Contemporary\_Chinese\_groups

DC

2023-12-07

Table of Contents

# Install packages

# List of required packages  
required\_packages <- c(  
 "tidyverse", # Data manipulation and visualization  
 "sf", # Handling spatial data  
 "raster", # Raster data operations  
 "ggmap", # Mapping and spatial visualization  
 "sp", # Spatial data classes and methods (supports sf)  
 "akima", # Interpolation of irregularly spaced data  
 "rnaturalearth", # Natural Earth map data  
 "car", # Companion to Applied Regression  
 "MASS", # Various statistical methods  
 "sandwich", # Robust covariance matrix estimators  
 "lmtest", # Testing linear regression models  
 "scales", # Scaling functions for ggplot2  
 "gstat" , # Geostatistical modeling and kriging  
 "terra"  
)  
  
# Identify missing packages and install them  
missing\_packages <- setdiff(required\_packages, rownames(installed.packages()))  
if (length(missing\_packages) > 0) {  
 install.packages(missing\_packages)  
}  
  
# Load all required libraries  
invisible(lapply(required\_packages, library, character.only = TRUE))

# 1. Prepare the data

Read the group data on contemporary populations

groups <- read\_csv("data/modern\_multiple\_groups.csv")

## Assign the climatic data

### Read the climatic data

# Load the TIFF file  
Min\_temp <- raster("data/wc2.1\_30s\_bio\_6.tif")  
Max\_temp <- raster("data/wc2.1\_30s\_bio\_5.tif")  
Min\_precip <- raster("data/wc2.1\_30s\_bio\_14.tif")  
Max\_precip <- raster("data/wc2.1\_30s\_bio\_13.tif")

### Assign the clmatic data to modern groups

coordinates <- groups %>%  
 dplyr::select(`Longitude (E)`, `Latitude (N)`) # Use backticks around column names with spaces  
  
# Convert to a spatial points object  
points <- SpatialPoints(coordinates)

# Extract raster values for each point  
min\_temp\_values <- extract(Min\_temp, points)  
max\_temp\_values <- extract(Max\_temp, points)  
min\_precip\_values <- extract(Min\_precip, points)  
max\_precip\_values <- extract(Max\_precip, points)

# Add the extracted data back to the dataset  
groups <- groups %>%  
 mutate(Min\_temp = min\_temp\_values,  
 Max\_temp = max\_temp\_values,  
 Min\_precip = min\_precip\_values,  
 Max\_precip = max\_precip\_values)

## Get China Map

shapefile\_path <- "data/bou1\_4p.shp"  
# Read the shapefile  
china\_shape <- st\_read(shapefile\_path)

## Reading layer `bou1\_4p' from data source   
## `C:\Users\83892\Long-Term-and-Geographic-Variation-in-Body-Size-Among-Chinese-Populations\data\bou1\_4p.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 894 features and 5 fields  
## Geometry type: POLYGON  
## Dimension: XY  
## Bounding box: xmin: 73.44696 ymin: 6.318641 xmax: 135.0858 ymax: 53.55793  
## Geodetic CRS: Beijing 1954

# 2. Regression analysis

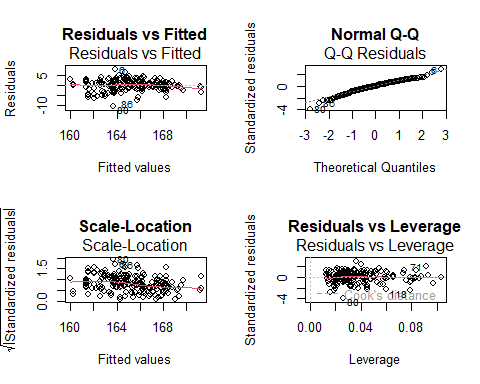
## Separate data by sex

# Use dplyr::filter() to avoid conflicts  
male\_data <- dplyr::filter(groups, Sex == "M")  
female\_data <- dplyr::filter(groups, Sex == "F")  
  
# Convert necessary columns to numeric  
male\_data <- male\_data %>%  
 mutate(Stature = as.numeric(Stature\_cm),  
 Body\_mass = as.numeric(Body\_mass\_kg),  
 Sitting\_height = as.numeric(R\_Sitting\_ht),  
 Min\_temp = as.numeric(Min\_temp),  
 Max\_temp = as.numeric(Max\_temp),  
 Min\_precip = as.numeric(Min\_precip),  
 Max\_precip = as.numeric(Max\_precip))  
  
  
female\_data <- female\_data %>%  
 mutate(Stature = as.numeric(Stature\_cm),  
 Body\_mass = as.numeric(Body\_mass\_kg),  
 Sitting\_height = as.numeric(R\_Sitting\_ht),  
 Min\_temp = as.numeric(Min\_temp),  
 Max\_temp = as.numeric(Max\_temp),  
 Min\_precip = as.numeric(Min\_precip),  
 Max\_precip = as.numeric(Max\_precip))

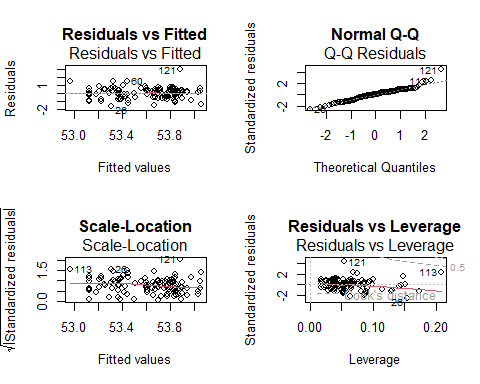
## 2.1 Ordinary linear regression

# Fit linear models for males  
male\_stature\_lm <- lm(Stature ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range + Time, data = male\_data)  
male\_sitting\_lm <- lm(Sitting\_height ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip +Altitude\_range+ Time, data = male\_data)  
male\_bodymass\_lm <- lm(Body\_mass ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range+ Time, data = male\_data)  
  
# Fit linear models for females  
female\_stature\_lm <- lm(Stature ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range+ Time, data = female\_data)  
female\_sitting\_lm <- lm(Sitting\_height ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range+ Time, data = female\_data)  
female\_bodymass\_lm <- lm(Body\_mass ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range+ Time, data = female\_data)

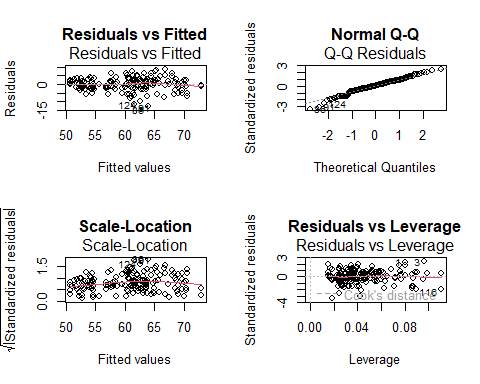
# Function to check model assumptions  
check\_assumptions <- function(model) {  
 par(mfrow=c(2,2)) # Set up plot area for 4 diagnostic plots  
  
 # 1. Residuals vs Fitted: Check for linearity  
 plot(model, which=1, main="Residuals vs Fitted")  
  
 # 2. Normal Q-Q: Check for normality of residuals  
 plot(model, which=2, main="Normal Q-Q")  
  
 # 3. Scale-Location: Check for homoscedasticity (constant variance)  
 plot(model, which=3, main="Scale-Location")  
  
 # 4. Residuals vs Leverage: Check for influential points  
 plot(model, which=5, main="Residuals vs Leverage")  
}  
  
# Function to check multicollinearity  
check\_vif <- function(model) {  
 vif(model) # VIF values above 5 indicate multicollinearity  
}  
  
# Check model assumptions and multicollinearity for males  
check\_assumptions(male\_stature\_lm)



check\_assumptions(male\_sitting\_lm)



check\_assumptions(male\_bodymass\_lm)



check\_vif(male\_stature\_lm)

## Min\_temp Max\_temp Min\_precip Max\_precip Altitude\_range   
## 3.293375 2.837753 2.078051 2.607430 2.126753   
## Time   
## 1.108396

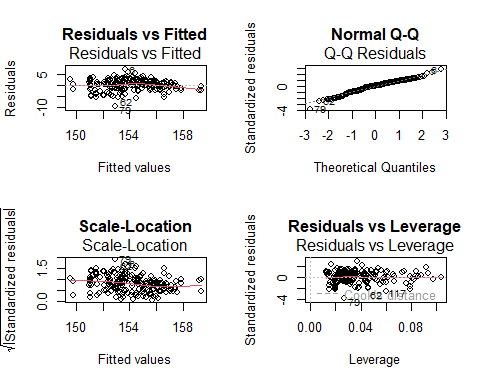
check\_vif(male\_sitting\_lm)

## Min\_temp Max\_temp Min\_precip Max\_precip Altitude\_range   
## 3.466295 3.247964 2.167538 2.681191 2.210909   
## Time   
## 1.215219

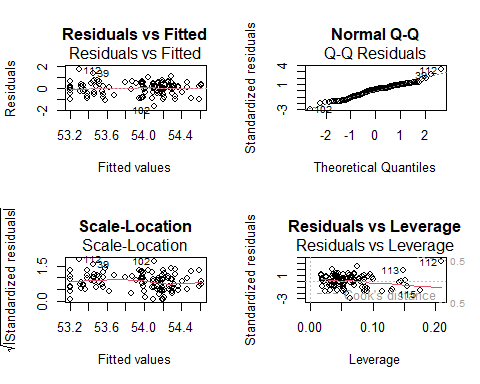
check\_vif(male\_bodymass\_lm)

## Min\_temp Max\_temp Min\_precip Max\_precip Altitude\_range   
## 3.157643 2.910266 2.009243 2.504799 2.211443   
## Time   
## 1.113348

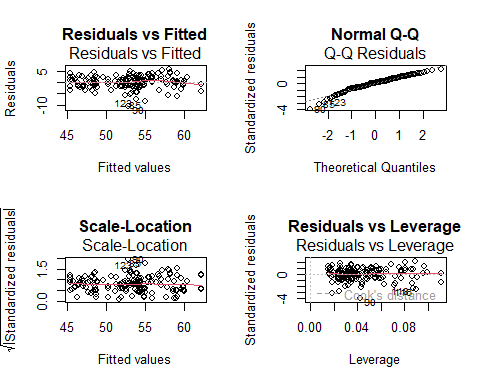
# Check model assumptions and multicollinearity for females  
check\_assumptions(female\_stature\_lm)



check\_assumptions(female\_sitting\_lm)



check\_assumptions(female\_bodymass\_lm)



check\_vif(female\_stature\_lm)

## Min\_temp Max\_temp Min\_precip Max\_precip Altitude\_range   
## 3.304653 2.833088 2.110015 2.609744 2.127544   
## Time   
## 1.105196

check\_vif(female\_sitting\_lm)

## Min\_temp Max\_temp Min\_precip Max\_precip Altitude\_range   
## 3.463109 3.241161 2.193182 2.697624 2.211682   
## Time   
## 1.211224

check\_vif(female\_bodymass\_lm)

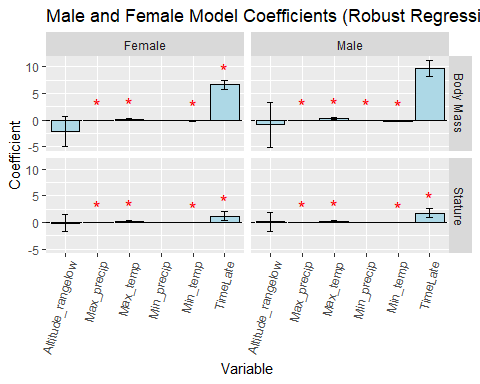
## Min\_temp Max\_temp Min\_precip Max\_precip Altitude\_range   
## 3.155644 2.905925 2.019718 2.510981 2.212148   
## Time   
## 1.111321

## 2.2 Robust regression

# Robust regression for male models  
male\_stature\_rlm <- rlm(Stature ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range +Time, data = male\_data)  
male\_sitting\_rlm <- rlm(Sitting\_height ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range +Time, data = male\_data)  
male\_bodymass\_rlm <- rlm(Body\_mass ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range +Time, data = male\_data)  
  
# Robust regression for female models  
female\_stature\_rlm <- rlm(Stature ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range +Time, data = female\_data)  
female\_sitting\_rlm <- rlm(Sitting\_height ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range +Time, data = female\_data)  
female\_bodymass\_rlm <- rlm(Body\_mass ~ Min\_temp + Max\_temp + Min\_precip + Max\_precip + Altitude\_range+Time, data = female\_data)

### Plot coefficients

# Combine male and female data for plotting  
combined\_data <- rbind(  
 data.frame(male\_combined\_data, Sex = "Male"),  
 data.frame(female\_combined\_data, Sex = "Female")  
)  
  
# Add a column for asterisks to indicate significant predictors  
combined\_data$Significant\_Asterisk <- ifelse(combined\_data$Significance == "Significant", "\*", "")  
  
combined\_plot <- ggplot(combined\_data, aes(y = Coefficient, x = Variable)) +  
 geom\_bar(stat = "identity", position = "dodge", color = "black", fill = "lightblue") + # Uniform fill color  
   
 geom\_errorbar(aes(ymin = Coefficient - 1.96 \* Std\_Error, ymax = Coefficient + 1.96 \* Std\_Error), width = 0.2, position = position\_dodge(0.9)) +  
 geom\_text(aes(label = Significant\_Asterisk), color = "red", size = 5, vjust = -0.5, position = position\_dodge(width = 0.9)) +  
 facet\_grid(Model ~ Sex) +  
 geom\_hline(yintercept = 0, linetype = "dashed", color = "black") + # Horizontal line at 0 on the y-axis  
 theme(axis.text.x = element\_text(angle = 75, hjust = 1), legend.position = "bottom") + # Adjusted axis text for readability  
 labs(title = "Male and Female Model Coefficients (Robust Regression)", y = "Coefficient", x = "Variable")  
  
# Display the combined plot  
print(combined\_plot)



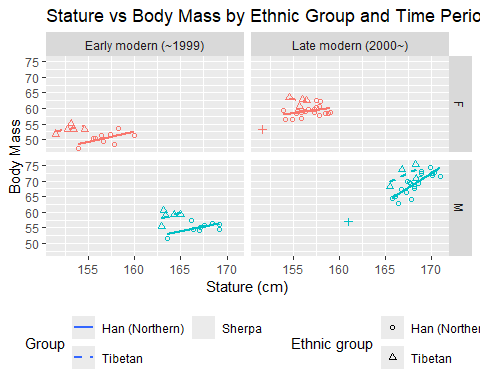
## 2.3 Compare the Han, Sherpa and Tibetan Groups

# Set the desired order for the Ethnic\_Group factor  
filtered\_data$Ethnic\_Group <- factor(filtered\_data$Ethnic\_Group, levels = c("Han", "Tibetan", "Sherpa"))  
# Convert Longitude and Latitude to numeric  
filtered\_data$Longitude\_E <- as.numeric(filtered\_data$Longitude\_E)  
filtered\_data$Latitude\_N <- as.numeric(filtered\_data$Latitude\_N)

# Set the desired order for the Ethnic\_Group factor, including all groups  
groups$Ethnic\_Group <- factor(groups$Ethnic\_Group, levels = unique(groups$Ethnic\_Group))  
  
# Create a new column named 'ethnic\_group\_2' for the desired classifications  
groups <- groups %>%  
 mutate(ethnic\_group\_2 = case\_when(  
 Ethnic\_Group == "Tibetan" ~ "Tibetan",  
 Ethnic\_Group == "Sherpa" ~ "Sherpa",  
 Ethnic\_Group == "Han" & `Latitude (N)` > 34 ~ "Han (Northern)",  
 Ethnic\_Group == "Han" & `Longitude (E)` <= 34 ~ "Han (Southern)",  
 TRUE ~ NA\_character\_  
 ))

# Plot with merged time periods for Body Mass with linear regression line using 'ethnic\_group\_2'  
body\_mass\_plot <- ggplot(groups\_filtered, aes(x = Stature\_cm, y = Body\_mass\_kg, color = Sex, shape = ethnic\_group\_2, linetype = ethnic\_group\_2)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = FALSE) + # Add linear regression line  
 labs(x = "Stature (cm)", y = "Body Mass", title = "Stature vs Body Mass by Ethnic Group and Time Period", shape = "Ethnic group", linetype = "Group") +  
 theme(legend.position = "bottom") +  
 facet\_grid(Sex ~ Time\_period\_merged) +  
 scale\_shape\_manual(values = c(1, 2, 3, 4)) + # Set different shapes for ethnic groups  
 scale\_linetype\_manual(values = c("solid", "dashed","twodash", "blank")) + # Set different line types for ethnic groups  
 guides(shape = guide\_legend(nrow = 2), linetype = guide\_legend(nrow = 2), color = guide\_legend(nrow = 2))  
body\_mass\_plot

## `geom\_smooth()` using formula = 'y ~ x'



# 3 Interpolation over time and space

## 3.1 Stature

# Convert columns to numeric, replacing non-numeric values with NA  
convert\_to\_numeric <- function(df, columns) {  
 for (col in columns) {  
 df[[col]] <- as.numeric(df[[col]])  
 }  
 return(df)  
}  
  
# Apply the function to convert 'Body\_mass(kg)' and 'R\_Sitting\_ht' to numeric  
groups <- convert\_to\_numeric(groups, c("Body\_mass(kg)", "R\_Sitting\_ht"))  
  
# Function to subset data based on sex and time period  
subset\_data <- function(df, sex, time\_periods) {  
 subset(df, Sex == sex & Time\_period %in% time\_periods,  
 select = c(`Latitude (N)`, `Longitude (E)`, `Stature(cm)`, `Body\_mass(kg)`, R\_Sitting\_ht,   
 Ethnic\_Group, Time, Altitude\_range, Relative\_Living\_Bi\_iliaca\_breadth))  
}  
  
# Subset data for each group  
ht\_china\_male\_early <- subset\_data(groups, "M", c(1, 2))  
ht\_china\_female\_early <- subset\_data(groups, "F", c(1, 2))  
ht\_china\_male\_late <- subset\_data(groups, "M", c(3))  
ht\_china\_female\_late <- subset\_data(groups, "F", c(3))

# Function to rename columns for all datasets  
rename\_columns <- function(df) {  
 names(df) <- c("Latitude\_N", "Longitude\_E", "Stature\_cm", "Body\_mass\_kg", "R\_Sitting\_ht","Ethnic\_Group", "Time", "Altitude\_range")  
 return(df)  
}  
  
# Apply the rename function to all datasets  
ht\_china\_male\_early <- rename\_columns(ht\_china\_male\_early)  
ht\_china\_female\_early <- rename\_columns(ht\_china\_female\_early)  
ht\_china\_male\_late <- rename\_columns(ht\_china\_male\_late)  
ht\_china\_female\_late <- rename\_columns(ht\_china\_female\_late)

# Define the IDW interpolation function  
perform\_idw\_interpolation <- function(df, filename) {  
 # Check if the dataframe has coordinates set properly  
 if (!("Longitude\_E" %in% names(df)) || !("Latitude\_N" %in% names(df)) || !("Stature\_cm" %in% names(df))) {  
 stop("The data frame must contain 'Longitude\_E', 'Latitude\_N', and 'Stature\_cm' columns.")  
 }  
   
 # Define the grid extent  
 xmin <- 70  
 xmax <- 140  
 ymin <- 6  
 ymax <- 60  
  
 # Set the number of cells in x and y dimensions to ensure equal cell size  
 cell\_size <- (xmax - xmin) / 400 # Assuming 400 cells in the x-direction  
  
 # Create raster with equal horizontal and vertical resolution  
 htrChina <- raster(  
 xmn = xmin, xmx = xmax,  
 ymn = ymin, ymx = ymax,  
 res = cell\_size, # Specify resolution to be the same in both x and y  
 crs = "+proj=longlat +datum=WGS84"  
 )  
  
 # Convert the raster to points to extract the coordinates  
 htrChina\_points <- rasterToPoints(htrChina, spatial = TRUE)  
  
 # Set spatial coordinates for the input data frame  
 coordinates(df) <- ~ Longitude\_E + Latitude\_N  
  
 # Calculate distances between observation points and raster grid points  
 dists <- spDists(  
 x = coordinates(df),  
 y = coordinates(htrChina\_points),  
 longlat = TRUE  
 )  
  
 # Inverse distance weighted interpolation  
 idp <- 2  
 inv.w <- (1 / (dists ^ idp))  
 z <- (t(inv.w) %\*% matrix(df$Stature\_cm)) / apply(inv.w, 2, sum)  
  
 # Update the raster with predicted values  
 htpred.China <- htrChina  
 values(htpred.China) <- z  
  
 # Export the data as a raster file in ASCII format  
 writeRaster(htpred.China, filename = filename, format = "ascii", overwrite = TRUE)  
}  
  
# Apply the function to each group  
# Interpolation for Early Female Population  
perform\_idw\_interpolation(ht\_china\_male\_early, "HT\_IDW2\_China\_male\_early.asc")  
  
perform\_idw\_interpolation(ht\_china\_female\_early, "HT\_IDW2\_China\_female\_early.asc")  
  
# Interpolation for Late Male Population  
perform\_idw\_interpolation(ht\_china\_male\_late, "HT\_IDW2\_China\_male\_late.asc")  
  
# Interpolation for Late Female Population  
perform\_idw\_interpolation(ht\_china\_female\_late, "HT\_IDW2\_China\_female\_late.asc")

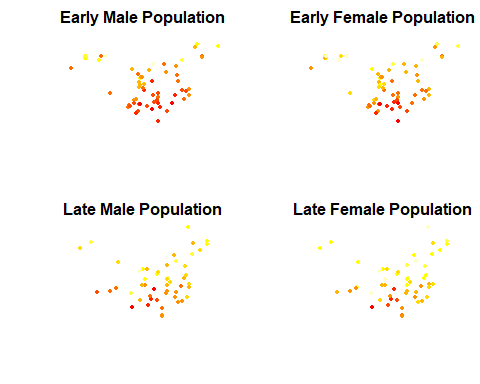
htpred\_male\_early <- raster("HT\_IDW2\_China\_male\_early.asc")  
htpred\_female\_early <- raster("HT\_IDW2\_China\_female\_early.asc")  
htpred\_male\_late <- raster("HT\_IDW2\_China\_male\_late.asc")  
htpred\_female\_late <- raster("HT\_IDW2\_China\_female\_late.asc")  
  
# Convert the rasters to data frames  
htpred\_male\_early\_df <- as.data.frame(htpred\_male\_early, xy = TRUE)  
colnames(htpred\_male\_early\_df) <- c("Longitude", "Latitude", "Stature")  
  
htpred\_female\_early\_df <- as.data.frame(htpred\_female\_early, xy = TRUE)  
colnames(htpred\_female\_early\_df) <- c("Longitude", "Latitude", "Stature")  
  
htpred\_male\_late\_df <- as.data.frame(htpred\_male\_late, xy = TRUE)  
colnames(htpred\_male\_late\_df) <- c("Longitude", "Latitude", "Stature")  
  
htpred\_female\_late\_df <- as.data.frame(htpred\_female\_late, xy = TRUE)  
colnames(htpred\_female\_late\_df) <- c("Longitude", "Latitude", "Stature")

# Function to plot raster heatmap overlaid on the China base map with dataset points by Ethnic Group using shapes and colors  
plot\_heatmap\_with\_points\_on\_china\_map <- function(raster\_df, points\_df, title) {  
 # Determine the number of unique ethnic groups  
 unique\_ethnic\_groups <- unique(points\_df$Ethnic\_Group)  
 num\_groups <- length(unique\_ethnic\_groups)  
  
 # Define shape values for each ethnic group  
 # Assign specific shapes to Han and Tibetan groups  
 shape\_values <- rep(c(15:22), length.out = num\_groups)  
 names(shape\_values) <- unique\_ethnic\_groups  
 shape\_values["Han"] <- 17 # Unify the Han group shape to a triangle  
 shape\_values["Tibetan"] <- 16 # Unify the Tibetan group shape to a circle  
shape\_values["Sherpa"] <- 15 # Unify the Tibetan group shape to a circle  
 ggplot() +  
 geom\_sf(data = china\_shape, fill = "white", color = "black") + # Plot the base map of China  
 geom\_raster(data = raster\_df, aes(x = Longitude, y = Latitude, fill = Stature), alpha = 0.7) + # Plot the heatmap  
 geom\_point(data = points\_df, aes(x = Longitude\_E, y = Latitude\_N, shape = Ethnic\_Group, color = Ethnic\_Group), size = 1) + # Plot points using different shapes and colors  
 scale\_shape\_manual(values = shape\_values) + # Assign custom shapes, including unified Han and Tibetan group shapes  
 scale\_color\_viridis\_d(option = "plasma") + # Assign colors using viridis color palette (option 'plasma' for variety)  
 scale\_fill\_viridis\_c() + # Use viridis color scale for the heatmap  
 theme\_minimal() +  
 labs(title = title,  
 x = "Longitude",  
 y = "Latitude",  
 fill = "Stature",  
 shape = "Ethnic Group",  
 color = "Ethnic Group") + # Include color legend for further differentiation  
 coord\_sf() # Keep the coordinate reference consistent  
}  
  
# Plotting each dataset separately with the heatmap and the China base map  
  
# Plot heatmap for Early Male Population with points  
plot1 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_male\_early\_df, ht\_china\_male\_early, "Heatmap with Points - Early Male Population")  
  
# Plot heatmap for Early Female Population with points  
plot2 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_female\_early\_df, ht\_china\_female\_early, "Heatmap with Points - Early Female Population")  
  
# Plot heatmap for Late Male Population with points  
plot3 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_male\_late\_df, ht\_china\_male\_late, "Heatmap with Points - Late Male Population")  
  
# Plot heatmap for Late Female Population with points  
plot4 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_female\_late\_df, ht\_china\_female\_late, "Heatmap with Points - Late Female Population")

# Set global min and max stature values across datasets for consistent heatmap scale  
global\_min <- min(htpred\_female\_early\_df$Stature, na.rm = TRUE)  
global\_max <- max(htpred\_male\_late\_df$Stature, na.rm = TRUE)  
  
# Function to plot raster heatmap overlaid on the China base map with dataset points by Ethnic Group using shapes and colors  
plot\_heatmap\_with\_points\_on\_china\_map <- function(raster\_df, points\_df, title, group\_name) {  
 # Determine the number of unique ethnic groups  
 unique\_ethnic\_groups <- unique(points\_df$Ethnic\_Group)  
 num\_groups <- length(unique\_ethnic\_groups)  
  
 # Define shape values for each ethnic group  
 # Assign specific shapes to Han, Tibetan, and Sherpa groups  
 shape\_values <- rep(c(1:14), length.out = num\_groups)  
 names(shape\_values) <- unique\_ethnic\_groups  
 shape\_values["Han"] <- 15 # Unify the Han group shape   
 shape\_values["Tibetan"] <-19 # Unify the Tibetan group shape  
 shape\_values["Sherpa"] <- 17 # Assign specific shape for Sherpa group  
 shape\_values["Mongols"] <- 18 # Assign specific shape for Mongols group  
 shape\_values["Zhuang"] <- 20 # Assign specific shape for Zhuang group  
 shape\_values["Uyghurs"] <- 21 # Assign specific shape for Uyghurs group  
 shape\_values["Gelao"] <- 22 # Assign specific shape for Gelao group  
 # Create the plot with unified layout settings  
 p <- ggplot() +  
 geom\_sf(data = china\_shape, fill = "white", color = "black") + # Plot the base map of China  
 geom\_raster(data = raster\_df, aes(x = Longitude, y = Latitude, fill = Stature), alpha = 0.7) + # Plot the heatmap  
 geom\_point(data = points\_df, aes(x = Longitude\_E, y = Latitude\_N, shape = Ethnic\_Group, color = Ethnic\_Group), size = 1) + # Plot points using different shapes and colors  
 scale\_shape\_manual(values = shape\_values) + # Assign custom shapes, including unified Han, Tibetan, and Sherpa group shapes  
 scale\_color\_viridis\_d(option = "plasma") + # Assign colors using viridis color palette (option 'plasma' for variety)  
 scale\_fill\_viridis\_c(limits = c(global\_min, global\_max)) + # Use viridis color scale for the heatmap with unified limits  
 theme\_minimal() +  
 labs(title = title,  
 x = "Longitude",  
 y = "Latitude",  
 fill = "Stature(cm)",  
 shape = "Ethnic Group",  
 color = "Ethnic Group",  
 caption = group\_name) + # Add the group name under the plot  
 theme(  
 plot.title = element\_text(size = 10, face = "bold", hjust = 0.5), # Center the title and make it bold  
 axis.title = element\_text(size = 10), # Standardize axis title font size  
 axis.text = element\_text(size = 10), # Standardize axis label size  
 legend.position = "right", # Move the legend (scale) to the left  
 legend.title = element\_text(size = 10),# Standardize legend title size  
 legend.text = element\_text(size = 8), # Standardize legend text size  
 plot.margin = unit(c(1, 1, 1, 1), "cm") # Set consistent plot margins  
 ) +  
 coord\_sf() # Keep the coordinate reference consistent  
  
 return(p)  
}  
  
# Plotting each dataset separately with the heatmap and the China base map  
  
# Plot heatmap for Early Male Population with points  
plot1 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_male\_early\_df, ht\_china\_male\_early, "Heatmap with Points - Early Male Population", "Early Male Population")  
  
# Plot heatmap for Early Female Population with points  
plot2 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_female\_early\_df, ht\_china\_female\_early, "Heatmap with Points - Early Female Population", "Early Female Population")  
  
# Plot heatmap for Late Male Population with points  
plot3 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_male\_late\_df, ht\_china\_male\_late, "Heatmap with Points - Late Male Population", "Late Male Population")  
  
# Plot heatmap for Late Female Population with points  
plot4 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_female\_late\_df, ht\_china\_female\_late, "Heatmap with Points - Late Female Population", "Late Female Population")

## 3.2 Body Mass

# Convert columns to numeric, replacing non-numeric values with NA  
convert\_to\_numeric <- function(df, columns) {  
 for (col in columns) {  
 df[[col]] <- suppressWarnings(as.numeric(df[[col]])) # Suppress warnings to avoid output clutter  
 }  
 return(df)  
}  
  
# Apply the function to convert columns to numeric  
groups <- convert\_to\_numeric(groups, c("Body\_mass(kg)", "R\_Sitting\_ht"))  
  
# Function to subset data based on sex and time period  
subset\_data <- function(df, sex, time\_periods) {  
 subset(df, Sex == sex & Time\_period %in% time\_periods,  
 select = c(`Latitude (N)`, `Longitude (E)`, `Stature(cm)`, `Body\_mass(kg)`, `R\_Sitting\_ht`,   
 Ethnic\_Group, Time, Altitude\_range, `Relative\_Living\_Bi\_iliaca\_breadth`))  
}  
  
# Subset data for each group  
ht\_china\_male\_early <- subset\_data(groups, "M", c(1, 2))  
ht\_china\_female\_early <- subset\_data(groups, "F", c(1, 2))  
ht\_china\_male\_late <- subset\_data(groups, "M", c(3))  
ht\_china\_female\_late <- subset\_data(groups, "F", c(3))  
  
# Function to rename columns for all datasets  
rename\_columns <- function(df) {  
 names(df) <- c("Latitude\_N", "Longitude\_E", "Stature\_cm", "Body\_mass\_kg", "R\_Sitting\_ht",   
 "Ethnic\_Group", "Time", "Altitude\_range", "Relative\_Living\_Bi\_iliaca\_breadth")  
 return(df)  
}  
  
# Apply the rename function to all datasets  
ht\_china\_male\_early <- rename\_columns(ht\_china\_male\_early)  
ht\_china\_female\_early <- rename\_columns(ht\_china\_female\_early)  
ht\_china\_male\_late <- rename\_columns(ht\_china\_male\_late)  
ht\_china\_female\_late <- rename\_columns(ht\_china\_female\_late)  
  
# Function to set spatial coordinates and plot points for each dataset  
plot\_points <- function(df, title, color\_function = heat.colors) {  
 # Set spatial coordinates  
 coordinates(df) <- ~ Longitude\_E + Latitude\_N  
   
 # Create a plot for the current dataset using a color function  
 df$color\_index <- as.numeric(cut(df$Body\_mass\_kg, breaks = 10)) # Categorize Body\_mass\_kg into 10 bins  
 plot(df, col = color\_function(10)[df$color\_index], pch = 16, cex = 0.6, main = title)  
}  
  
# Set up the plotting area to have 2 rows and 2 columns  
par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))  
  
# Plot each dataset  
plot\_points(ht\_china\_male\_early, title = "Early Male Population")  
plot\_points(ht\_china\_female\_early, title = "Early Female Population")  
plot\_points(ht\_china\_male\_late, title = "Late Male Population")  
plot\_points(ht\_china\_female\_late, title = "Late Female Population")



# Define the IDW interpolation function  
perform\_idw\_interpolation <- function(df, filename) {  
 # Check if the dataframe has coordinates set properly  
 if (!("Longitude\_E" %in% names(df)) || !("Latitude\_N" %in% names(df)) || !("Body\_mass\_kg" %in% names(df))) {  
 stop("The data frame must contain 'Longitude\_E', 'Latitude\_N', and 'Body\_mass\_kg' columns.")  
 }  
   
 # Remove rows with NA values in relevant columns  
 df <- df[!is.na(df$Longitude\_E) & !is.na(df$Latitude\_N) & !is.na(df$Body\_mass\_kg), ]  
   
 # Define the grid extent  
 xmin <- 70  
 xmax <- 140  
 ymin <- 6  
 ymax <- 60  
  
 # Set the number of cells in x and y dimensions to ensure equal cell size  
 cell\_size <- (xmax - xmin) / 400 # Assuming 400 cells in the x-direction  
  
 # Create raster with equal horizontal and vertical resolution  
 htrChina <- raster(  
 xmn = xmin, xmx = xmax,  
 ymn = ymin, ymx = ymax,  
 res = cell\_size, # Specify resolution to be the same in both x and y  
 crs = "+proj=longlat +datum=WGS84"  
 )  
  
 # Convert the raster to points to extract the coordinates  
 htrChina\_points <- rasterToPoints(htrChina, spatial = TRUE)  
  
 # Set spatial coordinates for the input data frame  
 coordinates(df) <- ~ Longitude\_E + Latitude\_N  
  
 # Calculate distances between observation points and raster grid points  
 dists <- spDists(  
 x = coordinates(df),  
 y = coordinates(htrChina\_points),  
 longlat = TRUE  
 )  
  
 # Inverse distance weighted interpolation  
 idp <- 2  
 inv.w <- (1 / (dists ^ idp))  
 z <- (t(inv.w) %\*% matrix(df$Body\_mass\_kg)) / apply(inv.w, 2, sum)  
  
 # Update the raster with predicted values  
 htpred.China <- htrChina  
 values(htpred.China) <- z  
  
 # Export the data as a raster file in ASCII format  
 writeRaster(htpred.China, filename = filename, format = "ascii", overwrite = TRUE)  
}  
  
# Apply the function to each group  
perform\_idw\_interpolation(ht\_china\_male\_early, "HT\_IDW2\_China\_male\_early.asc")  
  
# Interpolation for Early Female Population  
perform\_idw\_interpolation(ht\_china\_female\_early, "HT\_IDW2\_China\_female\_early.asc")  
  
# Interpolation for Late Male Population  
perform\_idw\_interpolation(ht\_china\_male\_late, "HT\_IDW2\_China\_male\_late.asc")  
  
# Interpolation for Late Female Population  
perform\_idw\_interpolation(ht\_china\_female\_late, "HT\_IDW2\_China\_female\_late.asc")

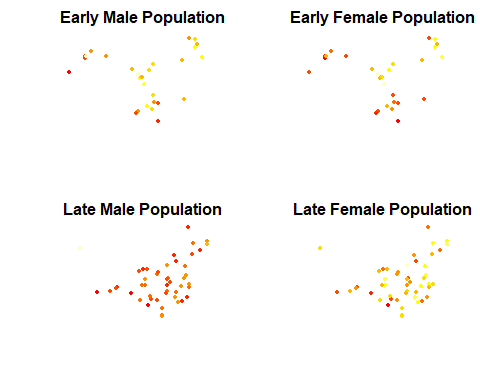
# Load the interpolated rasters  
htpred\_male\_early <- raster("HT\_IDW2\_China\_female\_early.asc")  
htpred\_female\_early <- raster("HT\_IDW2\_China\_female\_early.asc")  
htpred\_male\_late <- raster("HT\_IDW2\_China\_male\_late.asc")  
htpred\_female\_late <- raster("HT\_IDW2\_China\_female\_late.asc")  
  
# Convert the rasters to data frames  
htpred\_male\_early\_df <- as.data.frame(htpred\_male\_early, xy = TRUE)  
colnames(htpred\_male\_early\_df) <- c("Longitude", "Latitude", "Body\_mass\_kg")  
  
htpred\_female\_early\_df <- as.data.frame(htpred\_female\_early, xy = TRUE)  
colnames(htpred\_female\_early\_df) <- c("Longitude", "Latitude", "Body\_mass\_kg")  
  
htpred\_male\_late\_df <- as.data.frame(htpred\_male\_late, xy = TRUE)  
colnames(htpred\_male\_late\_df) <- c("Longitude", "Latitude", "Body\_mass\_kg")  
  
htpred\_female\_late\_df <- as.data.frame(htpred\_female\_late, xy = TRUE)  
colnames(htpred\_female\_late\_df) <- c("Longitude", "Latitude", "Body\_mass\_kg")

# Calculate global min and max for Body\_mass\_kg across all datasets to unify the scale  
global\_min <- min(c(min(htpred\_male\_early\_df $Body\_mass\_kg, na.rm = TRUE),  
 min(htpred\_female\_early\_df$Body\_mass\_kg, na.rm = TRUE),  
 min(htpred\_male\_late\_df$Body\_mass\_kg, na.rm = TRUE),  
 min(htpred\_female\_late\_df$Body\_mass\_kg, na.rm = TRUE)))  
  
global\_max <- max(c(max(htpred\_male\_early\_df$Body\_mass\_kg, na.rm = TRUE),  
 max(htpred\_female\_early\_df$Body\_mass\_kg, na.rm = TRUE),  
 max(htpred\_male\_late\_df$Body\_mass\_kg, na.rm = TRUE),  
 max(htpred\_female\_late\_df$Body\_mass\_kg, na.rm = TRUE)))

# Function to plot raster heatmap overlaid on the China base map with dataset points by Ethnic Group using shapes and colors  
plot\_heatmap\_with\_points\_on\_china\_map <- function(raster\_df, points\_df, title, group\_name) {  
 # Determine the number of unique ethnic groups  
 unique\_ethnic\_groups <- unique(points\_df$Ethnic\_Group)  
 num\_groups <- length(unique\_ethnic\_groups)  
  
 # Define shape values for each ethnic group  
 # Assign specific shapes to Han, Tibetan, and Sherpa groups  
 shape\_values <- rep(c(1:14), length.out = num\_groups)  
 names(shape\_values) <- unique\_ethnic\_groups  
 shape\_values["Han"] <- 15 # Unify the Han group shape   
 shape\_values["Tibetan"] <-19 # Unify the Tibetan group shape  
 shape\_values["Sherpa"] <- 17 # Assign specific shape for Sherpa group  
 shape\_values["Mongols"] <- 18 # Assign specific shape for Mongols group  
 shape\_values["Zhuang"] <- 20 # Assign specific shape for Zhuang group  
 shape\_values["Uyghurs"] <- 21 # Assign specific shape for Uyghurs group  
 shape\_values["Gelao"] <- 22 # Assign specific shape for Gelao group  
 # Create the plot with unified layout settings  
 p <- ggplot() +  
 geom\_sf(data = china\_shape, fill = "white", color = "black") + # Plot the base map of China  
 geom\_raster(data = raster\_df, aes(x = Longitude, y = Latitude, fill = Body\_mass\_kg), alpha = 0.7) + # Plot the heatmap  
 geom\_point(data = points\_df, aes(x = Longitude\_E, y = Latitude\_N, shape = Ethnic\_Group, color = Ethnic\_Group), size = 1) + # Plot points using different shapes and colors  
 scale\_shape\_manual(values = shape\_values) + # Assign custom shapes, including unified Han, Tibetan, and Sherpa group shapes  
 scale\_color\_viridis\_d(option = "plasma") + # Assign colors using viridis color palette (option 'plasma' for variety)  
 scale\_fill\_viridis\_c(limits = c(global\_min, global\_max)) + # Use viridis color scale for the heatmap with unified limits  
 theme\_minimal() +  
 labs(title = title,  
 x = "Longitude",  
 y = "Latitude",  
 fill = "Body mass (kg)",  
 shape = "Ethnic Group",  
 color = "Ethnic Group",  
 caption = group\_name) + # Add the group name under the plot  
 theme(  
 plot.title = element\_text(size = 10, face = "bold", hjust = 0.5), # Center the title and make it bold  
 axis.title = element\_text(size = 10), # Standardize axis title font size  
 axis.text = element\_text(size = 10), # Standardize axis label size  
 legend.position = "right", # Move the legend (scale) to the left  
 legend.title = element\_text(size = 10),# Standardize legend title size  
 legend.text = element\_text(size = 8), # Standardize legend text size  
 plot.margin = unit(c(1, 1, 1, 1), "cm") # Set consistent plot margins  
 ) +  
 coord\_sf() # Keep the coordinate reference consistent  
  
 return(p)  
}  
  
# Plotting each dataset separately with the heatmap and the China base map  
  
# Plot heatmap for Early Male Population with points  
plot1 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_male\_early\_df, ht\_china\_male\_early, "Heatmap with Points - Early Male Population", "Early Male Population")  
  
# Plot heatmap for Early Female Population with points  
plot2 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_female\_early\_df, ht\_china\_female\_early, "Heatmap with Points - Early Female Population", "Early Female Population")  
  
# Plot heatmap for Late Male Population with points  
plot3 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_male\_late\_df, ht\_china\_male\_late, "Heatmap with Points - Late Male Population", "Late Male Population")  
  
# Plot heatmap for Late Female Population with points  
plot4 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_female\_late\_df, ht\_china\_female\_late, "Heatmap with Points - Late Female Population", "Late Female Population")

## 3.3 Sitting height

# Convert columns to numeric, replacing non-numeric values with NA  
convert\_to\_numeric <- function(df, columns) {  
 for (col in columns) {  
 df[[col]] <- suppressWarnings(as.numeric(df[[col]])) # Suppress warnings to avoid output clutter  
 }  
 return(df)  
}  
  
# Apply the function to convert columns to numeric  
groups <- convert\_to\_numeric(groups, c("Body\_mass(kg)", "R\_Sitting\_ht"))  
  
# Function to subset data based on sex and time period  
subset\_data <- function(df, sex, time\_periods) {  
 subset(df, Sex == sex & Time\_period %in% time\_periods,  
 select = c(`Latitude (N)`, `Longitude (E)`, `Stature(cm)`, `Body\_mass(kg)`, `R\_Sitting\_ht`,   
 Ethnic\_Group, Time, Altitude\_range, `Relative\_Living\_Bi\_iliaca\_breadth`))  
}  
  
# Subset data for each group  
ht\_china\_male\_early <- subset\_data(groups, "M", c(1, 2))  
ht\_china\_female\_early <- subset\_data(groups, "F", c(1, 2))  
ht\_china\_male\_late <- subset\_data(groups, "M", c(3))  
ht\_china\_female\_late <- subset\_data(groups, "F", c(3))  
  
# Function to rename columns for all datasets  
rename\_columns <- function(df) {  
 names(df) <- c("Latitude\_N", "Longitude\_E", "Stature\_cm", "Body\_mass\_kg", "R\_Sitting\_ht",   
 "Ethnic\_Group", "Time", "Altitude\_range", "Relative\_Living\_Bi\_iliaca\_breadth")  
 return(df)  
}  
  
# Apply the rename function to all datasets  
ht\_china\_male\_early <- rename\_columns(ht\_china\_male\_early)  
ht\_china\_female\_early <- rename\_columns(ht\_china\_female\_early)  
ht\_china\_male\_late <- rename\_columns(ht\_china\_male\_late)  
ht\_china\_female\_late <- rename\_columns(ht\_china\_female\_late)  
  
# Function to set spatial coordinates and plot points for each dataset  
plot\_points <- function(df, title, color\_function = heat.colors) {  
 # Set spatial coordinates  
 coordinates(df) <- ~ Longitude\_E + Latitude\_N  
   
 # Create a plot for the current dataset using a color function  
 df$color\_index <- as.numeric(cut(df$R\_Sitting\_ht, breaks = 10)) # Categorize R\_Sitting\_ht into 10 bins  
 plot(df, col = color\_function(10)[df$color\_index], pch = 16, cex = 0.6, main = title)  
}  
  
# Set up the plotting area to have 2 rows and 2 columns  
par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))  
  
# Plot each dataset  
plot\_points(ht\_china\_male\_early, title = "Early Male Population")  
plot\_points(ht\_china\_female\_early, title = "Early Female Population")  
plot\_points(ht\_china\_male\_late, title = "Late Male Population")  
plot\_points(ht\_china\_female\_late, title = "Late Female Population")



# Define the IDW interpolation function  
perform\_idw\_interpolation <- function(df, filename) {  
 # Check if the dataframe has coordinates set properly  
 if (!("Longitude\_E" %in% names(df)) || !("Latitude\_N" %in% names(df)) || !("R\_Sitting\_ht" %in% names(df))) {  
 stop("The data frame must contain 'Longitude\_E', 'Latitude\_N', and 'R\_Sitting\_ht' columns.")  
 }  
   
 # Remove rows with NA values in relevant columns  
 df <- df[!is.na(df$Longitude\_E) & !is.na(df$Latitude\_N) & !is.na(df$R\_Sitting\_ht), ]  
   
 # Define the grid extent  
 xmin <- 70  
 xmax <- 140  
 ymin <- 6  
 ymax <- 60  
  
 # Set the number of cells in x and y dimensions to ensure equal cell size  
 cell\_size <- (xmax - xmin) / 400 # Assuming 400 cells in the x-direction  
  
 # Create raster with equal horizontal and vertical resolution  
 htrChina <- raster(  
 xmn = xmin, xmx = xmax,  
 ymn = ymin, ymx = ymax,  
 res = cell\_size, # Specify resolution to be the same in both x and y  
 crs = "+proj=longlat +datum=WGS84"  
 )  
  
 # Convert the raster to points to extract the coordinates  
 htrChina\_points <- rasterToPoints(htrChina, spatial = TRUE)  
  
 # Set spatial coordinates for the input data frame  
 coordinates(df) <- ~ Longitude\_E + Latitude\_N  
  
 # Calculate distances between observation points and raster grid points  
 dists <- spDists(  
 x = coordinates(df),  
 y = coordinates(htrChina\_points),  
 longlat = TRUE  
 )  
  
 # Inverse distance weighted interpolation  
 idp <- 2  
 inv.w <- (1 / (dists ^ idp))  
 z <- (t(inv.w) %\*% matrix(df$R\_Sitting\_ht)) / apply(inv.w, 2, sum)  
  
 # Update the raster with predicted values  
 htpred.China <- htrChina  
 values(htpred.China) <- z  
  
 # Export the data as a raster file in ASCII format  
 writeRaster(htpred.China, filename = filename, format = "ascii", overwrite = TRUE)  
}  
  
# Apply the function to each group  
# Interpolation for Early Female Population  
perform\_idw\_interpolation(ht\_china\_male\_early, "HT\_IDW2\_China\_male\_early.asc")  
  
# Interpolation for Early Female Population  
perform\_idw\_interpolation(ht\_china\_female\_early, "HT\_IDW2\_China\_female\_early.asc")  
  
# Interpolation for Late Male Population  
perform\_idw\_interpolation(ht\_china\_male\_late, "HT\_IDW2\_China\_male\_late.asc")  
  
# Interpolation for Late Female Population  
perform\_idw\_interpolation(ht\_china\_female\_late, "HT\_IDW2\_China\_female\_late.asc")

# Load the interpolated rasters  
htpred\_male\_early <- raster("HT\_IDW2\_China\_female\_early.asc")  
htpred\_female\_early <- raster("HT\_IDW2\_China\_female\_early.asc")  
htpred\_male\_late <- raster("HT\_IDW2\_China\_male\_late.asc")  
htpred\_female\_late <- raster("HT\_IDW2\_China\_female\_late.asc")  
  
# Convert the rasters to data frames  
htpred\_male\_early\_df <- as.data.frame(htpred\_male\_early, xy = TRUE)  
colnames(htpred\_male\_early\_df) <- c("Longitude", "Latitude", "R\_Sitting\_ht")  
  
htpred\_female\_early\_df <- as.data.frame(htpred\_female\_early, xy = TRUE)  
colnames(htpred\_female\_early\_df) <- c("Longitude", "Latitude", "R\_Sitting\_ht")  
  
htpred\_male\_late\_df <- as.data.frame(htpred\_male\_late, xy = TRUE)  
colnames(htpred\_male\_late\_df) <- c("Longitude", "Latitude", "R\_Sitting\_ht")  
  
htpred\_female\_late\_df <- as.data.frame(htpred\_female\_late, xy = TRUE)  
colnames(htpred\_female\_late\_df) <- c("Longitude", "Latitude", "R\_Sitting\_ht")

# Calculate global min and max for R\_Sitting\_ht across all datasets to unify the scale  
global\_min <- min(c(min(htpred\_male\_early\_df $R\_Sitting\_ht, na.rm = TRUE),  
 min(htpred\_female\_early\_df$R\_Sitting\_ht, na.rm = TRUE),  
 min(htpred\_male\_late\_df$R\_Sitting\_ht, na.rm = TRUE),  
 min(htpred\_female\_late\_df$R\_Sitting\_ht, na.rm = TRUE)))  
  
global\_max <- max(c(max(htpred\_male\_early\_df$R\_Sitting\_ht, na.rm = TRUE),  
 max(htpred\_female\_early\_df$R\_Sitting\_ht, na.rm = TRUE),  
 max(htpred\_male\_late\_df$R\_Sitting\_ht, na.rm = TRUE),  
 max(htpred\_female\_late\_df$R\_Sitting\_ht, na.rm = TRUE)))

# Function to plot raster heatmap overlaid on the China base map with dataset points by Ethnic Group using shapes and colors  
plot\_heatmap\_with\_points\_on\_china\_map <- function(raster\_df, points\_df, title, group\_name) {  
 # Determine the number of unique ethnic groups  
 unique\_ethnic\_groups <- unique(points\_df$Ethnic\_Group)  
 num\_groups <- length(unique\_ethnic\_groups)  
  
 # Define shape values for each ethnic group  
 # Assign specific shapes to Han, Tibetan, and Sherpa groups  
 shape\_values <- rep(c(1:14), length.out = num\_groups)  
 names(shape\_values) <- unique\_ethnic\_groups  
 shape\_values["Han"] <- 15 # Unify the Han group shape   
 shape\_values["Tibetan"] <-19 # Unify the Tibetan group shape  
 shape\_values["Sherpa"] <- 17 # Assign specific shape for Sherpa group  
 shape\_values["Mongols"] <- 18 # Assign specific shape for Mongols group  
 shape\_values["Zhuang"] <- 20 # Assign specific shape for Zhuang group  
 shape\_values["Uyghurs"] <- 21 # Assign specific shape for Uyghurs group  
 shape\_values["Gelao"] <- 22 # Assign specific shape for Gelao group  
 # Create the plot with unified layout settings  
 p <- ggplot() +  
 geom\_sf(data = china\_shape, fill = "white", color = "black") + # Plot the base map of China  
 geom\_raster(data = raster\_df, aes(x = Longitude, y = Latitude, fill = R\_Sitting\_ht), alpha = 0.7) + # Plot the heatmap  
 geom\_point(data = points\_df, aes(x = Longitude\_E, y = Latitude\_N, shape = Ethnic\_Group, color = Ethnic\_Group), size = 1) + # Plot points using different shapes and colors  
 scale\_shape\_manual(values = shape\_values) + # Assign custom shapes, including unified Han, Tibetan, and Sherpa group shapes  
 scale\_color\_viridis\_d(option = "plasma") + # Assign colors using viridis color palette (option 'plasma' for variety)  
 scale\_fill\_viridis\_c(limits = c(global\_min, global\_max)) + # Use viridis color scale for the heatmap with unified limits  
 theme\_minimal() +  
 labs(title = title,  
 x = "Longitude",  
 y = "Latitude",  
 fill = "Relative\nsitting height",  
 shape = "Ethnic Group",  
 color = "Ethnic Group",  
 caption = group\_name) + # Add the group name under the plot  
 theme(  
 plot.title = element\_text(size = 10, face = "bold", hjust = 0.5), # Center the title and make it bold  
 axis.title = element\_text(size = 10), # Standardize axis title font size  
 axis.text = element\_text(size = 10), # Standardize axis label size  
 legend.position = "right", # Move the legend (scale) to the left  
 legend.title = element\_text(size = 10),# Standardize legend title size  
 legend.text = element\_text(size = 8), # Standardize legend text size  
 plot.margin = unit(c(1, 1, 1, 1), "cm") # Set consistent plot margins  
 ) +  
 coord\_sf() # Keep the coordinate reference consistent  
  
 return(p)  
}  
  
# Plotting each dataset separately with the heatmap and the China base map  
  
# Plot heatmap for Early Male Population with points  
plot1 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_male\_early\_df, ht\_china\_male\_early, "Heatmap with Points - Early Male Population", "Early Male Population")  
  
# Plot heatmap for Early Female Population with points  
plot2 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_female\_early\_df, ht\_china\_female\_early, "Heatmap with Points - Early Female Population", "Early Female Population")  
  
# Plot heatmap for Late Male Population with points  
plot3 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_male\_late\_df, ht\_china\_male\_late, "Heatmap with Points - Late Male Population", "Late Male Population")  
  
# Plot heatmap for Late Female Population with points  
plot4 <- plot\_heatmap\_with\_points\_on\_china\_map(htpred\_female\_late\_df, ht\_china\_female\_late, "Heatmap with Points - Late Female Population", "Late Female Population")