Descriptive Statistics Project - study of a Texan real estate dataset on R

In this project we will use the R language and the Rstudio software. As a first step we will have to set up the Working Directory. To do this we will go to the drop-down menu on the right, Files, and select the working directory where we will have the data that will later be processed, and where our files will be saved. Once selected we set the directory.

It is possible to do this through code with the command **'setwd(...)'**, inside the brackets will go the address of the folder.

Once we have saved the dataset within our workbook we go to upload it to RStudio, using the **'read.csv'** command, in which we have also specified the data separator.

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

STUDY OF VARIABLES

We use the function **'str(texasDS)'** to get a general description of the data structure.

```
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 ...
```

Our dataset thus consists of eight distinct variables and 240 total objects, of which we can also see a brief excerpt and type. Note that the function reports some variables to us as num, i.e., numeric, but it would be more correct to consider them **double**, i.e., floating-point numbers.

For completeness we can use the '**summarise**' function of the **'dplyr'** library for a more precise description. After installing and loading the library we assign to the variable **'df_summary'** our data, which will be processed individually by the **'typeof'** function to create a new 'summarize' object.

```
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"
```

We note that numbers are now recognized as double types.

We move on to the study of the individual variables, which we can view from the Enviroment or by calling them individually with the **'head ()'** code, as below:

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

The output will be a list of the first six **'City'** data.

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

With only this small excerpt we may mistakenly think that there is only one useful value in **'City'**. To refute this we must either view the entire column under study or use the **'levels'** function. This is applicable to factorial variables only, so the first step is to convert the **'City'** column to a factor and then run the **'levels'** command:

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

Through this we will discover that 'City' is composed of four different cities.

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

We also transform the variables **'month'** and **'year'** into factors. Specifically, we make the display of months clearer by transforming the variable to nominal and displaying the result with **'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"
```

By studying each variable specifically, we can come to the following conclusions:

- **CITY - NOMINAL**
- **YEAR - QUALITATIVE ORDINAL.**
- **MONTH - NOMINAL QUALITATIVE, CYCLICAL.** The numbers represent specific months of the year, the order of which does not imply a scale of value.
- **SALES - CONTINUOUS QUANTITATIVE.** Sales can be measured on a continuous scale and take on different values within a range.
- **VOLUME - QUANTITATIVE CONTINUOUS**
- **MEDIAN_PRICE - QUANTITATIVE CONTINUOUS**
- **LISTINGS - DISCRETE QUANTITATIVE.** The values represent a count of discrete objects and there is no continuous scale between them.
- **MONTHS_INVENTORY - QUANTITATIVE CONTINUOUS**

INDICES OF POSITION, VARIABILITY, SHAPE AND FREQUENCY DISTRIBUTION

We will now calculate the position, variability and shape indices for the numerical variables '**sales**', '**volume**', '**median_price**', '**listings**' and '**months_inventory**'.

To do this we use the '**descr**' function of **summarytools**, excluding the variables we are not interested in. Important note, to make it easier to read the data let's remember to set the numbers to be displayed unscientifically.

```
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

To this we also add the **Variance**, which can be obtained by squaring the standard deviation.

```
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:

- The distribution is quite wide, with a range of 2553 listings.
- Kurtosis is negative, which indicates lighter tails than the normal distribution; therefore, its distribution is flatter and has fewer observations in the tails.
- The skewness is positive, suggesting a slight rightward skewness in the distribution.
- Considering the interquartile range (IQR) and the third quartile (Q3), the upper limit is 3792.50, lower than the maximum value of the variable. This suggests the absence of outliers.

Median Price:

- It has a negative kurtosis, with a less pointed distribution than a normal one.
- The skewness is slightly negative, suggesting a slight leftward skewness in the distribution.
- The maximum value (198100) does not exceed the upper limit. The figure suggests the absence of outliers.

Months inventory:

- The distribution has lighter tails than a normal distribution, with a Kurtosis very close to zero.
- The maximum value (14.9) does not exceed the upper limit. The figure suggests the absence of outliers.

Sales:

- The distribution is relatively wide, with a range of 344 sales.
- Kurtosis and skewness suggest a distribution that may be slightly less "sharp" and asymmetrical toward the right.
- No outliers are identified as the maximum value (423.00) does not exceed the upper limit.

Volume:

- The volume variable represents the volume of sales.
- The distribution appears to have heavier tails than a normal distribution (positive kurtosis).
- The skewness is positive, indicating a slight rightward skewness in the distribution.
- The maximum value (83.55) exceeds the upper limit. The figure suggests the presence of outliers.

To study on R the possible presence of outliers by the **Upper Limit** method, we combined the third quartiles and interquartile ranges in the **'desc_table'** table into two varibles, **Q3** and **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)
```

We then proceeded to create a dedicated data frame, in which we defined the five variables of our interest and their respective ceilings.

```
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
)
```

Finally, we added a column indicating the presence or absence of outliers; if the max value is greater than the upper limit of the respective variable it returns **"POSSIBLE OUTLIER,"** otherwise **"NO 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

For the variables '**city', 'year' and 'month'** we will construct frequency tables.

	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

The next step is to figure out which, among those available, is the variable with the highest variability.

To understand this we can study the Standard Deviation **(Std.Dev.)** and the Coefficient of Variation **(CV)**, indices that help us understand the dispersion of the data from the mean. We have to be careful, however, because with data that are developed at distinct scales we have to find a way to "normalize" our studies, thus obtaining a scaled value for all our variables. The standard deviation is not suitable, so we will focus on the coefficient of variation, since it is a relative measure.

In '**desc_table'** is thus returned:

CV	0.4330833	0.1708218	0.25060306	0.4142203	0.5370536

But for ease of reading let us turn the figure into percentages.

```
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
```

It is now clear to us that **Volume has the** highest coefficient of variation, at **53.705%**. This means that **Volume** has a relative variability from its mean of **53.7%**.

If instead we wanted to determine the most skewed variable, we would have to study **Skewness**. A **Skewness** value other than zero indicates the presence of skewness in the data.

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

It is apparent that the most skewed variable is **Volume**, with a skewness of **0.8792182**, indicating a positive skewed distribution.

In-depth study of a variable

We choose one of the quantitative variables in our dataset, and go on to construct a frequency distribution, its bar graph and its **Gini index**.

I chose the variable '**median_price**', which I will extract into a separate variable.

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

The next step will be to divide it into classes; to do this **I will use Sturges' algorithm**, which determines the number of classes to use based on the sample size.

The formula of the algorithm is **k=⌈log_2(N)+1⌉**, with **k** representing the number of classes and N the sample size. Recall that **'ceiling'** is the function for rounding the result.

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

We now generate a sequence of values that will represent the class intervals. We take the lower extremes of the data, then specify the total number of elements to be created.

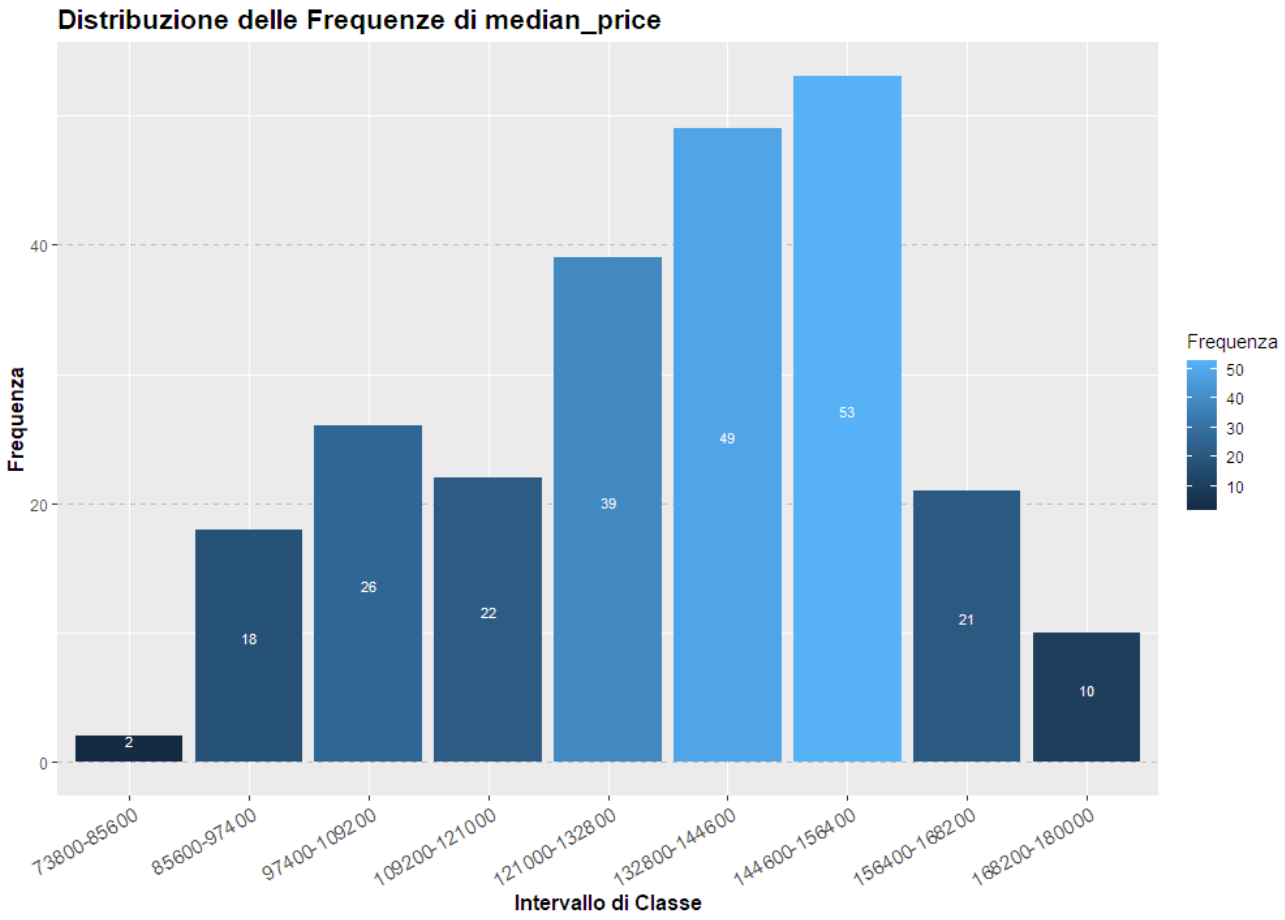
```
intervalli_classe <- seq(min(median_price), max(median_price),
                         length.out = num_classi + 1)
```

After that we create labels for these intervals, and assign them to the variable **"median_price"** based on the intervals we just defined.

```
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)
```

We gather the data from the frequency table into a data frame and proceed to construct our bar graph.

```
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
```



We conclude with the **Gini index**, a measure of inequality ranging from 0 to 1, where 0 is perfect equality and 1 is perfect inequality.

You start by calculating the total sum of frequencies, then the total number of observations in the distribution. Then you calculate the proportion of the class to the total by dividing each frequency by the total sum of frequencies.

The actual index is calculated as 1 minus the sum of the squares of the proportions of each class.

To help with the calculation we install the package **'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 this case, the index represents a moderately unequal distribution.

Let's go on to study another variable, such as '**City**'. We studied it previously, and we know that it consists of four different values, each repeated 60 times.

```
# 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")
```

As might be expected, the Gini index is **0**. This is a completely equal distribution across variables.

Probability study

To find out how likely it is that a row with city **'Beaumont' will** come up. We create a vector that takes from city all the Beaumonts present, and sum them together to get the total number of rows we are interested in. Dividing this by the total number of rows in the dataset we get the desired probability.

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

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

For the probabilities concerning the month of July:

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

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

And finally the probabilities of choosing a row with the month of December 2012, in which we concatenate the month and year of our interest:

```
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
```

Creation of new variables

With such a volume and variety of data, we can create new variables, such as average price. Let's derive it by dividing the volume by the sales, and, having done so, create a new column to add to the dataset thanks to the **'mutate'** function.

```
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 ...
```

Let us now try to understand the effectiveness, if any, of sales listings by looking at sales and listings. We will divide **'sales'** by '**listings**,' thus forming the variable '**effectiveness**' that we will integrate into our dataset.

```r
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 this new '**effectiveness'** column, a higher value could indicate higher effectiveness of sales ads than the number of active ads.

Creating summaries for cross-checking

Using the package '**dplyr**' we create '**summaries**' of the variables '**listings**', '**sales**' and '**average price**' by month, year and city. To do this we build an ad hoc function that will extract data from the chosen dataset, based on grouping, onto a list of vectors.

```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

We write a function to facilitate the creation of correlated graphs.

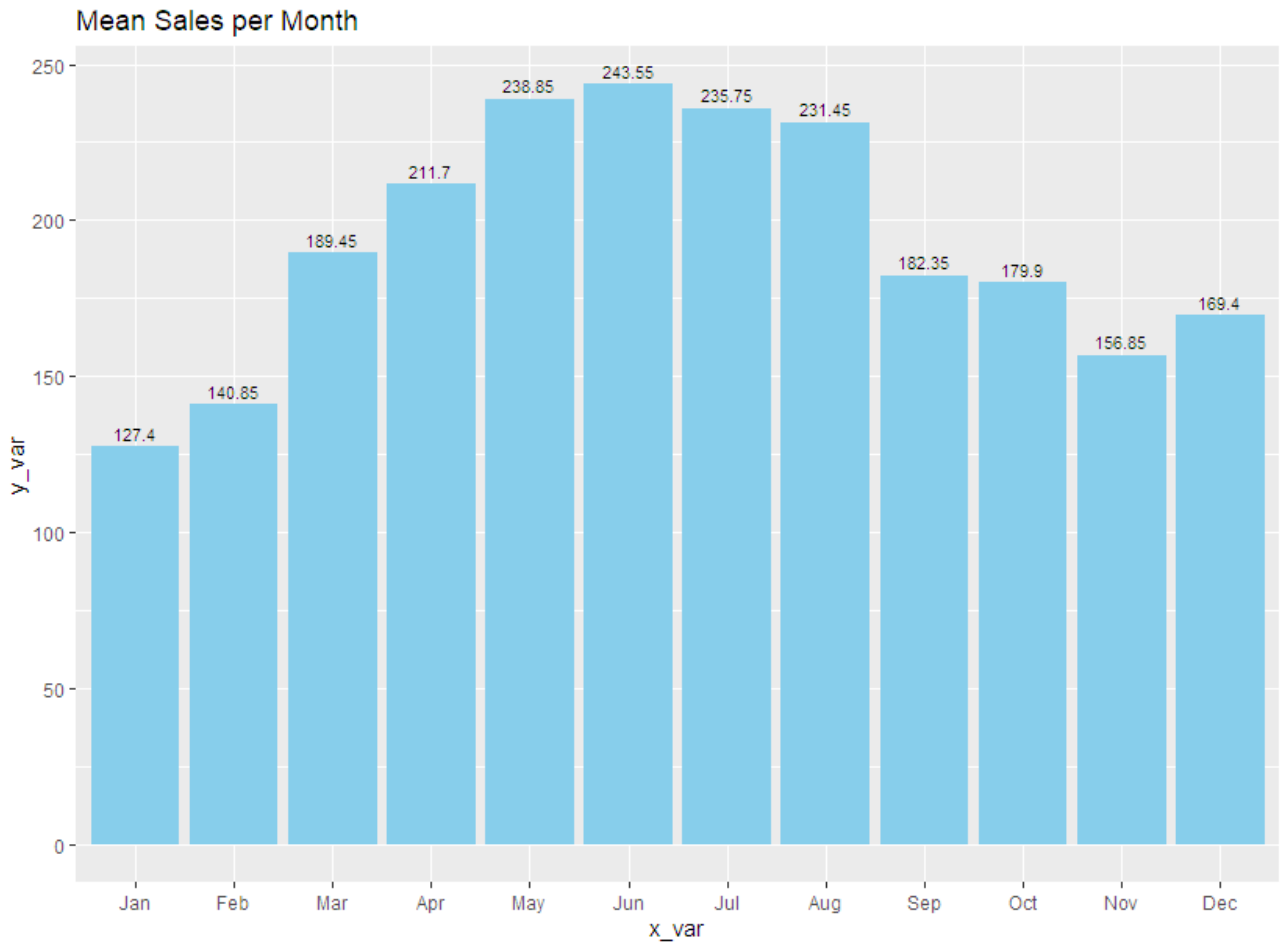
```r
#Creiamo una funzione per creare i grafici sui summary interessati.
create_bar_chart <- function(data, x_var, y_var, title) {
  ggplot(data, aes(x = {{ x_var }}, y = {{ y_var }})) +
    geom_bar(stat = "identity", fill = "skyblue") +
    geom_text(aes(label = round({{ y_var }}, 2)), vjust = -0.5,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

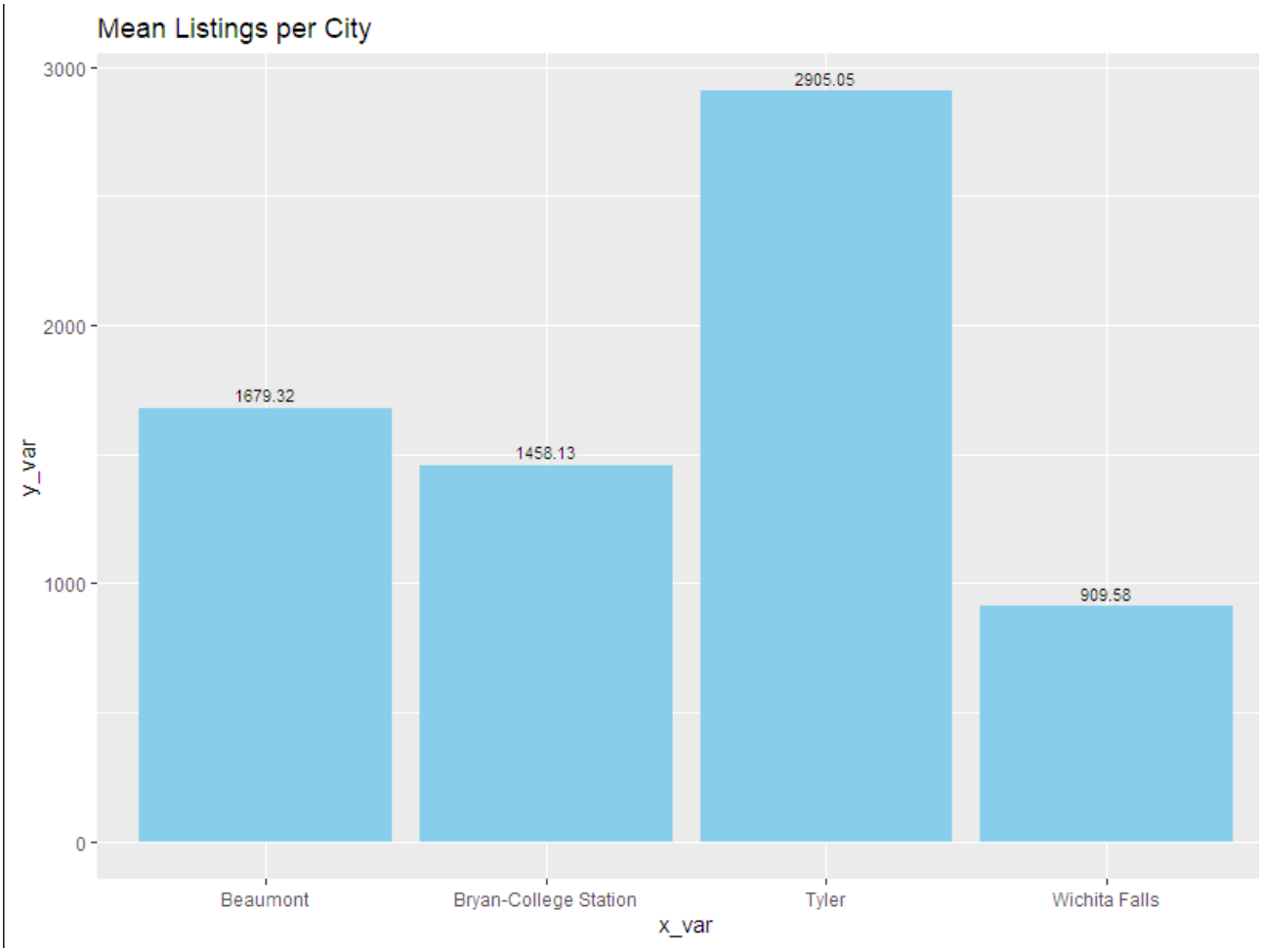
```r
# Creiamo i grafici per year
bar_chart_year_listings <- create_bar_chart(summary_year, year, listings_mean, "Mean Listings per Year")
bar_chart_year_sales <- create_bar_chart(summary_year, year, sales_mean, "Mean Sales per Year")
bar_chart_year_average_price <- create_bar_chart(summary_year, year,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

Of these nine graphs I will report the most interesting ones here, starting with the one focusing on the correlation between months and sales. The others can be found in the R file.

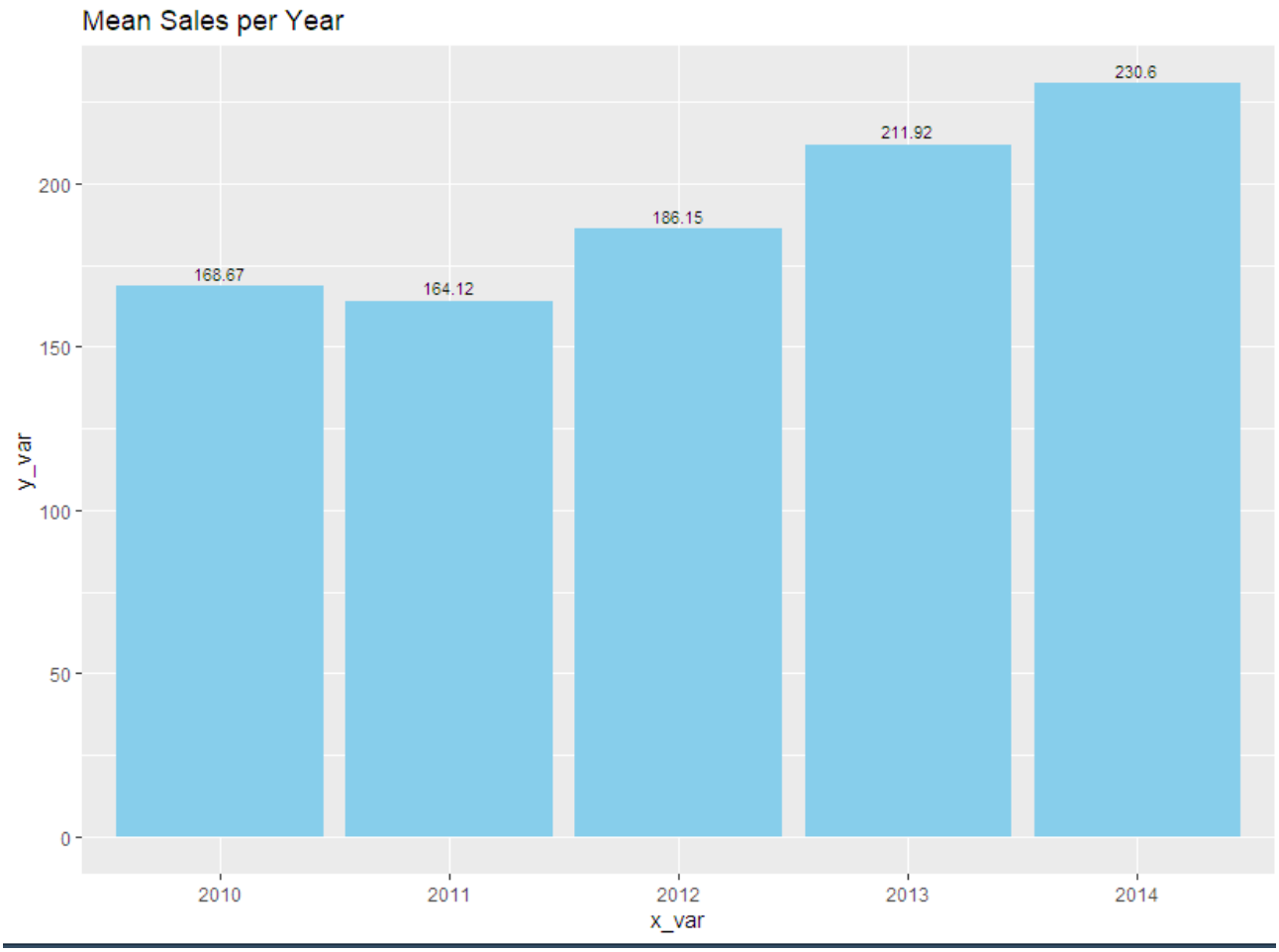


Sales are centered in the summer months, suffering a setback in September and reaching a low point in January.

Let us now ask which, among the four cities involved, has the highest number of for-sale listings and thus the largest housing market.



Tyler is undoubtedly the city with the most sales listings. Finally, we analyze sales by year.

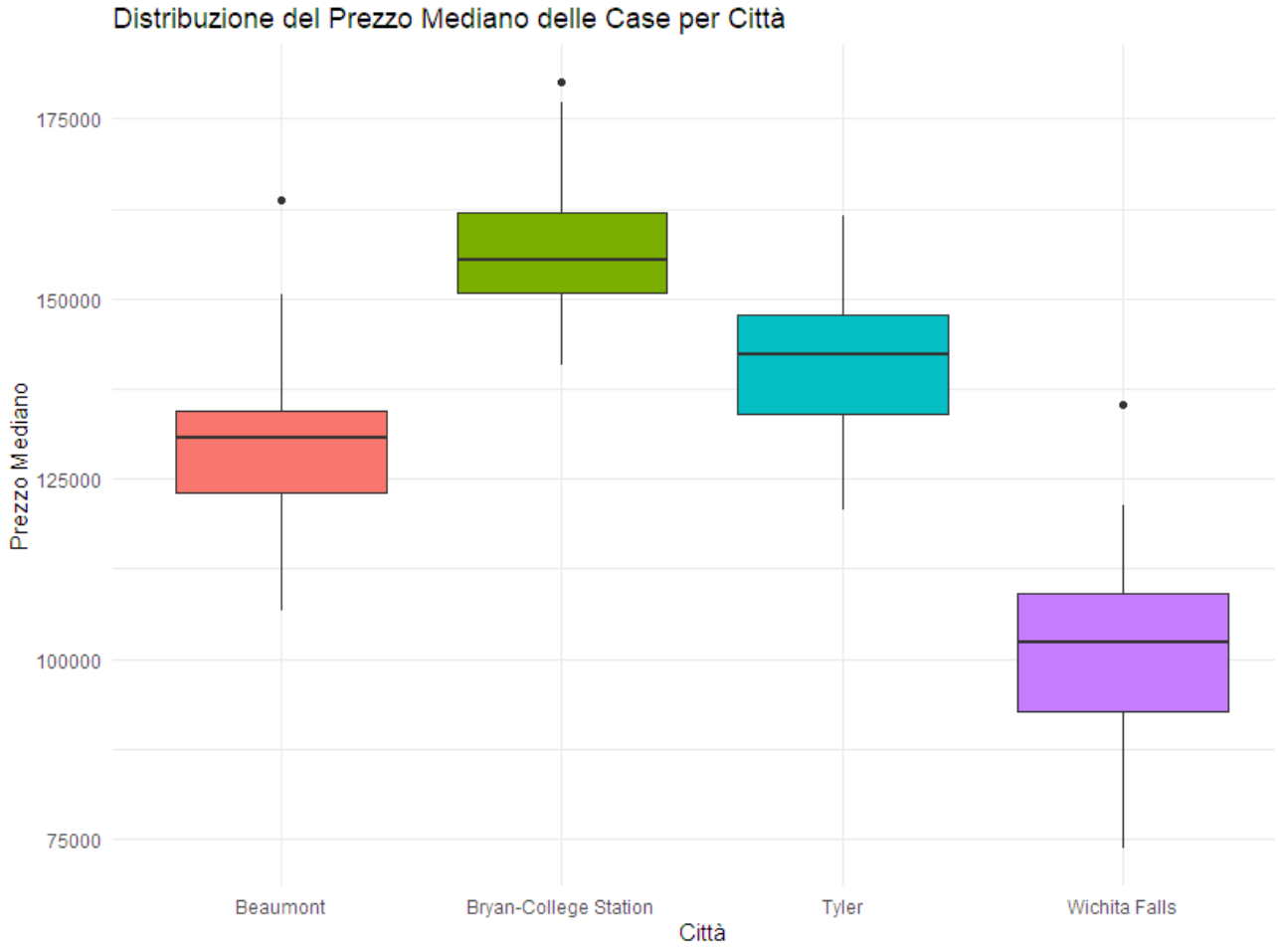


Since 2012 there has been a steady climb in the number of sales, until 2014, the last year recorded in our data. A clearly positive trend, with no hint of decline.

BOXPLOT

We now use a different type of graphical representation, Boxplots. We will compare the median price distribution among houses in our four cities. Let's start with the construction of the 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")
```

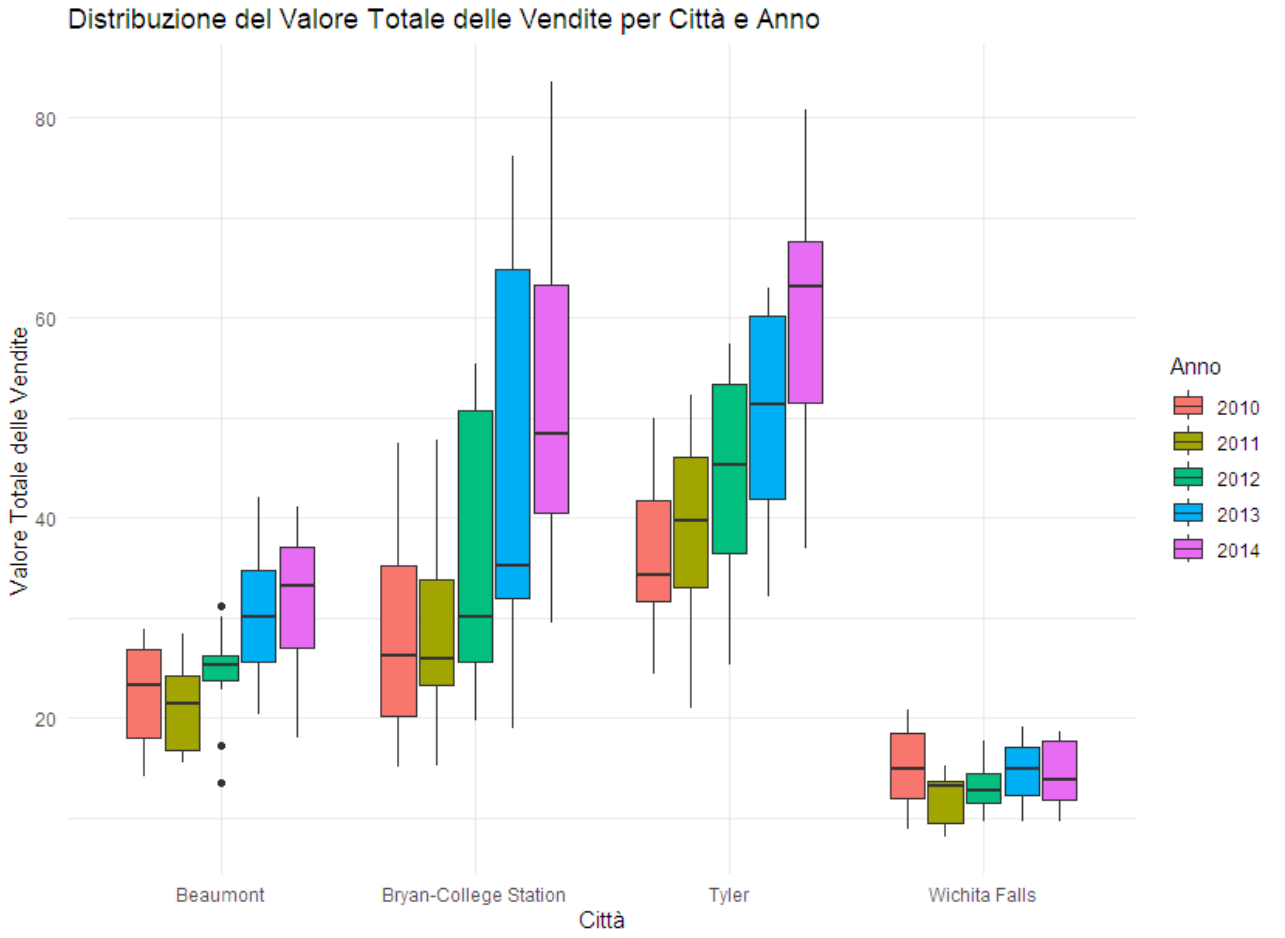


We notice early on the presence of positive outliers placed above the corresponding box. This indicates the presence of average prices far above the normal recorded distributions. In the case of Beaumont and Wichita Falls they are also located quite far from the "whiskers," which represent data above interquartile ranges. For Bryan-College Station the outlier is placed near the whiskers, so the outlier value is not too far from the "normal" values.

Among the four interested Beaumont is the variable in which the median deviates most from the center of the distribution, revealing a skewed distribution of data, with the data concentrated at the bottom. Wichita Falls has a median placed slightly above the center of the distribution due to possible higher values, but the lower whisker must also be taken into account, and thus a much more pronounced dispersion of data at the bottom.

The next boxplot will examine the distribution of total sales value by city and year.

```
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()
```



The first constant that stands out is the growth in the figure over the five years covered in the cities of Beaumont, Bryan-College Station and Tyler. Especially in the last two there is a positive trend in sales, extremely constant in Tyler (just note the clean ascendancy of the figure) and with some variables in Bryan-College Station,

where there were very well distributed sales in 2013, given the wide interquartile range.

Beaumont does not boast the numbers of the other two cities, but as of 2013 it has shown a positive trend, especially taking into account the particular 2012, with an extremely narrow range of sales, albeit up from 2011. This can be seen from the lower whisker of Beaumont 2012, which is very close to the box. Importantly, Beaumont 2012 is the only one in our total distribution to have outliers, two negative ones very far from the whiskers and one positive one, quite close to the whiskers and thus to the figure that can be defined as normal.

Bryan-College shows us very robust sales distributions, with greater variability near above the median especially in 2013, a trend that narrowed in 2014, with a more concentrated interquartile range.

Tyler has distributions that, as previously commented, are continually increasing, and also have a more consistent average variability than that of other cities. This may suggest a more robust and reliable market.

Wichita Falls sales are far below the other cities, although the scale of values shows a similar shape, trend to Beaumont and Bryan-College. A good 2010, with declines in 2011 and 2012 before a steady rise over the next three years. All of course related to a market of a completely different level. An interesting detail is that of the median in 2011, which is extremely close to the upper limit of its interquartile range. This shows extreme variety in sales below the median (positive asymmetry), while the highest sales prices are concentrated very close together.

NORMALIZED CROSS-STUDY BAR GRAPH

```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))
```

Given the amount of data in the image I recommend viewing it in Rstudio, here I will only put a preview so as to show the layout.

I added the sales percentages for each city for the specific time period to the visualization to make the bar graph more readable and immediate. To do this, I created a new column where I gather the percentages and **'with geom_text'** applied them to the graph.