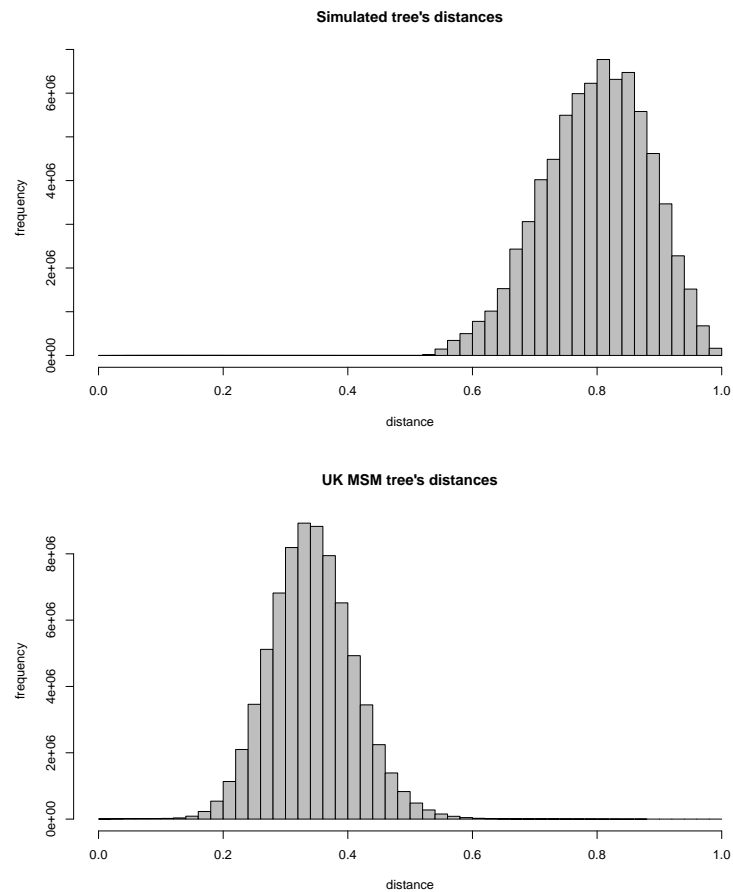


# Simulated tree clustered by UCSD soft. - Feb 2016

S. Le Vu  
(Dated: March 1, 2016)

## I. INTRO

- "time-based" distances have been extracted from simulated coalescent tree
- distances normalized from 0 to 1



## II. UCSD HIVCLUSTERING

Read saved results from UCSD hivclustering

```
### only cluster members
simclus <- readRDS(file = "data/simclus.rds")[[3]]
ukclus <- readRDS(file = "data/ukclus.rds")[[3]]
```

To construct clusters, thresholds were determined by quantiles of distances (0.05%, 0.1%, 1% and 10%). For simulated and UK trees

```
## read saved results of UCSD clustering
readRDS(file = "data/simclus.rds")[[1]]
```

0.05%	0.1%	1%	10%	25%	50%
0.2340219	0.5334507	0.5870679	0.6843790	0.7404369	0.8025368

```
readRDS(file = "data/ukclus.rds")[[1]]
```

0.05%	0.1%	1%	10%	25%	50%
0.06709549	0.10967080	0.19304638	0.25869004	0.29701520	0.34058792

Number of clusters and stats for simulated and UK trees (cluster size 1 does not exist in these outputs)

```
##- Calculate size(=Freq) of each cluster across different threshold
simfreqClust <- lapply(simclus,
  function(x) as.data.frame(table(x$ClusterID),
    stringsAsFactors = FALSE))
ukfreqClust <- lapply(ukclus,
  function(x) as.data.frame(table(x$ClusterID),
    stringsAsFactors = FALSE))
```

```
##- number of different clusters by threshold
sapply(simfreqClust, function(x) dim(x)[1])
```

```
0.23 0.53 0.59
1848 1357 529
```

```
sapply(ukfreqClust, function(x) dim(x)[1])
```

```
0.07 0.11 0.19
1490 1261 213
```

```
##- cluster size
sapply(simfreqClust, function(x) summary(x$Freq))
```

	0.23	0.53	0.59
Min.	2.000	2.000	2.00
1st Qu.	2.000	3.000	2.00
Median	4.000	5.000	4.00
Mean	5.924	8.565	22.49
3rd Qu.	7.250	10.000	8.00
Max.	47.000	989.000	8766.00

```
sapply(ukfreqClust, function(x) summary(x$Freq))
```

	0.07	0.11	0.19
Min.	2.000	2.000	2.00
1st Qu.	2.000	2.000	2.00
Median	3.000	3.000	2.00
Mean	5.117	7.427	54.67
3rd Qu.	4.000	5.000	3.00
Max.	162.000	2235.000	10900.00

Plots of log(size) for UK and simulated clusters

```
##- distr of cluster sizes: log(x) and log(y)
## how many plots
a <- length(simfreqClust)
b <- length(ukfreqClust)

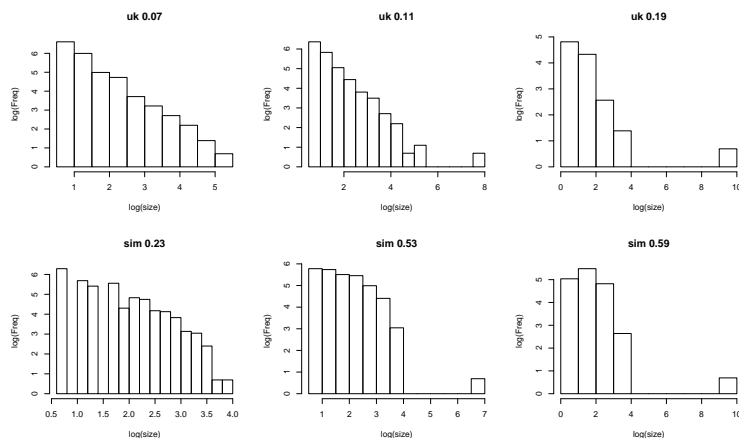
par(mfcol=c(2, max(a, b)))
for (i in 1:max(a, b)){
```

```

h <- hist(log(ukfreqClust[[i]]$Freq), plot = F)
h$counts <- log1p(h$counts) # log(y)
plot(h, ylab = "log(Freq)",
      main = paste("uk", names(ukfreqClust)[i]),
      xlab = "log(size)")

h <- hist(log(simfreqClust[[i]]$Freq), plot = F)
h$counts <- log1p(h$counts) # log(y)
plot(h, ylab = "log(Freq)",
      main = paste("sim", names(simfreqClust)[i]),
      xlab = "log(size)")
}

```



QQ plots UK vs simulated, untransformed and log-log

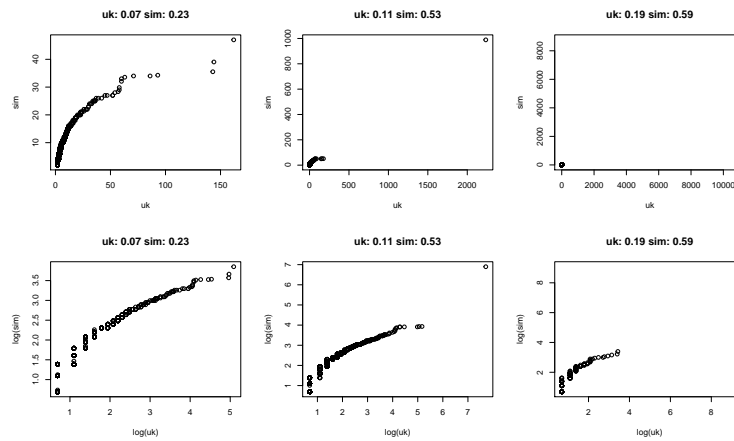
```

par(mfcol=c(2, max(a, b)))
for (i in 1:max(a, b)){
  qqplot(ukfreqClust[[i]]$Freq,
          simfreqClust[[i]]$Freq,
          main = paste("uk:", names(ukfreqClust)[i],
                        "sim:", names(simfreqClust)[i]),
          xlab = "uk", ylab = "sim")

  qqplot(log(ukfreqClust[[i]]$Freq),
          log(simfreqClust[[i]]$Freq),
          main = paste("uk:", names(ukfreqClust)[i],
                        "sim:", names(simfreqClust)[i]),
          xlab = "log(uk)", ylab = "log(sim)")
}
dev.off()

null device
1

```



### III. ASSOCIATIONS

After merging with co-variables allocated from demes states contained in tree, non-clustering individuals are assigned a cluster size of 1.

```
##- converting sample states in table of co-variables ?

if(TRUE){
# sampleTimes <- tree$sampleTimes
# sampleStates <- tree$sampleStates

demo <- matrix(NA, nrow = length(tree$sampleTimes), ncol = 6)
for (i in 1:dim(tree$sampleStates)[1]){ # dim(ss)[1]
  deme <- names(which.max(tree$sampleStates[i,])) # name of column which has max value
  patient <- as.numeric(rownames(tree$sampleStates)[i])
  time <- tree$sampleTimes[i]
  age <- as.numeric( regmatches( deme,
                                regexec( "\\age([0-9])", deme) )[[1]][2] )
  care <- as.numeric( regmatches( deme,
                                regexec( "care([0-9])", deme) )[[1]][2] )
  stage <- as.numeric( regmatches( deme,
                                regexec( "stage([0-9])", deme) )[[1]][2] )
  risk <- as.numeric( regmatches( deme,
                                regexec( "riskLevel([0-9])", deme) )[[1]][2] )
  demo[i,] <- cbind(patient, time, age, care, stage, risk)
}
colnames(demo) <- cbind("patient", "time",
                        "age", "care", "stage", "risk")

saveRDS(as.data.frame(demo), file = "demo.rds")
}
```

Error in matrix(NA, nrow = length(tree\$sampleTimes), ncol = 6): object 'tree' not found

The proportion of individuals into clusters and stats for "size of cluster for each individuals"

```
##-proportion in or out clusters
sapply(1, function(x) round(prop.table(table(x$binclus)),2))

0.23 0.53 0.59
0 0.1 0.04 0.02
1 0.9 0.96 0.98
```

```
##- cluster sizes (by individuals having such a size !!)
sapply(1, function(x) summary(x$size))
```

	0.23	0.53	0.59
Min.	1.000	1.00	1
1st Qu.	3.000	6.00	20
Median	8.000	13.00	8766
Mean	9.889	93.94	6320
3rd Qu.	14.000	25.00	8766
Max.	47.000	989.00	8766

#### IV. NAIVE REGRESSIONS ON SIMULATION

##### Linear

```
##### just on low and high threshold
simli <- listclus[c(1:length(listclus))]
```

```
lm_model_std = "scale(size) ~ scale(age) + scale(stage) + scale(time) + scale(risk)"
lapply(simli, function(x) summary(lm(lm_model_std, data = x)))
```

\$`0.23`

Call:

```
lm(formula = lm_model_std, data = x)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.5786	-0.6667	-0.2748	0.4126	5.0163

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-9.297e-16	8.715e-03	0.000	1.00000
scale(age)	-1.061e-02	8.783e-03	-1.208	0.22695
scale(stage)	-2.794e-02	8.788e-03	-3.180	0.00148 **
scale(time)	2.711e-01	8.749e-03	30.988	< 2e-16 ***
scale(risk)	-2.295e-02	8.717e-03	-2.633	0.00848 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9612 on 12159 degrees of freedom

Multiple R-squared: 0.07647, Adjusted R-squared: 0.07616

F-statistic: 251.7 on 4 and 12159 DF, p-value: < 2.2e-16

\$`0.53`

Call:

```
lm(formula = lm_model_std, data = x)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.5101	-0.3449	-0.2843	-0.2215	3.4788

Coefficients:

Estimate	Std. Error	t value	Pr(> t )
----------	------------	---------	----------

```

(Intercept)  5.133e-15  9.050e-03  0.000  1.000
scale(age)   -5.690e-03  9.121e-03 -0.624  0.533
scale(stage) -5.758e-03  9.127e-03 -0.631  0.528
scale(time)  -6.303e-02  9.086e-03 -6.937  4.2e-12 ***
scale(risk)   5.542e-03  9.052e-03  0.612  0.540
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9982 on 12159 degrees of freedom
Multiple R-squared:  0.003982, Adjusted R-squared:  0.003655
F-statistic: 12.15 on 4 and 12159 DF,  p-value: 7.344e-10

$`0.59`

Call:
lm(formula = lm_model_std, data = x)

Residuals:
    Min       1Q   Median       3Q      Max
-1.8811 -1.3122  0.4819  0.7101  1.0111

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.091e-14  8.851e-03  0.000    1.000
scale(age)   -3.231e-03  8.920e-03 -0.362    0.717
scale(stage) -5.990e-03  8.925e-03 -0.671    0.502
scale(time)  -2.184e-01  8.886e-03 -24.575 <2e-16 ***
scale(risk)  -1.773e-03  8.853e-03 -0.200    0.841
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9761 on 12159 degrees of freedom
Multiple R-squared:  0.04746, Adjusted R-squared:  0.04715
F-statistic: 151.5 on 4 and 12159 DF,  p-value: < 2.2e-16

##- univariate
summary(lm(
  scale(size) ~ scale(stage),
  data = simli[[3]]
))

Call:
lm(formula = scale(size) ~ scale(stage), data = simli[[3]])

Residuals:
    Min       1Q   Median       3Q      Max
-1.6223 -1.5925  0.6158  0.6303  0.6376

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.155e-14  9.067e-03  0.000    1.000
scale(stage)  9.264e-03  9.067e-03  1.022    0.307

Residual standard error: 1 on 12162 degrees of freedom
Multiple R-squared:  8.583e-05, Adjusted R-squared:  3.61e-06
F-statistic: 1.044 on 1 and 12162 DF,  p-value: 0.3069

```

```
summary(lm(
  scale(size) ~ scale(risk),
  data = simli[[1]])
))

Call:
lm(formula = scale(size) ~ scale(risk), data = simli[[1]])

Residuals:
    Min       1Q   Median       3Q      Max
-1.0636 -0.7670 -0.2354  0.4745  4.4386

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.548e-15  9.065e-03   0.000  1.00000
scale(risk)  -2.396e-02  9.065e-03  -2.643  0.00823 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9998 on 12162 degrees of freedom
Multiple R-squared:  0.000574, Adjusted R-squared:  0.0004918
F-statistic: 6.985 on 1 and 12162 DF, p-value: 0.008229
```

### Logistic

```
##- model: clus ~ age + stage + time + risk
##- care = 1 for all at diagnosis
## ex.
logit_model_std = "binclus ~ scale(age) + scale(stage) + scale(time) + scale(risk)"
lapply(simli , function(x) summary(glm(formula = logit_model_std,
                                       data = x,
                                       family = binomial(link = "logit")))
))

$`0.23`

Call:
glm(formula = logit_model_std, family = binomial(link = "logit"),
    data = x)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0628  0.2197  0.3236  0.4602  1.2497

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.60860    0.04097  63.672 < 2e-16 ***
scale(age)   -0.09705    0.03496  -2.776  0.0055 **
scale(stage) -0.13678    0.03232  -4.232 2.31e-05 ***
scale(time)  0.98278    0.03314  29.660 < 2e-16 ***
scale(risk)  -0.14937    0.02985  -5.005 5.59e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 7906.9  on 12163  degrees of freedom
```

Residual deviance: 6824.6 on 12159 degrees of freedom  
AIC: 6834.6

Number of Fisher Scoring iterations: 6

\$`0.53`

Call:

```
glm(formula = logit_model_std, family = binomial(link = "logit"),
     data = x)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-3.1771	0.1841	0.2435	0.3244	0.7247

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	3.35884	0.05545	60.571	< 2e-16 ***
scale(age)	-0.11499	0.04934	-2.331	0.01978 *
scale(stage)	-0.12598	0.04521	-2.786	0.00533 **
scale(time)	0.74855	0.04452	16.812	< 2e-16 ***
scale(risk)	-0.09854	0.04205	-2.344	0.01909 *

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 4425.6 on 12163 degrees of freedom  
Residual deviance: 4097.1 on 12159 degrees of freedom  
AIC: 4107.1

Number of Fisher Scoring iterations: 6

\$`0.59`

Call:

```
glm(formula = logit_model_std, family = binomial(link = "logit"),
     data = x)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.9316	0.1921	0.2072	0.2231	0.2734

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	3.82076	0.06319	60.461	< 2e-16 ***
scale(age)	-0.05900	0.06416	-0.919	0.35784
scale(stage)	-0.16270	0.06298	-2.583	0.00978 **
scale(time)	-0.03569	0.06271	-0.569	0.56921
scale(risk)	-0.09539	0.05813	-1.641	0.10079

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2559.8 on 12163 degrees of freedom



```
Residual deviance: 2548.8 on 12159 degrees of freedom
AIC: 2558.8
```

```
Number of Fisher Scoring iterations: 6
```

```
###--- sort dependency between individuals from same cluster
### 1. downsample to make analysis of one cluster size
### explained by median or mean of each co-variate.
### 2. plot the distribution of covariates by cluster size.
### With the intuition that smaller clusters are more explained
### by covariates and larger ones are more random.
### Do it on real data and simulation

# For each cluster size, compute mean of all covariates
```

## V. REGRESSIONS ON DOWN-SAMPLED SIMULATION

To sort out the dependency between individuals from same cluster

1. "downsample" to make analysis of each cluster size explained by mean of each co-variate (from here, only clusters from lower and higher threshold represented)

```
##- 1. down-sample: mean of each variable
down <- lapply(simli, function(x) aggregate(x[, 5:9], list("size" = x$size), mean))
# str(down)
```

```
##- linear regression
```

```
lapply(down, function(x) summary(lm(lm_model_std, data = x)))
```

```
$`0.23`
```

```
Call:
```

```
lm(formula = lm_model_std, data = x)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-1.1082 -0.5070 -0.2067  0.4938  2.7281
```

```
Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.337e-16  1.381e-01   0.000   1.0000
scale(age)   -1.385e-01  1.582e-01  -0.876   0.3878
scale(stage) -1.781e-01  1.684e-01  -1.058   0.2981
scale(time)   4.213e-01  1.963e-01   2.146   0.0396 *
scale(risk)  -8.551e-02  1.744e-01  -0.490   0.6274
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.84 on 32 degrees of freedom
```

```
Multiple R-squared:  0.3728, Adjusted R-squared:  0.2944
```

```
F-statistic: 4.756 on 4 and 32 DF, p-value: 0.003991
```

```
$`0.53`
```

```
Call:
```

```
lm(formula = lm_model_std, data = x)

Residuals:
    Min       1Q   Median       3Q      Max
-0.7323 -0.2221 -0.1173 -0.0221  6.3887

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.039e-16  1.500e-01   0.000   1.000
scale(age)   -1.030e-01  1.737e-01  -0.593   0.556
scale(stage) -1.217e-01  1.909e-01  -0.637   0.527
scale(time)  -2.281e-01  1.897e-01  -1.202   0.236
scale(risk)   3.922e-02  1.640e-01   0.239   0.812

Residual standard error: 1.029 on 42 degrees of freedom
Multiple R-squared:  0.03415, Adjusted R-squared:  -0.05783
F-statistic: 0.3713 on 4 and 42 DF,  p-value: 0.8278

$`0.59`

Call:
lm(formula = lm_model_std, data = x)

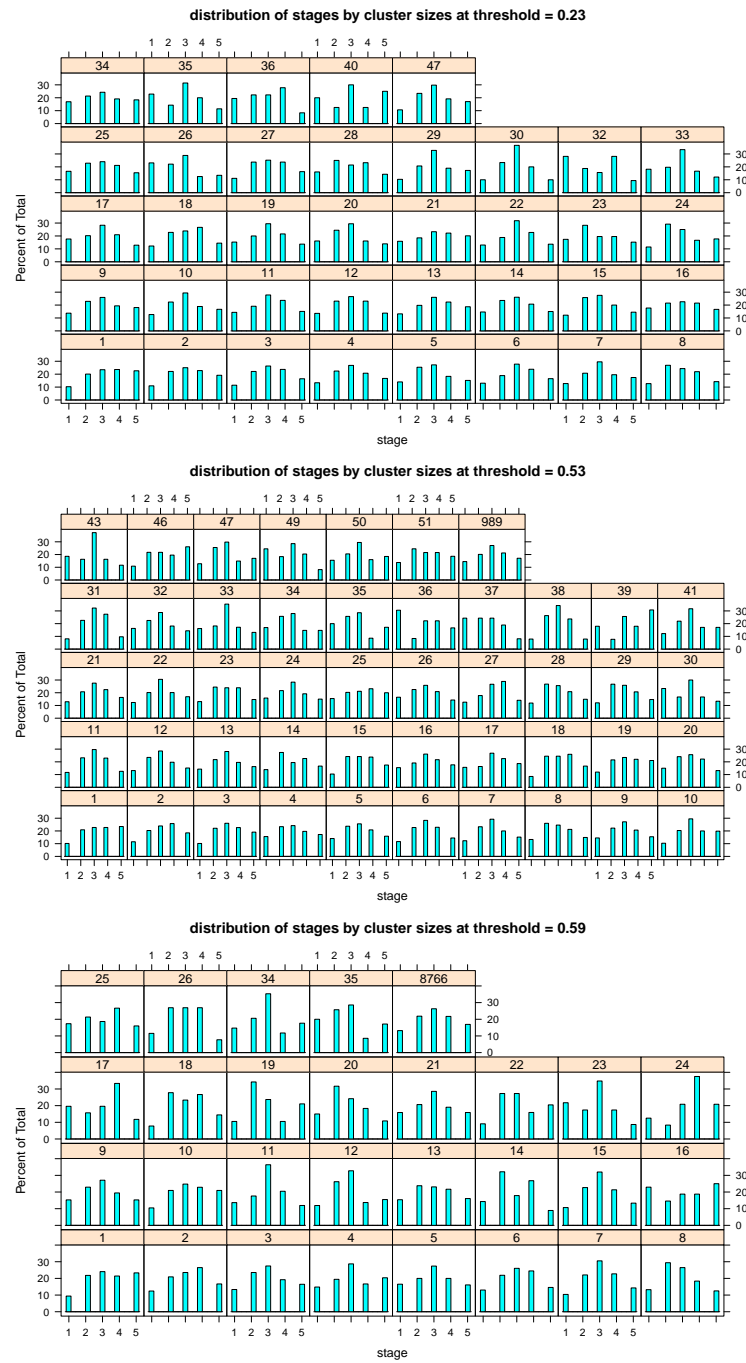
Residuals:
    Min       1Q   Median       3Q      Max
-1.0914 -0.4985 -0.0551  0.1814  3.7754

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.422e-16  1.711e-01   0.000  1.00000
scale(age)    1.214e-01  2.132e-01   0.570  0.57425
scale(stage) -2.134e-01  2.051e-01  -1.040  0.30864
scale(time)  -5.841e-01  1.967e-01  -2.970  0.00666 **
scale(risk)  -9.436e-02  2.090e-01  -0.451  0.65570
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9215 on 24 degrees of freedom
Multiple R-squared:  0.2722, Adjusted R-squared:  0.1509
F-statistic: 2.244 on 4 and 24 DF,  p-value: 0.09424
```

## 2. plot the distribution of covariates by cluster size

```
##- 2. plots
library(lattice)
# trellis.par.set(canonical.theme(color = FALSE))
for(i in 1:length(simli)){
  print(histogram(~ stage|factor(size),
    main = paste("distribution of stages by cluster sizes at threshold =", names(simli[i])),
    data = simli[[i]])
  )
}
```



## VI. ON REAL UK DATA

Same process ...

Proportion in and out clusters

```
##-proportion in or out clusters
sapply(1, function(x) round(prop.table(table(x$binclus)),2))

0.07 0.11 0.19
0 0.37 0.23 0.04
1 0.63 0.77 0.96
```

```
##- cluster sizes (by individuals having such a size !!)
sapply(l, function(x) summary(x$size))
```

```
      0.07  0.11  0.19
Min.    1.00    1.0    1
1st Qu.  1.00    2.0 10900
Median   3.00    8.0 10900
Mean    15.47  426.5  9762
3rd Qu. 12.00   62.0 10900
Max.   162.00 2235.0 10900
```

### Naive regressions

```
#### just on low and high threshold (but not too high !)
li <- listUKclus[ 1:(length(listUKclus)-1) ]
lm_model_std = "scale(size) ~ scale(agediag) + scale(sqrt(cd4)) + scale(ydiag)"
lapply(li, function(x) summary(lm(lm_model_std, data = x)))
```

```
$`0.07`
```

```
Call:
```

```
lm(formula = lm_model_std, data = x)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-0.8338 -0.4714 -0.2992 -0.0755  4.9307
```

```
Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.003156   0.009105   0.347  0.72893
scale(agediag) -0.025564   0.009495  -2.692  0.00711 **
scale(sqrt(cd4)) 0.048448   0.009239   5.244  1.6e-07 ***
scale(ydiag)    0.166237   0.009431  17.627 < 2e-16 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.9916 on 11857 degrees of freedom
```

```
(303 observations deleted due to missingness)
```

```
Multiple R-squared:  0.02869, Adjusted R-squared:  0.02844
```

```
F-statistic: 116.7 on 3 and 11857 DF, p-value: < 2.2e-16
```

```
$`0.11`
```

```
Call:
```

```
lm(formula = lm_model_std, data = x)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-1.1701 -0.5395 -0.3606 -0.1872  2.5272
```

```
Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.0001816  0.0089649   0.020  0.9838
scale(agediag)  0.0757349  0.0093484   8.101 5.97e-16 ***
scale(sqrt(cd4)) 0.0165371  0.0090965   1.818  0.0691 .
scale(ydiag)   -0.2140942  0.0092853 -23.057 < 2e-16 ***
```

```
---
```

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9763 on 11857 degrees of freedom  
 (303 observations deleted due to missingness)  
 Multiple R-squared: 0.04349, Adjusted R-squared: 0.04324  
 F-statistic: 179.7 on 3 and 11857 DF, p-value: < 2.2e-16

```
##- model: clus ~ age + stage + time + risk
##- care = 1 for all at diagnosis
## ex.
logit_model_std = "binclus ~ scale(agediag) + scale(sqrt(cd4)) + scale(ydiag)"
lapply(li, function(x) summary(glm(formula = logit_model_std,
                                   data = x,
                                   family = binomial(link = "logit")))
      ))
```

\$`0.07`

Call:

```
glm(formula = logit_model_std, family = binomial(link = "logit"),
    data = x)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.2101	-1.0772	0.6772	0.8944	2.0267

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.58725	0.02050	28.646	< 2e-16 ***
scale(agediag)	-0.14644	0.02125	-6.892	5.49e-12 ***
scale(sqrt(cd4))	0.26123	0.02067	12.640	< 2e-16 ***
scale(ydiag)	0.74823	0.02197	34.057	< 2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 15620 on 11860 degrees of freedom  
 Residual deviance: 14116 on 11857 degrees of freedom  
 (303 observations deleted due to missingness)  
 AIC: 14124

Number of Fisher Scoring iterations: 4

\$`0.11`

Call:

```
glm(formula = logit_model_std, family = binomial(link = "logit"),
    data = x)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.2923	0.4860	0.6171	0.7352	1.3728

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
--	----------	------------	---------	----------

```

(Intercept)      1.27858      0.02301    55.560    <2e-16 ***
scale(agediag)   -0.04952      0.02343    -2.114     0.0345 *
scale(sqrt(cd4))  0.23453      0.02248    10.433    <2e-16 ***
scale(ydiag)      0.43899      0.02247    19.540    <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 12734  on 11860  degrees of freedom
Residual deviance: 12217  on 11857  degrees of freedom
(303 observations deleted due to missingness)
AIC: 12225

Number of Fisher Scoring iterations: 4

```

### Down-sampled regressions

```

##- 1. down-sample: MEDIAN of each variable (with na.rm = T)

# head(li[[1]][, c("agediag", "sqrtcd4", "ydiag")])
down_median <- lapply(li, function(x)
  aggregate(x[, c("agediag", "sqrtcd4", "ydiag", "logv1")],
    list("size" = x$size),
    function(x) median(x, na.rm = TRUE)))

down_mean <- lapply(li, function(x)
  aggregate(x[, c("agediag", "sqrtcd4", "ydiag", "logv1")],
    list("size" = x$size),
    function(x) mean(x, na.rm = TRUE)))

# str(down_mean)
##- linear regression
lm_model_std = "scale(size) ~ scale(agediag) + scale(sqrtcd4) + scale(ydiag)"
lapply(down_median, function(x) summary(lm(lm_model_std, data = x)))

$`0.07`

Call:
lm(formula = lm_model_std, data = x)

Residuals:
    Min       1Q   Median       3Q      Max
-0.9520 -0.6592 -0.3118  0.3229  3.5106

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.131e-16  1.439e-01   0.000   1.000
scale(agediag) 9.038e-02  1.663e-01   0.544   0.589
scale(sqrtcd4) 6.494e-02  1.683e-01   0.386   0.701
scale(ydiag)   2.181e-02  1.480e-01   0.147   0.883

Residual standard error: 1.028 on 47 degrees of freedom
Multiple R-squared:  0.007271, Adjusted R-squared:  -0.05609
F-statistic: 0.1148 on 3 and 47 DF,  p-value: 0.951

$`0.11`

```

```

Call:
lm(formula = lm_model_std, data = x)

Residuals:
    Min       1Q   Median       3Q      Max
-1.0671 -0.3240 -0.1322  0.1768  6.1780

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.836e-15  1.347e-01   0.000   1.0000
scale(agediag) -3.660e-02  1.398e-01  -0.262   0.7946
scale(sqrtcd4)  5.980e-02  1.440e-01   0.415   0.6797
scale(ydiag)   -3.553e-01  1.469e-01  -2.418   0.0194 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9714 on 48 degrees of freedom
Multiple R-squared:  0.1119, Adjusted R-squared:  0.05639
F-statistic: 2.016 on 3 and 48 DF,  p-value: 0.1242

lapply(down_mean, function(x) summary(lm(lm_model_std, data = x)))

$`0.07`

Call:
lm(formula = lm_model_std, data = x)

Residuals:
    Min       1Q   Median       3Q      Max
-0.9289 -0.6309 -0.2532  0.2394  3.5106

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.206e-15  1.435e-01   0.000   1.000
scale(agediag) 7.281e-02  1.593e-01   0.457   0.650
scale(sqrtcd4) 5.084e-03  1.632e-01   0.031   0.975
scale(ydiag)   9.216e-02  1.489e-01   0.619   0.539

Residual standard error: 1.025 on 47 degrees of freedom
Multiple R-squared:  0.01317, Adjusted R-squared:  -0.04982
F-statistic: 0.209 on 3 and 47 DF,  p-value: 0.8896

$`0.11`

Call:
lm(formula = lm_model_std, data = x)

Residuals:
    Min       1Q   Median       3Q      Max
-1.0759 -0.3295 -0.1213  0.2026  6.1515

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.007e-15  1.347e-01   0.000   1.0000
scale(agediag) -3.380e-02  1.388e-01  -0.244   0.8086
scale(sqrtcd4)  3.820e-02  1.476e-01   0.259   0.7968
scale(ydiag)   -3.518e-01  1.482e-01  -2.374   0.0216 *

```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.9711 on 48 degrees of freedom
Multiple R-squared:  0.1124, Adjusted R-squared:  0.05693
F-statistic: 2.026 on 3 and 48 DF,  p-value: 0.1227
```

```
##- 2. plots
library(lattice)
# trellis.par.set(canonical.theme(color = FALSE))
for(i in 1:length(li)){
  print(histogram(~ sqrtcd4|factor(size),
    main = paste("distribution of sqrt(cd4) by cluster sizes at threshold =", names(li[i])),
    data = li[[i]])
)
}

plot(x = li[[1]]$size, y = li[[1]]$sqrtcd4)
```

