Updated:April 17, 2022

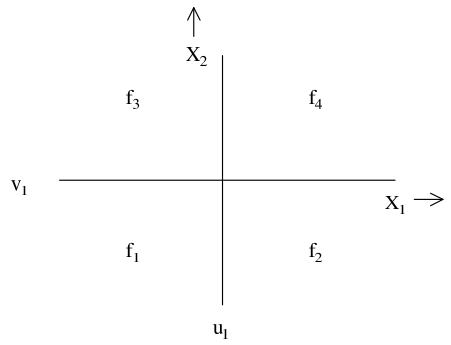
Assignment 1
Assigned 2022-01-16 Due 2022-01-23 9p

This is a individual assignment except for problem 8 for which students may work together and is for extra credit. For problems 1–7 your work must be completely independent. Your work must be a finished `html` (preferred) or `pdf` file. Submit it by sending a direct Zulip message to the TAs and to the instructor with the document as an attachment to the message.

1. Problem 1 in Chapter 2 of Regression Modeling Strategies
2. Problem 2
3. Problem 3
4. Problem 4
5. Problem 5. The SAT dataset may be created by using the `sat.r` code available from `hbiostat.org/doc/rms/sat.r`.
6. Derive the formulas for the restricted cubic spline component variables without cubing or squaring any terms.
7. Prove that each component variable is linear in X when $X \geq t_k$, the last knot, using general principles and not algebra or calculus. Derive an expression for the restricted spline regression function when $X \geq t_k$.
8. Consider a 3–dimensional surface relating X_1 and X_2 to $C(Y|X_1, X_2)$ defined by a patch–wise cubic polynomial. The patches are formed by a grid of knots for X_1 $(u_1, \ldots, u_k)$ and for X_2 $(v_1, \ldots, v_k)$. Each polynomial is of the form

$$f(X_1, X_2) = \sum_{i=j=0}^{3} \beta_{ij} X_1^i X_2^j,$$

but is written in terms of an offset from polynomials below and to the left to facilitate continuity restrictions in a) below. For example, if $k = 1$, define four polynomials for four quadrants



For bivariate knots (u_1, v_1) let $f_2 = f_1+$ terms in $(X_1 - u_1), f_3 = f_1+$ terms in $(X_2 - v_1)$, and f_4 involves both "knot crossings." Derive equations for $f(X_1, X_2)$ or for its component terms under the following two conditions.

(a) Restrict this 3–dimensional spline function so that $f(\cdot)$ is continuous and has continuous first and second derivatives — $f(\cdot)$ agrees at the rectangle boundaries and so does $\frac{\partial f}{\partial X_1}$, $\frac{\partial f}{\partial X_2}$, and $\frac{\partial^2 f}{\partial X_1 \partial X_2}$.

(b) Further restrict $f(\cdot)$ so that f is of the form $aX_1 + bX_2 + cX_1X_2$ if $X_1 \leq u_1$ and $X_2 \leq v_1$ or if $X_1 \geq u_k$ and $X_2 \geq v_k$.

Assignment 2 Assigned 2022-01-25 Due 2022-02-01

This is a group assignment. Groups are defined in the following table. One solution should be turned in per group, and the work should list all members who contributed meaningfully to the work. Groups must not help each other except for general questions posted to `Zulip`.

	Student	Group
1	Siwei Zhang	1
2	Marisa Hayli Blackman	1
3	Zoey Song	1
4	Max Rohde	2
5	Huiding Chen	2
6	Jackson Resser	2
7	Cara Tanaka Lwin	3
8	Ruby Xiong	3
9	Sarah Torrence	3
10	Justin Leon Jacobs	3

Consider a two–stage procedure in which one tests for linearity of the effect of a predictor X on a property of the response $C(Y|X)$ against a quadratic alternative. If the two–tailed test of linearity is significant at the α level, a two d.f. test of association between X and Y is done. If the test for linearity is not significant, the square term is dropped and a linear model is fitted. The test of association between X and Y is then (apparently) a one d.f. test.

1. Write a formal expression for the test statistic for association.
2. Write an expression for the nominal P–value for testing association using this strategy.
3. Write an expression for the actual P–value or alternatively for the type–I error if using a fixed critical value for the test of association.
4. For the same two–stage strategy consider an estimate of the effect on $C(Y|X)$ of increasing X from a to b. Write a brief symbolic algorithm for deriving a true two–sided $1-\alpha$ confidence interval for the $b:a$ effect (difference in $C(Y)$) using the bootstrap.

Assignment 3 Assigned 2022-02-02 Due 2022-02-09

This is a group assignment. Groups are defined in the following table. One solution should be turned in per group, and the work should list all members who contributed meaningfully to the work. Groups must not help each other.

	Student	Group
1	Ruby Xiong	1
2	Cara Tanaka Lwin	1
3	Jackson Resser	1
4	Max Rohde	2
5	Siwei Zhang	2
6	Zoey Song	2
7	Huiding Chen	3
8	Marisa Hayli Blackman	3
9	Justin Leon Jacobs	3
10	Sarah Torrence	3

To access the `support` dataset you can use the command `getHdata(support)` once you have access to the Hisc package (which is automatic if you access the rms package).

1. Chapter 3 Problem 1
2. Chapter 3 Problem 2
3. State briefly why single conditional median[1] imputation is OK here.
4. Use `transcan` to develop single imputations for total cost, commenting on the strength of the model fitted by `transcan` as well as how strongly each variable can be predicted from all the others.
5. Use predictive mean matching to multiply impute cost 10 times per missing observation. Describe graphically the distributions of imputed values and briefly compare these to distributions of non-imputed values. State in a simple way what the sample variance of multiple imputations for a single observation of a continuous predictor is approximating.
6. Using the multiple imputed values, develop an overall least squares model for total cost (using the log transformation) making optimal use of partial information, with variances computed so as to take imputation (except for cost) into account. The model should use the predictors in Problem 1 and should not assume linearity in any predictor but should assume additivity. Interpret one of the resulting ratios of imputation-corrected variance to apparent variance and explain why ratios greater than one do not mean that imputation is inefficient.

[1] We are anti-logging predicted log costs and we assume log cost has a symmetric distribution

Assignment 4 Assigned 2022-02-10 Due 2022-02-20

This is a group assignment. Groups are defined in the following table. One solution should be turned in per group, and the work should list all members who contributed meaningfully to the work. Groups must not help each other.

	Student	Group
1	Marisa Hayli Blackman	1
2	Huiding Chen	1
3	Jackson Resser	1
4	Siwei Zhang	2
5	Max Rohde	2
6	Zoey Song	2
7	Ruby Xiong	3
8	Cara Tanaka Lwin	3
9	Sarah Torrence	3
10	Justin Leon Jacobs	3

The goal is to understand the performance of various internal validation methods for binary logistic models using Monte Carlo simulation. Your assignment is to modify the simulation in at least two meaningful ways with regard to covariate distribution or number, sample size, true regression coefficients, and number of times certain strategies are averaged. You should interpret your findings and give recommendations for best practice for the type of configuration you studied. Store the simulation summary in an object named `valSimresult` just as is done below so that your results can later be combined with results of other simulations. Via `Zulip` turn in the summary object `valSimresult.rda` file electronically along with your report. This file contains, for each simulation, the difference between the estimated and the independently validated statistical index. R code that you can edit is at hbiostat.org/doc/rms/sol4.Rnw. This code handles the need to escape a validation if the model would not fit (e.g., for large p), and it parallelizes the simululations if you have multiple CPU cores, greatly increasing the speed. The code also suggests how to summarize the results with dot charts. To avoid an outlying simulation result, the focus is on median absolute validation error and its exact confidence interval.

See also hbiostat.org/doc/simval.html for simulations that include a null case to check how well different methods can correct for extreme overfitting.

Simulation Method For each of 200 simulations generate a training sample of 200 observations with p predictors ($p = 15, 30, 60, 90$) and a binary reponse. The predictors are independently $U(-0.5, 0.5)$. The response is sampled so as to follow a logistic model where the intercept is zero and all regression coefficients equal 0.5 (which is admittedly not very realistic). Modify the true βs as you wish. The "gold standard" is the predictive ability of the fitted model on a test sample containing 50,000 observations generated from the same population model.

Validation Methods For each of the 200 training and validation samples several validation methods were employed to estimate how the training sample model predicts responses in the 50,000 observations. These validation methods involving fitting 40 or 200 models per training sample.

g-fold cross-validation is done using the command
`validate(f, method='cross', B=4 or B=10)` using the rms package. This was repeated and averaged using an extra loop, shown below.

For bootstrap methods `validate(f, method='boot' or '632', B=40 or B=200)` was used. `method='632'` does Efron's ".632" method, labeled `632a` in the output. An ad-hoc modification of the .632 method, `632b` was also done. Here a "bias-corrected" index of accuracy is simply the index evaluated in the observation omitted from the bootstrap re-sample.

The "gold standard" external validations were done using the `val.prob` function in the rms package.

Indexes of Predictive Accuracy

D_{xy}**:** Somers' rank correlation between predicted probability that $Y = 1$ vs. the binary Y values. This equals $2(C - 0.5)$ where C is the "ROC Area" or concordance probability.

D**:** Discrimination index — likelihood ratio χ^2 divided by the sample size

U**:** Unreliability index — unitless index of how far the logit calibration curve intercept and slope are from $(0, 1)$

Q**:** Logarithmic accuracy score — a scaled version of the log-likelihood achieved by the predictive model

Intercept: Calibration intercept on logit scale

Slope: Calibration slope (slope of predicted log odds vs. true log odds)

Measure of Accuracy of Validation Estimates Median absolute error, mean absolute error, and root mean squared error of estimates (e.g., of D_{xy} from the bootstrap on the 200 observations) against the "gold standard" (e.g., Dxy for the fitted 200-observation model achieved in the 50,000 observations).

Assignment 5 Assigned 2022-02-20 Due 2022-02-28

This is a group assignment. Groups are defined in the following table. One solution should be turned in per group, and the work should list all members who contributed meaningfully to the work. Groups must not help each other.

	Student	Group
1	Huiding Chen	1
2	Ruby Xiong	1
3	Jackson Resser	1
4	Siwei Zhang	2
5	Max Rohde	2
6	Sarah Torrence	2
7	Cara Tanaka Lwin	3
8	Marisa Hayli Blackman	3
9	Zoey Song	3
10	Justin Leon Jacobs	3

Do the problems at the end of Chapter 8 in the second edition. Consider `stage` as linear for `transcan` because its excessive ties prevent knot identification.

This is an individual project, counting significantly more than group assignments. Your interpretations will be key.

Some R programming hints follow.

```
# To subset a data frame and run varclus on all remaining variables:
require(Hmisc)
m ← subset(mydata, row.subsetting.expression,
           select=-c(x17,x19,x21))    # exclude 3 vars
plot(varclus(~., data=m))

# Function to compute first k PCs of a matrix of numeric variables
pc1 ← function(x, k=1) {
  g ← prcomp(x, scale=TRUE)
  g$x[, 1:k]
}

# Correlate PC1 with some transformed individual variables
vars ← trans[,c('x1','x2','x3')]
cor(pc1(vars), vars)

# Cumulative proportion of variance explained by PCs: see
# addscree function in text

# Subset to complete cases for numeric variables
w ← subset(m, !is.na(x1 + x2 + x3))
# Cs is in Hmisc - allows one to omit quote marks
# Create a matrix containing numeric variables in data frame
a ← as.matrix(w[,Cs(x1,x3,x7)])
# Augment the matrix with binary variable translations of
# some categorical variables
a ← with(w, cbind(a,
                  male=1*(sex == 'male'),
                  ...))
# Get another set of variables
b ← as.matrix(w[,Cs(x2,x4)])
```

```
# Compute correlation of two PC1s for two sets of variables
cor(pc1(a), pc1(b))

# Function to compute transcan transformations
ttrans ← function(x) {
  # asis= tells transcan to leave some variables untransformed
  z ← transcan(x, transformed=TRUE, data=w, pr=FALSE, pl=FALSE,
               asis=Cs(male,...))
  z$transformed
}

# Function to compute first nonlinear principal component
npc1 ← function(x) pc1(ttrans(x))
cor(npc1(a), npc1(b))

# Function to compute first canonical correlation across 2 matrices
cancor1 ← function(X, Y) cancor(X, Y)$cor[1]
cancor1(a, b)
cancor1(ttrans(a), ttrans(b))  # Can. corr. on transformed vars

# Cox PH model
S ← with(mydata, Surv(followup.time, binarystatusindicator))
f ← cph(S ~ ..., x=TRUE)
X ← f$x   # save numeric design matrix for later

# Compute first 3 PCs for whole dataset
pc3orig ← pc1(X, 3)

# Simulate to get bootstrap percentile CLs
sim ← function(B, type=1) {
  # type = 2 to recompute PCs
  ...
  inversions ← 0
  for(i in 1:B) {
    j ← set.of.subscripts.in.current.bootstrap.sample
    if(type == 2) {
      pc3 ← pc1(X[j,], 3)
      Sj ← S[j,]
      # PCs are not unique to within a sign inversion
      # Flip PCs to make them positively correlated with original PCs
      for(k in 1:3) {
        if(cor(pc3orig[j,k], pc3[,k]) < 0) {
          inversions ← inversions + 1
          pc3[,k] ← - pc3[,k]
        }
      }
      f ← cph(Sj ~ pc3)
    } else f ← cph(S ~ pc3, subset=j)
    pc1coef[i] ← coef(f)[1]
    orig.coef ← coef(lsfit(X, predict(f)))
    ...
  }
  if(inversions > 0) cat('inversions:', inversions, '\n')
  ... # compute 3 quartiles of 2 estimates
```

```
}
```

Assignment 6 Assigned 2022-03-01 Due 2022-03-14

Do the problems at the end of Chapter 9 in the second edition.

This is a group assignment. Groups are defined in the following table. One solution should be turned in per group, and the work should list all members who contributed meaningfully to the work. Groups must not help each other.

	Student	Group
1	Max Rohde	1
2	Marisa Hayli Blackman	1
3	Justin Leon Jacobs	1
4	Huiding Chen	2
5	Ruby Xiong	2
6	Zoey Song	2
7	Cara Tanaka Lwin	3
8	Siwei Zhang	3
9	Jackson Resser	3
10	Sarah Torrence	3

Some R code for the last problem is given below. See http://www.sumsar.net/blog/2015/07/easy-bayesian-bootstrap-in-r (reproduced at http://www.r-bloggers.com/easy-bayesian-bootstrap-in-r) for a nice introduction to the Bayesian bootstrap and its approximation, including some R code snippets below.

Extra credit will be given for any group that adds the bootstrap-t confidence interval to this simulation study.

```
require(rms)
require(ProfileLikelihood)
n ← 200    # sample size
m ← 1000   # no. simulations
B ← 1000   # no. bootstrap reps per simulation
n2 ← max(n, 1000)   # no. to sample with approx Bayes bootstrap
set.seed(13)

x1 ← exp(rnorm(n))
X ← cbind(x1, x1 ^ 2)
logit ← 1 + x1 / 2
P ← plogis(logit)
dd ← datadist(x1);  options(datadist='dd')
trueLOR ← (5 - 1) / 2
lims ← c('Lower 0.95', 'Upper 0.95')
meths ← c('Wald', 'Sandwich', 'Bootstrap Percentile',
          'Bootstrap BCa', 'Bootstrap Basic', 'Bayesian Bootstrap',
          'Approx. Bayesian Bootstrap', 'Profile Likelihood')
r ← array(NA, dim=c(m, 8, 2),
          dimnames=list(NULL, meths, c('Lower', 'Upper')))
estLOR ← numeric(m)
options(showprogress=FALSE)

bayesboot ← function(type=c('bayes', 'approx')) {
  fit ← function(subs=1 : n, weights=rep(1, n))
    tryCatch(lrm.fit(X[subs, ], y[subs], weights=weights[subs],
                     normwt=TRUE)$coefficients,
             error=function(...) {cat('could not fit\n'); c(NA,NA,NA)})
  type ← match.arg(type)
```

```
  lors ← numeric(B)
  for(j in 1 : B) {
    wts ← rexp(n); wts ← wts / sum(wts)
    cof ← if(type == 'bayes') fit(weights=wts)
    else fit(subs=sample(1 : n, size=n2, replace=TRUE, prob=wts))
    lors[j] ← 4 * cof[2] + 24 * cof[3]
  }
  quantile(lors, c(0.025, 0.975), na.rm=TRUE)
}

## Compute design matrix that reparameterizes the model so that the
## last coefficient is the estimate of the log OR for x1=5:1, for
## use with profile likelihood method
## Original: a + bx + cx^2.  Estimand: 24c + 4b = k; c = (k - 4b)/24
## a + bx + (k -4b)/24 x^2 = a + bx  + k/24 x^2 - b/6 x^2 =
## a + b(x - x^2/6) + k(x^2/24)
Xp    ← cbind(x1 - x1 * x1 / 6, x1 * x1 / 24)
for(i in 1:m) {
  cat(i, '\n', file='/tmp/progress.txt')
  y ← ifelse(runif(n) ≤ P, 1, 0)
  f ← lrm(y ~ pol(x1,2), x=TRUE, y=TRUE)
  s ← summary(f, x1=c(1,5))
  estLOR[i] ← s['x1', 'Effect']
  r[i, 'Wald',] ← s['x1', lims]
  rob ← robcov(f)
  r[i, 'Sandwich',] ← summary(rob, x1=c(1,5))['x1', lims]
  b ← bootcov(f, B=B)
  r[i, 'Bootstrap Percentile',] ← summary(b, x1=c(1,5))['x1', lims]
  r[i, 'Bootstrap BCa',] ←
    summary(b, x1=c(1,5), boot.type='bca')['x1', lims]
  r[i, 'Bootstrap Basic',] ←
    summary(b, x1=c(1,5), boot.type='basic')['x1', lims]
  r[i, 'Bayesian Bootstrap',]          ← bayesboot('bayes')
  r[i, 'Approx. Bayesian Bootstrap',] ← bayesboot('approx')
  pdata ← data.frame(y=y, x1=Xp[,1], lor51=Xp[,2])
  fg ← glm(y ~ Xp, family=binomial)
  u  ← profilelike.glm(y ~ x1, data=pdata,
         profile.theta='lor51', family=binomial(link="logit"),
         length=300, round=3)
  pl.ci ← profilelike.summary(k=8, theta=u$theta,
              profile.lik.norm=u$profile.lik.norm, round=3)$LI.norm
  r[i, 'Profile Likelihood',] ← pl.ci
  ## Note: the following will not work because the R built-in
  ## profile likelihood method has false convergence
  ## r[i, 'Profile Likelihood',]          ← confint(fg, 3)
  Save(r)
}
truecl ← quantile(estLOR, c(.025, .975))
cat('True 0.95 CL:', truecl[1], truecl[2], '\nTail coverages:\n')
mn ← function(x) mean(x, na.rm=TRUE)
res ← cbind(Nlower=NA, Nupper=NA, r[1,,], Overall=NA)
for(z in meths) res[z,] ← c(
  sum(! is.na(r[,z,'Lower'])),
  sum(! is.na(r[,z,'Upper'])),
```

```
  mn(r[,z,'Lower'] > trueLOR),
  mn(r[,z,'Upper'] < trueLOR),
  mn(trueLOR > r[,z,'Lower'] & trueLOR < r[,z,'Upper']) )
print(res)
saveRDS(res, 'res.rds')
```

Assignment 7 Assigned 2022-03-19 Due 2022-03-27

This is a group assignment. Groups are defined in the following table. One solution should be turned in per group, and the work should list all members who contributed meaningfully to the work. Groups must not help each other.

	Student	Group
1	Cara Tanaka Lwin	1
2	Siwei Zhang	1
3	Sarah Torrence	1
4	Marisa Hayli Blackman	2
5	Ruby Xiong	2
6	Zoey Song	2
7	Max Rohde	3
8	Huiding Chen	3
9	Jackson Resser	3
10	Justin Leon Jacobs	3

Do the problems at the end of Chapter 13 in the second edition. The last problem is for significant extra credit. For problem 4. also use n groups where n is the number of distinct cost values. The `rms orm` function efficiently handles large numbers of distinct Y values.

For the model with no grouping of cost, estimate the 0-1 scaled Wilcoxon statistic for comparing two groups (the concordance probability) from the estimated odds ratio and compare this to the empirical concordance probability after reading `fharrell.com/post/wpo`. For this part have only one binary predictor in the model: an indicator for whether the patient is in the largest disease group.

For the original disease group variable and continuous cost check the proportional odds assumption with an appropriate graphic.

Assignment 8 Assigned 2022-03-28 Due 2022-04-08

This is a group assignment. Groups are defined in the following table. One solution should be turned in per group, and the work should list all members who contributed meaningfully to the work. Groups must not help each other.

	Student	Group
1	Marisa Hayli Blackman	1
2	Ruby Xiong	1
3	Justin Leon Jacobs	1
4	Cara Tanaka Lwin	2
5	Max Rohde	2
6	Sarah Torrence	2
7	Siwei Zhang	3
8	Huiding Chen	3
9	Zoey Song	3
10	Jackson Resser	3

Work the problems at the end Chapter 14 in the second edition. For problem 3, also assess PO by comparing a PO model with two non-PO models by mimicking what was done using the `impactPO` function in Section 13.3.5 of the course notes. The two non-PO models to consider are a full multinomial model fitted with the `multinom` function in the `nnet` package (which may already be installed in your R system) and a partial PO model fitted using the `vglm` function in the `VGAM` package which you will need to install. The calculations using these packages are done automatically with the `impactPO` function as demonstrated at `https://fharrell.com/post/impactpo` and in the course notes. You can load the source code for `impactPO` (until it is added to `rms` on CRAN) using `source('https://raw.githubusercontent.com/harrelfe/rms/master/R/impactPO.r')`. Interpret the results of the "predicted vs. observed" stacked bar charts, and interpret the results of a bootstrap analysis comparing PO and non-PO models. Compare these assessments with the graphical assessments that required binning continuous variables.

For problem 6. also run a Bayesian proportional odds model using default priors, using the `rmsb` package `blrm` function. Look at the uncertainty interval for one measure of predictive performance and interpret the interval, describing why it is not a measure of future model performance. Instead of doing any statistical tests, compute the posterior probability of an assertion of your choosing. Be sure to run MCMC diagnostics. See the new `blrm` example at the end of Chapter 10 notes. Check the `blrm` help file for how to use the `file=` argument to not have to re-run `blrm` when its inputs don't change. For extra credit fit a constrained partial PO model, relaxing only the PO assumption for the one baseline measure as you did with `impactPO`. Assess evidence for non-PO from the Bayesian posterior distribution for the non-PO (τ) parameter.

In addition do problem 10:

10. Attempt to derive the analytic solution of the MLE of β in a two-group proportional odds model without covariates. Use this notation: response variable on n observations is $Y_1, Y_2, \ldots, Y_n$, distinct values of Y are $y_1, y_2, \ldots, y_k$, first n_a observations are from group A and second n_b observations are on group B. You can use groups 1 and 2 instead of A and B if you prefer. If you are unable to derive an analytic solution, derive a one-step estimator (one iteration of the Newton-Raphson algorithm or simple Taylor series approximation). For Newton-Raphson note that the MLE of the intercepts when the first slope parameter is zero are very simple—this is what is used to start the iterations in the `lrm` and `orm` functions.

Check your analytic result against the following dataset.

```
require(rms)
```

```
w ← rbind(data.frame(x='a', y=c(0,3,5,5,10,11)),
          data.frame(x='b', y=c(1,3,5,16,17,20)))
coef(orm(y ~ x, data=w, eps=0.00001))
```

```
      y>=1       y>=3       y>=5      y>=10      y>=11      y>=16
y>=17
 1.9690146  1.1780923  0.2884028 -0.7347160 -1.1056198 -1.5630033 -2.1454948
     y>=20         x=b
-2.9889933  1.0199787
```

Derive an empirical solution to the problem by simulating a variety of sample sizes and integer-valued y and binary x. Use the c-index translation of the Wilcoxon statistic to predict $\hat{\beta}$, so as to be independent of sample size. You will be comparing c to $\hat{\beta}$ across simulations, possibly making a sensible transformation of c. To compute c you can do something like `somers2(y,x)['C']`. Use a variation of what is at `https://www.fharrell.com/post/po` and see also `https://www.fharrell.com/post/wpo`. To avoid outliers ruining the estimated regression coefficients, use robust regression e.g. `MASS::rlm`.

Assignment 9 Assigned 2022-04-09 Due 2022-04-17

This is a group assignment. Groups are defined in the following table. One solution should be turned in per group, and the work should list all members who contributed meaningfully to the work. Groups must not help each other except for the last problem.

	Student	Group
1	Huiding Chen	1
2	Max Rohde	1
3	Sarah Torrence	1
4	Ruby Xiong	2
5	Siwei Zhang	2
6	Justin Leon Jacobs	2
7	Marisa Hayli Blackman	3
8	Cara Tanaka Lwin	3
9	Zoey Song	3
10	Jackson Resser	3

1. Fit a binary logistic model to predict hospital death in the `support2` dataset. Use the union of predictors that were used in other assignments in the text, modeling them flexibly. For missing continuous variables, fill in their values using "most normal" values as done in the text. For categorical missing values use the mode.

2. Set up for re-simulation. Save the coefficients from the fitted model and pretend they are population coefficients. Get the predicted probability of hospital death for each patient. For the simulations to follow, simulate Bernoulli random variables having true probability that $Y = 1$ given by the probabilities of hospital death your just estimated.

3. Run an adequate number of Monte Carlo simulations to study properties of the model derivation methods below. For each simulation measure the predictive discrimination of the full model fit using c, Brier score, and generalized R^2, and the accuracy of the model by estimating the mean absolute difference and mean squared difference between predicted logit and population (true) logit. Decide upon a model approximation strategy that yields approximate models that are 0.95 as good as the full model. But start by checking your code by taking as the approximate model a least squares re-fit of the full model to the original linear predictor. Then repeat after removing the single least important predictor. If either of these two analyses suffer on root mean squared error, something is amiss. Then continue with a sequence of rougher approximations down to 0.95 approximation accuracy. Run Monte Carlo simulation to study the performance of all of these models. Compute the same statistical measures for the approximate model as you ran for the full model. If you see a dropoff in performance of predicting true $X\beta$ from an approximate model that seemed to well-predict the full model's $X\hat{\beta}$ look into the code and the approximation algorithm and perform any needed additional analyses to explain this dropoff.

4. Summarize the various indexes over all simulations.

5. Summarize the volatility of the model approximation step by depicting the variation in the entire list of variables selected by the approximation.

6. Consider the `sex-age-response` example at the beginning of the binary logistic regression chapter. Compute 0.95 confidence limits for the age and sex coefficients, and the estimated correlation between these two estimates. Use the R `rmsb` or `brms` package to do a Bayesian binary logistic analysis that differs from the one in the course notes. Use some sort of diffuse prior for the intercept, and slightly more constrained Gaussian priors for the age and sex parameters. Display posterior densities for the age and sex parameters, and the bivariate posterior distribution of the two. Compute the linear correlation coefficient between the two parameters over the posterior draws. Compare this to the frequentist sampling-based parameter estimate correlation coefficient. Compute the posterior probability that

either sex has a positive relationship with Y or age has a positive relationship. Compute the posterior probability that both have a positive relationship. Use `Zulip` to discuss this problem, allowing each group to help each other as well as the instructor and teaching assistant to help as you go.

Assignment 10 Assigned 2022-04-17 Due 2022-04-21

This is a group assignment. Groups are defined in the following table. No written work is to be turned in. Groups should project rough drawings or mock-ups of analyses or results on the screen. Don't spend any time making figures, charts, flowcharts, or tables look good. This is meant to be informal. Post any material to project as attachments in a `Zulip` message under the `Homework General` topic.

Each group is to create a non-trivial made-up data analysis problem that does not come from any of their final projects, with a specification of actual and effective sample size, optional use of unsupervised learning in addition to supervised learning, and possible solutions including model specification. The groups will present their problems and sketched solutions on the last day of class. One student from each group should be appointed to start the presentation but each student in each group must also present. Groups may want to have different group members present different aspects of the problem/solutions.

Each group will have 20 minutes. Different points of view within a group are encouraged.

Brainstorming about the mock data analysis problem can take place any time on the above topic on `Zulip` if you want to get suggestions from other groups, TAs, or instructor.

	Student	Group
1	Huiding Chen	1
2	Marisa Hayli Blackman	1
3	Zoey Song	1
4	Max Rohde	2
5	Siwei Zhang	2
6	Justin Leon Jacobs	2
7	Cara Tanaka Lwin	3
8	Ruby Xiong	3
9	Sarah Torrence	3
10	Jackson Resser	3