sono:

1. **city**: città
2. **year**: anno di riferimento
3. **month**: mese di riferimento
4. **sales**: numero totale di vendite
5. **volume**: valore totale delle vendite in milioni di dollari
6. **median_price**: prezzo mediano di vendita in dollari
7. **listings**: numero totale di annunci attivi
8. **months_inventory**: quantità di tempo necessaria per vendere tutte le inserzioni correnti al ritmo attuale delle vendite, espresso in mesi.

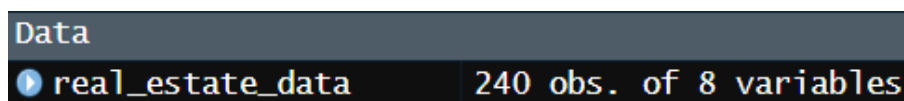
Soluzione Q1

Per importare il dataset “Real Estate Texas.csv” viene utilizzata la funzione **read.csv** che permette di leggere il file in formato tabella e di creare un dataframe da essa. Inoltre, una volta creato il dataframe, tramite la funzione **attach** si rende possibile accedere agli oggetti nel dataset semplicemente fornendo i loro nomi (Fig. 1.1).

```
real_estate_data <- read.csv("Real Estate Texas.csv")
attach(real_estate_data)
```

Fig. 1.1 Importazione del dataset con R e applicazione della funzione *attach*.

Viene dunque creato e salvato nell’ambiente globale di R un dataframe costituito da 240 osservazioni di 8 variabili (Fig. 1.2).



The screenshot shows the R IDE's 'Data' pane. It contains a single entry, 'real_estate_data', which is represented by a blue circular icon with a white 'r'. To the right of the icon, the text '240 obs. of 8 variables' is displayed, indicating the size and structure of the loaded dataset.

Fig. 1.2 Salvataggio del dataset come *data.frame* nell’ambiente globale di R.

Quesito 2

Indica il tipo di variabili contenute nel dataset.

Soluzione Q2

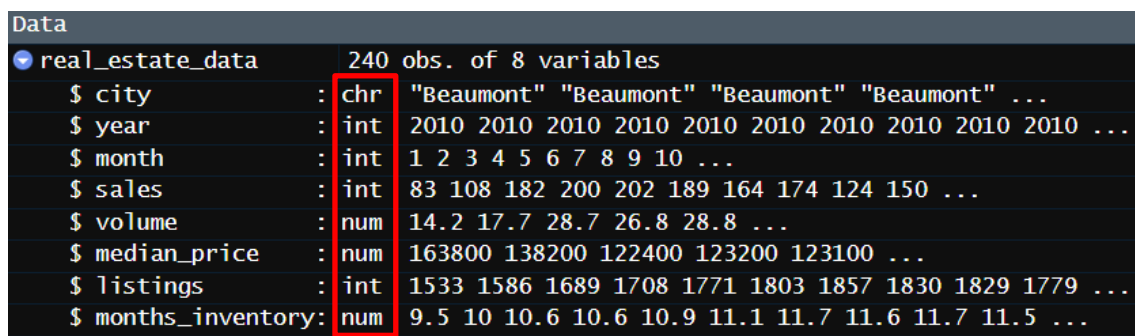
La variabile *city* è **qualitativa su scala nominale** poiché contiene in essa i nomi delle quattro città di riferimento che possono essere confrontate solo in termini di uguaglianza.

Le variabili *year* e *month* invece, nonostante il tempo sia da considerare come una variabile quantitativa continua, nel contesto del dataset possono essere considerate come **qualitative su scala ordinale** (gli anni ordinati dal 2010 al 2014 e i mesi da 1 a 12).

Le variabili *sales* e *listings* sono variabili **quantitative discrete** poiché assumono valori interi e rappresentano rispettivamente il conteggio del numero totale delle vendite e del numero totale di annunci attivi.

Infine, le variabili *volume*, *median_price* e *months_inventory* sono variabili **quantitative continue** (possono esserci numeri decimali oltre che interi).

In R, le tipologie di variabili presenti nel dataset possono essere visualizzate selezionando il dataframe salvato nell'ambiente globale. La tipologia è riportata subito dopo il nome della variabile (riquadro rosso nella Fig. 2.1).



The screenshot shows the R console output for the 'real_estate_data' dataframe. The first line indicates '240 obs. of 8 variables'. The subsequent lines list each variable with its type in parentheses. A red box highlights the 'chr' type for the 'city' variable.

Variable	Type	Sample Values
city	chr	"Beaumont" "Beaumont" "Beaumont" "Beaumont" ...
year	int	2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
month	int	1 2 3 4 5 6 7 8 9 10 ...
sales	int	83 108 182 200 202 189 164 174 124 150 ...
volume	num	14.2 17.7 28.7 26.8 28.8 ...
median_price	num	163800 138200 122400 123200 123100 ...
listings	int	1533 1586 1689 1708 1771 1803 1857 1830 1829 1779 ...
months_inventory	num	9.5 10 10.6 10.6 10.9 11.1 11.7 11.6 11.7 11.5 ...

Fig. 2.1 Tipologia delle variabili contenute nel dataset.

Dunque, la variabile *city* è di tipo **“character”** ossia è un oggetto di tipo stringa, le variabili *year*, *month*, *sales* e *listings* sono di tipo **“integer”** ossia sono numeri interi, e infine le variabili *volume*, *median_price* e *months_inventory* sono **“numeric”** ossia sono dei numeri reali (sia interi che decimali).

Quesito 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. Commenta tutto brevemente.

Soluzione Q3

Per prima cosa viene calcolata la **distribuzione di frequenze assolute** per ogni variabile per comprendere meglio i dati e capire per quali variabili ha senso calcolare gli indici. Per farlo si utilizza il comando **table** e la funzione **sapply** (Fig. 3.1). Questa funzione richiede come primo argomento la struttura dati su cui applicare la funzione per ogni elemento (in questo caso un dataframe) e come secondo argomento la funzione stessa (in questo caso table). Si ottiene così una **lista di frequenze assolute** per ognuna delle otto variabili.

```
abs_freq <- sapply(real_estate_data, table)
abs_freq
```

Fig. 3.1 Distribuzione di frequenze assolute per ciascuna variabile.

Prendendo in considerazioni le frequenze assolute delle variabili *city*, *year* e *month* (figura seguente 3.2) si vede come i dati delle vendite riguardano le quattro aree del Texas Beaumont, Bryan-College Station, Tyler e Wichita Falls e prendono in esame le annualità dal 2010 al 2014 ognuna suddivisa in mesi.

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

$year
2010 2011 2012 2013 2014
  48   48   48   48   48

$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
```

Fig. 3.2 Distribuzione di frequenze assolute delle variabili *city*, *year* e *month*.

Dunque, per queste tre variabili non ha senso calcolare nessun tipo di indice. La moda, ad esempio, sarebbe inutile da calcolare avendo lo stesso numero di dati per ogni città e per ogni anno. Si procede di seguito a calcolare gli indici di posizione, variabilità e forma delle restanti variabili.

Indici di posizione

● Minimo, massimo, media, mediana e quantili

Per ottenere i valori degli altri indici di posizione **minimo**, **massimo**, **media aritmetica**, **mediana** e **quantili** è possibile utilizzare la funzione **summary** (Fig. 3.5).

```
> summary(real_estate_data[4:8])
```

sales	volume	median_price	listings	months_inventory
Min. : 79.0	Min. : 8.166	Min. : 73800	Min. : 743	Min. : 3.400
1st Qu.:127.0	1st Qu.:17.660	1st Qu.:117300	1st Qu.:1026	1st Qu.: 7.800
Median :175.5	Median :27.062	Median :134500	Median :1618	Median : 8.950
Mean :192.3	Mean :31.005	Mean :132665	Mean :1738	Mean : 9.193
3rd Qu.:247.0	3rd Qu.:40.893	3rd Qu.:150050	3rd Qu.:2056	3rd Qu.:10.950
Max. :423.0	Max. :83.547	Max. :180000	Max. :3296	Max. :14.900

Fig. 3.3 Summary delle variabili: sales, volume, median_price, listings e months_inventory.

Inoltre, con la funzione **quantile** si possono ricavare ad esempio, oltre ai quantili, anche i **decili** e i **percentili** specificando l'intervallo di suddivisione con **seq** (Fig. 3.6).

```
deciles <- sapply(real_estate_data[4:8], quantile, probs=seq(0,1,0.1))
deciles

percentiles <- sapply(real_estate_data[4:8], quantile, probs=seq(0,1,0.01))
percentiles
```

Fig. 3.4 Decili e percentili tramite funzione quantile.

Nella figura 3.7 di seguito vengono mostrati i decili di ciascuna variabile.

```
> deciles <- sapply(real_estate_data[4:8], quantile, probs=seq(0,1,0.1))
> deciles
```

	sales	volume	median_price	listings	months_inventory
0%	79.0	8.1660	73800	743.0	3.40
10%	101.9	13.0967	99960	899.9	6.69
20%	120.6	16.1206	110000	968.0	7.50
30%	135.0	19.0344	121650	1208.7	7.97
40%	155.0	23.9976	130700	1525.2	8.40
50%	175.5	27.0625	134500	1618.5	8.95
60%	197.0	31.8436	141220	1687.8	9.40
70%	228.5	36.9307	147960	1796.0	10.53
80%	271.0	45.5920	152360	2721.4	11.40
90%	302.1	53.7391	158850	2946.7	12.21
100%	423.0	83.5470	180000	3296.0	14.90

Fig. 3.5 Decili delle variabili: sales, volume, median_price, listings e months_inventory.

In alternativa alla funzione **summary** potevano essere utilizzate le funzioni **min** e **max** per trovare i valori minimi e massimi, la funzione **mean** per calcolare la media aritmetica, la funzione **median** per trovare la mediana e la funzione **quantile** per i quantili.

Indici di variabilità

● Range o intervallo di variazione

Il **range** delle variabili può essere ricavato attraverso la funzione **range** (Fig. 3.8).

```
range <- sapply(real_estate_data[4:8], range)
range
```

```
> range <- sapply(real_estate_data[4:8], range)
> range
```

	sales	volume	median_price	listings	months_inventory
[1,]	79	8.166	73800	743	3.4
[2,]	423	83.547	180000	3296	14.9

Fig. 3.6 Range delle variabili: sales, volume, median_price, listings e months_inventory.

Con la funzione **range** si ottiene l'intervallo di variazione sotto forma di valore minimo e massimo della variabile. Se si vuole invece ottenere il valore dell'intervallo si possono semplicemente sottrarre questi due valori e creare una funzione che lo calcola (Fig. 3.9).

```
range_func <- function(variable){
  return(max(variable)-min(variable))
}

range_var <- sapply(real_estate_data[4:8], range_func)
range_var
```

```
> range_var <- sapply(real_estate_data[4:8], range_func)
> range_var
```

	sales	volume	median_price	listings	months_inventory
	344.000	75.381	106200.000	2553.000	11.500

Fig. 3.7 Range delle variabili: sales, volume, median_price, listings e months_inventory.

● Range interquartile

Allo stesso modo può essere calcolato il **range interquartile** con la funzione **IQR** (Fig. 3.10). In questo caso però si ottiene direttamente l'intervallo e non gli estremi dati al primo e dal terzo quartile.

```
IQR <- sapply(real_estate_data[4:8], IQR)
IQR
```

```
> IQR <- sapply(real_estate_data[4:8], IQR)
> IQR
```

	sales	volume	median_price	listings	months_inventory
	120.0000	23.2335	32750.0000	1029.5000	3.1500

Fig. 3.8 Range interquartile delle variabili: sales, volume, median_price, listings e months_inventory.

• Varianza

La **varianza** σ^2 considera la diversità delle unità dalla media aritmetica e può essere calcolata con la funzione **var** (Fig. 3.11).

```
sigma2 <- sapply(real_estate_data[4:8], var)
sigma2
```

```
> sigma2 <- sapply(real_estate_data[4:8], var)
> sigma2
```

sales	volume	median_price	listings	months_inventory
6.344300e+03	2.772707e+02	5.135730e+08	5.665690e+05	5.306889e+00

Fig. 3.9 Varianza delle variabili: sales, volume, median_price, listings e months_inventory.

• Deviazione standard

La **deviazione standard** σ (radice quadrata della varianza) fornisce un indice di variabilità nella stessa unità di misura dei dati osservati. Essa può essere calcolata in R tramite la funzione **sd** (Fig. 3.12).

```
sigma <- sapply(real_estate_data[4:8], sd)
sigma
```

```
> sigma <- sapply(real_estate_data[4:8], sd)
> sigma
```

sales	volume	median_price	listings	months_inventory
79.651111	16.651447	22662.148687	752.707756	2.303669

Fig. 3.10 Deviazione standard delle variabili: sales, volume, median_price, listings e months_inventory.

Dunque, le deviazioni standard ottenute per ciascuna variabile sono: **79.65 vendite**, **16.65 mln di \$**, **22 662.15 \$**, **752.71 annunci attivi** e **2.3 mesi**.

• Coefficiente di variazione

Infine, si calcola il **coefficiente di variazione CV** utile a confrontare le variabilità di diverse variabili. Per calcolarlo in R è necessario creare un'apposita funzione (Fig. 3.13).

```
CV <-function(variable){
  return( sd(variable)/mean(variable) * 100 )
}

CV_var <- sapply(real_estate_data[4:8], CV)
CV_var
```

```
> CV_var <- sapply(real_estate_data[4:8], CV)
> CV_var
```

sales	volume	median_price	listings	months_inventory
41.42203	53.70536	17.08218	43.30833	25.06031

Fig. 3.11 Coefficienti di variazione.

Indici di forma: asimmetria e curtosi

● Indice di asimmetria di Fisher

Per calcolare l'indice di asimmetria di Fisher γ_1 si può utilizzare la libreria **moments** e la funzione **skewness** (Fig. 3.14).

```
install.packages("moments")
library(moments)

gamma1 <- skewness(real_estate_data[4:8])
gamma1
```

```
> gamma1 <- skewness(real_estate_data[4:8])
> gamma1
```

	sales	volume	median_price	listings	months_inventory
	0.71810402	0.88474203	-0.36455288	0.64949823	0.04097527

Fig. 3.12 Indice di Fisher delle variabili: sales, volume, median_price, listings e months_inventory.

Osservando gli indici di Fisher ottenuti si può dedurre come tutte le variabili abbiano una *distribuzione asimmetrica positiva* ($\gamma_1 > 0$) tranne il prezzo mediano di vendite che ha una *distribuzione asimmetrica negativa* ($\gamma_1 < 0$).

● Coefficiente di curtosi

Anche per il calcolo del **coefficiente di curtosi** può essere utilizzata la libreria **moments** e in questo caso il comando è **kurtosis** (Fig. 3.15).

```
gamma2 <- kurtosis(real_estate_data[4:8]) - 3
gamma2
```

```
> gamma2 <- kurtosis(real_estate_data[4:8]) - 3
> gamma2
```

	sales	volume	median_price	listings	months_inventory
	-0.3131764	0.1769870	-0.6229618	-0.7917900	-0.1744475

Fig. 3.13 Coefficiente di curtosi delle variabili: sales, volume, median_price, listings e months_inventory.

Per ottenere il coefficiente di curtosi bisogna sottrarre tre al valore ottenuto con la funzione **kurtosis** per centrarlo in zero.

Osservando i coefficienti di curtosi ottenuti si può dedurre come tutte le variabili abbiano una *distribuzione platicurtica* ($\gamma_2 < 0$) tranne il volume che ha una *distribuzione leptocurtica* ($\gamma_2 > 0$).

Quesito 4

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

Soluzione Q4

Per trovare la variabile con variabilità più elevata è sufficiente osservare i coefficienti di variazioni calcolati in precedenza nella risposta al quesito 3 e vedere quale ha il valore maggiore.

```
> CV_var <- sapply(real_estate_data[4:8], CV)
> CV_var
```

sales	volume	median_price	listings	months_inventory
41.42203	53.70536	17.08218	43.30833	25.06031

La variabile con variabilità più elevata è **volume** (valore totale delle vendite in milioni di dollari) che ha una **deviazione standard pari a circa il 53.71 % della media**.

Per trovare la variabile più asimmetrica invece si possono osservare gli indici di asimmetria di Fisher calcolati sempre nel punto 3 e vedere quale ha il valore maggiore.

```
> gamma1 <- skewness(real_estate_data[4:8])
> gamma1
```

sales	volume	median_price	listings	months_inventory
0.71810402	0.88474203	-0.36455288	0.64949823	0.04097527

La variabile più asimmetrica (asimmetria positiva) è **volume** che ha un **indice di asimmetria di Fisher di circa 0.88**.

Quesito 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.

Soluzione Q5

La variabile quantitativa scelta da suddividere in classi è il **prezzo mediano di vendita**. Come primo passaggio si studiano gli indici di posizione della variabile (calcolati in precedenza) per cercare di comprendere la maniera migliore per suddividere i valori (Fig. 5.1)

```
> summary(median_price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 73800 117300 134500 132665 150050 180000
```

Fig. 5.1 Summary della variabile *median_price*.

La variabile si muove in un range di valori che va da 73 800 \$ a 180 000 \$ e può risultare ragionevole e conveniente suddividere i dati in **4 classi**, ognuna con la stessa ampiezza di **30 000 \$**, a partire dal valore di **70 000 \$** fino a **190 000 \$**. L'obiettivo infatti è quello di avere una tabella né troppo sparsa e né troppo densa che sia rappresentativa dell'andamento della variabile.

Per effettuare questa suddivisione in R si può utilizzare la funzione **cut** che associa ogni valore al suo intervallo di pertinenza, specificato tramite un vettore passato all'argomento *breaks* (Fig. 5.2).

```
summary(median_price)

median_price_cl <- cut(median_price,
                       breaks = c(70000,100000,130000,160000,190000))
```

Fig. 5.2 Suddivisione dei valori della variabile *median_price* in 4 classi di uguale ampiezza.

Una volta create le classi, si procede alla costruzione della **tabella delle distribuzioni di frequenza** (Fig. 5.3). Le **frequenze assolute** n_i dei dati suddivisi in classi si ottengono sempre con la funzione **table** vista in precedenza, per ottenere le **frequenze relative** f_i invece è sufficiente dividere le frequenze assolute per il numero totale di elementi N .

Le **frequenze cumulate** N_i invece si ottengono attraverso la funzione di R **cumsum** (utilizzando come argomento le frequenze assolute) e le **frequenze relative cumulate** F_i dividendo N_i per N .

Infine, per costruire la tabella delle distribuzioni di frequenze si usa la funzione **cbind** che combina tutte le distribuzioni.

```
N <- length(median_price)
ni <- table(median_price_cl)
fi <- ni/N
Ni <- cumsum(ni)
Fi <- Ni/N
distr_freq_median_price_cl <- cbind(ni,fi,Ni,Fi)
distr_freq_median_price_cl
```

	ni	fi	Ni	Fi
(7e+04,1e+05]	26	0.1083333	26	0.1083333
(1e+05,1.3e+05]	69	0.2875000	95	0.3958333
(1.3e+05,1.6e+05]	124	0.5166667	219	0.9125000
(1.6e+05,1.9e+05]	21	0.0875000	240	1.0000000

Fig. 5.3 Tabella delle distribuzioni di frequenze per la variabile `median_price`.

Grafici a barre

Una volta ottenute le distribuzioni di frequenze assolute, relative, cumulate e relative cumulate è possibile creare un **grafico a barre** per ognuna di esse tramite la funzione **`barplot`** (Fig. 5.4).

```
bp_ni <- barplot(main = "Distribuzione delle frequenze assolute del prezzo mediano",
  ni,
  xlab = "Classi di prezzo mediano di vendita in $",
  ylab = "Frequenze assolute",
  names.arg = c("70k-100k", "100k-130k", "130k-160k", "160k-190k"),
  ylim = c(0,140),
  col = "cyan3",
  border = "black")
text(x = bp_ni, y = ni, labels = ni, col = "black", pos = 1, cex = 1.1)

bp_fi <- barplot(main = "Distribuzione delle frequenze relative del prezzo mediano",
  fi,
  xlab = "Classi di prezzo mediano di vendita in $",
  ylab = "Frequenze relative",
  names.arg = c("70k-100k", "100k-130k", "130k-160k", "160k-190k"),
  ylim = c(0,1),
  col = "blue2",
  border = "black")
text(x = bp_fi, y = fi, labels = round(fi, 2), col = "white", pos = 1, cex = 1.1)

bp_Ni <- barplot(main = "Distribuzione delle frequenze cumulate del prezzo mediano",
  Ni,
  xlab = "Classi di prezzo mediano di vendita in $",
  ylab = "Frequenze cumulate",
  names.arg = c("70k-100k", "100k-130k", "130k-160k", "160k-190k"),
  ylim = c(0,250),
  col = "orange2",
  border = "black")
text(x = bp_Ni, y = Ni, labels = Ni, col = "black", pos = 1, cex = 1.1)

bp_Fi <- barplot(main = "Distribuzione delle frequenze relative cumulate del prezzo",
  Fi,
  xlab = "Classi di prezzo mediano di vendita in $",
  ylab = "Frequenze relative cumulate",
  names.arg = c("70k-100k", "100k-130k", "130k-160k", "160k-190k"),
  ylim = c(0,1),
  col = "red3",
  border = "black")
text(x = bp_Fi, y = Fi, labels = round(Fi, 2), col = "white", pos = 1, cex = 1.1)
```

Fig. 5.4 Grafici a barre delle distribuzioni di frequenze assolute, relative, cumulate e relative cumulate.

Di seguito vengono mostrati i grafici delle distribuzioni di frequenze ottenuti (figure 5.5, 5.6, 5.7 e 5.8).

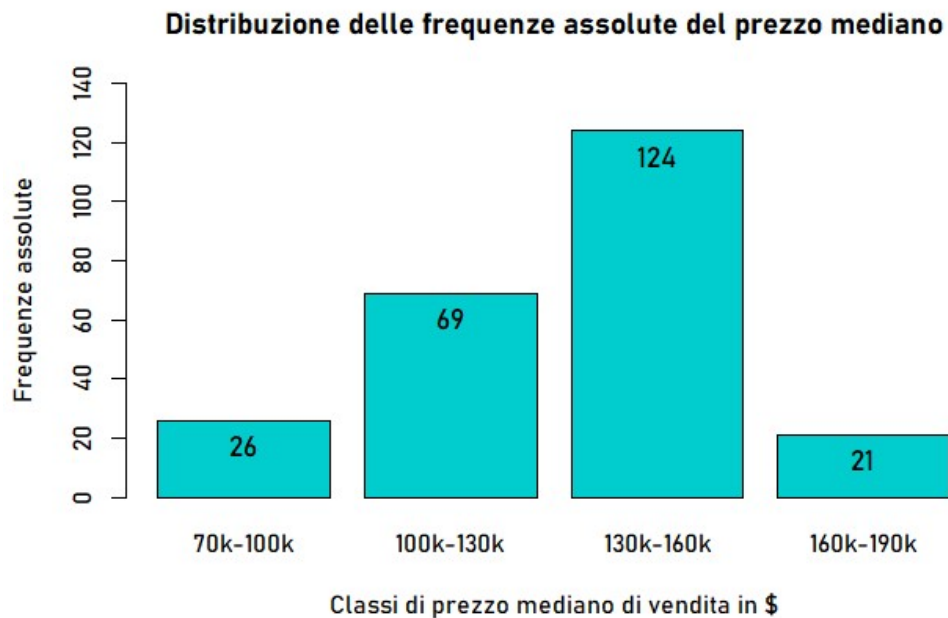


Fig. 5.5 Distribuzione delle frequenze assolute del prezzo mediano di vendita.

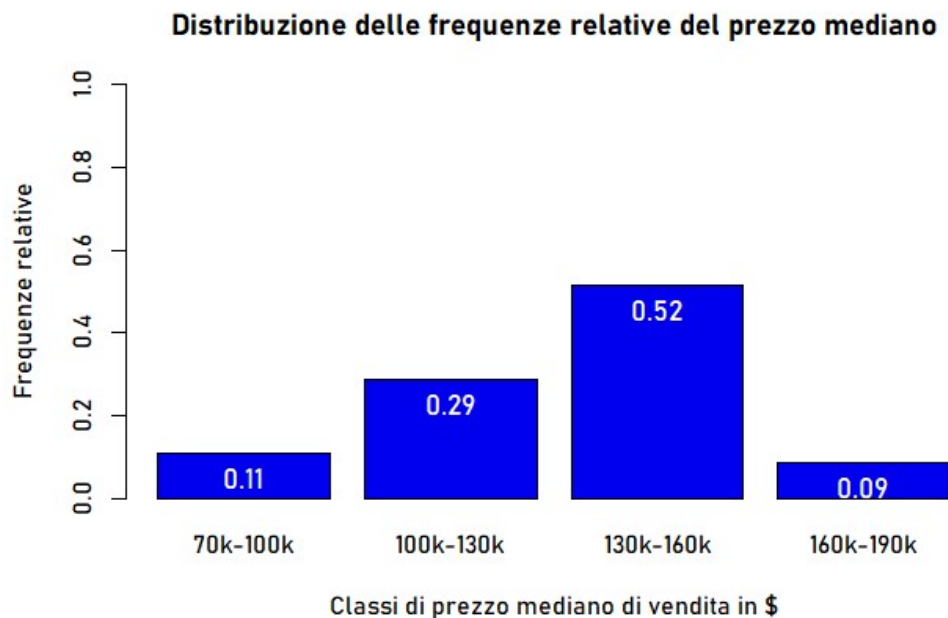


Fig. 5.6 Distribuzione delle frequenze relative del prezzo mediano di vendita.

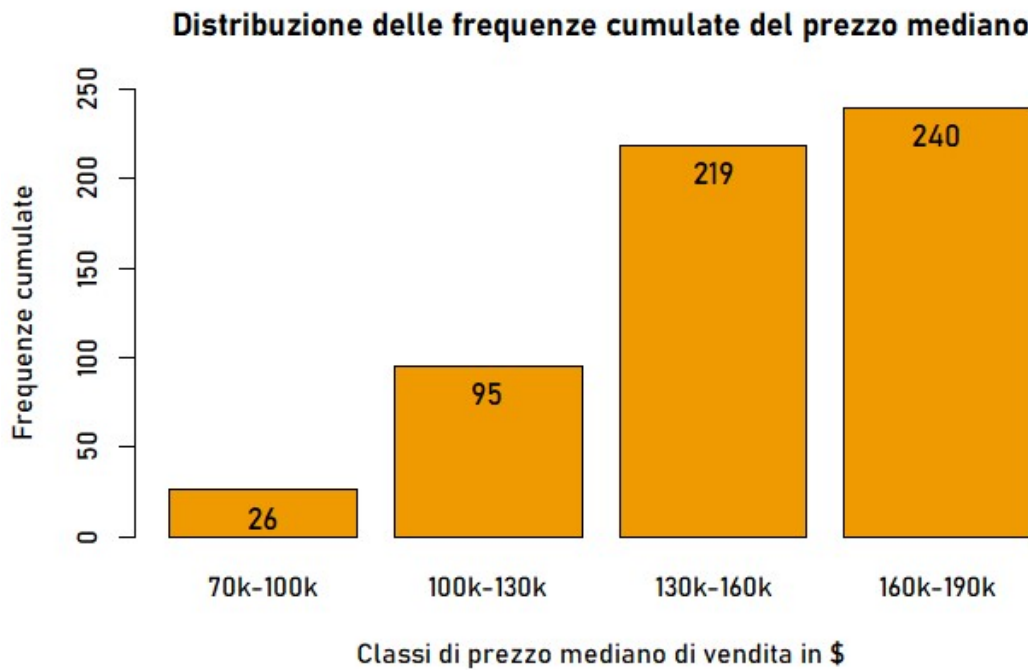


Fig. 5.7 Distribuzione delle frequenze cumulate del prezzo mediano di vendita.

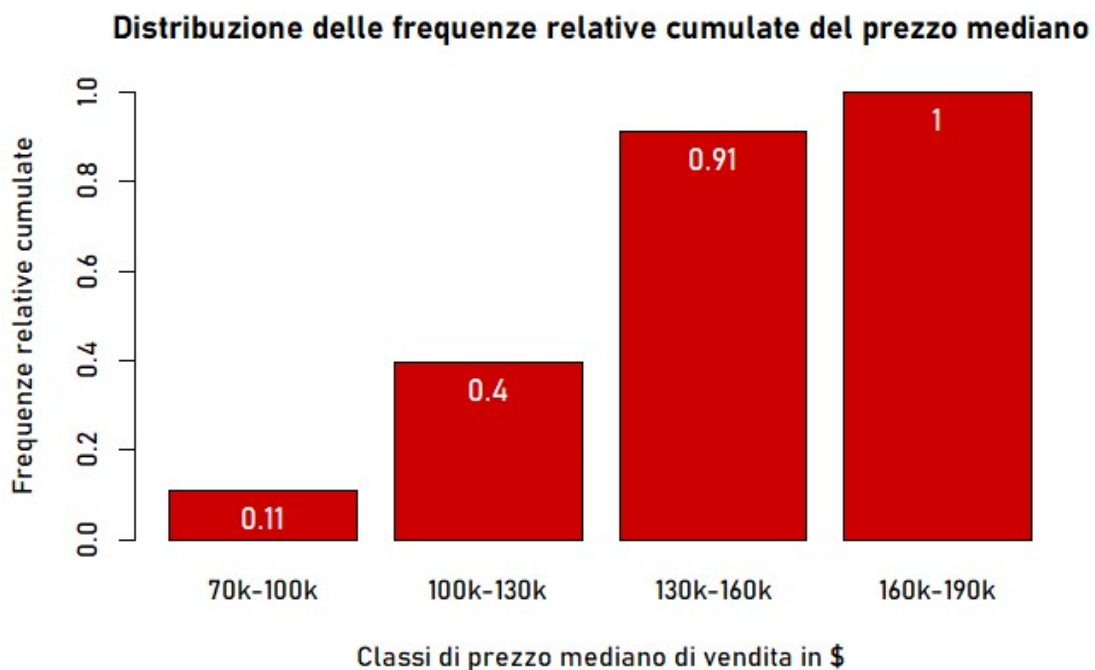


Fig. 5.8 Distribuzione delle frequenze relative cumulate del prezzo mediano di vendita.

Indice di Gini

Infine, si calcola l'**Indice di eterogeneità di Gini** del prezzo mediano di vendita che misura la propensione della variabile ad assumere le sue diverse modalità, in questo caso le quattro classi in cui è stata suddivisa. L'indice dipende dunque dalle frequenze relative f_i calcolate in precedenza, ed in particolare dal loro quadrato, e dal numero di modalità possibili, ossia quattro. In Fig. 5.9 viene mostrato il calcolo dell'indice di Gini effettuato con R.

```
J <- length(table(median_price_cl))
G <- 1-sum(fi^2)
gini_index <- G/((J-1)/J)
gini_index

> gini_index <- G/((J-1)/J)
> gini_index
[1] 0.8413426
```

Fig. 5.9 Indice di Gini del prezzo mediano di vendita suddiviso in classi.

Il valore dell'indice di Gini ottenuto è di circa **0.84** e si può quindi dedurre che il prezzo mediano di vendita è abbastanza equidistribuito nelle sue quattro classi. Un indice di Gini pari a 1 infatti rappresenta la massima eterogeneità mentre un indice pari a 0 eterogeneità nulla.

Quesito 6

Indovina l'indice di Gini per la variabile city.

Soluzione Q6

Si può trovare l'**indice di Gini** per la variabile *city* senza bisogno di effettuare nessun calcolo semplicemente osservando le frequenze assolute calcolate in risposta al Q3.

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

Si può notare infatti come le quattro città presenti nel dataset abbiano lo stesso numero di dati, coprendo lo stesso arco temporale, e quindi l'indice di eterogeneità di Gini è pari ad **1** rientrando nella casistica di **eterogeneità massima (equidistribuzione)**.

Quesito 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?

Soluzione Q7

La **probabilità** che presa una riga a caso del dataset essa riporti la città “Beaumont” è di **0.25** o **25%** alla luce di quanto detto in risposta al quesito 6.

Il calcolo della probabilità può però essere effettuato anche con R secondo l’accezione classica facendo l’operazione: **n° di casi favorevoli all’evento / numero di casi possibili**. Il numero di casi possibili è ottenuto con la funzione **length** e corrisponde al numero totale di righe del dataset (240 per ogni variabile) mentre i casi possibili sono ottenuti grazie alla funzione **sum**, con argomento **city == “Beaumont”**, che conteggia tutte le volte che la città “Beaumont” è all’interno della colonna **city**.

Con lo stesso procedimento vengono calcolate la probabilità di trovare il mese di luglio e il mese di dicembre 2012 utilizzando come argomento della funzione **sum** rispettivamente **month == “7”** e **month == “12” & year == “2012”**.

```
#città: Beaumont
poss_cases <- length(real_estate_data[,1])
fav_cases_Beau <- sum(city == "Beaumont")
p_Beaumont <- fav_cases_Beau/poss_cases
p_Beaumont

#mese: luglio
fav_cases_july <- sum(month == "7")
p_july <- fav_cases_july/poss_cases
p_july

#mese e anno: dicembre 2012
fav_cases_dec_12 <- sum(month == "12" & year == "2012")
p_dec_12 <- fav_cases_dec_12/poss_cases
p_dec_12
```

p_Beaumont	0.25
p_dec_12	0.0166666666666667
p_july	0.0833333333333333

Fig. 7.1 Probabilità calcolate con R.

In definitiva si ottengono le seguenti probabilità: probabilità “città Beaumont” **25%**; probabilità “mese di luglio” **8.33%**; probabilità “mese di dicembre 2012” **1.67%**.

Quesito 8

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

§ 1.2 Soluzione Q8

Il **prezzo medio di vendita** si può ottenere dividendo il valore totale delle vendite in milioni di dollari rappresentato dalla variabile **volume** per il numero totale di vendite dato dalla variabile **sales**. È importante notare che per ottenere il prezzo medio di vendita in dollari è necessario moltiplicare i valori totali delle vendite per un milione.

Per creare una nuova colonna con i prezzi medi nel dataframe esistente *real_estate_data* viene utilizzata la notazione del dollaro (\$) seguita dal nome della nuova variabile (**average_price**) e ad essa viene assegnato il **rapporto tra il volume totale e il numero di vendite** (Fig. 8.1).

```
real_estate_data$avarage_price <- (volume*10^6)/sales
```

real_estate_data		240 obs. of 9 variables									
\$ city	: chr	"Beaumont"	"Beaumont"	"Beaumont"	"Beaumont"	...					
\$ year	: int	2010	2010	2010	2010	2010	2010	2010	2010	2010	...
\$ month	: int	1	2	3	4	5	6	7	8	9	10 ...
\$ sales	: int	83	108	182	200	202	189	164	174	124	150 ...
\$ volume	: num	14.2	17.7	28.7	26.8	28.8	...				
\$ median_price	: num	163800	138200	122400	123200	123100	...				
\$ listings	: int	1533	1586	1689	1708	1771	1803	1857	1830	1829	1779 ...
\$ months_inventory	: num	9.5	10	10.6	10.6	10.9	11.1	11.7	11.6	11.7	11.5 ...
\$ average_price	: num	170627	163796	157698	134095	142738	...				

Fig. 8.1 Creazione della nuova colonna con il prezzo medio nel dataset esistente.

Quesito 9

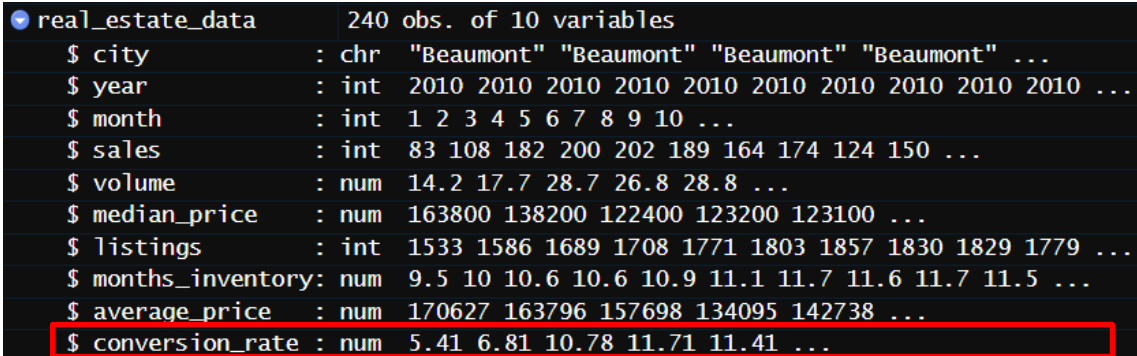
Prova a creare un'altra colonna che dia un'idea di "efficacia" degli annunci di vendita. Riesci a fare qualche considerazione?

Soluzione Q9

L'efficacia degli annunci di vendita può essere apprezzata ad esempio creando una nuova variabile (**conversion_rate**) che calcola il **rapporto in percentuale tra il numero totale di vendite e il numero totale di annunci attivi**. In questo modo è possibile avere una stima di quanti annunci sono stati convertiti in vendite.

La nuova colonna viene creata nel dataframe nello stesso modo in cui è stata creata in precedenza la colonna del prezzo medio (Fig. 9.1).

```
real_estate_data$conversion_rate <- (sales/listings)*100
```



```

real_estate_data      240 obs. of 10 variables
 $ city               : chr  "Beaumont" "Beaumont" "Beaumont" "Beaumont" ...
 $ year              : int   2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
 $ month             : int    1  2  3  4  5  6  7  8  9 10 ...
 $ sales             : int   83 108 182 200 202 189 164 174 124 150 ...
 $ volume            : num  14.2 17.7 28.7 26.8 28.8 ...
 $ median_price      : num 163800 138200 122400 123200 123100 ...
 $ listings          : int  1533 1586 1689 1708 1771 1803 1857 1830 1829 1779 ...
 $ months_inventory : num   9.5 10 10.6 10.6 10.9 11.1 11.7 11.6 11.7 11.5 ...
 $ average_price     : num 170627 163796 157698 134095 142738 ...
 $ conversion_rate   : num   5.41  6.81 10.78 11.71 11.41 ...

```

Fig. 9.1 Creazione della nuova colonna con il tasso di conversione degli annunci nel dataset esistente.

Se si vuole analizzare rapidamente la nuova variabile per fare qualche considerazione su di essa è possibile sempre utilizzare la funzione *summary* (Fig. 9.2). Prima di fare ciò però è bene applicare nuovamente la funzione *attach* per comprendere anche le due nuove variabili create (*average_price* e *conversion_rate*).

```
> summary(conversion_rate)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
5.014	8.980	10.963	11.874	13.492	38.713

Fig. 9.2 Summary della variabile *conversion_rate*.

Osservando gli indici di posizione ottenuti con *summary* si può vedere come il tasso di conversione degli annunci in vendite massimo ottenuto è di circa il **39%**, il tasso minimo del **5%** e il tasso mediano e medio rispettivamente del **10%** e dell'**11%**.

Quesito 10

Prova a creare dei *summary()*, o semplicemente media e deviazione standard, di alcune variabili a tua scelta, condizionatamente alla città, agli anni e ai mesi. Puoi utilizzare il linguaggio R di base oppure essere un vero Pro con il pacchetto *dplyr*.

Soluzione Q10

La libreria di R **dplyr** permette di manipolare i dati tramite funzioni di filtraggio, ordinamento, aggregazione e trasformazione. È importante notare che dplyr utilizza l'operator *pipe* (`%>%`) per concatenare le operazioni.

Con la funzione **group_by** di *dplyr* è possibile scegliere una o più variabili secondo cui raggruppare i dati mentre con il comando **summarise** si possono calcolare varie statistiche su di essi e raccoglierle in un dataframe.

Si può scegliere ad esempio di **raggruppare la variabile volume per città** e creare un **summary** (minimo e massimo, primo e terzo quartile, media e mediana) (Fig. 10.1). In questo modo si possono effettuare rapidamente dei confronti sul valore totale delle vendite in ogni città. Ad esempio, la città in cui si è raggiunto il massimo volume delle vendite in un mese è *Bryan-College Station* con **83.5 milioni di \$**.

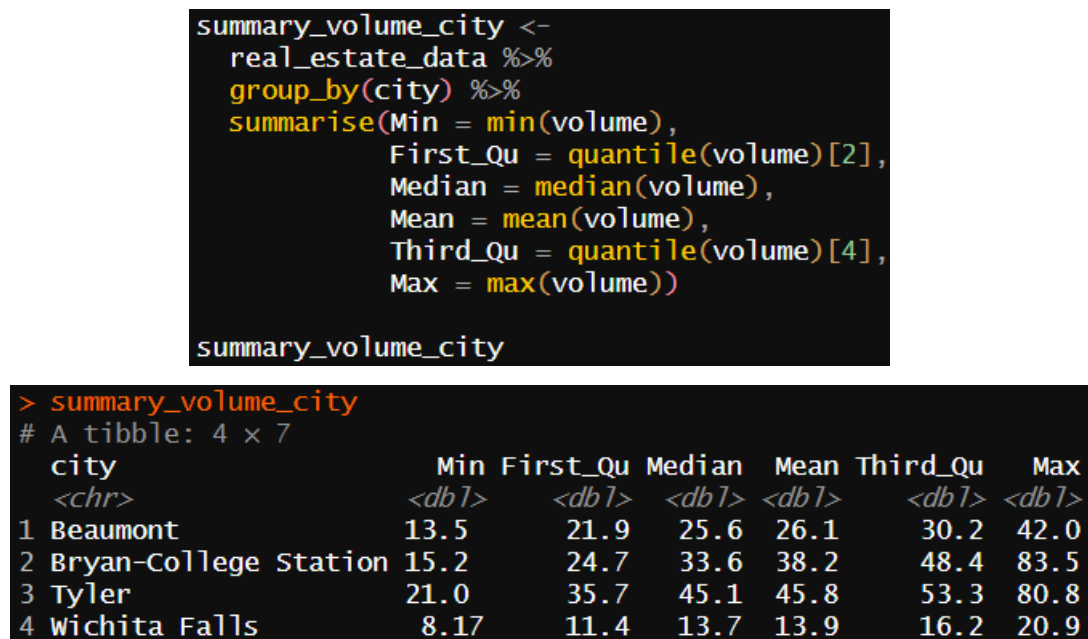


Fig. 10.1 Summary della variabile volume raggruppata per città.

Allo stesso modo si può **raggruppare la variabile listings per anno** e la **variabile average_price per mese** inserendo le corrette variabili di riferimento in *group_by* e *summarise* (Fig. 10.2 e Fig. 10.3). Si può ad esempio notare come il minor numero di annunci attivi totali in un mese si è avuto nel 2013 (**743 annunci attivi**) mentre il prezzo medio di vendita massimo è stato raggiunto nel mese di ottobre (**213 234 \$**).

```
> summary_listings_year
# A tibble: 5 × 7
  year   Min First_Qu Median   Mean Third_Qu   Max
<int> <int>   <dbl> <dbl> <dbl>   <dbl> <int>
1  2010   904   1230.  1630.  1826    2170.  3296
2  2011   844   1284.  1706.  1850.    2064.  3266
3  2012   801   1317.  1670.  1777.    2034.  3072
4  2013   743   1024.  1590.  1678.    1938.  2998
5  2014   746    970   1386.  1560.    1822.  2875
```

Fig. 10.2 Summary della variabile volume raggruppata per città.

```
> summary_average_price_month
# A tibble: 12 × 7
  month   Min First_Qu Median   Mean Third_Qu   Max
<int> <dbl>   <dbl> <dbl> <dbl>   <dbl> <dbl>
1     1 97848. 122301. 150107. 145640. 164702. 193796.
2     2 103099. 125894. 153255. 148840. 164476. 194889.
3     3 105126. 141728. 154818. 151137. 165769. 183287.
4     4 109065. 132110. 149347. 151461. 168728. 199937.
5     5 119398. 136326. 157273. 158235. 177218. 202425.
6     6 117396. 146125. 161139. 161546. 175163. 206851.
7     7 107039. 134958. 163629. 156881. 176547. 207313.
8     8 110193. 138506. 159392. 156456. 171936. 203487.
9     9 109227. 130567. 158103. 156522. 184020. 207941.
10    10  97010. 134203. 158121. 155897. 179980. 213234.
11    11 111794. 124231. 153240. 154233. 175595. 206527.
12    12 102783. 136104. 156541. 154996. 173222. 212765
```

Fig. 10.3 Summary della variabile average_price raggruppata per mese.

Inoltre, come detto in precedenza, con la funzione `group_by` è possibile anche raggruppare i dati secondo più di una variabile. Ad esempio, si può ottenere un summary della **variabile sales** raggruppandola per città e anno (Fig. 10.4). In questa maniera è possibile notare come nella città di *Beaumont* il numero massimo di vendite in un mese è stato ottenuto nel 2013 (**273 vendite**) mentre nella città di *Tyler* il numero massimo si è registrato nel 2014 (**423 vendite**).

```
summary_sales_city_year <-
  real_estate_data %>%
  group_by(city, year) %>%
  summarise(Min = min(sales),
            First_Qu = quantile(sales)[2],
            Median = median(sales),
            Mean = mean(sales),
            Third_Qu = quantile(sales)[4],
            Max = max(sales))

summary_sales_city_year
```

```
> summary_sales_city_year
```

A tibble: 20 × 8

Groups: city [4]

	city	year	Min	First_Qu	Median	Mean	Third_Qu	Max
	<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
1	Beaumont	2010	83	142	157	156.	184.	202
2	Beaumont	2011	108	126.	148.	144	161.	177
3	Beaumont	2012	110	162.	176.	172.	185.	218
4	Beaumont	2013	140	175.	202	201.	218.	273
5	Beaumont	2014	148	185	210	214.	248	262
6	Bryan-College Station	2010	89	106.	153	168.	205.	286
7	Bryan-College Station	2011	94	122.	148.	167.	206.	284
8	Bryan-College Station	2012	115	145.	161	197.	292.	296
9	Bryan-College Station	2013	125	166.	188.	238.	331.	402
10	Bryan-College Station	2014	152	193.	246.	260.	316.	403
11	Tyler	2010	155	197.	229	228.	257.	316
12	Tyler	2011	143	206.	247	239.	273.	313
13	Tyler	2012	169	232.	276	264.	292.	322
14	Tyler	2013	197	250.	288	287.	328.	369
15	Tyler	2014	238	296.	340.	332.	370.	423
16	Wichita Falls	2010	89	102.	123	123.	136.	167
17	Wichita Falls	2011	79	90	111	106.	121	135
18	Wichita Falls	2012	90	101.	116.	112.	124.	132
19	Wichita Falls	2013	79	99.2	122.	121.	145.	159
20	Wichita Falls	2014	89	100.	111	117	138.	150

Fig. 10.4 Summary della variabile sales raggruppata per città e anno.

PARTE 2

Da qui in poi utilizza ggplot2 per creare grafici fantastici! Ma non fermarti alla semplice soluzione del quesito, prova un po' a personalizzare i grafici utilizzando temi, colori e annotazioni, e aggiustando i vari elementi come le etichette, gli assi e la legenda.

Consiglio: Fai attenzione quando specifichi le variabili month e year tra le estetiche, potrebbe essere necessario considerarle come fattori.

Quesito 1

Utilizza i boxplot per confrontare la distribuzione del prezzo mediano delle case tra le varie città. Commenta il risultato.

Soluzione Q1

I **boxplot** sono dei diagrammi in grado di rappresentare contemporaneamente gli indici di posizione e la variabilità di una serie di dati. Essi sono costituiti da una linea centrale che solitamente rappresenta la mediana, un rettangolo (scatola) che rappresenta il range interquartile, altre due linee che rappresentano il resto dei valori fino al massimo e al minimo e alcuni punti singoli che rappresentano gli outliers.

Dei *boxplot* per confrontare la **distribuzione del prezzo medio delle case tra le varie città** possono essere costruiti utilizzando la geometria *geom_boxplot* della libreria *ggplot2* (libreria in grado di creare grafici utilizzando più strati sovrapposti di componenti grafici) (Fig. 1.1). Nelle estetiche (*aes*) vengono inserite le città sull'asse x del grafico, il prezzo medio sull'asse y e *lightblue2* come colore di riempimento (*fill*).

```
ggplot(real_estate_data) +
  geom_boxplot(aes(
    x = city,
    y = median_price),
    fill = "lightblue2") +
  labs(title = "Boxplot dei prezzi medi di vendita per città",
    x = "Città",
    y = "Prezzo mediano di vendita in $") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```

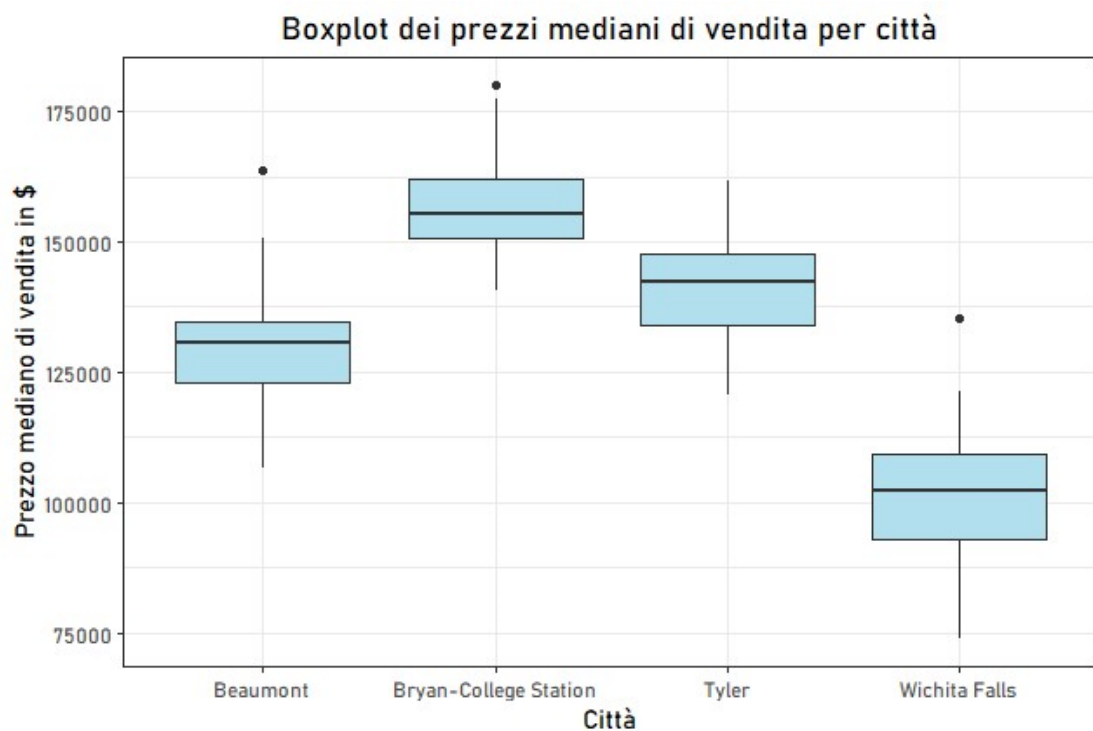


Fig. 1.1 Boxplot dei prezzi medi di vendita per città.

Dai boxplot nella figura precedente si può ad esempio notare come la città con i prezzi medi di vendita più elevati è *Bryan-College Station* mentre quella con i prezzi medi di vendita più bassi è *Wichita Falls*. Inoltre, la città di *Tyler* è l'unica che non presenta nessun valore anomalo (*outliers*) ossia punti esterni all'intervallo $[Q1 - 1.5 \times IQR ; Q3 + 1.5 \times IQR]$.

Quesito 2

Utilizza i boxplot o qualche variante per confrontare la distribuzione del valore totale delle vendite tra le varie città ma anche tra i vari anni. Qualche considerazione da fare?

Soluzione Q2

Per confrontare la **distribuzione del valore totale delle vendite tra le varie città ma anche tra i vari anni** è possibile utilizzare sempre dei *boxplot* ma in questo caso è necessario inserire nelle estetiche sull'asse x le città, sull'asse y il valore totale delle vendite e **come riempimento (fill) gli anni** trasformati in fattori tramite la funzione *factor*. In questo modo la variabile *year* viene considerata come qualitativa ordinale e ogni boxplot viene riempito con diversi colori a seconda dell'anno di riferimento.

Allo stesso modo può essere costruito un *boxplot* in grado di ottenere lo stesso risultato con gli anni trasformati in fattori sull'asse x, il valore totale delle vendite sull'asse y e la **variabile city come riempimento**. In Fig. 2.1 e 2.2 è mostrato il codice utilizzato per costruire i due boxplot mentre in Fig. 2.3 e 2.4 i grafici ottenuti.

```
ggplot(real_estate_data) +
  geom_boxplot(aes(
    x = city,
    y = volume,
    fill = factor(year))) +
  labs(title = "Boxplot del valore totale delle vendite per città e anno",
    x = "Città",
    y = "Valore totale delle vendite in milioni di $",
    fill = "Anno") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5),
    legend.position = "bottom")
```

Fig. 2.1 Costruzione dei boxplot delle vendite totali per città e anno.

```
ggplot(real_estate_data) +
  geom_boxplot(aes(
    x = factor(year),
    y = volume,
    fill = city)) +
  labs(title = "Boxplot del valore totale delle vendite per anno e città",
    x = "Anno",
    y = "Valore totale delle vendite in milioni di $",
    fill = "Città") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5),
    legend.position = "bottom")
```

Fig. 2.2 Costruzione dei boxplot delle vendite totali per anno e città.

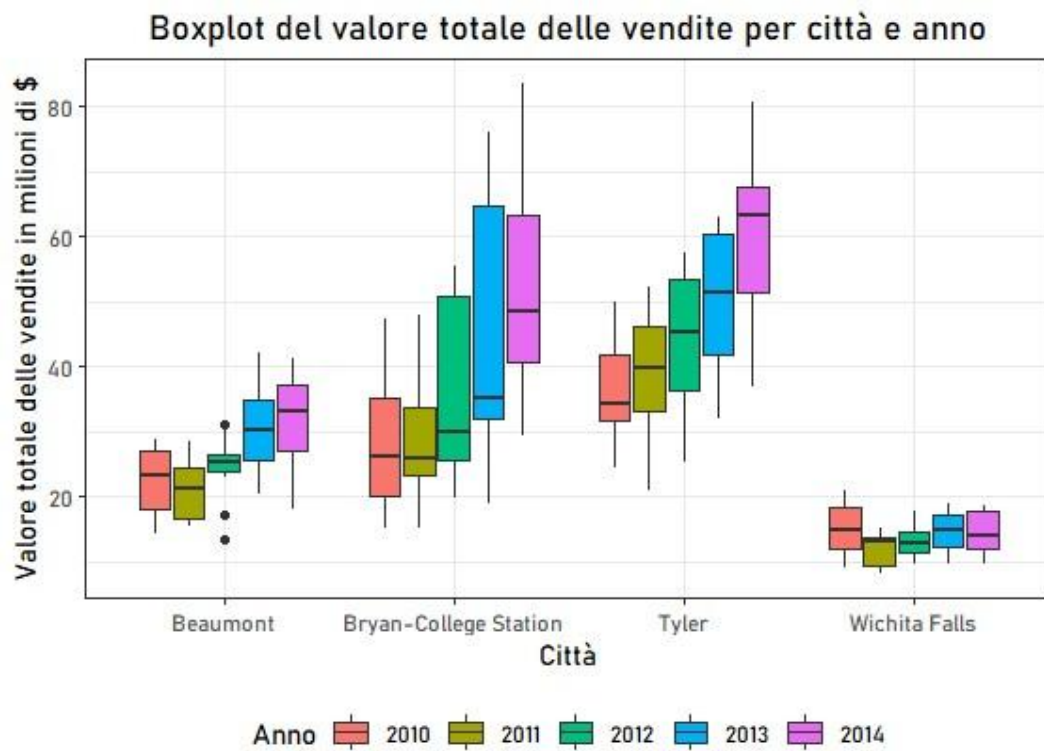


Fig. 2.3 Boxplot del valore totale delle vendite per città e anno.

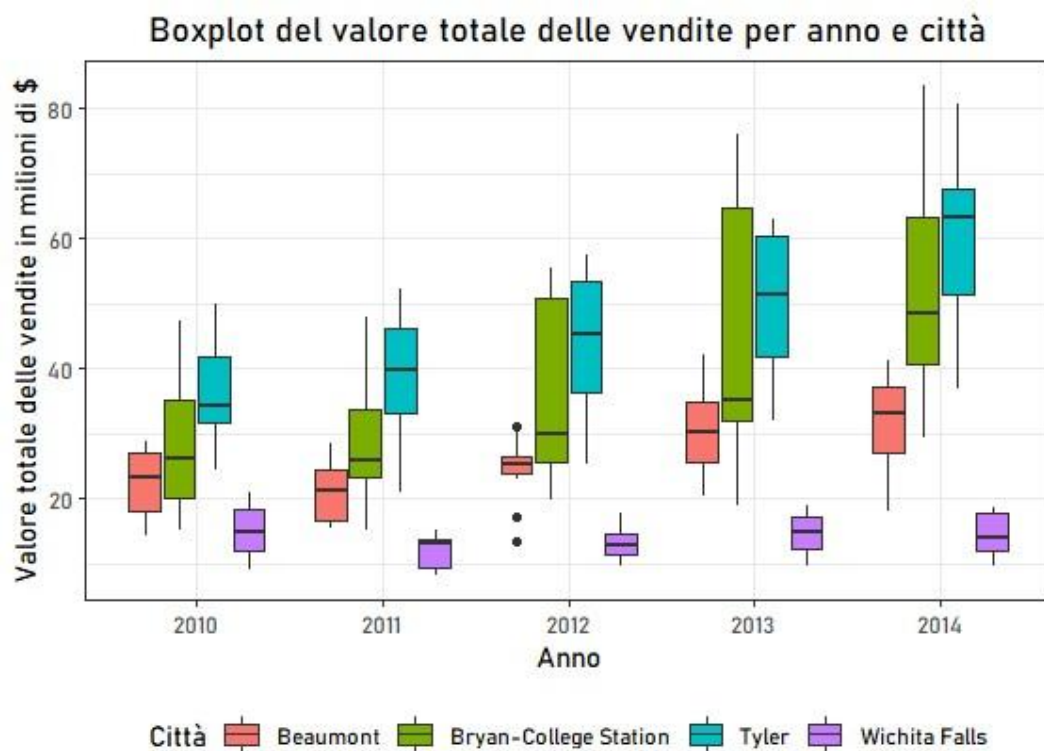


Fig. 2.4 Boxplot del valore totale delle vendite per anno e città.

Entrambi i precedenti boxplot confrontano la distribuzione del valore totale delle vendite tra le varie città e tra i vari anni.

Si può subito notare come per ogni città il valore totale delle vendite tenda ad aumentare col passare degli anni tranne per *Wichita Falls* in cui rimane molto simile in tutti gli anni. Inoltre, si evidenzia che il valore totale delle vendite più elevato si ha a Tyler e quello più basso a Wichita Falls. Infine, vale la pena notare come la città con i dati più variabili è *Bryan-College* avendo le “scatole” (che indicano il range interquartile) e i “baffi” (che indicano il range) di dimensioni maggiori rispetto alle altre città.

Quesito 3

Usa un grafico a barre sovrapposte per ogni anno, per confrontare il totale delle vendite nei vari mesi, sempre considerando le città. Prova a commentare ciò che viene fuori. Già che ci sei prova anche il grafico a barre normalizzato. Consiglio: Stai attento alla differenza tra `geom_bar()` e `geom_col()`. PRO LEVEL: cerca un modo intelligente per inserire ANCHE la variabile Year allo stesso blocco di codice, senza però creare accrocchi nel grafico.

Soluzione Q3

Con la libreria *ggplot2* è possibile creare dei grafici a barre tramite due diverse tipologie di geometrie: *geom_bar* e *geom_col*. Il primo viene utilizzato quando si hanno i dati aggregati in categorie e si vuole contare il numero di osservazioni per ognuna di esse, il secondo invece si utilizza quando si vuole fornire un valore specifico per ciascuna categoria senza effettuare nessun conteggio.

In questo caso per **confrontare il totale delle vendite nei vari mesi, sempre considerando le città**, viene utilizzato *geom_col* inserendo nelle estetiche sull’asse x i mesi trasformati in fattori, sull’asse y il numero totale di vendite e come riempimento la variabile *city*.

Inoltre, all’interno di *geom_col* è possibile anche specificare attraverso ***position*** il tipo di grafico a barre desiderato: con ***stack*** si ottiene un grafico a barre sovrapposte, con ***dodge*** un grafico a barre affiancate e con ***fill*** un grafico a barre normalizzato. Di seguito in Fig. 3.1 viene mostrato il codice R con cui si ottengono il grafico a barre sovrapposte

e il grafico a barre normalizzato come richiesto, partendo dall'anno 2010. I grafici ottenuti invece sono mostrati in Fig. 3.2 e Fig. 3.3.

```
ggplot(real_estate_data[year==2010, ])+
  geom_col(aes(
    x = factor(month),
    y = sales,
    fill = city),
  position = "stack",
  col = "black") +
  labs(title = "Grafico a barre sovrapposte delle vendite totali per mese e città - 2010",
    x = "Mese",
    y = "Numero totale di vendite - 2010",
    fill = "Città") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5),
    legend.position = "bottom") +
  scale_y_continuous(breaks = seq(0,1000,100))

ggplot(real_estate_data[year==2010, ] +
  geom_col(aes(
    x = factor(month),
    y = sales,
    fill = city),
  position = "fill",
  col = "black") +
  labs(title = "Grafico a barre normalizzato delle vendite totali per mese e città - 2010",
    x = "Mese",
    y = "Numero totale di vendite relative - 2010",
    fill = "Città") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5),
    legend.position = "bottom") +
  scale_y_continuous(breaks = seq(0,1,0.1))
```

Fig. 3.1 Costruzione del grafico a barre sovrapposte e normalizzato delle vendite totali nel 2010.

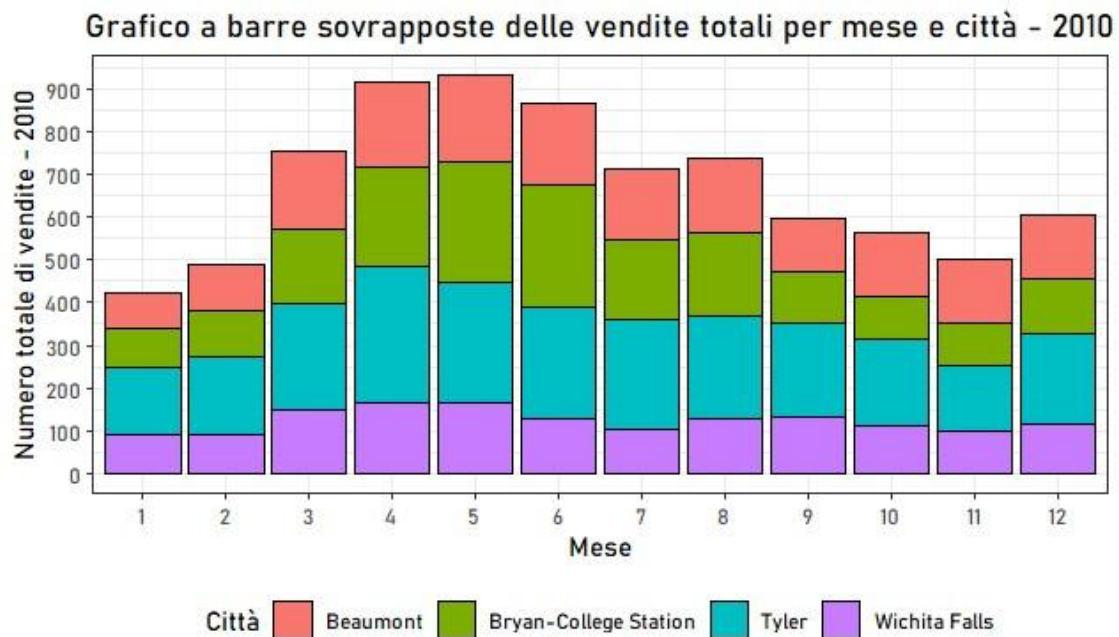


Fig. 3.2 Grafico a barre sovrapposte delle vendite totali per mese e città nell'anno 2010.

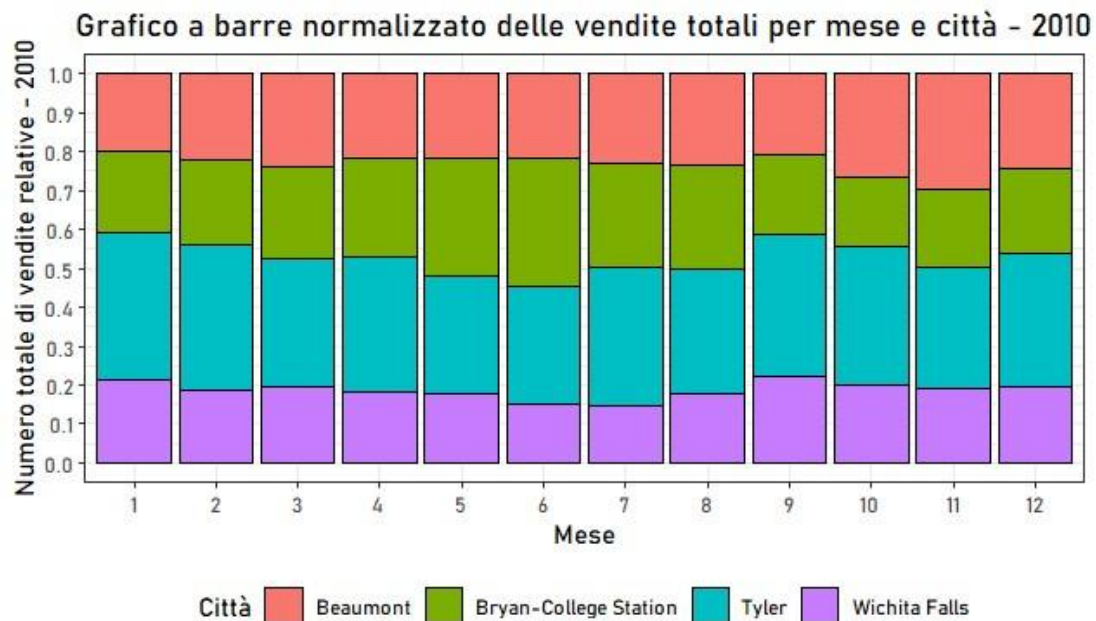


Fig. 3.3 Grafico a barre normalizzato delle vendite totali per mesi e città nell'anno 2010.

Tuttavia, al posto di ripetere lo stesso procedimento per ciascun anno restante è possibile inserire anche la variabile *year* nei precedenti blocchi di codice attraverso la funzione ***facet_wrap*** (si aggiunge lo strato ***facet_wrap(~year)*** alla fine) in modo da ottenere i grafici a barre per ogni anno in un unico grafico.

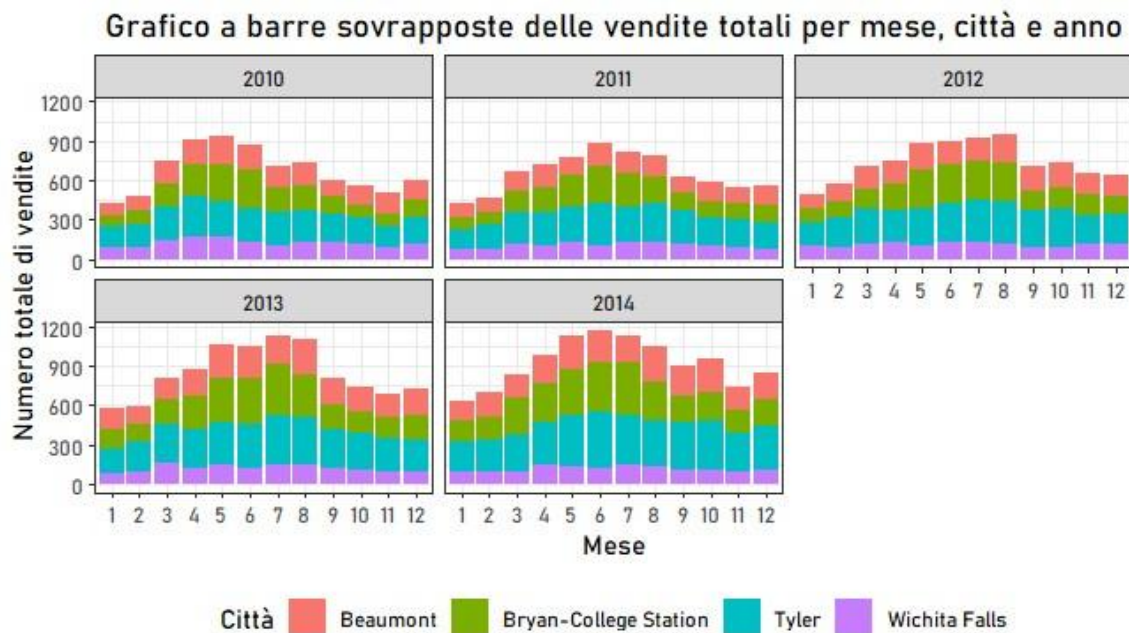


Fig. 3.3 Grafico a barre sovrapposte delle vendite totali per mese, città e anno.

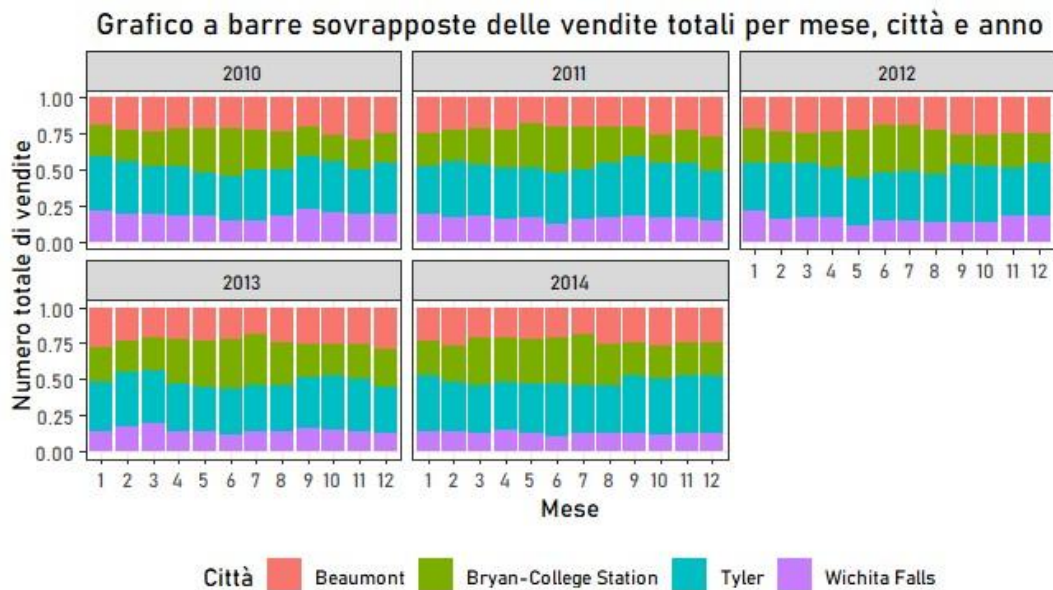


Fig. 3.4 Grafico a barre normalizzato delle vendite totali per mese, città e anno.

Osservando la Fig. 3.3 si può subito notare la tendenza, in tutti e cinque gli anni di riferimento, ad avere un numero totale di vendite complessive maggiore nei mesi primaverili ed estivi rispetto ai mesi autunnali ed invernali. L'anno 2014 invece risulta essere l'anno in cui si è registrato il maggior numero totale di vendite complessive ed in particolare nel mese di giugno si registra il numero totale di vendite complessive più elevato di tutti e cinque gli anni.

Dopo aver analizzato le vendite complessive negli anni, date dalla somma del numero di vendite in tutte e quattro le città, si possono fare delle considerazioni sui contributi dati dalle singole città osservando questa volta entrambe le figure precedenti. Si può notare come in tutti gli anni (e nella quasi totalità dei mesi) la città in cui c'è il più alto numero totale di vendite è Tyler, mentre quella con le vendite minori è Wichita Falls.

Quesito 4

Crea un line chart di una variabile a tua scelta per fare confronti commentati fra città e periodi storici. Ti avviso che probabilmente all'inizio ti verranno fuori linee storte e poco chiare, ma non demordere. Consigli: Prova inserendo una variabile per volta. Prova a usare variabili esterne al dataset, tipo vettori creati da te appositamente. Se non riesci proprio a venirne a capo inizia lavorando su dataset ridotti, ad esempio prendendo in considerazione un solo anno o una sola città. Aiutati con il pacchetto dplyr.

Soluzione Q4

Si sceglie di creare un *line chart* per analizzare la variazione nel tempo della variabile *listings* (numero totale di annunci attivi).

Con *ggplot2* è possibile utilizzare la geometria ***geom_line*** per creare un line chart. Nelle estetiche vengono inseriti sull'asse x i mesi, sull'asse y il numero totale di annunci attivi e per colorare i segmenti in base alle città viene utilizzato ***color = city***. Con *size* invece si può specificare lo spessore delle linee desiderato. Inoltre, con il layer geometrico ***geom_point*** è possibile anche aggiungere dei punti in corrispondenza dei valori utilizzando le stesse estetiche usate in *geom_line*. Infine, per avere un line chart per ogni anno in unico grafico è possibile utilizzare ancora la funzione ***facet_wrap*** (*facet_wrap(~year)*) (Fig. 4.1).

Se si vuole invece avere un unico grafico in cui sull'asse x sono presenti tutti e cinque gli anni suddivisi in mesi si possono sempre utilizzare *geom_line* e *geom_point* ma sostituendo nell'estetiche sull'asse x la formula: $(year - \min(year)) * 12 + month$. Vengono inoltre inseriti altri due strati: ***geom_vline*** con cui vengono inserite delle linee verticali tratteggiate che vanno a dividere il grafico nei diversi anni e ***annotate*** con cui vengono inseriti gli anni stessi come etichette nel grafico. Per l'inserimento delle linee tratteggiate, delle etichette e dei loro posizionamenti vengono utilizzati dei vettori. Infine, nello strato ***scale_x_continuous*** viene stabilito che sull'asse x devono essere rappresentati i mesi dall'1 al 12 per ognuno dei cinque anni (Fig. 4.2).

```
ggplot(real_estate_data) +
  geom_line(aes(
    x = month,
    y = listings,
    color = city,
    size = 1) +
  geom_point(aes(
    x = month,
    y = listings,
    color = city,
    size = 1.5) +
  labs(title = "Line chart del numero totale di annunci attivi per città, mese e anno",
    x = "Mese",
    y = "Numero totale di annunci attivi") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5),
    legend.position = "bottom") +
  scale_x_continuous(breaks = seq(0,12,1)) +
  facet_wrap(~year)
```

Fig. 4.1 Costruzione line chart del numero totale di annunci attivi per città, mese e anno con *facet_wrap*.

```

ggplot(real_estate_data) +
  geom_line(aes(
    x = (year - min(year)) * 12 + month,
    y = listings,
    color = city,
    size = 1) +
  geom_point(aes(
    x = (year - min(year)) * 12 + month,
    y = listings,
    color = city,
    size = 1.5) +
  geom_vline(xintercept = c(1,12,24,36,48,60),
    linetype = "dashed",
    color = "gray30") +
  labs(x = "Mese",
    y = "Numero totale di annunci attivi",
    title = "Line chart del numero totale di annunci attivi per città e mese") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5),
    legend.position = "bottom") +
  scale_x_continuous(breaks = seq(1, 60, 1),
    labels = rep(c(1:12), 5)) +
  annotate("text",
    x = c(6,18,30,42,54),
    y = -3,
    label = c("2010", "2011", "2012", "2013", "2014"),
    size = 4)

```

Fig. 4.2 Costruzione del line chart del numero totale di annunci attivi per città e mese.

Di seguito in Fig. 4.3 e 4.4 vengono mostrati i rispettivi grafici ottenuti.

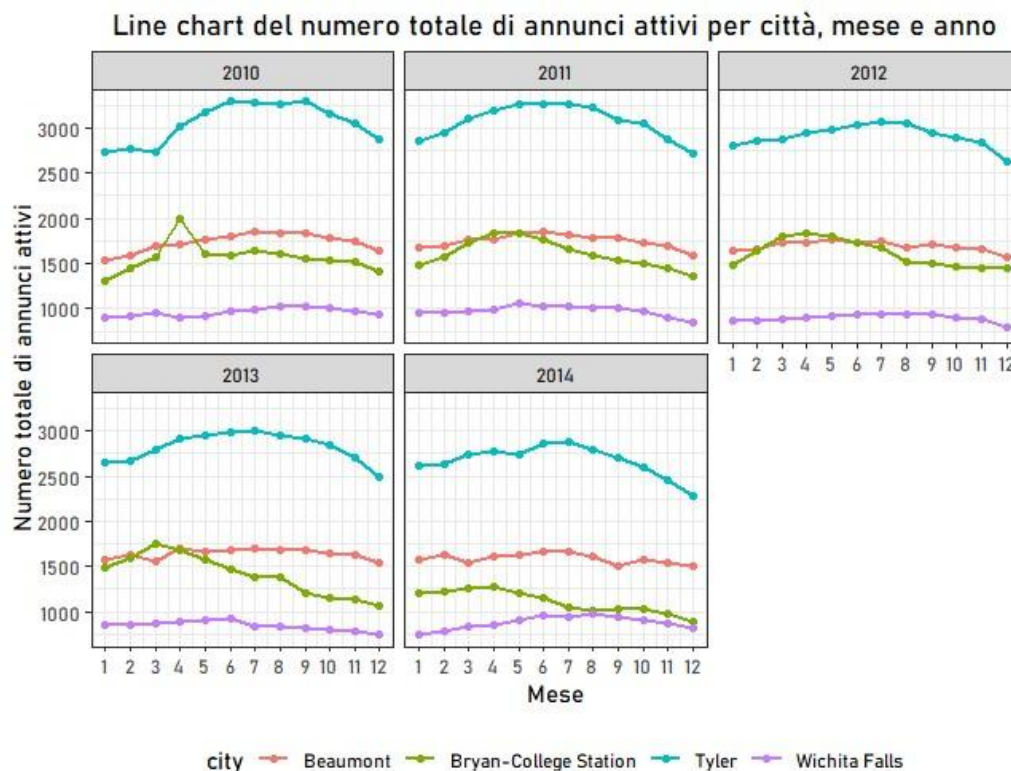


Fig. 4.3 Line chart del numero totale di annunci attivi per città, mese e anno con `facet_wrap`.

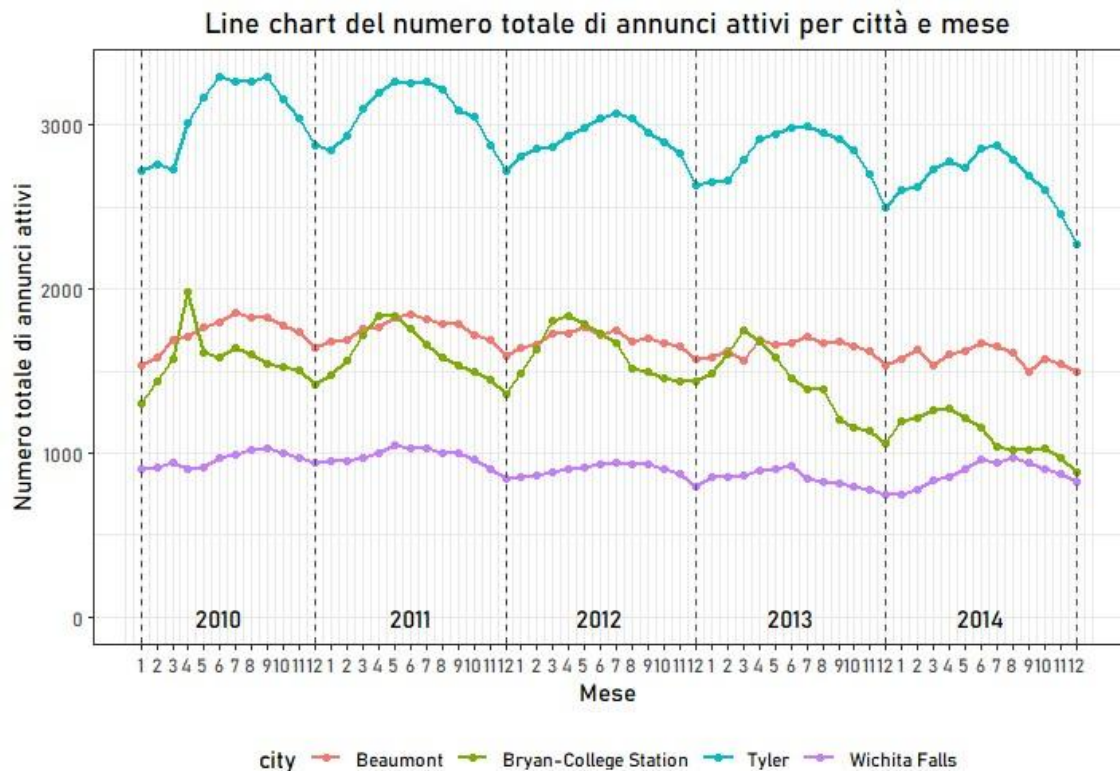


Fig. 4.4 Line chart del numero totale di annunci attivi per città e mese.

Dai precedenti grafici si può subito notare come in tutti e cinque gli anni *Tyler* sia la città in cui il numero totale di annunci attivi è di gran lunga più elevato rispetto alle altre; *Wichita Falls* risulta invece essere in tutto il periodo storico la città col numero di annunci attivi più basso.

Si può inoltre evidenziare la tendenza delle città di *Tyler* e *Bryan-College Station* ad avere un calo del numero di annunci col passare degli anni a differenza delle città di *Beaumont* e *Wichita Falls* in cui il numero di annunci rimane più costante.

Allo stesso modo si evidenzia come le città di *Tyler* e *Bryan-College Station* abbiano una differenza molto più netta tra il numero totale di annunci attivi nei mesi primaverili/estivi ed autunnali/invernali rispetto alle altre due città.