PV data - Parameter estimation and sample generation

Martina Kuchlbauer

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:

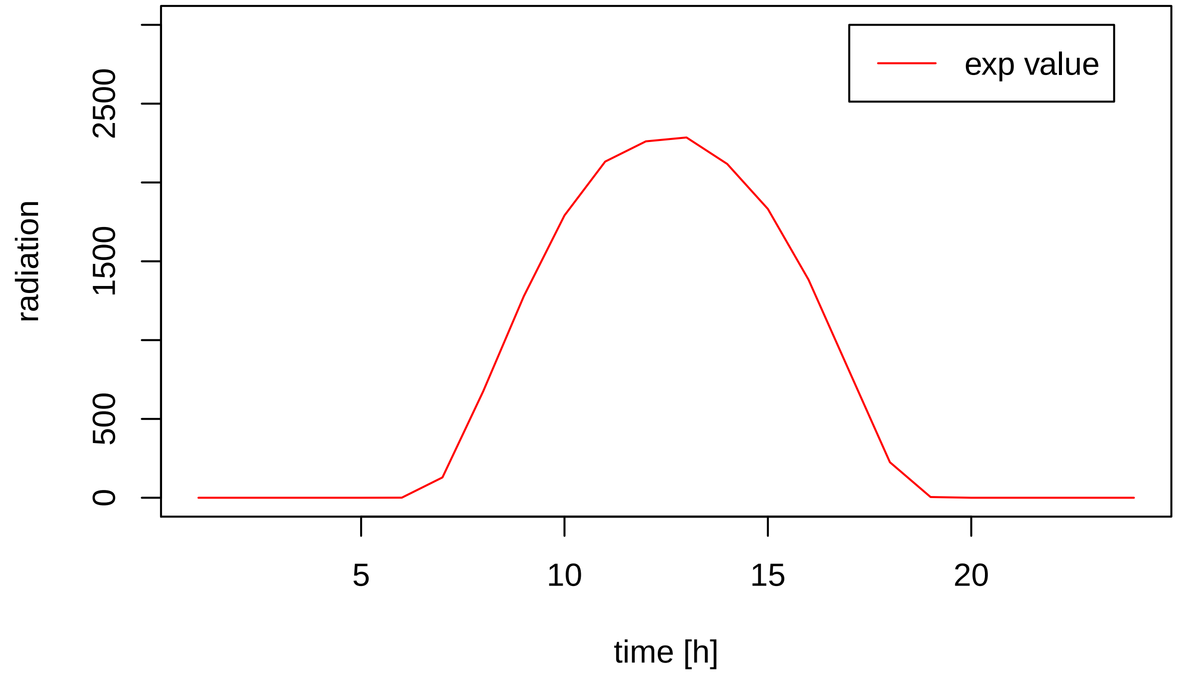
```
PV1h<-matrix(rep(0,8928),nrow=372) #hourly values -> take means, 24*372=8928
for(i in 1:372){
  for (j in 1:24){
    PV1h[i,j]<-mean(PV1[i,((j-1)*60+1):(j*60)])
  }
}

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

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

```
#estimates_n_dep$muhat #estimate of mean (mu)
#estimates_n_dep$sigmahat #estimate of covariance matrix (sigma)
```

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))
```



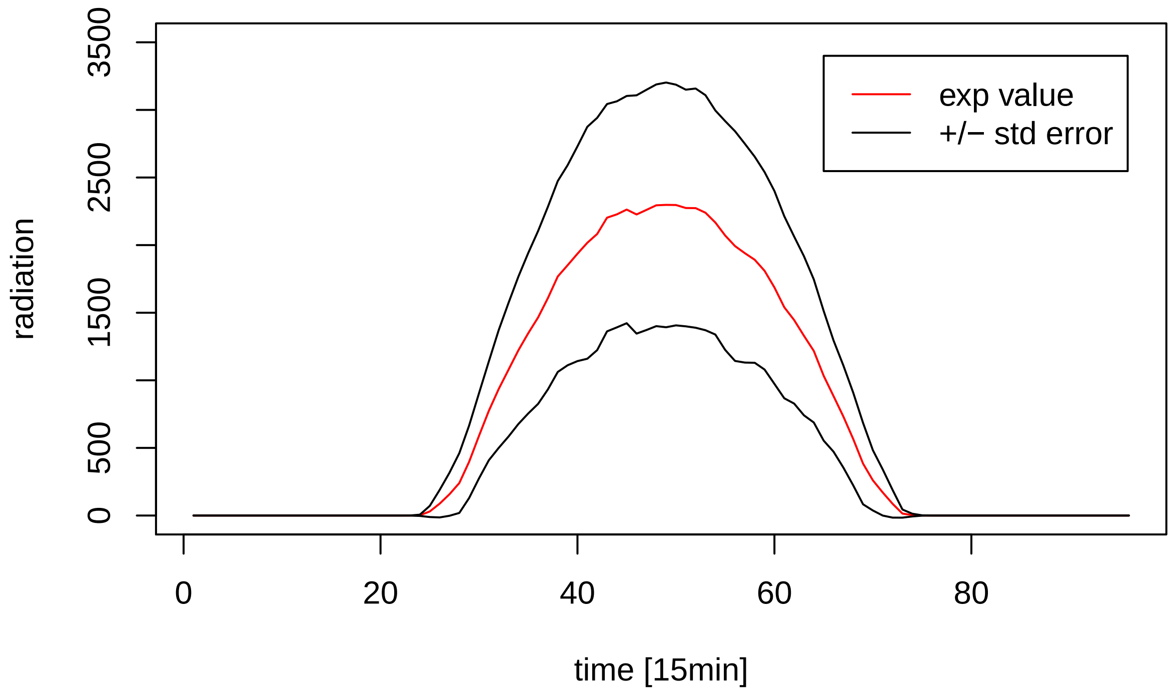
1.1.2 Parameter estimation - independent for 15min intervals

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(0,2*96),nrow=2) #under assumption of independence, 4*24=96=T
for (i in 1:(24*4)){
  estimates_n_ind[1,i]<-fitdistr(PV1quh[,i],"normal")$estimate[1] #estimate of mean (mu)
  estimates_n_ind[2,i]<-fitdistr(PV1quh[,i],"normal")$estimate[2] #estimate of std error (sigma)
  }
#estimates_n_ind
```

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))
```



1.1.3 Parameter estimation - sum of RVs

We now estimate the parameters under the assumption that the radiation values are distributed according to $X + Y + Z + W$ where X, Y, Z and W are normally distributed.

Let X be distributed with the parameters we estimated for the multivariate normal distribution for intervals of 1h.
For every interval of 4h, let Y be the random variable distributed according to a multivariate normal distribution for subintervals of 15min of these 4h.

```r
# mu<-rep(estimates_n_dep$muhat[1],4)
# for (i in 2:24){
#   cbind(mu, rep(estimates_n_dep$muhat[i],4))
# }
#
# for (i in 1:6){
#   estimates_n_conv[i]<-mlest(PV1h[,(i*4*4-1):(i*4*4)]) # 4h Intervalle
#   mu
# }
```

Let's visualize the expected value we estimated:

```
# plot(estimates_n_conv$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))
```

1.2 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:

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

To validate this sample generation, we estimate its distribution and standard error and compare them with the parameters we used:

```
test<-mlest(Nsample)
diffmu<-estimates_n_dep$muhat-test$muhat
diffsigma<-estimates_n_dep$sigmahat-test$sigmahat
max(diffmu)
```

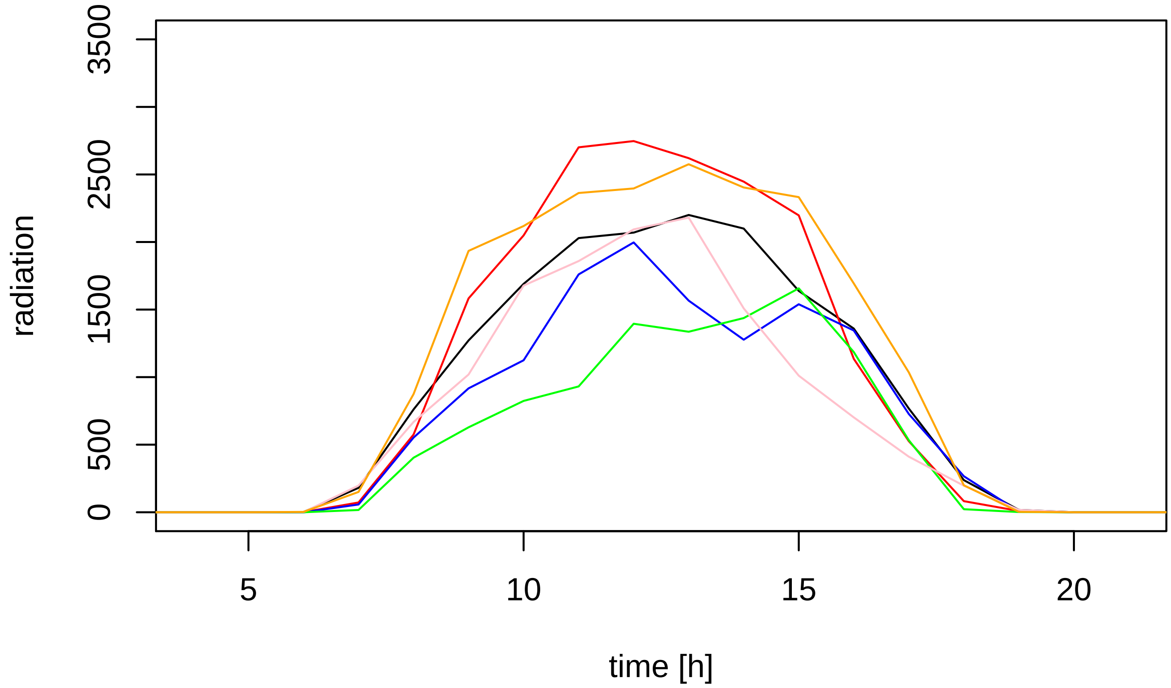
```
## [1] 0.1584
```

```
max(diffsigma)
```

```
## [1] 2867
```

These values show us that a sample size of $N = 1000$ is probably not enough.
To visualize, the first three realizations that were generated, look like this:

```
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")
```



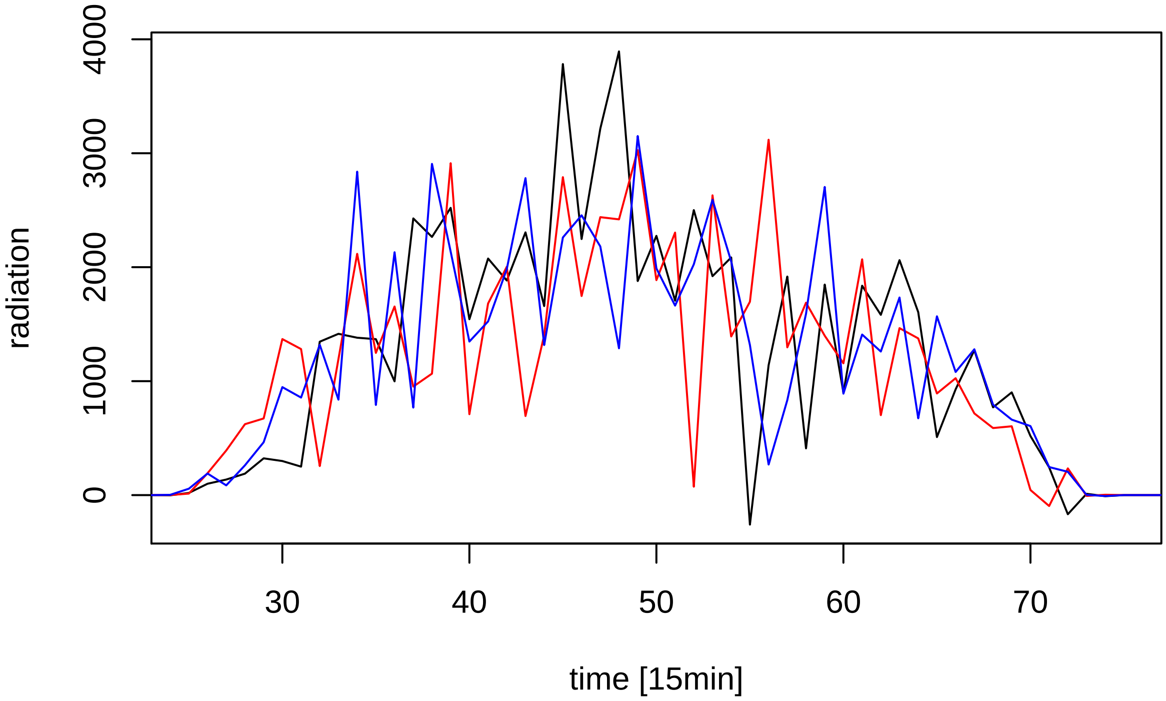
1.2.2 Sample generation - independent for 15min intervals

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,])
}
```

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(0,2*96),nrow=2) #under assumption of independence
for (i in 1:(4*24)){
  coli<-PV1quh[,i]
  coli<-coli[coli>0.001]
  if(length(coli)>0){
  estimates_w_ind[1,i]<-fitdist(coli,"weibull", lower=c(0.001,0.001))$estimate[1] #estimate of shape pa
  estimates_w_ind[2,i]<-fitdist(coli,"weibull", lower=c(0.001,0.001))$estimate[2] #estimate of scale pa
  }
  else {
    estimates_w_ind[1,i]<-0
    estimates_w_ind[2,i]<-0
  }
  }
estimates_w_ind
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [2,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]  [,23]  [,24]
## [1,]     0     0     0     0     0     0     0     0     0 0.8443 0.7082
## [2,]     0     0     0     0     0     0     0     0     0 0.4290 3.7979
##       [,25]    [,26]    [,27]    [,28]   [,29]   [,30]   [,31]    [,32]
## [1,]  0.833   0.7374   0.5262   0.8713   1.459   1.941   2.339    2.404
## [2,] 44.844 101.1215 111.0588 237.0161 445.277 671.108 884.892 1070.184
##         [,33]   [,34]    [,35]    [,36]    [,37]    [,38]    [,39]
## [1,]    2.481    2.53    2.551    2.635    2.859    3.076    3.032
## [2,] 1233.703 1392.08 1533.633 1669.478 1836.173 2010.368 2103.331
##         [,40]    [,41]    [,42]    [,43]    [,44]    [,45]    [,46]
## [1,]    2.932    2.808    2.918    3.282    3.395    3.453    3.129
## [2,] 2203.826 2300.371 2372.735 2497.168 2525.011 2562.038 2533.484
##         [,47]    [,48]    [,49]    [,50]    [,51]    [,52]    [,53]
## [1,]    3.142    3.188    3.118    3.208    3.257    3.235    3.264
## [2,] 2570.124 2604.948 2606.699 2608.304 2582.168 2581.541 2545.140
##         [,54]    [,55]    [,56]    [,57]    [,58]    [,59]    [,60]
## [1,]    3.295    2.954    2.781    2.888    3.043    2.993    2.778
## [2,] 2460.400 2358.222 2273.933 2212.213 2153.781 2056.588 1919.054
##         [,61]    [,62]   [,63]    [,64]    [,65]    [,66]   [,67]   [,68]
## [1,]    2.664    2.706    2.52    2.624    2.426    2.348   2.032   1.639
## [2,] 1758.177 1645.185 1510.12 1384.124 1181.509 1006.455 833.863 639.250
##        [,69]    [,70]    [,71]    [,72]   [,73]  [,74]  [,75] [,76] [,77]
## [1,]   1.054   0.7674   0.6895   0.8453  0.6927 0.6226 0.7186     0     0
## [2,] 398.357 251.8900 163.9173 112.4739 18.4446 5.6349 1.1743     0     0
##      [,78] [,79] [,80] [,81] [,82] [,83] [,84] [,85] [,86] [,87] [,88]
## [1,]     0     0     0     0     0     0     0     0     0     0     0
## [2,]     0     0     0     0     0     0     0     0     0     0     0
##      [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96]
## [1,]     0     0     0     0     0     0     0     0
## [2,]     0     0     0     0     0     0     0     0
```

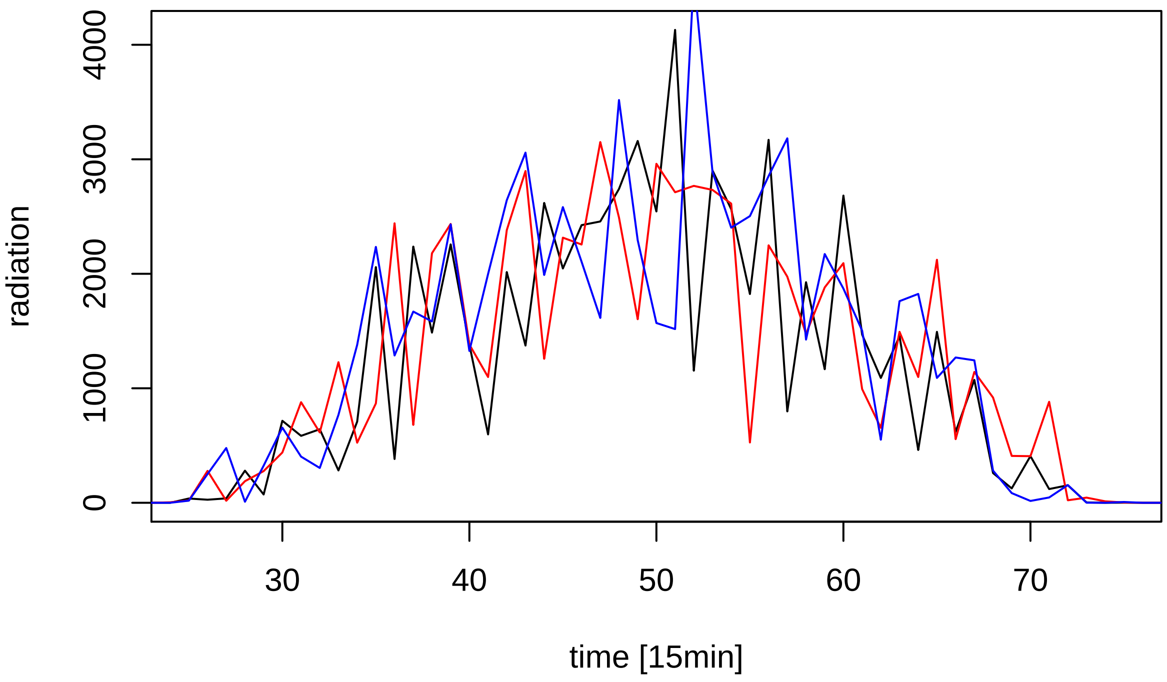
2.2 Sample generation

2.2.1 Sample generation - independent for 15min intervals

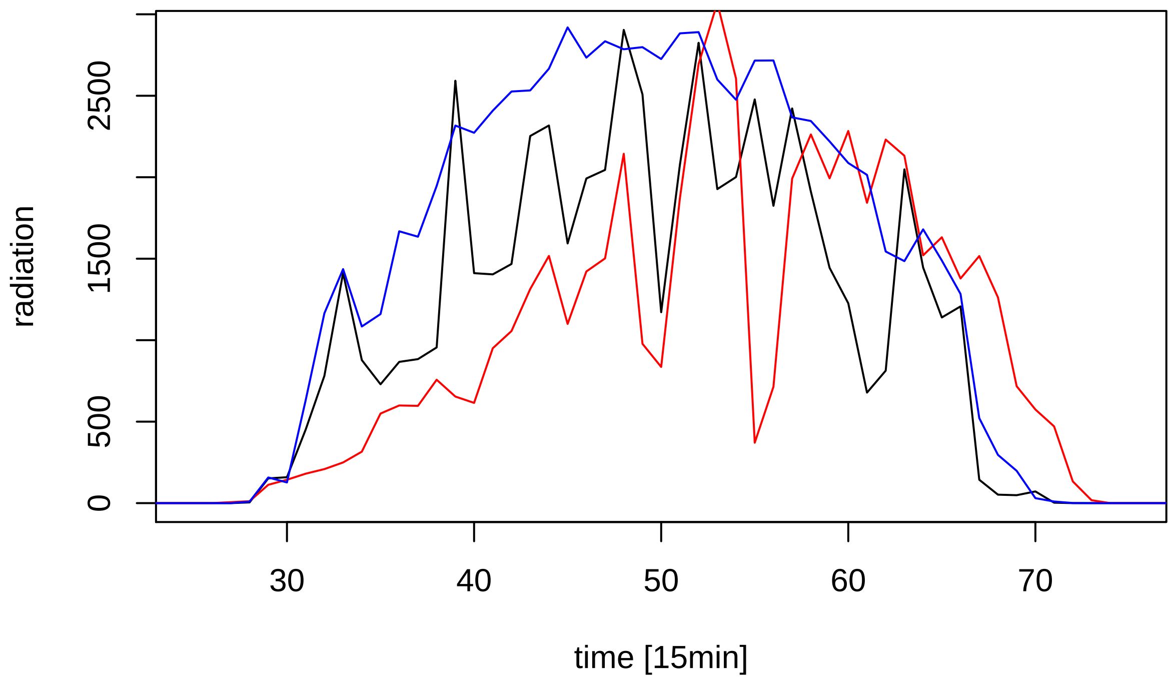
```
N<-10000
W_ind_sample<-matrix(rep(0,24*4*N), ncol=24*4) # 1 column for 1 time interval, 1 row for 1 realization
for(i in 1:N){
W_ind_sample[i,]<-rweibull(24*4,estimates_w_ind[1,], estimates_w_ind[2,])
}
```

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

```
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")
```



```
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]