
Exercise Solutions

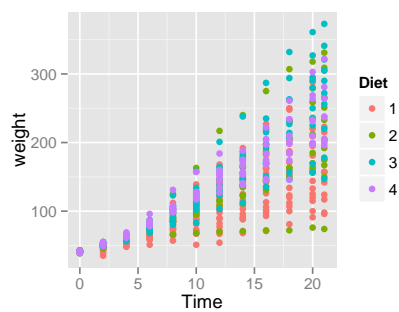
Chapter 1

```
> require(ggplot2)
```

1. The `ChickWeight` data set contains data on the effect of diet on early growth of chicks (?`ChickWeight` for more details). Use this data set to practice plotting longitudinal data:

- (a) Make a scatterplot that shows weights of individual chicks as a function of time and diet.

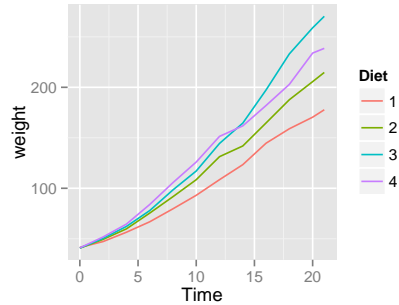
```
> ggplot(ChickWeight, aes(Time, weight, color=Diet)) +  
  geom_point()
```



- (b) Make a summary plot that shows average weight over time for each diet.

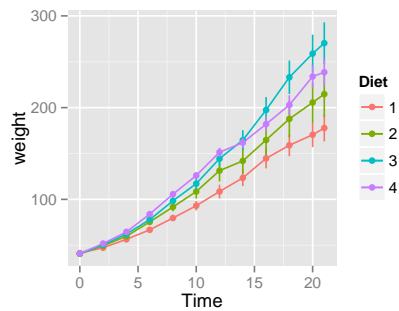
```
> ggplot(ChickWeight, aes(Time, weight, color=Diet)) +  
  stat_summary(fun.y=mean, geom="line")
```

2



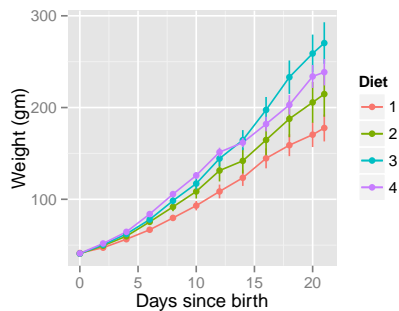
(c) Add an indicator of standard error to the averages.

```
> ggplot(ChickWeight, aes(Time, weight, color=Diet)) +  
  stat_summary(fun.y=mean, geom="line") +  
  stat_summary(fun.data=mean_se, geom="pointrange")
```



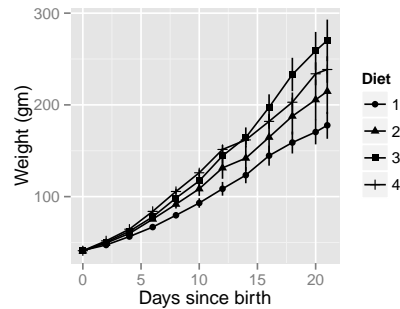
(d) Customize the plot with more informative axis labels.

```
> ggplot(ChickWeight, aes(Time, weight, color=Diet)) +  
  stat_summary(fun.y=mean, geom="line") +  
  stat_summary(fun.data=mean_se, geom="pointrange") +  
  labs(x="Days since birth", y="Weight (gm)")
```



(e) Make color and black-and-white versions of the plot.

```
> #use shapes to distinguish diets instead of colors
> ggplot(ChickWeight, aes(Time, weight, shape=Diet)) +
  stat_summary(fun.y=mean, geom="line") +
  stat_summary(fun.data=mean_se, geom="pointrange") +
  labs(x="Days since birth", y="Weight (gm)")
```



- (f) Use `ggsave` to export the plot as an image file and specify an image size and resolution.

```
> ggsave("ChickWeight.pdf", width=4, height=3, dpi=300)
```

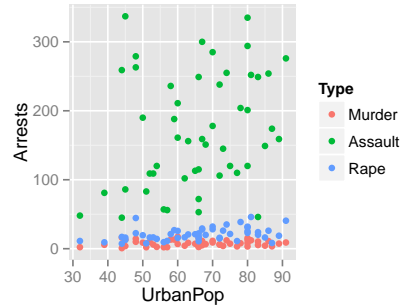
2. The `USArrests` data set contains violent crime arrests (per 100,000 residents) in each of the 50 states in the USA in 1973 and the percent of the population of each state that lived in urban areas (`?USArrests` for more details on this data set, and try `?state` for other information about US states).

- (a) Convert the `USArrests` data set from a wide to a long format so that instead of separate variables for each crime type (Murder, Assault, Rape), there is one variable that identifies the crime type and one variable that contains the rates for each crime type for each state.

```
> require(reshape2)
> USArrests.m <- melt(USArrests, id="UrbanPop",
  variable.name="Type",
  value.name="Arrests")
```

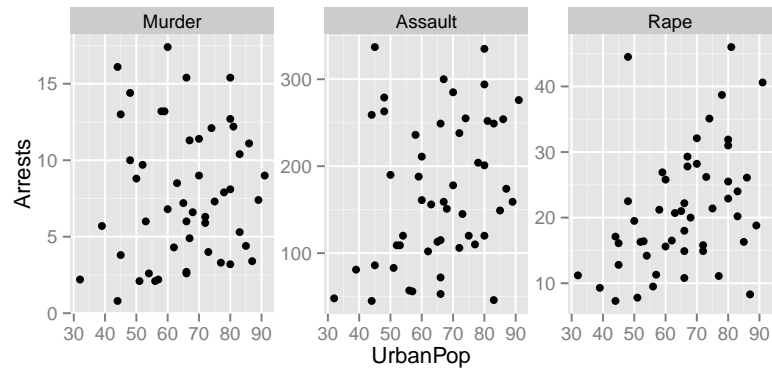
- (b) Make a scatterplot showing the relationship between each type of violent crime rate and percent of population living in urban areas.

```
> ggplot(USArrests.m,
  aes(UrbanPop, Arrests, color=Type)) +
  geom_point()
```



- (c) Plot the violent crime types in separate panels (tip: try using the `scales` and `nrow` or `ncol` options to customize the panels).

```
> ggplot(USArrests.m, aes(UrbanPop, Arrests)) +
  facet_wrap(~ Type, scales="free") + geom_point()
```



Chapter 2

The `wisqars.suicide` data frame contains annual suicide rate data by state from 1999 to 2007 collected from the Web-based Injury Statistics Query and Reporting System (WISQARS) hosted by the Centers for Disease Control and Prevention. Use these data to analyze trends in suicide rates by geographic region (tip: adjust the `Year` variable to treat 1999 as time 0 so that it corresponds to the intercept).

1. Did the overall suicide rate in the US increase during this time period? If yes, what was the estimated rate of change?

```
> require(lme4)
```

```

> load("Examples.Rdata")
> wisqars.suicide$Time <- wisqars.suicide$Year - 1999
> m.base <- lmer(Crude.Rate ~ Time + (Time | State),
                data = wisqars.suicide, REML=F)
> coef(summary(m.base))

```

	Estimate	Std. Error	t value
(Intercept)	11.75132	0.438836	26.7784
Time	0.12804	0.020554	6.2294

Yes, crude rate of suicide increased by about 0.128 per year.

2. Did the regions differ in their initial (1999) suicide rates?

```

> m.0 <- lmer(Crude.Rate ~ Time + Region + (Time | State),
              data = wisqars.suicide, REML=F)
> anova(m.base, m.0)

```

Data: wisqars.suicide
Models:

	Df	AIC	BIC	logLik	deviance	Chisq	Chi	Df	Pr(>Chisq)
m.base: Crude.Rate ~ Time + (Time State)									
m.0: Crude.Rate ~ Time + Region + (Time State)									
m.base	6	1426	1451	-707	1414				
m.0	9	1400	1437	-691	1382	32.4	3		4.3e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Yes, baseline (1999) crude rate of suicide differed between regions.

3. Did the regions differ in their rate of change of suicide rate during this period?

```

> m.1 <- lmer(Crude.Rate ~ Time * Region + (Time | State),
              data = wisqars.suicide, REML=F)
> anova(m.base, m.0, m.1)

```

Data: wisqars.suicide
Models:

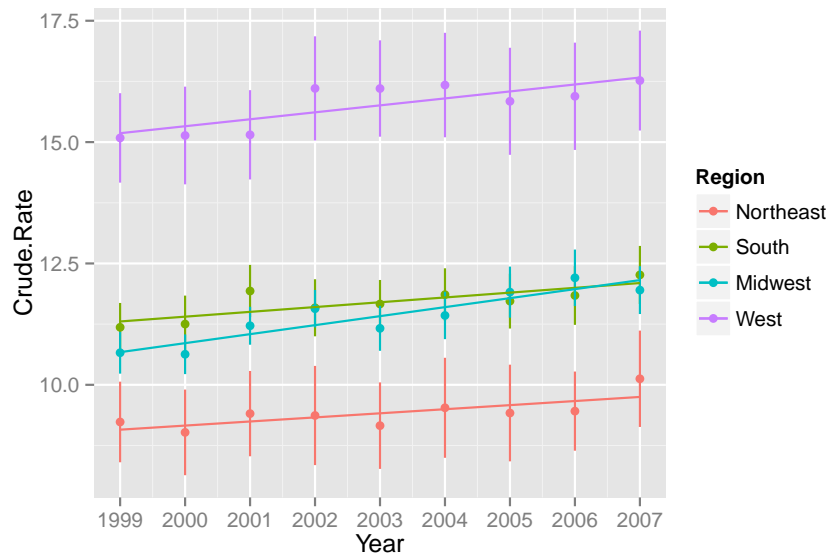
	Df	AIC	BIC	logLik	deviance	Chisq	Chi	Df	Pr(>Chisq)
m.base: Crude.Rate ~ Time + (Time State)									
m.0: Crude.Rate ~ Time + Region + (Time State)									
m.1: Crude.Rate ~ Time * Region + (Time State)									
m.base	6	1426	1451	-707	1414				
m.0	9	1400	1437	-691	1382	32.43	3		4.3e-07 ***
m.1	12	1402	1452	-689	1378	3.57	3		0.31

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

No, the evidence does not support overall differences in rate of change in suicide rate during this period.

- Plot the observed and model-fit suicide rates by region for this time period. Include an indicator of variability (e.g., standard error) for the observed data. Make color and black-and-white versions.

```
> require(ggplot2)
> ggplot(wisqars.suicide, aes(Year, Crude.Rate, color=Region)) +
  stat_summary(fun.data=mean_se, geom="pointrange") +
  stat_summary(aes(y=fitted(m.1)), fun.y=mean, geom="line") +
  scale_x_continuous(breaks=1999:2007)
```



Chapter 3

The CP data frame contains auditory discrimination data (d' , called “d prime”) for two continua of eight stimuli. The continua were created by morphing between two sounds from different categories, either along a temporal acoustic dimension or along a spectral acoustic dimension. The hypothesis was that there would be “categorical perception” – better discrimination near the category boundary than near the endpoints – for the temporal dimension but not for the spectral dimension (Mirman, Holt, & McClelland, 2004).

1. Analyze these data using growth curve analysis with second-order orthogonal polynomials. Which polynomial terms show statistically significant effects of continuum type?

```

> load("Examples.Rdata")
> summary(CP)

  Participant      Type      Stimulus      d.prime
1      : 8  Temporal:128  Min.   :1.00  Min.   :-2.17
2      : 8  Spectral:128  1st Qu.:2.75  1st Qu.: 0.40
3      : 8                      Median :4.50  Median : 1.16
4      : 8                      Mean   :4.50  Mean   : 1.28
5      : 8                      3rd Qu.:6.25  3rd Qu.: 2.22
6      : 8                      Max.   :8.00  Max.   : 4.77
(Other):208

> #make second-order orthogonal polynomial
> s <- poly(unique(CP$Stimulus), 2)
> #add it to data frame
> CP[, paste("ostim", 1:2, sep="")] <- s[CP$Stimulus, 1:2]
> ##fit the models
> #base model has no effects of continuum type
> m.base <- lmer(d.prime ~ (ostim1 + ostim2) +
                (ostim1 + ostim2 | Participant),
                data=CP, REML=FALSE)
> #add effect of continuum type on intercept
> m.0 <- lmer(d.prime ~ (ostim1 + ostim2) + Type +
              (ostim1 + ostim2 | Participant),
              data=CP, REML=FALSE)
> #add effect of continuum type on linear term
> m.1 <- lmer(d.prime ~ (ostim1 + ostim2) + Type +
              ostim1:Type + (ostim1 + ostim2 | Participant),
              data=CP, REML=FALSE)
> #add effect of continuum type on quadratic term (full model)
> m.cp <- lmer(d.prime ~ (ostim1 + ostim2)*Type +
              (ostim1 + ostim2 | Participant),
              data=CP, REML=FALSE)
> anova(m.base, m.0, m.1, m.cp)

Data: CP
Models:
m.base: d.prime ~ (ostim1 + ostim2) + (ostim1 + ostim2 | Participant)
m.0: d.prime ~ (ostim1 + ostim2) + Type + (ostim1 + ostim2 | Participant)
m.1: d.prime ~ (ostim1 + ostim2) + Type + ostim1:Type + (ostim1 +
m.1:      ostim2 | Participant)
m.cp: d.prime ~ (ostim1 + ostim2) * Type + (ostim1 + ostim2 | Participant)
      Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)

```

```

m.base 10 850 886 -415      830
m.0    11 852 891 -415      830 0.76      1      0.38
m.1    12 853 896 -415      829 0.02      1      0.88
m.cp   13 849 895 -411      823 6.55      1      0.01 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

There is a statistically significant effect of continuum type on the quadratic term.

2. Estimate parameter-specific p -values using the normal distribution. How does this evaluation of the effects of continuum type compare with the model comparisons approach? Repeat using `lmerTest`.

```

> #normal approximation
> coefs <- as.data.frame(coef(summary(m.cp)))
> coefs$p <- 2*(1-pnorm(abs(coefs[, 3])))
> coefs

              Estimate Std. Error t value      p
(Intercept)    1.35466    0.10593 12.7886 0.0000e+00
ostim1         -0.43154    0.28754 -1.5008 1.3341e-01
ostim2         -1.80192    0.41369 -4.3558 1.3261e-05
TypeSpectral    -0.15378    0.14980 -1.0265 3.0463e-01
ostim1:TypeSpectral 0.11268    0.40664  0.2771 7.8171e-01
ostim2:TypeSpectral 1.57713    0.58504  2.6958 7.0227e-03

> #Satterthwaite approximation using lmerTest
> require(lmerTest)
> m.cp.lT <- lmer(d.prime ~ (ostim1 + ostim2)*Type +
                  (ostim1 + ostim2 | Participant),
                  data=CP, REML=FALSE)
> coef(summary(m.cp.lT))

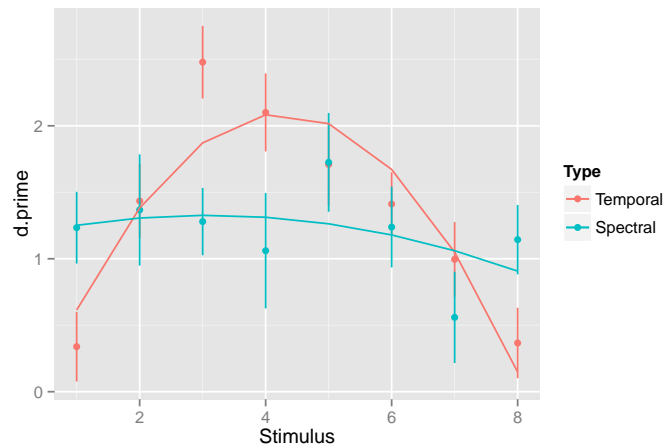
              Estimate Std. Error t value  Pr(>|t|)
(Intercept)    1.35466    0.10593 12.7886 1.8874e-14
ostim1         -0.43154    0.28754 -1.5008 1.3520e-01
ostim2         -1.80192    0.41369 -4.3558 1.2760e-04
TypeSpectral    -0.15378    0.14980 -1.0265 3.1198e-01
ostim1:TypeSpectral 0.11268    0.40664  0.2771 7.8203e-01
ostim2:TypeSpectral 1.57713    0.58504  2.6958 1.1099e-02

```

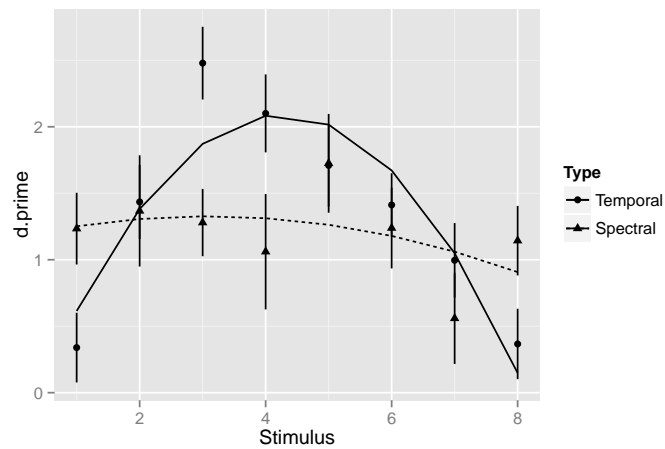
Pattern of results is the same (effect of continuum type on the quadratic term), but the normal approximation produces smaller p -values.

3. Plot the observed and model-fit d' by continuum. Include an indicator of variability (e.g., standard error) for the observed data. Make color and black-and-white versions.


```
> ggplot(CP, aes(Stimulus, d.prime, color=Type)) +
  stat_summary(fun.data=mean_se, geom="pointrange") +
  stat_summary(aes(y=fitted(m.cp)),
              fun.y=mean, geom="line")
```



```
> ggplot(CP, aes(Stimulus, d.prime, shape=Type)) +
  stat_summary(fun.data=mean_se, geom="pointrange") +
  stat_summary(aes(y=fitted(m.cp), linetype=Type),
              fun.y=mean, geom="line")
```



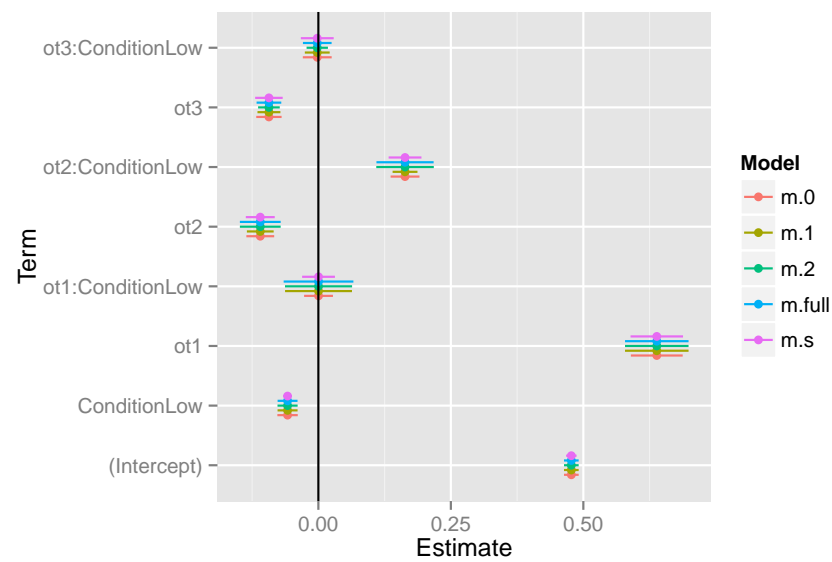
Chapter 4

1. Use the `TargetFix` data set to explore how random effect structure affects fixed effect estimates. Start from a simple model that only has `Subject` random effects and gradually add `Subject:Condition` random effects (intercept, linear, quadratic, cubic). For the addition of each time term to the `Subject:Condition` random effects, how do each of the fixed effect estimates change? How do their standard errors change? How do their t -values and p -values change?

```
> load("Examples.Rdata")
> t <- poly(unique(TargetFix$timeBin), 3)
> TargetFix[, paste("ot", 1:3, sep="")] <-
  t[TargetFix$timeBin, 1:3]
> m.s <- lmer(meanFix ~ (ot1+ot2+ot3)*Condition +
  (ot1+ot2+ot3 | Subject),
  control=lmerControl(optimizer = "bobyqa"),
  data=TargetFix, REML=FALSE)
> m.0 <- lmer(meanFix ~ (ot1+ot2+ot3)*Condition +
  (ot1+ot2+ot3 | Subject) +
  (1 | Subject:Condition),
  control=lmerControl(optimizer = "bobyqa"),
  data=TargetFix, REML=FALSE)
> m.1 <- lmer(meanFix ~ (ot1+ot2+ot3)*Condition +
  (ot1+ot2+ot3 | Subject) +
  (1+ot1 | Subject:Condition),
  control=lmerControl(optimizer = "bobyqa"),
  data=TargetFix, REML=FALSE)
> m.2 <- lmer(meanFix ~ (ot1+ot2+ot3)*Condition +
  (ot1+ot2+ot3 | Subject) +
  (1+ot1+ot2 | Subject:Condition),
  control=lmerControl(optimizer = "bobyqa"),
  data=TargetFix, REML=FALSE)
> m.full <- lmer(meanFix ~ (ot1+ot2+ot3)*Condition +
  (ot1+ot2+ot3 | Subject) +
  (1+ot1+ot2+ot3 | Subject:Condition),
  control=lmerControl(optimizer = "bobyqa"),
  data=TargetFix, REML=FALSE)
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
m.s	19	-667	-596	352	-705			
m.0	20	-712	-638	376	-752	48	1	0.000
m.1	22	-763	-682	404	-807	55	2	0.000
m.2	25	-816	-723	433	-866	59	3	0.000
m.full	29	-813	-706	436	-871	6	4	0.223

Effect of adding Subject:Condition random effects on model goodness of fit.



For Condition fixed effects, adding the corresponding Subject:Condition random effect does not change the estimates but it increases the SE, which has consequences for the t - and p -values. The effect on the SE might be easier to see in a table:

Term	m.0	m.1	m.2	m.full	m.s
(Intercept)	0.014	0.014	0.014	0.014	0.010
ConditionLow	0.019	0.019	0.019	0.019	0.008
ot1	0.049	0.060	0.060	0.060	0.050
ot1:ConditionLow	0.027	0.063	0.063	0.066	0.031
ot2	0.026	0.025	0.038	0.038	0.027
ot2:ConditionLow	0.027	0.023	0.054	0.054	0.031
ot3	0.024	0.022	0.020	0.023	0.026
ot3:ConditionLow	0.027	0.023	0.020	0.027	0.031

Effect of adding Subject:Condition random effects on SE for each fixed effect term.

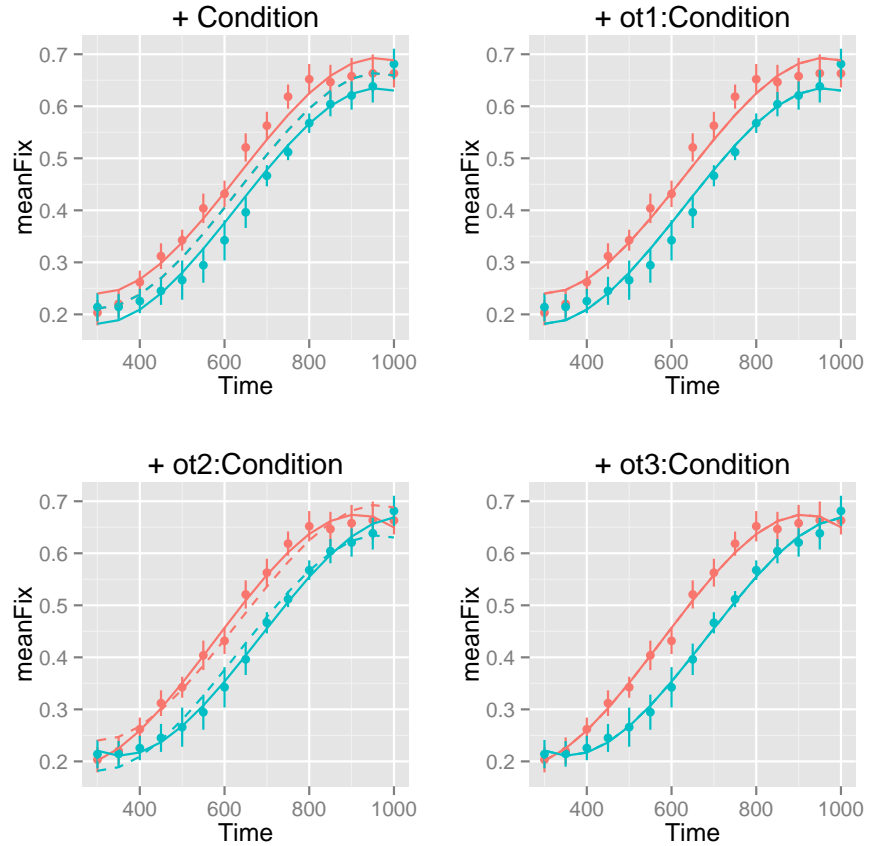
2. Make plots showing the effect of *Condition* on each of the time terms (intercept, linear, quadratic, and cubic) for the *TargetFix* data. Use these plots to explain what aspect of the data is captured by each term.

```

> m.base <- lmer(meanFix ~ (ot1+ot2+ot3) +
  (ot1+ot2+ot3 | Subject),
  control=lmerControl(optimizer = "bobyqa"),
  data=TargetFix, REML=FALSE)
> m.0 <- lmer(meanFix ~ (ot1+ot2+ot3) + Condition +
  (ot1+ot2+ot3 | Subject) +
  (1 | Subject:Condition),
  control=lmerControl(optimizer = "bobyqa"),
  data=TargetFix, REML=F)
> p1 <- ggplot(TargetFix,
  aes(Time, meanFix, color=Condition)) +
  stat_summary(fun.data=mean_se, geom="pointrange") +
  stat_summary(aes(y=fitted(m.0)),
  fun.y=mean, geom="line") +
  stat_summary(aes(y=fitted(m.base)),
  fun.y=mean, geom="line",
  linetype="dashed") +
  ggtitle("+ Condition") + guides(color=FALSE)
> m.1 <- lmer(meanFix ~ (ot1+ot2+ot3) + Condition +
  ot1:Condition +
  (ot1+ot2+ot3 | Subject) +
  (ot1 | Subject:Condition),
  control=lmerControl(optimizer = "bobyqa"),
  data=TargetFix, REML=FALSE)
> p2 <- ggplot(TargetFix,
  aes(Time, meanFix, color=Condition)) +
  stat_summary(fun.data=mean_se, geom="pointrange") +
  stat_summary(aes(y=fitted(m.1)),
  fun.y=mean, geom="line") +
  stat_summary(aes(y=fitted(m.0)),
  fun.y=mean, geom="line",
  linetype="dashed") +
  ggtitle("+ ot1:Condition") + guides(color=FALSE)
> m.2 <- lmer(meanFix ~ (ot1+ot2)*Condition + ot3 +
  (ot1+ot2+ot3 | Subject) +
  (ot1+ot2 | Subject:Condition),
  control=lmerControl(optimizer = "bobyqa"),
  data=TargetFix, REML=FALSE)
> p3 <- ggplot(TargetFix,
  aes(Time, meanFix, color=Condition)) +

```

```
stat_summary(fun.data=mean_se, geom="pointrange") +
stat_summary(aes(y=fitted(m.2)),
              fun.y=mean, geom="line") +
stat_summary(aes(y=fitted(m.1)),
              fun.y=mean, geom="line",
              linetype="dashed") +
  ggtitle("+ ot2:Condition") + guides(color=FALSE)
> m.full <- lmer(meanFix ~ (ot1+ot2+ot3)*Condition +
                 (ot1+ot2+ot3 | Subject) +
                 (ot1+ot2+ot3 | Subject:Condition),
                 control=lmerControl(optimizer = "bobyqa"),
                 data=TargetFix, REML=FALSE)
> p4 <- ggplot(TargetFix,
               aes(Time, meanFix, color=Condition)) +
  stat_summary(fun.data=mean_se, geom="pointrange") +
  stat_summary(aes(y=fitted(m.full)),
               fun.y=mean, geom="line") +
  stat_summary(aes(y=fitted(m.2)),
               fun.y=mean, geom="line",
               linetype="dashed") +
  ggtitle("+ ot3:Condition") + guides(color=FALSE)
> library(gridExtra)
> grid.arrange(p1, p2, p3, p4)
```



Chapter 5

1. The `FunctTheme` data frame contains data on the time course of activation of thematic and function relationships (discussed in Chapter 4). The study was a factorial design with two categorical predictor variables: `Condition` (Function vs. Thematic) and `Object` (related `Competitor` vs. `Unrelated` distractor; omit the `Target` object for this exercise).
 - (a) Analyze the competition data using GCA with fourth-order orthogonal polynomials and the default treatment coding of the factors. Interpret the parameter estimates – explicitly identify which aspect of the data or comparison is captured by each parameter estimate.

```

> #prep dat for GCA
> load("Examples.Rdata")
> FunctTheme <- droplevels(subset(FunctTheme,
                                Object != "Target"))
> FunctTheme$timeBin <- FunctTheme$Time/50 - 9
> t <- poly(1:max(FunctTheme$timeBin), 4)
> FunctTheme[,paste("ot", 1:4, sep="")] <-
  t[FunctTheme$timeBin, 1:4]
> #initial analysis
> m.FT <- lmer(meanFix ~ (ot1+ot2+ot3+ot4)*Object*Condition +
                (ot1+ot2+ot3+ot4 | Subject) +
                (ot1+ot2+ot3+ot4 | Subject:Object:Condition),
                control = lmerControl(optimizer="bobyqa"),
                data=FunctTheme, REML=FALSE)

> coef(summary(m.FT)) #output suppressed to save space

```

- (b) Re-code the factors to make the parameter estimates more intuitive to interpret: set the `Unrelated` object to be the reference level for the `Object` factor and change the contrast coding for the `Condition` factor from *treatment* to *sum*. Fit a new GCA model and explain any changes in the parameter estimates.

```

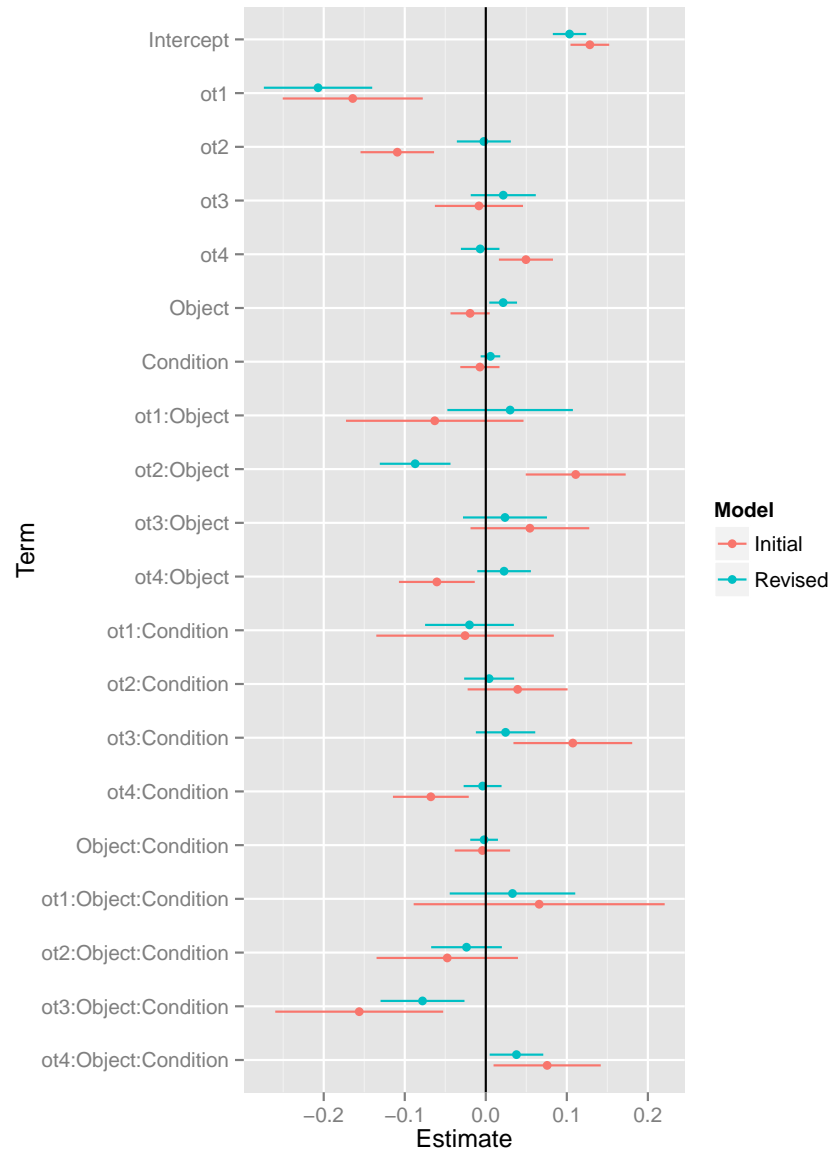
> #recode to make Unrelated the reference object
> FunctTheme$Object <- relevel(FunctTheme$Object, "Unrelated")
> #use sum contrasts for Condition
> FunctTheme$ConditionSum <- C(FunctTheme$Condition, sum)
> contrasts(FunctTheme$ConditionSum)

      [,1]
Function    1
Thematic   -1

> #refit model
> m.FT2 <- lmer(meanFix ~ (ot1+ot2+ot3+ot4)*Object*ConditionSum +
                    (ot1+ot2+ot3+ot4 | Subject) +
                    (ot1+ot2+ot3+ot4 | Subject:Object:ConditionSum),
                    control = lmerControl(optimizer="bobyqa"),
                    data=FunctTheme, REML=FALSE)

> coef(summary(m.FT2)) #output suppressed to save space

```



This figure shows all of the coefficient estimates ± 2 SE for the two models. Two key differences:

- In the revised model, the Object coefficient refers to the main effect of competition (competitor vs. unrelated) on the intercept across both conditions and is statistically significant (SE range does not include 0). In the initial model, the Object co-

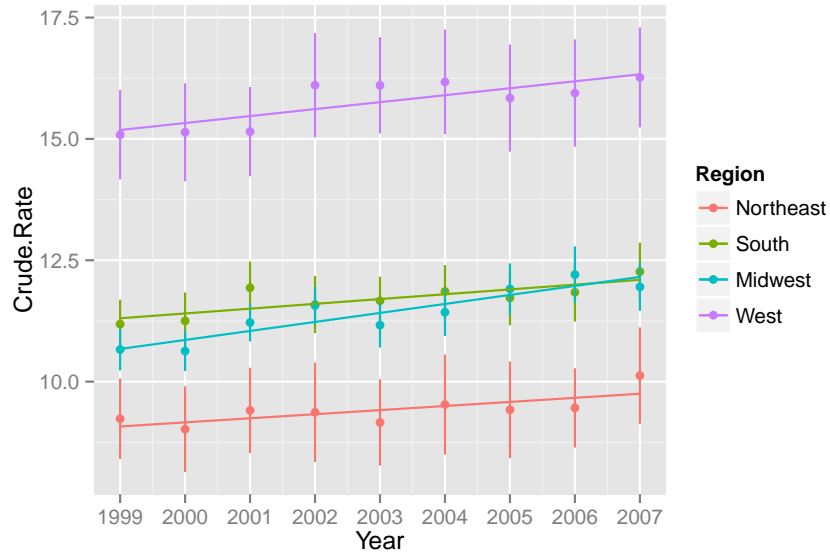
efficient is the simple effect in the Function condition and does not reach statistical significance (SE range includes 0).

- In the revised model, the Condition coefficients refer to the simple effect of condition (function vs. thematic) on the unrelated object and none of them are statistically significant, so there were no systematic differences in the time course of unrelated object fixations between conditions (as expected). In the initial model, the Condition coefficients refer to the simple effect of condition on the competitor object and some of the coefficients are statistically significant (effects on the cubic and quartic term) reflecting differences in competitor fixation between the two conditions, although the Object-by-Condition interaction coefficients better capture that effect.

2. Use the full model of the WISQARS suicide data to test all pairwise comparisons between regions.

```
> library(multcomp)
> wisqars.suicide$Time <- wisqars.suicide$Year - 1999
> m.sui <- lmer(Crude.Rate ~ Time * Region + (Time | State),
               data=wisqars.suicide, REML=FALSE)
> ggplot(wisqars.suicide,
         aes(Year, Crude.Rate, color=Region)) +
  stat_summary(fun.data=mean_se, geom="pointrange") +
  stat_summary(aes(y=fitted(m.sui)),
              fun.y=mean, geom="line") +
  scale_x_continuous(breaks=1999:2007)
> coef(summary(m.sui))
```

	Estimate	Std. Error	t value
(Intercept)	9.075728	0.761247	11.92219
Time	0.084278	0.047247	1.78377
RegionSouth	2.229082	0.941429	2.36776
RegionMidwest	1.596809	1.007035	1.58565
RegionWest	6.107605	0.990296	6.16745
Time:RegionSouth	0.014683	0.058430	0.25129
Time:RegionMidwest	0.101222	0.062502	1.61951
Time:RegionWest	0.059056	0.061463	0.96083



- (a) Which regions differ from which other regions in terms of baseline suicide rate?

```

> contrast.matrix.int = rbind(
  "NE vs. S" = c(0, 0, 1, 0, 0, 0, 0, 0),
  "NE vs. MW" = c(0, 0, 0, 1, 0, 0, 0, 0),
  "NE vs. W" = c(0, 0, 0, 0, 1, 0, 0, 0),
  "S vs. MW" = c(0, 0, -1, 1, 0, 0, 0, 0),
  "S vs. W" = c(0, 0, -1, 0, 1, 0, 0, 0),
  "MW vs. W" = c(0, 0, 0, 1, -1, 0, 0, 0))
> comps.int <- glht(m.sui, contrast.matrix.int)
> summary(comps.int, test = adjusted("none"))

```

	Estimate	SE	z	p
NE vs. S	2.229	0.941	2.37	0.018
NE vs. MW	1.597	1.007	1.59	0.113
NE vs. W	6.108	0.990	6.17	<1e-04
S vs. MW	-0.632	0.861	-0.73	0.463
S vs. W	3.879	0.841	4.61	<1e-04
MW vs. W	-4.511	0.914	-4.93	<1e-04

Baseline suicide rate differences: NE has lower rate than S, W has higher rate than all others.

- (b) Which regions differ from which other regions in terms of rate of change of suicide rate?

```

> contrast.matrix.lin = rbind(
  "NE vs. S" = c(0, 0, 0, 0, 0, 1, 0, 0),
  "NE vs. MW" = c(0, 0, 0, 0, 0, 0, 1, 0),
  "NE vs. W" = c(0, 0, 0, 0, 0, 0, 0, 1),
  "S vs. MW" = c(0, 0, 0, 0, 0, -1, 1, 0),
  "S vs. W" = c(0, 0, 0, 0, 0, -1, 0, 1),
  "MW vs. W" = c(0, 0, 0, 0, 0, 0, 1, -1))
> comps.lin <- glht(m.sui, contrast.matrix.lin)
> summary(comps.lin, test = adjusted("none"))

```

	Estimate	SE	z	p
NE vs. S	0.0147	0.0584	0.25	0.80
NE vs. MW	0.1012	0.0625	1.62	0.11
NE vs. W	0.0591	0.0615	0.96	0.34
S vs. MW	0.0865	0.0534	1.62	0.11
S vs. W	0.0444	0.0522	0.85	0.40
MW vs. W	0.0422	0.0567	0.74	0.46

No significant regional differences in rate of change in suicide rate.

- (c) Re-analyze the data with orthogonal time to estimate differences in overall suicide rate instead of “baseline” differences in 1999. Are any of the results different?

```

> y <- poly(0:max(wisqars.suicide$Time), 1)
> wisqars.suicide$OrthTime <- y[wisqars.suicide$Time + 1, ]
> summary(wisqars.suicide)
> m.sui.orth <- lmer(Crude.Rate ~ OrthTime * Region +
  (OrthTime | State),
  data=wisqars.suicide, REML=FALSE)
> comps.int.orth <- glht(m.sui.orth, contrast.matrix.int)
> summary(comps.int.orth, test = adjusted("none"))

```

	Estimate	SE	z	p
NE vs. S	2.2878	1.0140	2.26	0.024
NE vs. MW	2.0017	1.0847	1.85	0.065
NE vs. W	6.3438	1.0666	5.95	<1e-04
S vs. MW	-0.2861	0.9274	-0.31	0.758
S vs. W	4.0560	0.9063	4.48	<1e-04
MW vs. W	-4.3421	0.9847	-4.41	<1e-04

Relatively small differences. Mainly NE vs. MW is now marginal, due to the slightly higher increase in suicide rate in MW region compared to NE over this period.

```
> comps.lin.orth <- glht(m.sui.orth, contrast.matrix.lin)
> summary(comps.lin.orth, test = adjusted("none"))
```

	Estimate	SE	z	p
NE vs. S	0.1137	0.4526	0.25	0.80
NE vs. MW	0.7841	0.4841	1.62	0.11
NE vs. W	0.4574	0.4761	0.96	0.34
S vs. MW	0.6703	0.4140	1.62	0.11
S vs. W	0.3437	0.4045	0.85	0.40
MW vs. W	0.3266	0.4395	0.74	0.46

Estimates are different due to change in scale, but statistical test results are identical.

Chapter 6

The word learning accuracy data in `WordLearnEx` are proportions from a binary response variable (correct/incorrect). Re-analyze these data using logistic and quasi-logistic GCA and compare the results to linear GCA from Chapter 3.

1. Convert the accuracy proportions to number of correct and incorrect responses (there were 6 trials per block). Compute the empirical logits and corresponding weights.

```
> load("Examples.Rdata")
> #Convert accuracy to number of correct and total trials
> WordLearnEx$Correct <- round(WordLearnEx$Accuracy * 6)
> WordLearnEx$N <- 6
> #Compute the empirical logits and corresponding weights
> WordLearnEx$logit <- with(WordLearnEx,
  log((Correct+0.5) / (N-Correct+0.5)))
> WordLearnEx$wts <- with(WordLearnEx,
  1/(Correct+0.5) + 1/(N-Correct+0.5))
```

2. Analyze the data using logistic GCA. Are the results different from linear GCA? If so, how and why?

```
> #prep orthogonal polynomials
> t <- poly(unique(WordLearnEx$Block), 2)
> WordLearnEx[,paste("ot", 1:2, sep="")] <-
  t[WordLearnEx$Block, 1:2]
> #fit logistic model
> m.logit <- glmer(cbind(Correct, N-Correct) ~ (ot1+ot2)*TP +
  (ot1+ot2 | Subject),
  data=WordLearnEx, family=binomial)
> coef(summary(m.logit))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.582560	0.19623	8.06480	7.3358e-16
ot1	2.212270	0.30327	7.29469	2.9936e-13
ot2	-0.082114	0.21485	-0.38219	7.0232e-01
TPHigh	0.565771	0.28384	1.99327	4.6232e-02
ot1:TPHigh	0.347333	0.44560	0.77947	4.3571e-01
ot2:TPHigh	-0.989489	0.32163	-3.07649	2.0945e-03

In the logistic model, there is a significant effect of TP on the intercept and the quadratic term; in the linear model only the effect on the quadratic

term was significant, the effect on the intercept was marginal. The logistic model is more sensitive to the accuracy differences later in learning when overall accuracy is high relative to initial differences when accuracy is near chance, but the linear model treats those equally.

3. Analyze the data using quasi-logistic GCM with empirical logits. Are the results different from linear GCM? If so, how and why? Are they different from logistic GCM? If so, how and why?

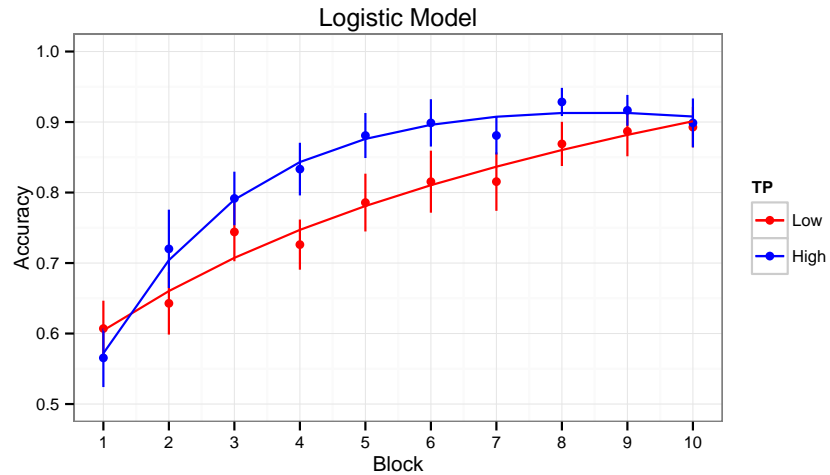
```
> m.elogit <- lmer(elog ~ (ot1+ot2)*TP + (ot1+ot2 | Subject),
                  data=WordLearnEx, weights=1/wts, REML=FALSE)
> coefs.elog <- data.frame(coef(summary(m.elogit)))
> coefs.elog$p <- 2 * (1 - pnorm(abs(coefs.elog$t.value)))
> coefs.elog
```

	Estimate	Std..Error	t.value	p
(Intercept)	1.022911	0.10795	9.47545	0.0000000
ot1	1.322930	0.15628	8.46531	0.0000000
ot2	-0.122646	0.14541	-0.84346	0.3989724
TPHigh	0.313514	0.15383	2.03808	0.0415422
ot1:TPHigh	0.070434	0.22486	0.31324	0.7541002
ot2:TPHigh	-0.596572	0.21072	-2.83105	0.0046395

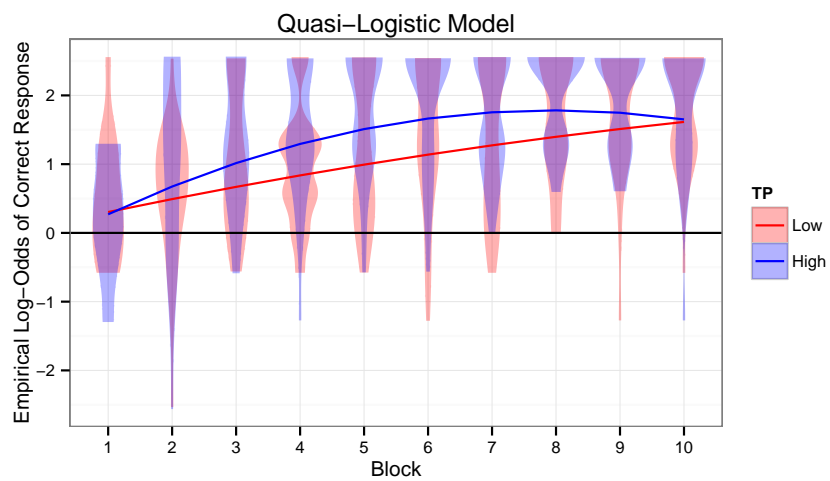
Results are very similar to the logistic model; they differ from the linear model in the same way as the logistic model did and for the same reasons.

4. Plot the model fits for each of the analyses.

```
> ggplot(WordLearnEx, aes(Block, Accuracy, color=TP)) +
  stat_summary(fun.data = mean_se, geom="pointrange") +
  stat_summary(aes(y=fitted(m.logit)),
              fun.y = mean, geom="line") +
  theme_bw(base_size=10) + expand_limits(y=c(0.5, 1.0)) +
  scale_color_manual(values=c("red", "blue")) +
  scale_x_continuous(breaks=1:10) +
  labs(y="Accuracy", title="Logistic Model")
```



```
> ggplot(WordLearnEx, aes(Block, elog, color=TP, fill=TP)) +
  geom_violin(aes(x=factor(Block)), position="identity",
    alpha=0.3, color=NA) +
  stat_summary(aes(y=fitted(m.elogit)),
    fun.y = mean, geom="line") +
  theme_bw(base_size=10) + geom_hline(yintercept=0) +
  scale_color_manual(values=c("red", "blue")) +
  scale_fill_manual(values=c("red", "blue")) +
  labs(y="Empirical Log-Odds of Correct Response",
    title="Quasi-Logistic Model")
```



Chapter 7

The `CohortRhyme` example data set contains data from an eye-tracking experiment (Mirman et al., 2011) that investigated phonological competition between *cohorts* (e.g., *penny – pencil*) and *rhymes* (e.g., *carrot – parrot*). Three groups of participants were tested: five individuals with Broca’s aphasia, three individuals with Wernicke’s aphasia, and 12 control participants.

1. Use fourth-order orthogonal polynomials to analyze (separately) the cohort and rhyme competition effects.

```
> load("Examples.Rdata")
> t <- poly(unique(CohortRhyme$timeBin), 4)
> CohortRhyme[,paste("ot", 1:4, sep="")] <-
  t[CohortRhyme$timeBin, 1:4]
```

- (a) Test group differences in cohort and rhyme competition effects.

```
> cohort.base <- lmer(FixProp ~ (ot1+ot2+ot3+ot4)*Object +
  (ot1+ot2+ot3+ot4)*Group +
  (1+ot1+ot2+ot3+ot4 | subjID) +
  (1+ot1+ot2 | subjID:Object),
  control = lmerControl(optimizer="bobyqa"),
  data=subset(CohortRhyme, Type == "Cohort"),
  REML=FALSE)
> cohort.group <- lmer(FixProp ~ (ot1+ot2+ot3+ot4)*Object*Group +
  (1+ot1+ot2+ot3+ot4 | subjID) +
  (1+ot1+ot2 | subjID:Object),
  control = lmerControl(optimizer="bobyqa"),
  data=subset(CohortRhyme, Type == "Cohort"),
  REML=FALSE)

> anova(cohort.base, cohort.group)
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
cohort.base	42	-2288.78	-2092.02	1186.39	-2372.78			
cohort.group	52	-2303.57	-2059.97	1203.78	-2407.57	34.79	10	0.0001

*Adding effects of group on competition (i.e., Group*Object interaction terms) significantly improved model fit ($\chi^2(10) = 34.79, p < 0.001$), indicating that there were overall group differences in cohort competition.*


```

> rhyme.base <- lmer(FixProp ~ (ot1+ot2+ot3+ot4)*Object +
  (ot1+ot2+ot3+ot4)*Group +
  (1+ot1+ot2+ot3+ot4 | subjID) +
  (1+ot1+ot2 | subjID:Object),
  control = lmerControl(optimizer="bobyqa"),
  data=subset(CohortRhyme, Type == "Rhyme"),
  REML=FALSE)
> rhyme.group <- lmer(FixProp ~ (ot1+ot2+ot3+ot4)*Object*Group +
  (1+ot1+ot2+ot3+ot4 | subjID) +
  (1+ot1+ot2 | subjID:Object),
  control = lmerControl(optimizer="bobyqa"),
  data=subset(CohortRhyme, Type == "Rhyme"),
  REML=FALSE)

> anova(rhyme.base, rhyme.group)

```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
rhyme.base	42	-2270.79	-2074.04	1177.40	-2354.79			
rhyme.group	52	-2265.18	-2021.58	1184.59	-2369.18	14.39	10	0.1559

*Adding effects of group on competition (i.e., Group*Object interaction terms) did not significantly improve model fit ($\chi^2(10) = 14.39, p = 0.1559$), indicating that overall group differences in rhyme competition did not reach significance.*

- (b) Evaluate all pairwise group comparisons for different time terms (see Chapter 5).

```

> x <- coef(summary(cohort.group))
> # group effects on competition are parameters 21-30
> library(multcomp)
> contrast.matrix = rbind(
  "Intr: Ctrl - Br" = c(rep(0, 20),
    1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
  "Intr: Ctrl - We" = c(rep(0, 20),
    0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),
  "Intr: We - Br" = c(rep(0, 20),
    1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0),
  "Lin: Ctrl - Br" = c(rep(0, 20),
    0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0),
  "Lin: Ctrl - We" = c(rep(0, 20),
    0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0),
  "Lin: We - Br" = c(rep(0, 20),
    0, 0, 1, -1, 0, 0, 0, 0, 0, 0, 0),
  "Quad: Ctrl - Br" = c(rep(0, 20),
    0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0),

```

```

"Quad: Ctrl - We" = c(rep(0, 20),
                      0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0),
"Quad: We - Br" = c(rep(0, 20),
                    0, 0, 0, 0, 0, 0, 1, -1, 0, 0, 0)
)

> summary(glht(cohort.group, contrast.matrix),
          test = adjusted("none"))

```

	Estimate	Std. Error	z value	Pr(> z)
Intr: Ctrl - Br	0.007	0.017	0.43	0.670
Intr: Ctrl - We	-0.050	0.020	-2.43	0.015
Intr: We - Br	0.057	0.023	2.46	0.014
Lin: Ctrl - Br	-0.047	0.060	-0.79	0.428
Lin: Ctrl - We	0.009	0.073	0.13	0.898
Lin: We - Br	-0.057	0.082	-0.69	0.490
Quad: Ctrl - Br	0.044	0.059	0.75	0.454
Quad: Ctrl - We	0.279	0.071	3.91	0.000
Quad: We - Br	0.322	0.081	4.00	0.000

Simultaneous Tests for General Linear Hypotheses: Cohort Competition

The cohort competition effect for the Wernicke's aphasia group differed from the other two groups on the intercept and quadratic terms.

```

> summary(glht(rhyme.group, contrast.matrix),
          test = adjusted("none"))

```

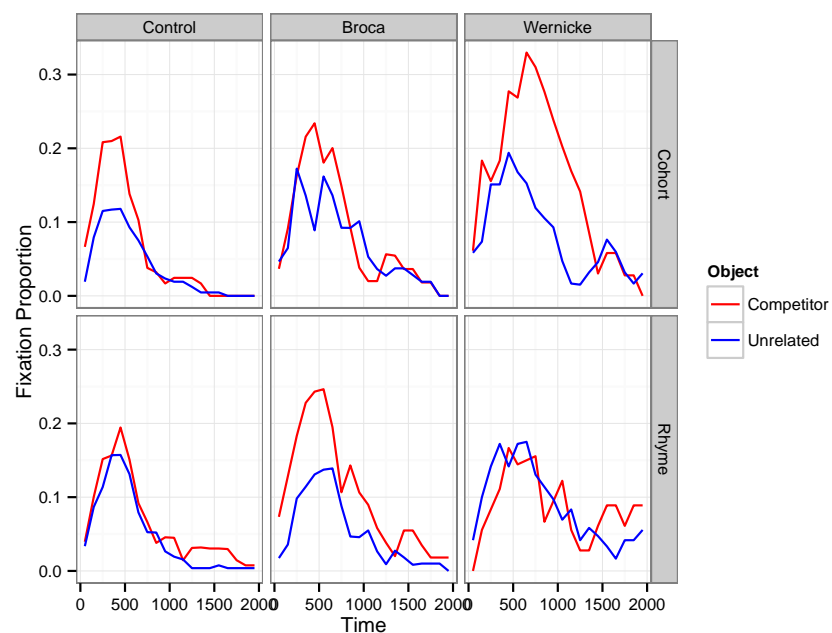
	Estimate	Std. Error	z value	Pr(> z)
Intr: Ctrl - Br	-0.036	0.016	-2.26	0.024
Intr: Ctrl - We	0.018	0.019	0.93	0.353
Intr: We - Br	-0.053	0.022	-2.47	0.014
Lin: Ctrl - Br	0.122	0.065	1.89	0.059
Lin: Ctrl - We	-0.128	0.078	-1.64	0.101
Lin: We - Br	0.250	0.089	2.82	0.005
Quad: Ctrl - Br	-0.007	0.056	-0.12	0.902
Quad: Ctrl - We	-0.011	0.068	-0.17	0.869
Quad: We - Br	0.048	0.078	0.62	0.533

Simultaneous Tests for General Linear Hypotheses: Rhyme Competition

The rhyme competition effect for the Broca's aphasia group differed from the other two groups on the intercept and linear terms (the Control-Broca's difference on the linear term was marginal).

- (c) Make a multi-panel plot that shows cohort and rhyme competition effects for each of the groups (you may want to use `facet_grid`).

```
> ggplot(CohortRhyme, aes(Time, FixProp, color=Object)) +
  facet_grid(Type ~ Group) +
  stat_summary(fun.y=mean, geom="line") +
  theme_bw(base_size=10) + ylab("Fixation Proportion") +
  scale_color_manual(values=c("red", "blue"))
```



2. Compute individual participant's cohort and rhyme competition effects on the intercept, linear, and quadratic time terms (tip: remember to remove the group fixed effect so that these effect sizes will be relative to the overall mean, not the diagnosis group mean).

```
> #fit cohort competition model without group effects
> cohort.nogroup <- lmer(FixProp ~ (ot1+ot2+ot3+ot4)*Object +
  (1+ot1+ot2+ot3+ot4 | subjID) +
  (1+ot1+ot2 | subjID:Object),
  control = lmerControl(optimizer="bobyqa"),
  data=subset(CohortRhyme, Type == "Cohort"),
  REML=FALSE)
> #compute individual cohort effect sizes
> re.cohort <- data.frame(
```

```

      colsplit(row.names(ranef(cohort.nogroup)$'subjID:Object'),
              ":", c("Subject", "Object")),
      ranef(cohort.nogroup)$'subjID:Object')
> ES.coh <- ddply(re.cohort, .(Subject), summarize,
  Cohort.Intercept = X.Intercept.[Object=="Competitor"] -
    X.Intercept.[Object=="Unrelated"],
  Cohort.Linear = ot1[Object=="Competitor"] -
    ot1[Object=="Unrelated"],
  Cohort.Quadratic = ot2[Object=="Competitor"] -
    ot2[Object=="Unrelated"])
> #fit rhyme competition model without group effects
> rhyme.nogroup <- lmer(FixProp ~ (ot1+ot2+ot3+ot4)*Object +
  (1+ot1+ot2+ot3+ot4 | subjID) +
  (1+ot1+ot2 | subjID:Object),
  control = lmerControl(optimizer="bobyqa"),
  data=subset(CohortRhyme, Type == "Rhyme"),
  REML=FALSE)
> #compute individual rhyme effect sizes
> re.rhyme <- data.frame(
  colsplit(row.names(ranef(rhyme.nogroup)$'subjID:Object'),
          ":", c("Subject", "Object")),
  ranef(rhyme.nogroup)$'subjID:Object')
> ES.rhy <- ddply(re.rhyme, .(Subject), summarize,
  Rhyme.Intercept = X.Intercept.[Object=="Competitor"] -
    X.Intercept.[Object=="Unrelated"],
  Rhyme.Linear = ot1[Object=="Competitor"] -
    ot1[Object=="Unrelated"],
  Rhyme.Quadratic = ot2[Object=="Competitor"] -
    ot2[Object=="Unrelated"])
> #combine cohort and rhyme individual effect sizes
> ES <- merge(ES.coh, ES.rhy, by="Subject")
> #add group assignments from original data frame
> group <- unique(subset(CohortRhyme, select=c(subjID, Group)))
> ES <- merge(ES, group, by.x="Subject", by.y="subjID")

```

3. Test correlations between cohort and rhyme effect sizes for the full set of participants and separately for the control and aphasic participants.

```

> #effect size correlations: Intercept
> cor.test(ES$Cohort.Intercept, ES$Rhyme.Intercept)
> cor.test(ES$Cohort.Intercept[ES$Group != "Control"],
  ES$Rhyme.Intercept[ES$Group != "Control"])
> cor.test(ES$Cohort.Intercept[ES$Group == "Control"],
  ES$Rhyme.Intercept[ES$Group == "Control"])
> #effect size correlations: Linear

```

```

> cor.test(ES$Cohort.Linear, ES$Rhyme.Linear)
> cor.test(ES$Cohort.Linear[ES$Group != "Control"],
           ES$Rhyme.Linear[ES$Group != "Control"])
> cor.test(ES$Cohort.Linear[ES$Group == "Control"],
           ES$Rhyme.Linear[ES$Group == "Control"])
> #effect size correlations: Quadratic
> cor.test(ES$Cohort.Quadratic, ES$Rhyme.Quadratic)
> cor.test(ES$Cohort.Quadratic[ES$Group != "Control"],
           ES$Rhyme.Quadratic[ES$Group != "Control"])
> cor.test(ES$Cohort.Quadratic[ES$Group == "Control"],
           ES$Rhyme.Quadratic[ES$Group == "Control"])

```

	Intercept	Linear	Quadratic
Overall	-0.383 .	-0.022	-0.022
Aphasic	-0.861 **	-0.438	0.056
Control	0.338	0.473	-0.17

Correlations between cohort and rhyme effect sizes.

4. Make a multi-panel scatterplot that shows the correlations for the intercept, linear, and quadratic terms in separate panels (tip: you may need to use `dcast`, which is the companion function to `melt` that lets you convert from a long data format to a wide data format).

```

> ES.m <- melt(ES, id=c("Subject", "Group"))
> ES.m <- cbind(ES.m, colsplit(ES.m$variable, "\\.",
                             c("Type", "Term")))
> ES.c <- dcast(ES.m, Subject + Group + Term ~ Type)
> ggplot(ES.c, aes(Cohort, Rhyme, color=Group)) +
  facet_wrap(~ Term, scales="free", ncol=1) +
  geom_point() + theme_bw(base_size=10) +
  scale_color_manual(values=c("grey", "red", "blue"))

```

