36-463/663: Multilevel and Hierarchical Models
Project, Part II
Example Solutions

1 Thinking about the response variable reacTime. [20 pts]

1(a). Initial exploration of reacTime shows an outlier at 46.605 (no reacTime should be greater than 10 seconds). Figure 1 shows a histogram of reacTime (on the left) and lrt = log(reacTime) (on the right), omitting this outlier. The histogram for reacTime is nearly symmetric and unimodal but is truncated on the left by zero and on the right at approximately 10 seconds. The truncation at zero cuts off more of the left tail than the truncation on the right; a linear model for reacTime might produce predictions outside the range 0 to 10; a prediction above 10 wouldn't be so bad, but a prediction below 0 indicated a negative reaction time, which is physically impossible.

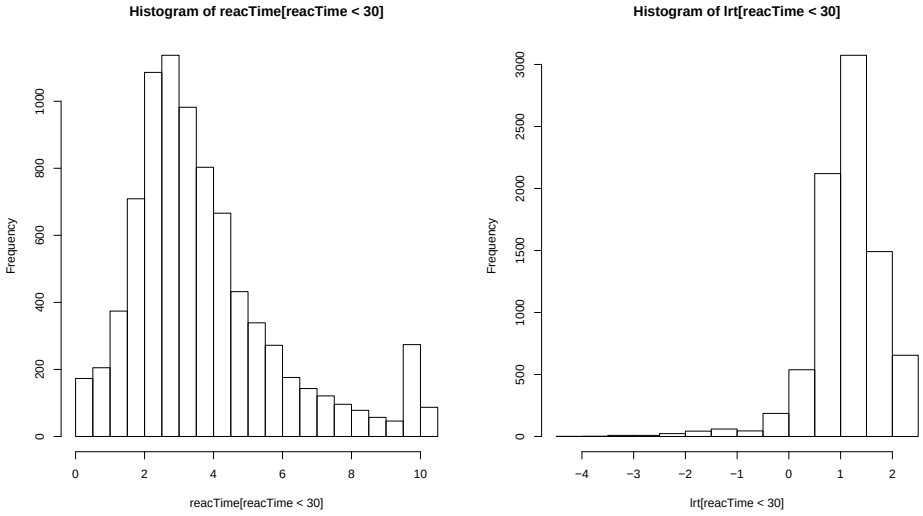


Figure 1: Histogram of reacTime (left) and lrt=log(reacTime) (right). The outlier above 30 sections (46.605sec) was omitted from both graphs.

The histogram for lrt still exhibits truncation on the right, around log(10)=2.303; there is also a long tail to the left. For using an ordinary normal-errors regression model, or a multilevel model with normal errors and normal random effects, the response variable does not need to be normally distributed (only the residuals and the random effects need to be normal); we are more concerned about outlying observations that will be unusually influential. For this reason we are not so concerned about the asymmetry per se, but we should watch for the effect of extremely negative lrt values in further modeling. The spike at log(10)=2.303 is not as obvious in this histogram as the spike at 10 in the reacTime histogram, but this is something else to pay attention to in further analyses.

1(b). In principle it should be the case that any player who takes more than timeLimit=10 seconds to respond will have their answer coded as "Time Out!!", and then the game moves on without allowing the player to complete their answer[1]. Thus we should see a cluster of observations at 10 seconds, and no times above 10 seconds. In fact, this rule was not enforced perfectly:

- There are 250 observations with reacTime $\leq$ 10sec (86 of these were exactly 10sec), that were given a "Time Out!!" coding. Most of these were above 9.6sec; the five observations below 9.6sec were: 8.657, 9.008, 9.046, 9.134, 9.197.
- There were 88 observations with reacTime $>$ 10sec. All of these were coded "Time Out!!" All but one of these were under 10.5sec, and all but five were under 10.2sec; these five were: 10.332, 10.453, 10.456, 10.465, 46.605.

The time limit affects modeling in two ways (at least):

1. The evidence from part Figure 1 is that, essentially, reaction times above 10 seconds are censored at 10 seconds and coded as (approximately) 10 seconds. Therefore our modeling cannot tell us anything interesting about longer reaction times (because we don't get to observe them, or the player's answer when he/she takes more than approximately 10sec to answer).
2. The clump of reaction times near 10 seconds caused by this game rule may draw the model away from other relationships that might exist when reaction time varies more continuously with other covariates.

Because the "observed" value will always be at or near 10 sec when "Time Out!!" is coded (except for the outlier at 46.605sec), we can expect residual plots to have a clear upper boundary line with slope -1, corresponding to predictions above 10 seconds; in these cases

$$\text{(raw residual)} = \text{(observed)} - \text{(expected)} \approx 10 - \text{(expected)}$$

will be a strictly decreasing function in "expected" value; the same will be true of a residual scaled by residual SD, studentized SD, etc.

Except for the outlier at 46.605sec, I did not see anything else in the reacTime variable to cause me to recode or remove observations. In the remainder of the analyses however, I will work with the data set that has the observation with reacTime=46.605 removed.

(There is also an argument for removing all of the observations coded as "Time Out!!"; this does not remove the first modeling problem above, but it does effectively remove the second one. It also makes the answerDec variable much more useful[1]. However, since I expect most students will have left the "Time Out!!" observations in the data set, so will I.)

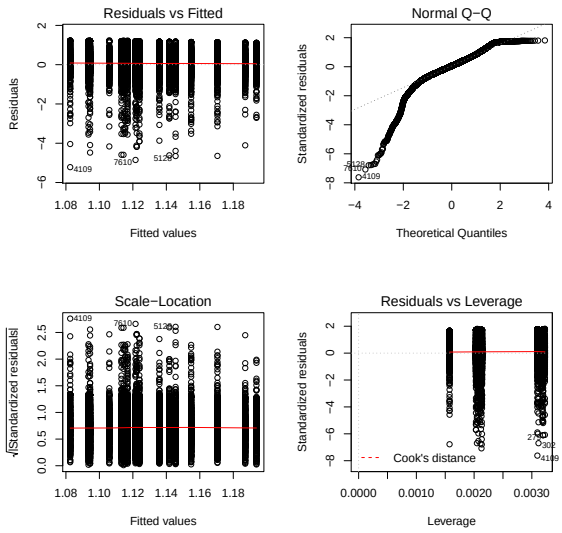
[1]Indeed, answerDec is coded as -1 in all "Time Out!!" cases, so we don't ever get to see the player's decimal answer when "Time Out!!" is coded, and the -1's seriously bias any regression coefficient if we try to use answerDec as a predictor variable.

2 Ordinary linear regression for question effects. [18 pts]

2(a). Figure 2(a) shows the coefficient estimates for a linear model for lrt, using currentQuestion *as a factor*, and omitting the intercept. Once again, omitting the intercept makes the coefficient for each question be the mean log reaction time over all players who attempted that question. $R^2 = 0.73$ for this model, which

	Estimate	Std. Error	t value
cq0.1	1.14552	0.03167	36.17
cq0.13	1.14180	0.03150	36.24
cq0.16	1.19384	0.03859	30.94
cq0.2	1.12388	0.03877	28.99
cq0.25	1.09443	0.03124	35.03
cq0.3	1.10592	0.03896	28.39
cq0.33	1.11441	0.03105	35.89
cq0.38	1.09350	0.03877	28.20
cq0.4	1.11318	0.03828	29.08
cq0.43	1.12393	0.03092	36.35
cq0.5	1.17053	0.02722	43.00
cq0.6	1.11543	0.03871	28.82
cq0.63	1.11683	0.03115	35.86
cq0.66	1.13584	0.03157	35.98
cq0.7	1.12229	0.03877	28.95
cq0.75	1.12216	0.03171	35.39
cq0.8	1.18714	0.03822	31.06
cq0.83	1.08255	0.03816	28.36
cq0.88	1.12159	0.03164	35.45
cq0.9	1.15482	0.03144	36.73

(a) Fitted regression coefficients.



(b) Diagnostic plots.

Figure 2: Ordinary linear regression predicting lrt from currentQuestion.

is a pretty good value given that we have not included any substantively interesting predictors yet. The diagnostic plots in Figure 2(b) initially look discouraging, but in fact they reflect features and flaws that we know exist in the data, and once these flaws are accounted for, they look pretty good: (i) The vertical lines in the residual plot are due to the fact that with 20 indicator variables for predictors, the model can only provide 20 distinct expected values for the reaction times; (ii) the censoring at 10sec can be seen in the upper limit of about 1.5 in the standardized residuals, and in the distinct elbow at approximaltely (2,1.5) in the qq plot; the long tail to the left in the histogram of lrt in Figure 1 is evident in the downward skewing in the residual plot, and in the deviation from the line $y = x$ in the qq plot below rougly the point $(-2, -2)$. If the values at 10sec and the long left tail are removed, the remainder of the residuals look reasonably normally distributed in the qq plot, as they should. The scale location plot shows no large deviation from the constant-scale assumption (the Cook's distance plot seems less useful here).

From Figure 2(a), the average lrt for each question is about 1.1, or exp(1.1)=3 or so seconds. Reaction times for the items located at 0.5 and near 0 and 1 are longest, approaching lrt=1.2 or exp(1.2)=3.3 sec.

2(b). The four plots, of the coefficients from the model in 2(a) against easiness, mean reacTime per question, mean lrt per question, and question location, are shown in Figure 3 (*Note*: the directions asked for a plot against log(mean of reacTime) but plotting against mean of log(reacTime) is more interesting!). We see

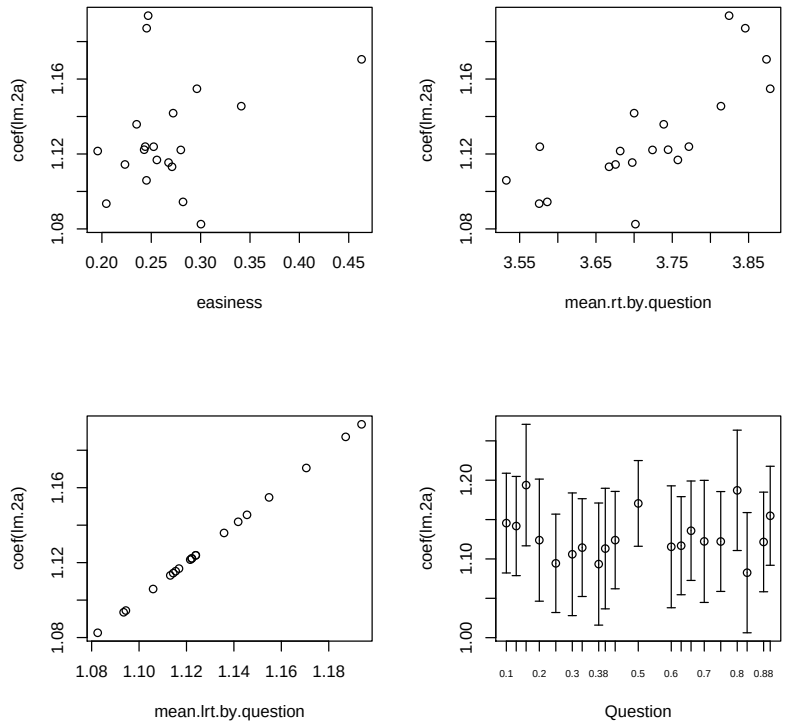


Figure 3: Plots of coefficient estimates in model from part 2(a) against question easiness; mean reacTime per question; mean log(reacTime) per question; and Question location.

an increasing relationship between the estimated coefficients and easiness (correlation=0.30) and between estimated coefficients and mean reacTime (correlation=0.73). As expected, the third plot shows that the coefficient estimates are exactly the means of log(reacTime) for each question. Finally, the fourth plot shows that questions with longer reaction times are at 0.5 and near 0 and 1 (though all the CI's are overlapping). It is somewhat surprising that the correlation with easiness is not higher, since this is the same pattern (easier questions at 0.5 and near 0 and 1) that we saw when we analyzed correct responses with logistic regression. (It's also interesting that easiness and reaction time are *positively* correlated: One does better locating the hidden sub, if one takes ones' time!)

2(c). I used the `stepAIC` function from `library(MASS)`, with smallest model `lrt ~ cq` and largest model `lrt ~ cq + experimentName + levelName + currentLevelNo + totalTrials + isGuidesEnabled + currentQuestion + hitType + currentAccuracy + avgAccuracy + itemsPlayed + currentStarCount + totalStarCount + fireType + sound + resp`. Again `answerDEC` was excluded because whenever `hitType=''Time Out!!''`, `answerDEC` is set to -1, which would seriously bias any regression coefficient

for answerDEC. Also, avgTime and bestTime were excluded becuse they are computed fromt the response variable lrt=log(reacTime). The final model was adjusted by removing the interecept, as with lm.2a above.

The final model was lrt ~ cq + hitType + experimentName + itemsPlayed + totalStarCount + currentAccuracy + levelName + avgAccuracy - 1. In this model cq is not significantly different from a constant, and in fact if we remove cq we can reduce the AIC by 14, from 15712 with cq to 15698 without. The coefficients of the other predictors are virtually identical as well. The coefficients on the continuous predictors show:

itemsPlayed: A slight decrease in reaction time is associated with playing more items. This could mean there is a practice effect for playing more items, or it might mean that players who are naturally faster play more items;

totalStarCount, currentAccuracy, avgAccuracy: A slight increase in reaction time is associated with getting more items right and with current and overall average accuracy. This is similar to the association between item easiness and longer reaction times that we saw earlier.

For the discrete predictors, each coefficient shows the effect relative to a baseline category:

hitType: No category has a reaction time significantly dfferent from the reaction time for "Miss!", except for the "Time Out!!" category, which naturally has a much longer reaction time (about exp(1.47)=4.35 seconds longer, on average).

experimentType: Although some categories have signifcantly longer or shorter reaction times than the baseline category "10thsG", there is no clear pattern about which categories have longer vs. shorter reaction times.

levelName: Interestingly all of these categories have shorter reaction times than the baseline category "10thsG" (even the "NoG" condition in which there are no tick- or guide-marks on the horizontal axis of the numberline). It may be that the "10thsG" condition just provides too many tick marks for fast visual estimation. However, it is surprising that the same effect did not show up for experimentName.

None of the variables resp, easiness, or currentQuestion showed up in the final model here. By adding them one-by-one to the final model above, we see:

- coef(summary(update(final.model, . . + resp - hitType)))["resp",] yields[2] a coefficient estimate of 0.1966 with SE=0.0191. This is significant and in the expected direction: longer response time is associated with correct response.
- coef(summary(update(final.model, . . + easiness - cq)))["easiness",] yields[3] a coefficient estimate of 0.0123 with SE=0.1101; this is in the expected direction but profoundly non-significant.
- coef(summary(update(final.model, . . + currentQuestion - cq)))["currentQuestion",] yields[4] a coefficient estimate of −0.0019 with SE=0.0268. It's not clear whether in fact there should be any linear association between reaction time and location of the question on the number line, so it's a good thing that this too is a profoundly non-significant effect.

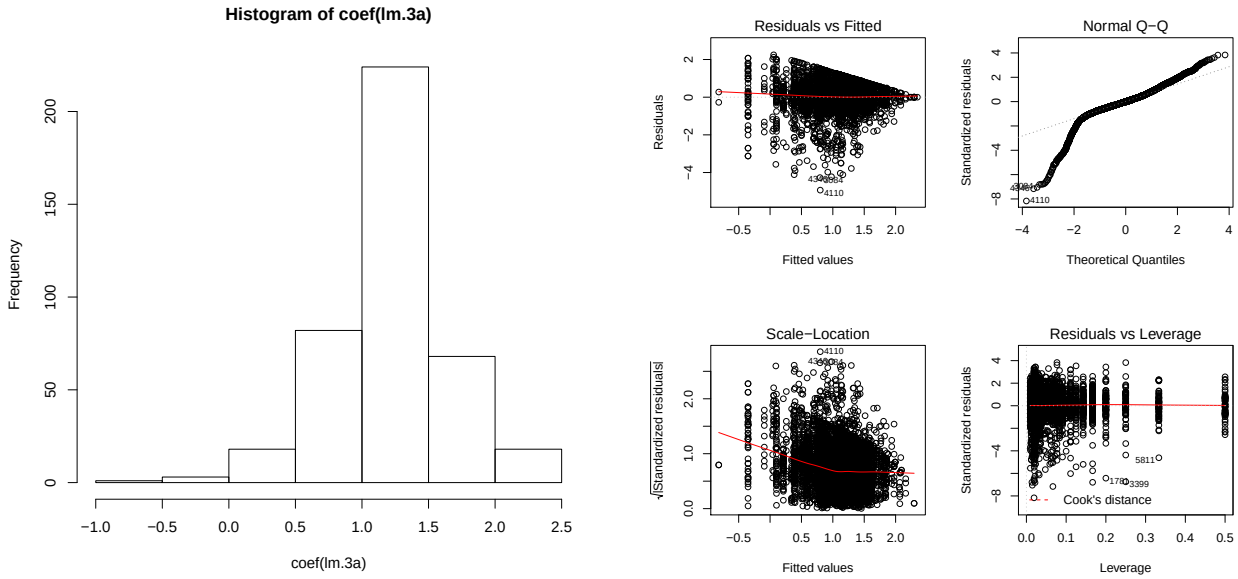
[2] hitType had to be removed, since resp is collinear with hitType.

[3] Actually, "easiness" had to be replaced with a longer variable that associated a question's easiness with the question, each time it was asked. See R code in Appendix 2 below. Also, cq had to be removed from the model because easiness is collinear with cq.

[4] Again, cq had to be removed for collinearity with "currentQuestion".

3 Ordinary linear regression for player effects. [20 pts]

3(a). Figure 4(a) provides a histogram of the coefficients from the mode `lrt ~ sID - 1`, where `sID <- as.factor(SID)` (it would take too much space to list all 414 coefficients). $R^2 \approx 0.80$ for this model. We



(a) Fitted regression coefficients.

(b) Diagnostic plots.

Figure 4: Ordinary linear regression predicting lrt from SID.

can see that the histogram of coefficients in Figure 4(a) is nearly symmetric and unimodal, so treating this as a random effect would not be crazy. There is a long tail to the left, reflecting what we saw in the histogram of lrt itself in Figure 1(b). The residual plot in Figure 4(b) exhibits the same long-right-tail behavior, and it also exhibits a sharp linear boundary with slope -1, as we predicted in our analysis in Section 1 above. Except for the long left tail, the residuals also look fairly normal in the qq plot. The scale-location plot shows some reduction in variance in the residuals for higher expected/fitted values; this isn't surprising given the sharp boundary on high residuals caused by the censoring at 10sec.

3(b). The four plots, of the coefficients from the model in 3(a) against proportion correct, player random effects from model `glmer.resp` above, mean reacTime per player, and mean lrt per player, are shown in Figure 5 (*Note*: the directions asked for a plot against log(mean of reacTime) but plotting against mean of log(reacTime) is more interesting!). There is almost no correlation (0.05) between the coefficients in the model and proportion-correct scores. There is a very modest correlation (0.18) between the model coefficients and random effects from `glmer.resp`. The stronger correlations are between the estimated coefficients and mean reaction time per player (0.90) and especially between the coefficients and mean of lrt (1.00). Once again, this confrms that the coefficients are just the means of lrt within each of the 414 players.

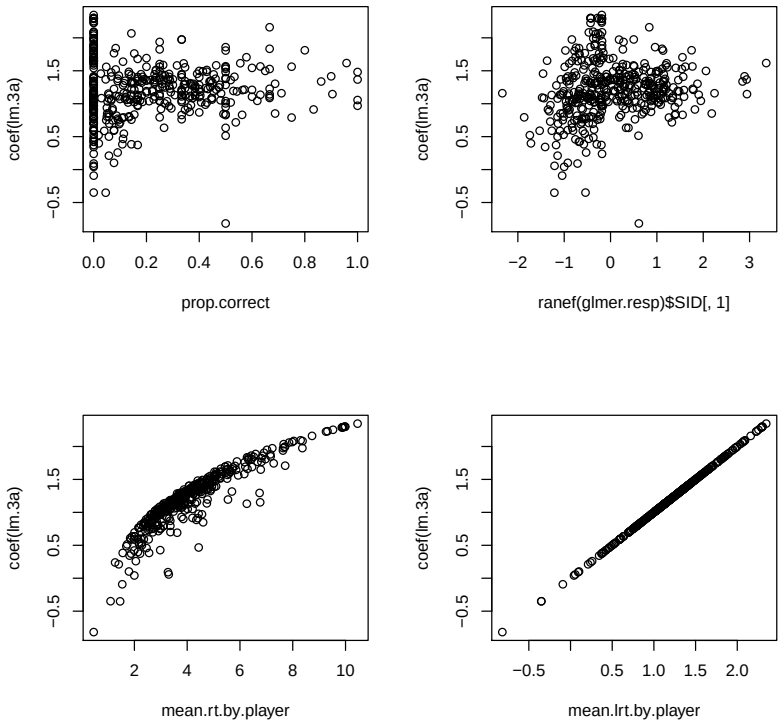


Figure 5: Plots of coefficient estimates in model from part 2(a) against proportion correct; random effects in model `glmer.resp`, mean reacTime per player; and mean log(reacTime) per question.

4 Mixed Effects Models [20 pts].

4(a). Figure 6(a) shows a partial summary of the fit of this initial mixed-effects model. Figure 6(b) gives a plot of the conditional residuals. The fixed effects for various levels of currentQuestion are similar to those in the model from 2(a). The residual plot in Figure 6(b) looks quite similar to the residual plot for the model from 3(a). As expected from "shrinkage", the SD of the random effect for SID, 0.3225, is not quite as large as `sd(coef(lm.3a))` = 0.4443.

4(b). I used the `fitLMER.fnc()` function from `library(LMERConvenienceFunctions)`, using AIC as the model comparison index. This backwards-eliminates fixed effects starting from a "largest" model, which I set to be `lrt ~ hitType + experimentName + itemsPlayed + totalStarCount + currentAccuracy + levelName + avgAccuracy + (1|SID)`, then forward selects random effects (which I took to be `(1|ip)`, `(1|qip2)` and `(1|ip3)`), and finally backwards eliminates fixed effects again, in case the random effects render some of the fixed effects unnecessary. Note that the "smallest" model is forced to be the intercept-only model, by this routine.

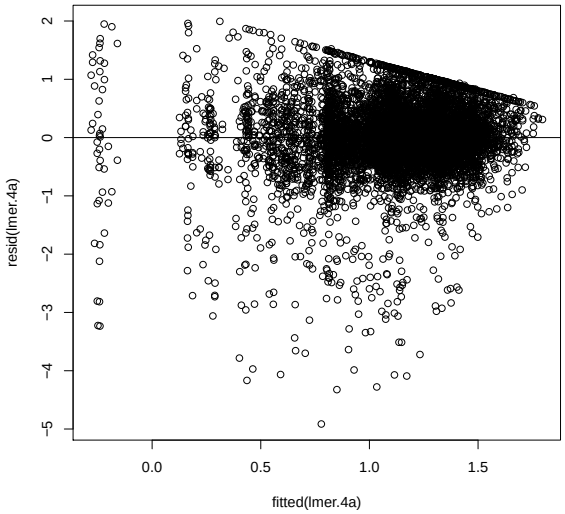
The final model chosen in this way was `lrt ~ hitType + experimentName + itemsPlayed + currentAccuracy + (1 | SID) + (1 | ip)`. Comparing the fit with the other models we've tried, we get

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: lrt ~ cq - 1 + (1 | SID)

Random effects:
 Groups   Name        Variance Std.Dev.
 SID      (Intercept) 0.1040   0.3225
 Residual             0.3725   0.6103
Number of obs: 8256, groups:  SID, 414

Fixed effects:
       Estimate Std. Error t value
cq0.1   1.18600    0.03348   35.43
cq0.13  1.16659    0.03327   35.06
cq0.16  1.24696    0.03902   31.95
cq0.2   1.17812    0.03913   30.11
cq0.25  1.13285    0.03306   34.27
cq0.3   1.15432    0.03930   29.37
cq0.33  1.16265    0.03293   35.31
cq0.38  1.14186    0.03914   29.17
cq0.4   1.15746    0.03874   29.87
cq0.43  1.15392    0.03279   35.19
cq0.5   1.20522    0.02996   40.23
cq0.6   1.16616    0.03912   29.81
cq0.63  1.16271    0.03302   35.21
cq0.66  1.16758    0.03332   35.04
cq0.7   1.16455    0.03917   29.73
cq0.75  1.16455    0.03354   34.72
cq0.8   1.22130    0.03869   31.57
cq0.83  1.12604    0.03869   29.10
cq0.88  1.16762    0.03342   34.93
cq0.9   1.18718    0.03320   35.75
```

(a) Fitted regression coefficients.



(b) Plot of conditional residuals.

Figure 6: Mixed linear model predicting lrt from fixed currentQuestion and random SID effects.

	lm.2a	final.model	lm.3a	lmer.4a	lmer.start	final.lmer
AIC	17226.26	15712.53	15678.63	15949.20	14840.12	14773.88
BIC	17373.65	16056.45	18591.39	16103.61	15057.70	14942.33

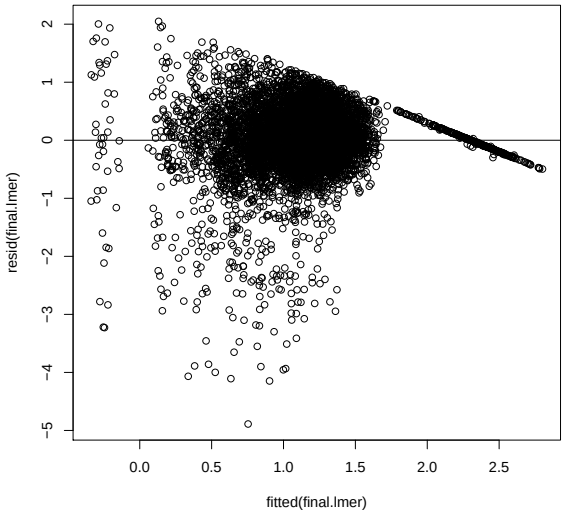
In this case the final lmer model fitz best. A summary is given in Figure 7, along with a plot of the conditional residuals. 75% of the ip addresses were used by five or more players, with a mean around 30 players per ip address, so perhaps it is not surprising that ip is a needed second random intercept; the model with both random intercepts looks like a 3-level hierarchical linear model, with the SID random effect nested within the ip random effect. The fixed effect coefficients have interpretations very similar to the corresponding fixed effects in part 2(c) above so I will not repeat that interepretation here. Instead we consider the residual plot in Figure 7(b), which looks a lot like the residual plot for the final model in 2(c) [not shown]. The upper boundary of slope -1 caused by the censoring at 10sec is obvious, as is the long-right-tail. Another feature of this residual plot (as well as for the one for the model in 2(c)) is the arm extending down and to the right. A little exploration confirms that the observations in this arm correspond exactly to "Time Oot!!" coding; this is picked up because hitType is in the model now.

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: lrt ~ hitType + experimentName + itemsPlayed + currentAccuracy +
    (1 | SID) + (1 | ip)

Random effects:
 Groups   Name        Variance Std.Dev.
 SID      (Intercept) 0.05932  0.2435
 ip       (Intercept) 0.02731  0.1652
 Residual             0.31922  0.5650
Number of obs: 8256, groups:  SID, 414; ip, 280

Fixed effects:
                                 Estimate Std. Error t value
(Intercept)                     1.1628030  0.0676759  17.182
hitTypeNear Miss!!              0.0566237  0.0266104   2.128
hitTypePartial Hit!!            0.0314439  0.0245331   1.282
hitTypePerfect Hit!!            0.0536558  0.0203327   2.639
hitTypeTime Out!!               1.2946426  0.0462350  28.001
experimentNameDecimal_10thsG_AB -0.1494909  0.0803225  -1.861
experimentNameDecimal_10thsG_BA -0.2300065  0.0872547  -2.636
experimentNameDecimal_3rdsG     -0.2111003  0.0861758  -2.450
experimentNameDecimal_3rdsG_AB  -0.1389740  0.0803330  -1.730
experimentNameDecimal_3rdsG_BA  -0.2324620  0.0820888  -2.832
experimentNameDecimal_4thsG     -0.1424232  0.0801761  -1.776
experimentNameDecimal_4thsG_AB  -0.0436584  0.0828812  -0.527
experimentNameDecimal_4thsG_BA  -0.2191193  0.0794286  -2.759
experimentNameDecimal_midG      -0.2180612  0.0772609  -2.822
experimentNameDecimal_MidG_AB   -0.2094482  0.0813258  -2.575
experimentNameDecimal_MidG_BA    0.0116804  0.0706191   0.165
experimentNameDecimal_NoG       -0.1600209  0.0781861  -2.047
experimentNameDecimal_NoG_AB    -0.1646918  0.0806824  -2.041
experimentNameDecimal_NoG_BA    -0.2358847  0.0804178  -2.933
itemsPlayed                     -0.0024034  0.0004107  -5.852
currentAccuracy                  0.0012404  0.0004492   2.761
```

(a) Fitted regression coefficients.



(b) Plot of conditional residuals.

Figure 7: Final mixed effects model for predicting lrt.

5 Models combining reaction time and correctness of response. [22 pts]

Many researchers believe that reaction time and correctness of response should be related in some way. For example, perhaps, as the question gets harder to answer correctly, the reaction time should go up. Or perhaps reaction time is not so much related to correctness of response as it is to how "engaging" the player finds the game to be.

5(a). From our work above (especially fitting and interpreting the final model in 2(c) with "resp" replacing "hitType"), and from work in Project Part I, resp seems to sometimes have a positive relation with lrt (or reacTime), and sometimes a negative relation, depending on the model it appaers in. We would think that "resp" and "lp" would function similarly in predicting lrt or reacTime.

5(b). The variable lp is the logit of the probability of a correct response; it depends on both lpayer ability and inherent question difficulty/easiness. Figure 8 plots lrt vs lp; the correlation between these variables is a modest 0.18. This is confirmed by linear regression; the fitted model relating lp to lrt is lrt = $1.27 + 0.11$(lp), with the coefficient on "lp" highly significant. The practical effect is nearly nil however; R^2 for this model is only 0.03. Note that this is the opposite of what we saw with resp—a small negative effect on lrt.

Using the same backward/forward/backward procedure as was used in section 4(b) with upper model

```
lrt ~ lp + hitType + experimentName + itemsPlayed + cq + currentStarCount + isGuidesEnabled
```

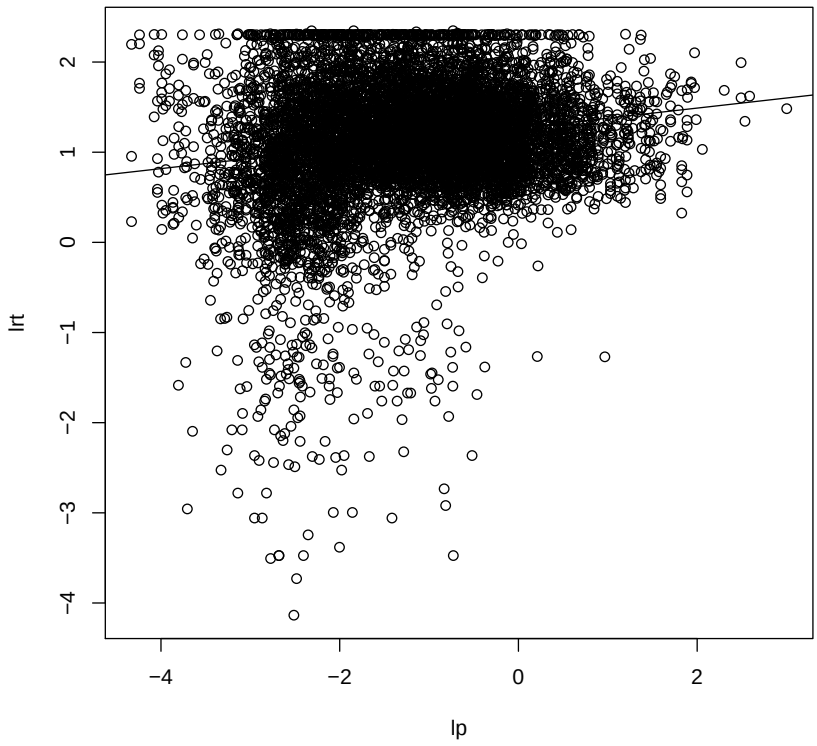


Figure 8: Scatter plot of lrt on lp with fitted regression line overlaid.

`+ currentAccuracy + avgAccuracy + fireType + (1|SID)`, and again considering adding random intercepts for ip, ip2 and ip3, we get as a final model

```
lrt ~ lp + hitType + itemsPlayed + cq + currentAccuracy + (1 | SID) + (1 | ip)
```

Most of the predictors in this model we have seen before, and they function in basically the same way as we have seen also. It is interesting that "lp" was reained, and again has a positive, not a negtive effect, on "lrt". The residual plot looks much like Figure 7(b), with a similar interpretation.

5(c)(i). The model in the JAGS code can be written as follows:

- Level l:

$$\begin{aligned} y_n &\sim Bernoulli(p_n) \\ \text{logit}(p_n) &= \beta_{j[n]} + \alpha_{k[n]} \\ \log(t_n) &\sim N(\rho \cdot \text{logit}(p_n), \sigma^2) \end{aligned}$$

where $n = 1, \ldots, 8257$ (observations), $j = 1, \ldots, 20$ (questions), and $k = 1, \ldots, 414$ (people); y_n is 0 or 1 depending on whether the player got the question right; and t_n is the reaction time for the player on that question.

- Fixed effects priors[5]:

$$\begin{aligned} \beta_j &\sim N(0, 10^{10}), \forall j = 1 \ldots, 20, \\ &\quad \text{with the additional constraint that } \textstyle\sum_j \beta_j = 0 \\ \rho &\sim N(0, 10^{10}) \end{aligned}$$

- Level 2:

$$\alpha_k \sim N(\beta_0, \tau^2), \forall k = 1, \ldots, 414$$

- Level 2 priors[5]:

$$\begin{aligned} \tau &\sim Unif(0, 10) \\ \sigma &\sim Unif(0, 10) \\ \beta_0 &\sim N(0, 10^{10}) \end{aligned}$$

5(c)(ii). Examining the output from `joint.mcmc.0`, we see that most of the $\hat{R}$'s are much larger, ranging from 2.84 to 47.15. The effective MCMC sample sizes are quite a bit smaller than the expected 1000 samples, raging all the way down to an n_{eff} of only 3. This indicates substantial autocorrelation in the chains, which makes it difficult to treat the chains as a simple random sample. A typical plot from the p3() function is shown in Figure 9(a). The chains do indeed exhibit high autocorrelation, and one chain is well-separated from othe other two—they aren't yet exploring the same part of the posterior distriutions (if they ever will!).

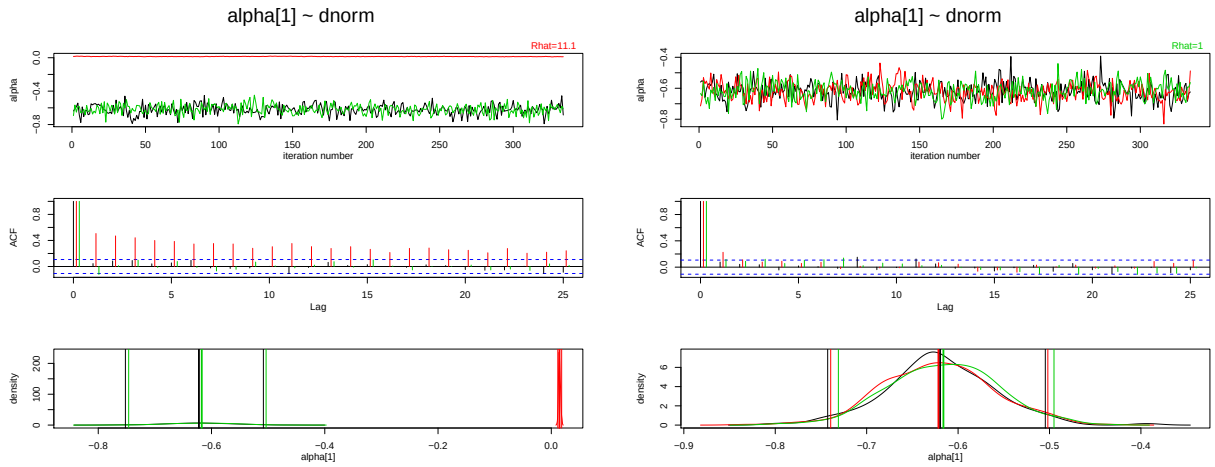
On the other hand, the output from `joint.mcmc.1` shows all $\hat{R}$'s less than or equal to 1.03, and most n_{eff}'s at or near 1000, indicating low autocorrelation in the chains. The p3() plot corresponding to Figure 9(a) is shown in Figure 9(b); it shows three well-behaved Markv chains that have converged on the same part of the posterior distribution.

The difference between the two runs is in the starting values. The inits function for `joint.mcmc.0` started all parameters at independent N(0,1) draws. The inits function for `joint.mcmc.1` started at random draws from normal densities centered at the posterior median for each parameter, and scaled by the corresponding posterior SD. Even though these are not great estimates (given how poorly joint.mcmc.0 ran!), they get us close enough that all the chains explore the same part of the posterior density, near the (true) posterior mode.

5(c)(iii). Because it is stable and better-behaved, we will choose to work with `joint.mcmc.1`. was the more successful and reliable fit, and interpret the fitted model. In Figure 10 we have displayed point estimates for the fixed effects, overall mean of the random effects, regression coefficient (`rho`) for `lp` predicting `lrt`, and the residual SD `sig` and random effect SD `tau`.

The fitted model in Figure 10 presents something of a riddle:

[5]Not required to receive full credit!



(a) A p3() trace from `joint.mcmc.0`.

(b) A p3() trace from `joint.mcmc.1`.

Figure 9: Exploring the two MCMC runs.

- From Figure 10(a), the overall intercept `b0` (overall mean lrt) is −0.9406, suggesting a mean reaction time around exp(−0.9406) = 0.3904sec, which seems wrong, since the mean of reacTime is around 3.73 sec.
- All but one of the question coefficients (`b1` through `b20`) are not significantly different from zero, and the absolute size of the significant coefficient (`b3`) is still very near zero. The nonsignificance of questions as a factor is at least consistent with the other models we have encountered.
- `rho`, the coefficient on `lp` in the model, is estimated to be −1.26. This is in contrast to the estimated value of 1.27 in the model relating `lp` to `lrt` in 5(b) above (and a positive estimate for the coefficient on `lp` in the more complex "final" model for `lrt` in 5(b) as well).
- The random effects in Figure 10 are mostly negative and skewed to the right; whereas the corresponding quantities in Figure 4(a) and similar models are mostly positive and skewed left.
- The residual plot looks *very much* like the residual plots in Figure 4(b), Figure 6(b), and for similar models that don't include `hitType`.

How can the signs be all wrong, and yet the residual plot looks pretty much the same? The answer is in realizing that the mean log reaction time in this model is

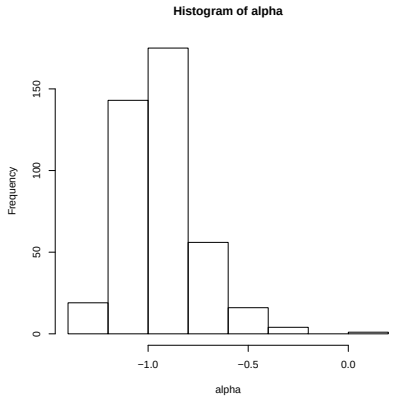
$$(\rho) \times (\beta_{j[n]} + \alpha_{k[n]}).$$

If the β_j's are all essentially zero, this reduces to approximately

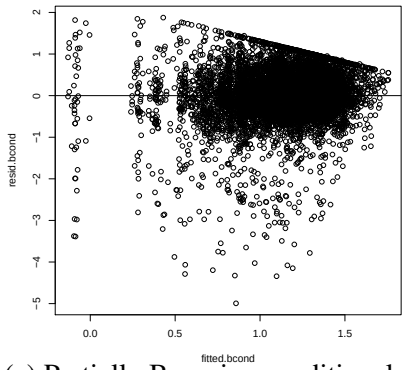
$$(\rho) \times (\alpha_{k[n]}) \qquad (*)$$

	mean	sd	z
b0	-0.9406	0.0267	-35.2793
b1	0.0053	0.0215	0.2476
b2	0.0033	0.0222	0.1468
b3	-0.0623	0.0265	-2.3503
b4	-0.0095	0.0265	-0.3588
b5	0.0327	0.0219	1.4952
b6	0.0075	0.0265	0.2848
b7	-0.0003	0.0210	-0.0146
b8	0.0054	0.0264	0.2059
b9	0.0107	0.0259	0.4131
b10	0.0080	0.0214	0.3742
b11	0.0210	0.0188	1.1192
b12	0.0033	0.0257	0.1292
b13	0.0047	0.0214	0.2185
b14	-0.0052	0.0219	-0.2383
b15	-0.0029	0.0262	-0.1111
b16	0.0082	0.0216	0.3795
b17	-0.0449	0.0256	-1.7503
b18	0.0382	0.0262	1.4595
b19	-0.0168	0.0216	-0.7773
b20	-0.0064	0.0211	-0.3057
rho	-1.2608	0.0313	-40.3282
sig	0.6169	0.0050	123.2127
tau	0.2318	0.0136	17.0733

(a) Posterior means and SD's of fixed effects.



(b) Scatter plot of random effect posterior means.



(c) Partially Bayesian conditional residuals.

Figure 10: Partial summary of the fitted model for 5(c)(iii).

and we get the same mean predictions from this formula as we do from

$$(-\rho) \times (-\alpha_{k[n]}) \qquad (**)$$

This is a kind of lack of identifiability in the model: two sets of parameters (positive and negative) give the same predictinos about the data. In this case, once we realize it, there's no real harm done, since we can choose whichever of $(*)$ or $(**)$ is more convenient for our work.

Also, note that if the β_j's had been substantially different from zero, the argument about the equivalence of $(*)$ and $(**)$ wouldn't have worked! So our lack of identifiability is in part due to data that does not distinguish the easiness of questions well.

Finally, the fact that the coefficient on `lp` was positive and significant in several of our models (all of them, if we take $(**)$ above), means that the basic idea of van der Linden's (2009) paper seems to be confirmed, at least for this dataset: there is a relationship between difficulty of questions for a player (based on `lp`=logit P[getting item right]) and the player's reaction time (as measured by log(`reacTime`)).

Appendix I: References

CRAN (2016). The Comprehensive R Archive Network[6] (https://cran.r-project.org/). Author.

Gelman, A. & Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models.* NY: Cambridge Univ Press.

Lomas, D., Patel, K., Forlizzi, J.L., and Koedinger, K.R. (2013). Optimizing Challenge in an Educational Game Using Large-Scale Design Experiments. Paper presented at CHI 2013, Paris, France. Obtained online at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.480.2493&rep=rep1&type=pdf

Lynch, Scott M. (2007). *Introduction to Applied Bayesian Statistics and Estimation for Social Scientists.* New York: Springer.

Schofield, L. S., Junker, B., Taylor, L. J., & Black, D. A. (2014). Predictive inference using latent variables with covariates. *Psychometrika,* 283–314.

van der Linden W.J. (2009). Conceptual Issues in Response-Time Modeling. *Journal of Educational Measurement, 46(3),* 247–272.

[6]For documentation on various packages.

Appendix II: Additional material, including R code.

Here is all the code that I used to prepare these solutions. I have indicated in comments which part of the project each block of code was for... (This is much more than I would expect you to include, and more than I would grade, but I thought you might find it interesting).

```
######################################################################

library(arm)
library(R2jags)
library(rube)
library(MASS)
library(LMERConvenienceFunctions)
library(plotrix)  # for plotting confidence intervals

dec.data <- read.csv("dec-data.csv",header=T)


######################################################################

# 1. Thinking about the response variable reacTime. [20 pts]
# Our main response variable for this part of the project will be
# reacTime.

# (a) Explain why lrt = log(reacTime) might make more sense to use as
# a response variable, than reacTime itself, if we are going to build
# standard hierarchical linear models with normal errors and normal
# random effects. Make sure to write complete, clear sentences; feel
# free to use figures or calculations, if it will help.

lrt <- log(dec.data$reacTime)

# initial plotting shows an extreme outlier in reacTime of 46.605 seconds.
# if we omit this observation, the plots are as follows:

attach(dec.data)
par(mfrow=c(1,2))
hist(reacTime[reacTime<30])
hist(lrt[reacTime<30])            # hist-reacTime-lrt.pdf
detach()

# There are tradeoffs in considering reacTime vs lrt.  reacTime is
# more nearly symmetric which is a strong point in its favor.  For
```

```
# building linear models however, it is good to have a response
# variable with domain the whole real line (which lrt has) so that we
# don't run the risk of making predictions with the linear model
# outside the valid range of the response variable.

#####

# (b) The variable timeLimit is equal to 10 for all observations in
# the data set; this is intended to mean that all players have a
# maximum of 10 seconds to answer, or their answer will be coded as
# "times up!!", which in our analysis is a wrong answer (resp=0).

# i. Look for, and report, any evidence about how well or poorly this
# time limit is adhered to, in the data.

# If timeLimit=10 is strictly enforced, then there should neer be a
# "times up!!" code if respTime<10, and always a "times up!!" code
# if respTime>10.

attach(dec.data)
par(mfrow=c(1,1))
reacTime[(reacTime<=10)&(hitType=="Time Out!!")]
length(reacTime[(reacTime<=10)&(hitType=="Time Out!!")])  # 250
min(reacTime[(reacTime<=10)&(hitType=="Time Out!!")])     # 8.657
hist(reacTime[(reacTime<=10)&(hitType=="Time Out!!")])
reacTime[(reacTime>10)&(hitType!="Time Out!!")]           # none!
reacTime[reacTime>10]
length(reacTime[reacTime>10])                             # 88
detach()

# ii. How might this time limit affect fitting standard linear
# regression models, hierarchical linear models with normal random
# effects, etc.? How might it affect residual plots?

# The time limit affects modeling in two ways (at least):
#
# (1) the evidence from part (i) is that, essentially, reaction times
# above 10 seconds are censored at 10 seconds and coded as
# (approximately) 10 seconds.  Therefore our modeling cannot tell us
# anything interesting about longer reaction times (because we don't
# get to observe them).
#
```

```
# (2) The clump of reaction times at 10 seconds may draw the model
# away from other relationships that might exist when reaction time
# varies more continuously with other covariates.
#
# We can expect residual plots to have a clear upper boundary line
# with slope -1, corresponding to predictions above 10 seconds; in
# this case residual = observed - expected = 10 - expected will be a
# strictly decreasing function in "expected".

# iii. Is there anything else about the reacTime variable that might
# lead you to recode or remove some values? If so, say what you found,
# and what you will do about it.
#
# Two commments:
#
# (1) I will delete the observation with reaction time 46.605 seconds
# and work with the remaining data set.
#
# (2) As I do analyses I will think about whether the clump of
# observations at approximately 10 seconds need special treatment.

dec.data <- dec.data[-which(dec.data$reacTime>30),]

lrt <- log(dec.data$reacTime)

# Page limit for section 1: 2 pages.

#
#
# ###########################################################################
#
# 2. Ordinary linear regression for question effects. [18 pts]

# (a) Fit a linear model for lrt, using currentQuestion as a factor,
# and omitting the intercept. Summa- rize the fit, and comment on why
# taking the intercept out might be a useful idea.

cq <- as.factor(dec.data$currentQuestion)

summary(lm.2a <- lm(lrt ~ cq - 1))

par(mfrow=c(2,2))
```

```
plot(lm.2a)

# (b) Plot the coefficients in this model against each of the
# following:

attach(dec.data)

par(mfrow=c(2,2))

# ¢ Item easiness (the proportion of players attemping the item, who
#   got it right (resp=1))

easiness <- sapply(split(resp,currentQuestion),mean)

plot(easiness, coef(lm.2a))

# ¢ The mean of all reaction times on this question

mean.rt.by.question <- sapply(split(reacTime,currentQuestion),mean)

plot(mean.rt.by.question,coef(lm.2a))

# ¢ The log of the mean reaction time on this question

# I asked for this
plot(log(mean.rt.by.question),coef(lm.2a))

# but this makes a better / more interesting plot
mean.lrt.by.question <- sapply(split(lrt,currentQuestion),mean)
plot(mean.lrt.by.question,coef(lm.2a))

# ¢ The target value for this question (currentQuestion as a numeric
#   variable)

# here's a simple plot
target.value <- as.numeric(names(easiness))
plot(target.value,coef(lm.2a))

# here's a more informative one, from library(plotrix)
est.2a <- coef(summary(lm.2a))
betahat <- est.2a[,1]
LL <- betahat - 2*est.2a[,2]
```

```
UL <- betahat + 2*est.2a[,2]
Question <- sort(unique(dec.data$currentQuestion))
plotCI(Question, betahat, li=LL,ui=UL,axes=F,ylab="coef(lm.2a)")
box()
axis(2)
axis(1,at=Question,labels=Question,tick=T,cex.axis=0.6)


# Summarize the results, and provide and comment on any interesting
# plots (you do not have to provide plots that arent interesting!).

round(cor(data.frame(coef=coef(lm.2a),easiness,mean.rt.by.question,
               log(mean.rt.by.question),target.value)),2)

round(cor(data.frame(coef=coef(lm.2a),easiness,mean.rt.by.question,
               mean.lrt.by.question,target.value)),2)

detach()

# (c) Use an automatic variable selection procedure to find the best
# model that does include current Question as a factor, but does not
# include SID in any form.

# i. Show or tell what variable selection method(s) you used, and what
# were the smallest and largest models you considered.

# I selected the following possible predictors, in addition to cq:

# experimentName, levelName, currentLevelNo, totalTrials,
# isGuidesEnabled, currentQuestion, hitType, currentAccuracy,
# avgAccuracy, itemsPlayed, avgTime, bestTime, currentStarCount,
# totalStarCount, fireType, sound, resp, reacTime

# again, answerDec was excluded because answerDec=-1 when
# hitType="Time Out!!"; also I removed avgTime and bestTime since
# they are computed from the response variable

lm.2a.1 <- lm(lrt ~ cq,data=dec.data)

final.model.cq <- stepAIC(lm.2a.1, scope=list(lower= ~ cq, upper= ~ cq +
experimentName + levelName + currentLevelNo + totalTrials +
isGuidesEnabled + currentQuestion + hitType + currentAccuracy +
```

```
avgAccuracy + itemsPlayed + currentStarCount +
totalStarCount + fireType + sound + resp))

final.model.cq <- update(final.model.cq, . ~ . -1)
# the model referred to in the text as "final.model" is in fact
# "final.model.cq".

final.model.0 <- stepAIC(lm.2a.1, scope=list(lower= ~ 1, upper= ~ cq +
experimentName + levelName + currentLevelNo + totalTrials +
isGuidesEnabled + currentQuestion + hitType + currentAccuracy +
avgAccuracy + itemsPlayed + currentStarCount +
totalStarCount + fireType + sound + resp))

final.model <- update(final.model.0, . ~ . - 1)
# this model allows cq to be removed. The predictors are the same as
# for "final.model.cq" (except, of course, cq is missing).  Since
# there is really no cq effect, it's not surprising that the
# coefficient estimates for the other predictors are nearly identical
# for the two models.

round(coef(summary(final.model)),3)

summary(lrt)

par(mfrow=c(2,2))
plot(final.model)
exp(c(2,2.5)) # 7.39, 12.18

bozo <- split(fitted(final.model),dec.data$hitType)
sapply(bozo,summary)

# comparing coefficients of final.model.cq and final.model (for
# the predictors in common)
round(cbind(coef(update(final.model, .~.+1))[-1],
            coef(final.model.cq)[-(1:20)]),4)

# examining the effect of cq directly in final.model.cq (not even
# close to needed!)
round(coef(summary(update(final.model.cq,.~.+1)))[1:20,],4)

coef(summary(update(final.model.cq, . ~ . + resp - hitType)))["resp",]
coef(summary(update(final.model.cq, . ~ . + easiness)))["easiness",]
```

```
   # doesn't work; see below for a fix...
coef(summary(update(final.model.cq,
                    . ~ . + currentQuestion - cq)))["currentQuestion",]

# so there are two interesting features of the residual plots:
# (1) the separation of "Time Out!!" data
# (2) the long left tail - suggesting the model is really overpredicing
#     some reaction times.

######

# ii. Show the final model and write a short paragraph interpreting
# the fitted model and parameter estimates etc. in the fitted
# model. Remember, you are writing for Dr. Lomas, so provide a clear
# interpretation that will be useful to him in understanding the
# relationships between predictors and outcome in the model.

# iii. if resp, or easiness, or CurrentQuestion as a numeric variable,
# is in the final model, does eacgh of their relationships with lrt
# make sense? Explain why or why not.

summary(update(final.model, . ~ . + resp - hitType + 1))

attach(dec.data)
long.easiness <- rep(NA,length(cq))    # code each instance of a question
for (i in names(easiness)) {           # with its easiness...
  long.easiness[currentQuestion==as.numeric(i)] <- easiness[i]
}
detach()

summary(update(final.model, . ~ . + long.easiness))
coef(summary(update(final.model.cq,
                    . ~ . + long.easiness - cq)))["long.easiness",]

summary(update(final.model, . ~ . + currentQuestion))

# Page limits for section 2: one page for each of the three subparts.

# ##############################################################################
#
# 3. Ordinary linear regression for player effects. [20 pts]
```

```
# (a) Fit a linear model for lrt, using SID as a factor, and omitting
# the intercept. Summarize and interpret the fit.

sID <- as.factor(dec.data$SID)
summary(lm.3a <- lm(lrt ~ sID - 1, data=dec.data))

# (b) Plot the estimated coefficients of this model agaist each of the
# following:

attach(dec.data)

par(mfrow=c(2,2))

# ¢ proportion correct: The proportion of items attempted by that
#   player, which the player got right.

prop.correct <- sapply(split(dec.data$resp,dec.data$SID),mean)

plot(prop.correct,coef(lm.3a))

# ¢ The player random effects from the model glmer.resp <- glmer(resp
#    cq - 1 + (1|SID),family=binomial), where cq <-
#   factor(currentQuestion).

glmer.resp <- glmer(resp ~ cq - 1 + (1|SID),family=binomial,data=dec.data)

plot(ranef(glmer.resp)$SID[,1],coef(lm.3a))

# ¢ The mean reaction time of each player, over all the questions that
#   player tried

mean.rt.by.player <- sapply(split(reacTime,SID),mean)

plot(mean.rt.by.player,coef(lm.3a))

# ¢ The log of the mean reaction time

# I asked for this
plot(log(mean.rt.by.player),coef(lm.3a))

# but this is more interesting
```

```
mean.lrt.by.player <- sapply(split(lrt,SID),mean)

plot(mean.lrt.by.player,coef(lm.3a))

detach()

# Summarize the results, and provide and comment on any interesting
# plots.

cor(data.frame(coef=coef(lm.3a),prop.corr=prop.correct,
               reff=ranef(glmer.resp)$SID[,1],rt=mean.rt.by.player,
               lrt=mean.lrt.by.player))

# Page limits for section 3: 1 page for each of the 2 subparts.
#
# ###########################################################################
#
# 4. Mixed Effects Models [20 pts]

# Our primary interest is in what factors affect the reaction time of
# each player on each question It is quite likely that questions
# answered by the same person will be dependent on each other through
# that players ability. For these reasons, we will focus on questions
# as fixed effects and players as random effects.

# (a) Fit a standard mixed effects linear regression model predicting
# lrt, using currentQuestion as a fixed effect factor, omitting the
# intercept, and with a random intercept grouped by SID (that is, all
# these should be implemented in your model). Summarize and interpret
# the fit.

summary(lmer.4a <- lmer(lrt ~ cq - 1 + (1|SID),data=dec.data,REML=F))

plot(fitted(lmer.4a),resid(lmer.4a))

rbind(AIC=
      c(lm.2a=AIC(lm.2a),final.model=AIC(final.model),lm.3a=AIC(lm.3a),
        lmer.4a=AIC(lmer.4a)),
      BIC=
      c(lm.2a=BIC(lm.2a),final.model=BIC(final.model),lm.3a=BIC(lm.3a),
        lmer.4a=BIC(lmer.4a)))
```

```
#        lm.2a final.model    lm.3a  lmer.4a
# AIC 17226.26    15698.38 15678.63 16049.76
# BIC 17373.65    15908.94 18591.39 16204.17

# (b) Use an automatic variable selection method to try to improve
# this model.

lmer.start <- lmer(lrt ~ hitType + experimentName + itemsPlayed +
    totalStarCount + currentAccuracy + levelName + avgAccuracy +
    (1|SID), data=dec.data,REML=F)


final.lmer <- fitLMER.fnc(lmer.start,method="AIC",
                          ran.effects=list(ran.intercepts=c("ip","ip2","ip3")))

formula(lmer.start)
# lrt ~ hitType + experimentName + itemsPlayed + totalStarCount +
#     currentAccuracy + levelName + avgAccuracy + (1 | SID)
formula(final.lmer)
# lrt ~ hitType + experimentName + itemsPlayed + currentAccuracy +
#     (1 | SID) + (1 | ip)

plot(fitted(final.lmer),resid(final.lmer))

rbind(AIC=
      c(lm.2a=AIC(lm.2a),final.model=AIC(final.model.cq),lm.3a=AIC(lm.3a),
        lmer.4a=AIC(lmer.4a),
        lmer.start=AIC(lmer.start),final.lmer=AIC(final.lmer)),
      BIC=
      c(lm.2a=BIC(lm.2a),final.model=BIC(final.model.cq),lm.3a=BIC(lm.3a),
        lmer.4a=BIC(lmer.4a),
        lmer.start=BIC(lmer.start),final.lmer=BIC(final.lmer)))

#        lm.2a final.model    lm.3a  lmer.4a lmer.start final.lmer
# AIC 17226.26    15698.38 15678.63 16049.76   14840.12   14773.88
# BIC 17373.65    15908.94 18591.39 16204.17   15057.70   14942.33


# i. Show or tell what variable selection method(s) you used, and what
# the smallest and largest models (both fixed and random effects) you
# considered were.
```

```
# ii. Show the final model and write a short paragraph interpreting
# the fitted model and parameter estimates etc. in the fitted
# model. Remember, as before, you are writing for Dr. Lomas.

# Page limits for section 4: 1 page for each of the 2 subparts.
#
# ###############################################################
#
# 5. Models combining reaction time and correctness of response. [22
# pts] Many researchers believe that reaction time and correctness of
# response should be related in some way. For example, perhaps, as the
# question gets harder to answer correctly, the reaction time should
# go up. Or perhaps reaction time is not so much related to
# correctness of response as it is to how engaging the player finds
# the game to be.

# (a) Review your work from Section 1 (and possibly some of your work
# from Project Part I;) and try to determine whether one of these
# theories (or some other theory), about the connection between
# reaction time and correctness of response, is suggested. Write a
# short paragraph, with figures or other displays if needed, giving
# evidence for your conclusion(s).

# (b) van der Linden (2009, p. 254) 3 suggests, essentially, that we
# make the logit of the probability of a correct response one of the
# predictors for the response time model. We can get the logits of the
# success probability for each question encountered by each player
# with lp <- predict(glmer.resp), where glmer.resp is the model
# referred to in section 3 above.

# i. Using graphs and an ordinary linear regression model, explore the
# relationship between lrt and lp. Is lp a good predictor of lrt? Is
# the direction of the relationship what you would expect?


lp <- predict(glmer.resp)

par(mfrow=c(1,1))
plot(lp,lrt)
summary(lm.lp <- lm(lrt ~ lp,data=dec.data))
abline(lm.lp)
```

```
# ii. Use a variable selection procedure to try to improve the linear
# model you fitted in part (i).

final.lp <- stepAIC(lm.lp, scope=list(lower = ~ 1, upper= ~ lp + cq +
experimentName + levelName + currentLevelNo + totalTrials +
isGuidesEnabled + currentQuestion + hitType + currentAccuracy +
avgAccuracy + itemsPlayed + currentStarCount +
totalStarCount + fireType + sound + resp))

summary(final.lp)

formula(final.model)

# lrt ~ hitType + experimentName + itemsPlayed + totalStarCount +
#     currentAccuracy + levelName + avgAccuracy - 1

formula(final.lp)

# lrt ~ lp + hitType + experimentName + itemsPlayed + cq + currentStarCount +
#     isGuidesEnabled + currentAccuracy + avgAccuracy + fireType


lp.lmer.start <- lmer(lrt ~ lp + hitType + experimentName +
      itemsPlayed + cq + currentStarCount + isGuidesEnabled +
      currentAccuracy + avgAccuracy + fireType + (1|SID),data=dec.data)

lp.lmer.final <- fitLMER.fnc(lp.lmer.start,method="AIC",
                          ran.effects=list(ran.intercepts=c("ip","ip2","ip3")))

summary(lp.lmer.final)

formula(lp.lmer.final)

plot(fitted(lp.lmer.final),resid(lp.lmer.final))

# Summarize and interpret the fit of your final model.
#
#
# (c) A basic problem with the linear regression in part (b) (i) here,
# is that it assumes that lp is fixed and known, but in fact lp is
# only estimated (through estimates of fixed and random effects in the
```

```
# model glmer.resp). The multilevel Bayesian model in Figure 1 (below)
# tries to correct this, by modeling lrt and resp at the same time,
# using the same components as the models in part (b) (i).

# i. Write, using the most precise mathematical language that you can
# use, the model repre- sented in Figure 1. It is not necessary to
# write out the prior distributions, just get the struc- ture of the
# model at levels 1 and 2 as clear and correct as possible. Feel free
# to use the code in Figure 1 as well as the modeling components in
# part (b) (i), to figure this out.

# ii. Unfortunately the mcmc fit does not run very fastit takes about
# two hours for a typical useful run. Rather than make you wait for
# the runs, I have produced three fitted model objects,
# joint.mcmc.test, joint.mcmc.0 and joint.mcmc.1, using the code in
# Figure 2.  You can load these fitted models into your R session by
# copying the file fitted-mcmc.RData from the Project Part 2 directory
# into your working directory on your laptop, and issuing the command
# 4 load("fitted-mcmc.RData") in R.

load("fitted-mcmc.RData")

# A. Inspect the fitted model objects joint.mcmc.0 and joint.mcmc.1
# with the print() and p3() commands, and any other tools if they seem
# useful. Does it look like, for either fit, the Markov chain has
# converged to its stationary distribution? Why or why not? Are the
# estimates from either run useful and reliable? Why or why not?
# Provide evidence for your conclusions.

joint.mcmc.0
joint.mcmc.1

p3(joint.mcmc.0)
p3(joint.mcmc.1)

# B. The setup for fitting joint.mcmc.1 was slightly different from
# the setup for fitting joint.mcmc.0. Describe the difference, and
# explain why this might have contributed to a different level of
# success for one fit vs the other.

# iii. Choose whichever of joint.mcmc.0 or joint.mcmc.1 was the more
# successful and reliable fit, and interpret the fitted
```

```
# model. Compare it with the fitted model in part (b) (i). Is the
# conclusion about the relationship between lrt and lp the same 5 ?

#   $ mean           :List of 7
#   ..$ alpha   : num [1:414(1d)] -0.619 -1.096 -0.846 -0.952 -0.695 ...
#   ..$ b       : num [1:20(1d)] 0.00533 0.00326 -0.06228 -0.00952 0.03273 ...
#   ..$ b0      : num [1(1d)] -0.941
#   ..$ deviance: num [1(1d)] 25347
#   ..$ rho     : num [1(1d)] -1.26
#   ..$ sig     : num [1(1d)] 0.617
#   ..$ tau     : num [1(1d)] 0.232

alpha.means <- joint.mcmc.1$mean$alpha

coef.means <- joint.mcmc.1$mean[2:7][c(2,1,4,5,6)]

coef.sds <- joint.mcmc.1$sd[2:7][c(2,1,4,5,6)] # drop deviance from coef list
                                               # and rearrange.

result <- cbind("mean"=unlist(coef.means),"sd"=unlist(coef.sds))
result <- cbind(result,"z"=result[,1]/result[,2])

round(result,4)

par(mfrow=c(1,1))
alpha <- unlist(alpha.means)
hist(alpha)

# cheap conditional residuals coming next -- faster than re-running to
# get fake data sets!

bb <- result[2:21,1]
brho <- result[22,1]

sID <- as.factor(dec.data$SID)
tmp <- data.frame(b=bb[as.numeric(cq)],a=alpha[as.numeric(sID)])
fitted.bcond <- with(tmp,brho*(b+a))
resid.bcond <- lrt - fitted.bcond

par(mfrow=c(1,1))
plot(fitted.bcond,resid.bcond)
abline(h=0)
```

```
# Page limits for section 5: 1 page each for parts (a) and (b); up to
# 3 pages for part (c).
#
# ####################################################################
```