UNIVERSITÄT
LEIPZIG

Simple Parametric Tests

Analysing The Sparrow Data Set

-

BASIC STATISTICS FOR BIOLOGISTS

Erik Kusch
Research Assistant
University of Leipzig
Faculty of Life Sciences
Institute of Biology
Behavioral Ecology Research Group
Talstrasse 33
D-04103 Leipzig
Germany
email: erik.kusch@uni-leipzig.de

Summary:

Welcome to our sixth practical experience in R. Throughout the following notes, I will introduce you to a couple of simple parametric test. Whilst parametric tests are used extremely often in biological statistics, they can be somewhat challenging to fit to your data as you will see soon.
To do so, I will enlist the sparrow data set we handled in our first exercise. Additionally, todays seminar is showing plotting via base plot instead of `ggplot2` to highlight the usefulness of base plot and show you the base notation.

Contents

1. Preparing Our Procedure

To ensure others can reproduce our analysis we run the following three lines of code at the beginning of our `R` coding file.

```r
rm(list = ls())  # clearing environment
Dir.Base <- getwd()  # soft-coding our working directory
Dir.Data <- paste(Dir.Base, "Data", sep = "/")  # soft-coding our data directory
```

1.1 Packages

Using the following, user-defined function, we install/load all the necessary packages into our current `R` session.

```r
# function to load packages and install them if they haven't been
# installed yet
install.load.package <- function(x) {
    if (!require(x, character.only = TRUE))
        install.packages(x)
    require(x, character.only = TRUE)
}
package_vec <- c("car")  # needed for the Levene Test for Homogeneity
sapply(package_vec, install.load.package)
```

```
##  car
## TRUE
```

1.2 Loading Data

During our first exercise (Data Mining and Data Handling - Fixing The Sparrow Data Set) we saved our clean data set as an RDS file. To load this, we use the `readRDS()` command that comes with base `R`.

```r
Data_df_base <- readRDS(file = paste(Dir.Data, "/1 - Sparrow_Data_READY.rds",
    sep = ""))
Data_df <- Data_df_base  # duplicate and save initial data on a new object
```

2. t-Test (unpaired)

Assumptions of the unpaired t-Test:

- Predictor variable is binary
- Response variable is metric and **normal distributed** within their groups
- Variable values are **independent** (not paired)

In addition, test whether variance of response variable values in groups are equal (`var.test()`) and adjust `t.test()` argument `var.equal` accordingly.

2.1 Testing For Normality And Homogeneity

We need to test the distribution of our response variables within each predictor variable group for their normality and variance. Since this involves two Shapiro tests and one variance test per variable for each response variable, we might want to write our own function to do so:

```r
ShapiroTest <- function(Variables, Grouping) {
    Output <- data.frame(x = Variables)
    for (i in 1:length(Variables)) {

        X <- Data_df[, Variables[i]]
        Levels <- levels(Data_df[, Grouping])

        Output[i, 2] <- shapiro.test(X[which(Data_df[, Grouping] == Levels[1])])$p.value
        Output[i, 3] <- shapiro.test(X[which(Data_df[, Grouping] == Levels[2])])$p.value
        Output[i, 4] <- var.test(x = X[which(Data_df[, Grouping] == Levels[1])],
            y = X[which(Data_df[, Grouping] == Levels[2])])$p.value
    }
    colnames(Output) <- c("Variable", "P.value1", "P.value2", "Var.Test")
    return(Output)
}
```

This function (`ShapiroTest()`) takes two arguments: (1) `Variables` - a vector of characters holding the names of the variables we want to have tested, and (2) `Grouping` - the binary variable by which to group our variables. The function returns a data frame holding the p-values of the Shapiro tests on each variable group values as well as the `var.test()` p-value.

2.2 Climate Warming/Extremes

Does sparrow morphology change depend on climate?

Using multiple different methods (i.e. Kruskal-Wallis and Mann-Whitney U Test), we have already identified climate (be it in its binary form or when recorded as a three-level variable) is a strong driving force of sparrow morphology. We expect the same results when using a t-Test.

Take note that we need to limit our analysis to our climate type testing sites again as follows (we include Manitoba this time as it is at the same latitude as the UK and Siberia and holds a semi-coastal climate type):

```
# prepare climate type testing data
Data_df <- Data_df_base
Index <- Data_df$Index
Rows <- which(Index == "SI" | Index == "UK" | Index == "RE" | Index ==
    "AU" | Index == "MA")
Data_df <- Data_df[Rows, ]
```

2.2.1 Testing for Normality and Variance

Before we can make use of our data with a t-Test, we need to do an **assumption check**. To this end, we first turn `Climate` records into a binary variable by turning records of a semi-coastal climate into a coastal one.

```
# Make climate binary
Data_df$Climate[which(Data_df$Climate == "Semi-Coastal")] <- "Coastal"
Data_df$Climate <- droplevels(Data_df$Climate)
```

Let's make sure our assumptions are met:

```
ShapiroTest(Variables = c("Weight", "Height", "Wing.Chord"), Grouping = "Climate")
```

```
##     Variable P.value1 P.value2 Var.Test
## 1     Weight    0.170     0.25   0.3262
## 2     Height    0.168     0.36   0.0106
## 3 Wing.Chord    0.054     0.17   0.0029
```

Luckily, all of our variables allow for the calculation of t-Test. Take note though that some need different specification of the `var.equal` argument than others.

2.2.2 Analyses

Sparrow Weight
Let's start with the weight of *Passer domesticus* individuals as grouped by the climate type present at the site weights have been recorded at:

```
t.test(Data_df$Weight ~ Data_df$Climate, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  Data_df$Weight by Data_df$Climate
## t = -10, df = 400, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.4 -1.9
## sample estimates:
##     mean in group Coastal mean in group Continental
##                        31                        33
```

According to our analysis, which has us **reject the null hypothesis**, we conclude that binary climate records are valuable information criteria for predicting sparrow weight with sparrows in coastal climates being lighter than sparrows in continental ones thus effectively varifying the results of our non-parametric approaches (Kruskal-Wallis, Mann-Whitney U).

Sparrow Height

Let's move on to the height of *Passer domesticus* individuals as grouped by the climate type present at the site weights have been recorded at:

```
t.test(Data_df$Height ~ Data_df$Climate, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  Data_df$Height by Data_df$Climate
## t = -0.3, df = 400, p-value = 0.8
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.23  0.18
## sample estimates:
##     mean in group Coastal mean in group Continental
##                        14                        14
```

Confirming the results of our Mann-Whitney U Test, we **accept the null hypothesis**.

Sparrow Wing Chord

Lastly, we test the wing chords of *Passer domesticus* individuals as grouped by the climate type present at the site weights have been recorded at:

```
t.test(Data_df$Wing.Chord ~ Data_df$Climate, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  Data_df$Wing.Chord by Data_df$Climate
## t = -0.1, df = 400, p-value = 0.9
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.040  0.035
## sample estimates:
##     mean in group Coastal mean in group Continental
##                       6.9                       6.9
```

Without confirming the results of our Mann-Whitney U Test, we **accept the null hypothesis**.

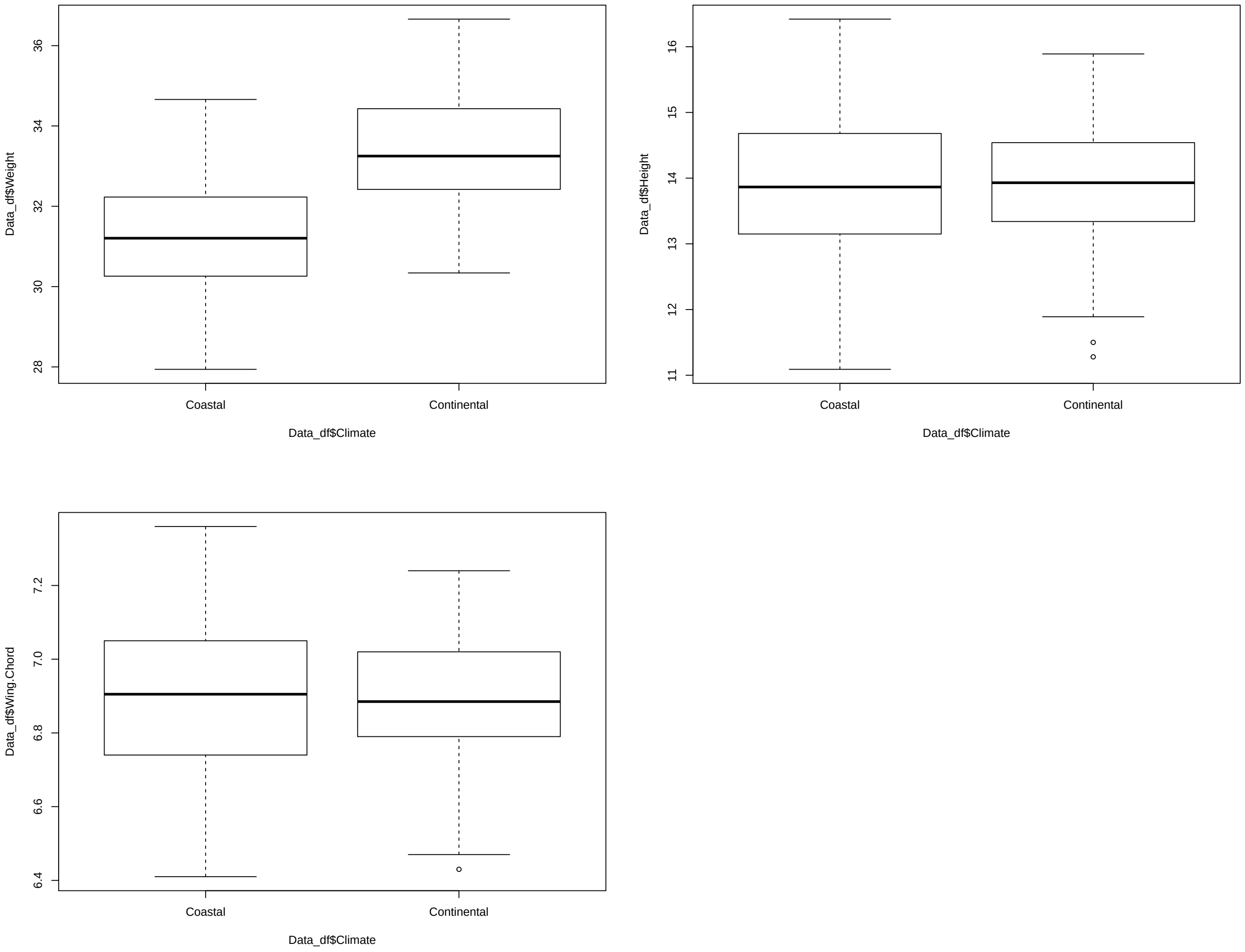
Conclusion

Here's what we've learned from the t-Test so far:

- Sparrow weight depends on (binary) climate types
- Sparrow height does not depend on (binary) climate types
- Sparrow wing chord does not depend on (binary) climate types

Let's end this by viusalising all of the data:

```r
par(mfrow = c(2, 2))
plot(Data_df$Weight ~ Data_df$Climate)
plot(Data_df$Height ~ Data_df$Climate)
plot(Data_df$Wing.Chord ~ Data_df$Climate)
```



2.3 Sexual Dimorphism

Does sparrow morphology change depend on Sex?

Using the Mann-Whitney U Test, we have already identified the sex of *Passer domesticus* is a good information criterion for understanding sparrow weight but not sparrow height or wing chord. Let's see if we can reproduce this using a t-Test approach.

We may wish to use the entirety of our data set again for this purpose:

```
Data_df <- Data_df_base
```

2.3.1 Testing for Normality and Variance

Again, before we can use our data in a t-Test for this purpose, we have to make sure that our assumptions are met. To this end, we can make use of our user defined `ShapiroTest()` function as follows:

```
ShapiroTest(Variables = c("Weight", "Height", "Wing.Chord"), Grouping = "Sex")
```

```
##     Variable P.value1 P.value2 Var.Test
## 1     Weight  2.9e-21  1.7e-21     0.75
## 2     Height  4.0e-17  6.6e-19     0.40
## 3 Wing.Chord  3.4e-25  1.6e-26     0.56
```

As it turns out, our data does not allow for any t-Test (this happens often in real studies). However, we can create sex-driven subgroups within each site and test whether these meet the requirements for our t-Test. In order to do so, we need to do some minor tweaking to our `ShapiroTest()` function:

```
ShapiroTestSites <- function(Variables, Grouping) {
    list <- list()
    for (k in 1:length(unique(Data_df$Index))) {
        Output <- data.frame(x = Variables)
        Data <- Data_df[which(Data_df$Index == unique(Data_df$Index)[k]),
            ]
        for (i in 1:length(Variables)) {

            X <- Data[, Variables[i]]
            Levels <- levels(Data[, Grouping])

            Output[i, 2] <- shapiro.test(X[which(Data[, Grouping] == Levels[1])])$p.value
            Output[i, 3] <- shapiro.test(X[which(Data[, Grouping] == Levels[2])])$p.value
            Output[i, 4] <- var.test(x = X[which(Data_df[, Grouping] ==
                Levels[1])], y = X[which(Data_df[, Grouping] == Levels[2])])
        }
        colnames(Output) <- c("Variable", "P.value1", "P.value2", "Var.Test")
        list[[k]] <- Output
    }
    return(list)
}
```

This function (`ShapiroTestSites()`) takes two arguments: (1) `Variables` - a vector of characters holding the names of the variables we want to have tested, and (2) `Grouping` - the binary variable by which to group our variables. The function returns a list of data frames for each site holding the p-values of the Shapiro tests on each variable group values as well as the `var.test()` p-value.

Let's put this function to the test:

```
ShapiroTestSites(Variables = c("Weight", "Height", "Wing.Chord"), Grouping = "Sex")
```

```
## [[1]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight     0.29    0.051     0.91
## 2     Height     0.29    0.047     0.91
## 3 Wing.Chord     0.26    0.046     0.90
##
## [[2]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight     0.40     0.32      1.3
## 2     Height     0.41     0.32      1.1
## 3 Wing.Chord     0.40     0.32      1.2
##
## [[3]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight     0.56     0.27     1.12
## 2     Height     0.56     0.19     1.00
## 3 Wing.Chord     0.51     0.19     0.99
##
## [[4]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight     0.34     0.80     1.13
## 2     Height     0.34     0.79     0.84
## 3 Wing.Chord     0.34     0.77     0.85
##
## [[5]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight     0.60     0.47     0.93
## 2     Height     0.59     0.47     1.08
## 3 Wing.Chord     0.58     0.45     1.08
##
## [[6]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight    0.037     0.41     0.93
## 2     Height    0.036     0.41     0.67
## 3 Wing.Chord    0.033     0.40     0.68
##
## [[7]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight     0.20     0.74     1.02
## 2     Height     0.20     0.74     0.69
## 3 Wing.Chord     0.21     0.69     0.69
##
## [[8]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight     0.88     0.16     0.86
## 2     Height     0.89     0.16     0.92
## 3 Wing.Chord     0.89     0.14     0.92
##
## [[9]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight     0.96     0.61      1.2
## 2     Height     0.96     0.61      1.0
## 3 Wing.Chord     0.92     0.57      1.0
##
```

```
## [[10]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight    0.011     0.45     0.66
## 2     Height    0.011     0.46     0.63
## 3 Wing.Chord    0.011     0.49     0.62
##
## [[11]]
##     Variable P.value1 P.value2 Var.Test
## 1     Weight     0.60     0.17     0.95
## 2     Height     0.61     0.17     0.76
## 3 Wing.Chord     0.62     0.17     0.75
```

With the exception for sparrow morphology records at:
- Siberia (SI, height and wing chord of males)
- Manitoba (MA, morphology of females)
- South Africa (SA, morphology of females)
all of our data groups variables are normal distributed with equal variances between the groups per site.

Since our problematic sites are still relatively close to fulfilling our requirements of the data, we will use them going forward as if they did.

2.3.2 Analyses

Running three t-Tests (Weight, Height, Wing Chord) for each of our eleven sites is absolute mania! Therefore, we write our own function again that let's us apply the tests exactly the way we want to:

```
t_testSite <- function(Variables, Grouping, data, VarEqual) {
    Data <- data
    Index <- unique(Data$Index)
    Indexes <- Data$Index
    list <- list()
    for (i in 1:length(Variables)) {
        Output <- data.frame(NA)
        for (k in 1:length(Index)) {
            # data and test
            X <- Data[, Variables[i]][which(Indexes == Index[k])]
            Y <- Data[, Grouping][which(Indexes == Index[k])]
            Test <- t.test(X ~ Y, paired = FALSE, var.equal = VarEqual)
            # filling data frame
            Output[1, k] <- Test[["p.value"]]
            Output[2, k] <- Test[["estimate"]][[1]]
            Output[3, k] <- Test[["estimate"]][[2]]
        }
        # data frame to list
        colnames(Output) <- Index
        rownames(Output) <- c("p", "Mean1", "Mean2")
        list[[i]] <- Output
    }
    return(list)
}
```

This `t_testSite()` function takes four arguments: (1) `Variables` - a vector of characters holding the names of the variables we want to have tested, (2) `Grouping` - the binary variable by which to group our variables, (3) `data` - the data frame which contains the `Variables` and the `Grouping` factor, and (4) `VarEqual` - a logical indicator of whether to perform a t-Test assuming equal variance of the groups or not.
The function returns a list of data frames (one per variable) containing the p-values of the unpaired t-Tests for each variable at every site as well as the predicted group means.

Although our function `t_testSite()` can handle multiple variables at once, we will now use it on each of our morphological sparrow variables individually to disentangle them a bit easier:

Does sparrow weight depend on sex when assessed at each of our sites individually?

```
t_testSite(Variables = "Weight", Grouping = "Sex", data = Data_df, VarEqual = TRUE)
```

```
## [[1]]
##            SI      UK      AU      RE      NU      MA      LO      BE      FG      SA
## p     1.5e-09 1.9e-05 1.6e-12 9.7e-11 2.6e-10 5.0e-10 2.3e-09 4.7e-10 6.1e-27 1.0e-11
## Mean1 3.3e+01 3.1e+01 3.2e+01 3.0e+01 3.1e+01 3.1e+01 3.1e+01 2.9e+01 2.2e+01 2.9e+01
## Mean2 3.5e+01 3.3e+01 3.4e+01 3.1e+01 3.3e+01 3.3e+01 3.2e+01 3.1e+01 2.3e+01 3.0e+01
##            FI
## p     3.1e-10
## Mean1 3.1e+01
## Mean2 3.2e+01
```

As it turns out, sex is a statistically significant predictor for sparrow weight at each site. This was to be expected. Using a binomial test in our second practical, we identified no bias in sexes for our sparrow populations. In addition, using the Mann-Whitney U-Test in our fourth practical, we identified sex to be an important information criterion for sparrow weight across all of our sites. Given these two conditions, we were expecting a results like the one presented here with males being, on average, heavier than females in *Passer domesticus* and we **reject the null hypothesis**.

Does sparrow height depend on sex when assessed at each of our sites individually?

```
t_testSite(Variables = "Height", Grouping = "Sex", data = Data_df, VarEqual = TRUE)
```

```
## [[1]]
##         SI    UK    AU    RE    NU    MA    LO   BE    FG   SA    FI
## p      0.9  0.54  0.67  0.49  0.14  0.19  0.92  0.9  0.98  0.4  0.35
## Mean1 13.6 13.67 14.14 14.49 13.30 13.38 14.15 14.5 18.90 14.9 13.25
## Mean2 13.6 13.49 14.22 14.36 13.68 13.71 14.13 14.5 18.90 14.7 13.46
```

Like with our Mann-Whitney U-Test, we fail to identify a significant effect of sex on sparrow height records at each of our sites and so we **accept the null hypothesis**.

Does sparrow wing chord depend on sex when assessed at each of our sites individually?

```
t_testSite(Variables = "Wing.Chord", Grouping = "Sex", data = Data_df,
    VarEqual = TRUE)
```

```
## [[1]]
##         SI   UK   AU   RE   NU   MA   LO   BE   FG   SA   FI
## p     0.88 0.55 0.67 0.47 0.15 0.19 0.94 0.88 0.97 0.41 0.37
## Mean1 6.78 6.80 6.98 7.07 6.72 6.75 6.94 7.13 8.59 7.24 6.74
## Mean2 6.79 6.77 6.99 7.05 6.78 6.80 6.94 7.12 8.59 7.21 6.77
```

Like with our Mann-Whitney U-Test, we fail to identify a significant effect of sex on sparrow wing chord records at each of our sites and so we **accept the null hypothesis**.

3. t-Test (paired)

Assumptions of the paired t-Test:

- Predictor variable is binary
- Response variable is metric
- *Difference of response variable pairs* is **normal distributed**
- Variable values are **dependent** (paired)

3.1 Preparing Data

For this purpose, we need an **additional data set with truly paired records** of sparrows and so we implement the same solution as we've used within our fourth seminar using the Wilcoxon Signed Rank Test. Within our study set-up, think of a **resettling experiment**, were you take *Passer domesticus* individuals from one site, transfer them to another and check back with them after some time has passed to see whether some of their characteristics have changed in their expression.
To this end, presume we have taken the entire *Passer domesticus* population found at our **Siberian** research station and moved them to the **United Kingdom**. Whilst this keeps the latitude stable, the sparrows *now experience a coastal climate instead of a continental one.* After some time (let's say: a year), we have come back and recorded all the characteristics for the same individuals again.

You will find the corresponding *new data* in `2 - Sparrow_Resettled_READY.rds`. Take note that this set only contains records for the transferred individuals in the **same order** as in the old data set.

```
Data_df_Resettled <- readRDS(file = paste(Dir.Data, "/2 - Sparrow_Resettled_READY.rds",
    sep = ""))
```

Since earlier analysis such as the Wilcoxon Signed Rank test (fourth practical) and the Friedman Test (fifth practical) showed that height and wing chord records do not change when sparrows are resettled at all, we have excluded these here and **focus solely on sparrow weight**.

3.2 Testing for Normality

Before being able to run our paired t-Test, we must make sure that the *difference of response variable pairs* is **normal distributed**. We can do so using the `shapiro.test()` of base `R` as follows:

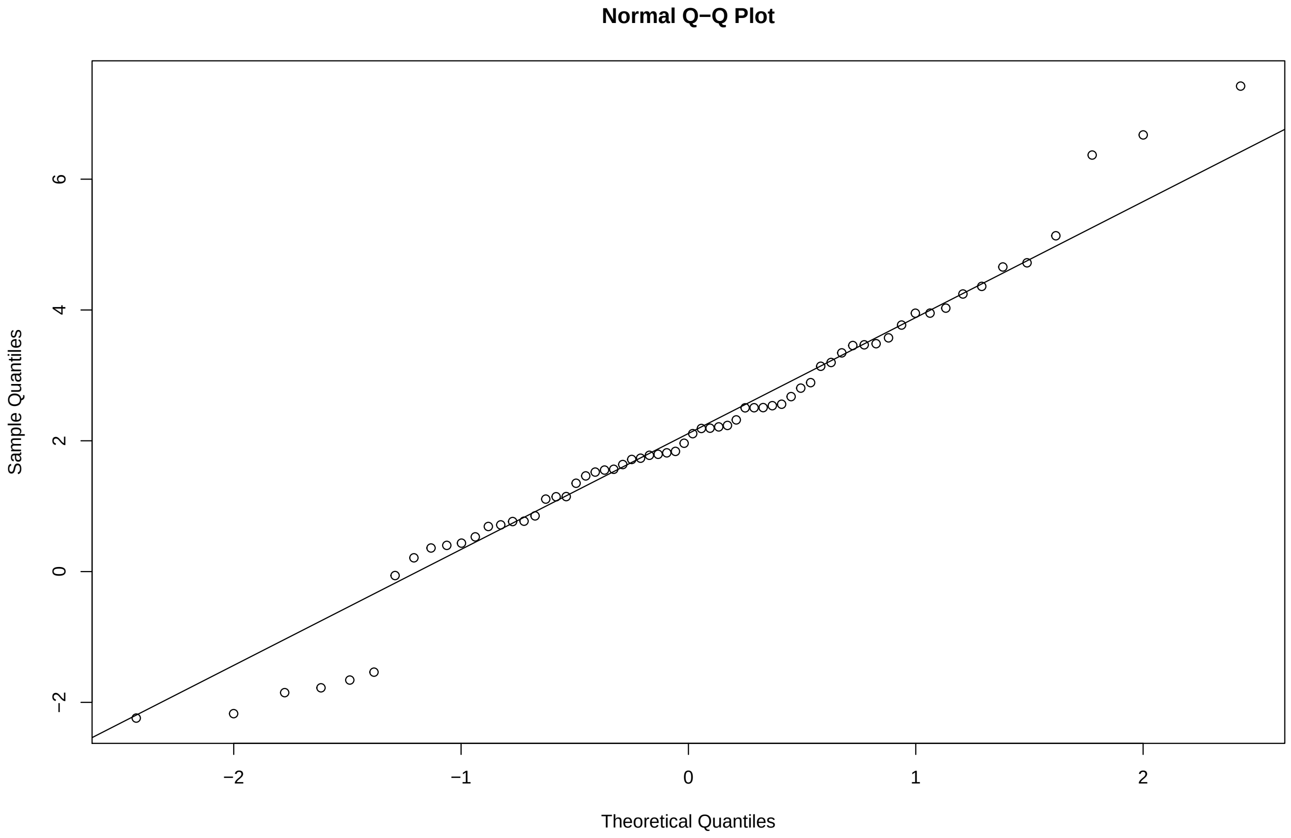
```
# selecting pre-resettling weights
DataSI <- Data_df$Weight[which(Data_df$Index == "SI")]
# calculating difference of before and after resettling weights
WeightDiff <- DataSI - Data_df_Resettled$Weight
# shapiro test
shapiro.test(WeightDiff)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  WeightDiff
## W = 1, p-value = 0.2
```

Thankfully, the **assumption of normality** is **met**.

Now let's visualise that using a qqplot:

```
qqnorm(WeightDiff)
qqline(WeightDiff)
```



3.3 Climate Warming/Extremes

Does sparrow morphology change depend on climate?

Now let's go on to test whether sparrow weights change significantly per individual due to our relocation experiment (we expect this from future test in our practicals):

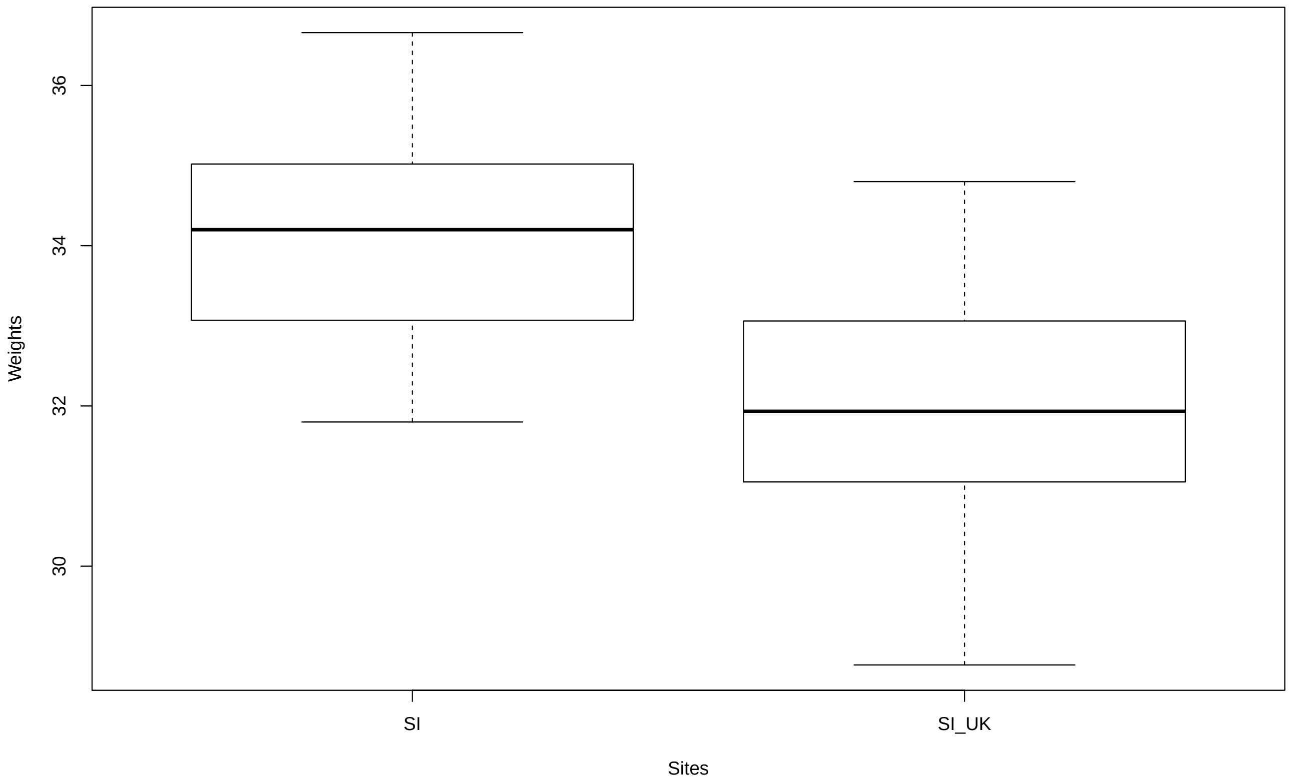
```
t.test(DataSI, Data_df_Resettled$Weight, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  DataSI and Data_df_Resettled$Weight
## t = 8, df = 60, p-value = 4e-12
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.6 2.6
## sample estimates:
## mean of the differences
##                     2.1
```

We were right, individual sparrow weights change significantly after our relocation experiment and we **reject the null hypothesis**. This is in accordance with the results of the Wilcoxon Signed Rank Test as well as the Friedman Test.

Let's go on to visualise our data to make better sense of what is going on here:

```
# Select the sparrow weights
Weights <- c(DataSI, Data_df_Resettled$Weight)
# Select the sites
Sites <- factor(rep(c("SI", "SI_UK"), each = length(DataSI)))
# Plot
plot(Weights ~ Sites)
```



Quite obviously sparrows observed in Siberia are heavier than when they are resettled to the United Kingdom (this may be due to the more forgiving climate in the UK). Just like the test stated, the difference of the average weights is roughly 2g between the sparrows at the two sites.

4. One-Way ANOVA

Assumptions of the One-Way ANOVA:

- Predictor variable is categorical
- Response variable is metric
- *Response variable residuals* are **normal distributed**
- Variance of populations/samples are equal (**homogeneity**)
- Variable values are **independent** (not paired)

4.1 Testing For Assumptions

Firstly, we need to test the assumptions of our One-Way ANOVA. For this purpose, we write another user-defined function.

```r
# User-defined function
ANOVACheck <- function(Variables, Grouping, data, plotting) {
    Output <- data.frame(x = NA)
    for (i in 1:length(Variables)) {
        # data
        Y <- as.numeric(data[, Variables[i]])
        X <- data[, Grouping]
        Levels <- levels(data[, Grouping])
        # Residuals?
        model <- lm(Y ~ X)
        Output[1, i] <- shapiro.test(residuals(model))$p.value
        # Homgeneity?
        Levene <- leveneTest(Y ~ X, center = median, data = data)
        Output[2, i] <- Levene[1, 3]
        # Plotting
        if (plotting == TRUE) {
            plot(model, 2)  # Normality
            plot(model, 3)  # Homogeneity
        }
    }
    colnames(Output) <- Variables
    rownames(Output) <- c("Residual Normality", "Homogeneity of Variances")
    return(Output)
}
```

This `ANOVACheck()` function takes four arguments: (1) `Variables` - a vector of characters holding the names of the variables we want to have tested, (2) `Grouping` - the categorical variable by which to group our variables, (3) `data` - the data frame which contains the `Variables` and the `Grouping` factor, and (4) `plotting` - a logical indicator of whether to produce plots visualising the test results or not.
The function returns a data frames containing the p-values indexing whether to accept or reject the notion of the normality of residuals per variable (`Residual Normality`), and the p-values indexing whether variances between groups are homogeneous or not (`Homogeneity of Variances`).

4.2 Climate Warming/Extremes

Does sparrow morphology change depend on climate?

Using the Kruskal-Wallis Test in our last exercise, we already identified climate to be an important factor in determining *Passer domesticus* morphology. Let's see if this holds true.

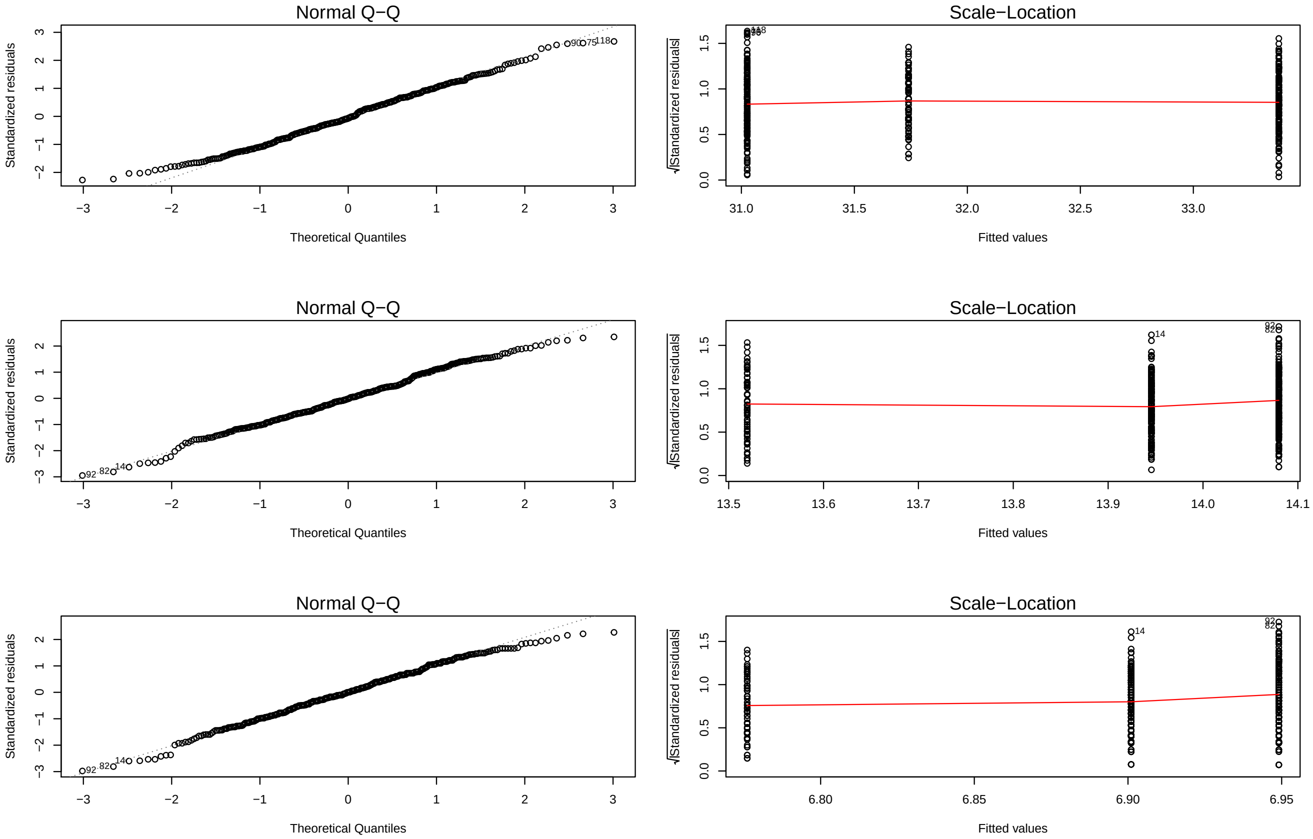
Take note that we need to limit our analysis to our climate type testing sites again as follows (we include Manitoba this time as it is at the same latitude as the UK and Siberia and holds a semi-coastal climate type):

```
# prepare climate type testing data
Data_df <- Data_df_base
Index <- Data_df$Index
Rows <- which(Index == "SI" | Index == "UK" | Index == "RE" | Index ==
    "AU" | Index == "MA")
Data_df <- Data_df[Rows, ]
```

4.2.1 Assumption Check

Let's use the `ANOVACheck()` function on our data:

```
par(mfrow = c(3, 2))
ANOVACheck(Variables = c("Weight", "Height", "Wing.Chord"), Grouping = "Climate",
    data = Data_df, plotting = TRUE)
```



```
##                           Weight Height Wing.Chord
## Residual Normality         0.045  0.099     0.0516
## Homogeneity of Variances   0.961  0.091     0.0064
```

Unfortunately, neither weight nor wing chord records fullfil our requirements.

4.2.2 Analysis

Let's run our analysis for height as grouped by the three-level climate variable:

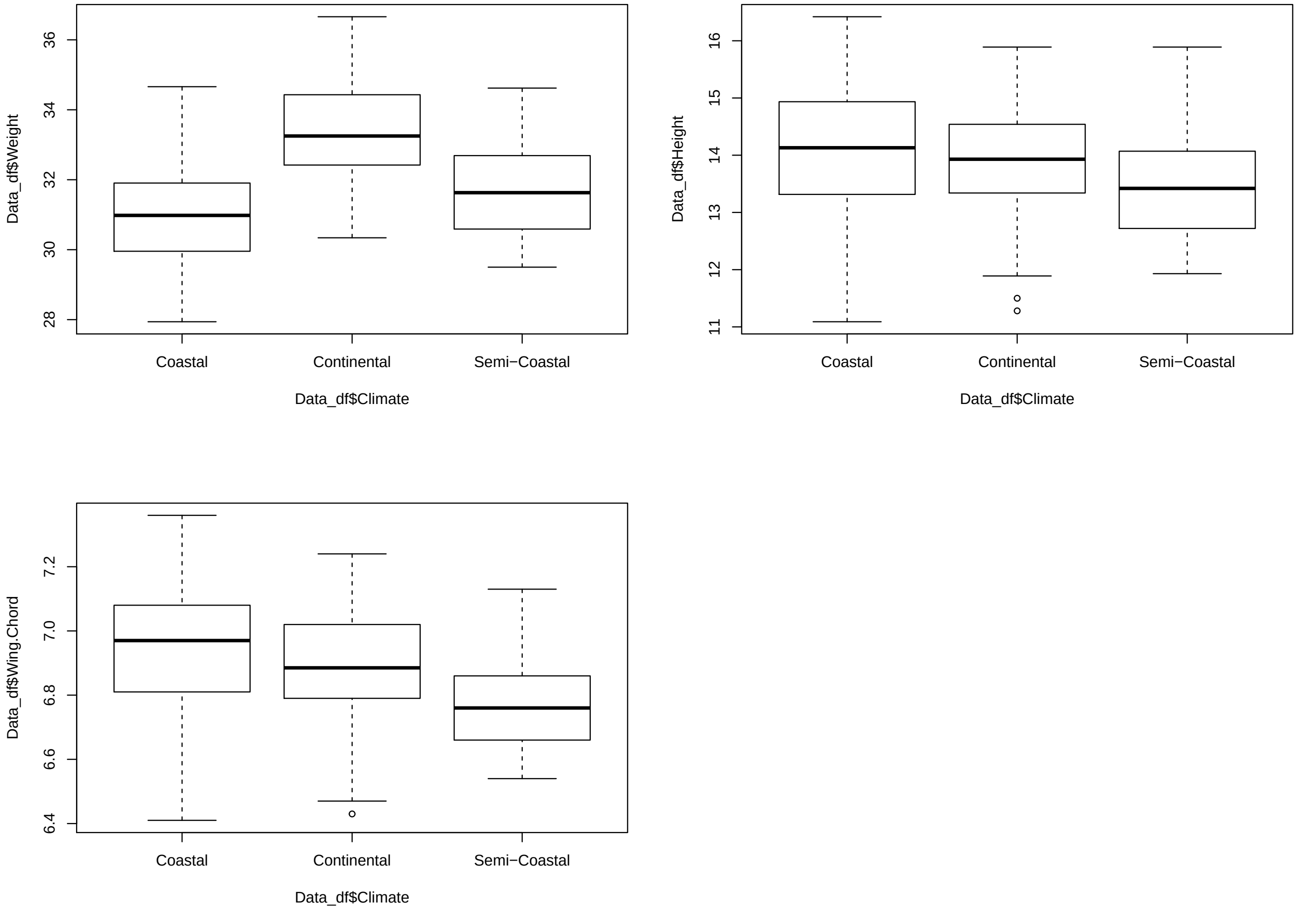
```
model <- lm(Data_df$Height ~ Data_df$Climate)
anova(model)
```

```
## Analysis of Variance Table
##
## Response: Data_df$Height
##                  Df Sum Sq Mean Sq F value  Pr(>F)
## Data_df$Climate   2     15    7.49    7.25 0.00081 ***
## Residuals       381    394    1.03
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

According to this, climate is a meaningful predictor of height of sparrows and we **reject the null hypothesis** thus confirming the results of our Kruskall-Wallis analysis.

Let's end this by plotting all of our data:

```
par(mfrow = c(2, 2))
plot(Data_df$Weight ~ Data_df$Climate)
plot(Data_df$Height ~ Data_df$Climate)
plot(Data_df$Wing.Chord ~ Data_df$Climate)
```



As you can see, the variances are definitely not equal between our groups which explains why part of our assumption test failed.

4.3 Predation

Does nesting height depend on predator characteristics?

Again, using the Kruskal-Wallis Test in our last exercise, we already identified predator characteristics to be an important factor in determining *Passer domesticus* nesting height. Let's see if this holds true.

We may wish to use the entirety of our data set again for this purpose:

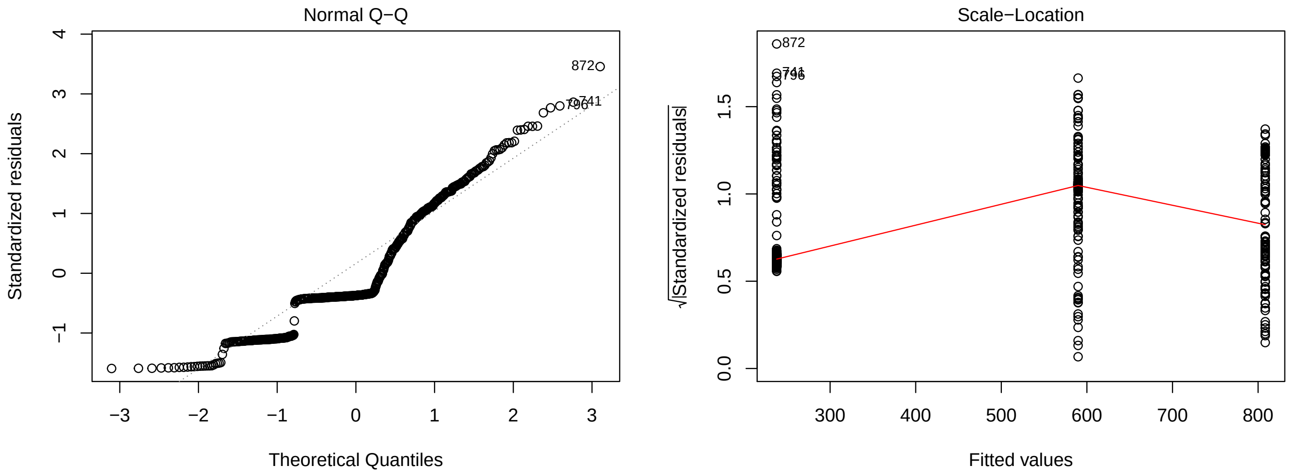
```
Data_df <- Data_df_base
```

4.3.1 Assumption Check

Let's use our ANOVACeck() function to test whether we can run our analysis. Before we can do so, however, we need to slightly adjust our predator type variable just like we did in our last exercise and as follows:

```
# changing levels in predator type
levels(Data_df$Predator.Type) <- c(levels(Data_df$Predator.Type), "None")
Data_df$Predator.Type[which(is.na(Data_df$Predator.Type))] <- "None"

# Assumption Check
par(mfrow = c(1, 2))
ANOVACheck(Variables = "Nesting.Height", Grouping = "Predator.Type", data = Data_df,
    plotting = TRUE)
```



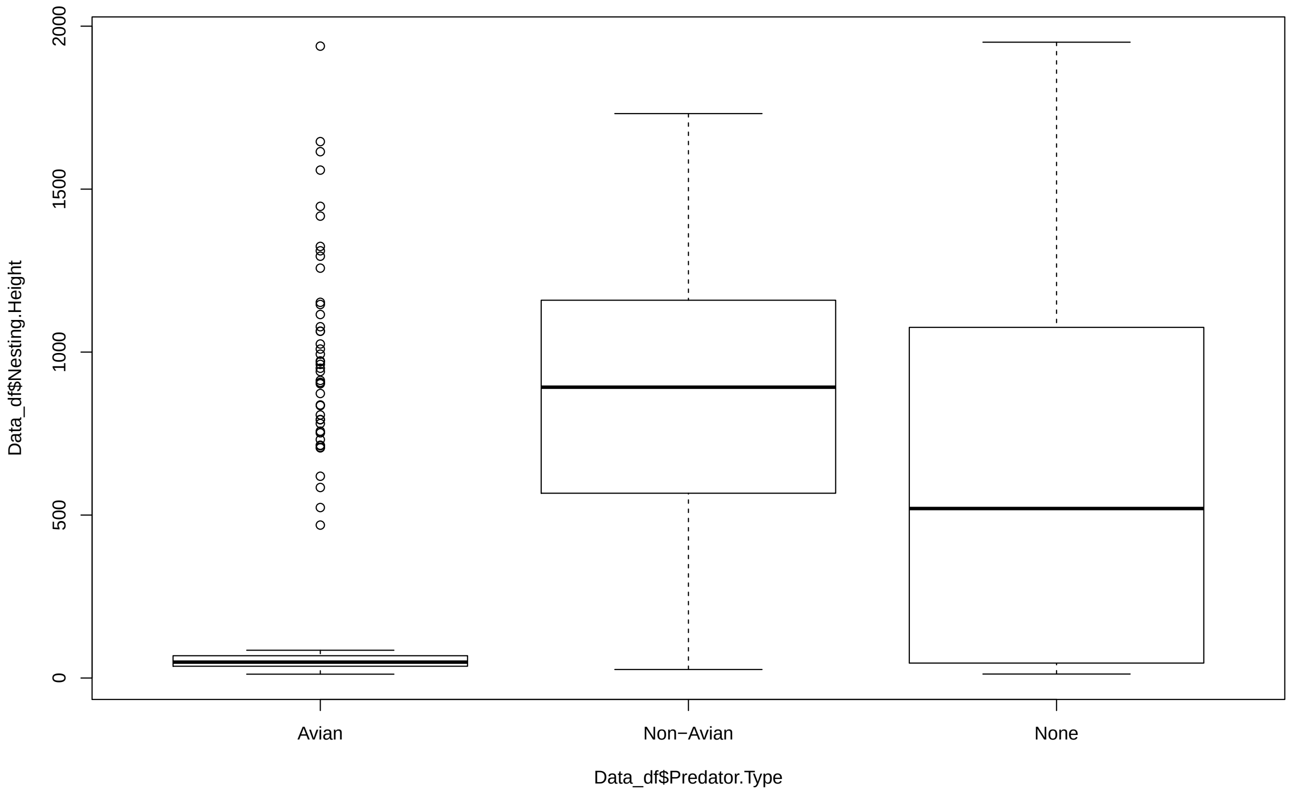
```
##                           Nesting.Height
## Residual Normality               1.2e-15
## Homogeneity of Variances         2.5e-20
```

Again, our data fails the assumption check. The residuals are definitely not normal distributed and the variance of nesting height records within our groups are not equal.

4.3.2 Analysis

Since none of our assumptions are met, we cannot run an ANOVA and therefore resort to data visualisation alone:

```
plot(Data_df$Nesting.Height ~ Data_df$Predator.Type)
```



Once more, we can see why our homogeneity of variances test failed.

4.4 Site-wise Variation

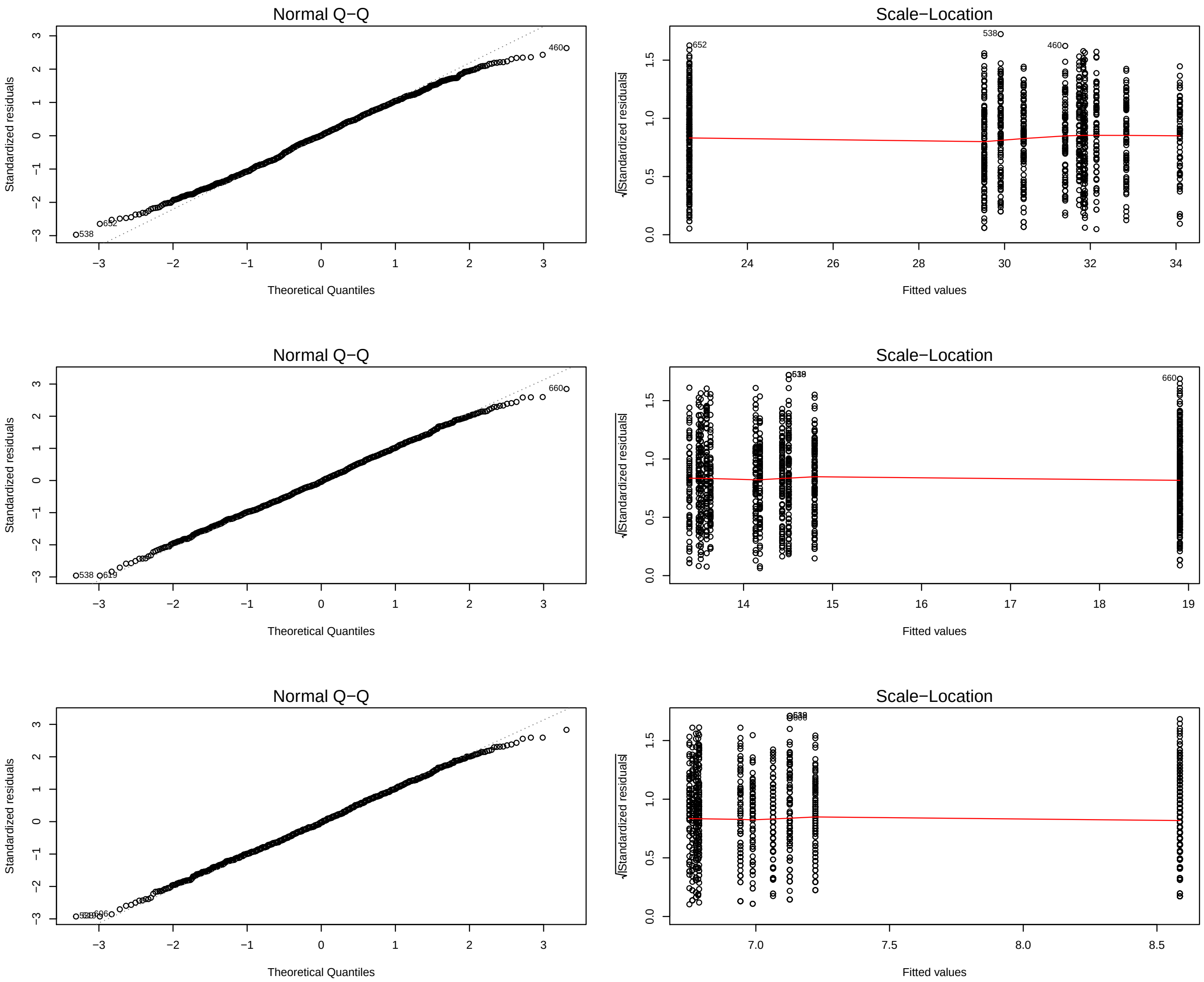
Does sparrow morphology depend on sites?

Again, using the Kruskal-Wallis Test in our last exercise, we already identified the site index to be an important factor in determining *Passer domesticus* morphology. Let's see if this holds true.

4.4.1 Assumption Check

Let's use our `ANOVACeck()` function to test whether we can run our analysis:

```
par(mfrow = c(3, 2))
ANOVACheck(Variables = c("Weight", "Height", "Wing.Chord"), Grouping = "Index",
    data = Data_df, plotting = TRUE)
```



```
##                          Weight Height Wing.Chord
## Residual Normality       0.0069   0.38       0.34
## Homogeneity of Variances 0.3856   0.26       0.25
```

This time, our data does not fail the assumption test (except for the weight attribute of our sparrows).

4.4.2 Analysis

Let's run only one One-Way ANOVA then, because we have already looked at this with our Kruskal-Wallis Test and site-wise variation wasn't really part of our big research questions (seminar 4);

```
LM_fit1 <- lm(Height ~ Index, data = Data_df)
anova(LM_fit1)
```

```
## Analysis of Variance Table
##
## Response: Height
##             Df Sum Sq Mean Sq F value Pr(>F)
## Index       10   4675     468     489 <2e-16 ***
## Residuals 1056   1009       1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(LM_fit1)
```

```
##
## Call:
## lm(formula = Height ~ Index, data = Data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8773 -0.6731 -0.0363  0.6919  2.7775
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.1839     0.1042  136.14  < 2e-16 ***
## IndexBE       0.3235     0.1413    2.29  0.02222 *
## IndexFG       4.7187     0.1211   38.95  < 2e-16 ***
## IndexFI      -0.7926     0.1572   -5.04  5.4e-07 ***
## IndexLO      -0.0476     0.1505   -0.32  0.75201
## IndexMA      -0.6643     0.1585   -4.19  3.0e-05 ***
## IndexNU      -0.6873     0.1606   -4.28  2.0e-05 ***
## IndexRE       0.2498     0.1446    1.73  0.08435 .
## IndexSA       0.6150     0.1387    4.43  1.0e-05 ***
## IndexSI      -0.5558     0.1591   -3.49  0.00050 ***
## IndexUK      -0.5981     0.1578   -3.79  0.00016 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.98 on 1056 degrees of freedom
## Multiple R-squared:  0.823,  Adjusted R-squared:  0.821
## F-statistic:  489 on 10 and 1056 DF,  p-value: <2e-16
```

The `anova()` command tells us that the influence of `Index` on sparrow `Height` records is statistical significant. The output yielded by the `summary()` command of our model object lets us interpret the effect of the site a sparrow is located at on its predicted height record. These effects are also referred to as **coefficients**.

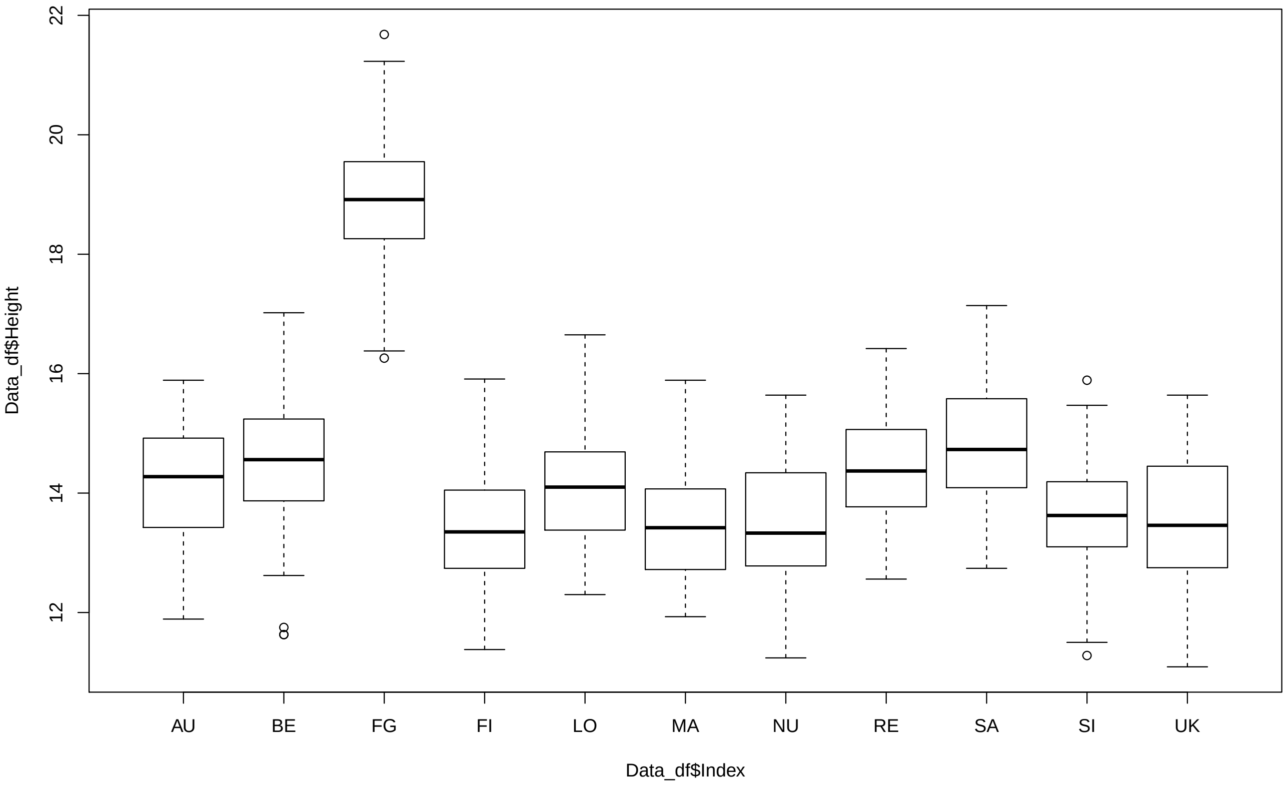
Let's interpret these:

- The mean sparrow height in Australia (AU) is 14.18cm (this is our **Intercept**/*Baseline*) - The mean sparrow height in Belize (BE) is 0.32cm bigger than the **Intercept**
- The mean sparrow height in Belize (BE) is 0.32cm bigger than the **Intercept**
- The mean sparrow height in French Guiana (FG) is 4.72cm bigger than the **Intercept**
- The mean sparrow height in the Falkland Isles (FI) is -0.79cm bigger than the **Intercept**
- The mean sparrow height in Louisiana (LO) is -0.05cm bigger than the **Intercept** (although this is not statistically significant)
- The mean sparrow height in Manitoba (MA) is -0.66cm bigger than the **Intercept**

- The mean sparrow height in Nunavut (NU) is -0.69cm bigger than the **Intercept**
- The mean sparrow height in the Reunion Island is 0.25cm bigger than the **Intercept** (although this is not statistically significant)
- The mean sparrow height in South America (SA) is 0.61cm bigger than the **Intercept**
- The mean sparrow height in Siberia (SI) is -0.56cm bigger than the **Intercept**
- The mean sparrow height in the United Kingdom (UK) is -0.56cm bigger than the **Intercept**

These relations can be visualised as follows:

```
plot(Data_df$Height ~ Data_df$Index)
```



5. Two-Way ANOVA

Assumptions of the Two-Way ANOVA:

- Predictor variables are categorical
- Response variable is metric
- *Response variable residuals* are **normal distributed**
- Variance of populations/samples are equal (**homogeneity**)
- Variable values are **independent** (not paired)

5.1 Testing For Assumptions

Yet again, we need to check if our assumptions are met first. Automating this procedure is definitely a good idea and only needs slight modification from our `ANOVACheck()` function.

```r
# User-defined function
ANOVACheck_TWO <- function(Formulas, data, plotting) {
    Output <- data.frame(x = NA)
    for (i in 1:length(Formulas)) {
        # Check how many formulas there are
        if (length(Formulas) == 1) {
            Expression <- Formulas[[1]]
        } else {
            Expression <- Formulas[[i]]
        }
        # Residuals?
        model <- lm(formula = Expression, data = data)
        Output[1, i] <- shapiro.test(residuals(model))$p.value
        # Homgeneity?
        Levene <- leveneTest(Expression, center = median, data = data)
        Output[2, i] <- Levene[1, 3]
        # Plotting
        if (plotting == TRUE) {
            plot(model, 2)  # Normality
            plot(model, 3)  # Homogeneity
        }
    }
    colnames(Output) <- as.character(Formulas)
    rownames(Output) <- c("RN", "HoV")
    return(Output)
}
```

This `ANOVACheck_TWO()` function takes four arguments: (1) `Formulas` - a vector of formula specification for our ANOVA models we want to have tested, (2) `data` - the data frame which contains the variables and the grouping factor called upon in our `Formulas`, and (3) `plotting` - a logical indicator of whether to produce plots visualising the test results or not.
The function returns a data frames containing the p-values indexing whether to accept or reject the notion of the normality of residuals per variable (`RN`), and the p-values indexing whether variances between groups are homogeneous or not (`HoV`).

5.2 Climate Warming/Extremes

Does sparrow morphology depend on climate and sex?

Now this is a novel research question to which we have no answer yet. Whilst we have shed light on whether sparrow morphology depends of sex (only weight does) and climate (it does) respectively, we lack knowledge on whether these two factors interact when determining sparrow morphology. We can use a two-way ANOVA to check this.

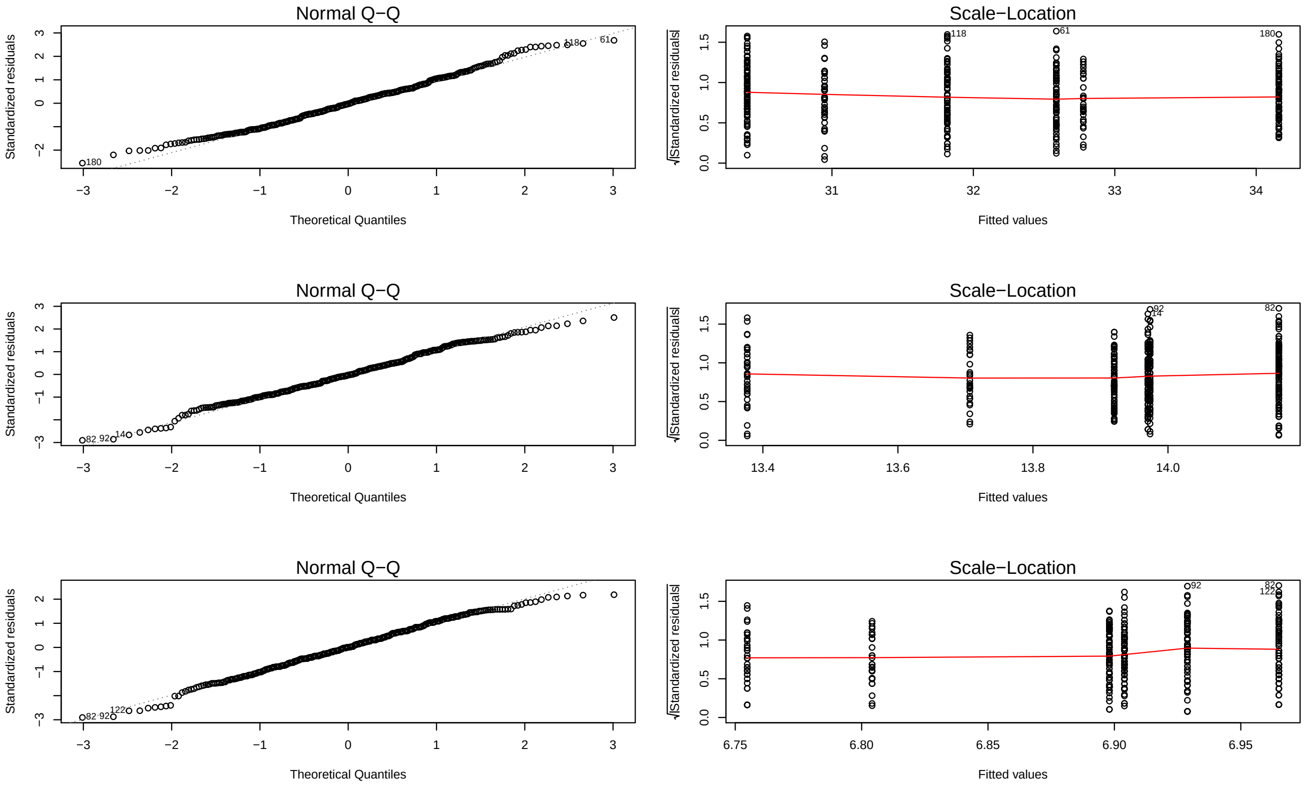
Take note that we need to limit our analysis to our climate type testing sites again as follows (we include Manitoba this time as it is at the same latitude as the UK and Siberia and holds a semi-coastal climate type):

```
# prepare climate type testing data
Data_df <- Data_df_base
Index <- Data_df$Index
Rows <- which(Index == "SI" | Index == "UK" | Index == "RE" | Index ==
    "AU" | Index == "MA")
Data_df <- Data_df[Rows, ]
```

5.2.1 Assumption Check

Let's put the ANOVACkeck_TWO() function to the test and check whether an ANOVA approach is viable give our data:

```
par(mfrow = c(3, 2))
ANOVACheck_TWO(Formulas = c(Weight ~ Climate * Sex, Height ~ Climate *
    Sex, Wing.Chord ~ Climate * Sex), data = Data_df, plotting = TRUE)
```



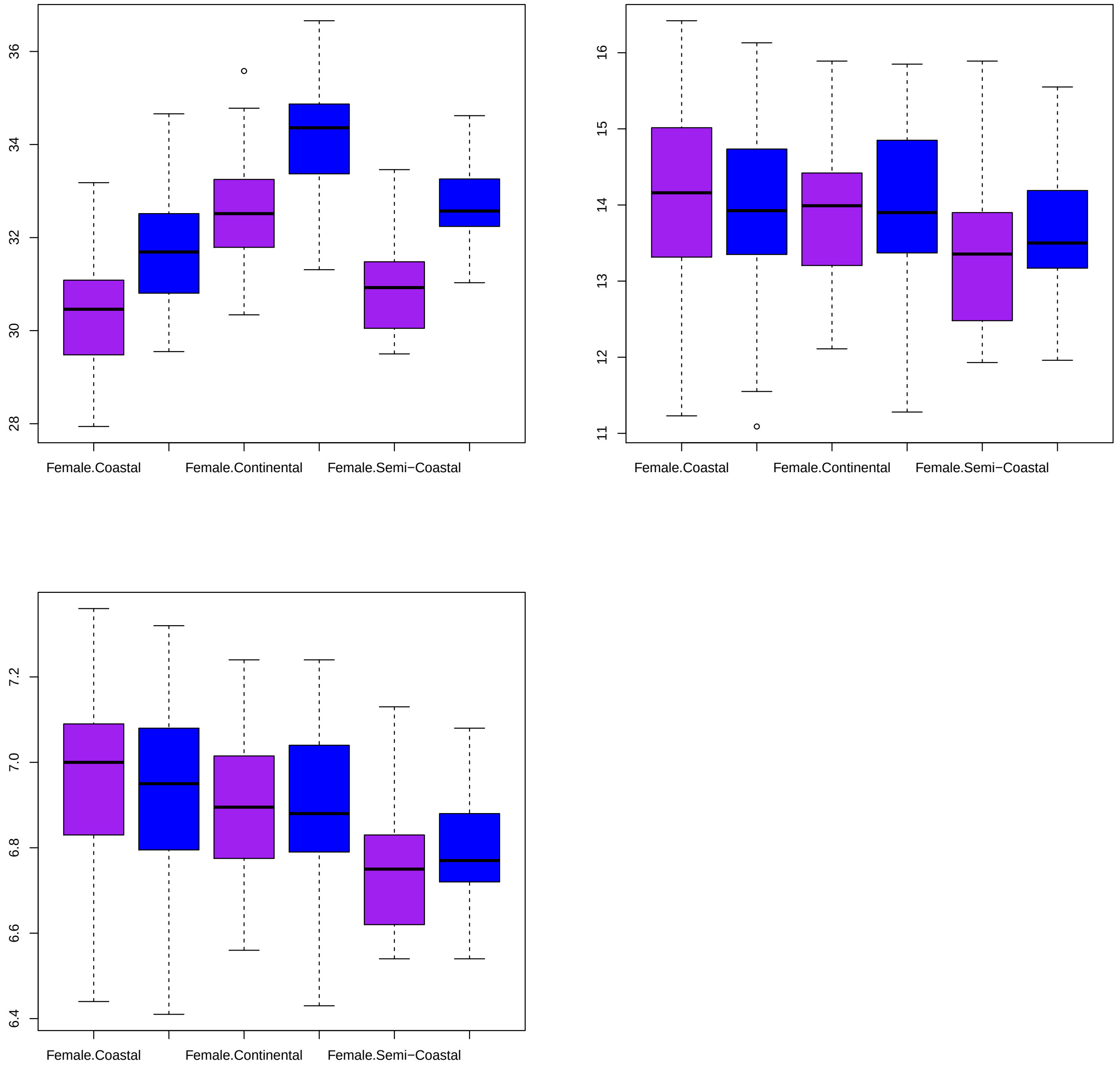
```
##      Weight ~ Climate * Sex Height ~ Climate * Sex Wing.Chord ~ Climate * Sex
## RN                    0.015                  0.083                      0.030
## HoV                   0.408                  0.403                      0.071
```

Sadly, not enough of our assumptions are met.

5.2.2 Analysis

Since none of our assumptions are met, we cannot run an ANOVA and therefore resort to data visualisation alone:

```r
par(mfrow = c(2, 2))
boxplot(Weight ~ Sex * Climate, data = Data_df, col = c("purple", "blue"))
boxplot(Height ~ Sex * Climate, data = Data_df, col = c("purple", "blue"))
boxplot(Wing.Chord ~ Sex * Climate, data = Data_df, col = c("purple", "blue"))
```



5.3 Sexual Dimorphism

Does sparrow morphology depend on population status and sex?

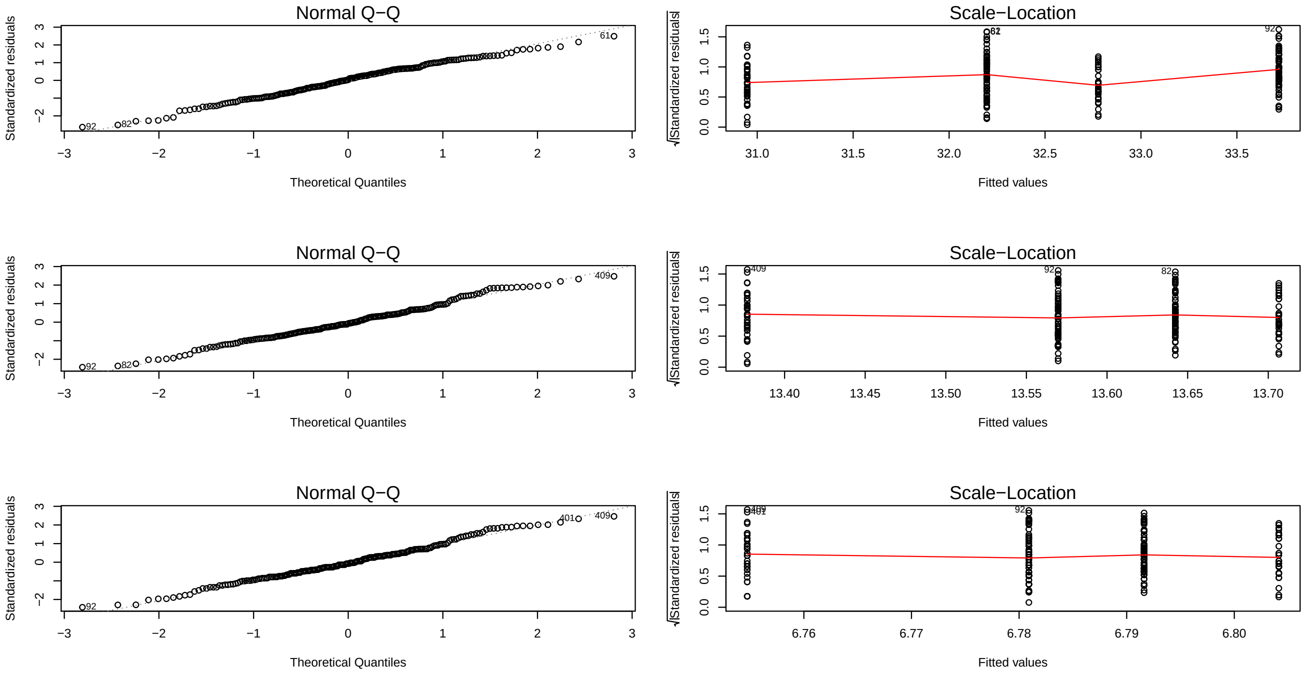
Given different factors affecting invasive species, we might expect different patterns of sexual dimorphism for invasive and native populations. Take note that we keep using the northern hemisphere subset our cimate testing sites as these present us with a nice set of invasive/native population records already whilst keeping confounding factors to a minimum.

5.3.1 Assumption Check

First, we need to check our assumptions:

```r
# prepare climate type testing data
Data_df <- Data_df_base
Index <- Data_df$Index
Rows <- which(Index == "SI" | Index == "UK" | Index == "MA")
Data_df <- Data_df[Rows, ]

# analysis
par(mfrow = c(3, 2))
ANOVACheck_TWO(Formulas = c(Weight ~ Population.Status * Sex, Height ~
    Population.Status * Sex, Wing.Chord ~ Population.Status * Sex), data = Data_df,
    plotting = TRUE)
```



```
##     Weight ~ Population.Status * Sex Height ~ Population.Status * Sex
## RN                            0.2880                             0.22
## HoV                           0.0045                             0.91
##     Wing.Chord ~ Population.Status * Sex
## RN                                  0.19
## HoV                                 0.92
```

Again our assumptions are not met except for sparrow height and wing chord as a product of sex and population status.

5.3.2 Analysis

Let's run our analysis:

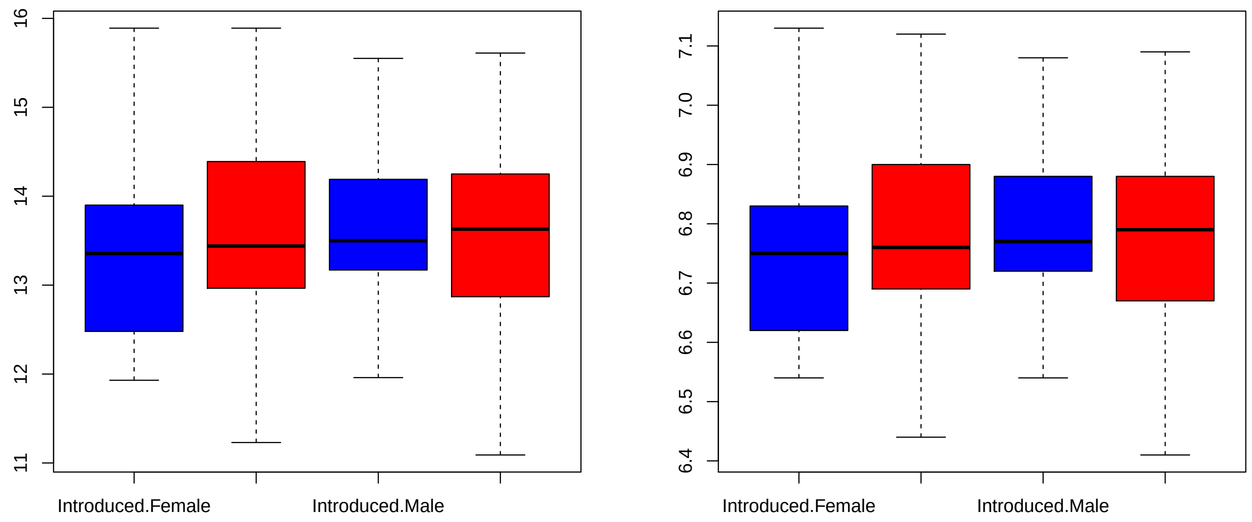
```
# height model
model <- lm(Height ~ Population.Status * Sex, data = Data_df)
anova(model)
```

```
## Analysis of Variance Table
##
## Response: Height
##                        Df Sum Sq Mean Sq F value Pr(>F)
## Population.Status       1    0.3   0.338    0.32   0.57
## Sex                     1    0.2   0.179    0.17   0.68
## Population.Status:Sex   1    1.8   1.786    1.68   0.20
## Residuals             197  208.9   1.060
```

```
# wing chord model
model <- lm(Wing.Chord ~ Population.Status * Sex, data = Data_df)
anova(model)
```

```
## Analysis of Variance Table
##
## Response: Wing.Chord
##                        Df Sum Sq Mean Sq F value Pr(>F)
## Population.Status       1   0.00  0.0047    0.20   0.66
## Sex                     1   0.00  0.0041    0.17   0.68
## Population.Status:Sex   1   0.04  0.0399    1.67   0.20
## Residuals             197   4.70  0.0239
```

```
# plotting
par(mfrow = c(1, 2))
boxplot(Height ~ Population.Status * Sex, data = Data_df, col = c("blue",
    "red"))
boxplot(Wing.Chord ~ Population.Status * Sex, data = Data_df, col = c("blue",
    "red"))
```



As it turns out, population status and sex are no viable predictors for sparrow height or wing chord and so we **accept the null hypothesis**.

6. ANCOVA

Assumptions of the ANCOVA:

- Predictor variables are categorical or continuous
- Response variable is metric
- *Response variable residuals* are **normal distributed**
- Variance of populations/samples are equal (**homogeneity**)
- Variable values are **independent** (not paired)
- Relationship between the response and covariate is **linear**.

6.1 Climate Warming/Extremes

Do sparrow characteristics depend on climate and latitude?

Latitude may have masked some effects of climate on sparrow morphology in our preceding analyses and vice-versa. At times, we have been able to account for this by including our site records, which can be seen as binned versions of latitude records. Let's test if the inclusion of raw latitude records are meaningful.

6.1.1 Assumption Check

Again, we need to do an assumption check. However, we need a new function for this, since we now need to test whether our response variable and the covariate are linear or not:

```
# overwriting prior changes in Data_df
Data_df <- Data_df_base
Data_df$Latitude <- abs(Data_df$Latitude)
# User-defined function
ANCOVACheck <- function(Variables, Grouping, Covariate, data, plotting) {
    Output <- data.frame(x = NA)
    for (i in 1:length(Variables)) {
        # data
        Y <- as.numeric(data[, Variables[i]])
        X <- data[, Grouping]
        Z <- data[, Covariate]
        # Residuals?
        model <- lm(Y ~ X * Z)
        Output[1, i] <- shapiro.test(residuals(model))$p.value
        # Homgeneity?
        Levene <- leveneTest(Y ~ X, center = median, data = data)
        Output[2, i] <- Levene[1, 3]
        # Plotting
        if (plotting == TRUE) {
            plot(model, 1)  # Linearity
            plot(model, 2)  # Normality
            plot(model, 3)  # Homogeneity
        }
    }
    colnames(Output) <- Variables
    rownames(Output) <- c("RN", "HoV")
    return(Output)
}
```

This ANCOVACheck() function takes five arguments: (1) Variables - a vector of response variables used in our models, (2) Grouping - the categorical variable by which to group our variables, (3) Covariate - the covariate of our analysis, (4)data - the data frame which contains the variables, the grouping factor and our covariate, and (5) plotting - a logical indicator of whether to produce plots visualising the test results or not.
The function returns a data frames containing the p-values indexing whether to accept or reject the notion of the normality of residuals per variable (RN), and the p-values indexing whether variances between groups are homogeneous or not (HoV).

```
ANCOVACheck(Variables = c("Weight", "Height", "Wing.Chord", "Nesting.Height",
    "Egg.Weight", "Number.of.Eggs", "Home.Range"), Grouping = "Climate",
    Covariate = "Latitude", data = Data_df, plotting = FALSE)
```

```
##      Weight  Height Wing.Chord Nesting.Height Egg.Weight Number.of.Eggs Home.Range
## RN  6.2e-10 9.4e-05    1.8e-16        5.5e-18      7e-02        9.6e-14    2.9e-20
## HoV 5.2e-24 1.1e-24    3.2e-28        4.0e-01      1e-06        1.3e-13    1.2e-08
```

The assumptions aren't met. I have set the plotting argument to FALSE tu suppress the plotting of model checking visualisation. The would be useful to judge linearity but not necessary here since the other two important assumptions (Homogeneity of variances and Normality of residuals) aren't met to begin with.

6.1.2 Analysis

Since none of our assumptions are met, we cannot run an ANOVA and therefore resort to data visualisation alone. We need a new function for this to do our plotting easily and automatically with some colours indicating our grouping factors whilst plotting response variables versus covariates.

```
PlotAncovas <- function(Variables, Grouping, Covariate, data){
  for(i in 1:length(Variables)){
    Y <- Data_df[,Variables[i]]
    X <- Data_df[,Covariate]
    G <- Data_df[, Grouping]
    plot(X, Y, col = G, xlab = Covariate, ylab = Variables[i])
    legend("top", # place legend at the top
           inset = -0.35, # move legend away from plot centre
           xpd = TRUE, # allow legend outside of plot area
           legend=levels(G), # what to include in legend
           bg = "white", col = unique(G), ncol=length(levels(G)), # colours
           pch = 1, # plotting symbols
           title = Variables[i] # title of legend
           )
  }
}
```

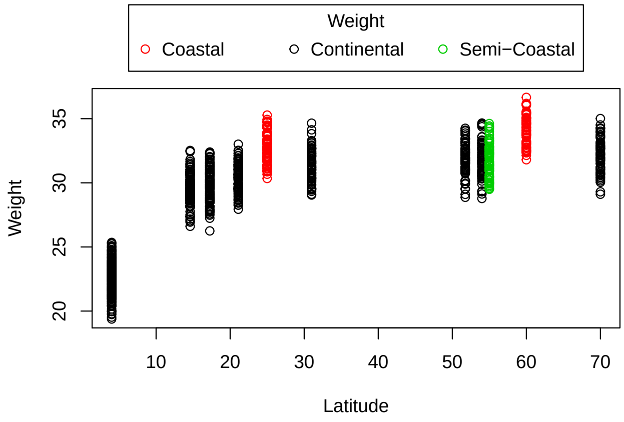
The PlotAncovas() returns a scatter plot and takes four arguments: (1) Variables - a vector of response variables, (2) Grouping - the name of the grouping factor according to which to colour the symbols in our plot, (3) Covariate - the covariate against which to plot individuals variables, and (4) data - the data frame which holds our variables.

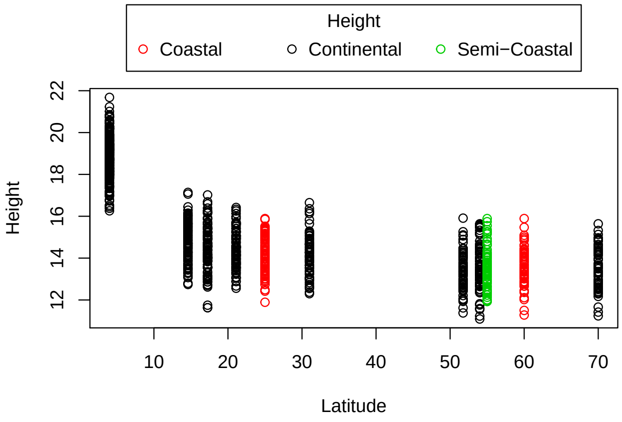
Let's use our function:

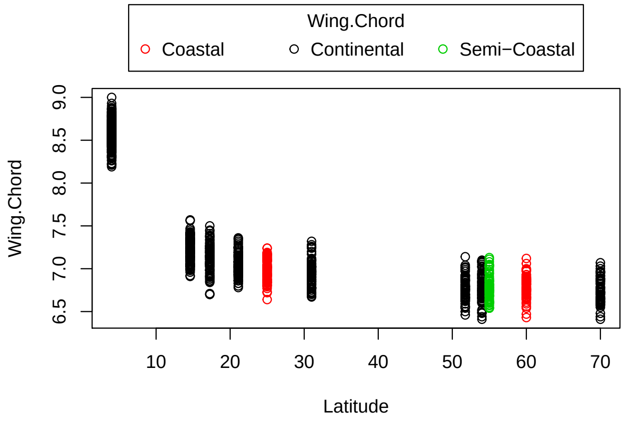
```
par(mfrow = c(1, 2))
PlotAncovas(Variables = c("Weight", "Height", "Wing.Chord", "Nesting.Height",
    "Egg.Weight", "Number.of.Eggs", "Home.Range"), Grouping = "Climate",
    Covariate = "Latitude", data = Data_df)
```

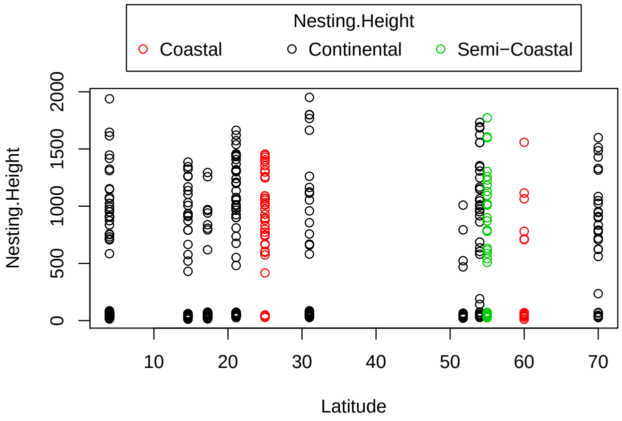
I will not interpret these plots here in text and leave this to you.

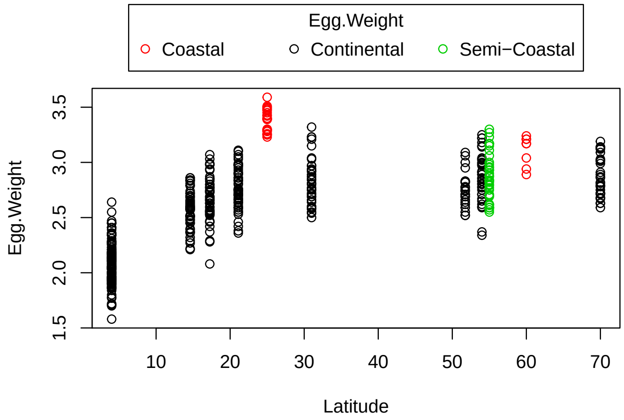
Take note that this **could've been achieved much easier with ggplot2!**

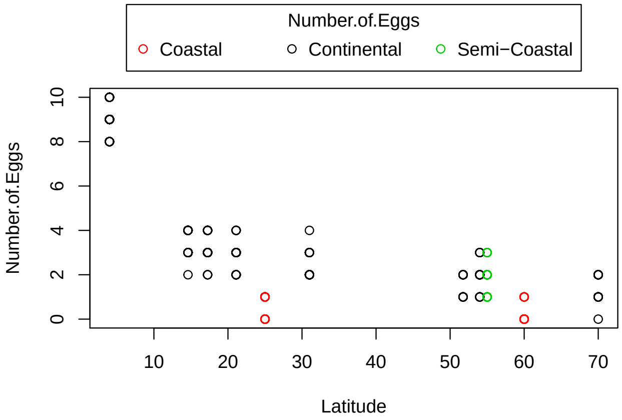

Weight
○ Coastal
○ Continental
○ Semi−Coastal
Weight
20
25
30
35
10
20
30
40
50
60
70
Latitude

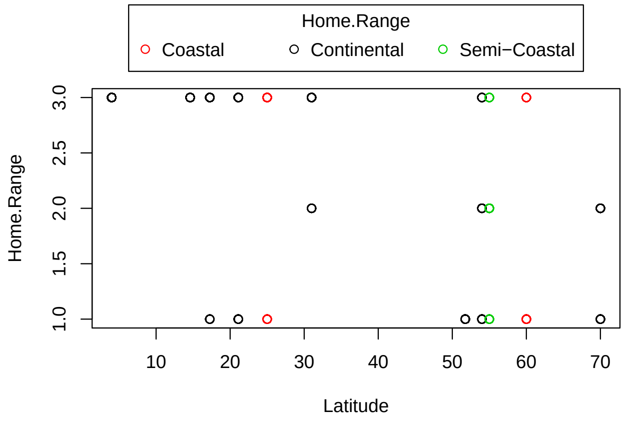

Height
○ Coastal
○ Continental
○ Semi−Coastal
Height
12
14
16
18
20
22
10
20
30
40
50
60
70
Latitude


Wing.Chord
○ Coastal
○ Continental
○ Semi−Coastal
Wing.Chord
6.5
7.0
7.5
8.0
8.5
9.0
10
20
30
40
50
60
70
Latitude


Nesting.Height
○ Coastal
○ Continental
○ Semi−Coastal
Nesting.Height
0
500
1000
1500
2000
10
20
30
40
50
60
70
Latitude


Egg.Weight
○ Coastal
○ Continental
○ Semi−Coastal
Egg.Weight
1.5
2.0
2.5
3.0
3.5
10
20
30
40
50
60
70
Latitude


Number.of.Eggs
○ Coastal
○ Continental
○ Semi−Coastal
Number.of.Eggs
0
2
4
6
8
10
10
20
30
40
50
60
70
Latitude


Home.Range
○ Coastal
○ Continental
○ Semi−Coastal
Home.Range
1.0
1.5
2.0
2.5
3.0
10
20
30
40
50
60
70
Latitude

6.2 Sparrow Characteristics And Sites

This was not part of what we set out to do according to the lecture slides but has been included as a logical conclusion to an earlier analysis.

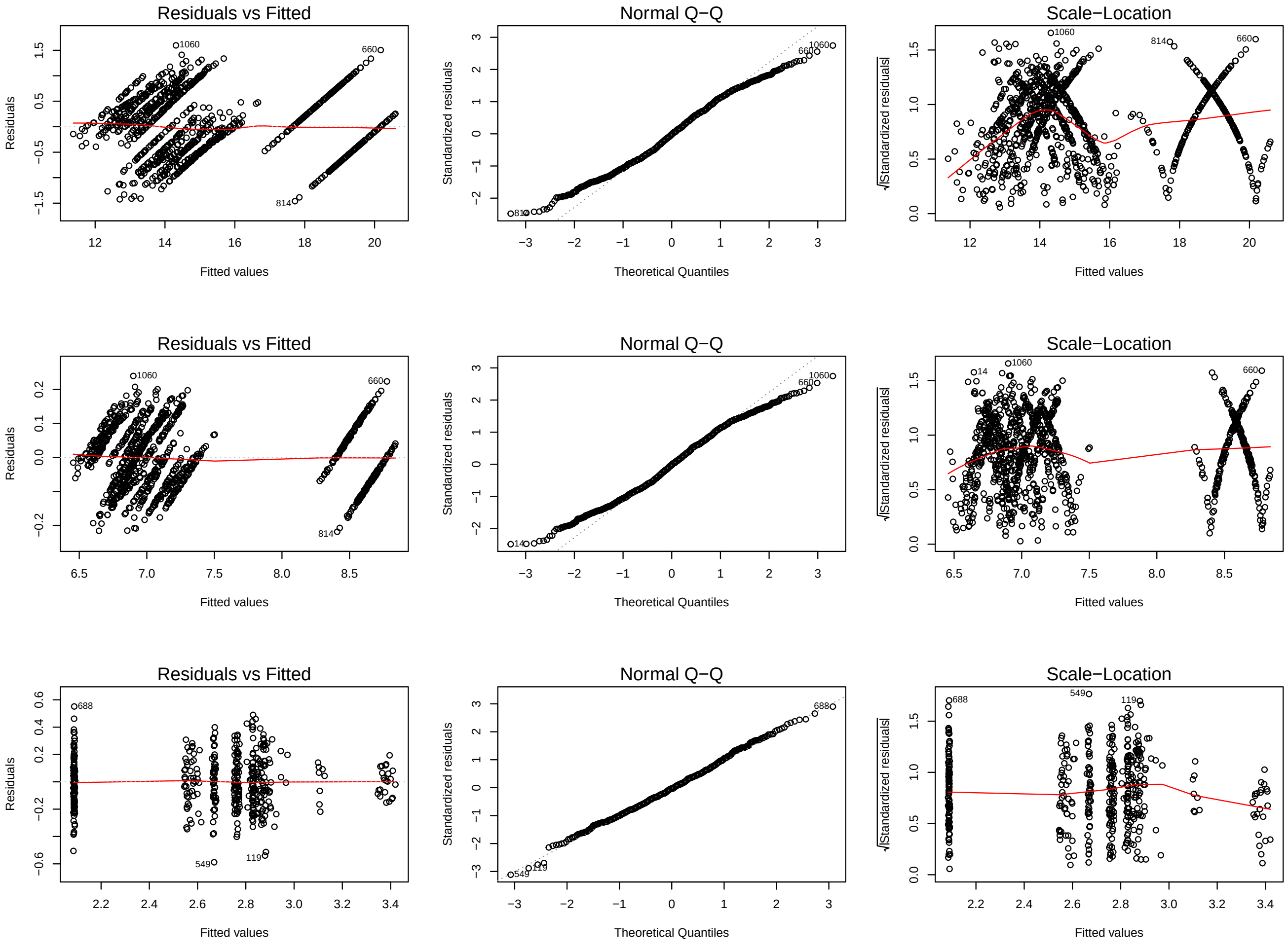
Unfortunately, our previous attempt at an ANCOVA didn't work. So what other covariate do we have available for sparrow characteristics?
- *Latitude* doesn't make sense to include when grouping by site index as these two are synonymous - *Longitude* doesn't make sense to include when grouping by site index as these two are synonymous - *Weight* is well explained by other variables and we know the causal links - *Height* is not that well explained by other variables - *Wing.Chord* is not that well explained by other variables

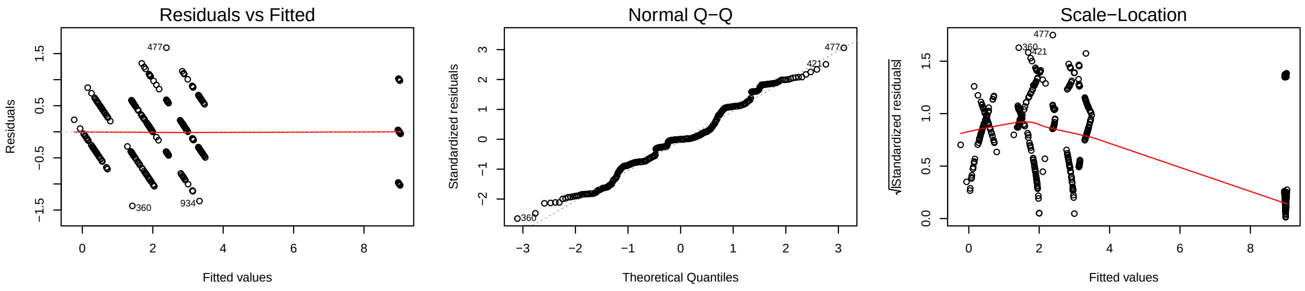
Of course, there are more within our data set but it has become apparent that `Weight` may make for an important covariate in our site-wise ANCOVA set-up. Using the Pearson correlation (third practical), we already identified a causal link between sparrow `Weight` and `Height` per site.

6.2.1 Assumption Check

Firstly, we test whether assumptions are met. For brevities sake, we only test four variables:

```r
par(mfrow = c(1, 3))
ANCOVACheck(Variables = c("Height", "Wing.Chord", "Egg.Weight", "Number.of.Eggs"),
    Grouping = "Index", Covariate = "Weight", data = Data_df, plotting = TRUE)
```





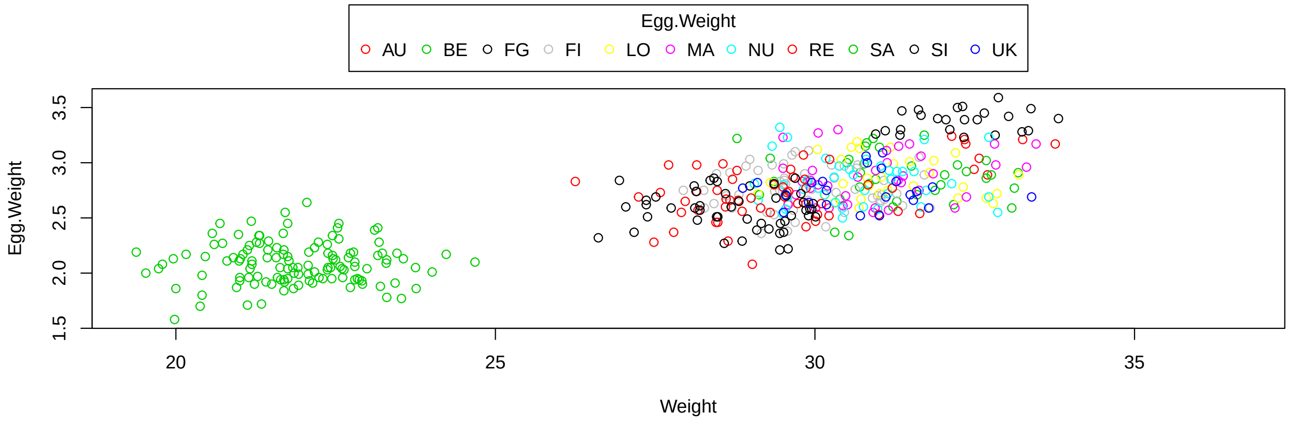
```
##      Height Wing.Chord Egg.Weight Number.of.Eggs
## RN  1.2e-07    1.2e-07      0.512        9.2e-07
## HoV 2.6e-01    2.5e-01      0.085        2.7e-02
```

As it turns out, we can run our ANCOVA on `Egg.Weight` when grouped by site `Index` and driven by `Weight`.

6.2.2 Analysis

First, let's visualise our data:

```
PlotAncovas(Variables = "Egg.Weight", Grouping = "Index", Covariate = "Weight",
    data = Data_df)
```



Quite obviously, Belize (BE) records are very different from the other stations, whose egg weight and weight records are grouped together. There seems to be some evidence for an overall linkage of sparrow weight and egg weight (a positive correlation).

Now we run the analysis:

```
LM_fit5 <- lm(Egg.Weight ~ Weight * Index, data = Data_df)
anova(LM_fit5)
```

```
## Analysis of Variance Table
##
## Response: Egg.Weight
##               Df Sum Sq Mean Sq F value Pr(>F)
## Weight         1   52.5    52.5 1442.56 <2e-16 ***
## Index         10    8.1     0.8   22.21 <2e-16 ***
## Weight:Index  10    0.1     0.0    0.35   0.97
## Residuals    455   16.6     0.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The above ANCOVA output tells us that there is no interaction effect between sites and sparrow weights when determining mean egg weight per nest of *Passer domesticus* and so we do another iteration of our model and remove the postulated interaction:

```
LM_fit6 <- lm(Egg.Weight ~ Weight + Index, data = Data_df)
anova(LM_fit6)
```

```
## Analysis of Variance Table
##
## Response: Egg.Weight
##            Df Sum Sq Mean Sq F value Pr(>F)
## Weight      1   52.5    52.5  1462.9 <2e-16 ***
## Index      10    8.1     0.8    22.5 <2e-16 ***
## Residuals 465   16.7     0.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By now, all of our model coefficients are significant and we can go on to interpret them:

```
summary(LM_fit6)
```

```
##
## Call:
## lm(formula = Egg.Weight ~ Weight + Index, data = Data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5889 -0.1315 -0.0062  0.1203  0.5514
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.34598    0.28083   11.91  < 2e-16 ***
## Weight       0.00108    0.00861    0.13  0.90020
## IndexBE     -0.70848    0.05486  -12.91  < 2e-16 ***
## IndexFG     -1.28117    0.09914  -12.92  < 2e-16 ***
## IndexFI     -0.62529    0.05744  -10.89  < 2e-16 ***
## IndexLO     -0.55014    0.05175  -10.63  < 2e-16 ***
## IndexMA     -0.51365    0.05135  -10.00  < 2e-16 ***
## IndexNU     -0.51702    0.05337   -9.69  < 2e-16 ***
## IndexRE     -0.61263    0.05142  -11.91  < 2e-16 ***
## IndexSA     -0.80605    0.05669  -14.22  < 2e-16 ***
## IndexSI     -0.27258    0.07787   -3.50  0.00051 ***
## IndexUK     -0.51167    0.05140   -9.95  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.19 on 465 degrees of freedom
##   (590 observations deleted due to missingness)
## Multiple R-squared:  0.784,  Adjusted R-squared:  0.779
## F-statistic:  153 on 11 and 465 DF,  p-value: <2e-16
```