

Data Acquisition and Survey Methods

Assignment 1

Fani Sentinella-Jerbić

2023-05-15

Contents

Introduction	1
Forecast for the next 48 hours	1
Detailed Hourly Forecast For the Next 24 Hours	3
Annual Weather Climate Averages	7

Introduction

I will be scraping data for Frankfurt as my name starts with F.

```
library(kableExtra)
library(dplyr)
library(rvest)
library(ggplot2)
```

Forecast for the next 48 hours

Reference image:








	Monday	Tuesday				Wednesday	
	Evening	Night	Morning	Afternoon	Evening	Night	Morning
Forecast							
Temperature	16 °C	11 °C	11 °C	15 °C	12 °C	8 °C	10 °C
	Isolated thunderstorms. Partly cloudy.	Passing showers. Mostly cloudy.	Passing showers. Overcast.	Passing showers. Mostly cloudy.	Overcast.	Cloudy.	Overcast.
Feels Like	16 °C	10 °C	8 °C	14 °C	11 °C	7 °C	8 °C
Wind Speed	8 km/h	7 km/h	17 km/h	18 km/h	16 km/h	9 km/h	12 km/h
Wind Direction	NNW ↘	NW ↘	N ↓	N ↓	N ↓	N ↓	N ↓
Humidity	69%	81%	80%	42%	52%	70%	63%
Dew Point	10 °C	8 °C	7 °C	3 °C	3 °C	3 °C	3 °C
Visibility	11 km	10 km	8 km	15 km	15 km	12 km	13 km
Probability of Precipitation	52%	19%	30%	31%	0%	1%	0%
Amount of Rain	1.3 mm	0.2 mm	1.3 mm	0.5 mm	-	-	-
* Updated Monday, 15 May 2023 20:44:06 Frankfurt time - Weather by CustomWeather, © 2023							

Figure 1: Forecast For the Next 48 Hours

I retrieved the whole table element from the HTML using xpath and its element id. Afterwards I used the built-in function to populate a table with the scraped data. I had to additionally remove the empty Forecast row which is filled with icons in the original, and the last row which was just the footer of the table. The result is printed below.

```
document <- read_html("https://www.timeanddate.com/weather/germany/frankfurt")

table <- document %>%
  html_elements(xpath="//*[@id="wt-48"]') %>%
  html_table(fill = TRUE)

table <- table[[1]] %>% as.data.frame()
table[1,1] = 'Time of Day'
table[4,1] = 'Description'
names(table)[1] = 'Aspect'

























# remove empty Forecast row and footer
table <- table[-c(2,13), ]
rownames(table) <- NULL

table %>% kable(format="latex", booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "scale_down", "HOLD_position"))
```

Aspect	Montag	Dienstag	Dienstag	Dienstag	Dienstag	Mittwoch	Mittwoch
Time of Day	Evening	Night	Morning	Afternoon	Evening	Night	Morning
Temperature	16 °C	11 °C	11 °C	15 °C	12 °C	8 °C	10 °C
Description	Isolated thunderstorms. Partly cloudy.	Passing showers. Mostly cloudy.	Passing showers. Overcast.	Passing showers. Mostly cloudy.	Overcast.	Cloudy.	Overcast.
Feels Like	16 °C	10 °C	8 °C	14 °C	11 °C	7 °C	8 °C
Wind Speed	8 km/h	7 km/h	17 km/h	18 km/h	16 km/h	9 km/h	12 km/h
Wind Direction	NNW↑	NW↑	N↑	N↑	N↑	N↑	N↑
Humidity	69%	81%	80%	42%	52%	70%	63%
Dew Point	10 °C	8 °C	7 °C	3 °C	3 °C	3 °C	3 °C
Visibility	11 km	10 km	8 km	15 km	15 km	12 km	13 km
Probability of Precipitation	52%	19%	30%	31%	0%	1%	0%
Amount of Rain	1.3 mm	0.2 mm	1.3 mm	0.5 mm	-	-	-

Detailed Hourly Forecast For the Next 24 Hours

Reference image:

Time	Conditions		Comfort			Precipitation	
	Temp	Weather	Feels Like	Wind	Humidity	Chance	Amount
22:00 Mon, 15 May	15 °C	 Passing showers. Partly cloudy.	15 °C	7 km/h	↘ 67%	26%	0.1 mm (rain)
23:00	14 °C	 Passing showers. Broken clouds.	14 °C	6 km/h	↘ 71%	23%	0.1 mm (rain)
00:00 Tue, 16 May	13 °C	 Passing showers. Broken clouds.	13 °C	7 km/h	↘ 75%	22%	0.1 mm (rain)
01:00	12 °C	 Passing showers. Broken clouds.	11 °C	7 km/h	↘ 79%	21%	0.1 mm (rain)
02:00	11 °C	 Passing showers. Partly cloudy.	10 °C	8 km/h	↘ 83%	21%	0.1 mm (rain)
03:00	10 °C	 Passing showers. Broken clouds.	10 °C	8 km/h	↘ 85%	22%	0.1 mm (rain)
04:00	10 °C	 Passing showers. Mostly cloudy.	9 °C	8 km/h	↘ 86%	24%	0.1 mm (rain)
05:00	10 °C	 Passing showers. Cloudy.	9 °C	8 km/h	↘ 86%	19%	0.1 mm (rain)
06:00	10 °C	 Passing showers. Overcast.	9 °C	10 km/h	↘ 86%	18%	0.1 mm (rain)
07:00	11 °C	 Passing showers. Overcast.	9 °C	12 km/h	↘ 85%	22%	0.1 mm (rain)
08:00	11 °C	 Passing showers. Overcast.	10 °C	14 km/h	↘ 83%	30%	0.2 mm (rain)
09:00	11 °C	 Passing showers. Overcast.	10 °C	13 km/h	↓ 80%	37%	0.2 mm (rain)
10:00	11 °C	 Passing showers. Overcast.	9 °C	13 km/h	↓ 77%	44%	0.3 mm (rain)
11:00	11 °C	 Passing showers. Overcast.	9 °C	13 km/h	↓ 74%	49%	0.4 mm (rain)
12:00	12 °C	 Passing showers. Overcast.	10 °C	15 km/h	↓ 64%	53%	0.3 mm (rain)
13:00	14 °C	 Passing showers. Cloudy.	13 °C	17 km/h	↓ 52%	55%	0.2 mm (rain)
14:00	15 °C	 Passing showers. Cloudy.	14 °C	19 km/h	↓ 44%	46%	0.2 mm (rain)
15:00	16 °C	 Passing showers. Mostly cloudy.	14 °C	18 km/h	↓ 41%	36%	0.1 mm (rain)
16:00	16 °C	 Passing showers. Cloudy.	16 °C	18 km/h	↗ 40%	25%	0.1 mm (rain)
17:00	16 °C	 Overcast.	16 °C	18 km/h	↗ 41%	0%	-
18:00	15 °C	 Overcast.	14 °C	18 km/h	↗ 42%	0%	-
19:00	15 °C	 Overcast.	13 °C	17 km/h	↓ 45%	0%	-
20:00	13 °C	 Overcast.	12 °C	17 km/h	↓ 49%	0%	-
21:00	13 °C	 Overcast.	11 °C	16 km/h	↓ 51%	0%	-

* Updated Monday, 15 May 2023 21:13:50 Frankfurt time - Weather by CustomWeather, © 2023

Figure 2: Hourly Forecast For the Next 24 Hours

Once again, I retrieved the whole table element from the HTML using xpath and its element id. Afterwards I used the the built-in function again to populate a table with the scraped data. I also removed the footer and icon row again. The result is printed below.

```
document_hourly <- read_html("https://www.timeanddate.com/weather/germany/frankfurt/hourly")

table <- document_hourly %>%
  html_elements(xpath='//*[@id="wt-hbh"]') %>%
  html_table(fill = TRUE)

table <- table[[1]] %>% as.data.frame()
table <- table[-c(26),-c(2)]
rownames(table) <- NULL

table %>% kable(format="latex", booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "scale_down", "HOLD_position"))
```

	Conditions	Conditions.1	Comfort	Comfort.1	Comfort.2	Comfort.3	Precipitation	Precipitation.1
Time	Temp	Weather	Feels Like	Wind		Humidity	Chance	Amount
22:00Mo, 15. Mai	15 °C	Passing showers. Partly cloudy.	15 °C	7 km/h	↑	67%	26%	0.1 mm (rain)
23:00	14 °C	Passing showers. Broken clouds.	14 °C	6 km/h	↑	71%	23%	0.1 mm (rain)
00:00Di, 16. Mai	13 °C	Passing showers. Broken clouds.	13 °C	7 km/h	↑	75%	22%	0.1 mm (rain)
01:00	12 °C	Passing showers. Broken clouds.	11 °C	7 km/h	↑	79%	21%	0.1 mm (rain)
02:00	11 °C	Passing showers. Partly cloudy.	10 °C	8 km/h	↑	83%	21%	0.1 mm (rain)
03:00	10 °C	Passing showers. Broken clouds.	10 °C	8 km/h	↑	85%	22%	0.1 mm (rain)
04:00	10 °C	Passing showers. Mostly cloudy.	9 °C	8 km/h	↑	86%	24%	0.1 mm (rain)
05:00	10 °C	Passing showers. Cloudy.	9 °C	8 km/h	↑	86%	19%	0.1 mm (rain)
06:00	10 °C	Passing showers. Overcast.	9 °C	10 km/h	↑	86%	18%	0.1 mm (rain)
07:00	11 °C	Passing showers. Overcast.	9 °C	12 km/h	↑	85%	22%	0.1 mm (rain)
08:00	11 °C	Passing showers. Overcast.	10 °C	14 km/h	↑	83%	30%	0.2 mm (rain)
09:00	11 °C	Passing showers. Overcast.	10 °C	13 km/h	↑	80%	37%	0.2 mm (rain)
10:00	11 °C	Passing showers. Overcast.	9 °C	13 km/h	↑	77%	44%	0.3 mm (rain)
11:00	11 °C	Passing showers. Overcast.	9 °C	13 km/h	↑	74%	49%	0.4 mm (rain)
12:00	12 °C	Passing showers. Overcast.	10 °C	15 km/h	↑	64%	53%	0.3 mm (rain)
13:00	14 °C	Passing showers. Cloudy.	13 °C	17 km/h	↑	52%	55%	0.2 mm (rain)
14:00	15 °C	Passing showers. Cloudy.	14 °C	19 km/h	↑	44%	46%	0.2 mm (rain)
15:00	16 °C	Passing showers. Mostly cloudy.	14 °C	18 km/h	↑	41%	36%	0.1 mm (rain)
16:00	16 °C	Passing showers. Cloudy.	16 °C	18 km/h	↑	40%	25%	0.1 mm (rain)
17:00	16 °C	Overcast.	16 °C	18 km/h	↑	41%	0%	-
18:00	15 °C	Overcast.	14 °C	18 km/h	↑	42%	0%	-
19:00	15 °C	Overcast.	13 °C	17 km/h	↑	45%	0%	-
20:00	13 °C	Overcast.	12 °C	17 km/h	↑	49%	0%	-
21:00	13 °C	Overcast.	11 °C	16 km/h	↑	51%	0%	-

Visualization

For the plots I used a simple line plot from ggplot.

```
times <- table[[1]][-1]
temperature <- table$Conditions[-1]
wind <- table$Comfort.1[-1]
humidity <- table$Comfort.3[-1]

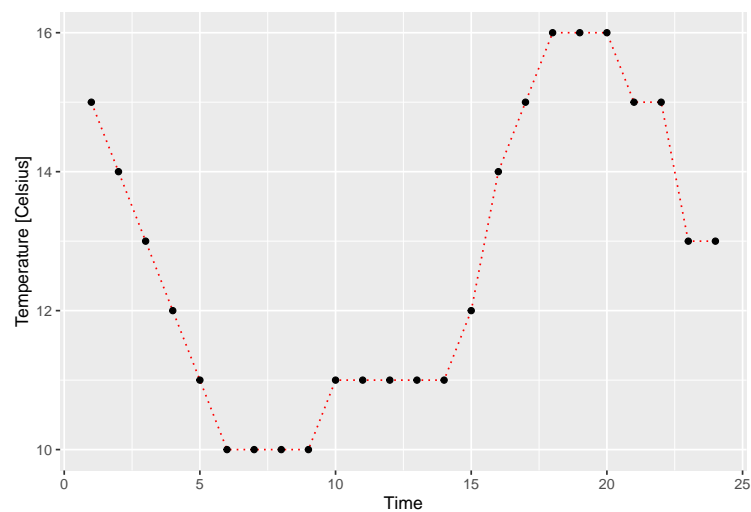
times <- unlist(lapply(times, FUN = substr, start=1, stop=5))
temperature <- substr(temperature, 1, nchar(temperature)-3) %>% as.integer()
wind <- substr(wind, 1, nchar(wind)-5) %>% as.integer()
humidity <- substr(humidity, 1, nchar(humidity)-1) %>% as.integer()

vis <- data.frame(times, temperature, wind, humidity)

vis %>% kable(format="latex", booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "HOLD_position"))
```

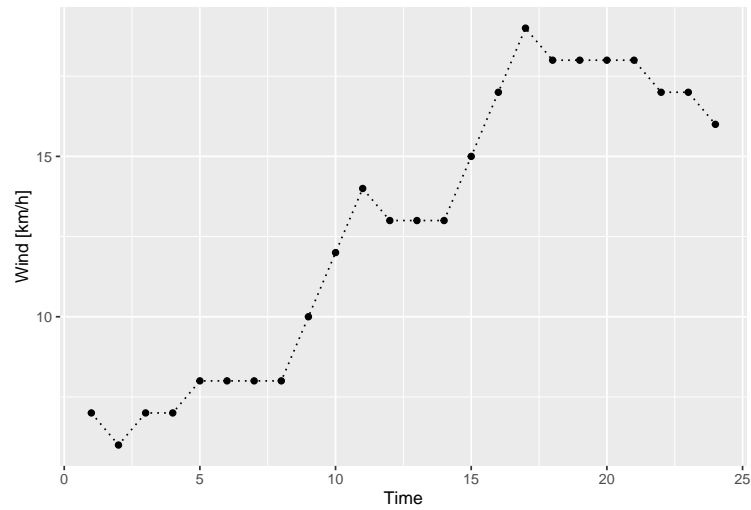
times	temperature	wind	humidity
22:00	15	7	67
23:00	14	6	71
00:00	13	7	75
01:00	12	7	79
02:00	11	8	83
03:00	10	8	85
04:00	10	8	86
05:00	10	8	86
06:00	10	10	86
07:00	11	12	85
08:00	11	14	83
09:00	11	13	80
10:00	11	13	77
11:00	11	13	74
12:00	12	15	64
13:00	14	17	52
14:00	15	19	44
15:00	16	18	41
16:00	16	18	40
17:00	16	18	41
18:00	15	18	42
19:00	15	17	45
20:00	13	17	49
21:00	13	16	51

```
vis %>% ggplot(aes(y=temperature, x=1:nrow(vis))) +
  geom_line(color="red", linetype="dotted") +
  geom_point() +
  xlab("Time") + ylab("Temperature [Celsius]")
```

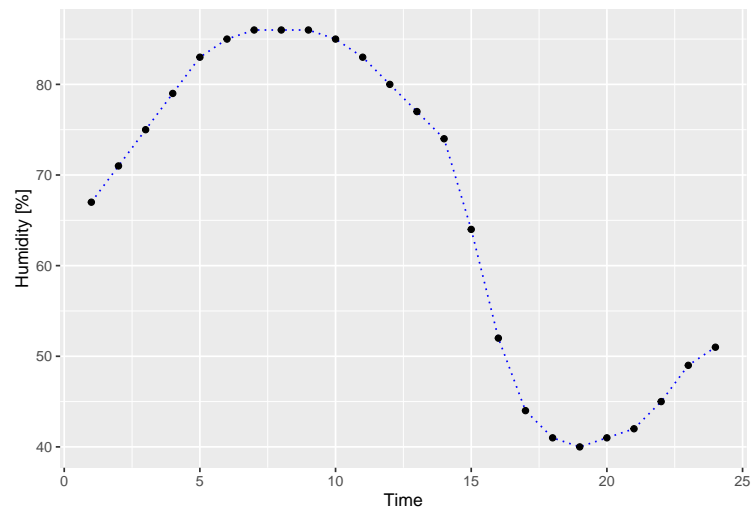


```
vis %>% ggplot(aes(y=wind, x=1:nrow(vis))) +
  geom_line(linetype="dotted") +
  geom_point() +
```

```
xlab("Time") + ylab("Wind [km/h]")
```



```
vis %>% ggplot(aes(y=humidity, x=1:nrow(vis))) +  
  geom_line(color="blue", linetype="dotted")+  
  geom_point() +  
  xlab("Time") + ylab("Humidity [%]")
```



Annual Weather Climate Averages

Reference image:

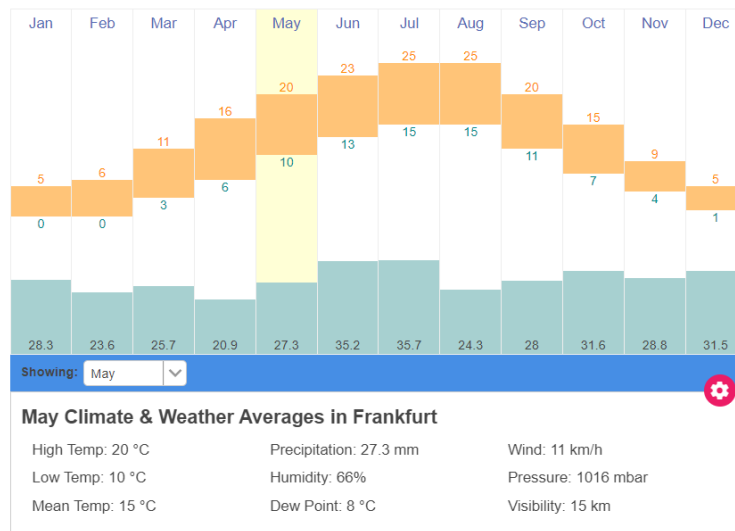


Figure 3: Annual Weather Climate Averages

This one was somewhat more complicated. I checked the HTML structure to find this:

```
January
//*[@id="climateTable"]/div[2]/div[1]/p[1]/text()
//*[@id="climateTable"]/div[2]/div[1]/p[2]/text()
//*[@id="climateTable"]/div[2]/div[1]/p[3]/text()

February
//*[@id="climateTable"]/div[3]/div[1]/p[1]/text()
//*[@id="climateTable"]/div[3]/div[1]/p[2]/text()
//*[@id="climateTable"]/div[3]/div[1]/p[3]/text()

...
```

From this I inferred how the indexes work for different elements of the table. Then I created a special function for generating xpaths of these elements. I also had to perform some data cleaning which I extracted to a special function as well.

```
document_climate <- read_html("https://www.timeanddate.com/weather/germany/frankfurt/climate")

# Cleans retrieved text of spaces and measuring units
clean <- function(s, n){
  substr(s, 2, nchar(s)-n) %>% as.numeric()
}

# Returns the xpath of the element
get_xpath <- function(month, temp){
  toggle = 1

  if(is.na(temp)){ # precipitation path is requested
    temp = 1
    toggle = 2
  }
}
```

```

paste('//*[id="climateTable"]/div[' , month, ']/div[' , toggle, ']/p[' , temp, ']/text()', sep='')
}

# Fetches the element from the html
fetch_element <- function(month, temp){
  document_climate %>%
    html_elements(xpath=get_xpath(month, temp)) %>%
    html_text()
}

month <- c("January", "February", "March", "April", "May",
           "June", "July", "August", "September", "October",
           "November", "December")
highest <- c()
lowest <- c()
mean <- c()
precipitation <- c()

for (m in c(2:13)){
  highest <- c(highest, fetch_element(m, 1) %>% clean(3))
  lowest <- c(lowest, fetch_element(m, 2) %>% clean(3))
  mean <- c(mean, fetch_element(m, 3) %>% clean(3))
  precipitation <- c(precipitation, fetch_element(m, NA) %>% clean(4))
}

final <- data.frame(month, highest, lowest, mean, precipitation)
final$month <- factor(final$month, levels=month)
final %>% kable(format="latex", booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "HOLD_position"))

```

month	highest	lowest	mean	precipitation
January	5	0	2	28.3
February	6	0	3	23.6
March	11	3	7	25.7
April	16	6	11	20.9
May	20	10	15	27.3
June	23	13	18	35.2
July	25	15	20	35.7
August	25	15	20	24.3
September	20	11	16	28.0
October	15	7	11	31.6
November	9	4	6	28.8
December	5	1	3	31.5

Visualization

I plotted the temperatures together with red line representing the highest, blue the lowest, and black the mean temperatures. The precipitation I plotted using bars like they usually do in meteorology.

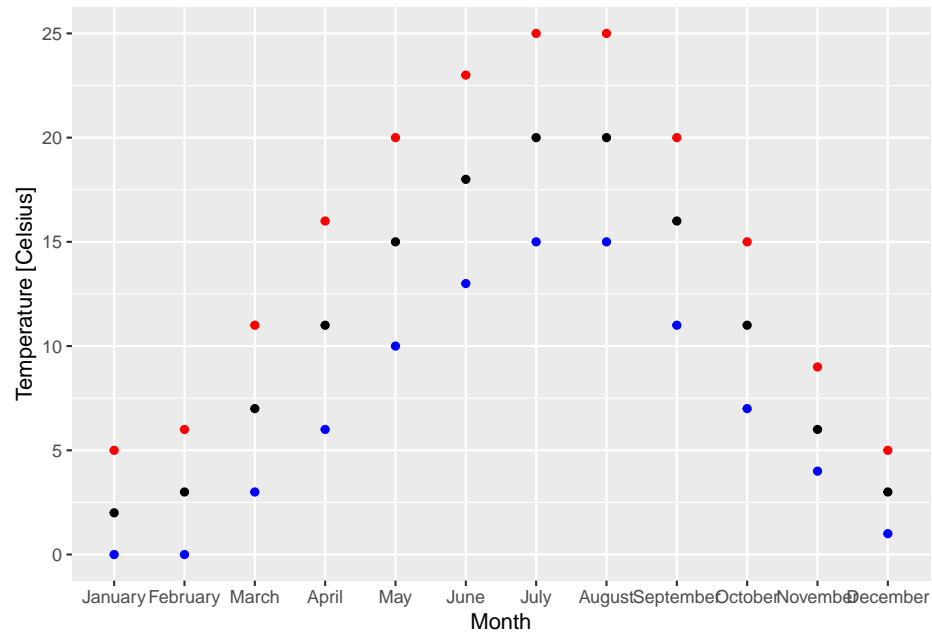
```

final %>% ggplot() +
  geom_point(aes(x=month, y=highest), color='red') +
  geom_point(aes(x=month, y=mean)) +

```



```
geom_point(aes(x=month, y=lowest), color='blue') +
xlab("Month") + ylab("Temperature [Celsius]")
```



```
final %>% ggplot() +
  geom_col(aes(x=month, y=precipitation)) +
  xlab("Month") + ylab("Precipitation [mm]")
```

