Meteorological Data of Porto Alegre

Camil Samer Zahlan Redwan

November 26, 2018

This is the final project for **Literate Programming and Statistics (CMP595)**[https://github.com/schnorr/lps] course. It is a *laboratory notebook* data exploration of a specific dataset for educational purposes.

The dataset register conventional meteorological station measures of the city of Porto Alegre starting from 01/01/1961 to 31/7/2018. The data has been made freely available by the *Instituto Nacional de Meteorologia*[http://www.inmet.gov.br/portal/index.php?r=estacoes/estacoesConvencionais]. I made it available in a convinient format for downloading in my personal github[https://github.com/camilz/lps/blob/camilz-patch-1/porto-alegre-metereological-data.csv].

The variables of the meteorological *dataset* are: - **Station** code of the conventional meteorological station; - **Date** of the observation; - **Hour** of the observation; - **Preciptation** measured in milimiter (mm); - **MaxTemperature** maximum temperature registred from 00:00 to 24:00 (that is, Max temperature in a 24 hours interval from midnight to midnight) in Celsius (ºC); - **MinTemperature**; minimum temperature registred from 12:00 to 12:00 (that is, Max temperature in a 24 hours interval from noon to noon) in Celsius (ºC); - **Insolation**; hours of light - **Evaporation**; - **MeanTemperature**; mean temperature registred from 00:00 to 00:00 (that is, the average temperature in a 24 hours interval from midnight to midnight) in Celsius (ºC); - **MeanRelativeHumidity**; in percentage and varies from 0% to 100%; - **MeanWindVelocity**; measured in meters per second (m/s)

Although theer are 8 variables that describe the meteorological conditions at a given time and date, I mainly consider for the purposes of this analysis the variables **MeanTemperature, MaxTemperature, MinTemperature** and **Preciptation**.

The questions that I try to answer are the following: – What are the average temperature (monthly, yearly and by decade) – The average precipitation (monthly, yearly and by decade); - Is there a global trend in Porto Alegre's weather reagarding temperature and preciptation ? - How the does the weather changes regarding the seasons? Specificaly: – Does the average winter's temperature is getting lower ? – Does the average summer's temperature is getting higher ? - When the top 10 lowest and higher temperatures were registred?

Preparing the Packages

```
library(readr)
library(magrittr)
library(ggplot2)
library(ggridges)
```

```
##
## Attaching package: 'ggridges'
```

```
## The following object is masked from 'package:ggplot2':
##
##     scale_discrete_manual
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(timeSeries)
```

```
## Loading required package: timeDate
```

```
library(viridis)     ## color palette
```

```
## Loading required package: viridisLite
```

```
library(hrbrthemes)  ## plot theme
```

```
## NOTE: Either Arial Narrow or Roboto Condensed fonts are *required* to use these themes.
##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and
##       if Arial Narrow is not on your system, please see http://bit.ly/arialnarrow
```

```
library(ggjoy)
```

```
## The ggjoy package has been deprecated. Please switch over to the
## ggridges package, which provides the same functionality. Porting
## guidelines can be found here:
## https://github.com/clauswilke/ggjoy/blob/master/README.md
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:timeSeries':
##
##     filter, lag
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(RColorBrewer)
```

Reading the Data

Note that by passing to *col_types* the precise type of each variable, we can be sure that, if no warning pop-up, the value of each column is correct. The name of the columns like **Estacao, Data, Hora, Precipitacao,...** are yet in portuguese.

```
file = "porto-alegre-metereological-data.csv"
if(!file.exists(file)){
  download.file("https://github.com/camilz/lps/tree/camilz-patch-1/porto-alegre-metereological-data.csv
    destfile=file)
```

```
}

all_content <- readLines(file);

df <- read_csv(file, locale = locale(encoding = "UTF-8"),
               col_types=cols(Estacao = col_integer(),
                              Data = col_date(format="%d/%m/%Y"),
                              Hora = col_factor(levels = c("00:00", "12:00")),
                              Precipitacao = col_double(),
                              TempMaxima = col_double(),
                              TempMinima = col_double(),
                              Insolacao = col_double(),
                              Evaporacao_Piche = col_double(),
                              Temp_Comp_Media = col_double(),
                              Umidade_Relativa_Media = col_double(),
                              Velocidade_do_Vento_Media = col_double()
));

df;
```

```
## # A tibble: 38,629 x 11
##    Estacao Data       Hora  Precipitacao TempMaxima TempMinima Insolacao
##      <int> <date>     <fct>        <dbl>      <dbl>      <dbl>     <dbl>
##  1   83967 1961-01-01 00:00         NA         33.8       NA        11.7
##  2   83967 1961-01-01 12:00         NA         NA         22.2      NA
##  3   83967 1961-01-02 00:00         NA         34.7       NA         9.5
##  4   83967 1961-01-02 12:00          0         NA         22.5      NA
##  5   83967 1961-01-03 00:00         NA         27.7       NA         2.3
##  6   83967 1961-01-03 12:00          0.2       NA         23.1      NA
##  7   83967 1961-01-04 00:00         NA         29.4       NA         8.7
##  8   83967 1961-01-04 12:00          2.4       NA         20.6      NA
##  9   83967 1961-01-05 00:00         NA         32.5       NA        10.1
## 10   83967 1961-01-05 12:00          0         NA         18.8      NA
## # ... with 38,619 more rows, and 4 more variables: Evaporacao_Piche <dbl>,
## #   Temp_Comp_Media <dbl>, Umidade_Relativa_Media <dbl>,
## #   Velocidade_do_Vento_Media <dbl>
```

Transform and cleanup the data

The headers are in portuguese, so we transform the data to conform the new variable names according to the following:

- **Station** in place of Estacao;
- **Date** is place of Data
- **Hour** in place of Hora;
- **Preciptation** in place of Preciptacao;
- **MaxTemperature** in place of TempMaxima;
- **MinTemperature** in place of TempMinima
- **Insolation** in place of Insolacao
- **Evaporation** in place of Evaporacao_Piche
- **MeanTemperature** in place of Temp_Comp_Media
- **MeanRelativeHumidity** in place of Umidade_Relativa_Media
- **MeanWindVelocity** in place of Velocidade_do_Vento_Media

```
colnames(df) <- c("Station",
                  "Date",
                  "Hour",
                  "Preciptation",
                  "MaxTemperature",
                  "MinTemperature",
                  "Insolation",
                  "Evaporation",
                  "MeanTemperature",
                  "MeanRelativeHumidity",
                  "MeanWindVelocity")

df
```

```
## # A tibble: 38,629 x 11
##    Station Date       Hour  Preciptation MaxTemperature MinTemperature
##      <int> <date>     <fct>        <dbl>          <dbl>          <dbl>
##  1   83967 1961-01-01 00:00         NA             33.8           NA
##  2   83967 1961-01-01 12:00         NA             NA             22.2
##  3   83967 1961-01-02 00:00         NA             34.7           NA
##  4   83967 1961-01-02 12:00          0             NA             22.5
##  5   83967 1961-01-03 00:00         NA             27.7           NA
##  6   83967 1961-01-03 12:00          0.2           NA             23.1
##  7   83967 1961-01-04 00:00         NA             29.4           NA
##  8   83967 1961-01-04 12:00          2.4           NA             20.6
##  9   83967 1961-01-05 00:00         NA             32.5           NA
## 10   83967 1961-01-05 12:00          0             NA             18.8
## # ... with 38,619 more rows, and 5 more variables: Insolation <dbl>,
## #   Evaporation <dbl>, MeanTemperature <dbl>, MeanRelativeHumidity <dbl>,
## #   MeanWindVelocity <dbl>
```

As the data is in tidyformat, each observation (set of variables) is represented by row. The data type is well formated and the types is well defined due to *read_csv* and *col_type* constrains. But, still, there may be a lot of work to do.

Let's take a look at the structure

```
str(df)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    38629 obs. of  11 variables:
##  $ Station             : int  83967 83967 83967 83967 83967 83967 83967 83967 83967 83967 ...
##  $ Date                : Date, format: "1961-01-01" "1961-01-01" ...
##  $ Hour                : Factor w/ 2 levels "00:00","12:00": 1 2 1 2 1 2 1 2 1 2 ...
##  $ Preciptation        : num  NA NA NA 0 NA 0.2 NA 2.4 NA 0 ...
##  $ MaxTemperature      : num  33.8 NA 34.7 NA 27.7 NA 29.4 NA 32.5 NA ...
##  $ MinTemperature      : num  NA 22.2 NA 22.5 NA 23.1 NA 20.6 NA 18.8 ...
##  $ Insolation          : num  11.7 NA 9.5 NA 2.3 NA 8.7 NA 10.1 NA ...
##  $ Evaporation         : num  2.4 NA 4 NA 3.1 NA 3.5 NA 4.7 NA ...
##  $ MeanTemperature     : num  27.1 NA 28.1 NA 24.3 ...
##  $ MeanRelativeHumidity: num  67 NA 62.5 NA 74.2 ...
##  $ MeanWindVelocity    : num  2.33 NA 1.33 NA 2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Estacao = col_integer(),
##   ..   Data = col_date(format = "%d/%m/%Y"),
##   ..   Hora = col_factor(levels = c("00:00", "12:00"), ordered = FALSE, include_na = FALSE),
```

```
##   ..   Precipitacao = col_double(),
##   ..   TempMaxima = col_double(),
##   ..   TempMinima = col_double(),
##   ..   Insolacao = col_double(),
##   ..   Evaporacao_Piche = col_double(),
##   ..   Temp_Comp_Media = col_double(),
##   ..   Umidade_Relativa_Media = col_double(),
##   ..   Velocidade_do_Vento_Media = col_double()
##   .. )
```

We can already see that *missing values* are present. A lot of missing values. For instance, **MaxTemperature** is from type num and the first values are *33.8 NA 34.7 NA 27.7 NA 29.4 NA 32.5 NA*. We see a clear pattern of *missing values* and numerical values.

As expected, **Station** is of integer type, **Date** is a *Date* type, **Hour** assumes values *00:00* or *12:00* (categorically). All other variables are numbers.

Now we take a look at the summary our dataset. Summary is useful for detecting anomalous values and for estimate the rate of missing items.

```
summary(df)
```

```
##     Station           Date                Hour        Preciptation
##  Min.   :83967   Min.   :1961-01-01   00:00:19315   Min.   :  0.000
##  1st Qu.:83967   1st Qu.:1974-04-22   12:00:19314   1st Qu.:  0.000
##  Median :83967   Median :1991-06-17                 Median :  0.000
##  Mean   :83967   Mean   :1990-01-07                 Mean   :  3.789
##  3rd Qu.:83967   3rd Qu.:2005-05-12                 3rd Qu.:  1.700
##  Max.   :83967   Max.   :2018-07-31                 Max.   :149.600
##                                                     NA's   :19325
##  MaxTemperature  MinTemperature    Insolation      Evaporation
##  Min.   : 7.40   Min.   :-0.2    Min.   : 0.000   Min.   : 0.000
##  1st Qu.:21.20   1st Qu.:12.5    1st Qu.: 2.400   1st Qu.: 1.300
##  Median :25.50   Median :16.2    Median : 7.000   Median : 2.200
##  Mean   :25.21   Mean   :15.7    Mean   : 6.072   Mean   : 2.481
##  3rd Qu.:29.40   3rd Qu.:19.4    3rd Qu.: 9.200   3rd Qu.: 3.300
##  Max.   :40.60   Max.   :27.9    Max.   :13.200   Max.   :20.700
##  NA's   :19344   NA's   :19345   NA's   :19456    NA's   :20018
##  MeanTemperature MeanRelativeHumidity MeanWindVelocity
##  Min.   : 5.16   Min.   :37.50        Min.   :   0.000
##  1st Qu.:16.30   1st Qu.:69.25        1st Qu.:   1.000
##  Median :20.08   Median :76.50        Median :   1.800
##  Mean   :19.66   Mean   :76.45        Mean   :   9.235
##  3rd Qu.:23.40   3rd Qu.:84.00        3rd Qu.:   2.767
##  Max.   :33.70   Max.   :99.75        Max.   :6216.000
##  NA's   :19360   NA's   :19364        NA's   :19316
```

We can draw from the summary some observations: - **Station** variables only assumes the value 83967 that corresponds to Porto Alegre Meteorological Station; - The **Date** ranges from 01/01/1961 to 31/07/2018 indeed; - The number of observations made at 00:00 and at 12:00 differ by one; - **Important** there are 19325 rows in which **Preciptation** is *NA*. Also, we see that each variable from **Preciptation** on has more than 19.000 rows with *NA* values. - There are precisely 19315 rows with '00:00' **Hour** and 19314 rows with '12:00' **Hour**. - **Preciptation** is between 0 and 149. The value 0 corresponds to no rain. - The **MeanWindVelocity** assumes a maximum value that is clearly an mistake. This maximum velocity is 6216 m/s, which corresponds to 22377,6 km/h, but thi is 18 times the velocity of sound. Also, considering that the wind velocity of a devastating hurricane is approximately 150 m/s ~ 540 km/h, we see that this wind velocity is totally inconsistent.

Since I will not use the values of **MeanWindVelocity, Insolation, Evaporation** and **MeanRelativeHumidity**, no attempt is made to clean up this variables.

Since **Station** variable assumes only the value 83967, as we already saw in the summary, there is no reason to keep it in the dataset. First, let be sure that only **Station** is a constant column.

```
#names(df[, sapply(df, function(v) var(v, na.rm=TRUE)==0)])
for (J in names(df)) {
  print(paste("Is", paste(J, "'", "s", sep = ""), "column constant?", all(duplicated(df[J])[-1L])))
}
```

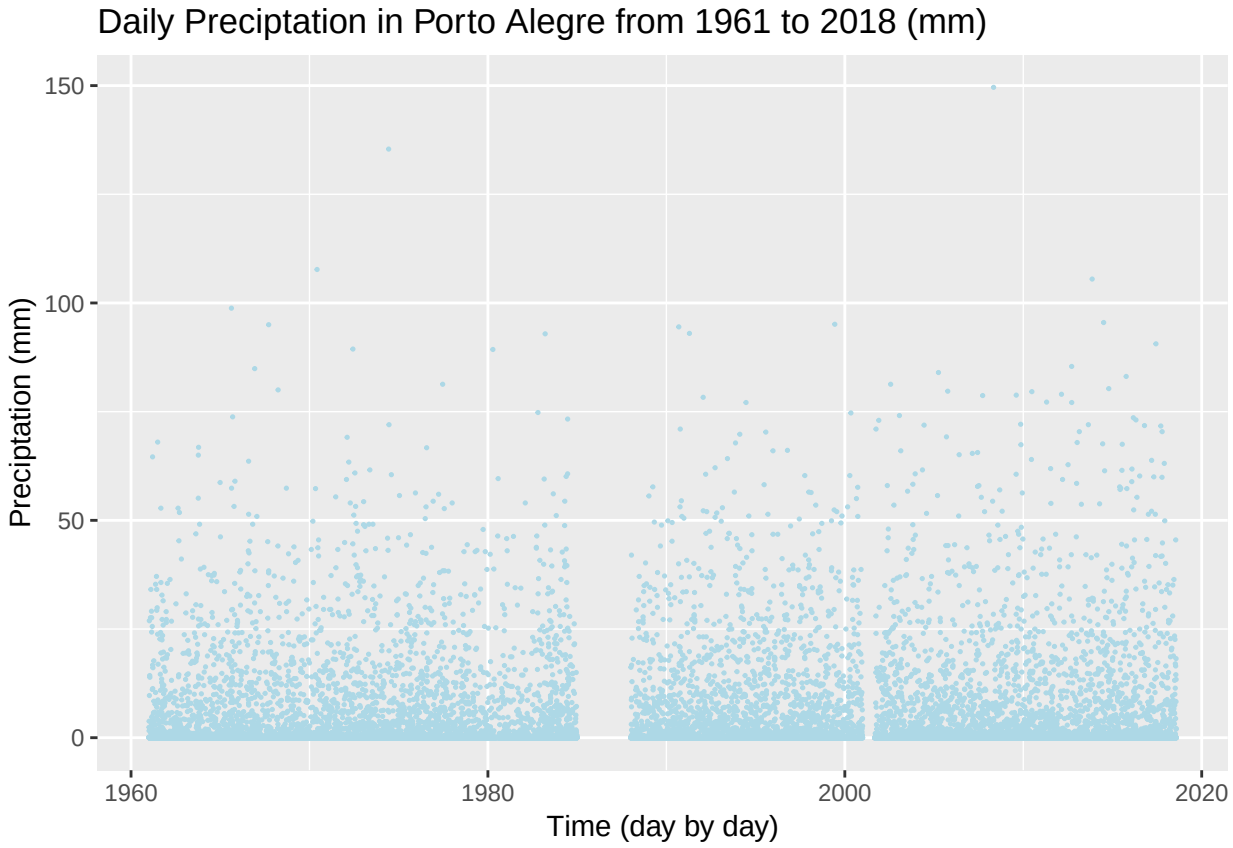
```
## [1] "Is Station's column constant? TRUE"
## [1] "Is Date's column constant? FALSE"
## [1] "Is Hour's column constant? FALSE"
## [1] "Is Preciptation's column constant? FALSE"
## [1] "Is MaxTemperature's column constant? FALSE"
## [1] "Is MinTemperature's column constant? FALSE"
## [1] "Is Insolation's column constant? FALSE"
## [1] "Is Evaporation's column constant? FALSE"
## [1] "Is MeanTemperature's column constant? FALSE"
## [1] "Is MeanRelativeHumidity's column constant? FALSE"
## [1] "Is MeanWindVelocity's column constant? FALSE"
```

Ok, if that is the case, we eliminate the **Station** column with:

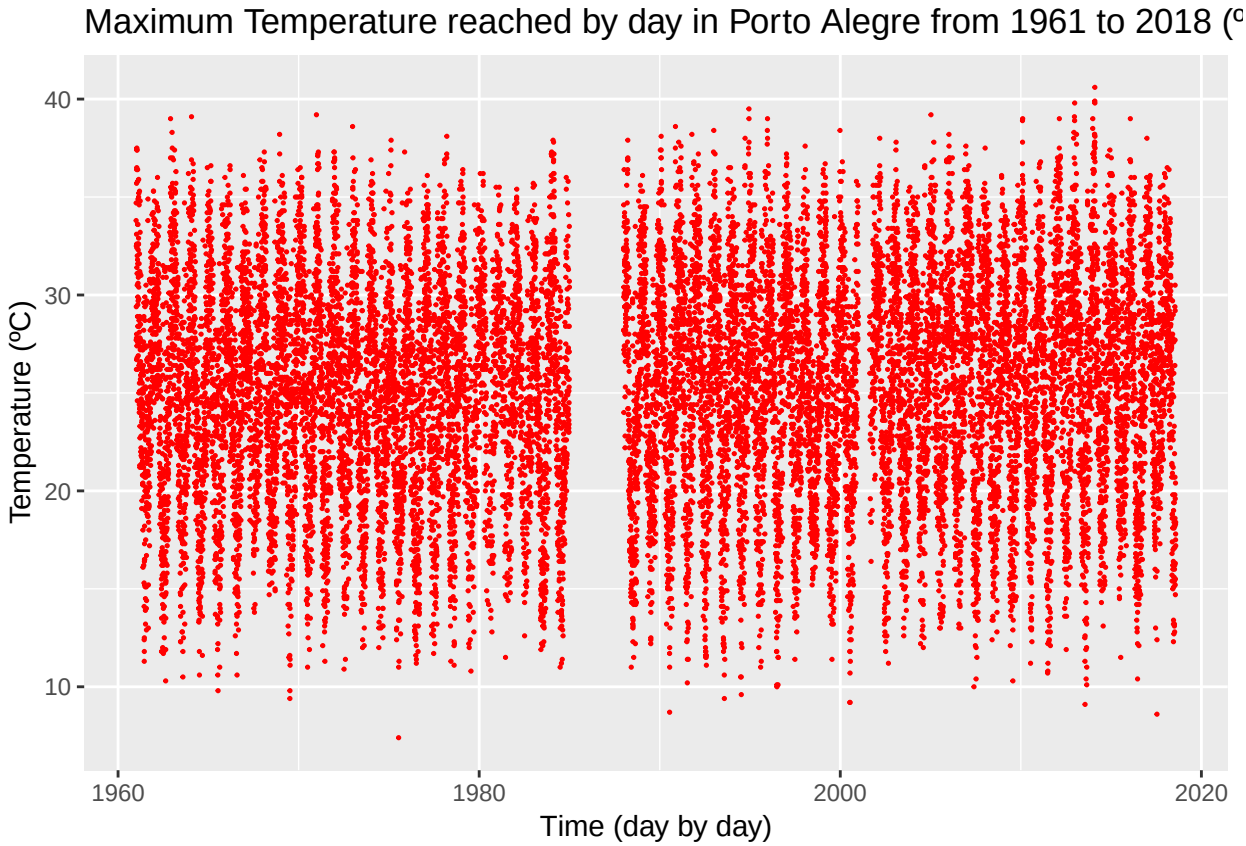
```
df$Station <- NULL
```

Now, for the first time, let's take a look in the variables **Preciptation, MinTemperature, MeanTemperature** and **MaxTemperature** visually.

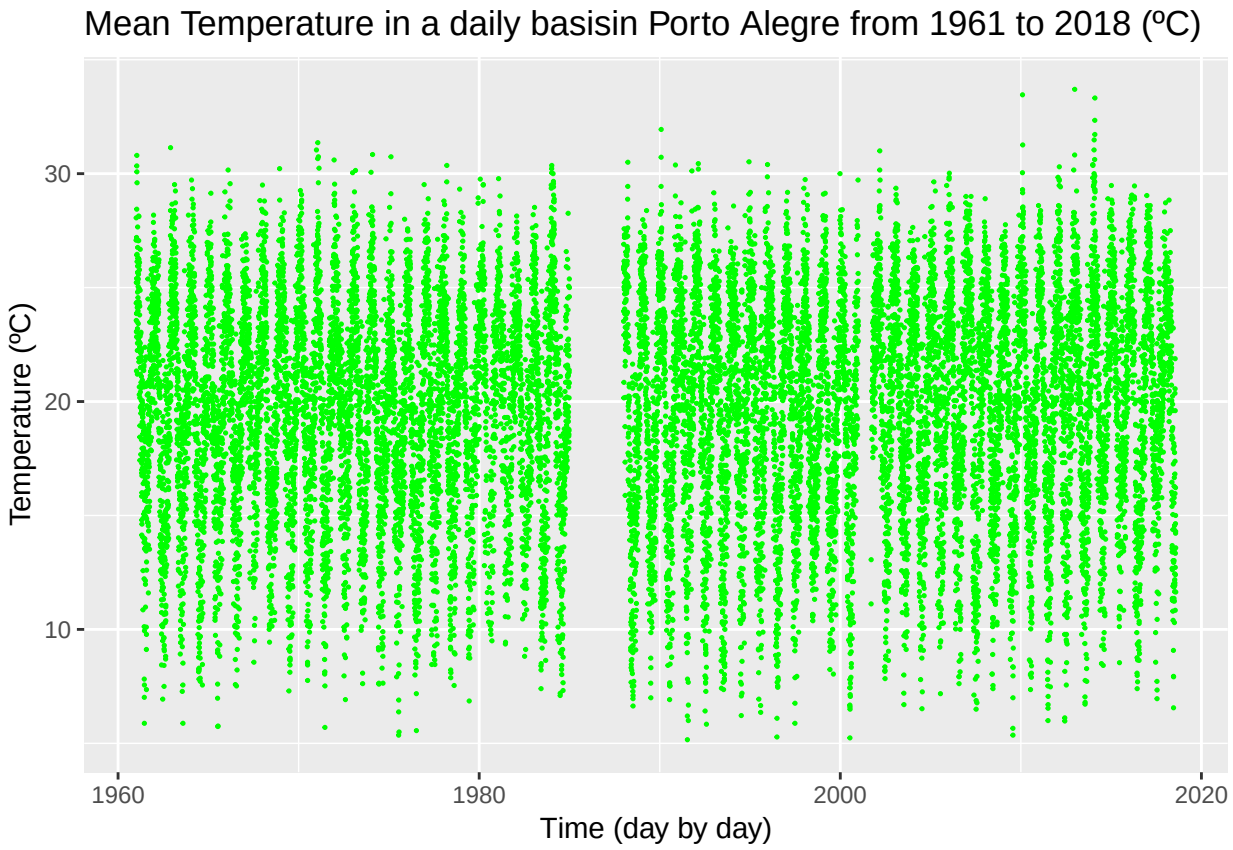
```
df %>%
  ggplot(aes(Date, Preciptation)) +
  geom_point(na.rm=TRUE, size=0.25, color="lightblue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm)")
```



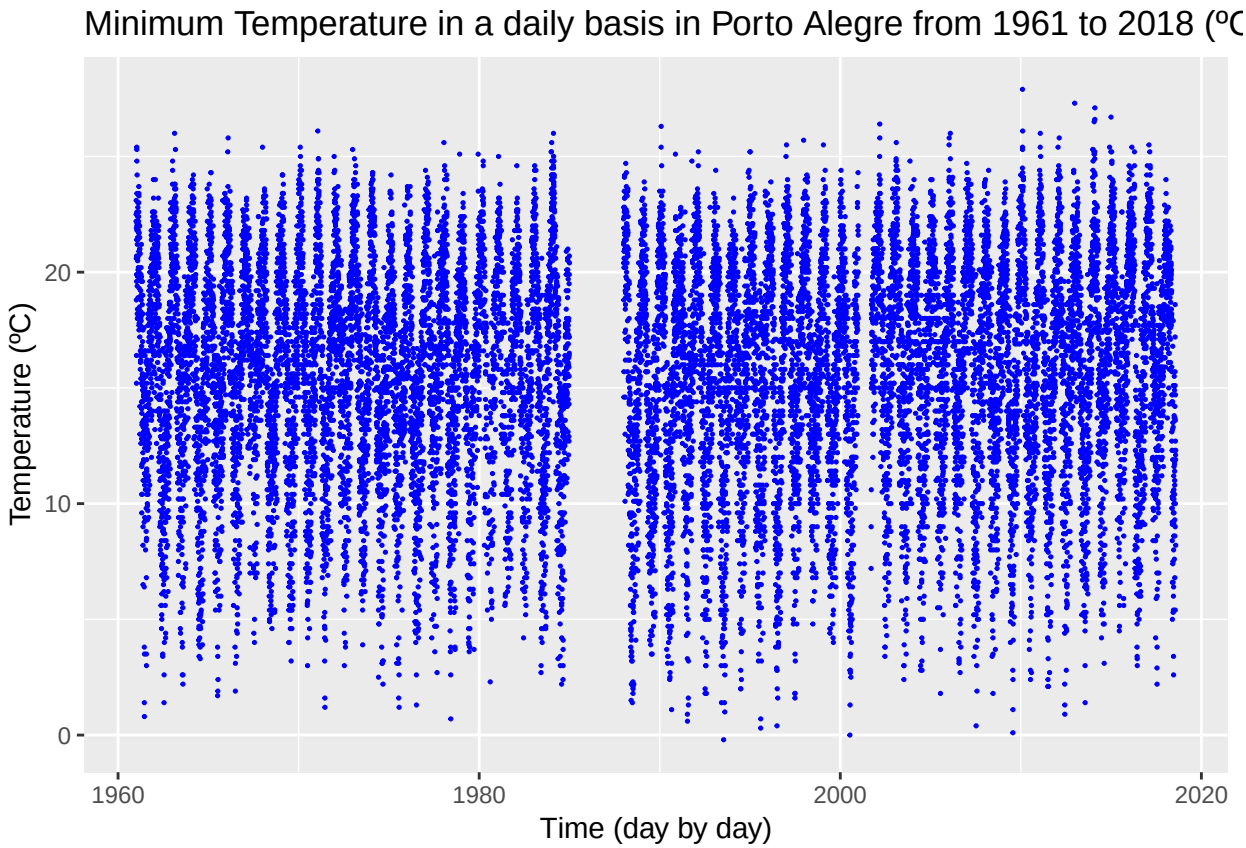

```
df %>%
  ggplot(aes(Date, MaxTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="red") +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Maximum Temperature reached by day in Porto Alegre from 1961 to 2018 (ºC)")
```




```
df %>%
  ggplot(aes(Date, MeanTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="green") +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Mean Temperature in a daily basisin Porto Alegre from 1961 to 2018 (ºC)")
```




```
df %>%
  ggplot(aes(Date, MinTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="blue") +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Minimum Temperature in a daily basis in Porto Alegre from 1961 to 2018 (ºC)")
```



We see in the first chart from above that there is *a big gap between* 1980 and 1990 and a small one from 2000 to 2010. The same is true to the other two charts. What's the reason for that? Let's dig in the question.

May be the case that there *is no* observation to this period? We check if this is the case indeed filtering those years intervals

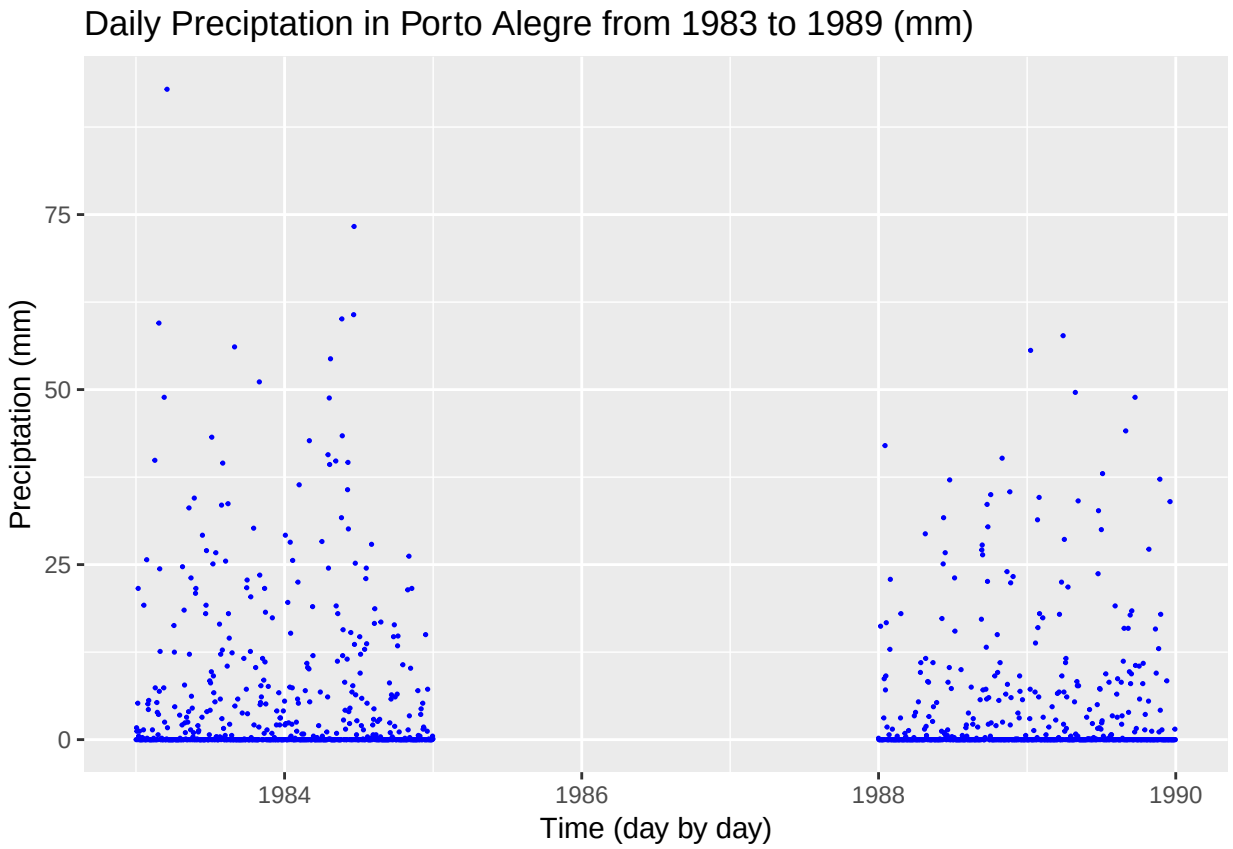
```
library(dplyr)

df %>%
  filter(Date >= as.Date('1985-01-01')) %>%
  filter(Date < as.Date('1990-01-01')) %>%
  summary()
```

```
##       Date                Hour      Preciptation    MaxTemperature
##  Min.   :1988-01-01   00:00:731   Min.   : 0.000   Min.   :11.00
##  1st Qu.:1988-07-01   12:00:731   1st Qu.: 0.000   1st Qu.:20.35
##  Median :1988-12-31               Median : 0.000   Median :24.90
##  Mean   :1988-12-31               Mean   : 3.121   Mean   :24.83
##  3rd Qu.:1989-07-01               3rd Qu.: 1.050   3rd Qu.:29.60
##  Max.   :1989-12-31               Max.   :57.700   Max.   :37.90
##                                   NA's   :731      NA's   :731
##  MinTemperature   Insolation      Evaporation    MeanTemperature
##  Min.   : 1.4   Min.   : 0.000   Min.   :0.000   Min.   : 6.64
##  1st Qu.:11.0   1st Qu.: 2.175   1st Qu.:1.500   1st Qu.:15.16
##  Median :14.8   Median : 6.900   Median :2.600   Median :19.06
##  Mean   :14.7   Mean   : 6.023   Mean   :2.711   Mean   :19.08
##  3rd Qu.:19.4   3rd Qu.: 9.225   3rd Qu.:3.700   3rd Qu.:23.51
##  Max.   :24.7   Max.   :13.200   Max.   :9.400   Max.   :30.50
```

```
##  NA's   :733    NA's   :734      NA's   :731     NA's   :731
##  MeanRelativeHumidity MeanWindVelocity
##  Min.   :38.00        Min.   :0.000
##  1st Qu.:66.50        1st Qu.:1.200
##  Median :72.75        Median :1.800
##  Mean   :73.47        Mean   :1.975
##  3rd Qu.:81.00        3rd Qu.:2.633
##  Max.   :96.75        Max.   :5.700
##  NA's   :731          NA's   :731
```

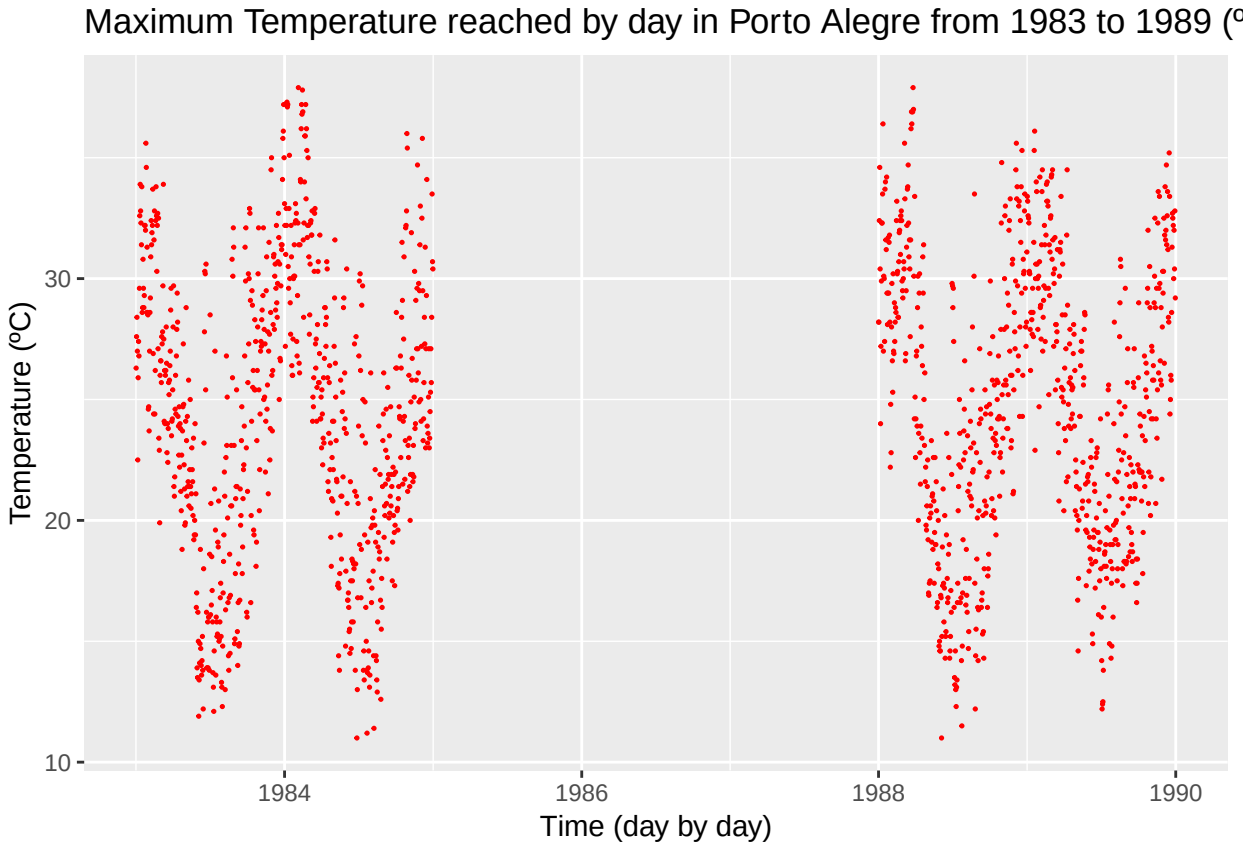
That is conclusive. Although we are filtering the dates ates 1985-01-01 and prior to 1990, the *minimum* value for **Date** variable is 1988-01-01, indicating that it is the case that there is **no** observations in this dataset for some years. This can be *visually* confirmed by the following charts:

```
df %>%
  filter(Date >= as.Date('1983-01-01')) %>%
  filter(Date < as.Date('1990-01-01')) %>%
  ggplot(aes(Date, Preciptation)) +
  geom_point(na.rm=TRUE, size=0.25, color="blue") +
  scale_x_date(limits = as.Date(c('1983-01-01','1990-01-01')) ) +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1983 to 1989 (mm)")
```

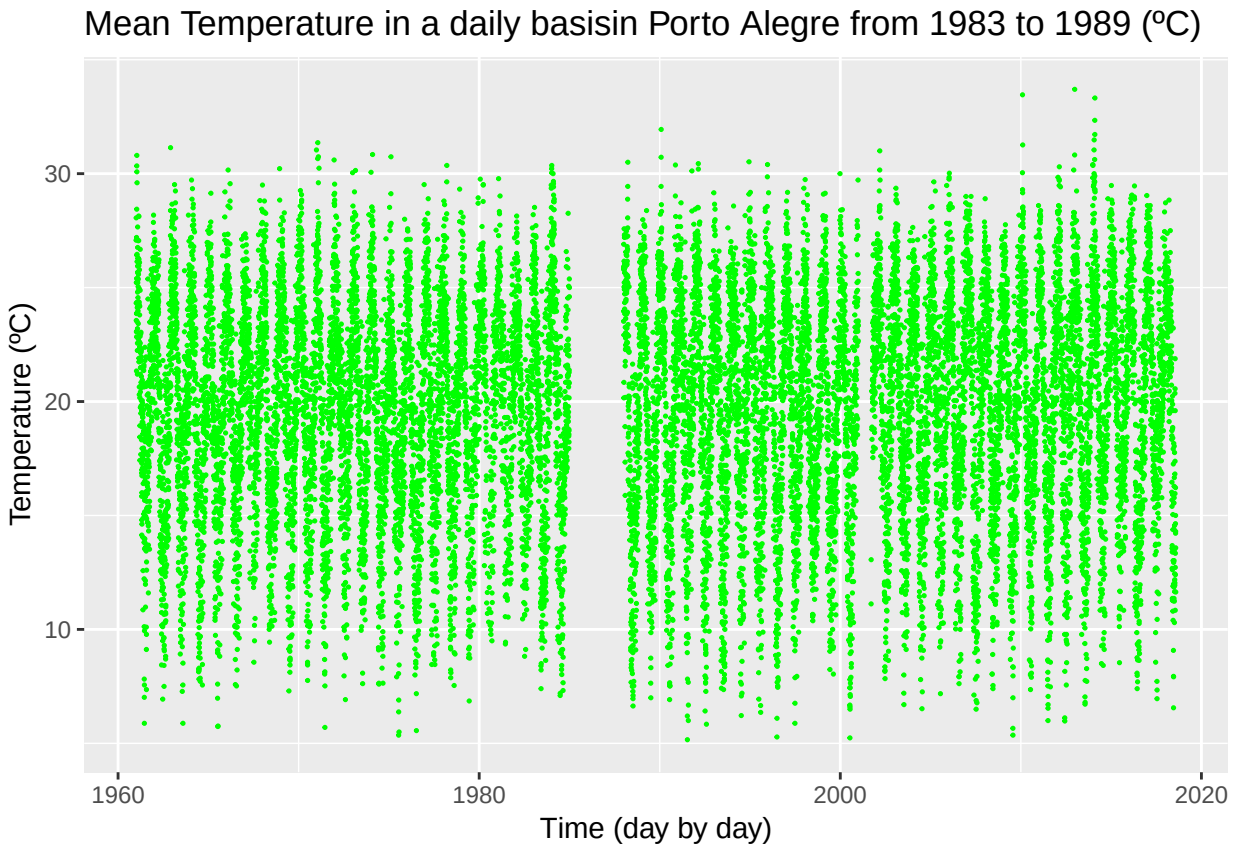



```
df %>%
  filter(Date >= as.Date('1983-01-01')) %>%
  filter(Date < as.Date('1990-01-01')) %>%
  ggplot(aes(Date, MaxTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="red") +
```

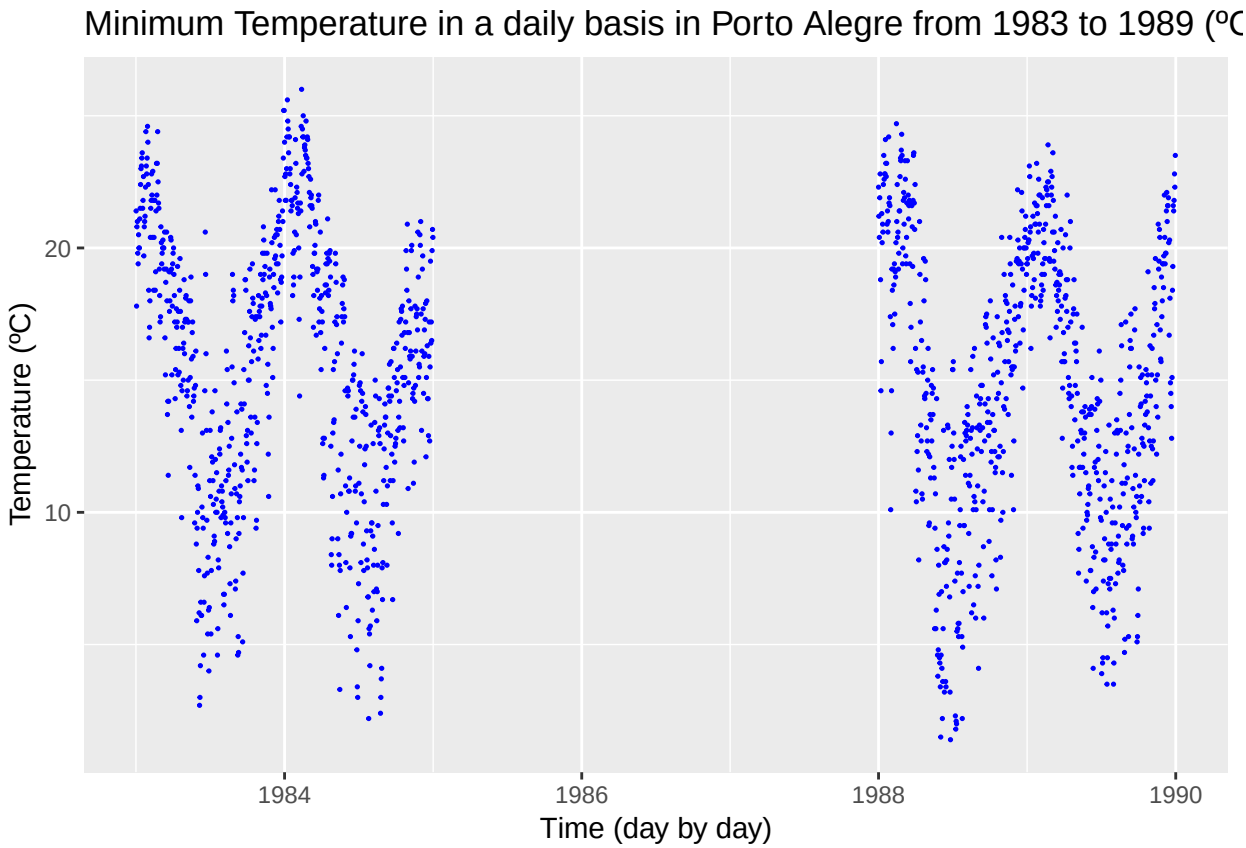
```
  scale_x_date(limits = as.Date(c('1983-01-01','1990-01-01')) ) +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Maximum Temperature reached by day in Porto Alegre from 1983 to 1989 (ºC)")
```




```
df %>%
ggplot(aes(Date, MeanTemperature)) +
geom_point(na.rm=TRUE, size=0.25, color="green") +
ylab("Temperature (ºC)") + xlab("Time (day by day)") +
ggtitle("Mean Temperature in a daily basisin Porto Alegre from 1983 to 1989 (ºC)")
```




```
df %>%
  filter(Date >= as.Date('1983-01-01')) %>%
  filter(Date < as.Date('1990-01-01')) %>%
  ggplot(aes(Date, MinTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="blue") +
  scale_x_date(limits = as.Date(c('1983-01-01','1990-01-01')) ) +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Minimum Temperature in a daily basis in Porto Alegre from 1983 to 1989 (ºC)")
```



But, since there are no observations between 1985 and 1988, I assume that there is no bias in the analysis. Also, the data seems to be comprehensive to the point that the 3 years missing will not represent a serious troble to answer the questions made in this work.

Reducing the number of rows

Lets take a look again in the headers of our dataset

```
head(df)
```

```
## # A tibble: 6 x 10
##   Date       Hour  Preciptation MaxTemperature MinTemperature Insolation
##   <date>     <fct>        <dbl>          <dbl>          <dbl>      <dbl>
## 1 1961-01-01 00:00         NA             33.8           NA         11.7
## 2 1961-01-01 12:00         NA             NA             22.2       NA
## 3 1961-01-02 00:00         NA             34.7           NA          9.5
## 4 1961-01-02 12:00          0             NA             22.5       NA
## 5 1961-01-03 00:00         NA             27.7           NA          2.3
## 6 1961-01-03 12:00          0.2           NA             23.1       NA
## # ... with 4 more variables: Evaporation <dbl>, MeanTemperature <dbl>,
## #   MeanRelativeHumidity <dbl>, MeanWindVelocity <dbl>
```

It would be nice to deal only with **Date** and don't bother with **Hour**. Remember, at 12:00 **Preciptation** and **MinTemperature** are collected and all other variables are collected at 00:00:

```
df %>%
  filter(Hour == '12:00') %>%
```

```
  head()
```

```
## # A tibble: 6 x 10
##   Date       Hour  Preciptation MaxTemperature MinTemperature Insolation
##   <date>     <fct>        <dbl>          <dbl>          <dbl>      <dbl>
## 1 1961-01-01 12:00         NA               NA           22.2         NA
## 2 1961-01-02 12:00          0               NA           22.5         NA
## 3 1961-01-03 12:00          0.2             NA           23.1         NA
## 4 1961-01-04 12:00          2.4             NA           20.6         NA
## 5 1961-01-05 12:00          0               NA           18.8         NA
## 6 1961-01-06 12:00          3.8             NA           21.9         NA
## # ... with 4 more variables: Evaporation <dbl>, MeanTemperature <dbl>,
## #   MeanRelativeHumidity <dbl>, MeanWindVelocity <dbl>
```

```
df %>%
  filter(Hour == '00:00') %>%
  head()
```

```
## # A tibble: 6 x 10
##   Date       Hour  Preciptation MaxTemperature MinTemperature Insolation
##   <date>     <fct>        <dbl>          <dbl>          <dbl>      <dbl>
## 1 1961-01-01 00:00           NA           33.8             NA       11.7
## 2 1961-01-02 00:00           NA           34.7             NA        9.5
## 3 1961-01-03 00:00           NA           27.7             NA        2.3
## 4 1961-01-04 00:00           NA           29.4             NA        8.7
## 5 1961-01-05 00:00           NA           32.5             NA       10.1
## 6 1961-01-06 00:00           NA           26.2             NA        0.4
## # ... with 4 more variables: Evaporation <dbl>, MeanTemperature <dbl>,
## #   MeanRelativeHumidity <dbl>, MeanWindVelocity <dbl>
```

At least, this **should** be the case, at least.

```
df %>%
  filter(Hour == '12:00') %>%
  summary()
```

```
##       Date                Hour        Preciptation     MaxTemperature
##  Min.   :1961-01-01   00:00:    0   Min.   :  0.000   Min.   :11.20
##  1st Qu.:1974-04-22   12:00:19314   1st Qu.:  0.000   1st Qu.:22.40
##  Median :1991-06-17                 Median :  0.000   Median :26.85
##  Mean   :1990-01-07                 Mean   :  3.774   Mean   :26.37
##  3rd Qu.:2005-05-11                 3rd Qu.:  1.700   3rd Qu.:31.18
##  Max.   :2018-07-31                 Max.   :149.600   Max.   :37.80
##                                     NA's   :274       NA's   :19060
##  MinTemperature    Insolation      Evaporation    MeanTemperature
##  Min.   :-0.20   Min.   : 0.000   Min.   :0.200   Min.   : 8.64
##  1st Qu.:12.50   1st Qu.: 1.700   1st Qu.:1.225   1st Qu.:17.88
##  Median :16.20   Median : 7.200   Median :2.050   Median :22.18
##  Mean   :15.68   Mean   : 6.014   Mean   :2.052   Mean   :21.10
##  3rd Qu.:19.40   3rd Qu.: 9.800   3rd Qu.:2.650   3rd Qu.:24.84
##  Max.   :27.90   Max.   :11.800   Max.   :4.900   Max.   :29.36
##  NA's   :284     NA's   :19071    NA's   :19254   NA's   :19065
##  MeanRelativeHumidity MeanWindVelocity
##  Min.   :45.25        Min.   :  0.000
##  1st Qu.:69.38        1st Qu.:  1.200
##  Median :76.25        Median :  1.600
```

```
##  Mean   :76.82        Mean   : 118.093
##  3rd Qu.:84.00        3rd Qu.:   2.401
##  Max.   :96.25        Max.   :4287.000
##  NA's   :19067        NA's   :19048
```

```
df %>%
  filter(Hour == '00:00') %>%
  summary()
```

```
##       Date                Hour        Preciptation    MaxTemperature
##  Min.   :1961-01-01   00:00:19315   Min.   : 0.000   Min.   : 7.40
##  1st Qu.:1974-04-21   12:00:    0   1st Qu.: 0.000   1st Qu.:21.20
##  Median :1991-06-17                 Median : 0.000   Median :25.50
##  Mean   :1990-01-07                 Mean   : 4.908   Mean   :25.19
##  3rd Qu.:2005-05-11                 3rd Qu.: 3.625   3rd Qu.:29.40
##  Max.   :2018-07-31                 Max.   :81.300   Max.   :40.60
##                                     NA's   :19051    NA's   :284
##  MinTemperature    Insolation      Evaporation     MeanTemperature
##  Min.   : 4.30   Min.   : 0.000   Min.   : 0.000   Min.   : 5.16
##  1st Qu.:14.20   1st Qu.: 2.400   1st Qu.: 1.300   1st Qu.:16.28
##  Median :18.20   Median : 7.000   Median : 2.200   Median :20.06
##  Mean   :17.24   Mean   : 6.072   Mean   : 2.482   Mean   :19.65
##  3rd Qu.:21.00   3rd Qu.: 9.200   3rd Qu.: 3.300   3rd Qu.:23.38
##  Max.   :25.60   Max.   :13.200   Max.   :20.700   Max.   :33.70
##  NA's   :19061   NA's   :385      NA's   :764      NA's   :295
##  MeanRelativeHumidity MeanWindVelocity
##  Min.   :37.50        Min.   :   0.000
##  1st Qu.:69.25        1st Qu.:   1.000
##  Median :76.50        Median :   1.800
##  Mean   :76.44        Mean   :   7.715
##  3rd Qu.:84.00        3rd Qu.:   2.767
##  Max.   :99.75        Max.   :6216.000
##  NA's   :297          NA's   :268
```

As we can see from above, **all** variables have some values registered in '12:00' and in '00:00' observations. As I stated above, it would be nice to deal only with **Date** and don't bother with **Hour**. So we merge the observations. There are only 12 hours of difference from 00:00 observations and 12:00 observations for the same say, I really don't expect that regarding all observations relating a specific day would bias the analysis significantly.

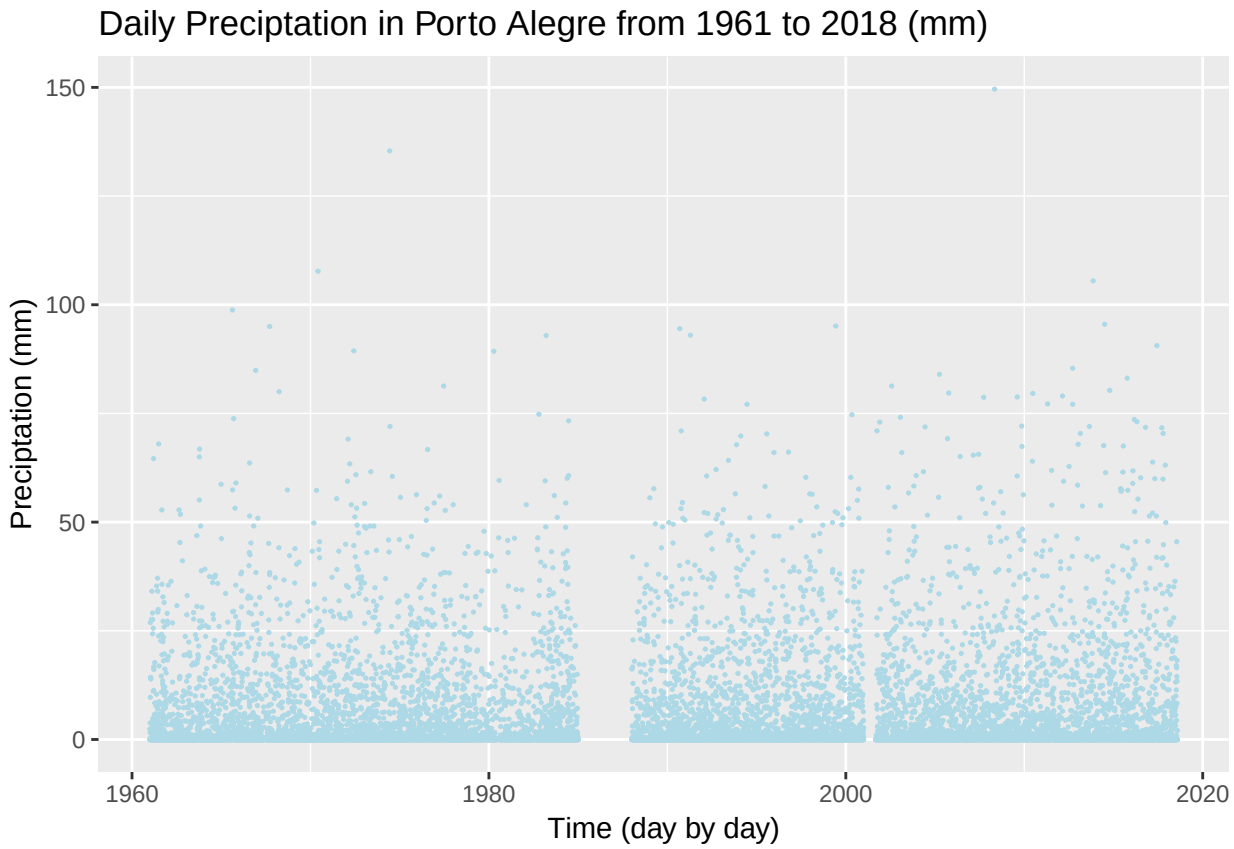
Let's start the process of merging the rows by **Date**!

For the first step, we filter the 12:00 observations (**Preciptation** and **MinTemperature**) from the 00:observations.

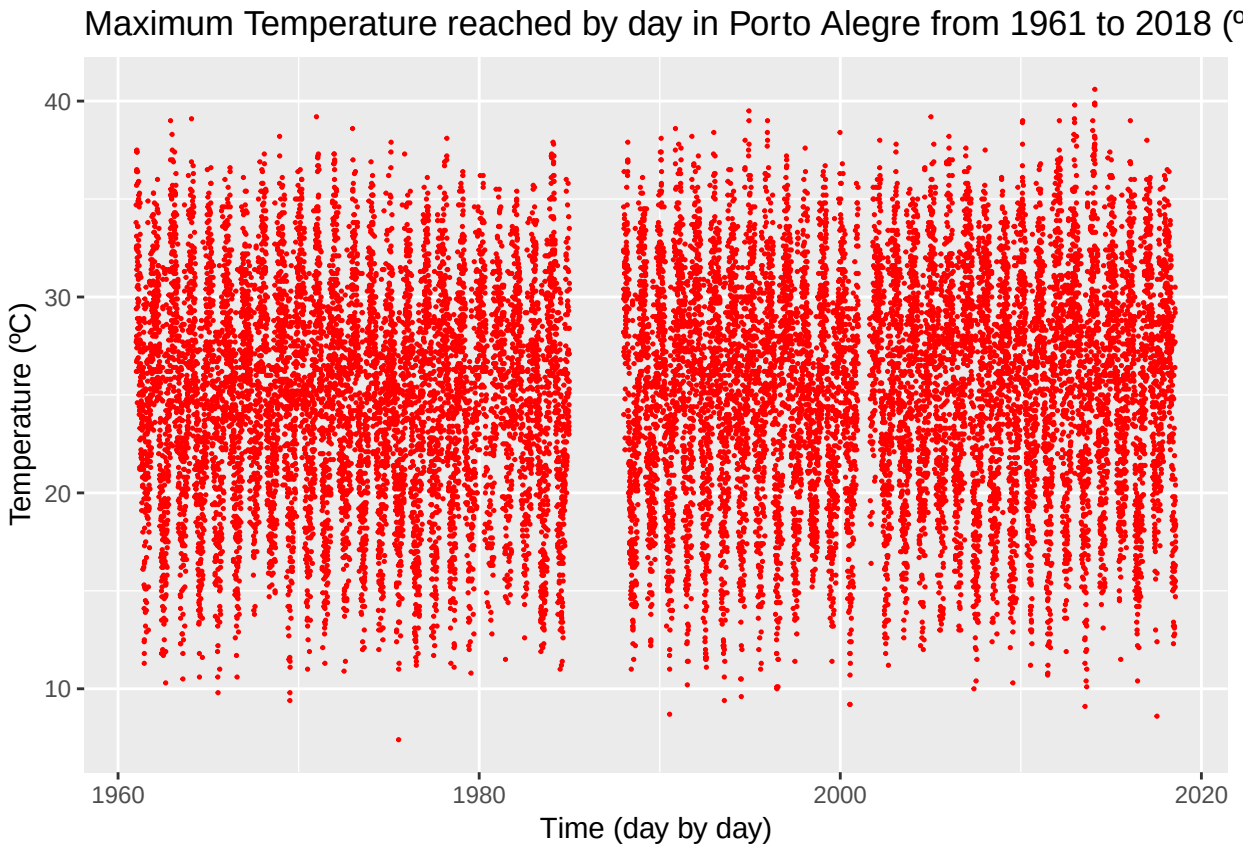
```
df12 <- (df %>%
           filter(Hour == '12:00')
         )

df00 <- (df %>%
           filter(Hour == '00:00')
         )
```

Here we have something interesting! Remember the second small gap in data between 2000 and 2005 ?? No need to go back, here it is:

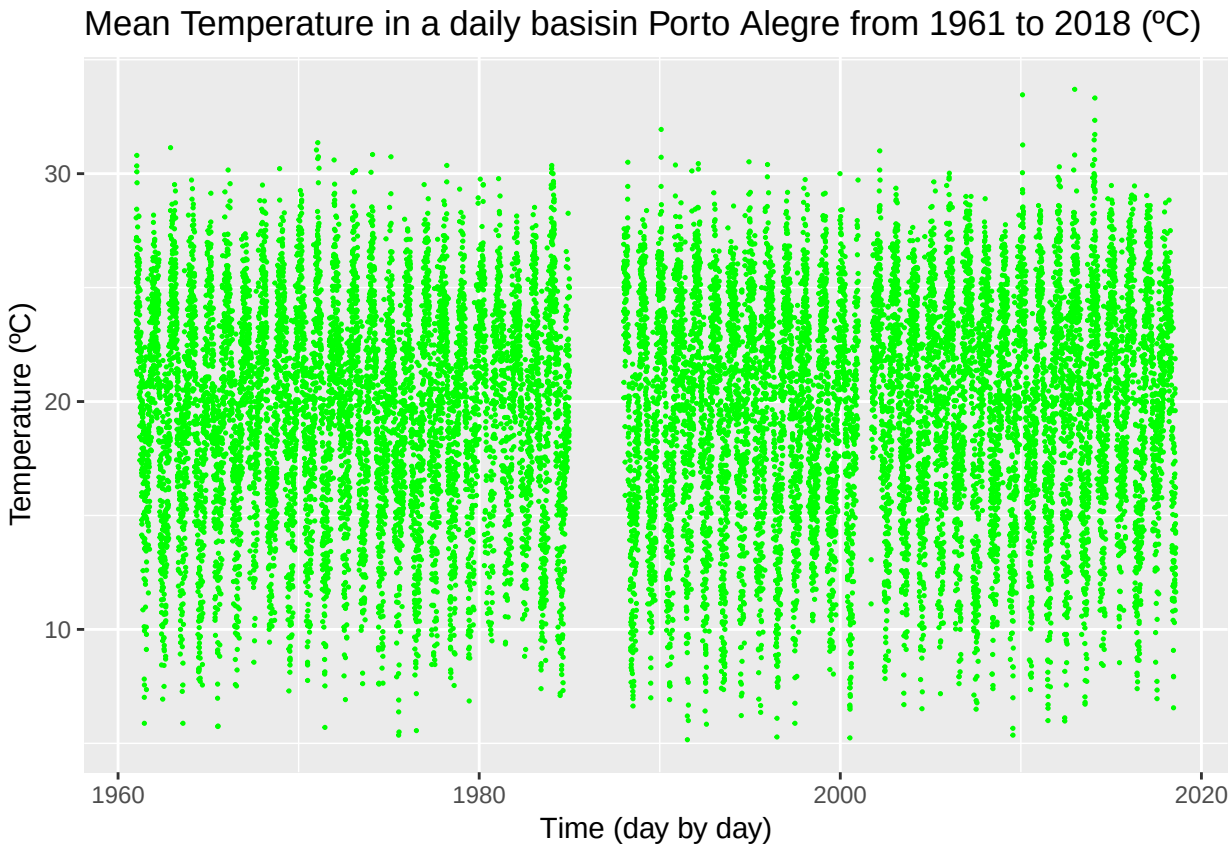
```
df %>%
  ggplot(aes(Date, Preciptation)) +
  geom_point(na.rm=TRUE, size=0.25, color="lightblue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm)")
```



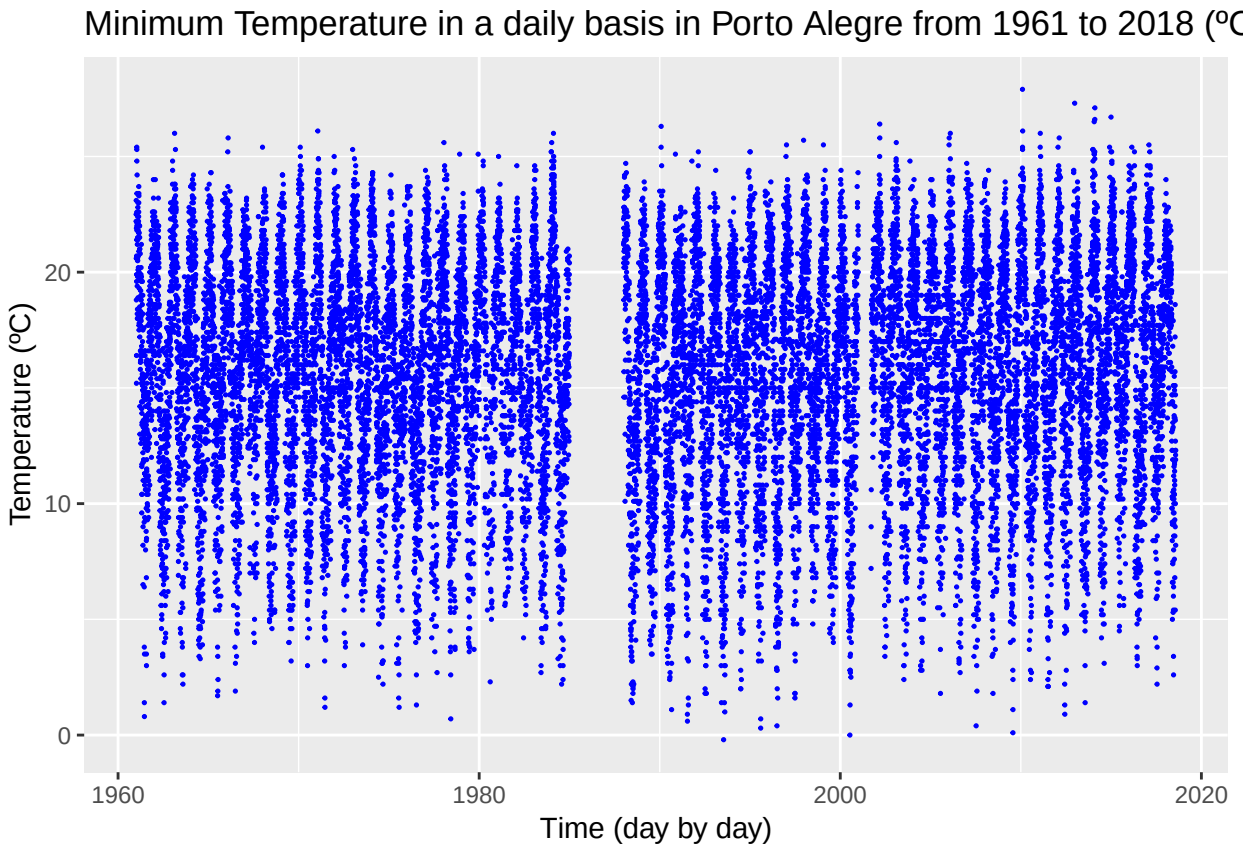

```
df %>%
  ggplot(aes(Date, MaxTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="red") +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Maximum Temperature reached by day in Porto Alegre from 1961 to 2018 (ºC)")
```




```
df %>%
  ggplot(aes(Date, MeanTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="green") +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Mean Temperature in a daily basisin Porto Alegre from 1961 to 2018 (ºC)")
```

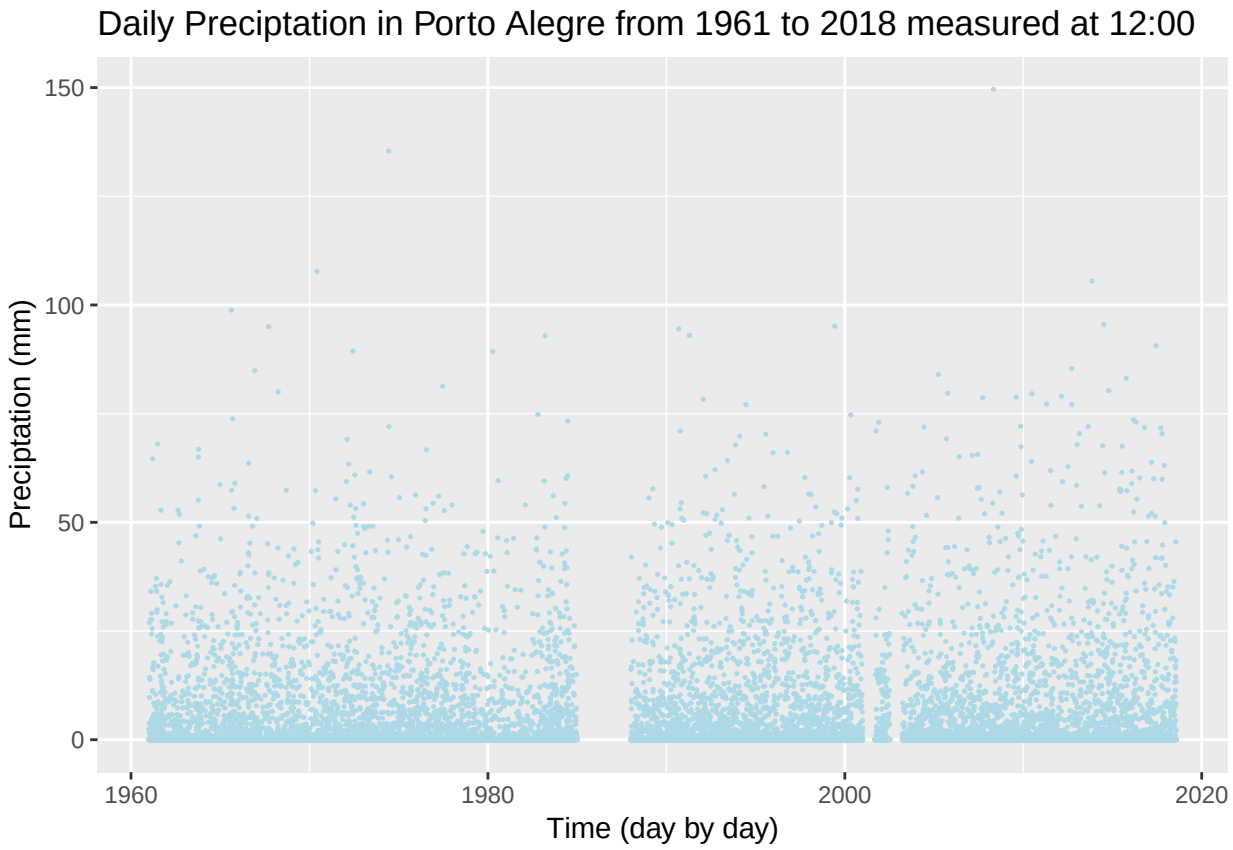



```
df %>%
  ggplot(aes(Date, MinTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="blue") +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Minimum Temperature in a daily basis in Porto Alegre from 1961 to 2018 (ºC)")
```

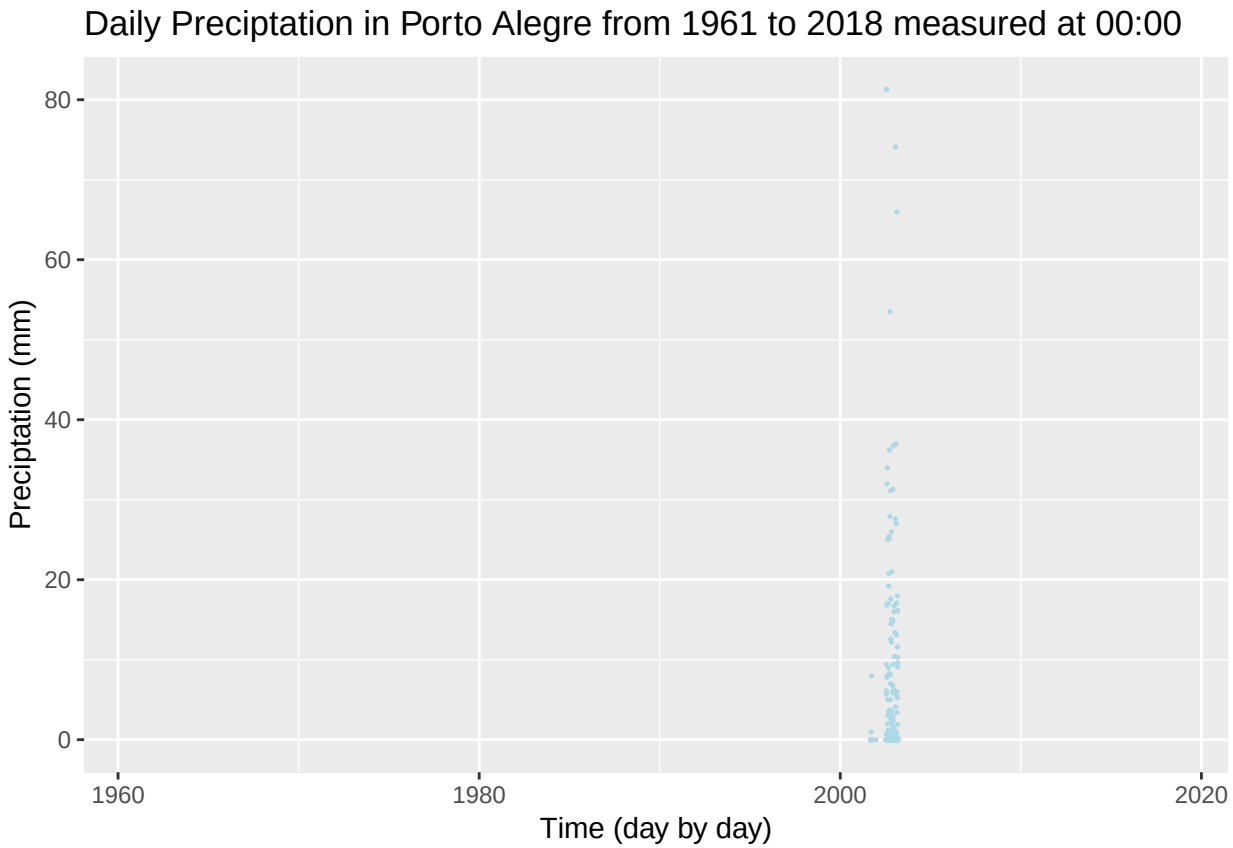


We can have a good picture of what is going on with this second gap with:

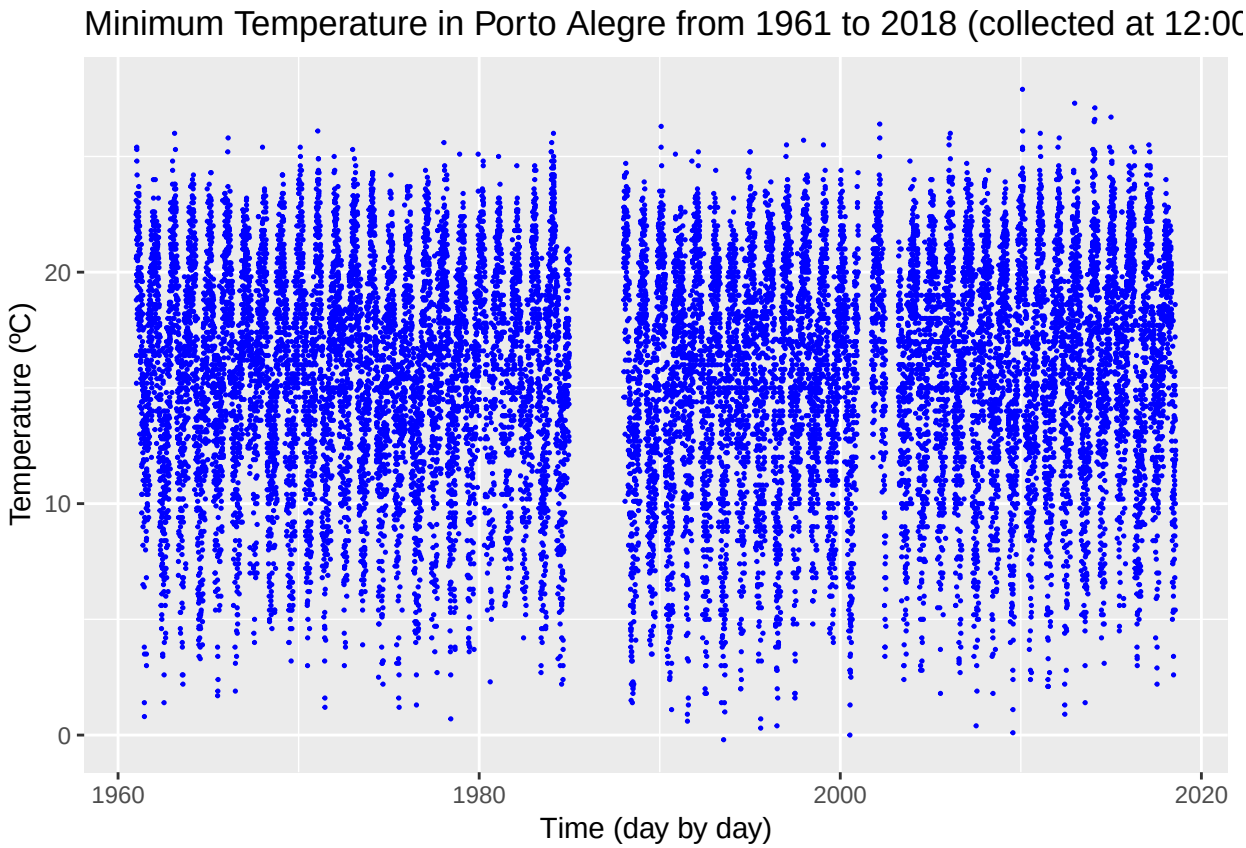
```
df12 %>%
  ggplot(aes(Date, Preciptation)) +
  geom_point(na.rm=TRUE, size=0.25, color='lightblue') +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 measured at 12:00")
```



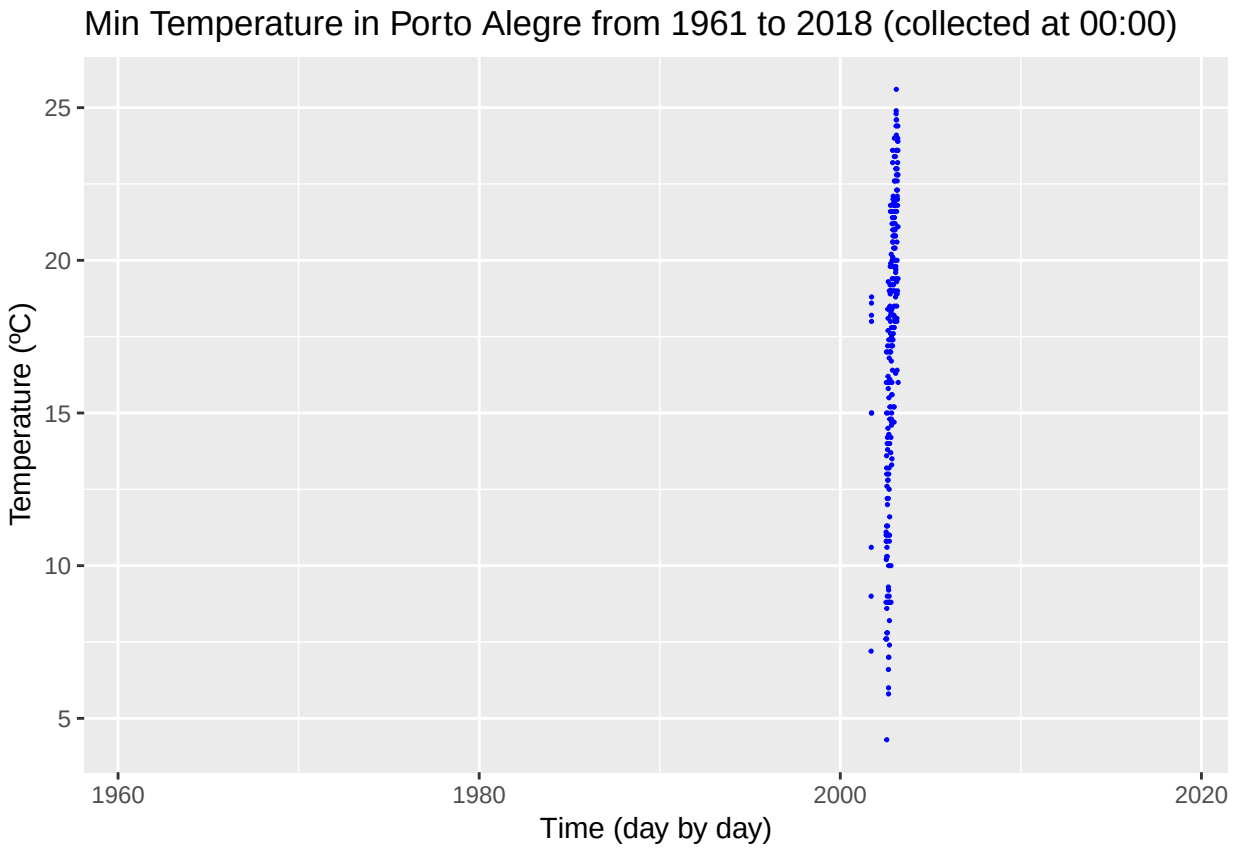

```
df00 %>%
  ggplot(aes(Date, Preciptation)) +
  geom_point(na.rm=TRUE, size=0.25, color='lightblue') +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 measured at 00:00")
```



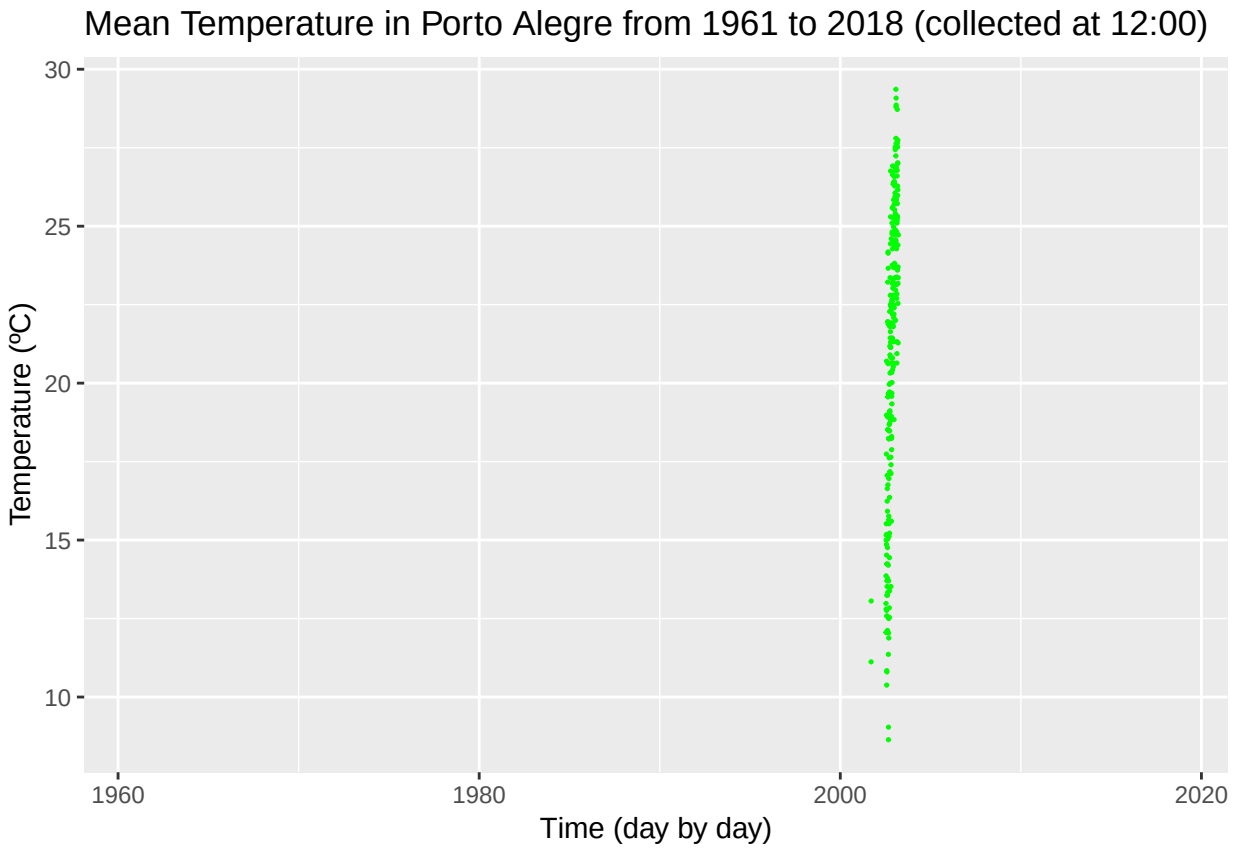

```
df12 %>%
  ggplot(aes(Date, MinTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color='blue') +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Minimum Temperature in Porto Alegre from 1961 to 2018 (collected at 12:00)")
```



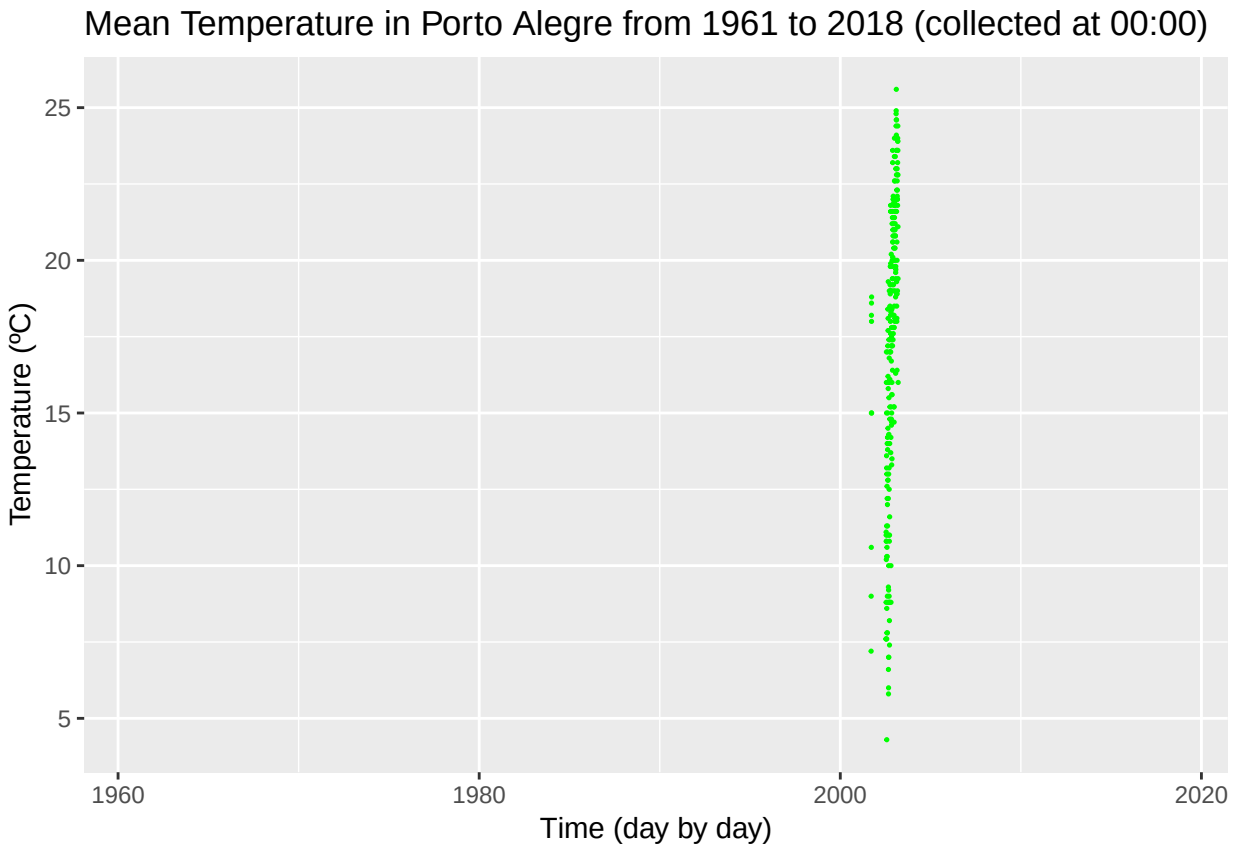

```
df00 %>%
  ggplot(aes(Date, MinTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color='blue') +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Min Temperature in Porto Alegre from 1961 to 2018 (collected at 00:00)")
```



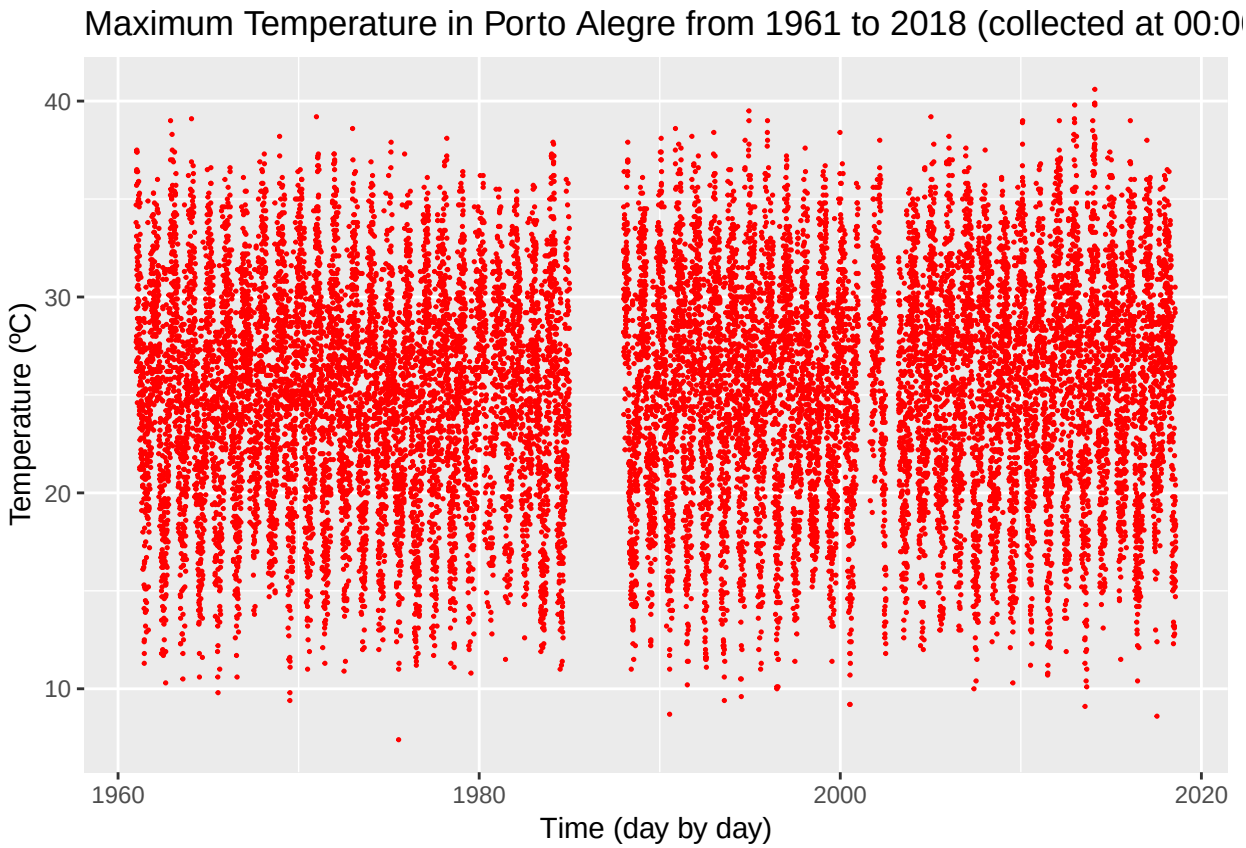

```
df12 %>%
  ggplot(aes(Date, MeanTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color='green') +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Mean Temperature in Porto Alegre from 1961 to 2018 (collected at 12:00)")
```



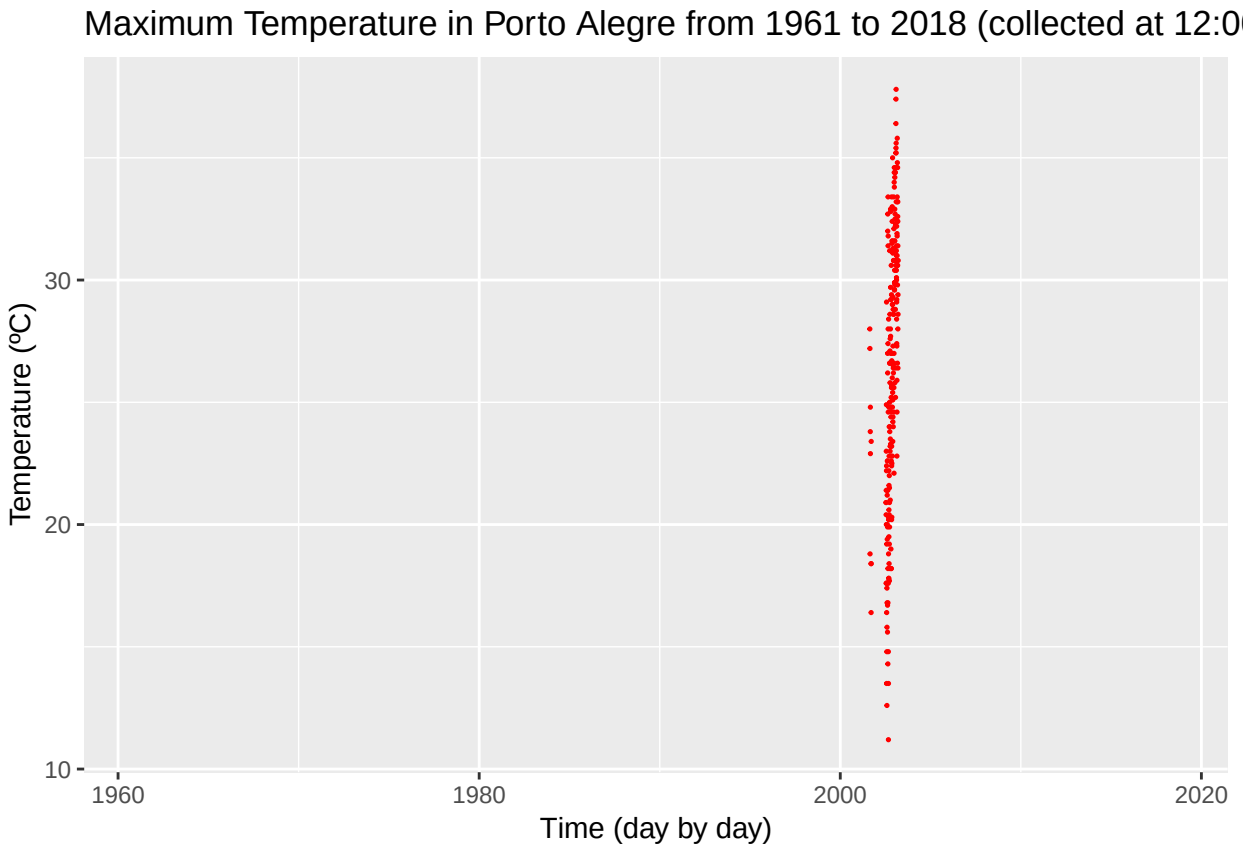

```
df00 %>%
  ggplot(aes(Date, MinTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color='green') +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Mean Temperature in Porto Alegre from 1961 to 2018 (collected at 00:00)")
```




```
df00 %>%
  ggplot(aes(Date, MaxTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color='red') +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Maximum Temperature in Porto Alegre from 1961 to 2018 (collected at 00:00)")
```




```
df12 %>%
  ggplot(aes(Date, MaxTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color='red') +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Maximum Temperature in Porto Alegre from 1961 to 2018 (collected at 12:00)")
```



It seems that someone made a mistake and collected the maximum temperature at 12:00 instead of 00:00, the preciptation at 00:00 instead of 12:00. This will no longer be a problem if we consolidate the observations made in a day for different hours in a single row. Let's do it.

First, we remove from df12 and df00 the **Hour** column as it is no longer important and will not be used in the merge process.

```
df12$Hour <- NULL
df00$Hour <- NULL
```

We merge df12 and df00 in a new dataset called **dfmerged**.

```
dfmerged <- merge(df00, df12, by="Date", all=FALSE)
```

Let's take a look in dfmerged.

```
summary(dfmerged)
```

```
##       Date            Preciptation.x   MaxTemperature.x MinTemperature.x
##  Min.   :1961-01-01   Min.   : 0.000   Min.   : 7.40    Min.   : 4.30
##  1st Qu.:1974-04-18   1st Qu.: 0.000   1st Qu.:21.20    1st Qu.:14.20
##  Median :1991-06-10   Median : 0.000   Median :25.50    Median :18.20
##  Mean   :1990-01-04   Mean   : 5.041   Mean   :25.19    Mean   :17.24
##  3rd Qu.:2005-05-15   3rd Qu.: 4.100   3rd Qu.:29.40    3rd Qu.:21.00
##  Max.   :2018-07-31   Max.   :81.300   Max.   :40.60    Max.   :25.60
##                       NA's   :19042    NA's   :275      NA's   :19045
##   Insolation.x    Evaporation.x    MeanTemperature.x
##  Min.   : 0.000   Min.   : 0.000   Min.   : 5.16
##  1st Qu.: 2.400   1st Qu.: 1.300   1st Qu.:16.28
##  Median : 7.000   Median : 2.200   Median :20.06
```

```
##  Mean   : 6.072   Mean   : 2.482   Mean   :19.65
##  3rd Qu.: 9.200   3rd Qu.: 3.300   3rd Qu.:23.38
##  Max.   :13.200   Max.   :20.700   Max.   :33.70
##  NA's   :371      NA's   :748      NA's   :282
##  MeanRelativeHumidity.x MeanWindVelocity.x Preciptation.y
##  Min.   :37.50          Min.   :   0.000   Min.   :  0.000
##  1st Qu.:69.25          1st Qu.:   1.000   1st Qu.:  0.000
##  Median :76.50          Median :   1.800   Median :  0.000
##  Mean   :76.44          Mean   :   7.200   Mean   :  3.775
##  3rd Qu.:84.00          3rd Qu.:   2.767   3rd Qu.:  1.700
##  Max.   :99.75          Max.   :6216.000   Max.   :149.600
##  NA's   :284            NA's   :260        NA's   :267
##  MaxTemperature.y MinTemperature.y  Insolation.y    Evaporation.y
##  Min.   :11.20    Min.   :-0.20    Min.   : 0.000   Min.   :0.200
##  1st Qu.:22.35    1st Qu.:12.50    1st Qu.: 1.600   1st Qu.:1.225
##  Median :26.65    Median :16.20    Median : 7.200   Median :2.050
##  Mean   :26.37    Mean   :15.68    Mean   : 6.006   Mean   :2.052
##  3rd Qu.:31.20    3rd Qu.:19.40    3rd Qu.: 9.800   3rd Qu.:2.650
##  Max.   :37.80    Max.   :27.90    Max.   :11.800   Max.   :4.900
##  NA's   :19051    NA's   :269      NA's   :19058    NA's   :19239
##  MeanTemperature.y MeanRelativeHumidity.y MeanWindVelocity.y
##  Min.   : 8.64     Min.   :45.25          Min.   :   0.000
##  1st Qu.:17.77     1st Qu.:69.44          1st Qu.:   1.200
##  Median :22.04     Median :76.25          Median :   1.543
##  Mean   :21.06     Mean   :76.83          Mean   : 101.248
##  3rd Qu.:24.87     3rd Qu.:84.06          3rd Qu.:   2.350
##  Max.   :29.36     Max.   :96.25          Max.   :4287.000
##  NA's   :19053     NA's   :19055          NA's   :19040
```

```
nrow(dfmerged)
```

```
## [1] 19299
```

As we can see, there is no great improve in this approach, *unless* we are able to create definitive columns for each original variable and get rid of this .x and .y attributes.

We create those *definitive* variables (the original ones) with the command

```
dfmerged <- (dfmerged %>%
  mutate(MeanRelativeHumidity = coalesce(MeanRelativeHumidity.x, MeanRelativeHumidity.y)) %>%
  mutate(MeanTemperature = coalesce(MeanTemperature.x, MeanTemperature.y)) %>%
  mutate(MeanWindVelocity = coalesce(MeanWindVelocity.x, MeanWindVelocity.y)) %>%
  mutate(MaxTemperature = coalesce(MaxTemperature.x, MaxTemperature.y)) %>%
  mutate(Evaporation = coalesce(Evaporation.x, Evaporation.y)) %>%
  mutate(Insolation = coalesce(Insolation.x, Insolation.y)) %>%
  mutate(Preciptation = coalesce(Preciptation.y, Preciptation.x)) %>%
  mutate(MinTemperature = coalesce(MinTemperature.y, MinTemperature.x))
)

summary(dfmerged)
```

```
##       Date            Preciptation.x   MaxTemperature.x MinTemperature.x
##  Min.   :1961-01-01   Min.   : 0.000   Min.   : 7.40    Min.   : 4.30
##  1st Qu.:1974-04-18   1st Qu.: 0.000   1st Qu.:21.20    1st Qu.:14.20
##  Median :1991-06-10   Median : 0.000   Median :25.50    Median :18.20
##  Mean   :1990-01-04   Mean   : 5.041   Mean   :25.19    Mean   :17.24
```

```
##  3rd Qu.:2005-05-15   3rd Qu.: 4.100   3rd Qu.:29.40    3rd Qu.:21.00
##  Max.   :2018-07-31   Max.   :81.300   Max.   :40.60    Max.   :25.60
##                       NA's   :19042    NA's   :275      NA's   :19045
##   Insolation.x    Evaporation.x    MeanTemperature.x
##  Min.   : 0.000   Min.   : 0.000   Min.   : 5.16
##  1st Qu.: 2.400   1st Qu.: 1.300   1st Qu.:16.28
##  Median : 7.000   Median : 2.200   Median :20.06
##  Mean   : 6.072   Mean   : 2.482   Mean   :19.65
##  3rd Qu.: 9.200   3rd Qu.: 3.300   3rd Qu.:23.38
##  Max.   :13.200   Max.   :20.700   Max.   :33.70
##  NA's   :371      NA's   :748      NA's   :282
##  MeanRelativeHumidity.x MeanWindVelocity.x Preciptation.y
##  Min.   :37.50          Min.   :   0.000   Min.   :  0.000
##  1st Qu.:69.25          1st Qu.:   1.000   1st Qu.:  0.000
##  Median :76.50          Median :   1.800   Median :  0.000
##  Mean   :76.44          Mean   :   7.200   Mean   :  3.775
##  3rd Qu.:84.00          3rd Qu.:   2.767   3rd Qu.:  1.700
##  Max.   :99.75          Max.   :6216.000   Max.   :149.600
##  NA's   :284            NA's   :260        NA's   :267
##  MaxTemperature.y MinTemperature.y  Insolation.y    Evaporation.y
##  Min.   :11.20    Min.   :-0.20    Min.   : 0.000   Min.   :0.200
##  1st Qu.:22.35    1st Qu.:12.50    1st Qu.: 1.600   1st Qu.:1.225
##  Median :26.65    Median :16.20    Median : 7.200   Median :2.050
##  Mean   :26.37    Mean   :15.68    Mean   : 6.006   Mean   :2.052
##  3rd Qu.:31.20    3rd Qu.:19.40    3rd Qu.: 9.800   3rd Qu.:2.650
##  Max.   :37.80    Max.   :27.90    Max.   :11.800   Max.   :4.900
##  NA's   :19051    NA's   :269      NA's   :19058    NA's   :19239
##  MeanTemperature.y MeanRelativeHumidity.y MeanWindVelocity.y
##  Min.   : 8.64     Min.   :45.25          Min.   :   0.000
##  1st Qu.:17.77     1st Qu.:69.44          1st Qu.:   1.200
##  Median :22.04     Median :76.25          Median :   1.543
##  Mean   :21.06     Mean   :76.83          Mean   : 101.248
##  3rd Qu.:24.87     3rd Qu.:84.06          3rd Qu.:   2.350
##  Max.   :29.36     Max.   :96.25          Max.   :4287.000
##  NA's   :19053     NA's   :19055          NA's   :19040
##  MeanRelativeHumidity MeanTemperature MeanWindVelocity   MaxTemperature
##  Min.   :37.50        Min.   : 5.16   Min.   :   0.000   Min.   : 7.40
##  1st Qu.:69.25        1st Qu.:16.30   1st Qu.:   1.000   1st Qu.:21.20
##  Median :76.50        Median :20.08   Median :   1.800   Median :25.50
##  Mean   :76.45        Mean   :19.66   Mean   :   8.463   Mean   :25.21
##  3rd Qu.:84.00        3rd Qu.:23.40   3rd Qu.:   2.767   3rd Qu.:29.40
##  Max.   :99.75        Max.   :33.70   Max.   :6216.000   Max.   :40.60
##  NA's   :40           NA's   :36      NA's   :1          NA's   :27
##   Evaporation       Insolation      Preciptation     MinTemperature
##  Min.   : 0.000   Min.   : 0.000   Min.   :  0.000   Min.   :-0.2
##  1st Qu.: 1.300   1st Qu.: 2.400   1st Qu.:  0.000   1st Qu.:12.5
##  Median : 2.200   Median : 7.000   Median :  0.000   Median :16.2
##  Mean   : 2.481   Mean   : 6.071   Mean   :  3.792   Mean   :15.7
##  3rd Qu.: 3.300   3rd Qu.: 9.200   3rd Qu.:  1.700   3rd Qu.:19.4
##  Max.   :20.700   Max.   :13.200   Max.   :149.600   Max.   :27.9
##  NA's   :688      NA's   :130      NA's   :10        NA's   :15
```

And get rid of the .x and .y variables with

```
dfmerged$Preciptation.x <- NULL
dfmerged$MaxTemperature.x <- NULL
dfmerged$MinTemperature.x <- NULL
dfmerged$Insolation.x <- NULL
dfmerged$Evaporation.x <- NULL
dfmerged$MeanTemperature.x <- NULL
dfmerged$MeanRelativeHumidity.x <- NULL
dfmerged$MeanWindVelocity.x <- NULL
dfmerged$Season.x <- NULL
dfmerged$Month.x <- NULL
dfmerged$Decade.x <- NULL
dfmerged$Preciptation.y <- NULL
dfmerged$MaxTemperature.y <- NULL
dfmerged$MinTemperature.y <- NULL
dfmerged$Insolation.y <- NULL
dfmerged$Evaporation.y <- NULL
dfmerged$MeanTemperature.y <- NULL
dfmerged$MeanRelativeHumidity.y <- NULL
dfmerged$MeanWindVelocity.y <- NULL
dfmerged$Season.y <- NULL
dfmerged$Month.y <- NULL
dfmerged$Decade.y <- NULL
```

Thus, we finally get *dfmerged* that in summary is

```
summary(dfmerged)
```

```
##       Date            MeanRelativeHumidity MeanTemperature
##  Min.   :1961-01-01   Min.   :37.50        Min.   : 5.16
##  1st Qu.:1974-04-18   1st Qu.:69.25        1st Qu.:16.30
##  Median :1991-06-10   Median :76.50        Median :20.08
##  Mean   :1990-01-04   Mean   :76.45        Mean   :19.66
##  3rd Qu.:2005-05-15   3rd Qu.:84.00        3rd Qu.:23.40
##  Max.   :2018-07-31   Max.   :99.75        Max.   :33.70
##                       NA's   :40           NA's   :36
##  MeanWindVelocity   MaxTemperature   Evaporation       Insolation
##  Min.   :   0.000   Min.   : 7.40   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:   1.000   1st Qu.:21.20   1st Qu.: 1.300   1st Qu.: 2.400
##  Median :   1.800   Median :25.50   Median : 2.200   Median : 7.000
##  Mean   :   8.463   Mean   :25.21   Mean   : 2.481   Mean   : 6.071
##  3rd Qu.:   2.767   3rd Qu.:29.40   3rd Qu.: 3.300   3rd Qu.: 9.200
##  Max.   :6216.000   Max.   :40.60   Max.   :20.700   Max.   :13.200
##  NA's   :1          NA's   :27      NA's   :688      NA's   :130
##   Preciptation     MinTemperature
##  Min.   :  0.000   Min.   :-0.2
##  1st Qu.:  0.000   1st Qu.:12.5
##  Median :  0.000   Median :16.2
##  Mean   :  3.792   Mean   :15.7
##  3rd Qu.:  1.700   3rd Qu.:19.4
##  Max.   :149.600   Max.   :27.9
##  NA's   :10        NA's   :15
```

and gives us a good idea of the number of missing values for **Preciptation, MaxTemperature, MeanTemperature** and **MinTemperature**. Because the is

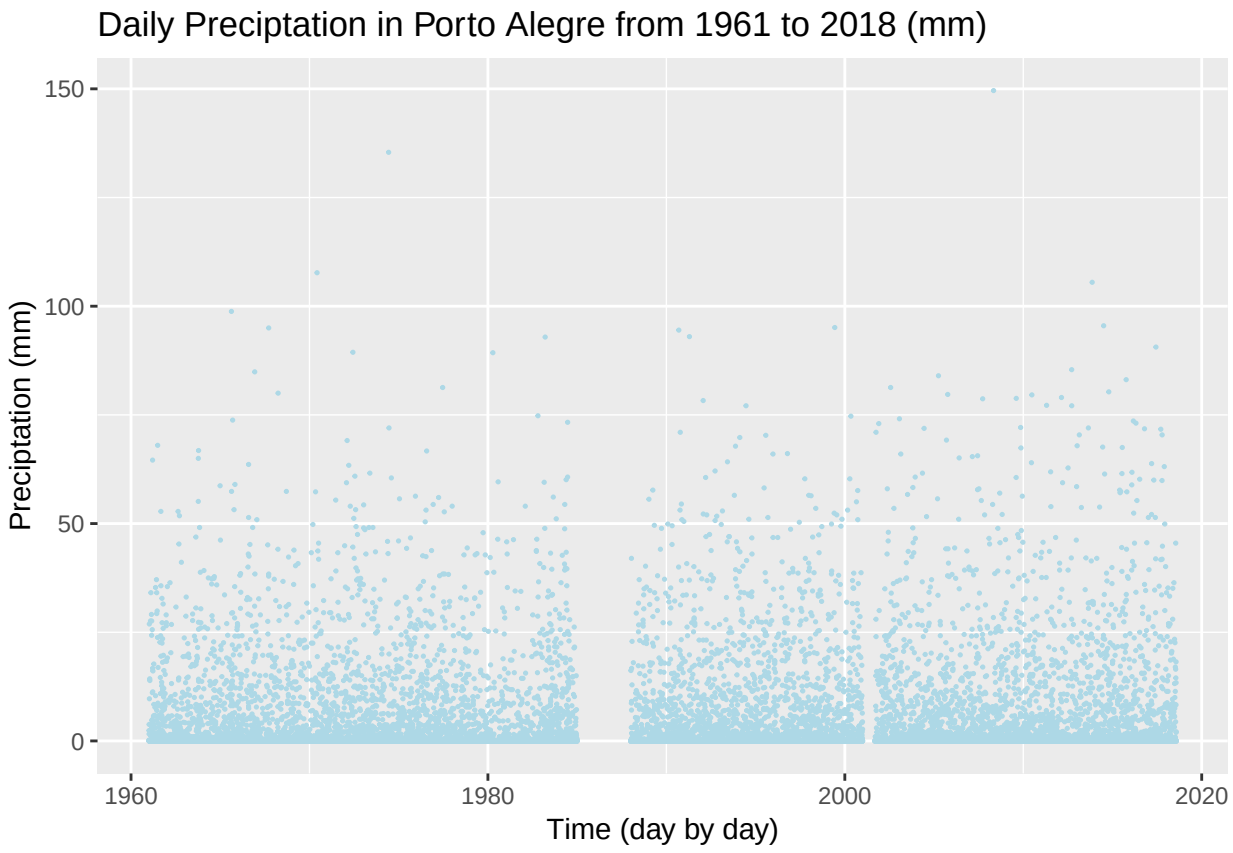
```
nrow(dfmerged)
```

```
## [1] 19299
```

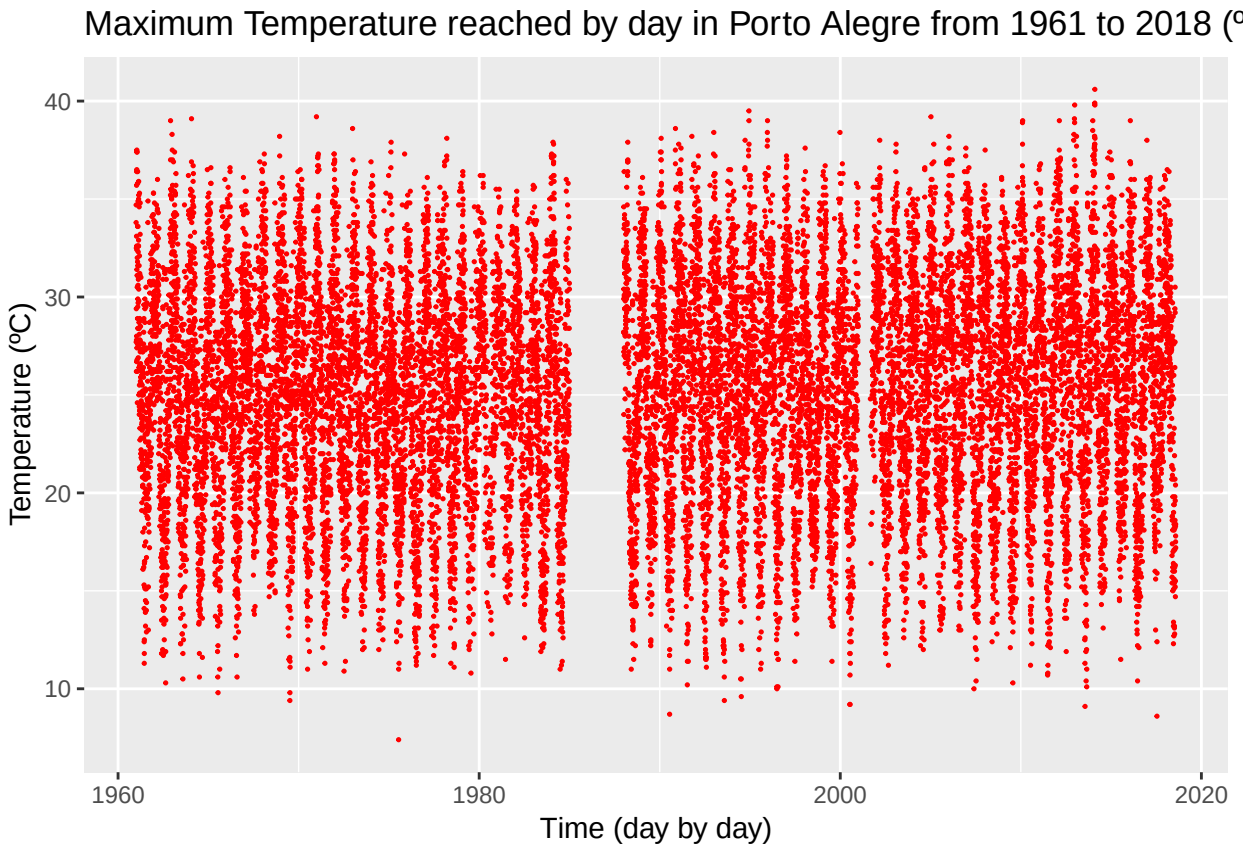
observations, we simple ignore the missing values that are, for those variables we are interested, less than 36, then less than 0,2% (36/19299) of missing values for the variables **MaxTemperature, MeanTemperature, MinTemperature** and **Preciptation**.

Let's see if *dfmerged* shows the little gap between 2000 and 2005.

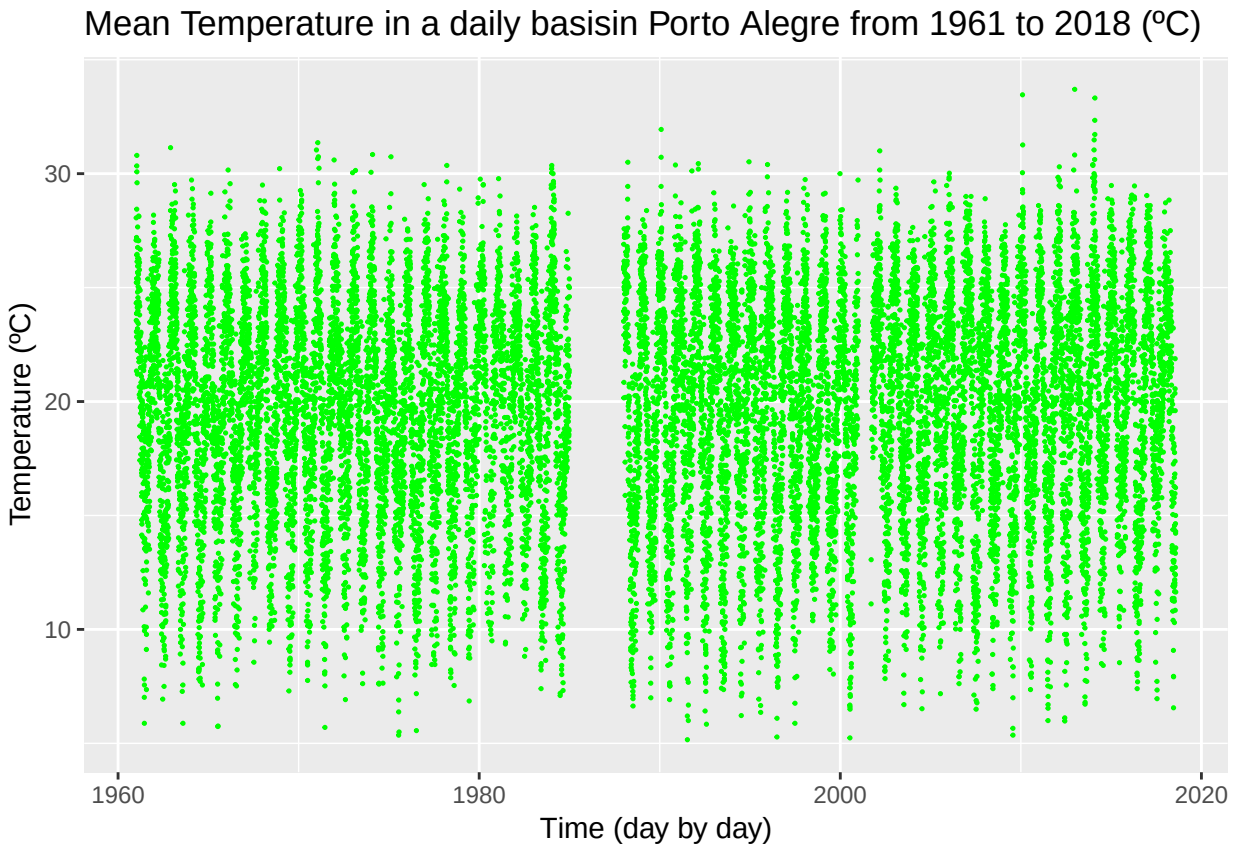
```
dfmerged %>%
  ggplot(aes(Date, Preciptation)) +
  geom_point(na.rm=TRUE, size=0.25, color="lightblue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm)")
```



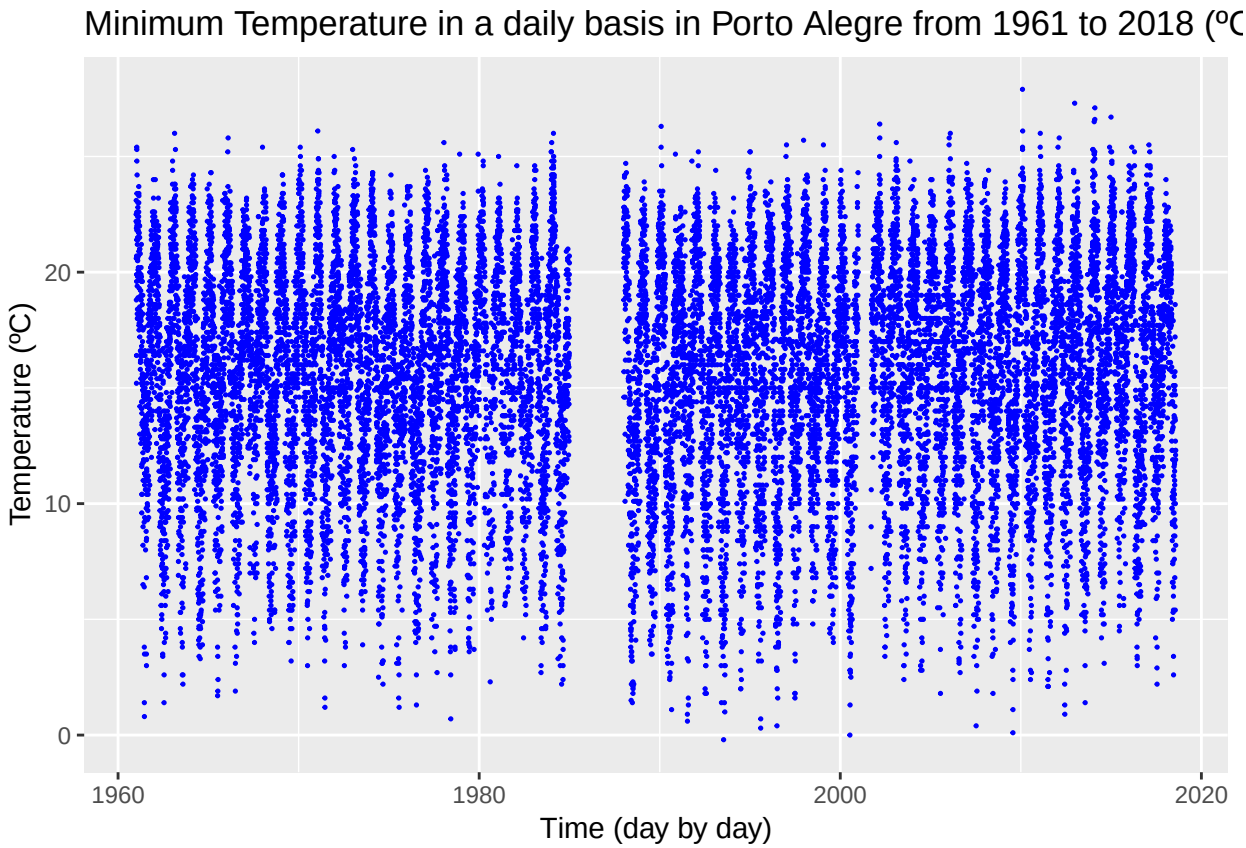

```
dfmerged %>%
  ggplot(aes(Date, MaxTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="red") +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Maximum Temperature reached by day in Porto Alegre from 1961 to 2018 (ºC)")
```




```
dfmerged %>%
  ggplot(aes(Date, MeanTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="green") +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Mean Temperature in a daily basisin Porto Alegre from 1961 to 2018 (ºC)")
```

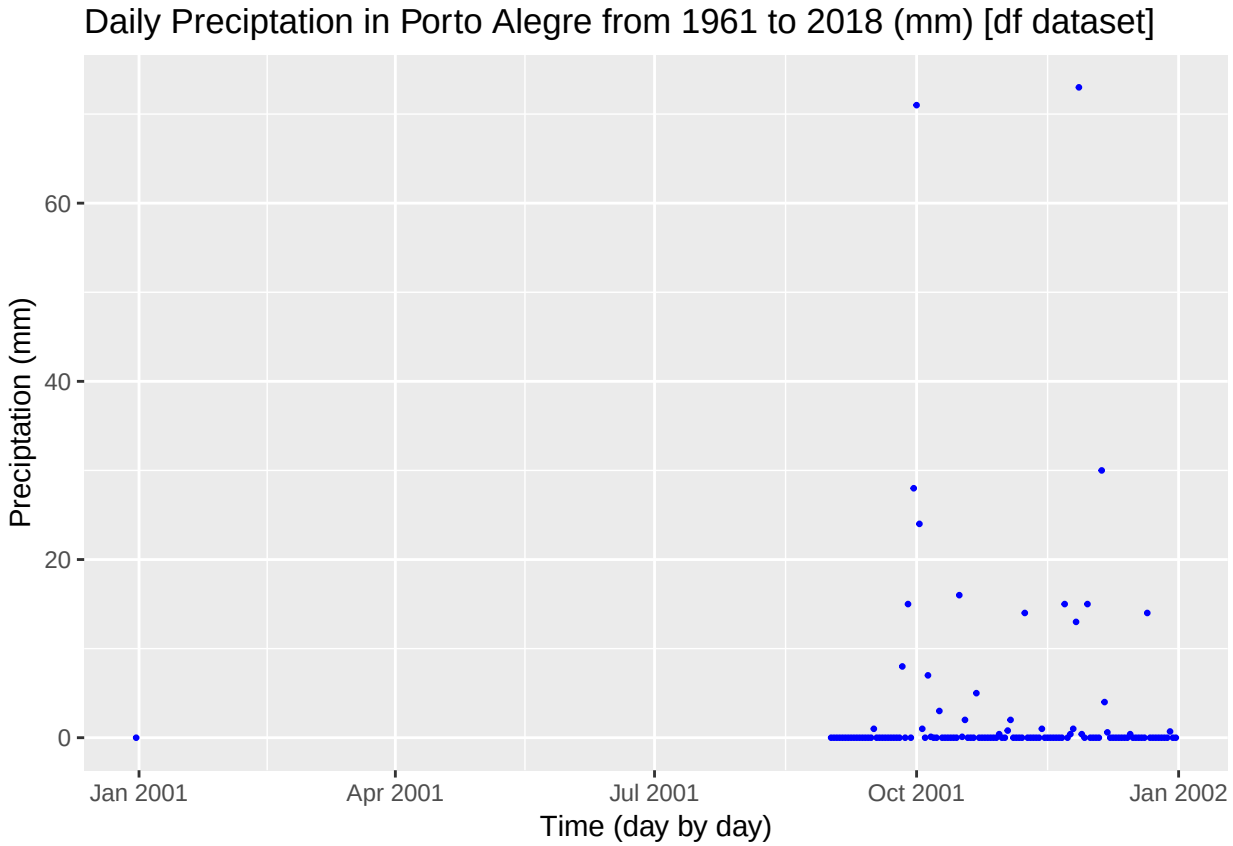



```
dfmerged %>%
  ggplot(aes(Date, MinTemperature)) +
  geom_point(na.rm=TRUE, size=0.25, color="blue") +
  ylab("Temperature (ºC)") + xlab("Time (day by day)") +
  ggtitle("Minimum Temperature in a daily basis in Porto Alegre from 1961 to 2018 (ºC)")
```

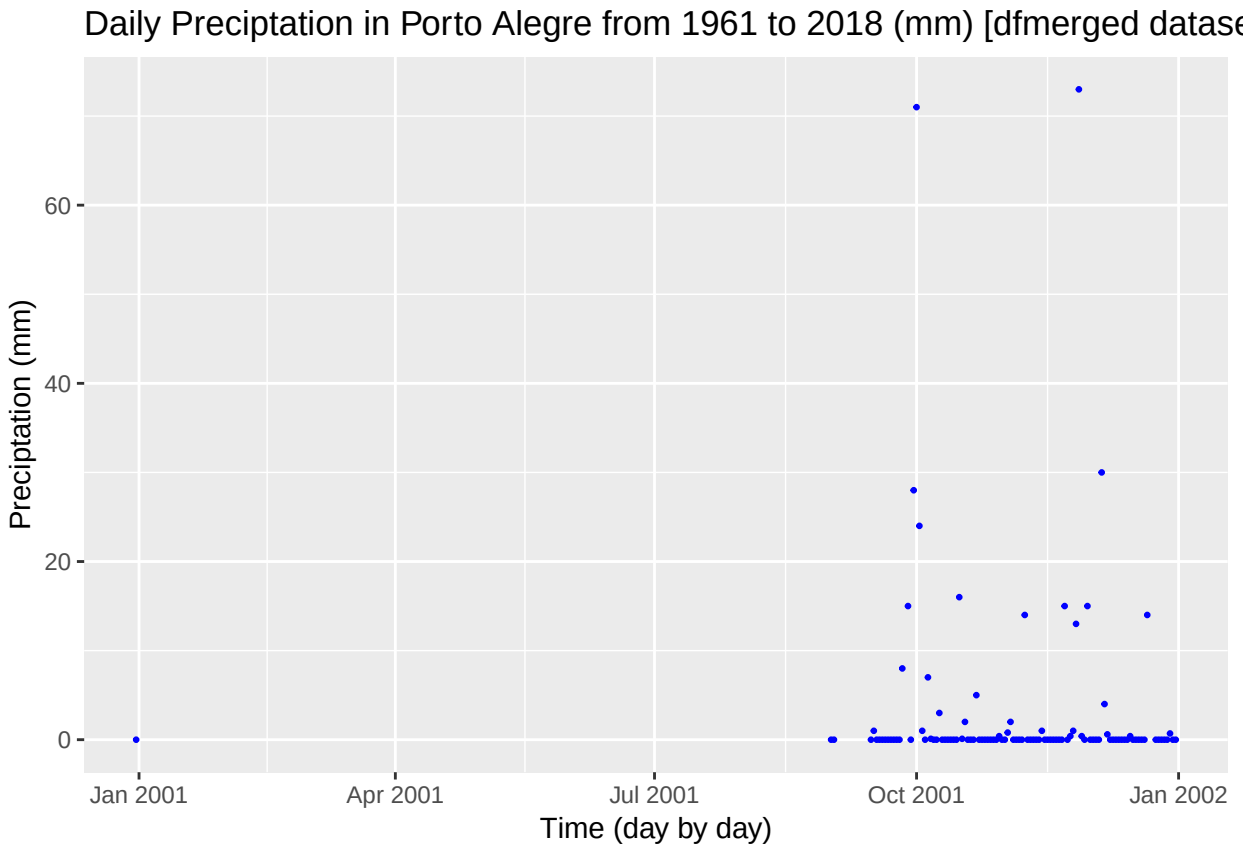


It feels like df, this is good. Let's take a closer look an 2000-2001 and compare df and dfmerged:

```
df %>%
  filter(Date >= as.Date('2000-12-31')) %>%
  filter(Date <= as.Date('2001-12-31')) %>%
  ggplot(aes(Date, Preciptation)) +
  geom_point(na.rm=TRUE, size=0.5, color="blue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm) [df dataset]")
```



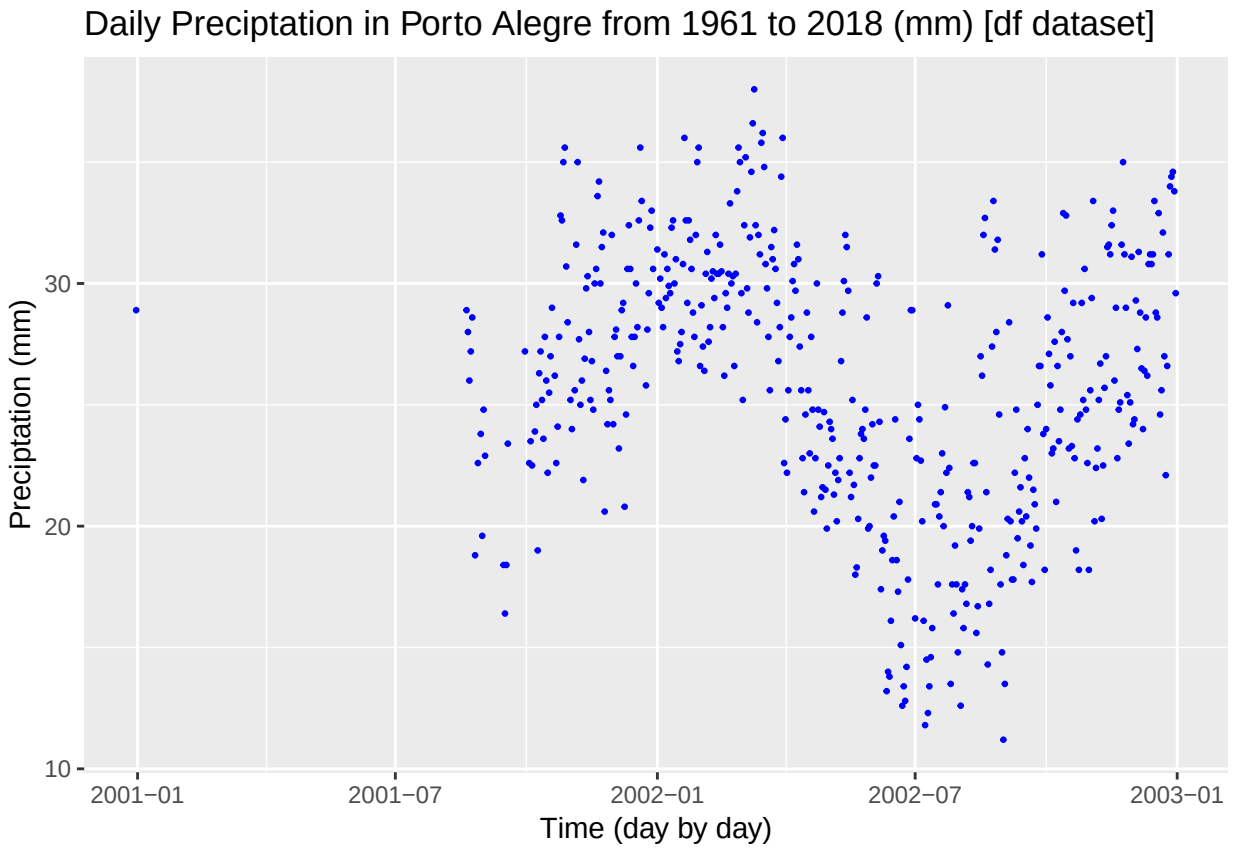

```
dfmerged %>%
  filter(Date >= as.Date('2000-12-31')) %>%
  filter(Date <= as.Date('2001-12-31')) %>%
  ggplot(aes(Date, Preciptation)) +
  geom_point(na.rm=TRUE, size=0.5, color="blue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm) [dfmerged dataset]")
```



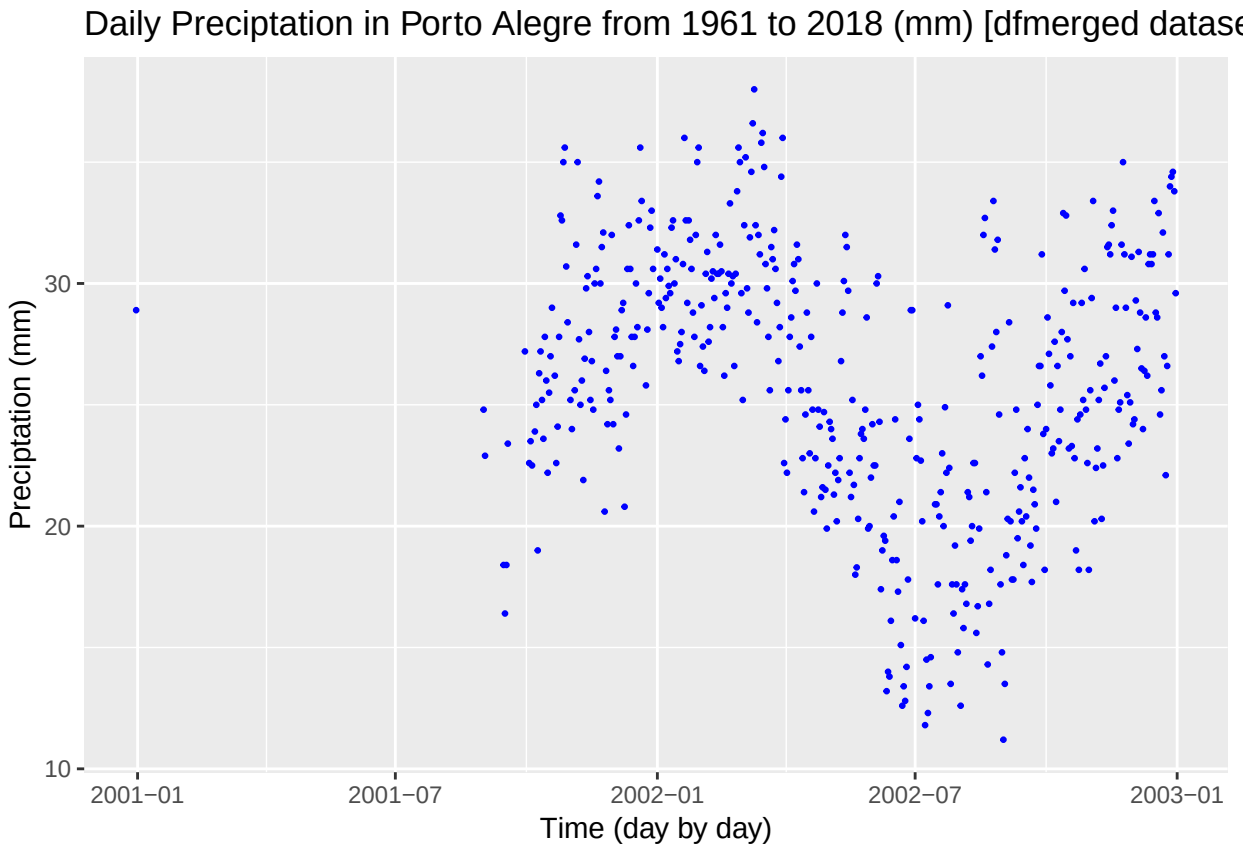
This is not good. dfmerged seems to have some rows missing... But how many? dfmerged wa supposed to have the same number of observations of df. Maybe there are some missing values for **Preciptation**.

Take a look at variables **MaxTemperature, MeanTemperature** and **MinTemperature** for this interval.

```
df %>%
  filter(Date >= as.Date('2000-12-31')) %>%
  filter(Date <= as.Date('2002-12-31')) %>%
  ggplot(aes(Date, MaxTemperature)) +
  geom_point(na.rm=TRUE, size=0.5, color="blue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm) [df dataset]")
```




```
dfmerged %>%
  filter(Date >= as.Date('2000-12-31')) %>%
  filter(Date <= as.Date('2002-12-31')) %>%
  ggplot(aes(Date, MaxTemperature)) +
  geom_point(na.rm=TRUE, size=0.5, color="blue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm) [dfmerged dataset]")
```



Precisely the same behaviour for the same interval, thus, the observations as a hole are missing. How many of them? Just 15 and almost sure that are those precise 12 dots of the previous chart and other 3 don't know where.

Observation I know that there are only 15 lines missing in dfmerged because this dataset has been made by the merge of df12 (19315) and df00 (19314) rows, but dfmerged has only 19299 rows.

But this should not happen. Maybe the coalesce function is not doing what I wanted to do, that is, merge two columns and select the non-missing value for each line.

I will try do redefine dfmerged by other means and will try to make the same plot again. Note that finding this mistake was a complete coincidence.

This time I try the function *pmax* instead of *coalesce*:

```
dfmerged <- merge(df00, df12, by="Date", all=FALSE)

dfmerged <- (dfmerged %>%
  mutate(MeanRelativeHumidity = pmax(MeanRelativeHumidity.x, MeanRelativeHumidity.y,na.rm=TRUE)) %>%
  mutate(MeanTemperature = pmax(MeanTemperature.x, MeanTemperature.y,na.rm=TRUE)) %>%
  mutate(MeanWindVelocity = pmax(MeanWindVelocity.x, MeanWindVelocity.y,na.rm=TRUE)) %>%
  mutate(MaxTemperature = pmax(MaxTemperature.x, MaxTemperature.y,na.rm=TRUE)) %>%
  mutate(Evaporation = pmax(Evaporation.x, Evaporation.y,na.rm=TRUE)) %>%
  mutate(Insolation = pmax(Insolation.x, Insolation.y,na.rm=TRUE)) %>%
  mutate(Preciptation = pmax(Preciptation.y, Preciptation.x,na.rm=TRUE)) %>%
  mutate(MinTemperature = pmax(MinTemperature.y, MinTemperature.x,na.rm=TRUE))
)

dfmerged$Preciptation.x <- NULL
```

```
dfmerged$MaxTemperature.x <- NULL
dfmerged$MinTemperature.x <- NULL
dfmerged$Insolation.x <- NULL
dfmerged$Evaporation.x <- NULL
dfmerged$MeanTemperature.x <- NULL
dfmerged$MeanRelativeHumidity.x <- NULL
dfmerged$MeanWindVelocity.x <- NULL
dfmerged$Season.x <- NULL
dfmerged$Month.x <- NULL
dfmerged$Decade.x <- NULL
dfmerged$Preciptation.y <- NULL
dfmerged$MaxTemperature.y <- NULL
dfmerged$MinTemperature.y <- NULL
dfmerged$Insolation.y <- NULL
dfmerged$Evaporation.y <- NULL
dfmerged$MeanTemperature.y <- NULL
dfmerged$MeanRelativeHumidity.y <- NULL
dfmerged$MeanWindVelocity.y <- NULL
dfmerged$Season.y <- NULL
dfmerged$Month.y <- NULL
dfmerged$Decade.y <- NULL
```

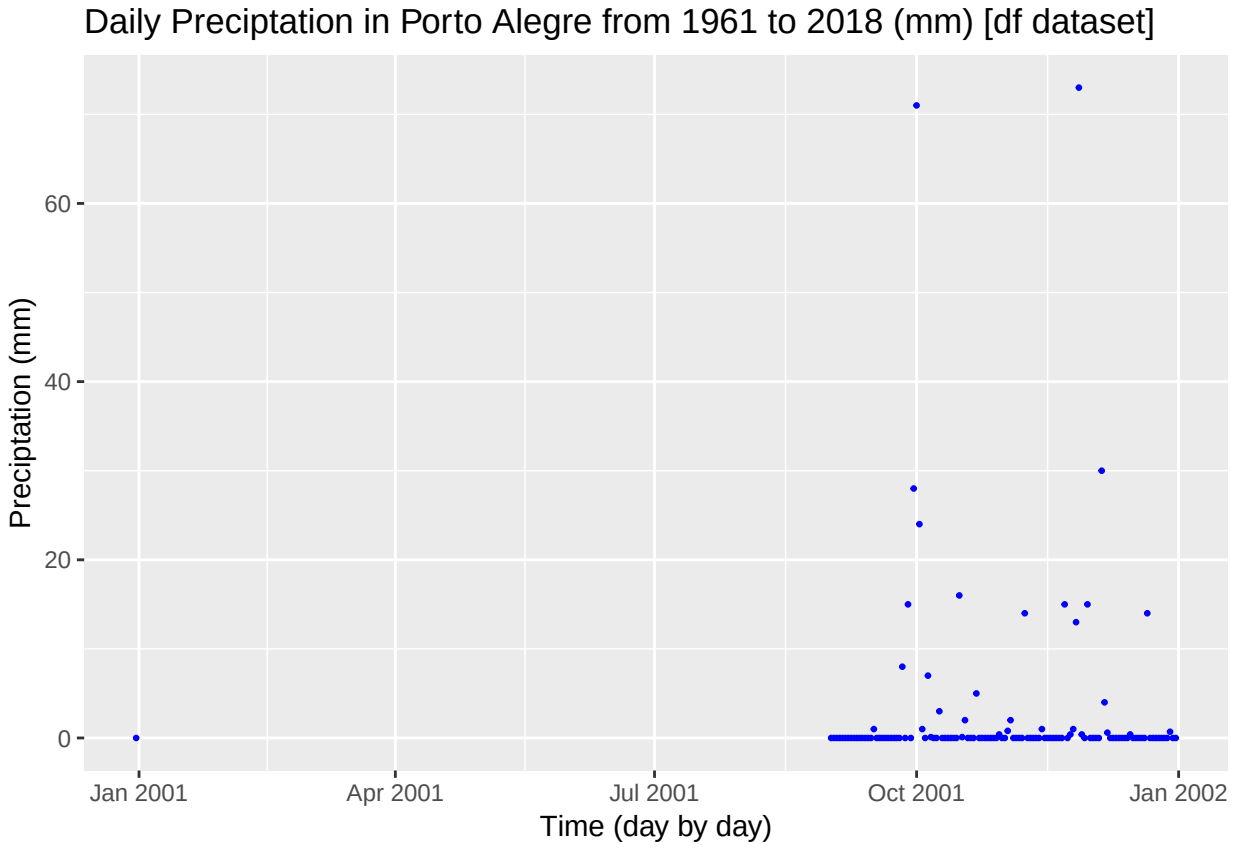
So, let's take a look in the summary of dfmerged and plot again those graphics:

```
summary(dfmerged)
```

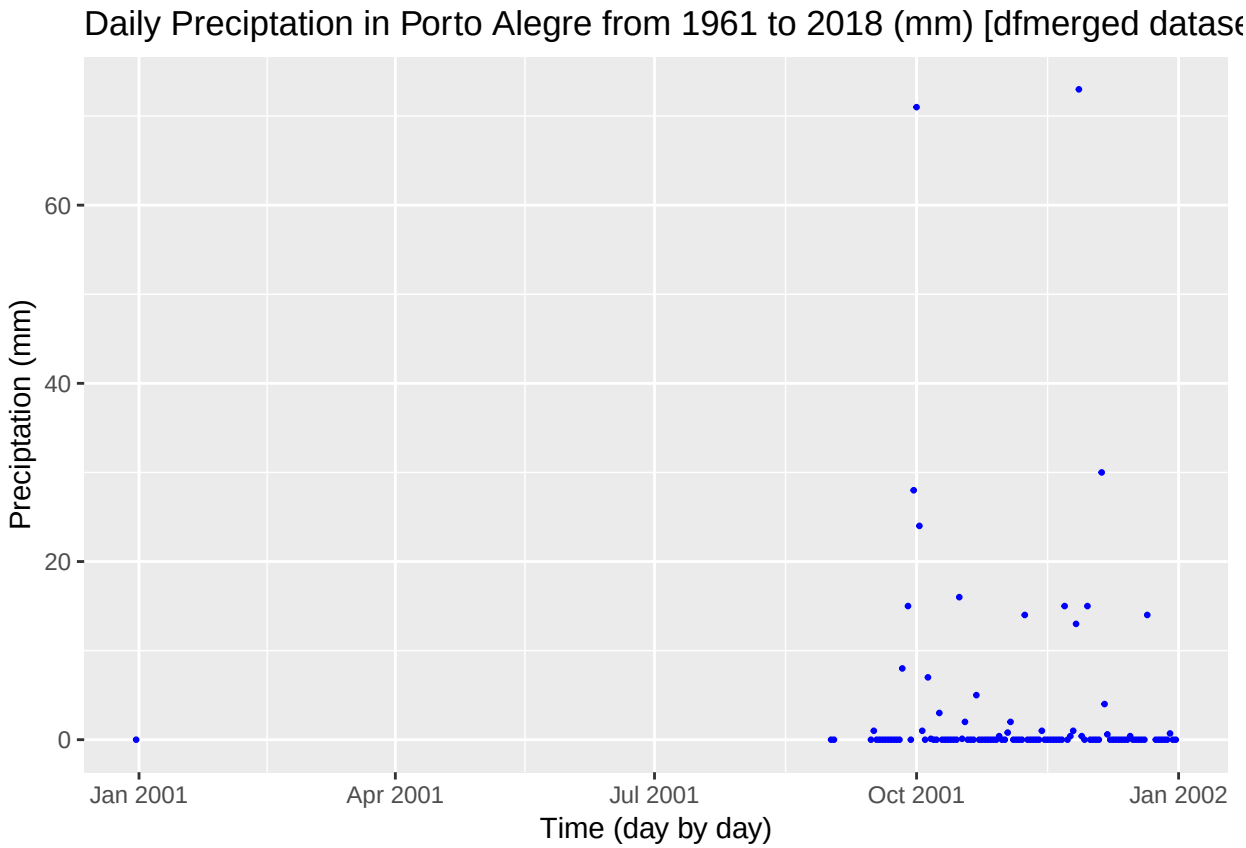
```
##       Date            MeanRelativeHumidity MeanTemperature
##  Min.   :1961-01-01   Min.   :37.50        Min.   : 5.16
##  1st Qu.:1974-04-18   1st Qu.:69.25        1st Qu.:16.30
##  Median :1991-06-10   Median :76.50        Median :20.08
##  Mean   :1990-01-04   Mean   :76.45        Mean   :19.66
##  3rd Qu.:2005-05-15   3rd Qu.:84.00        3rd Qu.:23.40
##  Max.   :2018-07-31   Max.   :99.75        Max.   :33.70
##                       NA's   :40           NA's   :36
##  MeanWindVelocity   MaxTemperature   Evaporation       Insolation
##  Min.   :   0.000   Min.   : 7.40   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:   1.000   1st Qu.:21.20   1st Qu.: 1.300   1st Qu.: 2.400
##  Median :   1.800   Median :25.50   Median : 2.200   Median : 7.000
##  Mean   :   8.463   Mean   :25.21   Mean   : 2.481   Mean   : 6.071
##  3rd Qu.:   2.767   3rd Qu.:29.40   3rd Qu.: 3.300   3rd Qu.: 9.200
##  Max.   :6216.000   Max.   :40.60   Max.   :20.700   Max.   :13.200
##  NA's   :1          NA's   :27      NA's   :688      NA's   :130
##   Preciptation     MinTemperature
##  Min.   :  0.000   Min.   :-0.2
##  1st Qu.:  0.000   1st Qu.:12.5
##  Median :  0.000   Median :16.2
##  Mean   :  3.792   Mean   :15.7
##  3rd Qu.:  1.700   3rd Qu.:19.4
##  Max.   :149.600   Max.   :27.9
##  NA's   :10        NA's   :15
```

```
df %>%
  filter(Date >= as.Date('2000-12-31')) %>%
  filter(Date <= as.Date('2001-12-31')) %>%
  ggplot(aes(Date, Preciptation)) +
```

```
  geom_point(na.rm=TRUE, size=0.5, color="blue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm) [df dataset]")
```



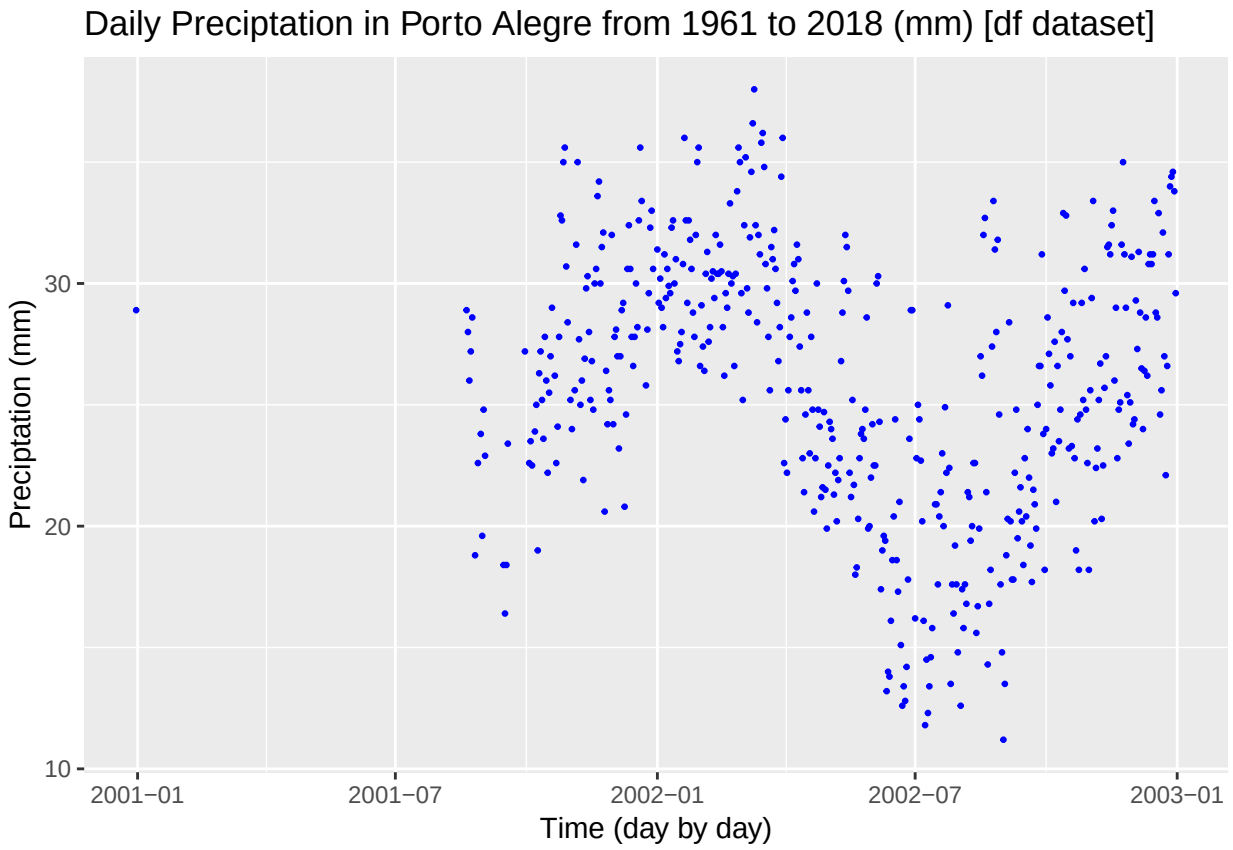

```
dfmerged %>%
  filter(Date >= as.Date('2000-12-31')) %>%
  filter(Date <= as.Date('2001-12-31')) %>%
  ggplot(aes(Date, Preciptation)) +
  geom_point(na.rm=TRUE, size=0.5, color="blue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm) [dfmerged dataset]")
```



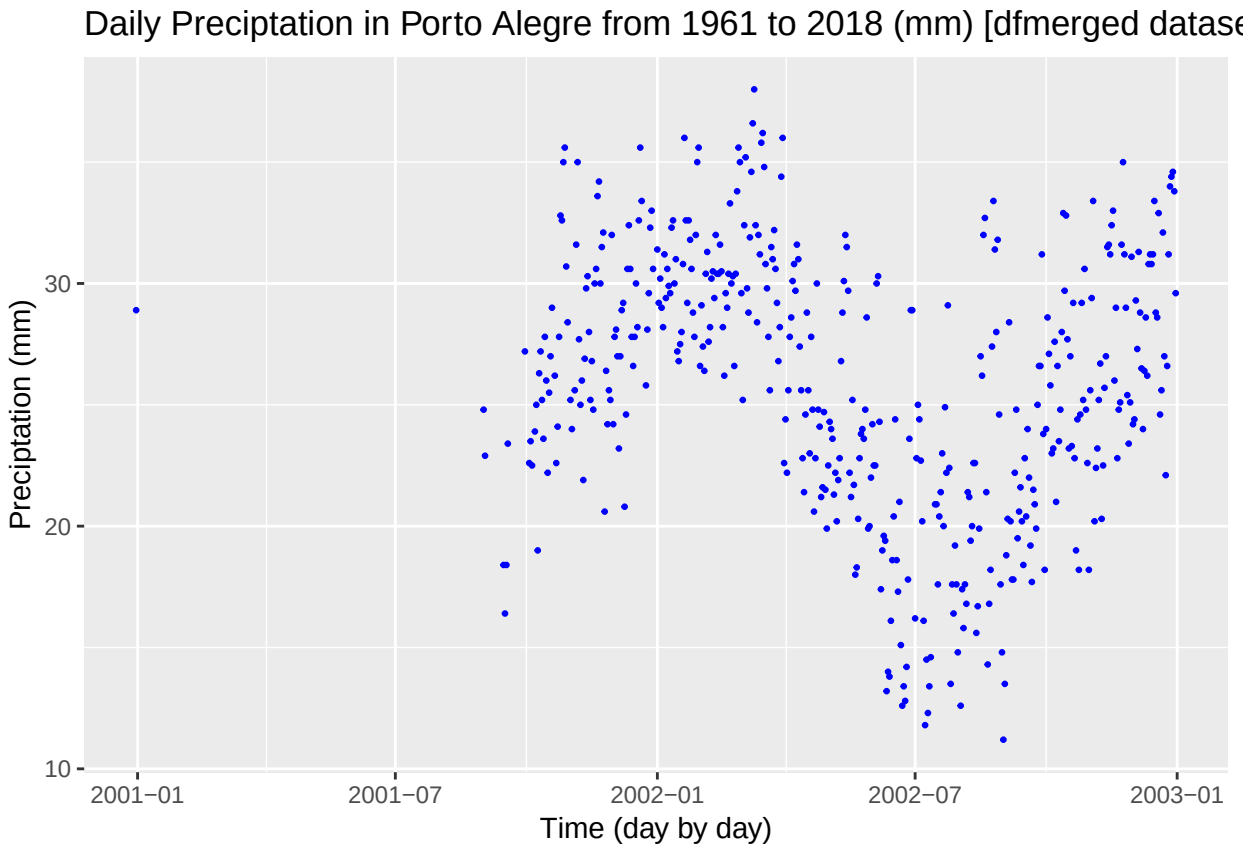
This does not solves the problem, and thus, it is an evidence that the method is good and there is something with the data.

And I belevie that those 15 rows missing in dfmerged are precisely those fell points that are not shown in dfmerged.

```
df %>%
  filter(Date >= as.Date('2000-12-31')) %>%
  filter(Date <= as.Date('2002-12-31')) %>%
  ggplot(aes(Date, MaxTemperature)) +
  geom_point(na.rm=TRUE, size=0.5, color="blue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm) [df dataset]")
```




```
dfmerged %>%
  filter(Date >= as.Date('2000-12-31')) %>%
  filter(Date <= as.Date('2002-12-31')) %>%
  ggplot(aes(Date, MaxTemperature)) +
  geom_point(na.rm=TRUE, size=0.5, color="blue") +
  ylab("Preciptation (mm)") + xlab("Time (day by day)") +
  ggtitle("Daily Preciptation in Porto Alegre from 1961 to 2018 (mm) [dfmerged dataset]")
```



Hence, I just assume that the data is well formatted to the task. Don't really know how to fix this problem.

New convinience variables

We can enhance the manipulation of the data adding new variables. Those new variables are **Month, Season, Year** and **Decade**.

We start with **Month** column:

```
dfmerged <- (dfmerged %>%
         mutate(Month = month(Date))
)
```

hence we add **Year** column

```
dfmerged <- (dfmerged %>%
          mutate(Year = as.factor(year(Date)))
)
```

and why no **Decade** column with

```
dfmerged <- (dfmerged %>%
         mutate(Decade = as.factor((year(Date)-1) -  ((year(Date) - 1) %% 10)  ))
)
```

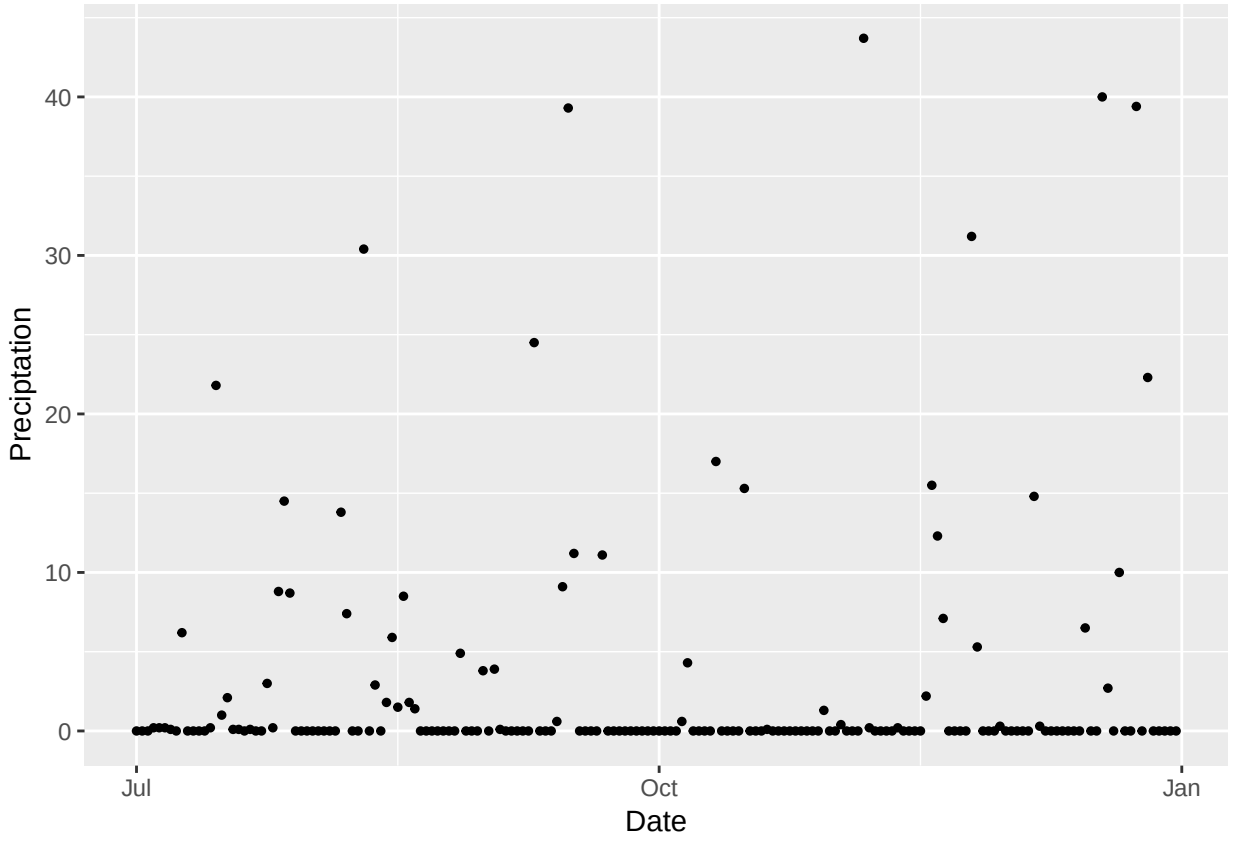
Let's take a look in this time variables

```
dfmerged %>%
  select(Date, Month, Year, Decade) %>%
  head()
```

```
##         Date Month Year Decade
## 1 1961-01-01     1 1961   1960
## 2 1961-01-02     1 1961   1960
## 3 1961-01-03     1 1961   1960
## 4 1961-01-04     1 1961   1960
## 5 1961-01-05     1 1961   1960
## 6 1961-01-06     1 1961   1960
```

Now it is simple to plot the preciptation between July and December 2017. Just for the sake of an example, we can plot now

```
dfmerged %>%
  filter(Year == 2006) %>%
  filter(Month >= 7) %>%
  filter(Month <= 12) %>%
  ggplot(aes(Date, Preciptation)) +
  geom_point(na.rm=TRUE, size=1)
```



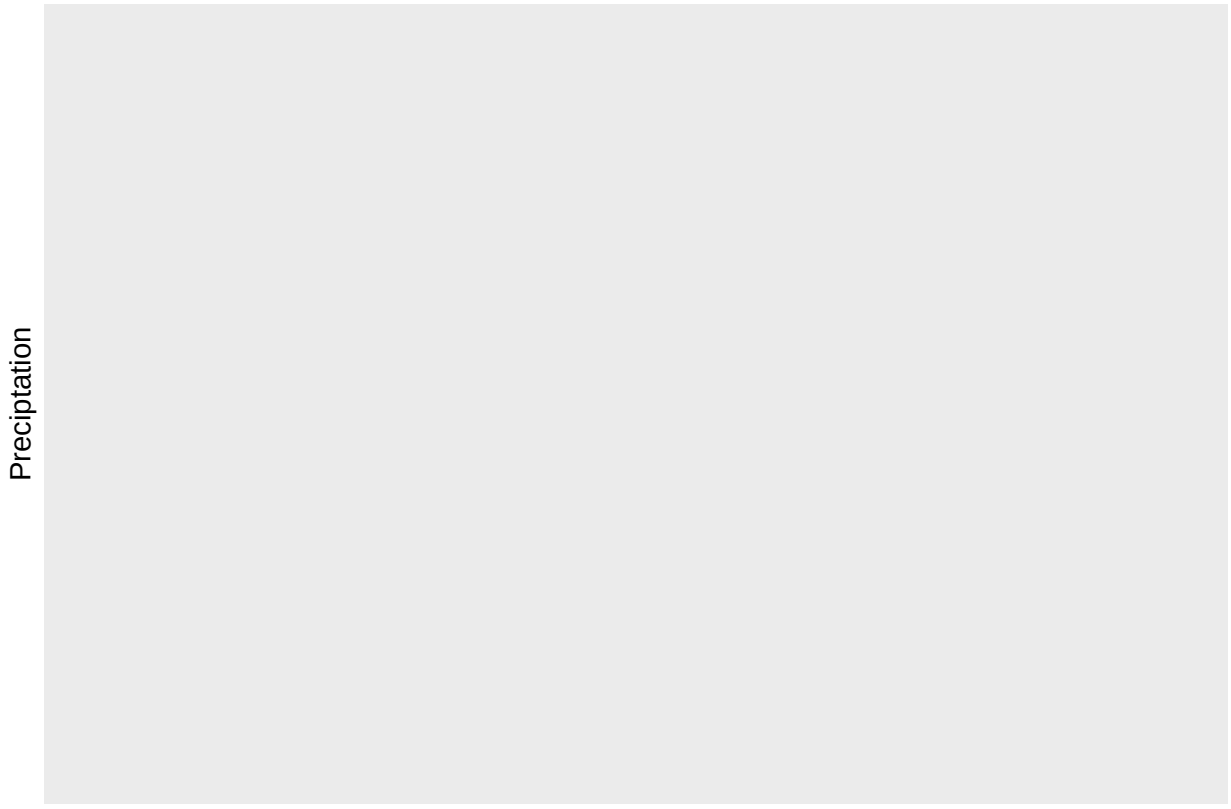
Additionaly, we can consider the **Season** variable in which a given observation is made. We can add the **Season** column by

```
# Adapted from
# https://stackoverflow.com/questions/9500114/find-which-season-a-particular-date-belongs-to
# for the local Winter Solstice, Spring Equinox, Summer Solstice and Fall Equinox values for Brazil
```

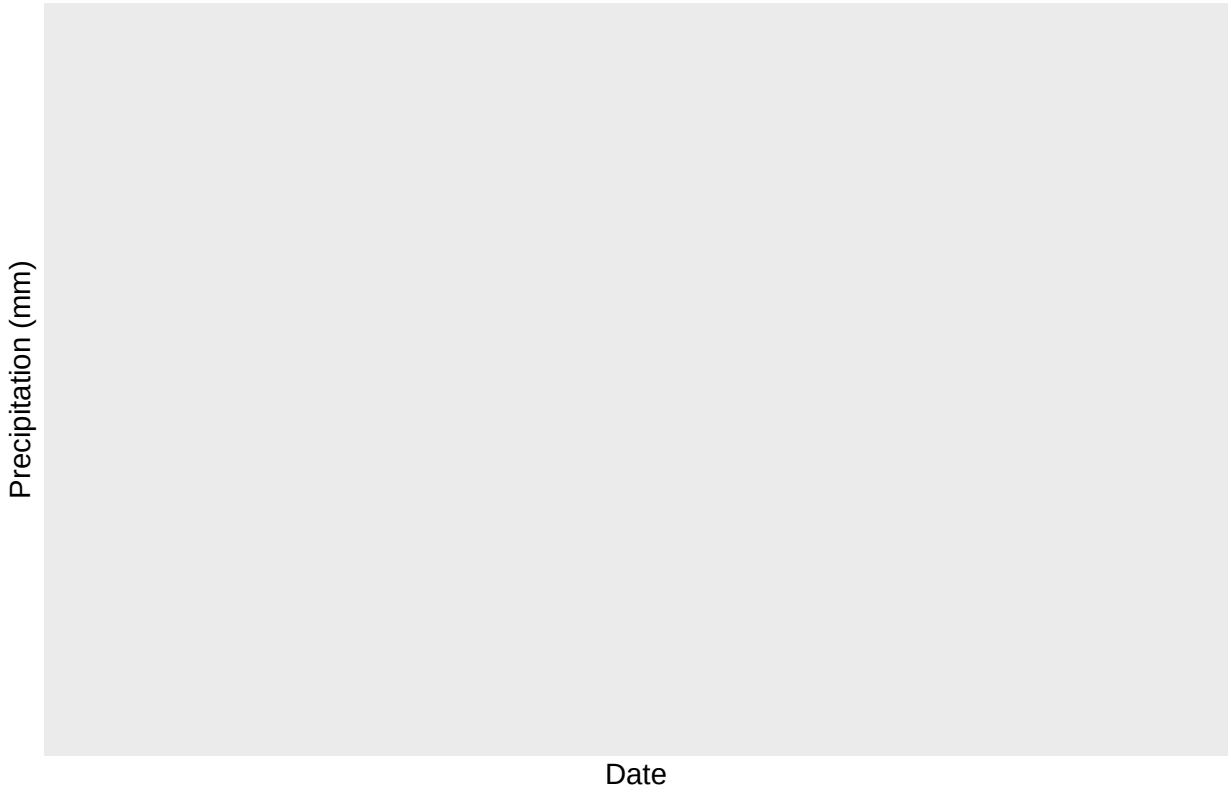
```
getSeason <- function(DATES) {
    WS <- as.Date("2012-5-21", format = "%Y-%m-%d") # Winter Solstice
    SE <- as.Date("2012-8-22",  format = "%Y-%m-%d") # Spring Equinox
    SS <- as.Date("2012-12-21",  format = "%Y-%m-%d") # Summer Solstice
    FE <- as.Date("2012-3-20",  format = "%Y-%m-%d") # Fall Equinox

    # Convert dates from any year to 2012 dates
    d <- as.Date(strftime(DATES, format="2012-%m-%d"))

    ifelse (d >= WS & d < SE, "Winter",
      ifelse (d >= SE & d < SS, "Spring",
        ifelse (d >= SS | d < FE, "Summer", "Fall")))
}

dfmerged <- (dfmerged %>%
  mutate(Season = as.factor(getSeason(Date))) %>%
  head()
)
```

Another example before trying to answer the questions made in the begining of this work, we can use the **Season** variable to give different collors to points:

```
dfmerged %>%
  filter(Year == 2006) %>%
  ggplot() +
  geom_jitter(aes(Season, Preciptation, colour=Season,), na.rm=TRUE, size=2) +
  theme(legend.position="none")
```



```
dfmerged %>%
  filter(Year == 2006) %>%
  ggplot() +
    geom_bar(aes(Date, Preciptation, color=Season), stat="identity", na.rm = TRUE) +
    ggtitle("") +
    xlab("Date") + ylab("Precipitation (mm)")
```




```
    #scale_x_date(labels=date_format ("%b %y"), breaks=date_breaks("1 year")) +
    #theme(plot.title = element_text(lineheight=.8, face="bold", size = 20)) +
    #theme(text = element_text(size=18))
```

Presenting the results

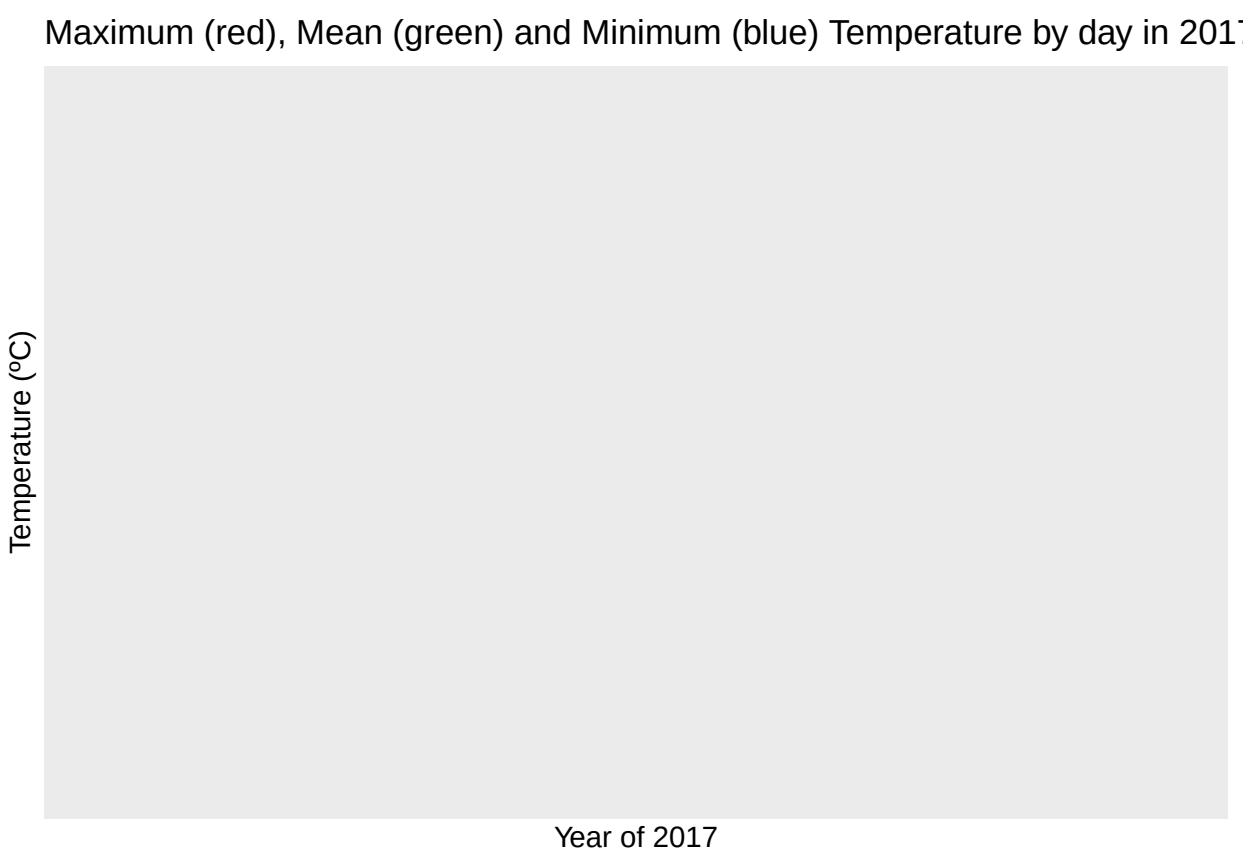
We recall the questions that are to be answered:

1. What are the average temperature (monthly, yearly and by decade)
2. The average precipitation (monthly, yearly and by decade);
3. Is there a global trend in Porto Alegre's weather reagarding temperature and preciptation ?
4. How the does the weather changes regarding the seasons? Specificaly: 4.1. Does the average winter's temperature is getting lower ? 4.2. Does the average summer's temperature is getting higher ?
5. When the top 10 lowest and higher temperatures were registred?

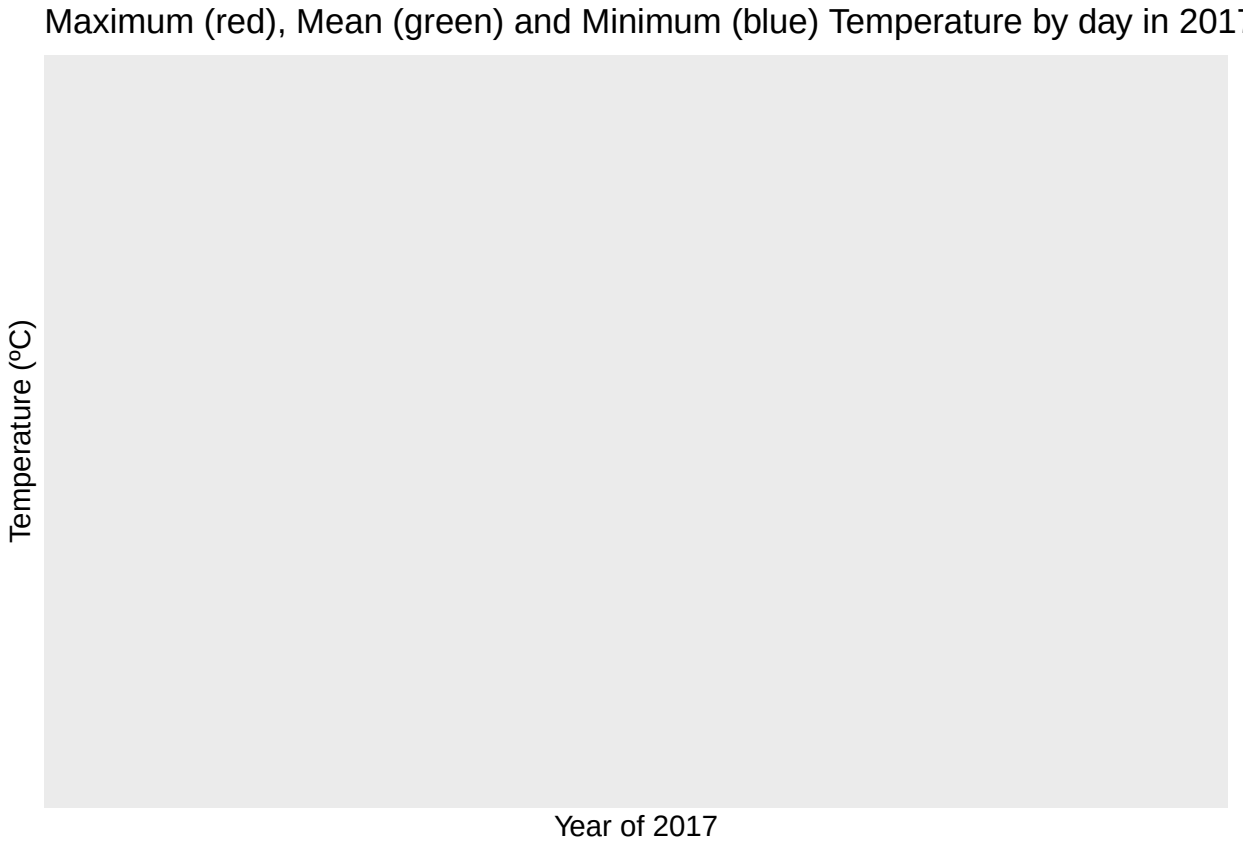
I have already presented the temperature by a scatterplot and by linechart.

```
dfmerged %>%
  filter(Date >= as.Date('2017-01-01')) %>%
  filter(Date <= as.Date('2017-12-31')) %>%
  ggplot() +
```

```
  geom_point(aes(Date, MaxTemperature), na.rm=TRUE, size=0.25, color="red") +
  geom_point(aes(Date, MeanTemperature), na.rm=TRUE, size=0.25, color="green") +
  geom_point(aes(Date, MinTemperature), na.rm=TRUE, size=0.25, color="blue") +
  xlab("Year of 2017") +
  ylab("Temperature (ºC)") +
  ggtitle("Maximum (red), Mean (green) and Minimum (blue) Temperature by day in 2017")
```




```
dfmerged %>%
  filter(Date >= as.Date('2017-01-01')) %>%
  filter(Date <= as.Date('2017-12-31')) %>%
  ggplot() +
  geom_line(aes(Date, MaxTemperature), na.rm=TRUE, size=0.25, color="red") +
  geom_line(aes(Date, MeanTemperature), na.rm=TRUE, size=0.25, color="green") +
  geom_line(aes(Date, MinTemperature), na.rm=TRUE, size=0.25, color="blue") +
  xlab("Year of 2017") +
  ylab("Temperature (ºC)") +
  ggtitle("Maximum (red), Mean (green) and Minimum (blue) Temperature by day in 2017")
```



But, there is still a lot of noisy in the results. Hence, I put in use a smooth curve as an estimate of the overall tendency of temperature:

```
for (y in 2010:2017){
dfmerged %>%
  filter(Year == y) %>%
  ggplot() +
  geom_line(aes(Date, MaxTemperature), na.rm=TRUE, size=0.25, color="red") +
  geom_smooth(aes(x=Date, y=MaxTemperature), colour = "red",size = 1) +
  geom_line(aes(Date, MeanTemperature), na.rm=TRUE, size=0.25, color="green") +
  geom_smooth(aes(x=Date, y=MeanTemperature), colour = "green",size = 1) +
  geom_line(aes(Date, MinTemperature), na.rm=TRUE, size=0.25, color="blue") +
  geom_smooth(aes(x=Date, y=MinTemperature), colour = "blue",size = 1) +
  facet_grid(rows=vars(Year))+
  xlab(paste("Year of ", y)) +
  ylab("Temperature (ºC)") +
  ggtitle(paste("Maximum (red), Mean (green) and Minimum (blue) Temperature by day in ", y))
}
```

This shows a pattern. The maximum, mean and minimum temperature by day seems to be strongly correlated and keep a constant distance from one to another.

But I try better approach to present the data.

What are the average temperature (monthly, yearly and by decade)

We first try to answer that question using a simple scatterplot. The chances are that it will not be a good solution due to overplotting.

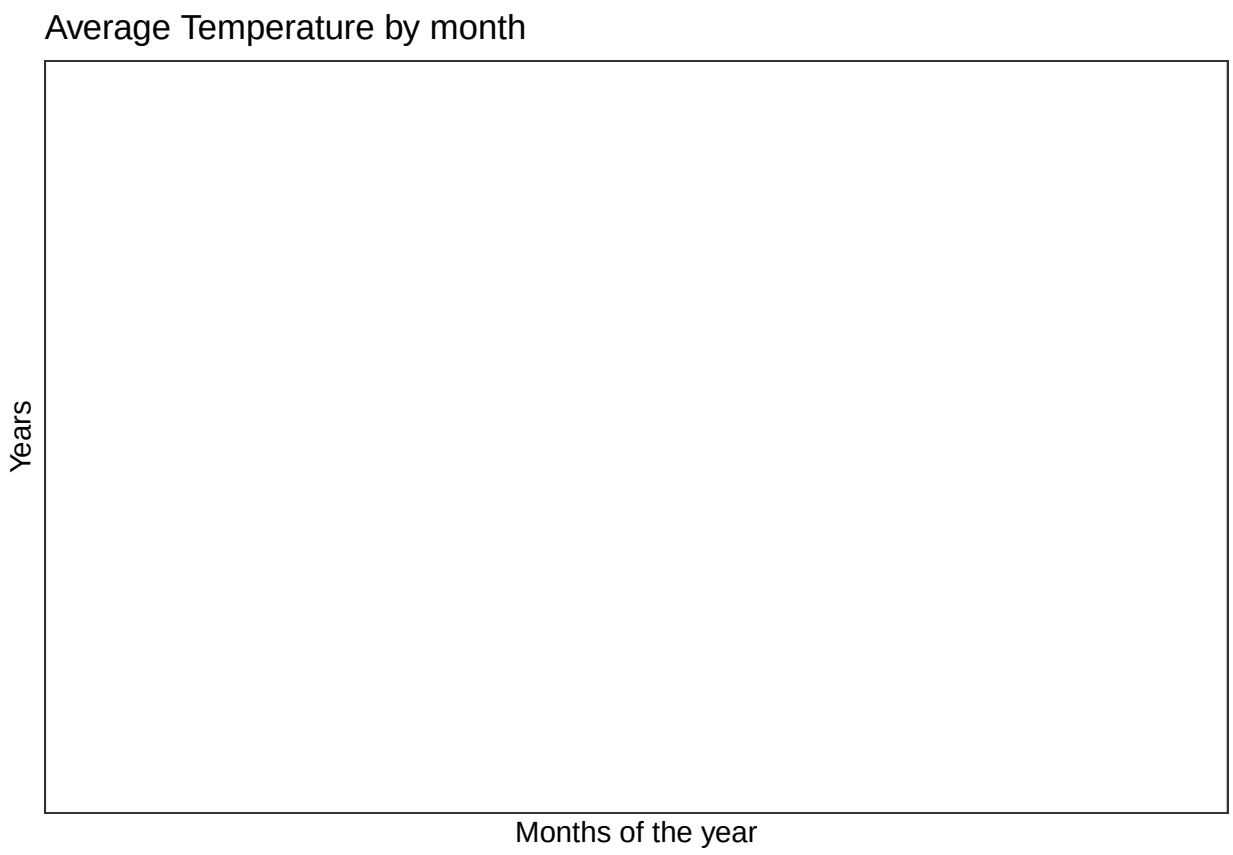
Being a narutal phenomena, we expect the temperature to oscilate from a tendence. I try to visualize this tendence using a smooth cuve.

Hence, we have a clear picture that in the year of 2017 the temperature, beginning in January 2017 have slowly, but not continuously, oscilating around a tendency of decreasy until reach July and August 2017 where the lowest temperature are registred.

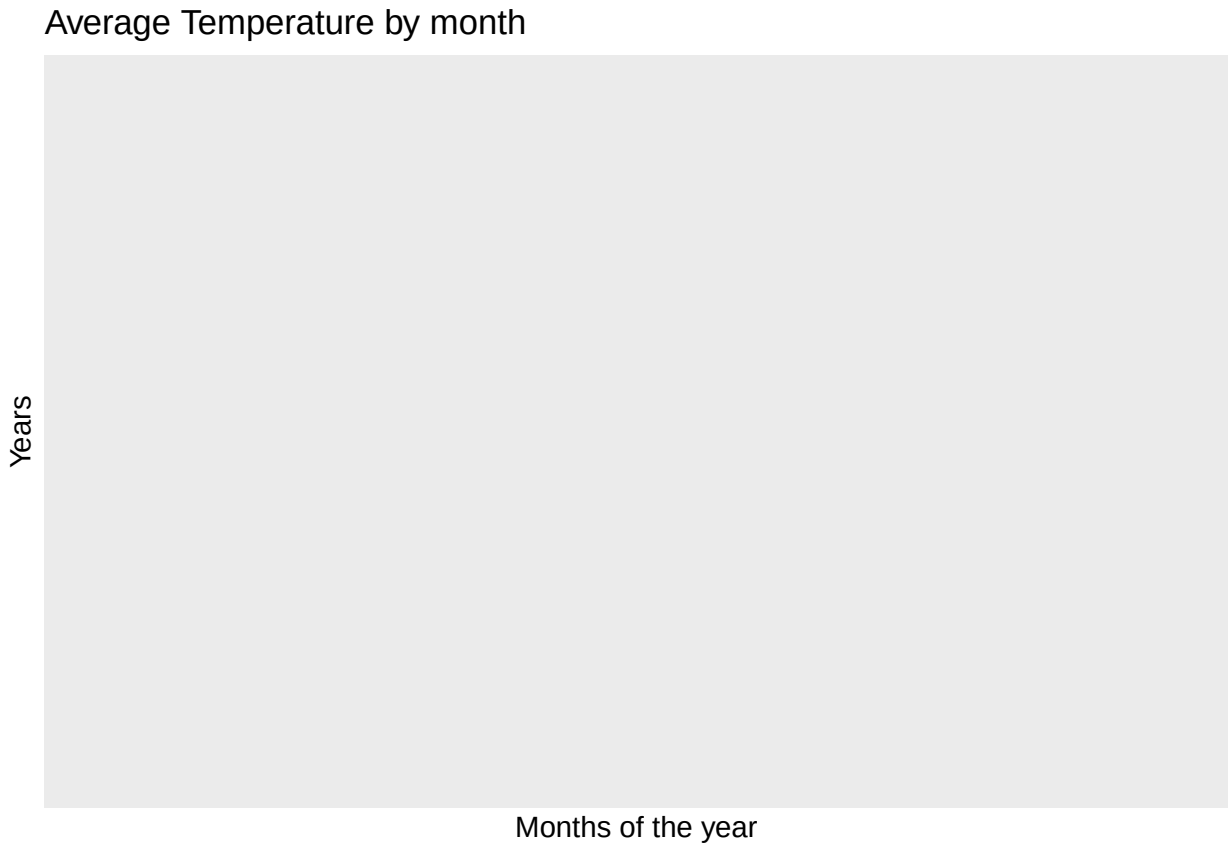
It would be nice to have a look in the other years os 2010 Decade. We can use a matrix plot to have a look on it.

Maybe other type of visualization could help me. Selecting only the observations from 2000-01-01 on we get

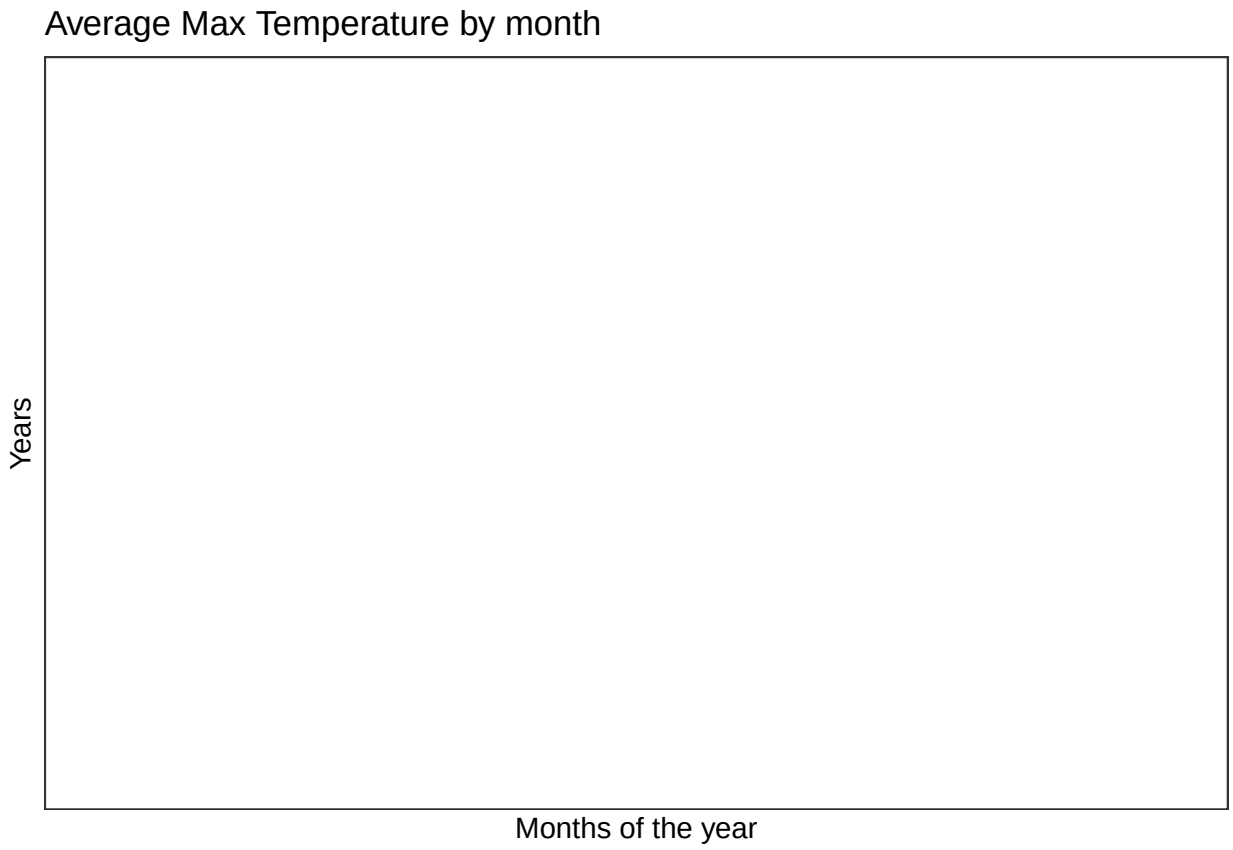
```
dfmerged %>%
  filter(Date >= as.Date('2000-01-01')) %>%
  ggplot(aes(x=as.factor(Month), y=Year, fill = MeanTemperature)) +
  geom_tile(colour = "white", na.rm = TRUE) +
  scale_color_brewer(palette="BuGn") +
  #scale_fill_gradient(low = "blue", high = "red") +
  #guides(fill=guide_legend(title="Total Incidents")) +
  theme_bw() +
  #theme_minimal() +
  labs(title = "Average Temperature by month",
       x = "Months of the year",
       y = "Years")
```



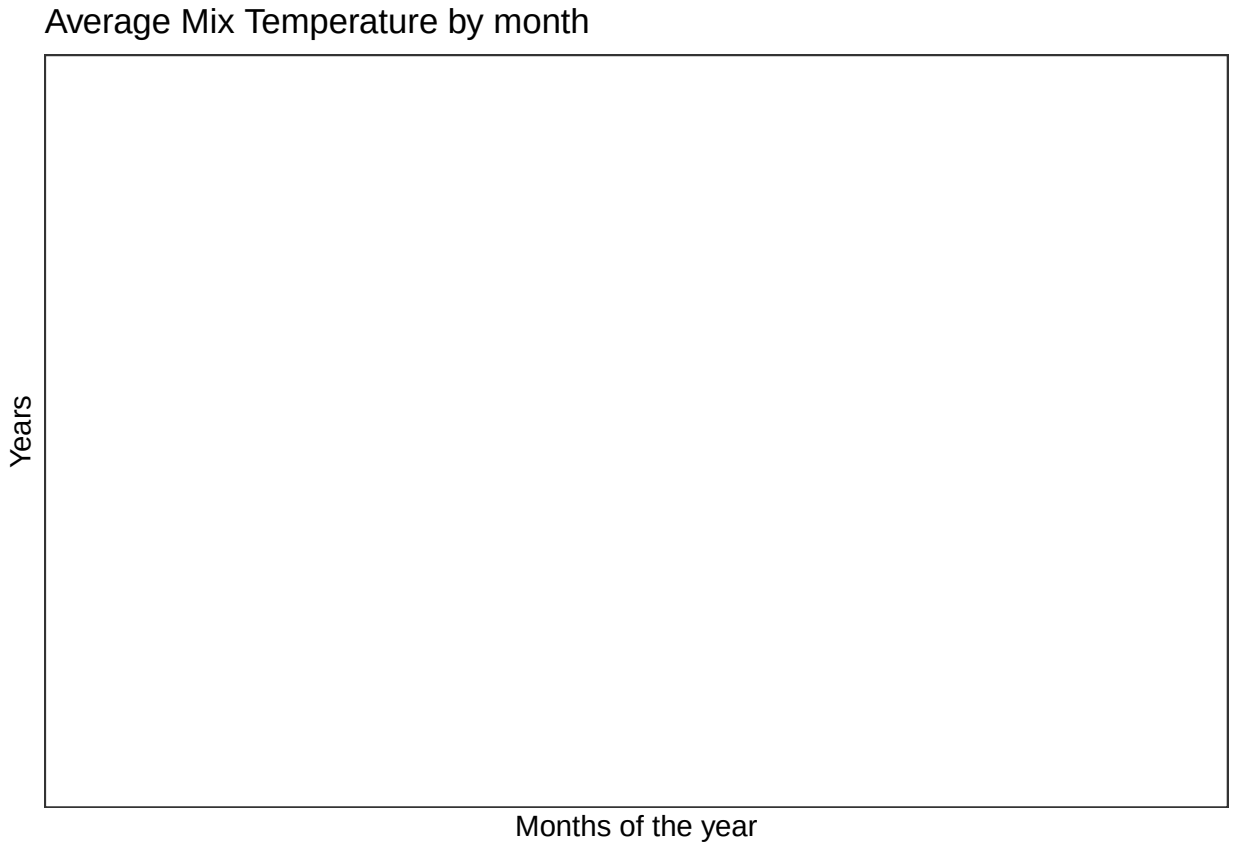
```
dfmerged %>%
  filter(Date >= as.Date('2000-01-01')) %>%
  ggplot(aes(x=as.factor(Month), y=Year, fill = Preciptation)) +
  geom_tile(colour = "white", na.rm = TRUE) +
  scale_color_brewer(palette="Blues") +
  labs(title = "Average Temperature by month", x = "Months of the year", y = "Years")
```




```
dfmerged %>%
  filter(Date >= as.Date('2000-01-01')) %>%
  ggplot(aes(x=as.factor(Month), y=Year, fill = MaxTemperature)) +
  geom_tile(colour = "white", na.rm = TRUE) +
  scale_fill_gradient(low = "blue", high = "red") +
  #guides(fill=guide_legend(title="Total Incidents")) +
  theme_bw() +
  #theme_minimal() +
  labs(title = "Average Max Temperature by month",
       x = "Months of the year",
       y = "Years")
```

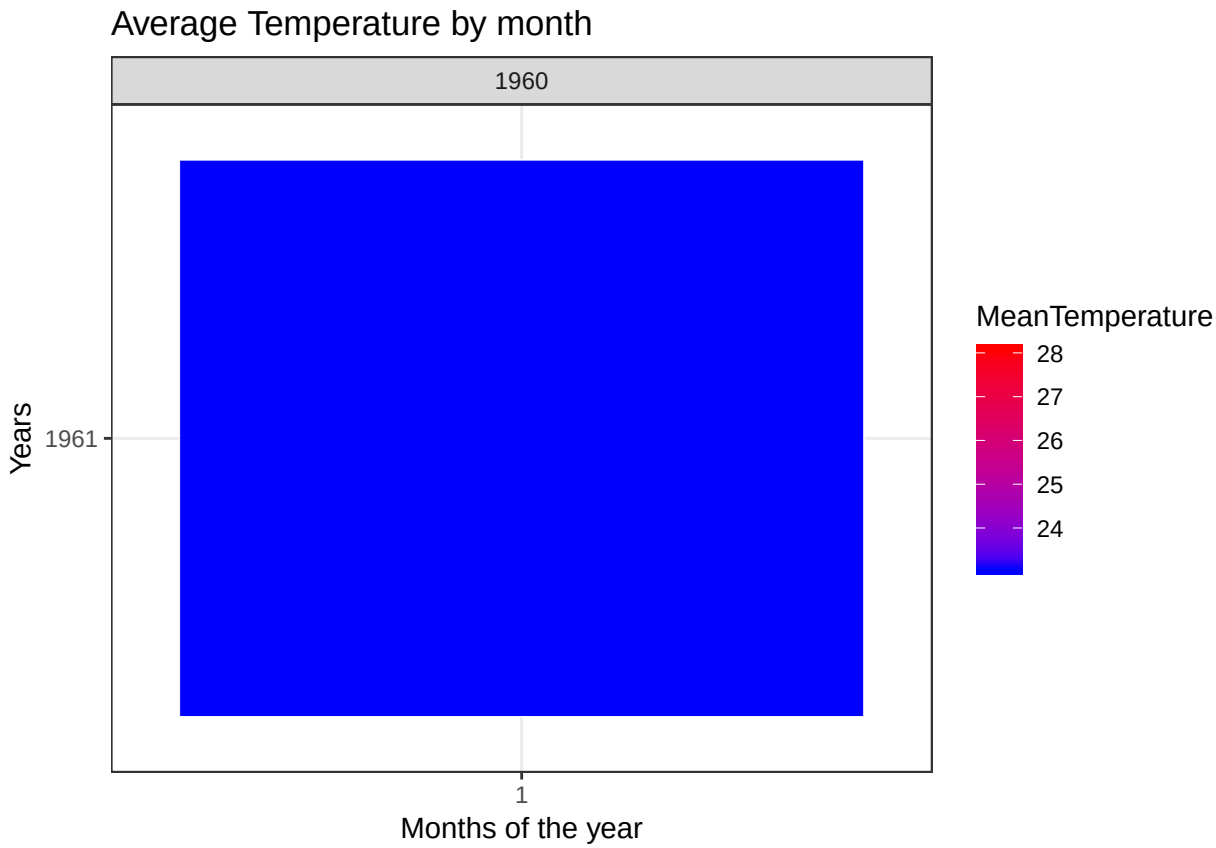



```
dfmerged %>%
  filter(Date >= as.Date('2000-01-01')) %>%
  ggplot(aes(x=as.factor(Month), y=Year, fill = MinTemperature)) +
  geom_tile(colour = "white", na.rm = TRUE) +
  scale_fill_gradient(low = "blue", high = "red") +
  #guides(fill=guide_legend(title="Total Incidents")) +
  theme_bw() +
  #theme_minimal() +
  labs(title = "Average Mix Temperature by month",
       x = "Months of the year",
       y = "Years")
```

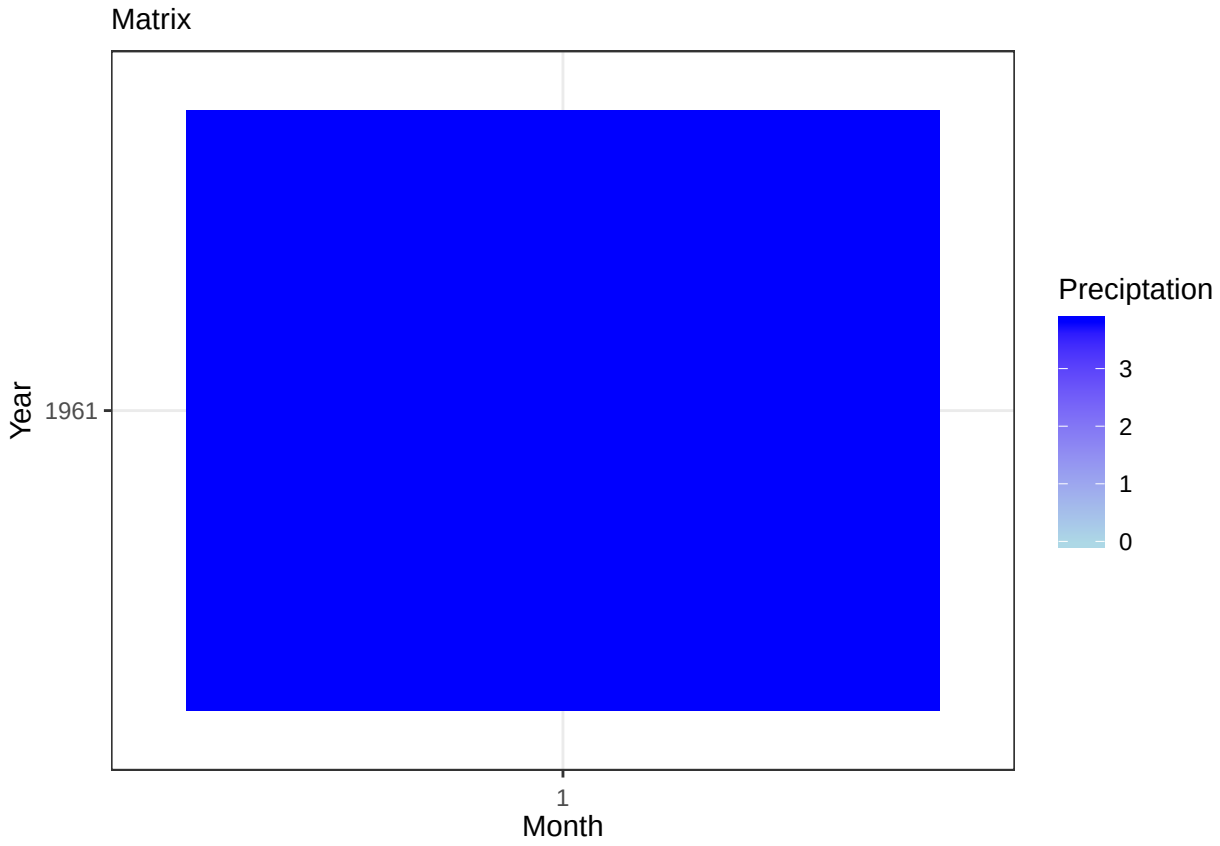


And scaling the data to feet properly from 1961 to 2018 in a **Decade** display we have

```
dfmerged %>%
  #filter(Date >= as.Date('2000-01-01')) %>%
  ggplot(aes(x=as.factor(Month), y=Year, fill = MeanTemperature)) +
  geom_tile(colour = "white", na.rm = TRUE) +
  scale_fill_gradient(low = "blue", high = "red") +
  #guides(fill=guide_legend(title="Total Incidents")) +
  theme_bw() +
  #theme_minimal() +
  labs(title = "Average Temperature by month", x = "Months of the year", y = "Years") +
  facet_wrap(~Decade, scale="free_y", nrow=2, ncol=3)
```

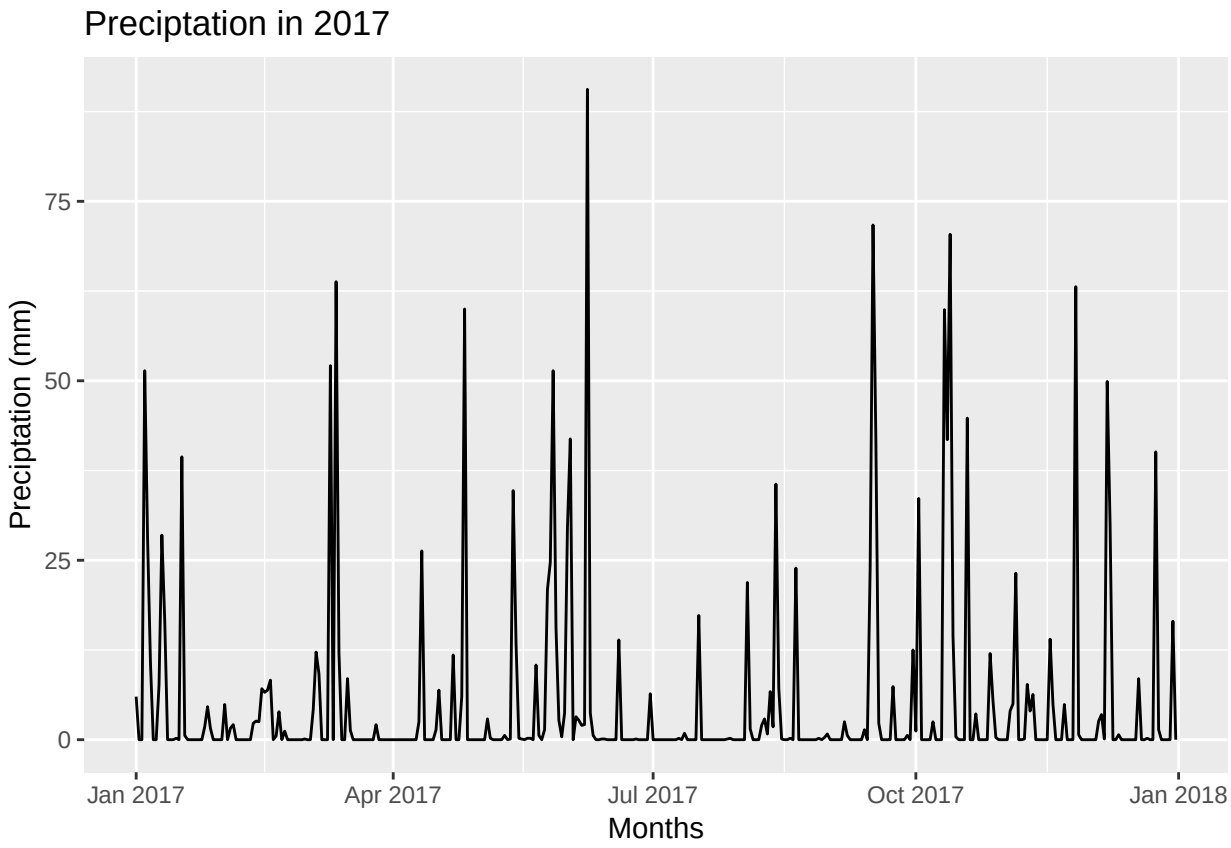



```
dfmerged %>%
  ggplot(aes(x = as.factor(Month), y = Year)) +
  geom_raster(aes(fill=Preciptation), stat="identity") +
  scale_fill_gradient(low="lightblue", high="blue") +
  labs(x="Month", y="Year", title="Matrix") +
  theme_bw() +
  theme(axis.text.x=element_text(size=9, angle=0, vjust=0.3),
        axis.text.y=element_text(size=9),
       plot.title=element_text(size=11)) +
  scale_y_discrete(expand = waiver(), position = "left")
```



The heatmaps show us that the lower temperatures occurs in month 5, 6, 7 and 8, months that coincides with the Winter season. Maybe it is a good ideia to separate the temperatures by month.

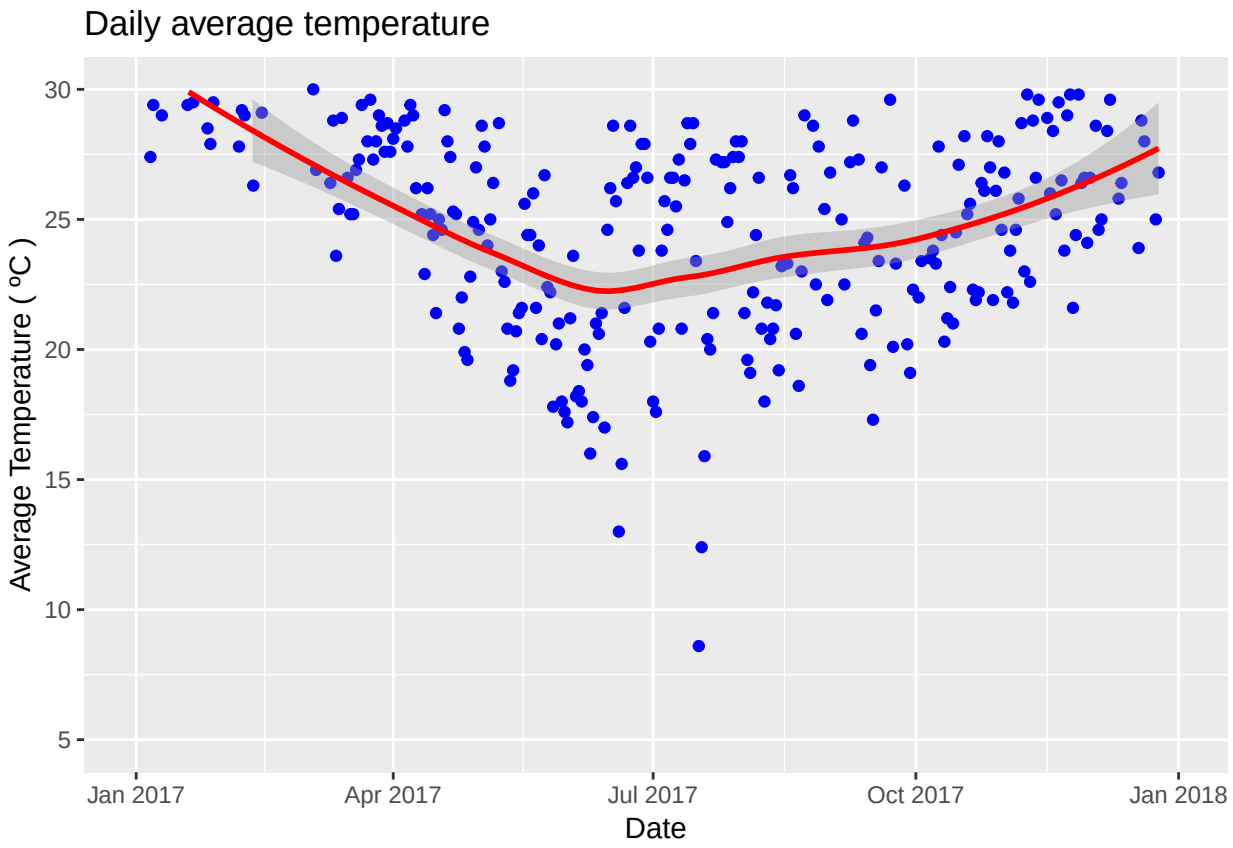
```
df %>%
  filter(Date >= '2017-01-01') %>%
  filter(Date <= '2017-12-31') %>%
  filter(Hour == '12:00') %>%
  ggplot() +
  geom_line(aes(x=Date, y = Preciptation)) +
  ggtitle ("Preciptation in 2017") +
  xlab("Months") +  ylab ("Preciptation (mm)")
```




```
#df %>%
#  filter(Date >= '2017-01-01') %>%
#  filter(Date <= '2017-12-31') %>%
#  filter(Hour == '12:00') %>%
#  group_by(Month) %>%
#  summarise(MeanPreciptationByMonth = mean(Preciptation, na.rm=TRUE))
#  ggplot() +
#  geom_boxplot(aes(x=Month, y = MeanPreciptationByMonth)) +
#  ggtitle ("Preciptation in 2017") +
#  xlab("Months") +  ylab ("Preciptation (mm)")
```

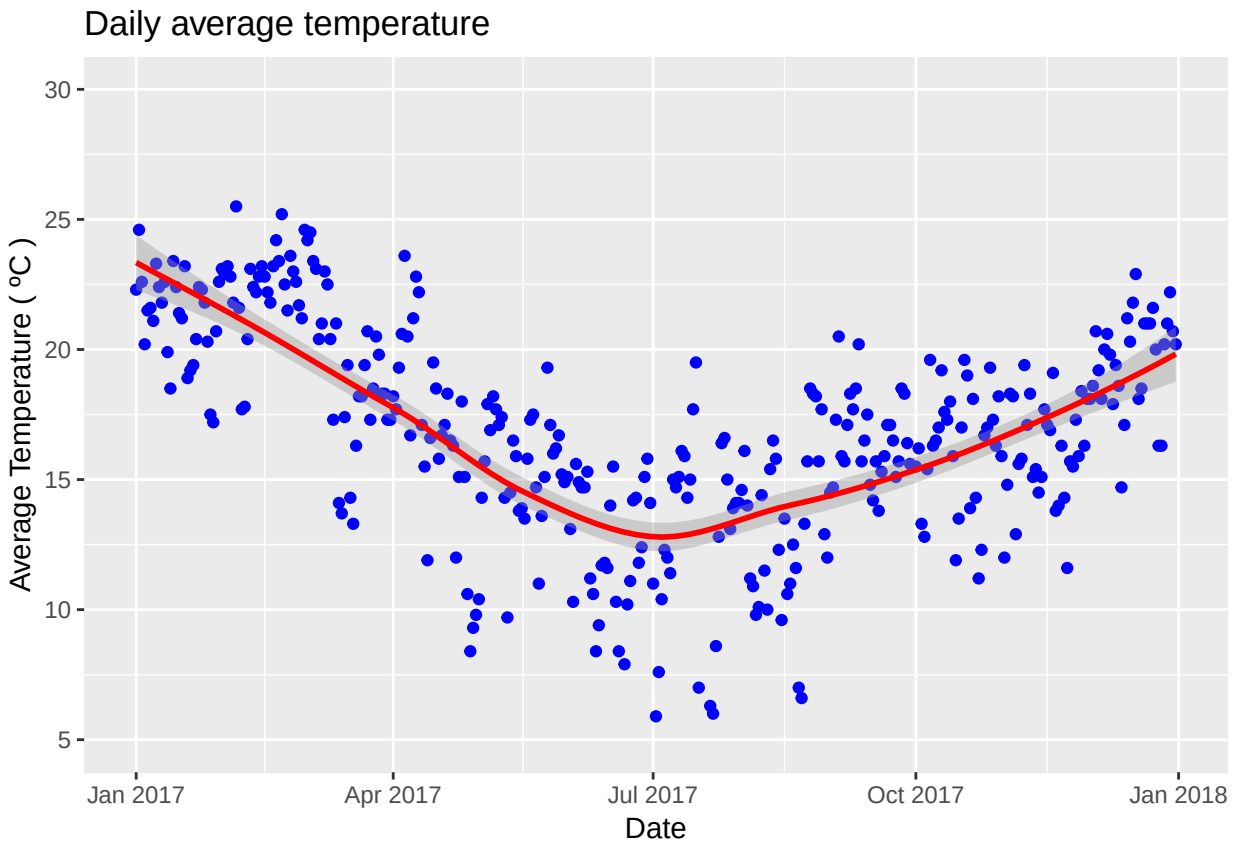
```
df %>%
  filter(Date >= '2017-01-01') %>%
  filter(Date <= '2017-12-31') %>%
  filter(Hour == '00:00') %>%
  ggplot(aes(x=Date, y=MaxTemperature)) +
  geom_point(colour = "blue") +
  geom_smooth(colour = "red",size = 1) +
  scale_y_continuous(limits = c(5,30), breaks = seq(5,30,5)) +
  ggtitle ("Daily average temperature") +
  xlab("Date") +  ylab ("Average Temperature ( ºC )")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 96 rows containing non-finite values (stat_smooth).

## Warning: Removed 96 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_smooth).
```



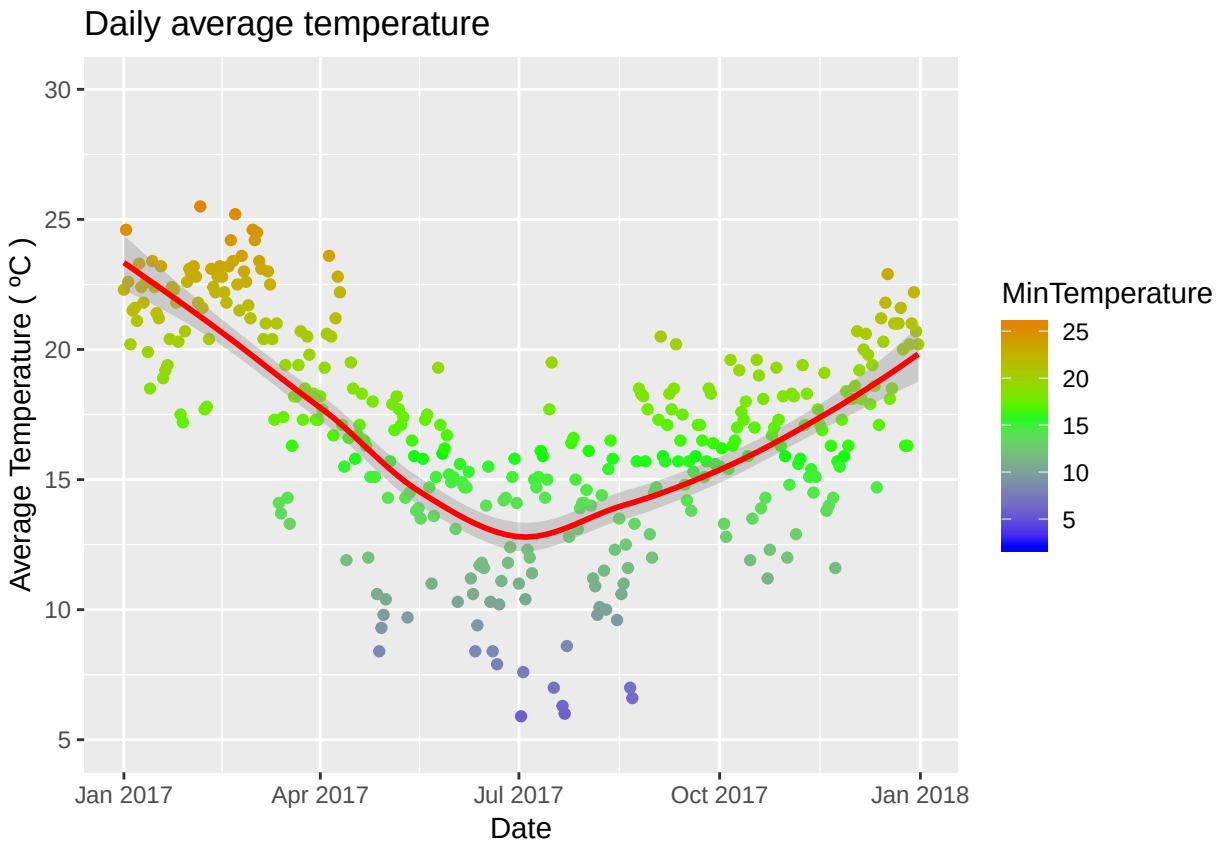

```
df %>%
  filter(Date >= '2017-01-01') %>%
  filter(Date <= '2017-12-31') %>%
  filter(Hour == '12:00') %>%
  ggplot(aes(x=Date, y=MinTemperature)) +
  geom_point(colour = "blue") +
  geom_smooth(colour = "red",size = 1) +
  scale_y_continuous(limits = c(5,30), breaks = seq(5,30,5)) +
  ggtitle ("Daily average temperature") +
  xlab("Date") +  ylab ("Average Temperature ( ºC )")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## Warning: Removed 4 rows containing non-finite values (stat_smooth).
## Warning: Removed 4 rows containing missing values (geom_point).
```




```
df %>%
  filter(Date >= '2017-01-01') %>%
  filter(Date <= '2017-12-31') %>%
  filter(Hour == '12:00') %>%
  ggplot(aes(x=Date, y=MinTemperature)) +
  geom_point(aes(colour = MinTemperature)) +
  scale_colour_gradient2(low = "blue", mid = "green" , high = "red", midpoint = 16) +
  geom_smooth(color = "red",size = 1) +
  scale_y_continuous(limits = c(5,30), breaks = seq(5,30,5)) +
  ggtitle ("Daily average temperature") +
  xlab("Date") +  ylab ("Average Temperature ( ºC )")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## Warning: Removed 4 rows containing non-finite values (stat_smooth).
## Warning: Removed 4 rows containing missing values (geom_point).
```

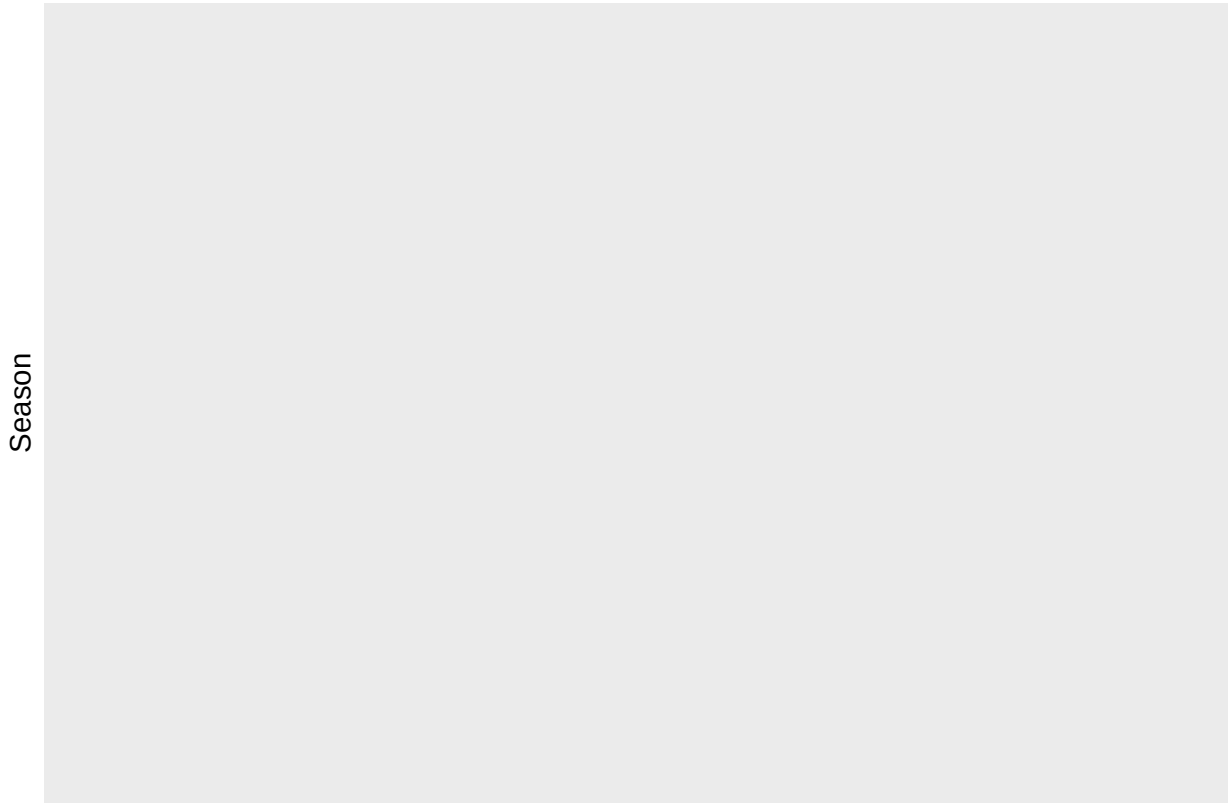



```
#scales
#mins <- min(df$MinTemperature)
#maxs <- max(df$MaxTemperature)


#df %>%
#  filter(Date >= '2017-01-01') %>%
#  filter(Date <= '2017-12-31') %>%
#  filter(Hour == '12:00') %>%
#  mutate(Month = month(Date)) %>%
#  group_by(Month) %>%
#  ## black and white
##  ggplot(aes(x = MeanTemperature ,y = Month) )+
#  geom_joy(scale=3) +
#  scale_x_continuous(limits = c(mins,maxs)) #+
#  theme_ipsum(grid=F)+
#  labs(title='Temperatures in Pittsburgh',
#       subtitle='Median temperatures (Fahrenheit) by month for 2016\nData: Original CSV from the Weath

## in color
#ggplot(pgh_weather, aes(x = `Mean.TemperatureF`, y = `months`, fill = ..x..)) +
#  geom_density_ridges_gradient(scale = 3, rel_min_height = 0.01, gradient_lwd = 1.) +
#  scale_x_continuous(expand = c(0.01, 0)) +
#  scale_y_discrete(expand = c(0.01, 0)) +
#  scale_fill_viridis(name = "Temp. [ºF]", option = "C") +
#  labs(title = 'Temperatures in Pittsburgh',
#       subtitle = 'Mean temperatures (Fahrenheit) by month for 2016\nData: Original CSV from the Weath
```

```
#       x = "Mean Temperature") +
#  theme_ridges(font_size = 13, grid = TRUE) + theme(axis.title.y = element_blank())
```

```
#scales
#mins <- min(df$MinTemperature)
#maxs <- max(df$MaxTemperature)


#df %>%
#  #filter(Date >= '2017-01-01') %>%
  #filter(Date <= '2017-12-31') %>%
  #filter(Hour == '12:00') %>%
  #mutate(Month = month(Date)) %>%
  #ggplot(aes(x = Month, y=Preciptation, group = Month)) +
  #geom_density_ridges(scale = 10, size = 0.25, rel_min_height = 0.03) +
  #theme_ridges() +
  #scale_x_continuous(limits=c(1, 200), expand = c(0.01, 0)) +
  #scale_y_reverse(breaks=c(2000, 1980, 1960, 1940, 1920, 1900), expand = c(0.01, 0))
```

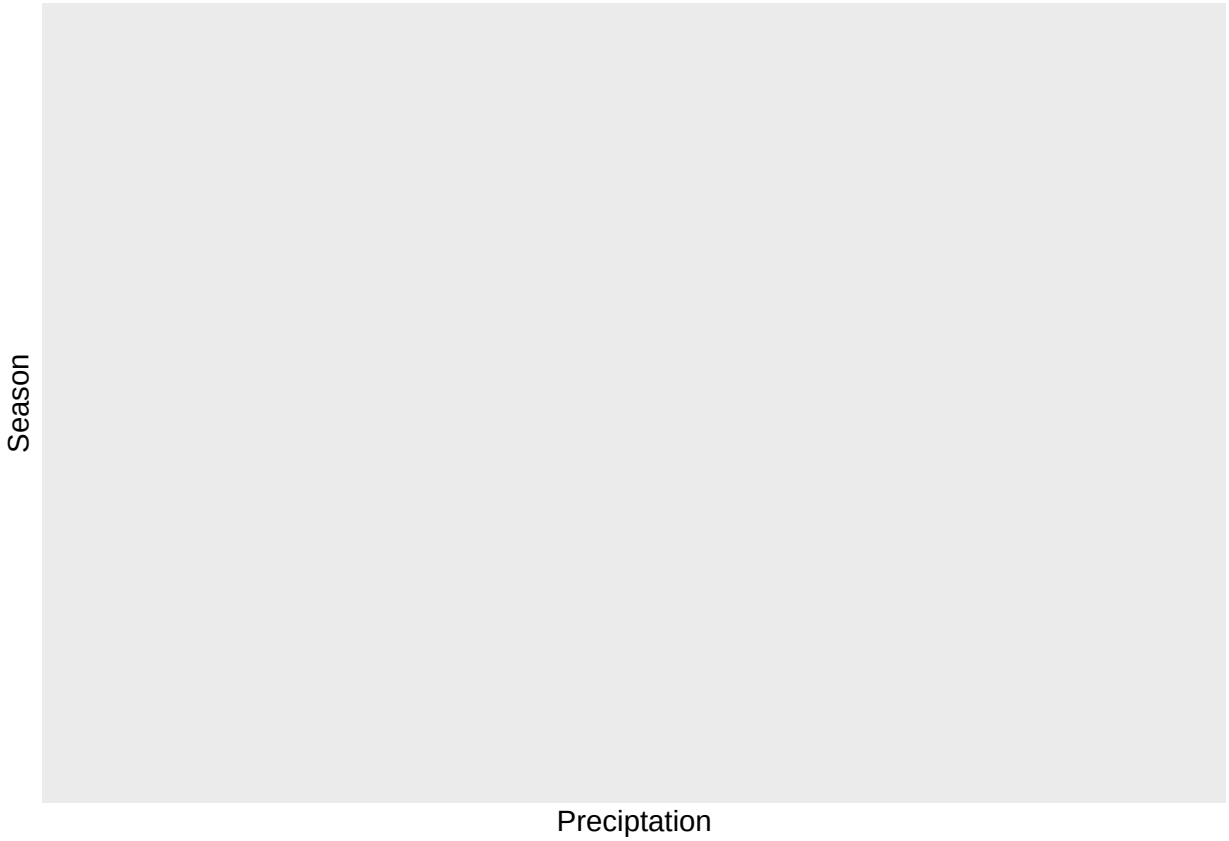
```
dfmerged %>%
  filter(Date >= '2017-01-01') %>%
  filter(Date <= '2017-12-31') %>%
  filter(Preciptation != 'NA') %>%
  ggplot(aes(x=Preciptation, y=Season)) + geom_density_ridges2()
```




```
dfmerged %>%
  filter(Date >= '2016-01-01') %>%
```

```
  filter(Date <= '2016-12-31') %>%
  #filter(Preciptation != 'NA') %>%
  #filter(Preciptation >= mean(Preciptation)) %>%
  ggplot(aes(x=Preciptation, y=Season)) + geom_density_ridges2()
```


```
dfmerged %>%
  filter(Date >= '2016-01-01') %>%
  filter(Date <= '2016-12-31') %>%
  #filter(Preciptation != 'NA') %>%
  filter(Preciptation >= mean(Preciptation)) %>%
  ggplot(aes(x=Preciptation, y=Season)) + geom_density_ridges2()
```

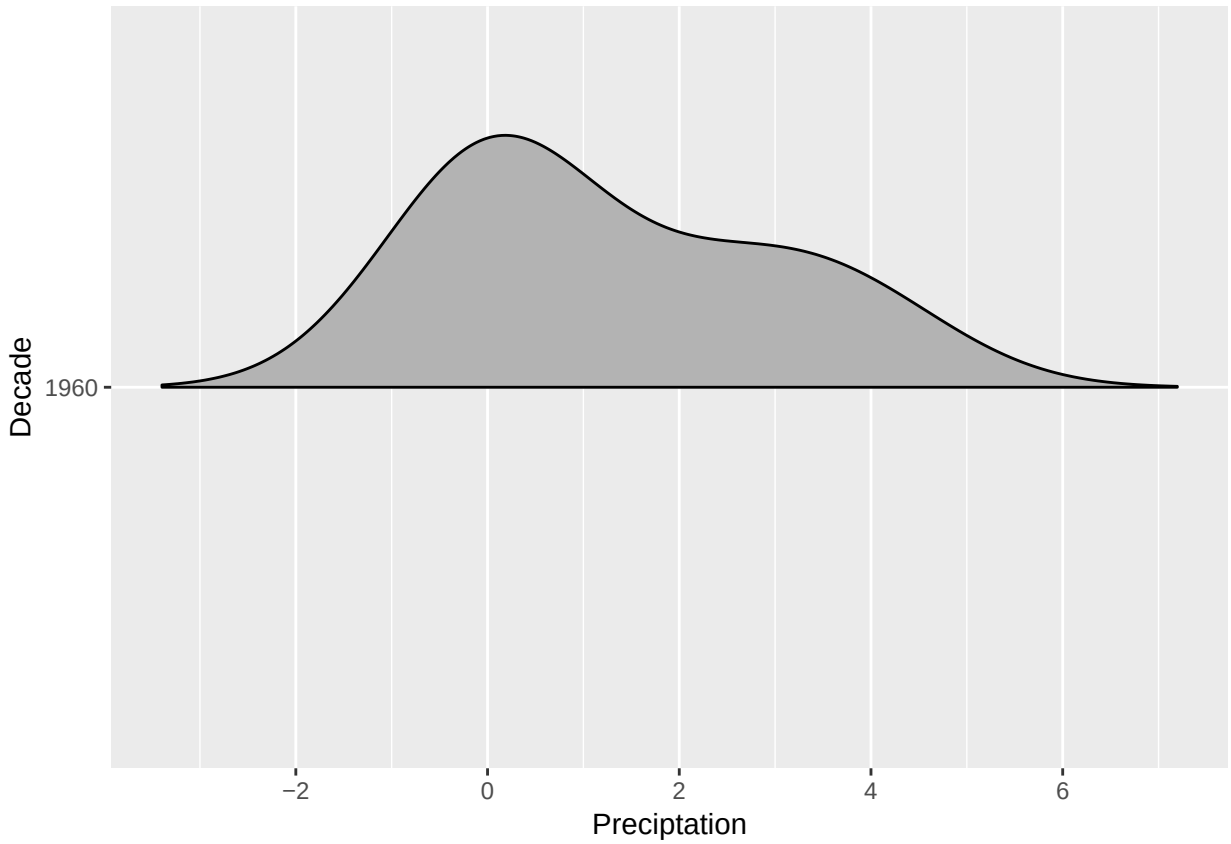



```
dfmerged %>%
  filter(Date >= '2016-01-01') %>%
  filter(Date <= '2016-12-31') %>%
  #filter(Preciptation != 'NA') %>%
  filter(Preciptation < mean(Preciptation)) %>%
  ggplot(aes(x=Preciptation, y=Season)) + geom_density_ridges2()
```

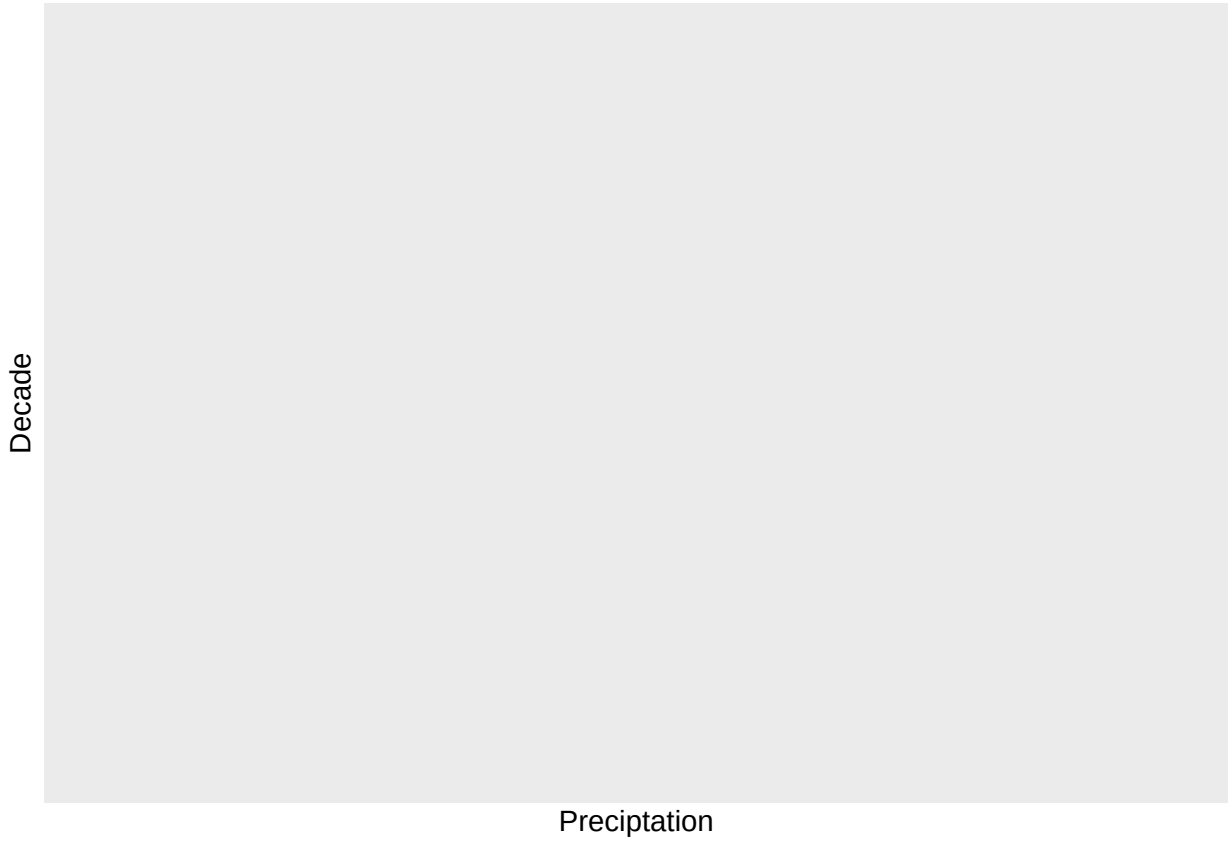

```
dfmerged %>%
  #filter(Decade == '1990') %>%
  #filter(Date <= '2016-12-31') %>%
  #filter(Preciptation != 'NA') %>%
  #filter(Preciptation >= mean(Preciptation)) %>%
  ggplot(aes(x=Preciptation, y=Decade)) + geom_density_ridges2()
```

```
## Picking joint bandwidth of 1.13
```

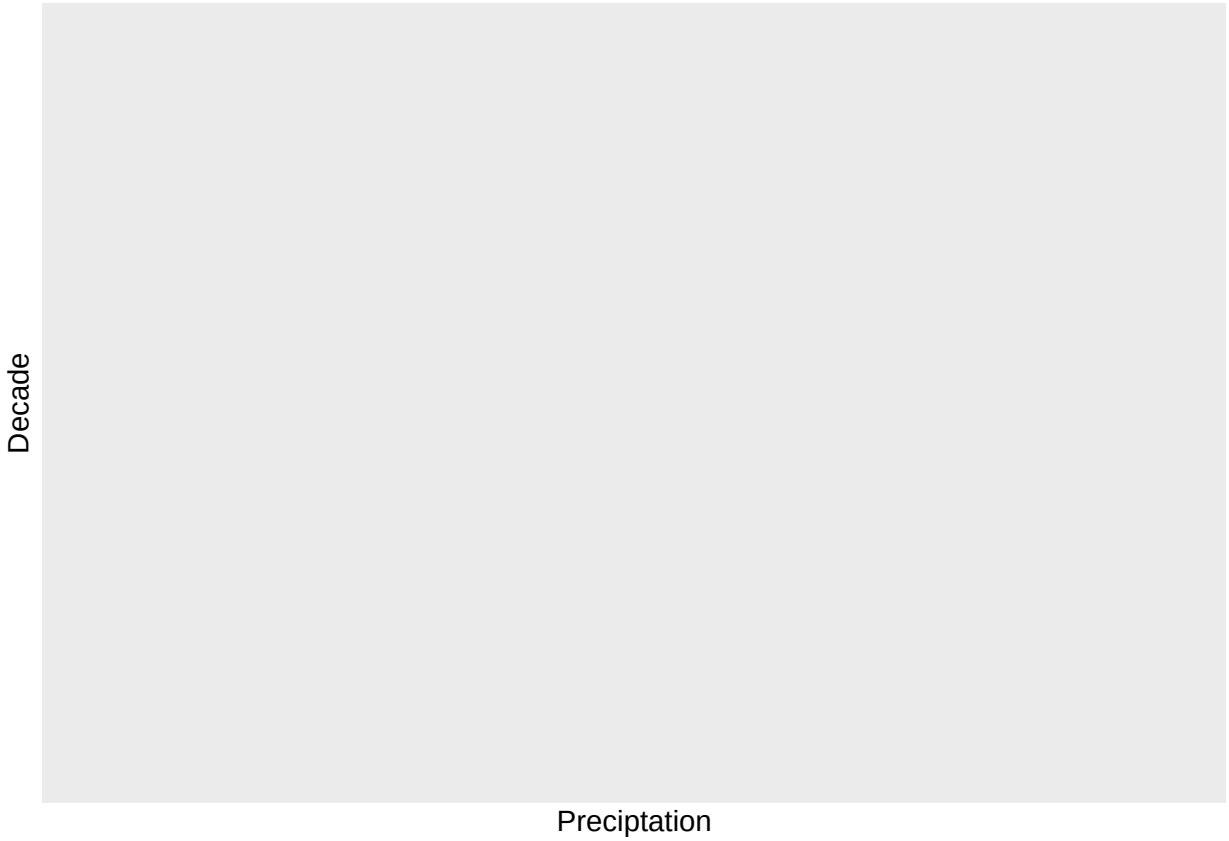
```
## Warning: Removed 1 rows containing non-finite values (stat_density_ridges).
```




```
dfmerged %>%
  #filter(Decade == '1990') %>%
  #filter(Date <= '2016-12-31') %>%
  #filter(Preciptation != 'NA') %>%
  filter(Preciptation >= 10) %>%
  ggplot(aes(x=Preciptation, y=Decade)) + geom_density_ridges2()
```




```
dfmerged %>%
  #filter(Decade == '1990') %>%
  #filter(Date <= '2016-12-31') %>%
  #filter(Preciptation != 'NA') %>%
  filter(Preciptation >= 20) %>%
  ggplot(aes(x=Preciptation, y=Decade)) + geom_density_ridges2()
```



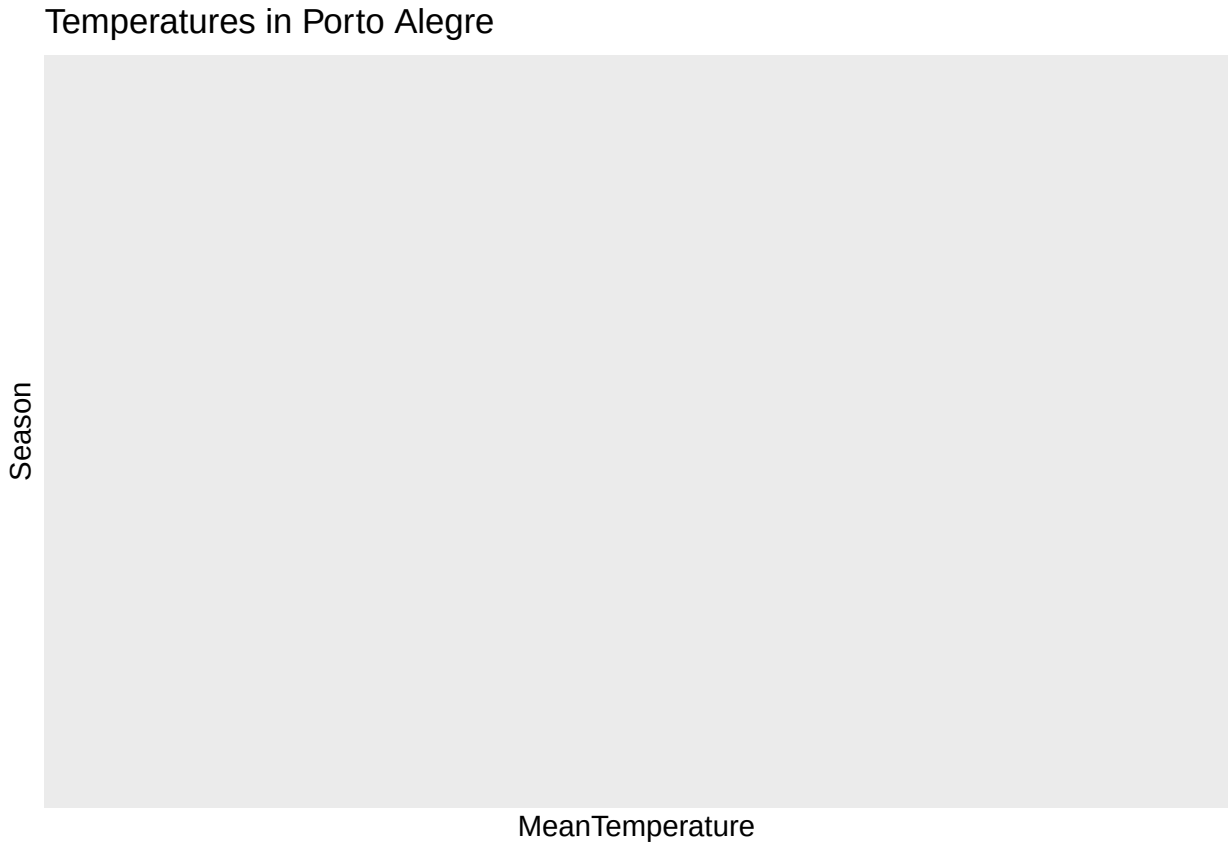

```
dfmerged %>%
  #filter(Decade == '1990') %>%
  #filter(Date <= '2016-12-31') %>%
  #filter(Preciptation != 'NA') %>%
  filter(Preciptation >= 10) %>%
  ggplot(aes(x=Preciptation, y=Year, group=Year)) +
  geom_density_ridges(scale=10, size=1, rel_min_height = 0.01, fill="lightblue") +
  theme_ridges() #+
```

Year

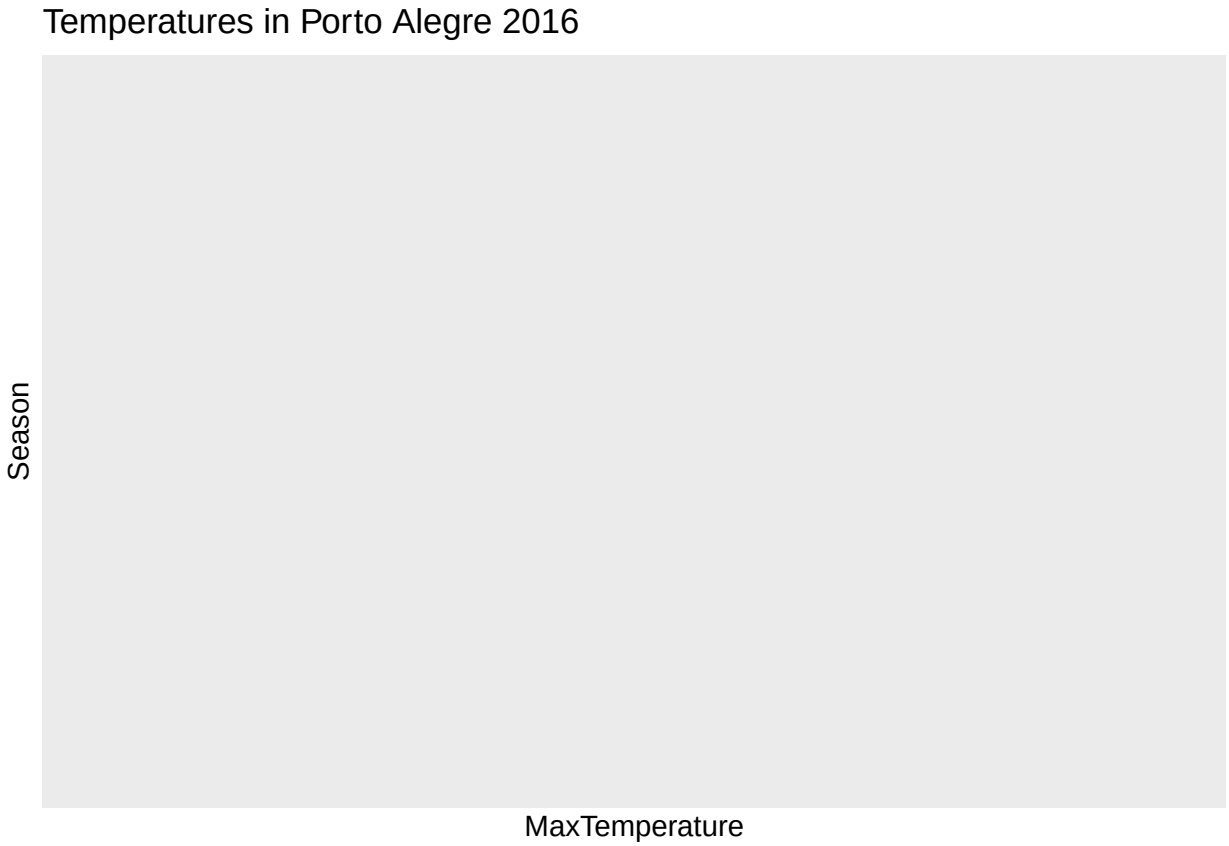
Preciptation

```
  #scale_x_discrete(limits=c(1, 200), expand = c(0.01, 0)) +
  #scale_y_reverse(breaks=c(2010, 2000, 1990, 1980, 1970, 1960), expand = c(0.01, 0))
```

```
dfmerged %>%
  filter(Year == '2017') %>%
  ggplot() +
  geom_density_ridges_gradient(aes(x=MeanTemperature, y=Season, fill= ..x..), scale = 3, rel_min_height
  scale_fill_viridis(name = "Temp. [C]", option = "C") +
  labs(title = 'Temperatures in Porto Alegre')
```

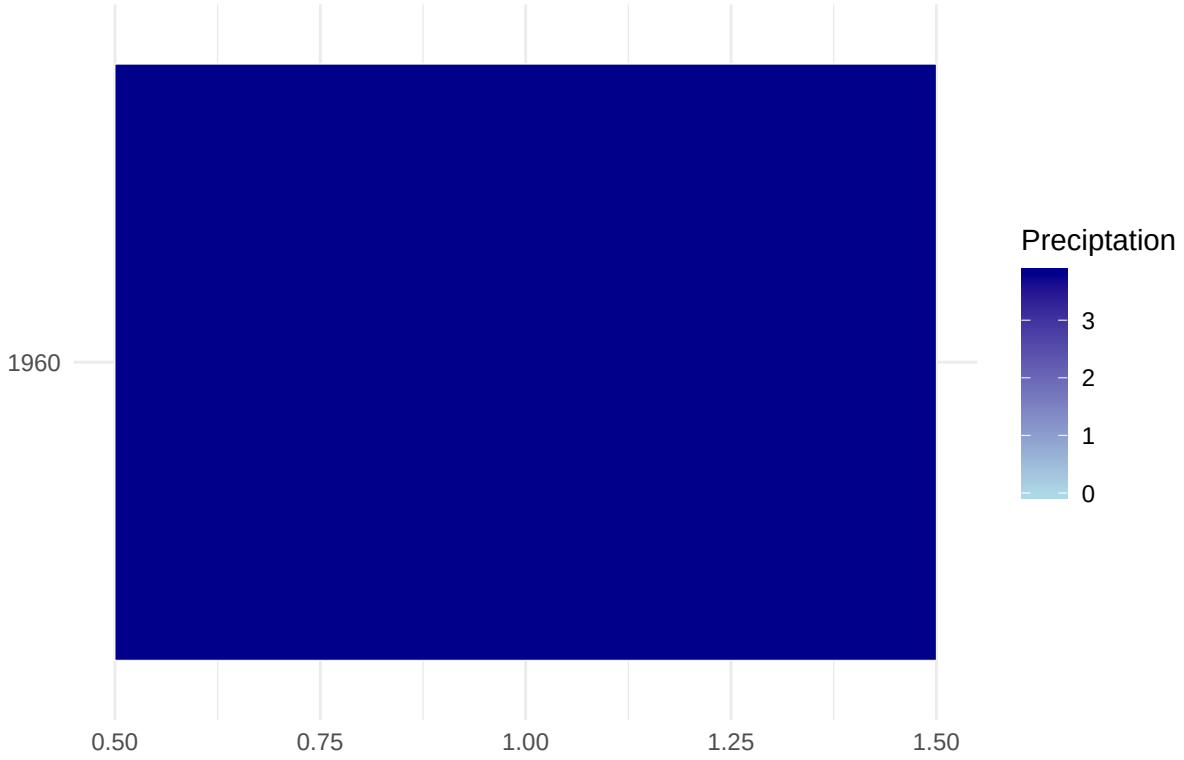



```
dfmerged %>%
  filter(Year == '2016') %>%
  #filter(Date <= '2016-12-31') %>%
  #filter(Preciptation != 'NA') %>%
  ggplot(aes(x=MaxTemperature, y=Season)) +
  geom_density_ridges_gradient(scale = 3, rel_min_height = 0.01) +
  scale_fill_viridis(name = "Temp. [F]", option = "C") +
  labs(title = 'Temperatures in Porto Alegre 2016')
```



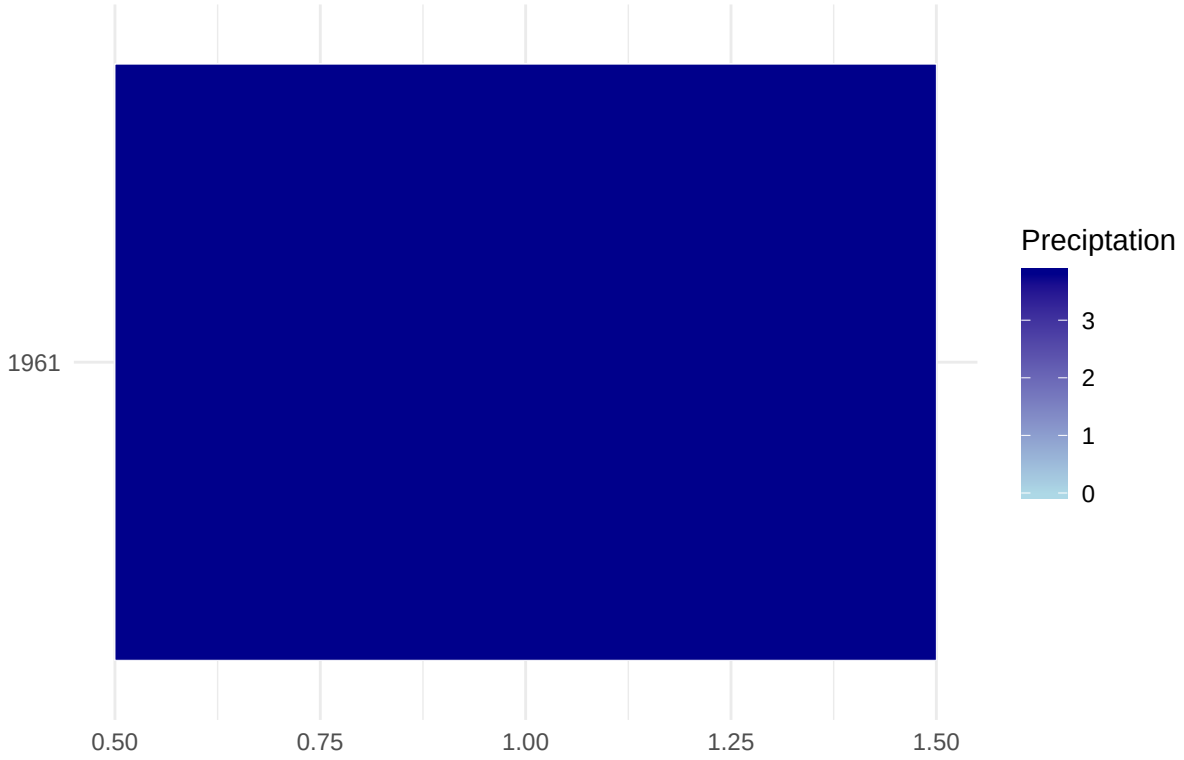
Fazendo um heatmap bem simples

```
dfmerged %>%
  ggplot(aes(Month,Decade)) +
  geom_tile(aes(fill = Preciptation),
            colour = "white",
            na.rm = TRUE) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  #guides(fill=guide_legend(title="Total Incidents")) +
  theme_bw() +
  theme_minimal() +
  labs(title = "",
       x = "",
       y = "") #+
```



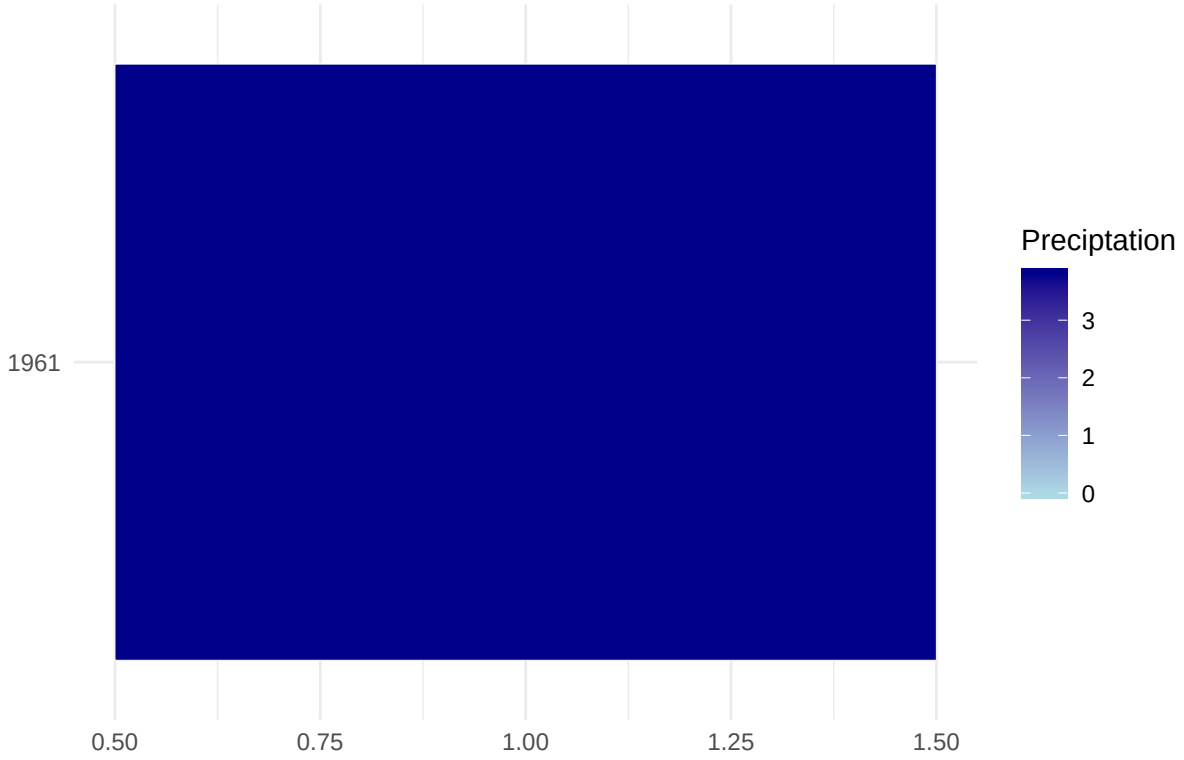

```
  #theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```

```
dfmerged %>%
  ggplot(aes(Month,Year)) +
  geom_tile(aes(fill = Preciptation),
            colour = "white",
            na.rm = TRUE) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  #guides(fill=guide_legend(title="Total Incidents")) +
  theme_bw() +
  theme_minimal() +
  labs(title = "",
       x = "",
       y = "") #+
```



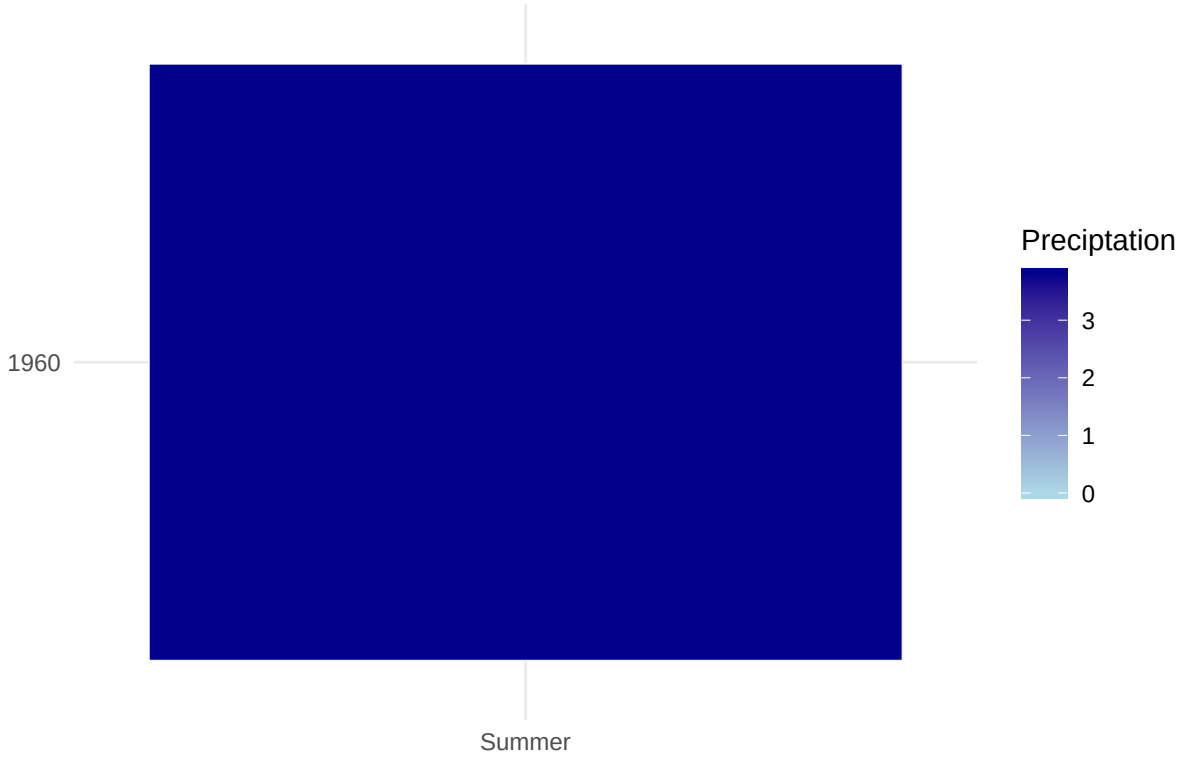

```
  #theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```

```
dfmerged %>%
  ggplot(aes(Month,Year)) +
  geom_tile(aes(fill = Preciptation),
            colour = "white",
            na.rm = TRUE) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  #guides(fill=guide_legend(title="Total Incidents")) +
  theme_bw() +
  theme_minimal() +
  labs(title = "",
       x = "",
       y = "") #+
```




```
  #theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```

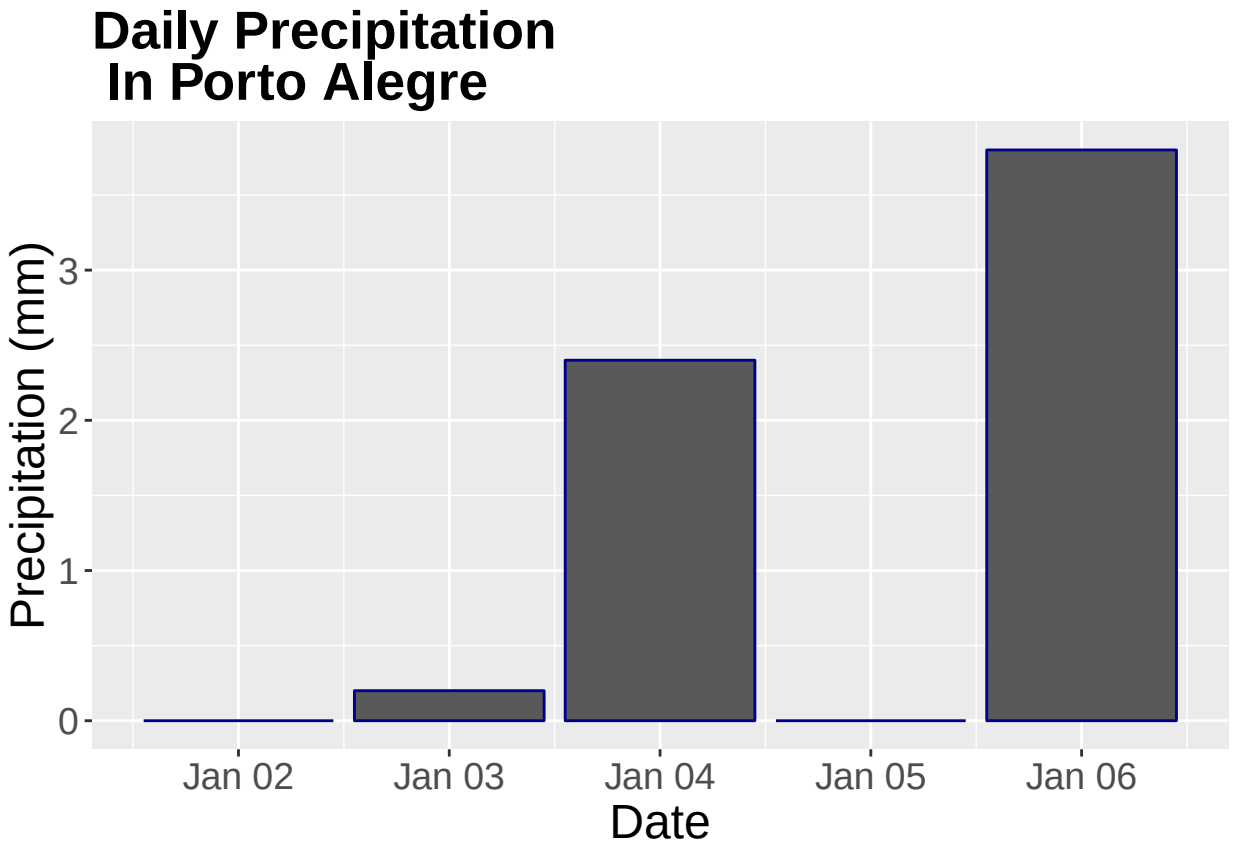
```
dfmerged %>%
  ggplot(aes(Season,Decade)) +
  geom_tile(aes(fill = Preciptation),
            colour = "white",
            na.rm = TRUE) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  #guides(fill=guide_legend(title="Total Incidents")) +
  theme_bw() +
  theme_minimal() +
  labs(title = "",
       x = "",
       y = "") #+
```




```
  #theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```

```
dfmerged %>%
  ggplot(aes(Date, Preciptation)) +
    geom_bar(stat="identity", na.rm = TRUE, color="darkblue") +
    ggtitle("Daily Precipitation\n In Porto Alegre") +
    xlab("Date") + ylab("Precipitation (mm)") +
    #scale_x_date(labels=date_format ("%b %y"), breaks=date_breaks("1 year")) +
    theme(plot.title = element_text(lineheight=.8, face="bold", size = 20)) +
    theme(text = element_text(size=18))
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```




```
    #+stat_smooth(colour="red")
```