Explore Trends of Snow Cover Related Indicators

AJ Perez-Luque (@ajpelu); FJ Bonet; J Herrero and R. Perez-Perez

2016 March

Prepare Data

1. Read data of Mann-Kendal Sen-Slope for each pixels and snow-cover related indicators (scd, scod, scmd, scmc).
2. Read data of topographic variable:
   - Convert radian to deg.
   - Create categorical variable for elevation (250 m)
   - Classify aspect into 8 categories
3. Read data from hydrological basin
4. Read spatial data:
   - Select only centroides of interest
5. Create two dataframes:
   - **Full Dataframe** with all variables and all pixels (`fulldf`)
   - **Filter dataframe** with all variables and filter by pixels above 1250 *m.a.s.l.* (`fulldf1250`)

```
# Trend analysis data
# Define name of indicators (see variables names)
indicadores <- c("scd", "scod", "scmd", "scmc")

# Loop to read files
for (j in indicadores){
  aux <- read.csv(file=paste(di, "/data/derived/", j, ".csv", sep= ""),
              header = TRUE,
              sep = ',')
  assign(j, aux)
}
# --

# Define pixels of interes
pixels_interes <- scd$nie_malla_modi_id

# Read Topographic data
rawtopo <- read.csv(file=paste(di, "/data/topo_nie_malla_modis.csv", sep=""),
                    header=TRUE,
                    sep = ",")

# function to convert radian to degree
rad2deg <- function(rad) {(rad * 180) / (pi)}


topo <- rawtopo %>%
  filter(id %in% pixels_interes) %>%
  mutate(nie_malla_modi_id = id,
         slope50mean_deg = rad2deg(slope50mean),
```

```r
         slope50median_deg = rad2deg(slope50median),
         aspect50mean_deg = rad2deg(aspect50mean),
         aspect50median_deg = rad2deg(aspect50median)) %>%
  dplyr::select(nie_malla_modi_id, dem50mean, dem50median, slope50mean_deg,
                slope50median_deg, aspect50mean_deg, aspect50median_deg)

## Create interval variables (250 m) for dem; and classify aspect into 8 categories
topo <- topo %>%
  mutate(dem50mean_group = cut(dem50mean,
                         breaks = seq(from=0, to=3500, by=250),
                         labels = c("0-250", "251-500","501-750","751-1000",
                                    "1001-1250","1251-1500","1501-1750","1751-2000",
                                    "2001-2250","2251-2500","2501-2750", "2751-3000",
                                    "3001-3250", "3251-3500")),
         aspect50mean_deg_group = cut(aspect50mean_deg,
                                      breaks= c(22.5, 67.5, 112.5, 157.5, 202.5, 247.5, 292.5, 337.5, 35
                                      labels = c("N", "NE", "E", "SE", "S", "SW", "W", "NW")))


# --

# Read spatial data and Get lat/long
centroides <- rgdal::readOGR(dsn=paste(di, "/data/geoinfo", sep=""),
                             layer = "centroides_selected", verbose = FALSE)
# Select only attributes of interest and rename them
centroides <- centroides[c("id")]

# Create lat/lng by id
xycentroides <- cbind(centroides@data, coordinates(centroides))
names(xycentroides) <- c("nie_malla_modi_id", "lon","lat")

xycentroides <- filter(xycentroides, nie_malla_modi_id %in% pixels_interes)
# --

# Hydrological basin
basin <- read.csv(file=paste(di, "/data/derived/pixel_region.csv", sep=""),
                    header=TRUE,
                    sep = ",")
# --


# Create un dataframe con todos los datos
fulldf <- topo %>%
  inner_join(scod,  by=c("nie_malla_modi_id")) %>%
  inner_join(scd,  by=c("nie_malla_modi_id")) %>%
  inner_join(scmd,  by=c("nie_malla_modi_id")) %>%
  inner_join(scmc,  by=c("nie_malla_modi_id")) %>%
  inner_join(xycentroides, by="nie_malla_modi_id") %>%
  inner_join(basin, by="nie_malla_modi_id")

# Create subset of pixels above 1250
fulldf1250 <- fulldf %>%
  filter(dem50mean > 1250)
```

Explore Snow-Cover trends by basin

We explore the pattern of snow-cover indicators trends by hydrological basin.

```
## Summary statistics
misvariables<- c('tau_scd', 'sen_slope_scd', 'tau_scod', 'sen_slope_scod',
                 'tau_scmd', 'sen_slope_scmd', 'tau_scmc', 'sen_slope_scmc')
variable_agrupa <- 'basin_name'
prefijo  <- 'basin_'
df <- fulldf1250

stats_basin <- data.frame()

for (i in misvariables){
  vnames <- c(variable_agrupa, i)
  aux <- df %>%
    dplyr::select(one_of(vnames)) %>%
    mutate_(vinterest = i) %>%
    group_by_(.dots=variable_agrupa) %>%
    summarise(mean=mean(vinterest),
              sd = sd(vinterest),
              se = sd / sqrt (length(vinterest)))
  aux <- aux %>%
    mutate(variable = i)

  stats_basin <- rbind(stats_basin, aux)
  # assign(paste0(prefijo,i), aux)

}
```

Snow cover duration

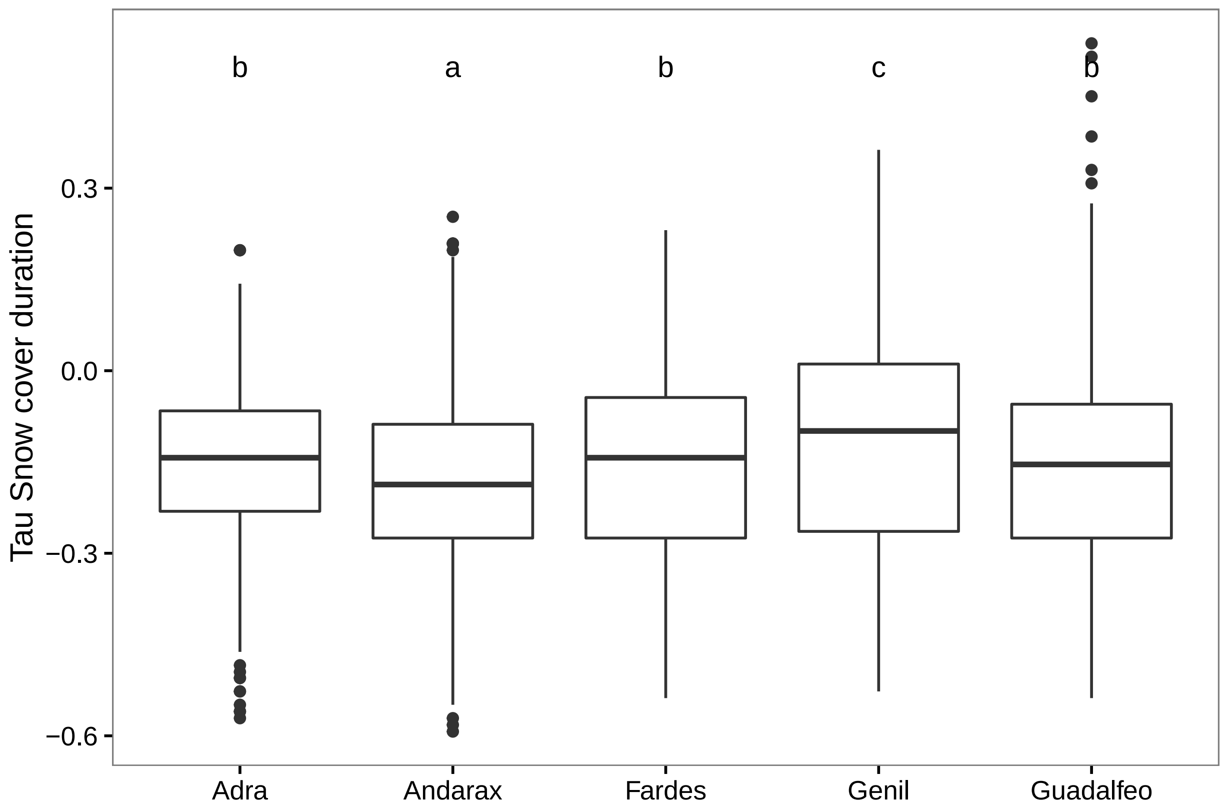
```
df_letter_aux <- data.frame()
df <- fulldf1250
```

```
# Tau scd
# ANOVA
variable <- 'tau_scd'
my_ylab <- 'Tau Snow cover duration'
mod <- aov(tau_scd ~ basin_name, data=df)
pander(tidy(mod), caption= my_ylab)
```

Table 1: Tau Snow cover duration

term	df	sumsq	meansq	statistic	p.value
basin_name	4	2.326	0.5814	25.07	1.239e-20
Residuals	6385	148.1	0.02319	NA	NA

```r
## Multiple comparisons
tuk <- glht(mod, linfct = mcp(basin_name = "Tukey"))
# Convert comparisons into letters
df_letter <- fortify(cld(tuk)) %>%
  transmute(basin_name = as.factor(lhs),
         tukey_basin_name = letters) %>%
  mutate(variable = variable)

df_letter_aux <- rbind(df_letter_aux, df_letter)

## ANOVA plots
ggplot(df, aes_string(x='basin_name', y=variable)) +
  geom_boxplot() +
  geom_text(data=df_letter, aes(label = tukey_basin_name, x=basin_name, y=0.5),
           position = position_dodge(width=0.9)) +
  theme_bw() + mythemeggplot +
  ylab(my_ylab) + xlab('')
```

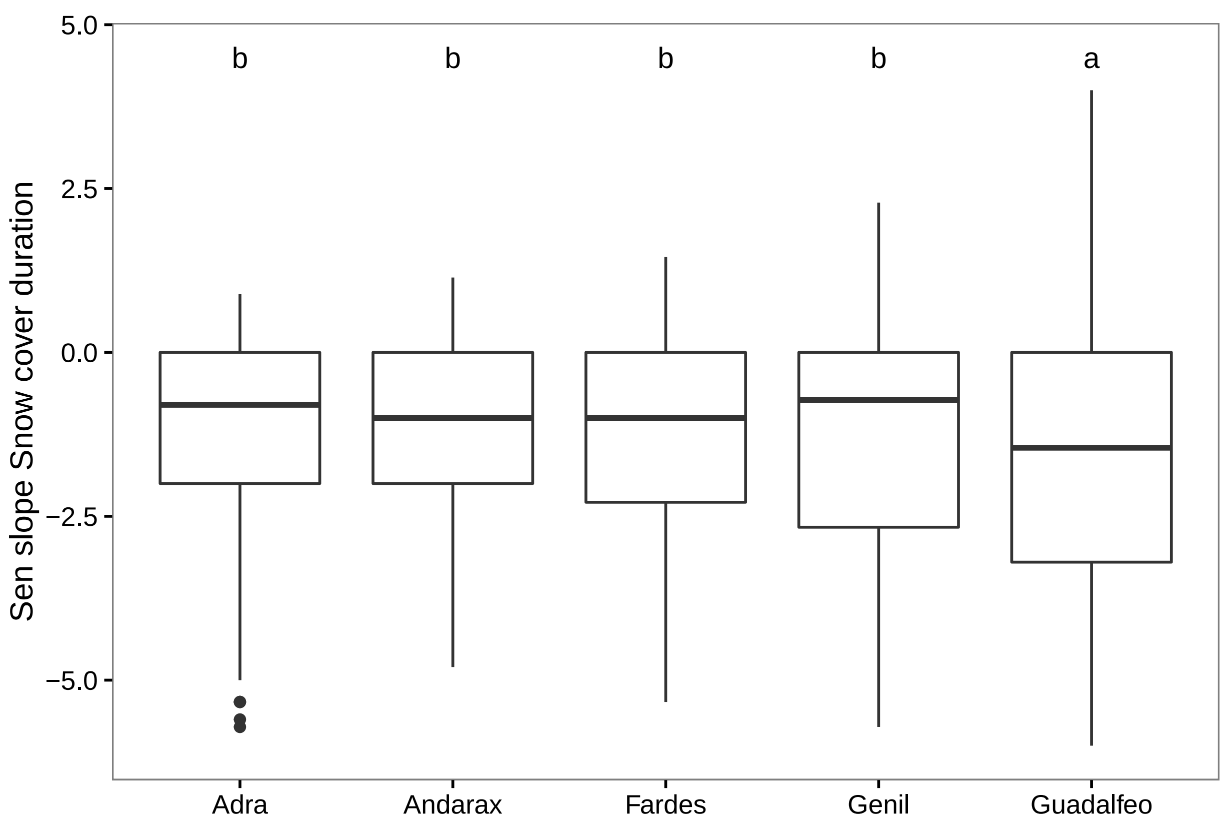



```r
# Sen scd
# ANOVA
variable <- 'sen_slope_scd'
my_ylab <- 'Sen slope Snow cover duration'
mod <- aov(sen_slope_scd ~ basin_name, data=df)
pander(tidy(mod), caption= my_ylab)
```

Table 2: Sen slope Snow cover duration

term	df	sumsq	meansq	statistic	p.value
basin_name	4	369.5	92.38	40.67	1.056e-33
Residuals	6385	14503	2.271	NA	NA

```
## Multiple comparisons
tuk <- glht(mod, linfct = mcp(basin_name = "Tukey"))
# Convert comparisons into letters
df_letter <- fortify(cld(tuk)) %>%
  transmute(basin_name = as.factor(lhs),
         tukey_basin_name = letters) %>%
  mutate(variable = variable)

df_letter_aux <- rbind(df_letter_aux, df_letter)

## ANOVA plots
ggplot(df, aes_string(x='basin_name', y=variable)) +
  geom_boxplot() +
  geom_text(data=df_letter, aes(label = tukey_basin_name, x=basin_name, y=4.5),
           position = position_dodge(width=0.9)) +
  theme_bw() + mythemeggplot +
  ylab(my_ylab) + xlab('')
```



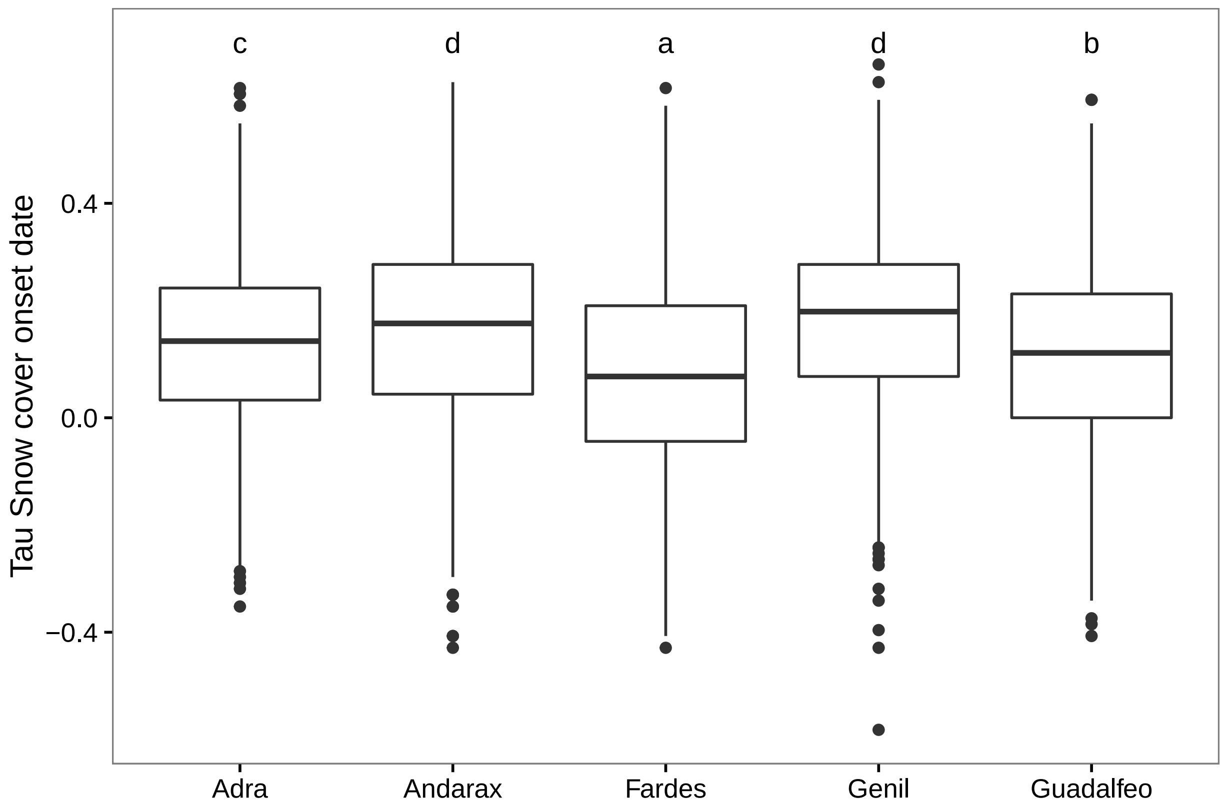
Snow cover onset date

```r
# Tau scod
# ANOVA
variable <- 'tau_scod'
my_ylab <- 'Tau Snow cover onset date'
mod <- aov(tau_scod ~ basin_name, data=df)
pander(tidy(mod), caption= my_ylab)
```

Table 3: Tau Snow cover onset date

term	df	sumsq	meansq	statistic	p.value
basin_name	4	6.29	1.573	55.62	3.55e-46
Residuals	6385	180.5	0.02827	NA	NA

```r
## Multiple comparisons
tuk <- glht(mod, linfct = mcp(basin_name = "Tukey"))
# Convert comparisons into letters
df_letter <- fortify(cld(tuk)) %>%
  transmute(basin_name = as.factor(lhs),
         tukey_basin_name = letters) %>%
  mutate(variable = variable)

df_letter_aux <- rbind(df_letter_aux, df_letter)

## ANOVA plots
ggplot(df, aes_string(x='basin_name', y=variable)) +
  geom_boxplot() +
  geom_text(data=df_letter, aes(label = tukey_basin_name, x=basin_name, y=0.7),
           position = position_dodge(width=0.9)) +
  theme_bw() + mythemeggplot +
  ylab(my_ylab) + xlab('')
```



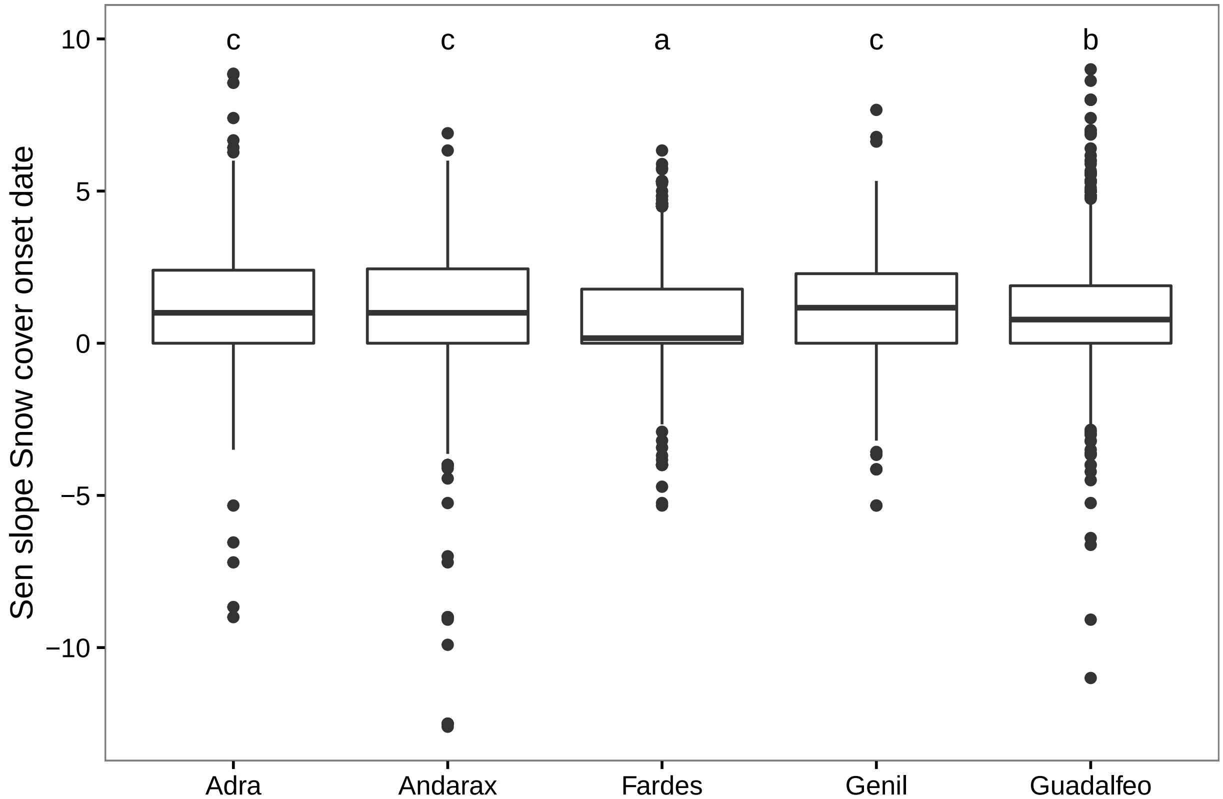

```r
# Sen scod
# ANOVA
variable <- 'sen_slope_scod'
my_ylab <- 'Sen slope Snow cover onset date'
mod <- aov(sen_slope_scod ~ basin_name, data=df)
pander(tidy(mod), caption= my_ylab)
```

Table 4: Sen slope Snow cover onset date

term	df	sumsq	meansq	statistic	p.value
basin_name	4	334	83.49	26.76	4.745e-22
Residuals	6385	19919	3.12	NA	NA

```r
## Multiple comparisons
tuk <- glht(mod, linfct = mcp(basin_name = "Tukey"))
# Convert comparisons into letters
df_letter <- fortify(cld(tuk)) %>%
  transmute(basin_name = as.factor(lhs),
         tukey_basin_name = letters) %>%
  mutate(variable = variable)

df_letter_aux <- rbind(df_letter_aux, df_letter)

## ANOVA plots
ggplot(df, aes_string(x='basin_name', y=variable)) +
  geom_boxplot() +
```

```
  geom_text(data=df_letter, aes(label = tukey_basin_name, x=basin_name, y=10),
            position = position_dodge(width=0.9)) +
  theme_bw() + mythemeggplot +
  ylab(my_ylab) + xlab('')
```



Snow cover melting date

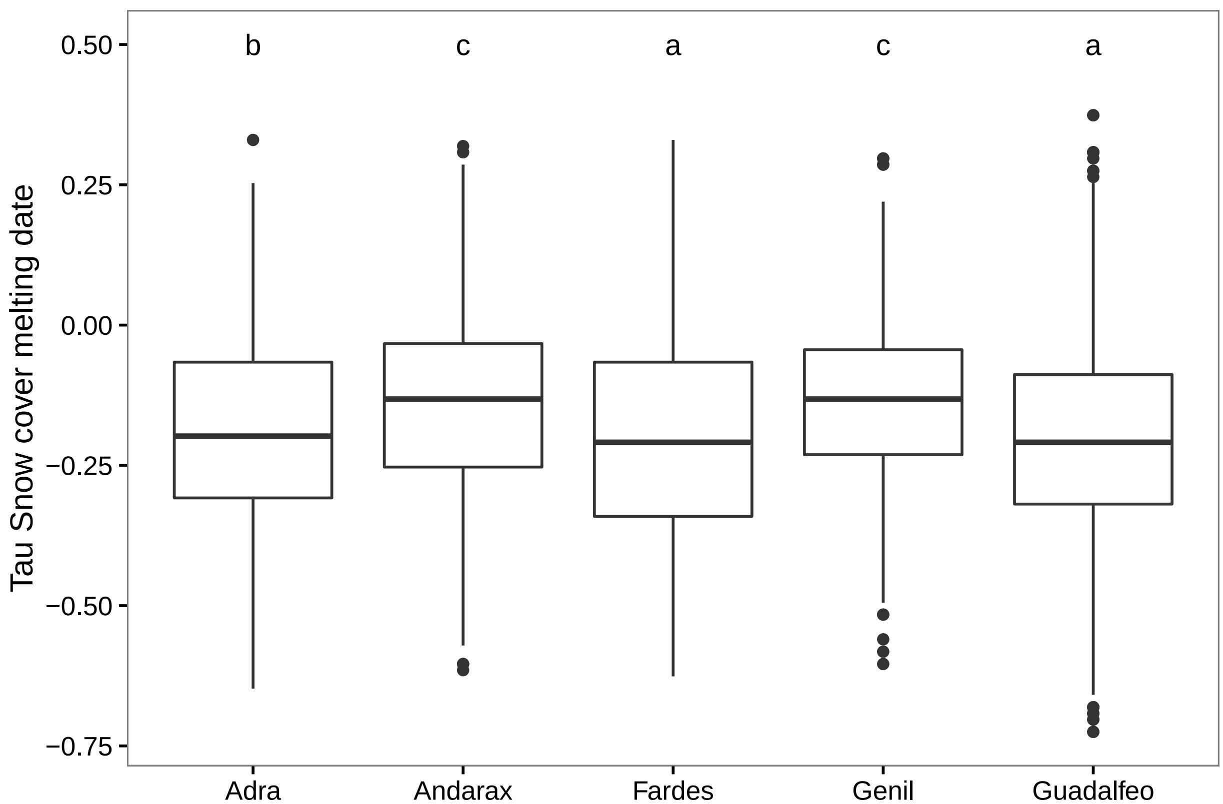
```
# Tau scmd
# ANOVA
variable <- 'tau_scmd'
my_ylab <- 'Tau Snow cover melting date'
mod <- aov(tau_scmd ~ basin_name, data=df)
pander(tidy(mod), caption= my_ylab)
```

Table 5: Tau Snow cover melting date

term	df	sumsq	meansq	statistic	p.value
basin_name	4	5.153	1.288	47.14	4.112e-39
Residuals	6385	174.5	0.02733	NA	NA

```
## Multiple comparisons
tuk <- glht(mod, linfct = mcp(basin_name = "Tukey"))
```

```r
# Convert comparisons into letters
df_letter <- fortify(cld(tuk)) %>%
  transmute(basin_name = as.factor(lhs),
         tukey_basin_name = letters) %>%
  mutate(variable = variable)

df_letter_aux <- rbind(df_letter_aux, df_letter)

## ANOVA plots
ggplot(df, aes_string(x='basin_name', y=variable)) +
  geom_boxplot() +
  geom_text(data=df_letter, aes(label = tukey_basin_name, x=basin_name, y=0.5),
           position = position_dodge(width=0.9)) +
  theme_bw() + mythemeggplot +
  ylab(my_ylab) + xlab('')
```

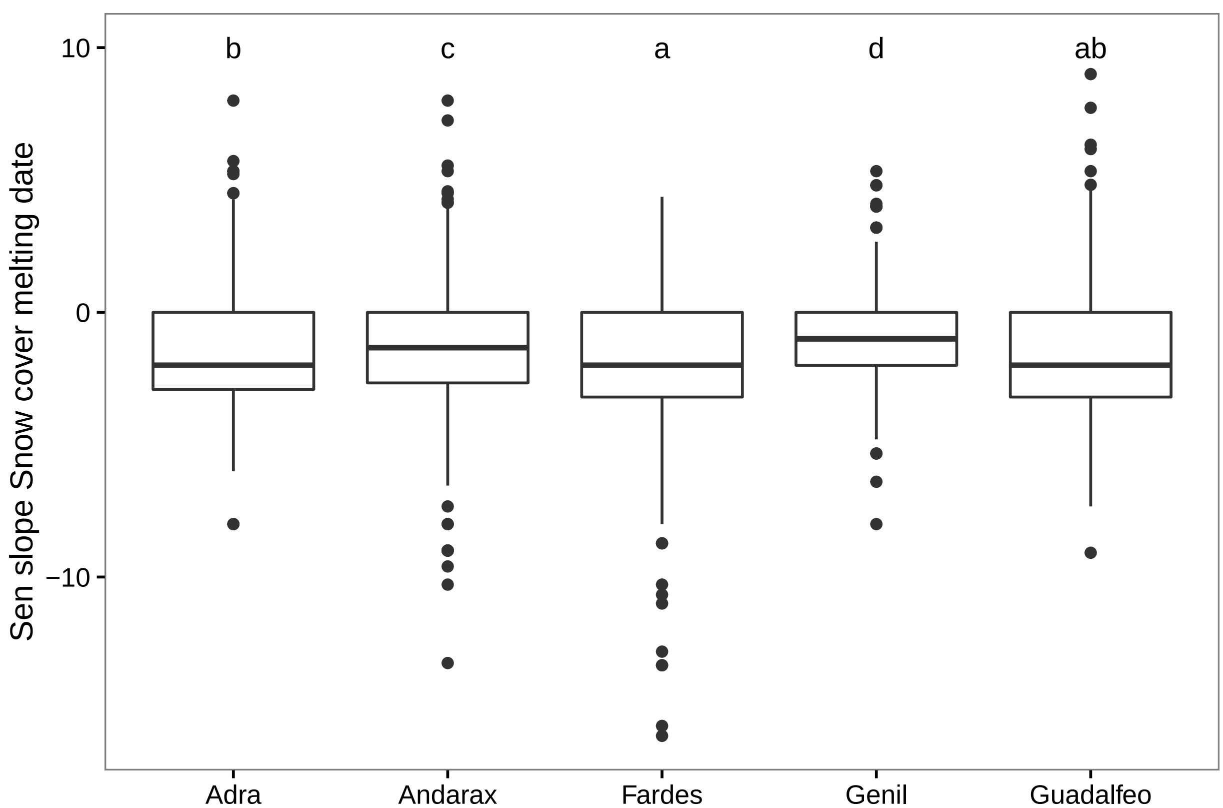



```r
# Sen scmd
# ANOVA
variable <- 'sen_slope_scmd'
my_ylab <- 'Sen slope Snow cover melting date'
mod <- aov(sen_slope_scmd ~ basin_name, data=df)
pander(tidy(mod), caption= my_ylab)
```

Table 6: Sen slope Snow cover melting date

term	df	sumsq	meansq	statistic	p.value
basin_name	4	555.2	138.8	39.15	1.982e-32
Residuals	6385	22638	3.545	NA	NA

```
## Multiple comparisons
tuk <- glht(mod, linfct = mcp(basin_name = "Tukey"))
# Convert comparisons into letters
df_letter <- fortify(cld(tuk)) %>%
  transmute(basin_name = as.factor(lhs),
         tukey_basin_name = letters) %>%
  mutate(variable = variable)

df_letter_aux <- rbind(df_letter_aux, df_letter)

## ANOVA plots
ggplot(df, aes_string(x='basin_name', y=variable)) +
  geom_boxplot() +
  geom_text(data=df_letter, aes(label = tukey_basin_name, x=basin_name, y=10),
           position = position_dodge(width=0.9)) +
  theme_bw() + mythemeggplot +
  ylab(my_ylab) + xlab('')
```



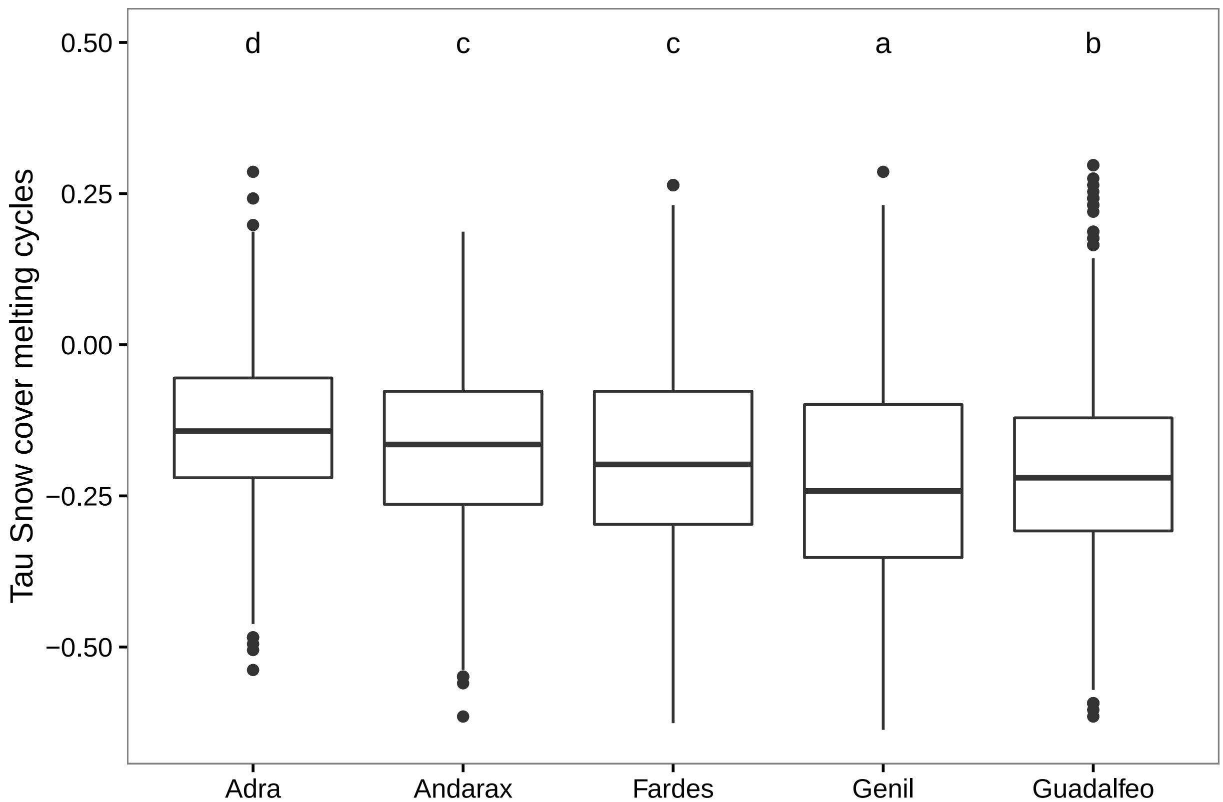
Snow cover melting cycles

```r
# Tau scmc
# ANOVA
variable <- 'tau_scmc'
my_ylab <- 'Tau Snow cover melting cycles'
mod <- aov(tau_scmc ~ basin_name, data=df)
pander(tidy(mod), caption= my_ylab)
```

Table 7: Tau Snow cover melting cycles

term	df	sumsq	meansq	statistic	p.value
basin_name	4	4.83	1.207	54.49	3.053e-45
Residuals	6385	141.5	0.02216	NA	NA

```r
## Multiple comparisons
tuk <- glht(mod, linfct = mcp(basin_name = "Tukey"))
# Convert comparisons into letters
df_letter <- fortify(cld(tuk)) %>%
  transmute(basin_name = as.factor(lhs),
         tukey_basin_name = letters) %>%
  mutate(variable = variable)

df_letter_aux <- rbind(df_letter_aux, df_letter)

## ANOVA plots
ggplot(df, aes_string(x='basin_name', y=variable)) +
  geom_boxplot() +
  geom_text(data=df_letter, aes(label = tukey_basin_name, x=basin_name, y=0.5),
           position = position_dodge(width=0.9)) +
  theme_bw() + mythemeggplot +
  ylab(my_ylab) + xlab('')
```



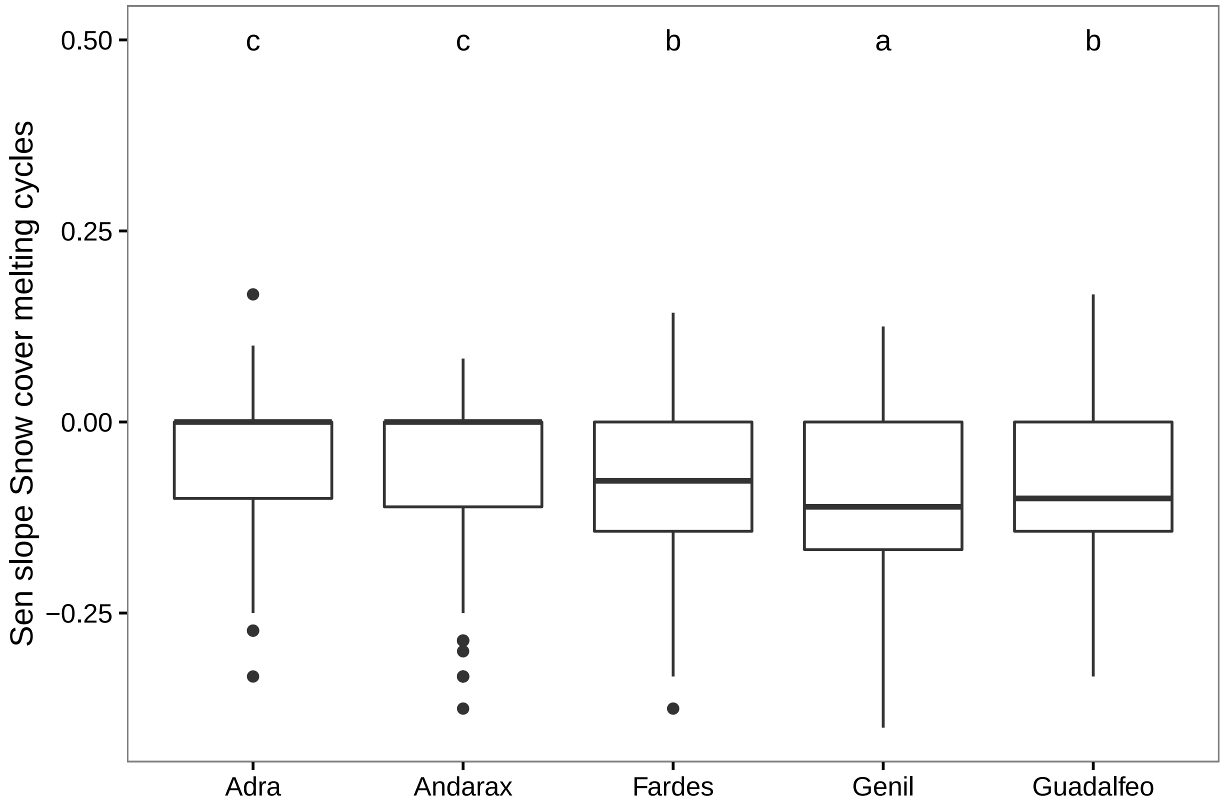

```r
# Sen scmc
# ANOVA
variable <- 'sen_slope_scmc'
my_ylab <- 'Sen slope Snow cover melting cycles'
mod <- aov(sen_slope_scmc ~ basin_name, data=df)
pander(tidy(mod), caption= my_ylab)
```

Table 8: Sen slope Snow cover melting cycles

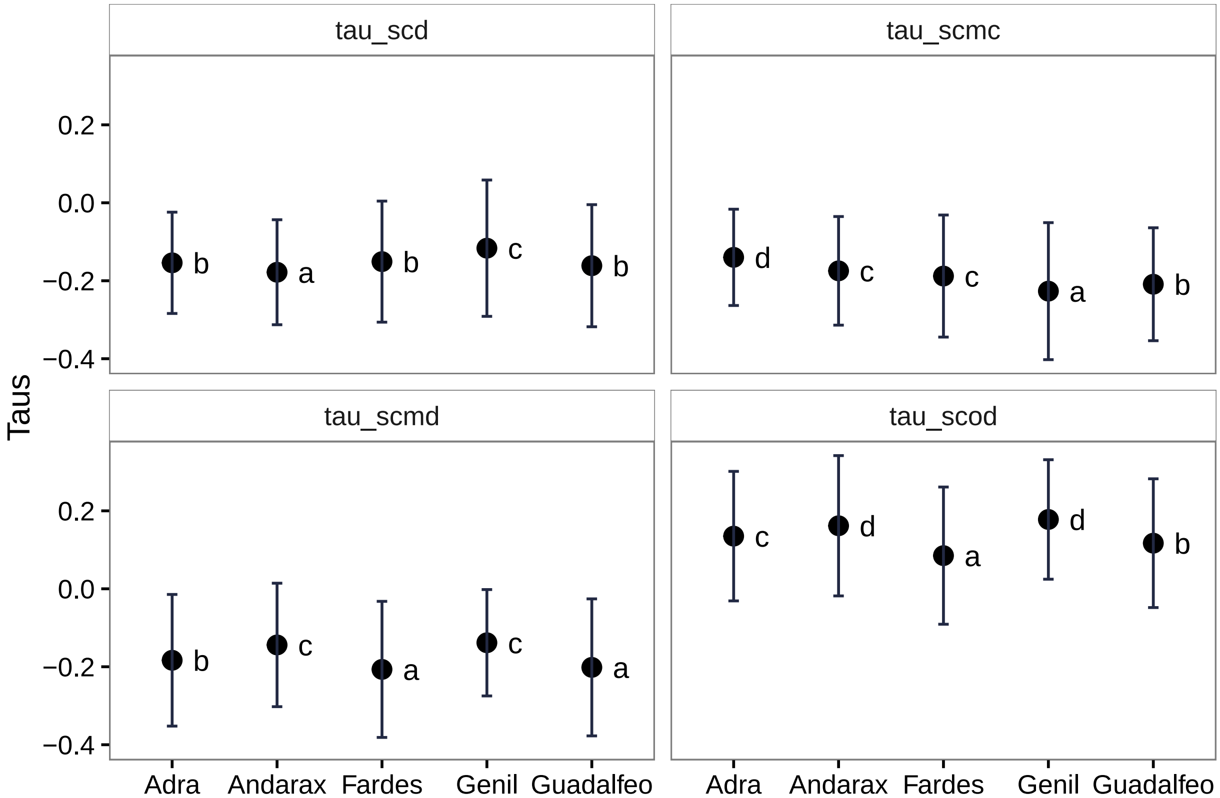
term	df	sumsq	meansq	statistic	p.value
basin_name	4	2.331	0.5827	77.78	1.63e-64
Residuals	6385	47.83	0.007492	NA	NA

```r
## Multiple comparisons
tuk <- glht(mod, linfct = mcp(basin_name = "Tukey"))
# Convert comparisons into letters
df_letter <- fortify(cld(tuk)) %>%
  transmute(basin_name = as.factor(lhs),
         tukey_basin_name = letters) %>%
  mutate(variable = variable)

df_letter_aux <- rbind(df_letter_aux, df_letter)

## ANOVA plots
ggplot(df, aes_string(x='basin_name', y=variable)) +
  geom_boxplot() +
```

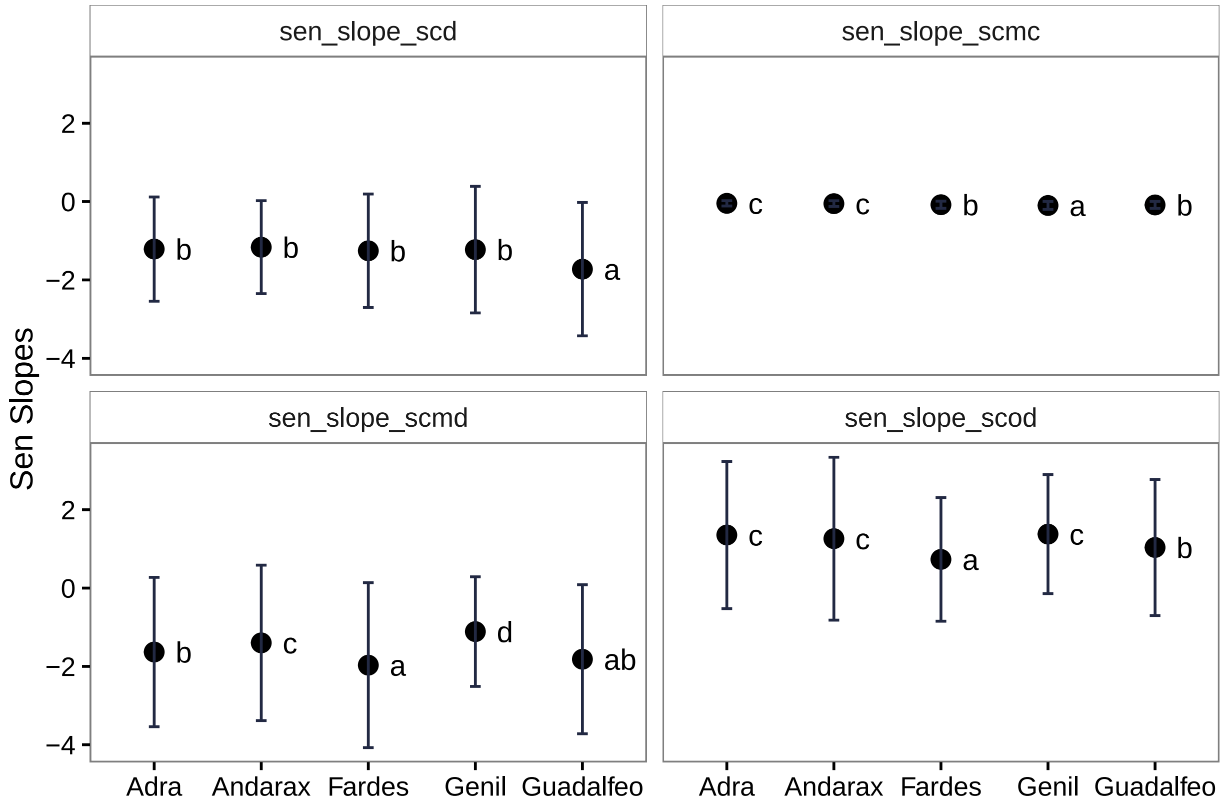
```
  geom_text(data=df_letter, aes(label = tukey_basin_name, x=basin_name, y=.5),
            position = position_dodge(width=0.9)) +
  theme_bw() + mythemeggplot +
  ylab(my_ylab) + xlab('')
```




```
# Create a df with letter and variables
df_basin <- stats_basin %>%
  dplyr::inner_join(df_letter_aux, by=c('basin_name', 'variable'))

df_basin_tau <- df_basin %>%
  dplyr::filter(grepl("tau", variable))


ggplot(df_basin_tau, aes(x=basin_name, y=mean)) +
  geom_point(stat='identity', size=3, fill='#222943') +
  geom_errorbar(aes(ymin=mean - sd, ymax= mean + sd),
                width=0.1,
                colour='#222943',
                position=position_dodge(.9)) +
  facet_wrap(~variable) +
  geom_text(aes(label = tukey_basin_name, x=basin_name, y=mean),
            hjust=0, nudge_x = 0.2) + mythemeggplot +
  xlab('') + ylab('Taus')
```




```r
# Create a df with letter and variables
df_basin_sen <- df_basin %>%
  dplyr::filter(grepl("sen", variable))


ggplot(df_basin_sen, aes(x=basin_name, y=mean)) +
  geom_point(stat='identity', size=3, fill='#222943') +
  geom_errorbar(aes(ymin=mean - sd, ymax= mean + sd),
                width=0.1,
                colour='#222943',
                position=position_dodge(.9)) +
  facet_wrap(~variable) +
  geom_text(aes(label = tukey_basin_name, x=basin_name, y=mean),
            hjust=0, nudge_x = 0.2) + mythemeggplot +
  xlab('') + ylab('Sen Slopes')
```



Tau Snow cover duration

```
pander(filter(df_basin_tau, variable=="tau_scd"))
```

basin_name	mean	sd	se	variable	tukey_basin_name
Adra	-0.1539916	0.1299276	0.004217628	tau_scd	b
Andarax	-0.1781580	0.1347095	0.003958622	tau_scd	a
Fardes	-0.1509319	0.1552694	0.004588631	tau_scd	b
Genil	-0.1164561	0.1748347	0.005312675	tau_scd	c
Guadalfeo	-0.1614019	0.1566468	0.003455537	tau_scd	b

Sen Slope Snow cover duration

```
pander(filter(df_basin_sen, variable=="sen_slope_scd"))
```

basin_name	mean	sd	se	variable	tukey_basin_name
Adra	-1.212126	1.330570	0.04319214	sen_slope_scd	b
Andarax	-1.165116	1.187804	0.03490522	sen_slope_scd	b
Fardes	-1.256298	1.450748	0.04287353	sen_slope_scd	b
Genil	-1.227084	1.616166	0.04911019	sen_slope_scd	b
Guadalfeo	-1.726302	1.703237	0.03757242	sen_slope_scd	a

Tau Snow cover onset date

```
pander(filter(df_basin_tau, variable=="tau_scod"))
```

basin_name	mean	sd	se	variable	tukey_basin_name
Adra	0.13508641	0.1661481	0.005393396	tau_scod	c
Andarax	0.16170725	0.1799857	0.005289125	tau_scod	d
Fardes	0.08508559	0.1760358	0.005202333	tau_scod	a
Genil	0.17786334	0.1532622	0.004657156	tau_scod	d
Guadalfeo	0.11700681	0.1651432	0.003642962	tau_scod	b

Sen Slope Snow onset date

```
pander(filter(df_basin_sen, variable=="sen_slope_scod"))
```

basin_name	mean	sd	se	variable	tukey_basin_name
Adra	1.3561486	1.880254	0.06103564	sen_slope_scod	c
Andarax	1.2622073	2.081075	0.06115520	sen_slope_scod	c
Fardes	0.7328856	1.579219	0.04667019	sen_slope_scod	a
Genil	1.3788957	1.519216	0.04616421	sen_slope_scod	c
Guadalfeo	1.0376273	1.738225	0.03834423	sen_slope_scod	b

Tau Snow cover melting date

```
pander(filter(df_basin_tau, variable=="tau_scmd"))
```

basin_name	mean	sd	se	variable	tukey_basin_name
Adra	-0.1833962	0.1688657	0.005481613	tau_scmd	b
Andarax	-0.1440147	0.1583987	0.004654762	tau_scmd	c
Fardes	-0.2067694	0.1745326	0.005157911	tau_scmd	a
Genil	-0.1384146	0.1364387	0.004145942	tau_scmd	c
Guadalfeo	-0.2016161	0.1757420	0.003876768	tau_scmd	a

Sen Slope Snow melting date

```
pander(filter(df_basin_sen, variable=="sen_slope_scmd"))
```

basin_name	mean	sd	se	variable	tukey_basin_name
Adra	-1.632706	1.907522	0.06192079	sen_slope_scmd	b
Andarax	-1.399011	1.984965	0.05833088	sen_slope_scmd	c
Fardes	-1.968318	2.106212	0.06224426	sen_slope_scmd	a
Genil	-1.111770	1.399082	0.04251370	sen_slope_scmd	d
Guadalfeo	-1.817089	1.902809	0.04197486	sen_slope_scmd	ab

Tau Snow cover melting cycles

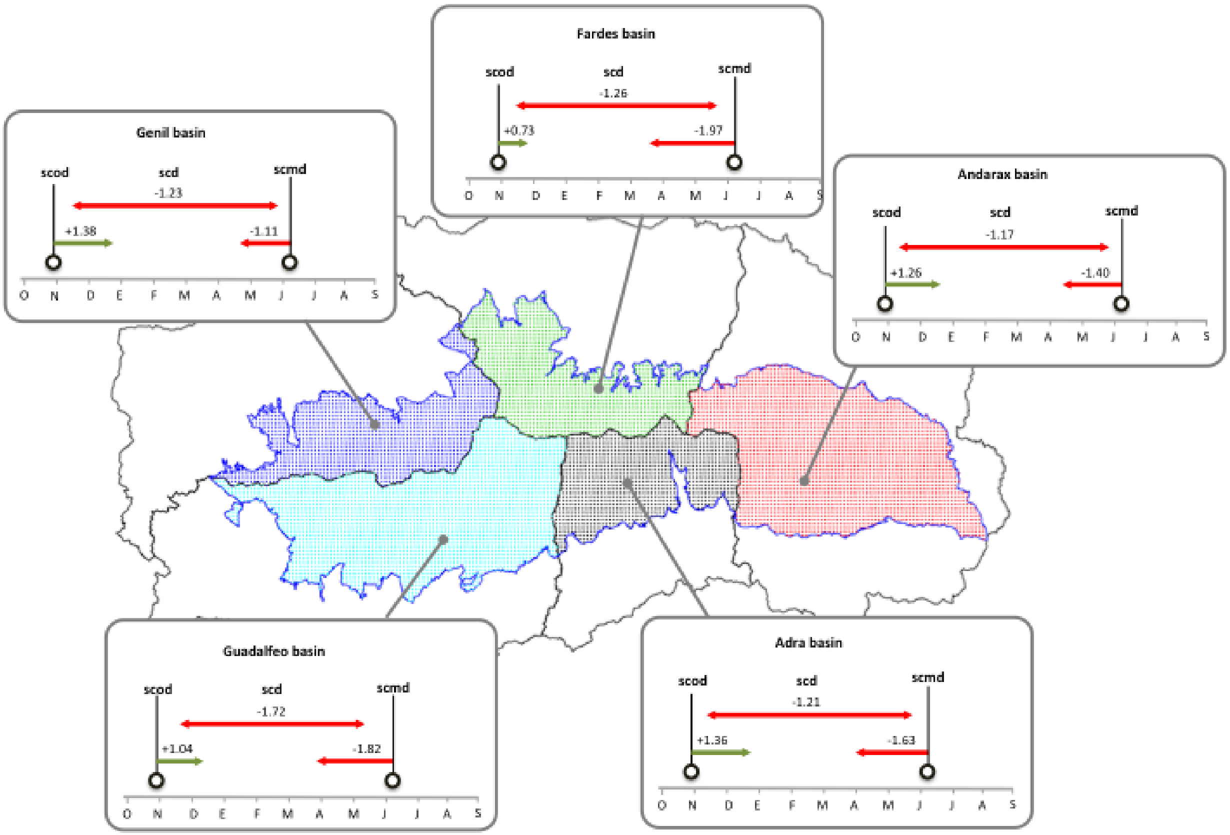
```r
pander(filter(df_basin_tau, variable=="tau_scmc"))
```

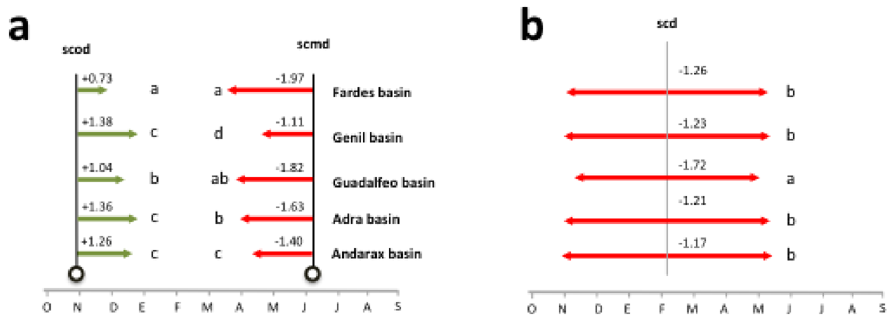
basin_name	mean	sd	se	variable	tukey_basin_name
Adra	-0.1399842	0.1235027	0.004009066	tau_scmc	d
Andarax	-0.1745829	0.1393072	0.004093732	tau_scmc	c
Fardes	-0.1879328	0.1565802	0.004627370	tau_scmc	c
Genil	-0.2266741	0.1757618	0.005340850	tau_scmc	a
Guadalfeo	-0.2089504	0.1448464	0.003195228	tau_scmc	b

Sen Slope Snow melting date

```r
pander(filter(df_basin_sen, variable=="sen_slope_scmc"))
```

basin_name	mean	sd	se	variable	tukey_basin_name
Adra	-0.04459536	0.06725779	0.002183280	sen_slope_scmc	c
Andarax	-0.05264162	0.07425655	0.002182129	sen_slope_scmc	c
Fardes	-0.07982009	0.09181152	0.002713279	sen_slope_scmc	b
Genil	-0.09900277	0.10182727	0.003094211	sen_slope_scmc	a
Guadalfeo	-0.08545109	0.08909295	0.001965339	sen_slope_scmc	b


Fardes basin
scod
scd
scmd
-1.26
+0.73
-1.97
Genil basin
scod
scd
scmd
-1.23
+1.38
-1.11
Andarax basin
scod
scd
scmd
-1.17
+1.26
-1.40
Guadalfeo basin
scod
scd
scmd
-1.72
+1.04
-1.82
Adra basin
scod
scd
scmd
-1.21
+1.36
-1.63
O N D E F M A M J J A S



Explore Snow-Cover trends by elevation

Taus

```
myvariable <- 'tau_scd'
myylab <- 'Tau Snow Cover Duration'

# GAM
myformula <- formula(paste(myvariable, '~ s(dem50mean)', sep=''))
modgam <- mgcv::gam(formula = myformula,
                    data= df)
anova(modgam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## tau_scd ~ s(dem50mean)
##
## Approximate significance of smooth terms:
```

```
##                 edf Ref.df     F p-value
## s(dem50mean) 8.263  8.854 410.4  <2e-16
```

```
mygam_summ <- summary(modgam)
mygam_summ
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## tau_scd ~ s(dem50mean)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.153844   0.001532  -100.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(dem50mean) 8.263  8.854 410.4  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.363   Deviance explained = 36.3%
## GCV = 0.015024  Scale est. = 0.015002  n = 6390
```

```
# Plot GAM
a <-ggplot(df, aes_string(x='dem50mean', y=myvariable)) + geom_point(col='grey') +
  geom_smooth(method="gam", formula = y ~ s(x), fill='red', col='red') +
  annotate("text", x = 3000, y=0.3,
           label= paste0("Dev. exp. = ", round((mygam_summ$dev.expl)*100, 2), " %")) +
  mythemeggplot +
  xlab('Elevation') + ylab(myylab)



myvariable <- 'tau_scod'
myylab <- 'Tau Snow Cover Onset Date'

# GAM
myformula <- formula(paste(myvariable, '~ s(dem50mean)', sep=''))
modgam <- mgcv::gam(formula = myformula,
                    data= df)
anova(modgam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## tau_scod ~ s(dem50mean)
```

```
##
## Approximate significance of smooth terms:
##                  edf Ref.df     F p-value
## s(dem50mean) 8.282  8.861 52.02  <2e-16
```

```
mygam_summ <- summary(modgam)
mygam_summ
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## tau_scod ~ s(dem50mean)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.132387   0.002066   64.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                  edf Ref.df     F p-value
## s(dem50mean) 8.282  8.861 52.02  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0671   Deviance explained = 6.83%
## GCV = 0.02732  Scale est. = 0.027281  n = 6390
```

```
# Plot GAM
b <- ggplot(df, aes_string(x='dem50mean', y=myvariable)) + geom_point(col='grey') +
  geom_smooth(method="gam", formula = y ~ s(x), fill='red', col='red') +
  annotate("text", x = 3000, y=0.6,
           label= paste0("Dev. exp. = ", round((mygam_summ$dev.expl)*100, 2), " %")) +
  mythemeggplot +
  xlab('Elevation') + ylab(myylab)


myvariable <- 'tau_scmd'
myylab <- 'Tau Snow Cover Melting Date'

# GAM
myformula <- formula(paste(myvariable, '~ s(dem50mean)', sep=''))
modgam <- mgcv::gam(formula = myformula,
                    data= df)
anova(modgam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
```

```
## tau_scmd ~ s(dem50mean)
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(dem50mean) 8.597  8.954 193.7  <2e-16
```

```
mygam_summ <- summary(modgam)
mygam_summ
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## tau_scmd ~ s(dem50mean)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.17868    0.00186  -96.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(dem50mean) 8.597  8.954 193.7  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.213   Deviance explained = 21.4%
## GCV = 0.02215  Scale est. = 0.022116  n = 6390
```

```
# Plot GAM
c <- ggplot(df, aes_string(x='dem50mean', y=myvariable)) + geom_point(col='grey') +
  geom_smooth(method="gam", formula = y ~ s(x), fill='red', col='red') +
  annotate("text", x = 3000, y=0.4,
           label= paste0("Dev. exp. = ", round((mygam_summ$dev.expl)*100, 2), " %")) +
  mythemeggplot +
  xlab('Elevation') + ylab(myylab)


myvariable <- 'tau_scmc'
myylab <- 'Tau Snow Cover Melting Cycles'

# GAM
myformula <- formula(paste(myvariable, '~ s(dem50mean)', sep=''))
modgam <- mgcv::gam(formula = myformula,
                    data= df)
anova(modgam)
```

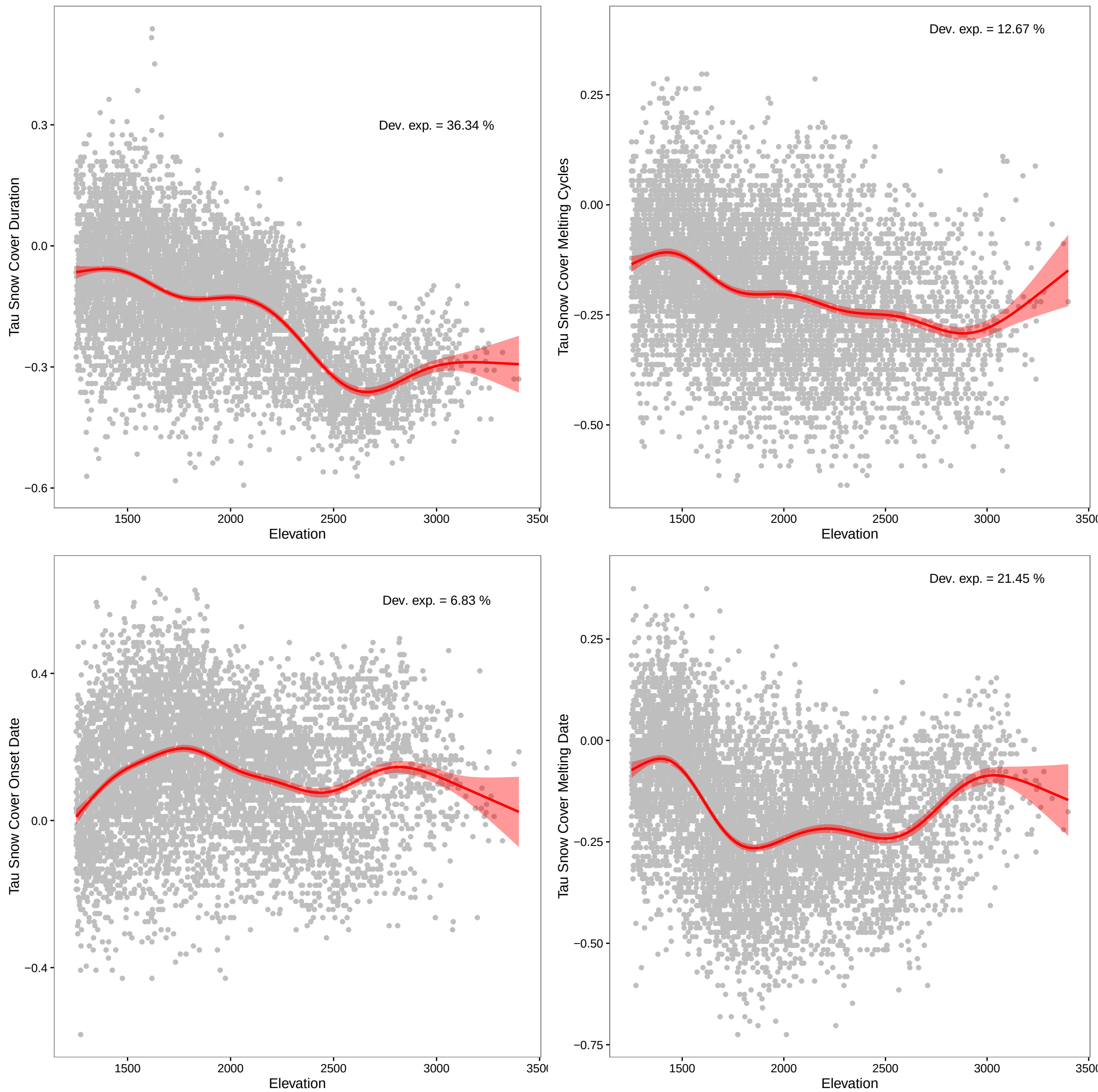
```
##
## Family: gaussian
## Link function: identity
##
```

```
## Formula:
## tau_scmc ~ s(dem50mean)
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(dem50mean) 8.211  8.834 104.2  <2e-16
```

```
mygam_summ <- summary(modgam)
mygam_summ
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## tau_scmc ~ s(dem50mean)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.19172    0.00177  -108.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(dem50mean) 8.211  8.834 104.2  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.126   Deviance explained = 12.7%
## GCV = 0.020053  Scale est. = 0.020024  n = 6390
```

```
# Plot GAM
d <- ggplot(df, aes_string(x='dem50mean', y=myvariable)) + geom_point(col='grey') +
  geom_smooth(method="gam", formula = y ~ s(x), fill='red', col='red') +
  annotate("text", x = 3000, y=0.4,
           label= paste0("Dev. exp. = ", round((mygam_summ$dev.expl)*100, 2), " %")) +
  mythemeggplot +
  xlab('Elevation') + ylab(myylab)


grid.arrange(a, d, b, c, nrow=2)
```



Sen

```r
myvariable <- 'sen_slope_scd'
myylab <- 'Sen Slope Snow Cover Duration'

# GAM
myformula <- formula(paste(myvariable, '~ s(dem50mean)', sep=''))
modgam <- mgcv::gam(formula = myformula,
                    data= df)
anova(modgam)
```

```
##
```

```
## Family: gaussian
## Link function: identity
##
## Formula:
## sen_slope_scd ~ s(dem50mean)
##
## Approximate significance of smooth terms:
##               edf Ref.df    F p-value
## s(dem50mean) 8.592  8.953 1115  <2e-16
```

```r
mygam_summ <- summary(modgam)
mygam_summ
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## sen_slope_scd ~ s(dem50mean)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.37941    0.01193  -115.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df    F p-value
## s(dem50mean) 8.592  8.953 1115  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.61   Deviance explained =   61%
## GCV = 0.91018  Scale est. = 0.90881   n = 6390
```

```r
# Plot GAM
a <-ggplot(df, aes_string(x='dem50mean', y=myvariable)) + geom_point(col='grey') +
  geom_smooth(method="gam", formula = y ~ s(x), fill='red', col='red') +
  annotate("text", x = 3000, y=2,
           label= paste0("Dev. exp. = ", round((mygam_summ$dev.expl)*100, 2), " %")) +
  mythemeggplot +
  xlab('Elevation') + ylab(myylab)



myvariable <- 'sen_slope_scod'
myylab <- 'Sen Slope Snow Cover Onset Date'

# GAM
myformula <- formula(paste(myvariable, '~ s(dem50mean)', sep=''))
modgam <- mgcv::gam(formula = myformula,
                    data= df)
anova(modgam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## sen_slope_scod ~ s(dem50mean)
##
## Approximate significance of smooth terms:
##                  edf Ref.df     F p-value
## s(dem50mean) 6.788  7.914 87.92  <2e-16
```

```
mygam_summ <- summary(modgam)
mygam_summ
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## sen_slope_scod ~ s(dem50mean)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.12886    0.02114    53.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                  edf Ref.df     F p-value
## s(dem50mean) 6.788  7.914 87.92  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0993   Deviance explained =   10%
## GCV = 2.8587  Scale est. = 2.8552    n = 6390
```

```
# Plot GAM
b <- ggplot(df, aes_string(x='dem50mean', y=myvariable)) + geom_point(col='grey') +
  geom_smooth(method="gam", formula = y ~ s(x), fill='red', col='red') +
  annotate("text", x = 3000, y=5,
           label= paste0("Dev. exp. = ", round((mygam_summ$dev.expl)*100, 2), " %")) +
  mythemeggplot +
  xlab('Elevation') + ylab(myylab)


myvariable <- 'sen_slope_scmd'
myylab <- 'Sen slope Snow Cover Melting Date'

# GAM
myformula <- formula(paste(myvariable, '~ s(dem50mean)', sep=''))
modgam <- mgcv::gam(formula = myformula,
                    data= df)
anova(modgam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## sen_slope_scmd ~ s(dem50mean)
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(dem50mean) 8.444  8.915 125.8  <2e-16
```

```
mygam_summ <- summary(modgam)
mygam_summ
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## sen_slope_scmd ~ s(dem50mean)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.62150    0.02199  -73.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(dem50mean) 8.444  8.915 125.8  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.149   Deviance explained =   15%
## GCV = 3.0936  Scale est. = 3.089     n = 6390
```

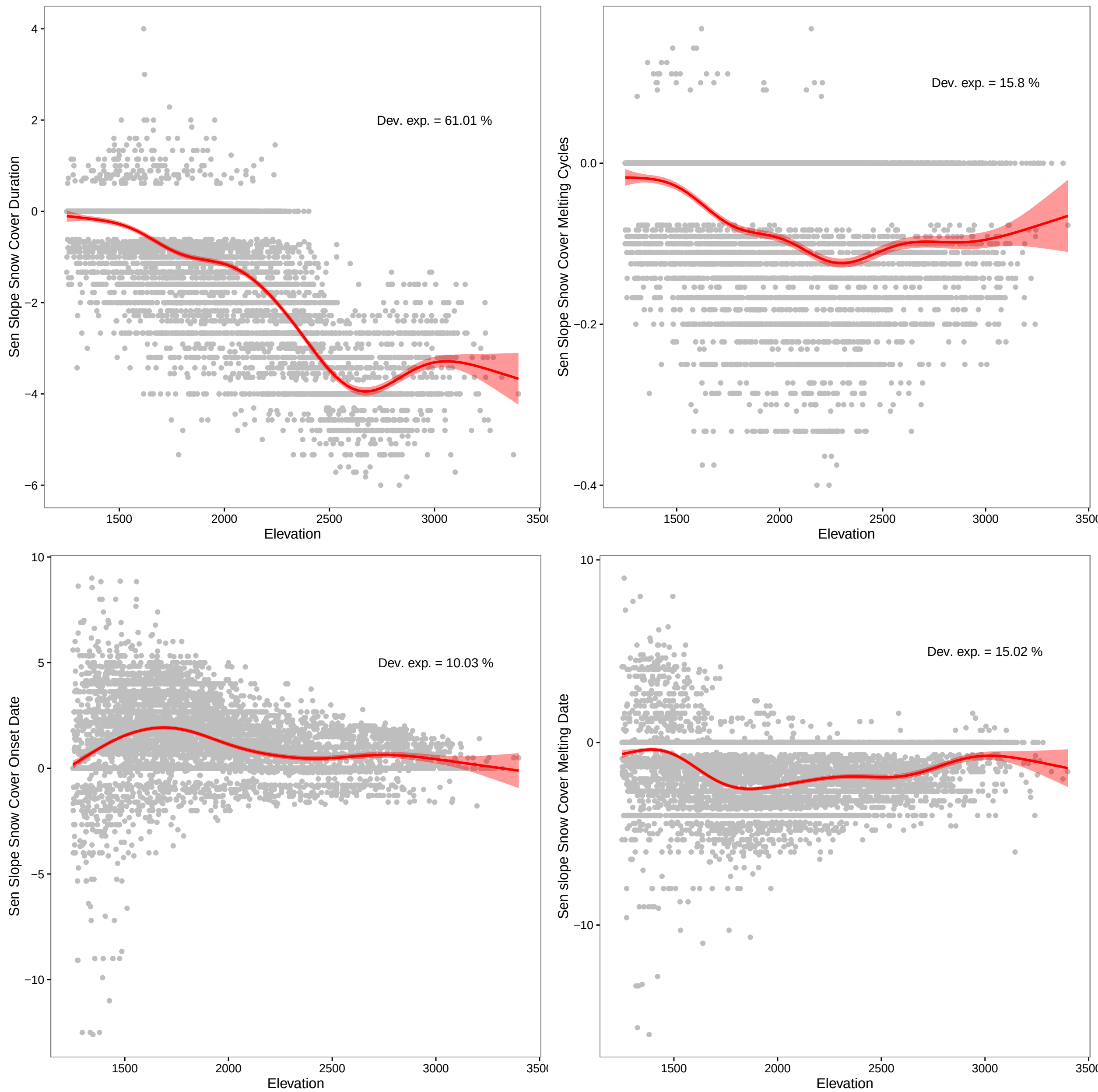
```
# Plot GAM
c <- ggplot(df, aes_string(x='dem50mean', y=myvariable)) + geom_point(col='grey') +
  geom_smooth(method="gam", formula = y ~ s(x), fill='red', col='red') +
  annotate("text", x = 3000, y=5,
           label= paste0("Dev. exp. = ", round((mygam_summ$dev.expl)*100, 2), " %")) +
  mythemeggplot +
  xlab('Elevation') + ylab(myylab)


myvariable <- 'sen_slope_scmc'
myylab <- 'Sen Slope Snow Cover Melting Cycles'

# GAM
myformula <- formula(paste(myvariable, '~ s(dem50mean)', sep=''))
modgam <- mgcv::gam(formula = myformula,
                    data= df)
anova(modgam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## sen_slope_scmc ~ s(dem50mean)
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(dem50mean) 7.828  8.651 137.6  <2e-16
```

```
mygam_summ <- summary(modgam)
mygam_summ
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## sen_slope_scmc ~ s(dem50mean)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.074726   0.001018  -73.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F p-value
## s(dem50mean) 7.828  8.651 137.6  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.157   Deviance explained = 15.8%
## GCV = 0.0066283  Scale est. = 0.0066191  n = 6390
```

```
# Plot GAM
d <- ggplot(df, aes_string(x='dem50mean', y=myvariable)) + geom_point(col='grey') +
  geom_smooth(method="gam", formula = y ~ s(x), fill='red', col='red') +
  annotate("text", x = 3000, y=0.1,
           label= paste0("Dev. exp. = ", round((mygam_summ$dev.expl)*100, 2), " %")) +
  mythemeggplot +
  xlab('Elevation') + ylab(myylab)


grid.arrange(a, d, b, c, nrow=2)
```


Sen Slope Snow Cover Duration
Dev. exp. = 61.01 %
Elevation
Sen Slope Snow Cover Melting Cycles
Dev. exp. = 15.8 %
Elevation
Sen Slope Snow Cover Onset Date
Dev. exp. = 10.03 %
Elevation
Sen slope Snow Cover Melting Date
Dev. exp. = 15.02 %
Elevation