Supplement to Wilber et al. (2024) – Mar Ecol Prog Ser 727: 123–142 – https://doi.org/10.3354/meps14482

Supplementary Material 2

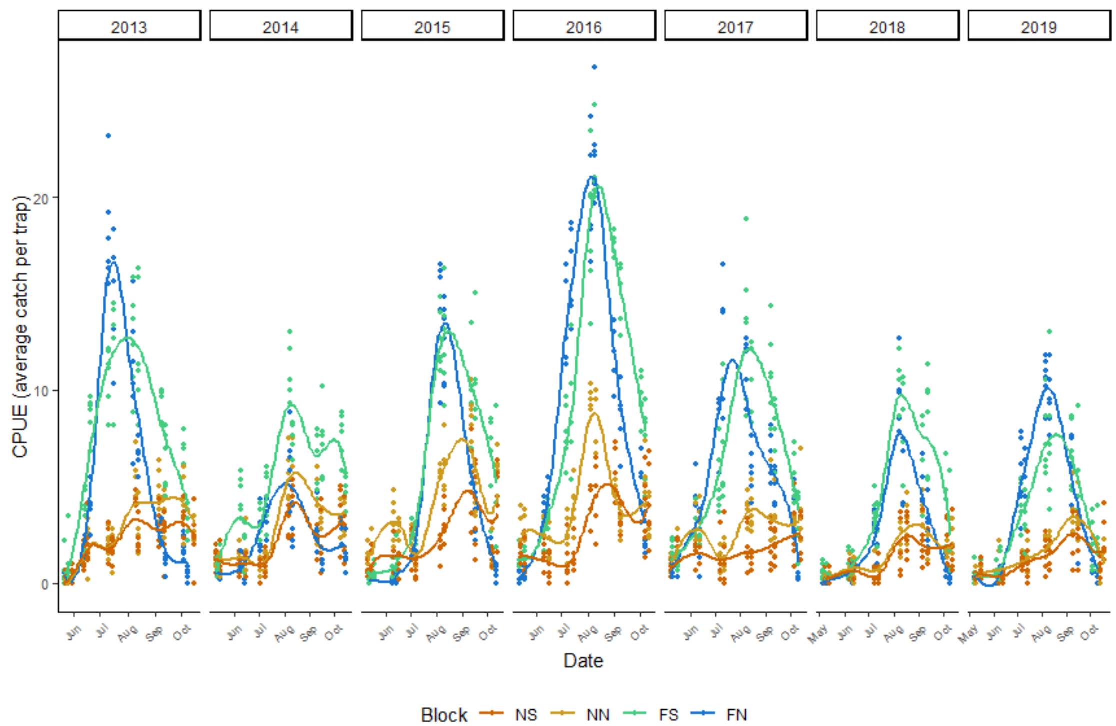
American lobster *Homarus americanus* responses to construction and operation of an offshore wind farm in Southern New England

```
   library(tidyverse)
   library(MASS)   #stepAIC
   library(lubridate)
   library(DHARMa) #diagnostics on GLMs
## This is DHARMa 0.4.3. For overview type '?DHARMa'. For recent changes, type news(p
ackage = 'DHARMa') Note: Syntax of plotResiduals has changed in 0.3.0, see ?plotResid
uals for details
  library(glmmTMB)
```

CPUE Catch Data and Model

```
#set working data frame
which.dat <- lobsters7.bySurvey %>% filter(!is.na(catch.trawl))%>%
    mutate(month.num = month(Date))
which.dat

## # A tibble: 2,001 x 23
##    TrawlID.lbr  Year Block Date       Month Trawl.num monthly.sample
##    <chr>       <int> <chr> <date>     <chr>     <int> <chr>
##  1 1163a        2013 FN    2013-05-22 May           7 A
##  2 1163b        2013 FN    2013-05-22 May           7 A
##  3 1164a        2013 FN    2013-05-22 May           8 A
##  4 1164b        2013 FN    2013-05-22 May           8 A
##  5 1166a        2013 FN    2013-05-22 May           9 A
##  6 1166b        2013 FN    2013-05-22 May           9 A
##  7 1167a        2013 FS    2013-05-22 May          10 A
##  8 1167b        2013 FS    2013-05-22 May          10 A
##  9 1168a        2013 FS    2013-05-22 May          11 A
## 10 1168b        2013 FS    2013-05-22 May          11 A
## # ... with 1,991 more rows, and 16 more variables: trawl.section <chr>,
## #   trap.count <int>, cpue <dbl>, Trawl.ID <int>, Trawl.numab <chr>,
## #   area <chr>, exclude <chr>, cpue.imp <dbl>, SampEvent <chr>, year.fac <fct>,
## #   avg.temp.C <dbl>, Period <chr>, catch.trawl <dbl>, catch.trawl.imp <dbl>,
## #   temp.ctr <dbl[,1]>, month.num <dbl>

#Show number of trawl arrays by block and year:
with(which.dat, table(Block, Year))

##      Year
## Block 2013 2014 2015 2016 2017 2018 2019
##    FN   72   72   72   72   72   72   72
##    FS   72   72   72   72   72   72   72
##    NN   72   72   72   72   68   70   67
##    NS   72   72   72   72   68   72   72

#Plot CPUE over time for every survey event.
ggplot(which.dat, aes(x=Date, y=catch.trawl/trap.count, colour=Block), facets=~year.f
```

```
ac) +
    geom_point(size=1) + facet_wrap(~year.fac, nrow=1, scales="free_x")+
    geom_smooth(se=FALSE, method="loess", span=0.5) +
    scale_colour_manual(breaks=c("NS","NN","FS","FN"),
    values=c("darkorange3","goldenrod3","seagreen3","dodgerblue3")) +
    labs(y="CPUE (average catch per trap)")+
    my.theme
```



Fit the catch model to 7 years of data, using stepAIC. First attempt with Poisson, then negative binomial, then zero-inflated negative binomial. Only showing the Poisson stepAIC iterations, others were similar.

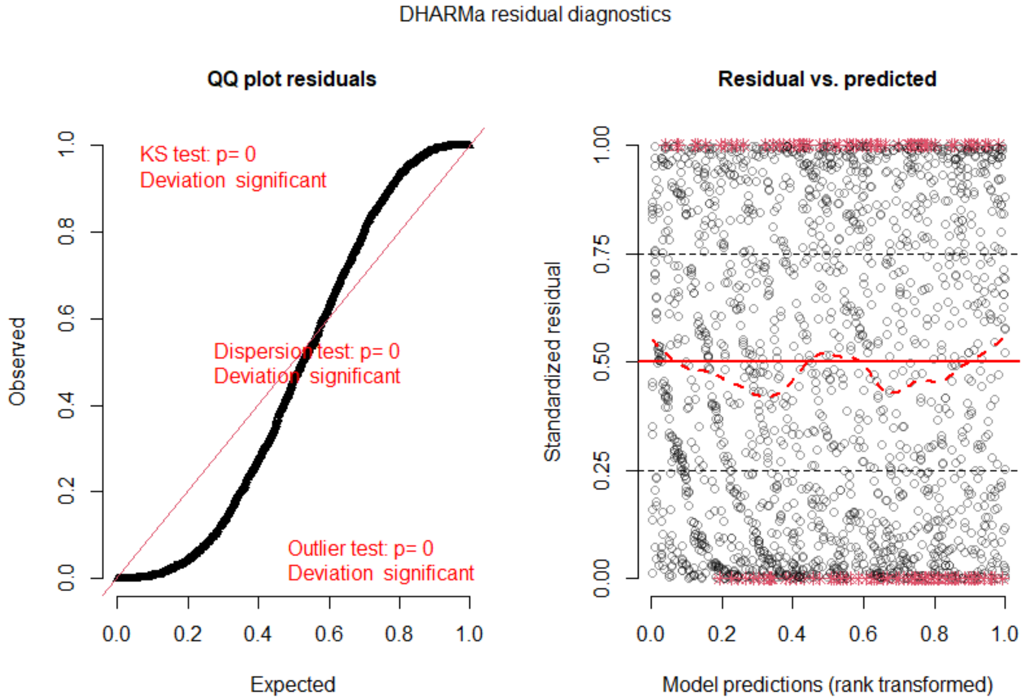
```
### Fit Model
which.dat <- lobsters7.bySurvey %>% filter(!is.na(catch.trawl)) %>%
    mutate(month.num = month(Date))
 ##use stepAIC
 #bounding models with the lowest = mean only; allows for temperature to dictate all
results.
 foo.null <- glm(catch.trawl ~ 1 + offset(log(trap.count)), family="poisson",
    data=which.dat)
 foo.stepaic <- stepAIC(foo.null, scope=list(upper=~0 + Block*year.fac +
    Month*Block + Month*year.fac + temp.ctr + I(temp.ctr^2)),
    direction="both")

## Start:  AIC=51939.11
## catch.trawl ~ 1 + offset(log(trap.count))
##
##                   Df Deviance   AIC
## + Month            5    26941 35558
## + Block            3    35885 44498
```

```
## + I(temp.ctr^2)  1    37808 46416
## + year.fac       6    38711 47329
## + temp.ctr       1    40385 48994
## <none>                43333 51939
##
## Step:  AIC=35557.52
## catch.trawl ~ Month + offset(log(trap.count))
##
##                 Df Deviance   AIC
## + Block          3    19401 28024
## + year.fac       6    22284 30913
## + temp.ctr       1    25808 34426
## + I(temp.ctr^2)  1    26674 35292
## <none>                26941 35558
## - Month          5    43333 51939
##
## Step:  AIC=28023.86
## catch.trawl ~ Month + Block + offset(log(trap.count))
##
##                 Df Deviance   AIC
## + year.fac       6    14677 23311
## + Block:Month   15    15445 24098
## + temp.ctr       1    19295 27919
## + I(temp.ctr^2)  1    19399 28024
## <none>                19401 28024
## - Block          3    26941 35558
## - Month          5    35885 44498
##
## Step:  AIC=23311.4
## catch.trawl ~ Month + Block + year.fac + offset(log(trap.count))
##
##                  Df Deviance   AIC
## + Block:Month    15    10734 19398
## + year.fac:Month 30    11952 20647
## + Block:year.fac 18    13579 22249
## + temp.ctr        1    14576 23212
## + I(temp.ctr^2)   1    14664 23300
## <none>                 14677 23311
## - year.fac        6    19401 28024
## - Block           3    22284 30913
## - Month           5    31199 39823
##
## Step:  AIC=19397.99
## catch.trawl ~ Month + Block + year.fac + Month:Block + offset(log(trap.count))
##
##                  Df Deviance   AIC
## + year.fac:Month 30   8014.3 16739
## + Block:year.fac 18   9641.3 18342
## + temp.ctr        1  10468.5 19135
## + I(temp.ctr^2)   1  10718.7 19385
## <none>               10733.5 19398
## - Month:Block    15  14677.0 23311
## - year.fac        6  15445.3 24098
##
## Step:  AIC=16738.74
```

```
## catch.trawl ~ Month + Block + year.fac + Month:Block + Month:year.fac +
##     offset(log(trap.count))
##
##                  Df Deviance   AIC
## + Block:year.fac 18   7143.6 15904
## + I(temp.ctr^2)   1   8003.3 16730
## + temp.ctr        1   8003.8 16730
## <none>                8014.3 16739
## - Month:year.fac 30  10733.5 19398
## - Month:Block    15  11952.3 20647
##
## Step:  AIC=15904.09
## catch.trawl ~ Month + Block + year.fac + Month:Block + Month:year.fac +
##     Block:year.fac + offset(log(trap.count))
##
##                  Df Deviance   AIC
## + temp.ctr        1   7076.2 15839
## <none>                7143.6 15904
## + I(temp.ctr^2)   1   7142.7 15905
## - Block:year.fac 18   8014.3 16739
## - Month:year.fac 30   9641.3 18342
## - Month:Block    15  10847.3 19578
##
## Step:  AIC=15838.6
## catch.trawl ~ Month + Block + year.fac + temp.ctr + Month:Block +
##     Month:year.fac + Block:year.fac + offset(log(trap.count))
##
##                  Df Deviance   AIC
## + I(temp.ctr^2)   1   7035.7 15800
## <none>                7076.2 15839
## - temp.ctr        1   7143.6 15904
## - Block:year.fac 18   8003.8 16730
## - Month:year.fac 30   9285.3 17988
## - Month:Block    15  10828.6 19561
##
## Step:  AIC=15800.15
## catch.trawl ~ Month + Block + year.fac + temp.ctr + I(temp.ctr^2) +
##     Month:Block + Month:year.fac + Block:year.fac + offset(log(trap.count))
##
##                  Df Deviance   AIC
## <none>                7035.7 15800
## - I(temp.ctr^2)   1   7076.2 15839
## - temp.ctr        1   7142.7 15905
## - Block:year.fac 18   7943.2 16672
## - Month:year.fac 30   9267.2 17972
## - Month:Block    15  10824.7 19559
```

Some model diagnostics, using DHARMa package:

```
simulationOutput <- simulateResiduals(fittedModel = foo.stepaic, plot = T, n=500)
```

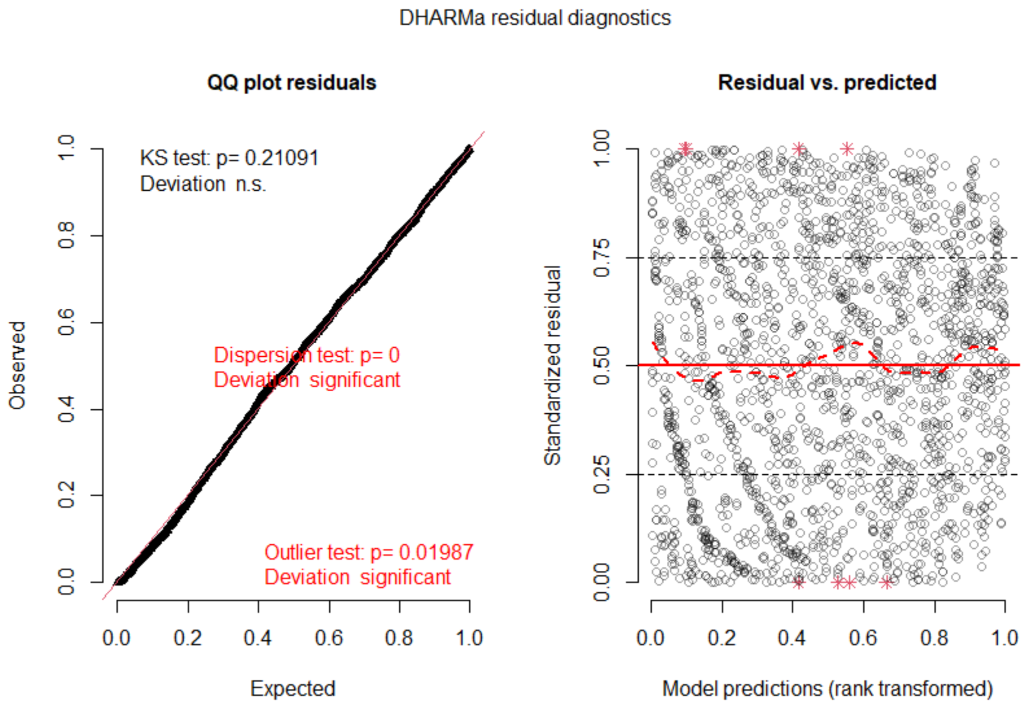



```
testDispersion(foo.stepaic, plot=F)
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 3.7498, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

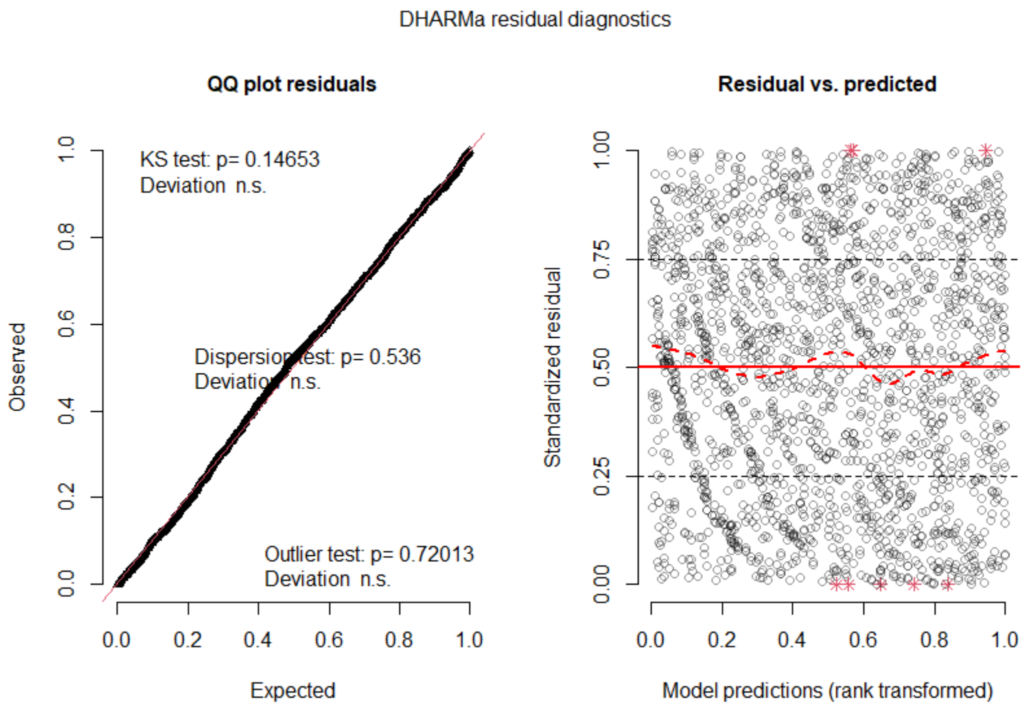
Residual plots indicate some problems. Tried negative binomial, and then zero-inflated negative binomial. Compare fits

```
#try negative binomial
foo.null <- glm.nb(catch.trawl ~ 1 + offset(log(trap.count)),
   data=which.dat)
foo.nb <- stepAIC(foo.null, scope=list(upper=~0 + Block*year.fac +
   Month*Block + Month*year.fac + temp.ctr + I(temp.ctr^2)),
   direction="both")

##<truncate results>
## Step:  AIC=13469.07
## catch.trawl ~ Month + year.fac + Block + temp.ctr + I(temp.ctr^2) +
##     Month:Block + Month:year.fac + year.fac:Block + offset(log(trap.count))
##
##                   Df   AIC
## <none>               13469
## - I(temp.ctr^2)    1 13475
## - temp.ctr         1 13491
## - year.fac:Block 18 13677
```

```
## - Month:year.fac 30 13913
## - Month:Block     15 14262

 simulationOutputNB <- simulateResiduals(fittedModel = foo.nb, plot = T, n=500)
```

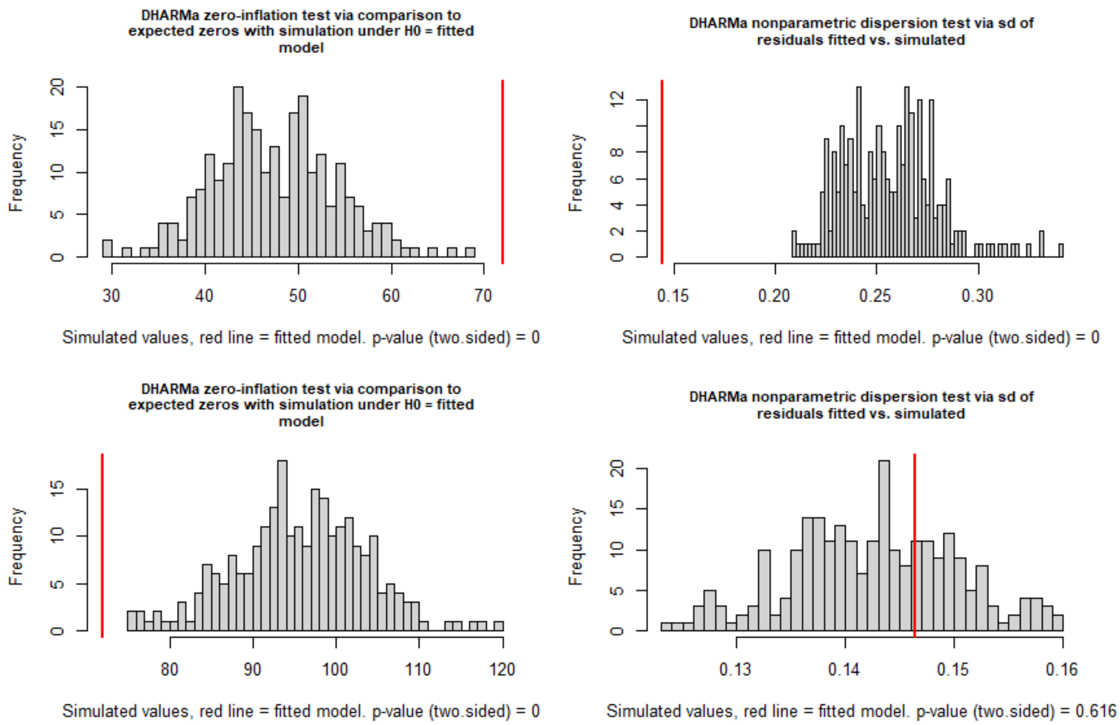



```
 #residuals uniform; appear to have problem (potentially) with outliers and dispersio
n
 testOutliers(simulationOutputNB, type="bootstrap", plot=F)

##
##  DHARMa bootstrapped outlier test
##
## data:  simulationOutputNB
## outliers at both margin(s) = 10, observations = 2001, p-value = 0.84
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0.001499250 0.006996502
## sample estimates:
## outlier frequency (expected: 0.00438780609695152 )
##                                         0.004997501

  #outliers ok.

#Try Zero-inflation (from glmmTMB)
#Most zeros in May and June, model zero-inflation by month
foo.znb <- glmmTMB(catch.trawl ~ Block*year.fac*Month - Block:year.fac:Month+
    temp.ctr + I(temp.ctr^2) + offset(log(trap.count)),
    ziformula = ~Month, data=which.dat, family=nbinom1,
    control=glmmTMBControl(optCtrl=list(iter.max=500, eval.max=500) ))
foo.znb.res <- simulateResiduals(foo.znb, plot=T, n=500)
```



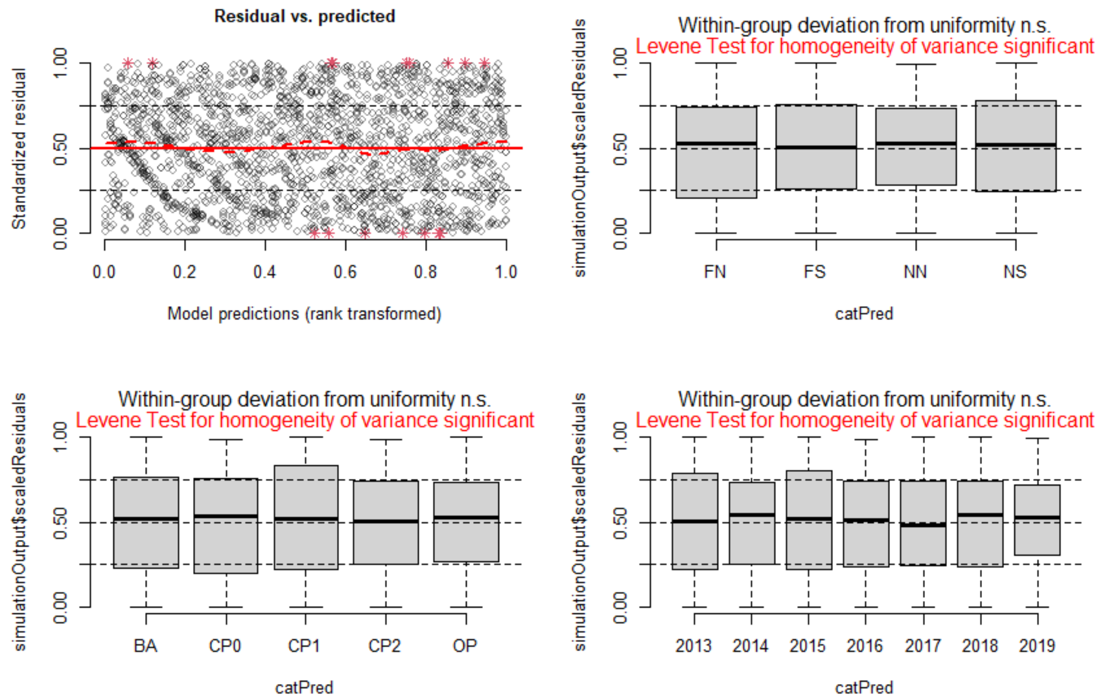

```
#compare results for negative binomial, without/with zero-inflation
 par(mfrow=c(2,2))
 testZeroInflation(foo.nb)

##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 1.5036, p-value < 2.2e-16
## alternative hypothesis: two.sided

 testDispersion(foo.nb) #now slightly overdispersed

##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.56295, p-value < 2.2e-16
## alternative hypothesis: two.sided

 testZeroInflation(foo.znb) #under with ratoObsSim=0.744, p=2.2e-16

##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 0.7486, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

```r
testDispersion(foo.znb)
```

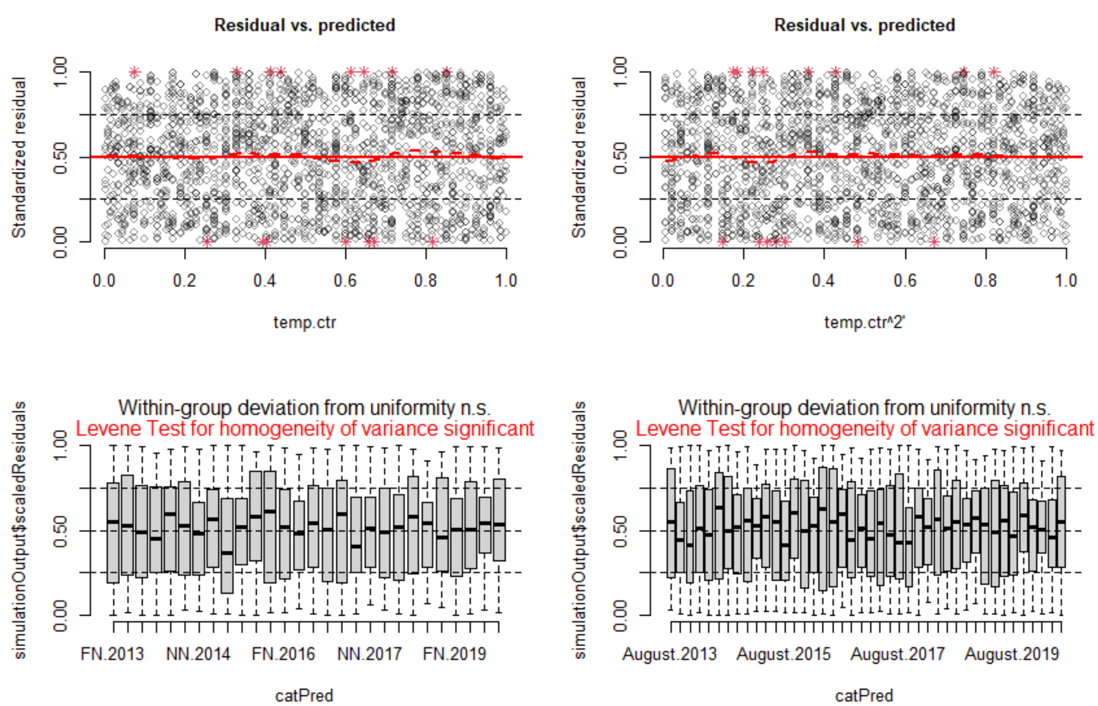



```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 1.0272, p-value = 0.616
## alternative hypothesis: two.sided
```

```r
#zero-inflated negative binomial model looks good.
#check for model misfit. plot residuals against all predictors
par(mfrow=c(2,2))
 plotResiduals(foo.znb)
 plotResiduals(foo.znb, form=factor(which.dat$Block))
 plotResiduals(foo.znb, form=factor(which.dat$Period))
 plotResiduals(foo.znb, form=which.dat$year.fac)
```




```
par(mfrow=c(2,2));
 plotResiduals(foo.znb, form=which.dat$temp.ctr, xlab="temp.ctr")
 plotResiduals(foo.znb, form=I(which.dat$temp.ctr^2), xlab="temp.ctr^2'")
 plotResiduals(foo.znb, form=interaction(which.dat$Block,which.dat$year.fac))
 plotResiduals(foo.znb, form=interaction(which.dat$Month,which.dat$year.fac))
```



```r
#tends to be some heteroscedasticity via Levene's test (perhaps due to large sample s
ize, detecting very small differences).  However, plots look good.
 #set this as final model and show result.
 lobsters7.FishRes.cpue.glm <- glmmTMB(catch.trawl ~ Block*year.fac*Month -
    Block:year.fac:Month+   temp.ctr + I(temp.ctr^2) + offset(log(trap.count)),
    ziformula = ~Month, data=lobsters7.bySurvey %>% filter(!is.na(catch.trawl)) %>%
    mutate(month.num = month(Date)), family=nbinom1,
    control=glmmTMBControl(optCtrl=list(iter.max=500, eval.max=500) ))

summary(lobsters7.FishRes.cpue.glm)
```

```
##  Family: nbinom1  ( log )
## Formula:
## catch.trawl ~ Block * year.fac * Month - Block:year.fac:Month +
##     temp.ctr + I(temp.ctr^2) + offset(log(trap.count))
## Zero inflation:               ~Month
## Data:
## lobsters7.bySurvey %>% filter(!is.na(catch.trawl)) %>% mutate(month.num = month(Da
te))
##
##      AIC      BIC   logLik deviance df.resid
##  13346.6  13833.9  -6586.3  13172.6     1914
##
##
## Dispersion parameter for nbinom1 family (): 2.64
##
## Conditional model:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                2.088992   0.063746   32.77  < 2e-16 ***
## BlockFS                    0.393845   0.065885    5.98 2.26e-09 ***
## BlockNN                   -1.222363   0.098282  -12.44  < 2e-16 ***
## BlockNS                   -1.316421   0.094052  -14.00  < 2e-16 ***
## year.fac2014              -0.521559   0.085132   -6.13 8.98e-10 ***
## year.fac2015               0.117372   0.078102    1.50 0.132888
## year.fac2016               0.902017   0.066300   13.61  < 2e-16 ***
## year.fac2017               0.322698   0.091542    3.53 0.000423 ***
## year.fac2018              -0.282282   0.083921   -3.36 0.000769 ***
## year.fac2019               0.199791   0.095254    2.10 0.035953 *
## MonthJuly                  0.673827   0.082460    8.17 3.05e-16 ***
## MonthJune                 -0.413588   0.159002   -2.60 0.009291 **
## MonthMay                  -2.169283   0.374790   -5.79 7.12e-09 ***
## MonthOctober              -2.128308   0.120972  -17.59  < 2e-16 ***
## MonthSeptember            -1.203257   0.084491  -14.24  < 2e-16 ***
## temp.ctr                   0.278425   0.047707    5.84 5.34e-09 ***
## I(temp.ctr^2)             -0.025165   0.007202   -3.49 0.000475 ***
## BlockFS:year.fac2014       0.392640   0.090415    4.34 1.41e-05 ***
## BlockNN:year.fac2014       0.704027   0.108636    6.48 9.14e-11 ***
## BlockNS:year.fac2014       0.549973   0.116418    4.72 2.31e-06 ***
## BlockFS:year.fac2015      -0.037326   0.083184   -0.45 0.653638
## BlockNN:year.fac2015       0.546853   0.102556    5.33 9.70e-08 ***
## BlockNS:year.fac2015       0.263030   0.113038    2.33 0.019970 *
## BlockFS:year.fac2016      -0.355851   0.068575   -5.19 2.11e-07 ***
## BlockNN:year.fac2016      -0.195892   0.091249   -2.15 0.031809 *
## BlockNS:year.fac2016      -0.465067   0.099048   -4.70 2.66e-06 ***
## BlockFS:year.fac2017      -0.194913   0.077828   -2.50 0.012266 *
```

```
## BlockNN:year.fac2017        -0.180817   0.103537   -1.75 0.080740 .
## BlockNS:year.fac2017        -0.459097   0.113371   -4.05 5.13e-05 ***
## BlockFS:year.fac2018         0.101130   0.091223    1.11 0.267601
## BlockNN:year.fac2018         0.207140   0.117281    1.77 0.077365 .
## BlockNS:year.fac2018         0.140344   0.126715    1.11 0.268053
## BlockFS:year.fac2019        -0.390795   0.088165   -4.43 9.31e-06 ***
## BlockNN:year.fac2019        -0.266948   0.116930   -2.28 0.022432 *
## BlockNS:year.fac2019        -0.252929   0.120830   -2.09 0.036326 *
## BlockFS:MonthJuly           -0.358773   0.061139   -5.87 4.41e-09 ***
## BlockNN:MonthJuly           -0.730557   0.090796   -8.05 8.54e-16 ***
## BlockNS:MonthJuly           -0.524637   0.102133   -5.14 2.79e-07 ***
## BlockFS:MonthJune            0.227821   0.090284    2.52 0.011623 *
## BlockNN:MonthJune            0.876631   0.103792    8.45  < 2e-16 ***
## BlockNS:MonthJune            0.955119   0.116161    8.22  < 2e-16 ***
## BlockFS:MonthMay             0.565878   0.139223    4.06 4.81e-05 ***
## BlockNN:MonthMay             1.820036   0.147587   12.33  < 2e-16 ***
## BlockNS:MonthMay             2.014377   0.148241   13.59  < 2e-16 ***
## BlockFS:MonthOctober         0.856006   0.089041    9.61  < 2e-16 ***
## BlockNN:MonthOctober         1.723198   0.108050   15.95  < 2e-16 ***
## BlockNS:MonthOctober         1.911092   0.102915   18.57  < 2e-16 ***
## BlockFS:MonthSeptember       0.516511   0.064208    8.04 8.67e-16 ***
## BlockNN:MonthSeptember       0.795679   0.097478    8.16 3.28e-16 ***
## BlockNS:MonthSeptember       0.885168   0.085823   10.31  < 2e-16 ***
## year.fac2014:MonthJuly      -0.721455   0.109814   -6.57 5.04e-11 ***
## year.fac2015:MonthJuly      -1.486015   0.106480  -13.96  < 2e-16 ***
## year.fac2016:MonthJuly      -0.517796   0.096461   -5.37 7.96e-08 ***
## year.fac2017:MonthJuly      -0.325063   0.095097   -3.42 0.000630 ***
## year.fac2018:MonthJuly      -1.048452   0.113121   -9.27  < 2e-16 ***
## year.fac2019:MonthJuly      -0.696715   0.100166   -6.96 3.51e-12 ***
## year.fac2014:MonthJune      -0.360016   0.130269   -2.76 0.005716 **
## year.fac2015:MonthJune      -1.024520   0.126610   -8.09 5.87e-16 ***
## year.fac2016:MonthJune      -0.325360   0.134882   -2.41 0.015858 *
## year.fac2017:MonthJune      -0.166079   0.113913   -1.46 0.144854
## year.fac2018:MonthJune      -1.081816   0.146094   -7.40 1.31e-13 ***
## year.fac2019:MonthJune      -1.394275   0.159257   -8.75  < 2e-16 ***
## year.fac2014:MonthMay        1.541159   0.214468    7.19 6.67e-13 ***
## year.fac2015:MonthMay        2.247503   0.432744    5.19 2.06e-07 ***
## year.fac2016:MonthMay        0.892313   0.196384    4.54 5.53e-06 ***
## year.fac2017:MonthMay        0.784697   0.238771    3.29 0.001015 **
## year.fac2018:MonthMay        0.711252   0.268805    2.65 0.008146 **
## year.fac2019:MonthMay        0.516184   0.225327    2.29 0.021974 *
## year.fac2014:MonthOctober   -0.015515   0.162870   -0.10 0.924108
## year.fac2015:MonthOctober   -0.447011   0.123625   -3.62 0.000299 ***
## year.fac2016:MonthOctober   -0.399095   0.134829   -2.96 0.003076 **
## year.fac2017:MonthOctober   -0.432203   0.146422   -2.95 0.003160 **
## year.fac2018:MonthOctober   -0.592731   0.162300   -3.65 0.000260 ***
## year.fac2019:MonthOctober   -0.771349   0.150860   -5.11 3.17e-07 ***
## year.fac2014:MonthSeptember -0.265078   0.125685   -2.11 0.034939 *
## year.fac2015:MonthSeptember  0.101207   0.094410    1.07 0.283719
## year.fac2016:MonthSeptember -0.138478   0.108777   -1.27 0.203003
## year.fac2017:MonthSeptember -0.083563   0.123963   -0.67 0.500250
## year.fac2018:MonthSeptember -0.161343   0.134907   -1.20 0.231713
## year.fac2019:MonthSeptember  0.030105   0.115707    0.26 0.794722
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Zero-inflation model:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -20.85008 1838.09585  -0.011    0.991
## MonthJuly        -0.04644 3202.02681   0.000    1.000
## MonthJune        -0.23297 3540.84203   0.000    1.000
## MonthMay         -0.91260 4426.43479   0.000    1.000
## MonthOctober     15.66708 1838.09610   0.009    0.993
## MonthSeptember   -1.60997 4503.40260   0.000    1.000
```

##Ovigery Models First show the data…

```
#We use both compromised and uncompromised traps for these models.
#But filter for:
# Mortality = FALSE
# Females (sex code = 2)
# Size >= 79mm
#NOTE:
# Late Stage/Cohort 1 ("collected in May and June … carrying eggs that were likely
#  to hatch in the coming weeks") - egg code = 2, 4 or 5
# Early Stage/Cohort 2 ("September and October … newly extruded egg masses") -
#  egg code = 2, 4, or 5
# Spent/Cohort 3 ("June, July, August … recently hatched (spent)") - egg code = 7

#set working data frame
which.dat <- lobsters7.ovig.bySurvey %>% filter(Cohort1 | Cohort2) %>%
    mutate(Ovig.f = factor(Ovig), EggStage=
    case_when(Cohort1 ~ "Late", Cohort2~"Early")) %>%
    rename(Yr=year.fac, Pd=Period, Mo=Month, Bl=Block)
which.dat

## # A tibble: 8,053 x 33
##     Year Mo    Bl    Fisher.ID Date       Trawl.. Trap.. Auto..  Size   Sex
##    <int> <chr> <chr> <chr>     <date>       <int>  <int>  <int> <dbl> <int>
##  1  2013 May   NS    BM1       2013-05-24       1      4  32316  83.4     2
##  2  2013 May   NS    BM1       2013-05-31       1     10  51905  86.2     2
##  3  2013 June  NS    BM1       2013-06-13       1     12  32452  86.1     2
##  4  2013 June  NS    BM1       2013-06-13       1     12  32455  96.9     2
##  5  2013 June  NS    BM1       2013-06-13       1     12  32456  84.2     2
##  6  2013 June  NS    BM1       2013-06-13       1     11  32449  82.1     2
##  7  2013 June  NS    BM1       2013-06-13       1     11  32451  82       2
##  8  2013 June  NS    BM1       2013-06-13       1      9  32445  86.6     2
##  9  2013 June  NS    BM1       2013-06-13       1      9  32446 101.      2
## 10  2013 June  NS    BM1       2013-06-13       1      8  32444 109.      2
## # ... with 8,043 more rows, and 23 more variables: Eggs <int>,
## #   Shell.Hardness <int>, Cull <int>, Pathology <int>, Shell.Disease <int>,
## #   Mortality <lgl>, Cement <int>, Ovary <lgl>, Flag_Compromised <lgl>,
## #   Flag_Lobster <lgl>, block <fct>, Yr <fct>, month.num <dbl>,
## #   avg.temp.C <dbl>, Pd <chr>, Cohort1 <lgl>, Cohort2 <lgl>, Cohort3 <lgl>,
## #   Ovig <dbl>, Spent <dbl>, temp.ctr <dbl[,1]>, Ovig.f <fct>, EggStage <chr>

#Counts of ovigerous/not ovigerous lobsters by Month and year for early and late-stag
e
with(which.dat, table(Bl, Yr, Ovig))
```

```
## , , Ovig = 0
##
##      Yr
## Bl    2013 2014 2015 2016 2017 2018 2019
##    FN  248  108  114  317  211   68  113
##    FS  523  309  221  478  302  178  197
##    NN  125  123  172  202  188   80   83
##    NS  154   83  106  115   91   51   55
##
## , , Ovig = 1
##
##      Yr
## Bl    2013 2014 2015 2016 2017 2018 2019
##    FN   26   43   54  108   94   26   43
##    FS  174  238  278  319  255  186   71
##    NN  101  105  145  144   88   69   44
##    NS   59   67   81   68   68   43   41
```

Fit single model for Late and Early Stage ovigerous females, using stepAIC

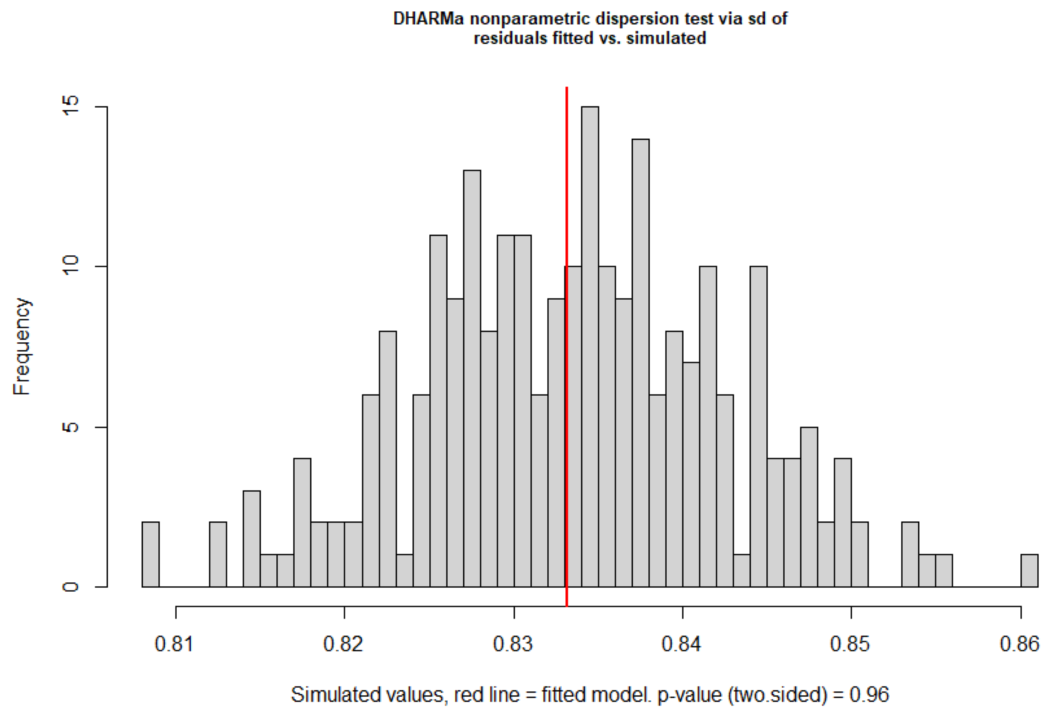
```
##use stepAIC
foo.null <- glm(Ovig.f ~ 1, family=binomial(link="logit"), data=which.dat)
foo.stepaic <- stepAIC(foo.null, scope=list(upper=~0 + Bl*Yr*Mo+
   EggStage*Bl + EggStage*Yr + temp.ctr + I(temp.ctr^2)), direction="both")
```

```
## Start:  AIC=10675.48
## Ovig.f ~ 1
##
##                  Df Deviance     AIC
## + Mo              3   9909.6  9917.6
## + I(temp.ctr^2)   1  10236.8 10240.8
## + Yr              6  10485.9 10499.9
## + Bl              3  10532.1 10540.1
## + EggStage        1  10665.2 10669.2
## <none>               10673.5 10675.5
## + temp.ctr        1  10672.6 10676.6
##
## Step:  AIC=9917.58
## Ovig.f ~ Mo
##
##                  Df Deviance     AIC
## + Yr              6   9744.7  9764.7
## + Bl              3   9821.9  9835.9
## + I(temp.ctr^2)   1   9894.3  9904.3
## + temp.ctr        1   9904.1  9914.1
## <none>                9909.6  9917.6
## - Mo              3  10673.5 10675.5
##
## Step:  AIC=9764.74
## Ovig.f ~ Mo + Yr
##
##                  Df Deviance     AIC
## + Yr:Mo          18   9562.3  9618.3
## + Bl              3   9661.8  9687.8
## <none>                9744.7  9764.7
```

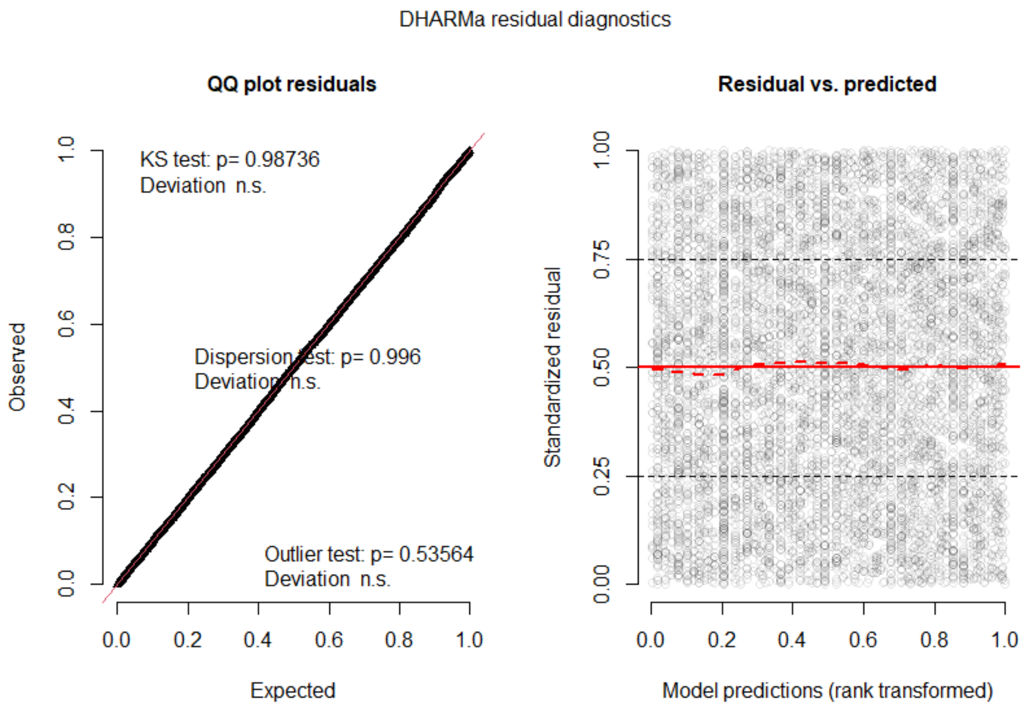
```
## + temp.ctr       1   9744.6  9766.6
## + I(temp.ctr^2)  1   9744.7  9766.7
## - Yr             6   9909.6  9917.6
## - Mo             3  10485.9 10499.9
##
## Step:  AIC=9618.31
## Ovig.f ~ Mo + Yr + Mo:Yr
##
##                 Df Deviance    AIC
## + Bl             3   9489.8 9551.8
## + I(temp.ctr^2)  1   9538.4 9596.4
## <none>               9562.3 9618.3
## + temp.ctr       1   9562.1 9620.1
## - Mo:Yr         18   9744.7 9764.7
##
## Step:  AIC=9551.75
## Ovig.f ~ Mo + Yr + Bl + Mo:Yr
##
##                 Df Deviance    AIC
## + Bl:Mo          9   9372.6 9452.6
## + Bl:Yr         18   9387.3 9485.3
## + I(temp.ctr^2)  1   9453.0 9517.0
## + temp.ctr       1   9469.3 9533.3
## <none>               9489.8 9551.8
## - Bl             3   9562.3 9618.3
## - Mo:Yr         18   9661.8 9687.8
##
## Step:  AIC=9452.63
## Ovig.f ~ Mo + Yr + Bl + Mo:Yr + Mo:Bl
##
##                 Df Deviance    AIC
## + Bl:Yr         18   9280.9 9396.9
## + temp.ctr       1   9361.4 9443.4
## <none>               9372.6 9452.6
## + I(temp.ctr^2)  1   9370.8 9452.8
## - Mo:Bl          9   9489.8 9551.8
## - Mo:Yr         18   9524.3 9568.3
##
## Step:  AIC=9396.91
## Ovig.f ~ Mo + Yr + Bl + Mo:Yr + Mo:Bl + Yr:Bl
##
##                 Df Deviance    AIC
## <none>               9280.9 9396.9
## + temp.ctr       1   9280.5 9398.5
## + I(temp.ctr^2)  1   9280.9 9398.9
## + Bl:Yr:Mo      54   9181.3 9405.3
## - Yr:Bl         18   9372.6 9452.6
## - Mo:Bl          9   9387.3 9485.3
## - Mo:Yr         18   9437.0 9517.0
```

Check residuals plots …

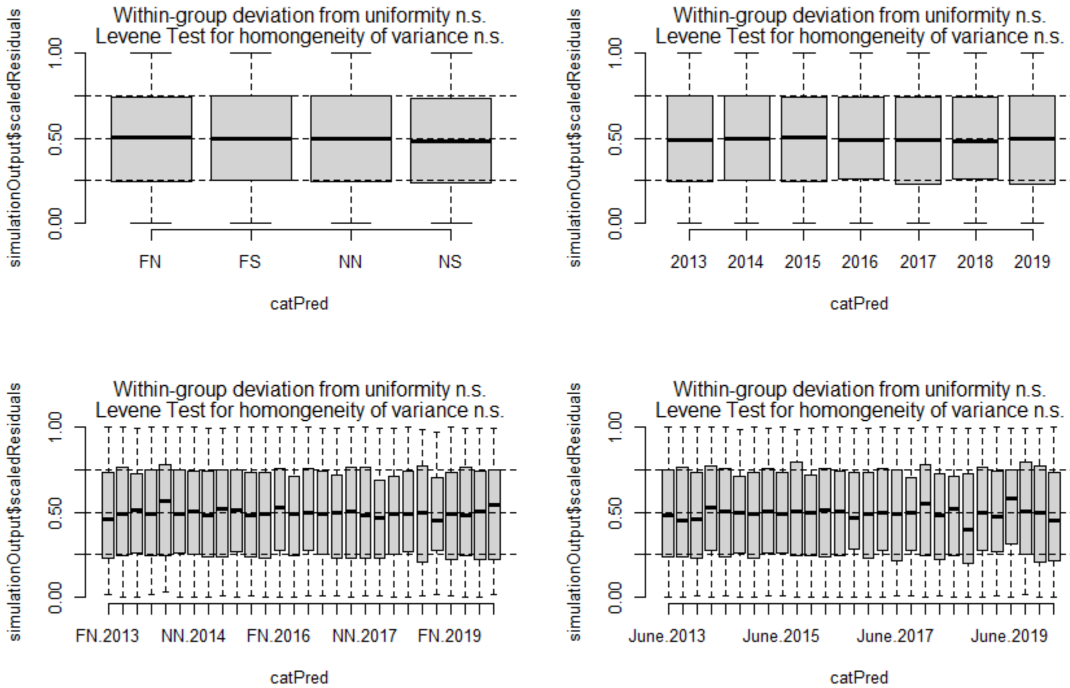
```
#check with DHARMa
 testDispersion(foo.stepaic)
```



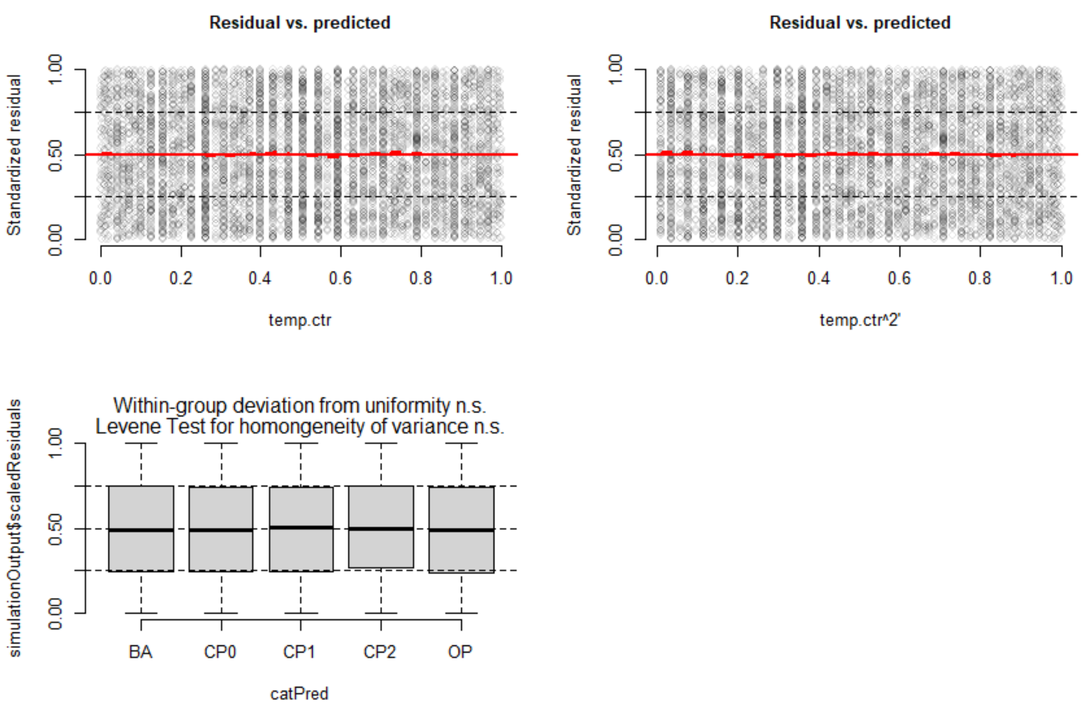

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.99968, p-value = 0.96
## alternative hypothesis: two.sided

 simulationOutput <- simulateResiduals(fittedModel = foo.stepaic, plot = T, n=500)
```




```
#check for model misfit. plot residuals against all predictors (both in and out of mo
del)
par(mfrow=c(2,2))
 plotResiduals(foo.stepaic, form=factor(which.dat$Bl))
 plotResiduals(foo.stepaic, form=which.dat$Yr)
 plotResiduals(foo.stepaic, form=interaction(which.dat$Bl,which.dat$Yr))
 plotResiduals(foo.stepaic, form=interaction(which.dat$Mo,which.dat$Yr))
```




```
par(mfrow=c(2,2));
 plotResiduals(foo.stepaic, form=which.dat$temp.ctr, xlab="temp.ctr")
 plotResiduals(foo.stepaic, form=I(which.dat$temp.ctr^2), xlab="temp.ctr^2'")
 plotResiduals(foo.stepaic, form=factor(which.dat$Pd))
```



Finalize model, and show results:

```
lobsters7.FishRes.Ovig.glm <- glm(formula = Ovig.f ~ Month + year.fac +
 Block + Month:year.fac + Month:Block + year.fac:Block,
 family = binomial(link = "logit"), data = lobsters7.ovig.bySurvey %>%
    filter(Cohort1 | Cohort2) %>%   mutate(Ovig.f = factor(Ovig)))

summary(lobsters7.FishRes.Ovig.glm)

##
## Call:
## glm(formula = Ovig.f ~ Month + year.fac + Block + Month:year.fac +
##     Month:Block + year.fac:Block, family = binomial(link = "logit"),
##     data = lobsters7.ovig.bySurvey %>% filter(Cohort1 | Cohort2) %>%
##         mutate(Ovig.f = factor(Ovig)))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3392  -0.8777  -0.6111   0.9314   2.4982
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -2.245481   0.220783 -10.171  < 2e-16 ***
## MonthMay                    1.859987   0.406026   4.581 4.63e-06 ***
## MonthOctober                0.247201   0.249212   0.992 0.321231
## MonthSeptember             -0.829881   0.242552  -3.421 0.000623 ***
## year.fac2014                1.874615   0.310494   6.038 1.56e-09 ***
## year.fac2015                2.106105   0.331679   6.350 2.16e-10 ***
## year.fac2016                1.156180   0.255981   4.517 6.28e-06 ***
## year.fac2017                0.961194   0.266897   3.601 0.000317 ***
## year.fac2018                2.009993   0.350489   5.735 9.76e-09 ***
## year.fac2019                1.437847   0.375023   3.834 0.000126 ***
## BlockFS                     1.004966   0.243453   4.128 3.66e-05 ***
## BlockNN                     1.321785   0.283017   4.670 3.01e-06 ***
## BlockNS                     1.113088   0.287813   3.867 0.000110 ***
## MonthMay:year.fac2014      -0.711097   0.396879  -1.792 0.073177 .
## MonthOctober:year.fac2014  -1.271852   0.234445  -5.425 5.80e-08 ***
## MonthSeptember:year.fac2014 -0.748175   0.249329  -3.001 0.002693 **
## MonthMay:year.fac2015      -0.453892   0.424474  -1.069 0.284933
## MonthOctober:year.fac2015  -0.531477   0.257984  -2.060 0.039387 *
## MonthSeptember:year.fac2015 -0.059118   0.257042  -0.230 0.818096
## MonthMay:year.fac2016      -0.044399   0.376005  -0.118 0.906003
## MonthOctober:year.fac2016   0.786090   0.226152   3.476 0.000509 ***
## MonthSeptember:year.fac2016  0.187603   0.236226   0.794 0.427098
## MonthMay:year.fac2017       0.759777   0.388618   1.955 0.050574 .
## MonthOctober:year.fac2017   0.815117   0.248146   3.285 0.001020 **
## MonthSeptember:year.fac2017  0.624811   0.241011   2.592 0.009529 **
## MonthMay:year.fac2018       0.040404   0.534332   0.076 0.939725
## MonthOctober:year.fac2018  -0.783102   0.294951  -2.655 0.007930 **
## MonthSeptember:year.fac2018 -0.882411   0.291433  -3.028 0.002463 **
## MonthMay:year.fac2019       0.522814   0.479921   1.089 0.275989
## MonthOctober:year.fac2019  -0.235566   0.362151  -0.650 0.515392
## MonthSeptember:year.fac2019  0.154697   0.328288   0.471 0.637481
## MonthMay:BlockFS           -0.003269   0.307751  -0.011 0.991526
## MonthOctober:BlockFS        0.501380   0.222050   2.258 0.023948 *
## MonthSeptember:BlockFS      0.215156   0.193635   1.111 0.266507
## MonthMay:BlockNN           -0.487988   0.307183  -1.589 0.112153
```

```
## MonthOctober:BlockNN         1.547643   0.250323   6.183 6.31e-10 ***
## MonthSeptember:BlockNN       0.996486   0.225951   4.410 1.03e-05 ***
## MonthMay:BlockNS            -1.008845   0.322079  -3.132 0.001734 **
## MonthOctober:BlockNS         0.328356   0.263910   1.244 0.213426
## MonthSeptember:BlockNS       0.606684   0.251463   2.413 0.015838 *
## year.fac2014:BlockFS        -0.321144   0.322883  -0.995 0.319924
## year.fac2015:BlockFS        -0.425714   0.320236  -1.329 0.183724
## year.fac2016:BlockFS        -0.818158   0.278651  -2.936 0.003323 **
## year.fac2017:BlockFS        -0.475865   0.293726  -1.620 0.105211
## year.fac2018:BlockFS         0.002378   0.378719   0.006 0.994990
## year.fac2019:BlockFS        -1.320620   0.352649  -3.745 0.000180 ***
## year.fac2014:BlockNN        -1.250288   0.360185  -3.471 0.000518 ***
## year.fac2015:BlockNN        -1.835234   0.350416  -5.237 1.63e-07 ***
## year.fac2016:BlockNN        -1.859675   0.322966  -5.758 8.51e-09 ***
## year.fac2017:BlockNN        -2.025152   0.336951  -6.010 1.85e-09 ***
## year.fac2018:BlockNN        -1.427552   0.413288  -3.454 0.000552 ***
## year.fac2019:BlockNN        -1.803523   0.395232  -4.563 5.04e-06 ***
## year.fac2014:BlockNS        -0.599369   0.376037  -1.594 0.110956
## year.fac2015:BlockNS        -1.226233   0.365056  -3.359 0.000782 ***
## year.fac2016:BlockNS        -1.163062   0.338696  -3.434 0.000595 ***
## year.fac2017:BlockNS        -1.008387   0.350986  -2.873 0.004066 **
## year.fac2018:BlockNS        -0.824944   0.435791  -1.893 0.058361 .
## year.fac2019:BlockNS        -0.972974   0.407881  -2.385 0.017059 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10673.5  on 8052  degrees of freedom
## Residual deviance:  9280.9  on 7995  degrees of freedom
## AIC: 9396.9
##
## Number of Fisher Scoring iterations: 4
```

Females with Eggs Spent

First show the data ...

```
#set working data frame for Spent cohort (June-Aug)
which.dat <- lobsters7.ovig.bySurvey %>% filter(Cohort3) %>%
    mutate(Spent.f = factor(Spent))
which.dat

## # A tibble: 10,188 x 32
##     Year Month Block Fisher.ID Date       Trawl.. Trap.. Auto..  Size   Sex
##    <int> <chr> <chr> <chr>     <date>       <int>  <int>  <int> <dbl> <int>
##  1  2013 June  NS    BM1       2013-06-13       1     12  32452  86.1     2
##  2  2013 June  NS    BM1       2013-06-13       1     12  32455  96.9     2
##  3  2013 June  NS    BM1       2013-06-13       1     12  32456  84.2     2
##  4  2013 June  NS    BM1       2013-06-13       1     11  32449  82.1     2
##  5  2013 June  NS    BM1       2013-06-13       1     11  32451  82       2
##  6  2013 June  NS    BM1       2013-06-13       1      9  32445  86.6     2
##  7  2013 June  NS    BM1       2013-06-13       1      9  32446 101.      2
##  8  2013 June  NS    BM1       2013-06-13       1      8  32444 109.      2
##  9  2013 June  NS    BM1       2013-06-13       1      4  32441  85.4     2
```

```
## 10  2013 June  NS    BM1       2013-06-13       1      1  32439  86.5     2
## # ... with 10,178 more rows, and 22 more variables: Eggs <int>,
## #   Shell.Hardness <int>, Cull <int>, Pathology <int>, Shell.Disease <int>,
## #   Mortality <lgl>, Cement <int>, Ovary <lgl>, Flag_Compromised <lgl>,
## #   Flag_Lobster <lgl>, block <fct>, year.fac <fct>, month.num <dbl>,
## #   avg.temp.C <dbl>, Period <chr>, Cohort1 <lgl>, Cohort2 <lgl>,
## #   Cohort3 <lgl>, Ovig <dbl>, Spent <dbl>, temp.ctr <dbl[,1]>, Spent.f <fct>
```

```
#summarize the counts:
with(which.dat, table(Block, Year, Spent))
```

```
## , , Spent = 0
##
##      Year
## Block 2013 2014 2015 2016 2017 2018 2019
##    FN  649  186  317  729  450  274  419
##    FS  691  390  343  597  529  378  331
##    NN  147  152  188  288  156  106   92
##    NS  128  105  105  127   78   74   56
##
## , , Spent = 1
##
##      Year
## Block 2013 2014 2015 2016 2017 2018 2019
##    FN  249   93   74  148  125   51   64
##    FS  307  137   91  135  124   52   75
##    NN   43   34   36   36   48   10   10
##    NS   54   18   35   11   25    8   10
```

Fit single model for Spent females, using stepAIC

```
##use stepAIC
foo.null <- glm(Spent.f ~ 1, family=binomial(link="logit"), data=which.dat)
foo.stepaic <- stepAIC(foo.null, scope=list(upper=~0 + Block*year.fac*Month +
   temp.ctr + I(temp.ctr^2)), direction="both")
```

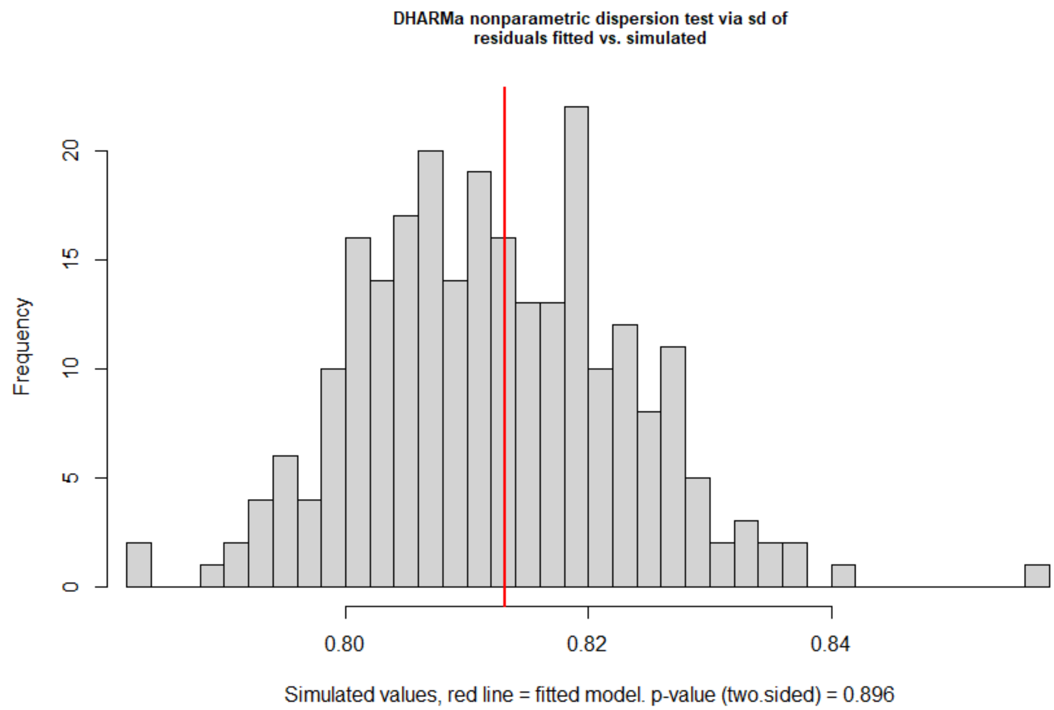
```
## Start:  AIC=10376.92
## Spent.f ~ 1
##
##                 Df Deviance     AIC
## + Month          2   9013.3  9019.3
## + temp.ctr       1   9619.5  9623.5
## + I(temp.ctr^2)  1   9985.4  9989.4
## + year.fac       6  10183.0 10197.0
## + Block          3  10351.0 10359.0
## <none>              10374.9 10376.9
##
## Step:  AIC=9019.31
## Spent.f ~ Month
##
##                 Df Deviance     AIC
## + year.fac       6   8903.5  8921.5
## + Block          3   8928.7  8940.7
## + I(temp.ctr^2)  1   9008.5  9016.5
## <none>               9013.3  9019.3
## + temp.ctr       1   9013.2  9021.2
```

```
## - Month          2  10374.9 10376.9
##
## Step:  AIC=8921.54
## Spent.f ~ Month + year.fac
##
##                  Df Deviance     AIC
## + year.fac:Month 12   8537.4  8579.4
## + Block           3   8815.8  8839.8
## + temp.ctr        1   8869.9  8889.9
## + I(temp.ctr^2)   1   8897.2  8917.2
## <none>                8903.5  8921.5
## - year.fac        6   9013.3  9019.3
## - Month           2  10183.0 10197.0
##
## Step:  AIC=8579.36
## Spent.f ~ Month + year.fac + Month:year.fac
##
##                  Df Deviance    AIC
## + Block           3   8449.9 8497.9
## + temp.ctr        1   8476.0 8520.0
## + I(temp.ctr^2)   1   8517.7 8561.7
## <none>                8537.4 8579.4
## - Month:year.fac 12   8903.5 8921.5
##
## Step:  AIC=8497.88
## Spent.f ~ Month + year.fac + Block + Month:year.fac
##
##                  Df Deviance    AIC
## + Block:Month     6   8392.2 8452.2
## + I(temp.ctr^2)   1   8443.5 8493.5
## <none>                8449.9 8497.9
## + temp.ctr        1   8449.3 8499.3
## + Block:year.fac 18   8426.4 8510.4
## - Block           3   8537.4 8579.4
## - Month:year.fac 12   8815.8 8839.8
##
## Step:  AIC=8452.19
## Spent.f ~ Month + year.fac + Block + Month:year.fac + Month:Block
##
##                  Df Deviance    AIC
## <none>                8392.2 8452.2
## + temp.ctr        1   8391.9 8453.9
## + I(temp.ctr^2)   1   8392.0 8454.0
## + Block:year.fac 18   8359.8 8455.8
## - Month:Block     6   8449.9 8497.9
## - Month:year.fac 12   8719.5 8755.5
```

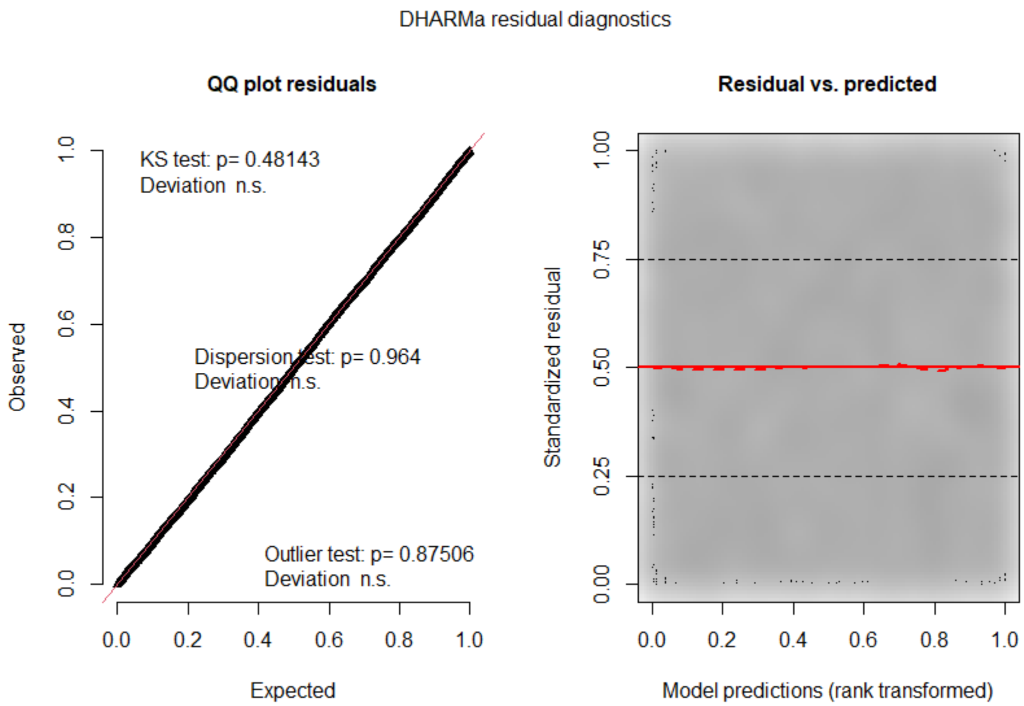
Re-ran stepAIC excluding month and considering temp.ctr variables. Like Month, temperature interacts with year and block. AIC values are similar. And we need month in the model to estimate contrasts. Stick with model above (M*Y*B + M:Y + M:B)

Check residuals plots …

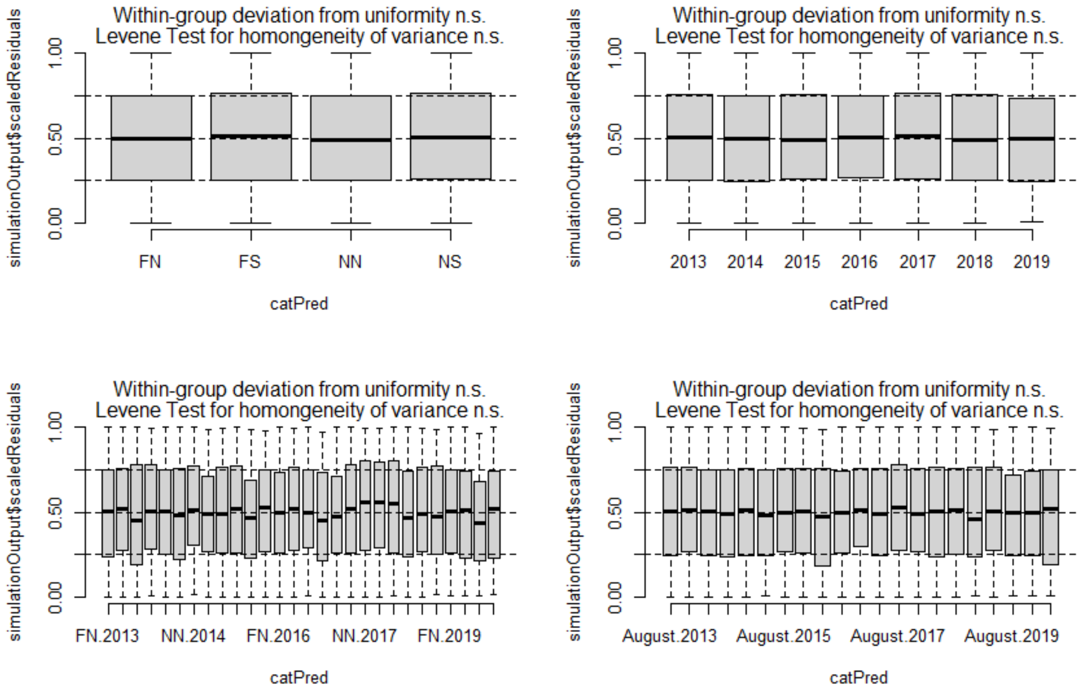
```
#check with DHARMa
 testDispersion(foo.stepaic)
```



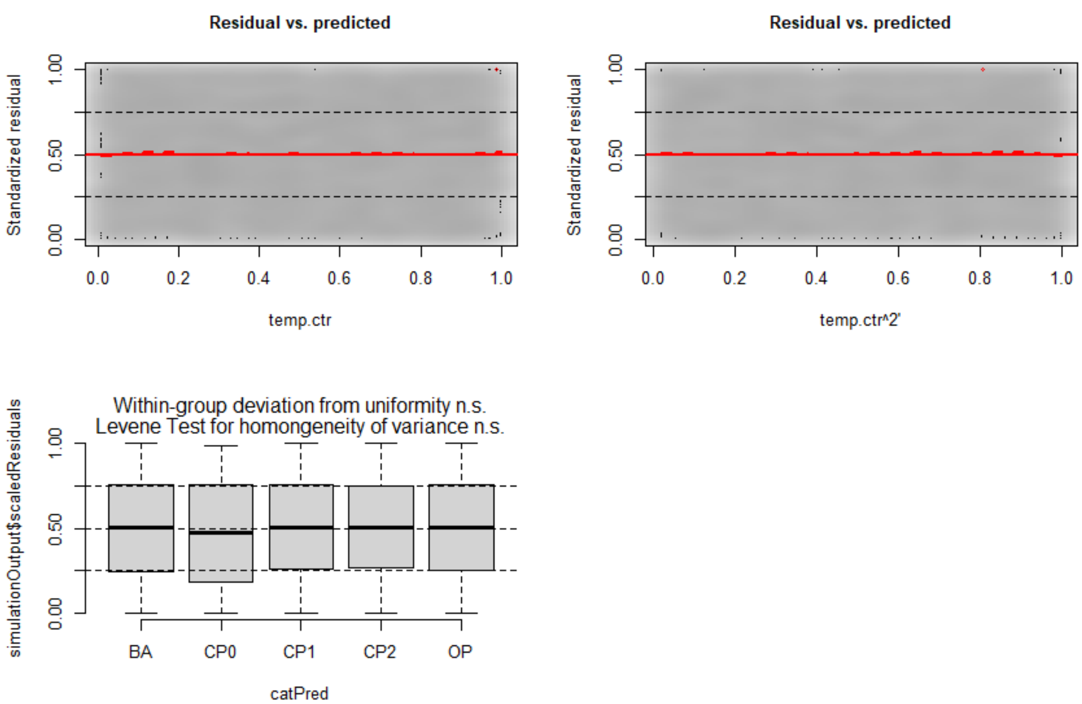

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 1.0012, p-value = 0.896
## alternative hypothesis: two.sided

 simulationOutput <- simulateResiduals(fittedModel = foo.stepaic, plot = T, n=500)
```




```
#check for model misfit. plot residuals against all predictors (both in and out of mo
del)
par(mfrow=c(2,2))
 plotResiduals(foo.stepaic, form=factor(which.dat$Block))
 plotResiduals(foo.stepaic, form=which.dat$year.fac)
 plotResiduals(foo.stepaic, form=interaction(which.dat$Block,which.dat$year.fac))
 plotResiduals(foo.stepaic, form=interaction(which.dat$Month,which.dat$year.fac))
```




```
par(mfrow=c(2,2));
 plotResiduals(foo.stepaic, form=which.dat$temp.ctr, xlab="temp.ctr")
 plotResiduals(foo.stepaic, form=I(which.dat$temp.ctr^2), xlab="temp.ctr^2'")
 plotResiduals(foo.stepaic, form=factor(which.dat$Period))
```



Finalize model, and show results:

```r
lobsters7.FishRes.Spent.glm <- glm(formula = Spent.f ~ Month + year.fac +
 Block + Month:year.fac + Month:Block, family = binomial(link = "logit"),
 data = lobsters7.ovig.bySurvey %>% filter(Cohort3) %>%
    mutate(Spent.f = factor(Spent)))
summary(lobsters7.FishRes.Spent.glm)

##
## Call:
## glm(formula = Spent.f ~ Month + year.fac + Block + Month:year.fac +
##     Month:Block, family = binomial(link = "logit"), data = lobsters7.ovig.bySurvey
%>%
##     filter(Cohort3) %>% mutate(Spent.f = factor(Spent)))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4358  -0.7403  -0.3756  -0.1643   3.1934
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -2.08042    0.15489 -13.431  < 2e-16 ***
## MonthJuly                 0.96827    0.17541   5.520 3.39e-08 ***
## MonthJune                 2.66988    0.19227  13.886  < 2e-16 ***
## year.fac2014              0.17410    0.19695   0.884 0.376705
## year.fac2015             -0.45816    0.20360  -2.250 0.024431 *
## year.fac2016             -1.26806    0.23828  -5.322 1.03e-07 ***
## year.fac2017             -1.24718    0.27910  -4.469 7.87e-06 ***
## year.fac2018             -1.59606    0.31972  -4.992 5.97e-07 ***
## year.fac2019             -2.14040    0.40278  -5.314 1.07e-07 ***
## BlockFS                  -0.07767    0.14811  -0.524 0.600021
## BlockNN                  -1.74418    0.42771  -4.078 4.54e-05 ***
## BlockNS                  -0.25669    0.29254  -0.877 0.380240
## MonthJuly:year.fac2014    0.85502    0.23572   3.627 0.000286 ***
## MonthJune:year.fac2014   -1.61814    0.26644  -6.073 1.25e-09 ***
## MonthJuly:year.fac2015    1.67886    0.24552   6.838 8.03e-12 ***
## MonthJune:year.fac2015   -1.32252    0.31824  -4.156 3.24e-05 ***
## MonthJuly:year.fac2016    1.01571    0.26430   3.843 0.000122 ***
## MonthJune:year.fac2016    0.36938    0.27456   1.345 0.178500
## MonthJuly:year.fac2017    0.81806    0.30709   2.664 0.007724 **
## MonthJune:year.fac2017    0.92852    0.30919   3.003 0.002673 **
## MonthJuly:year.fac2018    1.62281    0.35480   4.574 4.79e-06 ***
## MonthJune:year.fac2018    0.44260    0.38958   1.136 0.255910
## MonthJuly:year.fac2019    2.11582    0.42288   5.003 5.63e-07 ***
## MonthJune:year.fac2019    1.68053    0.46926   3.581 0.000342 ***
## MonthJuly:BlockFS         0.51289    0.16938   3.028 0.002461 **
## MonthJune:BlockFS        -0.38339    0.18856  -2.033 0.042030 *
## MonthJuly:BlockNN         1.33918    0.44956   2.979 0.002893 **
## MonthJune:BlockNN         0.89904    0.45052   1.996 0.045979 *
## MonthJuly:BlockNS         0.13393    0.32992   0.406 0.684782
## MonthJune:BlockNS        -0.84199    0.33883  -2.485 0.012956 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10374.9  on 10187  degrees of freedom
```

```
## Residual deviance:  8392.2  on 10158  degrees of freedom
## AIC: 8452.2
##
## Number of Fisher Scoring iterations: 7
```

Shell Disease in Females

First show the data ...

```
 #We use both compromised and uncompromised traps for these models.
 #But filter for:
 # Mortality = FALSE
 # Females (sex code = 2)
 # Ovigerous or spent (egg code non-zero and >0)
 # Month = May, June, or July
 # Shell.Disease = 0 (no) or >0 (yes)
 #Individuals with a blank in the shell disease field (n=5) were ignored
#set working data frame
which.dat <- lobsters7.disease.bySurvey %>%
  mutate(Diseased.f = factor(Diseased.yn), Block=factor(Block),
    Month=factor(Month), Period=factor(Period))
which.dat

## # A tibble: 4,085 x 28
##     Year Month Block Fisher.ID Date       Trawl.. Trap.. Auto..  Size   Sex
##    <int> <fct> <fct> <chr>     <date>       <int>  <int>  <int> <dbl> <int>
##  1  2013 May   NS    BM1       2013-05-31       1     10  51905  86.2     2
##  2  2013 June  NS    BM1       2013-06-13       1     12  32456  84.2     2
##  3  2013 June  NS    BM1       2013-06-13       1     11  32449  82.1     2
##  4  2013 June  NS    BM1       2013-06-13       1     11  32451  82       2
##  5  2013 June  NS    BM1       2013-06-13       1      9  32445  86.6     2
##  6  2013 June  NS    BM1       2013-06-13       1      4  32441  85.4     2
##  7  2013 June  NS    BM1       2013-06-13       1      1  32438  70       2
##  8  2013 June  NS    BM1       2013-06-13       1      1  32439  86.5     2
##  9  2013 June  NS    BM1       2013-06-13       1      7  32442  92.6     2
## 10  2013 June  NS    BM1       2013-06-18       1     11  32773  87       2
## # ... with 4,075 more rows, and 18 more variables: Eggs <int>,
## #   Shell.Hardness <int>, Cull <int>, Pathology <int>, Shell.Disease <int>,
## #   Mortality <lgl>, Cement <int>, Ovary <lgl>, Flag_Compromised <lgl>,
## #   Flag_Lobster <lgl>, block <fct>, year.fac <fct>, month.num <dbl>,
## #   avg.temp.C <dbl>, Period <fct>, Diseased.yn <dbl>, temp.ctr <dbl[,1]>,
## #   Diseased.f <fct>

#summarize the counts:
with(which.dat, table(Block, Year, Diseased.yn))

## , , Diseased.yn = 0
##
##      Year
## Block 2013 2014 2015 2016 2017 2018 2019
##    FN   35   17   12   32   34   17   23
##    FS   86   37   20   23   49   25   22
##    NN    4    1    3    1    5    3    1
##    NS    5    4    2    1    3    3    4
##
```

```
## , , Diseased.yn = 1
##
##      Year
## Block 2013 2014 2015 2016 2017 2018 2019
##    FN  307  145   69  232  215   72   86
##    FS  437  294  119  266  244  118  111
##    NN   63   92  113  102   85   38   35
##    NS   87   55   76   45   55   28   24
```

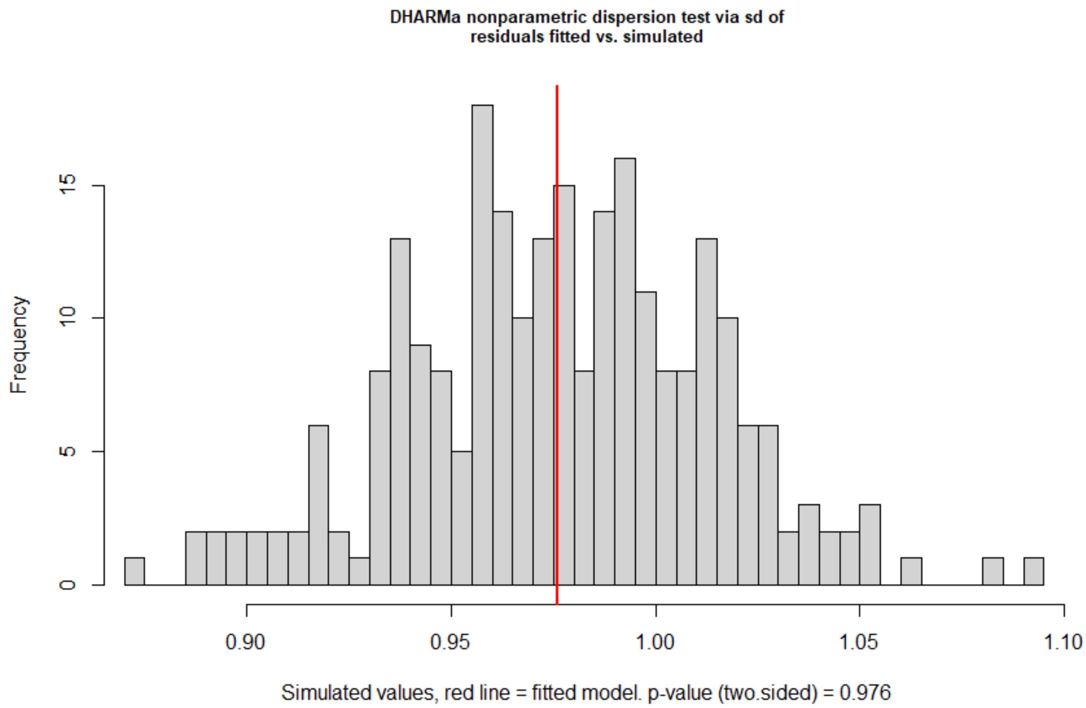
Fit model using stepAIC ...

```
##use stepAIC
foo.null <- glm(Diseased.f ~ 1, family=binomial(link="logit"), data=which.dat)
foo.stepaic <- stepAIC(foo.null, scope=list(upper=~0 +
   Block*year.fac*Month + Block*Period + temp.ctr + I(temp.ctr^2)),
   direction="both")
```

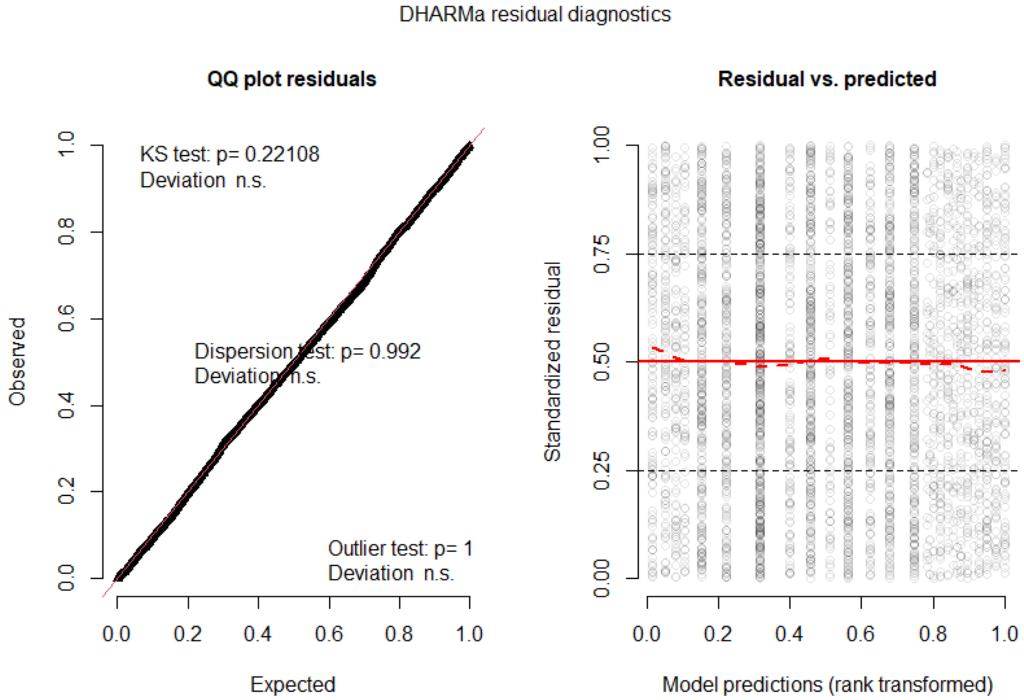
```
## Start:  AIC=2926.48
## Diseased.f ~ 1
##
##                  Df Deviance    AIC
## + Block           3   2844.5 2852.5
## + Period          2   2902.1 2908.1
## + year.fac        6   2894.6 2908.6
## + Month           2   2919.0 2925.0
## <none>                2924.5 2926.5
## + I(temp.ctr^2)   1   2922.8 2926.8
## + temp.ctr        1   2924.3 2928.3
##
## Step:  AIC=2852.51
## Diseased.f ~ Block
##
##                  Df Deviance    AIC
## + year.fac        6   2818.6 2838.6
## + Period          2   2826.6 2838.6
## <none>                2844.5 2852.5
## + I(temp.ctr^2)   1   2843.7 2853.7
## + temp.ctr        1   2844.0 2854.0
## + Month           2   2842.1 2854.1
## - Block           3   2924.5 2926.5
##
## Step:  AIC=2838.63
## Diseased.f ~ Block + year.fac
##
##                   Df Deviance    AIC
## <none>                 2818.6 2838.6
## + I(temp.ctr^2)    1   2817.5 2839.5
## + temp.ctr         1   2818.6 2840.6
## + Month            2   2816.9 2840.9
## - year.fac         6   2844.5 2852.5
## + Block:year.fac  18   2798.4 2854.4
## - Block            3   2894.6 2908.6
```

Check residuals plots ...

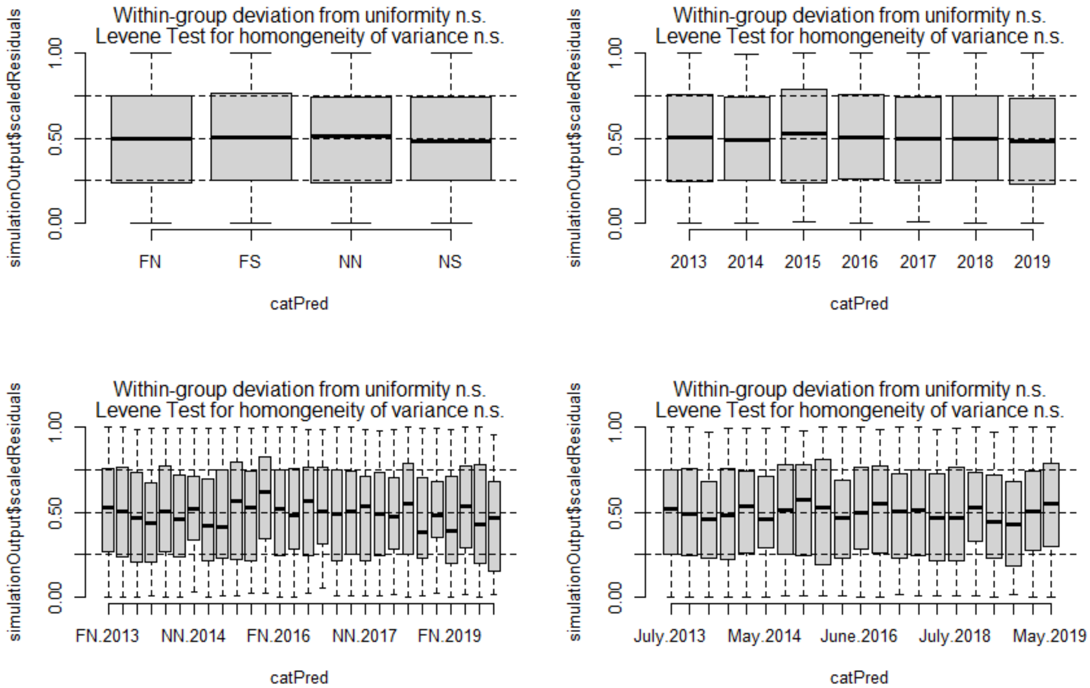
```r
#check with DHARMa
 testDispersion(foo.stepaic)
```



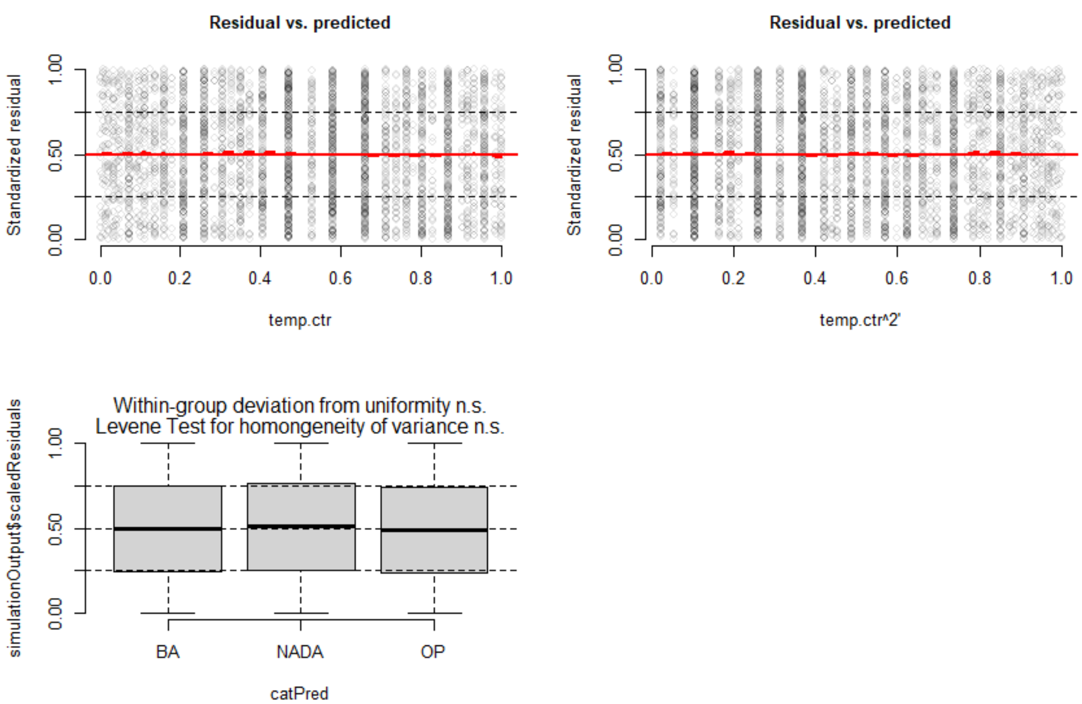

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.99931, p-value = 0.976
## alternative hypothesis: two.sided

 simulationOutput <- simulateResiduals(fittedModel = foo.stepaic, plot = T, n=500)
```




```
#check for model misfit. plot residuals against all predictors (both in and out of mo
del)
par(mfrow=c(2,2))
 plotResiduals(foo.stepaic, form=factor(which.dat$Block))
 plotResiduals(foo.stepaic, form=which.dat$year.fac)
 plotResiduals(foo.stepaic, form=interaction(which.dat$Block,which.dat$year.fac))
 plotResiduals(foo.stepaic, form=interaction(which.dat$Month,which.dat$year.fac))
```




```r
par(mfrow=c(2,2));
 plotResiduals(foo.stepaic, form=which.dat$temp.ctr, xlab="temp.ctr")
 plotResiduals(foo.stepaic, form=I(which.dat$temp.ctr^2), xlab="temp.ctr^2'")
 plotResiduals(foo.stepaic, form=factor(which.dat$Period))
```



Finalize model, and show results:

```r
lobsters7.FishRes.Disease.glm <- glm(formula = Diseased.f ~ Block + year.fac,
    family = binomial(link = "logit"), data = lobsters7.disease.bySurvey %>%
    mutate(Diseased.f = factor(Diseased.yn)))
summary(lobsters7.FishRes.Disease.glm)
```

```
##
## Call:
## glm(formula = Diseased.f ~ Block + year.fac, family = binomial(link = "logit"),
##     data = lobsters7.disease.bySurvey %>% mutate(Diseased.f = factor(Diseased.yn))
)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7514   0.2966   0.4887   0.5617   0.6614
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.85911    0.11582  16.051  < 2e-16 ***
## BlockFS      -0.09217    0.10677  -0.863   0.3880
## BlockNN       1.46737    0.25554   5.742 9.34e-09 ***
## BlockNS       0.94463    0.23618   4.000 6.35e-05 ***
## year.fac2014  0.29806    0.16701   1.785   0.0743 .
## year.fac2015  0.07926    0.20058   0.395   0.6927
## year.fac2016  0.43575    0.16826   2.590   0.0096 **
## year.fac2017 -0.11044    0.14785  -0.747   0.4551
## year.fac2018 -0.33832    0.18514  -1.827   0.0676 .
## year.fac2019 -0.35849    0.18268  -1.962   0.0497 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2924.5  on 4084  degrees of freedom
## Residual deviance: 2818.6  on 4075  degrees of freedom
## AIC: 2838.6
##
## Number of Fisher Scoring iterations: 6
```

Cull Rates in both Males and Females

First show the data ...

```r
 #We use both compromised and uncompromised traps for these models.
 #But filter for:
 # Mortality = FALSE
 # Sex = 1 (Males) or 2 (Females); omit 0 (unknown)
 # Intact lobsters (both claws present) = cull code = 0
 # Culls = cull code = 11 or 22
 # Individuals with a blank in the cull field (n=20) were ignored
#set working data frame:
 which.dat <- lobsters7.cull.bySurvey  %>% mutate(Sex.f = factor(Sex))
 which.dat
```

```
## # A tibble: 44,805 x 28
##     Year Month Block Fisher.ID Date       Trawl.. Trap.. Auto..  Size   Sex
```

```
##    <int> <chr> <chr> <chr>     <date>       <int>  <int>  <int> <dbl> <int>
##  1  2013 May   NS    BM1       2013-05-24       1      4  32316  83.4     2
##  2  2013 May   NS    BM1       2013-05-24       1      2  32315 101.      1
##  3  2013 May   NS    BM1       2013-05-31       1     10  51905  86.2     2
##  4  2013 June  NS    BM1       2013-06-13       1     12  32452  86.1     2
##  5  2013 June  NS    BM1       2013-06-13       1     12  32455  96.9     2
##  6  2013 June  NS    BM1       2013-06-13       1     12  32456  84.2     2
##  7  2013 June  NS    BM1       2013-06-13       1     11  32447  79       1
##  8  2013 June  NS    BM1       2013-06-13       1     11  32448  56.2     2
##  9  2013 June  NS    BM1       2013-06-13       1     11  32449  82.1     2
## 10  2013 June  NS    BM1       2013-06-13       1     11  32450  97.9     1
## # ... with 44,795 more rows, and 18 more variables: Eggs <int>,
## #   Shell.Hardness <int>, Cull <int>, Pathology <int>, Shell.Disease <int>,
## #   Mortality <lgl>, Cement <int>, Ovary <lgl>, Flag_Compromised <lgl>,
## #   Flag_Lobster <lgl>, block <fct>, year.fac <fct>, month.num <dbl>,
## #   avg.temp.C <dbl>, Period <chr>, cull.f <fct>, temp.ctr <dbl[,1]>,
## #   Sex.f <fct>
```

```
#summarize data:
 with(which.dat, table(Block, Month, cull.f, Sex.f))
```

```
## , , cull.f = 0, Sex.f = 1
## 
##      Month
## Block August July June  May October September
##    FN   2178 1253  125   25     471      1290
##    FS   2057  630  143   34     715      1553
##    NN   1023  172  192  137     548       861
##    NS    687  131  124  105     623       758
## 
## , , cull.f = 1, Sex.f = 1
## 
##      Month
## Block August July June  May October September
##    FN    209  102   17    9      60       134
##    FS    224   58   18    3      88       168
##    NN    148   18   34   11      60       112
##    NS     90   24   13   10      92       103
## 
## , , cull.f = 0, Sex.f = 2
## 
##      Month
## Block August July June  May October September
##    FN   2715 2366  764  145     264       860
##    FS   3313 2008 1055  355    1531      2565
##    NN   1020  544  628  356     738       936
##    NS    555  382  346  244     580       548
## 
## , , cull.f = 1, Sex.f = 2
## 
##      Month
## Block August July June  May October September
##    FN    240  155   42   11      43        98
##    FS    307  152   68   27     184       250
```

```
##    NN     128    48    51    18      100        110
##    NS      67    33    29    32       77         77
```

Fit model using stepAIC …

```
##use stepAIC
foo.null <- glm(cull.f ~ 1, family=binomial(link="logit"), data=which.dat)
foo.stepaic <- stepAIC(foo.null, scope=list(upper=~0 + Block*year.fac*Month +
   Sex.f + temp.ctr + I(temp.ctr^2)), direction="both")
```

```
## Start:  AIC=27661.74
## cull.f ~ 1
## 
##                  Df Deviance   AIC
## + temp.ctr        1    27569 27573
## + Month           5    27562 27574
## + Block           3    27599 27607
## + Sex.f           1    27628 27632
## + year.fac        6    27623 27637
## + I(temp.ctr^2)   1    27656 27660
## <none>                 27660 27662
## 
## Step:  AIC=27572.46
## cull.f ~ temp.ctr
## 
##                  Df Deviance   AIC
## + Block           3    27536 27546
## + Sex.f           1    27551 27557
## + year.fac        6    27541 27557
## + Month           5    27551 27565
## <none>                 27569 27573
## + I(temp.ctr^2)   1    27568 27574
## - temp.ctr        1    27660 27662
## 
## Step:  AIC=27546.42
## cull.f ~ temp.ctr + Block
## 
##                  Df Deviance   AIC
## + Sex.f           1    27519 27531
## + year.fac        6    27510 27532
## + Month           5    27517 27537
## <none>                 27536 27546
## + I(temp.ctr^2)   1    27536 27548
## - Block           3    27569 27573
## - temp.ctr        1    27599 27607
## 
## Step:  AIC=27531.08
## cull.f ~ temp.ctr + Block + Sex.f
## 
##                  Df Deviance   AIC
## + year.fac        6    27492 27516
## + Month           5    27503 27525
## <none>                 27519 27531
## + I(temp.ctr^2)   1    27519 27533
## - Sex.f           1    27536 27546
```

```
## - Block          3    27551 27557
## - temp.ctr       1    27571 27581
##
## Step:  AIC=27516.35
## cull.f ~ temp.ctr + Block + Sex.f + year.fac
##
##                   Df Deviance   AIC
## + Month            5    27471 27505
## <none>                  27492 27516
## + I(temp.ctr^2)   1    27491 27517
## + Block:year.fac 18    27458 27518
## - year.fac        6    27519 27531
## - Sex.f           1    27510 27532
## - Block           3    27523 27541
## - temp.ctr        1    27539 27561
##
## Step:  AIC=27504.97
## cull.f ~ temp.ctr + Block + Sex.f + year.fac + Month
##
##                   Df Deviance   AIC
## + year.fac:Month 30    27398 27492
## - temp.ctr        1    27472 27504
## <none>                  27471 27505
## + Block:Month    15    27442 27506
## + I(temp.ctr^2)   1    27471 27507
## + Block:year.fac 18    27440 27510
## - Month           5    27492 27516
## - Sex.f           1    27485 27517
## - year.fac        6    27503 27525
## - Block           3    27507 27535
##
## Step:  AIC=27491.82
## cull.f ~ temp.ctr + Block + Sex.f + year.fac + Month + year.fac:Month
##
##                   Df Deviance   AIC
## - temp.ctr        1    27398 27490
## + Block:year.fac 18    27362 27492
## <none>                  27398 27492
## + Block:Month    15    27368 27492
## + I(temp.ctr^2)   1    27397 27493
## - Sex.f           1    27410 27502
## - year.fac:Month 30    27471 27505
## - Block           3    27423 27511
##
## Step:  AIC=27489.82
## cull.f ~ Block + Sex.f + year.fac + Month + year.fac:Month
##
##                   Df Deviance   AIC
## <none>                  27398 27490
## + Block:year.fac 18    27363 27491
## + I(temp.ctr^2)   1    27397 27491
## + temp.ctr        1    27398 27492
## + Block:Month    15    27371 27493
## - Sex.f           1    27410 27500
```

```
## - year.fac:Month 30    27472 27504
## - Block           3    27436 27522
```

```r
summary(foo.stepaic)
```

```
##
## Call:
## glm(formula = cull.f ~ Block + Sex.f + year.fac + Month + year.fac:Month,
##     family = binomial(link = "logit"), data = which.dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6567  -0.4645  -0.4280  -0.3871   2.5820
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -2.2241075  0.0805819 -27.601  < 2e-16 ***
## BlockFS                     0.0496461  0.0422136   1.176 0.239568
## BlockNN                     0.2235868  0.0502051   4.453 8.45e-06 ***
## BlockNS                     0.2744836  0.0544742   5.039 4.69e-07 ***
## Sex.f2                     -0.1201344  0.0340528  -3.528 0.000419 ***
## year.fac2014                0.1990408  0.1046233   1.902 0.057112 .
## year.fac2015               -0.2172458  0.1044753  -2.079 0.037581 *
## year.fac2016               -0.0219329  0.0919815  -0.238 0.811532
## year.fac2017               -0.1456067  0.1106795  -1.316 0.188318
## year.fac2018               -0.1931072  0.1165205  -1.657 0.097463 .
## year.fac2019               -0.0497271  0.1157892  -0.429 0.667587
## MonthJuly                  -0.3295592  0.1107929  -2.975 0.002934 **
## MonthJune                  -0.4500888  0.1517300  -2.966 0.003013 **
## MonthMay                   -0.2232124  0.3762035  -0.593 0.552961
## MonthOctober                0.0451992  0.1308739   0.345 0.729820
## MonthSeptember             -0.1205969  0.1245691  -0.968 0.332988
## year.fac2014:MonthJuly      0.1823366  0.1859078   0.981 0.326696
## year.fac2015:MonthJuly     -0.0699797  0.2282467  -0.307 0.759150
## year.fac2016:MonthJuly      0.2032579  0.1481900   1.372 0.170187
## year.fac2017:MonthJuly      0.0436484  0.1777235   0.246 0.805994
## year.fac2018:MonthJuly      0.1215384  0.2298590   0.529 0.596977
## year.fac2019:MonthJuly     -0.0523035  0.2000361  -0.261 0.793730
## year.fac2014:MonthJune     -0.0706923  0.2435781  -0.290 0.771645
## year.fac2015:MonthJune      0.5813048  0.2365535   2.457 0.013995 *
## year.fac2016:MonthJune      0.2348169  0.2107240   1.114 0.265136
## year.fac2017:MonthJune      0.2980953  0.2233356   1.335 0.181961
## year.fac2018:MonthJune      0.3657810  0.3147958   1.162 0.245251
## year.fac2019:MonthJune     -0.4282316  0.4531509  -0.945 0.344654
## year.fac2014:MonthMay      -0.0124981  0.4322740  -0.029 0.976934
## year.fac2015:MonthMay       0.1125409  0.4597206   0.245 0.806609
## year.fac2016:MonthMay       0.0676188  0.4177878   0.162 0.871424
## year.fac2017:MonthMay      -0.5839193  0.4722377  -1.236 0.216275
## year.fac2018:MonthMay       0.5663412  0.5248429   1.079 0.280557
## year.fac2019:MonthMay      -0.1510948  0.5330102  -0.283 0.776813
## year.fac2014:MonthOctober   0.1426992  0.1729400   0.825 0.409294
## year.fac2015:MonthOctober   0.0003679  0.1863438   0.002 0.998425
## year.fac2016:MonthOctober  -0.2334201  0.1701629  -1.372 0.170143
## year.fac2017:MonthOctober   0.3903172  0.1856191   2.103 0.035484 *
## year.fac2018:MonthOctober   0.2569790  0.2064111   1.245 0.213137
```

```
## year.fac2019:MonthOctober    0.5297764  0.2121575   2.497 0.012522 *
## year.fac2014:MonthSeptember -0.2527255  0.1803763  -1.401 0.161184
## year.fac2015:MonthSeptember  0.2036417  0.1655746   1.230 0.218731
## year.fac2016:MonthSeptember  0.2161273  0.1506390   1.435 0.151362
## year.fac2017:MonthSeptember  0.2898563  0.1724321   1.681 0.092765 .
## year.fac2018:MonthSeptember  0.3312414  0.1816859   1.823 0.068280 .
## year.fac2019:MonthSeptember  0.3361817  0.1778423   1.890 0.058713 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27660  on 44804  degrees of freedom
## Residual deviance: 27398  on 44759  degrees of freedom
## AIC: 27490
##
## Number of Fisher Scoring iterations: 5
```

Sex coefficient has p=0.000419, indicating differences between sexes. Re-fit model on sexes separately. First Males

```
#MALES
foo.null <- glm(cull.f ~ 1, family=binomial(link="logit"), data=which.dat %>% filter
(Sex==1))
foo.stepaicM <- stepAIC(foo.null, scope=list(upper=~0 + Block*year.fac*Month +
   temp.ctr + I(temp.ctr^2)), direction="both")
```

```
## Start:  AIC=11650.05
## cull.f ~ 1
##
##                 Df Deviance   AIC
## + year.fac       6    11604 11618
## + Block          3    11622 11630
## + temp.ctr       1    11638 11642
## + Month          5    11633 11645
## + I(temp.ctr^2)  1    11643 11647
## <none>                11648 11650
##
## Step:  AIC=11618.13
## cull.f ~ year.fac
##
##                 Df Deviance   AIC
## + Block          3    11583 11603
## + temp.ctr       1    11598 11614
## + Month          5    11592 11616
## <none>                11604 11618
## + I(temp.ctr^2)  1    11604 11620
## - year.fac       6    11648 11650
##
## Step:  AIC=11602.67
## cull.f ~ year.fac + Block
##
##                  Df Deviance   AIC
## <none>                 11583 11603
## + temp.ctr        1    11581 11603
```

```
## + I(temp.ctr^2)   1     11582 11604
## + Month           5     11576 11606
## + Block:year.fac 18     11559 11615
## - Block           3     11604 11618
## - year.fac        6     11622 11630
```

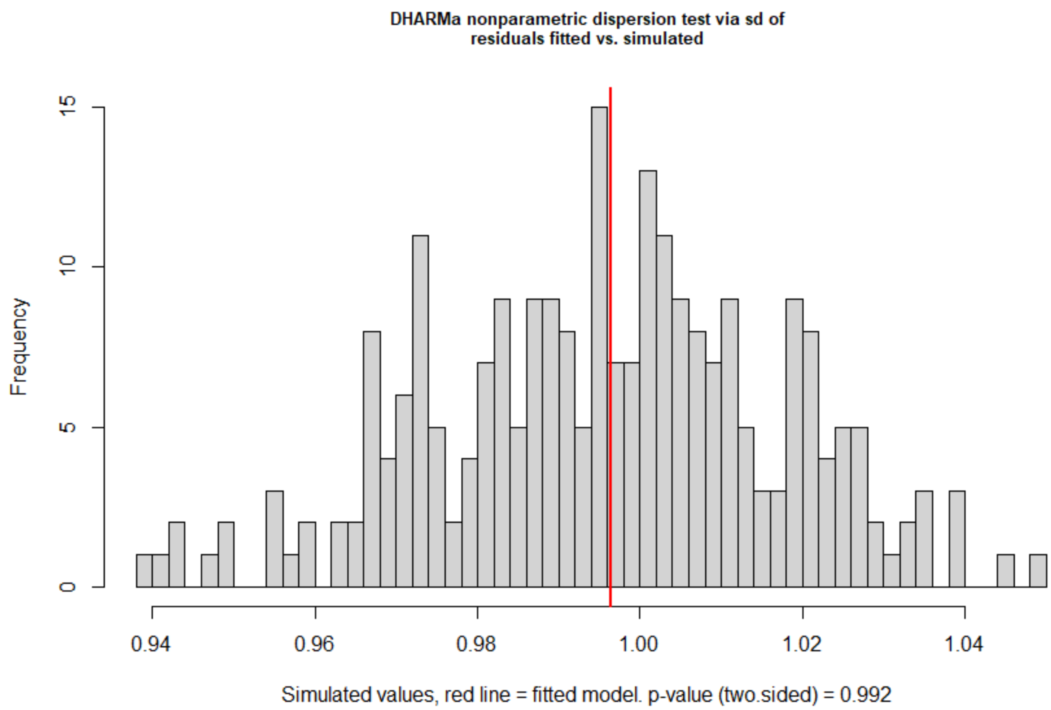
That model includes only block and year. Require block, month, and year so that all contrasts can be estimated.

```
foo.basic <- glm(cull.f ~ 0 + Block + Month + year.fac,family=binomial(link="logit"),
    data=which.dat %>% filter(Sex==1))
foo.stepaicM <- stepAIC(foo.basic, scope=list(lower=foo.basic, upper=~0 +
 Block*year.fac*Month +     temp.ctr + I(temp.ctr^2)),direction="both")

## Start:  AIC=11605.52
## cull.f ~ 0 + Block + Month + year.fac
##
##                 Df Deviance   AIC
## <none>                 11576 11606
## + I(temp.ctr^2)   1     11575 11607
## + temp.ctr        1     11575 11607
## + year.fac:Month 30     11518 11608
## + Block:Month    15     11550 11610
## + Block:year.fac 18     11552 11618

summary(foo.stepaicM)

##
## Call:
## glm(formula = cull.f ~ 0 + Block + Month + year.fac, family = binomial(link = "log
it"),
##     data = which.dat %>% filter(Sex == 1))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6282  -0.4809  -0.4473  -0.4169   2.3226
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## BlockFN        -2.380401   0.090639 -26.262  < 2e-16 ***
## BlockFS        -2.317385   0.088114 -26.300  < 2e-16 ***
## BlockNN        -2.149767   0.094340 -22.787  < 2e-16 ***
## BlockNS        -2.118752   0.098196 -21.577  < 2e-16 ***
## MonthJuly      -0.147336   0.086026  -1.713   0.0868 .
## MonthJune       0.167178   0.125958   1.327   0.1844
## MonthMay       -0.171056   0.190249  -0.899   0.3686
## MonthOctober    0.043827   0.074954   0.585   0.5587
## MonthSeptember  0.008163   0.062182   0.131   0.8956
## year.fac2014    0.428994   0.098535   4.354 1.34e-05 ***
## year.fac2015   -0.099834   0.101511  -0.983   0.3254
## year.fac2016    0.034433   0.089426   0.385   0.7002
## year.fac2017    0.128778   0.097204   1.325   0.1852
## year.fac2018    0.114682   0.107831   1.064   0.2875
## year.fac2019    0.220224   0.106509   2.068   0.0387 *
## ---
```

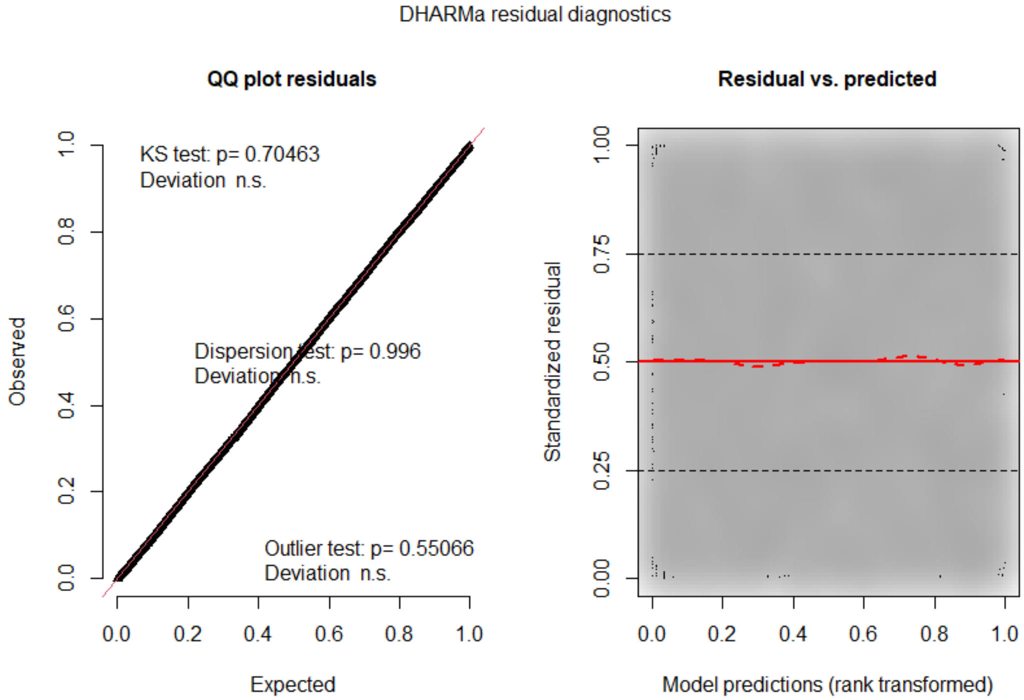
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 24454  on 17640  degrees of freedom
## Residual deviance: 11576  on 17625  degrees of freedom
## AIC: 11606
##
## Number of Fisher Scoring iterations: 5
```

Check residuals plots …

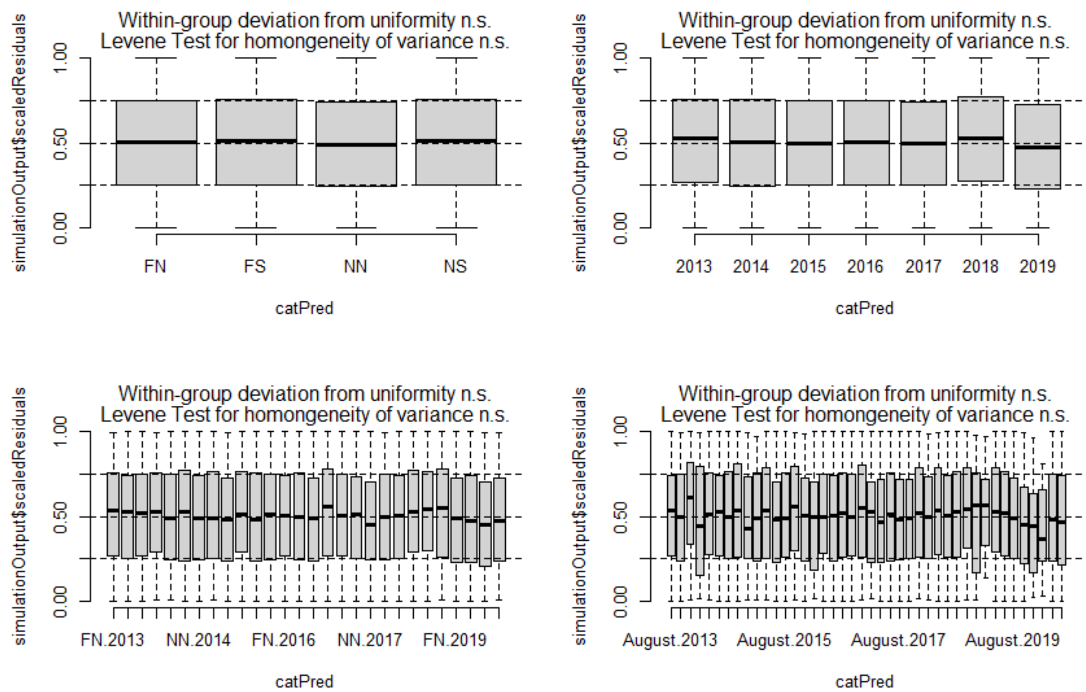
```
#check with DHARMa
 testDispersion(foo.stepaicM)
```



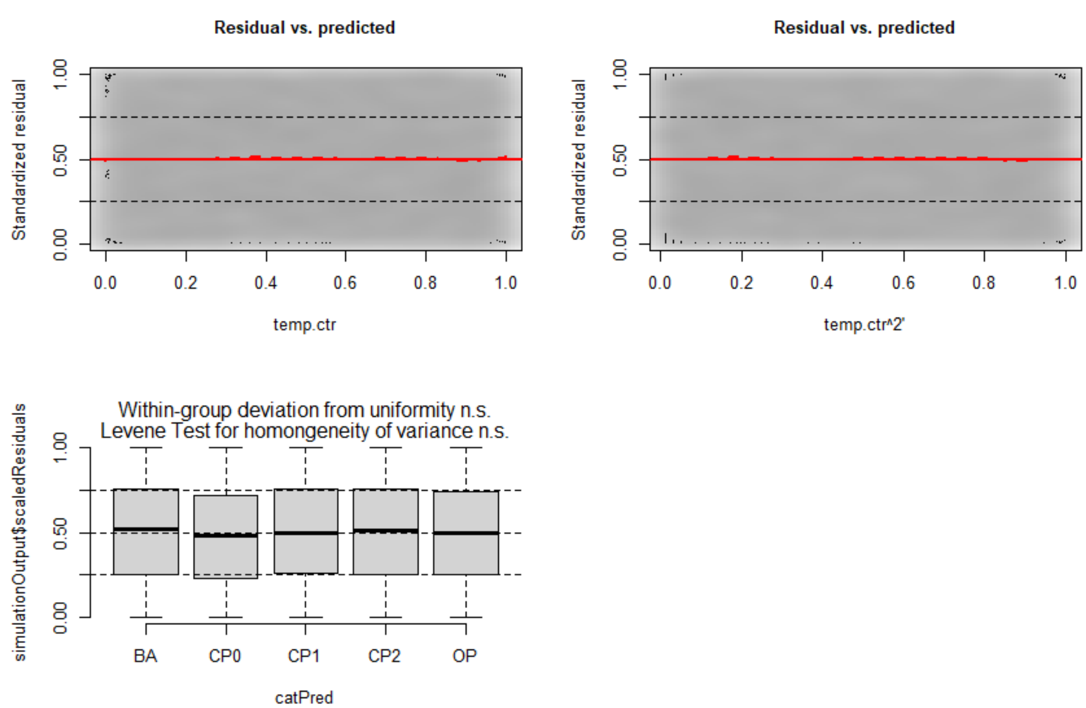

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 1.0007, p-value = 0.992
## alternative hypothesis: two.sided

 simulationOutput <- simulateResiduals(fittedModel = foo.stepaicM, plot = T, n=500)
```




```
#check for model misfit. plot residuals against all predictors (both in and out of mo
del)
 which.dat2 <- which.dat %>% filter(Sex==1)
par(mfrow=c(2,2))
 plotResiduals(foo.stepaicM, form=factor(which.dat2$Block))
 plotResiduals(foo.stepaicM, form=which.dat2$year.fac)
 plotResiduals(foo.stepaicM, form=interaction(which.dat2$Block,which.dat2$year.fac))
 plotResiduals(foo.stepaicM, form=interaction(which.dat2$Month,which.dat2$year.fac))
```




```r
par(mfrow=c(2,2));
 plotResiduals(foo.stepaicM, form=which.dat2$temp.ctr, xlab="temp.ctr")
 plotResiduals(foo.stepaicM, form=I(which.dat2$temp.ctr^2), xlab="temp.ctr^2'")
 plotResiduals(foo.stepaicM, form=factor(which.dat2$Period))
```



Finalize model and show results for Male Culls …

```r
lobsters7.FishRes.CullM.glm <- glm(formula=cull.f ~ 0 + Block + Month + year.fac,
    family = binomial(link = "logit"), data = lobsters7.cull.bySurvey  %>%
    filter(Sex == 1))
summary(lobsters7.FishRes.CullM.glm)
```

```
##
## Call:
## glm(formula = cull.f ~ 0 + Block + Month + year.fac, family = binomial(link = "log
it"),
##     data = lobsters7.cull.bySurvey %>% filter(Sex == 1))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6282  -0.4809  -0.4473  -0.4169   2.3226
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## BlockFN        -2.380401   0.090639 -26.262  < 2e-16 ***
## BlockFS        -2.317385   0.088114 -26.300  < 2e-16 ***
## BlockNN        -2.149767   0.094340 -22.787  < 2e-16 ***
## BlockNS        -2.118752   0.098196 -21.577  < 2e-16 ***
## MonthJuly      -0.147336   0.086026  -1.713   0.0868 .
## MonthJune       0.167178   0.125958   1.327   0.1844
## MonthMay       -0.171056   0.190249  -0.899   0.3686
## MonthOctober    0.043827   0.074954   0.585   0.5587
## MonthSeptember  0.008163   0.062182   0.131   0.8956
## year.fac2014    0.428994   0.098535   4.354 1.34e-05 ***
## year.fac2015   -0.099834   0.101511  -0.983   0.3254
## year.fac2016    0.034433   0.089426   0.385   0.7002
## year.fac2017    0.128778   0.097204   1.325   0.1852
## year.fac2018    0.114682   0.107831   1.064   0.2875
## year.fac2019    0.220224   0.106509   2.068   0.0387 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 24454  on 17640  degrees of freedom
## Residual deviance: 11576  on 17625  degrees of freedom
## AIC: 11606
##
## Number of Fisher Scoring iterations: 5
```

Female Culls

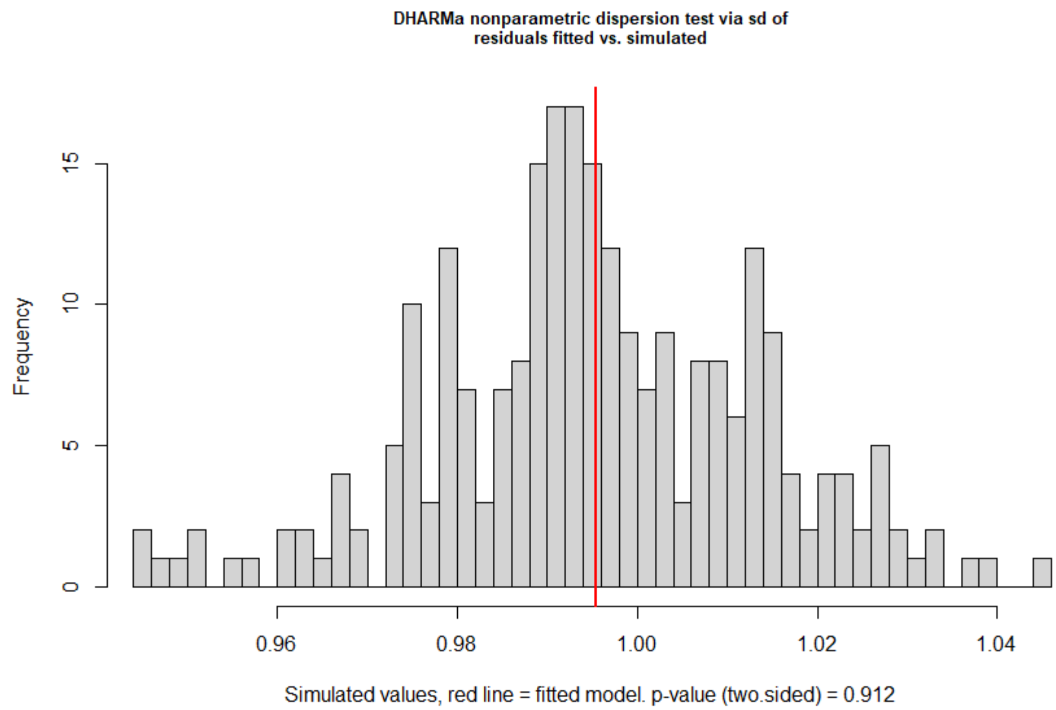
```r
 #FEMALES
 foo.null <- glm(cull.f ~ 1, family=binomial(link="logit"), data=which.dat %>% filter
(Sex==2))
 foo.stepaicF <- stepAIC(foo.null, scope=list(upper=~0 + Block*year.fac*Month +
    temp.ctr + I(temp.ctr^2)), direction="both")
```

```
## Start:  AIC=15981.75
## cull.f ~ 1
##
##                  Df Deviance   AIC
```

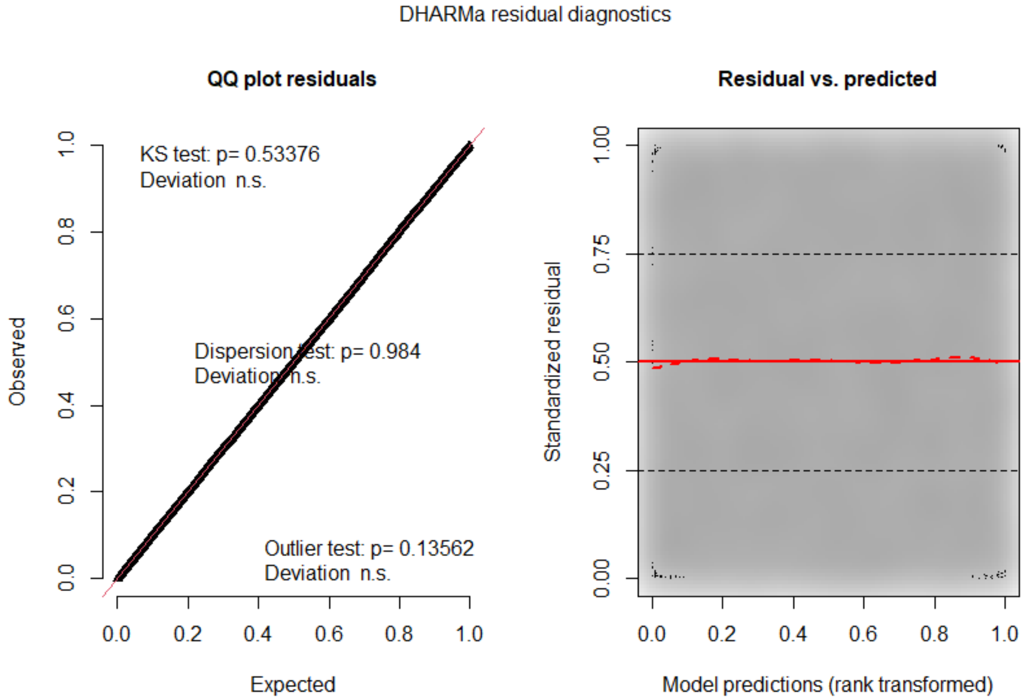
```
## + Month           5    15888 15900
## + temp.ctr        1    15904 15908
## + Block           3    15948 15956
## + year.fac        6    15967 15981
## <none>                 15980 15982
## + I(temp.ctr^2)   1    15978 15982
##
## Step:  AIC=15899.52
## cull.f ~ Month
##
##                  Df Deviance   AIC
## + Block           3    15866 15884
## <none>                 15888 15900
## + temp.ctr        1    15886 15900
## + I(temp.ctr^2)   1    15886 15900
## + year.fac        6    15876 15900
## - Month           5    15980 15982
##
## Step:  AIC=15883.96
## cull.f ~ Month + Block
##
##                  Df Deviance   AIC
## + year.fac        6    15854 15884
## <none>                 15866 15884
## + temp.ctr        1    15865 15885
## + I(temp.ctr^2)   1    15866 15886
## + Block:Month    15    15845 15893
## - Block           3    15888 15900
## - Month           5    15948 15956
##
## Step:  AIC=15883.55
## cull.f ~ Month + Block + year.fac
##
##                   Df Deviance   AIC
## <none>                  15854 15884
## + temp.ctr         1    15852 15884
## - year.fac         6    15866 15884
## + year.fac:Month  30    15794 15884
## + I(temp.ctr^2)    1    15854 15886
## + Block:year.fac  18    15823 15889
## + Block:Month     15    15833 15893
## - Block            3    15876 15900
## - Month            5    15935 15955
```

Check residuals plots …

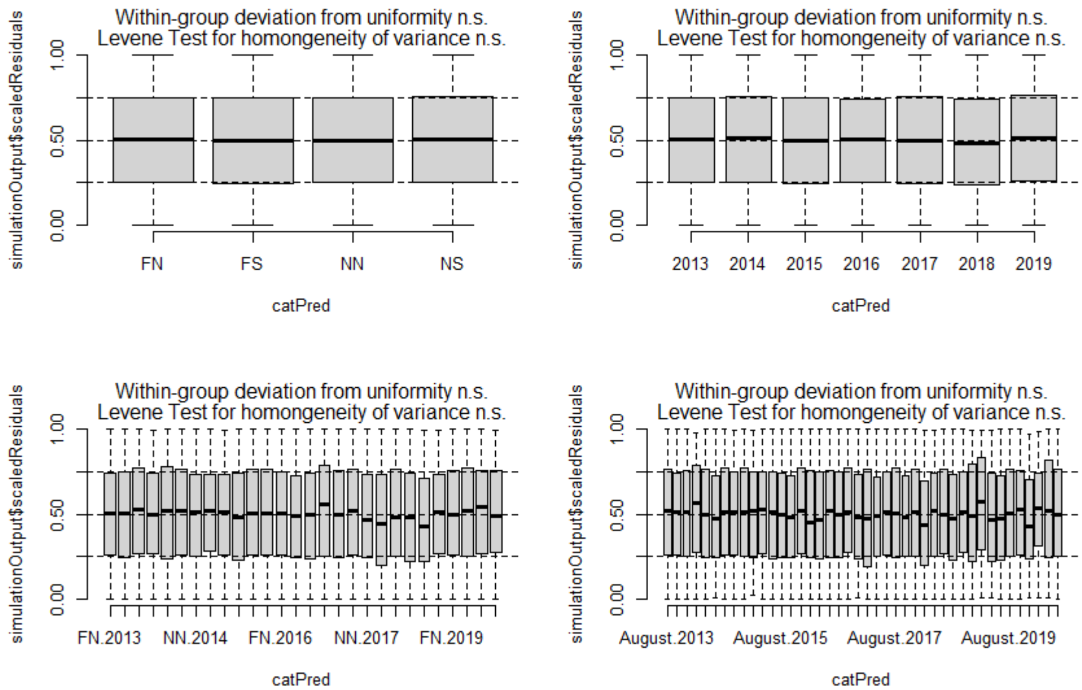
```
#check with DHARMa
 testDispersion(foo.stepaicF)
```



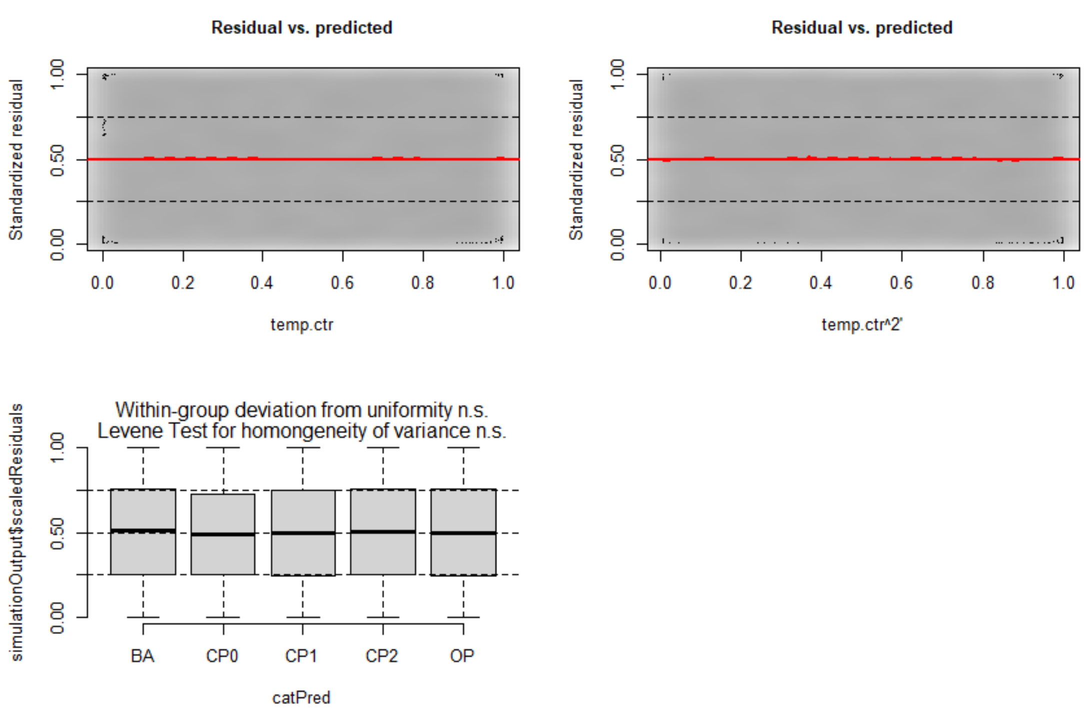

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 1, p-value = 0.912
## alternative hypothesis: two.sided

simulationOutput <- simulateResiduals(fittedModel = foo.stepaicF, plot = T, n=500)
```




```
#check for model misfit. plot residuals against all predictors (both in and out of mo
del)
 which.dat2 <- which.dat %>% filter(Sex==2)
par(mfrow=c(2,2))
 plotResiduals(foo.stepaicF, form=factor(which.dat2$Block))
 plotResiduals(foo.stepaicF, form=which.dat2$year.fac)
 plotResiduals(foo.stepaicF, form=interaction(which.dat2$Block,which.dat2$year.fac))
 plotResiduals(foo.stepaicF, form=interaction(which.dat2$Month,which.dat2$year.fac))
```




```
par(mfrow=c(2,2));
 plotResiduals(foo.stepaicF, form=which.dat2$temp.ctr, xlab="temp.ctr")
 plotResiduals(foo.stepaicF, form=I(which.dat2$temp.ctr^2), xlab="temp.ctr^2'")
 plotResiduals(foo.stepaicF, form=factor(which.dat2$Period))
```



Finalize model and show results for Female Culls …

```r
lobsters7.FishRes.CullF.glm <- glm(formula = cull.f ~ Month + year.fac +
 Block, family = binomial(link = "logit"), data = lobsters7.cull.bySurvey %>%
    filter(Sex==2))
summary(lobsters7.FishRes.CullF.glm)
```

```
##
## Call:
## glm(formula = cull.f ~ Month + year.fac + Block, family = binomial(link = "logit")
,
##     data = lobsters7.cull.bySurvey %>% filter(Sex == 2))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5569  -0.4504  -0.4179  -0.3706   2.4322
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.38395    0.07040 -33.864  < 2e-16 ***
## MonthJuly      -0.28596    0.06603  -4.331 1.49e-05 ***
## MonthJune      -0.39276    0.08528  -4.606 4.11e-06 ***
## MonthMay       -0.28365    0.11903  -2.383 0.017176 *
## MonthOctober    0.22835    0.06723   3.397 0.000682 ***
## MonthSeptember  0.09152    0.06039   1.516 0.129601
## year.fac2014    0.05615    0.07892   0.712 0.476767
## year.fac2015   -0.12777    0.07857  -1.626 0.103928
## year.fac2016    0.08036    0.06718   1.196 0.231584
## year.fac2017   -0.08290    0.07927  -1.046 0.295669
## year.fac2018   -0.08881    0.09142  -0.971 0.331323
## year.fac2019    0.02896    0.08862   0.327 0.743827
## BlockFS         0.02621    0.05597   0.468 0.639530
## BlockNN         0.20311    0.06795   2.989 0.002796 **
## BlockNS         0.29004    0.07626   3.804 0.000143 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15980  on 27164  degrees of freedom
## Residual deviance: 15854  on 27150  degrees of freedom
## AIC: 15884
##
## Number of Fisher Scoring iterations: 5
```