Exploration of NetCDF Data

Tobias Machnitzki and Finn Burgemeister

01 Dezember 2017

```r
library(knitr)
library(markdown)
library(ncdf4)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

1 Import Data

```r
ncfname = "../../WORK/Blatt6/weather-full.nc"

ncin = nc_open(ncfname)
print(ncin)
```

```
## File ../../WORK/Blatt6/weather-full.nc (NC_FORMAT_CLASSIC):
##
##      7 variables (excluding dimension variables):
##         short sf[longitude,latitude,time]
##             scale_factor: 7.3764124573405e-07
##             add_offset: 0.0241695530510217
##             _FillValue: -32767
##             missing_value: -32767
##             units: m of water equivalent
##             long_name: Snowfall
##             standard_name: lwe_thickness_of_snowfall_amount
##         short u10[longitude,latitude,time]
##             scale_factor: 0.00119632889000476
##             add_offset: -3.29942438209637
##             _FillValue: -32767
##             missing_value: -32767
##             units: m s**-1
##             long_name: 10 metre U wind component
##         short v10[longitude,latitude,time]
##             scale_factor: 0.00106014321386744
##             add_offset: 0.829670123705566
##             _FillValue: -32767
##             missing_value: -32767
##             units: m s**-1
##             long_name: 10 metre V wind component
##         short t2m[longitude,latitude,time]
##             scale_factor: 0.00203513170666401
##             add_offset: 257.975148205631
##             _FillValue: -32767
##             missing_value: -32767
##             units: K
##             long_name: 2 metre temperature
##         short e[longitude,latitude,time]
##             scale_factor: 3.68475681160163e-07
```

```
##             add_offset: -0.00823421201098244
##             _FillValue: -32767
##             missing_value: -32767
##             units: m of water equivalent
##             long_name: Evaporation
##             standard_name: lwe_thickness_of_water_evaporation_amount
##         short sund[longitude,latitude,time]
##             scale_factor: 0.659209863732776
##             add_offset: 21599.6703950681
##             _FillValue: -32767
##             missing_value: -32767
##             units: s
##             long_name: Sunshine duration
##         short tp[longitude,latitude,time]
##             scale_factor: 5.51507908704014e-06
##             add_offset: 0.180707081289331
##             _FillValue: -32767
##             missing_value: -32767
##             units: m
##             long_name: Total precipitation
##
##      3 dimensions:
##         longitude  Size:480
##             units: degrees_east
##             long_name: longitude
##         latitude  Size:241
##             units: degrees_north
##             long_name: latitude
##         time  Size:1096   *** is unlimited ***
##             units: hours since 1900-01-01 00:00:0.0
##             long_name: time
##             calendar: gregorian
##
##     2 global attributes:
##         Conventions: CF-1.0
##         history: 2015-06-03 08:02:17 GMT by grib_to_netcdf-1.13.1: grib_to_netcdf /data/data04/scratch/net
```

```
t2m = ncvar_get(ncin,"t2m")
tp = ncvar_get(ncin,"tp")
sund = ncvar_get(ncin,"sund")
lon = ncvar_get(ncin,"longitude")
lat = ncvar_get(ncin,"latitude")
time = ncvar_get(ncin,"time")

nc_close(ncin)
```

2 Limits of data

The file weather-nc-full contains datasets of three years: 1979, 1996, 2014 with daily data availiability.

```
timestamp = as.POSIXct("1900-01-01 00:00")+as.difftime(time,units="hours")

years = unique(as.integer(format(timestamp, "%Y")))

yearstep_begin = rep(0,3)
yearstep_end = rep(0,3)
i = 1
for(year in years){
  yearstep_begin[i] = min(which(grepl(year, format(timestamp, "%Y"))))
  yearstep_end[i] = max(which(grepl(year, format(timestamp, "%Y"))))
  print(c(timestamp[yearstep_begin[i]], timestamp[yearstep_end[i]]))
  i = i+1
}
```

```
## [1] "1979-01-01 12:00:00 CET" "1979-12-31 12:00:00 CET"
## [1] "1996-01-01 12:00:00 CET" "1996-12-31 12:00:00 CET"
## [1] "2014-01-01 12:00:00 CET" "2014-12-31 12:00:00 CET"
```

The longitude is subdivided in steps of 0.75 and has the range from 0.00 to 395.25. The latitude is subdivided in steps of 0.75 and has the range from 90.00 to -90.00.

```
print(head((lon)))
```

```
## [1] 0.00 0.75 1.50 2.25 3.00 3.75
```

```
print(tail((lon)))
```

```
## [1] 355.50 356.25 357.00 357.75 358.50 359.25
```

```
print(head((lat)))
```

```
## [1] 90.00 89.25 88.50 87.75 87.00 86.25
```

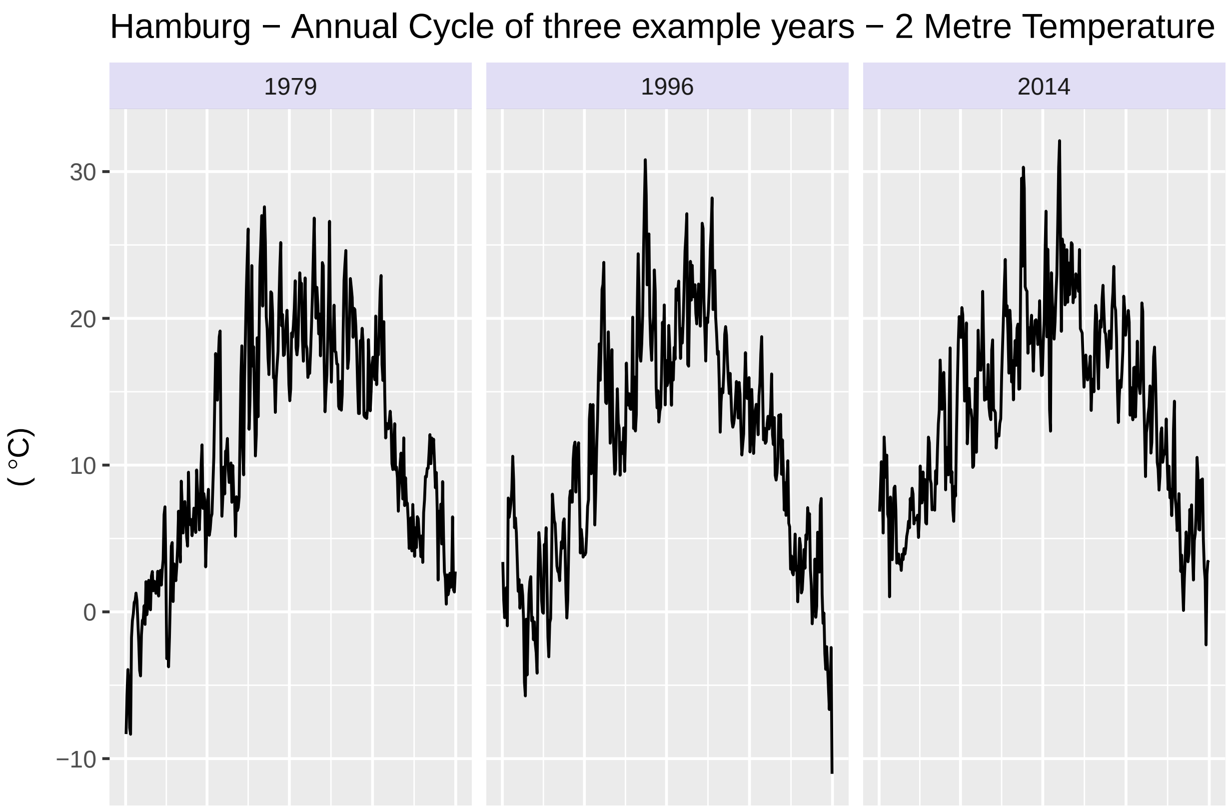
```
print(tail((lat)))
```

```
## [1] -86.25 -87.00 -87.75 -88.50 -89.25 -90.00
```

3 Temperature in Hamburg

3.1 Short overview about the Temperature data in Hamburg

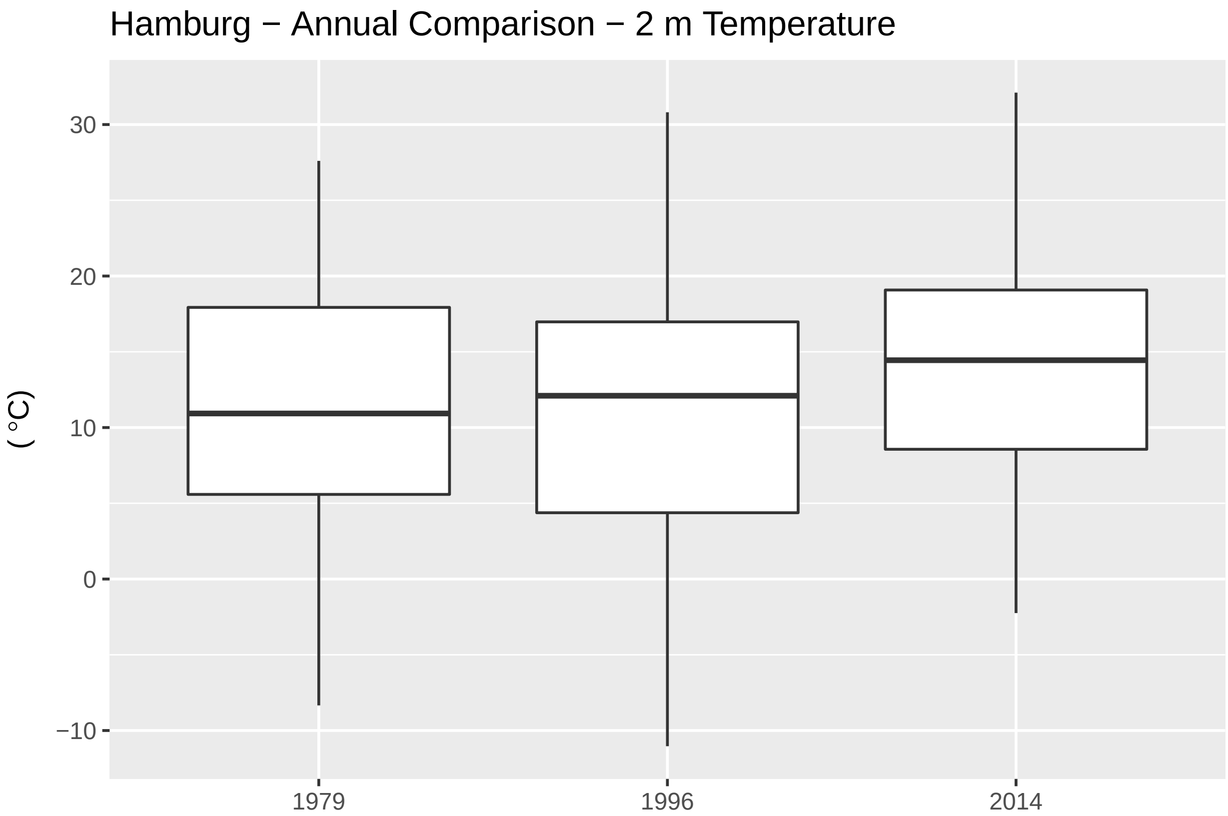
```
lat_HH_true = 53.551086
lon_HH_true = 9.993682

# "Interpolation nearest value"
lat_HH = lat[which.min(abs(lat - lat_HH_true))]
lon_HH = lon[which.min(abs(lon - lon_HH_true))]

#lon_TK = 35.652832
#lat_TK = 139.839478

data=data.frame(timestamp, t2m[lon_HH, lat_HH, ])


data %>% mutate(pyear = as.Date(cut(timestamp, breaks = "year"))) %>%
  ggplot(aes(x=timestamp, y=t2m[lon_HH, lat_HH, ]-273.15)) +
    geom_line() +
    ggtitle("Hamburg - Annual Cycle of three example years - 2 Metre Temperature") +
    theme(
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        strip.background = element_rect(fill=alpha("slateblue",0.2)),
        strip.placement="bottom"
        ) +
    xlab("") +
    ylab(expression("("*~degree*C*")")) +
    facet_wrap(~format(as.Date(pyear), "%Y"), scales="free_x", nrow=1)
```



This figure of the timeseries of the 2 m temperature in Hamburg shows only the data and we can see the annual cycle of the temperature. Additionally we gain some information about the range of the temperature in Hamburg and we can see some spikes.

3.2 Mean temperature in Hamburg

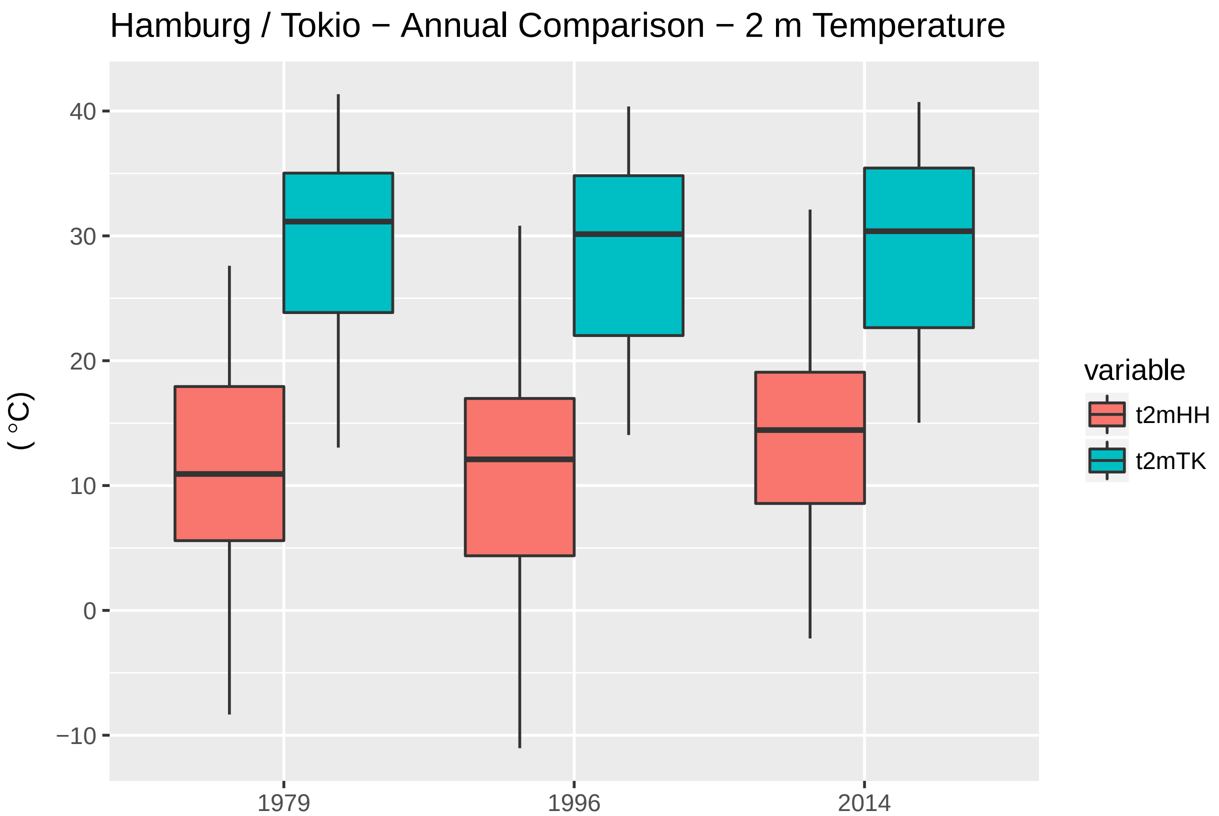
```
data=data.frame(timestamp, t2m[lon_HH, lat_HH, ])

data$year = format(timestamp, "%Y")

ggplot(data) +
  geom_boxplot(aes(x=year, y=t2m[lon_HH, lat_HH, ]-273.15)) +
  ggtitle("Hamburg - Annual Comparison - 2 m Temperature") +
  xlab("") +
  ylab(expression("("*~degree*C*")"))
```



The comparison of the statistics of the temperature data between the years 1979, 1996 and 2014 shows a increasing mean value. The year 1996 has the greates variation in general but a similar inner quarter range, so only there were more extreme values.

4 Comparison to Tokio

```
lat_TK_true = 139.839478
lon_TK_true = 35.652832

# "Interpolation nearest value"
lat_TK = lat[which.min(abs(lat - lat_TK_true))]
lon_TK = lon[which.min(abs(lon - lon_TK_true))]

library(reshape2)

data=merge(data.frame(timestamp, t2mHH = t2m[lon_HH, lat_HH, ]), data.frame(timestamp, t2mTK = t2m[lon_TK, la
data$year = format(timestamp, "%Y")

dat_comp = melt(data, id.vars="year", measure.vars=c("t2mHH", "t2mTK"))

ggplot(dat_comp) +
  geom_boxplot(aes(x=year, y=value-273.15, fill=variable)) +
  ggtitle("Hamburg / Tokio - Annual Comparison - 2 m Temperature") +
  xlab("") +
  ylab(expression("("*~degree*C*")"))
```



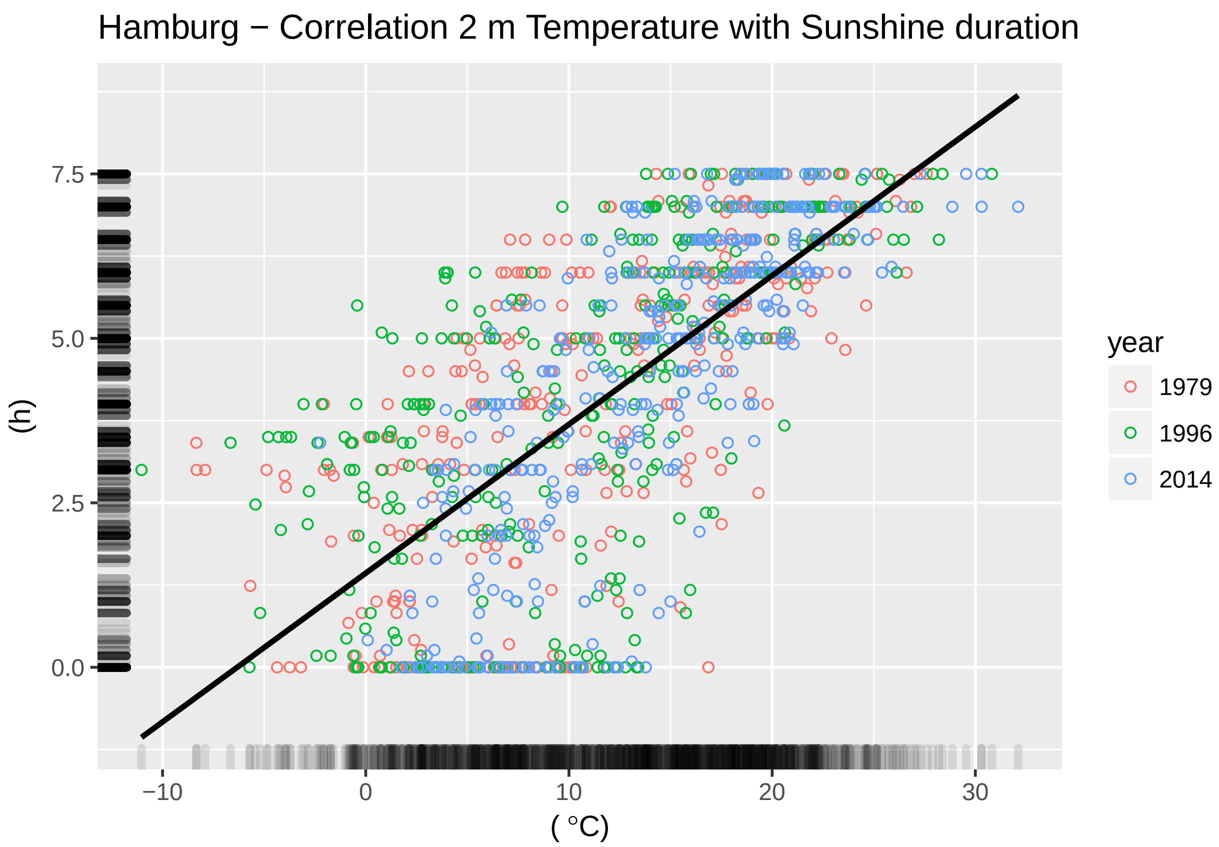
Tokio has in comparison with Hamburg a smaller variation. The annual cycle result probably in more variation of the temperature in Hamburg. The clima in Hamburg is obvisouly cold compared to the clima of Tokio.

5 Correlation between measurements

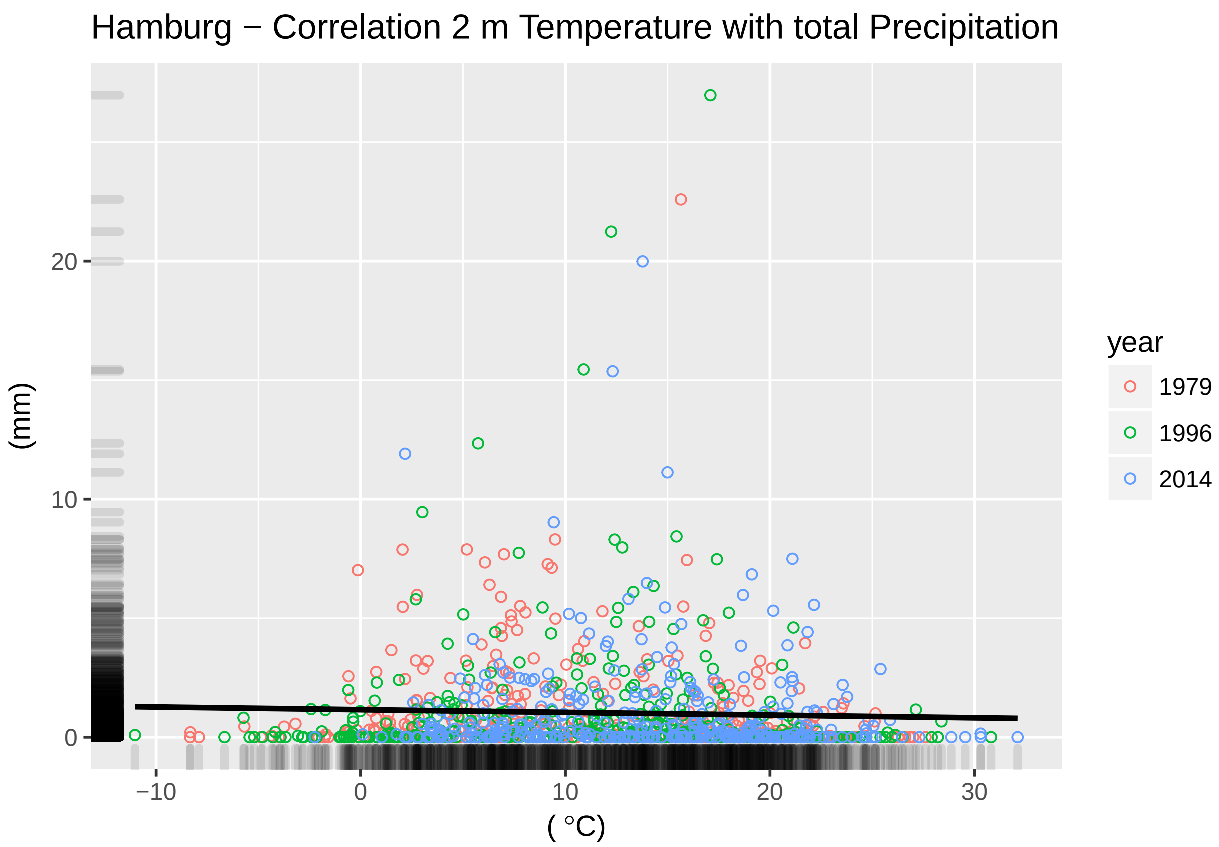
```
data = data.frame(t2m_HH = t2m[lon_HH, lat_HH, ], sund_HH = sund[lon_HH, lat_HH, ], tp_HH = tp[lon_HH, lat_HH

cor(data)
```

```
##              t2m_HH    sund_HH       tp_HH
## t2m_HH   1.00000000  0.7015230 -0.03897708
## sund_HH  0.70152298  1.0000000 -0.28885664
## tp_HH   -0.03897708 -0.2888566  1.00000000
```

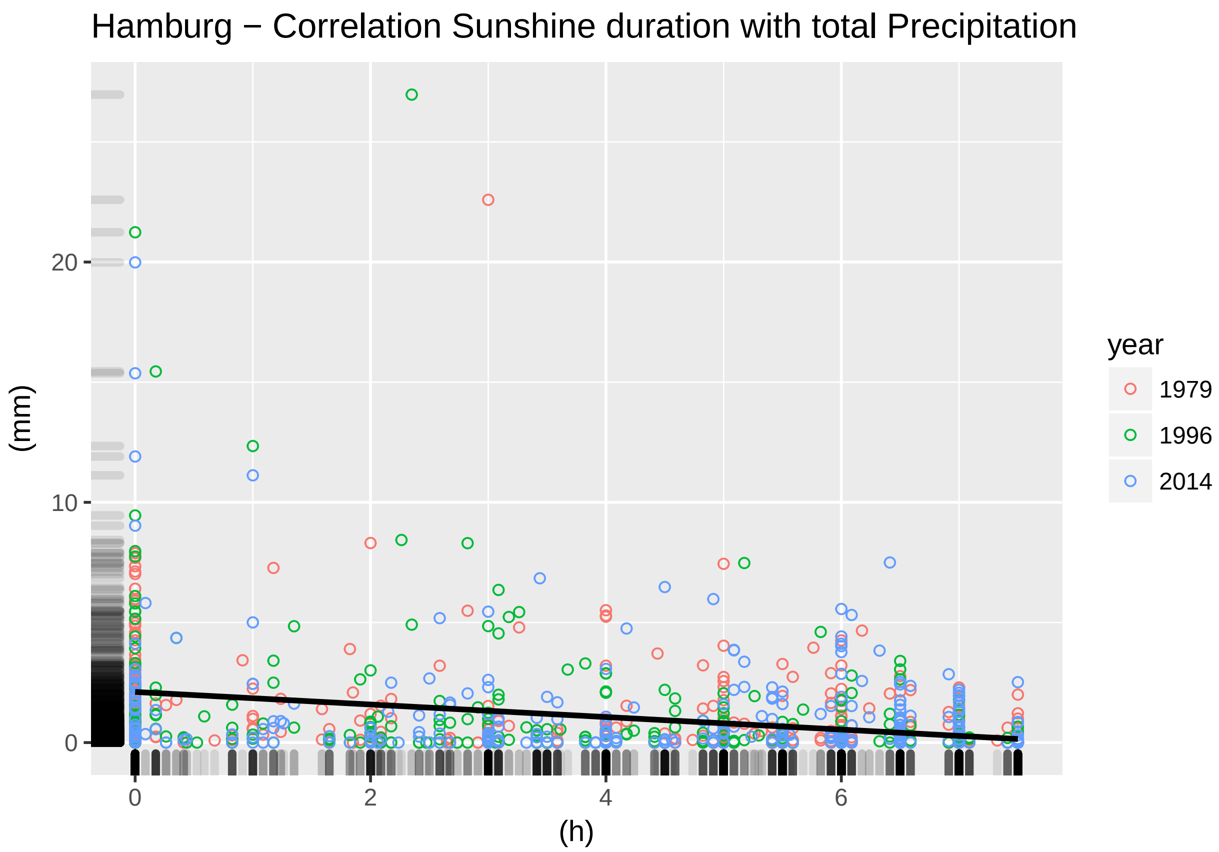
```
data$year = format(timestamp, "%Y")

ggplot(data, aes(t2m_HH-273.15, sund_HH / 3600., color=year)) +
  geom_point(shape=1) +
  geom_smooth(method=lm , color="black", se=FALSE) +
  geom_rug(col="black",alpha=0.1, size=1.5) +
  ggtitle("Hamburg - Correlation 2 m Temperature with Sunshine duration") +
  xlab(expression("("*~degree*C*")")) +
  ylab("(h)")
```




```
ggplot(data, aes(t2m_HH-273.15, tp_HH*1000, color=year)) +
  geom_point(shape=1) +
  geom_smooth(method=lm , color="black", se=FALSE) +
  geom_rug(col="black",alpha=0.1, size=1.5) +
  ggtitle("Hamburg - Correlation 2 m Temperature with total Precipitation") +
  xlab(expression("("*~degree*C*")")) +
  ylab("(mm)")
```




```
ggplot(data, aes(sund_HH /3600., tp_HH*1000, color=year)) +
  geom_point(shape=1) +
  geom_smooth(method=lm , color="black", se=FALSE) +
  geom_rug(col="black",alpha=0.1, size=1.5) +
  ggtitle("Hamburg - Correlation Sunshine duration with total Precipitation") +
  xlab("(h)") +
  ylab("(mm)")
```

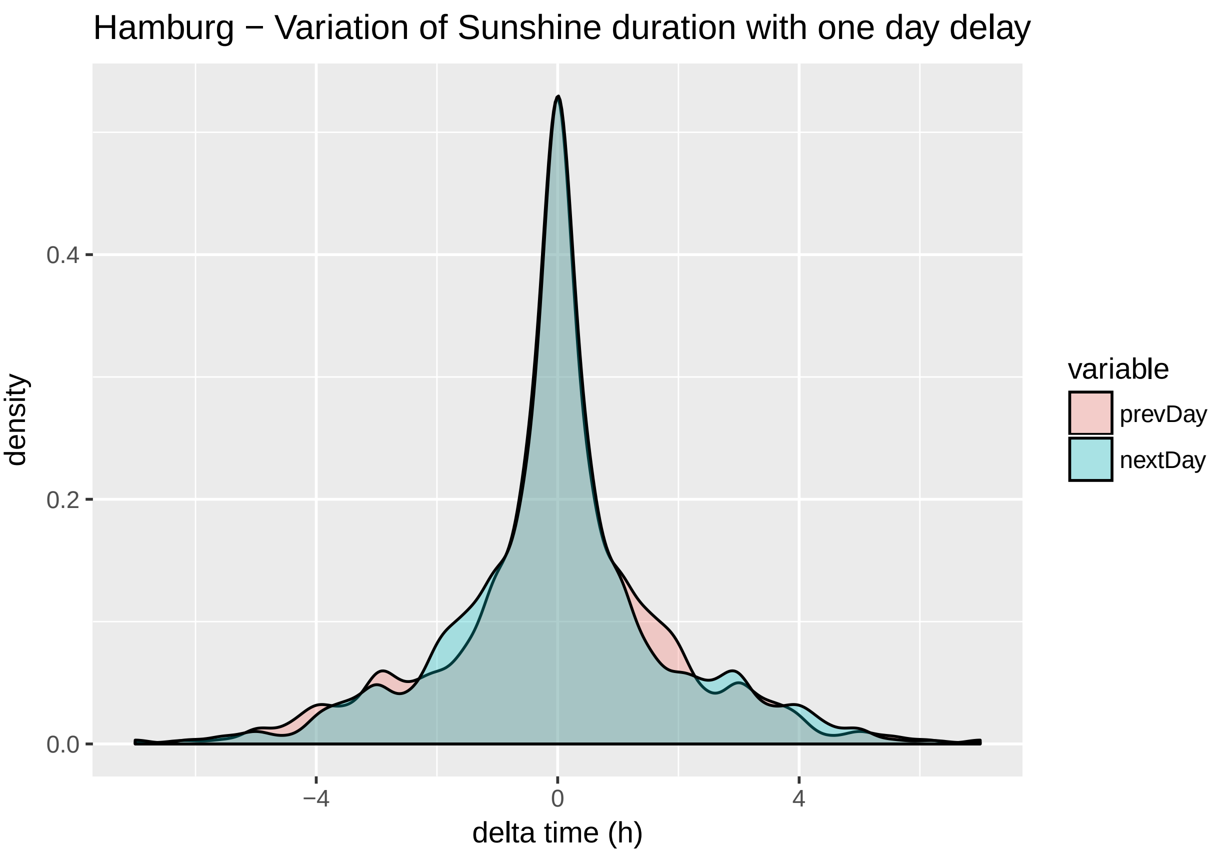


As expectet there is a good correlation between the sunshine duration and the temperature in 2 m, because the radiation of the sun results direct in the heating of the air near to the ground.

6 Difference between time values

```
offset = 1

sund_HH_tdiff = matrix(ncol = 2, nrow = (length(sund[lon_HH, lat_HH, ])-length(years)*2*offset))
t2m_HH_tdiff = matrix(ncol = 2, nrow = (length(t2m[lon_HH, lat_HH, ])-length(years)*2*offset))

iter = 1
for(iyear in c(1,2,3)){
  time_it = yearstep_begin[iyear] + offset

  while(time_it < (yearstep_end[iyear] - offset)){
    sund_HH_tdiff[iter,1] = sund[lon_HH, lat_HH, time_it] - sund[lon_HH, lat_HH, time_it-1]
    t2m_HH_tdiff[iter,1] = t2m[lon_HH, lat_HH, time_it] - t2m[lon_HH, lat_HH, time_it-1]
    sund_HH_tdiff[iter,2] = sund[lon_HH, lat_HH, time_it] - sund[lon_HH, lat_HH, time_it+1]
    t2m_HH_tdiff[iter,2] = t2m[lon_HH, lat_HH, time_it] - t2m[lon_HH, lat_HH, time_it+1]
    time_it = time_it+1
    iter = iter + 1
  }


}

data = data.frame(prevDay=sund_HH_tdiff[,1], nextDay=sund_HH_tdiff[,2])

dat_comp = melt(data, measure.vars=c("prevDay", "nextDay"))

ggplot(dat_comp, aes(value/3600, fill=variable)) +
  geom_density(alpha=0.3) +
  xlab("delta time (h)") +
  ggtitle("Hamburg - Variation of Sunshine duration with one day delay")
```

```
## Warning: Removed 6 rows containing non-finite values (stat_density).
```




```
data = data.frame(prevDay=t2m_HH_tdiff[,1], nextDay=t2m_HH_tdiff[,2])

dat_comp = melt(data, measure.vars=c("prevDay", "nextDay"))
```

```r
ggplot(dat_comp, aes(value, fill=variable)) +
  geom_density(alpha=0.3) +
  xlab("delta (K)") +
  ggtitle("Hamburg - Variation of 2 m Temperature with one day delay")
```

```
## Warning: Removed 6 rows containing non-finite values (stat_density).
```