Advanced Lab 4: More Linear Modeling and ANOVA

Introduction: Multiple predictors

We will now start considering more complex models, with more than one predictor. We will start with a model including all the predictors:

```
fitAll <- lm(time~., data=targetingFinal)
summary(fitAll)
```

Let's take a closer look at this model. We have:

- the intercept
- a term for the subject ID. We should probably remove this.
- iat
- a factor level for male gender. Presumably the female case is the base case.
- factor levels for age, with 18 being the base case.
- a factor level for race being white. Presumably the base case is race being black.
- a factor level for the weapon value being set to unarmed, with base case being armed.
- a factor level for action incorrect, with base case being the correct action.

Calling anova on the model will tell us about each factor as a whole, and whether its contribution is statistically significant:

```
anova(fitAll)
```

We see that the contribution with ther largest P-value, and hence smaller effect, is from the subject ID. This is good, there is no reason whatsoever that the subject ID should have anything to do with the data. Let's remove it and reconsider:

```
fitNoSubj <- lm(time~.-subject, data=targetingFinal)
summary(fitNoSubj)
anova(fitNoSubj)
```

We can see that the residual standard error did not change almost at all with the removal of the subject variable. We see a number of other variables that are potentially not influential.

One powerful utility that R offers is the ability to remove one variable at a time from a model, and consider all the resulting smaller models. The drop1 method is one way to proceed for that:

```
drop1(fitNoSubj)
```

What we see is the effect of removing each of the variables, in terms of how much the RSS will change. The AIC column portrays the "Akaike Information Criterion". The AIC is a number that takes into consideration the RSS but also penalizes models based on the number of parameters used. It is technically given by the formula:

$$-2\text{log-likelihood} + 2p$$

A larger model will fit the data better, and so will have a larger log-likelihood and therefore a smaller value for -2log-likelihood. But it will also have more parameters, so $2p$ will be larger. The formula aims to balanced the two: The smaller the AIC the more you have gained by the bigger model, while also accounting for how bigger the model was. A model with smaller AIC is fitting the data "more

economically". There is often an indeterminate additive constant involved in AIC computations, so they are only appropriate for comparisons between models.

The first line, marked <none>, contains the full model, while the remaining lines indicate the effect with the corresponding predictor removed.

In this case we see that the biggest gain in AIC can be obtained by dropping the age variable. This would make sense since that variable, coded as a factor variable, uses 5 parameters. Let's remove it and consider the model again:

```
fitNoSubjNoAge <- fitNoSubj %>% update(formula=.~.-age)
summary(fitNoSubjNoAge)
anova(fitNoSubjNoAge)
drop1(fitNoSubjNoAge)
```

We now see race as a next possible variable to remove.

```
fitNoSubjNoAgeNoRace <- fitNoSubjNoAge %>% update(formula=.~.-race)
summary(fitNoSubjNoAgeNoRace)
anova(fitNoSubjNoAgeNoRace)
drop1(fitNoSubjNoAgeNoRace)
```

Removing gender seems to be the next step:

```
fitNoSubjNoAgeNoRaceNoGender <- fitNoSubjNoAgeNoRace %>% update(formula=.~.-gender)
summary(fitNoSubjNoAgeNoRaceNoGender)
anova(fitNoSubjNoAgeNoRaceNoGender)
drop1(fitNoSubjNoAgeNoRaceNoGender)
```

We now have a model where removing a variable does not provide an AIC improvement.

We can also consider adding models in, one at a time:

```
add1(fitNoSubjNoAgeNoRaceNoGender, scope=~.+race+gender+age)
```

We see that none of them is an improvement.

It is clear however that the two most important effects are due to whether the target was armed or unarmed, and whether the action was correct or incorrect. Let's review some graphs that break the cases down by these two variables:

```
pBasic <- ggplot(targetingFinal) +
    aes(y=time) +
    facet_grid(weapon~action)

pBasic + aes(x=race) + geom_boxplot()
pBasic + aes(x=race, group=subject) + geom_line()
```

Interaction Terms

Before moving on to more complex model techniques, let's try to add interaction terms to our model. Our initial model was fitAll , but to keep it simple let's at least remove the age and subject:

```
fitStandard <- fitAll %>% update(formula=.~.-age-subject)
summary(fitStandard)
```

We would like to consider adding some interaction terms now. For example let's try to add an interaction term that accounts for the target's race and their weapon status:

```
fitInter1 <- fitStandard %>% update(formula=.~.+ race:weapon)
summary(fitInter1)
anova(fitInter1, fitStandard)
```

We see that this model does not add anything significant. Let's try an interaction term for weapon and action:

```
fitInter2 <- fitStandard %>% update(formula=.~.+ action:weapon)
summary(fitInter2)
anova(fitInter2, fitStandard)
```

This appears to be significant! Let's try to add an interaction term between race and action:

```
fitInter3 <- fitInter2 %>% update(formula=.~.+ action:race)
summary(fitInter3)
anova(fitInter3, fitInter2)
```

That does not appear to be significant.

Before we move on, let's use the step method to tell R to perform this step-wise model selection that we performed manually earlier:

```
step(fitInter2)
```

We see that R successively removed the race and gender factors from our model, using the AIC numbers a guides. You can look at the documentation for step to learn more about its use.

```
modelFinal <- step(fitInter2)
summary(modelFinal)
```

Mixed Effects Modeling

The above techniques have not accounted at all for the nature of the distinct participants. It is reasonable to assume that each subject has a different baseline reaction time, and we have to account for that effect. The most appropriate way to model that is likely via a *random effect*, i.e. assuming that there is a constant base reaction time component to each subject, that comes from a distribution whose parameters we might need to specify. So our model for the reaction time may end up looking something like this:

$$\text{time} = \mu + \beta_1 \times \text{iat} + \beta_2 \times \text{weapon} + \cdots + \text{tsubj} + \epsilon$$

where tsubj is an effect different for each subject and drawn from a distribution $N(0, \sigma_s)$, while ϵ is still the standard error term from a $N(0, \sigma)$ distribution, and β_1, β_2 etc are still the *fixed effects* from the various factors and predictors.

For simplicity we will start with a simple case. We need a new package for this, called lme4, as mixed effects modeling goes beyond the standard linear modeling techniques. The method lmer from that package is our main workhorse. Here is one example call:

```
library(lme4)
mixedFit1 <- lmer(time ~ race + weapon + action:weapon + (1|subject), data=targetingFinal)
summary(mixedFit1)

mixedFit2 <- lmer(time ~ iat + race + weapon + action:weapon + (1|subject), data=targetingFin
mixedFit3 <- lmer(time ~ iat + weapon + action:weapon + (1|subject), data=targetingFinal)
mixedFit4 <- lmer(time ~ weapon + action:weapon + (1|subject), data=targetingFinal)
anova(mixedFit1, mixedFit2)
anova(mixedFit3, mixedFit2)
anova(mixedFit4, mixedFit3)
```

TODO