Assignment 3

Mats Slik & Dennis Haandrikman

25-05-2022

Assignment 1: assess model validity

You have to compare the data from the experiment with the simulation data, to assess the model validity. The best way to do this is plotting the model results and the experimental results in one graph. The model results have one value per time unit, the experimental results have multiple values per time unit. Either plot all data points of the experimental data using a scatterplot or calculate the median results for each time unit and plot a line of these median points. You can do this for instance with the function aggregate.

```
# read the data file (if the wrapper doesn't work: put this as file: "./Data/MPL.csv"
data <- read.csv(file = data_file, na.strings =  "NA",)
# Extract the median for the low dose
median_MPL_01 <- median(data$MPL_conc[data$dose==0.1], na.rm=T)
median_MPL_01
```

```
## [1] 14.59
```

```
# Extract the median for the high dose
median_MPL_03 <- median(data$MPL_conc[data$dose==0.3], na.rm=T)
median_MPL_03
```

```
## [1] 39.925
```

```
# Generate the medians for the MPL concentration, mRNA & Free receptor
medians <- aggregate(data[,c("MPL_conc","mRNA","Free_receptor")],list(data$dose,data$time), median, na.
names(medians)[1:2] <- c("dose","time")
head(medians)
```

```
##   dose time MPL_conc   mRNA Free_receptor
## 1  0.0    0    0.000 3.7900        292.95
## 2  0.1    6   11.180 1.7025        124.70
## 3  0.3    6   31.295 1.7295         97.90
## 4  0.1   10   12.335 1.7515        157.80
## 5  0.3   10   36.960 1.4140         69.55
## 6  0.1   13   11.945 1.7045        152.50
```

```
# Subset the datasets so the low/high dose are seperated
data_low_dose <- subset(data, dose != 0.3)
data_high_dose <- subset(data, dose != 0.1)
```

```r
median_low_dose <- subset(medians, dose != 0.3)
median_high_dose <- subset(medians, dose != 0.1)
head(median_low_dose)
```

```
##    dose time MPL_conc   mRNA Free_receptor
## 1   0.0    0    0.000 3.7900        292.95
## 2   0.1    6   11.180 1.7025        124.70
## 4   0.1   10   12.335 1.7515        157.80
## 6   0.1   13   11.945 1.7045        152.50
## 8   0.1   18   29.270 2.6270         97.55
## 10  0.1   24   13.080 1.2055         74.25
```

```r
head(median_high_dose)
```

```
##    dose time MPL_conc   mRNA Free_receptor
## 1   0.0    0    0.000 3.7900        292.95
## 3   0.3    6   31.295 1.7295         97.90
## 5   0.3   10   36.960 1.4140         69.55
## 7   0.3   13   31.970 1.6905         34.30
## 9   0.3   18   46.300 1.5380         26.25
## 11  0.3   24   51.975 0.9330         31.85
```

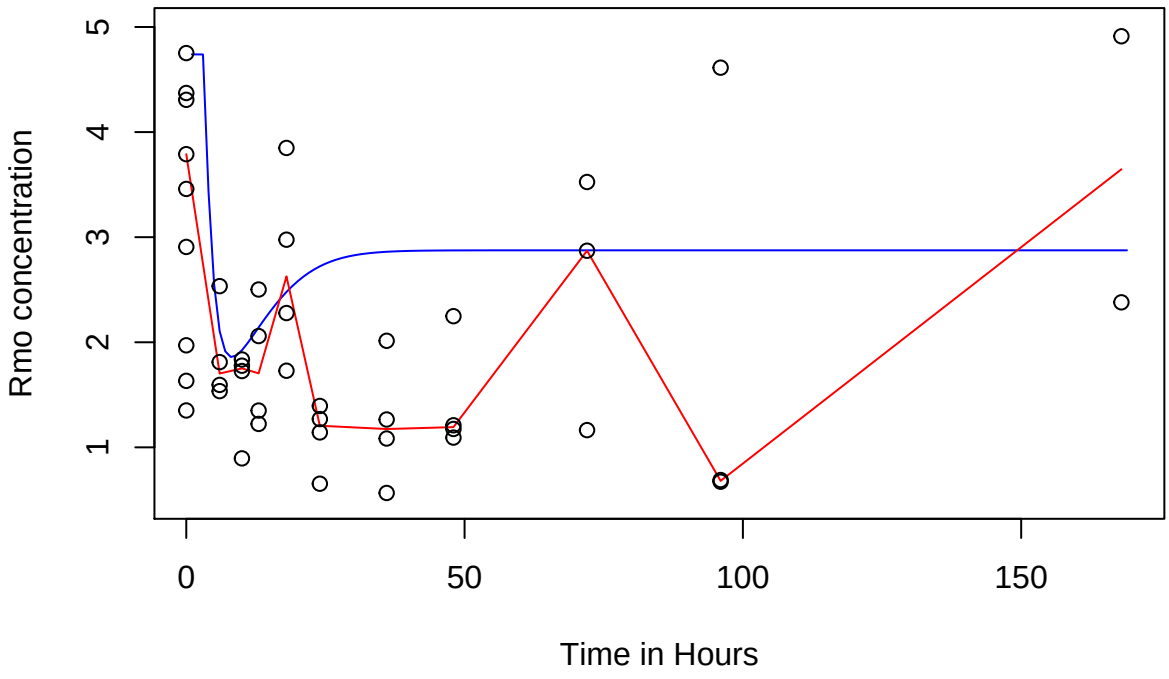
Questions:

[1] Why is it best practice to plot the median for the experimental data? Explain in your report

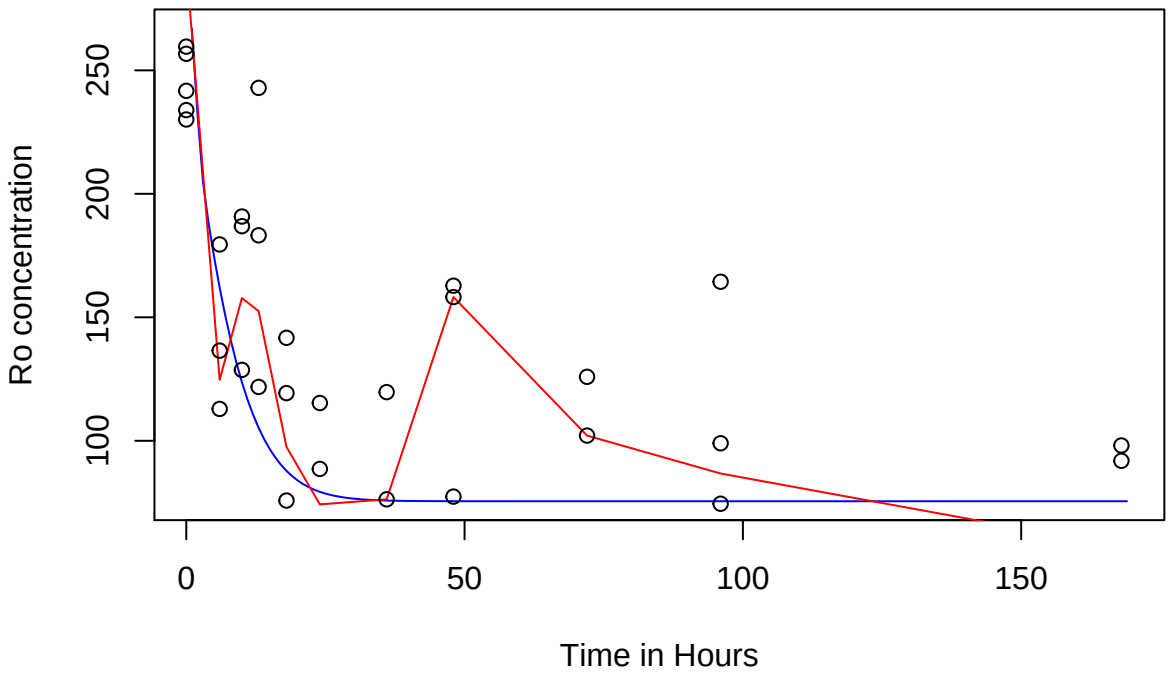
The reason why it's best to plot with the median is that it gives a representation of the central value.

```r
# Set the parameters, state and times for the low dose ode model
D <- (median_MPL_01 * 1000)/374.471
parameters <- c(Ks_rm = 2.90, ic50_rm = 26.2, Kon = 0.00329, Kt = 0.63, Kre = 0.57, Rf = 0.49, Kd_R = 0
state <- c(Rmo = 4.74, Ro = 267, DR = 0, DRN = 0)
times <- seq(0, 168,  by = 1)

# Define the model function for the ode model
Grd_model <- function(t, y, parms){
  with(as.list(c(parms, y)),{
    Rmo.delta <- Ks_rm * (1-(DRN / (ic50_rm + DRN)))- Kd_Rm * Rmo
    R.delta <- Ks_r *  Rmo + Rf * Kre * DRN - Kon * D * Ro - Kd_R * Ro
    DR.delta <- Kon * D * Ro - Kt * DR
    DRN.delta <- Kt * DR - Kre * DRN

        return(list(c(Rmo.delta, R.delta, DR.delta, DRN.delta)))
       })
}

# Run the ode model and save the output to the out variable
out <- ode(times = times, y = state,   parms = parameters, func = Grd_model, method = "euler")

# Plot the graph for mRNA with the model data
plot(as.data.frame(out)$Rmo, type = "l", xlab = "Time in Hours", ylab = "Rmo concentration", ylim =  c(
# Plot the median of the experimental data-set as a line and the full experimental data-set as points
```

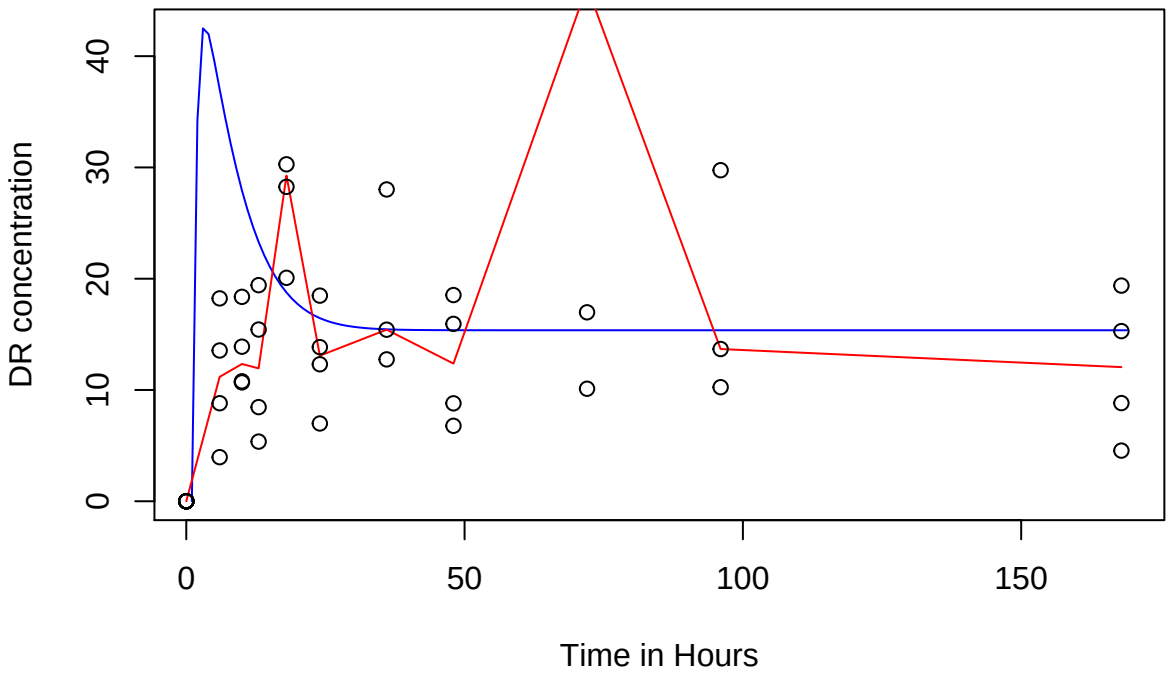
```
lines(mRNA~time, data = median_low_dose, col = "red")
points(mRNA~time, data = data_low_dose)
```



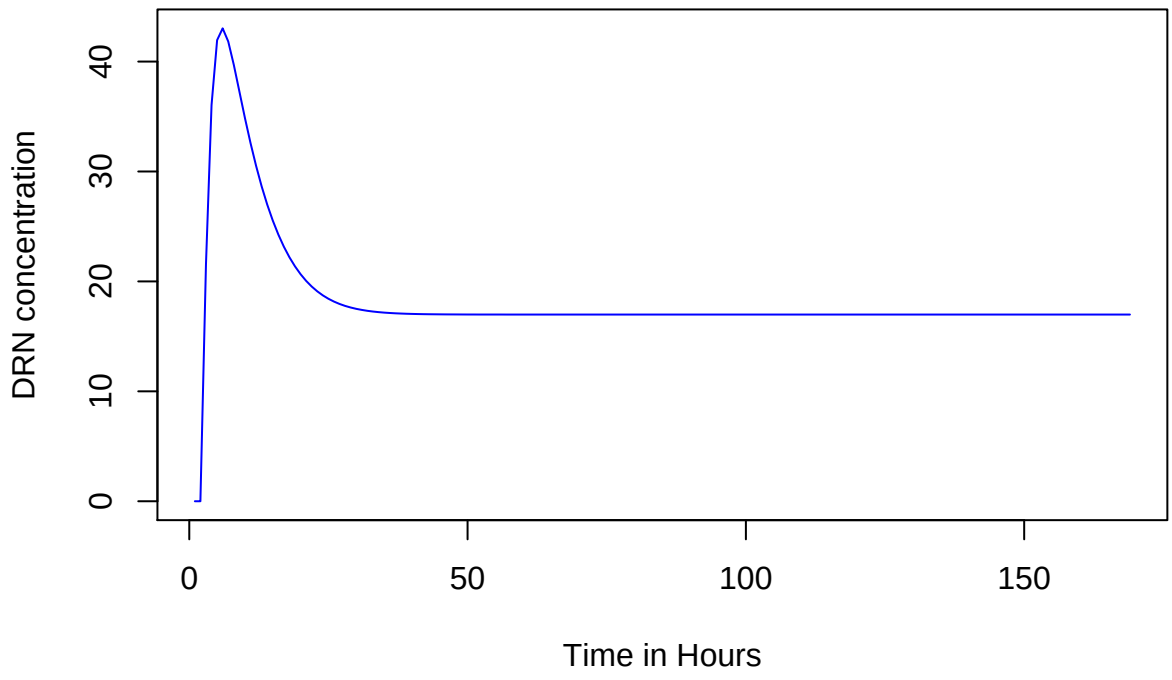

```
# Plot the graph for free receptor with the model data
plot(as.data.frame(out)$Ro, type = "l", xlab = "Time in Hours", ylab = "Ro concentration", col = "blue")
# Plot the median of the experimental data-set as a line and the full experimental data-set as points
lines(Free_receptor~time, data = median_low_dose, col = "red")
points(Free_receptor~time, data = data_low_dose)
```



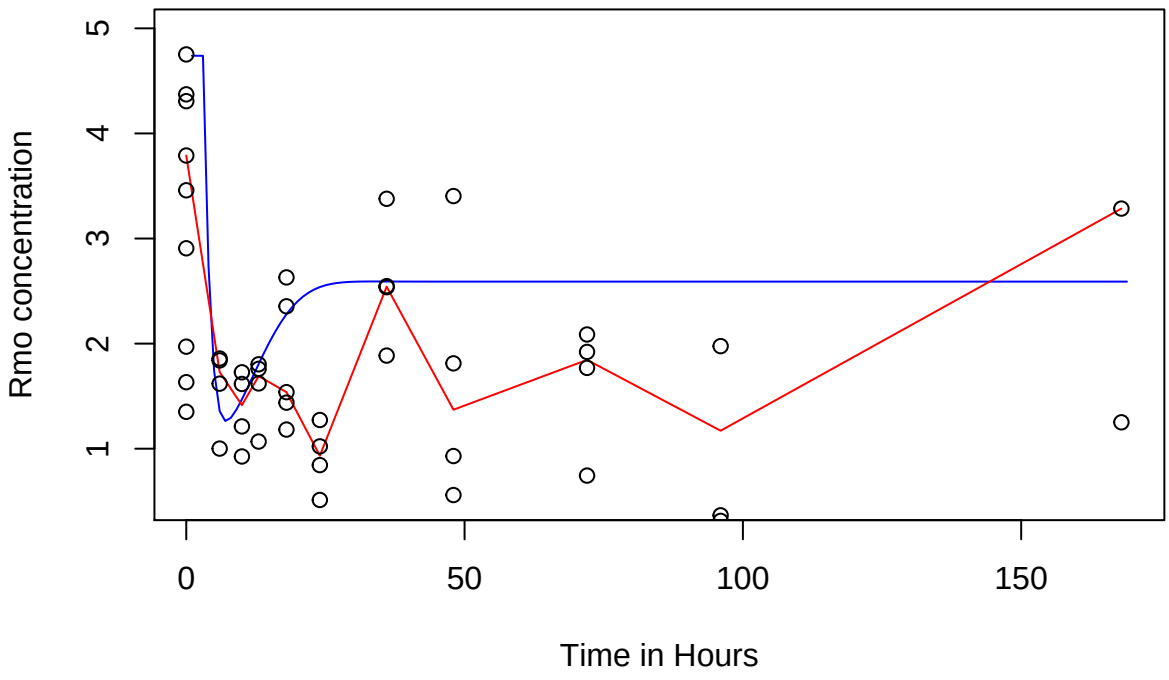

```
# Plot the graph for MPL_complex with the model data
plot(as.data.frame(out)$DR, type = "l", xlab = "Time in Hours", ylab = "DR concentration", col = "blue")
# Plot the median of the experimental data-set as a line and the full experimental data-set as points
lines(MPL_conc~time, data = median_low_dose, col = "red")
points(MPL_conc~time, data = data_low_dose)
```



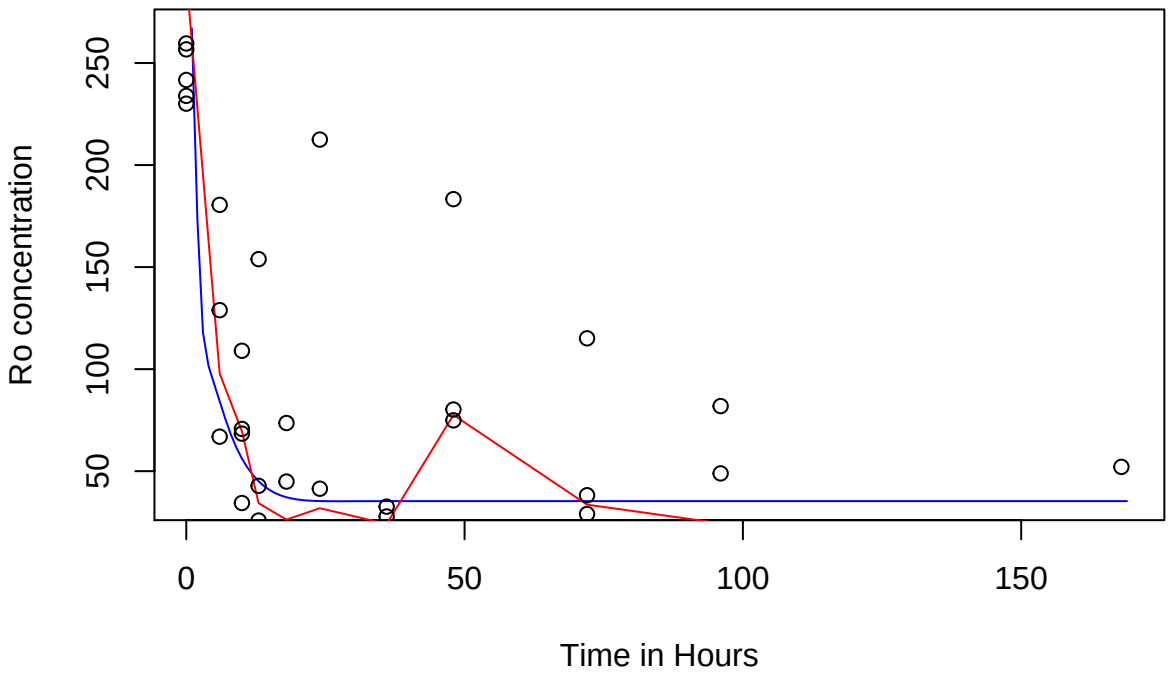

```
# Plot the graph for MPL_complex in nucleus with the model data
plot(as.data.frame(out)$DRN, type = "l", xlab = "Time in Hours", ylab = "DRN concentration", col = "blu
```



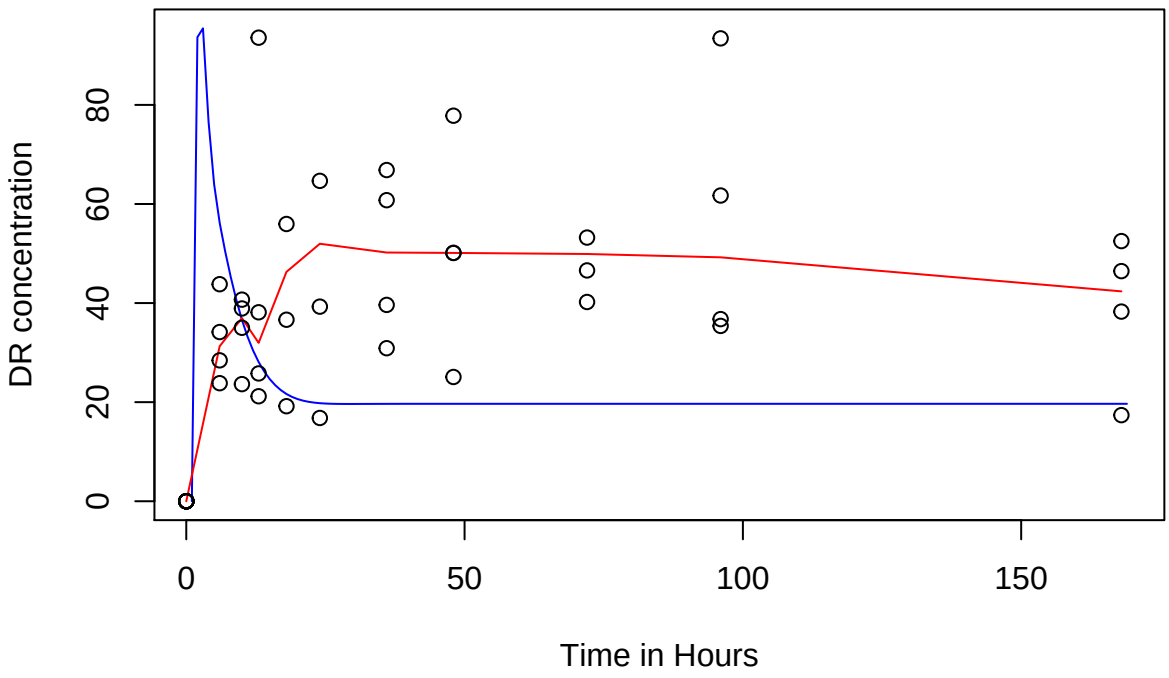

```
# Set the parameters for the high-dose ode model
D <- (median_MPL_03 * 1000)/374.471
parameters <- c(Ks_rm = 2.90, ic50_rm = 26.2, Kon = 0.00329, Kt = 0.63, Kre = 0.57, Rf = 0.49, Kd_R = 0

# Run the ode model and save the output to the out variable
out <- ode(times = times, y = state,   parms = parameters, func = Grd_model, method = "euler")

# Plot the graph for mRNA with the model data
plot(as.data.frame(out)$Rmo, type = "l", xlab = "Time in Hours", ylab = "Rmo concentration", ylim =  c(0
# Plot the median of the experimental data-set as a line and the full experimental data-set as points
lines(mRNA~time, data = median_high_dose, col = "red")
points(mRNA~time, data = data_high_dose)
```



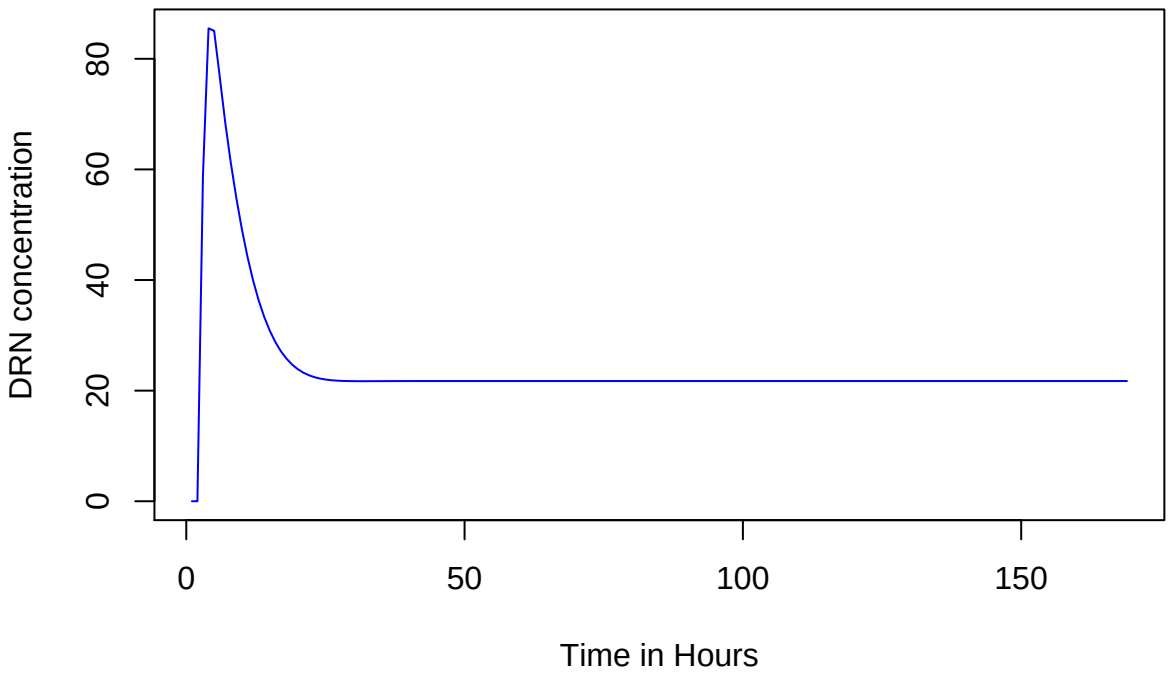

```
# Plot the graph for free receptor with the model data
plot(as.data.frame(out)$Ro, type = "l", xlab = "Time in Hours", ylab = "Ro concentration", col = "blue")
# Plot the median of the experimental data-set as a line and the full experimental data-set as points
lines(Free_receptor~time, data = median_high_dose, col = "red")
points(Free_receptor~time, data = data_high_dose)
```




```
# Plot the graph for MPL_complex with the model data

plot(as.data.frame(out)$DR, type = "l", xlab = "Time in Hours", ylab = "DR concentration", col = "blue")
# Plot the median of the experimental data-set as a line and the full experimental data-set as points
lines(MPL_conc~time, data = median_high_dose, col = "red")
points(MPL_conc~time, data = data_high_dose)
```




```
# Plot the graph for MPL_complex in nucleus with the model data
plot(as.data.frame(out)$DRN, type = "l", xlab = "Time in Hours", ylab = "DRN concentration", col = "blu
```



[2] How do the results of the simulations depend on the dose and concentration of the drug? Compare the model variables mRNA, R with the experimental data by running the simulations adjusting dosis D and plot these to find the answer.

With both doses, we can see that the mRNA concentration follows the same trend as the model, it decreases and then increases again. However, it's notable that with at both dose levels the mRNA doesn't stabilise like the model, and it performs a yoyo-effect before the mRNA stabilises in an upwards climb.

[3] Are the results of the model in line with experimental data? If not, what could be the reason? Think of at least one explanation. Try to test it with simulations (you will get bonus points for that, your explanation does not need to be correct, but should be logical).

Depending on what data is compared, it seems that both low and high dose don't properly follow the model to a point. The low dose data seemingly deviating more than the high-dose data. That is quite possible due to a significant amount more factors playing into the regulation of the receptors, mRNA & complexes. This is also possible due to the cells not being able to perfectly achieve a balance as the model, causing the rollercoaster effect of ups and downs.

Assignment 2: simulate scenario's

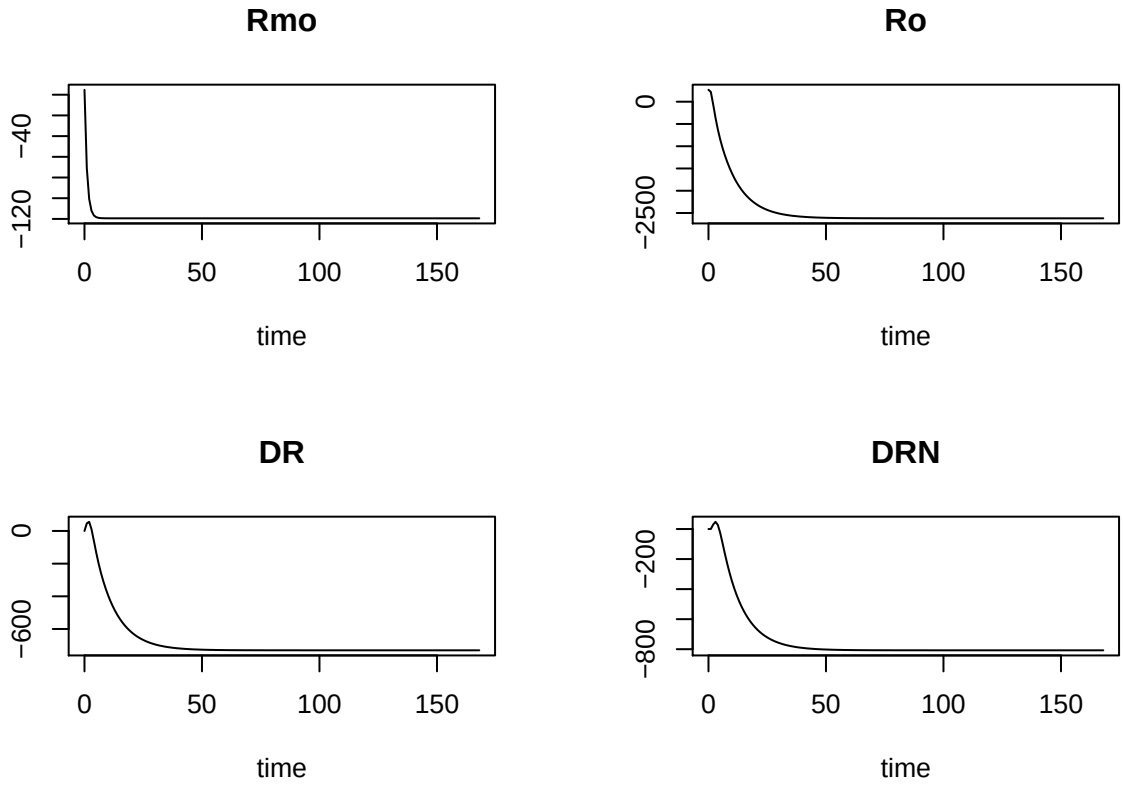
You are going to check what happens if you change the model and model parameters and compare the results to the basic scenario (from week 2). Unless stated otherwise, in the simulations for the solutions to this part, the values of different parameters and initial conditions should be as in the table in the assignments document. Value of kd_Rm should be 0.612, value of ks_r=3.22, value of D=20*1000/374.471.

Questions:

[1] What would be the time course concentration of the activated drug-receptor complex if there was no auto-regulation of glucocorticoid receptor, i.e. if there was not effect of drug on the synthesis of the receptor mRNA? What formula needs to be changed? Adjust the model, run the simulation and plot the results to find out.

Autoregulation means that the cell itself does not regulate the amount of the glucocoritcoid recepter. In our belief it is then that the influence of 'activated MPL complexes in the nucleus' (DRN) should not impede the production of the mRNA, as this would be a negative feedback loop and cause the mRNA to reduce the more DRN is active. To see what would happen to the concentration if there was no auto-regulation of the glucocorticoid receptor, we have rewritten the Rmo.delte formula from: Ks_rm * (1 - (DRN / (ic50_rm + DRN))) - Kd_Rm * Rmo To: Ks_rm * (1-ic50_rm) - Kd_Rm * Rmo

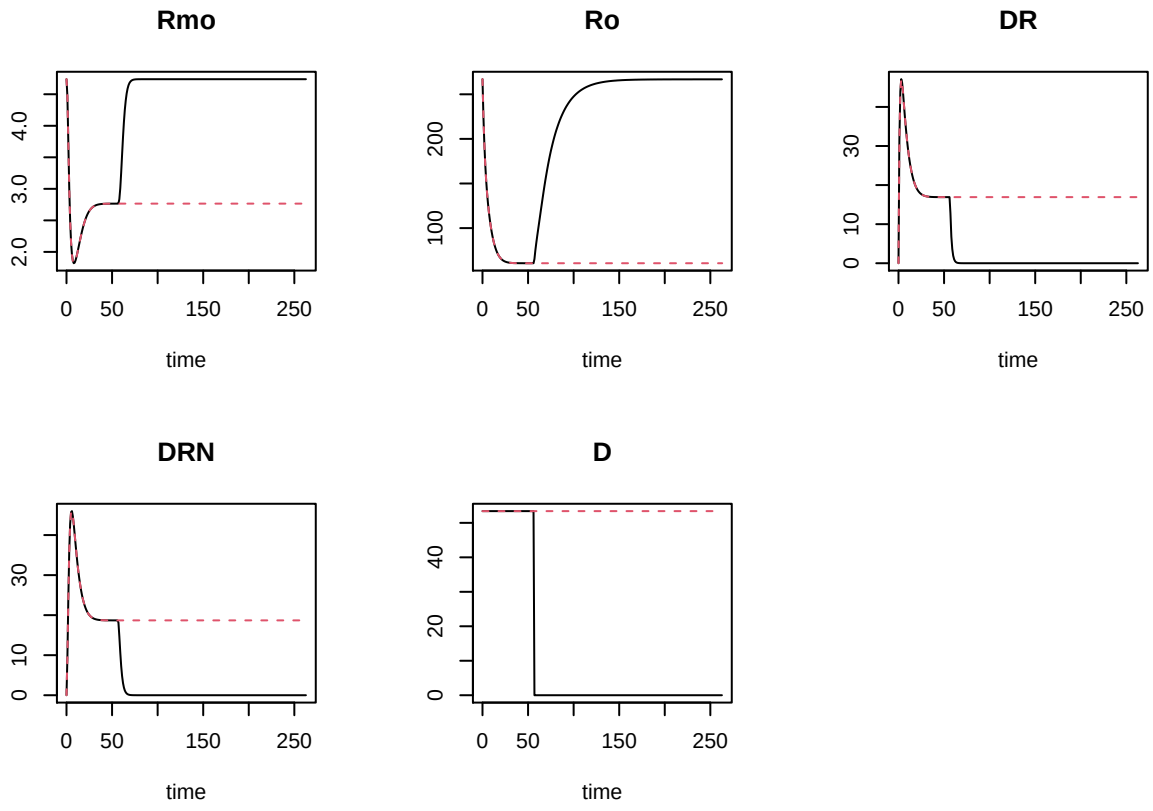
```
# Set the parameters, state and times for the ode-model
D <- (20 * 1000)/374.471
parameters <- c(Ks_rm = 2.90, ic50_rm = 26.2, Kon = 0.00329, Kt = 0.63, Kre = 0.57, Rf = 0.49, Kd_R = 0
state <- c(Rmo = 4.74, Ro = 267, DR = 0, DRN = 0)
times <- seq(0, 168,  by = 1)

# Define the model function for the ode model
Grd_model <- function(t, y, parms){
  with(as.list(c(parms, y)),{
    Rmo.delta <- Ks_rm * (1- ic50_rm)- Kd_Rm * Rmo
    R.delta <- Ks_r *  Rmo + Rf * Kre * DRN - Kon * D * Ro - Kd_R * Ro
    DR.delta <- Kon * D * Ro - Kt * DR
    DRN.delta <- Kt * DR - Kre * DRN

        return(list(c(Rmo.delta, R.delta, DR.delta, DRN.delta)))
       })
}

# Run the ode model and save the output to the out variable
out <- ode(times = times, y = state,   parms = parameters, func = Grd_model, method = "euler")

# plot the graph with the results from the model
plot(out)
```



[2] What is the time course of receptor and mRNA concentrations when the drug treatment is stopped? So After the steady state is reached (at time t_steady), D should be set to zero and the simulation should continue from time t_steady till the new steady state is reached (t_steady_second). Run the simulations and plot the results from t = 0 till t_steady_second.

```
# parameters
D <- (20 * 1000)/374.471
parameters <- c(Ks_rm = 2.90, ic50_rm = 26.2, Kon = 0.00329, Kt = 0.63, Kre = 0.57, Rf = 0.49, Kd_R = 0
state <- c(Rmo = 4.74, Ro = 267, DR = 0, DRN = 0, D = D)
times <- seq(0, 400,  by = 1)

# Define the model function for the ode model
Grd_model <- function(t, y, parms){
  with(as.list(c(parms, y)),{
    Rmo.delta <- Ks_rm * (1-(DRN / (ic50_rm + DRN)))- Kd_Rm * Rmo
    R.delta <- Ks_r *  Rmo + Rf * Kre * DRN - Kon * D * Ro - Kd_R * Ro
    DR.delta <- Kon * D * Ro - Kt * DR
    DRN.delta <- Kt * DR - Kre * DRN
    D.delta <- 0
    return(list(c(Rmo.delta, R.delta, DR.delta, DRN.delta, D.delta)))
       })
}

event_1 <- function(t, y, parms){
  y["D"] <-  0
  return(y)
```

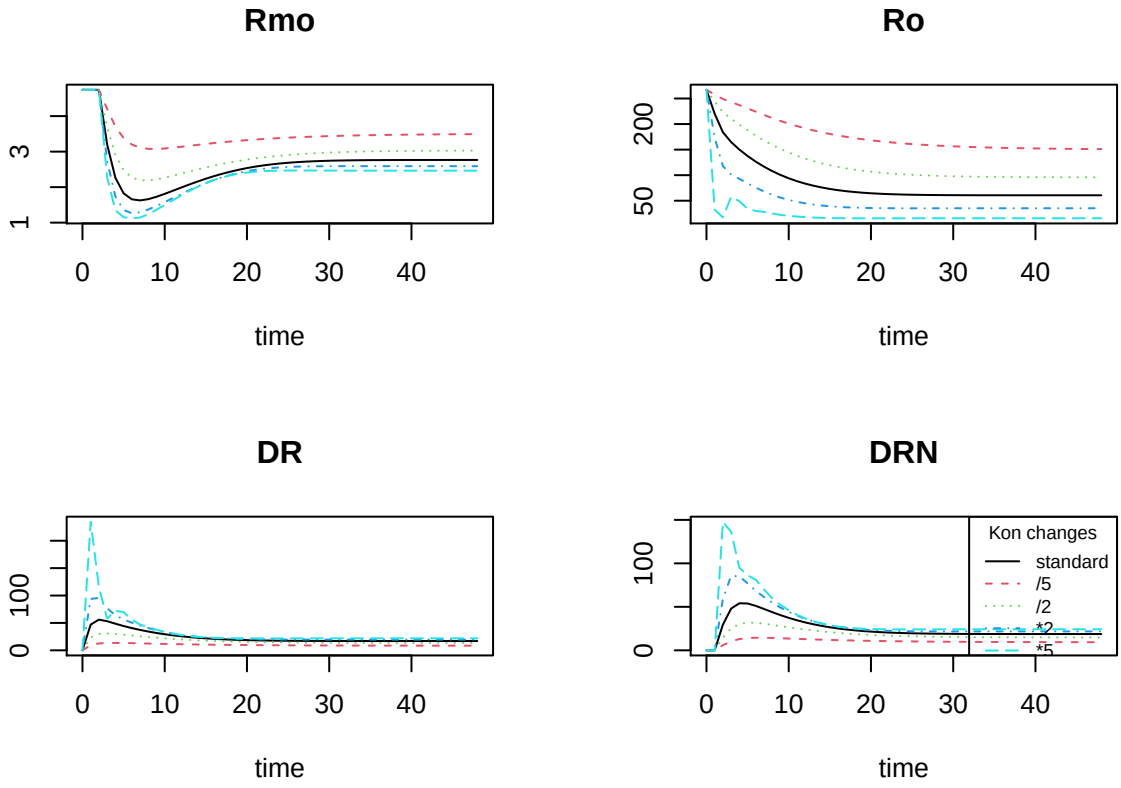
```
}

# checks when all states' delta are absolute and below 1e-4
root <- function(t, y, parms) {
  x <- unlist(Grd_model(t, y, parms))
  number <- sum(abs(x)) - 1e-4
  number2 <- number + y["D"]

  return(c(number, number2))
}

# Run the ode model and save the output to the out variable
out <- ode(times = times, y = state,   parms = parameters, func = Grd_model, rootfun = root,
           events = list(func = event_1, root = TRUE, terminalroot = 2))

last_root_time <- tail(attributes(out)$troot, n=1)
times_basic <- seq(0, round(last_root_time),  by = 1)

out_basic <- ode(times = times_basic, y = state,   parms = parameters, func = Grd_model)
# plot the graph with the data from the ode model
plot(out,out_basic)
```



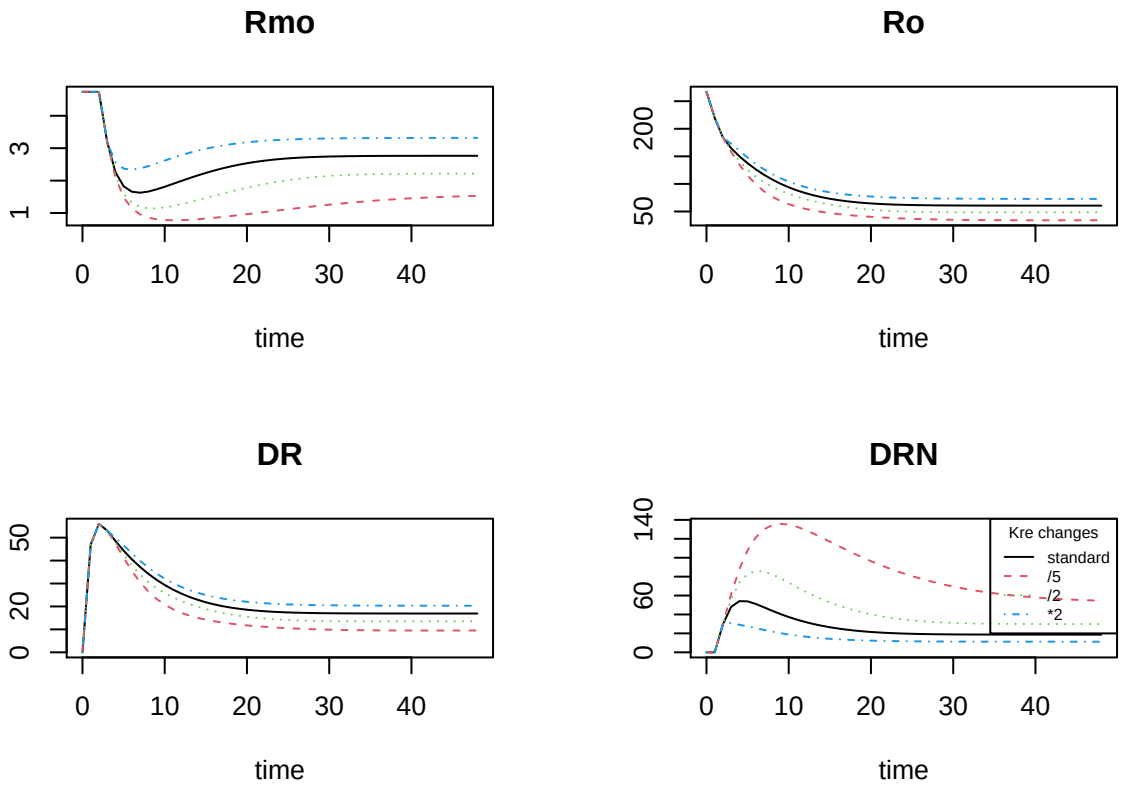
[3] Different corticosteroids show different association rates from receptors (kon) and different dissociation rates (in this model reflected by kre). Assuming the same concentrations of the drug, what is the effect of different values of kon and kre (consider 2 and 5 times increase and decrease of both parameters separately) on the receptor and mRNA dynamics? Adjust kon and kre as below and plot the results of the simulation for

each change. Note: Simulations should be run for 4 new values of kon: 0.00329/5, 0.00329/2, 0.00329*2 and 0.00329*5. The results should be compared to the basic scenario when kon=0.00329 Separately, simulations should be run for 4 new values of kre: 0.57/5, 0.57/2, 0.57*2 and 0.57*5. The results should be compared to the basic scenario when kre= 0.57.

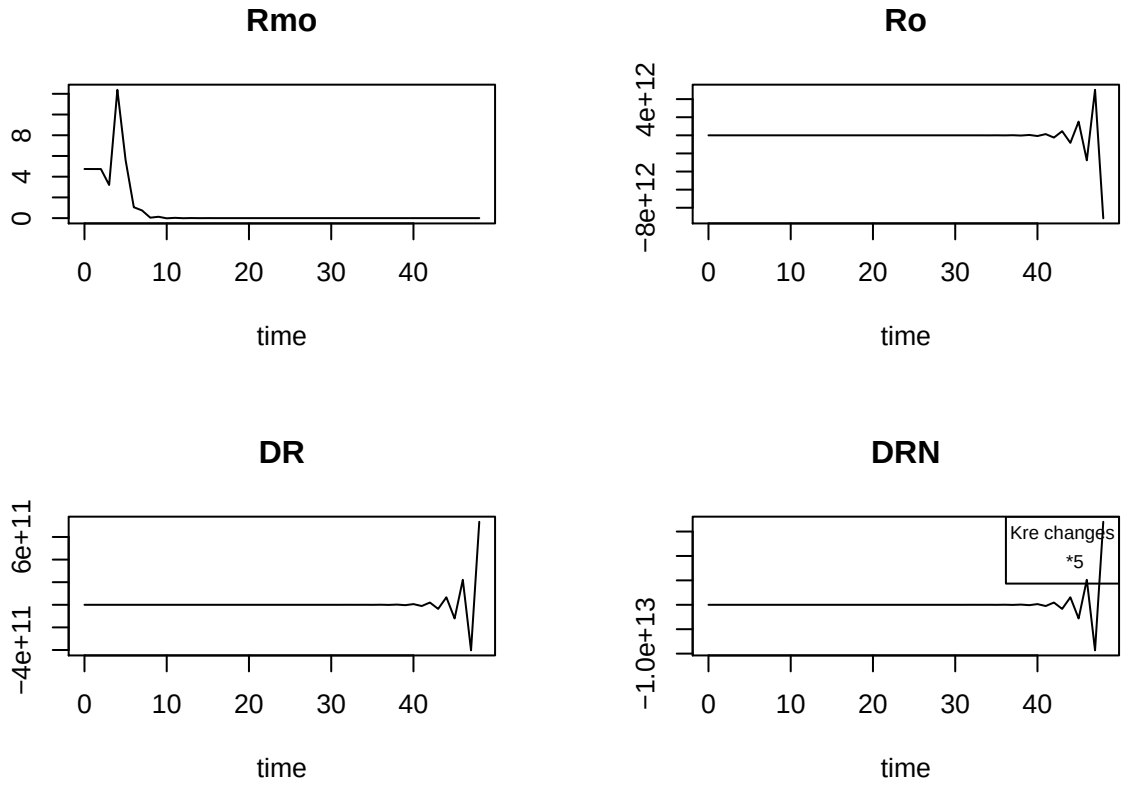
```
# Set the parameters, state and times for the ode model
D <- (20 * 1000)/374.471
parameters <- c(Ks_rm = 2.90, ic50_rm = 26.2, Kon = 0.00329, Kt = 0.63, Kre = 0.57, Rf = 0.49, Kd_R = 0
state <- c(Rmo = 4.74, Ro = 267, DR = 0, DRN = 0)
times <- seq(0, 48,  by = 1)

# Define the function for the ode model
Grd_model <- function(t, y, parms){
  with(as.list(c(parms, y)),{
    Rmo.delta <- Ks_rm * (1-(DRN / (ic50_rm + DRN)))- Kd_Rm * Rmo
    R.delta <- Ks_r *  Rmo + Rf * Kre * DRN - Kon * D * Ro - Kd_R * Ro
    DR.delta <- Kon * D * Ro - Kt * DR
    DRN.delta <- Kt * DR - Kre * DRN

        return(list(c(Rmo.delta, R.delta, DR.delta, DRN.delta)))
       })
}

#  Standard Kon
out <- ode(times = times, y = state, parms = parameters, func = Grd_model, method = "euler")

# Setting kon to 0.0329/5
parameters["Kon"] <- 0.00329/5
out2 <- ode(times = times, y = state, parms = parameters, func = Grd_model, method = "euler")

# setting Kon to 0.0329/2
parameters["Kon"] <- 0.00329/2
out3 <- ode(times = times, y = state, parms = parameters, func = Grd_model, method = "euler")

# setting Kon to 0.0329*2
parameters["Kon"] <- 0.00329*2
out4 <- ode(times = times, y = state, parms = parameters, func = Grd_model, method = "euler")

# setting Kon to 0.0329*5
parameters["Kon"] <- 0.00329*5
out5 <- ode(times = times, y = state, parms = parameters, func = Grd_model, method = "euler")

# Plotting everything Kon related
plot(out, out2, out3, out4, out5)
legend("topright", lty = 1:5, col = 1:5, cex = 0.7, legend = c("standard", "/5", "/2", "*2", "*5"), titl
```




```r
# Resetting Kon to default value and running standard kre
parameters["Kon"] <- 0.00329
out <- ode(times = times, y = state, parms = parameters, func = Grd_model, method = "euler")

# setting kre to 0.57/5
parameters["Kre"] <- 0.57/5
out2 <- ode(times = times, y = state, parms = parameters, func = Grd_model, method = "euler")

# setting kre to 0.57/2
parameters["Kre"] <- 0.57/2
out3 <- ode(times = times, y = state, parms = parameters, func = Grd_model, method = "euler")

# setting kre to 0.57*2
parameters["Kre"] <- 0.57*2
out4 <- ode(times = times, y = state, parms = parameters, func = Grd_model, method = "euler")

# setting kre to 0.57*5
parameters["Kre"] <- 0.57*5
out5 <- ode(times = times, y = state, parms = parameters, func = Grd_model, method = "euler")

# Plotting everything Kre related
plot(out, out2, out3, out4)
legend("topright", lty = 1:4, col = 1:4, cex =0.6, legend = c("standard", "/5", "/2", "*2"), title= "Kr
```




```
# plotted this scenario sepertately due to the value difference, it would ruin the plots
plot(out5)
legend("topright", legend = "*5", cex = 0.7, title = "Kre changes")
```



If the legend isn't showing properly in the PDF: First group of graphs are for Kon changes: The white line is for the standard Kon value. The red dotted line is for 5 times lower Kon value. The green dotted line is for 2 times lower Kon value. The blue dotted line is for 2 times higher Kon value. The cyan dotted line is for 5 times higher Kon value.

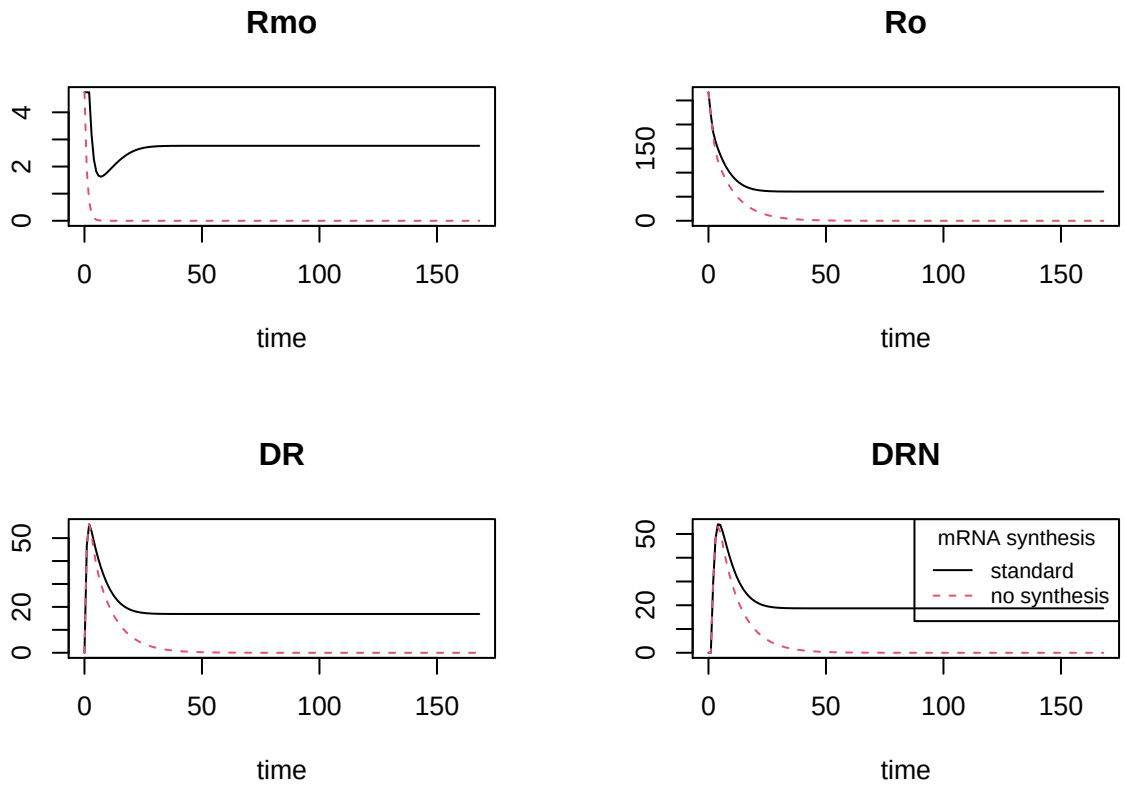
Second & third group of graphs are for Kre changes The white line in the first graph with multiple lines is for standard Kre value. The red dotted line is for 5 times lower Kre value. The green dotted line is for 2 times lower Kre value. The blue dotted line is for 2 times higher Kre value. The solo plotted white line is for 5 times higher Kre value.

[4] What would happen if the synthesis of the receptor was completely blocked? Which parameter needs to be put to zero? Adjust the parameter, run the simulations and plot the results.

Ks_rm needs to be set to zero, as this is the constant for mRNA synthesis.

```
D <- (20 * 1000)/374.471
parameters <- c(Ks_rm = 2.90, ic50_rm = 26.2, Kon = 0.00329, Kt = 0.63, Kre = 0.57, Rf = 0.49, Kd_R = 0
state <- c(Rmo = 4.74, Ro = 267, DR = 0, DRN = 0)
times <- seq(0, 168,  by = 1)

Grd_model <- function(t, y, parms){
  with(as.list(c(parms, y)),{
    Rmo.delta <- Ks_rm * (1-(DRN / (ic50_rm + DRN)))- Kd_Rm * Rmo
    R.delta <- Ks_r *  Rmo + Rf * Kre * DRN - Kon * D * Ro - Kd_R * Ro
    DR.delta <- Kon * D * Ro - Kt * DR
    DRN.delta <- Kt * DR - Kre * DRN

        return(list(c(Rmo.delta, R.delta, DR.delta, DRN.delta)))
```

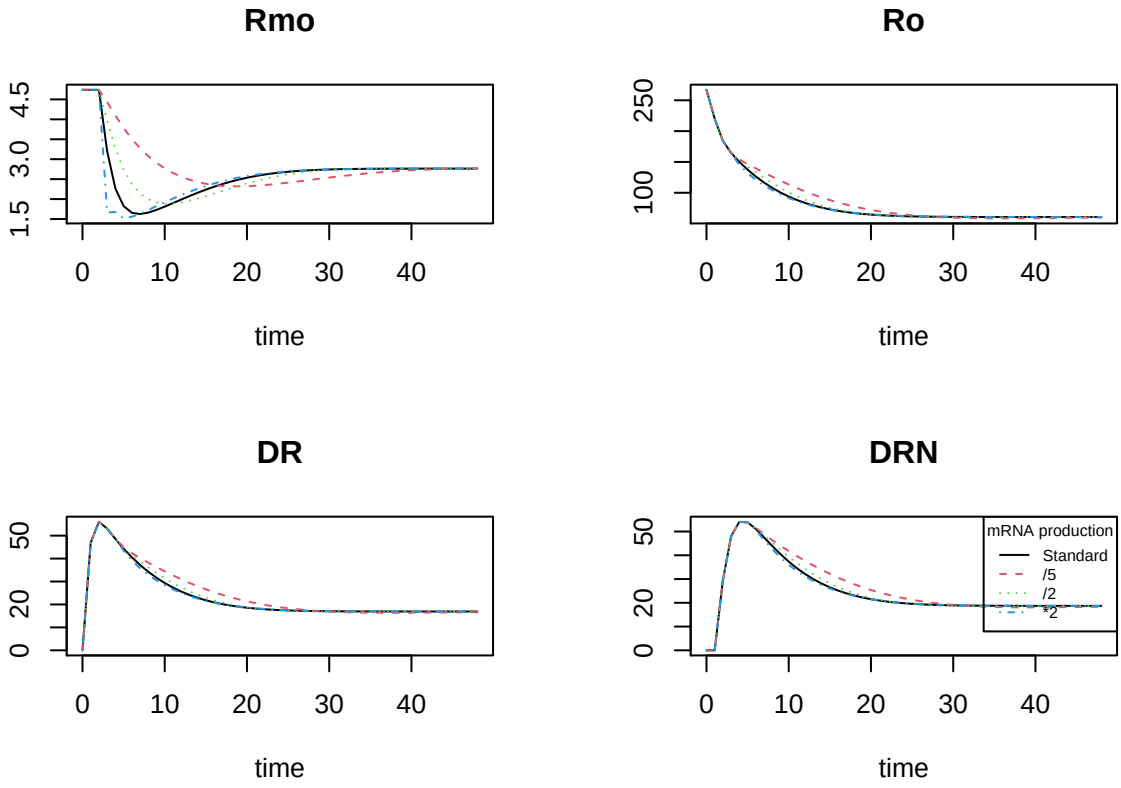
```
       })
}

# standard plot
out <- ode(times = times, y = state,   parms = parameters, func = Grd_model, method = "euler")

# setting mRNA synthesis to 0 (ks_Rm)
parameters["Ks_rm"] <- 0
out2 <- ode(times = times, y = state,   parms = parameters, func = Grd_model, method = "euler")

plot(out, out2)
legend("topright", lty = 1:2, col = 1:2, cex = 0.8, legend = c("standard", "no synthesis"), title = "mR
```



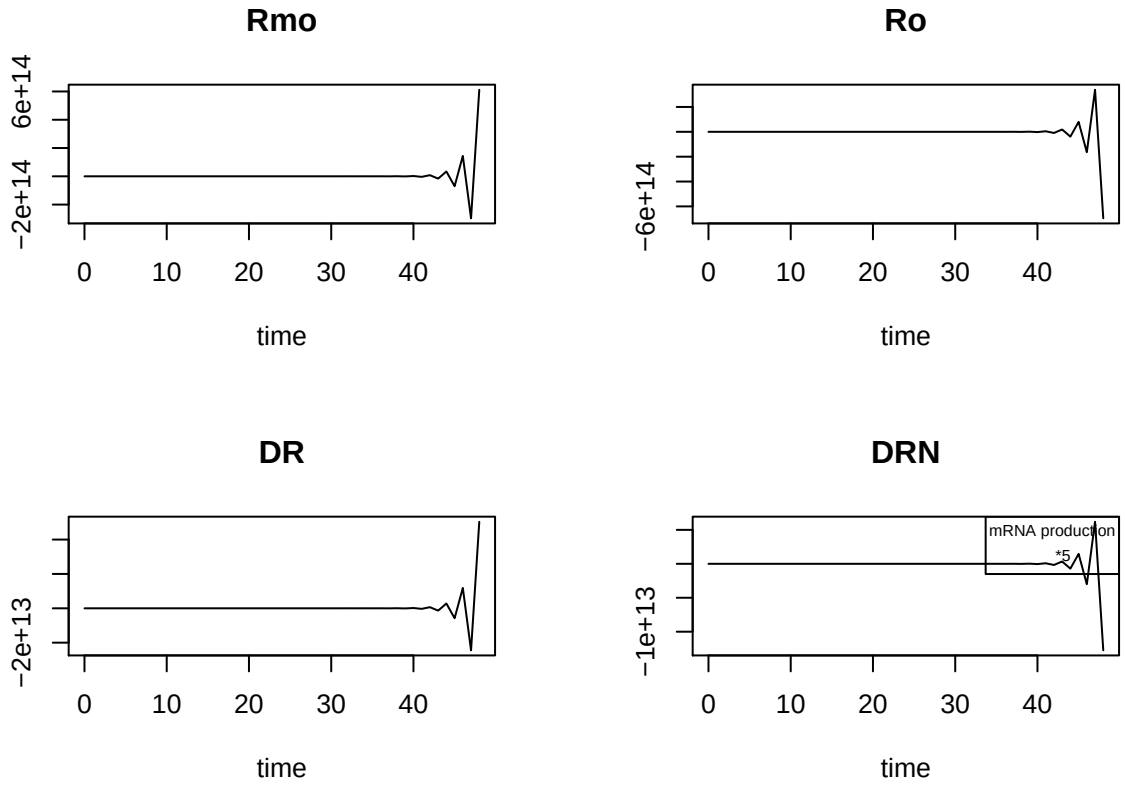
If the legend in the PDF isn't showing properly: The white line is for the standard mRNA synthesis. The red dotted line is for no mRNA synthesis.

[5] What is the dynamic of the system when the baseline rate of production of mRNA of the receptor is increased or decreased 2 or 5 fold (recalculate the rate of mRNA degradation so that the steady-state assumption at baseline (without the drug) is still valid, i.e. mRNA levels are constant when there is not drug)? Mind you: ks_Rm values should be changed, but we know that if without the drug the system is at steady-state then kd_Rm = ks_Rm/Rm0. Therefore if we change ks_Rm we need to change kd_Rm as well. Also after we recalculate the value of kd_Rm for the baseline conditions, the simulations should be run with drug present. Simulations should be run for 4 different scenarios: ks_Rm = 2.9/5 and kd_Rm=2.9/5/4.74 ks_Rm = 2.9/2 and kd_Rm=2.9/2/4.74 ks_Rm = 2.9*2 and kd_Rm=2.9*2/4.74 ks_Rm = 2.9*5 and kd_Rm=2.9*5/4.74

```r
# Set the parameters, state and times for the ode model
D <- (20 * 1000)/374.471
parameters <- c(Ks_rm = 2.90, ic50_rm = 26.2, Kon = 0.00329, Kt = 0.63, Kre = 0.57, Rf = 0.49, Kd_R = 0
state <- c(Rmo = 4.74, Ro = 267, DR = 0, DRN = 0)
times <- seq(0, 48,  by = 1)

# Define the function for the ode model.
Grd_model <- function(t, y, parms){
  with(as.list(c(parms, y)),{
    Rmo.delta <- Ks_rm * (1-(DRN / (ic50_rm + DRN)))- Kd_Rm * Rmo
    R.delta <- Ks_r *  Rmo + Rf * Kre * DRN - Kon * D * Ro - Kd_R * Ro
    DR.delta <- Kon * D * Ro - Kt * DR
    DRN.delta <- Kt * DR - Kre * DRN

        return(list(c(Rmo.delta, R.delta, DR.delta, DRN.delta)))
       })
}

# standard scenario
out <- ode(times = times, y = state,   parms = parameters, func = Grd_model, method = "euler")

# rate of production divided by 5
parameters["Ks_rm"] <- 2.9/5
parameters["Kd_Rm"] <- parameters["Ks_rm"]/4.74
out2 <- ode(times = times, y = state,   parms = parameters, func = Grd_model, method = "euler")

# rate of production divided by 2
parameters["Ks_rm"] <- 2.9/2
parameters["Kd_Rm"] <- parameters["Ks_rm"]/4.74
out3 <- ode(times = times, y = state,   parms = parameters, func = Grd_model, method = "euler")

# rate of production multiplied by 2
parameters["Ks_rm"] <- 2.9*2
parameters["Kd_Rm"] <- parameters["Ks_rm"]/4.74
out4 <- ode(times = times, y = state,   parms = parameters, func = Grd_model, method = "euler")

# rate of production multiplied by 5
parameters["Ks_rm"] <- 2.9*5
parameters["Kd_Rm"] <- parameters["Ks_rm"]/4.74
out5 <- ode(times = times, y = state,   parms = parameters, func = Grd_model, method = "euler")

plot(out, out2, out3, out4)
legend("topright", lty = 1:4, col = 1:4, cex = 0.6, legend = c("Standard", "/5", "/2", "*2"), title = "
```




```
# plotted this scenario sepertately due to the value difference, it would ruin the plots
plot(out5)
legend("topright", legend = "*5", cex = 0.6, title = "mRNA production")
```



If the legend isn't showing properly in the PDF: The white line in the first graph with multiple lines is for standard mRNA production. The red dotted line is for 5 times less mRNA production. The green dotted line is for 2 times less mRNA production. The blue dotted line is for 2 times more mRNA production. The single plotted white line is for 5 times more mRNA production.