Simulation of imputation of censored values LateJuly v1

Marc Roddis

7/15/2020

Finding appropriate simulation parameters from observed data

We created the test dataset `testdata_cen_omit` from the original observed data `pcb.csv` by omitting all missing values of `CB28` and `CB153` , removing all observations except those from herring species, removing all observations prior to 1989, re-indexing 1989 as "year zero", removing all variables except `YEAR` , `CB28` and `CB153` , omitting all censored observations, and replacing concentrations with log-concentrations.

Fitting linear models to the test data gave the following fixed parameters:

$$CB153 = -2.91 - 0.02 * YEAR$$

$$CB28 = -3.18 + 0.79 * CB153$$

$$sd(CB28) = sd(CB153) = 0.1$$

We will use parameters values estimated from real data for our first simulation to study the effectiveness of various methods of dealing with censored data. In subsequent studies, we will investigate how generally applicable these methods are for various other possible choices of parameter values. We will use logarithmised concentrations for CB28 and CB153 and refer to these as cb28 and cb153 respectively throughout.

100 values for cb153 per year, for 15 years, were generated and denoted as `cb153` from

$$CB153 = -2.91 - 0.02 * YEAR$$

with added noise (modeled with normal distribution with mean = 0 and sd = 0.1).

From every such CB153 value, the corresponding value for CB28 was generated from

$$CB28 = -3.18 + 0.79 * CB153$$

, again with added noise (modeled with normal distribution with mean = 0 and sd = 0.1). From these equations, we deduce that `true_beta28year` = 0.79 * -0.02; we will use this as the "true" value against which we evaluate the estimates for this parameter from applying various methods to censored data.

From real data for the 15 year period 2003-2017, 34 % of the cb28 values were censored, so we will use the parameter value `cprop=0.34` in this first simulation study. Values of cb28 below the value below the level of detection (LOD) were then censored. The LOD was calculated from the cprop*100th percentile of the simulated data at each iteration.

Applying and evaluating censoring methods

The regression coefficient `beta28year` for CB28 ~ YEAR was estimated by generating simulated datasets and applying five different methods to the censored values, and then estimating `beta28year` by fitting a linear model to the resulting datasets from each method. The methods were:

`omit` means censored values were omitted.

`subst2` means censored values were substituted with $\frac{LOD}{\sqrt(2)}$.

`subst1` means censored values were substituted with $\frac{LOD}{\sqrt{(1)}} = LOD$.

`subst4` means censored values were substituted with $\frac{LOD}{\sqrt{(4)}} = \frac{LOD}{2}$.

`censReg1` means censored values were imputed using the `censReg()` function from the `censReg` package using 1 predictor variables (cb153). The censreg MLE estimates for `beta28year` and the residual standard errors were then fed as mean and standard deviation respectively into the `etruncnorm()` function from the `truncnorm` package, from which every censored value was subsituted with the corresponding imputed value.

`censReg2` means censored values were imputed as described for `censReg1` , except that two predictor variables (cb153 and year) were used instead

`censReg1naive` and `censReg2naive` are the same as `censReg1` and `censReg2` respectively, except that a non-truncated normal distribution was used instead. This was done to check that we get a more biased estimate because it is possible that the imputed values are above LOD, despite the fact that the censored value are below LOD.

`censReg0impute` estimates `beta28year` directly from the MLE value generated by the `censReg()` function; no imputation is done at all in this method.

Each method for acting upon the censored data was then applied, then `beta28year` was estimated for each method. The MSE, squared-bias and variance for each estimate of `beta28year` was then reported and used to evaluate the censoring methods.

```
##                mse_beta bias_beta variance_beta
## censReg1         48.293     0.001        48.340
## censReg1year     48.612     0.000        48.661
## censReg2         48.707     0.000        48.756
## censReg0impute   48.612     0.000        48.661
## best             47.718     0.000        47.766
```

```
##                mse_beta bias_beta variance_beta
## censReg1         74.680     0.039        74.716
## censReg1year    102.894     0.175       102.822
## censReg2         96.849     0.129        96.816
## censReg0impute  102.894     0.175       102.822
## best             47.718     0.000        47.766
```

```
##                mse_beta bias_beta variance_beta
## censReg1        664.952     3.803       661.811
## censReg1year    676.973     3.983       673.664
## censReg2        676.917     3.933       673.658
## censReg0impute  676.973     3.983       673.664
## best            776.117     1.682       775.210
```

```
##                mse_beta bias_beta variance_beta
## censReg1        766.589     0.451       766.905
## censReg1year   1134.744     0.429      1135.451
## censReg2       1134.171     0.386      1134.921
## censReg0impute 1134.744     0.429      1135.451
## best            776.117     1.682       775.210
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        1215.581     0.622      1216.176
## censReg1year    1366.583     1.344      1366.606
## censReg2        1373.463     1.342      1373.494
## censReg0impute 1366.583     1.344      1366.606
## best              50.437     0.012        50.475
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        1078.619     0.006      1079.693
## censReg1year    1227.274     0.092      1228.410
## censReg2        1226.749     0.035      1227.943
## censReg0impute 1227.274     0.092      1228.410
## best              50.437     0.012        50.475
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        1231.826     3.463      1229.593
## censReg1year    1399.866     4.164      1397.099
## censReg2        1399.188     4.479      1396.105
## censReg0impute 1399.866     4.164      1397.099
## best              49.773     0.016        49.807
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        1159.193     2.132      1158.220
## censReg1year    1328.481     2.372      1327.436
## censReg2        1316.508     2.418      1315.405
## censReg0impute 1328.481     2.372      1327.436
## best              49.773     0.016        49.807
```

```
##                 mse_beta bias_beta variance_beta
## censReg1          51.168     0.010        51.209
## censReg1year      51.949     0.006        51.995
## censReg2          50.718     0.010        50.758
## censReg0impute    51.949     0.006        51.995
## best              49.749     0.137        49.662
```

```
##                 mse_beta bias_beta variance_beta
## censReg1         348.114     0.034       348.429
## censReg1year     504.412     0.033       504.883
## censReg2         377.791     0.132       378.037
## censReg0impute   504.412     0.033       504.883
## best              49.749     0.137        49.662
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        1565.358     1.829      1565.093
## censReg1year    1619.639     2.429      1618.829
## censReg2        1616.207     2.346      1615.476
## censReg0impute 1619.639     2.429      1618.829
## best              49.643     0.001        49.692
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       1556.209     0.127      1557.639
## censReg1year   1620.456     0.083      1621.995
## censReg2       1619.807     0.135      1621.294
## censReg0impute 1620.456     0.083      1621.995
## best             49.643     0.001        49.692
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       2250.367  2029.186       221.203
## censReg2       2250.120  2028.867       221.275
## censReg0impute 2250.228  2021.883       228.368
## best           2248.537  2022.076       226.484
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       2252.342  2021.636       230.730
## censReg2       2253.258  2021.613       231.668
## censReg0impute 2253.409  2021.897       231.535
## best           2251.415  2020.999       230.439
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       35949.97  32350.74      3599.590
## censReg2       35950.90  32364.72      3586.537
## censReg0impute 35957.69  32368.79      3589.264
## best           35949.88  32351.73      3598.511
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       35961.97  32336.92      3625.409
## censReg2       35962.49  32336.56      3626.298
## censReg0impute 35966.66  32336.30      3630.720
## best           35952.88  32340.93      3612.305
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       575172.4  517677.6      57500.55
## censReg2       575176.7  518001.3      57181.11
## censReg0impute 575183.7  517535.4      57654.10
## best           575172.5  517645.1      57533.11
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       575193.0  517542.0      57656.73
## censReg2       575193.3  517542.0      57657.07
## censReg0impute 575204.1  517498.5      57711.31
## best           575175.5  517670.0      57511.27
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       16.61423   0.00045      16.63041
## censReg2       16.76064   0.00076      16.77666
## censReg0impute 17.51873   0.00000      17.53627
## best           18.00498   0.02886      17.99411
```

```
##                mse_beta bias_beta variance_beta
## censReg1       26.43431   0.21090      26.24966
## censReg2       24.37459   0.23932      24.15943
## censReg0impute 27.22961   0.00008      27.25679
## best           18.00498   0.02886      17.99411
```

```
##                mse_beta bias_beta variance_beta
## censReg1       55.52362   0.82786      54.75051
## censReg2       57.67834   0.76642      56.96889
## censReg0impute 65.18440   0.09533      65.15422
## best           18.00498   0.02886      17.99411
```

```
##                mse_beta bias_beta variance_beta
## censReg1       17.69652   0.00517      17.70906
## censReg2       17.79374   0.00466      17.80688
## censReg0impute 19.68858   0.00507      19.70320
## best           18.50515   0.00130      18.52237
```

```
##                mse_beta bias_beta variance_beta
## censReg1       19.07749   0.01452      19.08205
## censReg2       25.54180   0.05757      25.50975
## censReg0impute 55.14101   0.08333      55.11280
## best           18.50515   0.00130      18.52237
```

```
##                  mse_beta bias_beta variance_beta
## censReg1       1103.73316 155.67633     949.00583
## censReg2        145.74558   1.58249     144.30739
## censReg0impute  279.95123   1.34337     278.88674
## best             18.50515   0.00130      18.52237
```

```
##                mse_beta bias_beta variance_beta
## censReg1       16.61423   0.00045      16.63041
## censReg2       16.76064   0.00076      16.77666
## censReg0impute 17.51873   0.00000      17.53627
## best           18.00498   0.02886      17.99411
```


```
##                mse_beta bias_beta variance_beta
## censReg1       26.43431   0.21090      26.24966
## censReg2       24.37459   0.23932      24.15943
## censReg0impute 27.22961   0.00008      27.25679
## best           18.00498   0.02886      17.99411
```

```
##                mse_beta bias_beta variance_beta
## censReg1       55.52362   0.82786      54.75051
## censReg2       57.67834   0.76642      56.96889
## censReg0impute 65.18440   0.09533      65.15422
## best           18.00498   0.02886      17.99411
```

```
##                mse_beta bias_beta variance_beta
## censReg1       19.14784   0.00197      19.16503
## censReg2       19.13053   0.00187      19.14781
## censReg0impute 19.84716   0.01193      19.85508
## best           18.66762   0.02995      18.65633
```

```
##                mse_beta bias_beta variance_beta
## censReg1       47.54019   0.55633      47.03089
## censReg2       39.82510   0.27998      39.58471
## censReg0impute 66.32982   0.37943      66.01641
## best           18.66762   0.02995      18.65633
```

```
##                mse_beta bias_beta variance_beta
## censReg1       30.23114   0.01432      30.24707
## censReg1year   30.77158   0.01726      30.78511
## censReg2       30.30106   0.01623      30.31514
## censReg0impute 30.77158   0.01726      30.78511
## best           26.69474   0.23165      26.48959
```

```
##                mse_beta bias_beta variance_beta
## censReg1       38.91720   0.02219      38.93394
## censReg1year   67.47814   0.03879      67.50685
## censReg2       48.58039   0.00948      48.61952
## censReg0impute 67.47814   0.03879      67.50685
## best           26.69474   0.23165      26.48959
```

```
##                mse_beta bias_beta variance_beta
## censReg1       48.29254   0.00068      48.34020
## censReg1year   48.61222   0.00000      48.66088
## censReg2       48.70748   0.00030      48.75593
## censReg0impute 48.61222   0.00000      48.66088
## best           47.71800   0.00015      47.76561
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        74.68023   0.03853      74.71641
## censReg1year   102.89422   0.17473     102.82231
## censReg2        96.84858   0.12921      96.81618
## censReg0impute 102.89422   0.17473     102.82231
## best            47.71800   0.00015      47.76561
```

```
##                mse_beta bias_beta variance_beta
## censReg1       664.9522    3.8030      661.8110
## censReg1year   676.9731    3.9829      673.6638
## censReg2       676.9170    3.9328      673.6579
## censReg0impute 676.9731    3.9829      673.6638
## best           776.1171    1.6824      775.2099
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        766.5894    0.4513      766.9050
## censReg1year   1134.7440    0.4290     1135.4505
## censReg2       1134.1714    0.3855     1134.9208
## censReg0impute 1134.7440    0.4290     1135.4505
## best            776.1171    1.6824      775.2099
```

```
##                mse_beta bias_beta variance_beta
## censReg1       26.37006   0.00001      26.39644
## censReg1year   27.25066   0.00542      27.27251
## censReg2       26.15604   0.00002      26.18220
## censReg0impute 27.25066   0.00542      27.27251
## best           28.27119   0.01127      28.28820
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        76.47639   0.08393      76.46892
## censReg1year   176.55625   0.55139     176.18104
## censReg2        89.61005   0.26591      89.43358
## censReg0impute 176.55625   0.55139     176.18104
## best            28.27119   0.01127      28.28820
```

```
##                  mse_beta bias_beta variance_beta
## censReg1       1215.58144   0.62189    1216.17573
## censReg1year   1366.58318   1.34364    1366.60615
## censReg2       1373.46263   1.34202    1373.49411
## censReg0impute 1366.58317   1.34364    1366.60614
## best             50.43653   0.01158      50.47542
```

```
##                  mse_beta bias_beta variance_beta
## censReg1       1078.61913   0.00631    1079.69251
## censReg1year   1227.27366   0.09164    1228.41043
## censReg2       1226.74945   0.03467    1227.94272
## censReg0impute 1227.27366   0.09164    1228.41043
## best             50.43653   0.01158      50.47542
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       1231.8263    3.4633     1229.5927
## censReg1year   1399.8662    4.1644     1397.0990
## censReg2       1399.1883    4.4791     1396.1053
## censReg0impute 1399.8662    4.1644     1397.0990
## best             49.7728    0.0157       49.8069
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       1159.1934    2.1319     1158.2197
## censReg1year   1328.4810    2.3721     1327.4364
## censReg2       1316.5080    2.4180     1315.4054
## censReg0impute 1328.4810    2.3721     1327.4364
## best             49.7728    0.0157       49.8069
```

```
##                mse_beta bias_beta variance_beta
## censReg1         20.5111    0.0143       20.5173
## censReg2         20.4769    0.0113       20.4861
## censReg0impute   22.5776    0.0101       22.5900
## best             18.6843    0.0008       18.7022
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       1065.6335  148.5794      917.9721
## censReg2        145.4707    1.6476      143.9671
## censReg0impute  296.9717    0.0096      297.2594
## best             18.6843    0.0008       18.7022
```

```
##                mse_beta bias_beta variance_beta
## censReg1        27.5397    0.0343       27.5329
## censReg1year    28.1721    0.0345       28.1657
## censReg2        27.3513    0.0341       27.3446
## censReg0impute  28.1721    0.0345       28.1657
## best            25.4504    0.0004       25.4755
```

```
##                mse_beta bias_beta variance_beta
## censReg1       145.1626    0.6054      144.7020
## censReg1year   358.6493    1.9699      357.0364
## censReg2       163.2629    0.5836      162.8422
## censReg0impute 358.6493    1.9699      357.0364
## best            25.4504    0.0004       25.4755
```

```
##                mse_beta bias_beta variance_beta
## censReg1        51.1676    0.0097       51.2092
## censReg1year    51.9490    0.0055       51.9955
## censReg2        50.7176    0.0102       50.7581
## censReg0impute  51.9490    0.0055       51.9955
## best            49.7489    0.1368       49.6617
```

```
##                mse_beta bias_beta variance_beta
## censReg1       348.1141    0.0340      348.4285
## censReg1year   504.4116    0.0332      504.8833
## censReg2       377.7914    0.1321      378.0373
## censReg0impute 504.4116    0.0332      504.8833
## best            49.7489    0.1368       49.6617
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       1565.3576    1.8293     1565.0934
## censReg1year   1619.6392    2.4288     1618.8292
## censReg2       1616.2068    2.3460     1615.4763
## censReg0impute 1619.6392    2.4288     1618.8292
## best             49.6435    0.0009       49.6923
```

```
##                 mse_beta bias_beta variance_beta
## censReg1       1556.2089    0.1273     1557.6393
## censReg1year   1620.4563    0.0830     1621.9952
## censReg2       1619.8073    0.1346     1621.2940
## censReg0impute 1620.4563    0.0830     1621.9952
## best             49.6435    0.0009       49.6923
```

```
##                mse_beta bias_beta variance_beta
## censReg1         48.293     0.001        48.340
## censReg1year     48.255     4.790        43.509
## censReg2         48.707     0.000        48.756
## censReg0impute   48.612     0.000        48.661
## censReg1naive    74.552    31.287        43.308
## best             47.718     0.000        47.766
```

```
##                mse_beta bias_beta variance_beta
## censReg1         74.680     0.039        74.716
## censReg1year    701.341   686.088        15.268
## censReg2         96.849     0.129        96.816
## censReg0impute  102.894     0.175       102.822
## censReg1naive   103.778    25.186        78.671
## best             47.718     0.000        47.766
```

```
##                mse_beta bias_beta variance_beta
## censReg1        664.952     3.803       661.811
## censReg1year    676.973     3.983       673.664
## censReg2        676.917     3.933       673.658
## censReg0impute  676.973     3.983       673.664
## censReg1naive   655.698   137.954       518.262
## best            776.117     1.682       775.210
```

```
##                mse_beta bias_beta variance_beta
## censReg1        766.589     0.451       766.905
## censReg1year    774.221   428.935       345.632
## censReg2       1134.171     0.386      1134.921
## censReg0impute 1134.744     0.429      1135.451
## censReg1naive   935.157    90.675       845.328
## best            776.117     1.682       775.210
```

```
##                mse_beta bias_beta variance_beta
## censReg1       1215.581     0.622      1216.176
## censReg1year   2504.906  2187.209       318.015
## censReg2       1373.463     1.342      1373.494
## censReg0impute 1366.583     1.344      1366.606
## censReg1naive  1814.717   267.527      1548.739
## best             50.437     0.012        50.475
```

```
##                mse_beta bias_beta variance_beta
## censReg1       1078.619     0.006      1079.693
## censReg1year   2436.182  2157.551       278.911
## censReg2       1226.749     0.035      1227.943
## censReg0impute 1227.274     0.092      1228.410
## censReg1naive  1680.460   303.159      1378.679
## best             50.437     0.012        50.475
```

```
##                mse_beta bias_beta variance_beta
## censReg1       1231.826     3.463      1229.593
## censReg1year   2355.015  2046.147       309.177
## censReg2       1399.188     4.479      1396.105
## censReg0impute 1399.866     4.164      1397.099
## censReg1naive  1916.112   362.451      1555.216
## best             49.773     0.016        49.807
```

```
##                mse_beta bias_beta variance_beta
## censReg1       1159.193     2.132      1158.220
## censReg1year   2356.127  2063.142       293.279
## censReg2       1316.508     2.418      1315.405
## censReg0impute 1328.481     2.372      1327.436
## censReg1naive  1798.397   346.491      1453.359
## best             49.773     0.016        49.807
```

```
##                mse_beta bias_beta variance_beta
## censReg1         51.168     0.010        51.209
## censReg1year    126.197    77.312        48.934
## censReg2         50.718     0.010        50.758
## censReg0impute   51.949     0.006        51.995
## censReg1naive   100.890    48.042        52.901
## best             49.749     0.137        49.662
```

```
##                 mse_beta bias_beta variance_beta
## censReg1         348.114     0.034       348.429
## censReg1year   21259.942 21231.485        28.485
## censReg2         377.791     0.132       378.037
## censReg0impute   504.412     0.033       504.883
## censReg1naive    374.436    29.037       345.745
## best              49.749     0.137        49.662
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        1565.358     1.829      1565.093
## censReg1year   11862.287 11640.703       221.806
## censReg2        1616.207     2.346      1615.476
## censReg0impute  1619.639     2.429      1618.829
## censReg1naive   2997.939  1052.541      1947.345
## best              49.643     0.001        49.692
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        1556.209     0.127      1557.639
## censReg1year   11984.523 11765.795       218.947
## censReg2        1619.807     0.135      1621.294
## censReg0impute  1620.456     0.083      1621.995
## censReg1naive   2922.282   982.121      1942.103
## best              49.643     0.001        49.692
```

```
##                mse_beta bias_beta variance_beta
## censReg1         53.645     0.020        53.679
## censReg1year    250.311   195.970        54.395
## censReg2         53.327     0.020        53.360
## censReg0impute   54.381     0.026        54.410
## censReg1naive    76.793    21.167        55.682
## best             48.739     0.012        48.775
```

```
##                 mse_beta bias_beta variance_beta
## censReg1         663.331     1.712       662.281
## censReg1year   94666.840 94606.340        60.561
## censReg2         693.720     1.800       692.613
## censReg0impute   967.624     2.508       966.082
## censReg1naive    676.668    16.228       661.101
## best              48.739     0.012        48.775
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        3425.774     0.053      3429.150
## censReg1year   68140.778 67857.204       283.858
## censReg2        3447.518     0.014      3450.955
## censReg0impute  3484.451     0.009      3487.930
## censReg1naive   5494.548  1766.024      3732.257
## best              45.419     0.044        45.420
```

```
##                 mse_beta bias_beta variance_beta
## censReg1        3335.115    22.650      3315.781
## censReg1year   67387.056 67108.023       279.312
## censReg2        3381.296    24.587      3360.070
## censReg0impute  3390.256    23.683      3369.943
## censReg1naive   5809.054  2212.042      3600.613
## best              45.419     0.044        45.420
```

```
##                mse_beta bias_beta variance_beta
## censReg1         54.857     0.026        54.886
## censReg1year   2036.314  1969.299        67.082
## censReg2         54.897     0.027        54.925
## censReg0impute   59.505     0.026        59.538
## censReg1naive    60.239     4.736        55.559
## best             52.143     0.014        52.182
```

```
##                    mse_beta   bias_beta variance_beta
## censReg1           872.515       1.142       872.245
## censReg1year   1569456.160 1569299.231       157.087
## censReg2           902.287       0.671       902.519
## censReg0impute    1364.773       4.365      1361.770
## censReg1naive      875.408       4.792       871.487
## best                52.143       0.014        52.182
```

```
##                    mse_beta   bias_beta variance_beta
## censReg1         14430.918     134.536     14310.693
## censReg1year   1527622.012 1526727.994       894.913
## censReg2         14532.623     149.934     14397.087
## censReg0impute   14630.316     136.687     14508.137
## censReg1naive    14966.538     805.787     14174.926
## best                49.687       0.000        49.736
```

```
##                    mse_beta   bias_beta variance_beta
## censReg1         14733.777      44.951     14703.529
## censReg1year   1531519.710 1530629.282       891.320
## censReg2         15019.149      44.288     14989.851
## censReg0impute   15182.028      44.028     15153.153
## censReg1naive    15210.938     554.997     14670.611
## best                49.687       0.000        49.736
```

Results

Estimation of the regression coefficient `beta28year`

Our first goal will be to screen our 11 methods for the estimation of `beta28year` to determine which methods we will use in our main analysis in a later section. We will assess these estimates from their MSE, squared-bias and variance in each case.

We first choose parameter values for this screening study: cprop = 0.3, beta153year = -0.02, sd28_153= 0.3. These values for cprop and beta153year are equal to our estimates from our real dataset `pcb.csv` , whereas this value for sd28_153 is equal to the mean of two estimates: one which is unconditional, and a second which is conditional on the variable `year` .

Evaluation of methods for smaller sample sizes

We will first obtain results from datasets with different sample sizes in order to decide an appropriate sample size for all our subsequent work. Our real dataset has approximately 100 observations per year for CB28 and CB153 from herring in years 2003-2017. However these observations are from various locations and have differences for various other variables such as age, fat-percentage etc., which means that any statistical analysis which controls for such variables would have a smaller sample size. We will test sample sizes that differ by a factor of 2: we do this by generating datasets by simulation using 10000 iterations, with sample sizes 50, 25, 12 and 6 respectively. The squared-bias of the estimates of `beta28year` from all 11 methods and all 4 sample sizes is shown below; note that all values shown in the table are 100000 times bigger than the actual values (to make them easier to read and compare). The column names `bias_ss50` , `bias_ss25` , … denote sample sizes 50, 25, … respectively.

```
##                 mse_beta bias_beta variance_beta
## omit             7.32619   6.30801       1.01828
## subst2           1.92885   0.18133       1.74770
## subst1           3.00110   2.21388       0.78729
## censReg1         1.37640   0.00011       1.37642
## censReg2         1.49855   0.00008       1.49862
## censReg0impute   1.49934   0.00008       1.49941
## best             1.36221   0.00120       1.36114
## subst4           8.86321   5.47356       3.39000
## censReg1naive    1.54927   0.66680       0.88256
## subst2lmimpute   6.53746   5.71740       0.82014
## omitlmimpute     7.78202   6.68353       1.09859
```

```
##                 mse_beta bias_beta variance_beta
## omit             8.44831   6.25923       2.18930
## subst2           3.91093   0.19689       3.71441
## subst1           3.85546   2.17862       1.67700
## censReg1         2.92659   0.00028       2.92661
## censReg2         3.17724   0.00023       3.17732
## censReg0impute   3.17923   0.00026       3.17929
## best             2.86629   0.00027       2.86631
## subst4          12.77266   5.58593       7.18745
## censReg1naive    2.55419   0.67152       1.88286
## subst2lmimpute   7.38429   6.02401       1.36041
## omitlmimpute     8.42709   6.69863       1.72863
```

```
##                 mse_beta bias_beta variance_beta
## omit            10.99143   6.28979       4.70211
## subst2           7.61409   0.15935       7.45548
## subst1           5.63409   2.27112       3.36331
## censReg1         5.85494   0.00088       5.85465
## censReg2         6.36029   0.00125       6.35968
## censReg0impute   6.36792   0.00127       6.36728
## best             5.65520   0.00163       5.65414
## subst4          19.73593   5.31486      14.42251
## censReg1naive    4.45252   0.70285       3.75005
## subst2lmimpute   9.17870   6.73582       2.44312
## omitlmimpute     9.64148   6.79578       2.84598
```

```
##                 bias_ss50 bias_ss25 bias_ss12 bias_ss6
## omit             626.3345  630.8013  625.9227 628.9789
## subst2            16.3590   18.1328   19.6888  15.9350
## subst1           224.6076  221.3881  217.8623 227.1120
## censReg1           0.0149    0.0113    0.0275   0.0875
## censReg2           0.0157    0.0077    0.0231   0.1246
## censReg0impute     0.0204    0.0081    0.0257   0.1274
## best               0.1186    0.1202    0.0269   0.1626
## subst4           532.5090  547.3556  558.5935 531.4857
## censReg1naive     69.4810   66.6803   67.1517  70.2850
## subst2lmimpute   559.5247  571.7404  602.4013 673.5820
## omitlmimpute     650.4391  668.3533  669.8625 679.5784
```

The following table below is the same as the previous one, except that it shows the variance of the estimates.

```
##                mse_beta bias_beta variance_beta
## omit            7.32619   6.30801       1.01828
## subst2          1.92885   0.18133       1.74770
## subst1          3.00110   2.21388       0.78729
## censReg1        1.37640   0.00011       1.37642
## censReg2        1.49855   0.00008       1.49862
## censReg0impute  1.49934   0.00008       1.49941
## best            1.36221   0.00120       1.36114
## subst4          8.86321   5.47356       3.39000
## censReg1naive   1.54927   0.66680       0.88256
## subst2lmimpute  6.53746   5.71740       0.82014
## omitlmimpute    7.78202   6.68353       1.09859
```

```
##                mse_beta bias_beta variance_beta
## omit            8.44831   6.25923       2.18930
## subst2          3.91093   0.19689       3.71441
## subst1          3.85546   2.17862       1.67700
## censReg1        2.92659   0.00028       2.92661
## censReg2        3.17724   0.00023       3.17732
## censReg0impute  3.17923   0.00026       3.17929
## best            2.86629   0.00027       2.86631
## subst4         12.77266   5.58593       7.18745
## censReg1naive   2.55419   0.67152       1.88286
## subst2lmimpute  7.38429   6.02401       1.36041
## omitlmimpute    8.42709   6.69863       1.72863
```

```
##                mse_beta bias_beta variance_beta
## omit           10.99143   6.28979       4.70211
## subst2          7.61409   0.15935       7.45548
## subst1          5.63409   2.27112       3.36331
## censReg1        5.85494   0.00088       5.85465
## censReg2        6.36029   0.00125       6.35968
## censReg0impute  6.36792   0.00127       6.36728
## best            5.65520   0.00163       5.65414
## subst4         19.73593   5.31486      14.42251
## censReg1naive   4.45252   0.70285       3.75005
## subst2lmimpute  9.17870   6.73582       2.44312
## omitlmimpute    9.64148   6.79578       2.84598
```

```
##                variance_ss50 variance_ss25 variance_ss12 variance_ss6
## omit                 47.2420      101.8282      218.9303     470.2108
## subst2               85.8241      174.7699      371.4413     745.5481
## subst1               38.5270       78.7294      167.7003     336.3310
## censReg1             67.6445      137.6421      292.6611     585.4652
## censReg2             73.2962      149.8619      317.7323     635.9679
## censReg0impute       73.3600      149.9412      317.9287     636.7285
## best                 71.7739      136.1141      286.6308     565.4141
## subst4              166.3544      338.9997      718.7448    1442.2512
## censReg1naive        44.0397       88.2556      188.2861     375.0047
## subst2lmimpute       60.0580       82.0139      136.0411     244.3123
## omitlmimpute         89.1732      109.8593      172.8633     284.5983
```

Allowing for random error from using only 10000 iterations, we can conclude that the squared-bias is independent of sample size, whereas the variance is inversely proportional sample size. Moreover since the bias_variance decomposition $ MSE = Bias^2 + Variance$, always holds, we need not look at the MSE values for the purpose of choosing sample size.

We find in additional experiments (details not shown) that the standard error of the estimates is inversely proportional to the square root of the number of simulation iterations, so we have three factors to balance:

1. We want our results to be potentially applicable for real data.
2. We want sample size to be sufficiently large to avoid MSE being dominated by variance alone.
3. We want the number of iterations to be sufficiently large that our estimates have sufficiently low standard error.

We therefore decide to use sample size = 12 for all of our subsequent experiments.

```
##                 mse_beta bias_beta variance_beta
## omit             6.73529   6.26334       0.47242
## subst2           1.02097   0.16359       0.85824
## subst1           2.63096   2.24608       0.38527
## censReg1         0.67592   0.00015       0.67645
## censReg2         0.73239   0.00016       0.73296
## censReg0impute   0.73307   0.00020       0.73360
## best             0.71821   0.00119       0.71774
## subst4           6.98697   5.32509       1.66354
## censReg1naive    1.13477   0.69481       0.44040
## subst2lmimpute   6.19523   5.59525       0.60058
## omitlmimpute     7.39523   6.50439       0.89173
```

```
##                 mse_beta bias_beta variance_beta
## omit             7.30307   6.31339       0.99067
## subst2           1.82651   0.13937       1.68883
## subst1           3.05353   2.30050       0.75378
## censReg1         1.32626   0.00127       1.32632
## censReg2         1.43939   0.00156       1.43926
## censReg0impute   1.43625   0.00133       1.43636
## best             1.37819   0.00010       1.37947
## subst4           8.41458   5.12293       3.29495
## censReg1naive    1.57556   0.71425       0.86217
## subst2lmimpute   6.65171   5.83757       0.81496
## omitlmimpute     7.78183   6.66494       1.11800
```

```
##                 mse_beta bias_beta variance_beta
## omit             8.41972   6.35020       2.07160
## subst2           3.82340   0.12135       3.70576
## subst1           3.97220   2.34445       1.62938
## censReg1         2.87939   0.00449       2.87778
## censReg2         3.14083   0.00496       3.13900
## censReg0impute   3.15323   0.00456       3.15182
## best             2.90273   0.00305       2.90258
## subst4          12.20168   4.96336       7.24557
## censReg1naive    2.55612   0.79263       1.76526
## subst2lmimpute   7.52554   6.19608       1.33079
## omitlmimpute     8.84331   7.15221       1.69280
```

```
##                mse_beta bias_beta variance_beta
## omit           10.93907   5.58942       5.35500
## subst2          8.11560   0.41070       7.71261
## subst1          5.49398   1.79078       3.70691
## censReg1        6.28797   0.04032       6.25390
## censReg2        6.81454   0.04134       6.77997
## censReg0impute  6.84659   0.03893       6.81448
## best            5.77008   0.00002       5.77584
## subst4         21.29759   6.86397      14.44806
## censReg1naive   4.60166   0.41858       4.18726
## subst2lmimpute  8.86108   6.20702       2.65672
## omitlmimpute    9.17897   6.22616       2.95576
```

Selection of censoring methods for further study

We will now use simulations with just 1000 iterations for all 10 methods (and also for our reference method `best`) to estimate `beta28year` for four sets of parameter values:

beta28year = -0.02 is held fixed.

a "low" and a "high" value for each of `cprop` and `sd28_153` are used. Concretely: (0.1, 0.1), (0.7, 0.1), (0.1, 0.5) and (0.7, 0.5) were used for (cprop, sd28_153) respectively.

The following four tables show the MSE, squared-bias, and variance of estimates of `beta28year` from all 11 methods, for the four sets of parameter values, respectively.

We see that there is a much bigger difference between different methods in the amount of bias than in the amount of variance. We will therefore focus primarily on the results for bias; we will use terms such as high and low to compare the bias from different methods. We see that the amount of bias for:

`best` serves as a reference value; a gold standard that we compare the other methods with.

`omit` is high for (0.1, 0.1) and (0.1, 0.5), and is very high for (0.7, 0.1) and (0.7, 0.5). It makes sense that there is higher bias with higher proportion of censored values since a higher proportion of the data has been omitted. Moreover, these generally high values are commensurate with our prior expectations (ref: Helsel's book) that `omit` is a poor method, so we will not study this method further.

Very high for: `subst1` for (0.7, 0.1) and (0.7, 0.5); `subst2` for (0.7, 0.5); `subst4` for (0.1, 0.1) and (0.7, 0.1). However, all three substitution methods also have low bias for at least one set of parameter values. This is intriguing and merits further investigation.

Very low for: `censReg1` , `censReg2` and `censReg0impute` for all four parameter value sets.

Highest of all four censReg methods for all four parameter sets for `censReg1naive` . This illustrates the necessity of conditioning on both the cb153 value and the condition cb28 < cb28_cprop by using a truncated normal distribution, and verifies the results we presented in our previous chapter on mathematical theory. `censReg1` , `censReg2` and `censReg0impute` all do this, whereas in contrast, `censReg1naive` conditions solely on the cb153 value, and thus uses a (non-truncated) normal distribution; this results in significant bias because cb28 values can be erroneously imputed to be higher than `cb28_cprop` . Consequently we will not discuss censReg1naive any further: it has served its purpose in showing the importance of conditioning oth the cb153 value and the condition cb28 < cb28_cprop.

The hybrid methods `subst2lmimpute` and `omitlmimpute` first use substitution and omission as in `subst2` and `omit` respectively, followed by imputation. Therefore `subst2lmimpute` should be compared with `subst2` , and `omitlmimpute` with `omit` . `subst2lmimpute` has higher bias (and also MSE) than `subst2` for all parameter value sets, and `omitlmimpute` has higher bias than `omit` for all sets except (0.1, 0.5). We have already rejected the `omit` method so we must also reject `omitlmimpute` since it performs no better than `omit` . Similarly we reject `subst2lmimpute` , since this method performed worse than `subst2` in all four cases.

In summary, we have rejected 4 of our 10 methods. We will limit our attention to six methods for all our subsequent work: the three substitution methods `subst1` , `subst2` , `subst4` , and the three censReg methods `censReg1` , `censReg2` and `censReg0impute` . We will use `best` as our reference method throughout.

Evaluation of methods for larger absolute values of `beta28year`

We will now focus our six chosen methods `subst1` , `subst2` , `subst4` , `censReg1` , `censReg2` , `censReg0impute` . We will use these methods to estimate `beta28year` from four simulations that use the same parameter values ss = 12, cprop = 0.3, sd28_153 = 0.3, n_iter = 10000 as before. We will use the four `cb153year` parameter values -0.02, -0.04, -0.08, -0.16 in these four simulations, respectively. The results from the simulations are shown in the four tables below.

We see that for these parameters value sets, censReg method give estimates that have much lower bias, in general. The `subst1` method is designed as a reference that gives biased estimates, since it substitutes cb28 values that are observed to be below LOD with the LOD value itself, so the substituted values will always be larger than the real values. Moreover, we have chosen to maintain a constant LOD level for all years of the same dataset. We are also simulating cb28 and cb153 data using a linear (degree 1 polynomial) function with a negative slope and a fixed constant (intercept) term. This means that the cb28 values decrease faster with years for larger values of abs(beta153year). This all means that it is an inevitable consequence of our design that the bias from `subst1` increases as abs(beta153year) increases, which is precisely what we see in these results.

In contrast, the bias from `subst4` first increases from abs(beta153year) = 0.02 to 0.08 and then decreases for abs(beta153year) = 0.16. This suggests that since `subst4` substitutes censored values with $\frac{LOD}{2}$, which are lower than the true values on average for low values of abs(beta153year) but not lower for the highest value abs(beta153year) = 0.16. This is also supported by the fact that the bias from `subst2` is much lower than that from `subst1` or `subst4` , which suggests that the real values of the censored data mostly lie between LOD and $\frac{LOD}{2}$.

The three censReg methods all give very similar results to one another; the values of MSE, squared-bias and variance are very similar from these methods for both abs(beta153year) = 0.08 and 0.16. However, for the lowest value abs(beta153year) = 0.02, the variance from `censReg1` is approximately 10% lower than from `censReg2` , which is a statistically significant difference. This fits with our prior knowledge that a more complex model generally has higher variance than the corresponding less complex one. Moreover, we also expected that the estimates from `censReg2` would improve relative to those from `censReg1` as the value of abs(beta153year) increases, since the difference between these two methods is that `censReg2` uses `year` as additional predictor variable. However, the fact that `censReg0impute` has a higher variance than `censReg1` seems puzzling in this respect, so perhaps our interpretation of model complexity is wrong is this context.

Our prior expectation was that `censReg0impute` would perform relatively worse compared to the other `censReg` methods for larger values of abs(beta153year). This is because we conjecture that the imputations from the predictor variables carry more information about cb28 as the absolute value of `beta28year` increases, and `censReg0impute` does not use imputation at all. However, these results fail to support our conjecture here too.

If we now compare the best performing models from each category, i.e. `subst2` and `censReg1` , we see that `censReg1` has much lower bias for all parameter values. However, MSE for `subst2` is lower for one of the values, abs(beta153year) = 0.08. In conclusion, we can say that `censReg1` gives better estimates than `subst2` for most, but not necessarily all, values of abs(beta153year).

```
##                mse_beta bias_beta variance_beta
## subst1           3.9272    2.2607        1.6667
## subst2           3.8331    0.1603        3.6732
## subst4          12.4075    5.3096        7.0986
## censReg1         2.8892    0.0003        2.8891
## censReg2         3.1412    0.0006        3.1409
## censReg0impute   3.1444    0.0006        3.1441
## best             2.8939    0.0001        2.8941
```

```
##                mse_beta bias_beta variance_beta
## subst1          10.3914    8.6084        1.7833
## subst2           4.0038    0.4912        3.5130
## subst4          25.2352   18.7980        6.4379
## censReg1         3.1584    0.0001        3.1586
## censReg2         3.2579    0.0002        3.2580
## censReg0impute   3.2621    0.0002        3.2622
## best             2.8074    0.0001        2.8076
```

```
##                mse_beta bias_beta variance_beta
## subst1          32.8102   30.5608        2.2497
## subst2           3.5467    0.3798        3.1672
## subst4          50.4127   45.7078        4.7055
## censReg1         3.6276    0.0002        3.6278
## censReg2         3.6445    0.0002        3.6446
## censReg0impute   3.6540    0.0003        3.6540
## best             2.8808    0.0006        2.8805
```

```
##                mse_beta bias_beta variance_beta
## subst1          99.3248   96.2660        3.0591
## subst2           6.0901    2.8591        3.2313
## subst4          44.8933   41.3417        3.5519
## censReg1         4.4961    0.0020        4.4945
## censReg2         4.4988    0.0020        4.4972
## censReg0impute   4.5284    0.0024        4.5265
## best             2.8542    0.0017        2.8528
```

Evaluation of methods for other values of `sd28_153`

We will now hold `beta28year` and `cprop` fixed at their original values (-0.02 and 0.3) and investigate the effect of larger `sd28_153` values, specifically: 0.1, 0.3, 0.5, and 0.7.

We see again that for these parameters value sets, censReg method give estimates that have very much lower bias, in general. Since the three censReg methods all give very similar results to one another and very different results from the three substitution methods, we will again begin by interpreting the results for these two method categories separately.

The bias from `subst4` decreases greatly as the value of `sd28_153` increases, whereas the bias from `subst1` is relatively independent of the value of `sd28_153` . The bias from `subst2` again follows a trend intermediate between that of `subst1` and `subst4` , since it decreases from sd28_153 = 0.1 to 0.5 and then decreases for sd28_153 = 0.7. Our interpretation is that since the censored values lie closer on average to LOD for smaller values of sd28_153, and further away for larger values. The low bias from `subst4` for sd28_153 = 0.7 indicates that the real values for the censored data lie close to $\frac{LOD}{2}$ on average for this parameter value.

The large gap between the uncensored cb28 data and the $\frac{LOD}{2}$ value means that `subst4` gives higher variance than all other methods for all values of sd28_153. Similarly, the smallest possible gap between LOD and the uncensored cb28 data explains the fact that `subst1` always gives the lowest variance. We conjecture that the same logic would also hold for other possible substitution values; the larger the gap between this value and LOD, the larger the resulting variance.

Again, we see that the variance from `censReg1` is approximately 10 % lower than that from `censReg2` for all four values of sd28_153. Surprisingly the results from `censReg2` and `censReg0impute` are almost identical. Is this a bug?

In conclusion, substitution methods give much higher bias than cenreg methods. Moreover, all three cenreg methods gave lower MSE than all three substitution methods for both sd28_153 = 0.1 and sd28_153 = 0.3. However, the variance from cenreg methods increases faster than from substitution methods as sd28_153 increases; in fact for higher values of sd28_153, `subst1` and `subst2` gave the lowest and second lowest MSE values, respectively. This relative failure of cenreg methods for relatively high values of sd28_153 makes sense, here is our explanation: A higher `sd28_153` value means that the correlation between cb28 and cb153 is weaker, which results in less accurate imputation by ´censReg1 `and` censReg2`, since the accuracy of imputation by these methods relies on the strength of correlation between cb28 and cb153.

```
##                 mse_beta bias_beta variance_beta
## subst1            2.4314    2.1164        0.3150
## subst2            9.3323    8.0908        1.2416
## subst4           54.2390   51.0318        3.2075
## censReg1          0.5166    0.0000        0.5167
## censReg2          0.5478    0.0001        0.5478
## censReg0impute    0.5575    0.0001        0.5574
## best              0.4887    0.0000        0.4887
```

```
##                 mse_beta bias_beta variance_beta
## subst1            3.9272    2.2607        1.6667
## subst2            3.8331    0.1603        3.6732
## subst4           12.4075    5.3096        7.0986
## censReg1          2.8892    0.0003        2.8891
## censReg2          3.1412    0.0006        3.1409
## censReg0impute    3.1444    0.0006        3.1441
## best              2.8939    0.0001        2.8941
```

```
##                 mse_beta bias_beta variance_beta
## subst1            6.5713    2.2449        4.3269
## subst2            7.3837    0.1013        7.2831
## subst4           12.4701    0.7425       11.7287
## censReg1          7.6126    0.0001        7.6133
## censReg2          8.3107    0.0001        8.3114
## censReg0impute    8.3088    0.0001        8.3095
## best              7.4450    0.0002        7.4455
```

```
##                 mse_beta bias_beta variance_beta
## subst1           10.7825    2.2270        8.5564
## subst2           12.9888    0.4014       12.5887
## subst4           18.2077    0.0507       18.1587
## censReg1         15.2039    0.0002       15.2052
## censReg2         16.5719    0.0001       16.5735
## censReg0impute   16.5751    0.0001       16.5767
## best             14.8431    0.0035       14.8411
```

Further comparisons between `subst2` and `censReg1`

From our previous results, `subst2` is generally the best performing substitution method and `censReg1` is the best censReg method. In the previous section, these methods gave similar MSE values for sd28_153 = 0.5, so we will fix this parameter at this value and investigate these estimation methods for four values of cprop: 0.1, 0.3, 0.5, 0.7. These cprop values correspond to censoring 10 %, 30 %, 50 %, and 70% of the data respectively, so they correspond to decreasing values of LOD, which is our variable of primary interest.

We see that `censReg1` gives estimates with very low bias for all values of `cprop`, whereas the bias from `subst2` increases greatly as `cprop` increases. We interpret this as meaning that the real cb28 values are unchanged when LOD is lowered, which means that a higher proportion are likely to lie closer to LOD for larger values of `cprop` which means that substituted values are increasingly biased towards being too small as `cprop` increases. Since `censReg1` fits a model to all the data (censored and uncensored) it maintains low bias as the LOD decreases, whilst the variance remains approximately constant. However, as a greater proportion of values are substituted for the same constant value by the `subst2` method, the variance decreases because a higher proportion of the data values are identical.

In conclusion, `censReg1` gives similar bias and variance for different values of cprop whereas `subst2` does not. From `subst2` the bias increases and the variance decreases as `cprop` increases.

THIS SECTION IS NOW COMPLETE :)

```
##          mse_beta bias_beta variance_beta
## subst2     8.0879    0.0041        8.0846
## censReg1   7.6265    0.0011        7.6261
## best       7.7061    0.0002        7.7067
```

```
##          mse_beta bias_beta variance_beta
## subst2     6.7352    1.3153        5.4205
## censReg1   7.4995    0.0001        7.5002
```

```
##          mse_beta bias_beta variance_beta
## subst2     6.8178    1.3197        5.4987
## censReg1   7.6068    0.0002        7.6074
```

```
##          mse_beta bias_beta variance_beta
## subst2     8.6654    5.3095        3.3562
## censReg1   8.0983    0.0001        8.0991
```

The MSE, squared-bias and variance of predictions of `cb28` annual means from various censoring methods

All the graphs in this section will shows MSE, squared-bias, or variance on the y-axis and year on the x-axis for the simulated 15-year period. We begin by looking at variance of predictions from our best three substitution methods, best three censReg methods. We will again use `best` as our gold standard.

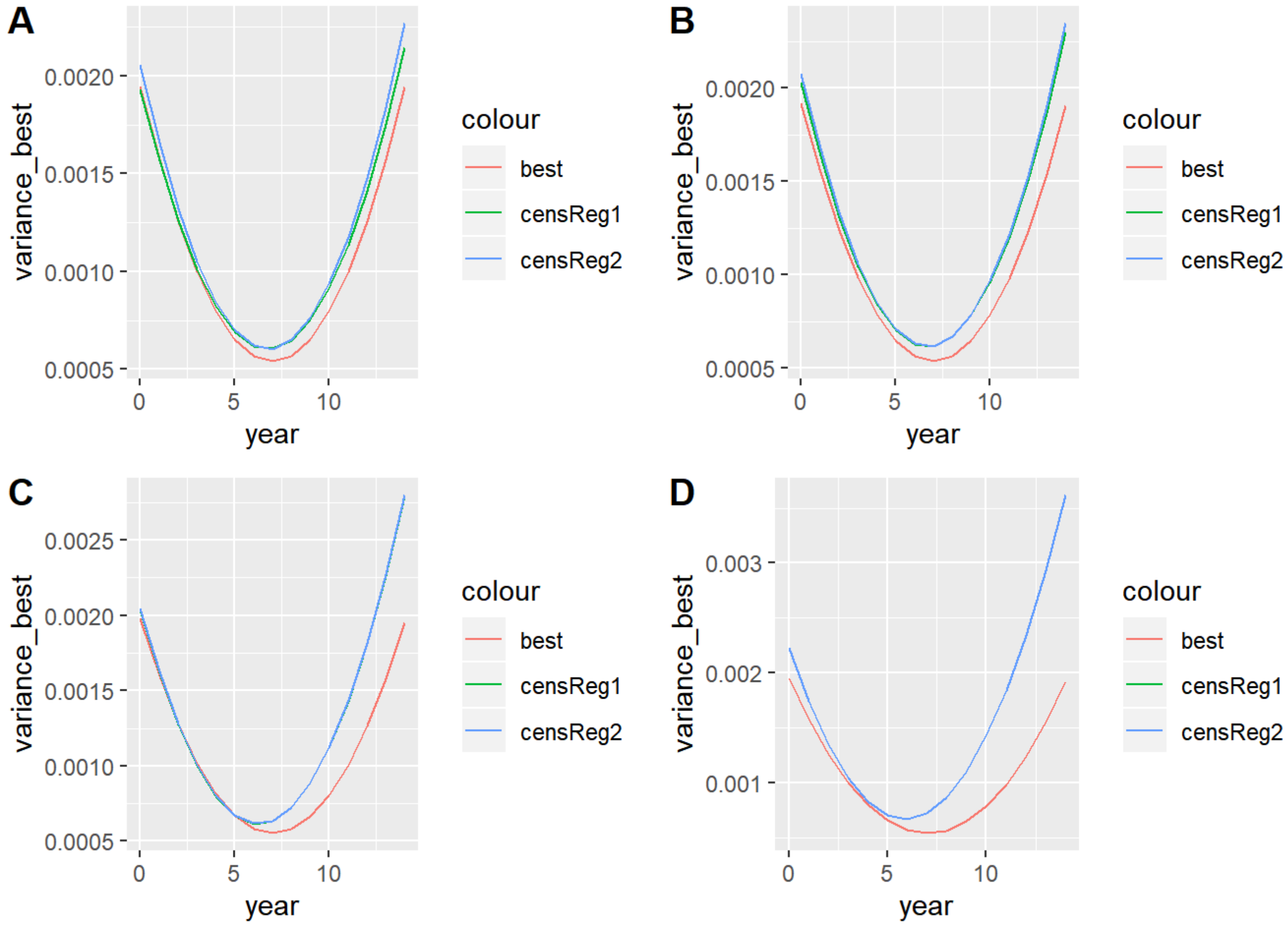
Variance of predictions of cb28 annual means from different methods

Predictions for different values of beta153year

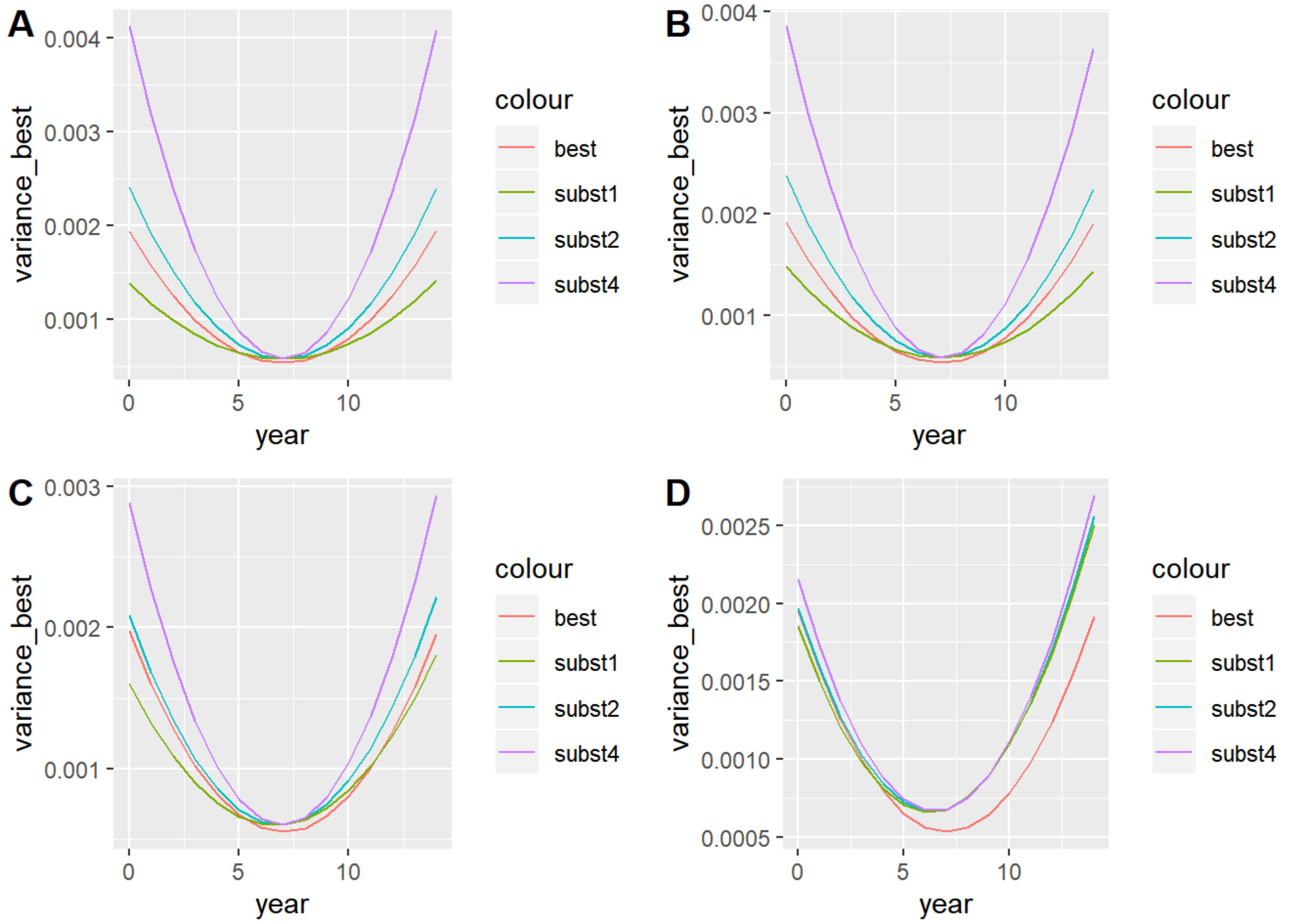
We will begin by using the same parameter values we used in our earlier section “Evaluation of methods for larger absolute values of `beta28year`”. These parameters are fixed: cprop = 0.3, sd28_153 = 0.3, whilst cb153year is given four values: -0.02, -0.04, -0.08 and -0.16 respectively.

We begin by showing graphs of the variance of predictions of cb28 annual means from our chosen censoring methods. A common feature of all these graphs is that they typically have an approximately parabolic “U” shape, with higher variance at each end of the time period than in the middle of the period. This is in accordance with our prior expectations because this is generally the case.

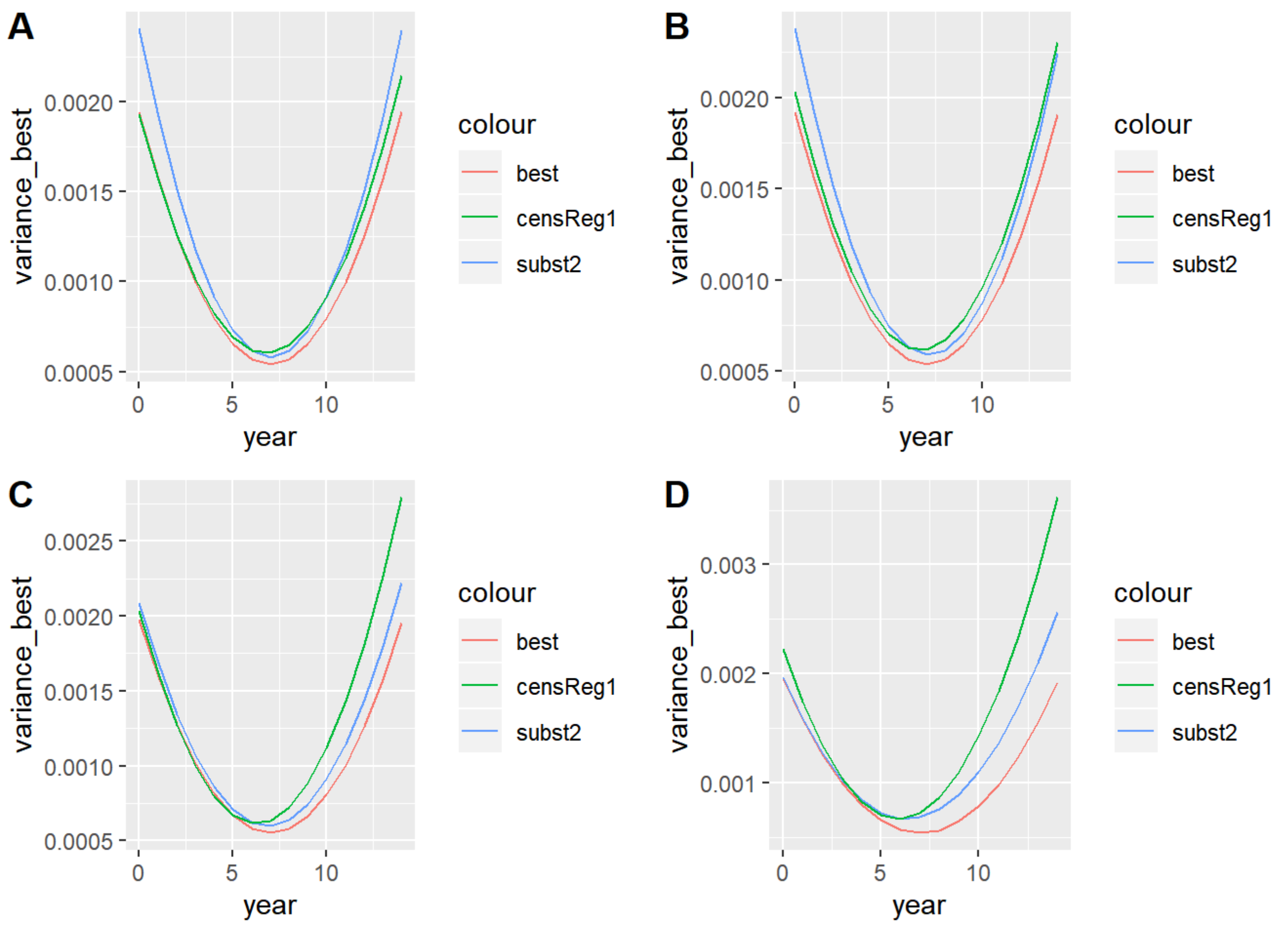
Our first set of four graphs show the variance of `censReg1` and `censReg2` methods relative to `best` method for beta153year equal to -0.02, -0.04, -0.08, -0.16, respectively.



Our second set of four graphs show the variance of `subst1` , `subst2` and `subst4` methods relative to `best` method for beta153year equal to -0.02, -0.04, -0.08, -0.16, respectively.

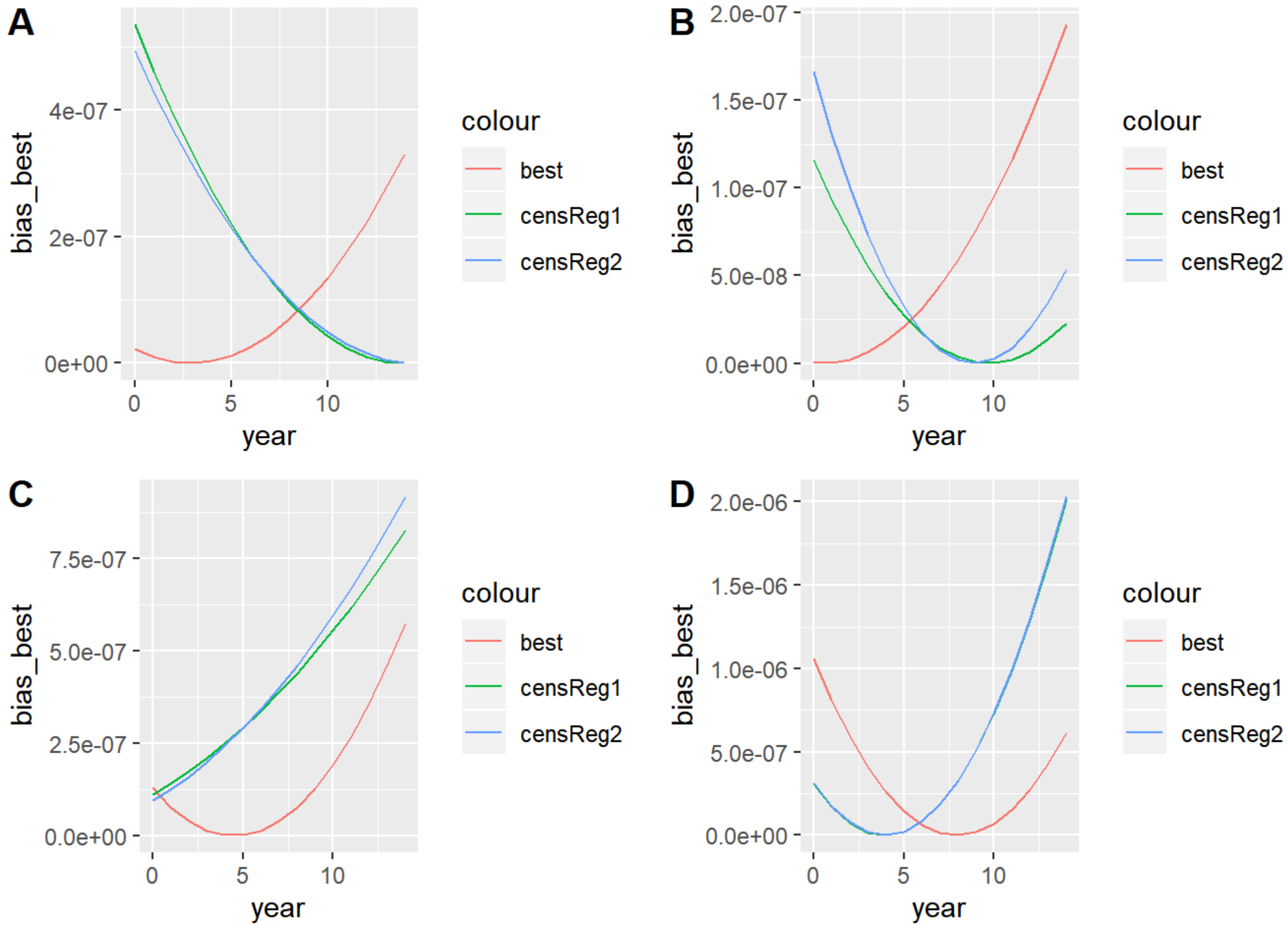


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

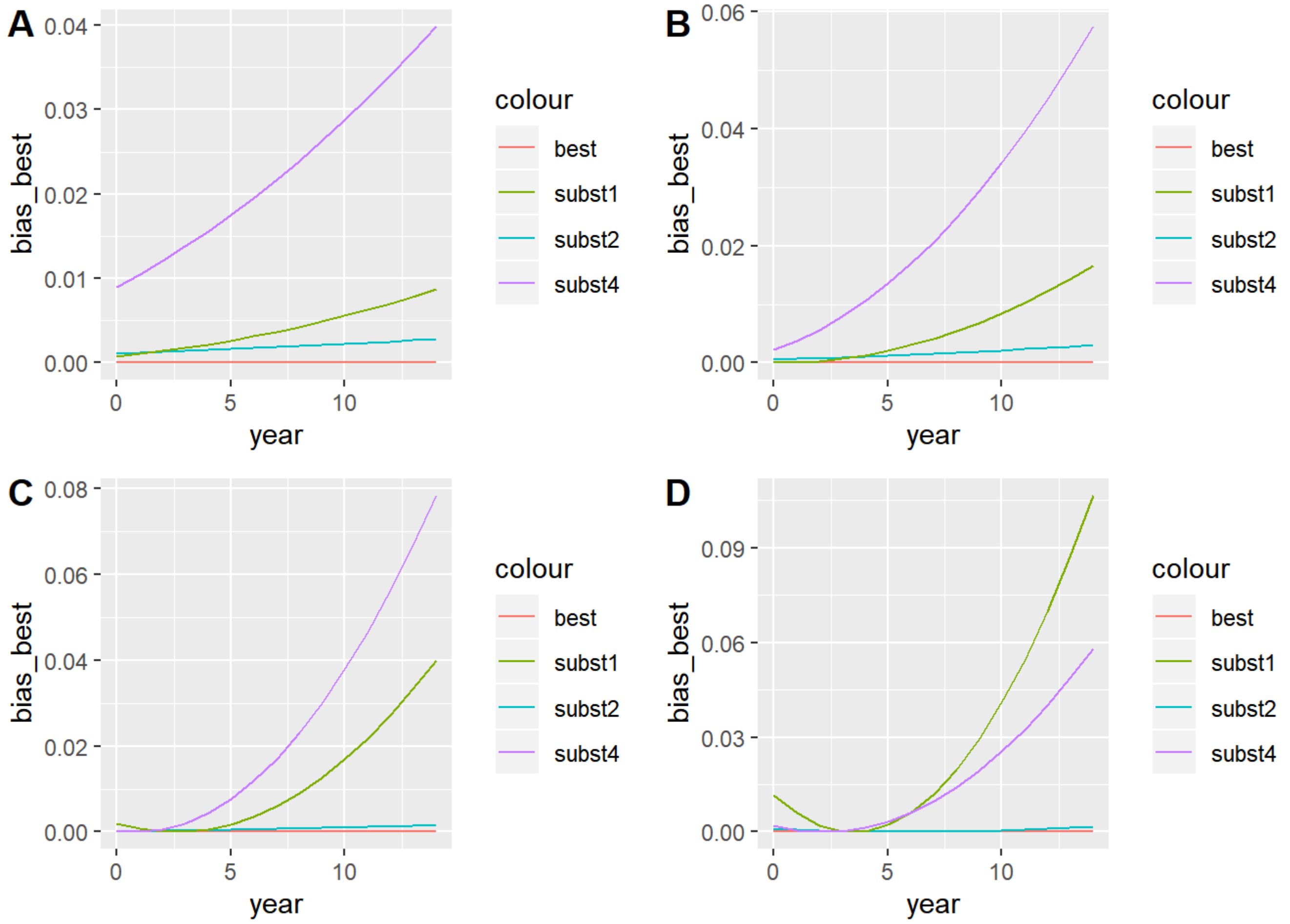


We will now show graphs of the bias of predictions of cb28 annual means from our chosen censoring methods.

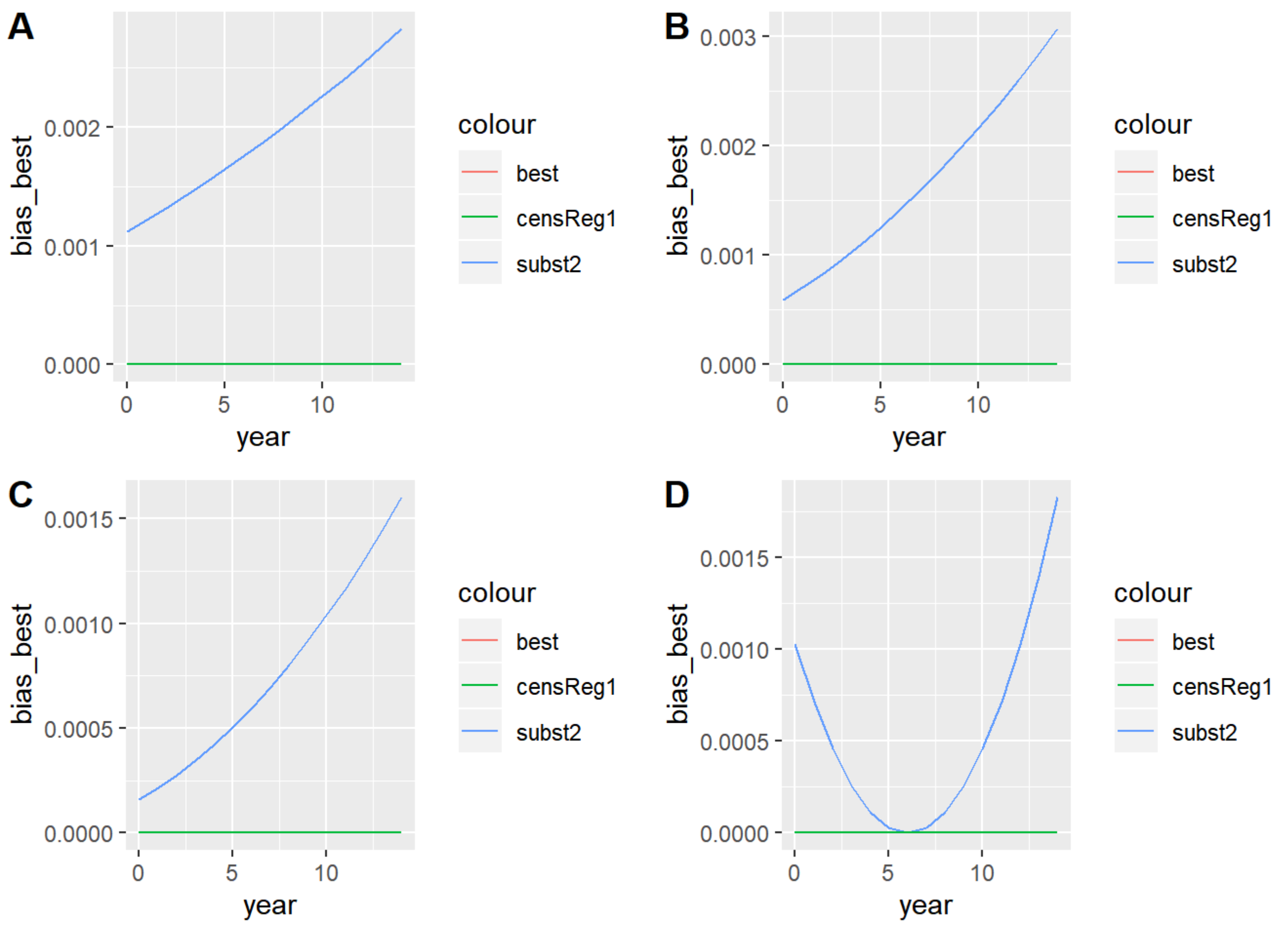
Our first set of four graphs show the bias of `censReg1` and `censReg2` methods relative to `best` method for beta153year equal to -0.02, -0.04, -0.08, -0.16, respectively.



Our second set of four graphs show the bias of `subst1` , `subst2` and `subst4` methods relative to `best` method for beta153year equal to -0.02, -0.04, -0.08, -0.16, respectively.

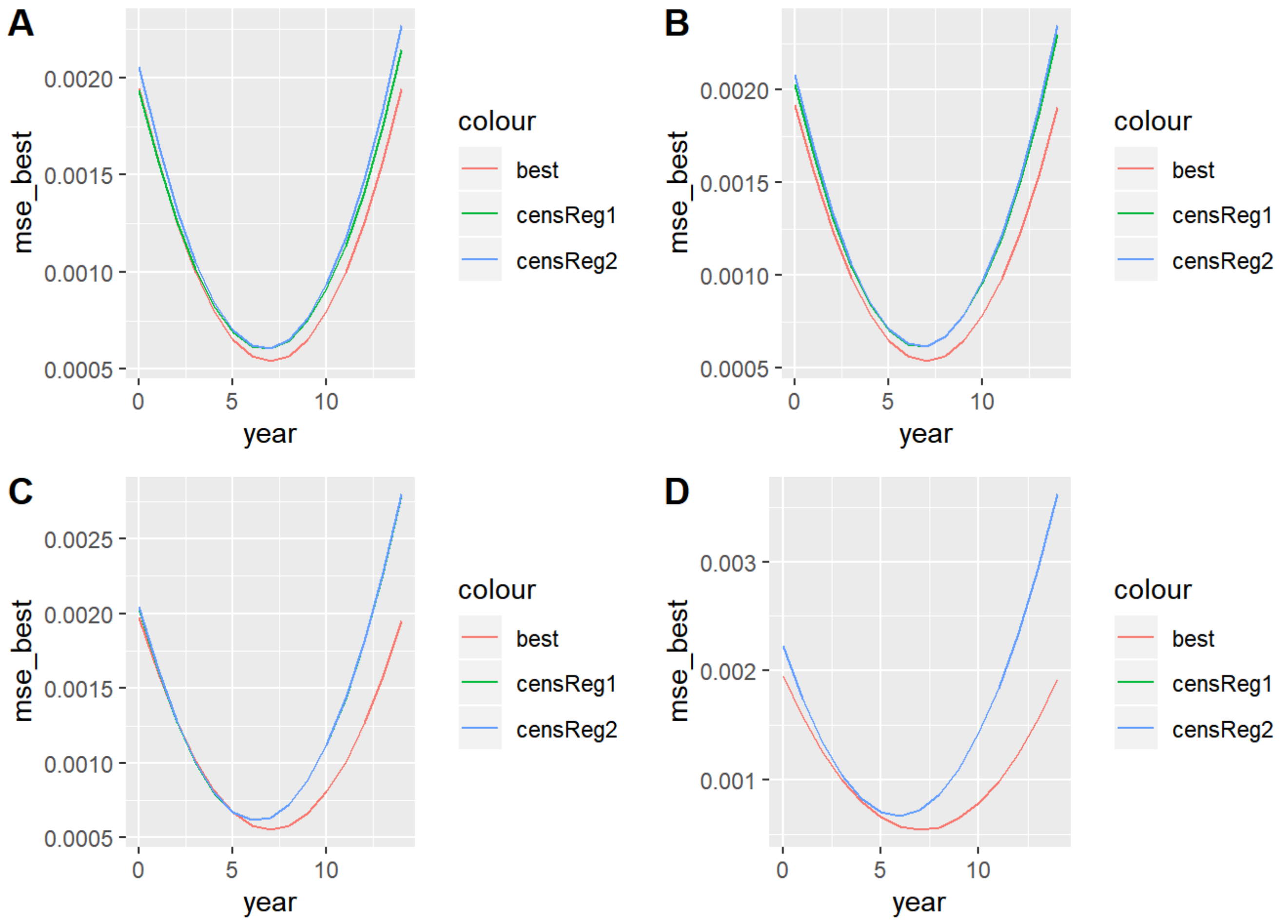


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

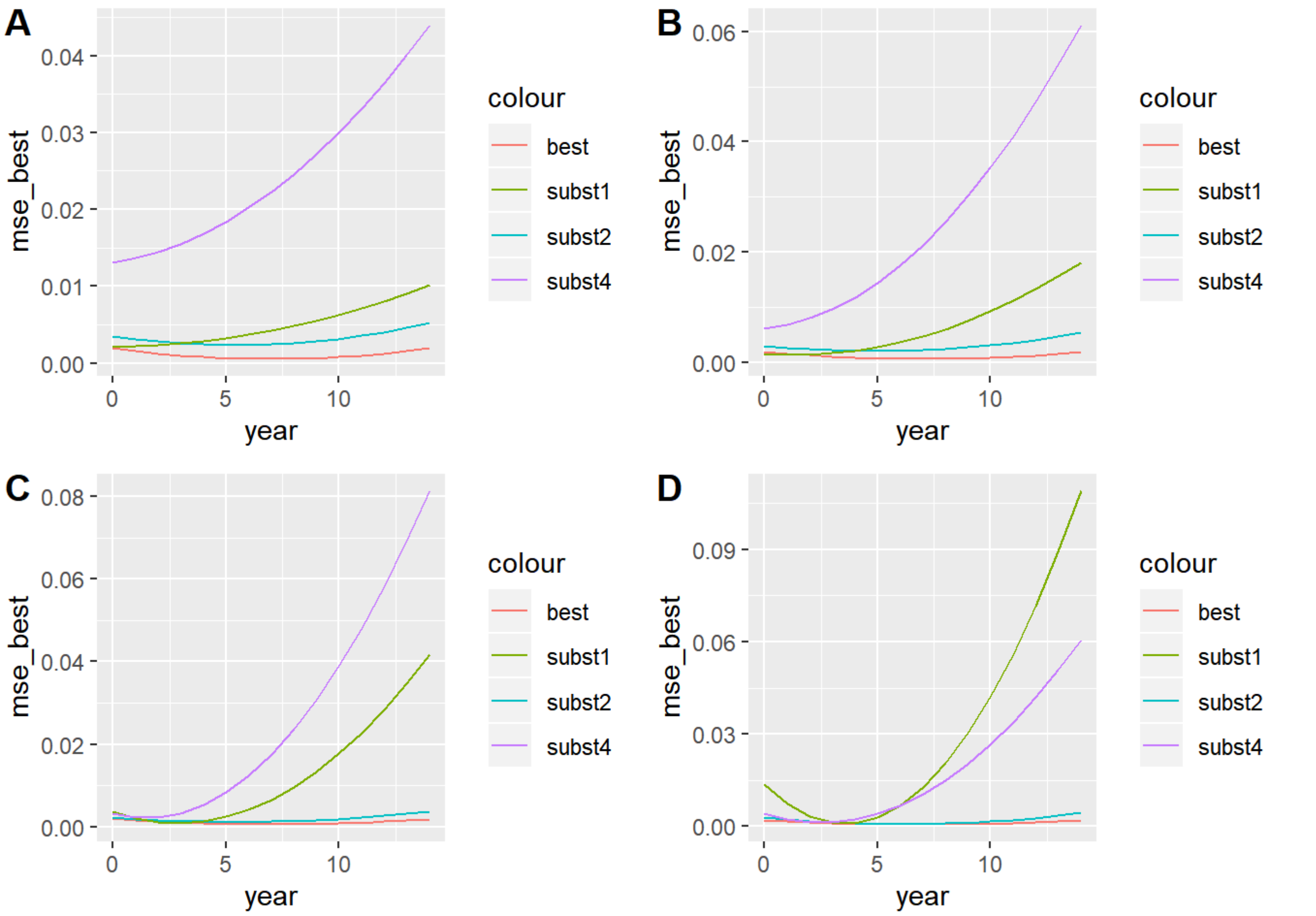


We will now show graphs of the MSE of predictions of cb28 annual means from our chosen censoring methods.

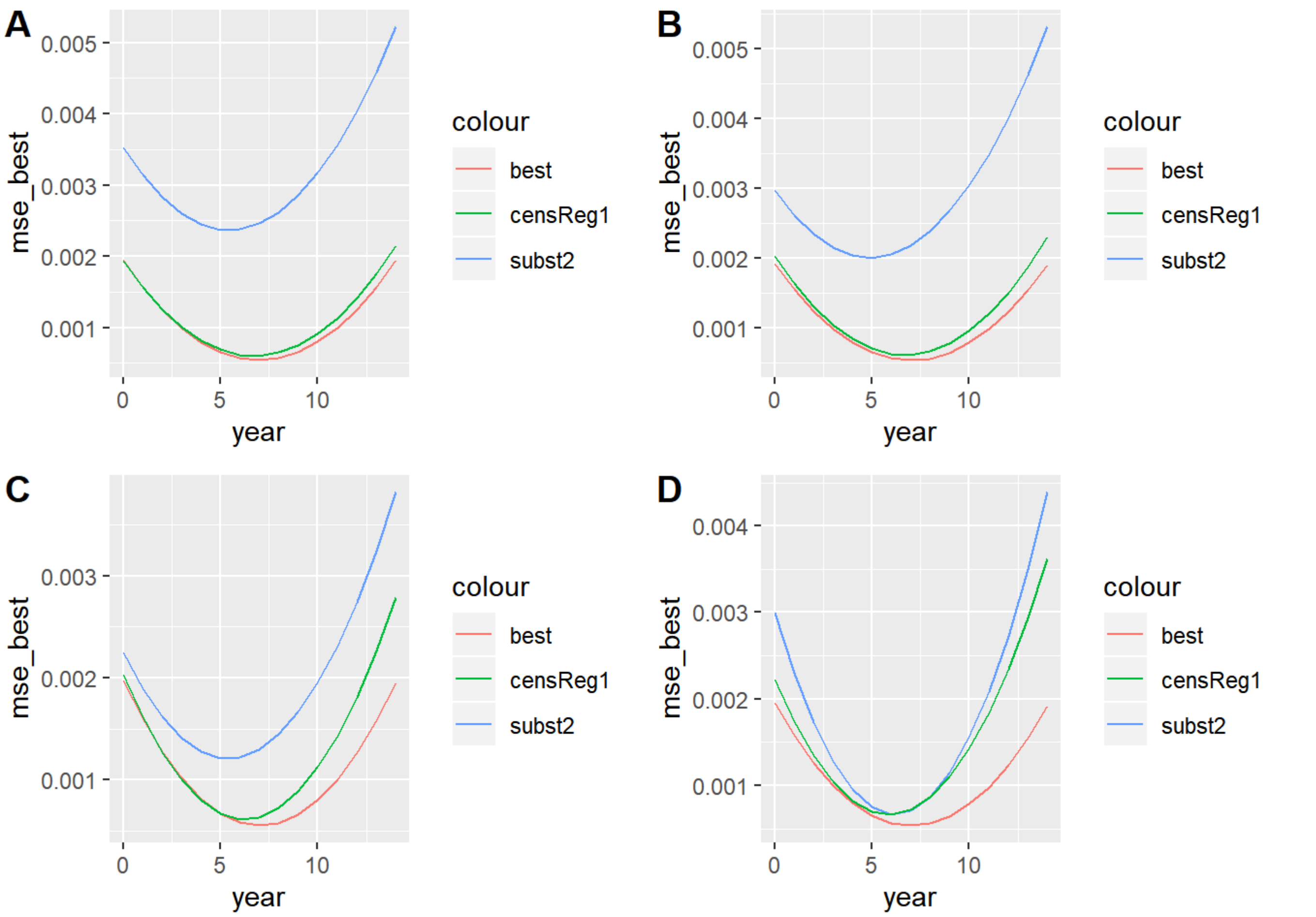
Our first set of four graphs show the MSE of `censReg1` and `censReg2` methods relative to `best` method for beta153year equal to -0.02, -0.04, -0.08, -0.16, respectively.



Our second set of four graphs show the MSE of `subst1`, `subst2` and `subst4` methods relative to `best` method for beta153year equal to -0.02, -0.04, -0.08, -0.16, respectively.



Our third set of four graphs simply displays the MSE from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

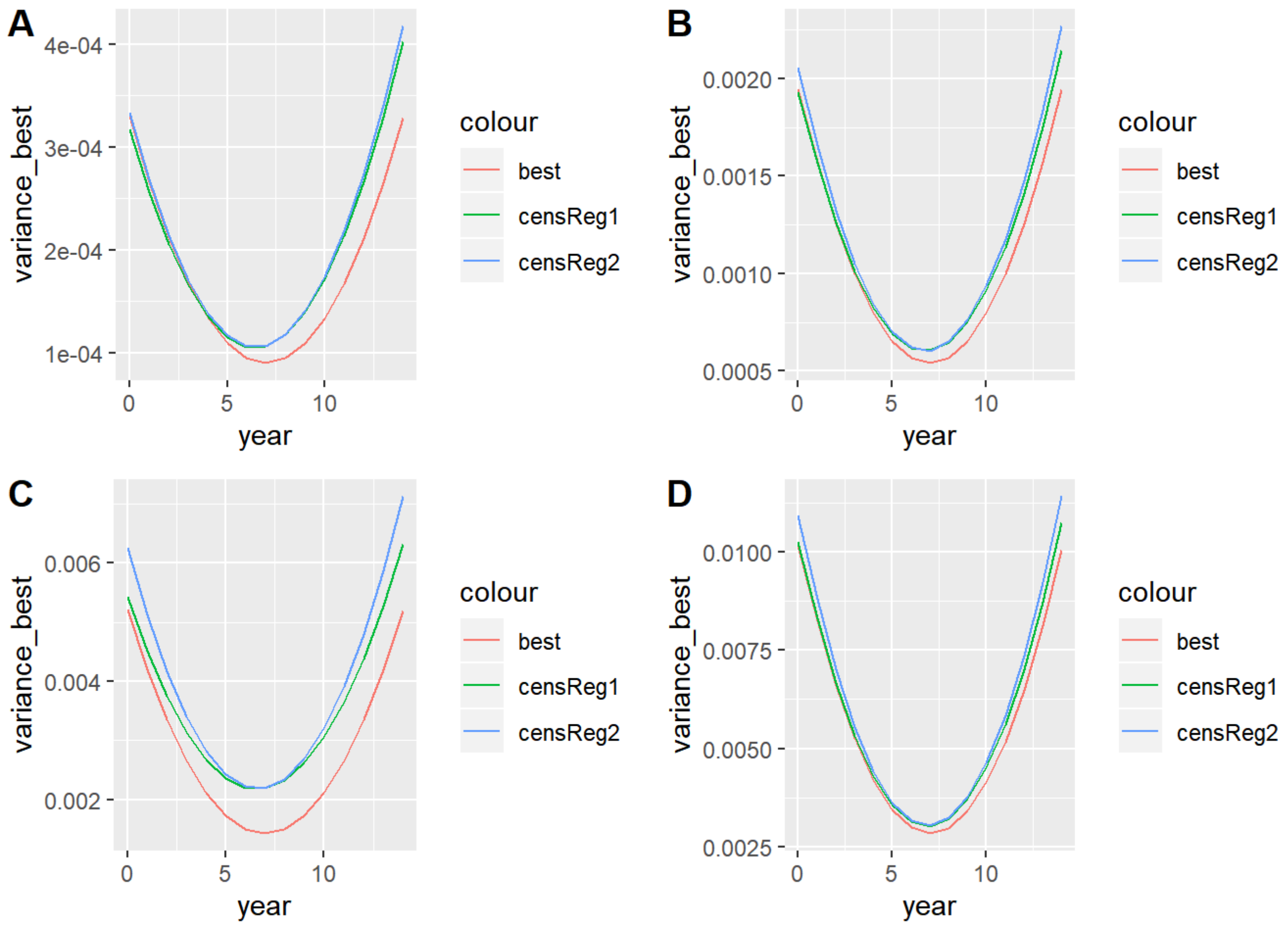


Predictions for different values of sd28vs153

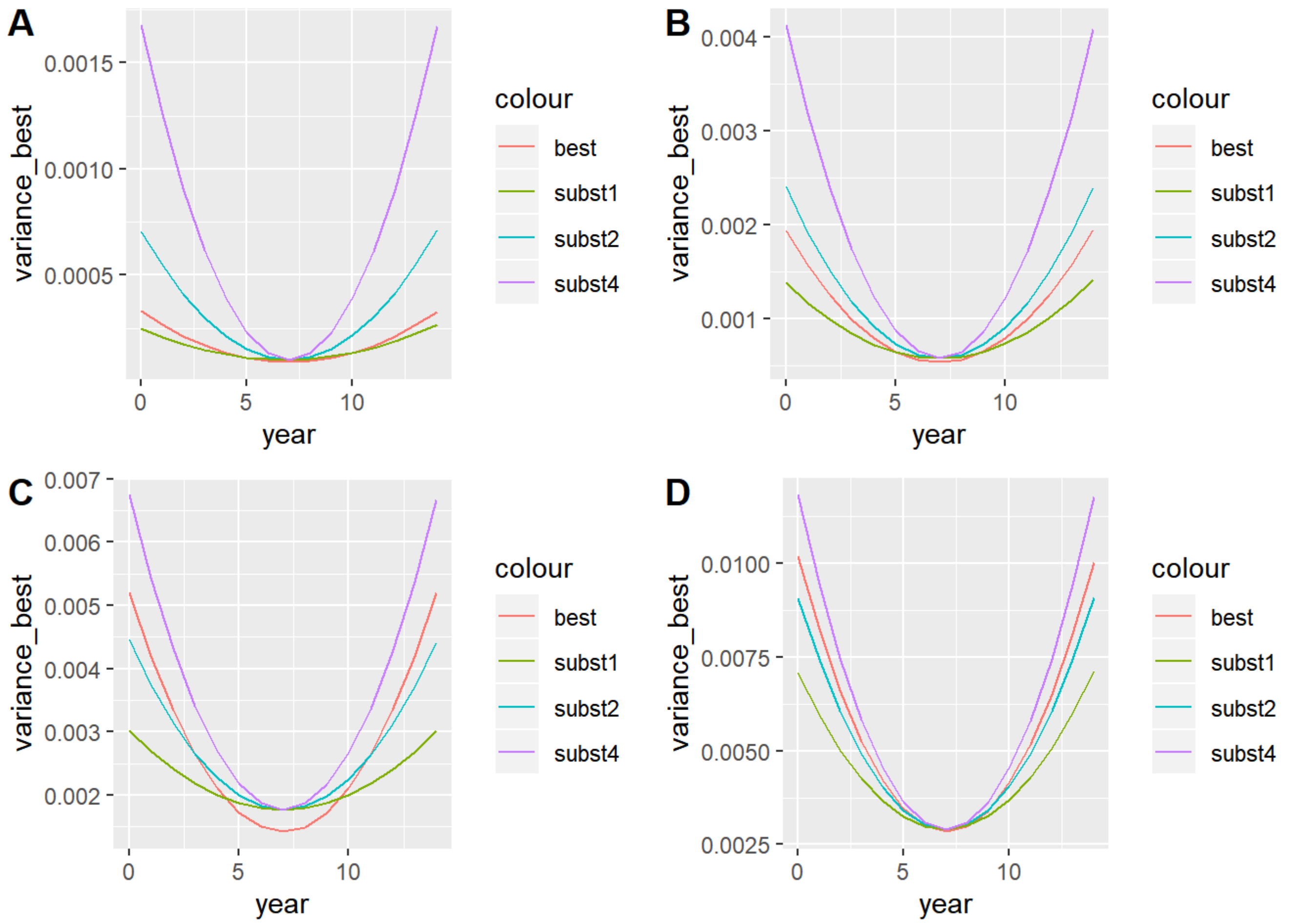
For all our predictions in this section, these parameters are fixed: cprop = 0.3, cb153year = -0.02, whilst sd28_153 is given four values: 0.1, 0.3, 0.5 and 0.7 respectively.

We begin by showing graphs of the variance of predictions of cb28 annual means from our chosen censoring methods. A common feature of all these graphs is that they typically have an approximately parabolic “U” shape, with higher variance at each end of the time period than in the middle of the period. This is in accordance with our prior expectations because this is generally the case.

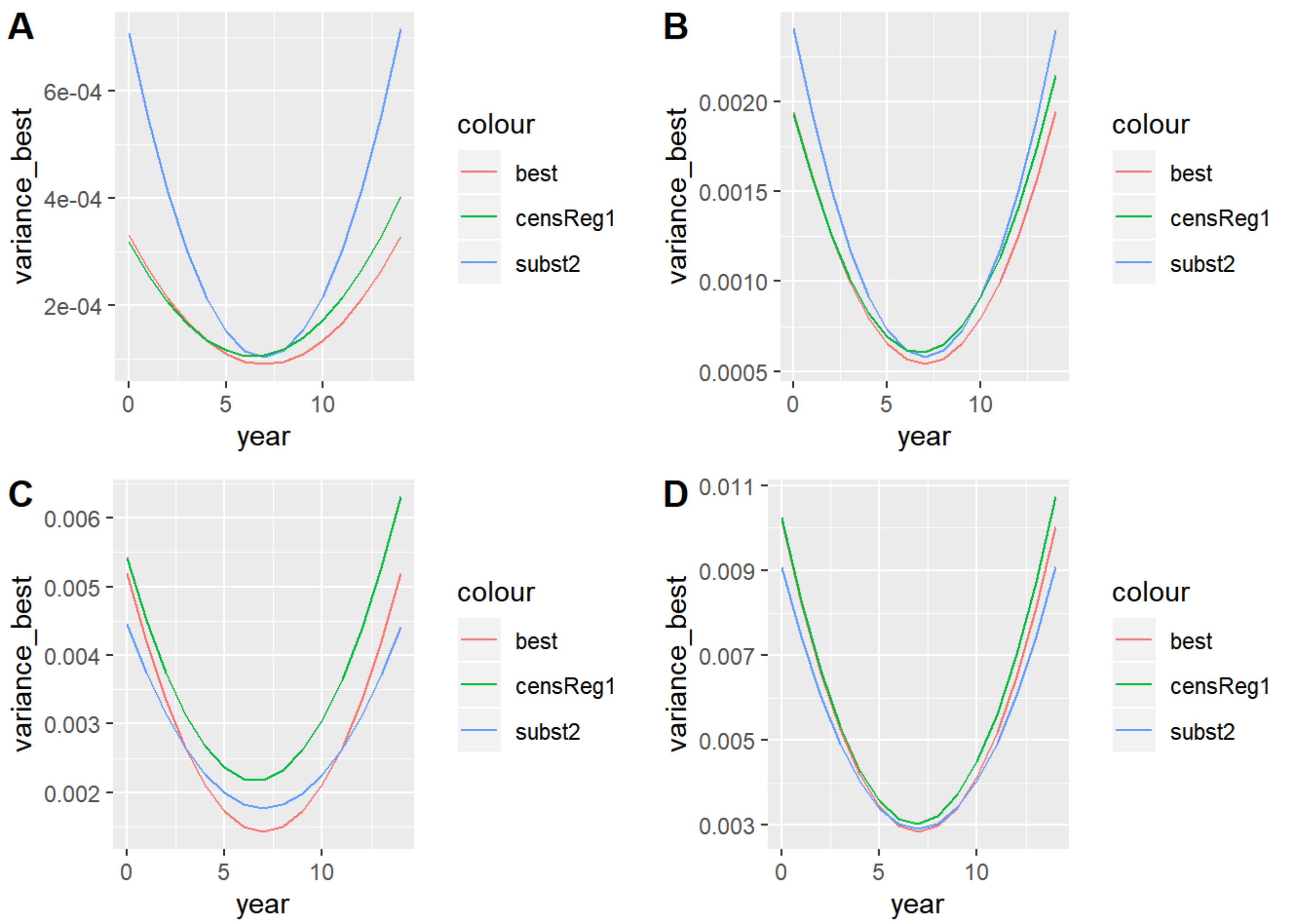
Our first set of four graphs show the variance of `censReg1` and `censReg2` methods relative to `best` method for sd28_153 equal to 0.1, 0.3, 0.5, 0.7, respectively.



Our second set of four graphs show the variance of `subst1` , `subst2` and `subst4` methods relative to `best` method for sd28_153 equal to 0.1, 0.3, 0.5, 0.7, respectively.

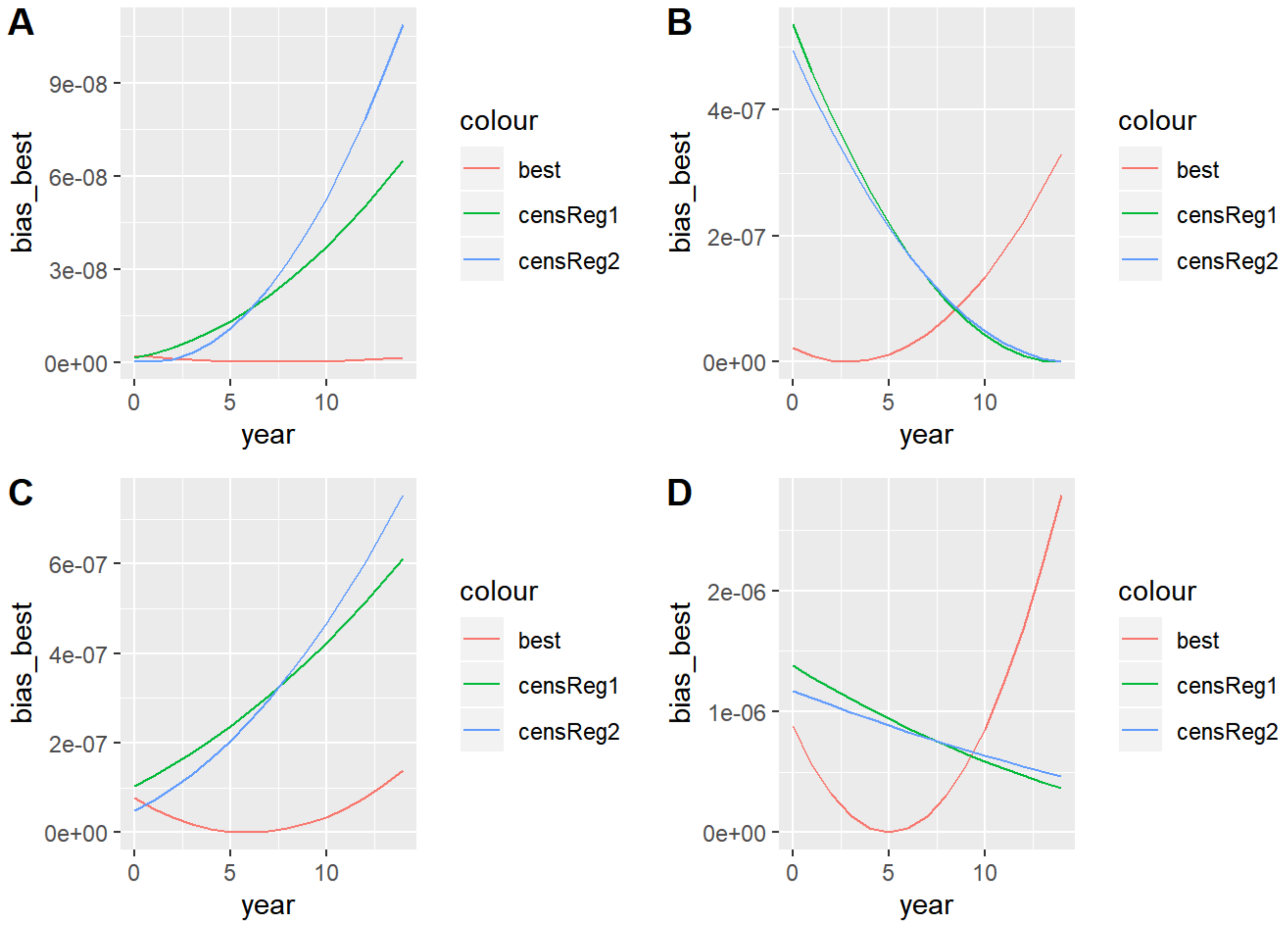


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

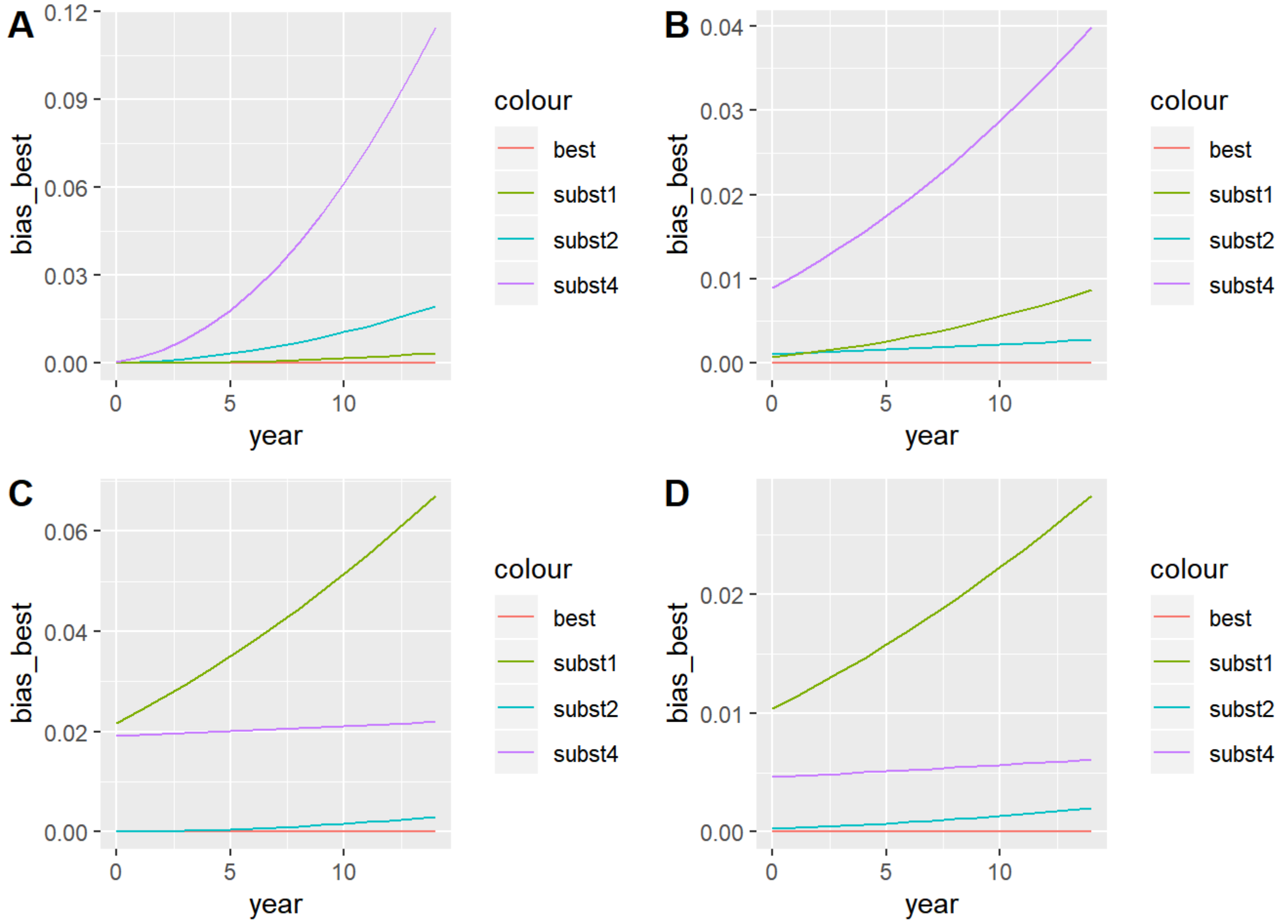


We will now show graphs of the bias of predictions of cb28 annual means from our chosen censoring methods.

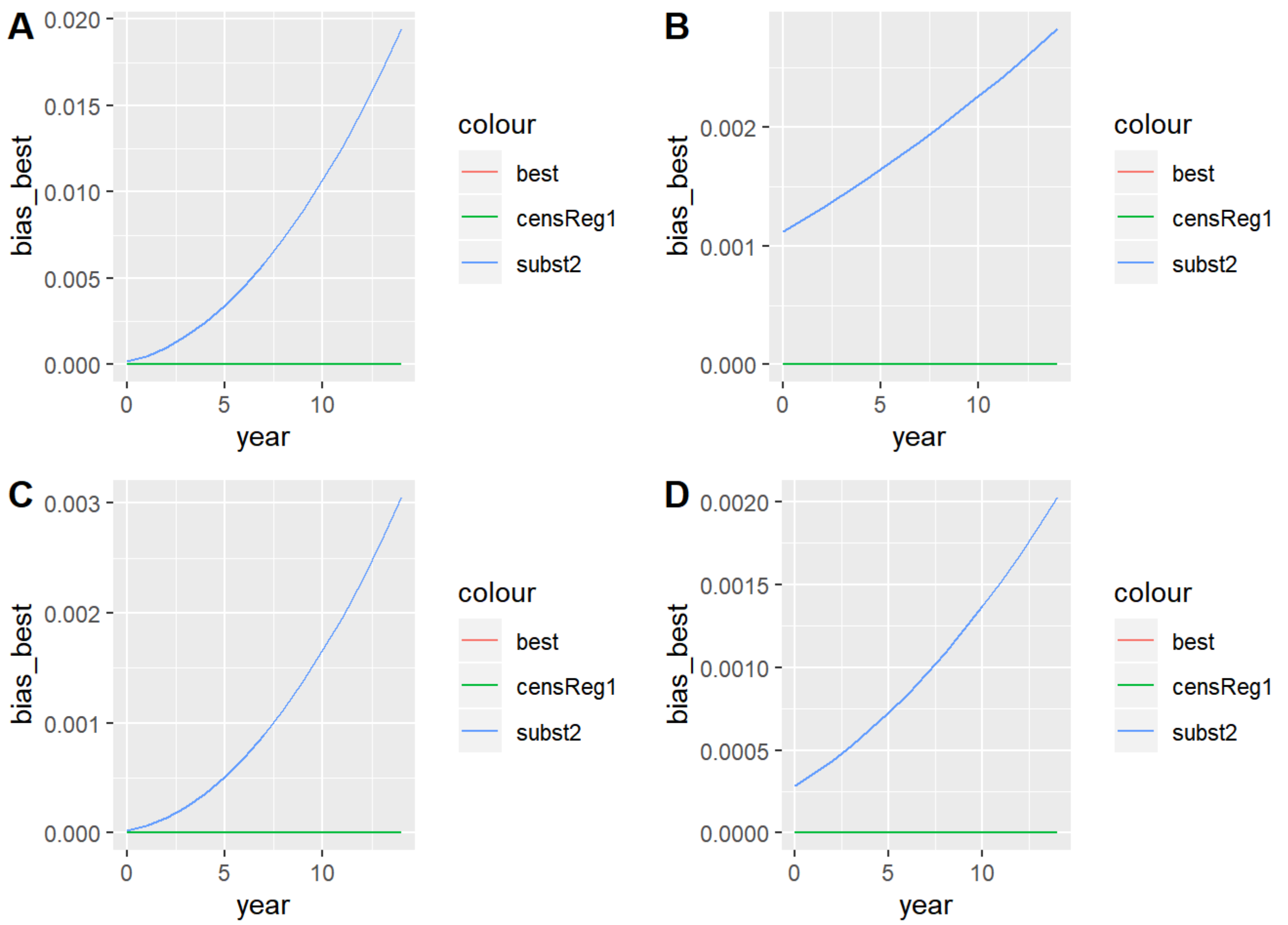
Our first set of four graphs show the bias of `censReg1` and `censReg2` methods relative to `best` method for sd28_153 equal to 0.1, 0.3, 0.5, 0.7, respectively.



Our second set of four graphs show the bias of `subst1`, `subst2` and `subst4` methods relative to `best` method for sd28_153 equal to 0.1, 0.3, 0.5, 0.7, respectively.

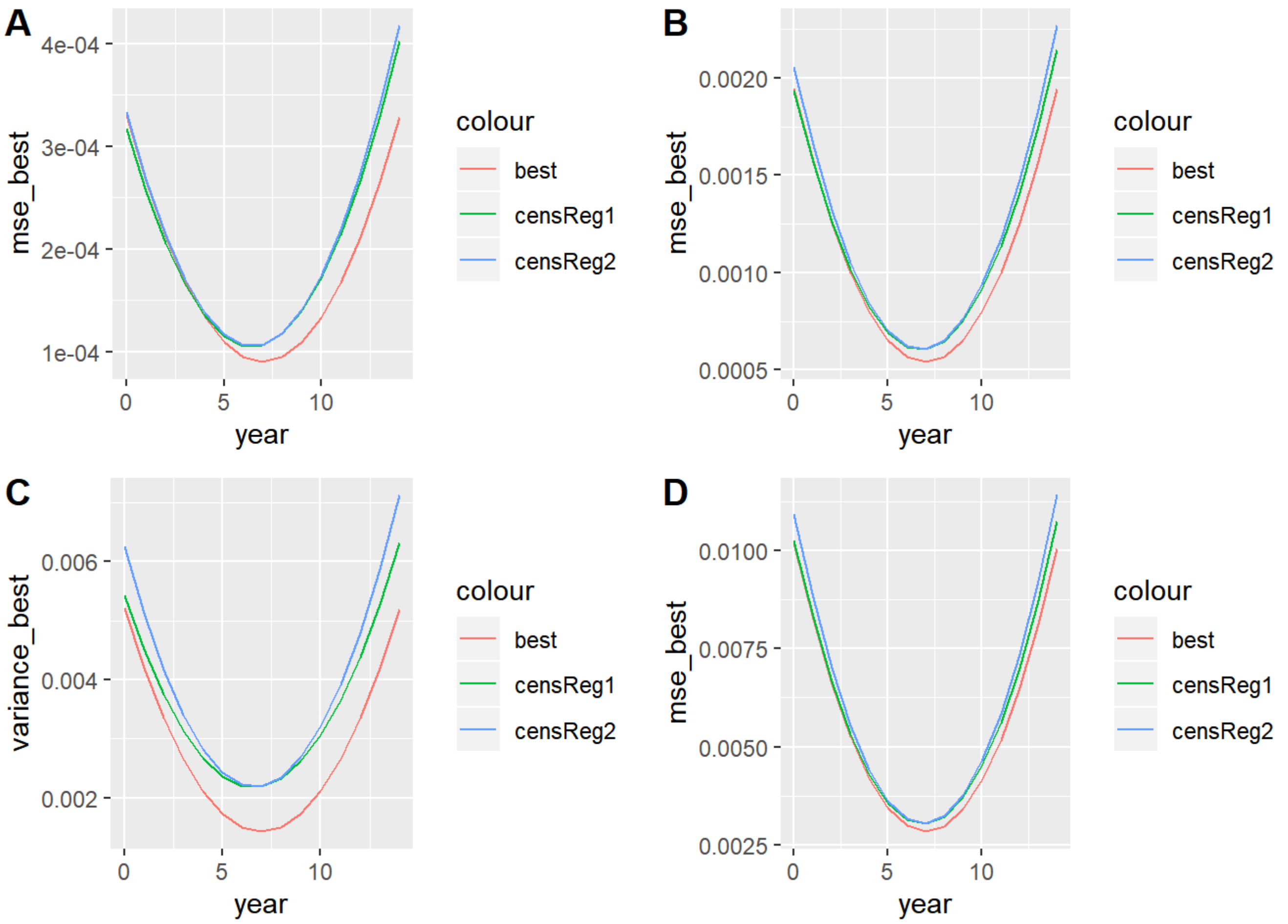


Our third set of four graphs simply displays the results from the `subst2` , `censReg1` and `best` methods together on the same plot, which is displayed below.

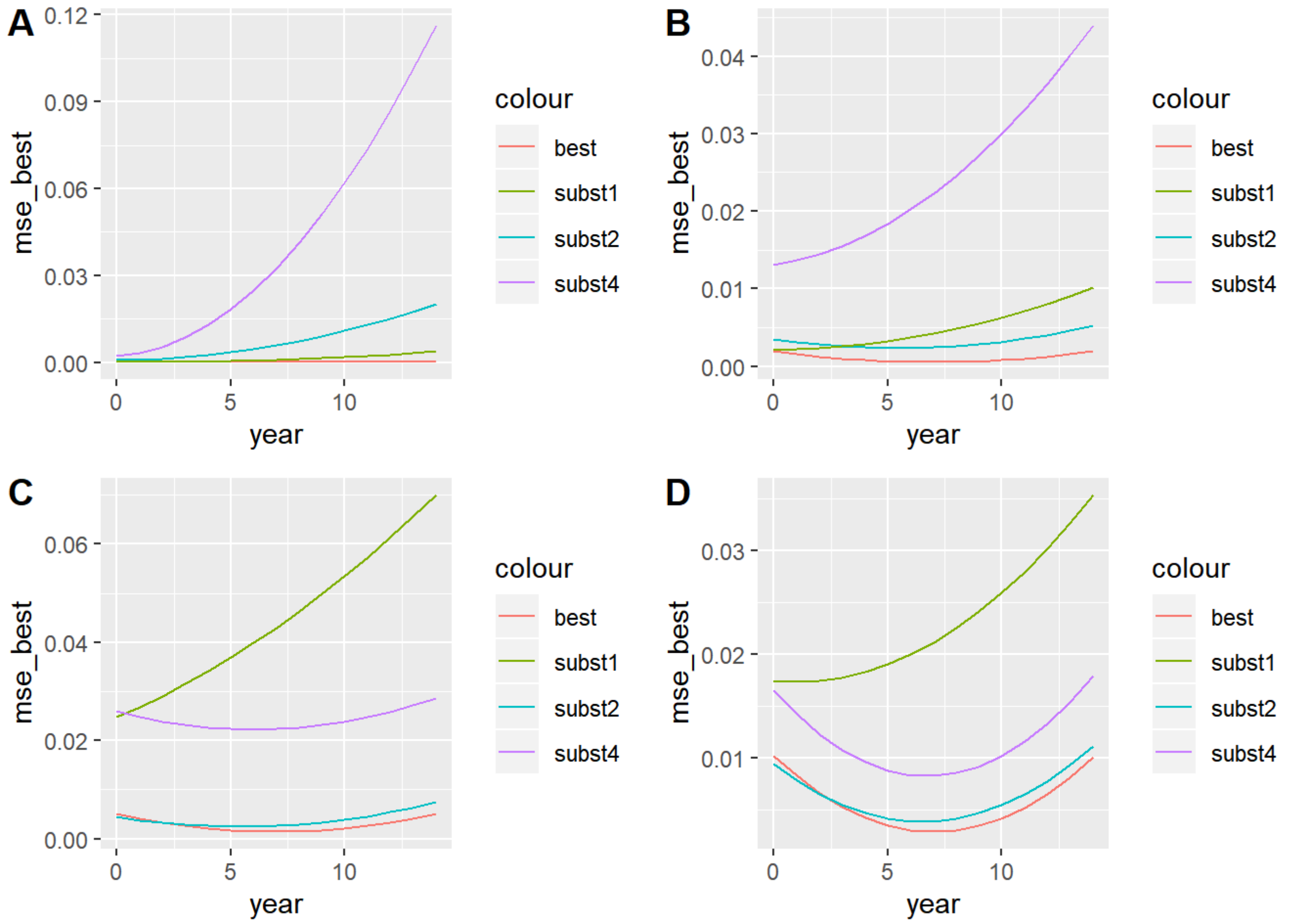


We will now show graphs of the MSE of predictions of cb28 annual means from our chosen censoring methods.

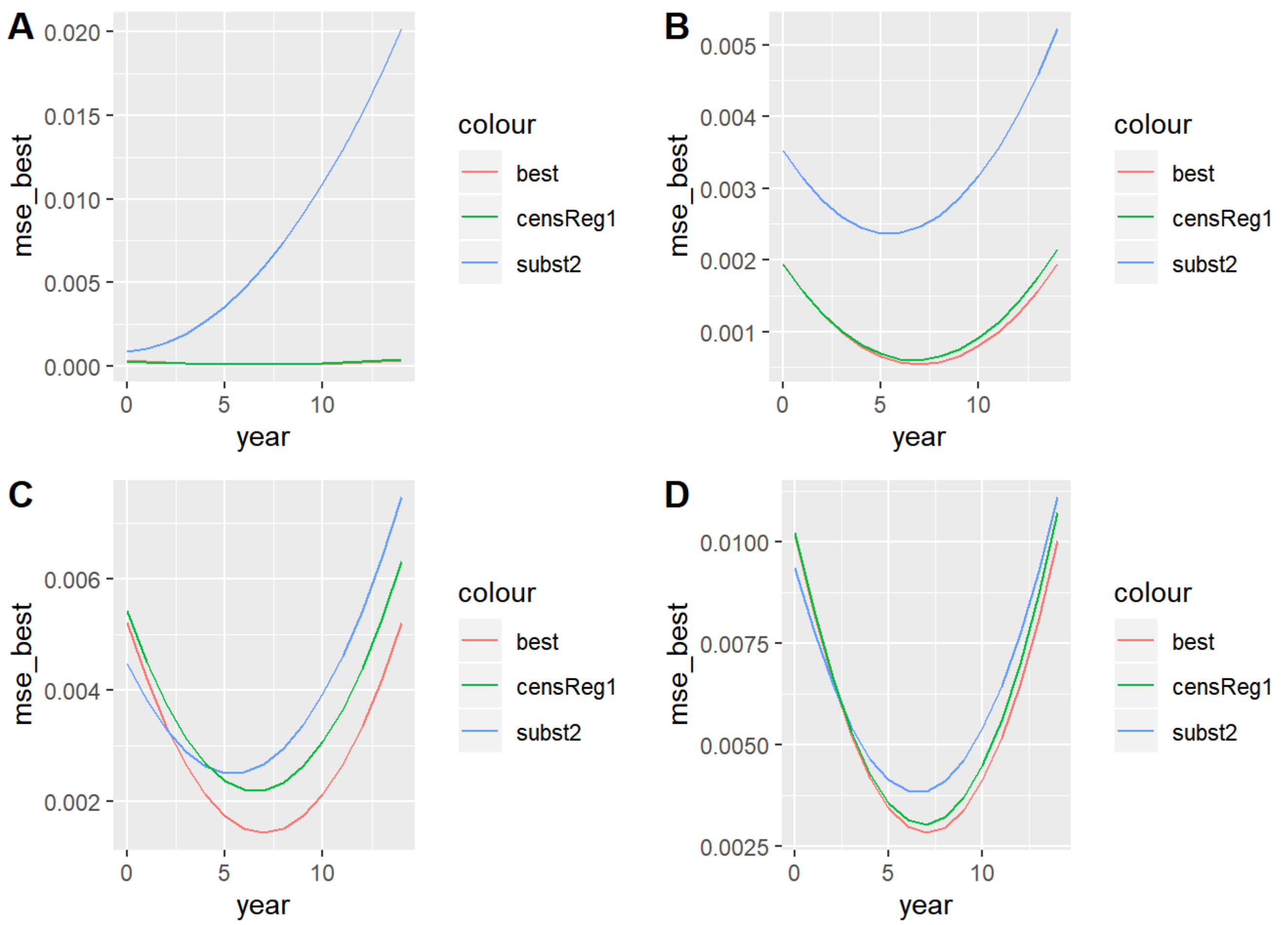
Our first set of four graphs show the MSE of `censReg1` and `censReg2` methods relative to `best` method for sd28_153 equal to 0.1, 0.3, 0.5, 0.7, respectively.



Our second set of four graphs show the MSE of `subst1`, `subst2` and `subst4` methods relative to `best` method for sd28_153 equal to 0.1, 0.3, 0.5, 0.7, respectively.



Our third set of four graphs simply displays the MSE from the `subst2` , `censReg1` and `best` methods together on the same plot, which is displayed below.

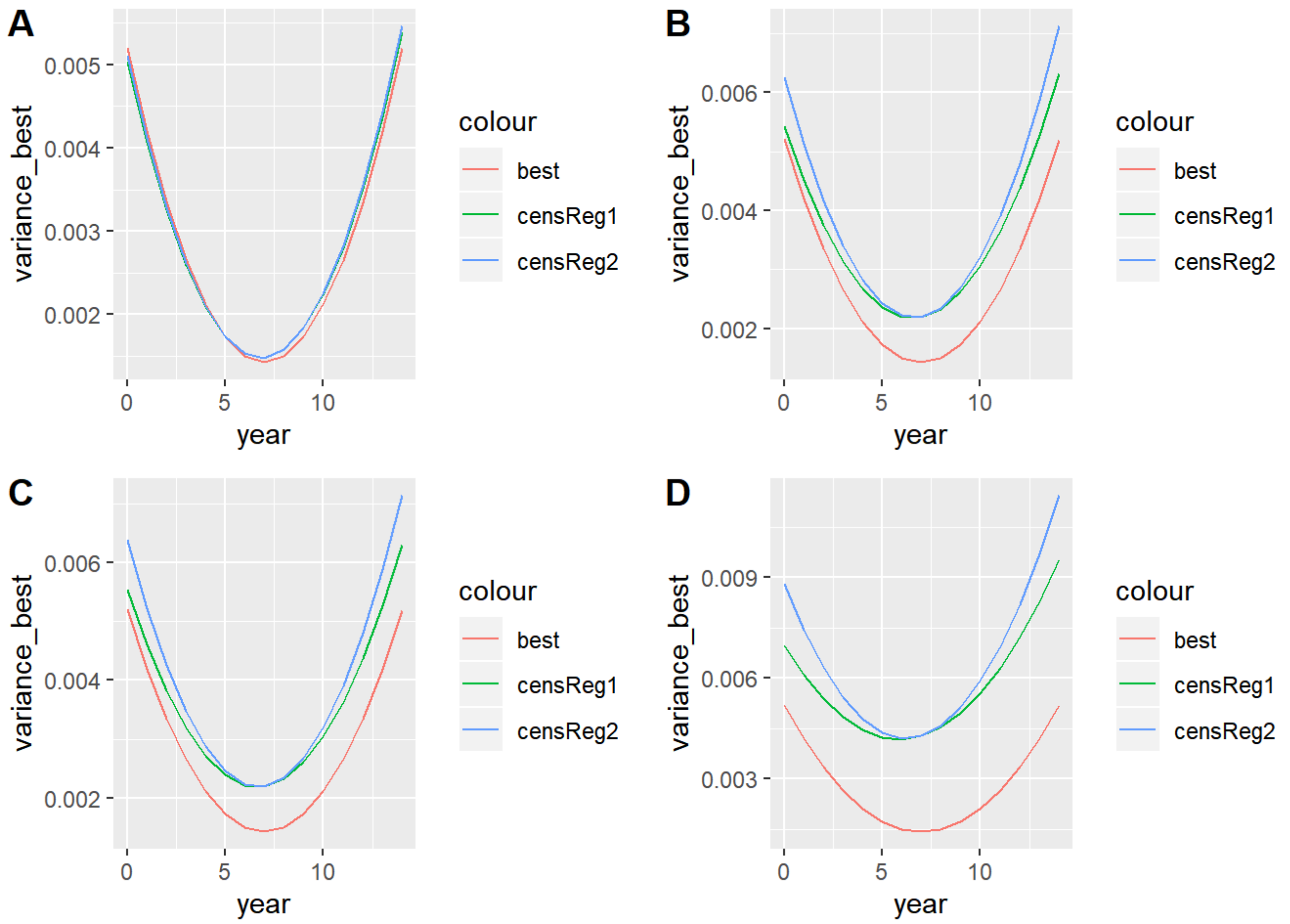


Predictions for different values of cprop

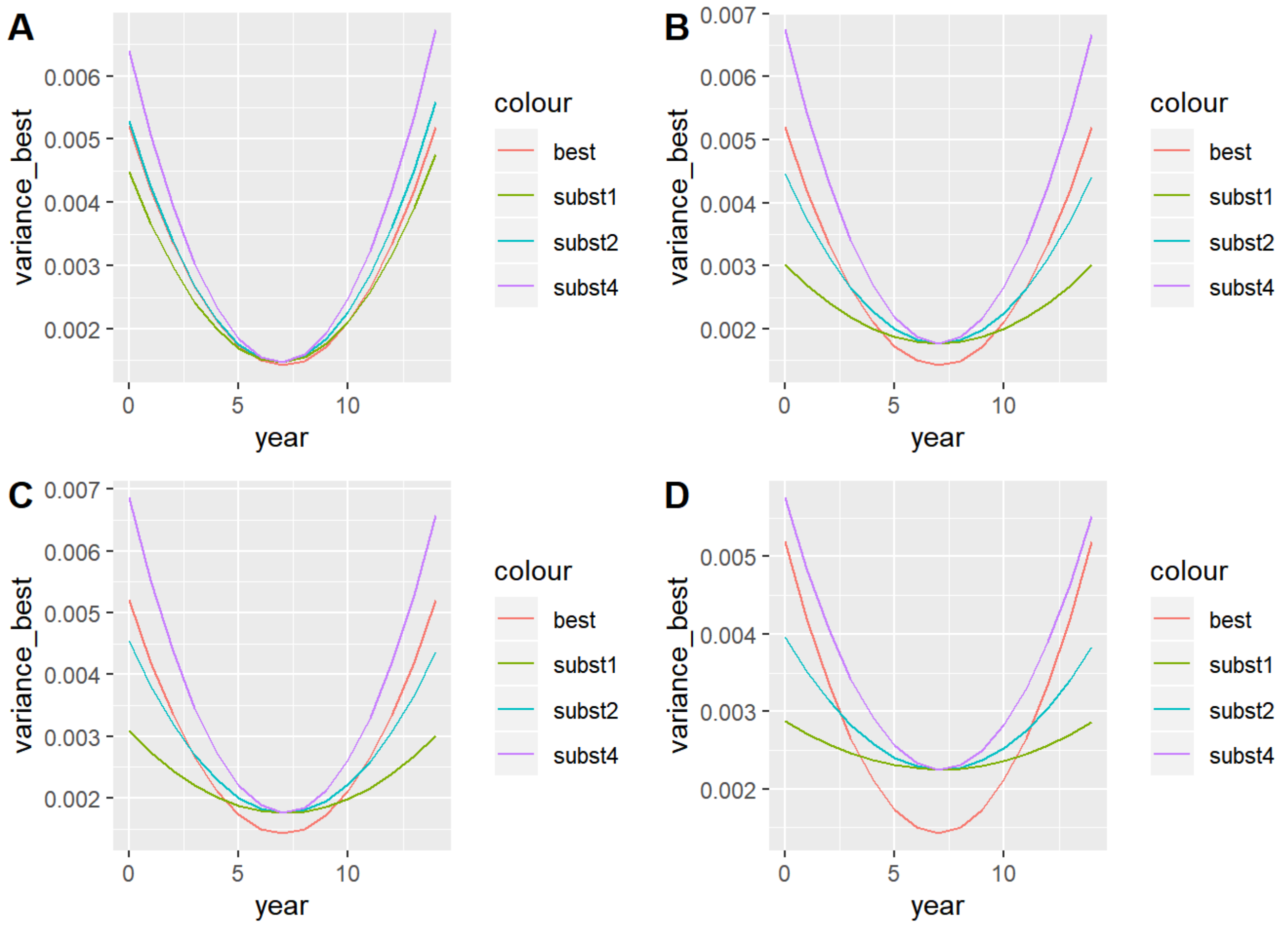
For all our predictions in this section, these parameters are fixed: sd28_153 = 0.5, cb153year = -0.02, whilst cprop is given four values: 0.1, 0.3, 0.5 and 0.7 respectively.

We begin by showing graphs of the variance of predictions of cb28 annual means from our chosen censoring methods.

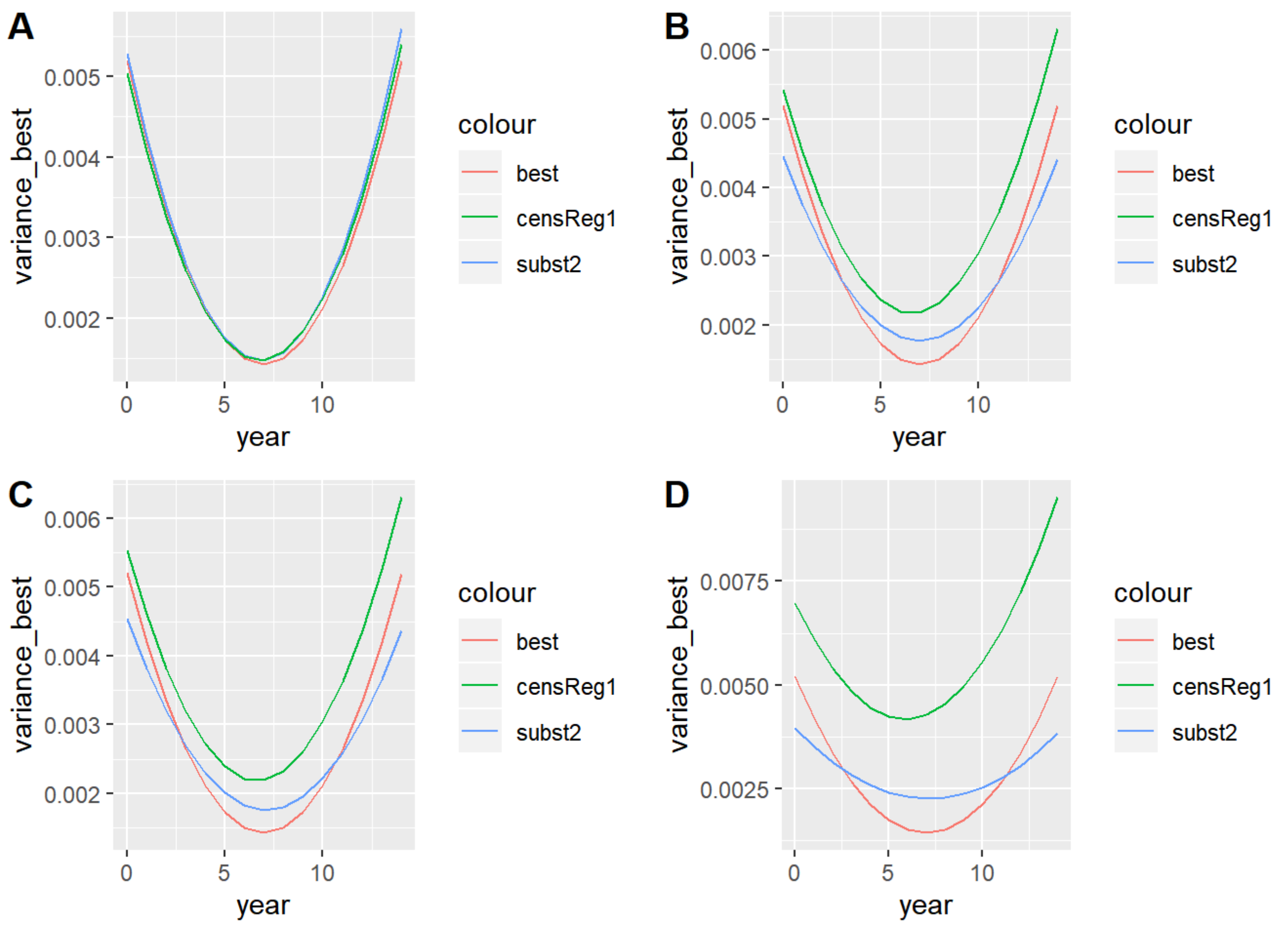
Our first set of four graphs show the variance of `censReg1` and `censReg2` methods relative to `best` method for cprop equal to 0.1, 0.3, 0.5, 0.7, respectively.



Our second set of four graphs show the variance of `subst1`, `subst2` and `subst4` methods relative to `best` method for cprop equal to 0.1, 0.3, 0.5, 0.7, respectively.

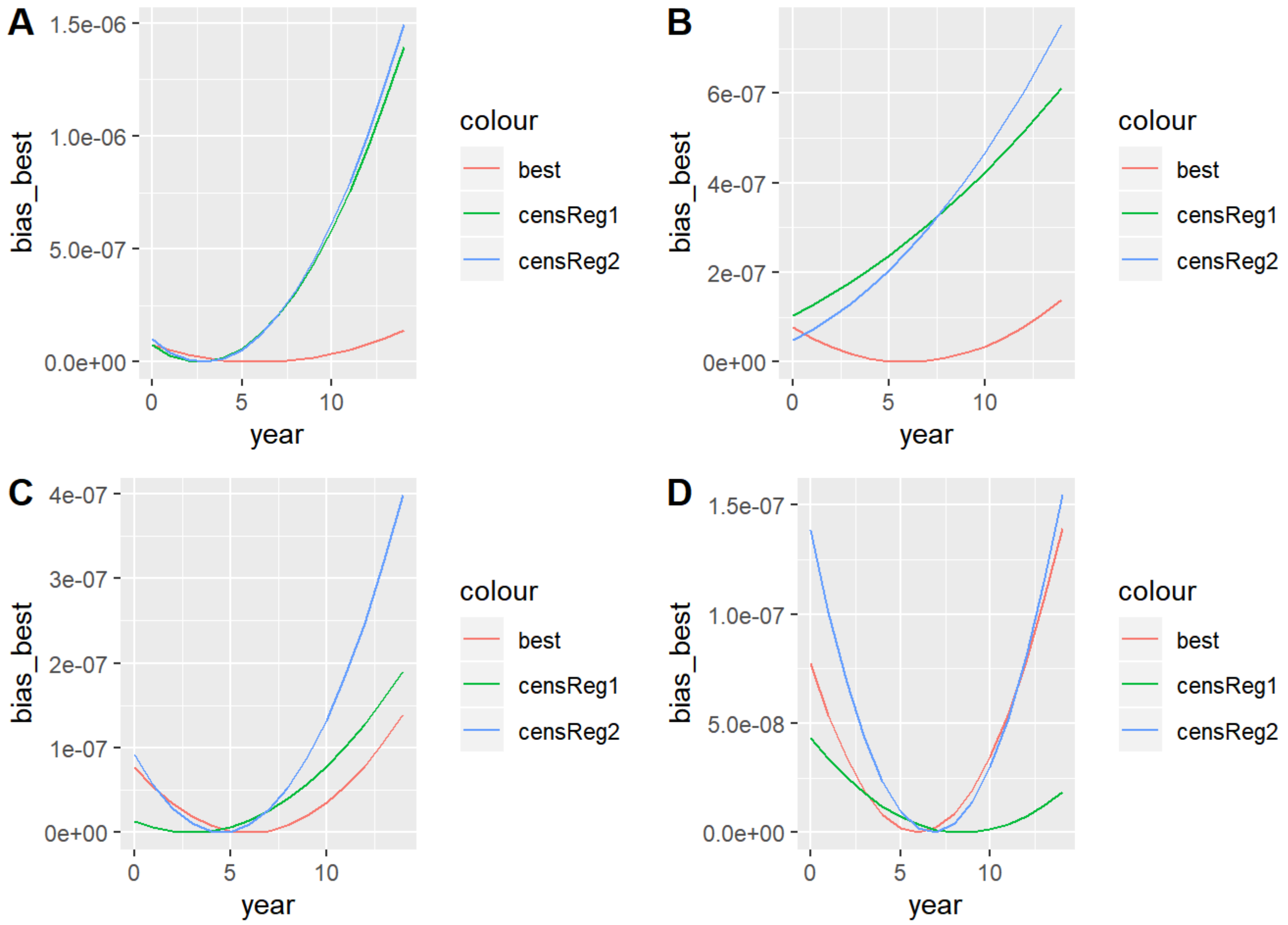


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

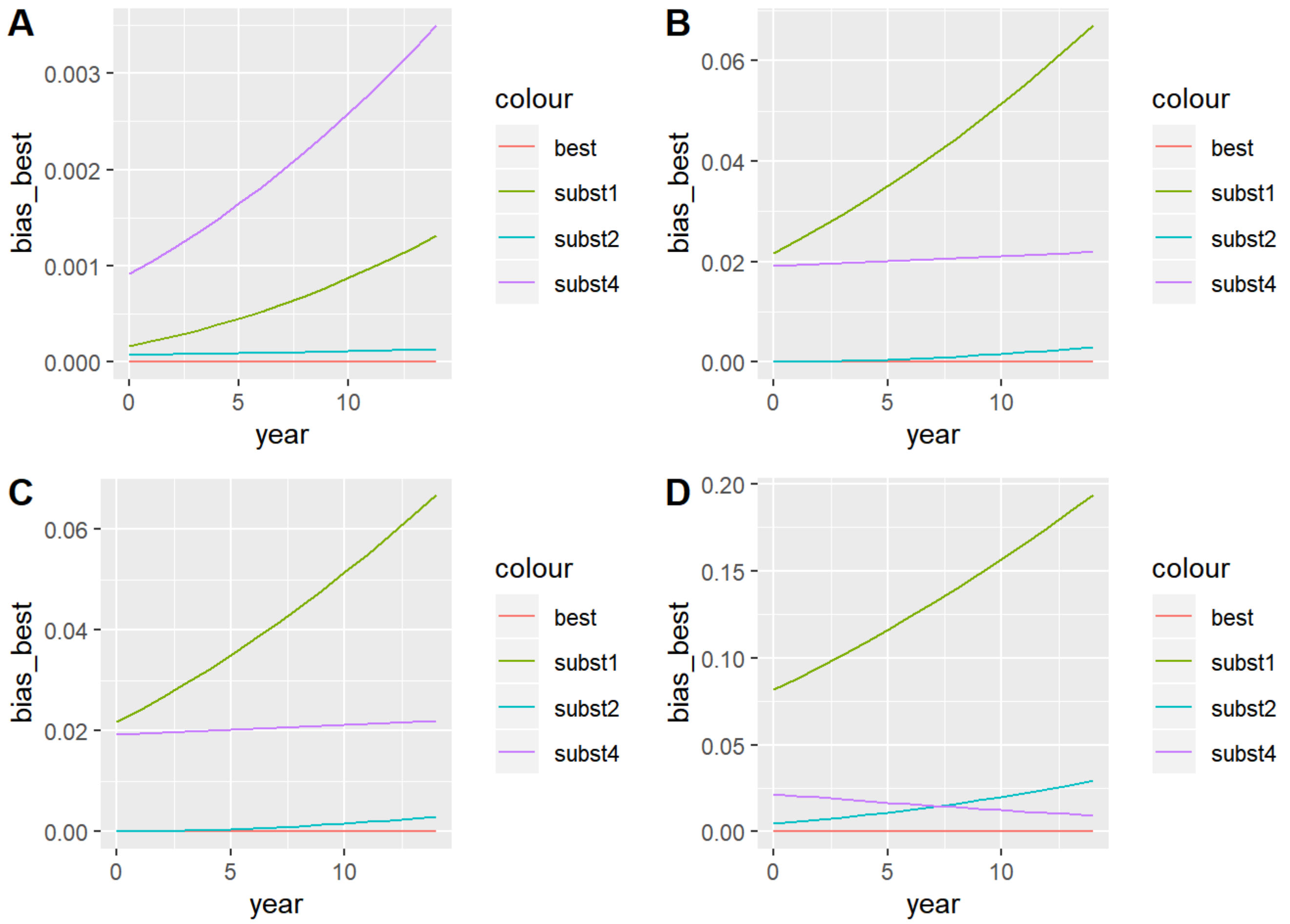


We will now show graphs of the bias of predictions of cb28 annual means from our chosen censoring methods.

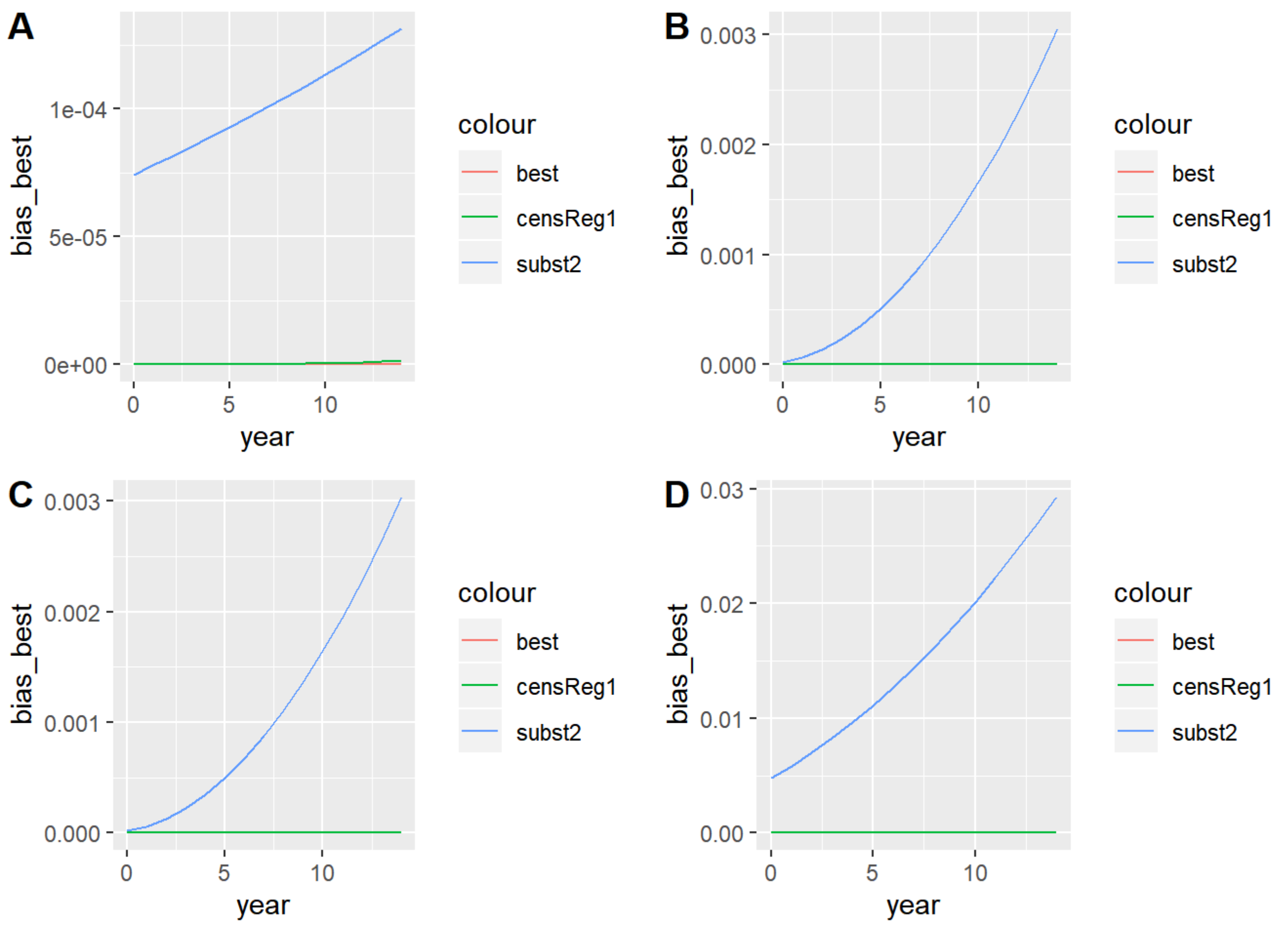
Our first set of four graphs show the bias of `censReg1` and `censReg2` methods relative to `best` method for cprop equal to 0.1, 0.3, 0.5, 0.7, respectively.



Our second set of four graphs show the bias of `subst1`, `subst2` and `subst4` methods relative to `best` method for cprop equal to 0.1, 0.3, 0.5, 0.7, respectively.

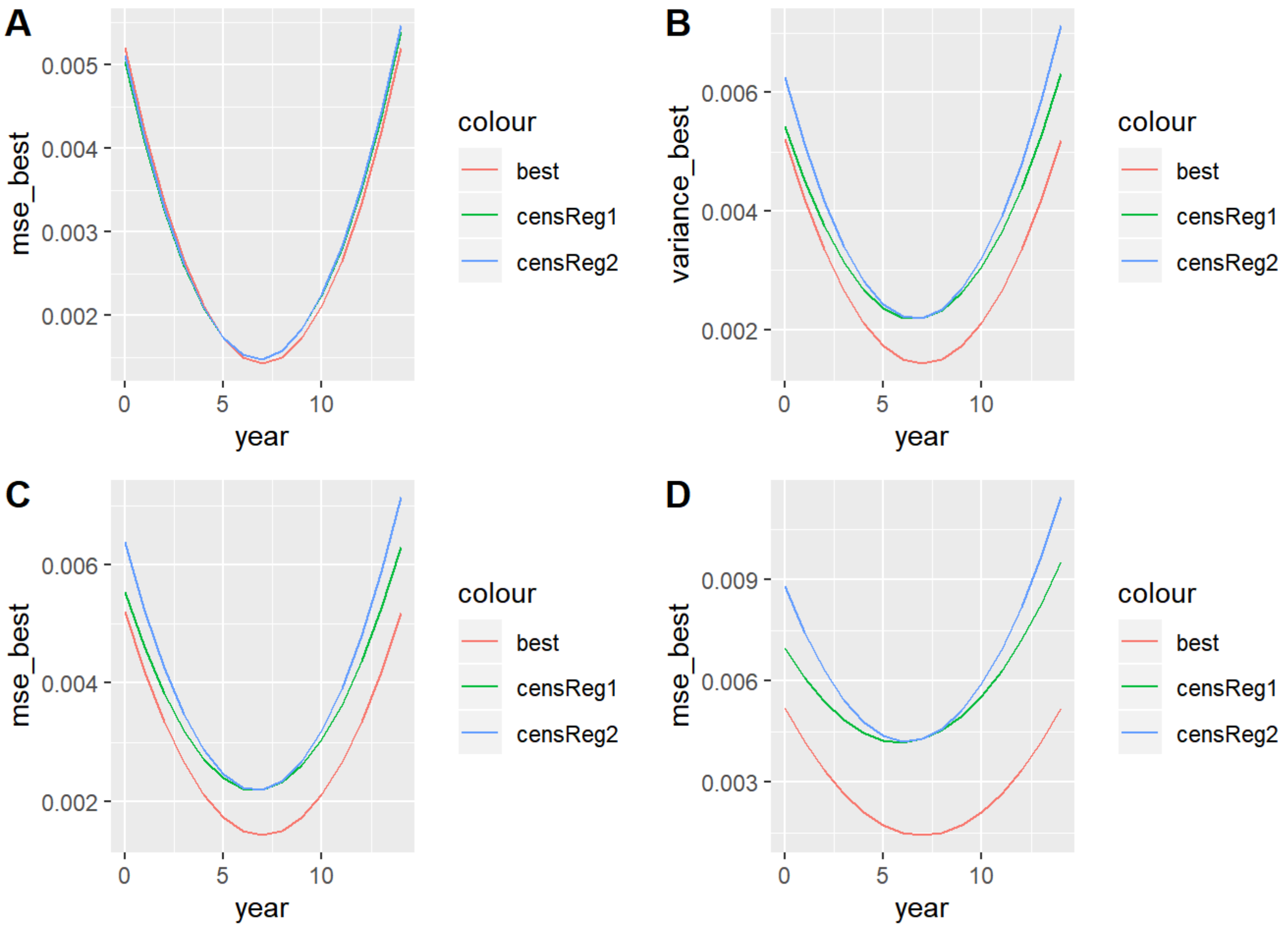


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

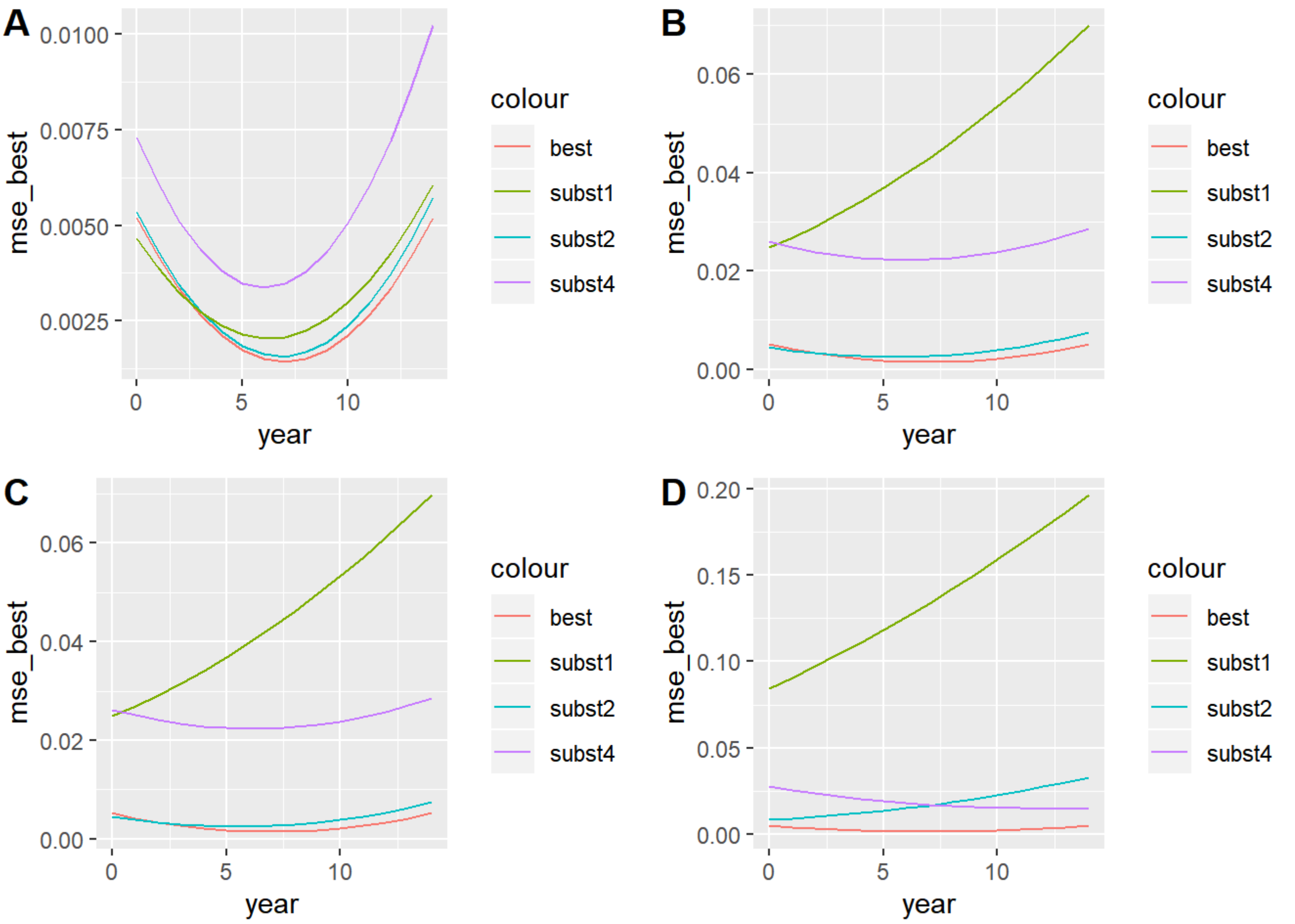


We will now show graphs of the MSE of predictions of cb28 annual means from our chosen censoring methods.

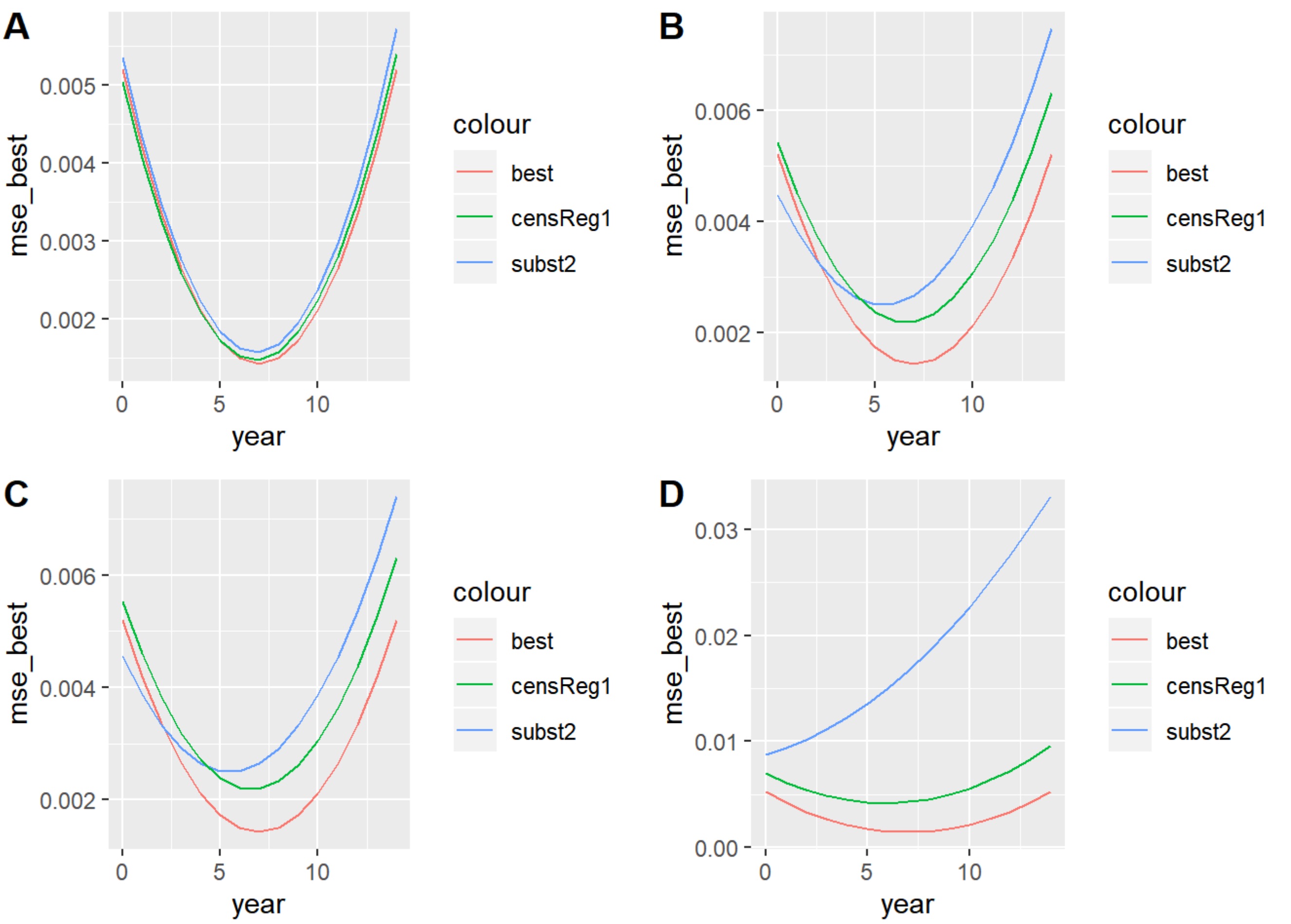
Our first set of four graphs show the MSE of `censReg1` and `censReg2` methods relative to `best` method for cprop equal to 0.1, 0.3, 0.5, 0.7, respectively.



Our second set of four graphs show the MSE of `subst1` , `subst2` and `subst4` methods relative to `best` method for cprop equal to 0.1, 0.3, 0.5, 0.7, respectively.



Our third set of four graphs simply displays the MSE from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

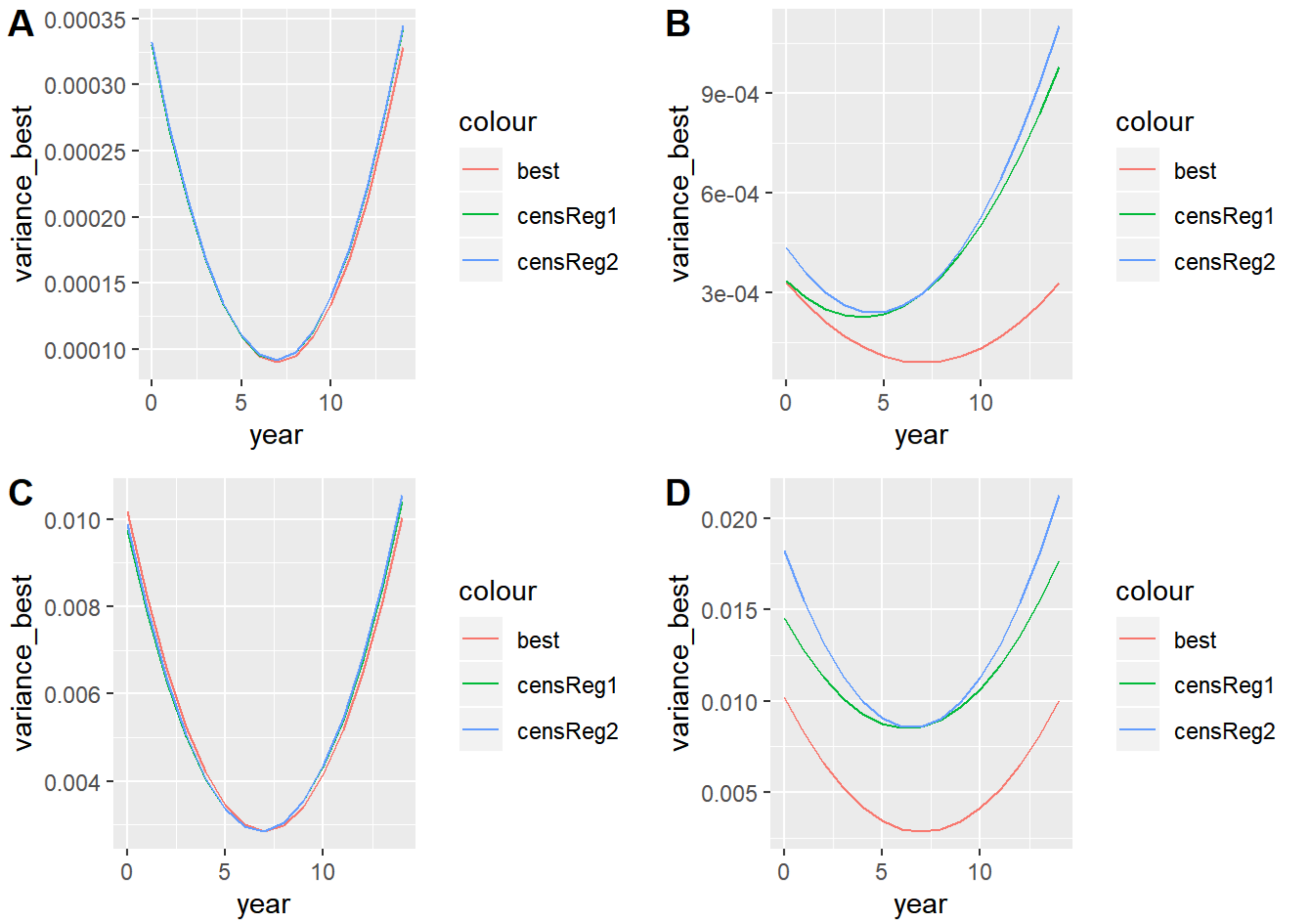


Predictions for low-low, high-low, low-high, high-high values of sd28vs153-cprop

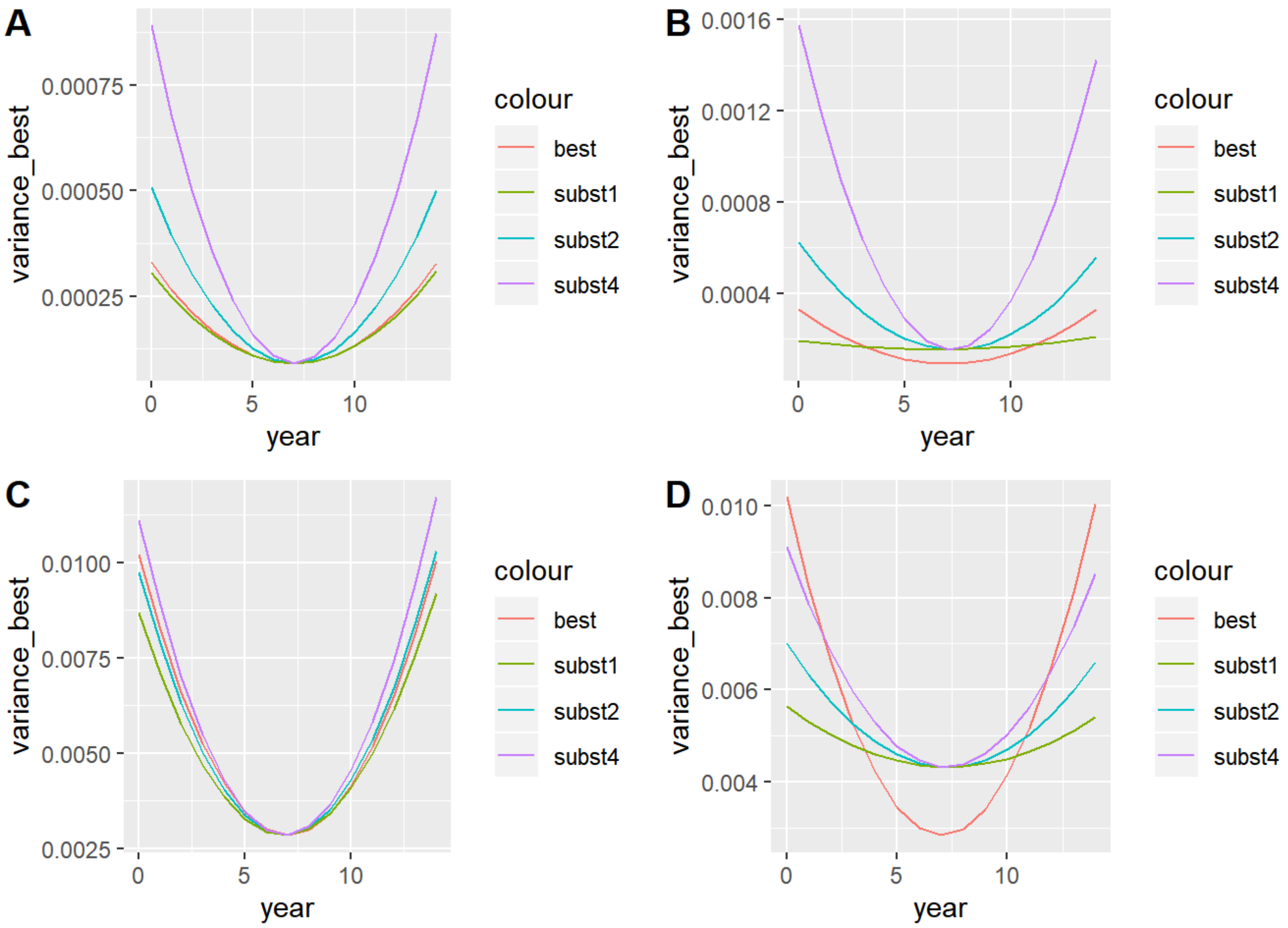
We will now use the same sets of parameter values that we used in our earlier section "Selection of censoring methods for further study". Concretely: beta28year = -0.02 is held fixed, whilst a "low" and a "high" value for each of `cprop` and `sd28_153` are used. Concretely: (0.1, 0.1), (0.7, 0.1), (0.1, 0.5) and (0.7, 0.5) were used for (cprop, sd28_153) respectively.

We begin by showing graphs of the variance of predictions of cb28 annual means from our chosen censoring methods.

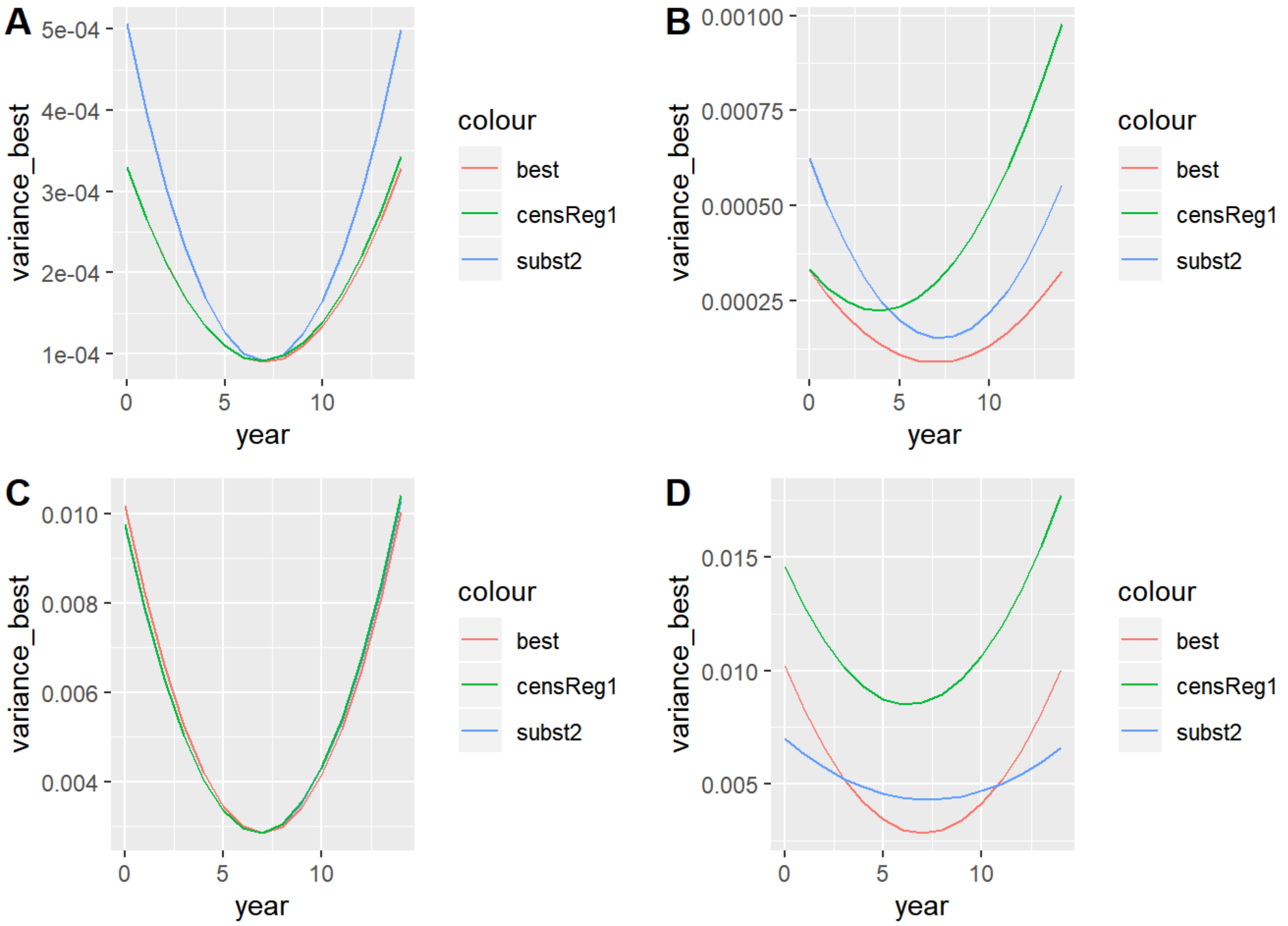
Our first set of four graphs show the variance of `censReg1` and `censReg2` methods relative to `best` method for (sd28_153, cprop) equal to (0.1, 0.1), (0.1, 0.7), (0.7, 0.1) and (0.7, 0.7), respectively.



Our second set of four graphs show the variance of `subst1`, `subst2` and `subst4` methods relative to `best` method for (sd28_153, cprop) equal to (0.1, 0.1), (0.1, 0.7), (0.7, 0.1) and (0.7, 0.7), respectively.

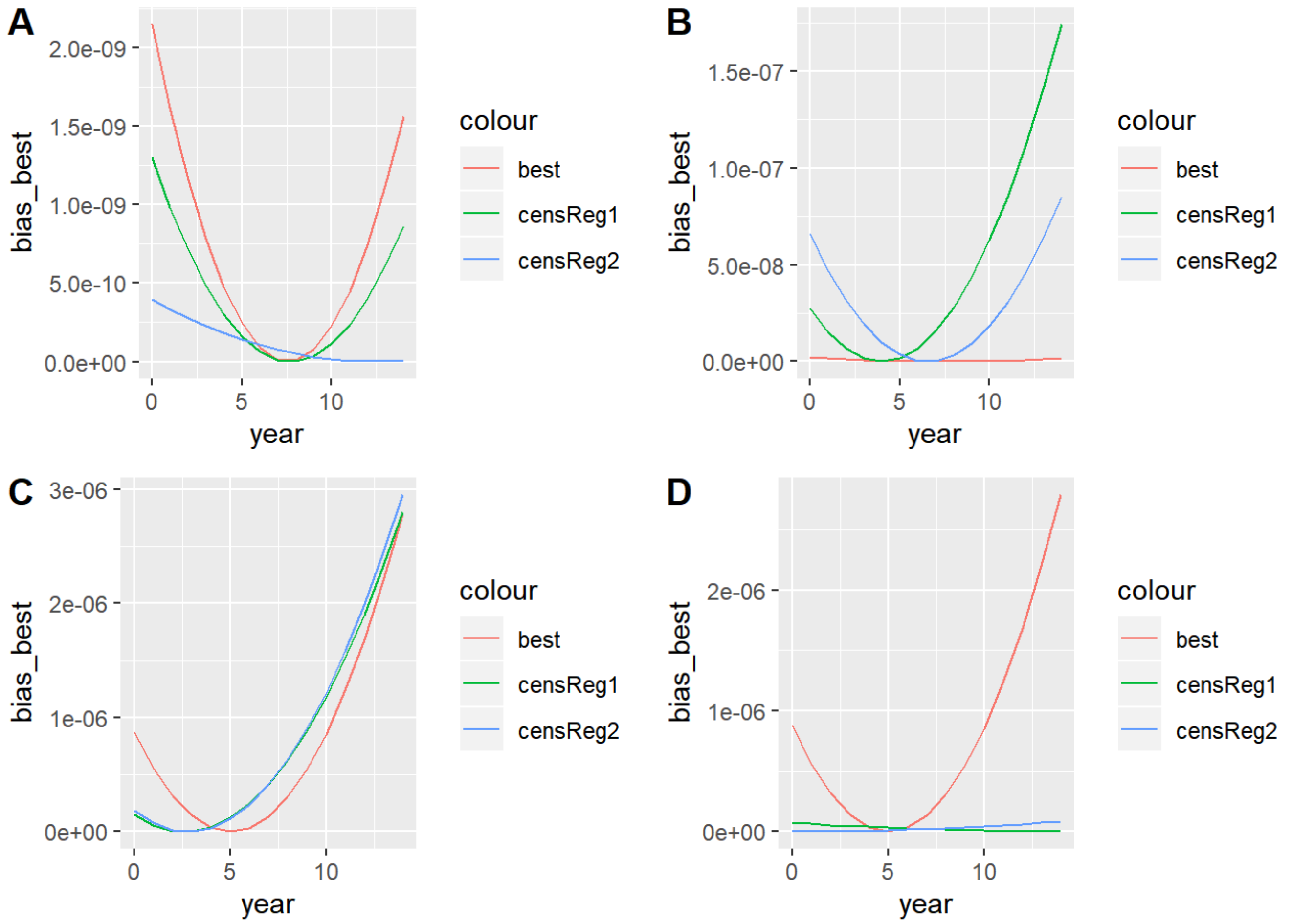


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

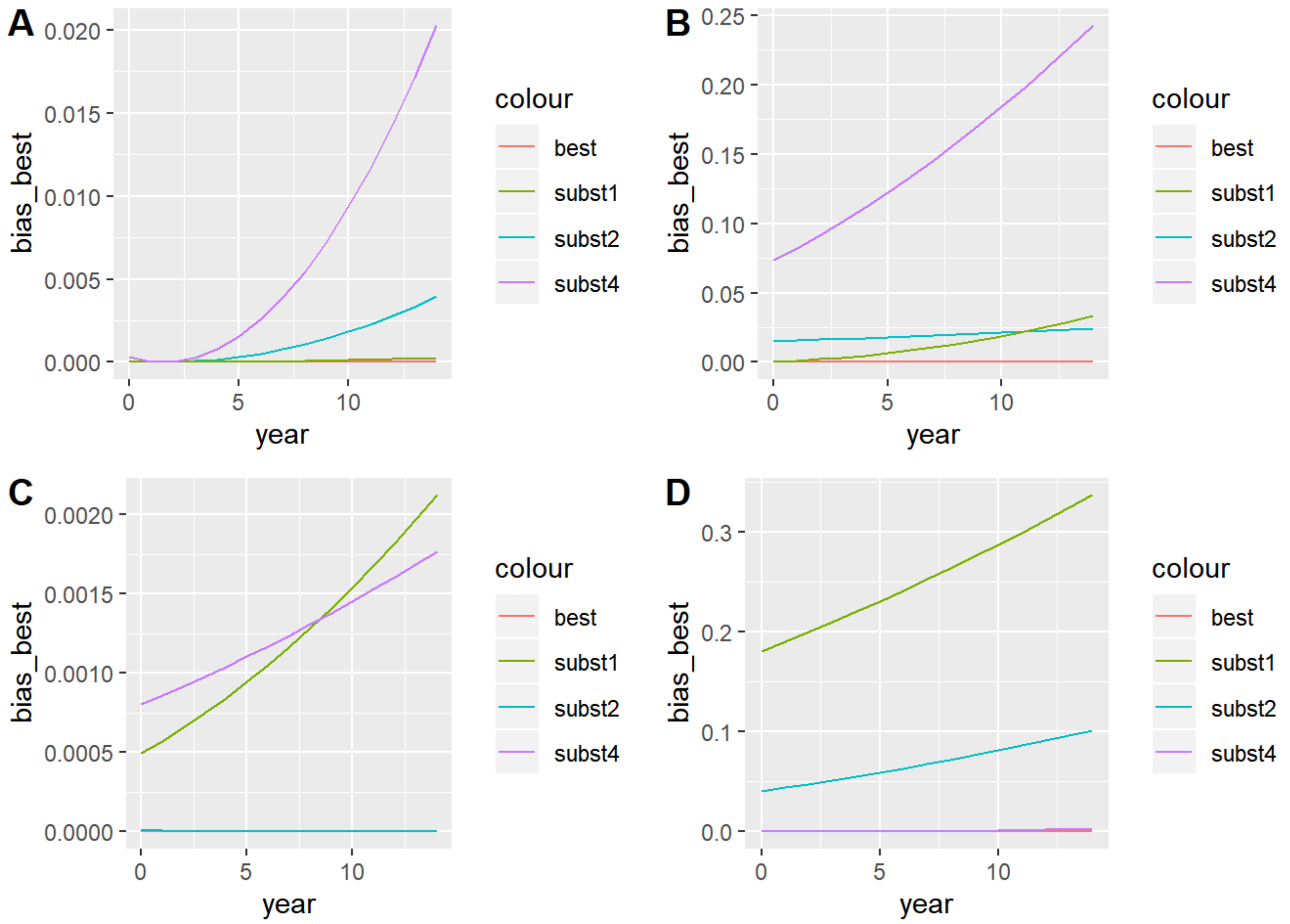


We will now show graphs of the bias of predictions of cb28 annual means from our chosen censoring methods.

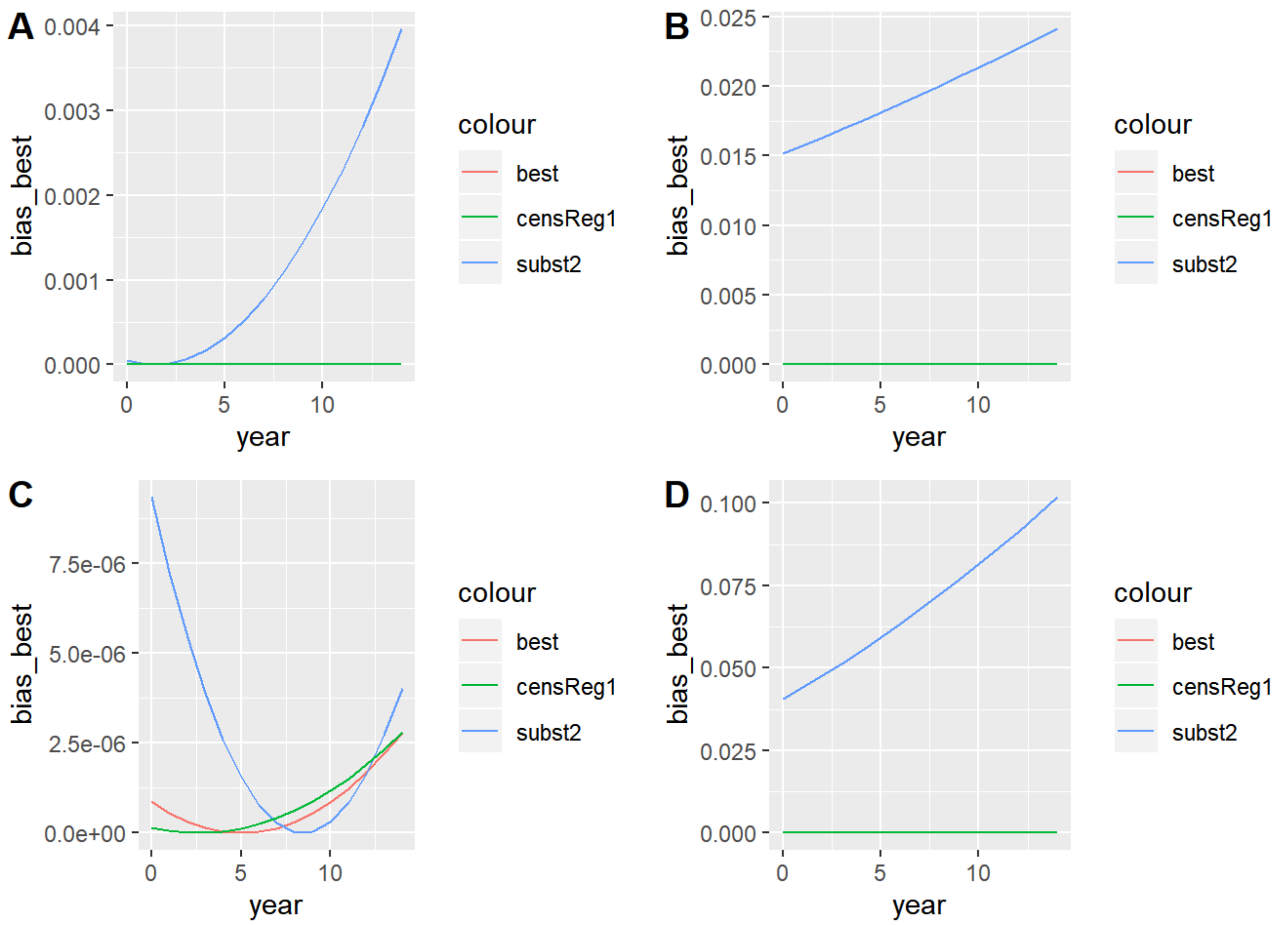
Our first set of four graphs show the bias of `censReg1` and `censReg2` methods relative to `best` method for (sd28_153, cprop) equal to (0.1, 0.1), (0.1, 0.7), (0.7, 0.1) and (0.7, 0.7), respectively.



Our second set of four graphs show the bias of `subst1`, `subst2` and `subst4` methods relative to `best` method for (sd28_153, cprop) equal to (0.1, 0.1), (0.1, 0.7), (0.7, 0.1) and (0.7, 0.7), respectively.

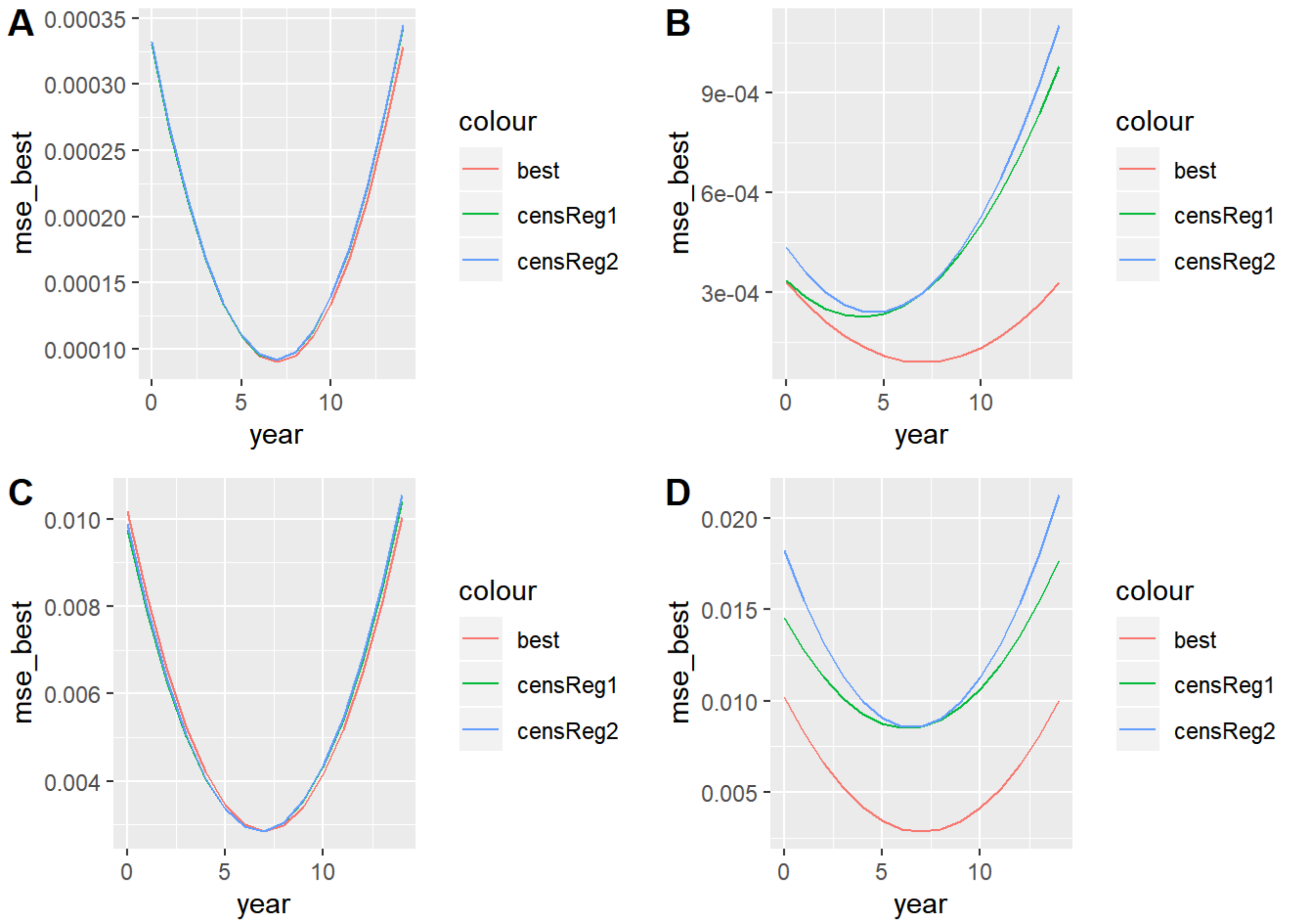


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

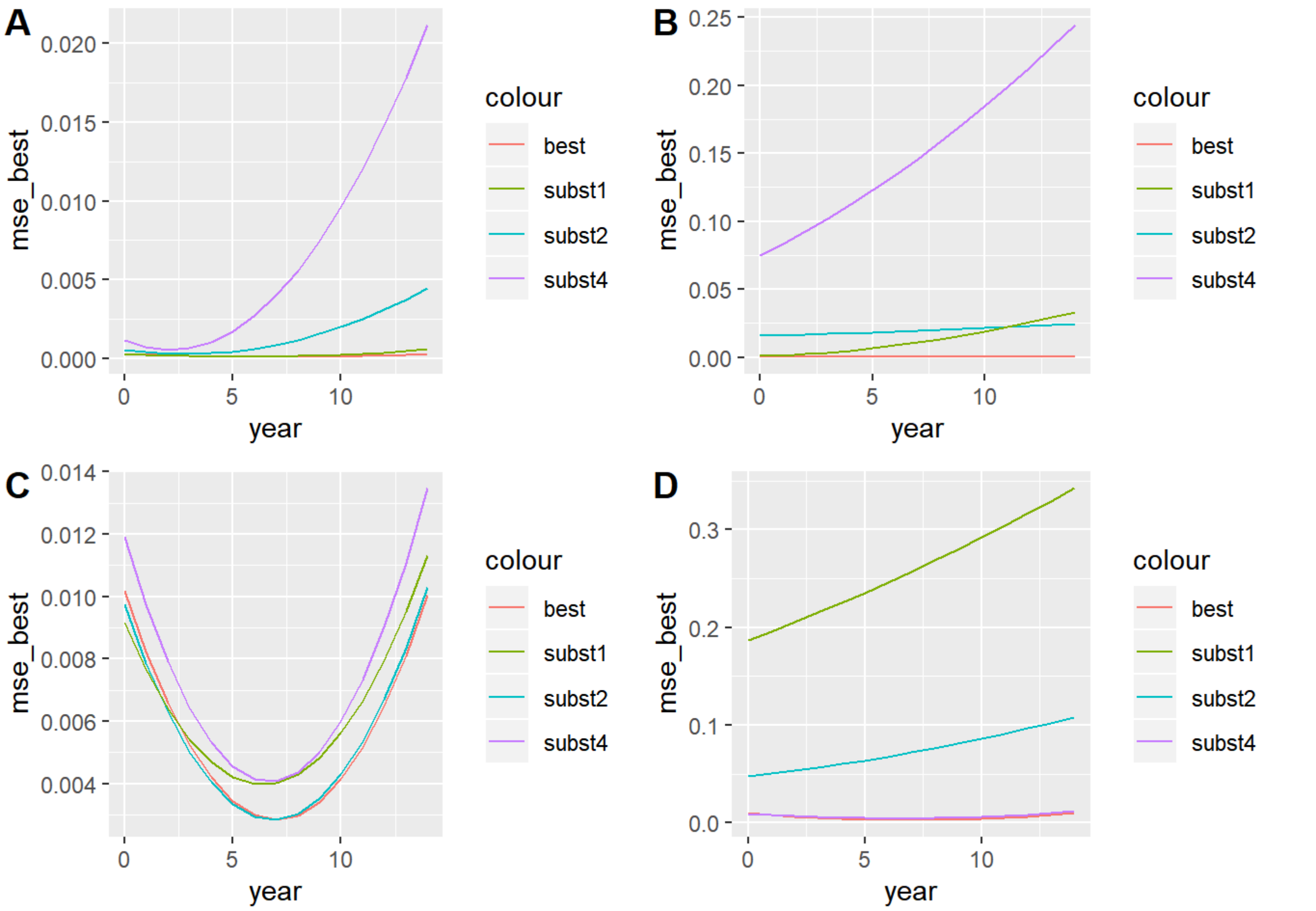


We will now show graphs of the MSE of predictions of cb28 annual means from our chosen censoring methods.

Our first set of four graphs show the MSE of `censReg1` and `censReg2` methods relative to `best` method for (sd28_153, cprop) equal to (0.1, 0.1), (0.1, 0.7), (0.7, 0.1) and (0.7, 0.7), respectively.



Our second set of four graphs show the MSE of `subst1`, `subst2` and `subst4` methods relative to `best` method for (sd28_153, cprop) equal to (0.1, 0.1), (0.1, 0.7), (0.7, 0.1) and (0.7, 0.7), respectively.



Our third set of four graphs simply displays the MSE from the `subst2` , `censReg1` and `best` methods together on the same plot, which is displayed below.