Wetteranalyse für die Wahl eines neuen Wohnorts

Julien Jäger

20 Juni 2024

1. Skript zum Einlesen der Daten und Berechnungen zum Variablitätsscore.

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE, fig.width=14)

# Laden der notwendigen Bibliotheken
library(tidyverse)
```

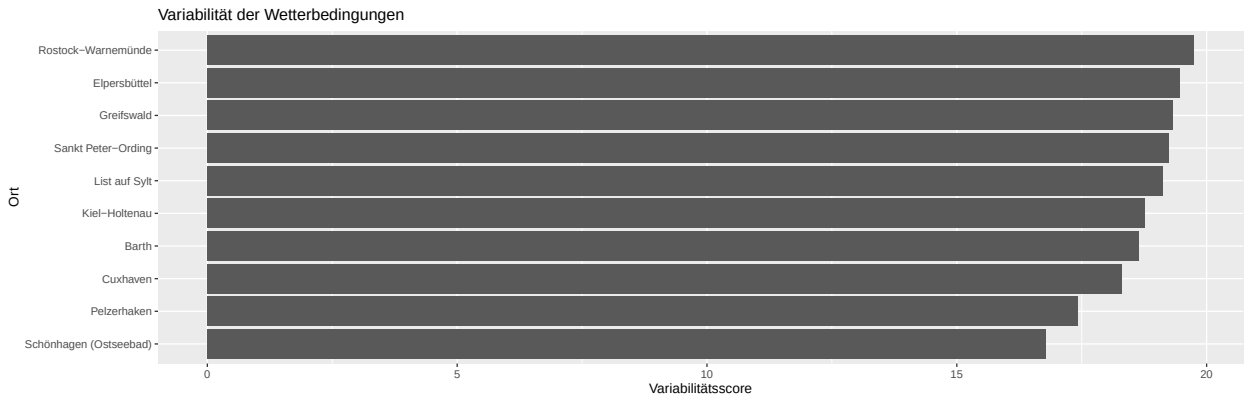
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(dplyr)
```

```
# Einlesen der Temperaturdaten
temp_data <- read_csv("data/data_OBS_DEU_P1D_T2M_X.csv",
                      col_names = c("Produkt_Code", "SDO_ID",
                                    "Zeitstempel", "Wert", "Qualitaet_Byte",
                                    "Qualitaet_Niveau"),
                      skip = 1,
                      col_types = cols(
                        Produkt_Code = col_character(),
                        SDO_ID = col_integer(),
                        Zeitstempel = col_date(format = "%Y-%m-%d"),
                        Wert = col_double(),
                        Qualitaet_Byte = col_integer(),
                        Qualitaet_Niveau = col_integer()
                      ))

# Einlesen der Feuchtigkeitsdaten
humidity_data <- read_csv("data/data_OBS_DEU_P1D_RF.csv",
                          col_names = c("Produkt_Code", "SDO_ID",
                                        "Zeitstempel","Wert", "Qualitaet_Byte",
                                        "Qualitaet_Niveau"),
```

```
                          skip = 1,
                          col_types = cols(
                            Produkt_Code = col_character(),
                            SDO_ID = col_integer(),
                            Zeitstempel = col_date(format = "%Y-%m-%d"),
                            Wert = col_double(),
                            Qualitaet_Byte = col_integer(),
                            Qualitaet_Niveau = col_integer()
                          ))

# Einlesen der Windgeschwindigkeitsdaten
wind_data <- read_csv("data/data_OBS_DEU_P1D_F.csv",
                      col_names = c("Produkt_Code", "SDO_ID", "Zeitstempel",
                                    "Wert", "Qualitaet_Byte",
                                    "Qualitaet_Niveau"),
                      skip = 1,
                      col_types = cols(
                        Produkt_Code = col_character(),
                        SDO_ID = col_integer(),
                        Zeitstempel = col_date(format = "%Y-%m-%d"),
                        Wert = col_double(),
                        Qualitaet_Byte = col_integer(),
                        Qualitaet_Niveau = col_integer()
                      ))

# Einlesen der Stationsmetadaten
# Hier reicht eine Datei, der drei Dimensionen, da die Stationsnamen
# und IDs gleich sind
station_data <- read_csv("data/sdo_OBS_DEU_P1D_F.csv",
                         col_names = c("SDO_ID", "SDO_Name", "Geogr_Laenge",
                                       "Geogr_Breite", "Hoehe_ueber_NN",
                                       "Metadata_Link"),
                         skip = 1,
                         col_types = cols(
                           SDO_ID = col_integer(),
                           SDO_Name = col_character(),
                           Geogr_Laenge = col_character(),
                           Geogr_Breite = col_character(),
                           Hoehe_ueber_NN = col_double(),
                           Metadata_Link = col_character()
                         ))

# Datenaufbereitung: Umbenennen der Spalten und Zusammenführen der Datensätze
colnames(temp_data) <- c("Product_Code", "Location_ID", "Date", "Temperature",
                         "Quality_Byte", "Quality_Level")
colnames(humidity_data) <- c("Product_Code", "Location_ID", "Date", "Humidity",
                             "Quality_Byte", "Quality_Level")
colnames(wind_data) <- c("Product_Code", "Location_ID", "Date", "WindSpeed",
                         "Quality_Byte", "Quality_Level")

# Zusammenführen der Wetterdatensätze
weather_data <- temp_data %>%
  select(Location_ID, Date, Temperature) %>%
```

```r
  left_join(humidity_data %>% select(Location_ID, Date, Humidity),
            by = c("Location_ID", "Date")) %>%
  left_join(wind_data %>% select(Location_ID, Date, WindSpeed),
            by = c("Location_ID", "Date"))

# Zusammenführen der Wetterdaten mit den Stationsnamen
weather_data <- weather_data %>%
  left_join(station_data %>% select(SDO_ID, SDO_Name),
            by = c("Location_ID" = "SDO_ID"))

# Sortieren nach Stationsnamen
# und Berechnung der Standardabweichung für jede Dimension
variability <- weather_data %>%
  group_by(SDO_Name) %>%
  summarise(
    temp_sd = sd(Temperature, na.rm = TRUE),
    humidity_sd = sd(Humidity, na.rm = TRUE),
    wind_sd = sd(WindSpeed, na.rm = TRUE)
  )

# Berechnung des Variabilitätsscores und als Spalte hinzufügen
variability <- variability %>%
  mutate(v_score = temp_sd + humidity_sd + wind_sd)

# Sortierung der Orte nach Variabilitätsscore
sorted_locations <- variability %>%
  arrange(v_score)

# Variabilitätsplot
print(ggplot(sorted_locations, aes(x = reorder(SDO_Name, v_score),
                                   y = v_score)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Variabilität der Wetterbedingungen", x = "Ort",
       y = "Variabilitätsscore")
)
```




```r
# Ergebnisse anzeigen
head(sorted_locations, 10)
```

```
## # A tibble: 10 x 5
##    SDO_Name               temp_sd humidity_sd wind_sd v_score
##    <chr>                    <dbl>       <dbl>   <dbl>   <dbl>
##  1 Schönhagen (Ostseebad)    6.71        7.74    2.32    16.8
##  2 Pelzerhaken               7.14        8.35    1.95    17.4
##  3 Cuxhaven                  7.00        9.19    2.11    18.3
##  4 Barth                     7.62        9.06    1.97    18.6
##  5 Kiel-Holtenau             7.31        9.80    1.66    18.8
##  6 List auf Sylt             6.62        9.79    2.72    19.1
##  7 Sankt Peter-Ording        6.99        9.48    2.77    19.2
##  8 Greifswald                7.88        9.90    1.56    19.3
##  9 Elpersbüttel              7.31        9.60    2.57    19.5
## 10 Rostock-Warnemünde        7.54       10.0     2.17    19.7
```

Der Plot zeigt die ausgewählten Orte aus den Daten. Dieser zeigt die ausgewählten zehn Orte, welche anhand des Variablitätsscores sortiert wurden. Somit hat der Ort Schönhagen die niedrigsten kombinierten Abweichungen von der Tageshöchsttemperatur, der mittleren Windgeschwindigkeit und der mittleren relativen Luftfeuchtigkeit.

2. Skript zum Berechnen der zukünftigen Daten für die nächsten 3 Jahre

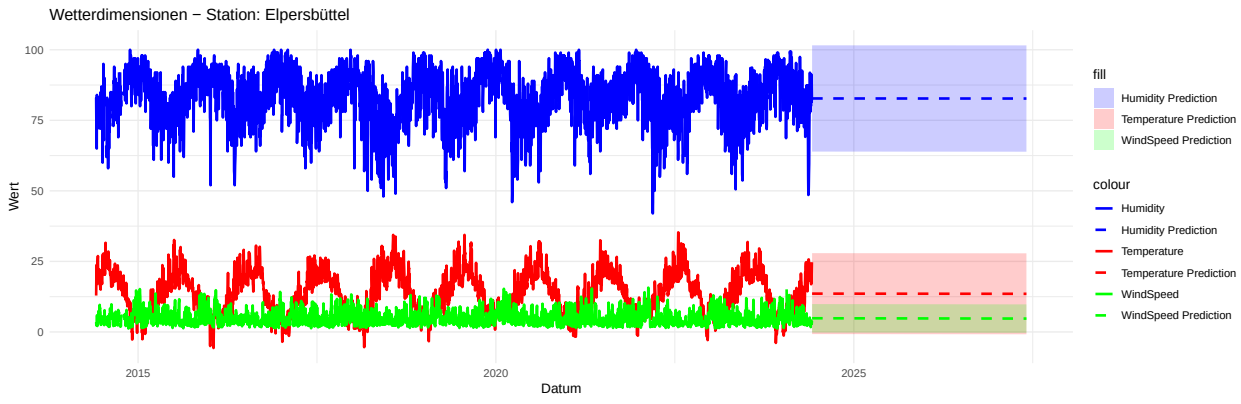
```r
# Stationsnamen extrahieren
stations <- unique(weather_data$SDO_Name)

# Leere Liste für die separaten Tabellen
station_tables <- list()

# Daten für jede Station extrahieren und in separate Tabellen speichern
for (station in stations) {
  station_data <- weather_data %>% filter(SDO_Name == station)
  station_tables[[station]] <- station_data
}
# Leere Liste für die kombinierten Vorhersagen
combined_predictions <- list()

# Für jede Station lineare Regression durchführen und Vorhersagen erstellen
for (station in stations) {
  # Daten für die aktuelle Station
  station_data <- station_tables[[station]]


  # length.out = 1095 -> 3 Jahre
  # Lineare Regression für Temperatur
  lm_temperature <- lm(Temperature ~ Date, data = station_data)
  future_dates <- seq(as.Date(max(station_data$Date)) + 1,
                      length.out = 1095, by = "day")
  future_data <- data.frame(Date = future_dates)
  predictions_temperature <- predict(lm_temperature, newdata = future_data,
                                     interval = "prediction")
  future_data$Predicted_Temperature <- predictions_temperature[, "fit"]
  future_data$Temperature_Upper <- predictions_temperature[, "upr"]
  future_data$Temperature_Lower <- predictions_temperature[, "lwr"]
```

```
  # Lineare Regression für Luftfeuchtigkeit
  lm_humidity <- lm(Humidity ~ Date, data = station_data)
  predictions_humidity <- predict(lm_humidity, newdata = future_data,
                                  interval = "prediction")
  future_data$Predicted_Humidity <- predictions_humidity[, "fit"]
  future_data$Humidity_Upper <- predictions_humidity[, "upr"]
  future_data$Humidity_Lower <- predictions_humidity[, "lwr"]

  # Lineare Regression für Windgeschwindigkeit
  lm_windspeed <- lm(WindSpeed ~ Date, data = station_data)
  predictions_windspeed <- predict(lm_windspeed, newdata = future_data,
                                   interval = "prediction")
  future_data$Predicted_WindSpeed <- predictions_windspeed[, "fit"]
  future_data$WindSpeed_Upper <- predictions_windspeed[, "upr"]
  future_data$WindSpeed_Lower <- predictions_windspeed[, "lwr"]

  # Station als Spalte hinzufügen
  future_data$SDO_Name <- station

  # Ergebnisse zur kombinierten Liste hinzufügen
  combined_predictions[[station]] <- future_data
}

# Daten für die letzten 10 Jahre für die Plots
historical_data <- weather_data

# Für jede Station historische und Vorhersagedaten kombinieren
combined_data <- lapply(names(combined_predictions), function(station) {
  historical <- historical_data %>% filter(SDO_Name == station)
  predicted <- combined_predictions[[station]]
  combined <- bind_rows(historical, predicted)

})

# Funktion zur Erstellung des Plots für jede Station
plot_station_data <- function(data) {
  ggplot(data, aes(x = Date)) +
    geom_line(aes(y = Temperature, color = "Temperature"), size = 1) +
    geom_line(aes(y = Predicted_Temperature, color = "Temperature Prediction"),
              size = 1, linetype = "dashed") +
    geom_ribbon(aes(ymin = Temperature_Lower, ymax = Temperature_Upper,
                    fill = "Temperature Prediction"),
                alpha = 0.2, color = NA) +
    geom_line(aes(y = Humidity, color = "Humidity"), size = 1) +
    geom_line(aes(y = Predicted_Humidity, color = "Humidity Prediction"),
              size = 1, linetype = "dashed") +
    geom_ribbon(aes(ymin = Humidity_Lower, ymax = Humidity_Upper,
                    fill = "Humidity Prediction"), alpha = 0.2, color = NA) +
    geom_line(aes(y = WindSpeed, color = "WindSpeed"), size = 1) +
    geom_line(aes(y = Predicted_WindSpeed, color = "WindSpeed Prediction"),
              size = 1, linetype = "dashed") +
    geom_ribbon(aes(ymin = WindSpeed_Lower, ymax = WindSpeed_Upper,
                    fill = "WindSpeed Prediction"), alpha = 0.2, color = NA) +
```

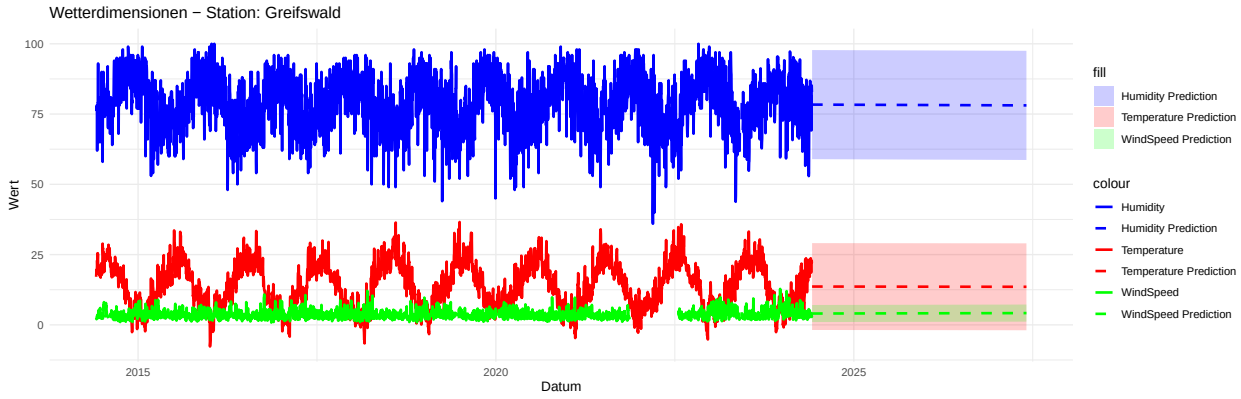
```
    labs(title = paste("Wetterdimensionen - Station:", unique(data$SDO_Name)),
         x = "Datum", y = "Wert") +
    scale_color_manual(values = c("Temperature" = "red", "Humidity" = "blue",
                                  "WindSpeed" = "green",
                                  "Temperature Prediction" = "red",
                                  "Humidity Prediction" = "blue",
                                  "WindSpeed Prediction" = "green"),
          labels = c("Temperature" = "Temperature",
                     "Humidity" = "Humidity",
                     "WindSpeed" = "WindSpeed",
                     "Temperature Prediction" = "Temperature Prediction",
                     "Humidity Prediction" = "Humidity Prediction",
                     "WindSpeed Prediction" = "WindSpeed Prediction")) +
    scale_fill_manual(values = c("Temperature Prediction" = "red",
                                 "Humidity Prediction" = "blue",
                                 "WindSpeed Prediction" = "green"),
          labels = c("Temperature Prediction" = "Temperature Prediction",
                     "Humidity Prediction" = "Humidity Prediction",
                     "WindSpeed Prediction" = "WindSpeed Prediction")) +
    theme_minimal()


}

# Plots für jede Station erstellen und anzeigen
plots <- lapply(combined_data, plot_station_data)
```

```
# Darstellen aller Wettervorhersagen als Plots
plots
```

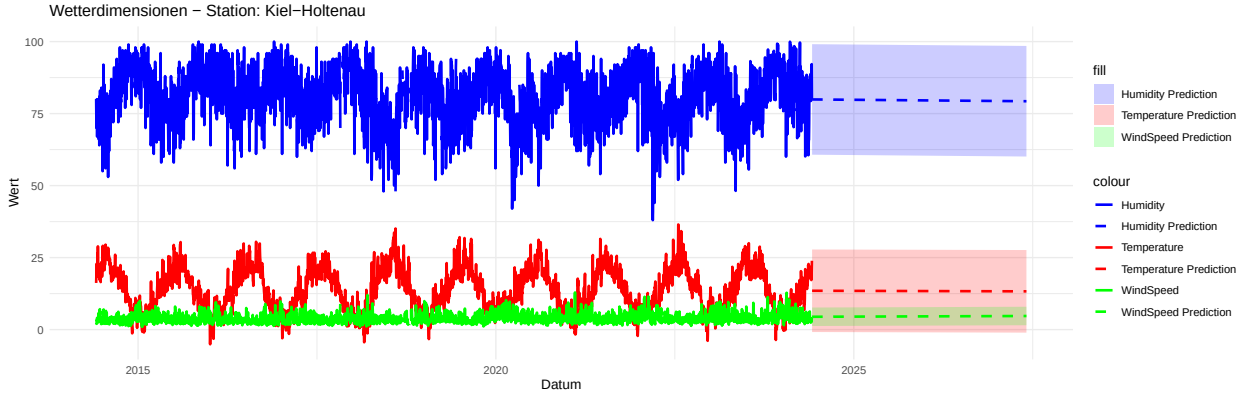
```
## [[1]]
```




```
##
## [[2]]
```


Wetterdimensionen − Station: Greifswald
Wert
Datum
100
75
50
25
0
2015
2020
2025
fill
Humidity Prediction
Temperature Prediction
WindSpeed Prediction
colour
Humidity
Humidity Prediction
Temperature
Temperature Prediction
WindSpeed
WindSpeed Prediction

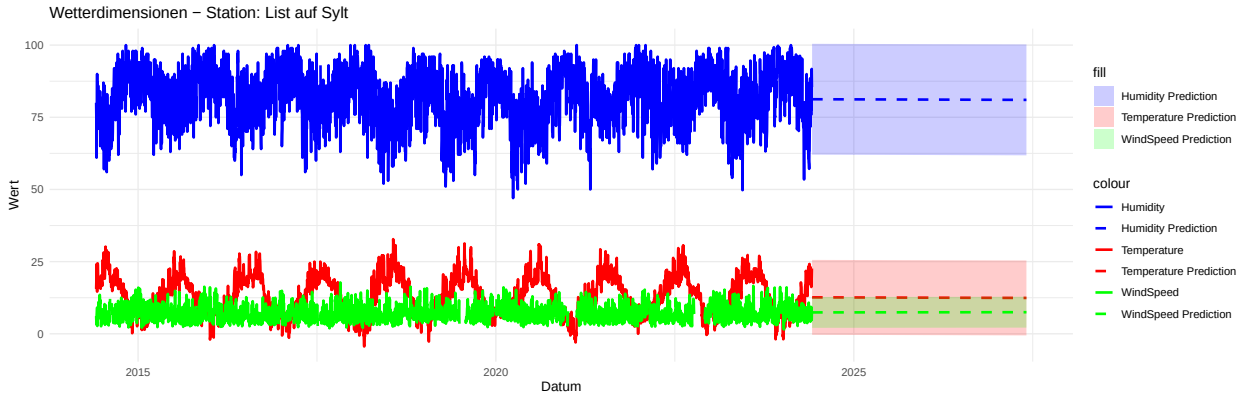
##
[[3]]


Wetterdimensionen − Station: Kiel−Holtenau
Wert
Datum
100
75
50
25
0
2015
2020
2025
fill
Humidity Prediction
Temperature Prediction
WindSpeed Prediction
colour
Humidity
Humidity Prediction
Temperature
Temperature Prediction
WindSpeed
WindSpeed Prediction

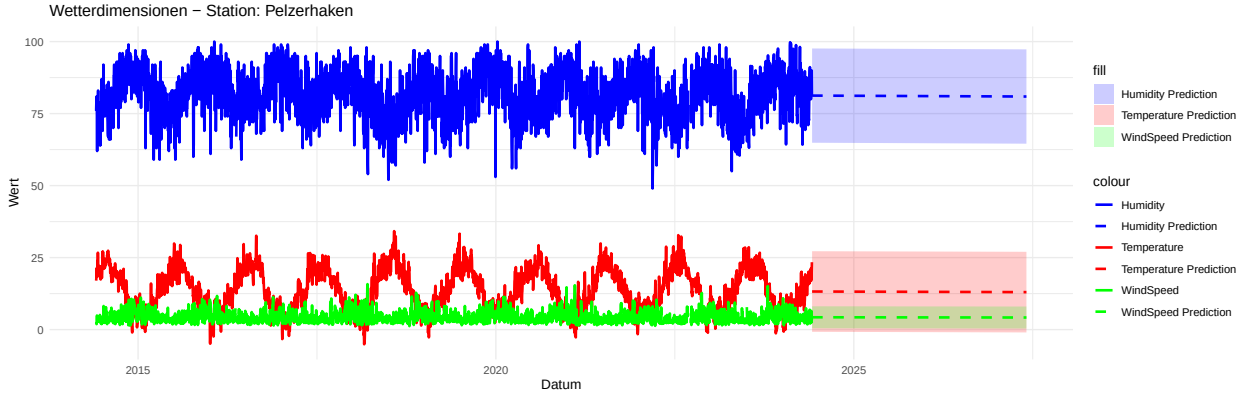
##
[[4]]


Wetterdimensionen − Station: Barth
Wert
Datum
100
75
50
25
0
2015
2020
2025
fill
Humidity Prediction
Temperature Prediction
WindSpeed Prediction
colour
Humidity
Humidity Prediction
Temperature
Temperature Prediction
WindSpeed
WindSpeed Prediction

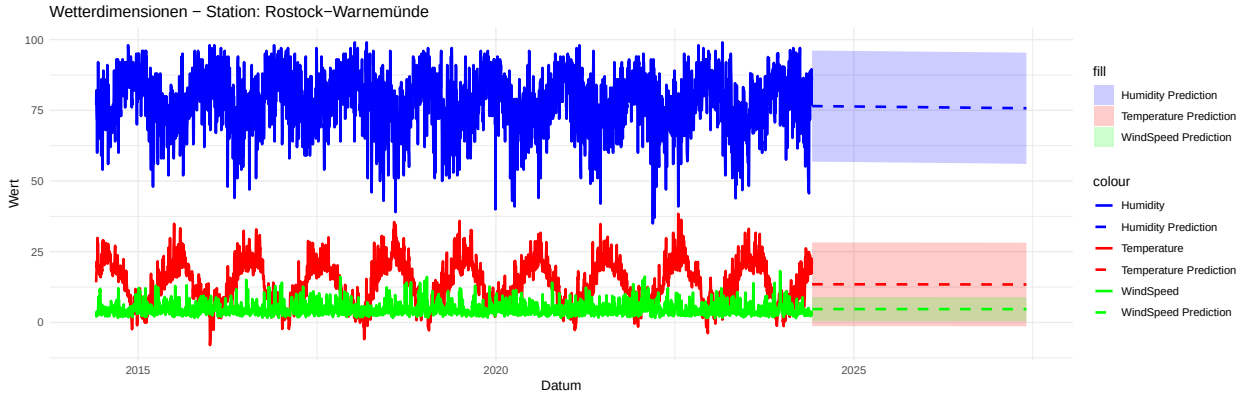
##
[[5]]


Wetterdimensionen − Station: List auf Sylt
Wert
Datum
100
75
50
25
0
2015
2020
2025
fill
Humidity Prediction
Temperature Prediction
WindSpeed Prediction
colour
Humidity
Humidity Prediction
Temperature
Temperature Prediction
WindSpeed
WindSpeed Prediction

##
[[6]]

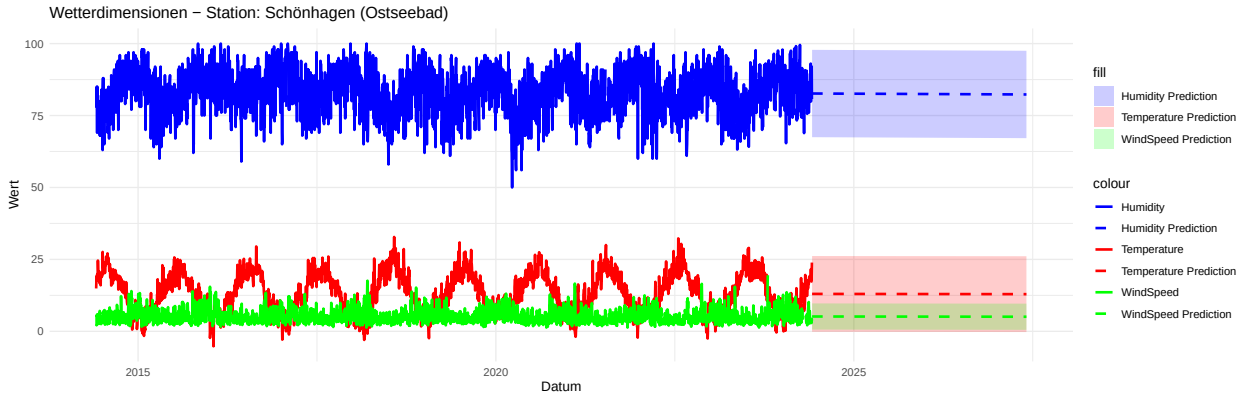

Wetterdimensionen − Station: Pelzerhaken
Wert
Datum
100
75
50
25
0
2015
2020
2025
fill
Humidity Prediction
Temperature Prediction
WindSpeed Prediction
colour
Humidity
Humidity Prediction
Temperature
Temperature Prediction
WindSpeed
WindSpeed Prediction

##
[[7]]


Wetterdimensionen − Station: Rostock−Warnemünde
Wert
Datum
100
75
50
25
0
2015
2020
2025
fill
Humidity Prediction
Temperature Prediction
WindSpeed Prediction
colour
Humidity
Humidity Prediction
Temperature
Temperature Prediction
WindSpeed
WindSpeed Prediction

##
[[8]]

##
[[9]]



##
[[10]]


```r
# Die ersten 10 Zeilen der Wettervorhersage von Schönhagen numerisch
head(combined_predictions[[9]], 10)
```

```
##         Date Predicted_Temperature Temperature_Upper Temperature_Lower
## 1 2024-06-02             13.00830          26.17660        -0.1600045
```

```
## 2  2024-06-03             13.00822          26.17653        -0.1600881
## 3  2024-06-04             13.00814          26.17645        -0.1601716
## 4  2024-06-05             13.00806          26.17638        -0.1602552
## 5  2024-06-06             13.00799          26.17631        -0.1603388
## 6  2024-06-07             13.00791          26.17624        -0.1604223
## 7  2024-06-08             13.00783          26.17617        -0.1605059
## 8  2024-06-09             13.00775          26.17609        -0.1605895
## 9  2024-06-10             13.00767          26.17602        -0.1606731
## 10 2024-06-11             13.00760          26.17595        -0.1607567
##    Predicted_Humidity Humidity_Upper Humidity_Lower Predicted_WindSpeed
## 1            82.64973       97.82891       67.47056            5.156362
## 2            82.64943       97.82861       67.47025            5.156287
## 3            82.64913       97.82832       67.46994            5.156211
## 4            82.64883       97.82803       67.46963            5.156136
## 5            82.64853       97.82773       67.46933            5.156060
## 6            82.64823       97.82744       67.46902            5.155984
## 7            82.64793       97.82714       67.46871            5.155909
## 8            82.64763       97.82685       67.46840            5.155833
## 9            82.64733       97.82656       67.46810            5.155757
## 10           82.64703       97.82626       67.46779            5.155682
##    WindSpeed_Upper WindSpeed_Lower              SDO_Name
## 1         9.704886       0.6078391 Schönhagen (Ostseebad)
## 2         9.704812       0.6077614 Schönhagen (Ostseebad)
## 3         9.704739       0.6076837 Schönhagen (Ostseebad)
## 4         9.704665       0.6076060 Schönhagen (Ostseebad)
## 5         9.704592       0.6075283 Schönhagen (Ostseebad)
## 6         9.704518       0.6074506 Schönhagen (Ostseebad)
## 7         9.704444       0.6073729 Schönhagen (Ostseebad)
## 8         9.704371       0.6072952 Schönhagen (Ostseebad)
## 9         9.704297       0.6072175 Schönhagen (Ostseebad)
## 10        9.704224       0.6071397 Schönhagen (Ostseebad)
```