Different models for different purposes: focused model comparison in R

Chris Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

Abstract

Typical methods of model comparison pick one "best" model, no matter what the estimates from the model are used for. "Focused" model comparison, by contrast, considers that different models may be better for different purposes. Different models may be preferred for estimating different "focus" quantities, functions of the basic parameters.

In the "focused information criterion" of Claeskens and Hjort (2006), data are assumed to be generated by a "wide" model, in which all models we would consider are nested. Fitting the wide model to the observed data, however, may give estimates that are not sufficiently precise. Therefore we might accept some bias in the estimate in return for greater precision. The optimal submodel for a particular focus is the one which minimises the mean squared error of the estimate of that focus from the submodel, assuming that the wide model is true. *[maybe other losses if can get bootstrap to work]*

The `fic` package calculates this error, and related quantities, straightforwardly for any class of models fitted by maximum likelihood. There are shortcuts for commonly-used model classes such as GLMs and parametric survival models. Cox regression models are also supported *[todo]*

Keywords: FIC,model comparison,AIC,BIC.

1. Introduction: principles for model comparison

To compare a set of statistical models fitted to the same data by maximum likelihood, it is common to rank them according to some "criterion". For example, Akaike's information criterion (AIC, Akaike (1973)) takes the form

$$-2\log \ell(\hat{\theta};\mathbf{x}) + 2p$$

where $\ell(\hat{\theta};\mathbf{x})$ is the maximised likelihood for the model fitted to the dataset $\mathbf{x}$, the likelihood is maximised at parameters $\hat{\theta}$, and p is the number of parameters.

The Bayesian information criterion (BIC, Schwarz (1978)) is

$$-2\log \ell(\hat{\theta};\mathbf{x}) + p\log(n)$$

These two criteria are based on very different principles. Thus they often rank models differently. The AIC is designed to choose the model with the best predictive ability, thus it tends to favour bigger models as the sample size increases. BIC is an approximation to Bayesian model comparison by Bayes factors, and selects the model with the highest posterior probability under an implicit weak prior (with an amount of information equivalent to one observation, see Kass and Wasserman (1995)). If there is a "true" model, the BIC will select it "consistently" as the sample size increases. In many situations there may not be a true model, and collecting more data will uncover more complexity in the process generating the data, in which case AIC may be more suitable. See e.g. Burnham and Anderson (2003), Claeskens and Hjort (2008) for more theory behind these criteria.

However both of these methods select one "best fitting" model for a given dataset. But that might not always be appropriate. Different models may be better for different purposes. This is the idea behind "focused" model comparison.

[describe what package does, structure of paper]

2. Focused model comparison: principles and formulae

Suppose the range of models we are willing to use is bounded by

- a *wide model*, in which all models we would use are nested, with parameters $(\boldsymbol{\theta}, \boldsymbol{\gamma})$
- a *narrow model*, the smallest model we are willing to use, defined by setting $\boldsymbol{\gamma} = \boldsymbol{\gamma}_0$ in the wide model.

`[list of examples. covariate selection obvious, but lots more too]`

Suppose also that the purpose of the model is to estimate some *focus* quantity, which could be any function of the basic parameters

$$\mu = g(\boldsymbol{\theta}, \boldsymbol{\gamma})$$

In focused model comparison, we prefer models which give better estimates of μ. A typical way to define "better" is by the *mean square error*. The mean square error of the estimate $\hat{\mu}_S$ under a submodel S of the wide model, compared to the true value μ, is

$$E\left\{(\hat{\mu}_S - \mu)^2\right\}$$

This expectation is calculated under the assumption that the data are generated from the wide model. While we believe the wide model is the most realistic, we also accept that there may not be enough data to give sufficiently precise estimates of μ. Therefore we are willing to accept some bias in this estimate, in return for a smaller variance, by selecting a smaller model than the wide model. The submodel S with the lowest mean square error is the one which makes the optimal trade-off between bias and variance.

The mean squared error MSE_S under model S can be decomposed as a sum of the squared bias B_S^2 and the variance V_S.

$$\begin{aligned} MSE_S = E\left\{(\hat{\mu}_S - \mu)^2\right\} &= \left\{E(\hat{\mu}_S) - \mu\right\}^2 + E\left\{(\hat{\mu}_S - E(\hat{\mu}_S))^2\right\} \\ &= B_S^2 + V_S \end{aligned} \tag{1}$$

Estimators for these quantities are constructed by Claeskens and Hjort (2003) under an asymptotic framework in which the data are assumed to be n independent observations generated from the wide model, but reparameterised so that $\boldsymbol{\gamma} = \gamma_0 + \boldsymbol{\delta}/\sqrt{n}$. Thus as the sample size increases, we aim to detect more subtle departures from the narrow model.

[any more basic assumptions?]

An obvious estimator for the bias B_S is $\hat{B_S} = \hat{\mu_S} - \hat{\mu_W}$, where $\hat{\mu}_W$ is the estimate of the focus quantity under the wide model, which is assumed to be unbiased. However, Claeskens and Hjort (2003) derive a more accurate estimator for the *squared* bias as

$$\widehat{B^2_S} = (\hat{\psi}_W - \hat{\psi}_S)^2/n$$

where

- $\hat{\psi}_W = \hat{\omega}^T\hat{\boldsymbol{\delta}}$ and $\hat{\psi}_S = \hat{\omega}^T G_S \hat{\boldsymbol{\delta}}$ are estimates of $\omega^T\boldsymbol{\delta}$ under the wide model and submodel respectively, where $\omega^T\boldsymbol{\delta}$ is the bias of the estimate of $\sqrt{n}\mu$ under the narrow model N, that is, the asymptotic mean of $\sqrt{n}(\hat{\mu}_N - \mu)$. Thus ω acts as a linear transformation from the biases of the basic parameters $\boldsymbol{\gamma}$ to the biases of the focus parameter μ. *[why not let all the ns cancel, and explain everything in terms of biases on the focus scale?]*
- ω is estimated as $\hat{\omega} = J_{10}J_{00}^{-1}\frac{d\mu}{d\theta} - \frac{d\mu}{d\gamma}$ using Taylor approximation arguments, where J is the information (inverse covariance) matrix under the wide model divided by n, *[but the ns cancel, so could we work in terms of the info under the wide model?]* and subscripts 0 and 1 select the rows and columns forming the submatrices of J that correspond to parameters $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ respectively. The partial derivatives of the focus μ are evaluated at the estimates from the wide model.
- $G_S = \pi^T Q_S \pi Q^{-1}$ is an estimate of the transformation that maps the wide model estimate of $\boldsymbol{\delta}$ to the submodel S estimate, where $Q_S = (\pi Q^{-1}\pi^T)^{-1}$, $Q^{-1} = J_{11}$ and π is the projection matrix consisting of 0s and 1s which maps a vector of the same length as $(\boldsymbol{\theta}, \boldsymbol{\gamma})$ to a subvector containing the elements corresponding to submodel S. *[again don't the ns in the Qs cancel]*
- $\hat{\boldsymbol{\delta}} = \hat{\boldsymbol{\gamma}}\sqrt{n}$, where $\hat{\boldsymbol{\gamma}}$ is the estimate of $\boldsymbol{\gamma}$ under the wide model.

The estimator for the variance of $\hat{\mu}_S$ under the wide model, derived by Claeskens and Hjort (2003), is

$$\hat{V} = (\hat{\tau_0}^2 + \hat{\omega}^T Q^0_S \hat{\omega})/n$$

where $\hat{\tau_0}^2/n$ estimates the variance of the narrow model focus estimate (using "delta method" principles, $\hat{\tau}_0^2 = \frac{d\mu}{d\theta}^T J_{00}^{-1}\frac{d\mu}{d\theta}$), and the additional term $(\hat{\omega}^T Q^0_S \hat{\omega})/n$ is the increase in variance we accept by using a wider but still misspecified model S, with $Q^0_S = \pi^T Q_S \pi$.

[again if we define J as info instead of info/n, then we wouldn't need to supply n]

Thus we compare models on the basis of the root mean square error, estimated by

$$\sqrt{\widehat{MSE}_S} = \sqrt{\widehat{B^2_S} + \hat{V}_S} \tag{2}$$

Claeskens and Hjort (2003) define the "focused information criterion" (FIC), which has a slightly simpler form due to excluding terms common to all submodels S, and is related to the MSE as

$$FIC_S = nMSE_S - \hat{\tau}_0^2 + \hat{\omega}^T Q \hat{\omega} \tag{3}$$

Models with lower FIC give better estimates of the focus quantity. However we prefer to use the (root) MSE as the model comparison statistic, due to its direct interpretation as the error of the focus estimate. *FIC would need knowledge of n*

2.1. Bias-corrected MSE

Claeskens and Hjort (2003) derive a further correction for the bias estimator which is necessary when the above estimate is negative. The adjusted squared bias estimator is

$$\widehat{B^{*2}} = max\left\{0, \quad \hat{\omega}^T(I - G_S)(\hat{\boldsymbol{\delta}}\hat{\boldsymbol{\delta}}^T - Q)(I - G_S)^T\hat{\omega}/n\right\} \tag{4}$$

The corresponding estimate of the bias B is $sign(\hat{\psi}_W - \hat{\psi}_S)\sqrt{\widehat{B^{*2}}}$, and the bias-corrected root MSE is $\sqrt{\widehat{MSE}} = \sqrt{\widehat{B^{*2}} + \hat{V}}$.

2.2. Average MSE over a range of focuses

Often we want a model that performs well in a range of situations. In covariate selection problems, for example, we might want to estimate a focus quantity accurately for a defined range of covariate values. We might simply define the "averaged MSE"

$$AMSE = \int MSE(u)dW(u)du$$

as a weighted average of the mean squared errors (1) for focuses defined by different covariate values u, weighted by their prevalence $W(u)$. However Claeskens and Hjort (2008) derived an alternative formula, so that if bias correction analogous to (4) is required, it only needs to be performed once.

$$AMSE = max(IS, 0) + IIS \tag{5}$$

where

$$\begin{aligned} IS &= Tr((I - G_S)(\hat{\boldsymbol{\delta}}\hat{\boldsymbol{\delta}}^T - Q)(I - G_S)^T A), \quad IIS = Tr(Q_S^0 A) \\ A &= J_{10}J_{00}^{-1}B_{00}J_{00}^{-1}J_{01} - J_{10}J_{00}^{-1}B_{01} - B_{10}J_{00}^{-1}J_{01} + B_{11} \\ B &= \int \begin{pmatrix} d\mu(u)/d\boldsymbol{\theta} \\ d\mu(u)/d\boldsymbol{\gamma} \end{pmatrix} \begin{pmatrix} d\mu(u)/d\boldsymbol{\theta} \\ d\mu(u)/d\boldsymbol{\gamma} \end{pmatrix}^T dW(u) \quad = \quad \begin{pmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{pmatrix} \end{aligned}$$

This is equivalent to AIC [where we average over all covariates in the data ?]

[What about equivalence to AIC where the focus is the log-likelihood?]

3. Software for focused model comparison

In order to calculate the MSE and related focused model comparison statistics for a submodel S, we just need to know

- the estimates $\hat{\boldsymbol{\theta}}_W$ and $\hat{\boldsymbol{\gamma}}_W$ under the wide model,
- the information matrix J or covariance matrix of these estimates,

- the focus function $\mu(\boldsymbol{\theta}, \boldsymbol{\gamma})$ and its derivatives, evaluated at $\hat{\boldsymbol{\theta}}_W, \hat{\boldsymbol{\gamma}}_W$,
- the definition of which parameters are included in submodel S and which are included in the narrow model N.
- *[doubtful we also need to know n]*

This allows the **fic** package to implement focused model selection for any class of models and focuses. Derivatives of the focus function can be calculated numerically in general, for which robust software exists — **numDeriv** (Gilbert and Varadhan 2016) is used here. Analytic derivatives are implemented in **fic** for two built-in focuses (the outcome probability in logistic regression, and the mean outcome in linear regression), but we have noticed no loss in accuracy from using numerical methods.

4. Example: covariate selection in logistic regression

The use of the `fic` package is illustrated for covariate selection in logistic regression, using the example originally presented in Claeskens and Hjort (2008) (Example 6.1). The dataset was originally presented by Hosmer and Lemeshow (1989). Data are taken from $n = 189$ women with newborn babies, and the binary outcome is whether the baby is born with a weight less than 2500g. We build a logistic regression model to predict the outcome, but are uncertain about what covariates should be included.

The data are provided as an object `birthwt` in the `fic` package. This is the same as `birthwt` in `MASS` (Venables and Ripley 2002) with the addition of a few extra columns defining interactions and transformations as in Claeskens and Hjort (2008).

The following covariates are always included (coefficient vector $\boldsymbol{\theta}$)

- x_1 Weight of mother in kg, `lwtkg`

The following covariates will be selected from (coefficient vector $\boldsymbol{\gamma}$)

- z_1 age, in years, `age`
- z_2 indicator for smoking, `smoke`
- z_3 history of hypertension, `ht`
- z_4 uterine irritability, `ui`
- interaction $z_5 = z_1 z_2$ between smoking and age, `smokeage`
- interaction $z_6 = z_2 z_4$ between smoking and uterine irritability, `smokeui`

Firstly the wide model, that includes all the above covariates, is defined and fitted.

```
library(fic)
wide.glm <- glm(low ~ lwtkg + age + smoke + ht + ui + smokeage + smokeui,
                data=birthwt, family=binomial)
```

The *focus function* is then defined. This should be an R function, mapping the parameters `par` of the wide model to the quantity of interest. The focus can optionally have an second argument. If supplied, this must be called `X`, and can be used to supply covariate values at which the focus function should be evaluated. Here we take the probability of low birth weight as the focus, for two covariate categories:

1. smokers with average or typical values of the other covariates. These values are given in the order supplied when specifying the model (for smokers: intercept, `lwtkg`=58.24, `age`=22.95, `smoke`=1, `ht`=0, `ui`=0, `smokeage`=22.95, `smokeui`=0).
2. non-smokers with average values of the other covariates

```
focus <- function(par, X)plogis(X %*% par)
vals.smoke <-    c(1, 58.24, 22.95, 1, 0, 0, 22.95, 0)
vals.nonsmoke <- c(1, 59.50, 23.43, 0, 0, 0, 0, 0)
X <- rbind("Smokers"=vals.smoke, "Non-smokers"=vals.nonsmoke)
```

We can illustrate these functions by calculating the probability of low birth weight, given the parameters of the fitted wide model, for each group. This is about twice as high for smokers.

```
focus(coef(wide.glm), X=X)

##              [,1]
## Smokers     0.345
## Non-smokers 0.168
```

The `fic` function can then be used to calculate the mean square error of the focus for one or more given submodels. For illustration we will compare two models, both including maternal weight, one including age and smoking, but the other including age, smoking and hypertension.

```
mod1.glm <- glm(low ~ lwtkg + age + smoke, data=birthwt, family=binomial)
mod2.glm <- glm(low ~ lwtkg + age + smoke + ht, data=birthwt, family=binomial)
```

We supply the following arguments to the `fic` function.

- `wide`: the fitted wide model. All the model fit statistics are computed using the estimates and covariance matrix from this model. `fic` will automatically recognise that this is a GLM fitted by the `glm` function in R, and extract the relevant information.
- `inds`: indicators for which parameters are included in the submodel, that is, which elements of $(\boldsymbol{\theta}, \boldsymbol{\gamma})$ are fixed to γ_0. This should have number of rows equal to the number of submodels to be assessed, and number of columns equal to $dim(\boldsymbol{\theta}) + dim(\boldsymbol{\gamma})$, the total number of parameters in the wide model, 8 in the case of `wide.glm`, which includes the intercept and the coefficients of seven covariates. It contains 1s in the positions where the parameter is included in the submodel, and 0s in positions where

the parameter is excluded. This should always be 1 in the positions defining the narrow model, as specified in `inds0` below. If just one submodel is to be assessed, `inds` can also be supplied as a vector of length $dim(\boldsymbol{\theta}) + dim(\boldsymbol{\gamma})$.

Note that `inds` indexes *parameters* rather than *linear model terms*, that is, in covariate selection problems where a variable is a factor with more than two levels, `inds` should contain separate entries for the coefficient of each factor level relative to the baseline level, not just one entry indicating the presence of the factor as a whole. A utility to construct this in the presence of factors is illustrated in Section 6.1.

- `inds0` vector of indicators for which parameters are included in the narrow model, in the same format as `inds`. This can be omitted, in which case the narrow model is assumed to be given by the first row of `inds`. In this case, just the first two parameters are included, the intercept and the coefficient of `lwtkg`.

```
inds <- rbind(mod1 = c(1,1,1,1,0,0,0,0),
              mod2 = c(1,1,1,1,1,0,0,0))
inds0 <- c(1,1,0,0,0,0,0,0)
```

- `focus` the focus function. As well as an R function, this argument can alternatively be supplied as a character string naming a built-in focus function supplied by the **fic** package. Currently these just include `"prob_logistic■`, the outcome probability in a logistic regression, and `"mean_normal■`, the mean outcome in a normal linear regression.

The main `fic` function then returns an object containing the model fit statistics and the estimate of the focus quantity for each model.

```
fic1 <- fic(wide=wide.glm, inds=inds, inds0=inds0, focus=focus, X=X)
fic1

##           vals mods   FIC   rmse rmse.adj   bias bias.adj
## 1      Smokers mod1 1.187 0.0723   0.0723 0.0548   0.0459
## 4      Smokers mod2 0.783 0.0556   0.0572 0.0237   0.0000
## 2  Non-smokers mod1 1.305 0.0804   0.0804 0.0765   0.0731
## 5  Non-smokers mod2 0.755 0.0596   0.0596 0.0525   0.0484
## 3          ave mod1 1.246 0.0844   0.0764 0.0657   0.0610
## 6          ave mod2 0.769 0.0678   0.0576 0.0381   0.0329
##        se focus
## 1 0.0558 0.398
## 4 0.0572 0.366
## 2 0.0334 0.243
## 5 0.0348 0.215
## 3 0.0460 0.320
## 6 0.0473 0.291
```

The object returned by `fic` is a matrix containing one row for each combination of focus covariate values indicated in the column `vals` and submodels indicated in the column `mods`. The focus estimate is returned in the final column `focus`, while the remaining columns contain the following model comparison statistics:

- `FIC` The FIC as originally defined by **??** (equation 3),
- `rmse` The root mean square error of the submodel focus estimate, calculated assuming the wide model is true (equation 2),
- `rmse.adj` The bias-adjusted root mean square error (Section 2.1),
- `bias` The estimated bias $\sqrt{\widehat{B^2}}$ (which may be undefined if $\widehat{B^2}$ is negative),
- `bias.adj` The adjusted bias estimate (Section 2.1),
- `se` The standard error $\sqrt{\hat{V}}$ of the submodel focus estimate, calculated assuming the wide model is true.

The submodels are fitted automatically within the `fic` function in order to produce the focus estimate, so it was not really necessary to fit `mod1.glm` and `mod2.glm` by hand, as above. As the wide model has class `glm`, it is recognised as a GLM, so `fic` assumes that our submodels correspond to models with different covariates included, as indicated by `inds`. The focus estimates from the submodels can then be returned alongside the model comparison statistics.

As well as the specific covariate categories, `fic` calculates model comparison statistics which are averaged over the categories, indicated by a value of `ave` in the column `vals`. TODO EXPLAIN WEIGHTS

Recall that `mod2` contains one more covariate than `mod1`. For each of the two focuses, and the average, the unadjusted and adjusted bias estimates are lower due to the inclusion of this covariate, while the standard error `se` is higher. Given the lower `rmse` and `rmse.adj` under `mod2`, the reduction in bias is deemed to be worth the increase in uncertainty.

4.1. Comparing a wide range of models

In covariate selection problems, we may want to examine a broad range of models. The function `all_inds` (a wrapper around `expand.grid`) creates a matrix of indicators that defines all submodels spanned by a given wide model (here `wide.glm` and a narrow model (here defined by `inds0`). This function works for all classes of model objects `x` for which the `terms(x)` function is understood, which includes standard R regression models such as `lm` and `glm`. *[can we use terms.formula more generally, e.g. in flexsurvreg?]* Factors are handled naturally.

```
combs <- all_inds(wide.glm, inds0)
```

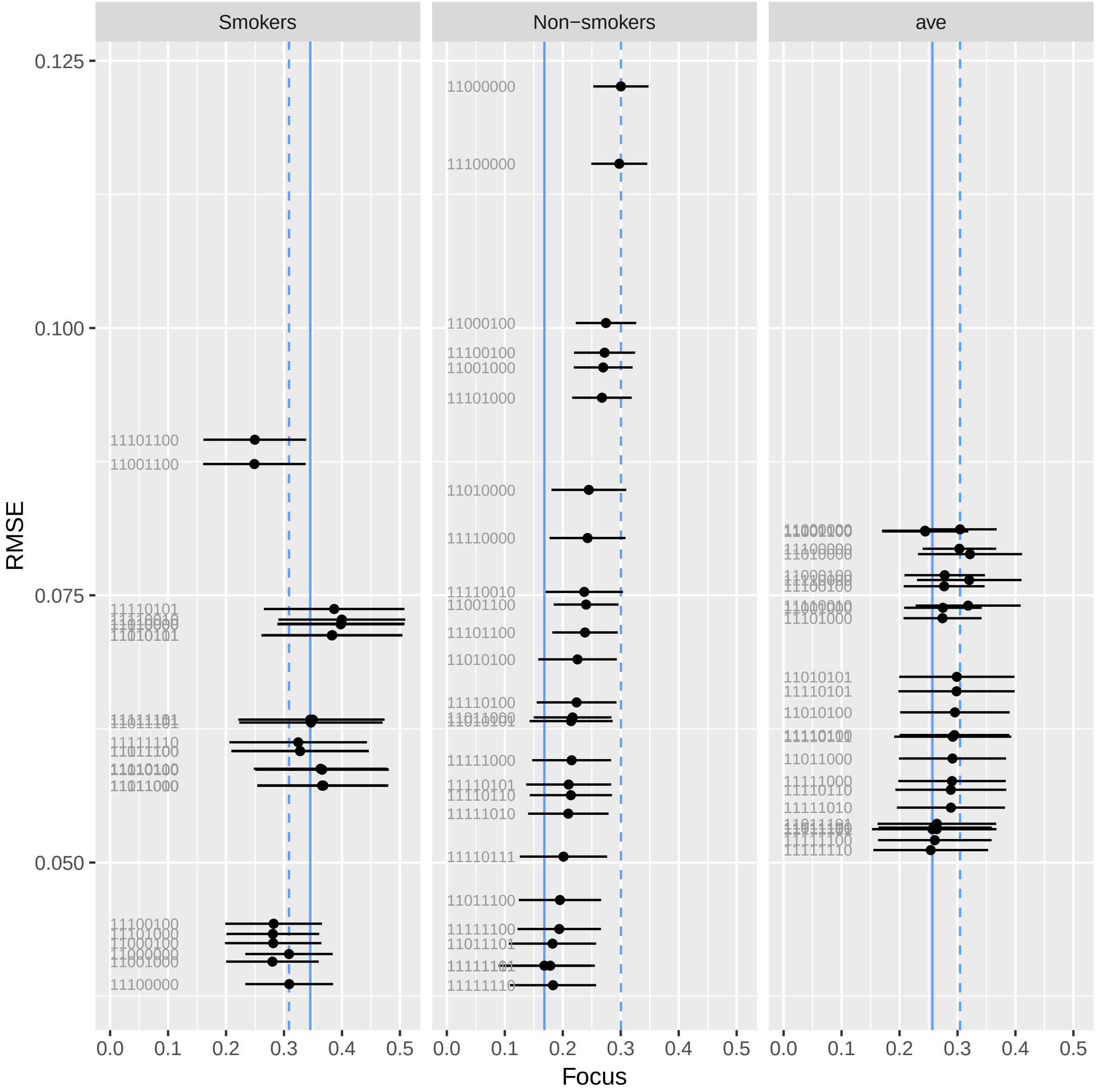
The resulting matrix can be used as the `inds` argument to `fic` to calculate focused model comparison statistics for all submodels in this example, again for a focus defined by the probability of low birth weight at covariate values defined by `X`. However before calling `fic` again, we redefine `combs` to exclude models with interactions but not both corresponding main effects.

```
combs <- with(combs,
              combs[!((smoke==0 & smokeage==1) |
                      (smoke==0 & smokeui==1) |
                      (age==0 & smokeage==1) |
                      (ui==0 & smokeui==1)),])
ficres <- fic(wide=wide.glm, inds=combs, inds0=inds0, focus=focus, X=X)
```

Notice that some of the `rmse` elements of `ficres` are `NaN`, since the first squared bias estimator $\widehat{B^2}$ is negative. The alternative estimate $\sqrt{\widehat{B^{*2}}}$, `rmse.adj`, can simply be used in these cases. *[this raises the question: why we don't just use the adjusted one all the time. Is the unadjusted one better in any circumstances?]*

A comparison of many models can be illustrated by a scatterplot of the focus estimate against the RMSE of each submodel. The default `plot` method for `fic` objects accomplishes this using base R graphics: try `plot(ficres)`. Alternatively a graph can be plotted using `ggplot2` if this package is installed. This is illustrated here.

```
ggplot_fic(ficres)
```



There is one panel for each of the two covariate categories (smokers and non-smokers) defining the focus (probability of low birth weight) and an average over the two categories. The solid blue line is the focus estimate under the wide model, and the dashed blue line is the focus estimate under the narrow model. An informal illustration of the uncertainty around the estimate of the focus quantity from each submodel is given by the estimate $\pm 1.96 \times \sqrt{\hat{V}}$.

Each submodel is labelled faintly using the row names of the matrix supplied as the `inds` argument to `fic`. In this case, these names were automatically constructed by the function `all_inds` and contain a string of binary 0/1 indicators for the inclusion of eight parameters. For smokers, the narrow model (labelled `11000000`) and similar smaller models give estimates of the probability of low birth weight with the lowest MSE, while by contrast, for non-smokers, the wide model (labelled `11111111`) and similar larger models give the most accurate estimates of the focus quantity. Note that in this dataset, there are 115 non-smokers and 74 smokers, thus more data enables bigger models to be identified for non-smokers. *[discuss average?]*

[Write some warning against fishing. If fitting a large number of models, uncertainty should be acknowledged. ref discussion about post-selection inference]

5. Other classes of models

Illustrate new concepts. Could be in separate vignettes.

Linear Illustrates alternative focuses: expected outcome at given covariate values, quantile for given covariate values. Polynomial order selection as well as covariate selection. Use a well known dataset, e.g. mtcars?

Multi-state Illustrates focuses that are complicated functions of the model parameters: package should facilitate this

Survival Increasingly flexible parametric models. Challenge to express models as nested within each other. Important health economic application: restricted mean survival.

Skew-normal Novel class of models, user-written model fitting function. Store est, vcov, nobs in result list. Narrow model parameters could be in the middle.

5.1. Calling "fic" for an unfamiliar class of models

Above, the `fic` function recognised the fitted model objects as GLMs, that is, objects of class `"glm"` returned by the `glm()` function in base R. But the package can be used to calculate focused model comparison statistics for any class of models, not just the special classes it recognises. To do this, it needs to know where three things are stored inside the fitted model objects:

1. `coef`: the vector of maximum likelihood estimates $(\hat{\boldsymbol{\theta}}, \hat{\gamma})$,

2. `nobs`: the number of observations n contributing to the model fit,

3. `vcov`: the covariance matrix of the maximum likelihood estimates, $(nJ)^{-1}$. *[not sure this is needed in theory unless want FIC on $\sqrt{n}$ scale]*

Given a fitted model object called `mod`, the `fic()` function assumes by default that `coef(mod)`, `nobs(mod)` and `vcov(mod)` respectively return these pieces of information. If one or more of these assumptions is not true, the defaults can be changed by supplying the argument `fns` to `fic()`, which should be a named list of three components. Each component should be a function with one argument (the fitted model) which extracts the required information from the fitted model and returns it. For example, the first component of the list below is a function which, when applied to a `glm` object, returns the maximum likelihood estimates of the regression coefficients. (TODO better explained with a more obscure class?)

```
fns <- list(coef = function(x)coef(x),
            nobs = function(x)nobs(x),
            vcov = function(x)vcov(x))
fic1 <- fic(wide=wide.glm, inds=inds, inds0=inds0, focus=focus, fns=fns,
            X=X, sub=sub)
```

6. Focused covariate selection in Cox proportional hazards regression

In a Cox regression model, time-to-event outcomes t_i are observed on individuals i, potentially with right-censoring. At time t, individual i is assumed to have a hazard $h_i(t)$ which is proportional to their covariate values. We wish to select between models that have different sets of covariates. In the most general "wide" model, $h_i(t) = h_0(t)\exp(\boldsymbol{\theta}^T x_i + \boldsymbol{\gamma}^T z_i)$. The baseline hazard $h_0(t)$ is left unspecified, while $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ are estimated by maximum partial likelihood.

We compare submodels S of this wide model, which include different subsets of covariates, according to how accurately they estimate some focus quantity $\mu = \mu(\boldsymbol{\theta}, \boldsymbol{\gamma}, H_0()|\mathbf{x}, t)$, where $H_0()$ is the cumulative baseline hazard function, which can be estimated nonparametrically by various methods. Typical focus quantities might depend on time t as well as covariate values $\mathbf{x}$, e.g. the probability $S(t|\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\gamma}, H_0())$ that a person with covariates $\mathbf{x}$ will survive t years.

Again the mean square error $MSE_S = B_S^2 + V_S$ of μ of the focus quantity under submodel S is estimated as $\widehat{B_S^2} + \widehat{V_S}$, using similar formulae to Section 2, which were derived by Hjort and Claeskens (2006) under the same theoretical principles:

$$\begin{aligned}
\widehat{B_S^2} &= (\hat{\psi}_W - \hat{\psi}_S)^2/n, \quad \hat{\psi}_W = (\hat{\omega} - \hat{\kappa})^T \hat{\boldsymbol{\delta}}, \quad \hat{\psi}_S = (\hat{\omega} - \hat{\kappa})^T G_S \hat{\boldsymbol{\delta}} \\
\widehat{V_S} &= \left\{\hat{\tau}_0^2 + (\hat{\omega} - \hat{\kappa})^T Q_S^0 (\hat{\omega} - \hat{\kappa})\right\}/n \\
\hat{\omega} &= J_{10} J_{00}^{-1} \frac{d\mu}{d\boldsymbol{\theta}} - \frac{d\mu}{d\boldsymbol{\gamma}} \\
\hat{\kappa}(t) &= (J_{10} J_{00}^{-1} F_0(t) - F_1(t)) \frac{d\mu}{dH_0},
\end{aligned}$$

where $Q_S^0, G_S, \hat{\boldsymbol{\delta}}, \tau_0^2, J_{10}, J_{00}$ are defined as in Section 2, except in terms of the partial likelihood instead of the likelihood. Newly-defined quantities are

$$F(t) = \int_0^t \left\{ G_n^{(1)}(u) / G_n^{(0)}(u) \right\} dH_0(u) = \begin{pmatrix} F_0(t) \\ F_1(t) \end{pmatrix}$$

where $F_0(t)$ and $F_1(t)$ have p and q components respectively, and

- $G_n^{(0)}(u) = \frac{1}{n}\sum_{i=1}^n Y_i(u)\exp(x_i^T \boldsymbol{\theta} + z_i^T \boldsymbol{\gamma})$
- $G_n^{(1)}(u) = \frac{1}{n}\sum_{i=1}^n Y_i(u)\exp(x_i^T \boldsymbol{\theta} + z_i^T \boldsymbol{\gamma}) \begin{pmatrix} x_i \\ z_i \end{pmatrix}$

both evaluated at the estimates of $\boldsymbol{\theta}$, $\boldsymbol{\gamma}$ and $H_0()$ from the wide model, and $Y_i(t) = I(t_i \geq t)$ is the indicator for individual i being at risk at time t. As before TODO BIAS CORRECTION.

6.1. Example: malignant melanoma

To illustrate the method, Hjort and Claeskens (2006) study a dataset from 205 patients with malignant melanoma, earlier analysed in detail by Andersen *et al.* (1993). This dataset is also provided in the **fic** package.

We compare models ranging from a wide model `wide` that includes 7 terms in the regression model formula, to a narrow model that includes only sex.

```
library(survival)
wide <- coxph(Surv(years, death==1) ~ sex + thick_centred + infilt + epith +
                ulcer + depth + age, data=melanoma)
```

In this example, we need to deal with *factor* terms in the model when setting up the `inds` and `inds0` indicators to supply to `fic`. Specifically, `infilt` and `depth` are factors with 4 and 3 levels respectively, represented in the model by 3 and 2 model parameters respectively, instead of one parameter for each. The remaining terms in the model are each associated with one parameter. Thus the wide model, with 7 terms, includes 10 parameters.

The function `expand_inds` can be used to construct `inds` or `inds0` terms in the presence of factors[1]. We supply a vector of 7 elements, indicating the presence or absence of each of the 7 terms in the model formula. In this case, only the first term, `sex`, is included in the narrow model. Then to create an `inds0` vector of 10 elements, indicating the presence or absence of each of the wide model's 10 parameters in the narrow model, we call

```
inds0 <- expand_inds(c(1,0,0,0,0,0,0), wide)
inds0

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    0    0    0    0    0    0    0    0     0
```

Note that in Cox regression there is no intercept parameter, therefore the model parameters include only the regression coefficients. In fully parametric regression models, for example GLMs, the vector supplied to `expand_inds` should contain an additional element indicating the presence of the intercept.

The `fic` package includes three built-in alternative focuses, as specified through the `focus` argument to `fic`.

- `focus="hr"`: the hazard ratio between an individual with covariates X and an individual with covariates 0 (which by definition of the Cox model is independent of time t)

- `focus="survival"`: the survival probability at time t, for an individual with covariates X

- `focus="cumhaz"`: the cumulative hazard at time t, for an individual with covariates X

[1] This function only works for classes of models for which the `model.matrix` function is understood and returns objects with an `"assign"` attribute. This includes all the commonly-used models in base R.

The required covariate values and/or time point(s) are supplied as the `X` and `t` arguments to `fic`.

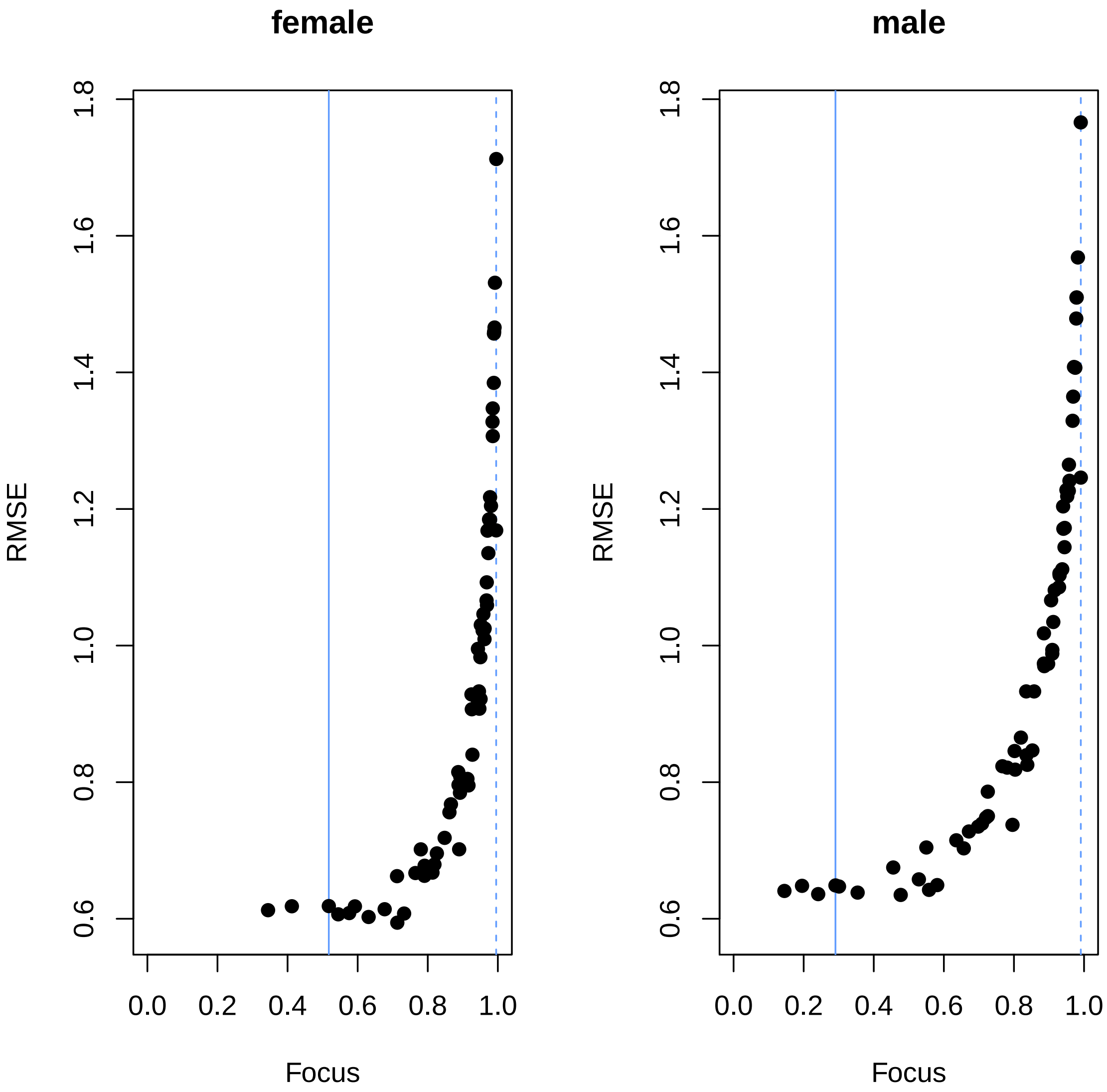
Supplying alternative focuses requires the user to supply a list of three functions as the `"focus"` argument to `fic`: one for the focus, and two for its derivatives with respect to $(\boldsymbol{\theta}, \boldsymbol{\gamma})$ and $H_0(t)$ respectively. The format of these functions is shown in the package source file `R/fic.coxph.R`, see also `help(fic.coxph) TODO EXAMPLES`.

In the melanoma example, all possible submodels spanning the wide and narrow model are compared. As before, a matrix of indicators describing these models is constructed, and the submodels are fitted automatically within the `fic` function.

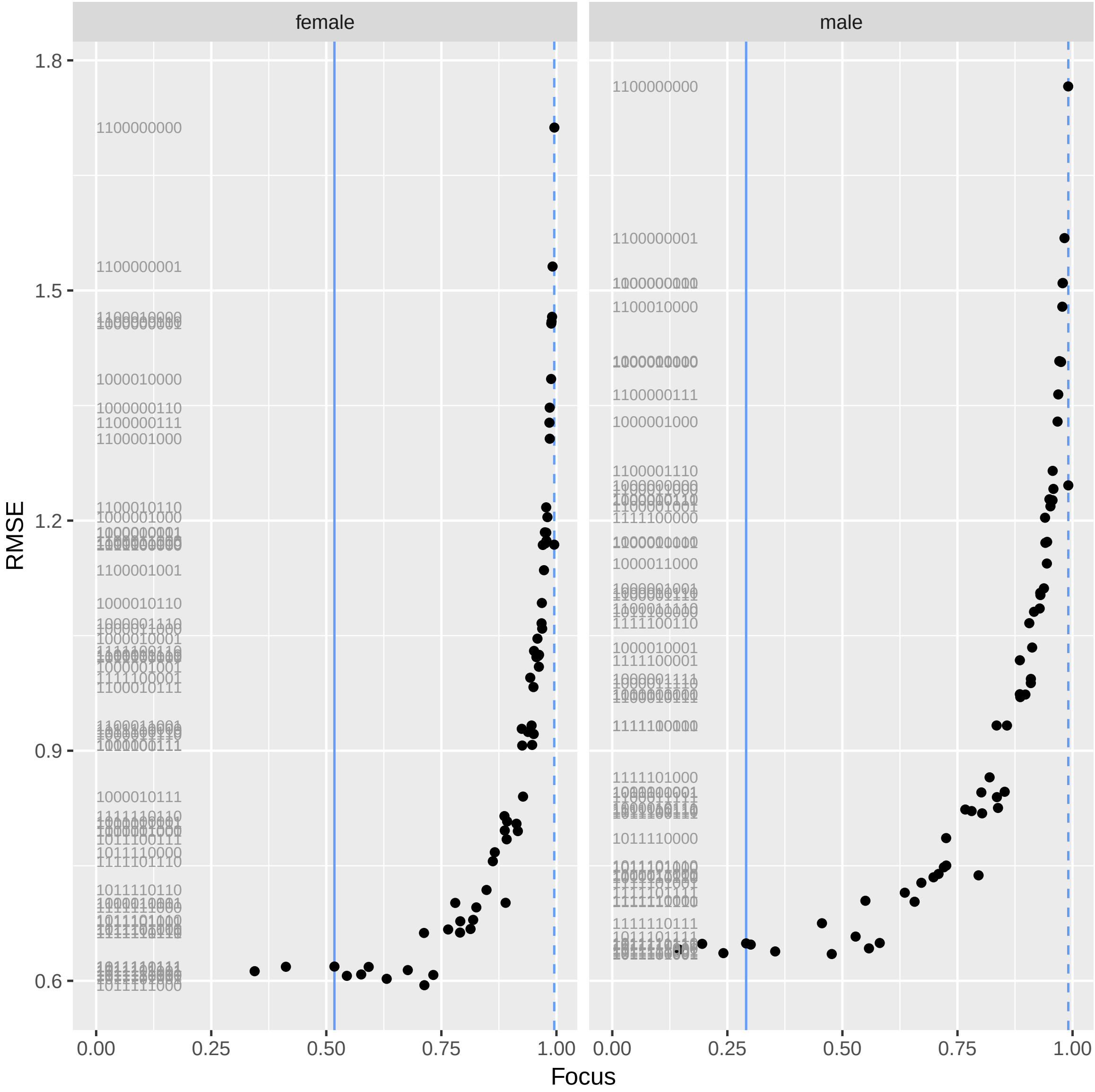
```
combs <- all_inds(wide,inds0,intercept=FALSE)
```

The focus is defined as the 5 year survival probability (`focus=■survival■,t=5`) for the covariate values defined by `X`, here taken as men and women separately, with average age and mean observed tumour thickness among men and women, infiltration level 4, epithelioid cells and ulceration present, and invasion depth 2. The utility `newdata_to_X` is used to convert the user-defined data frame `newdata` that identifies these covariate values, with one variable per covariate or factor, to a design matrix `X`, with one column for each of the 10 parameter values.

```
newdata <- with(melanoma,
                data.frame(sex = c("female","male"),
                           thick_centred = tapply(thick_centred, sex, mean),
                           infilt=4, epith=1, ulcer=1, depth=2,
                           age = tapply(age, sex, mean)))
X <- newdata_to_X(newdata, wide)
ficall <- fic(wide, inds=combs, inds0=inds0, focus="survival", X=X, t=5)
plot(ficall, xlim=c(0,1), ci=FALSE)
```



```
ggplot_fic(ficall, ci=FALSE)
```



[doesn't quite match fig 6.5 in book.]

[note big range of estimates with low FIC, model uncertainty]

7. Bootstrap

Would like to

- illustrate alternative losses to the mean square error
- use resampling to illustrate FIC principles

8. Discussion

Post-selection inference, model averaging. Could we recommend bootstrapping the model selection procedure?

High dimensional regression, (Gueuning and Claeskens 2018)

References

Akaike H (1973). "Information theory and an extension of the maximum likelihood principle." In B Petrov, F Csaki (eds.), *2nd International Symposium on Information Theory*, pp. 267–281. Akademiai Kiado, Budapest.

Andersen PK, Borgan O, Gill RD, Keiding N (1993). *Statistical models based on counting processes*. Springer, New York.

Burnham KP, Anderson DR (2003). *Model selection and multimodel inference: a practical information-theoretic approach*. Springer.

Claeskens G, Hjort N (2003). "The focused information criterion (with discussion)." *Journal of the American Statistical Association*, **98**(464), 900–945.

Claeskens G, Hjort N (2008). *Model selection and model averaging*. Cambridge University Press.

Gilbert P, Varadhan R (2016). *numDeriv: Accurate Numerical Derivatives*. R package version 2016.8-1, URL `https://CRAN.R-project.org/package=numDeriv`.

Gueuning T, Claeskens G (2018). "A High-dimensional Focused Information Criterion." *Scandinavian Journal of Statistics*, **45**(1), 34–61.

Hjort NL, Claeskens G (2006). "Focused information criteria and model averaging for the Cox hazard regression model." *Journal of the American Statistical Association*, **101**(476), 1449–1464.

Hosmer DW, Lemeshow S (1989). *Applied Logistic Regression*. John Wiley & Sons.

Kass RE, Wasserman L (1995). "A reference Bayesian test for nested hypotheses with large samples." *Journal of the American Statistical Association*, **90**, 928–934.

Schwarz G (1978). "Estimating the dimension of a model." *The Annals of Statistics*, **6**(2), 461–464.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition edition. Springer.