

Synchrony investigations - Fagus masting

Victor, Lizzie

2024-08-22

Get E-OBS climate data

```
rerun <- FALSE # switch because computation is too long

if(rerun){
  eobs_r <- rast(file.path(wd, "data/eobs", "tg_ens_mean_0.1deg_reg_v29.0e.nc"))

  # Loop on sites to get E-OBS daily temperature for the years recorded
  process_data <- data.frame()
  for(s in unique(journe_d$sitenewname)){

    lat <- unique(journe_d[journe_d$sitenewname == s, "Latitude"])
    lon <- unique(journe_d[journe_d$sitenewname == s, "Longitude"])
    years <- journe_d[journe_d$sitenewname == s, "Year"]
    years <- c(min(years)-2, min(years-1), years) # two years before seed fall

    sub_r <- subset(eobs_r, which(time(eobs_r, format = "years") %in% years))
    tmean <- terra::extract(
      sub_r,
      matrix(c(lon, lat), nrow = 1))

    process_data <- rbind(process_data, data.frame(Latitude = lat, Longitude = lon, sitenewname = s, Tmean = tmean,
                                                    Date = time(sub_r), Year = time(sub_r, format = "year")))
  }

  # Add annual seed production (including one and two years later) -- not a very elegant way
  journe_T2 <- journe_d[,c("Longitude", "Latitude", "sitenewname", "Year", "Value")]
  journe_T2$Year <- journe_T2$Year-2
  names(journe_T2)[5] <- "Value2"
  journe_T1 <- journe_d[,c("Longitude", "Latitude", "sitenewname", "Year", "Value")]
  journe_T1$Year <- journe_T1$Year-1
  names(journe_T1)[5] <- "Value1"
  process_data <- process_data %>%
    dplyr::left_join(journe_d[c("Longitude", "Latitude", "sitenewname", "Year", "Value")],
                     by = c("Longitude", "Latitude", "sitenewname", "Year")) %>%
    dplyr::left_join(journe_T1[c("Longitude", "Latitude", "sitenewname", "Year", "Value1")],
                     by = c("Longitude", "Latitude", "sitenewname", "Year")) %>%
    dplyr::left_join(journe_T2[c("Longitude", "Latitude", "sitenewname", "Year", "Value2")],
                     by = c("Longitude", "Latitude", "sitenewname", "Year"))

  saveRDS(process_data, file = file.path(wd, "data/processed", "journe2024_replicate.rds"))
}
```

```

} else{

  process_data <- readRDS(file.path(wd, "data/processed", "journe2024_replicate.rds"))

}

```

Computing the “moving window correlation”

From my understanding, the correlation is computed as follow.

```

## from Journe et al. "a moving Spearman correlation between
## log-transformed annual seed production and mean daily temperature"

dat <- data.frame(
  site = process_data$sitenewname,
  doy = as.numeric(strftime(process_data$Date, format = "%j")),
  tmean = process_data$Tmean,
  prod0 = process_data$value,
  prod1 = process_data$value1,
  prod2 = process_data$value2
)

# Moving window correlation across year for each site (note the log-transformation of seed production)
## according to the paper, log and not log+1, so I guess they did the calculation with log(0) = -Inf?
mwcov <- lapply(7:365, function(d) {

  dat %>%
    dplyr::filter(doy >= (d-6) & doy <= d) %>% # 7-day window
    dplyr::select(-doy) %>%
    group_by(site) %>%
    reframe(
      cor0 = cor(tmean, log(prod0), use="complete.obs", method = "spearman"),
      cor1 = cor(tmean, log(prod1), use="complete.obs", method = "spearman"),
      cor2 = cor(tmean, log(prod2), use="complete.obs", method = "spearman"),
      doy = d) %>%
    as.data.frame()

}

mwcov <- as.data.frame(do.call(rbind, mwcov)) %>%
  tidyr::pivot_longer(cols = c("cor0", "cor1", "cor2")) %>%
  # "trick" to plot in the correct order
  dplyr::mutate(doy_plot = if_else(name == "cor2", as.Date(doy, origin = "2001-01-01"),
                                    as.Date(ifelse(name == "cor1", as.Date(doy, origin = "2002-01-01"),
                                                   as.Date(doy, origin = "2003-01-01")))))

mwcov %>%
  group_by(doy_plot) %>%
  summarise(cor = mean(value)) %>%
  ggplot(aes(x=doy_plot)) +
  annotate("rect", xmin=as.Date("2001-01-01"), xmax=as.Date("2002-01-01"),
           ymin=-Inf, ymax=Inf, alpha=0.2, fill="lightgrey") +

```

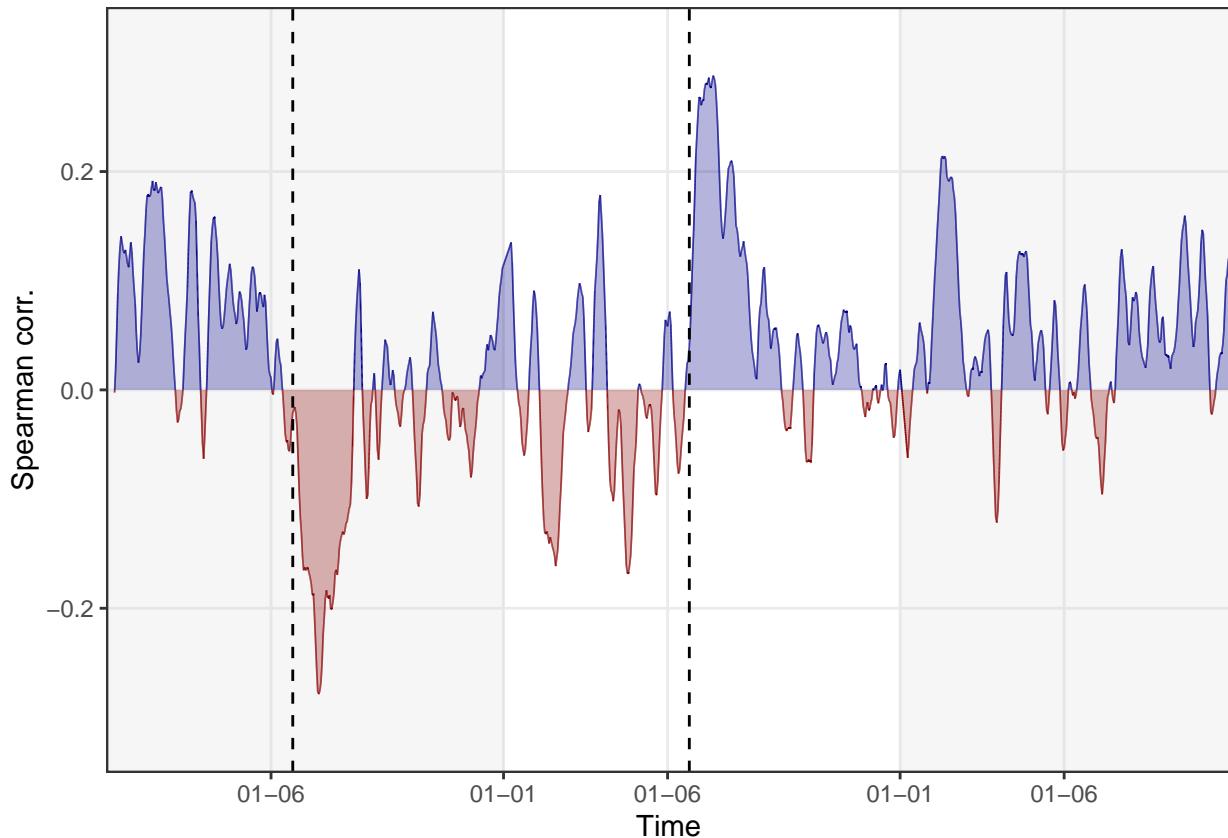
```

annotate("rect", xmin=as.Date("2003-01-01"), xmax=as.Date("2003-11-01"),
        ymin=-Inf, ymax=Inf, alpha=0.2, fill="lightgrey") +
ggforce::geom_link2(lwd=0.3, n = 50,
                     aes(y = cor, colour = after_stat(y < 0))) +
scale_colour_manual(values = c("darkblue", "darkred")) +
ggh4x::stat_difference(aes(ymin = 0, ymax = cor), alpha = 0.3) +
scale_fill_manual(values = c("darkblue", "darkred")) +
scale_x_date(date_labels = "%d-%m",
             breaks = as.Date(c("2001-06-01", "2002-01-01", "2002-06-01", "2003-01-01", "2003-06-01")),
             limits = as.Date(c(NA, "2003-11-01")), expand = c(0,0)) +
scale_y_continuous(limits = c(-0.35, 0.35), expand = c(0,0)) +
geom_vline(data = data.frame(sol = as.Date(c("2001-06-21", "2002-06-21"))),
           aes(xintercept = sol), linetype = "dashed") +
theme_bw() +
theme(legend.position = 'none',
      panel.grid.minor = element_blank()) +
labs(y = "Spearman corr.", x = "Time")

## Warning: Removed 61 rows containing non-finite outside the scale range
## ('stat_link2()').

## Warning: Removed 61 rows containing non-finite outside the scale range
## ('stat_difference()').

```



But we could also compute these correlations differently, right? If we take the correlation between moving average temperature and seed production, it gives quite similar results. (Note : my bad, this is actually what Valentin has done!)

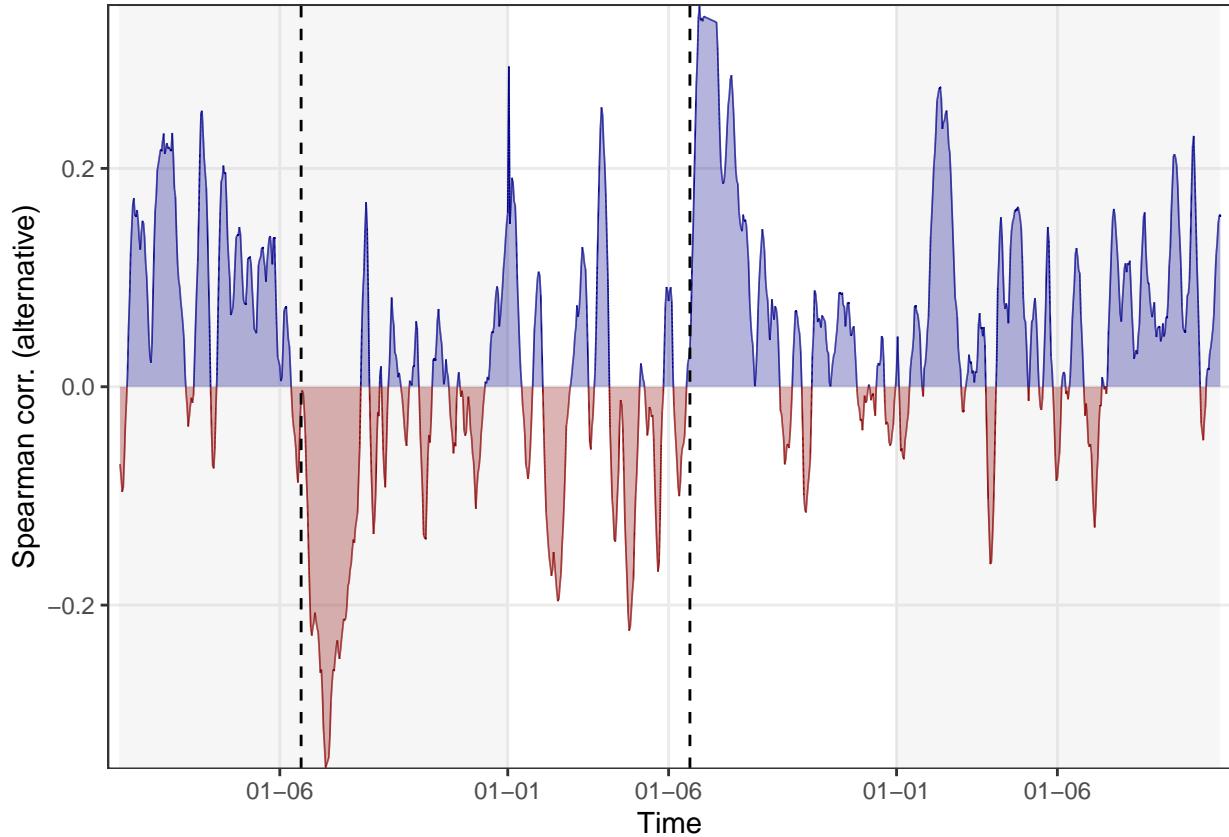
```
mwcor_alt <- dat %>%
  group_by(site) %>%
  dplyr::mutate(rwtmean = zoo::rollapplyr(tmean, 7, mean, fill = NA)) %>%
  ungroup() %>%
  group_by(site, doy) %>%
  dplyr::summarise(
    cor0 = cor(rwtmean, log(prod0), use="complete.obs", method = "spearman"),
    cor1 = cor(rwtmean, log(prod1), use="complete.obs", method = "spearman"),
    cor2 = cor(rwtmean, log(prod2), use="complete.obs", method = "spearman")
  ) %>%
  tidyr::pivot_longer(cols = c("cor0", "cor1", "cor2")) %>%
  dplyr::mutate(doy_plot = if_else(name == "cor2", as.Date(doy, origin = "2001-01-01"),
                                    as.Date(ifelse(name == "cor1", as.Date(doy, origin = "2002-01-01"),
                                    as.Date(doy, origin = "2003-01-01")))))
```

‘summarise()’ has grouped output by ‘site’. You can override using the
‘.groups’ argument.

```
mwcor_alt %>%
  group_by(doy_plot) %>%
  summarise(cor = mean(value)) %>%
  ggplot(aes(x=doy_plot)) +
  annotate("rect", xmin=as.Date("2001-01-01"), xmax=as.Date("2002-01-01"),
           ymin=-Inf, ymax=Inf, alpha=0.2, fill="lightgrey") +
  annotate("rect", xmin=as.Date("2003-01-01"), xmax=as.Date("2003-11-01"),
           ymin=-Inf, ymax=Inf, alpha=0.2, fill="lightgrey") +
  ggforce::geom_link2(lwd=0.3, n = 50,
    aes(y = cor, colour = after_stat(y < 0))) +
  scale_colour_manual(values = c("darkblue", "darkred")) +
  ggh4x::stat_difference(aes(ymin = 0, ymax = cor), alpha = 0.3) +
  scale_fill_manual(values = c("darkblue", "darkred")) +
  scale_x_date(date_labels = "%d-%m",
               breaks = as.Date(c("2001-06-01", "2002-01-01", "2002-06-01", "2003-01-01", "2003-06-01")),
               limits = as.Date(c(NA, "2003-11-01")), expand = c(0.01,0.01)) +
  scale_y_continuous(limits = c(-0.35, 0.35), expand = c(0,0)) +
  geom_vline(data = data.frame(sol = as.Date(c("2001-06-21", "2002-06-21"))),
             aes(xintercept = sol), linetype = "dashed") +
  theme_bw() +
  theme(legend.position = 'none',
        panel.grid.minor = element_blank()) +
  labs(y = "Spearman corr. (alternative)", x = "Time")
```

Warning: Removed 74 rows containing non-finite outside the scale range
('stat_link2()'').

Warning: Removed 74 rows containing non-finite outside the scale range
('stat_difference()'').



Looking at day length effect

```

# Computing day-length

## we need latitude
mwcov <- mwcov %>%
  dplyr::left_join(unique(journe_d[c("Latitude", "sitenewname")]),
                  by = c("site" = "sitenewname")) %>%
  dplyr::mutate(lat = Latitude) %>% dplyr::select(-Latitude)

doy <- mwcov$doy
phi <- (mwcov$lat*pi)/180 # in radian

## solar declination
delta <- 0.409 * sin(2 * pi/365 * doy - 1.39)

## solar hour angle
aux <- -tan(phi) * tan(delta)
# (handle particular cases)
aux[aux > 1] <- 1
aux[aux < -1] <- -1
wo <- acos(-tan(phi)*tan(delta))

## daylength

```

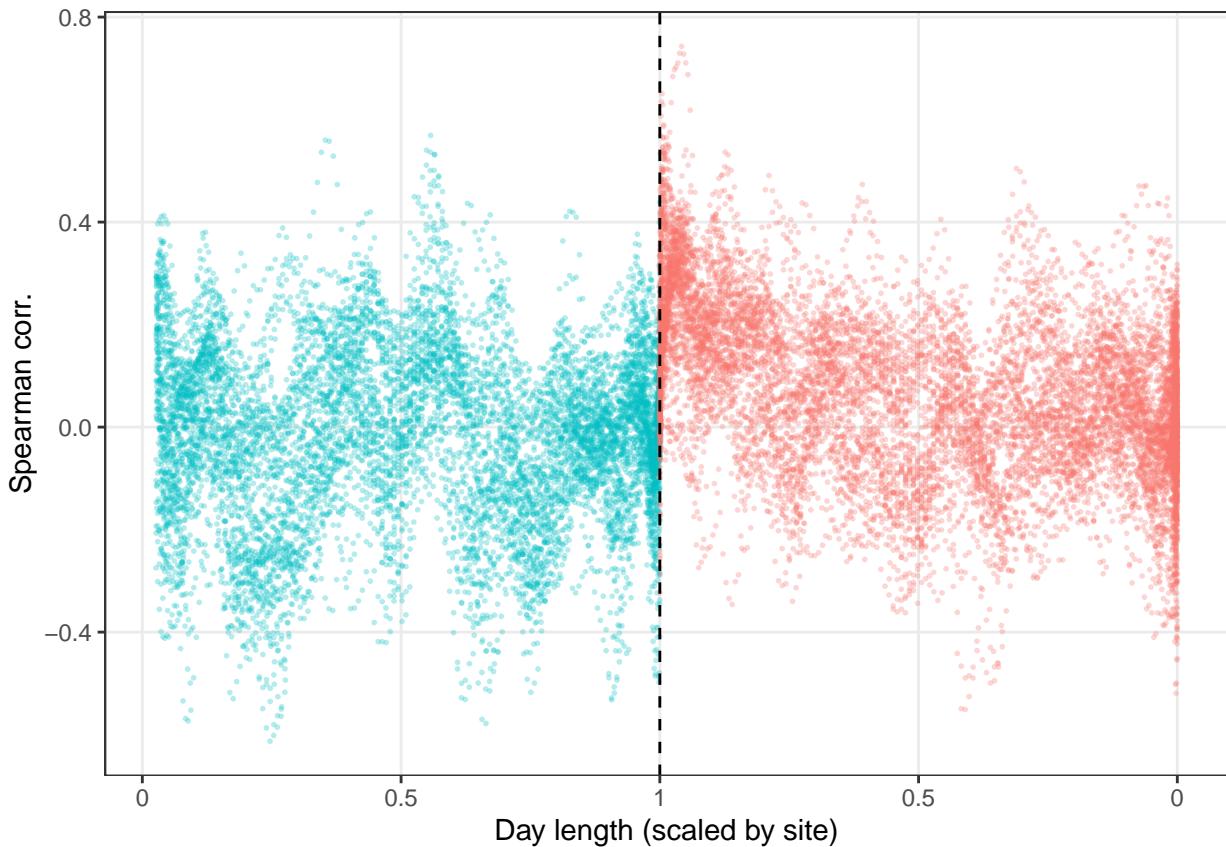
```

dl <- 24/pi*wo
mwcov$dl <- dl

mwcov <- mwcov %>%
  dplyr::group_by(site) %>%
  dplyr::mutate(dlnorm = (dl-min(dl))/(max(dl)-min(dl))) # "for each site, day length was normalized to

mwcov %>%
  dplyr::filter(name == "cor1") %>%
  dplyr::mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)) %>%
  ggplot(aes(x=dlnorm, y = value, color = doy < 172)) +
  # geom_smooth(method = 'loess') +
  geom_point(alpha = 0.3, shape=16, size = 1, stroke = 0) +
  geom_vline(data = data.frame(sol = 0),
             aes(xintercept = sol), linetype = "dashed") +
  theme_bw() +
  theme(legend.position = 'none',
        panel.grid.minor = element_blank()) +
  labs(y = "Spearman corr.", x = "Day length (scaled by site)") +
  scale_x_continuous(breaks = c(-1,-0.5,0,0.5,1), labels = c(0,0.5,1,0.5,0))

```



With the “alternative” way of computing correlation (moving window temperature):

```
# Computing day-length
```

```

## we need latitude
mwcov_alt <- mwcov_alt %>%
  dplyr::left_join(unique(journe_d[c("Latitude", "sitenewname")]),
    by = c("site" = "sitenewname")) %>%
  dplyr::mutate(lat = Latitude) %>% dplyr::select(-Latitude)

doy <- mwcov_alt$doy
phi <- (mwcov_alt$lat*pi)/180 # in radian

## solar declination
delta <- 0.409 * sin(2 * pi/365 * doy - 1.39)

## solar hour angle
aux <- -tan(phi) * tan(delta)
# (handle particular cases)
aux[aux > 1] <- 1
aux[aux < -1] <- -1
wo <- acos(-tan(phi)*tan(delta))

## daylength
dl <- 24/pi*wo
mwcov_alt$dl <- dl

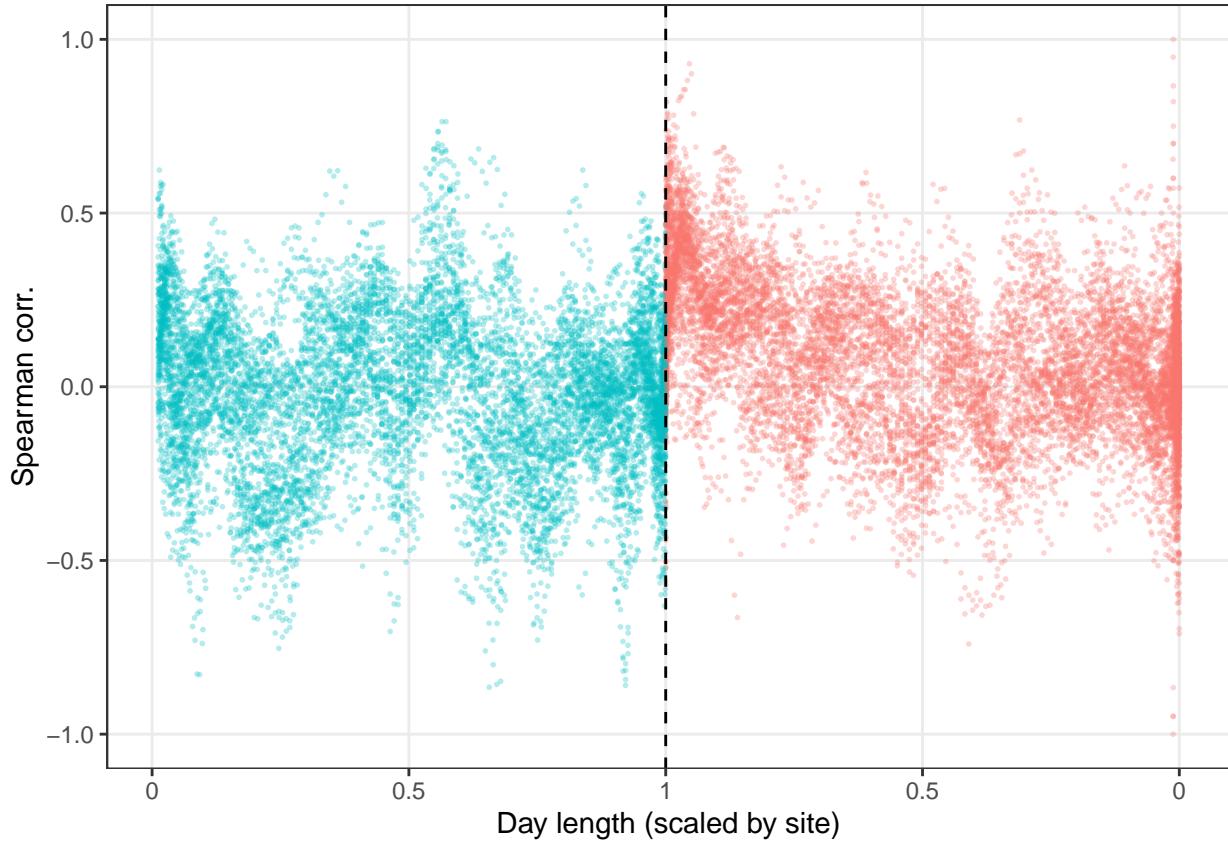
mwcov_alt <- mwcov_alt %>%
  dplyr::group_by(site) %>%
  dplyr::mutate(dlnorm = (dl-min(dl))/(max(dl)-min(dl))) # "for each site, day length was normalized to

mwcov_alt$dl <- dl

mwcov_alt <- mwcov_alt %>%
  dplyr::group_by(site) %>%
  dplyr::mutate(dlnorm = (dl-min(dl))/(max(dl)-min(dl))) # "for each site, day length was normalized to

mwcov_alt %>%
  dplyr::filter(name == "cor1") %>%
  dplyr::mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)) %>%
  ggplot(aes(x=dlnorm, y = value, color = doy < 172)) +
  # geom_smooth(method = 'loess') +
  geom_point(alpha = 0.3, shape=16, size = 1, stroke = 0) +
  geom_vline(data = data.frame(sol = 0),
             aes(xintercept = sol), linetype = "dashed") +
  theme_bw() +
  theme(legend.position = 'none',
        panel.grid.minor = element_blank()) +
  labs(y = "Spearman corr.", x = "Day length (scaled by site)") +
  scale_x_continuous(breaks = c(-1,-0.5,0,0.5,1), labels = c(0,0.5,1,0.5,0))

```



Looking at GAM-related stuff

See if we can fit a similar GAM than Journe et al.

```

data_fit <- mwcor_alt %>%
  dplyr::filter(name == "cor1") %>%
  dplyr::mutate(
    source = strsplit(site, split='\_')[[1]][1],
    prepost = factor(doy < 172)) %>%
  # "we rescaled the Spearman correlations to the 0-1 range. The scaling was done for each site separately"
  dplyr::group_by(site) %>%
  dplyr::mutate(
    maxv = max(value), minv = min(value),
    valuenorm = (value-minv)/(maxv-minv))

# dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)

journe_gam <- mgcv::gam(
  formula = valuenorm ~ prepost + s(dlnorm, by = prepost),
  data = data_fit,
  family=betar(link="logit", eps=.Machine$double.eps*1e6),
  method="REML")

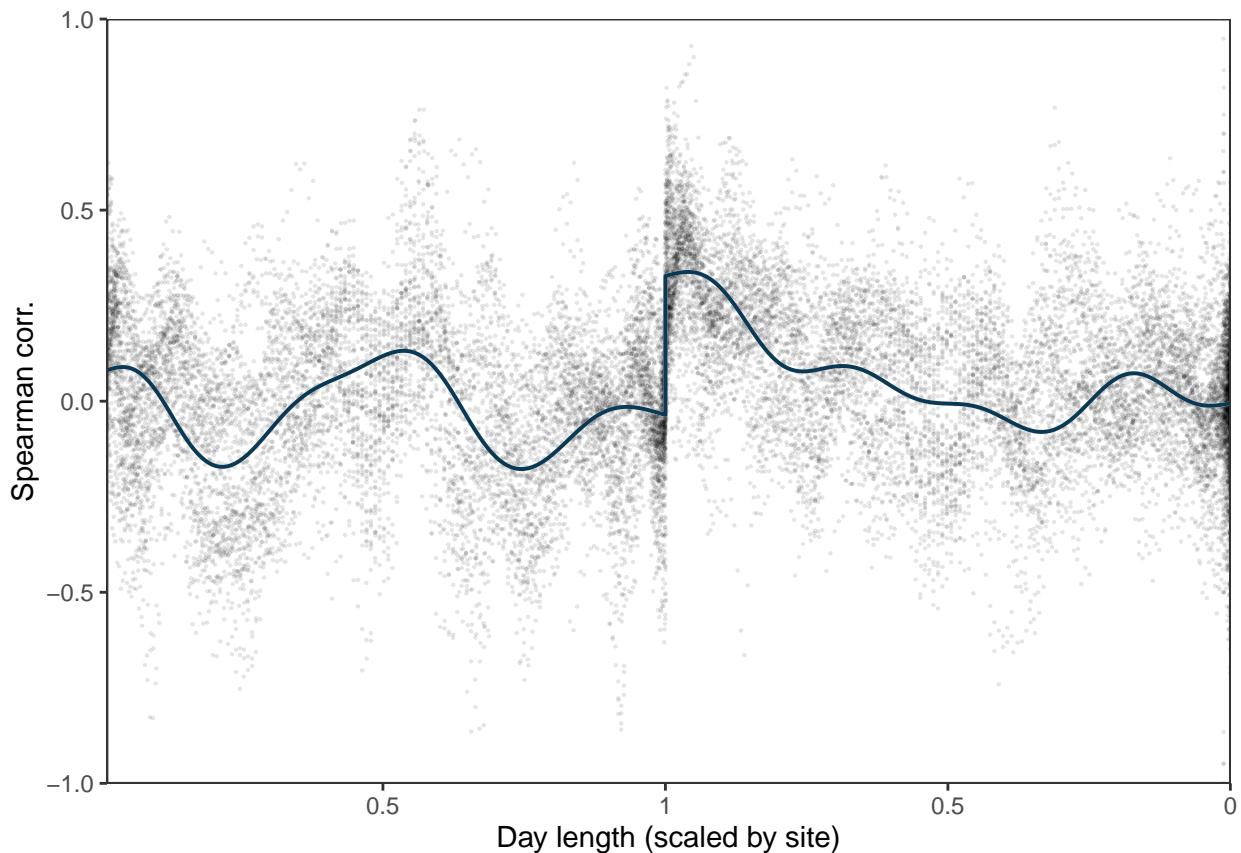
pred_gam <- data.frame(
  dlnorm = data_fit$dlnorm,
  
```

```

doy = data_fit$doy,
maxv = data_fit$maxv,
minv = data_fit$minv,
valuenorm = data_fit$valuenorm) %>%
  mutate(pred = predict(journe_gam, type="response", se.fit = FALSE)* (maxv-minv) + minv) %>%
  group_by(doy) %>%
  summarise(pred = mean(pred), dlnorm = mean(dlnorm))

ggplot() +
  geom_point(
    data = data_fit %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = dlnorm, y = value), size = 0.1, alpha = 0.1) +
  geom_line(
    data = pred_gam %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = dlnorm, y = pred), color = "white", linewidth = 0.7) +
  geom_line(
    data = pred_gam %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = dlnorm, y = pred), color = "#0B3954", linewidth = 0.7) +
  theme_bw() +
  theme(legend.position = 'none',
        panel.grid.minor = element_blank(), panel.grid.major = element_blank()) +
  labs(y = "Spearman corr.", x = "Day length (scaled by site)") +
  scale_x_continuous(breaks = c(-1,-0.5,0,0.5,1), labels = c(0,0.5,1,0.5,0)) +
  coord_cartesian(expand = FALSE)

```

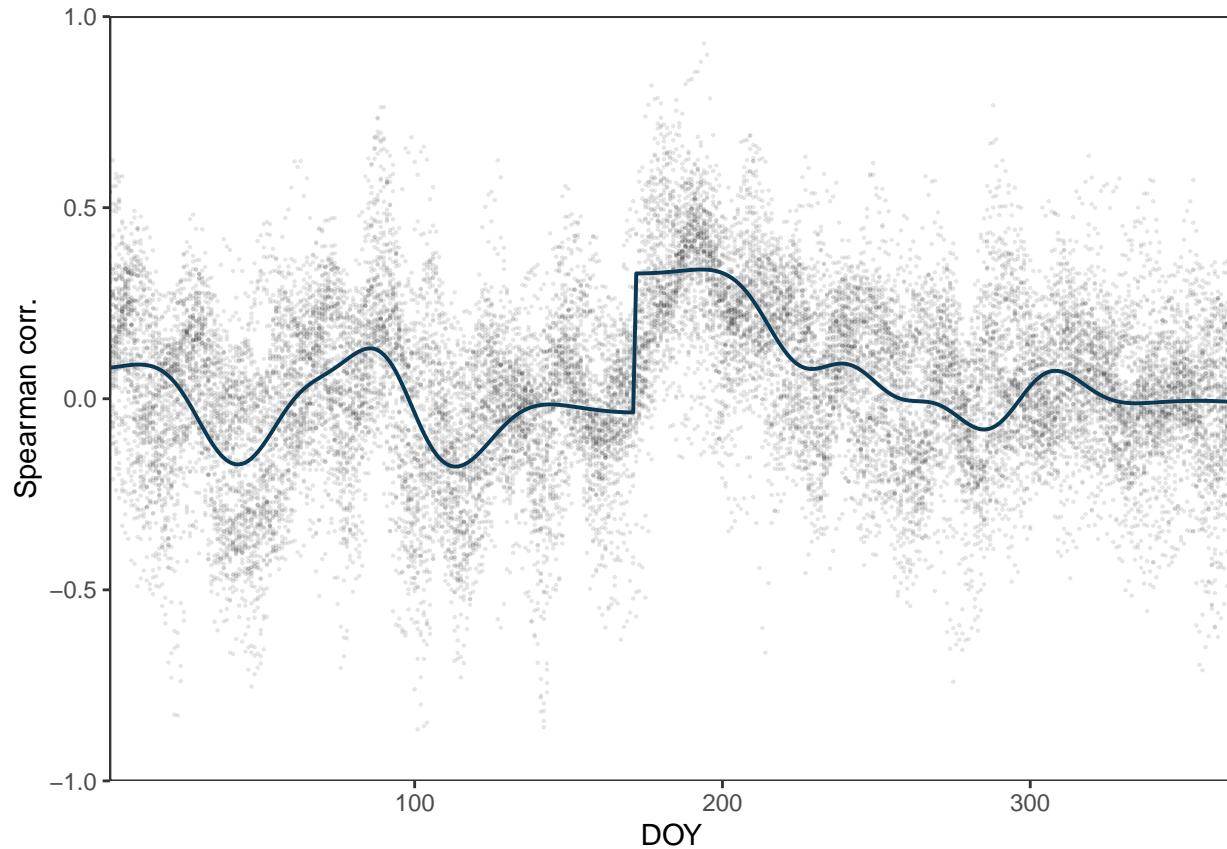


If we plot according to DOY (to avoid squeezing the data around the solstice...), the big step we assume at

soltice seems much more bizarre.

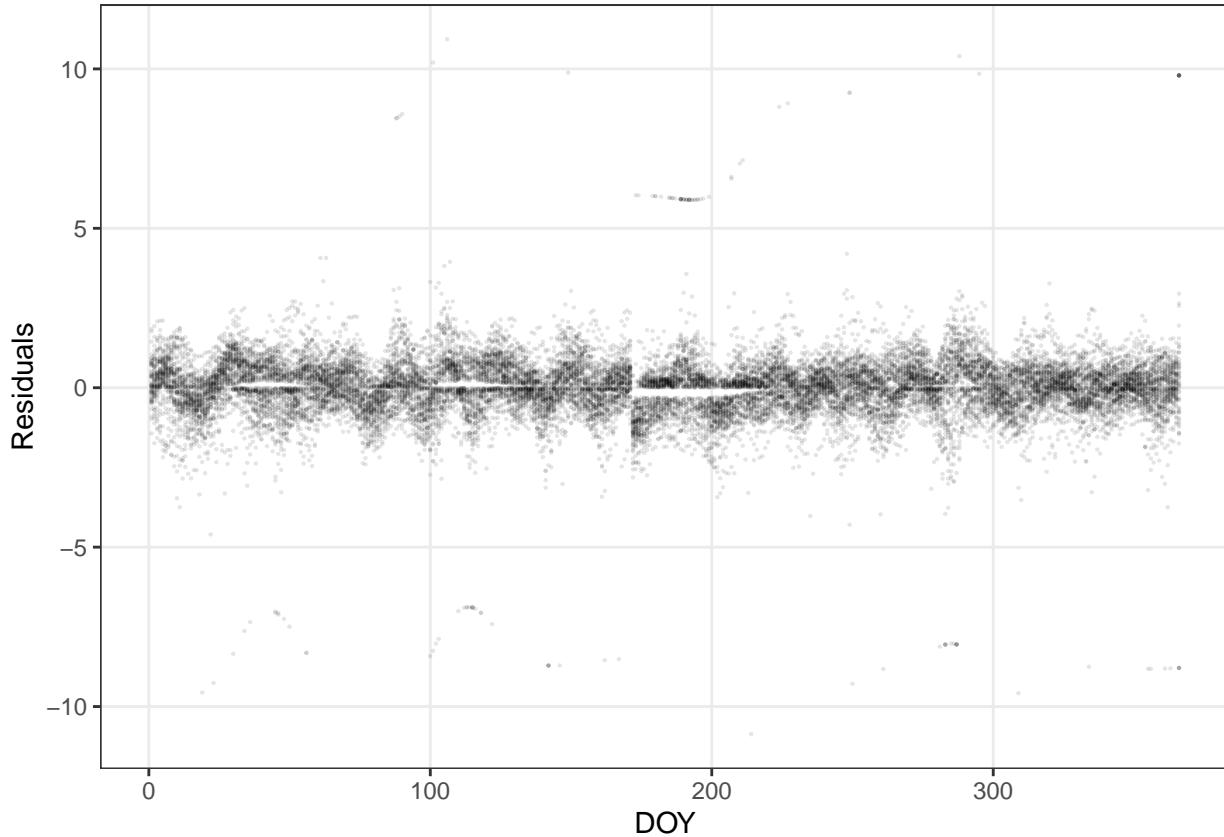
```
# pred_gam <- data.frame(
#   dlnorm = data_fit$dlnorm,
#   doy = data_fit$doy,
#   maxv = data_fit$maxv,
#   minv = data_fit$minv,
#   valuenorm = data_fit$valuenorm) %>%
#   mutate(pred = predict(journe_gam, type="response", se.fit = FALSE)* (maxv-minv) + minv) %>%
#   group_by(doy) %>%
#   summarise(pred = mean(pred), dlnorm = mean(dlnorm))

ggplot() +
  geom_point(
    data = data_fit %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = doy, y = value), size = 0.1, alpha = 0.1) +
  geom_line(
    data = pred_gam %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = doy, y = pred), color = "white", linewidth = 0.7) +
  geom_line(
    data = pred_gam %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = doy, y = pred), color = "#0B3954", linewidth = 0.7) +
  theme_bw() +
  theme(legend.position = 'none',
        panel.grid.minor = element_blank(), panel.grid.major = element_blank()) +
  labs(y = "Spearman corr.", x = "DOY") +
  coord_cartesian(expand = FALSE)
```



And if we look at the residuals/time... We find this wavy pattern.

```
data.frame(
  doy = data_fit$doy,
  res = residuals(journe_gam)
) %>%
  ggplot(aes(x = doy)) +
  geom_point(aes(y = res), size = 0.1, alpha = 0.1) +
  theme_bw() +
  theme(legend.position = 'none',
        panel.grid.minor = element_blank()) +
  labs(y = "Residuals", x = "DOY")
```



We may want to try to fit a GAM with AR1 correlation? Still assuming a priori a solstice effect as Journe et al. Still plotting against DOY to avoid squeezing stuff around summer solstice.

```

journe_gam_AR1 <- mgcv::gamm(
  formula = value ~ prepost + s(dlnorm, by = prepost),
  correlation = corAR1(form = ~ doy | site),
  data = data_fit,
  method="REML")

pred_gam_AR1 <- data.frame(
  dlnorm = data_fit$dlnorm,
  doy = data_fit$doy,
  maxv = data_fit$maxv,
  minv = data_fit$minv,
  value = data_fit$value)%>%
  mutate(pred = predict(journe_gam_AR1$gam, type="response", se.fit = FALSE)) %>%
  group_by(doy) %>%
  summarise(pred = mean(pred), dlnorm = mean(dlnorm))

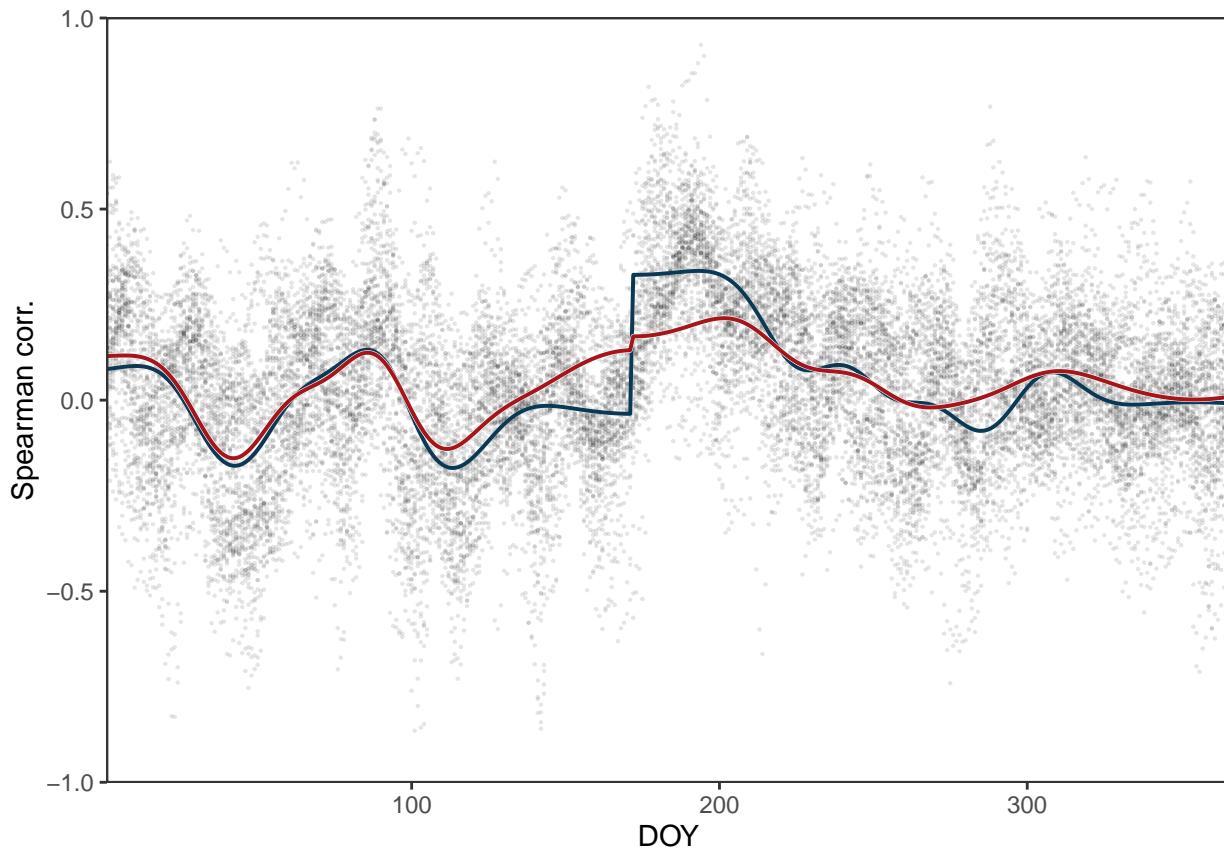
ggplot() +
  geom_point(
    data = data_fit %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = doy, y = value), size = 0.1, alpha = 0.1) +
  geom_line(
    data = pred_gam %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = doy, y = pred), color = "white", linewidth = 0.7) +

```

```

geom_line(
  data = pred_gam %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
  aes(x = doy, y = pred), color = "#0B3954", linewidth = 0.7) +
geom_line(
  data = pred_gam_AR1 %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
  aes(x = doy, y = pred), color = "white", linewidth = 0.9) +
geom_line(
  data = pred_gam_AR1%>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)) ,
  aes(x = doy, y = pred), color = "#A5151C", linewidth = 0.7) +
theme_bw() +
theme(legend.position = 'none',
      panel.grid.minor = element_blank(), panel.grid.major = element_blank()) +
labs(y = "Spearman corr.", x = "DOY") +
coord_cartesian(expand = FALSE)

```



What if we remove the solstice-thingy variable?

```

# data_fit$prepost <- factor(data_fit$doy < 195)

basic_gam_AR1 <- mgcv::gamm(
  formula = value ~ s(dlnorm),
  correlation = corAR1(form = ~ doy | site),
  data = data_fit %>% dplyr::mutate(site = factor(site)),
  method="REML")

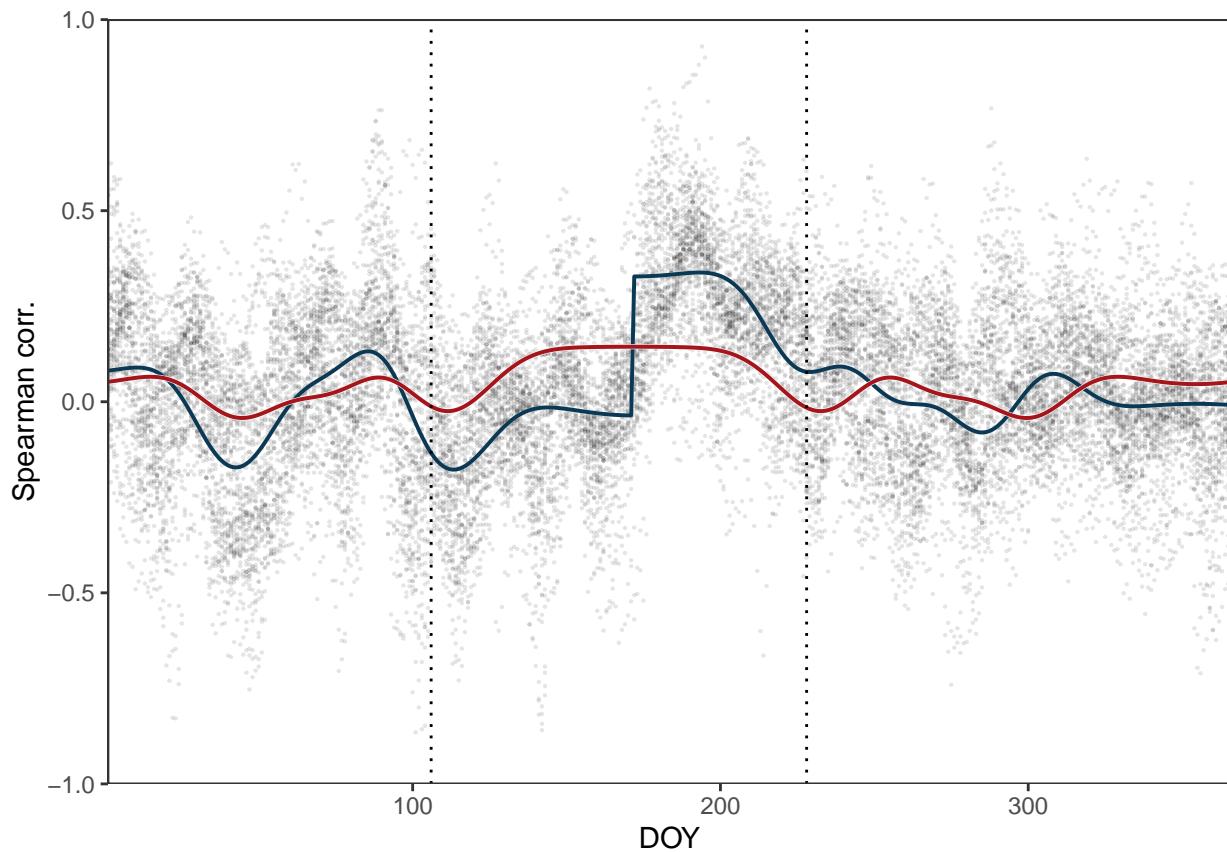
```

```

pred_gam_AR1 <- data.frame(
  dlnorm = data_fit$dlnorm,
  doy = data_fit$doy,
  maxv = data_fit$maxv,
  minv = data_fit$minv,
  value = data_fit$value) %>%
  mutate(pred = predict(basic_gam_AR1$gam, type="response", se.fit = FALSE)) %>%
  group_by(doy) %>%
  summarise(pred = mean(pred), dlnorm = mean(dlnorm))

ggplot() +
  geom_point(
    data = data_fit %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = doy, y = value), size = 0.1, alpha = 0.1) +
  geom_line(
    data = pred_gam %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = doy, y = pred), color = "white", linewidth = 0.7) +
  geom_line(
    data = pred_gam %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = doy, y = pred), color = "#0B3954", linewidth = 0.7) +
  geom_line(
    data = pred_gam_AR1 %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = doy, y = pred), color = "white", linewidth = 0.9) +
  geom_line(
    data = pred_gam_AR1 %>% mutate(dlnorm = ifelse(doy < 172, -1+dlnorm, 1-dlnorm)),
    aes(x = doy, y = pred), color = "#A5151C", linewidth = 0.7) +
  theme_bw() +
  theme(legend.position = 'none',
        panel.grid.minor = element_blank(), panel.grid.major = element_blank()) +
  labs(y = "Spearman corr.", x = "DOY") +
  coord_cartesian(expand = FALSE) +
  geom_vline(data = data.frame(doyp = c(106,228)),
             aes(xintercept = doyp), linetype = "dotted")

```



```
basic_gam_noAR <- mgcv::gamm(
  formula = value ~ prepost + s(dlnorm, by = prepost),
  data = data_fit,
  method="REML")

data.frame(
  doy = rep(data_fit$doy,2),
  res = c(resid(basic_gam_noAR$lme, type = "normalized"), resid(basic_gam_AR1$lme, type = "normalized")),
  type = c(rep("noAR - solstice", length(data_fit$doy)), rep("AR1 - no solstice", length(data_fit$doy)))
) %>%
  ggplot(aes(x = doy)) +
  facet_wrap(~ type) +
  geom_point(aes(y = res), size = 0.2, alpha = 0.2) +
  theme_bw() +
  theme(legend.position = 'none',
        panel.grid.minor = element_blank()) +
  labs(y = "Residuals (scaled)", x = "DOY") +
  coord_cartesian(ylim = c(-5,5))
```

