Homework 2 Rubric

Andrew Latimer

11 November 2019

Question 1

Using the Char Height data set from class (char_with_fake.csv), construct a model with random effects for these data, using CharHt as the response variable. Assume that Transect is the only relevant grouping factor, and that Steepness (of the topography) and Diameter (of the trees) are the only available predictors.

First some general throat-clearing and exploration.

Load data and explore structure:

```
char <- read.csv("char_with_fake.csv")
head(char)
```

```
##   X Fire Transect Treated Diameter ScorchHt CharHt Steepness        fake
## 1 1    1        1       1    0.084   -0.680  1.140     1.395  1.15183273
## 2 2    1        1       1   -0.133   -0.729  1.304     1.395 -0.04443366
## 3 3    1        1       1   -0.498   -0.567  0.707     1.395 -0.37586221
## 4 4    1        1       1   -0.133   -1.154  0.632     1.395  0.09068745
## 5 5    1        1       1   -0.573   -0.592  1.549     0.807  0.49703304
## 6 6    1        1       1   -0.350   -0.243  0.837     0.807 -0.06424229
```

```
str(char)
```

```
## 'data.frame':    776 obs. of  9 variables:
##  $ X        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Fire     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Transect : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Treated  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Diameter : num  0.084 -0.133 -0.498 -0.133 -0.573 ...
##  $ ScorchHt : num  -0.68 -0.729 -0.567 -1.154 -0.592 ...
##  $ CharHt   : num  1.14 1.304 0.707 0.632 1.549 ...
##  $ Steepness: num  1.395 1.395 1.395 1.395 0.807 ...
##  $ fake     : num  1.1518 -0.0444 -0.3759 0.0907 0.497 ...
```

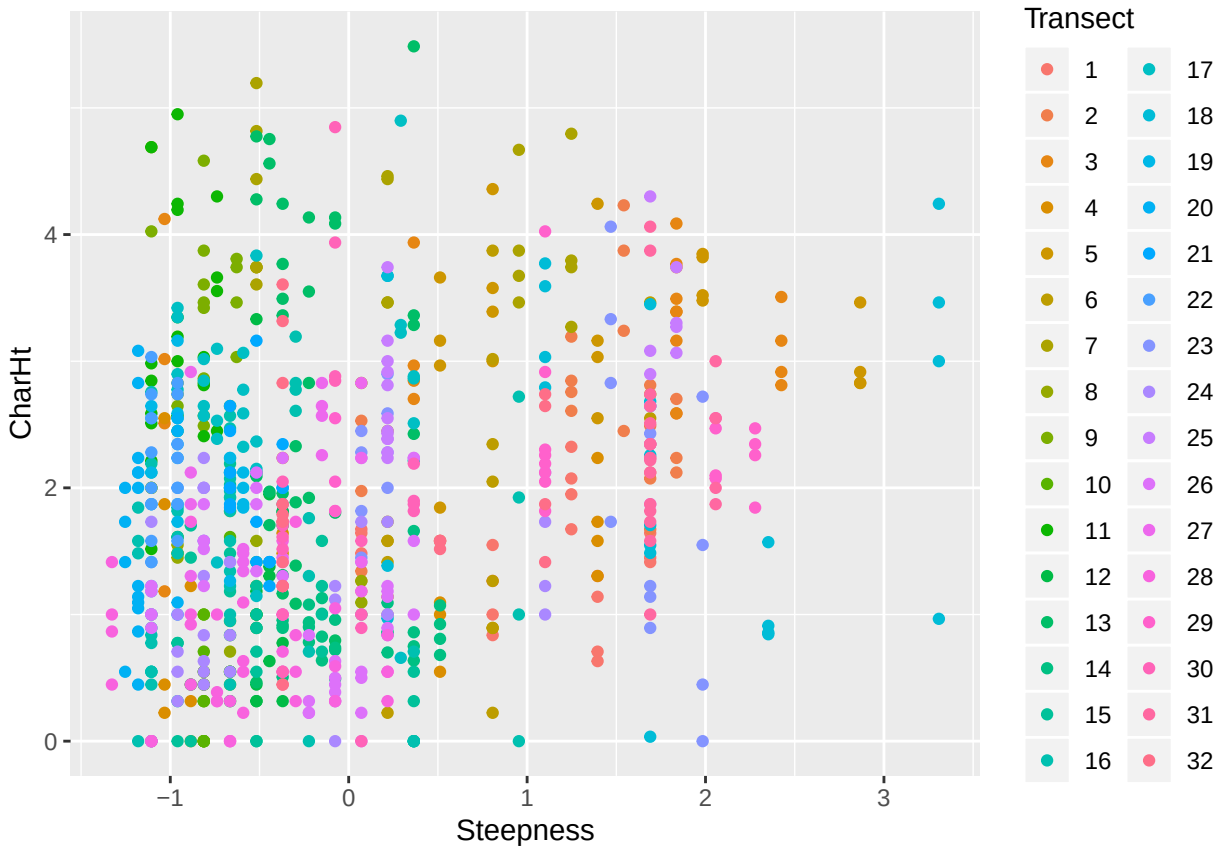
```
sum(complete.cases(char)) # any missing values to worry about?
```

```
## [1] 776
```

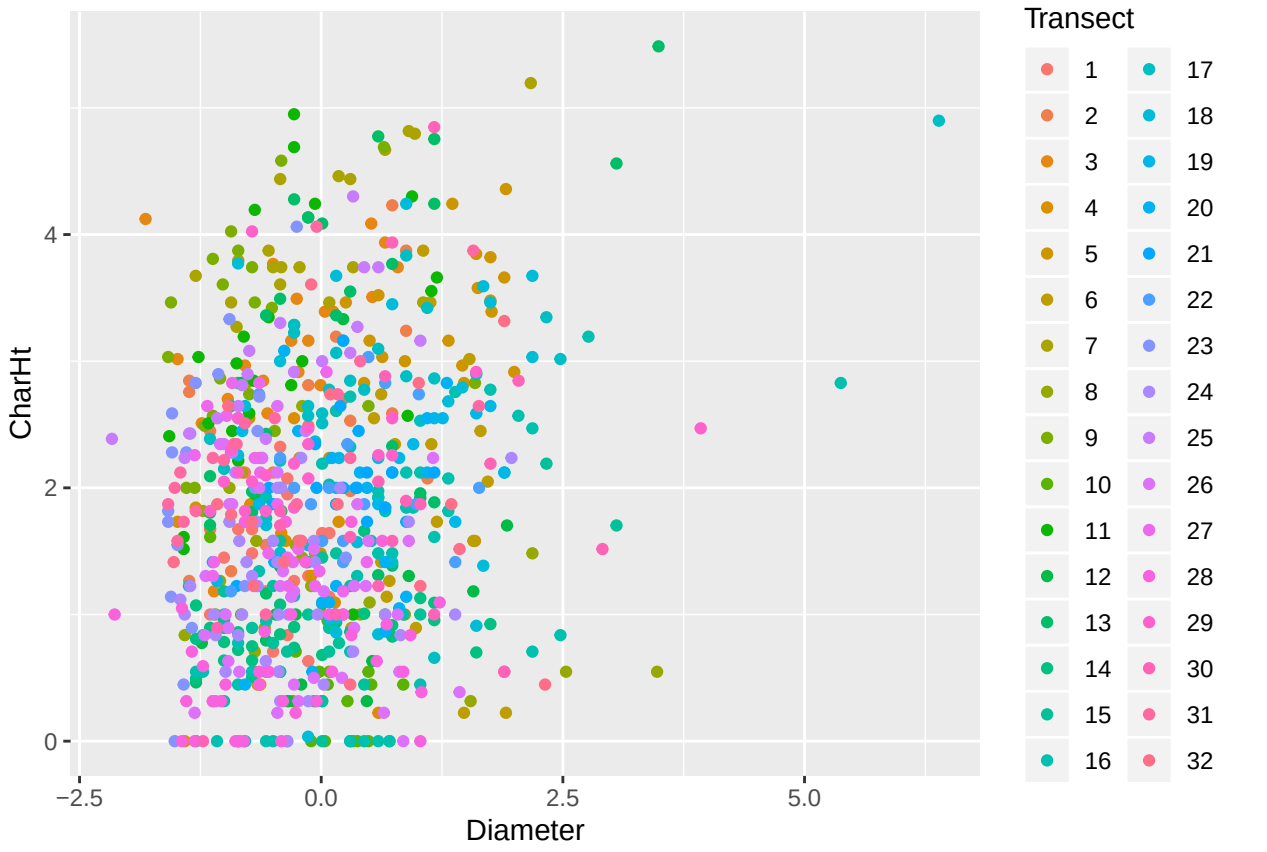
```
tabulate(char$Transect) # there is variation in sample size by transect, though most have 20 plots in t
```

```
##  [1] 20 20 20 20 20 20 20 20 20 20 24 16 32 28 28 56 28 24 20 20 20 20 20
## [24] 40 20 40 20 40 20 20 20 20
```

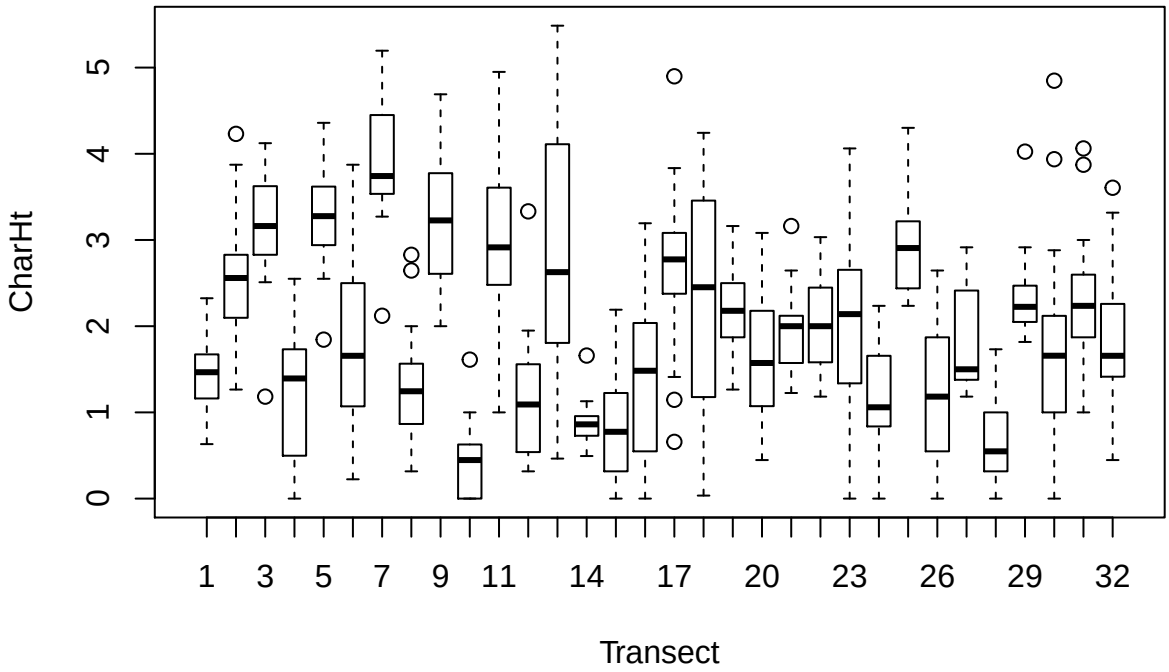
```
char$Transect <- as.factor(char$Transect) # maybe not necessary but to make sure

ggplot(char, aes(x = Steepness, y = CharHt, color = Transect))+
  geom_point()
```




```
ggplot(char, aes(x = Diameter, y = CharHt, color = Transect))+
  geom_point()
```




```
boxplot(CharHt~Transect, char)
```



Looks like a lot of variation both within and among transects, supporting including transect as a grouping factor.

a) A brief explanation of how you chose variables, and which (if any) you decided to allow to vary

by group (Transect).

In addition to showing variation, transect is a grouping factor in the design, so we should account for it in the analysis.

Some hint of positive relationship between char height and both terrain steepness (makes sense, fires often more intense uphill) and tree diameter (not so clear why).

Create a set of models to compare:

```
m1 <- lmer(CharHt~(1|Transect), data = char, REML=FALSE)
# using REML = F for model comparison
m2 <- lmer(CharHt~Diameter + (1|Transect), data = char, REML=FALSE)
m3 <- lmer(CharHt~Steepness + (1|Transect), data = char, REML=FALSE)
m4 <- lmer(CharHt~Steepness+Diameter+(1|Transect), data = char, REML=FALSE)
m5 <- lmer(CharHt~Steepness*Diameter+(1|Transect), data = char, REML=FALSE)

display(m1)
```

```
## lmer(formula = CharHt ~ (1 | Transect), data = char, REML = FALSE)
## coef.est  coef.se
##     1.98     0.15
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Transect (Intercept) 0.83
##  Residual             0.77
## ---
## number of obs: 776, groups: Transect, 32
## AIC = 1907.5, DIC = 1901.5
## deviance = 1901.5
```

```
display(m2)
```

```
## lmer(formula = CharHt ~ Diameter + (1 | Transect), data = char,
##     REML = FALSE)
##             coef.est coef.se
## (Intercept) 1.98     0.15
## Diameter    0.24     0.03
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Transect (Intercept) 0.83
##  Residual             0.74
## ---
## number of obs: 776, groups: Transect, 32
## AIC = 1846.4, DIC = 1838.4
## deviance = 1838.4
```

```
display(m3)
```

```
## lmer(formula = CharHt ~ Steepness + (1 | Transect), data = char,
##     REML = FALSE)
##             coef.est coef.se
## (Intercept) 1.97     0.14
```

```
## Steepness   0.14     0.05
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Transect (Intercept) 0.79
##  Residual             0.77
## ---
## number of obs: 776, groups: Transect, 32
## AIC = 1900.8, DIC = 1892.8
## deviance = 1892.8
```

```
display(m4)
```

```
## lmer(formula = CharHt ~ Steepness + Diameter + (1 | Transect),
##     data = char, REML = FALSE)
##             coef.est coef.se
## (Intercept) 1.97     0.14
## Steepness   0.14     0.04
## Diameter    0.24     0.03
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Transect (Intercept) 0.80
##  Residual             0.73
## ---
## number of obs: 776, groups: Transect, 32
## AIC = 1838.3, DIC = 1828.3
## deviance = 1828.3
```

```
display(m5)
```

```
## lmer(formula = CharHt ~ Steepness * Diameter + (1 | Transect),
##     data = char, REML = FALSE)
##                     coef.est coef.se
## (Intercept)         1.97     0.14
## Steepness           0.14     0.04
## Diameter            0.24     0.03
## Steepness:Diameter 0.03     0.03
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Transect (Intercept) 0.80
##  Residual             0.73
## ---
## number of obs: 776, groups: Transect, 32
## AIC = 1839.5, DIC = 1827.5
## deviance = 1827.5
```

```
AICc(m1, m2, m3, m4, m5)
```

```
##    df     AICc
## m1  3 1907.558
## m2  4 1846.495
## m3  4 1900.821
## m4  5 1838.412
## m5  6 1839.644
```

Of these, the best AIC score goes to the model with both explanatory variables, but no interaction.

It's possible the effect of terrain slope or tree diameter varies by transect. I have no strong theoretical or biological reason to think so, so it's justifiable to not go there. But because I'm in exploratory mode here and don't have very strong expectations about that, I'll give it a try.

```
m4.b <- lmer(CharHt~Steepness+Diameter + (1 + Diameter|Transect), data = char, REML=FALSE)
m4.c <- lmer(CharHt~Steepness + Diameter+(1 + Steepness|Transect), data = char, REML=FALSE)

AICc(m4, m4.b, m4.c)
```

```
##      df     AICc
## m4    5 1838.412
## m4.b  7 1822.403
## m4.c  7 1841.170
```

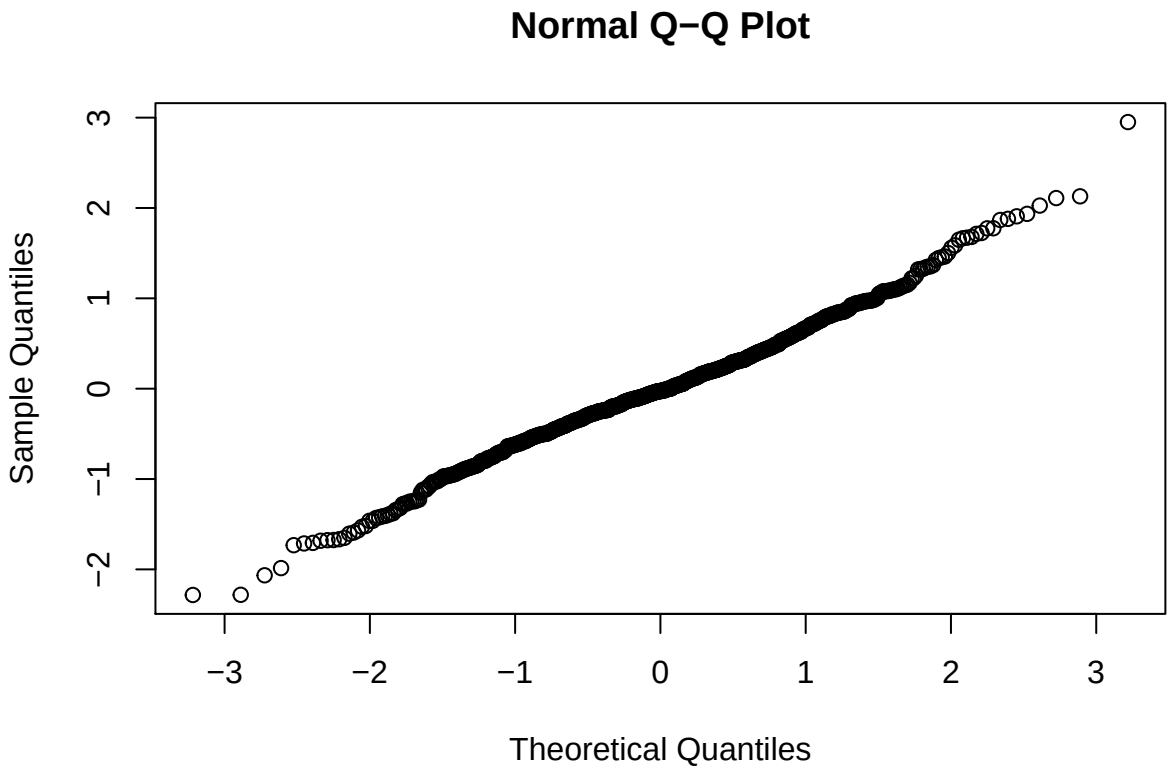
```
BIC(m4, m4.b, m4.c)
```

```
##      df      BIC
## m4    5 1861.605
## m4.b  7 1854.836
## m4.c  7 1873.603
```

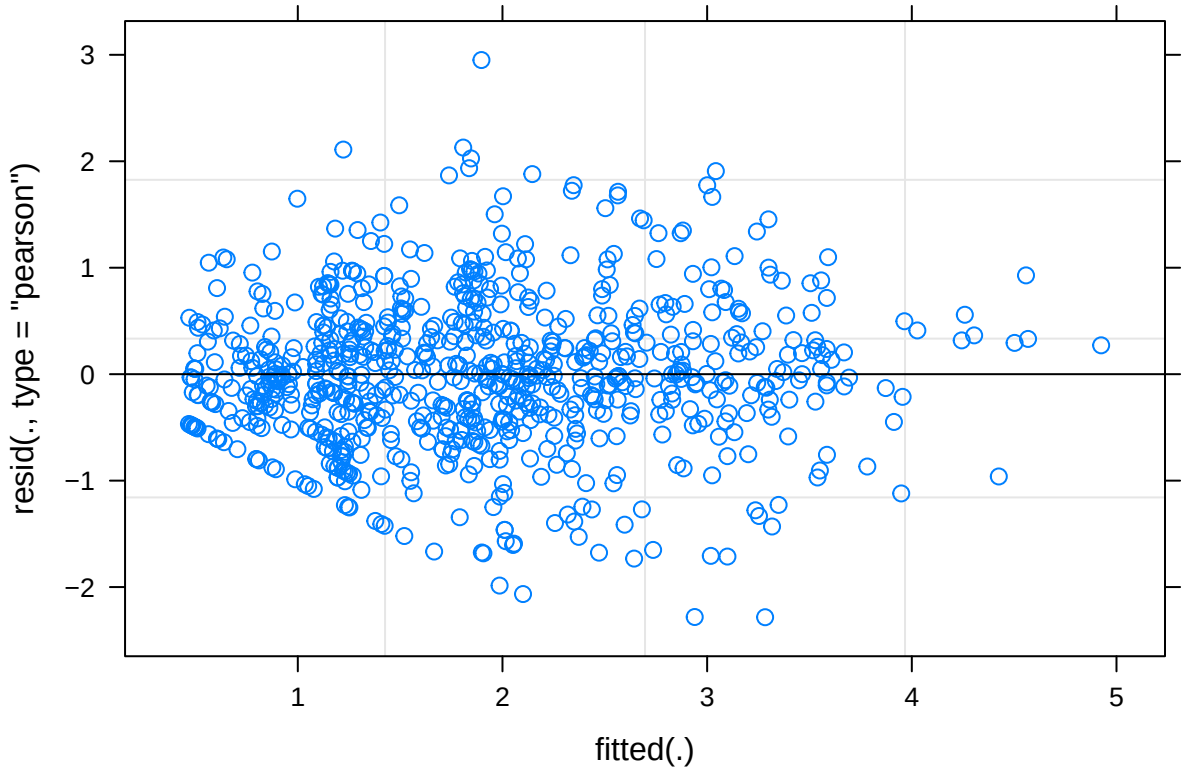
The model with random slope for Diameter is clearly best by model comparison. If I really wanted to refine this or stress-test it, I would go on and try holding out some data and predicting it using a few of the best candidate models and check for overfitting that way. This would also provide a reality check about absolute model performance. But for now, I'll accept model 4b.

It's worth quickly checking some assumptions though. Looks pretty good.

```
qqnorm(resid(m4.b))
```



```
plot(m4.b)
```

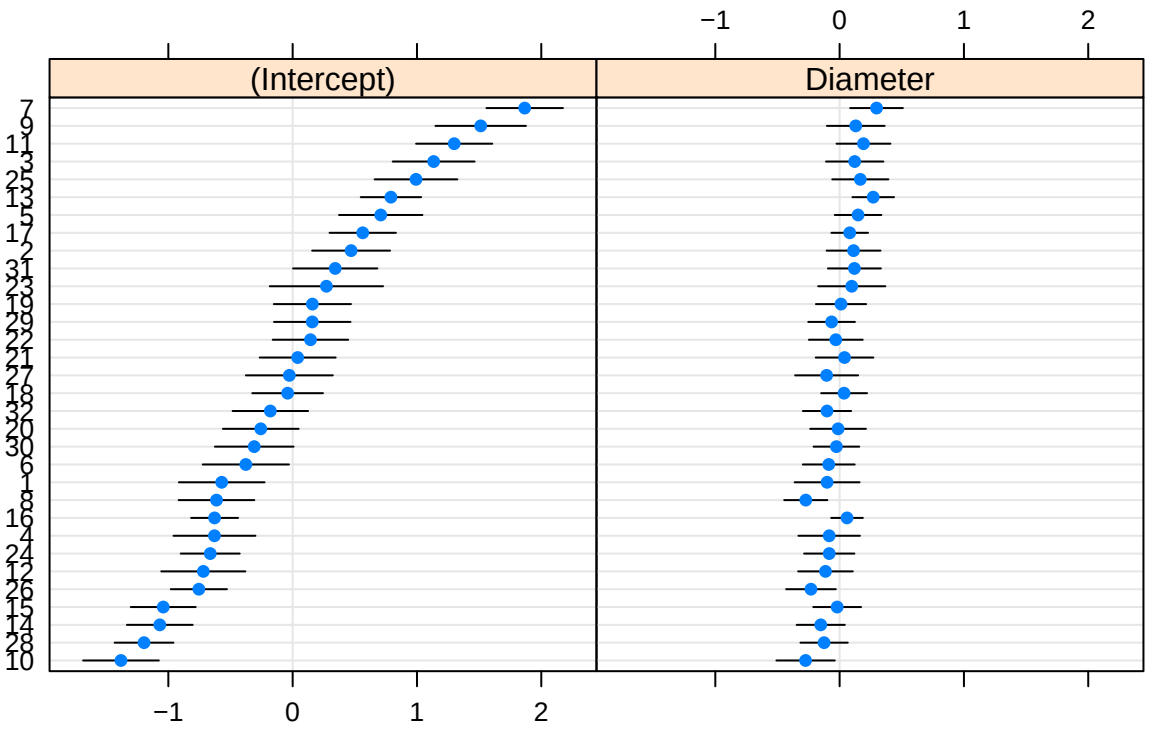


b) An assessment of how much variation there is in the group-level random effects.

Visualize random effects by transect

```
dotplot(ranef(m4.b))
```

```
## $Transect
```




```
summary(m4.b)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: CharHt ~ Steepness + Diameter + (1 + Diameter | Transect)
##    Data: char
##
##      AIC      BIC   logLik deviance df.resid
##   1822.3   1854.8   -904.1   1808.3      769
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1881 -0.5811 -0.0371  0.5626  4.1202
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Transect (Intercept) 0.66797  0.8173
##           Diameter    0.03154  0.1776   0.69
##  Residual             0.51292  0.7162
## Number of obs: 776, groups:  Transect, 32
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  1.97580    0.14728  13.415
## Steepness    0.11894    0.04445   2.676
## Diameter     0.23078    0.04514   5.113
##
## Correlation of Fixed Effects:
##           (Intr) Stpnss
## Steepness -0.023
```

```
## Diameter   0.481  0.010
```

There's a ton of variation among random intercepts, which is reflected in relatively high variance attributed to "Intercept" in the random effects table in the summary (0.67). Much less variation attributable to variation in the slope for Diameter (0.03). The total amount of residual variation explained by random intercepts is thus 0.67/(0.67+0.03+0.51) = 55%, and by random slopes is 2%.

c) A brief assessment of how well your selected model fits the data.

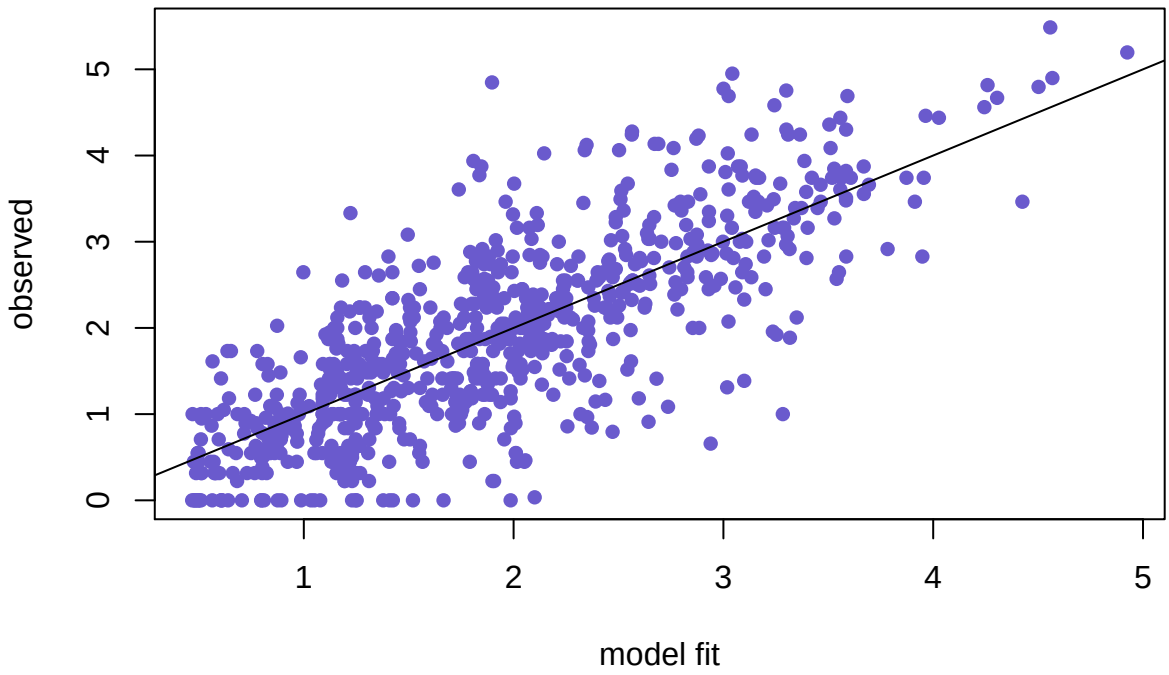
We can check conditional R2 for a single-number summary that reflects model fit. We can also plot predicted values vs observed. To make this more stringent, we could hold out values and predict them, and evaluate mean squared (or absolute) predictive error. For now I'll just check model fit.

```
r.squaredGLMM(m4.b)
```

```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help
## page.
```

```
##             R2m      R2c
## [1,] 0.05189772 0.598889
```

```
plot(char$CharHt~fitted(m4.b), pch=16, col="slateblue", ylab="observed", xlab="model fit"); abline(0,1)
```



Including the random effects, the model does a decent job fitting the data (conditional R2 ~ 0.60; scatterplot). The fit isn't obviously biased or underfitted. The residuals have standard deviation 0.7, so ~95% of the model fitted values are within ~1.4 meters of the observed values. Is that bad? Depends, I guess on context but here given all the variation in a wildfire, it seems not bad.

On the other hand, the fixed effects are not explaining that much (marginal R2 ~0.52), so that's a bit of a scientific disappointment.

Question 2

Let's assume the question is: which is the bigger source of variation in the scores for skating programs, the judges or the skating pair? Fit a mixed model for these data with "score" as the response variable, and random

effects for judge and skating pair ("judge" and "pair"). Interpret the results (coefficients and their standard errors, standard errors of the random effects). Is there a judge that tends to give consistently higher scores?

Load and explore data

```
oly <- read.csv("olympics.csv")
str(oly)
```

```
## 'data.frame':    49 obs. of  4 variables:
##  $ X    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ judge: int  1 1 1 1 1 1 1 2 2 2 ...
##  $ pair : int  1 2 3 4 5 6 7 1 2 3 ...
##  $ score: num  5.6 5.5 6 5.6 4.8 4.8 4.3 5.5 5.7 5.5 ...
```

```
head(oly)
```

```
##   X judge pair score
## 1 1     1    1   5.6
## 2 2     1    2   5.5
## 3 3     1    3   6.0
## 4 4     1    4   5.6
## 5 5     1    5   4.8
## 6 6     1    6   4.8
```

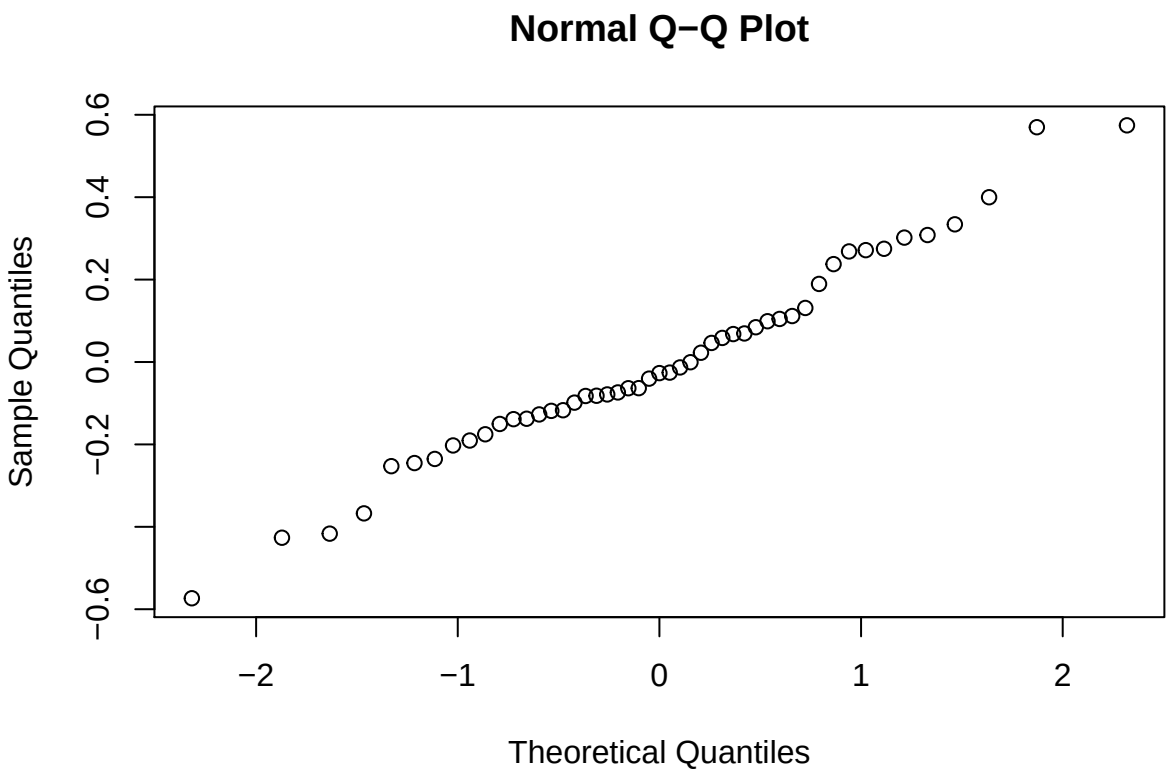
Fit a model

```
m.oly <- lmer(score~(1|pair) + (1|judge), data = oly)
display(m.oly)
```

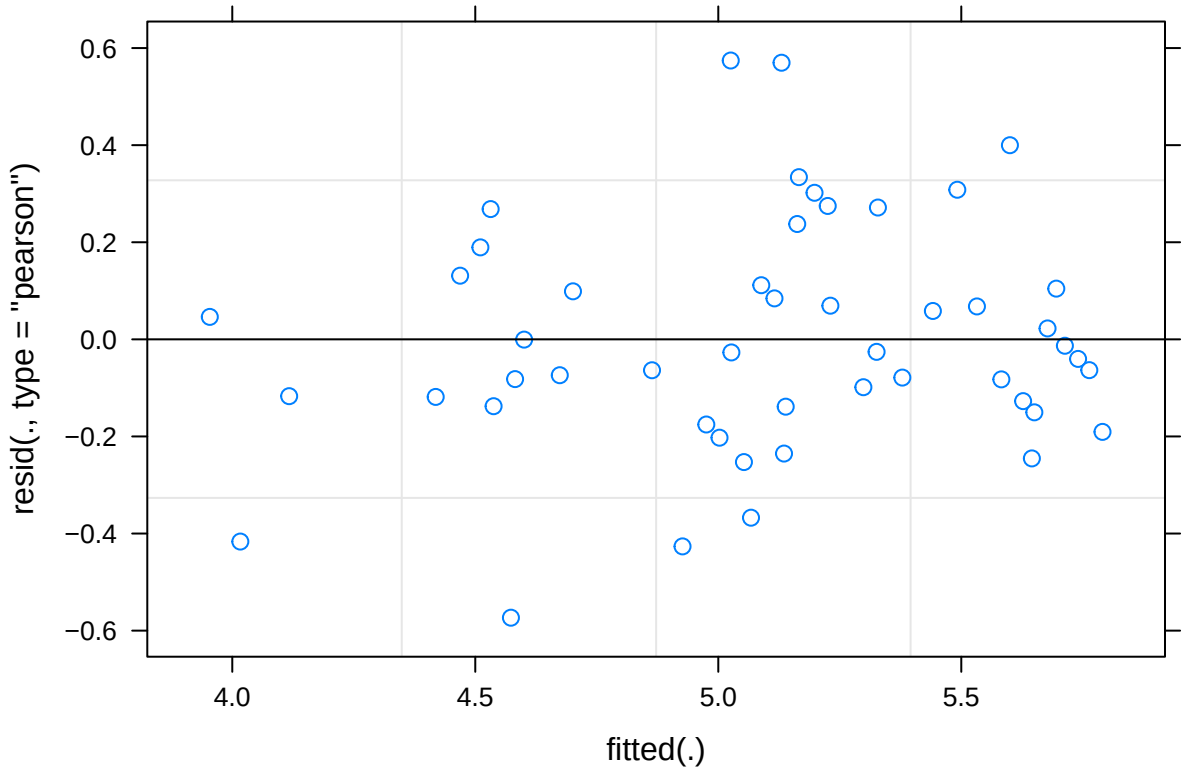
```
## lmer(formula = score ~ (1 | pair) + (1 | judge), data = oly)
## coef.est  coef.se
##     5.09     0.20
##
## Error terms:
##  Groups   Name        Std.Dev.
##  pair     (Intercept) 0.45
##  judge    (Intercept) 0.28
##  Residual             0.27
## ---
## number of obs: 49, groups: pair, 7; judge, 7
## AIC = 54.2, DIC = 43.4
## deviance = 44.8
```

The standard deviation associated with the pair factor is larger than that for judge, indicating that there's more variance explained or accounted for by skating pair than judge identity. This seems like a good thing. Note this is an example in which the two factors are fully crossed, so crossed random effects are appropriate.

```
qqnorm(resid(m.oly))
```

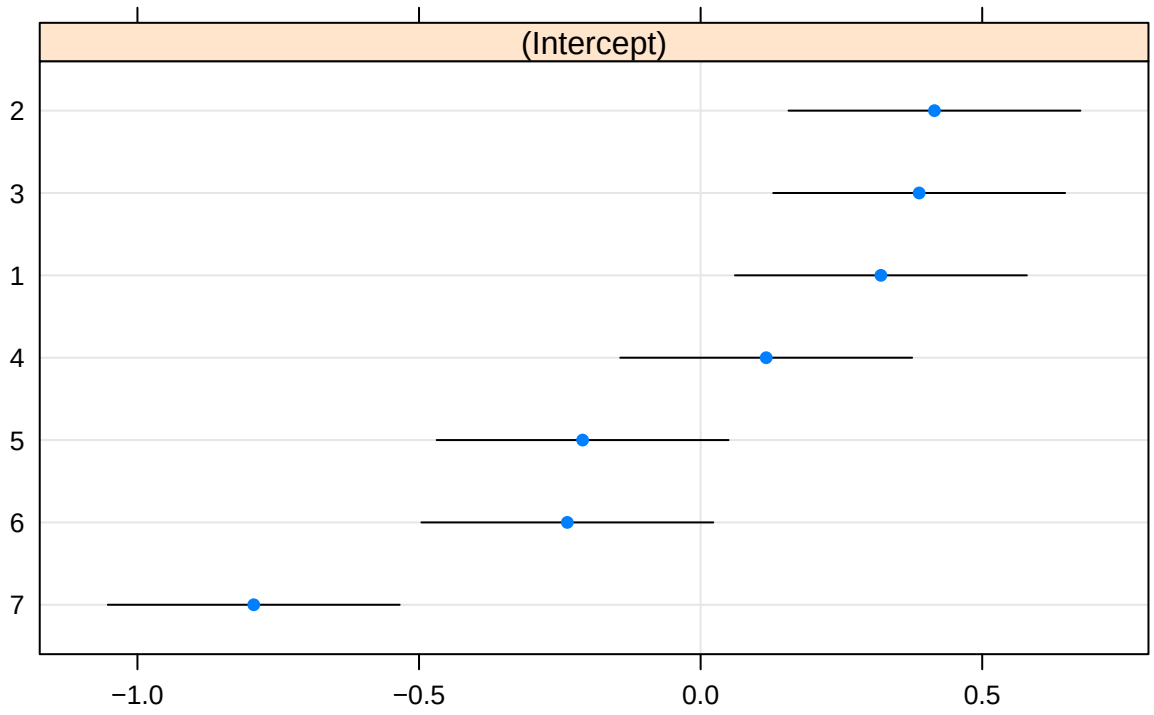



```
plot(m.oly)
```

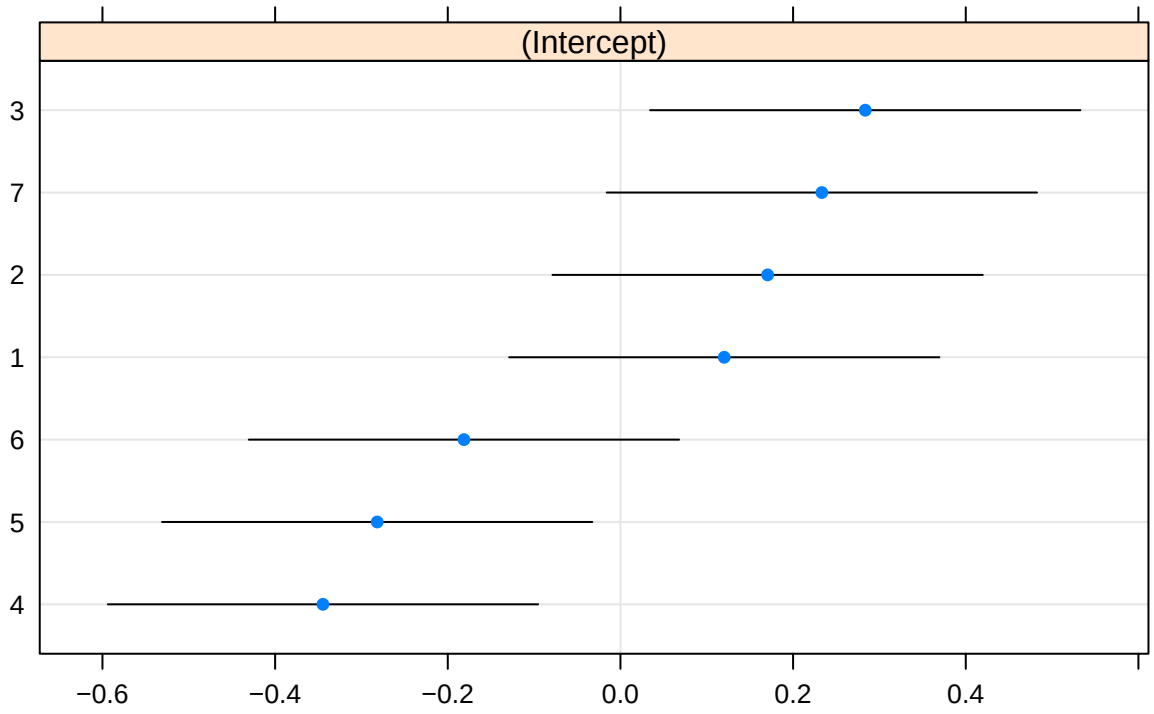



```
dotplot(ranef(m.oly))
```

```
## $pair
```


pair
(Intercept)
2
3
1
4
5
6
7
−1.0
−0.5
0.0
0.5

```
##
## $judge
```


judge
(Intercept)
3
7
2
1
6
5
4
−0.6
−0.4
−0.2
0.0
0.2
0.4

Judge 3 has the highest mean score. People disagreed whether this means this judge is scoring "consistently higher" as the question asks. There's pretty strong evidence this judge overall gives highest scores, and substantially higher than at least 5 and 4. But there is overlap in confidence intervals and in score rankings across pairs are variable, so is this "consistent"? Well, you be the judge :)