Assignment10; STAT 689

Philip Anderson; panders2@tamu.edu

3/26/2018

Question 10

10A

```
library("mgcv")
library("HRW")
data(ragweed)
str(ragweed)
```

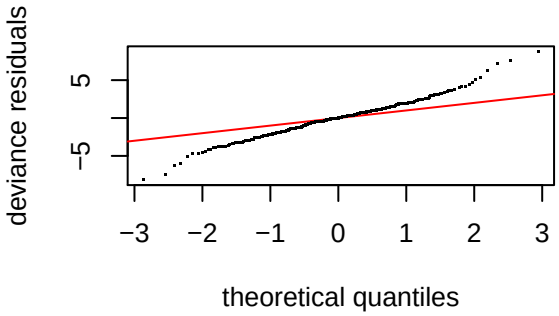
```
## 'data.frame':    334 obs. of  7 variables:
##  $ pollenCount        : int  7 7 2 5 4 0 0 0 13 62 ...
##  $ year               : int  1991 1991 1991 1991 1991 1991 1991 1991 1991 1991 ...
##  $ dayInSeason        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ temperature        : num  72.4 67.1 68.5 73.4 80.5 ...
##  $ temperatureResidual: num  0.649 -4.796 -3.521 1.207 8.14 ...
##  $ rain               : int  1 0 1 1 1 1 1 1 1 ...
##  $ windSpeed          : num  10.4 9.4 8.2 11.8 7.2 7.8 5.4 10.2 5.3 8.8 ...
```

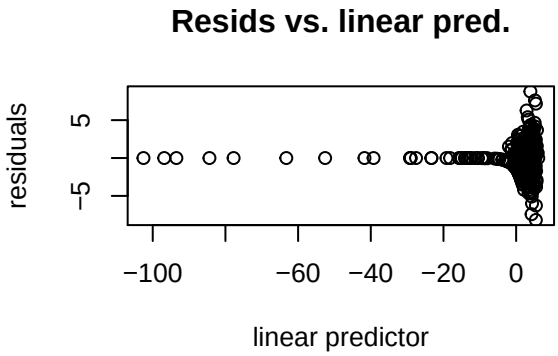
10B

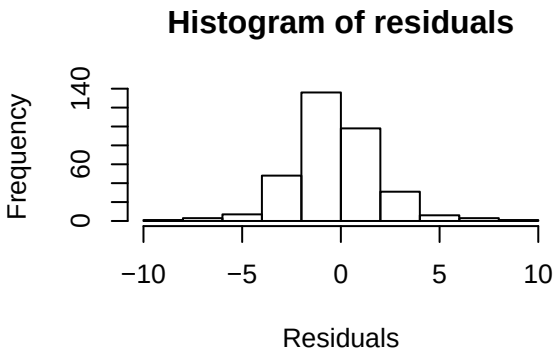
```
fit1 <- mgcv::gam(pollenCount ~ factor(year) + s(dayInSeason, k=27, by=factor(year))
                  + temperatureResidual + rain + s(windSpeed, k=27)
                 , data=ragweed
                 , family=poisson
                   )
```

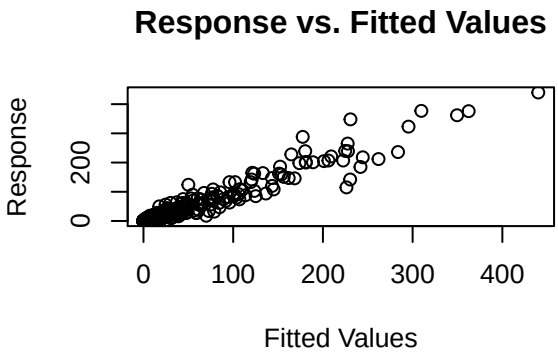
Obtain residual plots for the models.

```
mgcv::gam.check(fit1)
```



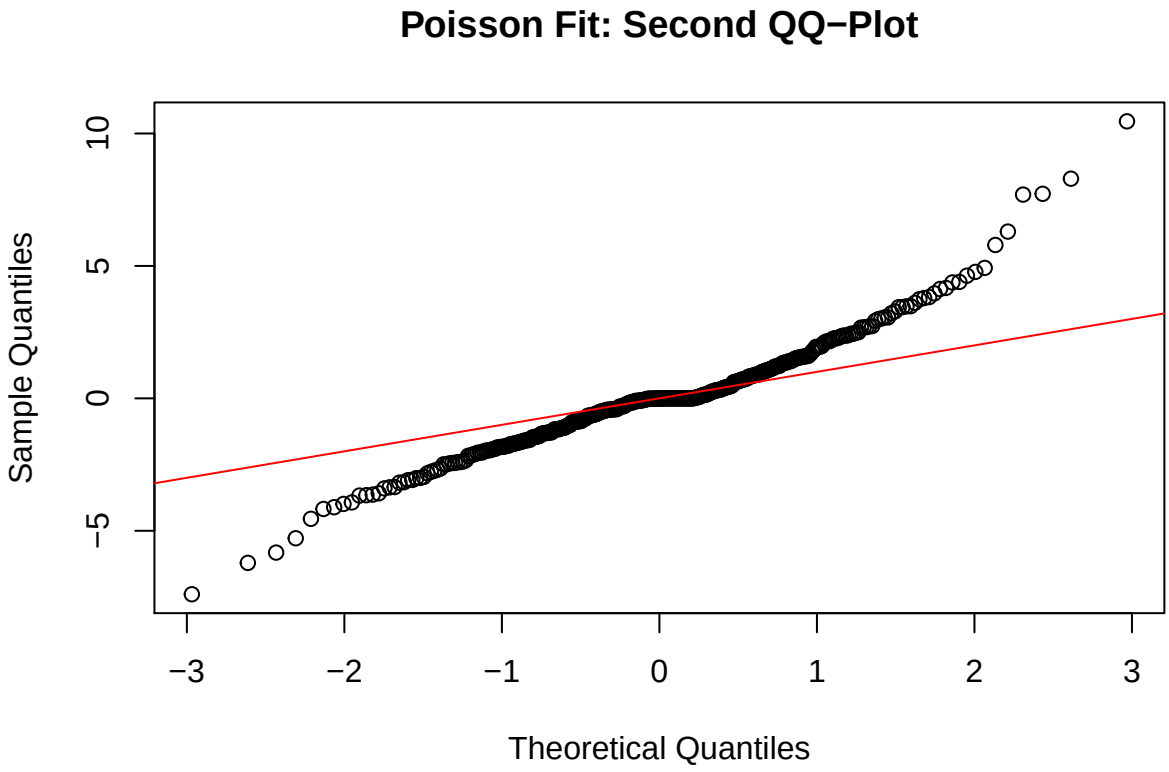







```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 16 iterations.
## Gradient range [-1.343788e-05,9.041719e-07]
## (score 4.452403 & scale 1).
## Hessian positive definite, eigenvalue range [0.0004695876,0.1070234].
## Model rank =  136 / 136
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                  k'  edf k-index p-value
## s(dayInSeason):factor(year)1991 26.0 25.9    0.92   0.075 .
## s(dayInSeason):factor(year)1992 26.0 22.2    0.92   0.055 .
## s(dayInSeason):factor(year)1993 26.0 24.4    0.92   0.070 .
## s(dayInSeason):factor(year)1994 26.0 22.9    0.92   0.070 .
## s(windSpeed)                    26.0 25.4    1.11   0.980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
qqnorm(residuals(fit1, type="scaled.pearson"), main="Poisson Fit: Second QQ-Plot")
abline(0, 1, col="red")
```



10C

Obtain alternative fits.

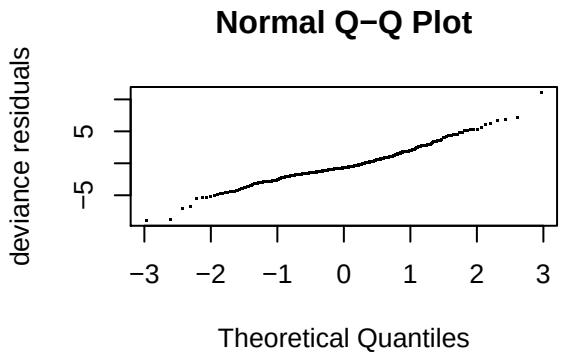
```
# quasipoisson
fit2 <- mgcv::gam(pollenCount ~ factor(year) + s(dayInSeason, k=27, by=factor(year))
                  + temperatureResidual + rain + s(windSpeed, k=27)
                  , data=ragweed
                  , scale=-1
                  , family=quasipoisson
                    )
```

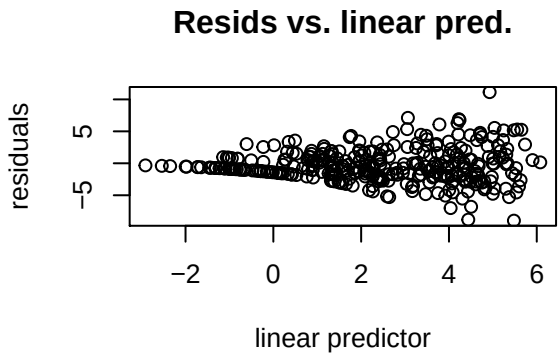
```
# square root variance transformation
fit3 <- mgcv::gam(sqrt(pollenCount) ~ factor(year) + s(dayInSeason, k=27, by=factor(year))
                  + temperatureResidual + rain + s(windSpeed, k=27)
                  , data=ragweed
                  , family=gaussian(link="identity")
                    )
```

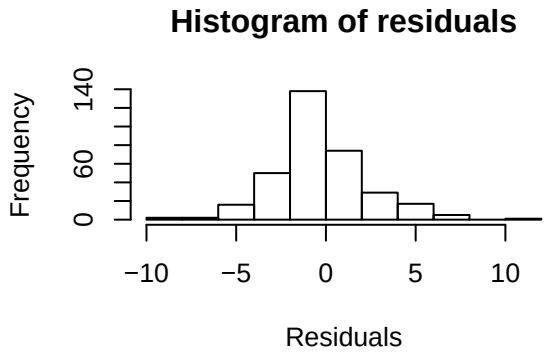
10D

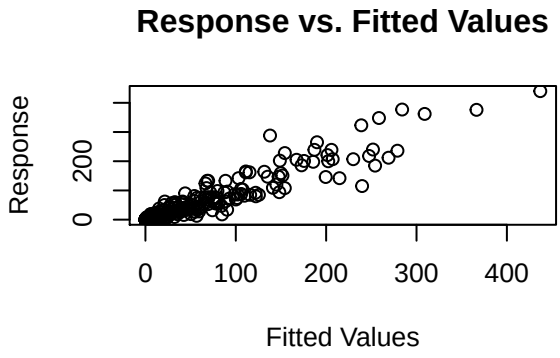
Obtain residual plots for the new fits, analogous to those obtained for fit1 in Part B

```
# quasipoisson
mgcv::gam.check(fit2)
```



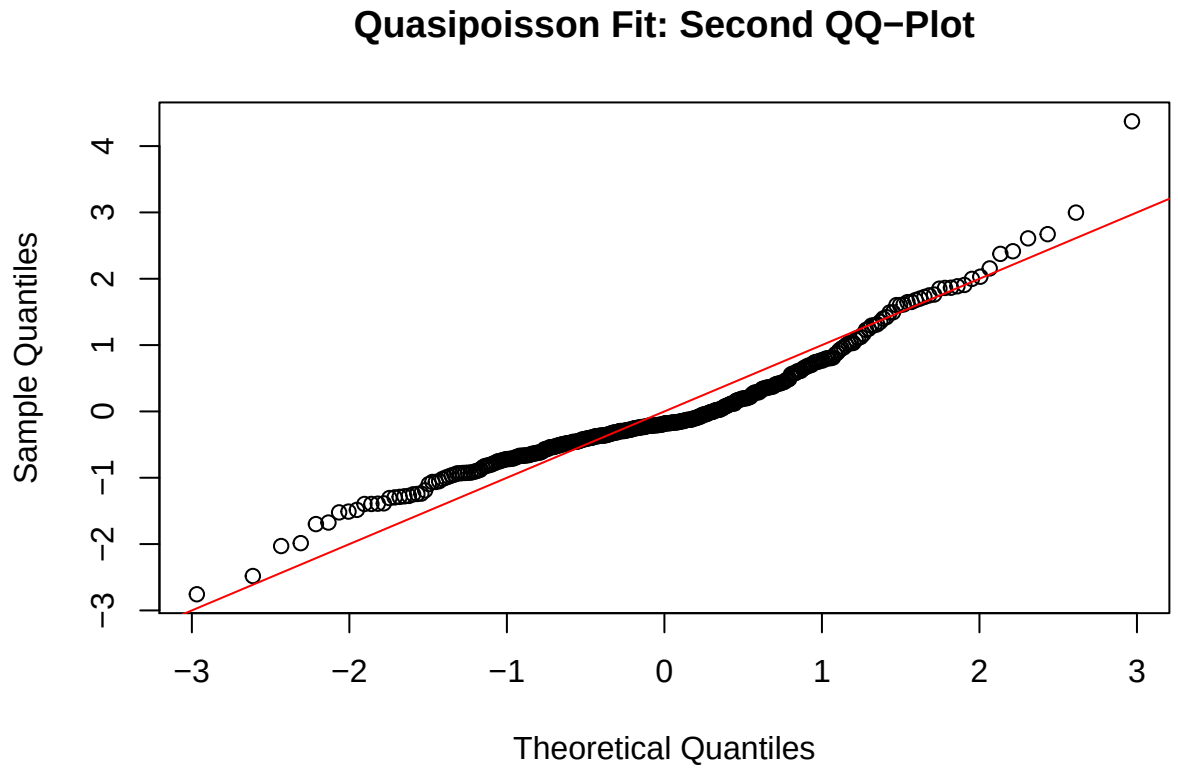




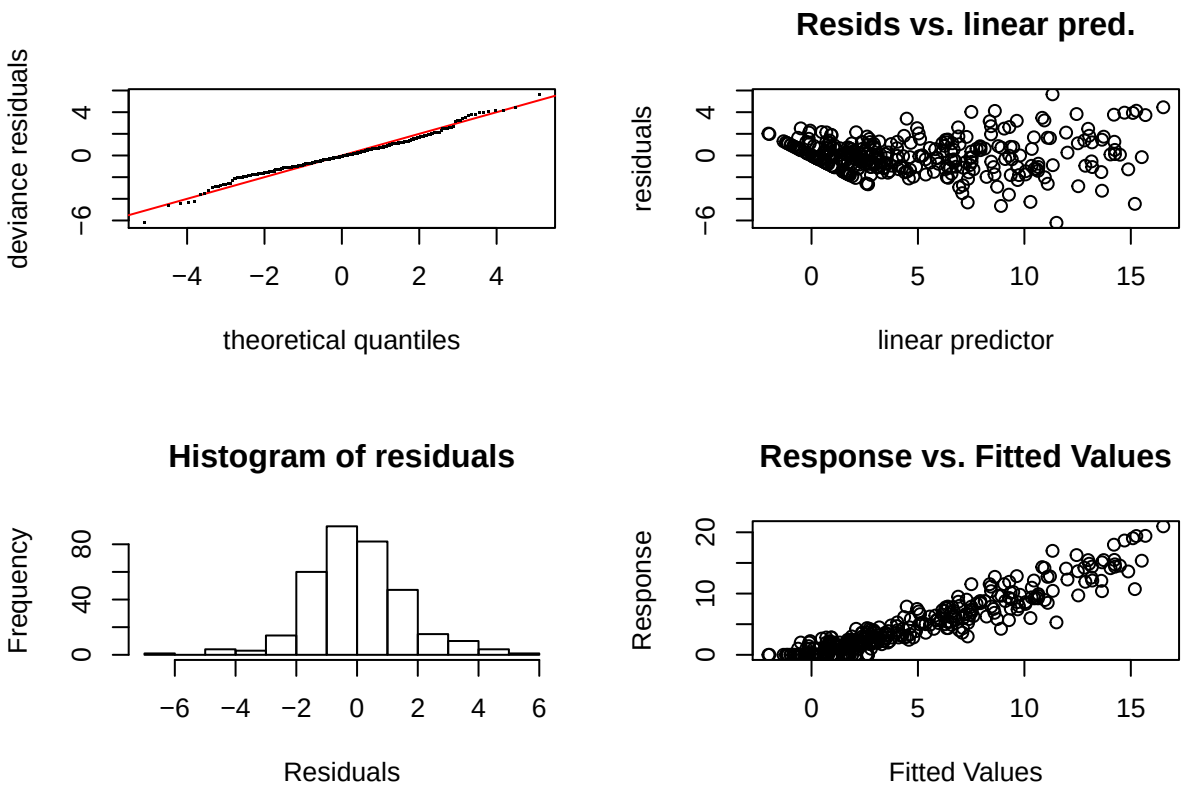



```
##
## Method: GCV   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [-6.282682e-07,3.073244e-05]
## (score 9.597656 & scale 8.54425).
## Hessian positive definite, eigenvalue range [0.01099335,0.05950797].
## Model rank =  136 / 136
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                    k'   edf k-index p-value
## s(dayInSeason):factor(year)1991 26.00  7.90    0.96    0.29
## s(dayInSeason):factor(year)1992 26.00 11.73    0.96    0.27
## s(dayInSeason):factor(year)1993 26.00  7.67    0.96    0.33
## s(dayInSeason):factor(year)1994 26.00  5.88    0.96    0.30
## s(windSpeed)                    26.00 14.65    1.05    0.88
```

```
qqnorm(residuals(fit2, type="scaled.pearson")
       , main="Quasipoisson Fit: Second QQ-Plot")
abline(0, 1, col="red")
```



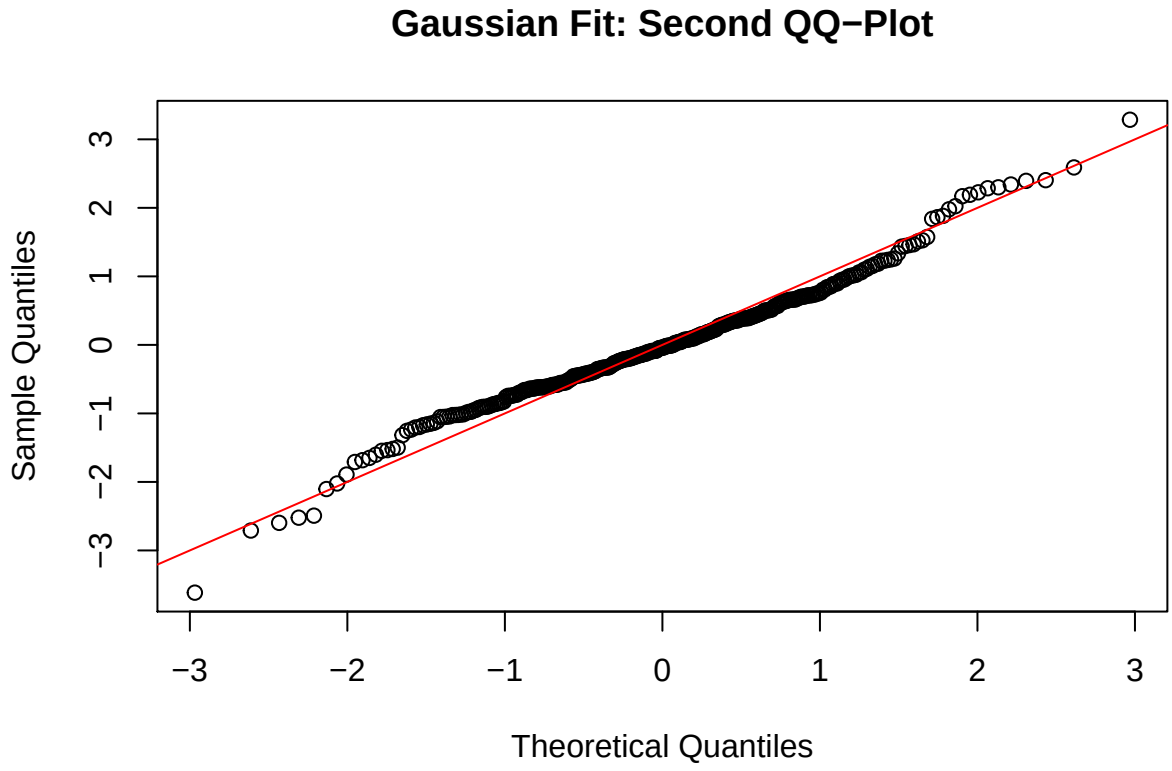

```
mgcv::gam.check(fit3)
```




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 7 iterations.
```

```
## The RMS GCV score gradient at convergence was 1.73366e-07 .
## The Hessian was positive definite.
## Model rank =  136 / 136
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                   k'   edf k-index p-value
## s(dayInSeason):factor(year)1991 26.00 15.23    0.97    0.26
## s(dayInSeason):factor(year)1992 26.00  9.31    0.97    0.26
## s(dayInSeason):factor(year)1993 26.00 10.17    0.97    0.27
## s(dayInSeason):factor(year)1994 26.00  5.69    0.97    0.27
## s(windSpeed)                    26.00  7.80    1.00    0.45
```

```
qqnorm(residuals(fit3, type="scaled.pearson")
       , main="Gaussian Fit: Second QQ-Plot"
       )
abline(0, 1, col="red")
```



10E

Based on the above summaries, I believe that the Gaussian fit for the square root transformation (fit3) is superior to the Poisson and QuasiPoisson fits. The scaled Pearson residual QQ-Norm plot shows that the Gaussian response distribution seems to result in a superior fit to the data.

```
# graphical representations are available in the previous question
# numerical summary immediately below.
mgcv::summary.gam(fit3)
```

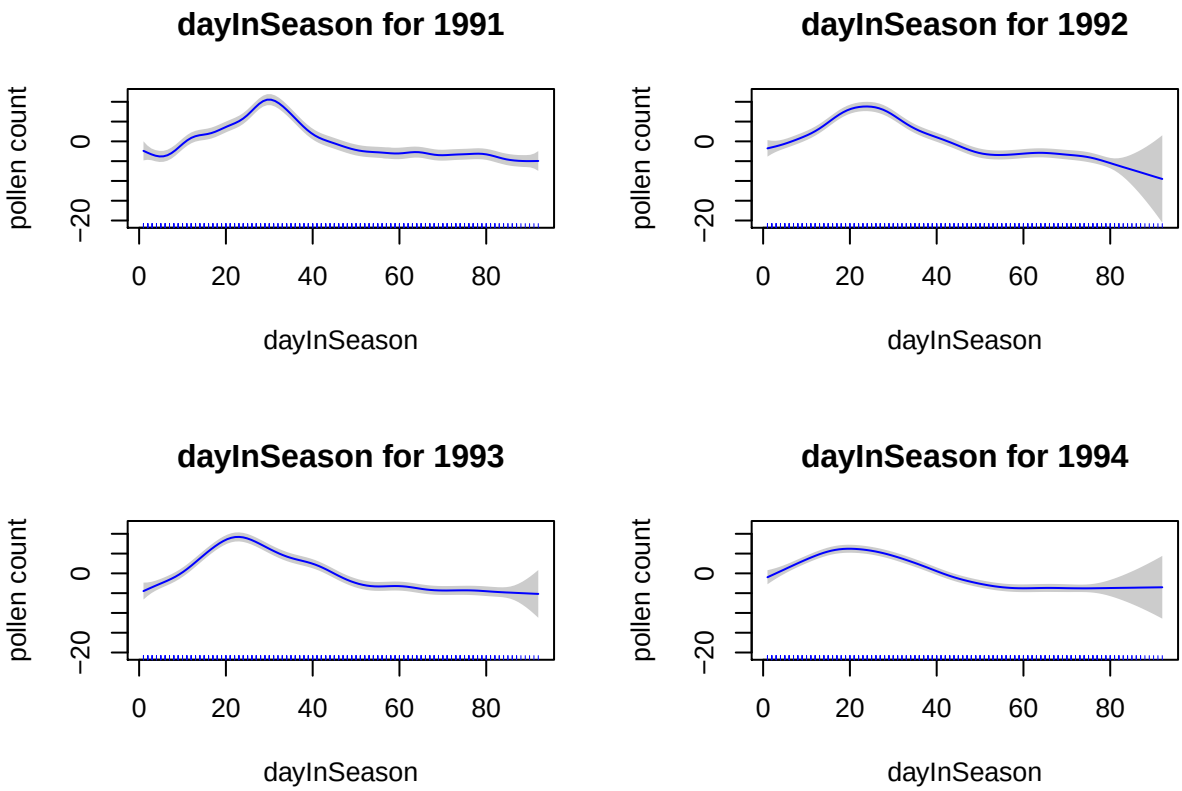
```
##
```

```
## Family: gaussian
## Link function: identity
##
## Formula:
## sqrt(pollenCount) ~ factor(year) + s(dayInSeason, k = 27, by = factor(year)) +
##     temperatureResidual + rain + s(windSpeed, k = 27)
##
## Parametric coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.60601    0.35948  10.031  < 2e-16 ***
## factor(year)1992      0.09090    0.29382   0.309 0.757277
## factor(year)1993     -0.20278    0.26391  -0.768 0.442920
## factor(year)1994     -1.13137    0.31720  -3.567 0.000425 ***
## temperatureResidual   0.10174    0.01902   5.348 1.85e-07 ***
## rain                  1.53507    0.34149   4.495 1.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                      edf Ref.df      F p-value
## s(dayInSeason):factor(year)1991 15.226 18.434 29.971 < 2e-16 ***
## s(dayInSeason):factor(year)1992  9.305 11.504 44.530 < 2e-16 ***
## s(dayInSeason):factor(year)1993 10.170 12.578 48.865 < 2e-16 ***
## s(dayInSeason):factor(year)1994  5.689  7.078 47.483 < 2e-16 ***
## s(windSpeed)                     7.798  9.655  6.015 4.4e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.867   Deviance explained = 88.8%
## GCV = 3.5274  Scale est. = 2.9551    n = 334
```

10F

Show the four penalized spline fits for the effect of dayInSeason on the mean response.

```
par(mfrow=c(2,2))
plot(fit3, shade=TRUE, col="blue", rug=TRUE, select=c(1)
     , main="dayInSeason for 1991"
     , ylab="pollen count"
     )
plot(fit3, shade=TRUE, col="blue", rug=TRUE, select=c(2)
     , main="dayInSeason for 1992"
     , ylab="pollen count"
     )
plot(fit3, shade=TRUE, col="blue", rug=TRUE, select=c(3)
     , main="dayInSeason for 1993"
     , ylab="pollen count"
     )
plot(fit3, shade=TRUE, col="blue", rug=TRUE, select=c(4)
     , main="dayInSeason for 1994"
     , ylab="pollen count"
     )
```

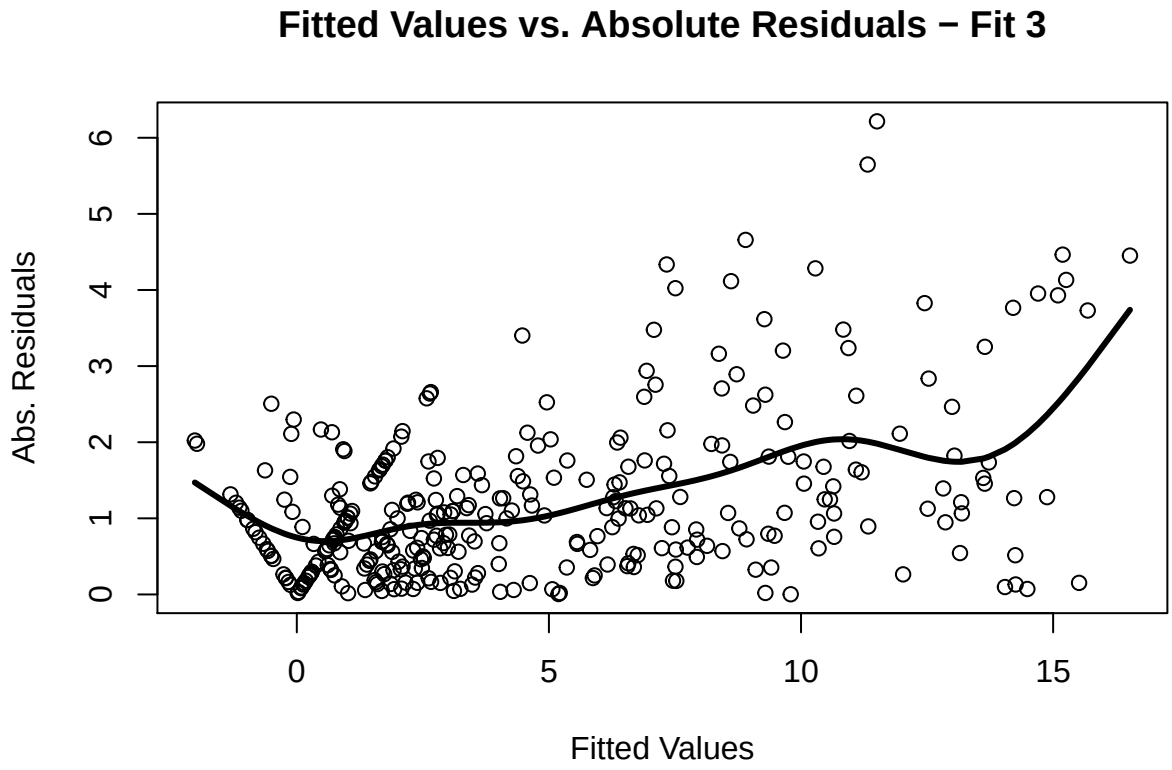



```
par(mfrow=c(1,1))
```

Question 2

For the square root transformation (fit3), see whether the idea of constant variance is reasonable.

```
# obtain the absolute residuals and the fitted values
abs_res <- abs(residuals(fit3))
fit_vals <- fitted(fit3)
# fit a new smoothed regression
resid_mod <- mgcv::gam(abs_res ~ s(fit_vals))
# initial plot
plot(fit_vals, abs_res
     , main="Fitted Values vs. Absolute Residuals - Fit 3"
     , xlab="Fitted Values"
     , ylab="Abs. Residuals"
     )
ord <- order(fitted(fit3))
# plot the line over the data
lines(fit_vals[ord], fitted(resid_mod)[ord], lwd=3)
```




```
max(fitted(resid_mod)) /  min(fitted(resid_mod))
```

```
## [1] 5.343122
```

From the above plot, it does not appear that the assumption of constant variance is reasonable. As our fitted values increase, the variance of our absolute residuals increases. Also our fitted value ratio heuristic shows that we have evidence of heteroscedacticity.

Question 3

From the estimated scale in the QuasiPoisson model, does the scale parameter imply that we may have concern about overdispersion?

```
print("Scale Parameter given by:")
```

```
## [1] "Scale Parameter given by:"
```

```
print(fit2$scale)
```

```
## [1] 8.54425
```

Considering that we are looking for a scale value of 1, I believe we do have legitimate concerns about overdispersion.

Question 4

Test whether the QuasiPoisson fit is statistically significantly different from the Poisson fit.

```
anova(fit1, fit2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: pollenCount ~ factor(year) + s(dayInSeason, k = 27, by = factor(year)) +
##     temperatureResidual + rain + s(windSpeed, k = 27)
## Model 2: pollenCount ~ factor(year) + s(dayInSeason, k = 27, by = factor(year)) +
##     temperatureResidual + rain + s(windSpeed, k = 27)
##   Resid. Df Resid. Dev      Df Deviance  Pr(>Chi)
## 1    205.43     1567.6
## 2    269.68     2255.5 -64.248  -687.93 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

It appears that we have statistically significant differences between the Poisson and QuasiPoisson fits. The QuasiPoisson is preferable in this case.

Question 5

In the Poisson fit, is there evidence that there is a year by dayInSeason interaction?

```
mgcv::summary.gam(fit1)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## pollenCount ~ factor(year) + s(dayInSeason, k = 27, by = factor(year)) +
##     temperatureResidual + rain + s(windSpeed, k = 27)
##
## Parametric coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.797350   0.083186  21.606   <2e-16 ***
## factor(year)1992     -2.939237  19.585504  -0.150   0.8807
## factor(year)1993     -1.422070   0.633147  -2.246   0.0247 *
## factor(year)1994    -44.641947  74.797144  -0.597   0.5506
## temperatureResidual   0.054357   0.004504  12.068   <2e-16 ***
## rain                  0.819443   0.059913  13.677   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                    edf Ref.df Chi.sq p-value
## s(dayInSeason):factor(year)1991 25.89  25.99 3287.6  <2e-16 ***
## s(dayInSeason):factor(year)1992 22.23  22.48 2343.0  <2e-16 ***
## s(dayInSeason):factor(year)1993 24.37  24.95 2439.6  <2e-16 ***
## s(dayInSeason):factor(year)1994 22.87  23.21 1103.9  <2e-16 ***
## s(windSpeed)                    25.40  25.94  711.1  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.888   Deviance explained = 94.6%
## UBRE = 4.4524  Scale est. = 1         n = 334
```

From the above output, it appears that we have a significant interaction between the year and dayInSeason

variables.

We can also create similar model that does not have the interaction, and test them against each other.

```
smaller_fit1 <- mgcv::gam(pollenCount ~ factor(year) + s(dayInSeason, k=27)
                  + temperatureResidual + rain + s(windSpeed, k=27)
                 , data=ragweed
                 , family=poisson
                  )
anova(fit1, smaller_fit1, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: pollenCount ~ factor(year) + s(dayInSeason, k = 27, by = factor(year)) +
##     temperatureResidual + rain + s(windSpeed, k = 27)
## Model 2: pollenCount ~ factor(year) + s(dayInSeason, k = 27) + temperatureResidual +
##     rain + s(windSpeed, k = 27)
##   Resid. Df Resid. Dev      Df Deviance  Pr(>Chi)
## 1    205.43     1567.6
## 2    276.14     3470.8 -70.717  -1903.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both evaluations suggest significant evidence of a year by dayInSeason interaction.

Question 6

Remove the interaction from the model, still doing the quasipoisson; do a stepwise regression - which model is seleted?

```
detach("package:mgcv", unload=TRUE)
library("gam")
```

```
## Warning: namespace 'mgcv' is not available and has been replaced
## by .GlobalEnv when processing object 'call.'

## Warning: namespace 'mgcv' is not available and has been replaced
## by .GlobalEnv when processing object 'call.'

## Warning: namespace 'mgcv' is not available and has been replaced
## by .GlobalEnv when processing object 'call.'
```

For the following, the quasipoisson response distribution appears to be incompatible with the step.gam function. I will use the gaussian response distribution for model selection.

```
gauss_init <- gam::gam(sqrt(pollenCount) ~ factor(year) + dayInSeason
                  + temperatureResidual + rain + windSpeed
                 , data=ragweed
                 , family=gaussian(link="identity")
                  )


gauss_step <- gam::step.gam(gauss_init, scope=list(
                        "year" = ~1 + year
                        , "dayInSeason" = ~1 + dayInSeason + s(dayInSeason, 2)
                        , "temperatureResidual" = ~1 + temperatureResidual
```

```
                        , "rain" = ~1 + rain
                        , "windSpeed" = ~1 + windSpeed + s(windSpeed, 2)
                        )
                          )
```

```
## Start:  sqrt(pollenCount) ~ factor(year) + dayInSeason + temperatureResidual +      rain + windSpeed
```

```
## Warning: namespace 'mgcv' is not available and has been replaced
## by .GlobalEnv when processing object 'call.'
```

```
## Step:1 sqrt(pollenCount) ~ factor(year) + s(dayInSeason, 2) + temperatureResidual +      rain + windS
```

The above GLM results in the following model:

```
print(names(gauss_step$model)[-1])
```

```
## [1] "factor(year)"        "s(dayInSeason, 2)"   "temperatureResidual"
## [4] "rain"                "windSpeed"
```

Question 7

Fit a model using cosso. COSSO does not have the Poisson or QuasiPoisson response distributions available, so I will use Guassian and transform the response in some of the models.

```
Y <- ragweed$pollenCount
Y_alt <- log(ragweed$pollenCount)
Y_alt2 <- sqrt(ragweed$pollenCount)
X <- ragweed[, c(2, 3, 5, 6, 7)]
# make sure everything is in 'integer format'
X$temperatureResidual <- as.integer(X$temperatureResidual)
X$windSpeed <- as.integer(X$windSpeed)
str(X)
```

```
## 'data.frame':    334 obs. of  5 variables:
##  $ year               : int  1991 1991 1991 1991 1991 1991 1991 1991 1991 1991 ...
##  $ dayInSeason        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ temperatureResidual: int  0 -4 -3 1 8 8 1 -2 -5 -1 ...
##  $ rain               : int  1 0 1 1 1 1 1 1 1 1 ...
##  $ windSpeed          : int  10 9 8 11 7 7 5 10 5 8 ...
```

```
cosso_mod1 <- cosso::cosso(x=X, y=Y, family="Gaussian")
```

```
## Error in solve.default(A + 1e-07 * diag(nrow(A)), b): system is computationally singular: reciprocal
```

```
cosso_mod2 <- cosso::cosso(x=X, y=Y_alt, family="Gaussian")
```

```
## Error in solve.default(A + 1e-07 * diag(nrow(A)), b): system is computationally singular: reciprocal
```

```
cosso_mod3 <- cosso::cosso(x=X, y=Y_alt2, family="Gaussian")
```

```
## Error in solve.default(A + 1e-07 * diag(nrow(A)), b): system is computationally singular: reciprocal
```

None of the cosso models were able to run successfully.