

# Carp Count Data

*David T. Callaghan*

*18 March, 2016*

## Background

The common carp *Cyprinus carpio* (henceforth carp) is one of the most trans-located species in the world, established on every continent except Antarctica. Carp have two basic habitat requirements: 1) a shallow marsh environment with abundant vegetation; 2) a deeper area to retreat to during colder months (McCrimmon 1968). Carp spawn in shallow flooded areas with abundant fixed vegetation on which eggs are deposited (Crivelli 1981). Spawning begins when water temperatures are  $\sim 15\text{--}16^{\circ}\text{C}$  (Crivelli 1981). Carp generally spawn in the spring (McCrimmon 1968), but can span March–August and even into October (Crivelli 1981). Most carp show high site fidelity, but a small percentage of the population may also exhibit high mobility (Crook 2004; Stuart and Jones 2006).

## Objectives

- 1) Determine what environmental variables (i.e. temperature and discharge) drive carp migration into the Delta Marsh using camera trap count data.
- 2) Determine the effect of sampling frequency on model results.

## Data exploration

I will loosely follow the protocol by Zuur et al. (2010) for exploring the data to avoid common statistical problems including type I (i.e. rejecting the null hypothesis when it is true) or type II errors (i.e. failure to reject the null hypothesis when it is untrue).

### 1. Data Distribution

A good place to start exploring the data is looking at the distribution of our response variable—carp counts. The distribution will give us a good indication of what kind of analysis we should use to deal with our data and if any problems may need to be addressed (i.e. many zeros). Because we are modelling count data, a generalized linear model (GLM) is an appropriate analysis (Zuur, Ieno, and Elphick 2010). The Poisson or negative binomial distributions are what we would expect. Looking at Figure 1, we quickly realize we are dealing with many zeros! Therefore, a Poisson or negative binomial GLM will likely produce biased parameter estimates and standard errors as well as over-dispersion. A zero-inflated or zero-altered GLM will likely be an appropriate analyses here.

### 2. Dealing with the zeros

Lets make sure we are dealing with true zero inflation. I have run the following models to see which ones produce similar number of zeros as our data:

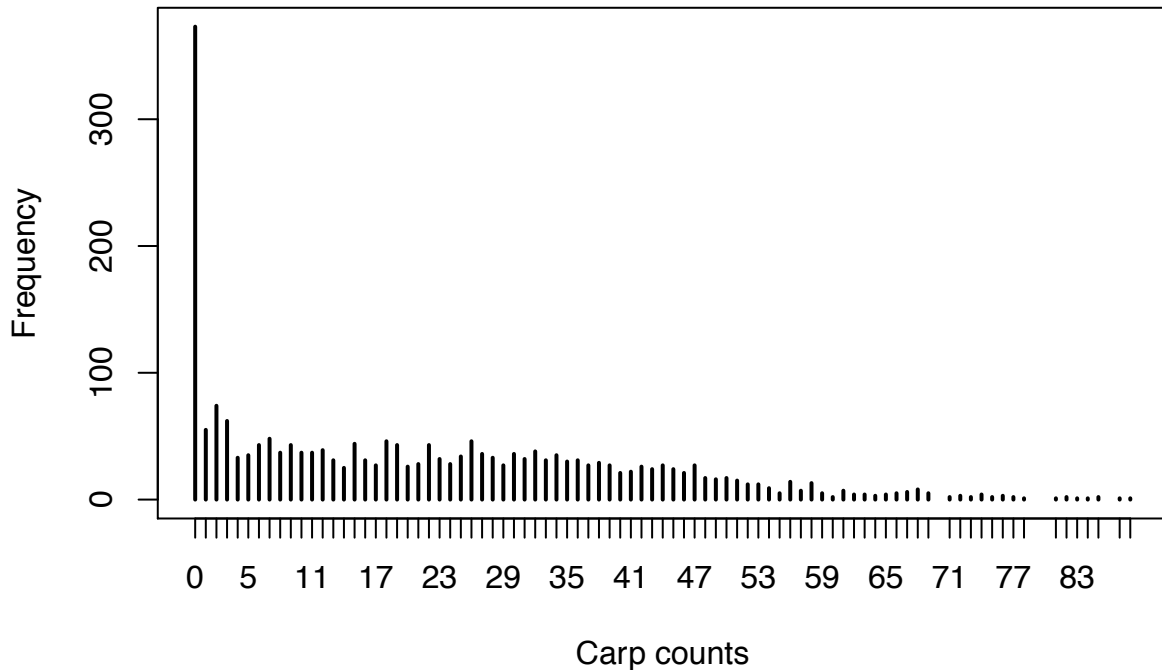


Figure 1: Frequency distribution of raw (unadjusted) carp count data. Notice the large number of zeros—a good indication of zero inflation

```
#Poisson glm
pois<-glm(Count~Discharge+Temp+Pelicans+Time,offset=log(adjArea), family="poisson", data=Z)

#Negative binomial glm model
negb<-glm.nb(Count~Discharge+Temp+Pelicans+Time+offset(log(adjArea)),data=Z)

#zero inflated poisson
zip<-zeroinfl(Count~Discharge+Temp++Pelicans+Time+offset(log(adjArea))|Pelicans+fExposure+fWaterClarity+
#zero inflated negative binomial
zinb<-zeroinfl(Count~Discharge+Temp+Pelicans+Time+offset(log(adjArea))|Pelicans+fExposure+fWaterClarity+
#zero altered poisson
zap<-hurdle(Count~Discharge+Temp+Pelicans+Time+offset(log(adjArea))|Pelicans+fExposure+fWaterClarity+f
#zero altered negative binomial
zanb<-hurdle(Count~Discharge+Temp+Pelicans+Time+offset(log(adjArea))|Pelicans+fExposure+fWaterClarity+f

##  Obs Pois   NB   ZIP  ZINB   ZAP  ZANB
##  373    4  178  373  364   373  364
```

It is quite apparent that only the zero-inflated models (ZIP, ZAP, ZINB and ZANB) have similar zero counts to our observations.

How we deal with the zeros is related to what kind of zeros we have. In general, techniques for dealing with zero inflation have two parts: a binomial part that deals with the zeros and a Poisson (or negative binomial) part that deals with the count data. zero-inflated models, or mixture-models, split the zeros into true zeros (fish truly absent) and false zeros (fish present but not seen). The true zeros are modelled in the Poisson GLM, while the process generating the false zeros is modelled in the binomial GLM. Two-part or hurdle models

do not discriminate between true and false zeros, the presence of an animal is the result of some covariate mechanism crossing a hurdle. The hurdle model is most appropriate when there is little chance of missing any items in the counts (Ver Hoef and Jansen 2007). In our case, some zeros may be false zeros (missed counts)—as a result of image obstruction, exposure, water clarity or image clarity—or true zeros (i.e. fish are not present because conditions are not appropriate). Therefore, I feel a zero-inflated mixture-model would best deal with the zero inflation.

### 3. Outliers

Now that we have an idea of what analysis we can perform lets look at the data in more detail. Outliers can be a problem for Poisson GLM (similarly for zero-inflated models) analyses and may cause over-dispersion. I will define outliers here as observations which values are relatively larger or smaller to the majority of of observations. Here I show a boxplot (Figure 2a) and a Cleveland dotplot (Figure 2b ) of 2222 carp count observations. The boxplot visualizes the median and spread of the data. Observations outside of the whiskers are labelled as outliers. Figure 1a shows that there are potentially 9 outliers. A good way to check if these are in fact outliers is a Cleveland dot plot (Figure 2b), in this graph the row number of observations is plotted vs. the frequency of carp, providing more detailed information than a boxplot. Figure 2b reveals that the possible outliers are not really outliers at all because they follow the pattern of peaks displayed by the observations.

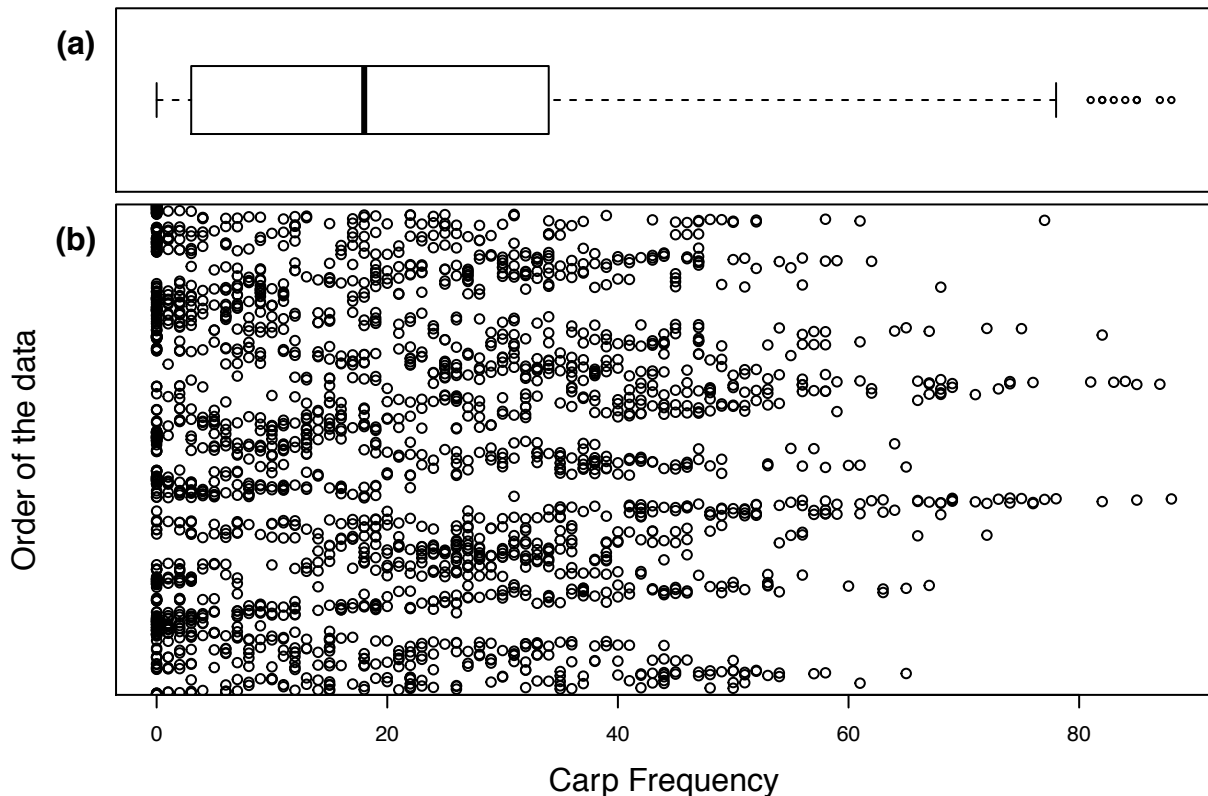


Figure 2: (a) Boxplot of carp frequency counts from 2222 observations taken at the same location. The line in the middle of the box represents the median, and the lower and upper ends of the box are the 25% and 75% quartiles respectively. The lines indicate 1.5 times the size of the hinge, which is the 75% minus 25% quartiles. Points beyond these lines are considered to be outliers. (b) Cleveland dot plot of the same data. The horizontal axis represents the carp counts, and the vertical axis corresponds to the order of the data.

Figure 2 displays a multi-panel Cleveland dot plot for our carp count data and all of our potential covariates that could influence the counts, including water discharge, water temperature, and time; as well as covariates

that may influence the zeros (or false zeros) including dead fish, pelican counts, exposure factor, image clarity factor, water clarity factor and percent obstructed. For the most part the covariates look fine except for the large values (points far to the right) in the dead fish and percent obstructed panels. It appears the large number of dead fish obstructed 83% of the image. The large number of dead fish is a rare observation and may be influential on our parameter estimates. We will have to investigate how sensitive the model is to these large values and decide if this observation is a candidate for removal.

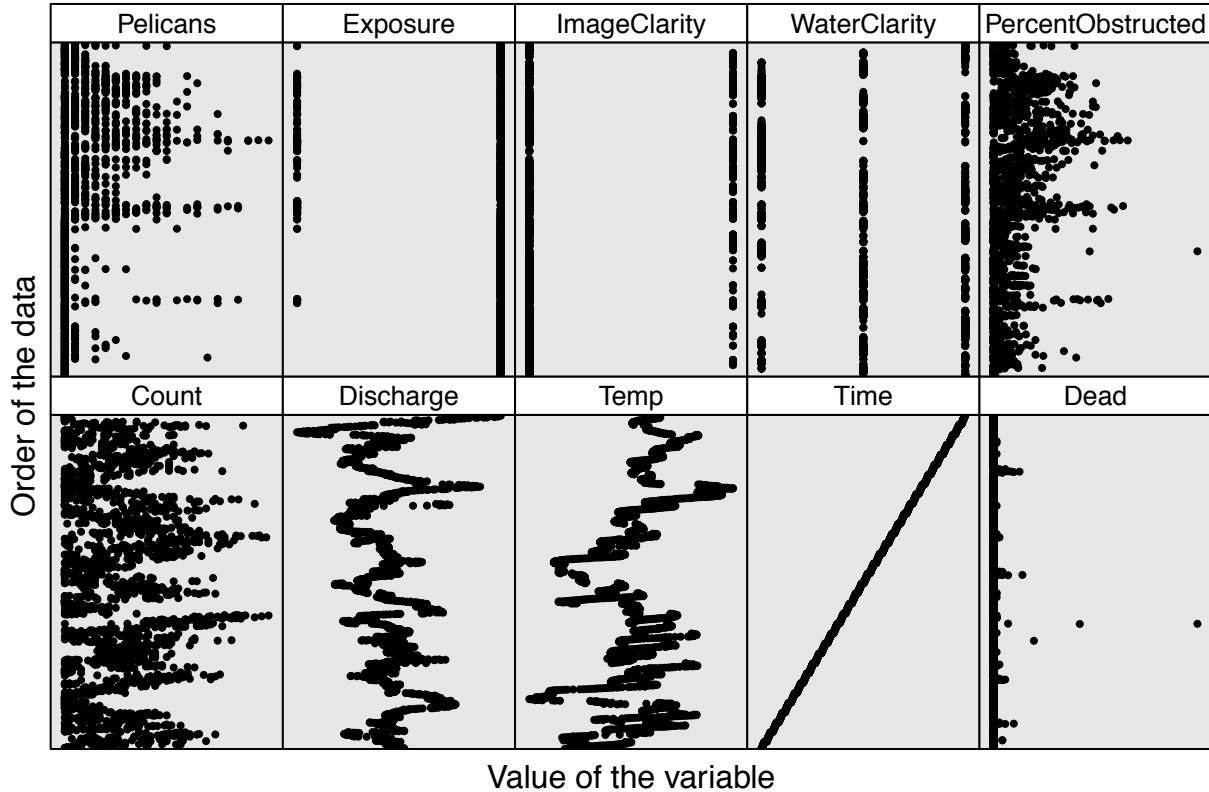


Figure 3: Multi-panel Cleveland dotplot for carp counts and ten covariates that may influence the counts. The plots are ordered by date along the y-axis. Notice the possible influential values in Obstructed and Dead—this observation may be a candidate for removal if model parameters are found to be sensitive to these inputs.

Over the study duration, the viewable area of the picture frame ranges from 3.6328499–4.3961477 m<sup>2</sup>. Further reduction in visible area as a result of obstructions, such as pelicans and dead fish (See Figure 3), required an offset for adjusted area (picture frame area X percent visible area). The results of our model will therefore be a density Carp per m<sup>2</sup>.

#### 4. Collinearity among covariates

Ignoring collinearity among covariates may lead to a confusing statistical output with nothing significant. This is because collinearity results in inflated standard errors of parameters which in turn increase *P*-values, making it difficult to detect an effect. We can easily test for collinearity by looking at the variance inflation factors (VIF), covariates with VIF greater than 3 will be sequentially removed. Unfortunately the *vif* function from the *car* package does not allow zero-inflated models—thus, I have split up our covariates into two groups: Poisson covariates including water discharge, water temperature, and time; as well as binomial covariates including water discharge, water temperature, time, presence of pelicans, exposure factor, image clarity factor and water clarity factor. This way I can examine the Poisson covariate VIF in a Poisson GLM and the binomial covariate VIF in a binomial GLM using the *vif* function.

```
##          VIF
## Discharge 1.230574
## Temp      1.095940
## Pelicans  1.125772
## Time      1.410620
```

Notice that all of our Poisson covariates VIF are  $< 3$ . No collinearity! Now let's look at the binomial side.

```
##          VIF
## Discharge 1.176526
## Temp      1.212129
## Time      1.138461
## Pelicans  1.532177
## fExposure 1.476387
## fWaterClarity 1.137934
## fImageClarity 1.036255
```

Our four covariates do not reveal any collinearity. Now let's look at the relationships between our covariates and the response variable.

## 5. Relationships Y & X

Looking at the relationship between Y (carp counts) and X covariates (discharge, temperature, pelicans and time) we begin to see strong nonlinear effects of discharge and temperature (Figure 4). Abundance appears to increase at negative discharges close to 0 and temperatures between 16–20°C. These nonlinear trends should be modelled with an additive model (i.e. GAM) or linear model with quadratic relationships.

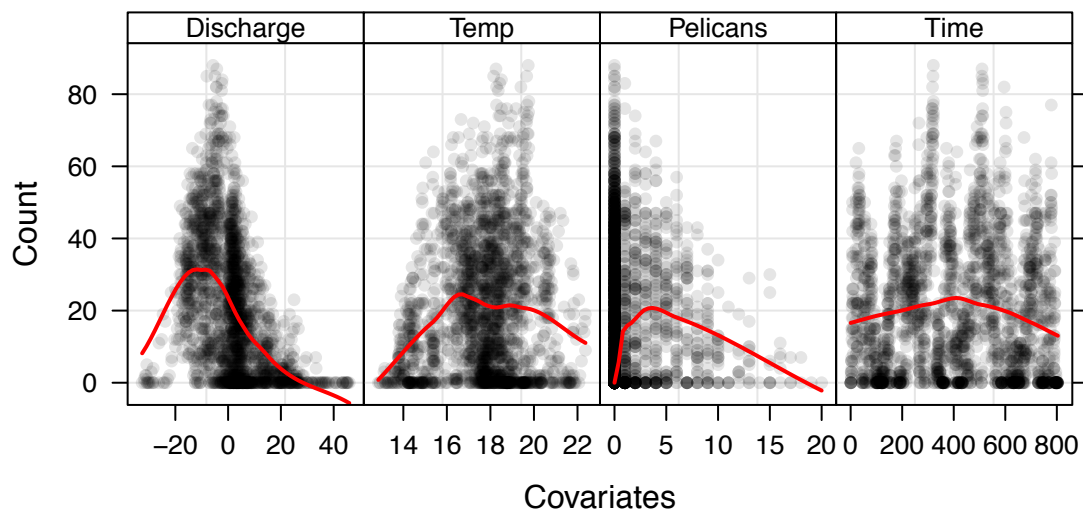


Figure 4: Relationship between carp counts and four potential covariates. The red line represents a LOESS smoother (span=2/3, degree=1) to visualize relationships between carp counts and covariates.

## 6. Response variable independence

A very important assumption of most statistical techniques is independence among observations. Ignoring dependence among observations (autocorrelation) can lead to underestimates of standard errors and increased

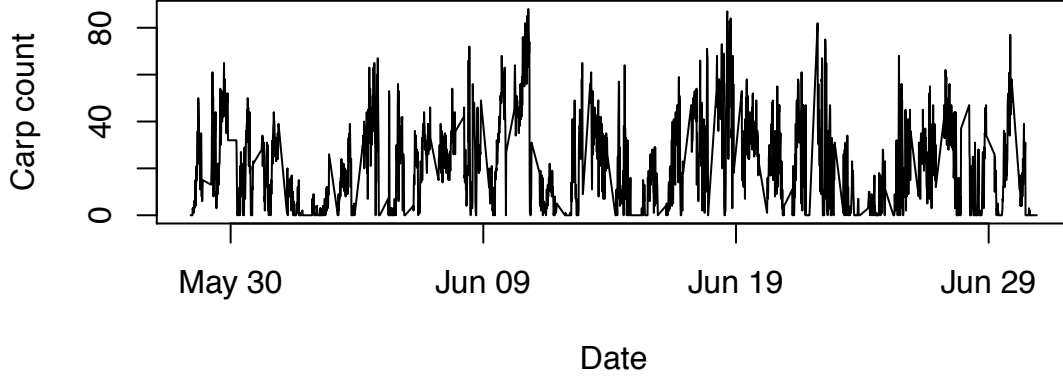


Figure 5: Carp count time-series.

false positives. The carp count data is part of a time-series analysis (Figure 5), thus dependence among observations is a potential problem.

Figure 6 displays the autocorrelation between lags for the carp count data and confirms that we have dependence among observations—all lags shown are significantly correlated. At lag 1, the correlation of 0.78 exponentially diminishes at larger lags is suggesting of a stationary autoregressive model.

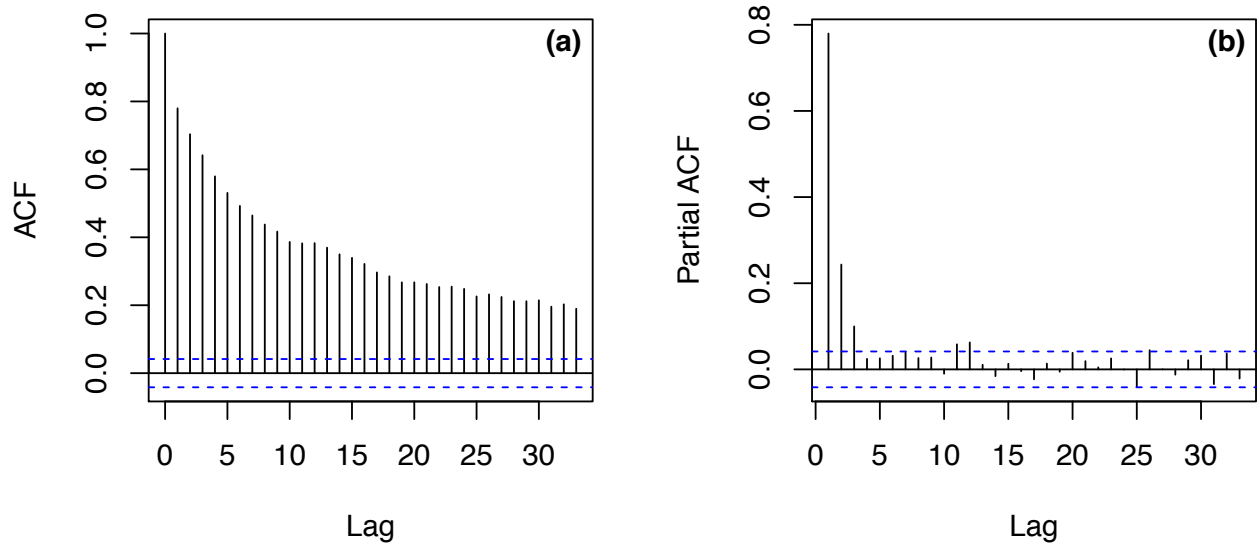


Figure 6: (a) Autocorrelation function plot and (b) Partial autocorrelation function plot for carp count data. The blue dashed lines represents significant correlations such that any lags greater than the positive blue line or less than the negative blue line are significantly correlated

To determine the order of the autoregressive process we can examine the partial autocorrelation function plot (Figure 6b). We see that significant partial autocorrelation up to lag 3 which suggests that a third order autoregressive process (AR3) could be used to model the data.

## 7. Model fitting

This data set has a few problems to overcome. first, we are dealing zero inflated counts so we need a model that can handle the overdispersion caused by the zeros—I will start with a zero-inflated poisson model. Second, we have dependence among observations which will need to be addressed in order to avoid unrealistic standard errors and false positives—I will start with adding a smoothed trend term for time. Lastly, we have

non-linear relationships between our count variable and explanatory variables discharge and temperature—I have added quadratic terms to account for these relationships. Let's take a look at our initial model:

```
##
## Call:
## zeroinfl(formula = Count ~ offset(log(adjArea)) + Discharge + I(Discharge^2) +
##      Temp + I(Temp^2) + Pelicans + ns(Time, df = 3) | Discharge +
##      I(Discharge^2) + Temp + I(Temp^2) + ns(Time, df = 3) + Pelicans +
##      fExposure + fWaterClarity + fImageClarity, data = Z, dist = "poisson")
##
## Pearson residuals:
##      Min      1Q   Median      3Q      Max
## -5.16742 -1.09920 -0.09712  1.19986 20.72441
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.8277525   1.4342261  -3.366 0.000762 ***
## Discharge     -0.0518203   0.0010064 -51.492 < 2e-16 ***
## I(Discharge^2) -0.0026248   0.0000568 -46.209 < 2e-16 ***
## Temp          0.6798871   0.1611804   4.218 2.46e-05 ***
## I(Temp^2)     -0.0169717   0.0044420  -3.821 0.000133 ***
## Pelicans      -0.0667707   0.0022573 -29.579 < 2e-16 ***
## ns(Time, df = 3)1 0.3093942   0.0216788  14.272 < 2e-16 ***
## ns(Time, df = 3)2 0.3593292   0.0433866   8.282 < 2e-16 ***
## ns(Time, df = 3)3 -0.2875832   0.0216134 -13.306 < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   14.8435085   4.9011680   3.029 0.002457 **
## Discharge      0.0126885   0.0083804   1.514 0.130010
## I(Discharge^2) 0.0002554   0.0004175   0.612 0.540602
## Temp         -2.2310607   0.5771916  -3.865 0.000111 ***
## I(Temp^2)      0.0663798   0.0168345   3.943 8.04e-05 ***
## ns(Time, df = 3)1 0.0753005   0.3440237   0.219 0.826742
## ns(Time, df = 3)2 0.6166728   0.6377267   0.967 0.333551
## ns(Time, df = 3)3 1.3321913   0.2689004   4.954 7.26e-07 ***
## Pelicans       0.1315337   0.0328897   3.999 6.35e-05 ***
## fExposure2     -1.0982663   0.3308181  -3.320 0.000901 ***
## fWaterClarity2  1.9405159   0.3177129   6.108 1.01e-09 ***
## fWaterClarity3  3.6085964   0.3453328  10.450 < 2e-16 ***
## fImageClarity2 -0.6600736   0.2650371  -2.490 0.012757 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 32
## Log-likelihood: -1.212e+04 on 22 Df
```

We added quadratic terms for discharge and temperature so I will double check the variance inflation factors to ensure we do not have collinearity.

```
##              VIF
## Discharge    1.329149
## I(Discharge^2) 1.227982
```

```
## Temp          17.232253
## I(Temp^2)     17.251432
## Pelicans      1.085463
## ns(Time, df = 3) 1.085332
```

The temperature and temperature<sup>2</sup> covariates are correlated, let's centre these variables and re-test for collinearity.

```
##          VIF
## Discharge 1.329149
## I(Discharge^2) 1.227982
## Temp.c     1.105827
## I(Temp.c^2) 1.064911
## Pelicans    1.085463
## ns(Time, df = 3) 1.085332
```

Now let's re-run the zero inflated model to see if our results changed.

```
##
## Call:
## zeroinfl(formula = Count ~ offset(log(adjArea)) + Discharge + I(Discharge^2) +
## Temp.c + I(Temp.c^2) + Pelicans + ns(Time, df = 3) | Discharge +
## I(Discharge^2) + Temp.c + I(Temp.c^2) + ns(Time, df = 3) + Pelicans +
## fExposure + fWaterClarity + fImageClarity, data = Z, dist = "poisson")
##
## Pearson residuals:
##      Min      1Q   Median      3Q      Max
## -5.16742 -1.09920 -0.09712  1.19986 20.72422
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.889e+00  1.756e-02 107.600 <2e-16 ***
## Discharge     -5.182e-02  7.819e-04 -66.276 <2e-16 ***
## I(Discharge^2) -2.625e-03  5.551e-05 -47.288 <2e-16 ***
## Temp.c        7.896e-02  3.386e-03  23.318 <2e-16 ***
## I(Temp.c^2)   -1.697e-02  1.488e-03 -11.409 <2e-16 ***
## Pelicans      -6.677e-02  2.256e-03 -29.592 <2e-16 ***
## ns(Time, df = 3)1  3.094e-01  2.121e-02  14.588 <2e-16 ***
## ns(Time, df = 3)2  3.593e-01  4.287e-02   8.382 <2e-16 ***
## ns(Time, df = 3)3 -2.876e-01  2.123e-02 -13.547 <2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.8496381  0.5106330 -7.539 4.74e-14 ***
## Discharge      0.0126888  0.0084199  1.507 0.131811
## I(Discharge^2)  0.0002554  0.0004190  0.610 0.542143
## Temp.c        0.1192868  0.0451877  2.640 0.008295 **
## I(Temp.c^2)    0.0663794  0.0150018  4.425 9.65e-06 ***
## ns(Time, df = 3)1  0.0752994  0.3465954  0.217 0.828010
## ns(Time, df = 3)2  0.6166158  0.6379996  0.966 0.333803
## ns(Time, df = 3)3  1.3321822  0.2676777  4.977 6.46e-07 ***
## Pelicans       0.1315330  0.0326904  4.024 5.73e-05 ***
## fExposure2     -1.0982792  0.3296562 -3.332 0.000864 ***
```



```
## fWaterClarity2      1.9405022  0.3204797   6.055 1.40e-09 ***
## fWaterClarity3      3.6085895  0.3378961  10.680 < 2e-16 ***
## fImageClarity2     -0.6600696  0.2633723  -2.506 0.012203 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 29
## Log-likelihood: -1.212e+04 on 22 Df
```

The zero inflation poisson model uses an offset of *adjArea* which is the percent of unobstructed frame multiplied by the maximum area of the frame to change our raw count into a density (fish m<sup>-2</sup>). We can see that our model has several non-significant predictors in the zero-inflation model including discharge, discharge<sup>2</sup> and 2 of the time variables natural splines. Before we start model selection we should check if the zero-inflated poisson model still has significant overdispersion. This is done by comparing the same model formula in a zero-inflated negative binomial model using the likelihood ratio test.

```
zinz<-zeroinfl(Count~offset(log(adjArea))+Discharge+I(Discharge^2)+Temp.c+I(Temp.c^2)
              +Pelicans+ns(Time,df=3)|Discharge+I(Discharge^2)+Temp.c+I(Temp.c^2)
              +ns(Time,df=3)+Pelicans+fExposure+fWaterClarity+fImageClarity,
              dist="negbin", data=Z)
library(lmtest)
lrtest(zip,zinz)
```

```
## Likelihood ratio test
##
## Model 1: Count ~ offset(log(adjArea)) + Discharge + I(Discharge^2) + Temp.c +
##      I(Temp.c^2) + Pelicans + ns(Time, df = 3) | Discharge + I(Discharge^2) +
##      Temp.c + I(Temp.c^2) + ns(Time, df = 3) + Pelicans + fExposure +
##      fWaterClarity + fImageClarity
## Model 2: Count ~ offset(log(adjArea)) + Discharge + I(Discharge^2) + Temp.c +
##      I(Temp.c^2) + Pelicans + ns(Time, df = 3) | Discharge + I(Discharge^2) +
##      Temp.c + I(Temp.c^2) + ns(Time, df = 3) + Pelicans + fExposure +
##      fWaterClarity + fImageClarity
##   #Df   LogLik Df  Chisq Pr(>Chisq)
## 1  22 -12118.0
## 2  23 -8002.5  1 8230.8 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(zip,zinz)
```

```
##      df      AIC
## zip  22 24279.92
## zinz  23 16051.09
```

The likelihood ratio test and AIC comparison reveal the zero-inflated negative binomial model performs significantly better than the zero-inflated poisson model. Now lets examine the output of our zero-inflated negative binomial model:

```
##
## Call:
## zeroinfl(formula = Count ~ offset(log(adjArea)) + Discharge + I(Discharge^2) +
```

```

##      Temp.c + I(Temp.c^2) + Pelicans + ns(Time, df = 3) | Discharge +
##      I(Discharge^2) + Temp.c + I(Temp.c^2) + ns(Time, df = 3) + Pelicans +
##      fExposure + fWaterClarity + fImageClarity, data = Z, dist = "negbin")
##
## Pearson residuals:
##      Min      1Q  Median      3Q      Max
## -1.5831 -0.6786 -0.0962  0.5158 10.2340
##
## Count model coefficients (negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.8238906  0.0549784  33.175 < 2e-16 ***
## Discharge      -0.0465465  0.0020347 -22.876 < 2e-16 ***
## I(Discharge^2) -0.0019242  0.0001266 -15.196 < 2e-16 ***
## Temp.c         0.0987924  0.0101002   9.781 < 2e-16 ***
## I(Temp.c^2)    -0.0091895  0.0042112  -2.182  0.02910 *
## Pelicans       -0.0717301  0.0065416 -10.965 < 2e-16 ***
## ns(Time, df = 3)1 0.3738051  0.0692895   5.395 6.86e-08 ***
## ns(Time, df = 3)2 0.3488192  0.1332084   2.619  0.00883 **
## ns(Time, df = 3)3 -0.2904917  0.0654607  -4.438 9.09e-06 ***
## Log(theta)      1.0230179  0.0401778  25.462 < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.9480884  0.5412319  -7.295 2.99e-13 ***
## Discharge       0.0129021  0.0085138   1.515 0.129661
## I(Discharge^2)  0.0005230  0.0004114   1.271 0.203643
## Temp.c         0.1219892  0.0458177   2.662 0.007756 **
## I(Temp.c^2)     0.0663073  0.0151494   4.377 1.20e-05 ***
## ns(Time, df = 3)1 0.1267602  0.3556590   0.356 0.721534
## ns(Time, df = 3)2 0.5440740  0.6514553   0.835 0.403624
## ns(Time, df = 3)3 1.3398428  0.2713473   4.938 7.90e-07 ***
## Pelicans       0.1247192  0.0345787   3.607 0.000310 ***
## fExposure2     -1.1465047  0.3394628  -3.377 0.000732 ***
## fWaterClarity2  2.0799008  0.3633605   5.724 1.04e-08 ***
## fWaterClarity3  3.7312532  0.3797125   9.827 < 2e-16 ***
## fImageClarity2 -0.7313836  0.2748815  -2.661 0.007797 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 2.7816
## Number of iterations in BFGS optimization: 31
## Log-likelihood: -8003 on 23 Df

```

Looks like we still have some non-significant terms. I will sequentially remove predictors until they are all significant predictors in the model. First, I will remove Discharge<sup>2</sup> from the zero model because it has the highest *P-values* in the zero part of the model. I will continue to sequentially remove insignificant parameters (time and Discharge) until all parameters are significant. Here is the resulting model:

```

##
## Call:
## zeroinfl(formula = Count ~ offset(log(adjArea)) + Discharge + I(Discharge^2) +
##      Temp.c + I(Temp.c^2) + Pelicans + ns(Time, df = 3) | Temp.c +
##      I(Temp.c^2) + Pelicans + fExposure + fWaterClarity + fImageClarity,
##      data = Z, dist = "negbin")

```

```
##
## Pearson residuals:
##      Min      1Q   Median      3Q      Max
## -1.57077 -0.68924 -0.09733  0.51812  8.67596
##
## Count model coefficients (negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.8275225  0.0550115  33.221 < 2e-16 ***
## Discharge     -0.0470943  0.0020299 -23.201 < 2e-16 ***
## I(Discharge^2) -0.0019839  0.0001228 -16.156 < 2e-16 ***
## Temp.c         0.1001531  0.0101141   9.902 < 2e-16 ***
## I(Temp.c^2)    -0.0089944  0.0042157  -2.134  0.0329 *
## Pelicans      -0.0718087  0.0065290 -10.999 < 2e-16 ***
## ns(Time, df = 3)1  0.3761480  0.0693060   5.427 5.72e-08 ***
## ns(Time, df = 3)2  0.3418135  0.1332763   2.565  0.0103 *
## ns(Time, df = 3)3 -0.3050934  0.0655296  -4.656 3.23e-06 ***
## Log(theta)      1.0208428  0.0401875  25.402 < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.87293    0.46340  -8.358 < 2e-16 ***
## Temp.c         0.23740    0.03970   5.979 2.24e-09 ***
## I(Temp.c^2)    0.07558    0.01401   5.395 6.86e-08 ***
## Pelicans       0.13424    0.03356   4.000 6.34e-05 ***
## fExposure2     -1.05138    0.33766  -3.114  0.00185 **
## fWaterClarity2  2.03260    0.34676   5.862 4.58e-09 ***
## fWaterClarity3  3.89854    0.35191  11.078 < 2e-16 ***
## fImageClarity2 -0.54567    0.22902  -2.383  0.01719 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 2.7755
## Number of iterations in BFGS optimization: 26
## Log-likelihood: -8019 on 18 Df
```

Now that all out our predictors are significant let's compare our new model with the full model.

```
lrtest(zinb,zinb2)
```

```
## Likelihood ratio test
##
## Model 1: Count ~ offset(log(adjArea)) + Discharge + I(Discharge^2) + Temp.c +
##      I(Temp.c^2) + Pelicans + ns(Time, df = 3) | Discharge + I(Discharge^2) +
##      Temp.c + I(Temp.c^2) + ns(Time, df = 3) + Pelicans + fExposure +
##      fWaterClarity + fImageClarity
## Model 2: Count ~ offset(log(adjArea)) + Discharge + I(Discharge^2) + Temp.c +
##      I(Temp.c^2) + Pelicans + ns(Time, df = 3) | Temp.c + I(Temp.c^2) +
##      Pelicans + fExposure + fWaterClarity + fImageClarity
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1  23 -8002.5
## 2  18 -8018.7 -5 32.293  5.199e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(zinb,zinb2)
```

```
##      df      AIC
## zinb  23 16051.09
## zinb2 18 16073.39
```

Both the likelihood ratio test and AIC comparison reveal the full model is a better fit than the updated model, even with non significant parameters. Lets examine the diagnostics plot for the full model to see if the model is valid (Figure 7).

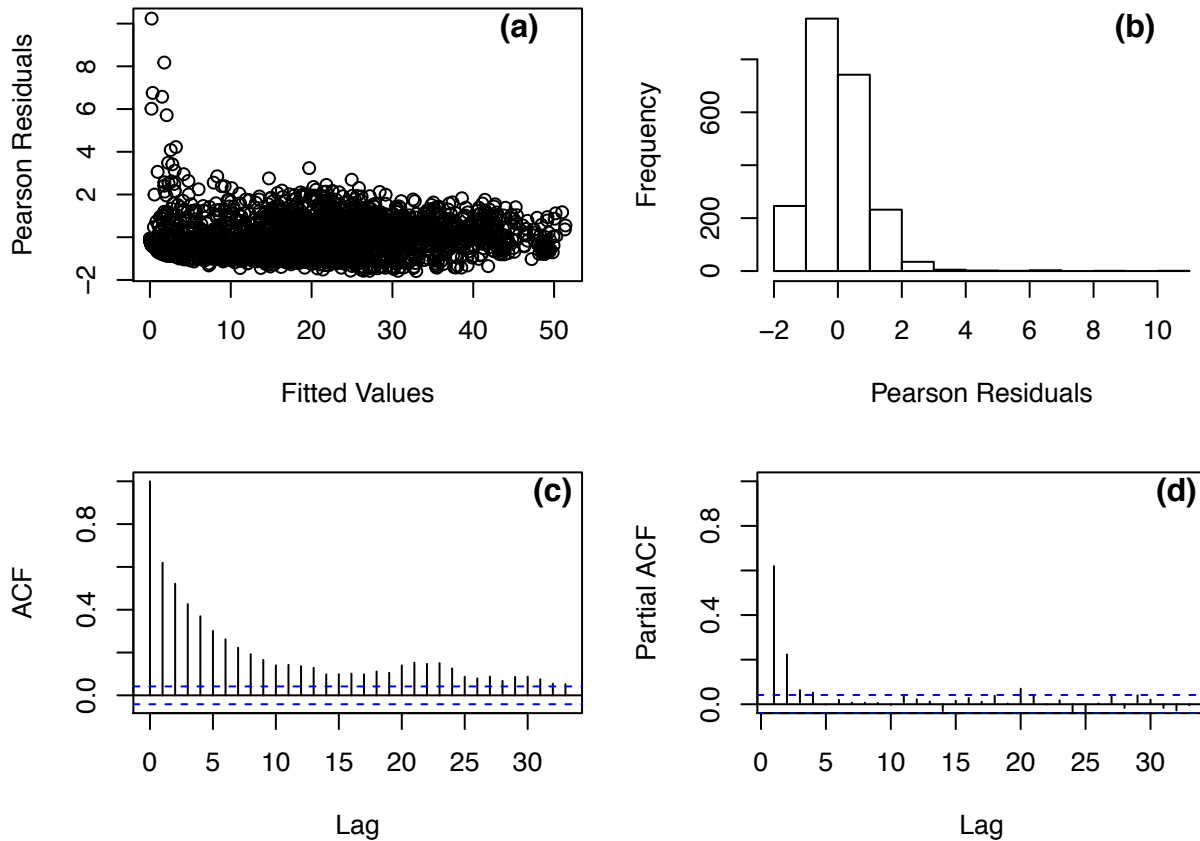


Figure 7: Model diagnostic plots. **a)** Pearson residuals versus fitted values, we should not see any clear pattern. **b)** histogram of pearson residuals, we should see residuals normally distributed around 0. **c)** Auto-correlation function of residuals and **d)** Partial autocorrelation function, vertical bars should be between the two horizontal blue lines if residuals are independent.

The diagnostic plots in figure 7 reveal that our model has residual problems (figure 7 a and b) and does not meet assumptions of independence (Figure 7 c and d). The time spline does not seem to reduce the autocorrelation in the model likely due to the strong serial correlation. The residual plots reveal that our model is unable to capture some of the variability in the data. We can examine what may be causing this through plotting our response versus our predictors and highlighting the points with large residual errors (Figure 8).

It appears from Figure 8a that all of the residual error is coming from an “interesting anomaly” of higher than expected fish counts at positive discharges  $>20 \text{ cm s}^{-1}$ . Unfortunately we do not have a variable to account for this occurrence, therefore high residual error occurs at these points. If we take a closer look at the raw data you will notice that these points for the most part are clustered together in time (ie. lines 1730–1734 and lines 1752–1754), thus accounting for autocorrelation may rectify this problem.

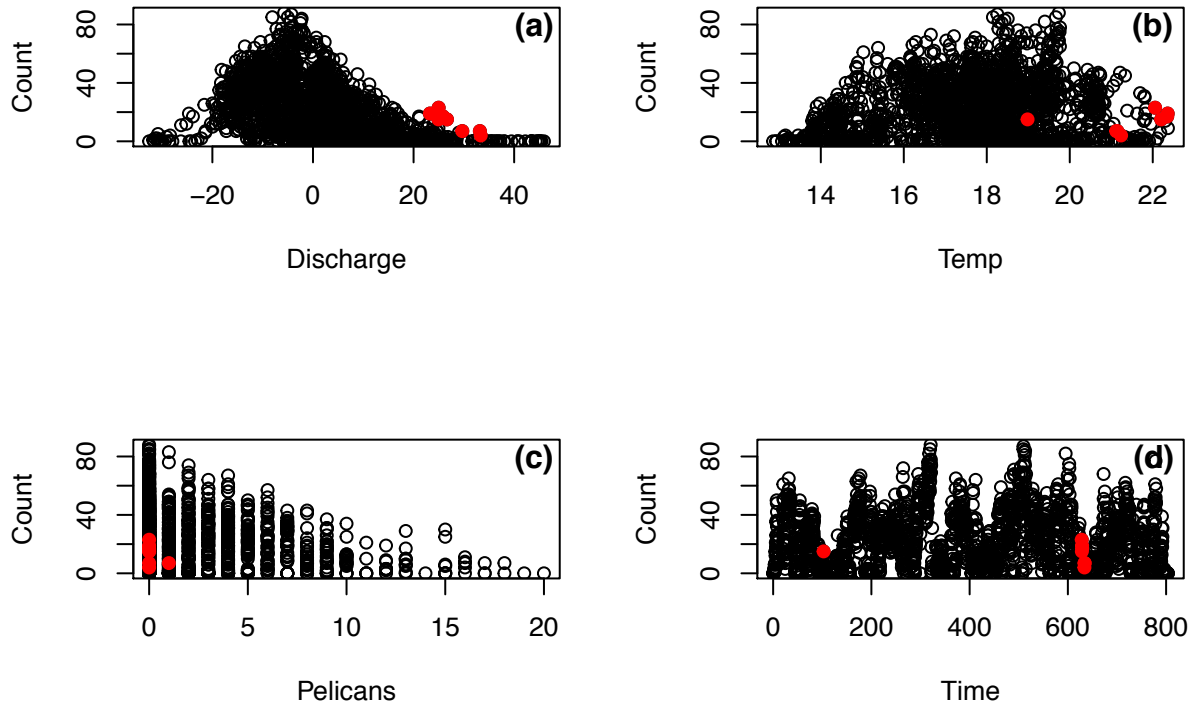


Figure 8: Count vs predictor plots. Black dots are raw data, red dots are observations with residuals  $> 4$  (i.e. poor fitting observations in the model)

##	Count	Discharge	Temp	DOY	Hour	Hour.M	Time	Dead	Pelicans	Exposure
## 288	15	26.643	18.983	152	16	16.75	102.75	0	0	2
## 1730	23	24.980	22.056	174	14	14.25	628.25	0	0	2
## 1731	15	24.943	22.203	174	14	14.50	628.50	0	0	2
## 1733	19	23.242	22.359	174	15	15.00	629.00	0	0	2
## 1734	17	25.858	22.348	174	15	15.25	629.25	0	0	2
## 1752	4	33.355	21.237	174	19	19.75	633.75	0	0	2
## 1753	7	29.689	21.145	174	20	20.00	634.00	0	1	2
## 1754	7	33.180	21.119	174	20	20.25	634.25	0	0	2
##	ImageClarity	WaterClarity	adjArea	PercentObstructed	Obs	Density				
## 288	1	3	4.211716	0.0	288	3.561494				
## 1730	1	2	3.542491	11.1	1730	6.492606				
## 1731	2	2	3.977092	0.0	1731	3.771600				
## 1733	2	2	3.979662	0.0	1733	4.774275				
## 1734	2	2	3.830908	3.8	1734	4.437590				
## 1752	1	3	3.930974	0.0	1752	1.017559				
## 1753	1	3	3.827324	2.7	1753	1.828954				
## 1754	1	3	3.930974	0.0	1754	1.780729				
##	fExposure	fImageClarity	fWaterClarity	bCount	fPelicans	Temp.c				
## 288	2	1	3	1	0	1.279231				
## 1730	2	1	2	1	0	4.352231				
## 1731	2	2	2	1	0	4.499231				
## 1733	2	2	2	1	0	4.655231				
## 1734	2	2	2	1	0	4.644231				
## 1752	2	1	3	1	0	3.533231				
## 1753	2	1	3	1	1	3.441231				
## 1754	2	1	3	1	0	3.415231				

Next let's add lagged response variables as predictors to the full model which will hopefully account for the autocorrelation among observations. I sequentially removed insignificant predictors until the resulting model:

```
##
## Call:
## zeroinfl(formula = Count ~ offset(log(adjArea)) + AR1 + AR2 + AR3 +
##      Discharge + I(Discharge^2) + Temp.c + Pelicans + Time | AR1 +
##      Temp.c + I(Temp.c^2) + Pelicans + fWaterClarity + fImageClarity +
##      Time, data = Z, dist = "negbin", x = TRUE)
##
## Pearson residuals:
##      Min      1Q  Median      3Q      Max
## -1.9091 -0.6405 -0.1444  0.4821  6.4606
##
## Count model coefficients (negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.179e+00  4.153e-02  28.397 < 2e-16 ***
## AR1           1.510e-02  1.075e-03  14.049 < 2e-16 ***
## AR2           5.814e-03  1.154e-03   5.036 4.74e-07 ***
## AR3           3.871e-03  1.066e-03   3.630 0.000283 ***
## Discharge     -2.518e-02  1.772e-03 -14.207 < 2e-16 ***
## I(Discharge^2) -1.370e-03  1.082e-04 -12.665 < 2e-16 ***
## Temp.c        3.079e-02  8.389e-03   3.670 0.000242 ***
## Pelicans      -4.067e-02  5.644e-03  -7.205 5.81e-13 ***
## Time          2.367e-04  6.912e-05   3.425 0.000616 ***
## Log(theta)    1.402e+00  4.422e-02  31.698 < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.3898025  0.4352232  -5.491 4.00e-08 ***
## AR1          -0.1675209  0.0163693 -10.234 < 2e-16 ***
## Temp.c       0.1250615  0.0463652   2.697 0.006990 **
## I(Temp.c^2)  0.0522976  0.0150401   3.477 0.000507 ***
## Pelicans     0.1342999  0.0284863   4.715 2.42e-06 ***
## fWaterClarity2 0.9585414  0.3861823   2.482 0.013061 *
## fWaterClarity3 1.9596312  0.3942031   4.971 6.66e-07 ***
## fImageClarity2 -0.5508982  0.2446019  -2.252 0.024308 *
## Time         0.0010673  0.0003742   2.852 0.004339 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 4.0619
## Number of iterations in BFGS optimization: 35
## Log-likelihood: -7607 on 19 Df
```

If we look at the diagnostic plots we see that we have much more heteroskedacity in the residuals (figure 9a), but the addition of lagged response predictors has reduced the autocorrelation (Figure 9c and d), although significant autocorrelation still occurs.

Lets see if our new model (zinb3) is any better than our original full model by comparing AIC.

```
##      df      AIC
## zinb 23 16051.09
## zinb3 19 15252.83
```

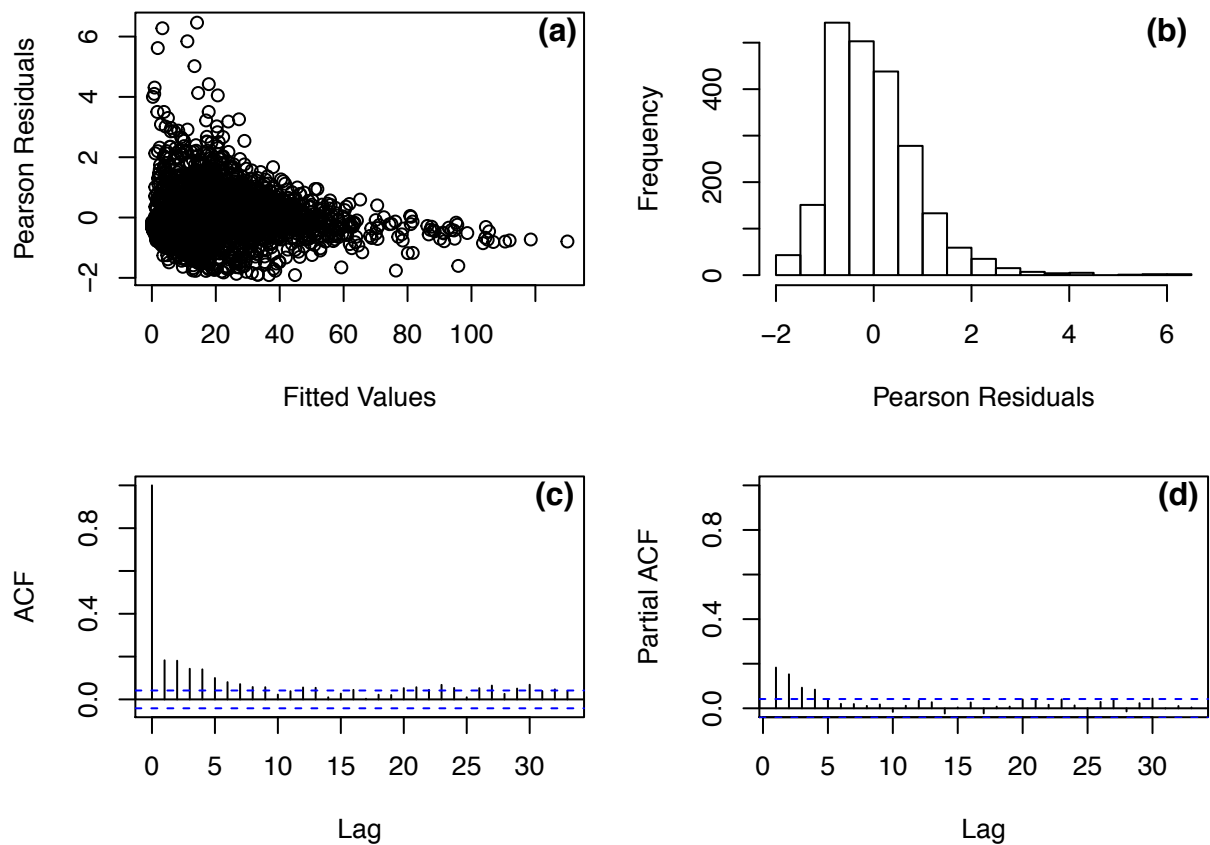


Figure 9: Model diagnostic plots. **a)** Pearson residuals versus fitted values, we should not see any clear pattern. **b)** histogram of pearson residuals, we should see residuals normally distributed around 0. **c)** Auto-correlation function of residuals and **d)** Partial autocorrelation function, vertical bars should be between the two horizontal blue lines if residuals are independent.

Looks like our new model has a better fit. We can use a sandwich estimator to generate more robust standard errors to account for both the heteroskedacity and autocorrelation in the residuals. We can then check that our predictors are still significant with better error estimates.

```
## Loading required package: sandwich

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## count_(Intercept)    1.1792e+00  4.5785e-02  25.7558 < 2.2e-16 ***
## count_AR1            1.5101e-02  1.1295e-03  13.3703 < 2.2e-16 ***
## count_AR2            5.8143e-03  1.1068e-03   5.2535 1.636e-07 ***
## count_AR3            3.8706e-03  1.0592e-03   3.6541 0.0002641 ***
## count_Discharge     -2.5177e-02  1.9862e-03 -12.6764 < 2.2e-16 ***
## count_I(Discharge^2) -1.3701e-03  1.4797e-04  -9.2591 < 2.2e-16 ***
## count_Temp.c         3.0790e-02  8.3930e-03   3.6686 0.0002497 ***
## count_Pelicans      -4.0667e-02  6.3976e-03  -6.3566 2.500e-10 ***
## count_Time          2.3673e-04  6.4682e-05   3.6599 0.0002583 ***
## zero_(Intercept)    -2.3898e+00  5.0493e-01  -4.7329 2.354e-06 ***
## zero_AR1            -1.6752e-01  2.7377e-02  -6.1191 1.111e-09 ***
## zero_Temp.c         1.2506e-01  4.7166e-02   2.6515 0.0080711 **
## zero_I(Temp.c^2)     5.2298e-02  1.4849e-02   3.5219 0.0004372 ***
## zero_Pelicans       1.3430e-01  3.0722e-02   4.3715 1.292e-05 ***
## zero_fWaterClarity2  9.5854e-01  4.1708e-01   2.2982 0.0216441 *
## zero_fWaterClarity3  1.9596e+00  4.3244e-01   4.5315 6.169e-06 ***
## zero_fImageClarity2 -5.5090e-01  2.3870e-01  -2.3079 0.0210961 *
## zero_Time           1.0673e-03  3.8793e-04   2.7512 0.0059867 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looks like all of the predictors remain significant. Let's compare the fitted model with the real data.

It appears the model does a decent job tracking the raw data (Figure 10 and 11) but over estimates counts when raw carp counts are high (Figure 10).

## References

- Crivelli, A. J. 1981. "The Biology of the Common Carp, *Cyprinus Carpio* L. in the Camargue, Southern France." *Journal of Fish Biology* 18 (3): 271–90.
- Crook, David A. 2004. "Movements Associated with Home-Range Establishment by Two Species of Lowland River Fish." *Canadian Journal of Fisheries and Aquatic Sciences* 61 (11): 2183–93. doi:[10.1139/f04-151](https://doi.org/10.1139/f04-151).
- McCrimmon, Hugh R. 1968. "Carp in Canada." *Bulletin of the Fisheries Research Board of Canada* 165: 1–93.
- Stuart, I. G., and M. J. Jones. 2006. "Movement of Common Carp, *Cyprinus Carpio*, in a Regulated Lowland Australian River: Implications for Management." *Fisheries Management and Ecology* 13 (4): 213–19. doi:[10.1111/j.1365-2400.2006.00495.x](https://doi.org/10.1111/j.1365-2400.2006.00495.x).
- Ver Hoef, Jay M., and John K. Jansen. 2007. "Spacetime Zero-Inflated Count Models of Harbor Seals." *Environmetrics* 18 (7): 697–712. doi:[10.1002/env.873](https://doi.org/10.1002/env.873).
- Zuur, Alain F., Elena N. Ieno, and Chris S. Elphick. 2010. "A Protocol for Data Exploration to Avoid Common Statistical Problems." *Methods in Ecology and Evolution* 1 (1): 3–14. doi:[10.1111/j.2041-210X.2009.00001.x](https://doi.org/10.1111/j.2041-210X.2009.00001.x).



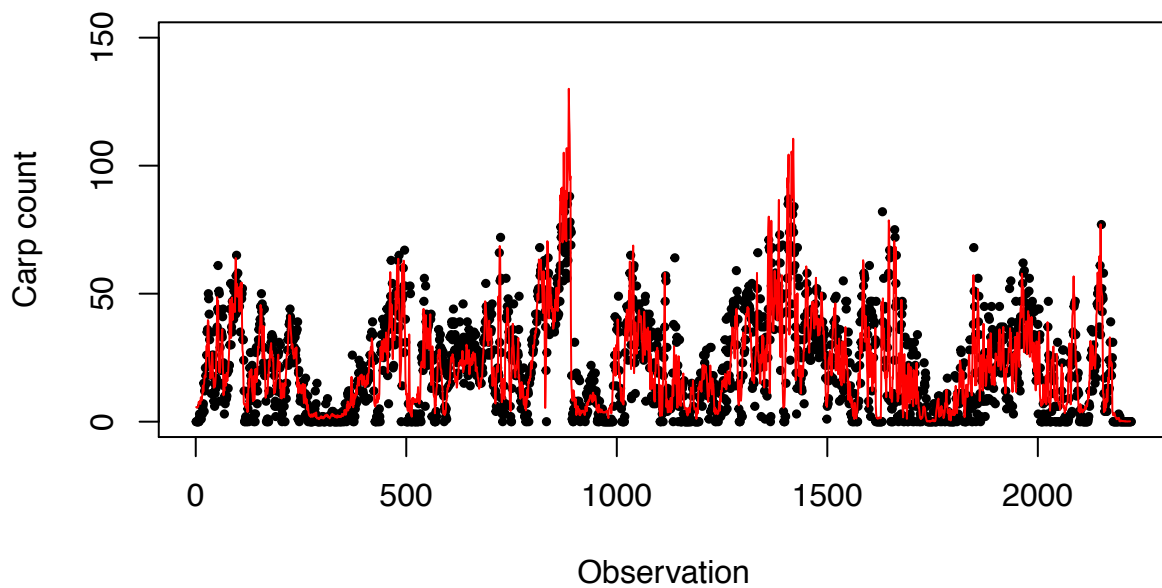


Figure 10: Carp counts (black circles) at each observation index overlaid by the fitted model zinb3 model (red line).

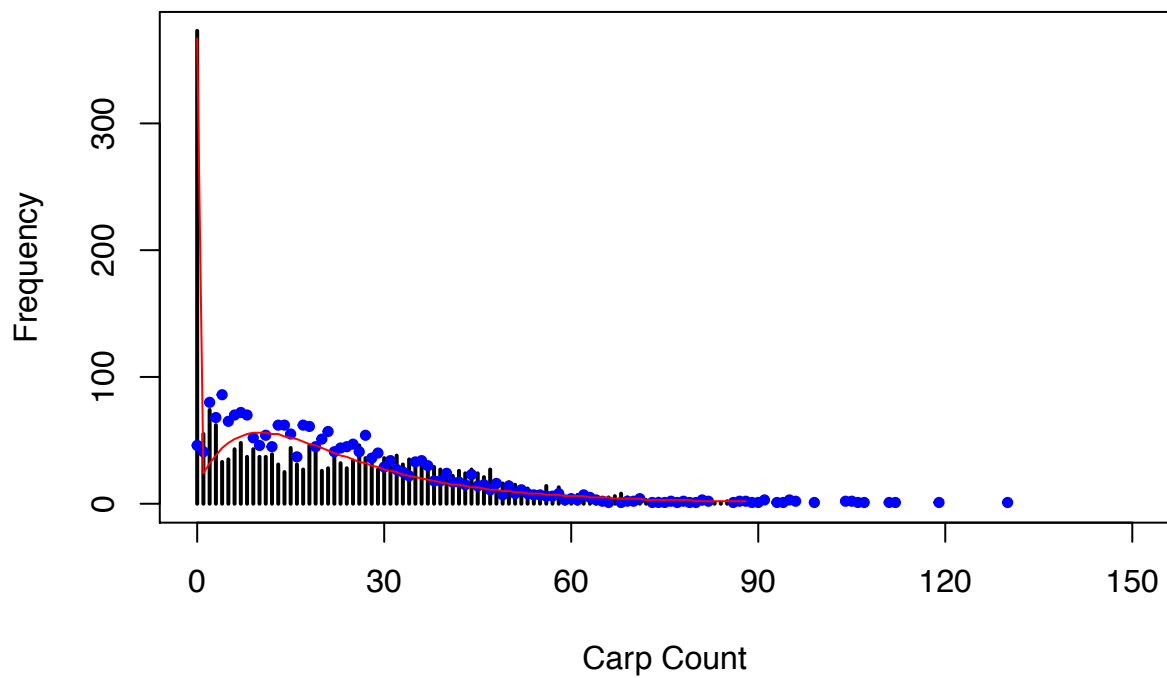


Figure 11: Original carp count histogram (black lines) with zinb3 modelled histogram (blue circles) and zero-inflated negative binomial probability curve overlaid (red line).