432 Class 25 Slides

github.com/THOMASELOVE/432-2018

2018-04-17

Preliminaries

```
library(skimr)
library(rms)
library(survival)
library(OIsurv)
library(survminer)
library(aplore3) # for a data set
library(ResourceSelection) # for Hosmer-Lemeshow test
library(bestglm) # for a demonstration of all subsets
library(broom)
library(tidyverse)
```

Today's Agenda

- Regression on Time-to-event data
  - Cox Proportional Hazards Model
- Some Loose Ends

Survival Analysis / Cox Regression

A Survival Analysis Example

Source: Chen and Peace (2011) *Clinical Trial Data Analysis Using R*, CRC Press, section 5.1

```
brca <- read.csv("data/breast_cancer.csv") %>% tbl_df
```

The `brca` trial

The `brca` data describes a parallel randomized trial of three treats, adjuvant to surgery in the treat of patients with stage-2 carcinoma of the breast. The three treat groups are:

- `S+CT` = Surgery plus one year of chemotherapy
- `S+IT` = Surgery plus one year of immunotherapy
- `S+CT+IT` = Surgery plus one year of chemotherapy and immunotherapy

The measure of efficacy were "time to death" in weeks. In addition to `treat`, our variables are:

- `trial_weeks`: time in the study, in weeks, to death or censoring
- `last_alive`: 1 if alive at last follow-up (and thus censored), 0 if dead
- `age`: age in years at the start of the trial

brca tibble

```
# A tibble: 31 x 5
   subject treat   trial_weeks last_alive   age
   <fct>   <fct>         <int>      <int> <int>
 1 A01     S+CT            102          0    55
 2 A02     S+IT            192          0    62
 3 A03     S+CT+IT          73          0    72
 4 A04     S+CT             58          1    48
 5 A05     S+CT             48          1    26
 6 A06     S+IT            182          1    52
 7 A07     S+IT            196          1    50
 8 A08     S+CT            177          1    49
 9 A09     S+IT            191          1    62
10 A10     S+CT+IT          36          0    60
# ... with 21 more rows
```

Analytic Objectives

This is a typical right-censored survival data set with interest in the comparative analysis of the three treats.

1. Does immunotherapy added to surgery plus chemotherapy improve survival? (Comparing S+CT+IT to S+CT)
2. Does chemotherapy add efficacy to surgery plus immunotherapy? (S+CT+IT vs. S+IT)
3. What is the effect of age on survival?

Create survival object

- `trial_weeks`: time in the study, in weeks, to death or censoring
- `last_alive`: 1 if alive at last follow-up (and thus censored), 0 if dead

So `last_alive` = 0 if the event (death) occurs.

What's next?

Create survival object

- `trial_weeks`: time in the study, in weeks, to death or censoring
- `last_alive`: 1 if alive at last follow-up (and thus censored), 0 if dead

So `last_alive` = 0 if the event (death) occurs.

```
brca$S <- with(brca, Surv(trial_weeks, last_alive == 0))

head(brca$S)
```

```
[1] 102  192   73   58+  48+ 182+
```

Build Kaplan-Meier Estimator

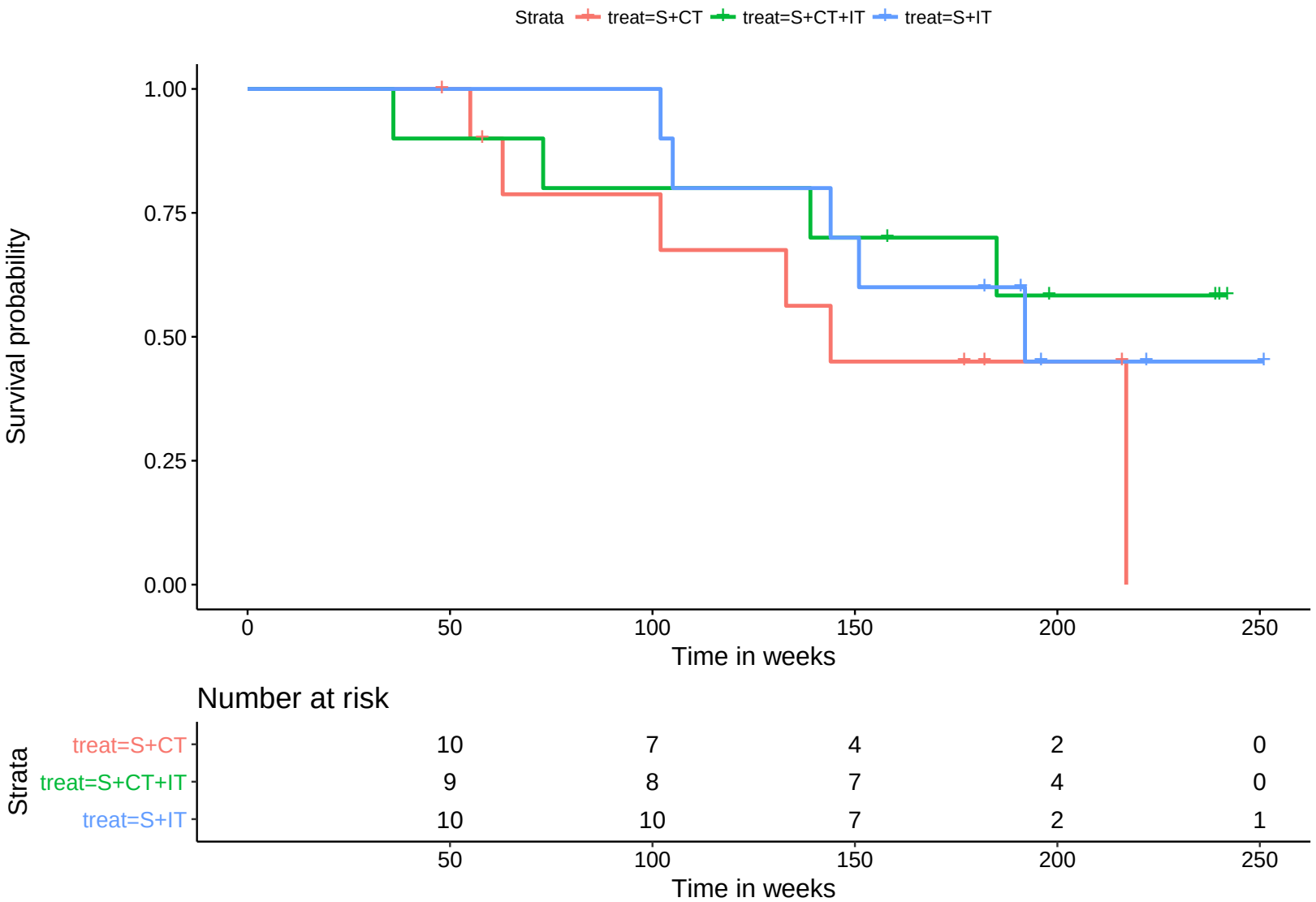
```
kmfit <- survfit(S ~ treat, dat = brca)

print(kmfit, print.rmean = TRUE)
```

```
Call: survfit(formula = S ~ treat, data = brca)

                n events *rmean *se(rmean) median 0.95LCL
treat=S+CT     11      6    153       21.1    144     102
treat=S+CT+IT  10      4    188       23.7     NA     139
treat=S+IT     10      5    188       17.9    192     144
               0.95UCL
treat=S+CT          NA
treat=S+CT+IT       NA
treat=S+IT          NA
    * restricted mean with upper limit =  242
```

summary(kmfit)

```
> summary(kmfit)
Call: survfit(formula = S ~ treat, data = brca)

                treat=S+CT
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   55     10       1    0.900  0.0949        0.732        1.000
   63      8       1    0.787  0.1340        0.564        1.000
  102      7       1    0.675  0.1551        0.430        1.000
  133      6       1    0.562  0.1651        0.316        1.000
  144      5       1    0.450  0.1660        0.218        0.927
  217      1       1    0.000     NaN           NA           NA

                treat=S+CT+IT
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   36     10       1    0.900  0.0949        0.732            1
   73      9       1    0.800  0.1265        0.587            1
  139      8       1    0.700  0.1449        0.467            1
  185      6       1    0.583  0.1610        0.340            1

                treat=S+IT
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
  102     10       1     0.90  0.0949        0.732        1.000
  105      9       1     0.80  0.1265        0.587        1.000
  144      8       1     0.70  0.1449        0.467        1.000
  151      7       1     0.60  0.1549        0.362        0.995
  192      4       1     0.45  0.1743        0.211        0.961
```

K-M Plot via survminer



K-M Plot via survminer (code)

```
ggsurvplot(kmfit, data = brca,
           risk.table = TRUE,
           risk.table.height = 0.25,
           xlab = "Time in weeks")
```

Testing the difference between curves

```
survdiff(S ~ treat, dat = brca)
```

```
Call:
survdiff(formula = S ~ treat, data = brca)

                N Observed Expected (O-E)^2/E (O-E)^2/V
treat=S+CT     11        6     3.80    1.2772    1.7647
treat=S+CT+IT 10        4     5.62    0.4676    0.7725
treat=S+IT     10        5     5.58    0.0605    0.0981

 Chisq= 1.9  on 2 degrees of freedom, p= 0.393
```

What do we conclude?

Fit Cox Model A: Treatment alone

```
modA <- coxph(S ~ treat, data = brca)
modA
```

```
Call:
coxph(formula = S ~ treat, data = brca)

                coef exp(coef) se(coef)     z    p
treatS+CT+IT -0.831     0.435    0.655 -1.27 0.20
treatS+IT    -0.583     0.558    0.609 -0.96 0.34

Likelihood ratio test=1.75  on 2 df, p=0.416
n= 31, number of events= 15
```

summary(modA)

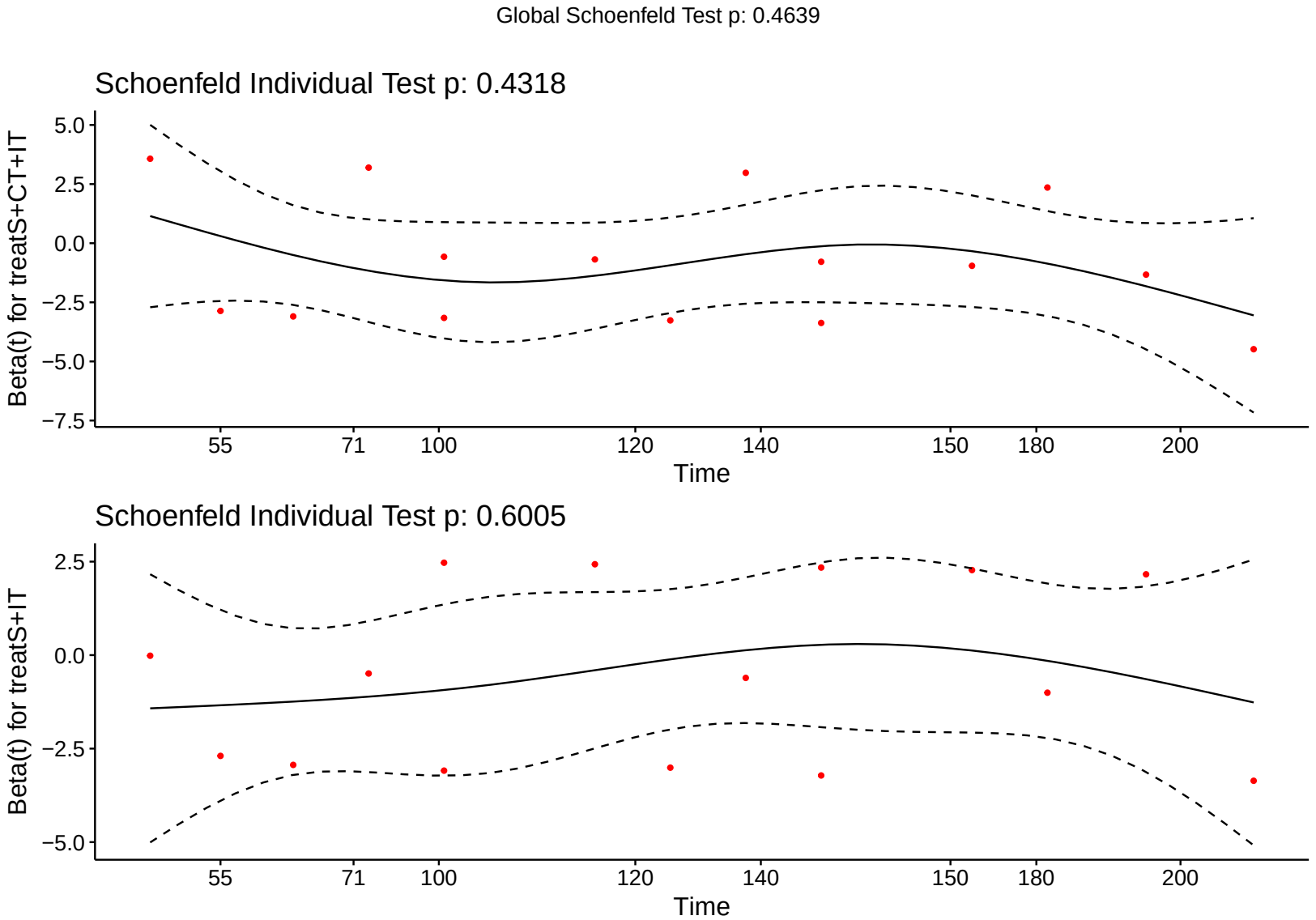
```
> summary(modA)
Call:
coxph(formula = S ~ treat, data = brca)

  n= 31, number of events= 15

                coef exp(coef) se(coef)      z Pr(>|z|)
treatS+CT+IT -0.8313    0.4355   0.6547 -1.270    0.204
treatS+IT    -0.5832    0.5581   0.6088 -0.958    0.338

             exp(coef) exp(-coef) lower .95 upper .95
treatS+CT+IT    0.4355      2.296    0.1207     1.571
treatS+IT       0.5581      1.792    0.1692     1.840

Concordance= 0.577  (se = 0.078 )
Rsquare= 0.055   (max possible= 0.944 )
Likelihood ratio test= 1.75  on 2 df,   p=0.4164
Wald test            = 1.82  on 2 df,   p=0.403
Score (logrank) test = 1.89  on 2 df,   p=0.3878
```

Check Proportional Hazards Assumption

```
cox.zph(modA)
```

```
                rho chisq     p
treatS+CT+IT -0.198 0.618 0.432
treatS+IT     0.138 0.274 0.601
GLOBAL           NA 1.536 0.464
```

Graphical PH Test `ggcoxzph(cox.zph(modA))`



Fit Cox Model B: Treatment + Age

```
modB <- coxph(S ~ treat + age, data = brca)
modB
```

```
Call:
coxph(formula = S ~ treat + age, data = brca)

                coef exp(coef) se(coef)     z     p
treatS+CT+IT -0.5996    0.5490   0.6574 -0.91 0.362
treatS+IT    -0.3116    0.7323   0.6094 -0.51 0.609
age           0.0781    1.0812   0.0367  2.13 0.034

Likelihood ratio test=6.99  on 3 df, p=0.0722
n= 31, number of events= 15
```

summary(modB)

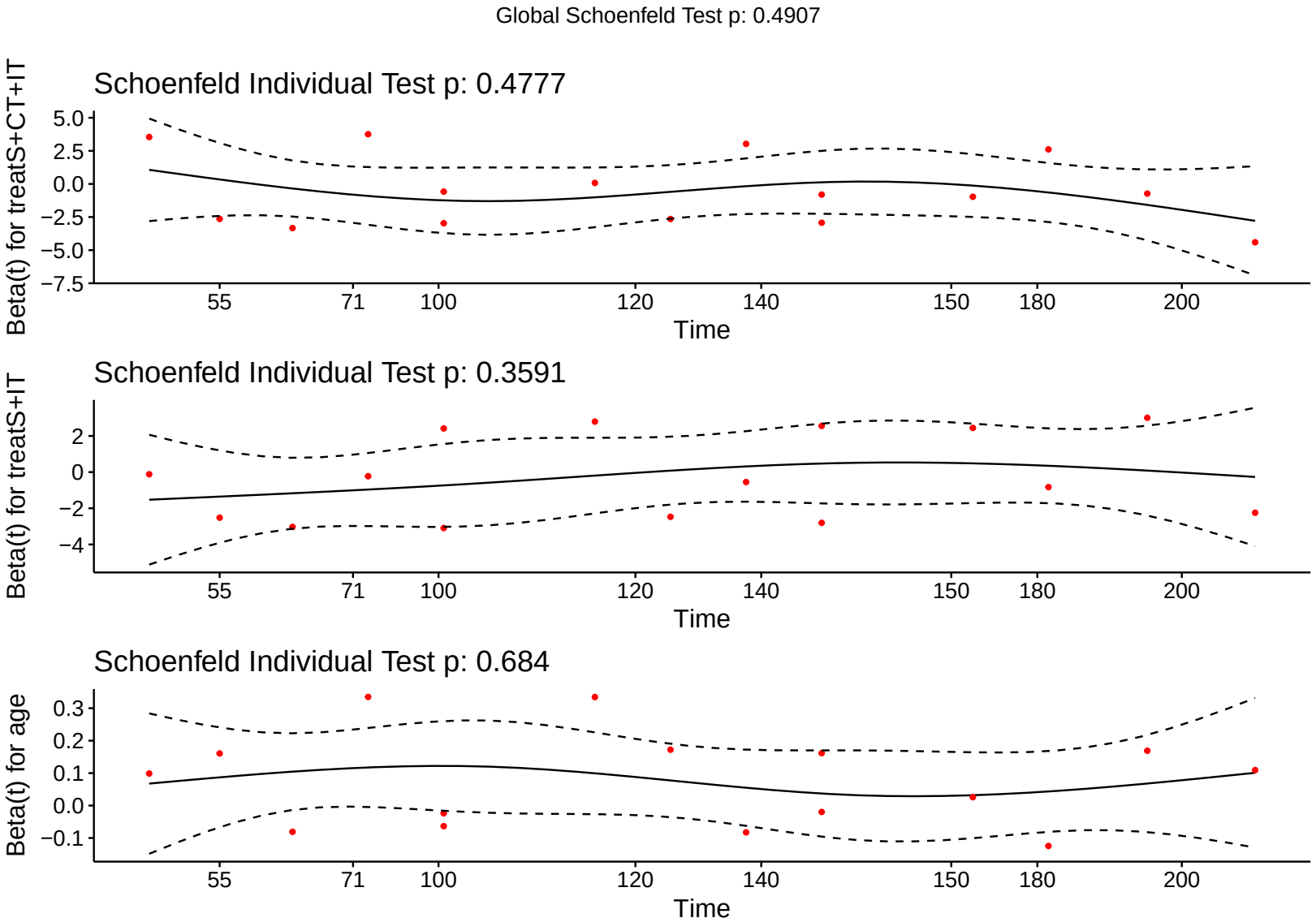
```
> summary(modB)
Call:
coxph(formula = S ~ treat + age, data = brca)

  n= 31, number of events= 15

                coef exp(coef) se(coef)      z Pr(>|z|)
treatS+CT+IT -0.59960   0.54903  0.65741 -0.912   0.3617
treatS+IT    -0.31161   0.73227  0.60936 -0.511   0.6091
age           0.07807   1.08119  0.03672  2.126   0.0335 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

             exp(coef) exp(-coef) lower .95 upper .95
treatS+CT+IT    0.5490     1.8214    0.1514     1.992
treatS+IT       0.7323     1.3656    0.2218     2.417
age             1.0812     0.9249    1.0061     1.162

Concordance= 0.701  (se = 0.083 )
Rsquare= 0.202   (max possible= 0.944 )
Likelihood ratio test= 6.99  on 3 df,   p=0.07224
Wald test            = 5.85  on 3 df,   p=0.1192
Score (logrank) test = 6.15  on 3 df,   p=0.1043
```

Proportional Hazards Assumption: Model B Check

```
cox.zph(modB)
```

```
                rho chisq     p
treatS+CT+IT -0.179 0.504 0.478
treatS+IT     0.244 0.841 0.359
age          -0.106 0.166 0.684
GLOBAL           NA 2.416 0.491
```

Graphical PH Test `ggcoxzph(cox.zph(modB))`



What to do if the PH assumption is violated

- If the PH assumption fails on a categorical predictor, fit a Cox model stratified by that predictor (use `strata(var)` rather than `var` in the specification of the `coxph` model.)
- If the PH assumption is violated, this means the hazard isn't constant over time, so we could fit separate Cox models for a series of time intervals.
- Use an extension of the Cox model that permits covariates to vary over time.

Visit https://cran.r-project.org/web/packages/survival/vignettes/timedep.pdf for details on building the relevant data sets and models, with examples.

Some Loose Ends

On Fitting Regression Models

Some things are always true. . .

- Being honest about your findings is important.
- Identifying a stable phenomenon to study is crucial.
- Measuring that phenomenon well is crucial.
- Being transparent about your work is important. Describing your work so it can be replicated, and then actually replicating it, are good things. Sharing your data and your materials is a good idea.
- Having a large sample size (n) is helpful in fitting models.

but the impact of lots of other things changes depending on why you're fitting a regression model.

The Key Point

Decide what your research question is, and use it to help you think about what's important in your modeling.

- Models that account for only a few of the possibly important dimensions of a problem don't lead to causal conclusions, but can help screen out important from less important areas for future work.
- Model development strategies that work well with large sample sizes (n) and small numbers of predictors to consider don't necessarily work well when the situation is reversed.
- Most problems involve missing data, and problems in measurement. Getting those issues settled effectively is often overlooked.
- The sample size you need to fit a regression model changes depending on your aims.

On Fitting Models

What are some of the reasons you might fit a linear (or generalized linear) model?

1. (All prediction, no explanations) Because you want to make predictions about an outcome in new data based on some training data you have, but you're happy to take those predictions as emerging from a mysterious magic "black box" that cannot be peered into without spoiling the surprise. You don't care if your results are a little biased, so long as the predictions are strong. Parsimony doesn't matter to you.

- Some especially useful tools here include: variable (feature) selection through cross-validation, stepwise approaches, AIC and BIC, machine learning tools like regression trees, and other means of quickly searching through many possible models.
- Sample size is rarely a big issue here. The big problem is having more variables than you can possibly plot at once. You usually have enough data to partition into separate development and test samples.

On Fitting Models

2. (All description, external validity is irrelevant) Because you want to describe, as accurately as possible, the nature of the associations you observe in the available data, but you don't care much at all whether the conclusions you draw will hold up in new data.

- Confidence intervals for coefficients (slopes, mostly), and sometimes you'll run the model on clinically relevant cutpoints rather than continuous predictors to see what's happening more simply. Simple polynomial models can be appealing, and you'll sometimes want to build this in the ANCOVA context, where you're looking for the impact of specific pre-specified interactions.
- Residual plots play a big role here in deciding whether the model "fits" well enough, or identifying cases when it doesn't.
- Cross-validation is useful, but not a big part of model selection or convincing people that the model is "right" or not.

On Fitting Models

3. (Clinical prediction) You want to do an excellent job predicting an outcome in new data, but you have a lot of prior knowledge about the predictors under consideration, and want to use that information to help produce prediction rules as effectively as possible. You welcome the fact that most relationships are non-linear, but would like to be parsimonious if possible, as data are often expensive.

- Some especially useful tools here include: scatterplot matrices (when the number of predictors is modest), cross-validation, assessments of discrimination and calibration, Spearman's ρ^2 plot to point the way to non-linear terms that might be impactful if present, restricted cubic splines, polynomial functions, and graphical tools like nomograms
- Most stepwise tools aren't helpful here. We try to not "peek" at the outcome-predictor relationships to maintain unbiased estimates of the relationship without extensive validation.

On Fitting Models

4. (Above all else, simplicity) You want the problem to look like one in a statistics textbook, where everything is fit with the simplest possible model, where every term adds statistically significant predictive value, and where obtaining an unbiased estimate of the outcome is especially important. You still care a bit about what happens in new data, but you're mostly concerned about parsimonious model development.

- Some especially useful tools include best subsets, stepwise approaches, and methods for pruning a set of predictors with clustering or principal components analysis. These models usually make the (often incorrect) assumption that relationships are linear.
- Often this approach is used by people who are trying to pre-specify their entire model in advance, and want to be sure they can "explain" the result when they are done. That may not be a reasonable thing to hope for. This is a place where the "rule of 20" can be very helpful in setting expectations in advance.

On Fitting Models

5. (Causal inference) You want to identify whether a particular causal pathway you have pre-specified matches up well with what you see in new data.
6. (Risk Adjustment) You want to identify the impact of a particular exposure/predictor on an outcome, while controlling for the effects of a series of additional predictors. Perhaps you've done a randomized experiment / clinical trial, and want to identify whether particular results meet a standard for statistical (as well as clinical) significance. Power is very important.

- In either case, bias is very important, and you want to avoid it. Careful design of a comparison group (like I teach in 500) is a very good way to go about this work, but it's also true that there's a lot of epidemiology that goes into drawing causal conclusions, or even thinking hard about an association.
- Often the details of modeling take a back seat here to the details of designing the study (and the comparison groups) in the first place.

Using Restricted Cubic Splines

Explaining a Model with a Restricted Cubic Spline

Restricted cubic splines are an easy way to include an explanatory variable in a smooth and non-linear fashion in your model.

- The number of knots, k, are specified in advance, and this is the key issue to determining what the spline will do. We could use AIC to select k, or follow the general idea that for small n, k should be 3, for large n, k should be 5, and so often k = 4.
- The location of those knots is not important in most situations, so R places knots by default where the data exist, at fixed quantiles of the predictor's distribution.
- The "restricted" piece means that the tails of the spline (outside the outermost knots) behave in a linear fashion.

The "Formula" from a Model with a Restricted Cubic Spline

- The best way to demonstrate what a spline does is to draw a picture of it. When in doubt, do that: show us how the spline affects the predictions made by the model.
- But you can get a model equation for the spline out of R (heaven only knows what you would do with it.) Use the `latex` function in the `rms` package, for instance.

An Example

```
d <- datadist(iris)
options(datadist = "d")
m1 <- ols(Sepal.Length ~ rcs(Petal.Length, 4) + Petal.Width,
          data = iris, x = TRUE, y = TRUE)
m1
```

```
Linear Regression Model

 ols(formula = Sepal.Length ~ rcs(Petal.Length, 4) + Petal.Wid
     data = iris, x = TRUE, y = TRUE)

                 Model Likelihood     Discrimination
                    Ratio Test           Indexes
 Obs      150    LR chi2    253.23    R2        0.815
 sigma0.3609    d.f.            4    R2 adj    0.810
 d.f.     145    Pr(> chi2) 0.0000    g         0.844
```

Function(m1)

```
Function(m1)
```

```
function (Petal.Length = 4.35, Petal.Width = 1.3)
{
    4.7226352 + 0.24335435 * Petal.Length + 0.021780541 * pmax
        1.3, 0)^3 - 0.037888523 * pmax(Petal.Length - 3.33, 0)
        0.00031123969 * pmax(Petal.Length - 4.8, 0)^3 + 0.0157
        pmax(Petal.Length - 6.1, 0)^3 - 0.33400958 * Petal.Wid
}
<environment: 0x0000000029b9ba70>
```

What's in `Function(m1)`?

```
4.72 + 0.243  * Petal.Length
     + 0.022  * pmax( Petal.Length-1.3,  0)^3
     - 0.038  * pmax( Petal.Length-3.33, 0)^3
     + 0.0003 * pmax( Petal.Length-4.8,  0)^3
     + 0.016  * pmax( Petal.Length-6.1,  0)^3
     - 0.334  * Petal.Width
```

where `pmax` is the maximum of the arguments inside its parentheses.

Asssessing the Quality of a Logistic Regression Model

A Quick Example

SOURCE: Hosmer and Lemeshow (2000) Applied Logistic Regression: Second Edition. These data are copyrighted by John Wiley & Sons Inc. and must be acknowledged and used accordingly. Data were collected at Baystate Medical Center, Springfield, Massachusetts during 1986.

```
# uses aplore3 package for data set
lbw <- aplore3::lowbwt
head(lbw,3)
```

```
  id      low age lwt  race smoke  ptl  ht  ui       ftv
1  4 < 2500 g  28 120 Other   Yes  One  No Yes      None
2 10 < 2500 g  29 130 White    No None  No Yes Two, etc.
3 11 < 2500 g  34 187 Black   Yes None Yes  No      None
   bwt
1  709
2 1021
3 1135
```

Fit a logistic regression model

```
model_10 <- glm(low ~ lwt + ptl + ht,
                data = lbw, family = binomial)
model_10
```

```
Call:  glm(formula = low ~ lwt + ptl + ht, family = binomial,

Coefficients:
 (Intercept)          lwt        ptlOne  ptlTwo, etc.
     1.17016      -0.01851       1.74219        0.15105
       htYes
     1.91234

Degrees of Freedom: 188 Total (i.e. Null);  184 Residual
Null Deviance:      234.7
Residual Deviance: 207.4    AIC: 217.4
```

What is this model predicting, exactly?

```
levels(lbw$low)
```

```
[1] ">= 2500 g" "< 2500 g"
```

```
lbw %>% count(low)
```

```
# A tibble: 2 x 2
  low            n
  <fct>      <int>
1 >= 2500 g    130
2 < 2500 g      59
```

The model predicts the probability of a LOW birth weight, because $<$ 2500 g is listed second here.

- Our `model_10` is a model fit to y $=$ 1 when low $<$ 2500 g
- If y $=$ 1 indicated that low $>=$ 2500 g, this would be the opposite of our `model_10`.

Proving the direction of model_10

```
lbw <- lbw %>% mutate(y1 = ifelse(low == "< 2500 g", 1, 0),
                      y2 = ifelse(low == ">= 2500 g", 1, 0))
mod_1 <- glm(y1 ~ lwt + ptl + ht,
                    data = lbw, family = binomial)
mod_2 <- glm(y2 ~ lwt + ptl + ht,
                    data = lbw, family = binomial)
```

- mod_1 predicts Pr(birth weight $<$ 2500 g)
- mod_2 predicts Pr(birth weight $>=$ 2500 g)

So, what does `model_10` predict?

- `mod_1` predicts Pr(birth weight $<$ 2500 g)
- `mod_2` predicts Pr(birth weight $>=$ 2500 g)

```
head(fitted(mod_1),3)
```

```
        1         2         3
0.6661398 0.2250375 0.4062585
```

```
head(fitted(mod_2),3)
```

```
        1         2         3
0.3338602 0.7749625 0.5937415
```

```
head(fitted(model_10),3)
```

```
        1         2         3
0.6661398 0.2250375 0.4062585
```

Classification Table for this Model

```
table(fitted(model_10) >= 0.5, lbw$low)
```

```
       >= 2500 g < 2500 g
FALSE        123        39
TRUE           7        20
```

The Hosmer-Lemeshow Test of Goodness of Fit

See this link from Jonathan Bartlett.

- The Hosmer-Lemeshow goodness of fit test is based on dividing the sample into g groups (g is often chosen to be 10) according to their predicted probabilities.
- We then calculate the expected number of Y = 1 outcomes (based on the model probabilities of Y = 1 in the group) in each of the 10 groups, and compare those results to what is observed in the data using a Pearson goodness of fit statistic.
- A significant result indicates statistically significant "lack of fit" but it's easy to criticize the test (among other things, if you change the choice of g, you can materially change the *p* value.)

Running the Hosmer-Lemeshow Test

```
# requires ResourceSelection package
hos <- hoslem.test(model_10$y, fitted(model_10), g = 10)
hos
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  model_10$y, fitted(model_10)
X-squared = 5.6459, df = 8, p-value = 0.6868
```

So there's no evidence of poor fit. We can also get a table of observed vs. expected.

Observed vs. Expected Table from H-L test

```
cbind(hos$observed, hos$expected)
```

	y0	y1	yhat0	yhat1
[0.0305,0.14]	16	3	17.226390	1.773610
(0.14,0.2]	16	4	16.600448	3.399552
(0.2,0.225]	20	4	18.773041	5.226959
(0.225,0.252]	12	2	10.599250	3.400750
(0.252,0.263]	16	4	14.817304	5.182696
(0.263,0.288]	14	3	12.265960	4.734040
(0.288,0.322]	12	6	12.521738	5.478262
(0.322,0.384]	11	8	12.395653	6.604347
(0.384,0.637]	7	12	9.658303	9.341697
(0.637,0.947]	6	13	5.141914	13.858086

Change the number of groups, g?

```
for (i in 5:15) {
  print(hoslem.test(model_10$y,
                    fitted(model_10), g=i)$p.value)
}
```

```
[1] 0.288379
[1] 0.1888862
[1] 0.6682719
[1] 0.7830653
[1] 0.6564895
[1] 0.6868256
[1] 0.4444073
[1] 0.4520327
[1] 0.6381505
[1] 0.6437715
[1] 0.07773936
```

Can we use "best subsets" or "all subsets" in logistic regression?

Using the `bestglm` package to do "best subsets" with logistic models

There are several available tools. If you're interested, I'd start with this link.

- The `bestglm` package does an exhaustive (all subsets) search through a glm (slowly) using AIC, BIC or cross-validation, for example.
- This expects the data to be in a certain form, for instance, the outcome must be named *y* and you can have no extraneous variables present.
- The tutorial you'll find at the link above is pre-tidyverse. I don't know how well `bestglm` works with the tidyverse, but it's not likely to be much worse than `leaps` does.
- This approach can also be used with linear models which include multi-categorical predictors.
- `bestglm` is going to do an exhaustive search, which will be slow with large data sets, or large pools of predictors.
- In addition to `bestglm` there are several other appealing tools for looking at large numbers of potential models quickly.

Preparing the Data Set

```
lbw_for_bestglm <- lbw %>%
    mutate(y = low) %>%
    select(age, lwt, race, smoke, ptl, ht, ui, ftv, y)
```

Must include only the variables we want to include in the search, and then the outcome, which must be labeled y, in that order.

`lbw_for_bestglm` data: First few observations

```
head(lbw_for_bestglm)
```

```
  age lwt  race smoke  ptl  ht  ui        ftv         y
1  28 120 Other   Yes  One  No Yes       None < 2500 g
2  29 130 White    No None  No Yes Two, etc. < 2500 g
3  34 187 Black   Yes None Yes  No       None < 2500 g
4  25 105 Other    No  One Yes  No       None < 2500 g
5  25  85 Other    No None  No Yes       None < 2500 g
6  27 150 Other    No None  No  No       None < 2500 g
```

Search through all subsets, with AIC as criterion

```
res.best.logistic <-
    bestglm(Xy = lbw_for_bestglm,
            family = binomial,
            IC = "AIC",
            method = "exhaustive")
```

```
Morgan-Tatar search since family is non-gaussian.
Note: factors present with more than 2 levels.
```

This approach (with a gaussian family) can also be used for OLS.

Show top 5 models chosen by AIC

```
res.best.logistic$BestModels
```

```
    age  lwt  race smoke  ptl   ht    ui   ftv Criterion
1 FALSE TRUE  TRUE  TRUE TRUE TRUE  TRUE FALSE  211.1647
2  TRUE TRUE  TRUE  TRUE TRUE TRUE  TRUE FALSE  211.9063
3 FALSE TRUE  TRUE  TRUE TRUE TRUE FALSE FALSE  212.2697
4  TRUE TRUE  TRUE  TRUE TRUE TRUE FALSE FALSE  212.6828
5  TRUE TRUE FALSE FALSE TRUE TRUE  TRUE FALSE  213.4539
```

Show result for best model, according to AIC

```
> summary(res.best.logistic$BestModel)

Call:
glm(formula = y ~ ., family = family, data = Xi, weights = weights)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8643  -0.7803  -0.5172   0.9308   2.2013

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.015463   0.982536  -0.016  0.98744
lwt          -0.016773   0.007081  -2.369  0.01786 *
raceBlack     1.250947   0.534975   2.338  0.01937 *
raceOther     0.827222   0.446817   1.851  0.06412 .
smokeYes      0.875302   0.410139   2.134  0.03283 *
ptlOne        1.463051   0.506699   2.887  0.00388 **
ptlTwo, etc. -0.163819   0.945560  -0.173  0.86245
htYes         1.875815   0.716529   2.618  0.00885 **
uiYes         0.828436   0.466987   1.774  0.07606 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 195.16  on 180  degrees of freedom
AIC: 213.16

Number of Fisher Scoring iterations: 4
```

Search all subsets, using BIC instead of AIC

```
res.best.logistic2 <-
    bestglm(Xy = lbw_for_bestglm,
            family = binomial,
            IC = "BIC",
            method = "exhaustive")
```

```
Morgan-Tatar search since family is non-gaussian.
Note: factors present with more than 2 levels.
```

Show top 5 models by BIC

```
res.best.logistic2$BestModels
```

```
    age   lwt  race smoke  ptl    ht    ui   ftv Criterion
1 FALSE  TRUE FALSE FALSE TRUE  TRUE FALSE FALSE  228.3477
2 FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  230.1914
3  TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  230.2738
4 FALSE  TRUE FALSE FALSE TRUE FALSE FALSE FALSE  230.3406
5 FALSE  TRUE FALSE FALSE TRUE  TRUE  TRUE FALSE  230.3604
```

Show result for best model, according to BIC

```
> summary(res.best.logistic2$BestModel)

Call:
glm(formula = y ~ ., family = family, data = Xi, weights = weights)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7662  -0.7929  -0.6853   0.9083   2.2843

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.170161   0.870866   1.344 0.179054
lwt         -0.018513   0.006957  -2.661 0.007788 **
ptlOne       1.742193   0.486423   3.582 0.000341 ***
ptlTwo, etc. 0.151047   0.905990   0.167 0.867590
htYes        1.912342   0.734647   2.603 0.009239 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 207.38  on 184  degrees of freedom
AIC: 217.38

Number of Fisher Scoring iterations: 4
```

Best Models, by Criterion

There are 8 predictors under consideration.

- `age`, `lwt`, `race`, `smoke`, `ptl`, `ht`, `ui` and `ftv`

Model (by AIC)	Rank	Model (by BIC)
`lwt`, `smoke`, `ptl`, `ht`, `ui`	1	`lwt`, `ptl`, `ht`
all except `ftv`	2	`ptl`
`lwt`, `race`, `smoke`, `ptl`, `ht`	3	`age`, `ptl`
all except `ui`, `ftv`	4	`lwt`, `ptl`
`age`, `lwt`, `ptl`, `ht`, `ui`	5	`lwt`, `ptl`, `ht`, `ui`

Search all subsets, using Cross-Validation

Can be done when all predictors are continuous or binary, but **not** if you have categorical predictors with more than two levels.

```
# not run here because we have
# some multi-categorical predictors
set.seed(432)
res.best.logistic_cv <-
    bestglm(Xy = lbw_for_bestglm,
            family = binomial,
            IC = "CV")
```

Next Time

- Retrospective Power and why most smart folks avoid it
  - Type S and Type M error: Saying something more useful
- Replicable Research and the Crisis in Science
  - ASA Statement on P values
  - Is changing the p value cutoff the right strategy?
  - Second-generation p values: A next step?