10. Applying random forest on field data - gene

Fay

2022-11-04

Aim:

- Applying the models established in the script: 9
- How are hybrid mice different to the parental species?

Load necessary libraries:

```
#install.packages("optimx", version = "2021-10.12") # this package is required for
#the parasite load package to work
library(tidyverse)
library(tidyr)
library(dplyr)
library(cowplot)
library(randomForest)
library(ggplot2)
library(VIM) # visualizing missing data
library(mice) # imputing missing data without predictors
library(ggpubr)
library(optimx)
library(rfUtilities) # Implements a permutation test cross-validation for
# Random Forests models
library(mice) #imputations
library(fitdistrplus) #testing distributions
library(logspline)
library(caret)
```

Field data

Import field data

```
hm <- read.csv("output_data/2.imputed_MICE_data_set.csv")
```

Clean data

```
Field <- hm %>%
  filter(origin == "Field") %>%
    drop_na(HI)
```

We have 1921 mice in total.

```
EqPCR.cols      <- c("delta_ct_cewe_MminusE", "MC.Eimeria", "Ct.Eimeria") #,"Ct.Mus""delta_ct_ilwe_Mmin

Genes_wild   <- c("IFNy", "CXCR3", "IL.6", "IL.13", #"IL.10",
                  "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                  "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                  "TICAM1", "TNF") #, "IL.12", "IRG6")
```

Prepare vectors for selecting

Actual Cleaning

```
#select the imputed gene columns
gene <-  Field %>%
  dplyr::select(c(Mouse_ID, "IFNy", "CXCR3", "IL.6", "IL.13", #"IL.10",
                  "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                  "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                  "TICAM1", "TNF"))

genes <- gene %>%
  dplyr::select(-Mouse_ID)

#remove rows with only nas
genes <- genes[,colSums(is.na(genes))<nrow(genes)]

#remove colums with only nas
genes <- genes[rowSums(is.na(genes)) != ncol(genes), ]

# select the same rows from the gene data
gene <- gene[row.names(genes),]

# select the same rows from the field data
Field <- Field[row.names(genes),]
```

Predicting weight loss in our imputed field data

Start by making the predictions for the field data.

```
# load predicting weight loss model
weight_loss_predict <- readRDS("r_scripts/models/predict_WL.rds")

set.seed(540)


#The predict() function in R is used to predict the values based on the input data.
predictions_field <- predict(weight_loss_predict, genes)

#make the vector positive so that the distributions further down work
predictions_field <- predictions_field * (-1)

# assign test.data to a new object, so that we can make changes
result_field <- genes
```

```r
#add the new variable of predictions to the result object
result_field <- cbind(result_field, predictions_field)

# add it to the field data
Field <- cbind(Field, predictions_field)
```

It is time to apply the package of Alice Balard et al. on our predictions!

Let's see if we indeed have differences across the hybrid index with our predicted weight loss.

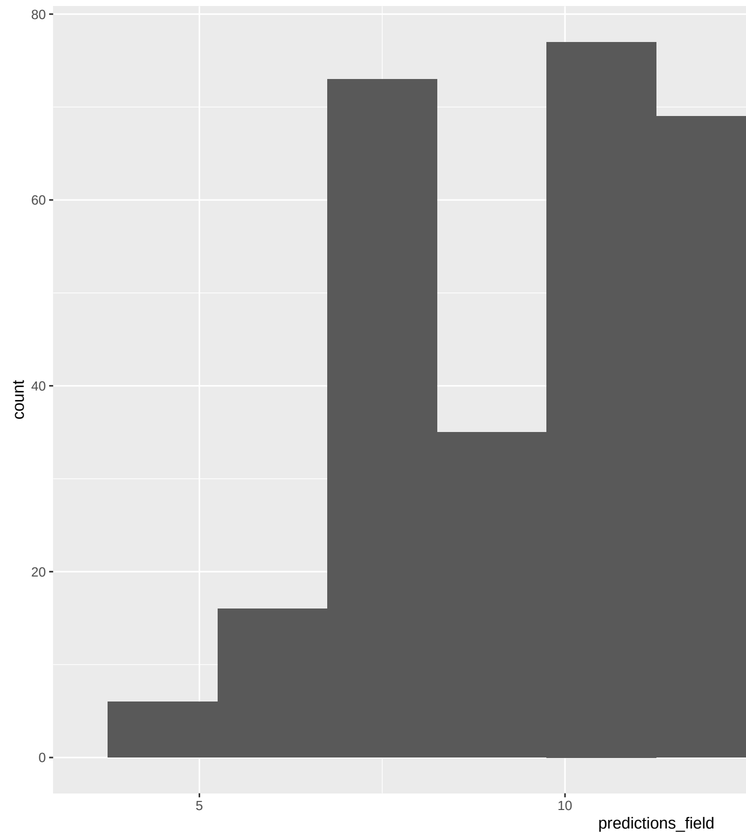
Install the package

```
##
## * checking for file '/tmp/Rtmpvb7qOk/remotes4d8407afeb8c6/alicebalard-parasiteLoad-1b43216/DESCRIPTI
## * preparing 'parasiteLoad':
## * checking DESCRIPTION meta-information ... OK
## * checking for LF line-endings in source and make files and shell scripts
## * checking for empty or unneeded directories
## * building 'parasiteLoad_0.1.0.tar.gz'
```

Data diagnostics

Visualizations

```r

Field %>% ggplot(aes(x = predictions_field)) +
  geom_histogram(binwidth = 1.5)
```



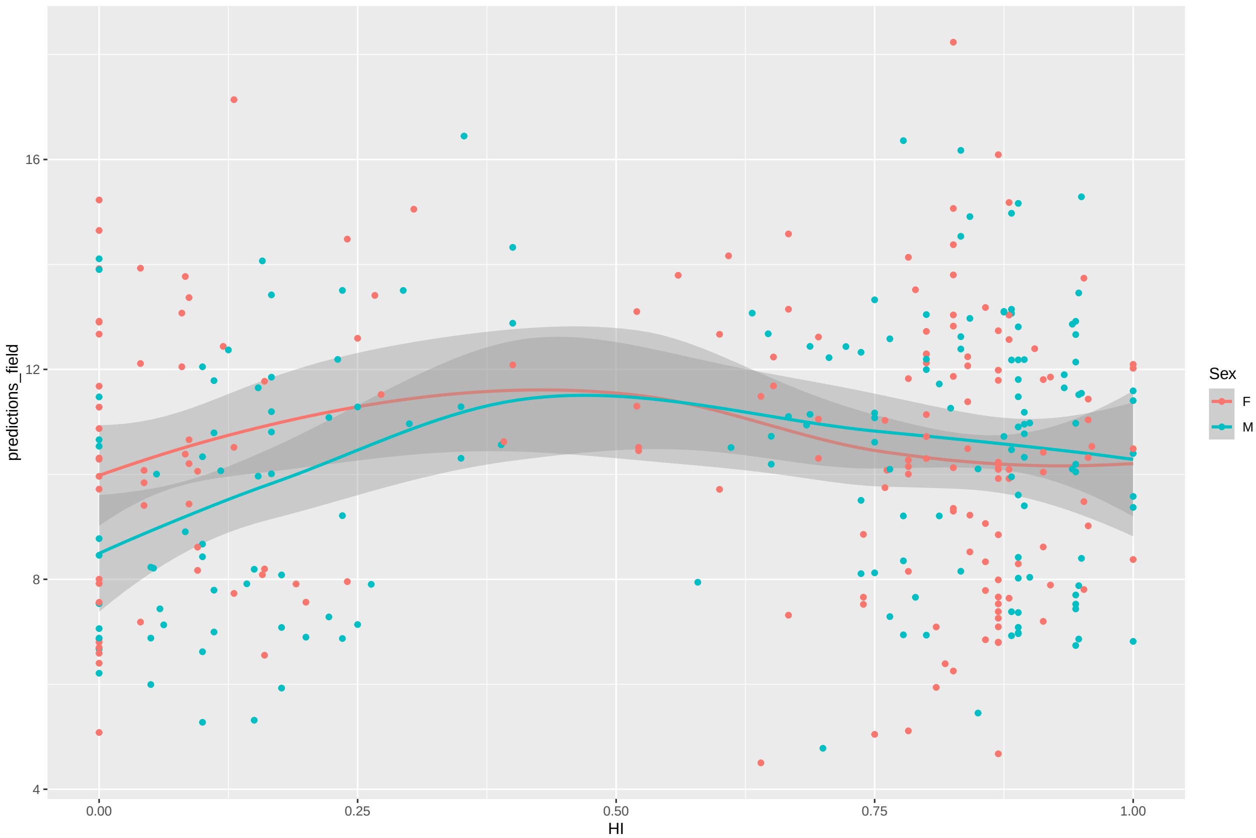
What is the distribution of the predicted weight loss?

Rough graph of our predictions against the hybrid index and against the

```
Field %>%
    ggplot(aes(x = HI , y = predictions_field , color = Sex)) +
    geom_smooth() +
    geom_point()
```

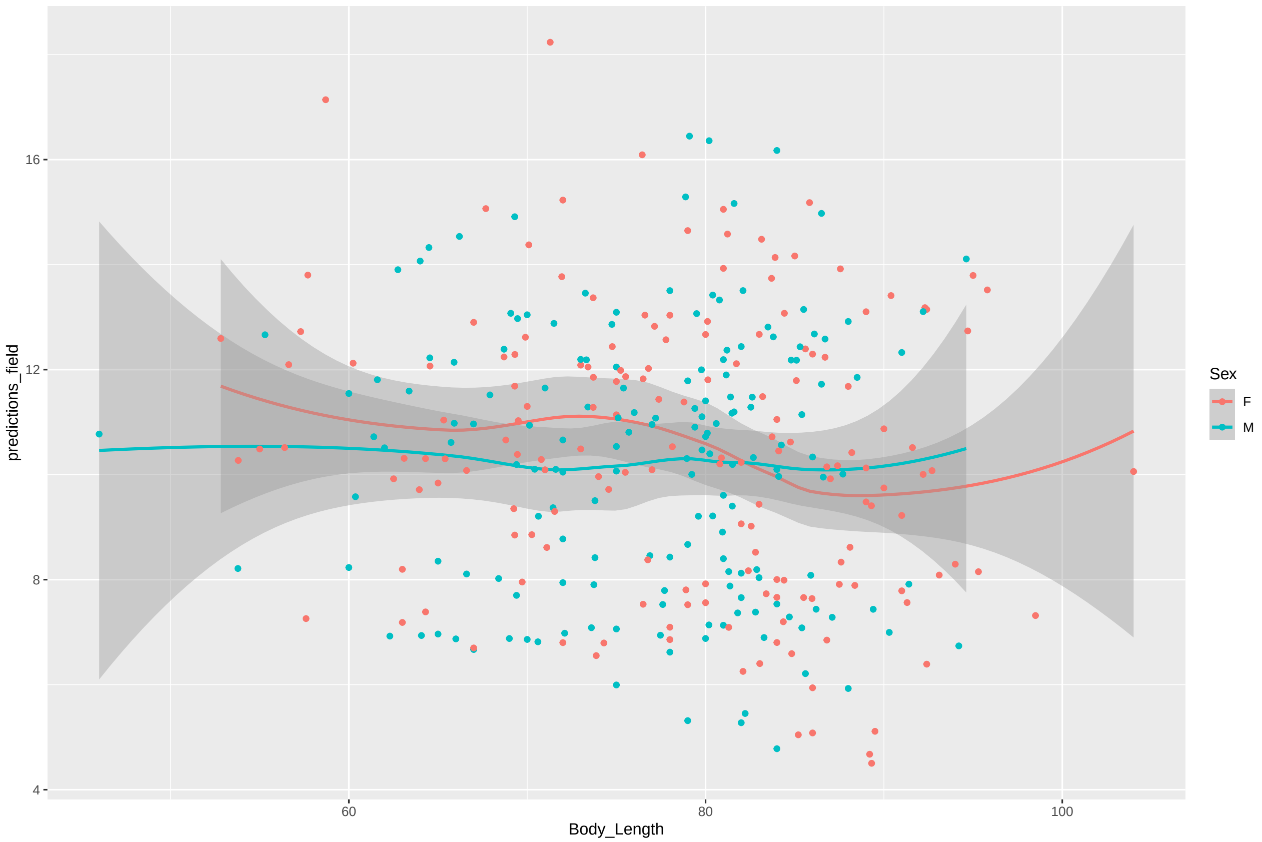
body length

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
Field %>%
    ggplot(aes(x = Body_Length , y = predictions_field , color = Sex)) +
    geom_smooth() +
    geom_point()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'

## Warning: Removed 1 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

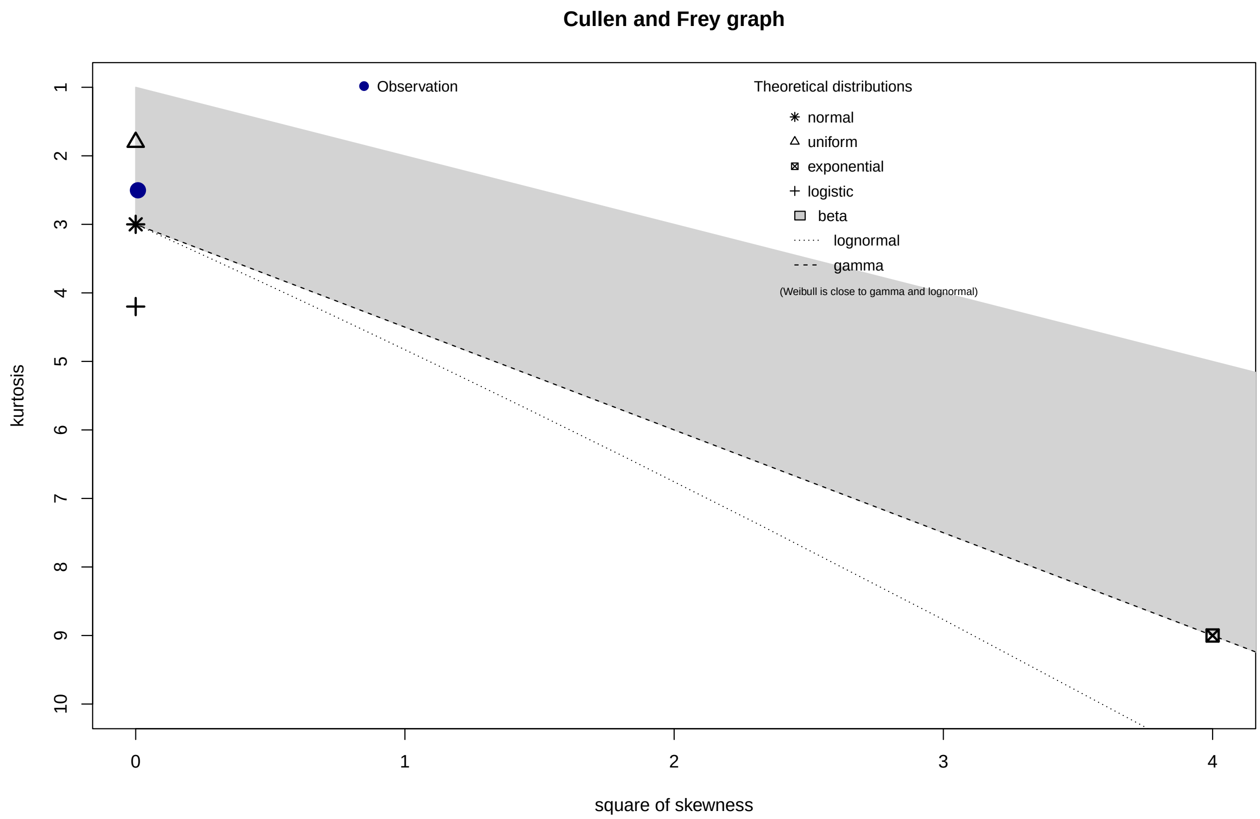


Fitting distributions??

Ratios / Percentages are not normally distributed. Weibull is a good distributions.

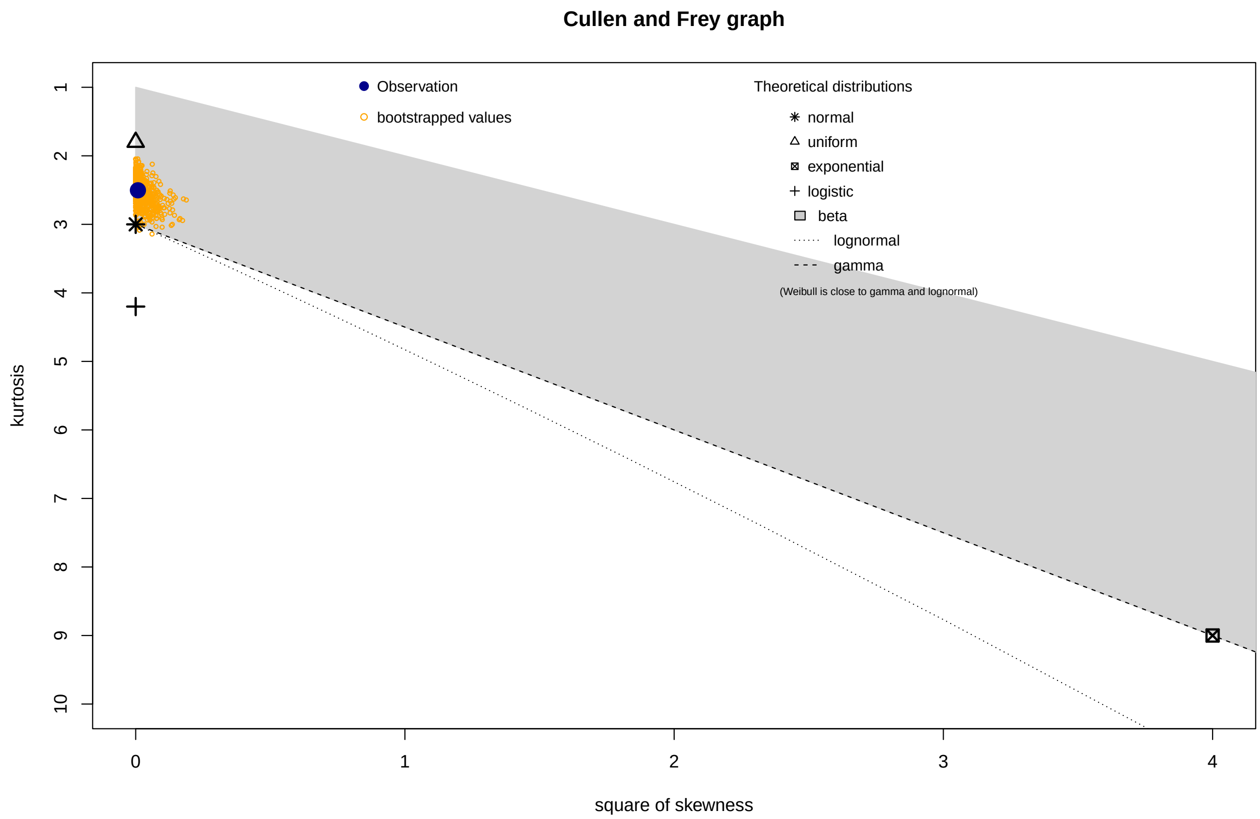
Alice used weibull for the qpcr data. (paper)

```
Field <- Field %>%
dplyr::mutate(WL = predictions_field)

x <- Field$WL

descdist(data = x, discrete = FALSE)
```




```
## summary statistics
## ------
## min:  4.502851   max:  18.23443
## median:  10.39781
## mean:  10.31462
## estimated sd:  2.595905
## estimated skewness:  0.09098517
## estimated kurtosis:  2.502986
```

```
descdist(data = x, discrete = FALSE, #data is continuous
         boot = 1000)
```




```
## summary statistics
## ------
## min:  4.502851   max:  18.23443
## median:  10.39781
## mean:  10.31462
## estimated sd:  2.595905
## estimated skewness:  0.09098517
## estimated kurtosis:  2.502986
```

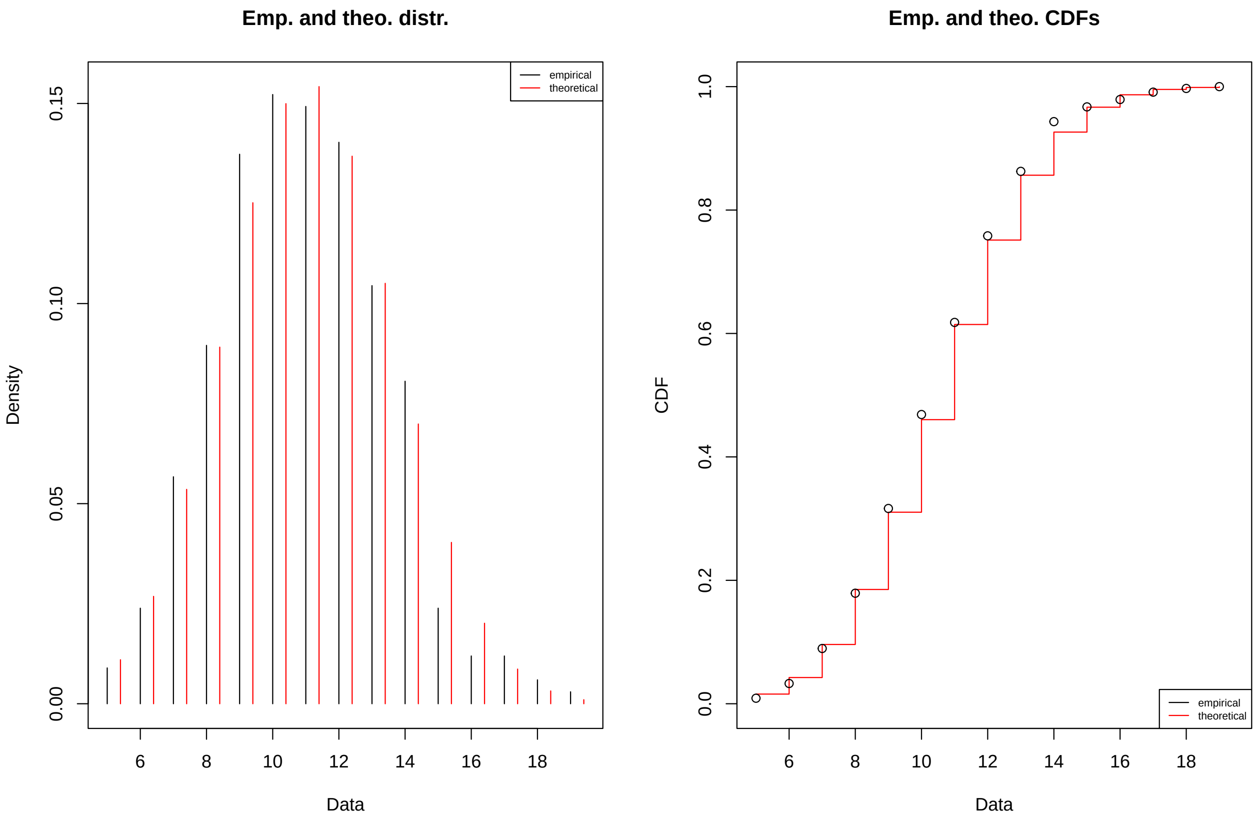
Test for binomial distribution

```
set.seed(10)
n = 25
size = 27
prob = .4
data = rbinom(x, size = size, prob = prob)
fit = fitdist(data = data, dist="binom",
                   fix.arg=list(size = size),
                   start=list(prob = 0.1))

summary(fit)
```

```
## Fitting of the distribution ' binom ' by maximum likelihood
## Parameters :
##      estimate  Std. Error
## prob 0.399558 0.005150141
## Fixed parameters:
##      value
```

```
## size    27
## Loglikelihood:  -779.317   AIC:  1560.634   BIC:  1564.448
```

```r
plot(fit)
```




```r
normal_ <- fitdist(x, "norm")
weibull_ <- fitdist(x, "weibull")
gamma_ <- fitdist(x, "gamma")


# Define function to be used to test, get the log lik and aic
tryDistrib <- function(x, distrib){
  # deals with fitdistr error:
  fit <- tryCatch(MASS::fitdistr(x, distrib), error=function(err) "fit failed")
  return(list(fit = fit,
              loglik = tryCatch(fit$loglik, error=function(err) "no loglik computed"),
              AIC = tryCatch(fit$aic, error=function(err) "no aic computed")))
}



findGoodDist <- function(x, distribs, distribs2){
  l =lapply(distribs, function(i) tryDistrib(x, i))
  names(l) <- distribs
  print(l)
  listDistr <- lapply(distribs2, function(i){
    if (i %in% "t"){
```

```
      fitdistrplus::fitdist(x, i, start = list(df =2))
    } else {
      fitdistrplus::fitdist(x,i)
    }}
  )
  par(mfrow=c(2,2))
  denscomp(listDistr, legendtext=distribs2)
  cdfcomp(listDistr, legendtext=distribs2)
  qqcomp(listDistr, legendtext=distribs2)
  ppcomp(listDistr, legendtext=distribs2)
  par(mfrow=c(1,1))
}
```

```
tryDistrib(x, "normal")
```

Functions for testing distributions

```
## $fit
##       mean          sd
##   10.3146177    2.5920279
##  ( 0.1416176) ( 0.1001388)
##
## $loglik
## [1] -794.412
##
## $AIC
## NULL
```

```
tryDistrib(x, "binomial")
```

```
## $fit
## [1] "fit failed"
##
## $loglik
## [1] "no loglik computed"
##
## $AIC
## [1] "no aic computed"
```

```
tryDistrib(x, "student")
```

```
## $fit
## [1] "fit failed"
##
## $loglik
## [1] "no loglik computed"
##
## $AIC
## [1] "no aic computed"
```

```
tryDistrib(x, "weibull")
```

```
## $fit
##      shape        scale
##    4.4087002   11.3158846
##  ( 0.1846660) ( 0.1480956)
```

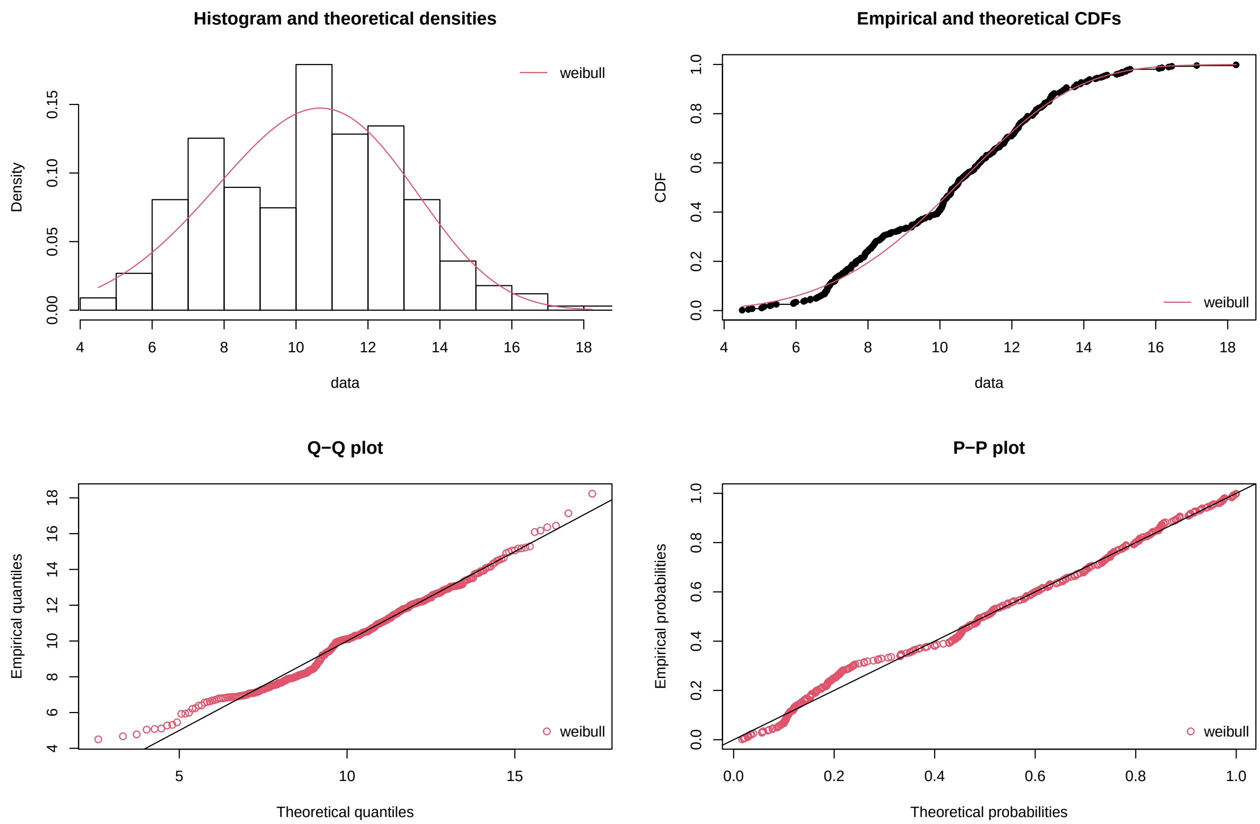
```
##
## $loglik
## [1] -795.0978
##
## $AIC
## NULL
```

```r
tryDistrib(x, "weibullshifted")
```

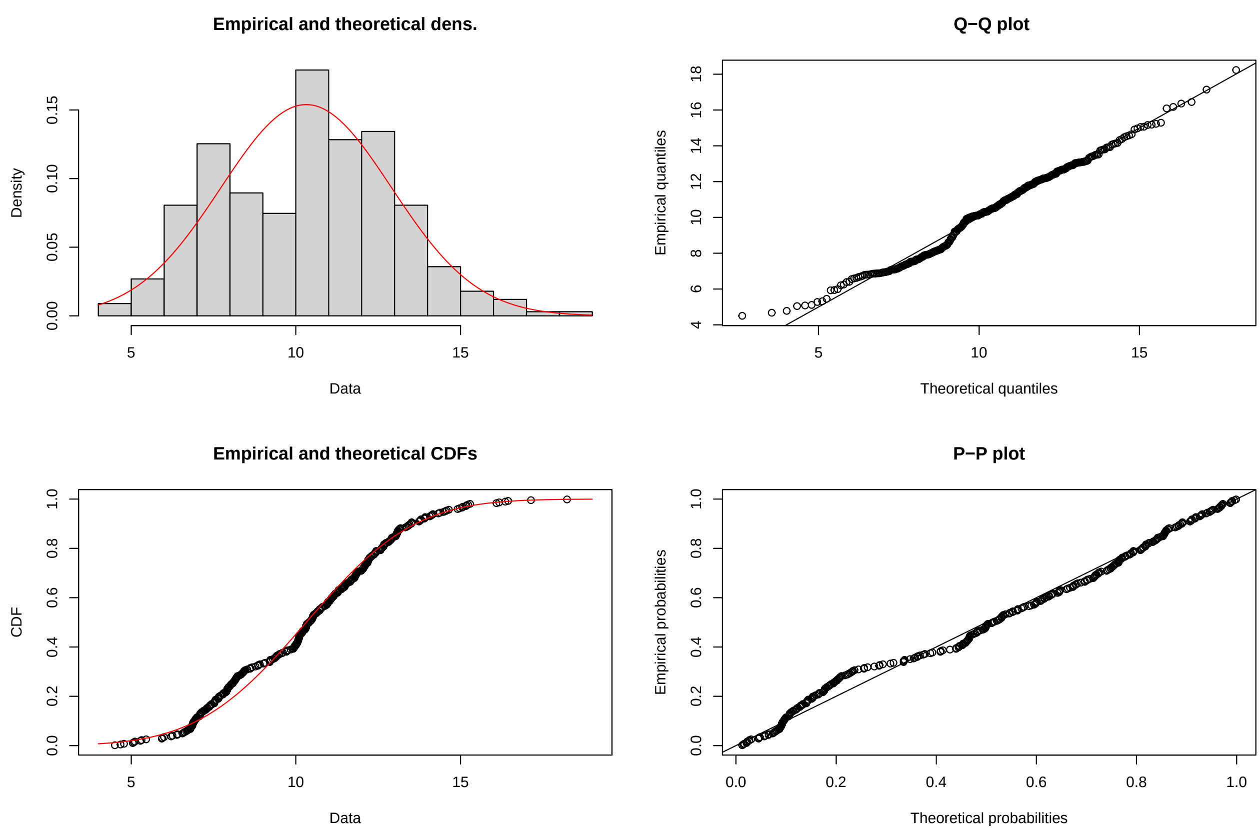
```
## $fit
## [1] "fit failed"
##
## $loglik
## [1] "no loglik computed"
##
## $AIC
## [1] "no aic computed"
```

```r
findGoodDist(x, "normal", "weibull")
```

```
## $normal
## $normal$fit
##       mean          sd
##   10.3146177    2.5920279
##  ( 0.1416176) ( 0.1001388)
##
## $normal$loglik
## [1] -794.412
##
## $normal$AIC
## NULL
```


Histogram and theoretical densities
weibull
Density
data
Empirical and theoretical CDFs
CDF
data
weibull
Q–Q plot
Empirical quantiles
Theoretical quantiles
weibull
P–P plot
Empirical probabilities
Theoretical probabilities
weibull

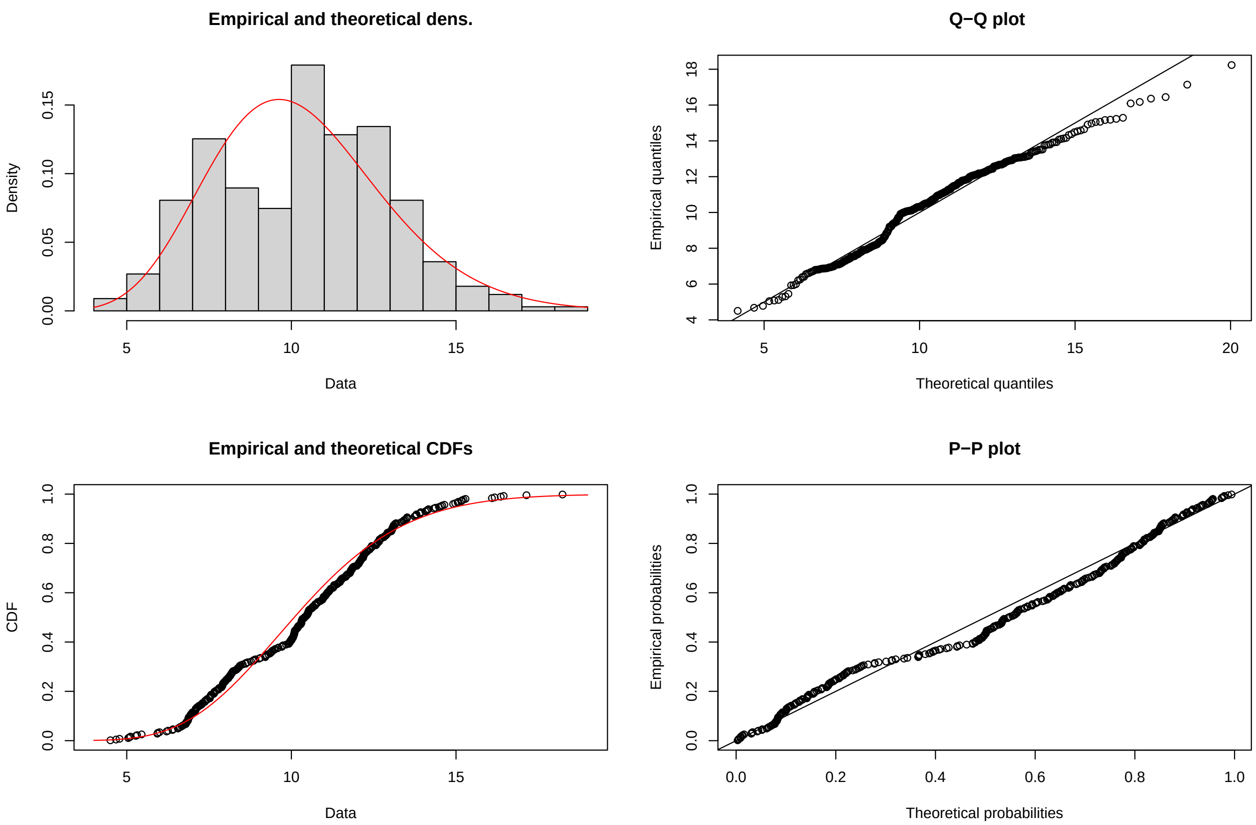
```
plot(normal_)
```


Empirical and theoretical dens.
Density
Data
Q–Q plot
Empirical quantiles
Theoretical quantiles
Empirical and theoretical CDFs
CDF
Data
P–P plot
Empirical probabilities
Theoretical probabilities

```r
summary(normal_)
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##       estimate Std. Error
## mean 10.314618  0.1416176
## sd    2.592028  0.1001387
## Loglikelihood:  -794.412   AIC:  1592.824   BIC:  1600.452
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1
```

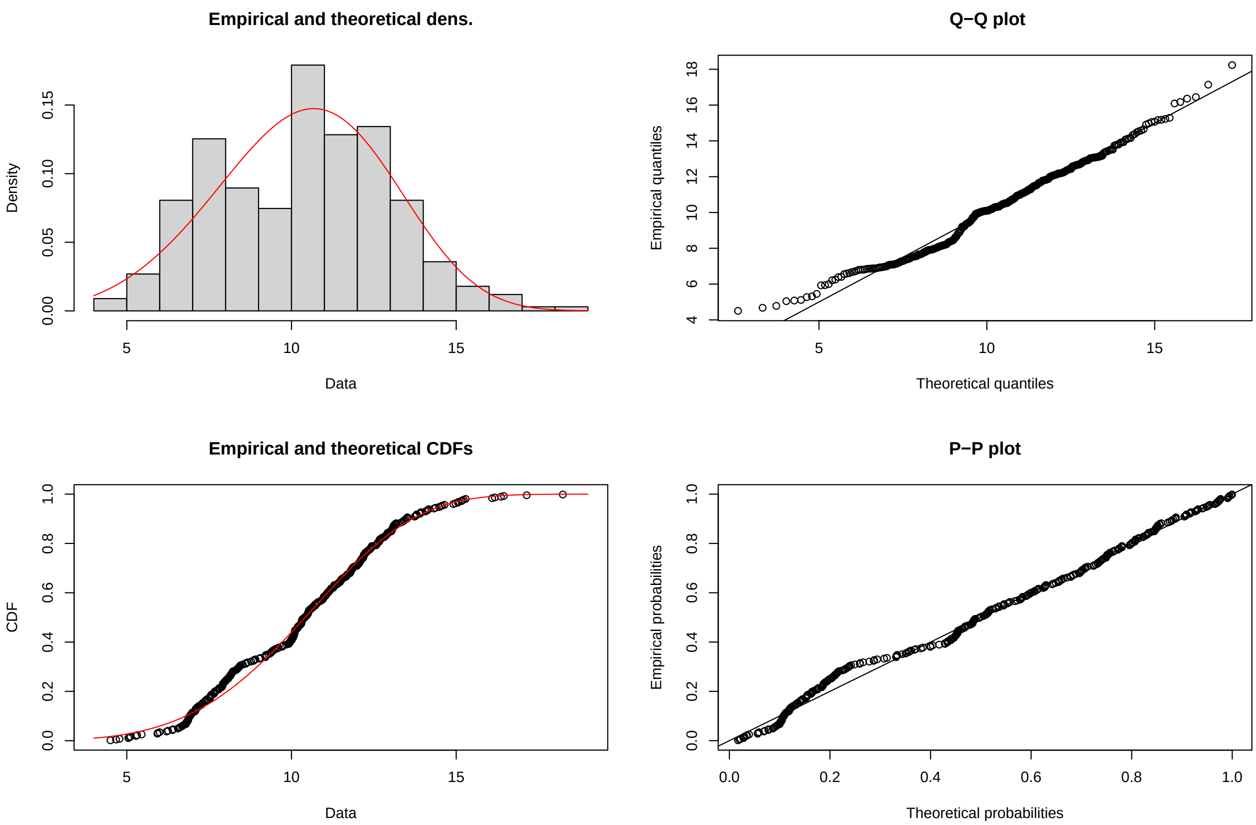
```r
plot(gamma_)
```




```r
summary(gamma_)
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## shape 14.981244  1.1448856
## rate   1.452465  0.1128767
## Loglikelihood:  -796.0754   AIC:  1596.151   BIC:  1603.779
## Correlation matrix:
##           shape      rate
## shape 1.0000000 0.9833656
## rate  0.9833656 1.0000000
```

```r
plot(weibull_)
```




```r
summary(weibull_)
```

```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## shape  4.408685  0.1846641
## scale 11.316386  0.1481056
## Loglikelihood:  -795.0978   AIC:  1594.196   BIC:  1601.824
## Correlation matrix:
##           shape     scale
## shape 1.0000000 0.3211869
## scale 0.3211869 1.0000000
```

Is alpha significant for each hypothesis?

```r
Field$Sex <- as.factor(Field$Sex)


parasiteLoad::getParamBounds("normal", data = Field, response = "WL")
```

```
##      L1start         L1LB         L1UB      L2start         L2LB         L2UB
## 10.314617691  4.502850517 18.234429241 10.314617691  4.502850517 18.234429241
##   alphaStart      alphaLB      alphaUB    mysdStart       mysdLB       mysdUB
##  0.000000000 -5.000000000  5.000000000  1.000000000  0.000000001 10.000000000
```

```r
speparam <- c(L1start = 10,
                     L1LB = 1e-9,
                     L1UB = 20,
                     L2start = 10,
                     L2LB = 1e-9,
                     L2UB = 20,
                     alphaStart = 0, alphaLB = -5, alphaUB = 5,
                     myshapeStart = 1, myshapeLB = 1e-9, myshapeUB = 5)

##All
fitWL_Sex <- parasiteLoad::analyse(data = Field,
                        response = "WL",
                        model = "normal",
                        group = "Sex")
```

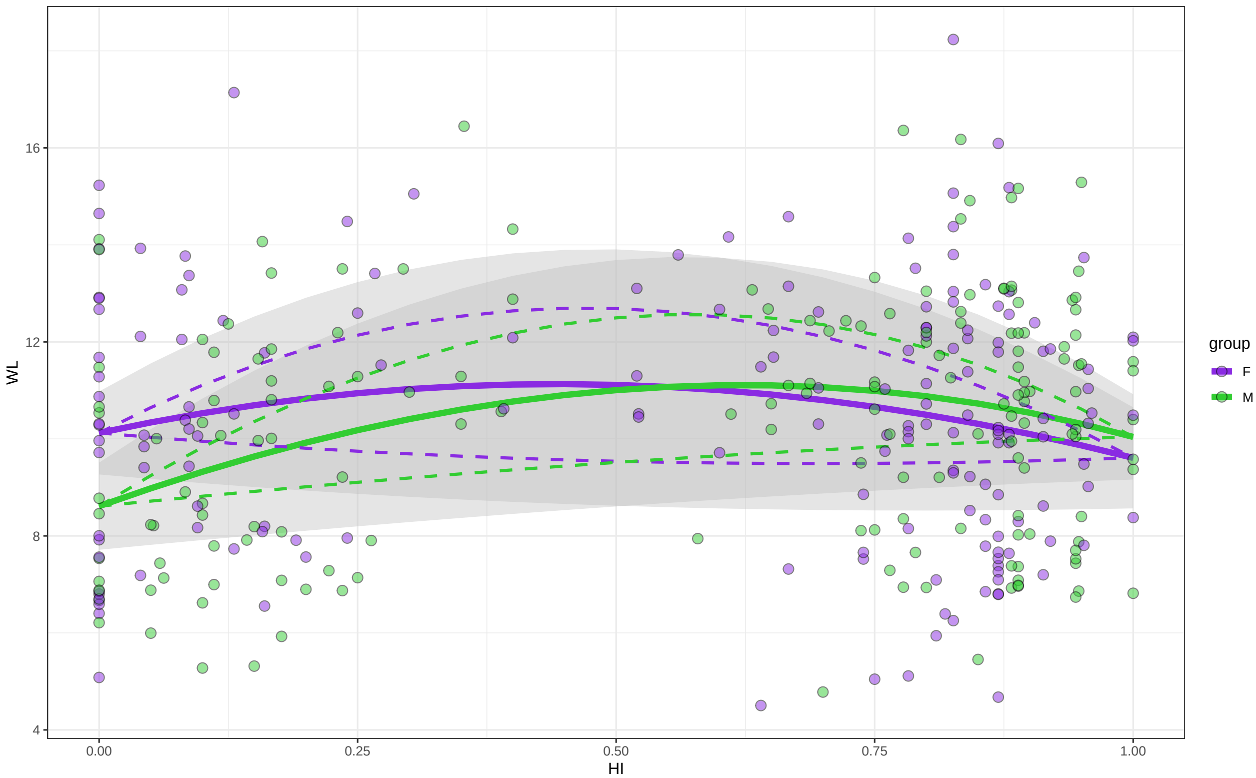
```
## [1] "Analysing data for response: WL"
## [1] "Fit for the response: WL"
## [1] "Fitting for all"
## [1] "Fitting model basic without alpha"
## [1] "Did converge"
## [1] "Fitting model basic with alpha"
## [1] "Did converge"
## [1] "Fitting model advanced without alpha"
## [1] "Did converge"
## [1] "Fitting model advanced with alpha"
## [1] "Did converge"
## [1] "Fitting for groupA : F"
## [1] "Fitting model basic without alpha"
## [1] "Did converge"
## [1] "Fitting model basic with alpha"
## [1] "Did converge"
## [1] "Fitting model advanced without alpha"
## [1] "Did converge"
## [1] "Fitting model advanced with alpha"
## [1] "Did converge"
## [1] "Fitting for groupB : M"
## [1] "Fitting model basic without alpha"
## [1] "Did converge"
## [1] "Fitting model basic with alpha"
## [1] "Did converge"
## [1] "Fitting model advanced without alpha"
## [1] "Did converge"
## [1] "Fitting model advanced with alpha"
## [1] "Did converge"
## [1] "Testing H0 no alpha vs alpha"
##   dLL dDF      pvalue
## 1 3.5   1 0.00819461
## [1] "Testing H1 no alpha vs alpha"
##    dLL dDF     pvalue
## 1 2.75   1 0.01906781
## [1] "Testing H2 groupA no alpha vs alpha"
##    dLL dDF    pvalue
## 1 0.99   1 0.1602889
## [1] "Testing H2 groupB no alpha vs alpha"
```

```
##    dLL dDF     pvalue
## 1 2.87   1 0.01649572
## [1] "Testing H3 groupA no alpha vs alpha"
##    dLL dDF    pvalue
## 1 1.33   1 0.1026051
## [1] "Testing H3 groupB no alpha vs alpha"
##    dLL dDF     pvalue
## 1 2.82   1 0.01760955
## [1] "Testing H1 vs H0"
##    dLL dDF    pvalue
## 1 0.77   1 0.2155243
## [1] "Testing H2 vs H0"
##    dLL dDF    pvalue
## 1 0.66   3 0.7262847
## [1] "Testing H3 vs H1"
##    dLL dDF     pvalue
## 1 4.29   4 0.07257475
## [1] "Testing H3 vs H2"
##   dLL dDF     pvalue
## 1 4.4   2 0.01228134
```

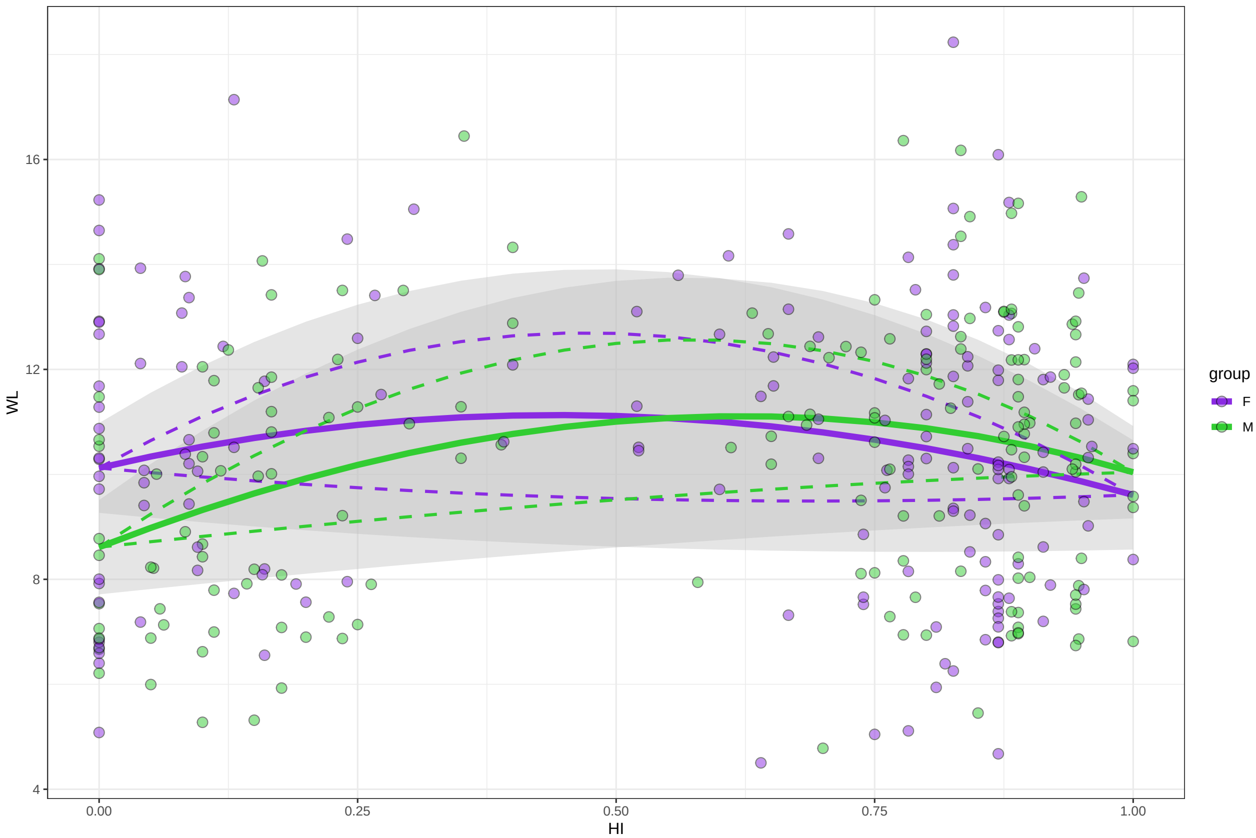
```
plot_WL_Sex<- bananaPlot(mod = fitWL_Sex$H3,
             data = Field,
             response = "WL",
             group = "Sex") +
    scale_fill_manual(values = c("blueviolet", "limegreen")) +
  scale_color_manual(values = c("blueviolet", "limegreen")) +
  theme_bw()
```

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

```
# Create HI bar
HIgradientBar <- ggplot(data.frame(hi = seq(0,1,0.0001)),
                        aes(x=hi, y=1, fill = hi)) +
  geom_tile() +
  theme_void() +
  scale_fill_gradient(low = "blue", high = "red")  +
  scale_x_continuous(expand=c(.01,0)) +
  scale_y_continuous(expand=c(0,0)) +
  theme(legend.position = 'none')

plot_grid(plot_WL_Sex,
          HIgradientBar,
          nrow = 2,
          align = "v",
          axis = "tlr",
          rel_heights = c(13, 1))
```



```
plot_WL_Sex
```



H0: the expected load for the subspecies and between 2 groups is the same

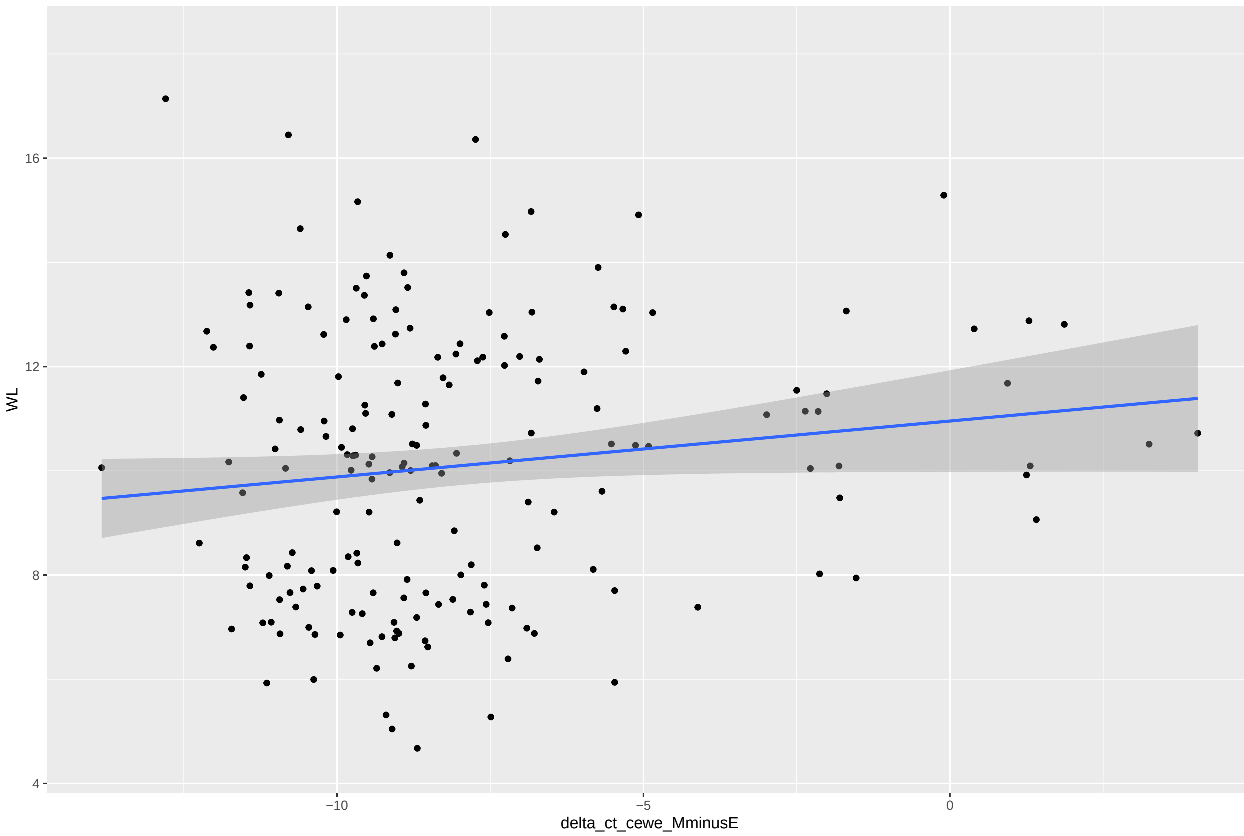
H1: the mean load across 2 groups is the same, but can differ across subspecies

H2: the mean load across subspecies is the same, but can differ between the 2 groups

H3: the mean load can differ both across subspecies and between 2 groups

```
ggplot(data = Field, aes(x = delta_ct_cewe_MminusE, y = WL)) +
  geom_point() +
  stat_smooth(method= "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 150 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 150 rows containing missing values (`geom_point()`).
```




```
Field2 <- Field %>%
  drop_na(delta_ct_cewe_MminusE)

cor(Field2$WL, Field2$delta_ct_cewe_MminusE)
```

```
## [1] 0.1372538
```

```
tolerance <- lm(WL ~  delta_ct_cewe_MminusE, data = Field)


summary(tolerance)
```

```
##
## Call:
## lm(formula = WL ~ delta_ct_cewe_MminusE, data = Field)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3475 -2.2227  0.0849  1.9078  7.5577
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           10.95537    0.49363  22.193   <2e-16 ***
## delta_ct_cewe_MminusE  0.10732    0.05725   1.874   0.0625 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.564 on 183 degrees of freedom
```

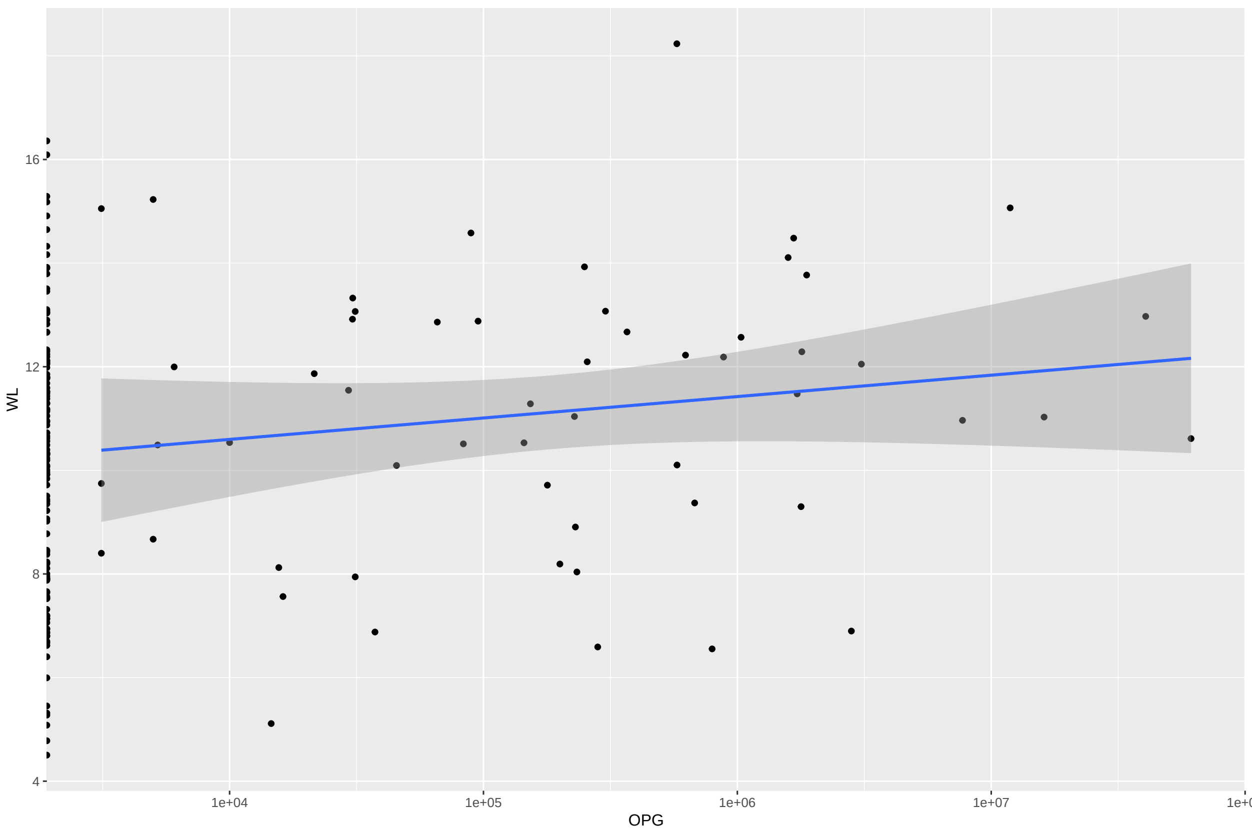
```
##   (150 observations deleted due to missingness)
## Multiple R-squared:  0.01884,    Adjusted R-squared:  0.01348
## F-statistic: 3.514 on 1 and 183 DF,  p-value: 0.06246
```

```
confint(tolerance)
```

```
##                              2.5 %     97.5 %
## (Intercept)             9.98142668 11.9293059
## delta_ct_cewe_MminusE -0.00564135  0.2202722
```

```
ggplot(data = Field, aes(x = OPG, y = WL)) +
  geom_point() +
  stat_smooth(method= "lm") +
  scale_x_log10()
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
## Transformation introduced infinite values in continuous x-axis

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

## Warning: Removed 280 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 157 rows containing missing values (`geom_point()`).
```




```
Field2 <- Field %>%
  drop_na(OPG)

cor(Field2$WL, Field2$OPG)
```

```
## [1] 0.0773176
```

```r
tolerance <- lm(WL ~  OPG, data = Field)


summary(tolerance)
```

```
##
## Call:
## lm(formula = WL ~ OPG, data = Field)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8889 -2.2039  0.1256  1.7549  7.8218
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.039e+01  2.022e-01  51.386   <2e-16 ***
## OPG         3.613e-08  3.512e-08   1.029    0.305
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.664 on 176 degrees of freedom
##   (157 observations deleted due to missingness)
## Multiple R-squared:  0.005978,   Adjusted R-squared:  0.0003302
## F-statistic: 1.058 on 1 and 176 DF,  p-value: 0.305
```

```r
confint(tolerance)
```

```
##                     2.5 %       97.5 %
## (Intercept)  9.992620e+00 1.079083e+01
## OPG         -3.318005e-08 1.054471e-07
```

```r
tolerance <- lm(WL ~  OPG * delta_ct_cewe_MminusE, data = Field)


summary(tolerance)
```

```
##
## Call:
## lm(formula = WL ~ OPG * delta_ct_cewe_MminusE, data = Field)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7231 -2.0827 -0.2738  2.0784  6.4227
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.187e+01  9.377e-01  12.657   <2e-16 ***
## OPG                       -2.170e-05  2.427e-05  -0.894   0.3759
## delta_ct_cewe_MminusE      2.497e-01  1.228e-01   2.033   0.0479 *
## OPG:delta_ct_cewe_MminusE  4.484e-06  7.110e-06   0.631   0.5314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.671 on 45 degrees of freedom
##   (286 observations deleted due to missingness)
```

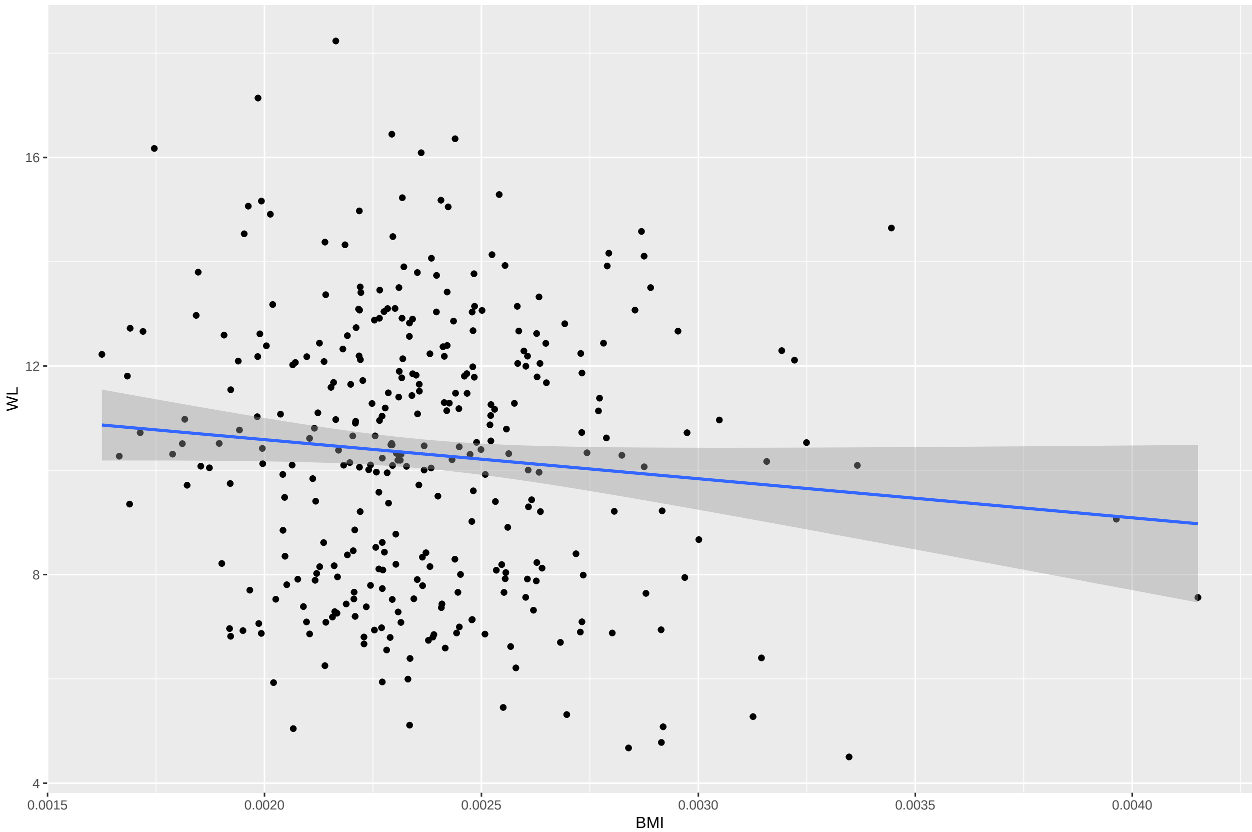
```
## Multiple R-squared:  0.1034, Adjusted R-squared:  0.04363
## F-statistic:  1.73 on 3 and 45 DF,  p-value: 0.1744
```

```
confint(tolerance)
```

```
##                                  2.5 %       97.5 %
## (Intercept)                9.979958e+00 1.375732e+01
## OPG                       -7.057859e-05 2.717201e-05
## delta_ct_cewe_MminusE      2.376044e-03 4.969401e-01
## OPG:delta_ct_cewe_MminusE -9.835569e-06 1.880448e-05
```

```
Field <- Field %>%
  dplyr::mutate(BMI = Body_Weight / (Body_Length)^2)

ggplot(data = Field, aes(x = BMI, y = WL)) +
  geom_point() +
  stat_smooth(method= "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 1 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```




```
bmi <- lm(WL ~ BMI, data = Field)

cor(Field$BMI, Field$WL, use = "complete.obs")
```

```
## [1] -0.09662733
```

```r
summary(bmi)
```

```
##
## Call:
## lm(formula = WL ~ BMI, data = Field)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4928 -2.2350  0.1203  1.8717  7.7695
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   12.087      1.014  11.921   <2e-16 ***
## BMI         -749.430    423.667  -1.769   0.0778 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.591 on 332 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.009337,   Adjusted R-squared:  0.006353
## F-statistic: 3.129 on 1 and 332 DF,  p-value: 0.07783
```

```r
confint(bmi)
```

```
##                  2.5 %   97.5 %
## (Intercept)    10.0924 14.08150
## BMI         -1582.8412 83.98102
```