Progetto di Statistica Descrittiva – studio di un dataset immobiliare Texano su R

In questo progetto utilizzeremo il linguaggio R e il software Rstudio. Come prima cosa si dovrà impostare la Directory di lavoro. Per farlo andremo sul menù a tendina a destra, Files, e selezioneremo la cartella di lavoro dove avremo i dati che saranno successivamente lavorati, e dove verranno salvati i nostri file. Una volta selezionata impostiamo la directory.

E' possibile farlo attraverso codice con il comando **'setwd(…)'**, all'interno delle parentesi andrà l'indirizzo della cartella.

Una volta salvato il dataset all'interno della nostra cartella di lavoro andiamo a caricarlo su RStudio, usando il comando **'read.csv'**, nel quale abbiamo anche specificato il separatore dei dati.

```
texasDS <- read.csv("realestate_texas.csv", sep = ",")
```

STUDIO DELLE VARIABILI

Usiamo la funzione **'str(texasDS)'** per ottenere una descrizione generale della struttura dei dati.

```
texasDS                    240 obs. of 8 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 ...
```

Il nostro set di dati è quindi composto da otto variabili distinte e 240 oggetti totali, di cui possiamo vedere anche un breve estratto e il tipo. Notare che la funzione ci segnala alcune variabili come **num**, ovvero numeriche, ma sarebbe più corretto considerarle **double**, ovvero numeri con virgola mobile.

Per completezza possiamo usare la funzione '**summarise**' della libreria **'dplyr'** per una descrizione più precisa. Dopo aver installato e caricato la libreria assegniamo alla variabile **'df_summary'** i nostri dati, che saranno lavorati singolarmente dalla funzione **'typeof'** per creare un nuovo oggetto "riassuntivo".

```
df_summary <- texasDS %>%
summarise(
    city_type = typeof(city),
    year_type = typeof(year),
    month_type = typeof(month),
    sales_type = typeof(sales),
    volume_type = typeof(volume),
    median_price_type = typeof(median_price),
    listings_type = typeof(listings),
    months_inventory_type = typeof(months_inventory)
)
```

```
df_summary                1 obs. of 8 variables
    $ city_type           : chr "character"
    $ year_type           : chr "integer"
    $ month_type          : chr "integer"
    $ sales_type          : chr "integer"
    $ volume_type         : chr "double"
    $ median_price_type   : chr "double"
    $ listings_type       : chr "integer"
    $ months_inventory_type: chr "double"
```

Notiamo che i numeri sono ora riconosciuti come tipi double.

Passiamo allo studio delle singole variabili, che possiamo visionare dall'Enviroment o richiamandole singolarmente con il codice **'head ()'**, come a seguire:

```
head(texasDS$city)
```

L'output sarà una lista dei primi sei dati di '**City**'.

```
[1] "Beaumont" "Beaumont" "Beaumont" "Beaumont" "Beaumont" "Beaumont"
```

Con solo questo piccolo estratto potremo erroneamente pensare che ci sia un solo valore utile in **'City'**. Per confutare ciò dobbiamo visionare l'intera colonna oggetto di studio o usare la funzione **'levels'**. Questa è applicabile alle sole variabili fattoriali, pertanto il primo passaggio è convertire la colonna **'City'** in un fattore e poi lanciare il comando **'levels'**:

```
texasDS$city <- as.factor(texasDS$city)
levels(texasDS$city)
```

Grazie a ciò scopriremo che **'City'** è composta da quattro diverse città.

```
> levels(texasDS$city)
[1] "Beaumont"              "Bryan-College Station" "Tyler"                 "Wichita Falls"
```

Trasformiamo in fattori anche le variabili **'month'** e **'year'**. Nello specifico rendiamo la visualizzazione dei mesi più chiara, trasformando la variabile in nominale e visualizzando il risultato con **'levels'**.

```
texasDS$month <- factor(texasDS$month, levels = 1:12, labels = month.abb)
```

```
> levels(texasDS$month)
 [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
```

Studiando nello specifico ogni variabile possiamo giungere alle seguenti conclusioni:

- **CITY – NOMINALE**
- **YEAR - QUALITATIVA ORDINALE.**
- **MONTH - QUALITATIVA NOMINALE, CICLICA.** I numeri rappresentano specifici mesi dell'anno, il cui ordine non implica una scala di valore.
- **SALES - QUANTITATIVA CONTINUA.** Le vendite possono essere misurate su una scala continua e assumere diversi valori all'interno di un intervallo.
- **VOLUME - QUANTITATIVA CONTINUA**
- **MEDIAN_PRICE - QUANTITATIVA CONTINUA**
- **LISTINGS - QUANTITATIVA DISCRETA.** I valori rappresentano un conteggio di oggetti distinti e non c'è una scala continua tra loro.
- **MONTHS_INVENTORY – QUANTITATIVA CONTINUA**

INDICI DI POSIZIONE, VARIABILITA', FORMA E DISTRIBUZIONE DI FREQUENZA

Adesso calcoleremo gli indici di posizione, variabilità e forma per le variabili numeriche '**sales**', '**volume**', '**median_price**', '**listings**' e '**months_inventory**'.

Per farlo usiamo la funzione '**descr**' di **summarytools**, escludendo le variabili che non ci interessano. Nota importante, per facilitare la lettura dei dati ricordiamoci di impostare la visualizzazione non scientifica dei numeri.

```
options(scipen = 999)
```

```
desc_table <- descr(texasDS, stats.exclude= c("city", "year", "month"))
```

	listings	median_price	months_inventory	sales	volume
Mean	1738.0208333	132665.4166667	9.19250000	192.2916667	31.0051875
Std.Dev	752.7077561	22662.1486865	2.30366862	79.6511112	16.6514472
Min	743.0000000	73800.0000000	3.40000000	79.0000000	8.1660000
Q1	1025.0000000	117100.0000000	7.80000000	127.0000000	17.6290000
Median	1618.5000000	134500.0000000	8.95000000	175.5000000	27.0625000
Q3	2128.0000000	150100.0000000	11.00000000	248.0000000	40.9030000
Max	3296.0000000	180000.0000000	14.90000000	423.0000000	83.5470000
MAD	879.9231000	24092.2500000	2.14977000	82.2843000	16.1566335
IQR	1029.5000000	32750.0000000	3.15000000	120.0000000	23.2335000
CV	0.4330833	0.1708218	0.25060306	0.4142203	0.5370536
Skewness	0.6454431	-0.3622768	0.04071944	0.7136206	0.8792182
SE.Skewness	0.1571376	0.1571376	0.15713758	0.1571376	0.1571376
Kurtosis	-0.8101534	-0.6427292	-0.19794476	-0.3355200	0.1505673
N.Valid	240.0000000	240.0000000	240.00000000	240.0000000	240.0000000
Pct.Valid	100.0000000	100.0000000	100.00000000	100.0000000	100.0000000

A ciò aggiungiamo anche la **Varianza**, ottenibile elevando al quadrato la deviazione standard.

```
variance_results <- texasDS %>%
  select(-city, -year, -month) %>%
  summarise_all(var)
```

	sales	volume	median_price	listings	months_inventory
1	6344.3	277.2707	513572983	566569	5.306889

Listing:

- La distribuzione è piuttosto ampia, con un intervallo di 2553 annunci.
- La Curtosi è negativa, il che indica code più leggere rispetto alla distribuzione normale; pertanto, la sua distribuzione è più piatta e ha meno osservazioni nelle code.
- La skewness è positiva, suggerendo una leggera asimmetria verso destra nella distribuzione.
- Considerando il range interquartile (IQR) e il terzo quartile (Q3) il limite superiore è 3792.50, inferiore al valore massimo della variabile. Ciò suggerisce l'assenza di outlier.

Median Price:

- Presenta una curtosi negativa, con una distribuzione meno appuntita rispetto ad una normale.
- La skewness è leggermente negativa, il che suggerisce una leggera asimmetria verso sinistra nella distribuzione.
- Il valore massimo (198100) non supera il limite superiore. Il dato suggerisce l'assenza di outlier.

Months inventory:

- La distribuzione ha code più leggere rispetto a una distribuzione normale, con una Curtosi molto vicina allo zero.
- Il valore massimo (14.9) non supera il limite superiore. Il dato suggerisce l'assenza di outlier.

Sales:

- La distribuzione è relativamente ampia, con un intervallo di 344 vendite.
- La kurtosi e la skewness suggeriscono una distribuzione che potrebbe essere leggermente meno "appuntita" e asimmetrica verso destra.
- Nessun outlier è identificato in quanto il massimo valore (423.00) non supera il limite superiore.

Volume:

- La variabile volume rappresenta il volume delle vendite.
- La distribuzione sembra avere code più pesanti rispetto a una distribuzione normale (kurtosi positiva).
- La skewness è positiva, indicando una leggera asimmetria verso destra nella distribuzione.
- Il valore massimo (83.55) supera il limite superiore. Il dato suggerisce la presenza di outlier.

Per studiare su R la possibile presenza di outlier tramite il metodo del **Limite superiore** abbiamo riunito i terzi quartili e i range interquartili presenti nella tabella **'desc_table'** in due varibili, **Q3** e **IQR.**

```
Q3 <- c(2128.00, 150100.00, 11.00, 248.00, 40.90)
IQR <- c(1029.50, 32750.00, 3.15, 120.00, 23.23)
```

Procedendo poi con la creazione di un data frame apposito, in cui abbiamo definito le cinque variabili di nostro interesse e i rispettivi massimali.

```
risultati_df <- data.frame(
  variabile = c("listings", "median_price", "months_inventory", "sales", "volume"),
  max_value = c(3296.00, 180000.00, 14.90, 423.00, 83.55),
  limite_superiore
)
```

Infine, abbiamo aggiunto una colonna che indicasse la presenza o meno di outlier; se il max value è superiore al limite superiore della rispettiva variabile restituisce **"POSSIBILI OUTLIER"**, altrimenti **"NESSUN OUTLIER".**

```
risultati_df$outlier <- ifelse(risultati_df$max_value > risultati_df$limite_superiore,
                               "POSSIBILI OUTLIER", "NESSUN OUTLIER")
```

	variabile	max_value	limite_superiore	outlier
1	listings	3296.00	3672.250	NESSUN OUTLIER
2	median_price	180000.00	199225.000	NESSUN OUTLIER
3	months_inventory	14.90	15.725	NESSUN OUTLIER
4	sales	423.00	428.000	NESSUN OUTLIER
5	volume	83.55	75.745	POSSIBILI OUTLIER

Per le variabili ‘**city’, ‘year’ e ‘month’** costruiremo delle tabelle di frequenza.

	Var1	Freq
1	Jan	20
2	Feb	20
3	Mar	20
4	Apr	20
5	May	20
6	Jun	20
7	Jul	20
8	Aug	20
9	Sep	20
10	Oct	20
11	Nov	20
12	Dec	20

	Var1	Freq
1	2010	48
2	2011	48
3	2012	48
4	2013	48
5	2014	48

	Var1	Freq
1	Beaumont	60
2	Bryan-College Station	60
3	Tyler	60
4	Wichita Falls	60

Il prossimo passo sarà capire quale, tra quelle disponibili, è la variabile con variabilità più elevata.

Per capirlo possiamo studiare la Deviazione Standard **(Std.Dev.)** e il Coefficiente di Variazione **(CV)**, indici che ci aiutano a comprendere la dispersione dei dati rispetto alla media. Dobbiamo però fare attenzione, perché con dati che si sviluppano in scale distinte dobbiamo trovare il modo di "normalizzare" i nostri studi, quindi ottenere un valore in scala per tutte le nostre variabili. La deviazione standard non è adatta, perciò ci concentreremo sul coefficiente di variazione, in quanto si tratta di una misura relativa.

In ‘**desc_table’** viene così restituito:

CV	0.4330833	0.1708218	0.25060306	0.4142203	0.5370536

Ma per facilità di lettura trasformiamo il dato in percentuali.

```
df <- data.frame(
  variable = c("Listings", "Median_price", "Months_inventory", "Sales", "Volume"),
  cv = c(0.4330833, 0.1708218, 0.25060306, 0.4142203, 0.5370536)
)
df$cv_Percent <- df$cv * 100
```

	Variable	CV	CV_Percent
1	Listings	0.4330833	43.30833
2	Median_price	0.1708218	17.08218
3	Months_inventory	0.2506031	25.06031
4	Sales	0.4142203	41.42203
5	Volume	0.5370536	53.70536

Ci è ora chiaro che **Volume** ha il coefficiente di variazione più alto, con il **53.705%**. Questo significa che **Volume** ha una variabilità relativa rispetto alla sua media del **53.7%**.

Se invece volessimo determinare la variabile più asimmetrica, dovremmo studiare la **Skewness**. Un valore di **Skewness** diverso da zero indica la presenza di asimmetria nei dati.

- **Listings:** 0.6454431
- **Median_price:** -0.3622768
- **Months_inventory:** 0.04071944
- **Sales:** 0.7136206
- **Volume:** 0.8792182

Risulta evidente che la variabile più asimmetrica è **Volume**, con uno skewness di **0.8792182**, ad indicare una distribuzione asimmetrica positiva.

Studio approfondito di una variabile

Scegliamo una delle variabili quantitative nel nostro dataset, e andiamo a costruire una distribuzione di frequenza, il suo grafico a barre e il relativo **indice di Gini**.

Ho scelto la variabile '**median_price'**, che estrarrò in una variabile a parte.

```
median_price <- texasDS$median_price
```

Il prossimo passò sarà dividerla in classi, per farlo **userò l'algoritmo di Sturges**, che determina il numero di classi da utilizzare in base alla dimensione del campione.

La formula dell'algoritmo è **k=[log_2(N)+1]**, con **k** che rappresenta il numero di classi e N la dimensione del campione. Ricordo che **'ceiling'** è la funzione per arrotondare il risultato.

```
num_classi <- ceiling(log2(length(median_price)) + 1)
```

Adesso generiamo una sequenza di valori che rappresenteranno gli intervalli di classe. Prendiamo gli estremi inferiori dei dati, poi specifichiamo il numero totale di elementi da creare.

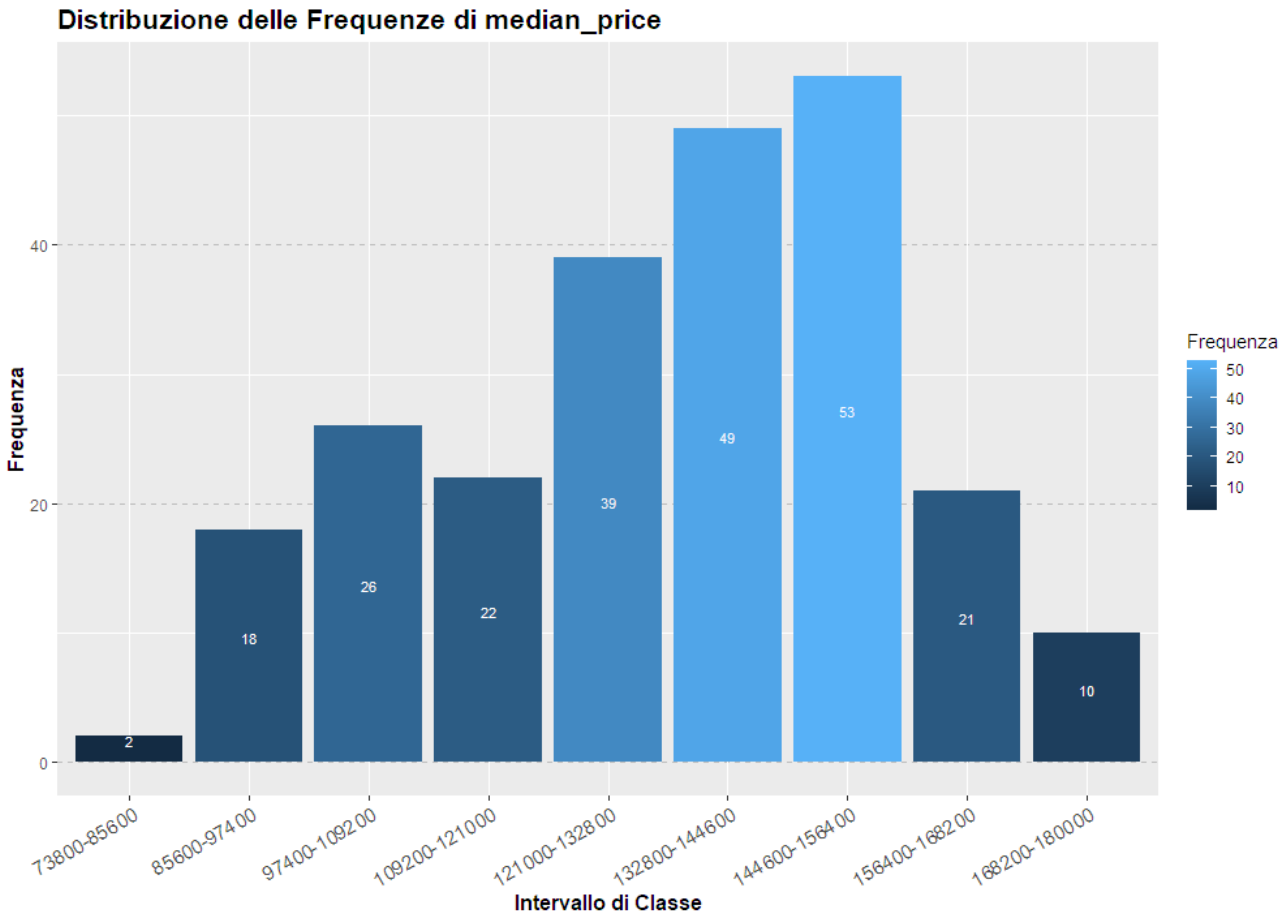
```
intervalli_classe <- seq(min(median_price), max(median_price),
                         length.out = num_classi + 1)
```

Dopo ciò creiamo le etichette per questi intervalli, e assegniamole alla variabile **"median_price"** in base agli intervalli che abbiamo appena definito.

```
etichette_classe <- paste(intervalli_classe[-length(intervalli_classe)],
                          intervalli_classe[-1], sep="-")
classi <- cut(median_price, breaks = intervalli_classe,
              labels = etichette_classe, include.lowest = TRUE)
```

Riuniamo i dati della tabella delle frequenze in un data frame e procediamo alla costruzione del nostro grafico a barre.

```
grafico_a_barre <- ggplot(data = tabella_frequenze_median_price,
                          aes(x = `Intervallo di Classe`, y = Frequenza, fill = Frequenza)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = Frequenza, y = Frequenza), vjust = -0.2, size = 3, color = "white",
            position = position_stack(vjust = 0.5)) +  # Etichette dei totali all'interno delle colonne
  labs(x = "Intervallo di Classe", y = "Frequenza",
       title = "Distribuzione delle Frequenze di median_price") +
  theme(axis.text.x = element_text(angle = 30, hjust = 1, size = 11),
        axis.title = element_text(size = 12, face = "bold"),  # Stile dei titoli degli assi
        plot.title = element_text(size = 16, face = "bold"),  # Stile del titolo del grafico
        panel.grid.major.y = element_line(color = "gray", linetype = "dashed"))  # Linee della griglia
```



Concludiamo con **l'indice di Gini**, una misura di disuguaglianza che va da 0 a 1, dove 0 è la perfetta uguaglianza e 1 la perfetta disuguaglianza.

Si inizia calcolando la somma totale delle frequenze, quindi il numero totale di osservazioni nella distribuzione. Poi si calcola la proporzione della classe rispetto al totale, dividendo ogni frequenza per la somma totale delle frequenze.

L'indice vero e proprio è calcolato come 1 meno la somma dei quadrati delle proporzioni di ciascuna classe.

Per aiutarci nel calcolo installiamo il pacchetto **'ineq'.**

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

frequencies_median_price <- tabella_frequenze_median_price$Freq
gini_index_median_price <- ineq::Gini(frequencies_median_price)
```

```
Indice di Gini per median_price: 0.3407407
```

In questo caso l'indice rappresenta una distribuzione moderatamente disuguale.

Andiamo a studiare un'altra variabile, come '**City**'. L'abbiamo studiata precedentemente, e sappiamo che consta in quattro diversi valori, ognuno ripetuto 60 volte.

```
# Calcoliamo l'indice di Gini per la variabile city

frequencies_city <- tabella_frequenze_city$Freq
gini_index_city <- ineq::Gini(frequencies_city)
cat("Indice di Gini per la variabile city:", gini_index_city, "\n")
```

Come prevedibile, l'indice di Gini è **0**. Si tratta di una distribuzione completamente uguale tra le variabili.

Studio delle probabilità

Per scoprire quante probabilità ci sono che esca una riga con città **'Beaumont'**. Creiamo un vettore che prende da city tutti i Beaumont presenti, e li sommiamo tra di loro per avere il numero totale di righe che ci interessano. Dividendolo per il numero totale di righe del dataset otterremo la probabilità desiderata.

```
prob_citta_beaumont <- sum(texasDS$city == "Beaumont") / nrow(texasDS)
```

```
Probabilità di scegliere una riga con la città di Beaumont: 0.25
```

Per le probabilità riguardanti il mese di luglio:

```
prob_mese_luglio <- sum(texasDS$month == "Jul") / nrow(texasDS)
```

```
Probabilità di scegliere una riga con il mese di Luglio: 0.08333333
```

E infine le probabilità di scegliere una riga con il mese di dicembre 2012, in cui concateniamo il mese e l'anno di nostro interesse:

```
prob_dicembre_2012 <- sum(texasDS$month == "Dec" & texasDS$year == 2012) / nrow(texasDS)
```

```
Probabilità di scegliere una riga con il mese di dicembre 2012: 0.01666667
```

Creazione di nuove variabili

Con una simile mole e varietà di dati possiamo creare nuove variabili, come ad esempio il prezzo medio. Ricaviamolo dividendo il volume per le vendite, e, fatto ciò, creiamo una nuova colonna da aggiungere nel dataset grazie alla funzione **'mutate'**.

```
texasDS <- texasDS %>%
  mutate(efficacy = sales / listings)
str(texasDS)
```

```
'data.frame':   240 obs. of  9 variables:
 $ city            : Factor w/ 4 levels "Beaumont","Bryan-College Station",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ year            : Factor w/ 5 levels "2010","2011",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ month           : Factor w/ 12 levels "Jan","Feb","Mar",..: 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  0.171 0.164 0.158 0.134 0.143 ...
```

Cerchiamo ora di capire l'eventuale efficacia degli annunci di vendita, prendendo in esame le vendite e gli annunci. Divideremo **'sales'** per '**listing'**, formando così la variabile '**efficacy'** che integreremo nel nostro dataset.

```
texasDS <- texasDS %>%
  mutate(efficacy = sales / listings)
str(texasDS)
```

```
'data.frame':   240 obs. of  10 variables:
 $ city            : Factor w/ 4 levels "Beaumont","Bryan-College Station",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ year            : Factor w/ 5 levels "2010","2011",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ month           : Factor w/ 12 levels "Jan","Feb","Mar",..: 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  0.171 0.164 0.158 0.134 0.143 ...
 $ efficacy        : num  0.0541 0.0681 0.1078 0.1171 0.1141 ...
```

In questa nuova colonna '**efficacy'**, un valore più alto potrebbe indicare una maggiore efficacia degli annunci di vendita rispetto al numero di annunci attivi.

Creazione di summary per controlli incrociati

Usando il pacchetto '**dplyr**' creiamo dei '**summary**' delle variabili '**listings**', '**sales**' e '**average price**' in base al mese, all'anno e alla città. Per farlo costruiamo una funzione ad hoc che estrarrà i dati dal dataset scelto, in base al grouping, su una lista di vettori.

```r
generate_summary <- function(data, grouping_var, value_vars) {
  summary_df <- data %>%
    group_by({{ grouping_var }}) %>%
    summarise(across({{ value_vars }},
                     list(mean = ~mean(., na.rm = TRUE),
                     sd = ~sd(., na.rm = TRUE)), .names = "{col}_{fn}"))

  return(summary_df)
}
```

```r
# Usa la funzione per ottenere i summary per month
summary_month <- generate_summary(texasDS,
                                  month, c(listings, sales, average_price))

# Usa la funzione per ottenere i summary per city
summary_city <- generate_summary(texasDS,
                                 city, c(listings, sales, average_price))

# Usa la funzione per ottenere i summary per year
summary_year <- generate_summary(texasDS,
                                 year, c(listings, sales, average_price))
```

summary_month

Filter

	month	listings_mean	listings_sd	sales_mean	sales_sd	average_price_mean	average_price_sd
1	Jan	1647.05	704.6140	127.40	43.38372	0.1456404	0.02981911
2	Feb	1692.50	711.2004	140.85	51.06783	0.1488405	0.02512042
3	Mar	1756.70	727.3546	189.45	59.17812	0.1511365	0.02323792
4	Apr	1825.70	770.4287	211.70	65.40489	0.1514613	0.02617430
5	May	1823.85	790.2234	238.85	83.11582	0.1582350	0.02578719
6	Jun	1833.25	811.6288	243.55	94.99832	0.1615458	0.02347046
7	Jul	1821.20	826.7196	235.75	96.27421	0.1568810	0.02722012
8	Aug	1786.30	815.8664	231.45	79.22883	0.1564556	0.02825321
9	Sep	1748.90	802.6563	182.35	72.51807	0.1565223	0.02966941
10	Oct	1710.35	779.1649	179.90	74.95395	0.1558974	0.03252729
11	Nov	1652.70	741.2533	156.85	55.46670	0.1542330	0.02968487
12	Dec	1557.75	692.5678	169.40	60.74658	0.1549955	0.02700887

summary_city

Filter

	city	listings_mean	listings_sd	sales_mean	sales_sd	average_price_mean	average_price_sd
1	Beaumont	1679.3167	91.13382	177.3833	41.48395	0.1466404	0.01123213
2	Bryan-College Station	1458.1333	252.52753	205.9667	84.98374	0.1835343	0.01514935
3	Tyler	2905.0500	226.75458	269.7500	61.96380	0.1676768	0.01235051
4	Wichita Falls	909.5833	73.75504	116.0667	22.15192	0.1194300	0.01139848

summary_year

Filter

	year	listings_mean	listings_sd	sales_mean	sales_sd	average_price_mean	average_price_sd
1	2010	1826.000	785.0201	168.6667	60.53708	0.1501886	0.02327955
2	2011	1849.646	780.3777	164.1250	63.87042	0.1482506	0.02493838
3	2012	1776.812	738.4492	186.1458	70.90509	0.1508987	0.02643850
4	2013	1677.604	743.5239	211.9167	83.99641	0.1587052	0.02652381
5	2014	1560.042	706.7086	230.6042	95.51490	0.1635587	0.03174053

Scriviamo una funzione per facilitare la creazione dei grafici correlati.

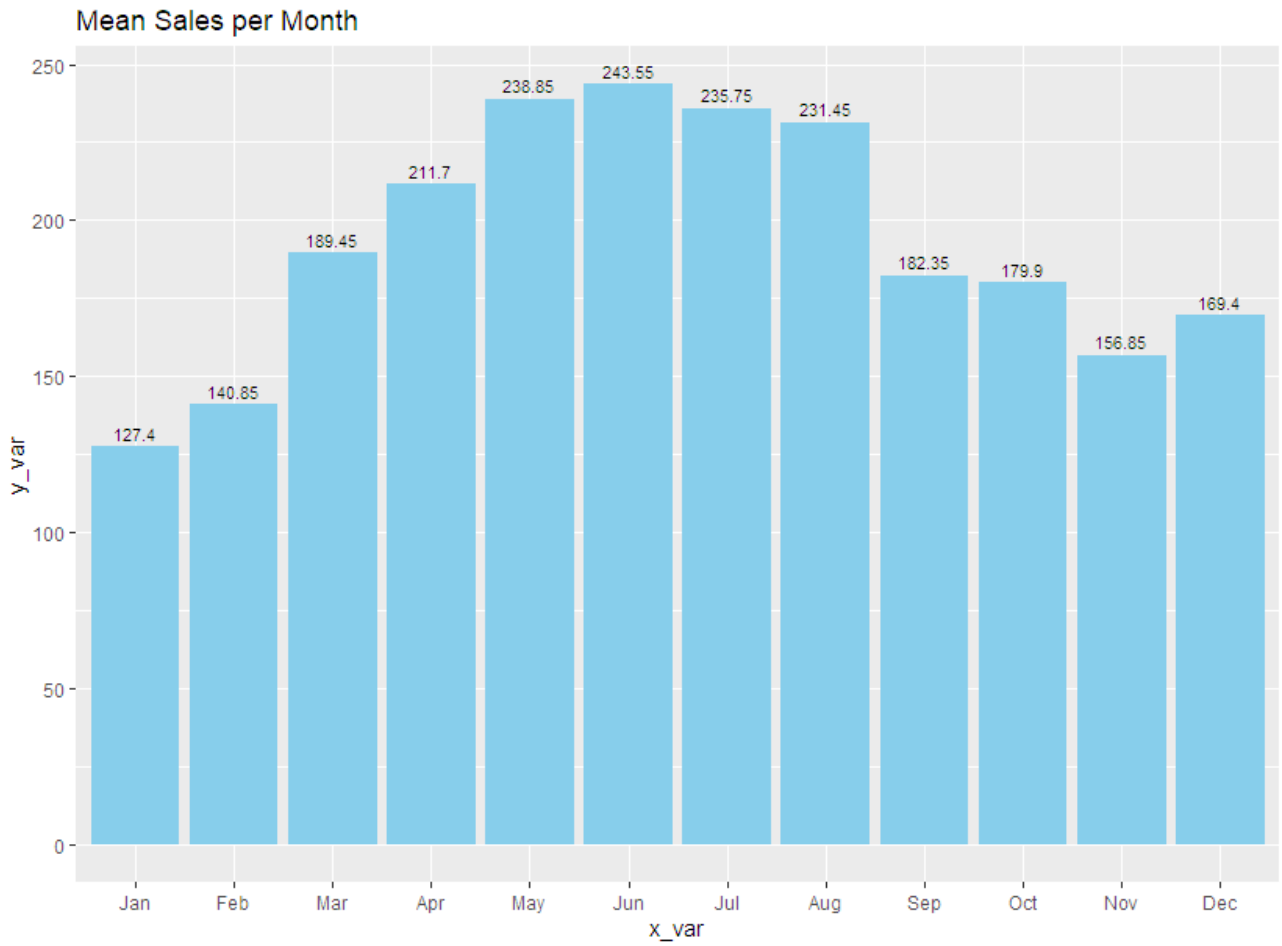
```r
#Creiamo una funzione per creare i grafici sui summary interessati.
create_bar_chart <- function(data, x_var, y_var, title) {
  ggplot(data, aes(x = {{ x_var }}, y = {{ y_var }})) +
    geom_bar(stat = "identity", fill = "skyblue") +
    geom_text(aes(label = round({{ y_var }}, 2)), vjust = -0.5, size = 3) +
    labs(title = title, x = as_label(quo(x_var)), y = as_label(quo(y_var)))
}
```

```r
# Creiamo i grafici per month
bar_chart_month_listings <- create_bar_chart(summary_month, month, listings_mean, "Mean Listings per Month")
bar_chart_month_sales <- create_bar_chart(summary_month, month, sales_mean, "Mean Sales per Month")
bar_chart_month_average_price <- create_bar_chart(summary_month, month, average_price_mean, "Mean Average Price per Month")
```

```r
# Creiamo i grafici per city
bar_chart_city_listings <- create_bar_chart(summary_city, city, listings_mean, "Mean Listings per City")
bar_chart_city_sales <- create_bar_chart(summary_city, city, sales_mean, "Mean Sales per City")
bar_chart_city_average_price <- create_bar_chart(summary_city, city, average_price_mean, "Mean Average Price per City")
```

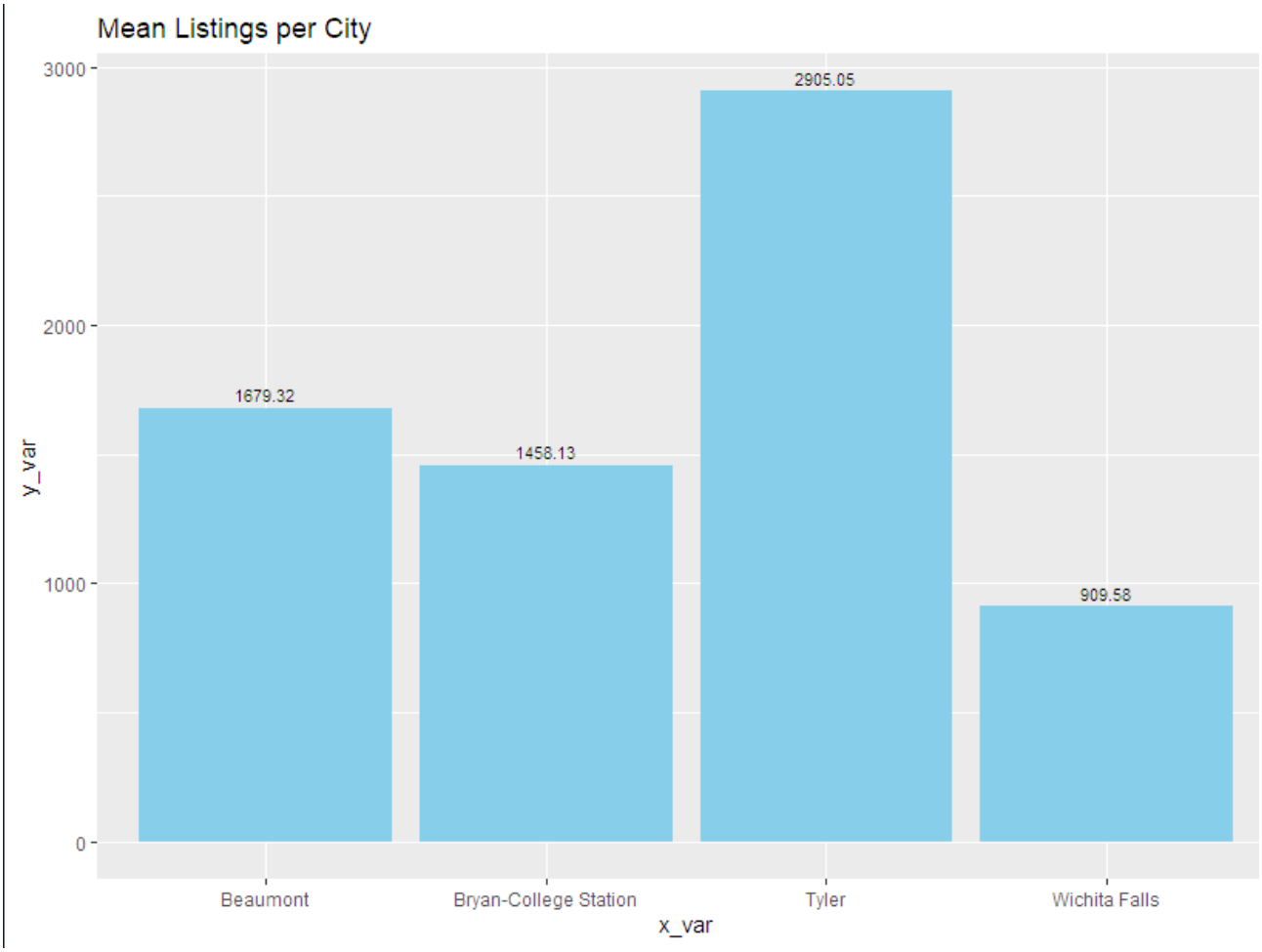
```r
# Creiamo i grafici per year
bar_chart_year_listings <- create_bar_chart(summary_year, year, listings_mean, "Mean Listings per Year")
bar_chart_year_sales <- create_bar_chart(summary_year, year, sales_mean, "Mean Sales per Year")
bar_chart_year_average_price <- create_bar_chart(summary_year, year, average_price_mean, "Mean Average Price per Year")
```

Di questi nove grafici riporterò qui i più interessanti, iniziando da quello incentrato sulla correlazione tra i mesi e le vendite. Gli altri sono consultabili nel file R.

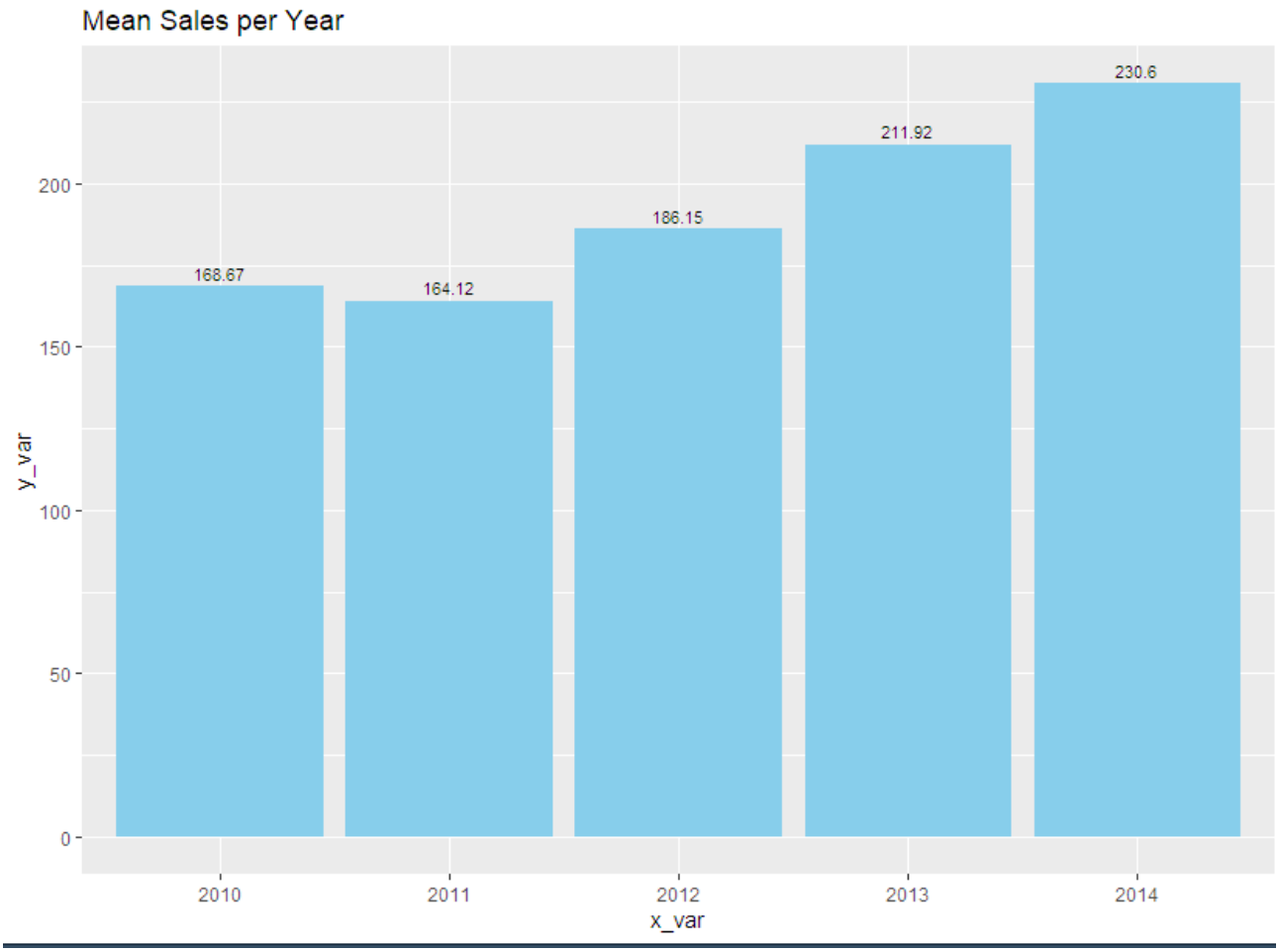


Le vendite sono incentrate nei mesi estivi, subiscono una battuta di arresto a settembre e raggiungono il punto più basso a gennaio.

Chiediamo ora quale, tra le quattro città interessate, ha il numero più alto di annunci di vendita e quindi il più ampio mercato immobiliare.



Tyler è senza ombra di dubbio la città con più annunci di vendita. In ultimo analizziamo le vendite in base all'anno.

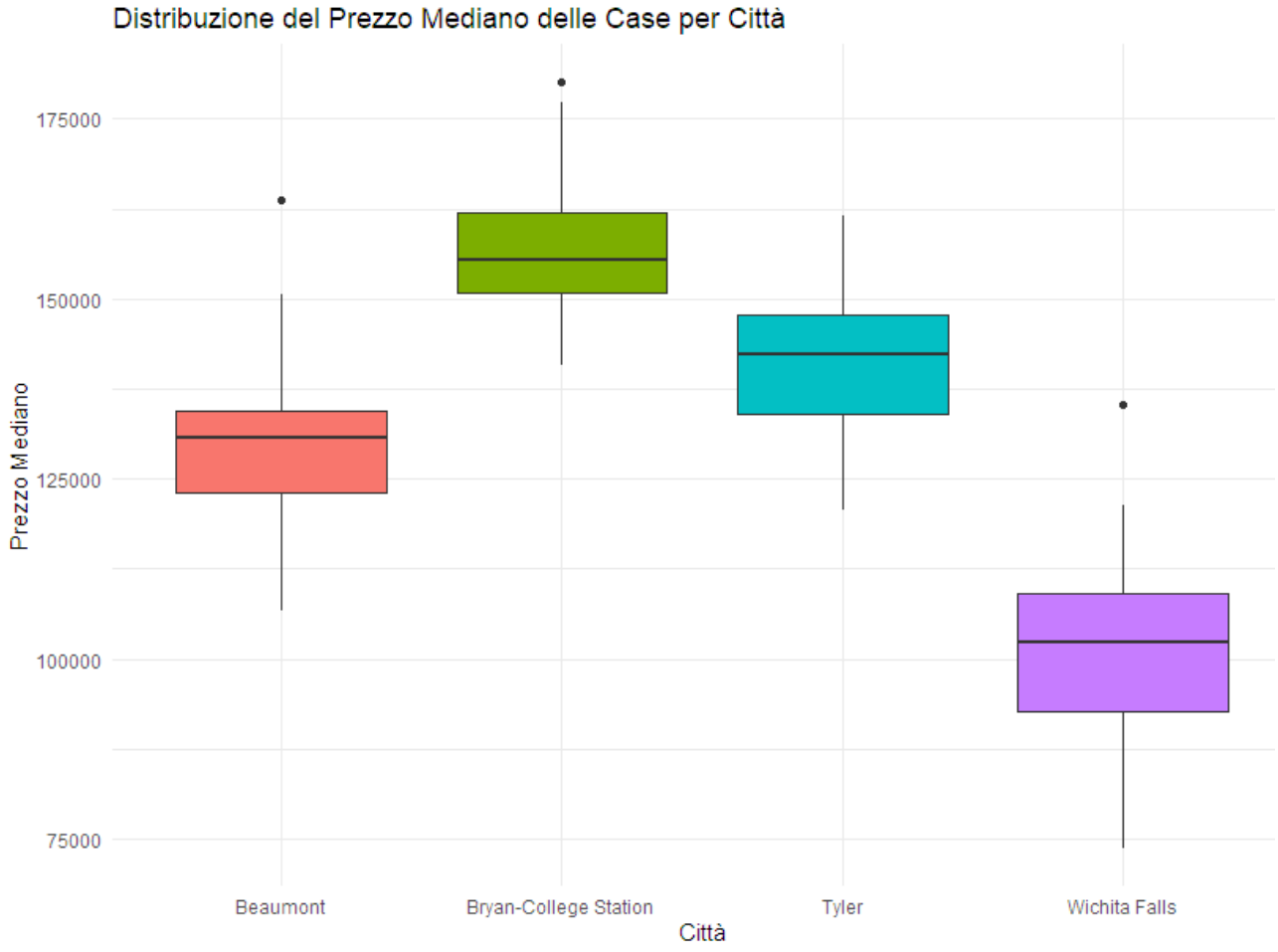


Dal 2012 si è registrata una continua risalita nel numero di vendite, fino al 2014, ultimo anno registrato nei nostri dati. Una tendenza chiaramente positiva, senza accenni di diminuzione.

BOXPLOT

Adesso usiamo un diverso tipo di rappresentazione grafica, i Boxplot. Confronteremo la distribuzione del prezzo mediamo tra le case nelle nostre quattro città. Iniziamo con la costruzione del boxplot:

```
ggplot(texasDS, aes(x = city, y = median_price, fill = city)) +
  geom_boxplot() +
  labs(title = "Distribuzione del Prezzo Mediano delle Case per Città",
       x = "Città",
       y = "Prezzo Mediano") +
  theme_minimal() +
  theme(legend.position="none")
```

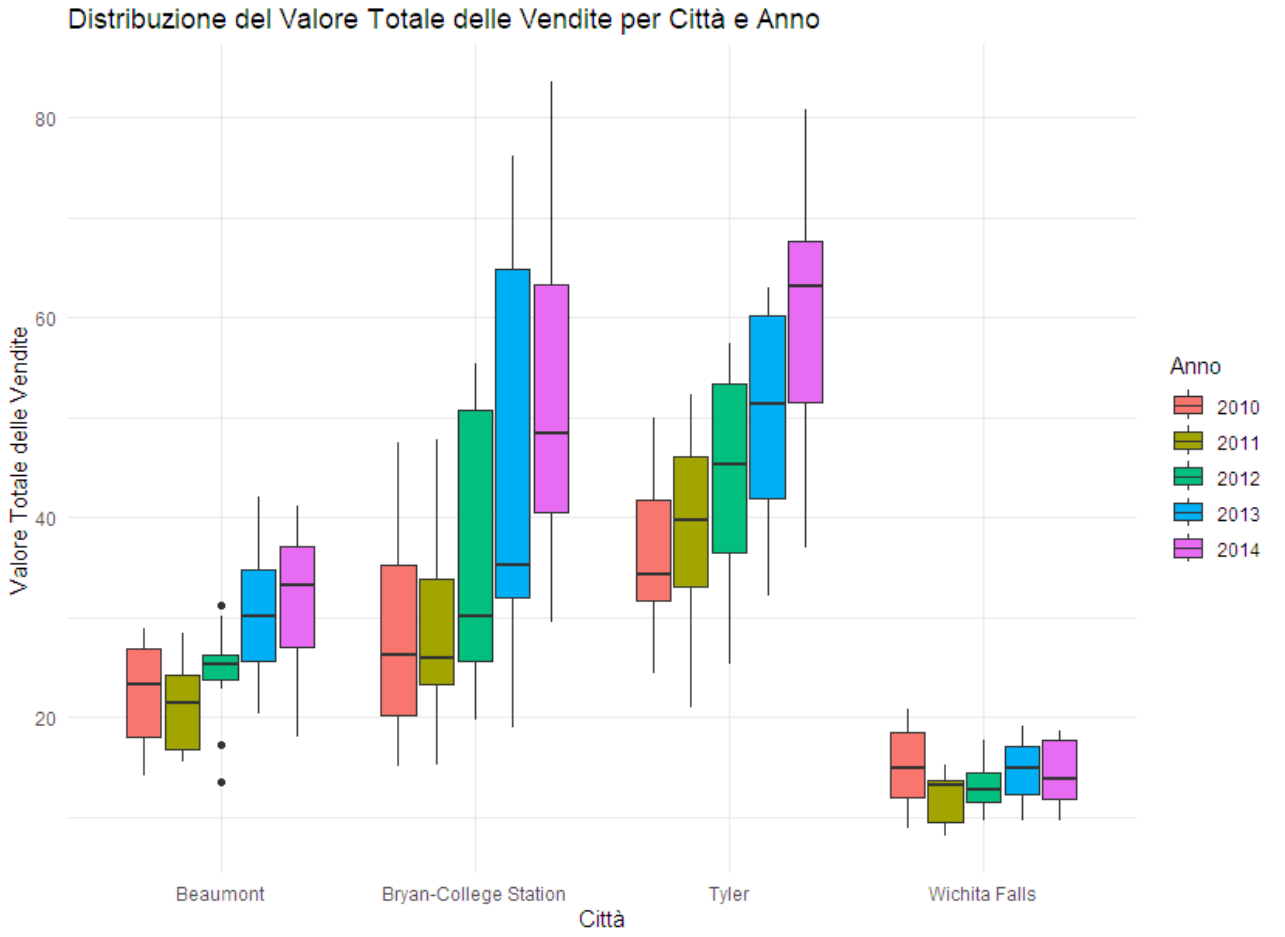


Notiamo fin da subito la presenza di outlier positivi, posti sopra la scatola corrispondente. Questo indica la presenza di prezzi medi molto al di sopra delle normali distribuzioni registrate. Nel caso di Beaumont e Wichita Falls si trovano anche piuttosto distanziati dai “baffi”, che rappresentano i dati oltre i range interquartili. Per Bryan-College Station l’outlier è posto vicino al baffo, pertanto il valore anomale non è troppo lontano dai valori “normali”.

Tra i quattro interessati Beaumont è la variabile in cui la mediana si discosta più dal centro della distribuzione, rivelando una distribuzione di dati asimmetrica, con i dati concentrati nella parte inferiore. Wichita Falls ha una mediana posta leggermente sopra il centro della distribuzione dovuta a possibili valori più alti, ma si deve tenere presente anche il baffo inferiore, e quindi una dispersione dei dati nella parte inferiore ben più pronunciata.

Il prossimo boxplot prenderà in esame la distribuzione del valore totale delle vendite per città e anno.

```
ggplot(texasDS, aes(x = city, y = volume, fill = as.factor(year))) +
  geom_boxplot() +
  labs(title = "Distribuzione del Valore Totale delle Vendite per Città e Anno",
       x = "Città",
       y = "Valore Totale delle Vendite",
       fill = "Anno") +
  theme_minimal()
```



La prima costante che salta all'occhio è la crescita del dato nei cinque anni interessati nelle città di Beaumont, Bryan-College Station e Tyler. Soprattutto nelle ultime due si registra una tendenza positiva nelle vendite, estremamente costante a Tyler (basta notare l'ascendenza pulita del dato) e con alcune variabili a Bryan-College Station,

dove nel 2013 ci sono state delle vendite molto ben distribuite, visto l'ampio range interquartile.

Beaumont non vanta i numeri delle altre due città, ma a partire dal 2013 ha dimostrato una tendenza positiva, soprattutto tenendo conto del particolare 2012, con un range di vendite estremamente ristretto, seppur in risalita rispetto al 2011. Questo si può evincere dal baffo inferiore di Beaumont 2012, molto vicino alla scatola. Importante notare come Beaumont 2012 sia l'unico nella nostra distribuzione totale a presentare outlier, due in negativo molto lontani dai baffi e uno positivo, piuttosto vicino ai baffi e quindi al dato definibile normale.

Bryan-College ci mostra delle distribuzioni di vendita molto solide, con una maggiore variabilità nei pressi superiori alla mediana soprattutto nel 2013, tendenza che si è ridotto nel 2014, con un range interquartile più concentrato.

Tyler presenta delle distribuzioni che, come precedentemente commentato, sono continuamente in crescita, e che inoltre presentano una variabilità mediamente più costante di quella delle altre città. Questo potrebbe suggerirci un mercato più solido ed affidabile.

Le vendite di Wichita Falls sono molto al di sotto delle altre città, anche se la scala di valori dimostra una forma, una tendenza simile a quello di Beaumont e Bryan-College. Un buon 2010, con discese nel 2011 e nel 2012 prima di una costante risalita nei successivi tre anni. Il tutto ovviamente rapportato ad un mercato di tutt'altro livello. Un particolare interessante è quello della mediana nel 2011, estremamente vicina al limite superiore del suo range interquartile. Questo dimostra un'estrema varietà nelle vendite sotto alla mediana (asimmetria positiva), mentre i prezzi di vendita più alti si concentrano molto vicini tra loro.

GRAFICO A BARRE NORMALIZZATO PER STUDIO INCROCIATO

```r
# Grafico a barre normalizzato, totale delle vendite per città e anno.
texasDS |>
  mutate(perc = sales / sum(sales), .by = c(year, month)) |>
  ggplot(aes(x = month, y = sales, fill = city)) +
  geom_bar(stat = 'identity', position = 'fill') +
  facet_grid(rows = vars(year), scales = 'free_y') +
  labs(x = 'Mese', y = 'Percentuale del TOtale delle Vendite', fill = 'Città',
       title = 'Confronto normalizzato del Totale delle Vendite') +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = 'top',
    strip.text.y = element_text(angle = 0)
  ) +
  scale_fill_brewer(palette = 'Set2') +
  geom_text(aes(label = scales::percent(perc, accuracy = .1)),
            position = position_fill(vjust = 0.5))
```

Vista la mole di dati presenti nell'immagine consiglio di visualizzarla su Rstudio, qui metterò solo una preview in modo da mostrare il layout.

Ho aggiunto alla visualizzazione le percentuali di vendita di ogni città per lo specifico periodo temporale, per rendere più leggibile ed immediato il grafico a barre. Per far ciò ho creato una nuova colonna dove riunire le percentuali e **‘con geom_text’** le ho applicate al grafico.