Analysis of Beer IBU with Gamma and Box-Cox

Alejandro Arellano

11/8/2021

Project code starting with box-cox transforamtion

```r
#Importing sample used in SAS taken from the master beer data set
BeerSample <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/STAT410_BeerSample.csv", 
head(BeerSample)
```
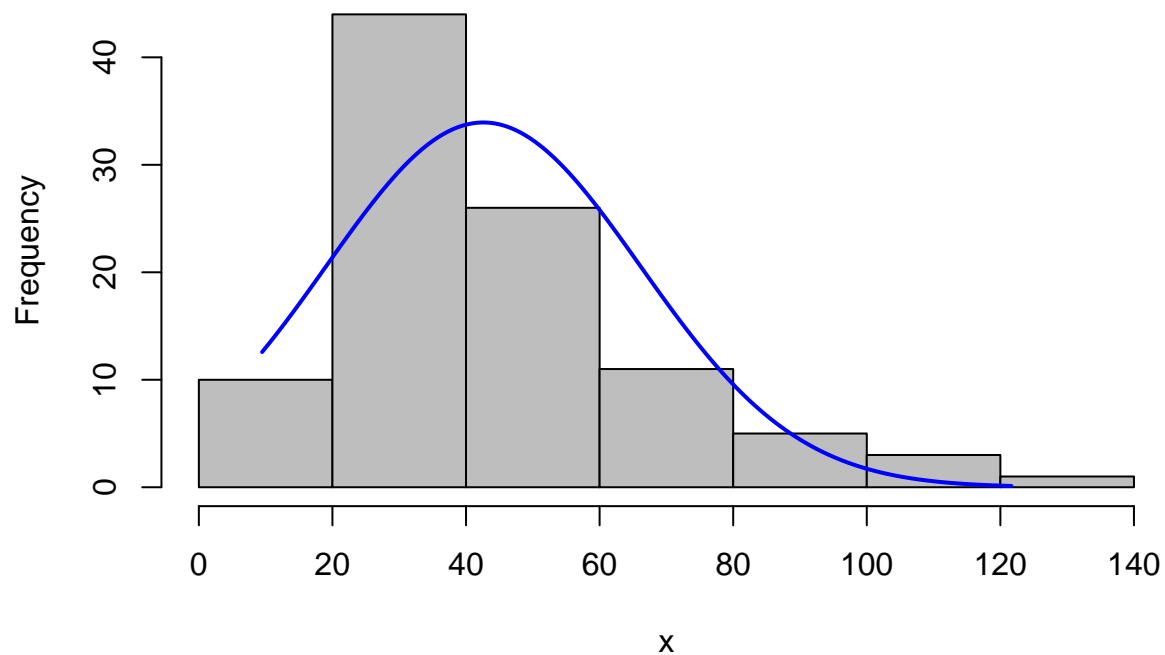
```
##      OG    FG    ABV    IBU Color BoilSize BoilTime Efficiency
## 1 1.064 1.012  6.79 79.58  5.11    24.61       60         35
## 2 1.107 1.024 10.92 64.61 19.23    26.50       60         70
## 3 1.057 1.015  5.54 25.39  4.77    45.42       60         75
## 4 1.082 1.012  9.17 64.80 10.46    31.00       60         70
## 5 1.083 1.018  8.53 24.48  4.83    12.00       60         60
## 6 1.076 1.021  7.21 58.91 31.24    28.39       60         73
```
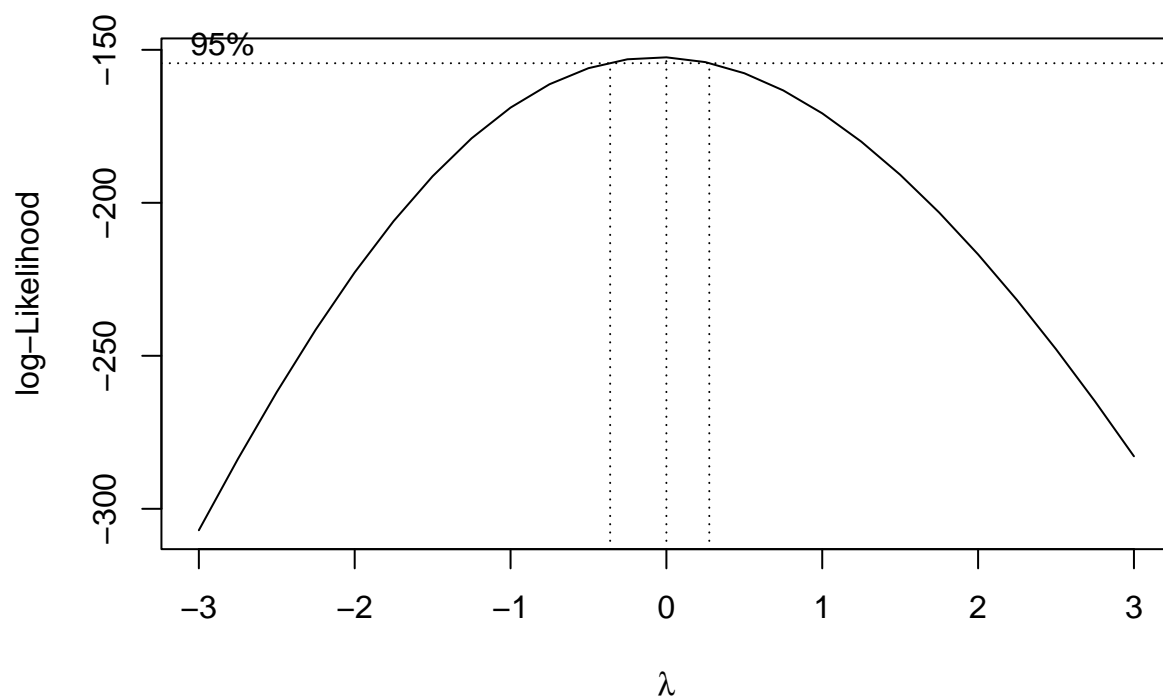
```r
#print(BeerSample$IBU)

#Checking for normality
plotNormalHistogram(BeerSample$IBU)
```

```r
#Shapiro Test for normality
shapiro.test(BeerSample$IBU)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  BeerSample$IBU
## W = 0.89101, p-value = 5.469e-07
```

```r
#Finding lambda for box-cox
BoxCox.fit<- boxcox(IBU ~ OG + FG + ABV + Color + BoilSize + BoilTime + Efficiency, data=BeerSample, lam
seq(-3,3,1/4), interp = FALSE)
```
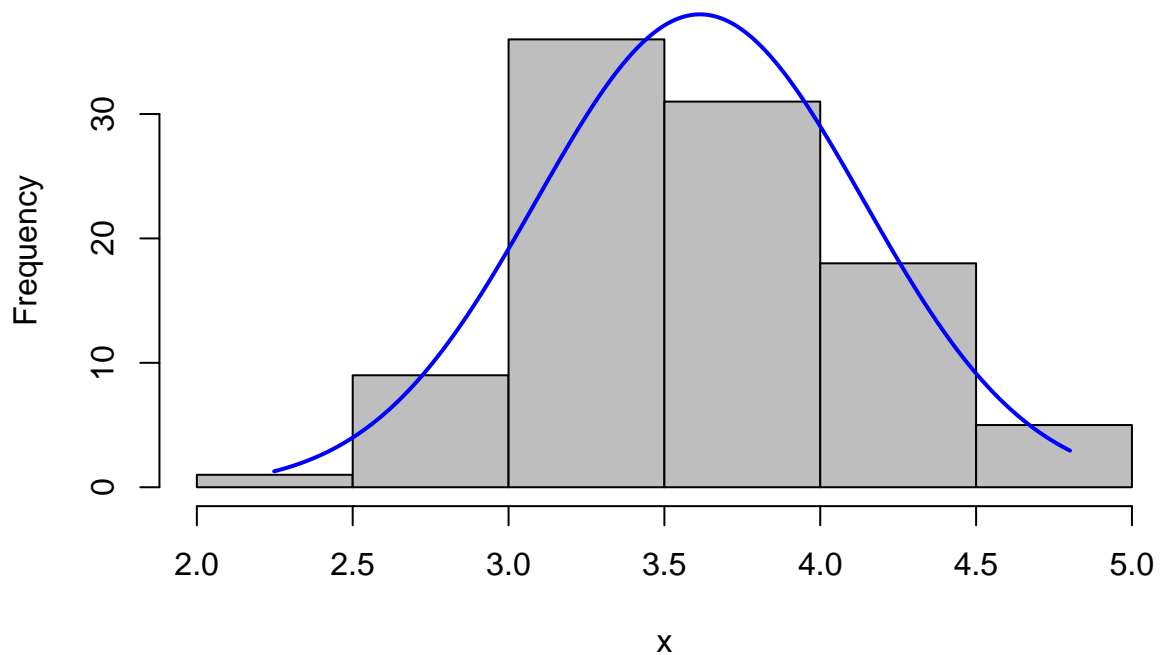
```
BoxCox.data<- data.frame(BoxCox.fit$x, BoxCox.fit$y)
ordered.data<- BoxCox.data[with(BoxCox.data, order(-BoxCox.fit.y)),]
ordered.data[1,]
```

```
##    BoxCox.fit.x BoxCox.fit.y
## 13            0    -152.4648
```

```
#Applying the box-cox transformation with lambda from above
IBU_Transformed<- log(BeerSample$IBU)

#Running normality test again but on the transformed variables
plotNormalHistogram(IBU_Transformed)
```

```
shapiro.test(IBU_Transformed)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  IBU_Transformed
## W = 0.98896, p-value = 0.5816
```

```
#Creating general linear model but on the transformed response
summary(fitted.model<- glm(IBU_Transformed ~ OG + FG + ABV + Color + BoilSize + BoilTime + Efficiency,
family=gaussian(link=identity)))
```

```
##
## Call:
## glm(formula = IBU_Transformed ~ OG + FG + ABV + Color + BoilSize +
##     BoilTime + Efficiency, family = gaussian(link = identity),
##     data = BeerSample)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.21340  -0.38050  -0.01175   0.33279   1.00577
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.673e+00  4.419e-01   8.312 8.05e-13 ***
```

```
## OG             1.492e-01  7.426e-02   2.009 0.047472 *
## FG            -8.652e-01  4.085e-01  -2.118 0.036863 *
## ABV            1.166e-01  3.211e-02   3.631 0.000463 ***
## Color          4.949e-03  4.698e-03   1.053 0.294958
## BoilSize       1.251e-05  6.428e-04   0.019 0.984514
## BoilTime       3.117e-03  5.256e-03   0.593 0.554573
## Efficiency    -4.843e-03  3.551e-03  -1.364 0.175922
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.229353)
##
##     Null deviance: 27.261  on 99  degrees of freedom
## Residual deviance: 21.100  on 92  degrees of freedom
## AIC: 146.2
##
## Number of Fisher Scoring iterations: 2
```

```r
#Estimated sigma
sigma(fitted.model)
```

```
## [1] 0.4789081
```

```r
#Checking again to see if the model fits after transformation
null.model<- glm(IBU_Transformed ~ 1, family=gaussian(link=identity))

#Printing deviance test for fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))
```

```
## 'log Lik.' 25.61772 (df=2)
```

```r
#Printing p-value for fit
print(p.value<- pchisq(deviance, df=7, lower.tail = FALSE))
```

```
## 'log Lik.' 0.0005893175 (df=2)
```

```r
#using fitted model for prediction
pred.IBU_Transformed<- predict(fitted.model, data.frame(OG=1.0000, FG=1.00000, ABV=7.00, Color=5.00, Bo
control="Cx"))
print(exp(pred.IBU_Transformed))
```

```
##        1
## 45.40326
```

Project code using gamma regression

```r
#Importing sample used in SAS taken from the master beer data set
BeerSample <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/STAT410_BeerSample.csv", 

head(BeerSample)
```

```
##       OG    FG    ABV    IBU Color BoilSize BoilTime Efficiency
## 1 1.064 1.012   6.79 79.58  5.11    24.61       60         35
## 2 1.107 1.024 10.92 64.61 19.23    26.50       60         70
## 3 1.057 1.015   5.54 25.39  4.77    45.42       60         75
## 4 1.082 1.012   9.17 64.80 10.46    31.00       60         70
## 5 1.083 1.018   8.53 24.48  4.83    12.00       60         60
## 6 1.076 1.021   7.21 58.91 31.24    28.39       60         73
```

```r
range(BeerSample$IBU)
```

```
## [1]   9.47 121.66
```

```r
mean(BeerSample$IBU)
```

```
## [1] 42.5893
```

```r
mean(BeerSample$ABV)
```

```
## [1] 6.1968
```

```r
#Creating the gamma model
summary(fitted.model<- glm(IBU ~ OG + FG + ABV + Color + BoilSize + BoilTime + Efficiency, data=BeerSamp
family=Gamma(link=log)))
```

```
##
## Call:
## glm(formula = IBU ~ OG + FG + ABV + Color + BoilSize + BoilTime +
##     Efficiency, family = Gamma(link = log), data = BeerSample)
##
## Deviance Residuals:
##     Min      1Q   Median      3Q      Max
## -1.0900  -0.4444  -0.1284   0.2295   1.0699
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.751e+00  4.512e-01   8.314 7.98e-13 ***
## OG           1.429e-01  7.582e-02   1.884 0.062705 .
## FG          -8.435e-01  4.171e-01  -2.022 0.046041 *
## ABV          1.236e-01  3.279e-02   3.770 0.000288 ***
## Color        3.374e-03  4.797e-03   0.703 0.483649
## BoilSize    -3.697e-05  6.563e-04  -0.056 0.955208
## BoilTime     3.059e-03  5.367e-03   0.570 0.570030
## Efficiency  -4.934e-03  3.626e-03  -1.361 0.176941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.2391185)
##
##     Null deviance: 27.484  on 99  degrees of freedom
## Residual deviance: 20.945  on 92  degrees of freedom
## AIC: 871.78
##
## Number of Fisher Scoring iterations: 6
```

```r
#Checking the fit of the model
null.model<- glm(IBU ~ 1, data=BeerSample, family=Gamma(link=log))

#Printing deviance test for fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))
```

```
## 'log Lik.' 28.25423 (df=2)
```

```r
#Printing p-value for fit
print(p.value<- pchisq(deviance,7,lower.tail = FALSE))
```

```
## 'log Lik.' 0.0001977445 (df=2)
```

```r
#Using the gamma model to predict values
print(predict(fitted.model, data.frame(OG=1.0000, FG=1.00000, ABV=7.00, Color=5.00, BoilSize=24.00, Boil
type="response"))
```

```
##        1
## 51.54828
```