Bayesian Causal Inference: elaborato MD2SL24

Dario Comanducci

15-01-2025

Introduzione

Il dataset da analizzare riguarda uno studio completamente randomizzato su un gruppo di individui disoccupati per promuovere un reimpiego di alta qualità e prevenire la depressione:

- i soggetti con trattamento hanno seguito delle sessioni di formazione;
- ai soggetti di controllo è stato dato solo un opuscolo con suggerimenti per la ricerca di lavoro.

Le variabili d'interesse consistono nel livello di depressione ed il reimpiego, valutate dopo 6 mesi dall'assegnazione del trattamento.

Caricamento e descrizione dei dati (esercizio 1)

Load the dataset in R (Filename: JOBSII_HR.dta)

```
library(haven)
```

```
## Warning: package 'haven' was built under R version 4.3.3
```

```
rm(list=ls())
setwd('C:\\Users\\dario\\Documents\\Projects\\Master\\BayesianCausalInference\\R')
df = as.data.frame(read_dta("JOBSII_HR.dta"))
print(head(df))
```

```
##   sex      age race nonmarried educ EconHard assertive motivation depress0 Z W
## 1   1 37.85205    1          1   14     4.33      3.25        6.0     2.90 1 1
## 2   0 39.45753    0          0   11     3.67      2.50        4.5     2.73 0 0
## 3   1 51.06575    0          0   17     3.33      4.00        4.5     2.45 0 0
## 4   1 49.14795    0          0   14     4.00      4.33        6.0     2.73 0 0
## 5   1 35.33151    1          0   13     4.67      3.50        5.5     2.82 1 0
## 6   1 36.88493    0          1   12     3.67      3.50        5.0     2.64 0 0
##   employ6 depress6
## 1       1 2.363636
## 2       1 3.181818
## 3       1 1.727273
## 4       0 2.545455
## 5       1 1.545455
## 6       1 2.727273
```

All'interno della tabella caricata

- le colonne `employ6` e `depress6` corrispondono alle variabili di uscita;
- `Z` indica i soggetti assegnati al trattamento (1) o al gruppo di controllo (0);
- `W` riporta i soggetti che hanno effettivamente partecipato alle lezioni (1) o meno (0);
- le restanti 9 colonne consistono in covariate, di cui

- `age` (età), `educ` (grado scolastico completato), `EconHard` (difficoltà economiche), `assertive` (livello di assertività), `motivation` (livello motivazionale), `depress0` (livello di depressione iniziale) sono riconducibili a variabili continue;
- le restanti 3, ossia `sex` (0=maschio), `race` (0=bianco) e `nonmarried` (0=sposato) sono binarie.

Analisi e visualizzazione dei dati (esercizio 2)

For each variable, calculate the mean for the whole sample and within each treatment group. For continuous covariates, also report the medians, standard deviation and ranges within each treatment group. Record your results in a table. In a few sentences, comment on what you see and whether it is expected.

Funzioni di appoggio

```
library(pracma)
```

```
## Warning: package 'pracma' was built under R version 4.3.3
```

```
ComputeStats <- function(df, vars, fun)
{
   idc = which(df$Z==0)
   idt = which(df$Z==1)

   resume = apply(df[,vars],2,fun)
   resume = rbind(resume, apply(df[idc,vars],2,fun))
   resume = rbind(resume, apply(df[idt,vars],2,fun))

   f_name <- deparse(substitute(fun))
   rownames(resume) = c(paste0(f_name,'(whole)'),
                        paste0(f_name,'(ctrl)'),
                        paste0(f_name,'(treat)'))
   return (resume)
}


SummarizeVisualData <- function (df, dataName, binary=F, breaks='Sturges')
{
   idV = which(names(df) == dataName)
   t = df$Z
   idC = which(t==0)
   idT = which(t==1)

   x = df[[idV]]
   xc = df[idC,idV]
   xt = df[idT,idV]

   # colore controllo
   colC = 'tan2'; color_data = col2rgb(colC)
   colC_tr = rgb(color_data[1]/255, color_data[2]/255, color_data[3]/255, alpha = 0.5)
   # colore trattati
   colT = 'olivedrab3'; color_data = col2rgb(colT)
   colT_tr = rgb(color_data[1]/255, color_data[2]/255, color_data[3]/255, alpha = 0.5)
   # colore generico
   colG = 'lightskyblue3'; color_data = col2rgb(colG)
   colG_tr = rgb(color_data[1]/255, color_data[2]/255, color_data[3]/255, alpha = 0.5)
```

```
# Filling colors & border colors
fcol= c(colC, colT, colG)
fcol_tr = c(colC_tr, colT_tr, colG_tr)
bcol = c('tan3', 'olivedrab4', 'lightskyblue4')

if (binary==T)
{
   par(mfrow=c(1,3), mar=c(2.5,4.0,1.5,0.5)) #margini bottom, left, top, right
   counts = table(x);
   barplot(height=counts/sum(counts)*100, col=fcol[3], border=bcol[3],
           names=c(0,1), ylab='perc [%]', ylim=c(0,100),
           main = paste(dataName, ' (whole)'))
   grid(lty='solid', lwd=0.5, col='white', nx=NA, ny=NULL)

   counts = table(x[idC]);
   barplot(height=counts/sum(counts)*100, col=fcol[1], border=bcol[1],
           names=c(0,1), ylab='perc [%]', ylim=c(0,100),
           main = paste(dataName, ' (ctrl)'))
   grid(lty='solid', lwd=0.5, col='white', nx=NA, ny=NULL)

   counts = table(x[idT]);
   barplot(height=counts/sum(counts)*100, col=fcol[2], border=bcol[2],
           names=c(0,1), ylab='perc [%]', ylim=c(0,100),
           main = paste(dataName, ' (treat)'))
   grid(lty='solid', lwd=0.5, col='white', nx=NA, ny=NULL)
}
else
{
   par(mfrow=c(2,2), mar=c(2.5,4.0,1.5,0.5)) #margini bottom, left, top, right

   h = hist(x=x, breaks=breaks, plot=F)
   hc = hist(x=xc, breaks=breaks, plot=F)
   ht = hist(x=xt, breaks=breaks, plot=F)

   maxh = ceil(max(c(max(h$density),
                     max(hc$density),
                     max(ht$density)))*100)/100;

   boxplot(x=x, frame.plot=F, col=fcol[3], border=bcol[3],
   ylab=dataName, main='whole')

   hist(x=x, freq=F, col=fcol[3], border=bcol[3], xlab=dataName,
        breaks=breaks, ylim=c(0,maxh), main='whole')
   grid(lty='solid', lwd=0.5, col='white', nx=NA, ny=NULL)

   boxplot(formula=x ~ t, data=df,
   frame.plot=F, col=fcol, border=bcol,
   ylab=dataName, xlab='Treated', main='ctrl vs treat.')

   hist(x=xc, freq=F, col=fcol_tr[1], border=bcol[1], xlab=dataName,
        breaks=breaks, ylim=c(0,maxh), main='ctrl vs treat.')
   hist(x=xt, freq=F, breaks=breaks, col=fcol_tr[2], border=bcol[2], add=T)
   grid(lty='solid', lwd=0.5, col='white', nx=NA, ny=NULL)
```

```
      #legend('topright', c('Not treated', 'Treated'), col=c(colC,colT), pch=15)
   }

   par(mfrow = c(1,1), mar=c(5,4,4,2) + 0.1)# Reset to default layout
}
```

Variabili binarie

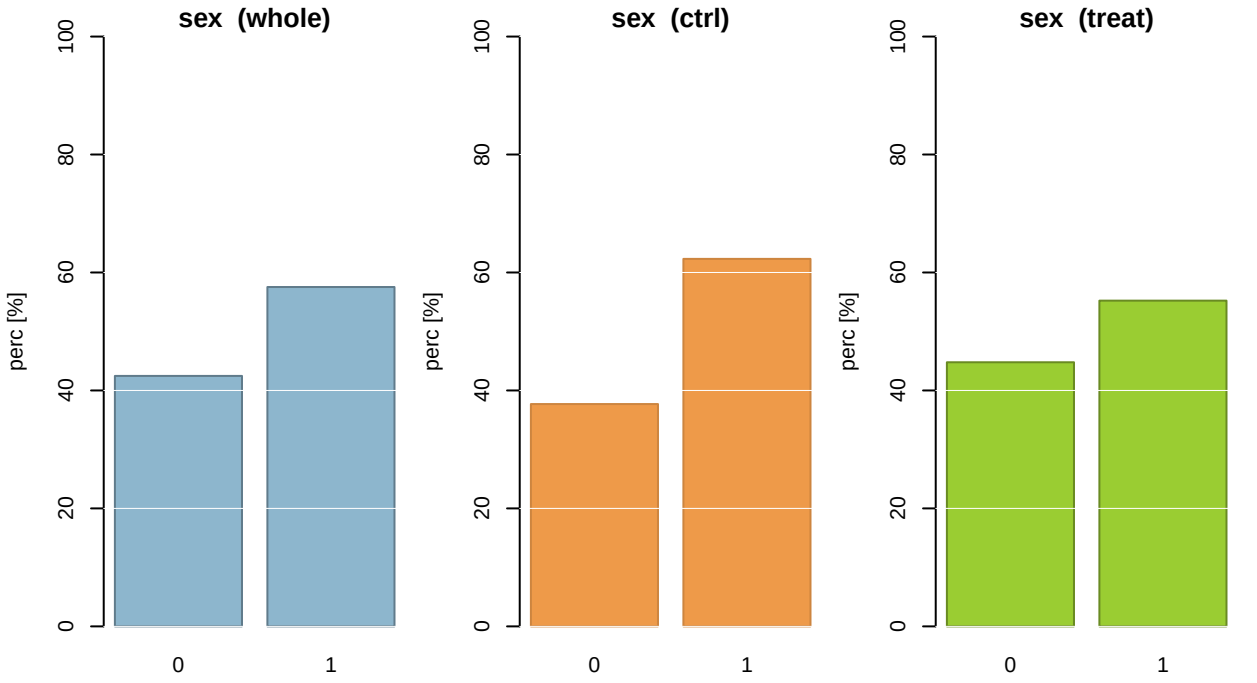
Notiamo che i soggetti sono stati divisi in rapporto 2/3 tra trattati (67%) e controlli (33%), in base agli assegnamenti riportati nella variabile Z (Z=1 indica trattamento)

```
# For each variable, calculate the mean for the whole sample
# and within each treatment group.

names_b = c('sex','race','nonmarried','Z','employ6')

resume_b = ComputeStats(df, names_b, mean)
resume_b = round(resume_b, 2) # per fit nella pagina
print(data.frame(resume_b))
```

```
##              sex race nonmarried    Z employ6
## mean(whole) 0.58 0.18       0.60 0.67    0.60
## mean(ctrl)  0.62 0.13       0.57 0.00    0.55
## mean(treat) 0.55 0.20       0.62 1.00    0.63
```

df$sex (0=maschio) La ripartizione tra controlli e trattati mantiene grossomodo la proporzione del campione complessivo, seppure la differenza tra le due percentuali sia leggermente inferiore tra i trattati (e di conseguenza maggiore tra i controlli). Si osserva una maggioranza di donne (60% circa) rispetto agli uomini: trattandosi di un esperimento randomizzato ed essendo rivolto a dei disoccupati, il dato presumibilmente riflette la discrepanza socio-lavorativa tra maschi e femmine.

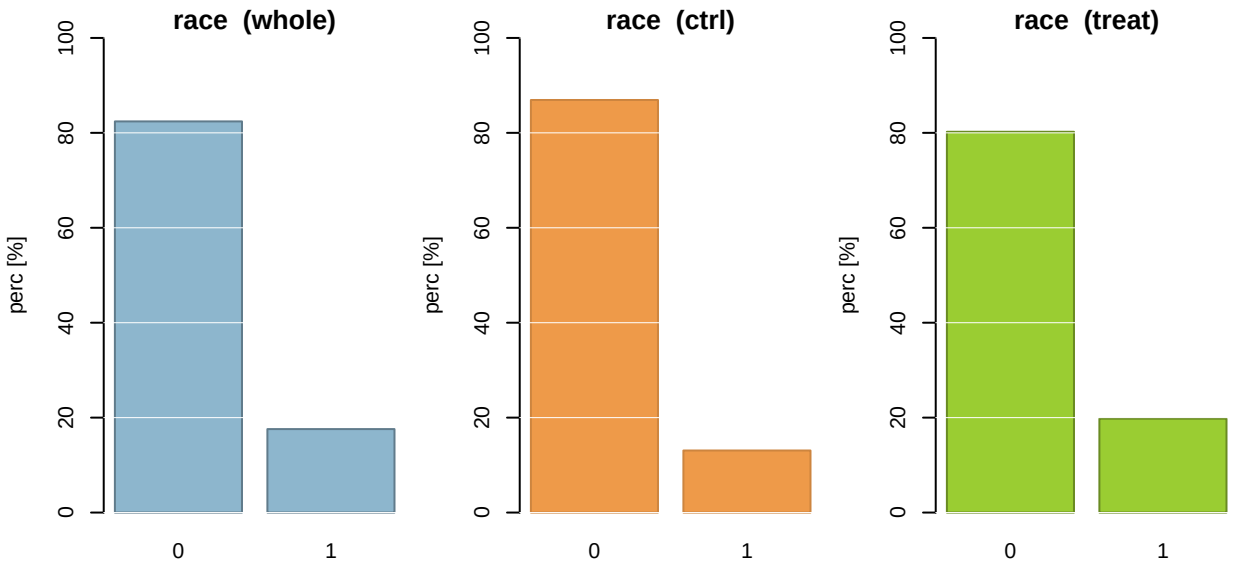
```
SummarizeVisualData(df, 'sex', binary=T)
```



df$race (0=bianco) Le proporzioni tra trattati e controlli tende a rispecchiare abbastanza bene quanto

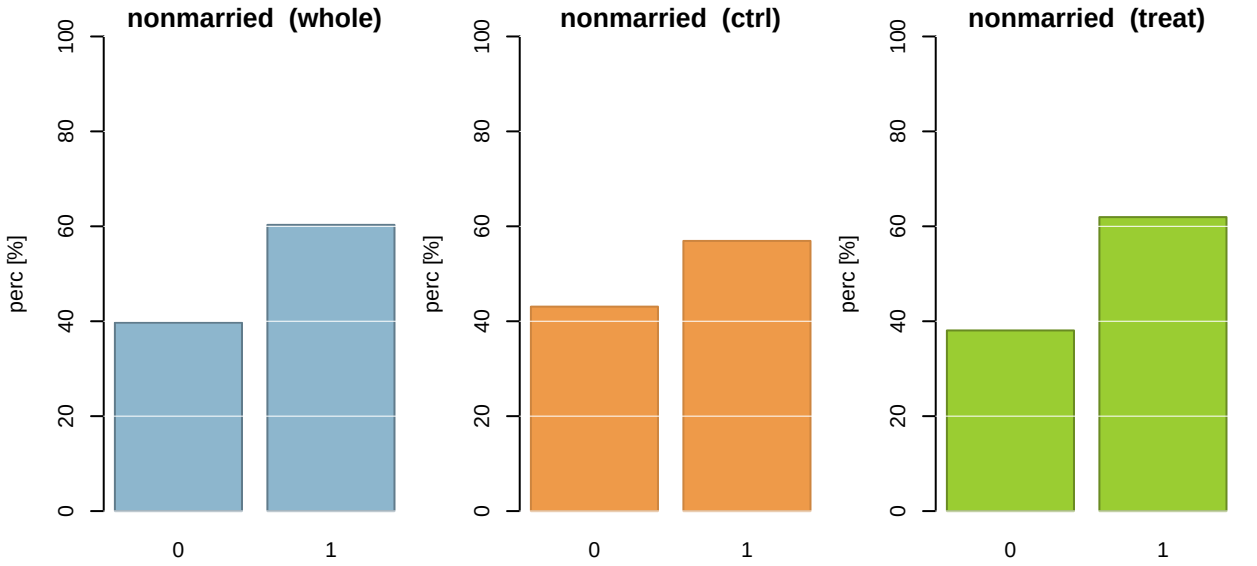
osservato sull'intero campione, dato che si tratta di un esperimento randomizzato. Il progetto Jobs II risale a metà degli anni '90; le persone bianche negli USA nel 1990 erano circa il 76% della popolazione (https://it.wikipedia.org/wiki/Bianchi_americani). Il campione riflette abbastanza la percentuale del Paese, anche se leggermente polarizzato verso persone bianche mentre sarebbe stato forse più plausibile che i disoccupati non bianchi fossero di più.

```
SummarizeVisualData(df, 'race', binary=T)
```



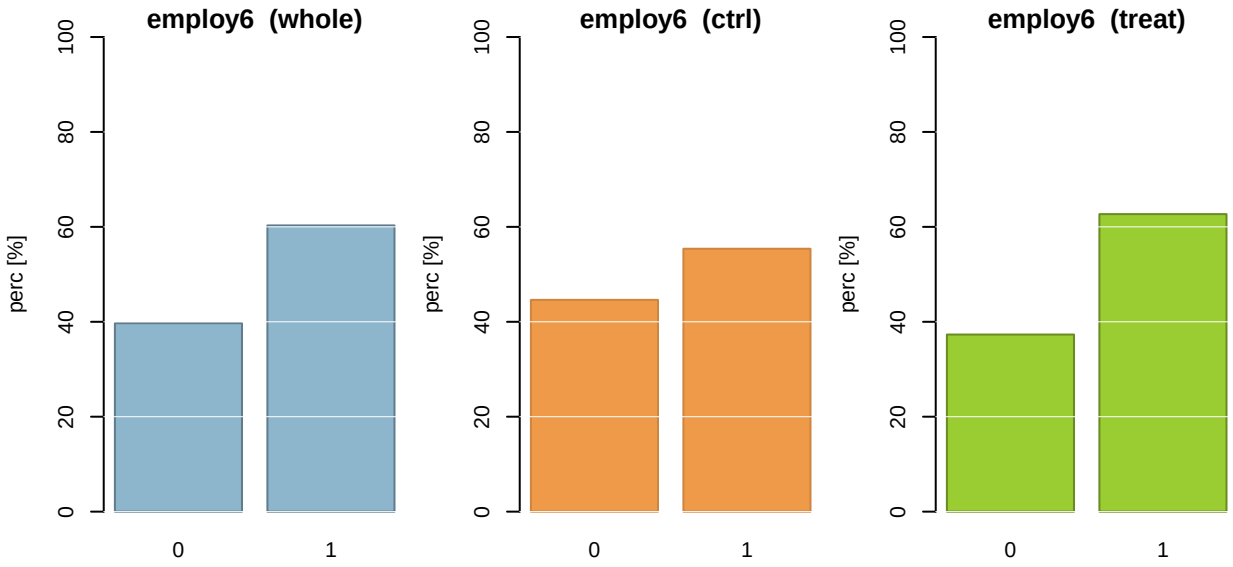
df$nonmarried (0=sposato) Il campione complessivo riporta una maggioranza di persone non sposate (60%), fatto abbastanza plausibile; le proporzioni sono rispettate anche nello split tra controlli e trattati.

```
SummarizeVisualData(df, 'nonmarried', binary=T)
```



df$employ6 (0=disoccupato) Dopo 6 mesi dal trattamento, oltre il 60% delle persone trattate ha trovato lavoro; tra i controlli la percentuale è invece inferiore al 60%: come prima impressione, trattandosi di un esperimento randomizzato, il progetto sembra avere esiti benefici.

```
SummarizeVisualData(df, 'employ6', binary=T)
```



Variabili continue

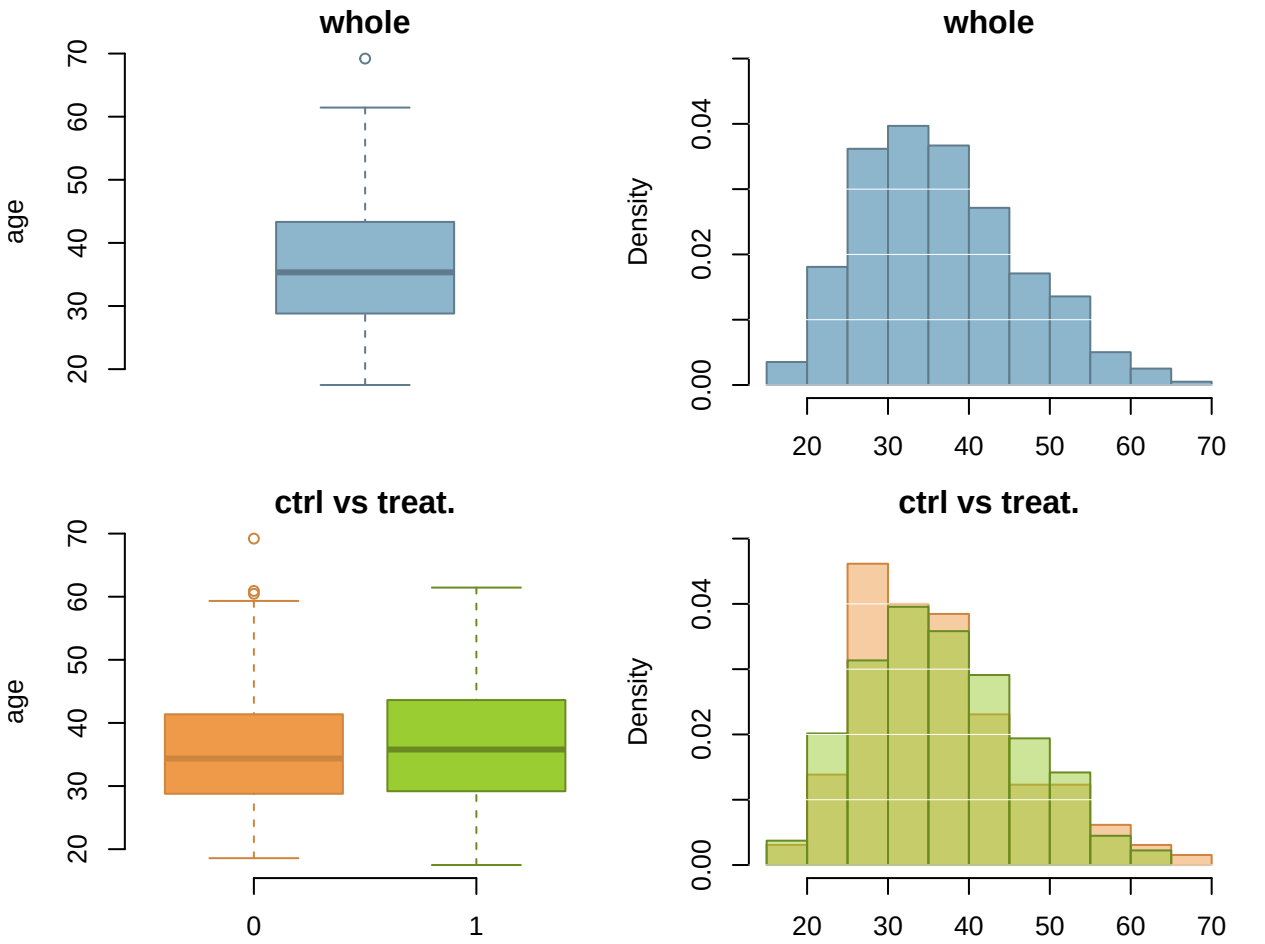
```
# For continuous covariates, also report the medians, standard deviation
# and ranges within each treatment group

names_c = c('age','educ','EconHard','assertive','motivation','depress0','depress6')

mean_c = ComputeStats(df, names_c, mean)
median_c = ComputeStats(df, names_c, median)
sd_c = ComputeStats(df, names_c, sd)
min_c = ComputeStats(df, names_c, min)
max_c = ComputeStats(df, names_c, max)

resume_c = rbind(mean_c, median_c, sd_c, min_c, max_c)
resume_c = round(resume_c, 2) # per fit nella pagina
print(data.frame(resume_c))
```

```
##                  age  educ EconHard assertive motivation depress0 depress6
## mean(whole)    36.52 13.36     3.62      3.05       5.33     2.45     2.06
## mean(ctrl)     36.28 13.28     3.47      3.01       5.32     2.49     2.15
## mean(treat)    36.65 13.40     3.70      3.06       5.33     2.44     2.01
## median(whole)  35.34 13.00     3.67      3.00       5.50     2.45     1.91
## median(ctrl)   34.36 13.00     3.33      3.00       5.50     2.45     2.00
## median(treat)  35.79 13.00     3.67      3.00       5.50     2.45     1.82
## sd(whole)       9.76  2.02     0.85      0.91       0.81     0.30     0.76
## sd(ctrl)        9.90  2.00     0.92      0.88       0.81     0.29     0.79
## sd(treat)       9.70  2.03     0.80      0.93       0.81     0.30     0.74
## min(whole)     17.49  8.00     1.33      1.00       3.50     1.82     1.00
## min(ctrl)      18.57  8.00     1.33      1.00       3.50     1.82     1.00
## min(treat)     17.49  8.00     1.33      1.00       3.50     1.91     1.00
## max(whole)     69.20 17.00     5.00      5.00       6.50     3.00     4.73
## max(ctrl)      69.20 17.00     5.00      5.00       6.50     3.00     4.73
## max(treat)     61.44 17.00     5.00      5.00       6.50     3.00     4.36
```

df$age L'età dei disoccupati nel campione varia tra un minimo di 17.49 e un massimo di 69.20, con un picco tra i 30 ed i 35 anni. Nel gruppo di controllo trale picco è leggermente anticipato, tra i 25 ed i 30 anni, ma

grossomodo la ripartizione tra i due gruppi rispecchia la distribuzione complessiva.

```
SummarizeVisualData(df, 'age', binary=F)
```

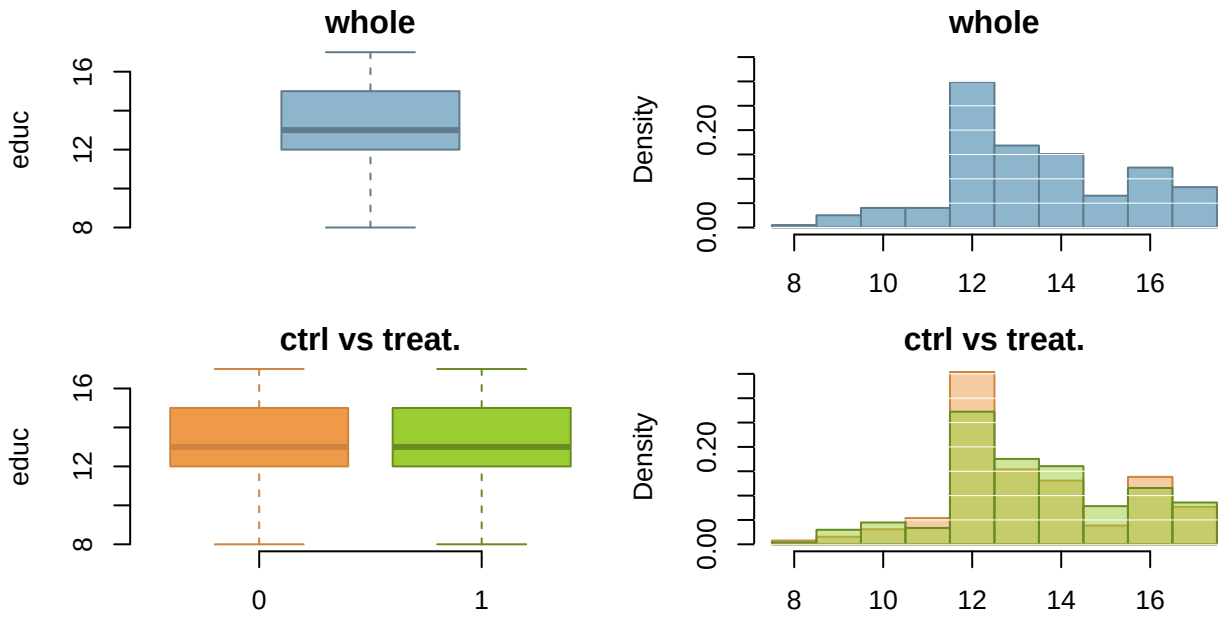


df$educ I livelli di educazione variano da 8 a 17, con un picco a 11. La ripartizione tra controlli e trattati è eseguita in modo bilanciato, come testimoniato da medie, dev. std. e boxplot seguenti.

```
vals = sort(unique(df$edu)); delta=sort(unique(vals[2:length(vals)]-vals[1:length(vals)-1]))
print(delta)
```

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

```
delta = 1; # 2*IQR(df$edu)*(length(df$edu))^(-1/3)
breaks = seq(min(vals)-delta/2, max(vals)+delta/2, by=delta)

SummarizeVisualData(df, 'educ', binary=F, breaks=breaks)
```

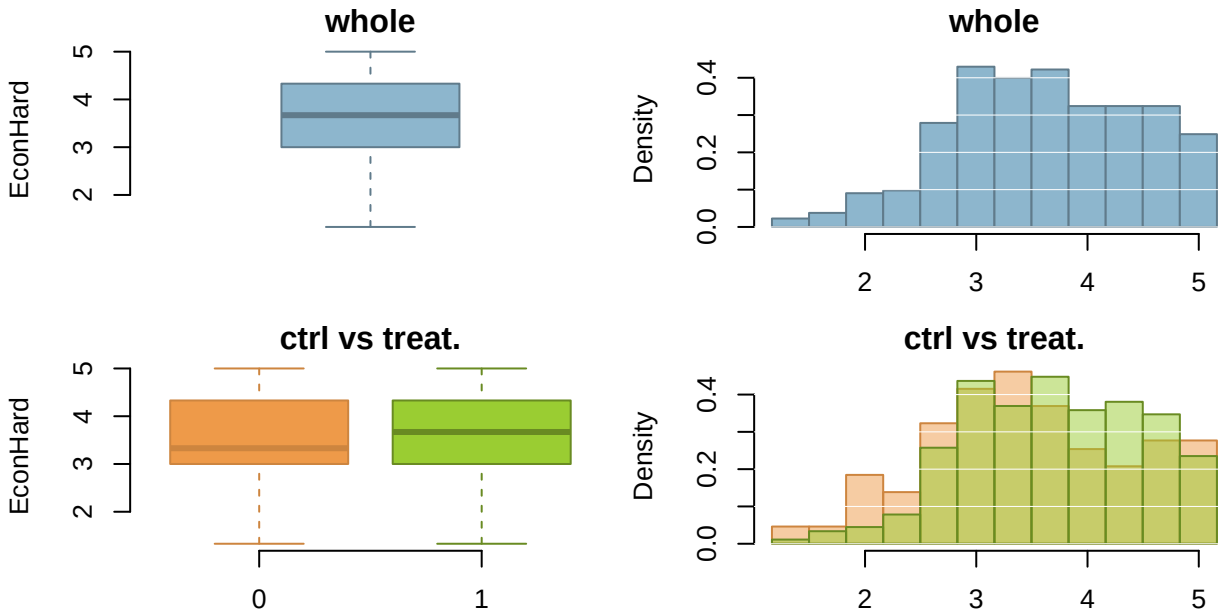


df$EconHard Gli indici relativi alle difficoltà economiche variano tra un minimo di 1.33 ad un massino di 5: dalle medie e mediane, si osserva nel gruppo di controllo dei valori leggermente più bassi rispetto ai trattati, ma meno concentrati (dev. std. maggiore per i controlli). Dai grafici si nota che gli indici sono più numerosi per valori superiori a 2.5

```
vals = sort(unique(df$EconHard)); delta=sort(unique(vals[2:length(vals)]-vals[1:length(vals)-1]))
print(delta)
```

```
## [1] 0.3299999 0.3300000 0.3399999 0.3400002
```

```
delta = 1/3; #2*IQR(df$EconHard)*(length(df$EconHard))^(-1/3)
breaks = seq(min(vals)-delta/2, max(vals)+delta, by=delta)

SummarizeVisualData(df, 'EconHard', binary=F, breaks=breaks)
```

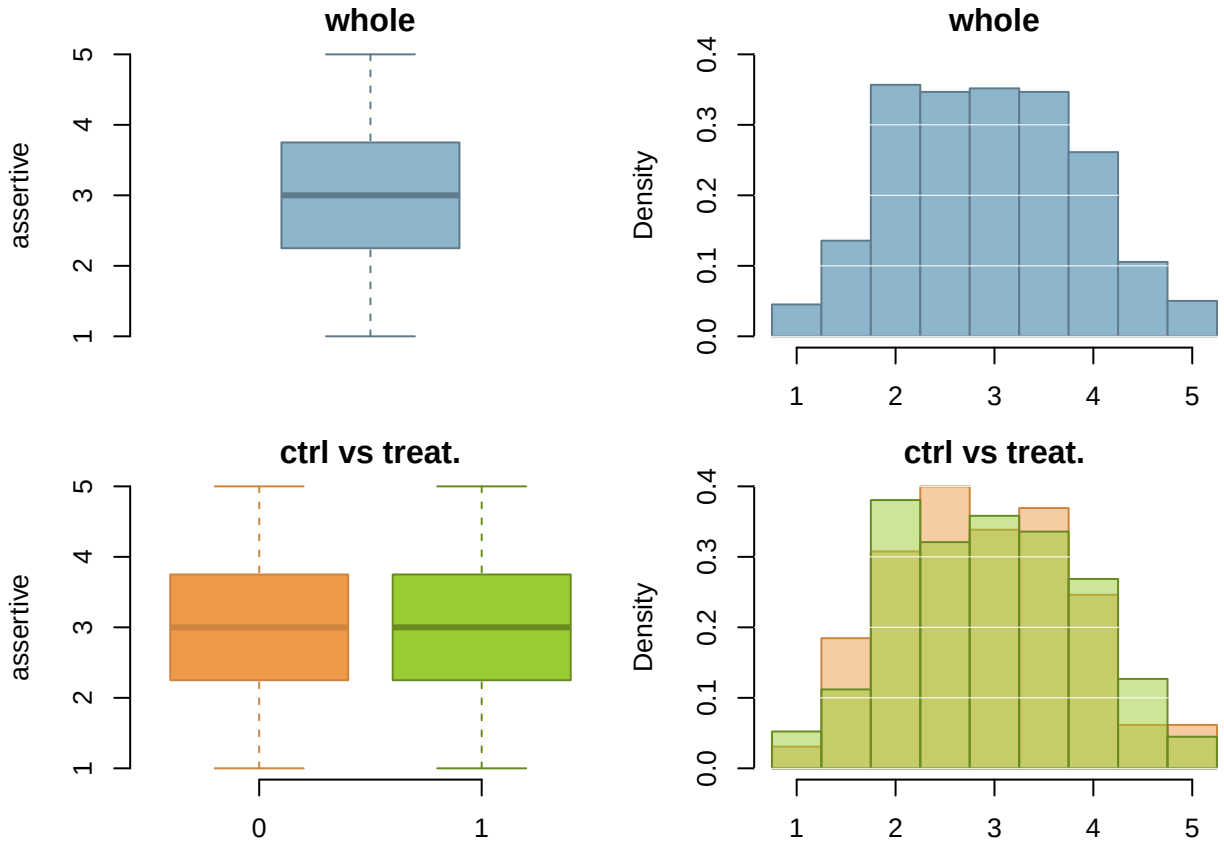


df$assertive I valori di assertività variano da 1 a 5, distribuiti con un livello piuttosto uniforme tra 2 e 4. La corrispondenza tra media, mediana e boxplot indica una ripartizione equilibrata tra controlli e trattati.

```
vals = sort(unique(df$assertive)); delta=sort(unique(vals[2:length(vals)]-vals[1:length(vals)-1]))
print(delta)
```

```
## [1] 0.07999992 0.08000004 0.16999996 0.17000008 0.25000000
```

```
delta = 0.5; #2*IQR(df$assertive)*(length(df$assertive))^(-1/3)
breaks = seq(min(vals)-delta/2, max(vals)+delta, by=delta)

SummarizeVisualData(df, 'assertive', binary=F, breaks=breaks)
```

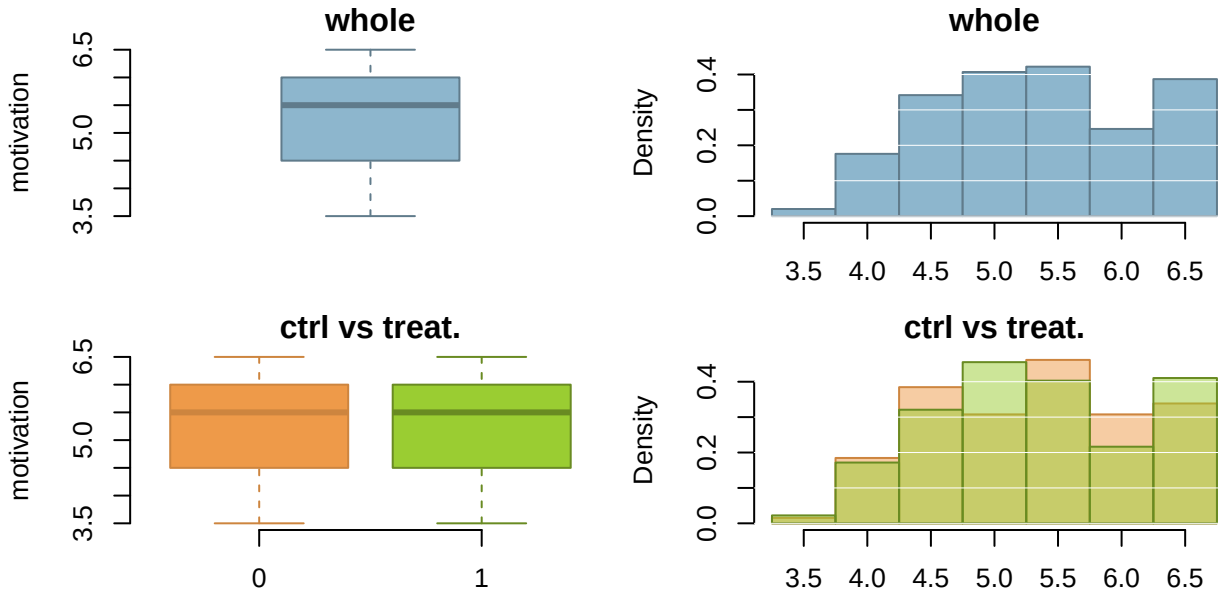


df$motivation I valori di motivazione variano da 3.5 a 5, distribuiti con un livello piuttosto uniforme per valori sopra 4. La corrispondenza tra media, mediana e boxplot indica una ripartizione equilibrata tra controlli e trattati.

```
vals = sort(unique(df$motivation)); delta=sort(unique(vals[2:length(vals)]-vals[1:length(vals)-1]))
print(delta)
```

```
## [1] 0.5
```

```
delta = 0.5; # 2*IQR(df$motivation)*(length(df$motivation))^(-1/3)
breaks = seq(min(vals)-delta/2, max(vals)+delta, by=delta)

SummarizeVisualData(df, 'motivation', binary=F, breaks)
```

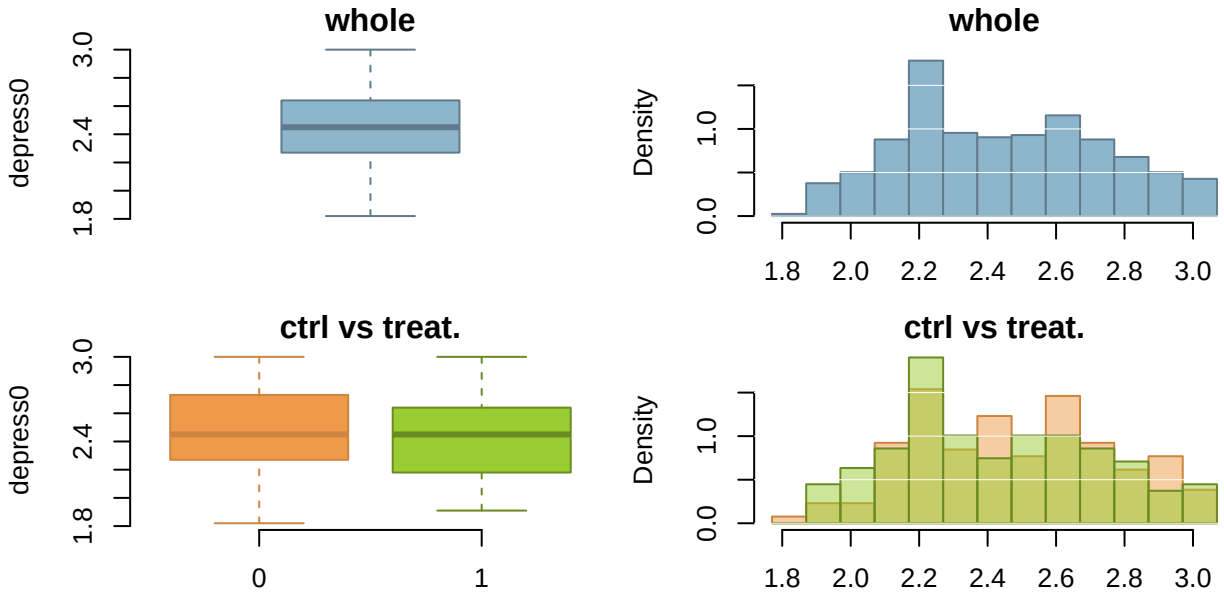


df$depress0 Gli indici di depressione all'inizio dello studio variano da 1.82 a 3: a parte un picco intorno a 2.2 (evidenziato dagli istogrammi), i valori si distribuiscono piuttosto uniformemente tra 2 e 3; sebbene le mediane si equivalgano, i boxplot tra controlli e trattati evidenziano una leggera asimmetria nella distribuzione dei controlli

```
vals = sort(unique(df$depress0)); delta=sort(unique(vals[2:length(vals)]-vals[1:length(vals)-1]))
print(delta)
```

```
##  [1] 0.00999999 0.01999998 0.02999997 0.03999996 0.04000020 0.04999995
##  [7] 0.05999994 0.06999993 0.08000016 0.08999991 0.09000003 0.09999990
```

```
delta = 0.1; #2*IQR(df$depress0)*(length(df$depress0))^(-1/3)
breaks = seq(min(vals)-delta/2, max(vals)+delta, by=delta)

SummarizeVisualData(df, 'depress0', binary=F, breaks)
```



df$depress6 Gli indici di depressione dopo 6 mesi mostrano dei valori che si concentrano su valori più bassi (tra 1 e 3) con picchi tra 1.5 e 2; tuttavia esistono anche valori superiori a 3 che non erano stati misurati

all'inizio dello studio, forse dovuti a persone ancora senza impiego a 6 mesi dall'iniziativa. L'istogramma per i trattati sembra più spostato verso sinistra rispetto a quello dei controlli, lasciando ben sperare per la riuscita dell'iniziativa (dato che si tratta di uno studio randomizzato).

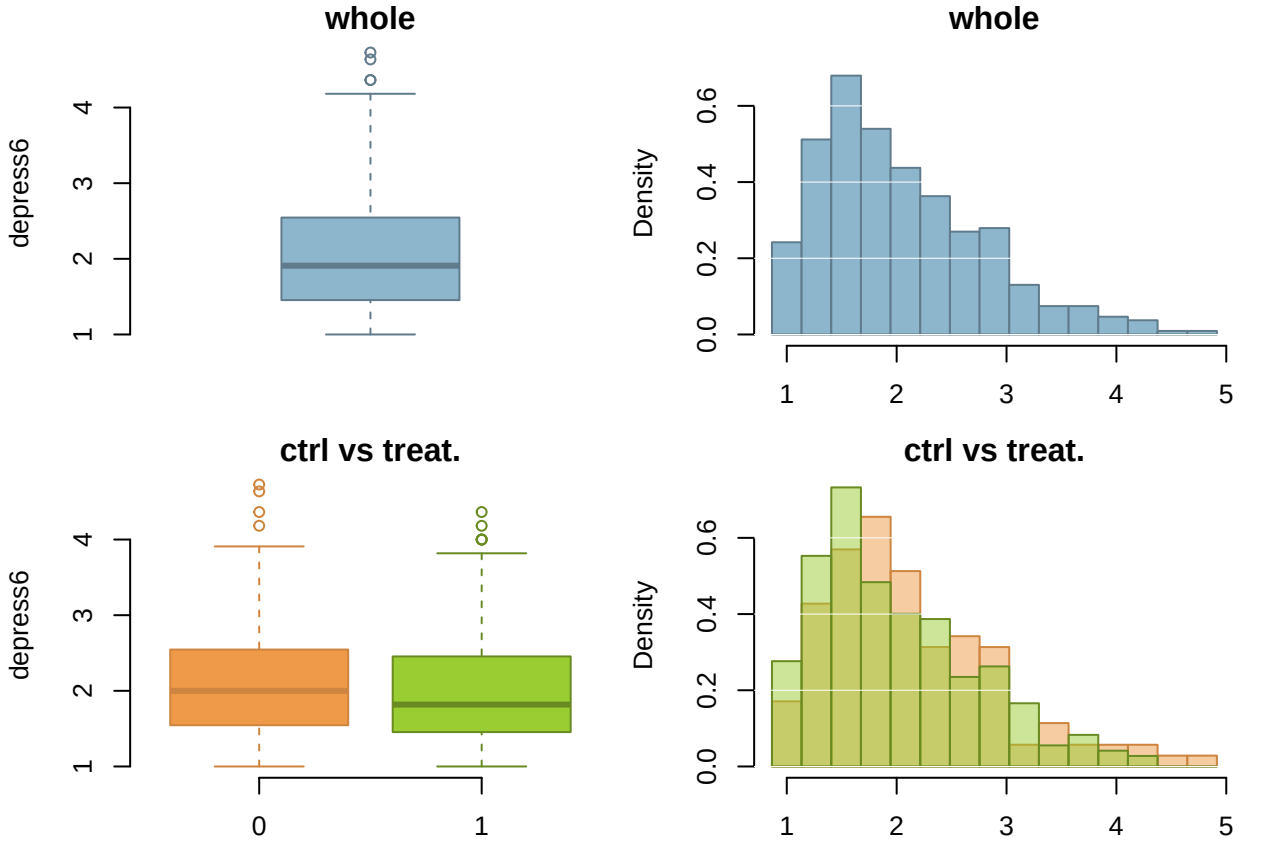
```
vals = sort(unique(df$depress6)); delta=sort(unique(vals[2:length(vals)]-vals[1:length(vals)-1]))
print(delta)
```

```
## [1] 0.09090900 0.09090912 0.09090924 0.18181801 0.18181849 0.27272701
```

```
delta = 0.27;
2*IQR(df$depress6)*(length(df$depress6))^(-1/3)
```

```
## [1] 0.290434
```

```
breaks = seq(min(vals)-delta/2, max(vals)+delta, by=delta)

SummarizeVisualData(df, 'depress6', binary=F, breaks)
```



Modello Beta-binomiale (esercizio 3)

Let π denote the probability of re-employment, n_c the number of individuals assigned to the control condition, and n_t the number of individuals assigned to the treatment condition. Compute the two posterior distributions, $p(\pi|x_1(0)\dots x_{n_c}(0))$ and $p(\pi|x_1(1)\dots x_{n_t}(1))$, under the exchangeability assumption. Represent and compare these posterior distributions, including the posterior means and 90% credible intervals. Comment on the results in a few sentences.

Posta π la probabilità di reimpiego, si considera il modello coniugato Beta-binomiale per valutare $p(\pi|\mathbf{y})$, la distribuzione di probabilità a posteriori di π: il modello è adeguato allo scenario considerato, in virtù della natura "Bernoulliana" che un individuo venga reimpiegato in un lavoro e dell'ipotesi fatta a priori su π

(Beta$(1,1)$ significa che a priori π è distribuito uniformemente tra 0 e 1)

Pertanto, in virtù del teorema di Bayes, si vuole valutare

$$p(\pi|\mathbf{y}) = \frac{p(\mathbf{y}|\pi)p(\pi)}{p(\mathbf{y})} \tag{1}$$

$$p(\mathbf{y}|\pi) = \pi^{N_s}(1-\pi)^{N-N_s} \tag{2}$$

$$\pi \sim \text{Beta}(1,1) \tag{3}$$

Trattandosi di un modello coniugato, la distribuzione a posteriori ha forma

$$\pi|\mathbf{y} \sim \text{Beta}(a_{\text{post}}, b_{\text{post}})$$
$$a_{\text{post}} = 1 + N_s$$
$$b_{\text{post}} = 1 + N - N_s$$

Gruppo di controllo

Il gruppo di controllo è composto da $N = 130$ persone; di esse $N_s = 72$ sono state reimpiegate: la distribuzione di probabilità a posteriori è quindi

$$\pi|\mathbf{y} \sim \text{Beta}(1+N_s, 1+N-N_s) = \text{Beta}(73, 59) \tag{4}$$

```
idc = which(df$Z==0)
idt = which(df$Z==1)

Nc = length(idc)
yc = df$employ6[idc]
Nsc = sum(yc)
print(c(Nc, Nsc))
```

```
## [1] 130  72
```

```
colC = 'tan3'
apost_c = 1+Nsc
bpost_c = 1+Nc-Nsc
print(c(apost_c, bpost_c))
```

```
## [1] 73 59
```

```
expected_value_c <- apost_c / (apost_c + bpost_c)
print(expected_value_c)
```

```
## [1] 0.5530303
```

```
qLc <- qbeta(0.05, shape1 = apost_c, shape2 = bpost_c)
qUc <- qbeta(0.95, shape1 = apost_c, shape2 = bpost_c)
print((c(qLc,qUc)))
```

```
## [1] 0.4816258 0.6235175
```

Gruppo trattato

Il gruppo trattato è composto da $N = 268$ persone; di esse $N_s = 168$ sono state reimpiegate: la distribuzione di probabilità a posteriori è quindi

$$\pi|\mathbf{y} \sim \text{Beta}(1+N_s, 1+N-N_s) = \text{Beta}(169, 101) \tag{5}$$

```r
Nt = length(idt)
yt = df$employ6[idt]
Nst = sum(yt)
print(c(Nt, Nst))
```

```
## [1] 268 168
```

```r
colT = 'olivedrab4'
apost_t = 1+Nst
bpost_t = 1+Nt-Nst
print(c(apost_t, bpost_t))
```

```
## [1] 169 101
```

```r
expected_value_t <- apost_t / (apost_t + bpost_t)
print(expected_value_t)
```

```
## [1] 0.6259259
```

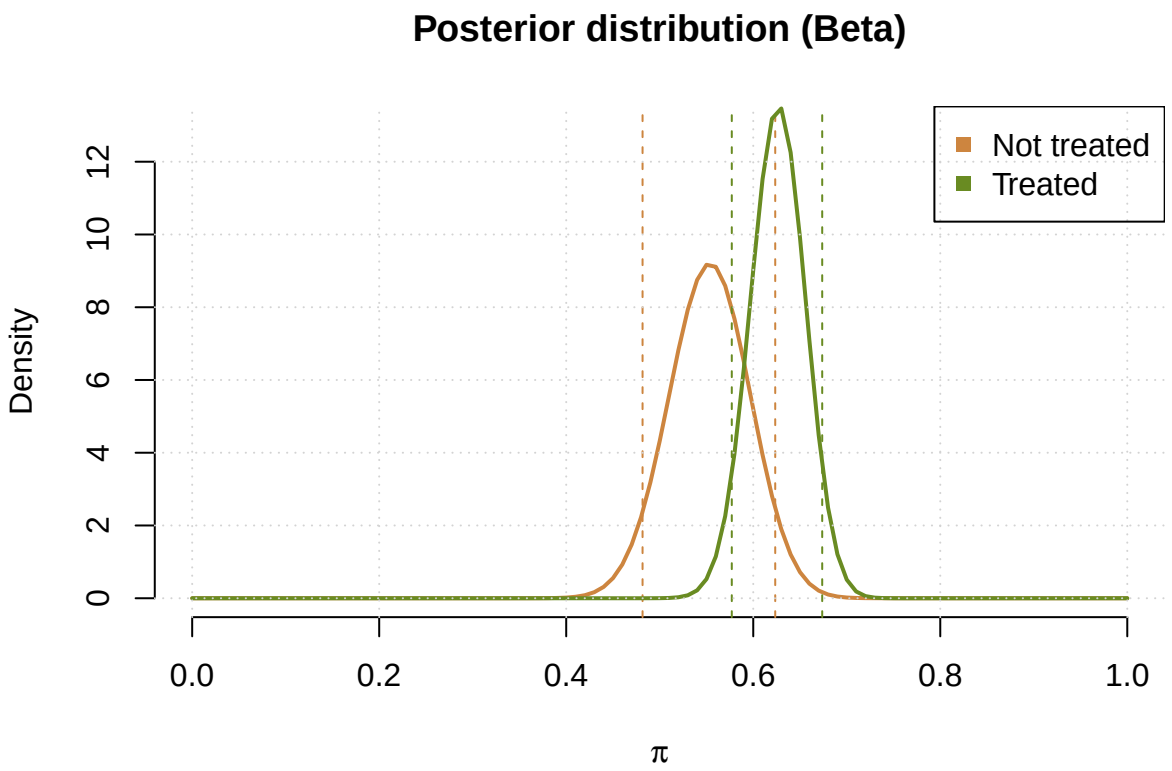
```r
qLt <- qbeta(0.05, shape1 = apost_t, shape2 = bpost_t)
qUt <- qbeta(0.95, shape1 = apost_t, shape2 = bpost_t)
print((c(qLt,qUt)))
```

```
## [1] 0.5770390 0.6737502
```

```r
library(latex2exp) #per label/titoli con LaTeX
```

```
## Warning: package 'latex2exp' was built under R version 4.3.3
```

```r
CompareBetaPosterior <- function(apost_c, bpost_c, colC, apost_t, bpost_t, colT)
{
   curve(dbeta(x, shape1 = apost_c, shape2 = bpost_c),
         from = 0, to = 1, ylim=c(0,13),
         main = paste('Posterior distribution (Beta)'),
         xlab=TeX(r'(\pi)'), ylab = 'Density', frame.plot = F,
         col = colC, lwd = 2)
   abline(v = qLc, col = colC, lty = "dashed", lwd = 1)
   abline(v = qUc, col = colC, lty = "dashed", lwd = 1)
   #
   curve(dbeta(x, shape1 = apost_t, shape2 = bpost_t),
         from = 0, to = 1, add=T,
         col = colT, lwd = 2)
   abline(v = qLt, col = colT, lty = "dashed", lwd = 1)
   abline(v = qUt, col = colT, lty = "dashed", lwd = 1)
   grid(lty='dotted')
   legend('topright', c('Not treated', 'Treated'), col=c(colC,colT), pch=15)
}

CompareBetaPosterior(apost_c, bpost_c, colC, apost_t, bpost_t, colT)
```



Le distribuzioni a posteriori stimate per la probabilità di reimpiego π mostrano che nel caso degli individui trattati la parte significativa della distribuzione è collocata in range maggiori (e con margini più stretti) rispetto alla stessa probabilità stimata per il gruppo di controllo:

- per i trattati il valor medio di π vale 0.626, e l'intervallo di credibilità al 90% vale (0.577 0.674)
- per i controlli il valor medio di π vale 0.553, e l'intervallo di credibilità al 90% vale (0.482 0.624)

Quindi ne deduciamo che le osservazioni mostrano una maggior probabilità per chi ha eseguito il trattamento di trovare un reimpiego.

Metodo MCMC (esercizio 4)

Bayesian model-based analysis for the outcome variable "depression six months after the intervention assignment". Derive the posterior distributions of the finite sample average causal effect and the super-population average causal effect. Plot the resulting posterior distributions in a histogram and report the following summary statistics of the resulting posterior distributions: mean, standard deviation, median, 2.5% and 97.5% percentiles.

Ritenendo valida la mancanza d'interazione tra il gruppo di controllo ed i trattati, ed ipotizzando uniformità d'insegnamento da parte dei docenti per i soggetti nel gruppo trattato, ci poniamo sotto l'assunzione SUTVA.

Si vuole determinare la distribuzione di probabilità a posteriori per l'effetto medio causale per il campione finito dei dati (SATE $\bar{Y}(1) - \bar{Y}(0)$) e per la super-popolazione (PATE $\mathbb{E}(Y(1) - Y(0))$).

Trascurando il problema di noncompliance, si fa riferimento alla variabile Z per distinguere tra chi ha effettuato il trattamento attivo e chi quello di controllo.

Gibbs Sampler

Avendo specificato media e varianza in modo indipendente, a tale scopo viene implementato un Gibbs Sampler per produrre un campionamento sia dei parametri del modello che dei valori mancanti nei risultati potenziali,

tenendo conto che i parametri sono relativi a valori in scala logaritmica a differenza degli estimands che devono essere invece calcolati sulla scala originale.

```
library(extraDistr)
```

```
## Warning: package 'extraDistr' was built under R version 4.3.3
```

```
GibbsBasic <- function(niter, nburn, sampling=1, prior_c, prior_t, Yobs, W,
                       rho=0, seed=NULL)
{
   if(is.null(seed)==F){ set.seed(seed) }

   idc = which(W==0)
   idt = which(W==1)

   N_c = length(idc)
   N_t = length(idt)

   Yl = log(Yobs) #devo inizializzare theta sul logaritmo delle Y osservate
   Y_c = Yl[idc]
   Y_t = Yl[idt]

   # perturbo la deviazione standard per non rischiare
   # di avere varianze negative
   theta <- list(mu_c = mean(Y_c) + rnorm(1,0, 1),
                 mu_t =  mean(Y_t) + rnorm(1,0, 1),
                 sigma2_c =  (sd(Y_c)  + rnorm(1,0, 1))^2,
                 sigma2_t =  (sd(Y_t)  + rnorm(1,0, 1))^2)
                 # sigma2_c =  (var(Yobs[idc])  + rnorm(1,0, 1)),
                 # sigma2_t =  (var(Yobs[idt])  + rnorm(1,0, 1)))

   #inizializzo i potential outcomes osservati
   Y0 = Y1 = NULL
   Y0[idc] = Yobs[idc]
   Y1[idt] = Yobs[idt]


   draws<- seq((nburn+1), niter, by=sampling)
   ndraws<- length(draws)

   estimands<- matrix(NA, ndraws, 2)
   colnames(estimands)<- c('sate', 'pate')

   j = 0
   for(l in 1:niter)
   {
      theta$mu_c = ApplyFullConditional_mu(prior_c, N_c, Y_c, theta$sigma2_c)
      theta$sigma2_c = ApplyFullConditional_sigma2(prior_c, N_c, Y_c, theta$mu_c)

      theta$mu_t = ApplyFullConditional_mu(prior_t, N_t, Y_t, theta$sigma2_t)
      theta$sigma2_t = ApplyFullConditional_sigma2(prior_t, N_t, Y_t, theta$mu_t)

      #GEstione del logaritmo sugli osservati

      if(sum(l == draws)==1)
```

```
      {
         # imputo i potential outcomes che sono missing
         if(rho==0)
         {
            # applico exp per rimuovere il logaritmo
            Y0[idt] = exp(rnorm(N_t, theta$mu_c, sqrt(theta$sigma2_c)))
            Y1[idc] = exp(rnorm(N_c, theta$mu_t, sqrt(theta$sigma2_t)))
         }
         else
         {
            mu0t = theta$mu_c +
               {rho*sqrt(theta$sigma2_c/theta$sigma2_t)}*{Yl[idt]-theta$mu_t}
            sigma0t = (1-rho^2)*sqrt(theta$sigma2_c)
            Y0[idt] = exp(rnorm(Nt, mu0t, sigma0t)) # applico exp per rimuovere il log

            mu1c = theta$mu_t +
               {rho*sqrt(theta$sigma2_t/theta$sigma2_c)}*{Yl[idc]-theta$mu_c}
            sigma1c = (1-rho^2)*sqrt(theta$sigma2_t)
            Y1[idc] = exp(rnorm(Nc, mu1c, sigma1c)) # applico exp per rimuovere il log
         }

         j = j+1
         estimands[j,'sate'] = mean(Y1)-mean(Y0)
         # rimuovo il logaritmo dal valore atteso
         estimands[j,'pate'] = ExpectedUnlog(theta$mu_t, theta$sigma2_t) -
                               ExpectedUnlog(theta$mu_c, theta$sigma2_c)
      }
   }

   return(estimands)
}

ApplyFullConditional_mu <- function(prior, N, Y, sigma2)
{
   precision = 1/prior$varsigma2 + N/sigma2
   varsigma2 = 1/precision
   nu = (prior$nu/prior$varsigma2 + sum(Y)/sigma2)/precision

   mu = rnorm(1, mean= nu, sd=sqrt(varsigma2))
   return (mu)
}

ApplyFullConditional_sigma2 <- function(prior, N, Y, mu)
{
   a = prior$a + N
   b2 = (prior$a*prior$b2 + sum((Y - mu)^2))/a

   sigma2 = rinvchisq(1, a, b2)
   return (sigma2)
}

ExpectedUnlog <- function(mu, sigma2)
{
```

```
   return (exp(mu+sigma2/2))
}
```

Scelta dei parametri per le distribuzioni a priori

Impiegando le prior nella forma dell'esercizio, per $i = c, t$

$$\beta_i \sim \mathsf{N}(\nu_{0i}, \varsigma_i^2)$$
$$\sigma_i^2 \sim \mathsf{SI}_{\chi^2}(a_i, b_i^2)$$

si vuole ricondurre la parametrizzazione per σ_i^2 a quella nella forma di distribuzione invers-gamma, dove

$$\sigma_i^2 \sim \mathsf{IG}(\eta_{0i}/2, \sigma_{0i}^2\eta_{0i}/2)$$

con η_{0_i} indice di numerosità a priori per σ_i^2, mentre σ_{0i} costiuiscono le stime a priori per σ_i^2.

Sfruttando $\mathsf{SI}_{\chi^2}(a, b^2) = \mathsf{IG}(a/2, b^2a/2)$ ne deriva che, per le ipotesi fatte sulla invers-gamma, rispetto alla scaled inverse-χ^2 deve valere

$$\frac{a_i}{2} = \frac{\eta_{0i}}{2} \quad \Rightarrow \quad a_i = \eta_{0i}$$
$$b_i^2\frac{a_i}{2} = \sigma_{0i}^2\frac{\eta_{0i}}{2} \quad \Rightarrow \quad b_i^2 = \sigma_{0i}^2\frac{\eta_{0i}}{a_i} = \sigma_{0i}^2$$

Prior plausibile Per indicare una prior *plausibile*, tenuto conto che il livello di depressione nei soggetti è stato valutato all'inizio dello studio (variabile `depress0`), possiamo utilizzare tali osservazioni per farsi un'idea dei parametri da impiegare nelle distribuzioni a priori.

Inoltre, al fine di evitare "bias" (dato che viene impiegato come campione per le prior lo stesso gruppo di individui che prende parte all'esperimento), tale prior verrà resa poco informativa: si vuole usare il campione in `depress0` solo per non inserire dei valori puramente a caso, non avendo altri supporti nella conoscenza del fenomeno.

Infine, volendo restare il più generici possibile, le stime per ν_{0i} e σ_{0i} $(i = c, t)$ non vengono ricavate sepatamente diversificando tra controlli e trattati ma usando l'intero campione, applicando quindi la stessa prior ad entrambi i gruppi dato che l'esperimento è randomizzato.

Pertanto, indicando con $\hat{\nu}$ e s^2 rispettivamente media e varianza campionaria calcolate su `log(depress0)`, poniamo i seguenti parametri per le prior:

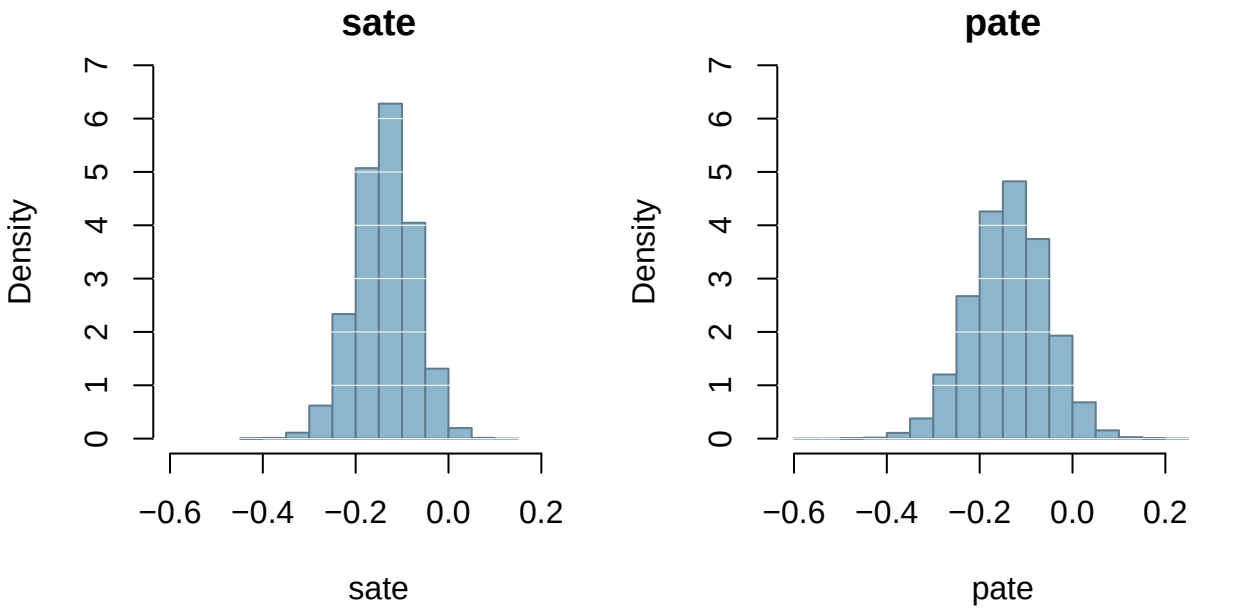
$$\nu_{0c} = \nu_{0t} = \hat{\nu}$$
$$a_c = a_t = 1$$
$$b_c^2 = b_t^2 = s^2$$

Per quanto riguarda ς_i, sempre per essere poco informativi poniamo $\varsigma_c = \varsigma_t = 10^2$.

```
yl = log(df$depress0)

nu_0 = mean(yl);
var_0=var(yl);

# prior POCO informativa, non diversificata tra controlli e trattati
prior_c = list(nu=nu_0, varsigma2 = 10^2, a=1, b2=var_0)
prior_t = list(nu=nu_0, varsigma2 = 10^2, a=1, b2=var_0)
```

```r
set.seed(42)
Niter = 102000
estimands = GibbsBasic(niter=Niter, nburn=2000, sampling=1,
                       prior_c, prior_t, Yobs=df[,'depress6'], W=df[,'Z'])
```

```r
par(mfrow=c(1,2), mar=c(4.5,4.0,1.5,0.5)) #margini bottom, left, top, right
hist(x=estimands[,'sate'], freq=F, col='lightskyblue3', border='lightskyblue4',
     xlim=c(-0.6,0.3), ylim=c(0,7), xlab='sate',main='sate')
grid(lty='solid', lwd=0.5, col='white', nx=NA, ny=NULL)
hist(x=estimands[,'pate'], freq=F, col='lightskyblue3', border='lightskyblue4',
     xlim=c(-0.6,0.3), ylim=c(0,7), xlab='pate',main='pate')
grid(lty='solid', lwd=0.5, col='white', nx=NA, ny=NULL)
```




```r
par(mfrow = c(1,1), mar=c(5,4,4,2) + 0.1)# Reset to default layout
```

Statistiche di riepilogo

Grazie alla libreria `coda`, con il solo comando `summary()` su `mcmc(estimands)` possiamo ricavare tutte le statistiche riassuntive delle distribuzioni posteriori risultanti in termini di media, deviazione standard, mediana, percentili del 2,5% e del 97,5%.

```r
library(coda)
```

```
## Warning: package 'coda' was built under R version 4.3.3
```

```r
gibbsResults = mcmc(estimands)
summary(gibbsResults)
```

```
##
## Iterations = 1:1e+05
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1e+05
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
```

```
##
##          Mean      SD  Naive SE Time-series SE
## sate -0.1370 0.06183 0.0001955      0.0001955
## pate -0.1374 0.08161 0.0002581      0.0002633
##
## 2. Quantiles for each variable:
##
##          2.5%     25%     50%      75%    97.5%
## sate -0.2623 -0.1777 -0.1356 -0.09480 -0.01918
## pate -0.3005 -0.1916 -0.1363 -0.08205  0.01945
```

```
# per "riprova"
(cbind(Mean=apply(estimands,2,mean), SD=apply(estimands,2,sd),
       Median= apply(estimands,2,median), t(apply(estimands, 2, quantile, probs=c( 0.025, 0.975)))))
```

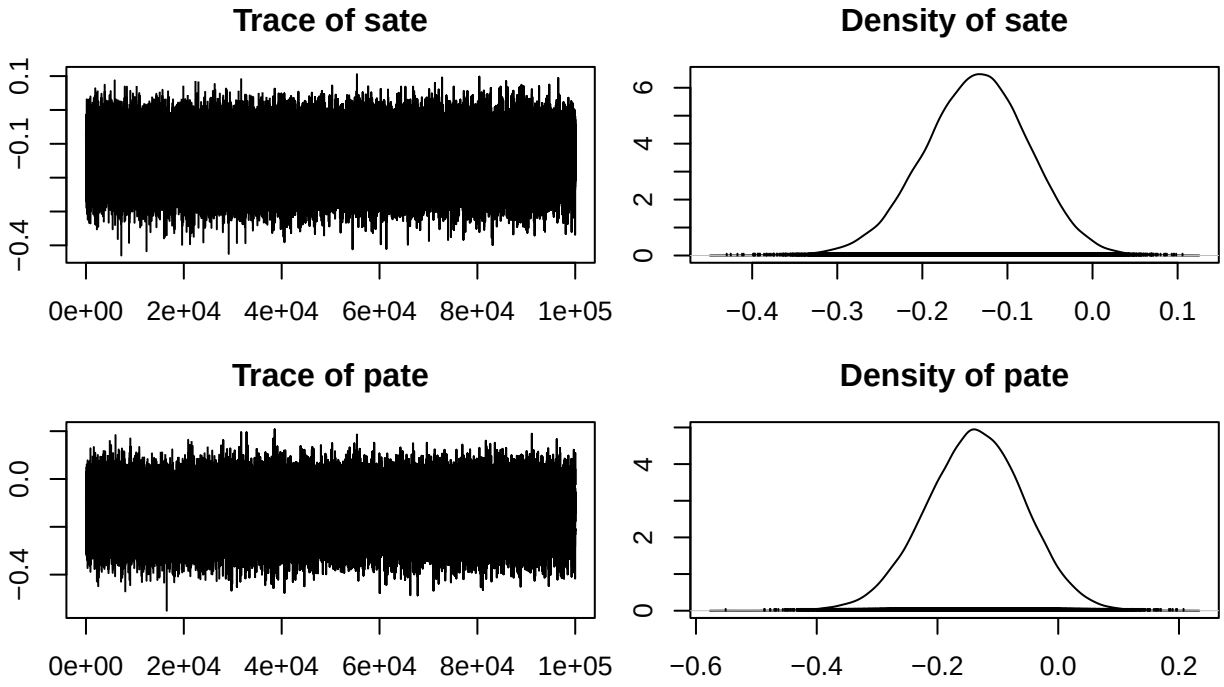
```
##             Mean         SD     Median      2.5%       97.5%
## sate -0.1369517 0.06183185 -0.1355671 -0.262327 -0.01918026
## pate -0.1374347 0.08161118 -0.1362930 -0.300519  0.01944679
```

Verifica della convergenza

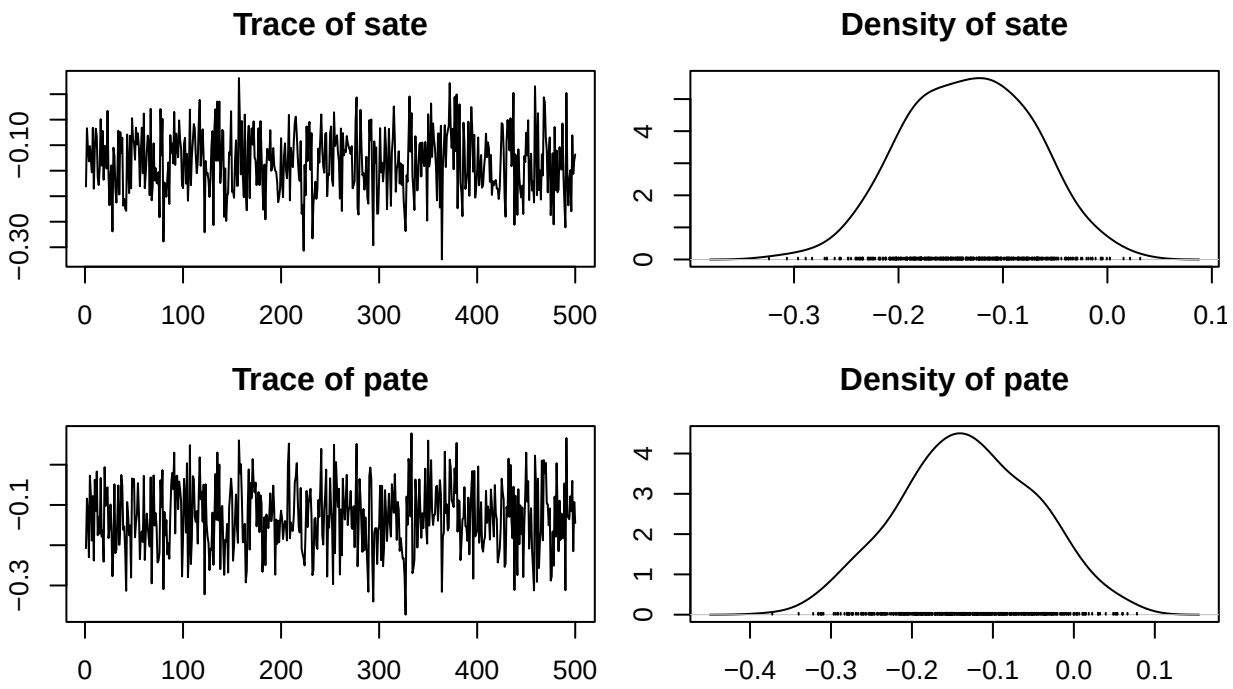
L'analisi visuale sulle tracce e autocorrelazione, assieme alla statistica di Geweke, concordano tutte sull'avvenuta convergenza della successione.

Ispezione visuale Il burnin di 2000 iterazioni sembra adeguato ad assicurare l'avvenuto andamento a regime della catena di Markov, non notando (specialmente nella parte iniziale) trend nella sequenza o tratti orizzontali.

```
par(mar=c(2,2,3,1)) #margini bottom, left, top, right
plot(gibbsResults)
```



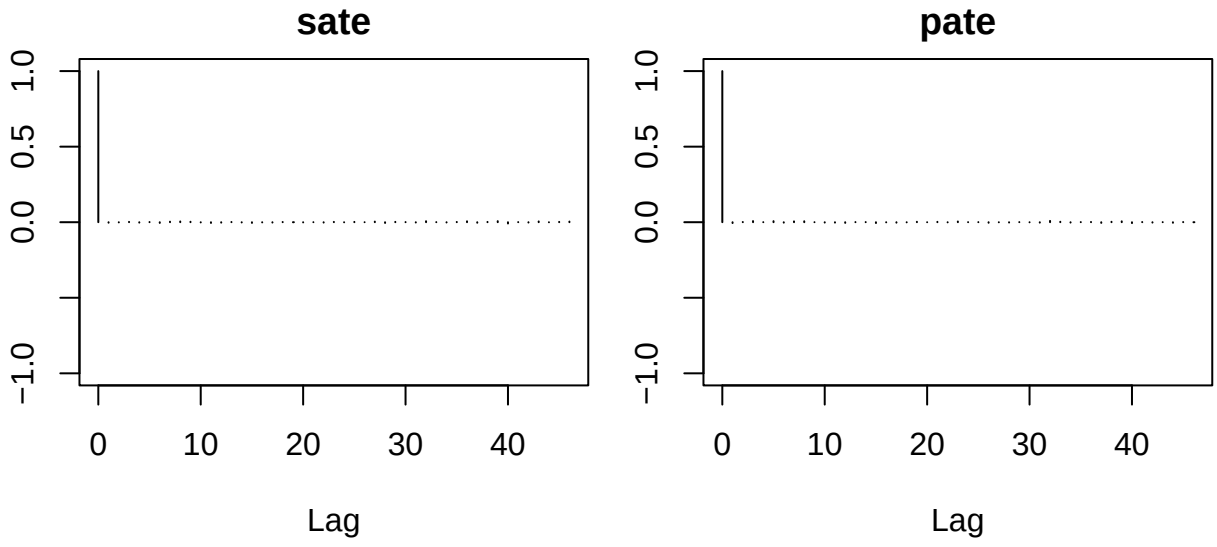

```
# visualizzo meglio la parte iniziale della sequenza
plot(mcmc(estimands[1:500,]))
```




```
par(mar=c(5,4,4,2) + 0.1) #Reset to default margins
```

Autocorrelazione Calcolando la correlazione della sequenza con se stessa a lag diversi, i valori piccoli che si ottengono per lag>1 indicano che ci sia poca dipendenza tra i campioni e li si possa considerare come iid.

```
par(mar=c(4.5,2,2,1)) #margini bottom, left, top, right
autocorr.plot(gibbsResults)
```




```
par(mar=c(5,4,4,2) + 0.1) #Reset to default margins
```

Test di Geweke Si verifica se le medie prese sulla prima parte (10%) e seconda metà della sequenza siano uguali o meno: se le medie non sono uguali allora i due blocchi non riproducono la stessa distribuzione e pertanto non siamo ancora a regime.

La statistica test per sate (0.8674) e pate (0.2503) rientra tra i quantili -1.96 e 1.96, pertanto si accetta

l'ipotesi nulla che le due medie siano uguali.

```
geweke.diag(gibbsResults)
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   sate   pate
## 0.8874 0.2503
```

Analisi di sensibilità

Avendo basato la specifica della prior sugli indici di depressione degli stessi soggetti nello studio, valutati all'avvio dell'esperimento, si ritiene necessaria un'analisi di sensibilità variando un po' i valori impostati in `prior_c` e `prior_t`:

- rispetto al valor medio trovato su `depress0` (`mean(df$depress0) = 2.453518`), si vuole provare ad ipotizzare uno scenario in cui i soggetti siano meno depressi o più depressi testando per `nu` i valori `c(log(1.5), log(3.5))`
- per quanto riguarda la varianza, valendo `var(log(df$depress0))=0.01494589`, imponiamo anche in questo caso due scenari opposti testando per `b2` i valori `c(0.001 e 0.1)`
- infine volendo variare un po' anche il grado di confidenza su `nu`, valutiamo i caso in cui `varsigma2` assuma valori in `c(1, 100^2)` rispetto al precedente valore impiegato (pari a 10^2)
- per a invece testiamo i valori in `c(0.1, 10)`.

```
sens_grid = expand.grid(nu = c(log(1.5),log(3.5)), varsigma2 = c(1, 100^2),
                        b2=c(0.001, 1.5), a=c(0.1,10))

sens_res = list()
for (p in 1:nrow(sens_grid))
{
   prior = list(nu=sens_grid[p,'nu'],
                varsigma2 = sens_grid[p,'varsigma2'],
                a = sens_grid[p,'a'],
                b2 = sens_grid[p,'b2'])

   set.seed(42)
   estimands_p = GibbsBasic(niter=Niter, nburn=2000, sampling=1,
                            prior_c=prior, prior_t=prior, Yobs=df[,'depress6'], W=df[,'Z'])

   gibbsResults_p = mcmc(estimands_p)
   resume_p = summary(gibbsResults_p)

   res = c(resume_p$statistics['sate','Mean'],
           resume_p$statistics['pate','Mean'],
           resume_p$statistics['sate','SD'],
           resume_p$statistics['pate','SD'])
   sens_res = rbind(sens_res, res)
}
colnames(sens_res) = c('sate avg', 'pate avg', 'sate stdev', 'pate stdev')
rownames(sens_res) = NULL

sens_res = data.frame(sens_grid, sens_res)

library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.3.3
```

```
kable(sens_res)
```

nu	varsigma2	b2	a	sate.avg	pate.avg	sate.stdev	pate.stdev
0.4054651	1	0.001	0.1	-0.1370263	-0.1374534	0.06199807	0.08182488
1.2527630	1	0.001	0.1	-0.1379386	-0.1383911	0.06207368	0.08193722
0.4054651	10000	0.001	0.1	-0.137368	-0.1378477	0.06204095	0.08189477
1.2527630	10000	0.001	0.1	-0.1373681	-0.1378477	0.06204096	0.08189478
0.4054651	1	1.500	0.1	-0.1376968	-0.1381551	0.06231419	0.08221134
1.2527630	1	1.500	0.1	-0.1386197	-0.1391065	0.06239091	0.08232524
0.4054651	10000	1.500	0.1	-0.1380423	-0.1385545	0.0623577	0.0822822
1.2527630	10000	1.500	0.1	-0.1380424	-0.1385546	0.06235771	0.08228222
0.4054651	1	0.001	10.0	-0.1316863	-0.1321631	0.05969632	0.07941065
1.2527630	1	0.001	10.0	-0.1325196	-0.1329995	0.05976261	0.07950938
0.4054651	10000	0.001	10.0	-0.1319992	-0.132519	0.05973419	0.07947277
1.2527630	10000	0.001	10.0	-0.1319993	-0.1325191	0.05973419	0.07947278
0.4054651	1	1.500	10.0	-0.1947469	-0.1972585	0.08772339	0.1139412
1.2527630	1	1.500	10.0	-0.1966048	-0.1994223	0.08791246	0.1142125
0.4054651	10000	1.500	10.0	-0.1954311	-0.198113	0.08783158	0.1141122
1.2527630	10000	1.500	10.0	-0.1954313	-0.1981132	0.0878316	0.1141123

Si nota che SATE e PATE mostrano media e varianza stabili al variare dei parametri impostati per le prior (sebbene si osservi un "salto" negli estremi per `a=10` e `b2=1.5`).

Sensibilità sulla dipendenza tra i risultati potenziali Pur avendo ipotizzato indipendenza tra $Y(0)$ e $Y(1)$, si esegue un'ultimo test impostando dei valori $\rho \neq 0$ per valutare l'impatto sul SATE.

Le simulazioni mostrano un valore piuttosto stabile sul valor medio del SATE, mentre la sua varianza pare influenzata da $\rho < 0$.

```
rho_grid = c(-0.8,-0.4,0.4,0.8)
prior = list(nu=nu_0, varsigma2 = 10^2, a = 1, b2 = var_0)
rho_res = list()
for (rho in rho_grid)
{
   set.seed(42)
   estimands_rho = GibbsBasic(niter=Niter, nburn=2000, sampling=1, prior_c=prior,
                              prior_t=prior, Yobs=df[,'depress6'], W=df[,'Z'], rho=rho)

   gibbsResults_rho = mcmc(estimands_rho)
   resume_rho = summary(gibbsResults_rho)

   res = c(resume_rho$statistics['sate','Mean'],
           resume_rho$statistics['sate','SD'])
   rho_res = rbind(rho_res, res)
}
colnames(rho_res) = c('sate avg', 'sate stdev')
rownames(rho_res) = NULL
rho_res = data.frame(rho_grid, rho_res)
kable(rho_res)
```

rho_grid	sate.avg	sate.stdev
-0.8	-0.1250593	0.0357622
-0.4	-0.1303883	0.04981249
0.4	-0.1305828	0.06776965
0.8	-0.1262737	0.07393289

Intervalli di credibilità

I valori stimati per SATE e PATE sono piuttosto vicini a 0: pertanto, per capire se siano significativamente diversi da 0, proviamo a determinare gli intervalli di credibilità al 90% per valutare che non contengano tale valore (sui risultati ottenuti per le prior `prior_c = prior_t = list(nu=nu_0, varsigma2 = 10^2, a=1, b2=var_0)`).

Nel caso del PATE siamo vicini all'inclusione dello 0: il risultato ci può stare, trattandosi di una stima soggetta a maggiore variabilità rispetto al SATE (considerando intervalli al 95%, dai quantili già determinati precedentemente con `summary()`, abbiamo che SATE $\in$ (-0.2623, -0.01918) e PATE $\in$ (-0.3005, 0.01945)).

```
(cred_sate = quantile(estimands[,'sate'], probs = c(0.05, 0.95)))
```

```
##          5%         95%
## -0.24034351 -0.03766003
```

```
(cred_pate = quantile(estimands[,'pate'], probs = c(0.05, 0.95)))
```

```
##           5%          95%
## -0.273811258 -0.005593727
```

Anche l'intervallo di "Highest posterior density" ottenuto con `hdi()` fornisce un range che per il PATE comprende per poco anche lo 0.

```
library(HDInterval)
```

```
## Warning: package 'HDInterval' was built under R version 4.3.3
```

```
hdi(estimands)
```

```
##              sate        pate
## lower -0.25942188 -0.29614119
## upper -0.01666874  0.02323502
## attr(,"credMass")
## [1] 0.95
```

Conclusioni

I test effettuati testimoniano la convergenza della distribuzione sia per i valori del SATE che del PATE. Come richiesto dalla teoria, il pate mostra una maggiore variabilità (sd(SATE)=0.06183 < sd(PATE)=0.08161), a fronte della maggiore fonte d'incertezza su cui è costruito.

Il fatto che entrambi mostrino dei valori medi negativi (mean(SATE)=-0.1370, mean(PATE)=-0.1374) indica che il trattamento produce dei benefici, in quanto comporta una diminuzione della depressione. Gli intervalli di credibilità al 90% escludono effetti nulli sia per SATE che per PATE.

L'analisi di sensibilità mostra inoltre dei valori stabili al variare dei parametri impiegati nella prior.