scratchwork oliver

Adi Arifovic, Oliver Baldwin Edwards, Leah Johnson

06 May, 2020

Predicting Hurricane Deaths

Group 2: Oliver Baldwin Edwards, Leah Johnson, Adi Arifovic

table of contents here

Purpose

We hope to predict the number of hurricane deaths based on variables such as maximum sustained windspeed, atmospheric pressure, and property damage from a dataset containing data on 94 named hurricanes that made landfall in the US mainland from 1950 through 2012. We would expect the number of deaths to increase as the maximum sustained windspeed and property damage increases, since it would make sense for more destructive hurricanes to also be more deadly.

Data

Source

We plan to use the dataset "hurricNamed" from the "DAAG" R package containing data on 94 named hurricanes that made landfall in the US mainland from 1950 through 2012. It contains information on the number of deaths, the name of the hurricane, the year of the hurricane, the damage caused by the hurricane, etc. The data is sourced from multiple places and was used in a research paper claiming that hurricanes with female names did more human damage (after adjusting for the severity of the storm) than those with male names. Therefore, the data seems fairly reliable.

Response Variables

Our response variable is the number of log_deaths that occurred due to a hurricane. The units are number of human deaths. We observe that the range of deaths are from 0 to 1836.

Explanatory Variables

The explanatory variables we will examine are LF.WindsMPH, LF.PressureMB, LF.times, BaseDam2014, NDAM2014, and deaths.

1. LF.WindsMPH describes the maximum sustained windspeed for each hurricane in miles per hour
2. LF.PressureMB describes the atmospheric pressure at landfall in millibars
3. LF.times describes the number of times the hurricane made landfall

4. BaseDam2014 describes the property damage caused by the hurricane in millions of 2014 US dollars
5. NDAM2014 describes the amount of damage the hurricane caused had it appeared in 2014 (no units given)
6. deaths describes the number of human deaths the hurricane caused

```
# load in full dataset
full_hurricane_df <- hurricNamed
glimpse(full_hurricane_df)
```

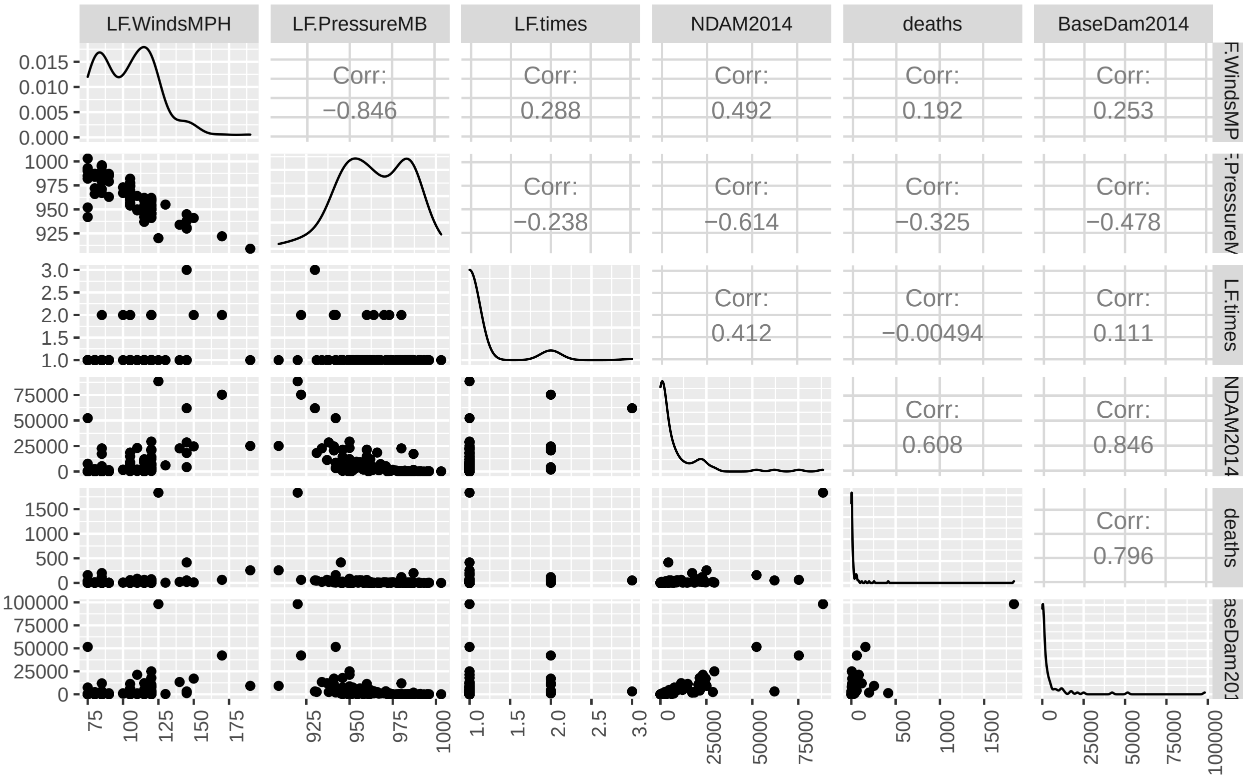
```
## Rows: 94
## Columns: 12
## $ Name           <chr> "Easy", "King", "Able", "Barbara", "Florence", "Caro...
## $ Year           <int> 1950, 1950, 1952, 1953, 1953, 1954, 1954, 1954, 1955...
## $ LF.WindsMPH    <int> 120, 130, 85, 85, 85, 120, 120, 145, 120, 85, 120, 1...
## $ LF.PressureMB  <int> 958, 955, 985, 987, 985, 960, 954, 938, 962, 987, 96...
## $ LF.times       <int> 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1...
## $ BaseDamage     <dbl> 3.3000, 28.0000, 2.7500, 1.0000, 0.2000, 460.2275, 4...
## $ NDAM2014       <dbl> 1870, 6030, 170, 65, 18, 21375, 3520, 28500, 2270, 1...
## $ AffectedStates <chr> "FL", "FL", "SC", "NC", "FL", "NC,NY,CT,RI", "MA,ME"...
## $ firstLF        <date> 1950-09-04, 1950-10-17, 1952-08-30, 1953-08-13, 195...
## $ deaths         <int> 2, 4, 3, 1, 0, 60, 20, 20, 0, 200, 7, 15, 416, 1, 0,...
## $ mf             <fct> f, m, m, f, f, f, f, f, f, f, m, f, f, f, f, f, f, f...
## $ BaseDam2014    <dbl> 32.419419, 275.073859, 24.569434, 8.867416, 1.773483...
```

```
# remove non numeric and BaseDamage for ggpairs call
hurricane_numeric <- full_hurricane_df %>%
  select(-c(Name, AffectedStates, firstLF, mf, BaseDamage, Year))

# removing death outlier
hurricane_death_outlier <- hurricane_numeric %>%
  filter(deaths != 1836)

# applying a log transformation to deaths
hurricane_logdeaths <- hurricane_numeric %>%
  mutate(log_deaths = ifelse(deaths == 0, 0, log(deaths))) %>%
  mutate(NDAM2014 = log(NDAM2014)) %>%
  select(-deaths)
```

Exploratory Data Analysis

GGpairs calls

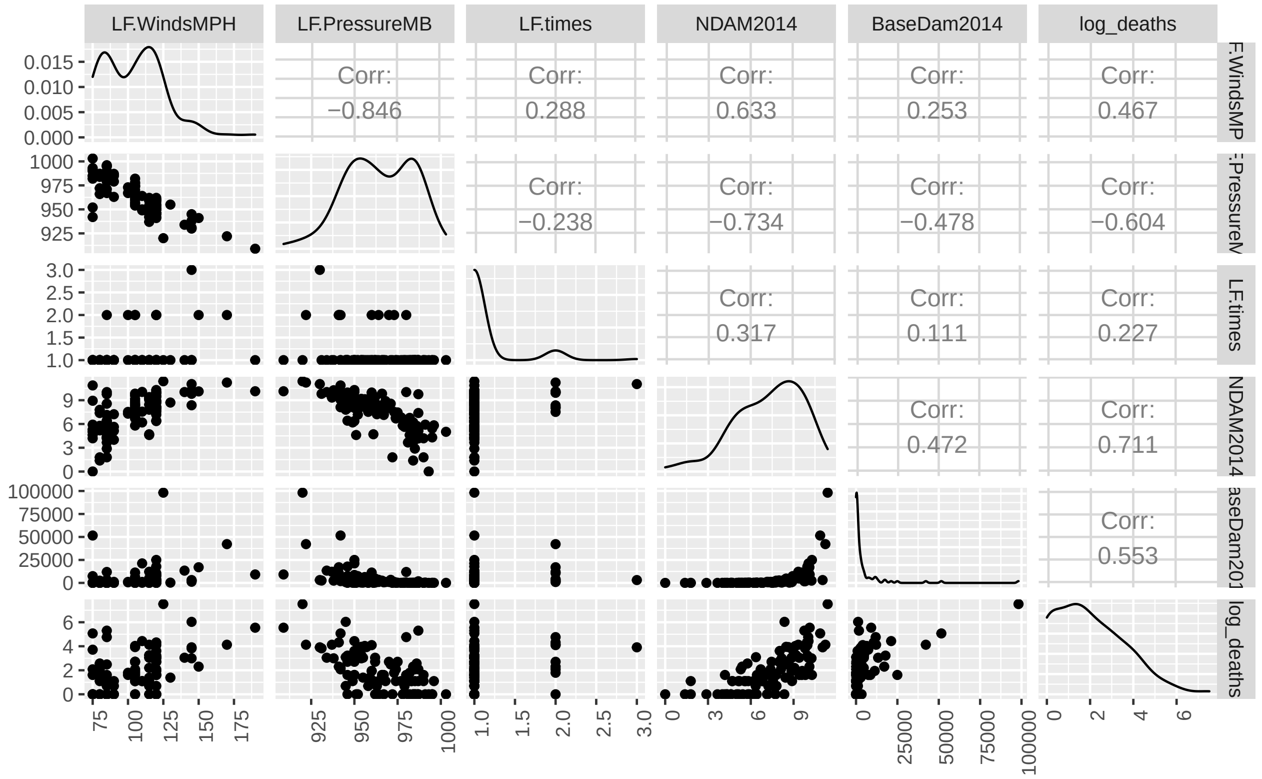
```
ggpairs(hurricane_numeric) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



talk about ggpairs here and how deaths looks bad

now we try log transformation on deaths:

```
ggpairs(hurricane_logdeaths) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



talk about above ggpairs calls here

correlation test?

Put a sentence explaining that we want kendalls over spearmans since more robust setup hypothesis test for posstivie association $\tau > 0$

```
# associated
cor.test(x=hurricane_logdeaths$NDAM2014, y=hurricane_logdeaths$log_deaths, method="kendall")
```

```
##
##  Kendall's rank correlation tau
##
## data:  x and y
## z = 7.9679, p-value = 1.615e-15
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5783972
```

```
cor.test(x=hurricane_logdeaths$LF.times, y=hurricane_logdeaths$log_deaths, method="kendall")
```

```
##
##  Kendall's rank correlation tau
##
## data:  x and y
## z = 2.3849, p-value = 0.01708
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.2094088
```

```
cor.test(x=hurricane_logdeaths$LF.PressureMB, y=hurricane_logdeaths$log_deaths, method="kendall")
```

```
##
##  Kendall's rank correlation tau
##
## data:  x and y
## z = -5.8914, p-value = 3.829e-09
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.4306761
```

```
cor(x=hurricane_logdeaths$NDAM2014, y=hurricane_logdeaths$log_deaths, method="kendall")
```

```
## [1] 0.5783972
```

low pvalue so reject the null, therefore they are positively associated

Kolmogorov Smirnov Test

kolmogorov smirnov test (with empirical CDF(s) plotted) LEAH

- between our linear regression model residuals and a normal distribution

Hypotheses:

$H_0 : F(t) = F^*(t)$

$H_A : F(t) \neq F^*(t)$ for at least one t

Where $F(t)$ refers to the estimated CDF of the distribution of residuals of our linear model, and $F^*(t)$ is the CDF of the normal distribution.

We'll use a significance level $\alpha = 0.05$

Assumptions:

1. Continuous: The test assumes that the theoretical distribution is continuous, which is reasonable.
2. Independence: This is reasonable, as there's no reason to believe the log deaths for one hurricane would affect another.
3. Parameters: This is also satisfied, we have mean $= 0$, and sd $= 1$ for the normal distribution, and neither of these is estimated.

Test Statistic and P-value:

```
MLR <- lm(log_deaths ~ NDAM2014 + LF.PressureMB, data = hurricane_logdeaths)

ks.test(x = resid(MLR), y = "pnorm", alternative = "two.sided")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  resid(MLR)
## D = 0.066231, p-value = 0.779
## alternative hypothesis: two-sided
```

Decision:

We have a $p-$value of $0.779 > 0.05$, so we fail to reject the null hypothesis at our significance level.

Conclusion:

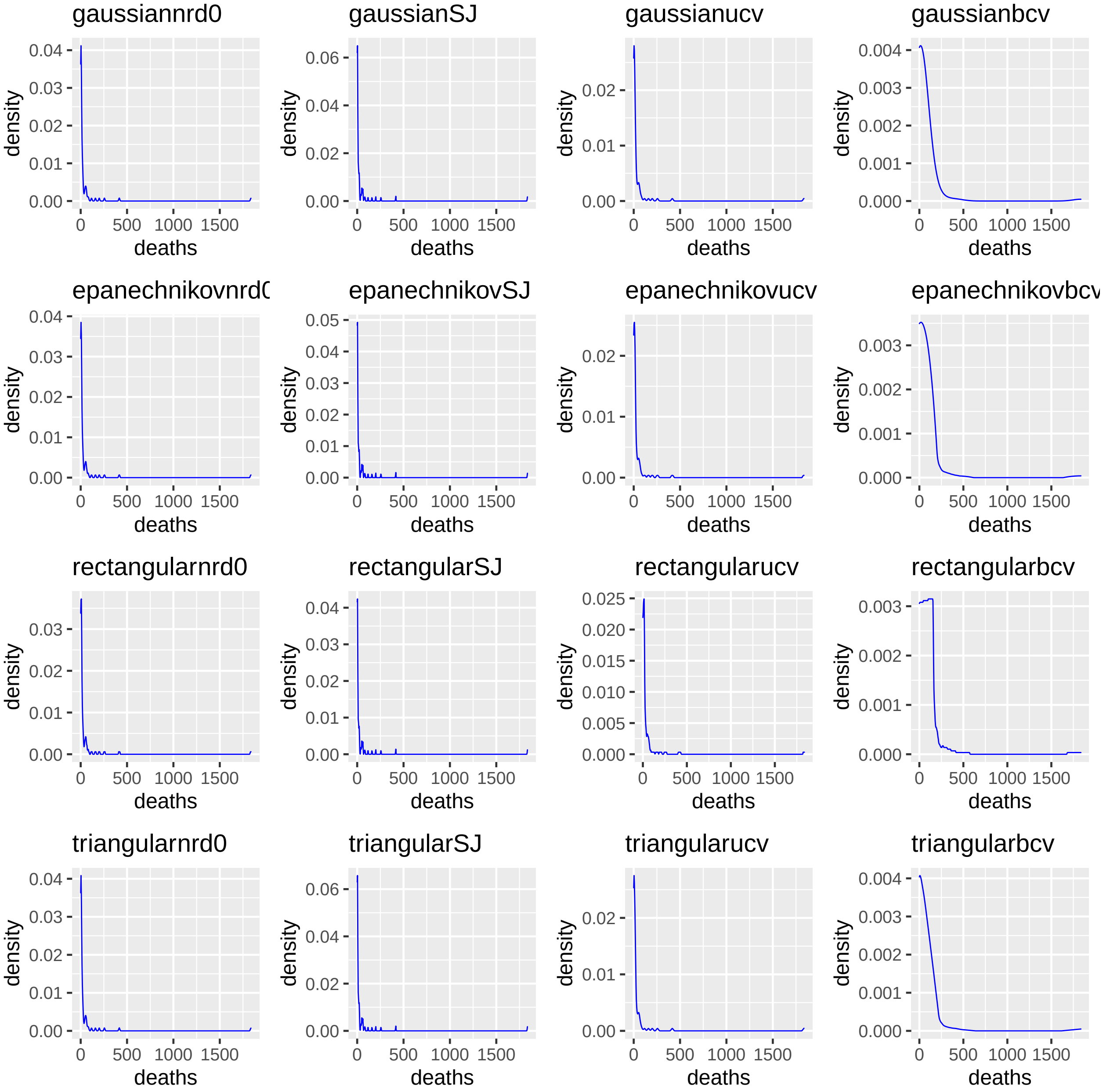
We conclude that there is insufficient evidence to suggest that the estimated CDF of the distribution of the residuals in our linear model is significantly different to the CDF of the normal distribution.

Kernel Density Estimation

kernel density estimation (not smoothing related) LEAH

Deaths using previous dataset (before log transformation)

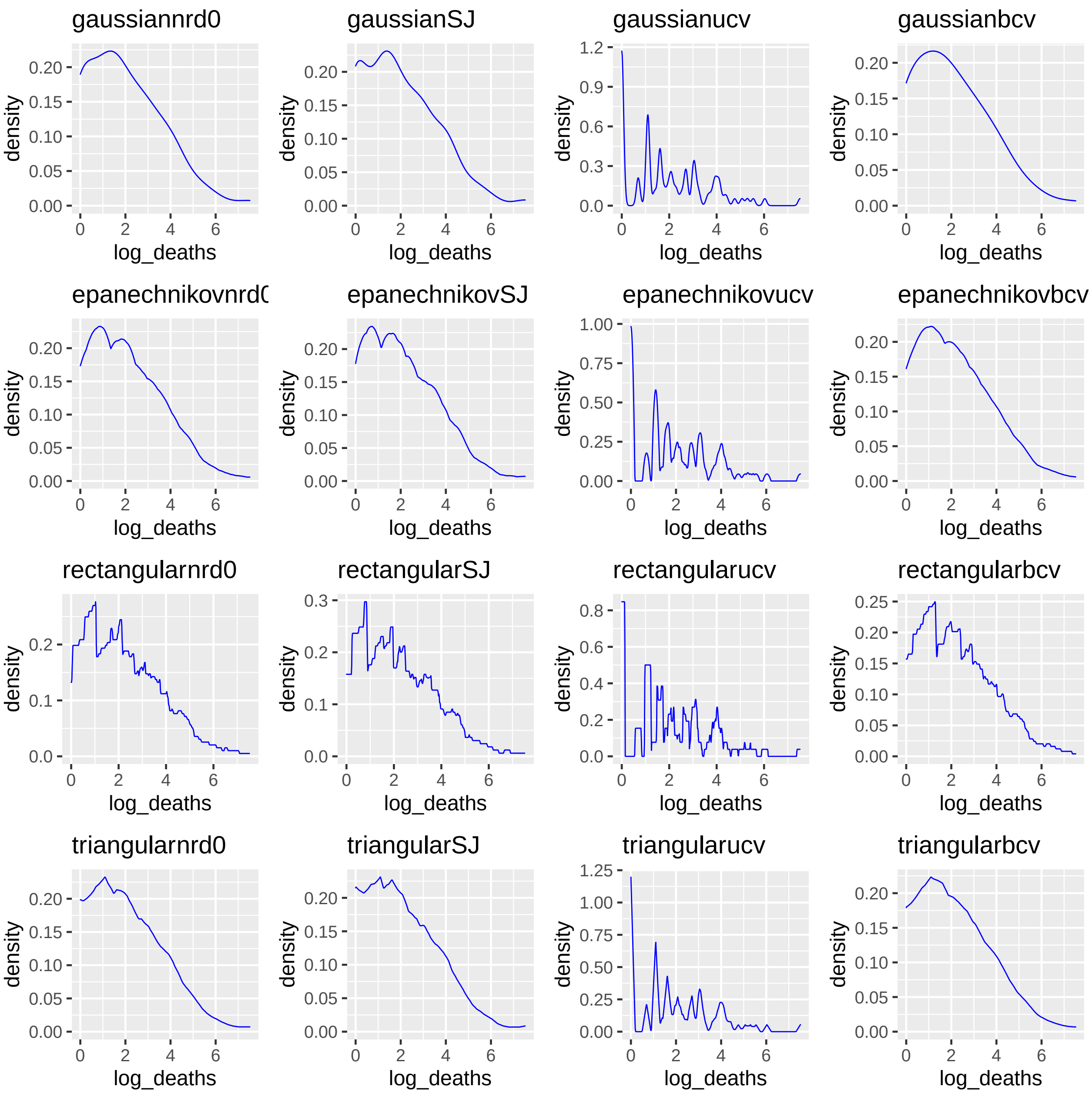
```r
i <- 1
plots <- list()
for(k in c("gaussian", "epanechnikov", "rectangular", "triangular")){
  for(bandwidth in c("nrd0", "SJ", "ucv", "bcv")){
    plots[[i]] <- ggplot(data = hurricane_numeric,
                         aes(x = hurricane_numeric$deaths)) +
                  geom_density(bw = bandwidth, kernel = k, size = 0.3,
                               color = "blue") +
                  ggtitle(paste0(k, bandwidth)) +
                  xlab("deaths")
    i <- i + 1
  }
}

cowplot::plot_grid(plotlist = plots, nrow = 4)
```



Regardless of kernel and bandwidth choice, these all look kind of terrible.

Log_Deaths:

```
i <- 1
plots <- list()
for(k in c("gaussian", "epanechnikov", "rectangular", "triangular")){
  for(bandwidth in c("nrd0", "SJ", "ucv", "bcv")){
    plots[[i]] <- ggplot(data = hurricane_logdeaths,
                  aes(x = hurricane_logdeaths$log_deaths)) +
                geom_density(bw = bandwidth, kernel = k, size = 0.3,
                          color = "blue") +
                ggtitle(paste0(k, bandwidth)) +
                    xlab("log_deaths")
    i <- i + 1
  }
}

cowplot::plot_grid(plotlist = plots, nrow = 4)
```



Rectangular kernel overfits, bandwidth selected by ucv is similarly unsuitable. Gaussian looks like it's oversmoothing. Triangular and epanechnikov are kind of similar, but I'd probably go with epanechnikov kernel with bandwidth selected using bcv. Looks pretty decent overall, and epanechnikov is generally a good choice of kernel.

OLS multiple linear regression

OLS multiple linear regression (for comparison purposes) OLIVER

```
# different models to predict log deaths
pressure_windspeed <- msummary(lm(log_deaths ~ LF.PressureMB + LF.WindsMPH,
                                  data = hurricane_logdeaths)) ;pressure_windspeed
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   62.56484   13.70080   4.567 1.55e-05 ***
## LF.PressureMB -0.06163    0.01313  -4.693 9.45e-06 ***
## LF.WindsMPH   -0.01120    0.01154  -0.971    0.334
##
## Residual standard error: 1.334 on 91 degrees of freedom
## Multiple R-squared:  0.3708, Adjusted R-squared:  0.357
## F-statistic: 26.81 on 2 and 91 DF,  p-value: 6.994e-10
```

```
pressure_ndam <- msummary(lm(log_deaths ~ LF.PressureMB + NDAM2014,
                                  data = hurricane_logdeaths)) ;pressure_ndam
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   13.311004   9.093812   1.464    0.147
## LF.PressureMB -0.014859   0.009005  -1.650    0.102
## NDAM2014       0.406769   0.074712   5.444 4.37e-07 ***
##
## Residual standard error: 1.165 on 91 degrees of freedom
## Multiple R-squared:  0.5205, Adjusted R-squared:  0.5099
## F-statistic: 49.39 on 2 and 91 DF,  p-value: 2.996e-15
```

```
pressure_basedam <- msummary(lm(log_deaths ~ LF.PressureMB + BaseDam2014,
                                  data = hurricane_logdeaths)) ;pressure_basedam
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.747e+01  7.187e+00   5.213 1.15e-06 ***
## LF.PressureMB -3.705e-02  7.424e-03  -4.991 2.87e-06 ***
## BaseDam2014    4.514e-05  1.161e-05   3.888 0.000192 ***
##
## Residual standard error: 1.242 on 91 degrees of freedom
## Multiple R-squared:  0.4548, Adjusted R-squared:  0.4428
## F-statistic: 37.96 on 2 and 91 DF,  p-value: 1.029e-12
```

```
wind_basedam <- msummary(lm(log_deaths ~ LF.WindsMPH + BaseDam2014,
                            data = hurricane_logdeaths)) ;wind_basedam
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.050e+00  6.416e-01  -1.636    0.105
## LF.WindsMPH  2.590e-02  6.105e-03   4.242 5.31e-05 ***
## BaseDam2014  6.117e-05  1.087e-05   5.628 2.00e-07 ***
##
## Residual standard error: 1.28 on 91 degrees of freedom
## Multiple R-squared:  0.4203, Adjusted R-squared:  0.4075
## F-statistic: 32.98 on 2 and 91 DF,  p-value: 1.686e-11
```

```
wind_ndam <- msummary(lm(log_deaths ~ LF.WindsMPH + NDAM2014,
                            data = hurricane_logdeaths)) ;wind_ndam
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.807280   0.583618  -3.097   0.0026 **
## LF.WindsMPH  0.002085   0.007041   0.296   0.7678    
## NDAM2014     0.484789   0.066503   7.290  1.1e-10 ***
## 
## Residual standard error: 1.181 on 91 degrees of freedom
## Multiple R-squared:  0.5066, Adjusted R-squared:  0.4958 
## F-statistic: 46.72 on 2 and 91 DF,  p-value: 1.096e-14
```

regsubsets

```
test <- regsubsets(log_deaths ~ ., data=hurricane_logdeaths, nbest=3)
summary(test)
```

```
## Subset selection object
## Call: regsubsets.formula(log_deaths ~ ., data = hurricane_logdeaths, 
##     nbest = 3)
## 5 Variables  (and intercept)
##               Forced in Forced out
## LF.WindsMPH       FALSE      FALSE
## LF.PressureMB     FALSE      FALSE
## LF.times          FALSE      FALSE
## NDAM2014          FALSE      FALSE
## BaseDam2014       FALSE      FALSE
## 3 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          LF.WindsMPH LF.PressureMB LF.times NDAM2014 BaseDam2014
## 1  ( 1 ) " "         " "           " "      "*"      " "        
## 1  ( 2 ) " "         "*"           " "      " "      " "        
## 1  ( 3 ) " "         " "           " "      " "      "*"        
## 2  ( 1 ) " "         " "           " "      "*"      "*"        
## 2  ( 2 ) " "         "*"           " "      "*"      " "        
## 2  ( 3 ) "*"         " "           " "      "*"      " "        
## 3  ( 1 ) " "         "*"           " "      "*"      "*"        
## 3  ( 2 ) "*"         " "           " "      "*"      "*"        
## 3  ( 3 ) " "         " "           "*"      "*"      "*"        
## 4  ( 1 ) "*"         "*"           " "      "*"      "*"        
## 4  ( 2 ) " "         "*"           "*"      "*"      "*"        
## 4  ( 3 ) "*"         " "           "*"      "*"      "*"        
## 5  ( 1 ) "*"         "*"           "*"      "*"      "*"
```

```
best_1pred <-  msummary(lm(log_deaths ~  NDAM2014, 
                            data = hurricane_logdeaths)) ;best_1pred
```

```
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.68026    0.39372  -4.268 4.79e-05 ***
## NDAM2014     0.49726    0.05121   9.710 9.29e-16 ***
## 
## Residual standard error: 1.175 on 92 degrees of freedom
## Multiple R-squared:  0.5061, Adjusted R-squared:  0.5008 
## F-statistic: 94.29 on 1 and 92 DF,  p-value: 9.293e-16
```

```
best_2pred <-  lm(log_deaths ~  NDAM2014 + LF.PressureMB, 
                            data = hurricane_logdeaths) ;best_2pred
```

```
## 
## Call:
## lm(formula = log_deaths ~ NDAM2014 + LF.PressureMB, data = hurricane_logdeaths)
## 
```

```
## Coefficients:
##   (Intercept)       NDAM2014  LF.PressureMB
##      13.31100        0.40677       -0.01486
```

```r
best_3pred <-  lm(log_deaths ~  NDAM2014 + LF.PressureMB + LF.WindsMPH,
                            data = hurricane_logdeaths) ;best_3pred
```

```
##
## Call:
## lm(formula = log_deaths ~ NDAM2014 + LF.PressureMB + LF.WindsMPH,
##     data = hurricane_logdeaths)
##
## Coefficients:
##   (Intercept)       NDAM2014  LF.PressureMB    LF.WindsMPH
##      26.53687        0.41011       -0.02718       -0.01310
```

```r
# so stick with 2 predictor model
anova(best_2pred, best_3pred)
```

```
## Analysis of Variance Table
##
## Model 1: log_deaths ~ NDAM2014 + LF.PressureMB
## Model 2: log_deaths ~ NDAM2014 + LF.PressureMB + LF.WindsMPH
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     91 123.40
## 2     90 121.11  1    2.2921 1.7033 0.1952
```

```r
best_4pred <-  msummary(lm(log_deaths ~  NDAM2014 + LF.PressureMB + LF.WindsMPH
                           + BaseDam2014, data = hurricane_logdeaths)) ;best_4pred
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.049e+01  1.412e+01   0.743  0.45943
## NDAM2014       3.610e-01  7.333e-02   4.922  3.9e-06 ***
## LF.PressureMB -1.144e-02  1.357e-02  -0.843  0.40144
## LF.WindsMPH   -2.870e-03  1.023e-02  -0.281  0.77966
## BaseDam2014    3.348e-05  1.129e-05   2.966  0.00387 **
##
## Residual standard error: 1.113 on 89 degrees of freedom
## Multiple R-squared:  0.5717, Adjusted R-squared:  0.5525
## F-statistic:  29.7 on 4 and 89 DF,  p-value: 1.082e-15
```

```r
full_model <- msummary(lm(log_deaths ~  NDAM2014 + LF.PressureMB + LF.WindsMPH
                           + BaseDam2014 + LF.times,
                          data = hurricane_logdeaths)) ;full_model
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.077e+01  1.425e+01   0.756  0.45177
## NDAM2014       3.576e-01  7.536e-02   4.746 7.99e-06 ***
## LF.PressureMB -1.176e-02  1.372e-02  -0.857  0.39381
## LF.WindsMPH   -3.222e-03  1.041e-02  -0.309  0.75768
## BaseDam2014    3.346e-05  1.135e-05   2.948  0.00409 **
## LF.times       7.558e-02  3.494e-01   0.216  0.82925
##
## Residual standard error: 1.119 on 88 degrees of freedom
## Multiple R-squared:  0.572,  Adjusted R-squared:  0.5476
## F-statistic: 23.52 on 5 and 88 DF,  p-value: 6.199e-15
```

JHM Multiple Regression

```
best_2pred_rank <-  rfit(log_deaths ~  NDAM2014 + LF.PressureMB,
                             data = hurricane_logdeaths) ;msummary(best_2pred_rank)
```

```
## Call:
## rfit.default(formula = log_deaths ~ NDAM2014 + LF.PressureMB,
##     data = hurricane_logdeaths)
##
## Coefficients:
##                 Estimate Std. Error t.value   p.value
## (Intercept)   13.6101739  8.5623612  1.5895   0.11541
## NDAM2014       0.3824041  0.0703365  5.4368 4.518e-07 ***
## LF.PressureMB -0.0149520  0.0084777 -1.7637   0.08114 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared (Robust): 0.4758527
## Reduction in Dispersion Test: 41.30765 p-value: 0
```

```
other_2pred_rank <-  rfit(log_deaths ~  BaseDam2014 + LF.PressureMB,
                             data = hurricane_logdeaths) ;msummary(other_2pred_rank)
```

```
## Call:
## rfit.default(formula = log_deaths ~ BaseDam2014 + LF.PressureMB,
##     data = hurricane_logdeaths)
##
## Coefficients:
##                   Estimate  Std. Error t.value   p.value
## (Intercept)    4.0191e+01  7.2359e+00  5.5545 2.739e-07 ***
## BaseDam2014    4.2799e-05  1.1685e-05  3.6626 0.0004189 ***
## LF.PressureMB -3.9935e-02  7.4715e-03 -5.3449 6.655e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared (Robust): 0.4121481
## Reduction in Dispersion Test: 31.90045 p-value: 0
```

```
best_3pred_rank <-  rfit(log_deaths ~  NDAM2014 + LF.PressureMB + LF.WindsMPH,
                             data = hurricane_logdeaths) ;msummary(best_3pred_rank)
```

```
## Call:
## rfit.default(formula = log_deaths ~ NDAM2014 + LF.PressureMB +
##     LF.WindsMPH, data = hurricane_logdeaths)
##
## Coefficients:
##                Estimate Std. Error t.value   p.value
## (Intercept)   30.908336  13.711624  2.2542   0.02661 *
## NDAM2014       0.398450   0.075119  5.3043 8.036e-07 ***
## LF.PressureMB -0.031112   0.013135 -2.3687   0.01999 *
## LF.WindsMPH   -0.017380   0.010125 -1.7166   0.08950 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared (Robust): 0.4704651
## Reduction in Dispersion Test: 26.65349 p-value: 0
```

```
# so again, we drop wind speed as a predictor
drop.test(best_3pred_rank, best_2pred_rank)
```

```
## 
## Drop in Dispersion Test
## F-Statistic     p-value 
##     2.55900     0.11317
```

```
# different models to predict log deaths using rfit
pressure_windspeed <- msummary(rfit(log_deaths ~ LF.PressureMB + LF.WindsMPH,
                                  data = hurricane_logdeaths)) ;pressure_windspeed
```

```
## Call:
## rfit.default(formula = log_deaths ~ LF.PressureMB + LF.WindsMPH, 
##     data = hurricane_logdeaths)
## 
## Coefficients:
##                 Estimate Std. Error t.value   p.value    
## (Intercept)   65.925312  13.675694  4.8206 5.705e-06 ***
## LF.PressureMB -0.064997   0.013107 -4.9590 3.271e-06 ***
## LF.WindsMPH   -0.012921   0.011514 -1.1223    0.2647    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Multiple R-squared (Robust): 0.3491129 
## Reduction in Dispersion Test: 24.4046 p-value: 0
```

```
pressure_ndam <- msummary(rfit(log_deaths ~ LF.PressureMB + NDAM2014,
                                  data = hurricane_logdeaths)) ;pressure_ndam
```

```
## Call:
## rfit.default(formula = log_deaths ~ LF.PressureMB + NDAM2014, 
##     data = hurricane_logdeaths)
## 
## Coefficients:
##                  Estimate Std. Error t.value   p.value    
## (Intercept)   13.6101739  8.5623612  1.5895   0.11541    
## LF.PressureMB -0.0149520  0.0084777 -1.7637   0.08114 .  
## NDAM2014       0.3824041  0.0703365  5.4368 4.518e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Multiple R-squared (Robust): 0.4758527 
## Reduction in Dispersion Test: 41.30765 p-value: 0
```

```
pressure_basedam <- msummary(rfit(log_deaths ~ LF.PressureMB + BaseDam2014,
                                  data = hurricane_logdeaths)) ;pressure_basedam
```

```
## Call:
## rfit.default(formula = log_deaths ~ LF.PressureMB + BaseDam2014, 
##     data = hurricane_logdeaths)
## 
## Coefficients:
##                   Estimate  Std. Error t.value   p.value    
## (Intercept)    4.0191e+01  7.2359e+00  5.5545 2.739e-07 ***
## LF.PressureMB -3.9935e-02  7.4715e-03 -5.3449 6.655e-07 ***
## BaseDam2014    4.2799e-05  1.1685e-05  3.6626 0.0004189 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared (Robust): 0.4121481
## Reduction in Dispersion Test: 31.90045 p-value: 0
```

```
wind_basedam <- msummary(rfit(log_deaths ~ LF.WindsMPH + BaseDam2014,
                             data = hurricane_logdeaths)) ;wind_basedam
```

```
## Call:
## rfit.default(formula = log_deaths ~ LF.WindsMPH + BaseDam2014,
##     data = hurricane_logdeaths)
##
## Coefficients:
##                Estimate  Std. Error t.value   p.value
## (Intercept) -1.2130e+00  6.1540e-01 -1.9710   0.05176 .
## LF.WindsMPH  2.7111e-02  5.6093e-03  4.8332 5.426e-06 ***
## BaseDam2014  6.2308e-05  9.9868e-06  6.2390 1.368e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared (Robust): 0.3933758
## Reduction in Dispersion Test: 29.50525 p-value: 0
```

```
wind_ndam <- msummary(rfit(log_deaths ~ LF.WindsMPH + NDAM2014,
                             data = hurricane_logdeaths)) ;wind_ndam
```

```
## Call:
## rfit.default(formula = log_deaths ~ LF.WindsMPH + NDAM2014, data = hurricane_logdeaths)
##
## Coefficients:
##                Estimate  Std. Error t.value   p.value
## (Intercept) -1.56736126  0.63209942 -2.4796   0.01499 *
## LF.WindsMPH  0.00013567  0.00750878  0.0181   0.98562
## NDAM2014     0.47809833  0.07091877  6.7415 1.401e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared (Robust): 0.4331468
## Reduction in Dispersion Test: 34.76769 p-value: 0
```

JHM multiple regression OLIVER

Generalized Additive Model

generalized additive model (which will require you attempting a variety of smoothers) ADI

We need to go about fitting this manually first, and then we can use step.gam to check

Manual Forward Selection

Looking at our `ggpairs` call from earlier, we see that the variables can be sorted in decreasing magnitude of correlation with `log_deaths` in this way: `NDAM2014`, `LF.PressureMB`, `BaseDam2014`, `LF.WindsMPH`, `LF.times`.

We'll begin by including NDAM2014 in a single predictor model, and then going through the list.

```
m0 <- gam(log_deaths ~ NDAM2014, data = hurricane_logdeaths) # NDAM2014 linear

# bsplines with varying df
#best
m1 <- gam(log_deaths ~ bs(NDAM2014), data = hurricane_logdeaths)
m2 <- gam(log_deaths ~ bs(NDAM2014, df = 2), data = hurricane_logdeaths)
m3 <- gam(log_deaths ~ bs(NDAM2014, df = 3), data = hurricane_logdeaths)
m4 <- gam(log_deaths ~ bs(NDAM2014, df = 4), data = hurricane_logdeaths)
m5 <- gam(log_deaths ~ bs(NDAM2014, df = 5), data = hurricane_logdeaths)
m6 <- gam(log_deaths ~ bs(NDAM2014, df = 6), data = hurricane_logdeaths)

# smoothing splines with varying df arguments (smoothing parameter)
m7 <- gam(log_deaths ~ s(NDAM2014), data = hurricane_logdeaths)
m8 <- gam(log_deaths ~ s(NDAM2014, df = 4), data = hurricane_logdeaths)
m9 <- gam(log_deaths ~ s(NDAM2014, df = 5), data = hurricane_logdeaths)
m10 <- gam(log_deaths ~ s(NDAM2014, df = 6), data = hurricane_logdeaths)
m11 <- gam(log_deaths ~ s(NDAM2014, df = 7), data = hurricane_logdeaths)
m12 <- gam(log_deaths ~ s(NDAM2014, df = 8), data = hurricane_logdeaths)

AIC(m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12)
```

```
##     df      AIC
## m0   3 301.1141
## m1   5 290.8135
## m2   5 290.8135
## m3   5 290.8135
## m4   6 292.6707
## m5   7 293.4938
## m6   8 293.2808
## m7   3 292.1727
## m8   3 292.1727
## m9   3 293.0366
## m10  3 293.7629
## m11  3 294.5505
## m12  3 295.4660
```

```
# lowest AIC is m11 which is the basis for our next step
```

```
# LF.PressureMB absent
m0 <- gam(log_deaths ~ bs(NDAM2014), data = hurricane_logdeaths)

m1 <- gam(log_deaths ~ bs(NDAM2014) + LF.PressureMB,
          data = hurricane_logdeaths) # LF.PressureMB linearly included

# bsplines with varying df
m2 <- gam(log_deaths ~ bs(NDAM2014) + bs(LF.PressureMB),
          data = hurricane_logdeaths)
m3 <- gam(log_deaths ~ bs(NDAM2014) + bs(LF.PressureMB, df = 3),
          data = hurricane_logdeaths)
m4 <- gam(log_deaths ~ bs(NDAM2014) + bs(LF.PressureMB, df = 4),
          data = hurricane_logdeaths)
m5 <- gam(log_deaths ~ bs(NDAM2014) + bs(LF.PressureMB, df = 5),
          data = hurricane_logdeaths)
```

```
m6 <- gam(log_deaths ~ bs(NDAM2014) + bs(LF.PressureMB, df = 6),
          data = hurricane_logdeaths)
m7 <- gam(log_deaths ~ bs(NDAM2014) + bs(LF.PressureMB, df = 7),
          data = hurricane_logdeaths)

# smoothing splines with varying df arguments (smoothing parameter)
m8 <- gam(log_deaths ~ bs(NDAM2014) + s(LF.PressureMB),
          data = hurricane_logdeaths)
m9 <- gam(log_deaths ~ bs(NDAM2014) + s(LF.PressureMB, df = 4),
          data = hurricane_logdeaths)
#best
m10 <- gam(log_deaths ~ bs(NDAM2014) + s(LF.PressureMB, df = 5),
           data = hurricane_logdeaths)
m11 <- gam(log_deaths ~ bs(NDAM2014) + s(LF.PressureMB, df = 6),
           data = hurricane_logdeaths)
m12 <- gam(log_deaths ~ bs(NDAM2014) + s(LF.PressureMB, df = 7),
           data = hurricane_logdeaths)
m13 <- gam(log_deaths ~ bs(NDAM2014) + s(LF.PressureMB, df = 3),
           data = hurricane_logdeaths)

AIC(m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13)
```

```
##     df      AIC
## m0   5 290.8135
## m1   6 292.5280
## m2   8 290.8837
## m3   8 290.8837
## m4   9 290.5165
## m5  10 291.5873
## m6  11 290.6747
## m7  12 292.1686
## m8   6 288.5303
## m9   6 288.5303
## m10  6 288.0150
## m11  6 288.0912
## m12  6 288.6405
## m13  6 289.2468
```

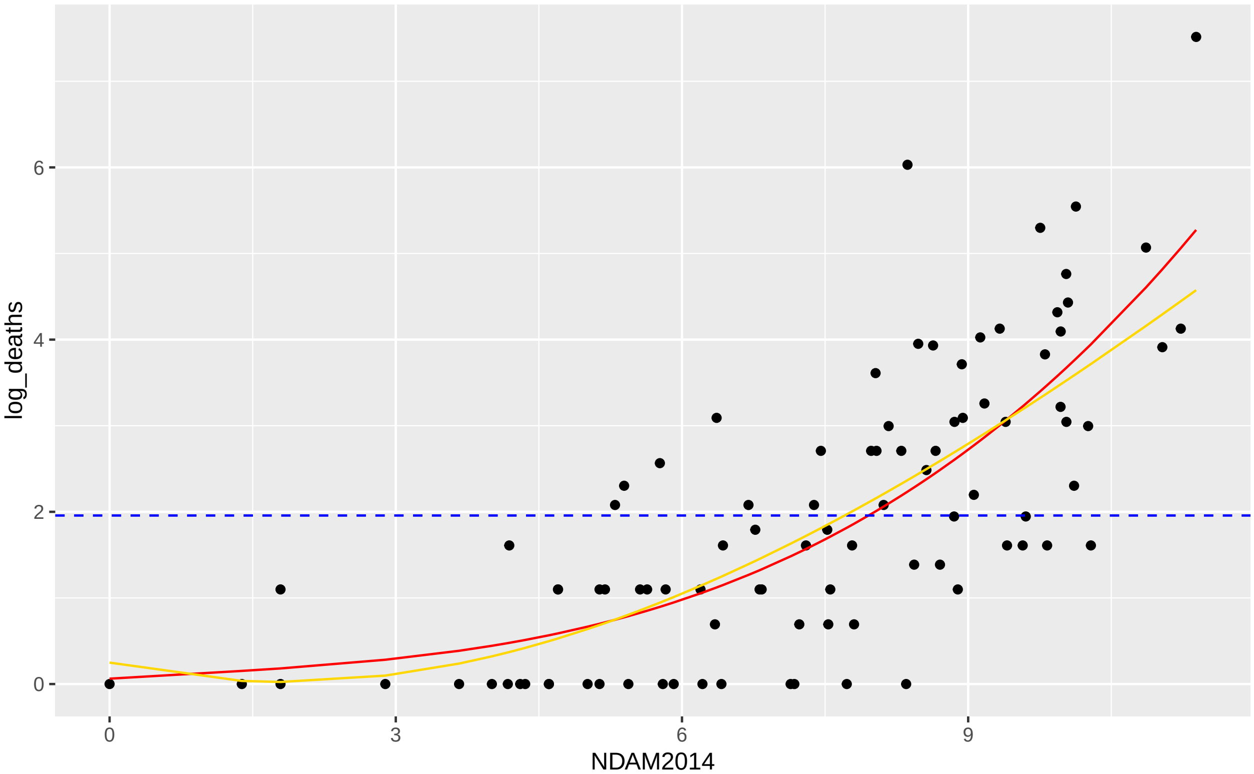
```
# min AIC is m5
```

FINAL MODEL

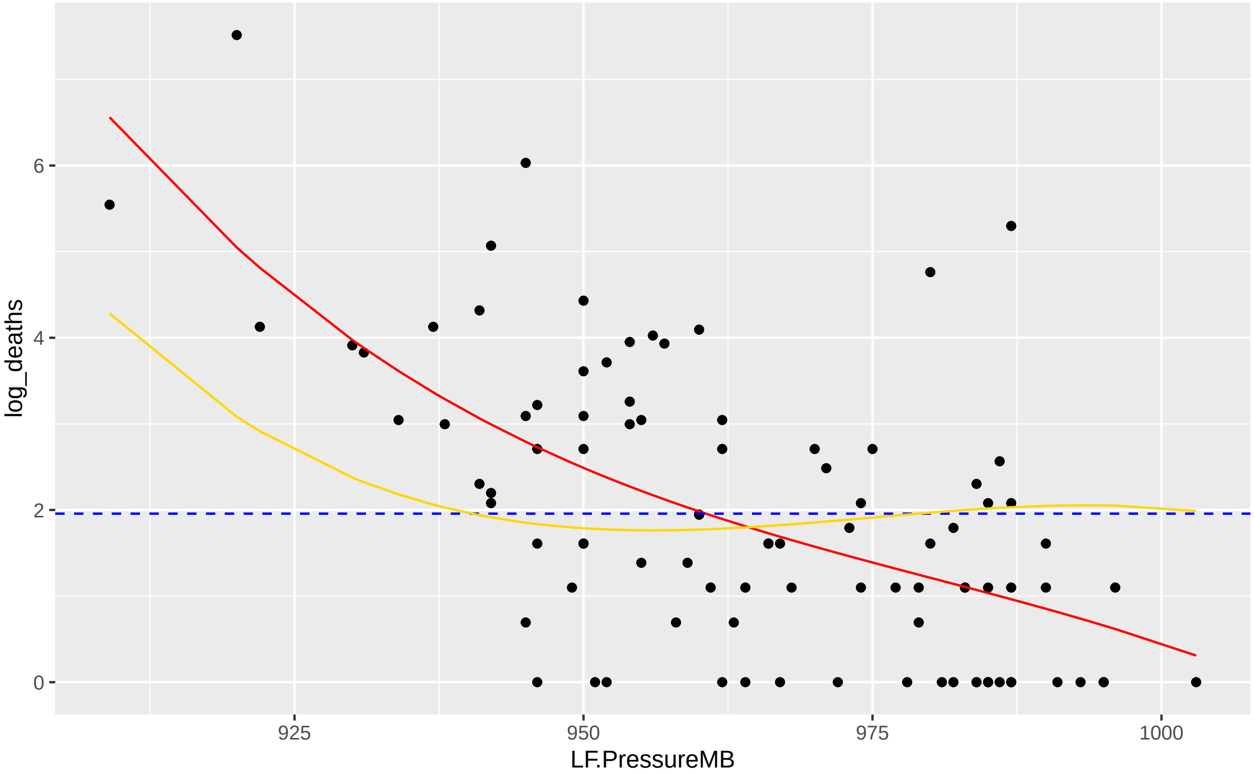
```
# all other predictors increased AIC when added
# result fits with OLS MLR, JHM
poly_gam <- gam(log_deaths ~ bs(NDAM2014) + poly(LF.PressureMB, 3),
          data = hurricane_logdeaths)
AIC(poly_gam)
```

```
## [1] 290.8837
```

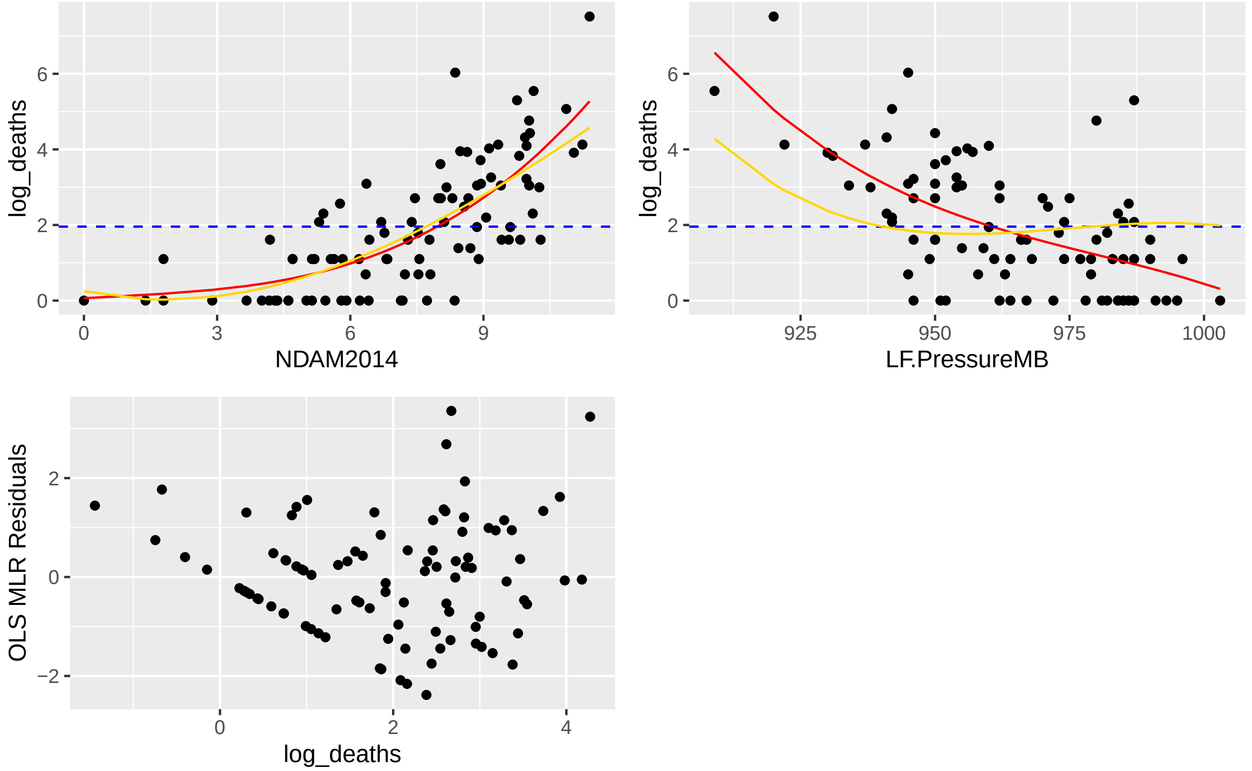
```
#final_gam <- gam(log_deaths ~ s(NDAM2014, df = 7) + bs(LF.PressureMB, df = 6),
#           data = hurricane_logdeaths)
#final_gam <- gam(log_deaths ~ bs(NDAM2014) + s(LF.PressureMB, df = 5),
#           data = hurricane_logdeaths)

final_gam <- gam(log_deaths ~ bs(NDAM2014) + poly(LF.PressureMB, 3),
          data = hurricane_logdeaths)
#summary(final_gam)
final_gam_df <- data.frame(predict(final_gam, type="terms"))
colnames(final_gam_df) <- c("NDAM_pred", "pressure_pred")

ybar <- mean(hurricane_logdeaths$log_deaths)
mean_adj_smooth <- predict(final_gam, type="terms") + ybar
colnames(mean_adj_smooth) <- c("NDAM_mean_adj", "pressure_mean_adj")
mean_adj_smooth <- cbind.data.frame(mean_adj_smooth, hurricane_logdeaths)

NDAM_gam <- gam(log_deaths ~ bs(NDAM2014), data = hurricane_logdeaths)

NDAM_df <- data.frame(pred_NDAM = predict(NDAM_gam, hurricane_logdeaths),
                      NDAM2014 = hurricane_logdeaths$NDAM2014)

#pressure_gam <- gam(log_deaths ~ s(LF.PressureMB, df=5), data = hurricane_logdeaths)
pressure_gam <- gam(log_deaths ~ poly(LF.PressureMB, 3), data = hurricane_logdeaths)
pressure_df <- data.frame(pred_pressure = predict(pressure_gam, hurricane_logdeaths),
                      pressure = hurricane_logdeaths$LF.PressureMB)

NDAM_plot <- ggplot(data = hurricane_logdeaths, aes(x=NDAM2014, y=log_deaths)) +
  geom_point() +
  geom_line(inherit.aes = FALSE, data=NDAM_df,
            aes(x=NDAM2014, y=pred_NDAM), color = "red") +
  geom_hline(yintercept = ybar, linetype = 2, color = "blue") +
  geom_line(inherit.aes = FALSE, data=mean_adj_smooth,
            aes(x=NDAM2014, y=NDAM_mean_adj), color="gold")

pressure_plot <- ggplot(data = hurricane_logdeaths, aes(x=LF.PressureMB, y=log_deaths)) +
  geom_point() +
  geom_line(inherit.aes = FALSE, data=pressure_df,
            aes(x=pressure, y=pred_pressure), color = "red") +
  geom_hline(yintercept = ybar, linetype = 2, color = "blue") +
  geom_line(inherit.aes = FALSE, data=mean_adj_smooth,
            aes(x=LF.PressureMB, y=pressure_mean_adj), color="gold")

NDAM_plot
```



```
pressure_plot
```



```
hurricane_ols <- lm(log_deaths ~ NDAM2014 + LF.PressureMB, data=hurricane_logdeaths)
residual_plot <- ggplot(hurricane_ols) +
  geom_point(aes(x=.fitted, y=.resid)) +
  labs(x = "log_deaths", y = "OLS MLR Residuals")

gridExtra::grid.arrange(NDAM_plot, pressure_plot, residual_plot, ncol=2)
```



Results

table of aggregated results and discussion of findings

Limitations

only data from US, small number of observations, small number of predictors for deaths

Potential shortcomings of the dataset include limited data on weather-related variables for each hurricane, as it only contains maximum sustained windspeed, atmospheric pressure at landfall, and number of landfalls. Additionally, the "NDAM2014" column contains data on hurricane damage had the hurricane appeared in 2014 and it is unclear how these estimates were calculated (also no units are given on these observations).

Conclusion