

Web_scraping

2025-04-03

Section A

in this section firstly we try to Retrieve the data from the summary table „Forecast for the next 48 hours“ and summarize its information. Firstly we read the whole of HTML file named”Weather for Sari, Iran.html”. The city I have choosen is Sari, a city in Iran. In the code below I have find the table with html_elements and looking for a text with the value of “Forecast for the next 48 hours” “. This section has lots of table but the one we are looking for is the first one so I used position() function and make it equal to 1. So the main table is assigned to main table and can be seen below. This table consists of the 2 days, Monday and Tuesday, and the times of the day.

```
library("rvest")

main_html = read_html("Weather for Sari, Iran.html")

selected_section <- main_html %>%
  html_elements(xpath = '//*[@text() = "Forecast for the next 48 hours"]') %>%
  html_elements(xpath = '..')

main_table <- selected_section %>%
  html_elements(xpath = '//*[@table[position() = 1]')

information = main_table %>% html_table()
print(information)

























## [[1]]
## # A tibble: 13 x 8
##   ` `      Wednesday Wednesday Thursday Thursday Thursday Thursday Friday
##   <chr>    <chr>      <chr>      <chr>      <chr>      <chr>      <chr>      <chr>
## 1 ""      "Evening" "Night"    "Mornin~ "Aftern~ "Evenin~ "Night"    "Morn~
## 2 "Forecast" ""         ""         ""         ""         ""         ""         ""
## 3 "Temperature" "12 °C"  "10 °C"   "11 °C"   "13 °C"   "11 °C"   "8 °C"    "12 °~
## 4 ""      "Rain sh~ "Rain sh~ "Rain s~ "Rain s~ "Rain s~ "Mostly~ "Over~
## 5 "Feels Like" "10 °C"  "8 °C"    "8 °C"    "11 °C"   "9 °C"    "6 °C"    "12 °~
## 6 "Wind Speed" "15 km/h" "22 km/h" "30 km/~ "14 km/~ "12 km/~ "10 km/~ "6 km~
## 7 "Wind Directi~ "WSW↑"   "WSW↑"   "WSW↑"   "W↑"      "S↑"      "S↑"      "S↑"
## 8 "Humidity"    "93%"    "94%"     "93%"     "85%"     "88%"     "88%"     "79%"
## 9 "Dew Point"   "11 °C"  "9 °C"    "10 °C"   "10 °C"   "9 °C"    "6 °C"    "8 °C"
## 10 "Visibility"  "6 km"   "8 km"    "6 km"    "8 km"    "5 km"    "14 km"   "11 k~
## 11 "Probability ~ "24%"    "17%"     "28%"     "23%"     "16%"     "0%"      "0%"
## 12 "Amount of Ra~ "3.9 mm" "3.0 mm"  "5.8 m~ "4.4 m~ "1.5 m~ "-"       "-"
## 13 "* Updated We~ "* Updat~ "* Updat~ "* Upda~ "* Upda~ "* Upda~ "* Upda~ "* Up~
```

Section B

In this section I visualize the predicted development of temperature, wind, and humidity for the next 24 hours (retrieve the data from „Detailed Hourly Forecast - Next 24 hours“), firstly I take the html file

named “Hourly forecast for Sari, Iran.html”. and get the related rows. I used lapply and rbind to combine the rows together. Below you can see all of the information in related to wether in 24 hours in this city.

Show weather on:
Next 24 hours

Time	Conditions			Comfort			Precipitation	
	Temp	Weather		Feels Like	Wind	Humidity	Chance	Amount
20:00 Wed, 16 Apr	12 °C	 Light showers. Overcast.		10 °C	15 km/h ↗	93%	24%	0.7 mm (rain)
21:00	11 °C	 Light showers. Overcast.		10 °C	15 km/h ↗	94%	26%	0.8 mm (rain)
22:00	11 °C	 Light showers. Overcast.		9 °C	18 km/h ↗	94%	25%	0.8 mm (rain)
23:00	11 °C	 A few showers. Overcast.		8 °C	21 km/h ↗	94%	23%	0.7 mm (rain)
00:00 Thu, 17 Apr	11 °C	 A few showers. Overcast.		8 °C	24 km/h ↗	93%	22%	0.6 mm (rain)
01:00	10 °C	 A few showers. Overcast.		8 °C	23 km/h ↗	94%	20%	0.5 mm (rain)
02:00	10 °C	 A few showers. Overcast.		8 °C	22 km/h ↗	94%	17%	0.4 mm (rain)
03:00	10 °C	 Drizzle. Overcast.		8 °C	20 km/h ↗	94%	17%	0.4 mm (rain)
04:00	10 °C	 A few showers. Overcast.		7 °C	25 km/h ↗	94%	19%	0.5 mm (rain)
05:00	10 °C	 A few showers. Overcast.		7 °C	29 km/h ↗	96%	22%	0.6 mm (rain)
06:00	10 °C	 Light showers. Overcast.		7 °C	33 km/h ↗	96%	25%	0.8 mm (rain)
07:00	10 °C	 Light showers. Overcast.		7 °C	31 km/h ↗	95%	26%	0.9 mm (rain)
08:00	11 °C	 Light showers. Overcast.		8 °C	30 km/h ↗	93%	28%	1.0 mm (rain)
09:00	11 °C	 Light showers. Overcast.		8 °C	30 km/h →	92%	28%	1.1 mm (rain)
10:00	12 °C	 Light showers. Overcast.		9 °C	26 km/h →	91%	28%	1.1 mm (rain)
11:00	12 °C	 Light showers. Overcast.		10 °C	22 km/h →	88%	28%	1.0 mm (rain)
12:00	12 °C	 Light showers. Overcast.		10 °C	19 km/h →	87%	28%	1.0 mm (rain)
13:00	12 °C	 Light showers. Overcast.		11 °C	16 km/h →	85%	26%	0.9 mm (rain)
14:00	13 °C	 A few showers. Overcast.		11 °C	14 km/h →	85%	23%	0.7 mm (rain)
15:00	13 °C	 A few showers. Overcast.		12 °C	12 km/h →	85%	21%	0.6 mm (rain)
16:00	12 °C	 A few showers. Overcast.		11 °C	10 km/h ↗	86%	22%	0.6 mm (rain)
17:00	12 °C	 A few showers. Overcast.		10 °C	11 km/h ↗	88%	23%	0.7 mm (rain)
18:00	11 °C	 A few showers. Overcast.		9 °C	13 km/h ↗	89%	23%	0.7 mm (rain)
19:00	10 °C	 A few showers. Overcast.		9 °C	12 km/h ↑	90%	20%	0.5 mm (rain)

* Updated Wednesday, 16 April 2025 17:12:55 Sari time - Weather by CustomWeather, © 2025

```

hourly_data_html <- read_html('Hourly forecast for Sari, Iran.html')
selected_section_hourly = hourly_data_html %>%
  html_elements(xpath = '//*[@id= "wt-hbh-title"]') %>% html_elements(xpath = '..')

main_hourly_rows <- selected_section_hourly %>%
  html_elements(xpath = '//*[@position() = 1]//tr[position() > 1]')

table <- main_hourly_rows %>%

```

```

lapply(function(row) {
  row %>% html_elements("td") %>% html_text()
}) %>% do.call(rbind, .) %>% as.data.frame()

colnames(table) <- c( "Conditions" , "Temp" ,      "Weather",  "Feels" , "Like" ,  "Wind" ,
                      "Humidity" ,  "Chance" , "Amount")

print(table)

```

	Conditions	Temp	Weather	Feels	Like	Wind	Humidity	Chance
## 1		11 °C	Light showers. Overcast.	10 °C	15 km/h	↑	94%	26%
## 2		11 °C	Light showers. Overcast.	9 °C	18 km/h	↑	94%	25%
## 3		11 °C	A few showers. Overcast.	8 °C	21 km/h	↑	94%	23%
## 4		11 °C	A few showers. Overcast.	8 °C	24 km/h	↑	93%	22%
## 5		10 °C	A few showers. Overcast.	8 °C	23 km/h	↑	94%	20%
## 6		10 °C	A few showers. Overcast.	8 °C	22 km/h	↑	94%	17%
## 7		10 °C	Drizzle. Overcast.	8 °C	20 km/h	↑	94%	17%
## 8		10 °C	A few showers. Overcast.	7 °C	25 km/h	↑	94%	19%
## 9		10 °C	A few showers. Overcast.	7 °C	29 km/h	↑	96%	22%
## 10		10 °C	Light showers. Overcast.	7 °C	33 km/h	↑	96%	25%
## 11		10 °C	Light showers. Overcast.	7 °C	31 km/h	↑	95%	26%
## 12		11 °C	Light showers. Overcast.	8 °C	30 km/h	↑	93%	28%
## 13		11 °C	Light showers. Overcast.	8 °C	30 km/h	↑	92%	28%
## 14		12 °C	Light showers. Overcast.	9 °C	26 km/h	↑	91%	28%
## 15		12 °C	Light showers. Overcast.	10 °C	22 km/h	↑	88%	28%
## 16		12 °C	Light showers. Overcast.	10 °C	19 km/h	↑	87%	28%
## 17		12 °C	Light showers. Overcast.	11 °C	16 km/h	↑	85%	26%
## 18		13 °C	A few showers. Overcast.	11 °C	14 km/h	↑	85%	23%
## 19		13 °C	A few showers. Overcast.	12 °C	12 km/h	↑	85%	21%
## 20		12 °C	A few showers. Overcast.	11 °C	10 km/h	↑	86%	22%
## 21		12 °C	A few showers. Overcast.	10 °C	11 km/h	↑	88%	23%
## 22		11 °C	A few showers. Overcast.	9 °C	13 km/h	↑	89%	23%
## 23		10 °C	A few showers. Overcast.	9 °C	12 km/h	↑	90%	20%
##	Amount							
## 1	0.8	mm	(rain)					
## 2	0.8	mm	(rain)					
## 3	0.7	mm	(rain)					
## 4	0.6	mm	(rain)					
## 5	0.5	mm	(rain)					
## 6	0.4	mm	(rain)					
## 7	0.4	mm	(rain)					
## 8	0.5	mm	(rain)					
## 9	0.6	mm	(rain)					
## 10	0.8	mm	(rain)					
## 11	0.9	mm	(rain)					
## 12	1.0	mm	(rain)					
## 13	1.1	mm	(rain)					
## 14	1.1	mm	(rain)					
## 15	1.0	mm	(rain)					
## 16	1.0	mm	(rain)					
## 17	0.9	mm	(rain)					
## 18	0.7	mm	(rain)					
## 19	0.6	mm	(rain)					
## 20	0.6	mm	(rain)					

```
## 21 0.7 mm (rain)
## 22 0.7 mm (rain)
## 23 0.5 mm (rain)
```

As you can see in the table above all of the nubers are charachter and they are not the pure number. For example the have some strings like : “°C”. we have to eliminate them and get the number. In order to this I used “gsub()” function which eliminate the other strings and return just the numbers. We have to use function “as.numeric()” to convert it to numeric. Now we can use this numbers to illustrate the scatter plot.

```
Humidity <- gsub("%", "", table[1 :23 , "Humidity"])
Humidity <- as.numeric(Humidity)
```

```
Temp <- gsub("[^0-9]", "", table[1 :23 , "Temp"])
Temp <- as.numeric(Temp)
```

```
Feels <- gsub("[^0-9]", "", table[1 :23 , "Feels"])
Feels <- as.numeric(Feels)
```

```
print(Humidity)
```

```
## [1] 94 94 94 93 94 94 94 94 96 96 95 93 92 91 88 87 85 85 85 86 88 89 90
```

```
print(Temp)
```

```
## [1] 11 11 11 11 10 10 10 10 10 10 10 11 11 12 12 12 12 13 13 12 12 11 10
```

```
print(Feels)
```

```
## [1] 10 9 8 8 8 8 8 7 7 7 7 8 8 9 10 10 11 11 12 11 10 9 9
```

Here we firstly use these numbers to a dataframe. and than use ggplot and geom_point() to draw them.

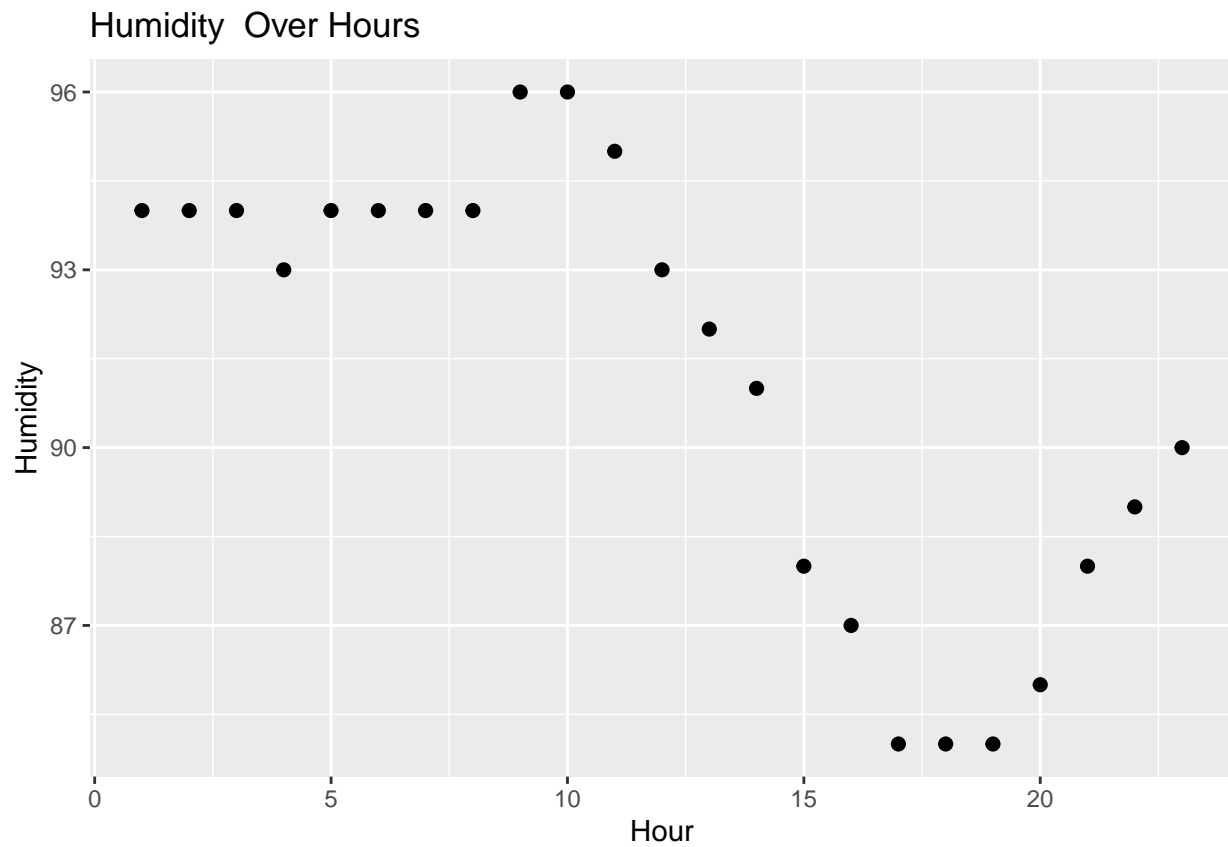
```
library(ggplot2)
```

```
Humidity_df <- data.frame(
  Hour = 1:23,
  Humidity_col = Humidity[1:23]
)
```

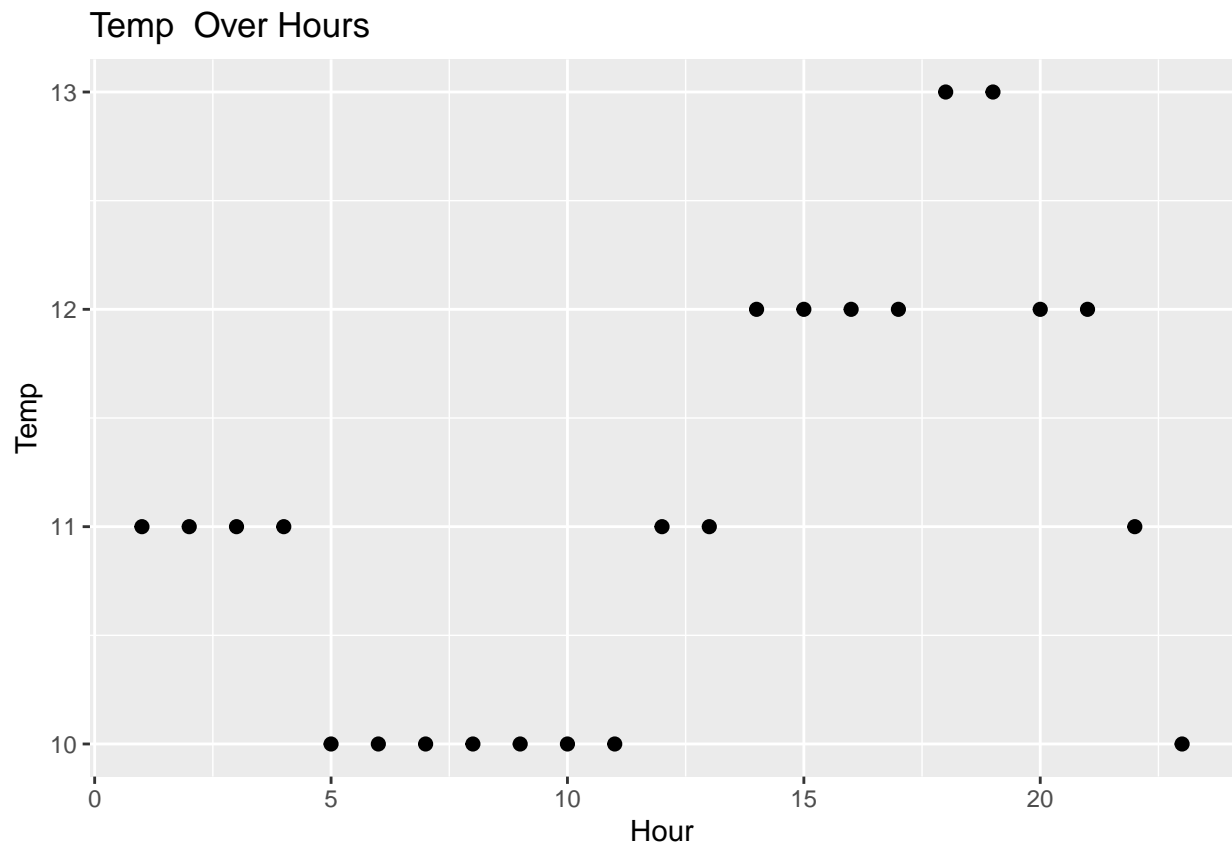
```
Temp_df <- data.frame(
  Hour = 1:23,
  Temp_col = Temp[1:23]
)
```

```
Feels_df <- data.frame(
  Hour = 1:23,
  Feels_col = Feels[1:23]
)
```

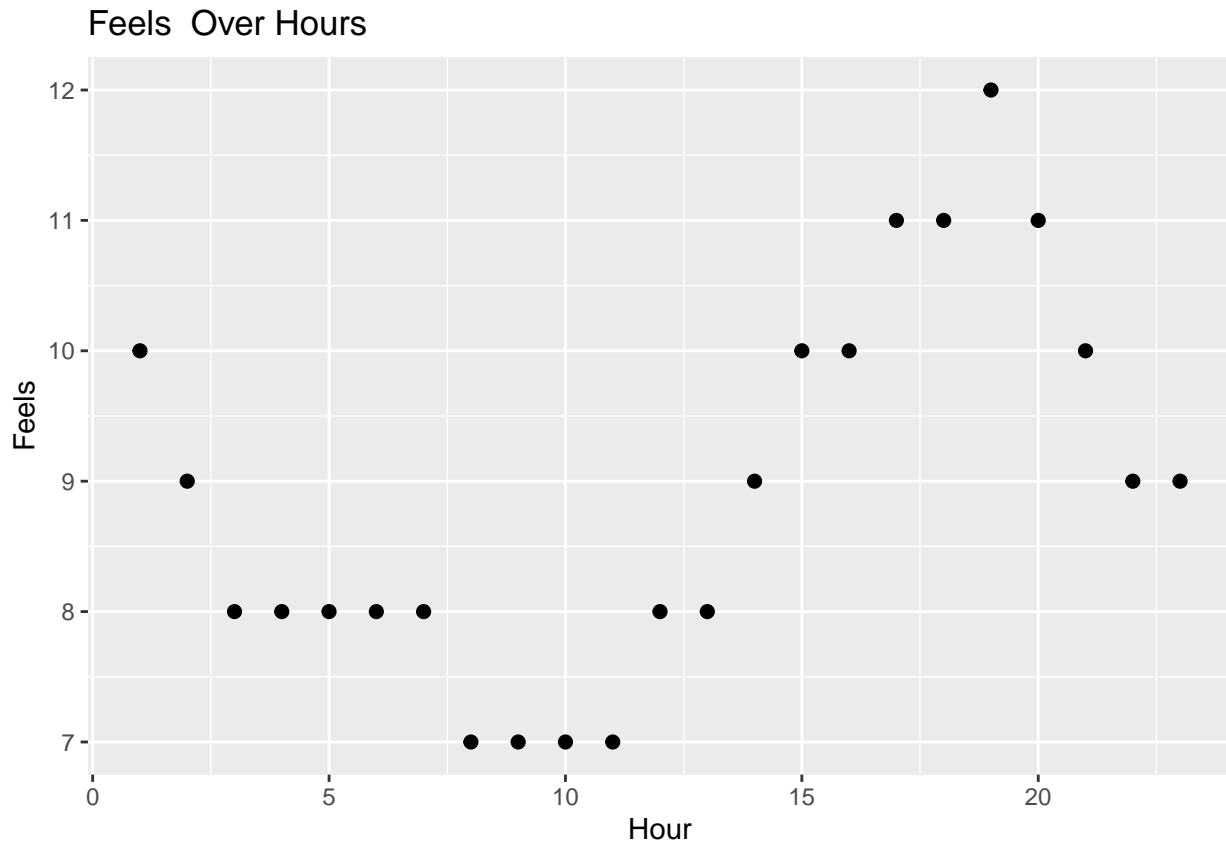
```
ggplot(Humidity_df, aes(x = Hour , y = Humidity_col)) +
  geom_point(size = 2) +
  labs(x = "Hour", y = "Humidity", title = "Humidity Over Hours")
```



```
ggplot(Temp_df, aes(x = Hour , y = Temp_col)) +  
  geom_point(size = 2) +  
  labs(x = "Hour", y = "Temp", title = "Temp Over Hours")
```



```
ggplot(Feels_df, aes(x = Hour , y = Feels_col)) + geom_point(size = 2) +  
  labs(x = "Hour", y = "Feels", title = "Feels Over Hours")
```



In the diagrams above, we have the “Feels of temperature, real temperature and over next 24 hours”. As it is shown the temperature and also its feeling is very low in the first hours of the day, however the percentage of humidity is high.

Section C

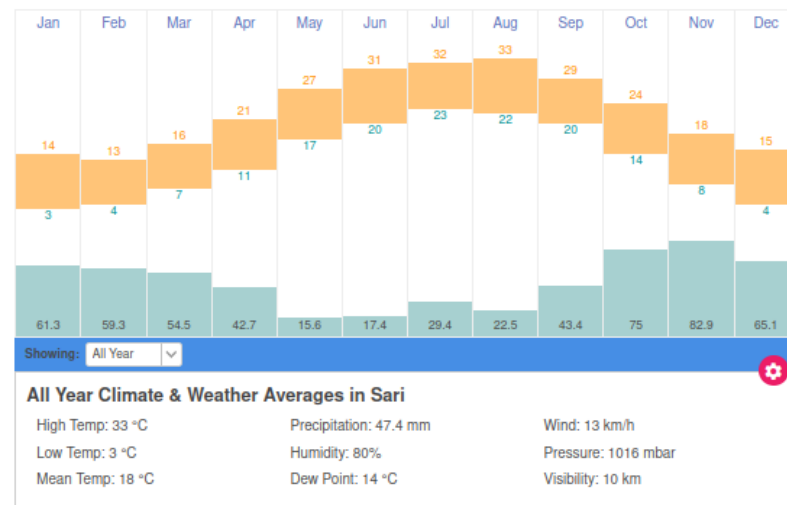
In this section we are going to check the avarage, highest, lowest temperature, Precipitation, Humidity, Wind, Pressure and the Visibility in whole of the year and also each month. So we will have 13 numbers for each of them. In the following code cell firstly we read the HTML file and use `html_element` function to access the table with Id of “climateTable”. This table contains the information related to the wether in the year.

Weather Today
Weather Hourly
14 Day Forecast
Yesterday/Past Weather
Climate (Averages)

Currently: 14 °C. Overcast. (Weather station: Babulsar, Iran). See more current weather >

Annual Weather Averages Near Sari

Averages are for Gharakhil, which is 25 kilometers from Sari.
Based on weather reports collected during 2012–2021.



Quick Climate Info

Hottest Month	August (27 °C avg)
Coldest Month	February (8 °C avg)
Wettest Month	November (82.9 mm avg)
Windiest Month	April (16 km/h avg)
Annual precip.	569.3 mm (per year)

Weather by CustomWeather, © 2025

Need some help?

```

annual_temp_avarage = read_html("Climate & Weather Averages in Sari, Iran.html")
information <- annual_temp_avarage %>%
  html_elements(xpath = '//*[@id= "climateTable"]//p') %>% html_text()
print(information)

```

```

## [1] "High Temp: 33 °C"      "Low Temp: 3 °C"
## [3] "Mean Temp: 18 °C"     "Precipitation: 47.4 mm"
## [5] "Humidity: 80%"        "Dew Point: 14 °C"
## [7] "Wind: 13 km/h"        "Pressure: 1016 mbar"
## [9] "Visibility: 10 km"     "High Temp: 14 °C"
## [11] "Low Temp: 3 °C"       "Mean Temp: 8 °C"

```

```

## [13] "Precipitation: 61.3 mm" "Humidity: 82%"
## [15] "Dew Point: 5 °C"      "Wind: 11 km/h"
## [17] "Pressure: 1020 mbar"   "Visibility: 11 km"
## [19] "High Temp: 13 °C"     "Low Temp: 4 °C"
## [21] "Mean Temp: 8 °C"      "Precipitation: 59.3 mm"
## [23] "Humidity: 84%"        "Dew Point: 5 °C"
## [25] "Wind: 13 km/h"        "Pressure: 1021 mbar"
## [27] "Visibility: 11 km"     "High Temp: 16 °C"
## [29] "Low Temp: 7 °C"       "Mean Temp: 12 °C"
## [31] "Precipitation: 54.5 mm" "Humidity: 82%"
## [33] "Dew Point: 8 °C"      "Wind: 16 km/h"
## [35] "Pressure: 1017 mbar"   "Visibility: 10 km"
## [37] "High Temp: 21 °C"     "Low Temp: 11 °C"
## [39] "Mean Temp: 16 °C"     "Precipitation: 42.7 mm"
## [41] "Humidity: 81%"        "Dew Point: 12 °C"
## [43] "Wind: 16 km/h"        "Pressure: 1016 mbar"
## [45] "Visibility: 10 km"     "High Temp: 27 °C"
## [47] "Low Temp: 17 °C"      "Mean Temp: 22 °C"
## [49] "Precipitation: 15.6 mm" "Humidity: 76%"
## [51] "Dew Point: 17 °C"     "Wind: 16 km/h"
## [53] "Pressure: 1012 mbar"   "Visibility: 10 km"
## [55] "High Temp: 31 °C"     "Low Temp: 20 °C"
## [57] "Mean Temp: 25 °C"     "Precipitation: 17.4 mm"
## [59] "Humidity: 74%"        "Dew Point: 20 °C"
## [61] "Wind: 15 km/h"        "Pressure: 1010 mbar"
## [63] "Visibility: 10 km"     "High Temp: 32 °C"
## [65] "Low Temp: 23 °C"      "Mean Temp: 27 °C"
## [67] "Precipitation: 29.4 mm" "Humidity: 77%"
## [69] "Dew Point: 22 °C"     "Wind: 14 km/h"
## [71] "Pressure: 1008 mbar"   "Visibility: 10 km"
## [73] "High Temp: 33 °C"     "Low Temp: 22 °C"
## [75] "Mean Temp: 27 °C"     "Precipitation: 22.5 mm"
## [77] "Humidity: 75%"        "Dew Point: 22 °C"
## [79] "Wind: 13 km/h"        "Pressure: 1010 mbar"
## [81] "Visibility: 10 km"     "High Temp: 29 °C"
## [83] "Low Temp: 20 °C"      "Mean Temp: 25 °C"
## [85] "Precipitation: 43.4 mm" "Humidity: 80%"
## [87] "Dew Point: 20 °C"     "Wind: 12 km/h"
## [89] "Pressure: 1013 mbar"   "Visibility: 10 km"
## [91] "High Temp: 24 °C"     "Low Temp: 14 °C"
## [93] "Mean Temp: 19 °C"     "Precipitation: 75.0 mm"
## [95] "Humidity: 81%"        "Dew Point: 15 °C"
## [97] "Wind: 11 km/h"        "Pressure: 1019 mbar"
## [99] "Visibility: 11 km"     "High Temp: 18 °C"
## [101] "Low Temp: 8 °C"       "Mean Temp: 13 °C"
## [103] "Precipitation: 82.9 mm" "Humidity: 84%"
## [105] "Dew Point: 10 °C"     "Wind: 9 km/h"
## [107] "Pressure: 1022 mbar"   "Visibility: 11 km"
## [109] "High Temp: 15 °C"     "Low Temp: 4 °C"
## [111] "Mean Temp: 9 °C"      "Precipitation: 65.1 mm"
## [113] "Humidity: 83%"        "Dew Point: 6 °C"
## [115] "Wind: 10 km/h"        "Pressure: 1022 mbar"
## [117] "Visibility: 11 km"

```

We need just the information about Temperature and precipitation.

```
information_high_temp <- annual_temp_avarage %>%
  html_elements(xpath = '//*[@id= "climateTable" ]
                //div[@class = "four columns" and position()=1]//p[position() = 1]') %>% html_text()

information_low_temp <- annual_temp_avarage %>%
  html_elements(xpath = '//*[@id= "climateTable" ]
                //div[@class = "four columns" and position()=1]//p[position() = 2]') %>% html_text()

information_mean_temp <- annual_temp_avarage %>%
  html_elements(xpath = '//*[@id= "climateTable" ]
                //div[@class = "four columns" and position()=1]//p[position() = 3]') %>% html_text()

information_precipitation <- annual_temp_avarage %>% html_elements(xpath = '//*[@id= "climateTable" ]
                //div[@class = "four columns" and position()=2]//p[position() = 1]') %>% html_text()

print(information_high_temp)

## [1] "High Temp: 33 °C" "High Temp: 14 °C" "High Temp: 13 °C"
## [4] "High Temp: 16 °C" "High Temp: 21 °C" "High Temp: 27 °C"
## [7] "High Temp: 31 °C" "High Temp: 32 °C" "High Temp: 33 °C"
## [10] "High Temp: 29 °C" "High Temp: 24 °C" "High Temp: 18 °C"
## [13] "High Temp: 15 °C"

print(information_low_temp)

## [1] "Low Temp: 3 °C" "Low Temp: 3 °C" "Low Temp: 4 °C" "Low Temp: 7 °C"
## [5] "Low Temp: 11 °C" "Low Temp: 17 °C" "Low Temp: 20 °C" "Low Temp: 23 °C"
## [9] "Low Temp: 22 °C" "Low Temp: 20 °C" "Low Temp: 14 °C" "Low Temp: 8 °C"
## [13] "Low Temp: 4 °C"

print(information_mean_temp)

## [1] "Mean Temp: 18 °C" "Mean Temp: 8 °C" "Mean Temp: 8 °C"
## [4] "Mean Temp: 12 °C" "Mean Temp: 16 °C" "Mean Temp: 22 °C"
## [7] "Mean Temp: 25 °C" "Mean Temp: 27 °C" "Mean Temp: 27 °C"
## [10] "Mean Temp: 25 °C" "Mean Temp: 19 °C" "Mean Temp: 13 °C"
## [13] "Mean Temp: 9 °C"

print(information_precipitation)

## [1] "Precipitation: 47.4 mm" "Precipitation: 61.3 mm"
## [3] "Precipitation: 59.3 mm" "Precipitation: 54.5 mm"
## [5] "Precipitation: 42.7 mm" "Precipitation: 15.6 mm"
## [7] "Precipitation: 17.4 mm" "Precipitation: 29.4 mm"
## [9] "Precipitation: 22.5 mm" "Precipitation: 43.4 mm"
## [11] "Precipitation: 75.0 mm" "Precipitation: 82.9 mm"
## [13] "Precipitation: 65.1 mm"
```

As I mention before the information in the tables have some other characters such as “C” or “%” which has to be deleted. so we use gsub function. this function can eliminate the characters that we want and replace them by another character. Hier we choose “”, maens nothing. It returns a character value, so we need to use as.numeric to convert it to numeric attribut.

```
high_temp <- as.numeric(gsub("[^0-9]", "", information_high_temp))
low_temp <- as.numeric(gsub("[^0-9]", "", information_low_temp))
mean_temp <- as.numeric(gsub("[^0-9]", "", information_mean_temp))
```

```
precipitation <- as.numeric(gsub("[^0-9]", "", information_precipitation))/10
```

```
print(high_temp)
```

```
## [1] 33 14 13 16 21 27 31 32 33 29 24 18 15
```

```
print(low_temp)
```

```
## [1] 3 3 4 7 11 17 20 23 22 20 14 8 4
```

```
print(mean_temp)
```

```
## [1] 18 8 8 12 16 22 25 27 27 25 19 13 9
```

```
print(precipitation)
```

```
## [1] 47.4 61.3 59.3 54.5 42.7 15.6 17.4 29.4 22.5 43.4 75.0 82.9 65.1
```

This is the results. Pay attention, we have 13 numbers. Beacuse we have also the onformation for whole of the year.

```
final_table <- rbind(high_temp , low_temp, mean_temp, precipitation)
colnames(final_table) <- c("All year" , "January" , "February" , "March" ,
                          "April" , "May", "June" , "July" , "August" , "September" ,
                          "October" , "November" , "December")
print(final_table)
```

```
##           All year January February March April  May June July August
## high_temp      33.0    14.0    13.0  16.0  21.0 27.0 31.0 32.0    33.0
## low_temp       3.0     3.0     4.0   7.0  11.0 17.0 20.0 23.0    22.0
## mean_temp      18.0     8.0     8.0  12.0  16.0 22.0 25.0 27.0    27.0
## precipitation  47.4    61.3    59.3  54.5  42.7 15.6 17.4 29.4    22.5
##           September October November December
## high_temp      29.0     24    18.0    15.0
## low_temp       20.0     14     8.0     4.0
## mean_temp      25.0     19    13.0     9.0
## precipitation  43.4     75    82.9    65.1
```

In this section, the diagrams are illustrated. As you can see, in the first and last month of the year the wethear is vey cold but in the middle which is sommer, This city is very warm.

```
High_df <- data.frame(
  Month = 2:13,
  High = high_temp[2:13]
)
```

```
Low_df <- data.frame(
  Month = 2:13,
  Low = low_temp[2:13]
)
```

```
Mean_df <- data.frame(
  Month = 2:13,
  Mean = mean_temp[2:13]
)
```

```
Precision_df <- data.frame(
```

```

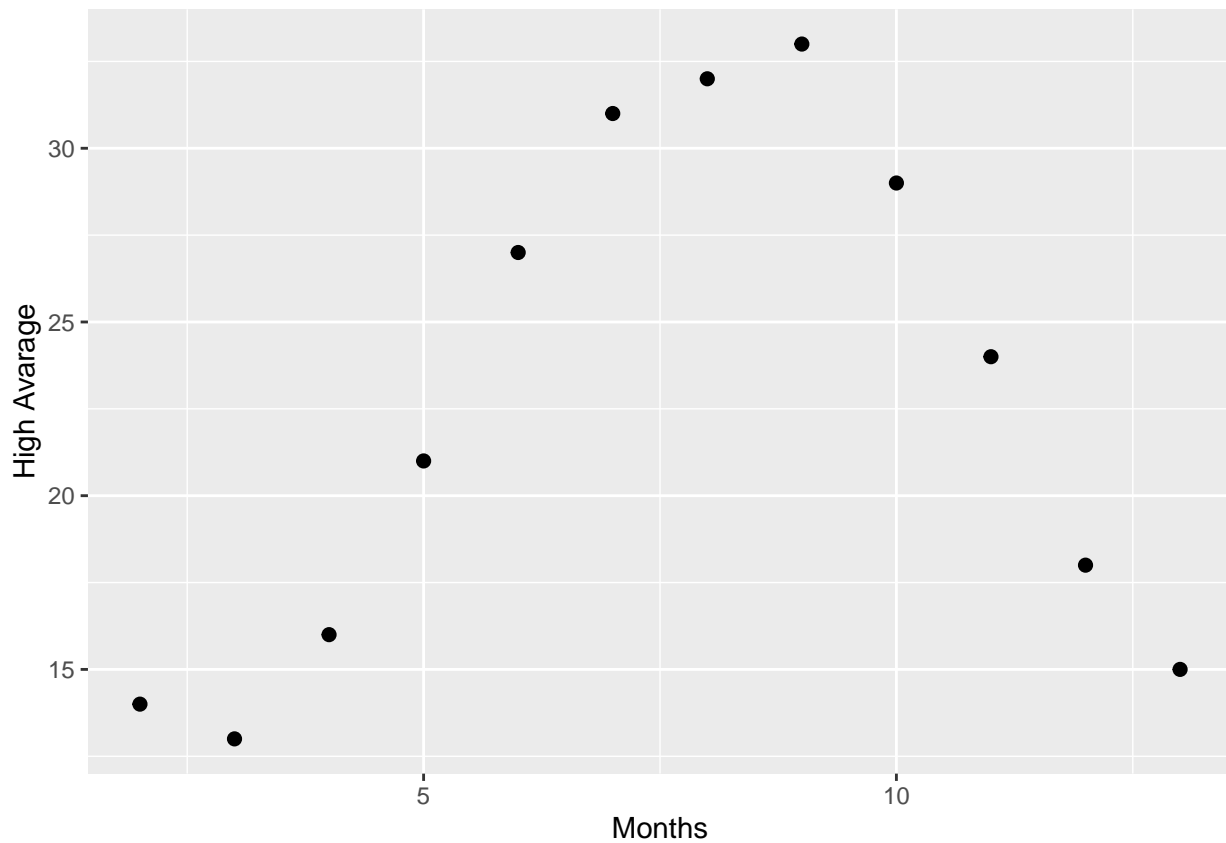
Month = 2:13,
Precision = precipitation[2:13]
)

```

```

ggplot(High_df, aes(x =Month , y = High )) + geom_point(size = 2) +
labs(x = "Months", y = "High Avarage" )

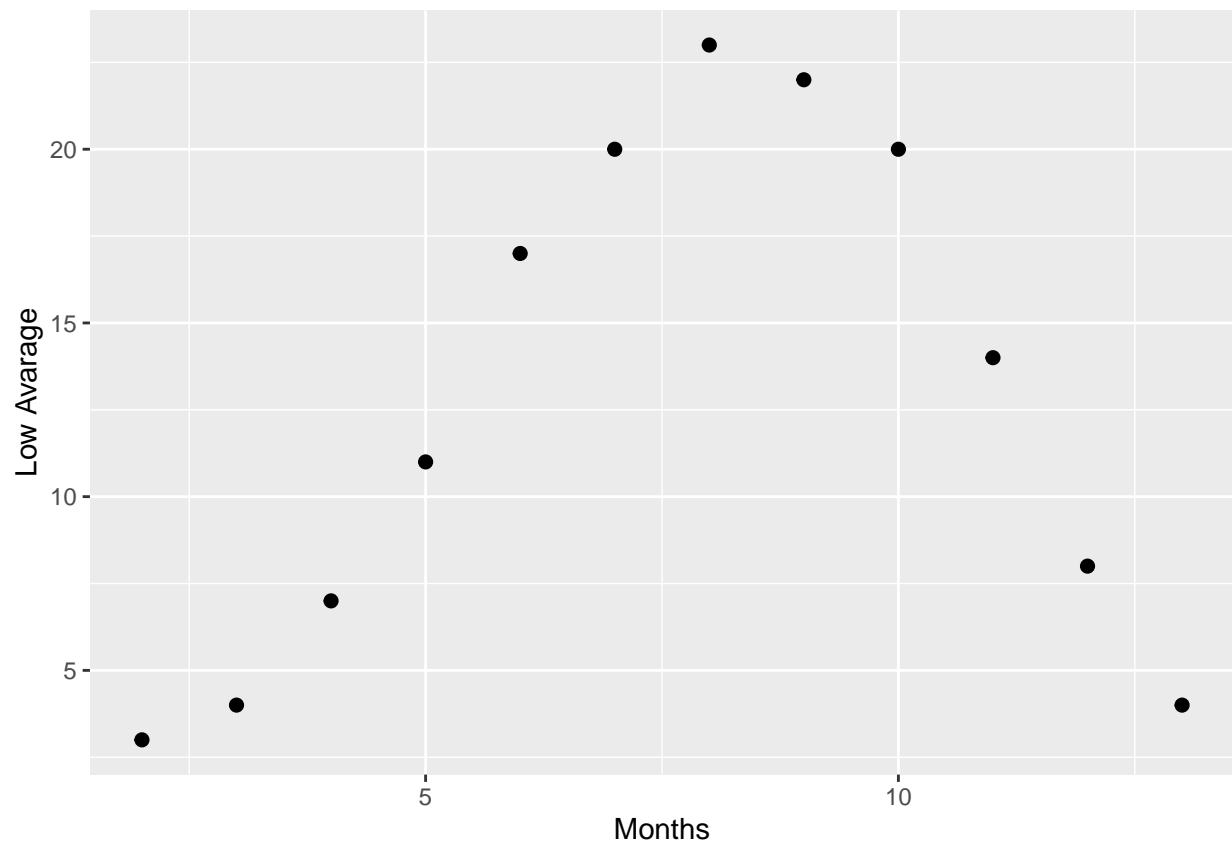
```



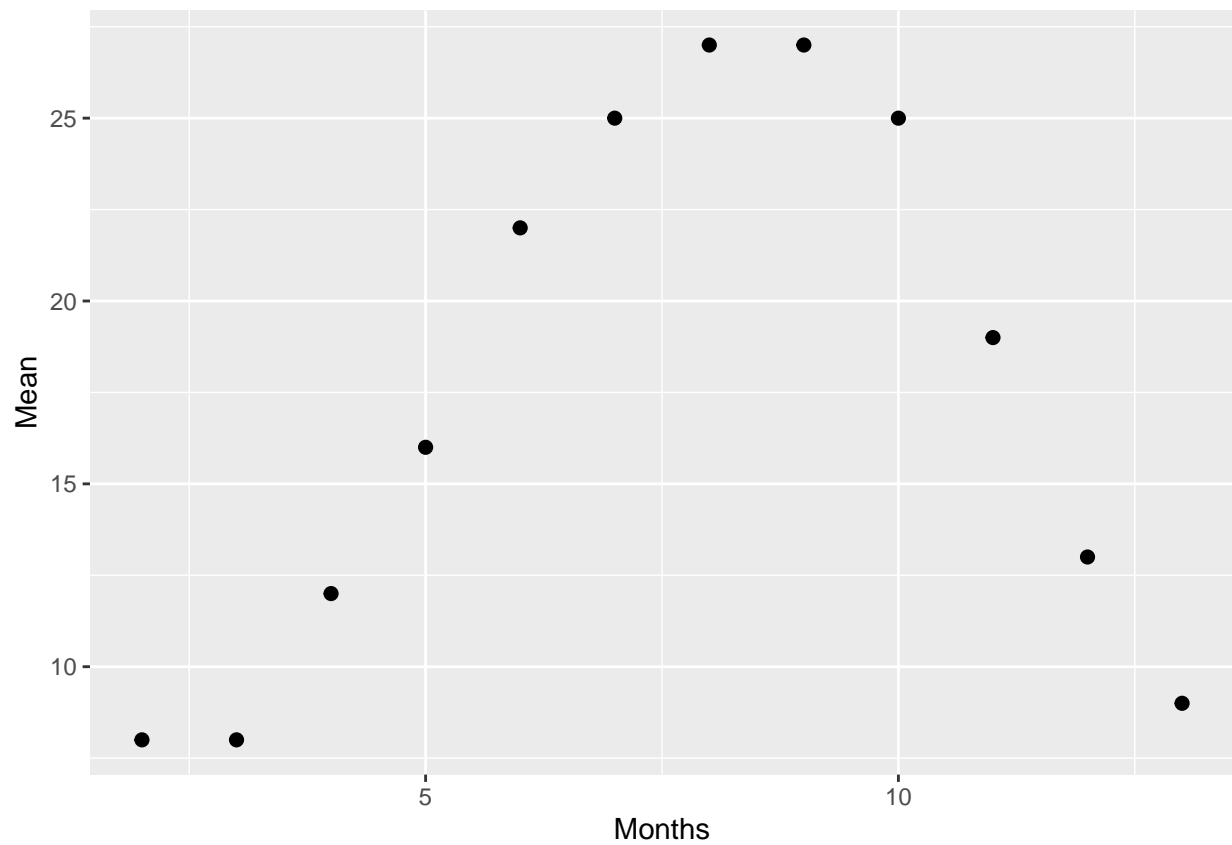
```

ggplot(Low_df, aes(x = Month , y = Low )) + geom_point(size = 2) +
labs(x = "Months", y = "Low Avarage" )

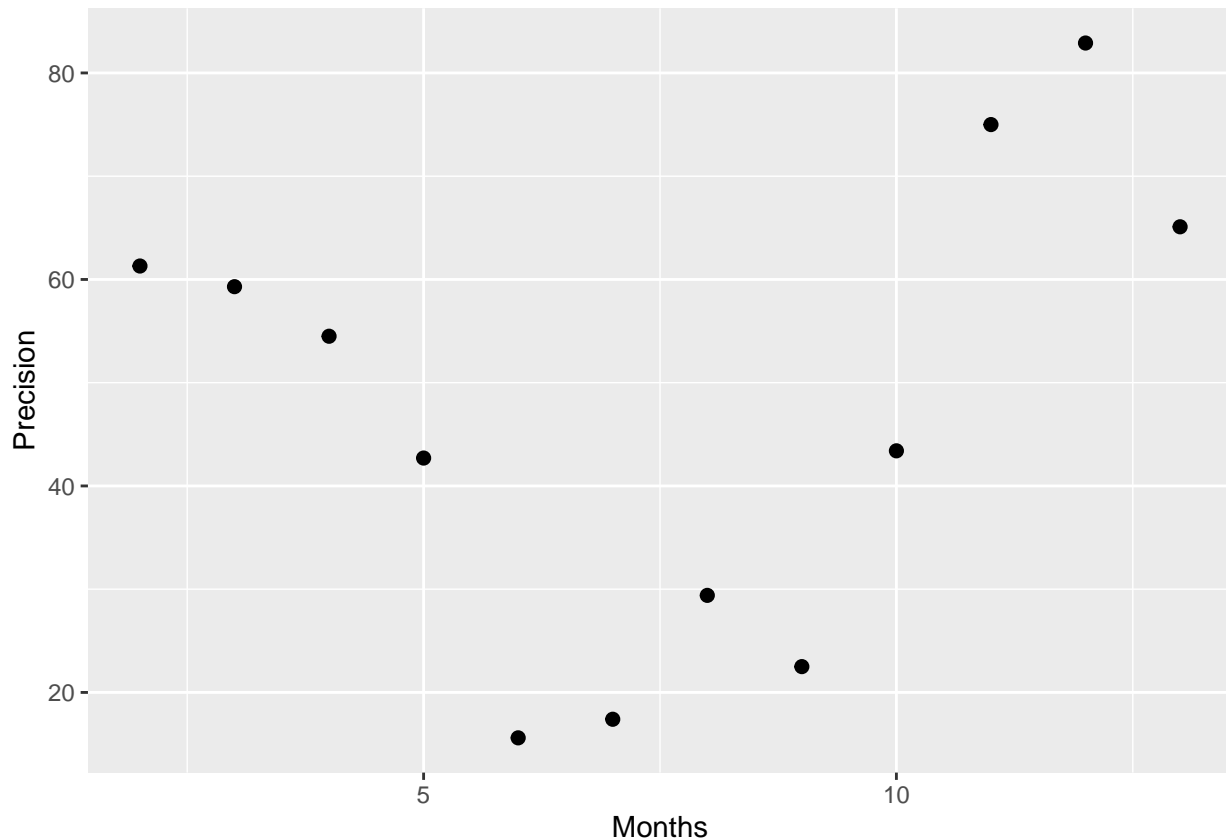
```



```
ggplot(Mean_df , aes(x = Month , y = Mean )) + geom_point(size = 2) +  
  labs(x = "Months", y = "Mean" )
```



```
ggplot(Precision_df , aes(x = Month , y = Precision )) + geom_point(size = 2) +  
  labs(x = "Months", y = "Precision" )
```



```

annual_max <- high_temp[1]
annual_min <- low_temp[1]
annual_mean <- mean_temp[1]
annual_precision <- precipitation[1]

print(paste("The annual min temperture is :", annual_max, "°C"))

## [1] "The annual min temperture is : 33 °C"
print(paste("The annual max temperture is :", annual_min, "°C"))

## [1] "The annual max temperture is : 3 °C"
print(paste("The annual mean temperture is :", annual_mean, "°C"))

## [1] "The annual mean temperture is : 18 °C"
print(paste("The annual precision is :", annual_precision, "mm"))

## [1] "The annual precision is : 47.4 mm"

```

Some of the functions I used.

1. read_html : With this function we can read an html file. 2.html_elements : with this function we can access to the structure and information of the html file. 3.html_table : whit this function we can read and return the information of a table in html as a table. 4.lapply: with this function we can apply a function on each row of a table. 5.html_text : with this function, it is possible to get just the text from an html element.
2. colnames : with this function we cant assign the names of the columns in a data frame using vector. ex: colnames(table) <- c("a", "b", "c") in the code above we assign "a" to the first column, "b" to the

second and “c” to the third one.

3. `gsub` : with this function, we can eliminate some characters and replace them with another character.
4. `as.numeric`, `as.DataFrame`, ... : using them we can read the variables as a specific type.