Modelling

2022-04-04

Exercise 1

(a)

1.1.1 Examining the Data

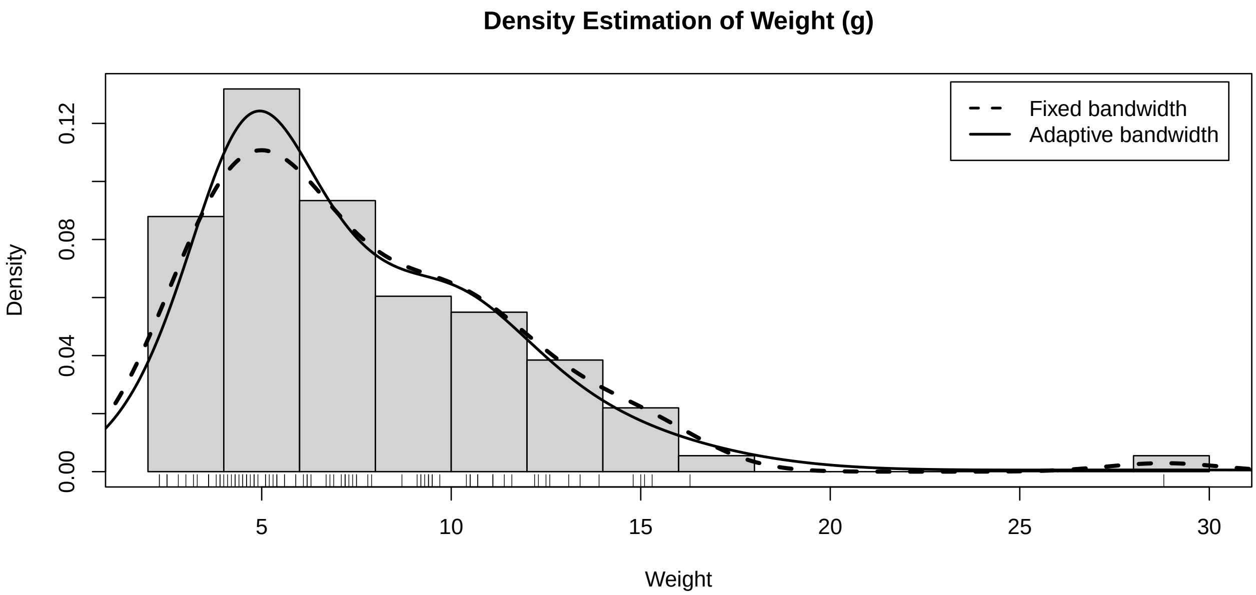
As a first step, we obtain summary statistics for the variables in `Darts.csv`:

```
suppressMessages(library(tidyverse))
suppressMessages(library(magrittr))
suppressMessages(library(car))
Darts= read.csv('Darts.csv') %>% as_tibble()
Darts$Name %<>% as.factor()
summary(Darts)
```

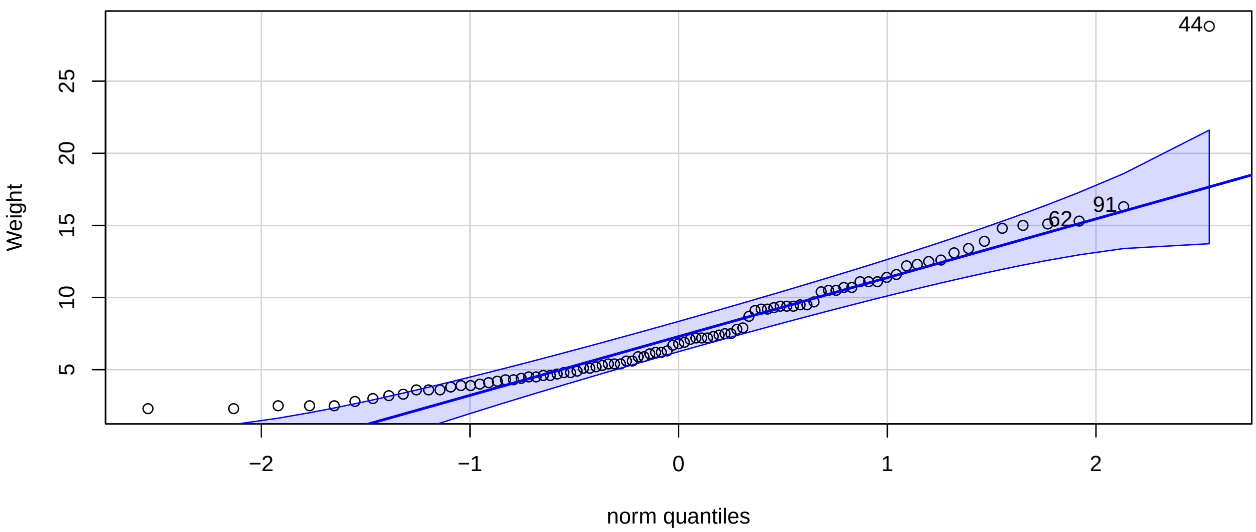
```
##      Weight           Length           Width          Thickness
##  Min.   : 2.300   Min.   : 30.60   Min.   :14.50   Min.   : 4.000
##  1st Qu.: 4.550   1st Qu.: 40.85   1st Qu.:18.55   1st Qu.: 6.250
##  Median : 6.800   Median : 47.10   Median :21.10   Median : 7.200
##  Mean   : 7.643   Mean   : 49.33   Mean   :22.08   Mean   : 7.271
##  3rd Qu.:10.050   3rd Qu.: 55.80   3rd Qu.:25.15   3rd Qu.: 8.250
##  Max.   :28.800   Max.   :109.50   Max.   :49.30   Max.   :10.700
##          Name
##  Darl      :28
##  Ensor     :10
##  Pedernales:32
##  Travis    :11
##  Wells     :10
##
```

Secondly, we view distribution of the variable `Weight`:

```
with(Darts, {
  hist(Weight, freq=FALSE, breaks="FD", main="Density Estimation of Weight (g)")
  lines(density(Weight, from=0), lwd=3, lty=2)
  lines(adaptiveKernel(Weight, from=0), lwd=2, lty=1)
  rug(Weight)
  legend("topright", c("Fixed bandwidth", "Adaptive bandwidth"),
         lty=2:1, lwd=2, inset=.02)
  box()
})
```




```r
qqPlot(~ Weight, data=Darts, id=list(n=3))
```

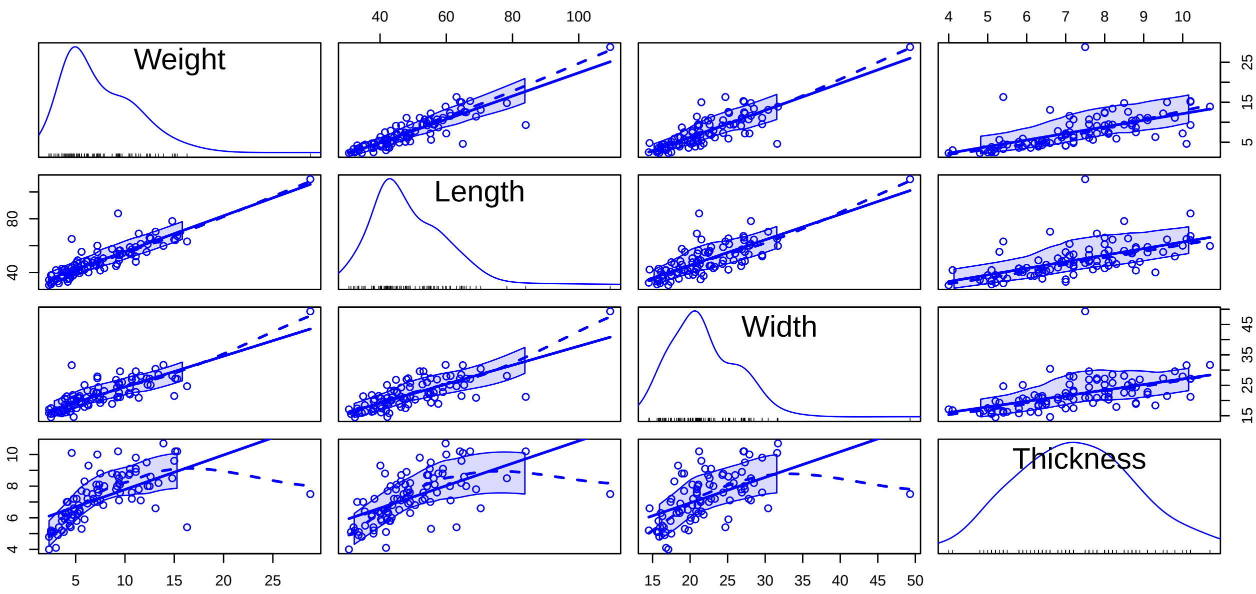



```
## [1] 44 91 62
```

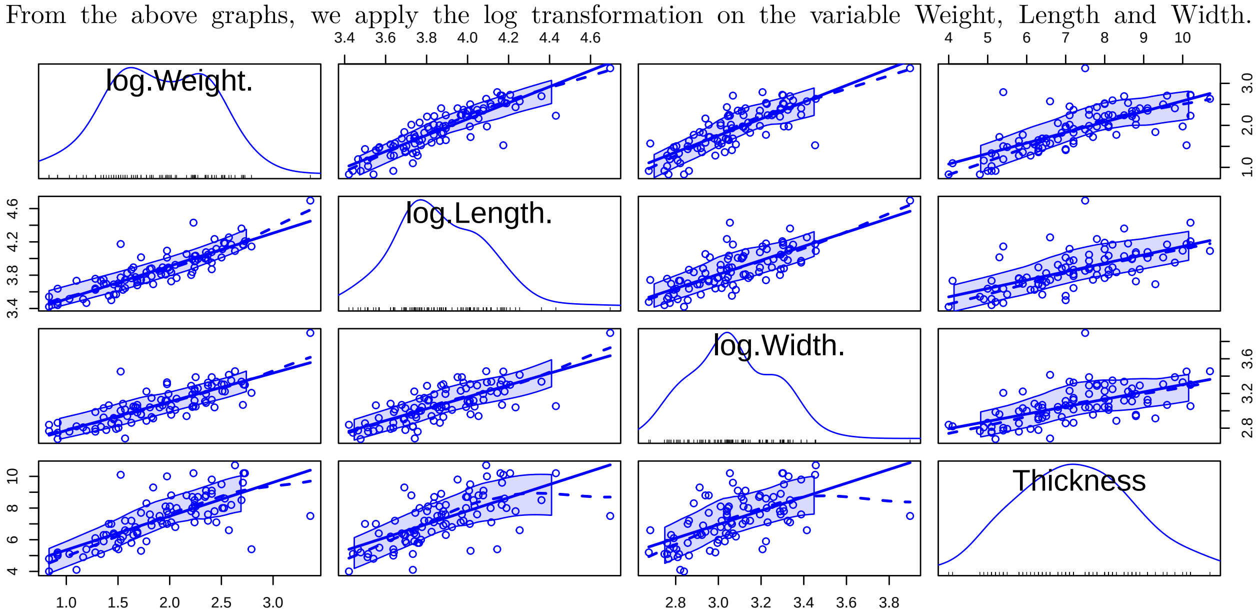
Both nonparametric density estimates and the histogram suggest a mode at around 5g, and all show that the distribution of weight is right- skewed. The fixed-bandwidth kernel estimate has more wiggle at the right where data are sparse, and the histogram is rough in this region, while the adaptive- kernel estimator is able to smooth out the density estimate in the low-density region. And because many points, especially at the left of the graph, are outside the confidence bounds, we have evidence that the distribution of weight is not like a sample from a normal population.

Then, we use scatterplots to provide summaries of the conditional distribution of a numeric response variable given a numeric predictor. The `scatterplotMatrix()` function produces scatterplots for all paris of numeric variables.

```r
scatterplotMatrix(~ Weight + Length + Width + Thickness, data= Darts)
```

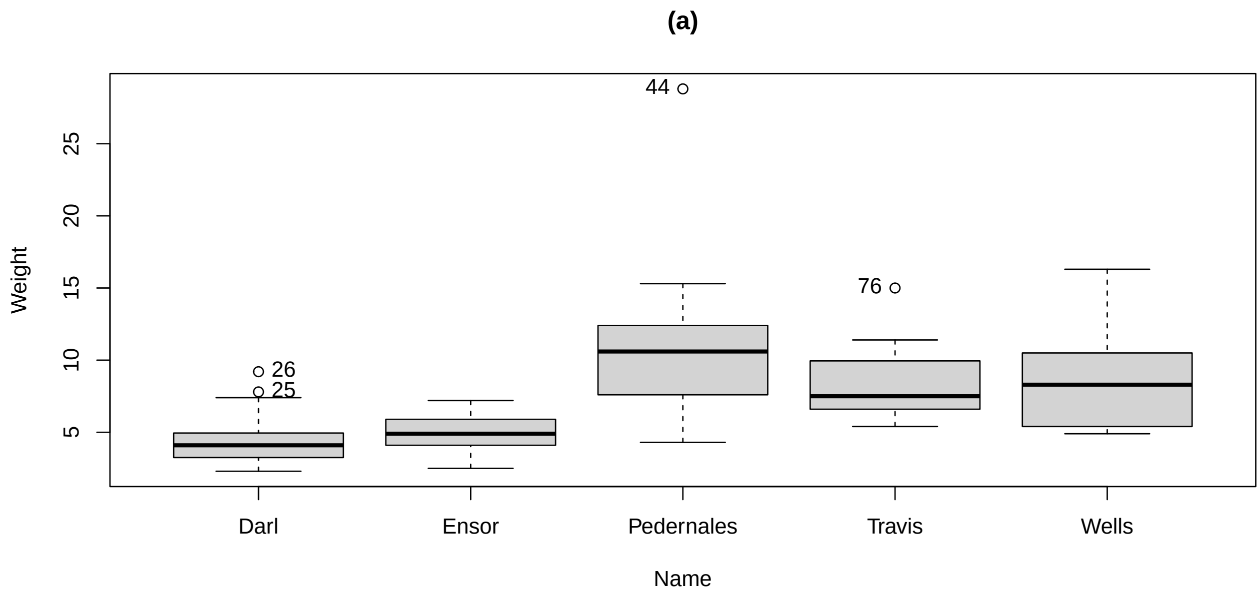


From the above graphs, we apply the log transformation on the variable Weight, Length and Width.



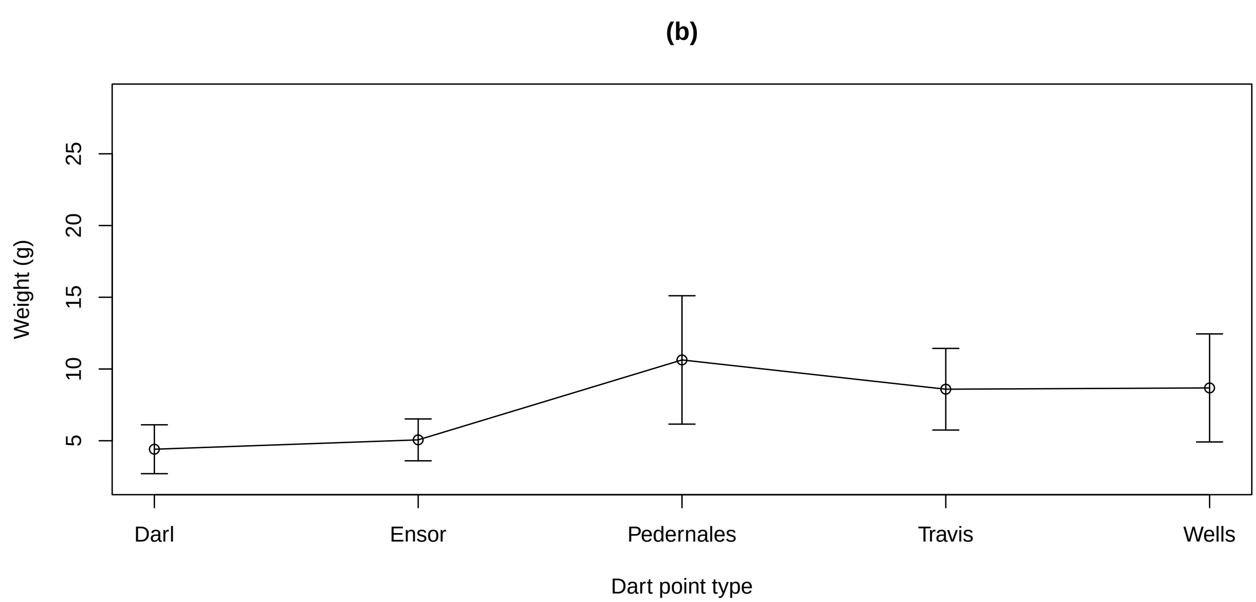
We can further explore the relationship between Weight and Name in the Darts using parallel boxplots.

```r
Boxplot(Weight ~ Name, data=Darts, main="(a)")
```




```
## [1] "25" "26" "44" "76"
```

```
library(plotrix)
means= Tapply(Weight ~ Name, mean, data=Darts)
sds= Tapply(Weight ~ Name, sd, data=Darts)
{plotCI(1:5, means, sds, xaxt="n", xlab="Dart point type",
   ylab="Weight (g)", main="(b)",
   ylim=range(Darts$Weight))
lines(1:5, means)
axis(1, at=1:5, labels = names(means))}
```



1.1.2 Regression Analysis

We use the`lm()` function to fit a linear regression model to the data:

```r
(model_full <- lm(log(Weight) ~ log(Length) + log(Width) + Thickness + Name, data=Darts)) %>% summary()
```

```
##
## Call:
## lm(formula = log(Weight) ~ log(Length) + log(Width) + Thickness +
##     Name, data = Darts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.17289 -0.08479  0.01706  0.11497  0.56577
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -5.04406    0.50942  -9.902 1.03e-15 ***
## log(Length)     1.00401    0.15590   6.440 7.34e-09 ***
## log(Width)      0.84842    0.19394   4.375 3.51e-05 ***
## Thickness       0.05734    0.02149   2.668  0.00918 **
## NameEnsor      -0.09834    0.08244  -1.193  0.23634
## NamePedernales  0.04303    0.08504   0.506  0.61421
## NameTravis      0.18590    0.08586   2.165  0.03325 *
## NameWells       0.07776    0.08767   0.887  0.37768
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.206 on 83 degrees of freedom
## Multiple R-squared:  0.8579, Adjusted R-squared:  0.8459
## F-statistic: 71.59 on 7 and 83 DF,  p-value: < 2.2e-16
```

We use stepwise regression to select best model for the variable Weight. We pass the full model to `step` function. It iteratively searches the full scope of variables in backwards directions by default, if scope is not given. It performs multiple iteractions by droping one X variable at a time. In each iteration, multiple models are built by dropping each of the X variables at a time. The AIC of the models is also computed and the model that yields the lowest AIC is retained for the next iteration.

```r
selectedMod <- step(model_full)
```

```
## Start:  AIC=-279.88
## log(Weight) ~ log(Length) + log(Width) + Thickness + Name
##
##               Df Sum of Sq    RSS     AIC
## <none>                     3.5236 -279.88
## - Name         4   0.36600 3.8896 -278.88
## - Thickness    1   0.30216 3.8258 -274.39
## - log(Width)   1   0.81247 4.3361 -262.99
## - log(Length)  1   1.76080 5.2844 -245.00
```

```r
summary(selectedMod)
```

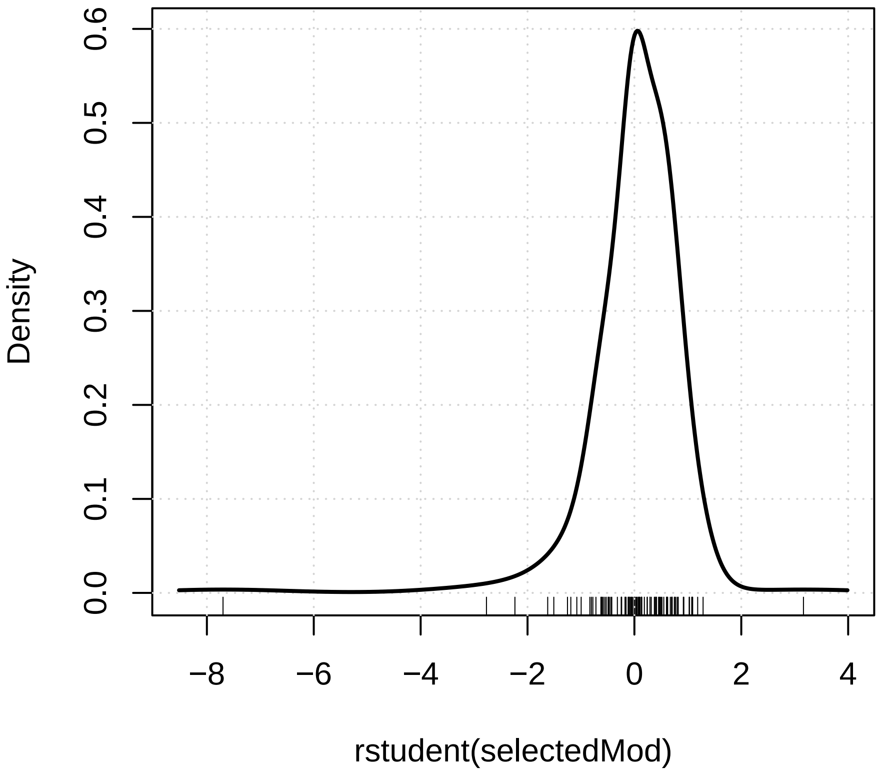
```
##
## Call:
## lm(formula = log(Weight) ~ log(Length) + log(Width) + Thickness +
```

```
##     Name, data = Darts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.17289 -0.08479  0.01706  0.11497  0.56577
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -5.04406    0.50942  -9.902 1.03e-15 ***
## log(Length)     1.00401    0.15590   6.440 7.34e-09 ***
## log(Width)      0.84842    0.19394   4.375 3.51e-05 ***
## Thickness       0.05734    0.02149   2.668  0.00918 **
## NameEnsor      -0.09834    0.08244  -1.193  0.23634
## NamePedernales  0.04303    0.08504   0.506  0.61421
## NameTravis      0.18590    0.08586   2.165  0.03325 *
## NameWells       0.07776    0.08767   0.887  0.37768
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.206 on 83 degrees of freedom
## Multiple R-squared:  0.8579, Adjusted R-squared:  0.8459
## F-statistic: 71.59 on 7 and 83 DF,  p-value: < 2.2e-16
```

1.1.3 Regression Diagnostics

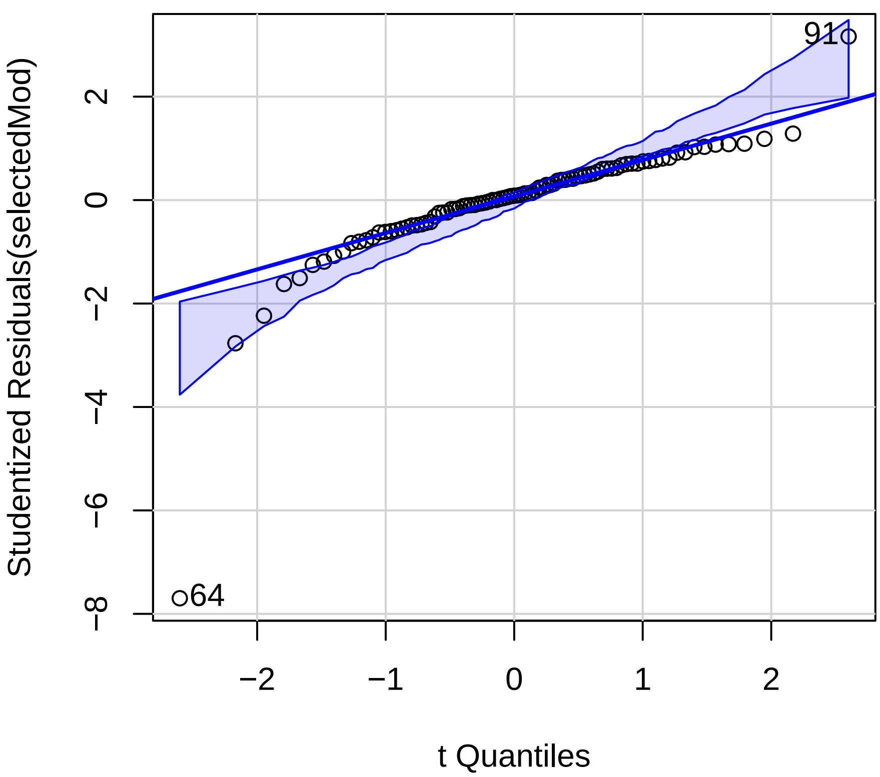
The `rstudent()` function returns studentized residuals, and the `densityPlot()` function fits an adaptive kernel density estimator to the distribution of the studentized residuals:

```
densityPlot(rstudent(selectedMod))
```



A `qqPlot()` can be used as a check for nonnormal errors, comparing the studentized residuals to a t-distribution:

```
qqPlot(selectedMod)
```



```
## [1] 64 91
```

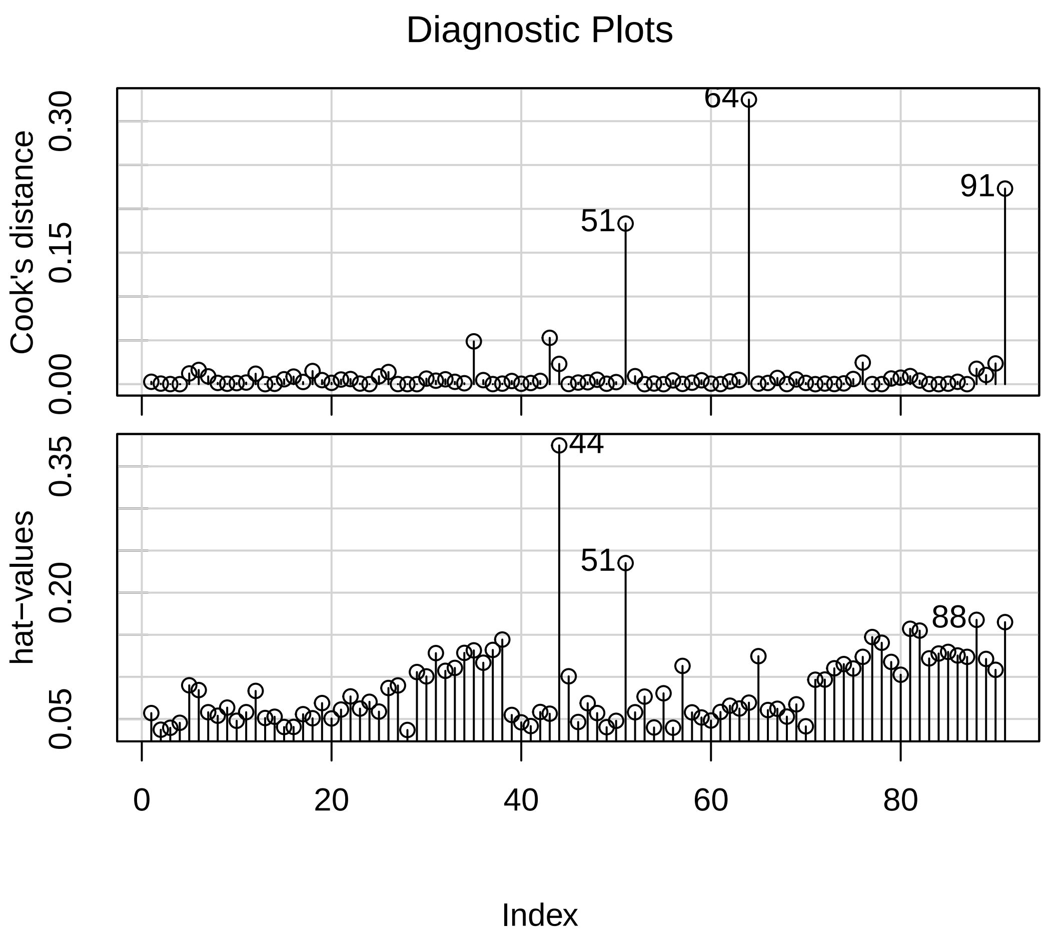
This next function tests for outliers in the regression:

```
outlierTest(selectedMod)
```

```
##     rstudent unadjusted p-value Bonferroni p
## 64 -7.698409         2.7563e-11   2.5082e-09
```

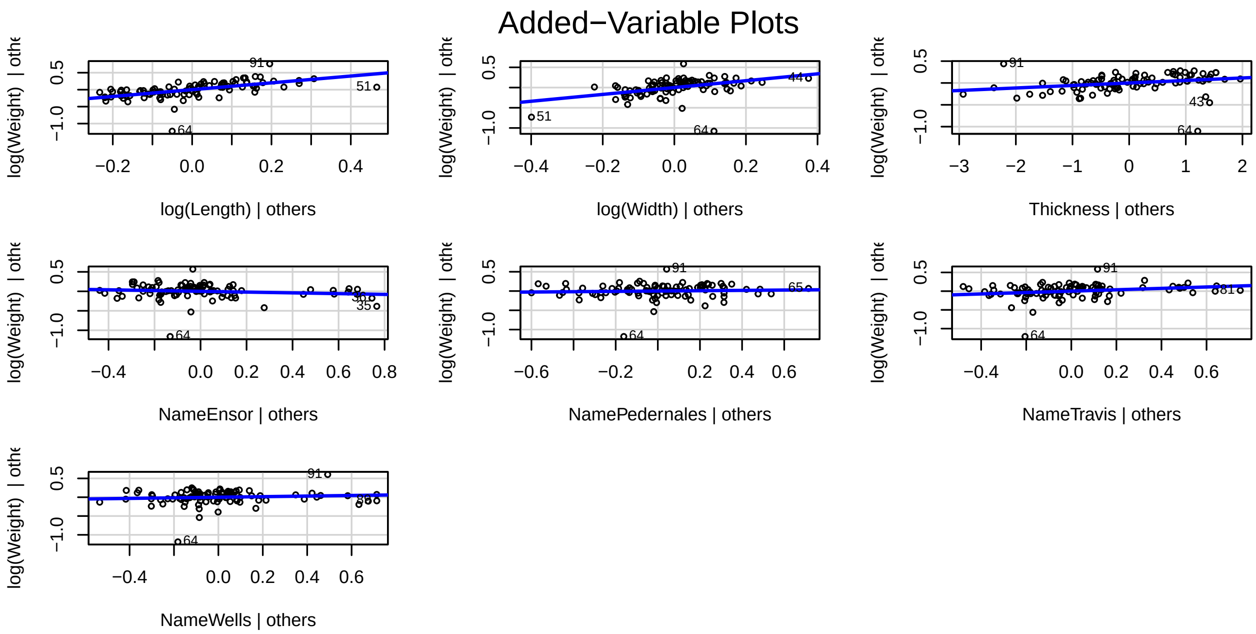
This graph displays influence measures in index plots:

```
influenceIndexPlot(selectedMod, vars=c("Cook", "hat"),
    id=list(n=3))
```



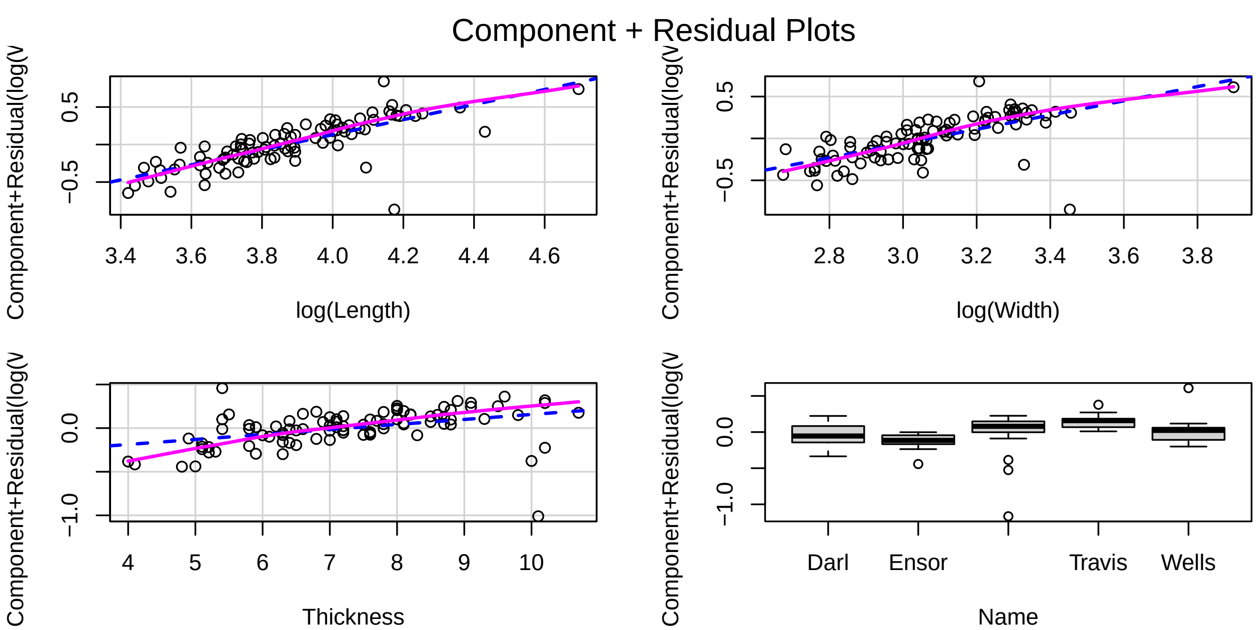
Added-variable plots for the regression, looking for influential cases:

```
avPlots(selectedMod,
    id=list(cex=0.75, n=3, method="mahal"))
```



Component-plus-residual plots for the regression, checking for nonlinearity:

```
crPlots(selectedMod, smooth=list(span=0.7))
```



Tests for non-constant error variance:

```
ncvTest(selectedMod)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 24.61847, Df = 1, p = 6.9879e-07
```

```r
ncvTest(selectedMod, var.formula= ~ log(Length) + log(Width) + Thickness +
    Name)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ log(Length) + log(Width) + Thickness + Name
## Chisquare = 36.82848, Df = 7, p = 5.0549e-06
```

Removing the 64th and 91th rows:

```r
whichNames(c("64", "91"), Darts)
```

```
## 64 91
## 64 91
```

```r
selectedMod_2 <- update(selectedMod, subset=-c(64, 91))
summary(selectedMod_2)
```

```
##
## Call:
## lm(formula = log(Weight) ~ log(Length) + log(Width) + Thickness +
##     Name, data = Darts, subset = -c(64, 91))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62920 -0.07798  0.01903  0.08655  0.27239
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -5.134122   0.355717 -14.433  < 2e-16 ***
## log(Length)     0.887774   0.108644   8.171 3.45e-12 ***
## log(Width)      0.957443   0.134109   7.139 3.63e-10 ***
## Thickness       0.091275   0.015371   5.938 6.88e-08 ***
## NameEnsor      -0.121528   0.056755  -2.141   0.0353 *
## NamePedernales  0.002826   0.058598   0.048   0.9617
## NameTravis      0.126051   0.059337   2.124   0.0367 *
## NameWells      -0.027716   0.062038  -0.447   0.6562
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1416 on 81 degrees of freedom
## Multiple R-squared:  0.9319, Adjusted R-squared:  0.926
## F-statistic: 158.4 on 7 and 81 DF,  p-value: < 2.2e-16
```

Comparing the regressions with and without these two cases:

```r
compareCoefs(selectedMod, selectedMod_2)
```

```
## Calls:
## 1: lm(formula = log(Weight) ~ log(Length) + log(Width) + Thickness + Name,
##   data = Darts)
## 2: lm(formula = log(Weight) ~ log(Length) + log(Width) + Thickness + Name,
```

```
##   data = Darts, subset = -c(64, 91))
##
##                 Model 1 Model 2
## (Intercept)      -5.044  -5.134
## SE                0.509   0.356
##
## log(Length)       1.004   0.888
## SE                0.156   0.109
##
## log(Width)        0.848   0.957
## SE                0.194   0.134
##
## Thickness        0.0573  0.0913
## SE               0.0215  0.0154
##
## NameEnsor       -0.0983 -0.1215
## SE               0.0824  0.0568
##
## NamePedernales 0.04303 0.00283
## SE             0.08504 0.05860
##
## NameTravis       0.1859  0.1261
## SE               0.0859  0.0593
##
## NameWells        0.0778 -0.0277
## SE               0.0877  0.0620
##
```

(b)

The 90% prediction interval for a new observation of Weight for a Dart of type Pedernales with Length = 50, Width = 20 and Thickness = 6 is (1.460714, 2.054469).

```
new_obs <- tibble(
  Length = 50,
  Width = 20,
  Thickness = 6,
  Name= c('Pedernales')
)
predict(selectedMod_2, newdata = new_obs, interval = "prediction", level = .9)
```

```
##        fit      lwr      upr
## 1 1.757592 1.509327 2.005857
```

Exercise 2

(a)

As a first step, we obtain summary statistics for the dataset `wheat`:

```r
wheat= read.table('wheat.txt', sep=" ") %>% as_tibble()
wheat$species= ifelse(wheat$species == "Rosa", 1, 0) %>% factor()
summary(wheat)
```

```
##       area         perimeter      compactness       asymmetry      species
##  Min.   :11.23   Min.   :12.63   Min.   :0.8392   Min.   :0.7651   0:70
##  1st Qu.:14.36   1st Qu.:14.34   1st Qu.:0.8714   1st Qu.:2.2200   1:70
##  Median :16.13   Median :15.13   Median :0.8819   Median :2.9730
##  Mean   :16.33   Mean   :15.21   Mean   :0.8818   Mean   :3.1561
##  3rd Qu.:18.72   3rd Qu.:16.20   3rd Qu.:0.8942   3rd Qu.:4.0220
##  Max.   :21.18   Max.   :17.25   Max.   :0.9183   Max.   :6.6850
```

Because the data follow the binomial distribution, the objective is to model the success probability p as a function of the covariates, i.e., to predict the species of the wheat seed based on the measurements of area, perimeter, compactness and asymmetry. We choose logistic regression from a series of generalised linear models.

```r
model= glm(species ~., data = wheat, family = "binomial")
model %>% summary()
```

```
##
## Call:
## glm(formula = species ~ ., family = "binomial", data = wheat)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.62957  -0.07678   0.00023   0.07125   2.37811
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -79.86844  605.46579  -0.132  0.89505
## area         -0.04596   18.66997  -0.002  0.99804
## perimeter     5.97347   39.89812   0.150  0.88099
## compactness -15.29775  343.34608  -0.045  0.96446
## asymmetry     1.10838    0.42658   2.598  0.00937 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 194.081  on 139  degrees of freedom
## Residual deviance:  32.469  on 135  degrees of freedom
## AIC: 42.469
##
## Number of Fisher Scoring iterations: 8
```

(b)

The probability that a seed with area = 13, perimeter=10, compactness=0.75, asymmetry=2 is of species Rosa is 9.472809e-14.

```r
new_seed <- tibble(
  area = 13,
  perimeter=10,
  compactness=0.75,
  asymmetry=2
)
predict(model, newdata = new_seed, type = 'response')
```

```
##            1
## 9.472809e-14
```

Perimeter=10 and compactness=0.75 are less than the minimum of these two variables in the data, respectively, which are not used in modelling the logistic regression. This could be harm the confidence of the prediction.