# homework 07

*Tingrui Huang*

*November 10, 2018*

## Data analysis

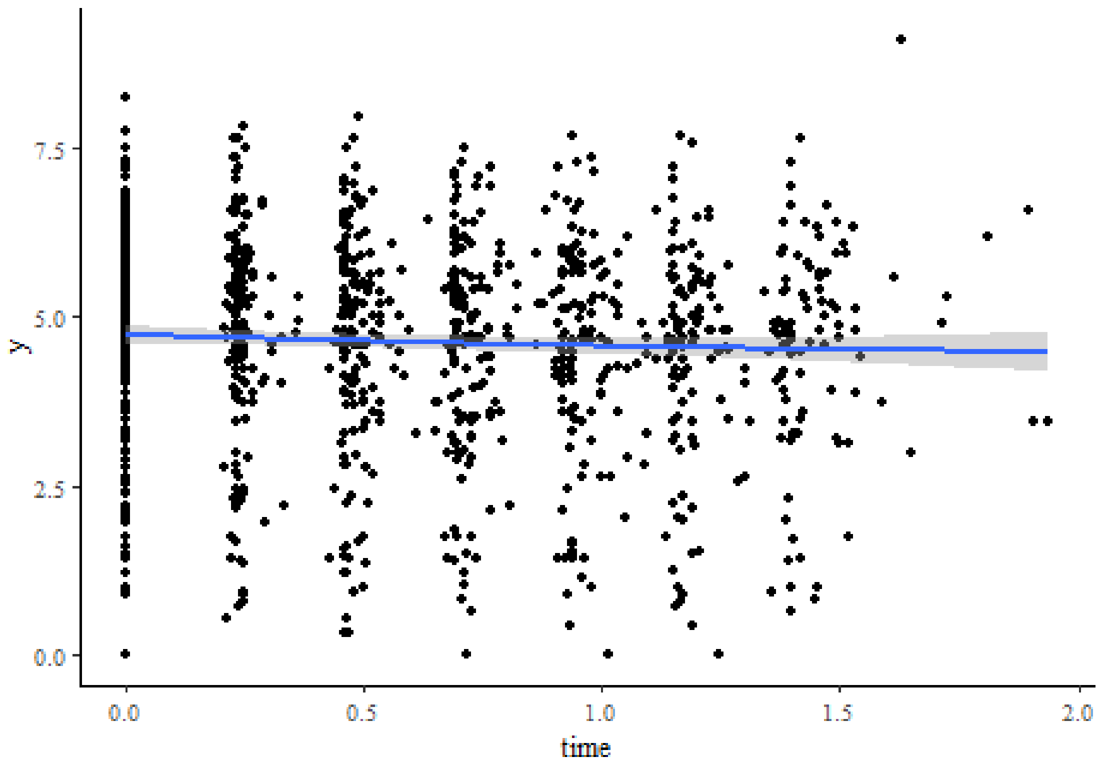
### CD4 percentages for HIV infected kids

The folder `cd4` has CD4 percentages for a set of young children with HIV who were measured several times over a period of two years. The dataset also includes the ages of the children at each measurement.

1. Graph the outcome (the CD4 percentage, on the square root scale) for each child as a function of time.

```r
ggplot(data = hiv.data, aes(x=time,y=y))+geom_point()+geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




2. Each child's data has a time course that can be summarized by a linear fit. Estimate these lines and plot them for all the children.

```r
# Build linear regression model - complete pooling
hiv_reg_np <- lm(y~time+factor(newpid)-1, data=hiv.data)
summary(hiv_reg_np)
```

```
##
## Call:
## lm(formula = y ~ time + factor(newpid) - 1, data = hiv.data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6595 -0.3293  0.0000  0.3347  4.0036
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## time               -0.38629    0.05455  -7.081 3.07e-12 ***
## factor(newpid)1     4.56368    0.34896  13.078  < 2e-16 ***
## factor(newpid)2     0.81507    0.54578   1.493 0.135716
## factor(newpid)3     5.95004    0.29534  20.146  < 2e-16 ***
## factor(newpid)4     5.61374    0.31677  17.722  < 2e-16 ***
## factor(newpid)5     4.00000    0.77180   5.183 2.76e-07 ***
## factor(newpid)6     5.36947    0.31738  16.918  < 2e-16 ***
## factor(newpid)7     5.61896    0.29436  19.088  < 2e-16 ***
## factor(newpid)8     5.14703    0.38791  13.268  < 2e-16 ***
## factor(newpid)9     6.21645    0.34732  17.898  < 2e-16 ***
## factor(newpid)10    5.71848    0.31739  18.017  < 2e-16 ***
## factor(newpid)11    2.44507    0.29417   8.312 3.89e-16 ***
## factor(newpid)12    4.36330    0.31699  13.765  < 2e-16 ***
## factor(newpid)13    5.33903    0.44635  11.962  < 2e-16 ***
## factor(newpid)14    3.00000    0.77180   3.887 0.000110 ***
## factor(newpid)15    5.24008    0.31759  16.499  < 2e-16 ***
## factor(newpid)16    2.39908    0.38705   6.198 9.03e-10 ***
## factor(newpid)17    6.10066    0.31839  19.161  < 2e-16 ***
## factor(newpid)18    6.02588    0.34608  17.412  < 2e-16 ***
## factor(newpid)19    4.10797    0.38783  10.592  < 2e-16 ***
## factor(newpid)20    5.00962    0.44580  11.237  < 2e-16 ***
## factor(newpid)21    5.00000    0.77180   6.478 1.60e-10 ***
## factor(newpid)22    6.16441    0.77180   7.987 4.66e-15 ***
## factor(newpid)23    1.59920    0.34723   4.606 4.76e-06 ***
## factor(newpid)24    4.81823    0.44728  10.772  < 2e-16 ***
## factor(newpid)25    4.76132    0.31717  15.012  < 2e-16 ***
## factor(newpid)26    4.63303    0.31656  14.636  < 2e-16 ***
## factor(newpid)27    4.38498    0.31672  13.845  < 2e-16 ***
## factor(newpid)28    5.65959    0.54590  10.367  < 2e-16 ***
## factor(newpid)29    4.52845    0.38717  11.696  < 2e-16 ***
## factor(newpid)30    1.00000    0.77180   1.296 0.195454
## factor(newpid)31    4.45824    0.54608   8.164 1.22e-15 ***
## factor(newpid)32    4.64821    0.34892  13.322  < 2e-16 ***
## factor(newpid)33    5.03494    0.29431  17.108  < 2e-16 ***
## factor(newpid)34    6.49167    0.54579  11.894  < 2e-16 ***
## factor(newpid)35    4.93661    0.38757  12.737  < 2e-16 ***
## factor(newpid)37    3.98526    0.54579   7.302 6.72e-13 ***
## factor(newpid)38    6.15939    0.44617  13.805  < 2e-16 ***
## factor(newpid)39    4.84721    0.34613  14.004  < 2e-16 ***
## factor(newpid)40    3.60555    0.77180   4.672 3.49e-06 ***
## factor(newpid)41    5.00000    0.77180   6.478 1.60e-10 ***
## factor(newpid)42    3.26132    0.29446  11.076  < 2e-16 ***
## factor(newpid)43    4.93493    0.29446  16.759  < 2e-16 ***
## factor(newpid)44    2.49104    0.44579   5.588 3.13e-08 ***
## factor(newpid)45    5.16288    0.31782  16.245  < 2e-16 ***
## factor(newpid)46    3.50085    0.31798  11.010  < 2e-16 ***
## factor(newpid)47    4.85968    0.31796  15.284  < 2e-16 ***
```

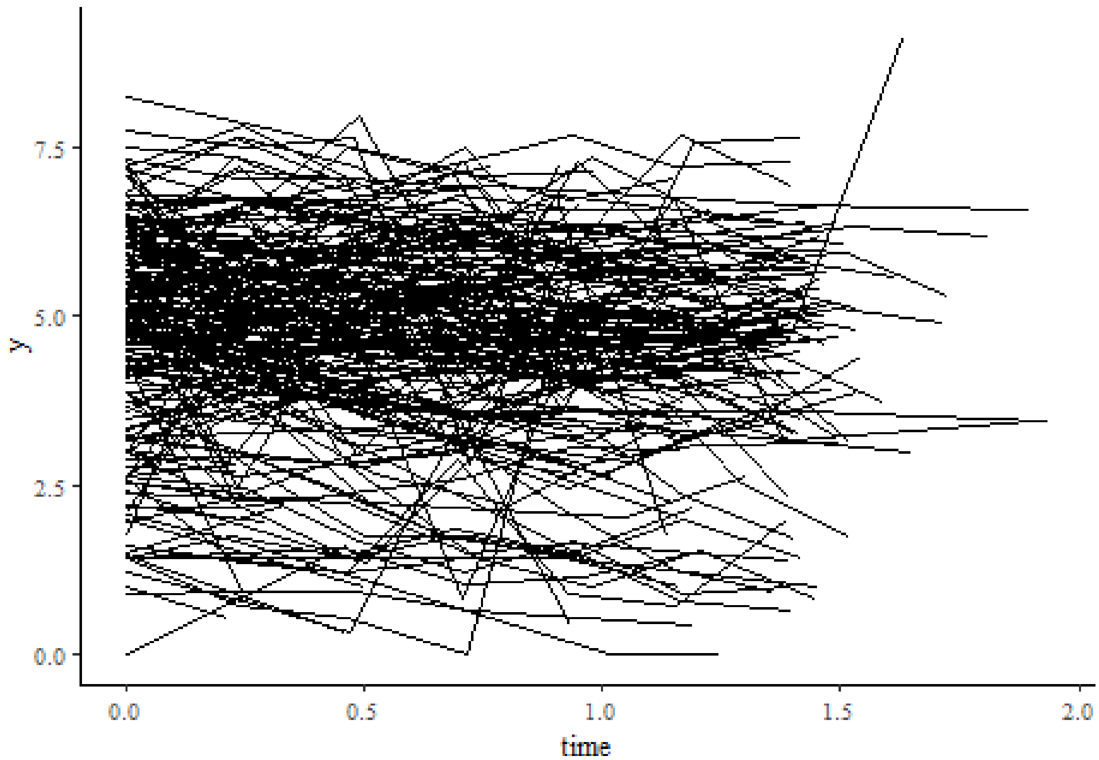
```
## factor(newpid)48   4.45407    0.38739  11.498  < 2e-16 ***
## factor(newpid)49   5.39827    0.29437  18.339  < 2e-16 ***
## factor(newpid)50   4.32745    0.29426  14.706  < 2e-16 ***
## factor(newpid)51   3.94551    0.34618  11.397  < 2e-16 ***
## factor(newpid)52   1.79719    0.29417   6.109 1.54e-09 ***
## factor(newpid)53   4.81554    0.29411  16.373  < 2e-16 ***
## factor(newpid)54   4.46903    0.29419  15.191  < 2e-16 ***
## factor(newpid)55   2.37752    0.29410   8.084 2.24e-15 ***
## factor(newpid)56   2.79201    0.54578   5.116 3.90e-07 ***
## factor(newpid)57   2.14991    0.31692   6.784 2.24e-11 ***
## factor(newpid)58   2.01600    0.31692   6.361 3.32e-10 ***
## factor(newpid)59   5.12724    0.29440  17.416  < 2e-16 ***
## factor(newpid)60   2.04462    0.54578   3.746 0.000192 ***
## factor(newpid)61   5.23903    0.31671  16.542  < 2e-16 ***
## factor(newpid)62   5.65826    0.29448  19.215  < 2e-16 ***
## factor(newpid)63   1.92512    0.29426   6.542 1.07e-10 ***
## factor(newpid)64   5.42219    0.29418  18.431  < 2e-16 ***
## factor(newpid)65   1.42126    0.34611   4.106 4.42e-05 ***
## factor(newpid)66   6.46556    0.44592  14.499  < 2e-16 ***
## factor(newpid)67   2.50677    0.54579   4.593 5.06e-06 ***
## factor(newpid)68   5.87367    0.77180   7.610 7.50e-14 ***
## factor(newpid)69   5.37708    0.39062  13.766  < 2e-16 ***
## factor(newpid)70   5.04789    0.38676  13.052  < 2e-16 ***
## factor(newpid)71   2.64575    0.77180   3.428 0.000638 ***
## factor(newpid)72   3.79504    0.38672   9.813  < 2e-16 ***
## factor(newpid)73   6.85565    0.77180   8.883  < 2e-16 ***
## factor(newpid)74   5.15287    0.29412  17.519  < 2e-16 ***
## factor(newpid)75   5.83766    0.29416  19.845  < 2e-16 ***
## factor(newpid)76   4.92242    0.34748  14.166  < 2e-16 ***
## factor(newpid)77   4.01660    0.38672  10.386  < 2e-16 ***
## factor(newpid)78   5.99278    0.29415  20.373  < 2e-16 ***
## factor(newpid)79   4.90326    0.44575  11.000  < 2e-16 ***
## factor(newpid)81   0.97153    0.54589   1.780 0.075492 .
## factor(newpid)82   3.25905    0.34636   9.409  < 2e-16 ***
## factor(newpid)83   0.94868    0.77180   1.229 0.219356
## factor(newpid)84   2.25870    0.34701   6.509 1.32e-10 ***
## factor(newpid)85   1.58969    0.34705   4.581 5.36e-06 ***
## factor(newpid)86   6.44121    0.34644  18.593  < 2e-16 ***
## factor(newpid)87   6.09731    0.29421  20.724  < 2e-16 ***
## factor(newpid)88   4.83296    0.54579   8.855  < 2e-16 ***
## factor(newpid)89   5.02052    0.34621  14.501  < 2e-16 ***
## factor(newpid)90   5.84808    0.77180   7.577 9.53e-14 ***
## factor(newpid)91   2.54897    0.38706   6.586 8.09e-11 ***
## factor(newpid)92   2.68623    0.54579   4.922 1.04e-06 ***
## factor(newpid)93   1.52443    0.38637   3.945 8.64e-05 ***
## factor(newpid)94   4.94328    0.44775  11.040  < 2e-16 ***
## factor(newpid)95   2.78151    0.54578   5.096 4.30e-07 ***
## factor(newpid)96   4.89898    0.77180   6.347 3.62e-10 ***
## factor(newpid)97   7.70878    0.44671  17.257  < 2e-16 ***
## factor(newpid)98   4.79583    0.77180   6.214 8.22e-10 ***
## factor(newpid)99   6.58753    0.38674  17.033  < 2e-16 ***
## factor(newpid)100  6.54584    0.34609  18.914  < 2e-16 ***
## factor(newpid)101  5.65685    0.77180   7.329 5.54e-13 ***
## factor(newpid)103  6.11117    0.29512  20.708  < 2e-16 ***
```

```
## factor(newpid)104  3.55877    0.31688  11.230  < 2e-16 ***
## factor(newpid)105  4.66845    0.29461  15.846  < 2e-16 ***
## factor(newpid)106  3.79964    0.38686   9.822  < 2e-16 ***
## factor(newpid)107  5.79041    0.38686  14.968  < 2e-16 ***
## factor(newpid)108  1.17737    0.38739   3.039 0.002447 **
## factor(newpid)109  4.04447    0.54579   7.410 3.13e-13 ***
## factor(newpid)110  5.32304    0.29448  18.076  < 2e-16 ***
## factor(newpid)111  2.13749    0.54580   3.916 9.74e-05 ***
## factor(newpid)112  4.04681    0.29465  13.734  < 2e-16 ***
## factor(newpid)113  6.34488    0.31739  19.991  < 2e-16 ***
## factor(newpid)114  4.95064    0.29459  16.805  < 2e-16 ***
## factor(newpid)115  5.62952    0.29454  19.113  < 2e-16 ***
## factor(newpid)116  4.25683    0.54612   7.795 1.95e-14 ***
## factor(newpid)117  4.41240    0.34852  12.660  < 2e-16 ***
## factor(newpid)118  5.31355    0.34636  15.341  < 2e-16 ***
## factor(newpid)119  1.92914    0.54582   3.534 0.000432 ***
## factor(newpid)120  6.83535    0.31712  21.555  < 2e-16 ***
## factor(newpid)121  6.12904    0.44703  13.711  < 2e-16 ***
## factor(newpid)122  5.43379    0.44651  12.169  < 2e-16 ***
## factor(newpid)123  2.96695    0.54578   5.436 7.18e-08 ***
## factor(newpid)124  3.16228    0.77180   4.097 4.60e-05 ***
## factor(newpid)126  4.48243    0.38753  11.567  < 2e-16 ***
## factor(newpid)127  5.25547    0.34628  15.177  < 2e-16 ***
## factor(newpid)128  4.75350    0.54668   8.695  < 2e-16 ***
## factor(newpid)129  0.97864    0.34636   2.825 0.004836 **
## factor(newpid)130  3.70472    0.38672   9.580  < 2e-16 ***
## factor(newpid)131  4.25708    0.38711  10.997  < 2e-16 ***
## factor(newpid)132  4.73853    0.38778  12.220  < 2e-16 ***
## factor(newpid)133  3.77490    0.31673  11.918  < 2e-16 ***
## factor(newpid)134  6.72519    0.29422  22.858  < 2e-16 ***
## factor(newpid)135  5.60776    0.29440  19.048  < 2e-16 ***
## factor(newpid)136  6.64977    0.29433  22.593  < 2e-16 ***
## factor(newpid)137  5.67273    0.29452  19.261  < 2e-16 ***
## factor(newpid)138  7.48331    0.77180   9.696  < 2e-16 ***
## factor(newpid)139  4.85189    0.29479  16.459  < 2e-16 ***
## factor(newpid)140  5.47249    0.29452  18.581  < 2e-16 ***
## factor(newpid)141  7.16773    0.29440  24.347  < 2e-16 ***
## factor(newpid)142  2.82420    0.31707   8.907  < 2e-16 ***
## factor(newpid)143  2.88106    0.29437   9.787  < 2e-16 ***
## factor(newpid)144  6.04833    0.29423  20.556  < 2e-16 ***
## factor(newpid)145  5.55106    0.31688  17.518  < 2e-16 ***
## factor(newpid)146  5.46320    0.31677  17.246  < 2e-16 ***
## factor(newpid)147  6.18166    0.34655  17.838  < 2e-16 ***
## factor(newpid)148  5.34407    0.44578  11.988  < 2e-16 ***
## factor(newpid)149  5.67007    0.34615  16.381  < 2e-16 ***
## factor(newpid)150  4.39422    0.38642  11.372  < 2e-16 ***
## factor(newpid)151  5.68779    0.38640  14.720  < 2e-16 ***
## factor(newpid)152  4.61519    0.77180   5.980 3.33e-09 ***
## factor(newpid)153  7.21403    0.44577  16.183  < 2e-16 ***
## factor(newpid)154  5.71394    0.44580  12.817  < 2e-16 ***
## factor(newpid)155  6.27073    0.44579  14.067  < 2e-16 ***
## factor(newpid)156  6.34439    0.54578  11.624  < 2e-16 ***
## factor(newpid)157  6.41098    0.44609  14.371  < 2e-16 ***
## factor(newpid)158  6.08632    0.34692  17.544  < 2e-16 ***
```

```
## factor(newpid)159  5.29916    0.54594   9.706  < 2e-16 ***
## factor(newpid)160  5.04712    0.54579   9.247  < 2e-16 ***
## factor(newpid)161  5.14072    0.38657  13.298  < 2e-16 ***
## factor(newpid)162  4.69277    0.44588  10.525  < 2e-16 ***
## factor(newpid)163  7.42011    0.38647  19.200  < 2e-16 ***
## factor(newpid)164  7.07418    0.34873  20.286  < 2e-16 ***
## factor(newpid)165  4.40042    0.34744  12.665  < 2e-16 ***
## factor(newpid)166  5.63845    0.54812  10.287  < 2e-16 ***
## factor(newpid)167  4.93276    0.38713  12.742  < 2e-16 ***
## factor(newpid)168  5.79989    0.29425  19.711  < 2e-16 ***
## factor(newpid)169  2.83271    0.54605   5.188 2.69e-07 ***
## factor(newpid)170  4.52041    0.34670  13.039  < 2e-16 ***
## factor(newpid)171  6.70820    0.77180   8.692  < 2e-16 ***
## factor(newpid)172  5.26891    0.34643  15.209  < 2e-16 ***
## factor(newpid)173  1.59625    0.54592   2.924 0.003551 **
## factor(newpid)174  3.80765    0.34709  10.970  < 2e-16 ***
## factor(newpid)175  5.86770    0.34640  16.939  < 2e-16 ***
## factor(newpid)176  5.71388    0.44591  12.814  < 2e-16 ***
## factor(newpid)177  4.65448    0.38715  12.022  < 2e-16 ***
## factor(newpid)178  6.64100    0.34712  19.132  < 2e-16 ***
## factor(newpid)179  5.42868    0.44577  12.178  < 2e-16 ***
## factor(newpid)180  5.38254    0.29417  18.297  < 2e-16 ***
## factor(newpid)181  7.58231    0.31737  23.891  < 2e-16 ***
## factor(newpid)182  6.87445    0.44674  15.388  < 2e-16 ***
## factor(newpid)183  4.73226    0.54591   8.669  < 2e-16 ***
## factor(newpid)184  4.69042    0.77180   6.077 1.87e-09 ***
## factor(newpid)185  5.32106    0.31790  16.738  < 2e-16 ***
## factor(newpid)186  2.26637    0.34754   6.521 1.22e-10 ***
## factor(newpid)187  5.96108    0.31804  18.743  < 2e-16 ***
## factor(newpid)188  5.64729    0.34676  16.286  < 2e-16 ***
## factor(newpid)189  0.89556    0.54589   1.641 0.101277
## factor(newpid)190  3.93221    0.54593   7.203 1.34e-12 ***
## factor(newpid)191  4.73072    0.44582  10.611  < 2e-16 ***
## factor(newpid)192  4.63493    0.29415  15.757  < 2e-16 ***
## factor(newpid)193  3.51569    0.29414  11.952  < 2e-16 ***
## factor(newpid)194  1.67399    0.31665   5.286 1.60e-07 ***
## factor(newpid)195  6.57259    0.44708  14.701  < 2e-16 ***
## factor(newpid)196  4.28686    0.38778  11.055  < 2e-16 ***
## factor(newpid)197  4.52015    0.38659  11.692  < 2e-16 ***
## factor(newpid)198  6.11686    0.34677  17.640  < 2e-16 ***
## factor(newpid)199  3.58154    0.38734   9.247  < 2e-16 ***
## factor(newpid)200  6.33062    0.31871  19.863  < 2e-16 ***
## factor(newpid)201  4.88817    0.38837  12.586  < 2e-16 ***
## factor(newpid)202  6.08433    0.54598  11.144  < 2e-16 ***
## factor(newpid)203  6.31594    0.38792  16.282  < 2e-16 ***
## factor(newpid)204  5.44066    0.38672  14.069  < 2e-16 ***
## factor(newpid)205  3.66210    0.34771  10.532  < 2e-16 ***
## factor(newpid)206  5.98915    0.29415  20.361  < 2e-16 ***
## factor(newpid)207  6.08204    0.31761  19.149  < 2e-16 ***
## factor(newpid)208  4.17020    0.34723  12.010  < 2e-16 ***
## factor(newpid)209  6.43027    0.31684  20.295  < 2e-16 ***
## factor(newpid)210  5.21148    0.29412  17.719  < 2e-16 ***
## factor(newpid)211  5.34459    0.29419  18.167  < 2e-16 ***
## factor(newpid)212  5.21535    0.31670  16.468  < 2e-16 ***
```

```
## factor(newpid)213  4.67607    0.44578  10.490  < 2e-16 ***
## factor(newpid)214  6.54179    0.29428  22.230  < 2e-16 ***
## factor(newpid)215  5.04463    0.31666  15.931  < 2e-16 ***
## factor(newpid)216  3.74901    0.34628  10.827  < 2e-16 ***
## factor(newpid)217  3.09943    0.54578   5.679 1.88e-08 ***
## factor(newpid)218  4.76821    0.29420  16.207  < 2e-16 ***
## factor(newpid)219  5.47723    0.77180   7.097 2.76e-12 ***
## factor(newpid)220  6.34478    0.29424  21.564  < 2e-16 ***
## factor(newpid)221  5.78464    0.31662  18.270  < 2e-16 ***
## factor(newpid)222  5.27235    0.31785  16.587  < 2e-16 ***
## factor(newpid)223  5.34864    0.31661  16.894  < 2e-16 ***
## factor(newpid)224  3.80821    0.54578   6.978 6.19e-12 ***
## factor(newpid)225  6.47400    0.29413  22.010  < 2e-16 ***
## factor(newpid)226  6.85178    0.34695  19.748  < 2e-16 ***
## factor(newpid)227  6.21616    0.31664  19.631  < 2e-16 ***
## factor(newpid)228  4.67312    0.31665  14.758  < 2e-16 ***
## factor(newpid)229  5.25787    0.34628  15.184  < 2e-16 ***
## factor(newpid)230  5.96217    0.34628  17.218  < 2e-16 ***
## factor(newpid)231  5.95432    0.38653  15.405  < 2e-16 ***
## factor(newpid)232  6.17519    0.44620  13.840  < 2e-16 ***
## factor(newpid)233  4.36377    0.38636  11.295  < 2e-16 ***
## factor(newpid)234  6.22240    0.54578  11.401  < 2e-16 ***
## factor(newpid)235  3.21066    0.44635   7.193 1.43e-12 ***
## factor(newpid)236  2.83698    0.34674   8.182 1.06e-15 ***
## factor(newpid)237  5.43365    0.31707  17.137  < 2e-16 ***
## factor(newpid)238  5.05647    0.38660  13.079  < 2e-16 ***
## factor(newpid)239  5.54035    0.44593  12.424  < 2e-16 ***
## factor(newpid)240  3.51138    0.34603  10.148  < 2e-16 ***
## factor(newpid)241  6.11555    0.77180   7.924 7.49e-15 ***
## factor(newpid)242  5.16910    0.44592  11.592  < 2e-16 ***
## factor(newpid)243  5.89800    0.44636  13.213  < 2e-16 ***
## factor(newpid)244  5.94175    0.54578  10.887  < 2e-16 ***
## factor(newpid)245  4.92484    0.38641  12.745  < 2e-16 ***
## factor(newpid)246  5.05558    0.54579   9.263  < 2e-16 ***
## factor(newpid)247  4.78539    0.77180   6.200 8.92e-10 ***
## factor(newpid)248  5.64132    0.54579  10.336  < 2e-16 ***
## factor(newpid)249  5.59464    0.77180   7.249 9.71e-13 ***
## factor(newpid)250  5.83524    0.54579  10.691  < 2e-16 ***
## factor(newpid)251  3.74166    0.77180   4.848 1.49e-06 ***
## factor(newpid)252  4.51291    0.54582   8.268 5.45e-16 ***
## factor(newpid)253  3.60555    0.77180   4.672 3.49e-06 ***
## factor(newpid)254  3.75520    0.54598   6.878 1.20e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7718 on 821 degrees of freedom
## Multiple R-squared:  0.9809, Adjusted R-squared:  0.9751
## F-statistic: 168.1 on 251 and 821 DF,  p-value: < 2.2e-16
```

```r
# Plot each child
ggplot(hiv.data, aes(x=time,y=y,group=newpid))+geom_line()
```




3. Set up a model for the children's slopes and intercepts as a function of the treatment and age at baseline. Estimate this model using the two-step procedure first estimate the intercept and slope separately for each child, then fit the between-child models using the point estimates from the first step.

```
# Create matrix to store coefficients
np_hiv_coef <- matrix(NA, nrow = 254, ncol = 3)
colnames(np_hiv_coef) <- c("newpid","intercept","slope")
# Insert value into the matrix
for (i in unique(hiv.data$newpid)) {
  cp <- lm(y~time, data = hiv.data[newpid==i,])
  np_hiv_coef[i,1] <- i
  np_hiv_coef[i,2] <- coef(cp)[1]
  np_hiv_coef[i,3] <- coef(cp)[2]
}
# Merge two matrix
treat_age <- hiv.data[,list(age.baseline=unique(age.baseline),treatment=unique(treatment)), by=newpid]
mergetwo <- merge(np_hiv_coef,treat_age,by="newpid")
# Regress intercept and slope
lm(intercept~ age.baseline+factor(treatment),data = mergetwo)
```

```
##
## Call:
## lm(formula = intercept ~ age.baseline + factor(treatment), data = mergetwo)
##
## Coefficients:
##       (Intercept)       age.baseline  factor(treatment)2
##            5.1179            -0.1210              0.1236
```

```r
lm(slope~ age.baseline+factor(treatment),data=mergetwo)
```

```
##
## Call:
## lm(formula = slope ~ age.baseline + factor(treatment), data = mergetwo)
##
## Coefficients:
##       (Intercept)        age.baseline  factor(treatment)2
##          -0.26568            -0.04223            -0.13926
```

4. Write a model predicting CD4 percentage as a function of time with varying intercepts across children. Fit using `lmer()` and interpret the coefficient for time.

```r
hiv_reg_vi <- lmer(y~time+(1|newpid), data = hiv.data)
summary(hiv_reg_vi)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ time + (1 | newpid)
##    Data: hiv.data
##
## REML criterion at convergence: 3140.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7379 -0.4379  0.0024  0.4324  5.0017
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  newpid   (Intercept) 1.9569   1.3989
##  Residual             0.5968   0.7725
## Number of obs: 1072, groups:  newpid, 250
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  4.76341    0.09648  49.372
## time        -0.36609    0.05399  -6.781
##
## Correlation of Fixed Effects:
##      (Intr)
## time -0.278
```

```r
head(ranef(hiv_reg_vi)$newpid)
```

```
##   (Intercept)
## 1  -0.2061589
## 2  -3.4278427
## 3   1.1207203
## 4   0.7977213
## 5  -0.5850113
## 6   0.5633424
```

Based on the result table, we have regression model: $y = 4.76 - 0.37time$

As time goes on, the CD4 percentage will be decrease. Meanwhile, different child will have different CD4 percentage at each time period, since there are random effects among children.

When calculating CD4 for each child, we need to add the random effects at the end of the model, for example, the model for the first child will be The first child: $y = 4.76 - 0.37time - 0.2$, where -0.2 is the random effect.

5. Extend the model in (4) to include child-level predictors (that is, group-level predictors) for treatment and age at baseline. Fit using `lmer()` and interpret the coefficients on time, treatment, and age at baseline.

```
hiv_reg_vis <- lmer(y~time+factor(treatment)+age.baseline+(1|newpid), data = hiv.data)
summary(hiv_reg_vis)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ time + factor(treatment) + age.baseline + (1 | newpid)
##    Data: hiv.data
##
## REML criterion at convergence: 3137.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7490 -0.4392  0.0097  0.4282  5.0141
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  newpid   (Intercept) 1.8897   1.3747
##  Residual             0.5969   0.7726
## Number of obs: 1072, groups:  newpid, 250
##
## Fixed effects:
##                    Estimate Std. Error t value
## (Intercept)         5.08614    0.18793  27.064
## time               -0.36216    0.05399  -6.708
## factor(treatment)2  0.18008    0.18262   0.986
## age.baseline       -0.11945    0.04000  -2.986
##
## Correlation of Fixed Effects:
##             (Intr) time   fct()2
## time        -0.135
## fctr(trtm)2 -0.462  0.010
## age.baselin -0.727 -0.017 -0.003
```

```
head(ranef(hiv_reg_vis)$newpid)
```

```
##   (Intercept)
## 1 -0.07346121
## 2 -3.47851396
## 3  1.50703667
## 4  0.74880700
## 5 -0.76603523
## 6  0.41326719
```

Based on the result table, we have the regression model: $y = 4.91 - 0.36time + 0.18treatment - 0.12age.baseline$

Time and age have negative effects on CD4 while treatment has positive effetcs.

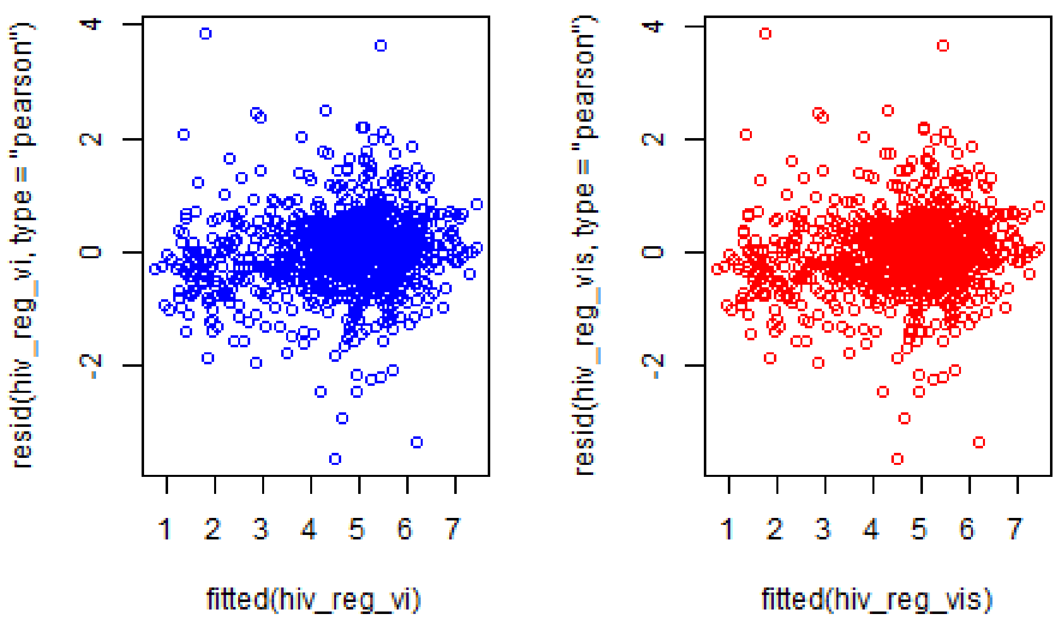
When calculating CD4 for each child, we need to add the random effects at the end of the model, for example, the model for the first child will be The first child: $y = 4.91 - 0.36time + 0.18treatment - 0.12age.baseline - 0.07$, where -0.07 is the random effect.

6. Investigate the change in partial pooling from (4) to (5) both graphically and numerically.

```r
anova(hiv_reg_vi,hiv_reg_vis)
```

```
## refitting model(s) with ML (instead of REML)

## Data: hiv.data
## Models:
## hiv_reg_vi: y ~ time + (1 | newpid)
## hiv_reg_vis: y ~ time + factor(treatment) + age.baseline + (1 | newpid)
##             Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## hiv_reg_vi   4 3141.9 3161.8 -1566.9   3133.9
## hiv_reg_vis  6 3136.1 3165.9 -1562.0   3124.1 9.7956      2   0.007463 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
par(mfrow=c(1,2))
plot(fitted(hiv_reg_vi),resid(hiv_reg_vi,type="pearson"),col="blue")
plot(fitted(hiv_reg_vis),resid(hiv_reg_vis,type="pearson"),col="red")
```




The model in (5) has a slightly better AIC and edviance.

7. Use the model fit from (5) to generate simulation of predicted CD4 percentages for each child in the dataset at a hypothetical next time point.

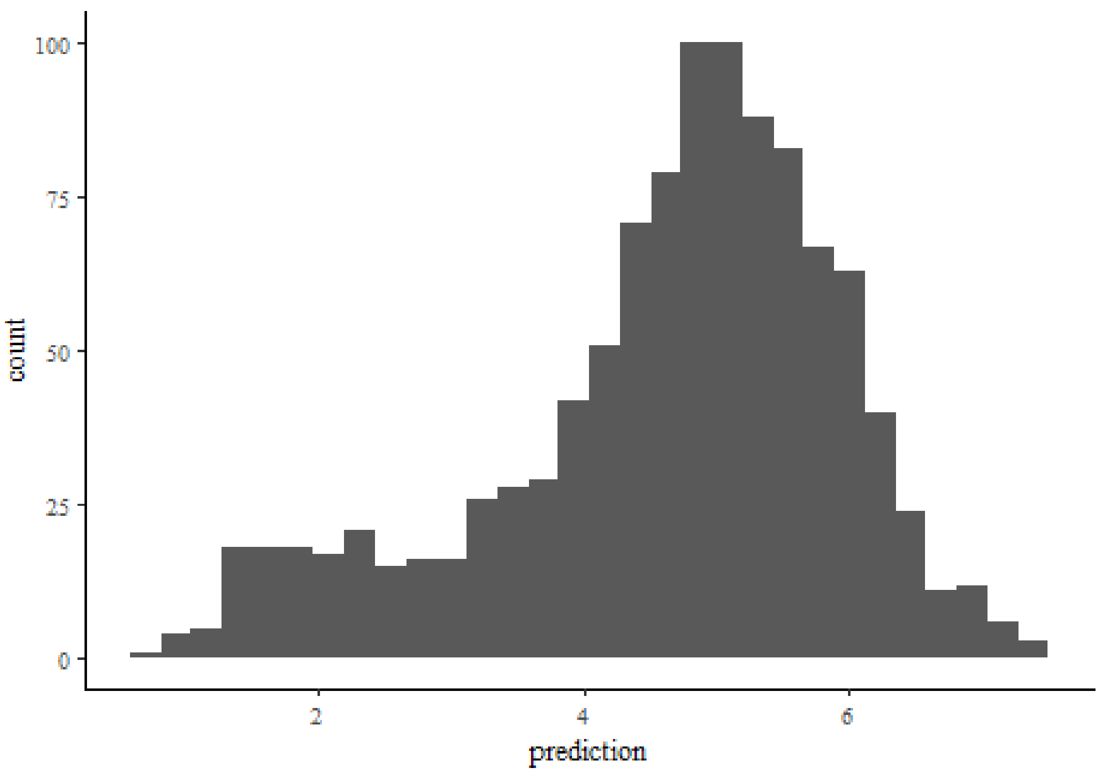
```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

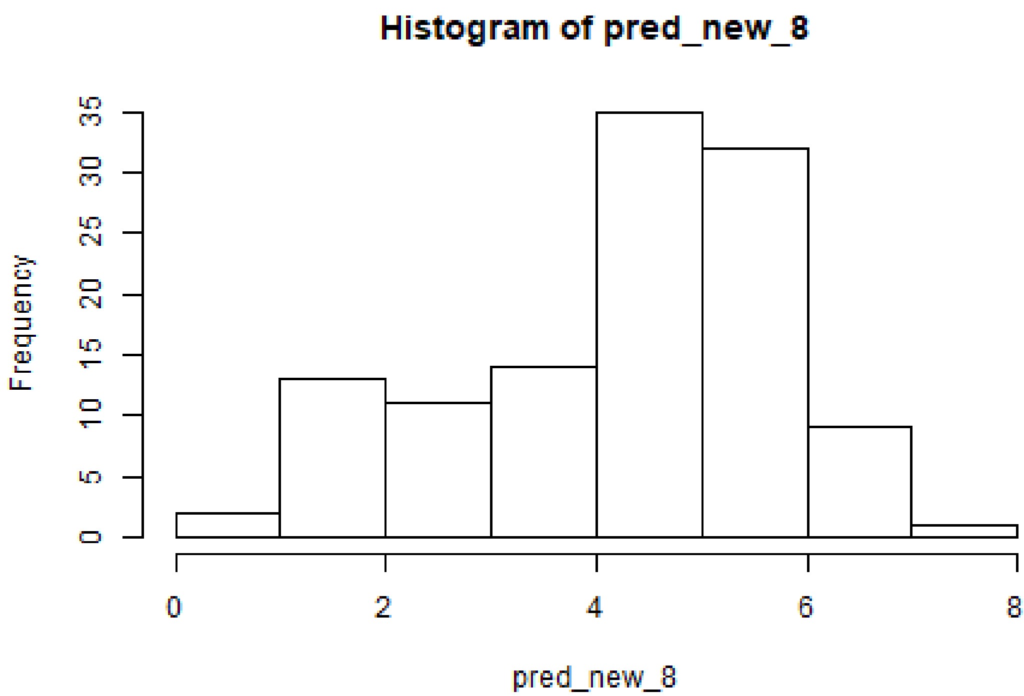
```
predict_data <- subset(hiv.data, !is.na(hiv.data$treatment) & !is.na(age.baseline))
predict_new <- predict(hiv_reg_vis,newdata=predict_data)
predict_cmb <- cbind(predict_new,predict_data)
colnames(predict_cmb)[1] <- c("prediction")
ggplot(predict_cmb,aes(x=prediction))+geom_histogram()
```

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



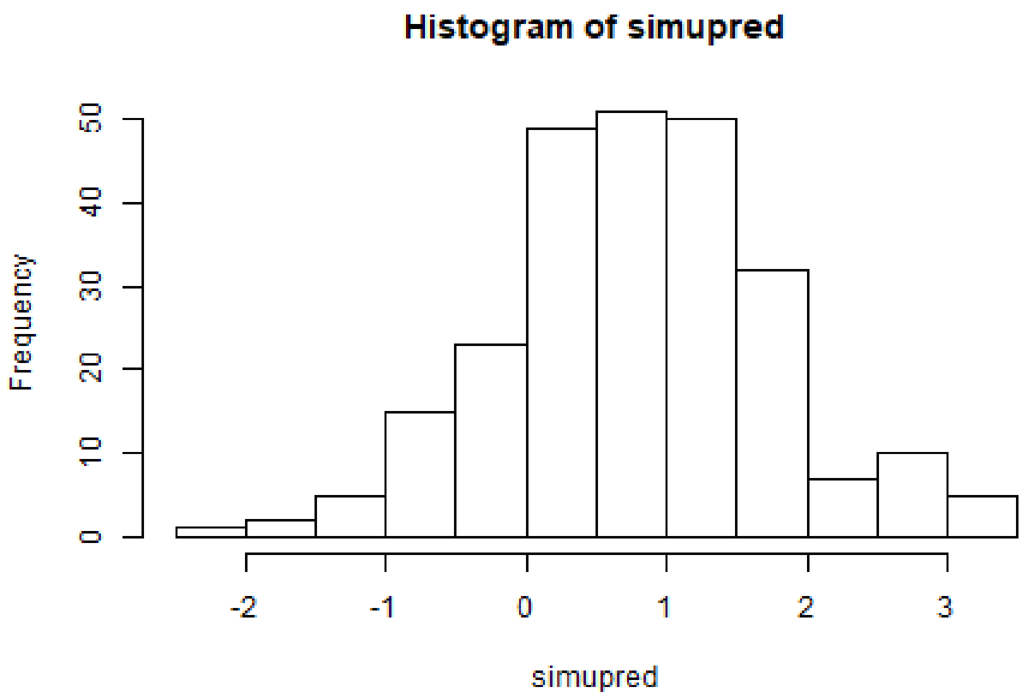

8. Use the same model fit to generate simulations of CD4 percentages at each of the time periods for a new child who was 4 years old at baseline.

```
pred_data_2 <- subset(hiv.data, !is.na(hiv.data$treatment) & !is.na(age.baseline))
pred_data_2 <- pred_data_2[, -c(1, 4, 5, 6, 8)]
pred_data_2 <- pred_data_2[which(round(pred_data_2$age.baseline) == 4 ),]
pred_new_8 <- predict(hiv_reg_vis, newdata=pred_data_2)
hist(pred_new_8)
```




9. Posterior predictive checking: continuing the previous exercise, use the fitted model from (5) to simulate a new dataset of CD4 percentages (with the same sample size and ages of the original dataset) for the final time point of the study, and record the average CD4 percentage in this sample. Repeat this process 1000 times and compare the simulated distribution to the observed CD4 percentage at the final time point for the actual data.

```
# Using model - hiv_reg_vis from (5)
pred_new_9 <- hiv.data[,list(time=max(time),age.baseline=unique(age.baseline),
                        treatment=unique(treatment)),by =newpid]
cm3<-coef(hiv_reg_vis)$newpid
sigy<-sigma.hat(hiv_reg_vis)$sigma$data
predy<-cm3[,1]+cm3[,2]*pred_new_9$time+cm3[,3]*pred_new_9$age.baseline+cm3[,4]*(pred_new_9$treatment-1)
avg.pred.CD4PCT<-NULL
simupred<-matrix(NA,nrow(pred_new_9),1000)
for (i in 1:1000){
  ytilde<-rnorm(predy,sigy)
  simupred[,1]<-ytilde
}
hist(simupred)
```




10. Extend the model to allow for varying slopes for the time predictor.

```r
# Assume random slope and intercept are correlated
hiv_reg_vslope <- lmer(y~time+factor(treatment)+age.baseline+(1+time|newpid), data = hiv.data)
summary(hiv_reg_vslope)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ time + factor(treatment) + age.baseline + (1 + time | newpid)
##    Data: hiv.data
##
## REML criterion at convergence: 3107
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0998 -0.4057  0.0174  0.4030  5.0157
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  newpid   (Intercept) 1.8464   1.3588
##           time        0.3374   0.5808   -0.04
##  Residual             0.5145   0.7173
## Number of obs: 1072, groups:  newpid, 250
##
## Fixed effects:
##                      Estimate Std. Error t value
## (Intercept)           5.10850    0.18594  27.474
## time                 -0.35258    0.06763  -5.214
## factor(treatment)2    0.15952    0.18137   0.880
## age.baseline         -0.12423    0.03971  -3.128
##
```

```
## Correlation of Fixed Effects:
##              (Intr) time   fct()2
## time         -0.114
## fctr(trtm)2 -0.463  0.010
## age.baselin -0.729 -0.013 -0.004
```

11. Next fit a model that does not allow for varying slopes but does allow for different coefficients for each time point (rather than fitting the linear trend).

```
hiv_reg_11 <- lmer(y~factor(time)+(1|newpid), data = hiv.data)
```

Since I factorized the time, there are lots of levels of time in the outcome table.

12. Compare the results of these models both numerically and graphically.

```
anova(hiv_reg_11,hiv_reg_vslope,hiv_reg_vis,hiv_reg_vi)
```

```
## refitting model(s) with ML (instead of REML)

## Data: hiv.data
## Models:
## hiv_reg_vi: y ~ time + (1 | newpid)
## hiv_reg_vis: y ~ time + factor(treatment) + age.baseline + (1 | newpid)
## hiv_reg_vslope: y ~ time + factor(treatment) + age.baseline + (1 + time | newpid)
## hiv_reg_11: y ~ factor(time) + (1 | newpid)
##                 Df    AIC    BIC  logLik deviance    Chisq Chi Df
## hiv_reg_vi       4 3141.9 3161.8 -1566.9   3133.9
## hiv_reg_vis      6 3136.1 3165.9 -1562.0   3124.1   9.7956      2
## hiv_reg_vslope   8 3110.3 3150.1 -1547.1   3094.3  29.7893      2
## hiv_reg_11     405 3244.5 5260.3 -1217.3   2434.5 659.7525    397
##                Pr(>Chisq)
## hiv_reg_vi
## hiv_reg_vis      0.007463 **
## hiv_reg_vslope  3.399e-07 ***
## hiv_reg_11      2.261e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The AIC an deviance of each model are pretty close, however, the varying slope model has the best AIC and lowest deviance.

## Figure skate in the 1932 Winter Olympics

The folder olympics has seven judges' ratings of seven figure skaters (on two criteria: "technical merit" and "artistic impression") from the 1932 Winter Olympics. Take a look at http://www.stat.columbia.edu/~gelman/arm/examples/olympics/olympics1932.txt

1. Construct a $7 \times 7 \times 2$ array of the data (ordered by skater, judge, and judging criterion).

```
performance <- olympics1932 %>% filter(criterion=="Performance")
program <- olympics1932 %>% filter(criterion=="Program")
```

2. Reformulate the data as a $49 \times 4$ array (similar to the top table in Figure 11.7), where the first two columns are the technical merit and artistic impression scores, the third column is a skater ID, and the fourth column is a judge ID.

```
new_olympics <- matrix(NA, nrow = 49, ncol = 4)
colnames(new_olympics) <- c("pair","judge","performance","program")
```

```
new_olympics[,1] <- c(rep(1,7),rep(2,7),rep(3,7),rep(4,7),rep(5,7),rep(6,7),rep(7,7))
new_olympics[,2] <- rep(c("judge_1","judge_2","judge_3","judge_4","judge_5","judge_6","judge_7"),7)
p_score <- as.vector(t(performance[,3:9]))
pro_score <- as.vector(t(program[,3:9]))
new_olympics[,3] <- p_score
new_olympics[,4] <- pro_score
new_olympics <- data.frame(new_olympics)
```

3. Add another column to this matrix representing an indicator variable that equals 1 if the skater and judge are from the same country, or 0 otherwise.

```
new_olympics2 <- new_olympics %>% mutate(samecountry=rep(0,49))
new_olympics2[5,5] <- 1
new_olympics2[14,5] <- 1
new_olympics2[15,5] <- 1
new_olympics2[22,5] <- 1
new_olympics2[49,5] <- 1
```

4. Write the notation for a non-nested multilevel model (varying across skaters and judges) for the technical merit ratings and fit using lmer().

```
techmer <- lmer(as.numeric(program)~1+(1|pair)+(1|judge), data = new_olympics2)
summary(techmer)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(program) ~ 1 + (1 | pair) + (1 | judge)
##    Data: new_olympics2
##
## REML criterion at convergence: 255.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.03112 -0.65091 -0.08745  0.53846  1.94730
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  pair     (Intercept) 9.009    3.002
##  judge    (Intercept) 4.880    2.209
##  Residual             6.528    2.555
## Number of obs: 49, groups:  pair, 7; judge, 7
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    9.163      1.455   6.297
```

5. Fit the model in (4) using the artistic impression ratings.

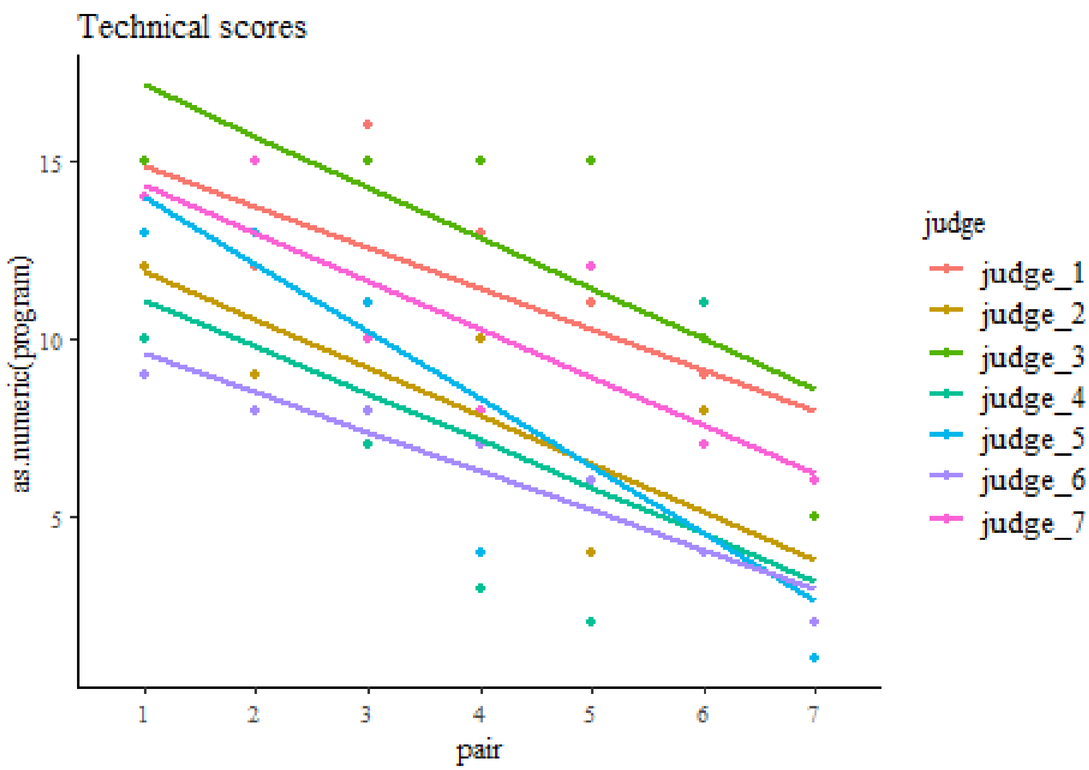
```
artimp <- lmer(as.numeric(performance)~1+(1|pair)+(1|judge), data = new_olympics2)
summary(artimp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(performance) ~ 1 + (1 | pair) + (1 | judge)
##    Data: new_olympics2
##
## REML criterion at convergence: 250.5
##
```

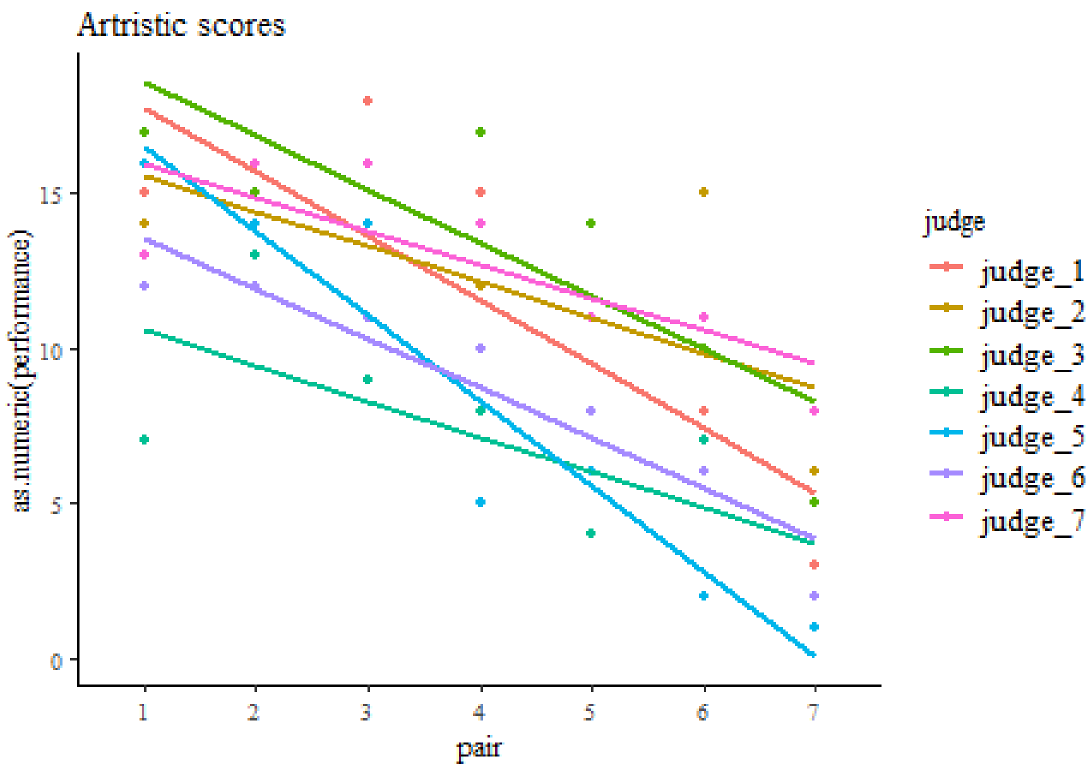
```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9711 -0.5224 -0.1428  0.4735  2.2102
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  pair     (Intercept) 14.074   3.751
##  judge    (Intercept)  5.353   2.314
##  Residual              5.294   2.301
## Number of obs: 49, groups:  pair, 7; judge, 7
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   10.571      1.698   6.226
```

6. Display your results for both outcomes graphically.

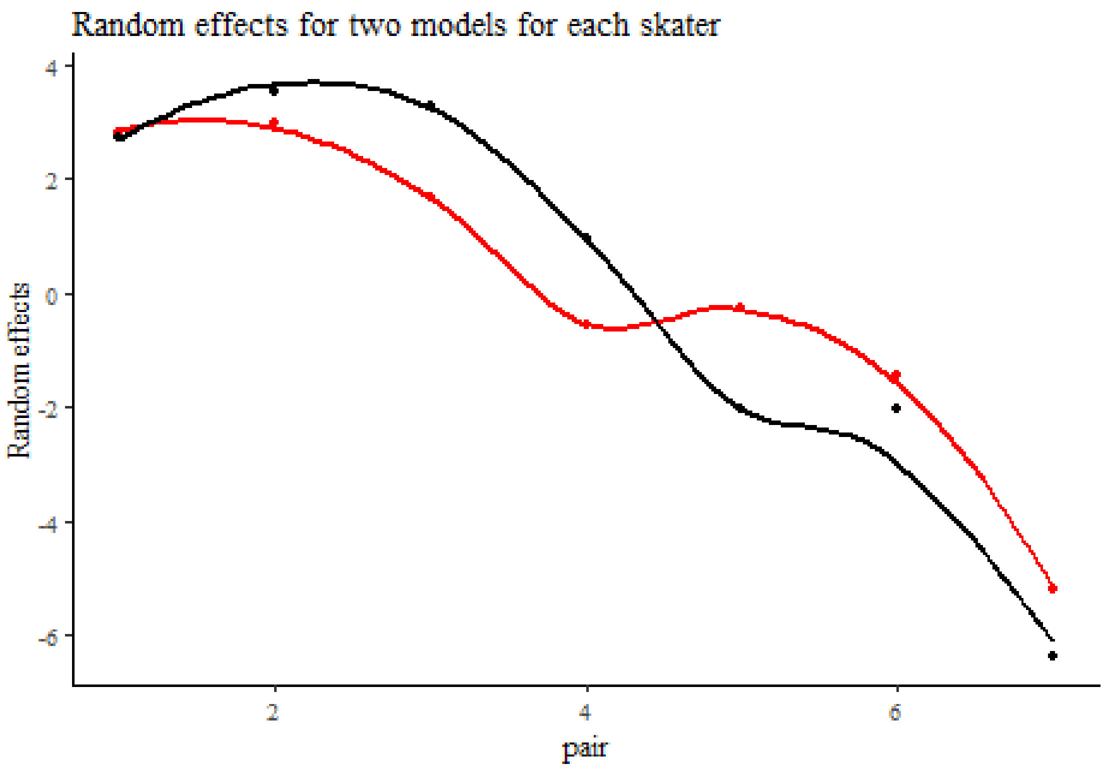
```
# Plot on raw data
ggplot(new_olympics2,aes(x=pair,y=as.numeric(program),group=judge,color=judge))+
  geom_point()+geom_smooth(method = "lm", se= FALSE)+ggtitle("Technical scores")
```



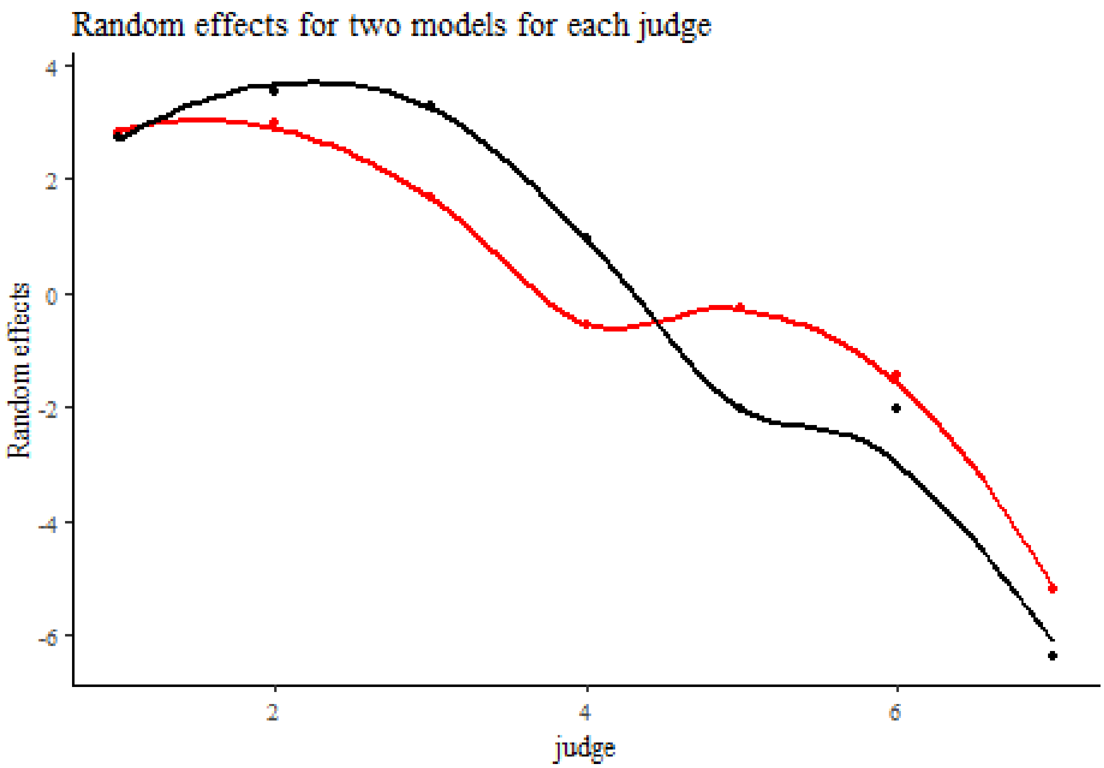

```
ggplot(new_olympics2,aes(x=pair,y=as.numeric(performance),group=judge,color=judge))+
  geom_point()+geom_smooth(method = "lm", se= FALSE)+ggtitle("Artristic scores")
```




```r
# Plot random effects among skaters
re_skater <- as.data.frame(cbind(unlist(ranef(techmer))[1:7],unlist(ranef(artimp))[1:7]))
re_skater$pair <-c(1:7)
ggplot(data=re_skater)+
  geom_point(col="red",aes(x=pair,y=V1))+geom_smooth(method="loess",col="red",aes(x=pair,y=V1),se=FALSE)
  geom_point(col="black",aes(x=pair,y=V2))+geom_smooth(method="loess",col="black",aes(x=pair,y=V2),se=FA
  ggtitle("Random effects for two models for each skater")+
  ylab("Random effects")
```




```
# Plot random effects among judges
re_judge <- as.data.frame(cbind(unlist(ranef(techmer))[1:7],unlist(ranef(artimp))[1:7]))
re_judge$judge <-c(1:7)
ggplot(data=re_judge)+
  geom_point(col="red",aes(x=judge,y=V1))+geom_smooth(method="loess",col="red",aes(x=judge,y=V1),se=FALS
  geom_point(col="black",aes(x=judge,y=V2))+geom_smooth(method="loess",col="black",aes(x=judge,y=V2),se=
  ggtitle("Random effects for two models for each judge")+
  ylab("Random effects")
```




### Different ways to write the model:

Using any data that are appropriate for a multilevel model, write the model in the five ways discussed in Section 12.5 of Gelman and Hill.

```
# Using the HIV dataset and model from the first problem
hiv_reg_vis <- lmer(y~time+factor(treatment)+age.baseline+(1|newpid), data = hiv.data)
summary(hiv_reg_vis)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ time + factor(treatment) + age.baseline + (1 | newpid)
##    Data: hiv.data
##
## REML criterion at convergence: 3137.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7490 -0.4392  0.0097  0.4282  5.0141
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  newpid   (Intercept) 1.8897   1.3747
##  Residual             0.5969   0.7726
## Number of obs: 1072, groups:  newpid, 250
##
## Fixed effects:
##                      Estimate Std. Error t value
## (Intercept)           5.08614    0.18793  27.064
```

```
## time               -0.36216    0.05399  -6.708
## factor(treatment)2  0.18008    0.18262   0.986
## age.baseline       -0.11945    0.04000  -2.986
##
## Correlation of Fixed Effects:
##             (Intr) time   fct()2
## time        -0.135
## fctr(trtm)2 -0.462  0.010
## age.baselin -0.727 -0.017 -0.003
```

The fixed effects part of the model: $y = \alpha_{j[i]} + \beta_{time} X_{itime} + \beta_{treatment} X_{itreatment} + \beta_{age.base} X_{iage.base} + \epsilon_i$

## 1st method: Allowing regression coefficeints to vary accross groups

$y = 4.91 + X_{itime} * (-0.36) + X_{itreatment} * (-0.12) + X_{iage.base} * 0.18 + 0.77$

$\alpha_j \sim$ N$(0, 1.37^2)$

## 2nd method: Combining separate local regressions

$y \sim N(4.91 + X_{itime} * (-0.36) + X_{itreatment} * (-0.12) + X_{iage.base} * (0.18), 0.77^2)$

$\alpha_j \sim$ N$(RandomIntercept, 1.37^2)$

## 3rd method: Modeling the coefficients of a large regression model

$y_i \sim N(4.91 + X_{itime} * (-0.36) + X_{itreatment} * (-0.12) + X_{iage.base} * (0.18), 0.77^2)$

$\beta_j \sim N(0, 1.37^2)$

## 4th method: Regression with multiple error terms

$y_i \sim N(4.91 + X_{itime} * (-0.36) + X_{itreatment} * (-0.12) + X_{iage.base} * (0.18) + 1.37^2, 0.77^2)$

## 5th method: Large regression with correlated errors

$y_i \sim N(4.91 + X_{itime} * (-0.36) + X_{itreatment} * (-0.12) + X_{iage.base} * (0.18), 1.37^2 + 0.77^2)$

### Models for adjusting individual ratings:

A committee of 10 persons is evaluating 100 job applications. Each person on the committee reads 30 applications (structured so that each application is read by three people) and gives each a numerical rating between 1 and 10.

1. It would be natural to rate the applications based on their combined scores; however, there is a worry that different raters use different standards, and we would like to correct for this. Set up a model for the ratings (with parameters for the applicants and the raters). $y_{score} = \alpha_{j[i]} + \beta_{cadidate} X_{iCadidate} + \beta_{rater} X_{iRater} + U_{RandomEffect-Rater}$

2. It is possible that some persons on the committee show more variation than others in their ratings. Expand your model to allow for this.

```
lmer(rating~applicants+raters+(1+raters|raters))
```