Interactive and data adaptive model selection with mplot

Garth Tarr
University of Newcastle
December 2015

Get it on Github

```
install.packages("devtools")
devtools::install_github("garthtarr/mplot")
require(mplot)
```

… or get it on CRAN

```
install.packages("mplot")
```

Main functions

- `vis()` for **variable inclusion** and **model stability** plots
- `af()` for the **adaptive fence**
- `bglmnet()` for **bootstrapping glmnet**
- `mplot()` for an interactive **shiny interface**

Current state of variable selection

Google

- 2 million pages with "model selection"
- 1/2 million pages with "variable selection"

Google Scholar

- 860,000 articles with "model selection"
- 200,000 articles with "variable selection"

Do we really need more?

A stability based approach

Aim: To provide scientists/researchers/analysts with tools that give them more information about the model selection choices that they are making.

Concept of **model stability** independently introduced by Meinshausen and Bühlmann (2010) and Müller and Welsh (2010) for different linear regression situations.

Key idea: small changes should have small effects

A smörgåsbord of tuning parameters…

Information Criterion

- Generalised IC: $\mathrm{GIC}(\alpha; \lambda) = -2 \times \mathrm{LogLik}(\alpha) + \lambda p_\alpha$

With important special cases:

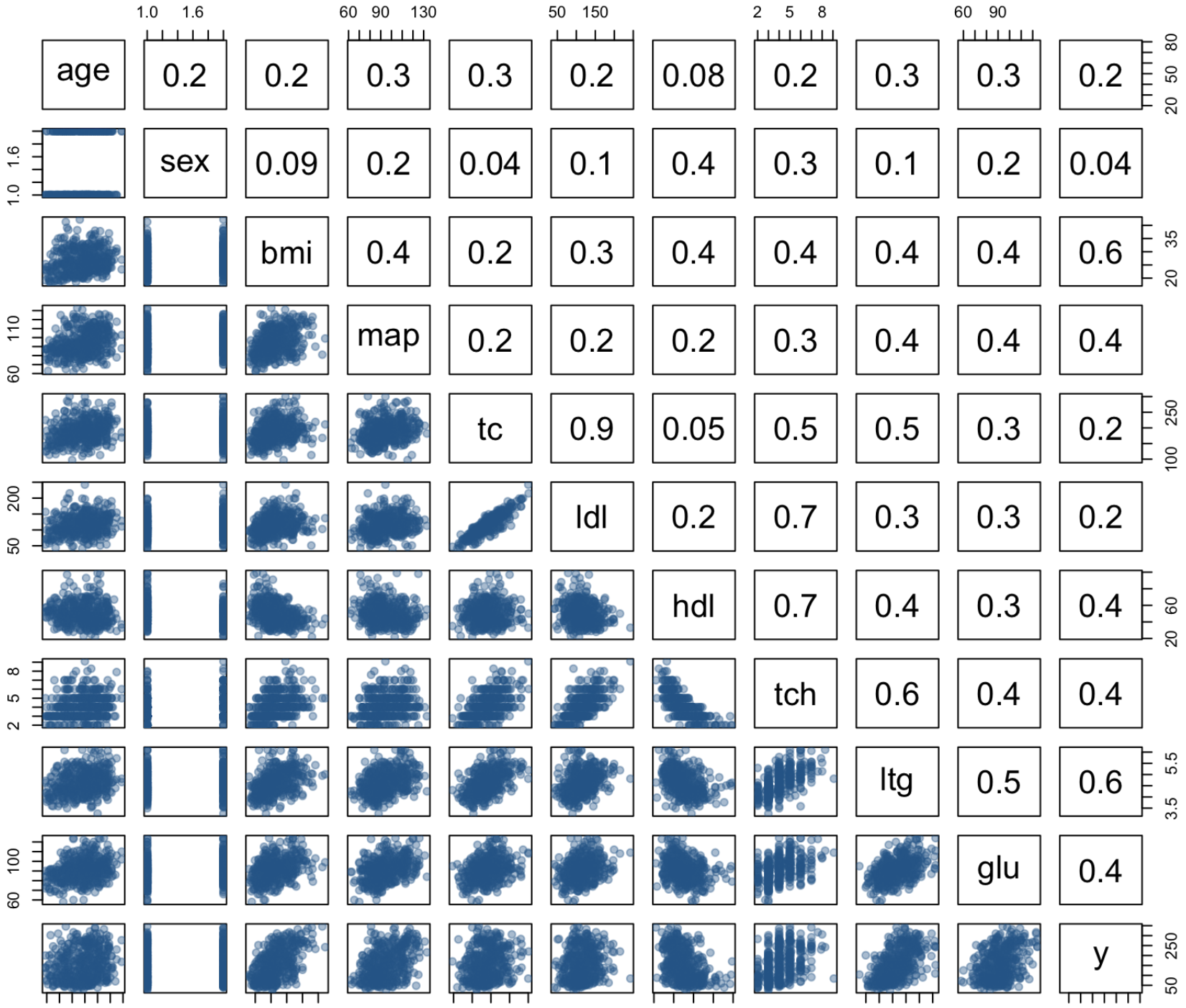
- **AIC**: $\lambda = 2$
- **BIC**: $\lambda = \log(n)$

Regularisation routines

- **Lasso**: minimises $-\mathrm{LogLik}(\alpha) + \lambda \, ||\beta_\alpha||_1$
- Many variants of the lasso, SCAD,…

Diabetes data

Variable	Description
age	Age
sex	Gender
bmi	Body mass index
map	Mean arterial pressure (average blood pressure)
tc	Total cholesterol (mg/dL)
ldl	Low-density lipoprotein ("bad" cholesterol)
hdl	High-density lipoprotein ("good" cholesterol)
tch	Blood serum measurement
ltg	Blood serum measurement
glu	Blood serum measurement (glucose?)
y	A quantitative measure of disease progression one year after baseline


age
sex
bmi
map
tc
ldl
hdl
tch
ltg
glu
y
0.2 0.2 0.3 0.3 0.2 0.08 0.2 0.3 0.3 0.2
0.09 0.2 0.04 0.1 0.4 0.3 0.1 0.2 0.04
0.4 0.2 0.3 0.4 0.4 0.4 0.4 0.6
0.2 0.2 0.2 0.3 0.4 0.4 0.4
0.9 0.05 0.5 0.5 0.3 0.2
0.2 0.7 0.3 0.3 0.2
0.7 0.4 0.3 0.4
0.6 0.4 0.4
0.5 0.6
0.4
1.0 1.6
60 90 130
50 150
2 5 8
60 90
20 50 80
20 35
60 110
100 250
50 200
20 60
2 5 8
3.5 5.5
60 100
50 250

Variable inclusion plots

Variable inclusion plots

Aim: To visualise **inclusion probabilities** as a function of the penalty multiplier $\lambda \in [0, 2\log(n)]$.

Procedure

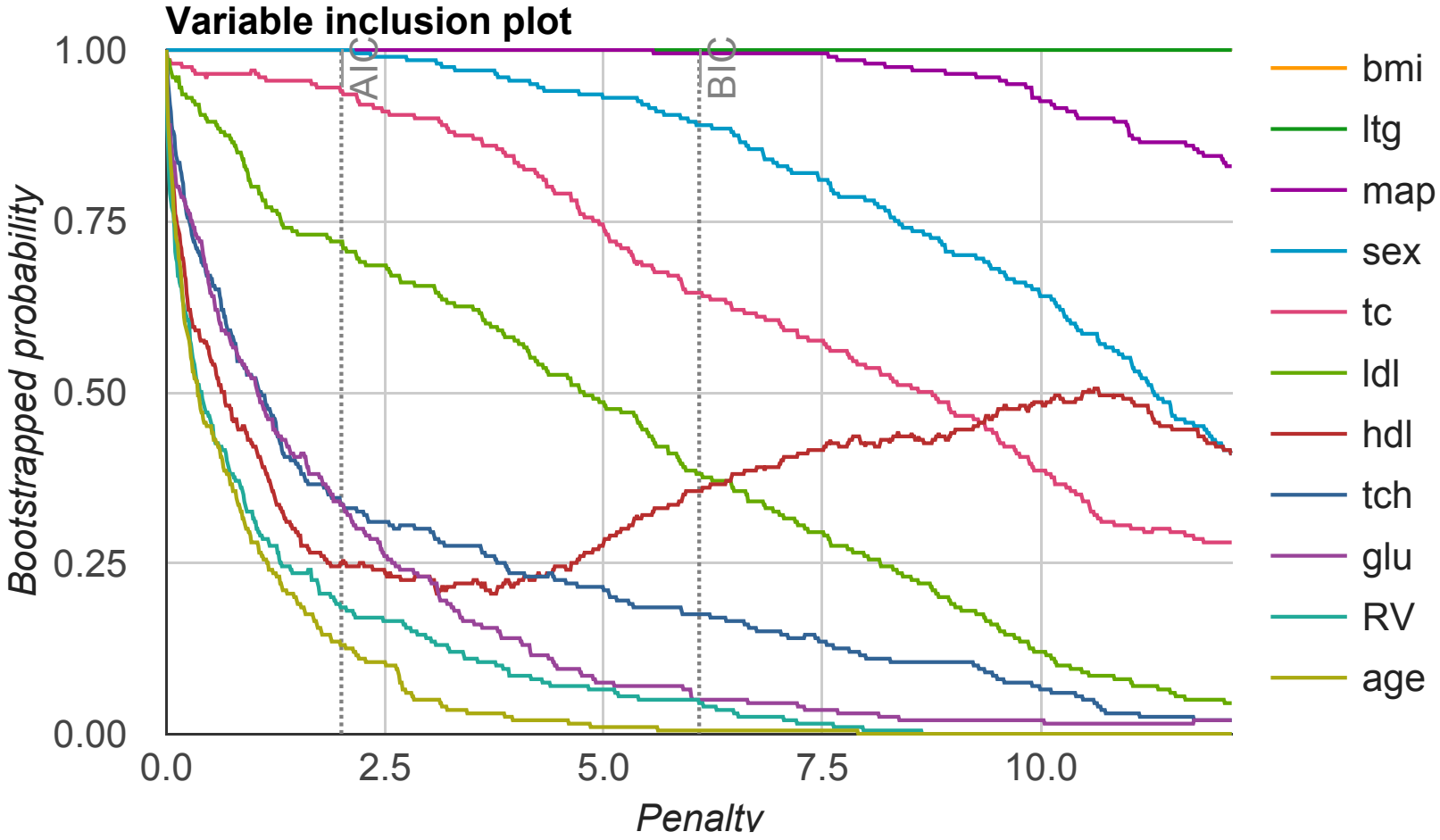
1. Calculate (weighted) bootstrap samples $b = 1, \ldots, B$.
2. For each bootstrap sample, at each λ value, find $\hat{\alpha}_\lambda^{(b)} \in \mathcal{A}$ as the model with smallest $\mathrm{GIC}(\alpha; \lambda) = -2 \times \mathrm{LogLik}(\alpha) + \lambda p_\alpha$.
3. The inclusion probability for variable x_j is estimated as $\frac{1}{B}\sum_{b=1}^{B} 1\{j \in \hat{\alpha}_\lambda^{(b)}\}$.

References

- Müller and Welsh (2010) for linear regression models
- Murray, Heritier, and Müller (2013) for generalised linear models

Diabetes data – VIP

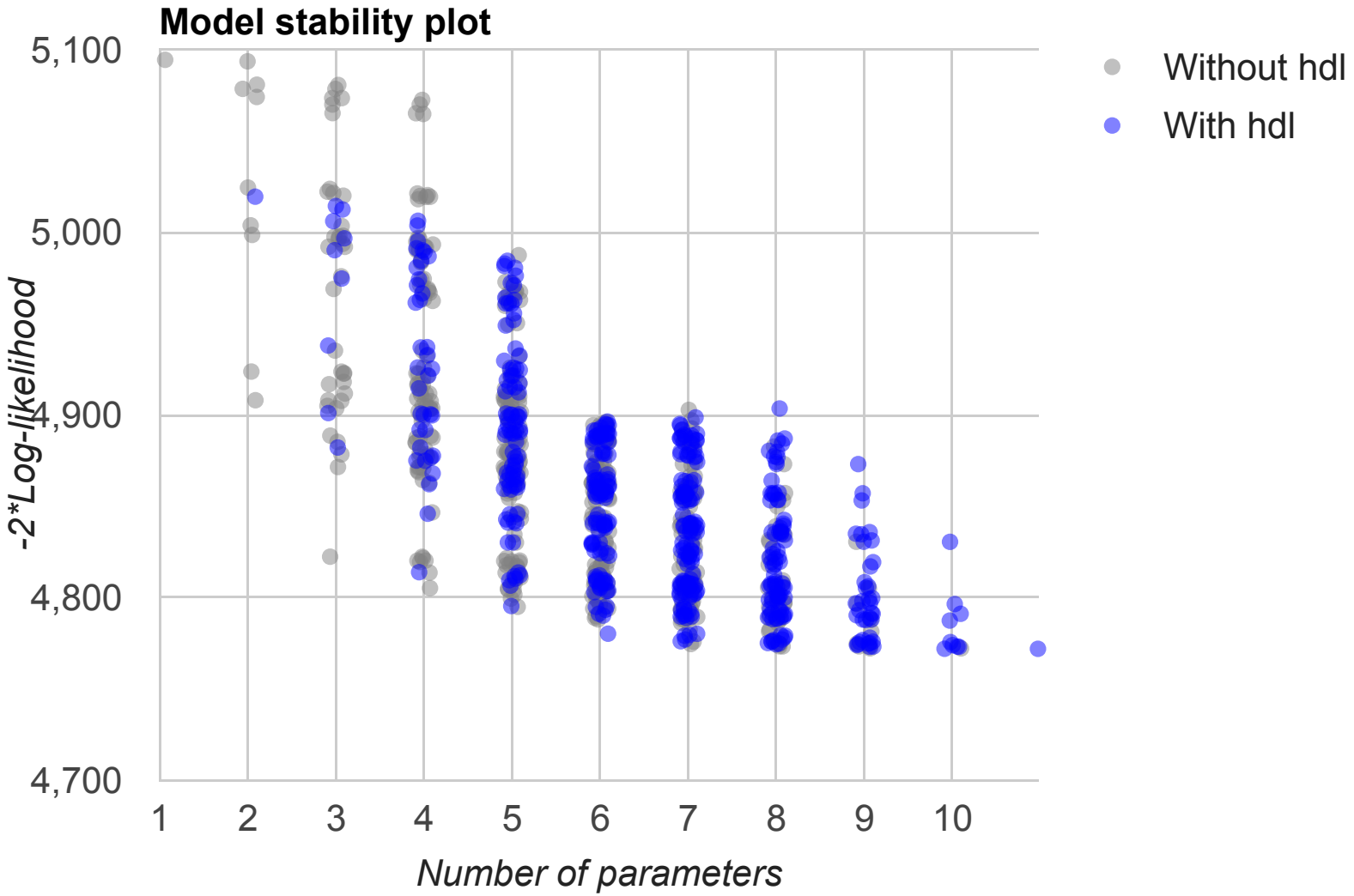
```
require(mplot)
lm.d = lm(y ~ ., data = diabetes)
vis.d = vis(lm.d, B = 200)
plot(vis.d, which = "vip")
```



Model stability plots

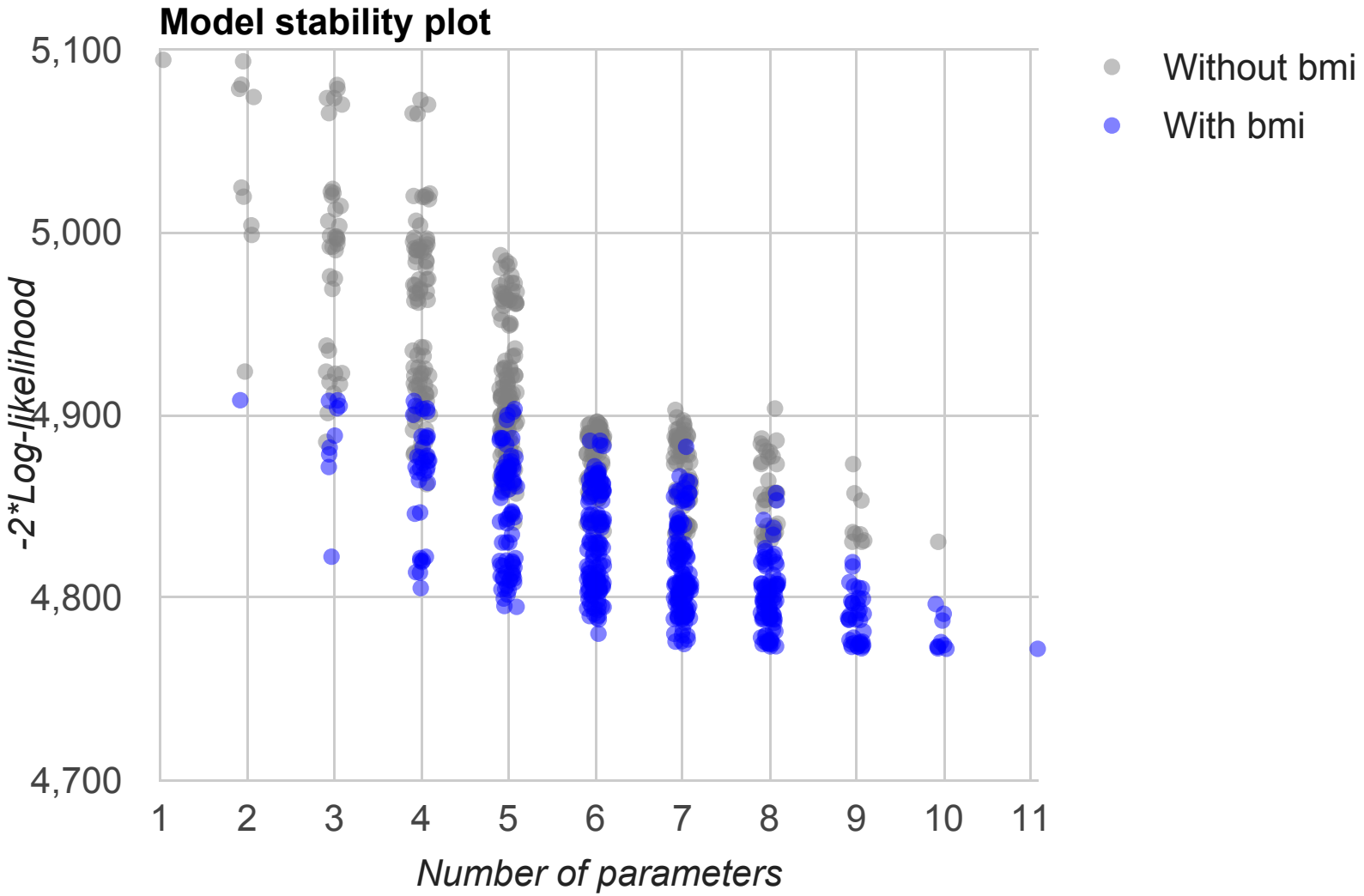
Diabetes data – Loss against size

```
plot(vis.d, which = "lvk", highlight = "hdl")
```



Diabetes data – Loss against size

```
plot(vis.d, which = "lvk", highlight = "bmi")
```



Model stability plots

Aim: To add value to the loss against size plots using a symbol size proportional to a measure of stability.

Procedure

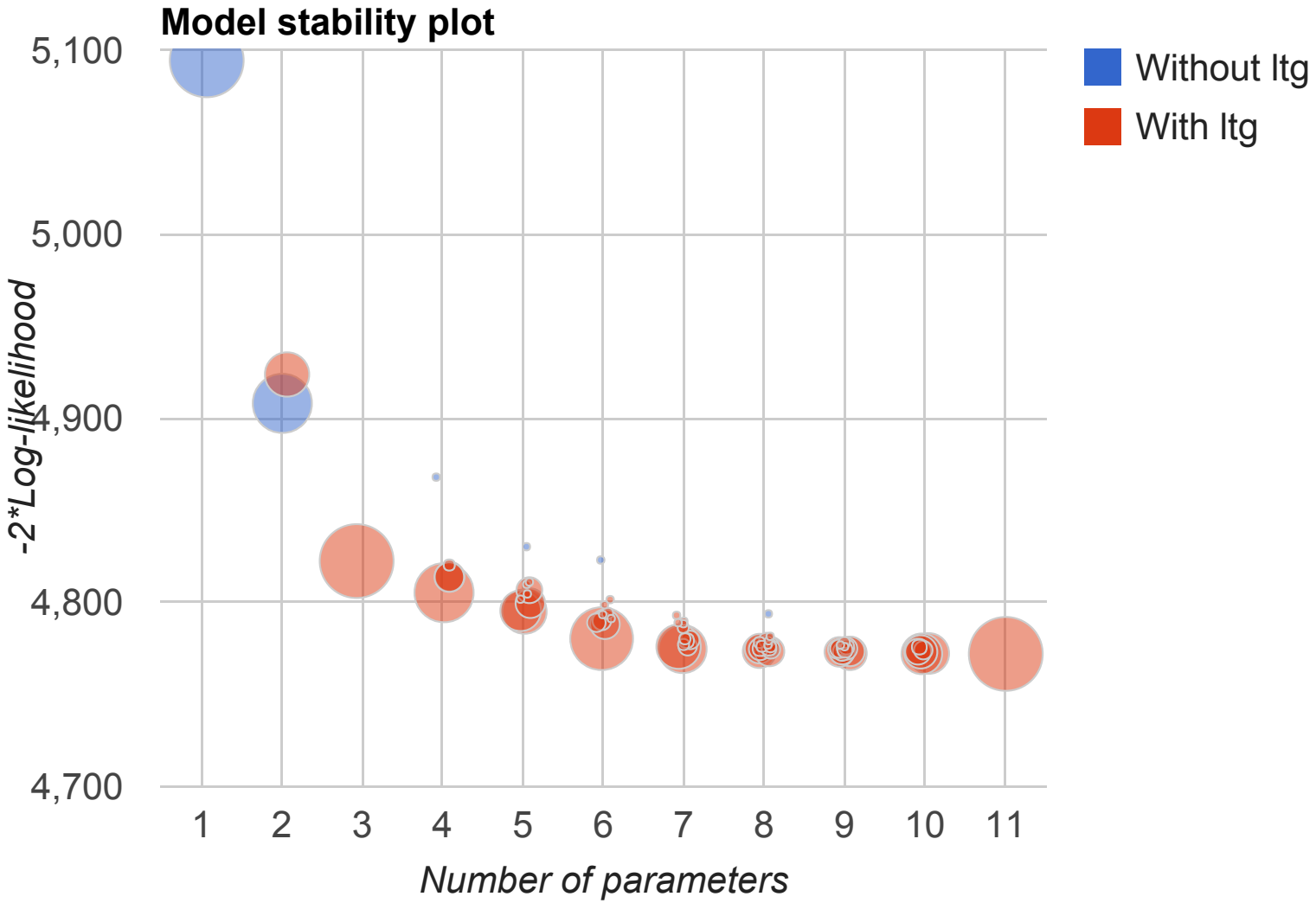
1. Calculate (weighted) bootstrap samples $b = 1, \ldots, B$.
2. For each bootstrap sample, identify the *best* model at each dimension.
3. Add this information to the loss against size plot using model identifiers that are proportional to the frequency with which a model was identified as being *best* at each model size.

References

- Murray, Heritier, and Müller (2013) for generalised linear models

Diabetes data – Model stability plot

```
plot(vis.d, which = "boot", highlight = "ltg")
```



The adaptive fence

The fence

- Let $Q(\alpha)$ be a measure of **lack of fit**
- Specifically we consider $Q(\alpha) = -2\text{LogLik}(\alpha)$

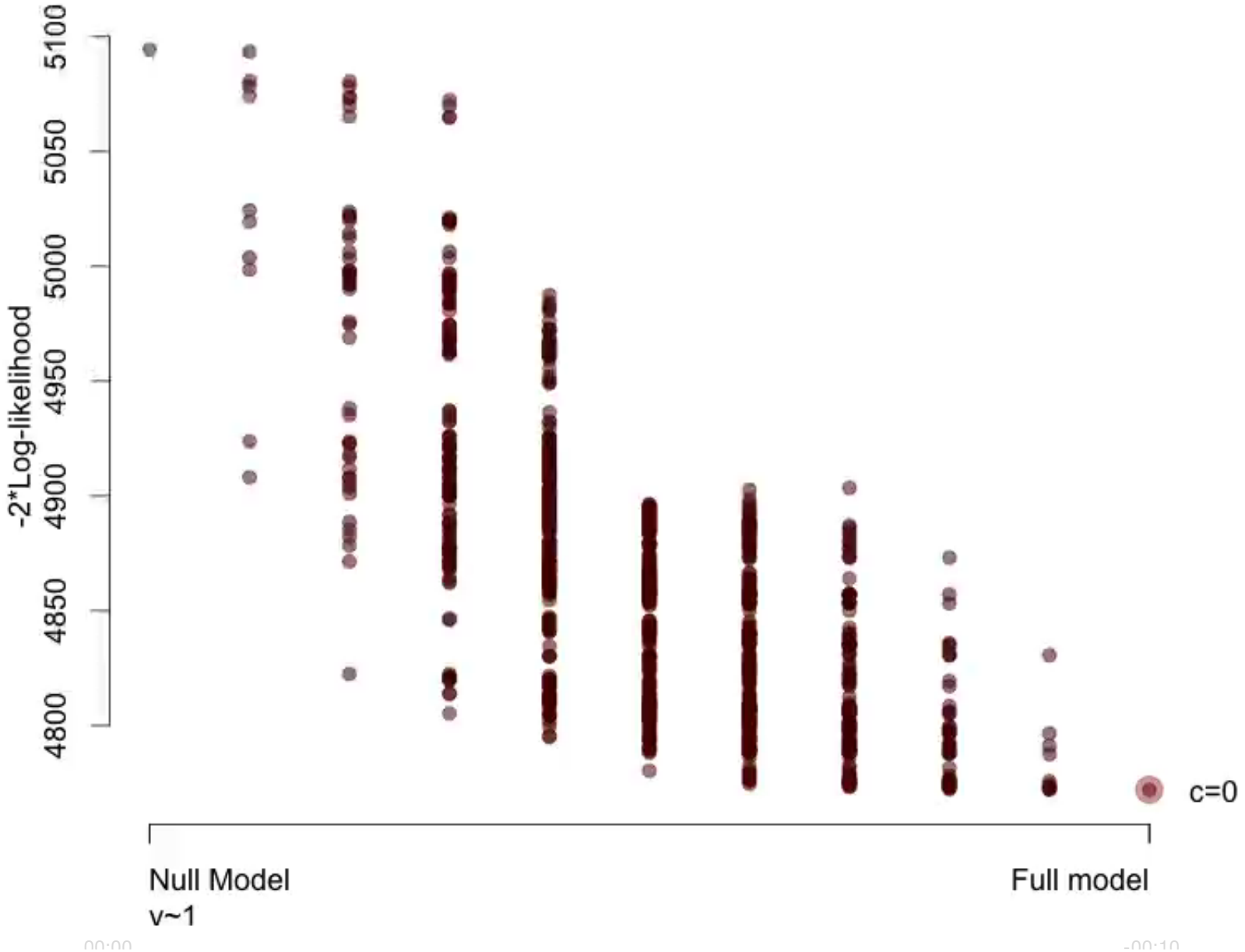
Main idea

The fence is based aroung the inequality:

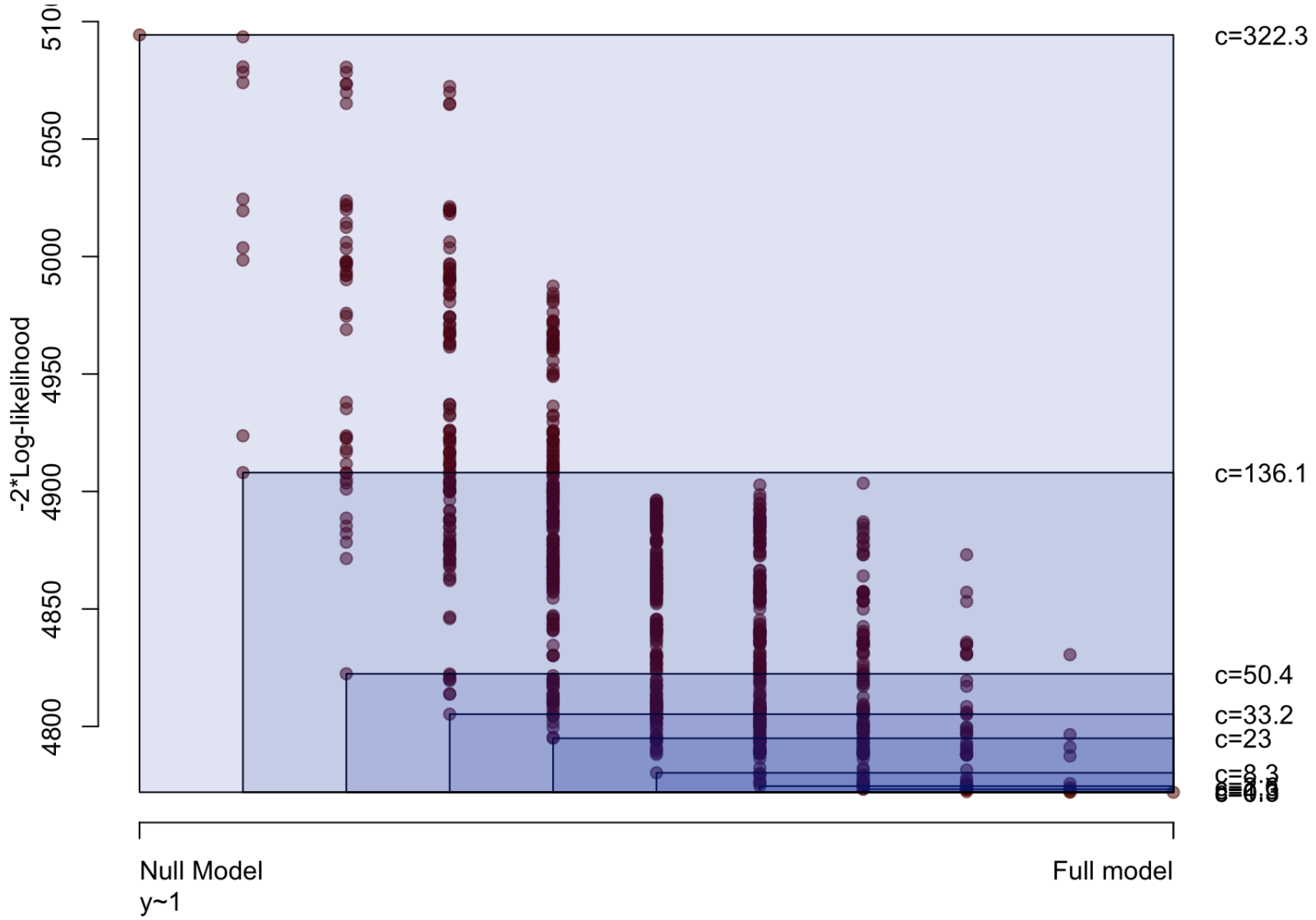
$$Q(\alpha) \leq Q(\alpha_f) + c.$$

- Model α is *inside the fence* if the inequality holds.
- For any $c \geq 0$, the full model α_f is always inside the fence.
- Among the set of models that are inside the fence, model(s) with **smallest dimension** are preferred.

Illustration



Illustration



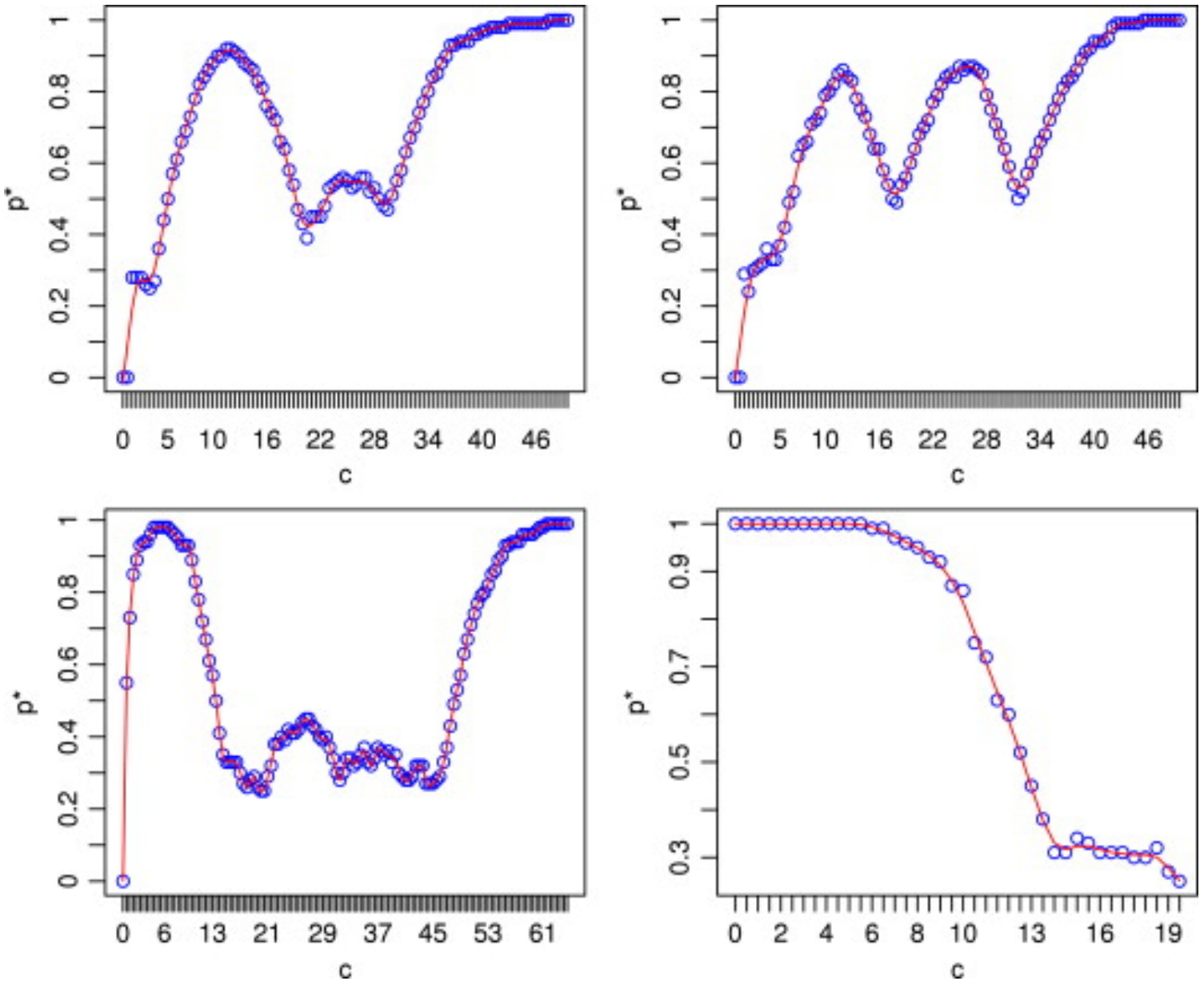
Problem: how to choose c?

Solution: Bootstrap over a range of values of c.

Procedure

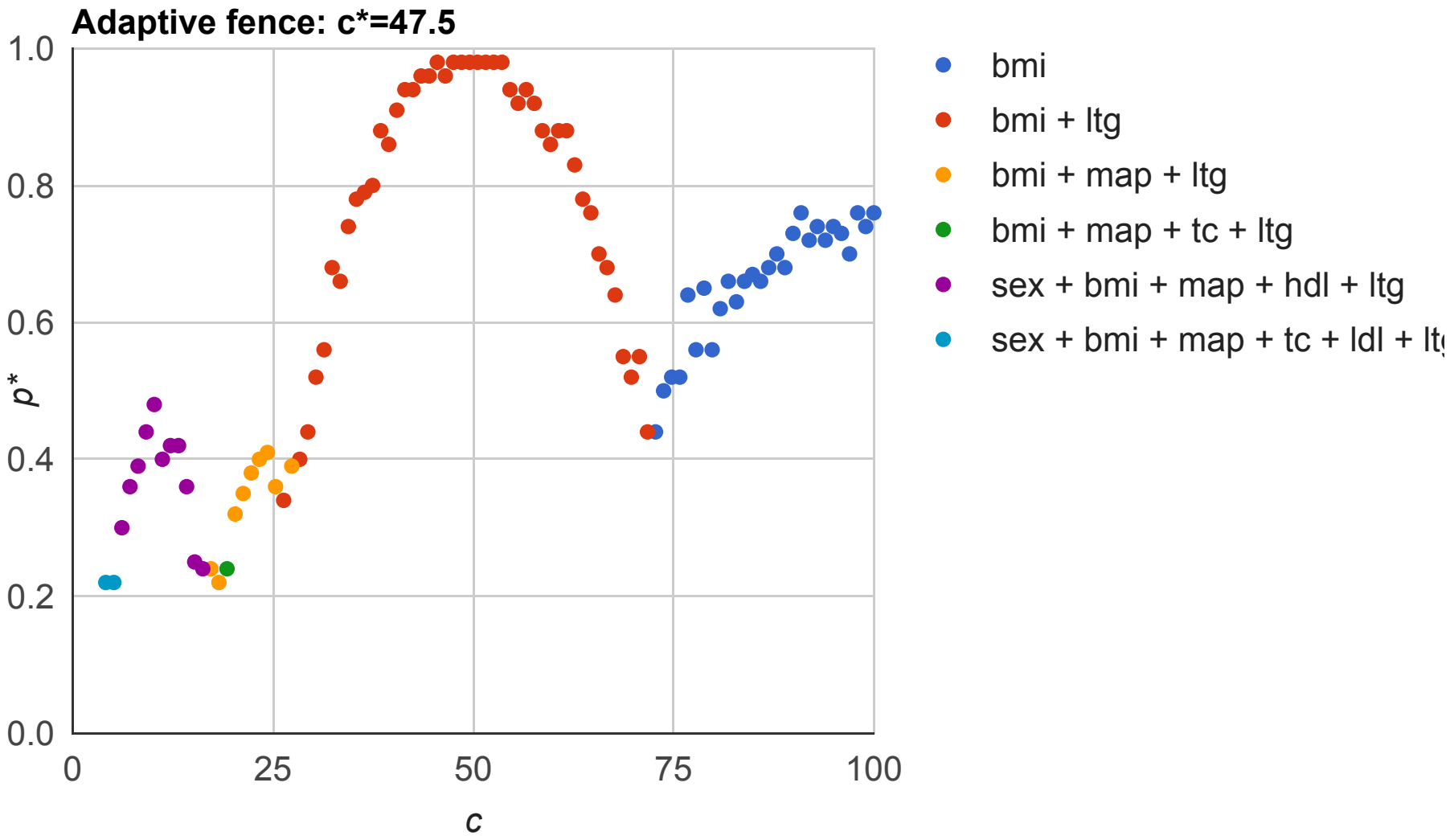
1. For each value of c:
   - Perform parametric bootstrap under α_f.
   - For each bootstrap sample, identify the smallest model that is inside the fence, $\hat{\alpha}(c)$. Jiang, Nguyen, and Rao (2009) suggest that if there is more than one model, choose the one with the smallest $Q(\alpha)$.
   - Let $p^*(\alpha) = P^*\{\hat{\alpha}(c) = \alpha\}$ be the empirical probability of selecting model α at a given value of c.
   - Calculate $p^* = \max_{\alpha \in \mathcal{A}} p^*(\alpha)$
2. Plot values of p^* against c and find first *peak*.
3. Use this value of c with the original data.

What does this look like?



Diabetes data – adaptive fence

```
af.d = af(lm.d, B = 200, n.c = 100, c.max = 100)
plot(af.d)
```



Bootstrapping the lasso

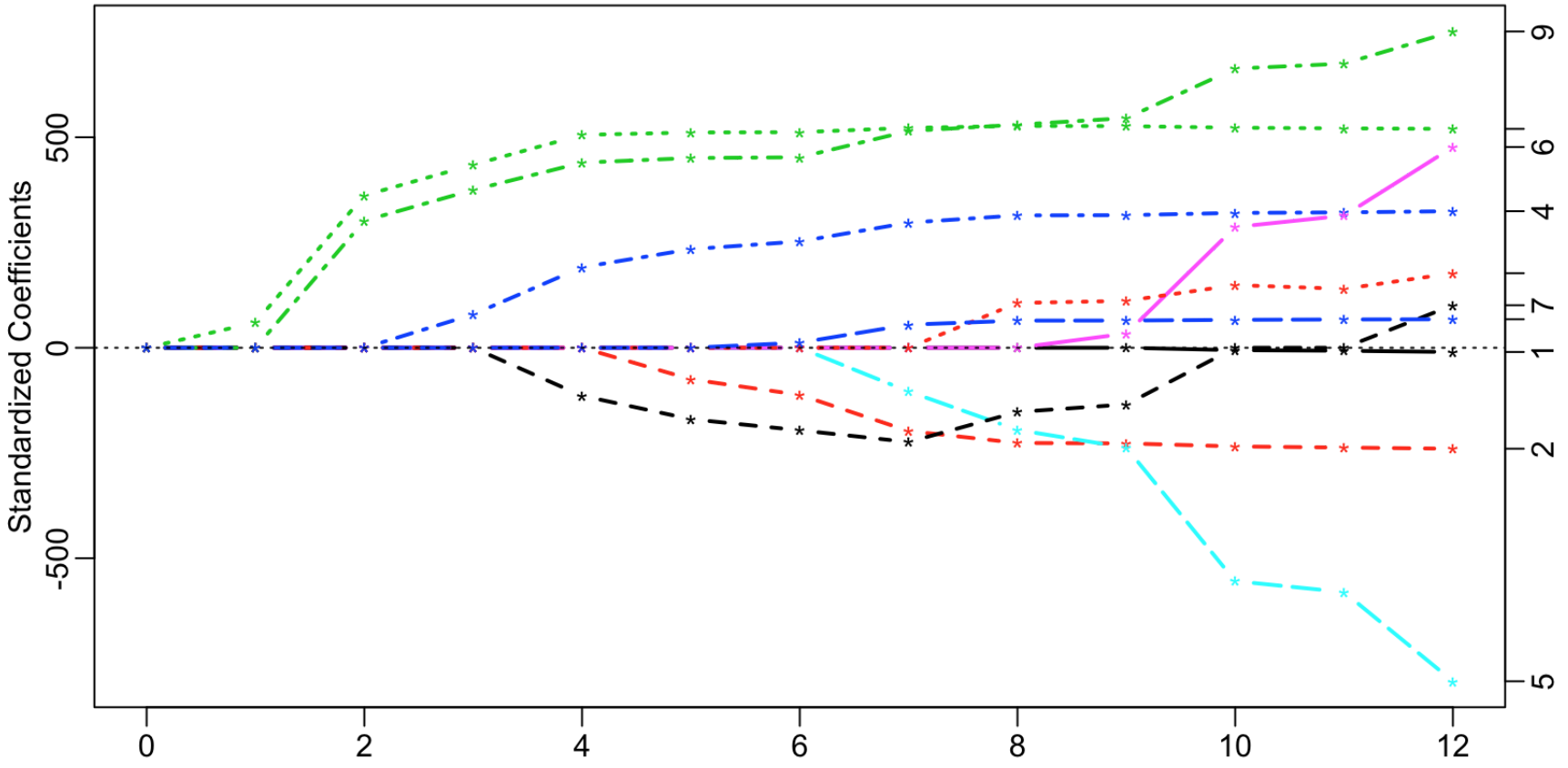
A short history of the lasso

- Tibshirani (1996) did regression with an L_1 norm penalty and called it the lasso (least absolute shrinkage and selection operator).
- The lasso parameter estimates are obtained by minimising the residual sum of squares subject to the constraint that

$$\sum_j |\beta_j| \leq t.$$

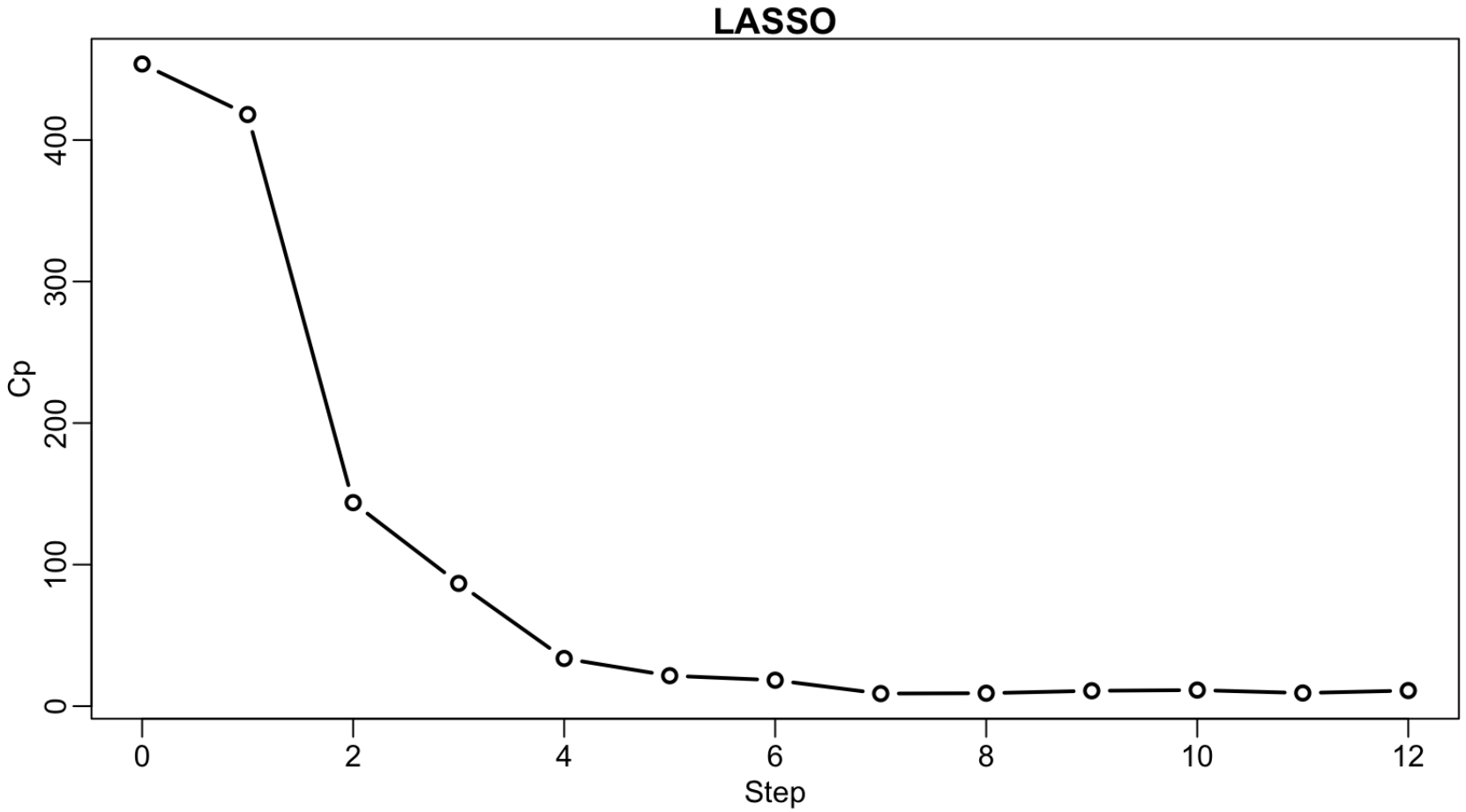
Diabetes data – Lasso

```
require(lars)
x = as.matrix(subset(diabetes, select = -y))
y = as.matrix(subset(diabetes, select = y))
par(mar = c(2.7, 2.7, 1, 2), mgp = c(1.5, 0.5, 0), bg = "transparent")
plot(art.lars, xvar = "step", breaks = FALSE, lwd = 2)
```



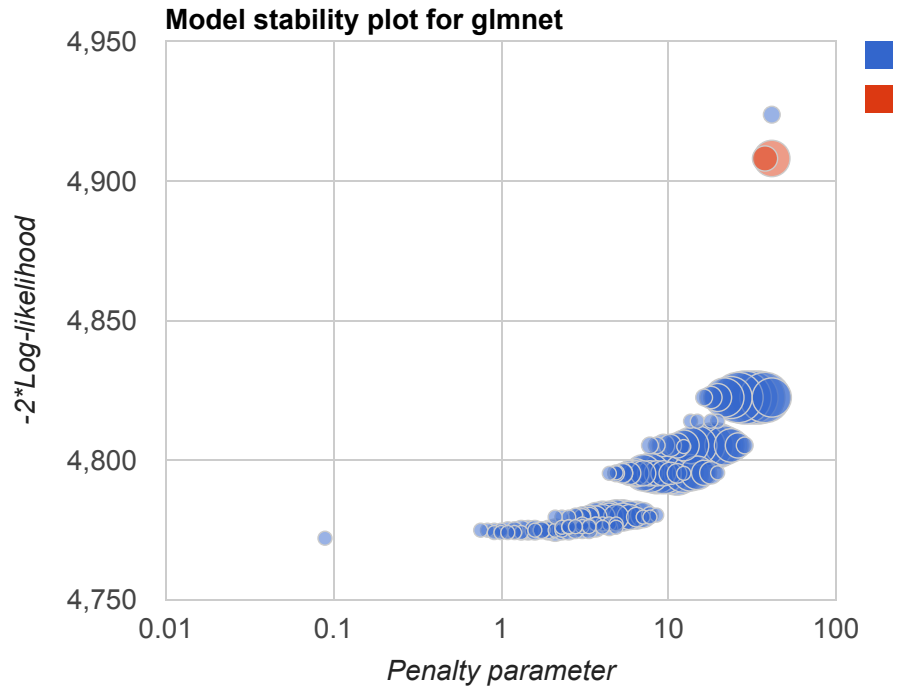
Diabetes data – Lasso

```
plot(art.lars, xvar = "step", plottype = "Cp", lwd = 2)
```



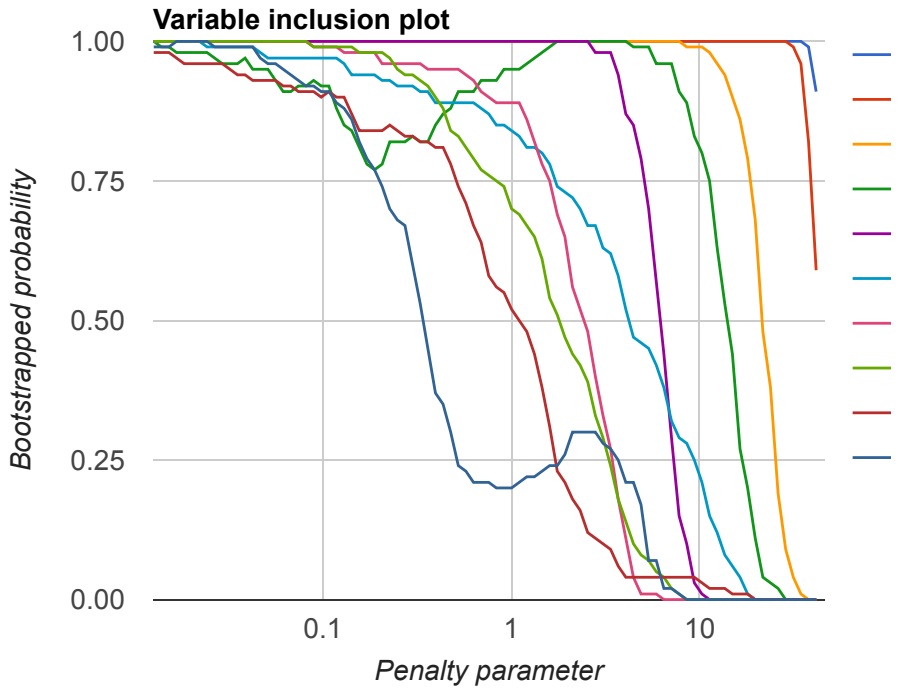
Bootstrapping the lasso

```
bgn.g = bglmnet(lm.d)
plot(bgn.g, which = "boot", highlight = "ltg")
```



Bootstrapping the lasso

```
bgn.d = bglmnet(lm.d)
plot(bgn.d, which = "vip")
```



Put it all together and you have mplot()

mplot: diabetes example

garthtarr.com/apps/mplot/

mplot: diabetes eg

Variable inclusion

Adaptive fence

Model stability

Bootstrap glmnet

Highlight models with:

age

Bootstrap?

Yes No

Min probability with label

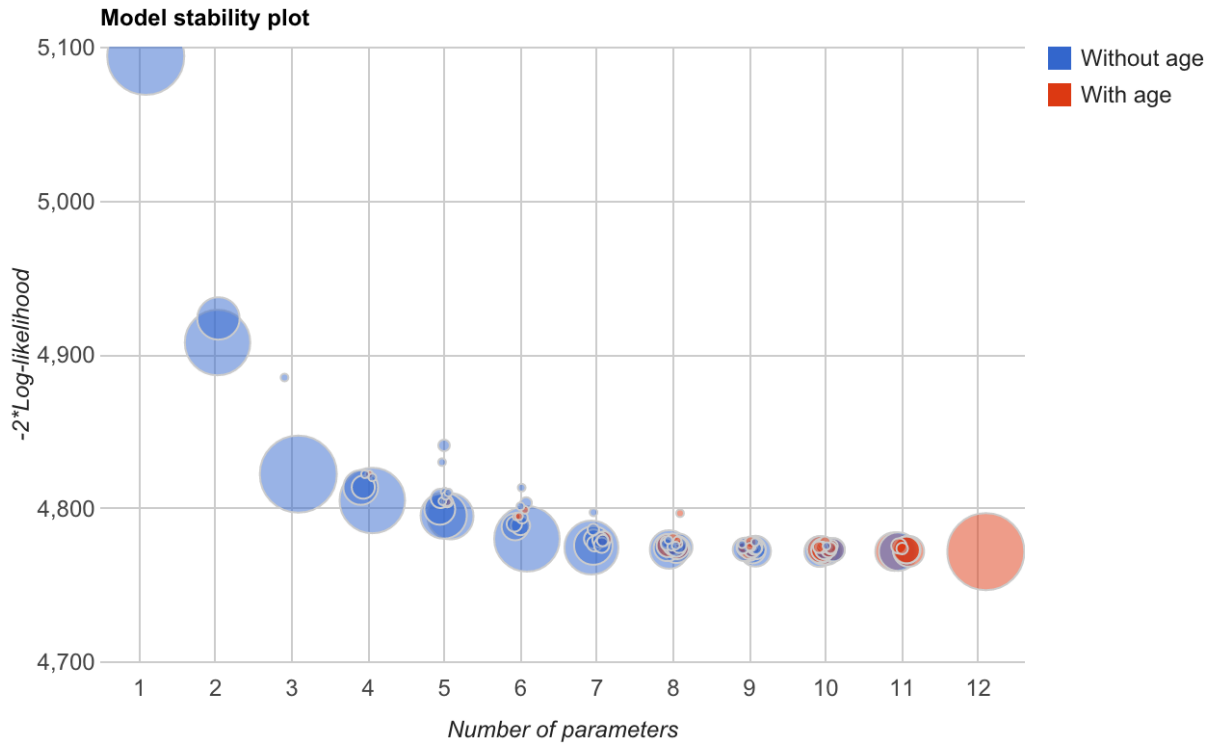
0 0.3 1

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Classic plots

Yes No

Model stability plot



R output

Future work

- Increase the speed of GLM (approximations)?
- Mixed models (speed is an issue here too)
- Robust alternatives (other than simple screening)
- Cox regression has been requested

Find out more

- Tarr G, Mueller S and Welsh AH (2015). “mplot: An R package for graphical model stability and variable selection.” arXiv:1509.07583 [stat.ME], http://arxiv.org/abs/1509.07583.

Slides: garthtarr.com/pres/hobart2015 🔗

Session Info

```
sessionInfo()
```

```
## R version 3.2.2 (2015-08-14)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.10.5 (Yosemite)
##
## locale:
## [1] en_AU.UTF-8/en_AU.UTF-8/en_AU.UTF-8/C/en_AU.UTF-8/en_AU.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] lars_1.2    mplot_0.7.7 knitr_1.11
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.2          codetools_0.2-14     digest_0.6.8
##  [4] foreach_1.4.3        mime_0.4             R6_2.1.1
##  [7] xtable_1.8-0         formatR_1.2.1        magrittr_1.5
## [10] evaluate_0.8         stringi_1.0-1        googleVis_0.5.10
## [13] rmarkdown_0.8.1      RJSONIO_1.3-0        iterators_1.0.8
## [16] tools_3.2.2          stringr_1.0.0        shiny_0.12.2
## [19] httpuv_1.3.3         yaml_2.1.13          parallel_3.2.2
## [22] shinydashboard_0.5.1 htmltools_0.2.6
```

References

Jiang, Jiming, Thuan Nguyen, and J. Sunil Rao. 2009. “A Simplified Adaptive Fence Procedure.” *Statistics & Probability Letters* 79 (5): 625–29. doi:10.1016/j.spl.2008.10.014.

Jiang, Jiming, J. Sunil Rao, Zhonghua Gu, and Thuan Nguyen. 2008. “Fence Methods for Mixed Model Selection.” *The Annals of Statistics* 36 (4): 1669–92. doi:10.1214/07-AOS517.

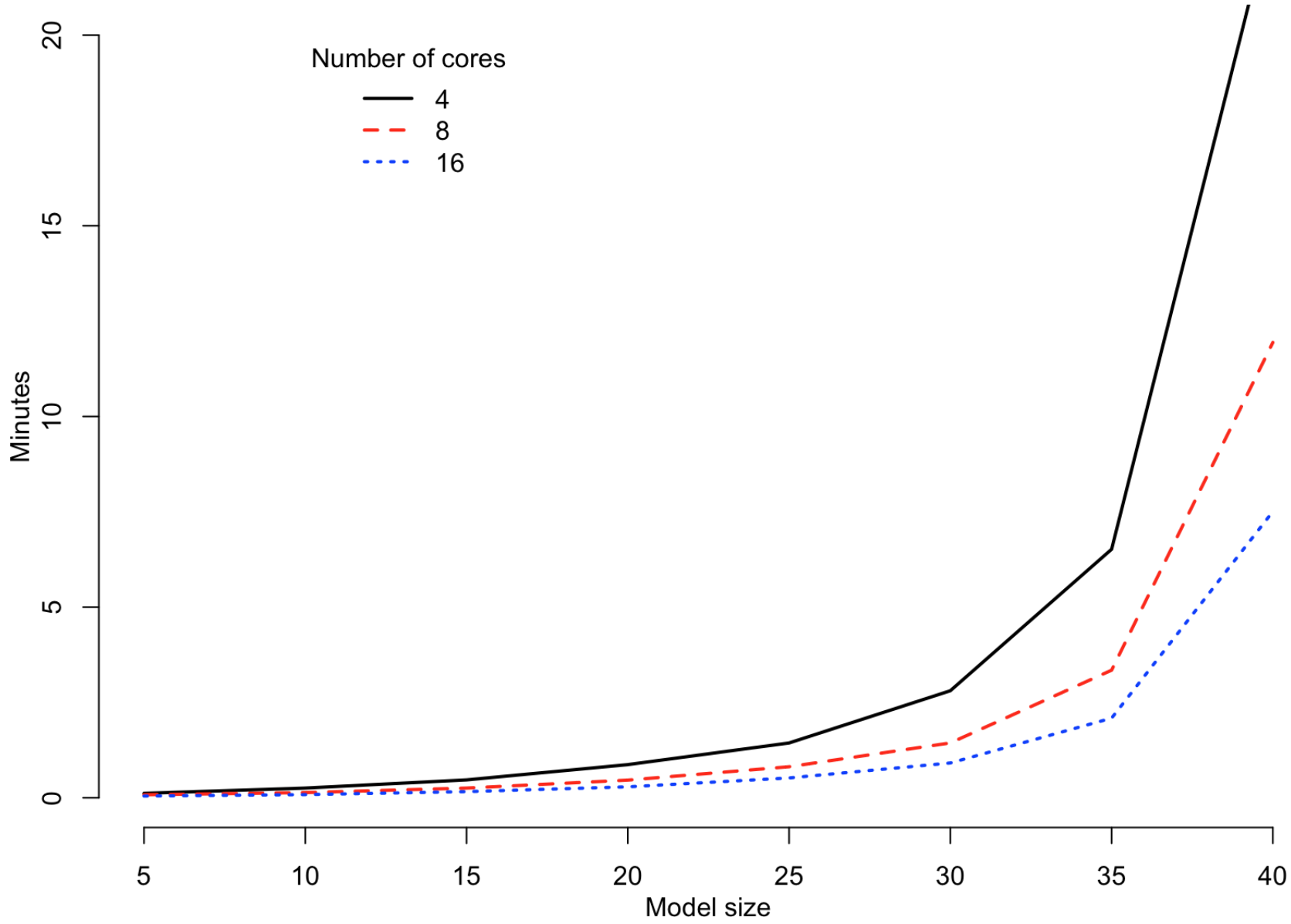
Meinshausen, N, and P Bühlmann. 2010. “Stability Selection.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72 (4): 417–73. doi:10.1111/j.1467-9868.2010.00740.x.

Murray, K, S Heritier, and S Müller. 2013. “Graphical Tools for Model Selection in Generalized Linear Models.” *Statistics in Medicine* 32 (25): 4438–51. doi:10.1002/sim.5855.

Müller, S, and AH Welsh. 2010. “On Model Selection Curves.” *International Statistical Review* 78 (2): 240–56. doi:10.1111/j.1751-5823.2010.00108.x.

Tibshirani, Robert. 1996. “Regression Shrinkage and Selection via the Lasso.” *Journal of the Royal Statistical Society: Series B (Methodological)*, 267–88.

Speed (linear models; B=50; n.c=25)



Speed (linear models; B=50; n.c=25)