ggplot2_tutorial

Raquel Baltazar

Fri 18 Dec 2020 (21:40:30)

Contents

Source https://cedricscherer.netlify.app/2019/08/05/a-ggplot2-tutorial-for-beautiful-plotting-in-r/

Load data and assign Dataset

```
chic <- readr::read_csv("https://raw.githubusercontent.com/Z3tt/R-Tutorials/master/ggplot2/chicago-nmma
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   city = col_character(),
##   date = col_date(format = ""),
##   death = col_double(),
##   temp = col_double(),
##   dewpoint = col_double(),
##   pm10 = col_double(),
##   o3 = col_double(),
##   time = col_double(),
##   season = col_character(),
##   year = col_double()
## )
```

```
tibble::glimpse(chic)
```

```
## Rows: 1,461
## Columns: 10
## $ city     <chr> "chic", "chic", "chic", "chic", "chic", "chic", "chic", "c...
## $ date     <date> 1997-01-01, 1997-01-02, 1997-01-03, 1997-01-04, 1997-01-0...
## $ death    <dbl> 137, 123, 127, 146, 102, 127, 116, 118, 148, 121, 110, 127...
## $ temp     <dbl> 36.0, 45.0, 40.0, 51.5, 27.0, 17.0, 16.0, 19.0, 26.0, 16.0...
## $ dewpoint <dbl> 37.500, 47.250, 38.000, 45.500, 11.250, 5.750, 7.000, 17.7...
## $ pm10     <dbl> 13.052268, 41.948600, 27.041751, 25.072573, 15.343121, 9.3...
## $ o3       <dbl> 5.659256, 5.525417, 6.288548, 7.537758, 20.760798, 14.9408...
## $ time     <dbl> 3654, 3655, 3656, 3657, 3658, 3659, 3660, 3661, 3662, 3663...
## $ season   <chr> "Winter", "Winter", "Winter", "Winter", "Winter", "Winter"...
## $ year     <dbl> 1997, 1997, 1997, 1997, 1997, 1997, 1997, 1997, 1997, 1997...
```

```
head(chic,10)
```

```
## # A tibble: 10 x 10
##    city  date       death  temp dewpoint  pm10    o3  time season  year
##    <chr> <date>     <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl> <chr>  <dbl>
##  1 chic  1997-01-01   137  36       37.5  13.1   5.66  3654 Winter  1997
##  2 chic  1997-01-02   123  45       47.2  41.9   5.53  3655 Winter  1997
##  3 chic  1997-01-03   127  40       38    27.0   6.29  3656 Winter  1997
##  4 chic  1997-01-04   146  51.5     45.5  25.1   7.54  3657 Winter  1997
##  5 chic  1997-01-05   102  27       11.2  15.3  20.8   3658 Winter  1997
##  6 chic  1997-01-06   127  17        5.75  9.36 14.9   3659 Winter  1997
##  7 chic  1997-01-07   116  16        7    20.2  11.9   3660 Winter  1997
##  8 chic  1997-01-08   118  19       17.8  33.1   8.68  3661 Winter  1997
##  9 chic  1997-01-09   148  26       24    12.1  13.4   3662 Winter  1997
## 10 chic  1997-01-10   121  16        5.38 24.8  10.4   3663 Winter  1997
```

A Default ggplot

##library(ggplot2)

```
library(tidyverse)
```

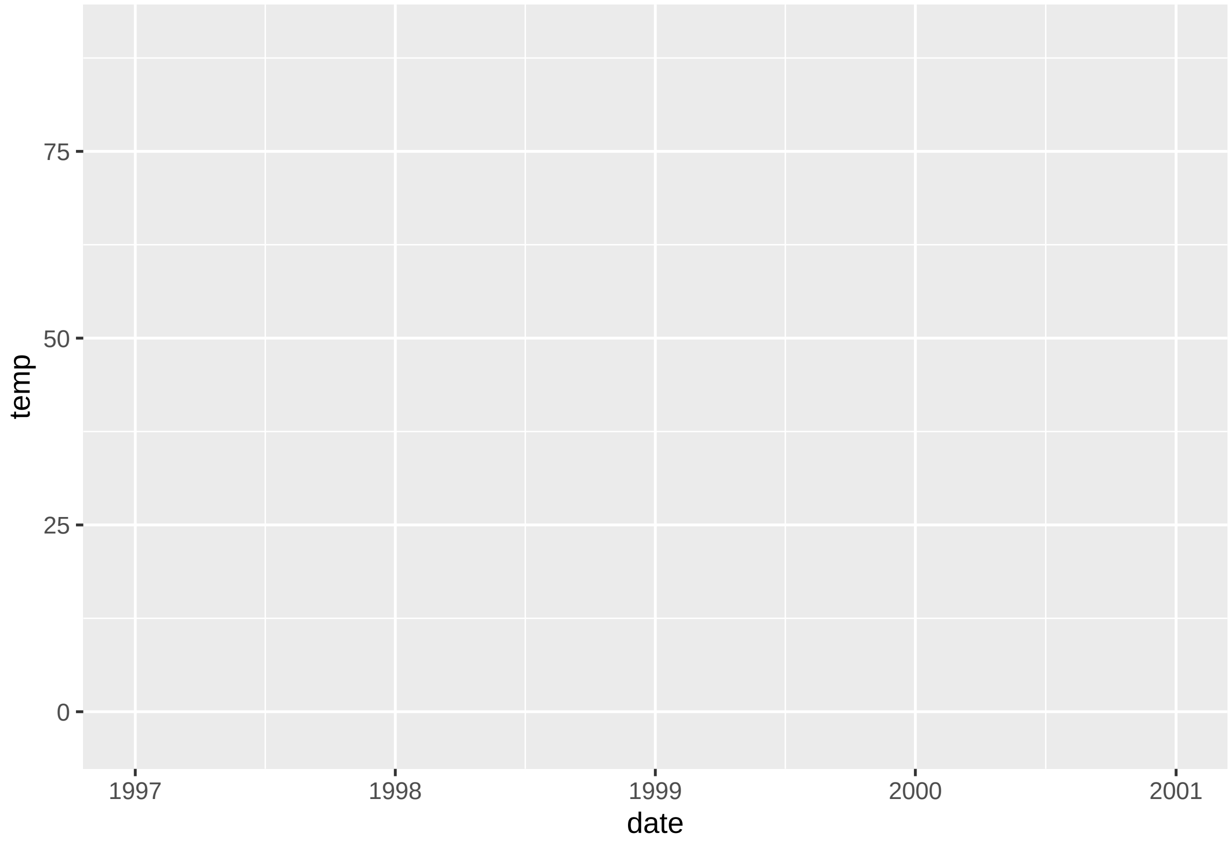
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

We specify the data outside aes() and add the variables that ggplot maps the aesthetics to inside aes()

1st: We map the variable date to the x position and temp to the y position

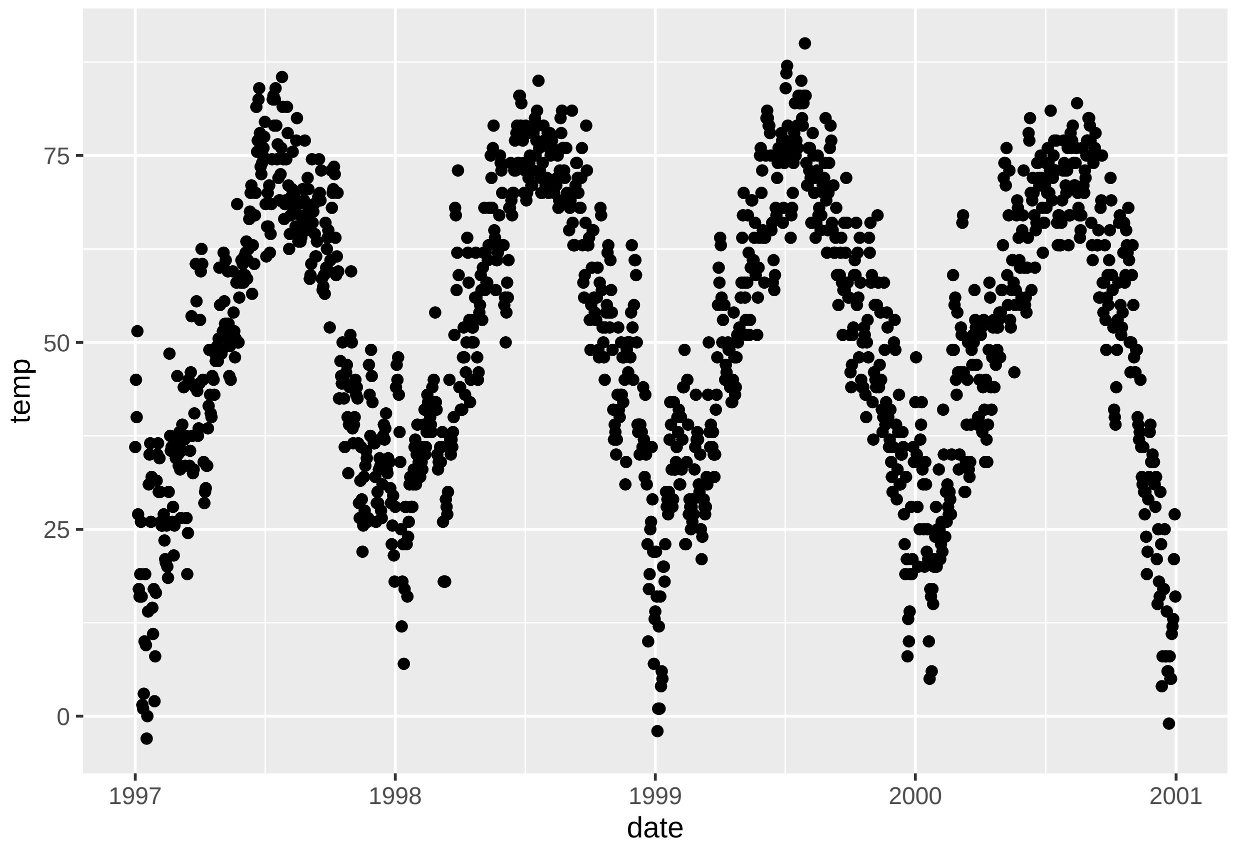
```
(g <- ggplot(chic, aes(x = date, y = temp)))
```



2nd: Now we need to provide geometry, so that ggplot knows how we want to plot that data!

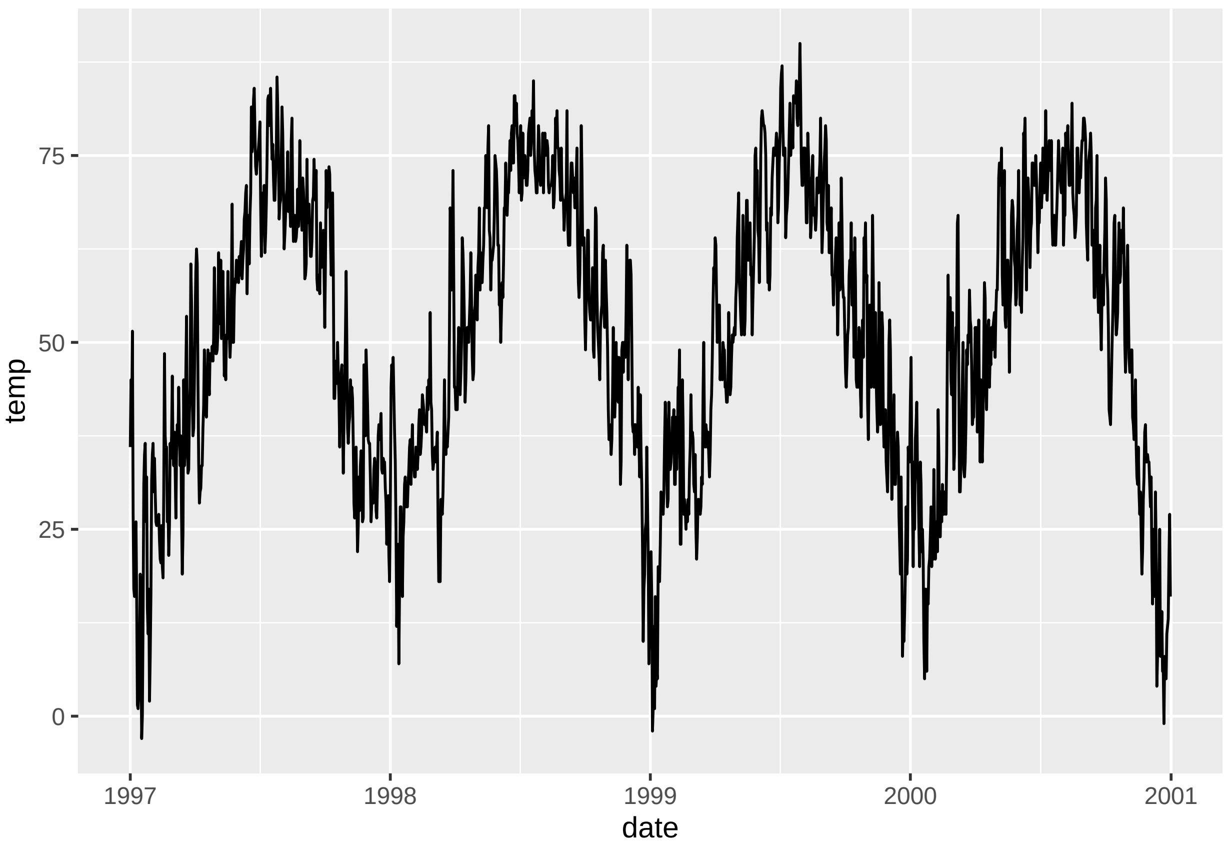
geom_point() to create a scatter plot:

```
# geom_point() to create a scatter plot:
g + geom_point()
```



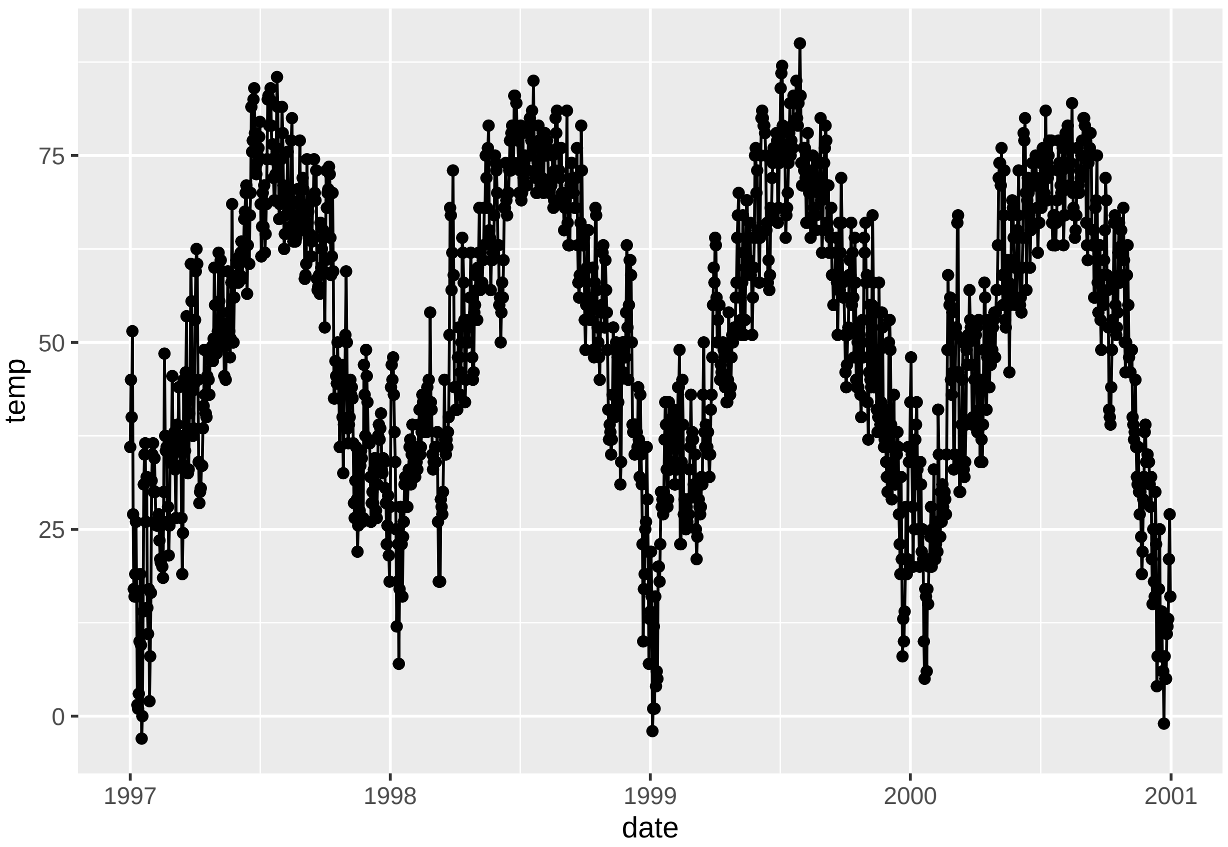
geom_line() to create a line plot (not optimal though):

```
# geom_line() to create a line plot (not optimal though):
g + geom_line()
```



combine both:

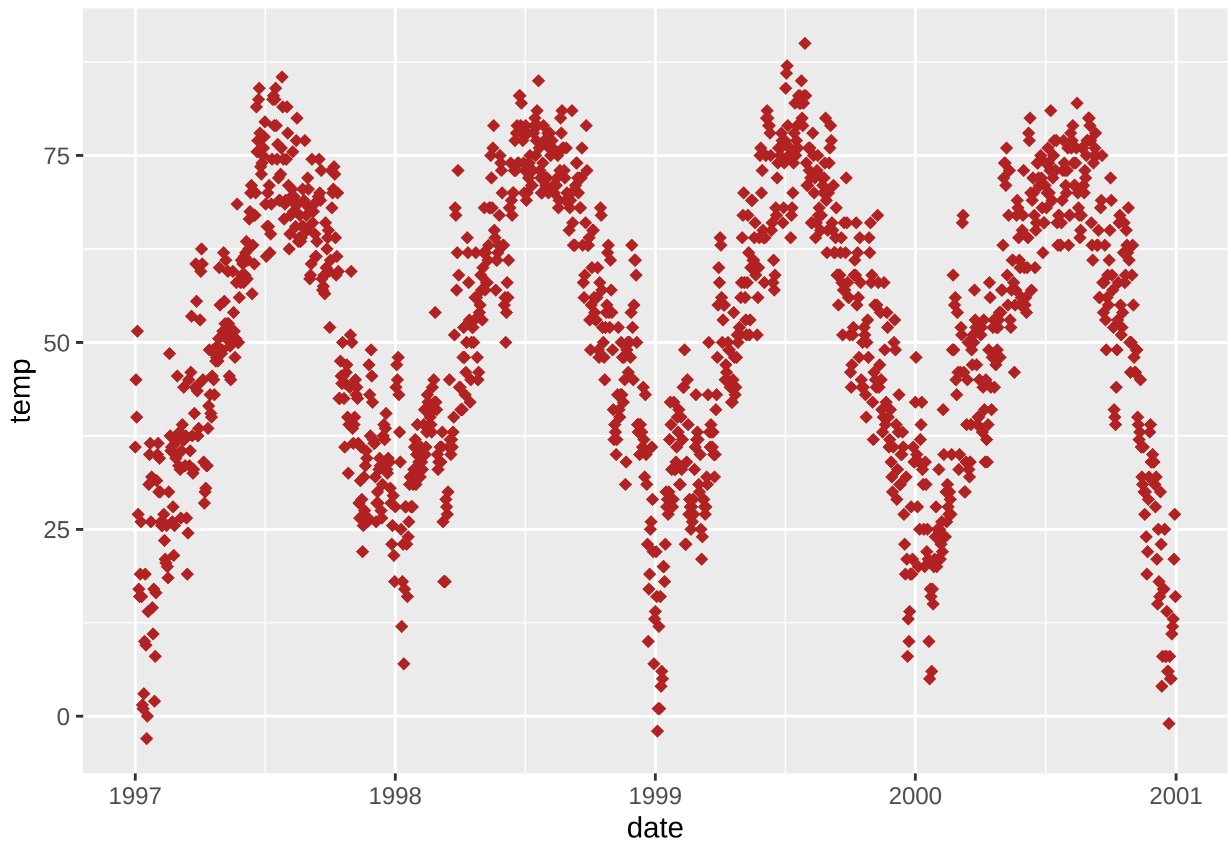
```
# combine both:
g + geom_line() + geom_point()
```



3rd: Change Properties of Geometries

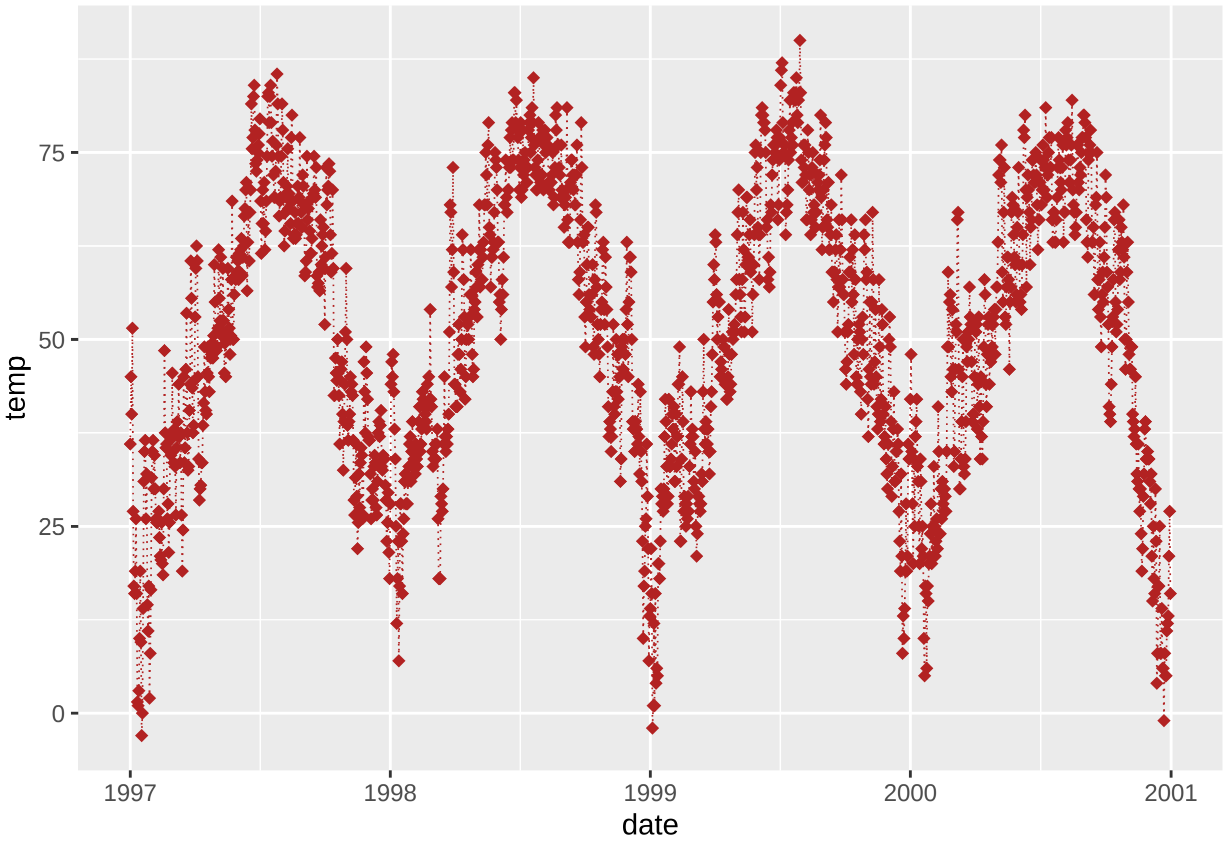
Within the geom_* , you can manipulate visual aesthetics such as the color, shape, and size of your points

```
g + geom_point(color = "firebrick", shape = "diamond", size = 2)
```



Each geom comes with its own properties (called arguments) and the same argument may result in a different change depending on the geom you are using.

```
g + geom_point(color = "firebrick", shape = "diamond", size = 2) +
    geom_line(color = "firebrick", linetype = "dotted", size = .3)
```

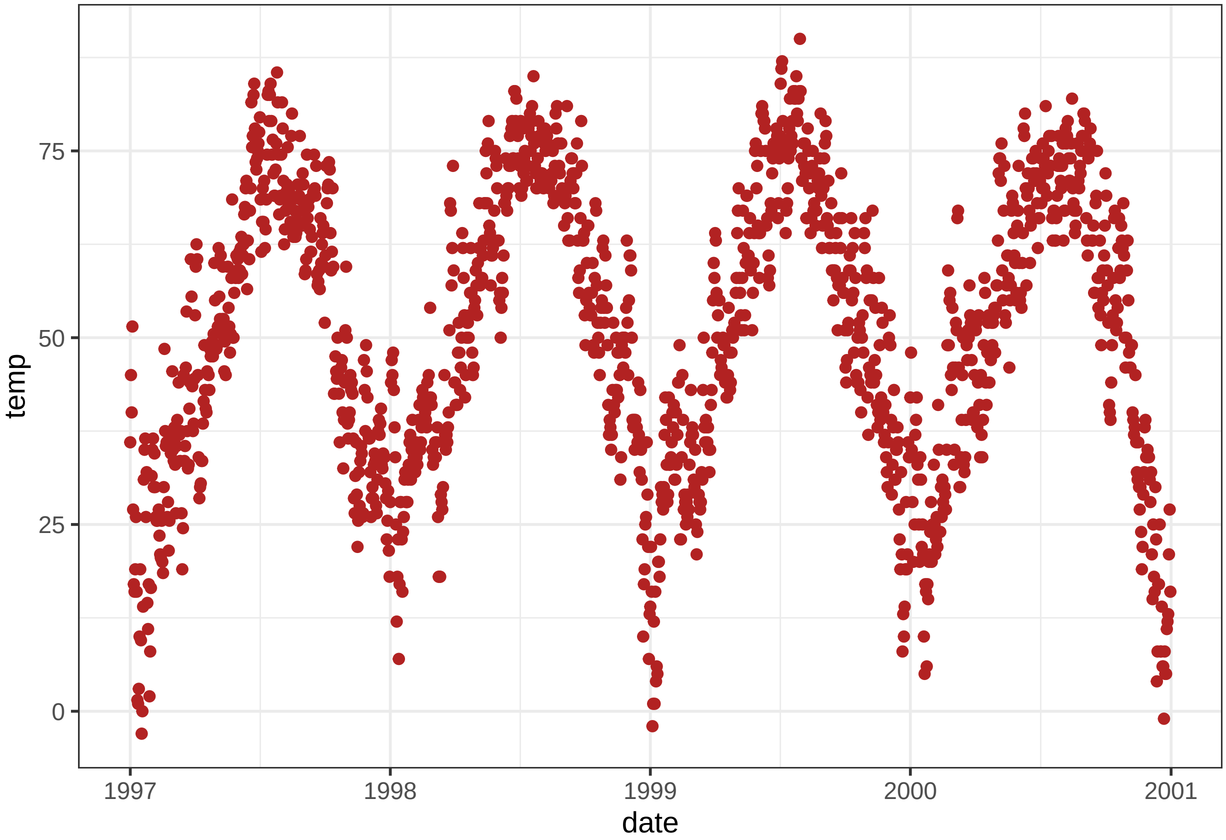


4th: Replace the default ggplot2 theme

And to illustrate some more of ggplot's versatility, let's get rid of the grayish default {ggplot2} look by setting a different built-in theme, e.g. theme_bw()—by calling theme_set() all following plots will have the same black'n'white theme. The red points look way better now!

```
theme_set(theme_bw())

g + geom_point(color = "firebrick")
```



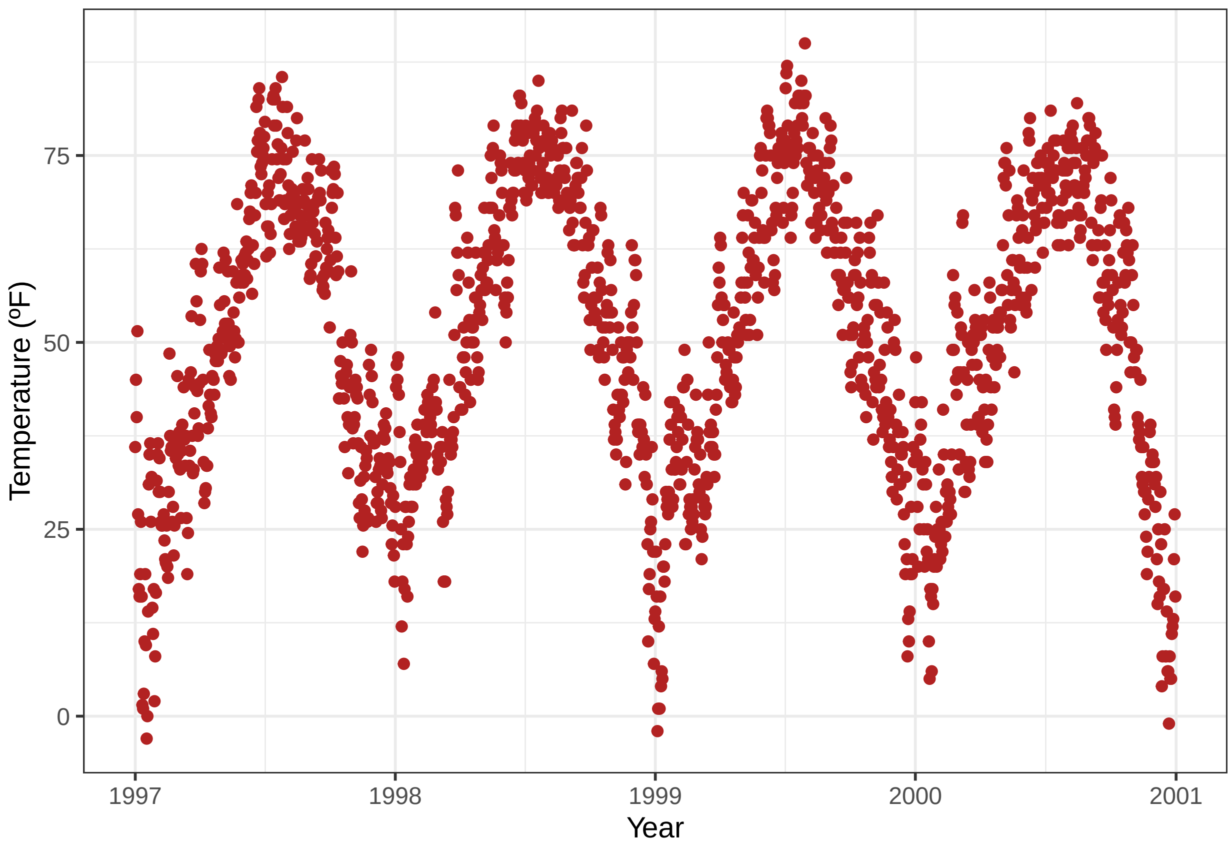
`theme()` is an essential command to manually modify all kinds of theme elements (texts, rectangles, and lines).

Working with Axes

Change Axis Titles

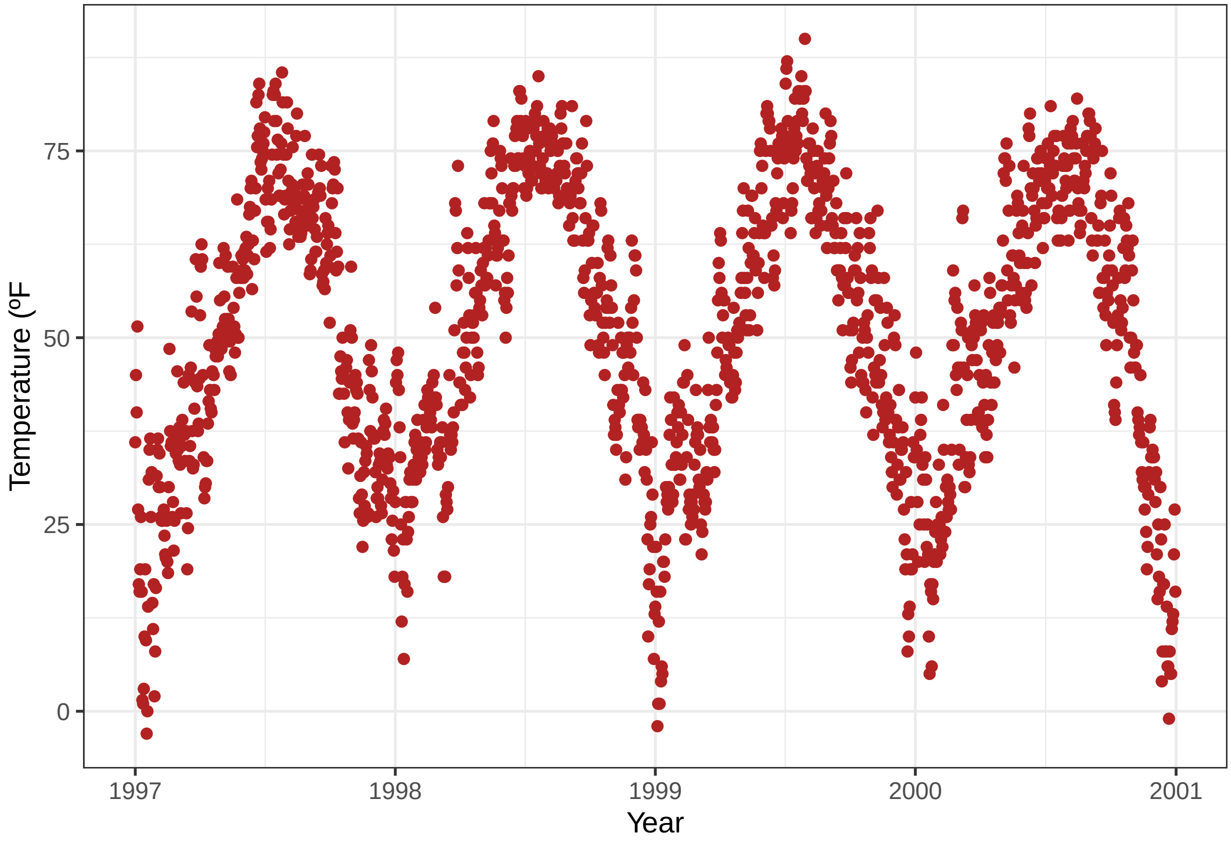
the labs() command provides a character string for each label we want to change (here x and y):

```
ggplot(chic, aes(x = date, y = temp)) +
  geom_point(color = "firebrick") +
  labs(x = "Year", y = "Temperature (ºF)")
```



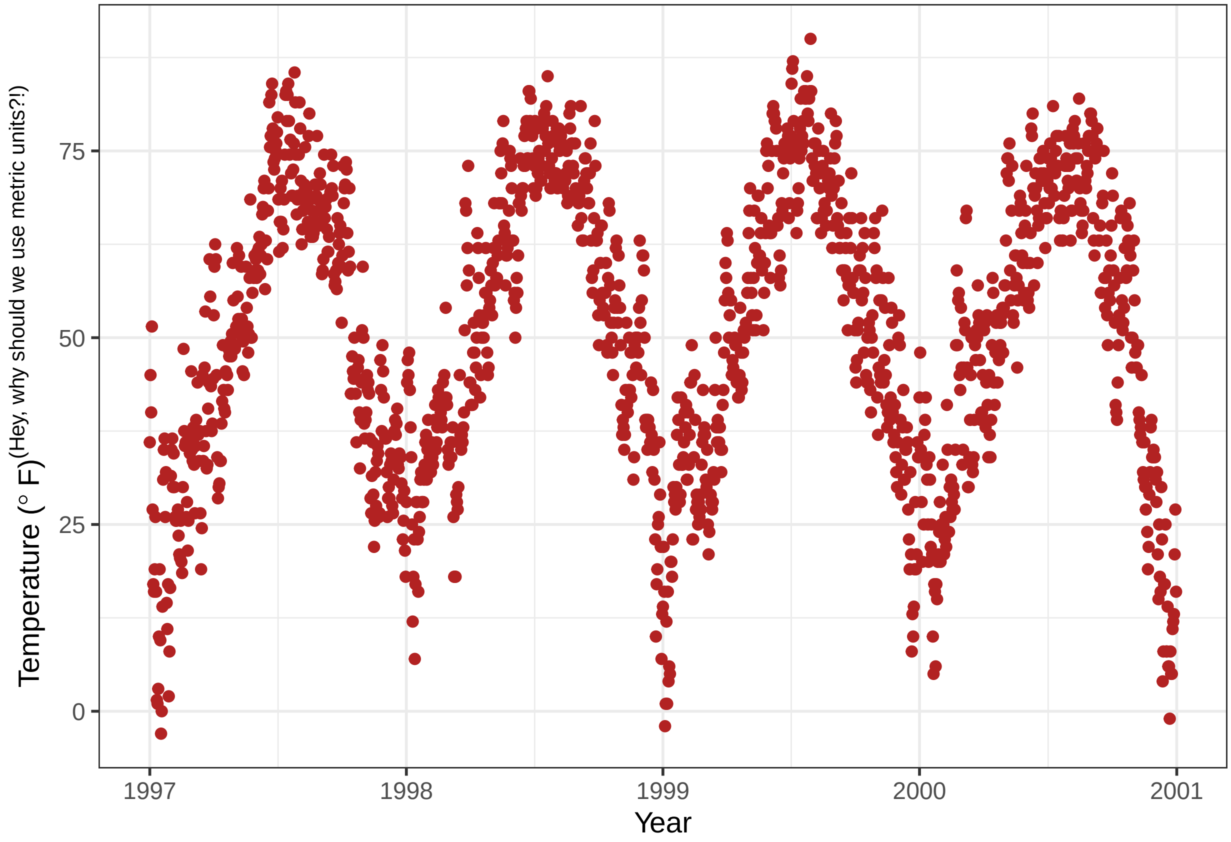
You can also add each axis title via xlab() and ylab() Example:

```
ggplot(chic, aes(x = date, y = temp)) +
  geom_point(color = "firebrick") +
  xlab("Year") +
  ylab("Temperature (ºF")
```



The code below also allows to add not only symbols but e.g. superscripts:

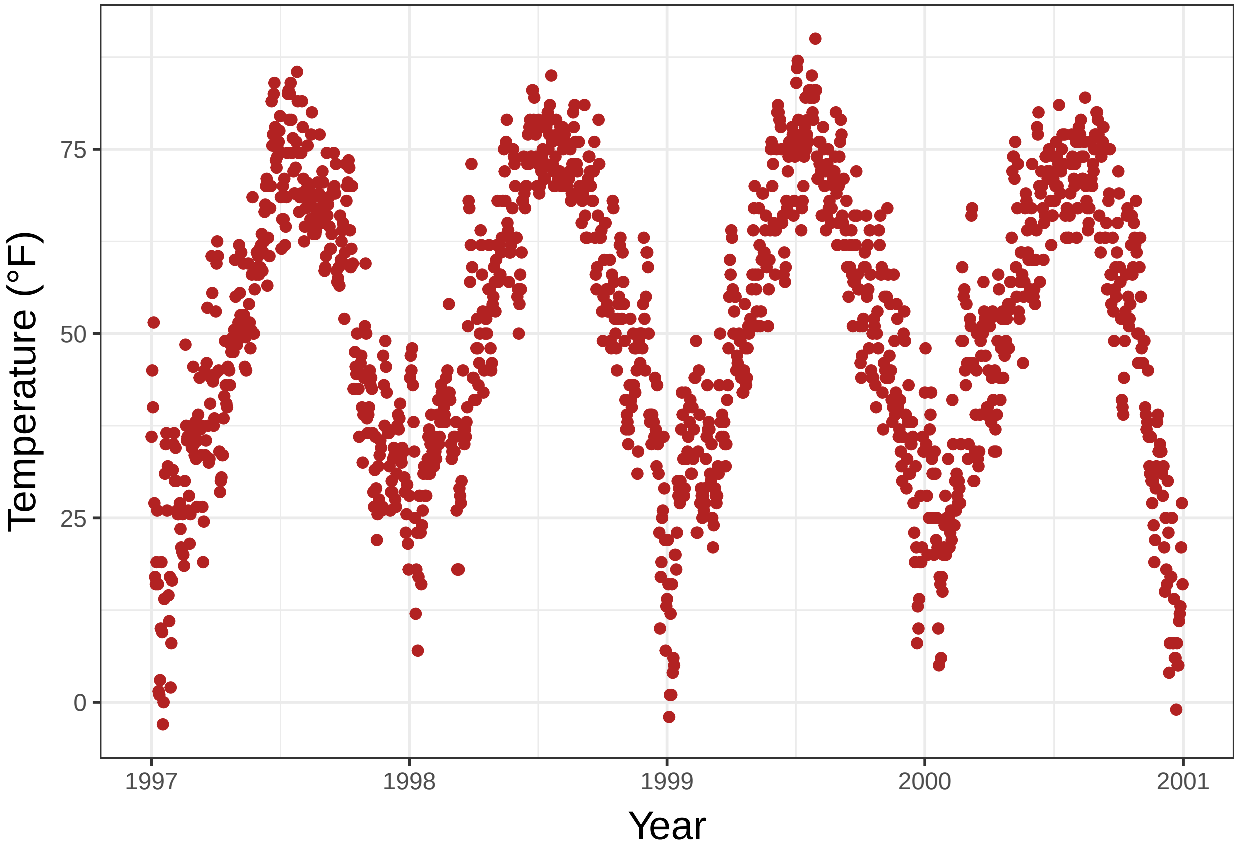
```
ggplot(chic, aes(x = date, y = temp)) +
  geom_point(color = "firebrick") +
  labs(x = "Year", y = expression(paste("Temperature (", degree ~ F, ")"^"(Hey, why should we use metri
```



Increase Space between Axis and Axis Titles

We can change the properties of all or particular text elements (here axis titles) by overwriting the default element_text() within the theme() call:

```
ggplot(chic, aes(x = date, y = temp)) +
  geom_point(color = "firebrick") +
  labs(x = "Year", y = "Temperature (°F)") +
  theme(axis.title.x = element_text(vjust = 0, size = 15),
        axis.title.y = element_text(vjust = 2, size = 15))
```


Temperature (°F)
75
50
25
0
1997
1998
1999
2000
2001
Year