Uncertainty in annual aggregates

Thomas Wutzler

Bias with $\boldsymbol{u}_{*Th}$

Motivation

We get a different annually aggregated NEE for each quantile of u_{*Th}. Usually NEE is lower with lower threshold.

For an uncertainty estimate we must compute the annual aggregate for each of the scenarios.

Getting the scenario names:

```
(tmp <- names(EProc$sGetUstarScenarios()))
uStarScens = tmp[-1]
```

```
## [1] "season" "uStar"  "U05"    "U50"    "U95"
```

Aggregate Reco for each scenario

```
fMean <- function(suffix){
  colName = paste0("Reco_",suffix)
  #colName = paste0("GPP_",suffix,"_f")
  mean(results[[colName]])
}
#fMean("U50")
meansUStar <- unlist(EProc$sApplyUStarScen(fMean))
names(meansUStar) <- uStarScens
meansUStar
```

```
##    uStar      U05      U50      U95
## 3.446351 3.374680 3.470419 3.369135
```

```
c(median(meansUStar), sd(meansUStar), 100*sd(meansUStar)/abs(median(meansUStar)))
```

```
## [1] 3.41051576 0.05093558 1.49348617
```

Here only small relative error of 1.5% in respiration introduced by unknown u_{*Th}

More robust analysis of u_{*Th} effects

For a thorough analysis, we would have needed to specify more than 3 quantiles when we estimated the u_{*Th} distribution.

But this would have multiplied the computation time.

```
EProc$sEstimateUstarScenarios(
    nSample = 200L, probs = seq(0.125,0.975, length.out = 30))
```

Random uncertainty

Random uncertainty

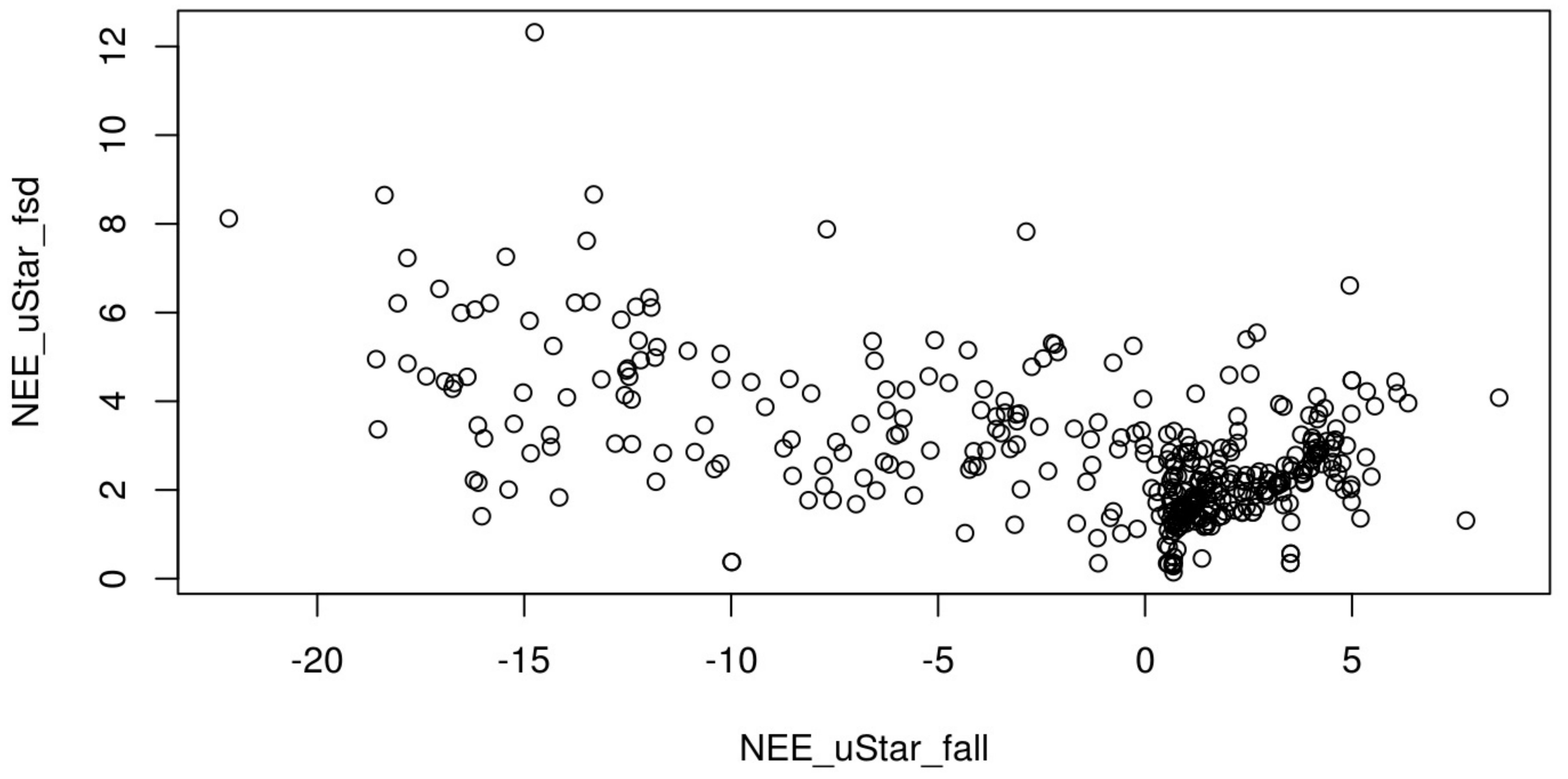
Random uncertainty is the scatter of NEE for otherwise same conditions. It includes measurement error and intrinsic stochasticity of the studied system.

There is low uncertainty during nighttime and higher uncertianty during daytime.

```
plot( NEE_uStar_fsd ~ DateTime, slice(results,400:600))
```

Uncertainty scales with flux (heteroscadastic)

```
plot( NEE_uStar_fsd ~ NEE_uStar_fall, slice(results, sample.int(nrow(results),400)))
```



Wrong aggregation without correlations

Error propagatin by adding variance results in an approximate reduction of the mean (root mean square) $\sqrt{\bar{\sigma^2}}$ by $\sqrt{n}$.

```
stat1 <- results %>% filter(NEE_uStar_fqc == 0) %>% summarise(
  nRec = sum(is.finite(NEE_uStar_f))
  , rms = sqrt(mean(NEE_uStar_fsd^2, na.rm = TRUE))
  , seMean = rms / sqrt(nRec)
  #, seMean = sqrt(sum(NEE_uStar_fsd^2))/nRec
  ) %>% select(nRec, seMean) %>% unlist()
stat1["seMean"]
```

```
##     seMean
## 0.03266103
```

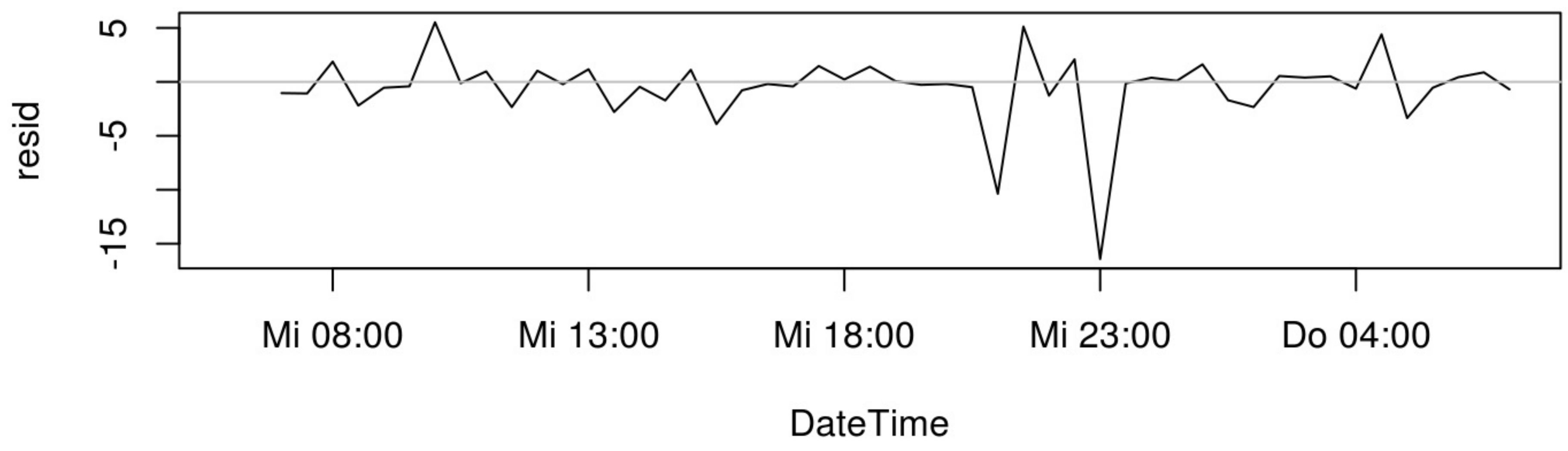
Due to the large number of records, the estimated uncertainty is very low.

Considering autocorrelation

Errors of subsequent measurements tend to have the same direction and magnitude. Subsequent errors tend to be similar.

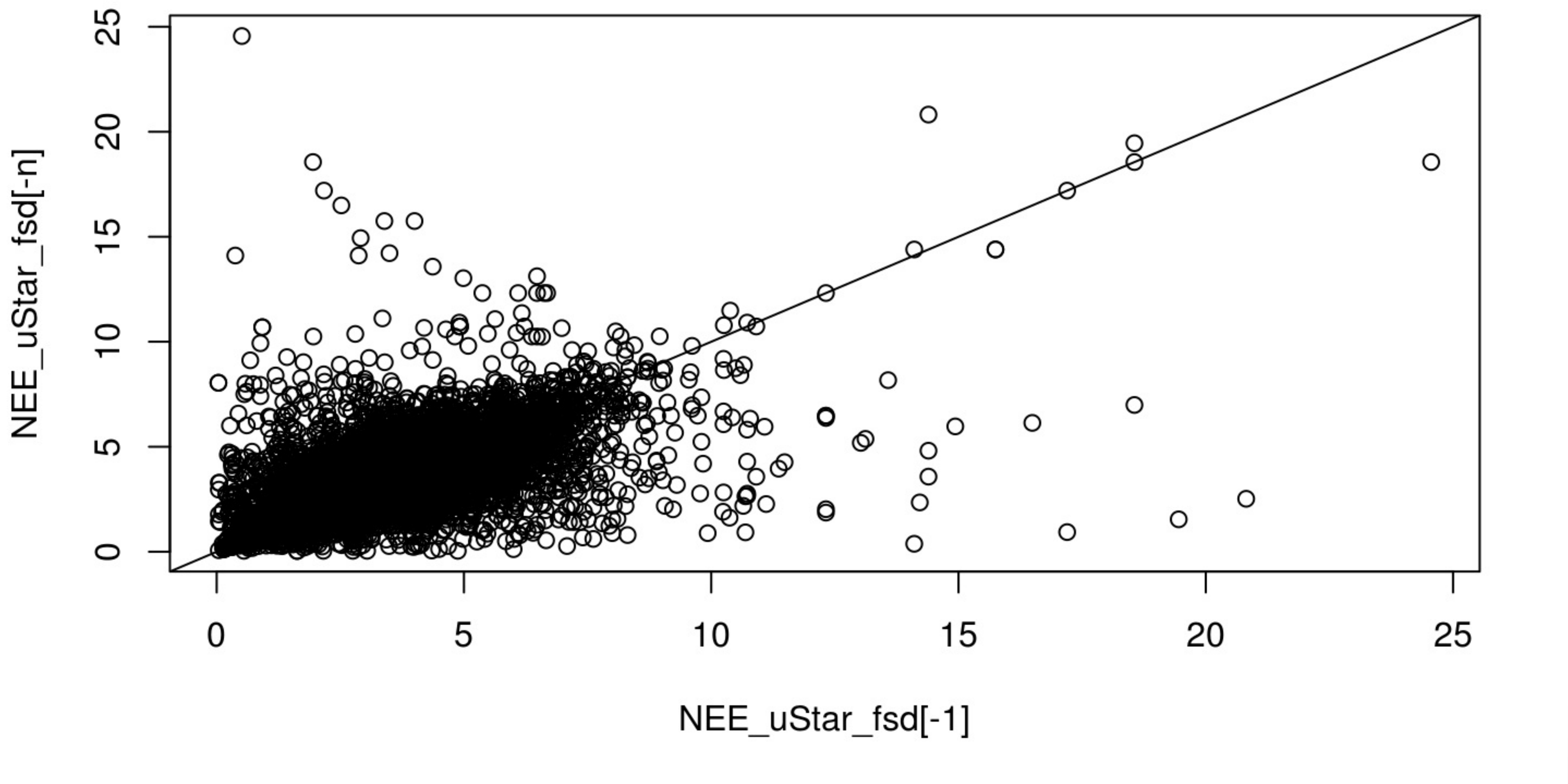
We approximate the error terms by the difference between original measurement and value from Marginal Distribution samples during gapfilling.

```
results <- results %>%
  mutate(resid = ifelse(NEE_uStar_fqc == 0, NEE_uStar_orig - NEE_uStar_fall, NA ))
```



Correlation between subsequent errors

```
n <- nrow(results)
plot(NEE_uStar_fsd[-n] ~ NEE_uStar_fsd[-1], results); abline(a = 0,b = 1)
```



Autocorrelation function

They are not independent of each other. The strenght of their dependence can be expressed by values ρ_i

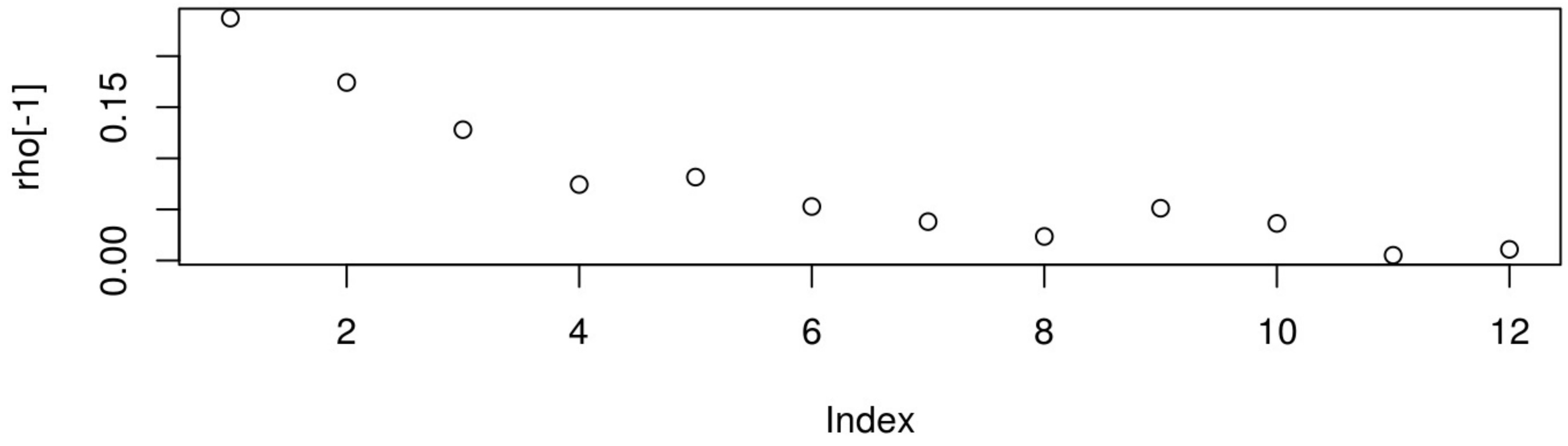
$$x_i = \rho_1 x_{i-1} + \rho_2 x_{i-2} + \ldots + \epsilon_i$$

The series ρ is called the autocorrelation function.

Empirical autocorrelation function

The series of estimates of those coefficients from the data is called the empirical autocorrelation function. But only the first terms of the empricial functions are reliable. (Take only the ones before the first negative coefficient)

```
#rho = acf(results$resid, lag.max = 20, main = "", na.action = na.exclude)
library(lognorm)
rho = computeEffectiveAutoCorr(results$resid)
plot(rho[-1])
```



Error propagation

If the observations are not independent the average of the mean does not scale by $\sqrt{n}$ but by approximately $\sqrt{n_{eff}}$ with $n_{eff} < n$.

$$\sigma(m) = \frac{\sqrt{\overline{\sigma^2}}}{\sqrt{n_{eff} - 1}}$$

Effectively fewer observations

```
library(lognorm)
nEff <- computeEffectiveNumObs(results$resid, na.rm = TRUE)
c( nEff = nEff, nObs = sum(is.finite(results$resid)))
```

```
##     nEff     nObs
## 3427.967 9688.000
```

Due to the high autocorrelation, the entire year of NEE records have only fewer effective observations.

When computing correlations or effective number of observations, make sure to use a complete time series with a records for each equidistant time step. Missing values will

Larger aggregated error

```r
stat2 <- results %>% filter(NEE_uStar_fqc == 0) %>% summarise(
  rms = sqrt(mean(NEE_uStar_fsd^2, na.rm = TRUE))
  , seMeanCor = rms / sqrt(nEff - 1)
  ) %>% select(seMeanCor) %>% unlist()
NEE <- mean(results$NEE_uStar_f)
c(stat2["seMeanCor"], stat1["seMean"], cv = stat2["seMeanCor"]/abs(NEE))
```

```
##    seMeanCor       seMean cv.seMeanCor
##   0.05491512   0.03266103   0.03396267
```

Combined uncertainty

Add in squares

The uncertainty due to random fluctuations can be assumed independent to the uncertainty introduced by the uncertain u_{*Th}. Then error propgation can be done by adding variances.

$$\sigma_{Combined}(x) = \sqrt{\sigma^2_{u_*}(x) + \sigma^2_{Random}(x)}$$

```
sdNEERandom <- sd(unlist(EProc$sApplyUStarScen(function(suffix){
  mean(results[[paste0("NEE_",suffix,"_f")]])
})))
sdNEE <- sqrt(stat2["seMeanCor"]^2 + sdNEERandom^2)
sdNEE
```

```
##  seMeanCor
## 0.05598646
```