Python & R For Data Science: Project Task 2

Dataset 2: Free Weather API

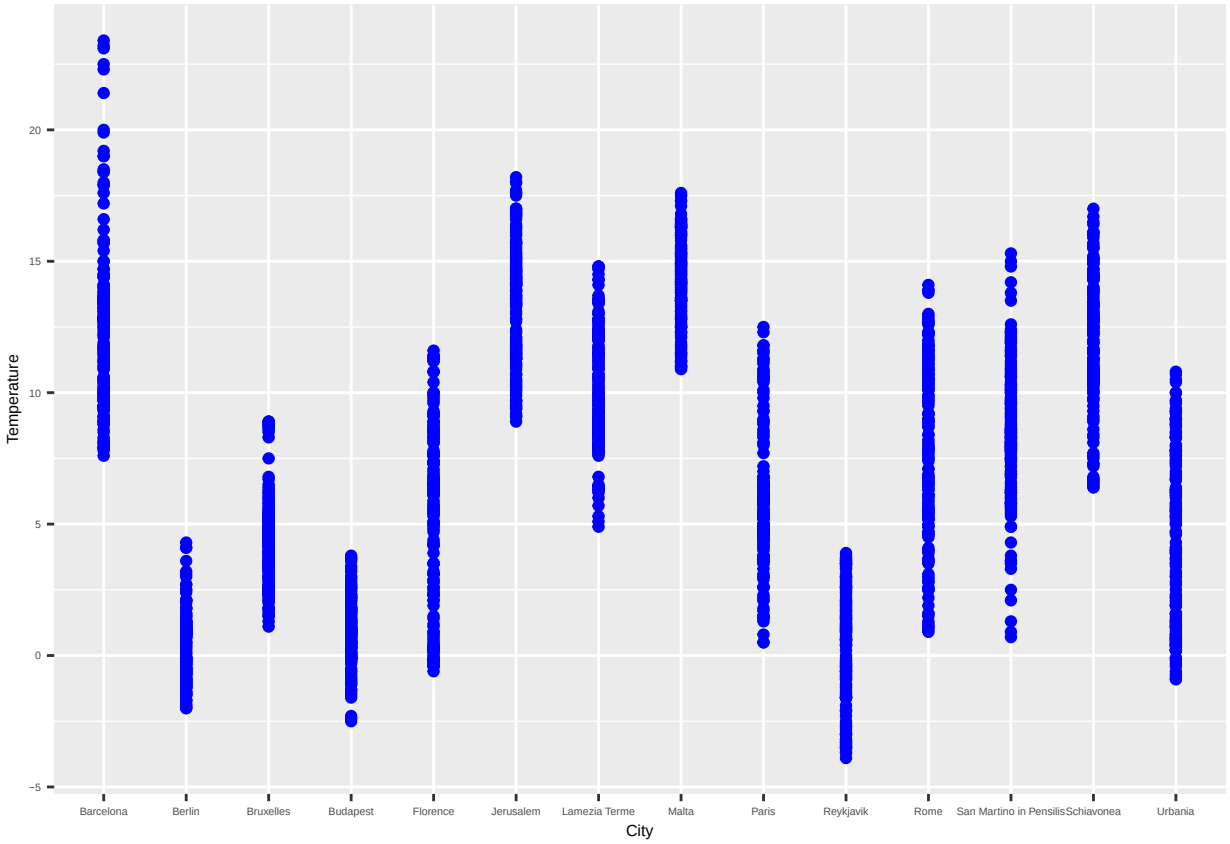
We use the Weather dataset from https://github.com/CP050/Progetto-Luiss.git. In this dataset, there are 2352 observations of 9 variables: city, time, temperature, apparent.temperature, relative.humidity, precipitation, dewpoint, freeing.level, and pressure. There are 168 observations in each city, which is the 7 days hourly data from 2021-12-04 00:00 to 2021-12-10 23:00.

First we import the weather database, the "Forecast_merged.csv" file.

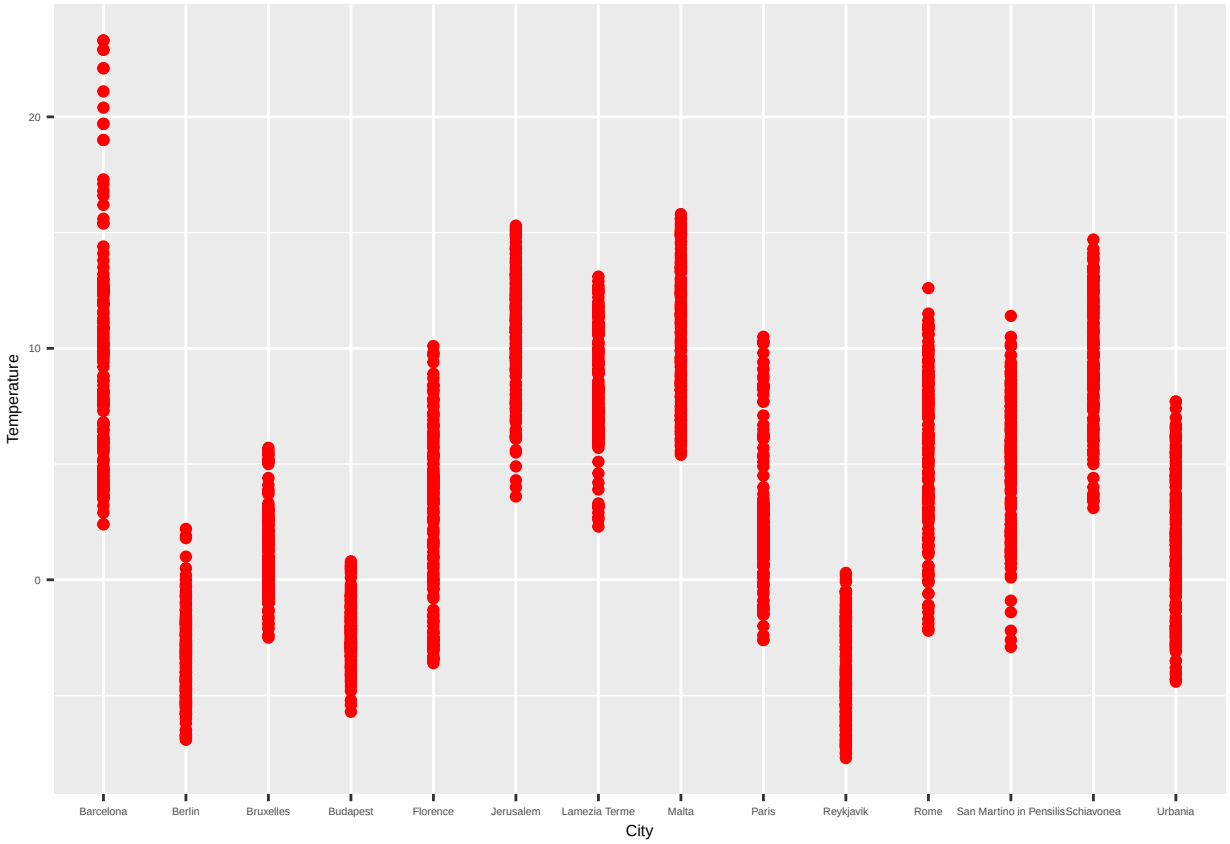
```
weather <- read.csv("Forecast_merged.csv")
View(weather)
str(weather)
summary(weather)
```

We then plot graphs with temperature in blue and apparent.temperature in red to see the range of temperature in 14 different cities.

```
ggplot(data = weather) +
  geom_point(mapping = aes(x = city, y = temperature), color = "blue")  +
  labs(y= "Temperature", x = "City") + theme(text = element_text(size= 5.6))
```




```
ggplot(data = weather) +
  geom_point(mapping = aes(x = city, y = apparent.temperature), color = "red") +
  labs(y= "Temperature", x = "City") + theme(text = element_text(size= 5.6))
```



Summarise the Variables by City

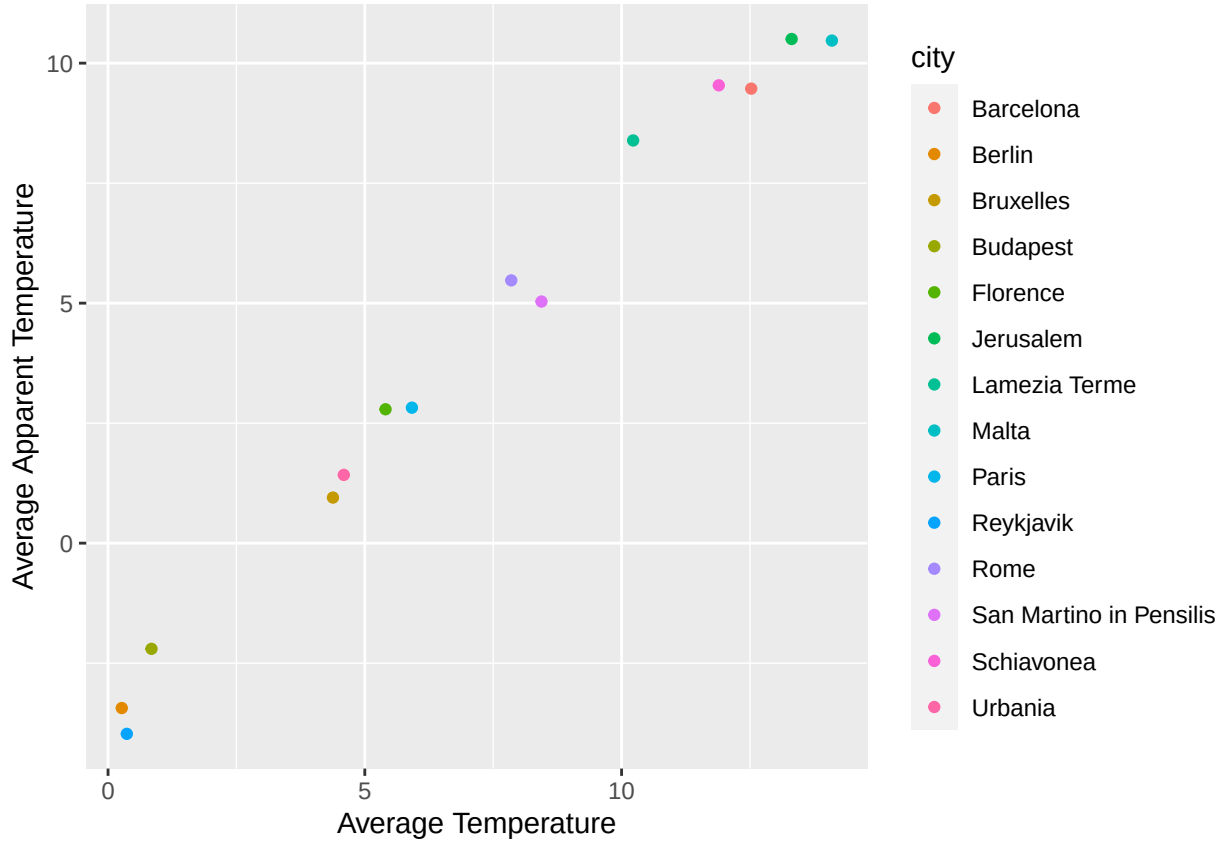
We use group_by() and summarise() function to create a table with the average data of the variables by city. We plot a graph using the average temperature and the average apparent temperature with different colors representing different cities.

```
mean_city = weather %>%
  group_by(city) %>%
  summarise(
    temp = mean(temperature, na.rm = TRUE),
    aptemp = mean(apparent.temperature, na.rm = TRUE),
    rehumid = mean(relative.humidity, na.rm = TRUE),
    precipit = mean(precipitation, na.rm = TRUE),
    dewpo = mean(dewpoint, na.rm = TRUE),
    freelv = mean(freeing.level, na.rm = TRUE),
    press = mean(pressure, na.rm = TRUE)
  )
print(mean_city)
```

```
## # A tibble: 14 x 8
##    city                    temp aptemp rehumid precipit dewpo freelv press
##    <chr>                  <dbl>  <dbl>   <dbl>    <dbl> <dbl>  <dbl> <dbl>
##  1 Barcelona              12.5    9.47    61.9  0        4.94  2131.  920.
##  2 Berlin                  0.267 -3.44    86.4  0.0575  -1.69   299. 1007.
```

```
##  3 Bruxelles                 4.38    0.95    88.6  0.170    2.59   940. 1002.
##  4 Budapest                  0.846  -2.20    84.7  0.124   -1.43   816. 1010.
##  5 Florence                  5.40    2.79    78.3  0.216    1.65  1397. 1008.
##  6 Jerusalem                13.3    10.5     59.9  0.00327  5.13  3593. 1023.
##  7 Lamezia Terme            10.2     8.39    83.6  0.228    7.95  1913. 1010.
##  8 Malta                    14.1    10.5     72.8  0.0382   8.59  2237. 1013.
##  9 Paris                     5.92    2.82    85.6  0.108    3.66  1186. 1005.
## 10 Reykjavik                 0.365  -3.97    81.9  0.295   -2.66   397.  989.
## 11 Rome                      7.85    5.48    75.6  0.0962   3.78  1640. 1008.
## 12 San Martino in Pensilis   8.44    5.03    72.7  0.0270   4.05  1707. 1008.
## 13 Schiavonea               11.9     9.54    67.8  0.0658   6.6   1926. 1009.
## 14 Urbania                   4.59    1.42    79.9  0.0245   1.41  1388. 1009.
```

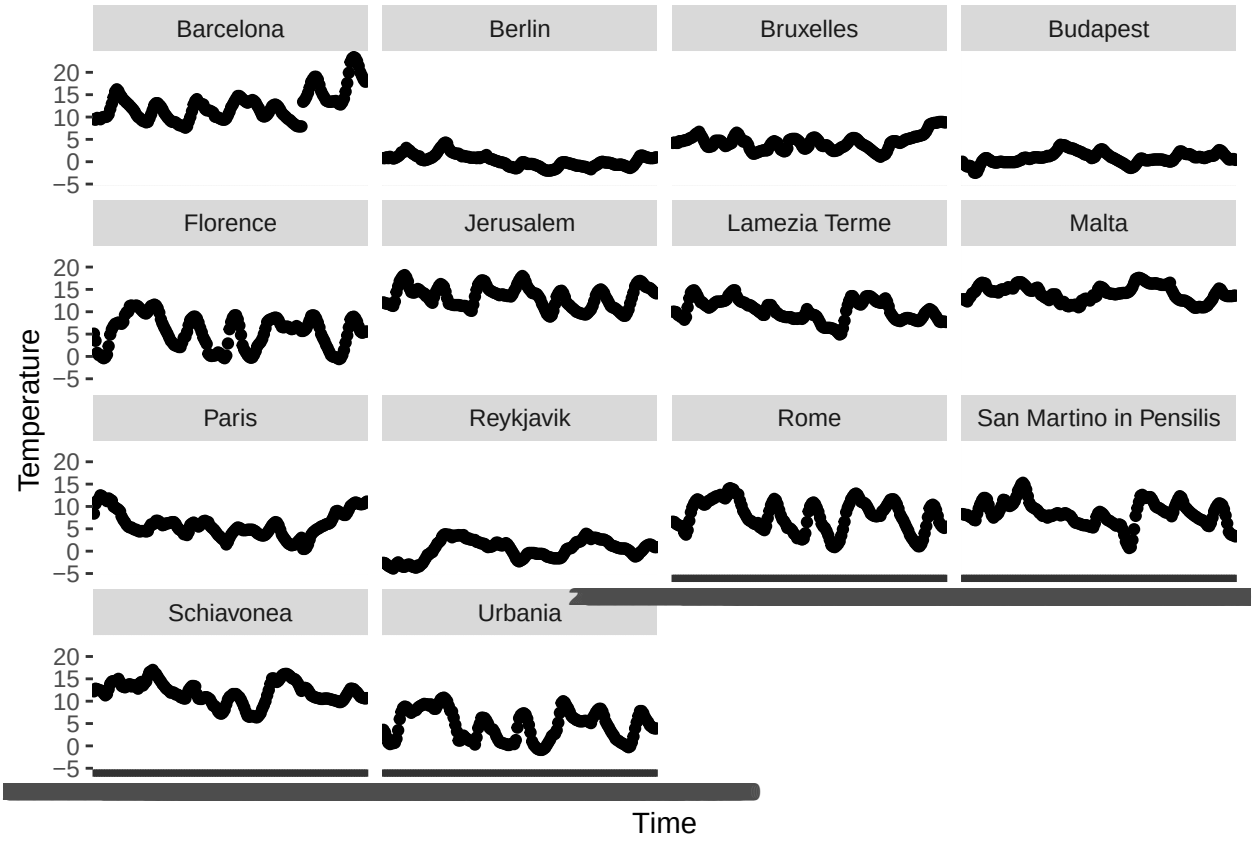
```
ggplot(data = mean_city) +
  geom_point(mapping = aes(x = temp, y = aptemp, color = city)) +
  labs(y= "Average Apparent Temperature", x = "Average Temperature")
```



Time and Temperature Trend

We plot the temperature by time, and we use facet_wrap() to create 14 graphs for all cities.

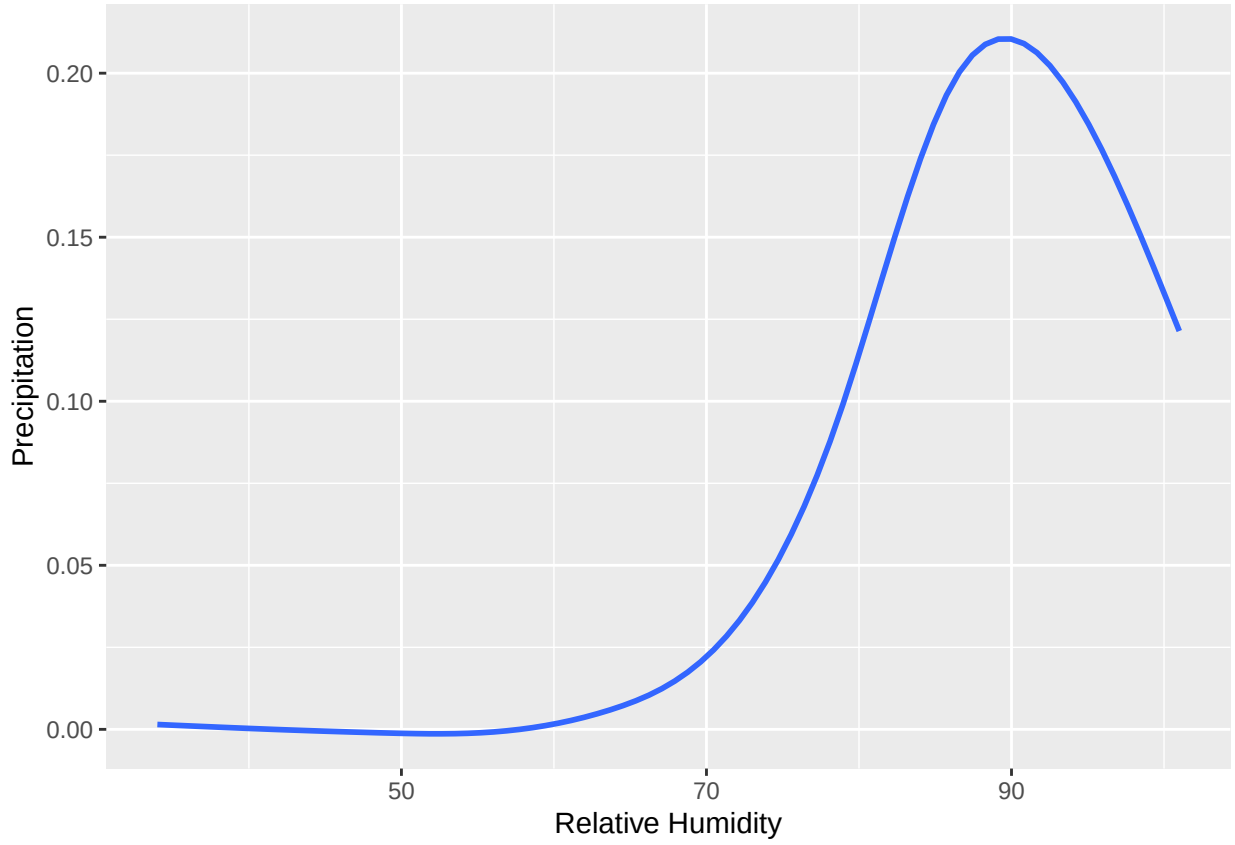
```
ggplot(data = weather) +
  geom_point(mapping = aes(x = time, y = temperature)) +
  labs(y= "Temperature", x = "Time") +
  facet_wrap(~ city, nrow = 4)
```



Relative Humidity and Precipitation

We find out that between 70% and 90%, the higher the relative humidity, the higher the precipitation.

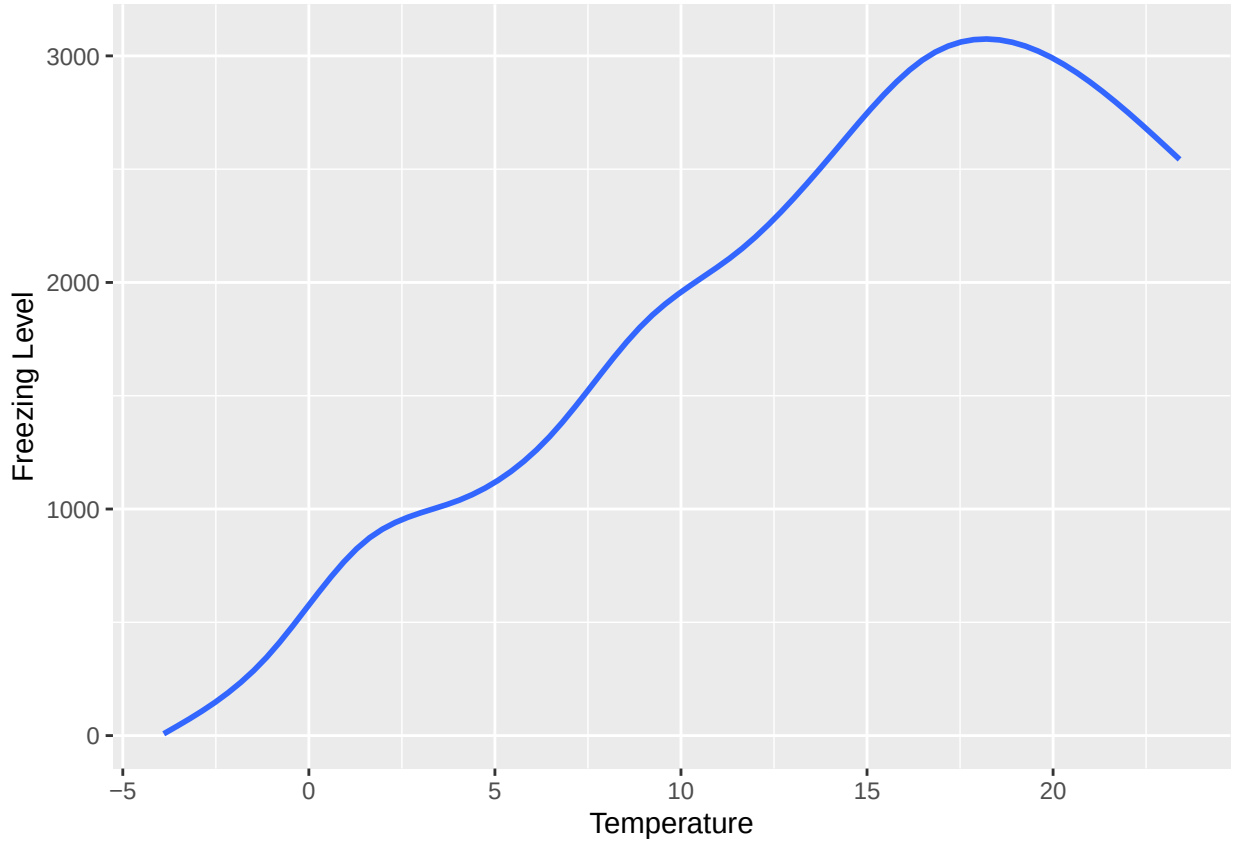
```
ggplot(data = weather) +
  geom_smooth(mapping = aes(x = relative.humidity, y = precipitation), se = FALSE) +
  labs(y= "Precipitation", x = "Relative Humidity")
```



Temperature and Freezing Level Height

The plot shows that below 17.5 degree, the higher the temperature, the greater the freezing level.

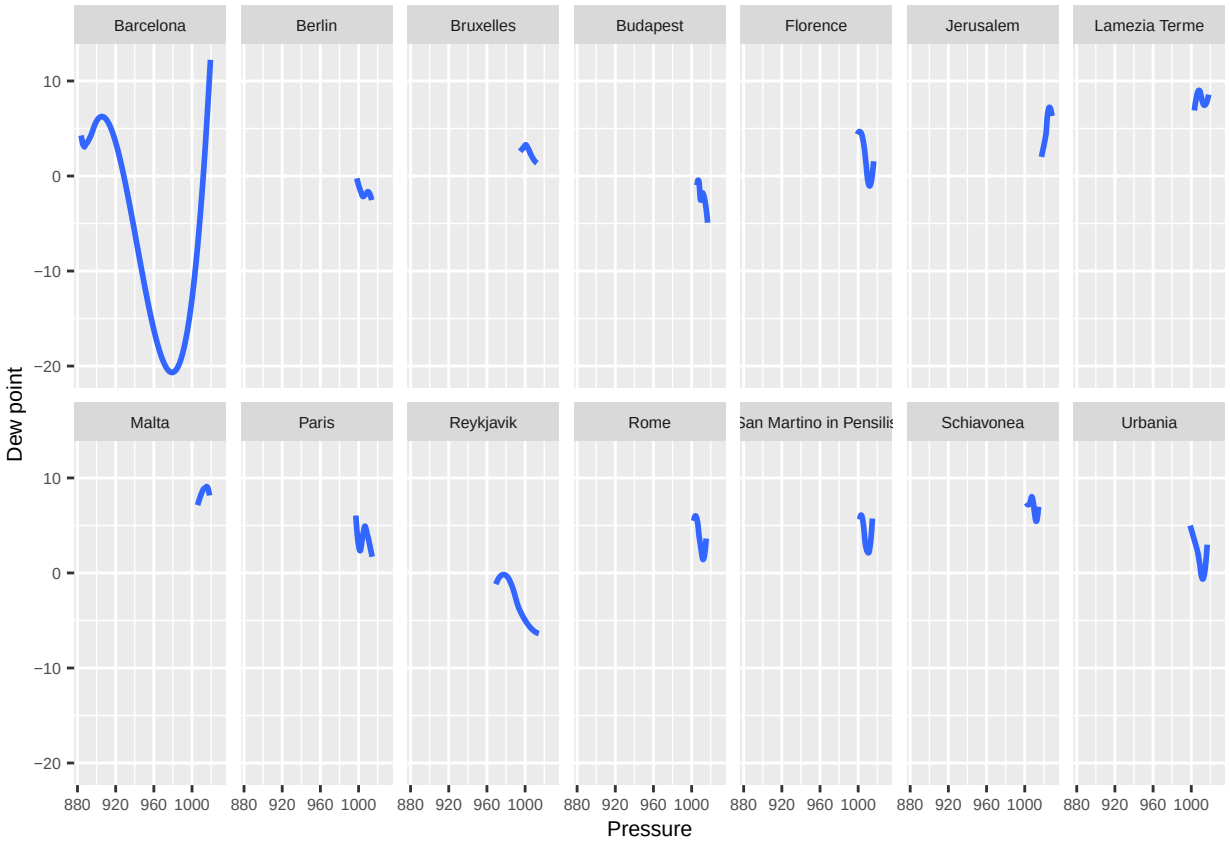
```
ggplot(data = weather) +
  geom_smooth(mapping = aes(x = temperature, y = freeing.level), se = FALSE) +
  labs(y= "Freezing Level", x = "Temperature")
```



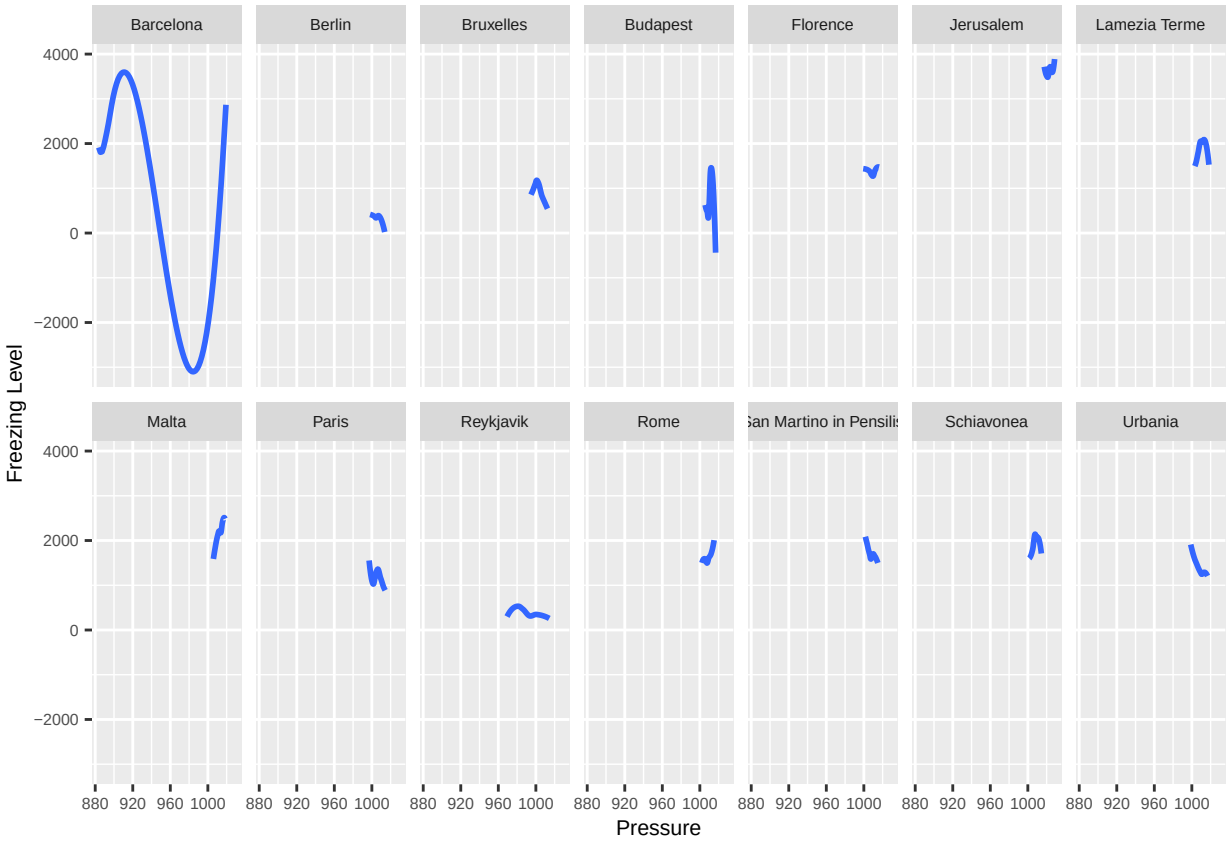
Pressure, Dew Point, and Freezing Level Height

We plot the pressure-dewpoint graph and the pressure-freeing.level graph for all 14 cities, and we decide to look into the data in Barcelona because it has the greatest range of dewpoint and freeing.level. From the Barcelona data, we know that between 920 and 980, the dew point and the freezing level height decrease when the pressure increases. This relationship reverses when the pressure is greater than 980.

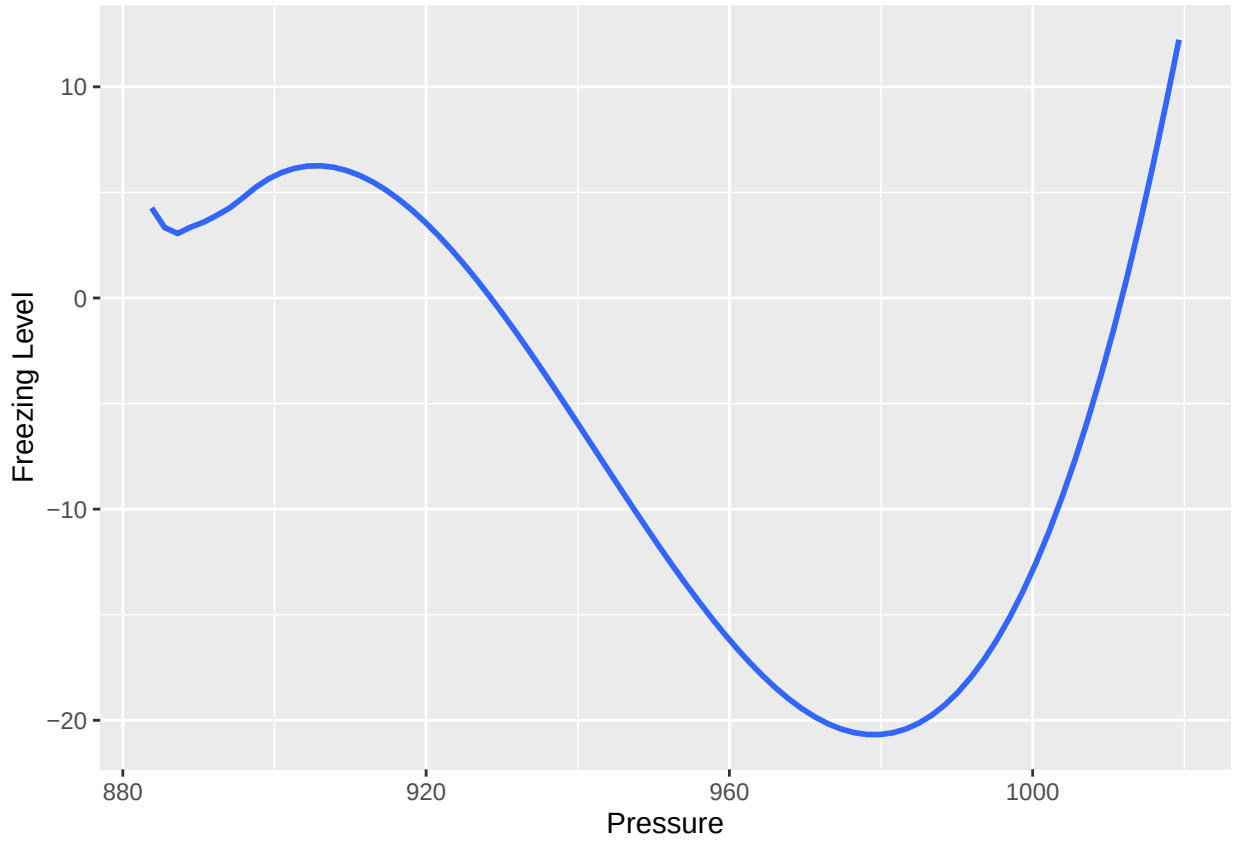
```
ggplot(data = weather) +
  geom_smooth(mapping = aes(x = pressure, y = dewpoint), se = FALSE) +
  facet_wrap(~ city, nrow = 2) + theme(text = element_text(size = 8)) +
  labs(y= "Dew point", x = "Pressure")
```



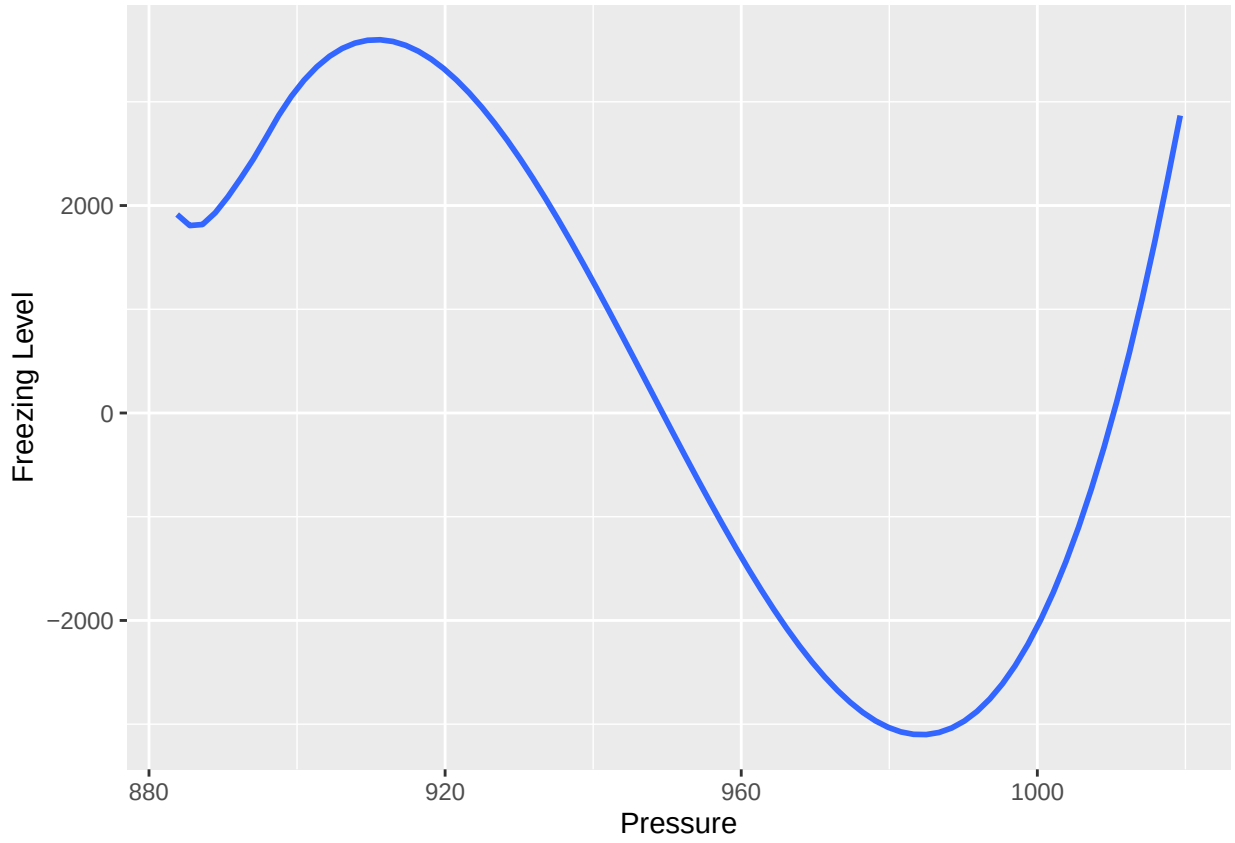
```
ggplot(data = weather) +
  geom_smooth(mapping = aes(x = pressure, y = freeing.level), se = FALSE) +
  facet_wrap(~ city, nrow = 2) + theme(text = element_text(size = 8)) +
  labs(y= "Freezing Level", x = "Pressure")
```



```
Barcelona <- weather[weather$city == "Barcelona", ]
ggplot(data = Barcelona) +
  geom_smooth(mapping = aes(x = pressure, y = dewpoint), se = FALSE) +
  labs(y= "Freezing Level", x = "Pressure")
```



```
Barcelona <- weather[weather$city == "Barcelona", ]
ggplot(data = Barcelona) +
  geom_smooth(mapping = aes(x = pressure, y = freeing.level), se = FALSE)  +
  labs(y= "Freezing Level", x = "Pressure")
```



Relative Humidity and the Temperature Difference

We add a new variable: tempdif = temperature - apparent.temperature. From the graph we know that, the higher the relative humidity, the smaller the difference in temperature and apparent temperature.

```
weather2 = weather %>%
  mutate(tempdif = temperature - apparent.temperature)
ggplot(data = weather2, mapping = aes(x = relative.humidity, y = tempdif)) +
  geom_point() +
  geom_smooth(se = FALSE)  +
  labs(y= "Temperature Difference", x = "Relative Humidity")
```