Report

Contents

LO1: Visualization basics, chart types

In diesem Kapitel setze ich mit den Grundlagen der Datenvisualisierungen auseinander. Dazu werde ich Visualisierungen erstellen und ein paar Grundsätze erläutern. Als Datensatz verwende ich die Wetterdaten von der Wetterstation Mythenquai der Seepolizei zürich aus der Wettermonitor-Challenge, welche ich letztes Jahr absolviert habe. Die Daten sind hier zu finden: [https://data.stadt-zuerich.ch/dataset/sid_wapo_wetterstationen] (https://data.stadt-zuerich.ch/dataset/sid_wapo_wetterstationen).

Der Auftraggebber dieser Challenge war der Segelclub Zürich. Die Fragen sollten so sein, dass sie vom Auftraggeber hätten kommen können.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attache Paket: 'lubridate'

## Die folgenden Objekte sind maskiert von 'package:base':
##
##     date, intersect, setdiff, union
```

```
mythenquai_2007_2021 <- read.csv("messwerte_mythenquai_2007-2021.csv")
```

```
mythenquai_2007_2021 %>%
  sample_n(10)
```

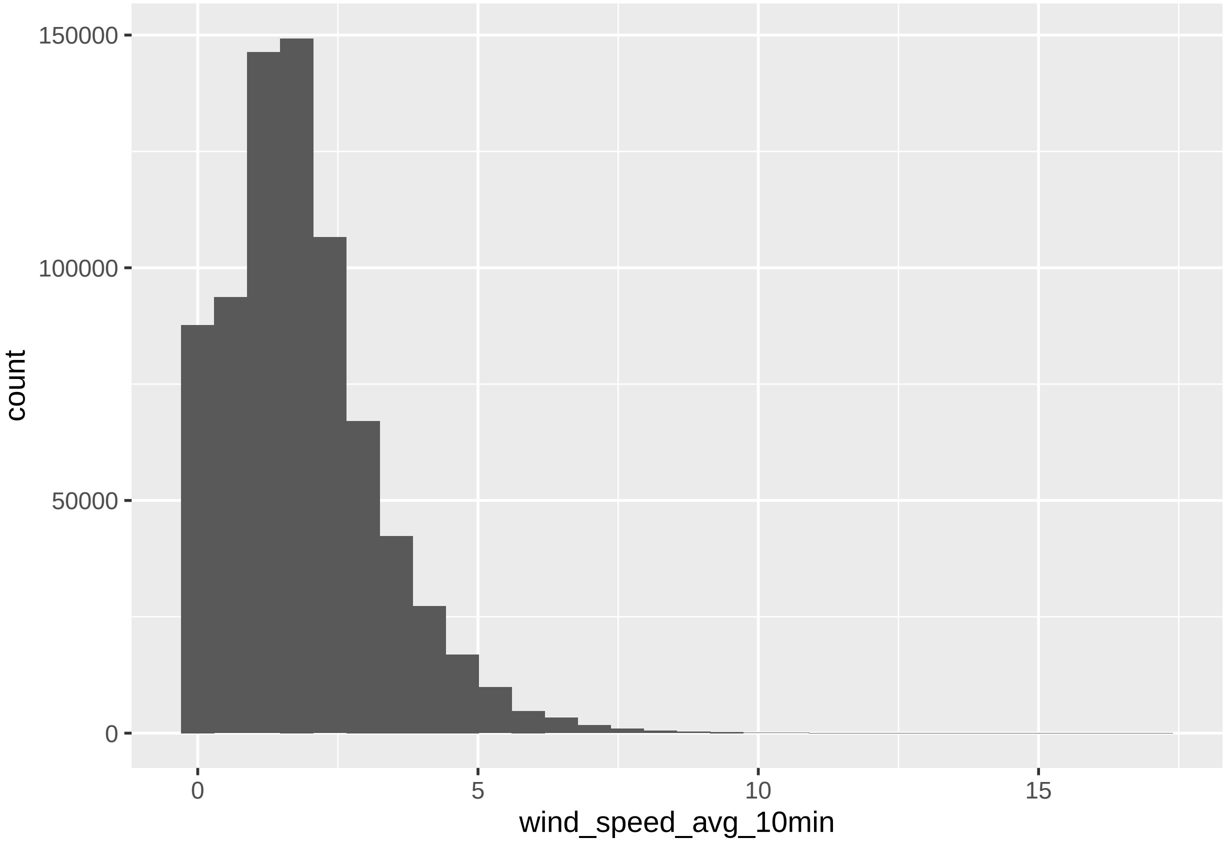
```
##                  timestamp_utc             timestamp_cet air_temperature
## 1  2020-11-07T00:10:00+00:00 2020-11-07T01:10:00+01:00             7.9
## 2  2014-07-21T15:30:00+00:00 2014-07-21T17:30:00+02:00            20.4
## 3  2007-07-24T09:00:00+00:00 2007-07-24T11:00:00+02:00            17.3
## 4  2020-01-16T20:40:00+00:00 2020-01-16T21:40:00+01:00             3.0
## 5  2019-06-26T21:10:00+00:00 2019-06-26T23:10:00+02:00            26.0
## 6  2011-11-12T15:30:00+00:00 2011-11-12T16:30:00+01:00            10.8
## 7  2019-12-12T03:50:00+00:00 2019-12-12T04:50:00+01:00             3.6
## 8  2009-02-09T23:40:00+00:00 2009-02-10T00:40:00+01:00             3.2
## 9  2009-11-24T21:00:00+00:00 2009-11-24T22:00:00+01:00            11.3
## 10 2014-02-08T23:40:00+00:00 2014-02-09T00:40:00+01:00             5.7
##    water_temperature wind_gust_max_10min wind_speed_avg_10min
## 1                 NA                 3.1                  2.3
## 2               21.3                 4.1                  1.8
## 3               20.1                11.9                  4.9
## 4                 NA                 0.8                  0.2
## 5               24.6                 0.0                  0.0
## 6               11.8                 2.5                  1.3
## 7                7.9                 5.1                  2.6
## 8                4.3                 5.7                  4.0
## 9               10.2                 2.9                  0.2
## 10               5.2                 7.8                  3.2
##    wind_force_avg_10min wind_direction windchill barometric_pressure_qfe
## 1                   2.0             66       6.2                  1030.0
## 2                   1.8            253      19.9                   966.0
## 3                   4.9            280      10.5                   962.0
## 4                   0.0             49       3.0                      NA
## 5                   0.0              0      26.0                   974.6
## 6                   1.3            348      10.9                   982.6
## 7                   2.0             88       0.9                   959.2
## 8                   4.0            165      -3.3                   955.7
## 9                   0.2            213      11.3                   974.1
## 10                  3.2            216       1.9                   954.1
##    precipitation dew_point global_radiation humidity water_level
## 1             NA       6.8               NA       93          NA
## 2              0      14.9              199       71      406.05
## 3              0       9.9               44       62      406.07
## 4             NA       1.5               NA       90          NA
## 5              0      20.1                0       70      406.07
## 6              0       6.3               17       73      405.90
## 7              0       1.5                0       86      405.94
## 8              0       1.3                3       87      405.67
## 9              0       5.2                5       66      405.94
## 10             0       2.3                1       79      405.68
```

```
summary(mythenquai_2007_2021)
```

```
##  timestamp_utc      timestamp_cet      air_temperature  water_temperature
##  Length:759119      Length:759119      Min.   :-13.40   Min.   : 2.40  
##  Class :character   Class :character   1st Qu.:  5.20   1st Qu.: 6.40  
##  Mode  :character   Mode  :character   Median : 11.30   Median :13.10  
##                                        Mean   : 11.53   Mean   :13.42  
##                                        3rd Qu.: 17.50   3rd Qu.:19.60  
##                                        Max.   : 37.70   Max.   :28.00  
##                                                         NA’s   :100397 
##  wind_gust_max_10min wind_speed_avg_10min wind_force_avg_10min wind_direction
##  Min.   :-0.100      Min.   : 0.000       Min.   : 0.000       Min.   :  0.0  
##  1st Qu.: 1.700      1st Qu.: 0.900       1st Qu.: 1.000       1st Qu.:103.0  
##  Median : 2.900      Median : 1.600       Median : 1.700       Median :176.0  
##  Mean   : 3.521      Mean   : 1.854       Mean   : 1.778       Mean   :184.5  
##  3rd Qu.: 4.700      3rd Qu.: 2.600       3rd Qu.: 2.400       3rd Qu.:286.0  
##  Max.   :32.000      Max.   :17.100       Max.   :16.800       Max.   :360.0  
##                                                                               
##    windchill      barometric_pressure_qfe precipitation      dew_point     
##  Min.   :-25.60   Min.   : 930.7          Min.   : 0.00    Min.   :-17.200  
##  1st Qu.:  3.70   1st Qu.: 966.1          1st Qu.: 0.00    1st Qu.:  1.900  
##  Median : 10.30   Median : 970.9          Median : 0.00    Median :  6.800  
##  Mean   : 10.38   Mean   : 975.6          Mean   : 0.02    Mean   :  6.797  
##  3rd Qu.: 16.80   3rd Qu.: 977.2          3rd Qu.: 0.00    3rd Qu.: 12.100  
##  Max.   : 37.80   Max.   :1037.5          Max.   :17.00    Max.   : 24.600  
##                   NA’s   :4741            NA’s   :100397                    
##  global_radiation    humidity       water_level   
##  Min.   :   0.0   Min.   : 16.00   Min.   :405.2  
##  1st Qu.:   0.0   1st Qu.: 65.00   1st Qu.:405.9  
##  Median :   7.0   Median : 79.00   Median :405.9  
##  Mean   : 137.7   Mean   : 75.15   Mean   :405.9  
##  3rd Qu.: 161.0   3rd Qu.: 87.00   3rd Qu.:406.0  
##  Max.   :4293.0   Max.   :100.00   Max.   :406.5  
##  NA’s   :100397                    NA’s   :100397 
```

```
mythenquai_2007_2021 %>%
  ggplot(aes(x=wind_speed_avg_10min)) +
  geom_histogram()
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```
mythenquai_2007_2021 <- mythenquai_2007_2021 %>%
  mutate(timestamp_utc = as.POSIXct(timestamp_utc, format="%Y-%m-%dT%H:%M:%S", tz="UTC")) %>%
  select(-timestamp_cet)
```

```
mythenquai_2007_2021 %>%
  filter(is.na(water_temperature)) %>%
  summarise(min(timestamp_utc))
```

```
##    min(timestamp_utc)
## 1 2019-12-31 23:00:00
```

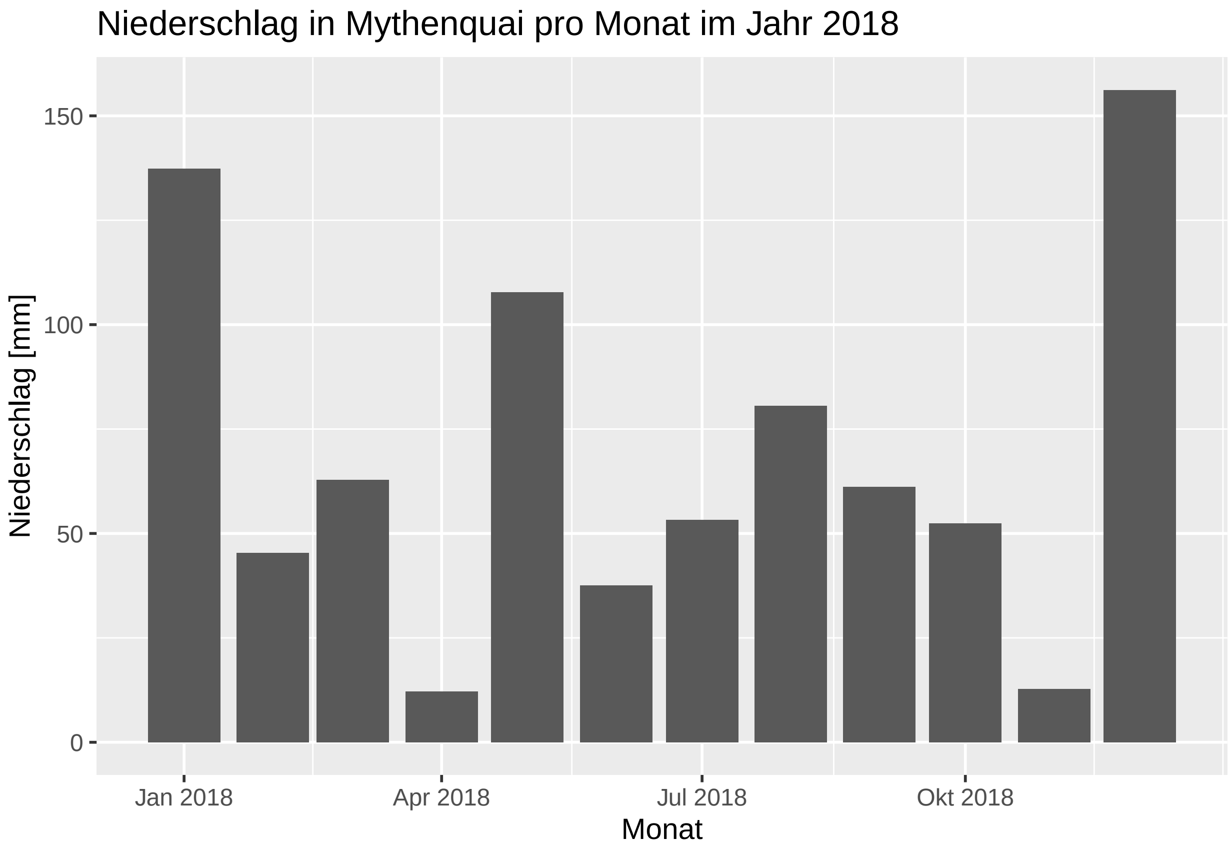
Seit Anfangs 2020 wird bei der Seepolizei gebaut, und deshalb sind folgende Messwerte nicht verfügbar: - Wassertemperatur - Strahlung - Niederschlag - Seespiegel

Deshalb werde ich nicht immer mit dem ganzen Zeitraum arbeiten können.

Fragestellungen

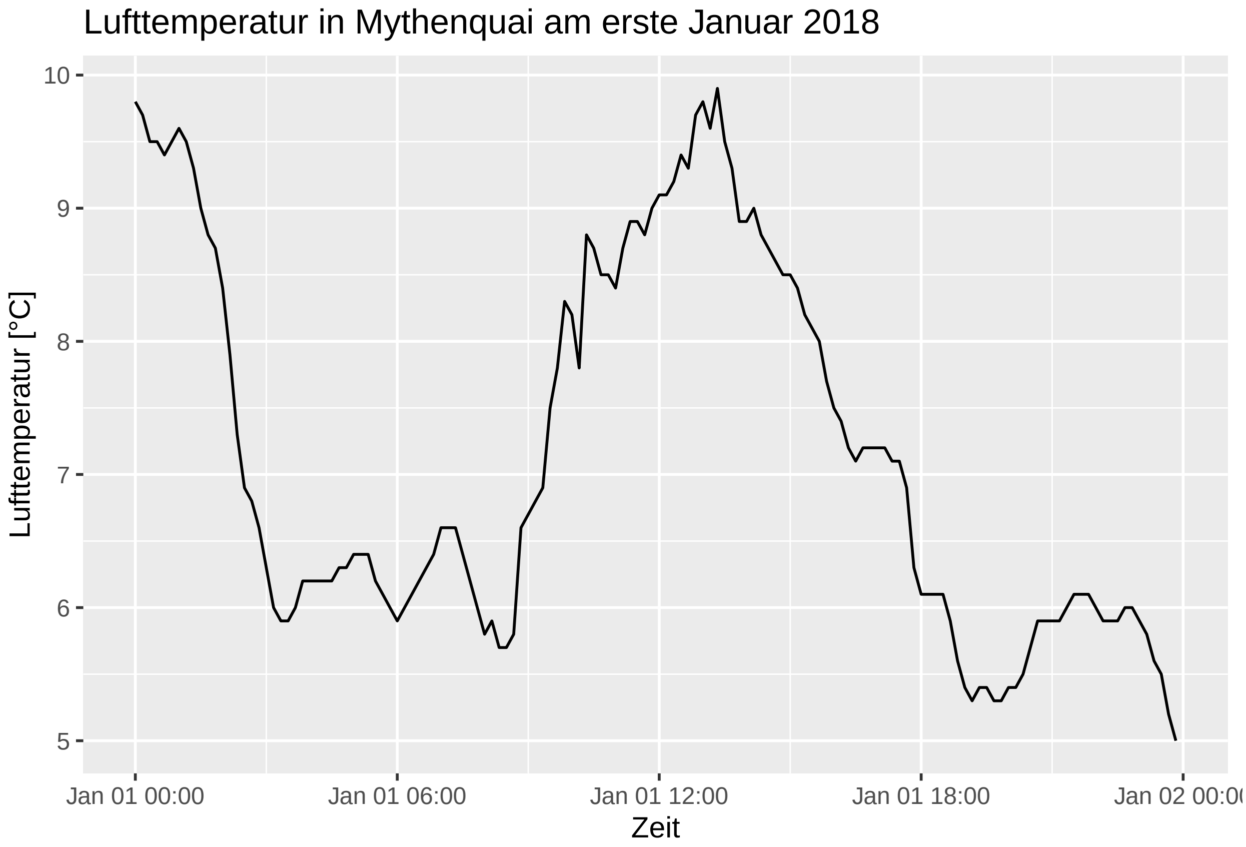
Wieviel hat es jeden Monat geregnet?

```
jan_2018 <- as.POSIXct("2018-01-01 00:00:00", tz="UTC")
jan_2019 <- as.POSIXct("2019-01-01 00:00:00", tz="UTC")
mythenquai_2007_2021 %>%
  select(timestamp_utc, precipitation) %>%
  filter(timestamp_utc >= jan_2018 & timestamp_utc < jan_2019) %>%
  group_by(month = lubridate::floor_date(timestamp_utc, "month")) %>%
  summarise(total_precipitation = sum(precipitation)) %>%
  ggplot(aes(x=month, y=total_precipitation)) +
    geom_bar(stat="identity") +
    labs(x="Monat", y="Niederschlag [mm]", title="Niederschlag in Mythenquai pro Monat im Jahr 2018")
```



Wie warm war es an Tag x?

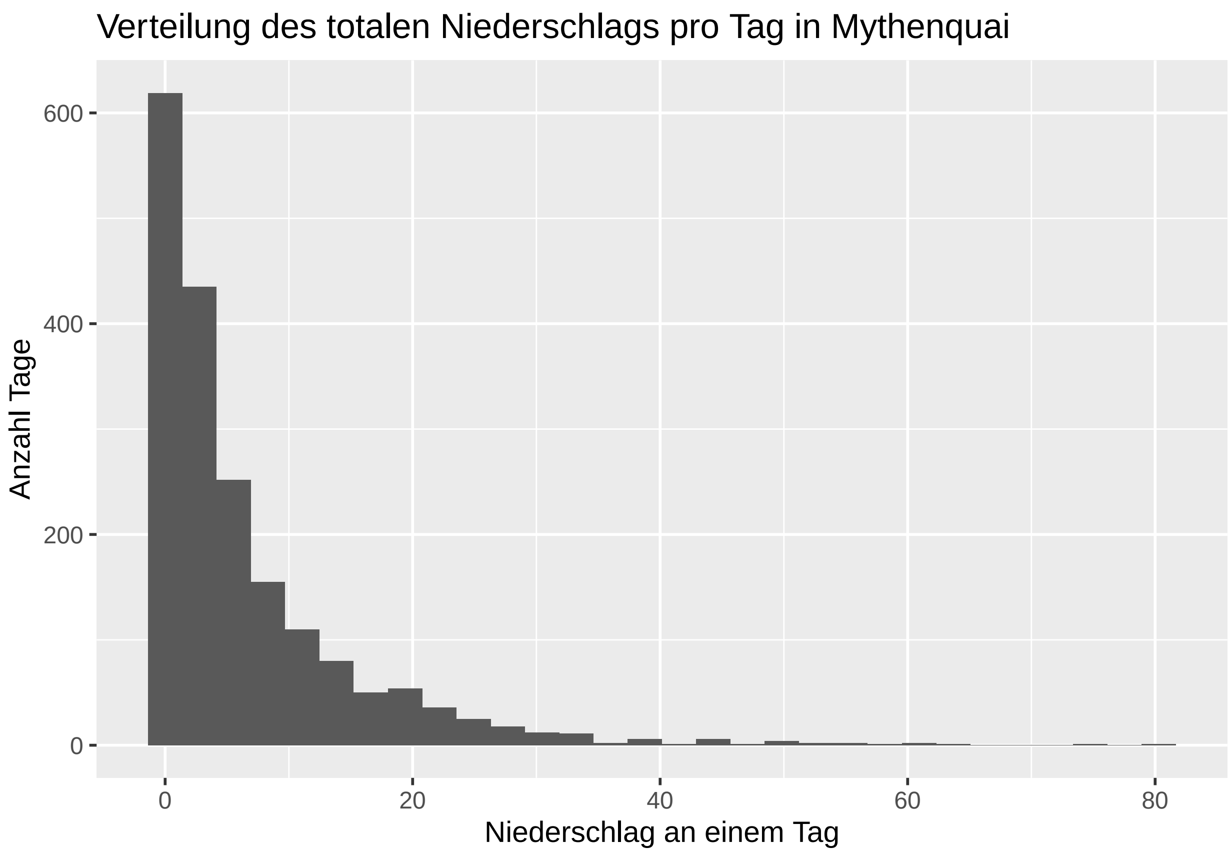
```
first_jan_2020 <- as.POSIXct("2018-01-01 00:00:00", tz="UTC")
second_jan_2020 <- as.POSIXct("2018-01-02 00:00:00", tz="UTC")

mythenquai_2007_2021 %>%
  select(timestamp_utc, air_temperature) %>%
  filter(timestamp_utc >= first_jan_2020 & timestamp_utc < second_jan_2020) %>%
  ggplot(aes(x=timestamp_utc, y=air_temperature)) +
    geom_line() +
    labs(x="Zeit", y="Lufttemperatur [°C]", title="Lufttemperatur in Mythenquai am erste Januar 2018")
```



Wie ist die Verteilung des Niederschlags im Jahr 2018?

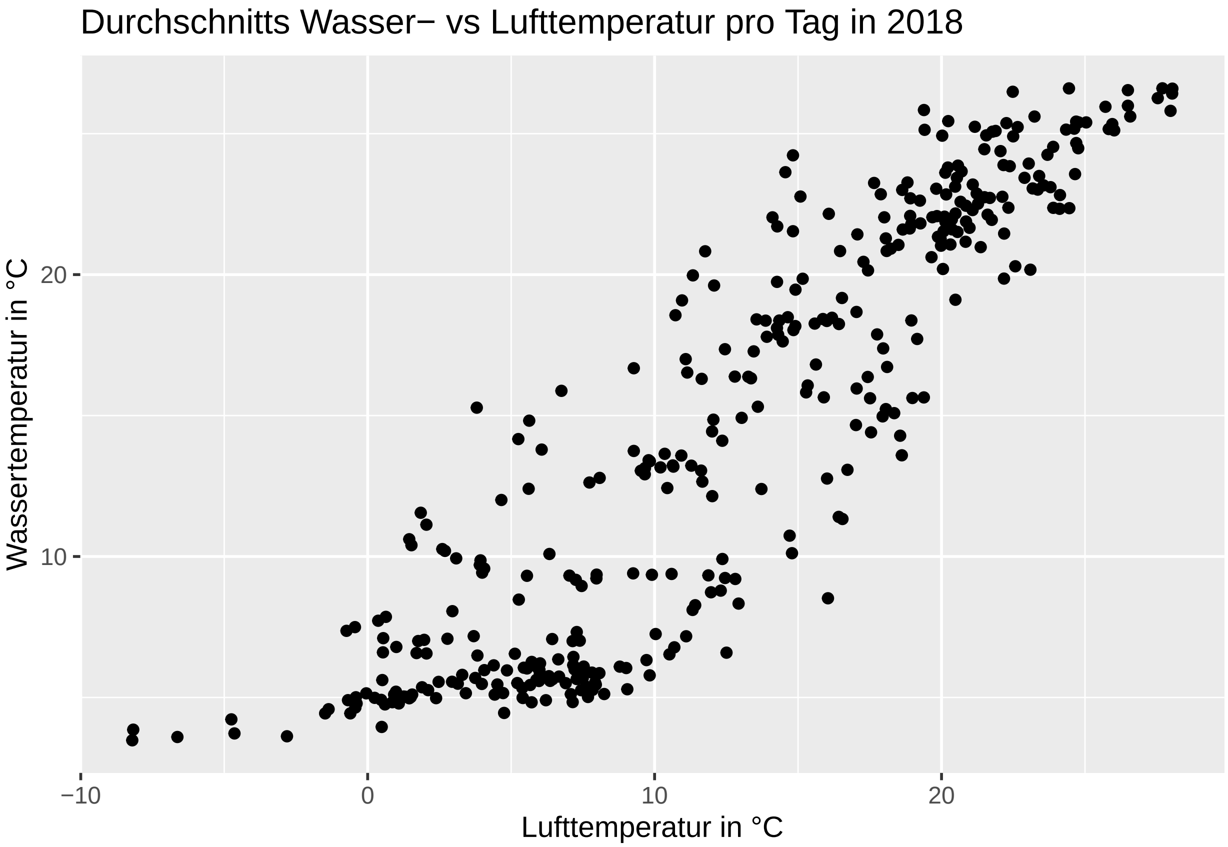
```
jan_2018 <- as.POSIXct("2007-01-01 00:00:00", tz="UTC")
jan_2019 <- as.POSIXct("2019-01-01 00:00:00", tz="UTC")
mythenquai_2007_2021 %>%
  select(timestamp_utc, precipitation) %>%
  filter(timestamp_utc >= jan_2018 & timestamp_utc < jan_2019) %>%
  group_by(day = lubridate::floor_date(timestamp_utc, "day")) %>%
  summarise(total_precipitation = sum(precipitation)) %>%
  filter(total_precipitation > 0) %>%
  ggplot(aes(x=total_precipitation)) +
    geom_histogram() +
    labs(x="Niederschlag an einem Tag", y="Anzahl Tage", title="Verteilung des totalen Niederschlags pr
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



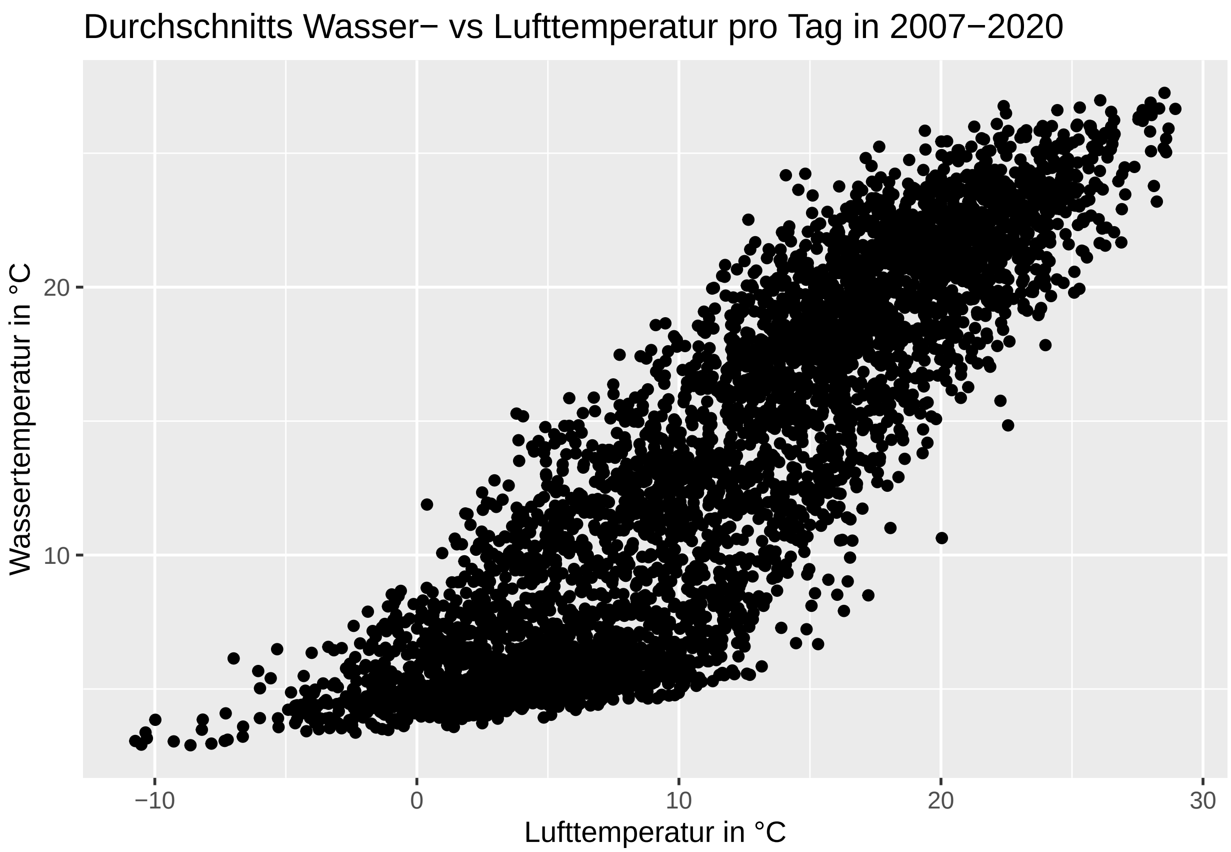
Hat die Lufttemperatur und Wassertemperatur einen Zusammenhang?

```
jan_2018 <- as.POSIXct("2018-01-01 00:00:00", tz="UTC")
jan_2019 <- as.POSIXct("2019-01-01 00:00:00", tz="UTC")
mythenquai_2007_2021 %>%
  select(timestamp_utc, air_temperature, water_temperature, water_level) %>%
  filter(timestamp_utc >= jan_2018 & timestamp_utc < jan_2019) %>%
  group_by(day = lubridate::floor_date(timestamp_utc, "day")) %>%
  summarise(mean_air_temperature = mean(air_temperature), mean_water_temperature = mean(water_temperatu
  ggplot(aes(x=mean_air_temperature, y=mean_water_temperature)) +
    geom_point() +
    labs(x="Lufttemperatur in °C", y="Wassertemperatur in °C", title="Durchschnitts Wasser- vs Lufttempe
```



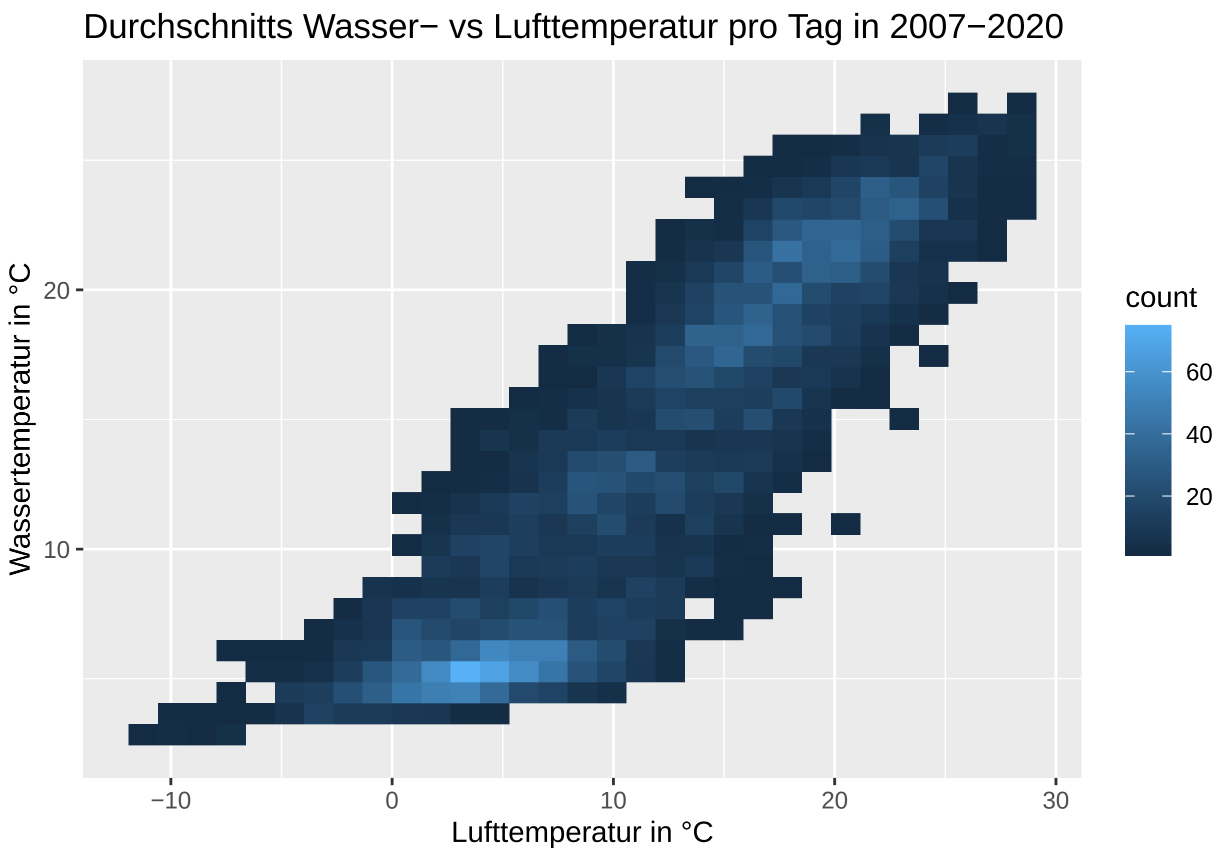
```
jan_2007 <- as.POSIXct("2007-01-01 00:00:00", tz="UTC")
jan_2020 <- as.POSIXct("2020-01-01 00:00:00", tz="UTC")
mythenquai_2007_2021 %>%
  select(timestamp_utc, air_temperature, water_temperature, water_level) %>%
  filter(timestamp_utc >= jan_2007 & timestamp_utc < jan_2020) %>%
  group_by(day = lubridate::floor_date(timestamp_utc, "day")) %>%
  summarise(mean_air_temperature = mean(air_temperature), mean_water_temperature = mean(water_temperatu
  ggplot(aes(x=mean_air_temperature, y=mean_water_temperature)) +
    geom_point() +
    labs(x="Lufttemperatur in °C", y="Wassertemperatur in °C", title="Durchschnitts Wasser- vs Lufttempe
```

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



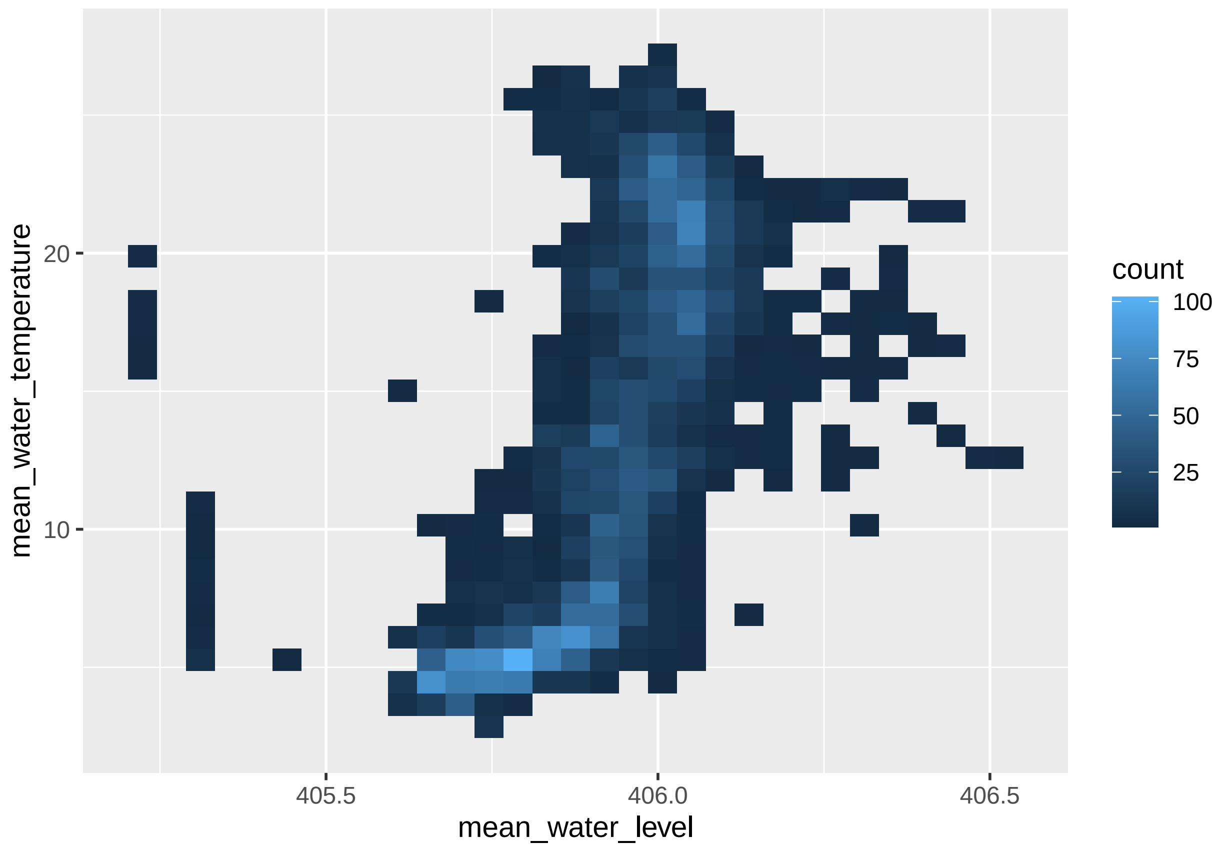
```
jan_2018 <- as.POSIXct("2007-01-01 00:00:00", tz="UTC")
jan_2019 <- as.POSIXct("2020-01-01 00:00:00", tz="UTC")
mythenquai_2007_2021 %>%
  select(timestamp_utc, air_temperature, water_temperature, water_level) %>%
  filter(timestamp_utc >= jan_2018 & timestamp_utc < jan_2019) %>%
  group_by(day = lubridate::floor_date(timestamp_utc, "day")) %>%
  summarise(mean_air_temperature = mean(air_temperature), mean_water_temperature = mean(water_temperatu
  ggplot(aes(x=mean_air_temperature, y=mean_water_temperature)) +
    geom_bin2d() +
    labs(x="Lufttemperatur in °C", y="Wassertemperatur in °C", title="Durchschnitts Wasser- vs Lufttempe
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin2d).
```



```
jan_2018 <- as.POSIXct("2007-01-01 00:00:00", tz="UTC")
jan_2019 <- as.POSIXct("2020-01-01 00:00:00", tz="UTC")
mythenquai_2007_2021 %>%
  select(timestamp_utc, air_temperature, water_temperature, water_level) %>%
  filter(timestamp_utc >= jan_2018 & timestamp_utc < jan_2019) %>%
  group_by(day = lubridate::floor_date(timestamp_utc, "day")) %>%
  summarise(mean_water_level = mean(water_level), mean_water_temperature = mean(water_temperature)) %>%
  ggplot(aes(x=mean_water_level, y=mean_water_temperature)) +
    geom_bin2d()
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin2d).
```



Von wo hat der Wind im letzten Jahr geweht?

```
# From: https://stackoverflow.com/a/17266781
# WindRose.R
require(ggplot2)
require(RColorBrewer)
```

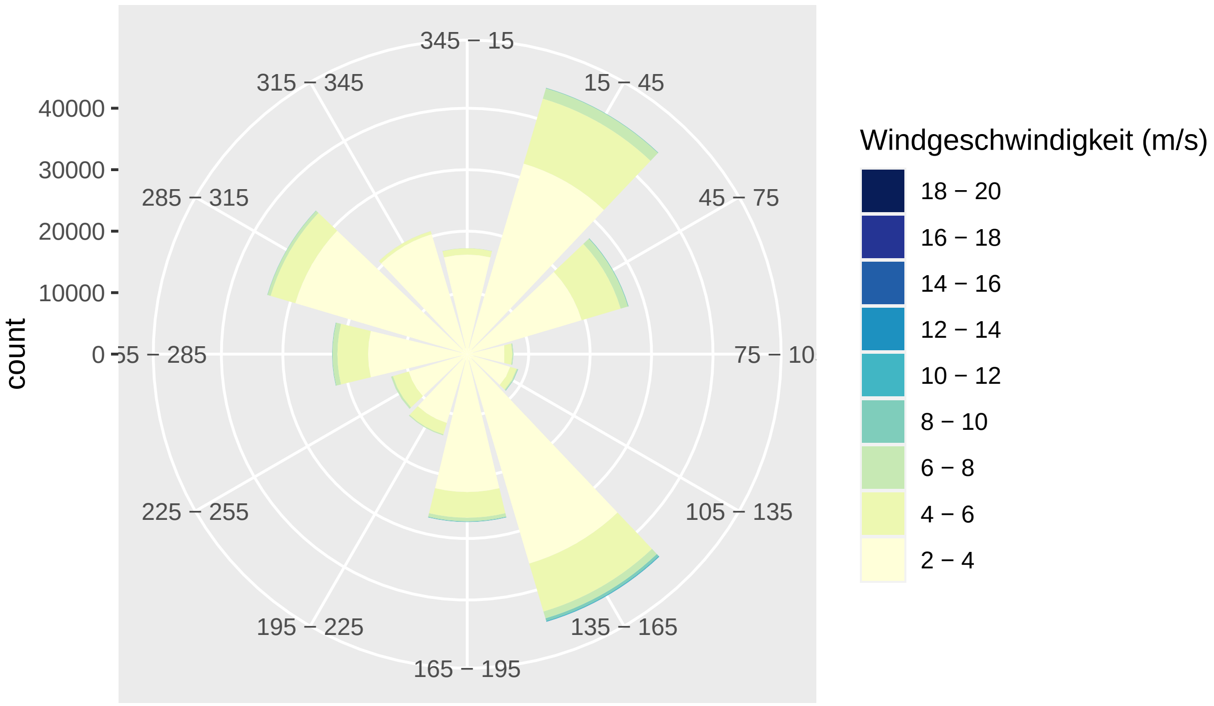
```
## Lade nötiges Paket: RColorBrewer
```

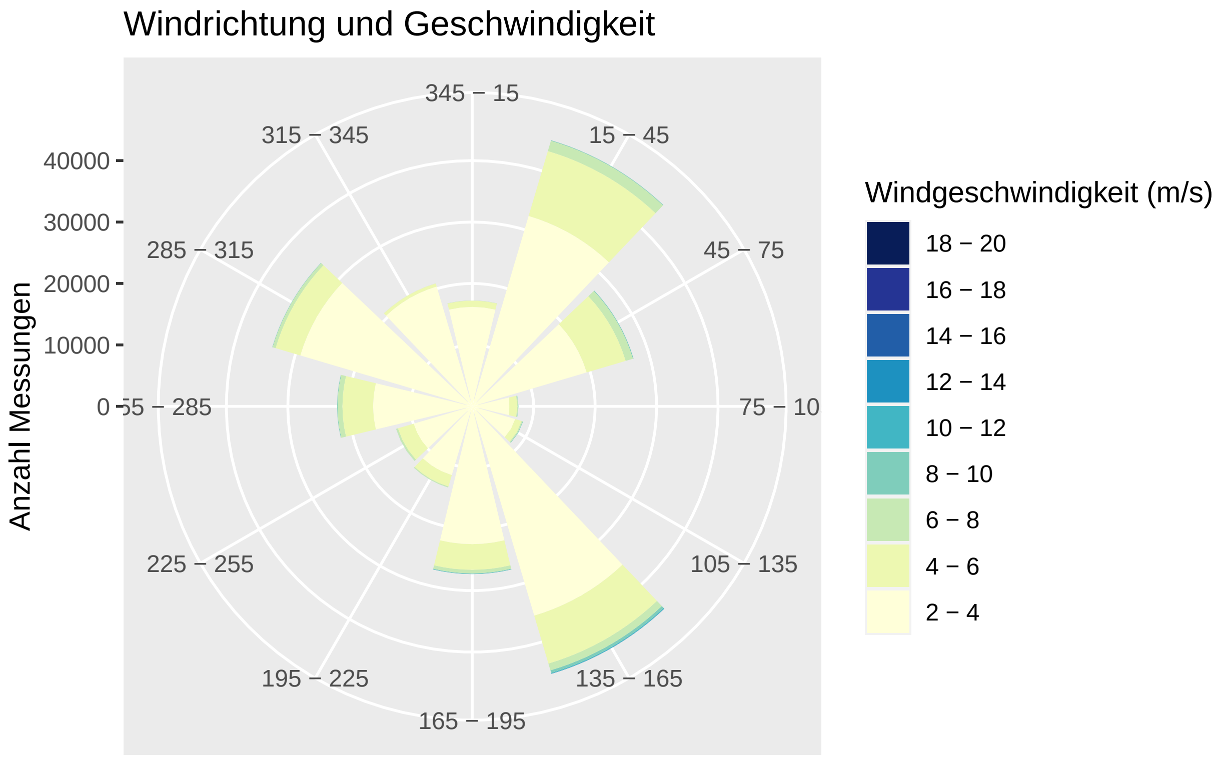
```
plot.windrose <- function(data,
                      spd,
                      dir,
                      spdres = 2,
                      dirres = 30,
                      spdmin = 2,
                      spdmax = 20,
                      spdseq = NULL,
                      palette = "YlGnBu",
                      countmax = NA,
                      debug = 0){


# Look to see what data was passed in to the function
```

```r
if (is.numeric(spd) & is.numeric(dir)){
  # assume that we've been given vectors of the speed and direction vectors
  data <- data.frame(spd = spd,
                     dir = dir)
  spd = "spd"
  dir = "dir"
} else if (exists("data")){
  # Assume that we've been given a data frame, and the name of the speed
  # and direction columns. This is the format we want for later use.
}

# Tidy up input data ----
n.in <- NROW(data)
dnu <- (is.na(data[[spd]]) | is.na(data[[dir]]))
data[[spd]][dnu] <- NA
data[[dir]][dnu] <- NA

# figure out the wind speed bins ----
if (missing(spdseq)){
  spdseq <- seq(spdmin,spdmax,spdres)
} else {
  if (debug >0){
    cat("Using custom speed bins \n")
  }
}
# get some information about the number of bins, etc.
n.spd.seq <- length(spdseq)
n.colors.in.range <- n.spd.seq - 1

# create the color map
spd.colors <- colorRampPalette(brewer.pal(min(max(3,
                                                  n.colors.in.range),
                                              min(9,
                                                  n.colors.in.range)),
                                          palette))(n.colors.in.range)

if (max(data[[spd]],na.rm = TRUE) > spdmax){
  spd.breaks <- c(spdseq,
                  max(data[[spd]],na.rm = TRUE))
  spd.labels <- c(paste(c(spdseq[1:n.spd.seq-1]),
                        '-',
                        c(spdseq[2:n.spd.seq])),
                  paste(spdmax,
                        "-",
                        max(data[[spd]],na.rm = TRUE)))
  spd.colors <- c(spd.colors, "grey50")
} else{
  spd.breaks <- spdseq
  spd.labels <- paste(c(spdseq[1:n.spd.seq-1]),
                      '-',
                      c(spdseq[2:n.spd.seq]))
}
data$spd.binned <- cut(x = data[[spd]],
                       breaks = spd.breaks,
                       labels = spd.labels,
                       ordered_result = TRUE)
# clean up the data
data. <- na.omit(data)

# figure out the wind direction bins
dir.breaks <- c(-dirres/2,
                seq(dirres/2, 360-dirres/2, by = dirres),
                360+dirres/2)
dir.labels <- c(paste(360-dirres/2,"-",dirres/2),
                paste(seq(dirres/2, 360-3*dirres/2, by = dirres),
                      "-",
                      seq(3*dirres/2, 360-dirres/2, by = dirres)),
                paste(360-dirres/2,"-",dirres/2))
# assign each wind direction to a bin
dir.binned <- cut(data[[dir]],
                  breaks = dir.breaks,
                  ordered_result = TRUE)
levels(dir.binned) <- dir.labels
data$dir.binned <- dir.binned

# Run debug if required ----
if (debug>0){
  cat(dir.breaks,"\n")
  cat(dir.labels,"\n")
  cat(levels(dir.binned),"\n")
}

# deal with change in ordering introduced somewhere around version 2.2
if(packageVersion("ggplot2") > "2.2"){
  cat("Hadley broke my code\n")
  data$spd.binned = with(data, factor(spd.binned, levels = rev(levels(spd.binned))))
  spd.colors = rev(spd.colors)
}

# create the plot ----
p.windrose <- ggplot(data = data,
                     aes(x = dir.binned,
                         fill = spd.binned)) +
  geom_bar() +
  scale_x_discrete(drop = FALSE,
                   labels = waiver()) +
  coord_polar(start = -((dirres/2)/360) * 2*pi) +
  scale_fill_manual(name = "Windgeschwindigkeit (m/s)",
                    values = spd.colors,
                    drop = FALSE) +
  theme(axis.title.x = element_blank())

# adjust axes if required
if (!is.na(countmax)){
  p.windrose <- p.windrose +
    ylim(c(0,countmax))
  }

  # print the plot
  print(p.windrose)

  # return the handle to the wind rose
  return(p.windrose)
}
```

```
mythenquai_2007_2021_no_wind_na <- mythenquai_2007_2021 %>% filter( wind_speed_avg_10min > 2)
plot.windrose(spd = mythenquai_2007_2021_no_wind_na$wind_speed_avg_10min,
                   dir = mythenquai_2007_2021_no_wind_na$wind_direction) + labs(y="Anzahl Messungen", ti
```

```
## Hadley broke my code
```





- Wie verändert sich der Wasserstand im Verlaufe eines Jahres? (Heatmap, 12 months, year)

Grundsätzliches

Pie und Donut sind scheisse, weil Winkel nicht gut. Nicht zu viele Variablen, bei z.B. stacked bar charts.