

Worked examples of decision curve analysis using R

A note about R versions

The R script files to implement decision curve analysis were developed using R version 2.3.1, and were tested last using R version 2.9.2 on April 27, 2010.

A note about this tutorial

This tutorial was developed using R version 2.3.1 with the `Design` and `Hmisc` libraries added. The code in the tutorial uses the following functions from these libraries: **`lrm`**, **`cph`**, and **`survest`**. We strongly recommend using these functions because they simplify the code required to obtain predicted probabilities from the models. If you do not use the `Design` and `Hmisc` libraries, then you will need to modify the relevant lines of code in the tutorial, using the corresponding functions available outside of these libraries (for example, **`glm`** in place of **`lrm`**) and appropriate rewriting of the programming logic.

Basic set up

This example assumes that the user has sourced the R files “`dca.R`” and “`stdca.R`”; loaded the `Design`, `Hmisc` and `cmprsk` libraries; and the example text files are saved locally.

```
dataset1 <- read.delim("c:\dca_example_dataset1.txt", header = TRUE, sep = "\t")  
attach(dataset1)
```

This is a data set giving the results of research study on early detection of cancer. The data include the results of a clinical examination for cancer, the levels of a molecular marker and then the results of biopsy, the “gold standard” for whether or not the patient had cancer.

The results of the clinical exam first have to be turned into a numeric variable.

```
pos_test <- clinical_exam=="suspicious for cancer"
```

We then wish to examine the value of a statistical model that incorporates both the clinical examination and the marker. We create a new variable, giving probabilities of cancer for each patient on the basis of the model.

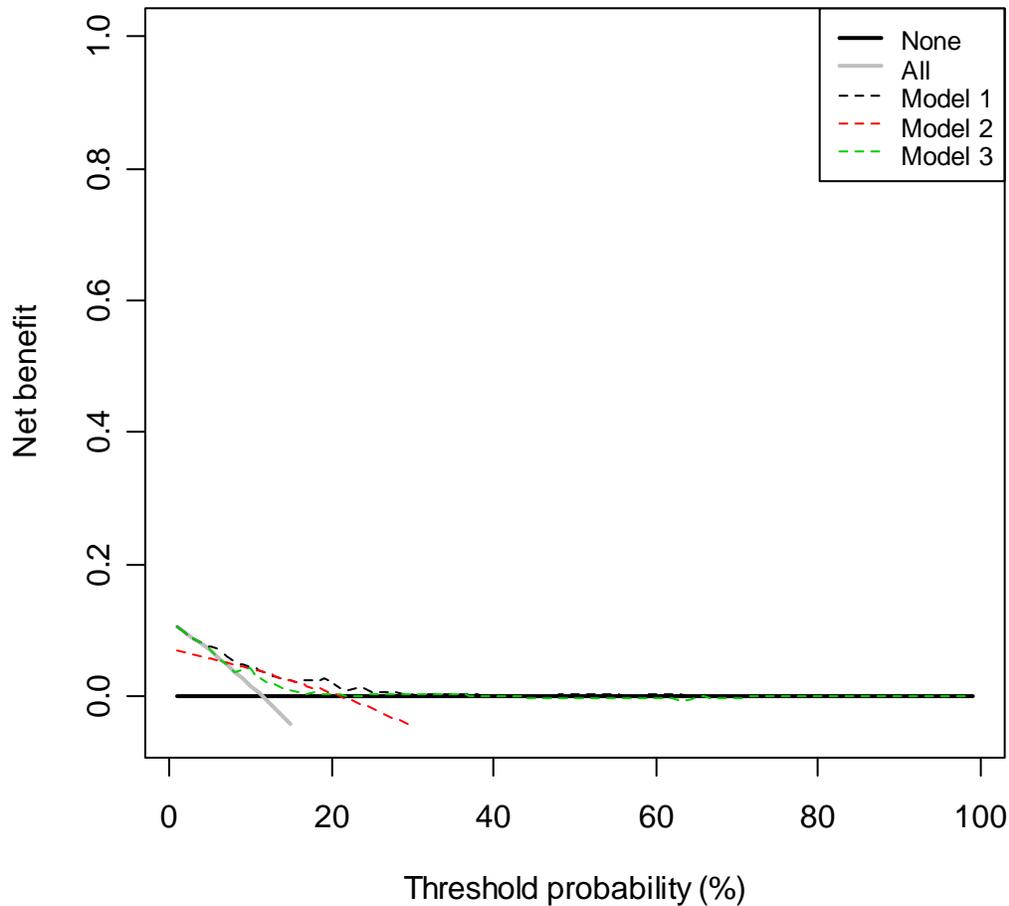
```
model <- predict(lrm(cancer ~ marker + pos_test), type="fitted")
```

Calculating a decision curve

We now want to compare our different approaches to cancer detection: biopsying everyone, biopsying no-one (i.e. no screening), biopsying on the basis of the clinical exam, biopsying on the basis of the molecular marker, or biopsying on the basis of a multivariable statistical model

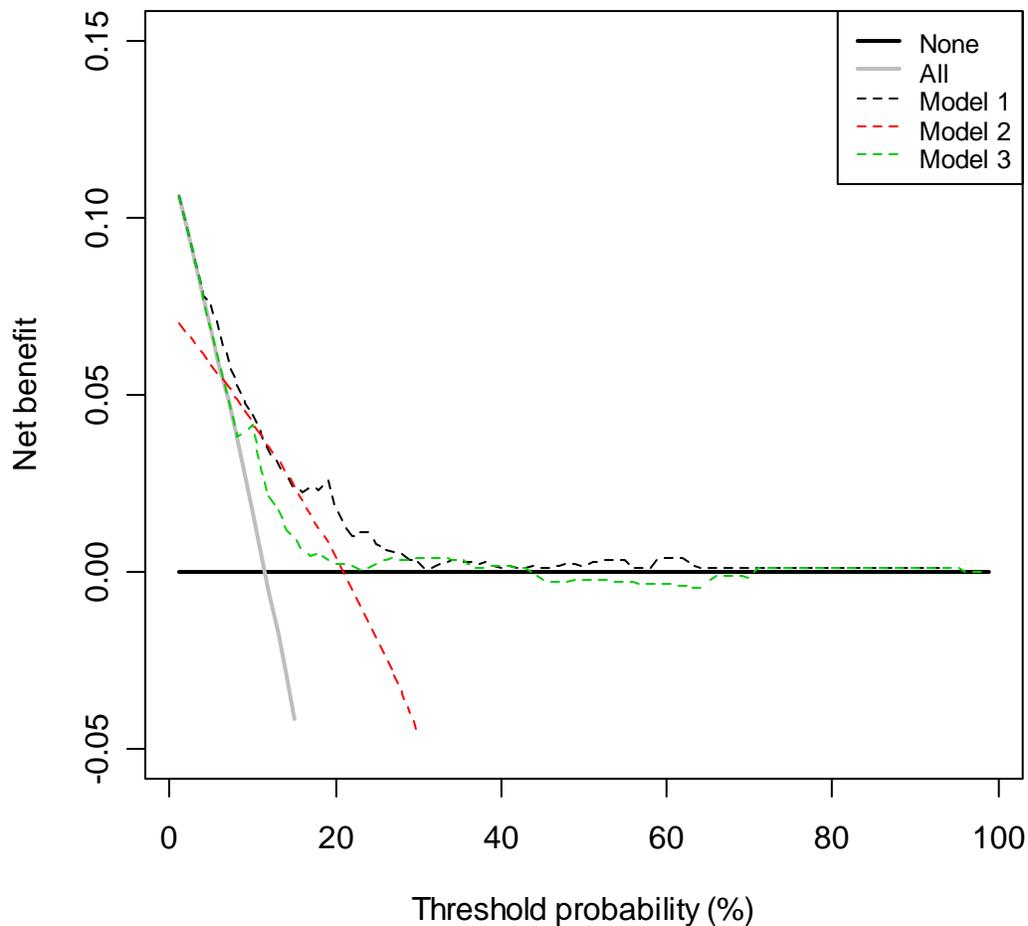
including both the marker and clinical examination data. We have to specify that the data for the marker variable – unlike those for the model and the test – are not probabilities.

```
dca(yvar=cancer, xmatrix=cbind(model, pos_test, marker), prob=c("Y", "Y", "N"))
```



In the figure legend, the models are numbered according to how they were entered in **xmatrix**. Note that without specifying **ymin** or **ymax**, the default is to plot the y-axis from -0.05 to 1.0. Since the maximum net benefit observed was near 0.15, we can get a clearer picture of the decision curve by specifying **ymax=0.15**.

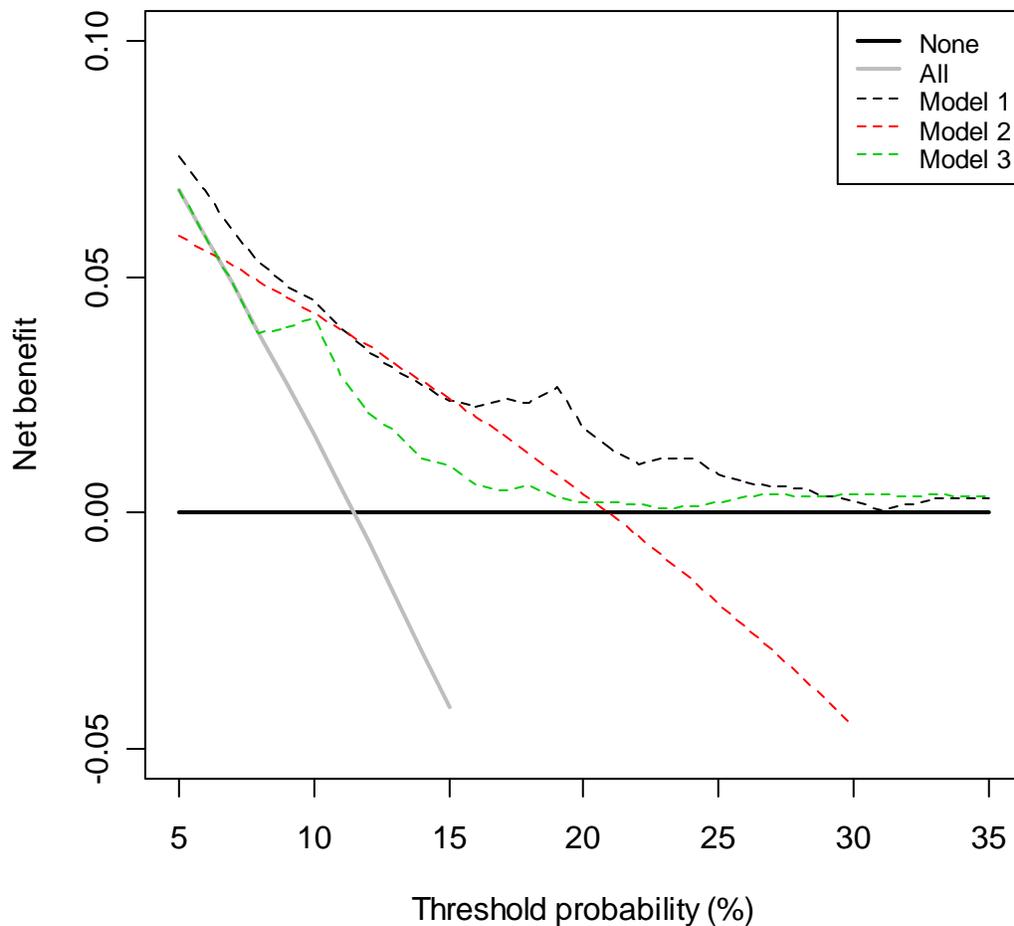
```
dca(yvar=cancer, xmatrix=cbind(model, pos_test, marker), prob=c("Y", "Y", "N"),  
    ymax=0.15)
```



This shows a lot of threshold probabilities that are uninteresting, for example, it is unlikely that someone would require an 80% probability of cancer before they would agree to biopsy. Let's imagine that, after discussing the matter with clinicians, we thought a reasonable range of threshold probabilities was 5% to 35%. In other words, some patients might opt for biopsy even if they had only a 5% chance of cancer, but it would be a rare clinician who would do more than 20 biopsies to find a cancer; on the other hand, few patients would refuse biopsy if their risk was greater than 1 in 3.

Here is a neater way of showing the graph. It focuses on the key range of threshold probabilities (5% - 35%).

```
dca(yvar=cancer, xmatrix=cbind(model, pos_test, marker),
    prob=c("Y", "Y", "N"), xstart=0.05, xstop=0.35, ymax=0.10)
```



The key aspect of the graph is to look at which strategy leads to the highest net benefit. The black dashed line corresponds to the model that includes both the marker levels and the results of the clinical exam. It has the greatest net benefit (i.e. it is the “highest” line) at all threshold probabilities. This is apart from two possible exceptions: between 10 and 15% it is roughly equivalent to using the clinical exam alone; above 30%, it is not much better than using the marker alone. Nonetheless, it is clear that, across the range of reasonable threshold probabilities, one cannot go far wrong basing clinical decisions on the model: it is generally superior, and unlike any alternative strategy, it is never more than trivially worse.

A few general points of interest. First, look at the grey line, the net benefit for “treat all”, that is, biopsy everyone. This crosses the y axis at the prevalence. Imagine that a man had a risk threshold of 12%, and asked his risk under the "biopsy all" strategy. He would be told that his risk was the prevalence (12%). When a patient’s risk threshold is the same as his predicted risk, the net benefit of biopsying and not biopsying are the same. Second, the decision curve for the

binary test (the clinical exam, red dashed line) crosses that for “biopsy all men” at 1 – negative predictive value and again, this is easily explained: the negative predictive value is 94%, so a patient with a negative test has a probability of disease of 6%; a patient with a threshold probability less than this – for example, a patient who would opt for biopsy even if risk was 3% - should therefore be biopsied even if the clinical exam is negative. The decision curve for the binary test is equivalent to biopsy no-one at the positive predictive value. This is because for a binary test, a patient with a positive test is given a risk at the positive predictive value.

Saving out net benefit

If we wanted to show the net benefits in a table, we could save them out. The following code can be used to produce a table giving the net benefits of each strategy, and the advantage of the model, at risk thresholds of 5%, 10%, 15% ... 35%.

```
output <- dca(yvar=cancer, xmatrix=cbind(model, pos_test, marker),  
              prob=c("Y", "Y", "N"), xstart=0.05, xstop=0.35, xby=0.05)  
delta <- output$modelp1 – output$all  
label(delta) <- “Increase in net benefit from using statistical model”
```

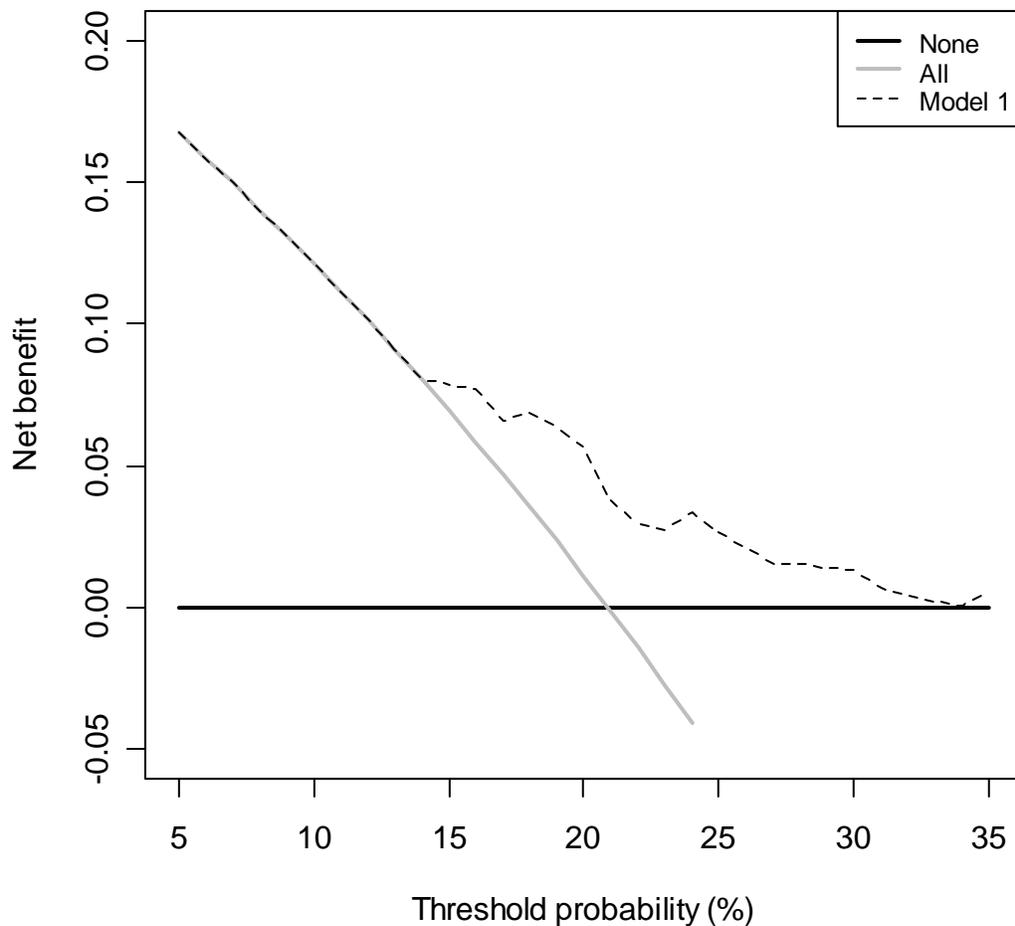
Showing net reduction in interventions

Now let’s imagine that the traditional approach was to biopsy everyone suspicious for cancer on clinical examination. We want to know whether assessing the marker in these patients would help reduce unnecessary biopsies. A typical research study to address this question would only have biopsy data on patients with a suspicious clinical examination. We’ll first create this data set.

```
dataset1 <- read.delim("c:\dca_example_dataset1.txt", header = TRUE, sep = "\t")  
attach(dataset1)  
subset.suspicious <- clinical_exam=="suspicious for cancer"
```

Here is the decision curve:

```
dca(yvar=cancer[subset.suspicious], xmatrix= marker[subset.suspicious],  
    prob="N", xstart=0.05, xstop=0.35, ymax=0.2)
```

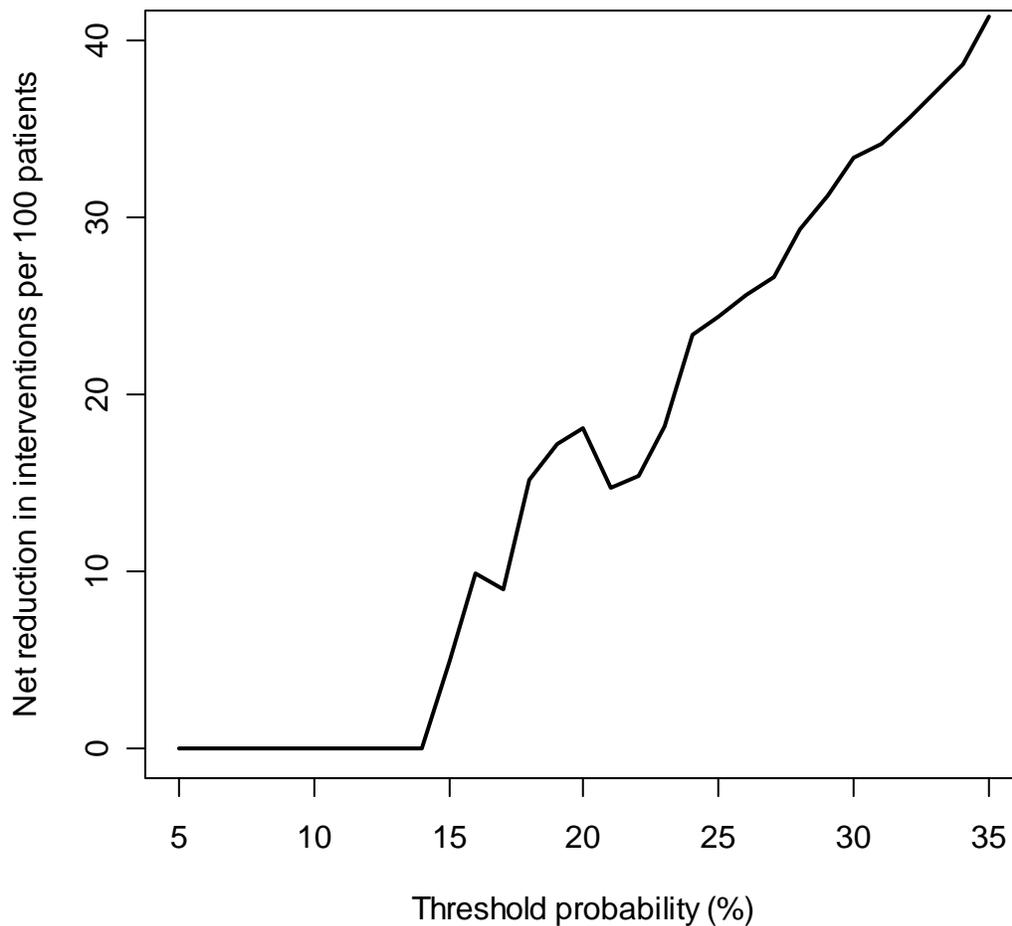


Use of the marker has superior net benefit from a threshold probability of about 10% or more. This suggests that measuring the marker would improve outcome for all but the most risk averse patients. The appropriate clinical conclusion would depend on the expense of measuring the marker, and the estimated prevalence of very risk averse patients. For example, if the marker were inexpensive, and few patients were willing to have biopsy at very low probabilities of cancer, it would be reasonable to measure the marker on all patients. If, on the other hand, the marker was expensive, or many patients were thought to have low threshold probabilities, a reasonable approach would be to conduct the clinical exam, discuss the results with the patient, and then order the marker if the patient was unsure what to do.

Net benefit has a ready clinical interpretation. In the figure above, the net benefit of 0.05 at a threshold probability of 20% can be interpreted as: "Comparing to conducting no biopsies, biopsying on the basis of the marker is the equivalent of a strategy that found 5 cancers per hundred patients without conducting any unnecessary biopsies." However, the routine clinical practice is to biopsy everyone with a suspicious clinical examination and what we want to know

is the effect on the number of unnecessary biopsies. We can show this using decision curve analysis by converting the net benefit to the reduction in biopsies, as follows.

```
# run the decision curve analysis, saving the output
dcaoutput <- dca(yvar=cancer[subset.suspicious], xmatrix= marker[subset.suspicious],
                prob="N", xstart=0.05, xstop=0.35, ymin=-1)
#compute reduction using formula
reduction <- (dcaoutput$modelp1 - dcaoutput$all) * 100 / (dcaoutput$threshold /
                (100 - dcaoutput$threshold))
#cut off reductions < 0
reduction <- pmax(0, reduction)
#now create the graph
plot(dcaoutput$threshold, reduction, type="l", lwd=2, xlim=c(5, 35),
     ylim=c(0, 40), xlab="Threshold probability (%)",
     ylab="Net reduction in interventions per 100 patients")
```



At a probability threshold of 20%, the net reduction in interventions is about 18 per 100 patients. In other words, at this probability threshold, biopsying patients on the basis of the marker is the equivalent of a strategy that reduced the biopsy rate by 18%, without missing any cancers.

Joint or conditional tests

Many decisions in medicine are based on joint or conditional test results. A classic example is where patients are categorized on the basis of one test as being at high, low or intermediate risk. Patients at high risk are referred immediately for treatment; patients at low risk are reassured and advised that no further action is necessary; patients at intermediate risk are sent for an additional test, with treatment decisions made accordingly. Alternatively, the results of two tests can be combined using an algorithm, rather than a multivariable model.

Decision curve analysis can incorporate joint or conditional testing. All that is required is that appropriate variables are calculated from the data set; decision curves are then calculated as normal. To see how this works in practice, see the following example.

```
dataset2 <- read.delim("c:\dca_example_dataset2.txt", header = TRUE, sep = "\t")  
attach(dataset2)
```

These data are from a heart disease study, with data on whether the patient had a subsequent cardiovascular event (yes or no), the result of a clinical exam (high low or intermediate risk) and the results of a scan. There are five clinical options:

1. Treat everyone with medication
2. Treat no-one (i.e. no screening)
3. Treat everyone at high risk from the clinical exam; don't do a scan
4. Scan everyone, then treat everyone who is either high risk from the clinical exam or who has a positive scan (i.e. joint approach)
5. Treat everyone at high risk; scan patients at intermediate risk and treat those with a positive scan (i.e. conditional approach)

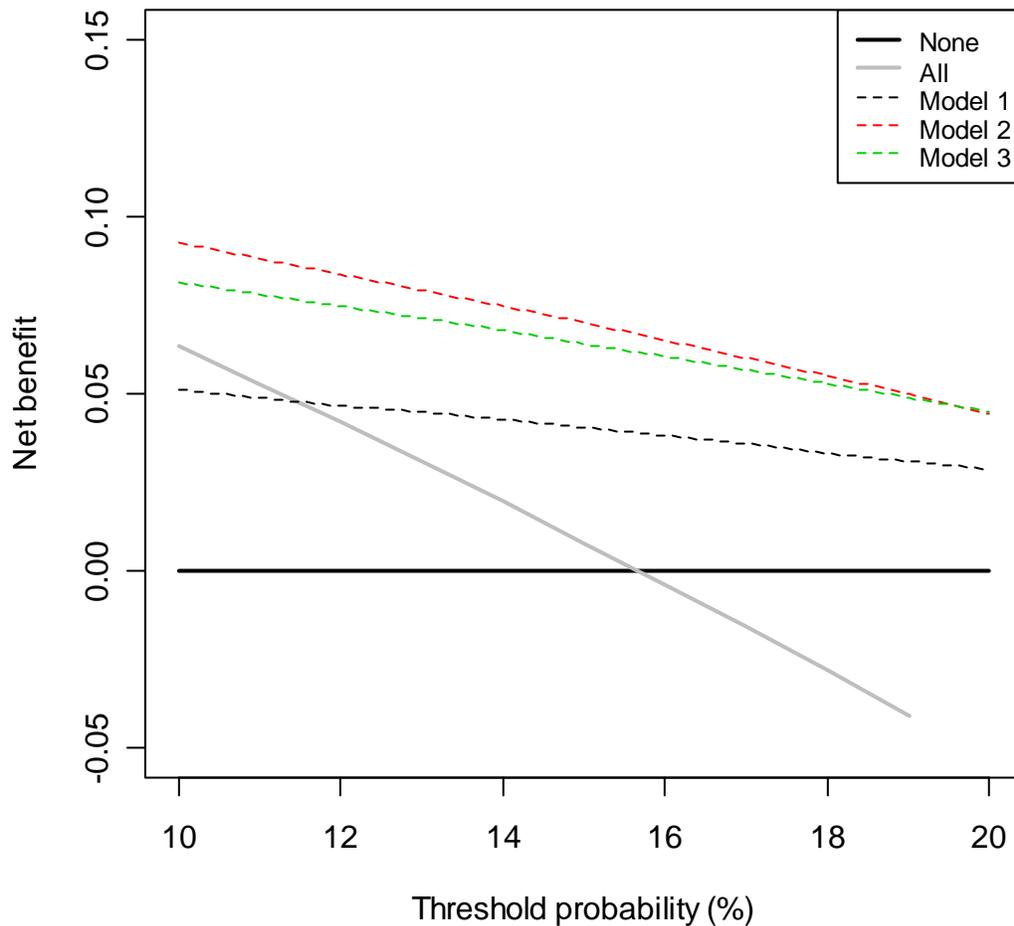
Here is how to create the variables to use in decision curve analysis.

```
#clinical exam: treat high risk patients only  
clinical_test <- clinical_exam=="high risk"  
#joint test is positive if either:  
#scan is positive or clinical exam gives high risk  
joint <- clinical_exam=="high risk" | scan==1  
#conditional test: treat if high risk; scan if intermediate risk  
conditional <- clinical_exam=="high risk" |  
  (clinical_exam=="intermediate risk" & scan==1)
```

Now we have the variables worked out, we can run the decision curve analysis. In cardiovascular disease, treatment thresholds are typically 10% - 20.

```
dca(yvar=event, xmatrix=cbind(clinical_test, joint, conditional),  
  prob=rep("Y", 3), xstart=0.1, xstop=0.2, ymax=0.15)
```

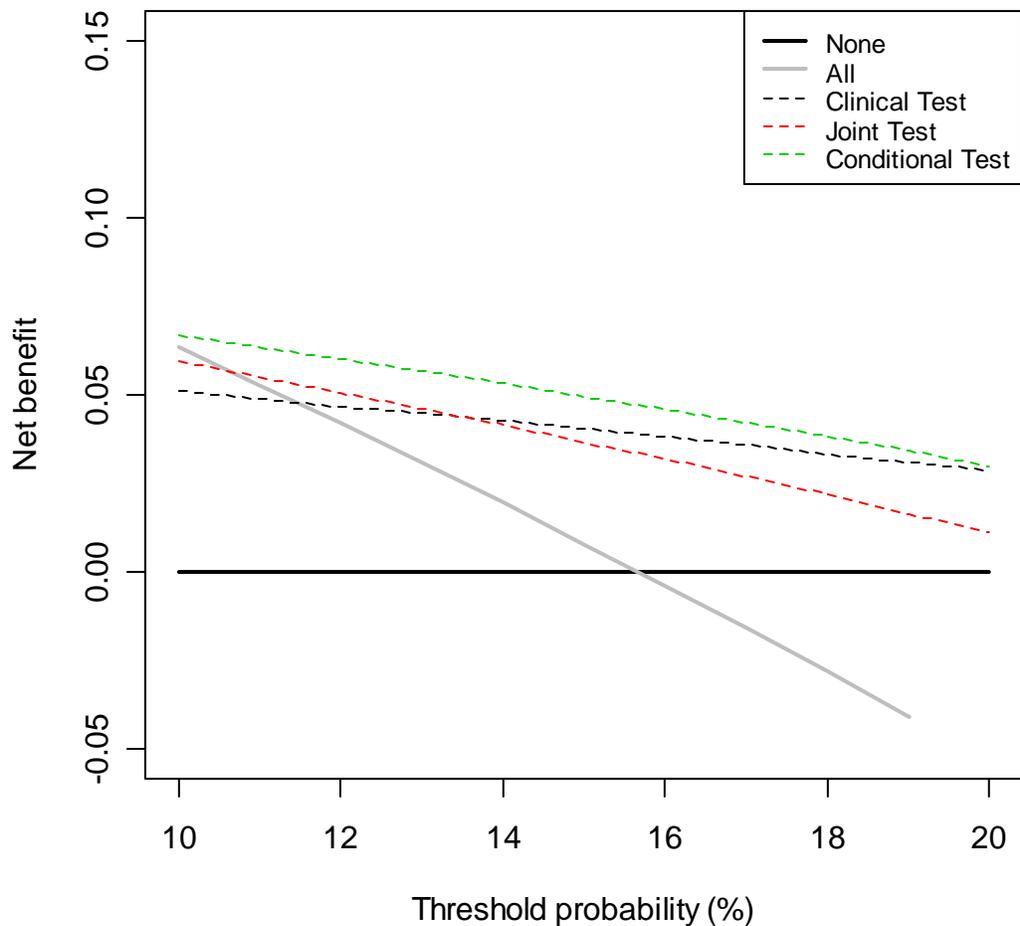
This gives the figure below:



This appears to show that the joint test is the best option, because it has the highest net benefit across almost the whole range of threshold probabilities. The obvious disadvantage of the joint test is that everyone needs to be scanned, and scanning is expensive and time consuming. To incorporate the harm of scanning, we ask a clinician, who tells us that, even if the scan was perfectly accurate, few clinicians would conduct more than 30 scans to predict one cardiovascular event. The “harm” of the scan is the reciprocal of 30, or 0.0333.

To construct the decision curves for each strategy we now incorporate harm. We have to calculate harm specially for the conditional test, because only patients at intermediate risk are scanned.

```
#the harm of a scan is stored in a scalar  
harm_scan <- 0.0333  
#in the conditional test, only pts at intermediate risk are scanned  
intermediate_risk<- clinical_exam=="intermediate risk"  
#harm of the conditional strategy is:  
#proportion scanned multiplied by harm of the scan  
harm_conditional = mean(intermediate_risk)*harm_scan  
#run the decision curve analysis  
dcaoutput <- dca(yvar=event, xmatrix=cbind(clinical_test, joint, conditional),  
                  prob=rep("Y", 3), xstart=0.1, xstop=0.2)  
#subtract the harm from the net benefit, as appropriate  
# the harm of model 1 (clinical test) is zero  
newmodelp1 <- dcaoutput$modelp1 - 0  
# the harm of model 2 (joint test) is the harm of the scan  
newmodelp2 <- dcaoutput$modelp2 - harm_scan  
# the harm of model 3 (conditional test) is the harm of the conditional test  
newmodelp3 <- dcaoutput$modelp3 - harm_conditional  
#now create the graph  
plot(dcaoutput$threshold, dcaoutput$none, type="l", lwd=2, xlim=c(10, 20),  
      ylim=c(-0.05, 0.15), xlab="Threshold probability (%)", ylab="Net benefit")  
lines(dcaoutput$threshold, dcaoutput$all, type="l", col=8, lwd=2)  
lines(dcaoutput$threshold, newmodelp1, type="l", col=1, lty=2)  
lines(dcaoutput$threshold, newmodelp2, type="l", col=2, lty=2)  
lines(dcaoutput$threshold, newmodelp3, type="l", col=3, lty=2)  
legend("topright", cex=0.8, legend=c("None", "All", "Clinical Test",  
          "Joint Test", "Conditional Test"), col=c(17, 8, 1, 2, 3),  
      lwd=c(2, 2, 1, 1, 1, 1), lty=c(1, 1, 2, 2, 2, 2))
```

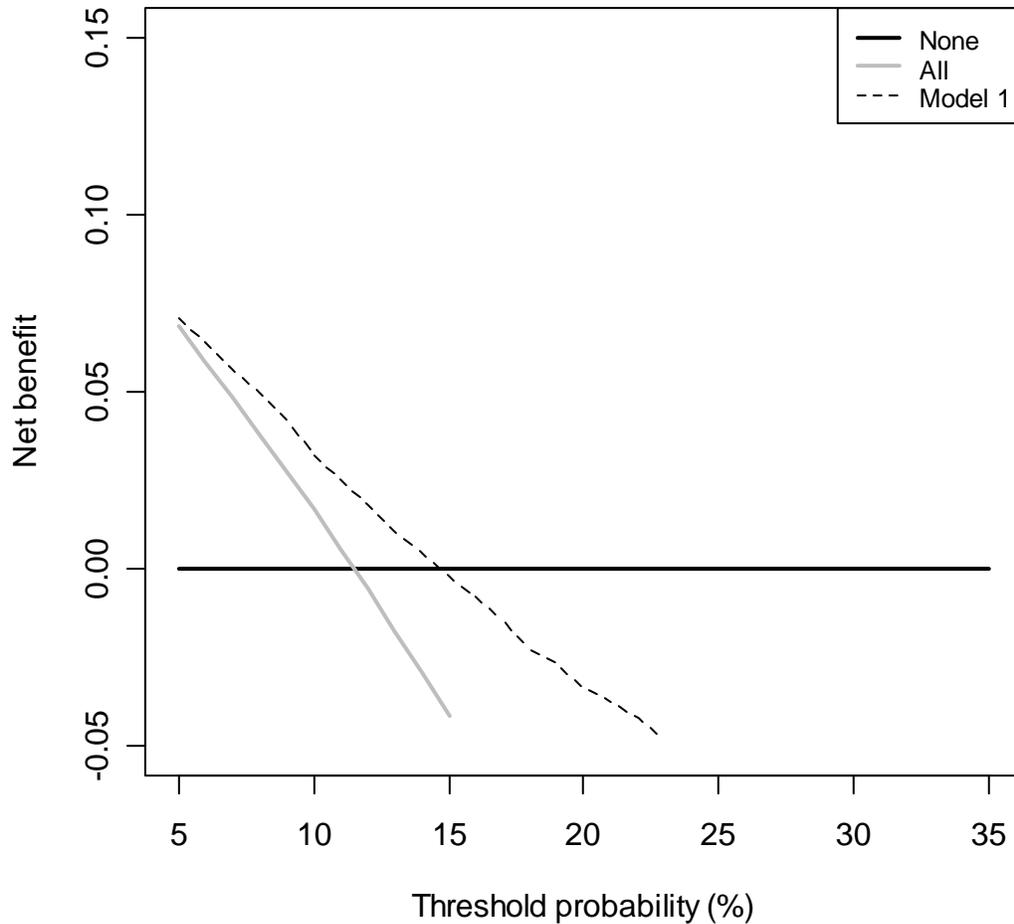


Here the conditional test is clearly the best option, in fact, once you take into account the harm of scanning, it is clearly not worth scanning everyone: the net benefit of just using the clinical exam is often higher than that of the joint test.

Evaluation of published models

Imagine that a model was published by Brown et al. with respect to our cancer biopsy data set. The authors reported a statistical model with coefficients of 1.5 for a positive clinical exam; 0.042 for each 10 point increase in the marker, and an intercept of -3.5. To test this formula on our data set:

```
dataset1 <- read.delim("c:\dca_example_dataset1.txt", header = TRUE, sep = "\t")
attach(dataset1)
logodds_Brown <- 1.5*(clinical_exam=="suspicious for cancer")+
  0.42 * (marker/10) - 3.5
phat_Brown <- exp(logodds_Brown)/(1+exp(logodds_Brown))
dca(yvar=cancer, xmatrix=phat_Brown, prob="Y", xstart=0.05, xstop=0.35, ymax=0.15)
```



This decision curve suggests that although the model might be useful in the most risk averse men, it is actually harmful in patients with more moderate threshold probabilities. As such, the Brown et al. model should not be used in clinical practice.

Application to case-control data

The problem with applying decision curve analysis to case-control data is that net benefit depends on prevalence, and prevalence in case-control studies is fixed by design. This problem can be solved by use of recalibration (sometimes called a “Bayes factor”).

```
dataset3 <- read.delim("c:\dca_example_dataset3.txt", header = TRUE, sep = "\t")  
attach(dataset3)
```

This is a case-control study to assess whether a certain genetic mutation, along with smoking history, can predict a cancer. The matching was 3:1, such that the prevalence of cancer in the study was 25%. We will assume that the true risk of cancer by age 75 was 5%. We will first build a model using the data set and then obtain the log odds of disease (linear predictor or fitted values).

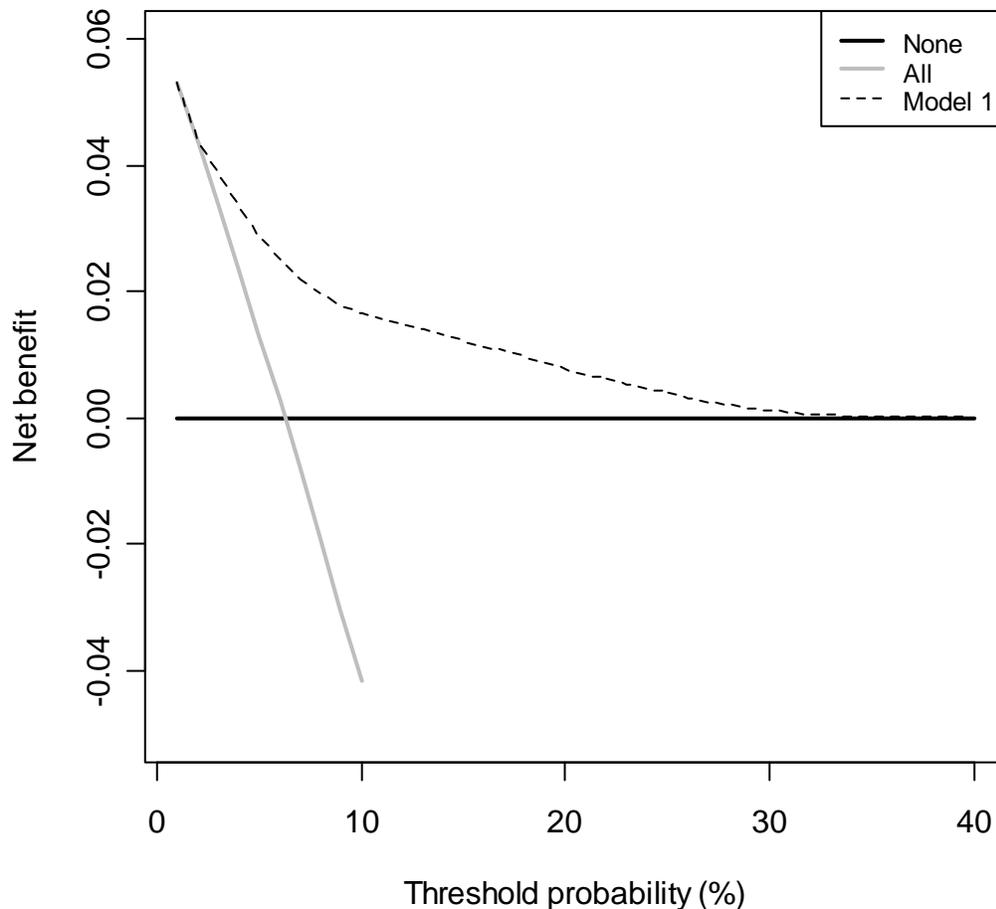
```
xb <- predict(lrm(cancer ~ gene + packyears))
```

We can then add the Bayes factor to the linear predictor, which is the log of the true odds of cancer divided by the odds of cancer in the case-control study.

```
true <- 0.05  
design <- mean(cancer)  
Bayes <- log((true/(1-true)) / (design/(1-design)))  
xb <- xb + Bayes
```

This is then converted to a probability. We use this probability both as the predictor and the outcome for the decision curve, using the assumption that the model is well-calibrated.

```
phat <- exp(xb) / (1 + exp(xb))  
dca(yvar=phat, xmatrix=phat, xstop=0.4, prob="Y", ymax=0.06)
```



This decision curve can be used to evaluate whether a statistical model based on smoking history and gene status should be used to determine intensity of cancer screening. The decision curve shows that the statistical model would be helpful for decisions with a threshold probability of 2% or above. In other words, in the unlikely event that a patient would agree to be screened even if he or she had a 1% lifetime risk of cancer, they should be screened irrespective of smoking history and gene status; otherwise, screening should be based on the risks from the model.

Application to survival data

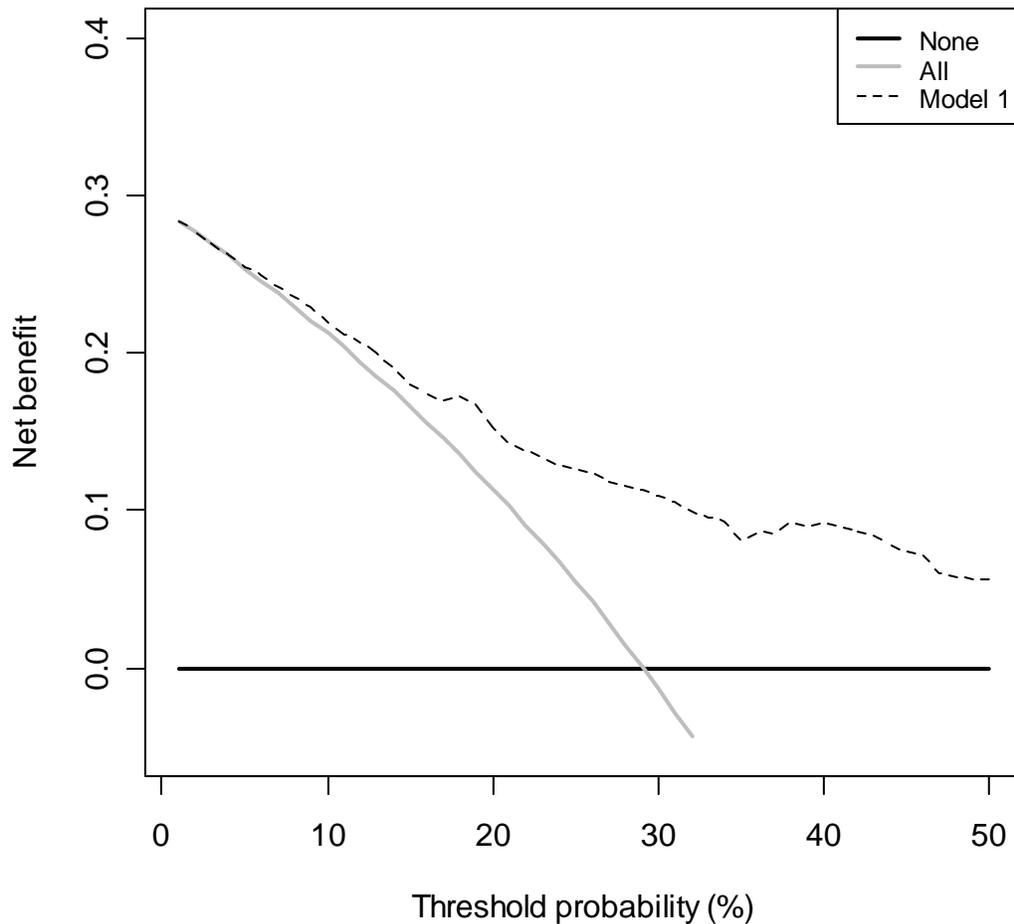
Again, we'll assume that the user has sourced the R files "dca.R" and "stdca.R"; loaded the Design and Hmisc libraries; and the example text files are saved locally.

```
dataset4 <- read.delim("c:\dca_example_dataset3.txt", header = TRUE, sep = "\t")
attach(dataset4)
```

This is a bladder cancer data set consisting of seven variables: some patient characteristics (age at surgery, pathologic stage, and nodal status), a time to event variable in months, whether the patient recurred, whether the patient died, and then predicted probability of recurrence by 60 months, obtained from a statistical model (or “nomogram”).

The code for running the decision curve analysis is straightforward. All we have to do is specify the timepoint we are interested in. Here we ask: how good is the nomogram at predicting recurrence by five years?

```
stdca(timepoint=60, time=ttlastfollowrecurrence, failure=recurrence,
      xmatrix=prednomogram, xstop=0.5, ymax=0.4)
```



This shows that using the nomogram to inform clinical decisions will lead to superior outcomes for any decision associated with a threshold probability of above 5% or so.

The input predictor variables for `stdca` must be given as probabilities, even when the prediction model consists of a single variable. In the previous example, the nomogram probability was calculated from a published statistical model. Let's say we wanted to build a model of our own, based on age, pathologic stage, and nodal status. The survival probability to time-point t can be derived from any type of survival model; here we use a Cox as this is the most common model in statistical practice. The formula for a survival probability from a Cox model is given by:

$$s(t|X) = s_0(t)^{\exp(XB)}$$

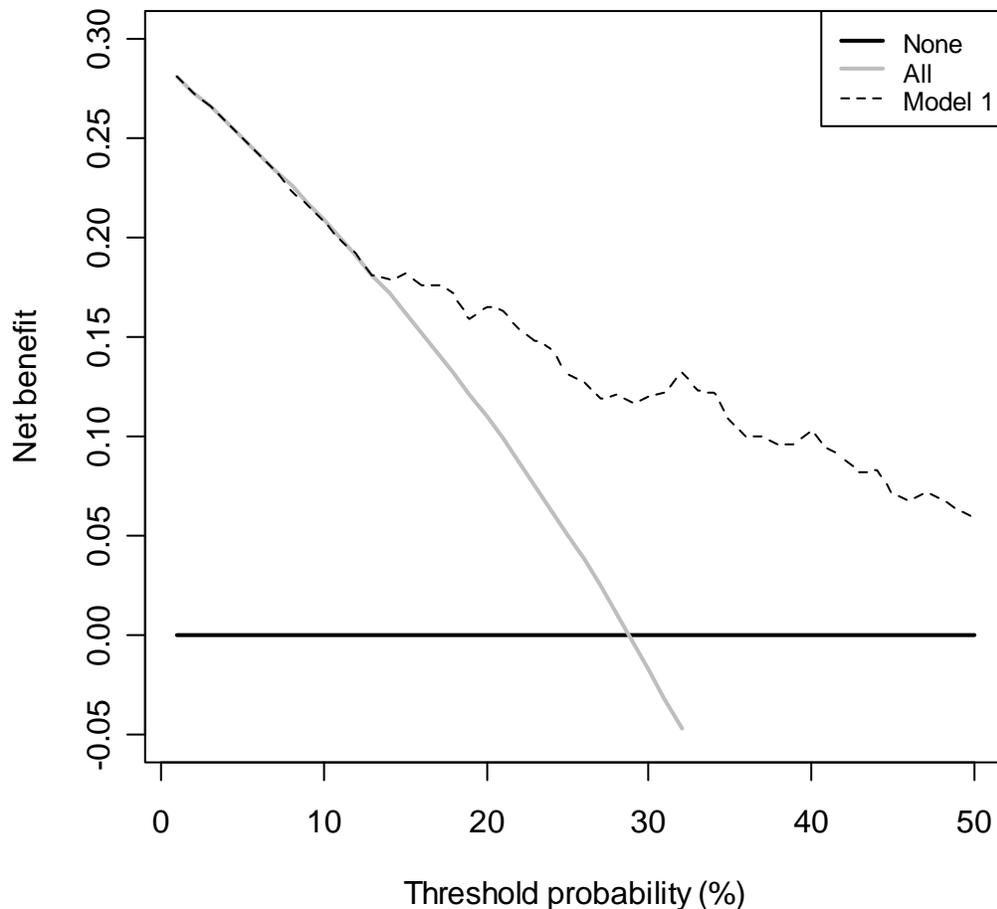
Where X is matrix of covariates in the Cox model, B is a vector containing the parameter estimates from the Cox model, and $s_0(t)$ is the baseline survival probability to time t .

This can be done in R by running a Cox model with age, pathologic stage, and nodal status as predictors:

```
fit <- cph(Surv(ttlastfollowrecurrence, recurrence) ~ age + pathstage + nodalstatus,  
           x=TRUE, y=TRUE)
```

The probability of failure to time t can then be computed using the `survest` function (note that you must specify `x=TRUE` and `y=TRUE` in the previous `cph` function for the `survest` function to work). We'll use a timepoint of 60, which corresponds to 60 months or 5 years in this data set.

```
f60 <- 1 - survest(fit, newdata=dataset4, times=60)$surv  
# now run decision curve  
stdca(timepoint=60, time=ttlastfollowrecurrence, failure=recurrence,  
       xmatrix=f60, xstop=0.5, ymax=0.3)
```

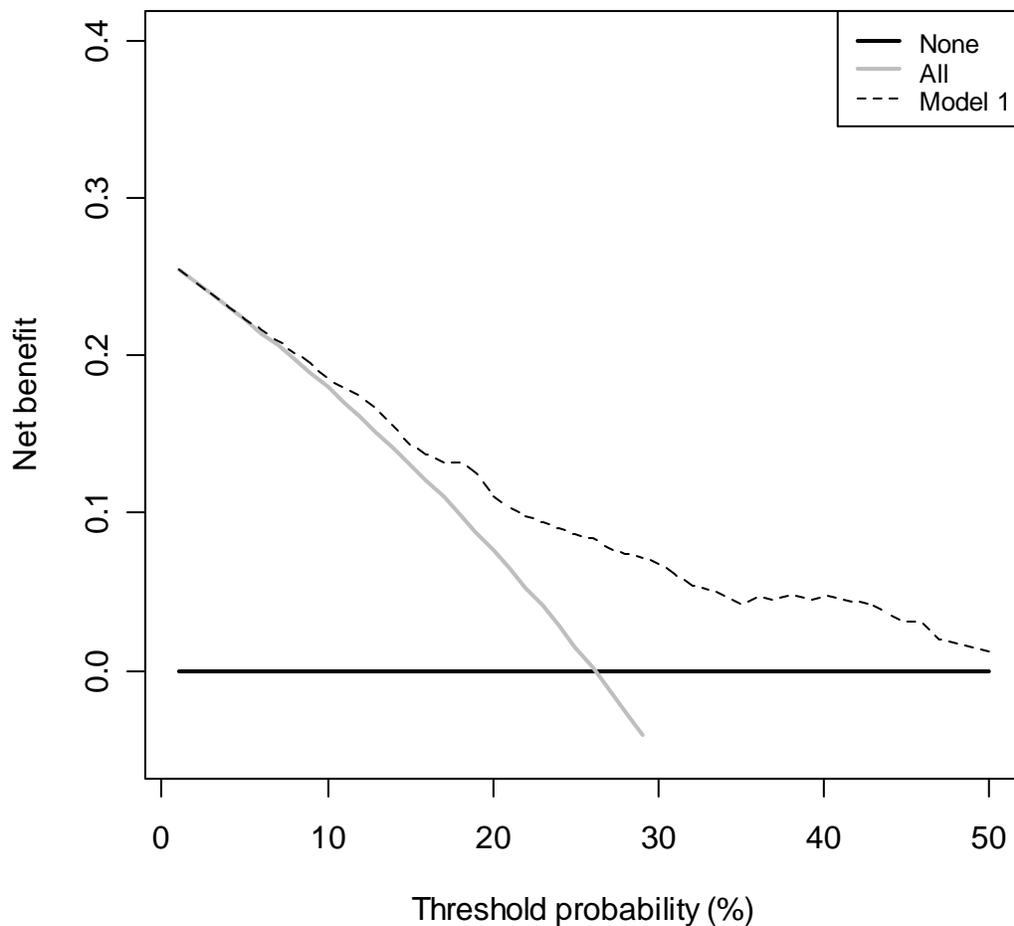


Bladder cancer data sets are subject to the competing risk of death. To run a competing risk analysis, we first create a failure variable that indicated patients who died before they recurred. Using the traditional approach, patients are coded 0 if they do not have an event, 1 if they have the event of interest before the competing event and 2 if they have the competing event before the event of interest.

```
firstevent <- recurrence + dead * (recurrence==0) * 2
```

We then run dca specifying the competing risk.

```
stdca(timepoint=60, time=ttlastfollowrecurrence, failure=firstevent,  
xmatrix=prednomogram, xstop=0.5, cmprsk="TRUE", ymax=0.4)
```



The competing risk model does not change our conclusion that the statistical model would improve clinical decision making for all decisions save those for which treatment would be recommended for even very low risks, such as 2%.

The code below shows how to get both the standard decision curve and the decision curves adjusted for competing risk on one graph. It is a good example of how graphs can be drawn by saving out results from *stdca.R* and then using the graphing commands.

```
# start with the standard Kaplan Meier model, saving the results matrix as "kmresult"
kmresult <- stdca(timepoint=60, time=ttlastfollowrecurrence,
  failure=recurrence, xmatrix=prednomogram, xstop=0.5)
```

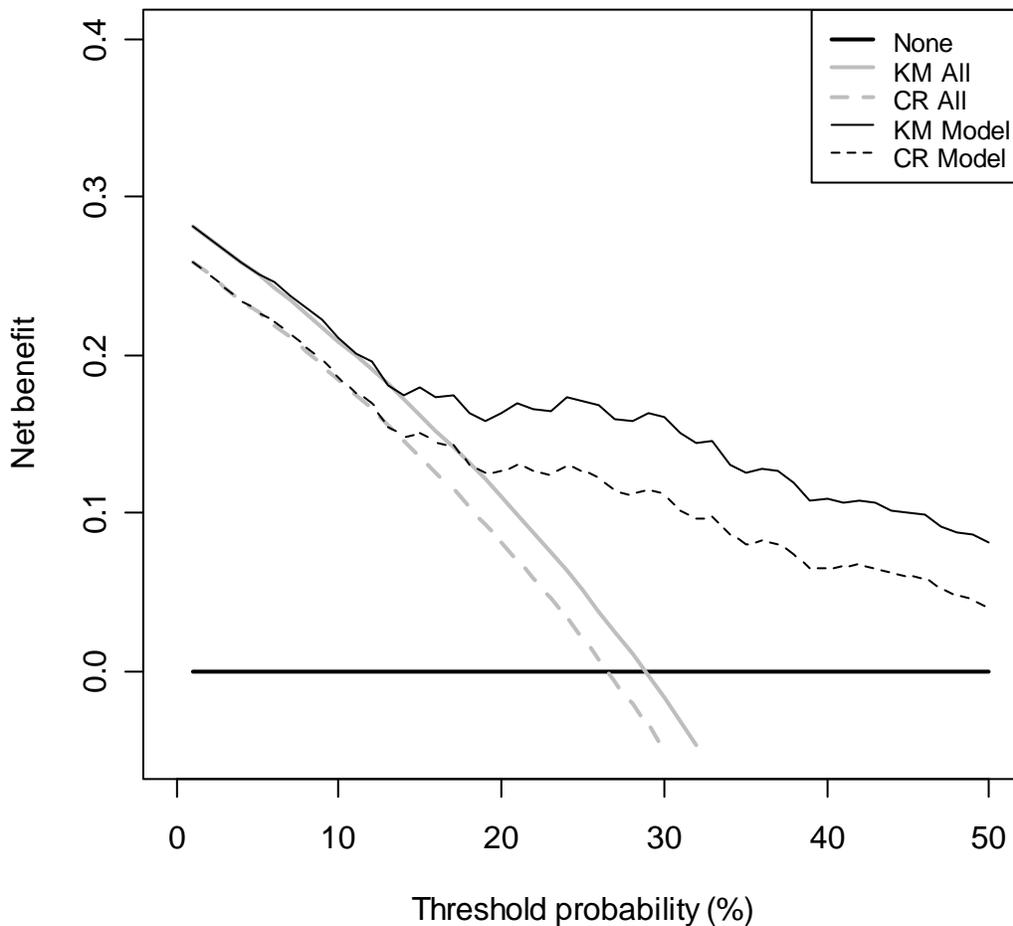
```
#now do the competing risk model, saving the results matrix as "crresult"
crresult <- stdca(timepoint=60, time=ttlastfollowrecurrence,
  failure=firstevent, xmatrix=prednomogram, xstop=0.5, cmprsk="TRUE")
```

```
#now create the graph
```

```

plot(kmresult$threshold, kmresult$none, type="l", lwd=2, xlim=c(0, 50),
     ylim=c(-0.05, 0.4), xlab="Threshold probability (%)", ylab="Net benefit")
lines(kmresult$threshold, kmresult$all, type="l", col=8, lwd=2)
lines(kmresult$threshold, crresult$all, type="l", col=8, lwd=2, lty=2)
lines(kmresult$threshold, kmresult$modelp1, type="l", col=1)
lines(crresult$threshold, crresult$modelp1, type="l", col=1, lty=2)
legend("topright", cex=0.8, legend=c("None", "KM All", "CR All", "KM Model",
  "CR Model"), col=c(17, 8, 8, 1, 1), lwd=c(2, 2, 2, 1, 1), lty=c(1, 1, 2, 1, 2))

```



Here the decision curves adjusting for competing risk are shown with dashes; the decision curves with patients censored at the time of death are shown as solid lines. Competing risk shifts all lines down, that is, estimates a lower net benefit, because censoring at the time of death overestimates the risk of death.

Applying smoothing to decision curves

As can be observed in the figures above, decision curves for some models are jagged, instead of smooth. These bumps reflect random error and are not clinically relevant. To improve the presentation of the decision curves, you can apply smoothing to the net benefits output from the `dca` command. As an example, the code below applies smoothing to the decision curve in the case of the molecular marker for cancer, applied only to patients with a clinical examination that was suspicious for cancer.

```
dataset1 <- read.delim("c:\dca_example_dataset1.txt", header = TRUE, sep = "\t")
attach(dataset1)
subset.suspicious <- clinical_exam=="suspicious for cancer"
# run the decision curve, saving out the results
dcaoutput <- dca(yvar=cancer[subset.suspicious], xmatrix= marker[subset.suspicious],
                prob="N", xstart=0.05, xstop=0.35, ymax=0.2)
# apply smoothing to the net benefit of the marker
smoothmodelp1 <- smooth(dcaoutput$modelp1, twiceit=TRUE)
# now plot the decision curve
plot(dcaoutput$threshold, dcaoutput$none, type="l", lwd=2, xlim=c(5, 35),
     ylim=c(-0.05, 0.2), xlab="Threshold probability (%)", ylab="Net benefit")
lines(dcaoutput$threshold, dcaoutput$all, type="l", col=8, lwd=2)
lines(dcaoutput$threshold, smoothmodelp1, type="l", col=1, lty=2)
legend("topright", cex=0.8, legend=c("None", "All", "Marker"),
     col=c(17, 8, 1, 2, 3), lwd=c(2, 2, 1), lty=c(1, 1, 2))
```

