brms

Bayesian regression models using Stan

The **brms** package provides an interface to fit Bayesian generalized (non-)linear multivariate multilevel models using Stan. The formula syntax is very similar to that of the package lme4 to provide a familiar and simple interface for performing regression analyses.

A wide range of distributions and link functions are supported, allowing users to fit – among others – linear, robust linear, count data, survival, response times, ordinal, zero-inflated, hurdle, and even self-defined mixture models all in a multilevel context. Further modeling options include non-linear and smooth terms, auto-correlation structures, censored data, meta-analytic standard errors, and quite a few more. In addition, all parameters of the response distribution can be predicted in order to perform distributional regression. Prior specifications are flexible and explicitly encourage users to apply prior distributions that actually reflect their beliefs. Model fit can easily be assessed and compared with posterior predictive checks and leave-one-out cross-validation.

Bayesian regression models using Stan

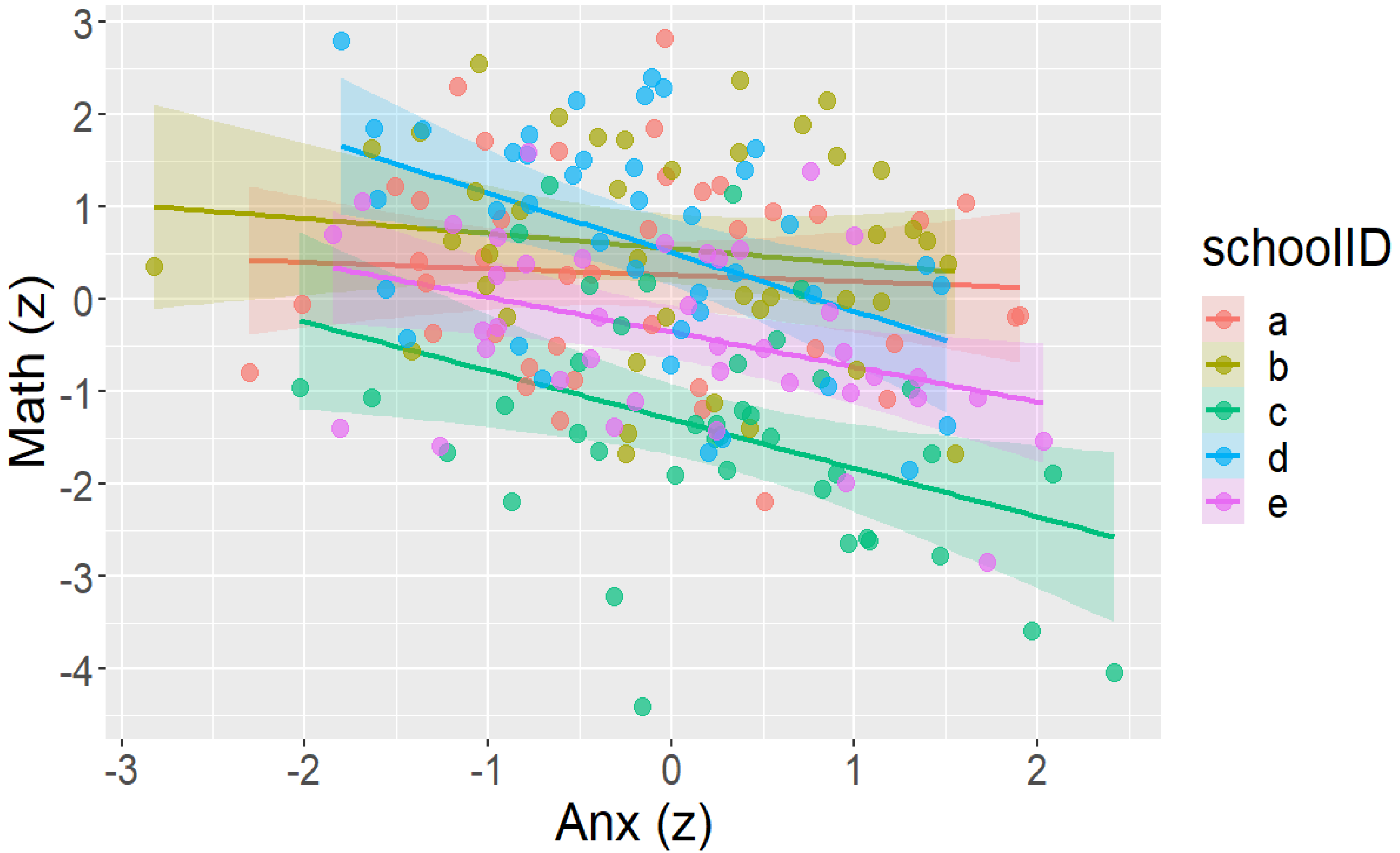
> library (brms)

funzioni base lm e glm
> library (lme4)
> library (metafor)
> library (ordinal) # anche > library (clmm)
> library (regbeta)
> library (lavaan) # almeno in parte
> library (effects)
and more? (es. *diffusion model*)
multivariate + multilevel

> library (blavaan)

> library (lavaan)
letteralmente le stesse cose ma con stima MCMC, mettendo la «b» davanti

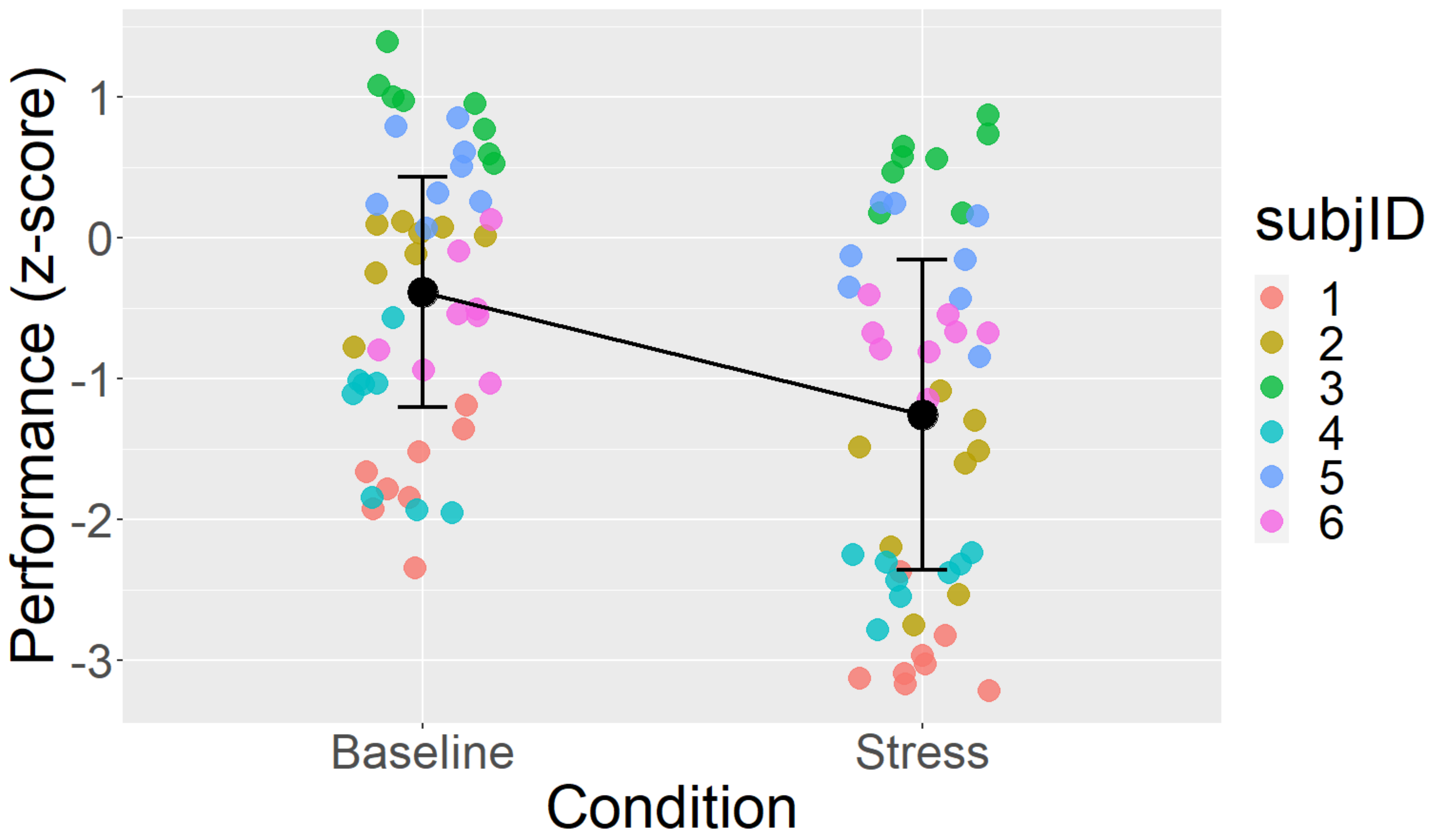
Caso semplice ma paradigmatico: relazione tra 2 variabili, con discreto sample size (N), ma partecipanti raccolti da un certo numero (limitato) di contesti diversi



(analogia: discreto numero di trial ripetuti in condizioni diverse within-participant, ma pochi partecipanti)

(avvertimento per gli scettici: questo è un caso semplice per finalità didattiche: non si vedrà *tutta questa gran differenza* tra i due approcci)

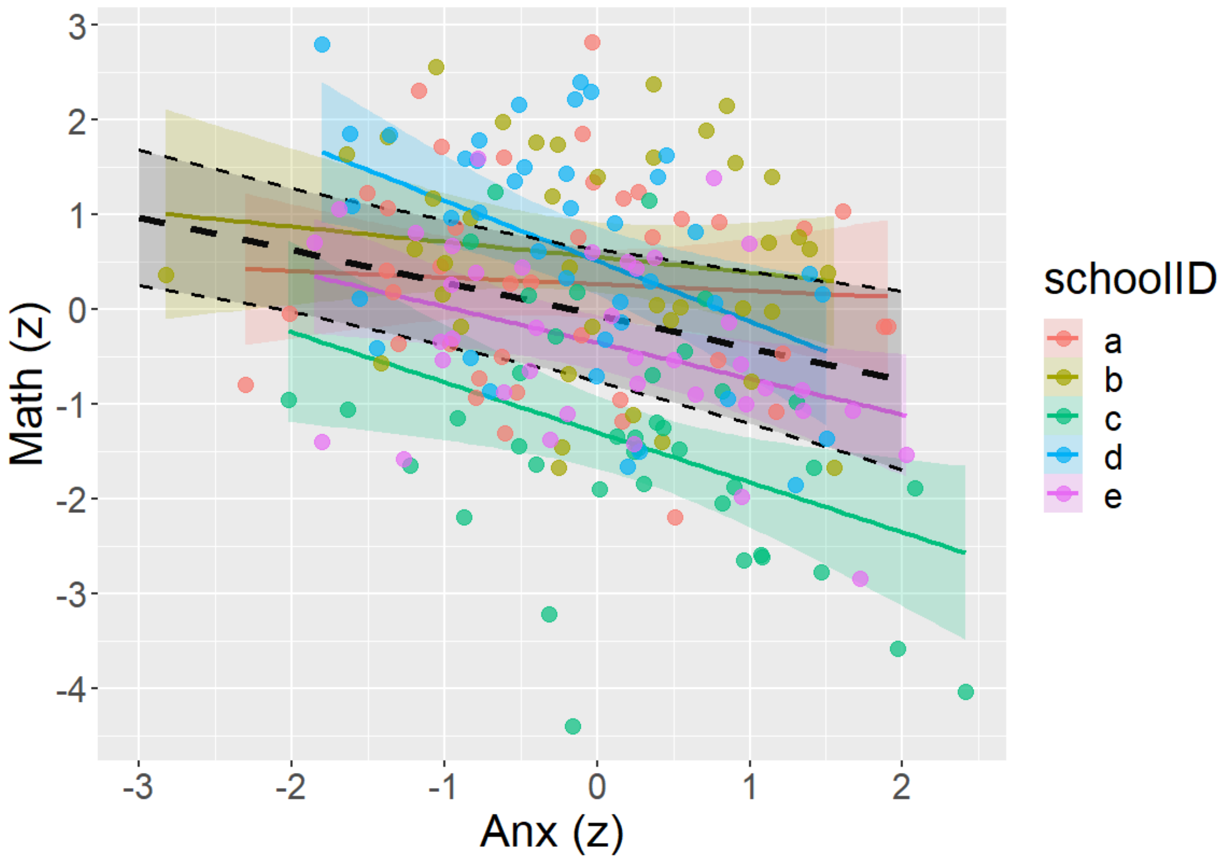
Plausibile esempio alternativo con osservazioni ripetute in trial per soggetto (qui il predittore è categoriale anziché continuo)



Stima «classica» con massima verosimiglianza (pacchetto «lme4»)

```
fit = lmer(math ~ anx + (anx|schoolID), data=d)
```

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 schoolID (Intercept) 0.58902  0.7675
          anx         0.02436  0.1561   0.64
 Residual             1.18516  1.0887
Number of obs: 200, groups:  schoolID, 5

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)
(Intercept) -0.06288    0.35185  3.98574  -0.179   0.8669
anx         -0.34413    0.10565  3.83849  -3.257   0.0331 *
```



Stima via MCMC (pacchetto «brms»)… bayesiana? (prior di default)

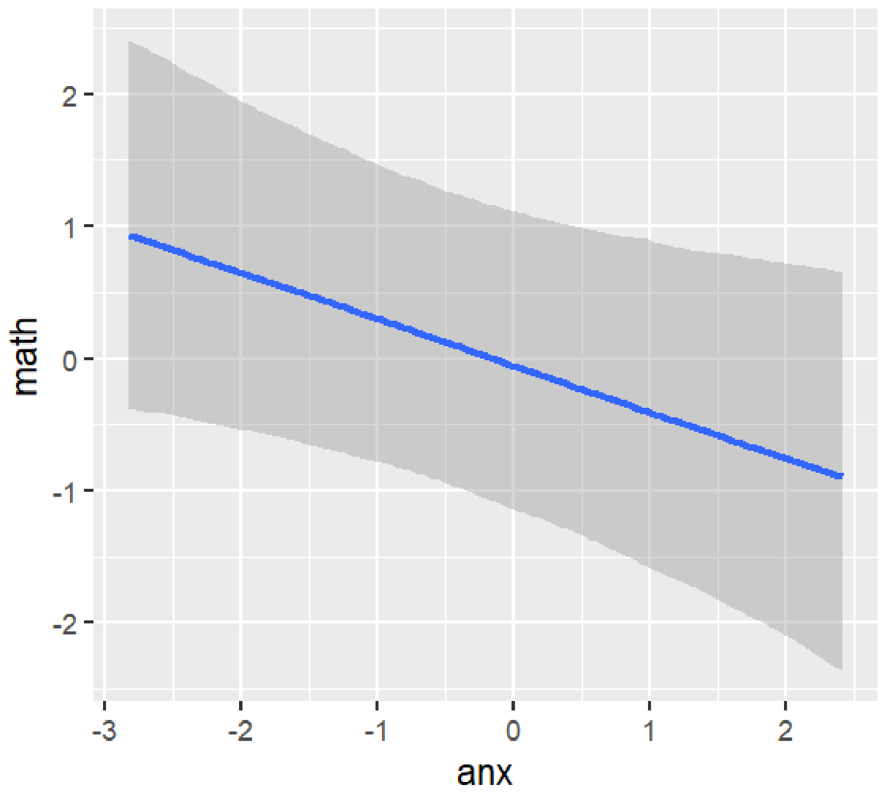
```
fitB = brm(math ~ anx + (anx|schoolID), data=d)
fitB = brm(math ~ anx + (anx|schoolID), data=d, cores=4, iter=4000)
```

```
Group-Level Effects:
~schoolID (Number of levels: 5)
                   Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)          1.16      0.57     0.50     2.70 1.00     1869     3612
sd(anx)                0.27      0.24     0.01     0.88 1.00     2016     2936
cor(Intercept,anx)     0.23      0.50    -0.81     0.96 1.00     5083     4614

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept    -0.04      0.55    -1.14     1.12 1.00     2075     2831
anx          -0.35      0.17    -0.71    -0.02 1.00     3071     2522

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     1.10      0.06     0.99     1.21 1.00     7571     5515

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

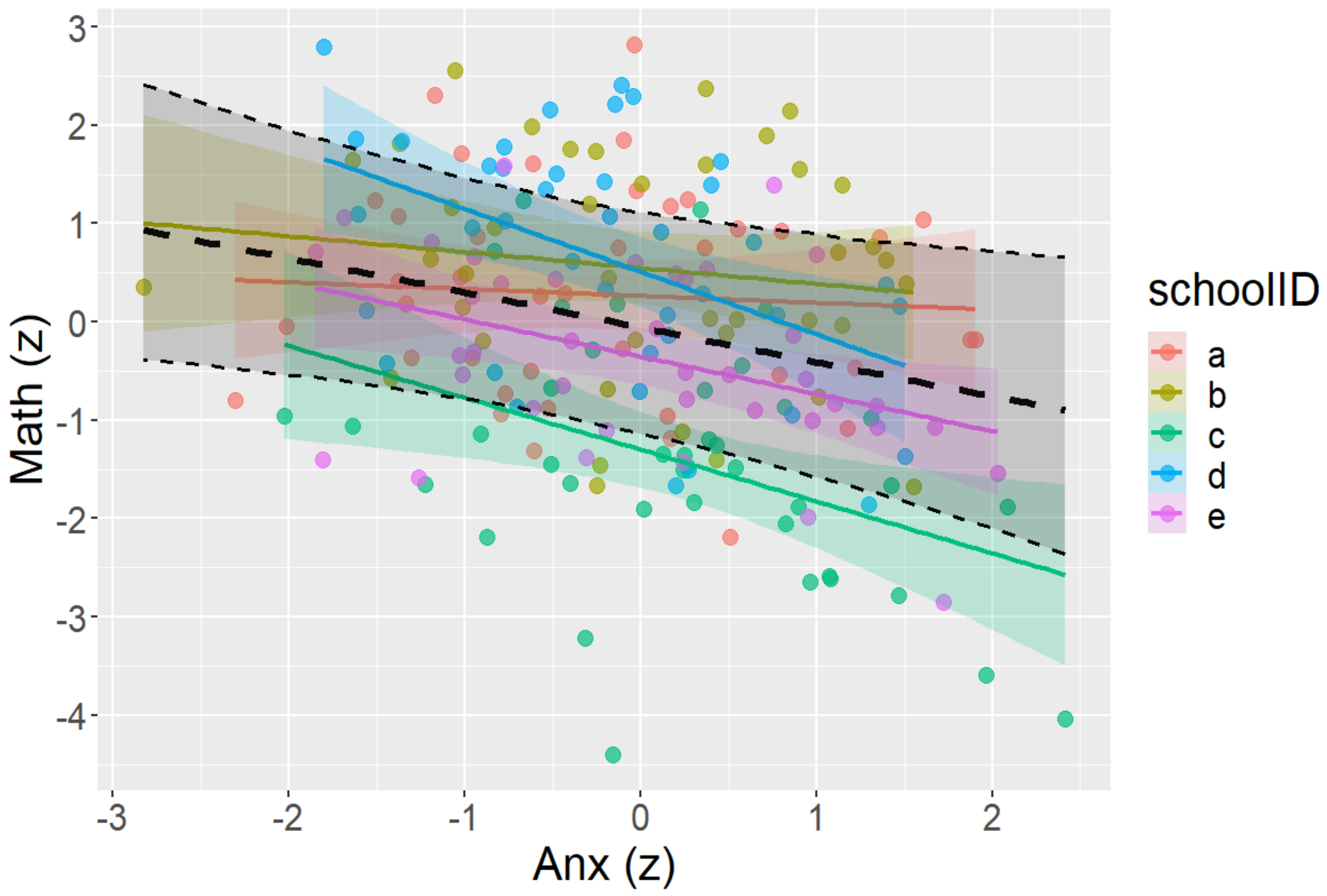
Stima via MCMC (pacchetto «brms»)… bayesiana? (prior di default)

visualizzazione degli effetti fissi

```
conditional_effects(fitB)
```




```
effB = data.frame(conditional_effects(fitB)$"anx")
# [un po’ di ggplot ... vedi codice R allegato]
```



Stima via MCMC (pacchetto «brms»)... bayesiana? (prior di default)

Estrazione «a mano» delle *posterior*

```r
post = data.frame(as_draws_matrix(fitB)) # oppure
post = as.data.frame(fitB)
```

tot: 18 parametri

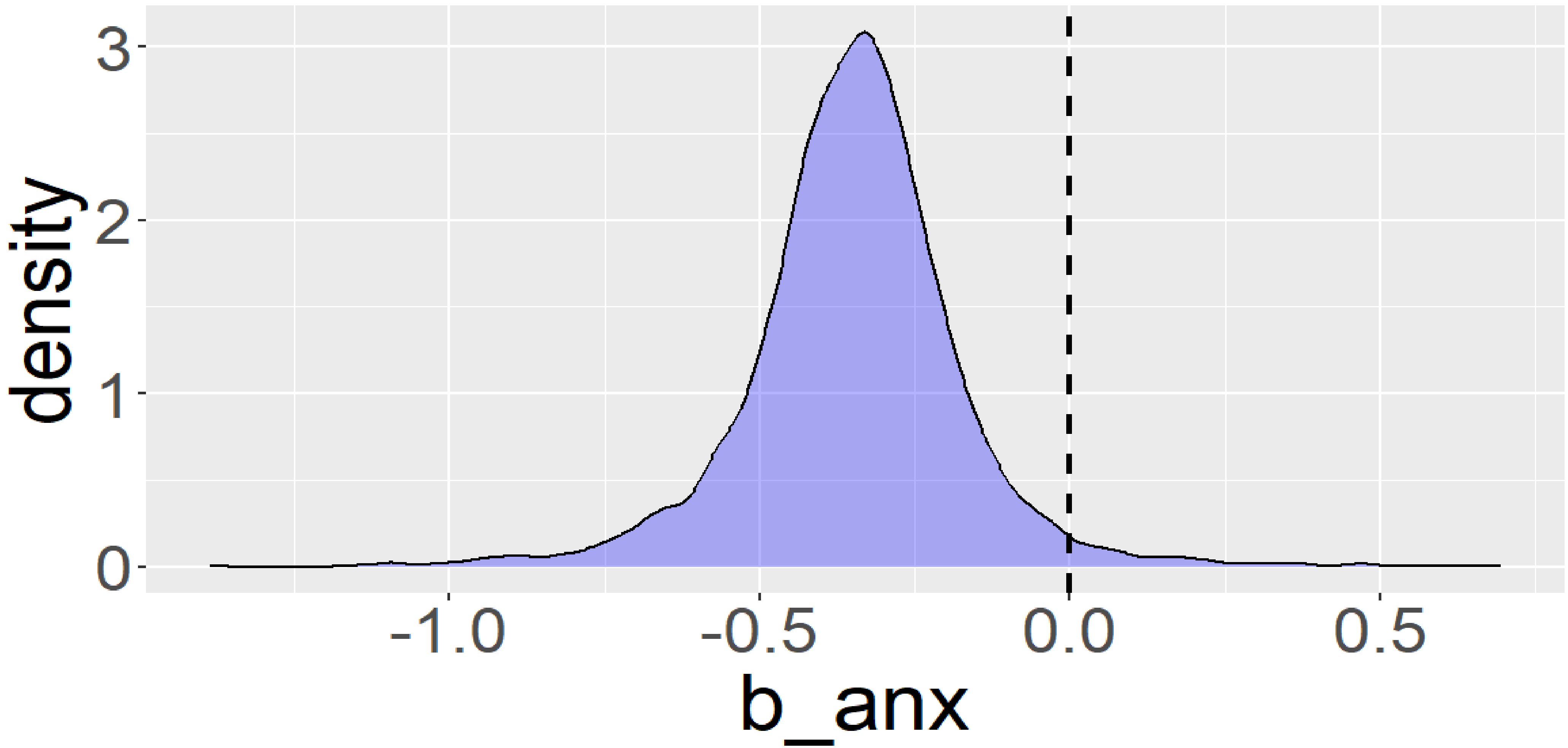
continua →

	b_Intercept	b_anx	sd_schoolID__Intercept	sd_schoolID__anx	cor_schoolID__Intercept__anx	sigma	r_schoolID[a,Intercept]
1	0.09732484	-0.39363610	0.6078522	0.08271371	0.68248862	1.063252	0.08476306
2	-0.06711074	-0.41490880	0.4381143	0.05354611	0.86113364	1.093894	0.05818761
3	-0.08659582	-0.19267467	0.3458250	0.15068947	0.20934918	1.144124	0.07956161
4	-0.50786777	-0.40659850	0.9439644	0.35496420	-0.20689405	1.066194	0.89329825
5	-0.33986422	-0.15269048	1.2948310	0.55051181	0.42998219	1.135692	0.77300713
6	-0.14365849	-0.02700252	1.0148356	0.82342384	0.32376034	1.150210	0.39076785
7	-0.08377013	-0.45759775	0.9473805	0.16929066	0.78457521	1.064966	0.25558200
8	-0.15744824	-0.37158530	0.9682059	0.25223813	0.79962919	1.080229	0.23650203
9	-0.41564670	-0.24482127	0.9010827	0.12483188	-0.37338904	1.036303	0.72314528
10	-0.13511957	-0.22576331	0.9227868	0.10526814	-0.02687139	1.042221	0.59298740
11	0.12080770	-0.30001005	1.1658855	0.23882713	0.94751535	1.076364	0.14622347
12	0.65113946	-0.18779958	0.9177827	0.04706738	0.91389736	1.181848	-0.53200615
13	0.66468590	-0.14912301	1.3375584	0.24421750	0.87085649	1.194443	-0.46234067
14	0.65101753	-0.23498624	0.8582524	0.40546304	0.46068562	1.039018	-0.20811711
15	0.62834762	-0.21298796	0.8376086	0.80959704	0.55261587	1.021166	-0.56011846
16	0.97259094	-0.16788498	1.6069303	0.27376123	0.44869555	1.125528	-0.56715761
17	-0.19557214	-0.29910407	0.9038788	0.27994624	0.75922223	1.016421	0.27966154
18	0.62965398	-0.27309022	1.5553414	0.05919264	-0.41937714	1.096416	-0.60461414
19	0.14936318	-0.38377965	1.3536523	0.04910347	0.08956834	1.069277	-0.10073081
20	0.06218971	-0.21987603	1.1302975	0.21003614	0.16887688	1.145105	0.34142600
21	0.41909011	-0.31171620	0.9106451	0.11524290	0.87696060	1.139371	-0.17406780
22	-0.27537720	-0.22630053	1.2058320	0.02214981	0.02844730	1.066912	0.47098752
23	0.09207754	-0.31170629	0.9400440	0.12735520	0.97694569	1.120761	0.13142526
24	0.09920329	-0.19921896	1.0768478	0.16906406	0.44424690	1.053699	0.13850833
25	-0.10071341	-0.30906657	1.4495380	0.26042531	0.85442073	1.083966	0.56517100
26	-0.05434019	-0.32256431	0.8730047	0.40912292	0.87042610	1.054219	0.25679519
27	0.56528400	-0.32332022	0.9804163	0.01813335	0.60183468	1.083722	-0.64764613
28	-0.34608991	-0.41702326	0.6922362	0.07702852	-0.31113537	1.106795	0.30640399

Stima via MCMC (pacchetto «brms»)… bayesiana? (prior di default)

Visualizzazione «a mano» della *posterior* dell'effetto fisso di interesse

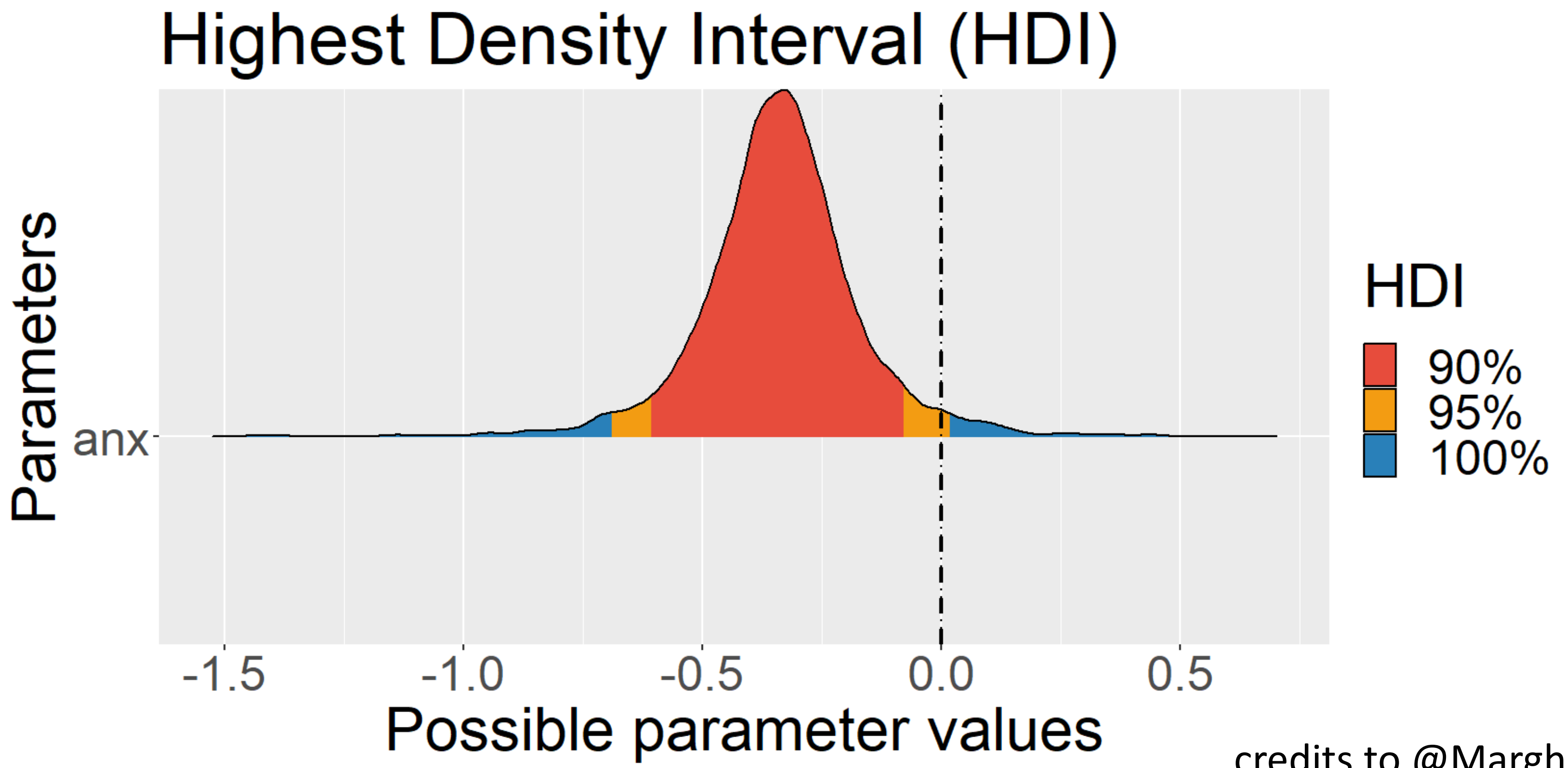
```
ggplot(post) + geom_density(aes(x=b_anx, y=after_stat(density)), fill=«blue», alpha=.3) + ...
```



Stima via MCMC (pacchetto «brms»)... bayesiana? (prior di default)

Visualizzazione della *posterior* dell'effetto fisso di interesse

```
x = bayestestR::hdi(fitB, ci=c(.90,.95))
plot(x)
```



credits to @Margherita Calderan

Stima via MCMC (pacchetto «brms»)… bayesiana? (prior di default)

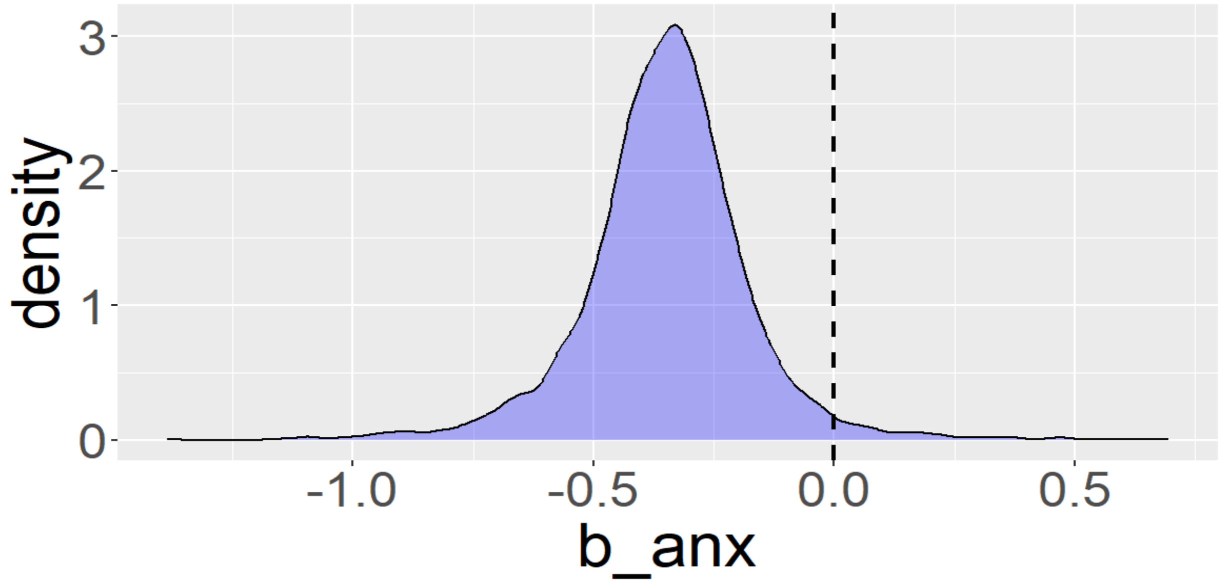
Test di ipotesi sul parametro (per chi non può fare a meno della stellina)

```
hypothesis(fitB,"anx < 0")
```

```
Hypothesis Tests for class b:
  Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio Post.Prob Star
1  (anx) < 0    -0.35      0.17    -0.63    -0.09      43.94      0.98    *
---
'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
```

```
sum(post$b_anx < 0) / sum(post$b_anx >= 0)
```

```
43.94382  # «Evidence Ratio»
```



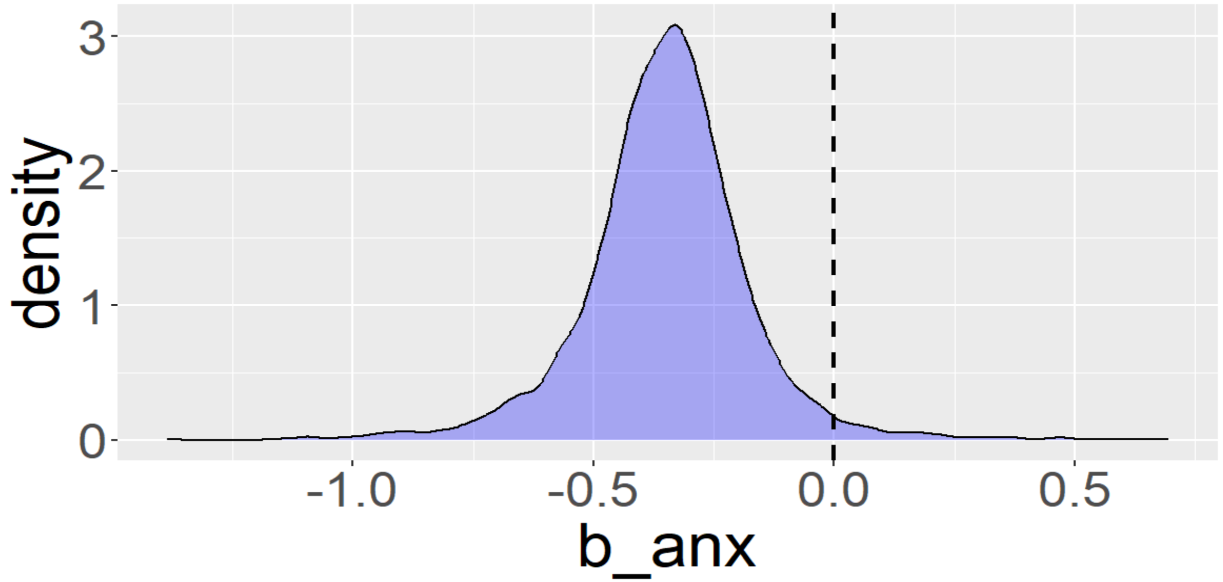
Stima via MCMC (pacchetto «brms»)... bayesiana? (prior di default)

Stima dei parametri di interesse a partire dalle *posterior*

```
mean(post$b_anx)
-0.3491871
```

```
sd(post$b_anx)
 0.1709841
```

```
quantile(post$b_anx, probs=c(.025,.975))
 2.5%          97.5%
-0.71145701   -0.01657338
```



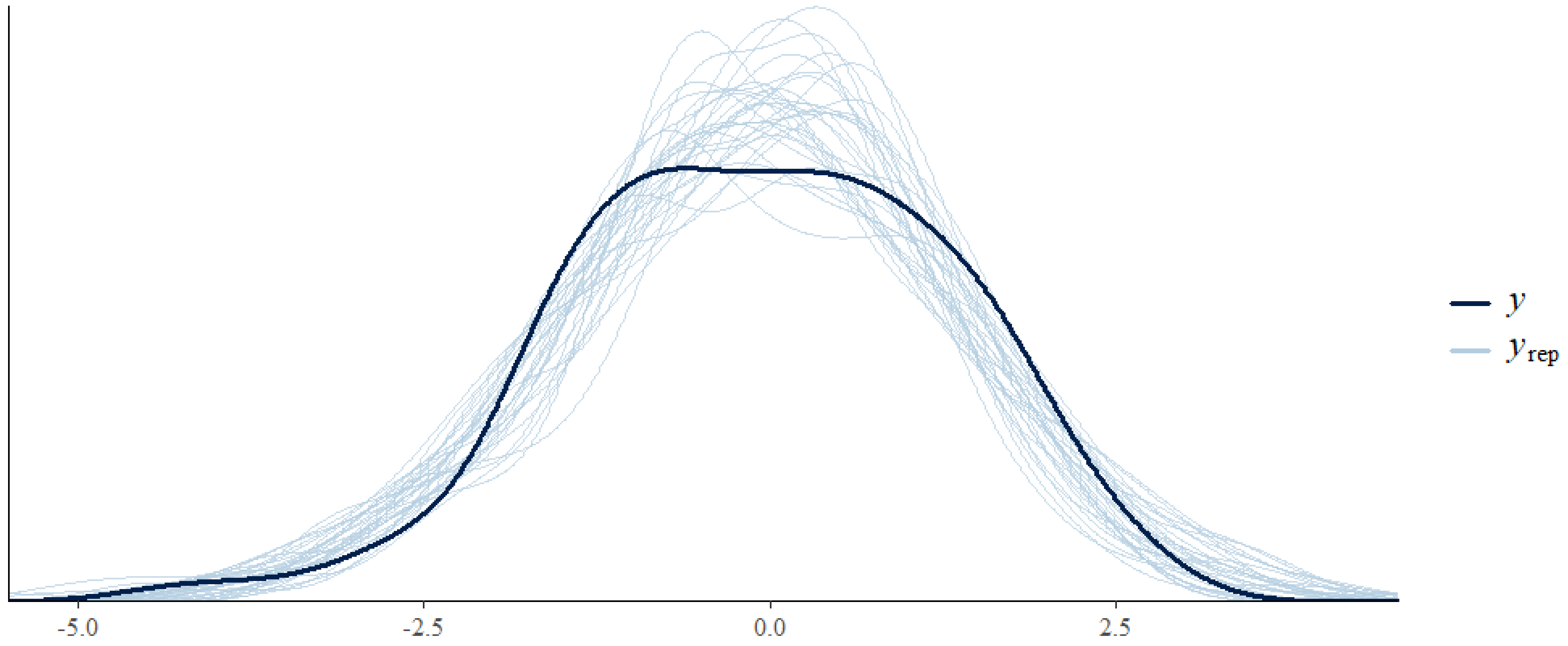
(dal summary del modello...)

```
Population-Level Effects:
         Estimate Est.Error l-95% CI u-95% CI Rhat
anx         -0.35      0.17    -0.71    -0.02 1.00
```

Stima via MCMC (pacchetto «brms»)… bayesiana? (prior di default)

***Posterior predictive check*: quanto bene le *posterior* dei parametri riproducono il set dei dati osservati?**

```
pp_check(fitB, ndraws=30)
```

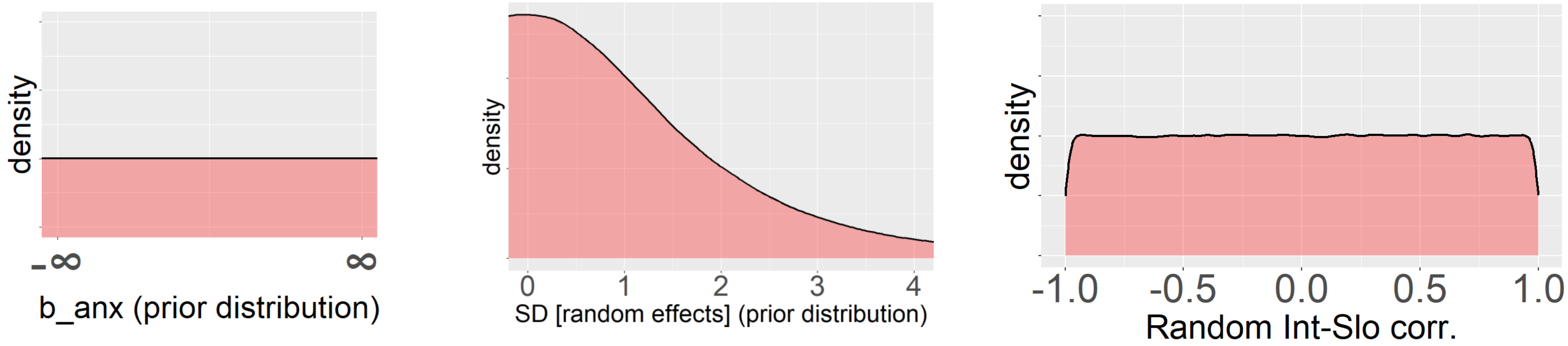


E le prior?!

```
prior_summary(fitB)
```

```
 prior                     class      coef     group resp dpar nlpar lb ub       source
                (flat)         b                                                default
                (flat)         b       anx                                 (vectorized)
 student_t(3, 0, 2.5) Intercept                                                default
 lkj_corr_cholesky(1)          L                                                default
 lkj_corr_cholesky(1)          L           schoolID                        (vectorized)
 student_t(3, 0, 2.5)         sd                                  0             default
 student_t(3, 0, 2.5)         sd           schoolID               0        (vectorized)
 student_t(3, 0, 2.5)         sd       anx schoolID               0        (vectorized)
 student_t(3, 0, 2.5)         sd Intercept schoolID               0        (vectorized)
 student_t(3, 0, 2.5)      sigma                                  0             default
```

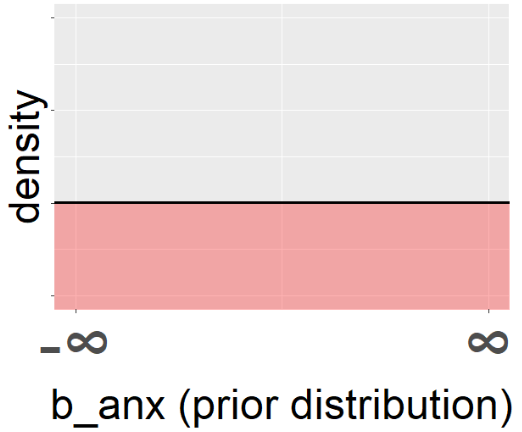
Quelle di default sono davvero «credibili»?!



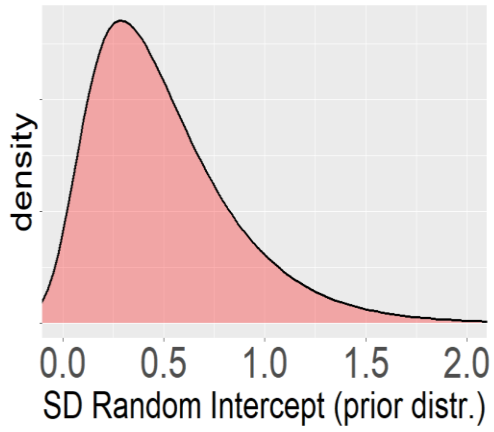
Rivediamo alcune prior, settandole su valori plausibili

```
pr1 = set_prior("gamma(2,4)", class="sd", coef="Intercept", group="schoolID")
pr2 = set_prior("gamma(1,2)", class="sd", coef="anx", group="schoolID")
pr3 = set_prior("lkj_corr_cholesky(4)", class="L")
fitB1 = brm(math ~ anx + (anx|schoolID), data=d, prior=c(pr0,pr1,pr2,pr3), iter=4000)
```

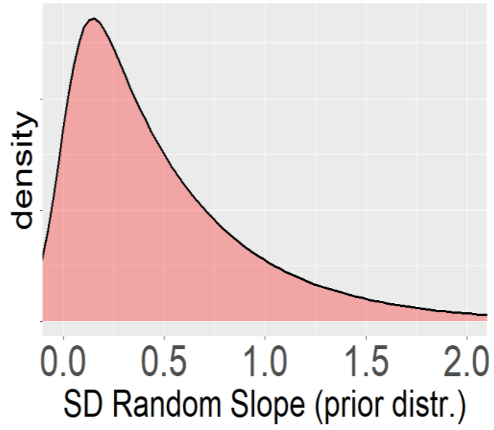
Ok, anche se non ha senso, NON tocco la prior di default dell'effetto fisso di interesse. In alternativa potrebbe avere senso una Normal(0,1)



Ritengo molto improbabili valori > 1; valori attorno a 0.5 ancora molto probabili



Ritengo molto improbabili valori > 1; valori attorno a 0.5 sono già meno probabili



Escludo valori troppo vicini a -1 e +1; plausibilmente il parametro è tra -0.5 e +0.5

Summary attuale con prior informative plausibili

```
 Group-Level Effects:
~schoolID (Number of levels: 5)
                   Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)          0.76      0.24     0.42     1.34 1.00     3561     4802
sd(anx)                0.20      0.16     0.01     0.59 1.00     2434     3553
cor(Intercept,anx)     0.09      0.31    -0.51     0.66 1.00     6961     5499

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept    -0.06      0.37    -0.78     0.71 1.01     2383     3077
anx          -0.35      0.13    -0.65    -0.10 1.00     4529     3514

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     1.10      0.06     0.99     1.22 1.00     8110     5816
```

```
hypothesis(fitB,"anx < 0")
```

```
Hypothesis Tests for class b:

  Hypothesis Estimate Est.Error
  (anx) < 0     -0.35      0.13

CI.Lower CI.Upper Evid.Ratio Post.Prob Star
  -0.58    -0.15     130.15      0.99    *
```

Summary precedente con prior di default

```
 Group-Level Effects:
~schoolID (Number of levels: 5)
                   Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)          1.16      0.57     0.50     2.70 1.00     1869     3612
sd(anx)                0.27      0.24     0.01     0.88 1.00     2016     2936
cor(Intercept,anx)     0.23      0.50    -0.81     0.96 1.00     5083     4614

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept    -0.04      0.55    -1.14     1.12 1.00     2075     2831
anx          -0.35      0.17    -0.71    -0.02 1.00     3071     2522

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     1.10      0.06     0.99     1.21 1.00     7571     5515
```

```
hypothesis(fitB,"anx < 0")
```

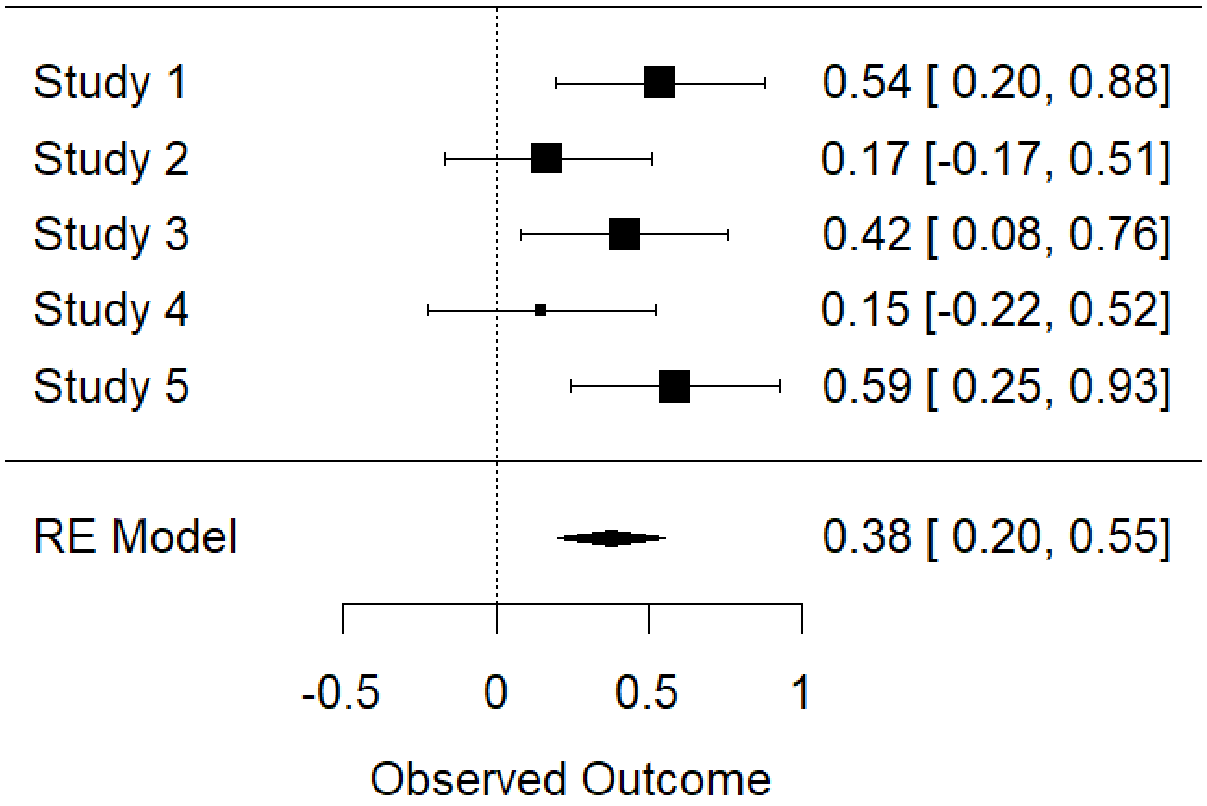
```
Hypothesis Tests for class b:

  Hypothesis Estimate Est.Error
  (anx) < 0     -0.35      0.17

CI.Lower CI.Upper Evid.Ratio Post.Prob Star
  -0.63    -0.09      43.94      0.98    *
```

Esempio di meta-analisi, random-effects model con «metafor»

Un problema frequente nelle meta-analisi in psicologia è lo scarso numero di studi, il che rende difficile la stima dell'eterogeneità, che pure è ritenuta una «certezza» nella nostra letteratura. D'altra parte rassegnarsi a stimare modelli con *effetti fissi* porterebbe a una grossolana sovrastima della precisione dell'effetto meta-analitico (95% CI troppo stretti; ogni studio diventa un caso influente)

```
fitMA = rma(yi=eff,vi=vi,data=dm)
summary(fitMA)
forest(fitMA)
```

```
tau^2 (estimated amount of total heterogeneity): 0.0093
tau (square root of estimated tau^2 value):      0.0963
I^2 (total heterogeneity / total variability):   23.02%
H^2 (total variability / sampling variability):  1.30

Test for Heterogeneity:
Q(df = 4) = 5.2363, p-val = 0.2639

Model Results:

estimate      se    zval    pval   ci.lb   ci.ub
  0.3784  0.0898  4.2146  <.0001  0.2024  0.5543  ***
```



in questo caso stimiamo comunque con effetti random, anche se l'eterogeneità non risulta significativa, e il tau viene sottostimato

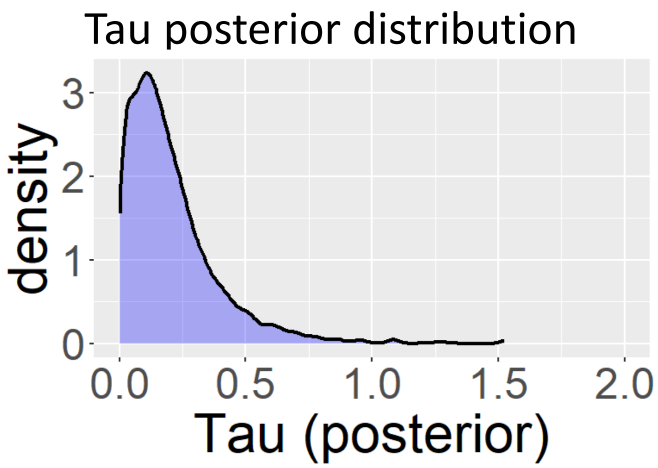
Esempio di meta-analisi con «brms»

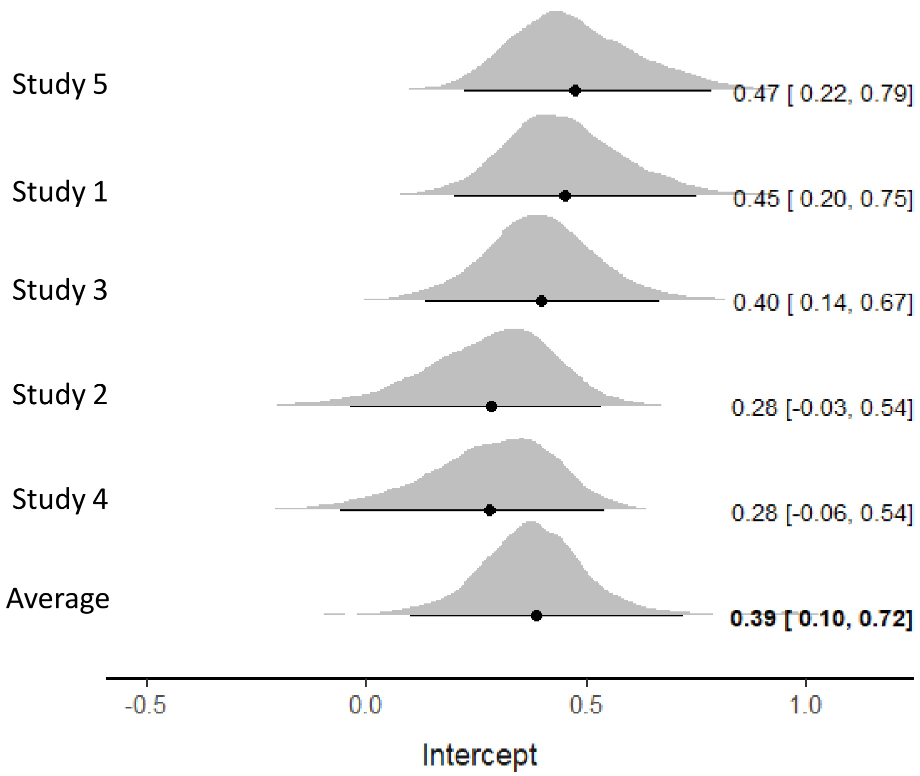
study	eff	vi	sei
a	0.537	0.030	0.173
b	0.171	0.030	0.173
c	0.420	0.030	0.173
d	0.150	0.036	0.189
e	0.586	0.030	0.173

Lascio prior di default (comunque NON ottimale)

```
fitMA_B = brm( eff | se(sei) ~ 1 + (1|study), data=dm, iter=5000)
```

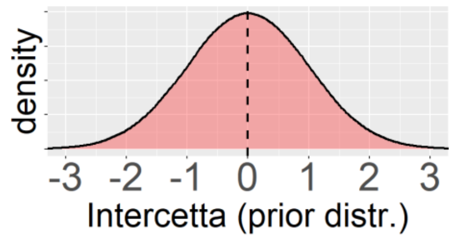
```
Group-Level Effects:
~study (Number of levels: 5)
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)     0.21      0.19     0.01     0.69 1.00     1177      894

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.39      0.15     0.10     0.72 1.00     1082      534
```

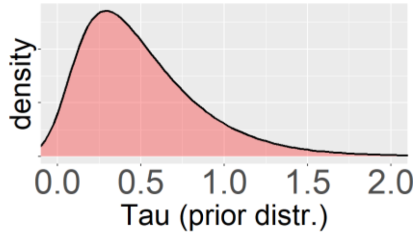




Esempio di meta-analisi con «brms»

Metto prior un po' ragionate






```
pr1 = set_prior("normal(0, 1)",class="Intercept",group="")
pr2 = set_prior("gamma(2, 4)",class="sd",group="study")
fitMA_B_info = brm( eff | se(sei) ~ 1 + (1|study), data=dm, prior=c(pr1,pr2), iter=5000)
```

Di fatto, in questo caso semplice, non ho comunque «guadagnato» praticamente niente

```
Group-Level Effects:
~study (Number of levels: 5)
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)     0.23      0.15     0.03     0.58 1.00     2443     2133

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.37      0.14     0.11     0.65 1.00     2540     2481
```