Exercise

Parameter estimation with R

A regression model for disease forecasting

The exercise

Parameter estimation with R

- Two functions for non linear models
  - nls() is specifically adapted to minimization of sums of squares and regression analysis.
  - optim() is a general purpose minimizer (minimize sum of squared errors)
  - If nls() works, that gives more information. If not use optim()
    - They require different functions to evaluate model. Sorry.

Example
Grain filling of wheat

- The model for grain filling

$$y^{\mathrm{mod}} = \frac{W}{1+\exp(B-(c)(DD))}$$

- y is grain weight (mg)
- DD is degree days from anthesis (the explanatory variable)
- W, B, c are parameters, to be estimated

R

```
rm(list=ls())
seedwt<-function(DD,W,B,c)
{
        weight<-W/(1+exp(B-c*DD))
        return(weight)
}
```

My function to calculate seed weight as a function of explanatory variables and parameters

```
wt<-c(8,13,17,21,26,27,28,27,27,29,28)
DD<-c(195,240,280,330,380,410,480,530,580,620,680)
```

The data. Observed values and explanatory variables

```
ols<-nls(y~seedwt(time,W,B,c),
        data=list(y=wt,time=DD),
        start=list(W=30,B=4,c=0.019))
summary(ols)
```

Call to nls. data has observed values and explanatory variables. start has parameters.

Result of nls

Some rules

- The function must return a vector of simulated values, in the same order as the observed values
- The arguments in the call must match the arguments in the function
  - First argument in call becomes first argument in function etc.
  - (Other possibility – use names)
- The start argument of nls defines parameters and gives starting values
- The data argument of nls gives values of everything else (observed data and explanatory variables)

Displaying and analyzing the results

```
summary(ols)

Formula: y ~ seedwt(time, W, B, c)

Parameters:
  Estimate Std. Error t value Pr(>|t|)
W 28.171255   0.408284   69.00 2.17e-12 ***
B  4.160223   0.360084   11.55 2.86e-06 ***
c  0.016492   0.001415   11.65 2.68e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8364 on 8 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 1.961e-06
```

```
# to retrieve coefficients for further calculation
coef(ols)
         W           B           c
28.17125533  4.16022349  0.01649195
```

```
# fitted() and residuals() use OLS parameters
cbind(wt,fitted(ols),residuals(ols))
      wt
 [1,]  8  7.888909  0.1110908
 [2,] 13 12.666715  0.3332847
 [3,] 17 17.252820 -0.2528196
 [4,] 21 22.052755 -1.0527547
 [5,] 26 25.116207  0.8837928
 [6,] 27 26.226215  0.7737854
 [7,] 28 27.527683  0.4723169
 [8,] 27 27.885439 -0.8854391
 [9,] 27 28.045232 -1.0452322
[10,] 29 28.105958  0.8940420
[11,] 28 28.146945 -0.1469453
```

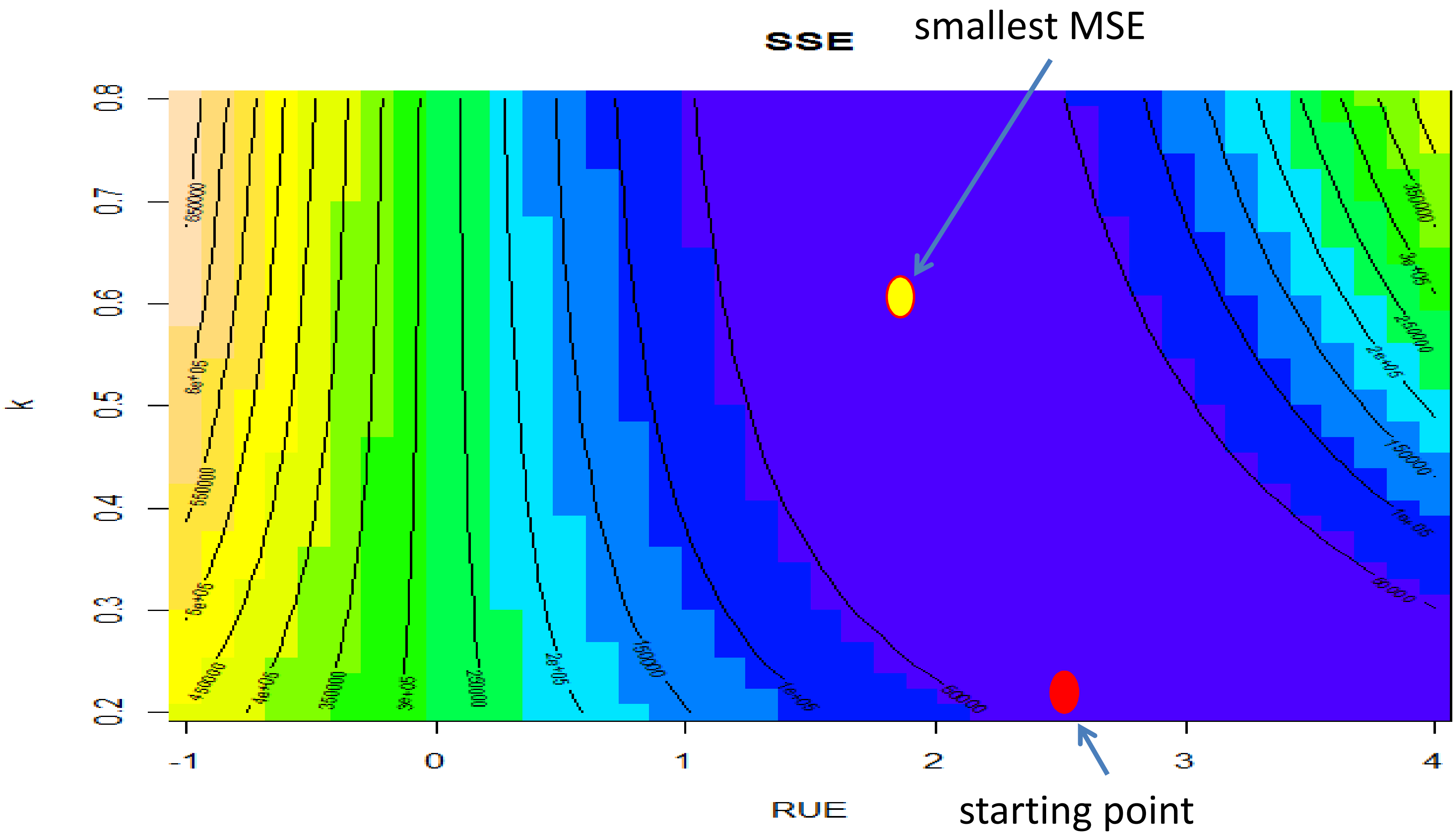
```
# predict() gives new predictions using OLS parameters
predict(ols,newdata=list(time=195:240))
 [1]  7.888909  7.982918  8.077601  8.172953  8.268967
8.365637  8.462956  8.560919  8.659517
[10]  8.758743  8.858590  8.959050  9.060115  9.161777
9.264026  9.366854  9.470251  9.574209
[19]  9.678718  9.783767  9.889346  9.995446 10.102055
10.209162 10.316756 10.424826 10.533361
[28] 10.642348 10.751775 10.861630 10.971900
11.082573 11.193636 11.305076 11.416879 11.529032
[37] 11.641521 11.754333 11.867452 11.980866
12.094560 12.208519 12.322729 12.437175 12.551842
[46] 12.666715
```

OLS in pictures

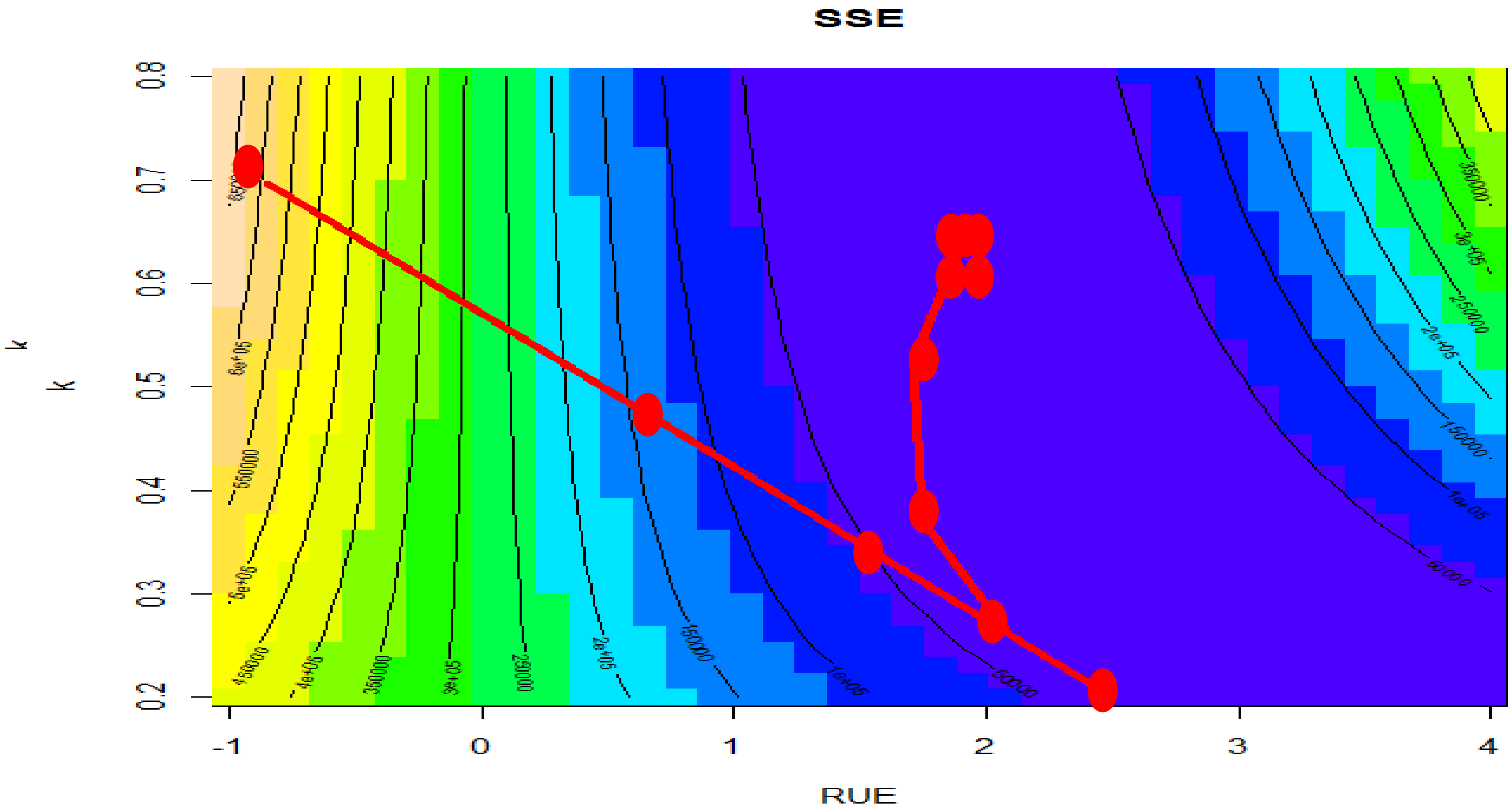
Gradient method

- Gauss-Newton. Default for nls()
  - Linearize model. (Need to calculate derivatives)
  - Calculate parameters that minimize MSE of linear problem (analytical)
  - Iterate (necessary because of non-linearity)

Path of Gauss Newton?

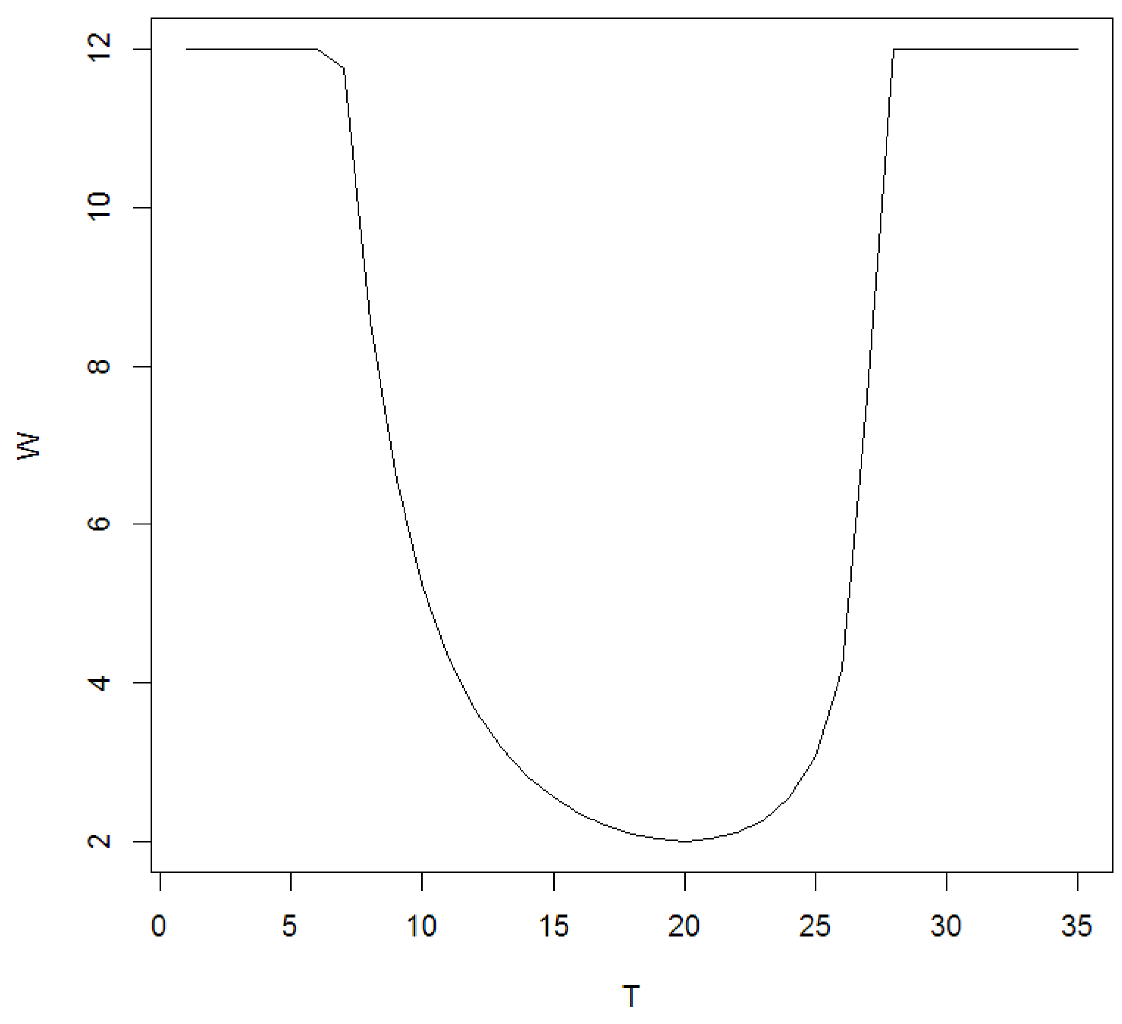


Path of Gauss Newton



Model of leaf wetness requirement

- Infection models of fungal foliar pathogens: often success of infection depends on temperature T and duration of leaf wetness W.
- $W=f(T;\theta)$ Each pathogen has requirement of W for infection to occur, depends on T
- Get parameters θ by fitting past data
  - Experiments: multiple T, W combinations
  - Each T, identify minimum W such that there is 20% disease incidence or 5% disease severity on an infected plant part at nonlimiting inoculum concentration.



12
10
8
6
4
2
W
0
5
10
15
20
25
30
35
T

Exercise

- Calibrate a model of leaf wetness requirement for fungal foliar pathogens

- Model is

$$W = Wmin / ft$$

$$fT = \left[(Tmax - T) / (Tmax - Topt)\right]\left[(T - Tmin) / (Topt - Tmin)\right]^{(Topt - T\,\mathrm{min})/(Tmax - Topt)}$$

$$W = Wmax \ \text{ if } \ W > Wmax$$

$$W = Wmax \ \text{ if } \ T < Tmin \text{ or } T > Tmax$$

- 5 parameters: Tmin, Topt, Tmax, Wmin, Wmax

- Data for Pseudoperonospora cubensis (downy mildew on cucumber)

T (°C)	W (hours)
4	12.0
8	8.6
12	3.7
16	2.3
20	2.0
24	2.5
28	12.0
32	12.0

- The disease might have parameters similar to Bremia lactucae on lettuce
  - Tmin=1, Topt=15, Tmax=25, Wmin=4, Wmax=10

To do

1. Write a function that calculates W as a function of T given Tmin, Topt, Tmax, Wmin, Wmax
2. Use the parameters of Bremia lactucae to calculate W as a function of temperature. Draw the graph.
3. Add the points for Pseudoperonospora cubensis to the graph
4. Use the data for Pseudoperonospora cubensis to calibrate the model for that disease. Start from Bremia lactucae parameters
5. Examine the output of nls. Are all the parameters well estimated?
6. Use the calibrated model to predict W for T<-0:40
7. Graph the model results for T from 0 to 40°C (type="l")
8. Add the data points. How well does the model fit the data?
9. Plot residuals (observed- simulated) versus simulated values.
10. Is this a "correct model"
11. Are the residuals "homoscedastic"?