Linear Non-Linear Course Project

Madhavi Polisetti

June 28, 2018

Part 2.

1. Explore possible types of dependence between one-minute counts and temperature.

2.2.1: Show that the dimension of the One-Minute Counts is equal to the dimension shown in the assignment before removing the NAs

```
dataPath <- "/Users/mspolisetti/Desktop/R Studio/Linear-Class/Course Project/"
Part2.Data<-read.csv(file=paste(dataPath,"OneMinuteCountsTemps.csv",sep="/"))
head(Part2.Data)
```

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

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

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

```
#Remove rows with NA
Part2.Data<-Part2.Data[complete.cases(Part2.Data),]
dim(Part2.Data)
```

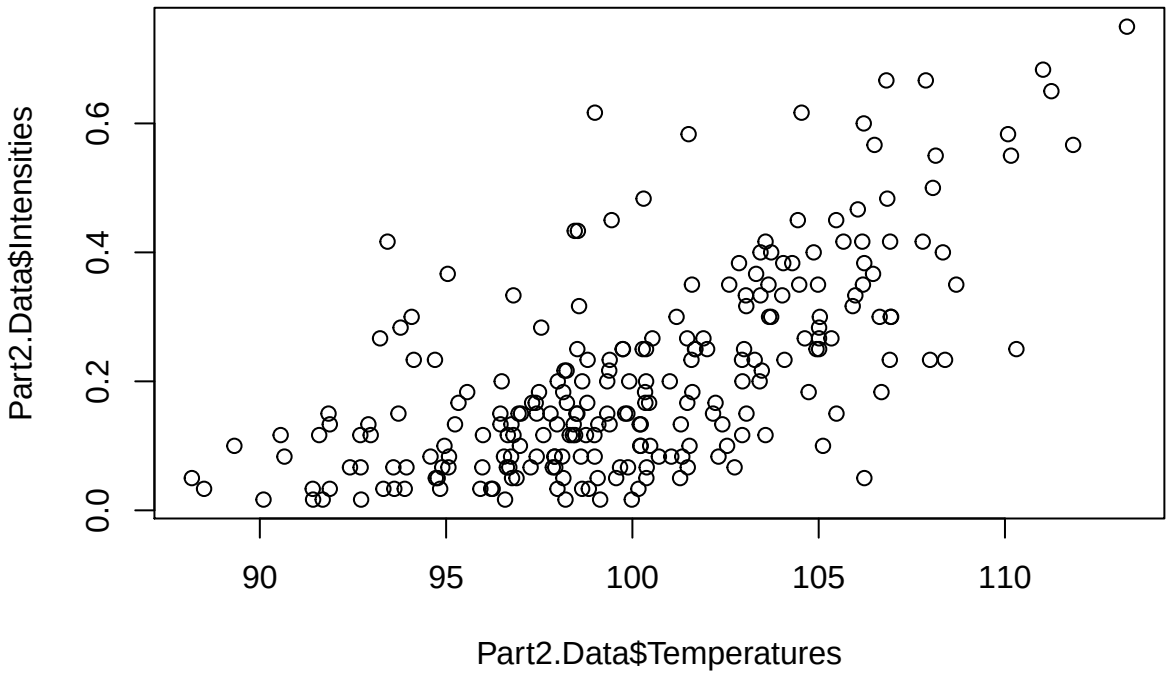
```
## [1] 242   3
```

```
#Add column with intensities.
Part2.Data<-as.data.frame(cbind(Part2.Data,Part2.Data[,3]/60))
colnames(Part2.Data)<-c("Temperatures","Times","Counts","Intensities")
head(Part2.Data)
```

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

2.2.3: Match the plot of temperature vs. intensities

```
#Visualize the data.
plot(Part2.Data$Temperatures,Part2.Data$Intensities)
```



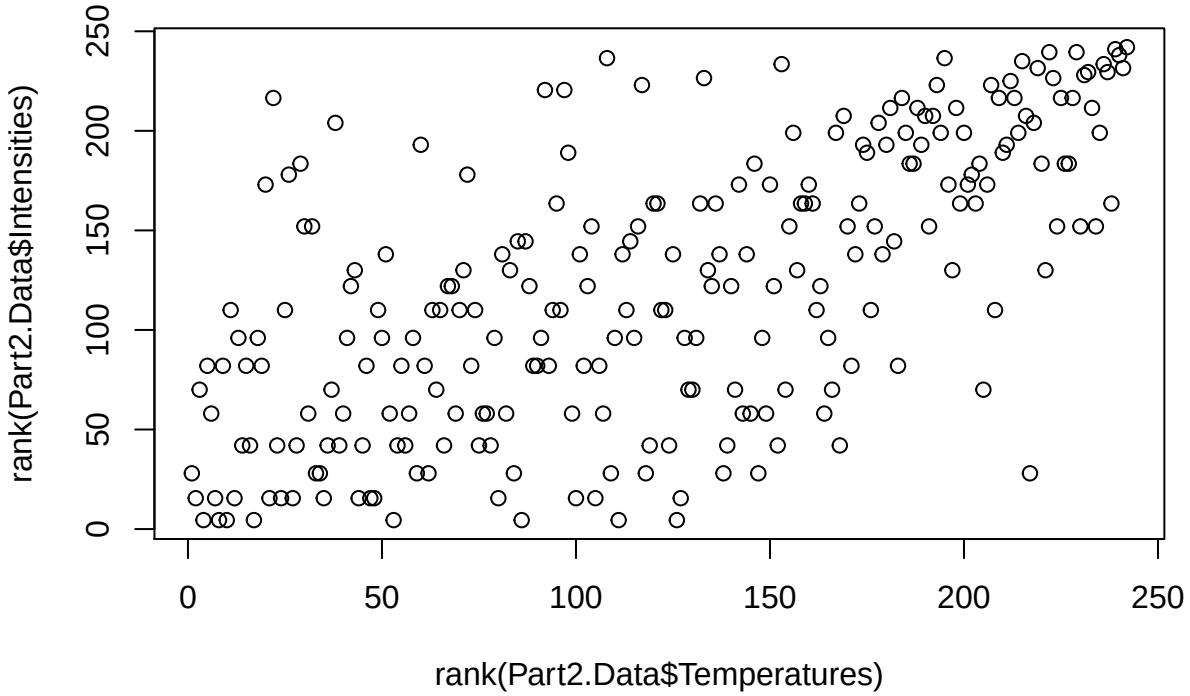
2.2.4: Interpret the plot and answer what type of relationship do you observe?

Temperature and Intensity appear to be correlated and have a positive co-monotnic relationship.

2.2.5: Plot and match the empirical copula and describe the type of dependency that is observed

Type of dependency mirrors a Gumbel copula as the data is more data centered on the top right corner.

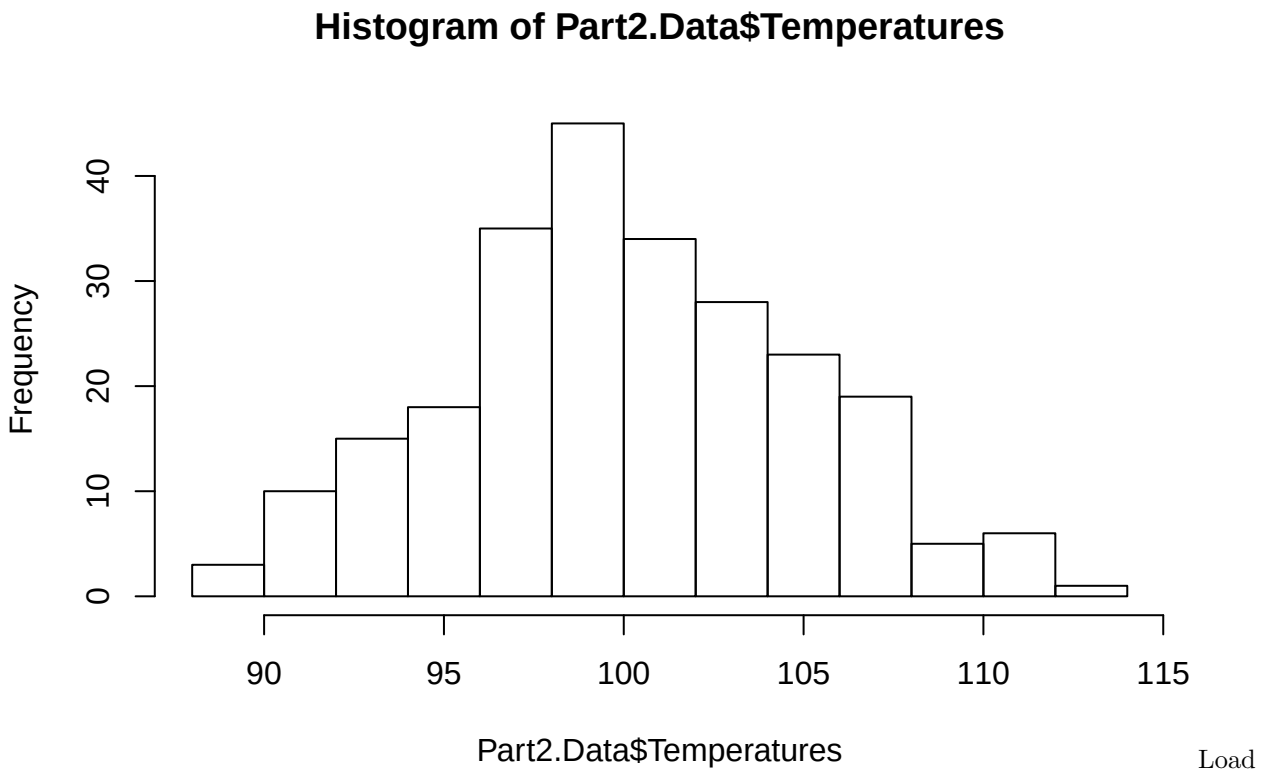
```
#Visualize the data.
plot(rank(Part2.Data$Temperatures),rank(Part2.Data$Intensities))
```



What is the distribution of temperatures?

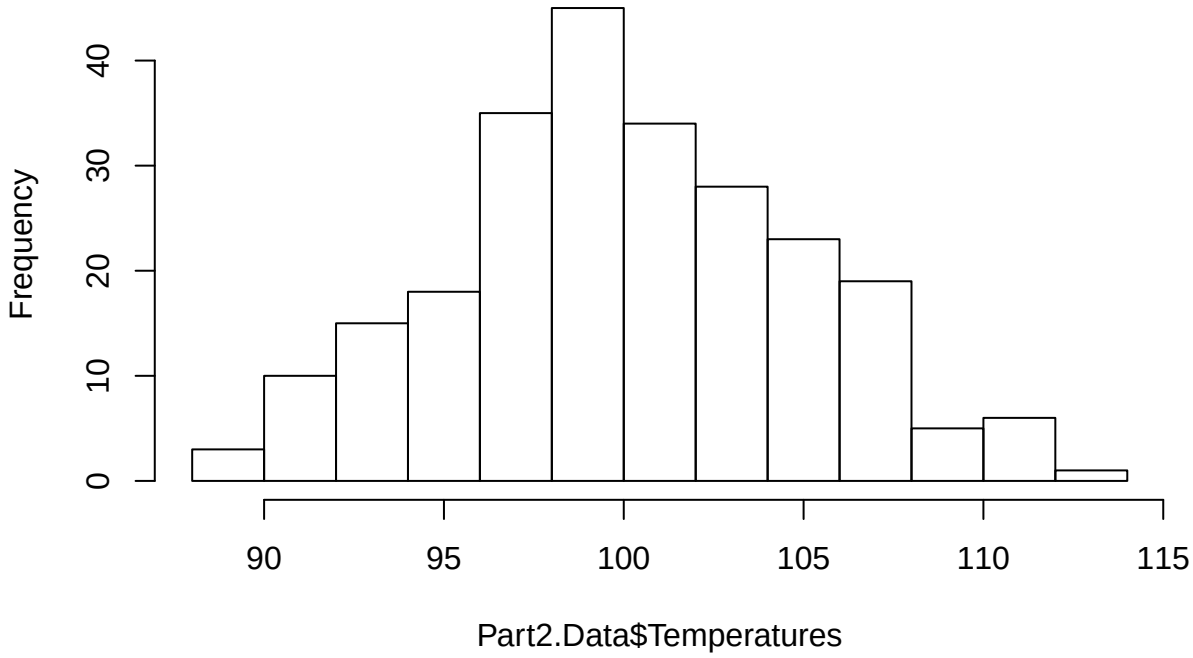
The distribution of temperatures looks like normal distribution based on the histogram.

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



Load package MASS to estimate distributions

```
suppressWarnings(library(MASS))
hist(Part2.Data$Temperatures)
```



Estimate and test normal distribution using fitdistr() from MASS.

2.2.6: Estimate the parameters of a normal distribution for the temperatures using the fitdistr() function from the package MASS – match the parameters shown

2.2.7: Use the KS Test to determine if the empirical distribution is equivalent to a theoretical normal distribution. Comment on the results

Results confirm the normality assumption for temperature.

```
(Fitting.Normal<-fitdistr(Part2.Data$Temperatures,"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
```

Fit a copula

2.2.8: Select a parametric copula that is appropriate for the observed type of dependence. Use the selected copula function to create a copula object and then estimate the parameters of this copula object. Observe the summary of the estimation and match the result shown in the assignment

```
#install.packages("copula")
suppressWarnings(library(copula))
dat <-  Part2.Data[,c(1,4)]
head(dat)
```

```
##   Temperatures Intensities
## 1     91.59307  0.11666667
## 2     97.30860  0.16666667
## 3     95.98865  0.11666667
## 4    100.38440  0.06666667
## 5     99.98330  0.01666667
## 6    102.54126  0.10000000
```

```
par(mfrow=c(2,2))
set.seed(8301735)

#Gumbel Copula

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

Copula.Fit <- fitCopula(Gumbel.Copula.2, pobs(dat, ties.method = "average"), method="ml")

Copula.Fit
```

```
## Call: fitCopula(copula, data = data, method = "ml")
## Fit based on "maximum likelihood" and 242 2-dimensional observations.
## Copula: gumbelCopula
## alpha
## 1.877
## The maximized loglikelihood is 74.18
## Optimization converged
```

```
#Simulate data using Copula.Fit with one variable normally distributed, as temperature and the other wi
```

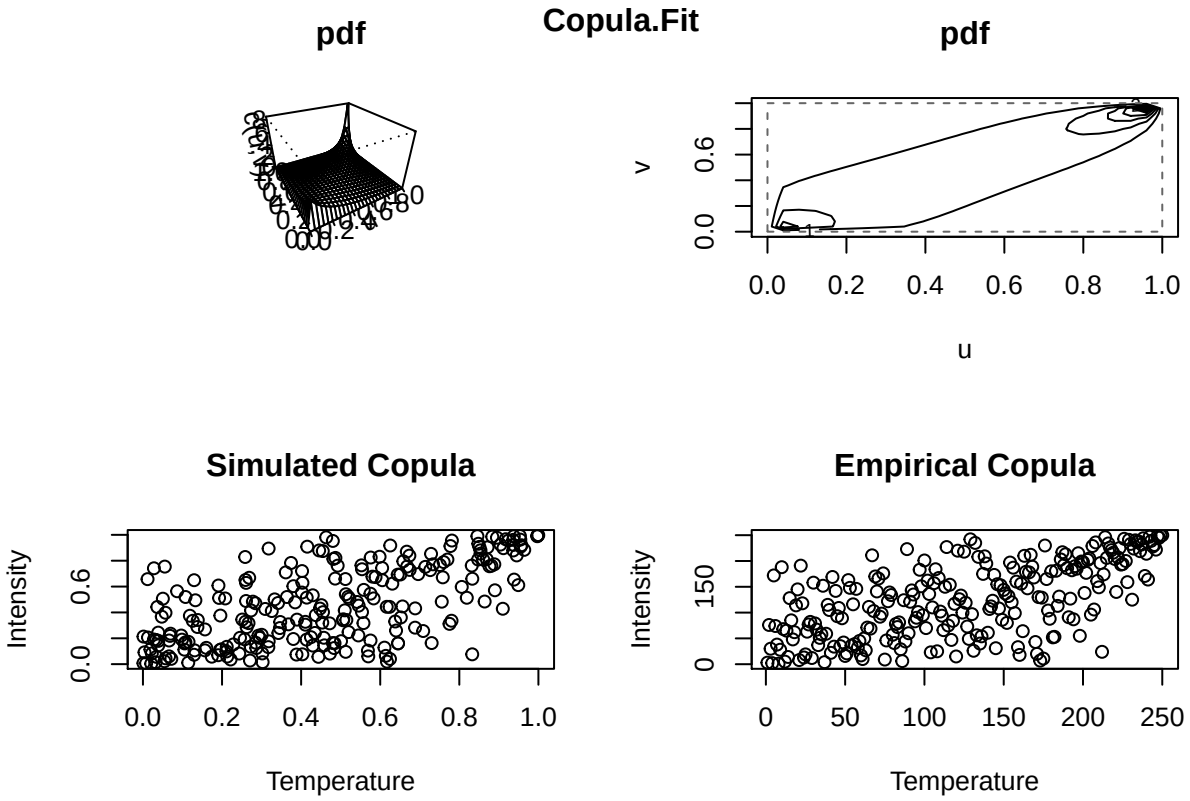
2.2.9: Create a new copula object with the estimated parameters from 2.2.8 above. Simulate 250 observations based on this copula object and plot the perspective plot, contour plot, and simulated and empirical copula

```
par(mfrow=c(2,2))
set.seed(8301735)
#Gaussian Copula, rho=0
persp(gumbelCopula(coef(Copula.Fit)), dCopula, main="pdf",xlab="u", ylab="v", zlab="c(u,v)")
contour(gumbelCopula(coef(Copula.Fit)), dCopula,main="pdf",xlab="u", ylab="v")


Simulated.Gumbel.Copula.2<-rCopula(250,gumbelCopula(coef(Copula.Fit)))
```

```
SimulatedN<-length(Simulated.Gumbel.Copula.2[,1])

head(Simulated.Gumbel.Copula.2)
```

```
##           [,1]       [,2]
## [1,] 0.3462948 0.27903991
## [2,] 0.9971721 0.99222938
## [3,] 0.6170434 0.01952008
## [4,] 0.2593859 0.65083082
## [5,] 0.2707132 0.02787195
## [6,] 0.8899826 0.57135896
```

```
plot(Simulated.Gumbel.Copula.2,main="Simulated Copula",xlab="Temperature",ylab="Intensity")
plot(apply(Simulated.Gumbel.Copula.2,2,function(x){ rank(x,ties.method="first") }) ,main="Empirical Copu
title("Copula.Fit",outer=TRUE,line=-2)
```




```
#Simulation of dependent random variables using copulas.
#Now run longer simulation to observe more tail events using estimated parameters for distributions of
```

2.2.10: Simulate 5000 observations based on the same copula object above (i.e. sample of U1 and U2 – two uniformly distributed random variables). Convert these samples into simulated temperatures and intensities using the normal quantile function for temperature and the quantile function of the distribution you chose in 5.3.11 for the intensities

2.2.11: Plot the simulated variables (temperature vs. intensity) and match the plot shown

2.2.12: Plot the empirical copula of the simulated variables (temperature vs. intensity) and match the plot shown

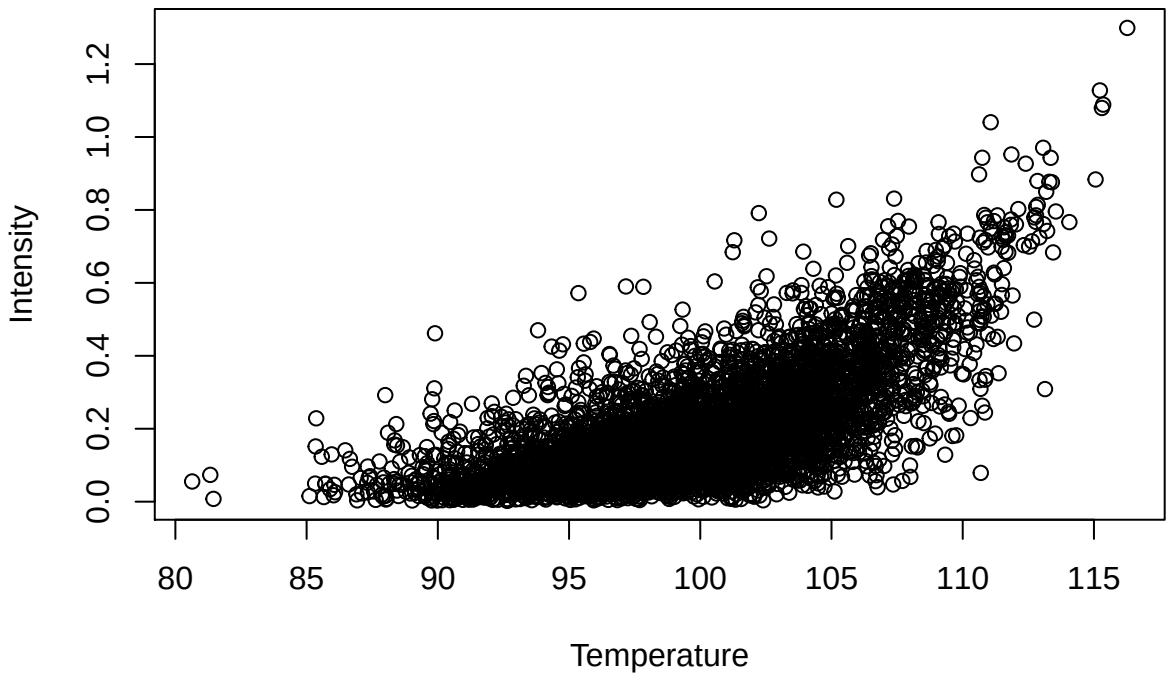
2.2.13: Use the initial sample of intensities and temperatures to fit a negative binomial regression for more regular ranges of intensity and temperature. Match the coefficients, deviance, degrees of freedom, and aic to

the quantities shown

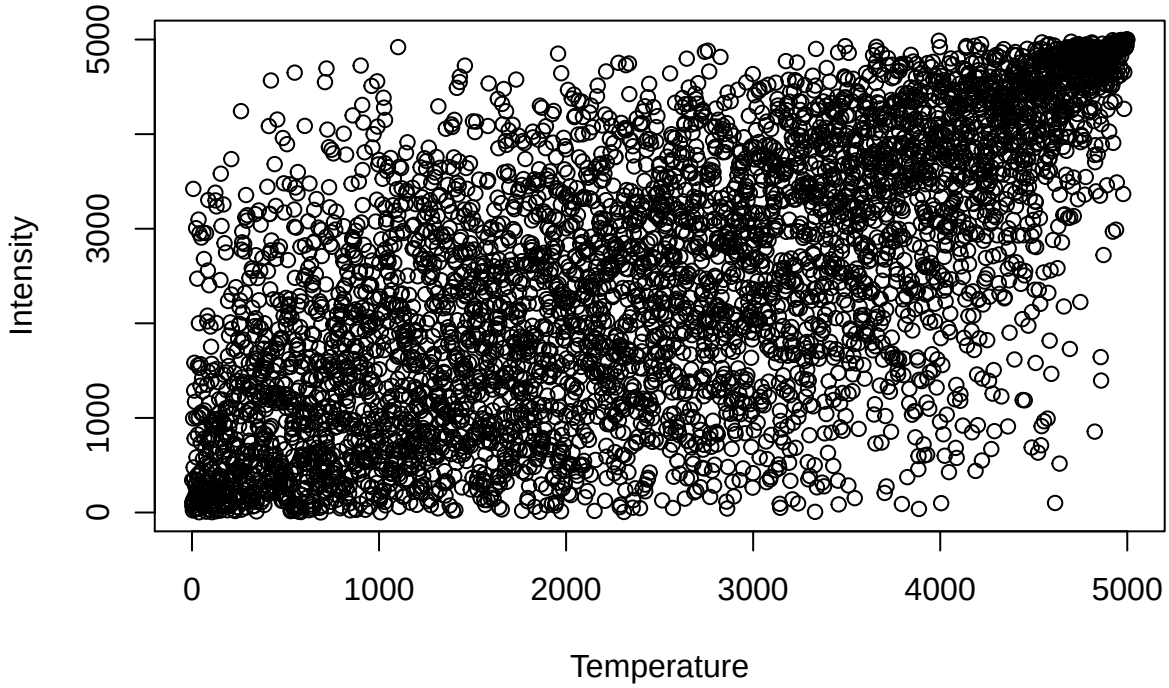
```
#Simulate 5000 pairs of intensities and temperatures using the estimated copula.
#Use the same seed.

par(mfrow=c(2,2))
set.seed(8301735)

Simulated.Gumbel.Copula.2.5000<-rCopula(5000,gumbelCopula(coef(Copula.Fit)))
SimulatedN<-length(Simulated.Gumbel.Copula.2.5000[,1])

Simulated.Temperature <- qnorm(Simulated.Gumbel.Copula.2.5000[,2],mean=mean(dat$Temperatures),sd(dat$Te
Simulated.Intensities <- qgamma(Simulated.Gumbel.Copula.2.5000[,1],rate=8.132313 , shape =1.655739)


Smiluated.data <- cbind(Temp = Simulated.Temperature, Intensity = Simulated.Intensities)

Simulated.Gumbel.Copula.2.5000.marginal <- c(Simulated.Temperature,Simulated.Intensities)

rankedData <- data.frame(apply(Simulated.Gumbel.Copula.2.5000,2,function(x){ rank(x,ties.method="first"
 #rankedData <- data.frame(apply(<<Copula Output Here>>,2,function(x){ rank(x,ties.method="first") } ))
head(rankedData)
```

```
##      X1   X2
## 1 1871 1614
## 2 4981 4981
## 3   43   91
## 4 3838 4214
## 5  342 2720
## 6 2227 2395
```

```
plot( Smiluated.data , xlab="Temperature", ylab="Intensity")
```



```r
plot(rankedData , xlab="Temperature", ylab="Intensity")
```




```r
#Now we can use the simulated data to analyze the tail dependency.
#Select the simulated pairs with intensity greater than 0.5 and temperature greater than 110.
#Use these data to fit negative binomial regression.

#Use the initial sample of intensities and temperatures to fit the negative binomial regression for mor

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

sim.dat <- as.data.frame(Smiluated.data)
head(sim.dat)
```

```
##        Temp   Intensity
## 1  97.76214 0.123832080
## 2 112.79833 0.808166247
## 3  89.80918 0.008095625
## 4 105.01129 0.286520963
## 5 100.54140 0.034073851
## 6  99.74784 0.145671233
```

```r
filtered.sim.dat <- sim.dat[sim.dat$Temp > 110 &  sim.dat$Intensity > 0.5,]
head(filtered.sim.dat)
```

```
##         Temp Intensity
## 2   112.7983 0.8081662
## 27  110.8182 0.7862040
## 29  110.7436 0.9430950
## 64  112.1175 0.8024398
## 129 113.2997 0.8771571
## 261 111.4605 0.5208392
```

```
NB.Fit.To.Sample <- glm.nb( Temp  ~ . ,filtered.sim.dat)
NB.Fit.To.Sample$coefficients
```

```
## (Intercept)   Intensity
##  4.67743324  0.05396254
```

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

```
## [1] 0.6457613
```

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

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

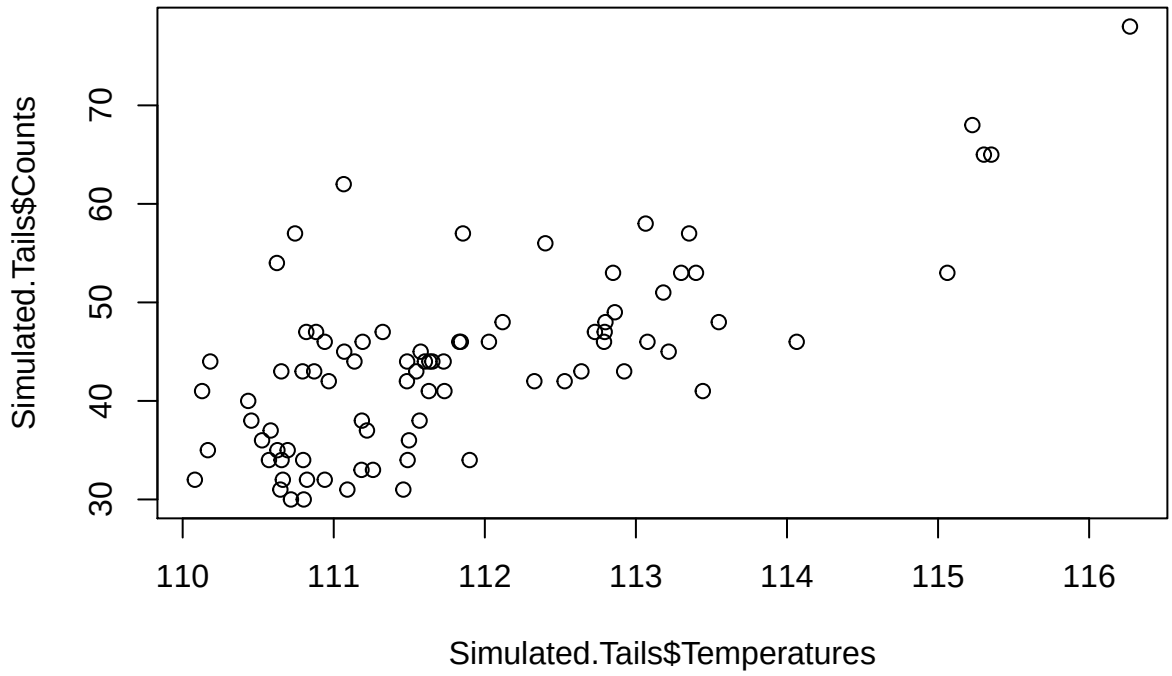
```
NB.Fit.To.Sample$aic
```

```
## [1] 557.3727
```

```
#Create 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)]))
colnames(Simulated.Tails)<-c("Counts","Temperatures")
```

2.2.14: Plot the simulated tail events (don't need to match the graph precisely but should represent a very similar pattern)

```
#Plot the simulated tail events.
plot(Simulated.Tails$Temperatures,Simulated.Tails$Counts)
```



2.2.15: Use the tail events of intensities and temperatures to fit a negative binomial regression for the tail observations

2.2.15: Use the tail events of intensities and temperatures to fit a negative binomial regression for the tail observations

```r
#Fit negative binomial model to the tail observations Simulated.Tails.
NB.Fit.To.Sample.Tails <- glm.nb( Counts~Temperatures ,Simulated.Tails)
summary(NB.Fit.To.Sample.Tails)
```

```
##
## Call:
## glm.nb(formula = Counts ~ Temperatures, data = Simulated.Tails,
##     init.theta = 385982.1063, 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)  -7.94607    1.28631  -6.177 6.52e-10 ***
## Temperatures  0.10479    0.01148   9.127  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(385982.1) 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
```

2.2.16: Compare the summaries of the 2 models – comment on the fit of each model as described by deviance vs. degrees of freedom and AIC

Compare the summaries of the two models. Note that the parameter theta estimated by glm.nb() defines the variance of the model as mu +mu 2/theta , where mu is the mean. In other words, theta defines overdispersion.

2.2.17: Respond to the question: What do the fitted parameters theta tell you about both models?

The first model theta is 4.203, and variance is larger than degrees of freedom, therefore, there is overdispersion. In the second model, there is large theta of 385982, and the variance is smaller than the degrees of freedom. Thus, it tells me there is no overdispersion. ######2.2.18: Respond to the question: Is there an alternative model that you would try to fit the simulated tail data? We can consider fitting a Poisson Model. ######2.2.19: Respond to the question: What do both models tell you about the relationship 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. ######2.2.20: Fit a Poisson model to the simulated tails data and roughly match the deviance, degrees of freedom, and AIC (won't match exactly)

2.2.21: Compare the Poisson model above in 2.2.20 with the negative binomial model above in 2.2.15

Is there overdispersion in the Poisson fit?

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.

```
#Fit negative binomial model to the tail observations Simulated.Tails.
Poisson.Fit <- glm( Counts~Temperatures,family="poisson" ,Simulated.Tails)


Poisson.Fit$deviance
```

```
## [1] 80.04322
```

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

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

```
Poisson.Fit$df.null
```

```
## [1] 83
```

```
Poisson.Fit$aic
```

```
## [1] 554.7415
```