CIVE 625 - Lab 1

Mike Talbot

2024-02-26

Setup

```r
# load model forcing data and add empty columns
data <- read_csv("Lab 1 data.csv", show_col_types=F) %>%
  na.omit() %>%
  mutate(T_K = 0,
         r_a = 0,
         e = 0,
         e_sat = 0,
         D = 0,
         g_R = 0,
         g_D = 0,
         g_T = 0,
         g_M = 0,
         g_S = 0,
         r_s = 0,
         L_u = 0,
         R_n = 0,
         lambda = 0,
         delta = 0,
         gamma = 0,
         C_interim = 0,
         C_actual = 0,
         C_over_S = 0,
         lambdaE_I = 0,
         lambdaE_T = 0,
         lambdaE = 0,
         D_canopy = 0,
         C_final = 0,
         H = 0,
         T_surf_K = 0,
         SM_last = 0,
         SM_new = 0)

# timestep
dt <- data$t[2] - data$t[1] # hours
```

Forest

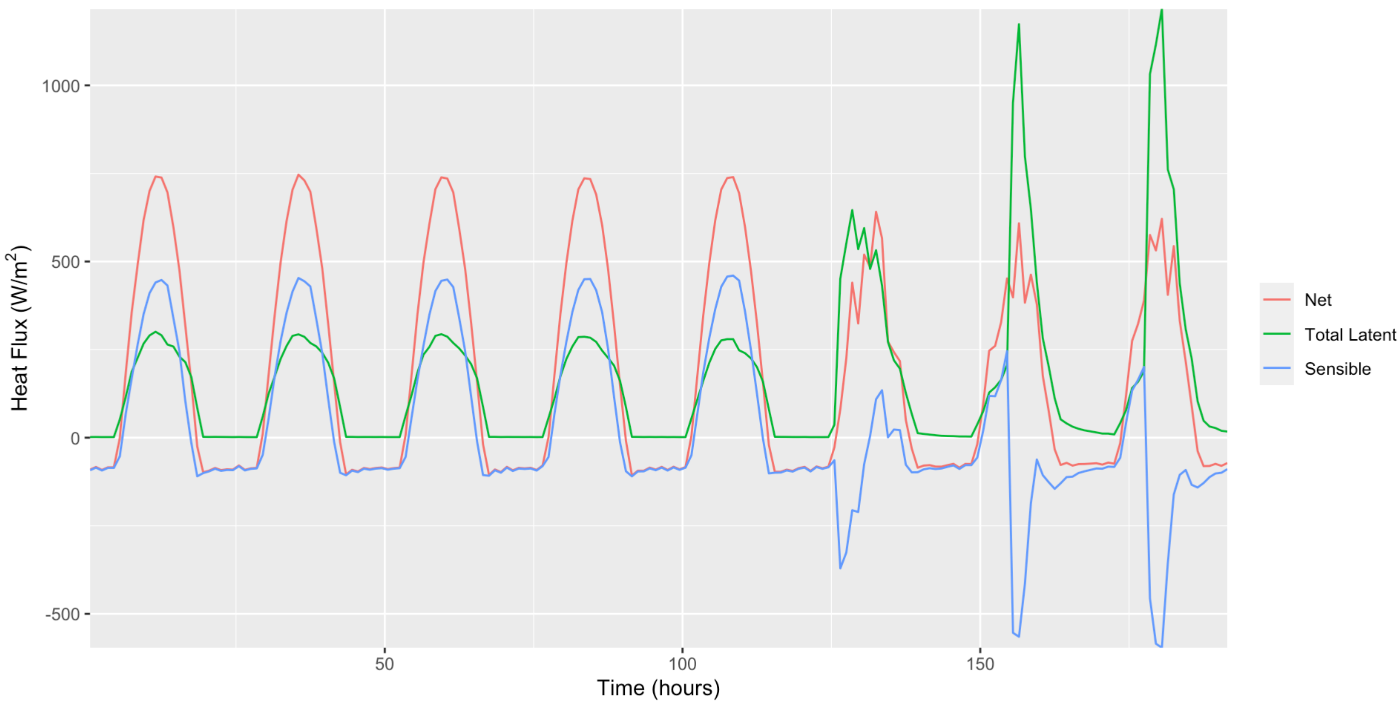
```r
crop <- "forest" # either "forest" or "grass"

# prescribed crop-specific values
LAI <- ifelse(crop == "forest", 4, 2) # leaf area index
h <- ifelse(crop == "forest", 20, 0.12) # m
alpha <- ifelse(crop == "forest", 0.12, 0.23) # surface albedo
g_0 <- ifelse(crop == "forest", 15, 30) / 1000 # m s^-1
SM_0 <- ifelse(crop == "forest", 80, 40) # maximum soil moisture holding capacity, mm
SM_init <- ifelse(crop == "forest", 40, 20) # initial soil moisture, mm
SM_max <- ifelse(crop == "forest", 80, 40) # maximum soil moisture available, mm
S <- ifelse(crop == "forest", 4, 2) # canopy capacity, mm
z_m <- ifelse(crop == "forest", 22, 2) # m
K_M_1 <- ifelse(crop == "forest", 3.36E-4, 1.87E-2) # unitless

# calculated
d <-  1.1 * h * log(1 + (LAI / 5) ^ 0.25) # m, equation 22.2
z_0 <- 0.3 * h * (1 - d/h) # m, equation 22.4

# prescribed crop-independent values
P <- 101.2 # kPa
k <- 0.4 # Von Kaman constant, unitless
epsilon_surf <- 0.95 # surface emissivity, unitless
sigma <- 5.67E-8 # W m-2 K-4
K_R <- 200 # W m-2
K_D_1 <- -0.307 # kPa-1
K_D_2 <- 0.019 # kPa-2
T_L <- 273 # K
T_0 <- 293 # K
T_H <- 313 # K
K_M_2 <- -0.1 # mm-1
rho_a <- 1.23 # kg m^-3
c_p <- 1013  # J kg^-1 K^-1
g_c <- 1 # unitless
alpha_T <- 1 # unitless
```

```
for (t in seq_range(data$t, by=dt)) {
  # extract inputs at time t
  u_m = data$u_m[data$t == t] # windspeed, m s-1
  q = data$q[data$t == t] / 1000 # specific humidity, converted to kg kg-1
  T_C = data$T_C[data$t == t] # measured temperature, C
  T_K = T_C + 273.17 # measured temperature, K
  R_s = data$R_s[data$t == t] # solar radiation, W m-2
  L_d = data$L_d[data$t == t] # downward longwave radiation, W m-2
  R = data$R[data$t == t] # precip, mm

  if (t == min(data$t)) {
    # initialize soil moisture
    SM_last = SM_init

    # initialize surface temperature as measured temperature
    T_surf_K = T_K
  } else {
    # set soil moisture to that from the previous timestep
    SM_last = data$SM_new[data$t == (t - dt)]

    # set surface temperature to that from the previous timestep
    T_surf_K = data$T_surf_K[data$t == (t - dt)]
  }

  # calculate surface resistance variables at time t
  r_a = (1 / (k ^ 2 * u_m)) * log((z_m - d) / z_0) * log((z_m - d) / (z_0 / 10)) # aerod
ynamic resistance, s m-1, eq. 22.9
  e = q * P / 0.622 # vapor pressure, kPa, eq. 2.9
  e_sat = 0.6108 * exp(17.27 * T_C / (237.3 + T_C)) # saturated vapor pressure at temper
ature T, kPa, eq. 2.20
  D = e_sat - e # vapor pressure deficit, kPa, eq. 2.17
  g_R = max(R_s * (1000 + K_R) / (1000 * (R_s + K_R)), 0) # eq. 24.2
  g_D = max(1 + K_D_1 * D + K_D_2 * D ^ 2, 0) # eq. 24.3
  g_T = max((T_K - T_L) * (T_H - T_K) ^ alpha_T / ((T_0 - T_L) * (T_H - T_0) ^ alpha_T),
0) # eq. 24.4
  g_M = max(1 - K_M_1 * exp(K_M_2 * (SM_last - SM_0)) , 0) # eq. 24.6
  g_S = max(g_0 * g_c * g_R * g_D * g_T * g_M, 0.0001) # eq. 24.1
  r_s = 1 / g_S # surface resistance, eq. 24.1

  # calculate radiation variables at time t
  if (t == min(data$t)) {
    # use initialized surface temp for first timestep
    L_u = -epsilon_surf * sigma * T_surf_K ^ 4
  } else if (t == (min(data$t) + dt)) {
    # use surface temp from previous timestep
    mean_T = data$T_surf_K[data$t == (t - dt)]
    L_u = -epsilon_surf * sigma * mean_T ^ 4
  } else {
    # use surface temps from previous two timesteps
    mean_T = mean(data$T_surf_K[data$t == (t - dt)], data$T_surf_K[data$t == (t - 2*d
t)])
    L_u = -epsilon_surf * sigma * mean_T ^ 4
```

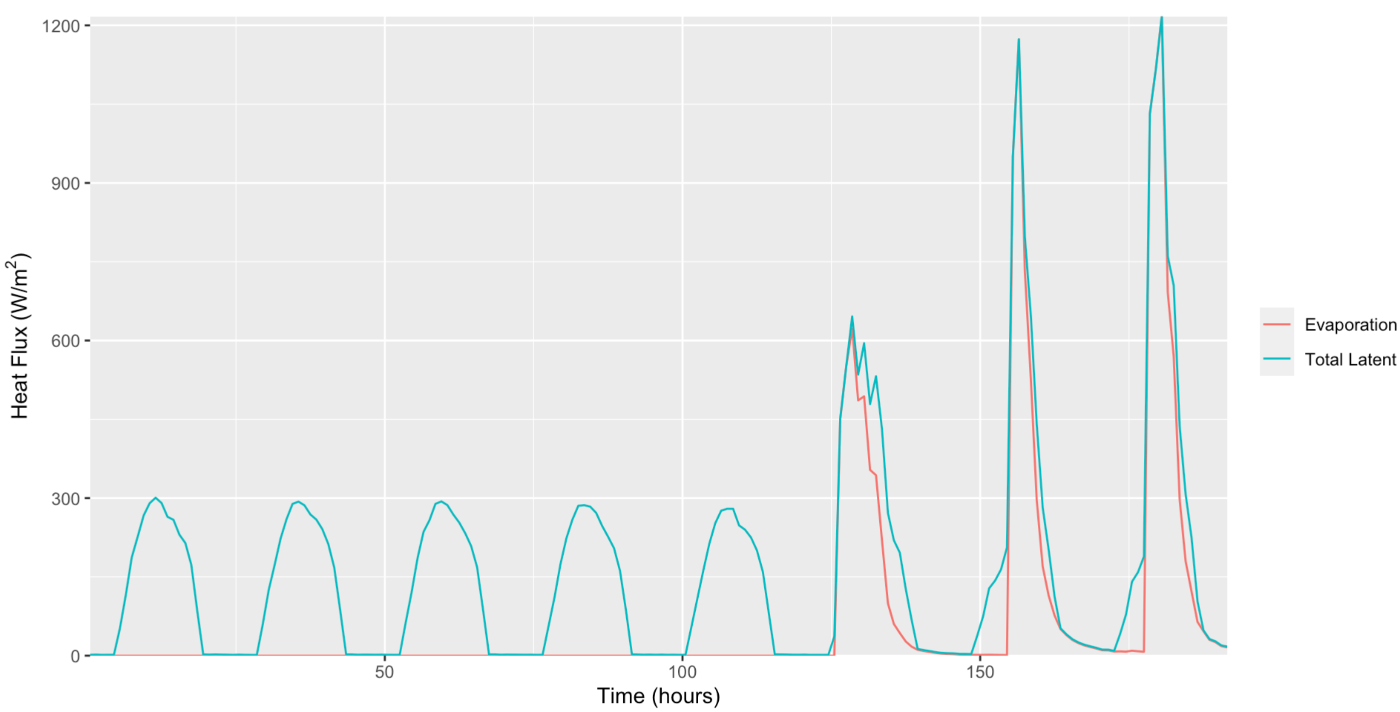
```
  }

  R_n = R_s * (1 - alpha) + L_d + L_u # incoming minus outgoing

  # calculate canopy and soil water balance variables at time t
  lambda = (2.501 - 0.002361 * T_C) * 1E6 # latent heat of vaporization, J kg-1, eq.
  delta = 4098 * e_sat / (237.7 + T_C) ^ 2 # slope of the saturation vapor pressure curve, kPa K-1
  gamma = c_p * P / (0.622 * lambda) # psychrometric constant, kPa K-1

  if (t == min(data$t)) {
    C_interim = 0
  } else {
    C_interim = R + data$C_final[data$t == (t - dt)] # mm
  }

  if (C_interim <= S) {
    C_actual = C_interim
    D_canopy = 0
  } else {
    C_actual = S
    D_canopy = C_interim - S
  }

  C_over_S = C_actual / S

  A = R_n # available energy (assumption)

  lambdaE_I = (C_over_S) * max((delta * A + rho_a * c_p * D / r_a) / (delta + gamma), 0) # W m-2, eq. 22.14
  lambdaE_T = max((delta * A + rho_a * c_p * D / r_a) / (delta + gamma * (1 + r_s / r_a)), 0) # W m-2, eq. 22.18
  lambdaE = lambdaE_I + (1 - C_over_S) * lambdaE_T # eq. 22.17 (corrected)
  H = R_n - lambdaE
  C_final = max(C_actual - lambdaE_I * (dt * 3600 / (lambda * rho_a)), 0) # convert lambdaE_I to mm
  SM_new = max(min(SM_last + D_canopy - lambdaE_T * (dt * 3600 / (lambda * rho_a)), SM_max), 0) # convert lambdaE_T to mm

  T_surf_K = T_K + H * r_a / (rho_a * c_p)

  # update values for time t
  data$T_K[data$t == t] = T_K # K
  data$r_a[data$t == t] = r_a # s m-1
  data$e[data$t == t] = e # kPa
  data$e_sat[data$t == t] = e_sat # kPa
  data$D[data$t == t] = D
  data$g_R[data$t == t] = g_R
  data$g_D[data$t == t] = g_D
  data$g_T[data$t == t] = g_T
  data$g_M[data$t == t] = g_M
  data$g_S[data$t == t] = g_S * 1000 # mm s-1
```

```
  data$r_s[data$t == t] = r_s
  data$L_u[data$t == t] = L_u
  data$R_n[data$t == t] = R_n
  data$lambda[data$t == t] = lambda # J kg-1
  data$delta[data$t == t] = delta # kPa K-1
  data$gamma[data$t == t] = gamma # kPa K-1
  data$C_interim[data$t == t] = C_interim # mm
  data$C_actual[data$t == t] = C_actual # mm
  data$C_final[data$t == t] = C_final # mm
  data$C_over_S[data$t == t] = C_over_S
  data$D_canopy[data$t == t] = D_canopy # mm
  data$lambdaE_I[data$t == t] = lambdaE_I # W m-2
  data$lambdaE_T[data$t == t] = lambdaE_T # W m-2
  data$lambdaE[data$t == t] = lambdaE # mm
  data$H[data$t == t] = H # W m-2
  data$T_surf_K[data$t == t] = T_surf_K # K
  data$SM_last[data$t == t] = SM_last # mm
  data$SM_new[data$t == t] = SM_new # mm

}

variable_levels <- c("u_m", "q", "R_s", "L_d", "R", "T_K",
                     "r_a", "e", "e_sat", "D", "g_R", "g_D", "g_T", "g_M", "g_S", "SM_la
st", "r_s",
                     "L_u", "R_n", "lambda", "delta", "gamma", "C_interim", "C_actual",
"C_final", "C_over_S",
                     "D_canopy", "lambdaE_I", "lambdaE_T", "lambdaE", "H", "T_surf_K",
"SM_new")

data_longer <- data %>%
  pivot_longer(cols=c(-t)) %>%
  filter(name %in% variable_levels) %>%
  mutate(name=factor(name, levels=variable_levels))
```

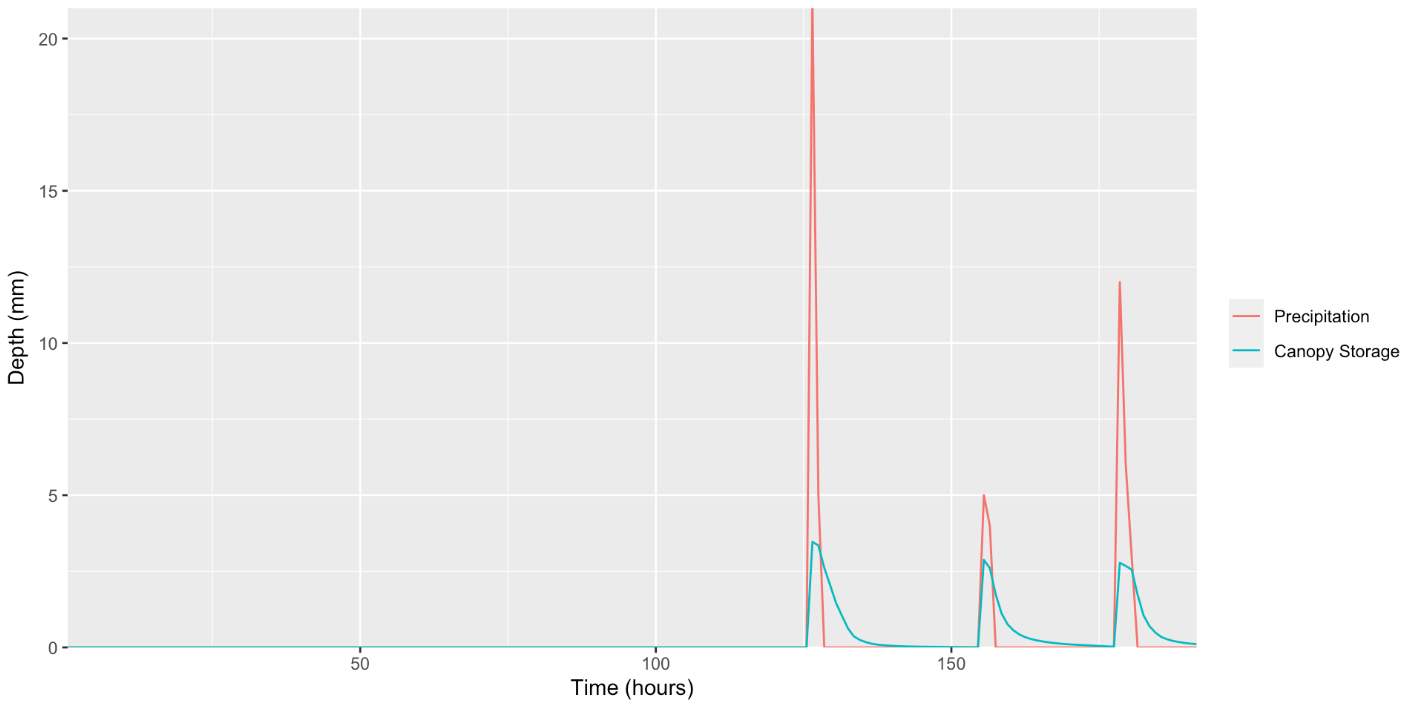
```
# I. Hourly values of the calculated net radiation, total latent heat flux, and sensible
heat flux (all in W m-2) on the same graph
ggplot(data_longer %>% filter(name %in% c("R_n", "lambdaE", "H")), aes(x=t, y=value, col
=name)) +
  geom_path() +
  scale_color_discrete(name=NULL, labels=c("Net", "Total Latent", "Sensible")) +
  scale_y_continuous(name=expression(paste("Heat Flux ", "(W/m"^2, ")")), expand=c(0,0))
+
  scale_x_continuous(name="Time (hours)", expand=c(0,0))
```



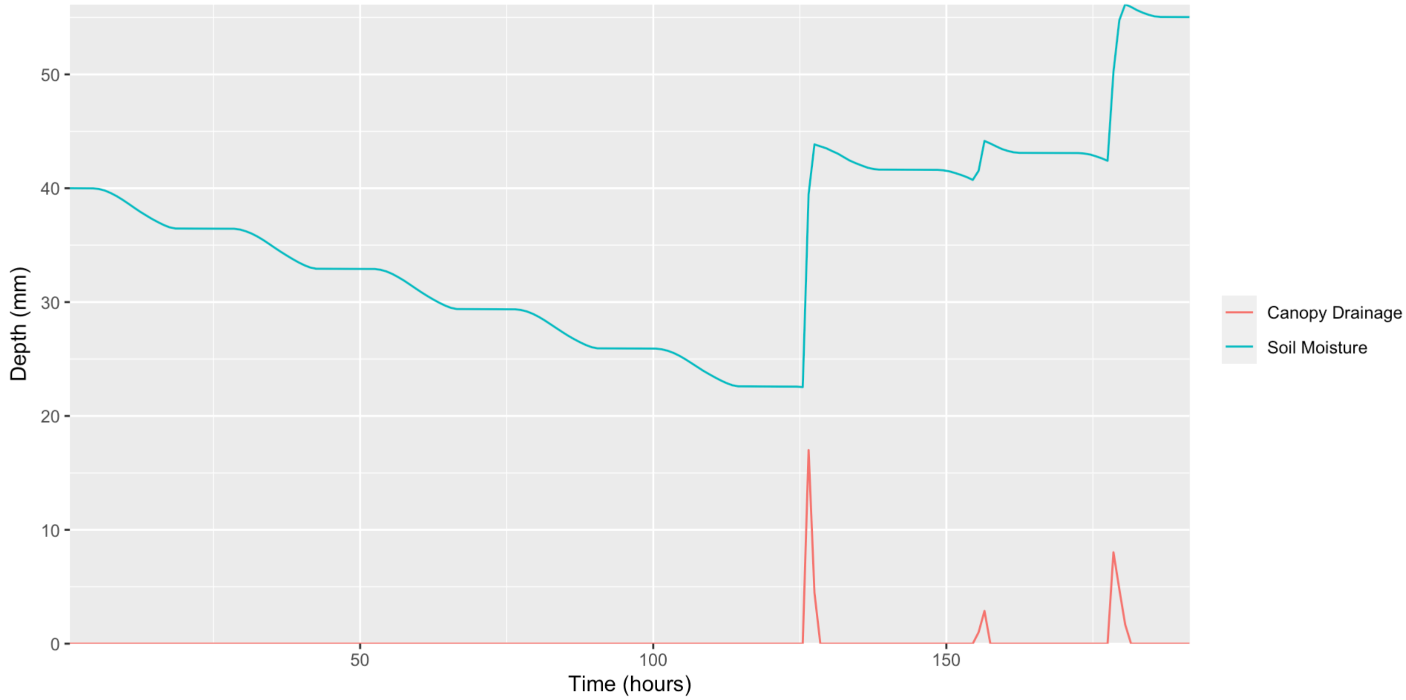

```
# II. Hourly values of the calculated total latent heat flux and the portion of the late
nt heat flux that originates from the evaporation of intercepted water (both in W m-2)
ggplot(data_longer %>% filter(name %in% c("lambdaE_I", "lambdaE")), aes(x=t, y=value, co
l=name)) +
  geom_path() +
  scale_color_discrete(name=NULL, labels=c("Evaporation", "Total Latent")) +
  scale_y_continuous(name=expression(paste("Heat Flux ", "(W/m"^2, ")")), expand=c(0,0))
+
  scale_x_continuous(name="Time (hours)", expand=c(0,0))
```



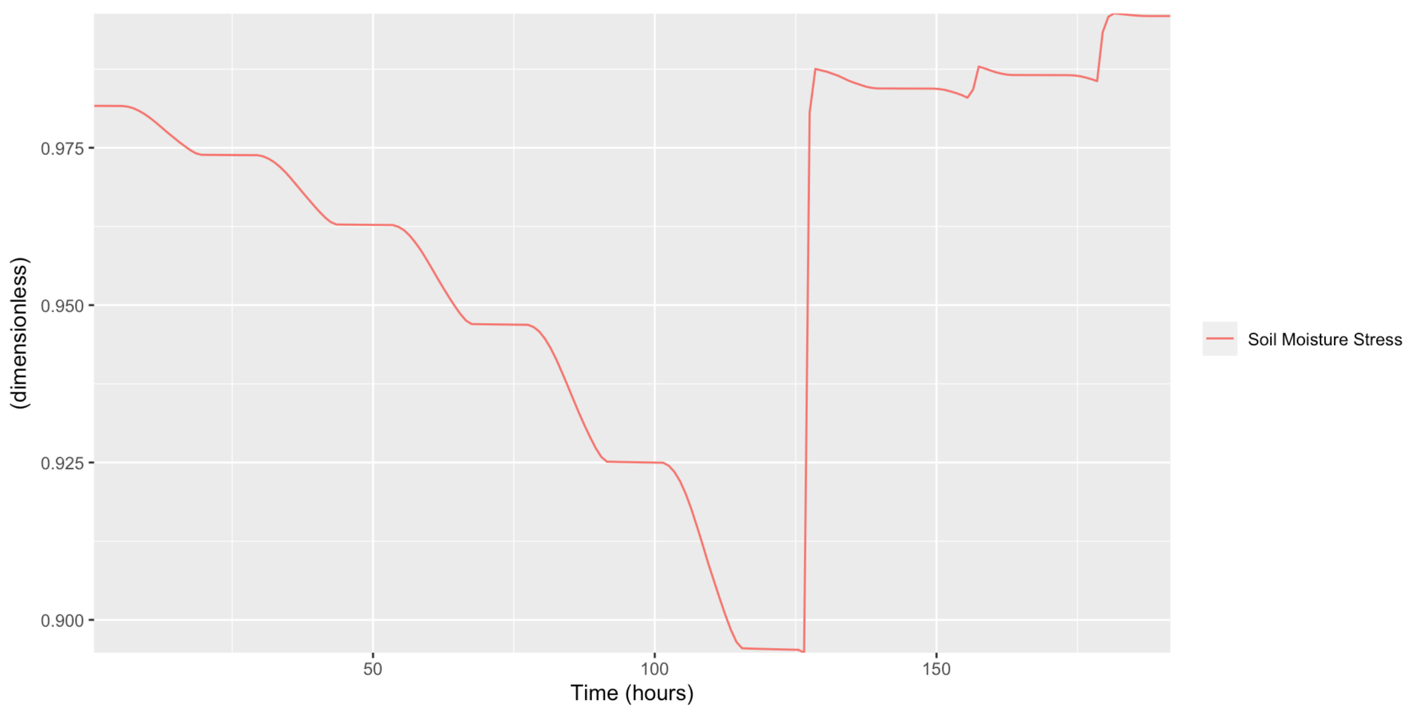
```
# III. Hourly values of the precipitation and the canopy storage C (both in mm)
ggplot(data_longer %>% filter(name %in% c("R", "C_final")), aes(x=t, y=value, col=name))
+
  geom_path() +
  scale_color_discrete(name=NULL, labels=c("Precipitation", "Canopy Storage")) +
  scale_y_continuous(name=expression(paste("Depth ", "(mm)")), expand=c(0,0)) +
  scale_x_continuous(name="Time (hours)", expand=c(0,0))
```




```
# IV. Hourly values of the canopy drainage [if interception was considered] the value of
SM (both in mm)
ggplot(data_longer %>% filter(name %in% c("D_canopy", "SM_new")), aes(x=t, y=value, col=
name)) +
  geom_path() +
  scale_color_discrete(name=NULL, labels=c("Canopy Drainage", "Soil Moisture")) +
  scale_y_continuous(name=expression(paste("Depth ", "(mm)")), expand=c(0,0)) +
  scale_x_continuous(name="Time (hours)", expand=c(0,0))
```




```
# V. Hourly values of the soil moisture stress function gM (no units)
ggplot(data_longer %>% filter(name %in% c("g_M")), aes(x=t, y=value, col=name)) +
  geom_path() +
  scale_color_discrete(name=NULL, labels=c("Soil Moisture Stress")) +
  scale_y_continuous(name=expression(paste("(dimensionless)")), expand=c(0,0)) +
  scale_x_continuous(name="Time (hours)", expand=c(0,0))
```



```r
# VI. A table giving for all 8 individual days the daily average values of  Rn, lambdaE,
H, and lambdaE_I (all in W m-2), and the values of daily average Bowen Ratio and the rat
io of daily average fractional contribution to daily average total evaporation that aris
es from the evaporation of intercepted water
data_daily <- data_longer %>%
  mutate(day = ((t-0.5) %/% 24 + 1)) %>%
  group_by(day, name) %>%
  summarize(value=mean(value)) %>%
  pivot_wider(id_cols=c("day")) %>%
  select(day, R_n, lambdaE, H, lambdaE_I, lambdaE_T) %>%
  mutate(bowen=H/lambdaE,
         evap_over_latent=lambdaE_I/lambdaE)

print(data_daily)
```

```
## # A tibble: 8 × 8
## # Groups:   day [8]
##     day   R_n lambdaE      H lambdaE_I lambdaE_T  bowen evap_over_latent
##   <dbl> <dbl>   <dbl>  <dbl>     <dbl>     <dbl>  <dbl>            <dbl>
## 1     1  215.    124.   90.9        0      124.   0.733            0
## 2     2  215.    124.   91.7        0      124.   0.743            0
## 3     3  215.    124.   91.0        0      124.   0.735            0
## 4     4  215.    121.   94.0        0      121.   0.779            0
## 5     5  215.    117.   98.2        0      117.   0.842            0
## 6     6  132.    216.  -84.3      159.      85.4 -0.390            0.733
## 7     7  127.    231. -104.       176.      84.5 -0.451            0.761
## 8     8  151.    280. -129.       230.      89.4 -0.462            0.822
```

```r
data_longer_forest <- data_longer
data_daily_forest <- data_daily
```

Grass

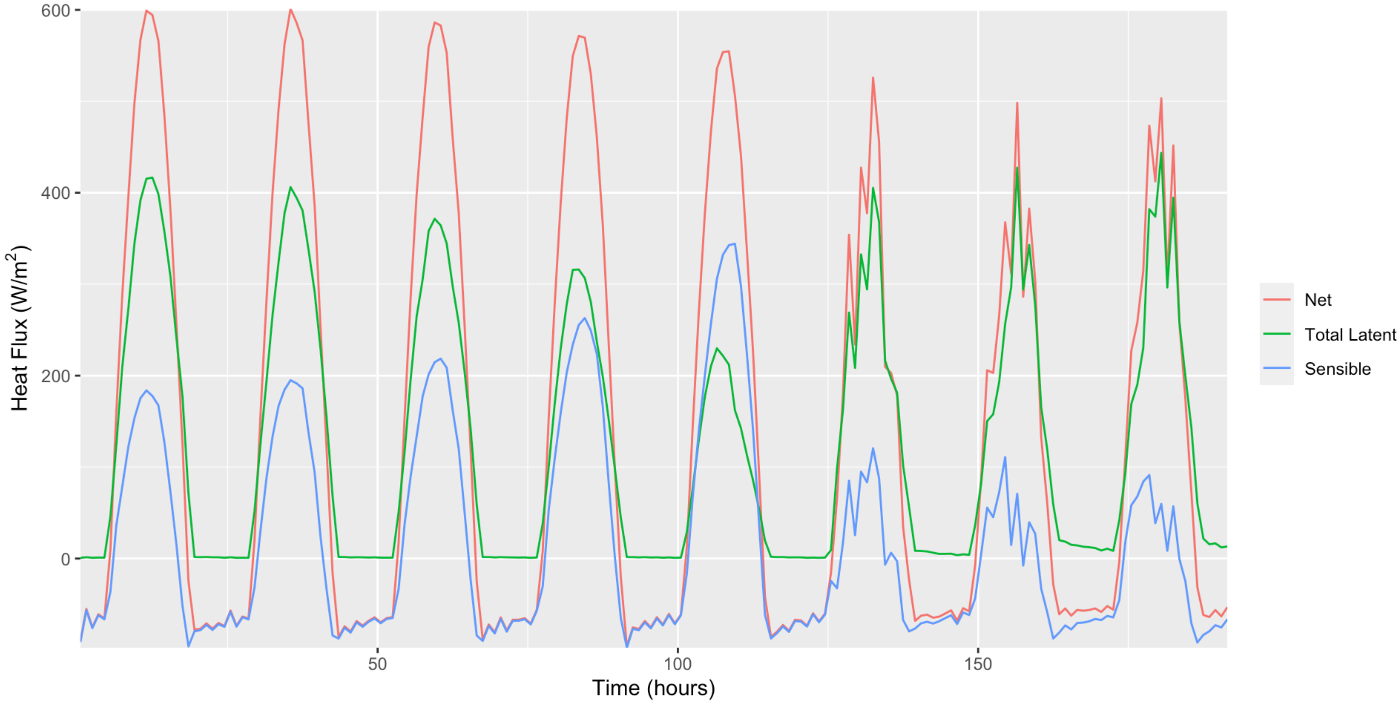
```r
crop <- "grass" # either "forest" or "grass"

# prescribed crop-specific values
LAI <- ifelse(crop == "forest", 4, 2) # leaf area index
h <- ifelse(crop == "forest", 20, 0.12) # m
alpha <- ifelse(crop == "forest", 0.12, 0.23) # surface albedo
g_0 <- ifelse(crop == "forest", 15, 30) / 1000 # m s^-1
SM_0 <- ifelse(crop == "forest", 80, 40) # maximum soil moisture holding capacity, mm
SM_init <- ifelse(crop == "forest", 40, 20) # initial soil moisture, mm
SM_max <- ifelse(crop == "forest", 80, 40) # maximum soil moisture available, mm
S <- ifelse(crop == "forest", 4, 2) # canopy capacity, mm
z_m <- ifelse(crop == "forest", 22, 2) # m
K_M_1 <- ifelse(crop == "forest", 3.36E-4, 1.87E-2) # unitless

# calculated
d <-  1.1 * h * log(1 + (LAI / 5) ^ 0.25) # m, equation 22.2
z_0 <- 0.3 * h * (1 - d/h) # m, equation 22.4

# prescribed crop-independent values
P <- 101.2 # kPa
k <- 0.4 # Von Kaman constant, unitless
epsilon_surf <- 0.95 # surface emissivity, unitless
sigma <- 5.67E-8 # W m-2 K-4
K_R <- 200 # W m-2
K_D_1 <- -0.307 # kPa-1
K_D_2 <- 0.019 # kPa-2
T_L <- 273 # K
T_0 <- 293 # K
T_H <- 313 # K
K_M_2 <- -0.1 # mm-1
rho_a <- 1.23 # kg m^-3
c_p <- 1013  # J kg^-1 K^-1
g_c <- 1 # unitless
alpha_T <- 1 # unitless
```

```
for (t in seq_range(data$t, by=dt)) {
  # extract inputs at time t
  u_m = data$u_m[data$t == t] # windspeed, m s-1
  q = data$q[data$t == t] / 1000 # specific humidity, converted to kg kg-1
  T_C = data$T_C[data$t == t] # measured temperature, C
  T_K = T_C + 273.17 # measured temperature, K
  R_s = data$R_s[data$t == t] # solar radiation, W m-2
  L_d = data$L_d[data$t == t] # downward longwave radiation, W m-2
  R = data$R[data$t == t] # precip, mm

  if (t == min(data$t)) {
    # initialize soil moisture
    SM_last = SM_init

    # initialize surface temperature as measured temperature
    T_surf_K = T_K
  } else {
    # set soil moisture to that from the previous timestep
    SM_last = data$SM_new[data$t == (t - dt)]

    # set surface temperature to that from the previous timestep
    T_surf_K = data$T_surf_K[data$t == (t - dt)]
  }

  # calculate surface resistance variables at time t
  r_a = (1 / (k ^ 2 * u_m)) * log((z_m - d) / z_0) * log((z_m - d) / (z_0 / 10)) # aerod
ynamic resistance, s m-1, eq. 22.9
  e = q * P / 0.622 # vapor pressure, kPa, eq. 2.9
  e_sat = 0.6108 * exp(17.27 * T_C / (237.3 + T_C)) # saturated vapor pressure at temper
ature T, kPa, eq. 2.20
  D = e_sat - e # vapor pressure deficit, kPa, eq. 2.17
  g_R = max(R_s * (1000 + K_R) / (1000 * (R_s + K_R)), 0) # eq. 24.2
  g_D = max(1 + K_D_1 * D + K_D_2 * D ^ 2, 0) # eq. 24.3
  g_T = max((T_K - T_L) * (T_H - T_K) ^ alpha_T / ((T_0 - T_L) * (T_H - T_0) ^ alpha_T),
0) # eq. 24.4
  g_M = max(1 - K_M_1 * exp(K_M_2 * (SM_last - SM_0)) , 0) # eq. 24.6
  g_S = max(g_0 * g_c * g_R * g_D * g_T * g_M, 0.0001) # eq. 24.1
  r_s = 1 / g_S # surface resistance, eq. 24.1

  # calculate radiation variables at time t
  if (t == min(data$t)) {
    # use initialized surface temp for first timestep
    L_u = -epsilon_surf * sigma * T_surf_K ^ 4
  } else if (t == (min(data$t) + dt)) {
    # use surface temp from previous timestep
    mean_T = data$T_surf_K[data$t == (t - dt)]
    L_u = -epsilon_surf * sigma * mean_T ^ 4
  } else {
    # use surface temps from previous two timesteps
    mean_T = mean(data$T_surf_K[data$t == (t - dt)], data$T_surf_K[data$t == (t - 2*d
t)])
    L_u = -epsilon_surf * sigma * mean_T ^ 4
```

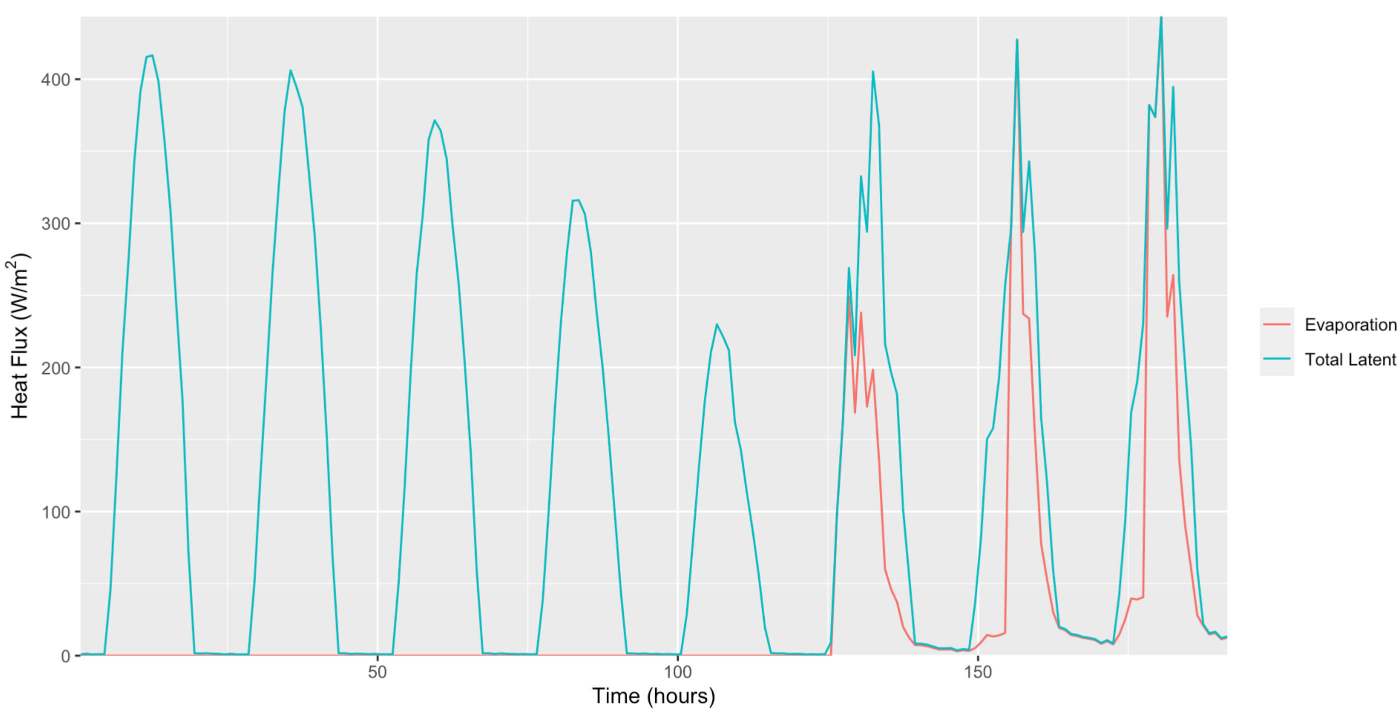
```
  }

  R_n = R_s * (1 - alpha) + L_d + L_u # incoming minus outgoing

  # calculate canopy and soil water balance variables at time t
  lambda = (2.501 - 0.002361 * T_C) * 1E6 # latent heat of vaporization, J kg-1, eq.
  delta = 4098 * e_sat / (237.7 + T_C) ^ 2 # slope of the saturation vapor pressure curv
e, kPa K-1
  gamma = c_p * P / (0.622 * lambda) # psychrometric constant, kPa K-1

  if (t == min(data$t)) {
    C_interim = 0
  } else {
    C_interim = R + data$C_final[data$t == (t - dt)] # mm
  }

  if (C_interim <= S) {
    C_actual = C_interim
    D_canopy = 0
  } else {
    C_actual = S
    D_canopy = C_interim - S
  }

  C_over_S = C_actual / S

  A = R_n # available energy (assumption)

  lambdaE_I = (C_over_S) * max((delta * A + rho_a * c_p * D / r_a) / (delta + gamma), 0)
# W m-2, eq. 22.14
  lambdaE_T = max((delta * A + rho_a * c_p * D / r_a) / (delta + gamma * (1 + r_s / r_
a)), 0) # W m-2, eq. 22.18
  lambdaE = lambdaE_I + (1 - C_over_S) * lambdaE_T # eq. 22.17 (corrected)
  H = R_n - lambdaE
  C_final = max(C_actual - lambdaE_I * (dt * 3600 / (lambda * rho_a)), 0) # convert lamb
daE_I to mm
  SM_new = max(min(SM_last + D_canopy - lambdaE_T * (dt * 3600 / (lambda * rho_a)), SM_m
ax), 0) # convert lambdaE_T to mm

  T_surf_K = T_K + H * r_a / (rho_a * c_p)

  # update values for time t
  data$T_K[data$t == t] = T_K # K
  data$r_a[data$t == t] = r_a # s m-1
  data$e[data$t == t] = e # kPa
  data$e_sat[data$t == t] = e_sat # kPa
  data$D[data$t == t] = D
  data$g_R[data$t == t] = g_R
  data$g_D[data$t == t] = g_D
  data$g_T[data$t == t] = g_T
  data$g_M[data$t == t] = g_M
  data$g_S[data$t == t] = g_S * 1000 # mm s-1
```

```
  data$r_s[data$t == t] = r_s
  data$L_u[data$t == t] = L_u
  data$R_n[data$t == t] = R_n
  data$lambda[data$t == t] = lambda # J kg-1
  data$delta[data$t == t] = delta # kPa K-1
  data$gamma[data$t == t] = gamma # kPa K-1
  data$C_interim[data$t == t] = C_interim # mm
  data$C_actual[data$t == t] = C_actual # mm
  data$C_final[data$t == t] = C_final # mm
  data$C_over_S[data$t == t] = C_over_S
  data$D_canopy[data$t == t] = D_canopy # mm
  data$lambdaE_I[data$t == t] = lambdaE_I # W m-2
  data$lambdaE_T[data$t == t] = lambdaE_T # W m-2
  data$lambdaE[data$t == t] = lambdaE # mm
  data$H[data$t == t] = H # W m-2
  data$T_surf_K[data$t == t] = T_surf_K # K
  data$SM_last[data$t == t] = SM_last # mm
  data$SM_new[data$t == t] = SM_new # mm

}

variable_levels <- c("u_m", "q", "R_s", "L_d", "R", "T_K",
                     "r_a", "e", "e_sat", "D", "g_R", "g_D", "g_T", "g_M", "g_S", "SM_la
st", "r_s",
                     "L_u", "R_n", "lambda", "delta", "gamma", "C_interim", "C_actual",
"C_final", "C_over_S",
                     "D_canopy", "lambdaE_I", "lambdaE_T", "lambdaE", "H", "T_surf_K",
"SM_new")

data_longer <- data %>%
  pivot_longer(cols=c(-t)) %>%
  filter(name %in% variable_levels) %>%
  mutate(name=factor(name, levels=variable_levels))
```

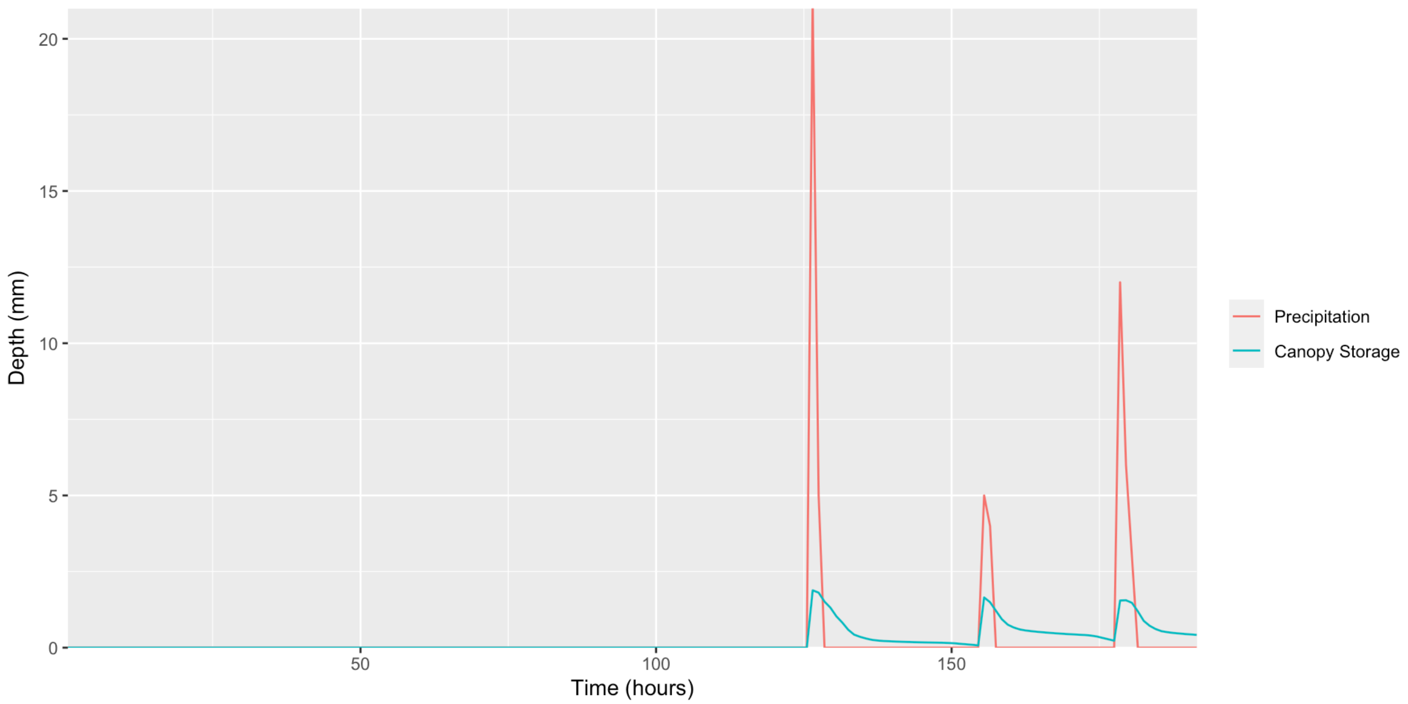
```
# I. Hourly values of the calculated net radiation, total latent heat flux, and sensible
heat flux (all in W m-2) on the same graph
ggplot(data_longer %>% filter(name %in% c("R_n", "lambdaE", "H")), aes(x=t, y=value, col
=name)) +
  geom_path() +
  scale_color_discrete(name=NULL, labels=c("Net", "Total Latent", "Sensible")) +
  scale_y_continuous(name=expression(paste("Heat Flux ", "(W/m"^2, ")")), expand=c(0,0))
+
  scale_x_continuous(name="Time (hours)", expand=c(0,0))
```



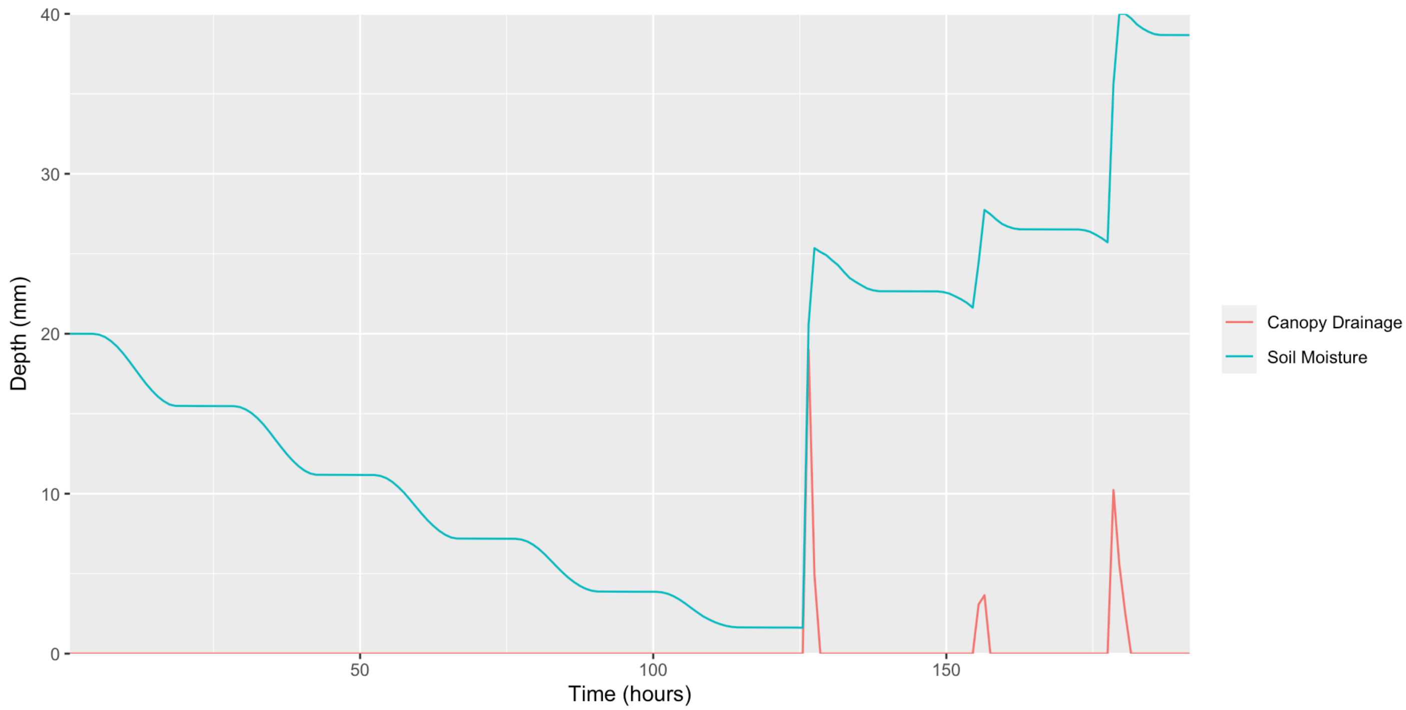

```
# II. Hourly values of the calculated total latent heat flux and the portion of the late
nt heat flux that originates from the evaporation of intercepted water (both in W m-2)
ggplot(data_longer %>% filter(name %in% c("lambdaE_I", "lambdaE")), aes(x=t, y=value, co
l=name)) +
  geom_path() +
  scale_color_discrete(name=NULL, labels=c("Evaporation", "Total Latent")) +
  scale_y_continuous(name=expression(paste("Heat Flux ", "(W/m"^2, ")")), expand=c(0,0))
+
  scale_x_continuous(name="Time (hours)", expand=c(0,0))
```



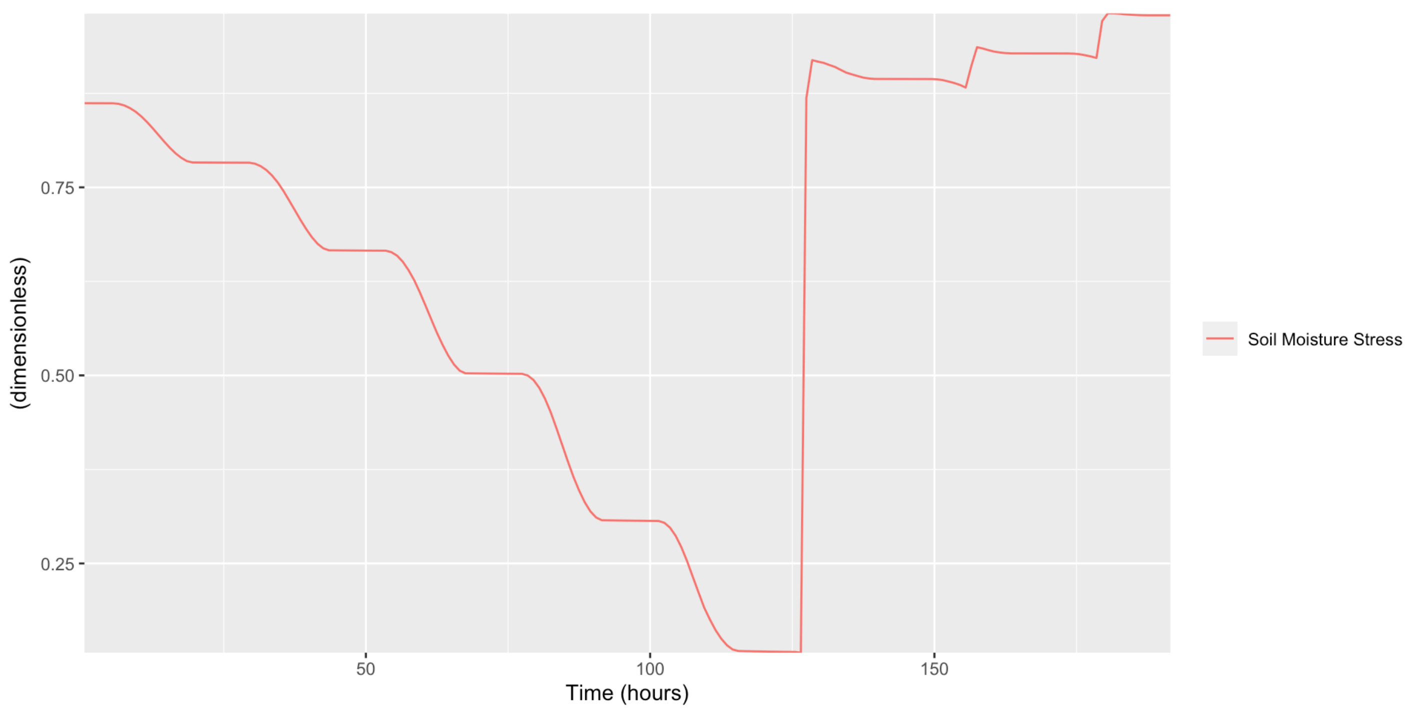
```r
# III. Hourly values of the precipitation and the canopy storage C (both in mm)
ggplot(data_longer %>% filter(name %in% c("R", "C_final")), aes(x=t, y=value, col=name))
+
  geom_path() +
  scale_color_discrete(name=NULL, labels=c("Precipitation", "Canopy Storage")) +
  scale_y_continuous(name=expression(paste("Depth ", "(mm)")), expand=c(0,0)) +
  scale_x_continuous(name="Time (hours)", expand=c(0,0))
```




```r
# IV. Hourly values of the canopy drainage [if interception was considered] the value of
SM (both in mm)
ggplot(data_longer %>% filter(name %in% c("D_canopy", "SM_new")), aes(x=t, y=value, col=
name)) +
  geom_path() +
  scale_color_discrete(name=NULL, labels=c("Canopy Drainage", "Soil Moisture")) +
  scale_y_continuous(name=expression(paste("Depth ", "(mm)")), expand=c(0,0)) +
  scale_x_continuous(name="Time (hours)", expand=c(0,0))
```




```r
# V. Hourly values of the soil moisture stress function gM (no units)
ggplot(data_longer %>% filter(name %in% c("g_M")), aes(x=t, y=value, col=name)) +
  geom_path() +
  scale_color_discrete(name=NULL, labels=c("Soil Moisture Stress")) +
  scale_y_continuous(name=expression(paste("(dimensionless)")), expand=c(0,0)) +
  scale_x_continuous(name="Time (hours)", expand=c(0,0))
```



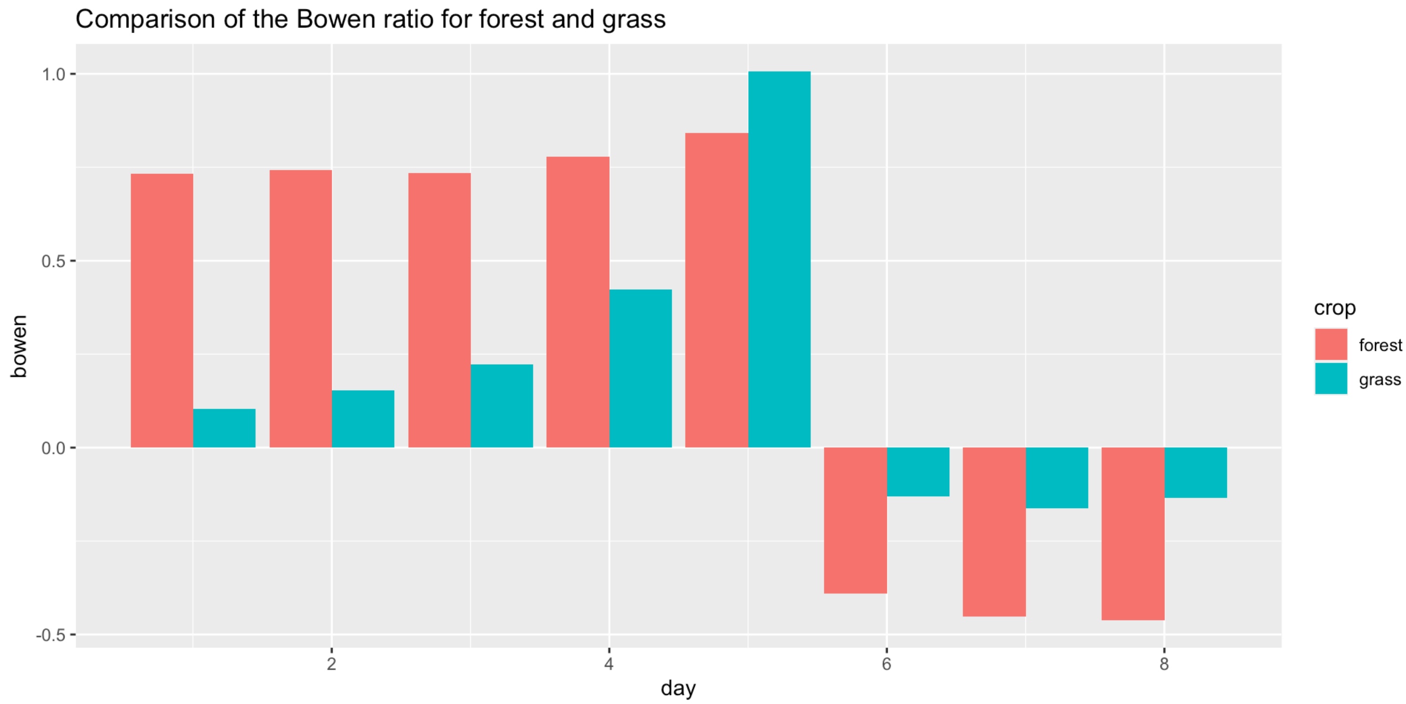
```
# VI. A table giving for all 8 individual days the daily average values of  Rn, lambdaE, 
H, and lambdaE_I (all in W m-2), and the values of daily average Bowen Ratio and the rat
io of daily average fractional contribution to daily average total evaporation that aris
es from the evaporation of intercepted water
data_daily <- data_longer %>%
  mutate(day = ((t-0.5) %/% 24 + 1)) %>%
  group_by(day, name) %>%
  summarize(value=mean(value)) %>%
  pivot_wider(id_cols=c("day")) %>%
  select(day, R_n, lambdaE, H, lambdaE_I, lambdaE_T) %>%
  mutate(bowen=H/lambdaE,
         evap_over_latent=lambdaE_I/lambdaE)

print(data_daily)
```

```
## # A tibble: 8 × 8
## # Groups:   day [8]
##     day   R_n lambdaE     H lambdaE_I lambdaE_T  bowen evap_over_latent
##   <dbl> <dbl>   <dbl> <dbl>     <dbl>     <dbl>  <dbl>            <dbl>
## 1     1  174.   158.   16.4       0       158.   0.104            0    
## 2     2  174.   151.   23.2       0       151.   0.154            0    
## 3     3  170.   139.   31.0       0       139.   0.223            0    
## 4     4  165.   116.   48.9       0       116.   0.423            0    
## 5     5  157.    78.3  78.9       0        78.3  1.01             0    
## 6     6  107.   123.  -16.0      68.0     101.  -0.130            0.555
## 7     7  103.   123.  -20.0      69.7     100.  -0.162            0.565
## 8     8  123.   142.  -19.0      95.6     112.  -0.134            0.673
```

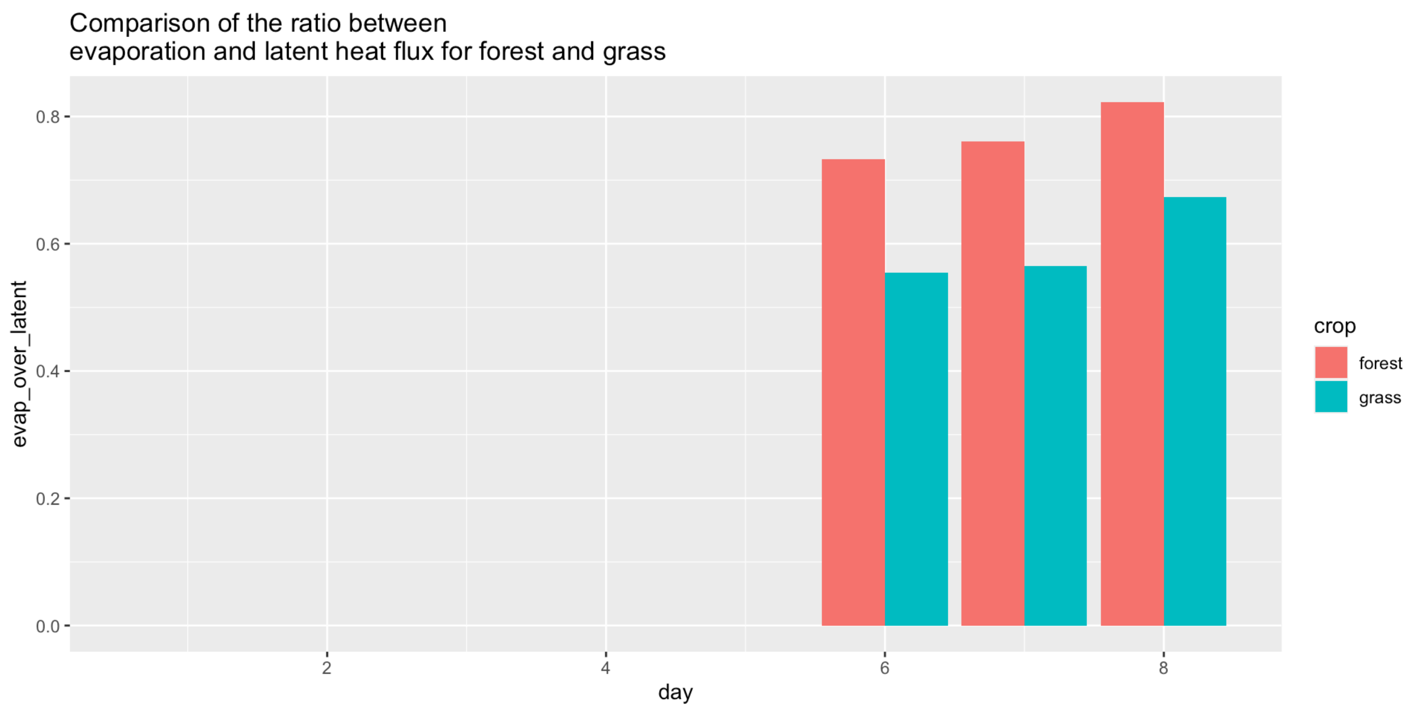
```
data_longer_grass <- data_longer
data_daily_grass <- data_daily
```

Commentary

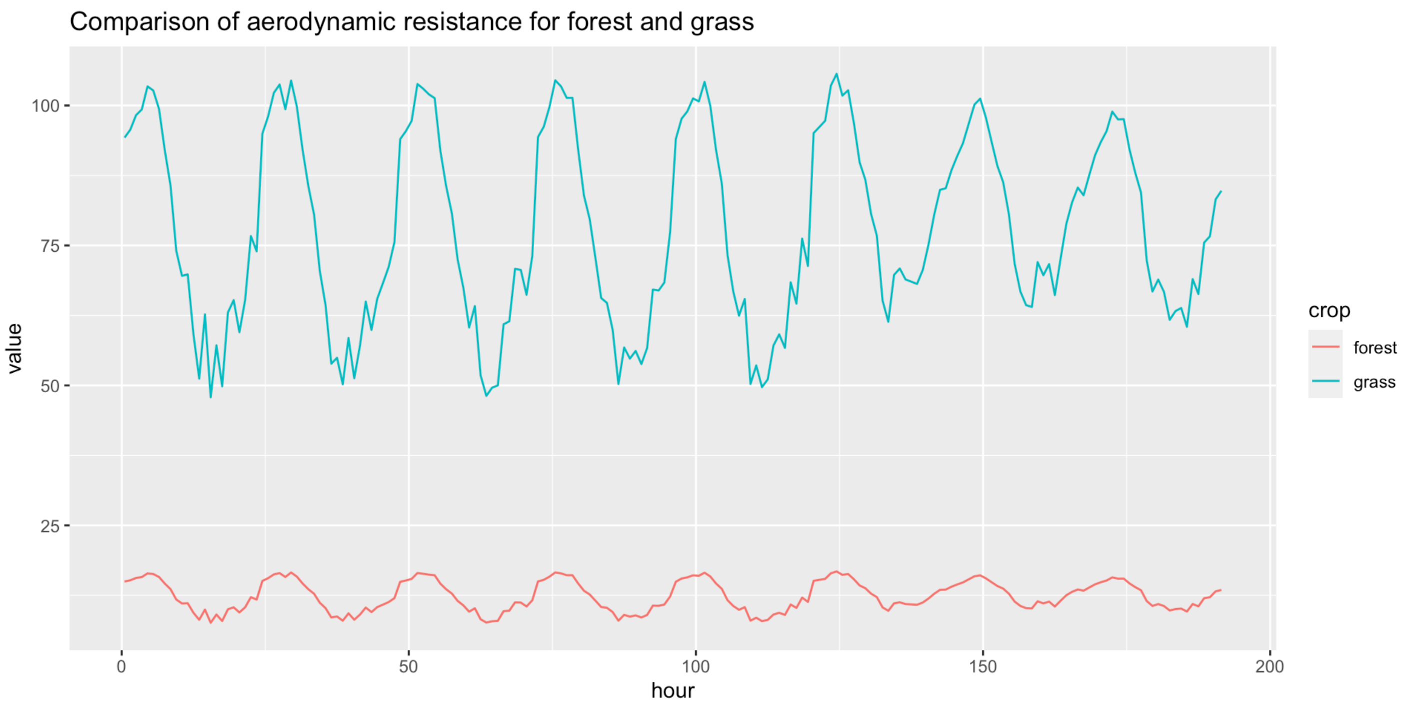
```
# plot the daily mean Bowen ratio for forest and grass
data_daily_combined <- rbind(data_daily_forest %>% mutate(crop="forest"),
                             data_daily_grass %>% mutate(crop="grass"))

ggplot(data_daily_combined, aes(x=day, y=bowen, fill=crop)) +
  geom_bar(stat="identity", position = "dodge") +
  ggtitle("Comparison of the Bowen ratio for forest and grass")
```



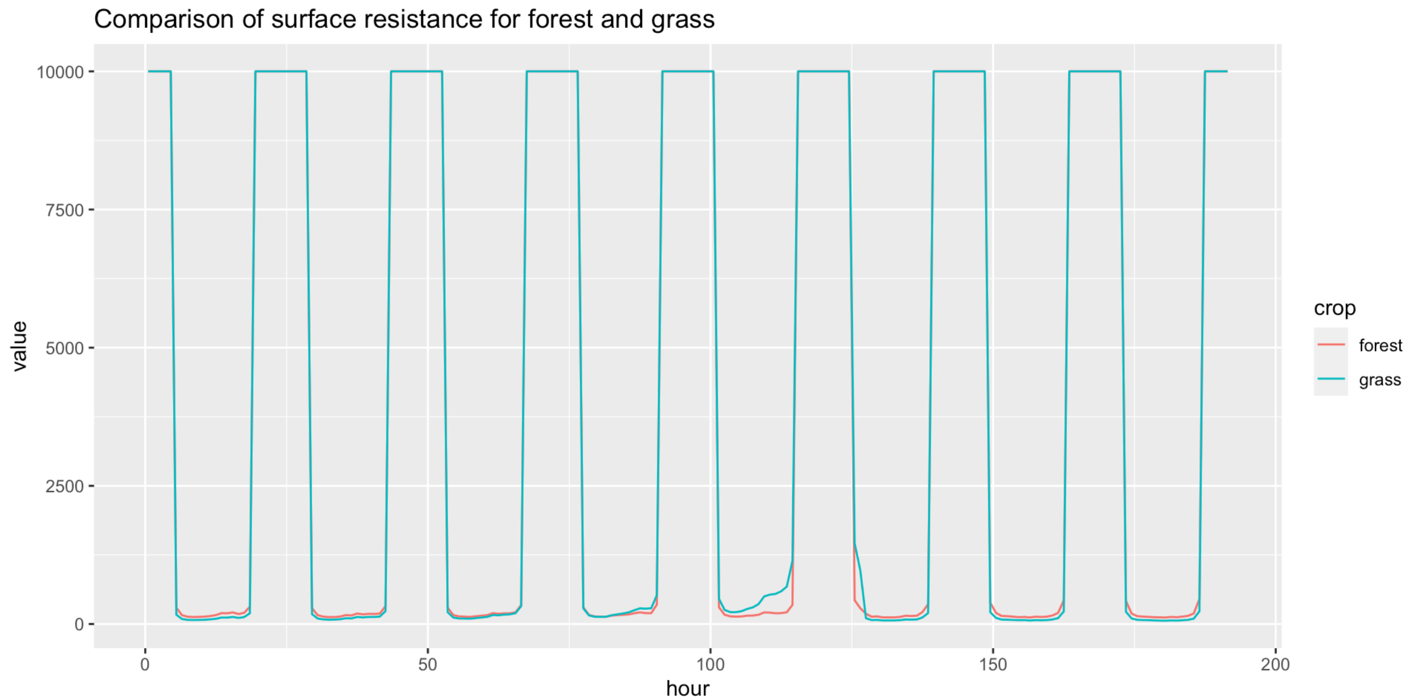

```
ggplot(data_daily_combined, aes(x=day, y=evap_over_latent, fill=crop)) +
  geom_bar(stat="identity", position = "dodge") +
  ggtitle("Comparison of the ratio between \nevaporation and latent heat flux for forest
and grass")
```



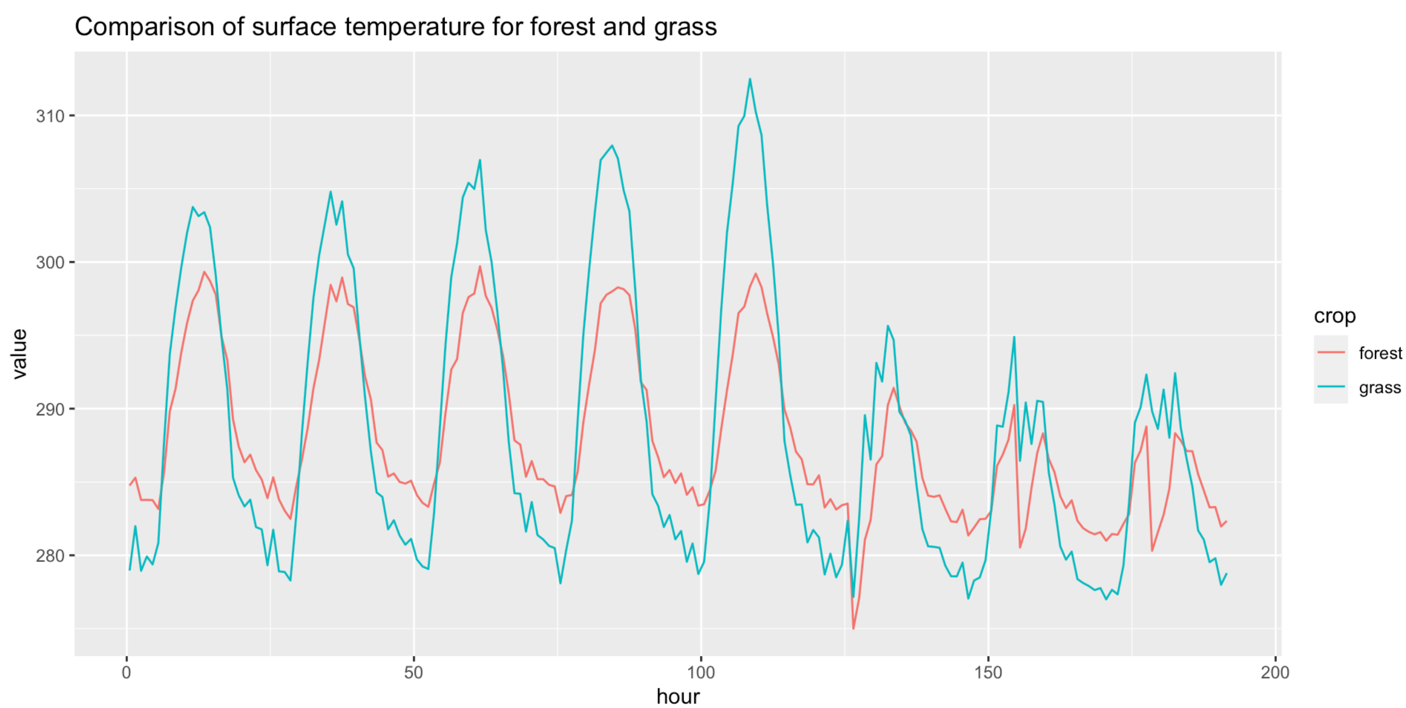
```r
# plot the hourly aerodynamic resistances for forest and grass
data_longer_combined <- rbind(data_longer_forest %>% mutate(crop="forest"),
                              data_longer_grass %>% mutate(crop="grass"))

saveRDS(data_longer_combined, "SVAT_outputs_longer.RDS")

ggplot(data_longer_combined %>% filter(name=="r_a"), aes(x=t, y=value, col=crop)) +
  geom_path() +
  ggtitle("Comparison of aerodynamic resistance for forest and grass") +
  xlab("hour")
```



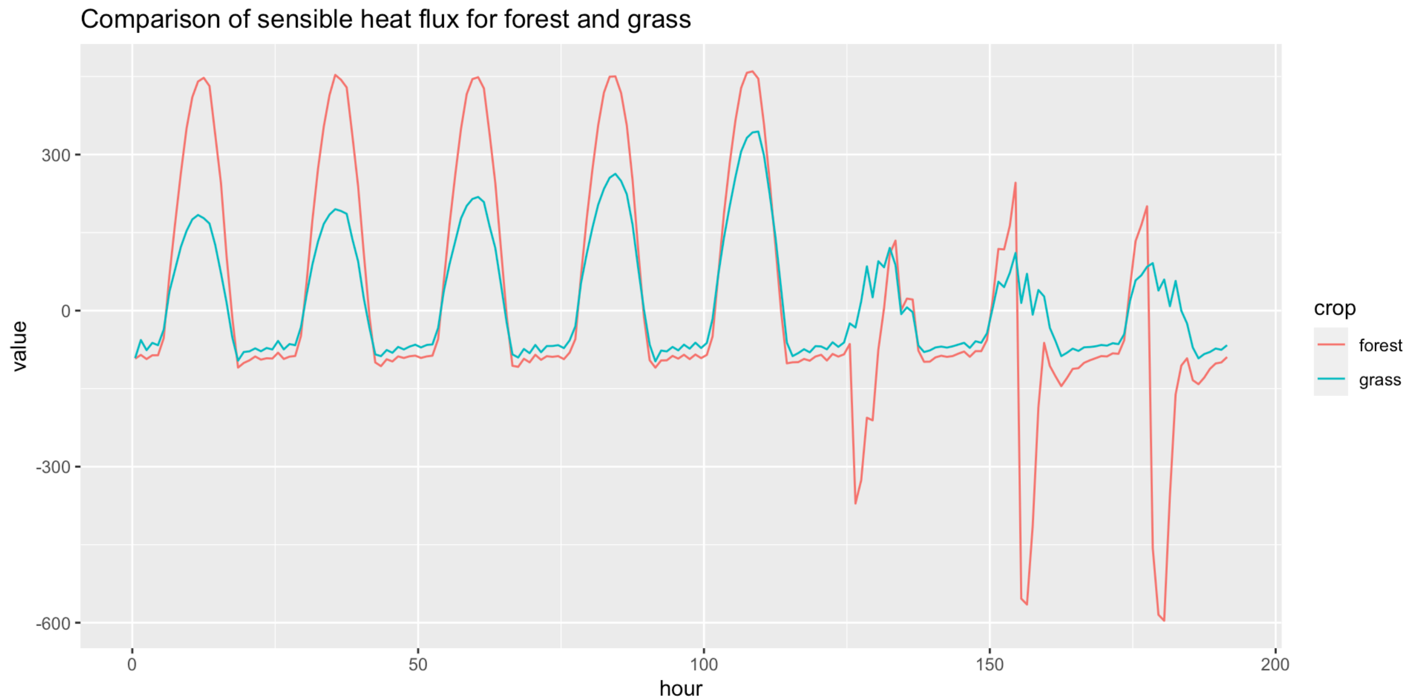

```r
ggplot(data_longer_combined %>% filter(name=="r_s"), aes(x=t, y=value, col=crop)) +
  geom_path() +
  ggtitle("Comparison of surface resistance for forest and grass") +
  xlab("hour")
```



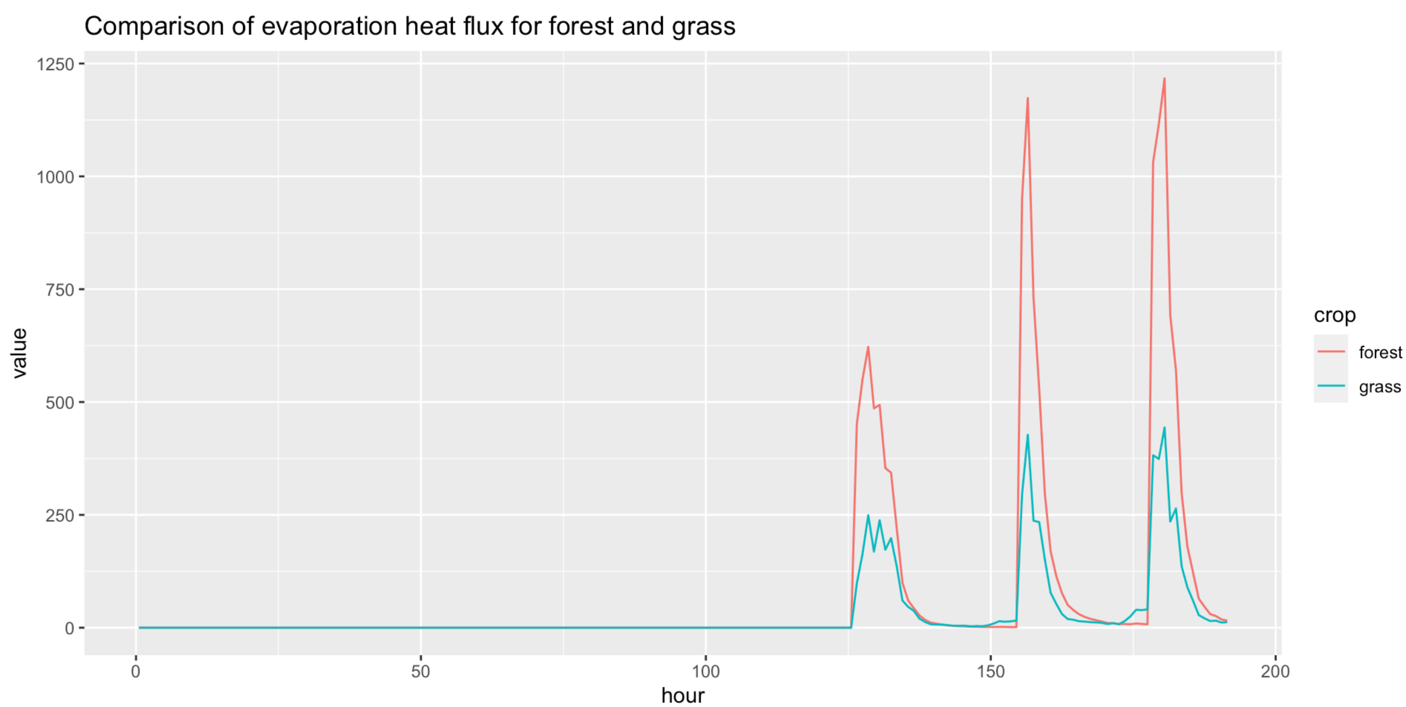

```
ggplot(data_longer_combined %>% filter(name=="T_surf_K"), aes(x=t, y=value, col=crop)) +
  geom_path() +
  ggtitle("Comparison of surface temperature for forest and grass") +
  xlab("hour")
```



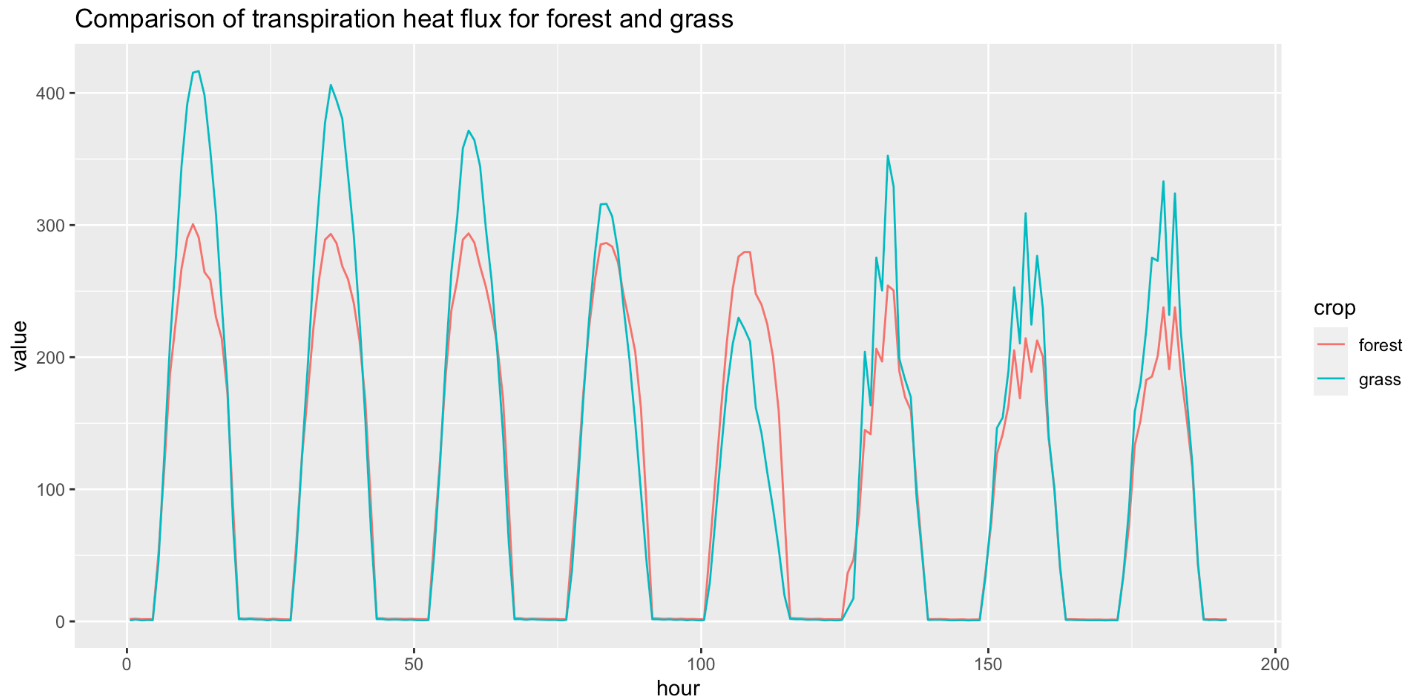

```
ggplot(data_longer_combined %>% filter(name=="H"), aes(x=t, y=value, col=crop)) +
  geom_path() +
  ggtitle("Comparison of sensible heat flux for forest and grass") +
  xlab("hour")
```



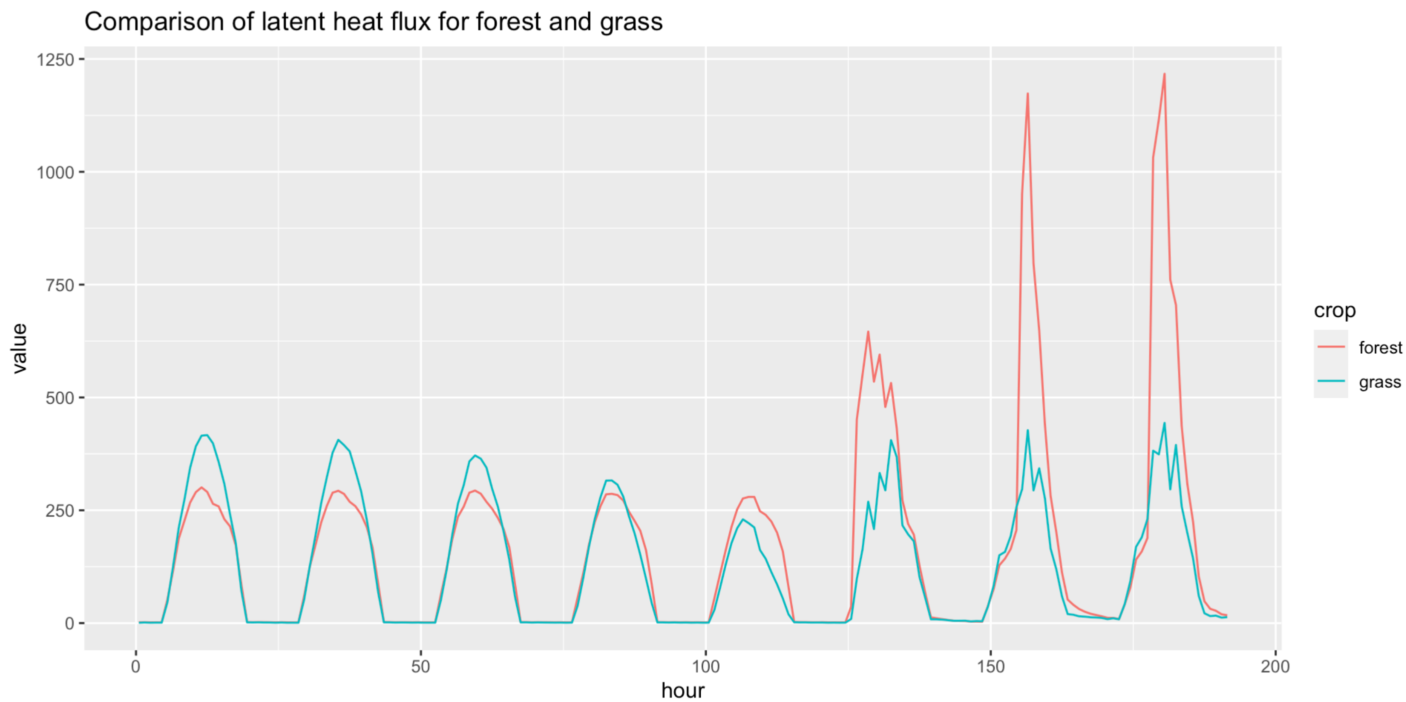

```
ggplot(data_longer_combined %>% filter(name=="lambdaE_I"), aes(x=t, y=value, col=crop))
+
  geom_path() +
  ggtitle("Comparison of evaporation heat flux for forest and grass") +
  xlab("hour")
```



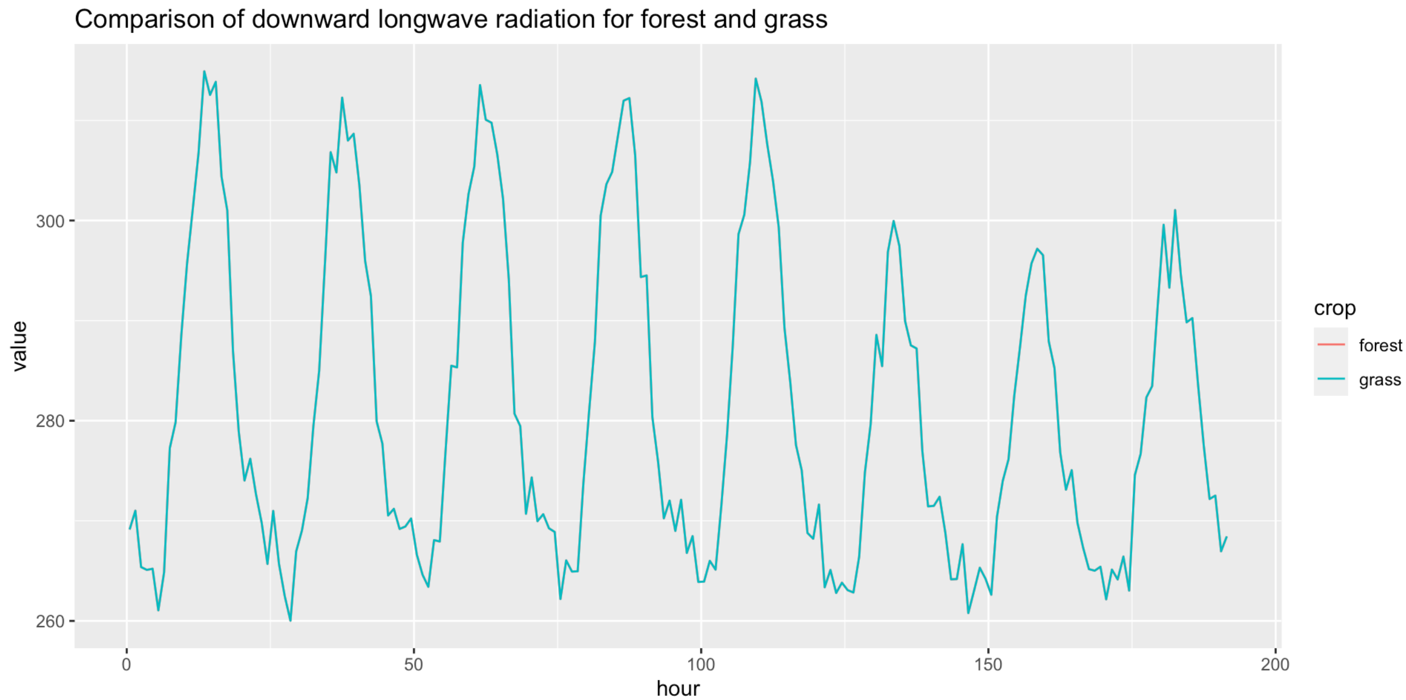

```
ggplot(data_longer_combined %>% filter(name=="lambdaE_T"), aes(x=t, y=value, col=crop))
+
  geom_path() +
  ggtitle("Comparison of transpiration heat flux for forest and grass") +
  xlab("hour")
```



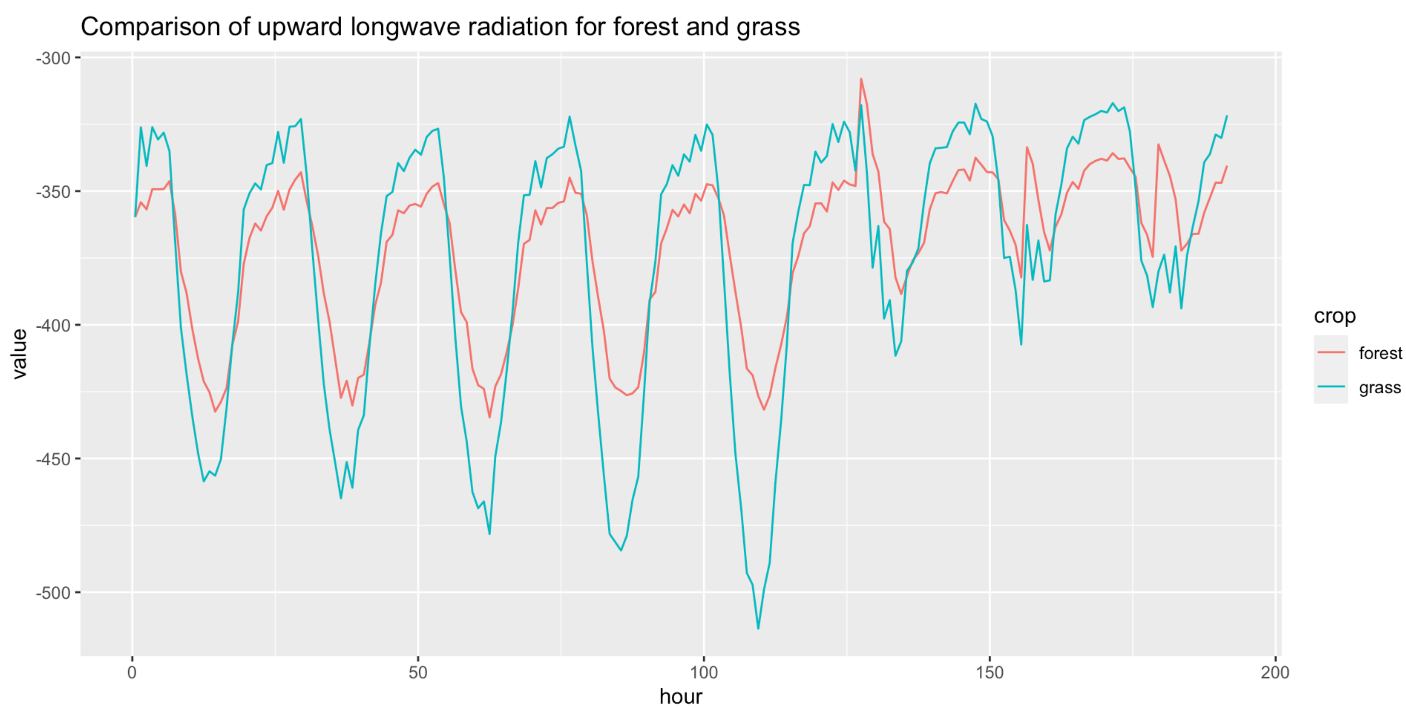

```
ggplot(data_longer_combined %>% filter(name=="lambdaE"), aes(x=t, y=value, col=crop)) +
  geom_path() +
  ggtitle("Comparison of latent heat flux for forest and grass") +
  xlab("hour")
```



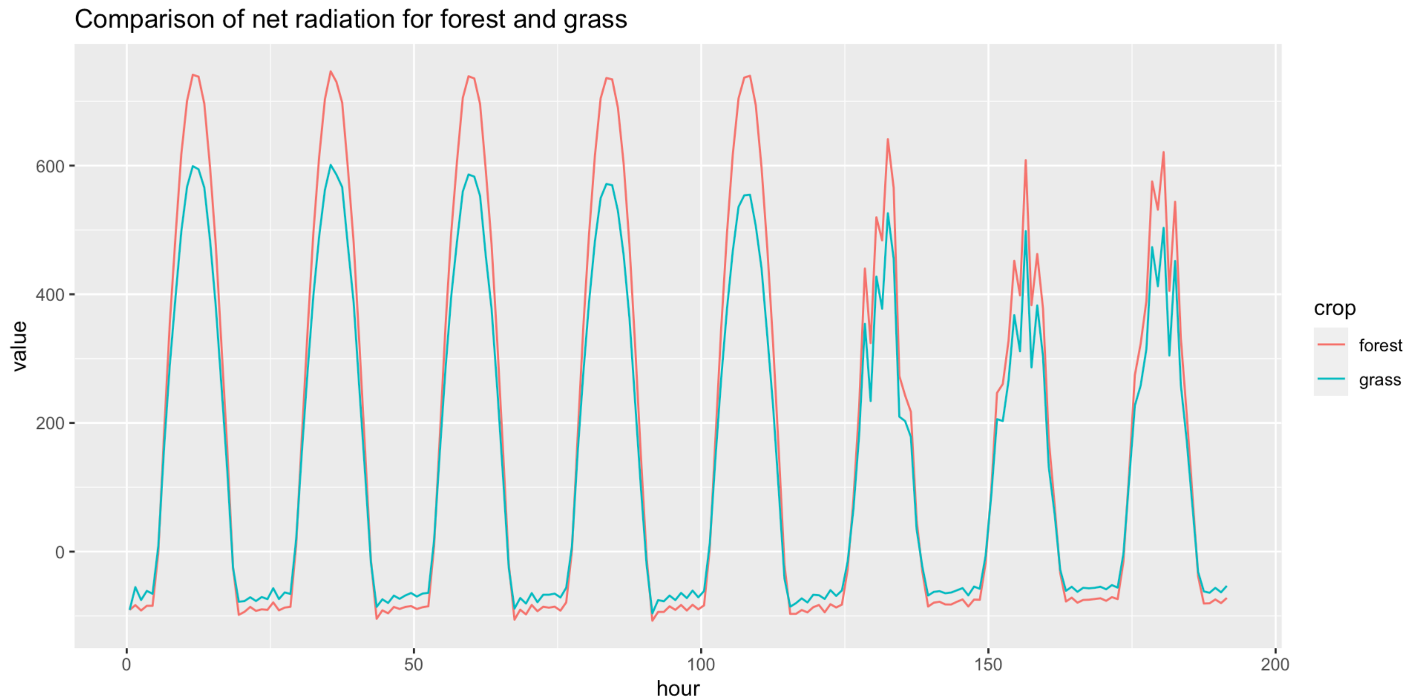

```
ggplot(data_longer_combined %>% filter(name=="L_d"), aes(x=t, y=value, col=crop)) +
  geom_path() +
  ggtitle("Comparison of downward longwave radiation for forest and grass") +
  xlab("hour")
```




```
ggplot(data_longer_combined %>% filter(name=="L_u"), aes(x=t, y=value, col=crop)) +
  geom_path() +
  ggtitle("Comparison of upward longwave radiation for forest and grass") +
  xlab("hour")
```




```
ggplot(data_longer_combined %>% filter(name=="R_n"), aes(x=t, y=value, col=crop)) +
  geom_path() +
  ggtitle("Comparison of net radiation for forest and grass") +
  xlab("hour")
```



Comment on the relative differences in values of daily average Bowen Ratio and on the ratio of daily average fractional contribution to daily average total evaporation that arises from the evaporation of intercepted water you find for forest and grass.

The Bowen ratio is the ratio of sensible to latent heat flux. The value of this ratio is larger for the forest crop for all of the dry days except the 5th day. This is due to the sensible heat flux being larger for forest than grass, as well as the latent heat flux being smaller for forest than grass (e.g., on day 1: 94.5 > 15.3 and 125 < 165, respectively), and makes sense on days where it did not rain, for several reasons:

1. The forest is darker (i.e., has a lower albedo) and will absorb more infrared radiation, increasing sensible heat flux;
2. Although the lower sensible heat flux for grass reduces the degree of surface temperature change, it is outweighed by the much higher aerodynamic resistance, which increases the degree of surface temperature change.

The net result of this is that more heat goes to warming the surface for the grass and more soil moisture is lost to transpiration. The reason the ratio has shifted on the 5th day is that the depleted soil moisture for the grass surface (which had lower initial and maximum available soil moisture than the forest) has reduced transpiration heat flux to the point where it is roughly equal to the sensible heat flux.

Conversely to the first five days of the time series, following the rainfall on day 6 we see larger rates of negative heat flux from the forest, perhaps due in part to the much larger surface area (LAI) for evaporation, but also likely due to the large difference in aerodynamic resistance previously discussed (since r_a is in the denominator, the larger value for grass decreases evaporation relative to the forest).

In summary:

- Conceptually:
    - The forest has lower rates of transpiration than grass due to the lower surface gradients of wind, temperature, humidity, etc., but higher rates of sensible heat flux due to the darker surface color.
    - The forest has higher rates of evaporation following rainfall due to both the darker color and the higher LAI (more surface area for evaporation).

    - The forest is less sensitive to changes in air temperature and soil moisture than the grass due again to the lower surface gradients and to the higher availability of soil moisture.
- Functionally:
    - The albedo and the parameterization of the resistance variables are highly influential on these differences in how forested and grass surfaces respond to the forcing data.