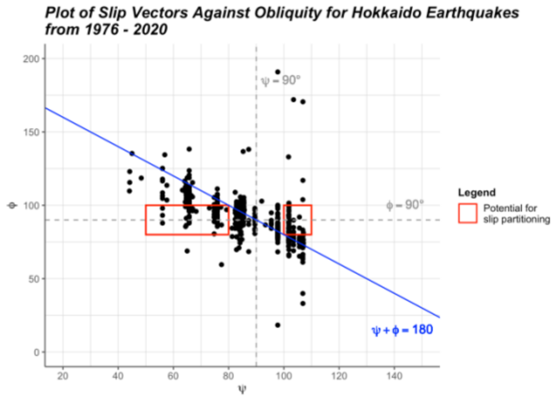
Hokkaido Slip Partitioning Earthquake Analysis

```
#data parsing for all earthquakes
library(tidyr)
lines <- readLines("all_earthquakes_data.ndk")
n_events <- length(lines) / 5
all_earthquakes <- data.frame(matrix(ncol = 8, nrow = n_events))
colnames(all_earthquakes) <- c("event_id", "date",
                        "lat", "long", "depth", "mw", "location",
                        "moment_tensor")
for (i in 1:n_events) {
  base <- (i - 1) * 5
  line1 <- lines[base + 1]
  line2 <- lines[base + 2]
  line3 <- lines[base + 3]
  line4 <- lines[base + 4]
  line5 <- lines[base + 5]
  all_earthquakes$event_id[i] <- substr(line2, 1, 16)
  all_earthquakes$date[i] <- substr(line1, 6, 15)
  all_earthquakes$lat[i] <- substr(line1, 28, 33)
  all_earthquakes$long[i] <- substr(line1, 35, 41)
  all_earthquakes$depth[i] <- substr(line1, 43, 47)
  all_earthquakes$mw[i] <- substr(line1, 49, 55)
  all_earthquakes$location[i] <- substr(line1, 57, 80)
  all_earthquakes$moment_tensor[i] <- substr(line5, 58, 80)
}

all_earthquakes_separated <- cbind(all_earthquakes[,1:7], split_columns <- do.call(rbind, strsplit(trimws(
all_earthquakes$moment_tensor), "\\s+")))
colnames(all_earthquakes_separated)[8:13] <- c("strike_1", "dip_1", "rake_1", "strike_2", "dip_2", "rake_2
")
all_earthquakes_separated <- separate(all_earthquakes_separated, mw, into = c("mb", "MS"), sep = " ", conv
ert = TRUE)

#filter for boundaries from Slab2 for Hokkaido region
earthquakes_hk <- all_earthquakes_separated[(all_earthquakes_separated$long >= 135 & all_earthquakes_separ
ated$long <= 155) & (all_earthquakes_separated$lat >= 38 & all_earthquakes_separated$lat <= 48),]

#filter for rake between 70-110
earthquakes_hk <- earthquakes_hk[(earthquakes_hk$rake_1 > 70 & earthquakes_hk$rake_1 < 110) & (earthquakes
_hk$rake_2 > 70 & earthquakes_hk$rake_2 < 110), ]

#filter for smaller dip between the 2 sets of values
earthquakes_hk_clean <- data.frame(matrix(ncol = 11, nrow = nrow(earthquakes_hk)))
colnames(earthquakes_hk_clean) <- c(colnames(earthquakes_hk)[1:8], "strike", "dip", "rake")

for(i in 1:nrow(earthquakes_hk)) {
  if (earthquakes_hk[i,"dip_1"] < earthquakes_hk[i,"dip_2"]) {
    earthquakes_hk_clean[i,] <- earthquakes_hk[i, c(1:11)]
  } else {
    earthquakes_hk_clean[i,] <- earthquakes_hk[i, c(1:8,12:14)]
  }
}

#filter for depth <= 50
earthquakes_hk_clean <- earthquakes_hk_clean[earthquakes_hk_clean$depth <= 50,]

#---------------------------------------------------------------------------------------------#

#get slip vectors
slip_vectors_hk <- data.frame()

for (i in 1:nrow(earthquakes_hk_clean)) {
  strike <- earthquakes_hk_clean[i,"strike"]*pi/180
  dip <- earthquakes_hk_clean[i,"dip"]*pi/180
  rake <- earthquakes_hk_clean[i,"rake"]*pi/180

  slip_x <- cos(rake)*sin(strike) - sin(rake)*cos(strike)*cos(dip)
```

```r
  slip_y <- cos(rake)*cos(strike) + sin(rake)*sin(strike)*cos(dip)
  slip_up <- sin(rake)*sin(dip)

  slip_angle <- atan(slip_x/slip_y)*180/pi

  if (slip_y < 0 & slip_x > 0) {
    slip_angle <- slip_angle + 180
  }

  if (slip_y <0 & slip_x <0) {
    slip_angle <- slip_angle - 180
  }

  all_data_earthquake <- cbind(earthquakes_hk_clean[i,], slip_x, slip_y, slip_up, slip_angle)

  slip_vectors_hk <- rbind(slip_vectors_hk, all_data_earthquake)
}

#-----------------------------------------------------------------------------------------------#

#get strikes at depth=0
library(collapse)
library(sf)
library(geosphere)

#getting the closest long and lat at depth = 0
depths_hk <- read.table("kur_slab2_dep_02.24.18.xyz", sep = ",")
depths_hk <- na.omit(depths_hk)
colnames(depths_hk) <- c("long", "lat", "depth")
depth_0 <- depths_hk[depths_hk$depth >= -10,]

extracted_hk_raw <- data.frame()
slip_and_strike_hk <- data.frame()
slip_and_strike_hk <- slip_vectors_hk
slip_and_strike_hk$long_at_depth_0 <- NA
slip_and_strike_hk$lat_at_depth_0 <- NA
slip_and_strike_hk$long <- ifelse(slip_and_strike_hk$long < 0, slip_and_strike_hk$long + 360, slip_and_str
ike_hk$long)

depth_coords <- st_as_sf(depth_0, coords = c("long", "lat"), crs = 4326)

for (i in 1:nrow(slip_and_strike_hk)) {
  long <- slip_and_strike_hk[i,"long"]
  long <- round(long/0.05)*0.05
  long <- as.numeric(format(long, nsmall = 2))
  if (long < min(depth_0$long) - 0.5 | long > max(depth_0$long) + 0.5) {
    slip_and_strike_hk[i,"long_at_depth_0"] <- NA
    slip_and_strike_hk[i,"lat_at_depth_0"] <- NA
  } else {
    eq_coords <- st_as_sf(slip_and_strike_hk[i,], coords = c("long", "lat"), crs = 4326)
    nearest_points <- st_nearest_feature(eq_coords, depth_coords)
    nearest <- depth_coords[nearest_points, ]
    nearest <- as.data.frame(st_coordinates(nearest))
    colnames(nearest) <- c("long", "lat")
    slip_and_strike_hk$long_at_depth_0[i] <- nearest$long
    slip_and_strike_hk$lat_at_depth_0[i] <- nearest$lat
  }
}

#getting the strikes at the longitude and latitudes at depth = 0
strike_hk <- read.csv("kur_slab2_str_02.24.18.xyz", sep = ",")
strike_hk <- na.omit(strike_hk)
colnames(strike_hk) <- c("long_at_depth_0", "lat_at_depth_0", "strike_at_depth_0")
slip_and_strike_hk$long_at_depth_0 <- round(slip_and_strike_hk$long_at_depth_0, 2)
strike_hk$long_at_depth_0 <- round(strike_hk$long_at_depth_0, 2)
slip_and_strike_hk$lat_at_depth_0 <- round(slip_and_strike_hk$lat_at_depth_0, 2)
strike_hk$lat_at_depth_0 <- round(strike_hk$lat_at_depth_0, 2)

slip_and_strike_hk <- left_join(slip_and_strike_hk, strike_hk, by = c("long_at_depth_0", "lat_at_depth_0")
```

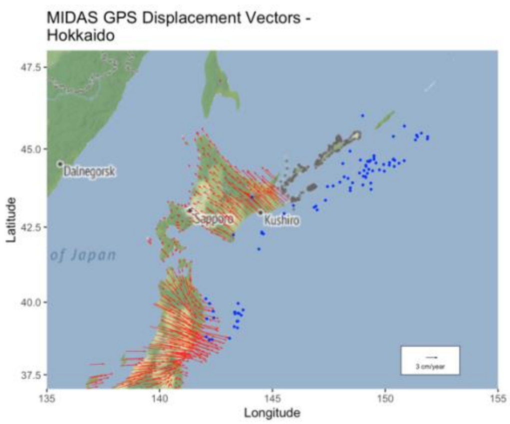
```
)


#editing angles so that they are between -180 and 180
for(i in 1:nrow(slip_and_strike_hk)) {
  if (!is.na(slip_and_strike_hk$strike_at_depth_0[i])) {
    if(slip_and_strike_hk$strike_at_depth_0[i] > 180) {
      slip_and_strike_hk$strike_at_depth_0[i] <- slip_and_strike_hk$strike_at_depth_0[i] - 360
    }
  }
}

#loading plate convergence data
plate_movement_hk <- read.csv("Plate Data - hk plate movement.csv")
colnames(plate_movement_hk) <- c("no", "lat", "long", "vel", "azimuth", "n_vel", "e_vel", "reference_plate
")

for(i in 1:nrow(plate_movement_hk)){
  if (plate_movement_hk$long[i] < 0) {
    plate_movement_hk$long[i] <- plate_movement_hk$long[i] + 360
  }
}

#assigning plate boundary data based on closest location
slip_and_strike_hk$plate_movement <- NA
for(i in 1:nrow(slip_and_strike_hk)) {
  if(is.na(slip_and_strike_hk$strike_at_depth_0[i])) {
    slip_and_strike_hk$plate_movement[i] <- NA
  } else {
    distances <- data.frame()
    for (j in 1:nrow(plate_movement_hk)) {
      distance <- distGeo(c(plate_movement_hk$long[j], plate_movement_hk$lat[j]),
                          c(slip_and_strike_hk$long[i], slip_and_strike_hk$lat[i]))
      distances <- rbind(distances, data.frame(long = plate_movement_hk$long[j],
                                               lat = plate_movement_hk$lat[j],
                                               distance = distance))
    }
    closest_point <- distances[which.min(distances$distance), ]
    slip_and_strike_hk$plate_movement[i] <- plate_movement_hk$azimuth[plate_movement_hk$long == closest_po
int$long & plate_movement_hk$lat == closest_point$lat]
  }
}

head(slip_and_strike_hk)
##            event_id       date   lat   long depth  mb  MS
## 1 B020377A         1977/02/03 45.37 150.44    33 5.5 0.0
## 2 B021877A         1977/02/18 41.41 142.04     5 5.5 5.6
## 3 B101677A         1977/10/16 47.05 153.91    33 5.6 5.6
## 4 B102077A         1977/10/20 47.15 154.09    44 5.4 5.4
## 5 B110877A         1977/11/08 47.39 154.31    33 5.5 4.6
## 6 B112777C         1977/11/27 46.42 153.28    33 5.5 5.7
##                   location strike dip rake    slip_x     slip_y   slip_up
## 1 KURIL ISLANDS               207  32   87 0.7308208 -0.4311098 0.5291930
## 2 HOKKAIDO, JAPAN REGION      184  28   73 0.8219153 -0.3505596 0.4489579
## 3 KURIL ISLANDS               217  31  101 0.7868186 -0.3539916 0.5055754
## 4 KURIL ISLANDS               211  34   86 0.6729656 -0.4857388 0.5578307
## 5 KURIL ISLANDS               207  31   82 0.6931257 -0.5093628 0.5100258
## 6 KURIL ISLANDS               227  39  100 0.6489584 -0.4413060 0.6197596
##   slip_angle long_at_depth_0 lat_at_depth_0 strike_at_depth_0 plate_movement
## 1   120.5363          152.95          45.50         -137.0058         298.90
## 2   113.0990          144.55          40.75         -163.4461         294.34
## 3   114.2231          154.40          46.80         -144.5335         300.98
## 4   125.8213          154.55          46.95         -144.5042         300.98
## 5   126.3113          154.80          47.15         -144.1680         300.98
## 6   124.2166          153.80          46.20         -143.9211         300.19

#plotting graph
library(ggplot2)
```

```r
#changing all the strikes to be between 0 and 360
slip_and_strike_hk$strike_at_depth_0 <- ifelse(slip_and_strike_hk$strike_at_depth_0 < 0, slip_and_strike_h
k$strike_at_depth_0 + 360, slip_and_strike_hk$strike_at_depth_0)

slip_and_strike_hk$bounds_slip <- NA
slip_and_strike_hk$bounds_convg <- NA

#generating all the plate conv and strike points
slip_and_strike_hk$bounds_slip <- slip_and_strike_hk$strike_at_depth_0 - slip_and_strike_hk$slip_angle
slip_and_strike_hk$bounds_convg <- abs(slip_and_strike_hk$strike_at_depth_0 - slip_and_strike_hk$plate_mov
ement)

#putting all the points into a separate data frame
points <- data.frame(bounds_slip = numeric(0), bounds_convg = numeric(0))
for(i in 1:nrow(slip_and_strike_hk)) {
  if(!is.na(slip_and_strike_hk$strike_at_depth_0[i])){
    bounds_slip <- slip_and_strike_hk$bounds_slip[i]
    bounds_convg <- slip_and_strike_hk$bounds_convg[i]
    points_current <- cbind(bounds_slip, bounds_convg)
    points <- rbind(points, points_current)
  }
}

#annotated graph for potential slip partitioning
ggplot(points, aes(x = abs(bounds_convg), y = bounds_slip)) +
  geom_point() +
  geom_rect(aes(xmin = 50, xmax = 80, ymin = 80, ymax = 100, color = "red"),
            fill = NA, linetype = "solid") +
  geom_rect(aes(xmin = 100, xmax = 110, ymin = 80, ymax = 100, color = "red"),
            fill = NA, linetype = "solid") +
  labs(title = "Plot of Slip Vectors Against Obliquity for Hokkaido Earthquakes
from 1976 - 2020", x = expression(psi), y = expression(phi), size = 7) +
  scale_x_continuous(limits = c(20, 150), breaks = seq(20, 140, 20)) +
  scale_y_continuous(limits = c(0, 200), breaks = seq(0, 200, 50)) +
  geom_vline(xintercept = 90, color = "#9C9C9C",
             linetype = "dashed", linewidth = 0.5) +
  geom_text(x = 99, y = 185, label = expression(psi == 90 * degree), color = "#9C9C9C") +
  geom_hline(yintercept = 90, color = "#9C9C9C",
             linetype = "dashed", linewidth = 0.5) +
  geom_text(x = 144, y = 100, label = expression(phi == 90 * degree), color = "#9C9C9C") +
  geom_abline(intercept = 180, slope = -1, color = "blue", linetype = "solid", linewidth = 0.5) +
  geom_text(x = 143, y = 15, label = expression(psi + phi == 180), color = "blue", size = 4) +
  scale_color_manual(name = "Legend",
                     values = c("red" = "red"),
                     labels = c("Potential for
slip partitioning")) +
  theme(plot.title = element_text(size = 14, face = "bold.italic"),
        panel.background = element_blank(),
        axis.line = element_line(colour = "#585858"),
        axis.ticks = element_line(colour = "#585858"),
        axis.text = element_text(colour = "#585858"),
        panel.grid.major = element_line(colour = "lightgrey", size = 0.2),
        panel.grid.minor = element_line(colour = "lightgrey", size = 0.2),
        legend.position = "right",
        legend.title = element_text(size = 9.5, face = "bold"),
        legend.text = element_text(size = 9))
```




```r
#plot base map
library(ggplot2)
library(ggmap)
library(dplyr)
library(sf)
map_bounds <- c(left = 135, bottom = 37, right = 155, top = 48)
map_base <- get_stadiamap(bbox = map_bounds, zoom = 5, maptype = "stamen_terrain")

#reading gps velocity data
gps_vel_hk <- read.table("midas.OK.txt", header = FALSE)
colnames(gps_vel_hk) <- c("sta", "label", "t_1", "t_m", "delt", "m", "mgood", "n", "ve50", "vn50", "vu50",
"sve", "svn", "svu", "xe50", "xn50", "xu50", "fe", "fn", "fu", "sde", "sdn", "sdu", "nstep", "lat", "long"
, "h")

#convert decimal date to YYYY-MM-DD format
decimal_year_to_date <- function(decimal_year) {
  # Calculate the year, month, and day
  year <- floor(decimal_year)
  fraction_of_year <- decimal_year - year
  days_in_year <- ifelse((year %% 4 == 0 & year %% 100 != 0) | (year %% 400 == 0), 366, 365)
  day_of_year <- round(fraction_of_year * days_in_year)
  # Create the date object
  as.Date(paste0(year, "-01-01")) + day_of_year
}

gps_vel_hk$t_1 <- decimal_year_to_date(gps_vel_hk$t_1)
gps_vel_hk$t_m <- decimal_year_to_date(gps_vel_hk$t_m)

#change ve50, vn50 and vu50 from m/yr to cm/yr
gps_vel_hk$ve50 <- gps_vel_hk$ve50*100
gps_vel_hk$vn50 <- gps_vel_hk$vn50*100
gps_vel_hk$vu50 <- gps_vel_hk$vu50*100

#calculating the magnitudes of the velocity vectors
gps_vel_hk$magnitude <- sqrt(gps_vel_hk$ve50^2 + gps_vel_hk$vn50^2)

#changing lat and long to be within the range of 0 and 360
gps_vel_hk$long <- ifelse(gps_vel_hk$long < 0, gps_vel_hk$long + 360, gps_vel_long)

#plot velocities with earthquakes that have suspected slip partitioning
ggmap(map_base) +
  geom_segment(data = gps_vel_hk,
               aes(x = long, y = lat,
                   xend = long + ve50 * scale_factor,
                   yend = lat + vn50 * scale_factor),
               arrow = arrow(angle = 20, length = unit(0.06, "cm"), ends = "last", type = "open"),
               color = "red", size = 0.2) +
  annotate("rect", xmin = 150.7, xmax = 153.3,
           ymin = 37.5, ymax = 38.5,
           alpha = 1.0, color = "black", fill = "white", size = 0.2) +
  annotate("segment", x = 151.8, y = 38.1,
           xend = 151.7 + legend_length, yend = 38.1,
           arrow = arrow(angle = 20, length = unit(0.06, "cm"), ends = "last", type = "open"),
```

```
           color = "black", size = 0.2) +
  annotate("text", x = 152, y = 37.8,
           label = "3 cm/year", size = 2) +
  labs(title = "MIDAS GPS Displacement Vectors - Hokkaido", x = "Longitude", y = "Latitude") +
  geom_point(data = slip_partitioning_earthquakes_hk, aes(x = long, y = lat), color = "blue", size = 0.5,
alpha = 1) +
  labs(title = "MIDAS GPS Displacement Vectors -
Hokkaido", x = "Longitude", y = "Latitude") +
  theme(plot.title = element_text(size = 14))
```




```
#looking at gps velocities along hokkaido region

#filter for strikes in this region
strike_hk_sp <- strike_hk[(strike_hk$long_at_depth_0 >= 143 & strike_hk$long_at_depth_0 <= 147) & (strike_
hk$lat_at_depth_0 >= 41.5 & strike_hk$lat_at_depth_0 <= 43.5),]

#filter for gps velocities in this region
gps_vel_hk_sp <- gps_vel_hk[(gps_vel_hk$long >= 140 & gps_vel_hk$long <= 147) & (gps_vel_hk$lat >= 41.5 &
gps_vel_hk$lat <= 46), ]

#using arctan to get the angle of the velocity vector (in rad)
#tetha = angle between Ve and Vn
#gps_vel_azimuth = direction of velocity vector relative to geographic north (between 0 and 2pi)

gps_vel_hk_sp$tetha <- NA
gps_vel_hk_sp$gps_vel_azimuth <- NA

for(i in 1:nrow(gps_vel_hk_sp)) {
  if (gps_vel_hk_sp$ve50[i] <= 0 & gps_vel_hk_sp$vn50[i] >= 0) {
    gps_vel_hk_sp$tetha[i] <- atan(abs(gps_vel_hk_sp$ve50[i])/gps_vel_hk_sp$vn50[i])
    gps_vel_hk_sp$gps_vel_azimuth[i] <- 2*pi - gps_vel_hk_sp$tetha[i]
  }

  if (gps_vel_hk_sp$ve50[i] <= 0 & gps_vel_hk_sp$vn50[i] <= 0) {
    gps_vel_hk_sp$tetha[i] <- atan(abs(gps_vel_hk_sp$ve50[i])/abs(gps_vel_hk_sp$vn50[i]))
    gps_vel_hk_sp$gps_vel_azimuth[i] <- pi + gps_vel_hk_sp$tetha[i]
  }
}

#removes points that have positive east component and negative north component
gps_vel_hk_sp <- na.omit(gps_vel_hk_sp)

########################

#assigning closest strike values
#getting the closest long and lat at depth = 0

gps_vel_hk_sp$long_at_depth_0 <- NA
gps_vel_hk_sp$lat_at_depth_0 <- NA

depth_coords <- st_as_sf(depth_0, coords = c("long", "lat"), crs = 4326)
```

```r
for (i in 1:nrow(gps_vel_hk_sp)) {

  long <- gps_vel_hk_sp[i,"long"]
  long <- round(long/0.05)*0.05
  long <- as.numeric(format(long, nsmall = 2))

  eq_coords <- st_as_sf(slip_and_strike_hk[i,], coords = c("long", "lat"), crs = 4326)
  nearest_points <- st_nearest_feature(eq_coords, depth_coords)
  nearest <- depth_coords[nearest_points, ]
  nearest <- as.data.frame(st_coordinates(nearest))
  colnames(nearest) <- c("long", "lat")
  gps_vel_hk_sp$long_at_depth_0[i] <- nearest$long
  gps_vel_hk_sp$lat_at_depth_0[i] <- nearest$lat
}

#getting the strikes at the longitude and latitudes at depth = 0
gps_vel_hk_sp <- left_join(gps_vel_hk_sp, strike_hk, by = c("long_at_depth_0", "lat_at_depth_0"))
#converting to radians
gps_vel_hk_sp$strike_at_depth_0 <- gps_vel_hk_sp$strike_at_depth_0*pi/180
#subtracting pi from strikes so that they fall within 0 and 90
gps_vel_hk_sp$strike_at_depth_0 <- gps_vel_hk_sp$strike_at_depth_0 - pi

#finding velocities perpendicular and parallel to the plate convergence
#assigning plate convergence angle for each velocity vector (in radians)
gps_vel_hk_sp$plate_movement <- NA

for(i in 1:nrow(gps_vel_hk_sp)) {
  if(is.na(gps_vel_hk_sp$strike_at_depth_0[i])) {
    gps_vel_hk_sp$plate_movement[i] <- NA
  } else {
    distances <- data.frame()
    for (j in 1:nrow(plate_movement_hk)) {
      distance <- distGeo(c(plate_movement_hk$long[j], plate_movement_hk$lat[j]),
                          c(gps_vel_hk_sp$long[i], gps_vel_hk_sp$lat[i]))
      distances <- rbind(distances, data.frame(long = plate_movement_hk$long[j],
                                               lat = plate_movement_hk$lat[j],
                                               distance = distance))
    }
    closest_point <- distances[which.min(distances$distance), ]
    gps_vel_hk_sp$plate_movement[i] <- plate_movement_hk$azimuth[plate_movement_hk$long == closest_point$l
ong & plate_movement_hk$lat == closest_point$lat]*pi/180
  }
}

#finding velocities perpendicular and parallel to plate convergence
gps_vel_hk_sp$convg_reflection_angle <- gps_vel_hk_sp$gps_vel_azimuth - gps_vel_hk_sp$plate_movement
gps_vel_hk_sp$v_plate_covg_parallel <- gps_vel_hk_sp$magnitude*cos(gps_vel_hk_sp$convg_reflection_angle)
gps_vel_hk_sp$v_plate_covg_perpen <- gps_vel_hk_sp$magnitude*sin(gps_vel_hk_sp$convg_reflection_angle)

#graph
gps_vel_hk_sp_convg <- pivot_longer(gps_vel_hk_sp, cols = c(v_plate_covg_parallel, v_plate_covg_perpen), n
ames_to = "velocity_type", values_to = "velocity")

ggplot(data = gps_vel_hk_sp_convg, aes(x = dist_from_trench, y = velocity, color = velocity_type)) +
  geom_point(data=filter(gps_vel_hk_sp_convg, dist_from_trench < 600)) +
  labs(title = "Velocity Against Distance from Trench for Slip Partitioning Region
in Hokkaido",
       x = "Distance from Trench (km)",
       y = "Velocity (cm/yr)",
      color = "Velocity") +
  scale_y_continuous(limits=c(0,4.5), breaks=seq(0,5,0.5)) +
  scale_x_continuous(limits=c(0,250), breaks=seq(0,250,50)) +
  scale_color_hue(labels = c("v_plate_covg_parallel" = "Plate Convergence Parallel", "v_plate_covg_perpen"
= "Plate Convergence Perpendicular"))+
  theme(
    plot.title = element_text(size = 14, face = "bold.italic"),
    panel.background = element_rect(fill="#FFFFFF"),
    axis.line = element_line(colour="#555555"),
    axis.ticks = element_line(colour="#111111"),
```

```
    panel.grid.major = element_line(colour = "lightgrey", size = 0.2),
    panel.grid.minor = element_line(colour = "lightgrey", size = 0.2),
    legend.text = element_text(size = 9),
    legend.title = element_text(size = 10))
```