DO Prediction Model

Cross-validation, and Outlier Scrutiny

Bay Oxygen Research Group

12 September, 2022

Presented by

Elgin Perry

Model Development Team

Peter Tango, Gary Shenk, Rebecca Murphy, Isabella Bertani, Breck Sullivan

Jim Hagy, Jon Harcum

Outline:

Review of Test data and Model

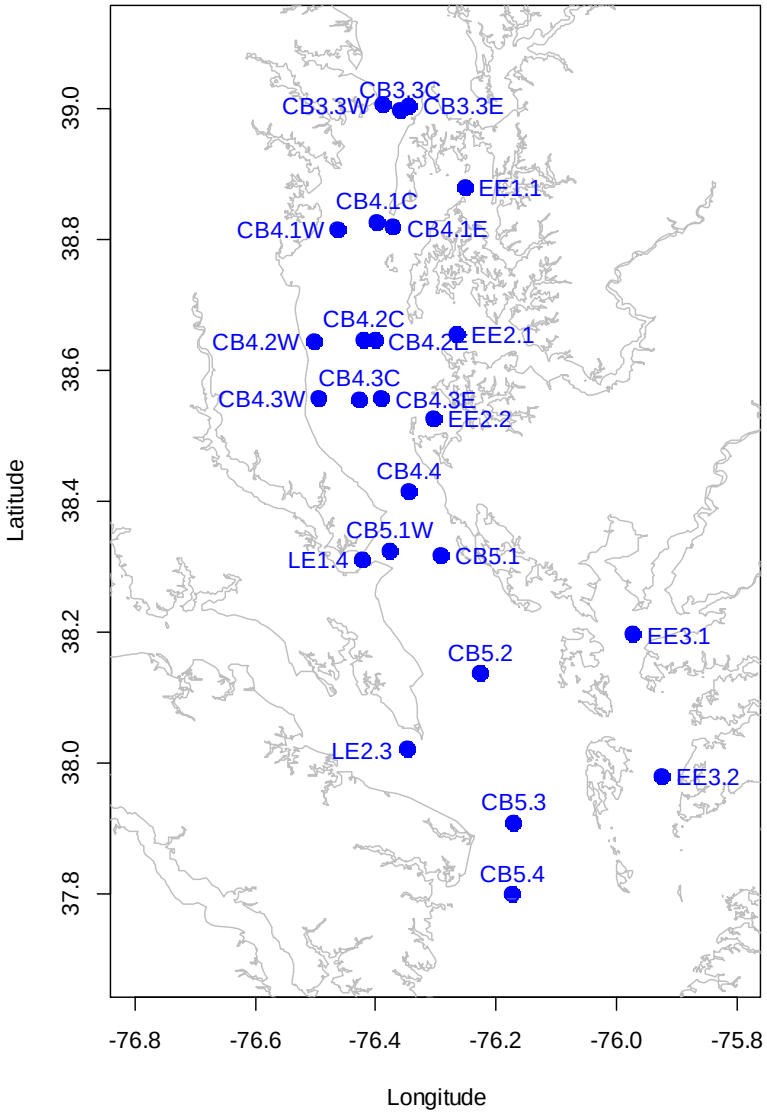
Cross Validation method and results

Plots for assessing regions of poor fit

Trials with increasing flexibility of smoothing functions

Trial tests of Tweedie Distribution

Map of stations in the test region. Test Period: 1990-2010, monthly and twice monthly observations.



Using Variable Selection methods, a prototype model (gs6a) with the terms shown below was fitted to the test data.

Summary of DO model runs in model development.

model	Term Added	rSquare	AIC	rmse	maxPres	maxNres	FitTime
gs1	cyear,doy	0.497	568892.9	2.63	12.2	10.2	0.98
gs2	wDepth	0.77	475598.4	1.78	9.5	8.8	1.34
gs3	LonKm	0.792	463792.4	1.69	10	8.5	1.72
gs4a	wDepth	0.794	462690.1	1.68	9.9	8.5	2.26
gs4b	LatKm	0.793	462895	1.68	9.6	8.3	2.21
gs4c	bDepth & LatKm	0.795	462121.3	1.68	9.5	8.4	2.67
gs5	wDepth*doy	0.835	435780	1.5	8.3	9.1	3.49
gs5a	all two*way	0.857	419385.1	1.4	7.5	10.5	18.33
gs6	wDepth*doy*cyear	0.858	418346.3	1.4	7.5	10.4	55.46
gs6a	wDepth*doy*LonKm.	0.859	417555.3	1.39	7.5	10.5	60.19

Rebecca Proposed New Model (gs7):

```
Approximate significance of smooth terms:
                           edf Ref.df         F p-value
s(decimal date)         18.956 19.000   121.920  <2e-16 ***
s(day of year)           7.970  8.000 48806.475  <2e-16 ***
s(water Depth)           8.410  8.702  1696.372  <2e-16 ***
s(Estuarine Longitude)   8.454  8.610   568.577  <2e-16 ***
s(Estuarine Latitude)    1.000  1.000     2.263   0.132
s(bottom Depth)          8.998  8.999    16.625  <2e-16 ***
ti(LatKm,wDepth,LonKm) 60.862 61.000   239.602  <2e-16 ***
ti(wDepth,bDepth)        9.722 10.361    41.268  <2e-16 ***
ti(wDepth,ddate)        12.421 14.340    14.721  <2e-16 ***
ti(LonKm,ddate)         14.071 15.335    13.521  <2e-16 ***
ti(LatKm,ddate)         13.572 14.981    10.724  <2e-16 ***
ti(bDepth,ddate)        14.185 15.424     6.001  <2e-16 ***
ti(wDepth,doy)          11.724 12.000  2432.797  <2e-16 ***
ti(LonKm,doy)           11.804 12.000   176.734  <2e-16 ***
ti(LatKm,doy)           11.888 12.000   130.902  <2e-16 ***
ti(bDepth,doy)          11.450 12.000    25.453  <2e-16 ***
ti(ddate,doy)           11.917 12.000   162.919  <2e-16 ***
---
R-sq.(adj) =  0.863 > AIC(gam7) = 414184.2
```

Creating Cross-Validation (CV) data

- **Trial uses mid-bay data 1990-2010 as in previous trials**
- **Treat each station-date as an event**
- **Identify months with two or more events**
- **In these months, randomly choose one event for CV**
- **Retain others for Training (TR).**

Designed to address the question: If we have monthly data, how well can we predict at dates in between observed dates. Cross-validation in space will be addressed in the future.

Training vs Validation distributions by Station

Station	Training Count	Validation Count	Training Percent	Validation Percent	Sum Count
CB3.3C	5208	2242	69.91	30.09	7450
CB3.3E	1173	750	61	39	1923
CB3.3W	1244	774	61.65	38.35	2018
CB4.1C	5875	2684	68.64	31.36	8559
CB4.1E	3074	2016	60.39	39.61	5090
CB4.1W	1203	777	60.76	39.24	1980
CB4.2C	5043	2341	68.3	31.7	7384
CB4.2E	1227	753	61.97	38.03	1980
CB4.2W	1183	764	60.76	39.24	1947
CB4.3C	5003	2300	68.51	31.49	7303
CB4.3E	2927	1892	60.74	39.26	4819
CB4.3W	1196	754	61.33	38.67	1950
CB4.4	5545	2510	68.84	31.16	8055
CB5.1	6115	2672	69.59	30.41	8787
CB5.1W	1135	639	63.98	36.02	1774
CB5.2	5298	2482	68.1	31.9	7780
CB5.3	4670	2059	69.4	30.6	6729
CB5.4	5392	1645	76.62	23.38	7037
EE1.1	3095	854	78.37	21.63	3949
EE2.1	1985	535	78.77	21.23	2520
EE2.2	2900	818	78	22	3718
EE3.1	3103	736	80.83	19.17	3839
EE3.2	3892	1009	79.41	20.59	4901
LE1.4	1622	932	63.51	36.49	2554
LE2.3	3580	1657	68.36	31.64	5237
Sum	82688	36595	69.32	30.68	119283

Training vs Validation distributions by Month.

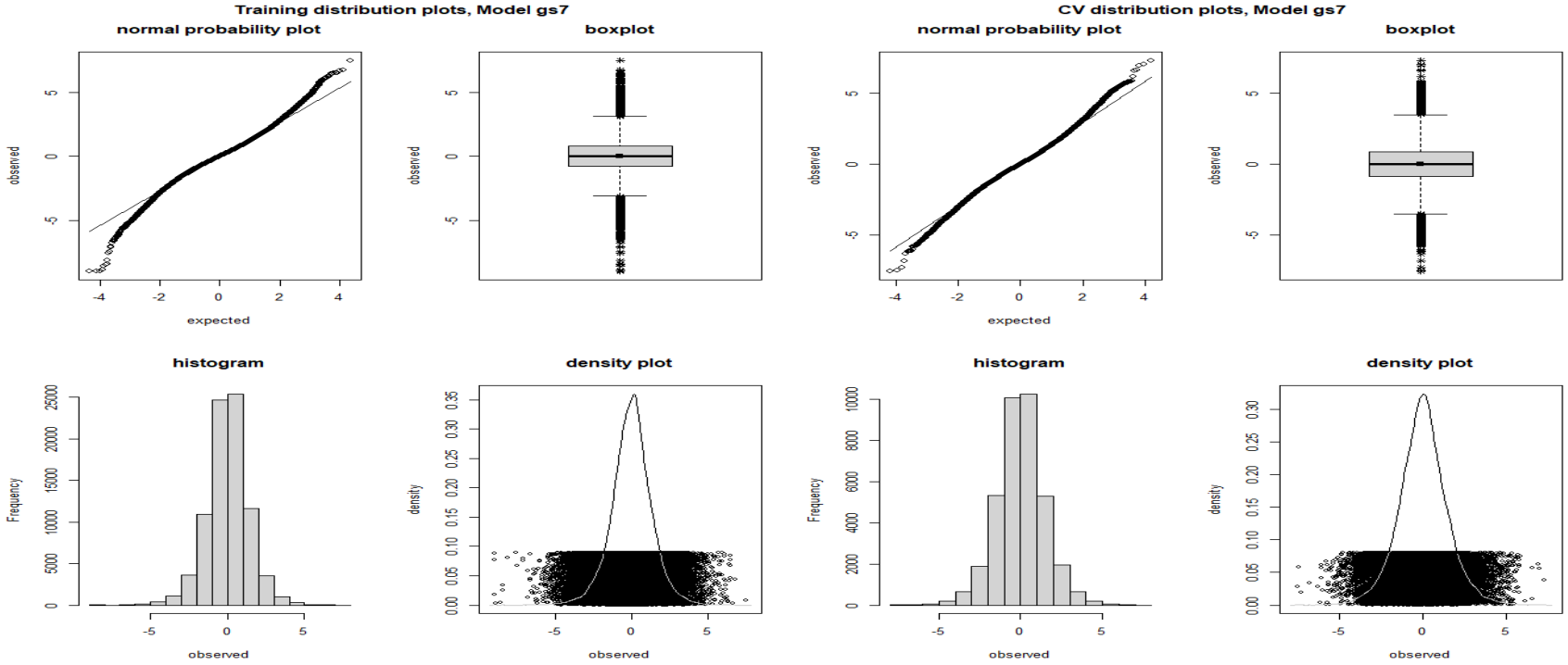
Month	Training Count	Validation Count	Training Percent	Validation Percent	Sum Count
1	5225	18	99.66	0.34	5243
2	5281	47	99.12	0.88	5328
3	7602	2082	78.5	21.5	9684
4	8041	5972	57.38	42.62	14013
5	7878	6310	55.53	44.47	14188
6	7935	4251	65.12	34.88	12186
7	7800	6537	54.4	45.6	14337
8	7862	6872	53.36	46.64	14734
9	7433	2457	75.16	24.84	9890
10	6853	1989	77.51	22.49	8842
11	5333	48	99.11	0.89	5381
12	5445	12	99.78	0.22	5457
Sum	82688	36595	69.32	30.68	119283

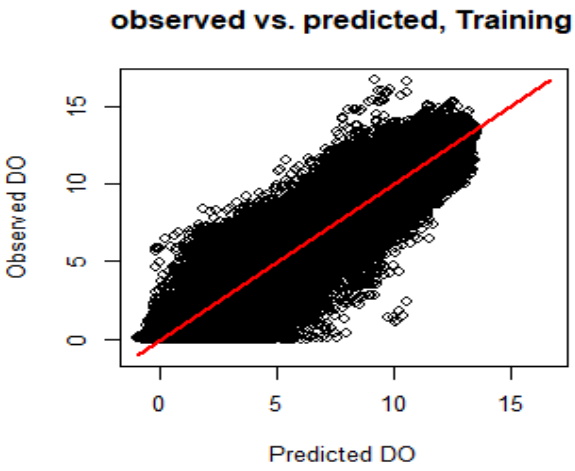
gs7: Model Developed by Rebecca Murphy.

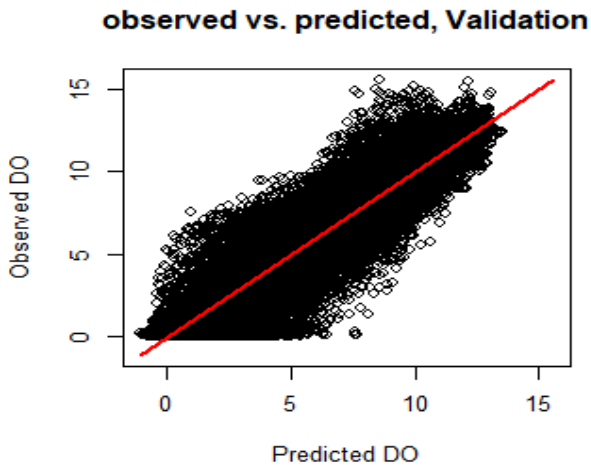
```
Approximate significance of smooth terms:
                           edf Ref.df          F p-value
s(cyear)                18.920 18.999    75.503 < 2e-16 ***
s(doy)                   7.968  8.000 37762.096 < 2e-16 ***
s(wDepth)                8.219  8.719   935.336 < 2e-16 ***
s(LonKm)                 8.691  8.813   417.221 < 2e-16 ***
s(LatKm)                 1.442  1.618    14.965 0.00028 ***
s(bDepth)                5.735  6.038     6.691 < 2e-16 ***
ti(LatKm,wDepth,LonKm) 61.000 61.000   167.430 < 2e-16 ***
ti(wDepth,bDepth)        9.975 10.653    22.616 < 2e-16 ***
ti(wDepth,cyear)         9.679 11.785    19.370 < 2e-16 ***
ti(LonKm,cyear)         13.789 15.203    13.075 < 2e-16 ***
ti(LatKm,cyear)         14.169 15.447    12.882 < 2e-16 ***
ti(bDepth,cyear)        14.932 15.777     7.745 < 2e-16 ***
ti(wDepth,doy)          11.747 12.000  1824.129 < 2e-16 ***
ti(LonKm,doy)           11.731 12.000   109.628 < 2e-16 ***
ti(LatKm,doy)           11.786 12.000    93.414 < 2e-16 ***
ti(bDepth,doy)          11.606 12.000    19.604 < 2e-16 ***
ti(cyear,doy)           11.898 12.000   122.548 < 2e-16 ***
---
```

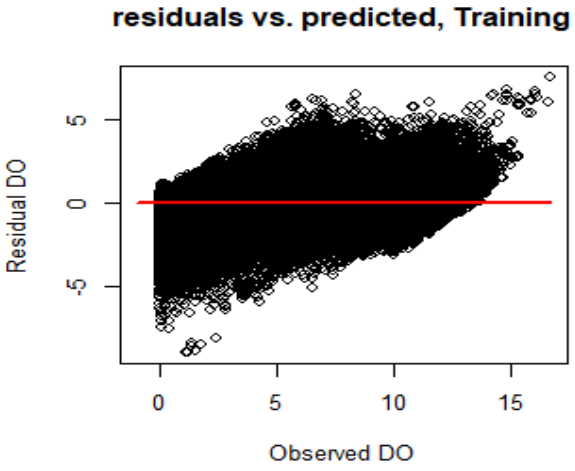
Statistic	Training	Validation
Route mean square error (rmse)	1.3354	1.4622
Median absolute deviation (mad)	0.7766	0.8755
R-square (rsq)	0.8656	0.838

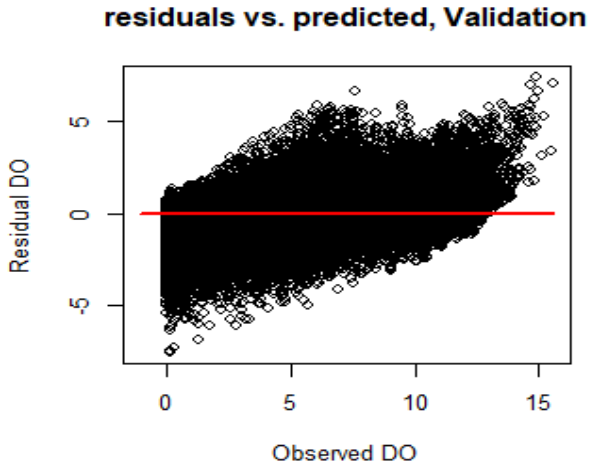
Training vs. Validation Residual plots




observed vs. predicted, Training
Observed DO
Predicted DO
0
5
10
15


observed vs. predicted, Validation
Observed DO
Predicted DO
0
5
10
15

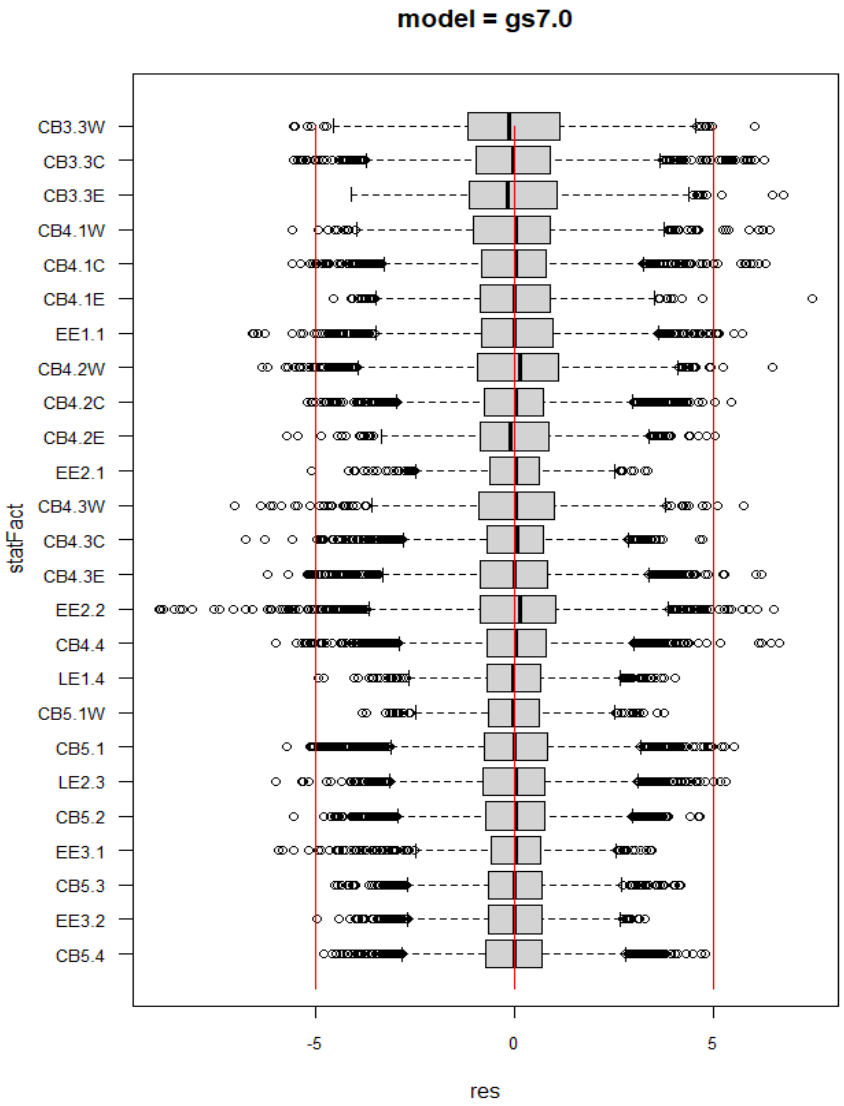

residuals vs. predicted, Training
Residual DO
Observed DO
-5
0
5
10
15

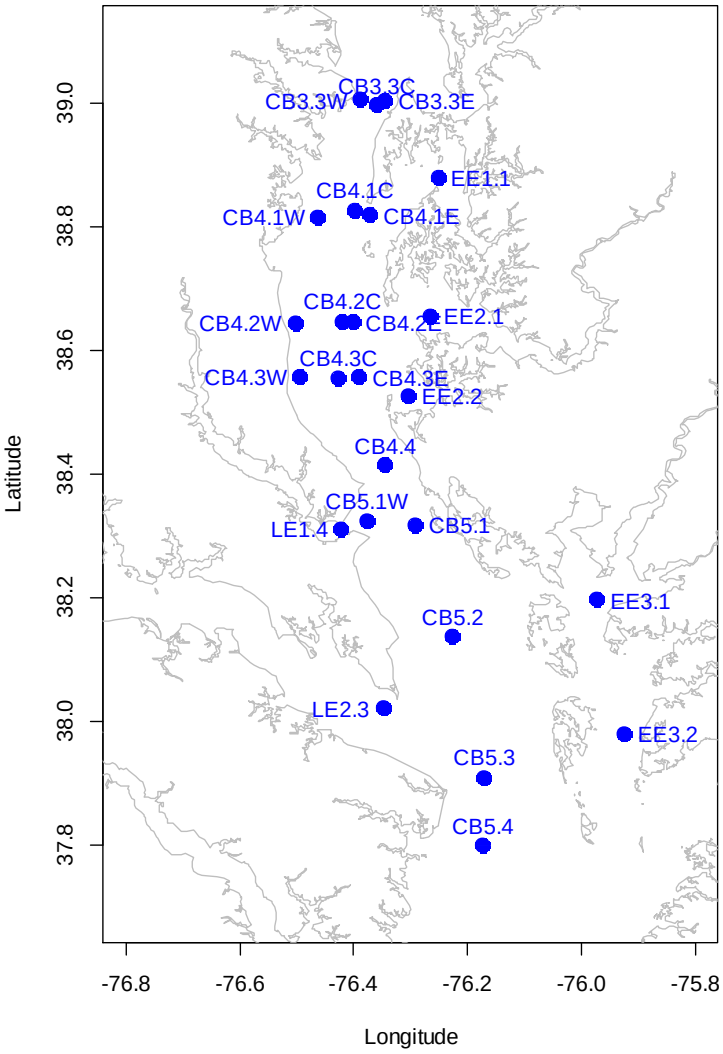

residuals vs. predicted, Validation
Residual DO
Observed DO
-5
0
5
10
15

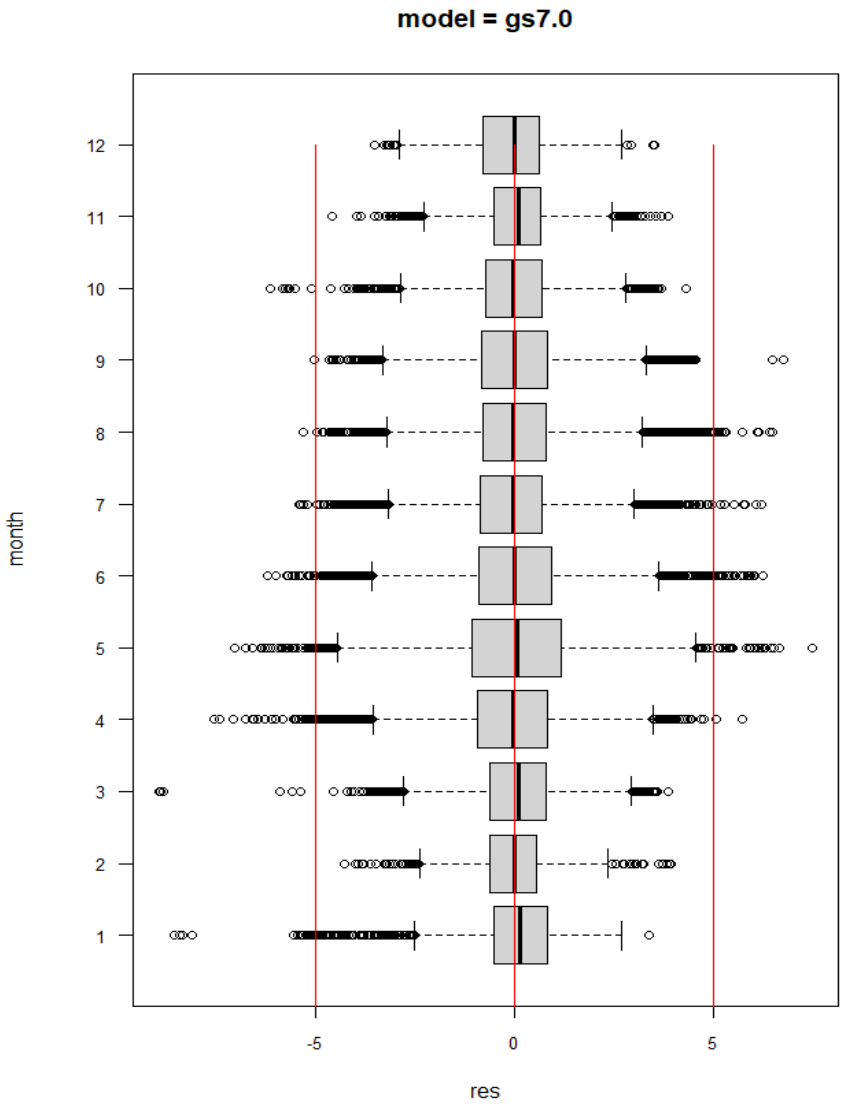
Training vs. Cross-Validation Summary of DO model runs

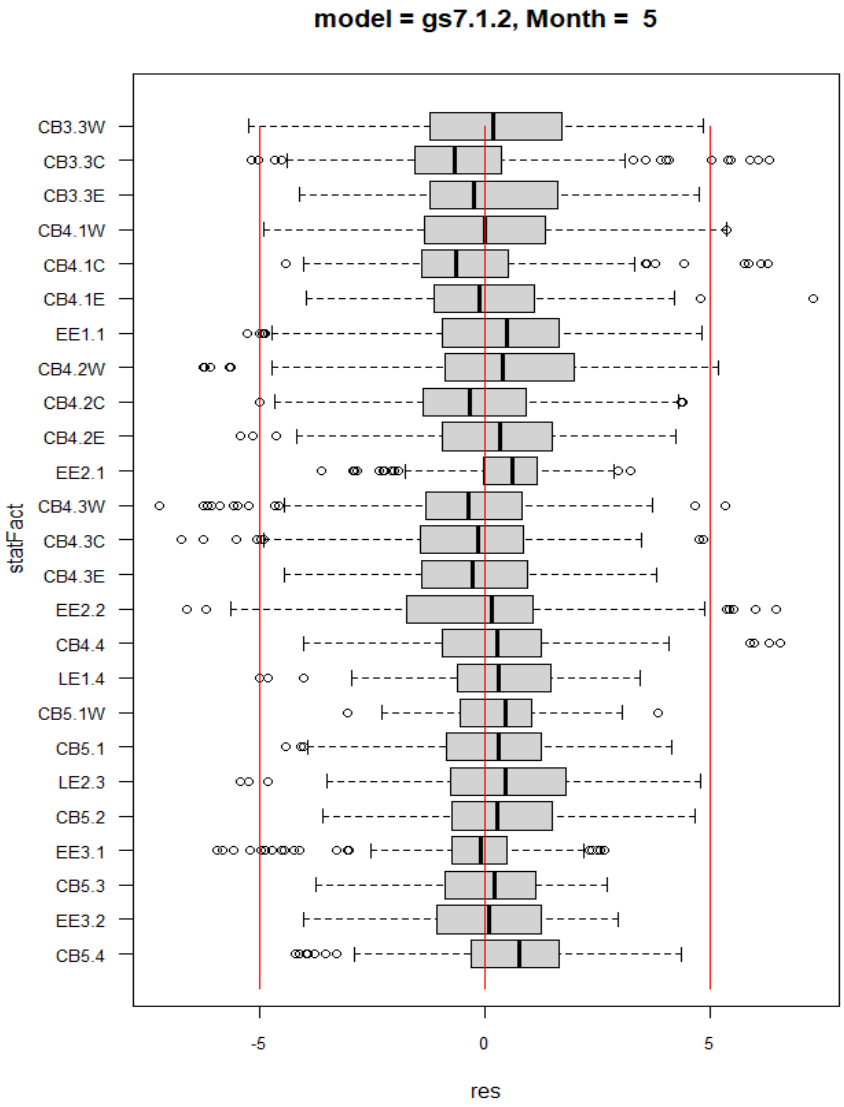
model	termAdd	TR_ rSquare	CV_ rSquare	TR_ rmse	CV_ rmse	TR_ mad	CV_ mad
gs1	cyear,doy	0.537	0.348	2.48	2.93	1.75	2.52
gs2	wDepth	0.77	0.748	1.78	1.82	1.28	1.31
gs3	LonKm	0.792	0.771	1.69	1.74	1.18	1.2
gs4a	wDepth	0.794	0.773	1.68	1.73	1.17	1.2
gs4b	LatKm	0.793	0.773	1.68	1.73	1.18	1.21
gs4c	bDepth & LatKm	0.795	0.774	1.68	1.73	1.17	1.2
gs5	wDepth*doy	0.835	0.811	1.5	1.58	0.91	0.95
gs5a	all two*way [1]	0.857	0.836	1.4	1.47	0.83	0.89
gs6	wDepth*doy*cyear	0.858	0.837	1.4	1.47	0.82	0.88
gs6a	wDepth*doy*LonKm.	0.859	0.838	1.39	1.46	0.82	0.87
gs7	Murphy	0.863	0.844	1.37	1.44	0.8	0.85

1 – not really


model = gs7.0
statFact
CB3.3W
CB3.3C
CB3.3E
CB4.1W
CB4.1C
CB4.1E
EE1.1
CB4.2W
CB4.2C
CB4.2E
EE2.1
CB4.3W
CB4.3C
CB4.3E
EE2.2
CB4.4
LE1.4
CB5.1W
CB5.1
LE2.3
CB5.2
EE3.1
CB5.3
EE3.2
CB5.4
-5
0
5
res


Latitude
39.0
38.8
38.6
38.4
38.2
38.0
37.8
CB3.3C
CB3.3W
CB3.3E
EE1.1
CB4.1C
CB4.1W
CB4.1E
CB4.2C
CB4.2W
CB4.2E
EE2.1
CB4.3C
CB4.3W
CB4.3E
EE2.2
CB4.4
CB5.1W
LE1.4
CB5.1
EE3.1
CB5.2
LE2.3
EE3.2
CB5.3
CB5.4
-76.8
-76.6
-76.4
-76.2
-76.0
-75.8
Longitude


model = gs7.0
month
12
11
10
9
8
7
6
5
4
3
2
1
-5
0
5
res


model = gs7.1.2, Month = 5
statFact
CB3.3W
CB3.3C
CB3.3E
CB4.1W
CB4.1C
CB4.1E
EE1.1
CB4.2W
CB4.2C
CB4.2E
EE2.1
CB4.3W
CB4.3C
CB4.3E
EE2.2
CB4.4
LE1.4
CB5.1W
CB5.1
LE2.3
CB5.2
EE3.1
CB5.3
EE3.2
CB5.4
-5
0
5
res

gs7 –
gs7.1.2 - ti(cyear,doy,bs=c("tp","cc"), k=24)
gs7.1.3 - ti(wDepth,LonKm,bs=c("tp","tp"), k=15)
gs7.1.4 - ti(LonKm,cyear, k=15)

gmLab	AIC	TRrmse	CVrmse	TRmad	CVmad	TRrsq	CVrsq	ex.time
gs7	282725.6168	1.3354	1.4622	0.7766	0.8755	0.8656	0.838	1.25
gs7.1.2	260182.1894	1.1613	1.3761	0.6524	0.8166	0.8984	0.857	38.4
gs7.1.3	260150.1774	1.1609	1.3757	0.6524	0.8178	0.8984	0.857	75.5
gs7.1.4	Failed[1] to run							

1:Error in magic(Gy, GX, msp, GS, Goff, L = G$L, lsp0 = G$lsp0, G$rank, :

'Calloc' could not allocate memory (47444544 of 8 bytes)

Trial of using tweedie distribution in place of Gaussian (normal) distribution.
Used gs7 with a change of family.

```
initTime <- Sys.time()
> modl <- "gs8"
> gs8<- gam(do~s(cyear,k=20)
      +s(doy,bs="cc")+s(wDepth)+s(LonKm)+s(LatKm)+s(bDepth)
      +ti(LatKm,wDepth,LonKm)+ti(wDepth,bDepth)
      +ti(wDepth,cyear)+ti(LonKm,cyear) +ti(LatKm,cyear)+ti(bDepth,cyear)
      +ti(wDepth,doy,bs=c("tp","cc"))+ti(LonKm,doy,bs=c("tp","cc"))
      +ti(LatKm,doy,bs=c("tp","cc"))+ti(bDepth,doy,bs=c("tp","cc"))
      +ti(cyear,doy,bs=c("tp","cc"))
      ,data=mbdo,family=tw)
> finiTime <- Sys.time()
>
> (gs8.Time <- finiTime-initTime)
Time difference of 4.953199 hours
```

(End of Presentation)

Extra Slides:

Slides from here down may help with fielding questions

```
#comparison of terms between gs6a and gs7
#
#  s(cyear, k=20) +                                    s(cyear,k=20)+                    1
#  s(doy,bs='cc') +                                    s(doy,bs="cc")+                   2
#  s(wDepth) +                                         s(wDepth)+                        3
#  s(LonKm) +                                          s(LonKm)+                         4
#  s(bDepth) +                                         s(bDepth) +                       6
#  s(LatKm) +                                          s(LatKm)+                         5
#  ti(wDepth,doy,bs=c("tp","cc"))        +             ti(wDepth,doy,bs=c("tp","cc"))+ 13
#  ti(LonKm,wDepth,bs=c("tp","tp")) +
#  ti(LonKm,doy,bs=c("tp","cc"))    +                  ti(LonKm,doy,bs=c("tp","cc")) + 14
#  ti(cyear,doy,bs=c('tp','cc'))         +             ti(cyear,doy,bs=c("tp","cc"))   17
#  ti(wDepth,bDepth,bs=c("tp","tp"))+                  ti(wDepth,bDepth) +               8
#  ti(cyear,wDepth,bs=c("tp","tp"))      +             ti(wDepth,cyear)+                 9
#  ti(LonKm,cyear,bs=c("tp","tp")) +                   ti(LonKm,cyear) +                10
#  ti(wDepth,doy,LonKm,bs=c("tp","cc","tp"))+
#  ti(wDepth,doy,cyear,bs=c("tp","cc","tp"))
#                                                    ti(LatKm,wDepth,LonKm)+           7
#                                                    ti(LatKm,cyear)+                 11
#                                                    ti(bDepth,cyear) +               12
#                                                    ti(LatKm,doy,bs=c("tp","cc"))+   15
#                                                    ti(bDepth,doy,bs=c("tp","cc")) + 16
```