ST2304 Solutions Week 8: Categorical Variables

Bob O'Hara

Problem 1

First, get the cake data

```
library(lme4) # use install.packages("lme4") if this doesn't work
```

```
## Loading required package: Matrix
```

```
data("cake")
contrasts(cake$temperature) <- contr.treatment(nlevels(cake$temperature))
```

Using the cake data, adapt the code above to look at the effect of replicate (i.e. date) on the angle, treating this as a factor. How does the angle vary with date (the replicates are recorded in order, so higher replicates were baked later)?

Fitting the model is straightforward:

```
mod.replicate <- lm(angle~replicate, data=cake)
round(coef(mod.replicate), 1)
```

```
## (Intercept)  replicate2  replicate3  replicate4  replicate5  replicate6
##        46.8        -1.3        -9.9       -13.6       -14.4       -18.1
##  replicate7  replicate8  replicate9 replicate10 replicate11 replicate12
##       -19.5       -19.4       -19.5       -18.0       -16.9       -15.9
## replicate13 replicate14 replicate15
##       -14.9       -18.9       -20.2
```

Just looking at the coefficients, we can see that they seem to get more negative, i.e. the angle is less than in the first replicate, where it is 46.8°, and in the last where it is 26.6°, i.e. it is roughly half. We can see this more clearly when it is plotted in Fig. 1 (note that this isn't a statistical graphics course, so I don't expect great plots!).

```
# Bind the coefficients and the confidence intervals together
# the first row is the intercept, which we can remove, so we can look at how the effect changes
# (i.e. the absolute value doesn't matter, just the differnces from any one level)
RepEffs <- cbind(coef=coef(mod.replicate), confint(mod.replicate))[-1,]

plot(1:nrow(RepEffs), RepEffs[,"coef"], ylim=range(RepEffs), xlab="Replicate", ylab="Contrast")
segments(1:nrow(RepEffs), RepEffs[,"2.5 %"], 1:nrow(RepEffs), RepEffs[,"97.5 %"])
```

The angle decreases over the first few replicates and then remains roughly the same. I would guess that this is due to the students learning how to best bake a cake.

How well does date explain the variation in the angle? We are not really interested in date, so what does this suggest about the model we should be fitting?

The R^2 for the model is 56%, so it explains a lot of the variation in the data. Although we are not interested in the effects of replicate, we have to acknowledge that it has an effect. We should include it in a final model: it will give us a model that better explains the data, and will make the estimates of the other effects

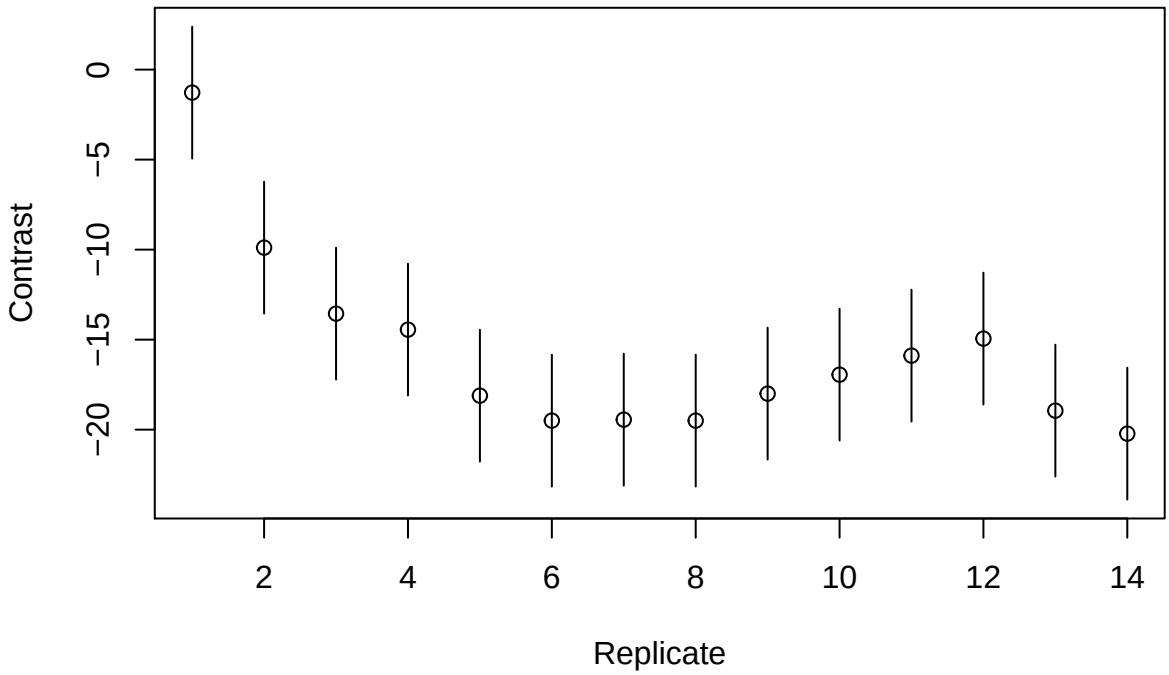

Figure 1: Estimated Effects of Replicate

better (i.e. have a lower standard error), because there is less unexplained variation, and it is the unexplained variation that makes the parameter estimates uncertain.

Does the model fit the data? Do the residuals look OK (are they normally distributed, are there outliers, is the variance constant etc.)?

Basically, yes fits well: see Fig. 2. There don't seem to be any outliers, the normal probability plot is pleasingly linear, and all of the Cook's D values are low.

```
cake$fitted <- fitted(mod.replicate)
cake$resid <- resid(mod.replicate)
cake$CooksD <- cooks.distance(mod.replicate)

par(mfrow=c(1,3), mar=c(4,3,5.5,1))
plot(cake$fitted, cake$resid, type="p", xlab="Fitted Values",
     main="Residuals vs Fitted")
qqnorm(cake$resid, main="Normal Probability Plot");
qqline(cake$resid)
plot(cake$fitted, cake$CooksD, xlab="Fitted Values", main="Cook' D")
```

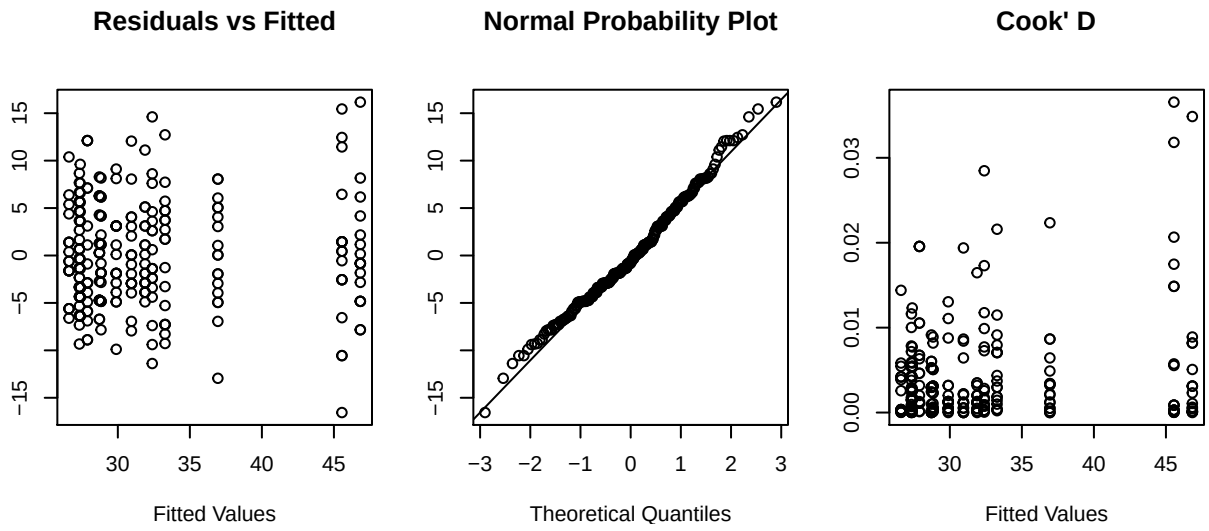

Figure 2: Plots for checking clutch size model

Problem 2

Fit a model with both temperature and recipe explaining the angle: the code to do this works the same way as the code for multiple regression. How much of the variation in the data is explained by these factors? What are the effects of temperature and recipe?

The model can be fitted like this:

```
mod.tr <- lm(angle~recipe + temperature, data=cake)
```

This model explains 12% of the data: this is a lot less than just the effect of Replicate. The effects are plotted in Fig. 3. We can see that there do not seem to be much difference in recipes, but increasing the temperature increases the angle.

```
TempRecEffs <- cbind(coef=coef(mod.tr), confint(mod.tr))[-1,]
rownames(TempRecEffs) <- gsub("recipe", "Rec. ", rownames(TempRecEffs))
rownames(TempRecEffs) <- gsub("temperature", "Temp. ", rownames(TempRecEffs))
Temps <- c(175, 185,195, 205, 215, 225)


plot(1:nrow(TempRecEffs), TempRecEffs[,"coef"],
     ylim=range(TempRecEffs), xlab="Replicate", ylab="Contrast", xaxt="n")
segments(1:nrow(TempRecEffs), TempRecEffs[,"2.5 %"],
         1:nrow(TempRecEffs), TempRecEffs[,"97.5 %"])
axis(1, gsub("Recipe", "Rec.", rownames(TempRecEffs)), at=1:nrow(TempRecEffs))
abline(h=0, lty=3)
```

Now add replicate (i.e. date) as an extra effect. Does this change the estimates of the effects of temperature and recipe? How about the uncertainty in these effects (i.e. the standard error)? Does this model explain more of the data?

We can fit the new model:

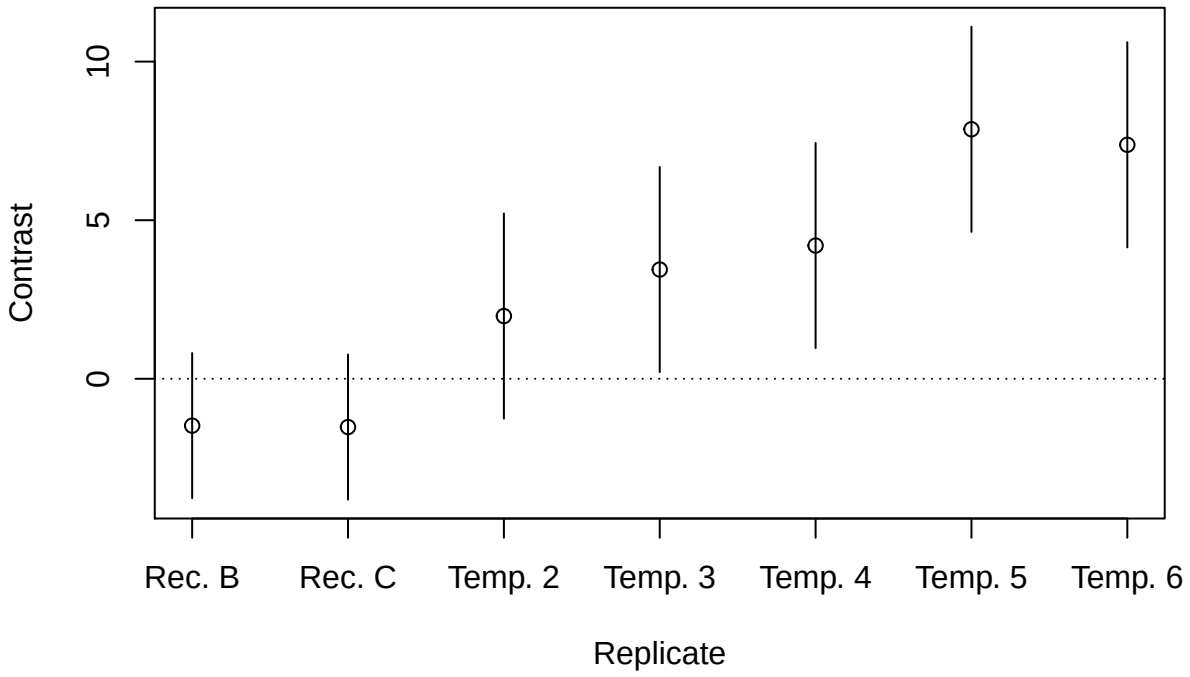

Figure 3: Estimated effects of recipe and temperature

```
mod.trr <- lm(angle~recipe + temperature + replicate, data=cake)
round(coef(mod.replicate), 1)
```

```
## (Intercept)  replicate2  replicate3  replicate4  replicate5  replicate6
##        46.8        -1.3        -9.9       -13.6       -14.4       -18.1
##  replicate7  replicate8  replicate9 replicate10 replicate11 replicate12
##       -19.5       -19.4       -19.5       -18.0       -16.9       -15.9
## replicate13 replicate14 replicate15
##       -14.9       -18.9       -20.2
```

We can compare the parameter estimates, in Figure 4. The point estimates are exactly the same, but when we have Replicate in the model, the standard errors (and hence confidence intervals) are smaller. It also explains more variation in the data: without replicate the R^2 is 12%, with it the R^2 is 69%.

```
# compare mod.trr to mod.tr
Effs.tr <- cbind(coef=coef(mod.tr), confint(mod.tr))[-1,]
Effs.trr <- cbind(coef=coef(mod.trr), confint(mod.trr))
Effs.trr <- Effs.trr[rownames(Effs.trr)%in%rownames(Effs.tr),]

plot((1:nrow(Effs.tr))-0.1, Effs.tr[,"coef"],
     ylim=range(Effs.tr), xlab="Replicate", ylab="Contrast", xaxt="n")
segments((1:nrow(Effs.tr))-0.1, Effs.tr[,"2.5 %"],
         (1:nrow(Effs.tr))-0.1, Effs.tr[,"97.5 %"])
points((1:nrow(Effs.trr))+0.1, Effs.trr[,"coef"], col=2)
segments((1:nrow(Effs.trr))+0.1, Effs.trr[,"2.5 %"],
         (1:nrow(Effs.trr))+0.1, Effs.trr[,"97.5 %"], col=2)

axis(1, gsub("Recipe", "Rec.", rownames(TempRecEffs)), at=1:nrow(TempRecEffs))
```

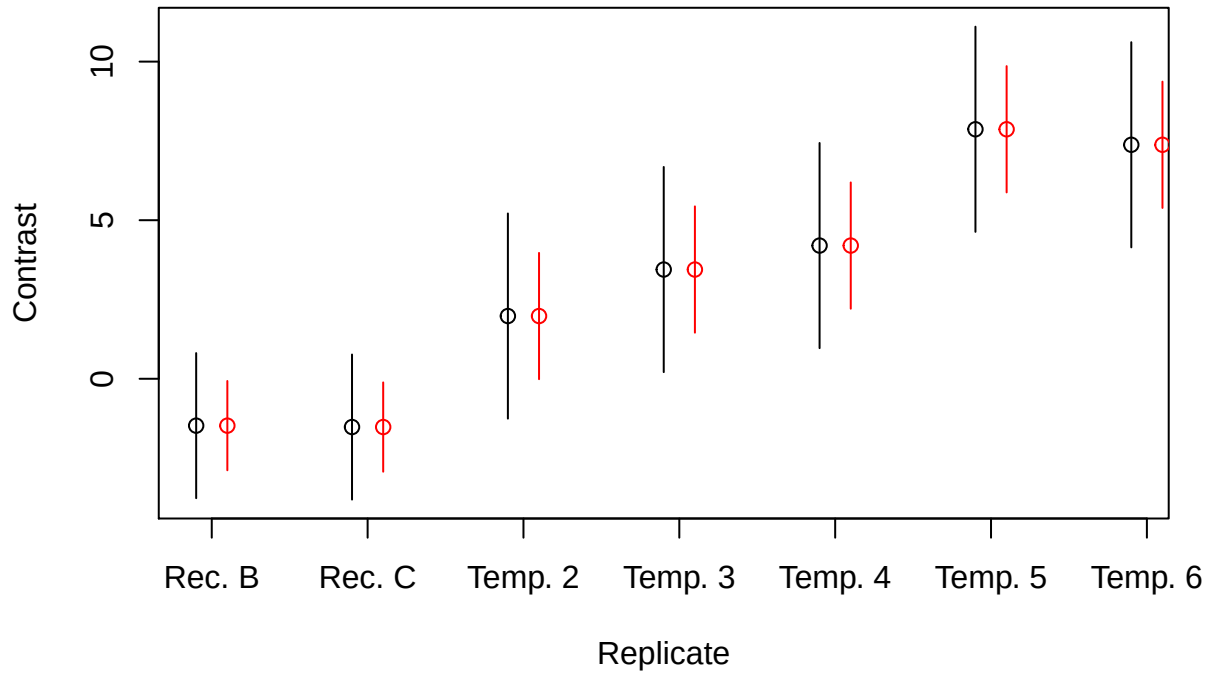

Figure 4: Estimated effects of recipe and temperature, with (red) and without (black) a replicae effect in the model

How good is the model fit (residuals etc)?

Not surprisingly, they look good (Fig. 5),

```
cake$fitted2 <- fitted(mod.trr)
cake$resid2 <- resid(mod.trr)
cake$CooksD2 <- cooks.distance(mod.trr)

par(mfrow=c(1,3), mar=c(4,3,5.5,1))
plot(cake$fitted2, cake$resid2, type="p", xlab="Fitted Values",
     main="Residuals vs Fitted")
qqnorm(cake$resid2, main="Normal Probability Plot");
qqline(cake$resid2)
plot(cake$fitted2, cake$CooksD2, xlab="Fitted Values", main="Cook' D")
```

Although we are not interested in the replicate effect, is it still worth including in the model? Can you comment more generally on what this means for designing (and analysing) your own experiments, and how you can deal with sources of variation that are difficult to control?

Yes, it is worth adding the replicate. It reduces the uncertainty in the parameter estimates, and so increases our confidence that we can explain the cake bending.

More generally, when we are designing experiments, we ewant to control as much variation as possible. We can do this by controlling the conditions (e.g. temperature), but some times this is not possible. For exampe, here there is an effect of time that cannot be controlled. So instead it is included in the experiment, in the design. Note that in every replicate, all combinations of recipe and temperature are tried. This makes the model nicer: it is why the point estimates are the same whether the replicate is included or not (technically,

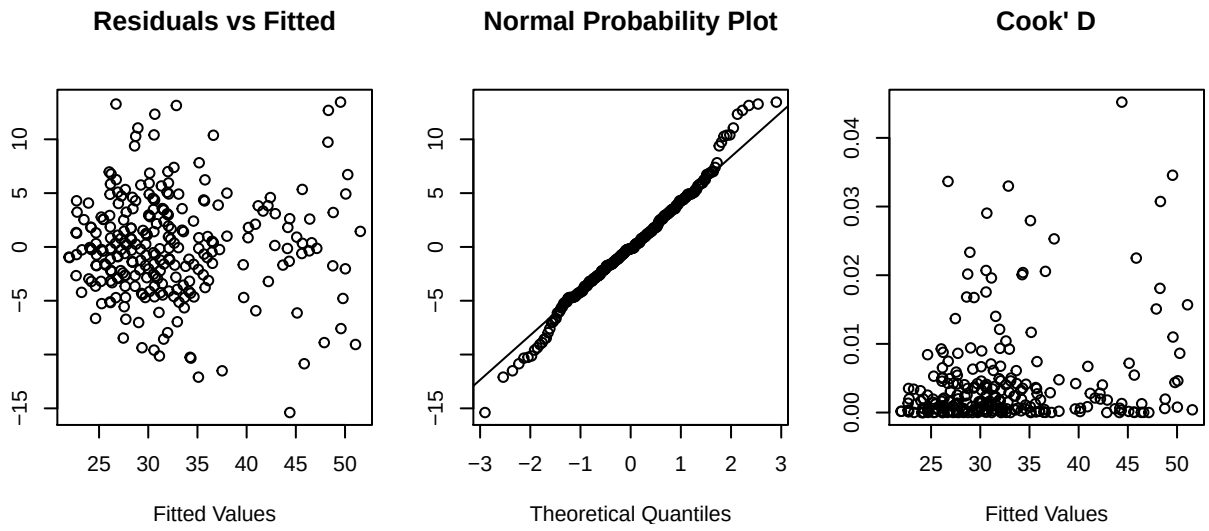

Figure 5: Plots for checking clutch size model

this is called a *balanced design*). Design of experiments is an important area of statistics, because we want to get good estimates of the prameters that we are intersted in, whilst controlling for whatever other factors might be having an effect. The analyses in this exercise are the same form as are used for many experimental design problems.