PV data - Parameter estimation and sample generation

Tuesday, May 17, 2016

We consider the data given in PVdata2.csv. Let's generate a matrix where each line represents a day and each column represents one minute of this day:

```
PV1<-PVdata[1:1440,1] #24*60=1440
for (i in 1:30) {
  PV1<-cbind(PV1,PVdata[((i*1440)+1):((i+1)*1440),1])
}
for(j in 2:12){
  for(i in 1:31){
    PV1<-cbind(PV1,PVdata[(((i-1)*1440)+1):(i*1440),j])
  }
}
PV1<-t(PV1)
```

1 Normal distribution

1.1 Parameter estimation

1.1.1 Parameter estimation - multivariate for 1h intervals

We estimate the values of expectation and the covariance matrix under the assumption of a **multivariate** normal distribution for intervals of 1h:

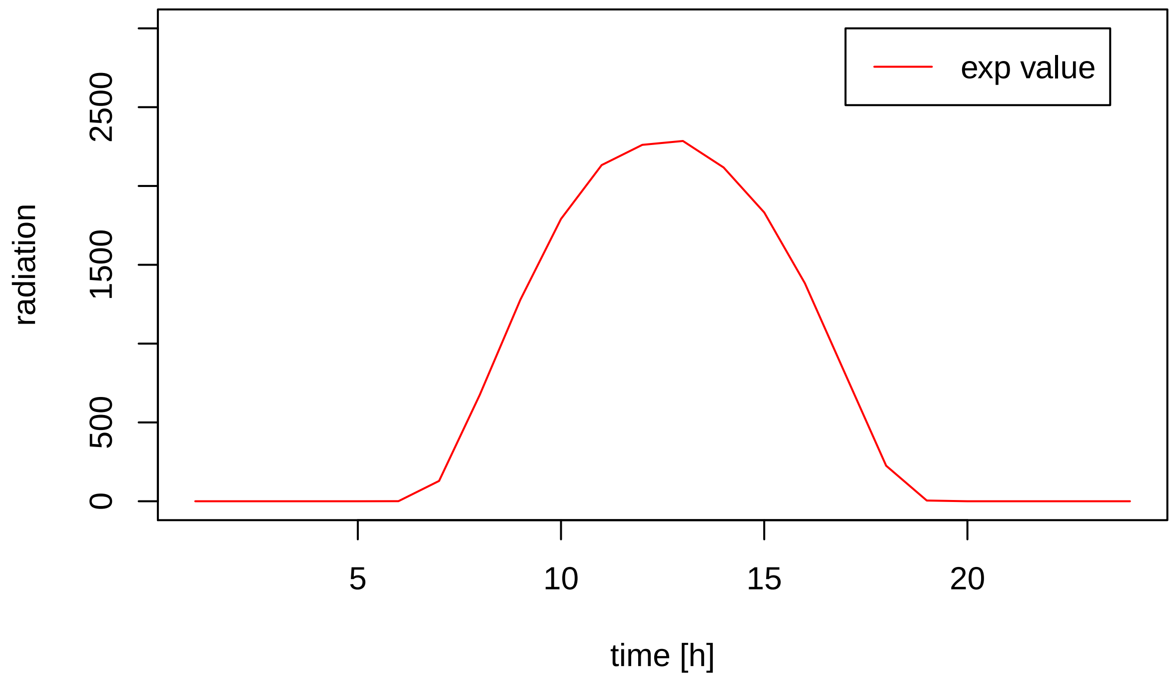
```
PV1h<-matrix(rep(0,8928),nrow=372) #hourly values -> take means, 24*372=8928
for(i in 1:372){
  for (j in 1:24){
    PV1h[i,j]<-mean(PV1[i,((j-1)*60+1):(j*60)])
  }
}

estimates_n_dep<-mlest(PV1h) #under assumption of no independence: hourly means and covariance matrix
```

```
## Warning: NA/Inf durch größte positive Zahl ersetzt
```

Let's visualize the expected value we estimated:

```
plot(estimates_n_dep$muhat,xlab="time [h]", ylab="radiation", type="l", col="red", ylim=c(0,3000))
legend(17,3000, c("exp value"), col=c("red"), lty=c(1))
```



1.1.2 Parameter estimation - independent for 15min intervals

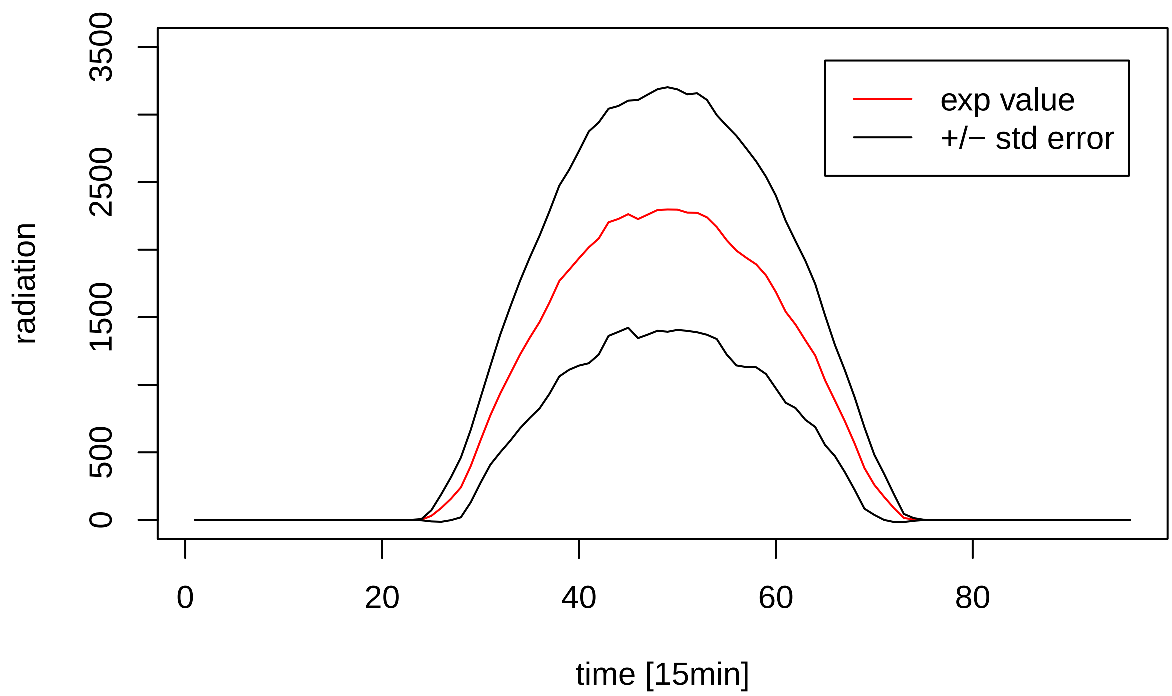
Using the package above it is not possible to analyse a multivariate normal distribution with more than 50 variables, hence we continue the analysis with the assumption of $4 \cdot 24$ **independently** distributed variables for every 15 minutes.

```
PV1quh<-matrix(rep(0,(8928*4)),nrow=372) #quarter hourly values -> take means, 24*372=8928
for(i in 1:372){
  for (j in 1:(24*4)){
    PV1quh[i,j]<-mean(PV1[i,((j-1)*15+1):(j*15)])
  }
}

estimates_n_ind<-matrix(rep(0,3*96),nrow=3) #under assumption of independence, 4*24=96=T
for (i in 1:(24*4)){
  fit_n_ind<-fitdistr(PV1quh[,i],"normal")
  estimates_n_ind[1,i]<-fit_n_ind$estimate[1] #estimate of mean (mu)
  estimates_n_ind[2,i]<-fit_n_ind$estimate[2] #estimate of std error (sigma)
  estimates_n_ind[3,i]<-fit_n_ind$loglik
  }
write.csv(estimates_n_ind[1:2,],"estimates_normal_independent.csv", row.names=c("mean","stderror"))
```

Let's visualize the expected value and standard errors we estimated:

```r
plot(estimates_n_ind[1,],xlab="time [15min]", ylab="radiation", type="l", col="red", ylim=c(0,3500))
lines(estimates_n_ind[1,]+estimates_n_ind[2,],type="l")
lines(estimates_n_ind[1,]-estimates_n_ind[2,],type="l")
legend(65,3400, c("exp value", "+/- std error"), col=c("red","black"), lty=c(1,1))
```



And let's check the loglikelihood:

```r
estimates_n_ind[3,]
```

```
##  [1]      Inf      Inf      Inf      Inf      Inf      Inf      Inf
##  [8]      Inf      Inf      Inf      Inf      Inf      Inf      Inf
## [15]      Inf      Inf      Inf      Inf      Inf      Inf      Inf
## [22]      Inf   -82.24 -1109.17 -1911.06 -2246.82 -2413.72 -2536.27
## [29] -2607.97 -2667.69 -2723.19 -2788.29 -2835.45 -2871.84 -2903.42
## [36] -2931.13 -2951.57 -2968.07 -2985.47 -3011.81 -3040.48 -3040.98
## [43] -3032.99 -3030.87 -3032.86 -3050.50 -3053.48 -3055.74 -3060.39
## [50] -3054.27 -3047.92 -3051.92 -3045.31 -3027.56 -3035.36 -3036.82
## [57] -3018.66 -2996.36 -2980.58 -2971.82 -2950.35 -2918.36 -2900.20
## [64] -2860.74 -2824.06 -2767.12 -2735.25 -2699.33 -2650.91 -2539.11
## [71] -2439.76 -2250.19 -1792.12 -1386.38  -581.53      Inf      Inf
## [78]      Inf      Inf      Inf      Inf      Inf      Inf      Inf
## [85]      Inf      Inf      Inf      Inf      Inf      Inf      Inf
## [92]      Inf      Inf      Inf      Inf      Inf
```

```
logn<-estimates_n_ind[3,]
logn<-logn[logn<Inf]
mean(logn)
```

```
## [1] -2670
```

1.1.3 Parameter estimation - sum of RVs

We now estimate the parameters under the assumption that the radiation values are distributed according to $\frac{1}{m} \cdot (X + Y + Z + W + ...)$ where $X, Y, Z, W, ...$ are normally distributed.

Let X be distributed with the parameters we estimated for the **independently** normal distribution for intervals of 15min.
For every interval of 1h, let Y be the random variable distributed according to a univariate normal distribution.

```
estimates_n_sum2<-matrix(rep(0,2*24),nrow=2) #under assumption of independence
for (i in 1:(24)){
  estimates_n_sum2[1,i]<-fitdistr(PV1h[,i],"normal")$estimate[1] #estimate of mean (mu)
  estimates_n_sum2[2,i]<-fitdistr(PV1h[,i],"normal")$estimate[2] #estimate of std error (sigma)
  }
```

For every interval of 3h, let Z be the random variable distributed according to a univariate normal distribution.
For every interval of 4h, let W be the random variable distributed according to a univariate normal distribution, and so on. . .

```
PV13h<-matrix(rep(0,(8*372)),nrow=372) #3h values -> take means, 8*372=2976
for(i in 1:372){
  for (j in 1:(8)){
    PV13h[i,j]<-mean(PV1h[i,((j-1)*3+1):(j*3)])
  }
}
estimates_n_sum3<-matrix(rep(0,2*8),nrow=2) #under assumption of independence
for (i in 1:(8)){
  estimates_n_sum3[1,i]<-fitdistr(PV13h[,i],"normal")$estimate[1] #estimate of mean
  estimates_n_sum3[2,i]<-fitdistr(PV13h[,i],"normal")$estimate[2] #estimate of std er
  }

PV14h<-matrix(rep(0,(6*372)),nrow=372) #4h values -> take means
for(i in 1:372){
  for (j in 1:(6)){
    PV14h[i,j]<-mean(PV1h[i,((j-1)*4+1):(j*4)])
  }
}
estimates_n_sum4<-matrix(rep(0,(2*6)),nrow=2) #under assumption of independence
for (i in 1:(6)){
  estimates_n_sum4[1,i]<-fitdistr(PV14h[,i],"normal")$estimate[1] #estimate of mean
  estimates_n_sum4[2,i]<-fitdistr(PV14h[,i],"normal")$estimate[2] #estimate of std er
  }

PV15h<-matrix(rep(0,(12*372)),nrow=372) #2h values -> take means
for(i in 1:372){
  for (j in 1:(12)){
```

```
    PV15h[i,j]<-mean(PV1h[i,((j-1)*2+1):(j*2)])
  }
}
estimates_n_sum5<-matrix(rep(0,2*12),nrow=2) #under assumption of independence
for (i in 1:(12)){
  estimates_n_sum5[1,i]<-fitdistr(PV15h[,i],"normal")$estimate[1] #estimate of mean
  estimates_n_sum5[2,i]<-fitdistr(PV15h[,i],"normal")$estimate[2] #estimate of std er
  }

PV16h<-matrix(rep(0,(48*372)),nrow=372) #30min values -> take means
for(i in 1:372){
  for (j in 1:(48)){
    PV16h[i,j]<-mean(PV1quh[i,((j-1)*2+1):(j*2)])
  }
}
estimates_n_sum6<-matrix(rep(0,2*48),nrow=2) #under assumption of independence
for (i in 1:(48)){
  estimates_n_sum6[1,i]<-fitdistr(PV16h[,i],"normal")$estimate[1] #estimate of mean
  estimates_n_sum6[2,i]<-fitdistr(PV16h[,i],"normal")$estimate[2] #estimate of std er
  }
```

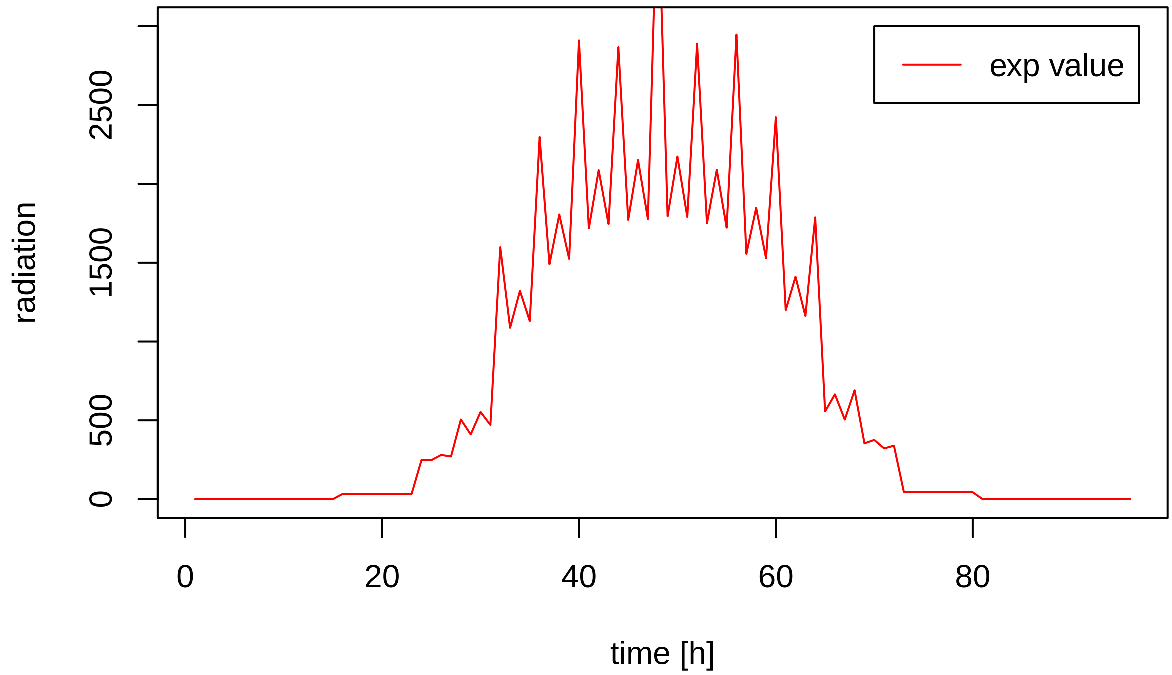
Now let's compute mean and standard error of the distribution of $\frac{1}{4}X + Y + Z + W$:

```
mu<-rep(0,24*4)
std<-rep(0,24*4)
for(i in 1:6){
  mu[((i-1)*4*4):(i*4*4)]<-mu[((i-1)*4*4):(i*4*4)]+estimates_n_sum4[1,i]
  std[((i-1)*4*4):(i*4*4)]<-std[((i-1)*4*4):(i*4*4)]+estimates_n_sum4[2,i]
  }
for(i in 1:12){
  mu[((i-1)*2*4):(i*2*4)]<-mu[((i-1)*2*4):(i*2*4)]+estimates_n_sum5[1,i]
  std[((i-1)*2*4):(i*2*4)]<-std[((i-1)*2*4):(i*2*4)]+estimates_n_sum5[2,i]
  }
for(i in 1:48){
  mu[((i-1)*2):(i*2)]<-mu[((i-1)*2):(i*2)]+estimates_n_sum6[1,i]
  std[((i-1)*2):(i*2)]<-std[((i-1)*2):(i*2)]+estimates_n_sum6[2,i]
  }
for(i in 1:8){
  mu[((i-1)*3*4):(i*3*4)]<-mu[((i-1)*3*4):(i*3*4)]+estimates_n_sum3[1,i]
  std[((i-1)*3*4):(i*3*4)]<-std[((i-1)*3*4):(i*3*4)]+estimates_n_sum3[2,i]
  }
for(i in 1:24){
  mu[((i-1)*4):(i*4)]<-mu[((i-1)*4):(i*4)]+estimates_n_sum2[1,i]
  std[((i-1)*4):(i*4)]<-std[((i-1)*4):(i*4)]+estimates_n_sum2[2,i]
  }
for(i in 1:24*4){
  mu[i]<-mu[i]+estimates_n_ind[1,i]
  std[i]<-std[i]+estimates_n_ind[2,i]
  }

mu<-mu*1/6
std<-std*1/6
```

Let's visualize the expected value we estimated:

```r
plot(mu,xlab="time [h]", ylab="radiation", type="l", col="red", ylim=c(0,3000))
legend(70,3000, c("exp value"), col=c("red"), lty=c(1))
```



1.2 Sample generation

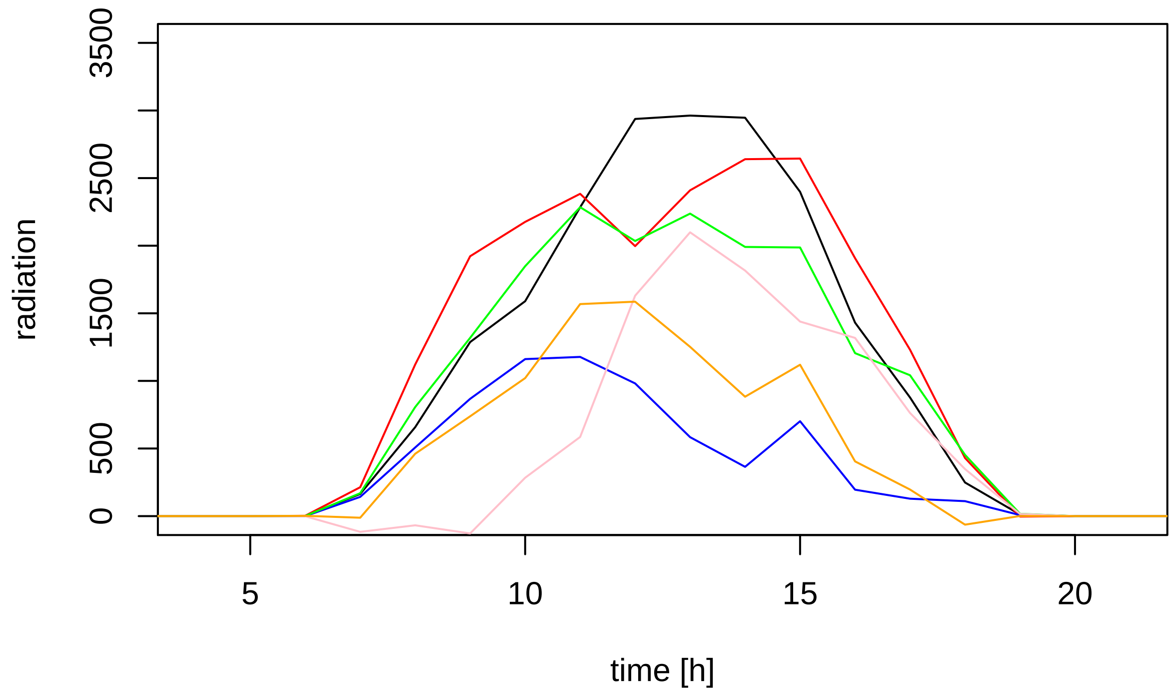
1.2.1 Sample generation - dependent for 1h intervals

We generate a sample of size N of a **multivariate** normal distribution with the parameters estimated above:

```r
N<-1000
Nsample<-mvrnorm(n=N, estimates_n_dep$muhat, estimates_n_dep$sigmahat)
```

To visualize, the first three realizations that were generated, look like this:

```r
plot(Nsample[1,], type="l", xlim=c(4,21), ylab="radiation", xlab="time [h]", ylim=c(0,3500))
lines(Nsample[2,], type="l", col="red")
lines(Nsample[3,], type="l", col="blue")
lines(Nsample[4,], type="l", col="green")
lines(Nsample[5,], type="l", col="pink")
lines(Nsample[6,], type="l", col="orange")
```



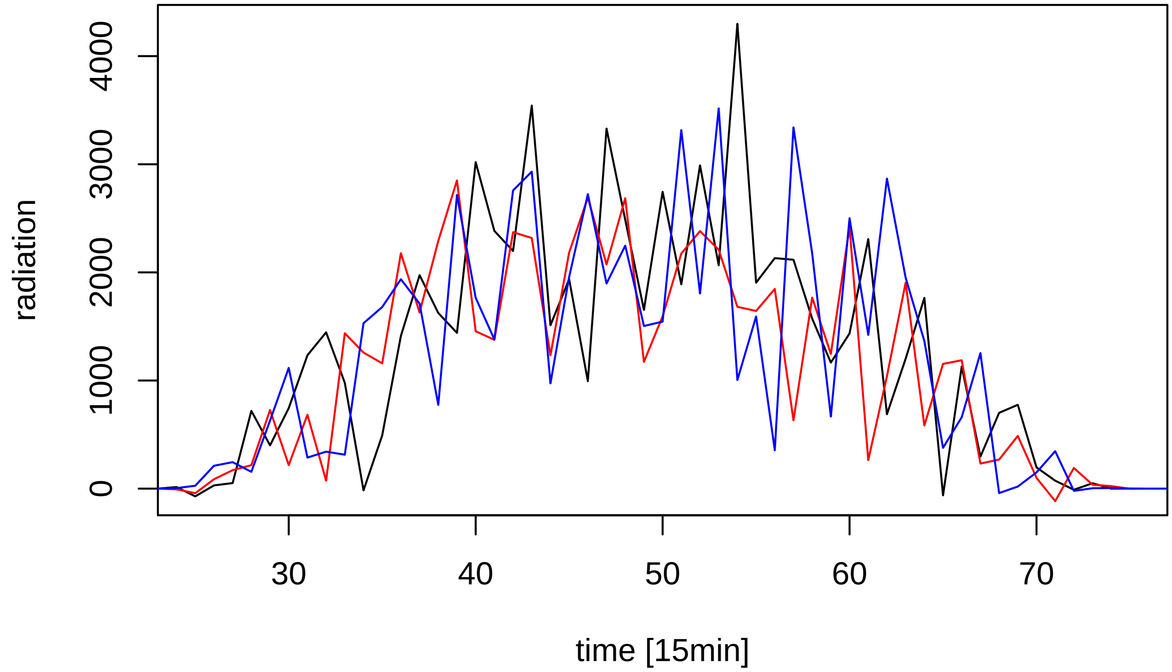
1.2.2 Sample generation - independent for 15min intervals

We generate a sample of N **independently** identically distributed random variables of the normal distribution with the parameters estimated above:

```
N<-10000
N_ind_sample<-matrix(rep(0,24*4*N), ncol=24*4) # 1 column for 1 time interval, 1 row for 1 realization
for(i in 1:N){
N_ind_sample[i,]<-rnorm(24*4,estimates_n_ind[1,], estimates_n_ind[2,])
}
write.csv(N_ind_sample,"sample_normal_independent.csv", row.names=FALSE)
```

To visualize, the first three realizations that were generated, look like this:

```
plot(N_ind_sample[1,], type="l", xlim=c(25,75), ylab="radiation", xlab="time [15min]")
lines(N_ind_sample[2,], type="l", col="red")
lines(N_ind_sample[3,], type="l", col="blue")
```



2 Weibull distribution

2.1 Parameter estimation

2.1.1 Parameter estimation - independent for 15min intervals

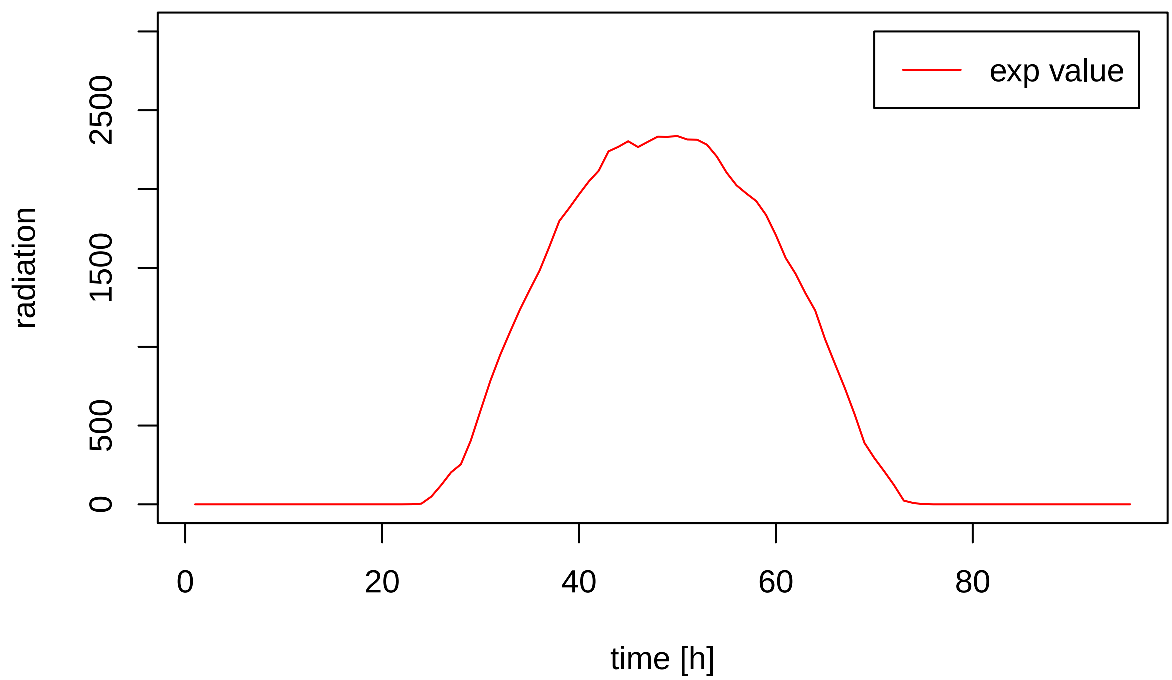
We assume **independently** distributed values for every 15 minutes and estimate a and b called the *shape parameter* and the *scale parameter*.

```
estimates_w_ind<-matrix(rep(0,3*96),nrow=3) #under assumption of independence
for (i in 1:(4*24)){
  coli<-PV1quh[,i]
  coli<-coli[coli>1E-6]
  if(length(coli)>0){
    fit_w_ind<-fitdist(coli,"weibull", lower=c(1E-6,1E-6))
  estimates_w_ind[1,i]<-fit_w_ind$estimate[1] #estimate of shape paramter
  estimates_w_ind[2,i]<-fit_w_ind$estimate[2] #estimate of scale parameter
  estimates_w_ind[3,i]<-fit_w_ind$loglik
  }
  else {
```

```
    estimates_w_ind[1,i]<-1
    estimates_w_ind[2,i]<-1E-6
  }
  }
write.csv(estimates_w_ind[1:2,],"estimates_weibull_independent.csv", row.names=c("shape","scale"))
```

Let's visualize the expected value we estimated:

```
muw<-rep(0,24*4)
for(i in 1:(24*4)){
muw[i]=estimates_w_ind[2,i]*gamma(1+1/estimates_w_ind[1,i])}
plot(muw,xlab="time [h]", ylab="radiation", type="l", col="red", ylim=c(0,3000))
legend(70,3000, c("exp value"), col=c("red"), lty=c(1))
```



And let's check the loglikelihoods:

```
estimates_w_ind[3,]
```

```
##  [1]     0.00     0.00     0.00     0.00     0.00     0.00     0.00
##  [8]     0.00     0.00     0.00     0.00     0.00     0.00     0.00
## [15]     0.00     0.00     0.00     0.00     0.00     0.00     0.00
## [22]     0.00   -12.83  -334.98 -1126.07 -1597.00 -2064.46 -2367.24
## [29] -2524.43 -2614.82 -2666.83 -2727.62 -2775.18 -2817.14 -2849.89
## [36] -2871.86 -2881.96 -2897.33 -2917.56 -2944.05 -2973.73 -2971.14
## [43] -2958.27 -2951.73 -2953.06 -2975.15 -2978.53 -2981.83 -2990.65
```

```
## [50] -2979.06 -2971.17 -2976.18 -2967.05 -2949.92 -2965.37 -2969.22
## [57] -2949.12 -2924.04 -2914.04 -2911.17 -2887.34 -2860.09 -2850.61
## [64] -2807.21 -2764.84 -2716.49 -2683.66 -2633.91 -2543.22 -2382.11
## [71] -2010.59 -1551.94  -899.98  -411.81   -83.70     0.00     0.00
## [78]     0.00     0.00     0.00     0.00     0.00     0.00     0.00
## [85]     0.00     0.00     0.00     0.00     0.00     0.00     0.00
## [92]     0.00     0.00     0.00     0.00     0.00
```

```r
logw<-estimates_w_ind[3,]
logw<-logw[logw<0]
mean(logw)
```

```
## [1] -2496
```

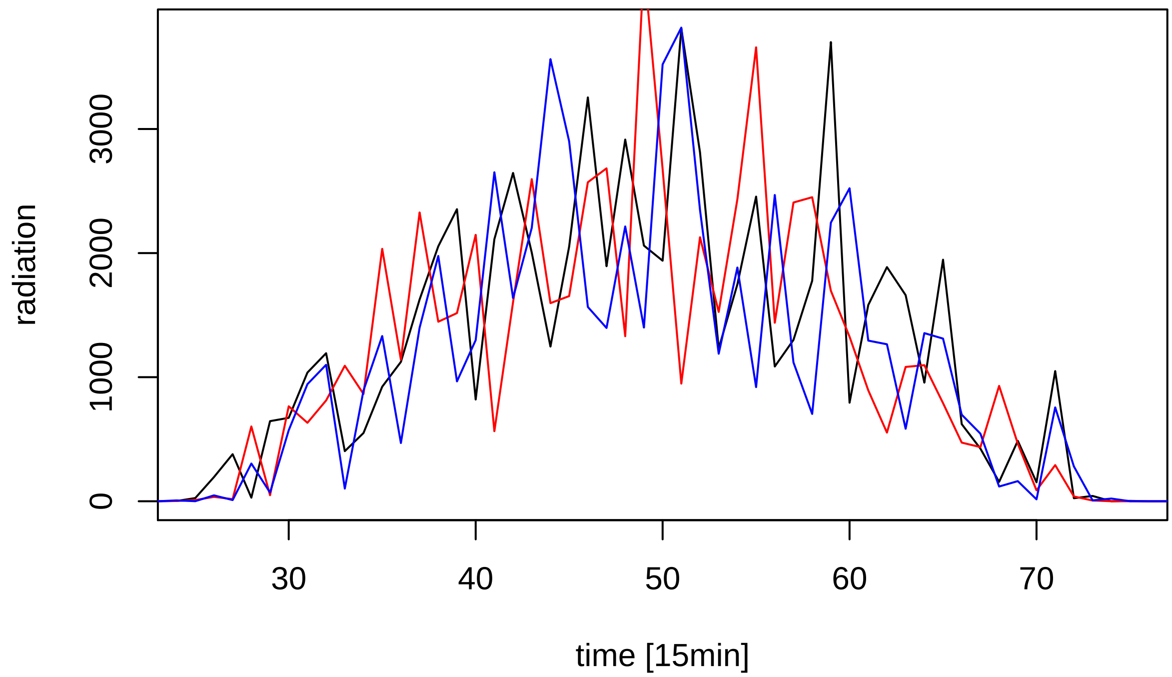
2.2 Sample generation

2.2.1 Sample generation - independent for 15min intervals

```r
N<-10000
W_ind_sample<-matrix(rep(0,24*4*N), ncol=24*4) # 1 column for 1 time interval, 1 row for 1 realization
for(i in 1:N){
W_ind_sample[i,]<-rweibull(24*4,estimates_w_ind[1,], estimates_w_ind[2,])
}
write.csv(W_ind_sample,"sample_weibull_independent.csv", row.names=FALSE)
```

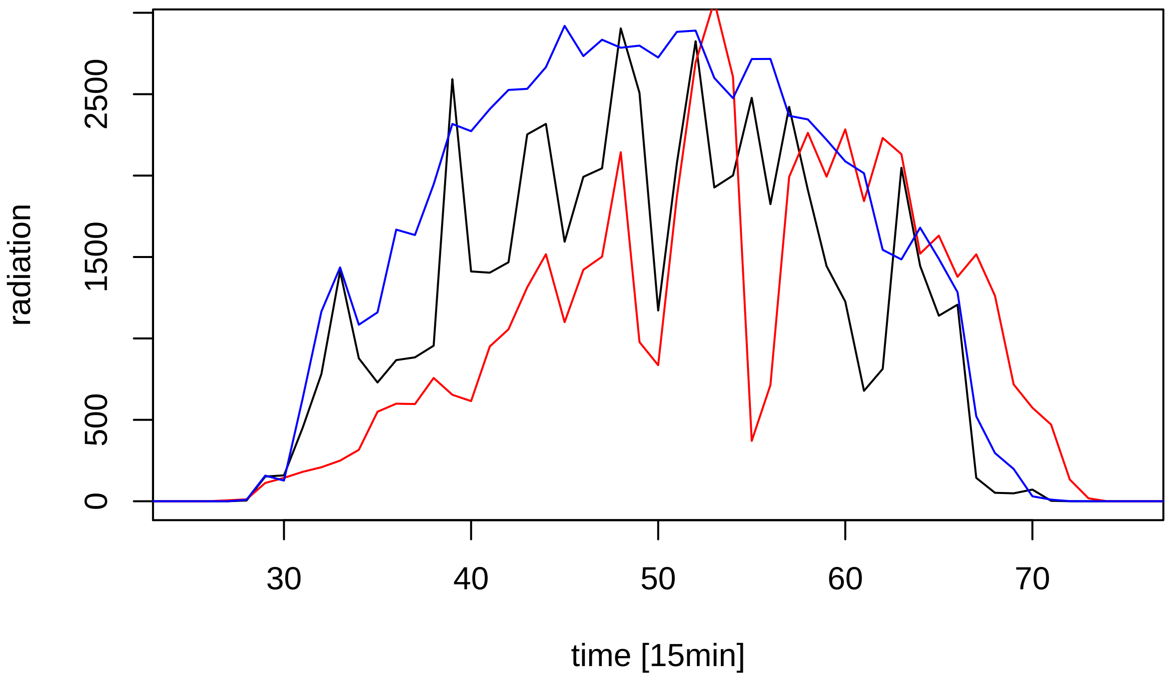
To visualize, the first three realizations that were generated, look like this:

```r
plot(W_ind_sample[1,], type="l", xlim=c(25,75), ylab="radiation", xlab="time [15min]")
lines(W_ind_sample[2,], type="l", col="red")
lines(W_ind_sample[3,], type="l", col="blue")
```



Just to visualize what our original data look like:

```
plot(PV1quh[10,], type="l", xlim=c(25,75), ylab="radiation", xlab="time [15min]")
lines(PV1quh[88,], type="l", col="red")
lines(PV1quh[9,], type="l", col="blue")
```


radiation
0
500
1500
2500
30
40
50
60
70
time [15min]