Final Project: Spatial Statistis & Advanced Statistical Modeling

Nathan Oliver

2023-11-23

Initial Set Up

```
suppressMessages(library(fields))
suppressMessages(library(dplyr))
suppressMessages(library( lubridate))
library( LatticeKrig)
library(mapdata)
```

```
## Loading required package: maps
```

```
library(statmod)
```

```
data <- read.csv("CMA_Best_Track_Data.csv")
```

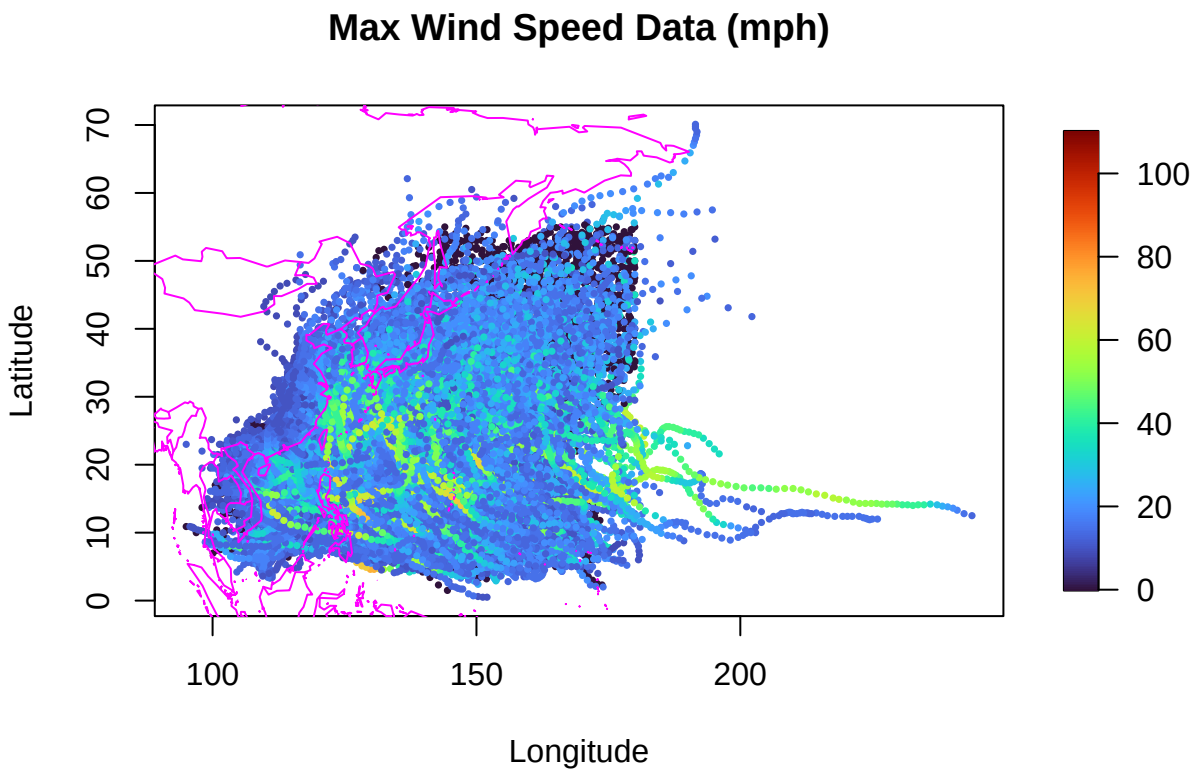
This storm data gathered over many years of the western pacific ocean along the coast of Asia. Many weather stations have been built since data gathering began, so more and more locations have become available over the years. This project focuses on wind speeds that are considered dangerous, so only maximum wind speed will be considered as the primary variable of interest.

To start, some EDA.

Initial Visualizations

```
x <- data$Longitude
y <- data$Latitude
s <- cbind(x,y)
zGrade <- data$Grade
ztmp <- data$Maximum.Wind.Speed
```

```
bubblePlot(s, ztmp, col = turbo(256), size = .5, highlight = FALSE, main = "Max Wind Speed Data (mph)",
           xlab = "Longitude", ylab = "Latitude")
map('world', fill = FALSE, add = TRUE, col = 'magenta')
```



Before we can run any models, we need to get rid of certain duplicates, as both spatialProcess and LatticKrig don't like them.

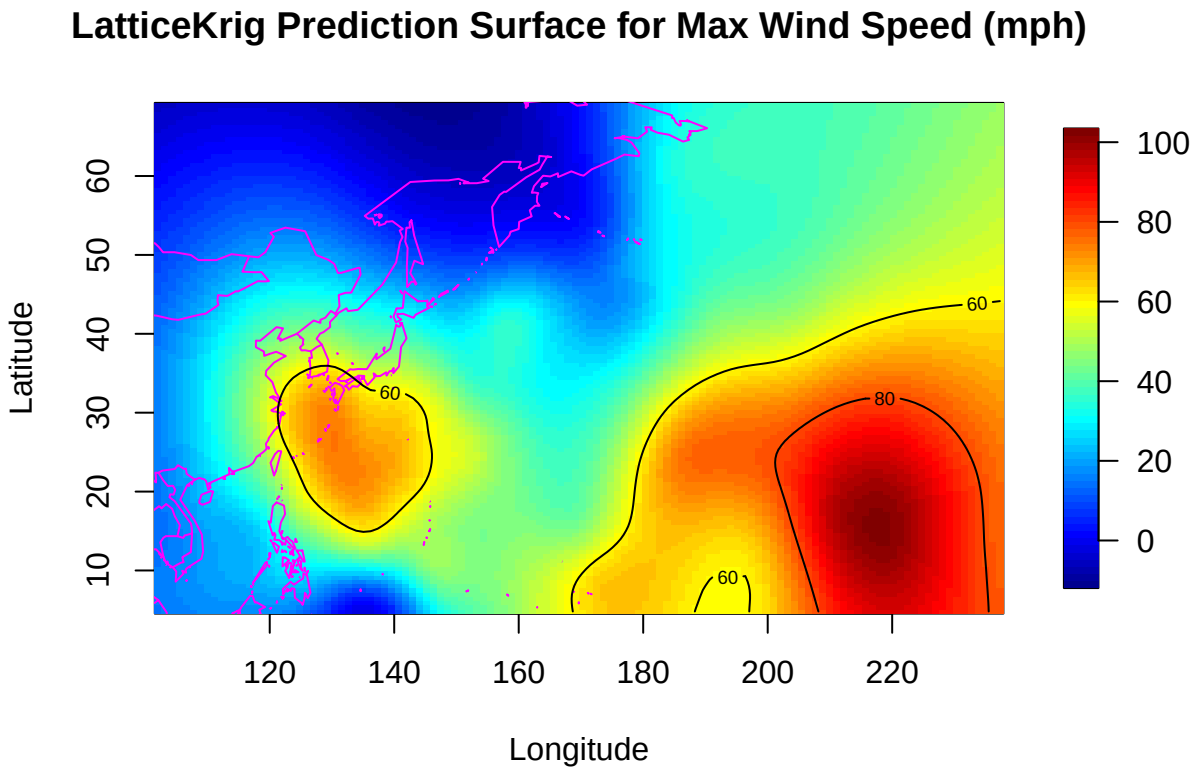
Comparing Spatial Models: LatticeKrig vs spatialProcess

```
data <- data[!duplicated(data$Longitude), ]
#data <- data[!duplicated(data$Latitude), ]
s<- cbind( data$Longitude, data$Latitude)
ztmp <- data$Maximum.Wind.Speed*2.23694

#model1 <- LatticeKrig(s, ztmp, nlevel = 5, findAwght = TRUE, na.rm = TRUE)
```

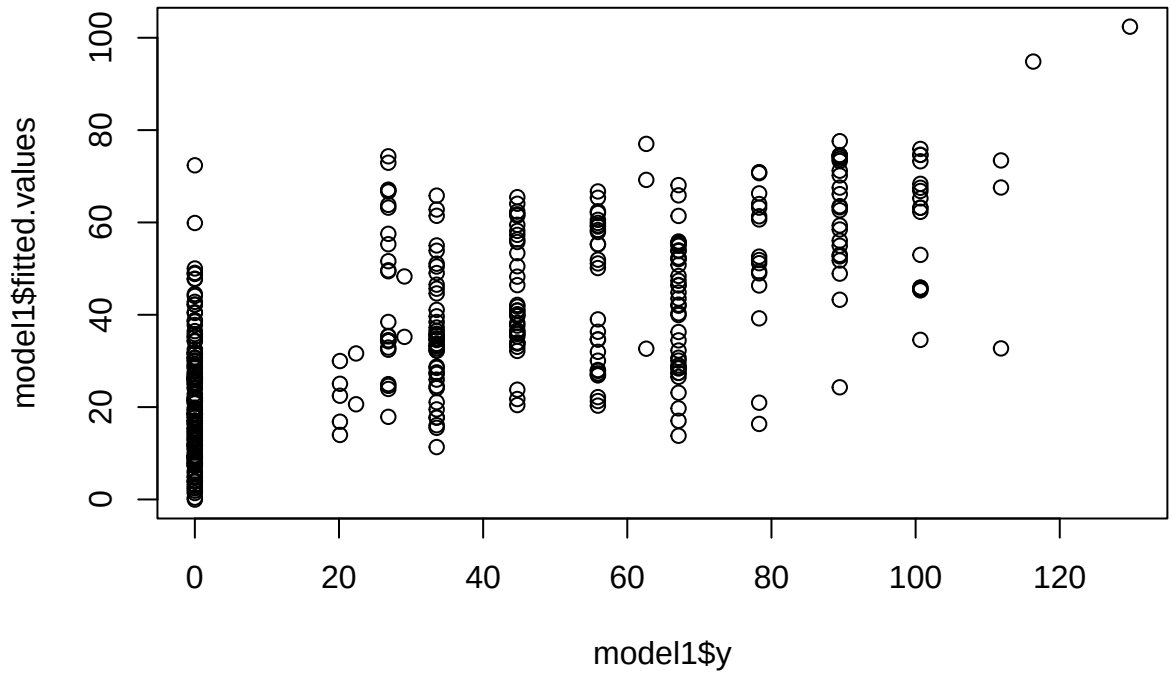
Fitting this model takes awhile, so I did it once and saved it for later. Much for efficient!

```
# RUN <- FALSE
# if(RUN){
#   save(model1, file="myFitLat.rda")
# }
load("myFitLat.rda")
```

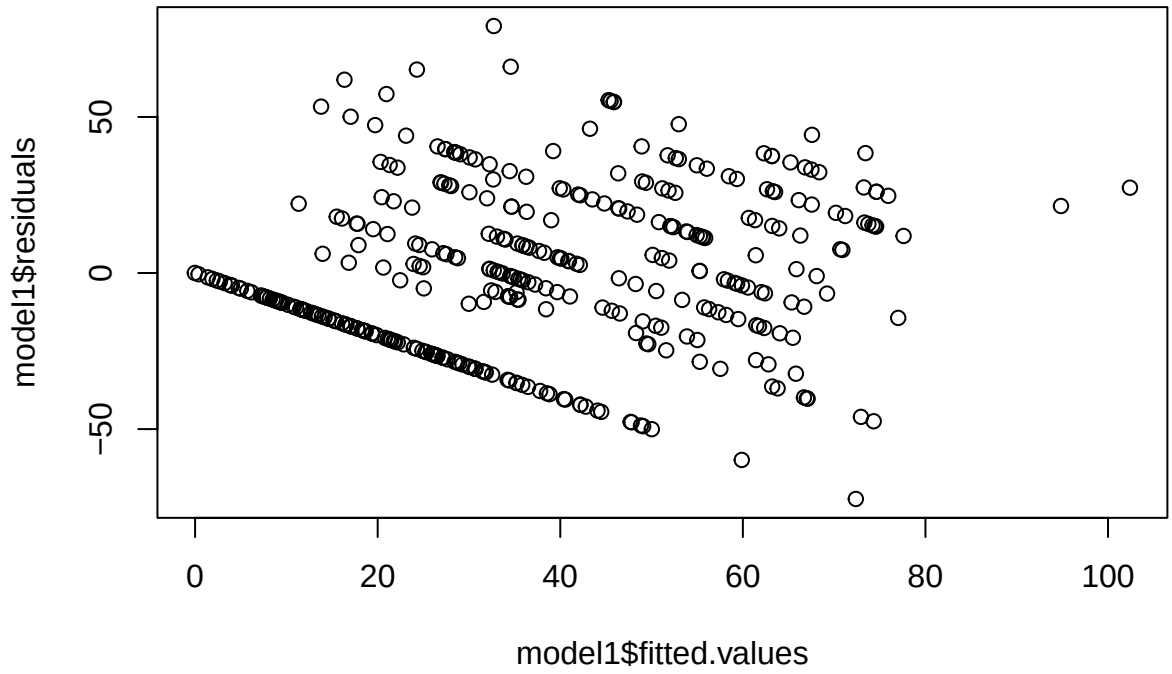
```
#summary(model1)
 fHat <- predictSurface(model1, nx = 120, ny = 120, extrap = TRUE)
 imagePlot(fHat, main = "LatticeKrig Prediction Surface for Max Wind Speed (mph)",
           xlab = "Longitude", ylab = "Latitude")
 map('world', fill = FALSE, add = TRUE, col = 'magenta')
 contour( fHat, add=TRUE, levels=60, col="black")
 contour( fHat, add=TRUE, levels=80, col="black")
```



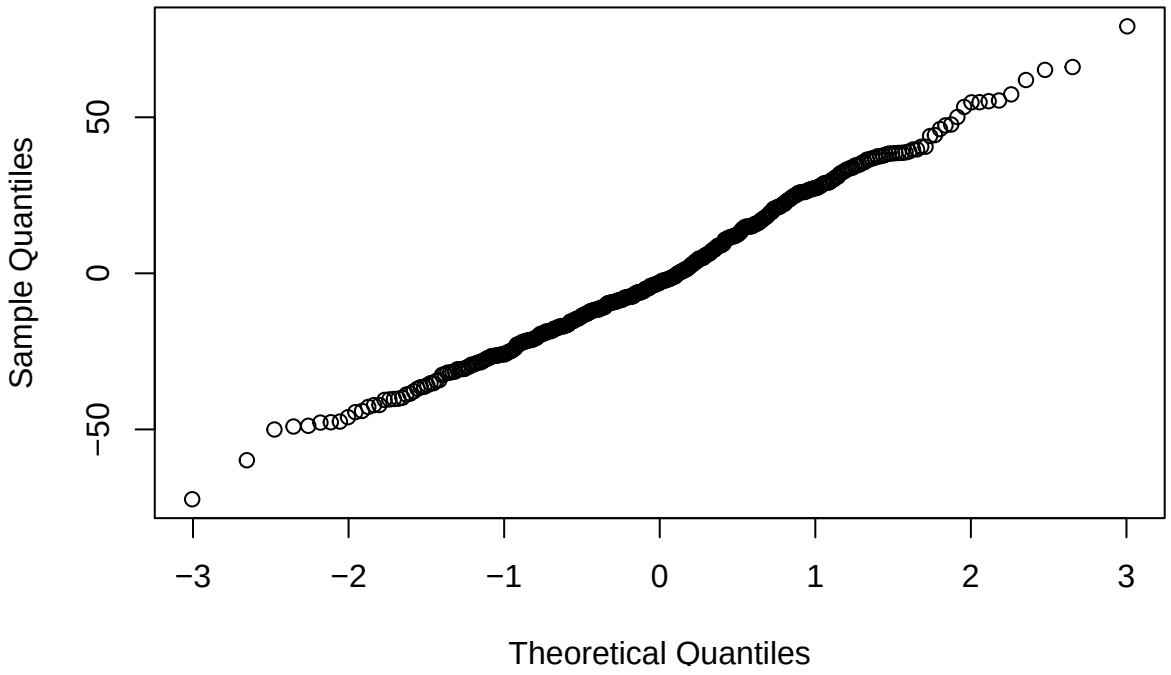

```
#summary(model1)
plot(model1$y, model1$fitted.values)
```




```
plot(model1$fitted.values, model1$residuals)
```



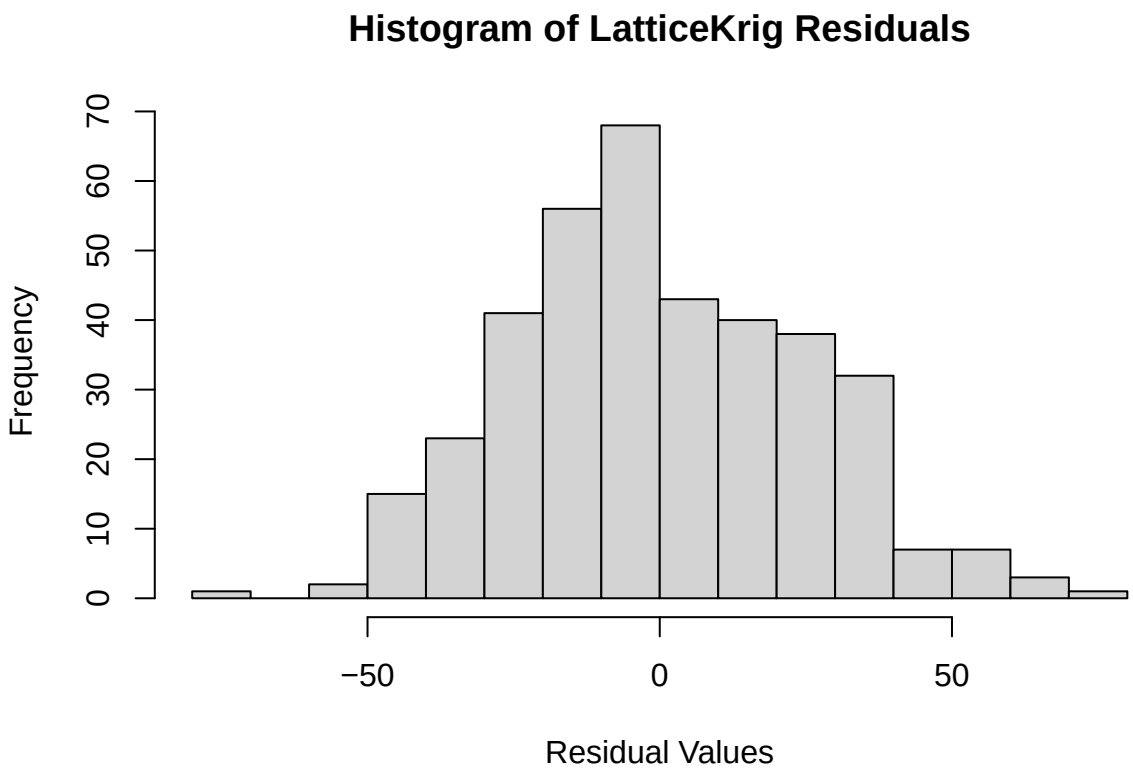

```
qqnorm(model1$residuals)
```




```
res <- model1$residuals
stats(res)
```

```
##                         [,1]
## N                3.770000e+02
## mean            -5.719798e-14
## Std.Dev.         2.519297e+01
## min             -7.237428e+01
## Q1              -1.775570e+01
## median          -2.940494e+00
## Q3               1.764183e+01
## max              7.912915e+01
## missing values   0.000000e+00
```

```
hist(res, breaks = 12, xlab = "Residual Values", main = "Histogram of LatticeKrig Residuals")
```



Overall, nothing here is terribly special. Residuals look a bit normal, but the model clearly favors a certain range of values. There is a clear "shelf" present in the histogram of residuals.

Let's compare to spatialProcess.

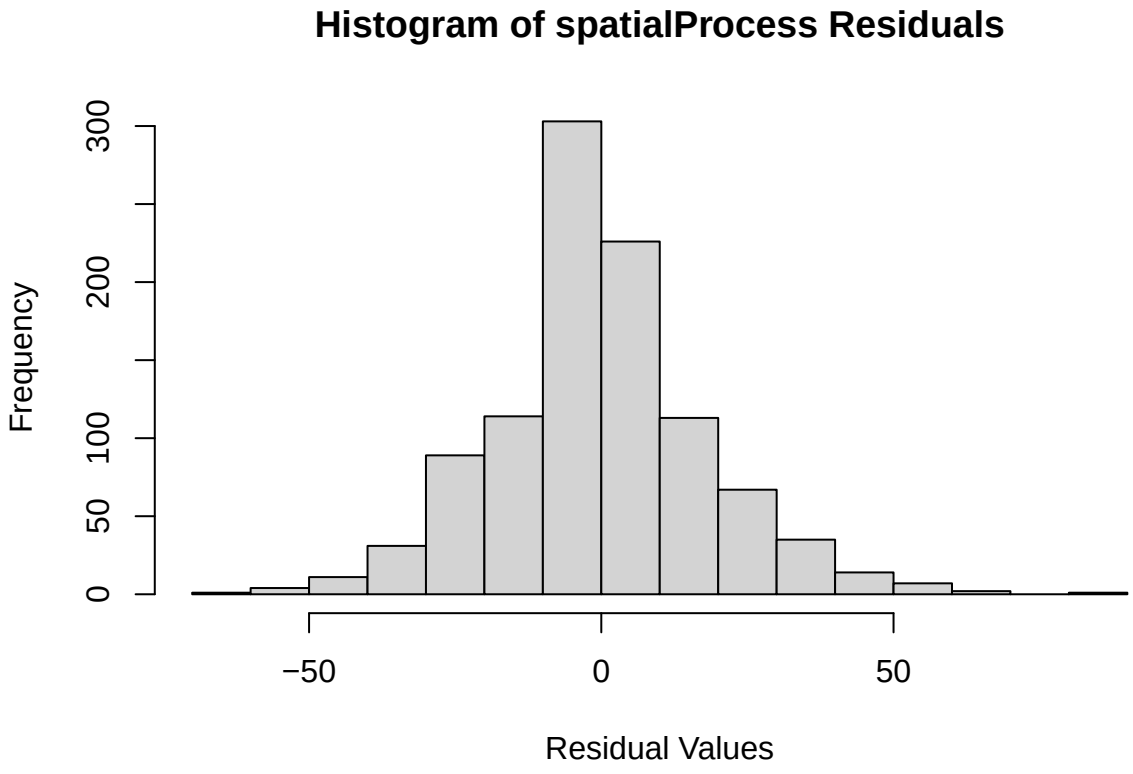
Spatial Process

```
obj3 <- spatialProcess(s, ztmp)
```

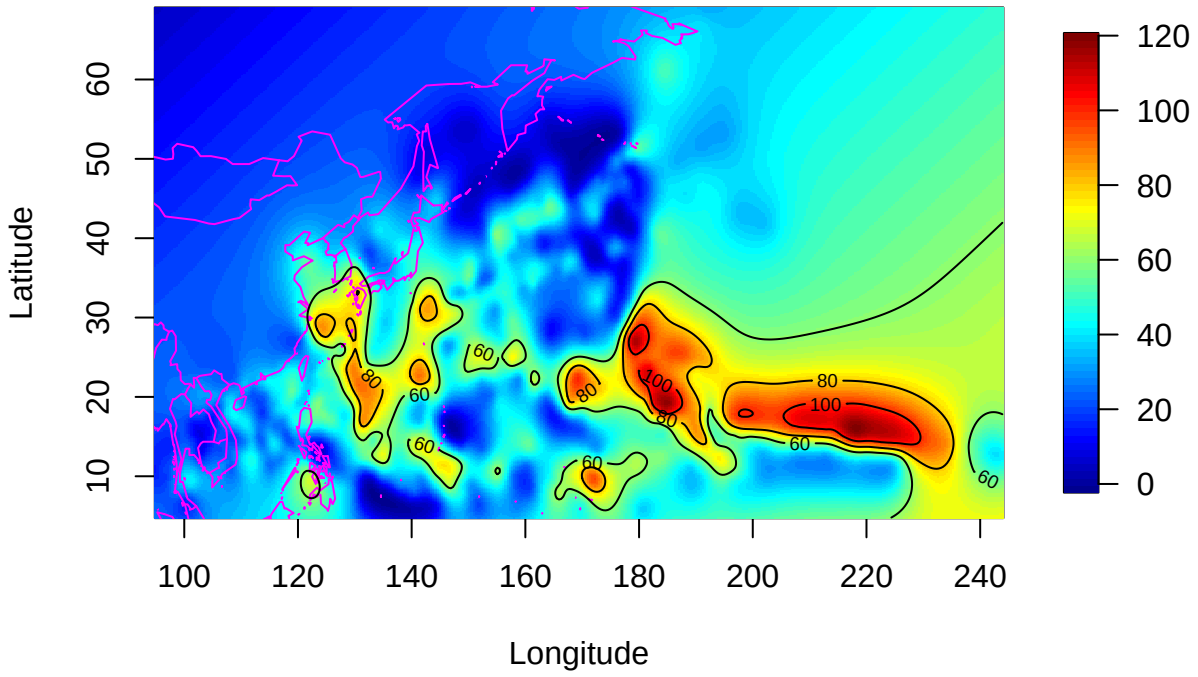
```
res2 <- obj3$residuals
stats(res2)
```

```
##                         [,1]
## N                1.018000e+03
## mean             7.760655e-16
## Std.Dev.         1.810384e+01
## min             -6.844968e+01
## Q1              -9.729409e+00
## median          -9.226195e-01
## Q3               8.946100e+00
## max              8.928803e+01
## missing values   0.000000e+00
```

```
hist(res2, breaks = 12, xlab = "Residual Values", main = "Histogram of spatialProcess Residuals")
```




```
fHat2 <- predictSurface(obj3, nx = 400, ny = 400, extrap = TRUE)
imagePlot(fHat2, main = "SpatialProcess Prediction Surface for Max Wind Speeds (mph)",
          xlab = "Longitude", ylab = "Latitude")
map('world', fill = FALSE, add = TRUE, col = 'magenta')
contour( fHat2, add=TRUE, levels=60, col="black")
contour( fHat2, add=TRUE, levels=80, col="black")
contour( fHat2, add=TRUE, levels=100, col="black")
```




```
obj3$MLESummary
```

```
## lnProfileLike.FULL lnProfileREML.FULL        lnLike.FULL        lnREML.FULL
##      -4811.7234158      -4824.0858044                 NA                 NA
##              lambda                tau             sigma2             aRange
##           0.6341409         21.3983731        722.0641043          2.6012081
##              eff.df                GCV           function           gradient
##         278.7171412        620.8530952         12.0000000          3.0000000
##              lambda             aRange
##           0.6341409          2.6012081
```

Clearly, spatialProcess performs much better. When using it to create a prediction surface, it is much more articulate and defined. It looks to have a tighter fit around the data, and makes less of a "general" estimate.

A Different View: Storm Occurrences Over Time

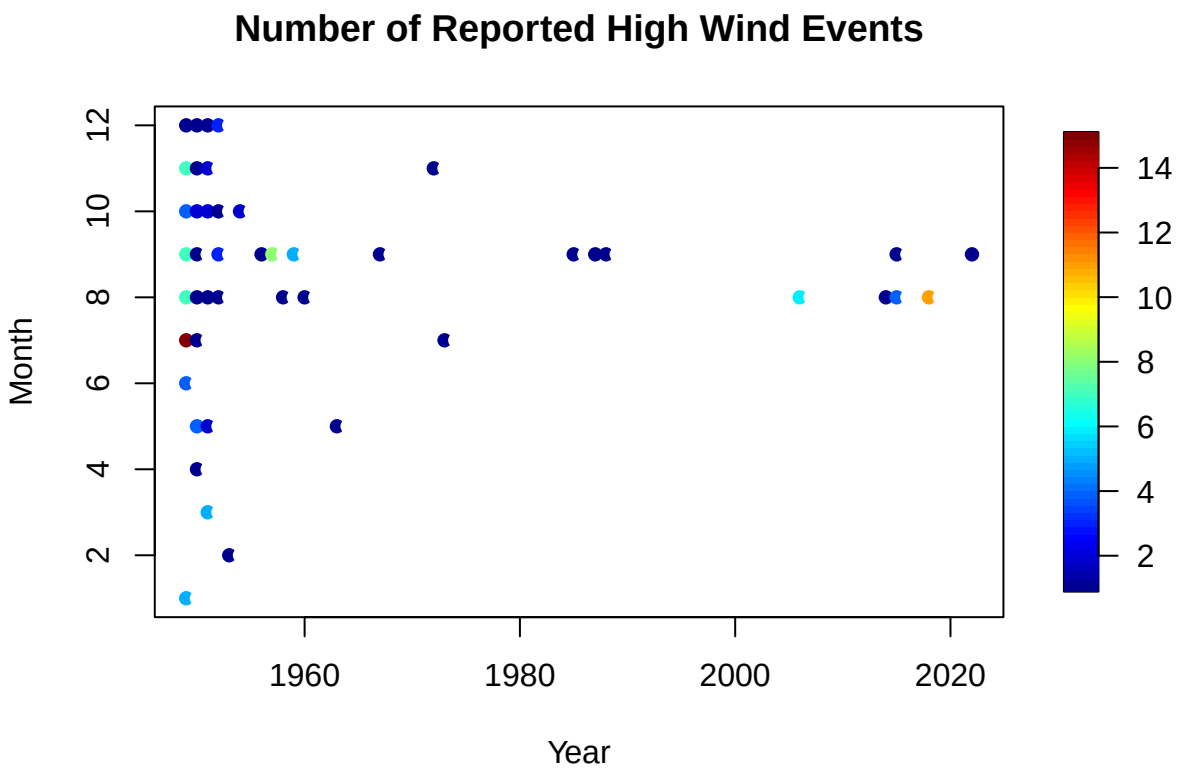
Let's create a new variable that represents the number of storms in a given period of time. This will be called "count".

```
glm_data <- data

glm_data$max_mph <- glm_data$Maximum.Wind.Speed*2.23694
adj_data <- glm_data[which(glm_data$max_mph >= 60),]

adj_data$month <- substr(adj_data$Time, 6, 7)
adj_data$month <- as.numeric(adj_data$month)
adj_data$year <- substr(adj_data$Time, 1, 4)
adj_data$year <- as.numeric(adj_data$year)
adj_data$day <- substr(adj_data$Time, 9, 10)
adj_data$day <- as.numeric(adj_data$day)

mon <- 1:12
yr <-  min(adj_data$year):max(adj_data$year)
N <- length( yr)
mDay <- countDay <- matrix(0, nrow=12, ncol=N)


for(j in 1:12 ){
  for( k in 1:N ){

     ind <- adj_data$year == yr[k] & adj_data$month == j

     dayLabels <- unique(adj_data$day[ind])
     countDay[j,k] <- length(dayLabels)

     mDay[j,k] <- days_in_month( ymd(paste(yr[k], j, 1, sep="-" )))
  }
}

count <- c(countDay)
year <- rep( yr, each= 12)
month <- rep(1:12, length(yr))

AsiaMonthlyCyclones <- data.frame(year=year, month=month, count=count,
                                  m=c(mDay))
```

Visualization of Count Data

The figure below shows the number of high wind speed recordings over each month of the year for all years in the data set. This ranges from 1949 to present day. From what we can see, there are high winds year round, but months August through October tend to have much more observations. This is a little late in the year, but this timeline does include the later end of cyclone season. This indicates that there may be some predictability of when we can expect high wind speeds.

```
tmp<- AsiaMonthlyCyclones$count
tmp[ tmp==0]<- NA
bubblePlot(AsiaMonthlyCyclones$year, AsiaMonthlyCyclones$month,
  tmp, xlab="Year", ylab="Month", col= tim.colors() , highlight=FALSE, size=1 )
title("Number of Reported High Wind Events")
```



Initial Model

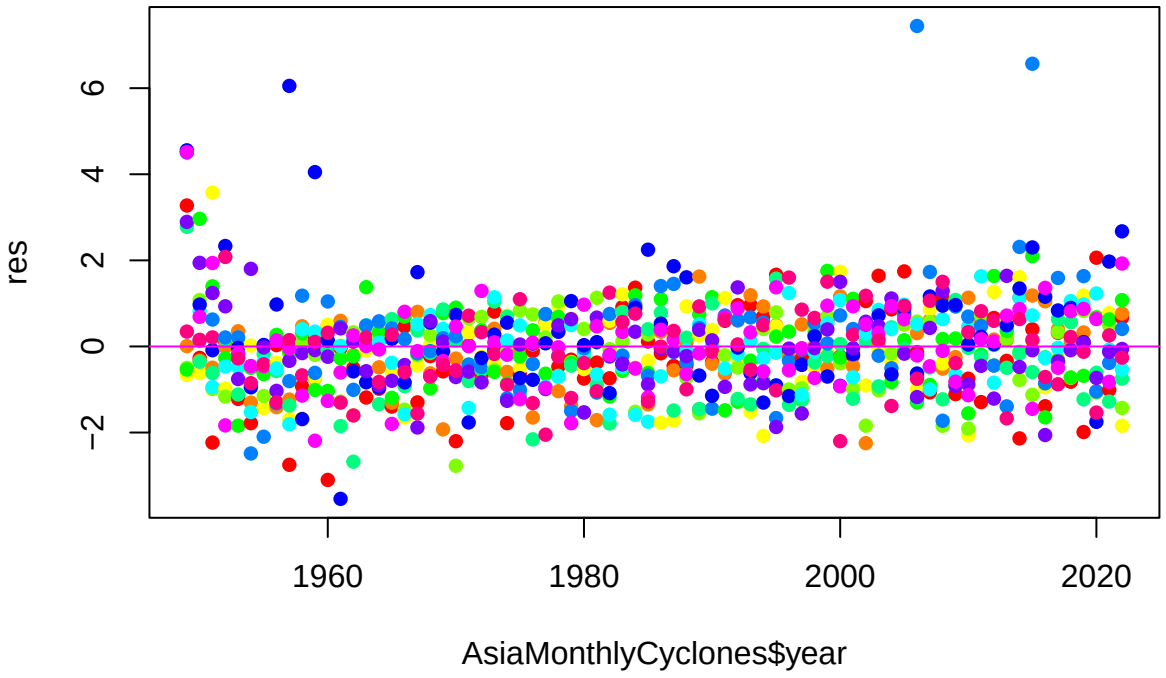
This model appears to be extremely statistically significant, but the AIC value is very high. This could be concerning in the future.

```
GLMFit1 <- glm(cbind(count, (m - count)) ~ year, family = binomial(), data = AsiaMonthlyCyclones)
summary(GLMFit1)
```

```
##
## Call:
## glm(formula = cbind(count, (m - count)) ~ year, family = binomial(),
##     data = AsiaMonthlyCyclones)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) 117.881629  12.009748   9.815   <2e-16 ***
## year         -0.062429   0.006115 -10.209   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 922.02  on 887  degrees of freedom
## Residual deviance: 766.65  on 886  degrees of freedom
## AIC: 886.18
##
## Number of Fisher Scoring iterations: 7
```

Residuals appear to be somewhat normal but there does seem to be unequal variance given the range is not symmetrical. There does not appear to be any fanning or cornucopia behavior though.

```
res <- qres.binom(GLMFit1)
plot(AsiaMonthlyCyclones$year, res, col = rainbow(12), pch = 16)
abline( h = 0, col = 'magenta')
```



Attempting A Full Seasonal Model

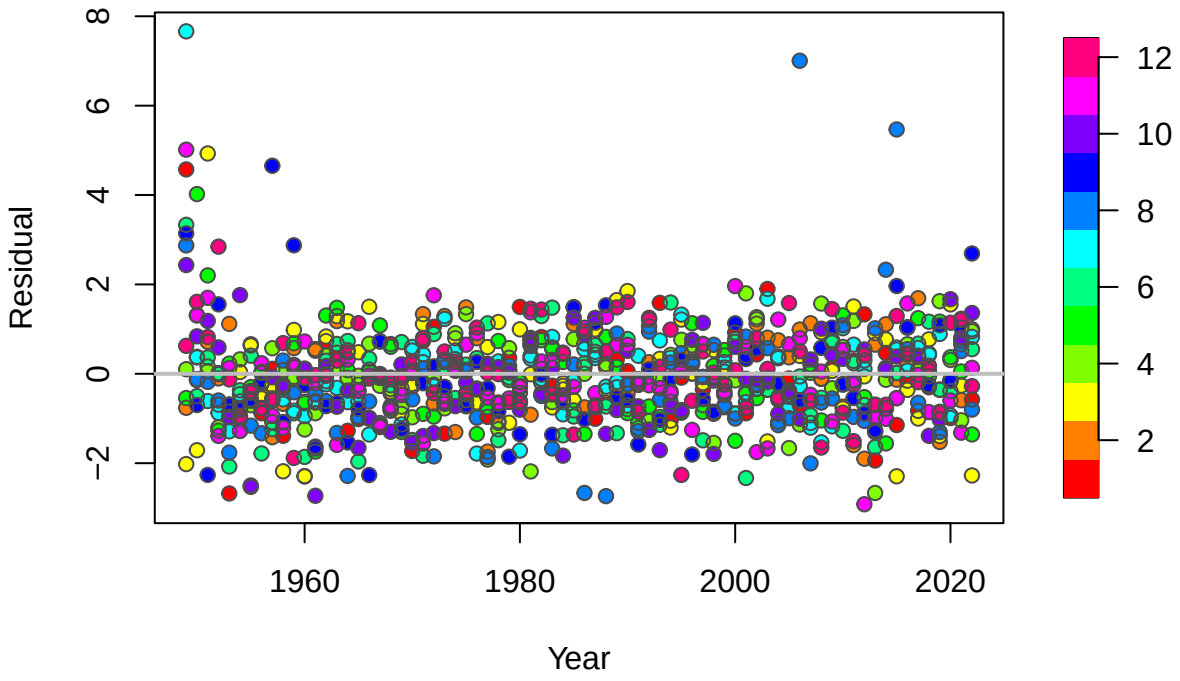
```
SC <- cbind(
  sin(2*pi*AsiaMonthlyCyclones$month/12),
  cos(2*pi*AsiaMonthlyCyclones$month/12),
  sin(2*pi*AsiaMonthlyCyclones$month/6),
  cos(2*pi*AsiaMonthlyCyclones$month/6)
  )


GLMFit2 <- glm(
  cbind(AsiaMonthlyCyclones$count,
                      AsiaMonthlyCyclones$m - AsiaMonthlyCyclones$count) ~
                  year + SC, family=binomial(),
                  data= AsiaMonthlyCyclones)
summary(GLMFit2 )
```

```
##
## Call:
## glm(formula = cbind(AsiaMonthlyCyclones$count, AsiaMonthlyCyclones$m -
##     AsiaMonthlyCyclones$count) ~ year + SC, family = binomial(),
##     data = AsiaMonthlyCyclones)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 118.15372   12.05826   9.799  < 2e-16 ***
## year         -0.06276    0.00614 -10.222  < 2e-16 ***
## SC1          -1.14236    0.17374  -6.575 4.86e-11 ***
## SC2          -0.14874    0.16826  -0.884   0.3767
## SC3           0.29747    0.14914   1.995   0.0461 *
## SC4          -0.21634    0.14823  -1.460   0.1444
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 922.02  on 887  degrees of freedom
## Residual deviance: 672.97  on 882  degrees of freedom
## AIC: 800.5
##
## Number of Fisher Scoring iterations: 7
```

This model seems to be very effective. We can see that the variables are all extremely statistically significant and the standard error is decently low. The AIC could be lower but it is not concerning given the rest of the results. This AIC value and the standard error values are lower than the initial model which is a sign of improvement. This implies that the model is statistically significant.

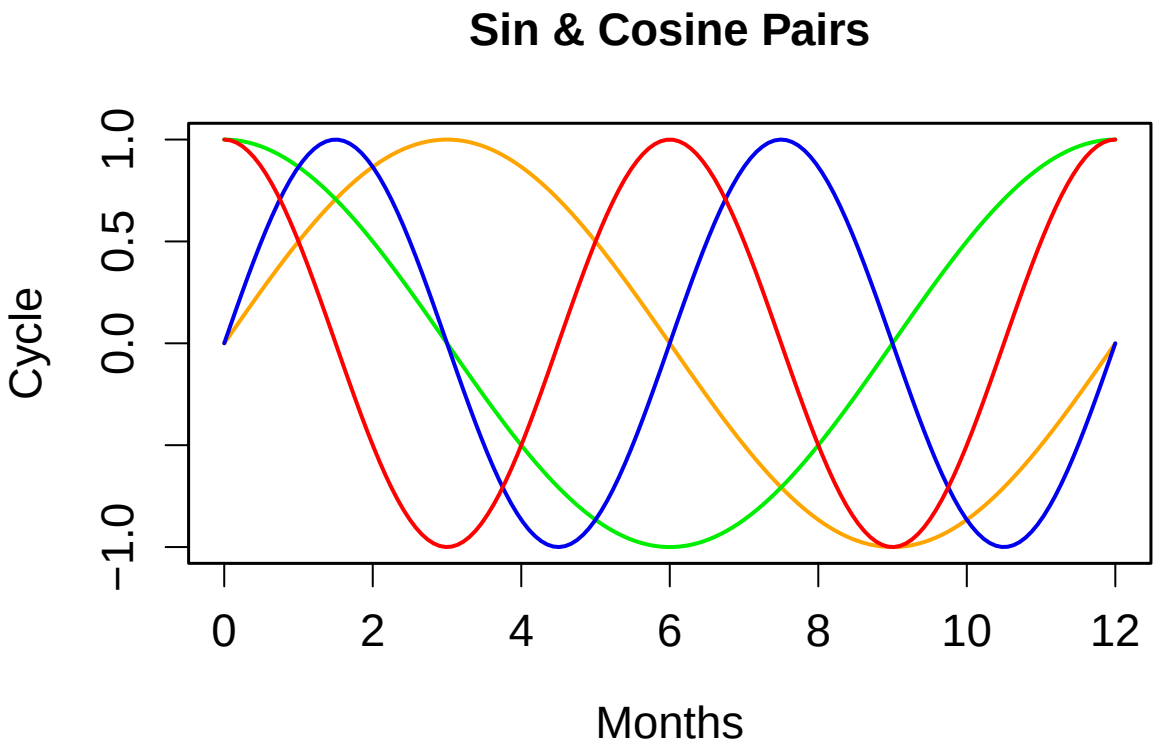
The residuals for this plot appear to be better compared to the initial residual plot. The majority of the data lies within the the range of -4 to 4 which is promising. There are quite a few outliers which could be concerning. Overall, though, the model has better behavior than the previous model and could have larger predictive power.

```
res2 <- qres.binom(GLMFit2)
bubblePlot(AsiaMonthlyCyclones$year, res2, AsiaMonthlyCyclones$month, col = rainbow(12), xlab = 'Year',
abline(h = 0, col = 'grey', lwd = 2)
```



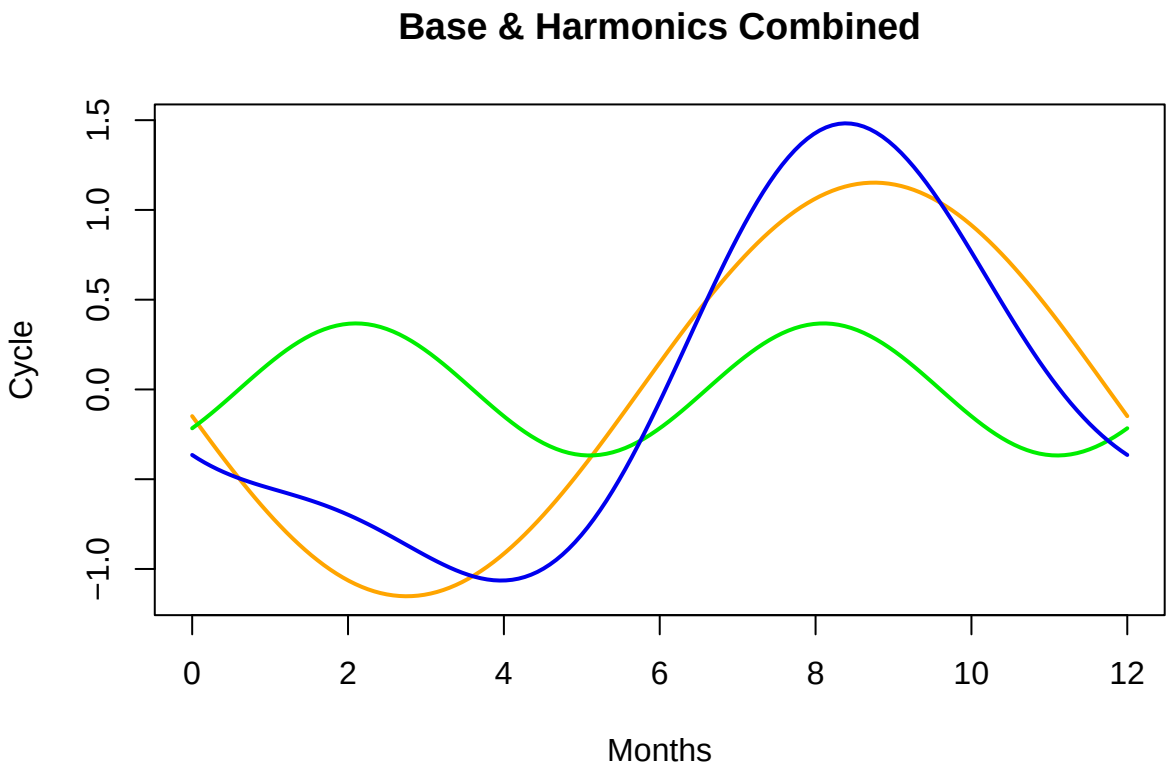
Visualization of the Seasonal Model

```
tm<- seq( 0, 12, length.out=150)
S1<- sin(2*pi*tm/12)
C1<- cos(2*pi*tm/12)
S2<- sin(4*pi*tm/12)
C2<- cos(4*pi*tm/12)
SCAmp<- GLMFit2$coefficients[3: 6]
#print( SCAmp)
fields.style()
matplot( tm,cbind(S1,C1,S2,C2 ), type="l", xlab="Months", ylab="Cycle",
    lty=1, lwd=2)
title("Sin & Cosine Pairs")
```




```
base<- S1*SCAmp[1] + C1*SCAmp[2]
harm<- S2*SCAmp[3] + C2*SCAmp[4]
combined<- base +harm
```

```
matplot( tm,cbind(base, harm, combined ), type="l", xlab="Months",
   ylab="Cycle", lty=1, lwd=2)
title("Base & Harmonics Combined")
```



Validation of the Model

To check the validity of the seasonal model, we performed likelihood testing. Here we see that the log likelihood is much larger than the chi square value, indicating that we should reject.

```
LFull <- logLik(GLMFit2)
LTrend <- logLik(GLMFit1)

2*(LFull - LTrend)
```

```
## 'log Lik.' 93.68201 (df=6)
```

```
qchisq(.95,4)
```

```
## [1] 9.487729
```