

Posprocesamiento de líneas de costa creadas en Earth Engine

Ana Valera, Carolain Pérez, Yulisa Arias, José Martínez (tali)

2023-02-16

Representar líneas de costa, transectos, EPR

Cargar paquetes y funciones

```
library(tidyverse)
library(coastr)
library(sf)
library(lubridate)
library(RColorBrewer)
library(zoo)
library(kableExtra)
options(knitr.kable.NA = '')
my_kable_style <- function(df, caption = '', full_width = T) {
  df %>% kable(format = 'latex', escape = F, booktabs = T,
    digits = 2, caption = caption) %>%
    kable_styling(bootstrap_options = c("hover", "condensed"),
      latex_options = "HOLD_position",
      full_width = full_width, position = "center")
}
# Función para clasificar las partes de los transectos en mar o tierra respecto de referencia
devtools::source_url('https://raw.githubusercontent.com/geofis/RCoastSat/master/R/classify-transects.R')
# Función para calcular la distancia de cada línea de costa respecto de la de referencia sobre cada transecto
devtools::source_url('https://raw.githubusercontent.com/geofis/RCoastSat/master/R/extract-points-distances.R')
# Función para suavizar el relleno de los gráficos de series temporales
devtools::source_url('https://raw.githubusercontent.com/geofis/RCoastSat/master/R/interpolate.R')
# La línea siguiente fija el tamaño de los gráficos a unas proporciones "razonables". Editar a conveniencia
## Función para series temporales, tasas de erosión/acreción
```

```

calcular_tasa_de_cambio <- function(distancias = distances, transecto,
                                      metodo_descomposicion_loess = T, plot = F) {
  # Paquetes
  library(zoo)
  library(lubridate)
  # Datos
  df_selec <- distancias[distancias$transect==transecto, c('date', 'distance_sign')]
  todas_las_fechas <- data.frame(date = seq(min(df_selec$date), max(df_selec$date), by = "day"))
  df_completo <- merge(todas_las_fechas, df_selec, all.x = T)
  serie_regular <- zoo(df_completo$distance_sign, df_completo$date)

  # Rellenado de la serie usando medias móviles
  serie_rellena <- na.approx(serie_regular, na.rm = F)
  # Serie clase ts
  serie_rellena_ts <- ts(data = as.vector(serie_rellena), frequency = 365.25,
                         start = c(year(min(df_selec$date)), yday(min(df_selec$date)))))

  if(metodo_descomposicion_loess) {
    # Obtención de la tendencia por descomposición de la serie mediante LOESS, y obtención tasa de cambio (-erosión, +acreción)
    descomposicion <- stl(serie_rellena_ts, 'periodic', na.action = na.trim)
    if(plot) plot(descomposicion)
    tendencia_por_dia <- zoo(as.matrix(descomposicion$time.series[, 'trend']), todas_las_fechas$date)
    tendencia_por_mes <- aggregate(tendencia_por_dia, as.yearmon, tail, 1)
    tasa_de_cambio_por_mes <- diff(tendencia_por_mes)
    tasa_de_cambio_por_mes
    tendencia_por_ano <- aggregate(tendencia_por_dia, function(x) year(floor(as.yearmon(x))), tail, 1)
    tasa_de_cambio_por_ano <- diff(tendencia_por_ano)
    tasa_de_cambio_por_ano
    tasa_de_cambio_anual <- mean(tasa_de_cambio_por_ano, na.rm = T)
    tasa_de_cambio_anual }
  else {
    # Obtención de la tendencia por descomposición de la serie mediante medias móviles, y obtención tasa de cambio (-erosión, +acreción)
    descomposicion <- decompose(serie_rellena_ts)
    if(plot) plot(descomposicion)
    tasa_de_cambio_por_dia <- diff(descomposicion$trend)
    tendencia_por_dia <- zoo(descomposicion$trend, todas_las_fechas[-1, ])
    tendencia_por_mes <- aggregate(tendencia_por_dia, as.yearmon, tail, 1)
    tasa_de_cambio_por_mes <- diff(tendencia_por_mes)
  }
}

```

```

tasa_de_cambio_por_mes
tendencia_por_ano <- aggregate(tendencia_por_dia, function(x) year(floor(as.yearmon(x))), tail, 1)
tasa_de_cambio_por_ano <- diff(tendencia_por_ano)
tasa_de_cambio_por_ano
tasa_de_cambio_anual <- mean(tasa_de_cambio_por_ano, na.rm = T)
tasa_de_cambio_anual
}
return(
  list(
    `Periodo y tasa de cambio anual` = data.frame(
      `Periodo` = paste0(year(min(todas_las_fechas$date)),
        '-',
        year(max(todas_las_fechas$date)),
        ' (',
        year(max(todas_las_fechas$date)) - year(min(todas_las_fechas$date)),
        ' años)'),
      `Tasa de cambio anual, en metros (-erosión,+acreción)` = tasa_de_cambio_anual,
      check.names = F),
    `Tasa de cambio por año, en metros (-erosión,+acreción)` = tasa_de_cambio_por_ano))
}
generar_tabla_de_tasas_de_cambio <- function(tasas_de_cambio, titulo, csv = F) {
  df <- as.data.frame(t(sapply(tasas_de_cambio, function(x) x[[c(1)]])))) %>%
    mutate(across(starts_with('Tasa'), as.numeric)) %>%
    rownames_to_column('Transecto')
  if(!csv) {
    df_kable <- df %>%
      my_kable_style(caption = titulo,
                     full_width = F) %>%
      column_spec(column = 3, width = "15em")
    print(df_kable)
  }
  invisible(return(df))
}
generar_tabla_de_tasas_de_cambio_por_ano <- function(tasas_de_cambio, titulo, csv = F) {
  df <- sapply(names(tasas_de_cambio), function(x) {
    data.frame(
      foo = tasas_de_cambio[[x]][2]$`Tasa de cambio por año, en metros (-erosión,+acreción)`,
      check.names = F) %>%

```

```

rownames_to_column('Año') %>%
  setNames(c('Año', x))
}, simplify = F) %>% plyr::join_all()
if(!csv) {
  df_kable <- df %>%
    my_kable_style(caption = titulo,
                   full_width = T)
  print(df_kable)
}
return(df)
}

```

Pre-2009

Tramo Este

- Cargar líneas de costa

```

lineas <- st_read('lineas-de-costa/PalenqueNizaoPC_L5_output_lines.gpkg') %>%
  filter(grepl('palenque', tramo, ignore.case = T)) %>%
  st_cast('LINESTRING')

## Reading layer `PalenqueNizaoPC_L5_output_lines` from data source
##   `/home/jose/Documentos/git/tesis-ana-carolain/lineas-de-costa/PalenqueNizaoPC_L5_output_lines.gpkg'
##   using driver `GPKG'
## replacing null geometries with empty geometries
## Simple feature collection with 51 features and 5 fields (with 1 geometry empty)
## Geometry type: LINESTRING
## Dimension:      XY
## Bounding box:  xmin: 368865 ymin: 2015212 xmax: 377395.8 ymax: 2016785
## Projected CRS: WGS 84 / UTM zone 19N

st_geometry(lineas) <- "geometry"
lineas$longitud <- units::drop_units(st_length(lineas))
lineas <- lineas[lineas$longitud > 0, ]

```

- Representar las líneas de costa

```

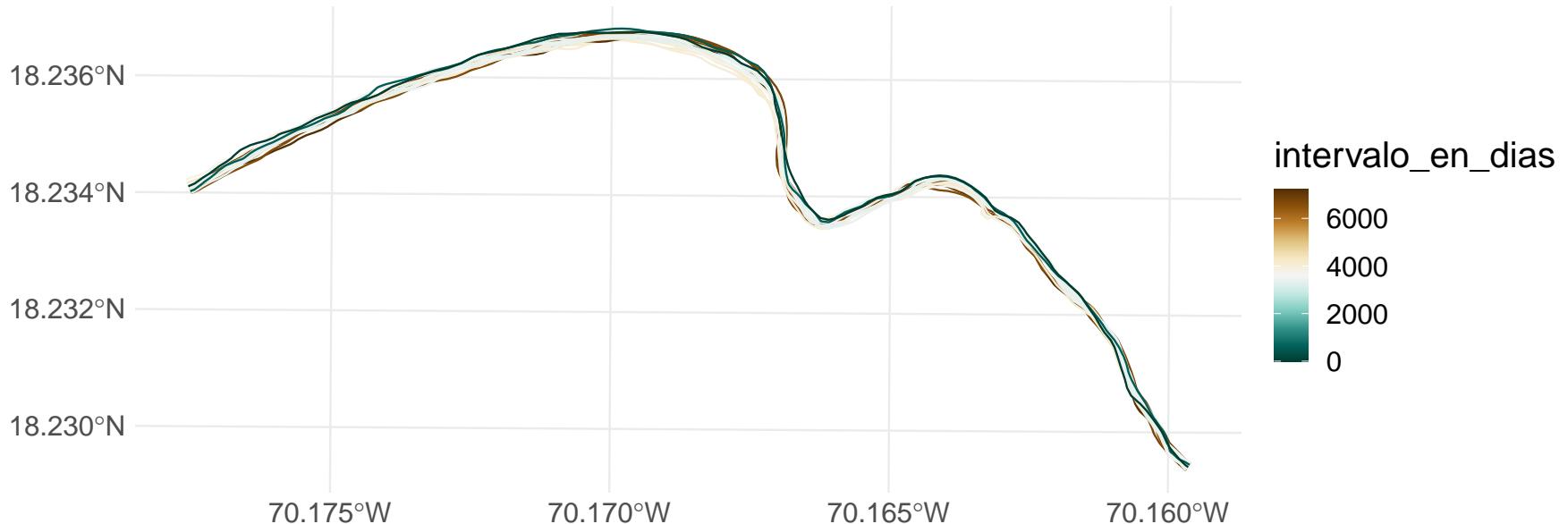
lineas$intervalo_en_dias <- round(as.numeric(interval(lineas$date, max(lineas$date))), 'days'), 0)
escala_color <- 'BrBG'
mapa_lineas <- lineas %>% ggplot + aes(color=intervalo_en_dias) + geom_sf() +

```

```

theme_minimal() +
  theme(text = element_text(size = 18)) +
  scale_color_gradientn(colors = rev(RColorBrewer::brewer.pal(11, escala_color)))
mapa_lineas

```



- Crear transectos respecto de línea de costa de referencia y representarlos

```

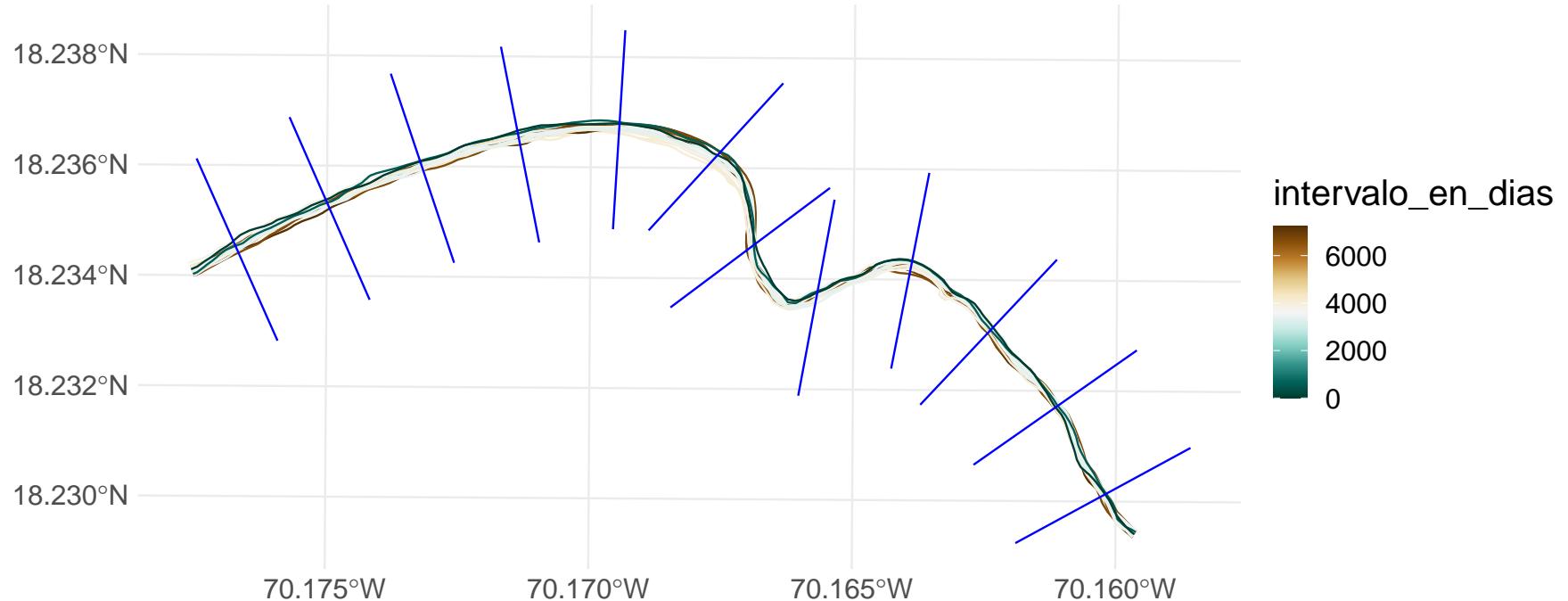
# Umbral de longitud para líneas que podrían usarse como referencia
umbral_longitud <- 1000

# Elegir una línea de referencia
linea_ref <- lineas %>% filter(longitud > umbral_longitud) %>% filter(date == min(date))

# Crear transectos
transectos <- create_transect(x = linea_ref, 200, reverse = T) %>% rename(transect=coastr_id)

# Mapa
mapa_lineas + geom_sf(data = transectos, color = 'blue')

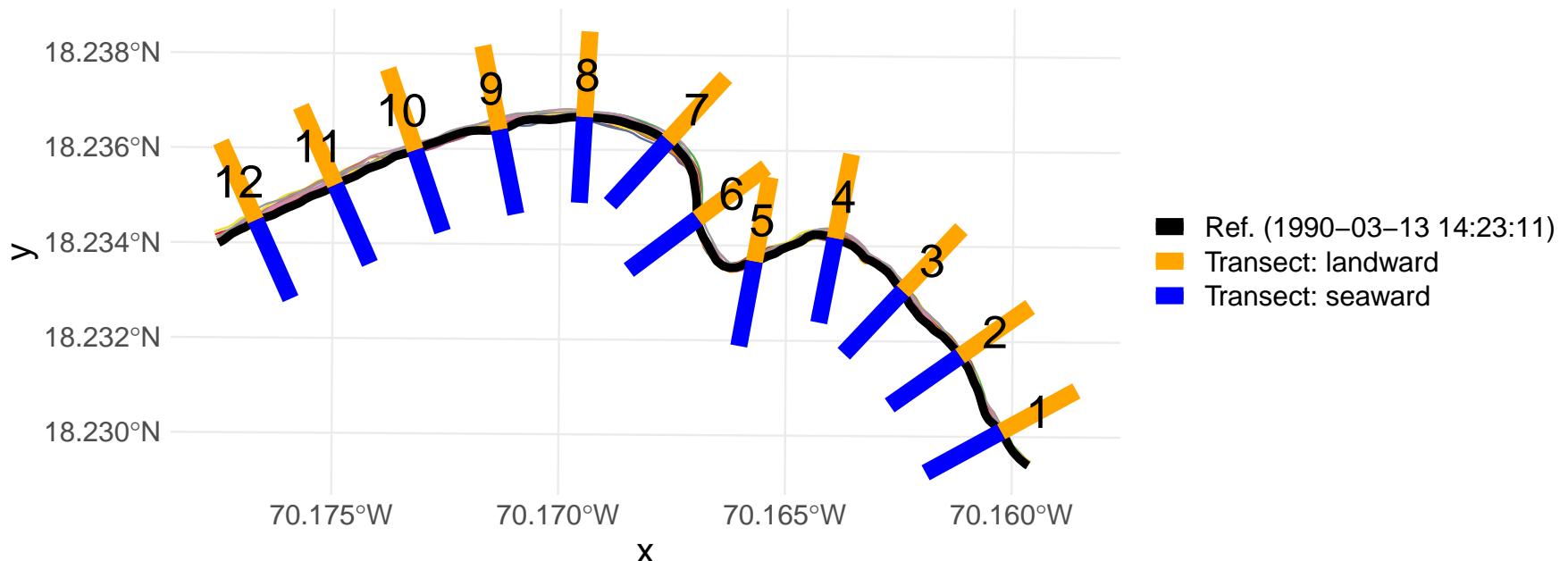
```



- Clasificar las distintas partes del transecto en tierra o mar

```

transectos_clasif <- transclas(tr = transectos, rl = linea_ref)
cols <- colorRampPalette(brewer.pal(9, 'Set1'))(nrow(lineas))
ggplot() +
  geom_sf(data = lineas %>% mutate(date = factor(date)), color = cols) +
  geom_sf(
    data = linea_ref %>% mutate(linetype = paste0('Ref. ', date, ' ')),
    aes(color=linetype), linewidth = 2, show.legend = 'line') +
  geom_sf(
    data = transectos_clasif %>% mutate(sealand=paste0('Transect: ', sealand)),
    aes(color = sealand), show.legend = 'line', linewidth = 4) +
  scale_color_manual(values = c('black', 'orange', 'blue')) +
  geom_sf_text(
    data = transectos_clasif %>% filter(sealand=='landward') %>%
      st_centroid, aes(label = transect), size = 8) +
  theme_minimal() +
  theme(legend.title = element_blank(), text = element_text(size = 18))
  
```

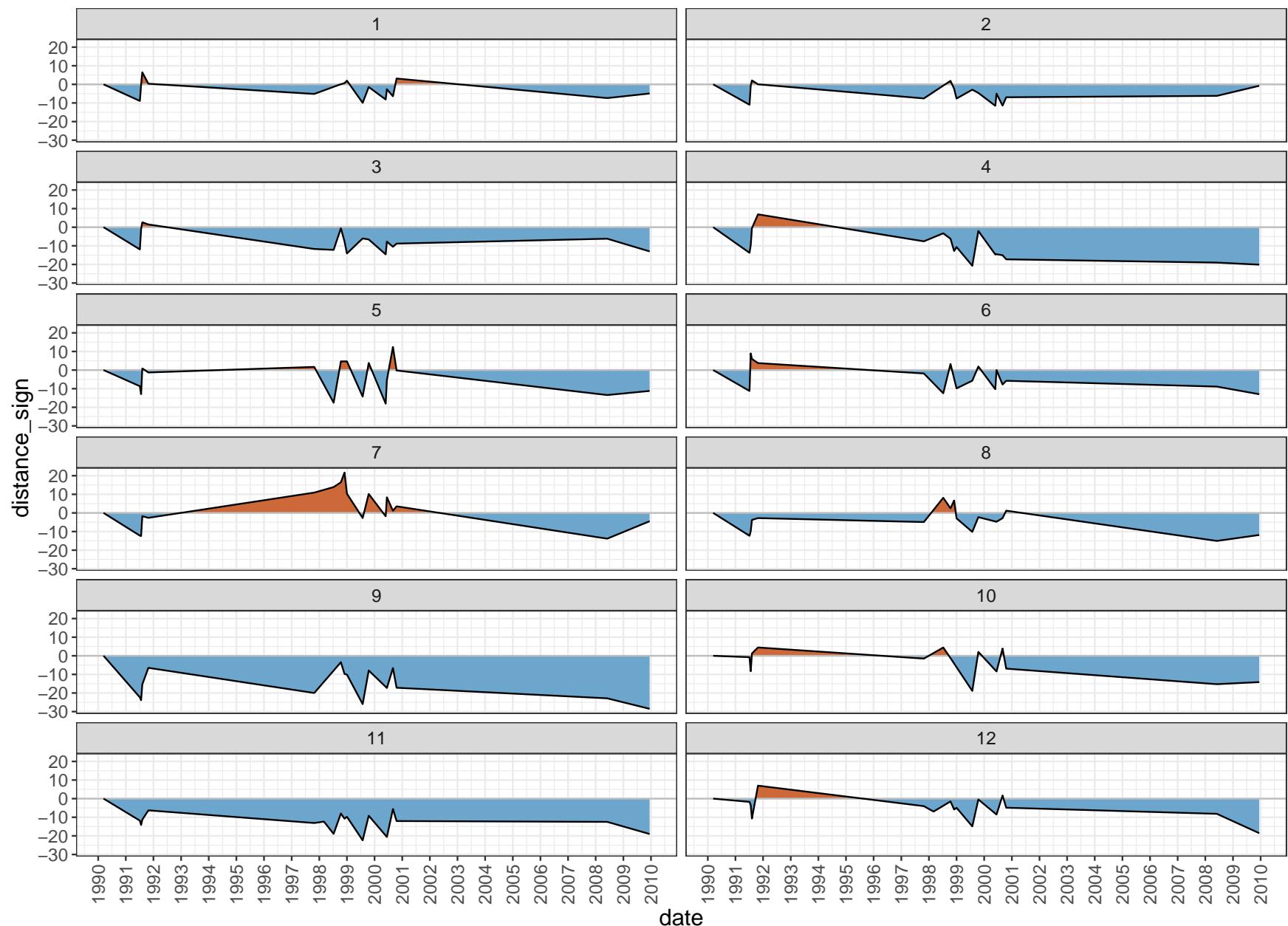


- Calcular distancias de cada línea de costa respecto de la línea de referencia

```
distl <- pointdist(sh = lineas, re = linea_ref, tr = transectos_clasif, rtr = transectos)
```

- Generar las series temporales de distancia de la línea de costa respecto a la de referencia

```
interdist <- map(distl, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances <- plyr::ldply(distl) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
#  scale_y_continuous(limits = c(-30, 30)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)
```



- Revisión de seguridad: determinar si hay transectos que corten dos veces una misma línea de costa

```
test <- sapply(unique(distances$transect), function(x) {
  conteo_cortes <- table(distances[distances$transect==x, 'date', drop=T])
  mas_de_1 <- length(which(conteo_cortes>1))>0
  ifelse(mas_de_1,
    paste('El transecto', x, 'corta', conteo_cortes[which(conteo_cortes>1)],
      'veces la línea de costa de fecha', names(conteo_cortes[which(conteo_cortes>1)])),
    paste('El transecto', x, 'pasa la prueba'))
})
test

## [1] "El transecto 1 pasa la prueba" "El transecto 2 pasa la prueba"
## [3] "El transecto 3 pasa la prueba" "El transecto 4 pasa la prueba"
## [5] "El transecto 5 pasa la prueba" "El transecto 6 pasa la prueba"
## [7] "El transecto 7 pasa la prueba" "El transecto 8 pasa la prueba"
## [9] "El transecto 9 pasa la prueba" "El transecto 10 pasa la prueba"
## [11] "El transecto 11 pasa la prueba" "El transecto 12 pasa la prueba"
```

- Tasas de cambio (-erosión, +acreción)

```
tasas_de_cambio <- lapply(
  unique(distances$transect),
  function (x) calcular_tasa_de_cambio(transecto = x)) %>%
  setNames(paste('Transecto', unique(distances$transect)))
invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(titulo = 'Tasa de cambio anual promediada pre-2009 para transectos del Tramo Este'))
```

Table 1: Tasa de cambio anual promediada pre-2009 para transectos del Tramo Este

Transecto	Periodo	Tasa de cambio anual, en metros (-erosión,+acreción)
Transecto 1	1990-2009 (19 años)	0.00
Transecto 2	1990-2009 (19 años)	0.28
Transecto 3	1990-2009 (19 años)	-0.34
Transecto 4	1990-2009 (19 años)	-0.64
Transecto 5	1990-2009 (19 años)	-0.35
Transecto 6	1990-2009 (19 años)	-0.38
Transecto 7	1990-2009 (19 años)	0.13
Transecto 8	1990-2009 (19 años)	-0.25
Transecto 9	1990-2009 (19 años)	-0.80
Transecto 10	1990-2009 (19 años)	-0.73
Transecto 11	1990-2009 (19 años)	-0.63
Transecto 12	1990-2009 (19 años)	-0.92

```
invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(
    titulo = 'Tasa de cambio por año pre-2009 (en metros; -erosión, +acreción) para transectos del Tramo Este'))
```

Table 2: Tasa de cambio por año pre-2009 (en metros; -erosión, +acreción) para transectos del Tramo Este

Año	Transecto 1	Transecto 2	Transecto 3	Transecto 4	Transecto 5	Transecto 6	Transecto 7	Transecto 8	Transecto 9	Transecto 10	Transecto 11	Transecto 12
1991	5.46	5.78	7.32	12.24	3.61	9.55	4.55	3.72	4.33	3.81	-0.35	5.10
1992	-1.10	-0.91	-1.53	-0.08	0.88	-0.55	2.75	0.39	0.00	-0.02	-0.08	0.88
1993	-0.91	-1.26	-2.20	-2.42	0.49	-0.93	2.26	-0.35	-2.25	-1.00	-1.14	-1.84
1994	-0.91	-1.26	-2.19	-2.42	0.50	-0.93	2.26	-0.35	-2.25	-1.00	-1.13	-1.84
1995	-0.91	-1.26	-2.19	-2.41	0.50	-0.93	2.26	-0.35	-2.25	-0.99	-1.13	-1.83
1996	-0.92	-1.27	-2.20	-2.42	0.49	-0.93	2.26	-0.35	-2.26	-1.00	-1.14	-1.84
1997	0.55	1.29	-1.77	-0.47	-5.75	-4.09	2.73	3.95	2.22	1.14	-1.38	-2.42
1998	2.35	2.00	2.89	-5.02	2.19	-0.97	-1.12	-0.06	4.10	-6.55	0.18	-1.35
1999	-3.48	-3.51	-0.51	1.20	-3.95	2.55	-6.56	-3.62	-2.09	1.30	-1.74	0.84
2000	6.01	-0.72	0.32	-6.92	6.74	-2.29	-1.25	4.08	-2.01	-1.01	3.33	1.13
2001	0.45	0.65	0.58	-0.72	-2.81	-0.38	-2.05	-1.08	-2.14	-2.41	-0.34	-1.05
2002	-1.38	0.10	0.34	-0.23	-1.74	-0.41	-2.28	-2.13	-0.75	-1.10	-0.06	-0.44
2003	-1.37	0.10	0.35	-0.23	-1.72	-0.41	-2.27	-2.13	-0.74	-1.09	-0.05	-0.43
2004	-1.38	0.10	0.35	-0.23	-1.74	-0.41	-2.29	-2.14	-0.75	-1.10	-0.06	-0.43
2005	-1.38	0.10	0.35	-0.23	-1.73	-0.41	-2.28	-2.13	-0.75	-1.10	-0.06	-0.43
2006	-1.37	0.10	0.34	-0.23	-1.73	-0.41	-2.28	-2.13	-0.74	-1.09	-0.05	-0.43
2007	-1.36	0.12	0.31	-0.24	-1.71	-0.43	-2.22	-2.10	-0.77	-1.08	-0.09	-0.48
2008	0.33	2.07	-2.41	-0.51	0.09	-1.71	2.53	0.29	-2.43	-0.05	-2.47	-4.08
2009	1.39	3.06	-4.68	-0.86	0.83	-3.07	5.50	1.82	-3.72	0.52	-4.19	-6.59

Exportar las tablas anteriores a CSV.

```
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(csv = T) %>%
  write_csv('tasas_de_cambio_anual_promediada_pre_2009_por_transecto_tramo_este.csv')
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(csv = T) %>%
  write_csv('tasas_de_cambio_anual_pre_2009_por_transecto_tramo_este.csv')
```

- Suavizado de la serie con media móvil

```
ventana_de_promediado <- 3 #Número de observaciones para obtener la media móvil (ventana de promediado)
dist1_med <- sapply(unique(distances$transect),
  function(x){
```

```

df <- distances[distances$transect==x, ]
df <- df[order(df$date), ]
x <- zoo(df$distance_sign, df$date)
mm <- as.numeric(rollmean(x, ventana_de_promediado, fill = NA))
df$distance_sign <- mm
df <- df #%>% slice(1:(n()-1))
return(df)
}, simplify=F)
interdist_med <- map(distl_med, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances_med <- plyr::ldply(distl_med) %>% mutate(date = as.Date(date, "%Y-%m-%d"))

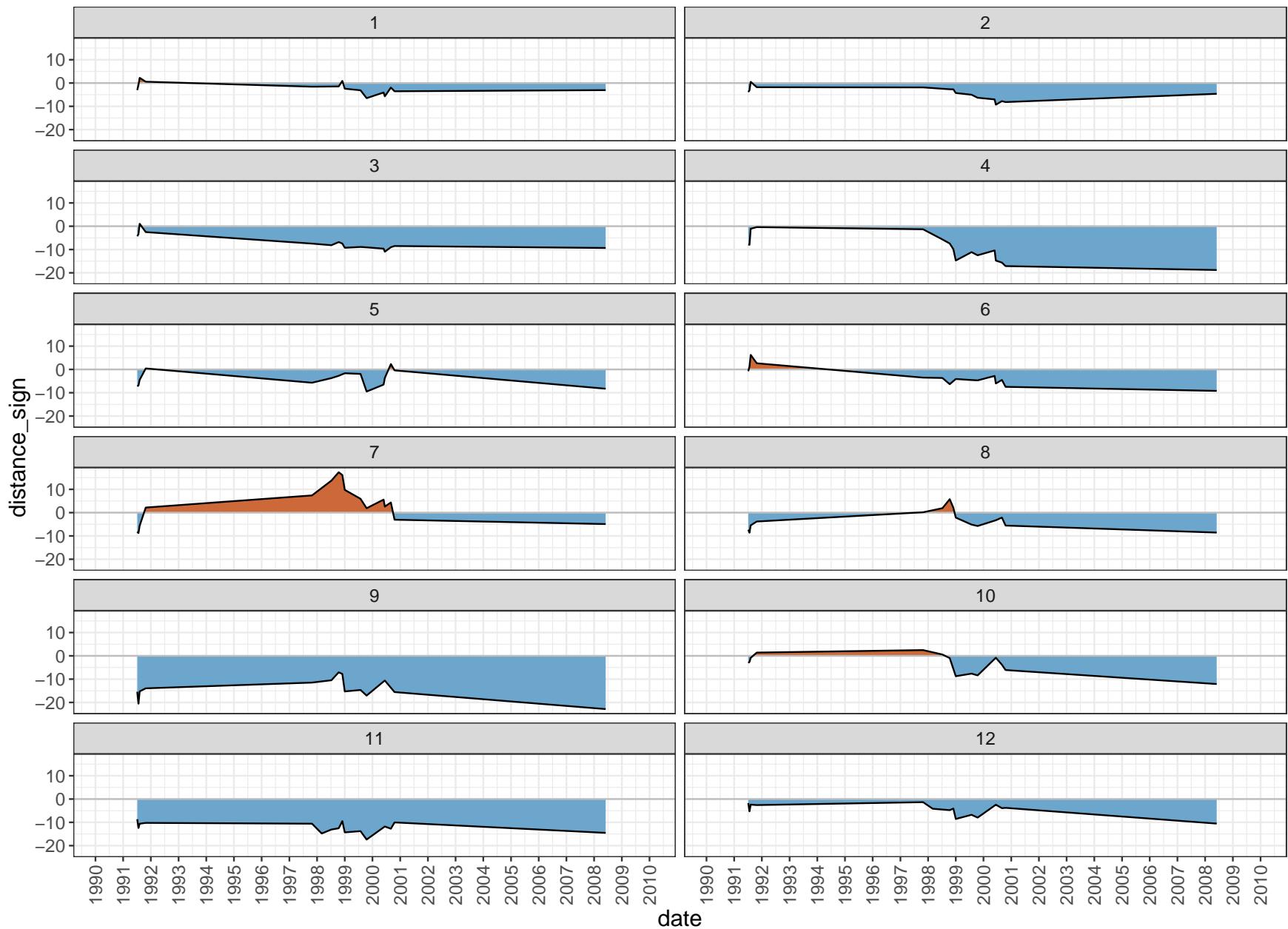
```

- Representación de la serie suavizada

```

distances_med %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist_med, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist_med, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)

```



Tramo Oeste

Nota: el suavizado, para este tramo, no funciona adecuadamente.

- Cargar líneas de costa

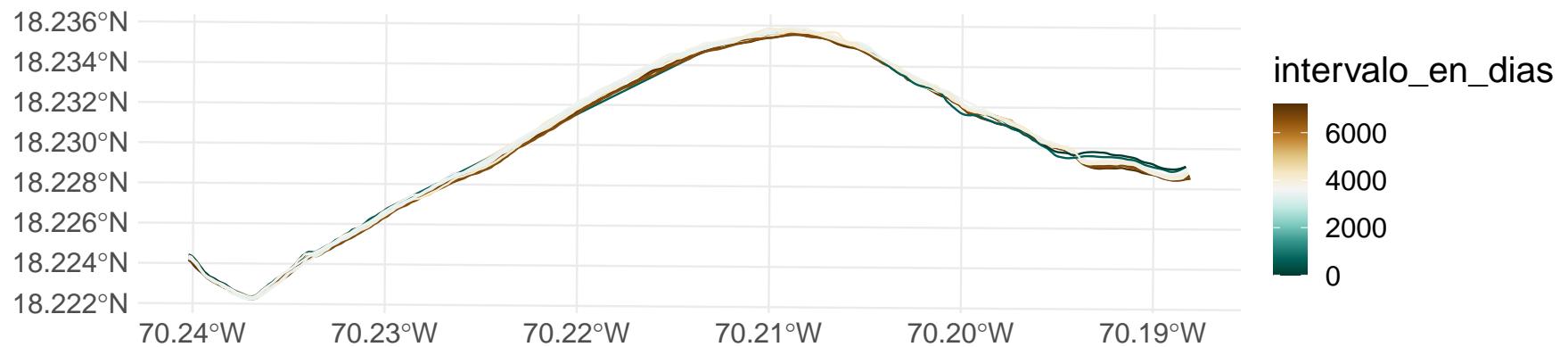
```
lineas <- st_read('lineas-de-costa/PalenqueNizaoPC_L5_output_lines.gpkg') %>%
  filter(grepl('nizao', tramo, ignore.case = T)) %>%
  st_cast('LINESTRING')

## Reading layer `PalenqueNizaoPC_L5_output_lines` from data source
##   `/home/jose/Documentos/git/tesis-ana-carolain/lineas-de-costa/PalenqueNizaoPC_L5_output_lines.gpkg'
##   using driver `GPKG'
## replacing null geometries with empty geometries
## Simple feature collection with 51 features and 5 fields (with 1 geometry empty)
## Geometry type: LINESTRING
## Dimension:      XY
## Bounding box:  xmin: 368865 ymin: 2015212 xmax: 377395.8 ymax: 2016785
## Projected CRS: WGS 84 / UTM zone 19N

st_geometry(lineas) <- "geometry"
lineas$longitud <- units::drop_units(st_length(lineas))
lineas <- lineas[lineas$longitud > 0, ]
```

- Representar las líneas de costa

```
lineas$intervalo_en_dias <- round(as.numeric(interval(lineas$date, max(lineas$date))), 'days'), 0)
escala_color <- 'BrBG'
mapa_lineas <- lineas %>% ggplot + aes(color=intervalo_en_dias) + geom_sf() +
  theme_minimal() +
  theme(text = element_text(size = 18)) +
  scale_color_gradientn(colors = rev(RColorBrewer::brewer.pal(11, escala_color)))
mapa_lineas
```



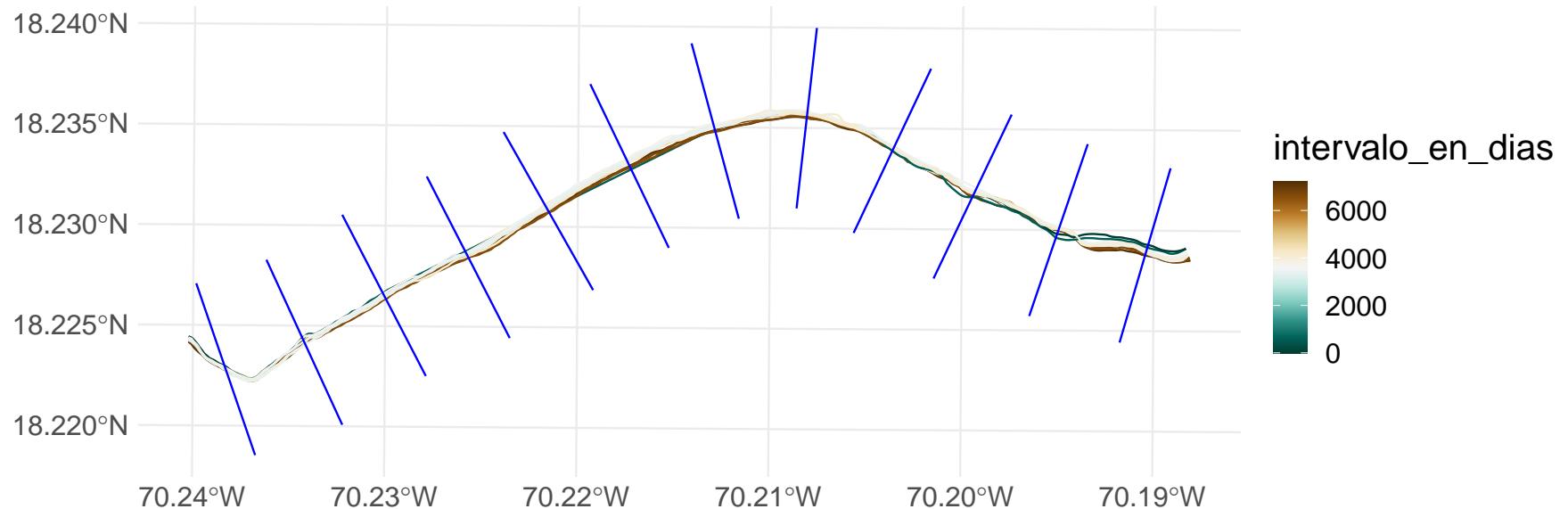
- Crear transectos respecto de línea de costa de referencia y representarlos

```
# Umbral de longitud para líneas que podrían usarse como referencia
umbral_longitud <- 1000

# Elegir una linea de referencia
linea_ref <- lineas %>% filter(longitud > umbral_longitud) %>% filter(date == min(date))

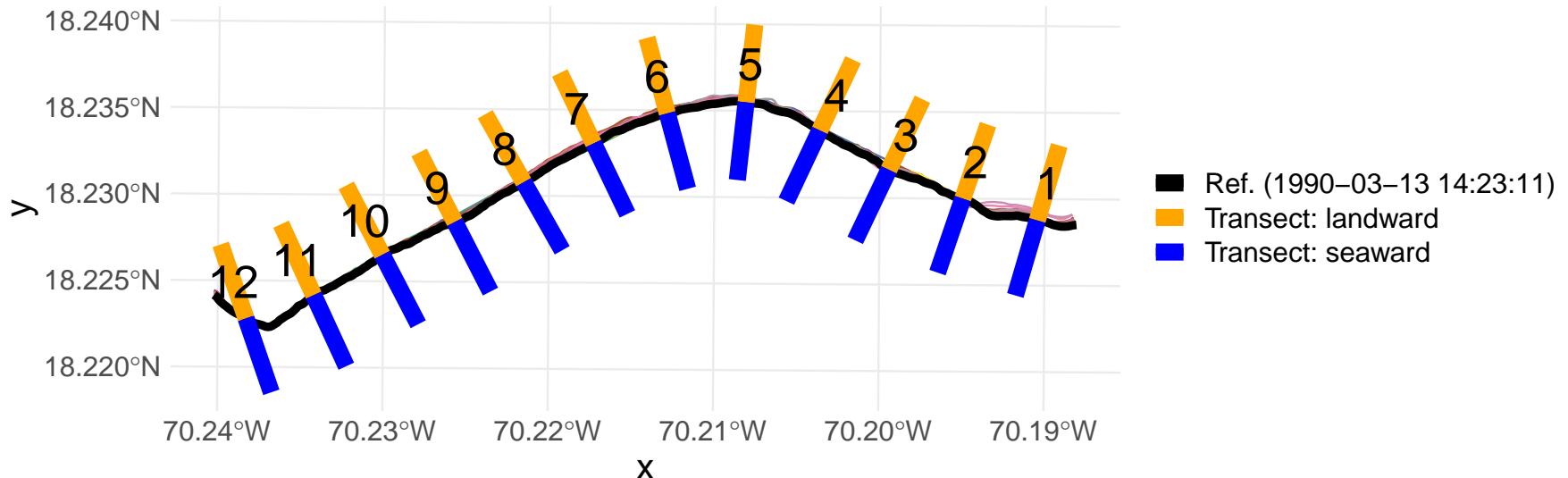
# Crear transectos
transectos <- create_transect(x = linea_ref, 500, reverse = T) %>% rename(transect=coastr_id)

# Mapa
mapa_lineas + geom_sf(data = transectos, color = 'blue')
```



- Clasificar las distintas partes del transecto en tierra o mar

```
transectos_clasif <- transclas(tr = transectos, rl = linea_ref)
cols <- colorRampPalette(brewer.pal(9, 'Set1'))(nrow(lineas))
ggplot() +
  geom_sf(data = lineas %>% mutate(date = factor(date)), color = cols) +
  geom_sf(
    data = linea_ref %>% mutate(linetype = paste0('Ref. ', date, ' ')),
    aes(color=linetype), linewidth = 2, show.legend = 'line') +
  geom_sf(
    data = transectos_clasif %>% mutate(sealand=paste0('Transect: ', sealand)),
    aes(color = sealand), show.legend = 'line', linewidth = 4) +
  scale_color_manual(values = c('black', 'orange', 'blue')) +
  geom_sf_text(
    data = transectos_clasif %>% filter(sealand=='landward') %>%
      st_centroid, aes(label = transect), size = 8) +
  theme_minimal() +
  theme(legend.title = element_blank(), text = element_text(size = 18))
```

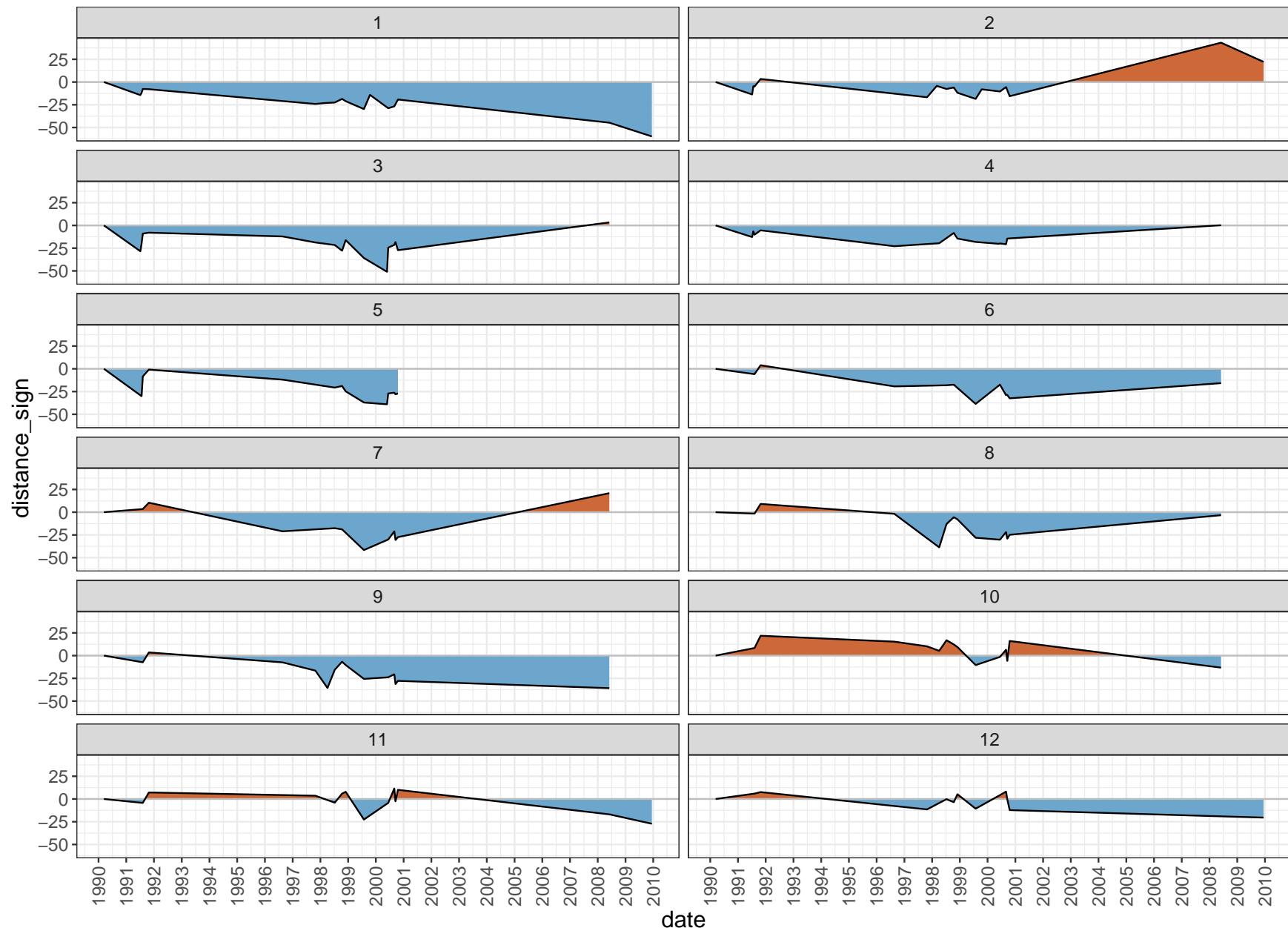


- Calcular distancias de cada línea de costa respecto de la línea de referencia

```
distl <- pointdist(sh = lineas, re = linea_ref, tr = transectos_clasif, rtr = transectos)
```

- Generar las series temporales de distancia de la línea de costa respecto a la de referencia

```
interdist <- map(distl, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances <- plyr::ldply(interdist) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
#  scale_y_continuous(limits = c(-30, 30)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)
```



- Revisión de seguridad: determinar si hay transectos que corten dos veces una misma línea de costa

```
test <- sapply(unique(distances$transect), function(x) {
  conteo_cortes <- table(distances[distances$transect==x, 'date', drop=T])
  mas_de_1 <- length(which(conteo_cortes>1))>0
  ifelse(mas_de_1,
    paste('El transecto', x, 'corta', conteo_cortes[which(conteo_cortes>1)],
      'veces la línea de costa de fecha', names(conteo_cortes[which(conteo_cortes>1)])),
    paste('El transecto', x, 'pasa la prueba'))
})
test

## [1] "El transecto 1 pasa la prueba" "El transecto 2 pasa la prueba"
## [3] "El transecto 3 pasa la prueba" "El transecto 4 pasa la prueba"
## [5] "El transecto 5 pasa la prueba" "El transecto 6 pasa la prueba"
## [7] "El transecto 7 pasa la prueba" "El transecto 8 pasa la prueba"
## [9] "El transecto 9 pasa la prueba" "El transecto 10 pasa la prueba"
## [11] "El transecto 11 pasa la prueba" "El transecto 12 pasa la prueba"
```

- Tasas de cambio (-erosión, +acreción)

```
tasas_de_cambio <- lapply(
  unique(distances$transect),
  function (x) calcular_tasa_de_cambio(transecto = x)) %>%
  setNames(paste('Transecto', unique(distances$transect)))
invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(titulo = 'Tasa de cambio anual promediada pre-2009 para transectos del Tramo Oeste'))
```

Table 3: Tasa de cambio anual promediada pre-2009 para transectos del Tramo Oeste

Transecto	Periodo	Tasa de cambio anual, en metros (-erosión,+acreción)
Transecto 1	1990-2009 (19 años)	-2.70
Transecto 2	1990-2009 (19 años)	1.59
Transecto 3	1990-2008 (18 años)	1.15
Transecto 4	1990-2008 (18 años)	0.44
Transecto 5	1990-2000 (10 años)	-0.95
Transecto 6	1990-2008 (18 años)	-0.70
Transecto 7	1990-2008 (18 años)	1.05
Transecto 8	1990-2008 (18 años)	-0.11
Transecto 9	1990-2008 (18 años)	-1.73
Transecto 10	1990-2008 (18 años)	-0.99
Transecto 11	1990-2009 (19 años)	-1.31
Transecto 12	1990-2009 (19 años)	-1.25

```
invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(
    titulo = 'Tasa de cambio por año pre-2009 (en metros; -erosión, +acreción) para transectos del Tramo Oeste'))
```

Table 4: Tasa de cambio por año pre-2009 (en metros; -erosión, +acreción) para transectos del Tramo Oeste

Año	Transecto 1	Transecto 2	Transecto 3	Transecto 4	Transecto 5	Transecto 6	Transecto 7	Transecto 8	Transecto 9	Transecto 10	Transecto 11	Transecto 12
1991	-0.26	8.75	7.35	0.51	13.09	4.37	5.64	7.50	5.19	14.19	7.38	3.13
1992	-2.22	-1.27	0.54	-2.56	0.72	-2.84	-4.93	-0.20	-0.21	1.25	1.53	-2.69
1993	-2.70	-3.36	-0.86	-3.62	-2.29	-4.82	-6.53	-2.25	-2.27	-1.34	-0.59	-3.17
1994	-2.70	-3.36	-0.85	-3.62	-2.28	-4.82	-6.53	-2.25	-2.27	-1.35	-0.60	-3.17
1995	-2.70	-3.35	-0.86	-3.62	-2.28	-4.82	-6.52	-2.26	-2.27	-1.34	-0.59	-3.18
1996	-2.71	-3.37	-2.65	-1.51	-3.18	-2.77	-3.40	-9.98	-4.31	-2.49	-0.60	-3.18
1997	-1.45	3.25	-5.21	2.52	-4.60	0.59	1.71	-19.01	-12.34	-4.35	-2.90	1.22
1998	0.41	-1.01	-4.74	5.28	-7.99	-5.96	-7.52	15.62	8.04	-3.60	-3.11	6.35
1999	0.98	1.31	-17.27	-4.90	-10.83	-3.93	-9.55	-15.10	-9.95	-11.50	-11.18	-1.87
2000	0.25	-1.51	16.08	4.36	10.11	-1.98	9.94	4.27	-2.75	17.30	20.57	-5.48
2001	-1.53	5.65	2.94	2.82		0.49	5.88	3.08	-1.73	-0.14	-1.66	-4.66
2002	-3.34	7.71	4.01	1.90		2.18	6.34	2.82	-1.05	-3.84	-3.54	-0.90
2003	-3.33	7.72	4.01	1.90		2.19	6.34	2.83	-1.05	-3.83	-3.53	-0.90
2004	-3.35	7.74	4.02	1.90		2.19	6.36	2.83	-1.06	-3.85	-3.55	-0.90
2005	-3.34	7.72	4.01	1.90		2.18	6.34	2.83	-1.05	-3.84	-3.54	-0.90
2006	-3.33	7.72	4.01	1.90		2.19	6.34	2.83	-1.05	-3.83	-3.53	-0.90
2007	-3.39	7.56	4.17	1.93		2.15	6.34	2.94	-0.94	-3.79	-3.56	-0.90
2008	-7.09	-4.59	2.09	0.88		0.81	2.66	1.50	-0.14	-1.52	-5.38	-0.90
2009	-9.53	-13.16									-6.47	-0.74

Exportar las tablas anteriores a CSV.

```
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(csv = T) %>%
  write_csv('tasas_de_cambio_anual_promediada_pre_2009_por_transecto_tramo_oeste.csv')
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(csv = T) %>%
  write_csv('tasas_de_cambio_anual_pre_2009_por_transecto_tramo_oeste.csv')
```

- Suavizado de la serie con media móvil

```
ventana_de_promediado <- 1 #Número de observaciones para obtener la media móvil (ventana de promediado)
distl_med <- sapply(unique(distances$transect),
  function(x){
```

```

df <- distances[distances$transect==x, ]
df <- df[order(df$date), ]
x <- zoo(df$distance_sign, df$date)
mm <- as.numeric(rollmean(x, ventana_de_promediado, fill = NA))
df$distance_sign <- mm
df <- df #%>% slice(1:(n()-1))
return(df)
}, simplify=F)
interdist_med <- map(distl_med, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances_med <- plyr::ldply(distl_med) %>% mutate(date = as.Date(date, "%Y-%m-%d"))

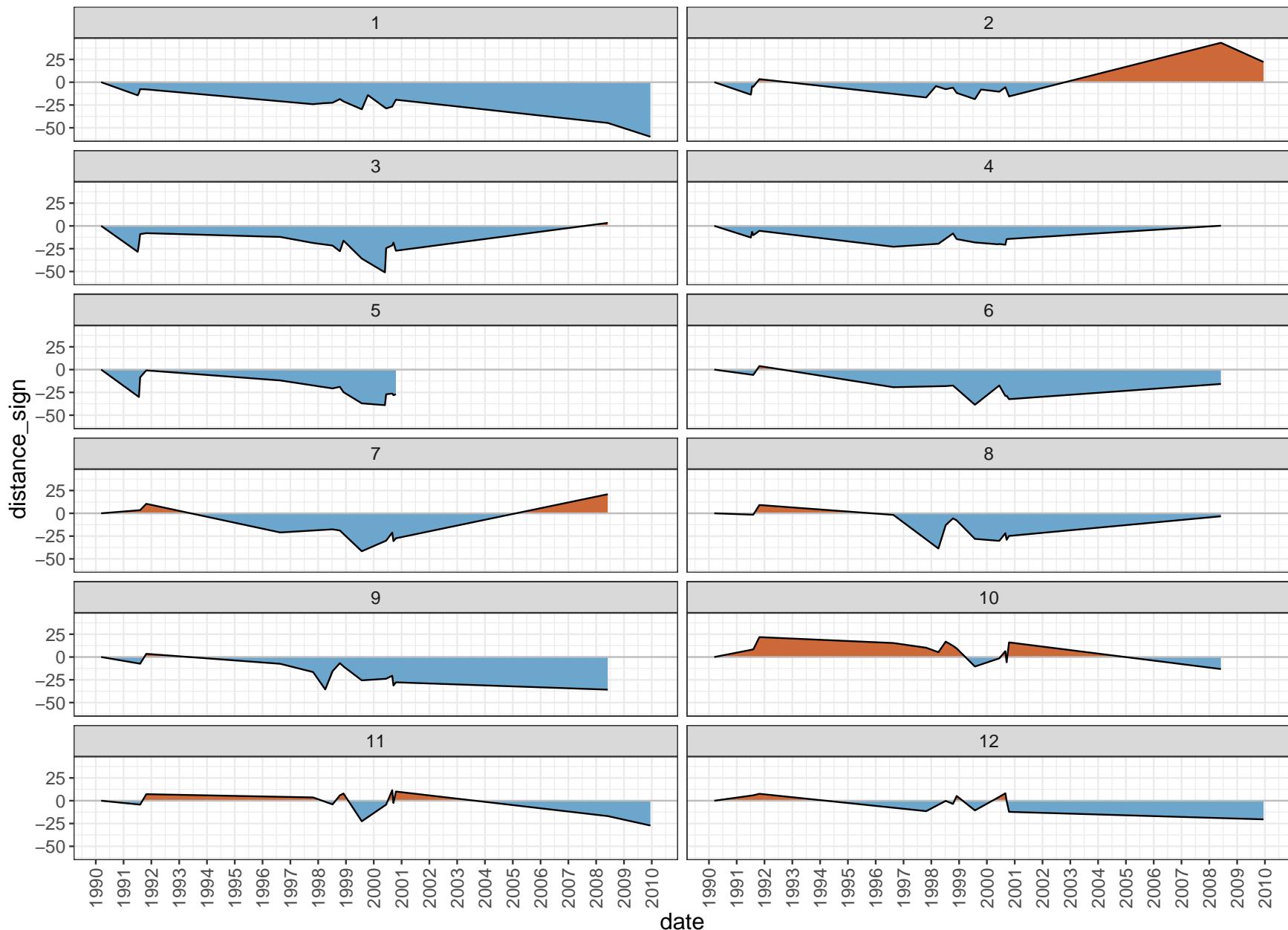
```

- Representación de la serie suavizada

```

distances_med %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist_med, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist_med, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)

```



- Cálculo de EPR (*end point rates*)

```

calcular_epr <- function(fecha_ref = '2013-06-16', fecha_ini = '2013-06-16', fecha_fin = '2015-04-19',
                           campo_trans = 'transect', trans = 1, campo_fecha = 'date',
                           tabla_dist = distances, campo_dist = 'distance_sign'){
  d_trans <- tabla_dist[tabla_dist[, campo_trans] == trans, ]
  selector_fecha <- d_trans[, campo_fecha, drop = T]
#  selector_trans <- d_trans[, campo_trans, drop = T]
  T <- as.numeric(as.Date(fecha_fin, "%Y-%m-%d") - as.Date(fecha_ini, "%Y-%m-%d"))
  d0 <- tryCatch(as.numeric(d_trans[fecha_ini == selector_fecha, campo_dist, drop = T]),)
  d1 <- tryCatch(as.numeric(d_trans[fecha_fin == selector_fecha, campo_dist, drop = T]),)
  D <- tryCatch(d1 - d0)
  EPR <- (D/T)*365
  return(EPR)
}

# EPR anual, periodo 1, 2013-06-16 y 2015-04-19
periodo_1 <- c('2013-06-16', '2015-04-19')
epr_periodo_1 <- data.frame(col = sapply(
  paste('Transecto', (1:15)),
  function(x) calcular_epr(trans = as.integer(gsub('Transecto ', '', x))),
  simplify = T)) %>% setNames(paste('EPR anual, periodo 1, desde', periodo_1[1], 'a', periodo_1[2]))
epr_periodo_1

# EPR anual, periodo 2, 2015-04-19 y 2021-01-29
periodo_2 <- c('2015-04-19', '2021-01-29')
epr_periodo_2 <- data.frame(col = sapply(
  paste('Transecto', (1:15)),
  function(x)
    calcular_epr(
      trans = as.integer(gsub('Transecto ', '', x)),
      fecha_ini = periodo_2[1], fecha_fin = periodo_2[2]),
    simplify = T)) %>% setNames(paste('EPR anual, periodo 2, desde', periodo_2[1], 'a', periodo_2[2]))
epr_periodo_2

# EPR anual, periodo 3, 2021-01-29 y 2021-12-15
periodo_3 <- c('2021-01-29', '2021-12-15')
epr_periodo_3 <- data.frame(col = sapply(
  paste('Transecto', (1:15)),
  function(x)

```

```

calcular_epr(
  trans = as.integer(gsub('Transecto ', '', x)),
  fecha_ini = periodo_3[1], fecha_fin = periodo_3[2]),
simplify = T)) %>% setNames(paste('EPR anual', periodo_2, 'desde', periodo_3[1], 'a', periodo_3[2]))
epr_periodo_3

```

Pos-2013 (Landsat 8)

Tramo Palenque Este

- Cargar líneas de costa

```

lineas <- st_read('lineas-de-costa/PalenqueNizaoPC_L8_output_lines.gpkg') %>%
  filter(grepl('palenque este', tramo, ignore.case = T)) %>%
  st_cast('LINESTRING')

## Reading layer `PalenqueNizaoPC_L8_output_lines` from data source
##   `/home/jose/Documentos/git/tesis-ana-carolain/lineas-de-costa/PalenqueNizaoPC_L8_output_lines.gpkg'
##   using driver `GPKG'
## Simple feature collection with 483 features and 5 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: 368857.5 ymin: 2015230 xmax: 377399 ymax: 2016803
## Projected CRS: WGS 84 / UTM zone 19N

st_geometry(lineas) <- "geometry"
lineas$longitud <- units::drop_units(st_length(lineas))
lineas <- lineas[lineas$longitud > 0, ]

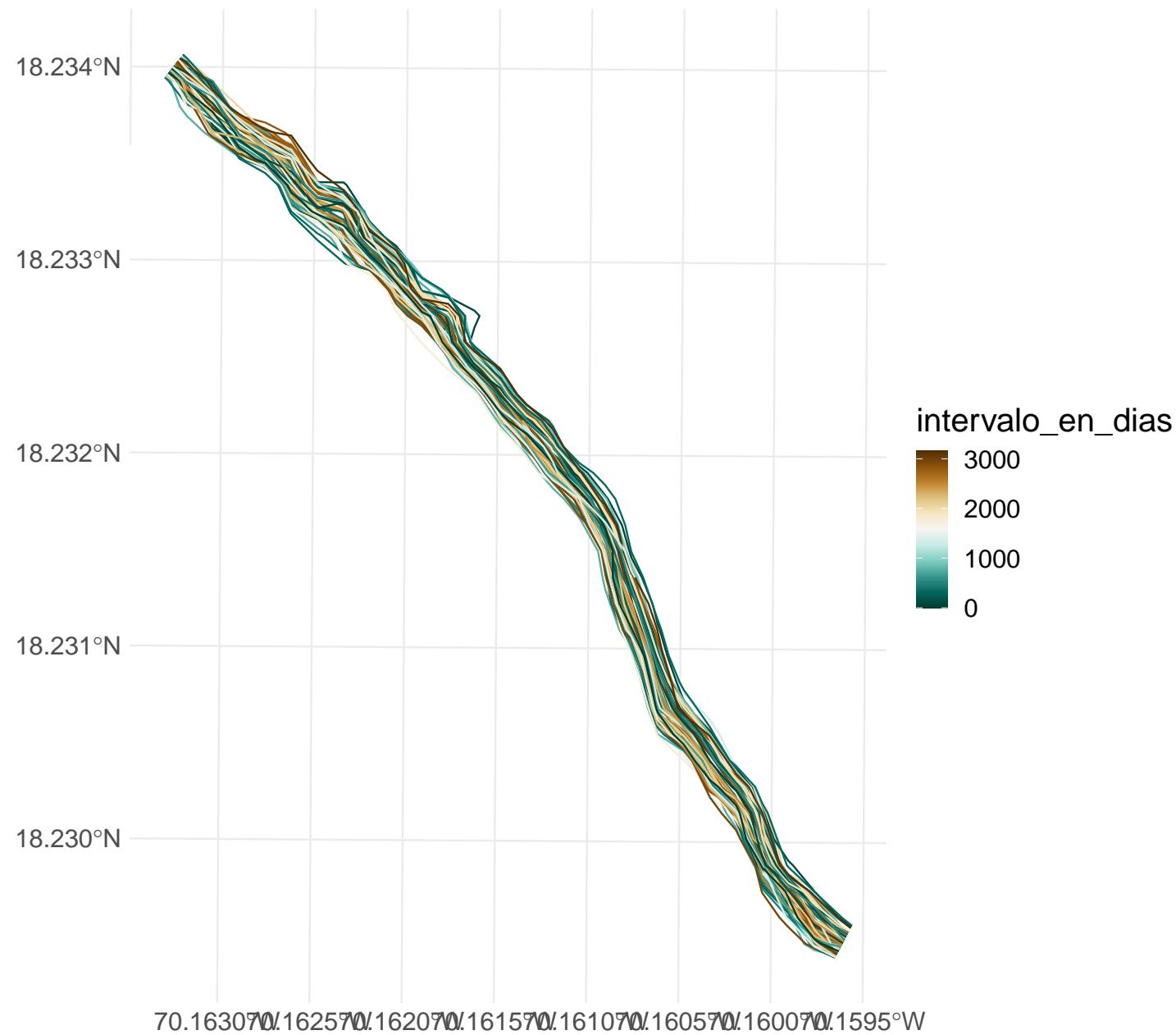
```

- Representar las líneas de costa

```

lineas$intervalo_en_dias <- round(as.numeric(interval(lineas$date, max(lineas$date))), 'days'), 0)
escala_color <- 'BrBG'
mapa_lineas <- lineas %>% ggplot + aes(color=intervalo_en_dias) + geom_sf() +
  theme_minimal() +
  theme(text = element_text(size = 18)) +
  scale_color_gradientn(colors = rev(RColorBrewer::brewer.pal(11, escala_color)))
mapa_lineas

```



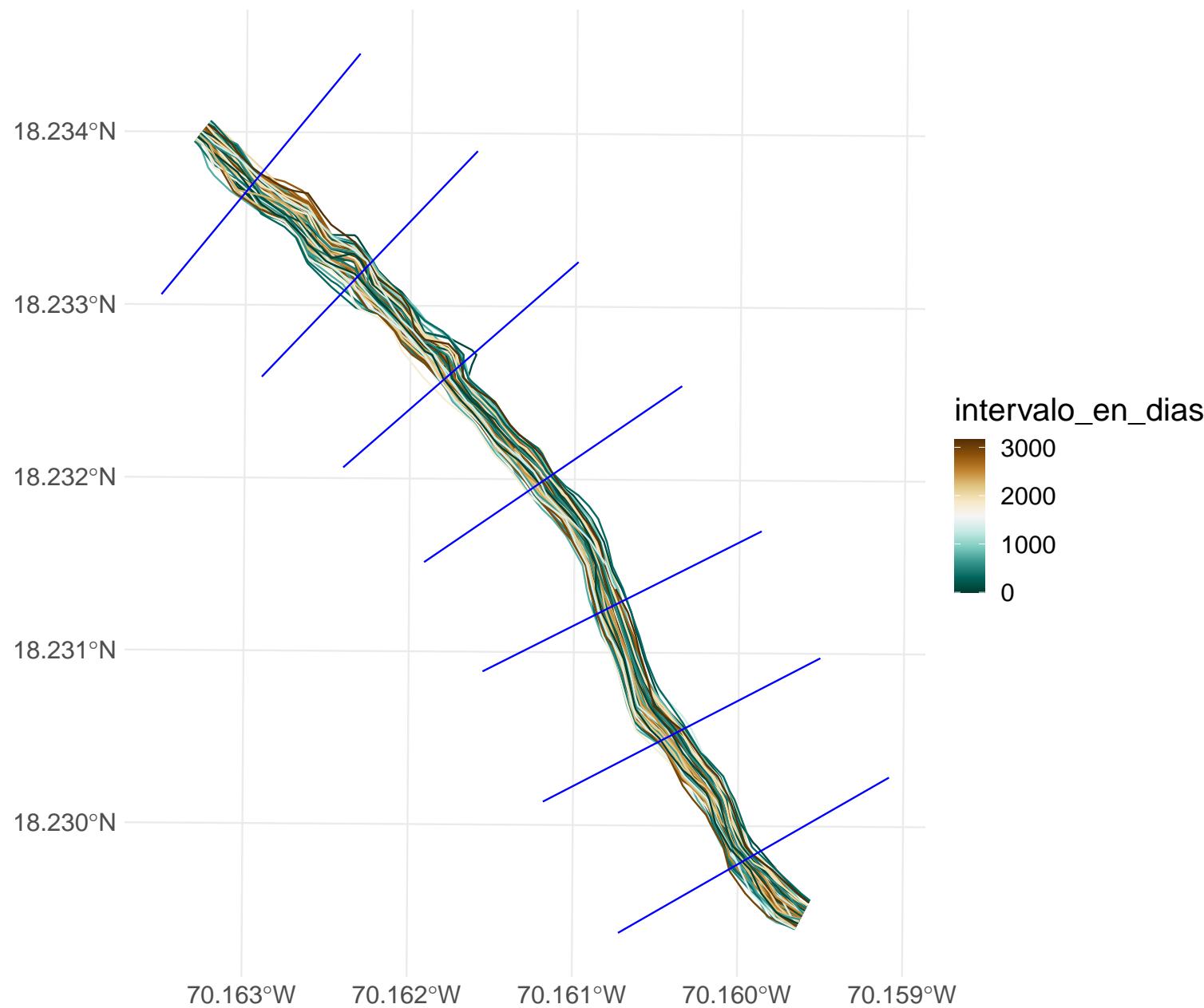
- Crear transectos respecto de línea de costa de referencia y representarlos

```
# Umbral de longitud para líneas que podrían usarse como referencia
umbral_longitud <- 100

# Elegir una línea de referencia
linea_ref <- lineas %>% filter(longitud > umbral_longitud) %>% filter(date == min(date, na.rm = T))

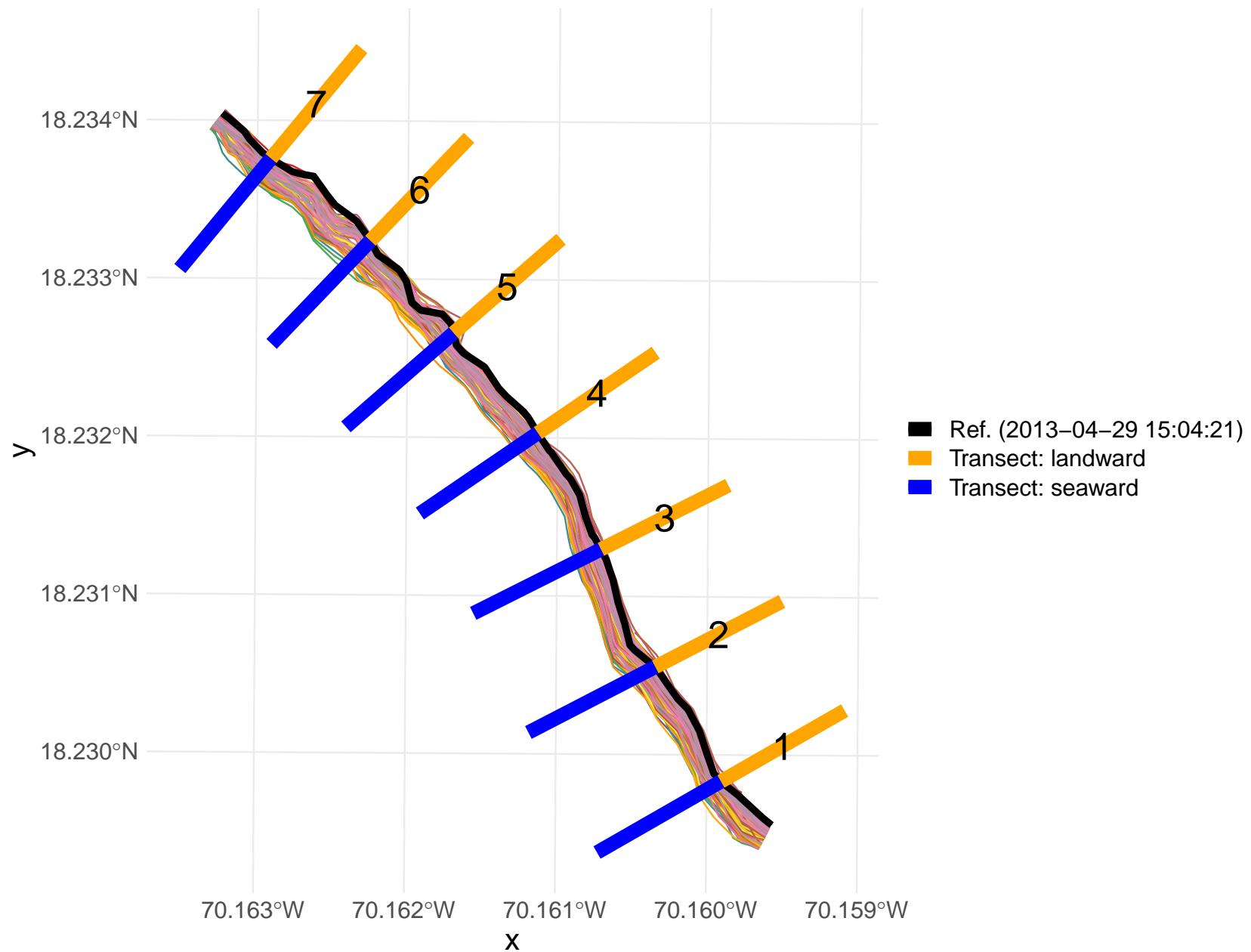
# Crear transectos
transectos <- create_transect(x = linea_ref, 100, reverse = T) %>% rename(transect=coastr_id)

# Mapa
mapa_lineas + geom_sf(data = transectos, color = 'blue')
```



- Clasificar las distintas partes del transecto en tierra o mar

```
transectos_clasif <- transclas(tr = transectos, rl = linea_ref)
cols <- colorRampPalette(brewer.pal(9, 'Set1'))(nrow(lineas))
ggplot() +
  geom_sf(data = lineas %>% mutate(date = factor(date)), color = cols) +
  geom_sf(
    data = linea_ref %>% mutate(linetype = paste0('Ref. (', date, ')')),
    aes(color=linetype), linewidth = 2, show.legend = 'line') +
  geom_sf(
    data = transectos_clasif %>% mutate(sealand=paste0('Transect: ', sealand)),
    aes(color = sealand), show.legend = 'line', linewidth = 4) +
  scale_color_manual(values = c('black', 'orange', 'blue')) +
  geom_sf_text(
    data = transectos_clasif %>% filter(sealand=='landward') %>%
      st_centroid, aes(label = transect), size = 8) +
  theme_minimal() +
  theme(legend.title = element_blank(), text = element_text(size = 18))
```

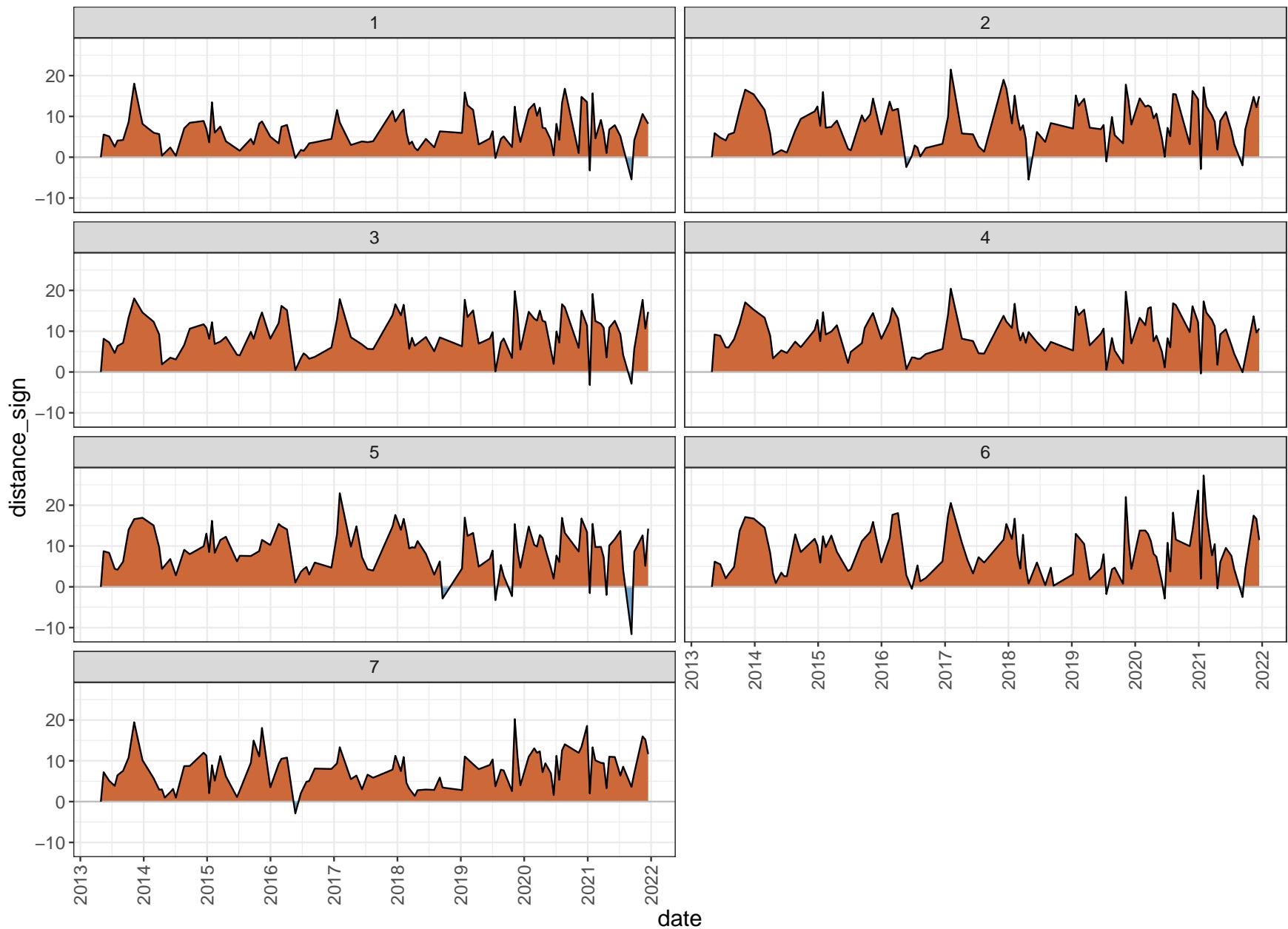


- Calcular distancias de cada línea de costa respecto de la línea de referencia

```
distl <- pointdist(sh = lineas, re = linea_ref, tr = transectos_clasif, rtr = transectos)
```

- Generar las series temporales de distancia de la línea de costa respecto a la de referencia

```
interdist <- map(distl, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances <- plyr::ldply(distl) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
#  scale_y_continuous(limits = c(-30, 30)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)
```



- Revisión de seguridad: determinar si hay transectos que corten dos veces una misma línea de costa

```

test <- sapply(unique(distances$transect), function(x) {
  conteo_cortes <- table(distances[distances$transect==x, 'date', drop=T])
  mas_de_1 <- length(which(conteo_cortes>1))>0
  ifelse(mas_de_1,
    paste('El transecto', x, 'corta', conteo_cortes[which(conteo_cortes>1)],
      'veces la línea de costa de fecha', names(conteo_cortes[which(conteo_cortes>1)])),
    paste('El transecto', x, 'pasa la prueba'))
})
test

## [1] "El transecto 1 pasa la prueba" "El transecto 2 pasa la prueba"
## [3] "El transecto 3 pasa la prueba" "El transecto 4 pasa la prueba"
## [5] "El transecto 5 pasa la prueba" "El transecto 6 pasa la prueba"
## [7] "El transecto 7 pasa la prueba"

```

- Tasas de cambio (-erosión, +acreción)

```

tasas_de_cambio <- lapply(
  unique(distances$transect),
  function (x) calcular_tasa_de_cambio(transecto = x)) %>%
  setNames(paste('Transecto', unique(distances$transect)))
invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(titulo = 'Tasa de cambio anual promediada pos-2013 (L8) para transectos del Tramo Palenque Este'))

```

Table 5: Tasa de cambio anual promediada pos-2013 (L8) para transectos del Tramo Palenque Este

Transecto	Periodo	Tasa de cambio anual, en metros (-erosión,+acreción)
Transecto 1	2013-2021 (8 años)	-0.29
Transecto 2	2013-2021 (8 años)	-0.06
Transecto 3	2013-2021 (8 años)	-0.12
Transecto 4	2013-2021 (8 años)	-0.44
Transecto 5	2013-2021 (8 años)	-0.47
Transecto 6	2013-2021 (8 años)	-0.19
Transecto 7	2013-2021 (8 años)	0.16

```

invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(
    titulo = 'Tasa de cambio por año pos-2013 (en metros; -erosión, +acreción) para transectos del Tramo Palenque Este'))

```

Table 6: Tasa de cambio por año pos-2013 (en metros; -erosión, +acreción) para transectos del Tramo Palenque Este

Año	Transecto 1	Transecto 2	Transecto 3	Transecto 4	Transecto 5	Transecto 6	Transecto 7
2014	-0.44	-1.01	-1.72	-2.18	-1.68	-1.17	-0.35
2015	-1.47	-0.07	1.25	0.96	0.23	1.20	0.52
2016	-0.08	-1.94	-1.94	-1.35	-0.56	-2.20	-0.60
2017	1.68	2.20	2.25	1.37	1.51	0.34	-1.47
2018	0.25	0.06	-1.09	-1.02	-5.10	-4.06	-0.01
2019	0.42	0.94	1.08	0.83	1.90	3.52	2.87
2020	0.79	0.21	0.51	1.85	2.47	3.41	2.10
2021	-3.50	-0.93	-1.28	-4.01	-2.52	-2.57	-1.74

Exportar las tablas anteriores a CSV.

```

tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(csv = T) %>%
  write_csv('tasas_de_cambio_anual_promediada_pos_2013_por_transecto_tramo_palenque_este.csv')
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(csv = T) %>%
  write_csv('tasas_de_cambio_anual_pos_2013_por_transecto_tramo_palenque_este.csv')

```

- Suavizado de la serie con media móvil

```

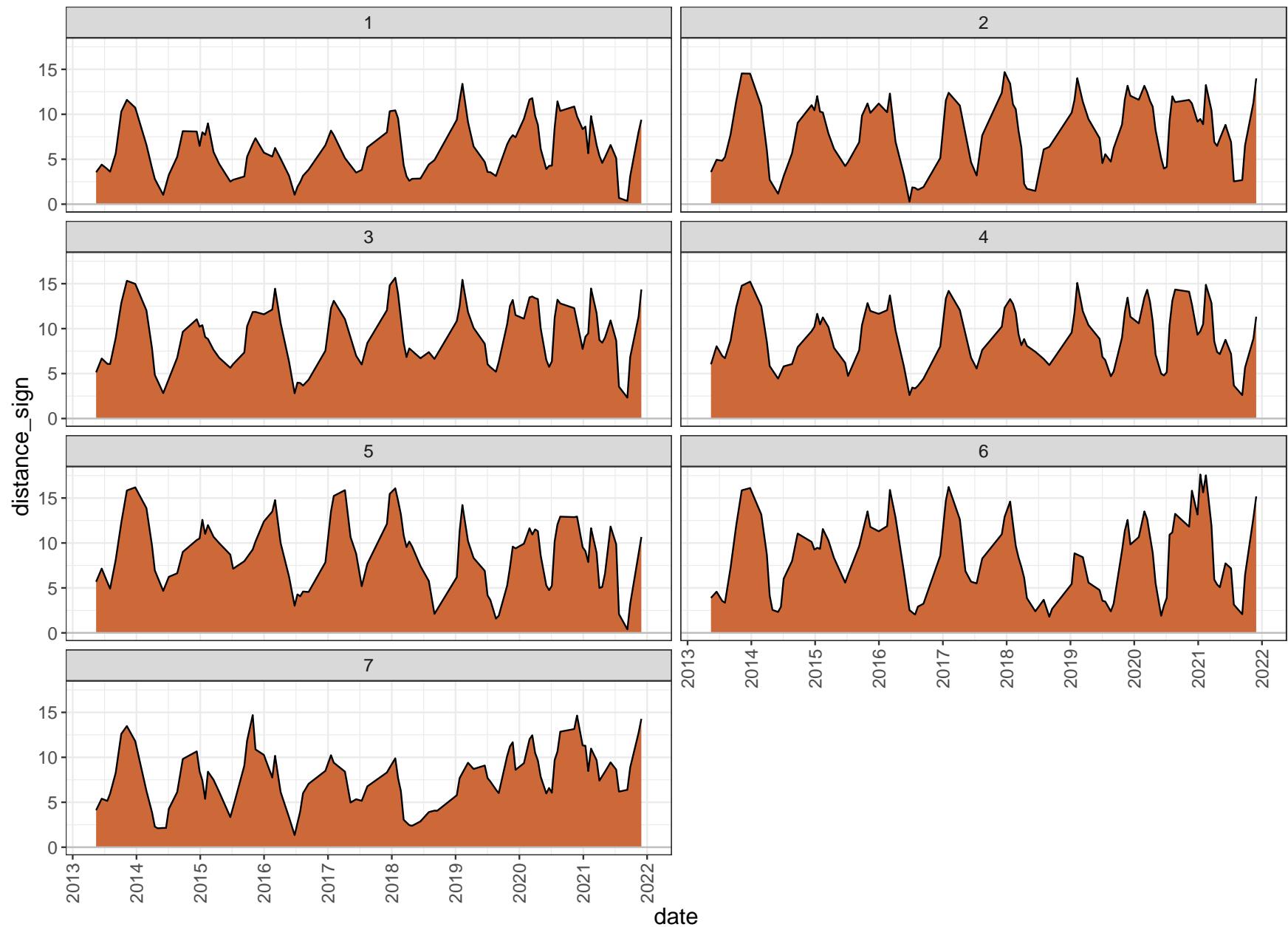
ventana_de_promediado <- 3 #Número de observaciones para obtener la media móvil (ventana de promediado)
dist1_med <- sapply(unique(distances$transect),
  function(x){
    df <- distances[distances$transect==x, ]
    df <- df[order(df$date), ]
    x <- zoo(df$distance_sign, df$date)
    mm <- as.numeric(rollmean(x, ventana_de_promediado, fill = NA))
    df$distance_sign <- mm
    df <- df #%>% slice(1:(n()-1))
    return(df)
  }, simplify=F)

```

```
interdist_med <- map(distl_med, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances_med <- plyr::ldply(distl_med) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
```

- Representación de la serie suavizada

```
distances_med %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist_med, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist_med, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)
```



Tramo Palenque Centro

- Cargar líneas de costa

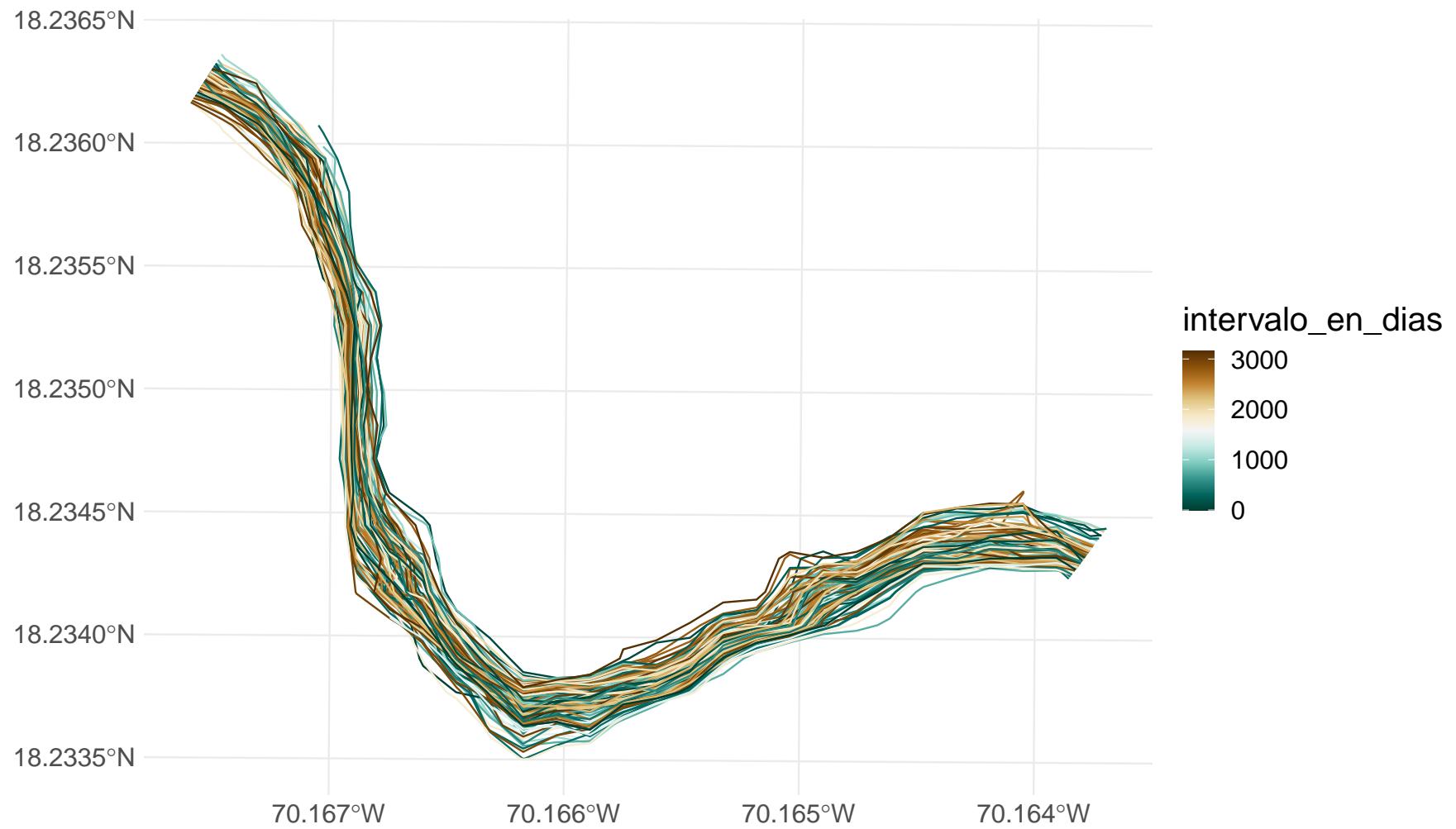
```
lineas <- st_read('lineas-de-costa/PalenqueNizaoPC_L8_output_lines.gpkg') %>%
  filter(grepl('palenque centro', tramo, ignore.case = T)) %>%
  st_cast('LINESTRING')

## Reading layer `PalenqueNizaoPC_L8_output_lines` from data source
##   `/home/jose/Documentos/git/tesis-ana-carolain/lineas-de-costa/PalenqueNizaoPC_L8_output_lines.gpkg'
##   using driver `GPKG'
## Simple feature collection with 483 features and 5 fields
## Geometry type: LINESTRING
## Dimension:      XY
## Bounding box:  xmin: 368857.5 ymin: 2015230 xmax: 377399 ymax: 2016803
## Projected CRS: WGS 84 / UTM zone 19N

st_geometry(lineas) <- "geometry"
lineas$longitud <- units::drop_units(st_length(lineas))
lineas <- lineas[lineas$longitud > 0, ]
```

- Representar las líneas de costa

```
lineas$intervalo_en_dias <- round(as.numeric(interval(lineas$date, max(lineas$date))), 'days'), 0)
escala_color <- 'BrBG'
mapa_lineas <- lineas %>% ggplot + aes(color=intervalo_en_dias) + geom_sf() +
  theme_minimal() +
  theme(text = element_text(size = 18)) +
  scale_color_gradientn(colors = rev(RColorBrewer::brewer.pal(11, escala_color)))
mapa_lineas
```



- Crear transectos respecto de línea de costa de referencia y representarlos

```
# Umbral de longitud para líneas que podrían usarse como referencia
umbral_longitud <- 300

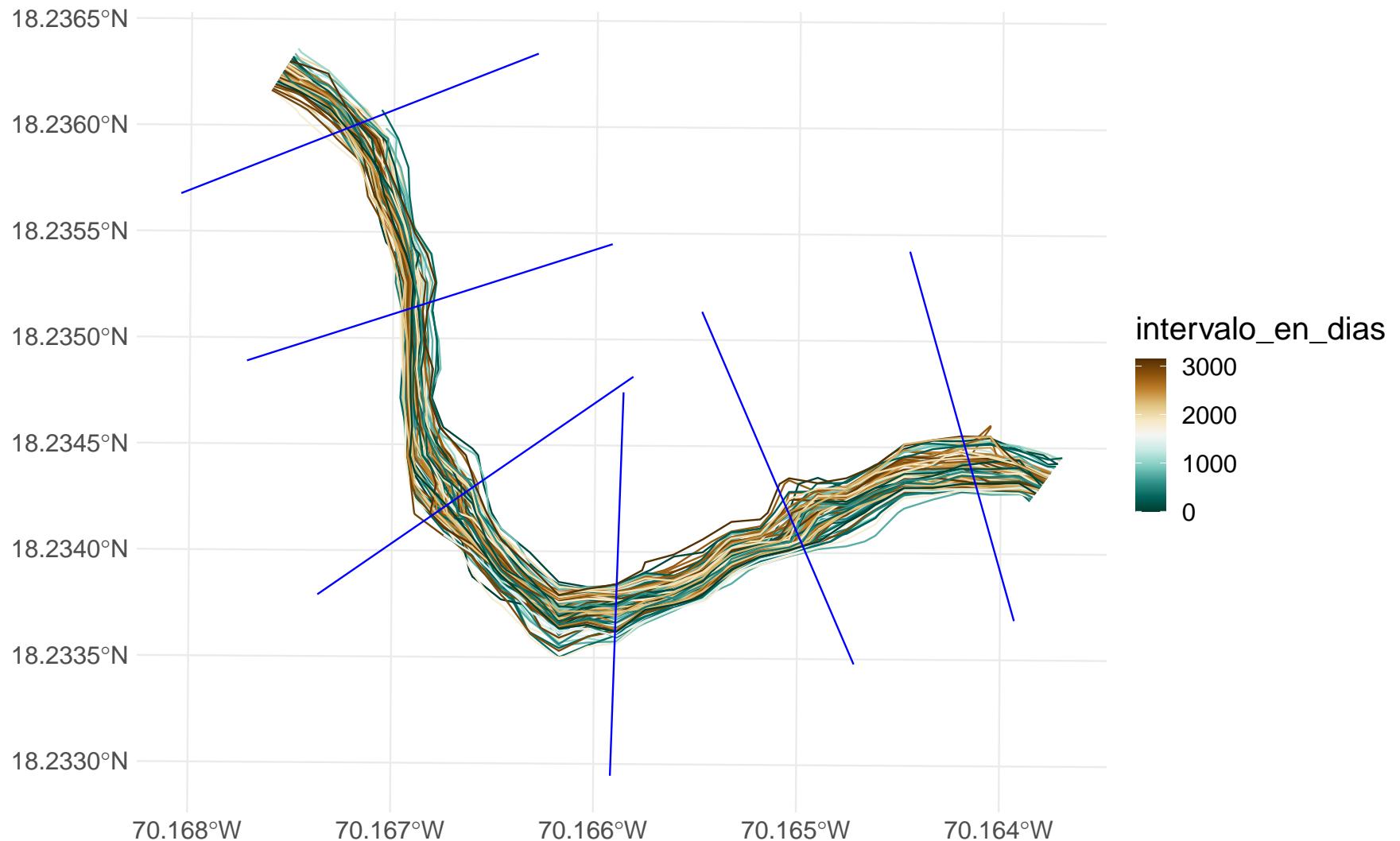
# Elegir una línea de referencia
linea_ref <- lineas %>% filter(longitud > umbral_longitud) %>% filter(date == min(date, na.rm = T))
```

```

# Crear transectos
transectos <- create_transect(x = linea_ref, 100, reverse = T) %>% rename(transect=coastr_id)

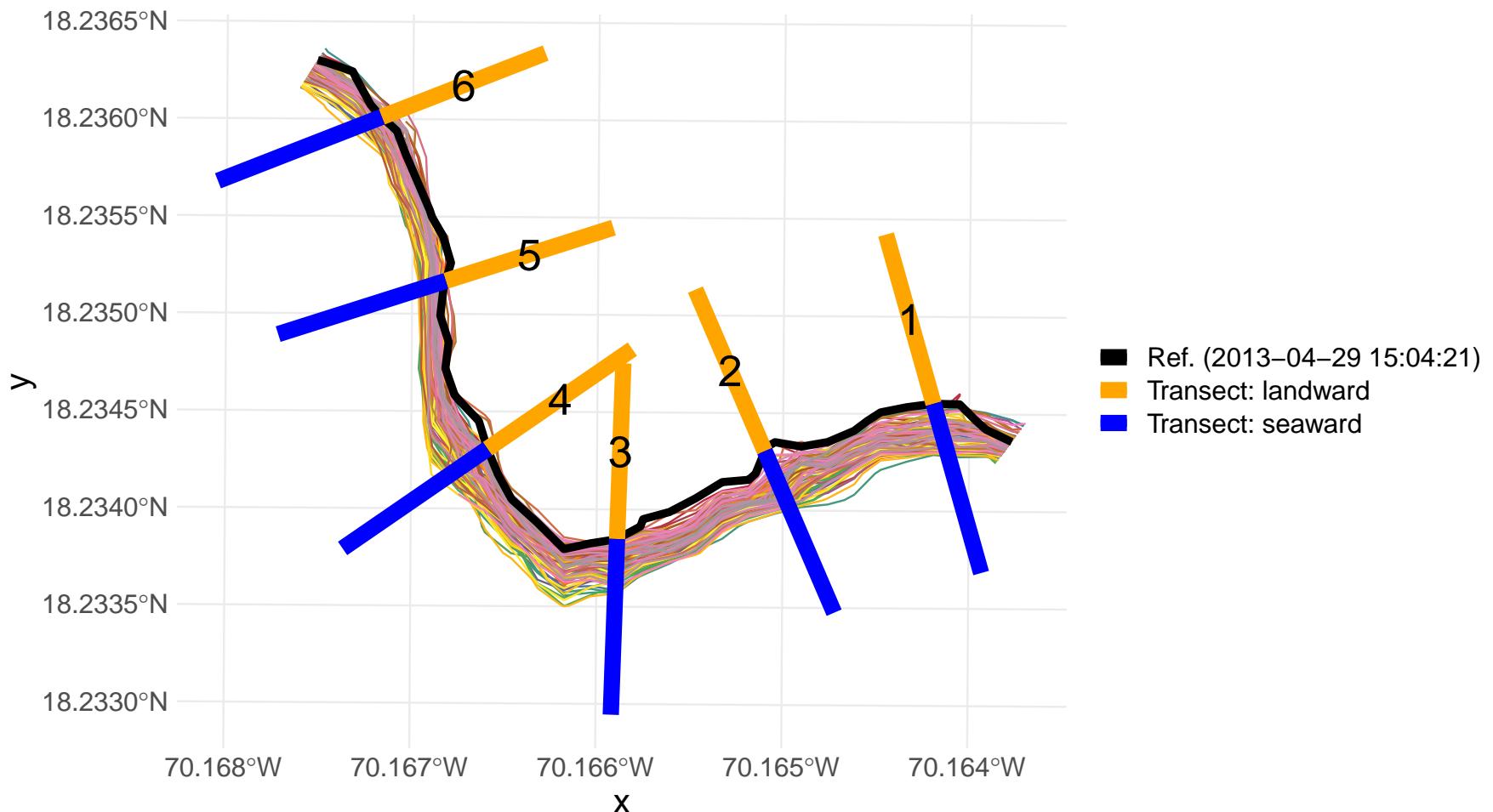
# Mapa
mapa_lineas + geom_sf(data = transectos, color = 'blue')

```



- Clasificar las distintas partes del transecto en tierra o mar

```
transectos_clasif <- transclas(tr = transectos, rl = linea_ref)
cols <- colorRampPalette(brewer.pal(9, 'Set1'))(nrow(lineas))
ggplot() +
  geom_sf(data = lineas %>% mutate(date = factor(date)), color = cols) +
  geom_sf(
    data = linea_ref %>% mutate(linetype = paste0('Ref. (', date, ')')),
    aes(color=linetype), linewidth = 2, show.legend = 'line') +
  geom_sf(
    data = transectos_clasif %>% mutate(sealand=paste0('Transect: ', sealand)),
    aes(color = sealand), show.legend = 'line', linewidth = 4) +
  scale_color_manual(values = c('black', 'orange', 'blue')) +
  geom_sf_text(
    data = transectos_clasif %>% filter(sealand=='landward') %>%
      st_centroid, aes(label = transect), size = 8) +
  theme_minimal() +
  theme(legend.title = element_blank(), text = element_text(size = 18))
```



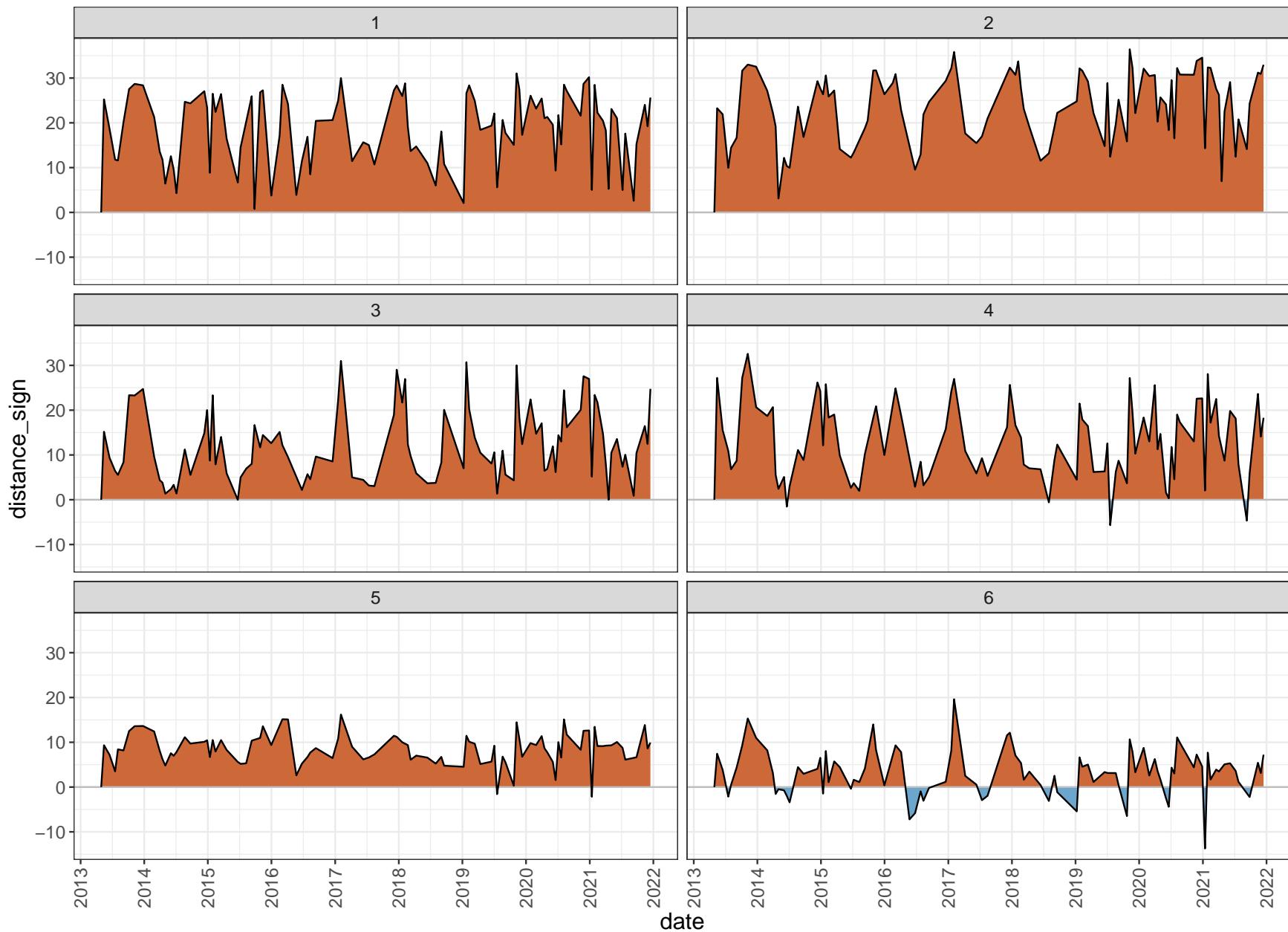
- Calcular distancias de cada línea de costa respecto de la línea de referencia

```
distl <- pointdist(sh = lineas, re = linea_ref, tr = transectos_clasif, rtr = transectos)
```

- Generar las series temporales de distancia de la línea de costa respecto a la de referencia

```
interdist <- map(distl, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances <- plyr::ldply(distl) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
```

```
geom_ribbon(data = interdist, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
#  scale_y_continuous(limits = c(-30, 30)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)
```



- Revisión de seguridad: determinar si hay transectos que corten dos veces una misma línea de costa

```

test <- sapply(unique(distances$transect), function(x) {
  conteo_cortes <- table(distances[distances$transect==x, 'date', drop=T])
  mas_de_1 <- length(which(conteo_cortes>1))>0
  ifelse(mas_de_1,
    paste('El transecto', x, 'corta', conteo_cortes[which(conteo_cortes>1)],
      'veces la línea de costa de fecha', names(conteo_cortes[which(conteo_cortes>1)])),
    paste('El transecto', x, 'pasa la prueba'))
})
test

## [1] "El transecto 1 pasa la prueba" "El transecto 2 pasa la prueba"
## [3] "El transecto 3 pasa la prueba" "El transecto 4 pasa la prueba"
## [5] "El transecto 5 pasa la prueba" "El transecto 6 pasa la prueba"

```

- Tasas de cambio (-erosión, +acreción)

```

tasas_de_cambio <- lapply(
  unique(distances$transect),
  function (x) calcular_tasa_de_cambio(transecto = x)) %>%
  setNames(paste('Transecto', unique(distances$transect)))
invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(titulo = 'Tasa de cambio anual promediada pos-2013 (L8) para transectos del Tramo Palenque Centro'))

```

Table 7: Tasa de cambio anual promediada pos-2013 (L8) para transectos del Tramo Palenque Centro

Transecto	Periodo	Tasa de cambio anual, en metros (-erosión,+acreción)
Transecto 1	2013-2021 (8 años)	-0.66
Transecto 2	2013-2021 (8 años)	0.10
Transecto 3	2013-2021 (8 años)	-0.37
Transecto 4	2013-2021 (8 años)	-0.60
Transecto 5	2013-2021 (8 años)	-0.12
Transecto 6	2013-2021 (8 años)	-0.47

```

invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(

```

```
titulo = 'Tasa de cambio por año pos-2013 (en metros; -erosión, +acreción) para transectos del Tramo Palenque Centro'))
```

Table 8: Tasa de cambio por año pos-2013 (en metros; -erosión, +acreción) para transectos del Tramo Palenque Centro

Año	Transecto 1	Transecto 2	Transecto 3	Transecto 4	Transecto 5	Transecto 6
2014	0.18	-2.05	-3.52	-2.89	-1.28	-3.10
2015	-4.15	1.84	0.88	-0.44	1.20	1.02
2016	2.27	0.89	0.38	-1.07	-1.41	-0.97
2017	0.07	-0.15	1.25	-1.28	-0.28	1.18
2018	-4.28	-1.16	0.36	-2.04	-2.28	-3.94
2019	6.56	2.64	0.47	2.77	1.16	2.83
2020	0.55	1.95	3.65	3.90	2.09	1.03
2021	-6.44	-3.19	-6.45	-3.73	-0.16	-1.77

Exportar las tablas anteriores a CSV.

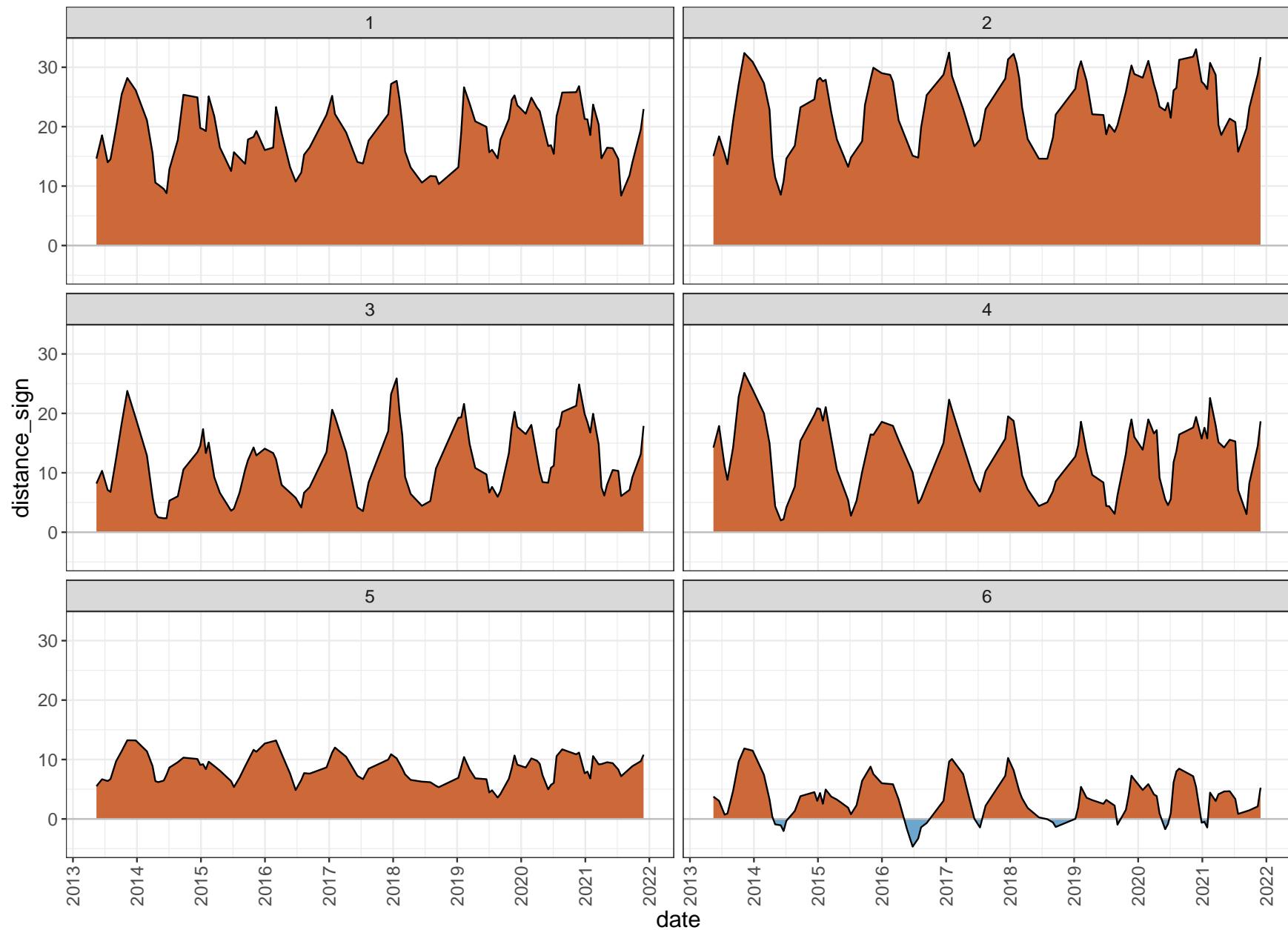
```
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(csv = T) %>%
  write_csv('tasas_de_cambio_anual_promediada_pos_2013_por_transecto_tramo_palenque_centro.csv')
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(csv = T) %>%
  write_csv('tasas_de_cambio_anual_pos_2013_por_transecto_tramo_palenque_centro.csv')
```

- Suavizado de la serie con media móvil

```
ventana_de_promediado <- 3 #Número de observaciones para obtener la media móvil (ventana de promediado)
distl_med <- sapply(unique(distances$transect),
  function(x){
    df <- distances[distances$transect==x, ]
    df <- df[order(df$date), ]
    x <- zoo(df$distance_sign, df$date)
    mm <- as.numeric(rollmean(x, ventana_de_promediado, fill = NA))
    df$distance_sign <- mm
    df <- df #%>% slice(1:(n()-1))
    return(df)
  }), simplify=F)
interdist_med <- map(distl_med, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances_med <- plyr::ldply(distl_med) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
```

- Representación de la serie suavizada

```
distances_med %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist_med, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist_med, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)
```



Tramo Palenque Oeste

- Cargar líneas de costa

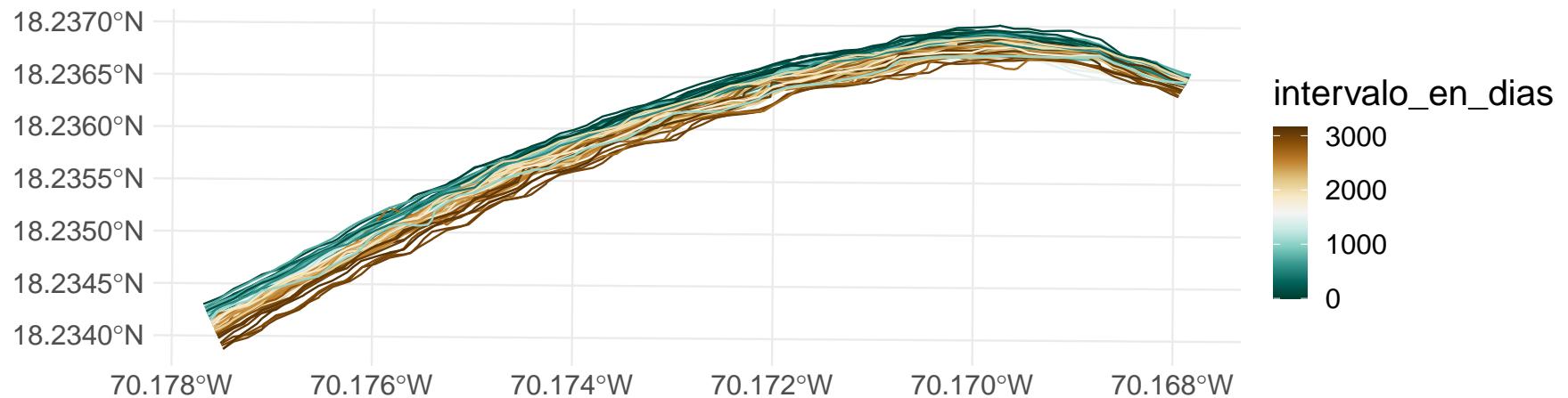
```
lineas <- st_read('lineas-de-costa/PalenqueNizaoPC_L8_output_lines.gpkg') %>%
  filter(grepl('palenque oeste', tramo, ignore.case = T)) %>%
  st_cast('LINESTRING')

## Reading layer `PalenqueNizaoPC_L8_output_lines` from data source
##   `/home/jose/Documentos/git/tesis-ana-carolain/lineas-de-costa/PalenqueNizaoPC_L8_output_lines.gpkg'
##   using driver `GPKG'
## Simple feature collection with 483 features and 5 fields
## Geometry type: LINESTRING
## Dimension:      XY
## Bounding box:  xmin: 368857.5 ymin: 2015230 xmax: 377399 ymax: 2016803
## Projected CRS: WGS 84 / UTM zone 19N

st_geometry(lineas) <- "geometry"
lineas$longitud <- units::drop_units(st_length(lineas))
lineas <- lineas[lineas$longitud > 0, ]
```

- Representar las líneas de costa

```
lineas$intervalo_en_dias <- round(as.numeric(interval(lineas$date, max(lineas$date))), 'days'), 0)
escala_color <- 'BrBG'
mapa_lineas <- lineas %>% ggplot + aes(color=intervalo_en_dias) + geom_sf() +
  theme_minimal() +
  theme(text = element_text(size = 18)) +
  scale_color_gradientn(colors = rev(RColorBrewer::brewer.pal(11, escala_color)))
mapa_lineas
```



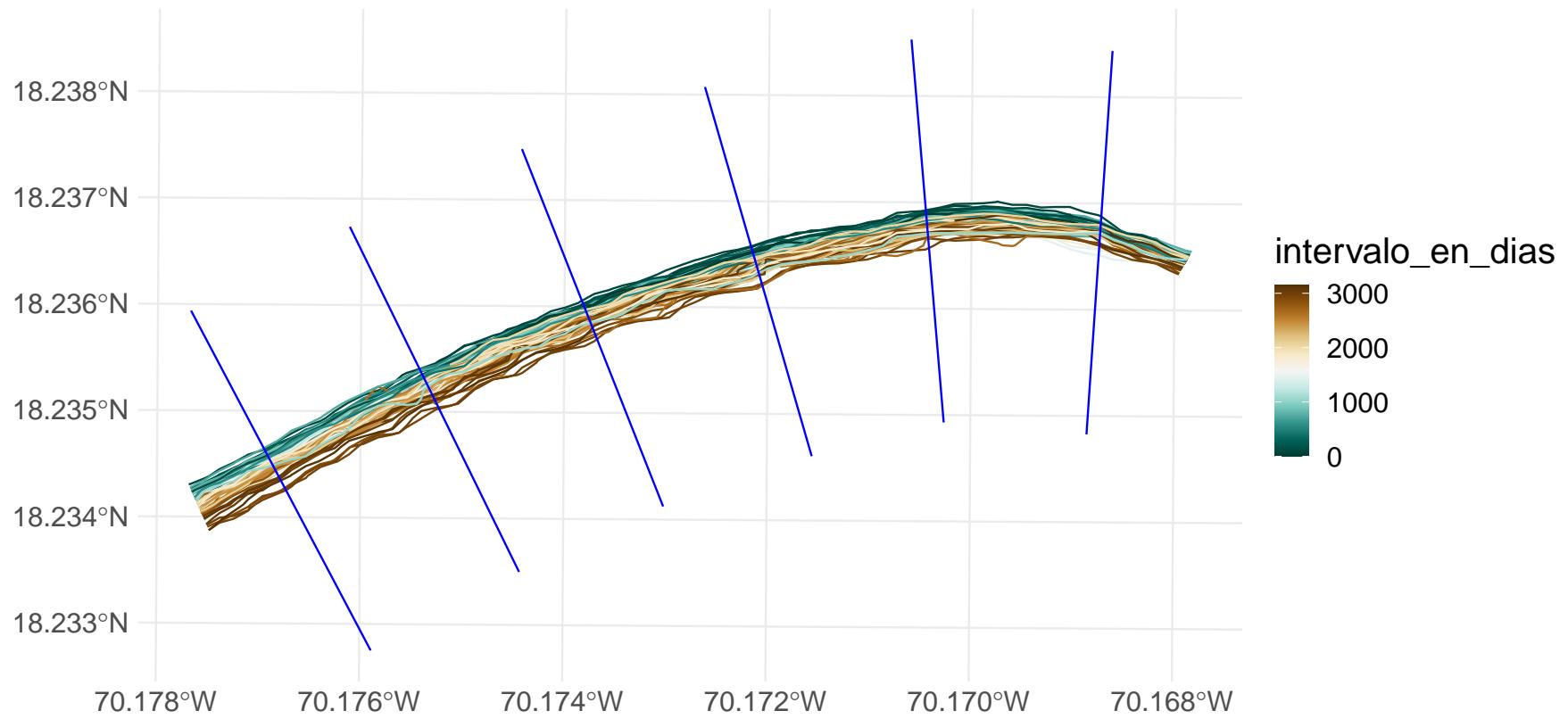
- Crear transectos respecto de línea de costa de referencia y representarlos

```
# Umbral de longitud para líneas que podrían usarse como referencia
umbral_longitud <- 800

# Elegir una línea de referencia
linea_ref <- lineas %>% filter(longitud > umbral_longitud) %>% filter(date == min(date, na.rm = T))

# Crear transectos
transectos <- create_transect(x = linea_ref, 200, reverse = T) %>% rename(transect=coastr_id)

# Mapa
mapa_lineas + geom_sf(data = transectos, color = 'blue')
```



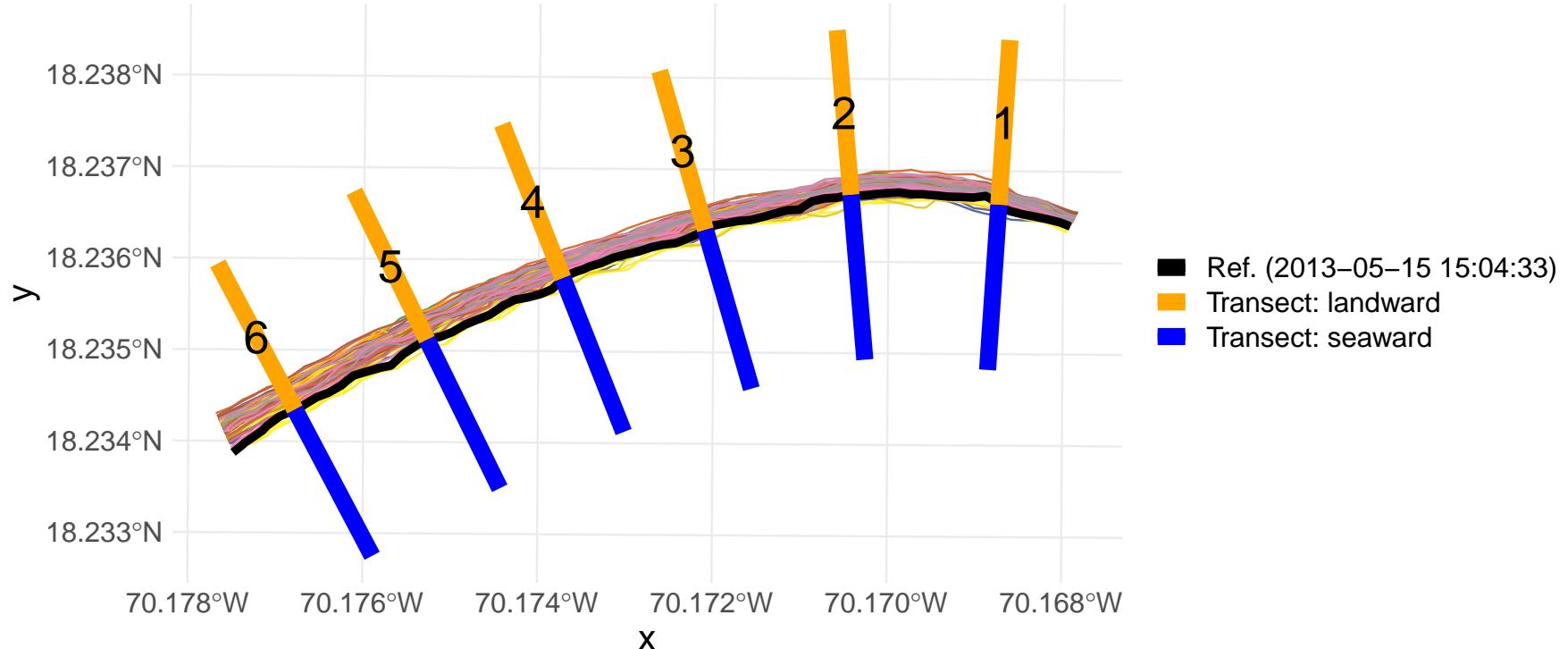
- Clasificar las distintas partes del transecto en tierra o mar

```
transectos_clasif <- transclas(tr = transectos, rl = linea_ref)
cols <- colorRampPalette(brewer.pal(9, 'Set1'))(nrow(lineas))
ggplot() +
  geom_sf(data = lineas %>% mutate(date = factor(date)), color = cols) +
  geom_sf(
    data = linea_ref %>% mutate(linetype = paste0('Ref. (', date, ')')),
    aes(color=linetype), linewidth = 2, show.legend = 'line') +
  geom_sf(
    data = transectos_clasif %>% mutate(sealand=paste0('Transect: ', sealand)),
    aes(color = sealand), show.legend = 'line', linewidth = 4) +
  scale_color_manual(values = c('black', 'orange', 'blue')) +
  geom_sf_text(
```

```

data = transectos_clasif %>% filter(sealand=='landward') %>%
  st_centroid, aes(label = transect), size = 8) +
theme_minimal() +
theme(legend.title = element_blank(), text = element_text(size = 18))

```



- Calcular distancias de cada línea de costa respecto de la línea de referencia

```
distl <- pointdist(sh = lineas, re = linea_ref, tr = transectos_clasif, rtr = transectos)
```

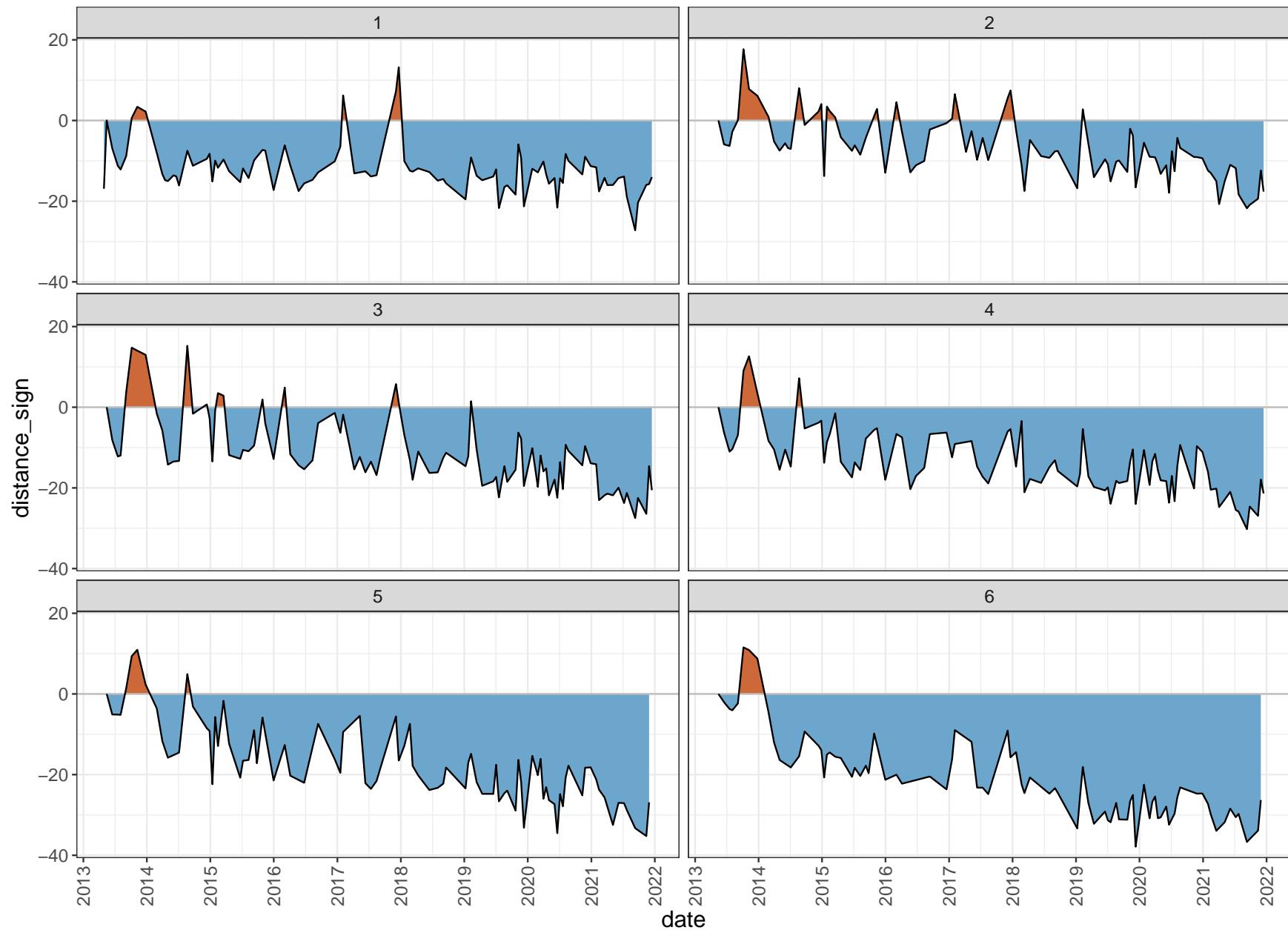
- Generar las series temporales de distancia de la línea de costa respecto a la de referencia

```

interdist <- map(distl, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances <- plyr::ldply(distl) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +

```

```
geom_ribbon(data = interdist, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
#  scale_y_continuous(limits = c(-30, 30)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)
```



- Revisión de seguridad: determinar si hay transectos que corten dos veces una misma línea de costa

```

test <- sapply(unique(distances$transect), function(x) {
  conteo_cortes <- table(distances[distances$transect==x, 'date', drop=T])
  mas_de_1 <- length(which(conteo_cortes>1))>0
  ifelse(mas_de_1,
    paste('El transecto', x, 'corta', conteo_cortes[which(conteo_cortes>1)],
      'veces la línea de costa de fecha', names(conteo_cortes[which(conteo_cortes>1)])),
    paste('El transecto', x, 'pasa la prueba'))
})
test

## [1] "El transecto 1 pasa la prueba" "El transecto 2 pasa la prueba"
## [3] "El transecto 3 pasa la prueba" "El transecto 4 pasa la prueba"
## [5] "El transecto 5 pasa la prueba" "El transecto 6 pasa la prueba"

```

- Tasas de cambio (-erosión, +acreción)

```

tasas_de_cambio <- lapply(
  unique(distances$transect),
  function (x) calcular_tasa_de_cambio(transecto = x)) %>%
  setNames(paste('Transecto', unique(distances$transect)))
invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(titulo = 'Tasa de cambio anual promediada pos-2013 (L8) para transectos del Tramo Palenque Oeste'))

```

Table 9: Tasa de cambio anual promediada pos-2013 (L8) para transectos del Tramo Palenque Oeste

Transecto	Periodo	Tasa de cambio anual, en metros (-erosión,+acreción)
Transecto 1	2013-2021 (8 años)	-1.76
Transecto 2	2013-2021 (8 años)	-2.77
Transecto 3	2013-2021 (8 años)	-3.43
Transecto 4	2013-2021 (8 años)	-3.17
Transecto 5	2013-2021 (8 años)	-4.14
Transecto 6	2013-2021 (8 años)	-4.13

```

invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(

```

```
titulo = 'Tasa de cambio por año pos-2013 (en metros; -erosión, +acreción) para transectos del Tramo Palenque Oeste'))
```

Table 10: Tasa de cambio por año pos-2013 (en metros; -erosión, +acreción) para transectos del Tramo Palenque Oeste

Año	Transecto 1	Transecto 2	Transecto 3	Transecto 4	Transecto 5	Transecto 6
2014	-5.26	-2.60	-4.19	-4.05	-6.91	-12.96
2015	-0.45	-3.67	-4.70	-4.46	-7.12	-4.10
2016	1.89	1.62	-0.31	1.64	3.25	1.18
2017	3.05	-1.39	-1.14	-3.62	-3.25	-1.17
2018	-8.39	-5.20	-4.33	-3.45	-5.51	-8.42
2019	0.60	-0.46	-2.39	-0.25	-2.27	-1.86
2020	1.33	-1.36	-0.87	-0.41	0.01	1.54
2021	-6.88	-9.07	-9.48	-10.78	-11.31	-7.23

Exportar las tablas anteriores a CSV.

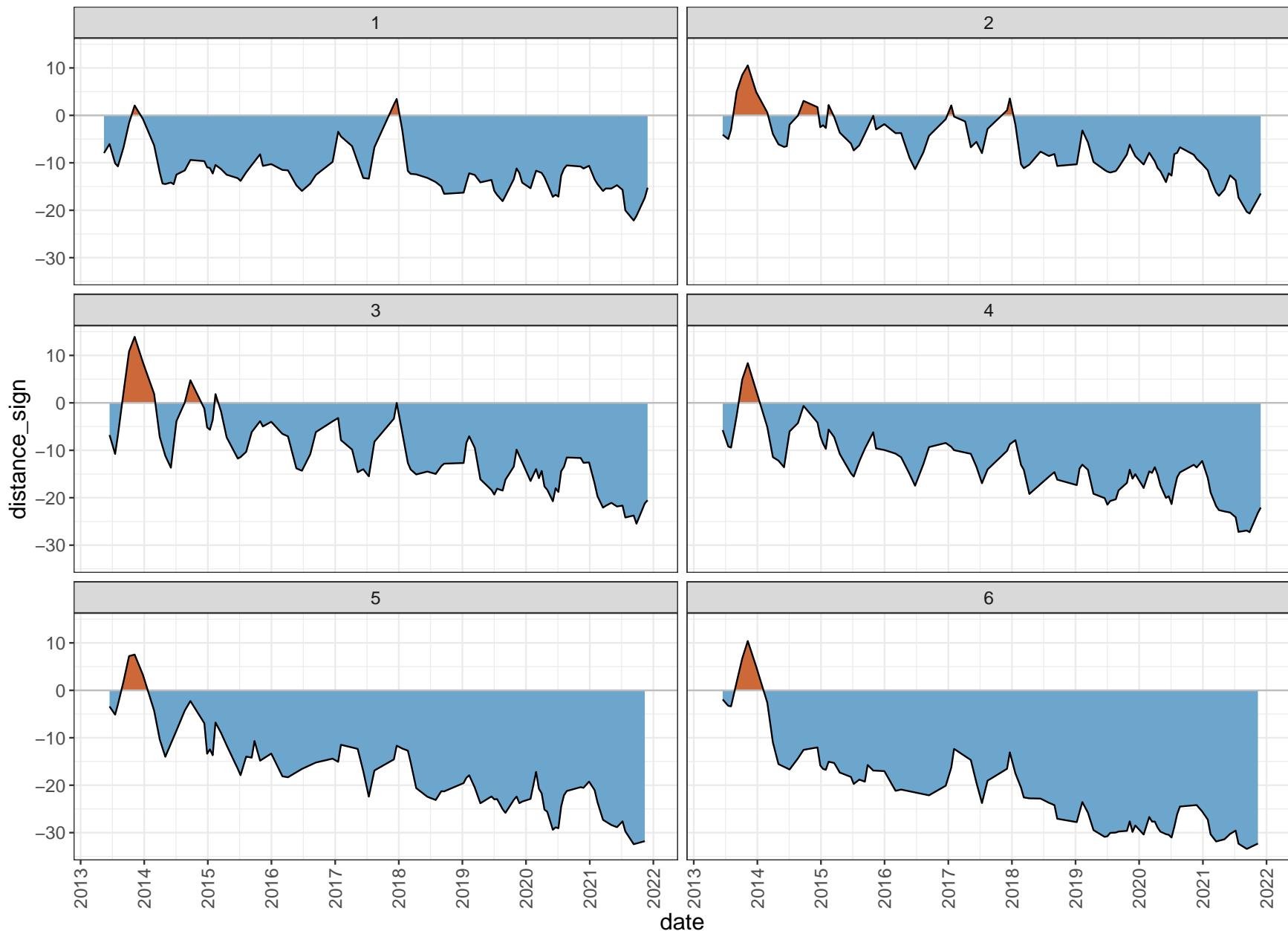
```
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(csv = T) %>%
  write_csv('tasas_de_cambio_anual_promediada_pos_2013_por_transecto_tramo_palenque_oeste.csv')
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(csv = T) %>%
  write_csv('tasas_de_cambio_anual_pos_2013_por_transecto_tramo_palenque_oeste.csv')
```

- Suavizado de la serie con media móvil

```
ventana_de_promediado <- 3 #Número de observaciones para obtener la media móvil (ventana de promediado)
distl_med <- sapply(unique(distances$transect),
  function(x){
    df <- distances[distances$transect==x, ]
    df <- df[order(df$date), ]
    x <- zoo(df$distance_sign, df$date)
    mm <- as.numeric(rollmean(x, ventana_de_promediado, fill = NA))
    df$distance_sign <- mm
    df <- df #%>% slice(1:(n()-1))
    return(df)
  }), simplify=F)
interdist_med <- map(distl_med, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances_med <- plyr::ldply(distl_med) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
```

- Representación de la serie suavizada

```
distances_med %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist_med, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist_med, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)
```



Tramo Nizao este

- Cargar líneas de costa

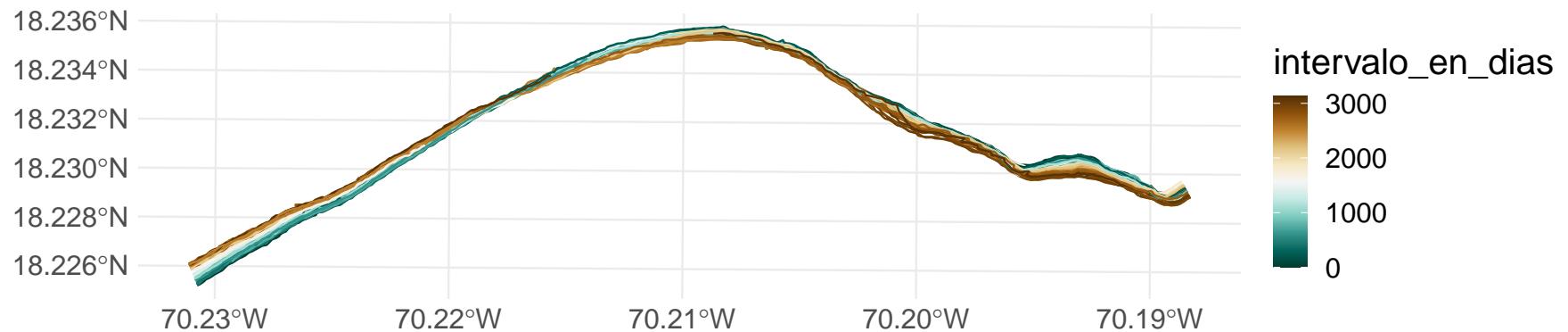
```
lineas <- st_read('lineas-de-costa/PalenqueNizaoPC_L8_output_lines.gpkg') %>%
  filter(grepl('nizao este', tramo, ignore.case = T)) %>%
  st_cast('LINESTRING')

## Reading layer `PalenqueNizaoPC_L8_output_lines` from data source
##   `/home/jose/Documentos/git/tesis-ana-carolain/lineas-de-costa/PalenqueNizaoPC_L8_output_lines.gpkg'
##   using driver `GPKG'
## Simple feature collection with 483 features and 5 fields
## Geometry type: LINESTRING
## Dimension:      XY
## Bounding box:  xmin: 368857.5 ymin: 2015230 xmax: 377399 ymax: 2016803
## Projected CRS: WGS 84 / UTM zone 19N

st_geometry(lineas) <- "geometry"
lineas$longitud <- units::drop_units(st_length(lineas))
lineas <- lineas[lineas$longitud > 0, ]
```

- Representar las líneas de costa

```
lineas$intervalo_en_dias <- round(as.numeric(interval(lineas$date, max(lineas$date))), 'days'), 0)
escala_color <- 'BrBG'
mapa_lineas <- lineas %>% ggplot + aes(color=intervalo_en_dias) + geom_sf() +
  theme_minimal() +
  theme(text = element_text(size = 18)) +
  scale_color_gradientn(colors = rev(RColorBrewer::brewer.pal(11, escala_color)))
mapa_lineas
```



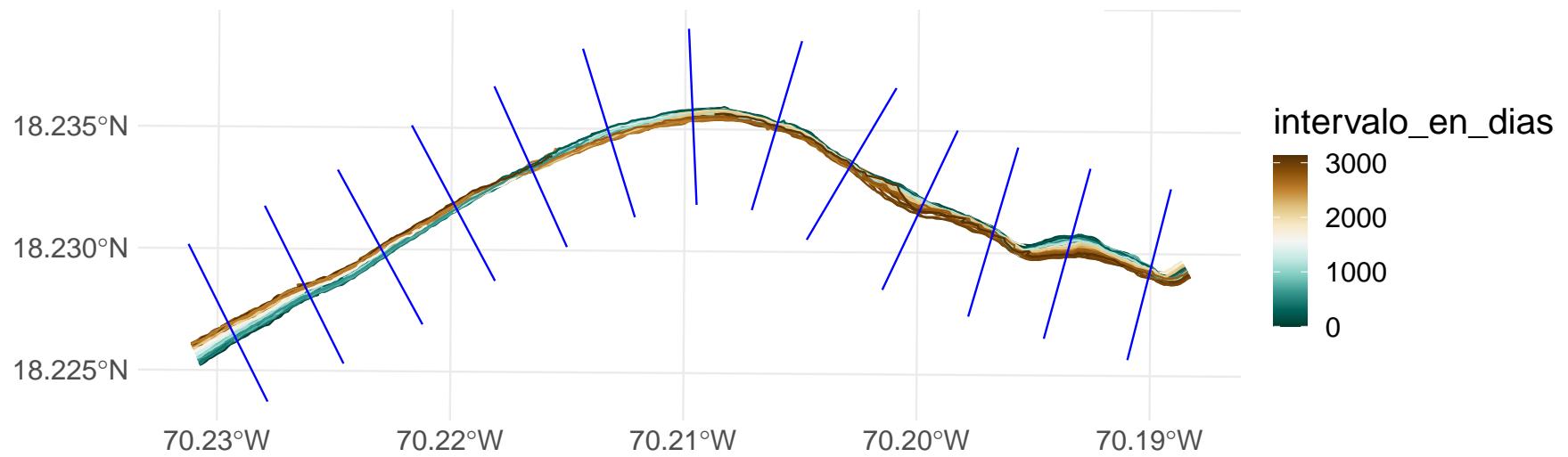
- Crear transectos respecto de línea de costa de referencia y representarlos

```
# Umbral de longitud para líneas que podrían usarse como referencia
umbral_longitud <- 1000

# Elegir una línea de referencia
linea_ref <- lineas %>% filter(longitud > umbral_longitud) %>% filter(date == "2013-08-03 15:04:35 AST")

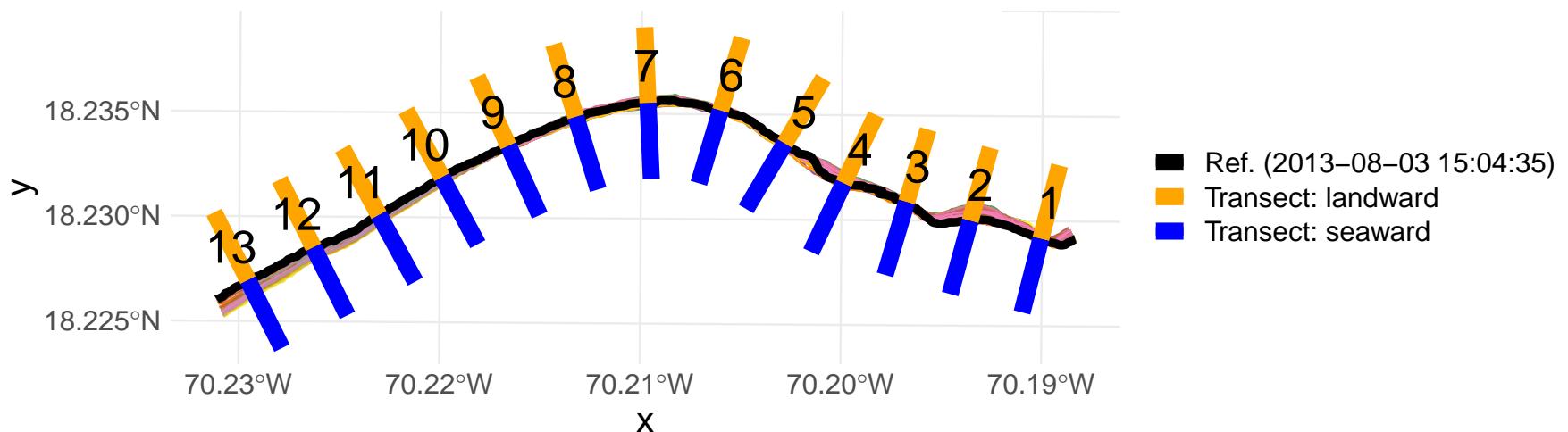
# Crear transectos
transectos <- create_transect(x = linea_ref, 400, reverse = T) %>% rename(transect=coastr_id)

# Mapa
mapa_lineas + geom_sf(data = transectos, color = 'blue')
```



- Clasificar las distintas partes del transecto en tierra o mar

```
transectos_clasif <- transclas(tr = transectos, rl = linea_ref)
cols <- colorRampPalette(brewer.pal(9, 'Set1'))(nrow(lineas))
ggplot() +
  geom_sf(data = lineas %>% mutate(date = factor(date)), color = cols) +
  geom_sf(
    data = linea_ref %>% mutate(linetype = paste0('Ref. ', date, ' ')),
    aes(color=linetype), linewidth = 2, show.legend = 'line') +
  geom_sf(
    data = transectos_clasif %>% mutate(sealand=paste0('Transect: ', sealand)),
    aes(color = sealand), show.legend = 'line', linewidth = 4) +
  scale_color_manual(values = c('black', 'orange', 'blue')) +
  geom_sf_text(
    data = transectos_clasif %>% filter(sealand=='landward') %>%
      st_centroid, aes(label = transect), size = 8) +
  theme_minimal() +
  theme(legend.title = element_blank(), text = element_text(size = 18))
```

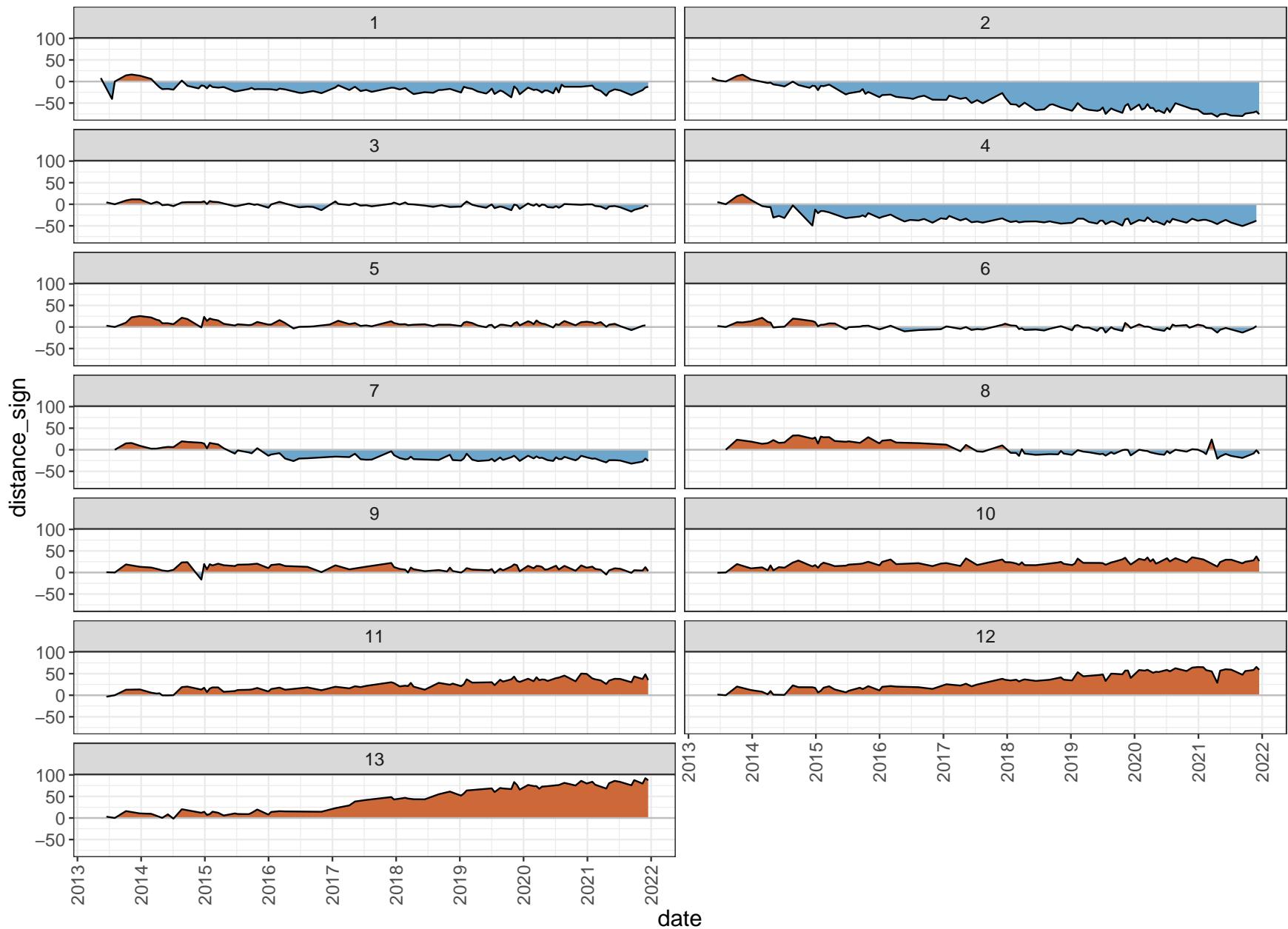


- Calcular distancias de cada línea de costa respecto de la línea de referencia

```
distl <- pointdist(sh = lineas, re = linea_ref, tr = transectos_clasif, rtr = transectos)
```

- Generar las series temporales de distancia de la línea de costa respecto a la de referencia

```
interdist <- map(distl, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances <- plyr::ldply(distl) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
#  scale_y_continuous(limits = c(-30, 30)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)
```



- Revisión de seguridad: determinar si hay transectos que corten dos veces una misma línea de costa

```

test <- sapply(unique(distances$transect), function(x) {
  conteo_cortes <- table(distances[distances$transect==x, 'date', drop=T])
  mas_de_1 <- length(which(conteo_cortes>1))>0
  ifelse(mas_de_1,
    paste('El transecto', x, 'corta', conteo_cortes[which(conteo_cortes>1)],
      'veces la línea de costa de fecha', names(conteo_cortes[which(conteo_cortes>1)])),
    paste('El transecto', x, 'pasa la prueba'))
})
test

## [1] "El transecto 1 pasa la prueba" "El transecto 2 pasa la prueba"
## [3] "El transecto 3 pasa la prueba" "El transecto 4 pasa la prueba"
## [5] "El transecto 5 pasa la prueba" "El transecto 6 pasa la prueba"
## [7] "El transecto 7 pasa la prueba" "El transecto 8 pasa la prueba"
## [9] "El transecto 9 pasa la prueba" "El transecto 10 pasa la prueba"
## [11] "El transecto 11 pasa la prueba" "El transecto 12 pasa la prueba"
## [13] "El transecto 13 pasa la prueba"

```

- Tasas de cambio (-erosión, +acreción)

```

tasas_de_cambio <- lapply(
  unique(distances$transect),
  function (x) calcular_tasa_de_cambio(transecto = x)) %>%
  setNames(paste('Transecto', unique(distances$transect)))
invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(titulo = 'Tasa de cambio anual promediada pos-2013 (L8) para transectos del Tramo Nizao Este'))

```

Table 11: Tasa de cambio anual promediada pos-2013 (L8) para transectos del Tramo Nizao Este

Transecto	Periodo	Tasa de cambio anual, en metros (-erosión,+acreción)
Transecto 1	2013-2021 (8 años)	-3.19
Transecto 2	2013-2021 (8 años)	-10.19
Transecto 3	2013-2021 (8 años)	-1.88
Transecto 4	2013-2021 (8 años)	-5.76
Transecto 5	2013-2021 (8 años)	-1.95
Transecto 6	2013-2021 (8 años)	-2.23
Transecto 7	2013-2021 (8 años)	-4.84
Transecto 8	2013-2021 (8 años)	-4.11
Transecto 9	2013-2021 (8 años)	-0.79
Transecto 10	2013-2021 (8 años)	1.93
Transecto 11	2013-2021 (8 años)	3.69
Transecto 12	2013-2021 (8 años)	5.78
Transecto 13	2013-2021 (8 años)	9.36

```
invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(
    titulo = 'Tasa de cambio por año pos-2013 (en metros; -erosión, +acreción) para transectos del Tramo Nizao Este'))
```

Table 12: Tasa de cambio por año pos-2013 (en metros; -erosión, +acreción) para transectos del Tramo Nizao Este

Año	Transecto												
	1	2	3	4	5	6	7	8	9	10	11	12	13
2014	-13.55	-15.18	-2.14	-24.96	-2.65	-0.41	2.79	8.89	2.46	7.73	6.67	5.66	3.50
2015	-6.38	-18.12	-4.23	-5.64	-5.07	-10.93	-21.24	-6.06	3.93	3.36	0.25	2.47	1.72
2016	-0.38	-8.91	-2.14	-6.48	-1.05	-2.54	-7.20	-10.35	-6.63	-1.79	3.02	3.65	9.64
2017	-0.84	-7.73	2.60	-3.21	0.62	3.18	0.15	-12.11	1.77	2.31	6.36	12.36	21.74
2018	-0.45	-14.38	-2.82	-1.71	-1.36	-1.49	-3.97	-5.41	-6.73	-0.54	3.55	7.46	14.47
2019	-1.61	-1.82	-1.37	0.37	1.60	0.93	1.36	2.35	5.49	5.02	8.23	11.20	12.46
2020	5.89	-4.71	1.62	1.23	1.24	0.83	-1.64	2.20	-1.54	0.76	4.08	5.82	7.10
2021	-8.22	-10.64	-6.54	-5.66	-8.96	-7.43	-8.96	-12.40	-5.04	-1.43	-2.65	-2.35	4.30

Exportar las tablas anteriores a CSV.

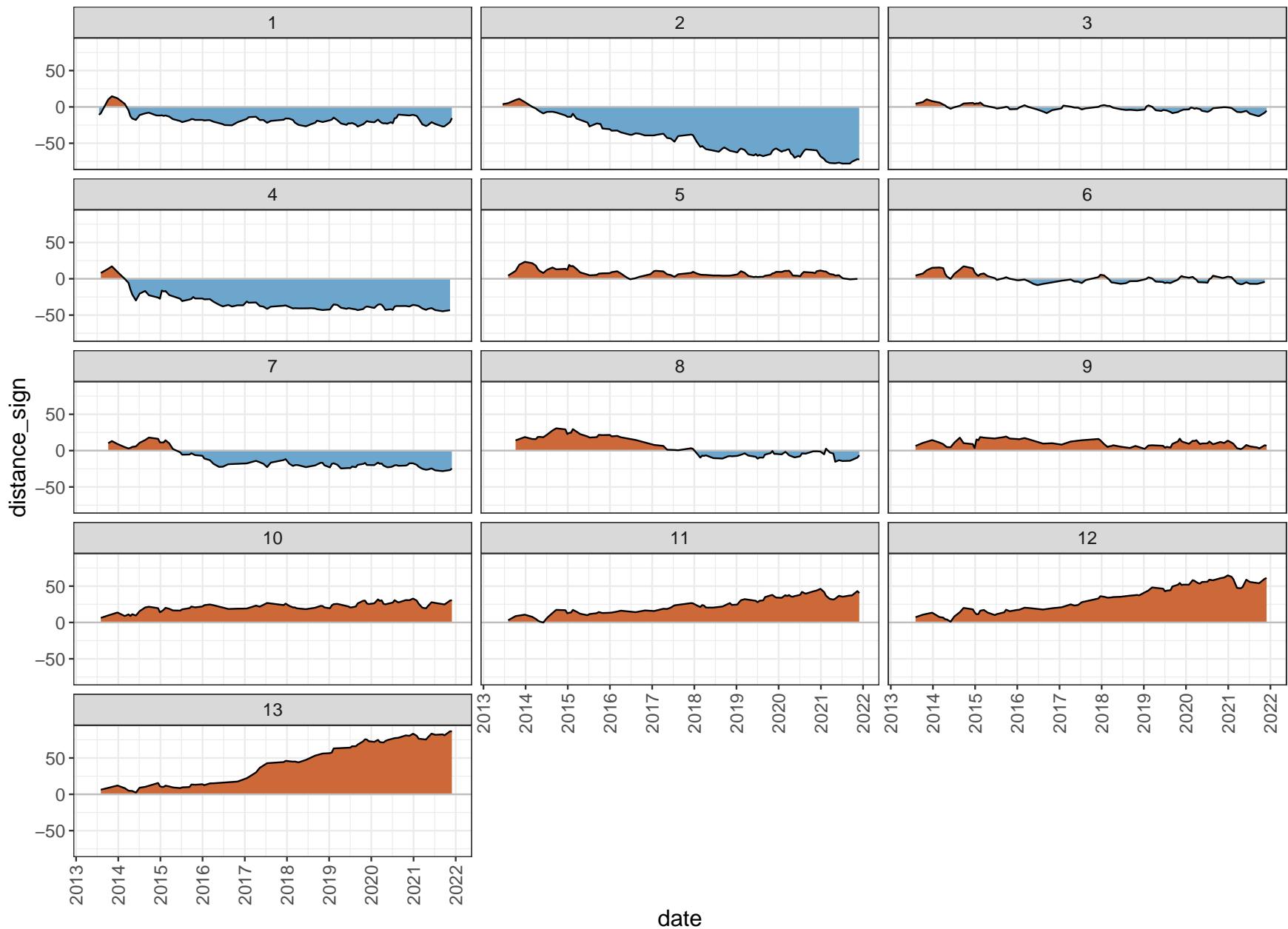
```
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(csv = T) %>%
  write_csv('tasas_de_cambio_anual_promediada_pos_2013_por_transecto_tramo_nizao_este.csv')
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(csv = T) %>%
  write_csv('tasas_de_cambio_anual_pos_2013_por_transecto_tramo_nizao_este.csv')
```

- Suavizado de la serie con media móvil

```
ventana_de_promediado <- 3 #Número de observaciones para obtener la media móvil (ventana de promediado)
distl_med <- sapply(unique(distances$transect),
  function(x){
    df <- distances[distances$transect==x, ]
    df <- df[order(df$date), ]
    x <- zoo(df$distance_sign, df$date)
    mm <- as.numeric(rollmean(x, ventana_de_promediado, fill = NA))
    df$distance_sign <- mm
    df <- df #%>% slice(1:(n()-1))
    return(df)
  }, simplify=F)
interdist_med <- map(distl_med, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances_med <- plyr::ldply(distl_med) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
```

- Representación de la serie suavizada

```
distances_med %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist_med, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist_med, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 3)
```



Tramo Nizao oeste

- Cargar líneas de costa

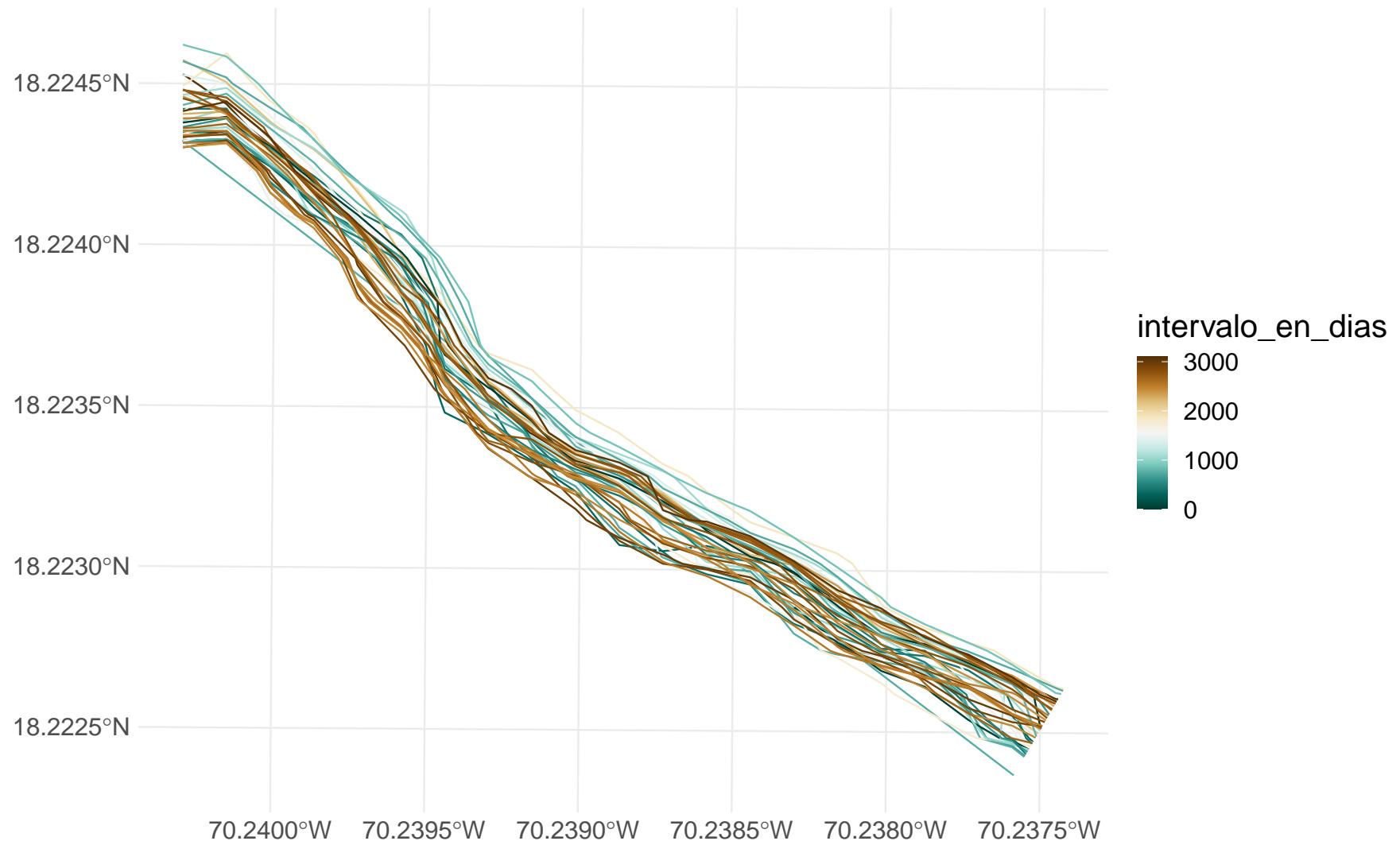
```
lineas <- st_read('lineas-de-costa/PalenqueNizaoPC_L8_output_lines.gpkg') %>%
  filter(grepl('nizao oeste', tramo, ignore.case = T)) %>%
  st_cast('LINESTRING')

## Reading layer `PalenqueNizaoPC_L8_output_lines` from data source
##   `/home/jose/Documentos/git/tesis-ana-carolain/lineas-de-costa/PalenqueNizaoPC_L8_output_lines.gpkg'
##   using driver `GPKG'
## Simple feature collection with 483 features and 5 fields
## Geometry type: LINESTRING
## Dimension:      XY
## Bounding box:  xmin: 368857.5 ymin: 2015230 xmax: 377399 ymax: 2016803
## Projected CRS: WGS 84 / UTM zone 19N

# QUITANDO COSTAS QUE SE CORTAN DOS VECES EN LOS TRANSECTOS,
# LO CUAL SE COMPROBÓ EN LOS TESTS (MÁS ADELANTE)
lineas <- lineas %>%
  mutate(foo = as.character(as.Date(date, '%Y%m%d'))) %>%
  filter(!foo %in% c("2015-10-28", "2021-11-29")) %>%
  select(-foo)
st_geometry(lineas) <- "geometry"
lineas$longitud <- units::drop_units(st_length(lineas))
lineas <- lineas[lineas$longitud > 0, ]
```

- Representar las líneas de costa

```
lineas$intervalo_en_dias <- round(as.numeric(interval(lineas$date, max(lineas$date)), 'days'), 0)
escala_color <- 'BrBG'
mapa_lineas <- lineas %>% ggplot + aes(color=intervalo_en_dias) + geom_sf() +
  theme_minimal() +
  theme(text = element_text(size = 18)) +
  scale_color_gradientn(colors = rev(RColorBrewer::brewer.pal(11, escala_color)))
mapa_lineas
```



- Crear transectos respecto de línea de costa de referencia y representarlos

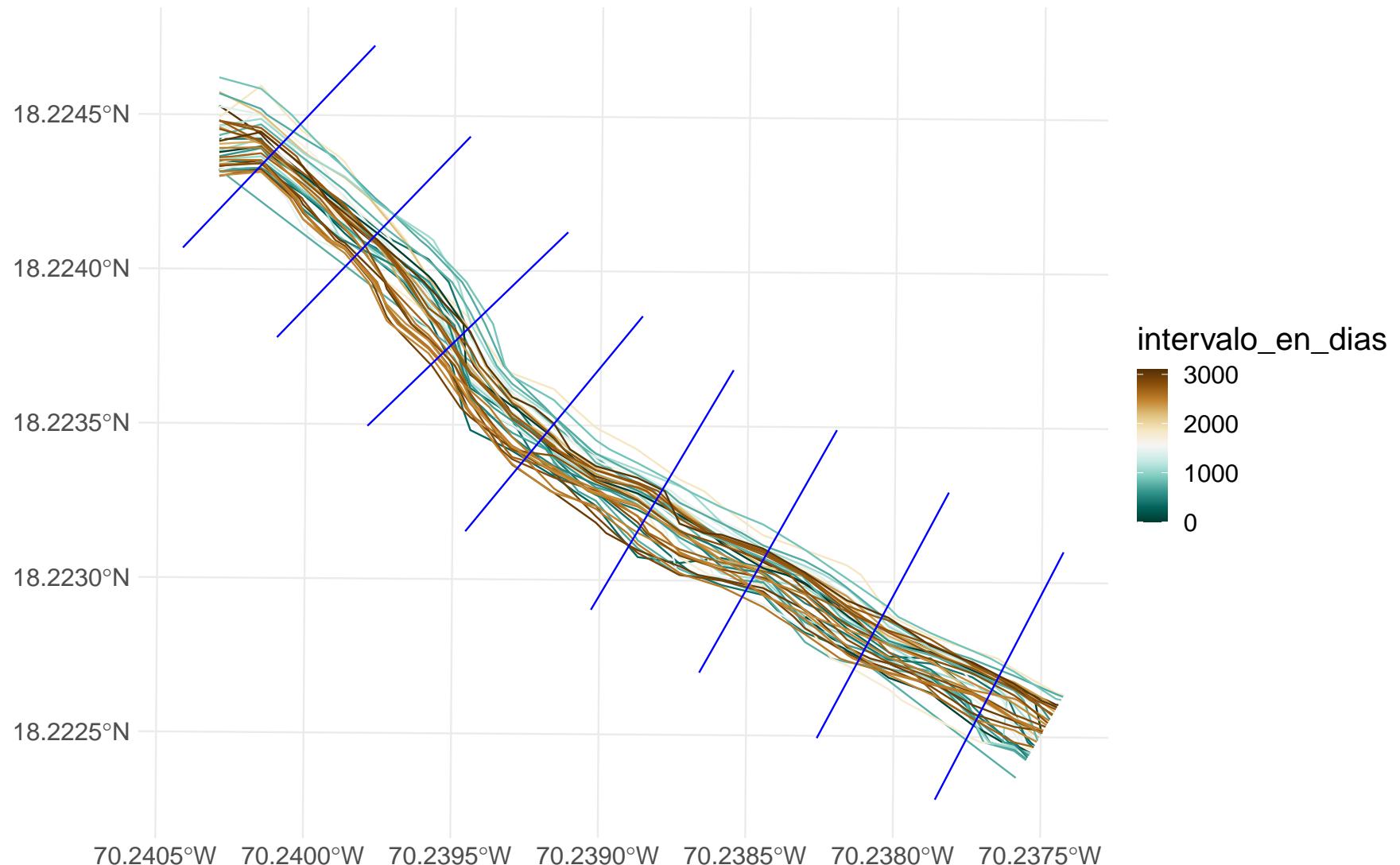
```
# Umbral de longitud para líneas que podrían usarse como referencia
umbral_longitud <- 200
```

```
# Elegir una línea de referencia
```

```
linea_ref <- lineas %>% filter(longitud > umbral_longitud) %>% filter(date == "2013-08-03 15:04:35 AST")

# Crear transectos
transectos <- create_transect(x = linea_ref, 50, reverse = T) %>% rename(transect=coastr_id)

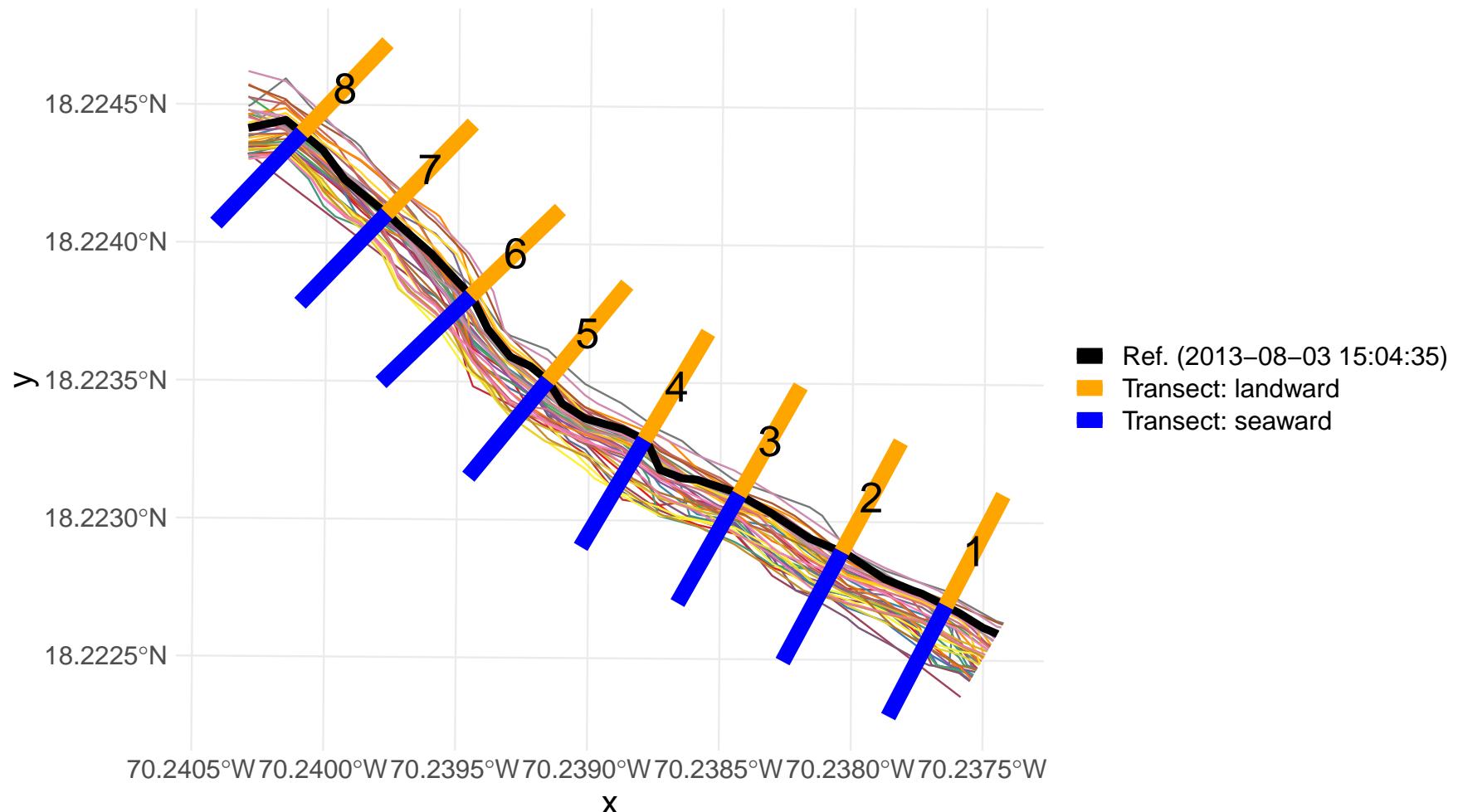
# Mapa
mapa_lineas + geom_sf(data = transectos, color = 'blue')
```



- Clasificar las distintas partes del transecto en tierra o mar

```
transectos_clasif <- transclas(tr = transectos, rl = linea_ref)
cols <- colorRampPalette(brewer.pal(9, 'Set1'))(nrow(lineas))
ggplot() +
```

```
geom_sf(data = lineas %>% mutate(date = factor(date)), color = cols) +
geom_sf(
  data = linea_ref %>% mutate(linetype = paste0('Ref. (', date, ')')),
  aes(color=linetype), linewidth = 2, show.legend = 'line') +
geom_sf(
  data = transectos_clasif %>% mutate(sealand=paste0('Transect: ', sealand)),
  aes(color = sealand), show.legend = 'line', linewidth = 4) +
scale_color_manual(values = c('black', 'orange', 'blue')) +
geom_sf_text(
  data = transectos_clasif %>% filter(sealand=='landward') %>%
    st_centroid, aes(label = transect), size = 8) +
theme_minimal() +
theme(legend.title = element_blank(), text = element_text(size = 18))
```



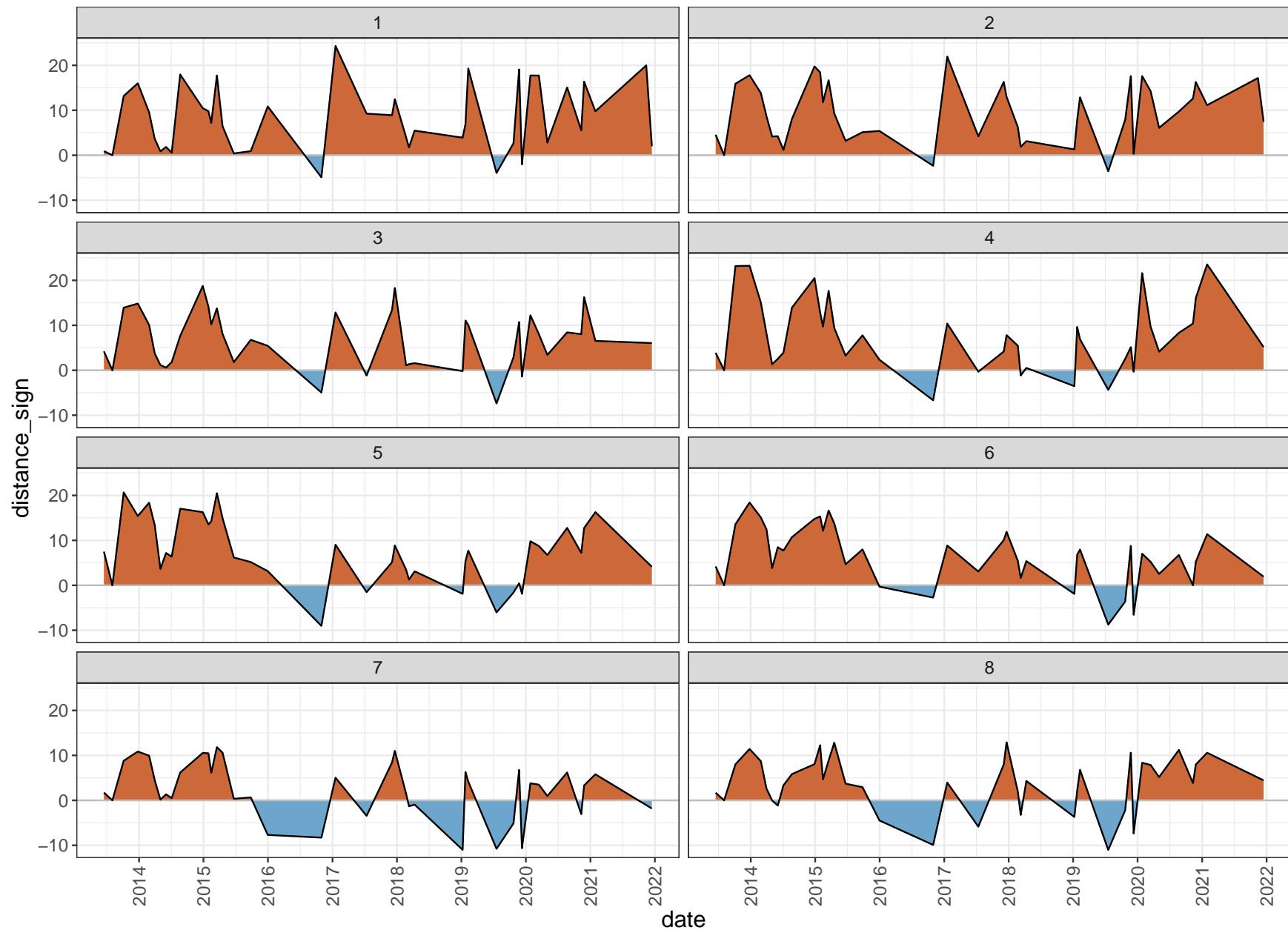
- Calcular distancias de cada línea de costa respecto de la línea de referencia

```
distl <- pointdist(sh = lineas, re = linea_ref, tr = transectos_clasif, rtr = transectos)
```

- Generar las series temporales de distancia de la línea de costa respecto a la de referencia

```
interdist <- map(distl, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances <- plyr::ldply(distl) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances %>%
```

```
ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
#  scale_y_continuous(limits = c(-30, 30)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 2)
```



- Revisión de seguridad: determinar si hay transectos que corten dos veces una misma línea de costa

```

test <- sapply(unique(distances$transect), function(x) {
  conteo_cortes <- table(distances[distances$transect==x, 'date', drop=T])
  mas_de_1 <- length(which(conteo_cortes>1))>0
  ifelse(mas_de_1,
    paste('El transecto', x, 'corta', conteo_cortes[which(conteo_cortes>1)],
      'veces la línea de costa de fecha', names(conteo_cortes[which(conteo_cortes>1)])),
    paste('El transecto', x, 'pasa la prueba'))
})
test

## [1] "El transecto 1 pasa la prueba" "El transecto 2 pasa la prueba"
## [3] "El transecto 3 pasa la prueba" "El transecto 4 pasa la prueba"
## [5] "El transecto 5 pasa la prueba" "El transecto 6 pasa la prueba"
## [7] "El transecto 7 pasa la prueba" "El transecto 8 pasa la prueba"

```

- Tasas de cambio (-erosión, +acreción)

```

tasas_de_cambio <- lapply(
  unique(distances$transect),
  function (x) calcular_tasa_de_cambio(transecto = x)) %>%
  setNames(paste('Transecto', unique(distances$transect)))
invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(titulo = 'Tasa de cambio anual promediada pos-2013 (L8) para transectos del Tramo Nizao Oeste'))

```

Table 13: Tasa de cambio anual promediada pos-2013 (L8) para transectos del Tramo Nizao Oeste

Transecto	Periodo	Tasa de cambio anual, en metros (-erosión,+acreción)
Transecto 1	2013-2021 (8 años)	1.29
Transecto 2	2013-2021 (8 años)	0.78
Transecto 3	2013-2021 (8 años)	-0.26
Transecto 4	2013-2021 (8 años)	-0.46
Transecto 5	2013-2021 (8 años)	-0.68
Transecto 6	2013-2021 (8 años)	-0.77
Transecto 7	2013-2021 (8 años)	-0.70
Transecto 8	2013-2021 (8 años)	0.08

```

invisible(tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(
    titulo = 'Tasa de cambio por año pos-2013 (en metros; -erosión, +acreción) para transectos del Tramo Nizao Oeste'))

```

Table 14: Tasa de cambio por año pos-2013 (en metros; -erosión, +acreción) para transectos del Tramo Nizao Oeste

Año	Transecto 1	Transecto 2	Transecto 3	Transecto 4	Transecto 5	Transecto 6	Transecto 7	Transecto 8
2014	1.54	1.17	2.11	-0.42	1.54	0.92	1.24	1.40
2015	-5.15	-7.91	-6.66	-10.56	-12.19	-10.22	-12.33	-9.81
2016	4.61	4.42	-0.85	-1.05	-2.34	0.89	2.47	-1.07
2017	-2.15	-0.48	2.66	1.04	3.19	3.42	4.47	6.63
2018	-0.88	-4.06	-3.82	-2.34	-2.19	-5.29	-6.88	-3.75
2019	0.95	4.44	2.12	5.67	2.17	-0.37	3.23	2.48
2020	4.17	3.81	4.07	9.54	9.24	6.23	4.89	6.83
2021	7.23	4.87	-1.73	-5.55	-4.89	-1.75	-2.67	-2.05

Exportar las tablas anteriores a CSV.

```

tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio(csv = T) %>%
  write_csv('tasas_de_cambio_anual_promediada_pos_2013_por_transecto_tramo_nizao_oeste.csv')
tasas_de_cambio %>%
  generar_tabla_de_tasas_de_cambio_por_ano(csv = T) %>%
  write_csv('tasas_de_cambio_anual_pos_2013_por_transecto_tramo_nizao_oeste.csv')

```

- Suavizado de la serie con media móvil

```

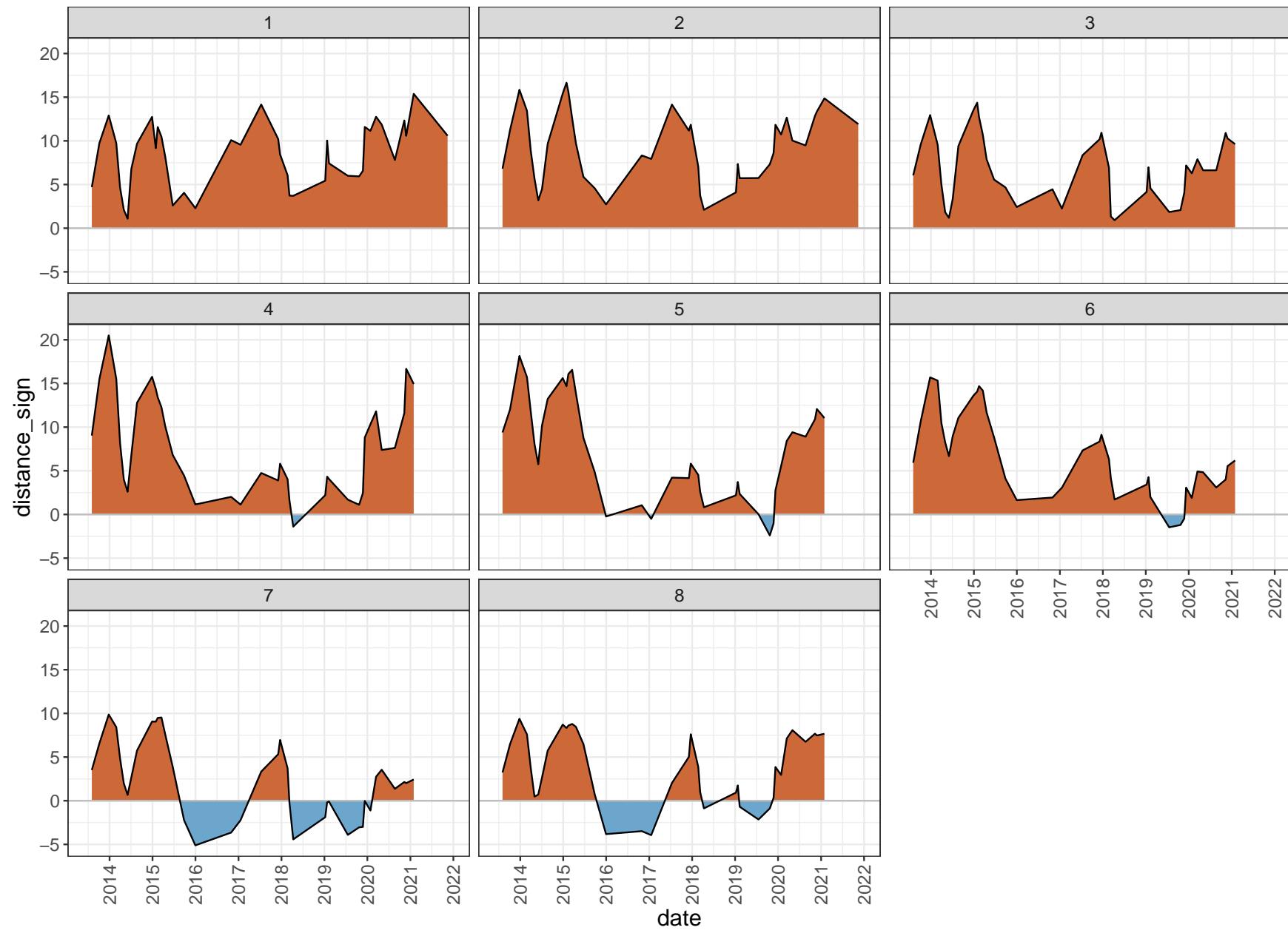
ventana_de_promediado <- 3 #Número de observaciones para obtener la media móvil (ventana de promediado)
dist1_med <- sapply(unique(distances$transect),
  function(x){
    df <- distances[distances$transect==x, ]
    df <- df[order(df$date), ]
    x <- zoo(df$distance_sign, df$date)
    mm <- as.numeric(rollmean(x, ventana_de_promediado, fill = NA))
    df$distance_sign <- mm
    df <- df #%>% slice(1:(n()-1))
    return(df)
  }, simplify=F)

```

```
interdist_med <- map(distl_med, interpolate) %>% plyr::ldply() %>% mutate(date = as.Date(date, "%Y-%m-%d"))
distances_med <- plyr::ldply(distl_med) %>% mutate(date = as.Date(date, "%Y-%m-%d"))
```

- Representación de la serie suavizada

```
distances_med %>%
  ggplot() + theme_bw() + aes(x = date, y = distance_sign) +
  geom_ribbon(data = interdist_med, aes(ymax = pmax(distance_sign, 0), ymin = 0), fill = "sienna3") +
  geom_ribbon(data = interdist_med, aes(ymin = pmin(distance_sign, 0), ymax = 0), fill = "skyblue3") +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_line(colour='black', lwd = 0.5) +
  scale_x_date(date_labels = "%Y", date_breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5), text = element_text(size = 14)) +
  facet_wrap(~transect, ncol = 3)
```



Pos-2015 (Sentinel 2)

COMO QUE NO HACE FALTA TALI, MEJOL DEJA ESO.

Referencias

GBIF.org (2023). *What is GBIF?* Retrieved from <https://www.gbif.org/what-is-gbif>

H3 (2022). *Introduction / H3.* Retrieved from <https://h3geo.org/docs>

José Ramón Martínez-Batlle (2022). Estadística zonal multipropósito sobre información geoespacial de República Dominicana, usando Google Earth Engine, Python y R. Versión “Let there be environmental variables (v0.0.0.9000)”. Zenodo <https://doi.org/10.5281/zenodo.7367256>