Simulation of imputation of censored values July v14

Marc Roddis

7/10/2020

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.

Results

Estimation of the regression coefficient `beta28year`

The MSE, squared-bias and variance for the estimation of `beta28year` from each method from our first simulation study are displayed in the table 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 values of the parameters for the simulation were based on estimates from the real dataset `pcb.csv.`: `true_beta153year = -0.02`, `sd28_153 = 0.1`, `cprop = 0.30`. A sample size of 100 was used because this is the total number of observations of cb153 from herring per year for the 15 year period 2003-2017. This represents the maximum sample size; subsets of the full dataset would correspond to smaller sizes.

We see that `censReg1`, `censReg2` and `censReg0impute` resulted in extremely low bias. Next best were `censReg1naive´ and` censReg2naive` estimates which had very similar squared-bias and variance to one another. In contrast, omission and all substitution methods resulted in very high bias relative to the aforementioned methods.

Evaluation of methods for smaller sample sizes

We will now focus on six methods `omit`, `subst1`, `subst2`, `censReg1`, `censReg2`, `censReg0impute`. We will use these methods to estimate `beta28year` using simulations that are each the same as the corresponding previous one, except that smaller values for `sample_size` will be used. Previous simulations used `sample_size=100` which was based on the fact that 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, so we will use 50, 25, 12, and 6 respectively. We will fix the proportion of censored data at our original value `cprop=0.30`, which is based on the real dataset.

As expected, the variance from every method is inversely proportional to the sample size. Moreover since $ MSE = Bias^2 + Variance$, we only need compare the values of squared-bias from each method. The columns of the following table show the squared-bias from our methods for sample sizes 50, 25, 12 and 6

respectively. We see that squared-bias has no clear association with sample size as far as `sample_size=12`. So, we will use 12 as our sample size in our subsequent work.

```
##                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
```

Evaluation of methods for larger absolute values of `beta28year`

We will now focus on six methods `omit`, `subst1`, `subst2`, `censReg1`, `censReg2`, `censReg0impute`. We will use these methods to estimate `beta28year` using simulations that are each the same as the corresponding previous one, except that we will compare our methods for three different `cb153year` parameter values: -0.02, -0.10, -0.50. We see that `cenreg1` and `cenreg2` perform best, and that `subst1` and `subst2` perform worse at larger absolute values of `beta28year`. Moreover, as expected, `censReg0impute` performs relatively worse compared to `censReg` methods since the imputations from the predictor variables carry more information about cb28 as the absolute value of `beta28year` increases. The fact that `subst1` and `subst2` perform much worse than `omit` for large absolute values of `beta28year` is to be expected since these methods use substitution with the same values irrespective of year, which results in increased bias.

```
##                mse_beta bias_beta variance_beta
## omit            6.10087   5.69345       0.40784
## subst2          9.42442   8.13811       1.28760
## subst1          2.40353   2.09472       0.30912
## censReg1        0.52123   0.00004       0.52171
## censReg2        0.55203   0.00002       0.55257
## censReg0impute  0.55840   0.00001       0.55895
## best            0.49985   0.00002       0.50033
## subst4         54.51066  51.16240       3.35161
## censReg1naive   0.66694   0.22863       0.43876
## subst2lmimpute 11.42905  10.68798       0.74181
## omitlmimpute    7.88702   7.32369       0.56389
```

```
##                 mse_beta bias_beta variance_beta
## omit            17.78808  17.20075       0.58792
## subst2          17.84205  17.01515       0.82773
## subst1           7.70523   7.32084       0.38477
## censReg1         0.58792   0.00069       0.58781
## censReg2         0.59509   0.00081       0.59488
## censReg0impute   0.62242   0.00128       0.62176
## best             0.47122   0.00001       0.47168
## subst4         121.76406 120.02496       1.74084
## censReg1naive    1.05709   0.49107       0.56659
## subst2lmimpute  80.88880  76.59546       4.29764
## omitlmimpute    32.12321  30.23173       1.89337
```

```
##                 mse_beta bias_beta variance_beta
## omit            28.57220  27.20639       1.36718
## subst2          14.32433  13.65106       0.67395
## subst1          23.32945  22.72465       0.60541
## censReg1         0.76418   0.00000       0.76494
## censReg2         0.76422   0.00002       0.76497
## censReg0impute   0.86846   0.00086       0.86847
## best             0.49749   0.00137       0.49662
## subst4         148.61779 147.78064       0.83799
## censReg1naive    1.04905   0.26156       0.78828
## subst2lmimpute 297.02326 286.27880      10.75522
## omitlmimpute   210.57431 204.67152       5.90870
```

```
##                   mse_beta  bias_beta variance_beta
## omit              19.65053   17.14620       2.50683
## subst2             0.89591    0.02318       0.87361
## subst1            79.66651   78.80083       0.86654
## censReg1           0.88616    0.00033       0.88672
## censReg2           0.89312    0.00044       0.89357
## censReg0impute     1.09064    0.00052       1.09121
## best               0.48739    0.00012       0.48775
## subst4            85.19200   84.29936       0.89353
## censReg1naive      0.98295    0.08077       0.90308
## subst2lmimpute   882.55339  882.15037       0.40343
## omitlmimpute    1117.76260 1101.95450      15.82392
```

Evaluation of methods for larger values of `sd28_153`

We will now hold `beta28year` fixed at our original value -0.02 and investigate the effect of larger `sd28_153` values, specifically: 0.1, 0.3, and 0.5. We see that larger values of this parameter result in much relatively larger variance for `censReg` methods, although these methods still have very low bias. The bias from `subst2` decreases as the `sd28_153` increased. Whereas the MSE from our methods differs greatly for `sd28_153=0.1`, there is very little difference at `sd28_153=0.5` for all methods except `omit`. This higher value means that the correlation between cb28 and cb153 is weaker, which results in less accurate imputation, since the accuracy of imputation relies on the strength of correlation between cb28 and cb153.

```
##                mse_beta bias_beta variance_beta
## omit            6.10087   5.69345       0.40784
## subst2          9.42442   8.13811       1.28760
## subst1          2.40353   2.09472       0.30912
## censReg1        0.52123   0.00004       0.52171
## censReg2        0.55203   0.00002       0.55257
## censReg0impute  0.55840   0.00001       0.55895
## best            0.49985   0.00002       0.50033
## subst4         54.51066  51.16240       3.35161
## censReg1naive   0.66694   0.22863       0.43876
## subst2lmimpute 11.42905  10.68798       0.74181
## omitlmimpute    7.88702   7.32369       0.56389
```

```
##                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           12.50736   6.40982       6.10364
## subst2          7.69824   0.10804       7.59779
## subst1          6.87072   2.27899       4.59632
## censReg1        7.98448   0.00042       7.99205
## censReg2        8.70062   0.00052       8.70881
## censReg0impute  8.69041   0.00058       8.69853
## best            7.76117   0.01682       7.75210
## subst4         12.80996   0.72630      12.09576
## censReg1naive   5.76405   0.79955       4.96947
## subst2lmimpute  9.42664   5.90885       3.52131
## omitlmimpute   11.04559   6.58590       4.46416
```

Further comparisons using `sd28_153=0.5`

The five methods `subst1`, `subst2`, `censReg1`, `censReg2`, `censReg0impute` estimated `beta28year` with similar MSE values for parameter values: `beta153year= -0.02`, `sample_size = 12`. So we will fix these parameters at these values and return to investigating the effect of `LOD`, which is our primary variable of interest. The following results were obtained for the three `LOD` levels given by `cprop=0.1`, `cprop=0.5`, and `cprop=0.7` (see the previous section for the corresponding results for `cprop=0.3`). Three sets of three (nine in all) results tables are shown for these three levels of `cprop`, each of which is paired with the three values `sd28_153=0.1`, `sd28_153=0.3`, and `sd28_153=0.5`.

The following three tables show the MSE, squared-bias, and variance of estimates of `beta28year` from simulations with parameter values fixed at `beta153year = -0.02` and `sd28_153 = 0.1`, in which various methods were used to process censored values. These three tables show results from using the three values 0.1, 0.5, and 0.7 for the `cprop` parameter, respectively.

```
##                mse_beta bias_beta variance_beta
## omit            1.94573   1.50568       0.44049
## subst2          3.28387   2.47542       0.80926
## subst1          0.62231   0.19495       0.42779
## censReg1        0.48293   0.00001       0.48340
## censReg2        0.48707   0.00000       0.48756
## censReg0impute  0.48612   0.00000       0.48661
## best            0.49985   0.00002       0.50033
## subst4         14.44620  12.87532       1.57245
## censReg1naive   0.74552   0.31287       0.43308
## subst2lmimpute  4.33385   3.80822       0.52616
## omitlmimpute    2.22148   1.66736       0.55467
```

```
##                mse_beta bias_beta variance_beta
## omit            6.10087   5.69345       0.40784
## subst2          9.42442   8.13811       1.28760
## subst1          2.40353   2.09472       0.30912
## censReg1        0.52123   0.00004       0.52171
## censReg2        0.55203   0.00002       0.55257
## censReg0impute  0.55840   0.00001       0.55895
## best            0.49985   0.00002       0.50033
## subst4         54.51066  51.16240       3.35161
## censReg1naive   0.66694   0.22863       0.43876
## subst2lmimpute 11.42905  10.68798       0.74181
## omitlmimpute    7.88702   7.32369       0.56389
```

```
##                mse_beta bias_beta variance_beta
## omit           10.13803   9.64742       0.49110
## subst2          7.42768   6.09041       1.33860
## subst1          6.54621   6.34115       0.20527
## censReg1        0.60419   0.00127       0.60352
## censReg2        0.70704   0.00120       0.70654
## censReg0impute  0.74332   0.00173       0.74233
## best            0.49985   0.00002       0.50033
## subst4         59.42892  55.56087       3.87193
## censReg1naive   0.54294   0.00103       0.54245
## subst2lmimpute 11.57360  11.18111       0.39288
## omitlmimpute   14.21151  13.82835       0.38355
```

```
##                mse_beta bias_beta variance_beta
## omit           14.13397  13.45675       0.67790
## subst2          1.49817   0.62596       0.87308
## subst1         12.62066  12.52761       0.09314
## censReg1        0.74680   0.00039       0.74716
## censReg2        0.96849   0.00129       0.96816
## censReg0impute  1.02894   0.00175       1.02822
## best            0.49985   0.00002       0.50033
## subst4         28.93460  26.23278       2.70452
## censReg1naive   1.03778   0.25186       0.78671
## subst2lmimpute 10.28893  10.15964       0.12942
## omitlmimpute   18.82095  18.70101       0.12007
```

The following three tables show the MSE, squared-bias, and variance of estimates of `beta28year` from simulations with parameter values fixed at `beta153year = -0.02` and `sd28_153 = 0.3`, in which various methods were used to process censored values. These three tables show results from using the three values 0.1, 0.5, and 0.7 for the `cprop` parameter, respectively.

```
##                mse_beta bias_beta variance_beta
## omit            4.22217   1.85563       2.36891
## subst2          3.71252   0.22153       3.49449
## subst1          2.67055   0.22364       2.44936
## censReg1        2.87773   0.00012       2.88049
## censReg2        2.92403   0.00011       2.92685
## censReg0impute  2.92885   0.00013       2.93165
## best            2.95767   0.00138       2.95925
## subst4          7.10011   2.00007       5.10515
## censReg1naive   3.13037   0.97510       2.15743
## subst2lmimpute  4.31052   2.29059       2.02195
## omitlmimpute    3.86178   1.58012       2.28394
```

```
##                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           12.23109   9.83474       2.39875
## subst2          3.32426   0.08792       3.23958
## subst1          7.37509   6.27917       1.09702
## censReg1        3.11989   0.00035       3.12266
## censReg2        3.76850   0.00038       3.77189
## censReg0impute  3.80356   0.00028       3.80709
## best            2.95767   0.00138       2.95925
## subst4         10.68667   3.65877       7.03493
## censReg1naive   2.49539   0.00087       2.49702
## subst2lmimpute  9.13916   8.24748       0.89257
## omitlmimpute   13.78530  12.51601       1.27057
```

```
##                mse_beta bias_beta variance_beta
## omit           17.24361  14.02438       3.22245
## subst2          4.53033   2.64511       1.88710
## subst1         12.98170  12.51734       0.46483
## censReg1        3.07060   0.00034       3.07334
## censReg2        4.40549   0.00094       4.40896
## censReg0impute  4.42517   0.00198       4.42761
## best            2.95767   0.00138       2.95925
## subst4          4.72768   0.08135       4.65098
## censReg1naive   4.05629   0.65479       3.40490
## subst2lmimpute 11.65907  11.07330       0.58636
## omitlmimpute   19.08027  18.25694       0.82415
```

The following three tables show the MSE, squared-bias, and variance of estimates of `beta28year` from simulations with parameter values fixed at `beta153year = -0.02` and `sd28_153 = 0.5`, in which various methods were used to process censored values. These three tables show results from using the three values 0.1, 0.5, and 0.7 for the `cprop` parameter, respectively.

```
##                mse_beta bias_beta variance_beta
## omit            8.16691   2.32836       5.84439
## subst2          6.97480   0.01114       6.97063
## subst1          6.08940   0.44654       5.64852
## censReg1        6.64952   0.03803       6.61811
## censReg2        6.76917   0.03933       6.73658
## censReg0impute  6.76973   0.03983       6.73664
## best            7.76117   0.01682       7.75210
## subst4          9.06100   0.20899       8.86088
## censReg1naive   6.55698   1.37954       5.18262
## subst2lmimpute  7.42060   2.45574       4.96983
## omitlmimpute    6.95762   1.84795       5.11478
```

```
##                mse_beta bias_beta variance_beta
## omit           12.50736   6.40982       6.10364
## subst2          7.69824   0.10804       7.59779
## subst1          6.87072   2.27899       4.59632
## censReg1        7.98448   0.00042       7.99205
## censReg2        8.70062   0.00052       8.70881
## censReg0impute  8.69041   0.00058       8.69853
## best            7.76117   0.01682       7.75210
## subst4         12.80996   0.72630      12.09576
## censReg1naive   5.76405   0.79955       4.96947
## subst2lmimpute  9.42664   5.90885       3.52131
## omitlmimpute   11.04559   6.58590       4.46416
```

```
##                mse_beta bias_beta variance_beta
## omit           17.09506  10.28862       6.81325
## subst2          6.88958   1.10421       5.79116
## subst1          8.84301   6.02310       2.82274
## censReg1        8.02075   0.01354       8.01523
## censReg2        9.87517   0.01560       9.86944
## censReg0impute  9.87430   0.01390       9.87027
## best            7.76117   0.01682       7.75210
## subst4         10.63445   0.12431      10.52066
## censReg1naive   6.15120   0.00861       6.14873
## subst2lmimpute 10.32535   8.06495       2.26266
## omitlmimpute   14.53694  10.98382       3.55668
```

```
##                mse_beta bias_beta variance_beta
## omit           21.18031  12.48679       8.70222
## subst2          8.37304   5.15594       3.22032
## subst1         13.19163  11.98684       1.20600
## censReg1        7.66589   0.00451       7.66905
## censReg2       11.34171   0.00386      11.34921
## censReg0impute 11.34744   0.00429      11.35451
## best            7.76117   0.01682       7.75210
## subst4          7.81905   1.16453       6.66118
## censReg1naive   9.35157   0.90675       8.45328
## subst2lmimpute 12.84661  11.45671       1.39129
## omitlmimpute   19.55833  17.22978       2.33089
```

Our next task will be to interpret all of our results so far.

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

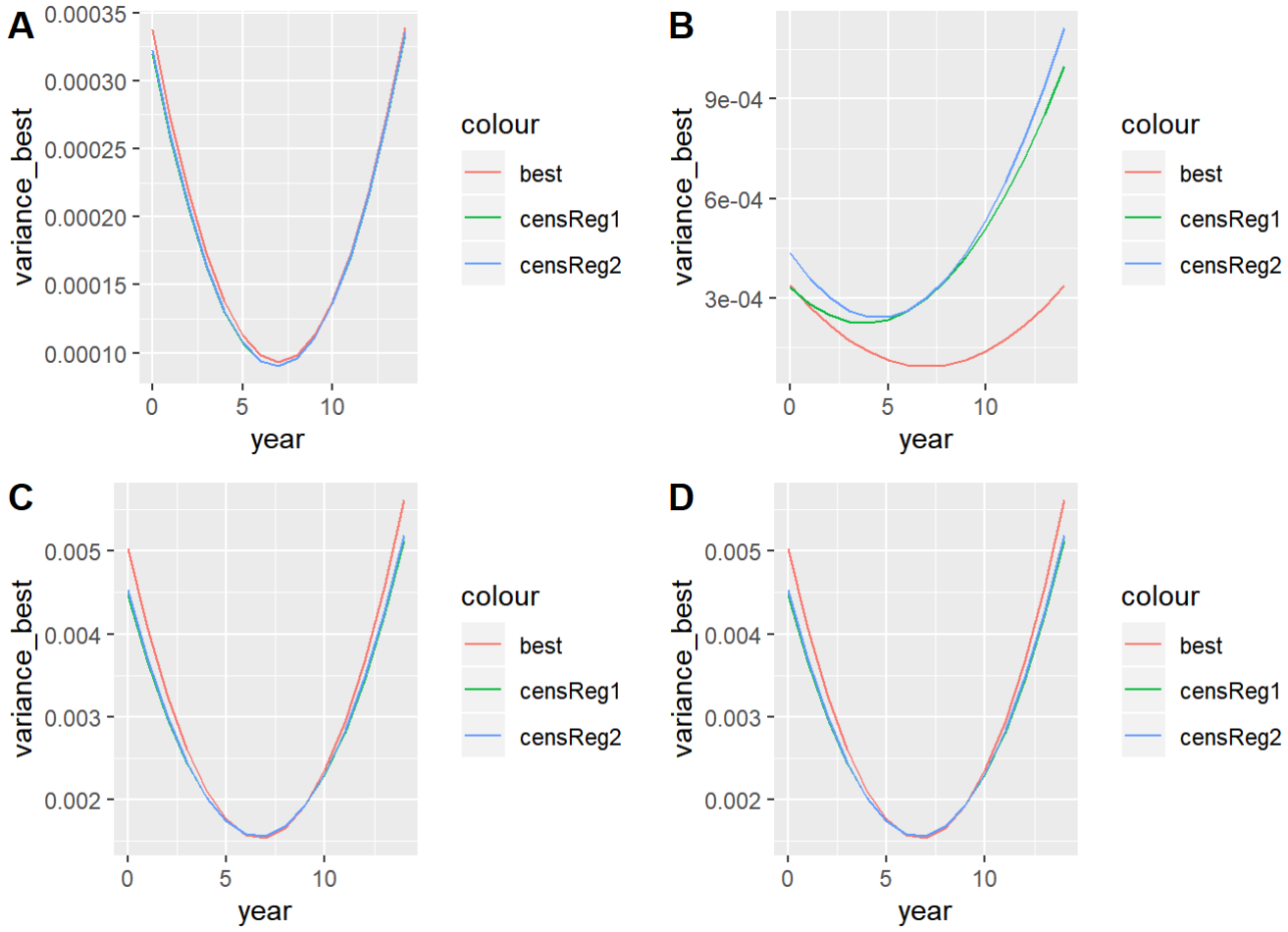
All the graphs in this section will shows MSE, squared-bias, or variance on the y-axis and year on the x-axis for the simulated 15-year period. We begin by looking at squared-bias.

Variance of predictions of cb28 annual means from different methods

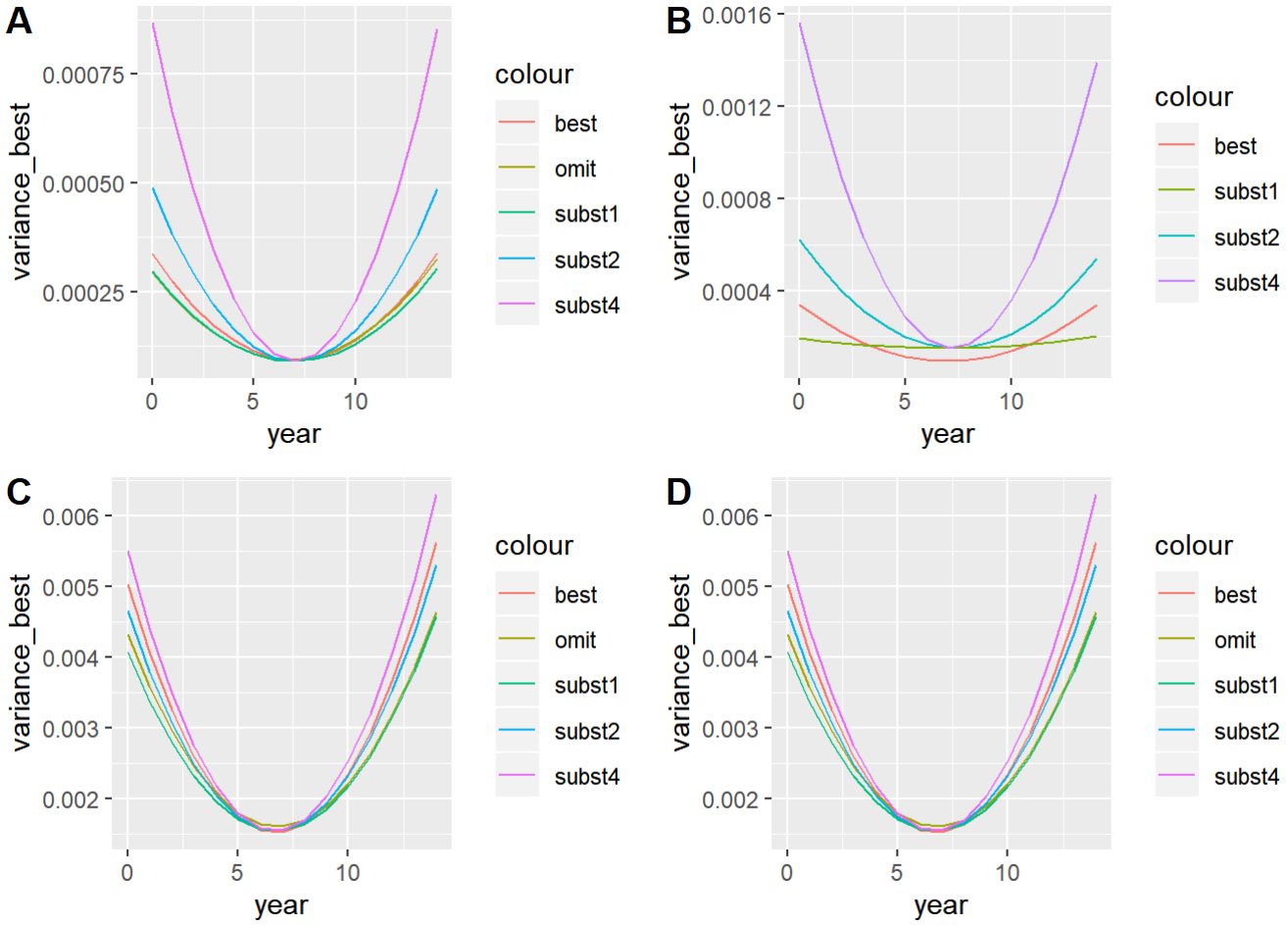
Our censoring methods give predictions with similar amounts of variance

Every graph in this section shows the variance of predictions of cb28 annual means from some of our 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.

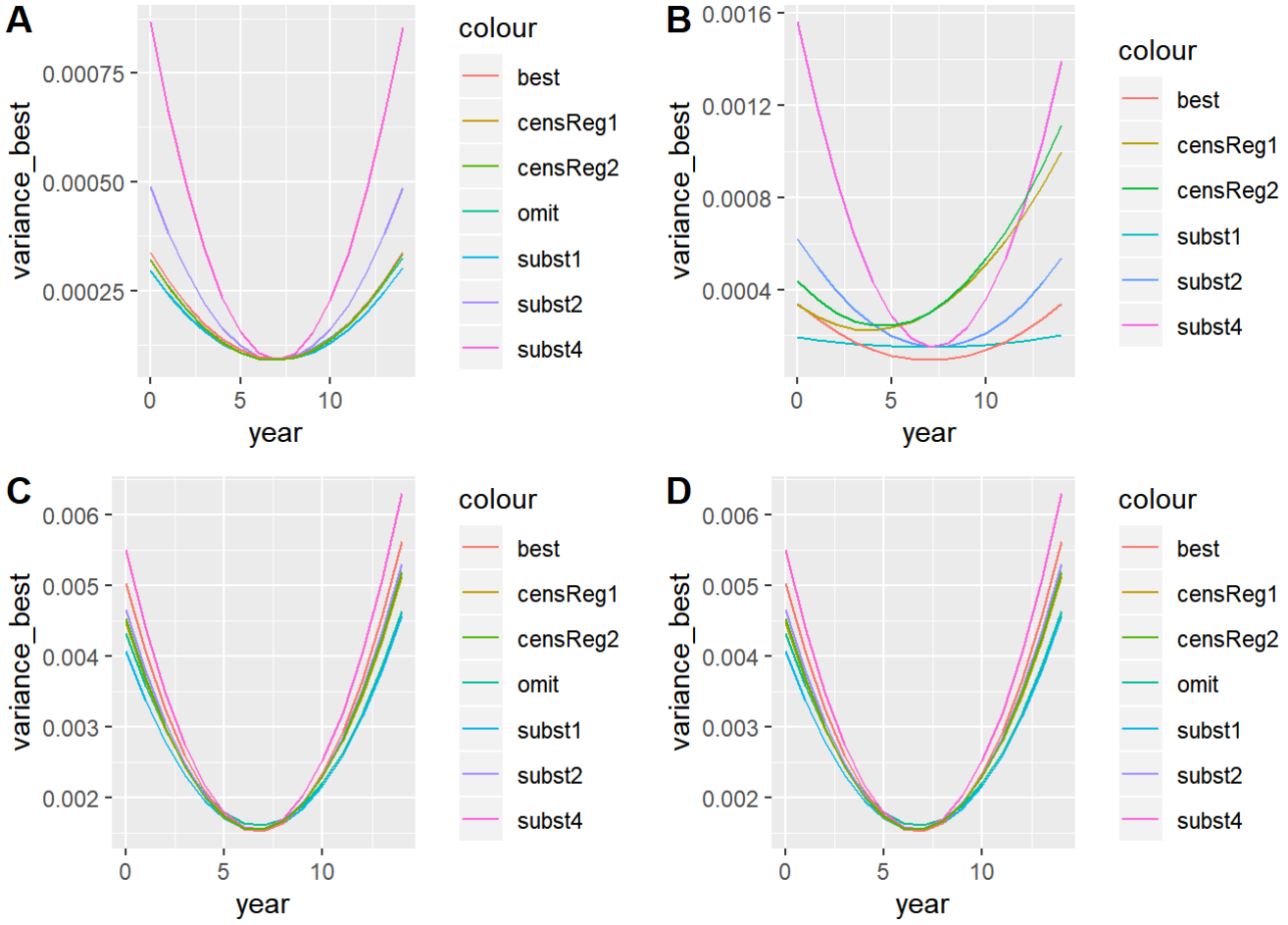
Our first set of four graphs show the variance of `censReg1` and `censReg2` methods relative to `best` method for $(cprop, sd)$ equal to $(0.1, 0.1)$, $(0.7, 0.1)$, $(0.1, 0.5)$, $(0.7, 0.5)$, respectively. We see that the `censReg2` method consistently gives higher variance than `censReg1`. This is in accordance with our prior expectations because models with more predictor variables generally have higher variance than models with fewer predictors.



Our second set of four graphs show the variance of `subst1`, `subst2` and `subst4` methods relative to `best` method for $(cprop, sd)$ equal to $(0.1, 0.1)$, $(0.7, 0.1)$, $(0.1, 0.5)$, $(0.7, 0.5)$, respectively. In addition, the `omit` method can be used to obtain results for `cprop = 0.1` but not for `cprop = 0.7`, so the variance from those methods is shown on the two graphs for which `cprop = 0.1`.



Our third set of four graphs simply displays the previous two sets together on the same plot, which is displayed below.

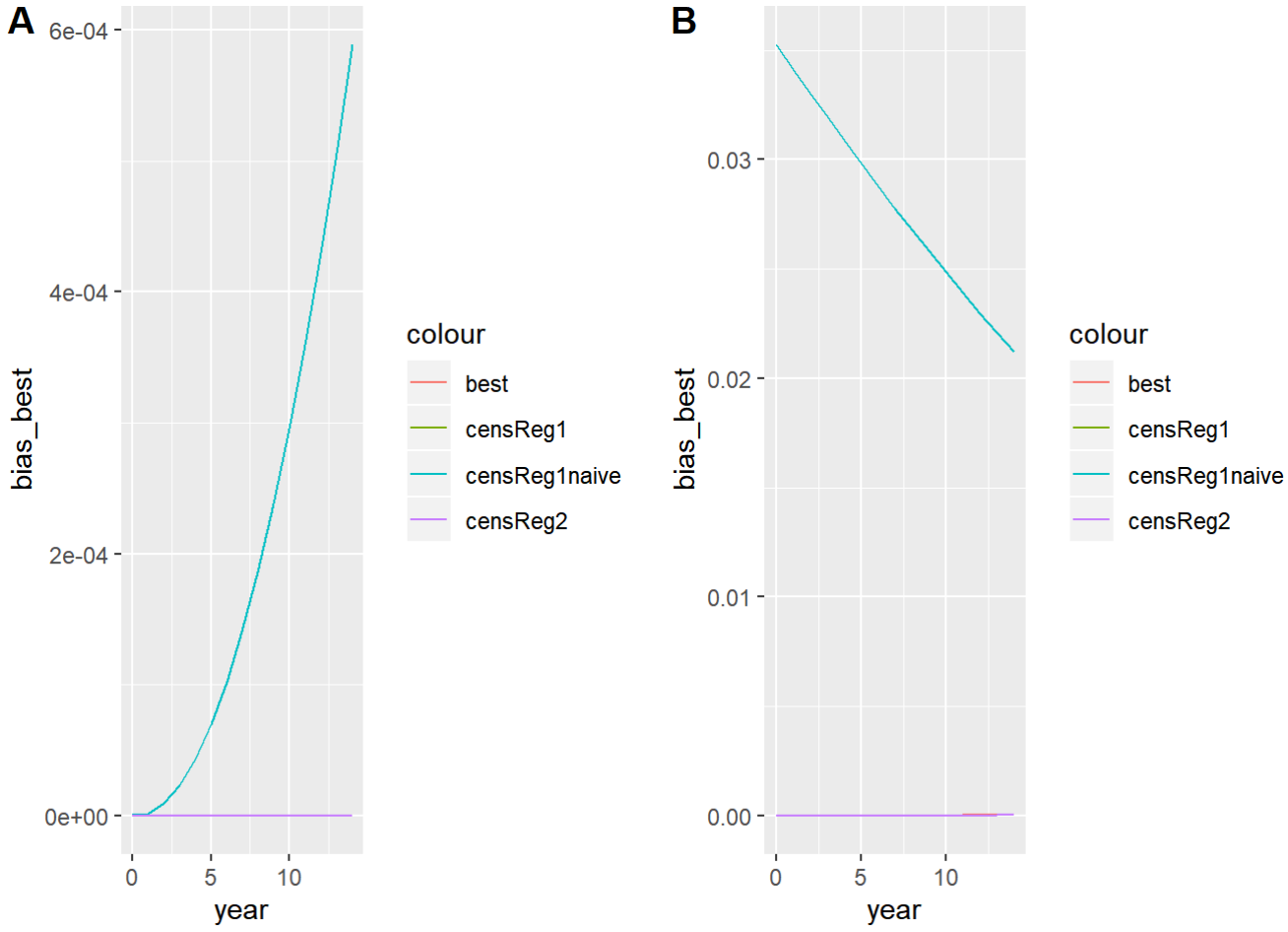


Bias of predictions of cb28 annual means from all of our censoring methods

The `censReg1naive` method gives extremely biased predictions

The `censReg1naive` method gives extremely biased predictions for all 12 parameter value-pairs (four cprop levels, and three sd28_153 levels) .
Two graphs are presented below to illustrate this: they show the bias of predictions of cb28 annual means from the lowest levels (cprop = 0.1, sd28_153 = 0.1), and highest levels (cprop = 0.7, sd28_153 = 0.5), for each of these parameters, respectively.

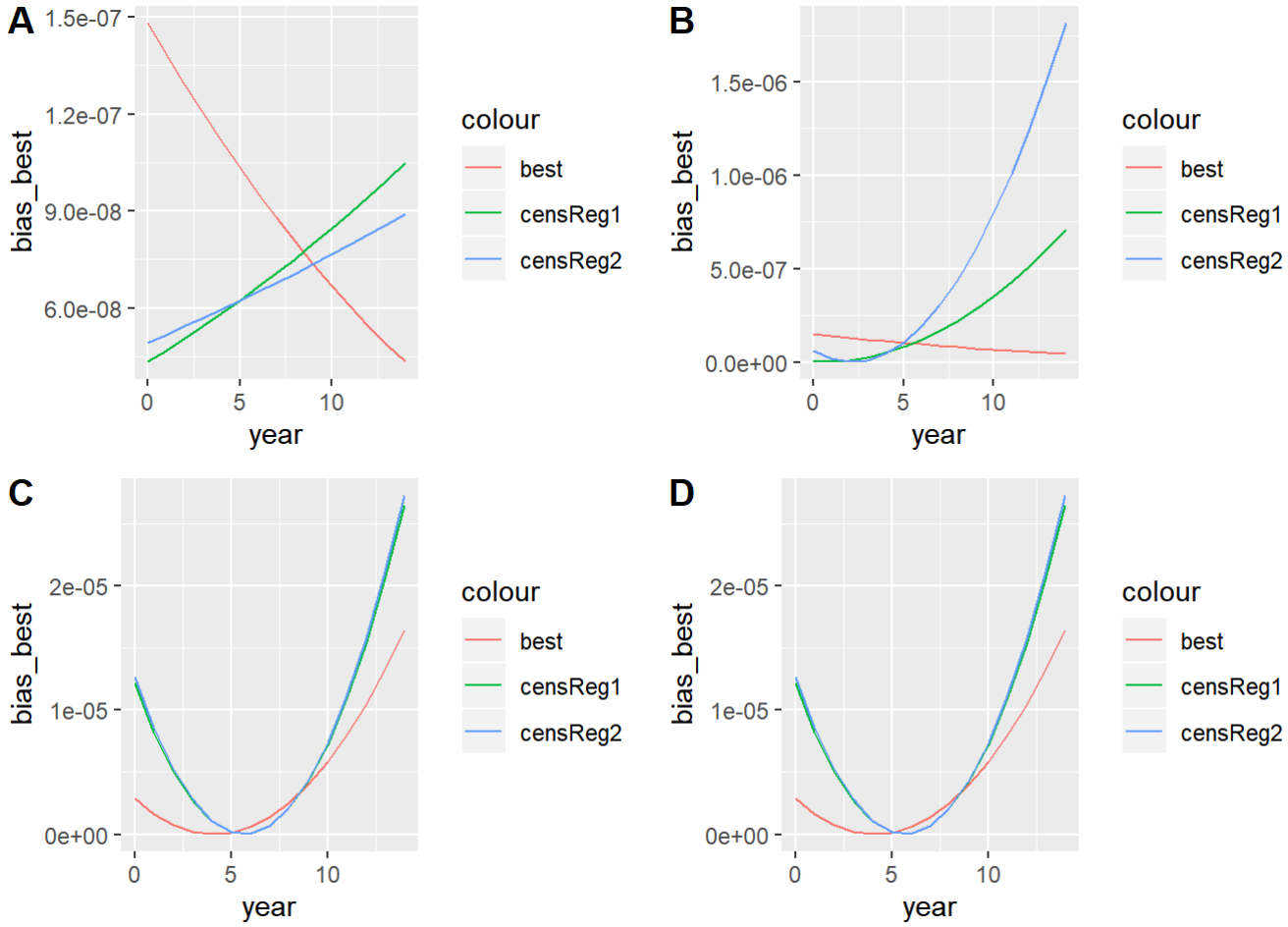
This shows the necessity of conditioning on both the cb153 value and the condition cb28 < cb28_cprop by using a truncated normal distribution as presented in our previous chapter on mathematical theory. In contrast, censReg1naive conditions on the cb153 value only and uses a non-truncated normal distribution which results in significant bias because imputed cb28 values can be higher than cb28_cprop. Consequently we will not discuss censReg1naive any further: we expected this method to give biased estimates and it did.



Bias of predictions of cb28 annual means from all of our censoring methods except for `censReg1naive`

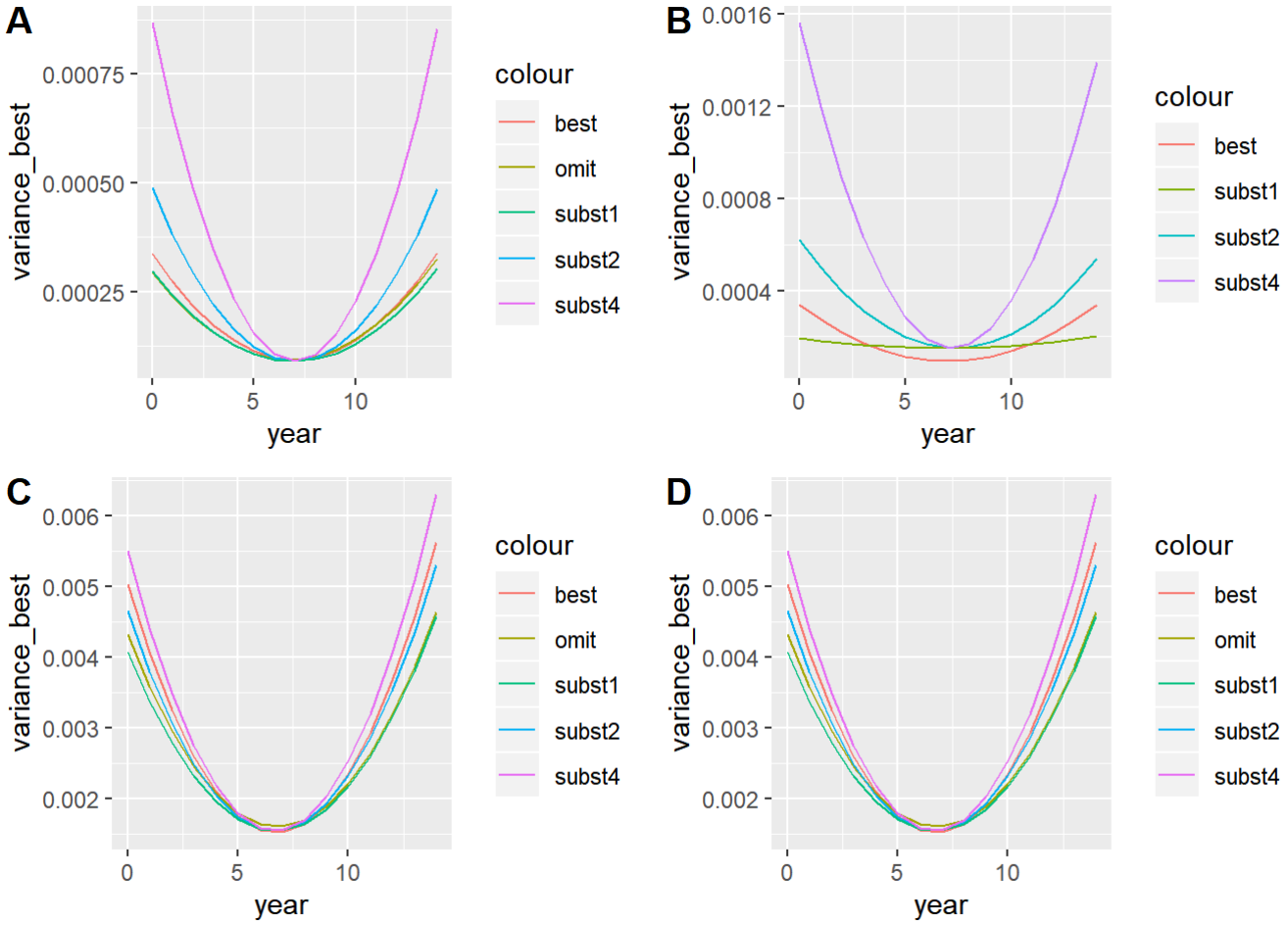
Every graph in this section shows the squared-bias of predictions of cb28 annual means from some of our censoring methods.

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

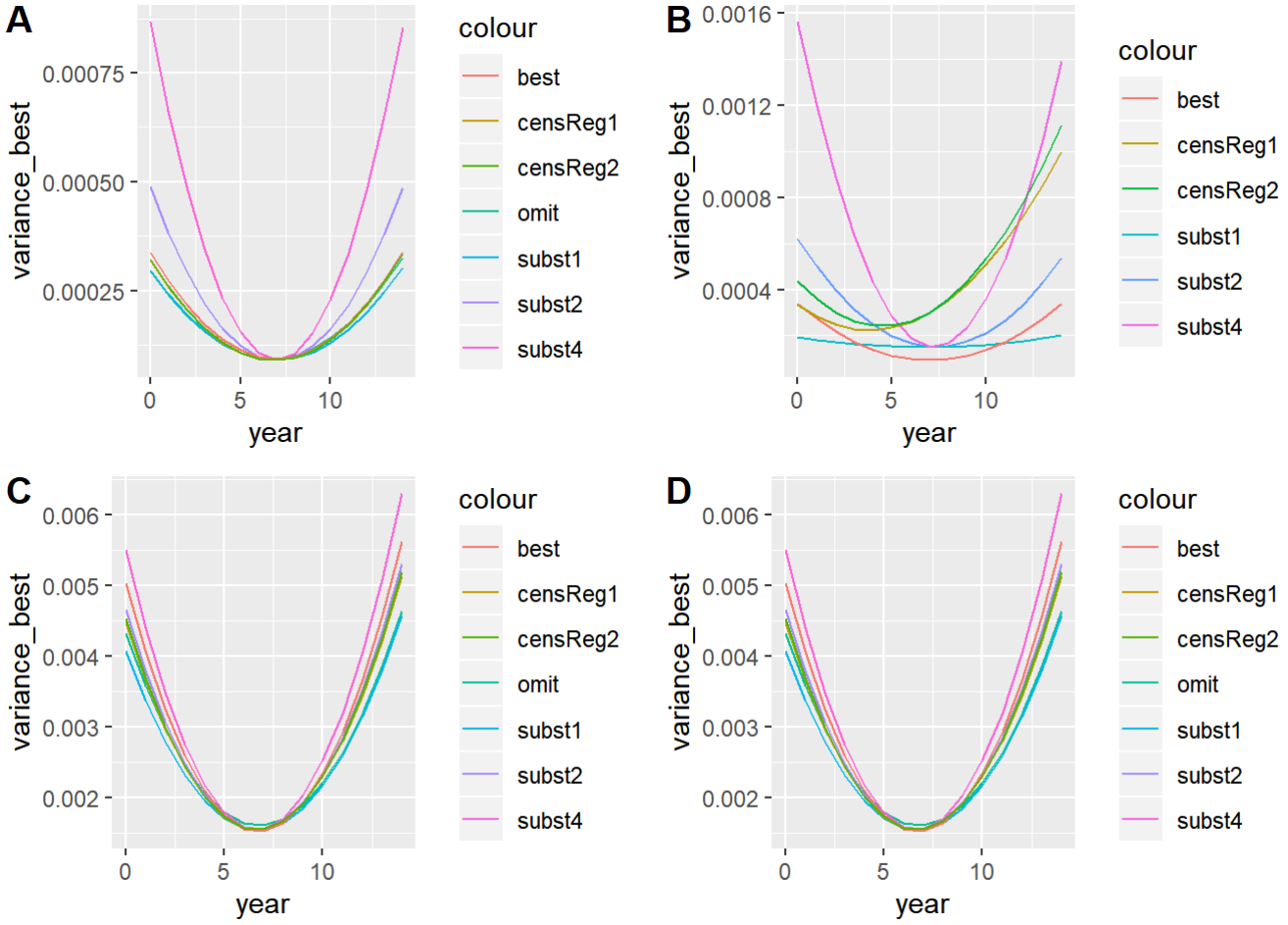


Our second set of four graphs show the bias of `subst1`, `subst2` and `subst4` methods relative to `best` method for $(cprop, sd)$ equal to $(0.1, 0.1)$, $(0.7, 0.1)$, $(0.1, 0.5)$, $(0.7, 0.5)$, respectively. In addition, the `omit` method can be used to obtain results for `cprop = 0.1` but not for `cprop = 0.7`, so the variance from those methods is shown on the two graphs for which `cprop = 0.1`.

We see that (ref: my Google Doc)



Our third set of four graphs simply displays the previous two sets together on the same plot, which is displayed below.



The remainder of this report is only VERY PRELIMINARY. BIG CHANGE IS GONNA COME.

Accuracy of yearly predictions for cprop = 0.1 and sd28_153 = 0.1.

Accuracy of yearly predictions for cprop = 0.3 and sd28_153 = 0.1.

Accuracy of yearly predictions for cprop = 0.5 and sd28_153 = 0.1.

Accuracy of yearly predictions for cprop = 0.7 and sd28_153 = 0.1.

Accuracy of yearly predictions for cprop = 0.1 and sd28_153 = 0.3.

Accuracy of yearly predictions for cprop = 0.3 and sd28_153 = 0.3.

Accuracy of yearly predictions for cprop = 0.5 and sd28_153 = 0.3.

Accuracy of yearly predictions for cprop = 0.7 and sd28_153 = 0.3.

Accuracy of yearly predictions for cprop = 0.1 and sd28_153 = 0.5.

Accuracy of yearly predictions for cprop = 0.3 and sd28_153 = 0.5.

Accuracy of yearly predictions for cprop = 0.5 and sd28_153 = 0.5.

Accuracy of yearly predictions for cprop = 0.7 and sd28_153 = 0.5.

Miscellaneous brain-storming-type notes

Cenreg did not work reliably (see v1 of this doc), so all censored regression will be done with censReg (followed by etruncnorm).

Sqrt(2) seems to be the best denominator. I could also try other numbers denominators and compare.

Found mse, squared-bias and variance with respect to estimation of beta for:

best, omit, subst2 (substitution with $\frac{LOD}{\sqrt{(2)}}$), subst1, subst4, censReg1, censReg2,

Simulation study reference: `Tekindal2017_EvaluatingLeft-CensoredDataBySimulationStudy.pdf`).

Could find the boundaries of the parameter space, especially:

`cprop` # censoring proportion

`true_beta28year` #beta for cb28 ~ year

Could try censReg with or without `year` .

Could try different substitutions:

LOD, LOD/sqrt(2), LOD/2, 0.

Appendices

Appendix 1

The two graphs A, B below show the variation of MSE (red curve) and squared-bias-plus-variance (black curve) from `best_fit` and `omit_fit` respectively, over the simulated 15-year period. The famous result “Bias-variance decomposition” states

$$MSE = Bias^2 + Variance$$

so we expect the black and red curves to coincide (be superposed); happily they are :)

Appendix 2

We now generate the dataset `omit_yearly_mean` from the fixed parameters as follows:

12 values for the log-concentration of CB153 per year, for ten years, were generated and denoted as `cb153sim`.

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

`median(cb28)` was used as the level of quantification `LOQ_p50`.

Observations with CB28 < LOQ_p50 were removed from the dataset.

Annual geometric means for CB28 and CB153 concentrations were generated.

The code chunk below generates `omit_yearly_mean` for 1000 iterations, fits a linear model `omit_fit` at each iteration, and computes the corresponding mse, squared-bias and variance.

Appendix 3