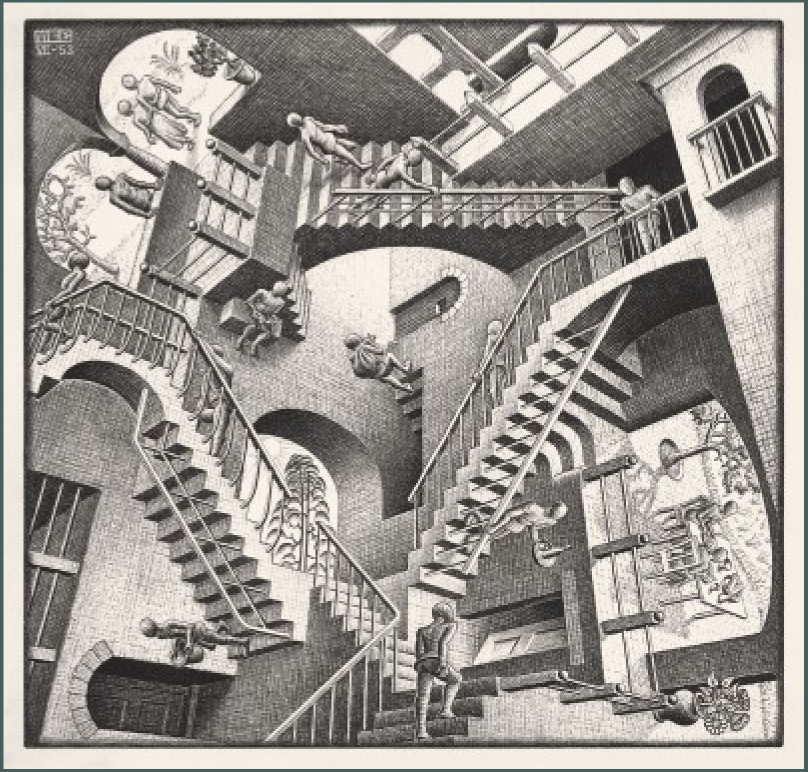
2.3 - Forest water/energy balance (exercise)

Miquel De Cáceres, Victor Granda, Aitor Ameztegui

Ecosystem Modelling Facility

2022-06-14



Exercise setting

Overall goal

Learn how to use *medfate* for forest water balance simulations.

Exercise setting

Overall goal

Learn how to use *medfate* for forest water balance simulations.

Specific objectives

1. Perform a basic water balance run on a real-case data and inspect the results
2. Evaluate the performance of the water balance model with observed data
3. Perform an advanced water balance run on the same data and inspect the results
4. Compare the results and performance between the two models

Exercise setting

Overall goal

Learn how to use *medfate* for forest water balance simulations.

Specific objectives

1. Perform a basic water balance run on a real-case data and inspect the results
2. Evaluate the performance of the water balance model with observed data
3. Perform an advanced water balance run on the same data and inspect the results
4. Compare the results and performance between the two models

Exercise material

- Exercise_2.Rmd
- fontblanche.rds

Exercise setting

Font-Blanche research forest

- The Font-Blanche research forest is located in southeastern France (43°14'27'' N 5°40'45'' E) at 420 m elevation)

Exercise setting

Font-Blanche research forest

- The Font-Blanche research forest is located in southeastern France (43°14'27'' N 5°40'45'' E) at 420 m elevation)
- The stand is composed of a top strata of *Pinus halepensis* (Aleppo pine) reaching about 12 m, a lower strata of *Quercus ilex* (holm oak), reaching about 6 m, and an understorey strata dominated by *Quercus coccifera* and *Phillyrea latifolia*.

Exercise setting

Font-Blanche research forest

- The Font-Blanche research forest is located in southeastern France (43º14'27'' N 5°40'45'' E) at 420 m elevation)
- The stand is composed of a top strata of *Pinus halepensis* (Aleppo pine) reaching about 12 m, a lower strata of *Quercus ilex* (holm oak), reaching about 6 m, and an understorey strata dominated by *Quercus coccifera* and *Phillyrea latifolia*.
- Soils are shallow and rocky, have a low retention capacity and are of Jurassic limestone origin.

Exercise setting

Font-Blanche research forest

- The Font-Blanche research forest is located in southeastern France (43º14'27'' N 5°40'45'' E) at 420 m elevation)
- The stand is composed of a top strata of *Pinus halepensis* (Aleppo pine) reaching about 12 m, a lower strata of *Quercus ilex* (holm oak), reaching about 6 m, and an understorey strata dominated by *Quercus coccifera* and *Phillyrea latifolia*.
- Soils are shallow and rocky, have a low retention capacity and are of Jurassic limestone origin.
- The climate is Mediterranean, with a water stress period in summer, cold or mild winters and most precipitation occurring between September and May.

Exercise setting

Font-Blanche research forest

- The Font-Blanche research forest is located in southeastern France (43º14'27'' N 5°40'45'' E) at 420 m elevation)
- The stand is composed of a top strata of *Pinus halepensis* (Aleppo pine) reaching about 12 m, a lower strata of *Quercus ilex* (holm oak), reaching about 6 m, and an understorey strata dominated by *Quercus coccifera* and *Phillyrea latifolia*.
- Soils are shallow and rocky, have a low retention capacity and are of Jurassic limestone origin.
- The climate is Mediterranean, with a water stress period in summer, cold or mild winters and most precipitation occurring between September and May.

Target stand

- The experimental site, which is dedicated to study forest carbon and water cycles, has an enclosed area of 80×80 m but our target stand is a quadrat of dimensions 25×25 m.

Exercise setting

Font-Blanche research forest

- The Font-Blanche research forest is located in southeastern France (43°14'27'' N 5°40'45'' E) at 420 m elevation)
- The stand is composed of a top strata of *Pinus halepensis* (Aleppo pine) reaching about 12 m, a lower strata of *Quercus ilex* (holm oak), reaching about 6 m, and an understorey strata dominated by *Quercus coccifera* and *Phillyrea latifolia*.
- Soils are shallow and rocky, have a low retention capacity and are of Jurassic limestone origin.
- The climate is Mediterranean, with a water stress period in summer, cold or mild winters and most precipitation occurring between September and May.

Target stand

- The experimental site, which is dedicated to study forest carbon and water cycles, has an enclosed area of 80×80 m but our target stand is a quadrat of dimensions 25×25 m.
- The following observations are available for year 2014:
    - Stand total evapotranspiration estimated using an Eddy-covariance flux tower.
    - Soil moisture content of the topmost (0-30 cm) layer.
    - Transpiration estimates per leaf area, derived from sapflow measurements for *Q. ilex* and *P. halepensis*.
    - Pre-dawn and midday leaf water potentials for *Q. ilex* and *P. halepensis*.

Exercise solution

Step 1. Load Font-Blanche data

We are given all the necessary data, bundled in a single list:

```
fb <- readRDS("StudentRdata/fontblanche.rds")
names(fb)
```

```
## [1] "treeData"     "shrubData"    "customParams" "measuredData" "meteoData"    "soilData"
## [7] "terrainData"
```

Exercise solution

Step 1. Load Font-Blanche data

We are given all the necessary data, bundled in a single list:

```
fb <- readRDS("StudentRdata/fontblanche.rds")
names(fb)
```

```
## [1] "treeData"     "shrubData"    "customParams" "measuredData" "meteoData"    "soilData"
## [7] "terrainData"
```

Step 2. Build forest object

We can easily assemble the tree and shrub data into a forest object:

```
fb_forest <- emptyforest()
fb_forest$treeData <- fb$treeData
fb_forest$shrubData <- fb$shrubData
```

and examine its characteristics:

```
summary(fb_forest, SpParamsMED)
```

```
## Tree density (ind/ha): 4608
## Tree BA (m2/ha): 24.4861797
## Cover (%) trees (open ground): 100  shrubs: 0
## Shrub crown phytovolume (m3/m2): 0
## LAI (m2/m2) total: 3.0064027  trees: 3.0064027  shrubs: 0
## Live fine fuel (kg/m2) total: 0.9520124  trees: 0.9520124  shrubs: 0
```

Exercise solution

Step 3. Build soil object

A data frame with soil physical attributes are defined in:

```
fb$soilData
```

```
##   widths clay sand om   bd rfc
## 1    300   39   26  6 1.45  50
## 2    700   39   26  3 1.45  65
## 3   1000   39   26  1 1.45  85
## 4   2500   39   26  1 1.45  90
```

Exercise solution

Step 3. Build soil object

A data frame with soil physical attributes are defined in:

```
fb$soilData
```

```
##   widths clay sand om   bd rfc
## 1    300   39   26  6 1.45  50
## 2    700   39   26  3 1.45  65
## 3   1000   39   26  1 1.45  85
## 4   2500   39   26  1 1.45  90
```

We need, however, to build a `soil` object:

```
fb_soil <- soil(fb$soilData)
```

Exercise solution

Step 3. Build soil object

A data frame with soil physical attributes are defined in:

```
fb$soilData
```

```
##   widths clay sand om   bd rfc
## 1    300   39   26  6 1.45  50
## 2    700   39   26  3 1.45  65
## 3   1000   39   26  1 1.45  85
## 4   2500   39   26  1 1.45  90
```

We need, however, to build a `soil` object:

```
fb_soil <- soil(fb$soilData)
```

From which we can estimate the extractable water capacity for each layer (in mm):

```
soil_waterExtractable(fb_soil)
```

```
## [1] 26.06443 41.96683 25.45599 42.42664
```

Exercise solution

Step 3. Build soil object

A data frame with soil physical attributes are defined in:

```
fb$soilData
```

```
##   widths clay sand om   bd rfc
## 1    300   39   26  6 1.45  50
## 2    700   39   26  3 1.45  65
## 3   1000   39   26  1 1.45  85
## 4   2500   39   26  1 1.45  90
```

We need, however, to build a `soil` object:

```
fb_soil <- soil(fb$soilData)
```

From which we can estimate the extractable water capacity for each layer (in mm):

```
soil_waterExtractable(fb_soil)
```

```
## [1] 26.06443 41.96683 25.45599 42.42664
```

The same information can be found in the output of `print()`.

Exercise solution

Step 4. Species parameters

We will normally take `SpParamsMED` as starting point for species parameters:

```
data("SpParamsMED")
```

However, sometimes one may wish to override species defaults with custom values. In the case of FontBlanche there is a table of preferred values for some parameters, especially in the case of *Quercus ilex* (code 168):

```
fb$customParams
```

```
##   SpIndex Cohort   g Kmax_stemxylem VCleaf_kmax VCleaf_c VCleaf_d LeafPI0 LeafEPS LeafAF    Al2As
## 1     142 T1_142 0.8             NA        3.00       NA       NA      NA      NA     NA       NA
## 2     148 T2_148 1.0             NA        4.00       NA       NA      NA      NA     NA  631.000
## 3     168 T3_168 0.8            0.4        2.63     5.41    -4.18   -2.66   10.57   0.43 1540.671
```

Exercise solution

Step 4. Species parameters

We will normally take `SpParamsMED` as starting point for species parameters:

```
data("SpParamsMED")
```

However, sometimes one may wish to override species defaults with custom values. In the case of FontBlanche there is a table of preferred values for some parameters, especially in the case of *Quercus ilex* (code 168):

```
fb$customParams
```

```
##   SpIndex Cohort   g Kmax_stemxylem VCleaf_kmax VCleaf_c VCleaf_d LeafPI0 LeafEPS LeafAF    Al2As
## 1     142 T1_142 0.8             NA        3.00       NA       NA      NA      NA     NA       NA
## 2     148 T2_148 1.0             NA        4.00       NA       NA      NA      NA     NA  631.000
## 3     168 T3_168 0.8            0.4        2.63     5.41    -4.18   -2.66   10.57   0.43 1540.671
```

We can use function `modifySpParams()` to replace the values of parameters for the desired traits, leaving the rest unaltered:

```
fb_SpParams <- modifySpParams(SpParamsMED, fb$customParams)
```

Exercise solution

Steps 5-6. Basic water balance

Since we are about to run a basic water balance simulation, we initialize a simulation control parameter list with `transpirationMode = "Granier"`, i.e.:

```
fb_control <- defaultControl("Granier")
```

and we assemble our inputs into a `spwbInput` object, using:

```
fb_x1 <- forest2spwbInput(fb_forest, fb_soil, fb_SpParams, fb_control)
```

Exercise solution

Steps 5-6. Basic water balance

Since we are about to run a basic water balance simulation, we initialize a simulation control parameter list with `transpirationMode = "Granier"`, i.e.:

```
fb_control <- defaultControl("Granier")
```

and we assemble our inputs into a `spwbInput` object, using:

```
fb_x1 <- forest2spwbInput(fb_forest, fb_soil, fb_SpParams, fb_control)
```

The daily weather data comprises one year:

```
fb_meteo <- fb$meteoData
nrow(fb_meteo)
```

```
## [1] 365
```

Exercise solution

Steps 5-6. Basic water balance

Since we are about to run a basic water balance simulation, we initialize a simulation control parameter list with `transpirationMode = "Granier"`, i.e.:

```
fb_control <- defaultControl("Granier")
```

and we assemble our inputs into a `spwbInput` object, using:

```
fb_x1 <- forest2spwbInput(fb_forest, fb_soil, fb_SpParams, fb_control)
```

The daily weather data comprises one year:

```
fb_meteo <- fb$meteoData
nrow(fb_meteo)
```

```
## [1] 365
```

Now, we are ready to launch the simulation:

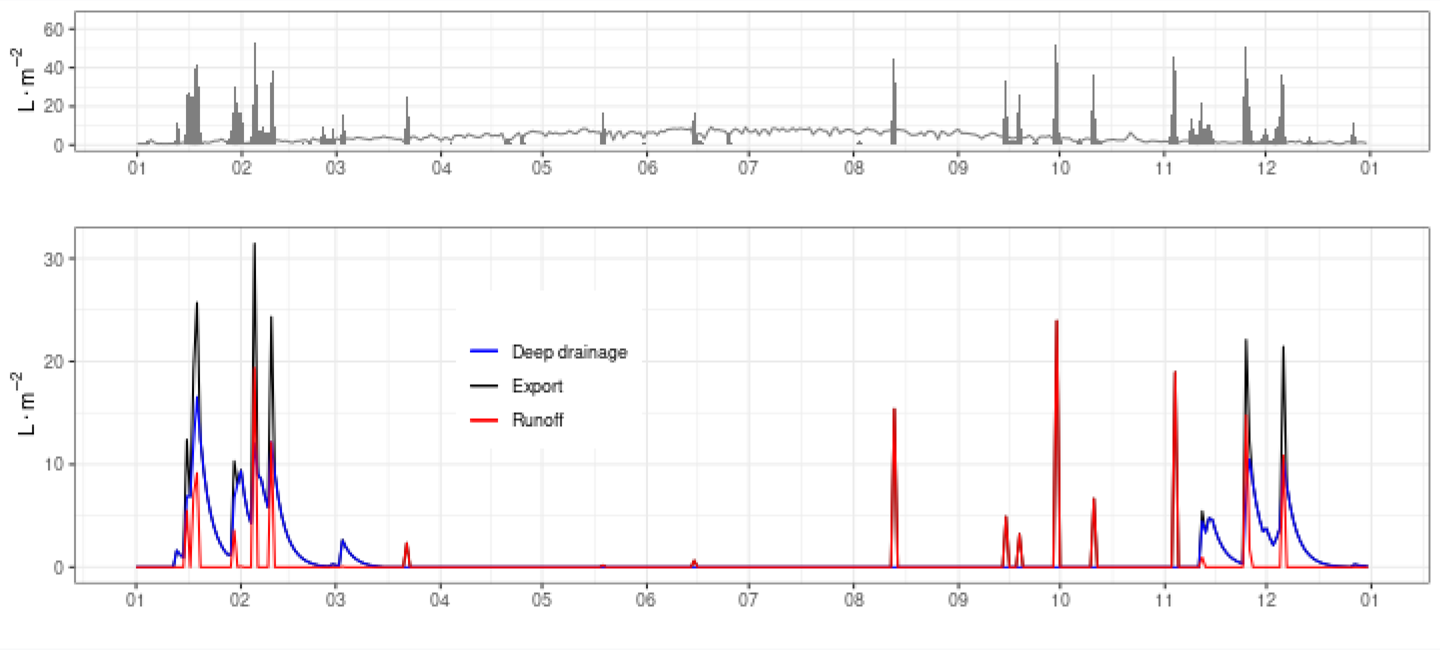
```
fb_basic <- spwb(fb_x1, fb_meteo,  elevation = 420, latitude = 43.24083)
```

Exercise solution

Step 7. Examine precipitation events, runoff and deep drainage

Surface run-off occurs the same day as precipitation events, whereas deep drainage can last for some days after the event:

```
g1<-plot(fb_basic)+scale_x_date(date_breaks = "1 month", date_labels = "%m")+
  theme(legend.position = "none")
g2<-plot(fb_basic, "Export")+scale_x_date(date_breaks = "1 month", date_labels = "%m")+
  theme(legend.position = c(0.35,0.60))
plot_grid(g1,g2, ncol=1, rel_heights = c(0.5,1))
```

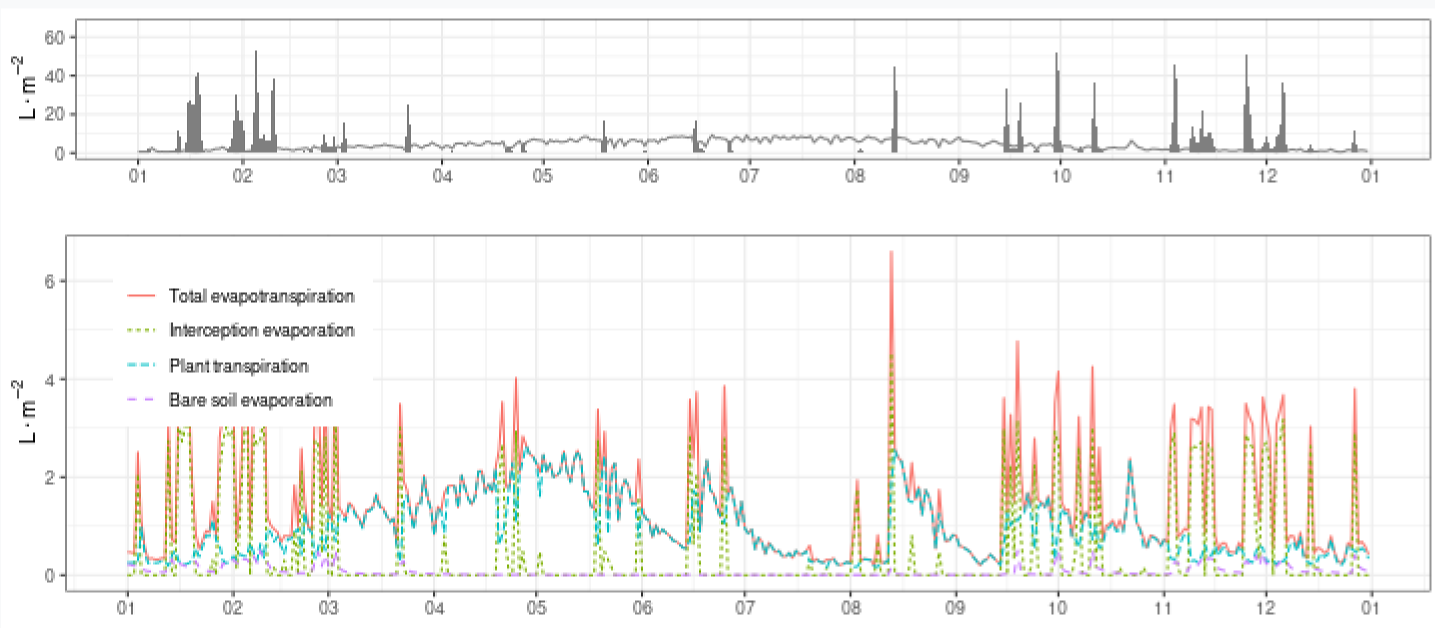


Exercise solution

Step 8. Examine evapotranspiration flows

Precipitation events also generate flows of intercepted water the same day of the event. Evaporation from the bare soil can proceed some days after the event. Transpiration flow is the dominant one in most days, decreasing in summer due to drought.

```
g1<-plot(fb_basic)+scale_x_date(date_breaks = "1 month", date_labels = "%m")+
  theme(legend.position = "none")
g2<-plot(fb_basic, "Evapotranspiration")+scale_x_date(date_breaks = "1 month", date_labels = "%m"
  theme(legend.position = c(0.13,0.73))
plot_grid(g1,g2, ncol=1, rel_heights = c(0.5,1))
```

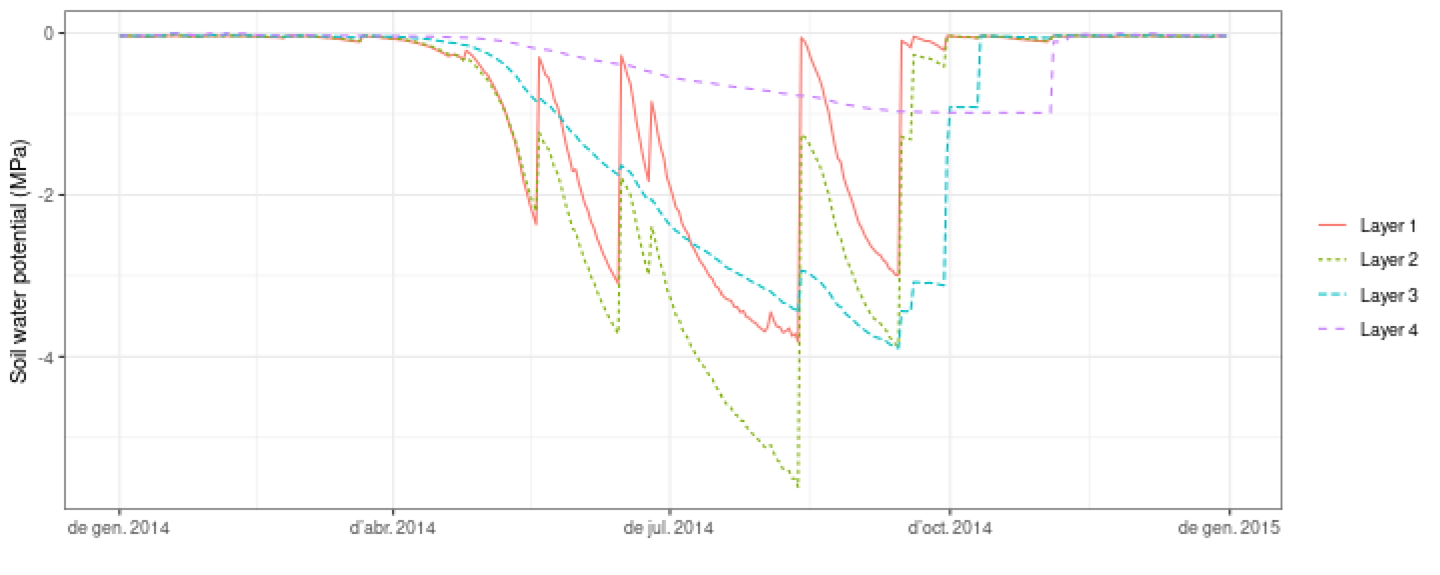


Exercise solution

Step 9. Soil water potential dynamics

We can display the dynamics of water potential in different soil layers using:

```
plot(fb_basic, "SoilPsi")
```

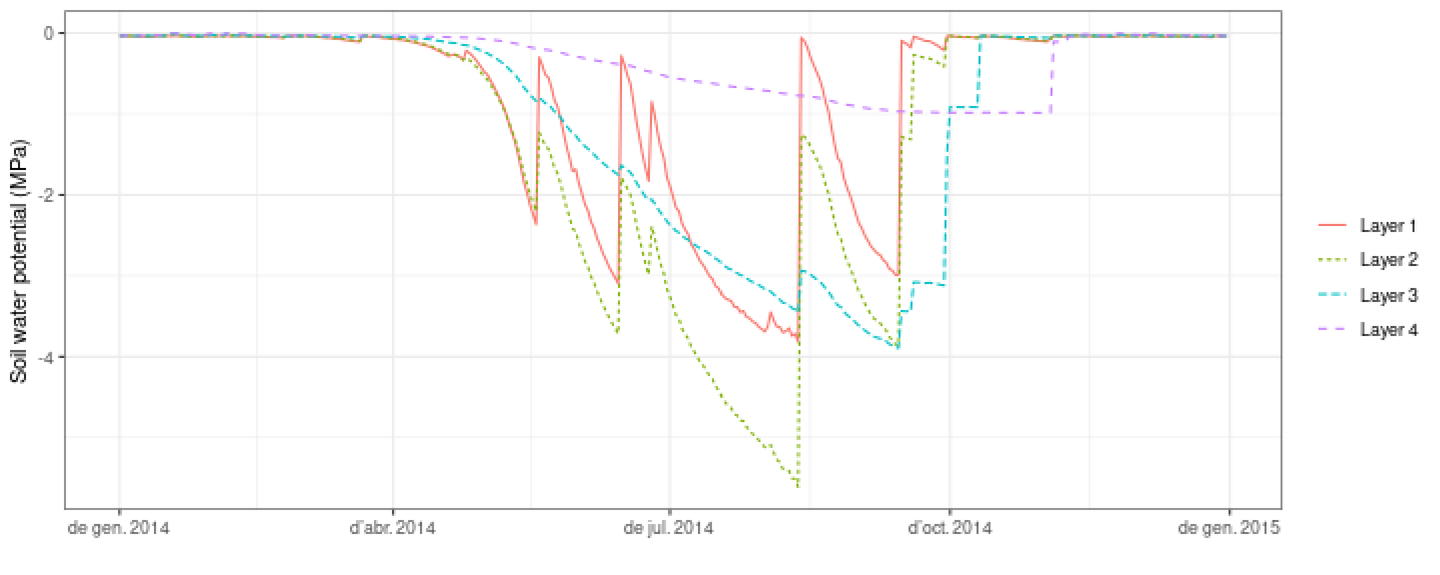


Exercise solution

Step 9. Soil water potential dynamics

We can display the dynamics of water potential in different soil layers using:

```
plot(fb_basic, "SoilPsi")
```



Tip: Normally, we should expect lower layers to have a less dynamic behaviour, but strange results can occur if, for instance, a large proportion of roots is in deeper layers.

Exercise solution

Steps 10-12. Evaluation of stand evapotranspiration

Observations are in element `measuredData` of the list:

```
fb_observed <- fb$measuredData
```

Exercise solution

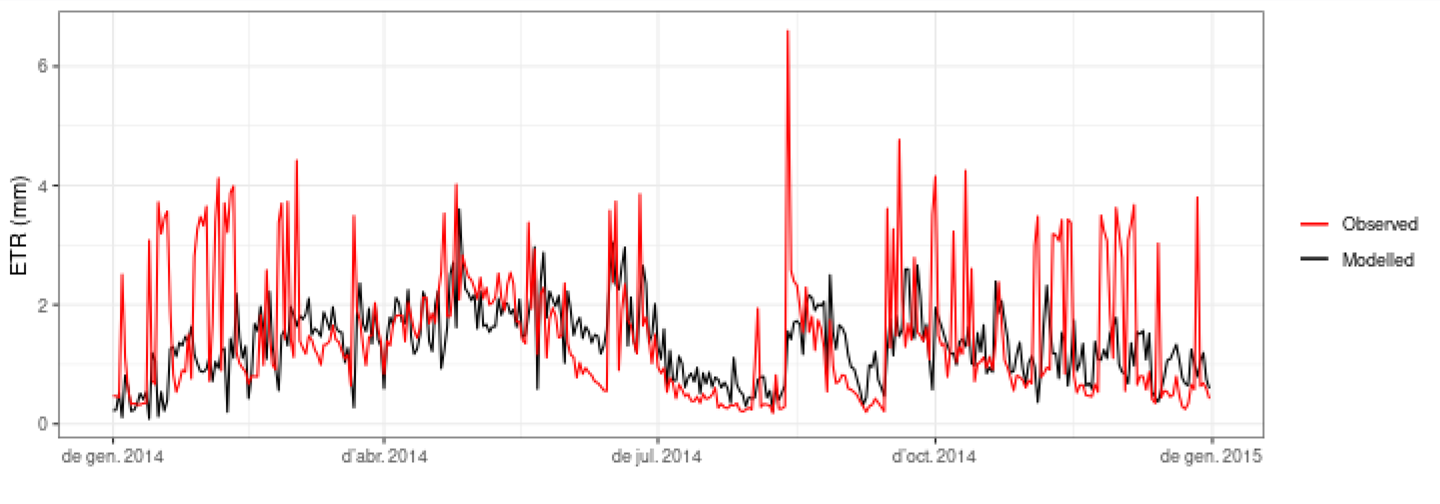
Steps 10-12. Evaluation of stand evapotranspiration

Observations are in element `measuredData` of the list:

```
fb_observed <- fb$measuredData
```

We can compare the observed vs modelled total evapotranspiration by plotting the two time series:

```
evaluation_plot(fb_basic, fb_observed, type = "ETR", plotType="dynamics")
```



Exercise solution

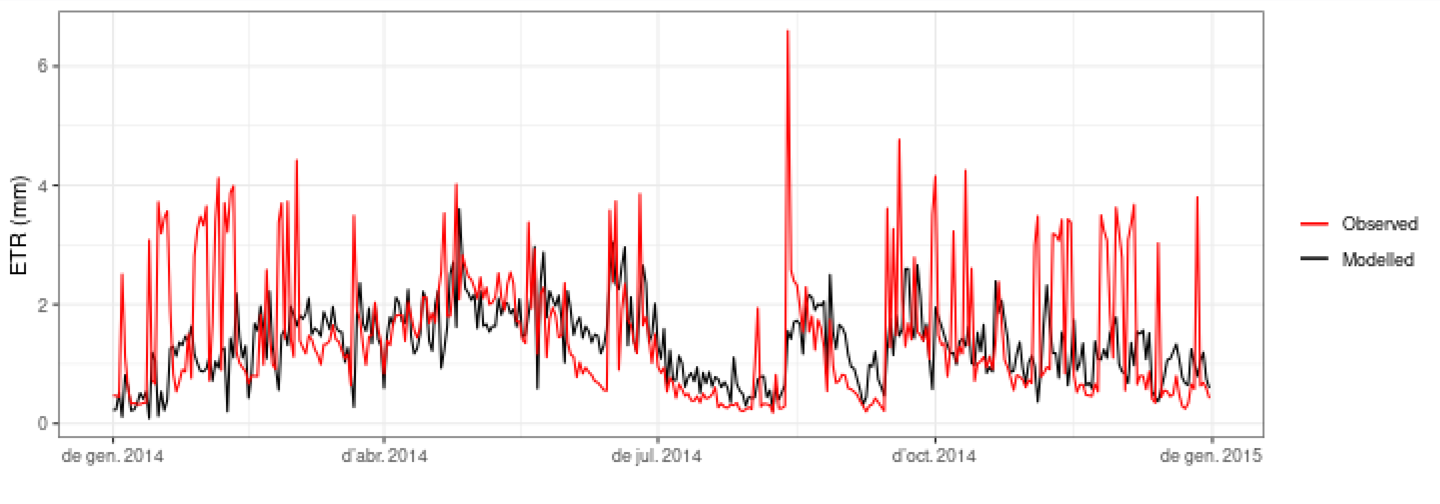
Steps 10-12. Evaluation of stand evapotranspiration

Observations are in element `measuredData` of the list:

```
fb_observed <- fb$measuredData
```

We can compare the observed vs modelled total evapotranspiration by plotting the two time series:

```
evaluation_plot(fb_basic, fb_observed, type = "ETR", plotType="dynamics")
```



It is easy to see that in rainy days the predicted evapotranspiration is much higher than that of the observed data.

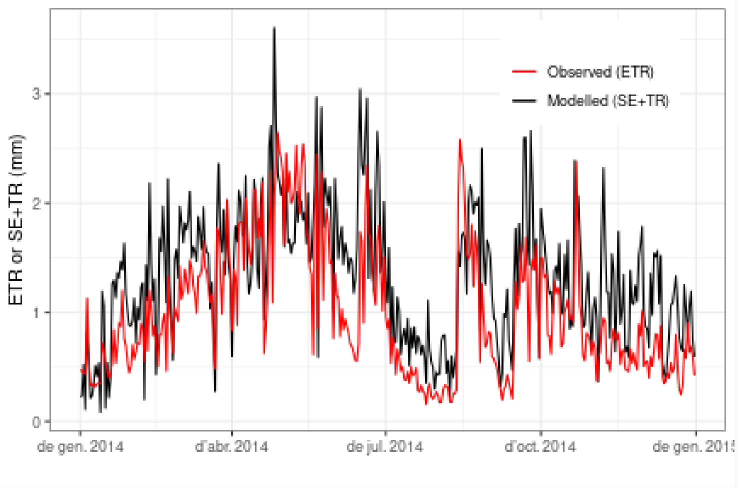
Exercise solution

Steps 10-12. Evaluation of stand evapotranspiration

We repeat the comparison but excluding the intercepted water from modeled results:

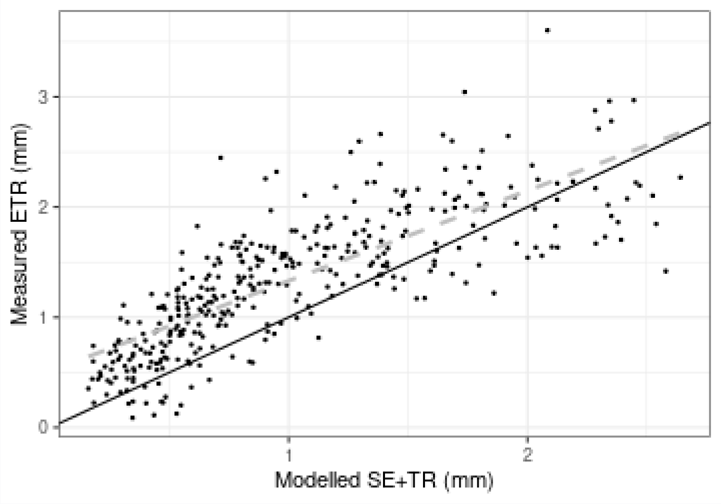
Time series plot

```
evaluation_plot(fb_basic, fb_observed,
    type = "SE+TR", plotType="dynamics")+
  theme(legend.position = c(0.8,0.85))
```



Scatter plot

```
evaluation_plot(fb_basic, fb_observed,
    type = "SE+TR", plotType="scatter")
```



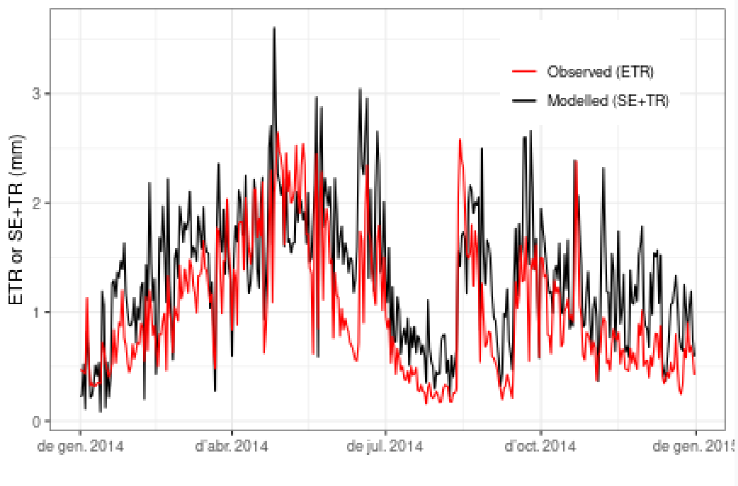
Exercise solution

Steps 10-12. Evaluation of stand evapotranspiration

We repeat the comparison but excluding the intercepted water from modeled results:

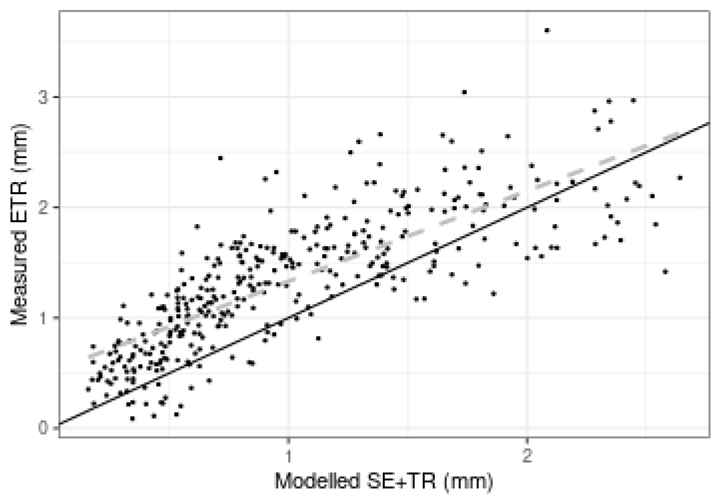
Time series plot

```
evaluation_plot(fb_basic, fb_observed,
    type = "SE+TR", plotType="dynamics")+
  theme(legend.position = c(0.8,0.85))
```



Scatter plot

```
evaluation_plot(fb_basic, fb_observed,
    type = "SE+TR", plotType="scatter")
```



Where we see a reasonably good relationship, but the model tends to underestimate total evapotranspiration during seasons with low evaporative demand.

Exercise solution

Steps 10-12. Evaluation of stand evapotranspiration

Function `evaluation_stats()` allows us to generate evaluation statistics:

```
evaluation_stats(fb_basic, fb_observed, type = "SE+TR")
```

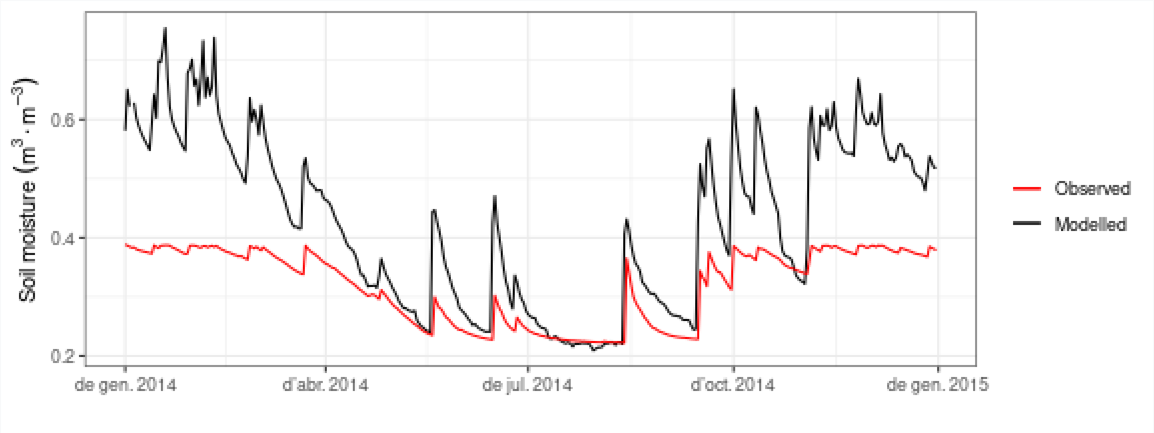
```
##           n        Bias    Bias.rel         MAE     MAE.rel           r         NSE     NSE.abs
## 365.0000000  -0.3296444 -24.7410562   0.4264928  32.0098958   0.7901774   0.3061136   0.1434467
```

Exercise solution

Step 13. Evaluation of soil moisture content

We can now compare the soil moisture content dynamics using:

```
evaluation_plot(fb_basic, fb_observed, type = "SWC", plotType="dynamics")
```

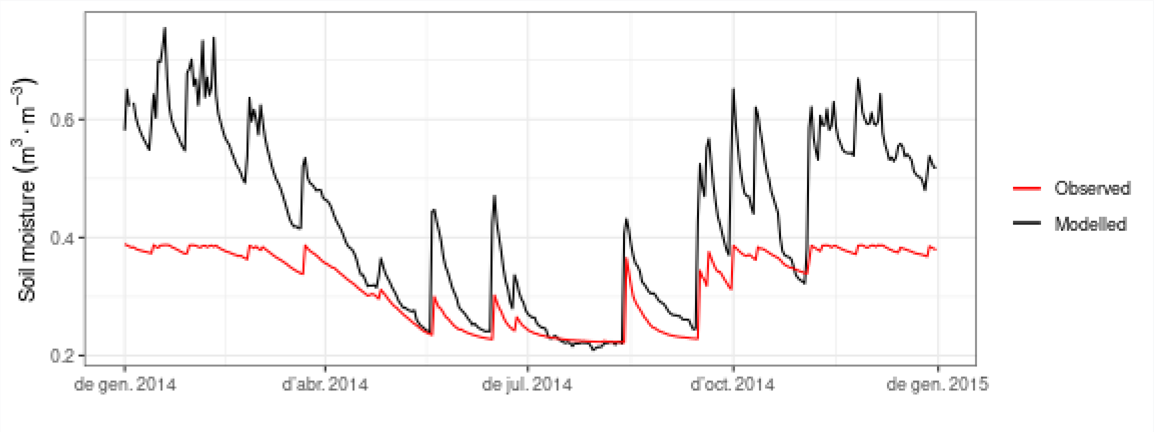


Exercise solution

Step 13. Evaluation of soil moisture content

We can now compare the soil moisture content dynamics using:

```
evaluation_plot(fb_basic, fb_observed, type = "SWC", plotType="dynamics")
```



The two series have similar shape but not absolute values. This may be an indication that the parameters of the soil water retention curve do not match the data produced by the moisture sensor.

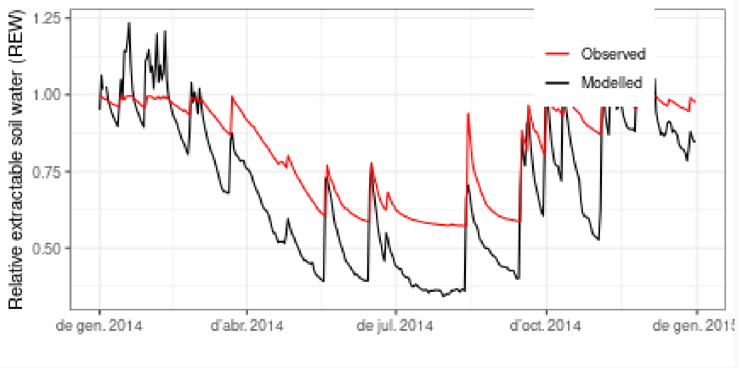
Exercise solution

Step 13. Evaluation of soil moisture content

We repeat the same comparison but after relativizing both series, using `type = "REW"`:

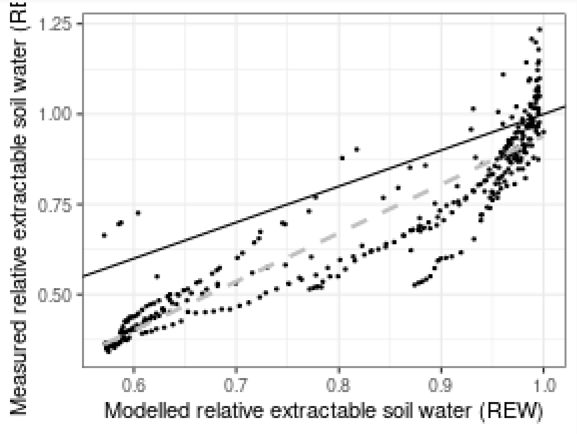
Time series plot

```
evaluation_plot(fb_basic, fb_observed,
    type = "REW", plotType="dynamics")+
  theme(legend.position = c(0.8,0.85))
```



Scatter plot

```
evaluation_plot(fb_basic, fb_observed,
    type = "REW", plotType="scatter")
```



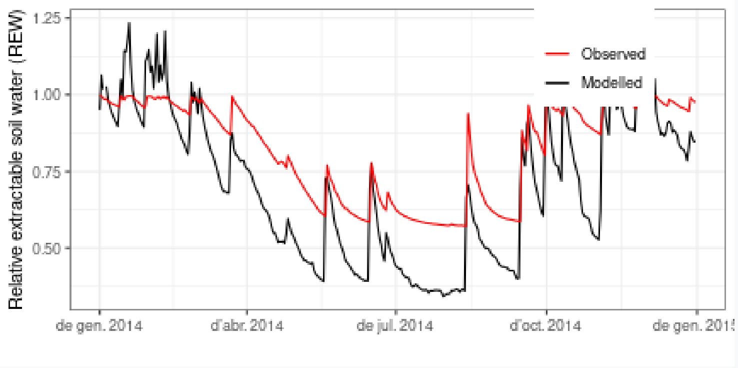
Exercise solution

Step 13. Evaluation of soil moisture content

We repeat the same comparison but after relativizing both series, using `type = "REW"`:

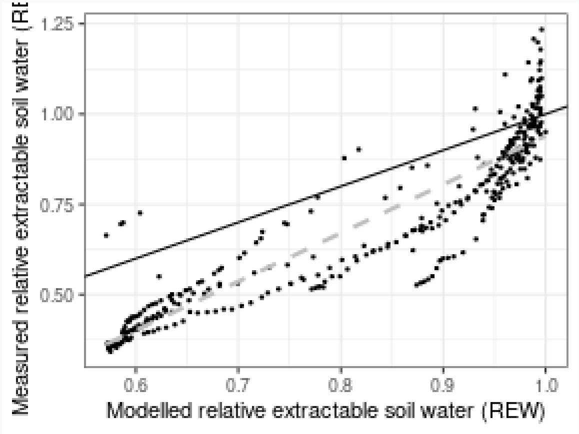
Time series plot

```
evaluation_plot(fb_basic, fb_observed,
    type = "REW", plotType="dynamics")+
  theme(legend.position = c(0.8,0.85))
```



Scatter plot

```
evaluation_plot(fb_basic, fb_observed,
    type = "REW", plotType="scatter")
```




```
evaluation_stats(fb_basic, fb_observed, type = "REW")
```

```
##           n      Bias   Bias.rel       MAE    MAE.rel         r       NSE   NSE.abs
## 364.0000000 0.1216330 17.3433611 0.1429304 20.3801066 0.9195429 0.5225623 0.3151146
```

Exercise solution

Step 14. Advanced water/energy balance

Since we are about to run a advanced water balance simulation, we initialize a simulation control parameter list with `transpirationMode = "Sperry"`, i.e.:

```
fb_control <- defaultControl("Sperry")
```

Exercise solution

Step 14. Advanced water/energy balance

Since we are about to run a advanced water balance simulation, we initialize a simulation control parameter list with `transpirationMode = "Sperry"`, i.e.:

```
fb_control <- defaultControl("Sperry")
```

and assemble our inputs into a `spwbInput` object, using:

```
fb_x2 <- forest2spwbInput(fb_forest, fb_soil, fb_SpParams, fb_control)
```

Exercise solution

Step 14. Advanced water/energy balance

Since we are about to run a advanced water balance simulation, we initialize a simulation control parameter list with `transpirationMode = "Sperry"`, i.e.:

```
fb_control <- defaultControl("Sperry")
```

and assemble our inputs into a `spwbInput` object, using:

```
fb_x2 <- forest2spwbInput(fb_forest, fb_soil, fb_SpParams, fb_control)
```

Finally, we launch the simulation (takes 8 seconds in ver. 2.7.4):

```
fb_adv <- spwb(fb_x2, fb_meteo,  elevation = 420, latitude = 43.24083)
```

Exercise solution

Step 15. Comparing the performance of the two models

To compare the performance of the two models with respect to observed data we can calculate the evaluation statistics for soil moisture:

```
evaluation_stats(fb_basic, fb_observed, type = "REW")
```

```
##            n       Bias   Bias.rel        MAE    MAE.rel          r        NSE    NSE.abs
## 364.0000000  0.1216330 17.3433611  0.1429304 20.3801066  0.9195429  0.5225623  0.3151146
```

```
evaluation_stats(fb_adv, fb_observed, type = "REW")
```

```
##            n        Bias    Bias.rel         MAE     MAE.rel           r         NSE
## 364.00000000  0.06479196  9.23853458  0.09458863 13.48717096  0.92964414  0.78554058
##      NSE.abs
##   0.54675571
```

Exercise solution

Step 15. Comparing the performance of the two models

To compare the performance of the two models with respect to observed data we can calculate the evaluation statistics for soil moisture:

```
evaluation_stats(fb_basic, fb_observed, type = "REW")
```

```
##           n        Bias    Bias.rel         MAE     MAE.rel           r         NSE     NSE.abs
## 364.0000000   0.1216330  17.3433611   0.1429304  20.3801066   0.9195429   0.5225623   0.3151146
```

```
evaluation_stats(fb_adv, fb_observed, type = "REW")
```

```
##            n         Bias     Bias.rel          MAE      MAE.rel            r          NSE
## 364.00000000   0.06479196   9.23853458   0.09458863  13.48717096   0.92964414   0.78554058
##      NSE.abs
##   0.54675571
```

… and for stand evapotranspiration:

```
evaluation_stats(fb_basic, fb_observed, type = "SE+TR")
```

```
##           n        Bias    Bias.rel         MAE     MAE.rel           r         NSE     NSE.abs
## 365.0000000  -0.3296444 -24.7410562   0.4264928  32.0098958   0.7901774   0.3061136   0.1434467
```

```
evaluation_stats(fb_adv, fb_observed, type = "SE+TR")
```

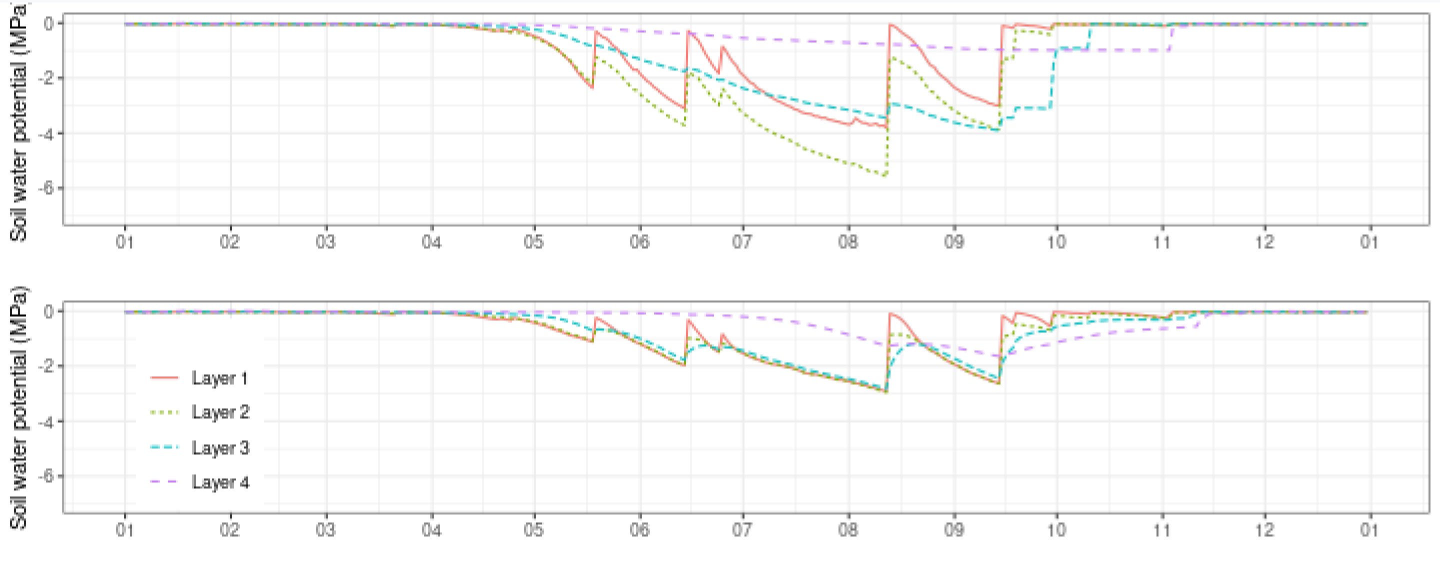
```
##           n        Bias    Bias.rel         MAE     MAE.rel           r         NSE     NSE.abs
## 365.0000000  -0.3117613 -23.3988607   0.4413206  33.1227742   0.7257303   0.1948774   0.1136671
```

Exercise solution

Step 16. Comparing soil moisture dynamics

We can compare soil layer moisture dynamics by drawing soil water potentials:

```
g1<-plot(fb_basic, "SoilPsi", ylim= c(-7,0))+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+theme(legend.position = "none")
g2<-plot(fb_adv, "SoilPsi", ylim= c(-7,0))+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+theme(legend.position = c(0.1,0.47))
plot_grid(g1,g2, ncol=1)
```

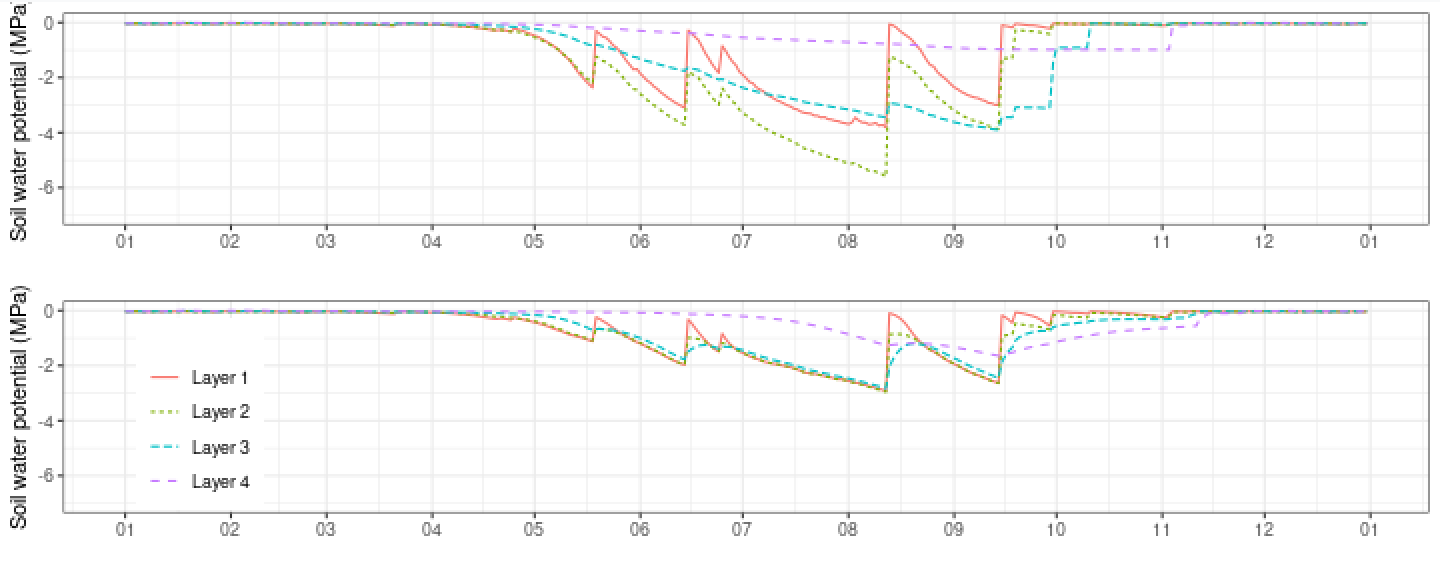


Exercise solution

Step 16. Comparing soil moisture dynamics

We can compare soil layer moisture dynamics by drawing soil water potentials:

```
g1<-plot(fb_basic, "SoilPsi", ylim= c(-7,0))+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+theme(legend.position = "none")
g2<-plot(fb_adv, "SoilPsi", ylim= c(-7,0))+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+theme(legend.position = c(0.1,0.47))
plot_grid(g1,g2, ncol=1)
```



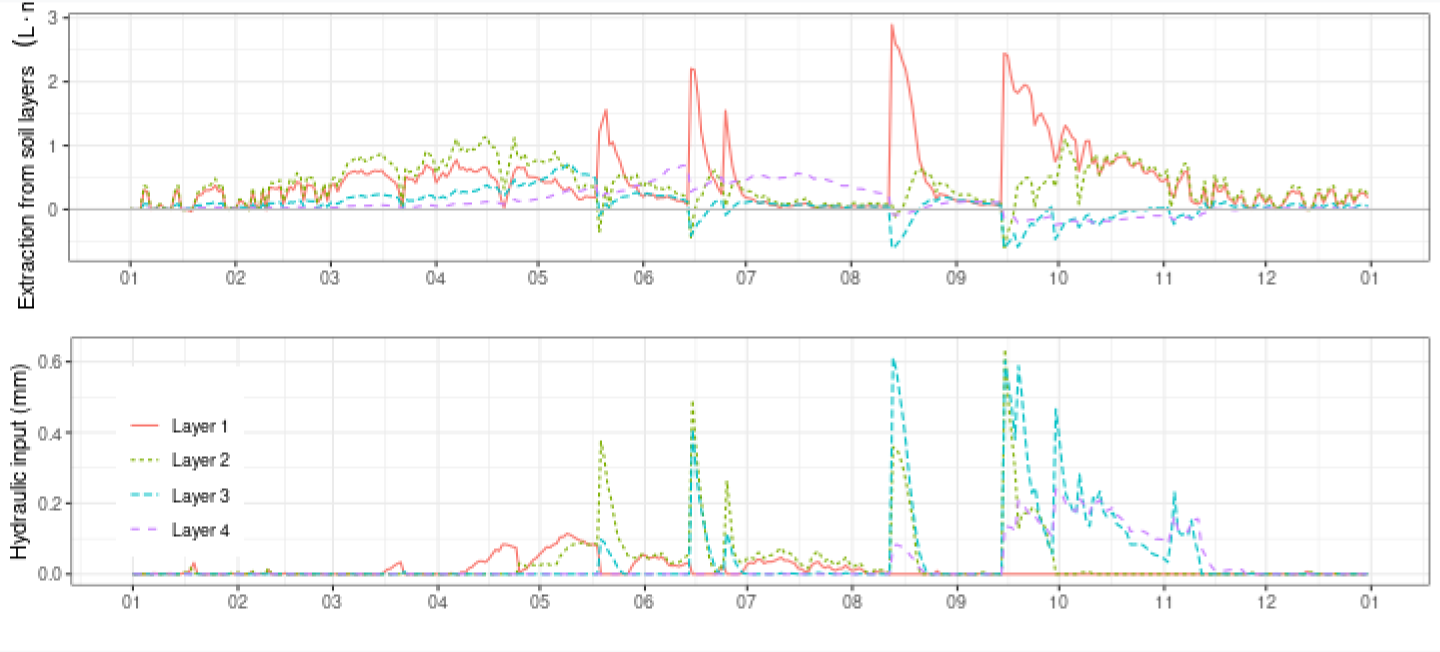
The basic model dries the soil more than the advanced model, which produces a stronger coupling between soil layers because of hydraulic redistribution.

Exercise solution

Step 17. Understanding extraction and hydraulic redistribution

The following shows the daily root water uptake (or release) from different soil layers, and the daily amount of water entering soil layers due to hydraulic redistribution:

```
g1<-plot(fb_adv, "PlantExtraction")+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+theme(legend.position = "none")
g2<-plot(fb_adv, "HydraulicRedistribution")+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+theme(legend.position = c(0.08,0.5))
plot_grid(g1, g2, rel_heights = c(0.8,0.8), ncol=1)
```

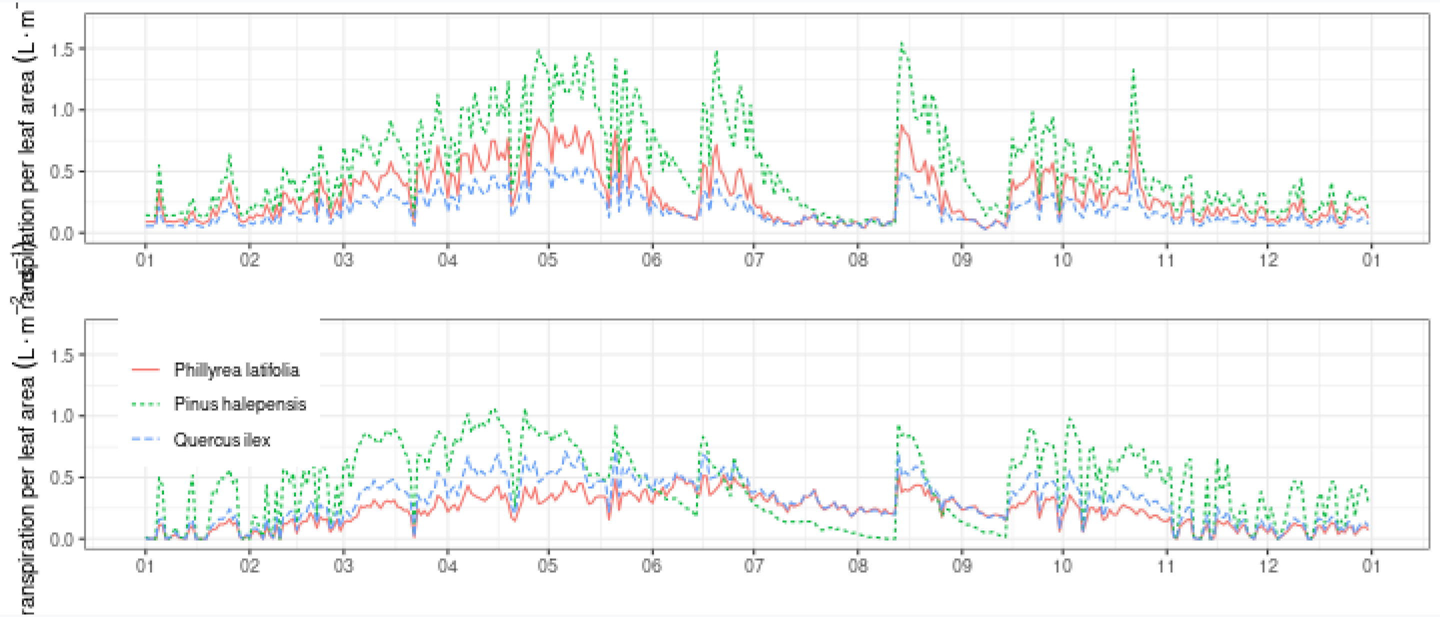


Exercise solution

Step 18. Comparing leaf-level transpiration dynamics

We can display the transpiration per leaf area unit basis using `"TranspirationPerLeaf"`.

```
g1<-plot(fb_basic, "TranspirationPerLeaf", bySpecies = TRUE, ylim = c(0,1.7))+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+theme(legend.position = "none")
g2<-plot(fb_adv, "TranspirationPerLeaf", bySpecies = TRUE, ylim = c(0,1.7))+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+theme(legend.position = c(0.1,0.7))
plot_grid(g1,g2, ncol=1)
```

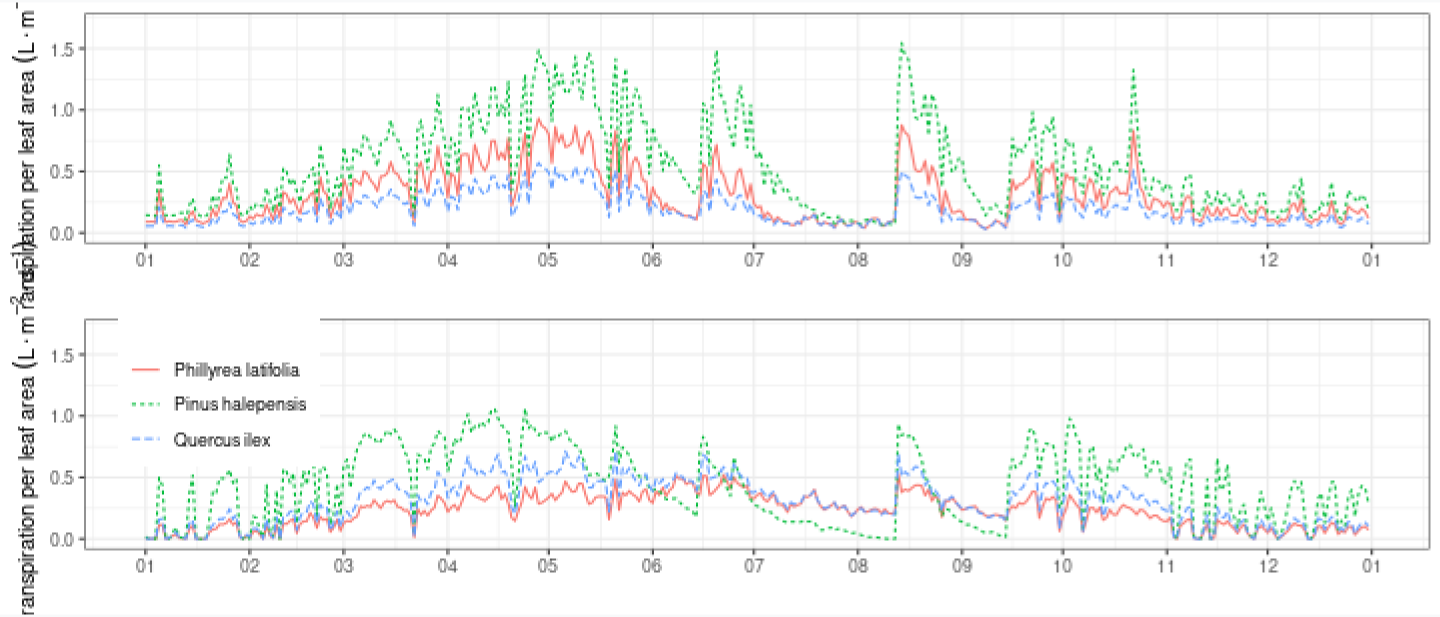


Exercise solution

Step 18. Comparing leaf-level transpiration dynamics

We can display the transpiration per leaf area unit basis using `"TranspirationPerLeaf"`.

```
g1<-plot(fb_basic, "TranspirationPerLeaf", bySpecies = TRUE, ylim = c(0,1.7))+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+theme(legend.position = "none")
g2<-plot(fb_adv, "TranspirationPerLeaf", bySpecies = TRUE, ylim = c(0,1.7))+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+theme(legend.position = c(0.1,0.7))
plot_grid(g1,g2, ncol=1)
```



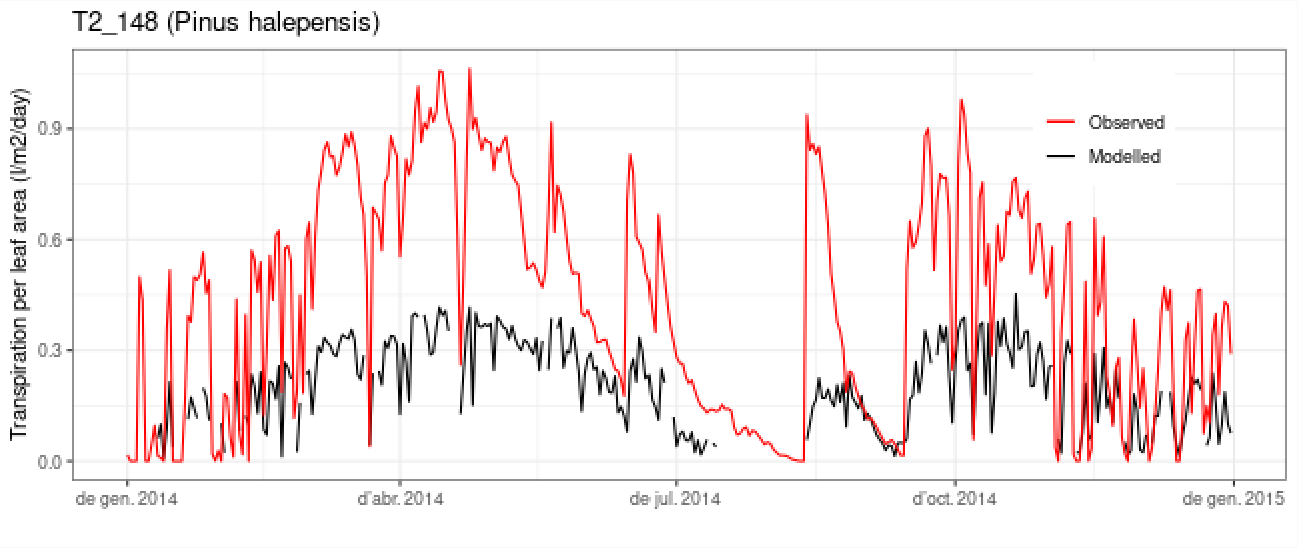
The basic model produces higher transpiration than the advanced model.

Exercise solution

Step 19. Evaluation of tree transpiration

The following displays the observed and predicted transpiration for *Pinus halepensis* …

```
evaluation_plot(fb_adv, fb_observed, cohort = "T2_148", type="E", plotType = "dynamics")+
  theme(legend.position = c(0.85,0.83))
```




```
evaluation_stats(fb_adv, fb_observed, cohort = "T2_148", type="E")
```

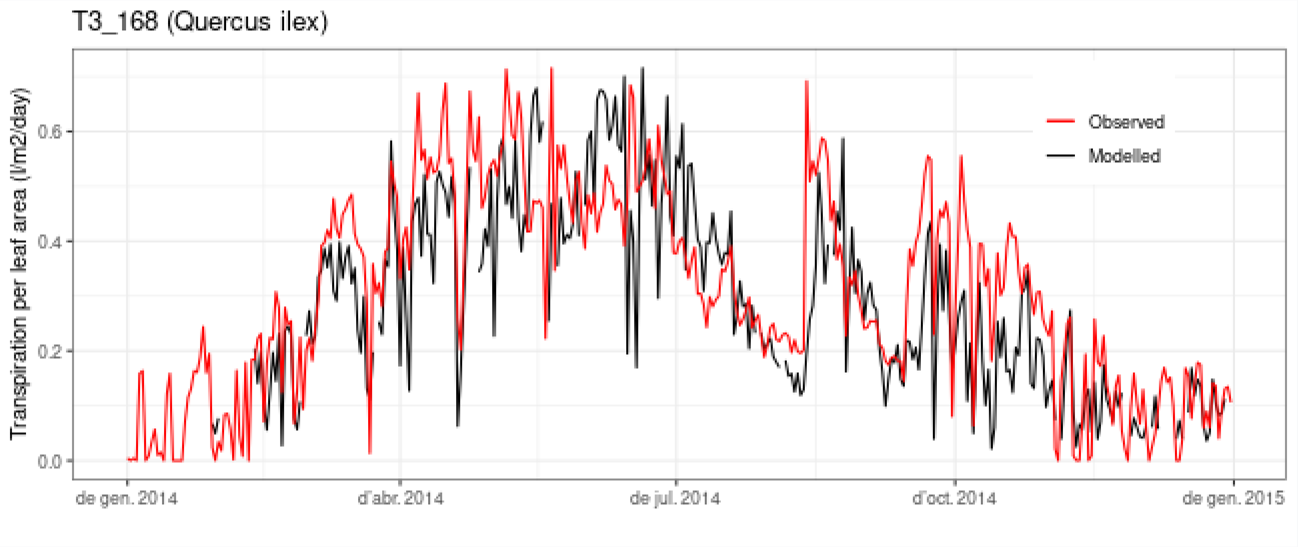
```
##            n        Bias     Bias.rel         MAE      MAE.rel           r         NSE     NSE.abs
## 300.0000000   0.2801236 136.1994298   0.2871963 139.6382323   0.8308882  -8.1053801  -1.9022659
```

Exercise solution

Step 19. Evaluation of tree transpiration

… and *Quercus ilex*:

```
evaluation_plot(fb_adv, fb_observed, cohort = "T3_168", type="E", plotType = "dynamics")+
  theme(legend.position = c(0.85,0.83))
```




```
evaluation_stats(fb_adv, fb_observed, cohort = "T3_168", type="E")
```

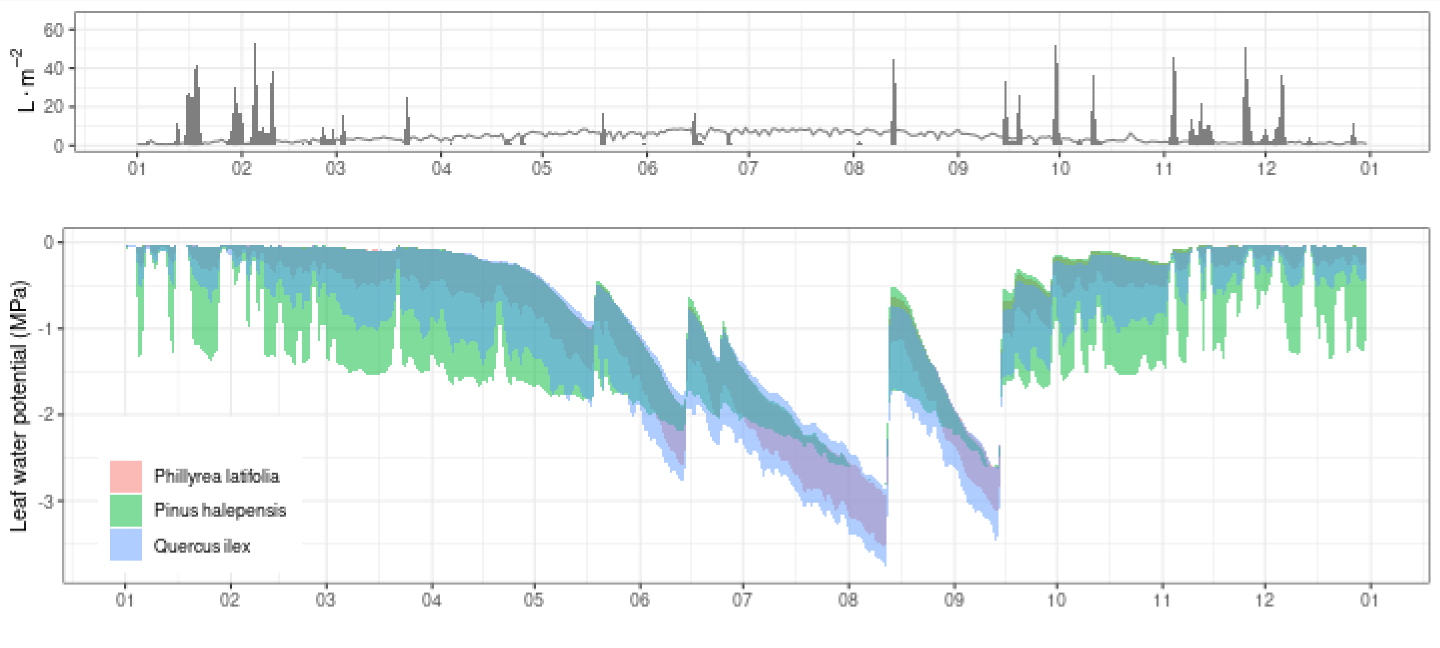
```
##             n         Bias     Bias.rel          MAE      MAE.rel            r          NSE
## 309.00000000   0.04554656  15.73508383   0.09663686  33.38538163   0.76931949   0.46800110
##       NSE.abs
##    0.34163336
```

Exercise solution

Step 20. Examining leaf water potentials

The following plots illustrate that the model simulates a tighter stomatal control for *Pinus halepensis*.

```
g1<-plot(fb_adv)+scale_x_date(date_breaks = "1 month", date_labels = "%m")+
  theme(legend.position = "none")
g2<-plot(fb_adv, "LeafPsiRange", bySpecies = TRUE)+
  scale_x_date(date_breaks = "1 month", date_labels = "%m")+
  theme(legend.position = c(0.1,0.25)) + ylab("Leaf water potential (MPa)")
plot_grid(g1, g2, ncol=1, rel_heights = c(0.4,0.8))
```

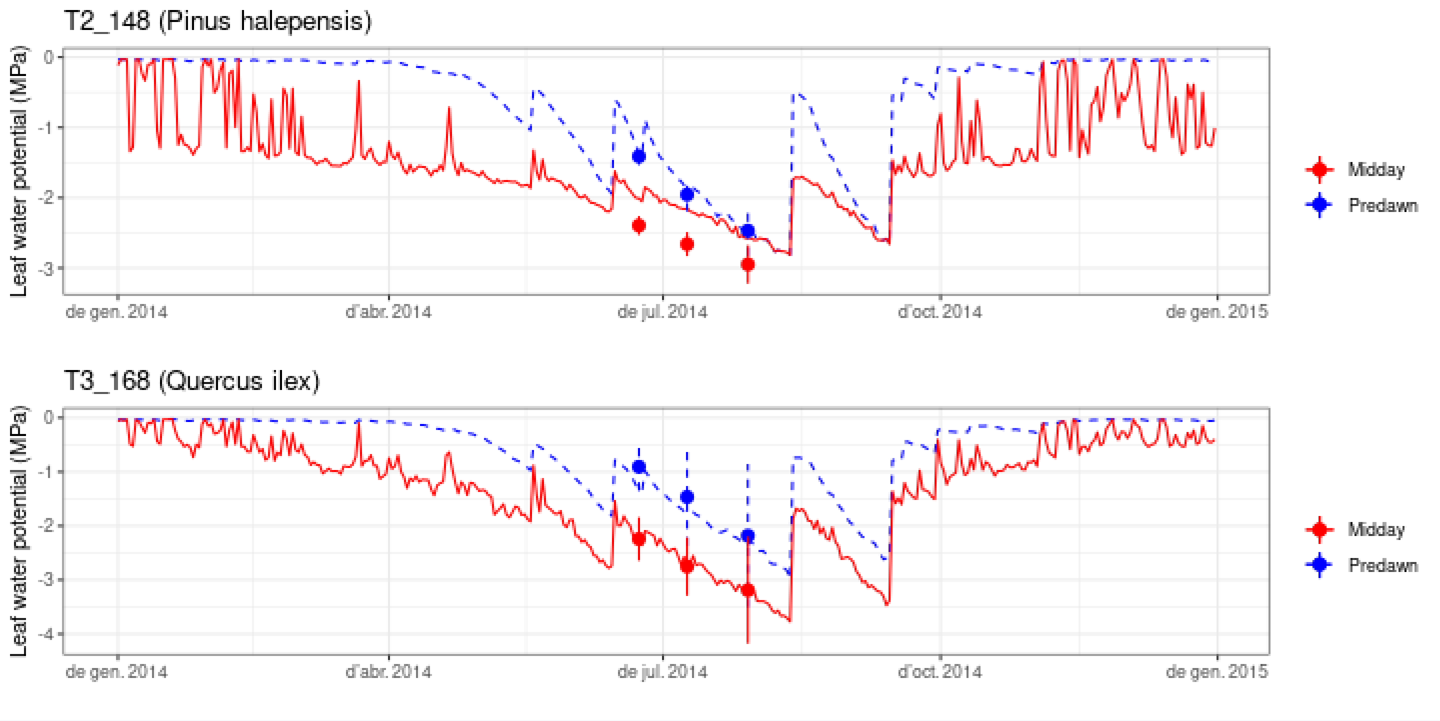


Exercise solution

Step 21. Comparing leaf water potentials with observations

If we compare leaf water potentials against observations (`type = "WP"`) we obtain a rather good performance for *Q. ilex*, but midday water potentials are less well approximated for *P. halepensis*.

```
g1<-evaluation_plot(fb_adv, fb_observed, cohort = "T2_148", type="WP", plotType = "dynamics")
g2<-evaluation_plot(fb_adv, fb_observed, cohort = "T3_168", type="WP", plotType = "dynamics")
plot_grid(g1, g2, ncol=1)
```

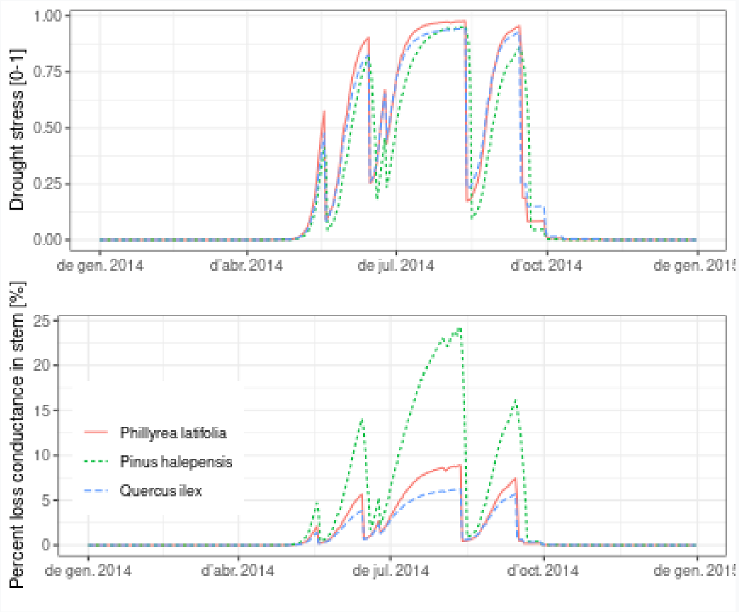


Exercise solution

Steps 22-23. Drought stress and PLC

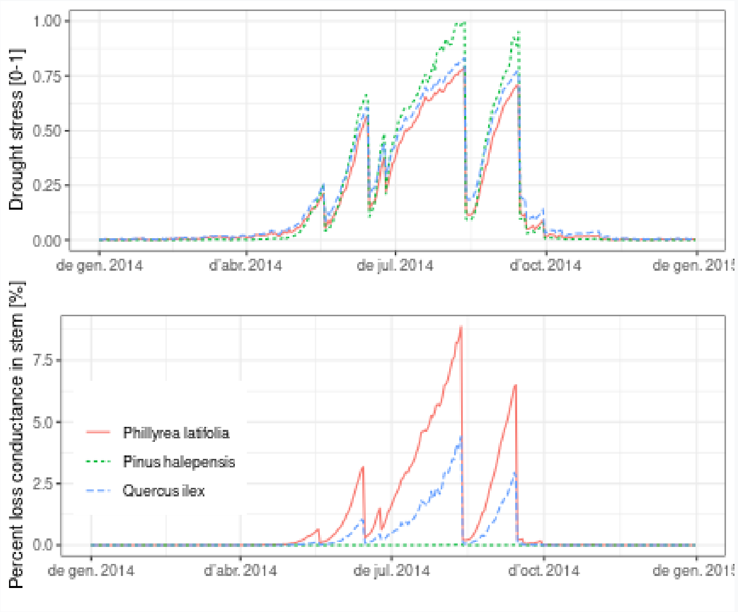
Basic model

```
g1<-plot(fb_basic, "PlantStress", bySpecies =
  theme(legend.position = "none")
g2<-plot(fb_basic, "StemPLC", bySpecies = TRU
  theme(legend.position = c(0.15,0.45))
plot_grid(g1, g2, ncol=1)
```



Advanced model

```
g3<-plot(fb_adv, "PlantStress", bySpecies = T
  theme(legend.position = "none")
g4<-plot(fb_adv, "StemPLC", bySpecies = TRUE)
  theme(legend.position = c(0.15,0.45))
plot_grid(g3, g4, ncol=1)
```



Exercise solution

Steps 22-23. Drought stress and PLC

The basic model seems to overestimate PLC for *Pinus halepensis*, compared to the advanced model.

Exercise solution

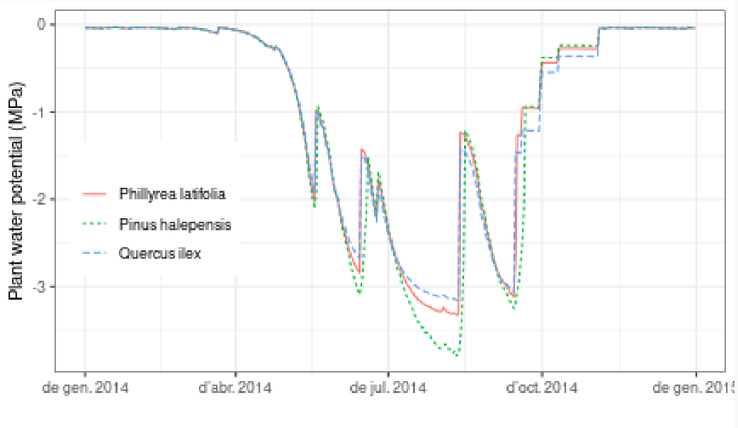
Steps 22-23. Drought stress and PLC

The basic model seems to overestimate PLC for *Pinus halepensis*, compared to the advanced model.

This could arise from a difference in the parameters determining PLC or differences in the water potential simulated by both models. We examine the first option using:

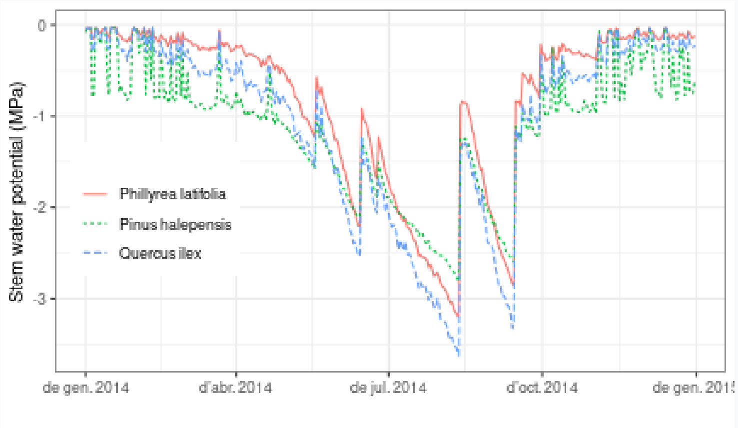
Basic model

```
plot(fb_basic, "PlantPsi", bySpecies = TRUE)+
  theme(legend.position = c(0.15,0.45))
```



Advanced model

```
plot(fb_adv, "StemPsi", bySpecies = TRUE)+
  theme(legend.position = c(0.15,0.45))
```



Exercise solution

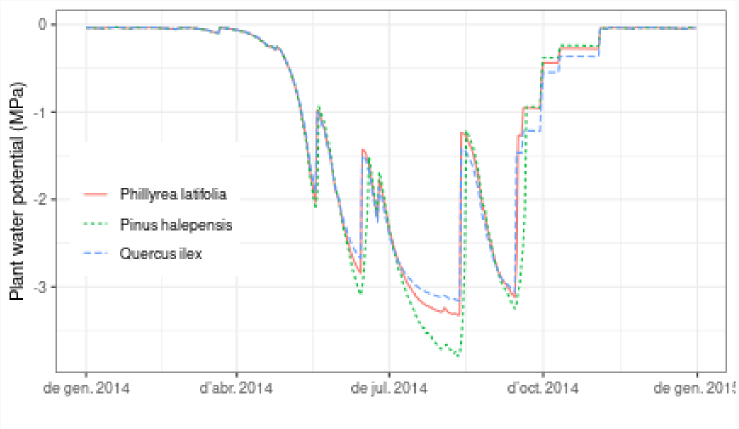
Steps 22-23. Drought stress and PLC

The basic model seems to overestimate PLC for *Pinus halepensis*, compared to the advanced model.

This could arise from a difference in the parameters determining PLC or differences in the water potential simulated by both models. We examine the first option using:

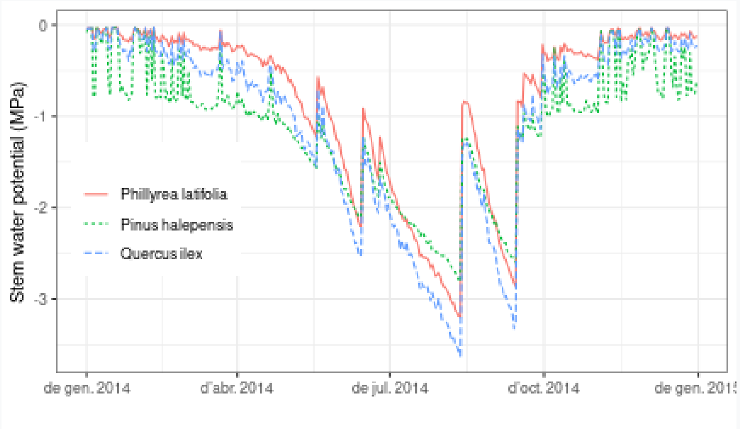
Basic model

```
plot(fb_basic, "PlantPsi", bySpecies = TRUE)+
  theme(legend.position = c(0.15,0.45))
```



Advanced model

```
plot(fb_adv, "StemPsi", bySpecies = TRUE)+
  theme(legend.position = c(0.15,0.45))
```



The basic model predicts much lower *plant* water potentials than the advanced model, probably as a result of lacking the process of hydraulic redistribution.

M.C. Escher - Relativity, 1953