

Code for: ‘The mechanics of male courtship display behaviour in the Ptiloris riflebirds (Aves: Paradisaeidae)’

Thomas MacGillavry

2024-08-08

Start by loading the R packages required for our analyses etc..

```
# Load required packages
library(phytools)
library(ggtree)
library(dplyr)
library(ggplot2)
library(nlme)
library(patchwork)
library(viridis)
library(ggtree)
library(scico)
library(car)
library(performance)
library(RRphylo)
```

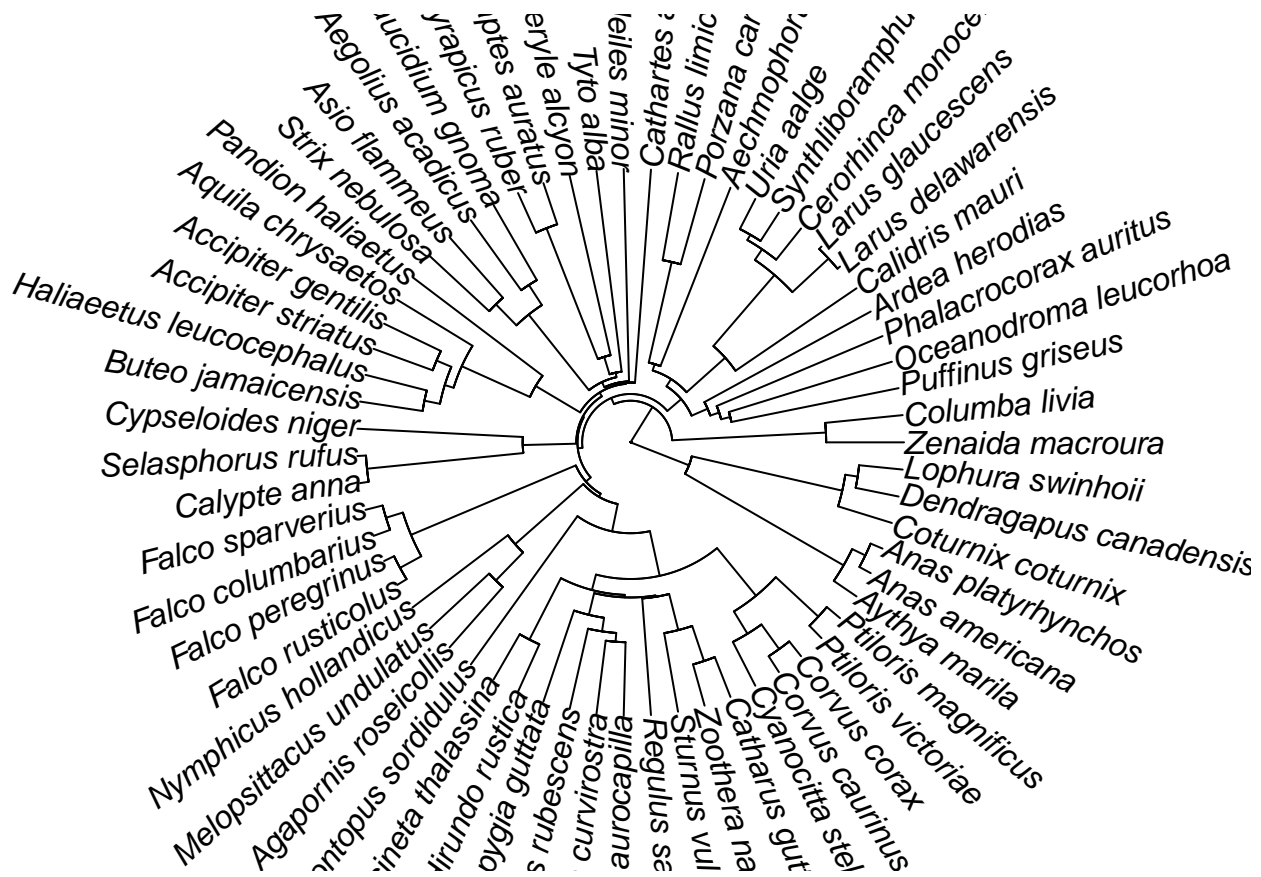
START OF PART 1: ‘WRIST’ HYPER-MOBILITY

```
## Import dataset
data <- read.csv("Baliga.2019.ManusMax.csv")
head(data, 4)

## Specify row names so the data matches the tree
rownames(data) <- data$species_phylo

## Read the tree file
## The concatenated tree, already ultrametricized in Geneious
tree <- read.nexus("ManusTree_Consesus.nex")
print(tree, printlen = 2)

## First, let's plot the tree
plotTree(tree, fsize=0.9, ftype="i", lwd=1)
```

```
## Convert to a vector
xx<-setNames(data$manusAngle.max,rownames(data))
xx

## Now we want to reconstruct ancestral states, and plot these results on a phylogeny.

## Read in data frame again
df <-read.csv("Baliga.2019.ManusMax.csv",row.names=1)
xx <- select(df, manusAngle.max) # Select only brain mass and species variable

## Change data frame to a vector
xx<-as.matrix(xx)[,1]
xx

## Estimate ancestral states
fit <- fastAnc(tree, xx, vars=TRUE, CI=TRUE)
fit

## We can also calculate 95% CIs
fit$CI[1,]
range(xx)

### Plot again in ggtree ###

# Make a dataframe with trait values at the tips
td <- data.frame(
```

```

node = nodeid(tree, names(xx)),
trait = xx)

# Make a dataframe with estimated trait values at the nodes
nd <- data.frame(node = names(fit$ace), trait = fit$ace)

# Combine these with the tree data for plotting with ggtree
d <- rbind(td, nd)
d$node <- as.numeric(d$node)
tree <- full_join(tree, d, by = 'node')

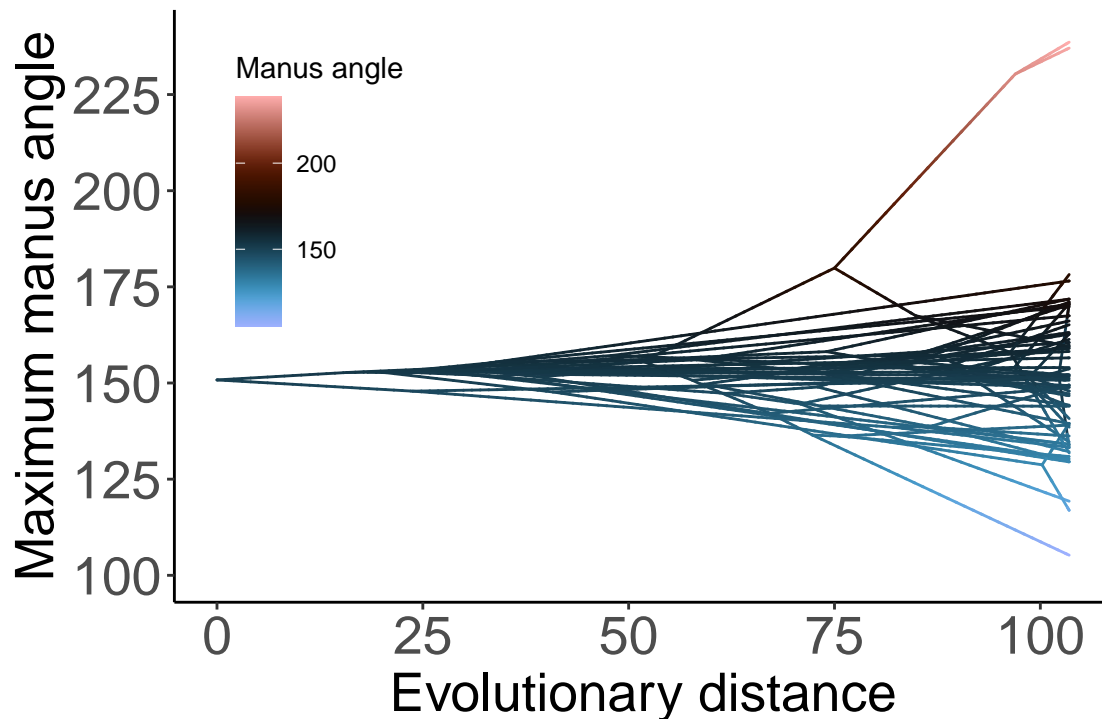
# Plot a Phenogram:

# Create the plot with the entire tree but only display tip labels for selected species
ggtree(tree, aes(color = trait), continuous = 'colour', yscale = "trait") +
  theme_classic() + # Choose theme
  scale_x_continuous(name = "Evolutionary distance",
                     breaks = seq(0, 100, by = 25)) +
  scale_y_continuous(name = "Maximum manus angle", limits = c(100, 240),
                     breaks = seq(100, 240, by = 25)) +
  theme(
    plot.margin = margin(1, 1, 1, 1, "cm"), # Adjust the margins as needed
    axis.text.x = element_text(size = 20),   # Set x-axis tick label size
    axis.text.y = element_text(size = 20),   # Set y-axis tick label size
    axis.title.x = element_text(size = 20),  # Set x-axis title size
    axis.title.y = element_text(size = 20),  # Set y-axis title size
    legend.position = c(0.05, 0.95),         # Move legend to the top left
    legend.justification = c(0, 1)           # Justify legend to the top left
  ) +
  labs(color = "Manus angle") + # Change the legend title
  scale_color_continuous(breaks = seq(0, 240, by = 25)) +
  scale_color_scico(palette = "berlin")

## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
## 3.5.0.
## i Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.

```



```
### Now we want to plot the data normally to better visualize the values of each species
```

```
# Remove underscores and replace them with spaces
```

```
data$species_phylo <- gsub("_", " ", data$species_phylo)
```

```
# Order the levels of species_phylo based on manusAngle.max
```

```
data$species_phylo <- reorder(data$species_phylo, data$manusAngle.max)
```

```
# Now, create your ggplot with the modified data
```

```
ggplot(data = data, aes(x = species_phylo, y = manusAngle.max, color = manusAngle.max)) +  
  theme_classic() +  
  geom_segment(aes(xend = species_phylo, yend = 100), color = "grey90", size = 2) +  
  geom_hline(yintercept=180, linetype="dotted", color = "grey70", size=1) +  
  geom_point(size = 5, shape = 20) +  
  ylim(100, 240) +  
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +  
  scale_color_scico(palette = "berlin") +  
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),  
        legend.position = "none") + # Remove legends  
  xlab(NULL) +  
  theme(  
    plot.margin = margin(1, 1, 1, 1, "cm"), # Adjust the margins as needed  
    axis.text.x = element_text(size = 10), # Set x-axis tick label size  
    axis.text.y = element_text(size = 8)) +  
  scale_y_continuous(name = "Maximum manus angle", limits = c(100, 240),  
                     breaks = seq(100, 240, by = 25))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
```

```
## i Please use `linewidth` instead.
```

```
## This warning is displayed once every 8 hours.
```

```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
```

```
## Adding another scale for y, which will replace the existing scale.
```



5.520142 s.d. above mean for control species!

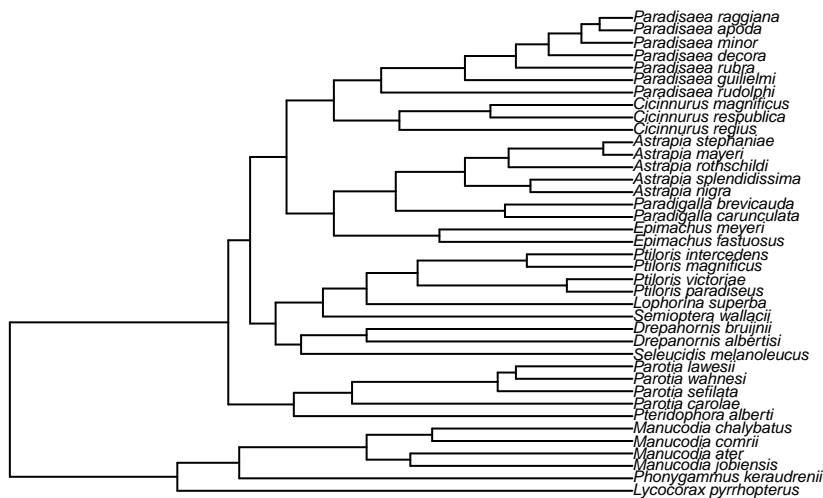
END OF PART 1

START OF PART 2: EXPLORING WING LENGTH IN THE BIRDS OF PARADISE

```
# Clear the R environment before we take a look at
# family-wide patterns of sexual dimorphism.
rm(list = ls())

# Read the dataset with species average values
df <- read.csv("Frith_Wing_species.csv", header = TRUE)
# Read the bird of paradise phylogenetic tree from Ligon et al. (2018) [modified from Irestedt et al., 2015]
tree <- read.nexus("Ligon.et.al._UltrametricTree")

# Drop the Eastern parotia from the tree [no available data for this species]
tree <- drop.tip(tree, "Parotia_helenae")
str(tree) # Inspect
plot(tree, cex = 0.5) # Plot the tree to have a look
```



```
# First, we calculate residual wing length.
# We will use residual wing length for comparative analyses

# Run model for males
Brownian.M <- gls(log10(as.numeric(wing_M)) ~ log10(as.numeric(tarsus_M)), data=df,
                  correlation=corBrownian(1,phy=tree, form = ~species), method="ML")
summary(Brownian.M)

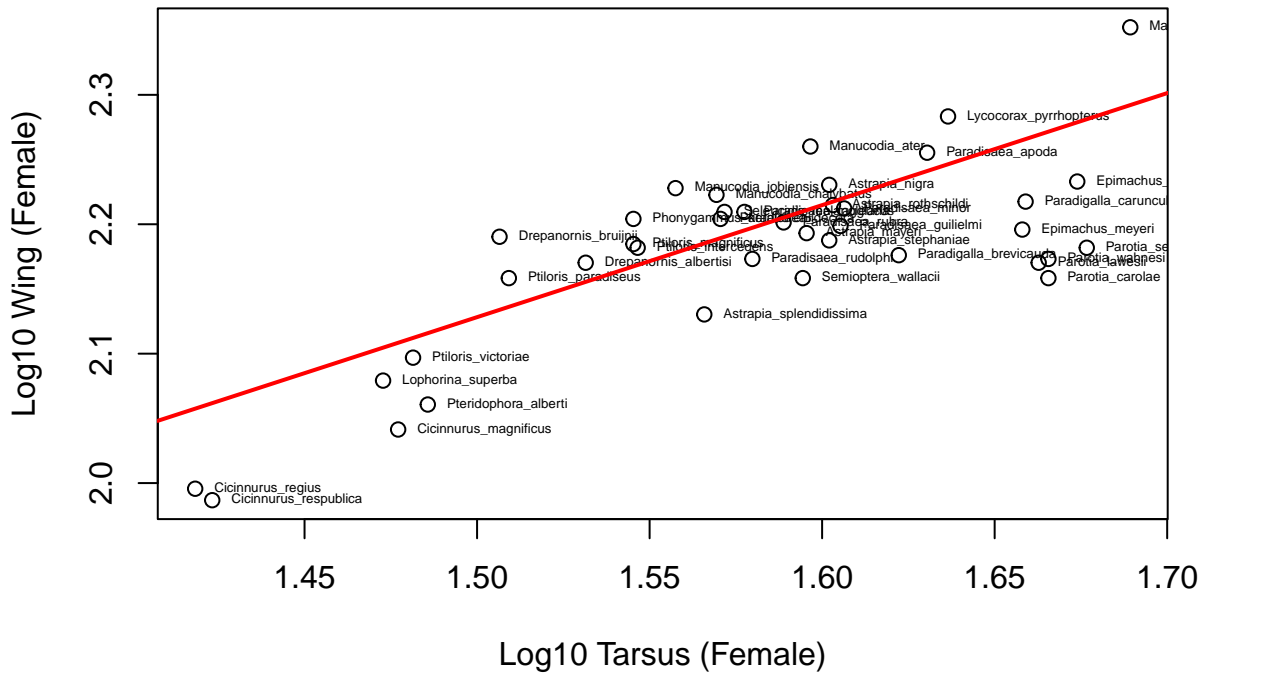
# Run model for females
Brownian.F <- gls(log10(as.numeric(wing_F)) ~ log10(as.numeric(tarsus_F)), data=df,
                  correlation=corBrownian(1,phy=tree, form = ~species), method="ML")
summary(Brownian.F)

df$resid.wing.M <- residuals(Brownian.M) # Male residual wing length
df$resid.wing.F <- residuals(Brownian.F) # Female residual wing length

# Quick plot
plot(log10(df$tarsus_F), log10(df$wing_F),
```

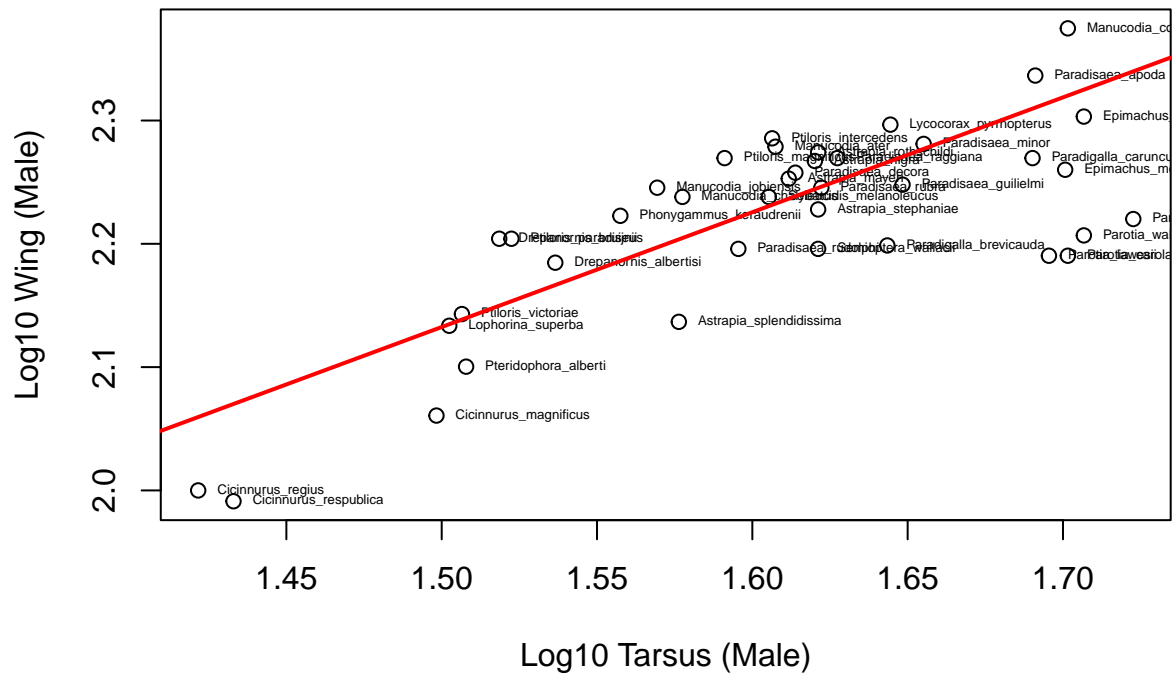
```
xlab="Log10 Tarsus (Female)", ylab="Log10 Wing (Female)",
  main="Log10 Tarsus vs Wing (Female)")
text(log10(df$tarsus_F), log10(df$wing_F), labels=df$species, pos=4, cex=0.4)
abline(a=0.8297976, b=0.8656079, col="red", lwd=2)
```

Log10 Tarsus vs Wing (Female)



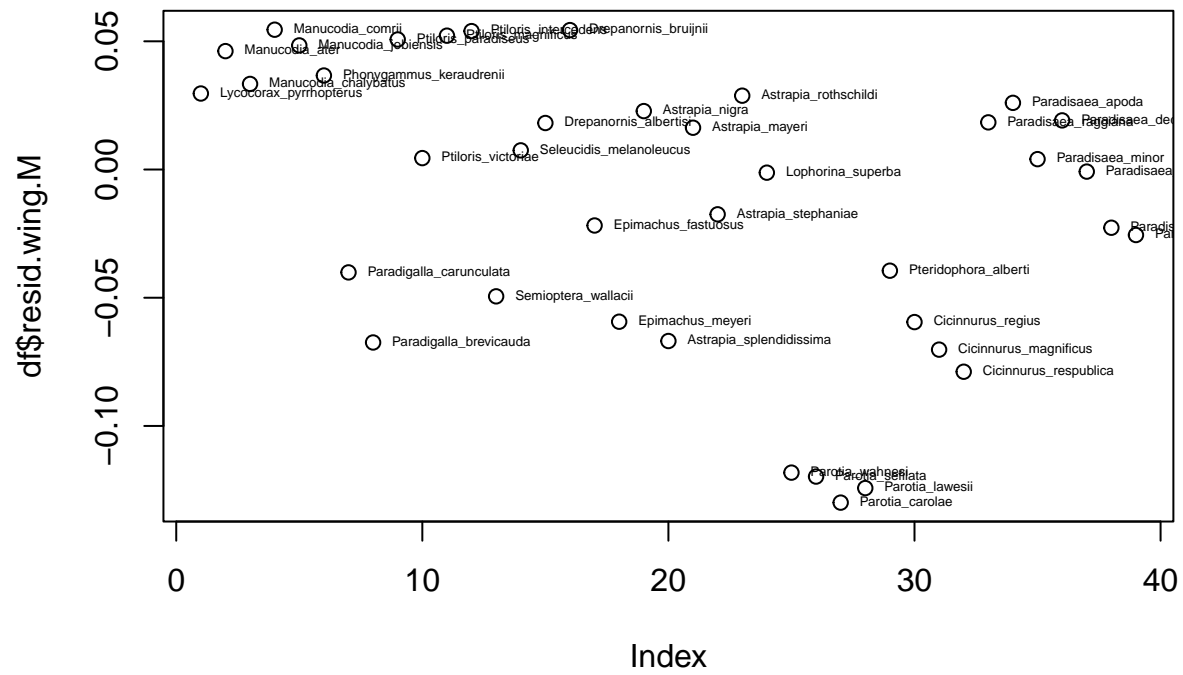
```
# Plot for tarsus_M vs. wing_M
plot(log10(df$tarsus_M), log10(df$wing_M),
     xlab="Log10 Tarsus (Male)", ylab="Log10 Wing (Male)",
     main="Log10 Tarsus vs Wing (Male)")
text(log10(df$tarsus_M), log10(df$wing_M), labels=df$species, pos=4, cex=0.4)
abline(a=0.7355498, b=0.9313050, col="red", lwd=2)
```


Log10 Tarsus vs Wing (Male)

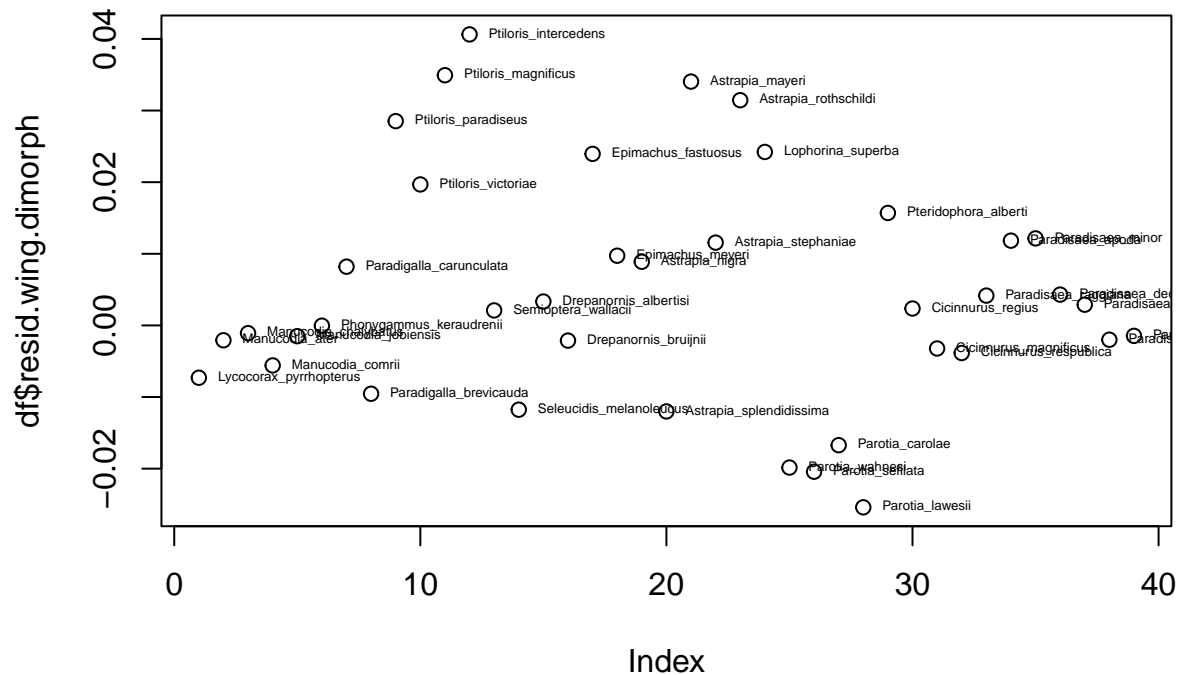


```
# Since we use residuals, which can be negative or positive,
# we calculate dimorphism by subtracting female values from male values:
df$resid.wing.dimorph <- (df$resid.wing.M) - (df$resid.wing.F)

plot(df$resid.wing.M)
text(df$resid.wing.M, labels=df$species, pos=4, cex=0.4)
```



```
plot(df$resid.wing.dimorph)
text(df$resid.wing.dimorph, labels=df$species, pos=4, cex=0.4)
```



```
head(df, 4)
```

```
# Did male riflebirds evolve much larger residual wing lengths than other birds of paradise?

# Convert 'taxon' into a factor variable
df$taxon <- factor(df$taxon)

# Now, relevel 'taxon' with "Monogamous" as reference level
df$tx <- relevel(df$taxon, ref = "Monogamous")

# Reorder levels of 'taxon' so every clade is compared to
# a baseline of the monogamous species
df$taxon <- factor(df$taxon, levels = c("Monogamous", "Core", "Core: Ptiloris"))

# Do male riflebirds have greater relative wing lengths
# than other bird of paradise species?

# Fit pGLS model
full.residwing.pgls <- gls(resid.wing.M ~ tx, data=df,
                          correlation=corPagel(1,
                                                phy=tree,
                                                form = ~species, fixed = FALSE),
                          method="ML")

# Note that we fit corPagel here to estimate Lambda using ML

# What is the estimate for lambda
coef(full.residwing.pgls$modelStruct$corStruct, unconstrained = FALSE) # 0.9258628

# Fit OLS model
```

```

full.residwing.ols <- gls(resid.wing.M ~ tx, data=df, method="ML")

# Let's have a look at the output of the models
summary(full.residwing.pgls)
summary(full.residwing.ols)

# to get confidence intervals use
xx=cbind(orig=coef(full.residwing.pgls), confint(object=full.residwing.pgls))
xx=round(xx, digits=4)
xx

# For the OLS model
xx=cbind(orig=coef(full.residwing.ols), confint(object=full.residwing.ols))
xx=round(xx, digits=4)
xx

# Check sample sizes
nobs(full.residwing.pgls) # 39 species in the model

# What about sexual dimorphism in residual wing length?

# Do male riflebirds have greater relative wing lengths
# than other bird of paradise species?

# Fit pGLS model
full.SSD.pgls <- gls(resid.wing.dimorph ~ tx, data=df,
                    correlation=corPagel(1,
                                          phy=tree,
                                          form = ~species, fixed = FALSE),
                    method="ML")

# What is the estimate for lambda
coef(full.SSD.pgls$modelStruct$corStruct, unconstrained = FALSE) # 0.7354168

# Fit OLS model
full.SSD.ols <- gls(resid.wing.dimorph ~ tx, data=df, method="ML")

# Let's have a look at the output of the models
summary(full.SSD.pgls)
summary(full.SSD.ols)

# to get confidence intervals use
xx=cbind(orig=coef(full.SSD.pgls), confint(object=full.SSD.pgls))
xx=round(xx, digits=4)
xx

# For the OLS model
xx=cbind(orig=coef(full.SSD.ols), confint(object=full.SSD.ols))
xx=round(xx, digits=4)
xx

# Check sample sizes
nobs(full.SSD.pgls) # 39 species in the model

```

```

# Plotting

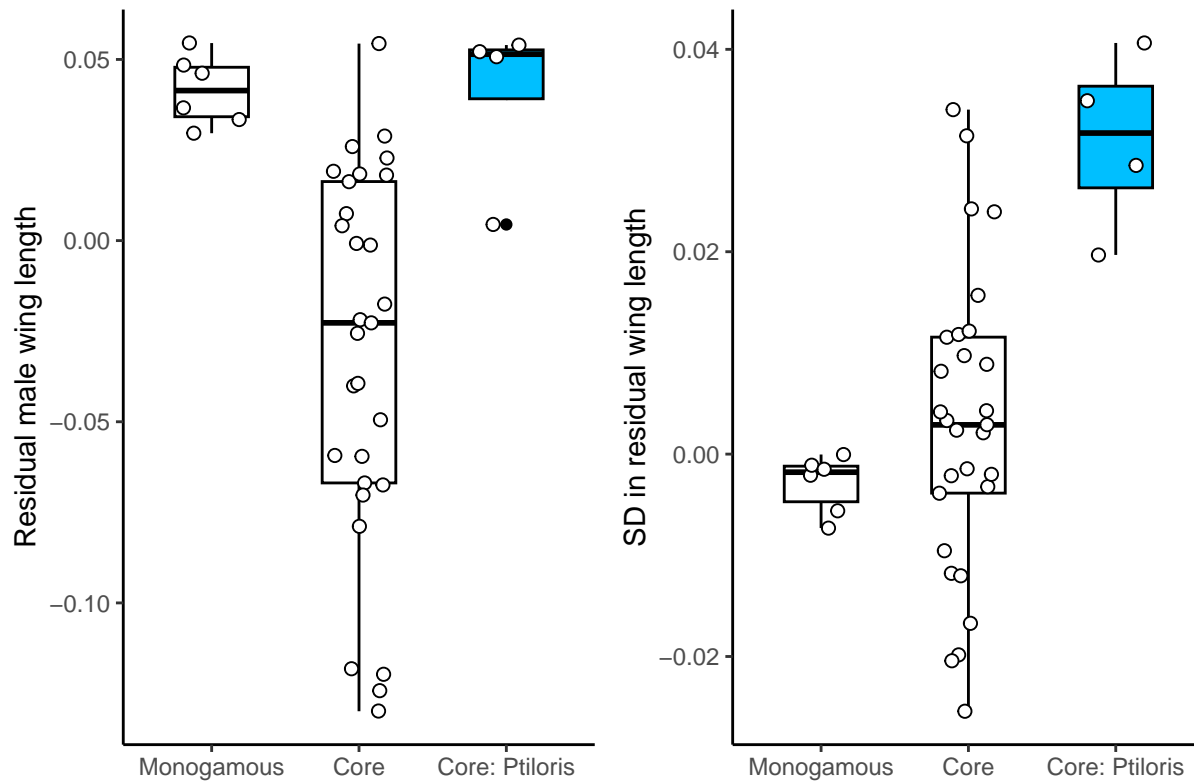
# Define color palette
cols <- c("Monogamous" = "white", "Core" = "white", "Core: Ptiloris" = "deepskyblue1")

# For resid.wing.dimorph
p.SSD <- ggplot(df, aes(x = taxon, y = resid.wing.dimorph, fill = taxon)) +
  geom_boxplot(colour = "black", width = 0.5) +
  geom_point(position = position_jitter(width = 0.2),
             size = 2, shape = 21, fill = "white") +
  scale_fill_manual(values = cols, guide = "none") +
  scale_color_manual(values = cols, guide = "none") +
  labs(y = "SD in residual wing length", x = "") +
  theme_classic() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

# For resid.wing.M
p.Res <- ggplot(df, aes(x = taxon, y = resid.wing.M, fill = taxon)) +
  geom_boxplot(colour = "black", width = 0.5) +
  geom_point(position = position_jitter(width = 0.2),
             size = 2, shape = 21, fill = "white") +
  scale_fill_manual(values = cols, guide = "none") +
  scale_color_manual(values = cols, guide = "none") +
  labs(y = "Residual male wing length", x = "") +
  theme_classic() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

# Stitch the plots together using the patchwork package
p.Res|p.SSD

```

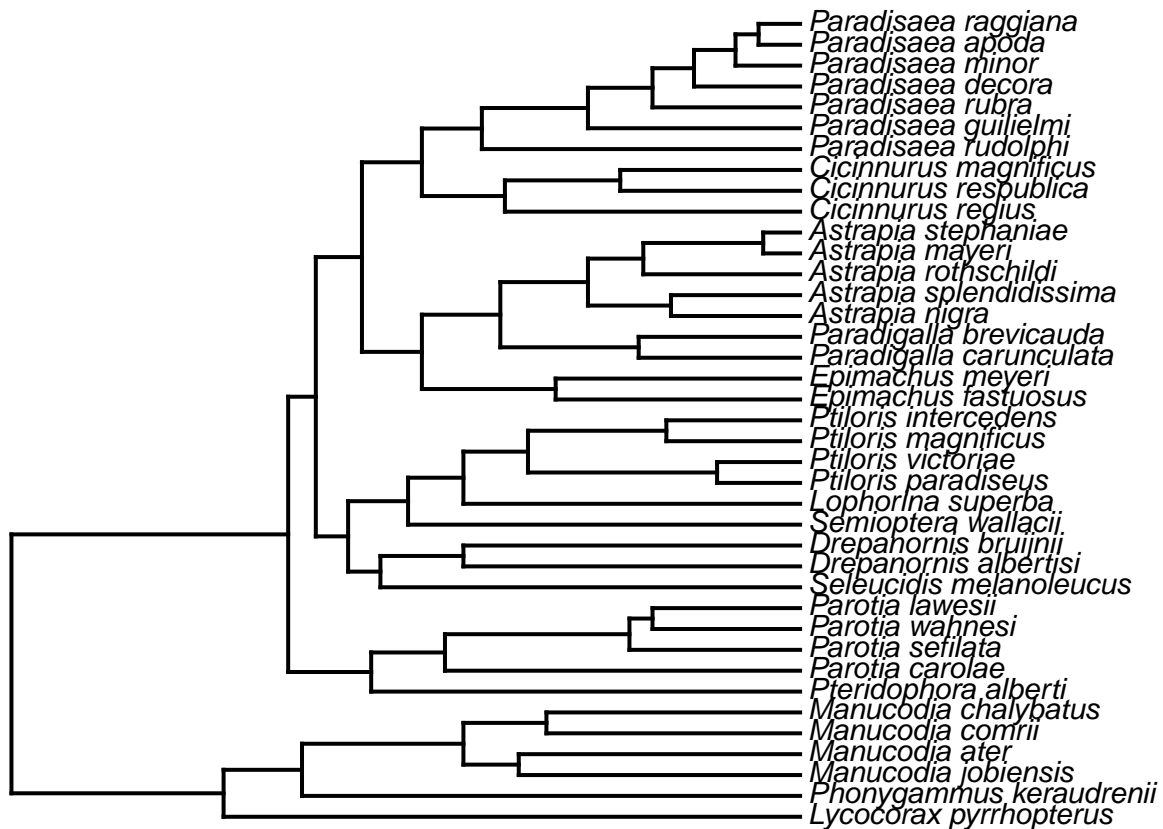


#####

Ancestral reconstruction

```
## Import the data by specifying species as row names
df <- read.csv("Frith_Wing_species.csv", row.names = 2)
tree <- read.nexus("Ligon.et.al._UltrametricTree")
# Drop specified species from the tree
tree <- drop.tip(tree, "Parotia_helenae") # We do not have data for this species

## First, let's plot the tree
plotTree(tree, fsize=0.9, ftype="i", lwd=2)
```



```
## Convert to a vector
xx<-setNames(df$resid.wing.dimorph,rownames(df))
xx

## Change data frame to a vector
xx<-as.matrix(xx)[,1]
xx

## Estimate ancestral states
fit <- fastAnc(tree, xx, vars=TRUE, CI=TRUE)
fit

## We can also calculate 95% CIs
fit$CI[1,]
range(xx)

# Make a dataframe with trait values at the tips
td <- data.frame(
  node = nodeid(tree, names(xx)),
  trait = xx)

# Make a dataframe with estimated trait values at the nodes
nd <- data.frame(node = names(fit$ace), trait = fit$ace)

# Combine these with the tree data for plotting with ggtree
d <- rbind(td, nd)
d$node <- as.numeric(d$node)
```

```
tree2 <- full_join(tree, d, by = 'node')
```

```
# Adjust plot margins and xlim
```

```
p <- ggtree(
  tree2,
  aes(color = trait),
  layout = "fan",
  ladderize = TRUE, continuous = "color", size = 1) +
  # scale_color_viridis(option = "cividis") +
  scale_color_scico(palette = "berlin") +
  theme(
    legend.position = c(-.1, .90),
    legend.text = element_text(size = 6),
    legend.title = element_text(size = 6),
    legend.key.size = unit(0.5, "lines"),
    legend.key.height = unit(0.4, "lines"),
    legend.key.width = unit(0.4, "lines"),
    axis.text.x = element_text(size = 0)) +
  labs(color = "Sexual dimorphism in
residual wing length
(male - female)") +
  geom_tiplab(aes(label = gsub("_", " ", label),
    fontface = "italic"), size = 2.5, hjust = -0.05) +
  xlim(-20, 60)
```

```
## Scale for y is already present.
```

```
## Adding another scale for y, which will replace the existing scale.
```

```
p2 <- open_tree(p, 180)
```

```
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.
```

```
## Scale for y is already present.
```

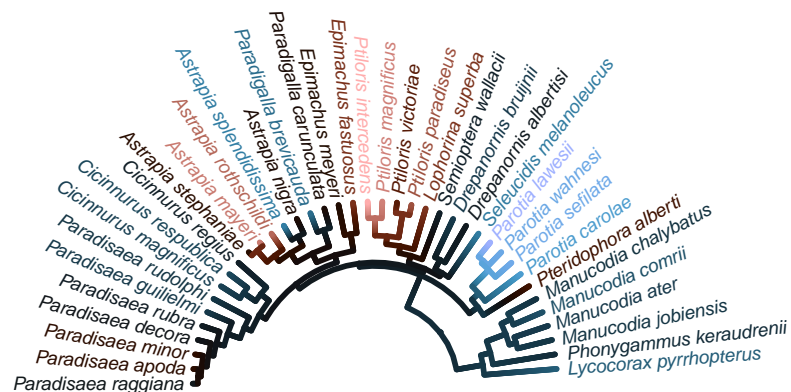
```
## Adding another scale for y, which will replace the existing scale.
```

```
## Scale for y is already present.
```

```
## Adding another scale for y, which will replace the existing scale.
```

```
print(p2)
```

Sexual dimorphism in
residual wing length
(male - female)



```
#####
```

```
#####
```

```

## Convert to a vector
xx<-setNames(df$resid.wing.M,rownames(df))
xx

## Change data frame to a vector
xx<-as.matrix(xx)[,1]
xx

## Estimate ancestral states
fit <- fastAnc(tree, xx, vars=TRUE, CI=TRUE)
fit

## We can also calculate 95% CIs
fit$CI[1,]
range(xx)

# Fit an ancestral state character reconstruction
fit <- phytools::fastAnc(tree, xx, vars = TRUE, CI = TRUE)

# Make a dataframe with trait values at the tips
td <- data.frame(
  node = nodeid(tree, names(xx)),
  trait = xx)

# Make a dataframe with estimated trait values at the nodes
nd <- data.frame(node = names(fit$ace), trait = fit$ace)

# Combine these with the tree data for plotting with ggtree
d <- rbind(td, nd)
d$node <- as.numeric(d$node)
tree3 <- full_join(tree, d, by = 'node')

# Adjust plot margins and xlim
p <- ggtree(
  tree3,
  aes(color = trait),
  layout = "fan",
  ladderize = TRUE, continuous = "color", size = 1) +
  # scale_color_viridis(option = "magma") +
  scale_color_scico(palette = "berlin") +
  theme(
    legend.position = c(-.1, .93),
    legend.text = element_text(size = 6),
    legend.title = element_text(size = 6),
    legend.key.size = unit(0.5, "lines"),
    legend.key.height = unit(0.4, "lines"),
    legend.key.width = unit(0.4, "lines"),
    axis.text.x = element_text(size = 0)) +
  labs(color = "Residual male wing length") +
  geom_tiplab(aes(label = gsub("_", " ", label),
    fontface = "italic"), size = 2.5, hjust = -0.05) +
  xlim(-20, 60)

```

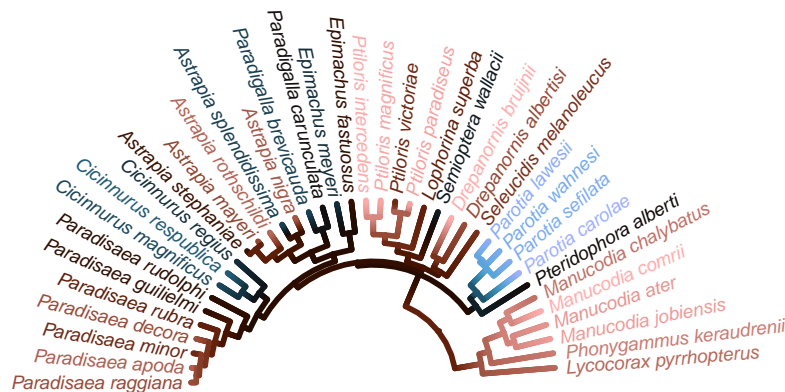
Scale for y is already present.


```
## Adding another scale for y, which will replace the existing scale.
p2 <- open_tree(p, 180)

## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.

print(p2)
```

Residual male wing length



END OF PART 2

START OF PART 3: EXPLORING WING LENGTH IN THE RIFLEBIRDS

```
# Clear the environment before running the following code to
# make things a bit less confusing
rm(list = ls())

# Load dataframes
df <- read.csv("Frith_Wing_Riflebirds.csv")
# This dataset is already clean;
# I only included observations where both tarsus and wing length were known
df <- filter(df, Age == "Adult")

# Convert "Wing" and "Tarsus" columns to numeric
df$Wing <- as.numeric(df$Wing)
df$Tarsus <- as.numeric(df$Tarsus)

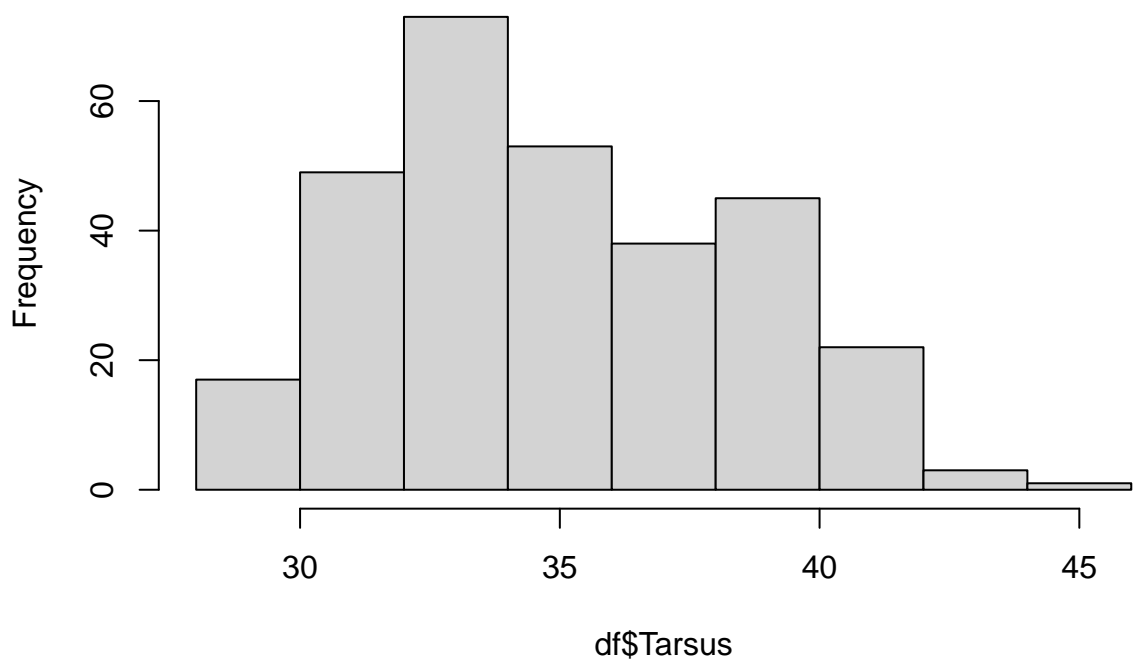
# Make sure all cases in the dataframe are either "M" or "F"
# and select only sexed individuals
df <- df %>%
  filter(Sex %in% c("F", "M"))
str(df)

# First, let's try to fit a model including species, sex,
# and tarsus length as predictors

# Inspect vars
```

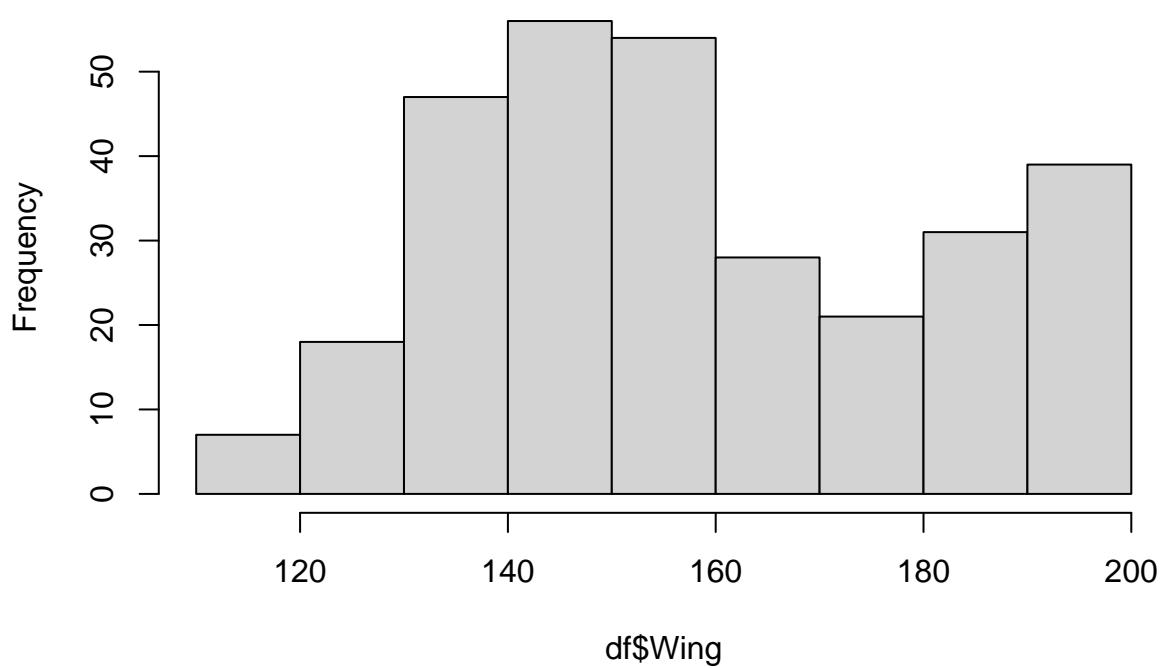
```
hist(df$Tarsus)
```

Histogram of df\$Tarsus



```
hist(df$Wing)
```

Histogram of df\$Wing



```

# z-transform tarsus length
df$z.Tarsus=as.vector(scale(df$Tarsus))

###
# Building full model with all riflebird species
###

# Build full model
full=lm(Wing ~ z.Tarsus*Sex*Species, data=df)

# Check for heteroscedasticity by inspecting VIFs
round(vif(full), 3)

## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

# No obvious issue here
vif(full)[, 3]^2

## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

# There seem to be some issues with VIF in the full model;
# to be on the safe side, run separate model for each species

```

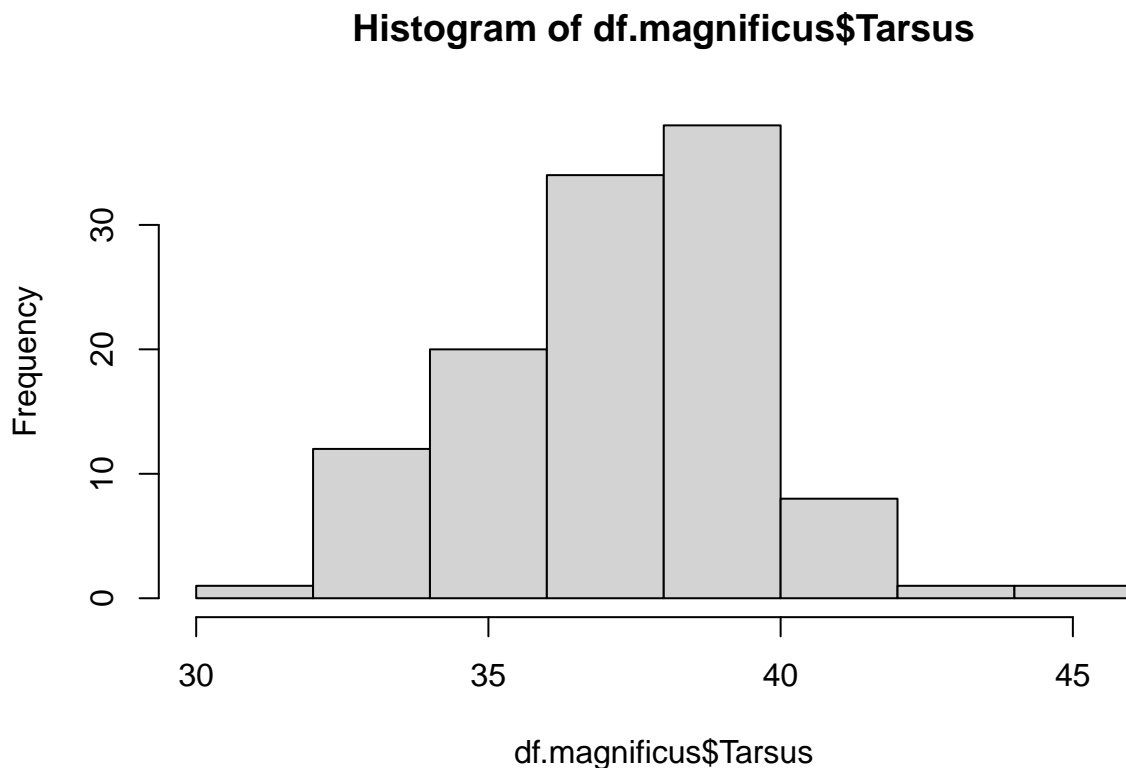
Magnificent riflebirds

```

df.magnificus <- filter(df, Species == "Ptiloris_magnificus", Age == "Adult")

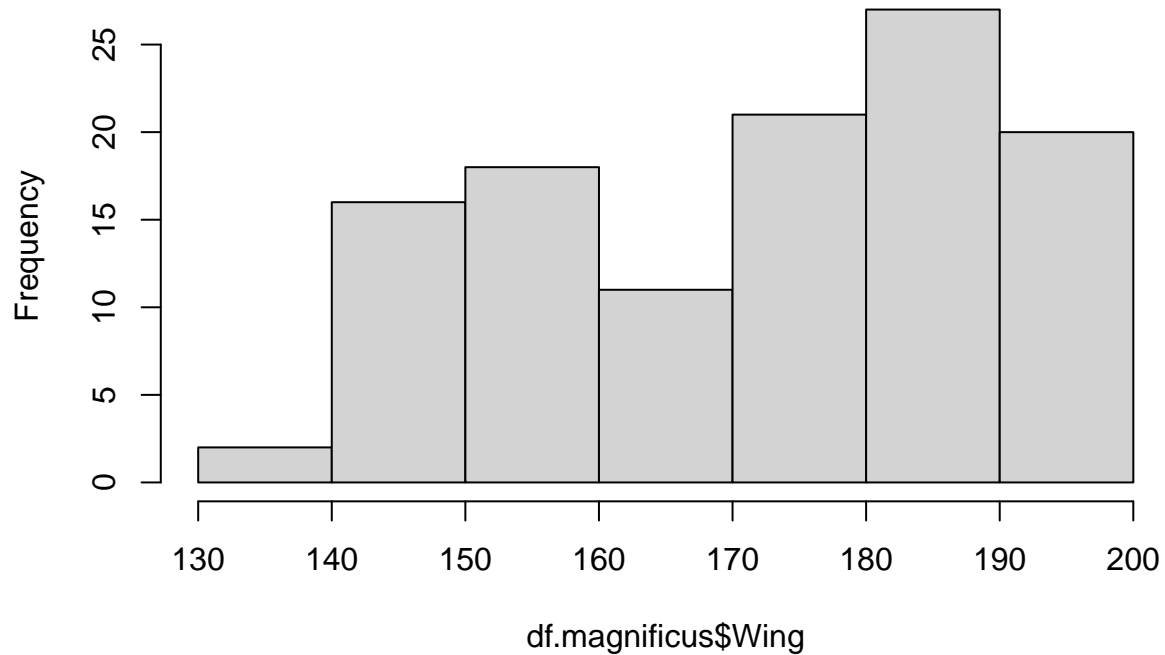
hist(df.magnificus$Tarsus)

```



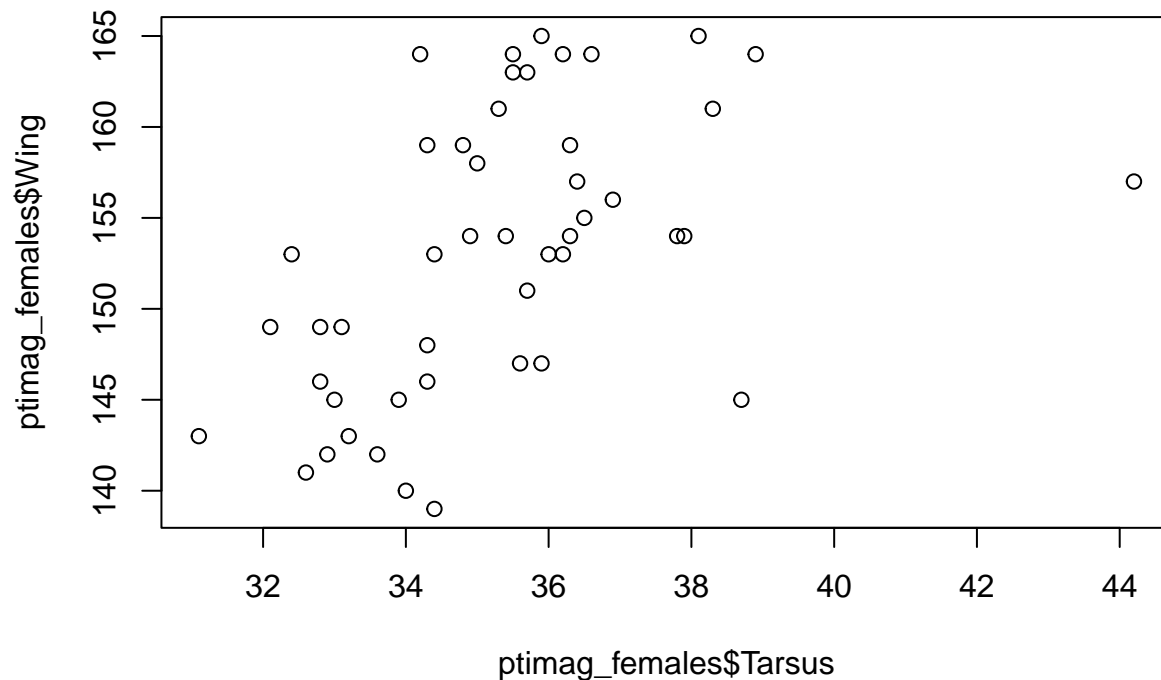
```
hist(df.magnificus$Wing)
```

Histogram of df.magnificus\$Wing



```
# z-transform tarsus length
df.magnificus$z.Tarsus=as.vector(scale(df.magnificus$Tarsus))

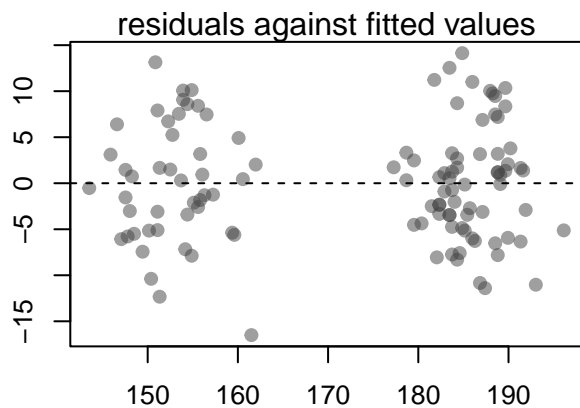
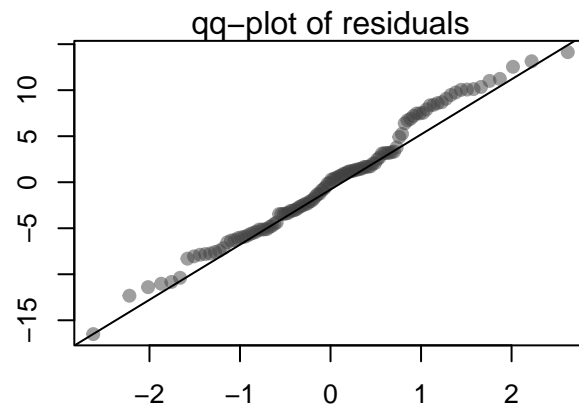
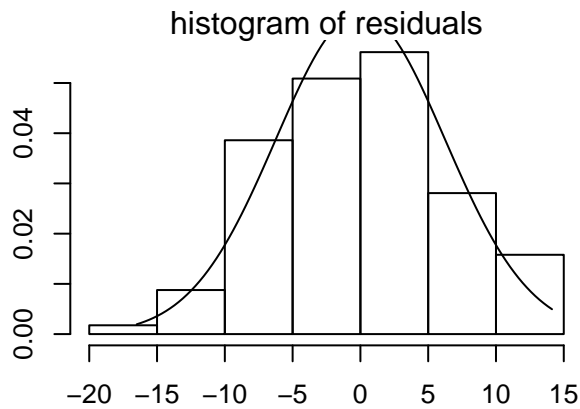
# Earlier inspection of the data suggests there may be an
# outlier in tarsus length among the females;
# One individual has a much longer tarsus length compared to other individuals
# First, fit a model with only females:
ptimag_females <- subset(df.magnificus, Sex == "F")
plot(ptimag_females$Tarsus, ptimag_females$Wing)
```



```
F.magnificus=lm(Wing ~ z.Tarsus, data=ptimag_females)
# Check for outliers:
outliers <- check_outliers(F.magnificus, method = c("zscore robust", "mcd"))
which(outliers)
# We can remove this data point for modelling,
# as it's been identified as an outlier (ID = AM018102).
head(ptimag_females, 38)
# Let's quickly remove that from the dataset:
df.magnificus <- df.magnificus[df.magnificus$ID != "AM018102", ]

# Fit full model: Wing ~ z.Tarsus + Sex + z.Tarsus x Sex
full.magnificus=lm(Wing ~ z.Tarsus*Sex, data=df.magnificus)

# Basic diagnostics
setwd("/Users/ThomasMac_1/Desktop/Mundry_Course/Roger_Functions")
source("diagnostic_fcns.r")
diagnostics.plot(full.magnificus)
```



```
# Model stability
my.dfbeta<-function(m) {
  xx=cbind(coef(m), coef(m) +
           t(apply(X=dfbeta(m), MARGIN=2, FUN=range)))
  colnames(xx)=c("orig", "min", "max")
  return(xx)
}

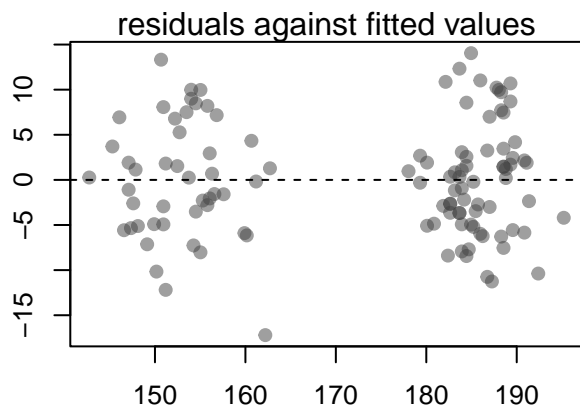
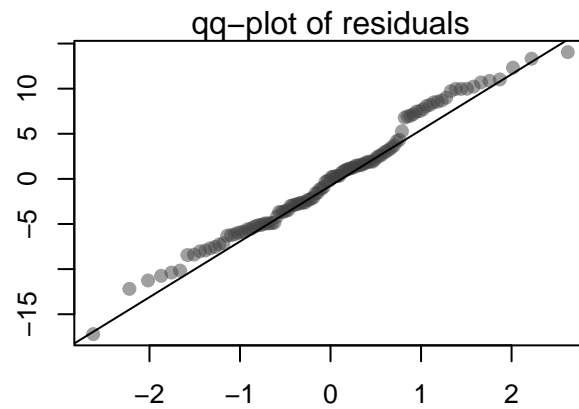
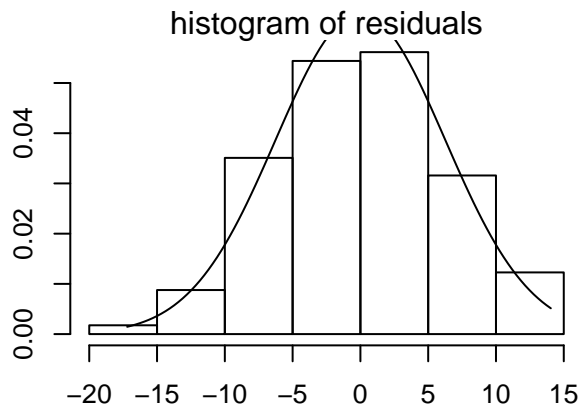
# Inspect results
round(my.dfbeta(m=full.magnificus), 3) # Generally looks quite stable

# Check for heteroscedasticity by inspecting VIFs
library(car)
round(vif(full.magnificus), 3) # No obvious issues here
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

# Fit null model without the interaction between sex and tarsus length
null.magnificus=lm(Wing ~ z.Tarsus + Sex, data=df.magnificus)

# Diagnostics for the null model
diagnostics.plot(null.magnificus)
```



```
# Compare model fits
round(as.data.frame(anova(null.magnificus, full.magnificus, test="F")), 3)
# it is non-significant (F1, 111=2.363, P = 0.127)
# We fit the null model in this case

# Let's have a look at the output of the best model
round(summary(null.magnificus)$coefficients, 3)

# to get confidence intervals use
xx=cbind(orig=coef(null.magnificus), confint(object=null.magnificus))
xx=round(xx, digits=3)
xx

# Stability check for null model
round(my.dfbeta(m=null.magnificus), 3) # Generally looks quite stable

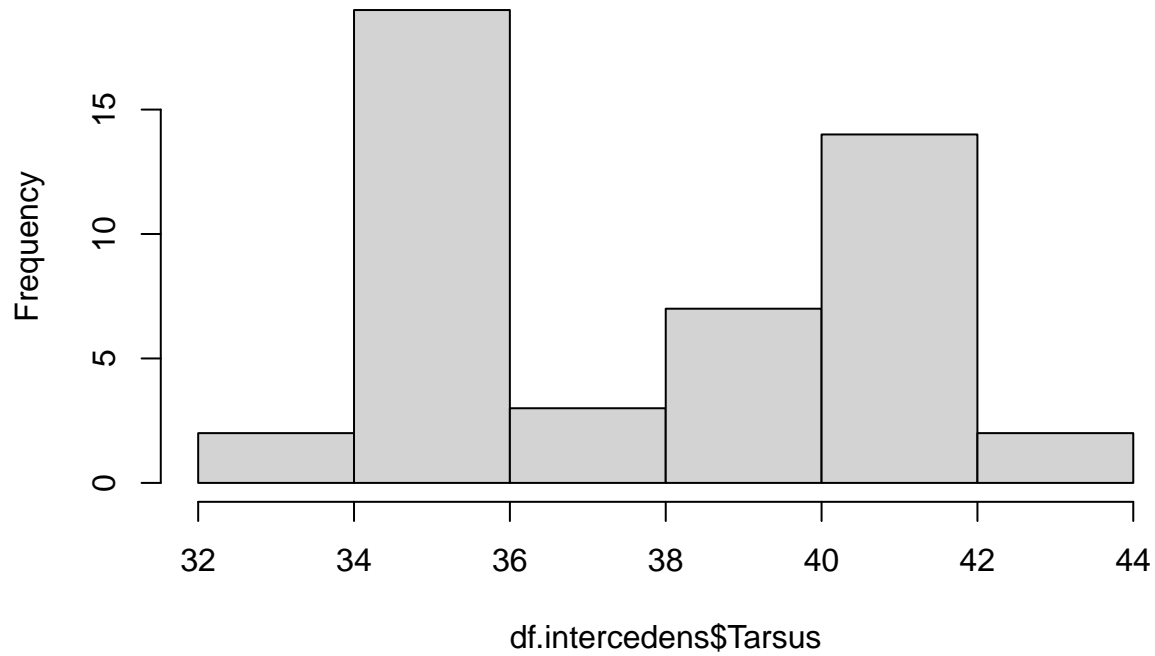
# Check sample sizes
table(df.magnificus$Sex)
```

Growling riflebirds

```
df.intercedens <- filter(df, Species == "Ptiloris_intercedens", Age == "Adult")

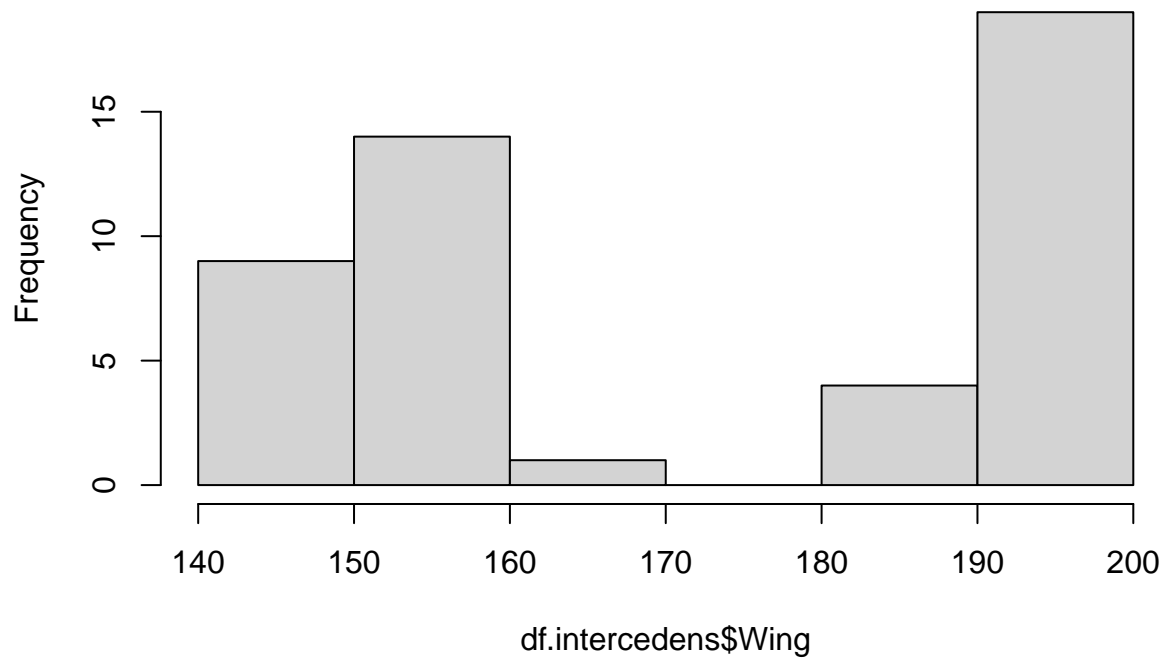
hist(df.intercedens$Tarsus)
```

Histogram of df.intercedens\$Tarsus



```
hist(df.intercedens$Wing)
```

Histogram of df.intercedens\$Wing

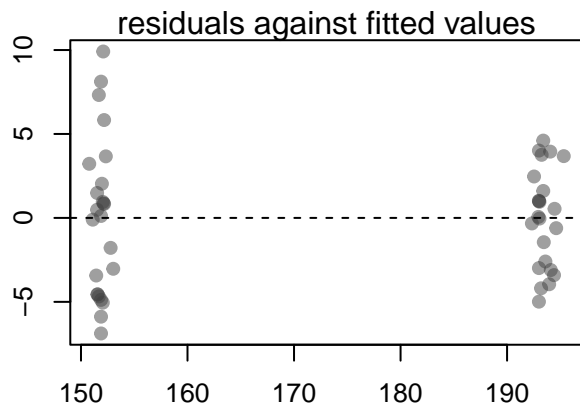
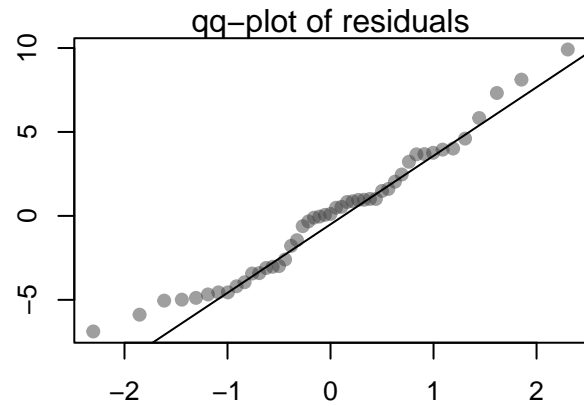
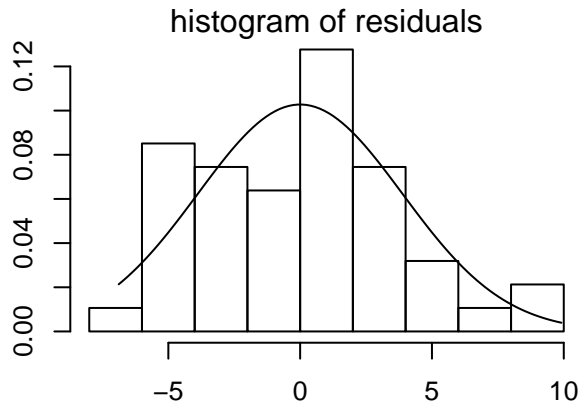


```
# z-transform tarsus length  
df.intercedens$z.Tarsus=as.vector(scale(df.intercedens$Tarsus))
```



```
# Fit full model: Wing ~ z.Tarsus + Sex + z.Tarsus x Sex
full.intercedens=lm(Wing ~ z.Tarsus*Sex, data=df.intercedens)
```

```
# Basic diagnostics
diagnostics.plot(full.intercedens)
```



```
# Inspect DF-Beta
round(my.dfbeta(m=full.intercedens), 3) # Generally looks quite stable
```

```
# Check for heteroscedasticity by inspecting VIFs
round(vif(full.intercedens), 3)
```

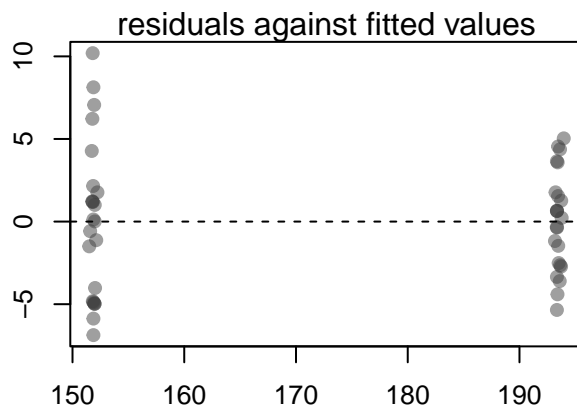
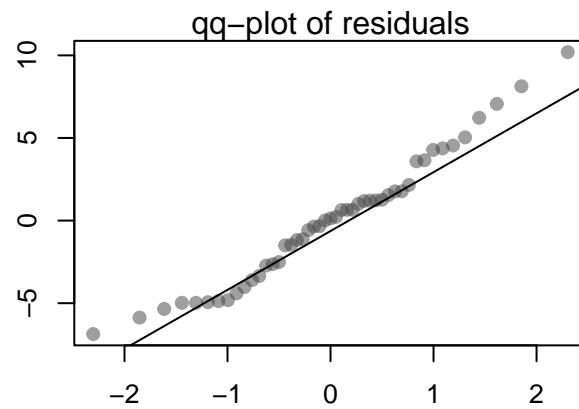
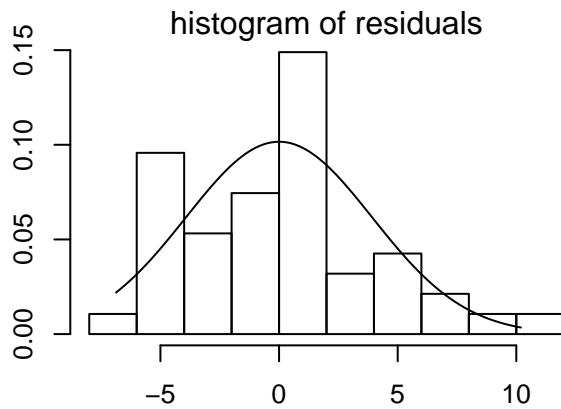
```
## there are higher-order terms (interactions) in this model
```

```
## consider setting type = 'predictor'; see ?vif
```

```
# Looks like there are some VIF issues, especially for tarsus length
```

```
# Fit null model without the interaction between sex and tarsus length
null.intercedens=lm(Wing ~ z.Tarsus + Sex, data=df.intercedens)
```

```
# Diagnostics for the null model
diagnostics.plot(null.intercedens)
```



```
# Compare model fits
round(as.data.frame(anova(null.intercedens, full.intercedens, test="F")), 3)
# it is non-significant (F1, 43 = 15.59, P = 0.331)
# We fit the null model in this case

# Let's have a look at the output of the best model
round(summary(null.intercedens)$coefficients, 3)

# to get confidence intervals use
xx=cbind(orig=coef(null.intercedens), confint(object=null.intercedens))
xx=round(xx, digits=3)
xx

# Stability check for null model
round(my.dfbeta(m=null.intercedens), 3) # Generally looks quite stable

# VIF for null model
round(vif(null.intercedens), 3) # No major VIF issues here

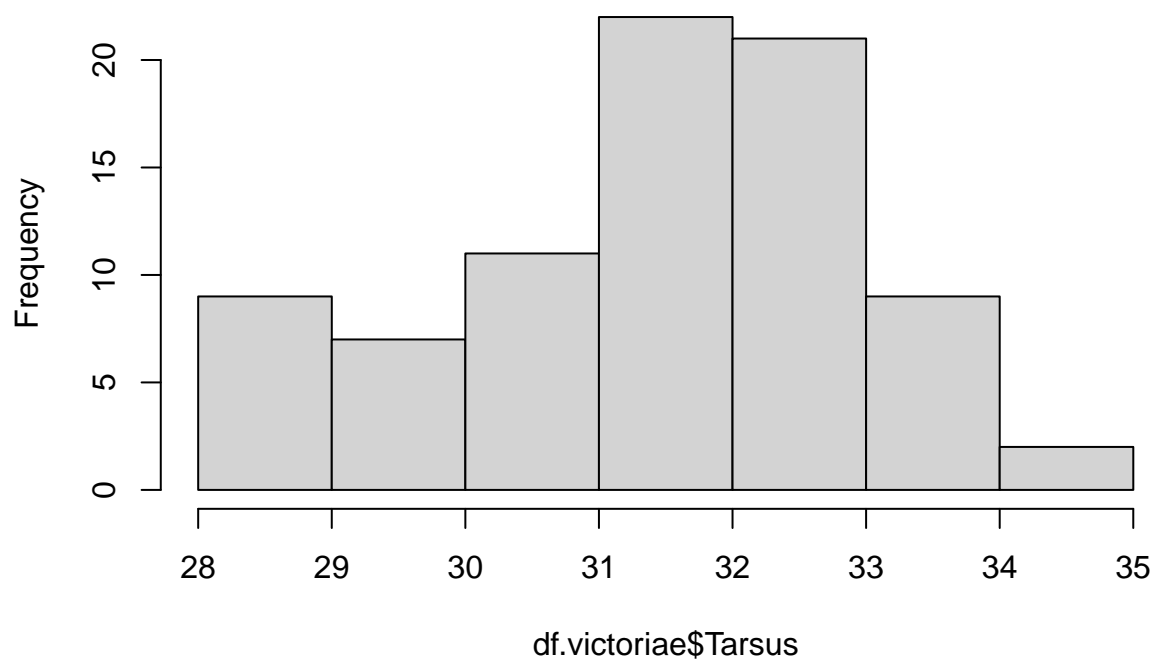
# Check sample sizes
table(df.intercedens$Sex)
```

Victoria's riflebirds

```
df.victoriae <- filter(df, Species == "Ptiloris_victoriae", Age == "Adult")

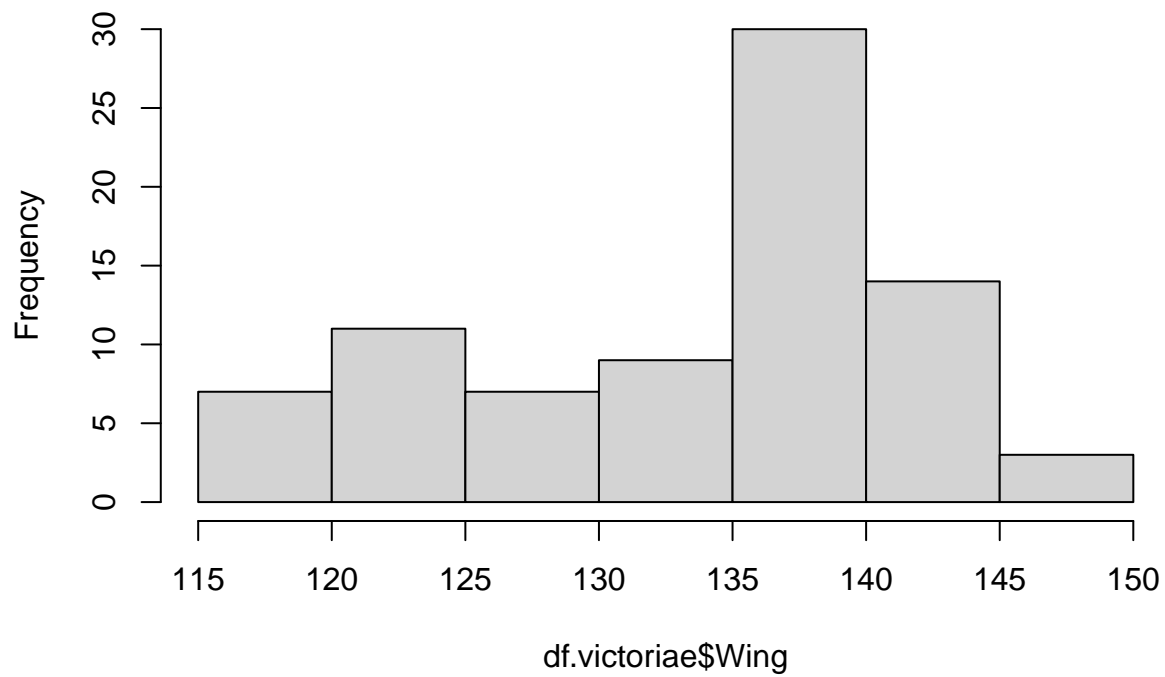
hist(df.victoriae$Tarsus)
```

Histogram of df.victoriae\$Tarsus



```
hist(df.victoriae$Wing)
```

Histogram of df.victoriae\$Wing

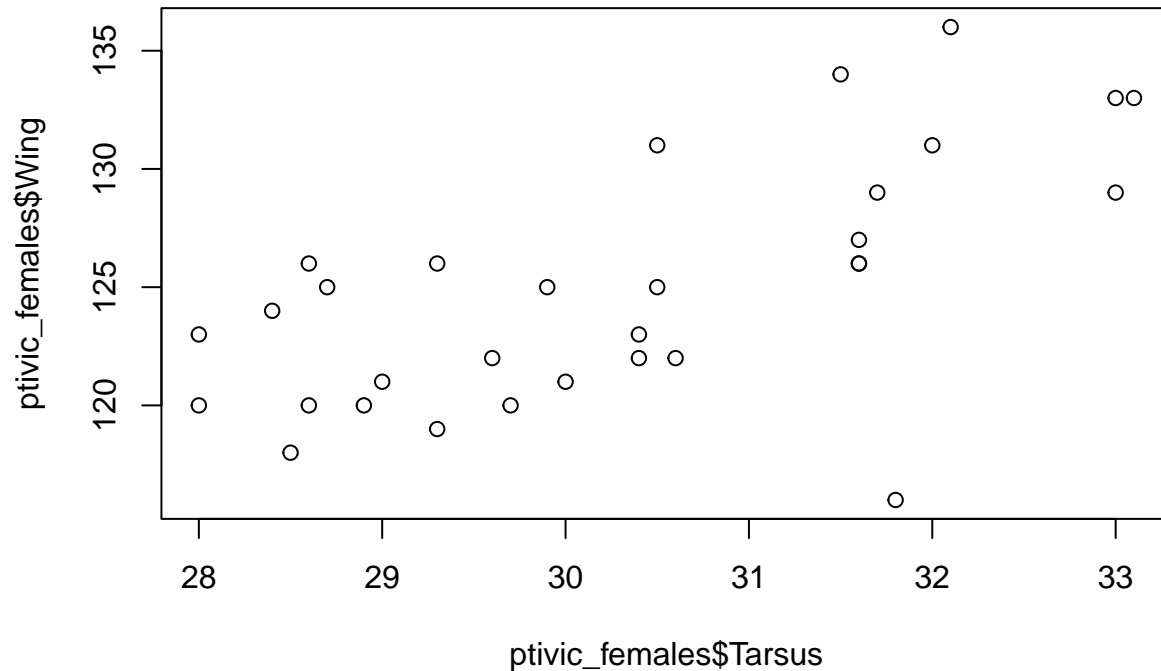


```
# z-transform tarsus length  
df.victoriae$z.Tarsus=as.vector(scale(df.victoriae$Tarsus))
```

```

# Earlier inspection of the data suggests there may be an
# outlier in wing length among the females;
# One individual has a much smaller wing length compared to tarsus length
# First, fit a model with only females:
ptivic_females <- subset(df.victoriae, Sex == "F")
plot(ptivic_females$Tarsus, ptivic_females$Wing)

```



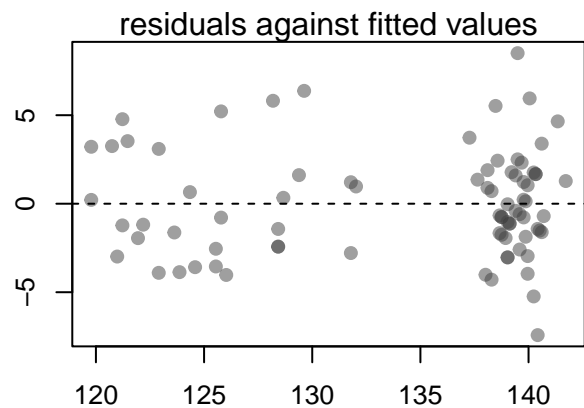
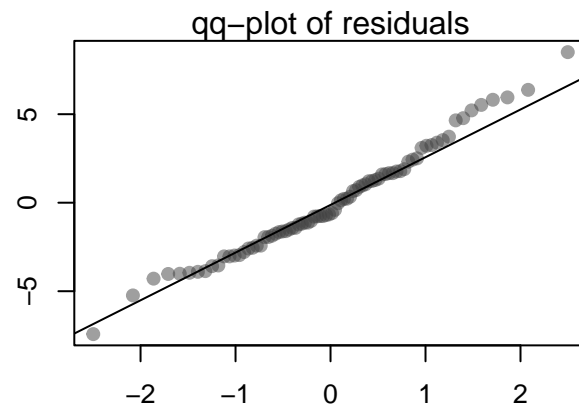
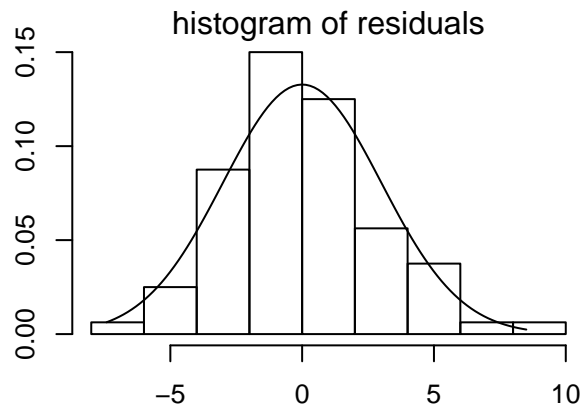
```

F.victoriae=lm(Wing ~ z.Tarsus, data=ptivic_females)
# Check for outliers:
outliers <- check_outliers(F.victoriae, method = c("zscore robust", "mcd"))
which(outliers) # We can remove this data point for modelling, as it's been identified as an outlier (I
head(ptivic_females, 10)
# Let's quickly remove that from the dataset:
df.victoriae <- df.victoriae[df.victoriae$ID != "CSIRO 43891", ]

# Fit full model: Wing ~ z.Tarsus + Sex + z.Tarsus x Sex
full.victoriae=lm(Wing ~ z.Tarsus*Sex, data=df.victoriae)

# Basic diagnostics
diagnostics.plot(full.victoriae)

```



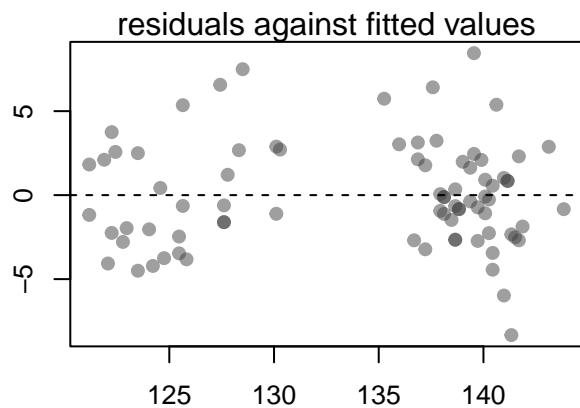
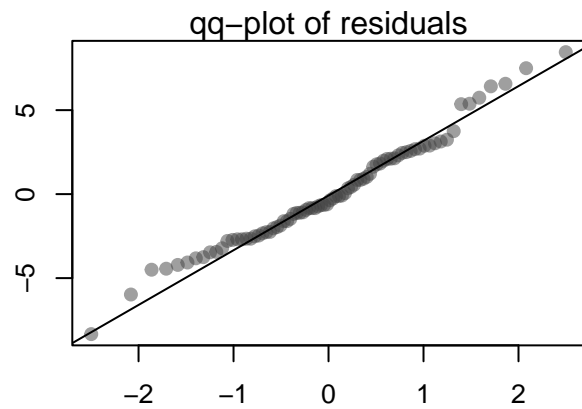
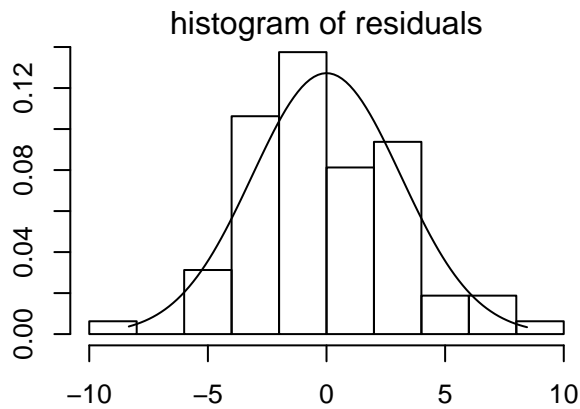
```
# Inspect DF-Beta
round(my.dfbeta(m=full.victoriae), 3) # Generally looks quite stable

# Check for heteroscedasticity by inspecting VIFs
round(vif(full.victoriae), 3) # No obvious issues here

## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

# Fit null model without the interaction between sex and tarsus length
null.victoriae=lm(Wing ~ z.Tarsus + Sex, data=df.victoriae)

# Basic diagnostics
diagnostics.plot(null.victoriae)
```



```
# Compare model fits
round(as.data.frame(anova(null.victoriae, full.victoriae, test="F")), 3)
# it is significant ( $F_{1, 76} = 6.716$ ,  $P = 0.011$ )
# We fit the full model in this case

# Let's have a look at the output of the best model
round(summary(full.victoriae)$coefficients, 3)

# to get confidence intervals use
xx=cbind(orig=coef(full.victoriae), confint(object=full.victoriae))
xx=round(xx, digits=3)
xx

# Stability check for null model
round(my.dfbeta(m=null.victoriae), 3) # Generally looks quite stable

# VIF for null model
round(vif(null.victoriae), 3) # No VIF issues here

# Check sample sizes
table(df.victoriae$Sex)
```

Paradise riflebirds

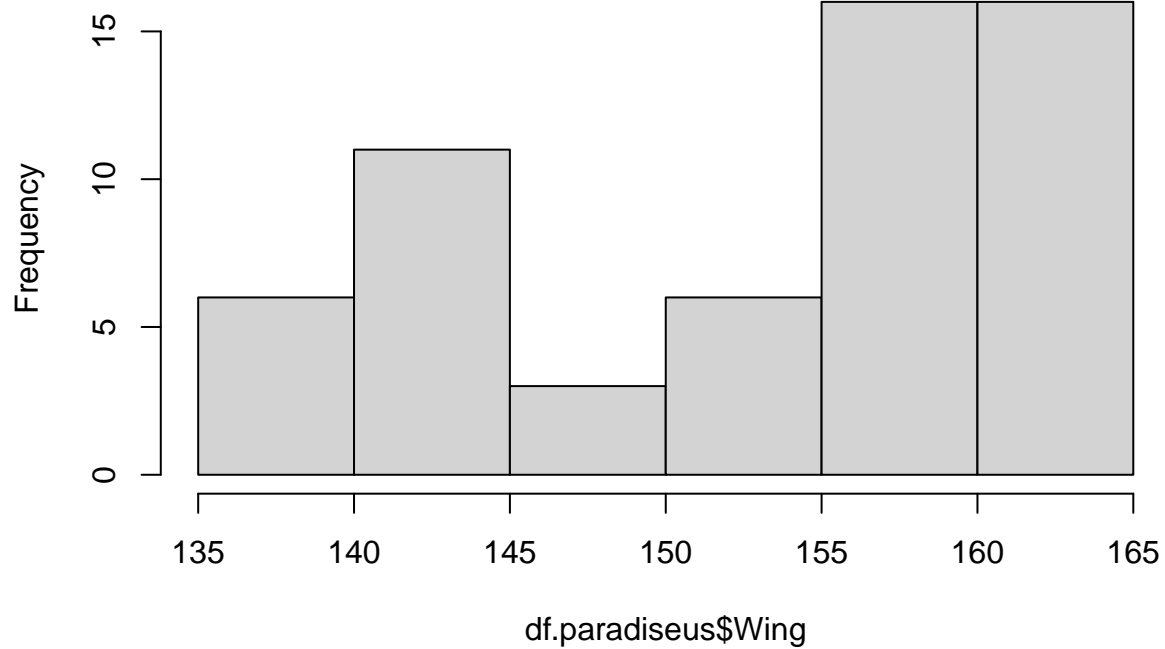
```
df.paradiseus <- filter(df, Species == "Ptiloris_paradiseus", Age == "Adult")
hist(df.paradiseus$Tarsus)
```

Histogram of df.paradiseus\$Tarsus



```
hist(df.paradiseus$Wing)
```

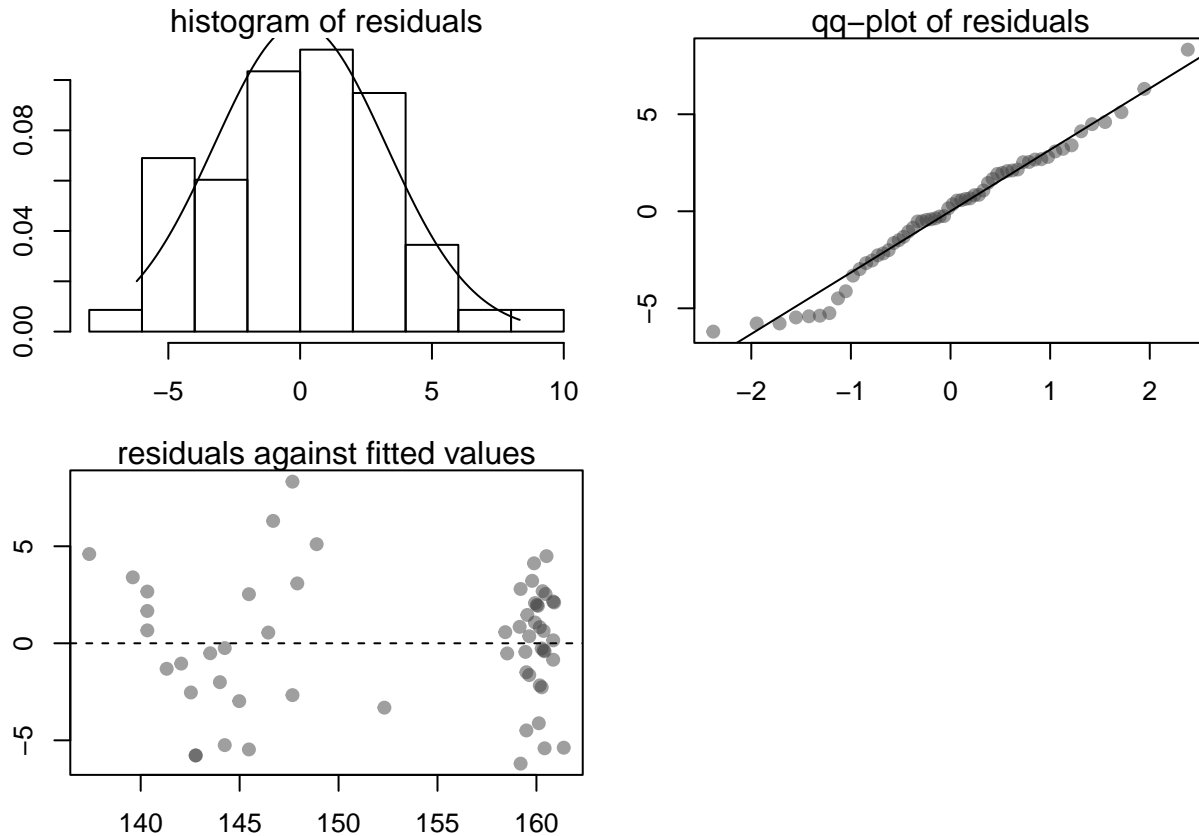
Histogram of df.paradiseus\$Wing



```
# z-transform tarsus length  
df.paradiseus$z.Tarsus=as.vector(scale(df.paradiseus$Tarsus))
```

```
# Fit full model: Wing ~ z.Tarsus + Sex + z.Tarsus x Sex
full.paradiseus=lm(Wing ~ z.Tarsus*Sex, data=df.paradiseus)
```

```
# Basic diagnostics
diagnostics.plot(full.paradiseus)
```



```
# Inspect DF-Beta
round(my.dfbeta(m=full.paradiseus), 3) # Generally looks quite stable
```

```
# Check for heteroscedasticity by inspecting VIFs
round(vif(full.paradiseus), 3) # No obvious issues here
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
# Fit null model without the interaction between sex and tarsus length
null.paradiseus=lm(Wing ~ z.Tarsus + Sex, data=df.paradiseus)
```

```
# Compare model fits
round(as.data.frame(anova(null.paradiseus, full.paradiseus, test="F")), 3)
# it is significant ( $F_{1, 54} = 8.96$ ,  $P = 0.004$ )!
# We fit the full model in this case
```

```
# Let's have a look at the output of the best model
round(summary(full.paradiseus)$coefficients, 3)
```

```
# to get confidence intervals use
xx=cbind(orig=coef(full.paradiseus), confint(object=full.paradiseus))
```



```
xx=round(xx, digits=3)
xx

# Check sample sizes
table(df.paradiseus$Sex)
```

Plotting

```
### Magnificent riflebird ###

# Plotting

# Predict values for males
ptimag.fit_male <- predict(null.magnificus, newdata = filter(df.magnificus, Sex == "M"))

# Create data frame for males with predicted values
df_predicted_male_ptimag <- filter(df.magnificus, Sex == "M") %>%
  mutate(Predicted_Wing_ptimag = ptimag.fit_male)

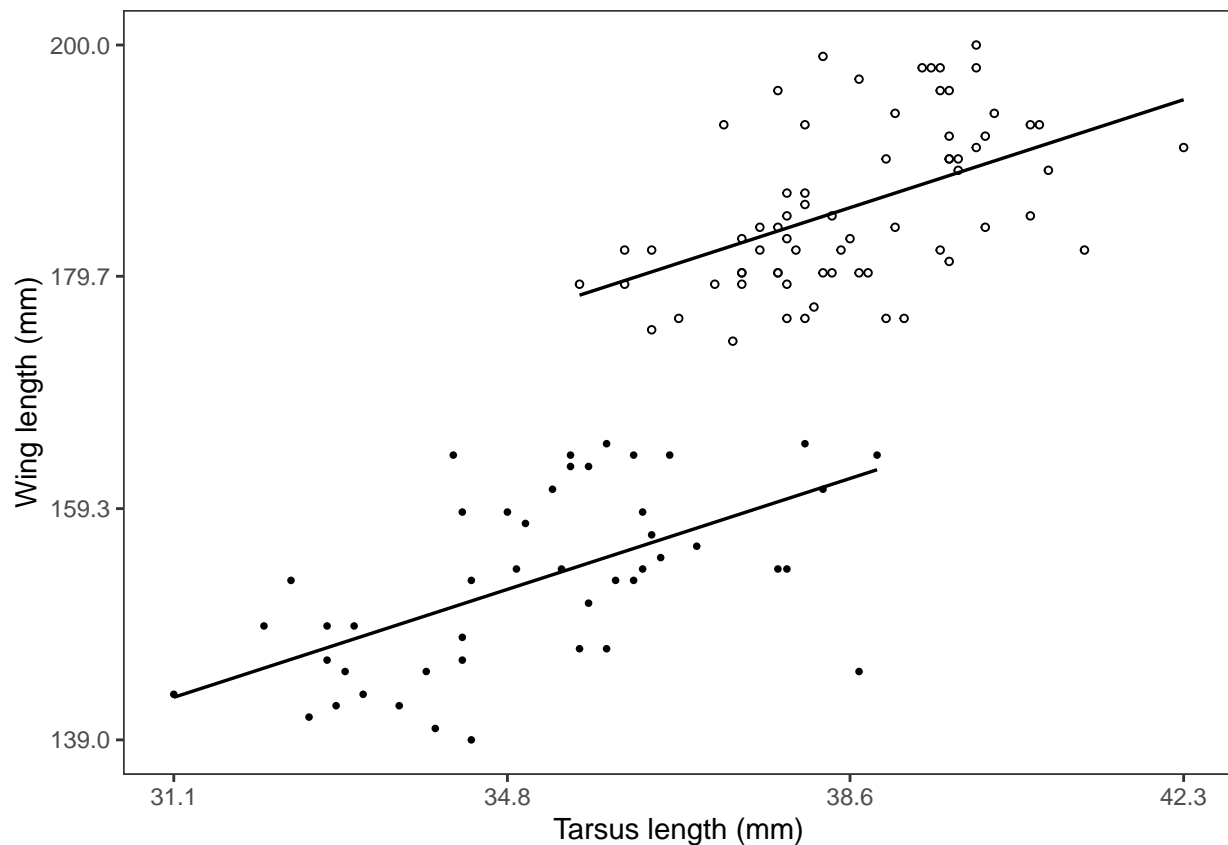
# Predict values for females
ptimag.fit_female <- predict(null.magnificus, newdata = filter(df.magnificus, Sex == "F"))

# Create data frame for females
df_predicted_female_ptimag <- filter(df.magnificus, Sex == "F") %>%
  mutate(Predicted_Wing_ptimag = ptimag.fit_female)

ptimag <- ggplot(df, aes(Tarsus, as.numeric(Wing), shape = Sex)) +
  geom_point(data = df.magnificus, size = 1, alpha = 1) +
  geom_line(data = df_predicted_male_ptimag, aes(y = Predicted_Wing_ptimag),
            color = "black", linetype = 1, linewidth = 0.6) +
  geom_line(data = df_predicted_female_ptimag, aes(y = Predicted_Wing_ptimag),
            color = "black", linetype = 1, linewidth = 0.6) +
  scale_shape_manual(values = c(16, 1)) +
  theme_bw() + ylab("Wing length (mm)") + xlab("Tarsus length (mm)") +
  # ggtitle("Magnificent riflebird") + theme(plot.title=element_text(size=9)) +
  scale_x_continuous(breaks = round(seq(min(df.magnificus$Tarsus), max(df.magnificus$Tarsus), length.out = 10)),
                    labels = round(seq(min(df.magnificus$Tarsus), max(df.magnificus$Tarsus), length.out = 10))) +
  scale_y_continuous(breaks = round(seq(min(df.magnificus$Wing),
                                         max(df.magnificus$Wing), length.out = 4), 1)) +

  guides(shape = FALSE) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

ptimag
```



```
#####

### Growling riflebird ###

# Plotting

# Predict values for males
ptiint.fit_male <- predict(null.intercedens, newdata = filter(df.intercedens, Sex == "M"))

# Create data frame for males with predicted values
df_predicted_male_ptiint <- filter(df.intercedens, Sex == "M") %>%
  mutate(Predicted_Wing_ptiint = ptiint.fit_male)

# Predict values for females
ptiint.fit_female <- predict(null.intercedens, newdata = filter(df.intercedens, Sex == "F"))

# Create data frame for females
df_predicted_female_ptiint <- filter(df.intercedens, Sex == "F") %>%
  mutate(Predicted_Wing_ptiint = ptiint.fit_female)

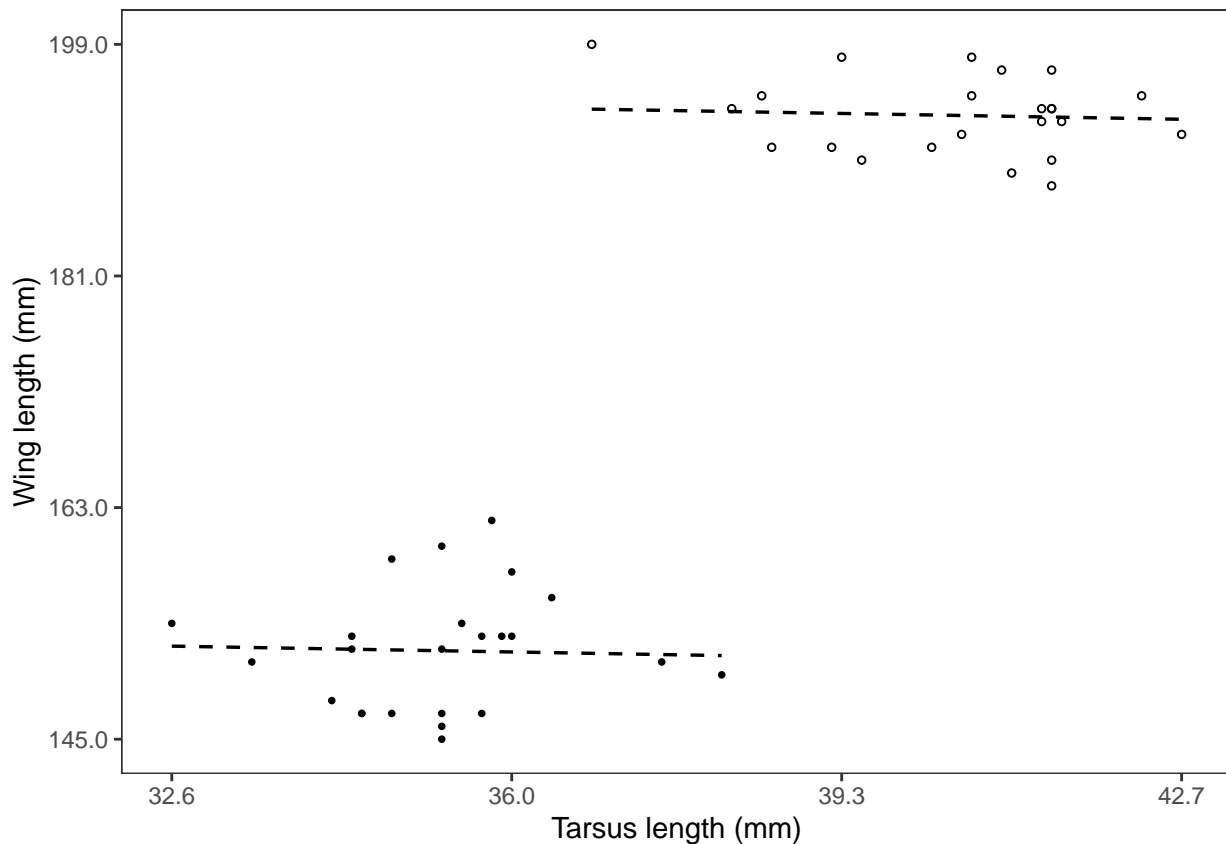
ptiint <- ggplot(df, aes(Tarsus, as.numeric(Wing), shape = Sex)) +
  geom_point(data = df.intercedens, size = 1, alpha = 1) +
  geom_line(data = df_predicted_male_ptiint, aes(y = Predicted_Wing_ptiint),
    color = "black", linetype = 2, linewidth = 0.6) +
  geom_line(data = df_predicted_female_ptiint, aes(y = Predicted_Wing_ptiint),
    color = "black", linetype = 2, linewidth = 0.6) +
  scale_shape_manual(values = c(16, 1)) +
```

```

theme_bw() +
ylab("Wing length (mm)") +
xlab("Tarsus length (mm)") +
scale_x_continuous(breaks = round(seq(min(df.intercedens$Tarsus), max(df.intercedens$Tarsus), length.out = 4), 1),
scale_y_continuous(breaks = round(seq(min(df.intercedens$Wing),
max(df.intercedens$Wing), length.out = 4), 1),
labels = function(x) sprintf("%.1f", x)) + # Format labels to one decimal place
guides(shape = FALSE) +
theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

```

ptiint



```

#####

### Victoria's riflebird ###

# Plotting

# Predict values for males
ptivic.fit_male <- predict(full.victoriae, newdata = filter(df.victoriae, Sex == "M"))

# Create data frame for males with predicted values
df_predicted_male_ptivic <- filter(df.victoriae, Sex == "M") %>%
  mutate(Predicted_Wing_ptivic = ptivic.fit_male)

# Predict values for females
ptivic.fit_female <- predict(full.victoriae, newdata = filter(df.victoriae, Sex == "F"))

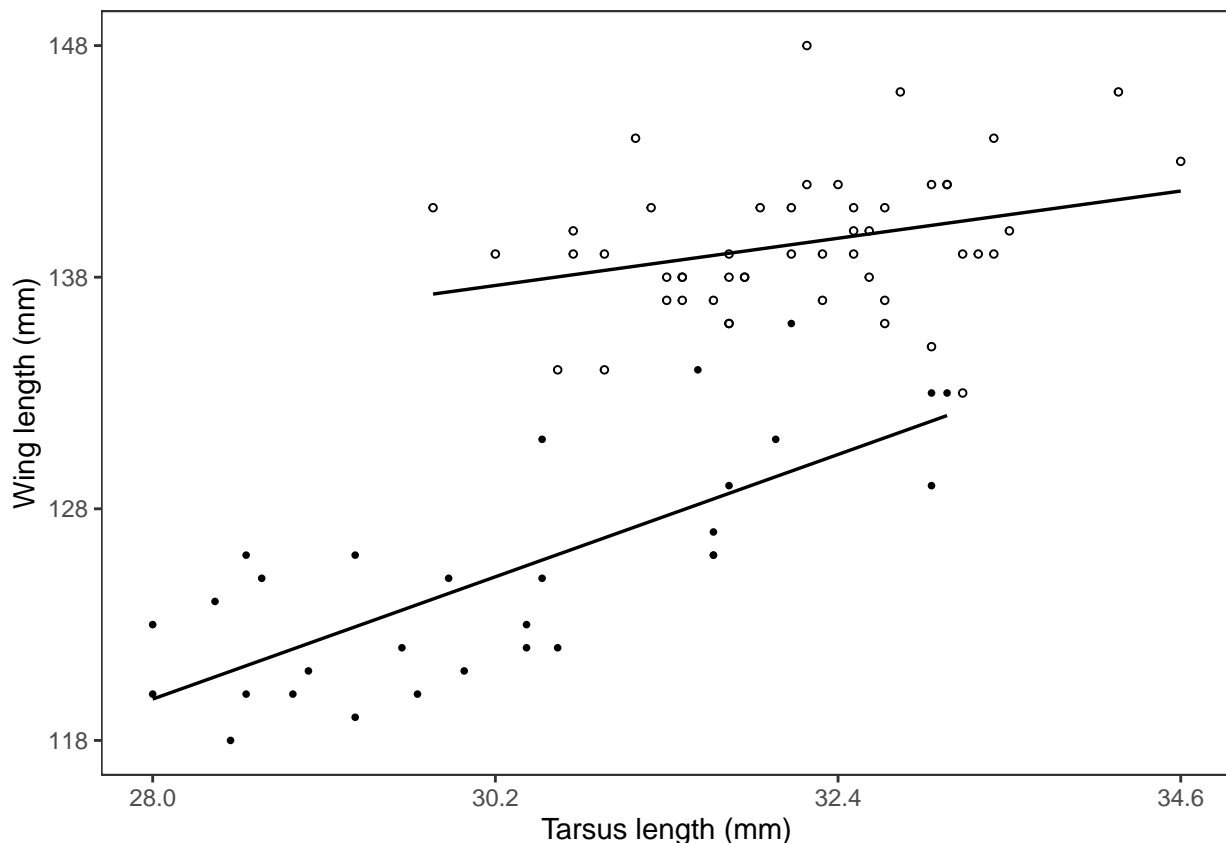
```

```

# Create data frame for females
df_predicted_female_ptivic <- filter(df.victoriae, Sex == "F") %>%
  mutate(Predicted_Wing_ptivic = ptivic.fit_female)

ptivic <- ggplot(df.victoriae, aes(Tarsus, as.numeric(Wing), shape = Sex)) +
  geom_point(data = df.victoriae, size = 1, alpha = 1) +
  geom_line(data = df_predicted_male_ptivic, aes(y = Predicted_Wing_ptivic),
    color = "black", linetype = 1, linewidth = 0.6) +
  geom_line(data = df_predicted_female_ptivic, aes(y = Predicted_Wing_ptivic),
    color = "black", linetype = 1, linewidth = 0.6) +
  scale_shape_manual(values = c(16, 1)) +
  theme_bw() + ylab("Wing length (mm)") + xlab("Tarsus length (mm)") +
  # ggtitle("Victoria's riflebird") + theme(plot.title=element_text(size=9)) +
  scale_x_continuous(breaks = round(seq(min(df.victoriae$Tarsus),
    max(df.victoriae$Tarsus), length.out = 4), 1)) +
  scale_y_continuous(breaks = round(seq(min(df.victoriae$Wing),
    max(df.victoriae$Wing), length.out = 4), 1)) +
  guides(shape = FALSE) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
ptivic

```



```

#####
# Plotting

```

```

# Predict values for males
ptipar.fit_male <- predict(full.paradiseus, newdata = filter(df.paradiseus, Sex == "M"))

# Create data frame for males with predicted values
df_predicted_male_ptipar <- filter(df.paradiseus, Sex == "M") %>%
  mutate(Predicted_Wing_ptipar = ptipar.fit_male)

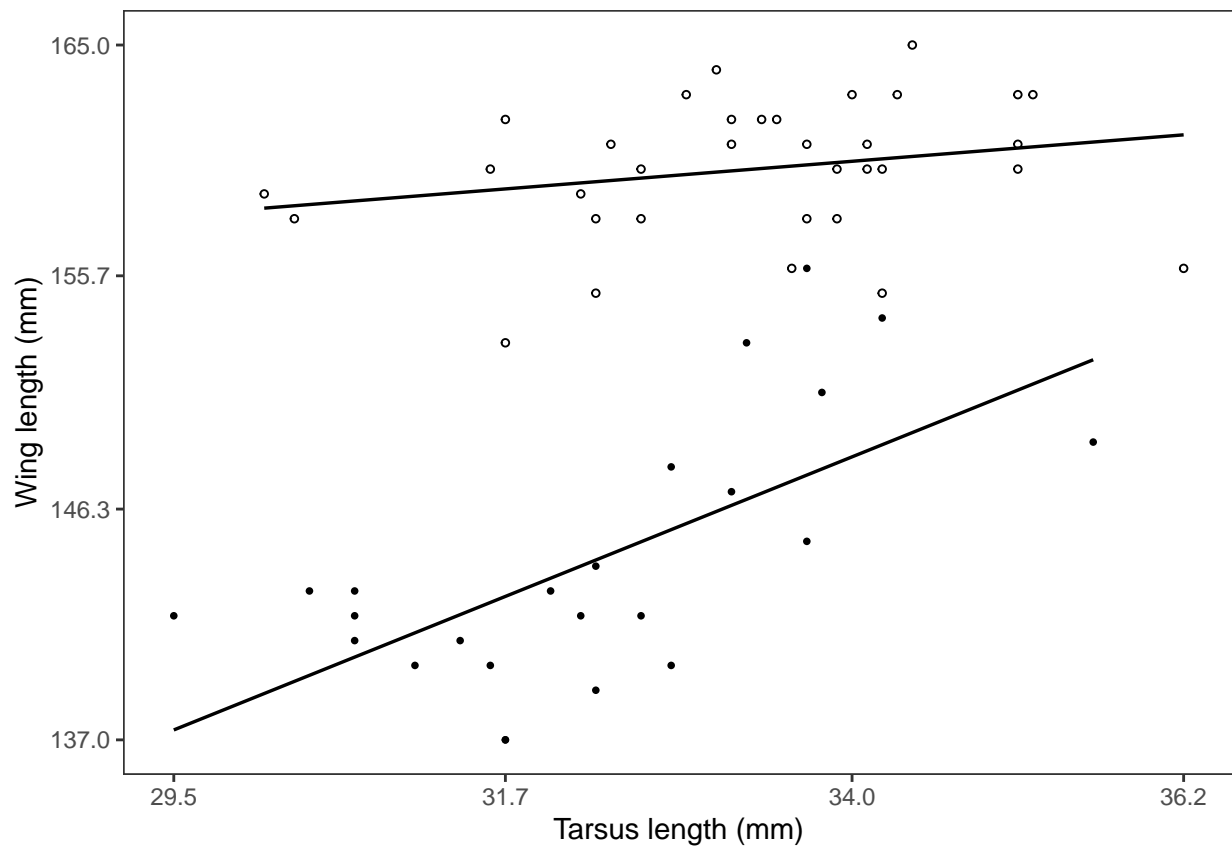
# Predict values for females
ptipar.fit_female <- predict(full.paradiseus, newdata = filter(df.paradiseus, Sex == "F"))

# Create data frame for females
df_predicted_female_ptipar <- filter(df.paradiseus, Sex == "F") %>%
  mutate(Predicted_Wing_ptipar = ptipar.fit_female)

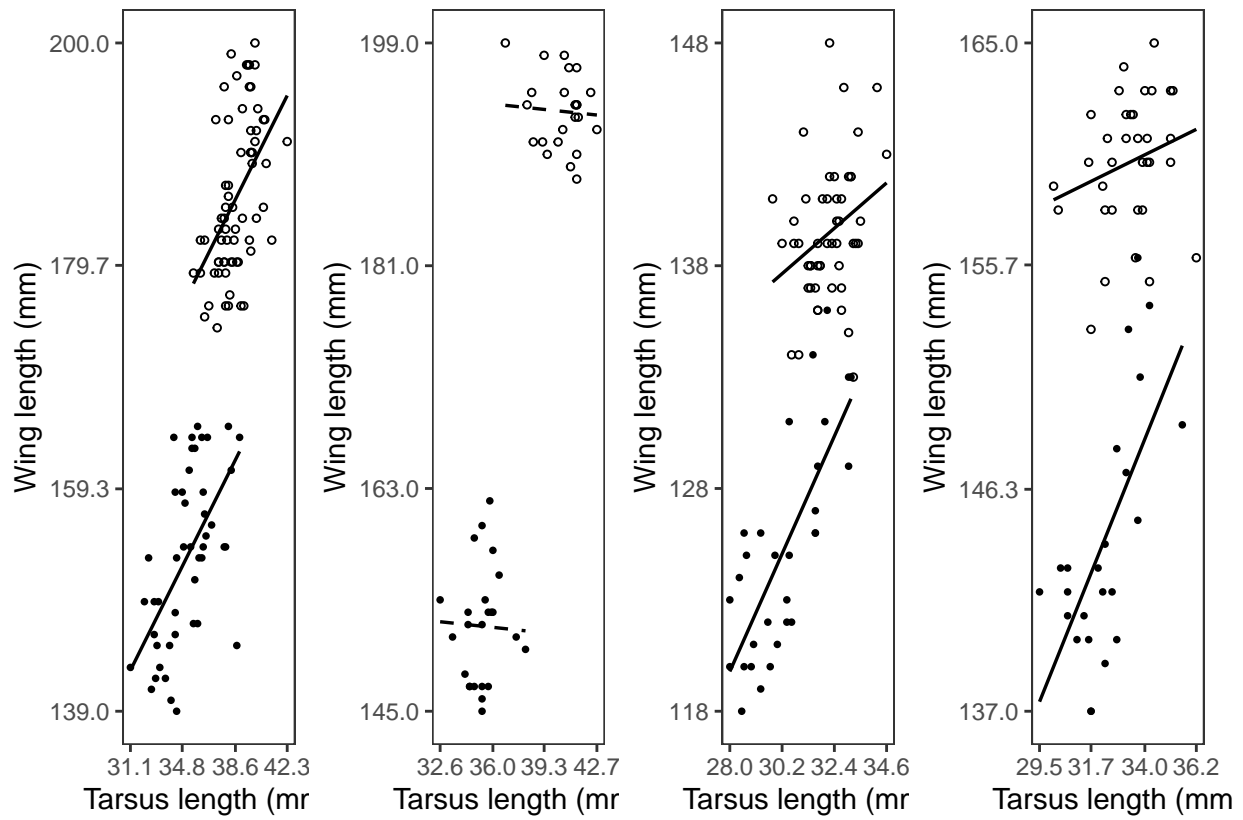
ptipar <- ggplot(df.paradiseus, aes(Tarsus, as.numeric(Wing), shape = Sex)) +
  geom_point(data = df.paradiseus, size = 1, alpha = 1) +
  geom_line(data = df_predicted_male_ptipar, aes(y = Predicted_Wing_ptipar),
    color = "black", linetype = 1, linewidth = 0.6) +
  geom_line(data = df_predicted_female_ptipar, aes(y = Predicted_Wing_ptipar),
    color = "black", linetype = 1, linewidth = 0.6) +
  scale_shape_manual(values = c(16, 1)) +
  theme_bw() + ylab("Wing length (mm)") + xlab("Tarsus length (mm)") +
  # ggtitle("Paradise riflebird") + theme(plot.title=element_text(size=9)) +
  scale_x_continuous(breaks = round(seq(min(df.paradiseus$Tarsus), max(df.paradiseus$Tarsus), length.out = 10), 1)) +
  scale_y_continuous(breaks = round(seq(min(df.paradiseus$Wing),
    max(df.paradiseus$Wing), length.out = 4), 1)) +
  guides(shape = FALSE) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

ptipar

```



```
#####
# Stitch the plots together!
ptimag | ptiint | ptivic | ptipar
```



END OF PART 3

END OF DOCUMENT