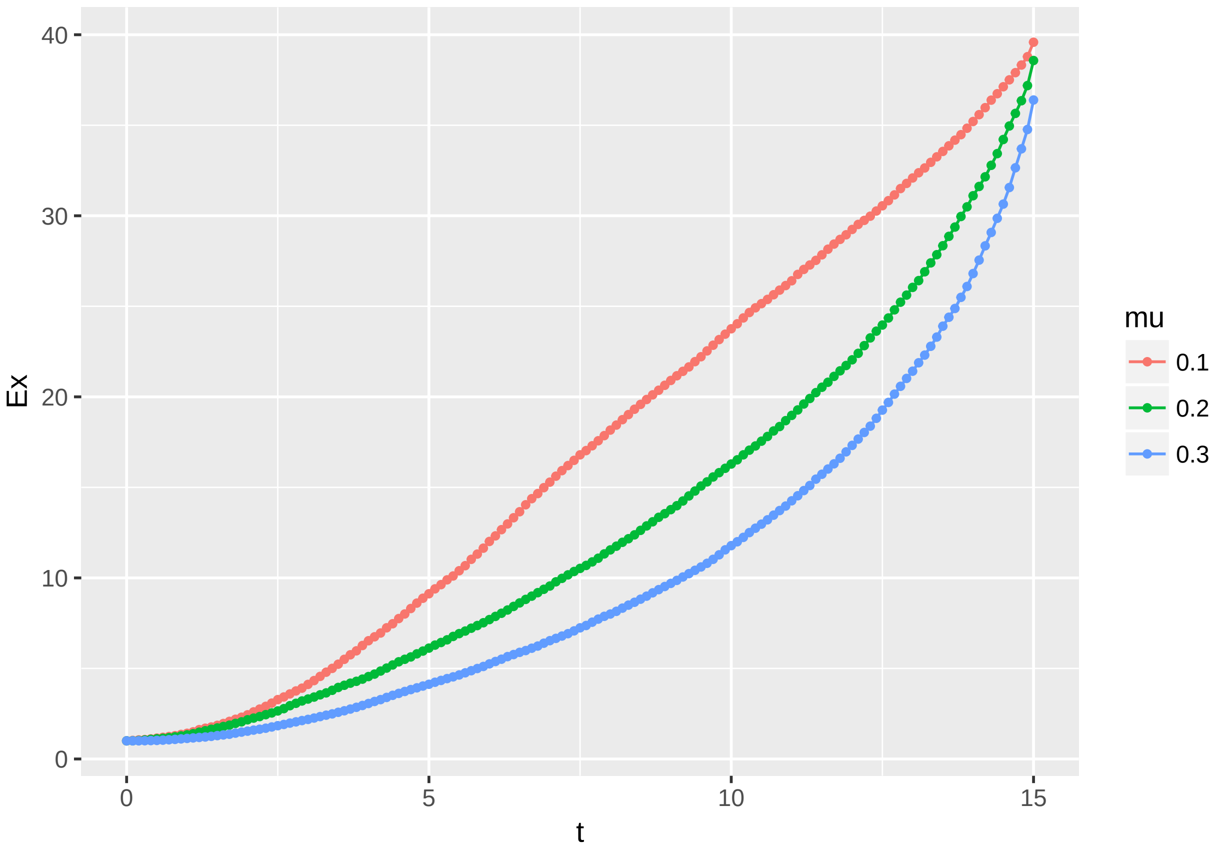
What extant species can tell us about extinction?

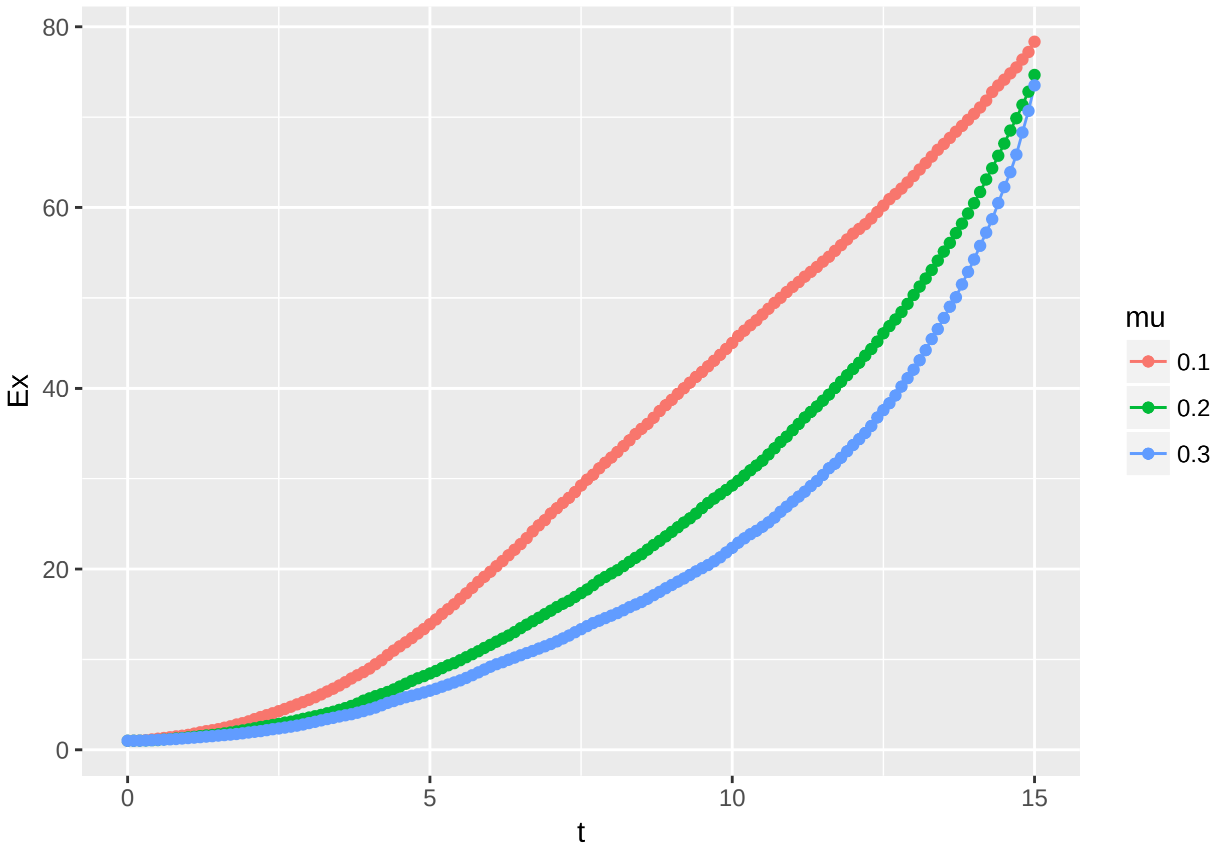
Because extinct species are rarelly included on phylogenetic trees, we are interested on invetigate the information that extant species contains about extinction rates. On the plot below we can see the expected Ltt plot, of extant-species only trees, for 3 different extinction rates



we can see a clear difference on Ltt plots of extant species, smaller extinction rates tents to grow faster on the begining whereas higger extinction rate seems to have a slow grow on the begining.

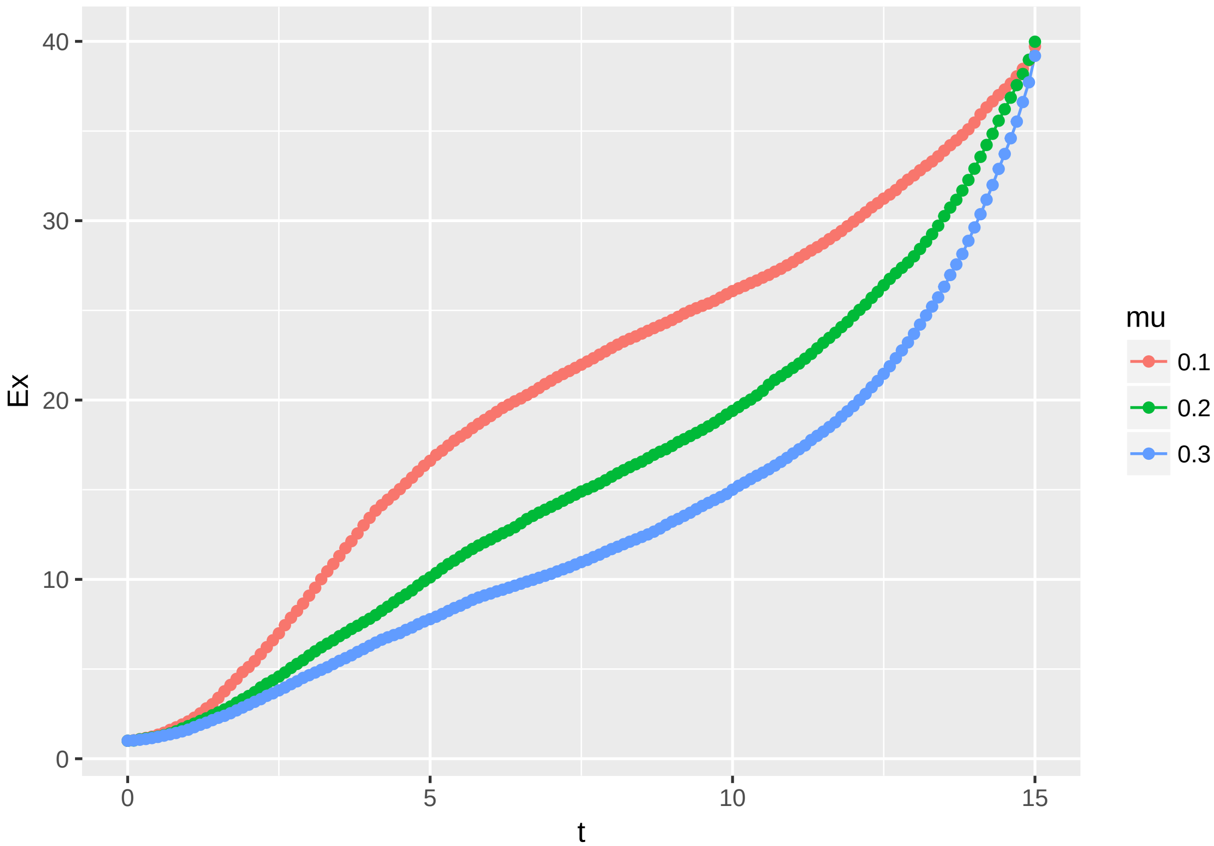
It seems also that is a matter of the first derivate, we can look at that also

Now we do it again, but with $K = 80$ rather than $K = 40$ in order to check some influence on the K parameter

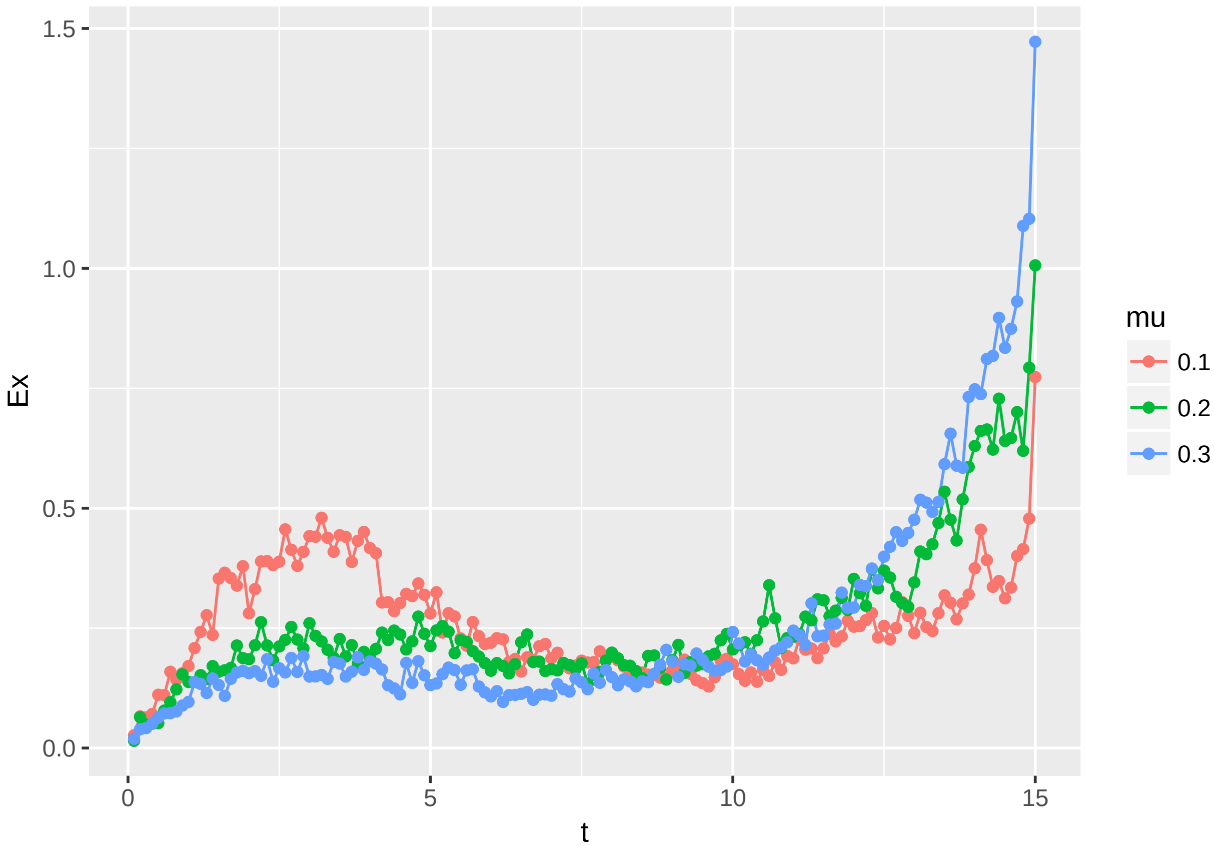


It seems it is just a change on scale. Now, what about λ?

We set $= \lambda = 1.2$ rather than $\lambda = 0.8$ and we see again the ltt plot



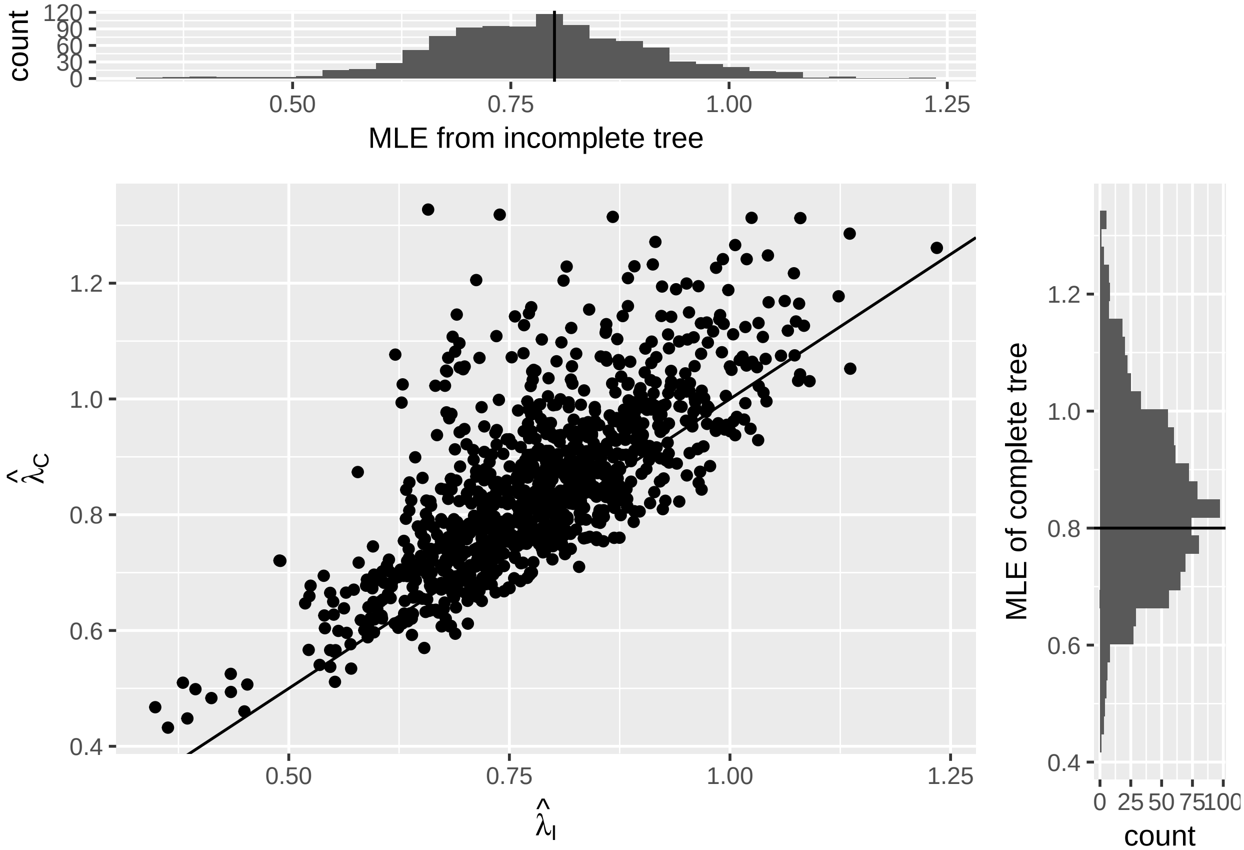
and we check the derivative again



2 parameter estimation (fixing μ)

```
p = proc.time()
n_it = 1000
mu = 0.1
n_trees=10
MP = matrix(nrow=n_it, ncol=3)
RP = matrix(nrow=n_it, ncol=3)
for (i in 1:n_it){
  s = sim_phyl()
  p <- subplex(par = c(2,0.2,60), fn = llik, n = s$n, E = s$E, t = s$wt)$par
  RP[i,] = p
  wt = (s$newick.extant.p)$wt
  trees = sim_srt(wt=wt, pars=c(p[1],mu,p[3]), parallel = F, n_trees = n_trees)
  pars = subplex(par = c(2,60), fn = llik_st , setoftrees = trees, mu = mu, impsam = FALSE)$par
  MP[i,] = c(pars[1],mu,pars[2])
}
par_est_vis(P=MP,par=1,PR=RP)
```

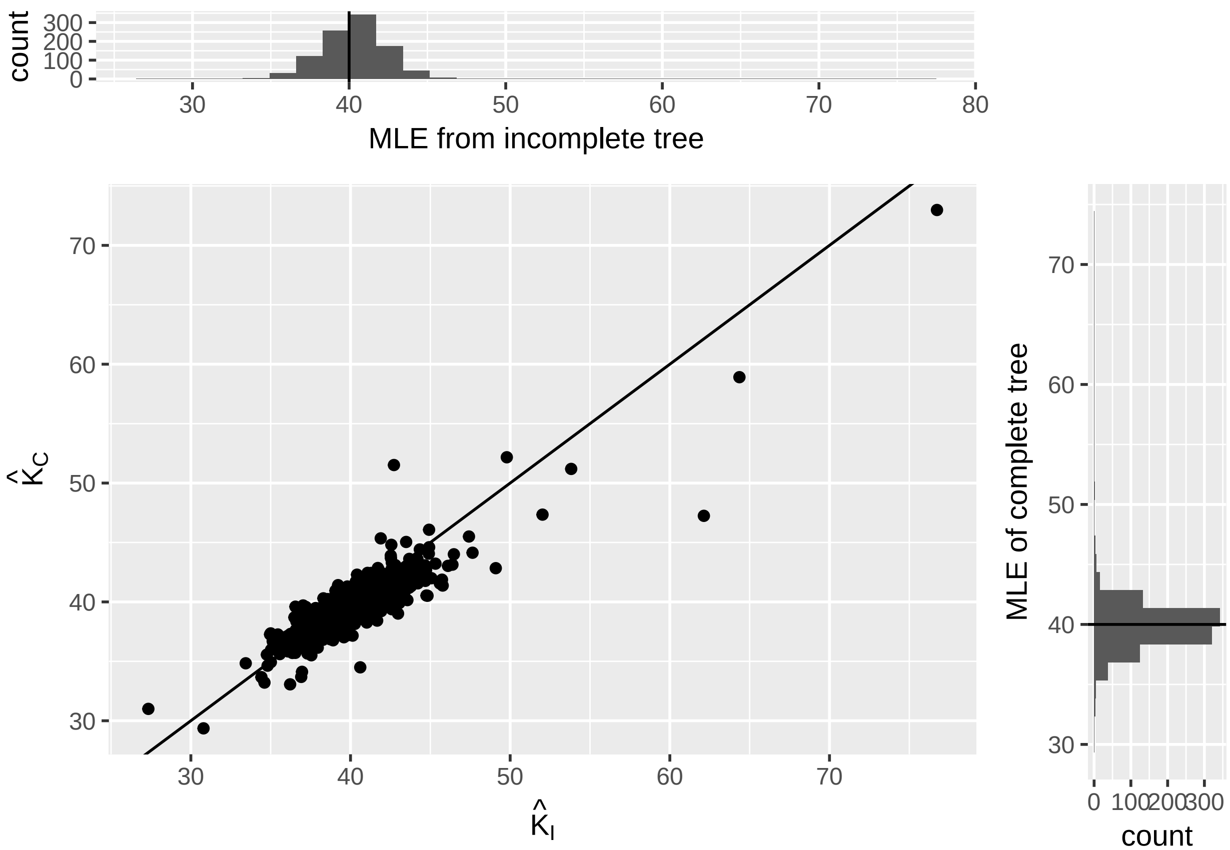
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
par_est_vis(P=MP,par=3,PR=RP)
```

```
## [1] "0.005 proportion of data was excluded for vizualization purposes"
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



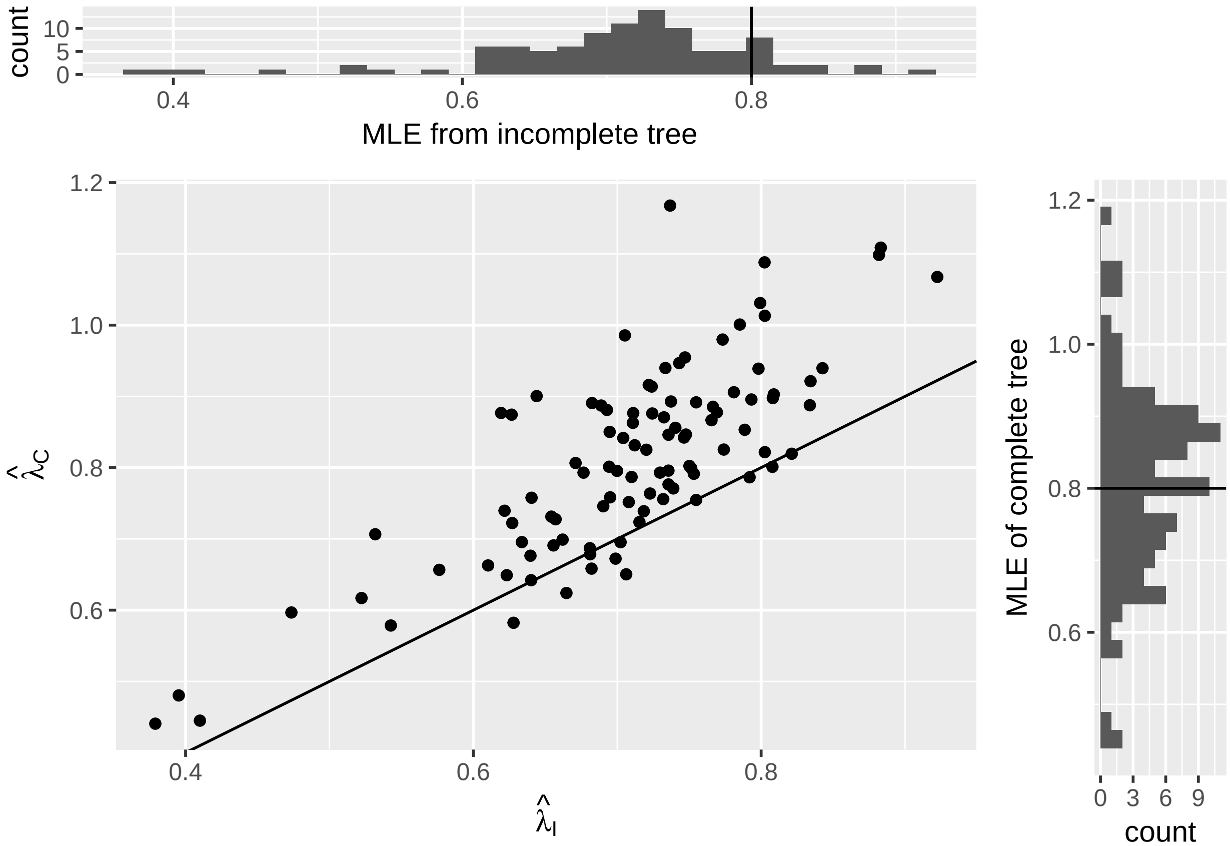

```
print(proc.time()-p)
```

```
## Warning in proc.time() - p: longer object length is not a multiple of
## shorter object length

##        user      system     elapsed
## 376.4501373  -0.1041349 332.9037043
```

```
p = proc.time()
n_it = 100
mu = 0.1
n_trees=100
MP = matrix(nrow=n_it, ncol=3)
RP = matrix(nrow=n_it, ncol=3)
for (i in 1:n_it){
  s = sim_phyl()
  p <- subplex(par = c(2,0.2,60), fn = llik, n = s$n, E = s$E, t = s$wt)$par
  RP[i,] = p
  wt = (s$newick.extant.p)$wt
  trees = sim_srt(wt=wt, pars=c(p[1],mu,p[3]), parallel = F, n_trees = n_trees)
  pars = subplex(par = c(2,60), fn = llik_st , setoftrees = trees, mu = mu, impsam = FALSE)$par
  MP[i,] = c(pars[1],mu,pars[2])
}
par_est_vis(P=MP,par=1,PR=RP)
```

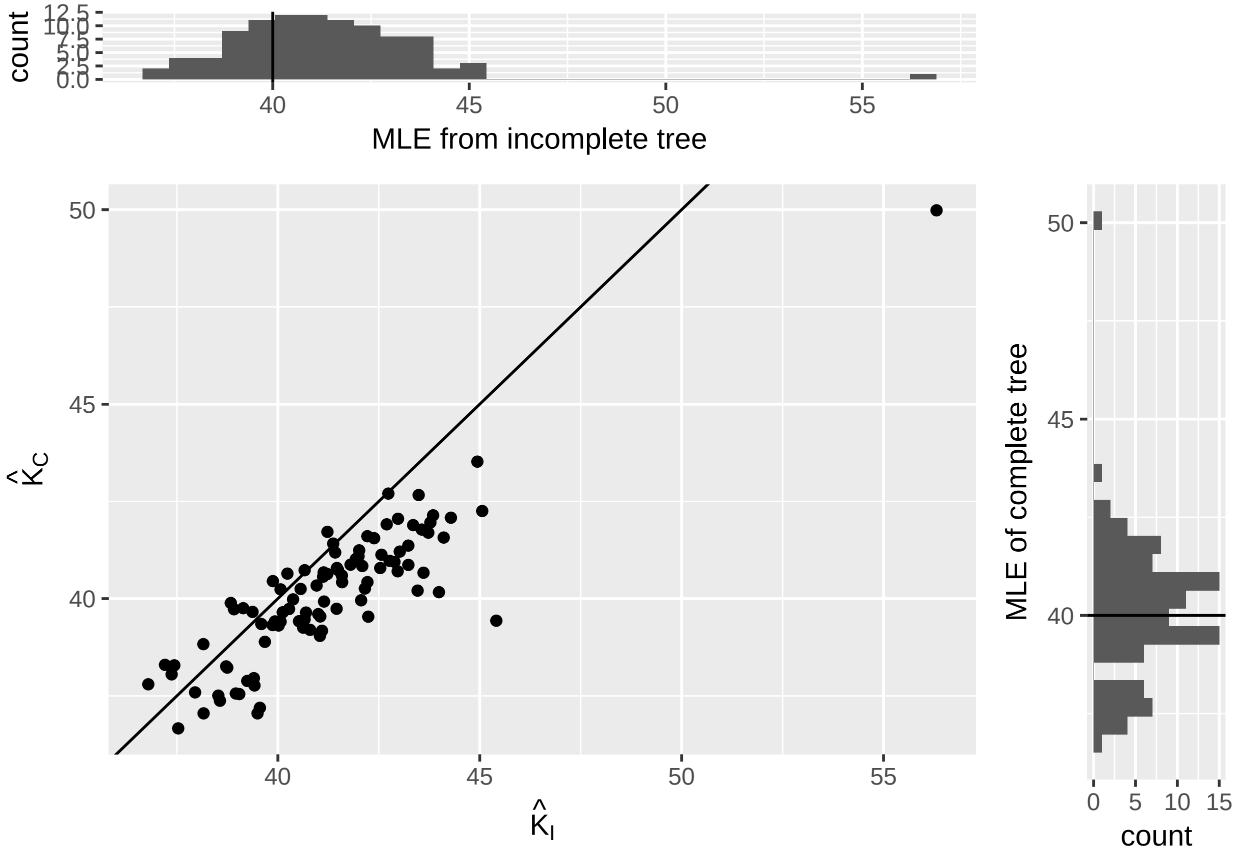
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
par_est_vis(P=MP,par=3,PR=RP)
```

```
## [1] "0.03 proportion of data was excluded for vizualization purposes"
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
print(proc.time()-p)
```

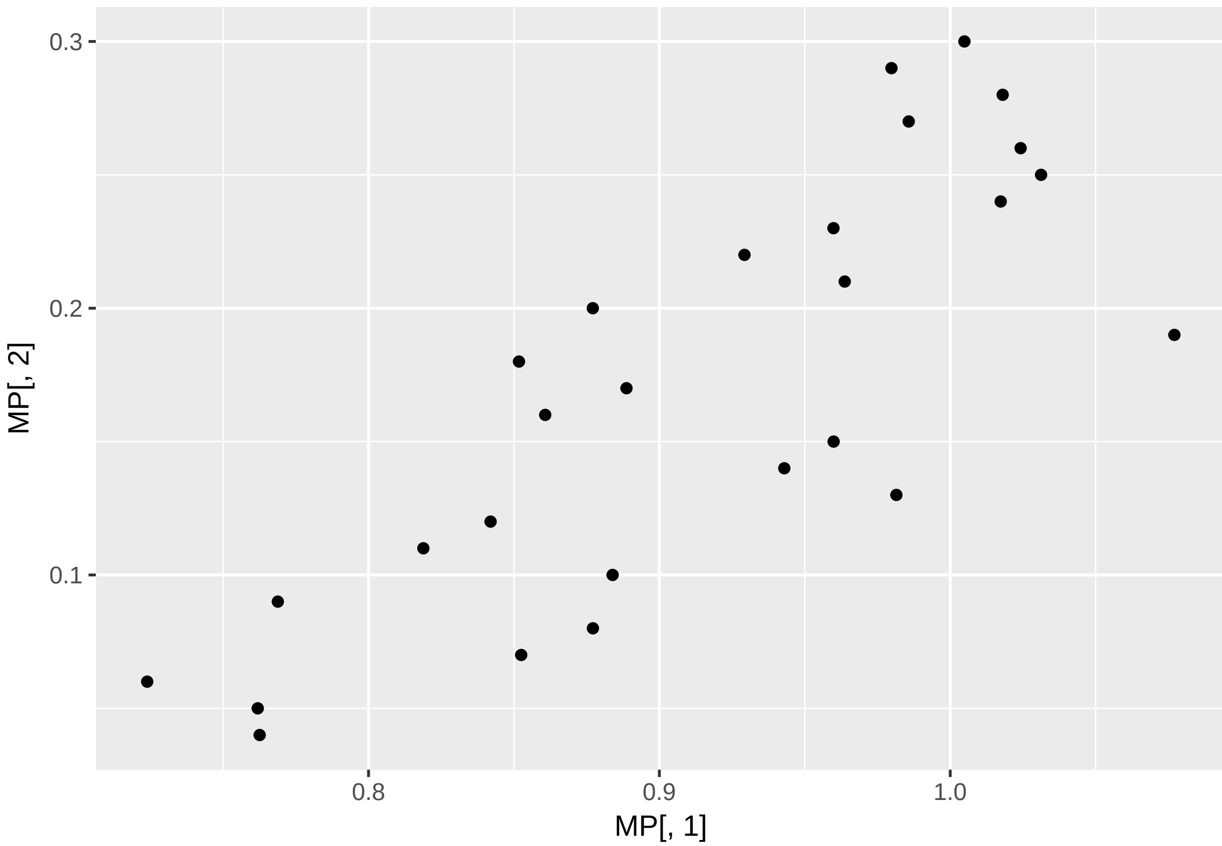
```
## Warning in proc.time() - p: longer object length is not a multiple of
## shorter object length

##          user       system      elapsed
## 193.24884721  -0.08130052 154.40579926
```

Ok. The estimations are fine.

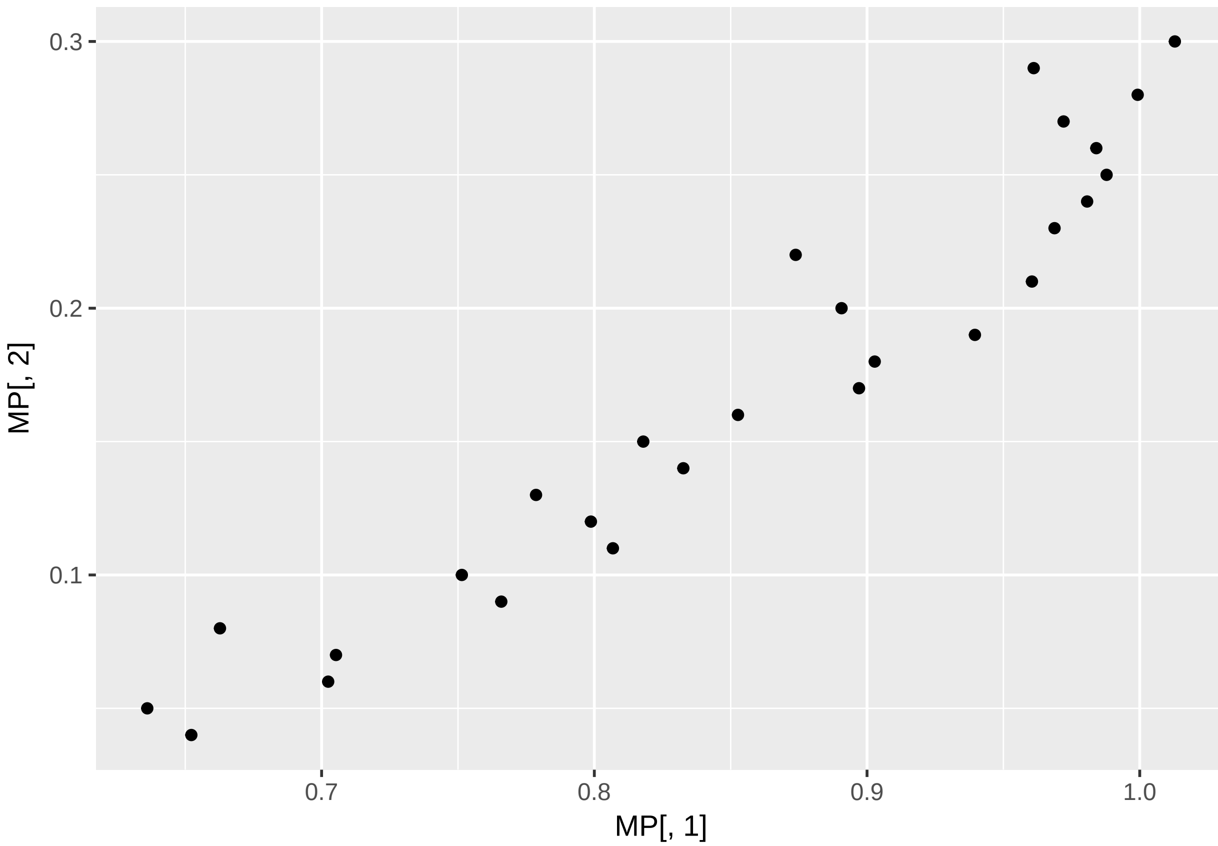
Now let´s try for a grid of μ

```r
mu0 = seq(0.04,0.3,by=0.01)
s = sim_phyl(seed=3)
p <- subplex(par = c(2,0.2,60), fn = llik, n = s$n, E = s$E, t = s$wt)$par
wt = (s$newick.extant.p)$wt
MP = matrix(nrow=length(mu0), ncol=3)
n_trees = 10
for(i in 1:length(mu0)){
  mu = mu0[i]
  trees = sim_srt(wt=wt, pars=c(p[1],mu,p[3]), parallel = F, n_trees = n_trees)
  pars = subplex(par = c(2,60), fn = llik_st , setoftrees = trees, mu = mu, impsam = FALSE)$par
  MP[i,] = c(pars[1],mu,pars[2])
}
qplot(MP[,1],MP[,2])
```



Does it help 100 trees (probably not)

```
mu0 = seq(0.04,0.3,by=0.01)
s = sim_phyl(seed=3)
p <- subplex(par = c(2,0.2,60), fn = llik, n = s$n, E = s$E, t = s$wt)$par
wt = (s$newick.extant.p)$wt
MP = matrix(nrow=length(mu0), ncol=3)
n_trees = 100
for(i in 1:length(mu0)){
  mu = mu0[i]
  trees = sim_srt(wt=wt, pars=c(p[1],mu,p[3]), parallel = F, n_trees = n_trees)
  pars = subplex(par = c(2,60), fn = llik_st , setoftrees = trees, mu = mu, impsam = FALSE)$par
  MP[i,] = c(pars[1],mu,pars[2])
}
qplot(MP[,1],MP[,2])
```



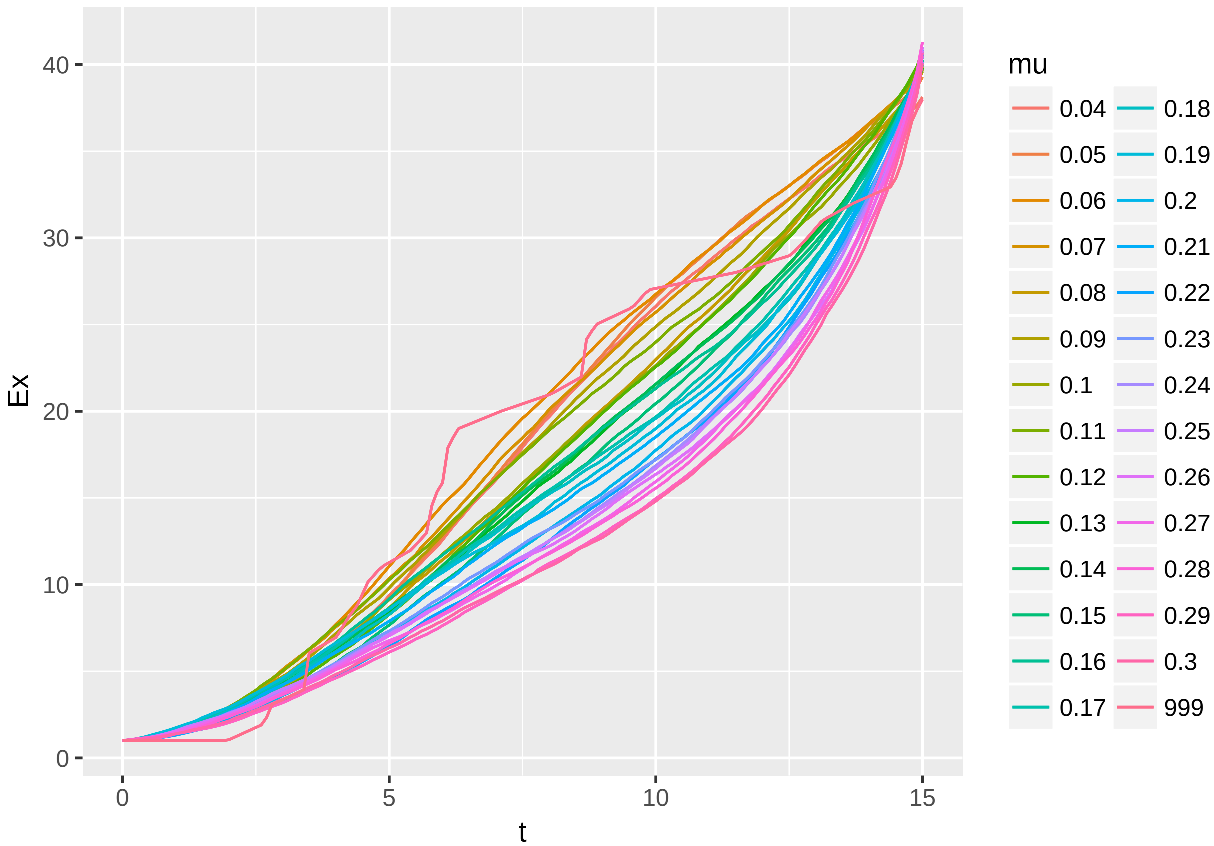
Actually, the variance decreases.

The Ltt plot

Now let's try to minimize ltt, but first vizualize it

```
ct = 15
dt = 0.1
grid = seq(0,ct, by=dt)
Ltt = data.frame(t=grid, Ex = approx(cumsum(wt), (s$newick.extant.p)$n, xou=grid,  rule = 2)$y, mu=999)

for(i in 1:length(mu0)){
  mu = mu0[i]
  pars = c(MP[i,1],MP[i,2],MP[i,3])
  ltt = data.frame(expectedLTT(pars,drop.extinct = TRUE),mu=mu)
  Ltt = rbind(Ltt,ltt)
}
Ltt$mu = as.factor(Ltt$mu)
ggplot(data=Ltt, aes(x=t, y=Ex, colour = mu)) + geom_line() +   geom_line()
```




```
ltt1 = Ltt[Ltt$mu == 999,]
diff_ltt = NaN
for(i in 1:length(mu0)){
 mu = mu0[i]
 ltt = Ltt[Ltt$mu == mu,]
 ltt$Ex = abs(ltt1$Ex-ltt$Ex)
 diff_ltt[i] = sum(ltt$Ex)
}
diff_ltt_M = data.frame(mu = mu0, diff_ltt = diff_ltt)
choosed_mu = diff_ltt_M[diff_ltt_M$diff_ltt == min(diff_ltt_M$diff_ltt) ,]
choosed_mu$mu
```

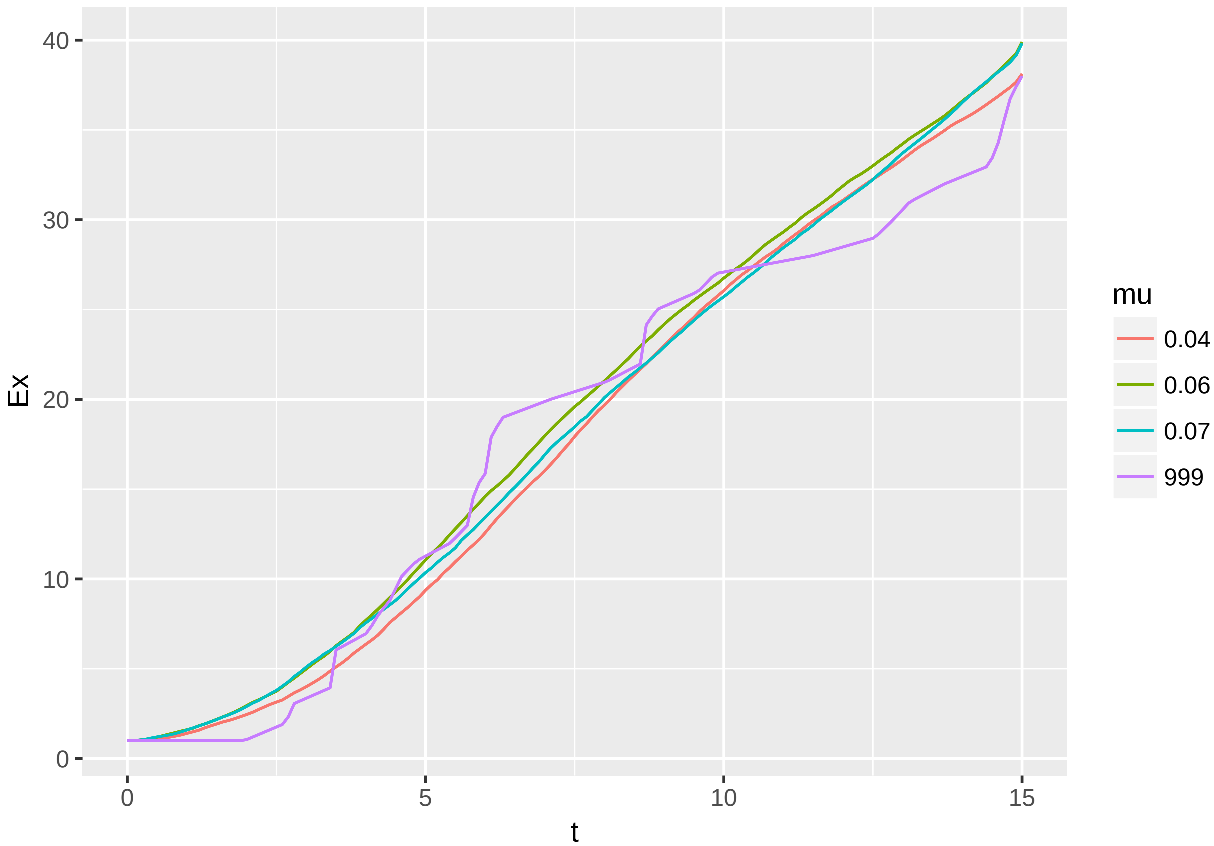
```
## [1] 0.06
```

```
MP[MP[,2] == choosed_mu$mu]
```

```
## [1]  0.7024064  0.0600000 40.2568662
```

```
#qqplot()
```

```
ch_mu = diff_ltt_M[which(diff_ltt_M$diff_ltt %in% sort(diff_ltt_M$diff_ltt)[1:3]),]$mu
Lttb = data.frame(t=grid, Ex = approx(cumsum(wt), (s$newick.extant.p)$n, xou=grid,  rule = 2)$y, mu=999)
for(i in 1:3){
  ltt = Ltt[Ltt$mu == ch_mu[i],]
  Lttb = rbind(Lttb,ltt)
}
ggplot(data=Lttb, aes(x=t, y=Ex, colour = mu)) + geom_line() +   geom_line()
```



Meta Analysis

Now we are prepared to estimate parameters ofr a set of (100) trees and see the distribution.

The algorithm is:

1. simulate tree and save MLE
2. drop extinct species and save ltt
3. create a grid μ_g and run monte-carlo for every $\mu \in mu_g$, then get $(\lambda(\mu), mu, K(\mu)), \forall \mu \in \mu_g$ and the corresponding ltt
4. take the best $(\lambda(\mu), \mu, K(\mu))$ taking min ltt

```
ct = 15
dt = 0.1
#grid = seq(0,ct, by=dt)
n_it = 10
mu0 = seq(0.04,0.3,by=0.01)
n_trees = 10
MMP = matrix(nrow=n_it, ncol=3)
RMP = matrix(nrow=n_it, ncol=3)
for(j in 1:n_it){
  s = sim_phyl()
  p <- subplex(par = c(2,0.2,60), fn = llik, n = s$n, E = s$E, t = s$wt)$par
  MMP[j,] = p
  s2 = s$newick.extant.p
  grid = s2$wt
```

```
  ltt1 = data.frame(t=grid, Ex = approx(cumsum(s2$wt), (s2$newick.extant.p)$n, xou=grid,  rule = 2)$y, m
  Ltt = ltt1
  MP = matrix(nrow=length(mu0), ncol=3)
  for(i in 1:length(mu0)){
    mu = mu0[i]
    trees = sim_srt(wt=wt, pars=c(p[1],mu,p[3]), parallel = F, n_trees = n_trees)
    pars = subplex(par = c(2,60), fn = llik_st , setoftrees = trees, mu = mu, impsam = FALSE)$par
    pars = c(pars[1],mu,pars[2])
    MP[i,] = pars
    ltt = data.frame(expectedLTT(pars,drop.extinct = TRUE, grid=grid),mu=mu)
    Ltt = rbind(Ltt,ltt)
  }
  diff_ltt = NaN
  for(i in 1:length(mu0)){
    mu = mu0[i]
    ltt = Ltt[Ltt$mu == mu,]
    ltt$Ex = abs(ltt1$Ex-ltt$Ex)
    diff_ltt[i] = sum(ltt$Ex)
  }
  diff_ltt_M = data.frame(mu = mu0, diff_ltt = diff_ltt)
  choosed_mu = diff_ltt_M[diff_ltt_M$diff_ltt == min(diff_ltt_M$diff_ltt) ,]
  RMP[j,] = MP[MP[,2]==choosed_mu$mu , ]
}
```

```
RMP
MMP
```