Methods for processing censored data FV3

Marc Roddis

7/17/2020

Introduction

The Swedish National Monitoring Programme for Contaminants (SNMPC) in freshwater biota has various goals and large scope (citation needed).

Our main goal in this study was to explore the viability of alternative methodologies for parameter estimation from censored data and to compare these alternatives with the methodology used by SNMPC.

Censored data is very common in environmental chemistry so our research area has been researched extensively by others. We will select well-regarded methods and apply these according to best practice, according to the cited works (citation needed).

At the outset, we limited the scope of our study by choosing to focus on the estimation of long-term time trends for the concentration of polychlorinated biphenyls (PCBs) in biological samples.

Our main idea was that since PCBs have similar chemical and physical properties their concentrations may be correlated such that censored measurements can be imputed using censored regression.

Our idea is supported by the SNMPC dataset since it has no censored data for CB153, whereas 34 % of the data for CB28 is censored.

More importantly, our exploratory data analysis of SNMPC data showed that CB153 and CB28 concentrations are strongly correlated.

Moreover, CB153 and CB28 also show a very similar rate of decrease over the time period 2003-2017.

Concretely, our idea is to impute censored CB28 values from the corresponding uncensored observations for CB153.

The resulting "imputed datasets" could then be used to obtain better parameter estimates than from the methodology currently used by SNMPC, which uses "substituted datasets" instead.

Specifically, SNMPC substitute all censored data by $\frac{LOD}{\sqrt{2}}$.

Our workflow

We now present an eight-step overview of our workflow in this section. We will give a more detailed description of steps 1-7 in the subsequent sections.

1. Selection of a set of parameter values, and generation of the simulation dataset accordingly.
2. Estimation of β from the simulation dataset by simple linear regression to get a benchmark to compare other methods against.
3. Selection of the proportion `cprop` of CB28 values to censor, and generation of the censored dataset accordingly.
4. Creation of a "completed dataset" by replacing censored data using some or all of our six methods. The completed datasets created by different methods will be distinct.
5. Estimation of β from each completed dataset by simple linear regression (by the same procedure as Step 2).
6. Estimation of β directly from the censored dataset by censored regression.
7. Presentation of the MSE, squared-bias, and variance, of the estimates from each method.
8. Repetition of steps 1-7 for various selections of parameter values.
9. Discussion of all results.

Model selection for generation of our simulation datasets

We begin by performing exploratory data analysis and model fitting from datasets from the SNMPC.

We do this in order to design our simulation studies to have real real-world relevance.

A large dataset `pcb.csv` was provided from SNMPC.

This dataset has 5056 observations of 18 variables; these variables include: measured concentrations of seven PCBs (CB28, CB53, CB101, CB118, CB138, CB153, CB180); year (1984-2017); an ID for each observation; and nine other variables such as species and age.

Our exploratory data analysis showed that

1. The most recent 15-year period 2003-2017 had sufficient relevant data, so we will focus solely on this time period.
2. It is reasonable to model the observed pcb concentrations as log-normal distributed.
3. The data for CB153 had no censored values, whereas CB28 data had the highest proportion of censored values. This proportion was 0.34.
4. Species is clearly a confounding variable for the association between CB153 and CB28, so we will focus solely on herring (since this was the species for which there were most observations). No other variable showed clear evidence for confounding.

From this basis, we create our test dataset from the original dataset `pcb.csv` by omitting all missing values of `CB28` and `CB153`, removing all observations except those from herring species, removing all observations prior to 2003, re-indexing 2003 as "year zero", removing all variables except `YEAR`, `CB28` and `CB153`, and omitting all censored observations.

We fit linear regression models to our test dataset for $y = CB28, x = CB153$ and for $y = log(CB28), x = log(CB153)$; the adjusted R-squared values were 0.93 and 0.96 respectively.

Based on this, we decide we will use logarithmised concentrations throughout, which we will model as normally distributed.

We have three variables $log(CB28)$, $log(CB153)$, and $YEAR$; we will denote these as Y, X and A respectively, throughout the rest of our work.

We make the key observation that Y and X are strongly correlated in our test dataset, which means that our key idea of using censored regression for Y on X to make imputations for censored Y values is plausible.

We also fit a model to our test dataset for the regression X on A, which gave

$$E(X|A) = -2.91 - 0.02A$$

the corresponding fitted model for Y on X is

$$E(Y|X) = -3.18 + 0.79X$$

the residual standard error was equal to 0.1 from both models.

From this basis, we will generate our simulation datasets as follows:

1. We will also always simulate a 15-year period; we will use $A \in \{0, 1, 2, ..., 14\}$ to denote year.
2. For every year, we will generate the same number of observations for Y and X, we will call this number the sample size N. For our first simulation we will use `sample size` $= 100$ because there are typically 100 observations on herring each year by the Monitoring Program.
3. We generate all x_i from

$$x_i = -2.91 - \beta_A a_i + e_i$$

where $i \in \{1, 2, ..., N\}$ denotes the ith observation, and the noise is modeled as normally distributed with $mean = 0$ and $variance = 0.1^2$, i.e. $e_i \sim N(0, 0.1^2)$.

We will be interested in evaluating our methods for various values of β_A, so this will be a variable parameter for our simulations.

4. We generate all y_i from

$$y_i = -3.18 + 0.79x_i + \epsilon_i$$

where $\epsilon_i \sim N(0, \sigma^2)$.

We will be interested in evaluating our methods for various values of β_A and σ^2, so these will be the two variable parameters for our simulation datasets.

Estimation of β from the simulation dataset by simple linear regression

The main body of our work will be to evaluate various methods for the estimation of the regression coefficient $\widehat{\beta}$ for datasets containing censored values.

In this section, we will instead assume that there are no censored values, which allows us to find estimates by simple linear regression.

We will later use these estimates as the benchmark for evaluating the methods we use in the main body of our work.

Our primary goal is to find $\widehat{\beta}$, which means we will find estimates for β where $Y_i = \alpha + \beta a_i + \varepsilon_i$.

To specify this model, we first substitute

$$x_i = -2.91 - \beta_A a_i + e_i$$

nto

$$y_i^* = -3.18 + 0.79 x_i + \epsilon_i$$

which gives

$$y_i^* = -3.18 + 0.79(-2.91 - \beta_A a_i + e_i) + \epsilon_i$$

$$= -3.18 + 0.79 - 2.91 - 0.79\beta_A a_i + 0.79 e_i + \epsilon_i$$

$$= \alpha + \beta a_i + \varepsilon_i$$

where $\alpha = -3.18 + 0.79 \times -2.91 = -5.4789$, and $\beta = 0.79\beta_A$.

Also $\varepsilon_i = 0.79 e_i + \epsilon_i$, where $e_i \sim N(0, 0.1^2)$ and $\epsilon_i \sim N(0, \sigma^2)$.

Creation of datasets with censored values

For our simple linear regression of Y on X we used $y_i = -3.18 + 0.79 x_i + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma^2)$.

We now use y_i^* instead of y_i, where y_i^* refers to the ith observation prior to it being observed.

This means that after y_i has been observed and left-censoring at LOD has been applied, we have $y_i = y_i^*$ if $y_i^* > LOD$ and $y_i = LOD$ if $y_i^* \leq LOD$.

We will determine LOD by censoring a fixed proportion, which we denote as *cprop*, of all observed y_i values, for each of our simulations.

We will be interested in evaluating our methods for various values of *cprop*, which is our variable parameter of primary interest.

Moreover since $LOD = cprop * 100$th percentile of all y_i values, the value of $LOD|cprop$ is constant and thus independent of A.

Our primary goal is to find $\widehat{\beta}$, which means we will find estimates for β where $Y_i = \alpha + \beta a_i + \varepsilon_i$.

Creation of a "completed dataset" by replacing censored data using some or all of our six methods

We view every censored observation as having a true but unknown value within the interval $[0, LOD]$.

Our goal is to replace all such unknown values with a known value such that the resulting values are as close to the true values as possible.

The most straightforward way to this is by substitution, which means that all censored values from a censored dataset are substituted by the same fixed value, which is a fraction of LOD.

The monitoring program that motivates our work uses substitution by $\frac{LOD}{\sqrt{2}}$, which is the most commonly used value based in the research literature cited in this report.

The second most commonly used value is $\frac{LOD}{2}$.

The largest possible value that can be used for substitution is LOD, since all of the censored values are known to lie within the interval $[0, LOD]$.

Our three substitution methods will use substitution by either LOD, $\frac{LOD}{\sqrt{2}}$ or $\frac{LOD}{2}$; we name them `subst1`, `subst2`, and `subst4`, respectively.
Our notation is based on the fact that $LOD = \frac{LOD}{1}$, and that $2 = \sqrt{4}$ and $1 = \sqrt{1}$, respectively.

Our rationale for choosing these three methods is that since $\frac{LOD}{2} < \frac{LOD}{\sqrt{2}} < LOD$ we can compare results from the `subst2` method that SNMPC uses with two alternative substitution methods, which use substitution by lower and higher values, respectively.

However, such substitution methods are limited since they do not use observations from other variables of the dataset.

Our conjecture is that we can use censored regression to impute censored y_i values from the corresponding uncensored x_i values, thus leveraging the strong correlation between the Y and X.

We call our main imputation by censored regression method `censReg1` , because it is based on a censored regression model with 1 predictor variable X, as described in the following section.

Creation of completed datasets by censored regression by our main method `censReg1`

Each observation y_i^* is from a normal distribution with mean $\mu_{i_X} = \alpha_X + \beta_X x_i$ and variance σ^2, which has pdf

$$f(y_i^*) = \frac{exp[(-1/2)((y_i^* - \mu_{i_X})/\sigma)^2]}{\sigma\sqrt{2\pi}}$$

which we can write as

$$f(y_i^*) = \frac{\phi((y_i^* - \mu_{i_X})/\sigma)}{\sigma}$$

where ϕ is the pdf of a normal distribution with $mean = 0$ and $variance = 1$.

The probability that y_i^* is censored equals

$$P(y_i^* \leq LOD) = \Phi((LOD - \mu)/\sigma)$$

where Φ is the cdf of a normal distribution with $mean = 0$ and $variance = 1$.

Every y_i^* is either censored or not, so we will use the indicator variable $I = 1$ for censored, and $I = 0$ for not censored. Moreover, we assume that y_i are all independent, which means that the joint likelihood over all observations is the product of the density functions for all y_i.

This gives us the likelihood function L

$$L = \prod_{i=1}^{A} [[(1/\sigma)\phi((y_i - \mu_{i_X})/\sigma)]^{1-I} \times \Phi((LOD - \mu_{i_X})/\sigma)^I]$$

So the log-likelihood function is

$$\log(L) = \sum_{i=1}^{A} [(1-I)[\log(\phi((y_i - \mu_{i_X})/\sigma)) - \log(\sigma)] + I \times \log[\Phi((LOD - \mu_{i_X})/\sigma)]]$$

which equals

$$\log(L) = \sum_{i=1}^{A}[(1-I)[\log(\phi((y_i-(\alpha_X+\beta_X x_i))/\sigma))-\log(\sigma)]+I\times\log[\Phi((LOD-(\alpha_X+\beta_X x_i))/\sigma)]]$$

We will use the `censReg()` function from the censReg package in R to maximise this log-likelihood function to obtain the maximum likehood estimates $\widehat{\alpha_X}$, $\widehat{\beta_X}$ and $\hat{\sigma}$.

We will then perform imputation as follows.

Every censored value y_i is substituted by the expected value of a truncated normal distribution, which we describe as originating from a normal distribution with $mean = \widehat{\mu}_{i_X} = \widehat{\alpha}_X + \widehat{\beta}_X x_i$, $variance = \sigma^2$, and with truncation at $y = LOD$.

In our practice, we used the `etruncnorm()` function from the `truncnorm` R package to calculate every such expected value.

We will use the term "completed dataset" for every dataset that results from imputation or substitution.

Variations on the `censReg1` method

We will also use two methods that are closely related to `censReg1` for the purpose of comparison; we call these methods `censReg1naive` and `censReg2`.

The `censReg1naive` method

The only difference from `censReg1` in our `censReg1naive` method is that the latter uses the corresponding non-truncated normal distribution rather than the truncated one. Our conjecture is that estimates of β from `censReg1naive` will have significantly higher bias than the corresponding estimates from `censReg1`. Our rationale is that the censored y_i values could be substituted by values that are higher than LOD, whereas the true value is known to be not higher than LOD.

`censReg1naive` is the same as `censReg1` except that a non-truncated normal distribution is used in the imputation step. We conjecture that this will result in estimates with higher bias than from `censReg1`. 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.

The censReg2 method

censReg2 uses two predictor variables X and N.

We conjecture that using one additional redundant predictor variable will result in estimates with higher variance than from censReg1.

The mathematical formulation for this method corresponds to that presented above for censReg1, except that we model each observation y_i^* as from a normal distribution with mean $\mu_{i_{X,A}} = \alpha_{X,A} + \beta_X x_i + \beta_A a_i$ and variance σ^2.

This means that the likelihood function for censReg2 will have the same form as that for censReg1; it will differ only in having $\mu_{i_{X,A}}$ in place of μ_{i_X}.

Consequently, maximisation of the corresponding log-likelihood function gives the maximum likehood estimates $\widehat{\alpha_X}$, $\widehat{\beta_X}$, $\widehat{\beta_X}$ and $\hat{\sigma}$.

Therefore, every censored value y_i is substituted by the expected value of a truncated normal distribution, which we describe as originating from a normal distribution with $mean = \widehat{\mu}_{i_{X,A}} = \widehat{\alpha}_X + \widehat{\beta}_X x_i + \widehat{\beta}_A a_i$, $variance = \sigma^2$, and with truncation at $y = LOD$.

Estimation of β directly from the censored dataset by censored regression by the censReg0impute method

This method differs from censReg1 in the choice of predictor variable for the model for the maximum likelihood estimation step.

We have seen that the censReg1 method uses X_i as the predictor for this step.

The censReg0impute method uses A_i as the predictor instead for this step.

Thus the estimate $\widehat{\beta_A}$ is found directly from the maximisation of the corresponding log-likelihood function, without any imputation step.

We conjecture that since $|\beta_X| = 0.79$ is much greater than $|\beta_A|$, censReg0impute will result in estimates with higher variance than from censReg1.

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 = Biasˆ2 + 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 corre-

lation between cb28 and cb153 is weaker, which results in less accurate imputation by ´censReg1andcensReg2‘, 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.

```
##          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.

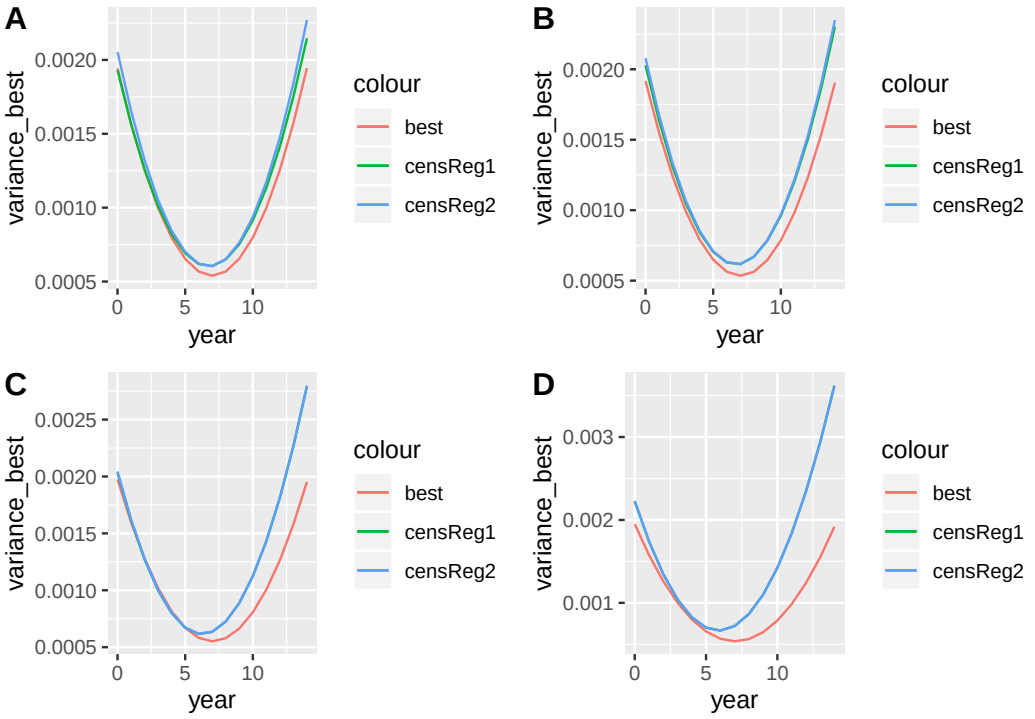
Variance of predictions of cb28 annual means from different methods

Predictions for different values of beta153year

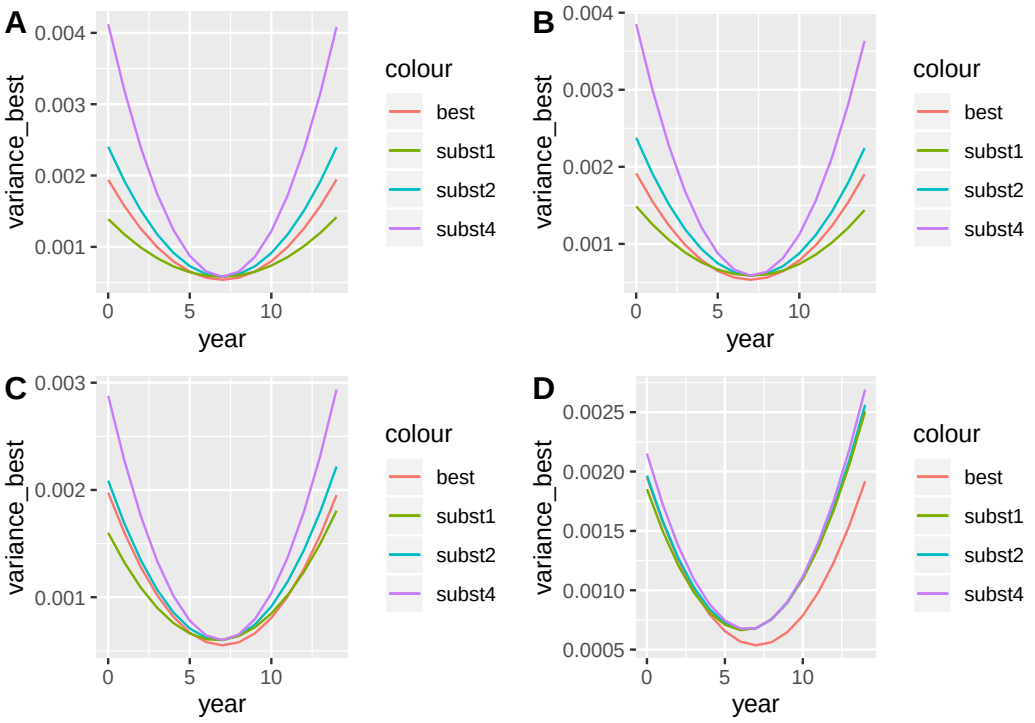
We will begin by using the same parameter values we used in our earlier section "Evaluation of methods for larger absolute values of `beta28year`". These parameters are fixed: cprop = 0.3, sd28_153 = 0.3, whilst cb153year is given four values: -0.02, -0.04, -0.08 and -0.16 respectively.

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.

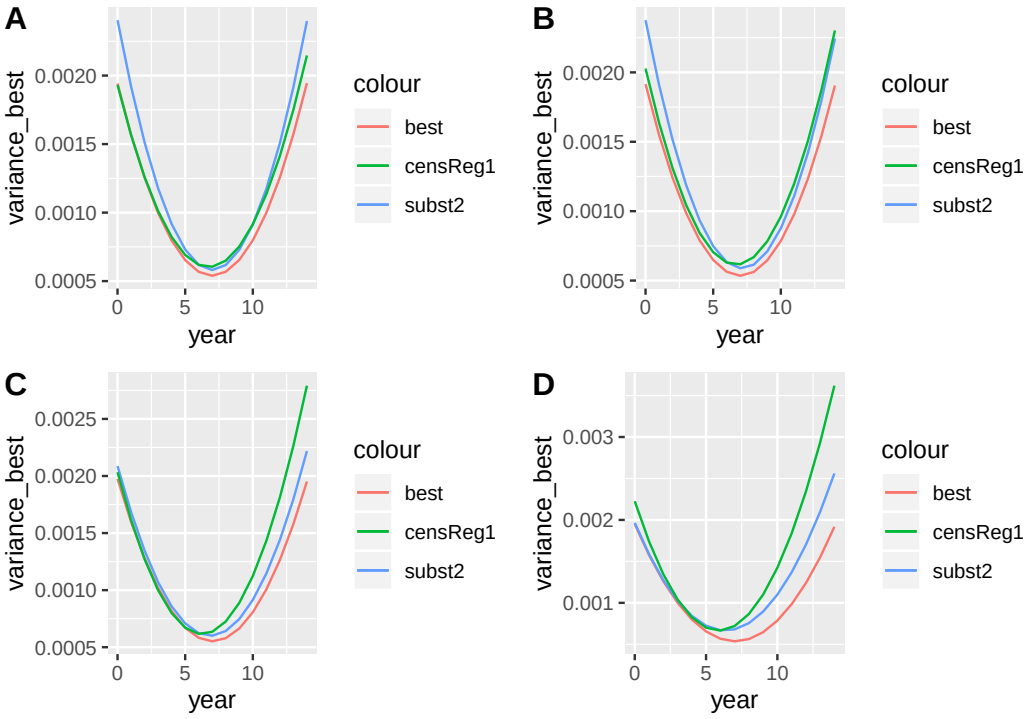
Our first set of four graphs show the variance of `censReg1` and `censReg2` methods relative to `best` method for beta153year equal to -0.02, -0.04, -0.08, -0.16, respectively.



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.

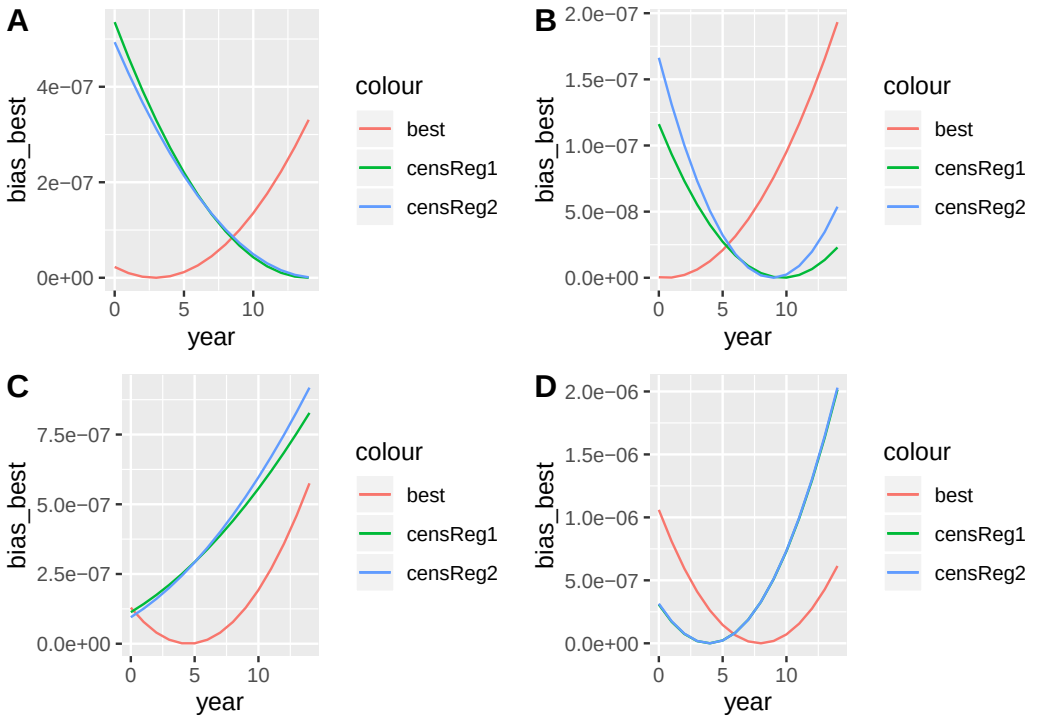


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

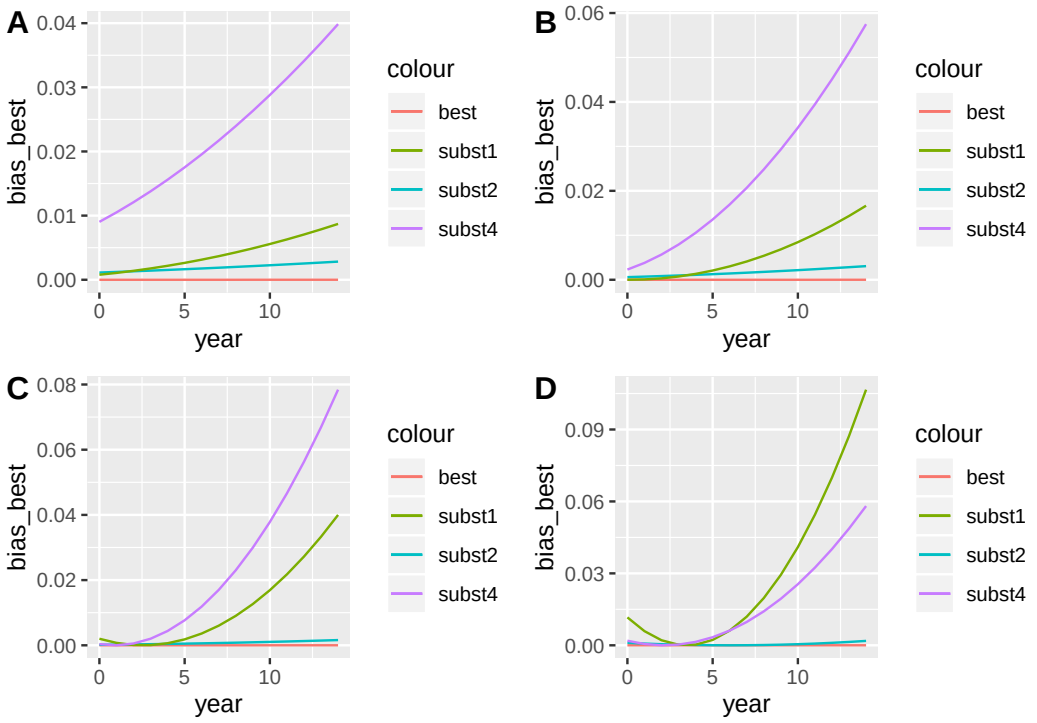


We will now show graphs of the bias of predictions of cb28 annual means from our chosen censoring methods.

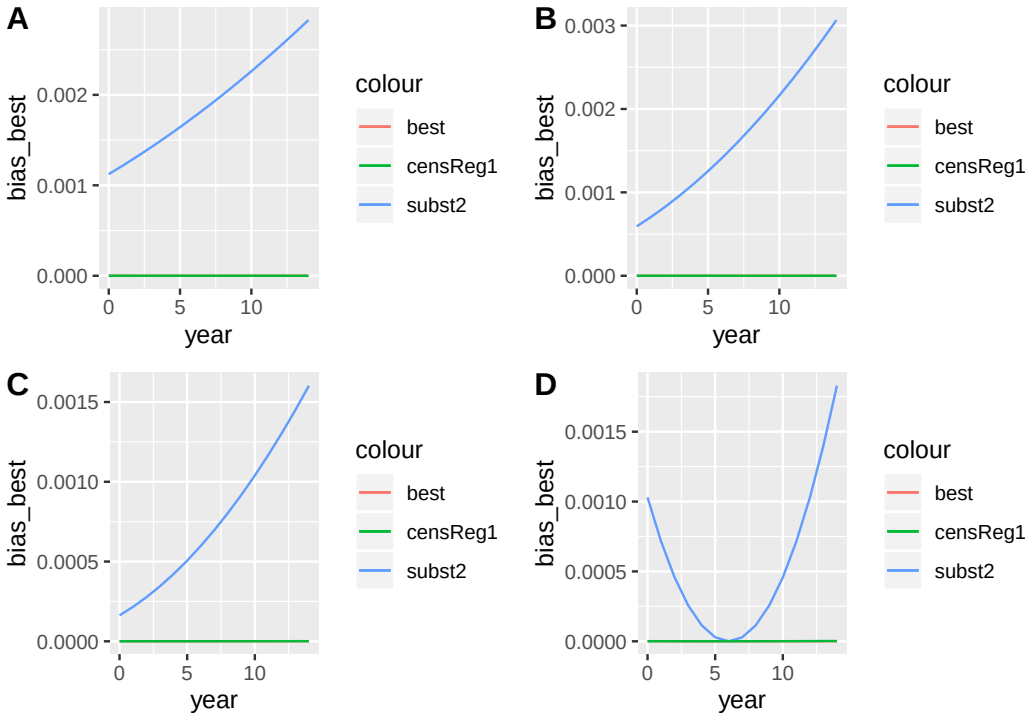
Our first set of four graphs show the bias of `censReg1` and `censReg2` methods relative to `best` method for beta153year equal to -0.02, -0.04, -0.08, -0.16, respectively.



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.

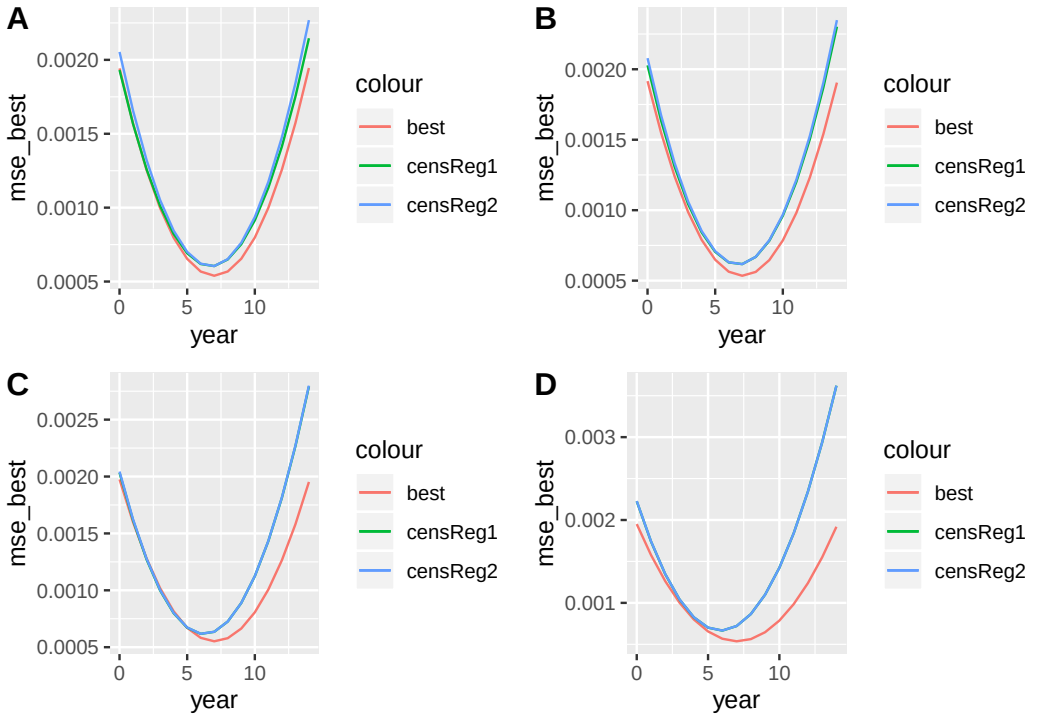


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

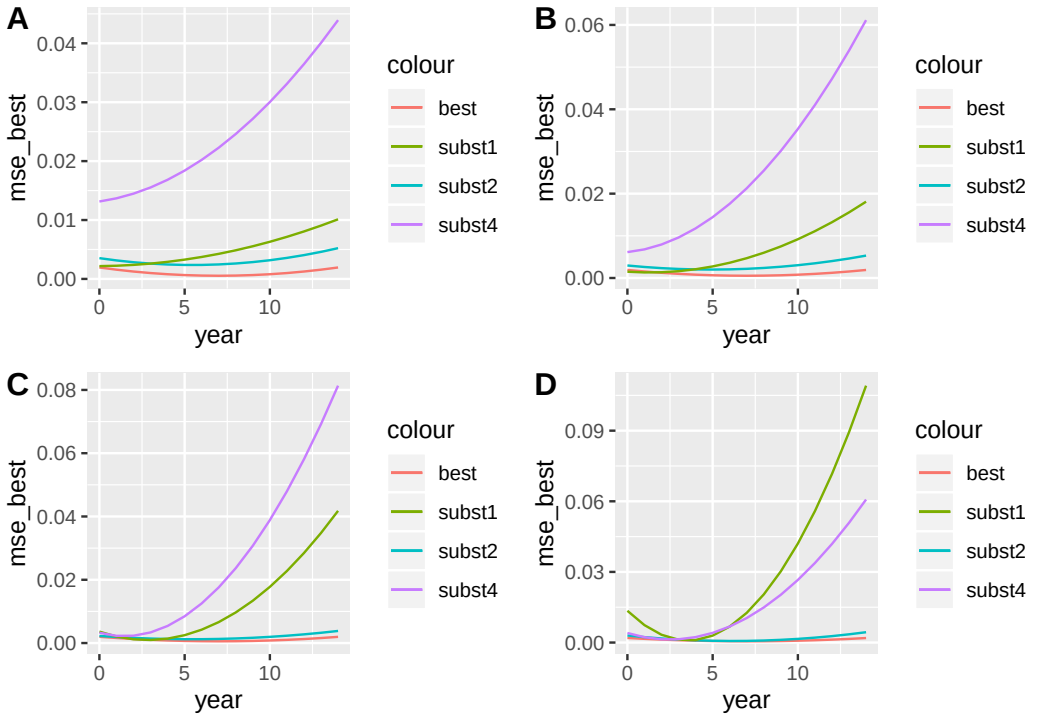


We will now show graphs of the MSE of predictions of cb28 annual means from our chosen censoring methods.

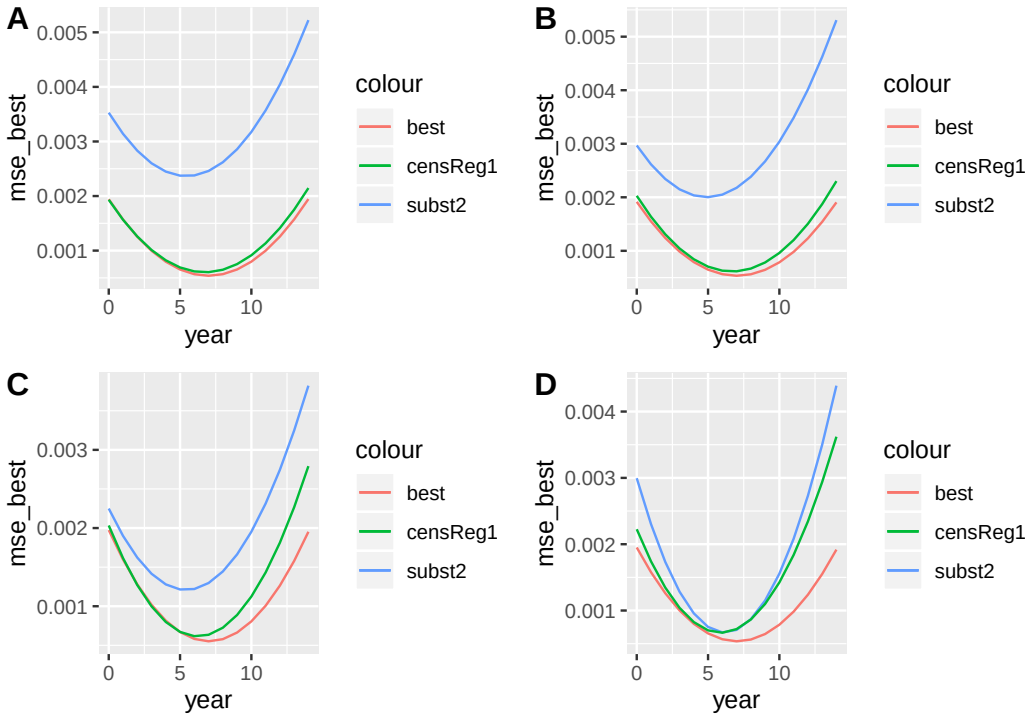
Our first set of four graphs show the MSE of `censReg1` and `censReg2` methods relative to `best` method for beta153year equal to -0.02, -0.04, -0.08, -0.16, respectively.



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.

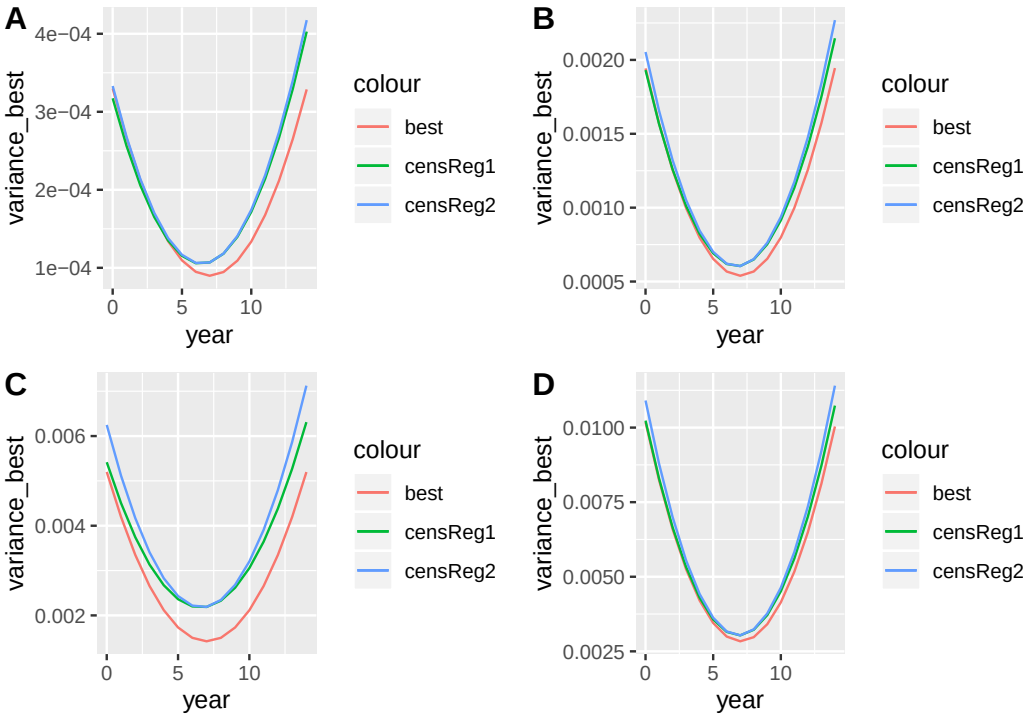


Predictions for different values of sd28vs153

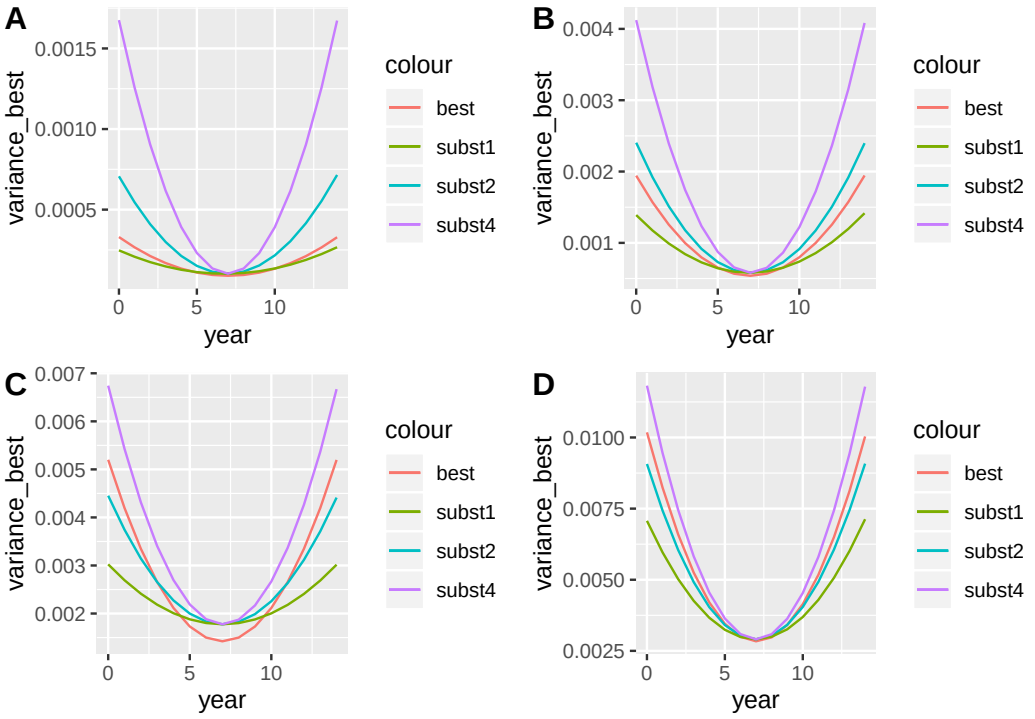
For all our predictions in this section, these parameters are fixed: cprop = 0.3, cb153year = -0.02, whilst sd28_153 is given four values: 0.1, 0.3, 0.5 and 0.7 respectively.

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.

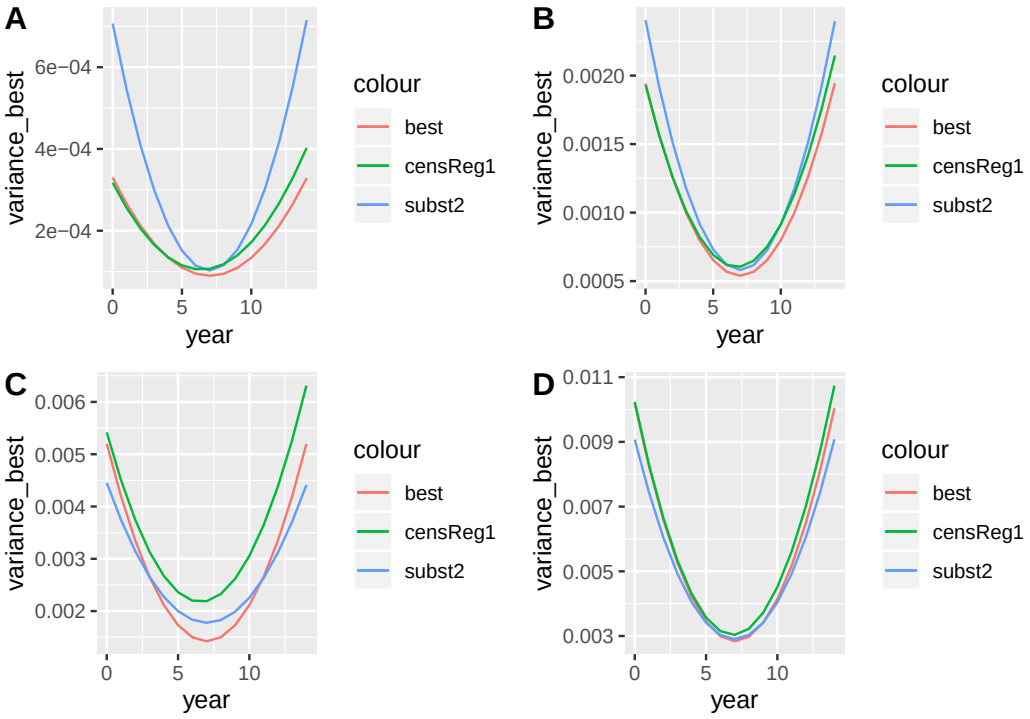
Our first set of four graphs show the variance of `censReg1` and `censReg2` methods relative to `best` method for sd28_153 equal to 0.1, 0.3, 0.5, 0.7, respectively.



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.

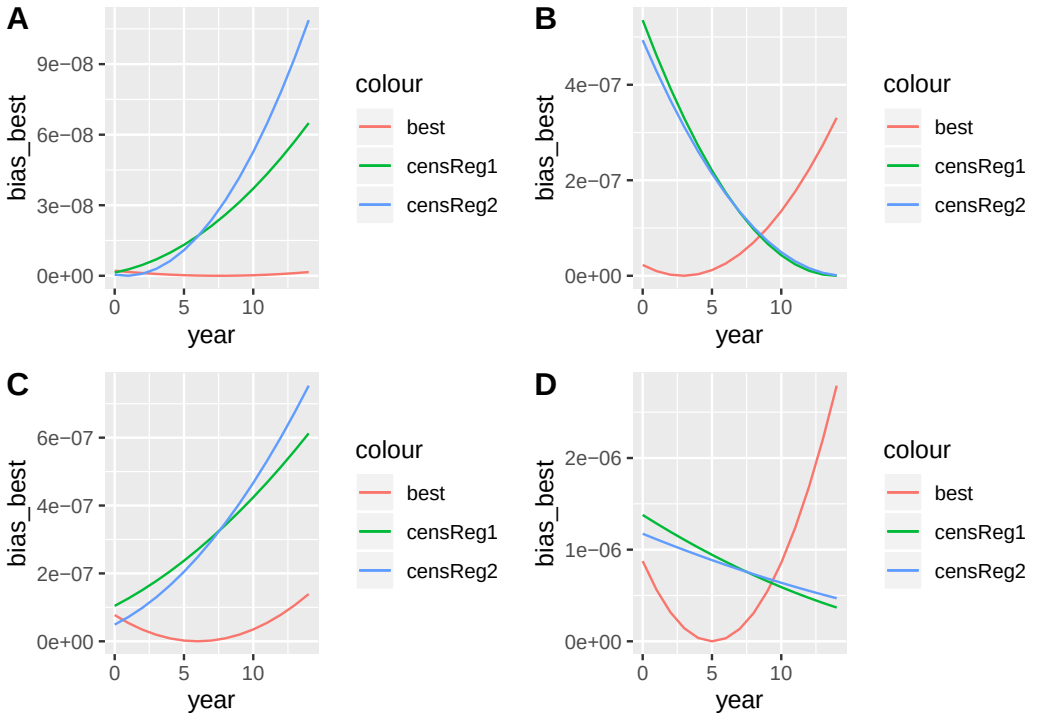


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

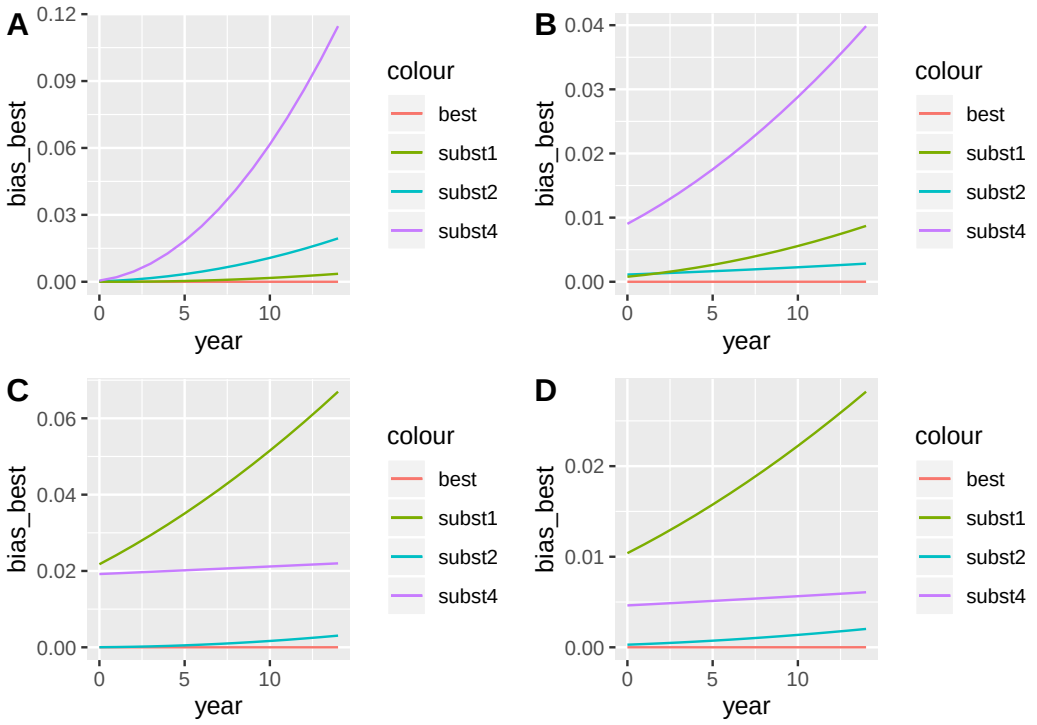


We will now show graphs of the bias of predictions of cb28 annual means from our chosen censoring methods.

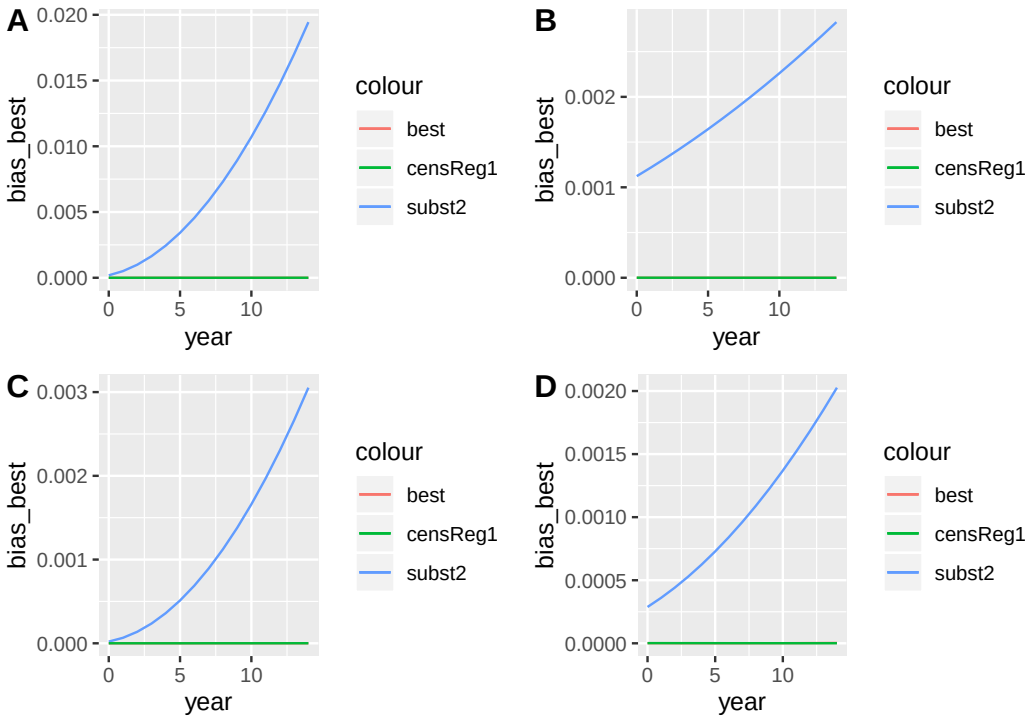
Our first set of four graphs show the bias of `censReg1` and `censReg2` methods relative to `best` method for sd28_153 equal to 0.1, 0.3, 0.5, 0.7, respectively.



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.

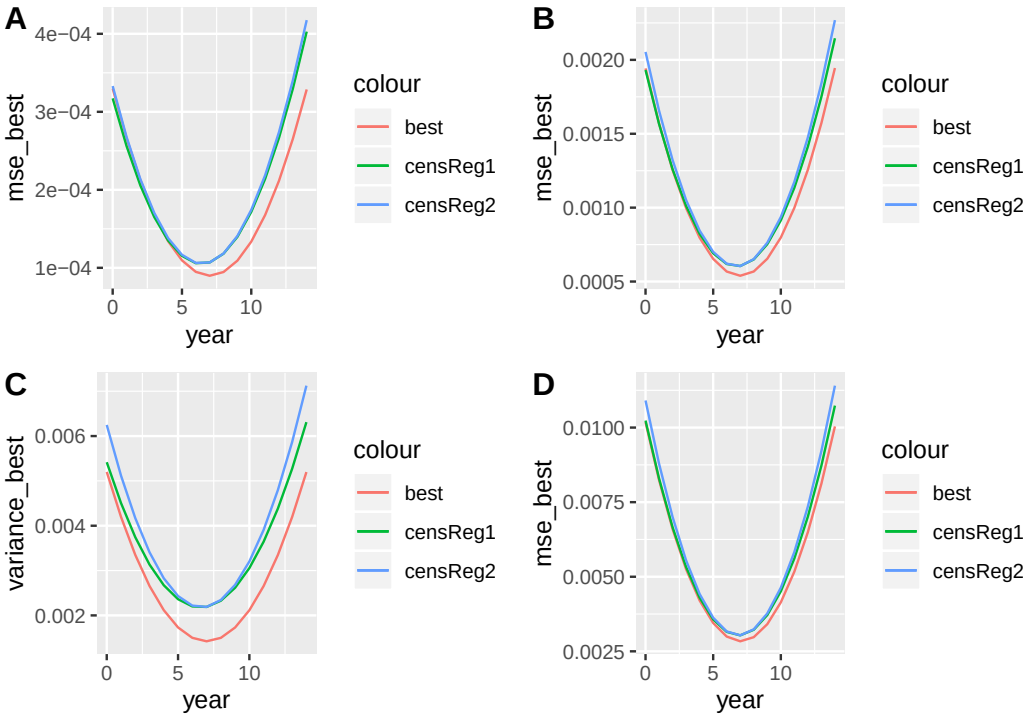


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

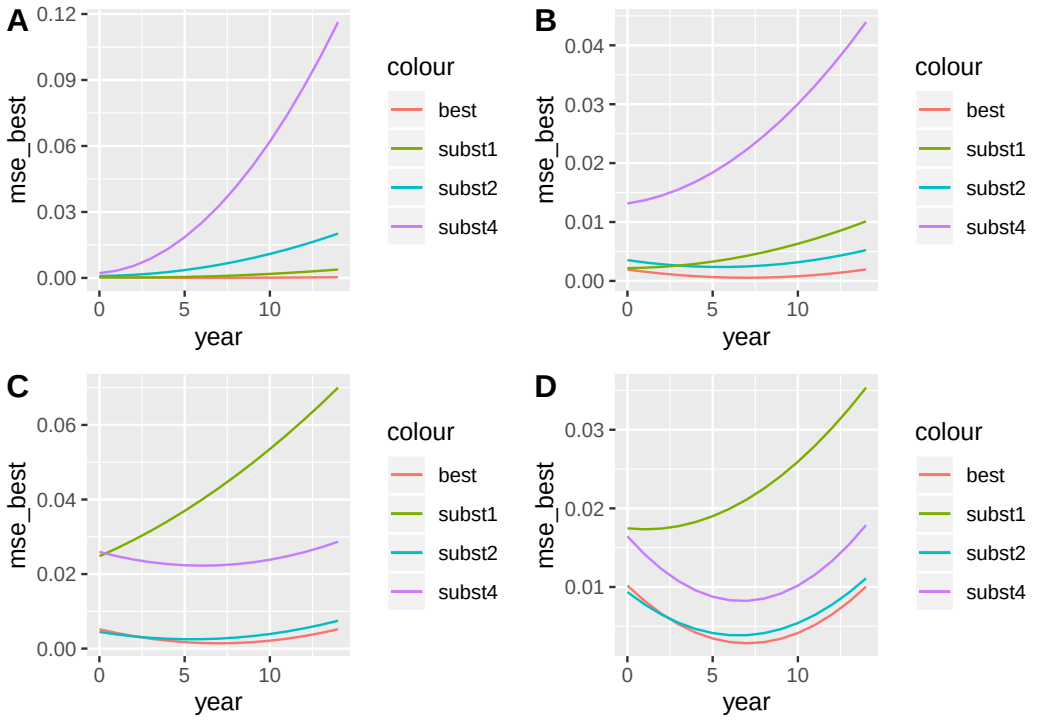


We will now show graphs of the MSE of predictions of cb28 annual means from our chosen censoring methods.

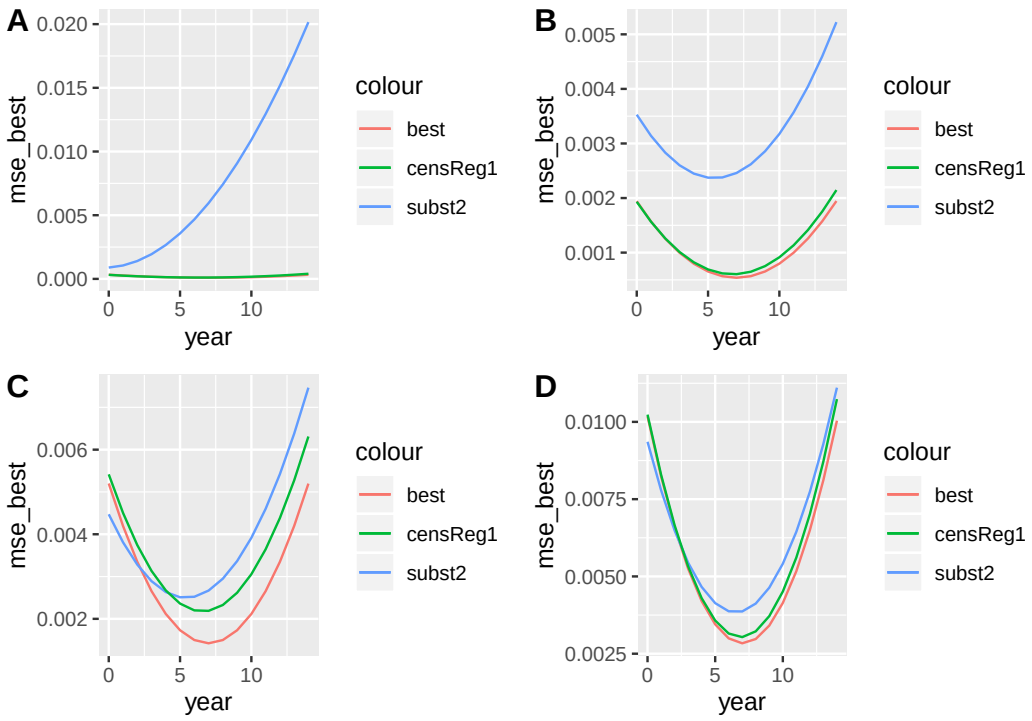
Our first set of four graphs show the MSE of `censReg1` and `censReg2` methods relative to `best` method for sd28_153 equal to 0.1, 0.3, 0.5, 0.7, respectively.



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.

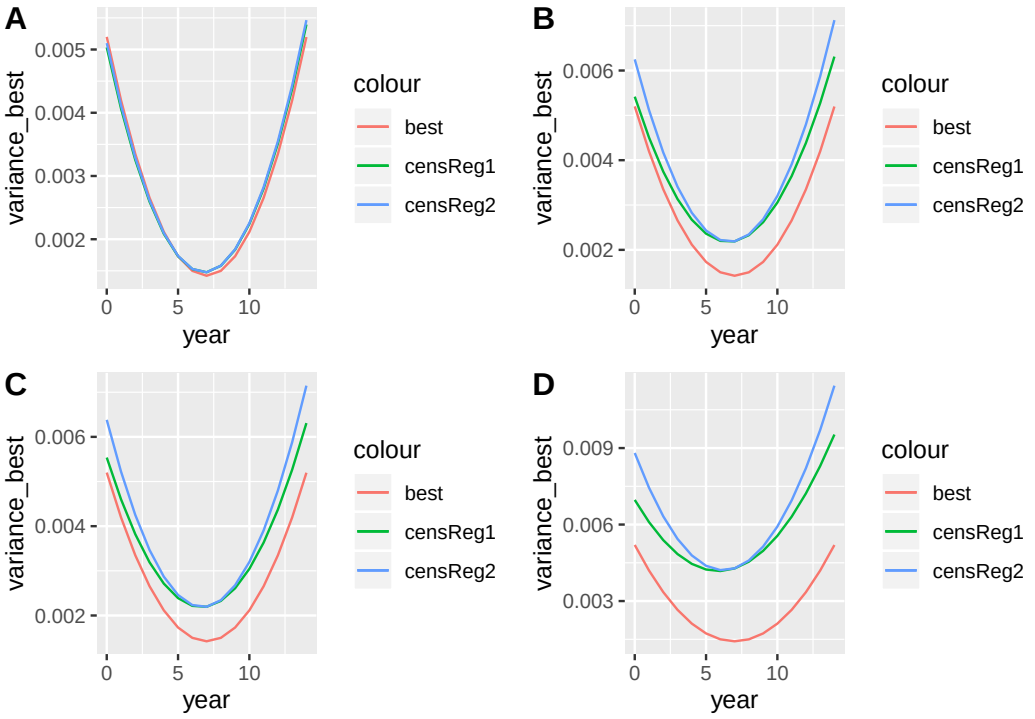


Predictions for different values of cprop

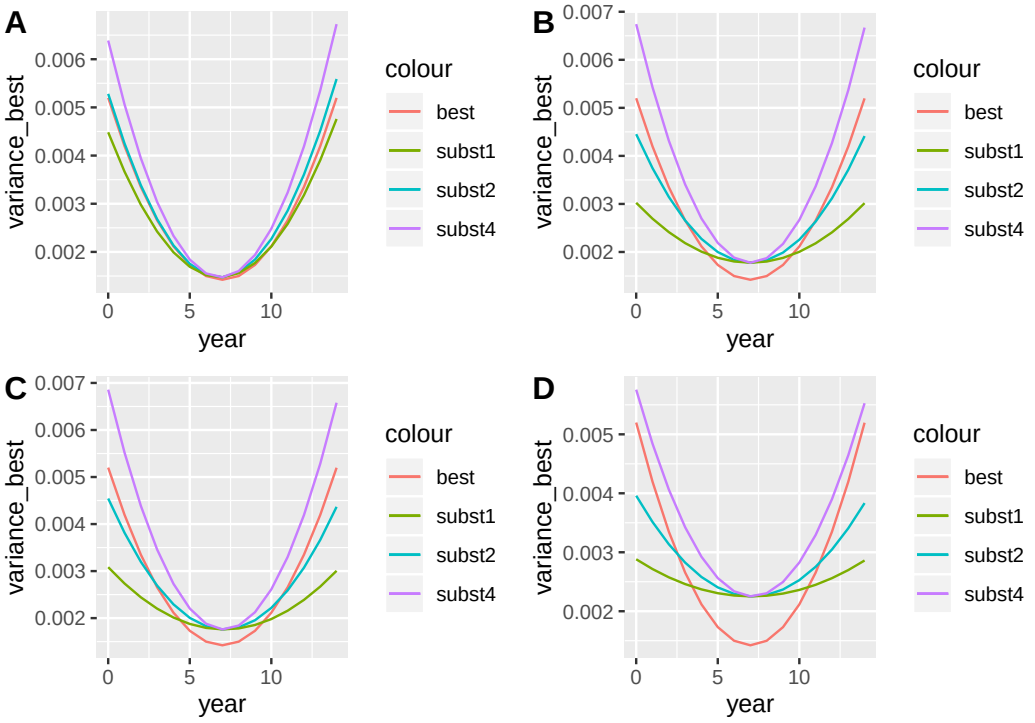
For all our predictions in this section, these parameters are fixed: sd28_153 = 0.5, cb153year = -0.02, whilst cprop is given four values: 0.1, 0.3, 0.5 and 0.7 respectively.

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

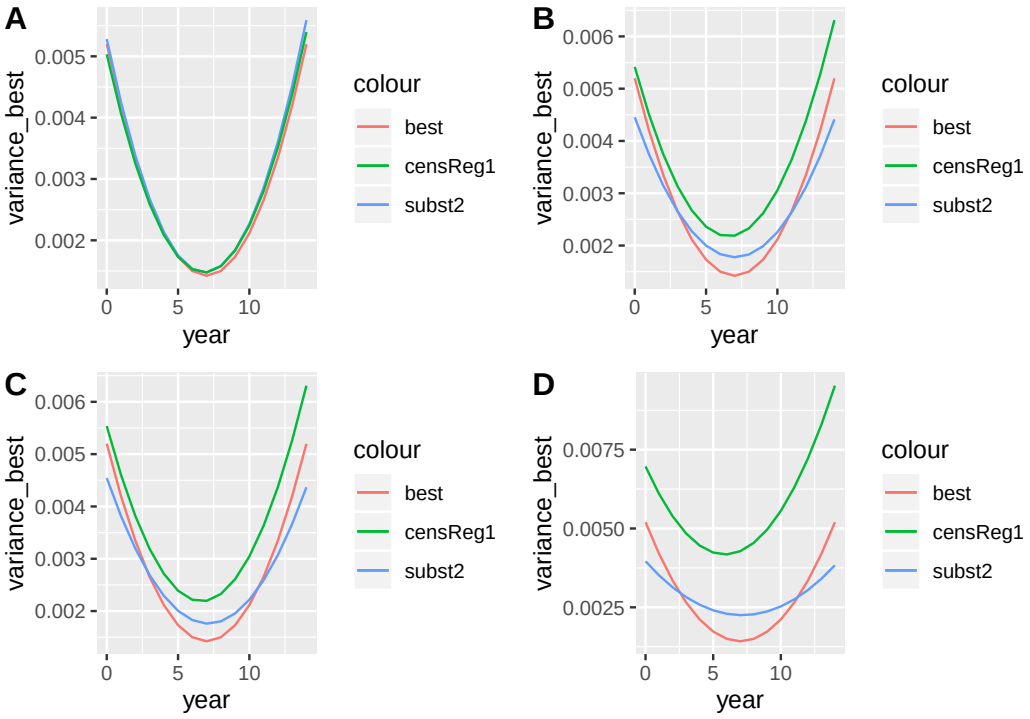
Our first set of four graphs show the variance of `censReg1` and `censReg2` methods relative to `best` method for cprop equal to 0.1, 0.3, 0.5, 0.7, respectively.



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.

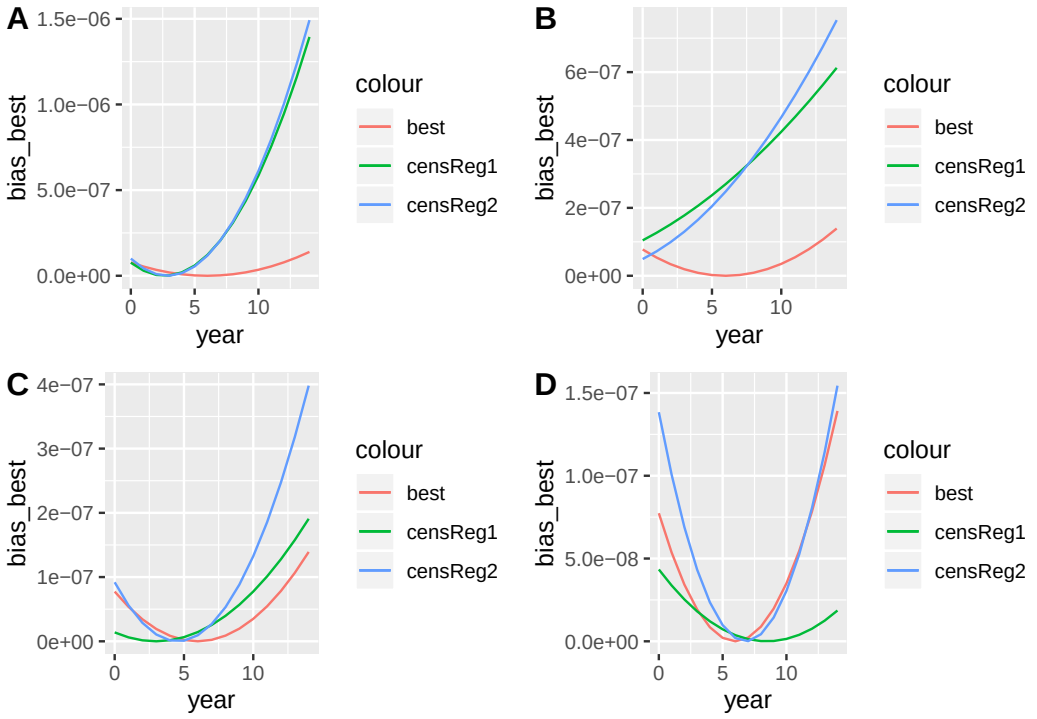


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

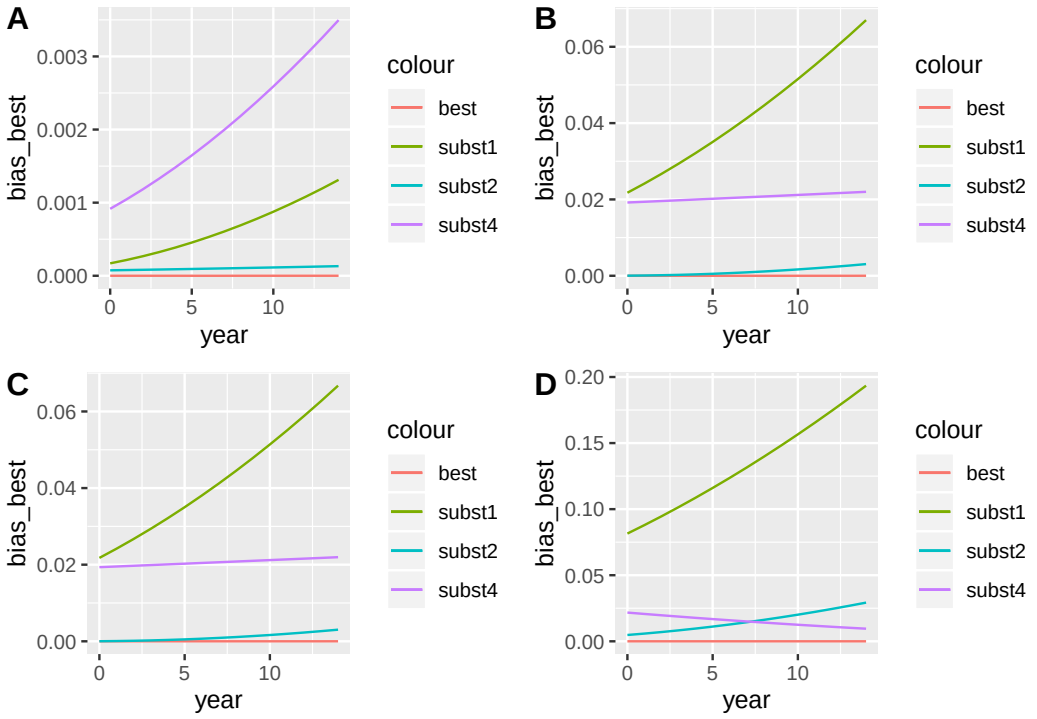


We will now show graphs of the bias of predictions of cb28 annual means from our chosen censoring methods.

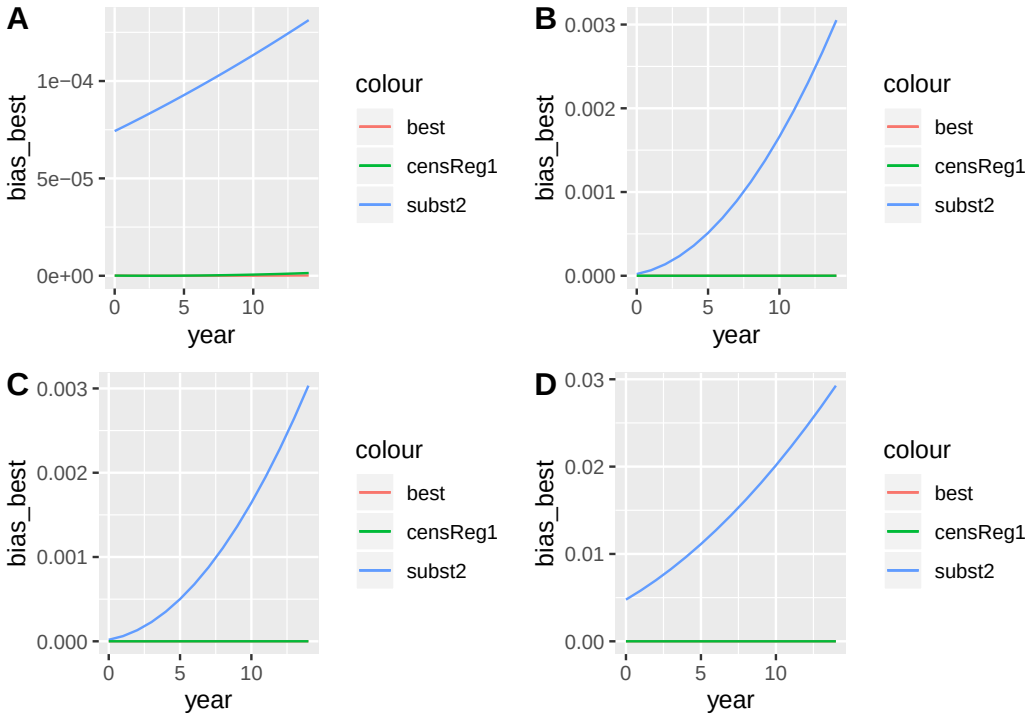
Our first set of four graphs show the bias of `censReg1` and `censReg2` methods relative to `best` method for cprop equal to 0.1, 0.3, 0.5, 0.7, respectively.



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.

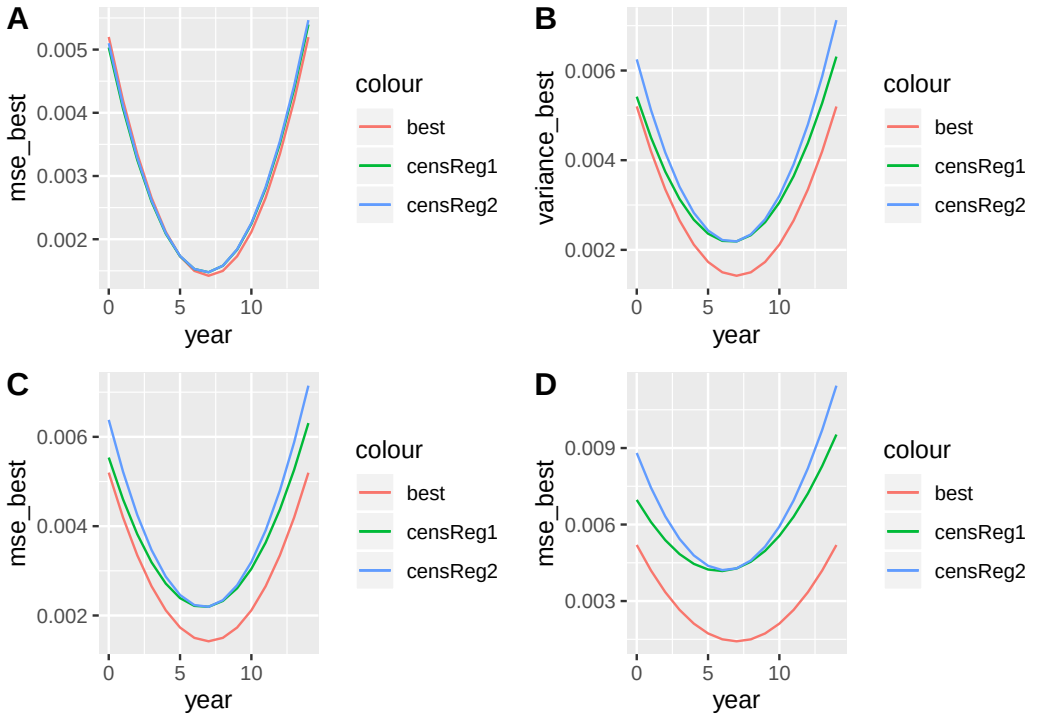


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

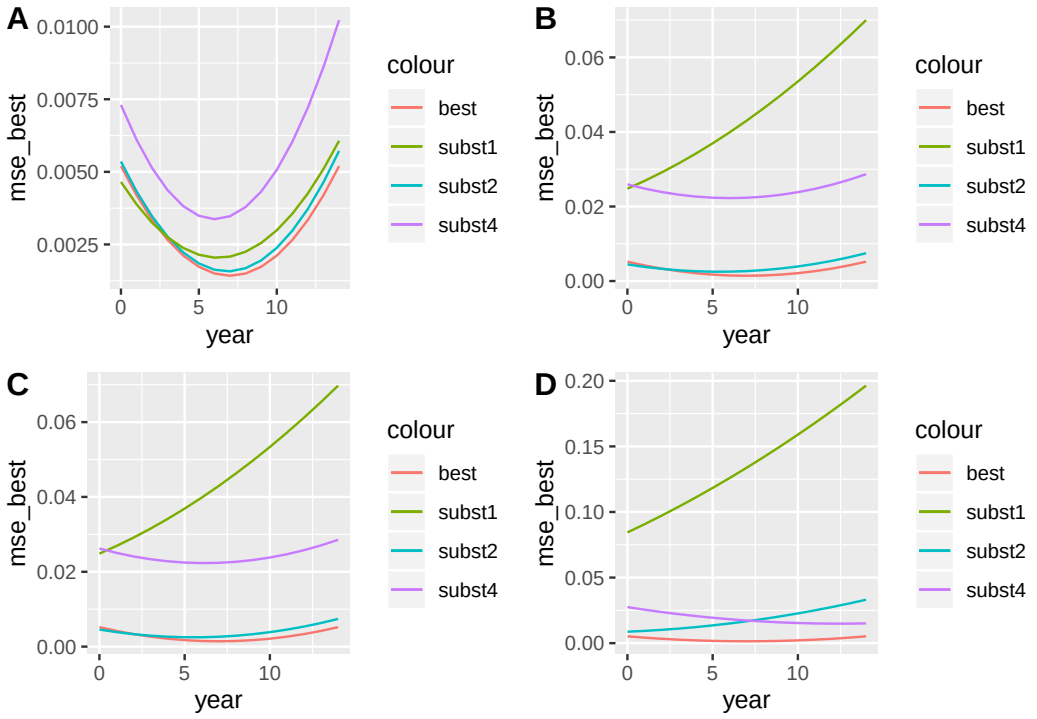


We will now show graphs of the MSE of predictions of cb28 annual means from our chosen censoring methods.

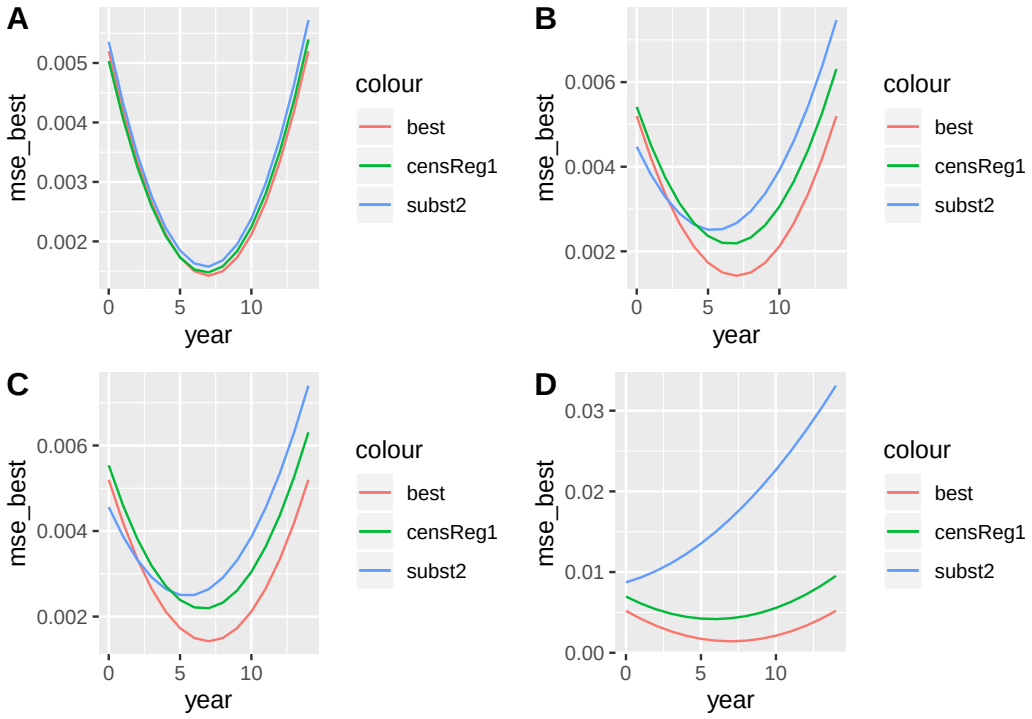
Our first set of four graphs show the MSE of `censReg1` and `censReg2` methods relative to `best` method for cprop equal to 0.1, 0.3, 0.5, 0.7, respectively.



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.

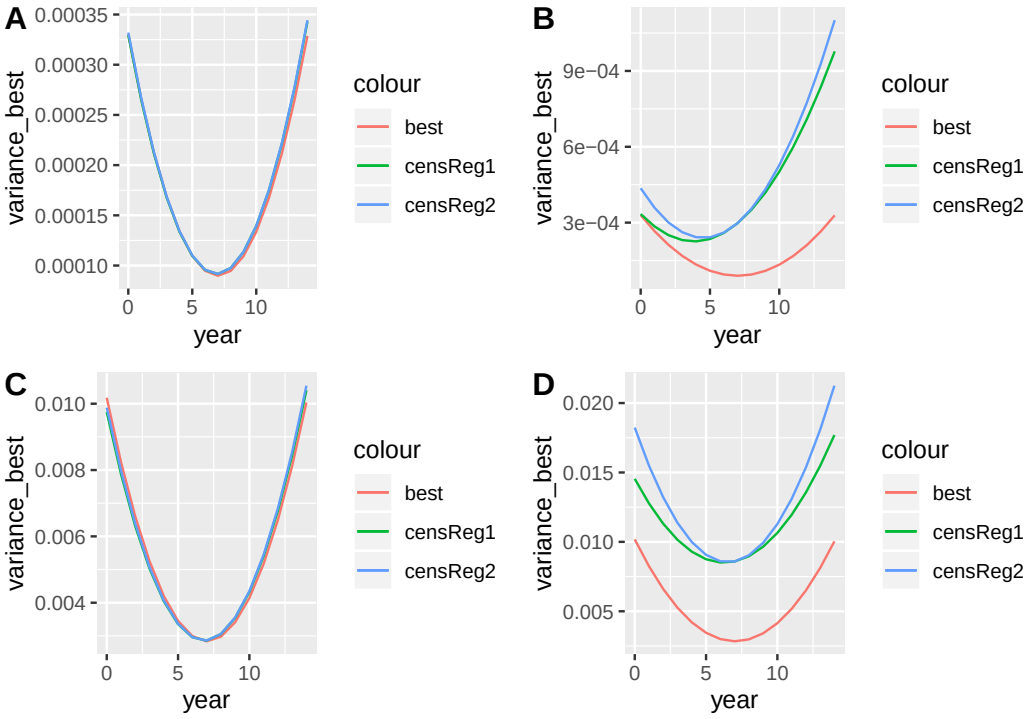


Predictions for low-low, high-low, low-high, high-high values of sd28vs153-cprop

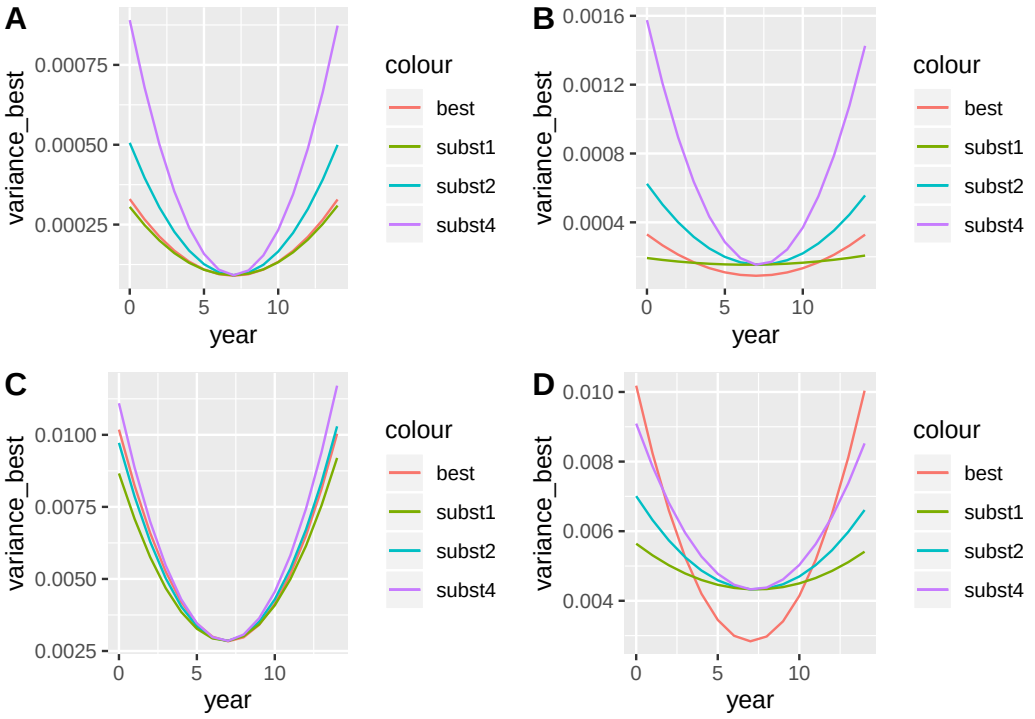
We will now use the same sets of parameter values that we used in our earlier section "Selection of censoring methods for further study". Concretely: beta28year = -0.02 is held fixed, whilst a "low" and a "high" value for each of `cprop` and `sd28_153` are used. Concretely: (0.1, 0.1), (0.7, 0.1), (0.1,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

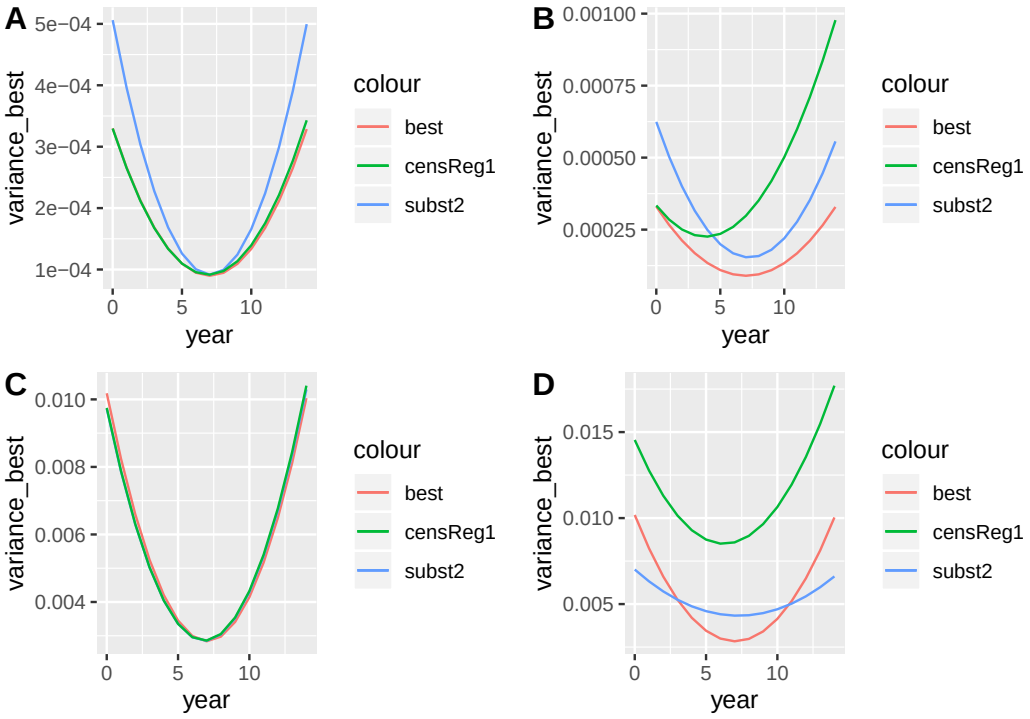
Our first set of four graphs show the variance of `censReg1` and `censReg2` methods relative to `best` method for (sd28_153, cprop) equal to (0.1, 0.1), (0.1, 0.7), (0.7, 0.1) and (0.7, 0.7), respectively.



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.

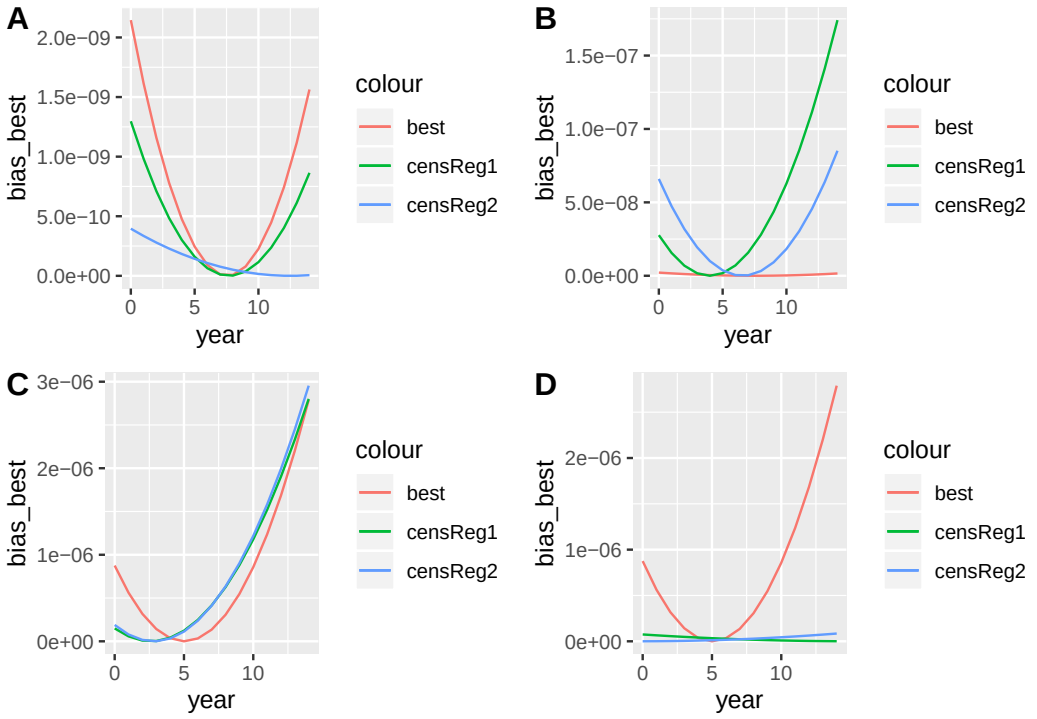


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

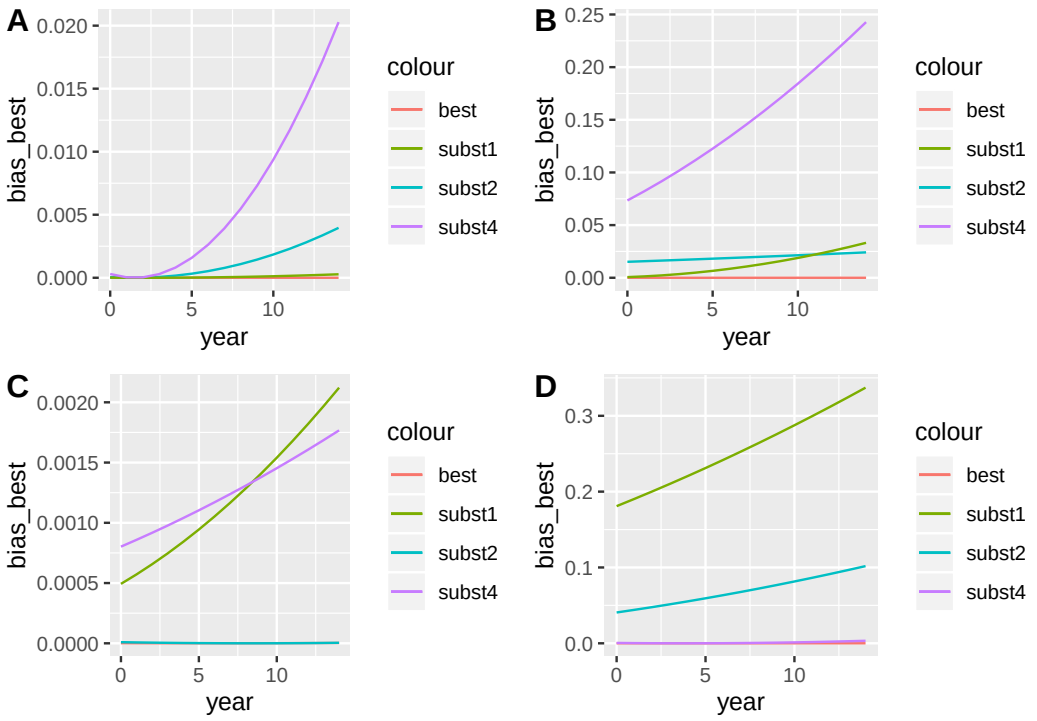


We will now show graphs of the bias of predictions of cb28 annual means from our chosen censoring methods.

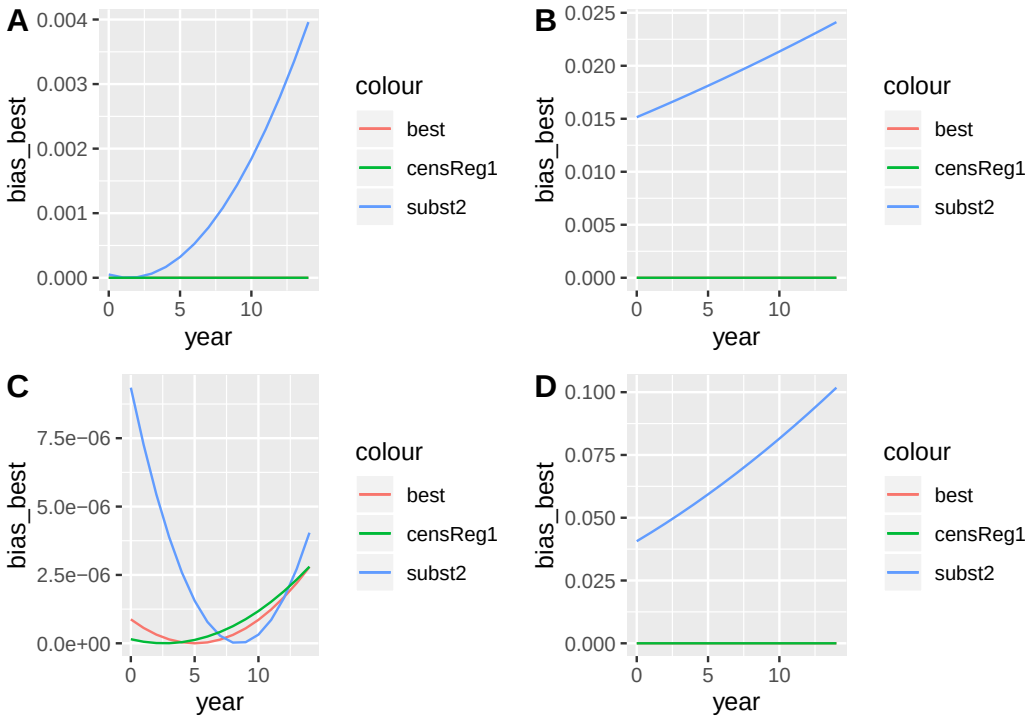
Our first set of four graphs show the bias of `censReg1` and `censReg2` methods relative to `best` method for (sd28_153, cprop) equal to (0.1, 0.1), (0.1, 0.7), (0.7, 0.1) and (0.7, 0.7), respectively.



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.

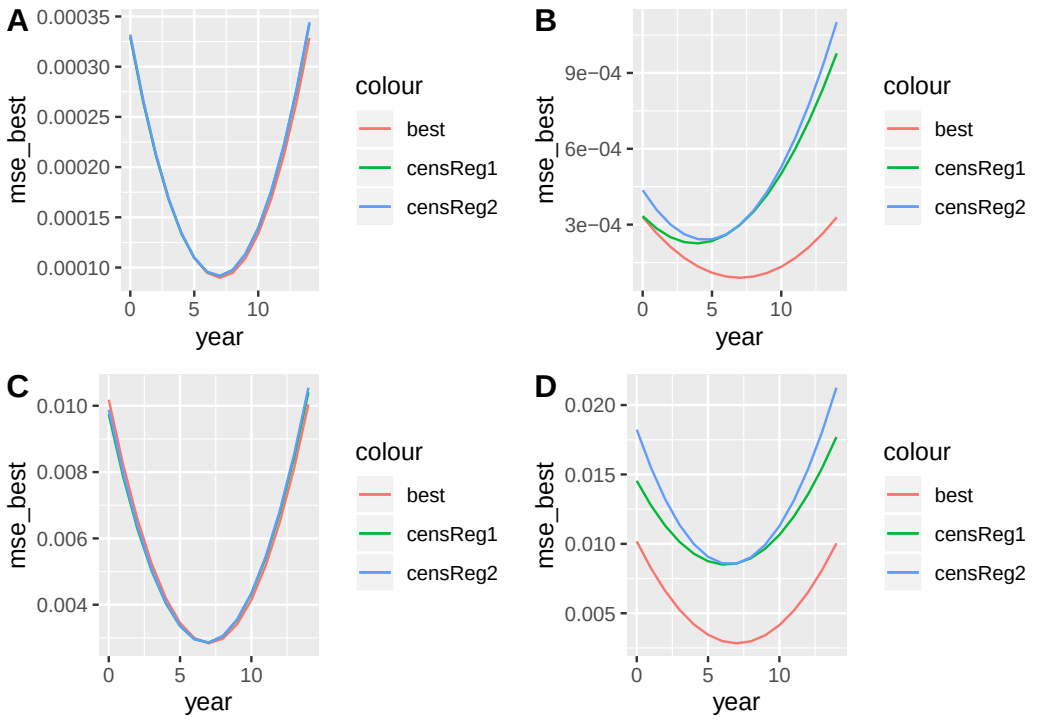


Our third set of four graphs simply displays the results from the `subst2`, `censReg1` and `best` methods together on the same plot, which is displayed below.

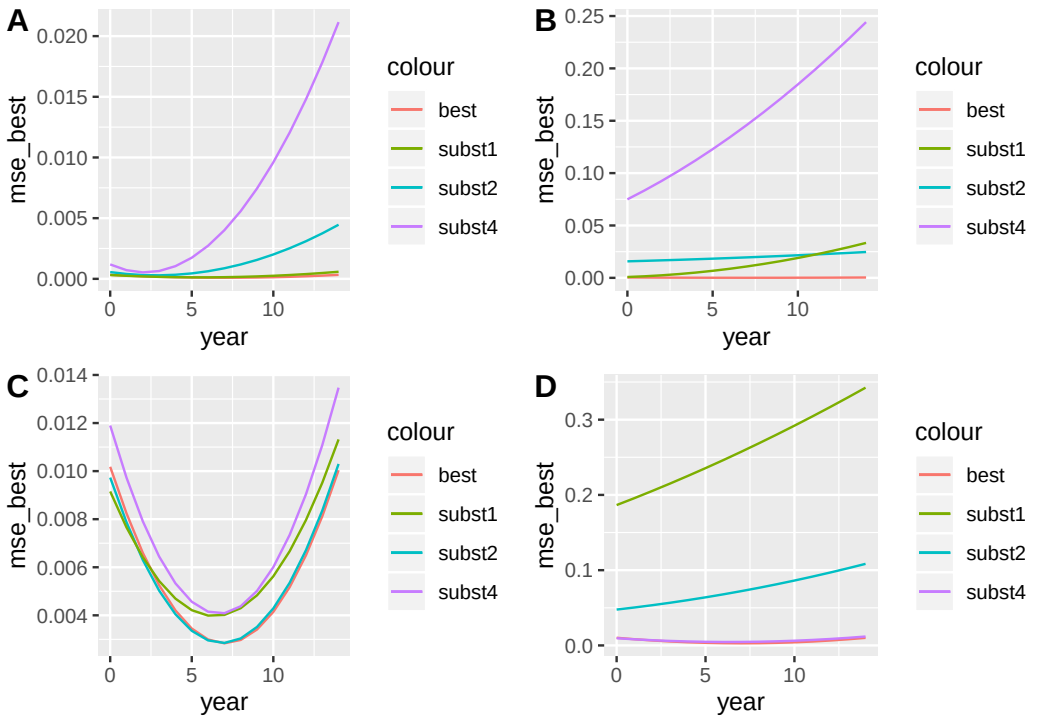


We will now show graphs of the MSE of predictions of cb28 annual means from our chosen censoring methods.

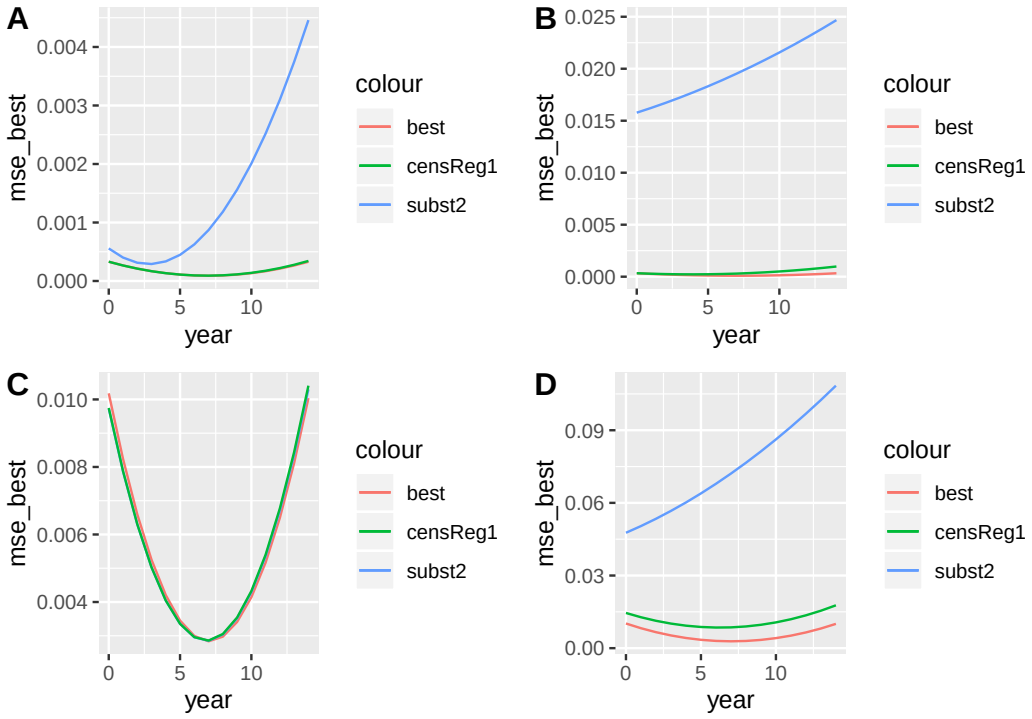
Our first set of four graphs show the MSE of `censReg1` and `censReg2` methods relative to `best` method for (sd28_153, cprop) equal to (0.1, 0.1), (0.1, 0.7), (0.7, 0.1) and (0.7, 0.7), respectively.



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.


A
mse_best
0.004
0.003
0.002
0.001
0.000
0
5
10
year
colour
best
censReg1
subst2
B
mse_best
0.025
0.020
0.015
0.010
0.005
0.000
0
5
10
year
colour
best
censReg1
subst2
C
mse_best
0.010
0.008
0.006
0.004
0
5
10
year
colour
best
censReg1
subst2
D
mse_best
0.09
0.06
0.03
0.00
0
5
10
year
colour
best
censReg1
subst2