Time series - Lab 2 (solutions)

Contents

Data

For this exercise we will use data from a flux tower located in a black spruce forest near Chibougamau.

> **Reference**: Bergeron, Margolis, Black, Coursolle, Dunn, Barr, & Wofsy. (2007). Comparison of carbon dioxide fluxes over three boreal black spruce forests in Canada. Global Change Biology, 13(1), 89–107. https://doi.org/10.1111/j.1365-2486.2006.01281.x.

Flux towers measure net ecosystem exchange or the amount of gas that is exchanged between the atmosphere and the ecosystem using eddy covariance technique.

> **Weblink**: https://www.neonscience.org/3d-interactive-flux-tower
>
> **Weblink**: https://www.youtube.com/watch?v=CR4Anc8Mkas
>
> **Weblink**: https://www.neonscience.org/data-collection/meteorology

We will start loading the required packages and the data.

```
library(fpp3)
library(dplyr)
library(ggplot2)
library(cowplot)
EOBS_fluxnet <- read.csv("../donnees/EOBS_fluxnet2.csv")
head(EOBS_fluxnet)
```

```
##   Year Day GapFilled_NEP GapFilled_R GapFilled_GEP TimeSteps
## 1 2004   1    -0.3702583   0.3702583             0        48
## 2 2004   2    -0.3226569   0.3226569             0        48
## 3 2004   3    -0.3143513   0.3143513             0        48
## 4 2004   4    -0.3108769   0.3108769             0        48
## 5 2004   5    -0.3105173   0.3105173             0        48
## 6 2004   6    -0.3069446   0.3069446             0        48
```

The columns are:

- *Year* is the year of the observation
- *Day* is the day of the year of the observation (1-365)
- *GapFilled_NEP* is the daily net ecosystem productivity (*umol C m-2 of stand s-1*)
- *GapFilled_R* is the daily ecosystem respiration (*umol C m-2 of stand s-1*)
- *GapFilled_GEP* is the daily gross ecosystem productivity (*umol C m-2 of stand s-1*)

- *TimeSteps* is an integer saying how many half hourly data composed the daily aggregates

1. Process and explore NEP time series

(1a) Create a temporal data frame (*tsibble*). As a first step, you must add a column containing the date using the information in *Year* and *Day*. Consult the following website to understand how to deal with date/time data in *R*: https://www.stat.berkeley.edu/~s133/dates.html (1 point)

```
EOBS_fluxnet = mutate(EOBS_fluxnet,
                      Date = as.Date(paste(EOBS_fluxnet$Year,EOBS_fluxnet$Day), format='%Y %j'))
EOBS_fluxnet = as_tsibble(EOBS_fluxnet, index = Date)
head(EOBS_fluxnet)
```

```
## # A tsibble: 6 x 7 [1D]
##    Year   Day GapFilled_NEP GapFilled_R GapFilled_GEP TimeSteps Date
##   <int> <int>         <dbl>       <dbl>         <dbl>     <int> <date>
## 1  2004     1        -0.370       0.370             0        48 2004-01-01
## 2  2004     2        -0.323       0.323             0        48 2004-01-02
## 3  2004     3        -0.314       0.314             0        48 2004-01-03
## 4  2004     4        -0.311       0.311             0        48 2004-01-04
## 5  2004     5        -0.311       0.311             0        48 2004-01-05
## 6  2004     6        -0.307       0.307             0        48 2004-01-06
```

(1b) One of the problems working with daily data is to deal with leap years. In this case we load data with constant 365 days per year. This is a common solution to simplify the data processing, especially in modelling. In order to add one more day per each leap year we can use the functions *fill_gaps* (https://www.rdocumentation.org/packages/tsibble/versions/1.0.0/topics/fill_gaps) and *tidyr::fill* (https://www.rdocumentation.org/packages/tidyr/versions/1.1.3/topics/fill). We can specify that the added rows have *Day* equal to 366 and *GapFilled_NEP*, *GapFilled_R*, *GapFilled_GEP*, and *Year* equal to the value of the preceding row. (1 point)

```
EOBS_fluxnet = EOBS_fluxnet %>%
              fill_gaps(Day = 366) %>%
              tidyr::fill(GapFilled_NEP, .direction = "down") %>%
              tidyr::fill(GapFilled_R, .direction = "down") %>%
              tidyr::fill(GapFilled_GEP, .direction = "down") %>%
              tidyr::fill(Year, .direction = "down")
EOBS_fluxnet[EOBS_fluxnet$Day==366,]
```

```
## # A tsibble: 2 x 7 [1D]
##    Year   Day GapFilled_NEP GapFilled_R GapFilled_GEP TimeSteps Date
##   <int> <dbl>         <dbl>       <dbl>         <dbl>     <int> <date>
## 1  2004   366        -0.227       0.227             0        NA 2004-12-31
## 2  2008   366        -0.644       0.644             0        NA 2008-12-31
```

Here above we visualize the two added lines for the leap years.

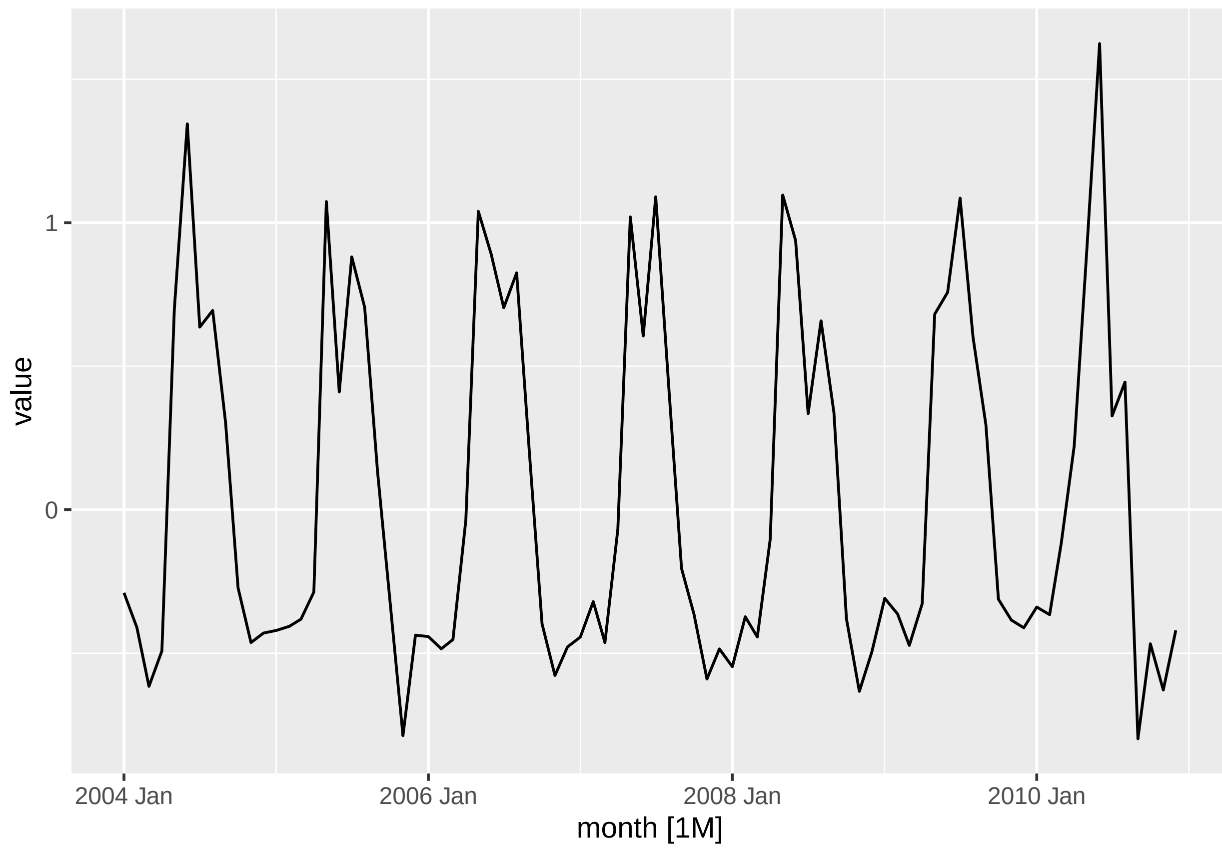
(1c) Obtain a new temporal data frame (*tsibble*) containing mean monthly values of *GapFilled_NEP*. Plot the obtained time series and comment it. How does the time series vary over time? What do negative values mean? (1 point)

```
EOBS_fluxnet_monthly <- index_by(EOBS_fluxnet, month = yearmonth(Date)) %>%
  summarize(GapFilled_NEP = mean(GapFilled_NEP))
head(EOBS_fluxnet_monthly)
```

```
## # A tsibble: 6 x 2 [1M]
##      month GapFilled_NEP
```

```
##       <mth>          <dbl>
## 1 2004 Jan         -0.289
## 2 2004 Feb         -0.410
## 3 2004 Mar         -0.616
## 4 2004 Apr         -0.492
## 5 2004 May          0.701
## 6 2004 Jun          1.34
```

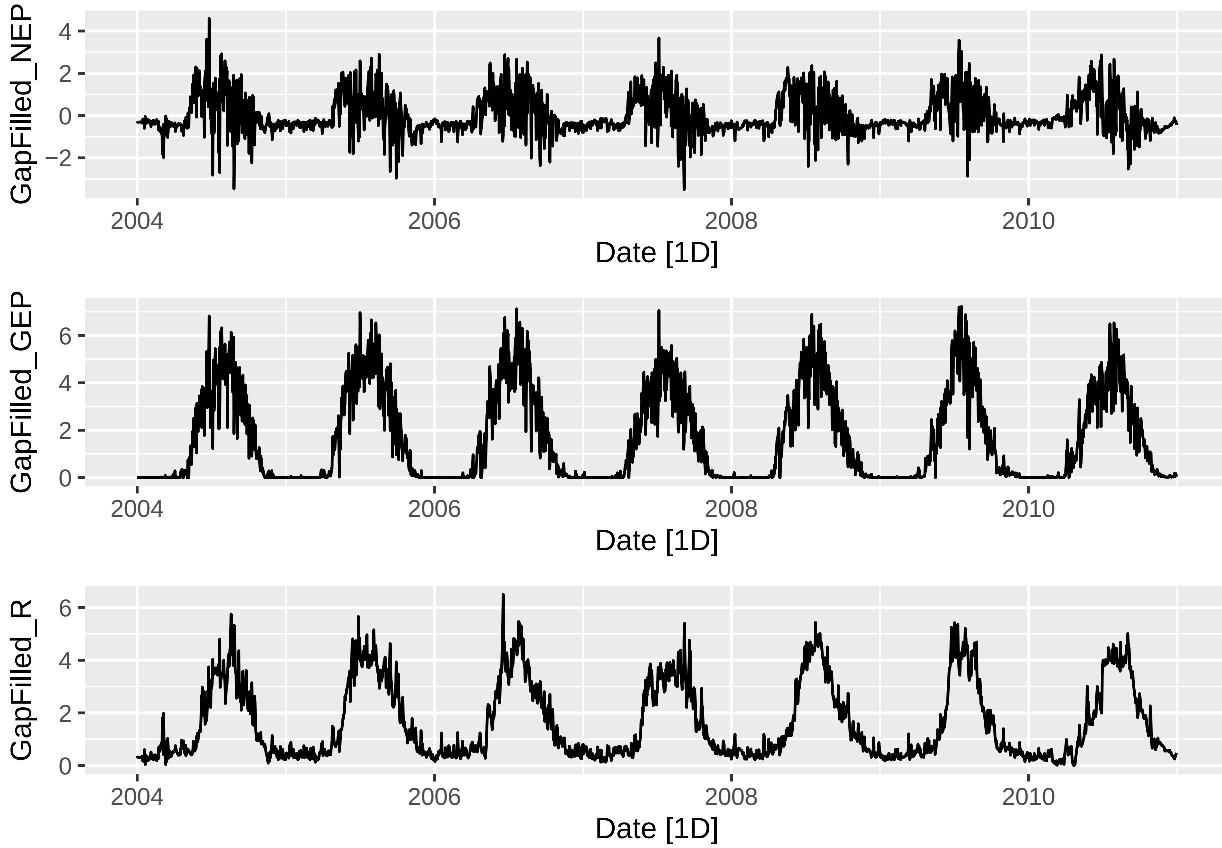
```r
autoplot(EOBS_fluxnet_monthly,vars(GapFilled_NEP))
```



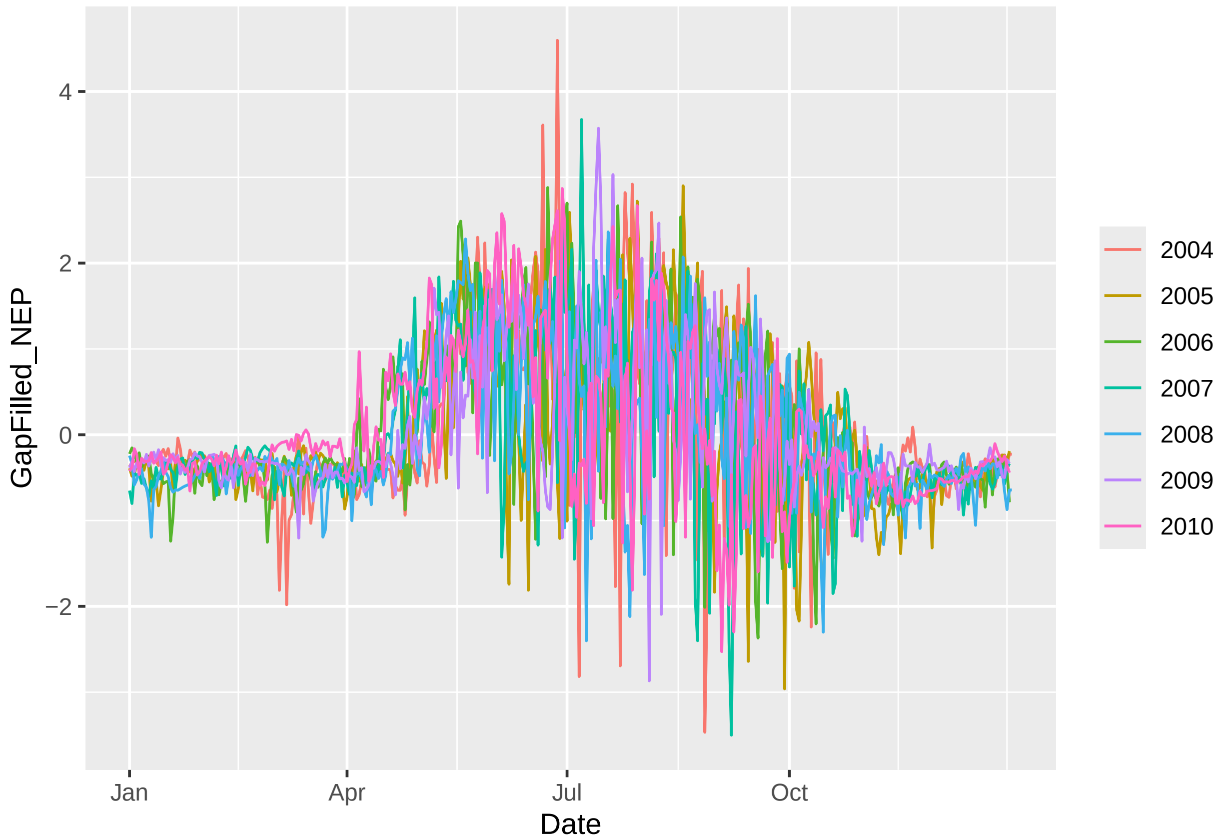
In the plot we see the typical annual cycle of the productivity of a boreal forest ecosystem with contrasted winter dormancy and strong uptake over the growing season. The negative values during winter time mean that the ecosystem is a carbon source (respiration higher than photosynthesis).

(1d) Plot the 3 time-series of daily values (*GapFilled_NEP*, *GapFilled_R*, *GapFilled_GEP*), the annual seasonality of *GapFilled_NEP* (use the daily dataset as well as the monthly dataset providing two distinct plots), and the trend of *GapFilled_NEP* data for each month over time (use the monthly dataset). When does the growing season start and end at the study site? When does the peak of photosynthesis occur? Is there any evident trend in the mean monthly values? (1 point)

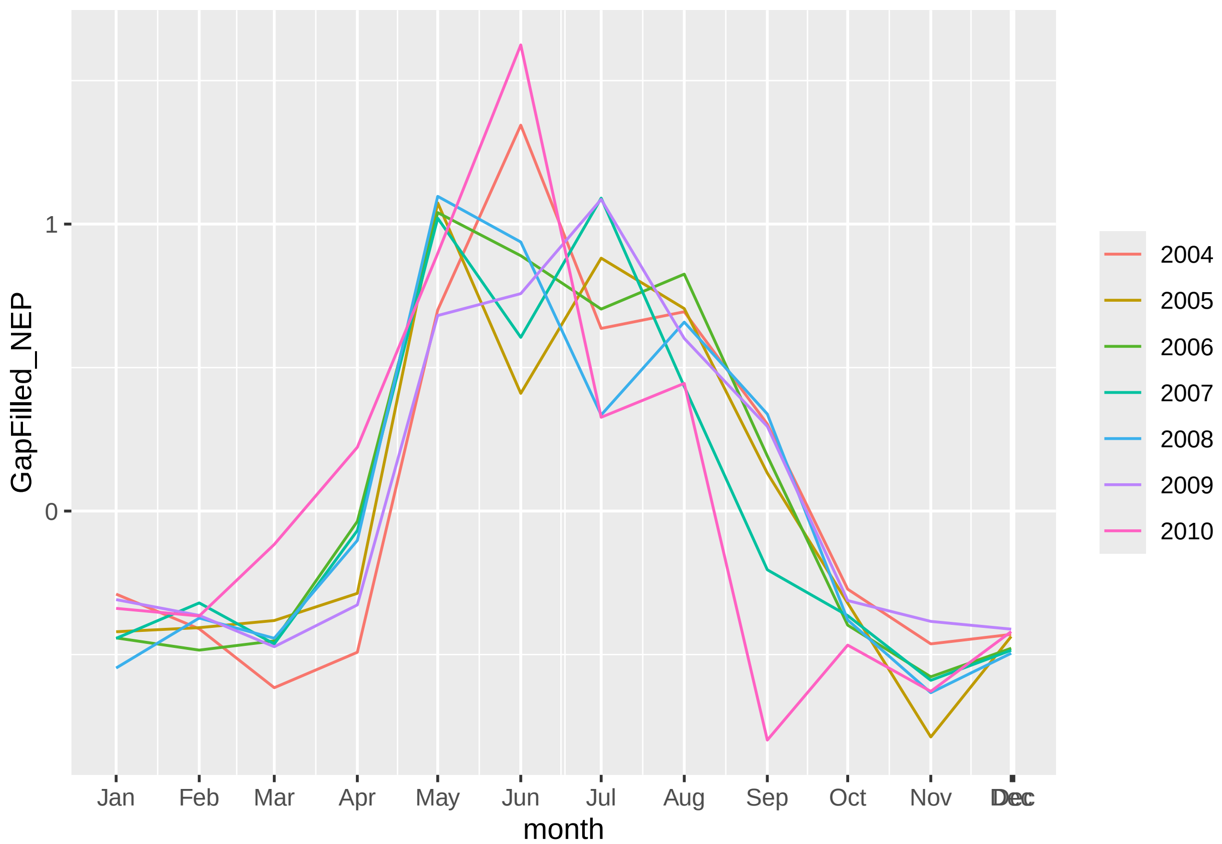
```r
plot_grid(
    autoplot(EOBS_fluxnet, GapFilled_NEP),
    autoplot(EOBS_fluxnet, GapFilled_GEP),
    autoplot(EOBS_fluxnet, GapFilled_R),
    ncol = 1, align = "v")
```




```
gg_season(EOBS_fluxnet, y = GapFilled_NEP)
```

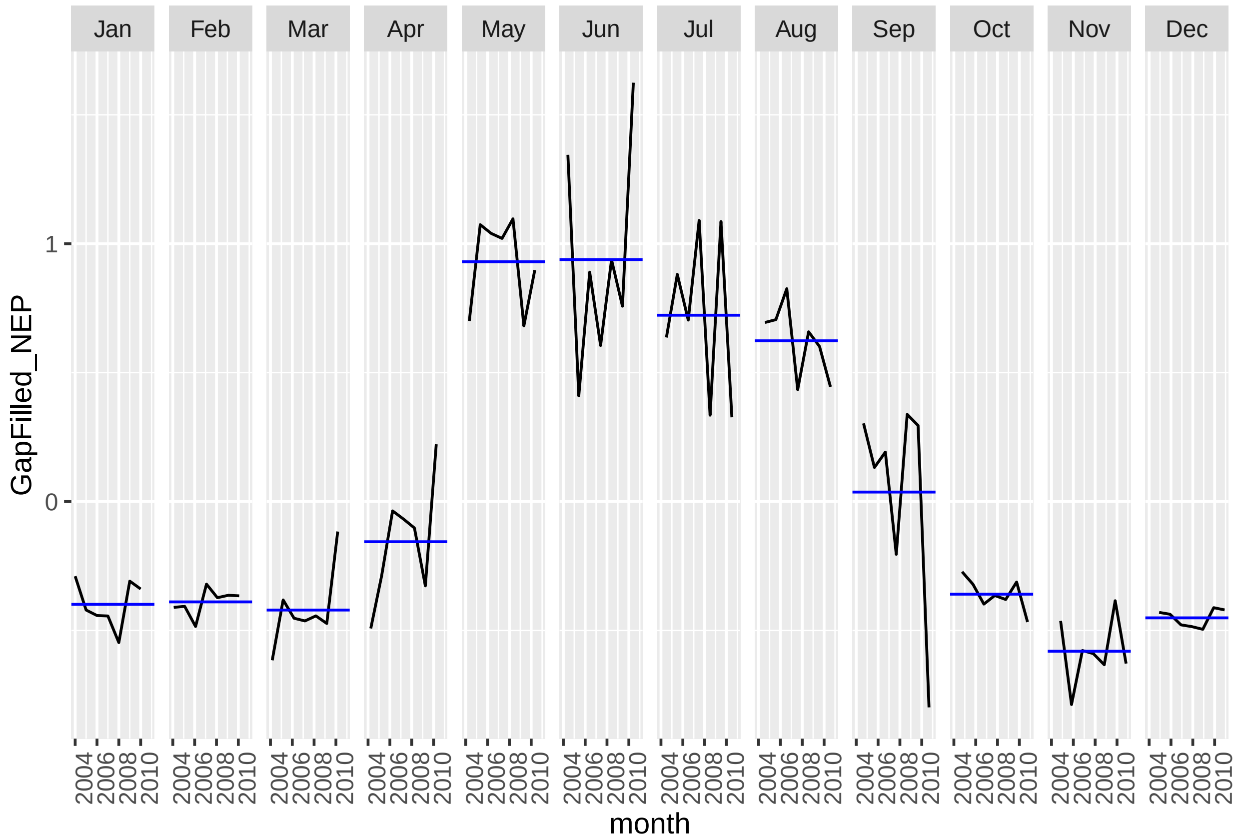



```
gg_season(EOBS_fluxnet_monthly, y = GapFilled_NEP)
```



The growing season starts in May and ends in September. The peak of photosynthesis is between May and July according to the year.

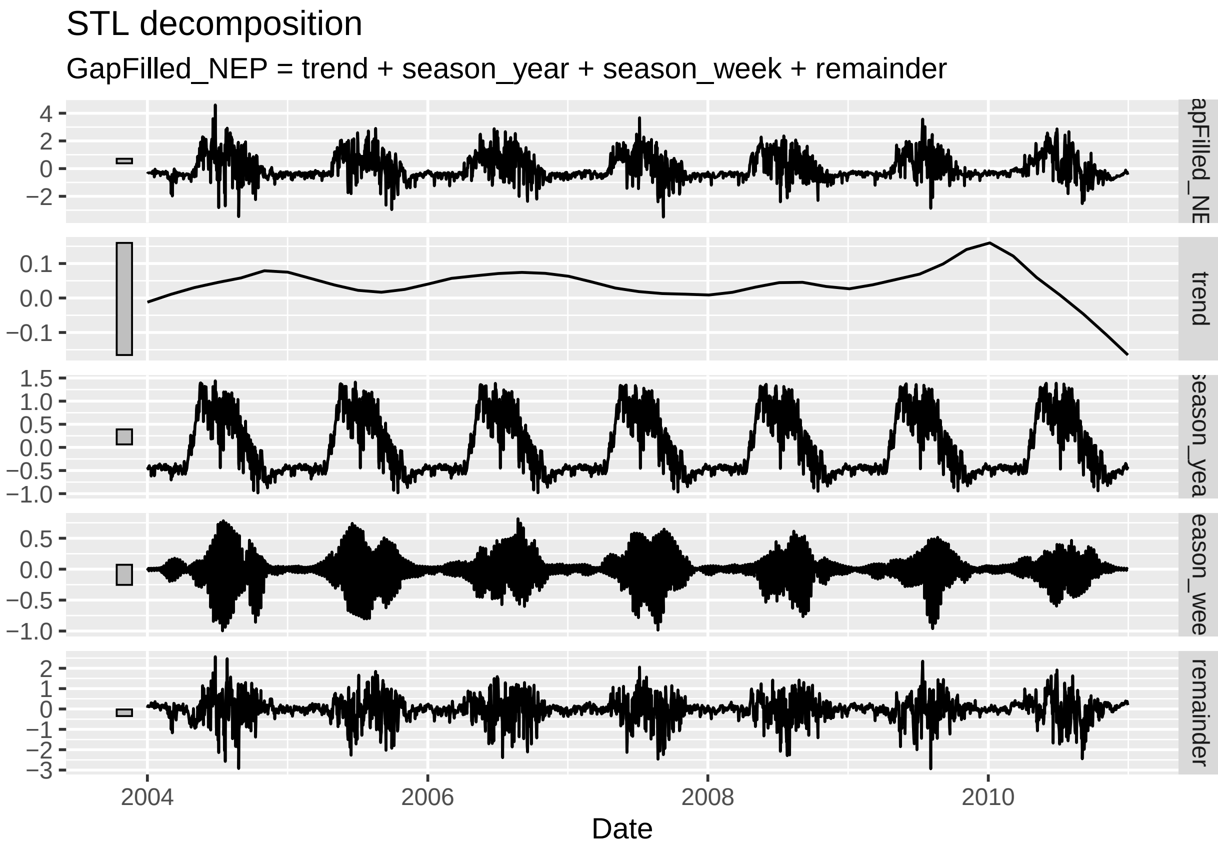
```r
gg_subseries(EOBS_fluxnet_monthly, y = GapFilled_NEP)
```



There is no evident common trend in the monthly values. However, an upward trend in April and two negative trends in August and September are visible.

(1e) Extract the several components of the *GapFilled_NEP* daily time series (trend, seasonality, and residuals). What is the components' relative importance? What does it mean? Finally, store the components into a new temporal data frame (*tsibble*). (1 point)

```
decomp <- model(EOBS_fluxnet, STL(GapFilled_NEP))
autoplot(components(decomp))
```

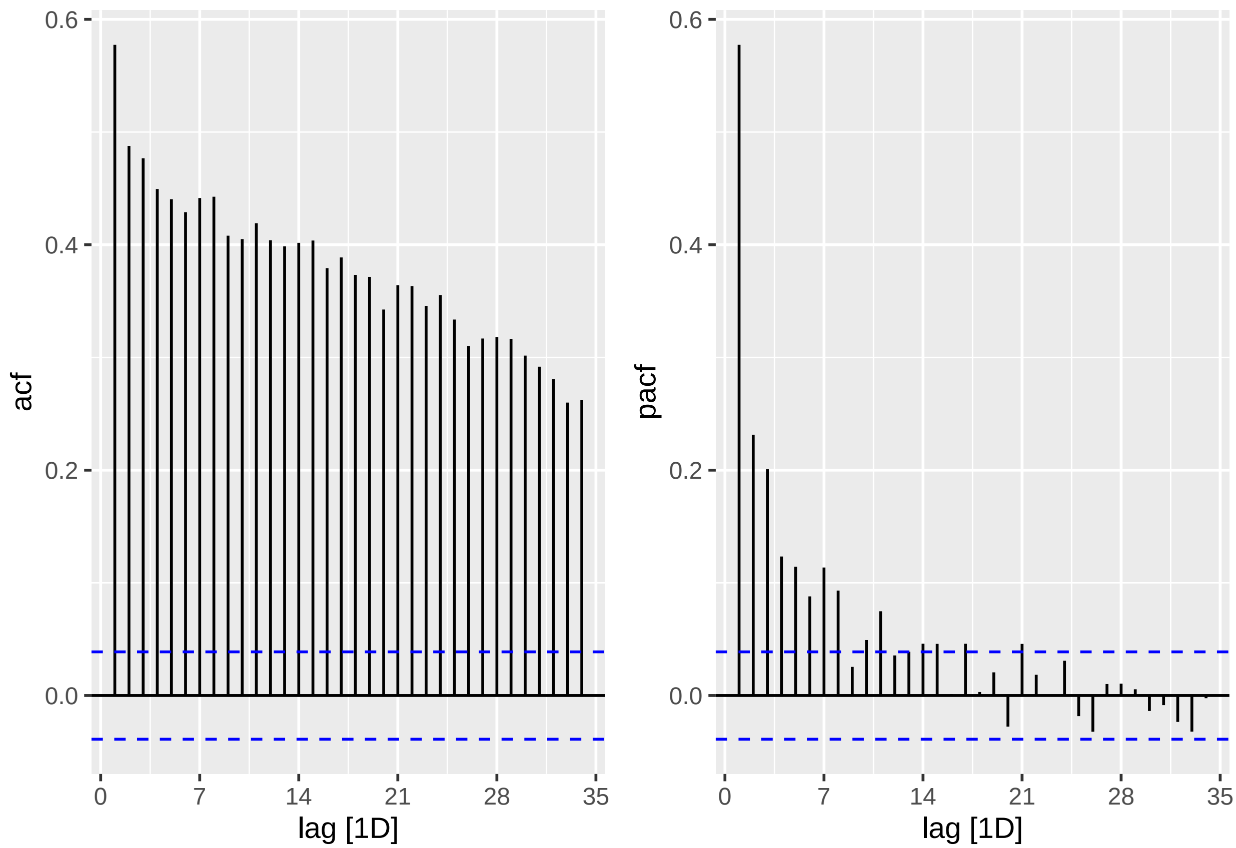



```
NEP_components = components(decomp)
```

The residual component is the more important showing that the daily meteorology over the growing season strongly influences the productivity. The second more important component is the annual seasonality.

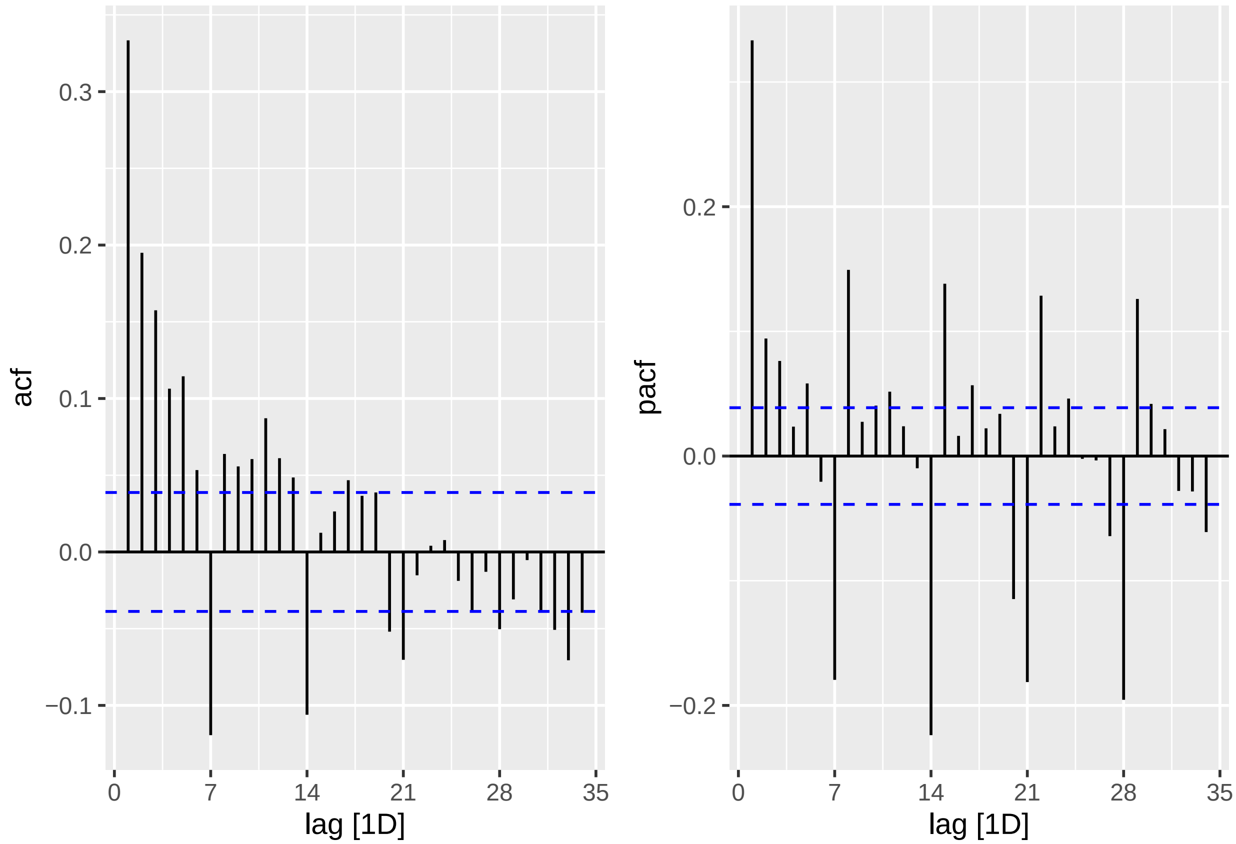
(1f) Analyze the autocorrelation and the partial autocorrelation of the *GapFilled_NEP* daily time series and of its residual component extracted in 1e. What do you deduce from these plots? (1 point)

```
plot_grid(autoplot(ACF(EOBS_fluxnet, GapFilled_NEP)), autoplot(PACF(EOBS_fluxnet, GapFilled_NEP)))
```



The autocorrelation structure of the NEP data is very long lasting because of the clear annual seasonality.

```
plot_grid(autoplot(ACF(NEP_components, remainder)), autoplot(PACF(NEP_components, remainder)))
```



The seasonal components obtained in 1e are not well adjusted to the data because the residuals show a weekly seasonality.

2. GAM model for NEP

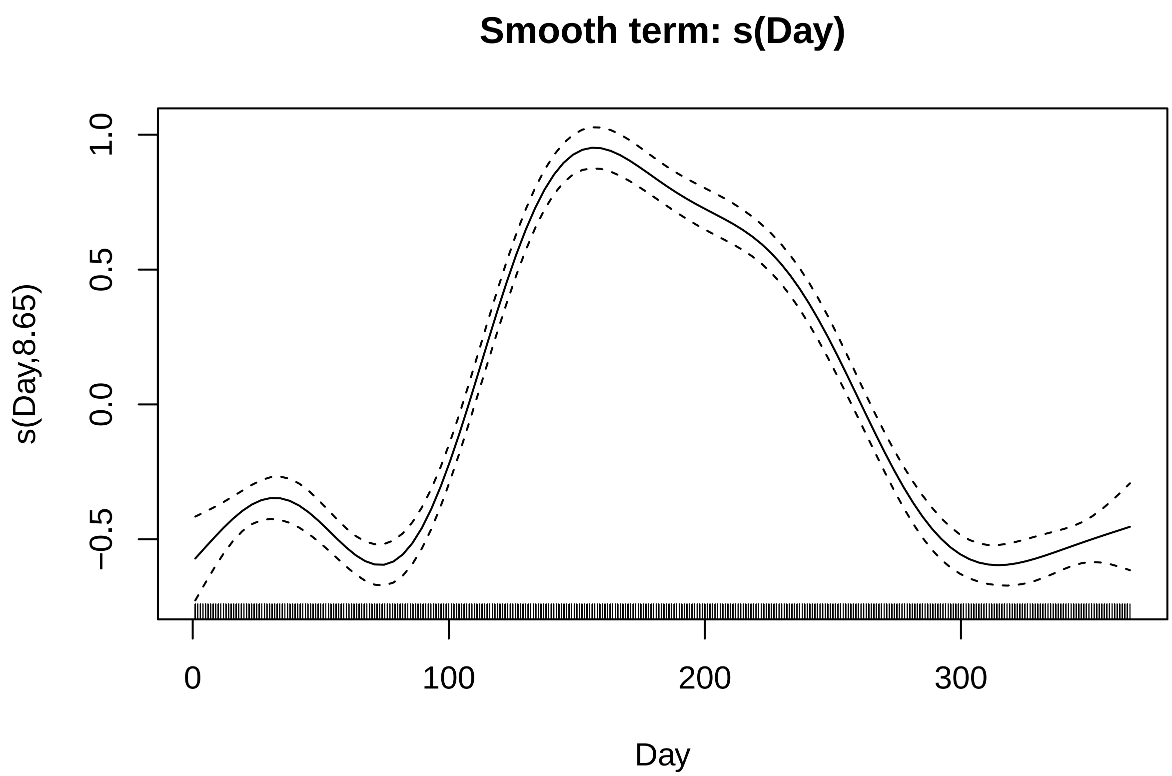
(2a) Fit a Generalized Additive Model (GAM) on *GapFilled_NEP* using Day of Year (DOY) as a smooth term. Plot the estimated smooth function. Evaluate whether the model residuals meet key assumptions: Homoscedasticity (residuals vs fitted), Normality (QQ-plot), Absence of temporal autocorrelation (time series plot and ACF). (1 point)

```
# Load required package
library(mgcv)

# Fit a GAM with a smooth term for Day of Year (DOY)
NEP_gamm1 <- gamm(GapFilled_NEP ~ s(Day), data = EOBS_fluxnet)

# Inspect the smooth term of the model
summary(NEP_gamm1$gam)
```

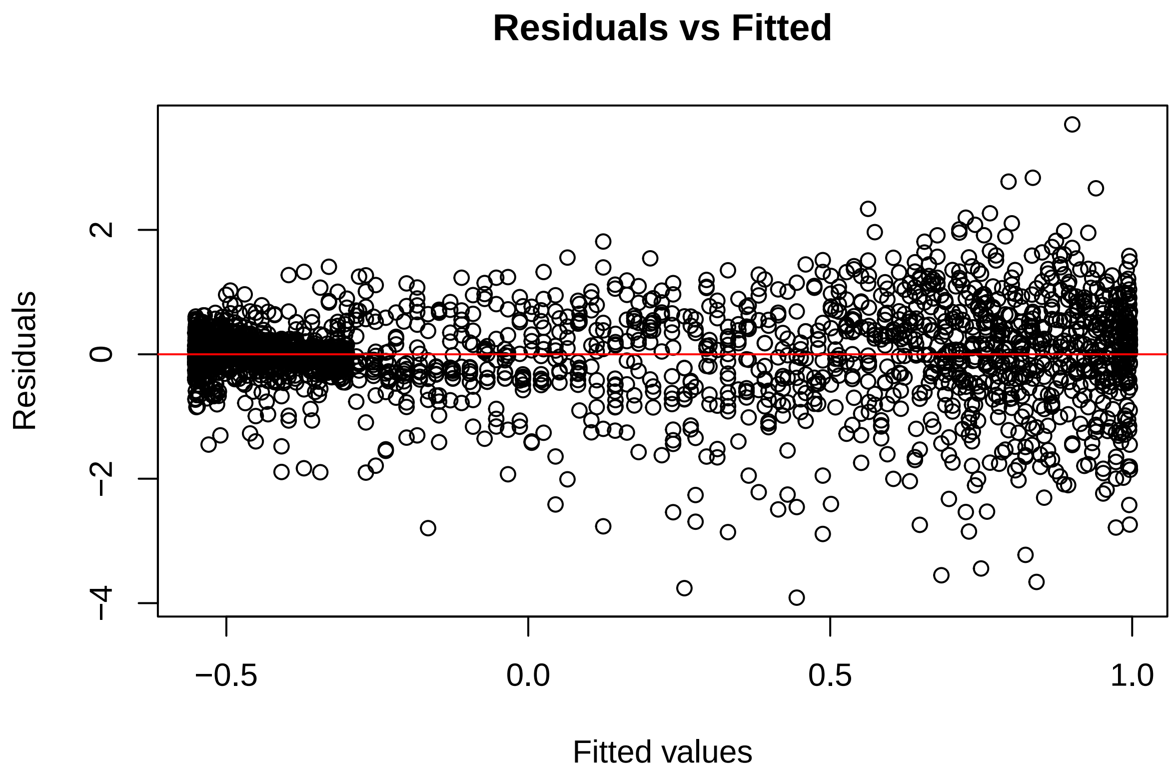
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## GapFilled_NEP ~ s(Day)
##
## Parametric coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.04443    0.01360   3.268   0.0011 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df   F p-value
## s(Day) 8.654  8.654 200  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.404
##   Scale est. = 0.47262   n = 2557
```

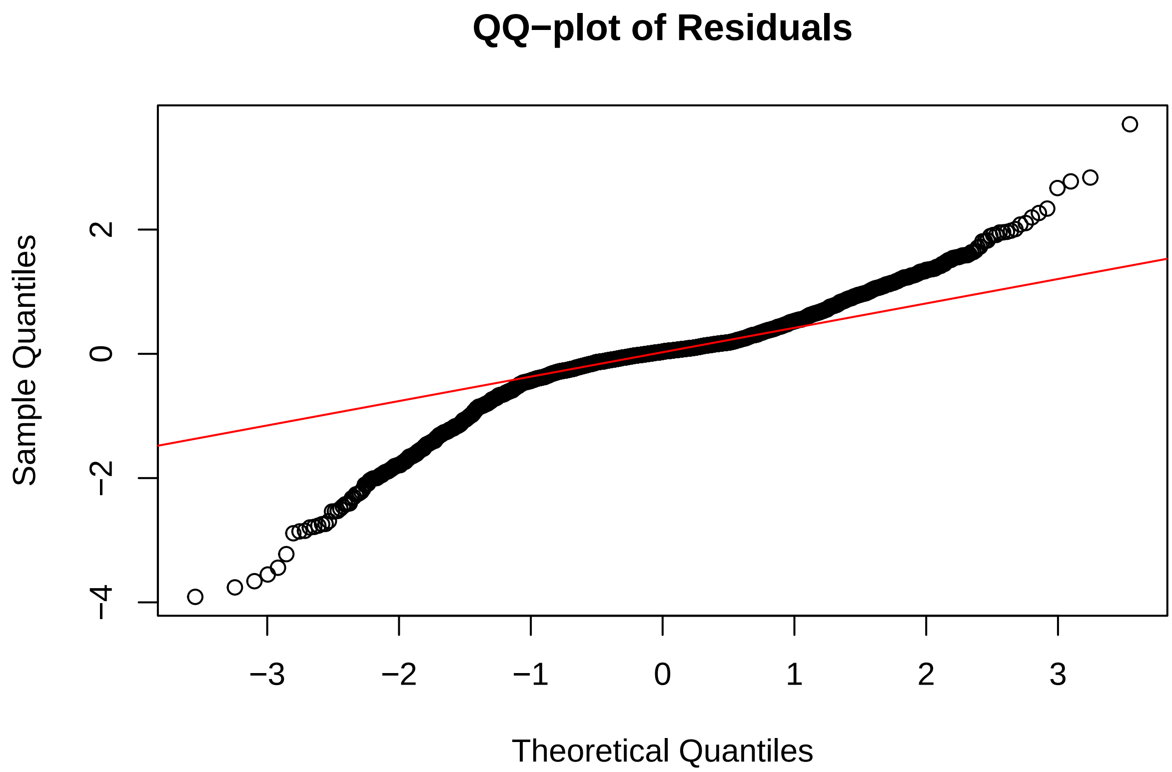
```r
# Plot the estimated smooth function s(Day)
plot(NEP_gamm1$gam, pages = 1, main = "Smooth term: s(Day)")
```



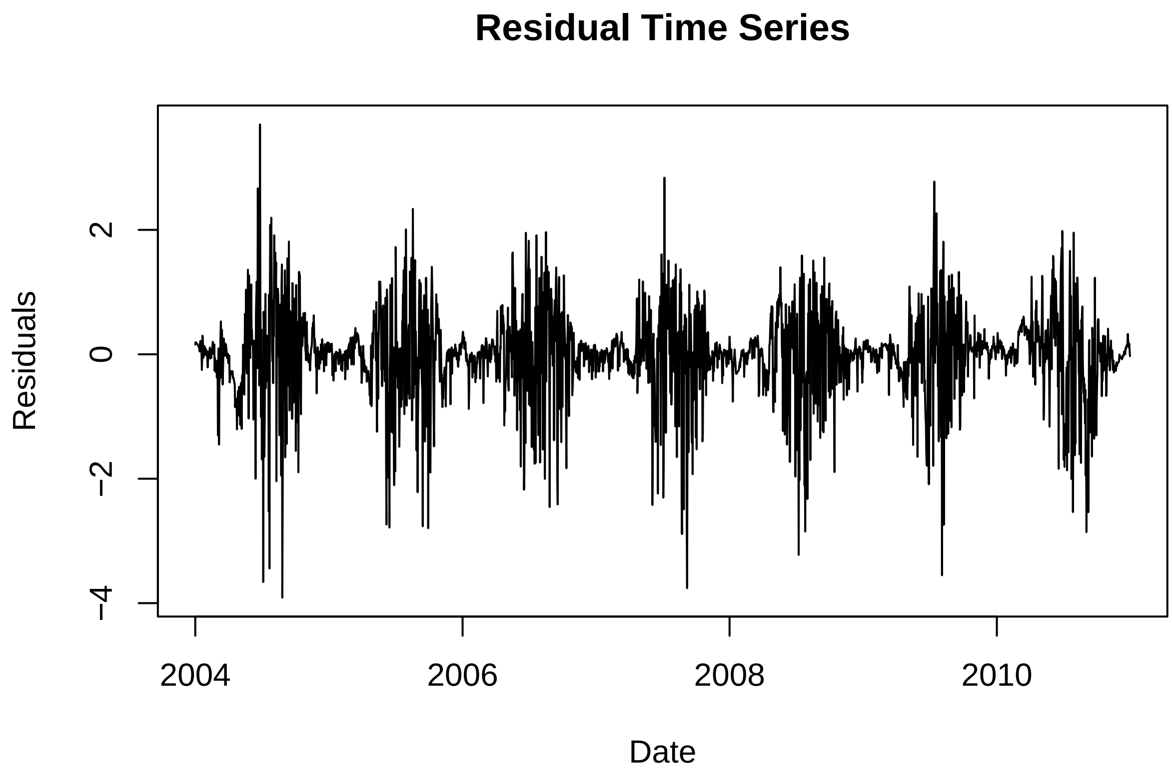

```r
# --- DIAGNOSTIC CHECKS ---

# 1. Homoscedasticity: Plot residuals vs fitted values
plot(fitted(NEP_gamm1$lme), resid(NEP_gamm1$lme),
     xlab = "Fitted values", ylab = "Residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red")
```



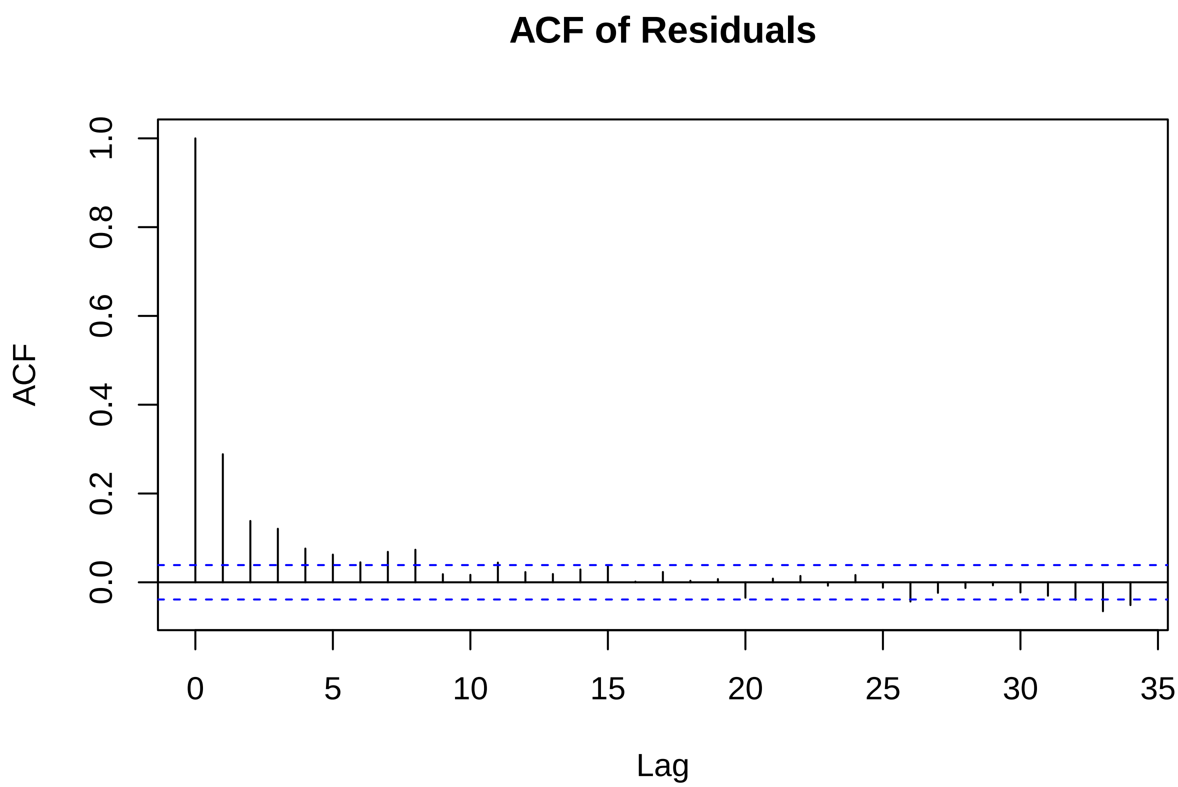

```
# 2. Normality: QQ-plot of residuals
qqnorm(resid(NEP_gamm1$lme), main = "QQ-plot of Residuals")
qqline(resid(NEP_gamm1$lme), col = "red")
```




```
# 3. Temporal autocorrelation: Plot residual time series
res <- resid(NEP_gamm1$lme)

# Ensure Date column exists and is in Date format
plot(EOBS_fluxnet$Date, res, type = "l", xlab = "Date", ylab = "Residuals",
     main = "Residual Time Series")
```




```r
# 4. Autocorrelation Function (ACF): Check for temporal autocorrelation
acf(res, main = "ACF of Residuals")
```



(2b) Extend the previous GAM by incorporating temporal autocorrelation in the residuals using an AR correlation structure (p = 1, q = 0). Which model performs best (with or without temporal autocorrelation)? Why? Do the residuals of the best model meet the model's assumptions (homoscedasticity, normality, no remaining temporal autocorrelation)? (1 point)

```r
# Fit GAMM with AR(1) structure
gamm_ar1 <- gamm(GapFilled_NEP ~ s(Day), data = EOBS_fluxnet,
                 correlation = corARMA(p = 1, q = 0, form = ~ Date | 1))

# # Fit GAMM with MA(1) structure
# gamm_ma1 <- gamm(GapFilled_NEP ~ s(Day), data = EOBS_fluxnet,
#                  correlation = corARMA(p = 0, q = 1, form = ~ Date | 1))
#
# # Fit GAMM with ARMA(1,1) structure
# gamm_arma11 <- gamm(GapFilled_NEP ~ s(Day), data = EOBS_fluxnet,
#                     correlation = corARMA(p = 1, q = 1, form = ~ Date | 1))

# Compare models using AIC
AIC(NEP_gamm1$lme, gamm_ar1$lme)
```

```
##                df      AIC
## NEP_gamm1$lme   4 5389.834
## gamm_ar1$lme    5 5164.798
```

```r
# Suppose ARMA(1,1) gives the best AIC (replace if needed)
best_model <- gamm_ar1
summary(best_model$lme)
```

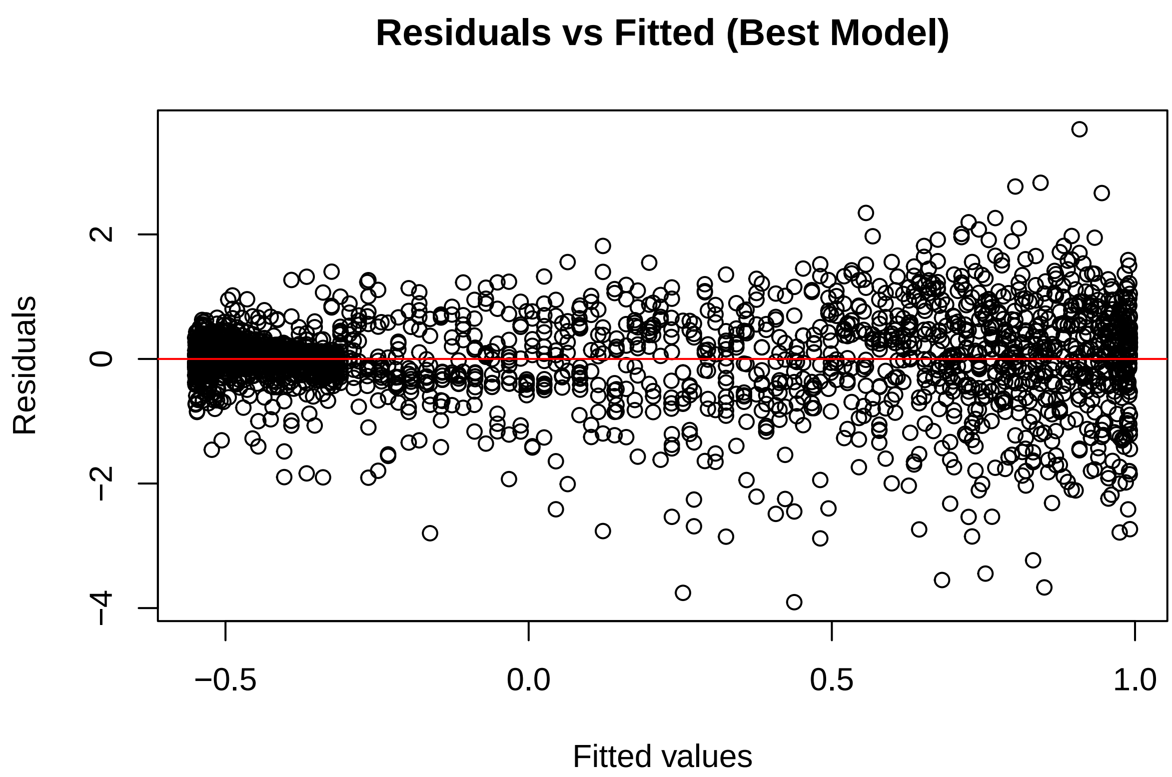
```
## Linear mixed-effects model fit by maximum likelihood
##   Data: strip.offset(mf)
##        AIC      BIC    logLik
##   5164.798 5194.031 -2577.399
##
## Random effects:
##  Formula: ~Xr - 1 | g
##  Structure: pdIdnot
##             Xr1     Xr2     Xr3     Xr4     Xr5     Xr6     Xr7     Xr8
## StdDev: 5.11061 5.11061 5.11061 5.11061 5.11061 5.11061 5.11061 5.11061
##          Residual
## StdDev: 0.6884267
##
## Correlation Structure: AR(1)
##  Formula: ~Date | g
##  Parameter estimate(s):
##       Phi
## 0.2926479
## Fixed effects:  y ~ X - 1
##                  Value Std.Error   DF  t-value p-value
## X(Intercept) 0.0444497 0.0184083 2555 2.414652  0.0158
## Xs(Day)Fx1   0.6269983 0.3567933 2555 1.757315  0.0790
##  Correlation:
##            X(Int)
## Xs(Day)Fx1 0
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -5.67242935 -0.33985060  0.05103841  0.41888514  5.35656006
##
## Number of Observations: 2557
## Number of Groups: 1
```

```
summary(best_model$gam)
```

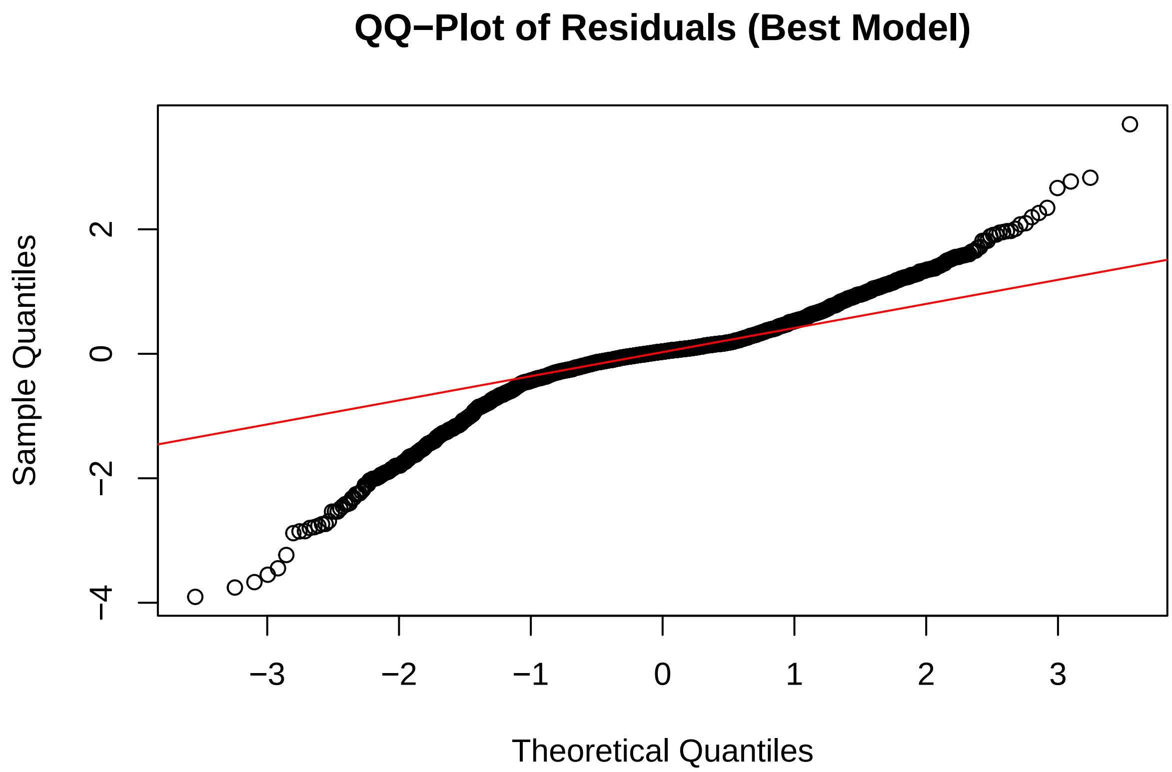
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## GapFilled_NEP ~ s(Day)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.04445    0.01840   2.415   0.0158 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df     F p-value
## s(Day) 8.378  8.378 112.3  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.404
```

```
##   Scale est. = 0.47393   n = 2557
```

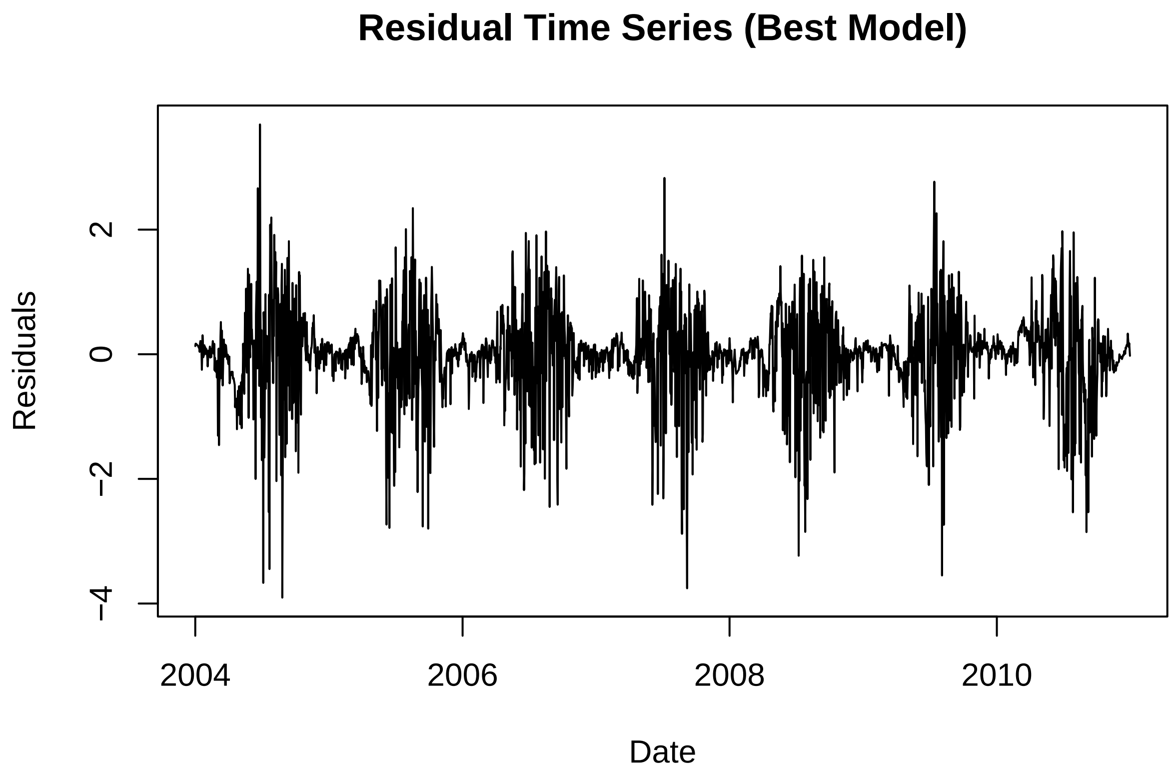
```r
# --- Residual Diagnostics for Best Model ---

# Plot residuals vs fitted
plot(fitted(best_model$lme), resid(best_model$lme),
     xlab = "Fitted values", ylab = "Residuals",
     main = "Residuals vs Fitted (Best Model)")
abline(h = 0, col = "red")
```



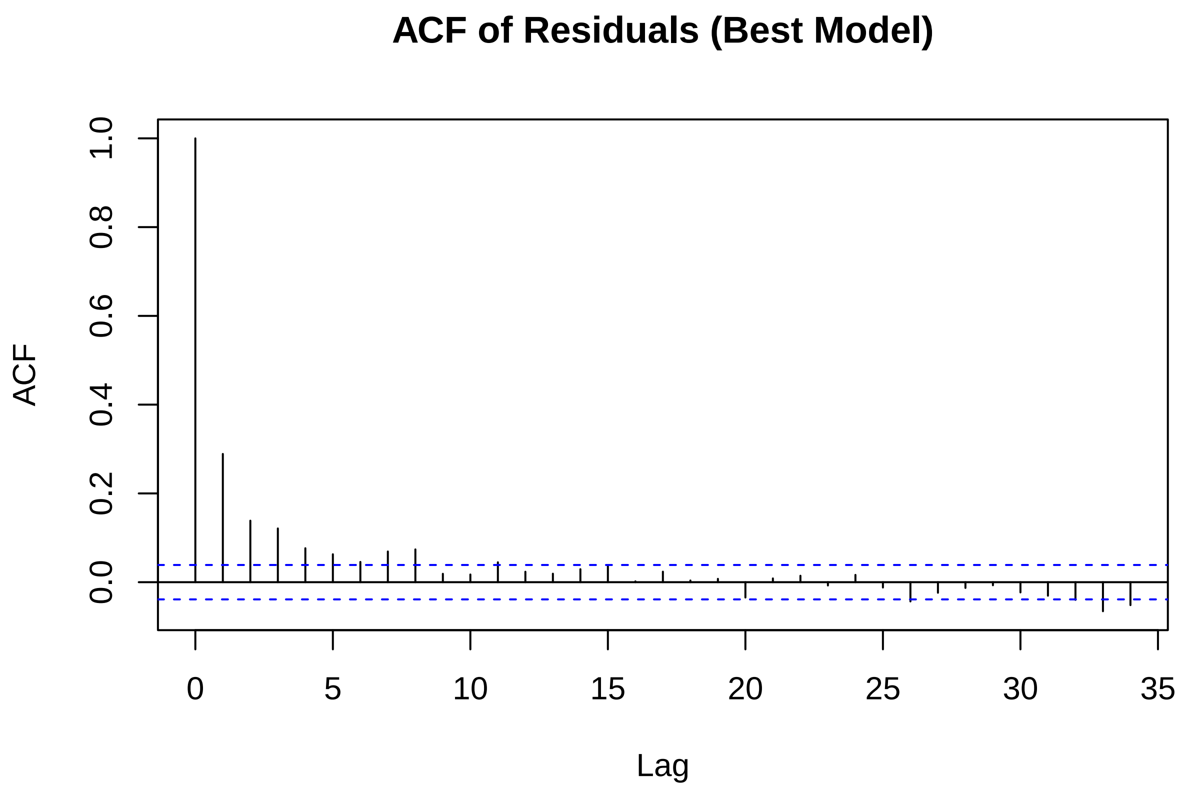

```r
# QQ-plot for normality
qqnorm(resid(best_model$lme), main = "QQ-Plot of Residuals (Best Model)")
qqline(resid(best_model$lme), col = "red")
```




```r
# Time series of residuals
res_best <- resid(best_model$lme)
plot(EOBS_fluxnet$Date, res_best, type = "l", xlab = "Date", ylab = "Residuals",
     main = "Residual Time Series (Best Model)")
```




```r
# ACF of residuals
acf(res_best, main = "ACF of Residuals (Best Model)")
```



3. GAM model for NEP with external predictors

We will start loading meteorological data for the flux tower site.

```
My_meteo=read.delim("../donnees/EOBS_fluxnet_inmet2.txt",skip=1,header=F)
names(My_meteo)= c("Year","Day","Tmax","Tmin","Precip","CO2")
head(My_meteo)
```

```
##   Year Day       Tmax      Tmin Precip      CO2
## 1 2004   1 -14.280000 -20.70000  0.306 384.4005
## 2 2004   2 -13.340000 -17.90000  0.203 384.3611
## 3 2004   3  -1.859991 -13.34000  0.401 384.3216
## 4 2004   4  -8.299994 -24.65999  0.000 384.2822
## 5 2004   5 -18.000010 -28.14001  0.157 384.2427
## 6 2004   6 -17.900000 -20.32000  0.109 384.2033
```

The columns are:

- *Year* is the year of the observation
- *Day* is the day of the year of the observation (1-365)
- *Tmax* is the daily maximum temperature ($°C$)
- *Tmin* is the daily minimum temperature ($°C$)
- *Precip* is the daily precipitation sum (cm)
- *CO2* is daily CO2 concentration (ppm)

Once you have loaded the meteorological data you must create a temporal data frame with these data (same procedure than 1a) and gap fill these data (same procedure than 1b).

```
My_meteo = mutate(My_meteo, Date = as.Date(paste(My_meteo$Year,My_meteo$Day), format='%Y %j'))
My_meteo = as_tsibble(My_meteo, index = Date)
My_meteo = My_meteo %>%
  fill_gaps(Day = 366) %>%
  tidyr::fill(Tmax, .direction = "down") %>%
  tidyr::fill(Tmin, .direction = "down") %>%
  tidyr::fill(Precip, .direction = "down") %>%
  tidyr::fill(CO2, .direction = "down") %>%
  tidyr::fill(Year, .direction = "down")
```

(3a) Find the meteorological or environmental variable (*Tmax*, *Tmin*, *Precip* or *CO2*) that correlates the most with the *GapFilled_NEP* daily time series. (1 point)

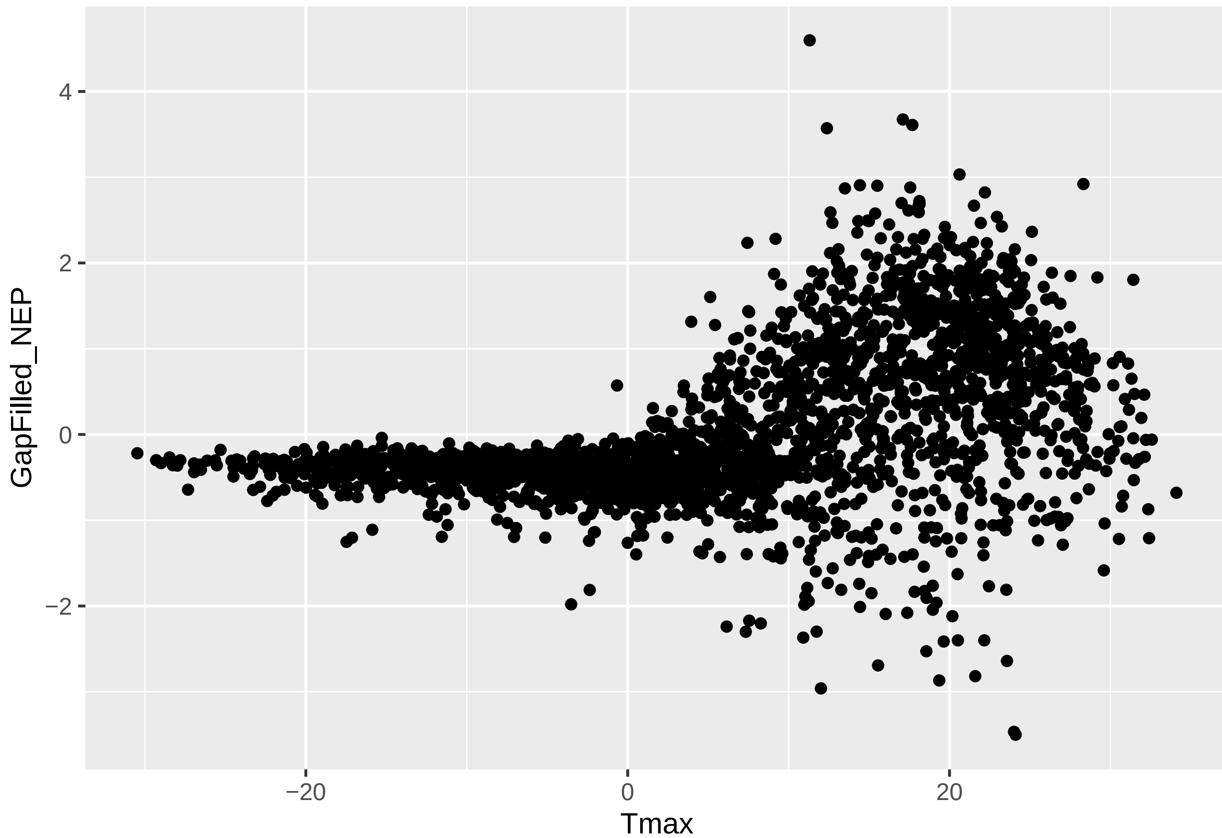
```
cor(EOBS_fluxnet$GapFilled_NEP, My_meteo[, 3:6])
```

```
##           Tmax      Tmin     Precip       CO2
## [1,] 0.4778448 0.3749279 -0.1627362 -0.150494
```

The meteorological variable that correlates the most with the *GapFilled_NEP* daily time series is the daily maximum temperature.

(3b) Join the flux and meteorological tables (*inner_join*) and plot the relationship (scatterplot) between the variable found in 3a (x-axis) and *GapFilled_NEP* (y-axis). Comment on this relationship. (1 point)

```
EOBS_fluxnet <- inner_join(EOBS_fluxnet, My_meteo)
ggplot(EOBS_fluxnet, aes(x = Tmax, y = GapFilled_NEP)) +
  geom_point()
```



No carbon uptake is possible if Tmax is lower than about 3°C. After this threshold, a positive relationship is present with an important variability.

(3c) Extend the previous GAM model (2b) including a quadratic term for the variable found in 2a. Inspect and interpret the output of the new GAM model (summary function). Compare the AIC of this new model to the models from 2a and 2b. Visualize the model predictions (fitted values) against the observed data, using: a scatterplot of GapFilled_NEP vs the used climate variable; a scatterplot of GapFilled_NEP vs DOY (Day of Year). (2 point).

```
# Fit GAMM with AR(1) structure
gamm_ar1_climate <- gamm(GapFilled_NEP ~ Tmax + I(Tmax^2) + s(Day), data = EOBS_fluxnet,
                  correlation = corARMA(p = 1, q = 0, form = ~ Date | 1))
summary(gamm_ar1_climate$lme)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: strip.offset(mf)
##        AIC      BIC    logLik
##   5158.988 5199.914 -2572.494
##
## Random effects:
##  Formula: ~Xr - 1 | g
##  Structure: pdIdnot
##              Xr1      Xr2      Xr3      Xr4      Xr5      Xr6      Xr7      Xr8
## StdDev: 5.46158 5.46158 5.46158 5.46158 5.46158 5.46158 5.46158 5.46158
##            Residual
## StdDev: 0.6849163
##
```

```
## Correlation Structure: AR(1)
##  Formula: ~Date | g
##  Parameter estimate(s):
##       Phi
## 0.2832408
## Fixed effects:  y ~ X - 1
##                    Value Std.Error   DF   t-value p-value
## X(Intercept)  0.1095416 0.0306874 2553  3.569599  0.0004
## XTmax         0.0006491 0.0027161 2553  0.238985  0.8111
## XI(Tmax^2)   -0.0003100 0.0000986 2553 -3.144608  0.0017
## Xs(Day)Fx1    0.7127692 0.3593881 2553  1.983286  0.0474
##  Correlation:
##            X(Int) XTmax  XI(T^2
## XTmax      -0.402
## XI(Tmax^2) -0.592 -0.223
## Xs(Day)Fx1 -0.014  0.121 -0.076
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -5.62494002 -0.33934038  0.05090357  0.42519102  5.22547357
##
## Number of Observations: 2557
## Number of Groups: 1
```

```
summary(gamm_ar1_climate$gam)
```

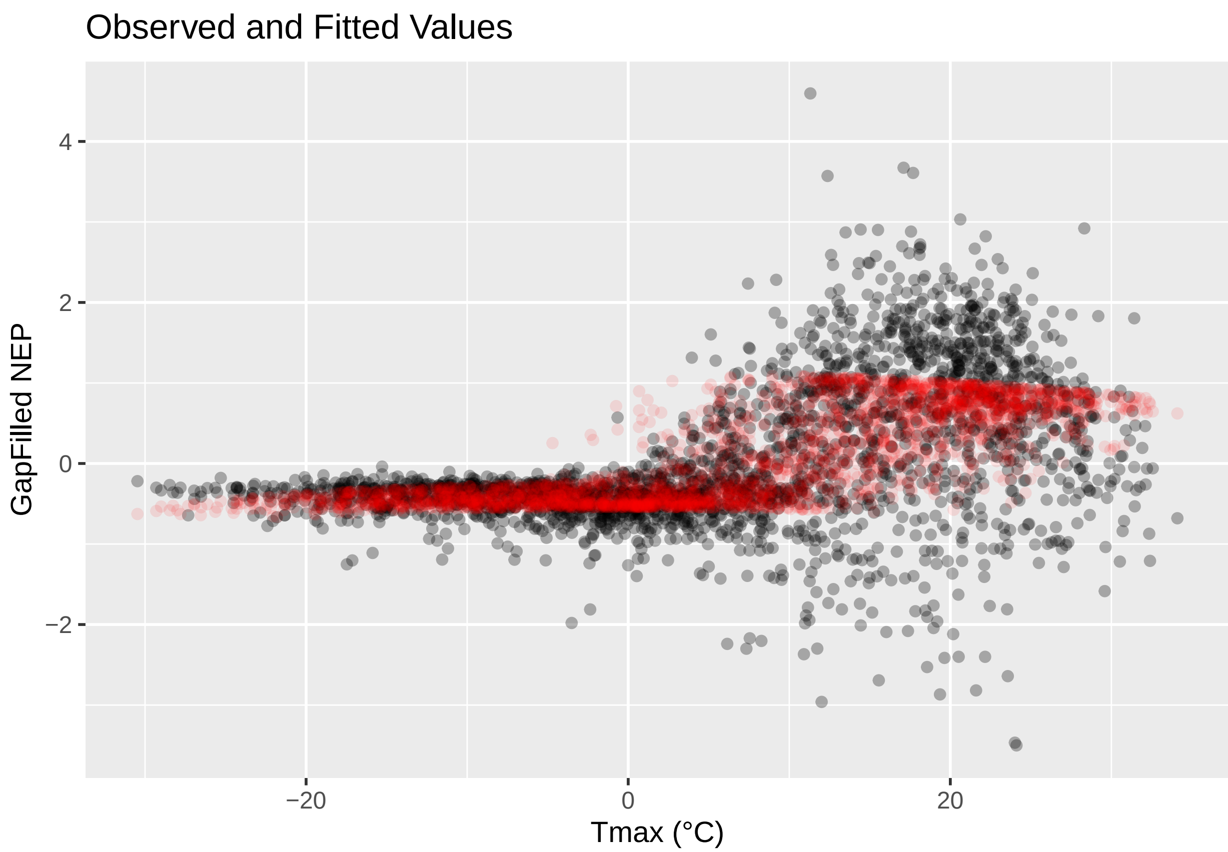
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## GapFilled_NEP ~ Tmax + I(Tmax^2) + s(Day)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.095e-01  3.068e-02   3.570 0.000363 ***
## Tmax         6.491e-04  2.716e-03   0.239 0.811100
## I(Tmax^2)   -3.100e-04  9.856e-05  -3.145 0.001679 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df     F p-value
## s(Day) 8.449  8.449 51.84  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.41
##   Scale est. = 0.46911   n = 2557
```

```
AIC(NEP_gamm1$lme, gamm_ar1$lme, gamm_ar1_climate$lme)
```

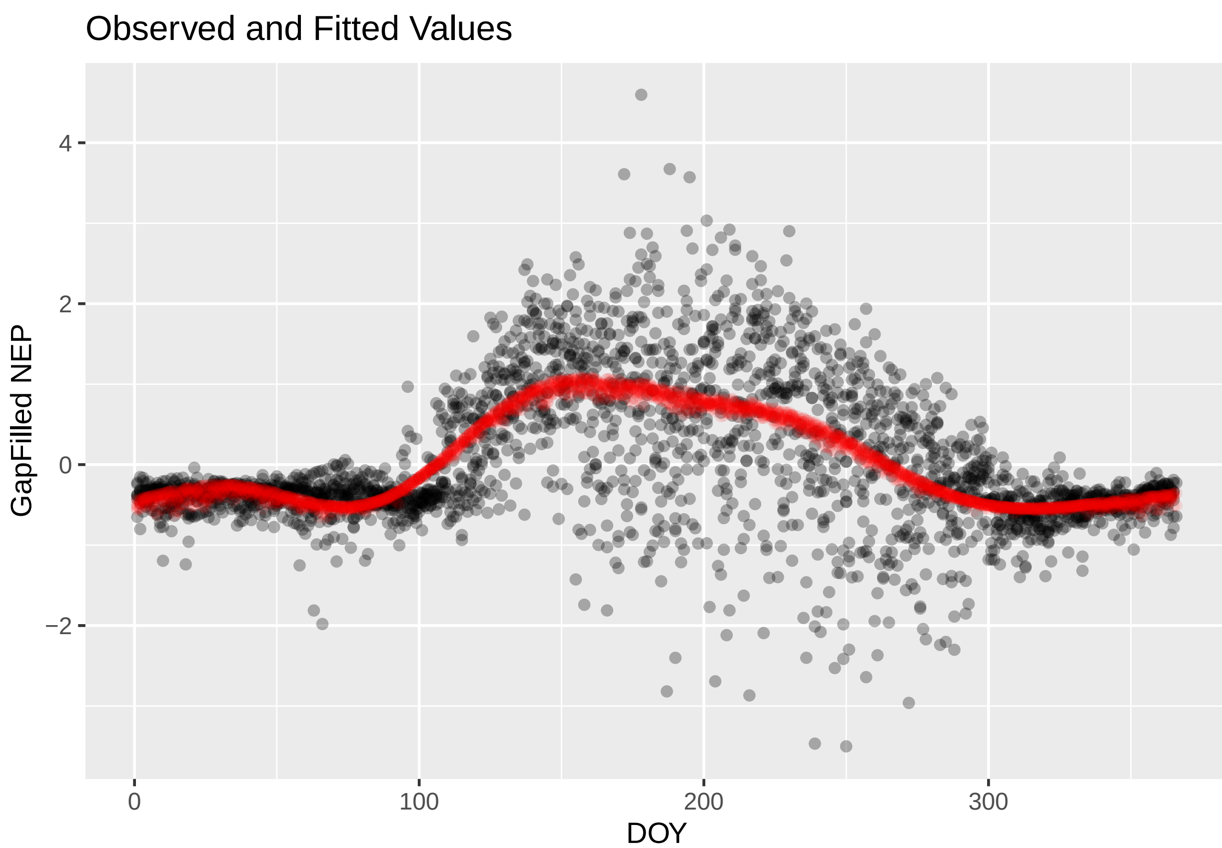
```
##                      df      AIC
## NEP_gamm1$lme         4 5389.834
## gamm_ar1$lme          5 5164.798
```

```
## gamm_ar1_climate$lme  7 5158.988
```

```r
# Create a data frame with original values and model predictions
EOBS_fluxnet$fit <- fitted(gamm_ar1_climate$lme)

# Plot observed vs fitted values
ggplot(EOBS_fluxnet, aes(x = Tmax, y = GapFilled_NEP)) +
  geom_point(alpha = 0.3) +
  geom_point(aes(y = fit), color = "red", alpha = 0.1) +
  labs(title = "Observed and Fitted Values",
       y = "GapFilled NEP",
       x = "Tmax (°C)")
```




```r
ggplot(EOBS_fluxnet, aes(x = Day, y = GapFilled_NEP)) +
  geom_point(alpha = 0.3) +
  geom_point(aes(y = fit), color = "red", alpha = 0.1) +
  labs(title = "Observed and Fitted Values",
       y = "GapFilled NEP",
       x = "DOY")
```


Observed and Fitted Values
GapFilled NEP
4
2
0
−2
0
100
200
300
DOY