Pipeline_Analysis

1 Import data

Our first step is to import packages and data into R. The trait data and OMI data also need to be merged.

Load packages

```
library(tinytex)
library(ape)
library(dplyr)
library(usdm)
library(caret)
library(corrplot)
library(phylolm)
```

Import data

The datasets can be found in different GitHub repositories. The ungulate dataset and tree can be found in the trait-organismal-ungulates repository. The OMI data is found in the trait-geo-diverse-ungulates repository.

```
ungulatesData <- read.csv("https://raw.githubusercontent.com/naturalis/trait-organismal-ungulates/maste
omi <- read.csv("https://raw.githubusercontent.com/naturalis/trait-geo-diverse-ungulates/master/results
tree <- read.tree("https://raw.githubusercontent.com/naturalis/trait-organismal-ungulates/master/data/u
```

Merge datasets

The ungulate data and OMI data have to be merged into one dataset. The EoL-ID is removed and the data is merged by the canonical name (present in both the datasets). The last step is to replace the spaces in the canonical name with underscores, to match the species names in the tree.

```
ungulatesData <- ungulatesData[2:60]
names(omi)[names(omi)=="X"] <- "CanonicalName"
dataset <- merge(ungulatesData, omi, by="CanonicalName")
dataset$CanonicalName <- gsub(" ", "_", dataset$CanonicalName)

# Clean up the global environment
rm(ungulatesData, omi)
```

2 Preprocessing

Equalize species in tree and dataset

To start, 'Equus asinus' is renamed to the 'Equus africanus' in the tree, to match the dataset. The species that aren't in the tree are dropped from the dataset. The species that aren't in the dataset are dropped from the tree.

```
# Changed Equus asinus to Equus africanus in the tree
tree$tip.label[tree$tip.label=="Equus_asinus"] <- "Equus_africanus"

# Check Which species aren't in the tree
dropRows <- setdiff(dataset$CanonicalName, tree$tip.label)

# Drop rows that aren't in the tree (check manually if these are domesticated)
row.names(dataset) <- dataset$CanonicalName
dataset <- dataset[!(row.names(dataset) %in% dropRows), ]

# Drop tips that aren't in dataset
dropTips <- setdiff(tree$tip.label, dataset$CanonicalName)
tree <- drop.tip(tree, dropTips)

# Final check to see if there are any differences
setdiff(dataset$CanonicalName, tree$tip.label)
setdiff(tree$tip.label, dataset$CanonicalName)

rm(dropRows, dropTips)
```

Miscellaneous preprocessing

The dots in the column names are replaced with underscores. After that, the traits that consist of more than 100 missing values, traits that have no information gain and traits that are almost identical to other traits are removed.

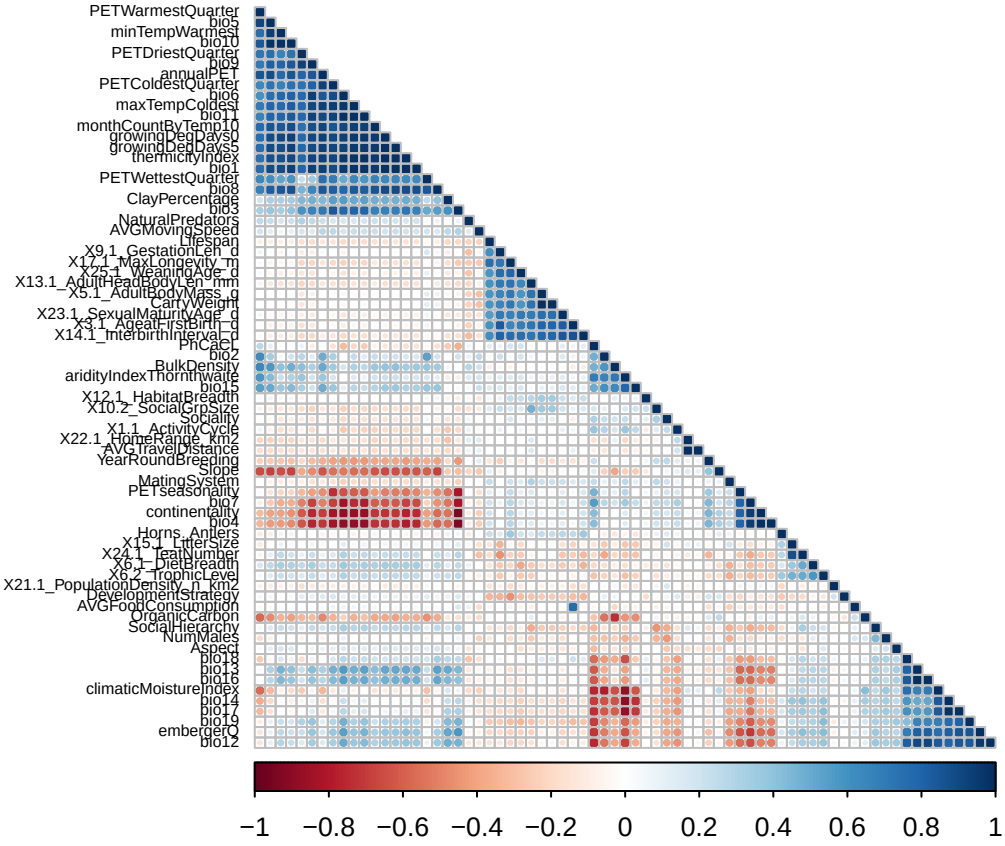
```
# Rename columns with dots in the name
names(dataset)[names(dataset)=="Horns.Antlers"] <- "Horns_Antlers"
names(dataset)[names(dataset)=="X21.1_PopulationDensity_n.km2"] <- "X21.1_PopulationDensity_n_km2"

# Remove traits that (almost) only consist of missing values (>100 NA)
dataset <- subset(dataset, select = -c(X5.4_WeaningBodyMass_g, X13.3_WeaningHeadBodyLen_mm, X13.2_Neona

# Remove traits without any information gain (only consist of one value)
dataset <- subset(dataset, select = -c(Motility, ParentalCare, X12.2_Terrestriality))

# Remove traits that are almost identical to other traits
dataset <- subset(dataset, select = -c(PullStrength, NumOffspring, BreedingInterval,
                                       Diet, AVGWeight, MaturityReachFemale, MaturityReachMale,
                                       X22.2_HomeRange_Indiv_km2, X5.3_NeonateBodyMass_g,
                                       X16.1_LittersPerYear,X7.1_DispersalAge_d))
```

3 VIF-Analysis

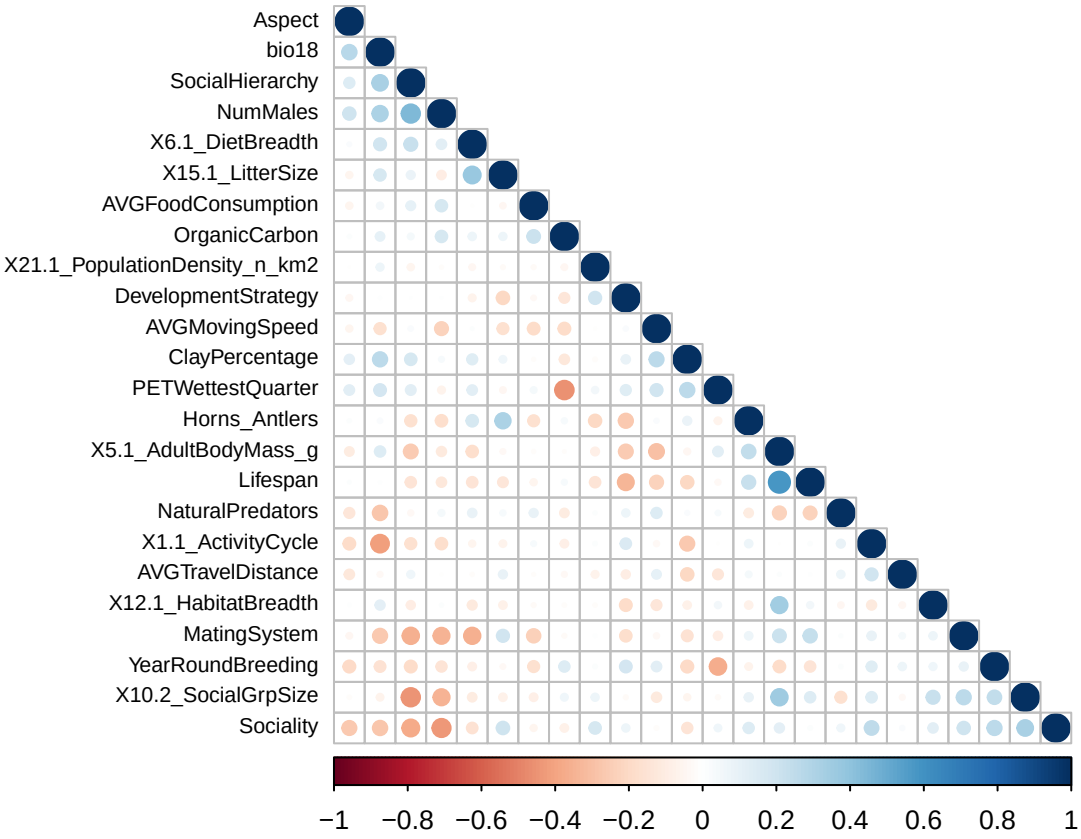
There is probably collinearity present amongst the traits in the dataset. Collinearity can lead to bias in the model, so we must correct for this. This can be done by running a variable inflation factor analysis (VIF). The VIF.R script contains the whole VIF analysis. The R script is sourced below. This script only requires the 'dataset' variable and after it is run, it will output the 'predictors' variable.

```
source("VIF.R")
```

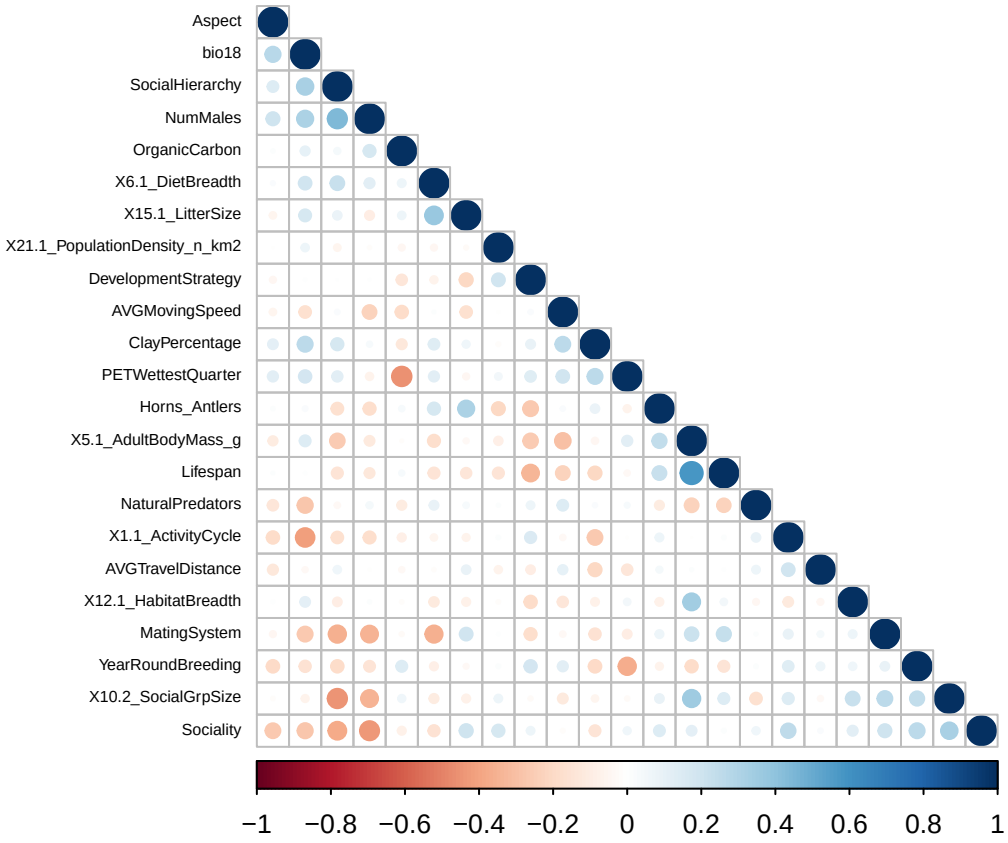
```
## The first correlation matrix is a visualization of the dataset without the removal of any traits.
##     There is a lot of collinearity present. The VIF analysis will take care of this.
```




```
## The second corrplot is a visualization of the dataset after cutoff values are implemented and
##     adultbodymass is added.
```




```
## The third correlation matrix is our final product after removing all highly correlated traits.
```



4 Model Selection

A Generalized Linear Model (GLM) must be made. The Domestication column is used as the dependent variable, and the other columns are the predictor variables. For the model selection the phylolm package and phyloglmstep function are required.

```
source("ModelSelection.R")
```

Results:

```
   AIC     logLik Pen.logLik
28.346     -7.173     11.847

Parameter estimate(s) from MPLE:
alpha: 0.01145475

Coefficients:
     (Intercept) X5.1_AdultBodyMass_g  DevelopmentStrategy        Horns_Antlers
-23.861324750517       0.000006403856       4.695920350903       8.733986057383
  AVGMovingSpeed    AVGTravelDistance
 -0.271395434194       0.006078751863
```

5 Modelling analysis

The phylogenetic generalized linear modelling analysis optimizes the model. The function phyloglm from the phylolm package is used for this.

```
# Converting dependent variable to binary state
# Domesticated = 1, wild = 0
dataset$Domestication[dataset$Domestication==2] <- 0

# Construct model
GLM <- phyloglm(formula = finalFormula, data = dataset, phy = tree, method = "logistic_MPLE", btol = 36
summary(GLM)
```

Results:

```
Coefficients:
                         Estimate        StdErr z.value  p.value
(Intercept)          -9.3503949900  5.1361469886 -1.8205 0.068682 .
X5.1_AdultBodyMass_g  0.0000045743  0.0000020387  2.2437 0.024850 *
DevelopmentStrategy   1.9876564874  1.1964451827  1.6613 0.096653 .
Horns_Antlers         4.9891203727  1.8993795774  2.6267 0.008621 **
AVGMovingSpeed       -0.2389830249  0.0950728567 -2.5137 0.011948 *
AVGTravelDistance     0.0046891017  0.0033020300  1.4201 0.155588
```

6 Model evaluation

The model is evaluated to check whether the model constructed, is still in compliance with the assumptions that have to be met when performing a GLM. The package 'performance' is used for this. The function 'check_model()' is used for a comprehensive model check. This function is not appropriate for models including phylogeny. The check will still be done, but the resulting interpretation may be skewed.

```
# Load packages
library(performance)
library(see)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

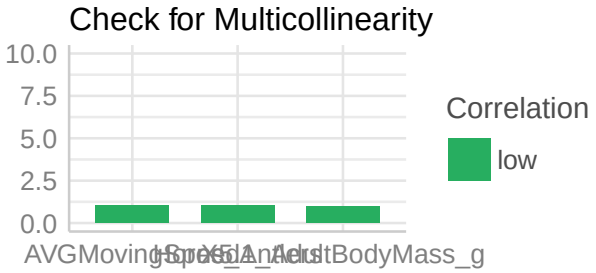
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

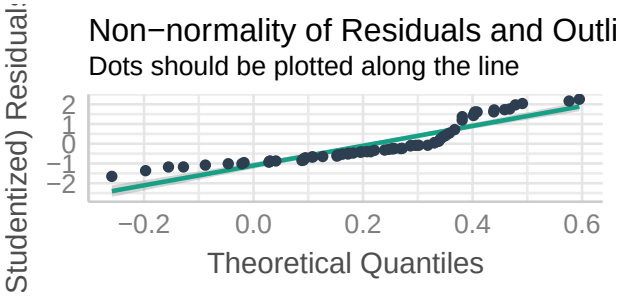
```
# Without DevelopmentStrategy
model1_Gaussian <- glm(Domestication ~ 1 + X5.1_AdultBodyMass_g + Horns_Antlers + AVGMovingSpeed, data =
model1_Binomial <-  glm(Domestication ~ 1 + X5.1_AdultBodyMass_g + Horns_Antlers + AVGMovingSpeed, data

# With DevelopmentStrategy
model2_Gaussian <- glm(Domestication ~ 1 + X5.1_AdultBodyMass_g + DevelopmentStrategy + Horns_Antlers +
model2_binomial <- glm(Domestication ~ 1 + X5.1_AdultBodyMass_g + DevelopmentStrategy + Horns_Antlers +

check_model(model1_Gaussian)
```

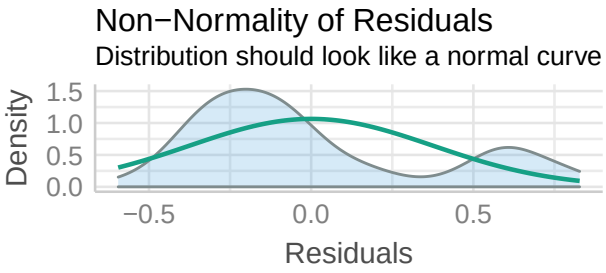
```
## `geom_smooth()` using formula 'y ~ x'
```

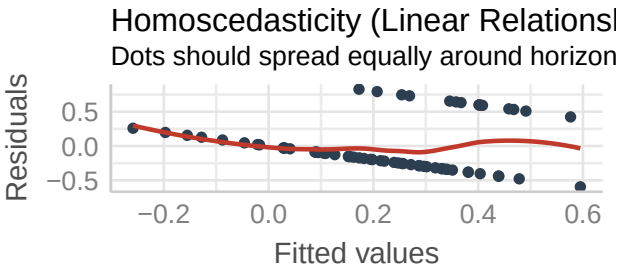
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

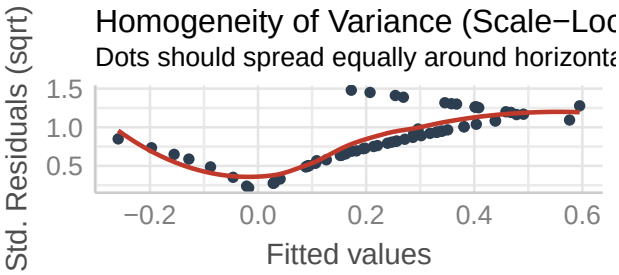
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

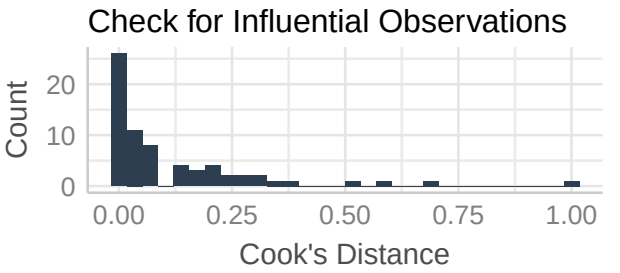










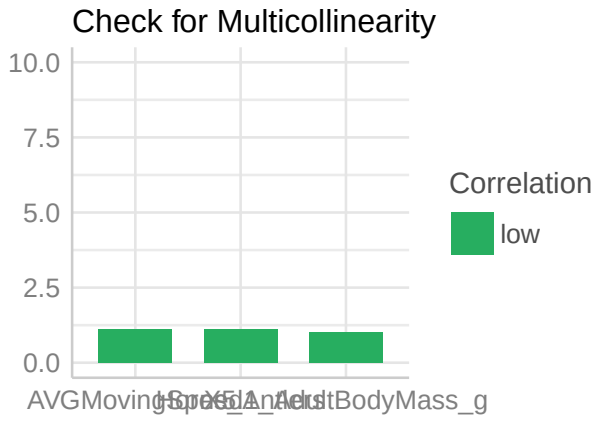


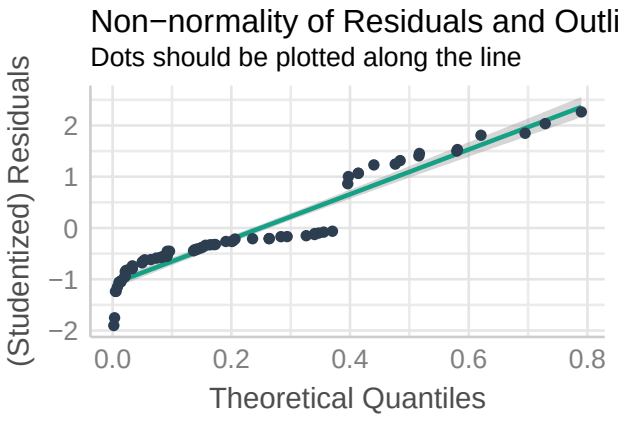

```
check_model(model1_Binomial)
```

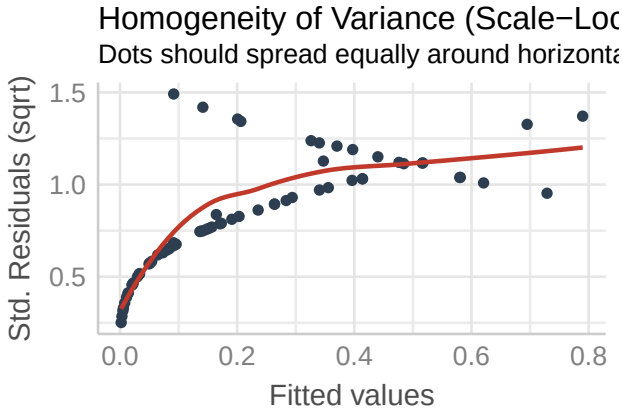
```
## `geom_smooth()` using formula 'y ~ x'
```

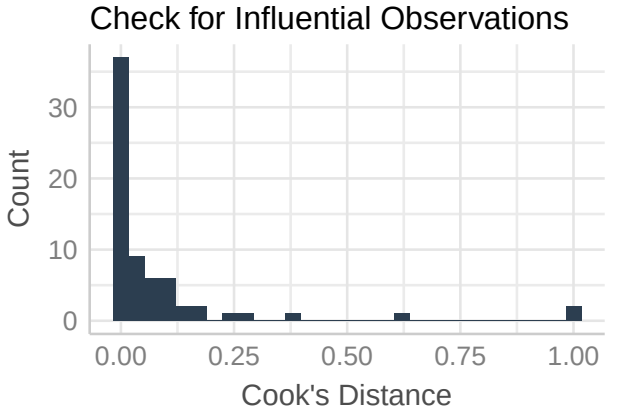
```
## `geom_smooth()` using formula 'y ~ x'
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```







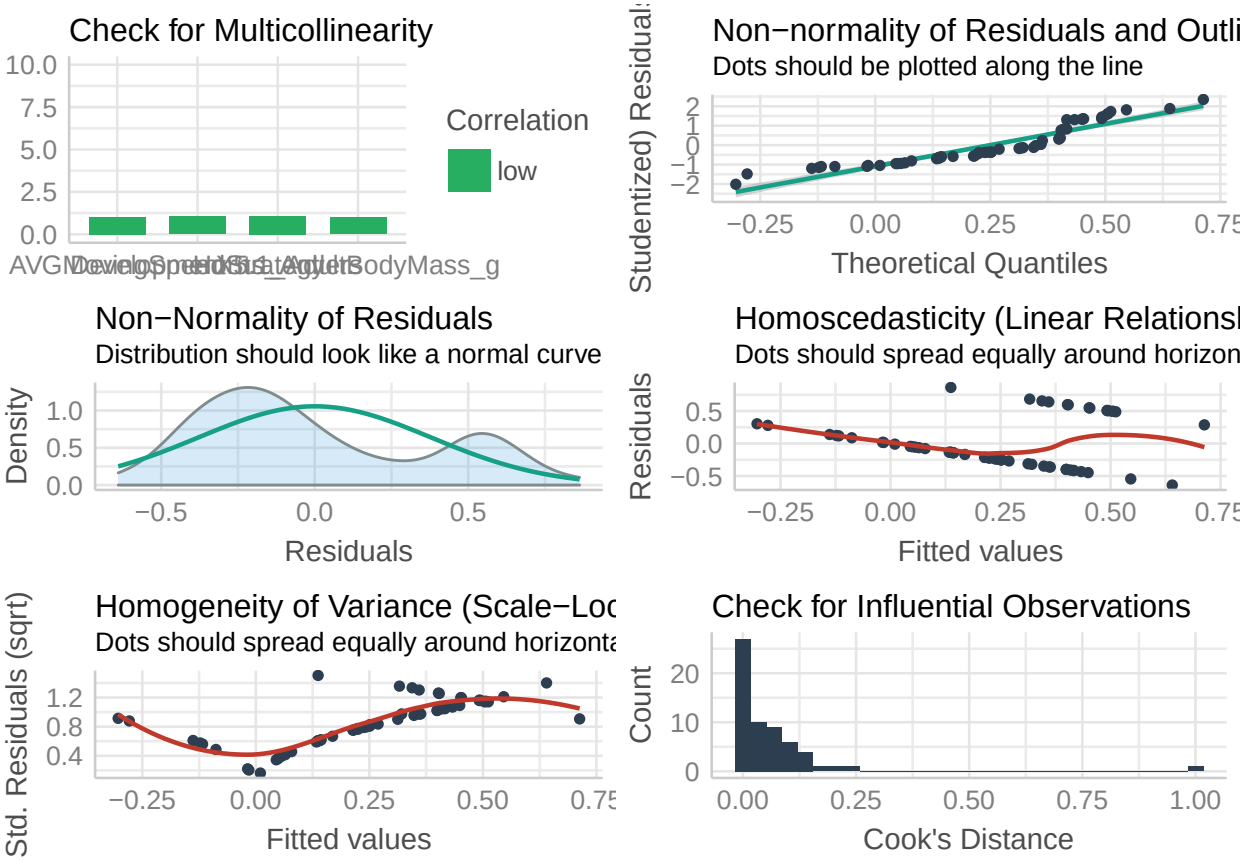



```
check_model(model2_Gaussian)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

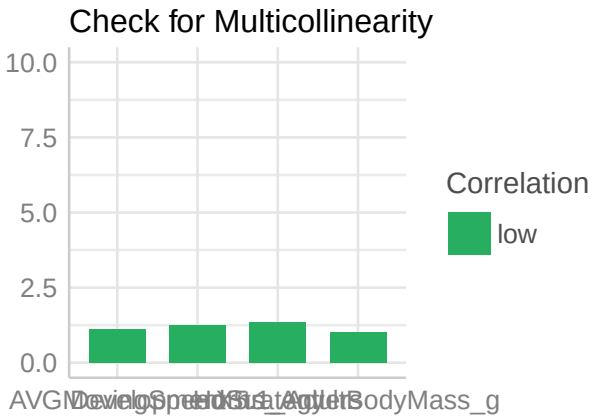
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

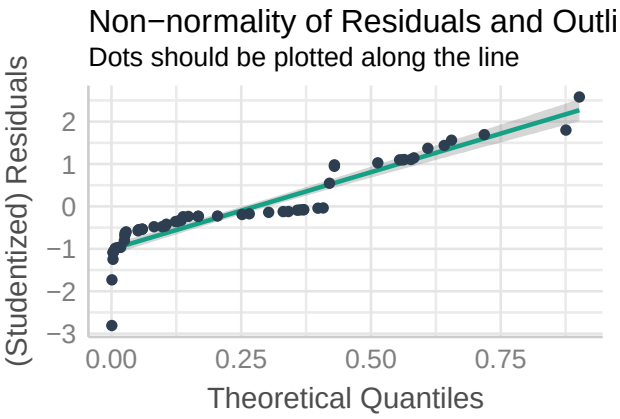
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

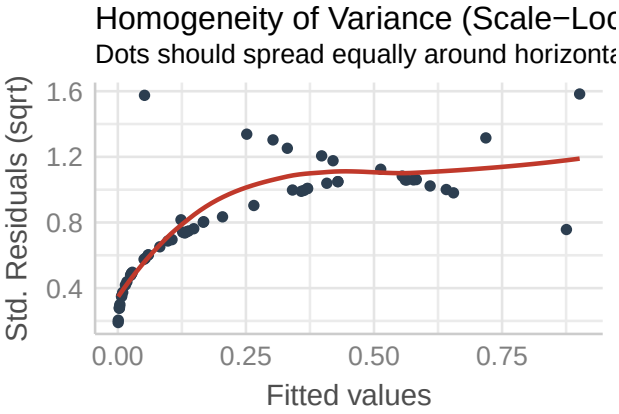


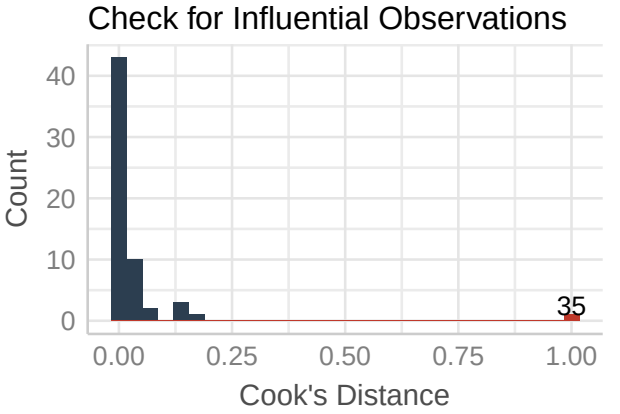

```
check_model(model2_binomial)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





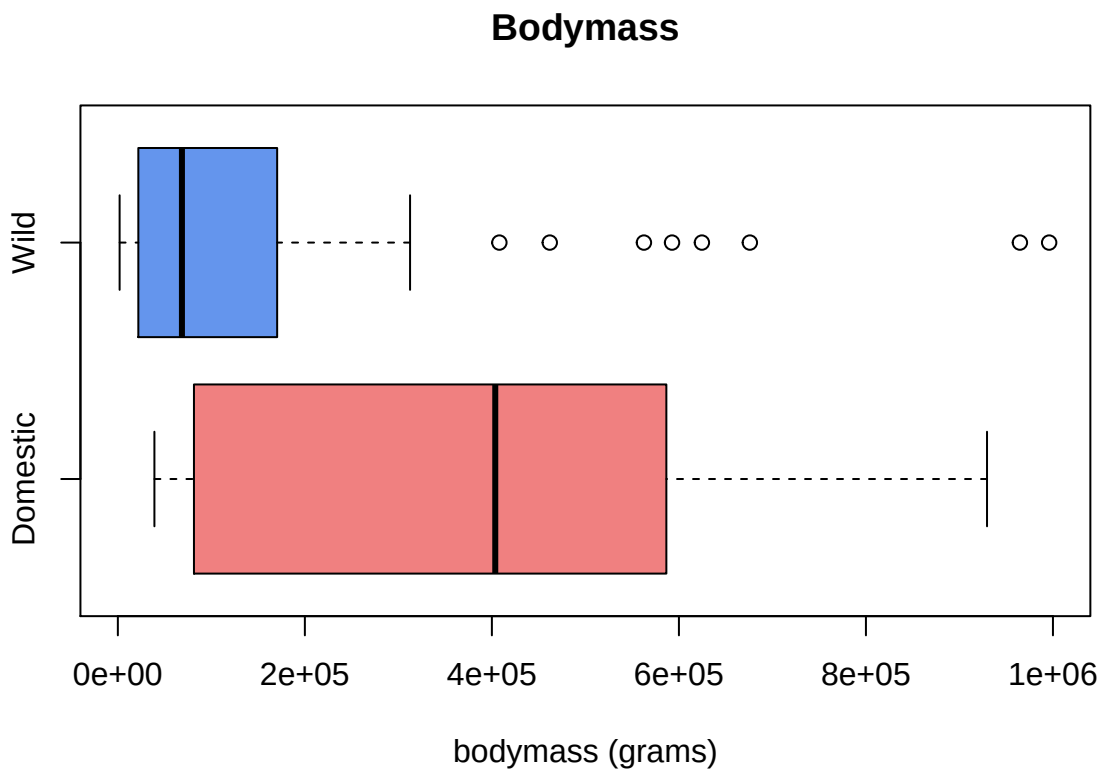


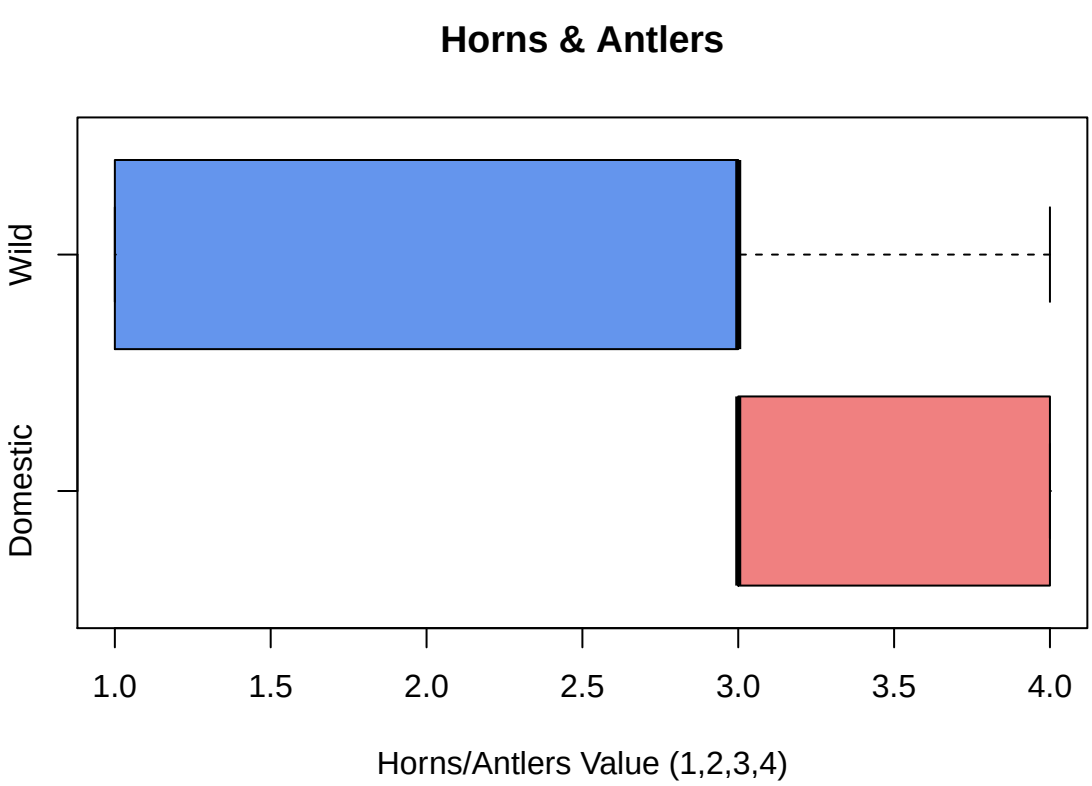


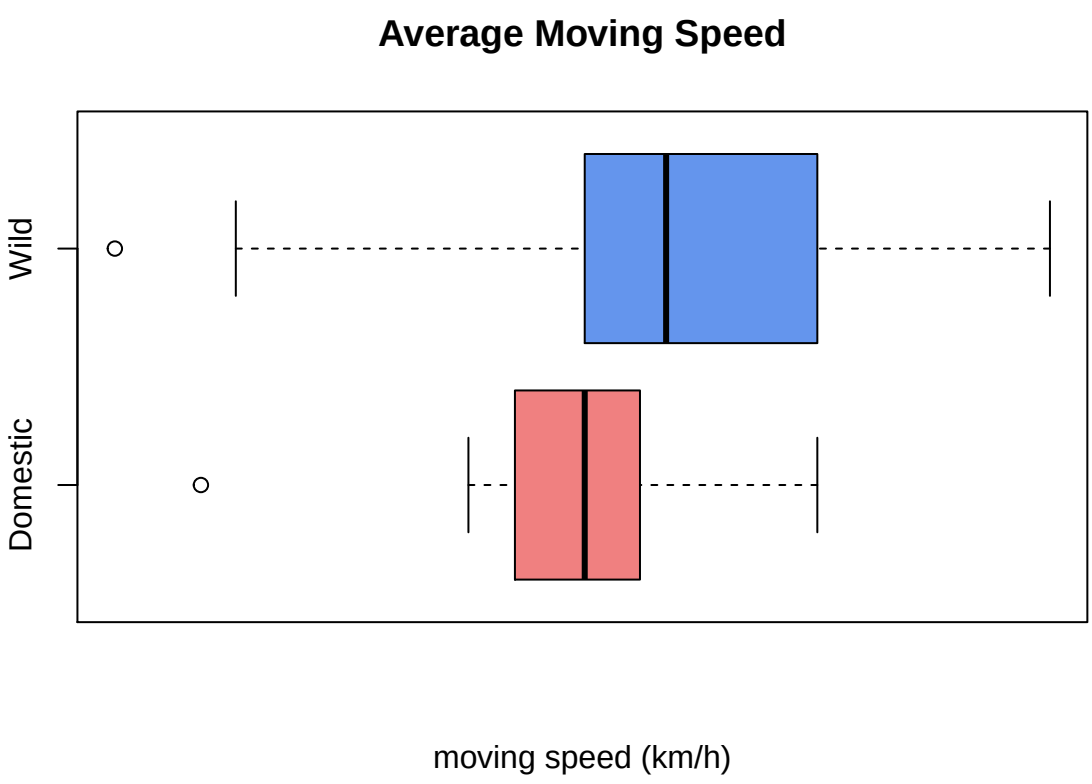
7 Plots

The results are visualized using different plots.

```
source("Plots.R")
```


Bodymass
Wild
Domestic
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
bodymass (grams)


Horns & Antlers
Wild
Domestic
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Horns/Antlers Value (1,2,3,4)


Average Moving Speed
Wild
Domestic
moving speed (km/h)