Vegetation Indices – Monte Carlo

Uncertainty propagation in spatial environmental modelling

2024, Sytze de Bruin & Gerard Heuvelink

WAGENINGEN UR
For quality of life

Ex. 1 - With which *n* SR results are stable?

```
> set.seed(1234567)
> n <- 200       # sample size
> Z <- matrix(rnorm(2*n),2,n) # 2 rows, n columns; independent draws
> #                             from standard normal distribution
> devs <- t(M %*% Z)
> SRsamp <- MC_SR(0.1, 0.6, devs)
> sd(SRsamp)
[1] 2.837269
> mean(SRsamp)
[1] 6.441485
> Z <- matrix(rnorm(2*n),2,n)
> devs <- t(M %*% Z)
> SRsamp <- MC_SR(0.1, 0.6, devs)
> sd(SRsamp)
[1] 1.559619
> mean(SRsamp)
[1] 6.38809
```

2 replications of a MC experiment producing very different results

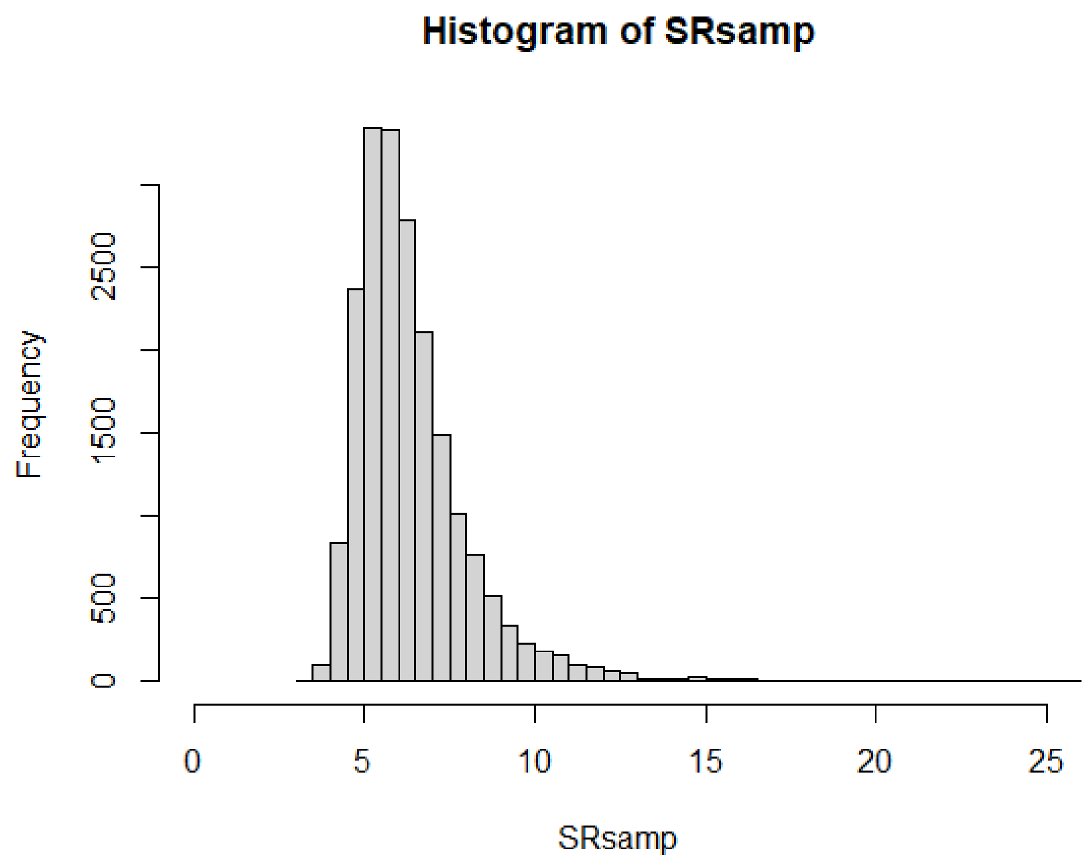
Ex. 1 - Continued

```
> set.seed(1234567)
> n <- 20000     # new sample size
> Z <- matrix(rnorm(2*n),2,n) # 2 rows, n columns; independent draws
> #                                from standard normal distribution
> devs <- t(M %*% Z)
> SRsamp <- MC_SR(0.1, 0.6, devs)
> sd(SRsamp)
[1] 1.97321
> mean(SRsamp)
[1] 6.425931
> Z <- matrix(rnorm(2*n),2,n)
> devs <- t(M %*% Z)
> SRsamp <- MC_SR(0.1, 0.6, devs)
> sd(SRsamp)
[1] 1.931545
> mean(SRsamp)
[1] 6.39086
```

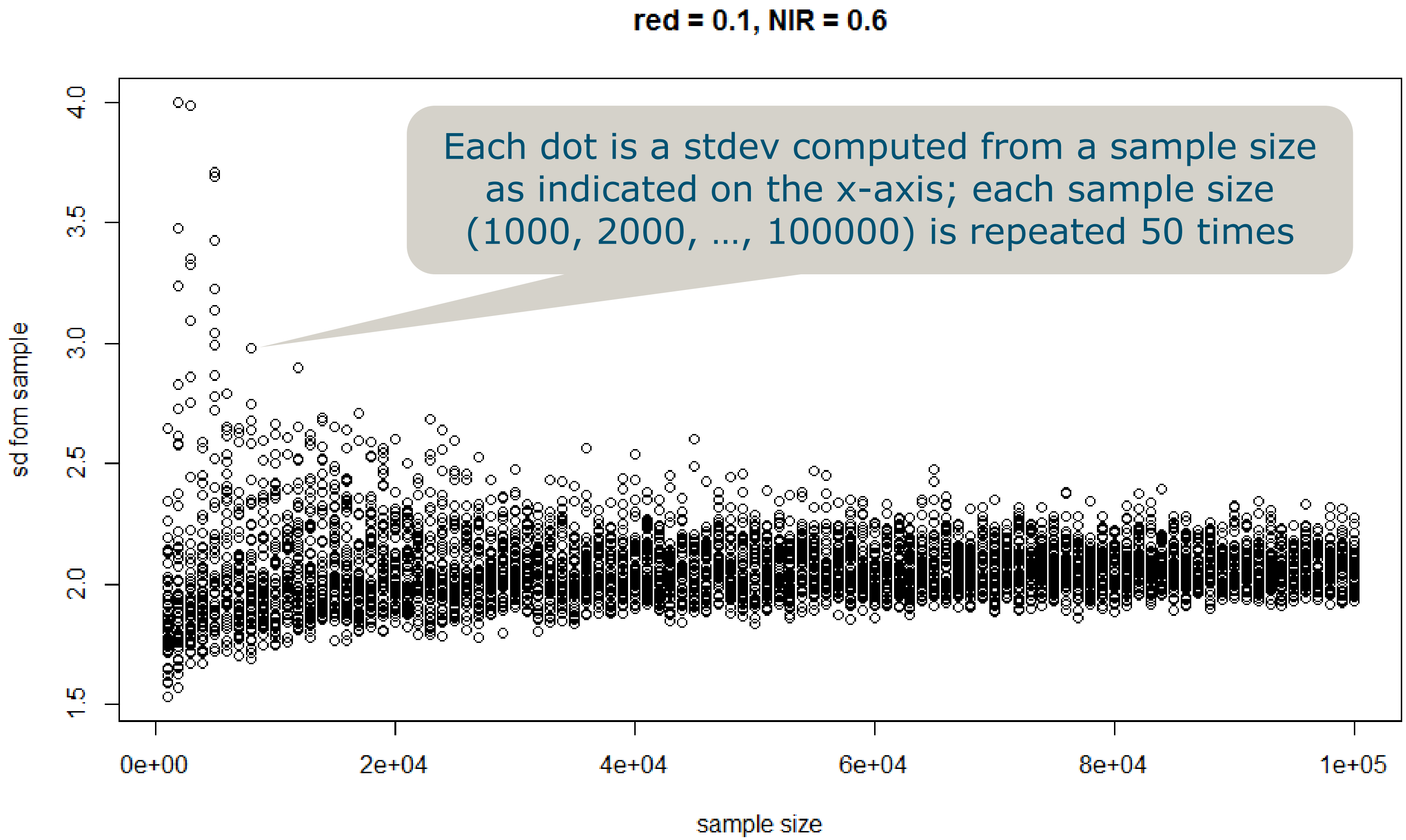
Standard deviation stabilizes

Ex. 1 – distribution of uncertain SR (n=20000)

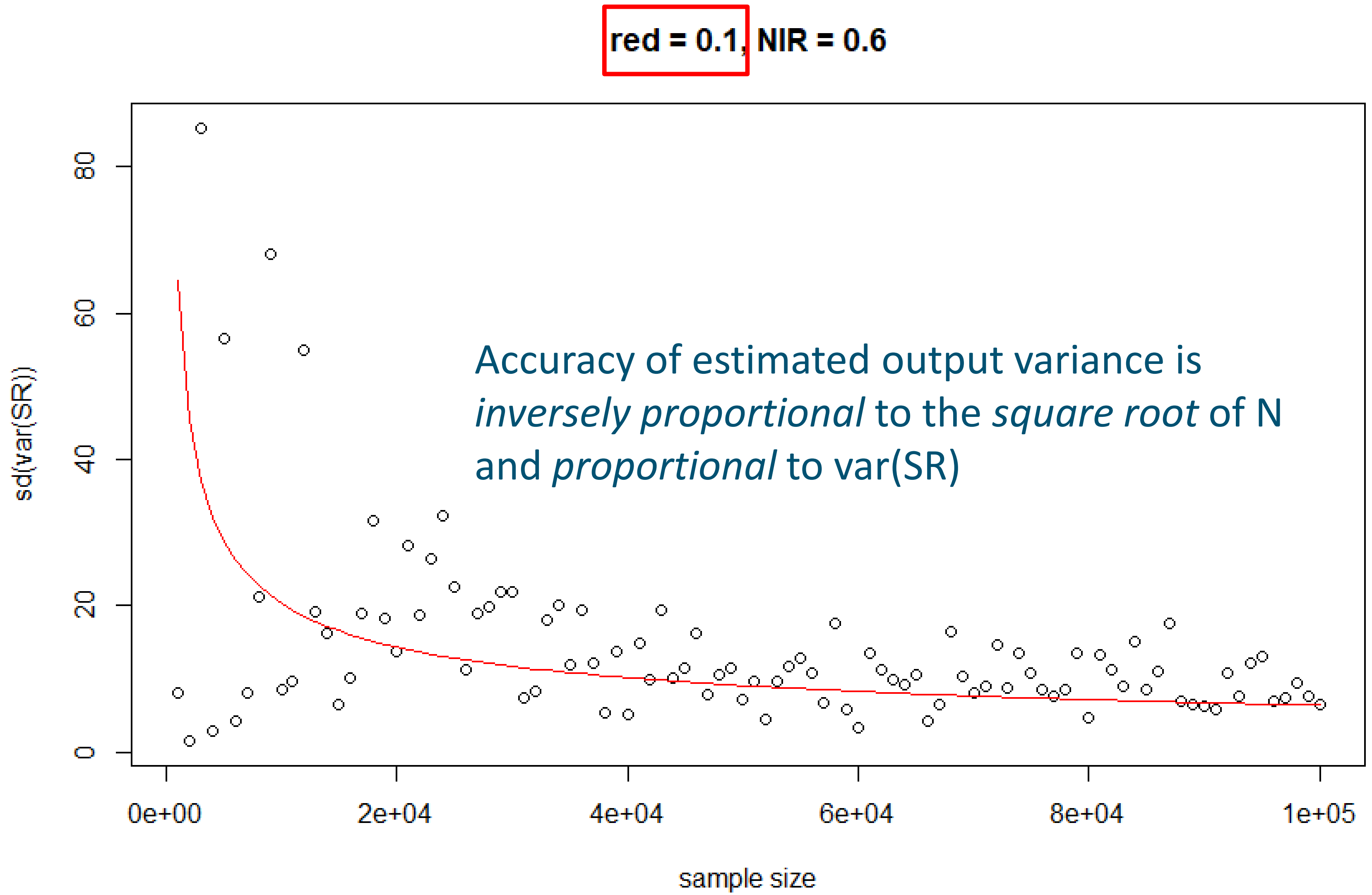
```
hist(SRsamp, breaks = 100, xlim=c(0, 25))
```



Ex. 2 – Plot many stdevs. versus sample size

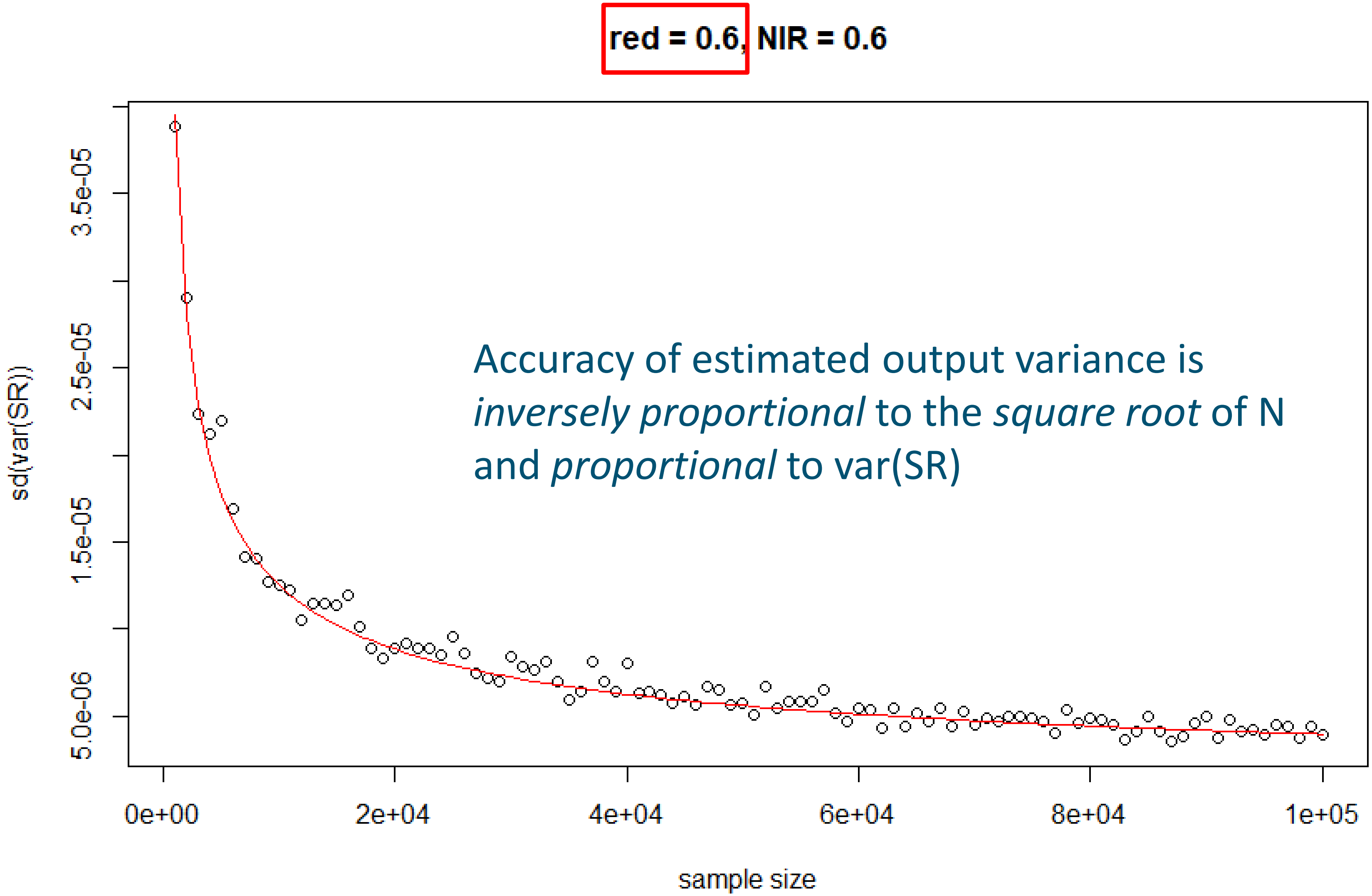


Ex. 2 – sd(var(SR)) versus sample size



WAGENINGEN UR
For quality of life

Ex. 2 – sd(var(SR)) versus sample size

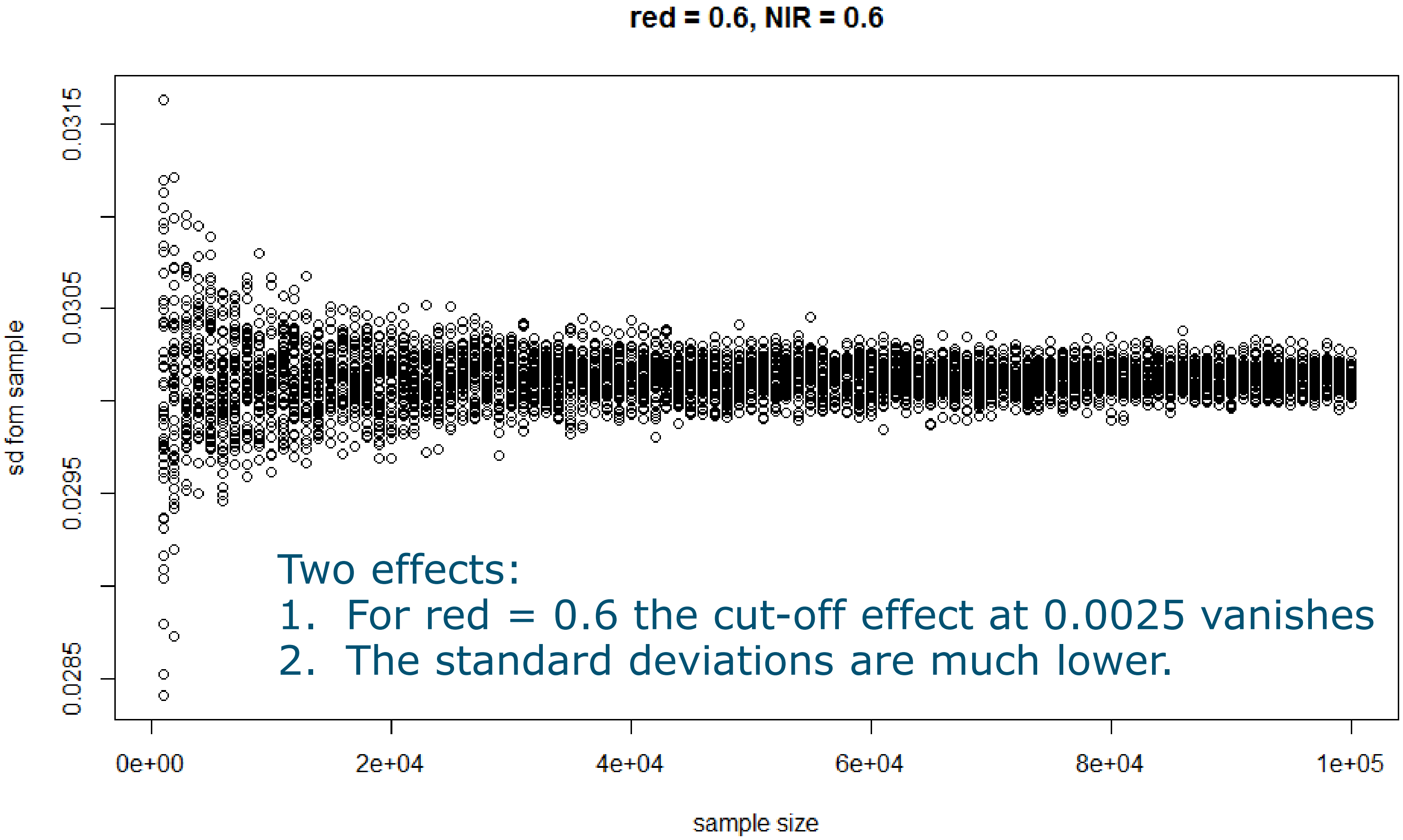


Accuracy of estimated output variance is *inversely proportional* to the *square root* of N and *proportional* to var(SR)

WAGENINGEN UR
For quality of life

Ex. 2 – Plot many stdevs. versus sample size

Is the number of required runs affected when using higher (e.g. 0.6) values for the reflectance in the red band? Explain.



Two effects:

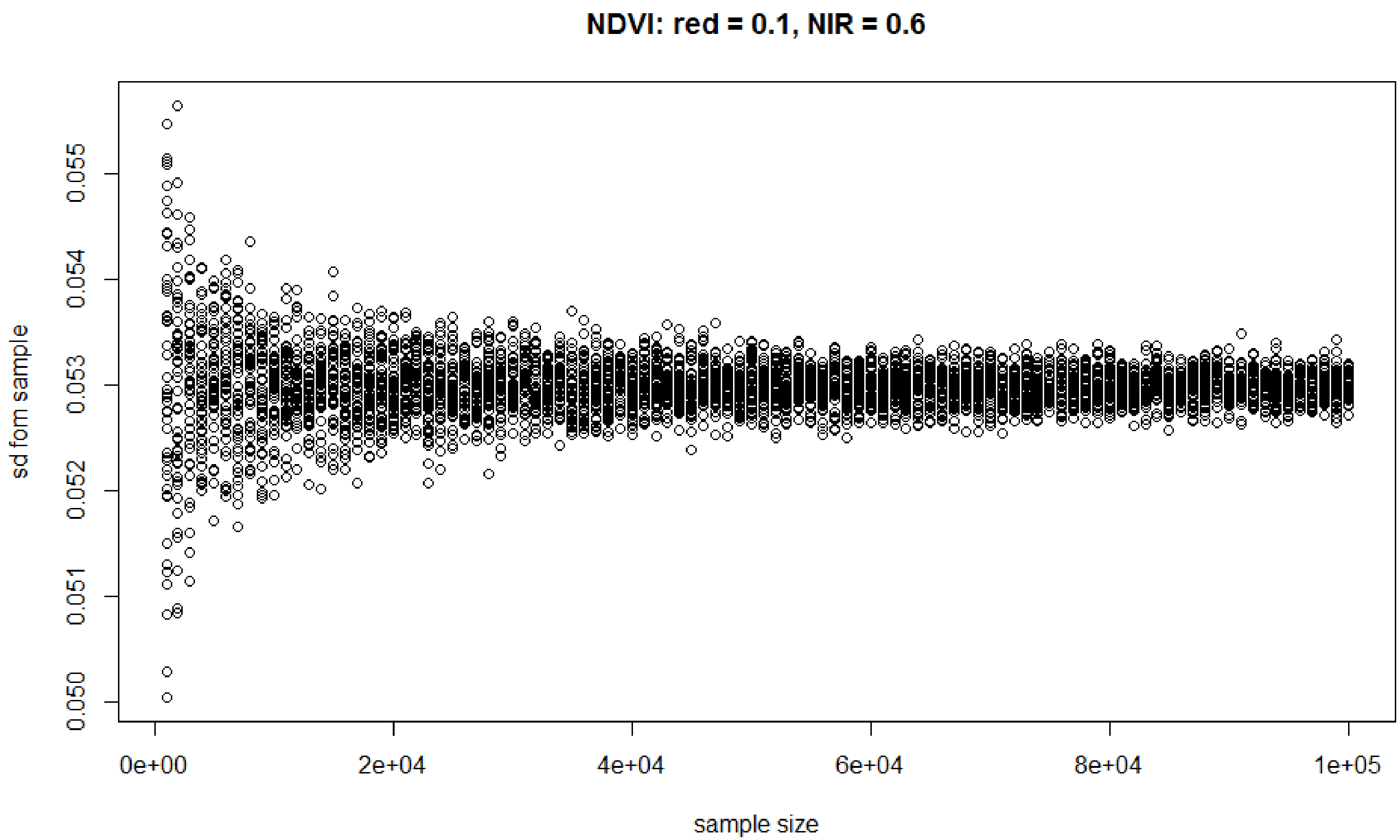
1. For red = 0.6 the cut-off effect at 0.0025 vanishes
2. The standard deviations are much lower.

Exercise 3 - Function for NDVI

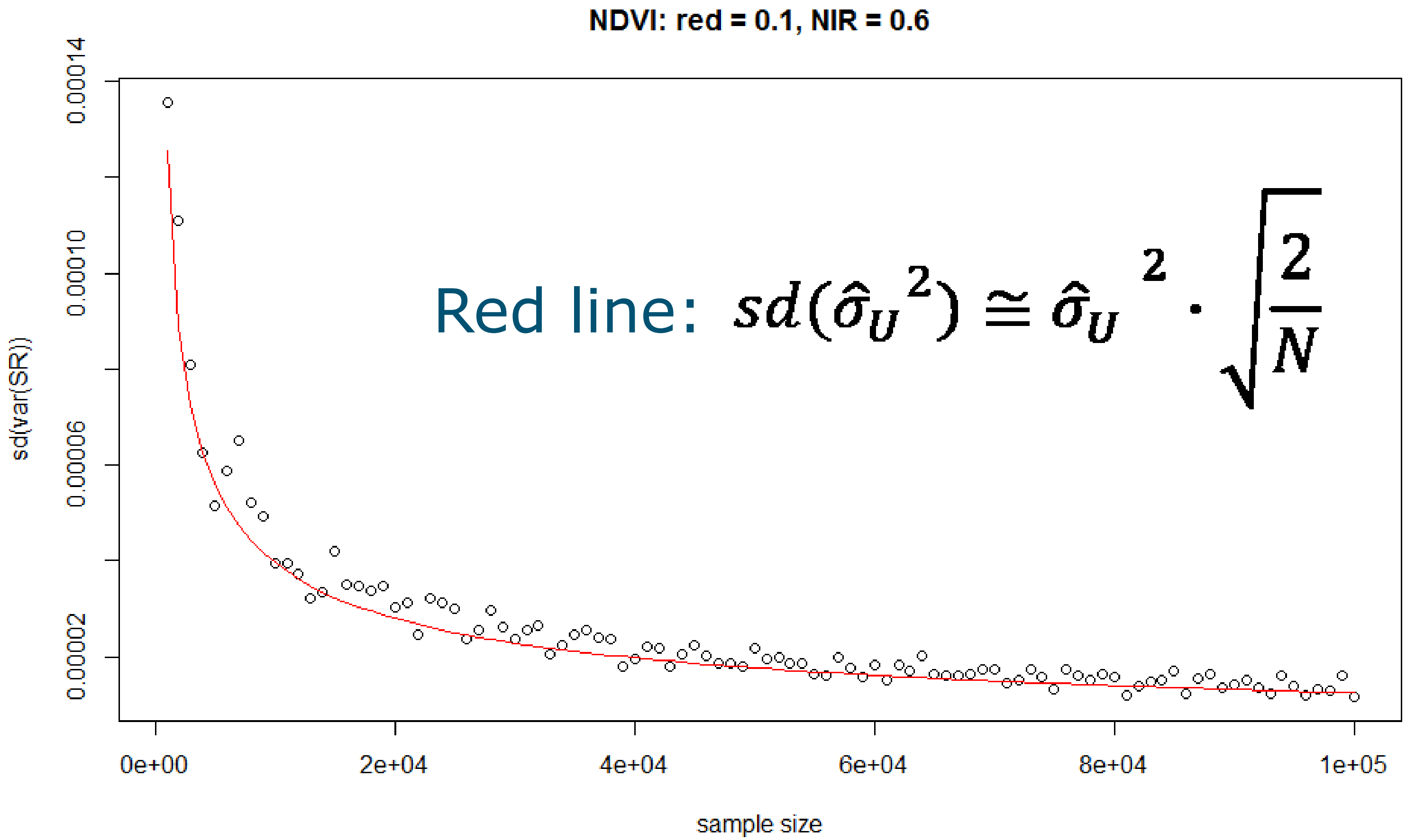
```
MC_sd_NDVI <- function(x) {
  RED <- x[1]+devs[,1]
  NIR <- x[2]+devs[,2]
  Den <- RED + NIR
  # ignore sampled denominators near zero
  Den[abs(Den) < 0.0025] <- NA
  samp <- (NIR-RED)/Den
  return(sd(samp, na.rm=T))
}

MC_sd_NDVI(c(0.1, 0.6))
[1] 0.053
```

WAGENINGEN UR
For quality of life

Ex. 4 – Plot many stdevs. versus sample size



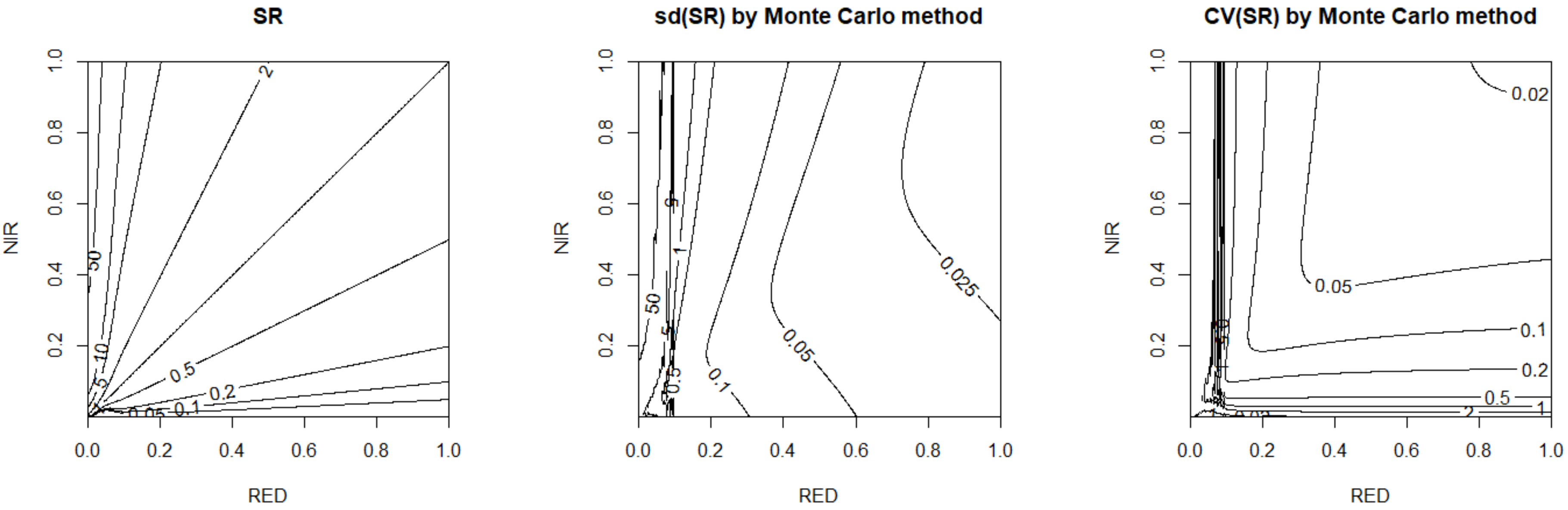
Ex. 4 - sd(var(SR)) versus sample size



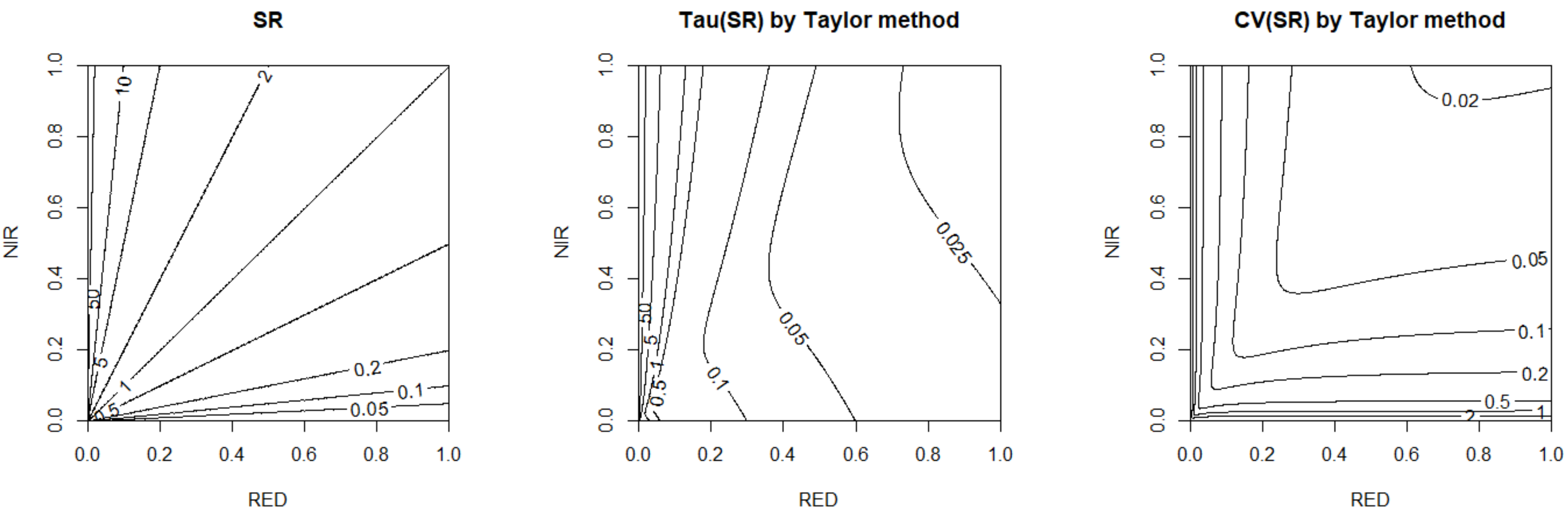
Red line: $sd(\hat{\sigma}_U{}^2) \cong \hat{\sigma}_U{}^2 \cdot \sqrt{\frac{2}{N}}$

For approximately normally distributed variable

Ex. 5 - Contour plot SR – Monte Carlo (n = 20000)

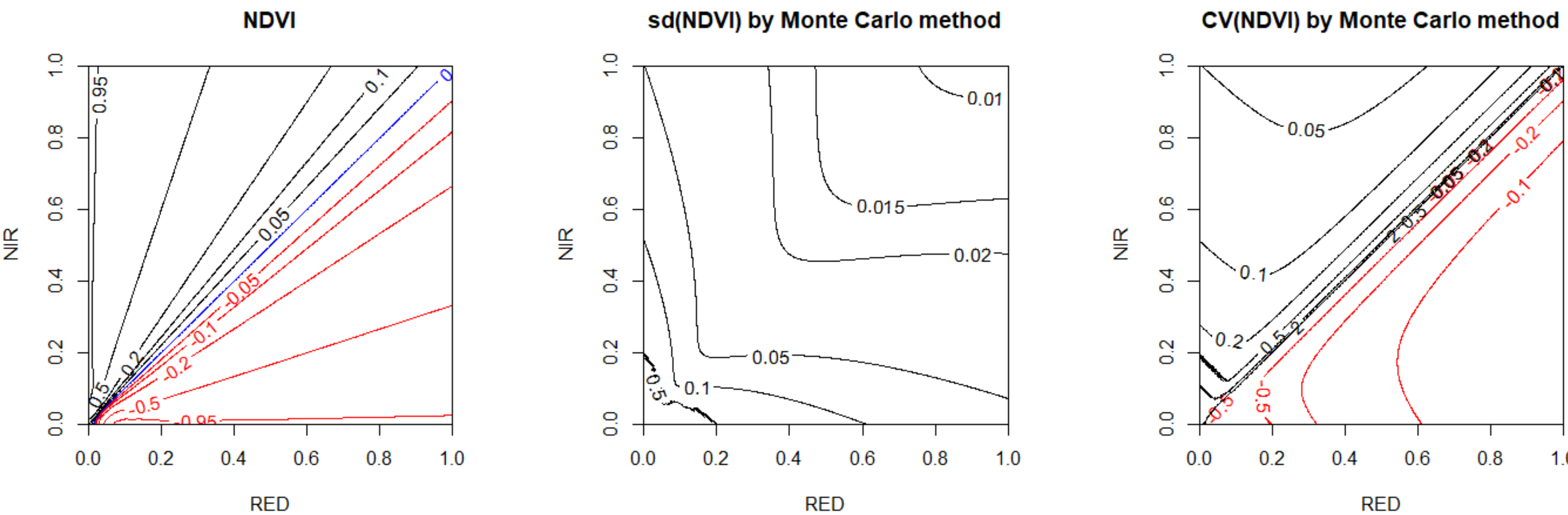


Ex. 5 - Contour plot SR – 1st order Tayor

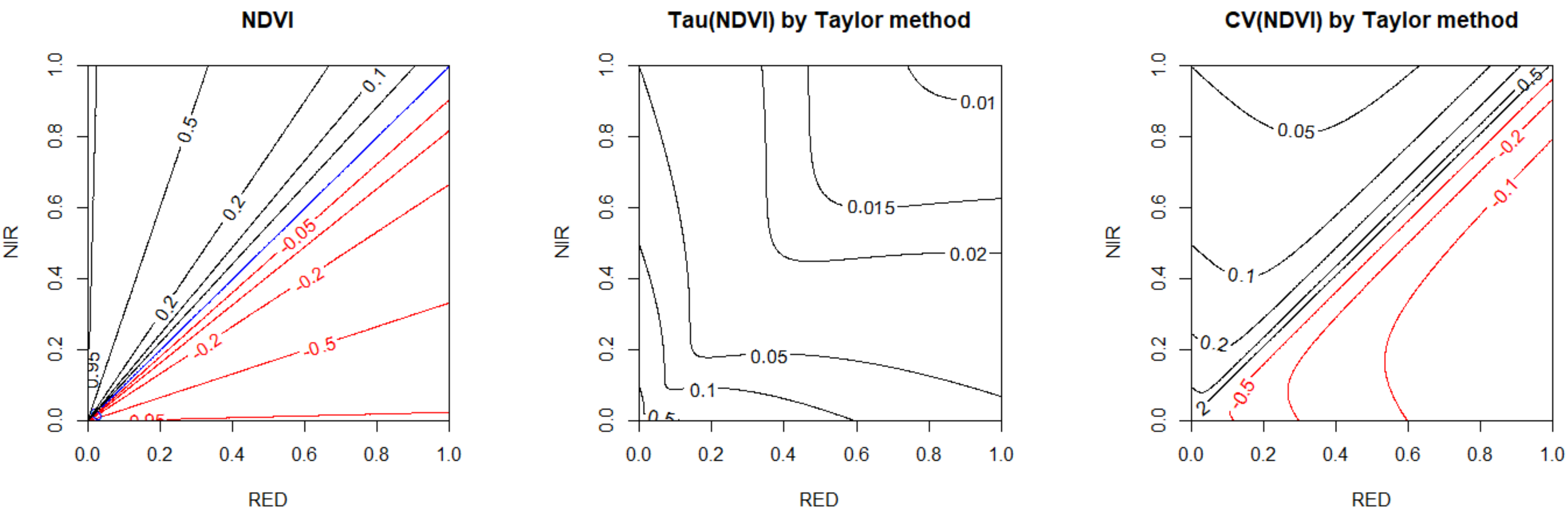


WAGENINGEN UR
For quality of life

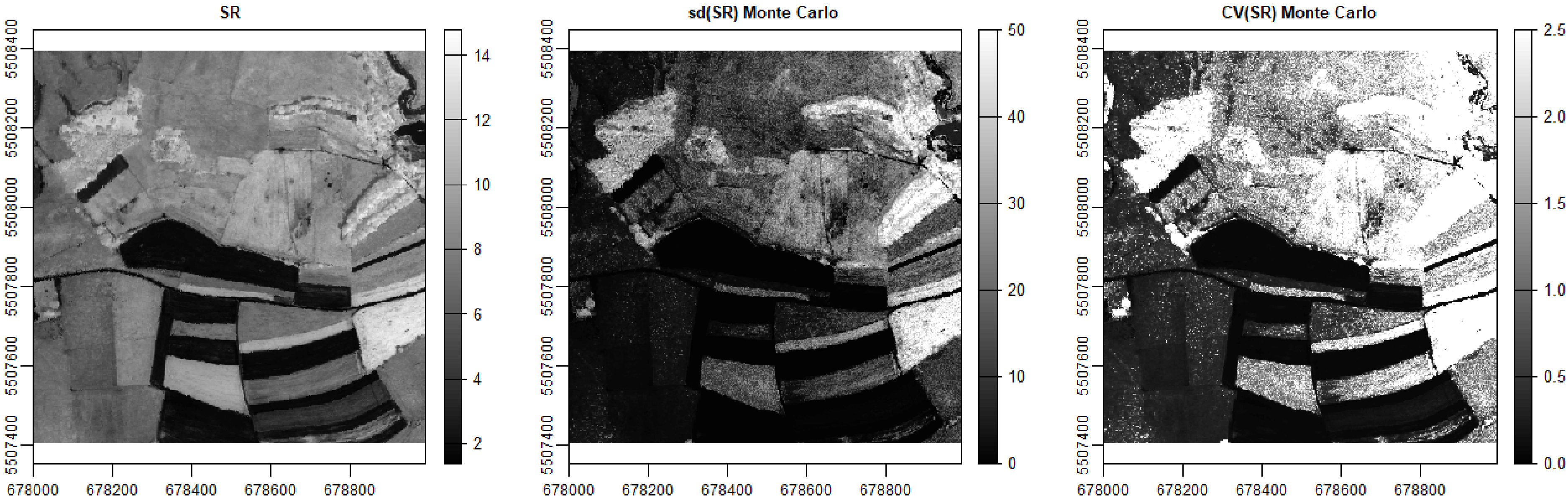
Ex. 5 - Contour plot NDVI – Monte Carlo (n = 20000)



Ex. 5 - Contour plot NDVI – 1st order Tayor

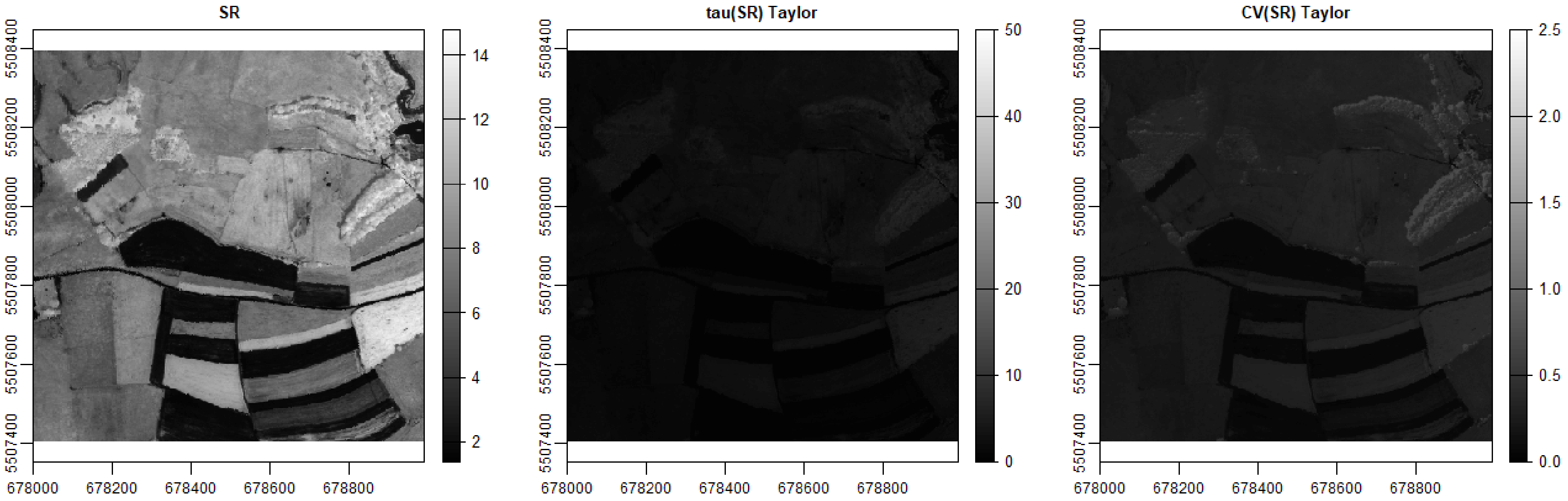


Ex. 6 - SR: Monte Carlo (sd)

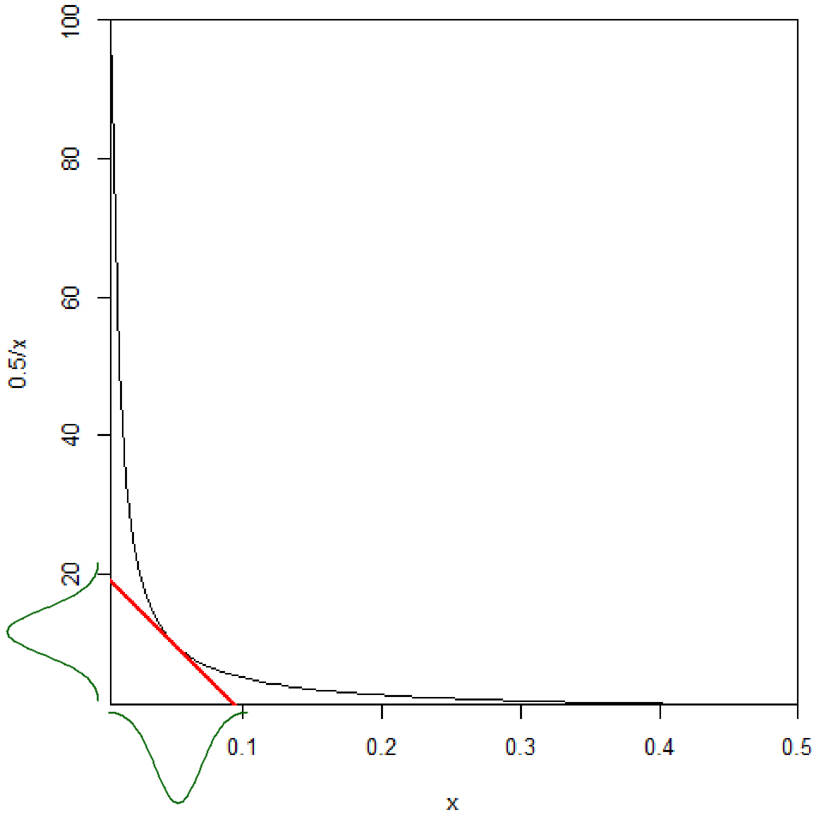


WAGENINGEN UR
For quality of life

Ex. 6 - SR: 1st order Taylor (tau)



Ex. 6 – Partial explanation

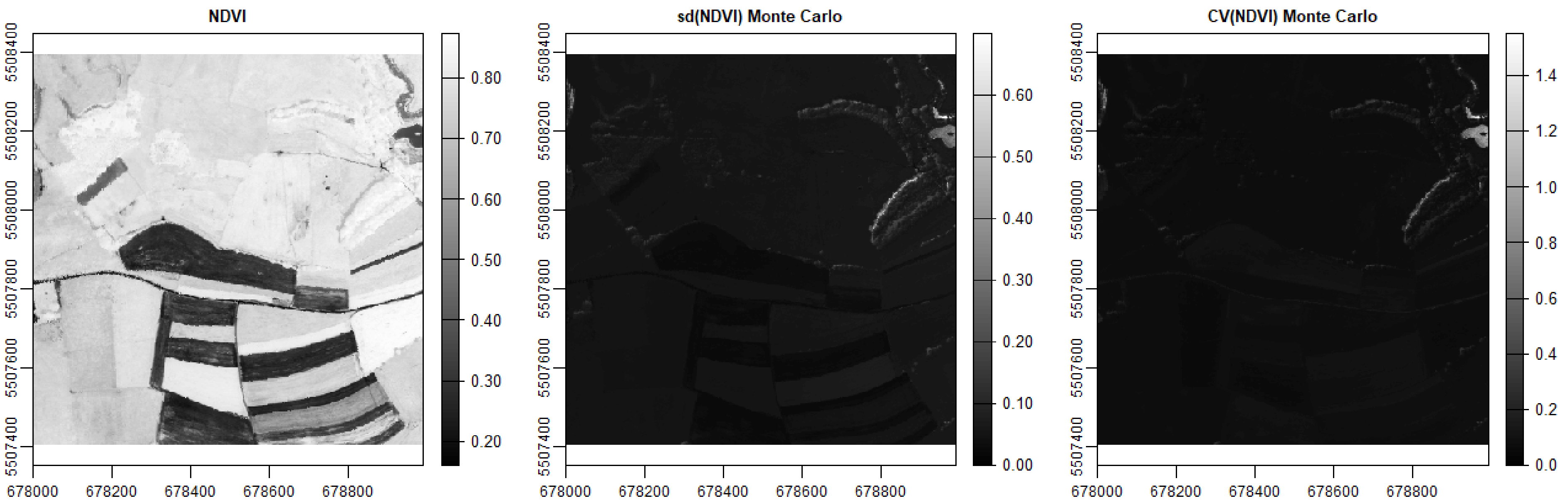


Plot of 0.5/x in the range 0 – 0.5

The larger part of the image is covered by vegetation with low reflectance in the red band (the denominator). Linearization of SR in this region underestimates uncertainty propagation.

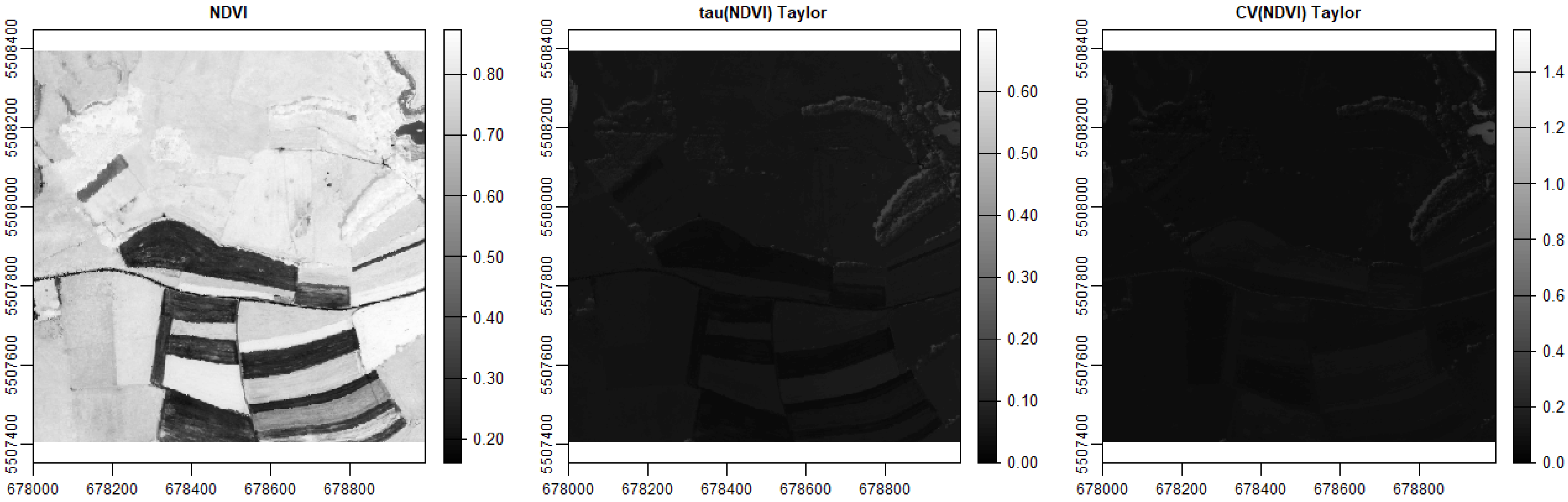
WAGENINGEN UR
For quality of life

Ex. 6 – NDVI: Monte Carlo (sd)



WAGENINGEN UR
For quality of life

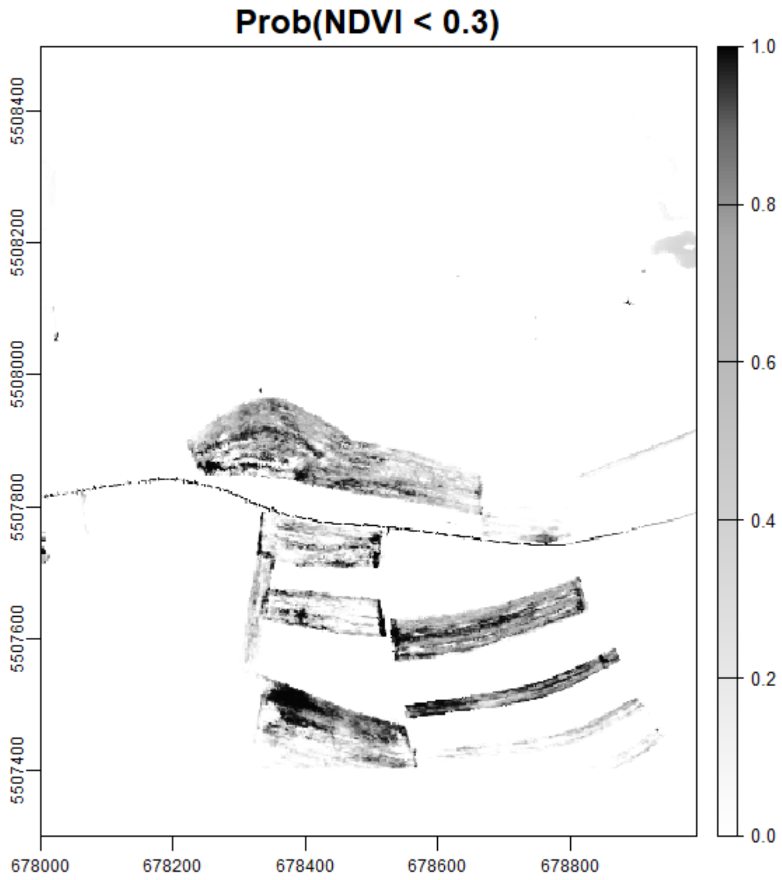
Ex. 6 – NDVI: 1st order Taylor (tau)



WAGENINGEN UR
For quality of life

Ex. 7 – Threshold on probability – Monte Carlo

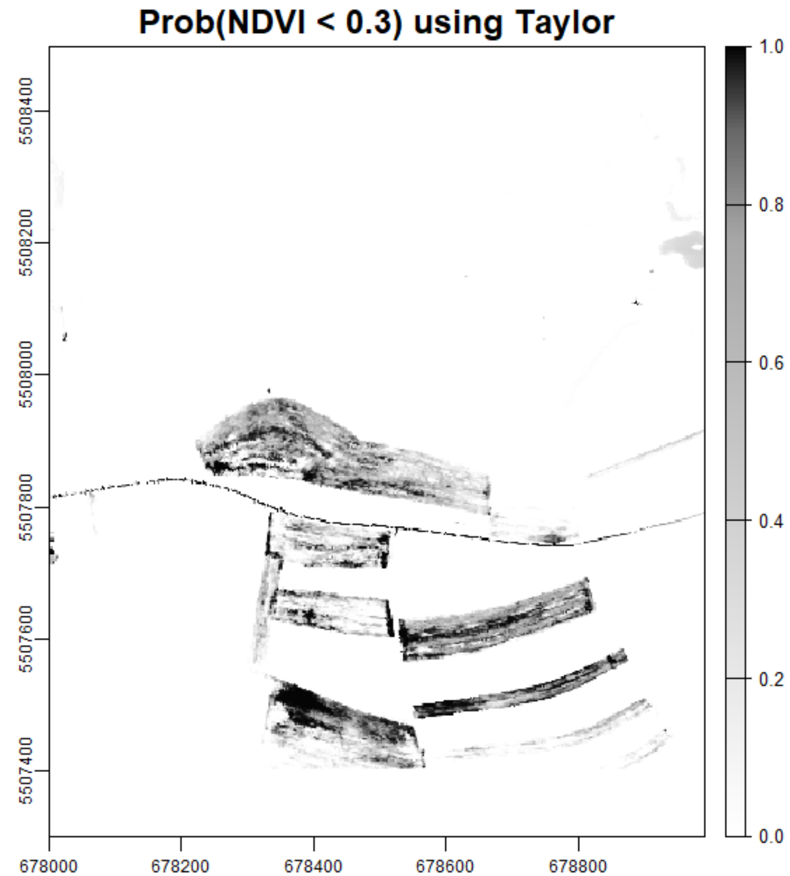
```
prob03MC <- app(false_color, fun = function(x)
  mean(ifelse(MC_NDVI(x[2], x[3], devs) < 0.3,1, 0)),
  filename="prob03MC.tif", overwrite = T)
```



WAGENINGEN UR
For quality of life

6) Taylor: assume normal distribution

```
NDVIim <- rast("NDVI.tif")
tau_NDVI <- rast("tau_NDVI.tif")
TaylorProb <- app(c(NDVIim, tau_NDVI),
                  fun = function(x) pnorm(0.3, x[1], x[2]))
```



But NDVI is no linear function of the inputs

Take home

- Monte Carlo works for any model, including black box models
- Produces sample of the *output distribution* (not just variance)
- The “approximation error” can be reduced by increasing the sample size
- It is computationally demanding; particularly for complex models
- It can be applied on spatial data; so far we did not consider spatial correlation (recall discussion @morning lecture)
- There are smarter sampling methods than random sampling (e.g., stratified random sampling, Latin hypercube sampling)