Run 2

Contents


```
require(MASS)
require(rstan)
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
```

Set up simulation

Difference from Run1 is that this perturbs the orientation of the small random faults. And uses new `dmin` arg to `addRandomFaults()` to eliminate random fault elements too close to other elements (Run1 could just get rid of elements whose center x-coords were too close to zero since all elements were parallel to xz-plane).

Build fault model

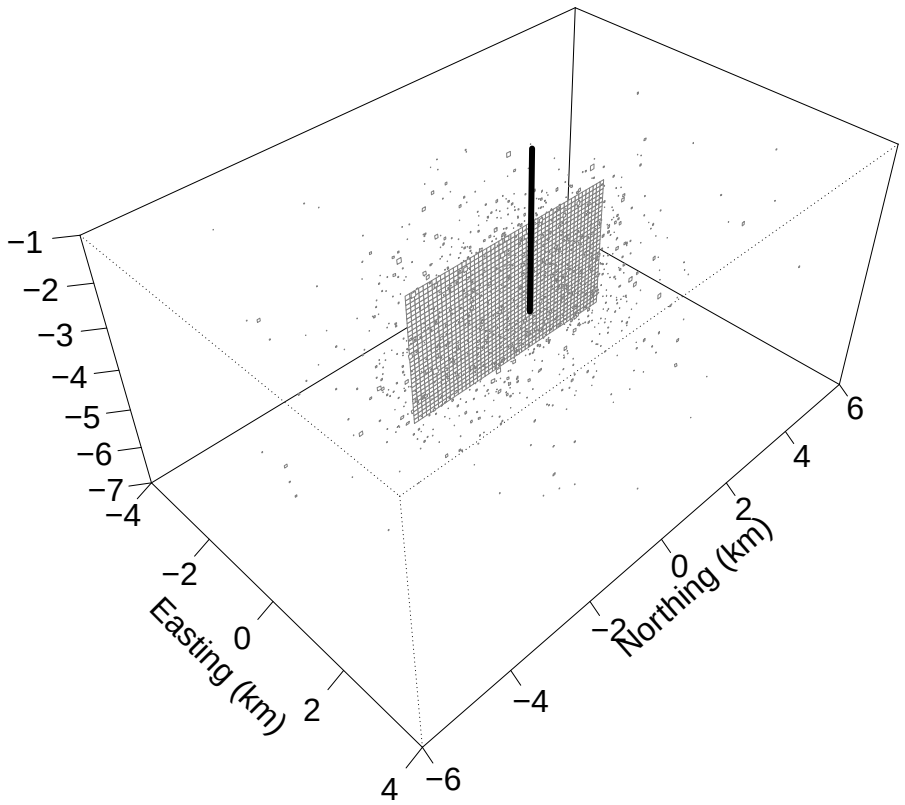
Make a simple fault model consisting of a single, vertical, planar strikeslip fault, and then add random small faults around it.

```
L = 5e3 # fault length
W = 3e3 # fault vertical extent
y0 = -L/2 # center fault on origin
zmax = -2e3 # top edge of fault
l = w = 100 # element length and width
strike = 0
ddot = 0
system2(file.path(RSQSimHome, "BuildFaultModels/singleStrikeSlip"),
        paste(0, y0, zmax - W, zmax, L, round(L/l), round(W/w), strike, ddot=0),
         stdout="test.flt")
```

```
flt = readFault("test.flt", findlminmax=TRUE)

nrand = 2000 # number of random faults to add
b = 1        # desired GR b-value
MrandMin = 0 # min magnitude on random faults
MrandMax = 2 # max magnitude on random faults
d0 = 1000    # mean of exponentially distributed distance of random faults
             # to main fault element
set.seed(123) # for reproducibility

addRandomFaults(flt, nrand, b, MrandMin, MrandMax, rexp, outFname="test.addRandom.10.flt",
                ddot=0, sdrot=10, rate=1/d0, dmin=10)
flt2 = readFault("test.addRandom.10.flt")
```

Plot of resulting fault system

```
# injection point that we will use below
xi = matrix(c(500, 0, -3.5e3),1, 3) # well location: 500 m to east of center pt. of main fault

p = plotFault3d(flt2,  theta=45, phi=40, xlim=c(-4,4), ylim=c(-6,6), zlim=c(-7,-1), lwd=0.5)
lines(trans3d(c(xi[1,1], xi[1,1])/1e3, c(xi[1,2], xi[1,2])/1e3, c(0, xi[1,3])/1e3, p), lwd=3)
```



Set up injection history and resulting stressing file

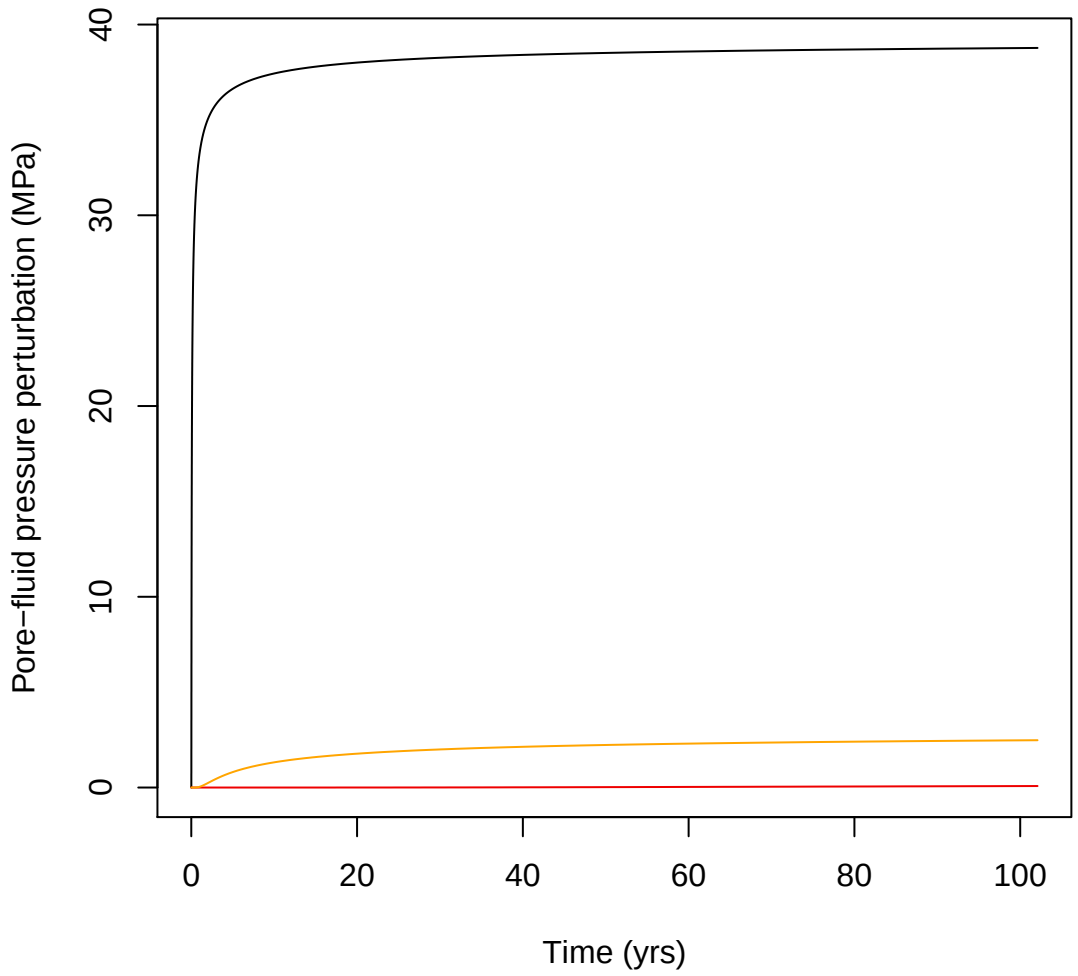
Make a simple injection history and RSQSim external stressing file from it.

```
yr = 365.25*86400

injHist = makeInjectionHistory(xi=xi) # use defaults for all hydrologic params

t = findSampleTimes(flt2, injHist, tmax=100*yr, eps=0.05, minh=1000, maxErr=100)
writePoreFluidStressingHistoryFile(flt2, injHist, t, "test.externalStress.txt")

s = readExternalStressingHistoryFile("test.externalStress.txt", flt2)
```

Plots of pressure histories

The closest element, the farthest element, and the median distance element.

```
dwell = elementWellDist(flt2, injHist)
ord = order(dwell)
plot(s$t/yr, -s$sigma[ord[1],], typ='l', xlab="Time (yrs)",
     ylab="Pore-fluid pressure perturbation (MPa)")
lines(s$t/yr, -s$sigma[ord[flt2$np],], col="red2")
lines(s$t/yr, -s$sigma[ord[round(flt2$np/2)],], col="orange")
```



Run simulation

I made `test.addRandom.10.in` by hand.

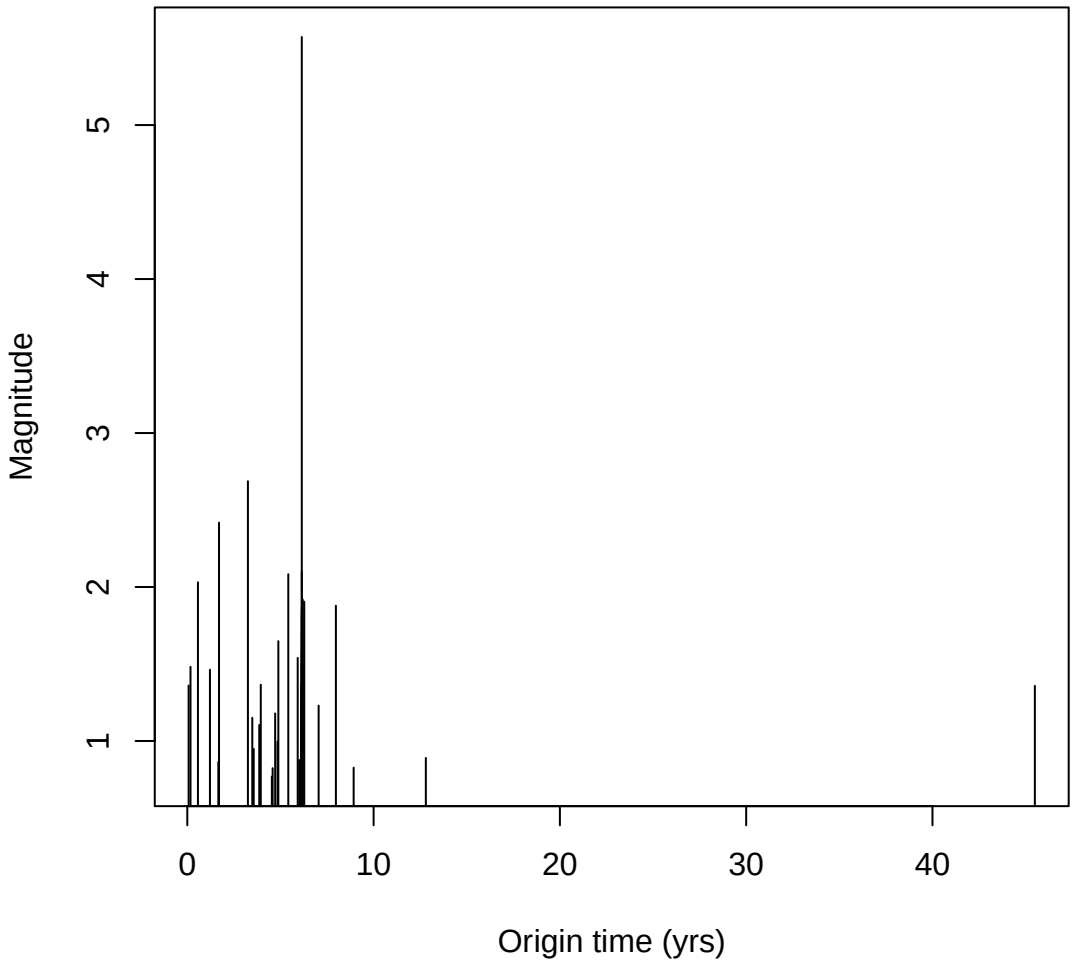
```
unlink("eqs.test.addRandom.10.out")
system2("mpirun", args="-np 4 ~/RSQSim.github.master/runRSQSim test.addRandom.10.in",
        stdout="test.addRandom.10.std", stderr="test.addRandom.10.std")
```

Results

```
params = list(outFnameInfix="test.addRandom.10")
eqs = readEqs("eqs.test.addRandom.10.out")
if (max(eqs$t0) > max(s$t)) {
  eqMax = min(which(eqs$t0 > max(s$t))) - 1
  eqs = readEqs(paste("eqs.", params$outFnameInfix, ".out", sep=""), eqMax = eqMax)
}
```

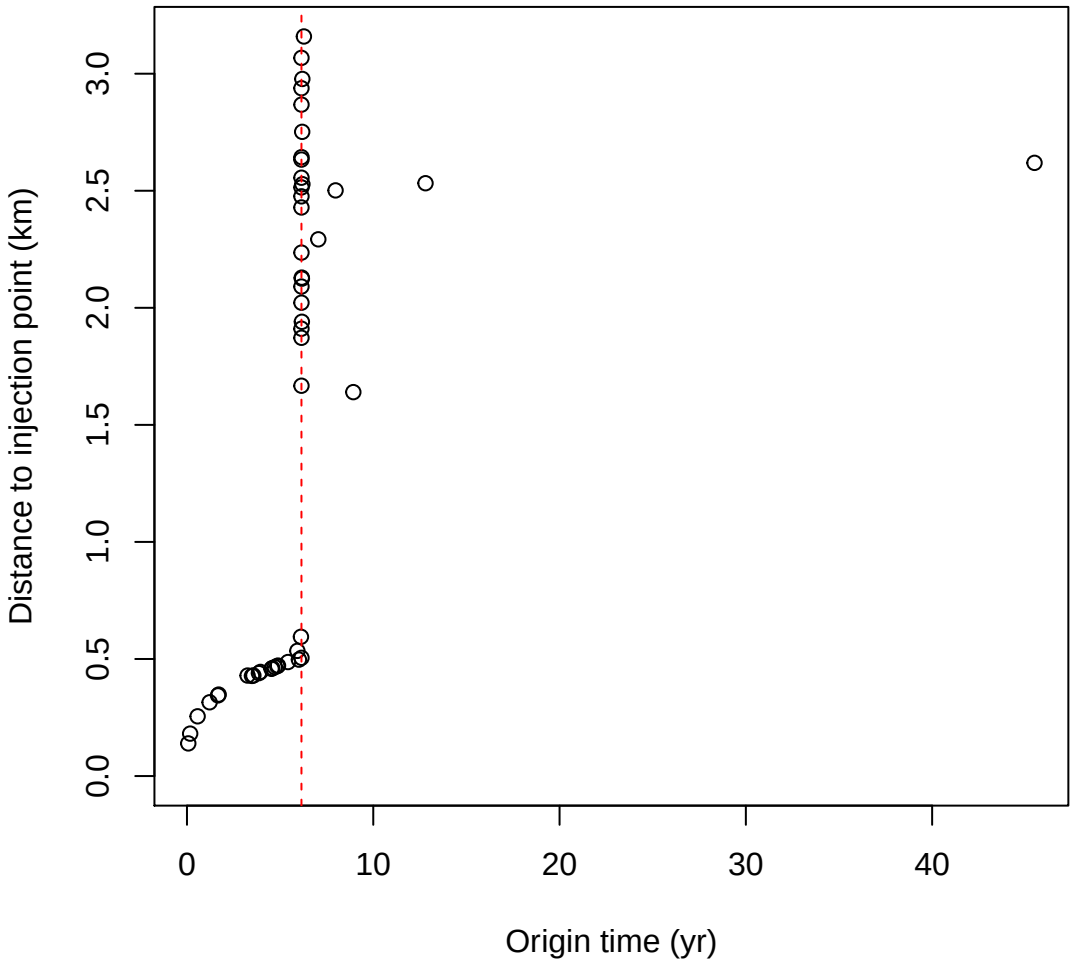
Magnitude-time plot

```
plot(eqs$t0yr, eqs$M, typ='h', xlab="Origin time (yrs)", ylab="Magnitude")
```



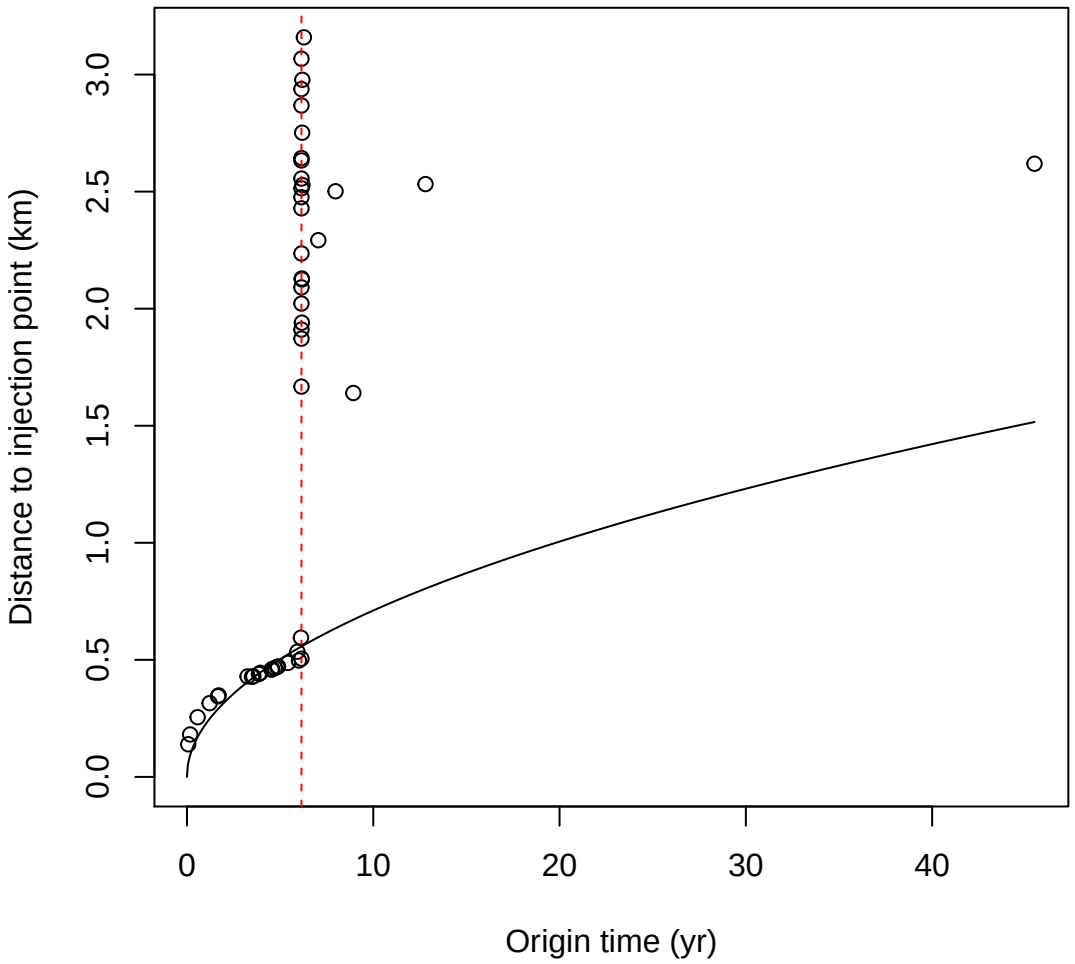
Hypocenter-well distance

```
d = hypoWellDist(eqs, injHist)
plot(eqs$t0yr, d/1e3, ylim=c(0,max(d/1e3)),
     xlab="Origin time (yr)", ylab="Distance to injection point (km)")
abline(v=eqs$t0yr[eqs$M>5], col="red", lty="dashed")
```



The evolution of the distance of the induced events from the injection point with time looks roughly like the expected $\sqrt{t}$ (until the M5+). It is not clear what the exact relationship between the coefficient of $\sqrt{t}$ and the hydraulic parameters should be. *Shapiro et al.* [1997] advocate for $x = \sqrt{4\pi\kappa t}$, where κ is the diffusivity. The argument of the erfc in our pore-fluid pressure expression is equal to one at $x = \sqrt{4\kappa t}$, but there is also a $|x|^{-1}$ factor in front of the erfc. Adding in the least-squares fit to the pre-M5 events:

```
plot(eqs$t0yr, d/1e3, ylim=c(0,max(d/1e3)),
     xlab="Origin time (yr)", ylab="Distance to injection point (km)")
abline(v=eqs$t0yr[eqs$M>5], col="red", lty="dashed")

use = 1:(which(eqs$M>5)-1)
k = sum(d[use]*sqrt(eqs$t0[use])) / sum(eqs$t0[use]) # in m/s^{1/2}
kplot = k * (1/1e3) * sqrt(86400*365.25) # in km/yr^{1/2}
tplot = seq(0, max(eqs$t0yr), length=1000)
lines(tplot, kplot*sqrt(tplot))
```



The $\sqrt{t}$ is not a great fit: event times are systematically earlier at near distances and later at farther distances than the best fit $\sqrt{t}$ curve. I would attribute this to the $|x|^{-1}$ factor in the pore-fluid pressure expression. It could also be due to RSF evolution effects. If we ignore that and ask what value of κ is implied by the *Shapiro et al.* [1997] expression or just setting the argument of the erfc to 1, we get 1.3×10^{-4} and 4×10^{-4} m^2/s, respectively. These are both a bit over an order of magnitude lower than the actual diffusivity used in generating the pore-fluid pressures (0.008 m^2/s). Note that this difference here between the actual diffusivity and that estimated by assuming $x = \sqrt{4\pi\kappa t}$ is not necessarily relevant to the data in *Shapiro et al.* [1997]. In that paper the diffusivity is much higher and the time scales are much shorter (a few days), so that RSF evolution effects may be less important.

Elizabeth mentions that some papers of Talwani estimates the diffusivity by fitting curves of the form $x = \sqrt{\kappa t}$. Also said that might have been for fluid pressures diffusing in 2-d (*e.g.* if a fault is a high permeability conduit) instead of in the 3-d bulk. If we ignore the 2-d/3-d issue and just estimate κ from curves of that form, we get 0.0016 m^2/s, about 1/5 the actual value.

Pore-fluid pressure at failure

As an indication of how important RSF evolution effects may be, let's look at the pore-fluid pressure one each hypocentral element just before it failed. For friction with a fixed failure stress (*e.g.* slip-weakening) and uniform initial stresses as we have here, this would be expected to be the same for all events as long as the stress transfer between the elements is neglible (which it will be in this fault system everywhere except the main fault). However, with RSF, the more distant, later-failing elements will fail at a lower pore-fluid pressure because (once they are above steady-state) they will have been weakening.

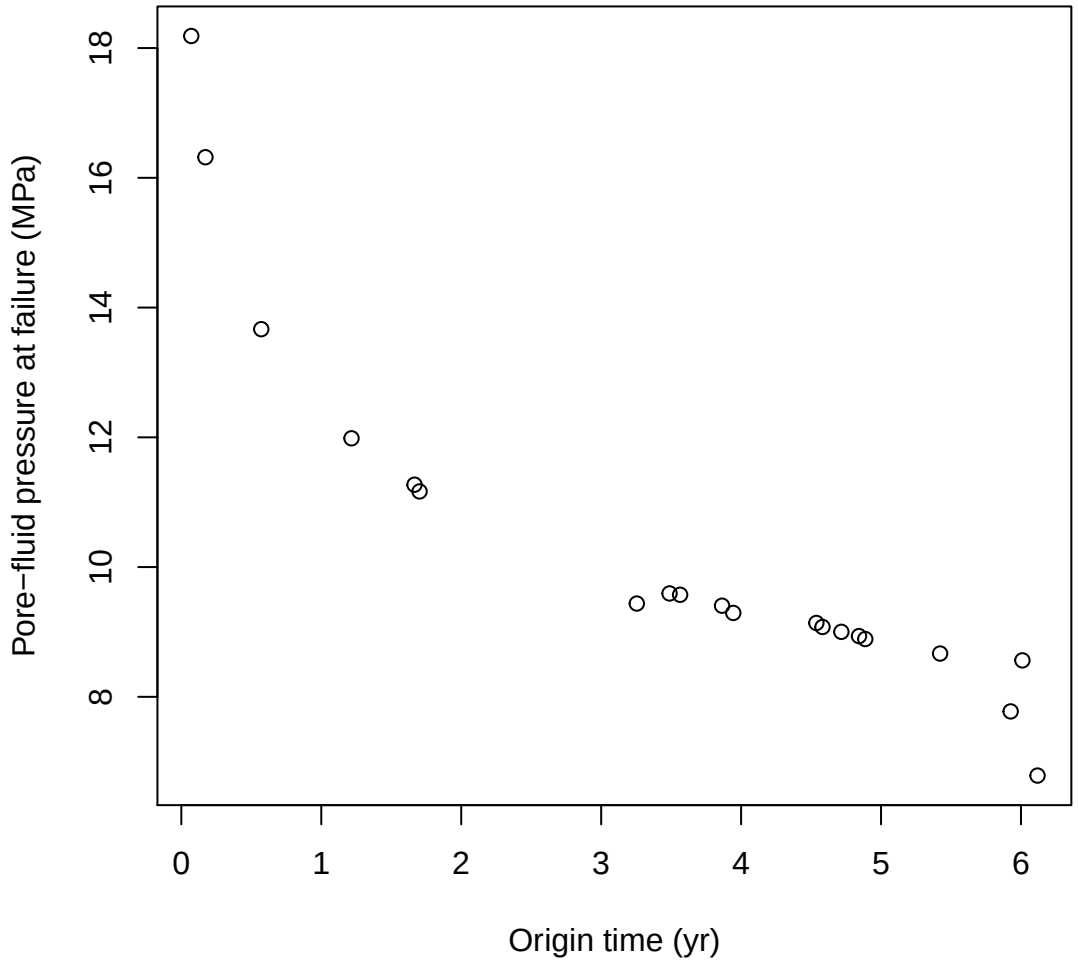
Note that the instantaneous pore-fluid pressure increment that would cause immediate failure is given by that P which solves

$$V^* \exp\left(\frac{\tau - \mu_0(\sigma_0 - P)}{a(\sigma_0 - P)}\right)\left(\theta_0\left(\frac{\sigma_0 - P}{\sigma_0}\right)^{\alpha/b}\frac{V^*}{D_c}\right)^{-b/a} = V^{\mathrm{EQ}}.$$

For this simulation, this comes out to about 26.9 MPa.

Below I plot the pore-fluid pressure perturbation at failure for all events before the M5 (after which stress transfer dominates).

```
itfail = findInterval(eqs$t0, s$t)
pFail = s$sigma[cbind(eqs$patch, itfail)] +
        (s$sigma[cbind(eqs$patch, itfail+1)] - s$sigma[cbind(eqs$patch, itfail)]) *
        (eqs$t0 - s$t[itfail])/(s$t[itfail+1] - s$t[itfail])
pFail = -pFail
plot(eqs$t0yr[use], pFail[use], xlab="Origin time (yr)",
     ylab="Pore-fluid pressure at failure (MPa)")
```



These are indeed all less than 26.9 MPa and are smaller at later times / farther distances.

So, for now, it is an open question how much of the departure from $\sqrt{t}$ behavior is due to the $|x|^{-1}$ and how much due to RSF evolution effects.

Estimation of Omori p-value and aftershock fraction n

See `templateFamilies.[Rmd|pdf]`

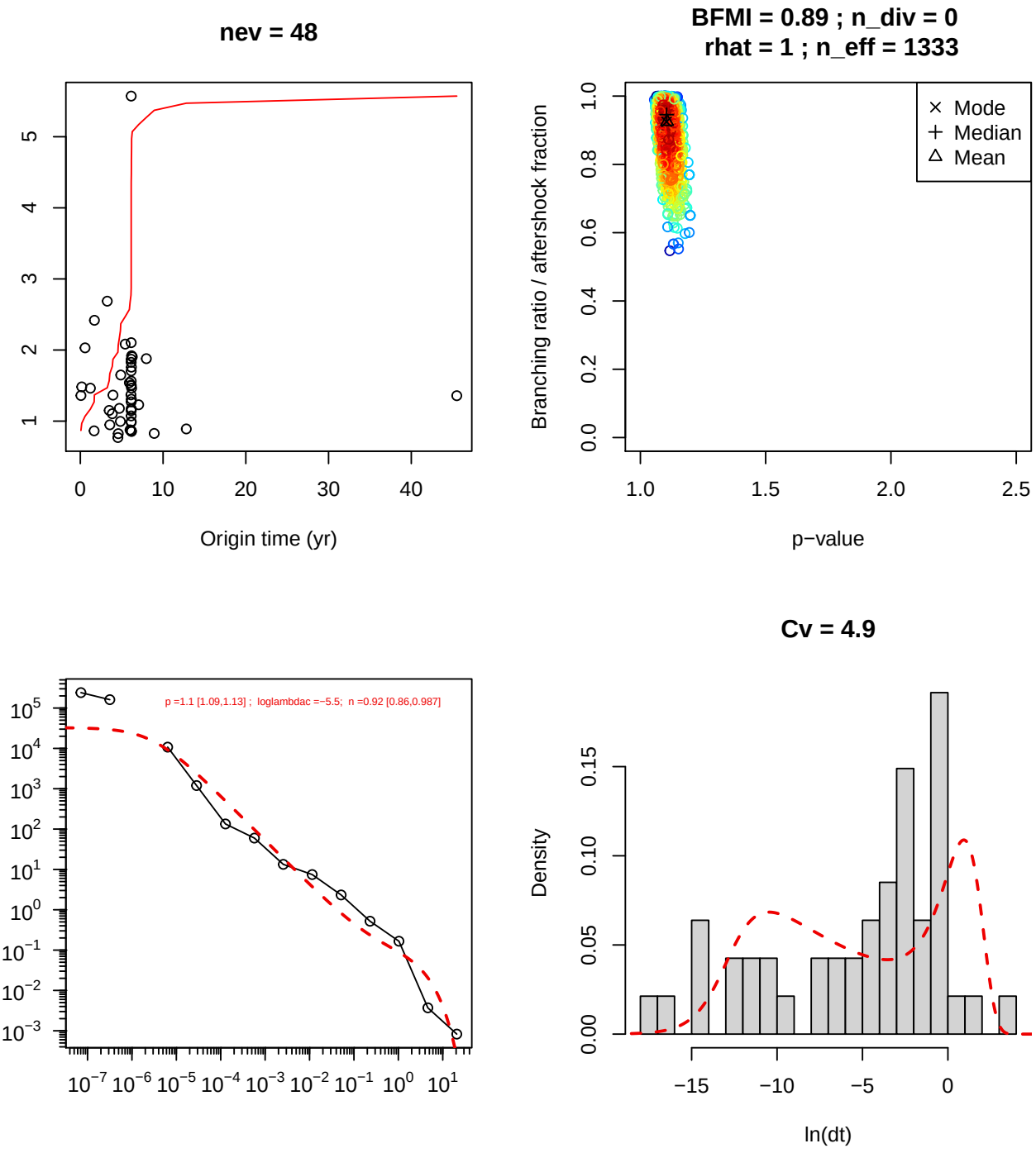
```
use = 1:length(eqs$t0)
alpha = 0.32 # so 68% CIs
c = 90 # seconds, needs to match dSS2007.krd.fixedloglambdac.stan
st = do.stan(eqs, use=use, alpha=alpha, c=c)
```

Plotting function definition

The functions to make plots like in `templateFamilies.pdf`; modified from there. Not printing them out because they take a lot of page space.

Actually make the plots

```
mkFourPanelPlot(eqs, use, st$bfmi, st$n_div, st$rhat, st$n_eff,
                st$f.stan,
                st$p.stan.mode, st$pCI.stan, st$loglambdac.stan.mode,
                st$n.stan.mode, st$nCI.stan)
```



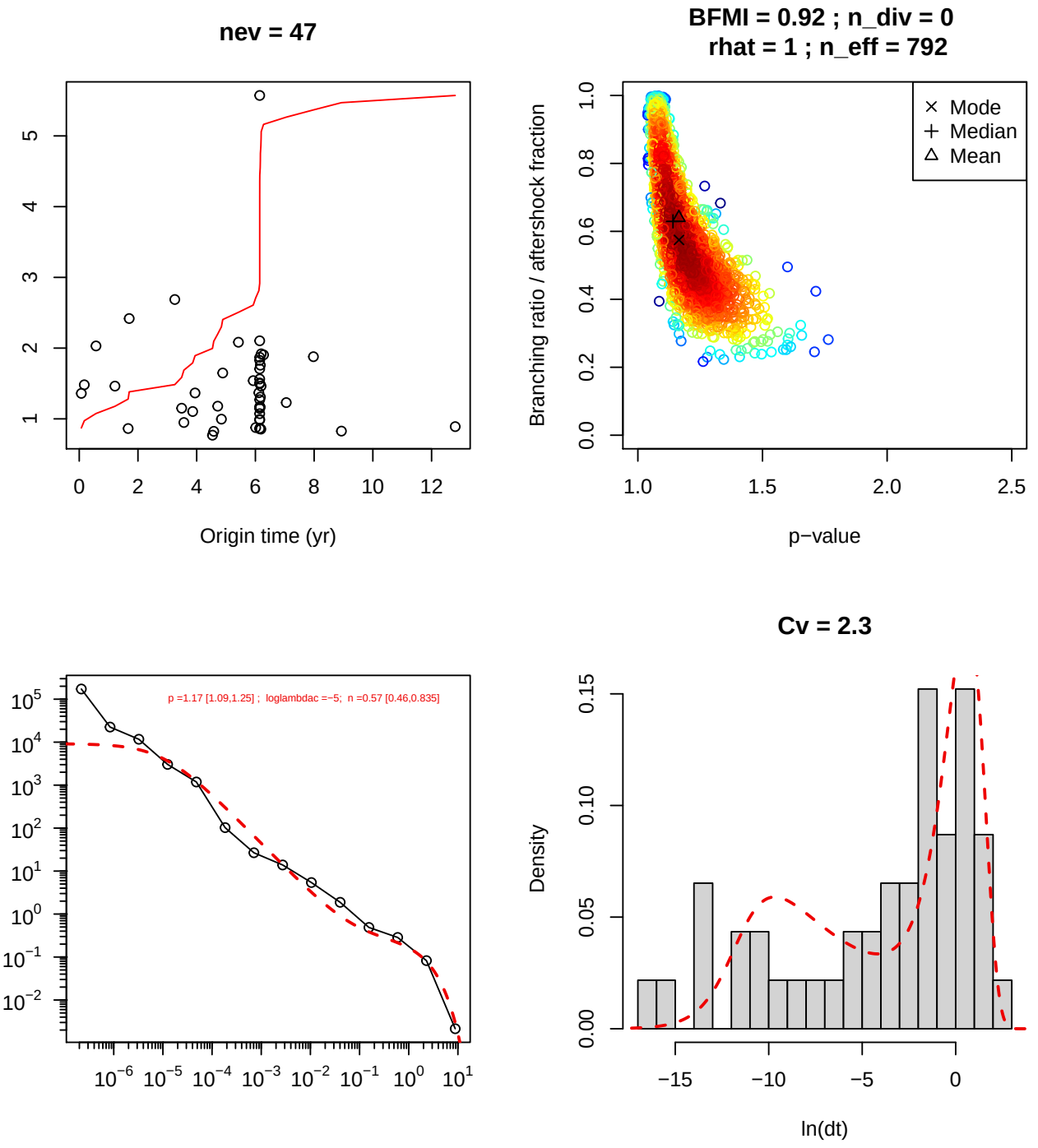
This estimates the aftershock fraction to be 0.92. Just given the data asked for my opinion, I probably would have counted the 25 events that occurred in the couple of years after the M5+ as aftershocks and said the aftershock fraction was $25/48 = 0.52$.

Leaving out very late events

Perhaps those last 1 events way off by themselves are biasing things? What happens if we leave them out?

```
use2 = 1:(length(eqs$t0)-nout)
st2 = do.stan(eqs, use=use2, alpha=alpha, c=c)
```

```
mkFourPanelPlot(eqs, use2, st2$bfmi, st2$n_div, st2$rhat, st2$n_eff,
                st2$f.stan,
                st2$p.stan.mode, st2$pCI.stan, st2$loglambdac.stan.mode,
                st2$n.stan.mode, st2$nCI.stan)
```



The estimated aftershock fraction is now closer to the naive estimate: 0.57.

Note that the estimate of *Hainzl et al.* [2006] would give 0.96 and 0.81, when using all events and all but the last 1 events, respectively.

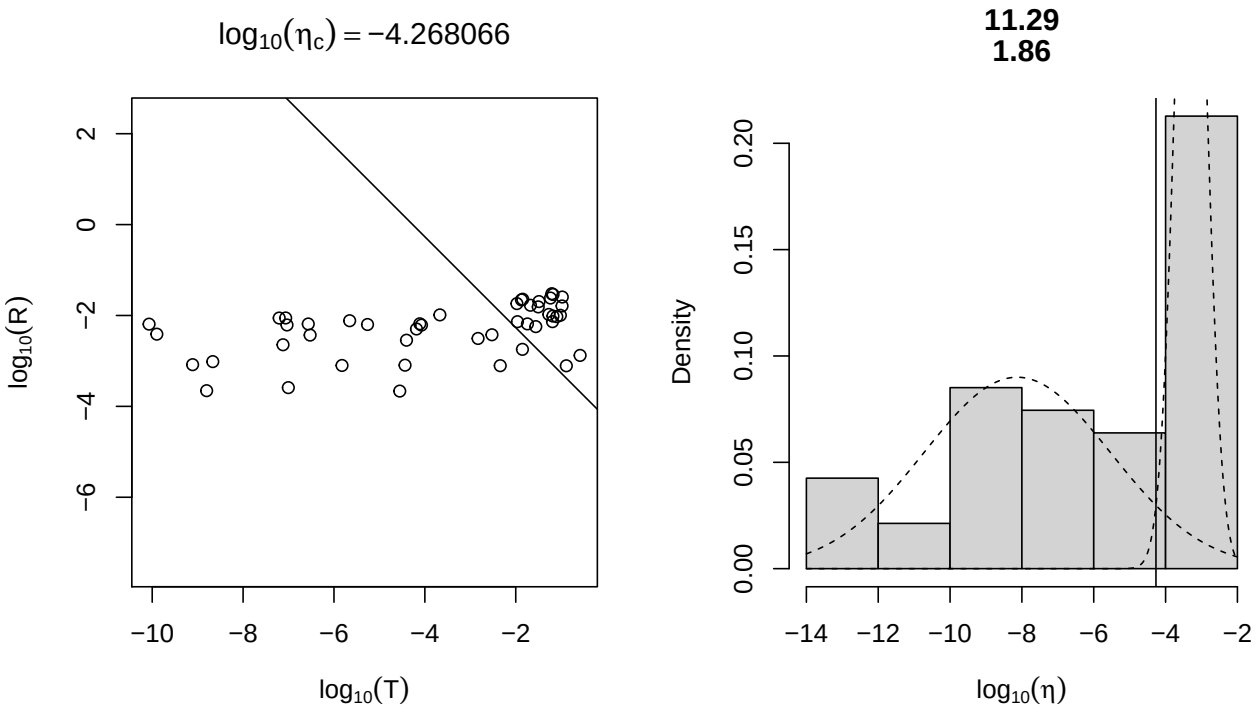
Both the estimated aftershock fraction from fitting the *Saichev and Sornette* [2007] functional form and the C_v are very affected by these outliers. If we used the 1-norm version of the C_v, call it C_{v1}, $C_{v1} = \text{smad}(dt)/\,\text{median}(dt)$, it would be essentially completely unaffected: including the event gives 1.4824747 and excluding it gives 1.4825836.

Aftershock fraction ala *Zaliapin and Ben-Zion* [2016]

```
sderrh = 10
z = zaliapin(eqs, use=use, sderrh=sderrh, est.se=TRUE)
z2 = zaliapin(eqs, use=use2, sderrh=sderrh)
```

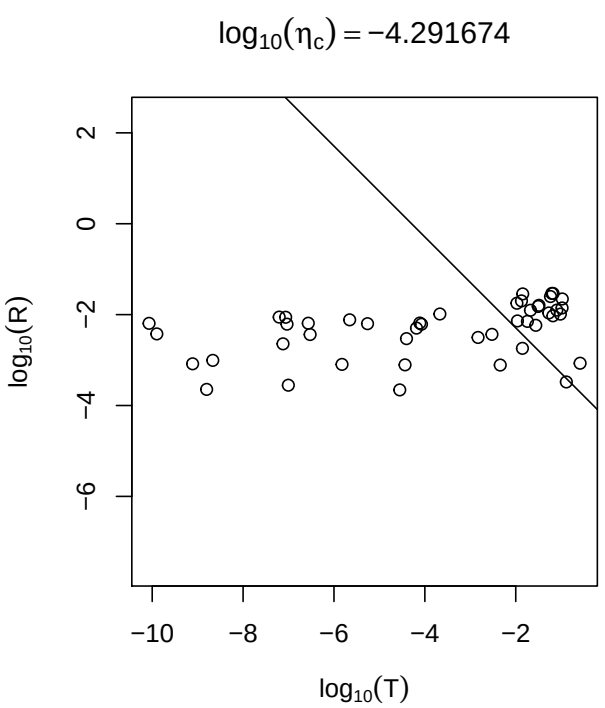
Using all events

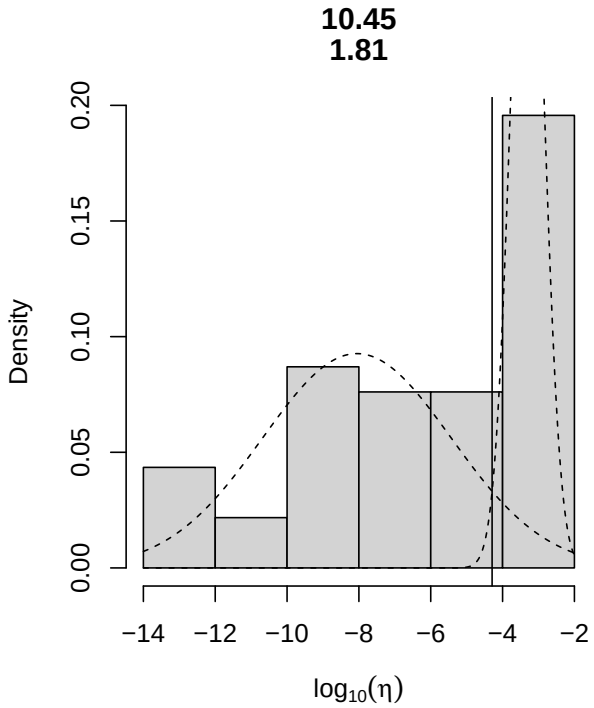
```
source("../mkZaliapinPlots.R")
mkZaliapinPlots(z)
```



Excluding late events

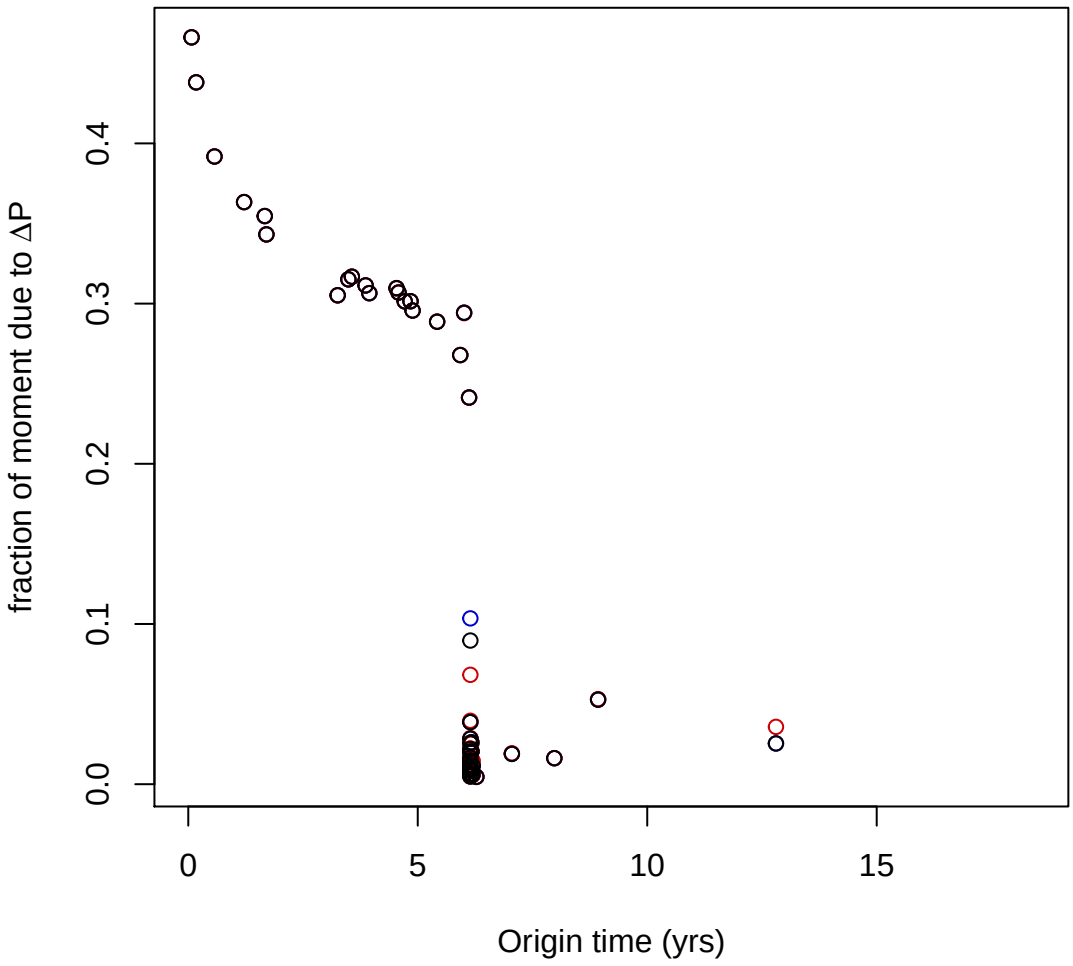
```
mkZaliapinPlots(z2)
```

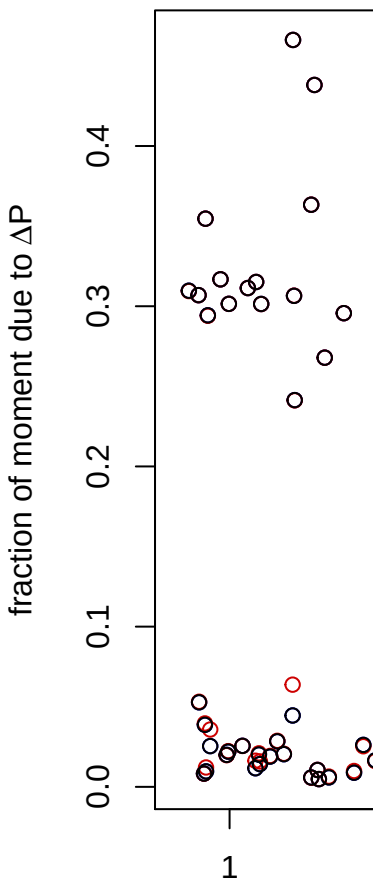


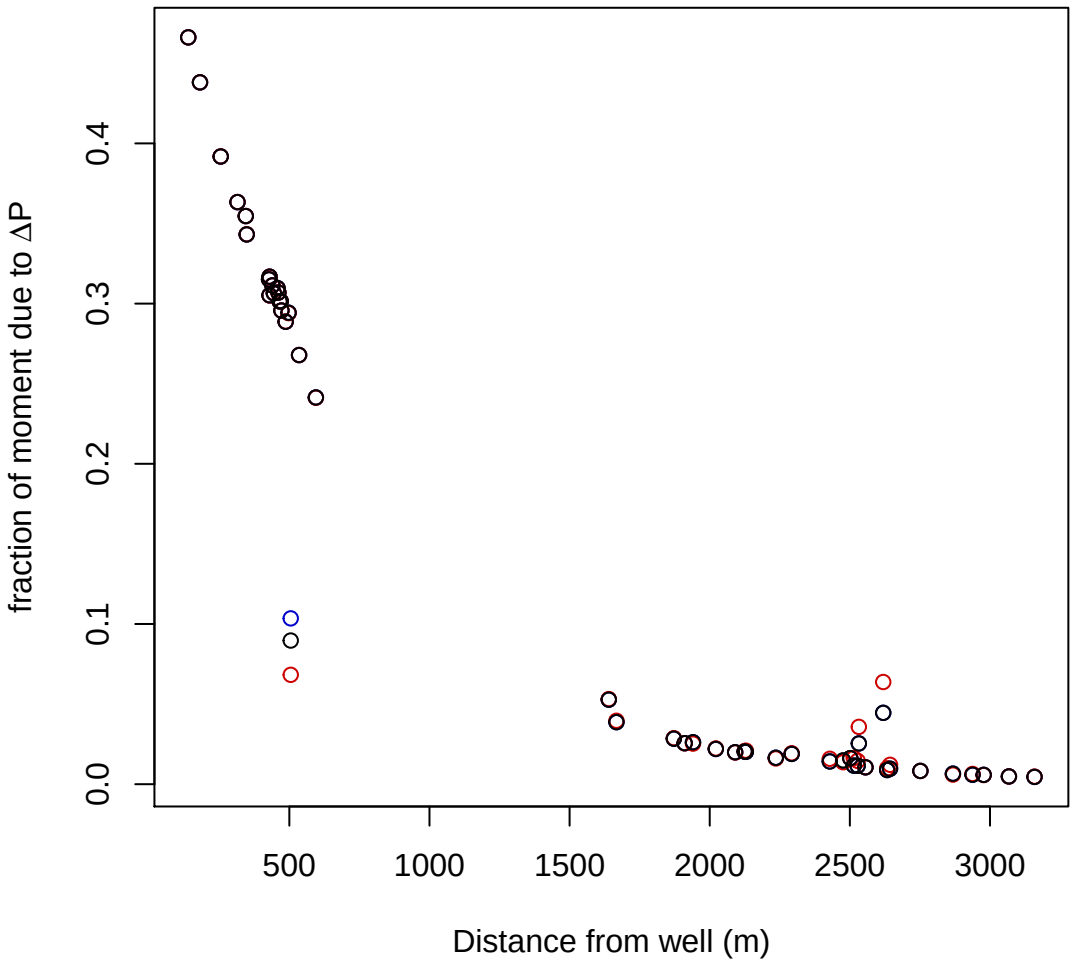


Moment due to ΔP

```
source("../plotMomentDueToDeltaP.R")
plotMomentDueToDeltaP(eqs, dwell)
```


fraction of moment due to ΔP
0.0
0.1
0.2
0.3
0.4
0
5
10
15
Origin time (yrs)


fraction of moment due to ΔP
0.0
0.1
0.2
0.3
0.4
1



Summary

```
source("../writeResults.R")
writeResults(paste(params$outFnameInfix, ".results.txt", sep=""),
             eqs, nout, nas, st, st2, z, z2)
results = readResults(paste(params$outFnameInfix, ".results.txt", sep=""))
```

- Full-main-fault event at 6.1 yrs
- Number of events
  - total: 48
  - before full-main-fault event: 20
  - after full-main-fault event: 27
- Estimated aftershock fractions
  - naive count: 0.52
  - SS2007
    - using all events: 0.92
    - excluding last 1 events: 0.57
  - H2006
    - using all events: 0.96
    - excluding last 1 events: 0.81
  - ZB2016
    - using all events: 0.55

* excluding last 1 events: 0.59

References

Hainzl, S., F. Scherbaum, and C. Beauval (2006), Estimating background activity based on interevent-time distribution, *Bulletin of the Seismological Society of America*, *96*(1), 313–320.

Saichev, A., and D. Sornette (2007), Theory of earthquake recurrence times, *Journal of Geophysical Research: Solid Earth*, *112*(B4).

Shapiro, S. A., E. Huenges, and G. Borm (1997), Estimating the crust permeability from fluid-injection-induced seismic emission at the ktb site, *Geophysical Journal International*, *131*(2), F15–F18.

Zaliapin, I., and Y. Ben-Zion (2016), Discriminating characteristics of tectonic and human-induced seismicity, *Bulletin of the Seismological Society of America*, *106*(3), 846–859.