LNL_Course_Proj_Part2

Anupriya Thirumurthy

8/24/2018

Course Project Part 2:

Read the data and create a data frame with one-minute breaks counts and temperature measurements.

Ceate data frame with necessary data.

```
dataPath <- "/Users/anupriyathirumurthy/Documents/University/MScA_UoC/Courses/LinearAndNonLinearModels/
Part2.Data<-read.csv(file=paste(dataPath,"OneMinuteCountsTemps.csv",sep="/"))
head(Part2.Data)
```

```
##   Minute.times Minute.counts Minute.Temps
## 1           30             7     91.59307
## 2           90            10     97.30860
## 3          150             7     95.98865
## 4          210             4    100.38440
## 5          270             1     99.98330
## 6          330             6    102.54126
```

```
dim(Part2.Data)
```

```
## [1] 250   3
```

Removing rows with NA.

```
Part2.Data<-Part2.Data[complete.cases(Part2.Data),]
dim(Part2.Data)
```

```
## [1] 242   3
```

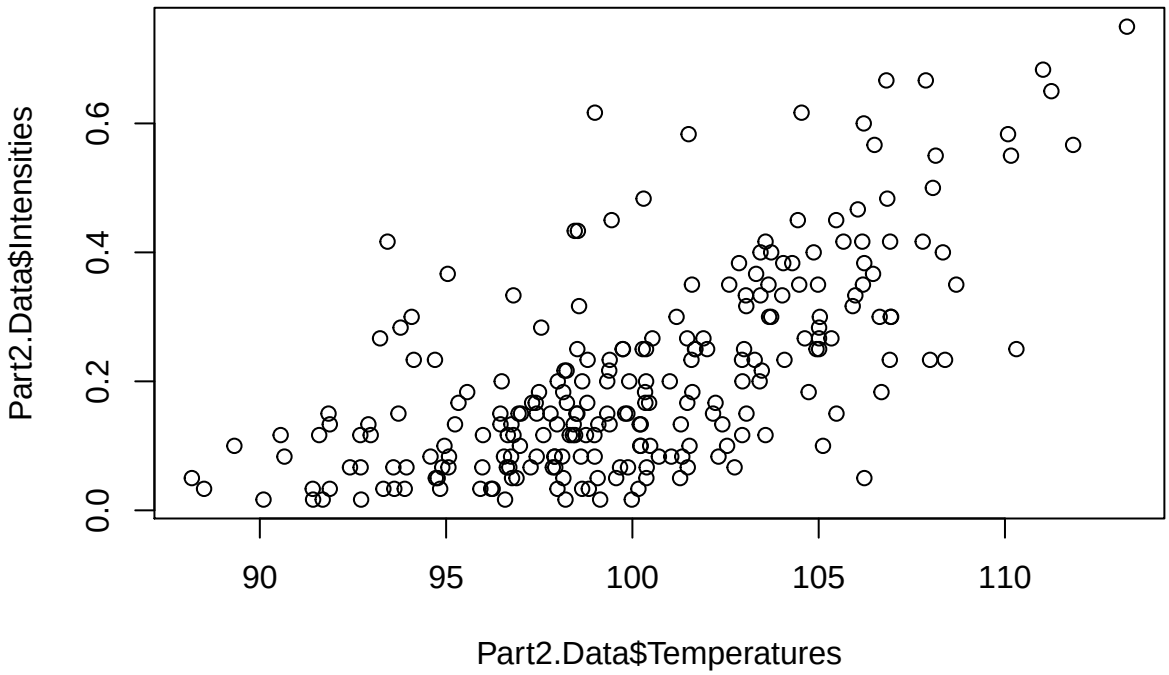
Adding column with intensities.

```
Part2.Data<-as.data.frame(cbind(Part2.Data,Part2.Data[,2]/60))
colnames(Part2.Data)<-c("Times","Counts","Temperatures","Intensities")
head(Part2.Data)
```

```
##   Times Counts Temperatures Intensities
## 1    30      7     91.59307  0.11666667
## 2    90     10     97.30860  0.16666667
## 3   150      7     95.98865  0.11666667
## 4   210      4    100.38440  0.06666667
## 5   270      1     99.98330  0.01666667
## 6   330      6    102.54126  0.10000000
```

Visualizing the data.

```
plot(Part2.Data$Temperatures,Part2.Data$Intensities)
```

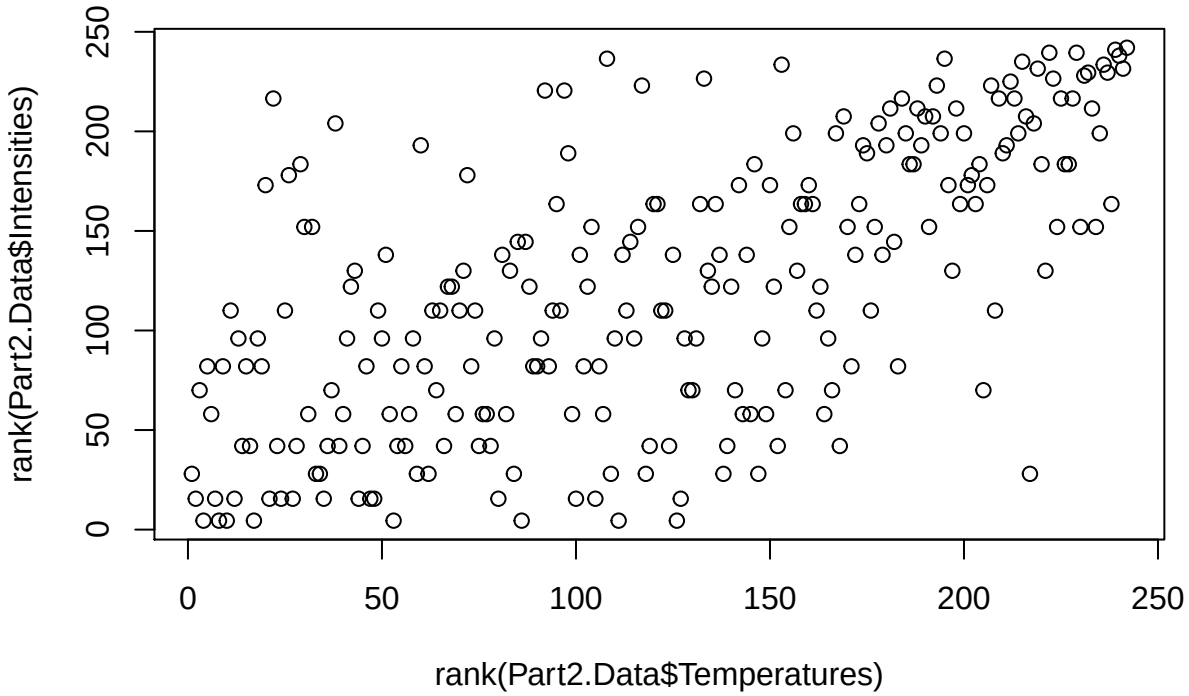


Interpreting the plot.

I could see that there is an direct and possitive relationship between temperature and Intensities.

Analyzing empirical copula.

```
plot(rank(Part2.Data$Temperatures),rank(Part2.Data$Intensities))
```



Type of dependency in empirical copula

This looks like a Gumbel copula as the data is more data centered on the top right corner.

Distribution of temperatures?

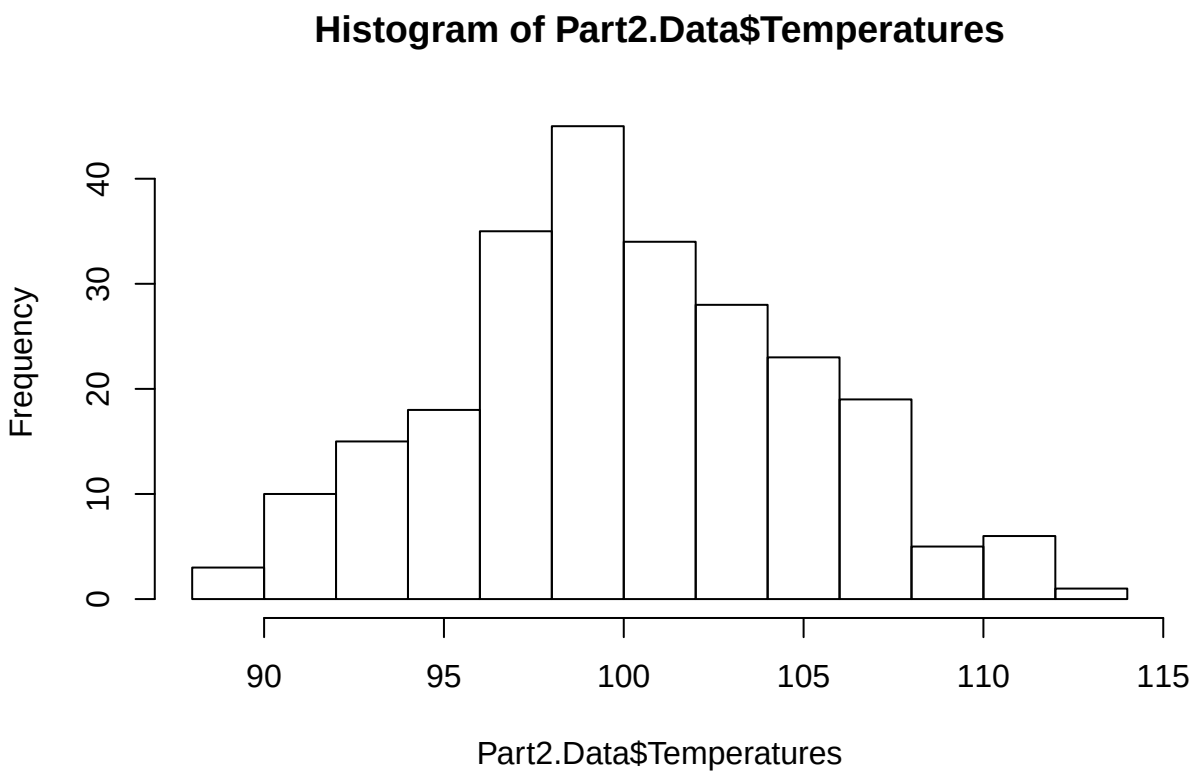
Based on the below histogram, the distribution of temperatures looks like normal distribution.

Load package MASS to estimate distributions

```
suppressWarnings(library(MASS))
```

Observing the histogram

```
hist(Part2.Data$Temperatures)
```



Estimating and testing normal distribution using fitdistr() from MASS.

Using Kolmogorov-Smirnov test function ks.test() to confirm correctness of normal assumption for temperature.

```
Fitting.Normal <- fitdistr(Part2.Data$Temperatures, densfun = "normal")
Fitting.Normal
```

```
##       mean           sd
##   100.0698530     4.8124839
##  (  0.3093582) (  0.2187493)
```

```
(KS.Normal <- ks.test(Part2.Data$Temperatures,"pnorm", mean=mean(Part2.Data$Temperatures), sd=sd(Part2.
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Part2.Data$Temperatures
## D = 0.048981, p-value = 0.6071
## alternative hypothesis: two-sided
```

Result

The null hypothesis that the two samples are drawn from the same distribution cannot be rejected at this level of p-value.

Fiting a copula

Select a parametric copula appropriate for the observed type of dependence.

Fit the copula Copula.Fit and use it for simulation of rare events.

```
suppressWarnings(library(copula))

dat <-  Part2.Data[,c(1,4)]
head(dat)
```

```
##   Times Intensities
## 1    30  0.11666667
## 2    90  0.16666667
## 3   150  0.11666667
## 4   210  0.06666667
## 5   270  0.01666667
## 6   330  0.10000000
```

```
par(mfrow=c(2,2))

#Gumbel Copula

Gumbel.Copula.2<-gumbelCopula(param=2,dim=2)

Copula.Object<-gumbelCopula(param=5,dim=2)
Copula.Fit<-fitCopula(Copula.Object,
          pobs(Part2.Data[,3:4],ties.method = "average"),
          method = "ml",
          optim.method = "BFGS",
          optim.control = list(maxit=1000))

summary(Copula.Fit)
```

```
## Call: fitCopula(copula, data = data, method = "ml", optim.method = "BFGS",
##     optim.control = ..3)
## Fit based on "maximum likelihood" and 242 2-dimensional observations.
## Gumbel copula, dim. d = 2
##       Estimate Std. Error
## alpha    1.877      0.099
## The maximized loglikelihood is 74.18
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##       11        4
```

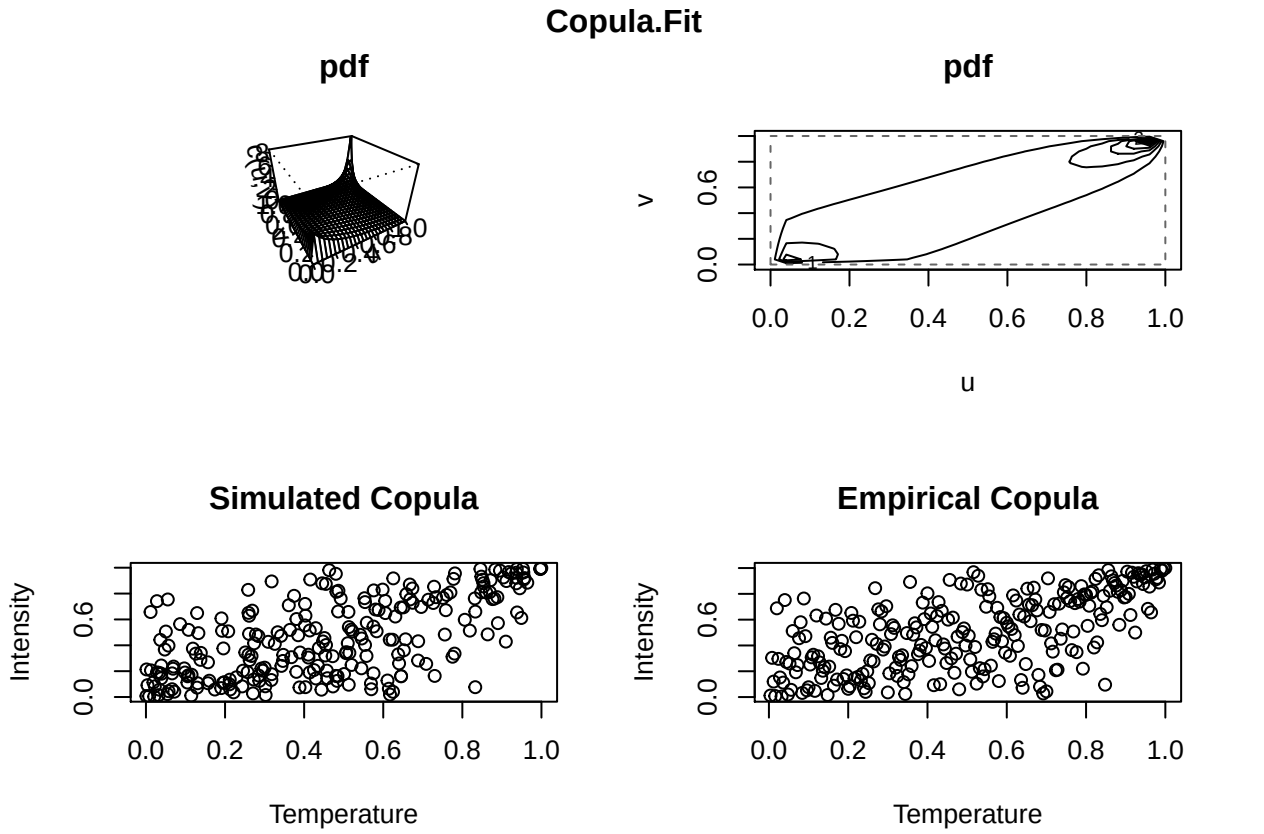
Simulating data using Copula.Fit with one variable normally distributed, as temperature and the other with the distribution of your choice for the intensities.

In order to make comparison possible, using set.seed(8301735).

First simulating 250 observations and making a 4-panel graph that we use to represent copula.

Creating a copula object before running simulation.

```
par(mfrow=c(2,2))
Gumbel.Copula.1<-gumbelCopula(param=Copula.Fit@estimate,dim=2)
set.seed(8301735)
Simulated.Gumbel.Copula.1<-rCopula(250,Gumbel.Copula.1)
persp(Gumbel.Copula.1, dCopula, main="pdf",xlab="u", ylab="v", zlab="c(u,v)")
contour(Gumbel.Copula.1,dCopula, main="pdf",xlab="u", ylab="v")
SimulatedN<-length(Simulated.Gumbel.Copula.1[,1])
plot(Simulated.Gumbel.Copula.1,main="Simulated Copula",xlab="Temperature",ylab="Intensity")
plot(apply(Simulated.Gumbel.Copula.1,2,rank)/SimulatedN,main="Empirical Copula",
     xlab="Temperature",ylab="Intensity")
title("Copula.Fit",outer=TRUE,line=-1)
```



Now running longer simulation to observe more tail events using estimated parameters for distributions of temperatures and intensities.

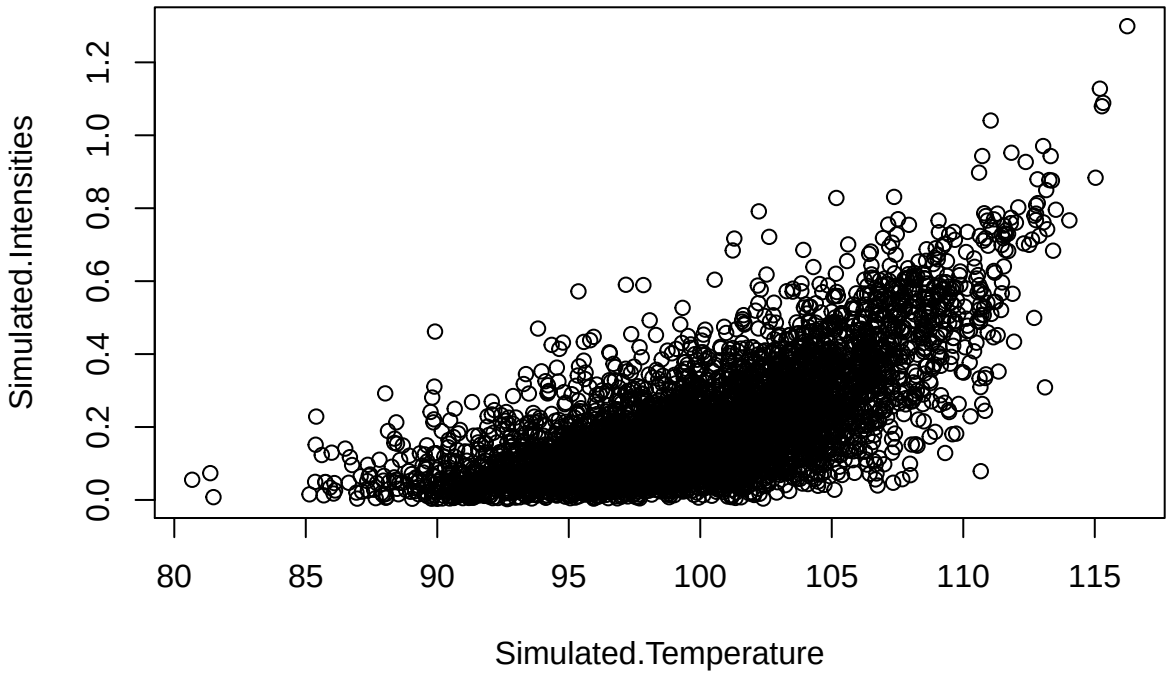
Simulating 5000 pairs of intensities and temperatures using the estimated copula.

Using the same seed.

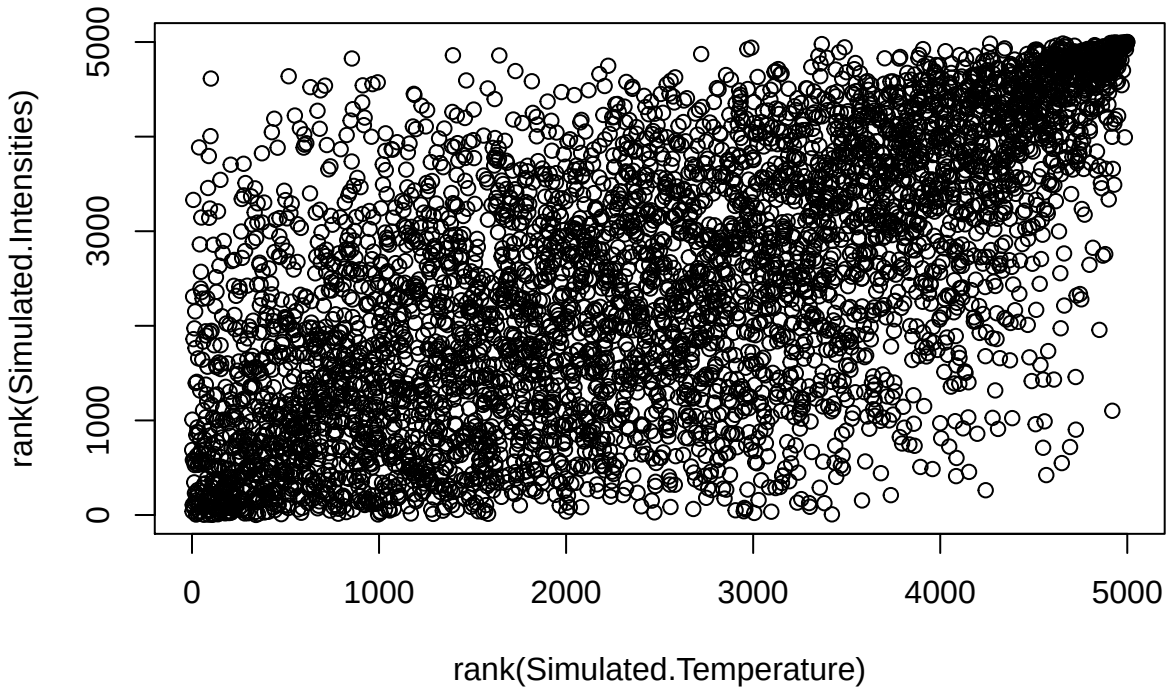
```
set.seed(8301735)
Simulated.Gumbel.new<-rCopula(5000,Gumbel.Copula.1)
Simulated.Temperature<-qnorm(Simulated.Gumbel.new[,2],Fitting.Normal$estimate[1],
                             Fitting.Normal$estimate[2])
Simulated.Intensities<-qgamma(Simulated.Gumbel.new[,1],shape=1.655739,rate=8.132313)
```

Ploting the simulated variables and their empirical copula.

```
plot(Simulated.Temperature,Simulated.Intensities)
```




```
plot(rank(Simulated.Temperature),rank(Simulated.Intensities))
```



Now we use the simulated data to analyze the tail dependency.

Selecting the simulated pairs with intensity greater than 0.5 and temperature greater than 110.

Using these data to fit negative binomial regression.

Using the initial sample of intensities and temperatures to fit the negative binomial regression for more regular ranges of intensity and temperature.

First, fiting the model to the sample, the name of the fitted model is NB.Fit.To.Sample.

```
NB.Fit.To.Sample<-suppressWarnings(glm.nb(Counts ~ Temperatures, Part2.Data))
summary(NB.Fit.To.Sample)
```

```
##
## Call:
## glm.nb(formula = Counts ~ Temperatures, data = Part2.Data, init.theta = 4.202611757,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5358  -0.9162  -0.1509   0.4795   3.2190
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -7.431375   0.785456  -9.461   <2e-16 ***
## Temperatures  0.098432   0.007793  12.631   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(4.2026) family taken to be 1)
##
##     Null deviance: 431.60  on 241  degrees of freedom
## Residual deviance: 257.19  on 240  degrees of freedom
## AIC: 1557.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  4.203
##           Std. Err.:  0.549
##
##  2 x log-likelihood:  -1551.817
```

Analyze the summary of the fit. Below are the returned parameters.

```
NB.Fit.To.Sample$coefficients
```

```
##  (Intercept) Temperatures
##  -7.43137538   0.09843241
```

```
NB.Fit.To.Sample$deviance
```

```
## [1] 257.1937
```

```
NB.Fit.To.Sample$df.residual
```

```
## [1] 240
```

```
NB.Fit.To.Sample$aic
```

```
## [1] 1557.817
```

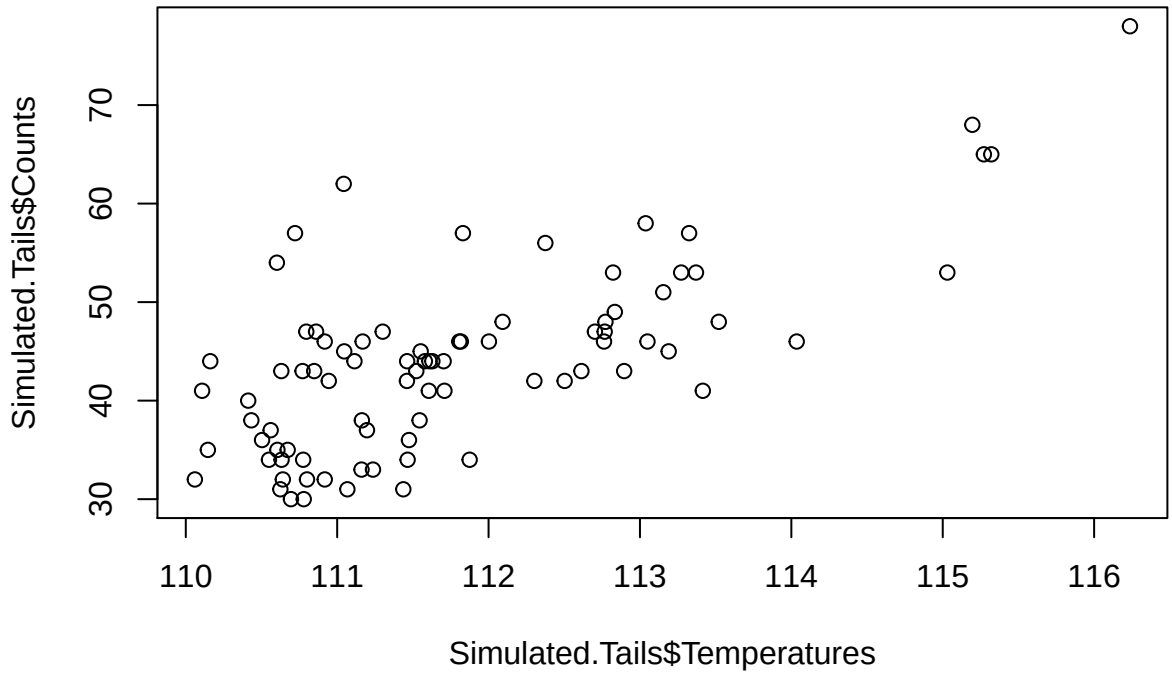
Creating the simulated sample for tail events.

```
Simulated.Tails<-as.data.frame(
  cbind(round(Simulated.Intensities[(Simulated.Temperature>110)&(Simulated.Intensities>.5)]*60),
        Simulated.Temperature[(Simulated.Temperature>110)&(Simulated.Intensities>.5)]))
```

```r
colnames(Simulated.Tails)<-c("Counts","Temperatures")
```

Ploting the simulated tail events.

```r
plot(Simulated.Tails$Temperatures,Simulated.Tails$Counts)
```



Fiting negative binomial model to the tail observations Simulated.Tails.

```r
NB.Fit.To.Sample2<-suppressWarnings(glm.nb(Counts ~ Temperatures, Simulated.Tails))
summary(NB.Fit.To.Sample2)
```

```
##
## Call:
## glm.nb(formula = Counts ~ Temperatures, data = Simulated.Tails,
##     init.theta = 385981.6374, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7586  -0.7146   0.0311   0.5559   3.1897
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -7.9678     1.2887  -6.183  6.3e-10 ***
## Temperatures   0.1050     0.0115   9.127  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(385981.6) family taken to be 1)
##
##     Null deviance: 158.968  on 83  degrees of freedom
## Residual deviance:  80.034  on 82  degrees of freedom
## AIC: 556.74
##
## Number of Fisher Scoring iterations: 1
```

```
##
##
##               Theta:  385982
##           Std. Err.:  8986395
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -550.742
```

Comparing the summaries of the two models.

The first model has residual deviance larger than the degrees of freedom, therefore, it is not a good fit. The second model has a residual deviance smaller than the degrees of freedom, denoting a good fit.

Note that the parameter ?? estimated by glm.nb() defines the variance of the model as ??+??2/??, where ?? is the mean. In other words, ?? defines overdispersion.

The first model has a theta of 4.203, and variance is larger than degrees of freedom, therefore, there is an overdispersion. As for the second model, it has a extremely large theta of 385983, and the variance is smaller than the degrees of freedome. Thus, it tells me there is no overdispersion.

Additionally I might may be try to fit a Poisson Model.

Relationships between the temperature and the counts?

Higher the temperature, higher is the counts. This confirms, there is a direct and possitive relationship between temperature and counts.

Fiting poisson model to Simulated.Tails$Counts and comparing the fit with the nagative binomial fit for Part2.Data.

```
Poisson.Fit<-glm(Counts~Temperatures,data=Simulated.Tails,family=poisson)
summary(Poisson.Fit)
```

```
##
## Call:
## glm(formula = Counts ~ Temperatures, family = poisson, data = Simulated.Tails)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7587  -0.7147   0.0311   0.5559   3.1899
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -7.9678     1.2886  -6.183 6.28e-10 ***
## Temperatures   0.1050     0.0115   9.128  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 158.987  on 83  degrees of freedom
## Residual deviance:  80.043  on 82  degrees of freedom
## AIC: 554.74
##
## Number of Fisher Scoring iterations: 4
```

```
Poisson.Fit$deviance
```

```
## [1] 80.04321
```

```
Poisson.Fit$df.residual
```

```
## [1] 82
```

```
Poisson.Fit$aic
```

```
## [1] 554.7414
```

Overdispersion in the Poisson fit?

We notice that the residual deviance is 80.043 and is lower than the degrees of freedom which is 82, indicating roboust fit. They also are pretty much close to each other. This tells us there is no overdispersion.