

Caso studio 1: dati di uno studio osservazionale su tumori prostatici

Codice R, grafici e commenti

Leonardo Egidi

3-4 aprile 2024

Contents

Introduzione ai dati e all'esperimento	1
Analisi esplorativa dei dati	2
Modello di regressione lineare classico	4
Regolarizzazione di tipo Lasso e Ridge	5
Modello lineare di regressione bayesiano	7
Bibliografia essenziale	13

Introduzione ai dati e all'esperimento

Questi dati, analizzati per la prima volta da Stamey et al. (1989), provengono da uno studio osservazionale che ha esaminato la correlazione tra il livello di antigene prostatico specifico e una serie di misure cliniche negli uomini che stavano per ricevere una prostatectomia radicale. È un frame di dati con 97 righe, corrispondenti ad altrettanti pazienti, e 9 colonne:

- **lpsa**: livello di antigene specifico della prostata
- **lcavol**: logaritmo del volume del tumore
- **lweight**: logaritmo del peso della prostata
- **age**: età del paziente
- **lbph**: logaritmo iperplasia prostatica benigna
- **svi**: invasione vescicole seminali
- **lcp**: logaritmo penetrazione capsulare
- **gleason**: score di Gleason
- **pgg45**: percentuale Gleason score 4 o 5

Analisi esplorativa dei dati

Per prima cosa diamo un'occhiata preliminare ai dati con alcuni grafici delle correlazioni, in cui si notano delle correlazioni positive piuttosto forti tra alcune delle covariate, e tra alcune delle covariate e la variabile risposta.

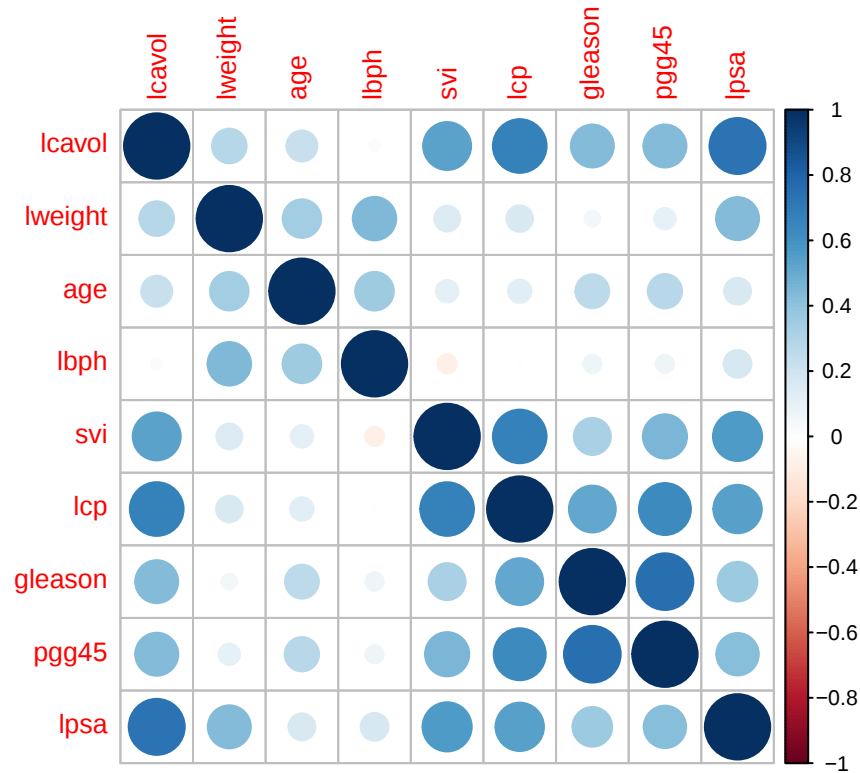
```
library(glmnet)
library(arm)
library(bayesplot)
library(ggplot2)
library(rstanarm)
library(corrplot)
library(ggplot2)
library(GGally)

Prostate <- read.csv("prostate.csv")
Prostate <- Prostate[,-c(1,11)]

# sample size
N <- dim(Prostate)[1]
# predictor matrix
X <- Prostate[,1:8]
# number of predictors
p <- dim(Prostate)[2]
# outcome of interest/response variable
y <- Prostate$lpsa
# to have a quick look
head(Prostate)

##          lcavol  lweight age          lbph svi          lcp gleason pgg45          lpsa
## 1 -0.5798185  2.769459  50 -1.386294    0 -1.386294          6      0 -0.4307829
## 2 -0.9942523  3.319626  58 -1.386294    0 -1.386294          6      0 -0.1625189
## 3 -0.5108256  2.691243  74 -1.386294    0 -1.386294          7     20 -0.1625189
## 4 -1.2039728  3.282789  58 -1.386294    0 -1.386294          6      0 -0.1625189
## 5  0.7514161  3.432373  62 -1.386294    0 -1.386294          6      0  0.3715636
## 6 -1.0498221  3.228826  50 -1.386294    0 -1.386294          6      0  0.7654678

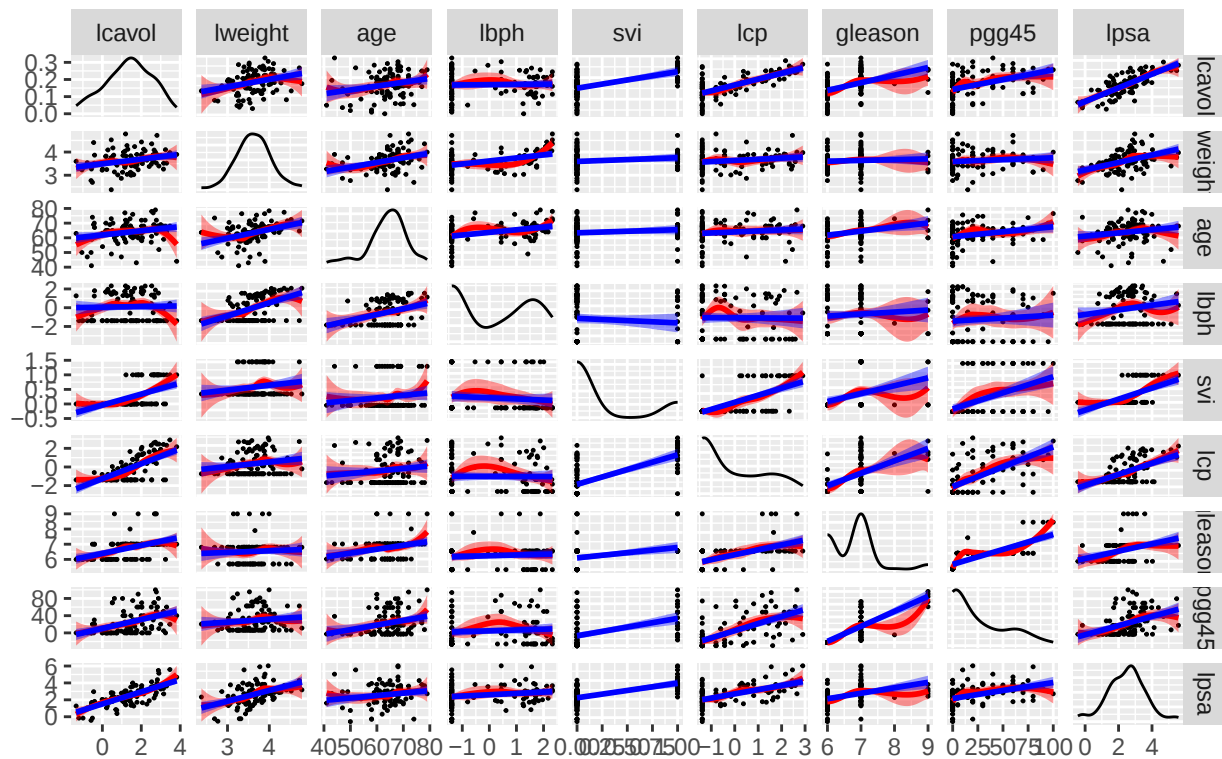
# correlation plot
corrplot(cor(Prostate))
```



```
# pairs plot
```

```
my_fn <- function(data, mapping, ...){
  p <- ggplot(data = data, mapping = mapping) +
    geom_point(size = 0.2) +
    geom_smooth(method=loess, fill="red", color="red", ...) +
    geom_smooth(method=lm, fill="blue", color="blue", ...)
  p
}
```

```
ggpairs(Prostate, upper = list(continuous = my_fn),
        lower = list(continuous = my_fn ))
```



Modello di regressione lineare classico

Iniziamo stimando un semplice modello lineare classico di regressione per la variabile risposta `lpsa` studiando eventuali associazioni statisticamente significative tra le varie covariate e tale outcome di interesse:

```
#####
## CLASSICAL LM
#####

mod_lm <- lm(lpsa ~ ., data = Prostate)
summary(mod_lm)

##
## Call:
## lm(formula = lpsa ~ ., data = Prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.76644 -0.35510 -0.00328  0.38087  1.55770
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)

```

```
## (Intercept)  0.181561    1.320568    0.137  0.89096
## lcavol      0.564341    0.087833    6.425 6.55e-09 ***
## lweight     0.622020    0.200897    3.096 0.00263 **
## age        -0.021248    0.011084   -1.917 0.05848 .
## lbph       0.096713    0.057913    1.670 0.09848 .
## svi        0.761673    0.241176    3.158 0.00218 **
## lcp       -0.106051    0.089868   -1.180 0.24115
## gleason    0.049228    0.155341    0.317 0.75207
## pgg45      0.004458    0.004365    1.021 0.31000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6995 on 88 degrees of freedom
## Multiple R-squared:  0.6634, Adjusted R-squared:  0.6328
## F-statistic: 21.68 on 8 and 88 DF,  p-value: < 2.2e-16
```

```
mod_lm$coefficients
```

```
## (Intercept)      lcavol      lweight      age      lbph      svi
## 0.181560862 0.564341279 0.622019787 -0.021248185 0.096712523 0.761673403
##          lcp      gleason      pgg45
## -0.106050939 0.049227933 0.004457512
```

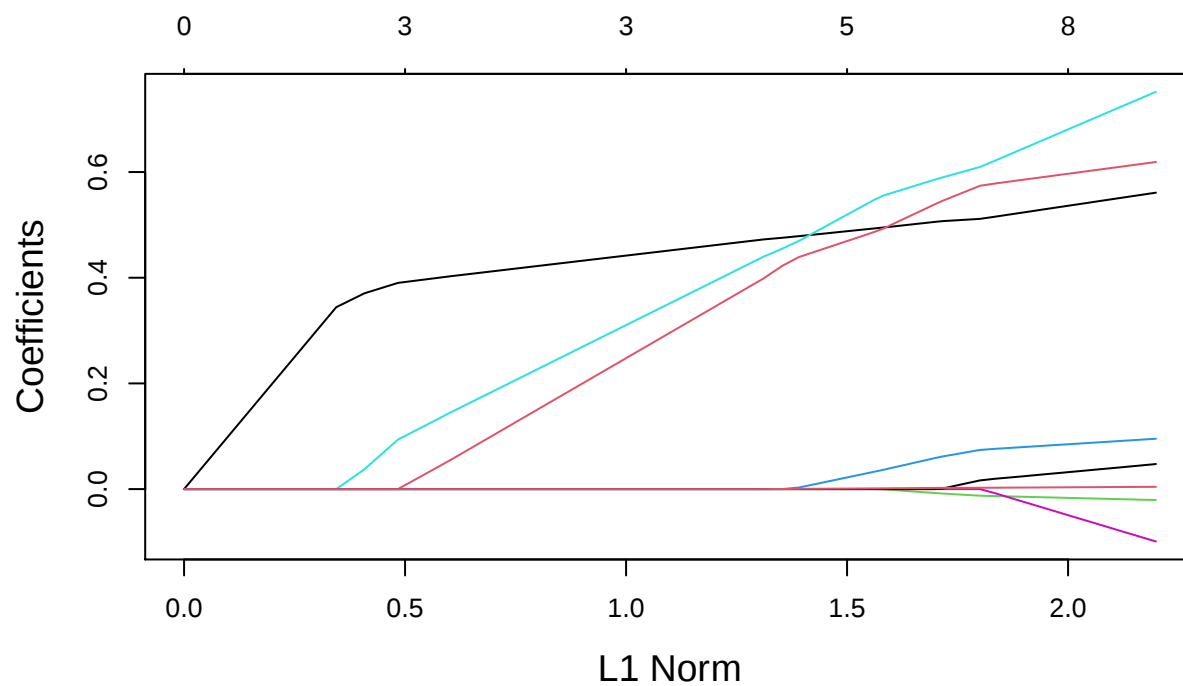
Variabili statisticamente significative al 95%: `lcavol`, `lweight` e `svi`, associate positivamente a `lpsa`.

Regolarizzazione di tipo Lasso e Ridge

Si potrebbero utilizzare delle cosiddette tecniche statistiche di *regolarizzazione*, in grado di spingere ulteriormente verso lo zero (la non significatività statistica) i coefficienti legati a covariate che non risultino significative da un punto di vista statistico. Per lo steso metodo, stimiamo il cosiddetto parametro di *shrinkage* tramite un cosiddetto approccio di cross-validazione:

```
#####
## LASSO/RIDGE
#####

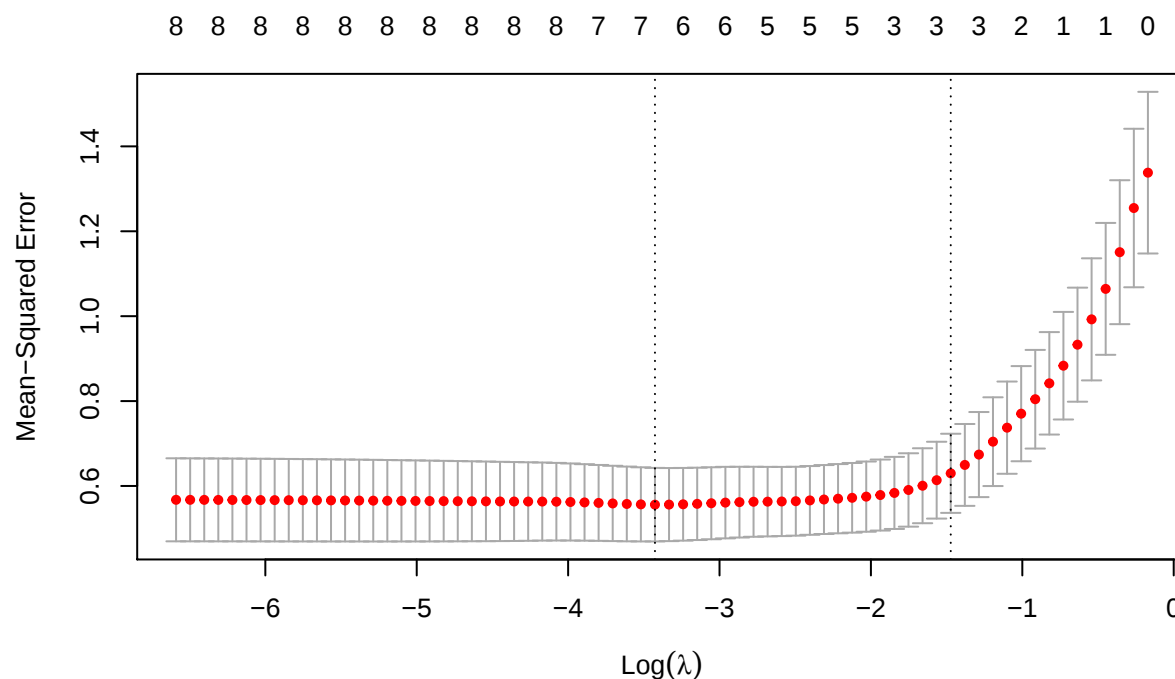
# regularization techniques
mod_glmnet <- glmnet(x = X, y = y, alpha = 1, family = "gaussian")
plot(mod_glmnet, cex.lab = 1.4)
```



```
cvfit <- cv.glmnet(x = as.matrix(X), y = y)
cvfit
```

```
##
## Call:  cv.glmnet(x = as.matrix(X), y = y)
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure      SE Nonzero
## min 0.0325   36  0.5560 0.08661         7
## 1se 0.2293   15  0.6298 0.09319         3
```

```
plot(cvfit)
```



```
cvfit$lambda.min
```

```
## [1] 0.03250172
```

```
coef(cvfit, s = "lambda.min")
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
```

```
##              s1
## (Intercept) 0.161190508
## lcavol      0.508157222
## lweight     0.552486888
## age         -0.009374709
## lbph        0.064736545
## svi         0.594693521
## lcp         .
## gleason     0.004797183
## pgg45       0.002351087
```

```
mod_ridge <- glmnet(x = X, y = y, alpha = 0)
```

```
cvfit_ridge <- cv.glmnet(x = as.matrix(X), y = y, alpha = 0)
```

Modello lineare di regressione bayesiano

Ora stimiamo sempre un modello di regressione ma secondo l'ottica bayesiana, ipotizzando prima delle distribuzioni a priori debolmente informative gaussiane e poi di tipo Cauchy per

i coefficienti di regressione:

```
#####  
## Bayesian model (with MCMC sampling)  
#####  
  
# with gaussian priors  
mod_lm_bayes <- stan_glm(lpsa~., data =Prostate,  
                        family=gaussian)  
  
# with cauchy priors  
mod_lm_bayes2 <- stan_glm(lpsa~., data =Prostate,  
                        family=gaussian,  
                        prior = cauchy(),  
                        prior_intercept = cauchy())
```

Possiamo adesso rappresentare graficamente le stime a posteriori dei parametri:

```
print(mod_lm_bayes)  
  
## stan_glm  
## family:      gaussian [identity]  
## formula:     lpsa ~ .  
## observations: 97  
## predictors:  9  
## -----  
##              Median MAD_SD  
## (Intercept)  0.2      1.3  
## lcavol       0.6      0.1  
## lweight      0.6      0.2  
## age          0.0      0.0  
## lbph         0.1      0.1  
## svi          0.8      0.2  
## lcp          -0.1     0.1  
## gleason      0.0      0.2  
## pgg45        0.0      0.0  
##  
## Auxiliary parameter(s):  
##      Median MAD_SD  
## sigma 0.7      0.1  
##  
## -----  
## * For help interpreting the printed output see ?print.stanreg  
## * For info on the priors used see ?prior_summary.stanreg
```

```
prior_summary(mod_lm_bayes)
```

```
## Priors for model 'mod_lm_bayes'
## -----
## Intercept (after predictors centered)
##   Specified prior:
##     ~ normal(location = 2.5, scale = 2.5)
##   Adjusted prior:
##     ~ normal(location = 2.5, scale = 2.9)
##
## Coefficients
##   Specified prior:
##     ~ normal(location = [0,0,0,...], scale = [2.5,2.5,2.5,...])
##   Adjusted prior:
##     ~ normal(location = [0,0,0,...], scale = [2.45,6.74,0.39,...])
##
## Auxiliary (sigma)
##   Specified prior:
##     ~ exponential(rate = 1)
##   Adjusted prior:
##     ~ exponential(rate = 0.87)
## -----
## See help('prior_summary.stanreg') for more details
```

```
print(mod_lm_bayes2)
```

```
## stan_glm
## family:      gaussian [identity]
## formula:      lpsa ~ .
## observations: 97
## predictors:   9
## -----
##              Median MAD_SD
## (Intercept)  0.3      1.3
## lcavol       0.6      0.1
## lweight      0.6      0.2
## age          0.0      0.0
## lbph         0.1      0.1
## svi          0.7      0.2
## lcp         -0.1      0.1
## gleason      0.0      0.2
## pgg45        0.0      0.0
##
## Auxiliary parameter(s):
##      Median MAD_SD
```

```

## sigma 0.7    0.1
##
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg

prior_summary(mod_lm_bayes2)

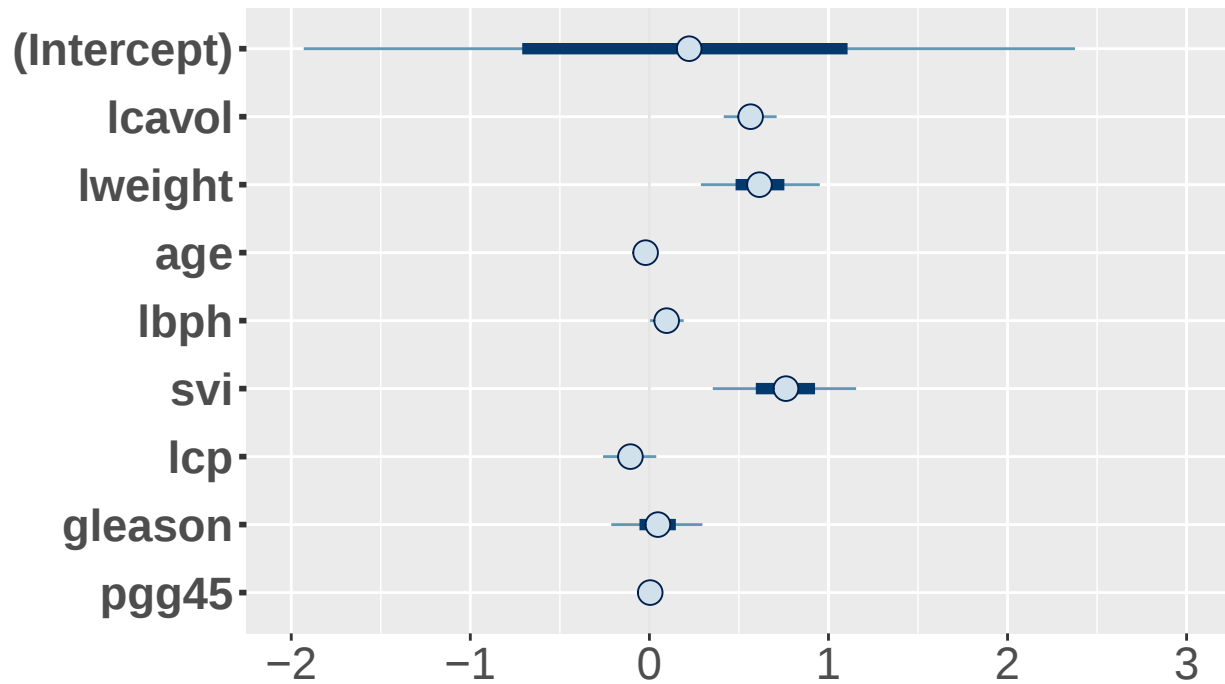
## Priors for model 'mod_lm_bayes2'
## -----
## Intercept (after predictors centered)
## ~ cauchy(location = 0, scale = 2.5)
##
## Coefficients
## ~ cauchy(location = [0,0,0,...], scale = [2.5,2.5,2.5,...])
##
## Auxiliary (sigma)
## Specified prior:
## ~ exponential(rate = 1)
## Adjusted prior:
## ~ exponential(rate = 0.87)
## -----
## See help('prior_summary.stanreg') for more details

posterior <- as.matrix(mod_lm_bayes)
posterior2 <- as.matrix(mod_lm_bayes2)

# depict posterior distributions intervals for the parameters
beta_names <- paste0("beta[", 9:1, "]")
mcmc_intervals(posterior,
               pars=c("(Intercept)", "lcavol", "lweight",
                     "age", "lbph", "svi", "lcp",
                     "gleason", "pgg45"))+
  xaxis_text(on =TRUE, size=rel(1.9))+
  yaxis_text(on =TRUE, size=rel(1.9))+
  # scale_y_discrete(labels = rev((parse(text= beta_names))))+
  ggtitle("Weakly informative")+
  theme(plot.title = element_text(hjust = 0.5, size =rel(2)))+
  xlim(-2,3)

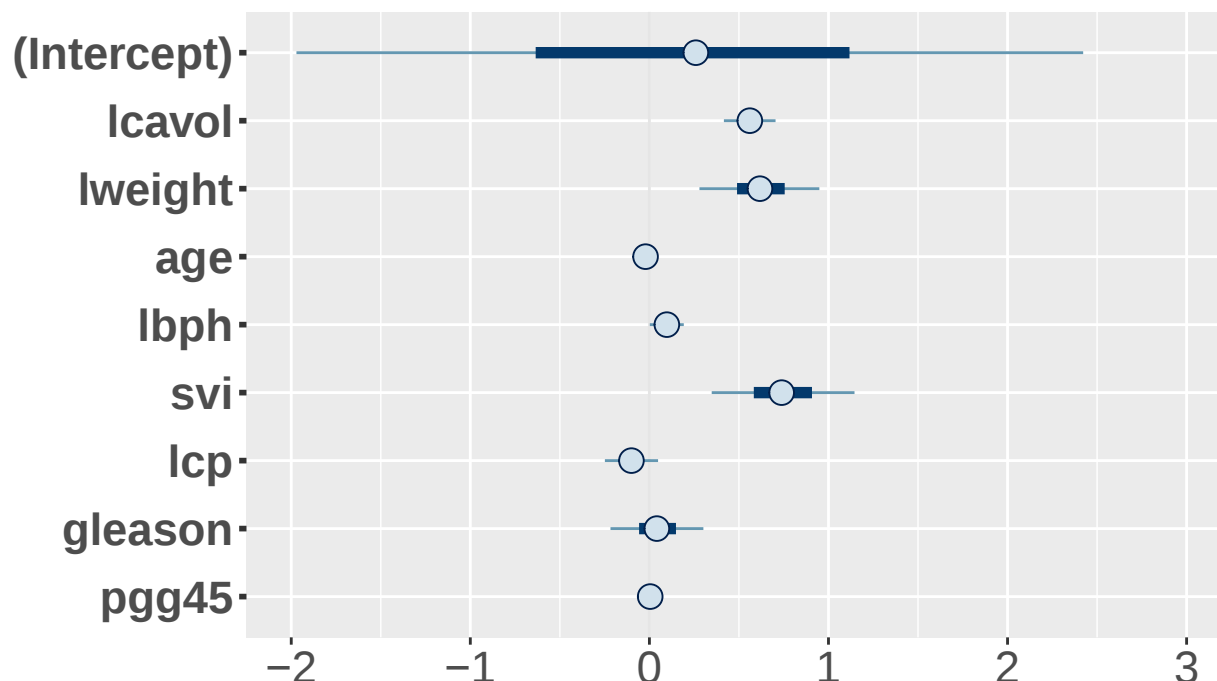
```

Weakly informative



```
mcmc_intervals(posterior2,
               pars=c("(Intercept)", "lcavol", "lweight",
                      "age", "lbph", "svi", "lcp",
                      "gleason", "pgg45"))+
  xaxis_text(on =TRUE, size=rel(1.9))+
  yaxis_text(on =TRUE, size=rel(1.9))+
  # scale_y_discrete(labels = rev((parse(text= beta_names))))+
  ggtitle("Weakly informative")+
  theme(plot.title = element_text(hjust = 0.5, size =rel(2)))+
  xlim(-2,3)
```

Weakly informative



Possiamo notare che otteniamo conclusioni coerenti con il metodo frequentista classico: le variabili che risultano significative, ovvero sia i cui intervalli a posteriori non includono lo zero, risultano essere: `lcavol`, `lweight` e `svi`.

Infine possiamo fare un confronto per le diverse stime ottenute:

	lm	lasso	ridge	bayes1	bayes2
(Intercept)	0.182	0.161	0.012	0.222	0.259
lcavol	0.564	0.508	0.492	0.565	0.561
lweight	0.622	0.552	0.604	0.615	0.617
age	-0.021	-0.009	-0.017	-0.021	-0.022
lbph	0.097	0.065	0.086	0.096	0.097
svi	0.762	0.595	0.685	0.763	0.739
lcp	-0.106	0.000	-0.040	-0.106	-0.100
gleason	0.049	0.005	0.064	0.047	0.042
pgg45	0.004	0.002	0.003	0.005	0.004

Come possiamo notare, un metodo di regolarizzazione come il Lasso tende a spingere le variabili non significative ad essere ulteriormente non significative—ovverossia, ad andare più verso lo zero. In linea generale, le stime dei modelli tendono ad assomigliarsi molto: tuttavia, un approccio di tipo bayesiano può consentire l’incorporazione di conoscenze preliminari degli sperimentatori per quanto riguarda i coefficienti di alcune delle covariate inserite.

Bibliografia essenziale

Stamey, Thomas A, John N Kabalin, John E McNeal, Iain M Johnstone, Fuad Freiha, Elise A Redwine, and Norman Yang. 1989. “Prostate Specific Antigen in the Diagnosis and Treatment of Adenocarcinoma of the Prostate. II. Radical Prostatectomy Treated Patients.” *The Journal of Urology* 141 (5): 1076–83.