Course Project Part 2

Brian Ritz

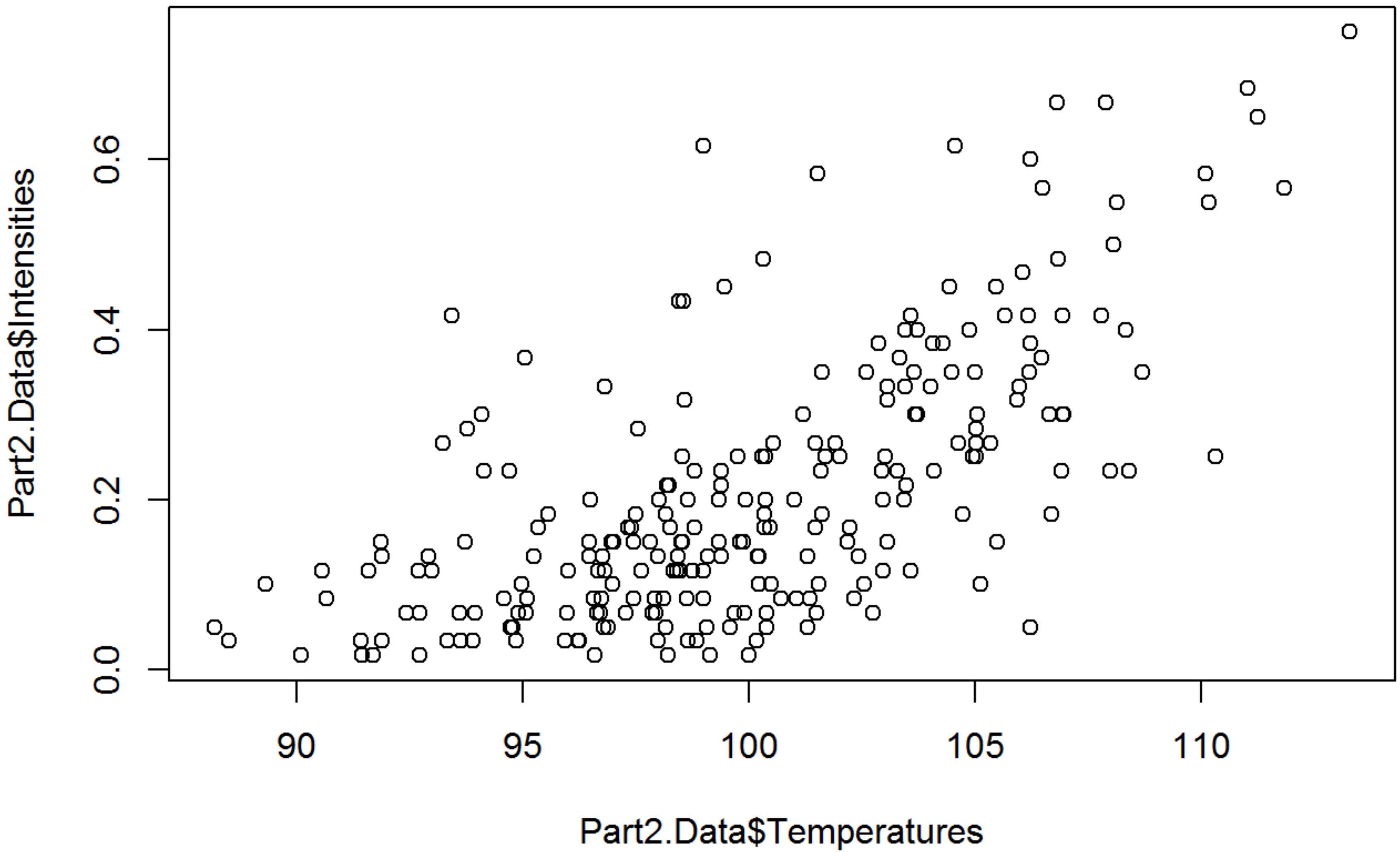
Saturday, February 21, 2015

```
Part2.Data<-read.csv(file="../input/MScA_LinearNonLinear_CourseProject_Part2.csv")
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
```

Visualize the data

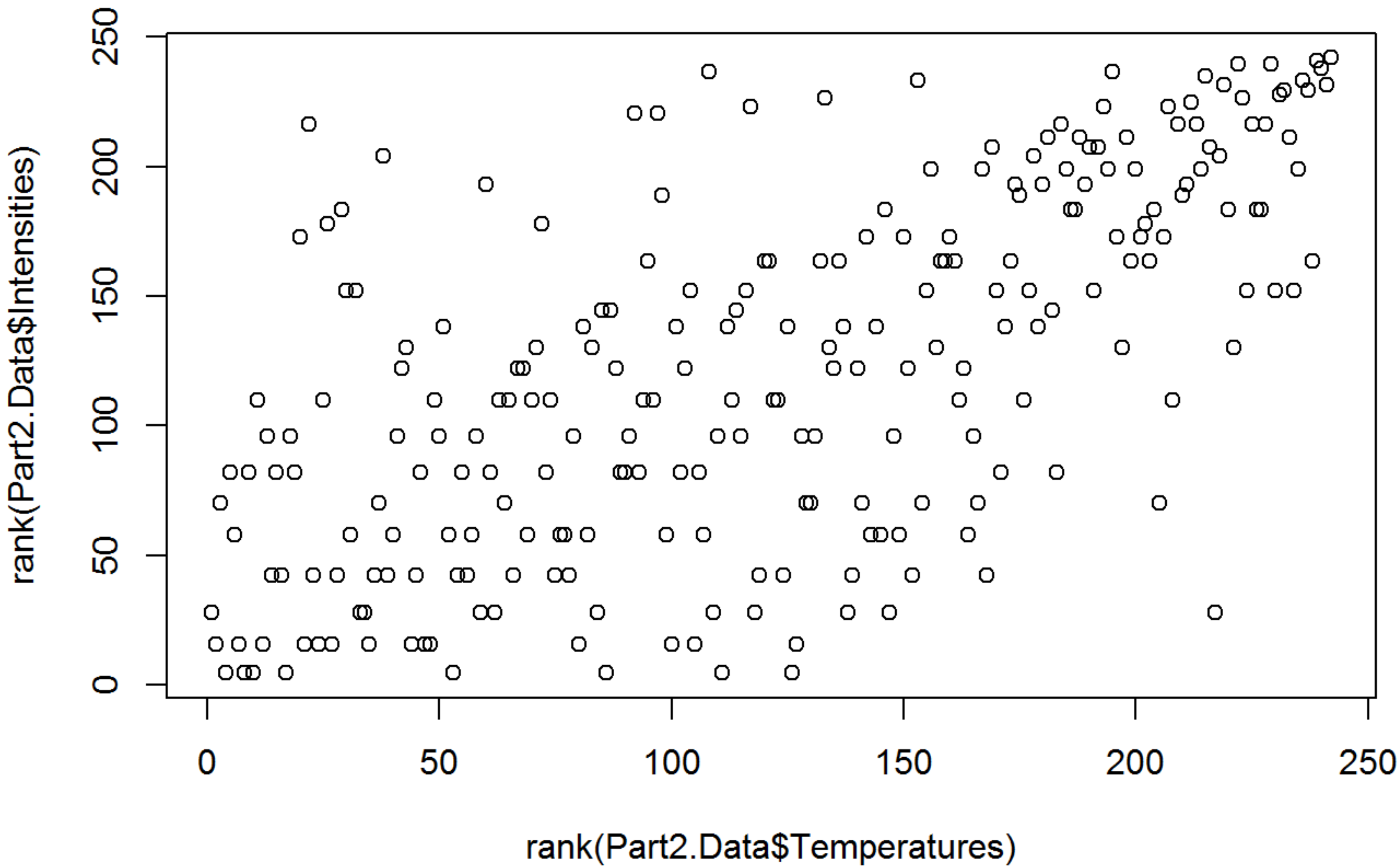
```
plot(Part2.Data$Temperatures,Part2.Data$Intensities)
```



Interpret the plot. What type od relationship you observe? I observe a comonotonic relationship – as temperatures increase, so does the temperature. However, the relationship does not look linear. It appears that the slope increases as the temperature rises.

Empirical copula should be normalized– divide all by n, so all numbersare between 0 and 1.

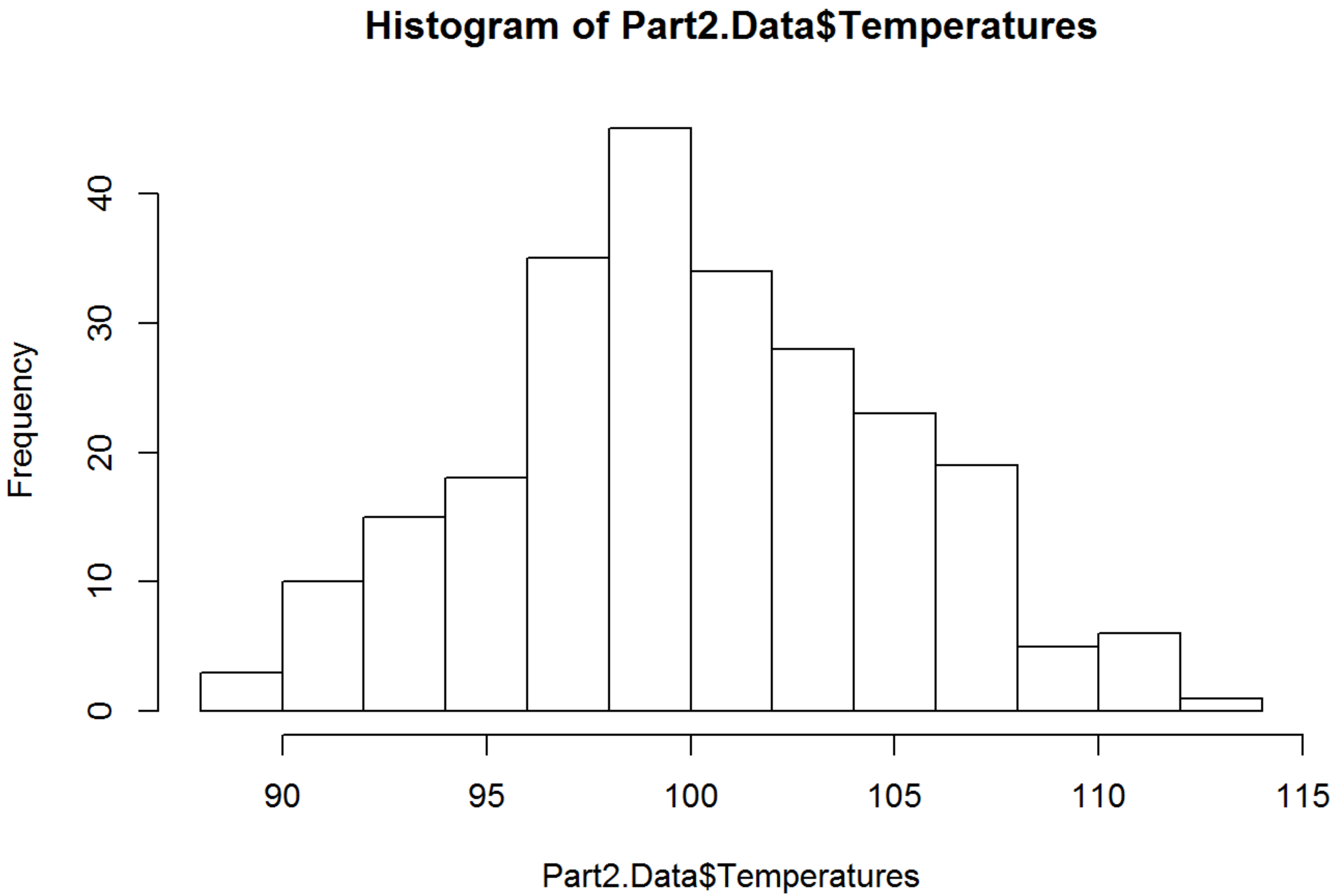
```
plot(rank(Part2.Data$Temperatures),rank(Part2.Data$Intensities))
```



We see comonotonic behavior, and tail dependence – pinched corners. We should investigate this. We are seeing Gumbel's copula (there is upper tail dependence).

What is the distribution of temperatures?

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



The distribution looks fairly normal, however there aren't as many high temperatures as expected.

We test if it is normal.

```
library(MASS)
```

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

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Part2.Data$Temperatures
## D = 0.049, p-value = 0.6071
## alternative hypothesis: two-sided
```

We find little evidence to reject the hypothesis that this data is not a normal distribution.

The copula can show stratified parts of the model. Use clayton because left tail dependence gumball is for right tail dependence

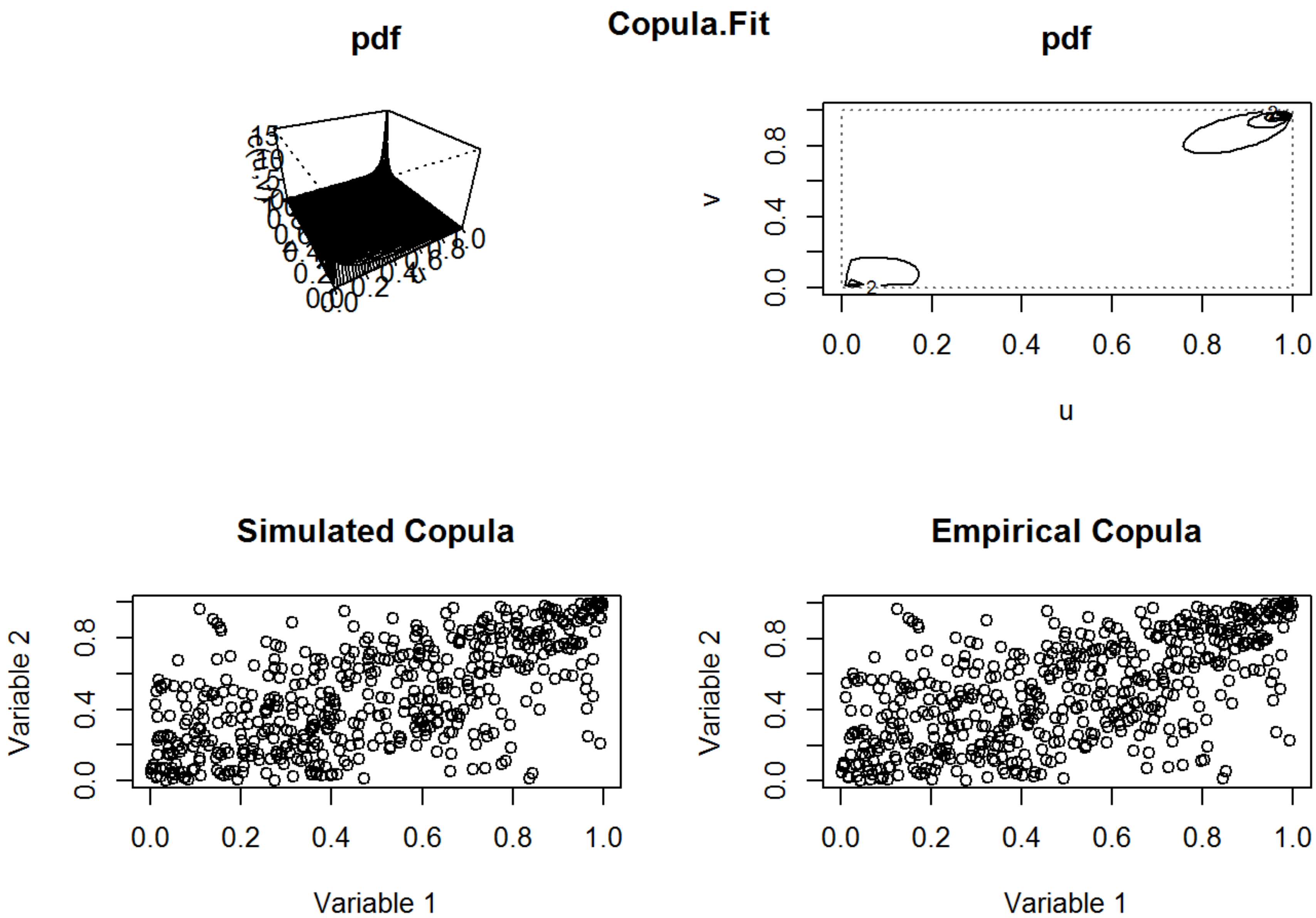
```
library(copula)
```

```
## Warning: package 'copula' was built under R version 3.1.3
```

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

```
## fitCopula() estimation based on 'maximum likelihood'
## and a sample of size 242.
##       Estimate Std. Error z value Pr(>|z|)
## param  1.87746    0.09899   18.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## The maximized loglikelihood is  74.18
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##       11        4
```

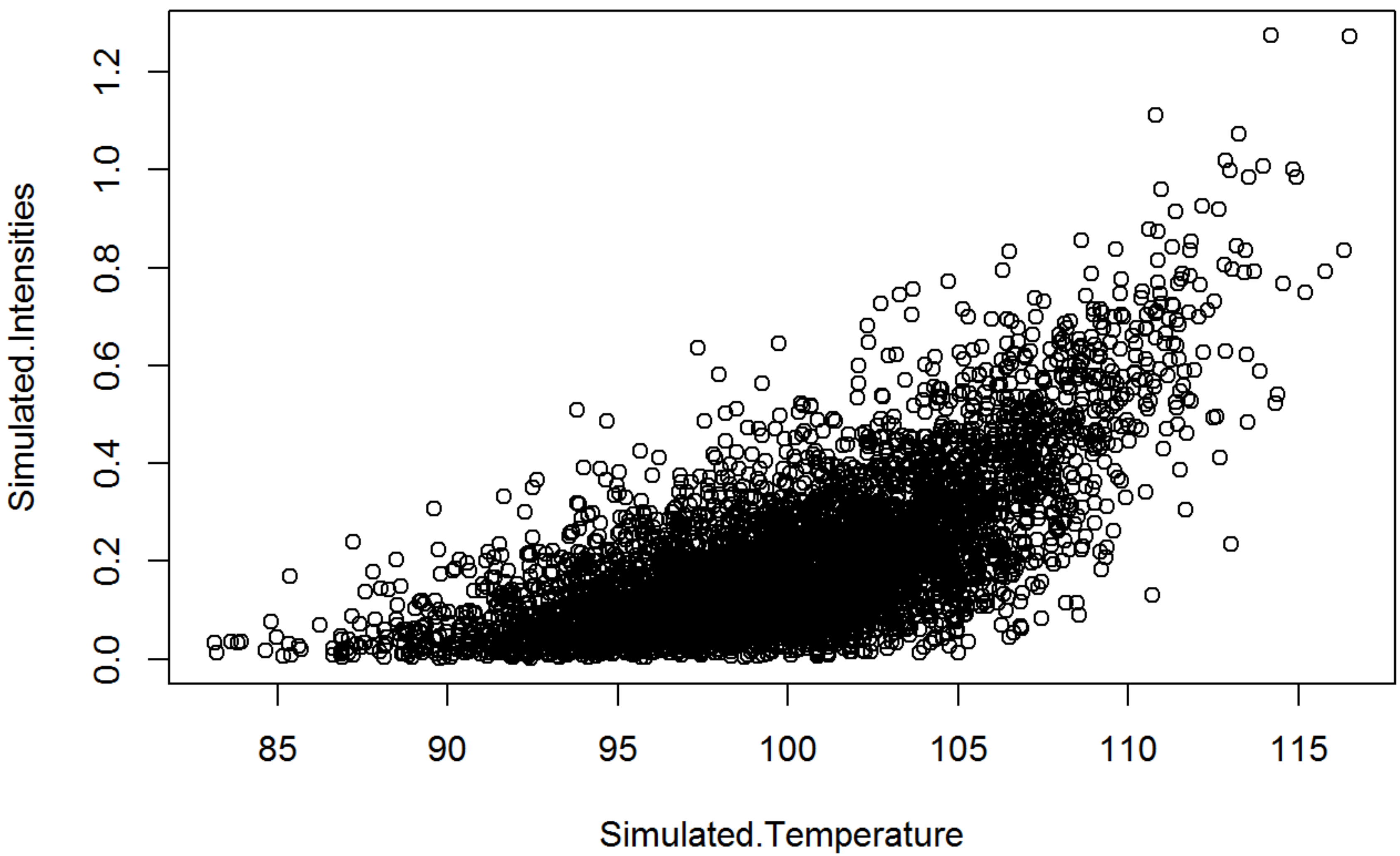
```
par(mfrow=c(2,2))
set.seed(8301735)
Defined.copula<-gumbelCopula(param=summary(Copula.Fit)$coefficients[1,1],dim=2)
persp(Defined.copula, dCopula, main="pdf",xlab="u", ylab="v", zlab="c(u,v)")
contour(Defined.copula,dCopula, main="pdf",xlab="u", ylab="v")
Simulated.Gumbel.Copula<-rCopula(500,Defined.copula)
plot(Simulated.Gumbel.Copula,main="Simulated Copula",xlab="Variable 1",ylab="Variable 2")
plot(apply(Simulated.Gumbel.Copula,2,rank)/length(Simulated.Gumbel.Copula[,1]),main="Empirical Copula",xlab
="Variable 1",ylab="Variable 2")
title("Copula.Fit",outer=TRUE,line=-2)
```



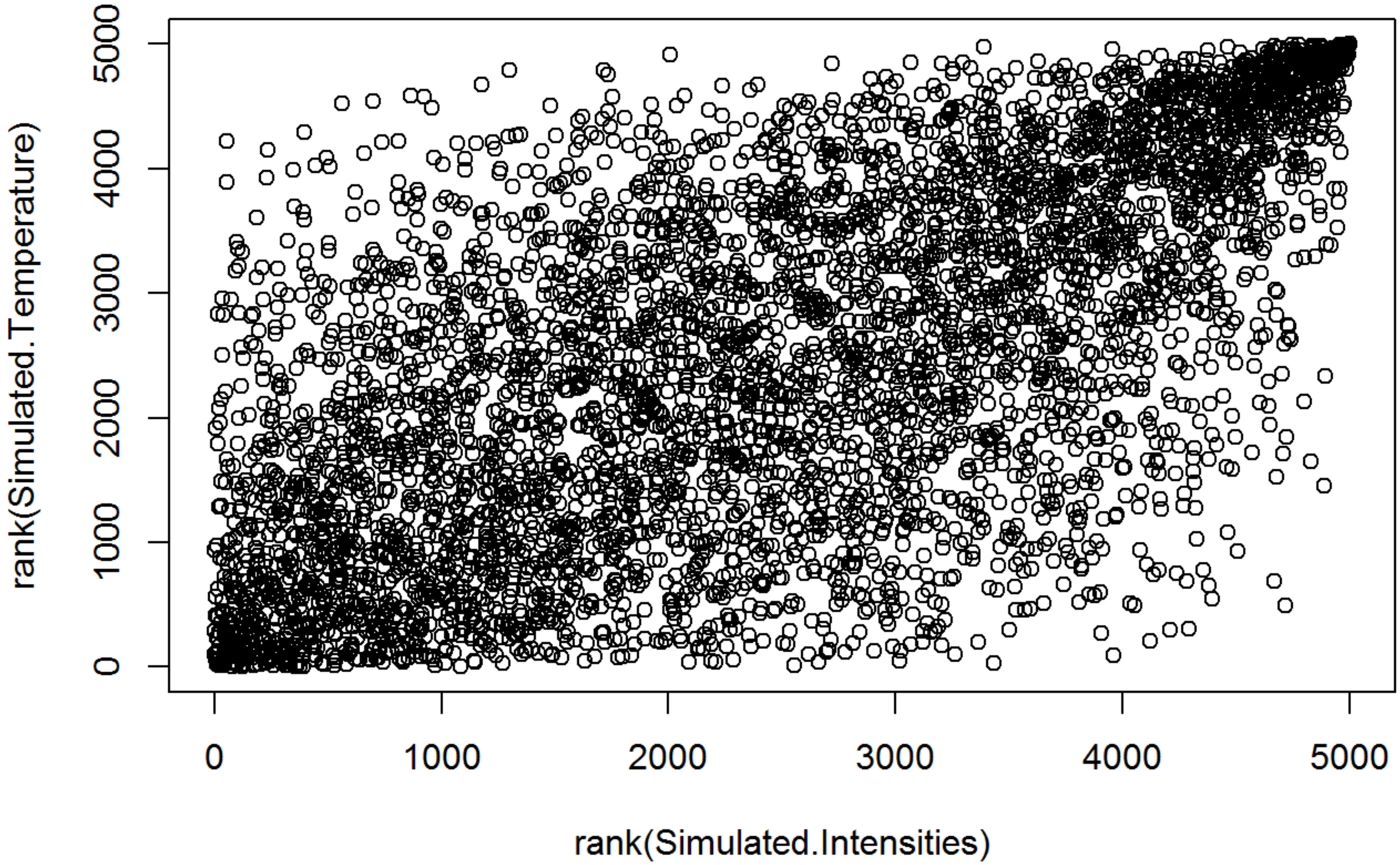

```
Gamma.Rate<- 8.132
Gamma.Shape<-1.656
Fit.Normal<- c(mean(Part2.Data$Temperatures),sd(Part2.Data$Temperatures))
```

WE now how two distributions – one is gamma, one is normal … fit copula and simulate now.

```
par(mfrow=c(1,1))
Simulated.Copula <- rCopula(5000, Defined.copula)
Simulated.Intensities<-qgamma(Simulated.Copula[,1], shape=Gamma.Shape, scale=1/Gamma.Rate)
Simulated.Temperature<-qnorm(Simulated.Copula[,2],Fit.Normal[1],Fit.Normal[2])
plot(Simulated.Temperature,Simulated.Intensities)
```




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



Select simulated pairs with intensity greater than 0.5 and temperature greater than 110. We then fit a model to just the tail events.

```
pairs<-data.frame(intensity=Simulated.Intensities, temp=Simulated.Temperature)
```

```
tail.events<- pairs[Simulated.Intensities>0.5 & Simulated.Temperature > 110,]
```

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

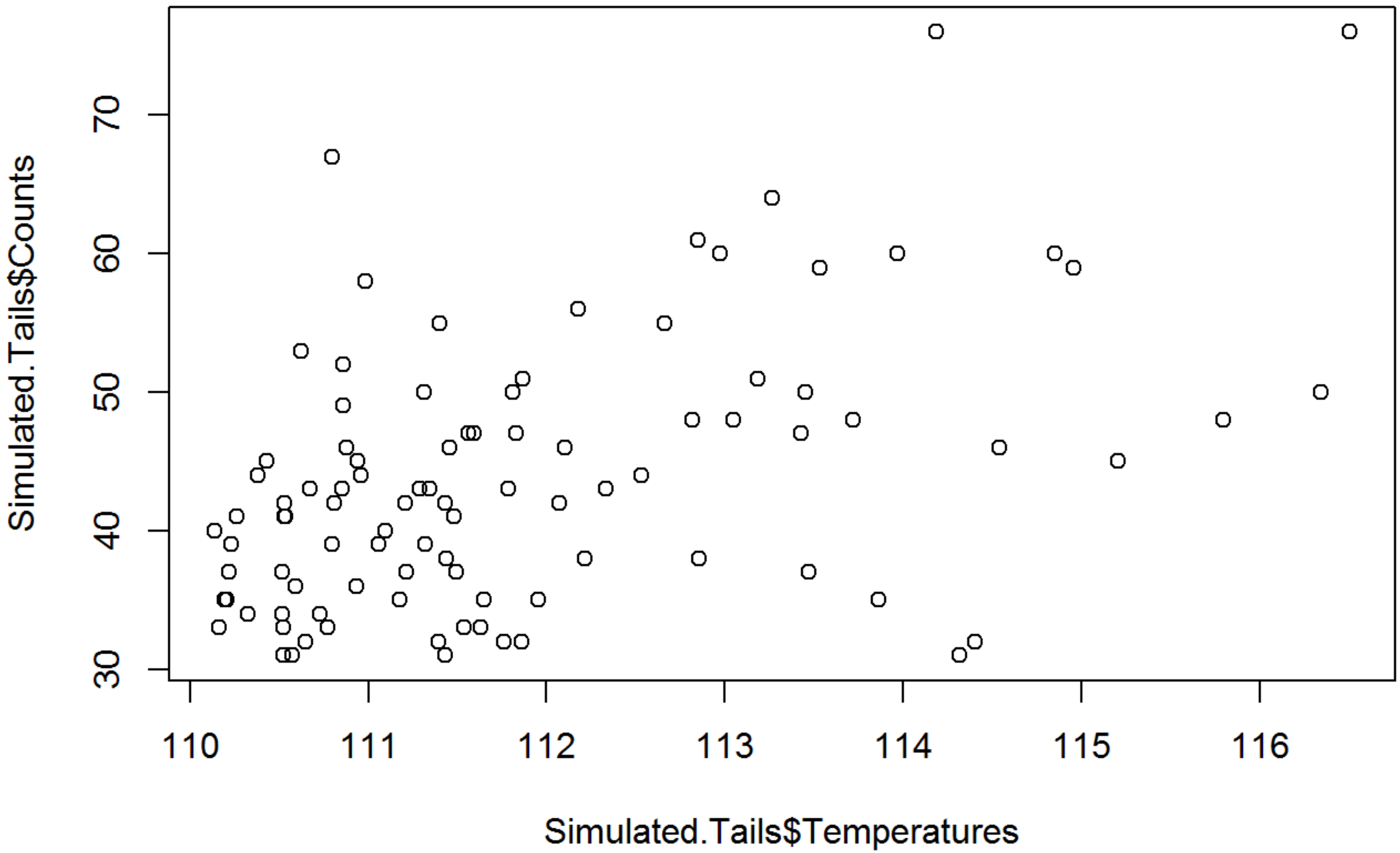
```
summary(main.model <- glm.nb(Counts~Temperatures, data=Part2.Data, init.theta=4.202611755, link=log))
```

```
##
## Call:
## glm.nb(formula = Counts ~ Temperatures, data = Part2.Data, init.theta = 4.202611826,
##     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
```

Then I create a simulated sample for tail events and fit another model to the tail events sample.

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




```
summary(tail.model<-glm.nb(Counts~Temperatures, data=Simulated.Tails, init.theta = 86.27217344, link=log))
```

```
##
```

```
## Call:
## glm.nb(formula = Counts ~ Temperatures, data = Simulated.Tails,
##     init.theta = 71.68488671, link = log)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.41654  -0.78178  -0.02841   0.51680   2.94753
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.81540    1.43435  -2.660  0.00781 **
## Temperatures  0.06783    0.01281   5.295 1.19e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(71.6849) family taken to be 1)
##
##     Null deviance: 119.959  on 94  degrees of freedom
## Residual deviance:  92.145  on 93  degrees of freedom
## AIC: 674.82
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  71.7
##           Std. Err.:  26.7
##
##  2 x log-likelihood:  -668.82
```

In the tail model, the theta parameter is much higher than for the main model, indicating that this data follows a poisson distirubtion closer than the normal model with the bulk fo the data. Additional evidence that poisson model might be suffient is that we have a residual deviance of 67 compared to 65 degress of freedom. This tells me that the tail events look closer to poisson than the regular events. This is common in practice according to the in-class discussions. I could try to use a poisson model to model the tail data, instead of a negative binomial model.

Both models tell me that the relationship between the temperature and the counts is a positive one. I know this because the estimate for both models on temperatures is significant and positive. Both estimates are a little less than .1, so we can say that on average an increase in the temperature by one degree leads to a change in the logs of the counts of .1.