**Chapter 9 Additional Linear Modeling Topics**

(2022.2.12) 박규서

| **9 Additional Linear Modeling Topics**  **9.1 Handling Highly Correlated Variables**  9.1.1 An Initial Linear Model of Online Spend  9.1.2 Remediating Collinearity  **9.2 Linear Models for Binary Outcomes: Logistic Regression**  9.2.1 Basics of the Logistic Regression Model  9.2.2 Data for Logistic Regression of Season Passes  9.2.3 Sales Table Data  9.2.4 Language Brief: Classes and Attributes of Objects  9.2.5 Finalizing the Data  9.2.6 Fitting a Logistic Regression Model  9.2.7 Reconsidering the Model  9.2.8 Additional Discussion  **9.3 Hierarchical Models**  9.3.1 Some HLM Concepts  9.3.2 Ratings-Based Conjoint Analysis for the Amusement Park  9.3.3 Simulating Ratings-Based Conjoint Data  9.3.4 An Initial Linear Model  9.3.5 Initial Hierarchical Linear Model with lme4  9.3.6 Complete Hierarchical Linear Model  9.3.7 Conclusion for Classical HLM  **9.4 Bayesian Hierarchical Linear Models**  9.4.1 Initial Linear Model with MCMCregress()  9.4.2 Hierarchical Linear Model with MCMChregress()  9.4.3 Inspecting Distribution of Preference  **9.5 A Quick Comparison of the Effects**  **9.6 Key Points**  9.7 Data Sources  9.8 Learning More |
| --- |

| **9장 코드 위치**:  <http://r-marketing.r-forge.r-project.org/code/chapter9-ChapmanFeit2e.R> |
| --- |

**[개요]**

□ 4가지 추가 주제에 대하여 논의

As we noted in Chap. 7, the range of applications and methods in linear modeling and regression is vast. In this chapter, we discuss four additional topics in linear modeling that often arise in marketing:

**① 공선성**

Handling highly correlated observations, which pose a problem known as collinearity, as mentioned in Sect. 7.2.1. In Sect. 9.1 we examine the problem in detail, along with ways to detect and remediate collinearity in a data set.

**② 로지스틱 회귀모형 (예/아니오 또는 이진결과 모델)**

Fitting models for yes/no, or binary outcomes, such as purchasing a product. In Sect. 9.2 we introduce logistic regression models to model binary outcomes and their influences.

**③ hierarchical linear models (HLM)- 위계적 선형모형**

표본 전체 뿐만 아니라 개별 선호도와 반응에 대한 모델 도출 - 마케팅에서 종종 개별 소비자와 전체 집단 사이의 행동 및 제품 관심의 다양성을 이해하기를 원함 - ratings-based conjoint analysis data에서 소비자 선호에 대한 hierarchical linear models (HLM) 고려

Finding a model for the preferences and responses of individuals, not only for the sample as a whole. In marketing, we often wish to understand individual consumers and the diversity of behavior and product interest among people. In Sect. 9.3 we consider hierarchical linear models (HLM) for consumer preference in ratings-based conjoint analysis data.

**④ 베이지안 방법을 사용한 위계적 선형모형** - HLM by introducing hierarchical Bayesian (HB) methods, HB for ratings-based conjoint analysis

In marketing, hierarchical models of individual preference are most often estimated using Bayesian methods. In Sect. 9.4 we continue the discussion of HLM by introducing hierarchical Bayesian (HB) methods, and we apply HB for ratings-based conjoint analysis.

Except for the two HLM sections, these topics are not especially closely related to one another; unlike other chapters in this book, they may be read independently within this chapter. Still, each section builds on models presented earlier in the book and will extend your knowledge of issues and applications for linear modeling. More importantly, each is a foundational part of a compete toolbox for marketing analysis.

**9.1 Handling Highly Correlated Variables**

We have mentioned several times (as in Sect. 7.2.1) that highly correlated explanatory variables cause problems with linear models. In this section, we examine why that is the case and strategies to address the problem.

We consider a question that might arise with the retail sales data in Chap. 4, which simulated summaries of 12 month online and in-store transactions by customer (see Sect. 4.1). The question is this: **which variables are most predictive of online spending? If we wished to increase online spending by customers, which factors might we consider?**

***9.1.1 An Initial Linear Model of Online Spend***

Either create the simulated retail sales data (Sect. 4.1) or load it from the book’s website:

| | > ######  > ###### Collinearity  > ######  >  > cust.df <- read.csv("http://goo.gl/PmPkaG")  > summary(cust.df)  cust.id age credit.score email distance.to.store  Min. : 1.0 Min. :19.34 Min. :543.0 Length:1000 Min. : 0.2136  1st Qu.: 250.8 1st Qu.:31.43 1st Qu.:691.7 Class :character 1st Qu.: 3.3383  Median : 500.5 Median :35.10 Median :725.5 Mode :character Median : 7.1317  Mean : 500.5 Mean :34.92 Mean :725.5 Mean : 14.6553  3rd Qu.: 750.2 3rd Qu.:38.20 3rd Qu.:757.2 3rd Qu.: 16.6589  Max. :1000.0 Max. :51.86 Max. :880.8 Max. :267.0864    online.visits online.trans online.spend store.trans store.spend  Min. : 0.00 Min. : 0.000 Min. : 0.00 Min. : 0.000 Min. : 0.00  1st Qu.: 0.00 1st Qu.: 0.000 1st Qu.: 0.00 1st Qu.: 0.000 1st Qu.: 0.00  Median : 6.00 Median : 2.000 Median : 37.03 Median : 1.000 Median : 30.05  Mean : 28.29 Mean : 8.385 Mean : 170.32 Mean : 1.323 Mean : 47.58  3rd Qu.: 31.00 3rd Qu.: 9.000 3rd Qu.: 177.89 3rd Qu.: 2.000 3rd Qu.: 66.49  Max. :606.00 Max. :169.000 Max. :3593.03 Max. :12.000 Max. :705.66    sat.service sat.selection  Min. :1.00 Min. :1.000  1st Qu.:3.00 1st Qu.:2.000  Median :3.00 Median :2.000  Mean :3.07 Mean :2.401  3rd Qu.:4.00 3rd Qu.:3.000  Max. :5.00 Max. :5.000  NA's :341 NA's :341 | | --- | |  | | | >   | > head(cust.df)  cust.id age credit.score email distance.to.store online.visits online.trans  1 1 22.89437 630.6089 yes 2.582494 20 3  2 2 28.04994 748.5746 yes 48.175989 121 39  3 3 35.87942 732.5459 yes 1.285712 39 14  4 4 30.52740 829.5889 yes 5.253992 1 0  5 5 38.73575 733.7968 no 25.044693 35 11  6 6 42.41277 685.8721 yes 18.462041 1 1  online.spend store.trans store.spend sat.service sat.selection  1 58.42999 4 140.32321 3 3  2 756.88008 0 0.00000 3 3  3 250.32801 0 0.00000 NA NA  4 0.00000 2 95.91194 4 2  5 204.69331 0 0.00000 1 1  6 19.01065 0 0.00000 NA NA | | --- | |  | | | >  > head(cust.df[,-1])  age credit.score email distance.to.store online.visits online.trans online.spend  1 22.89437 630.6089 yes 2.582494 20 3 58.42999  2 28.04994 748.5746 yes 48.175989 121 39 756.88008  3 35.87942 732.5459 yes 1.285712 39 14 250.32801  4 30.52740 829.5889 yes 5.253992 1 0 0.00000  5 38.73575 733.7968 no 25.044693 35 11 204.69331  6 42.41277 685.8721 yes 18.462041 1 1 19.01065  store.trans store.spend sat.service sat.selection  1 4 140.32321 3 3  2 0 0.00000 3 3  3 0 0.00000 NA NA  4 2 95.91194 4 2  5 0 0.00000 1 1  6 0 0.00000 NA NA | | --- | | | | --- | --- | --- | --- | --- | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |

(회귀분석, online.spend가 있는 것 대상)

Now we use lm() to model spend as a function of all other variables (online.spend ∼.).We omit customers with zero online spend; having exactly zero spend is probably related to different factors than positive spend, and we are interested here in the associations for those who spend anything. We also index [ , -1] to omit the customer ID column:

| | > #### lm to predict online spend  > spend.m1 <- lm(online.spend ~ .,  + data=subset(cust.df[ , -1], online.spend > 0))  > summary(spend.m1)  Call:  lm(formula = online.spend ~ ., data = subset(cust.df[, -1], online.spend >  0))  Residuals:  Min 1Q Median 3Q Max  -234.097 -8.828 0.519 9.956 227.238  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 6.718948 33.537665 0.200 0.8413  age 0.422773 0.450825 0.938 0.3489  credit.score -0.033698 0.043977 -0.766 0.4440  emailyes -5.689283 5.806621 -0.980 0.3278  distance.to.store -0.043548 0.100539 -0.433 0.6651  online.visits -0.072269 0.204061 -0.354 0.7234  online.trans 20.610744 0.667450 30.880 <2e-16 \*\*\*  store.trans 0.135018 3.211943 0.042 0.9665  store.spend 0.001796 0.078732 0.023 0.9818  sat.service 5.638769 3.016181 1.870 0.0623 .  sat.selection -4.370606 2.909073 -1.502 0.1338  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 42.77 on 407 degrees of freedom  (214 observations deleted due to missingness)  Multiple R-squared: 0.9831, Adjusted R-squared: 0.9827  F-statistic: 2363 on 10 and 407 DF, p-value: < 2.2e-16 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

- 교재는 생략되었으나 여기에서는 모든 것이 다 나타남

- onlien spend는 online transcations와 밀접한 관련 있음(계수 =20.6). 그러나, online visits와는 관련 없음

- 상기 모델은 거의 모든 분산을 설명(R2 = 0.98)

- online transactions은 visit에 의존해야 하기에 이 두변수가 비슷한 패턴을 보여야 하지 않는지?

- store.trans의 표준오차가 상당히 커서 추정치가 매우 불확실하다는 것을 보여줌

We have omitted much of the summary to show a few key points. First, online spend is closely related the number of online transactions (coefficient = 20.6) but not the number of online visits. That is puzzling. Second, the model accounts for almost all the available variance, R2 = 0.98. These results should cause concern.

Because online transactions are dependent on visits, shouldn’t those two variables show a similar pattern? How could we be so lucky as to fit a model that nearly perfectly predicts online spending (insofar as it is assessed by R2)? And notice that the standard error on store.trans is quite large, showing that its estimate is very uncertain.

\*\*시각화\*\*

- scatterplotMatrix()로 시각화

- 그림 9.1의 결과는 극도로 치우친 변수와 상관관계가 매우 높은 변수쌍을 보여줌

- Box-Cox변환을 이용하여 데이터를 변환

- scale()을 사용하여 데이터를 표준화함

If we turn to data visualization using scatterplotMatrix() (Sect. 7.2.1), we see some problems:

| 🡺 scatterplotMatrix가 에러가 발생 🡺 대신 아래와 같이 설치하여 실행   | > install.packages("psych")  Installing package into ‘C:/Users/박규서/Documents/R/win-library/4.0’  (as ‘lib’ is unspecified)  trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/psych\_2.1.9.zip'  Content type 'application/zip' length 4243294 bytes (4.0 MB)  downloaded 4.0 MB  package ‘psych’ successfully unpacked and MD5 sums checked  The downloaded binary packages are in  C:\Users\Public\Documents\ESTsoft\CreatorTemp\RtmpAf7kI6\downloaded\_packages  >  > library(psych)  > pairs.panels(cust.df, stars = TRUE, lm = TRUE) | | --- | |  | | |  | | --- | | |
| --- | --- | --- | --- | --- |

The result in Fig. 9.1 shows variables with extreme skew and pairs of variables that are very highly correlated.

Our first step to remediate the situation is to **transform the data using a Box-Cox transformation**. Building on the transformation routines we saw in Sect. 4.5.5, we write a short function that uses BoxCox.lambda() from the forecast package to select the transformation lambda automatically [103]. At the same time, we standardize the data with **scale()** (Sect. 7.3.3):

| **\* 박스-콕스 변환(Box-Cox Transformation)**  - 선형회귀모형의 정규성 가정이 성립한다고 보기 어려울 경우 **종속변수를 정규분포에 가깝게 변환시키는 기법**  λ ≠ 0, ωi(λ) = (yi^λ-1)/λ  λ = 0, ωi(λ) = log(yi)  변수에 상수 λ를 투입하면 정규분포에 근사할 수 있다는 전제하에서 분석이 수행되는데 일반적으로 우도함수의 최우추정법으로 도출하며 변수값은 양수여야 함  **[Box-cox 변환 python]**  pip install ipympl  %matplotlib ipympl  import matplotlib.pyplot as plt  import numpy as np  import seaborn as sns  from scipy import stats  from scipy.stats import boxcox  from scipy.special import inv\_boxcox  from google.colab import output  output.enable\_custom\_widget\_manager()  plt.figure(figsize = (8, 8))  data = np.random.beta(1, 3, 5000)  sns.distplot(data)  plt.show()    tdata = boxcox(data)[0]  plt.figure(figsize = (8, 8))  sns.distplot(tdata)  plt.show() |
| --- |

| > install.packages('forecast')   | > ### Automatic data transformation  >  > ### autotransform function  > ### NOTE: install package "forecast" first, if needed  > autoTransform <- function(x) {  + library(forecast)  + return(scale(BoxCox(x, BoxCox.lambda(x))))  + } | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

- 데이터 프레임에서 전체 사례를 선택하고 예측 변수가 아니므로 고객 ID 열([, -1])을 삭제함

- 그런 다음 online spend가 양수인 행만 가져옴

- 이메일(숫자가 아님)을 제외한 모든 열을 인덱싱하는 벡터를 만든 다음 각 숫자 열에 lapply() autoTransform() 함수를 만듦

We select the complete cases from our data frame, dropping the customer ID column ([, -1]) because it is not a predictor. Then we take only the rows with positive online spend. We create a vector to index all the columns except email (which is not numeric), and then lapply() the autoTransform() function to each numeric column:

| > cust.df.bc <- cust.df[complete.cases(cust.df), -1]  > cust.df.bc <- subset(cust.df.bc, online.spend > 0)  > numcols <- which(colnames(cust.df.bc) != "email")  > cust.df.bc[ , numcols] <- lapply(cust.df.bc[ , numcols], autoTransform ) |
| --- |

The result is a **data frame with standardized, more normally distributed values**, which we can check with summary() and scatterplotMatrix():

[정규화한 데이터 확인]

| | > summary(cust.df.bc) # output not shown in book  age.V1 credit.score.V1 email distance.to.store.V1  Min. :-2.6028151 Min. :-3.549645 Length:418 Min. :-4.151249  1st Qu.:-0.6102340 1st Qu.:-0.664190 Class :character 1st Qu.:-0.642231  Median : 0.0319899 Median : 0.024254 Mode :character Median : 0.110540  Mean : 0.0000000 Mean : 0.000000 Mean : 0.000000  3rd Qu.: 0.6444580 3rd Qu.: 0.687857 3rd Qu.: 0.723209  Max. : 2.4751221 Max. : 2.691592 Max. : 2.235170  online.visits.V1 online.trans.V1 online.spend.V1 store.trans.V1  Min. :-2.5478012 Min. :-1.6105773 Min. :-1.9847582 Min. :-1.3303670  1st Qu.:-0.6422584 1st Qu.:-0.9065743 1st Qu.:-0.7686352 1st Qu.:-1.3303670  Median : 0.1265838 Median : 0.1092692 Median : 0.1358798 Median : 0.5382470  Mean : 0.0000000 Mean : 0.0000000 Mean : 0.0000000 Mean : 0.0000000  3rd Qu.: 0.8231873 3rd Qu.: 0.8739755 3rd Qu.: 0.8897866 3rd Qu.: 0.7660742  Max. : 1.7768679 Max. : 1.8204545 Max. : 1.7784232 Max. : 1.3605408  store.spend.V1 sat.service.V1 sat.selection.V1  Min. :-1.3373428 Min. :-2.2411659 Min. :-1.6476883  1st Qu.:-1.3373428 1st Qu.:-0.0917797 1st Qu.:-0.3270153  Median : 0.5997467 Median :-0.0917797 Median :-0.3270153  Mean : 0.0000000 Mean : 0.0000000 Mean : 0.0000000  3rd Qu.: 0.7674898 3rd Qu.: 1.1080508 3rd Qu.: 0.7455990  Max. : 1.2717696 Max. : 2.3654420 Max. : 2.5321752  > # scatterplotMatrix(cust.df.bc) # output not shown in book  > # 상기 scatterplotMatrix가 오류가 나서 아래 것으로 임시 대체  >  > pairs.panels(cust.df.bc, stars = TRUE, lm = TRUE) | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

We refit the model using the transformed data:

| | > #### lm to predict online spend, after transform  > spend.m2 <- lm(online.spend ~ ., data=cust.df.bc)  > summary(spend.m2)  Call:  lm(formula = online.spend ~ ., data = cust.df.bc)  Residuals:  Min 1Q Median 3Q Max  -0.38976 -0.05409 0.00027 0.05591 0.26628  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) -0.0059639 0.0108943 -0.547 0.584  age 0.0001875 0.0044821 0.042 0.967  credit.score -0.0026632 0.0045185 -0.589 0.556  emailyes 0.0071023 0.0119316 0.595 0.552  distance.to.store -0.0020362 0.0048800 -0.417 0.677  online.visits -0.0003913 0.0126165 -0.031 0.975  online.trans 0.9960378 0.0126687 78.622 <2e-16 \*\*\*  store.trans -0.0266674 0.0480675 -0.555 0.579  store.spend 0.0274099 0.0475888 0.576 0.565  sat.service 0.0059429 0.0052732 1.127 0.260  sat.selection 0.0030628 0.0052624 0.582 0.561  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.08747 on 407 degrees of freedom  Multiple R-squared: 0.9925, Adjusted R-squared: 0.9923  F-statistic: 5410 on 10 and 407 DF, p-value: < 2.2e-16 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

- 데이터가 표준화되었기 때문에 계수가 더 작아졌음

- 데이터를 변환하고 표준화하는 것은 좋은 생각이긴 하지만 online spend가 거래와 관련이 높지만 방문과는 관련이 없다는 추정치를 바꾸지 못했음

- 실제로 전체 모델은 단순히 트랜잭션 수만으로 지출을 예측하는 모델보다 나을 것이 없음(모델 비교를 위해 anova() 사용에 대하여 섹션 6.5.1 참조).

The coefficients are smaller now because the data have been standardized. Transforming and standardizing the data, although a good idea, have not changed the unbelievable estimate that online spend is highly related to transactions yet unrelated to visits. Indeed, the full model is no better than one that simply predicts spending from the number of transactions alone (see Sect. 6.5.1 on using anova() to compare models):

| (모델 비교를 위해 anova() 사용)   | > # which is almost identical to a bivariate solution  > spend.m3 <- lm(online.spend ~ online.trans, data=cust.df.bc)  > anova(spend.m3, spend.m2)  Analysis of Variance Table  Model 1: online.spend ~ online.trans  Model 2: online.spend ~ age + credit.score + email + distance.to.store +  online.visits + online.trans + store.trans + store.spend +  sat.service + sat.selection  Res.Df RSS Df Sum of Sq F Pr(>F)  1 416 3.1539  2 407 3.1139 9 0.040001 0.5809 0.8129 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

모형 적합치 간의 작은 차이는 높은 p-값(p = 0.8129)에 반영됨. 모형 간에 차이가 없다는 귀무가설을 기각할 수 없음

여기서 문제는 공선성임. visits과 transactions가 매우 밀접하게 관련되어 있고, 선형 모델이 효과가 가산적이라고 가정하기 때문에 한 변수(예: 거래)에 기인한 효과는 다른 변수에 공동으로 기인하는 모델에서 사용할 수 없음. 그것은 높은 상관 관계(방문)임.

이로 인해 예측 변수의 표준 오차가 증가하며, 이는 계수 추정치가 매우 불확실하거나 불안정하다는 것을 의미함.

기본 관계가 동일한 경우에도 데이터의 사소한 변동으로 인해 계수 추정치가 샘플마다 크게 다를 수 있음

The small difference between the model fits is reflected in the high p-value (p =0.8129), and thus the null hypothesis of no difference between the models cannot be rejected.

The problem here is collinearity: because visits and transactions are so highly related, and also because a linear model assumes that effects are additive, an effect attributed to one variable (such as transactions) is not available in the model to be attributed jointly to another that is highly correlated (visits). This will cause the standard errors of the predictors to increase, which means that the coefficient estimates will be highly uncertain or unstable. As a practical consequence, this may cause coefficient estimates to differ dramatically from sample to sample due to minor variations in the data even when underlying relationships are the same.

***9.1.2 Remediating Collinearity***

- 데이터의 공선성 정도는 분산 팽창 계수(VIF)로 평가할 수 있음.

- 이는 변수가 상관되지 않거나 단순 단일 예측변수 회귀를 수행한 경우와 비교하여 다른 변수와의 공유 분산으로 인해 선형 모델에서 계수의 표준 오차(분산)가 얼마나 증가하는지 추정함.

- car 패키지의 vif()를 사용하여 spend.m2 모델에서 VIF를 평가함

The degree of collinearity in data can be assessed as the variance inflation factor (VIF). This estimates how much the standard error (variance) of a coefficient in a linear model is increased because of shared variance with other variables, compared to the situation if the variables were uncorrelated or simple single predictor regression were performed.

We assess VIF in the spend.m2 model using vif() from the car package:

| > # the problem  > library(car)  > vif(spend.m2)  age credit.score email distance.to.store online.visits  1.094949 1.112784 1.046874 1.297978 8.675817  online.trans store.trans store.spend sat.service sat.selection  8.747756 125.931383 123.435407 1.515576 1.509377 |
| --- |

- 일반적으로 VIF > 5.0은 공선성을 완화할 필요가 있음을 나타냄.

- spend.m2에서 VIF는 공선성이 online... 및 store... 변수에 대해 공선성을 해결해야 함을 나타냄.

- 공선성을 완화하기 위한 **세 가지 일반적인 전략**이 있음

**① 상관관계가 높은 변수는 생략함**

**② 상관관계가 높은 예측변수 세트에 대한 주성분 또는 요인을 추출하여 상관관계를 제거함(8장 참조)**

**③ 공선성에 강력한 방법, 즉 기존 선형 모델링과 다른 방법을 사용함**. 이 가능성을 철저하게 고려하기에는 너무 많은 옵션이 있지만 고려해야 할 한 가지 방법은 한 번에 변수들의 하위 집합만을 사용하는 random forest approach임(섹션 11.4.2 참조).

- 현재 데이터에 대한 또 다른 옵션은 collinear 변수(예: spend per transaction)를 결합하는 새로운 관심 측정값을 구성하는 것임. 여기에서 목적을 위해 위의 처음 두 가지 옵션을 탐색하고 spend.m4 및 spend.m5 모델을 생성

- 수식에서 -를 사용하여 online.trans와 store.trans를 제거하여 모델 spend.m4에 대해 상관관계가 높은 변수를 생략함

A common rule of thumb is that VIF > 5.0 indicates the need to mitigate collinearity. In spend.m2, the VIF indicates that collinearity should be addressed for the online... and store... variables.

There are three general strategies for mitigating collinearity: • Omit variables that are highly correlated. • Eliminate correlation by extracting principal components or factors for sets of highly-correlated predictors (see Chap. 8). • Use a method that is robust to collinearity, i.e., something other than traditional linear modeling. There are too many options to consider this possibility exhaustively, but one method to consider would be a random forest approach, which only uses a subset of variables at a time (see Sect. 11.4.2).

Another option for the present data would be to construct a new measure of interest that combines the collinear variables (such as spend per transaction). For purposes here, we explore the first two options above and create models spend.m4 and spend.m5.

We omit highly correlated variables for model spend.m4 by excluding online. trans and store. trans, using - in the formula:

| | > # solution 1: omit covariates  > #  > spend.m4 <- lm(online.spend ~ . -online.trans -store.trans,  + data=cust.df.bc)  >  > vif(spend.m4)  age credit.score email distance.to.store online.visits  1.081411 1.103586 1.033945 1.211607 1.026148  store.spend sat.service sat.selection  1.215208 1.507866 1.509001  > summary(spend.m4)  Call:  lm(formula = online.spend ~ . - online.trans - store.trans, data = cust.df.bc)  Residuals:  Min 1Q Median 3Q Max  -1.36373 -0.13135 0.05888 0.18476 1.03794  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) -0.0923395 0.0435047 -2.123 0.0344 \*  age -0.0333779 0.0178813 -1.867 0.0627 .  credit.score -0.0084524 0.0180637 -0.468 0.6401  emailyes 0.1099655 0.0476011 2.310 0.0214 \*  distance.to.store 0.0001702 0.0189271 0.009 0.9928  online.visits 0.9295374 0.0174184 53.365 <2e-16 \*\*\*  store.spend 0.0092463 0.0189552 0.488 0.6260  sat.service -0.0121405 0.0211147 -0.575 0.5656  sat.selection 0.0048591 0.0211226 0.230 0.8182  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.3511 on 409 degrees of freedom  Multiple R-squared: 0.8791, Adjusted R-squared: 0.8767  F-statistic: 371.6 on 8 and 409 DF, p-value: < 2.2e-16 | | --- | |  | |  | |
| --- | --- | --- | --- |

- **(방법1)** 이제 VIF가 수용될 수 있으면 online trasnations를 제외하면 online visits이 online spend을 가장 잘 예측할 수 있음. email status와 age도 online spend와 약간 관련이 있음을 알 수 있음.

The VIF is now acceptable and we see that online visits are now the best predictor of online spend once we’ve left out online transactions. We can see that email status

and age are also slightly related to online spend.

**(방법2) – PCA 주성분분석 사용 (8장 내용)**

- 또 다른 접근 방식은 상관 데이터의 주요 구성 요소를 사용하는 것임. 8장에서와 같이 주성분은 상관관계가 없음(직교). 따라서, PCA는 동일한 PCA에 포함된 다른 변수와 공선성이 없는 합성 변수를 추출하는 방법을 제공함.

- PCA를 사용하여 online 변수에 대한 첫 번째 구성 요소를 추출한 다음 store 변수에 대해 다시 이 작업을 수행하고 두 개의 초기 구성 요소를 데이터 프레임에 추가함.

Another approach is to use the principal components of the correlated data. As you will recall from Chap. 8, principal components are uncorrelated (orthogonal). Thus, PCA provides a way to extract composite variables that are guaranteed to be free of collinearity with other variables that are included in the same PCA.

We use PCA to extract the first component for the online variables, and then do this again for the store variables, and add those two initial components to the data frame:

| | > # solution 2: principal components  > #  > pc.online <- prcomp(cust.df.bc[ , c("online.visits", "online.trans")])  > cust.df.bc$online <- pc.online$x[ , 1]  >  > pc.store <- prcomp(cust.df.bc[ , c("store.trans", "store.spend")])  > cust.df.bc$store <- pc.store$x[ , 1] | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

Then we fit a new model:

| > spend.m5 <- lm(online.spend ~ email + age + credit.score +  + distance.to.store + sat.service + sat.selection +  + online + store,  + data=cust.df.bc)  >  > summary(spend.m5)  Call:  lm(formula = online.spend ~ email + age + credit.score + distance.to.store +  sat.service + sat.selection + online + store, data = cust.df.bc)  Residuals:  Min 1Q Median 3Q Max  -0.83697 -0.08532 0.01288 0.09664 0.73327  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) -3.928e-02 2.410e-02 -1.630 0.1039  emailyes 4.678e-02 2.638e-02 1.773 0.0769 .  age -1.695e-02 9.882e-03 -1.715 0.0871 .  credit.score -3.649e-03 9.981e-03 -0.366 0.7148  distance.to.store -2.666e-05 1.051e-02 -0.003 0.9980  sat.service -2.762e-03 1.167e-02 -0.237 0.8130  sat.selection 3.153e-03 1.167e-02 0.270 0.7872  online -7.019e-01 6.933e-03 -101.247 <2e-16 \*\*\*  store -2.712e-03 7.455e-03 -0.364 0.7162  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.1941 on 409 degrees of freedom  Multiple R-squared: 0.9631, Adjusted R-squared: 0.9623  F-statistic: 1333 on 8 and 409 DF, p-value: < 2.2e-16  > vif(spend.m5)  email age credit.score distance.to.store sat.service  1.039458 1.081430 1.103206 1.224019 1.508487  sat.selection online store  1.509001 1.032362 1.228073  > |
| --- |

- VIF는 이 모델에서 문제를 일으키지 않음

- online spend은 여전히 주로 online activity(PCA 모델의 첫 번째 구성 요소인 online에서 캡처됨)과 관련이 있으며 email status 및 age와 약간 관련되어 있음을 알 수 있음.

- 주성분을 설명변수로 사용하는 결과를 해석할 때 주의할 점은 성분이 임의의 수치적 방향을 갖는다는 것임. 여기에서 online에 대한 음수 계수는 online activity가 판매 감소를 초래한다는 것을 의미하지 않음.

- online sales가 주로 online activity와 관련이 있다는 이 결과는 처음에는 흥미롭지 않은 것처럼 보일 수 있지만 잘못된 결과보다 분명한 결과를 얻는 것이 좋음.

- 이 결과는 online spend와 관련된 요소를 더 완벽하게 이해하기 위해 웹사이트나 온라인 쇼핑에 대한 태도와 같은 다른 데이터를 수집하도록 유도할 수 있음.

VIF poses no problem in this model, and we see that online spend is still associated primarily with online activity (as captured in the first component of the PCA model, online) and perhaps slightly with email status and age. One caution when interpreting results that use principal components as explanatory variables is that the components have arbitrary numerical direction; the negative coefficient for online here does not imply that online activity results in lower sales.

Although this result—that online sales relate primarily to online activity—may at first appear to be uninteresting, it is better to have an obvious result than an incorrect result. This result might prompt us to collect other data, such as attitudes about our website or online shopping, to build a more complete understanding of factors associated with online spending.

| **Key Points**  **Collinearity**  • Collinearity occurs when two or more variables are highly associated. Including them in a linear model can result in confusing, nonsensical, or misleading results, because the model cannot differentiate the contribution from each of them (Sect. 9.1).  • The variance inflation factor (VIF) provides a measure of shared variance among variables in a model. A rule of thumb is that collinearity should be addressed for a variable when VIF > 5 (Sect. 9.1.2).  • Common approaches to fixing collinearity include omitting highly-correlated variables, and using principal components or factor scores (see Chap. 8) instead of individual items (Sect. 9.1.2). |
| --- |

**9.2 Linear Models for Binary Outcomes: Logistic Regression 이진 결과에 대한 선형 모델: 로지스틱 회귀**

| **[로짓모형, 로지스틱회귀]**  \* 로짓모형 또는 로지스틱 회귀모형이란 종속/반응변수가 범주형 혹은 이상형 값을 가지는 경우에 적용될 수 있는 회귀모형임. 이 때 독립변수를 종종 공변량(covariate)이라고 함 (**로지스틱 회귀(Logistic Regression)**는 회귀를 사용하여 데이터가 어떤 범주에 속할 확률을 0에서 1 사이의 값으로 예측하고 그 확률에 따라 가능성이 더 높은 범주에 속하는 것으로 분류해주는 방법.)  - 로짓 모형의 종류는 종속변수가 가지는 범주값에 따라 구분함. 0,1인 이분형을 가지는 경우를 ‘이항 로짓 모형’, 세가지 이상의 범부값을 가지는 다분형인 경우를 ‘다항 로짓 모형’, 그리고 순서형 범부값을 가지는 경우를 ‘순서형 로짓 모형’으로 구분함  -----------------------------------  - 스팸 메일 분류기 같은 예시를 생각하면 쉽다. 어떤 메일을 받았을 때 그것이 스팸일 확률이 0.5 이상이면 spam으로 분류하고, 확률이 0.5보다 작은 경우 ham으로 분류하는 거다. 이렇게 데이터가 2개의 범주 중 하나에 속하도록 결정하는 것을 **2진 분류(binary classification)**라고 한다.  예를 들어 어떤 학생이 공부하는 시간에 따라 시험에 합격할 확률이 달라진다고 해보자. **선형 회귀를 사용하면** 아래와 같은 그림으로 나타낼 수 있다.    공부한 시간이 적으면 시험에 통과 못하고, 공부한 시간이 많으면 시험에 통과한다는 식으로 설명할 수 있다. 그런데 이 회귀선을 자세히 살펴보면 **확률이 음과 양의 방향으로 무한대까지 뻗어 간다.**  말 그대로 ‘선’이라서. 그래서 공부를 2시간도 안 하면 시험에 통과할 확률이 0이 안 된다. 이건 말이 안 된다.  만약 **로지스틱 회귀**를 사용하면 아래와 같이 나타난다.    시험에 합격할 **확률이 0과 1사이의 값**으로 그려진다.  -----------------------------------------------  **[이항 로짓모형]**  - 일반적인 회귀모형에서는 회귀계수(β) 추정치의 핵석이 의미가 있는 반면, **로짓 모형에서는 종속변수가 특정 범주값을 가지는 확률과 확률들의 비인 승산(odds), 승산비(odds ratio)가 보다 중요**함  - Y가 사망 유무, 보험사고 발생 유무 등 0, 1 (관심있는 사건, 판매, 미판매 발생)으로 코드화할 때, Y의 기대값 E(Y)는 곧 ‘Y=1인 확률’이 됨  - E(Y) ≡ π 로 하고, π에 로지스틱분포의 CDF(누적분포함수)를 연결하면 다음과 같이 선형화시킬 수 있음   | ⇒ ⇒ | | --- |   - 여기서 ⇒ 를 logit이라 함  - 로짓모형은 일반회귀모형과 달리 π인 ‘Y=1인 확률’을 추정하는 것이 궁극적인 목표로서, 승산(odds)을 분석에 중요하게 사용함   * **Odds = π/(1-π)** (즉, )   🡺 odds는 사건이 일어날 확률(π)과 일어나지 않을 확률(1-π)의 비율임  - 한편 독립변수 x가 한단위 증가하면 이때 승산은 → , 이 되어, 승산은 배 증가함  - 즉, 독립변수 한단위 증가시킬 때 승산의 증가배수를 승산비(odds ratio)라고 부름  ---------------------------------------------  ※ β1와 승산비의 관계   * β1 > (<) 0: 독립변수가 증가하면 승산은 증가(감소) * β1 = 0 : 독립변수가 승산에 아무런 영향도 미치지 않음 |
| --- | --- |

- 마케터는 종종 예/아니오 결과를 관찰함. 고객이 제품을 구매했습니까? 그녀는 시승을 했습니까? 그녀는 신용 카드를 등록했습니까, 아니면 구독을 갱신했습니까? 또는 판촉에 응답 등.

**- 이러한 종류의 결과는 모두 예 또는 아니오라는 두 가지 가능한 관찰 상태만 있기 때문에 binary임.**

- 특성의 선형 조합으로 결과(1 = 예, 0 = 아니오)를 예측하고자 처음에는 이러한 모델을 7장에서 본 전형적인 선형 회귀 모델에 맞추고자 함. 그렇게 하는 것이 옳지 않은 것은 아니지만, 그러한 결과를 맞추기 위한 보다 유연하고 유용한 방법은 로지스틱 모델(logit model이라고도 함)을 사용하는 것임.

Marketers often observe yes/no outcomes: did a customer purchase a product? Did she take a test drive? Did she sign up for a credit card, or renew her subscription, or respond to a promotion? All of these kinds of outcomes are binary because they have only two possible observed states: yes or no.

At first it is tempting to fit such a model with a typical linear regression model as we saw in Chap. 7, predicting the outcome (1 = yes, 0 = no) as a linear combination of the features. That is not incorrect to do, but a more flexible and useful way to fit such outcomes is with a logistic model (also called a logit model for reasons we’ll discuss below).

***9.2.1 Basics of the Logistic Regression Model***

- 로지스틱 모델의 핵심 기능은 다음과 같음. 결과 확률을 예측 변수의 지수 함수와 연결함. 이것이 왜 바람직한 속성이고 기본 선형 모델의 개선 사항인지 생각해 보겠음.

- 결과의 확률을 모델링함으로써 로지스틱 모델은 두 가지를 달성함. 첫째, 특정 고객이 제품을 구매할 가능성 또는 판촉에 응답할 세그먼트의 예상 비율과 같은 확률 또는 비율인 우리가 관심 있는 대상을 보다 직접적으로 모델링함. 둘째, 모델을 비율에 대한 적절한 범위인 [0, 1]로 제한함. lm()으로 생성된 기본 선형 모델에는 이러한 제한이 없으며 1.05 또는 -0.04와 같은 무의미한 확률을 추정할 수 있음.

- 모델이 작동하는 방식을 이해하는 데 중요한 역할을 하기 때문에 여기에서 공식을 고려해 보기바람. 로지스틱 함수의 방정식은 다음과 같음

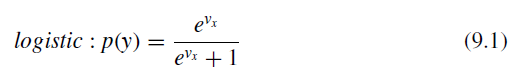
- 이 방정식에서 관심 결과는 y이고 가능성 p(y)를 vx의 함수로 계산함. 일반적으로 vx를 가격과 같은 제품의 기능(x)의 함수로 추정함. vx는 모든 실수 값을 취할 수 있으므로 선형 모델에서 연속 함수로 처리할 수 있음. 이 경우 vx는 하나 이상의 모델의 계수 및 제품의 해당 기능의 중요성을 나타냄.

- 이 공식은 [0, 1] 사이의 값을 제공함. y의 가능성은 vx가 음수일 때 50% 미만, vx = 0일 때 50%, vx가 양수일 때 50% 초과임. 이것을 먼저 직접 계산한 다음 동등한 내장 **plogis() 함수**로 전환함.

The core feature of a logistic model is this: it relates the probability of an outcome to an exponential function of a predictor variable. We’ll illustrate that and show the formula in a moment, but before examining that, let’s consider why those are desirable properties and are improvements on a basic linear model.

By modeling the probability of an outcome, a logistic model accomplishes two things. First, it more directly models what we’re interested in, which is a probability or proportion, such as the likelihood of a given customer to purchase a product, or the expected proportion of a segment who will respond to a promotion. Second, it limits the model to the appropriate range for a proportion, which is [0, 1]. A basic linear model as generated with lm() does not have such a limit and could estimate a nonsensical probability such as 1.05 or −0.04.

We ask indulgence to consider the formula here because it is instrumental in understanding how the model works. The equation for the logistic function is:



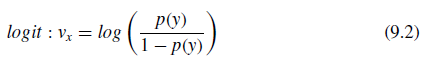
In this equation, the outcome of interest is y, and we compute its likelihood p(y) as a function of vx. We typically estimate vx as a function of the features (x) of a product, such as price. vx can take any real value, so we are able to treat it as a continuous function in a linear model. In that case, vx is composed from one or more coefficients of the model and indicates the importance of the corresponding features of the product.

This formula gives a value between [0, 1]. The likelihood of y is less than 50% when vx is negative, is 50% when vx = 0, and is above 50% when vx is positive. We compute this first by hand, and then switch to the equivalent, built-in plogis() function:

| > ######  > ###### Logistic Regression  > ######  >  > exp(0) / (exp(0) + 1) # computing logistic by hand; could use plogis()  [1] 0.5  > plogis(-Inf) # infinite dispreference = likelihood 0  [1] 0  > plogis(2) # moderate preference = 88% chance (e.g., of purchase)  [1] 0.8807971  > plogis(-0.2) # weak dispreference  [1] 0.450166 |
| --- |

Such a model is known as a logit model, which determines the value of vx from the

logarithm of the relative probability of occurrence of y:



Again, R includes a built-in function **qlogis()** for the logit function:

| > log(0.5 / (1-0.5)) # indifference = 50% likelihood = 0 utility  [1] 0  > log(0.88 / (1-0.88)) # moderate high likelihood  [1] 1.99243  > qlogis(0.88) # equivalent to hand computation  [1] 1.99243  > |
| --- |

In practice, the expressions logit model and logistic regression are used interchangeably.

***9.2.2 Data for Logistic Regression of Season Passes***

- 7장에서 놀이 공원의 예를 고려했음. 이제 공원의 시즌 티켓 판매에 대한 데이터가 있다고 가정함.

- 데이터는 season ticket pass sales (yes 또는 no 값을 가짐) 테이블로 구성됨. 이는 두 가지 요소(제안을 연장하는 데 사용된 채널(이메일, 우편 또는 공원 직접 방문) 및 무료 주차와 같은 다른 기능이 포함된 시즌권을 제공하는 번들로 프로모션되었는지 여부)를 기초로 함

- **마케팅 질문은 다음과 같음: 고객이 번들로 제공되는 시즌 패스(무료 주차 포함)를 구매하거나 구매하지 않습니까?**

- 이 섹션에서는 이러한 데이터를 시뮬레이션하는 방법과 표로 만든 데이터에서 전체 데이터 프레임을 만드는 방법을 살펴봄.

- 데이터 생성을 통해 작업하는 대신 웹사이트에서 데이터를 로드하려면 다음을 사용하여 데이터를 검색할 수 있음.

We considered an amusement park example in Chap. 7. Suppose that we now have data on the the sales of season tickets to the park. The data consist of a table of season ticket pass sales (with values of yes or no), on the basis of two factors: the channel used to extend the offer (email, postal mail, or in-person at the park) and whether it was promoted in a bundle offering the season ticket with another feature such as free parking, or not. The marketing question is this: are customers more likely to purchase the season pass when it is offered in the bundle (with free parking), or not?

In this section, we see how to simulate such data, and how to create a full data frame from tabulated data. If you wish to load the data from the website instead of working through the data creation, you can retrieve it with:

| | > ### season pass data  >  > # alternative code to load the data from website  > pass.df <- read.csv("http://goo.gl/J8MH6A")  > summary(pass.df)  Channel Promo Pass  Length:3156 Length:3156 Length:3156  Class :character Class :character Class :character  Mode :character Mode :character Mode :character  > pass.df$Promo <- factor(pass.df$Promo, levels=c("NoBundle", "Bundle"))  > summary(pass.df)  Channel Promo Pass  Length:3156 NoBundle:1482 Length:3156  Class :character Bundle :1674 Class :character  Mode :character Mode :character  > pass.df$Channel <- factor(pass.df$Channel, levels=c("Email", "Mail", "Park")) # 이것도 해주어야 교재처럼 나옴  > pass.df$Pass <- factor(pass.df$Pass, levels=c("NoPass", "YesPass")) # 이것도 해주어야 교재처럼 나옴  > summary(pass.df)  Channel Promo Pass  Email: 633 NoBundle:1482 NoPass :1567  Mail :1328 Bundle :1674 YesPass:1589  Park :1195 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

위의 두 번째 명령은 섹션9.2.5.에서 설명하는 이유로 필요함. 반드시 CSV를 로드한 후 실행하고 summary()가 위와 일치하는지 확인하기 바람.

Note that the second command above is required for reasons we describe in Sect. 9.2.5. Be sure to run it after loading the CSV and check that the summary() matches the above.

We encourage you to read the rest of this simulation section and the R language lessons it contains. But if you loaded the data and prefer to skip ahead to analysis, you could continue with Sect. 9.2.6.

***9.2.3 Sales Table Data***

Suppose that we have been given sales data as shown in

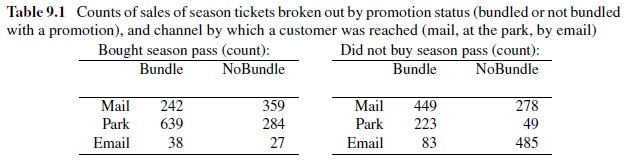


표 9.1에 나와 있는 것처럼 표 데이터를 분석하는 방법은 카이제곱 분석(6.2절)을 포함하여 여러 가지가 있지만 데이터 세트가 너무 크지 않은 경우에는 긴 형식으로 변환하고 개별 관찰의 데이터 프레임을 다시 만드는 다용도 접근 방식이 있음. 이를 통해 번거로움을 최소화하면서 선형 모델링과 같은 모든 접근 방식을 사용할 수 있음.

데이터를 이러한 형식으로 변환하려면 먼저 R에서 croess-tab 데이터 테이블을 다시 생성함. 먼저 Table 9.1의 값을 한 번에 한 열씩 읽어 벡터에 넣음.

There are several ways to analyze tabular data as shown in Table 9.1, including chisquare analysis (Sect. 6.2), but a versatile approach when the data set is not too large is to convert it to long form and recreate a data frame of individual observations. This lets us use a full range of approaches such as linear modeling with minimal hassle.

To convert the data into such format, we first recreate the cross-tab data table in R. We begin this by reading the values from Table 9.1 one column at a time, putting them into a vector:

| > # simulation that generates the data  > # construct the data  >  > pass.tab <- c(242, 639, 38, 359, 284, 27, 449, 223, 83, 278, 49, 485) |
| --- |

Next we add dimensions to the vector, which reformats it as a 3 × 2 × 2 array, and set it to be an object of class "table": 다음으로 벡터에 차원을 추가하여 3 × 2 × 2 배열로 다시 포맷하고 "table" 클래스의 객체로 설정함

| > dim(pass.tab) <- c(3, 2, 2)  > class(pass.tab) <- "table" |
| --- |

We add the marginal labels to the table by setting its dimnames attribute: dimnames 속성을 설정하여 테이블에 marginal label을 추가함

| > dimnames(pass.tab) <- list(Channel=c("Mail", "Park", "Email"),  + Promo=c("Bundle", "NoBundle"),  + Pass=c("YesPass", "NoPass") ) |
| --- |

We describe more about class, table, and dimnames in optional Sect. 9.2.4 below. For now, we inspect the resulting table and confirm that it matches Table 9.1:

- 클래스, 테이블, dimnames에 대한 자세한 내용은 섹션9.2.4에서 설명함.

- 여기에서는 결과 테이블을 검사하고 테이블 9.1과 일치하는지 확인함

| | > pass.tab  , , Pass = YesPass  Promo  Channel Bundle NoBundle  Mail 242 359  Park 639 284  Email 38 27  , , Pass = NoPass  Promo  Channel Bundle NoBundle  Mail 449 278  Park 223 49  Email 83 485 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

We now have the data in R and are ready to create a full data frame from the table. Before that, we take a brief detour into the R language to understand the commands we just used. - 이제 R에 데이터가 있고 테이블에서 전체 데이터 프레임을 만들 준비가 되었음. 그 전에 상기 사용한 명령을 이해하기 위해 R 언어를 잠시 살펴 봄.

***9.2.4 Language Brief: Classes and Attributes of Objects***

- 이 선택적 섹션에서는 R 언어가 데이터 유형을 이해하는 방법을 살펴봄. 로지스틱 회귀 모델을 계속하려면 섹션9.2.5.으로 건너뛸 수 있음.

- R의 모든 객체에는 객체를 처리하는 방법을 결정하는 데 사용되는 관련 class가 있음. 예를 들어 실수 벡터에는 numeric 클래스가 있는 반면 데이터 프레임은 data.frame임.

- 객체의 클래스는 class()에 의해 직접 검사될 수 있음:

In this optional section, we explore how the R language understands data types. If you just want to continue with the logistic regression model, you could skip ahead to Sect. 9.2.5.

Every object in R has an associated class, which functions use to determine how to handle the object. For example, a vector of real numbers has a class of numeric, while a data frame is a data.frame. The class of an object may be inspected directly by class():

| | > ### detour into classes and attributes  > ###  > class(c(1, pi, exp(1)))  [1] "numeric"  > class(data.frame(1:10))  [1] "data.frame" | | --- | |
| --- | --- |

- str()에서 가장 먼저 나열되는 것은 객체의 클래스와 원래 값임.

When we examine str(), the first thing listed is the class of the object and its raw values:

| | > str(pass.tab)  'table' num [1:3, 1:2, 1:2] 242 639 38 359 284 27 449 223 83 278 ...  - attr(\*, "dimnames")=List of 3  ..$ Channel: chr [1:3] "Mail" "Park" "Email"  ..$ Promo : chr [1:2] "Bundle" "NoBundle"  ..$ Pass : chr [1:2] "YesPass" "NoPass" | | --- | |  | |
| --- | --- | --- |

This code shows that pass.tab is an object of class table that comprises values 242 639 .... The is.\*() set of functions tests whether an object is of some class (abbreviated here with \*). For example:

| > is.table(pass.tab)  [1] TRUE  > is.character(pass.tab)  [1] FALSE |
| --- |

Class membership is non-exclusive. For example, tables are composed of counts, and counts are numeric:

| > is.numeric(pass.tab)  [1] TRUE |
| --- |

- as.\*() 함수는 객체를 다른 클래스로 취급(변환 또는 강제 변환)함

The as.\*() functions attempt to treat (convert, or coerce) objects as other classes:

| > as.numeric(pass.tab)  [1] 242 639 38 359 284 27 449 223 83 278 49 485  > as.character(pass.tab)  [1] "242" "639" "38" "359" "284" "27" "449" "223" "83" "278"  [11] "49" "485" |
| --- |

- 이것은 공원 테이블에서 카운트 벡터를 추출하는 방법과 인쇄, 차트 레이블 지정 및 유사한 목적을 위해 문자열로 다시 포맷하는 방법을 보여줌.

- 클래스 외에도 객체는 다른 속성을 가질 수 있음. 속성은 데이터가 아닌 객체의 속성이며 일반적으로 R에 객체에 대해 중요한 것을 알려줌

-책 전체에서 사용한 공통 속성은 열 이름에 대한 names, 행렬 또는 데이터 프레임의 차원에 대한 dim, 객체 유형을 지정하는 class임.

- 이들 각각은 객체에 대해 쿼리할 수 있음.

This shows how we could extract the vector of counts from our park table, and how we might reformat them as character strings for printing, chart labeling, and similar purposes.

In addition to class, objects can have other attributes. An attribute is a property of an object other than its data, and typically tells R something important about the object. Common attributes that we have used throughout the book are names for the names of columns, dim for the dimensions of a matrix or data frame, and class to specify the type of object. Each of these can be queried for an object:

| > names(pass.tab)  NULL  > dim(pass.tab)  [1] 3 2 2  > class(pass.tab)  [1] "table" |
| --- |

이 경우 pass.tab의 이름은 데이터 프레임이나 이름이 유용한 다른 개체가 아니기 때문에 NULL임. 그러나 dim 및 class 속성이 있음을 알 수 있음. 테이블에는 행과 열에 대한 이름도 있으며 이를 dimname이라고 함

In this case, the names for pass.tab are NULL because it is not a data frame or other object for which names are useful. However, we see that it has dim and class attributes. A table also has names for its rows and columns, which are known as dimnames:

| > dimnames(pass.tab)  $Channel  [1] "Mail" "Park" "Email"  $Promo  [1] "Bundle" "NoBundle"  $Pass  [1] "YesPass" "NoPass" |
| --- |

Thus, Channel, the first dimension of the table, has elements "Mail", "Park", and "Email".

You can see all the attributes of an object with attributes():

| > attributes(pass.tab)  $dim  [1] 3 2 2  $class  [1] "table"  $dimnames  $dimnames$Channel  [1] "Mail" "Park" "Email"  $dimnames$Promo  [1] "Bundle" "NoBundle"  $dimnames$Pass  [1] "YesPass" "NoPass" |
| --- |

Attributes may be changed using the assignment operator (<-). We often use this feature to set names of data frames, using names(DATA) <- c("name1","name2", ...). In the code above, we converted pass.tab from a simple vector to a table by assigning class(pass.tab) <- "table" and setting its dim attribute.

As you might imagine, this must be done very carefully! Setting an inappropriate class or dimension of an object will render it useless (but you can usually just change it back to make things work again).

We’ll see another use for classes in Sect. 12.3.3, where we use objects’ classes to determine how to handle multiple data types inside a function. To learn more about the R class and attribute system, review the R language reference [158] and Wickham’s Advanced R [197].

***9.2.5 Finalizing the Data***

그대로 분석에 적합한 pass.tab 테이블에 데이터가 있음. 그러나 대부분의 데이터 세트는 응답자당 하나의 관찰이 있는 확장 데이터 프레임의 형태로 제공되기 때문에 분석이 일반적인 데이터 구조와 일치하도록 테이블에서 완전한 데이터 프레임으로 확장함.

- vcdExtra 패키지[66]의 expand.dft()를 사용하여 테이블을 데이터 프레임으로 확장함

We have the data in a table pass.tab, which is suitable for analysis as is. However, because most data sets come in the form of an extended data frame with one observation per respondent, we expand it from a table to a complete data frame so the analysis will match typical data structures.

We use expand.dft() from the vcdExtra package [66] to expand the table to a data frame:

| | > ### create data frame from table  >  > library(vcdExtra) # install if needed  필요한 패키지를 로딩중입니다: vcd  필요한 패키지를 로딩중입니다: grid  필요한 패키지를 로딩중입니다: gnm  다음의 패키지를 부착합니다: ‘vcdExtra’  The following object is masked from ‘package:carData’:  Burt  Warning messages:  1: 패키지 ‘vcdExtra’는 R 버전 4.0.5에서 작성되었습니다  2: 패키지 ‘vcd’는 R 버전 4.0.5에서 작성되었습니다  3: 패키지 ‘gnm’는 R 버전 4.0.5에서 작성되었습니다  > pass1.df <- expand.dft(pass.tab) # 교재에서는 pass.df로 되어 있으나 상기에서 pass.df와 pass.tab.tab을 이용한 pass1.df를 구분함  > str(pass.df)  'data.frame': 3156 obs. of 3 variables:  $ Channel: Factor w/ 3 levels "Email","Mail",..: 2 2 2 2 2 2 2 2 2 2 ...  $ Promo : Factor w/ 2 levels "NoBundle","Bundle": 2 2 2 2 2 2 2 2 2 2 ...  $ Pass : Factor w/ 2 levels "NoPass","YesPass": 2 2 2 2 2 2 2 2 2 2 ...  > str(pass1.df)  'data.frame': 3156 obs. of 3 variables:  $ Channel: Factor w/ 3 levels "Email","Mail",..: 2 2 2 2 2 2 2 2 2 2 ...  $ Promo : Factor w/ 2 levels "Bundle","NoBundle": 1 1 1 1 1 1 1 1 1 1 ...  $ Pass : Factor w/ 2 levels "NoPass","YesPass": 2 2 2 2 2 2 2 2 2 2 ... | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

이제 고객이 프로모션(프로모션) 유무에 관계없이 채널별로 패스를 구매하는지 여부에 대한 3156개의 관찰이 있는 데이터 프레임을 가짐.

이 데이터에 대해 table()을 사용하여 표 9.1에 있는 것과 다른 cross-tab을 만들 수 있음. 예를 들어, 프로모션 번들(Promo)별 패스(Pass) 구매를 다음과 같이 볼 수 있음

We now have a data frame with 3156 observations for whether a customer purchases a Pass, by Channel, with and without promotion (Promo).

We can use table() on this data to create cross-tabs other than those in Table 9.1. For example, to see purchases of a pass (Pass) by promotion bundle (Promo):

| > table(pass.df$Pass, pass.df$Promo)    NoBundle Bundle  NoPass 812 755  YesPass 670 919  > table(pass.df$Channel, pass.df$Pass) # 추가 예제    NoPass YesPass  Email 568 65  Mail 727 601  Park 272 923  > table(pass.df$Pass, pass.df$Promo, pass.df$Channel) # 추가 예제  , , = Email    NoBundle Bundle  NoPass 485 83  YesPass 27 38  , , = Mail    NoBundle Bundle  NoPass 278 449  YesPass 359 242  , , = Park    NoBundle Bundle  NoPass 49 223  YesPass 284 639 |
| --- |

통계 모델링은 세부 지향적 프로세스이며 데이터에서 모델을 작성하기 전에 주의해야 할 사소한 세부 사항이 하나 있음.

- pass.df의 요소는 알파벳순으로 정렬되어 있음. 이것이 R이 기본적으로 요소 이름을 처리하는 방식임. 그러나 이는 직관적이지 않음. 우리는 NoBundle이 Bundle("bundle = 1"일 수 있음)보다 더 낮은 암시적 값(예: "bundle = 0")을 가져야 한다고 생각할 수 있음. 그러나 방금 본 표에서 NoBundle은 알파벳 순서로 인해 더 높은 값을 가지므로 두 번째 열에 나타남.

- 회귀 모델에서 이는 Bundle의 긍정적인 효과가 음의 값을 갖는다는 것을 의미함. 그런 복잡한 논리를 기억해야 하는 것보다("우리는 no bundle 경우 부정적인 효과를 봄. 이는 부호를 뒤집은 후 bundle에 대한 긍정적인 효과를 의미함...")

- 원하는 순서대로 factor \*\*\*levels\*\*\*로 해당 변수를 재할당하여 순서를 바로 설정하는 것이 더 쉬움.

Statistical modeling is a detail-oriented process, and before building amodel from the data, there is one minor detail to attend to: the factors in pass.df are alphabetized— which is how R handles factor names by default—but that is counterintuitive. We might think that NoBundle should have a lower implicit value (such as “bundle = 0”) than Bundle (which might be “bundle = 1”). However, in the table we just saw, NoBundle appears in the second column because it has a higher value thanks to alphabetic ordering.

In a regression model, that would mean that a positive effect of Bundle would have a negative value (think about it). Rather than having to remember such convoluted logic (“we see a negative effect for no bundle, which really means a positive effect for bundle after we reverse the signs …”), it is easier just to set the order straight by reassigning that variable with the factor levels in the order we want:

| | > pass.df$Promo <- factor(pass.df$Promo, levels=c("NoBundle", "Bundle"))  > table(pass.df$Pass, pass.df$Promo)    NoBundle Bundle  NoPass 812 755  YesPass 670 919 | | --- | | > summary(pass1.df)  Channel Promo Pass  Email: 633 Bundle :1674 NoPass :1567  Mail :1328 NoBundle:1482 YesPass:1589  Park :1195  > summary(pass.df)  Channel Promo Pass  Email: 633 NoBundle:1482 NoPass :1567  Mail :1328 Bundle :1674 YesPass:1589  Park :1195 | |  | |
| --- | --- | --- | --- |

With the data ordered sensibly (Bundle > NoBundle, YesPass > NoPass), we proceed with modeling.

***9.2.6 Fitting a Logistic Regression Model***

- R의 로지스틱 회귀 모델은 7장에서 본 lm()으로 선형 회귀와 유사한 프로세스를 사용하여 일반화 선형 모델(GLM)로 적합시킴. 그러나, GLM이 정규 분포를 따르지 않는 종속 변수를 처리할 수 있다는 차이점이 있음.

- 따라서 일반화된 선형 모델을 사용하여 data counts(예: 구매 횟수) 또는 time interval(예: 웹사이트에서 보낸 시간) 또는 binary variables(예: 구매/구매하지 않음)를 모델링할 수 있음.

- 모든 GLM 모델의 공통된 특징은 link라고 하는 기능을 사용하여 정규 분포 예측 변수를 비정규 결과와 연결한다는 것임.

- 이는 단일의 일관된 프레임워크를 사용하여 다양한 분포에 대한 모델을 fit할 수 있음을 의미함.

---

- 이 책에서 다루는 케이스의 경우 이진 결과를 모델링하고 적절한 분포는 이항 분포임(6.3절 참조).

- R에서 GLM을 추정할 수 있는 여러 함수와 패키지가 있지만 가장 일반적인 것은 glm(...) 함수임.

- glm()은 결과 변수의 분포를 지정하는 family= 인수를 취함. 이진 결과의 경우 family= binomial을 설정함.

- 이항 모델의 기본 링크 함수는 섹션9.2.1에서 본 로짓 함수이므로 지정할 필요가 없음. (그러나 예를 들어 probit link function를 대신 사용하려면 family= binomial(link= "probit")을 지정할 수 있고 다른 연결 함수에 대해서도 유사하게 지정할 수 있음.)

---

- 우리의 마케팅 질문은 "프로모션 번들이 시즌권 판매에 영향을 줍니까?"임.

- glm(..., family=binomial) 및 다르지 않은 부분에 대하여는 lm()과 동일한 구문을 사용하여 처음에 Pass를 Promo에 대하여 로지스틱 회귀로 모델링함.

A logistic regression model in R is fit as a generalized linear model (GLM) using a process similar to linear regression that we saw in Chap. 7 with lm(), but with the difference that a GLM can handle dependent variables that are not normally distributed. Thus, generalized linear models can be used to model data counts (such as number of purchases) or time intervals (such as time spent on a website) or binary variables (e.g., did/didn’t purchase). The common feature of all GLM models is that they relate normally distributed predictors to a non-normal outcome using a function known as a link. This means that they are able to fit models for many different distributions using a single, consistent framework.

In the present case, we model a binary outcome, and the appropriate distribution is a binomial distribution (see Sect. 6.3). There are multiple functions and packages that can estimate a GLM in R, but the most common is the glm(...) function. glm() takes an argument family= that specifies the distribution for the outcome variable. For a binary outcome, set family= binomial. The default link function for a binomial model is the logit function that we saw in Sect. 9.2.1, so we do not have to specify that. (But, as an example, if we wished to use a probit link function instead, we could specify family= binomial(link= "probit"), and similarly for other link functions.)

Our marketing question was, “does the promotion bundle have an effect on season pass sales?” and we model this initially with a logistic regression of Pass on Promo, using glm(..., family=binomial) and syntax otherwise identical to lm():

| > ###  > ### Logistic regression with glm()  >  > # initial logistic regression model  > pass.m1 <- glm(Pass ~ Promo, data=pass.df, family=binomial) |
| --- |

The initial model appears to confirm that the bundle is effective:

| > summary(pass.m1)  Call:  glm(formula = Pass ~ Promo, family = binomial, data = pass.df)  Deviance Residuals:  Min 1Q Median 3Q Max  -1.262 -1.097 1.095 1.095 1.260  Coefficients:  Estimate Std. Error z value Pr(>|z|)  (Intercept) -0.19222 0.05219 -3.683 0.000231 \*\*\*  PromoBundle 0.38879 0.07167 5.425 5.81e-08 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  (Dispersion parameter for binomial family taken to be 1)  Null deviance: 4375.0 on 3155 degrees of freedom  Residual deviance: 4345.4 on 3154 degrees of freedom  AIC: 4349.4  Number of Fisher Scoring iterations: 3 |
| --- |

- bundle 조건에 대해 양의 계수이고 그 효과는 통계적으로 유의함

- 0.3888의 계수는 무엇을 의미하는가? success(plogis() 사용)과 non-success(1 - success)의 비율을 조사하여 패스 판매와 프로모션 번들 요소의 연관성을 계산하는 데 사용할 수 있음. 이를 수행하는 수작업은 plogis()를 직접 사용하는 것임

There is a positive coefficient for the bundle condition, and the effect is statistically significant.

What does a coefficient of 0.3888 mean? We can use it to calculate the association of pass sales to the promotion bundle factor, by examining the ratio of success (using plogis()) to non-success (1 − success). A manual way to do this is to use plogis() directly:

| > # how the coef translates to an odds ratio  > plogis(0.3888) # outcome %  [1] 0.5959938  > plogis(0.3888) / (1-plogis(0.3888)) # ratio of outcome % to alternative %  [1] 1.475209 |
| --- |

**[해석]**

- 이것은 Buncle의 효과가 1.475의 추정 승산비임을 보여줌

- 이는 고객이 번들로 제공되는 패스를 구매할 가능성이 1.475배 더 높다는 것을 의미함.

- 이에 대해 또 다른 방법 해석은 번들이 구매 가능성을 47.5% 증가시킨다는 것임.

- 이것을 계산하는 더 쉽고 동등한 방법은 계수를 지수화하는 것임

This shows that the effect of Bundle is an estimated odds ratio of 1.475, meaning that customers are 1.475 times more likely to purchase the pass when it is offered in the bundle. Another way to think about this is that the bundle increases the purchase likelihood by 47.5%.An easier and equivalent way to calculate this is to exponentiate the coefficient:

| > exp(0.3888) # identical  [1] 1.475209 |
| --- |

- coef()로 계수를 추출하고 exp()를 사용하여 모델에서 승산비를 찾을 수 있음.

We can find the odds ratios from the model by extracting the coefficients with coef() and using exp():

| > # odds ratio for sales  > exp(coef(pass.m1))  (Intercept) PromoBundle  0.8251232 1.4751962 |
| --- |

- exp(confint (model))를 사용하여 승산비에 대한 신뢰 구간을 얻을 수 있음

We can obtain a confidence interval for the odds ratio using exp(confint (model)):

| > # confidence intervals  > exp(confint(pass.m1))  Waiting for profiling to be done...  2.5 % 97.5 %  (Intercept) 0.744749 0.9138654  PromoBundle 1.282055 1.6979776 |
| --- |

- 프로모션 번들에 대한 승산비는 1.28–1.70으로 추정되며 상당한 양의 효과임.

- 이것은 프로모션이 매우 효과적이라는 것을 보여준다고 볼 수 있는가? 반드시 그런 것은 아님.

효과는 모델이 우리가 해석하기 원하는 모델이라는 가정 하에 추정되기 때문임. 그런데, Pass ∼ Promo 모델이 과연 우리가 해석해야 할 모델인가?

The odds ratio for the promotion bundle is estimated to be 1.28–1.70, a significant positive effect. This demonstrates that the promotion is highly effective, right? Not necessarily, because the effects are estimated under the assumption that the model is the one we want to interpret. But is the model Pass ∼ Promo really the one we

should interpret?

***9.2.7 Reconsidering the Model 모델 재검토***

데이터를 더 탐색하면 흥미로운 사실을 알 수 있음. 채널별 시즌 패스 구매 table을 고려해 보자.

If we explore the data further, we notice something interesting. Consider a table of season pass purchases by channel:

| > ### at first it looks like the promotion is working  > ### but is this really the right model? check Channel  > table(pass.df$Pass, pass.df$Channel)    Email Mail Park  NoPass 568 727 272  YesPass 65 601 923 |
| --- |

- 시즌권 판매에 가장 성공한 채널은 프로모션 여부와 상관없이 공원이었음

- 표를 시각화하는 좋은 방법은 표의 개수에 해당하는 면적의 "타일"을 배치하는 모자이크 플롯을 사용하는 것임.

- vcd 패키지[140]는 모자이크 플롯을 생성하는 여러 가지 방법을 제공함(다소 명백한 mosaic() 함수 포함).

- 현재 데이터에서 관계를 특히 명확하게 하기 때문에 doubledecker plot을 사용함

The channel that was most successful in selling season tickets was at the park, regardless of whether the promotion was offered.

A good way to visualize tables is with mosaic plots, which lay out “tiles” whose areas correspond to counts in a table. The vcd package [140] provides several ways to create mosaic plots (including the rather obvious mosaic() function). We use a so-called doubledecker plot here as it makes the relationships particularly clear in the present data:

| > # visualization  > library(vcd) # install if needed  >  > doubledecker(table(pass.df))    Fig. 9.2 A mosaic plot created with doubledecker() [140] for sales of season passes by channel and promotion in simulated amusement park data. Season passes (“YesPass,” plotted as dark areas) are sold most frequently at the park and least frequently by email. The promotion bundle (“Bundle,” the second column within each channel) is associated with higher sales through the email channel, but lower sales in regular mail and at the park, thus showing an interaction effect |
| --- |

- 결과는 그림 9.2에 나와 있으며, 여기에서 세 개의 채널이 약간 다른 효과를 가지고 있음을 알 수 있음.

- 시즌 패스 판매는 공원에서 매우 성공적이며 이메일로는 성공적이지 않음.

- 이는 우리의 모델 Pass ∼ Promo가 부적절할 수 있고 Channel의 효과를 설명할 필요가 있음을 의미함.

- 모델 공식에 + Channel을 추가하여 채널의 효과를 모델링함

The result is shown in Fig. 9.2, where we see that the three channels have somewhat different effects. Sales of season passes are very successful at the park, and very unsuccessful by email. This implies that our model Pass ∼ Promo may be inadequate and needs to account for the effect of Channel.

We model a main effect of channel by adding + Channel to the model formula:

| > # Model 2: add the effect of channel  > pass.m2 <- glm(Pass ~ Promo + Channel, data=pass.df, family=binomial)  > summary(pass.m2)  Call:  glm(formula = Pass ~ Promo + Channel, family = binomial, data = pass.df)  Deviance Residuals:  Min 1Q Median 3Q Max  -1.9079 -0.9883 0.5946 0.7637 2.3272  Coefficients:  Estimate Std. Error z value Pr(>|z|)  (Intercept) -2.07860 0.13167 -15.787 < 2e-16 \*\*\*  PromoBundle -0.56022 0.09031 -6.203 5.54e-10 \*\*\*  ChannelMail 2.17617 0.14651 14.854 < 2e-16 \*\*\*  ChannelPark 3.72176 0.15964 23.313 < 2e-16 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  (Dispersion parameter for binomial family taken to be 1)  Null deviance: 4375.0 on 3155 degrees of freedom  Residual deviance: 3490.2 on 3152 degrees of freedom  AIC: 3498.2  Number of Fisher Scoring iterations: 4 |
| --- |

- 결과 모델은 promotion bundle의 강력한 부정적인 기여를 추정함. 승산비와 신뢰 구간을 계산함.

The resulting model now estimates a strong negative contribution of the promotion bundle. We compute the odds ratios and their confidence intervals:

| | > # updated coefs and odds ratios  > exp(coef(pass.m2))  (Intercept) PromoBundle ChannelMail ChannelPark  0.1251054 0.5710846 8.8125066 41.3371206  > exp(confint(pass.m2))  Waiting for profiling to be done...  2.5 % 97.5 %  (Intercept) 0.09577568 0.1606189  PromoBundle 0.47793969 0.6810148  ChannelMail 6.65770550 11.8328173  ChannelPark 30.42959274 56.9295369 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

- 이 모델에서 프로모션은 시즌 패스를 구매할 가능성이 32-53% 낮음(1.0-0.681에서 1.0-0.477 값 반영).

- 반면에 공원에서 직접 제안하면 이 모델에서 시즌 티켓 판매가 30~56배 더 높음.

- **그러나 이것이 적절한 모델인가? 채널별로 Promo가 다른 영향을 미칠 수 있는 상호작용 효과도 고려해야 하는가?**

- 데이터 탐색은 그림 9.2의 Email 채널에서 Bundle 영향에 대한 패턴이 극적으로 다르기 때문에 가능한 상호 작용 효과를 보여줌.

- : 연산자를 사용하여 교호 작용을 추가함

In this model, promotion is associated with a 32–53% lower likelihood (reflecting the values 1.0–0.681 to 1.0–0.477) of purchasing a season pass. On the other hand, offers in person at the park are associated with season ticket sales 30–56x higher in this model.

But is this the appropriate model? Shouldwealso consider an interaction effect, where Promo might have a different effect by Channel? Our data exploration suggests a possible interaction effect, especially because of the dramatically different pattern for the influence of Bundle in the Email channel in Fig. 9.2.

We add an interaction term using the : operator, as noted in Sect. 7.5:

| > # Model 3: add the interaction of promotion and channel  > pass.m3 <- glm(Pass ~ Promo + Channel + Promo:Channel,  + data=pass.df, family=binomial)  > summary(pass.m3)  Call:  glm(formula = Pass ~ Promo + Channel + Promo:Channel, family = binomial,  data = pass.df)  Deviance Residuals:  Min 1Q Median 3Q Max  -1.9577 -0.9286 0.5642 0.7738 2.4259  Coefficients:  Estimate Std. Error z value Pr(>|z|)  (Intercept) -2.8883 0.1977 -14.608 < 2e-16 \*\*\*  PromoBundle 2.1071 0.2783 7.571 3.71e-14 \*\*\*  ChannelMail 3.1440 0.2133 14.743 < 2e-16 \*\*\*  ChannelPark 4.6455 0.2510 18.504 < 2e-16 \*\*\*  PromoBundle:ChannelMail -2.9808 0.3003 -9.925 < 2e-16 \*\*\*  PromoBundle:ChannelPark -2.8115 0.3278 -8.577 < 2e-16 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  (Dispersion parameter for binomial family taken to be 1)  Null deviance: 4375.0 on 3155 degrees of freedom  Residual deviance: 3393.5 on 3150 degrees of freedom  AIC: 3405.5  Number of Fisher Scoring iterations: 5 |
| --- |

- 채널과 프로모션의 교호 작용은 통계적으로 중요함. 이러한 시뮬레이션된 데이터의 baseline(생략된) 이메일 채널과 달리 mail 및 in-park 채널에 대해 negative임.

The interaction of promotion with channel is statistically significant, and is strongly negative for the mail and in-park channels, as opposed to the baseline (omitted) email channel in these simulated data.

| > # updated coefs and odds ratios  > exp(confint(pass.m3))  Waiting for profiling to be done...  2.5 % 97.5 %  (Intercept) 0.03688720 0.08032263  PromoBundle 4.78970184 14.31465957  ChannelMail 15.54800270 35.97860059  ChannelPark 64.74364028 173.57861021  PromoBundle:ChannelMail 0.02795867 0.09102369  PromoBundle:ChannelPark 0.03135437 0.11360965 |
| --- |

- 승산비에서 프로모션이 이메일(생략된 참조 수준) 대비 mail 및 in-park 채널을 통해 2–11% 효과적임을 알 수 있음.

In the odds ratios, we see that the promotion is only 2–11% as effective through the mail and in-park channels as it is in email (the omitted reference level):

- 질문에 훨씬 더 나은 답을 얻음. 프로모션 번들이 효과적인가?

- 채널에 따라 다름. 이메일로 프로모션 캠페인을 계속해야 할 충분한 이유가 있지만, 성공했다고 해서 반드시 번들 프로모션이 공원에서 또는 일반 우편 캠페인을 통해 작동할 것이라는 의미는 아님.

- 통계 모델이 그림 9.2를 단순히 해석하는 데 어떤 이점이 있는지 궁금한 경우 한 가지 대답은 모델이 효과에 대한 신뢰 구간과 통계적 유의성을 추정한다는 것임.

We nowhave amuch better answer to our question. Is the promotion bundle effective? It depends on channel. There is good reason to continue the promotion campaign by email, but its success there does not necessarily imply the bundle promotion will work at the park or through a regular mail campaign. In case you’re wondering

how the statistical model is advantageous to simply interpreting Fig. 9.2, one answer is that the model estimates confidence intervals and statistical significance for the effect.

***9.2.8 Additional Discussion***

계층적 모델의 주제로 이동하기 전에 현재 섹션에 대한 몇 가지 고려 사항이 있음

• 놀이 공원 판매 데이터의 구조로 인해 범주형 예측 변수(요인 변수)를 사용하여 여기에서 로지스틱 회귀를 수행했지만 glm()에서 연속 예측 변수를 사용할 수도 있음. 7장에서 lm()으로 했던 것처럼 모델 공식의 오른쪽에 그것들을 추가하기만 하면 됨.

• 이러한 데이터에서 추정된 판촉 효과가 한 모델을 추정할 때 긍정적인 반면 다른 모델을 추정할 때는 부정적임을 보았고, 이는 모델을 모델링하거나 해석하기 전에 데이터를 철저히 탐색하는 것이 중요함을 보여줌. 대부분의 마케팅 데이터의 경우 어떤 모델도 절대적이지 않음. 그러나 여러 모델에 대한 신중한 데이터 탐색과 고려에도 불구하고 모델과 그로부터 도출된 추론에 대한 확신을 높일 수 있음.

• 여기에 있는 데이터는 Simpson 역설의 한 예임. 즉, 집계 효과의 추정치가 오해의 소지가 있고 기본 범주에서 볼 수 있는 효과와 현저히 다른 경우임. 유명한 예가 University of California at Berkeley의 대학원 입학에서 발생했음. 여기서 명백한 편향은 학과마다 전반적인 입학률과 지원자 수가 다르기 때문임[12]. R에서 Berkeley 데이터는 표준 데이터 세트 패키지의 UCBAdmissions 테이블로 사용할 수 있음.

로지스틱 회귀는 강력한 방법이며 이진 결과가 있는 많은 마케팅 문제에 특히 적합함. 자세한 내용은 섹션9.8 참조.

Before moving to the topic of hierarchical models, we have a few observations for the current section:

• Although we performed logistic regression here with categorical predictors (factor variables) due to the structure of the amusement park sales data, we could also

use continuous predictors in glm(). Just add those to the right hand side of the model formula as we did with lm() in Chap. 7.

• We saw that the estimated effect of promotion in these data was positive when we estimated one model, yet negative when we estimated another, and this shows that

it is crucial to explore data thoroughly before modeling or interpreting a model. Formostmarketing data, no model is ever definitive. However, though careful data exploration and consideration ofmultiple models, we may increase our confidence in our models and the inferences drawn from them.

• The data here are an example of Simpson’s paradox, which is when the estimate of an aggregate effect is misleading and markedly different than the effect seen in underlying categories. A famous example occurred in graduate admissions at the University of California at Berkeley, where an apparent bias in admissions was due instead to the fact that different departments had different overall admissions rates and numbers of applicants [12]. In R, the Berkeley data are available as the table UCBAdmissions in the standard datasets package.

Logistic regression is a powerful method and one that is a particularly good fit for many marketing problems that have binary outcomes. To learn more, see Sect. 9.8. For modeling product choice among sets of alternatives, we cover choice models in Chap. 13.

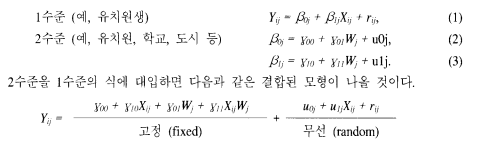
| | > install.packages('coefplot')  >  > ##########################################################  > ####  > #### ==> this section is NOT in the book  > #### extras on visualization for logistic coefficients  >  > # plot the coefs  > library(coefplot)  > coefplot(pass.m2, intercept=FALSE, outerCI=1.96, lwdOuter=1.5,  + title="Coefficients for Season Pass by Factor", ylab="Factor")  >  > #### plot the odds ratio confidence intervals  > ####  > pass.ci <- data.frame(confint(pass.m2)) # coef confidence intervals  Waiting for profiling to be done...  > pass.ci$X50 <- coef(pass.m2) # add the midpoint estimate  >  > # plot odds  > library(ggplot2)  > pass.ci$Factor <- rownames(pass.ci) # for ggplot2 to use in its model  > pass.ci  X2.5.. X97.5.. X50 Factor  (Intercept) -2.3457465 -1.8287206 -2.078599 (Intercept)  PromoBundle -0.7382707 -0.3841712 -0.560218 PromoBundle  ChannelMail 1.8957749 2.4708768 2.176172 ChannelMail  ChannelPark 3.4154156 4.0418143 3.721761 ChannelPark  >  > # ggplot of odds ratios  > # first: a plot by factor (x=) of the midpoint (y), high (ymax) and low (ymin)  > p <- ggplot(pass.ci[-1, ],  + aes(x=Factor, y=exp(X50), ymax=exp(X97.5..), ymin=exp(X2.5..)))  >  > # ... displaying those elements as points & error bars  > p <- p + geom\_point(size=4) + geom\_errorbar(width=0.25)  >  > # ... adding a vertical line at an odds ratio of 1.0 (no change)  > p <- p + geom\_hline(yintercept=1, linetype="dotted", size=1.5, color="red")  >  > # now plot it with titles  > p + ylab("Likehood by Factor (odds ratio, main effect)") +  + ggtitle(paste("95% CI: Card sign up odds by factor")) + coord\_flip()  >  >  > ### exercise for reader ==> NOT in book  > ### does this add anything to our interpretation? Intercept model  > pass.m3 <- glm(Pass ~ Promo \* Channel, data=pass.df, family=binomial)  > summary(pass.m3)    Call:  glm(formula = Pass ~ Promo \* Channel, family = binomial, data = pass.df)  Deviance Residuals:  Min 1Q Median 3Q Max  -1.9577 -0.9286 0.5642 0.7738 2.4259  Coefficients:  Estimate Std. Error z value Pr(>|z|)  (Intercept) -2.8883 0.1977 -14.608 < 2e-16 \*\*\*  PromoBundle 2.1071 0.2783 7.571 3.71e-14 \*\*\*  ChannelMail 3.1440 0.2133 14.743 < 2e-16 \*\*\*  ChannelPark 4.6455 0.2510 18.504 < 2e-16 \*\*\*  PromoBundle:ChannelMail -2.9808 0.3003 -9.925 < 2e-16 \*\*\*  PromoBundle:ChannelPark -2.8115 0.3278 -8.577 < 2e-16 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  (Dispersion parameter for binomial family taken to be 1)  Null deviance: 4375.0 on 3155 degrees of freedom  Residual deviance: 3393.5 on 3150 degrees of freedom  AIC: 3405.5  Number of Fisher Scoring iterations: 5    > ########################################################## | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

| **Key Points**  **Logistic Regression**  • Logistic regression relates a binary outcome such as purchase to predictors that may include continuous and factor variables, by modeling the variables’ association with the probability of the outcome (Sect. 9.2.1).  • A logistic regression model, also known as a logit model, is a member of the generalized linear models family, and is fit using **glm( , family=binomial)** (Sect. 9.2.6).  • Coefficients in a logit model can be interpreted in terms of odds ratios, the degree to which they are associated with the increased or decreased likelihood of an outcome. This is done simply by exponentiating the coefficients with exp() (Sect. 9.2.6).  • A statistically significant result does not always mean that the model is appropriate. It is important to explore data thoroughly and to construct models on the basis of careful consideration (Sect. 9.2.7). |
| --- |

**9.3 Hierarchical Models**

**\* 아래 양 칸으로 된 부분은 초벌 번역하려고 했던 부분인데 시간이 부족하여 그대로 둔 상태로 문맥이나 번역이 안 맞을 수도 있음**

| **9.3 Hierarchical Models**  In Chap. 7 we saw how to estimate a linear model for data for a sample of respondents. **What if we want to estimate the coefficients in the model for each respondent? As marketers, it can be very useful to determine individual-level effects such as which customers are more interested in a product or service, who among them wants which features, and who is most or least price sensitive.** We can use such information to see the diversity of preference or for purposes such as customer targeting or segmentation (see Chap. 11).  **To estimate both a population-level effect and an individual-level effect, we can use a hierarchical linear model (HLM).** The model is hierarchical because individual responses are governed by a lower-level linear model and coefficients for each individual follow a distribution across the population (the upper-level model). There are various algorithms to fit such models, but the general approach is that the algorithm fits the overall model to all the data, and then attempts to determine best fit for each individual within that overall estimate (and repeats as necessary).  In general, a data set for HLM at an individual level needs **multiple observations per individual**. Such observations may come from responses over time (as in transactions or a customer relationship management system) or from multiple responses at one time (as in a survey with repeated measures). **We consider the case of conjoint analysis, where a respondent rates multiple items on a survey at one time.**  How is this different from simply adding the individual, store, or other grouping variable as a factor variable in the model? **The key difference is that a factor variable would add a single term that adjusts the model up or down according to the individual.** In HLM, however, **we can estimate every coefficient—or any that we wish—for each individual**.  There are other uses for hierarchical models besides customer-level estimation. For example, one might wish to estimate differences by a factor such as geographic region, store, salesperson, product, or promotion campaign. Each of these might provide many responses that could be grouped and allow estimation of a group-level effect within an overall hierarchy. We can’t cover every application of HLM here— hierarchical models are the subject of entire books (e.g., Gelman and Hill [72])—yet we hope this discussion will help you to understand **when and how they may be useful, and how to begin with them in R**. | 9.3 Hierarchical Models  챕터7에서 우리는 응답자 표본에 대한 데이터에 대한 선형 모델을 추정하는 방법을 보았습니다. **각 응답자에 대한 모델의 계수를 추정하려면 어떻게 해야 합니까?** 마케터로서, **어떤 고객이 제품이나 서비스에 더 관심이 있는지, 그 중 누가 어떤 기능을 원하는지, 가장 가격에 민감한 사람이 누구인지와 같은 개인 수준의 효과를 결정하는 것이 매우 유용할 수 있습니다.** 우리는 이러한 정보를 사용하여 고객 타겟팅 또는 세분화와 같은 목적 또는 선호도의 다양성(11장 참조)을 알게 됩니다.  모집단 수준 효과와 개인 수준 효과를 모두 추정하기 위해 계층적 선형 모델(HLM)을 사용할 수 있습니다. 개별 응답이 하위 수준 선형 모델에 의해 관리되고 각 개인에 대한 계수가 모집단 전체의 분포를 따르기 때문에 모델은 계층적입니다(상위 수준 모델). 있다 이러한 모델에 맞는 다양한 알고리즘이 있지만 일반적인 접근 방식은 알고리즘이 전체 모델을 모든 데이터에 맞춘 다음 해당 전체 추정치 내에서 각 개인에 가장 적합한 것을 결정하려고 시도하는 것입니다(필요한 경우 반복).  일반적으로 개인 수준에서 HLM에 대한 데이터 세트는 개인당 여러 관찰이 필요합니다. 이러한 관찰은 시간 경과에 따른 응답(거래 또는 고객 관계 관리 시스템에서와 같이) 또는 한 번에 여러 응답에서 나올 수 있습니다(반복 측정을 사용한 설문조사에서와 같이). 응답자가 설문조사에서 한 번에 여러 항목을 평가하는 결합 분석의 경우를 고려합니다.  이것은 단순히 개인, 상점 또는 기타 그룹화 변수를 모델의 요인 변수로 추가하는 것과 어떻게 다릅니까? 주요 차이점은 요인 변수 개인에 따라 모형을 위 또는 아래로 조정하는 단일 항을 추가합니다. 그러나 HLM에서는 각 개인에 대한 모든 계수 또는 원하는 계수를 추정할 수 있습니다.  고객 수준 추정 외에 계층적 모델의 다른 용도가 있습니다. 예를 들어, 지리적 지역, 상점, 영업 사원, 제품 또는 판촉 캠페인과 같은 요인으로 차이를 추정하고자 할 수 있습니다. 이들 각각은 그룹화될 수 있는 많은 응답을 제공하고 그룹 수준의 추정을 허용할 수 있습니다. 전체 계층 내에서 효과를 나타냅니다. 여기에서 HLM의 모든 응용 프로그램을 다룰 수는 없습니다. 계층적 모델은 전체 책의 주제입니다(예: Gelman 및 Hill [72]). 그러나 이 논의가 언제 어떻게 유용할 수 있으며 어떻게 유용할 수 있는지 이해하는 데 도움이 되기를 바랍니다. R에서 시작합니다. |
| --- | --- |



| ***9.3.1 Some HLM Concepts***  A few words of jargon are required. **Hierarchical models distinguish two types of effects**. **One type is fixed effects, which are effects that are the same for every respondent.** In a standard linear model (Chap. 7) all effects are fixed effects. For instance, in Sect. 9.1.2, we saw that online spend was highly associated with online transactions and slightly associated with age. Both of those estimates are fixed effects that predict the same pattern of association for everyone in the sample.  An HLM also estimates **random effects**, **which are additional adjustments to the model coefficients estimated for each individual (or group)**. These are known as “random” because they are estimated as random variables that follow a distribution around the fixed estimates. However, for the estimate of each individual, they are best estimates according to the model, not random guesses in that sense.  Such models are also known as multilevel models, where individuals and the full sample are at different levels. They are a subset of models known as mixed effect models, where mixed reflects the fact that **the total effect for each respondent has (at least) two effects that are combined: the overall fixed effect plus the individual-level random effect.**  A final variation on mixed effects models is a **nested model,** where a factor of interest might occur only within subgroups of the total sample. For example, if we consider sales in response to different promotions that each occur at different stores, we might model both the effect of store (as a random effect, such that there are different sales intercepts for different stores) and the effect of promotion within store as a nested effect. We do not examine a nested model here, yet they may be also be fit using the lme4 package used below. | 9.3.1 Some HLM Concepts  전문용어 몇 마디가 필요합니다. 계층적 모델은 두 가지 유형의 효과를 구분합니다. 한 가지 유형은 모든 응답자에게 동일한 효과인 고정 효과입니다. 표준 선형 모델(7장)에서 모든 효과는 고정 효과입니다. 예를 들어, Sect. 9.1.2에서 우리는 온라인 지출이 온라인 거래와 높은 관련이 있고 연령과 약간의 관련이 있음을 확인했습니다. 이 두 추정치는 모두 표본의 모든 사람에 대해 동일한 연관 패턴을 예측하는 고정 효과입니다.  HLM은 또한 각 개인(또는 그룹)에 대해 추정된 모델 계수에 대한 추가 조정인 무작위 효과를 추정합니다. 이들은 고정 추정치 주변의 분포를 따르는 무작위 변수로 추정되기 때문에 "임의"라고 합니다. 그러나 각 개인의 추정은 그런 의미에서 무작위 추측이 아니라 모델에 따른 최선의 추정입니다.  이러한 모델은 개인과 전체 표본이 서로 다른 수준에 있는 다중 수준 모델이라고도 합니다. 그것들은 혼합 효과 모델로 알려진 모델의 하위 집합이며, 여기서 혼합은 각 응답자에 대한 총 효과가 결합된 두 가지 효과, 즉 전체 고정 효과와 개별 수준 무작위 효과가 있다는 사실을 반영합니다.  혼합 효과 모델의 최종 변형은 관심 요인이 전체 샘플의 하위 그룹 내에서만 발생할 수 있는 중첩 모델입니다. 예를 들어, 각기 다른 상점에서 발생하는 서로 다른 판촉에 대한 응답으로 판매를 고려하는 경우 상점의 효과(다른 상점에 대해 다른 판매 절편이 있는 것과 같은 임의의 효과로)와 판촉의 효과를 모두 모델링할 수 있습니다. 중첩 효과로 저장합니다. 여기서는 중첩 모델을 살펴보지 않지만 아래에 사용된 lme4 패키지를 사용하여 적합할 수도 있습니다. |
| --- | --- |

| ***9.3.2 Ratings-Based Conjoint Analysis for the Amusement Park***  For a hierarchical model, we return to the fictional amusement park from Sect. 7.1. **The park is now considering designs for a new roller coaster and hopes to find out which roller coaster features appeal to its customers.** They are considering coasters with various possible levels of maximum speed (40, 50, 60 or 70 mph), height (200, 300, or 400 feet), construction type (wood or steel), and theme (dragon or eagle). The stakeholders wish to know **which combination of features would be most popular according to customers’ stated preference**.  One way to examine this is a **survey that asks customers to rate different roller coasters** (illustrated with photographs or videos for more realism). For example:  On a 10 point scale, where 10 is the best and 1 is the worst, how would you rate a roller coaster that is made of wood, is 400 feet high, has a maximum speed of 50mph, with a dragon theme?  Customers’ ratings could be analyzed with a **linear model** where the ratings are predicted from the different features of the roller coasters. This would tell us the contribution of each feature to the total rating.  Additionally, we wish to understand these preferences at an individual level, such that we can see the distribution of preference or identify individuals for potential marketing actions. **To do this, we use a hierarchical linear model (HLM) that estimates both the overall fixed effect and the individual level random effect**.  In the following section we simulate consumers’ ratings for such a survey. The code is brief and illustrative of the data, but if you wish to skip the simulation, you can load the data from the book’s website: | **9.3.2 Ratings-Based Conjoint Analysis for the Amusement Park**  계층적 모델의 경우 Sect에서 가상의 놀이 공원으로 돌아갑니다. 7.1. 공원은 현재 새로운 롤러코스터의 디자인을 고려하고 있으며 어떤 롤러코스터 기능이 고객에게 어필하는지 알아내기를 희망합니다. 그들은 최대 속도(40, 50, 60 또는 70mph), 높이(200, 300 또는 400피트), 건축 유형(목재 또는 강철) 및 테마(용 또는 독수리)의 다양한 가능한 수준을 가진 코스터를 고려하고 있습니다. 이해 관계자는 고객의 명시된 선호도에 따라 어떤 기능 조합이 가장 인기가 있을지 알고 싶어합니다.  이를 조사하는 한 가지 방법은 고객에게 다양한 롤러코스터를 평가하도록 요청하는 설문조사입니다(좀 더 사실적인 사진이나 비디오로 설명됨). 예를 들어:  10이 최고이고 1이 최악인 10점 척도에서 드래곤 테마로 높이 400피트, 최고 속도 50mph의 나무로 만든 롤러코스터를 어떻게 평가하시겠습니까?  고객의 평가는 롤러코스터의 다양한 기능에서 평가가 예측되는 선형 모델로 분석할 수 있습니다. 이것은 총 평가에 대한 각 기능의 기여도를 알려줍니다.  또한 개인 수준에서 이러한 선호도를 이해하여 선호도 분포를 확인하거나 잠재적인 마케팅 활동에 대한 개인을 식별하고자 합니다. 이를 위해 전체 고정 효과 및 개인 수준 무작위 효과를 추정하는 계층적 선형 모델(HLM)을 사용합니다.  다음 섹션에서는 그러한 설문조사에 대한 소비자의 평가를 시뮬레이션합니다. 코드는 간단하고 데이터를 설명하지만 시뮬레이션을 건너뛰려면 책 웹사이트에서 데이터를 로드할 수 있습니다. |
| --- | --- |

데이터 로드

| ##########################################################  #####  ##### Hierarchical linear model  #####  ### Ratings-based conjoint analysis data  # alternative to load the data  conjoint.df <- read.csv("http://goo.gl/G8knGV")  conjoint.df$speed  <- factor(conjoint.df$speed)  conjoint.df$height <- factor(conjoint.df$height)  conjoint.df$const <- factor(conjoint.df$const)   # 추가 (이것으로 Steel, Wood )  summary(conjoint.df)  [결과]  resp.id rating speed height const  Min. : 1.00 Min. : 1.000 40: 800 200:1400 Steel:1400  1st Qu.: 50.75 1st Qu.: 3.000 50:1200 300:1200 Wood :1800  Median :100.50 Median : 5.000 60: 800 400: 600  Mean :100.50 Mean : 5.268 70: 400  3rd Qu.:150.25 3rd Qu.: 7.000  Max. :200.00 Max. :10.000  theme  Length:3200  Class :character  Mode :character |
| --- |

| ***9.3.3 Simulating Ratings-Based Conjoint Data***  In this section we simulate responses for a hypothetical conjoint analysis survey with 200 respondents who each rate the same set of 16 roller coaster profiles. If you have worked through the data simulation in previous chapters, this code should be relatively simple in structure, although a few functions are new.  We set the structure: 200 respondents who rate 16 designs, each with 4 roller coaster attributes: | \*\*\*9.3.3 Simulating Ratings-Based Conjoint Data\*\*\*  이 섹션에서 우리는 16개의 롤러코스터 프로파일의 동일한 세트를 각각 평가하는 200명의 응답자와 함께 가상의 결합 분석 설문조사에 대한 응답을 시뮬레이션합니다. 이전 장의 데이터 시뮬레이션을 통해 작업한 경우 이 코드는 몇 가지 새로운 기능이 있지만 구조가 비교적 단순해야 합니다.  우리는 구조를 설정했습니다. 16개의 디자인을 평가하는 200명의 응답자는 각각 4개의 롤러코스터 속성을 가지고 있습니다.  In this section we simulate responses for a hypothetical conjoint analysis survey with 200 respondents who each rate the same set of 16 roller coaster profiles. If you have worked through the data simulation in previous chapters, this code should be relatively simple in structure, although a few functions are new.  We set the structure: 200 respondents who rate 16 designs, each with 4 roller coaster attributes: |
| --- | --- |
| > ###  > ### Construct the data  > ###  > set.seed(12814)  > resp.id <- 1:200 # respondent ids  > nques <- 16 # number of conjoint ratings per respondent  > speed <- sample(as.factor(c("40", "50", "60", "70")), size=nques, replace=TRUE)  > height <- sample(as.factor(c("200", "300", "400")), size=nques, replace=TRUE)  > const <- sample(as.factor(c("Wood", "Steel")), size= nques, replace=TRUE)  > theme <- sample(as.factor(c("Dragon", "Eagle")), size=nques, replace=TRUE) | |

| In this example we assume that all respondents rate the same set of designs. Depending on your study’s goal, you might instead want to have a different, random set for each respondent. A single set of designs is convenient for printed surveys, while an online study could easily have a different set for every respondent; we will see an example in Chap. 13.  Next we create a model matrix for the combinations of features to rate. We draw multivariate random normal values for respondents’ preferences using **mvrnorm() from the MASS package** [192]: | 이 예에서는 모든 응답자가 동일한 디자인 세트를 평가한다고 가정합니다. 연구 목표에 따라 대신 각 응답자에 대해 다른 무작위 세트를 원할 수 있습니다. 단일 디자인 세트는 인쇄된 설문조사에 편리하지만 온라인 연구는 모든 응답자에 대해 다른 세트를 쉽게 가질 수 있습니다. 우리는 챕터에서 예를 볼 것입니다.  다음으로 평가할 기능 조합에 대한 모델 행렬을 만듭니다. MASS 패키지[192]에서 mvrnorm()을 사용하여 응답자의 선호도에 대한 다변량 무작위 정규 값을 그립니다. |
| --- | --- |

| > profiles.df <- data.frame(speed, height, const, theme)  > profiles.model <- model.matrix(~ speed + height + const + theme,  + data=profiles.df)  >  > library(MASS) # a standard library in R  > weights <- mvrnorm(length(resp.id),  + mu=c(-3, 0.5, 1, 3, 2, 1, 0, -0.5),  + Sigma=diag(c(0.2, 0.1, 0.1, 0.1, 0.2, 0.3, 1, 1))) |
| --- |

| **model.matrix() converts the list of design attributes (profiles.df) into coded variables**; it is similarly used by functions such as lm() to convert factors into variables for regression equations. You can compare profiles.model to profiles.df to see how this works. We use mvrnorm() to draw unique preference weights for each respondent. Estimating those later is the key feature that distinguishes a hierarchical model from a standard linear model.  Given the designs to be rated and individuals’ preferences, we compile the simulated individual ratings. For each respondent, we multiply the preference weights by the design matrix to get the total preference (utility) for each design, adding some random noise with rnorm(). We convert the utility to a 10-point rating scale using cut() (see Sect. 12.4.1), and add the respondent’s result to the overall data frame: | model.matrix()는 디자인 속성 목록(profiles.df)을 코드화된 변수로 변환합니다. lm()과 같은 함수에서 요인을 회귀 방정식의 변수로 변환하는 데 유사하게 사용됩니다. 어떻게 작동하는지 확인하려면 profiles.model과 profiles.df를 비교할 수 있습니다. mvrnorm()을 사용하여 각 응답자에 대한 고유한 선호도 가중치를 그립니다. 나중에 추정하는 것은 표준 선형 모델과 계층적 모델을 구별하는 핵심 기능입니다.  평가할 디자인과 개인의 선호도를 고려하여 시뮬레이션된 개별 평가를 집계합니다. 각 응답자에 대해 선호도 가중치를 designmatrix와 곱하여 각 디자인에 대한 총 선호도(유틸리티)를 얻고 rnorm()을 사용하여 임의의 노이즈를 추가합니다. cut()을 사용하여 유틸리티를 10점 평가 척도로 변환합니다(참조 섹션 12.4.1), 전체 데이터 프레임에 응답자의 결과를 추가합니다. |
| --- | --- |

| > # create df to hold the data  > # better would be to preallocate; but for small data set, building up is OK  > conjoint.df <- NULL # make sure there's no data yet  >  > # create per-respondent ratings  > for (i in seq\_along(resp.id)) {  + # create one respondent's ratings of the 16 items, plus error  + utility <- profiles.model %\*% weights[i, ] + rnorm(nques) # preference  + rating <- as.numeric(cut(utility, 10)) # put on a 10-point scale  + conjoint.resp <- cbind(resp.id=rep(i, nques), rating, profiles.df)  + # and add that respondent to the total data set  + conjoint.df <- rbind(conjoint.df, conjoint.resp)  + }  > summary(conjoint.df)  resp.id rating speed height const theme  Min. : 1.00 Min. : 1.000 40:1000 200: 800 Steel:1400 Dragon:1400  1st Qu.: 50.75 1st Qu.: 3.000 50: 600 300:1400 Wood :1800 Eagle :1800  Median :100.50 Median : 5.000 60:1200 400:1000  Mean :100.50 Mean : 5.191 70: 400  3rd Qu.:150.25 3rd Qu.: 7.000  Max. :200.00 Max. :10.000  > head(conjoint.df)  resp.id rating speed height const theme  1 1 10 70 300 Steel Eagle  2 1 1 40 400 Steel Dragon  3 1 2 40 400 Wood Eagle  4 1 5 40 400 Wood Eagle  5 1 4 40 200 Wood Dragon  6 1 7 70 400 Wood Dragon |
| --- |

| Building a data frame using rbind() repeatedly instead of preallocating a whole matrix is not efficient, but it is easy to understand and it is fast enough for this data set. For large data sets, it would be better to preallocate the data frame for the size needed and fill in the rows. With a bit of matrix manipulation, one might instead create the whole data frame at once; but a simple, readable method like the one here may be more effective overall if it’s easier and more reliable to code. | 전체 행렬을 사전 할당하는 대신 rbind()를 반복적으로 사용하여 데이터 프레임을 구축하는 것은 효율적이지 않지만 이해하기 쉽고 이 데이터 세트에 대해 충분히 빠릅니다. 큰 데이터 세트의 경우 필요한 크기에 대한 데이터 프레임을 미리 할당하고 행을 채우는 것이 좋습니다. 약간의 행렬 조작으로 전체 데이터 프레임을 한 번에 생성할 수 있습니다. 그러나 여기에 있는 것과 같은 간단하고 읽기 쉬운 방법이 더 쉽고 안정적으로 코딩할 수 있다면 전반적으로 더 효과적일 수 있습니다. |
| --- | --- |

| [참고] 데이터 프레임 결합 |
| --- |

***9.3.4 An Initial Linear Model***

We begin as always with a quick summary of our conjoint data to check it (create or load the data as described in Sect. 9.3.2 if needed):

| > summary(conjoint.df)  resp.id rating speed height const theme  Min. : 1.00 Min. : 1.000 40:1000 200: 800 Steel:1400 Dragon:1400  1st Qu.: 50.75 1st Qu.: 3.000 50: 600 300:1400 Wood :1800 Eagle :1800  Median :100.50 Median : 5.000 60:1200 400:1000  Mean :100.50 Mean : 5.191 70: 400  3rd Qu.:150.25 3rd Qu.: 7.000  Max. :200.00 Max. :10.000  > head(conjoint.df)  resp.id rating speed height const theme  1 1 10 70 300 Steel Eagle  2 1 1 40 400 Steel Dragon  3 1 2 40 400 Wood Eagle  4 1 5 40 400 Wood Eagle  5 1 4 40 200 Wood Dragon  6 1 7 70 400 Wood Dragon |
| --- |

Ratings of the designs range from 1 (strongly disprefer) to 10 (strong prefer). We also see the counts of the features that were shown in various combinations: speed, height, const and theme.

**Our goal is to determine how the four features relate to the ratings.** At an aggregate level, we might use by() to find the average rating for levels of each attribute.(총합수준에서 by()를 사용하여 각 특성의 평균 rating을 구함) For example, the averages by height are:

| > by(conjoint.df$rating, conjoint.df$height, mean)  conjoint.df$height: 200  [1] 3.32125  -----------------------------------------------------------------------  conjoint.df$height: 300  [1] 6.512143  -----------------------------------------------------------------------  conjoint.df$height: 400  [1] 4.836  [교재] |
| --- |

The average rating for designs with 300 foot height is 7.25 points on the 10-point scale, compared to 3.66 and 5.05 for heights of 200 and 400 feet. So, respondents prefer the middle of our height range. We could examine each individual feature in that way, but a more comprehensive linear model considers all of the effects in combination. To start, we’ll estimate a regular linear model without a hierarchical component using lm() (Chap. 7) – **기본 선형모형**:

| > # basic linear model  > ride.lm <- lm(rating ~ speed + height + const + theme, data=conjoint.df)  > summary(ride.lm)  Call:  lm(formula = rating ~ speed + height + const + theme, data = conjoint.df)  Residuals:  Min 1Q Median 3Q Max  -8.1247 -1.5113 -0.0864 1.4097 7.4887  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 2.27518 0.12391 18.362 < 2e-16 \*\*\*  speed50 0.68925 0.14134 4.877 1.13e-06 \*\*\*  speed60 1.65269 0.09923 16.655 < 2e-16 \*\*\*  speed70 4.43213 0.12652 35.032 < 2e-16 \*\*\*  height300 3.15851 0.12538 25.192 < 2e-16 \*\*\*  height400 1.64684 0.11450 14.383 < 2e-16 \*\*\*  constWood 0.23617 0.08735 2.704 0.00689 \*\*  themeEagle -0.74115 0.08489 -8.731 < 2e-16 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 2.067 on 3192 degrees of freedom  Multiple R-squared: 0.4609, Adjusted R-squared: 0.4597  F-statistic: 389.8 on 7 and 3192 DF, p-value: < 2.2e-16 |
| --- |

In this abbreviated output, the coefficients indicate the association with preference (the rating). The highest rated roller coaster on average would have a top speed of 70 mph, a height of 300 ft, steel construction, and the dragon theme (**steel and dragon because wood and eagle have negative values**). We estimate an overall rating for this **most-desired coaster**; it would be the **intercept + speed70 + height300 (steel and dragon are included in the intercept), or 3.07 + 4.49 + 2.94 = 10.46** points on our 10-point rating scale.

(이것은 단순히 "평균" 결과를 해석하는 것이 오해의 소지가 있을 수 있음)But wait! That’s not possible; our scale is capped at 10 points. This shows that simply interpreting the “**average” result can be misleading**. The coefficients are estimated on the basis of designs that mostly combine both desirable and undesirable attributes, and are not as reliable at the extremes of preference. Additionally, it could happen that few people prefer that exact combination even though the individual features are each best on average.

Consider that the coefficient for constWood is near zero. Are people indifferent between wood and steel coasters, or do they have strong preferences that cancel out when averaged? If people are strongly but almost equally divided, that’s important for us to know as marketers; it might suggest that we construct different rides that appeal to two different groups. On the other hand, if they are truly indifferent, we could choose between steel and wood on the basis of cost and other factors.

응답자를 더 잘 이해하기 위해 전체 평균 선호도 수준과 그룹 내 개별 선호도를 모두 추정하는 위계적 선형모형(**hierarchical model**)을 살펴봄

To understand our respondents better, we turn next to **a hierarchical model that will estimate both the overall average preference level and individual preferences within the group**.

| ***9.3.5 Initial Hierarchical Linear Model with lme4***  The linear model ride.lm has only fixed effects that are estimated at the sample level. **In a hierarchical linear model, we add one or more individual-level effects to those**.  The simplest HLM allows individuals to vary only in terms of the constant intercept. For example, we might expect that individuals vary in their usage of a rating scale such that some will rate our roller coaster designs higher or lower than the average respondent. This would be an individual-level random effect for the intercept term. To estimate an HLM with fixed effects plus a per-respondent intercept, we change the lm() model from above in three ways. First, instead of lm(), we use a **hierarchical estimation function, lmer() from the lme4 package** [8].  Second, in the formula for lmer(), **we specify the term(s) for which to estimate random effects.** For the intercept, that is signified as simply “1”. Third, we specify the grouping variable, for which a random effect will be estimated for each unique group. In our conjoint data, the group in the set of responses for a single respondent, which is identified in the data frame by respondent number, resp.id. With lme4, we specify the random effect and grouping variable with syntax using a vertical bar (“|”) as + (predictors | group), or in this case for the intercept only, + (1 | resp.id).  We estimate this model using lme4, where the only difference from the call to lm() above is the addition of a term for random intercept by respondent: | ***9.3.5 Initial Hierarchical Linear Model with lme4***  선형 모델 ride.lm에는 샘플 수준에서 추정되는 고정 효과만 있습니다. 계층적 선형 모델에서는 여기에 하나 이상의 개별 수준 효과를 추가합니다.  가장 단순한 HLM은 개인이 일정한 절편의 관점에서만 변하도록 허용합니다. 예를 들어, 우리는 개인이 평가 척도를 사용하는 방식이 다양하여 일부 응답자는 평균 응답자보다 높거나 낮은 평가를 받을 것이라고 예상할 수 있습니다. 이것은 절편 항에 대한 개별 수준의 무작위 효과입니다. 고정 효과와 응답자당 절편으로 HLM을 추정하기 위해 위에서 3가지 방법으로 lm() 모델을 변경합니다. 먼저 lm() 대신 lme4 패키지[8]의 계층적 추정 함수인 lmer()를 사용합니다.  둘째, lmer()의 공식에서 무작위 효과를 추정할 용어를 지정합니다. 절편의 경우 단순히 "1"로 표시됩니다.  셋째, 각 고유 그룹에 대해 무작위 효과를 추정할 그룹화 변수를 지정합니다. 결합 데이터에서 단일 응답자에 대한 응답 집합의 그룹은 데이터 프레임에서 응답자 번호 resp.id로 식별됩니다.  lme4에 있어 수직 막대("|")를 사용하여 +(예측자 | 그룹) 또는 이 경우 절편에 대해서만 +(1 | resp.id)라는 구문으로 무작위 효과 및 그룹화 변수를 지정합니다.  위의 lm() 호출과의 유일한 차이점은 응답자별 무작위 절편(random intercept)이 추가된 lme4를 사용하여 이 모델을 추정합니다. |
| --- | --- |

| | > # hierarchical model  > library(lme4)  > # 0 = HLM with intercept only  > # model with random intercept by respondent = (1 | resp.id)  > ride.hlm1 <- lmer(rating ~ speed + height + const + theme + (1 | resp.id),  + data=conjoint.df)  >  > summary(ride.hlm1)  Linear mixed model fit by REML ['lmerMod']  Formula: rating ~ speed + height + const + theme + (1 | resp.id)  Data: conjoint.df  REML criterion at convergence: 13664  Scaled residuals:  Min 1Q Median 3Q Max  -3.7314 -0.6968 0.0034 0.6608 3.3952  Random effects:  Groups Name Variance Std.Dev.  resp.id (Intercept) 0.315 0.5613  Residual 3.957 1.9892  Number of obs: 3200, groups: resp.id, 200  Fixed effects:  Estimate Std. Error t value  (Intercept) 2.27518 0.12569 18.101  speed50 0.68925 0.13604 5.067  speed60 1.65269 0.09551 17.304  speed70 4.43213 0.12178 36.396  height300 3.15851 0.12068 26.173  height400 1.64684 0.11021 14.943  constWood 0.23617 0.08407 2.809  themeEagle -0.74115 0.08171 -9.071  Correlation of Fixed Effects:  (Intr) sped50 sped60 sped70 hgh300 hgh400 cnstWd  speed50 -0.147  speed60 -0.600 0.351  speed70 -0.189 0.389 0.324  height300 -0.548 -0.498 0.191 -0.236  height400 -0.650 0.020 0.432 -0.085 0.595  constWood -0.458 -0.321 0.148 -0.074 0.463 0.231  themeEagle -0.049 0.333 -0.137 0.130 -0.367 -0.213 -0.395 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

**(고정효과는 상기 lm()와 동일) In this output, we see that the fixed effects are identical to those estimated by lm() above.** But now we have also estimated **a unique intercept term adjustment for each respondent**. The output section labeled **“Random effects” shows 3200 total observations** (survey questions) grouped into 200 respondents for which a random effect was estimated (such as the effect for (Intercept)).

fixef() is an easy way to extract just the fixed (population level) effects:

| > fixef(ride.hlm1)  (Intercept) speed50 speed60 speed70 height300 height400 constWood  2.2751812 0.6892458 1.6526910 4.4321327 3.1585113 1.6468438 0.2361680  themeEagle  -0.7411508 |
| --- |

The 200 respondent-level random effect estimates for intercept, which summary (ride.hlm1) does not display because there could be many of them, are accessed with ranef() (and we additionally use head() to shorten the output):

| > head(ranef(ride.hlm1)$resp.id)  (Intercept)  1 -0.4919233  2 0.1733110  3 0.5584467  4 -0.2118246  5 -0.1067876  6 -0.7720219 |
| --- |

**The complete effect for each respondent** comprises the **overall fixed effects that apply to everyone, plus the individually-varying random effects** (in this case, just the intercept). Those are accessed using coef():

| | > head(coef(ride.hlm1)$resp.id)  (Intercept) speed50 speed60 speed70 height300 height400 constWood themeEagle  1 1.783258 0.6892458 1.652691 4.432133 3.158511 1.646844 0.236168 -0.7411508  2 2.448492 0.6892458 1.652691 4.432133 3.158511 1.646844 0.236168 -0.7411508  3 2.833628 0.6892458 1.652691 4.432133 3.158511 1.646844 0.236168 -0.7411508  4 2.063357 0.6892458 1.652691 4.432133 3.158511 1.646844 0.236168 -0.7411508  5 2.168394 0.6892458 1.652691 4.432133 3.158511 1.646844 0.236168 -0.7411508  6 1.503159 0.6892458 1.652691 4.432133 3.158511 1.646844 0.236168 -0.7411508 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

여러 그룹화 요소(계층적 수준)에 대한 무작위 효과를 추정할 수 있으므로 이러한 효과는 $resp.id라는 그룹화 변수의 이름을 사용하여 추출해야 합니다. coef(ride.hlm1)$resp.id에서 각 응답자는 절편을 제외한 모든 계수에 대한 효과의 전체 샘플 수준 값을 가지며 최종 절편 계수는 고정 효과에 무작위 효과를 더한 것과 같습니다. 예를 들어, 응답자 1의 경우 절편은 3.07(fixef) −0.65(ranef) = 2.42(coef)입니다.

**It is possible to estimate random effects for multiple grouping factors (hierarchical levels), so these effects must be extracted using the name of the grouping variable**, which is $resp.id. In coef(ride.hlm1)$resp.id, each respondent has the overall sample-level value of the effect on all coefficients except for intercept, and the final intercept coefficient is the same as the fixed effect plus the random effect. For example, for respondent 1, the intercept is 3.07 (fixef) −0.65 (ranef) = 2.42 (coef) .

***9.3.6 Complete Hierarchical Linear Model***

마케팅 관행에서 가장 일반적인 계층 모델은 모든 응답자에 대한 모든 관심 계수에 대한 무작위 효과 매개변수를 추정하는 것임. 이것은 lme4 구문으로 쉽게 수행 가능. **(predictors | group)**의 예측자에 관심 있는 모든 변수를 추가하기만 하면 됨.

The most common hierarchical model in marketing practice is to estimate a random effect parameter for every coefficient of interest for every respondent. This is easy to do with the **lme4** syntax; simply add all the variables of interest to the predictors in the random effects specification **(predictors | group)**.

결합 데이터의 경우 공식의 무작위 효과 부분을 (속도 + 높이 + const + 테마 | resp.id)로 씁니다. 그 모델을 추정하기 전에 이것이 위의 절편 모델보다 훨씬 더 복잡한 모델이라는 점에 유의해야 합니다. random intercept-only HLM 이 8개의 고정 매개변수와 200개의 무작위 효과를 추정한 반면, 전체 모델은 8개의 고정 효과와 8 \* 200개의 무작위 효과를 추정합니다. 그리고 총 3200개의 관찰 데이터 프레임에 대해 이 작업을 수행합니다.

이 사실은 두 가지 의미를 갖는다. 첫째, 추정이 다소 느릴 수 있으며 작성 당시 현재 모델에 대해 몇 분이 소요됩니다. 둘째, 3200개의 관측값도 많지 않을 정도로 매개변수가 너무 많아 안정적인 수렴 모델을 찾는 데 다소 어려움이 있을 수 있습니다.

이러한 사실을 염두에 두고 전체 모델을 다음과 같이 추정합니다(시간이 걸릴 수 있으며 몇 분 정도 소요될 수 있음).

For the conjoint data, we write the random effects part of the formula as (speed + height + const + theme | resp.id). Before estimating that model, we should note that this is a much more complex model than the intercept model above. Whereas the random intercept-only HLM estimated 8 fixed parameters and 200 random effects, the full model will estimate 8 fixed effects plus 8 ∗ 200 random effects. And it will do this for a total data frame of 3200 observations.

This fact has **two implications**. First, the estimation can be rather slow, taking several minutes for the present model at the time of writing. Second, there are so many parameters that even 3200 observations is not a lot, and one can expect some difficulty finding a stable converged model.

With those facts in mind, we estimate the full model as follows (this will take some time, perhaps several minutes):

| [교재 코드 시도 1 – 에러 발생]  > # model with random intercept & slope by respondent = (predictors | resp.id)  > #  > # consider 1M+ iterations; using 100K for somewhat faster time (~5 min)  > # WARNING: slow, takes several minutes!  > #  > # NOTE: limited degrees of freedom and overfitting in this data set!  > # we will use the "Nelder-Mead" algorithm to force optimization  > #  > ride.hlm2 <- lmer(rating ~ speed + height + const + theme +  + (speed + height + const + theme | resp.id),  + data=conjoint.df,  + control=lmerControl(optCtrl=list(maxfun=100000),  + optimizer ="Nelder\_Mead")) # default = 10000  Warning messages:  1: In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  unable to evaluate scaled gradient  2: In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  Model failed to converge: degenerate Hessian with 1 negative eigenvalues  [교재 수정 시도 2] (참고: <https://stats.stackexchange.com/questions/242109/model-failed-to-converge-warning-in-lmer>)  > # model with random intercept & slope by respondent = (predictors | resp.id)  > #  > # consider 1M+ iterations; using 100K for somewhat faster time (~5 min)  > # WARNING: slow, takes several minutes!  > #  > # NOTE: limited degrees of freedom and overfitting in this data set!  > # we will use the "Nelder-Mead" algorithm to force optimization  > #  > ride.hlm2 <- lmer(rating ~ speed + height + const + theme +  + (speed + height + const + theme | resp.id),  + data=conjoint.df, REML = FALSE,  + control=lmerControl(optCtrl=list(maxfun=100000),  + optimizer ="Nelder\_Mead")) # default = 10000  Warning message:  In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  Model failed to converge with max|grad| = 29.1079 (tol = 0.002, component 1)  > |
| --- |

Compared to model ride.hlm1 above, this model has two changes. First, we added all four roller coaster factors to be estimated for random effects. This will give us individual estimates of preference for every feature, for each respondent. Second, we added a control argument to lmer(), which increases the maxfun number of iterations to attempt convergence from 10,000 iterations (the default) to 100,000. This allows the model to converge better, although it may issue warnings in some cases that depend on the package versions. (If so, ignore the warnings for now. Our discussion here is for illustration, not for an important business decision. For a model of importance, we recommend to run to convergence when possible.)

**When you run into warnings, we suggest five potential remedies.** First, increase the control maxfun argument by a factor of 2, 5, or 10 to see if convergence results (and repeat that if necessary). Second, check whether the max|grad| (maximum absolute value of the gradient in the optimization function; cf. [8]) is small, such as max < 0.001; if so, you may be okay. Alternatively, if max >> .01, such as max = 0.10, increase the iterations. Third, do a web search for the warnings you receive and consider the suggestions offered on R discussion forums. Fourth, consider using a different optimization function (see lme4 documentation [8]). Fifth, consider collecting more data, or evaluate your data for internal consistency. Again, we skip these steps now primarily for convenience.

Fixed effects are extracted with fixef():

| > # population estimate  > fixef(ride.hlm2)  (Intercept) speed50 speed60 speed70 height300 height400 constWood  2.2751812 0.6892458 1.6526910 4.4321327 3.1585113 1.6468438 0.2361680  themeEagle  -0.7411508 |
| --- |

(coefficient는 동일) This part of the ride.hlm2 model is identical to the model estimated for ride.hlm1 above, so the coefficients are identical.

The random effects now include an estimate for each parameter for each respondent. Again, because we grouped by resp.id and could have had multiple grouping factors, we request the $resp.id portion of the random effects using ranef():

| | > # individual estimates, 1 row per respondent  > # just the random (individual variation) part  > head(ranef(ride.hlm2)$resp.id)  (Intercept) speed50 speed60 speed70 height300 height400 constWood themeEagle  1 -0.4547890 -0.5667905 -0.4126250 -0.17795472 -0.07594611 -0.6281164 0.4146259 -0.07819392  2 -0.1925079 0.3508743 0.3539491 0.13179349 -1.16386853 0.6390667 0.7125488 0.37984364  3 1.4801985 1.1758509 0.4729510 0.15613420 -0.34228565 1.1466328 -0.3585846 -1.92805912  4 -0.7032868 -0.4230287 -0.3695183 -0.19672129 -1.11991556 0.1467960 1.9384813 -0.05523270  5 1.5145722 0.7398731 1.1183968 0.56293280 -0.37629862 -0.1635045 -2.6906027 -1.04981876  6 -1.0616826 -0.7266240 -0.1564066 -0.02821501 -0.11531446 0.3000688 -0.7066539 0.93385544 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

(상기 모델 ride.hlm1과 상이함) Notice that the random intercepts are no longer identical to those estimated in model ride.hlm1, because we added seven explanatory variables and the predicted outcome rating points are distributed differently across the predictors.

We obtain the total coefficients for each respondent with coef():

| | > # the total effect for each respondent (fixed + random)  > head(coef(ride.hlm2)$resp.id)  (Intercept) speed50 speed60 speed70 height300 height400 constWood themeEagle  1 1.820392 0.12245534 1.240066 4.254178 3.082565 1.018727 0.6507938 -0.8193447  2 2.082673 1.04012012 2.006640 4.563926 1.994643 2.285911 0.9487168 -0.3613072  3 3.755380 1.86509672 2.125642 4.588267 2.816226 2.793477 -0.1224166 -2.6692100  4 1.571894 0.26621709 1.283173 4.235411 2.038596 1.793640 2.1746493 -0.7963835  5 3.789753 1.42911897 2.771088 4.995066 2.782213 1.483339 -2.4544347 -1.7909696  6 1.213499 -0.03737819 1.496284 4.403918 3.043197 1.946913 -0.4704860 0.1927046 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

Notice that the coefficients for constWood vary widely across respondents. Respondent 2 has a strong preference for wood over steel (1.42), while respondent 1 is almost indifferent (0.11). Histograms of the coefficients for each respondent can illustrate these large differences between customers, as we will illustrate in Sect. 9.4.3.

As a final sanity check to confirm that the model matches expectations, we choose a respondent (ID 196) and see that the coefficients are indeed the sum of the fixed and random effects:

| | > # the coefficients are the fixed effects + individual (random) effects  > # demonstrating this for an arbitrary respondent (id #196)  > fixef(ride.hlm2) + ranef(ride.hlm2)$resp.id[196, ]  (Intercept) speed50 speed60 speed70 height300 height400 constWood themeEagle  196 1.675721 -0.2049656 1.114377 4.220156 3.27549 2.861334 0.2412267 -0.8680005  > coef(ride.hlm2)$resp.id[196, ]  (Intercept) speed50 speed60 speed70 height300 height400 constWood themeEagle  196 1.675721 -0.2049656 1.114377 4.220156 3.27549 2.861334 0.2412267 -0.8680005 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

In this code, the random effect and coefficient values for respondent 196 are retrieved by indexing that row within the corresponding $resp.id matrix.

***9.3.7 Conclusion for Classical HLM***

This concludes our discussion of classical hierarchical models; in the next section, we consider the Bayesian approach to HLM, which uses the same general conceptual model but a different estimation method.

이 섹션에서 관심 있는 개인 또는 기타 그룹화 요인에 대해 여러 관측치가 있는 경우 표본 수준 및 개인 또는 그룹 수준 효과를 모두 추정하는 계층적 모델을 고려해야 함. 이러한 모델은 **lme4 패키지를 사용**하여 비교적 간단하게 추정할 수 있음.

In this section, we hope to have convinced you that, when you have multiple observations for an individual or other grouping factor of interest, you should consider a hierarchical model that estimates both sample-level and individual- or group-level effects. These models are relatively straightforward to estimate using the lme4 package.

마케팅에서 가장 일반적인 고객 수준 모델 외에도 계층적 모델을 추정할 수 있는 다른 요소에는 매장, 국가, 지리적 지역, 광고 캠페인, 광고 크리에이티브, 채널, 번들 및 브랜드가 있음.

Besides customer-level models, which are most common in marketing, other factors for which one might wish to estimate a hierarchical model include store, country, geographic region, advertising campaign, advertising creative, channel, bundle, and brand.

If this section has inspired you to consider adding hierarchical modeling to your toolbox, see “Learning More” (Sect. 9.8) for pointers to other resources.

[추가]

| | > ### BEGIN: this section NOT IN BOOK  > ### LME also has a distribution; e.g., standard errors  > library(arm)  arm (Version 1.11-2, built: 2020-7-27)  Working directory is C:/Users/박규서/Documents  Warning message:  패키지 ‘arm’는 R 버전 4.0.5에서 작성되었습니다  > head(se.ranef(ride.hlm2)$resp.id)  (Intercept) speed50 speed60 speed70 height300 height400 constWood themeEagle  1 0.5459842 0.6220761 0.4567278 0.2178071 0.588196 0.6980605 0.6877928 0.6895545  2 0.5459842 0.6220761 0.4567278 0.2178071 0.588196 0.6980605 0.6877928 0.6895545  3 0.5459842 0.6220761 0.4567278 0.2178071 0.588196 0.6980605 0.6877928 0.6895545  4 0.5459842 0.6220761 0.4567278 0.2178071 0.588196 0.6980605 0.6877928 0.6895545  5 0.5459842 0.6220761 0.4567278 0.2178071 0.588196 0.6980605 0.6877928 0.6895545  6 0.5459842 0.6220761 0.4567278 0.2178071 0.588196 0.6980605 0.6877928 0.6895545  > ranef(ride.hlm2)$resp.id[196, ] - 1.96\*se.ranef(ride.hlm2)$resp.id[1,]  (Intercept) speed50 speed60 speed70 height300 height400 constWood themeEagle  196 -1.669589 -2.113481 -1.433501 -0.6388783 -1.035885 -0.1537082 -1.343015 -1.478377  > ranef(ride.hlm2)$resp.id[196, ] + 1.96\*se.ranef(ride.hlm2)$resp.id[1,]  (Intercept) speed50 speed60 speed70 height300 height400 constWood themeEagle  196 0.4706691 0.3250577 0.3568723 0.2149256 1.269843 2.582689 1.353133 1.224677  > ###  > ### END: not in book | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

| **Key points**  **Hierarchical Linear Models**  • In common marketing discussion, a hierarchical model estimates both group level effects and individual differences in effects. Such models are popular in marketing because they provide insight into differences among customers (heterogeneity) and distribution of preference. Hierarchical linear models (HLM) are exemplified when we estimate the importance of effects for individuals as well as for an overall population (Sect. 9.3).  • Effects that are associated with all observations are known as **fixed effects**, and those that differ across various grouping levels are known as **random effects** (Sect. 9.3.1).  • These models are also known as **mixed effect models**, because the total effect for each person is composed of the effect for the overall population (the fixed effect) plus the per-individual (random) effect. We estimated an **HLM using lmer() from the lme4 package** (Sect. 9.3.5).  • The difference between estimating hierarchical effects, as opposed to including the grouping variable as a factor in a standard linear model, is that a hierarchical model estimates every specified effect for each individual or group, not only a single adjustment term.  • The formula for a mixed effect model includes a **grouping term, + ( ... |group)**. Common models have a different intercept by group using (1 | group) or different intercepts and slopes for predictors within each group using (predictor | group) (Sects. 9.3.5, 9.3.6). To estimate an individual level model, the grouping term is typically the respondent identifier.  • Hierarchical models can be used to group observations **at other levels** than the individual level. For example, we might wish to group by store, advertising campaign, salesperson, or some other factor, if we want to estimate effects that are specific to such a grouping (Sect. 9.3.7).  • A common marketing application of HLM is conjoint analysis, to estimate **both overall preference and individual differences in preference**. In this chapter, we demonstrated ratings-based, or metric conjoint analysis (Sect. 9.3.2). |
| --- |

**9.4 Bayesian Hierarchical Linear Models\***

This is an optional section that you may skip if you are not interested in the Bayesian approach to estimate hierarchical models.

Hierarchical models may be fit with classical estimation procedures (such as the lme4 package we saw above) yet they are particularly well-suited to Bayesian estimation. The method we use here is known as a hierarchical Bayes approach; *hierarchical* because it models individuals in relationship to an overarching distribution, and Bayes because it uses Bayesian estimation techniques to fit the models (see Sects. 6.6.1 and 6.6.2 for an introduction).

계층적 모델은 고전적 추정 절차(예: 위에서 본 lme4 패키지)에 적합할 수 있지만 특히 베이지안 추정에 적합함. 여기에서 사용하는 방법은 계층적 Bayes 접근 방식으로 알려져 있음. 가장 중요한 분포와 관련하여 개인을 모델링하기 때문에 계층적이고, 모델에 맞추기 위해 베이지안 추정 기술을 사용하기 때문에 Bayes임 (소개는 섹션 6.6.1 및 6.6.2 참조).

In this section, we apply a hierarchical Bayes (HB) method to estimate the hierarchical linear model for ratings-based (metric) conjoint analysis, using the same data set that we analyzed with classical hierarchical models in Sect. 9.3 above. Before continuing this section you should:

• Review the concepts of Bayesian linear models and MCMC estimation in Sect. 7.5.4

• Review the concepts of hierarchical linear models in Sects. 9.3 and 9.3.1

• Review the description of the amusement park conjoint analysis data in Sect. 9.3.2

Download the simulated amusement park conjoint analysis data as follows, or see Sect. 9.3.2:

| > conjoint.df <- read.csv("http://goo.gl/G8knGV")  > conjoint.df$speed <- factor(conjoint.df$speed)  > conjoint.df$height <- factor(conjoint.df$height)  > conjoint.df$const <- factor(conjoint.df$const)  > conjoint.df$theme <- factor(conjoint.df$theme)  > summary(conjoint.df)  resp.id rating speed height const theme  Min. : 1.00 Min. : 1.000 40: 800 200:1400 Steel:1400 Dragon:1600  1st Qu.: 50.75 1st Qu.: 3.000 50:1200 300:1200 Wood :1800 Eagle :1600  Median :100.50 Median : 5.000 60: 800 400: 600  Mean :100.50 Mean : 5.268 70: 400  3rd Qu.:150.25 3rd Qu.: 7.000  Max. :200.00 Max. :10.000 |
| --- |

***9.4.1 Initial Linear Model with MCMCregress()\****

- 베이지안 방법을 사용하여 비계층적 모델을 추정하는 것으로 시작. 이를 통해 복잡한 모델을 시도하기 전에 기본 추정 절차가 작동하는지 확인할 수 있음

We start by estimating a non-hierarchical model using Bayesian methods, which allows us to check that our basic estimation procedures are working before we attempt a complex model. We model respondents’ ratings of roller coaster designs as a function of roller coaster features using MCMCregress() to fit a simple linear model as we did in Sect. 7.5.4:

| | > install.packages('MCMCpack')  > ####  > #### Hierarchical Bayes linear model, metric conjoint analysis  > ####  >  > ### LOAD conjoint.df DATA AS ABOVE  >  > # standard lm with MCMC  > library(MCMCpack) # install if needed  필요한 패키지를 로딩중입니다: coda  다음의 패키지를 부착합니다: ‘coda’  The following object is masked from ‘package:arm’:  traceplot  ##  ## Markov Chain Monte Carlo Package (MCMCpack)  ## Copyright (C) 2003-2022 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park  ##  ## Support provided by the U.S. National Science Foundation  ## (Grants SES-0350646 and SES-0350613)  ##  Warning messages:  1: 패키지 ‘MCMCpack’는 R 버전 4.0.5에서 작성되었습니다  2: 패키지 ‘coda’는 R 버전 4.0.5에서 작성되었습니다  > set.seed(97439)  > ride.mc1 <- MCMCregress(rating ~ speed + height + const + theme,  + data=conjoint.df)  > summary(ride.mc1)  Iterations = 1001:11000  Thinning interval = 1  Number of chains = 1  Sample size per chain = 10000  1. Empirical mean and standard deviation for each variable,  plus standard error of the mean:  Mean SD Naive SE Time-series SE  (Intercept) 3.0729 0.08112 0.0008112 0.0008112  speed50 0.8208 0.11061 0.0011061 0.0011126  speed60 1.5754 0.12889 0.0012889 0.0012889  speed70 4.4873 0.15002 0.0015002 0.0015002  height300 2.9444 0.09122 0.0009122 0.0009337  height400 1.4461 0.12934 0.0012934 0.0013367  constWood -0.1187 0.11310 0.0011310 0.0011310  themeEagle -0.7533 0.11308 0.0011308 0.0011308  sigma2 3.8705 0.09737 0.0009737 0.0009737  2. Quantiles for each variable:  2.5% 25% 50% 75% 97.5%  (Intercept) 2.9175 3.0175 3.0722 3.12754 3.2349  speed50 0.6027 0.7470 0.8225 0.89443 1.0331  speed60 1.3230 1.4872 1.5766 1.66176 1.8302  speed70 4.1944 4.3855 4.4850 4.59031 4.7802  height300 2.7640 2.8834 2.9450 3.00725 3.1190  height400 1.1920 1.3582 1.4442 1.53329 1.7048  constWood -0.3403 -0.1953 -0.1190 -0.04138 0.1023  themeEagle -0.9714 -0.8301 -0.7522 -0.67797 -0.5303  sigma2 3.6839 3.8039 3.8683 3.93464 4.0691 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

**전반적인 효과는 Sect9.3.5의 고전적 선형 모델에 의해 추정된 것과 거의 동일함.**

MCMCregress를 사용하여 추정 루틴을 변경했지만 모델은 변경하지 않았음.

**데이터가 충분하면 동일한 모델에 대해 베이지안 및 빈도주의적 추정치가 거의 동일할 것입니다.**

The overall effects are nearly identical to those estimated by the classical linear models in Sect. 9.3.5, which is it should be. We changed the estimation routine by using MCMCregress, but we did not change the model. With sufficient data, Bayesian and frequentist estimates will be nearly identical for the same model. Now that we’ve confirmed that our esimation algorithm is working, **we are ready to add the hierarchical component to the model.**

***9.4.2 Hierarchical Linear Model with MCMChregress()\****

MCMChregress(fixed, random, group, data, r, R)를 사용하여 계층적 모델을 추정함

이것은 고정 효과 사양과 임의 효과 사양을 구분하기 때문에 lme4에서 사용하는 것과 약간 다른 구문임

We estimate a hierarchical model using **MCMChregress**(fixed, random, group, data, r, R). Note the h for hierarchical buried in that function name. This is a slightly different syntax than lme4 uses (as we reviewed in Sect. 9.3.5), as it separates the fixed and random effect specifications. The key arguments we use here are:

**fixed** : formula for fixed effects at the higher level that are the same for all respondents

**random** : formula for random effects that are estimated for each respondent

**group** : name of the column with identifiers that group observations for the random effects

**data** : the data frame with observations

**r, R** : pooling arguments. We’ll just set them for now; see below for detail

- 고정 효과의 경우 평가할 기본 모델을 지정합니다: **rating ∼ speed +height + const + theme**

- 무작위 효과의 경우 마케팅에서 가장 일반적인 모델은 모든 응답자에 대한 모델의 모든 매개변수를 추정하므로 **random = ∼ speed + height + const + theme** 를 지정

- 개인별로 추정하기 때문에 그룹은 응답자 식별자인 "resp.id"임

**For fixed effects** we specify the primary model to estimate: **rating ∼ speed +height + const + theme**. **For random effects**, the most common models in marketing estimate all parameters of the model for every respondent, so we specify **random = ∼ speed + height + const + theme**. Because we are estimating by individual, group is the respondent identifier, "**resp.id**".

Estimation of this model may take several minutes to run. Here is the final code:

| | > # hierarchical lm with MCMC  > # WARNING: SLOW! Takes approx. 2 minutes on 2018 Macbook Pro  > #  > set.seed(97439)  > ride.mc2 <- MCMChregress(fixed = rating ~ speed + height + const + theme,  + random = ~ speed + height + const + theme,  + group="resp.id", data=conjoint.df, r=8, R=diag(8))  Running the Gibbs sampler. It may be long, keep cool :)  \*\*\*\*\*\*\*\*\*\*:10.0%  \*\*\*\*\*\*\*\*\*\*:20.0%  \*\*\*\*\*\*\*\*\*\*:30.0%  \*\*\*\*\*\*\*\*\*\*:40.0%  \*\*\*\*\*\*\*\*\*\*:50.0%  \*\*\*\*\*\*\*\*\*\*:60.0%  \*\*\*\*\*\*\*\*\*\*:70.0%  \*\*\*\*\*\*\*\*\*\*:80.0%  \*\*\*\*\*\*\*\*\*\*:90.0%  \*\*\*\*\*\*\*\*\*\*:100.0% | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

모델은 개인의 계수와 해당 개인을 가장 잘 설명하는 고차 분포 모두에 대해 수천 개의 추정치를 만들기 때문에 느림

While the model runs, let’s examine the **two arguments r and R**. A hierarchical model assumes that each respondent has a set of preferences (coefficients) drawn from a larger distribution that defines the range of possible preferences. The model is slow because it makes thousands of estimates of both the individuals’ coefficients and the higher-order distributions that best describe those individuals.

Of course there are only a few observations for each respondent, and a model for a single person can not be estimated very well with such limited data. To improve estimation, the MCMC model pools information across respondents, allowing estimates to be more or less similar to one another based on the data. If several respondents dislike a feature, it’s more likely (but not certain) that another randomly selected respondent will also dislike it; this expected similarity is used to improve estimates given sparse data. (각 응답자에 대한 관찰은 소수에 불과하며 한 사람에 대한 모델은 이러한 제한된 데이터로 잘 추정될 수 없음. 추정을 개선하기 위해 MCMC 모델은 응답자 간에 정보를 풀링하여 데이터를 기반으로 추정이 서로 다소 유사할 수 있도록 함. 여러 응답자가 기능을 싫어하면 무작위로 선택한 다른 응답자도 싫어할 가능성이 더 높음 (그러나 확실하지 않음). 이 예상 유사성은 희소 데이터가 주어진 추정치를 개선하는 데 사용됨.)

응답자 전체의 풀링 정도는 마지막 두 인수 r 및 R에 의해 조정됨. 대부분의 분석에서 r을 모델의 매개변수 수와 동일하게 설정하고 R을 대각 행렬과 동일하게 설정할 수 있음. 모델의 매개변수 수와 알고리즘이 데이터에서 최적의 풀링 수준을 결정함. 이것은 간단한 함수 diag(K)로 수행할 수 있음. 여기서 K는 r과 동일한 숫자임. 그러나 계층적 베이지안 모델을 정기적으로 실행할 계획이라면 풀링에 대해 더 배우고 싶을 것임. 섹션9.8 참조.

**That degree of pooling across respondents is adjusted by the final two arguments r and R**. For most analyses, you can set r equal to the number of parameters in your model and R equal to a diagonal matrix with values along the diagonal equal to the number of parameters in your model, and the algorithm will determine the optimal level of pooling from the data. This can be done with the simple function **diag(K), where K is the same number as r.** However, if you plan to run hierarchical Bayesian models regularly, you will wish to learn more about pooling; check the references in Sect. 9.8.

By now, MCMChregress() from above should have finished, and we can review its result:

| | > str(ride.mc2)  List of 2  $ mcmc : 'mcmc' num [1:1000, 1:1674] 3.04 2.87 2.9 3.06 2.98 ...  ..- attr(\*, "dimnames")=List of 2  .. ..$ : NULL  .. ..$ : chr [1:1674] "beta.(Intercept)" "beta.speed50" "beta.speed60" "beta.speed70" ...  ..- attr(\*, "mcpar")= num [1:3] 1001 10991 10  $ Y.pred: num [1:3200] 4.94 2.69 5.73 6.24 4.67 ... | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

MCMChregress의 출력은 두 개의 항목이 있는 목록임.

- 이 목록의 첫 번째 항목은 매개변수의 사후 분포에서 가져온 결과를 포함하는 mcmc 객체임. 주목할만한 점은 ride.mc2$mcmc에 1674개의 열이 포함되어 있다는 것임. 왜 그렇게 많이? 이 모델은 200명의 응답자 모두에 대해 롤러코스터의 각 속성에 대한 선호도인 8개의 계수 집합을 추정합니다. 이는 1600개의 매개변수에 전체 인구 분포를 설명하는 몇 가지를 더한 것임.

각 매개변수에 대해 모든 응답자의 사후 분포에서 1000개의 추정치를 도출했습니다(섹션 6.6.2 참조).

**The output of MCMChregress is a list with two items**. The first item in this list is an **mcmc object containing the draws from the posterior distribution of the parameters**. A notable thing is that ride.mc2$mcmc contains 1674 columns. Why so many? The model estimates a set of 8 coefficients—the preferences for each attribute of our roller coasters—for every one of the 200 respondents. That’s 1600 parameters plus a few more that describe the overall population distribution. For each of those parameters, it drew 1000 estimates from the posterior distribution for every respondent (see Sect. 6.6.2).

Let’s look at the first 8 columns, estimated coefficients for the overall, population-level preferences:

| | > # Try it!: dimnames(b2$mcmc)  >  > # overall estimates  > summary(ride.mc2$mcmc[ ,1:8])  Iterations = 1001:10991  Thinning interval = 10  Number of chains = 1  Sample size per chain = 1000  1. Empirical mean and standard deviation for each variable,  plus standard error of the mean:  Mean SD Naive SE Time-series SE  beta.(Intercept) 3.0739 0.1694 0.005356 0.005457  beta.speed50 0.8168 0.1398 0.004422 0.004422  beta.speed60 1.5691 0.1618 0.005117 0.005569  beta.speed70 4.4849 0.1862 0.005889 0.005889  beta.height300 2.9474 0.1235 0.003904 0.003681  beta.height400 1.4578 0.1796 0.005680 0.005680  beta.constWood -0.1128 0.1952 0.006172 0.005615  beta.themeEagle -0.7542 0.1857 0.005871 0.005871  2. Quantiles for each variable:  2.5% 25% 50% 75% 97.5%  beta.(Intercept) 2.7389 2.9594 3.0764 3.18818 3.4099  beta.speed50 0.5421 0.7251 0.8114 0.91274 1.0801  beta.speed60 1.2604 1.4636 1.5725 1.68365 1.8804  beta.speed70 4.1213 4.3599 4.4834 4.60792 4.8599  beta.height300 2.7114 2.8642 2.9501 3.03263 3.1779  beta.height400 1.0898 1.3429 1.4589 1.58500 1.8017  beta.constWood -0.5219 -0.2464 -0.1105 0.01628 0.2698  beta.themeEagle -1.0999 -0.8745 -0.7571 -0.63284 -0.3609 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

**이러한 추정치는 위의 모델 ride.mc1의 비계층적 MCMCregress 결과와 거의 동일함**.

speed70은 여전히 선호되며 평가 척도에서 4.5점의 가치가 있으며 목재 건축에 대한 선호도는 0에 가까움

These estimates are nearly identical to the result of non-hierarchical MCMCregress in model ride.mc1 above. speed70 is still preferred and worth 4.5 points on our rating scale, preference for wood construction is near zero, and so forth.

Let’s look at an example respondent; we pull and summarize the posterior draws for the parameters that are associated with respondent 196. We do this by finding columns that are named with “196” (the resp.id that we want). We accomplish that by indexing the columns with the results of the **grepl()** function that identifies elements of a character vector (in this case, column names) containing a particular string:

| | > # estimates for one respondent (respondent 196)  > summary(ride.mc2$mcmc[ , grepl(".196", colnames(ride.mc2$mcmc), fixed=TRUE)])  Iterations = 1001:10991  Thinning interval = 10  Number of chains = 1  Sample size per chain = 1000  1. Empirical mean and standard deviation for each variable,  plus standard error of the mean:  Mean SD Naive SE Time-series SE  b.(Intercept).196 -1.03806 0.6780 0.02144 0.02144  b.speed50.196 0.44049 0.5434 0.01718 0.01718  b.speed60.196 0.10442 0.6335 0.02003 0.02003  b.speed70.196 0.03807 0.7167 0.02266 0.02357  b.height300.196 -0.35414 0.5441 0.01721 0.01797  b.height400.196 -0.55132 0.7357 0.02327 0.02327  b.constWood.196 2.57915 0.8370 0.02647 0.02647  b.themeEagle.196 -1.41955 0.8220 0.02599 0.02599  2. Quantiles for each variable:  2.5% 25% 50% 75% 97.5%  b.(Intercept).196 -2.358 -1.5051 -1.04955 -0.560096 0.2484  b.speed50.196 -0.617 0.0772 0.43020 0.785476 1.5038  b.speed60.196 -1.116 -0.3015 0.09268 0.520582 1.3024  b.speed70.196 -1.330 -0.4398 0.02173 0.535568 1.4501  b.height300.196 -1.397 -0.7228 -0.35641 0.009468 0.7580  b.height400.196 -1.994 -1.0493 -0.57508 -0.044254 0.8956  b.constWood.196 1.072 1.9835 2.53838 3.141001 4.2232  b.themeEagle.196 -3.033 -1.9355 -1.44426 -0.881212 0.1661 | | --- | |  | |  | |
| --- | --- | --- | --- |

응답자 196은 wood coaster를 선호함. 평가는 steel(기본 수준)에 대한 평가보다 10점 척도에서 2.5점 더 높음. 반면에 그녀는 독수리 테마 디자인을 싫어하여 드래곤 테마보다 평균 -1.4점 낮게 평가함.

Respondent 196 strongly prefers wood coasters; her ratings for them are 2.5 points higher on our 10 point scale than those for steel construction (the default level). On the other hand, she dislikes the eagle-themed design, rating it −1.4 points lower on average than the dragon theme. These preferences are rather different than the population averages above.

**(정교한 개별 마케팅 활용 가능)**

이 정보를 어떻게 사용할 수 있습니까? 응답자 196에게 이상적인 롤러코스터는 최고 속도가 50mph이고 높이가 200피트(기본 수준은 표시되지 않음)인 용을 테마로 한 나무 코스터임. 롤러 코스터의 경우 개별 맞춤화는 비실용적이지만, 그럴듯한 마케팅 사용은 코스터의 혼합을 결정하기 위해 응답자의 선호도를 세분화하는 것임(11장 참조). 예를 들어, 어떤 새로운 코스터가 공원이 이미 가지고 있는 코스터보다 선호도를 극대화할 것인지 물을 수 있음. 즉, 제품 라인 확장을 조사할 수 있음..

**How could we use this information?** The ideal roller coaster for respondent 196, according to her responses, would be a dragon-themed wood coaster with a top speed of 50mph and a height of 200 feet (the default level not shown). Although individual customization is impractical for roller coasters, a plausible marketing use would be to segment respondents’ preferences to determine a mix of coasters (see Chap. 11). For instance, we might ask which new coaster would maximize preference over and above the coasters the park already has; in other words, we could investigate a product line extension. More immediately, if we have respondents’ contact information, we could tailor marketing communications to this and similar respondents and tell them about wooden coasters at the park.

MCMC 결과물은 또한 추정치의 신뢰도를 알려줌. 평균 추정치의 표준 오차를 사용할 수도 있지만 대신 결과물의 분위수 섹션 값을 사용하는 것이 좋음.

The MCMC output also informs our **confidence of estimates**. One could use the standard error of the mean estimate, but we recommend **instead to use the values from the Quantiles section of the output**. Let’s look at the population estimates again, but focus on the quantiles::

| | > # Try it!: dimnames(b2$mcmc)  >  > # overall estimates  > summary(ride.mc2$mcmc[ ,1:8])  Iterations = 1001:10991  Thinning interval = 10  Number of chains = 1  Sample size per chain = 1000  1. Empirical mean and standard deviation for each variable,  plus standard error of the mean:  Mean SD Naive SE Time-series SE  beta.(Intercept) 3.0739 0.1694 0.005356 0.005457  beta.speed50 0.8168 0.1398 0.004422 0.004422  beta.speed60 1.5691 0.1618 0.005117 0.005569  beta.speed70 4.4849 0.1862 0.005889 0.005889  beta.height300 2.9474 0.1235 0.003904 0.003681  beta.height400 1.4578 0.1796 0.005680 0.005680  beta.constWood -0.1128 0.1952 0.006172 0.005615  beta.themeEagle -0.7542 0.1857 0.005871 0.005871  2. Quantiles for each variable:  2.5% 25% 50% 75% 97.5%  beta.(Intercept) 2.7389 2.9594 3.0764 3.18818 3.4099  beta.speed50 0.5421 0.7251 0.8114 0.91274 1.0801  beta.speed60 1.2604 1.4636 1.5725 1.68365 1.8804  beta.speed70 4.1213 4.3599 4.4834 4.60792 4.8599  beta.height300 2.7114 2.8642 2.9501 3.03263 3.1779  beta.height400 1.0898 1.3429 1.4589 1.58500 1.8017  beta.constWood -0.5219 -0.2464 -0.1105 0.01628 0.2698  beta.themeEagle -1.0999 -0.8745 -0.7571 -0.63284 -0.3609 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

이것은 speed70에 대한 고정 효과 추정치가 사후 분포의 95%에서 4.12–4.86 사이의 값을 가졌음을 알려줌. 따라서 이 값을 사용하여 보고하는 매개변수의 신뢰할 수 있는 구간을 표현할 수 있습니다. 베이지안 통계의 장점은 귀무 가설에 대한 논의에 의존하지 않고 추정치의 신뢰도를 직접적으로 진술할 수 있다는 것임.

This tells us that the fixed effect estimate for speed70 had a value between 4.12–4.86 in 95% of the draws from the posterior distribution. Thus, we can use these values to express the credible interval for the parameters we report. An **advantage of Bayesian statistics** is that confidence in estimates can be stated directly, without resorting to discussion of null hypotheses.

***9.4.3 Inspecting Distribution of Preference\****

우리는 응답자들이 나무와 강철로 된 coaster에 그저 무관심한 것인지, 아니면 상당한 차이가 있는지 궁금함. 추정된 모델에서 이것을 조사하려면 약간의 작업을 수행해야 합니다. 먼저, 목재 건축에 대한 선호도에 대한 개별 수준 추정치인 b.constWood로 표시된 모든 계수를 추출함. 이 계수에는 데이터 세트의 각 고객에 대해 하나씩 200개의 열이 있음.

이러한 값은 각각 전체 모집단과 관련된 개인의 차이를 나타내므로 기본 모집단 추정치인 beta.constWood에 값을 추가함. MCMC 추출에서 1000개의 추정 세트가 있기 때문에 사후 분포에서 1000개의 추출 각각에 대한 총계(개인 + 모집단 평균)를 계산하고 해당 합계를 요약합니다. (먼저 요약한 다음 추가하지 마십시오.) 이 프로세스가 복잡하게 들릴 수 있지만 비밀스럽긴 하지만 단일 명령으로 수행됨

We wondered above **whether respondents were just indifferent to wooden versus steel coasters, or had significant differences**. To investigate this in the estimated model, we need to do a bit of work. First, we extract out all the coefficients labeled b.constWood, which are the individual-level estimates for preference for wood construction. There are 200 columns for these coefficients, one for each customer in our data set.

Those values each represent a difference for the individual relative to the overall population, so we add the values to the baseline population estimate, beta.constWood. Because we have 1000 sets of estimates from the MCMC draws, we compute the total (individual plus population mean) for each of the 1000 draws from the posterior distribution, and summarize those totals. (Do not summarize first and then add.) Although this process may sound complex, it is accomplished in a single, albeit cryptic, command:

| | > # estimates for wood construction  > ride.constWood <- summary(ride.mc2$mcmc[ , grepl("b.constWood",  + colnames(ride.mc2$mcmc))]  + + ride.mc2$mcmc[ , "beta.constWood"]) | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

이 코드를 분해하면 이름에 "b.constWood"가 포함된 mcmc draw의 열을 찾습니다. 선호도의 개인차입니다. 각 응답자에 대한 전체 선호도를 얻기 위해 모집단 값 beta.constWood를 추가합니다. 그런 다음 결과를 요약합니다. (R 콘솔에서 이 코드의 일부를 시도하여 작동 방식을 확인할 수 있습니다.)

결과는 ride.constWood가 강철 코스터보다 나무를 선호하는 개별 수준에 대한 사후 분포의 추정치를 포함한다는 것입니다. 우리는 wood coasters 에 대한 개인의 선호도 분포를 확인하기 위해 이것을 플롯함

Deconstructing this code, it finds the columns in mcmc draws with “b.constWood” in their names; those are the individual differences in preference. It adds the population value, beta.constWood, to obtain the total preference for each respondent. Then it summarizes the result. (You might try parts of this code in the R console to see how this works.)

The result is that **ride.constWood** contains estimates from the posterior distribution for individual-level preference of wood over steel coasters. We plot these to see the distribution of individuals’ preferences for wood coasters:

| > hist(ride.constWood$statistics[,1],  + main="Preference for Wood vs. Steel",  + xlab="Rating points", ylab="Count of Respondents", xlim=c(-4,4))  > |
| --- |

결과 차트는 그림 9.3에 나와 있음. 목재 대 강철 건축을 선호하는 개인 선호도의 범위가 넓음 일부 응답자는 목재를 부정적으로 선호하여 강철을 선호하는 반면 다른 응답자는 목재를 선호함. 등급의 차이가 최대 4점에 해당하는 규모가 일부에게는 매우 강력함

We compare that to the distribution of preference for 60mph speed (versus baseline 40mph):

| | > # 60 mph  > ride.speed60 <- summary(ride.mc2$mcmc[,grepl("b.speed60",  + colnames(ride.mc2$mcmc))]  + + ride.mc2$mcmc[,"beta.speed60"])  >  > hist(ride.speed60$statistics[,1],  + main="Preference for 60mph vs. 40mph",  + xlab="Rating points", ylab="Count of Respondents", xlim=c(-4,4)) | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

.이에 비해 두 번째 차트에서는 40mph보다 60mph 코스터에 대한 선호도가 덜 다양합니다. 모든 응답자는 더 빠른 속도를 선호합니다.

The resulting charts are shown in Fig. 9.3. In the first, we see a wide range across individuals in preference of wood versus steel construction; some respondents have negative preference for wood, and thus prefer steel, while others prefer wood. The magnitude is very strong for some, corresponding to a difference in rating of up to 4 points. By comparison, in the second chart, preference for 60mph coasters over 40mph is less diverse; **all respondents prefer the faster speed**.

This degree of variation among respondents is known as **heterogeneity**, and in addition to estimating the parameters (coefficients) for the population (beta.<predictor name> as we saw above), MCMC hregress() also estimates their variance and covariance across the population of respondents. The results are named VCV. <predictor name>. <predictor name> in the output, where “VCV” abbreviates variance covariance. When the two predictor names are the same, this gives the variance estimate for a single parameter; when they are different, it is the covariance of two parameters. (응답자 간의 이러한 변동 정도를 이질성(heterogeneity)이라고 하며, 모집단에 대한 매개변수(계수) (beta.<predictor name>)를 추정하는 것 외에도 MCMC hregress()는 모집단 전체에서 분산 및 공분산을 추정함. 출력에서의 결과의 이름은 VCV.<predictor name>.<predictor name>이고, 여기서 "VCV"는 분산 공분산(variance covariance)임. 두 예측 변수(predictor) 이름이 같으면 단일 매개변수에 대한 분산 추정값을 주고, 다를 때는 그것은 두 매개변수의 공분산임.)

For example, we can find the population mean and variance of the wood and 60mph parameters:

| | > summary(ride.mc2$mcmc[,c("beta.constWood", "VCV.constWood.constWood",  + "beta.speed60","VCV.speed60.speed60")])  Iterations = 1001:10991  Thinning interval = 10  Number of chains = 1  Sample size per chain = 1000  1. Empirical mean and standard deviation for each variable,  plus standard error of the mean:  Mean SD Naive SE Time-series SE  beta.constWood -0.1128 0.1952 0.006172 0.005615  VCV.constWood.constWood 2.3458 0.3749 0.011855 0.014056  beta.speed60 1.5691 0.1618 0.005117 0.005569  VCV.speed60.speed60 0.5782 0.1351 0.004273 0.004939  2. Quantiles for each variable:  2.5% 25% 50% 75% 97.5%  beta.constWood -0.5219 -0.2464 -0.1105 0.01628 0.2698  VCV.constWood.constWood 1.7028 2.0760 2.3155 2.57251 3.1834  beta.speed60 1.2604 1.4636 1.5725 1.68365 1.8804  VCV.speed60.speed60 0.3559 0.4797 0.5664 0.66069 0.8711 | | --- | |  | |
| --- | --- | --- |

constWood에 대한 추정된 분산은 2.34로 상당히 커서 나무 롤러코스터를 선호하는 응답자 사이에 큰 이질성이 있음을 보여줍니다. 반면에 speed60에 대한 추정치의 분산은 0.58로 훨씬 작습니다. 이것은 그림 9.3의 히스토그램에서 본 분포의 차이를 반영합니다.

The estimated variance for constWood is quite large at 2.34, demonstrating that there is large heterogeneity between respondents in preference for wooden roller coasters. On the other hand, the variance of the estimates for speed60 is much smaller at 0.58. This reflects the difference in distributions that we saw in the histograms in Fig. 9.3.

개별 기능에 대한 관심이 아니라 하나 이상의 완전히 지정된 롤러 코스터 디자인에 대한 응답자의 관심을 예측할 수 있음. 이러한 평가는 제품 관심을 예측하기 위한 공동 분석에서 일반적이며 종종 시장 시뮬레이션으로 알려져 있음. 그러나 lm()과 같이 MCMC 모델에 대한 적절한 predict() 함수는 아직 없음. 디자인에 대한 전반적인 선호도를 추정하기 위해 두 가지 선택이 있음. 한 가지 옵션은 설계와 일치하는 MCMC 추첨 열을 추가하여 순 관심 수준을 계산한 다음(기본 모집단 추정치 포함) 각 응답자에 대한 관심 수준을 요약하는 것임. 또 다른 옵션은 선택 간의 선호도를 비교하는 시장 시뮬레이션 루틴을 사용하는 것임. 예는 Chapman et al. [33]에서 볼 수 있음. 13장에서 preference share estimation에 대해 더 논의함

You might wish to predict respondents’ interest in one or more fully-specified roller coaster designs, as opposed to interest in individual features. Such assessment is typical in conjoint analysis to predict product interest and is often known as market simulation. However, there is not yet an appropriate predict() function for MCMC models as there is for lm(). To obtain estimates of overall preference for a design, there are two choices. One option is to calculate the net level of interest by adding the columns of the MCMC draws that match your design (plus the baseline population estimates), and then summarize the level of interest for each respondent. Another option is to use a market simulation routine that compares preference between choices, such as the relative preference for your design versus some other design; an example is available in Chapman et al. [33]. We discuss preference share estimation further in Chap. 13.

이 모델과 관련하여 언급해야 하는 또 다른 사항은 데이터 시뮬레이션과 그림 9.3 및 모델의 가정에서 볼 수 있듯이 개인의 추정치(임의 계수)가 다변량 정규 분포를 따른다고 가정한다는 것임. 이는 모델이 대부분의 사람들의 선호도가 분포의 중간에 있다고 가정하고 선호도가 강한 응답자는 더 적다고 가정함. 다양하고 강력한 선호도를 가진 별도의 그룹이 있다고 의심되는 경우 이 장의 범위를 벗어나는 혼합 또는 잠재 클래스 모델을 고려할 수 있음. (see [168], Chap. 5)

One other thing we should mention with regard to this model—as is illustrated in our data simulation and Fig. 9.3 as well as in the model’s assumptions—is that individuals’ estimates (random coefficients) are assumed to follow a multivariate normal distribution. This means that the model assumes most people’s preferences are in the middle of the distribution, with fewer respondents who have strong preferences. If you have reason to suspect that there are separate groups with divergent and strong preferences, you might consider a mixture or latent class model, which is outside the scope of this chapter (see [168], Chap. 5).

- 계층적 베이지안 모델은 개별 고객을 이해하는 데 그 가치가 있음

- 계층적 모델링은 우리가 개별 수준에서 모델 추정치를 얻고 고객 전반의 다양성을 이해할 수 있게 해주기 때문에 마케팅에서 널리 퍼졌음.

- 이러한 결합 분석 모델에 대해서는 선택 기반 결합 분석의 형태로 13장에서 더 자세히 설명하겠음

- 이러한 모델은 고객 관계 관리(CRM) 응용 프로그램에서도 흔히 볼 수 있음. 여기서 목표는 개별 고객에 대한 어떤 종류의 응답이나 결과를 추정하는 것임. 선형 모델에 관심이 있을 때마다 베이지안 접근 방식을 고려하는 것이 좋음.

We hope this introduction to hierarchical Bayesian models has demonstrated their value in understanding individual customers. **Hierarchical modeling has become widespread in marketing because it allows us both to obtain model estimates at an individual level and to understand the diversity across customers**. We’ll have more to say about such models for conjoint analysis, in the form of choice-based conjoint analysis, in Chap. 13. **These models are also common in customer-relationship management (CRM) applications**, where the goal is to estimate a likely response or outcome of some sort for individual customers. We suggest to consider a Bayesian approach anytime that you are interested to fit a linear model.

**9.5 A Quick Comparison of the Effects\***

이 섹션에서 **고전적 방법(섹션 9.3)과 베이지안 방법(섹션 9.4)을 사용하여 동일한 데이터 세트를 모델링했음**. 고정 효과의 추정치는 두 모델에서 거의 동일하다는 것을 알았음(9.4.1절). 무작위 개인 수준 효과는 어떻습니까? 얼마나 비슷합니까?

This is an optional section for those who completed both of the previous sections. In those sections we modeled the same data set using classical methods (Sect. 9.3) and Bayesian methods (Sect. 9.4). We saw that **the estimates of the fixed effects are nearly identical in the two models** (Sect. 9.4.1). **What about the random, individual-level effects? How similar are they?**

첫째, 고정 효과는 총 3200개의 관찰이 있더라도 두 방법 간에 정확히 동일하지 않다고 생각할 수 있음.

둘째, 응답자당 관찰이 16개뿐인 개인 수준 효과가 훨씬 더 많은 분산을 가질 것으로 예상해야 합니다(분산은 관찰 수의 제곱근에 반비례하기 때문에). 단지 16개의 관찰이 주어지면 응답자당 8개의 무작위 효과를 추정한다는 점을 고려할 때 추정치에서 많은 불확실성을 예상해야 함.

셋째, 우리는 어느 모델도 참으로 간주될 수 없으며 단지 편향되지 않은 추정치일 것으로 예상된다는 점을 이해해야 함.

Before examining those effects, let’s try to apply a bit of intuition to the problem.

First, we might consider that the fixed effects, even with 3200 total observations are not exactly identical between the two methods.

Second, we should expect that the individual-level effects, with only 16 observations per respondent would have much more variance (because variance is inversely proportional to the square root of the number of observations). When we consider that we are estimating 8 random effects per respondent given only 16 observations, we should expect a lot of uncertainty in the estimates.

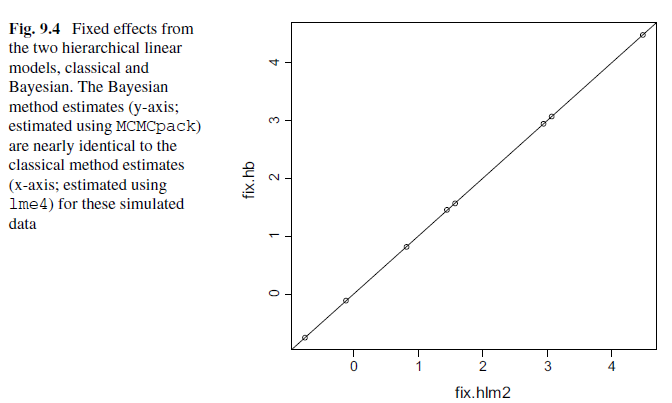
Third, we should understand that neither model can be regarded as true, but only expected to be (one hopes) an unbiased estimate.

여기에서 모델을 비교하려면 위에서 했던 것처럼 ride.hlm2 및 ride.mc2 모델을 모두 적합시켜야 함(각각 섹션 9.3.6 및 9.4.2).

평균 고정 효과 추정치가 매우 유사하다는 것을 확인했습니다. 우리는 각각의 8개 매개변수를 다른 모델의 매개변수와 대조하여 시각적으로 확인할 수 있음

To compare the models here, you need to fit both the ride.hlm2 and ride.mc2 models as we did above (Sects. 9.3.6 and 9.4.2, respectively).

We’ve seen that the mean fixed effect estimates are quite similar. We can check that visually by plotting the eight parameters of each against those from the other model.



First we get the fixed effects from each, then we plot them against one another and add a 45◦ line to see how closely they align.

| > #### Reflections on Model Comparison  >  > # now that we have models from 2 models, we might compare the fixed effects  > fix.hlm <- fixef(ride.hlm2)  > fix.hb <- colMeans(ride.mc2$mcmc[ , 1:8])  >  > plot(fix.hlm, fix.hb)  > abline(0,1)  > |
| --- |

그림 9.4는 고정 효과가 두 모델에서 거의 동일함을 보여줌. 두 모델은 계층적 선형 모델이지만 다른 방법으로 추정되지만 베이지안 모델의 경우 "HB"와 구별하기 위해 lme4에서 추정한 모델을 참조하기 위해 약어 "HLM"을 사용함.

Figure 9.4 shows that the **fixed effects** are nearly identical in the two models. Note that we use the abbreviation **“HLM”** to refer to the model estimated by lme4 in order to distinguish it from **“HB”** for the Bayesian model, although both models are hierarchical linear models yet estimated with different methods.

무작위 효과는 응답자 내에서 비교되어야 함. 위에서 고려한 ID 196이라는 한 명의 응답자에 대해서만 이 작업을 수행. 먼저 각 랜덤 효과의 평균 추정치를 살펴보겠음. mme4 모델(9.3.6절)에 ranef()를 사용하고 MCMC 모델(9.4.2절)의 결과에서 추정된 평균 효과를 취하기 위해 colMeans()를 사용하여 추출함.

The **random effects** have to be compared within respondent. We’ll do this for just one respondent, ID 196 who we considered above. First, let’s just consider the mean estimates of each random effect. We extract those using **ranef()** for the lme4 model (Sect. 9.3.6) and **colMeans()** to take the mean effect estimated in the draws of the MCMC model (Sect. 9.4.2):

| | > # or compare random effects (in this case, for one respondent)  > # in general, would want to compare full coefficients (fixed + random)  > #  > # but in this case, the fixed are nearly identical between the two,  > # so we'll omit those for convenience  > #  > # LME random effects for ID #196  > ranef(ride.hlm2)$resp.id[196, ]  (Intercept) speed50 speed60 speed70 height300 height400 constWood themeEagle  196 -0.5994599 -0.8942114 -0.5383142 -0.2119763 0.1169787 1.21449 0.005058678 -0.1268496  >  > # MCMC random effects for ID #196  > colMeans(ride.mc2$mcmc[ , grepl(".196", colnames(ride.mc2$mcmc),  + fixed=TRUE)])  b.(Intercept).196 b.speed50.196 b.speed60.196 b.speed70.196 b.height300.196  -1.03806213 0.44049447 0.10441996 0.03807113 -0.35414215  b.height400.196 b.constWood.196 b.themeEagle.196  -0.55131679 2.57914806 -1.41954714 | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

동일한 고정 효과에 비해 독수리 테마에 대한 강한 부정적인 효과와 나무 롤러 코스트에 대한 강한 긍정적인 것과 같이 응답자 196에 대한 두 가지 추정치에는 전반적인 유사점이 있습니다. 그러나 일부 평균 추정치에는 약간의 차이가 있습니다. MCMC 프로세스는 베이지안 방법이 점 추정치(위에 보고된 평균 효과 추정치)뿐만 아니라 불확실성을 반영하는 사후 분포도 추정한다는 점을 상기해야 합니다.

There are some overall similarities in the two sets of estimates for respondent 196, such as the strong negative effect for the eagle theme, relative to the same fixed effect, and strong positive for a wooden roller coasts. However, there are small to modest differences in some of the mean estimates. The MCMC process should prompt you to recall that Bayesian methods estimate not only a point estimate (the mean effect estimate reported above), but also a posterior distribution that reflects uncertainty.

다양한 방법으로 추정치를 비교할 수 있습니다. 이 경우 시각적으로 비교합니다. 두 추정치 세트에 대한 분포 곡선을 겹쳐서 이를 수행함.

(1)HB 추정치의 경우 각 매개변수에 대해 1000 MCMC 드로우가 있으므로 해당 드로우의 밀도() 추정치를 플로팅합니다.

(2) HLM 추정에 대해 다음과 같은 방식으로 유사한 밀도 추정을 구성합니다. ranef()에서 평균 효과를 얻고 ranef() 무작위 효과 추정의 "postVar"(분산) 속성에서 추정의 표준 편차를 얻습니다. 한 명의 응답자가 해당 매개변수를 사용하여 해당 분포에서 임의의 점을 추출합니다.

이 프로세스를 한 번 수행하면(MCMC 그리기의 밀도를 플로팅한 다음 HLM 추정값의 평균 및 표준 편차에 대한 분포도를 추가하여) 한 가지 속도 또는 설계에 대한 선호도와 같은 한 가지 매개변수 집합을 비교할 수 있습니다. 여러 매개변수를 비교하기 위해 이를 반복합니다. 다음과 같이 처음 4개의 비절편 매개변수인 매개변수 2–5에 대해 이를 수행합니다.

One might compare estimates in various ways; in this case, we compare them visually. We’ll do this by overlaying distribution curves for the two sets of estimates. In the case of the **HB estimates**, we have 1000 MCMC draws for each parameter, so we plot the **density()** estimate of those draws. For the **HLM estimates**, we construct a similar density estimate in the following way: we obtain the mean effect from ranef() and the standard deviation of the estimation from the “postVar” (variance) attribute of the ranef() random effect estimates for one respondent, and use those parameters to draw random points from that distribution.

Doing this process one time—**plotting the density of the MCMC draws and then adding a distribution plot for the mean and standard deviation of the HLM estimate**— would give us a comparison of one set of parameters such as the preference for one speed or design. We iterate that to compare multiple parameters. We do that for parameters 2–5, the first four non-intercept parameters, as follows:

| > # compare them graphically:  > # .. plot the distribution of the MCMC draws of the random effects for ID 196  > # .. and then add distribution for the LME random effects for ID 196  > # .. doing this for only the first 4 of the 7 non-intercept parameters  >  >  > par(mfrow=c(2,2)) # make a 2x2 plot surface  > plot.xlim <- c(-3, 3) # define limits for the x-axis  >  > for (i in 2:5) { # first four parameters only, for convenience  + # plot the MCMC density for random effect i  + mcmc.col <- which(grepl(".196", colnames(ride.mc2$mcmc), fixed=TRUE))[i]  + plot(density(ride.mc2$mcmc[ , mcmc.col]), xlab="",  + ylim=c(0, 1.4), xlim=plot.xlim,  + main=paste("HB & lmer density:",  + colnames(ride.mc2$mcmc)[mcmc.col] ))  + # add the HLM density for random effect i  + hlm2.est <- ranef(ride.hlm2)$resp.id[196, i] # mean estimate  + hlm2.sd <- sqrt(attr(ranef(ride.hlm2, condVar=TRUE)$resp.id,  + "postVar")[ , , 196][i, i])  + seq.pts <- seq(from=plot.xlim[1], to=plot.xlim[2], length.out=1000) # range  + # .. find density at x-axis points using dnorm() and add that to the plot  + points(seq.pts, dnorm(seq.pts, mean=hlm2.est, sd=hlm2.sd),  + col="red", pch=20, cex=0.4)  + legend("topright", legend=c("red = lmer", "black = HB"),  + text.col=c("red", "black"))  + }  >      > # compare them graphically:  > # .. plot the distribution of the MCMC draws of the random effects for ID 196  > # .. and then add distribution for the LME random effects for ID 196  > # .. doing this for only the first 4 of the 7 non-intercept parameters  >  >  > par(mfrow=c(2,2)) # make a 2x2 plot surface  > plot.xlim <- c(-3, 3) # define limits for the x-axis  >  > for (i in 2:5) { # first four parameters only, for convenience  + # plot the MCMC density for random effect i  + mcmc.col <- which(grepl(".196", colnames(ride.mc2$mcmc), fixed=TRUE))[i]  + plot(density(ride.mc2$mcmc[ , mcmc.col]), xlab="",  + ylim=c(0, 2.0), xlim=plot.xlim,  + main=paste("HB & lmer density:",  + colnames(ride.mc2$mcmc)[mcmc.col] ))  + # add the HLM density for random effect i  + hlm2.est <- ranef(ride.hlm2)$resp.id[196, i] # mean estimate  + hlm2.sd <- sqrt(attr(ranef(ride.hlm2, condVar=TRUE)$resp.id,  + "postVar")[ , , 196][i, i])  + seq.pts <- seq(from=plot.xlim[1], to=plot.xlim[2], length.out=1000) # range  + # .. find density at x-axis points using dnorm() and add that to the plot  + points(seq.pts, dnorm(seq.pts, mean=hlm2.est, sd=hlm2.sd),  + col="red", pch=20, cex=0.4)  + legend("topright", legend=c("red = lmer", "black = HB"),  + text.col=c("red", "black"))  + }  > |
| --- |

그림 9.5의 결과 차트는 두 가지 방법의 밀도 추정치가 크게 겹친다는 것을 보여줍니다. 결과는 다르지만 크게 다르지 않기 때문에 위의 직관과도 일치하고 분포가 일반적으로 범위와 중심 경향이 비슷하고 MCMC 추정 분산이 약간 높기 때문에 두 방법이 크게 불일치한다고 의심할 이유가 없습니다. 물론 이것은 단일 응답자에 대한 4개의 매개변수만을 비교한 것입니다.

200명의 응답자 모두를 그래픽으로든 통계적으로든 유사하게 비교할 수 있음 그렇게 한다면 우리는 무엇을 보게 될까요? 고정 효과가 거의 동일하다는 점을 감안할 때 무작위 효과의 모델 간의 편차는 0에 가까우며 0을 중심으로 대칭이 될 것으로 예상합니다.

200명의 모든 개인에 대한 효과는 8개의 매개변수에 대해 -0.015–0.020 범위이고 중앙값은 0.003입니다.

This code is lengthy but should not be difficult for you to deconstruct by this point. The two significant new elements here are that it uses **attr(..., "postVar")** to obtain the variance of the random effect estimate for the HLM model, and uses **dnorm()** to obtain a density estimate for 1000 points that match the HLM parameter distribution estimate, which it adds to the plot with points().

The resulting chart in Fig. 9.5 shows that **the density estimates from the two methods are largely overlapping.** It is also congruent with our intuition above, as the results are different but not enormously so, and there is no reason to suspect either method is highly discrepant because the distributions are generally similar in range and central tendency, with just slightly higher variance in the MCMC estimates. Of course this is a comparison of only four parameters for a single respondent.

We could compare similarly across all 200 respondents, either graphically or statistically, but will leave that as an exercise for the reader. If we did so, what would we expect to see? Given that the fixed effects are nearly identical, we would expect that deviations between the models in the random effects would be close to zero and symmetric around zero. If you want to try this on your own, we can give you a preview: the median difference between the models’ mean estimates of the random

effects, across all 200 individuals, for the 8 parameters, ranges from −0.015–0.020, with a median of 0.003.

**(모델이 유사하지만 동일하지 않은 경우 고전모델과 베이지안 중 어느 것이 나을까? 모델이 대답을 주는 것은 아니고 가정을 검토해야 함)**

**Given that the models are similar but not identical, you might wonder which is better, the classical or the Bayesian?** **The models themselves do not answer that; you would need to consider your assumptions**, the degree to which you believe each model is appropriate (see Sect. 6.6.1), and if possible, which works better for your situation in regards to other metrics such as external validity. As we have noted, the models tend to show increasingly similar estimates with larger samples, while the Bayesian methods may yield more intuitive or useful estimates with small numbers of observations.

(모델은 더 큰 샘플에서 점점 더 유사한 추정치를 표시하는 경향이 있는 반면 베이지안 방법은 적은 수의 관측값으로 더 직관적이거나 유용한 추정치를 산출할 수 있음.)

[추가] 교재외

| | > ###  > ### BEGIN: NOT in the book  > ###  > ### Comparing the respondents' deviations between models more systematically  > ###  >  > med.diff <- rep(NA, 8) # median difference between models on each parameter  > # WARNING: next part is sloow, takes 20-60 seconds  > for (i in 1:8) { # loop over the parameters  + hlm.diff <- rep(NA, 200) # difference between the models for each respondent (for parameter i)  +  + for (j in 1:200) { # loop over the respondents  + # find the MCMC mean for random effect of parameter i, respondent j  + mcmc.col <- which(grepl(paste0(".", j), colnames(ride.mc2$mcmc), fixed=TRUE))[i]  + mcmc.est <- mean(ride.mc2$mcmc[ , mcmc.col])  + # find the lme4 estimate of the random effect  + hlm2.est <- ranef(ride.hlm2)$resp.id[j, i]  + hlm.diff[j] <- mcmc.est - hlm2.est  + }  + print (i)  + print(summary(hlm.diff)) # diffs for parameter i  + med.diff[i] <- median(hlm.diff) # save the median of those diffs  + }  [1] 1  Min. 1st Qu. Median Mean 3rd Qu. Max.  -4.027041 -1.133778 -0.044538 0.000816 0.974429 4.509951  [1] 2  Min. 1st Qu. Median Mean 3rd Qu. Max.  -3.38332 -0.53573 -0.03780 -0.04349 0.52229 2.45265  [1] 3  Min. 1st Qu. Median Mean 3rd Qu. Max.  -1.957260 -0.501135 -0.008328 0.003733 0.600568 2.220459  [1] 4  Min. 1st Qu. Median Mean 3rd Qu. Max.  -2.031021 -0.320397 -0.009678 -0.011586 0.254430 2.088404  [1] 5  Min. 1st Qu. Median Mean 3rd Qu. Max.  -2.723272 -0.469974 0.037959 0.006953 0.458093 2.214734  [1] 6  Min. 1st Qu. Median Mean 3rd Qu. Max.  -2.676577 -0.636286 0.008515 0.054868 0.663719 4.379162  [1] 7  Min. 1st Qu. Median Mean 3rd Qu. Max.  -4.340965 -1.251864 0.021680 0.000588 1.235733 5.598068  [1] 8  Min. 1st Qu. Median Mean 3rd Qu. Max.  -4.24550 -1.29993 -0.02918 -0.01100 1.26415 3.75724  > summary(med.diff) # summary of the 8 median diffs  Min. 1st Qu. Median Mean 3rd Qu. Max.  -0.044538 -0.031336 -0.009003 -0.007671 0.011806 0.037959  >  > ### END: not in book | | --- | |  | | | > | | --- | | |
| --- | --- | --- | --- | --- |

| **[요약] Bayesian Methods for Hierarchical Linear Models**  • Hierarchical models in marketing are often estimated with Bayesian methods that are able to pool information and produce best estimates of both group and individual effects using potentially sparse data (Sect. 9.4.2).  • A Bayesian hierarchical linear model can be estimated using **MCMChregress()** in the MCMCpack package (Sect. 9.4.2).  • Model coefficients from a hierarchical model are inspected using summaries of the many estimates that are collected in an mcmc object (Sects. 9.4.2, 9.4.3). |
| --- |

| **[베이지안 회귀]**  [**https://mons1220.tistory.com/212**](https://mons1220.tistory.com/212)  회귀는 데이터로부터 모델을 추정하는 한 방법이다. 최소자승법이 잔차를 최소화 시키는 방법이라면, **베이지안 회귀는 가능도 최대화가 목적이다**.   이 글의 최종 목표는 베이지안 회귀의 원리를 이해하고, python package인 pymc3을 활용까지 다뤄보는 것이다.   이번 포스팅에서는 우선 원리를 이해하는 것을 목표로 한다.  **모델**   가장 기본적인 선형 모델은 다음과 같이 쓸 수 있다.   y=θ1x + θ2 + ϵ   이 때 θ1, θ2는 모델의 파라미터, x, y는 관측 값, ϵ은 오차를 나타낸다.   x, y라는 관측값으로 부터 θ11, θ2를 추정해 나가는 것이 목표다.  **가능도(**likelihood**)**  확률이 모델에서 관측값이 나올 확률이라면, 가능도는 관측값에서 모델이 지닌 가능도를 평가한다.  선형 모델에서 오차 ϵ은 평균 0, 표준편차 σ의 정규분포를 가정한다. 이를 식으로 표현하면 다음과 같다.  ϵ∼N(0,σ2)   그리고 이를 통해 y의 확률 분포를 나타내면 다음과 같다.  y∼N(θ1x+θ2, σ2)   가능도는 정규분포의 확률밀도 함수를 통해 구할 수 있다.   평균 μ, 표준편차 σ인 정규분포를 생각해보자.  x∼N(μ,σ2)  이 정규분포의 확률밀도는 다음과 같다.     이 때, 관측된 값 xi의 가능도는 xi가 확률밀도에서 갖는 값을 나타낸다. 그리고 가능도는 모든 관측된 값의 가능도의 곱으로 나타낸다. 우리의 관측값은 모델로부터 독립적으로 추출된 값이므로, 각 관측값에 대한 확률밀도의 곱이 가능도가 된다.  이를 이용해, 위의 y식의 가능도, 즉, 관측값 x, y에서 현재 모델의 모수의 가능도를 구하면 다음과 같다.    **베이즈 추론**  베이즈 추론은 관측값을 통해, 모델의 확률분포를 업데이트 하는 추론이다.   | **[통계] 베이즈 추론**  우리가 직접 알 수 없는 것들에 대해서 추론할 때, 모델을 상정하고 관측을 한다. 베이즈 추론은 관측이 추론으로 이어지는 과정을 담고 있다.   먼저, 베이즈 추론의 바탕이 되는 베이즈 정리를 살펴보자. 베이즈 정리는 사건 A와 B의 교집합의 확률에 관점을 부여하며 의미를 갖는다.  A와 B의 교집합의 확률은 A와 B에게 있어서는 다음과 같다.  P(A∩B) = P(A|B)P(B) = P(B|A)P(A)  이 공통 사건에 'B라는 조건에서 A가 발생할 확률'이라는 관점을 부여하면 다음과 같이 표현되고, 이를 베이즈 정리라고 부른다.  P(A|B)= P(A∩B) / P(B) = P(B|A)P(A) / P(B)  **베이즈 추론에서는,  위의 정리에서 관측할 수 있는 것과 추론 해야 할 것을 각각 B와 A로 둔다**. **보통 관측할 수 있는 것인 B는 데이터, 추론해야 할 것인 A는 모델에 해당한다**  **우리가 추정해야 할 모델 A의 확률분포 P(A)는** 모델 A에서 관측 B가 가능한 확률, 바꿔 말하면 **관측 B가 모델 A에서 가능한 정도(likelihood)를 곱해줌으로써 새로운 확률분포 P(A|B)로 갱신**된다.   모델 A에 대한 추론은 관측 B를 통해서 할 수 밖에 없기 때문에, P(A|B)P(A|B)가 이루는 확률 분포의 특징이 우리가 알 수 있는 최대한의 모델 A라는 것이 베이즈 추론의 관점이다. | | --- |    관측값 x, y를 줄여서 **x**, 파라미터 θ1, θ2 줄여서 **θ**로 두자. 그럼 베이즈 식은 다음과 같다.  P(θ|x) = P(x|θ)P(θ) / P(x)    **목적은 x를 통해 θ의 확률분포를 구하는 것이다.**   여기서 P(θ)를 사전분포, P(θ|x)를 사후분포라고 한다. 이를 이용하면, 적당한 θ 범위에 대해서 관측값에 대한 사후분포를 구할 수 있다. 하지만, θ의 개수가 늘어난다면, 우리가 사후분포를 탐색해야 하는 공간은 지수 함수 꼴로 늘어나기 때문에 파라미터 공간을 탐색을 하기 위한 전략을 짜야 한다.  **MCMC 샘플링**   여기서 등장하는 것이 마르코프 체인 몬테카를로법(MCMC)다.   마르코프 체인은 어떤 상태가 바로 전 상태에 의존하는 경우를 말한다. 몬테카를로는 무작위 샘플링을 하는 것이니 둘을 합친 MCMC는 어떤 상태가 끊임없이 움직이고 있는 체인 상태의 샘플링을 말한다.   MCMC는 특정 조건에서 마르코프 체인이 정상 상태 분포(steady-state distribution)로 ergodic한다는 특성을 이용한다. 마르코프 연쇄가 ergodic하기 위해서는, 전이 상태가 하나의 값으로 정착 되지 않을 조건, 그리고 전이가 주기성을 띄지 않을 조건을 충족해야 한다. 이 조건을 만족하는 전이 알고리즘을 이용하면 어떤 파라미터 값에서 탐색을 시작하더라도 탐색 과정에서 발견된 값들이 이루는 분포가 정상 상태 분포를 이룰 것이고, 이 분포를 통해 모델 파라미터의 확률분포를 추정할 수 있다는 것이 MCMC 베이지안 추론이다.   MCMC의 정상 상태 분포는 파라미터의 확률분포를 타겟으로 한다. 이를 위해서는 파라미터 값을 탐색할 때 마르코프 체인의 상태 전이 확률이 더 그럴싸한 파라미터 값의 상태로 전이를 해줘야 한다. 즉, **샘플링이 더 그럴싸한 파라미터를 중심으로 이루어져야, 샘플링이 이루는 분포가 우리가 타겟으로 하는 파라미터 주변으로 이루어질 것이다.**   이를 위해 **Metropolis-Hastings(MH) 알고리즘이 고안**되었다. MH 알고리즘에서는 사전분포를 제안하고, 현재 샘플링 파라미터 θt가 다음 샘플 θnew로 전이될 확률을 사후확률의 비로 결정한다.   우선 사후확률의 비를 식으로 표현하면 다음과 같다.    그리고 전이 여부를 다음 식으로 결정한다.  α(θt,θnew)=min(1,r)  윗 식들을 보면 p(x)가 사라지고, **가능도 p(x|θ)와 사전확률분포 p(θ)만으로 파라미터 탐색을 하고 있다는 것을 알 수 있다.** 사전 확률 분포는 MH알고리즘에서는 proposal 분포라고도 하는데, 사전에 파라미터에 대한 정보가 없는 경우, 이 분포는 정보를 포함하지 않는 uniform 분포나 normal 분포를 이용한다. 위 파트에서 가능도를 구했고, 분포에 대한 정보는 설계자가 지정하므로, 우리는 r을 구할 수 있다!   그럼, 전이 확률을 살펴보자. 만약, θnew에서의 사후확률이 θt 보다 크다면 r은 1보다 크게 되어 다음 상태 θt+1은 θnew로 전이된다. 만약 새로 샘플링 된 파라미터가 기존 파라미터보다 작은 사후확률을 갖는다면 r이 작아지며, θnew로 전이될 확률이 낮아지고, θt는 그대로 유지되어 θt+1이 되는 경우가 발생한다.   이런 식으로 파라미터 공간을 탐색하면 우리가 찾고자 하는 파라미터 부근 공간에서 효율적인 탐색을 할 수 있게 된다.  **베이즈 추론에서 MH 알고리즘이 작동하는 원리**를 간략하자면,  proposal 분포에서 추출한 파라미터를 그럴싸한 공간을 위주로 탐색하고, 그 탐색 경로가 이루는 분포를 통해 파라미터의 확률분포를 추정하는 알고리즘이다. |
| --- | --- |

| [**http://www.secmem.org/blog/2019/01/11/mcmc/**](http://www.secmem.org/blog/2019/01/11/mcmc/)  **Markov Chain Monte Carlo 샘플링의 마법**  [MCMC](http://www.secmem.org/tags/MCMC) , [sampling](http://www.secmem.org/tags/sampling) , [machine-learning](http://www.secmem.org/tags/machine-learning)  여기에서는 강력한 샘플링 기법 중 하나인 Markov Chain Monte Carlo(MCMC)에 대해 알아보겠습니다. MCMC의 활용도는 굉장히 넓어서 머신러닝을 비롯한 베이지안 통계학, 통계물리학, 컴퓨터비전, 자연어처리 등의 분야에 널리 쓰이고 있습니다. 하지만 MCMC는 역사가 짧고 (1990년대에 들어서 영향력을 발휘하기 시작) MCMC를 설명한 대부분의 자료가 수학적인 언어로만 쓰여진 탓에, 아직까지 개발자들에게 읽고 적용하기 좋은 가이드가 부족한 상태입니다. 특히나 한글 자료는 많이 부족합니다. 따라서 국내 개발자들도 MCMC를 이해하고 실제로 사용할 수 있도록 알고리즘을 체득하면 좋겠다고 생각해서 이 글을 작성하게 되었습니다. 부족하지만 도움이 되었으면 좋겠습니다.  **목차**   1. 샘플링이란?    1. 샘플링이 어려운 이유    2. 샘플링이 필요한 이유 2. MCMC의 개념과 수학적 원리    1. Markov Chain    2. MCMC 속 Markov Chain 3. Metropolis-Hastings 알고리즘을 익혀보자    1. 말로 풀어쓴 Metropolis-Hastings 알고리즘    2. 장점, 단점 및 구현 4. 참고문헌   **샘플링이란?**  우리에게 가장 익숙한 샘플링의 의미는 단연코 “배열에서 랜덤하게 원소를 뽑는 작업”일 것입니다. 심지어 대부분의 언어가 sample 또는 random\_sample이란 이름의 라이브러리 함수를 제공하고 있어서 한 줄의 코드로 샘플링 작업을 할 수 있지요. 하지만 통계학에서 사용하는 조금 더 넓은 의미의 샘플링은 “**임의의 확률 분포**p(x)**로부터 표본을 추출하는 작업**“입니다. 주사위로 예를 들어 보겠습니다. 우리가 주사위를 여러번 던질 때, 매번 도출된 앞면의 숫자가 표본입니다. 이 때 확률 분포는 각 주사위의 숫자가 도출될 상대적인 확률을 의미하는 값이므로 p(1) = p(2) = ⋯ = p(6) = 1/6 이 됩니다.  주사위는 상태가 1부터 6까지로 매우 적지만, **상태 개수가 매우 많거나 아예 연속적인 분포도 있습니다. 그러한 확률 분포로부터 샘플링을 하는 것은 때때로 쉬운 경우도 있으나 대부분은 매우 어려운 문제**입니다. 아무리 계산을 잘하는 컴퓨터라도 기존의 방법으로는 처리하지 못하는 범위라는게 존재하니까요.  **샘플링이 어려운 이유**    다양한 정규분포  샘플링이 non-trivial한 가장 간단한 예시는 바로 정규분포입니다. 이 분포는 −∞부터 ∞ 까지 연속적인 상태를 가지기 때문에 단순히 배열로 처리할 수는 없습니다. 가장 간단한 방법은 적절히 범위를 정해놓고 구간을 N개로 쪼갠 다음에 각 구간의 중심값으로만 샘플링 하는 것입니다. 하지만 이 방법은 상태 갯수가 N개로 고정되어 버리는 치명적인 단점이 있고, 또 양 끝쪽의 샘플을 정확하게 얻을 수가 없습니다. 만약에 이 정도의 오차를 허용하더라도 정규분포가 아닌 더 복잡한 모양의 분포, 특히 작은 구간에서 요동치는 분포의 경우에는 같은 방법을 적용할 수 없습니다. 짧지만 확실한 변화를 캐치할 수 없으니까요. **정확하고 빠르게 샘플링을 하려면 다른 방법을 생각**해야 합니다.  다행히도 정규분포와 같은 일부 “좋은 형태”의 분포들은 상수 시간으로 쉽게 샘플링을 할 수 있는 방법이 개발되었습니다. 하지만 아직까지 일반적으로 모든 확률 분포에 적용할 수 있는 매우 빠른 샘플링 기법은 개발되지 않은 상태입니다.  좀 더 철학적으로 접근해 봅시다. **만약 샘플링이 쉬운 문제라면, 즉 임의의 확률 분포로부터 빠르게 샘플링이 가능하다면 무슨 일이 일어날까요? 바로 우리는 NP-Hard 최적화 문제를 빠르게 풀 수 있게 됩니다.** 임의의 함수 f(x)의 global optimum을 찾는 최적화 문제를 푼다고 할 때, 임의의 상수 K에 대해 eKf(x)에 비례하는 확률 분포 p(x)가 존재하고 저희는 이 확률분포 p(x)로부터 샘플을 쉽게 얻을 수 있습니다. 만약 K에 큰 값을 부여한다면, f(x)의 값이 높은 x일 수록 상대적으로 도출 빈도가 매우 높아질 것입니다. 따라서 K를 무한대에 가깝게 설정하면, 결국 쉽게 얻어진 샘플들은 global optimum으로 수렴하게 됩니다. 또한 x를 subset으로 f를 subset에 속한 원소의 합으로 치환하면, 그 유명한 [**Subset sum problem**](https://en.wikipedia.org/wiki/Subset_sum_problem)이 됩니다.  **샘플링이 필요한 이유**  따라서 샘플링은 기본적으로 어렵습니다. 하지만 그럼에도 불구하고 샘플링은 꼭 필요합니다. 샘플링이 필요한 경우는 근본적으로 2가지입니다.  첫번째는 **높은 차원의 최적화 문제**를 풀 때, 두번째는 **높은 차원의 적분**을 할 때입니다. 여기서 높은 차원이란, 말 그대로 높은 차원의 실수 공간을 말하기도 하고 때로는 컴퓨터가 처리하기에 상태개수가 너무 많은 경우를 말하기도 합니다.  Subset sum problem이 어려운 이유가 선택가능한 부분집합의 개수가 너무 많아서, 즉 2N개로 높은 차원이기 때문입니다. 하지만 만약 위에서 정의한 확률분포에 대해 샘플링이 가능하다면, 그 많은 상태를 다 보지 않고도 선택적 표본을 통해 이 최적화 문제를 푸는게 가능하겠죠. 물론 이 문제를 샘플링으로 다항시간내에 해결한 것은 아니지만, 유사한 방법으로 다른 높은 차원의 non-trivial한 최적화 문제를 샘플링을 풀 수 있을지 모릅니다.  어떤 함수 f(x)를 계산할 수 있고 확률분포 p(x)를 샘플링 할 수 있을 때, 우리는 다음 형태의 적분을 아주 효과적으로 풀수 있습니다.  ∫f(x)p(x)dx  심지어 p(x)p(x)를 계산하지 못하더라도 샘플링만 할 수 있으면 됩니다. 위 적분은 다음과 같이 근사시킬 수 있습니다.    여기서 Xi들은 i번째 표본 값이고 총 N개의 표본을 추출 했다는 뜻입니다. 왜 오른쪽 항에는 p(x)가 없을까요? 조금만 생각해보면 답을 쉽게 찾을 수 있습니다. 그 이유는 표본은 확률분포를 반영하므로, 전체 표본에서 각 상태의 출현 횟수는 자신의 확률분포 값에 비례하게 됩니다. 즉, Xi가 표본이기 때문에, 샘플이기 때문에 p(x)를 곱한 것과 같은 효과가 나타나는 것입니다. N이 점점 커질 수록 대수의 법칙에 의해 각 표본의 출현 횟수는 점점 정확한 확률 분포에 수렴하게 되고 결국은 정확한 적분 값에 도달하게 됩니다. 물론 수렴속도가 너무 느리면 안되지만요.  놀랍지 않나요? 대학교 1학년 미적분학 시간때 배우는 [**리만 합**](https://ko.wikipedia.org/wiki/%EB%A6%AC%EB%A7%8C_%ED%95%A9)도 적분을 근사하지만 높은 차원에서는 불가능합니다. 결국 전체 공간을 다량의 미세한 구간으로 쪼개서 계산하는 것인데, 상태 개수가 많아지면 너무 오래 걸리니까요. 하지만 샘플링은 높은 차원에서도 적분을 가능하게 합니다. [**Wolframalpha**](https://www.wolframalpha.com/)는 적분값을 어떻게 구하고 있을지 한번 상상해봅시다.  이러한 적분 테크닉 덕분에 MCMC와 같은 샘플링 기법들이 머신러닝, 베이지안 통계학, 전산생물학 등에 쓰일 수 있는 것입니다. 언급하지는 않았지만 샘플링의 기본적인 속성인 simulation 덕분에 통계물리학에서도 널리 사용되고 있습니다.  **MCMC의 개념과 수학적 원리**  MCMC가 무엇인지 설명하기 전에 다른 샘플링 기법들에 비교하여 어떠한 점이 특별한지 먼저 언급하겠습니다. 앞에서 샘플링이 효과를 보려면 수렴속도가 느리면 안된다고 했습니다. 상태가 많은 문제, 즉 높은 차원의 문제를 풀기 위함인데 수렴속도가 느려버리면 의미가 없으니까요. 그렇습니다. **MCMC가 각광받는 이유 중 첫번째는 빠른 수렴속도입니다.** 절대적으로 빠르다고 단언할 수는 없지만 다른 샘플링 기법에 비해 상당히 빠르고 실용적으로 사용이 가능합니다. 두번째는 더 많은 확률분포를 샘플링 할 수 있습니다. Rejection sampling, Importance sampling과 같은 다른 기법들은 많은 형태의 분포를 다루지 못합니다.  이제 **MCMC가 무엇인지 자세히 알아보겠습니다. MCMC는 샘플링을 하는 일종의 패러다임**입니다. 이름에서도 알 수 있듯이 Markov Chain과 Monte Carlo라는 두가지 수학적 특성을 기반으로 샘플링을 하는 것입니다. 즉, 특정한 알고리즘을 지칭하는 말은 아닙니다. MCMC의 샘플링 방법은 이전의 샘플이 다음 샘플의 추출에 영향을 줍니다. 정확히 표현하면 MCMC는 임의의 랜덤한 표본에서 시작하여, i번째 표본을 참고하여 i+1번째 표본을 뽑습니다. 엄청나게 큰 영역에서 어떤 부분은 적게 뽑고 어떤 부분은 많이 뽑아야 하는데, 아무것도 참고를 안하면 안 되니까요.  MCMC가 놀라운 점은 겨우 바로 전 표본만을 이용하는데도, 표본을 100개, 1000개, … , 필요한 만큼 많이 뽑게 되면 현재까지 뽑은 전체 표본이 확률 분포를 거의 정확히 모방한다는 점입니다. 이 놀라운 효과를 보장하기 위해 Markov Chain을 사용합니다.  MCMC가 특정 확률 분포를 샘플링 하는 과정  **Markov Chain**  Markov Chain은 부루마블이나 모두의마블 아니면 모노폴리 같은 보드게임을 생각하면 이해하기 쉽습니다. 이와 같은 보드게임들은 여러 개의 장소가 있고 주사위를 굴려서 나온 수 만큼 다른 장소로 이동하게 됩니다. 주목할 점은 각 장소에서 다른 장소로 이동할 수 있는 특별한 확률이 존재한다는 것인데요, 만약 주사위를 2개 굴린다면 앞으로 2칸 부터 12칸 까지 이동할 수 있는데 각각의 확률이 다 다릅니다. 행여나 찬스카드나 벌칙에 걸려서 다른 장소로 이동하게 되는 경우도 있습니다. 이 모든 경우를 생각하면 장소들을 node로, 각 장소에서 다른 장소로 이동하게 될 확률을 directed edge로 표현하여 보드의 모양을 directed graph로 만들 수 있습니다. 즉, 특정한 플레이어가 매턴 마다 보드위에서 주사위를 굴리며 다른 장소로 이동하는 것은, 이 그래프 위에서 매 턴마다 정해진 확률에 따라 다른 노드로 이동하는 것과 같습니다.  Markov Chain 예시  Markov Chain도 똑같습니다. 여러 상태들 (x1,x2,⋯,xn,⋯)이 있고 xi에서 xj로 이동할 조건부 확률분포 **transition distribution** T(xj|xi)이 주어져 있어서 매턴마다 이 확률 값에 따라 상태들 사이를 이동하는 것입니다. 그렇다면 확률이 정해져 있다면 동선에 특정한 패턴이 존재하지는 않을까요? 예를 들어, “100번 이동했다면 평균적으로 3번은 출발지점에 돌아올 것이다”와 같은 예측을 할 수도 있을 것 같습니다.  예 맞습니다. 항상은 아니지만 확률이 정해져 있으므로 “특정조건”을 만족할 때 일정한 패턴이 나타납니다. 어떤 지점에서 시작하더라도, 상태 사이를 충분히 많은 횟수 이동하게 되면 각 상태의 방문횟수의 비율이 일정한 값으로 수렴하게 됩니다. 다시 말하면, 상태들의 방문횟수의 비율이 특정 확률분포로 수렴하게 되고 이 분포를 **stationary distribution**이라 부릅니다. “특정조건”에 대해서 더 궁금하신 분은 [**Markov Chain**](https://en.wikipedia.org/wiki/Markov_chain)을 더 심도있게 공부해보시길 추천드립니다.  MCMC 속 Markov Chain  어쩐지 Markov Chain의 일정한 패턴이 앞에서 언급한 MCMC의 특성과 비슷해보이지 않나요? 맞습니다. 사실 MCMC에 Markov Chain이 정확히 녹아들었기 때문입니다. **Markov Chain의 상태들을 이동하면서, 매번 이동했던 상태가 MCMC의 표본이 되는 것입니다**. 결국 Markov Chain의 특성에 의해 MCMC들의 표본들은 정확히 우리가 원하는 target distribution p(x)을 반영하게 됩니다.  그렇다면 다음 단계는 stationary distribution이 target distribution p(x)가 되도록 Markov Chain을 잡는 것입니다. 앞에서 “특정조건”에 대해서는 자세히 설명하지 않았는데 그 이유가 있습니다. 왜냐하면 **detailed balance**라 불리는 다음의 간단한 식 하나를 만족시키면 “특정조건”이 성립하기 때문입니다. (역은 성립하지 않습니다.)  p(x)T(y|x) = p(y)T(x|y)  여기서 x, y는 임의의 두 상태 입니다. 이 식을 어떻게 사용하는지는 다음 알고리즘 단원에서 자세히 알아보겠습니다.  마지막 단계는 앞에서 설정한 Markov Chain으로부터 상태간의 이동을 의도한 확률 대로 구현하는 것입니다. 이 단계를 어떤 방식으로 구현하냐에 따라서 MCMC의 여러 알고리즘이 나눠지게 됩니다.  **Metropolis-Hastings 알고리즘**  대부분의 MCMC 알고리즘들은 모두 Metropolis-Hastings(MH)의 특별한 경우이거나 MH를 확장한 알고리즘입니다. 따라서 MH가 MCMC의 가장 기본적인 알고리즘이라고 할 수 있습니다. 알고리즘이 짧지만 쉽지는 않기 때문에, pseudo code를 적기보다는 알고리즘의 흐름을 말로 풀어서 설명하겠습니다.  말로 풀어 쓴 Metropolis-Hastings 알고리즘  **1.** Target distribution을 p(x)라 합시다. MH는 p(x)에 비례하는 어떤 함수 f(x)를 계산할 수 있을 때 p(x)를 MCMC샘플링 할 수 있는 알고리즘입니다. 즉, p(x)를 몰라도 알고리즘이 성립합니다.  **2.**그렇다면 f(x)만으로 p(x)를 샘플링 할 수 있을까요? 아쉽게도 q(xt|x)라는 **쉽게 샘플링이 가능한** 조건부 확률 분포 **proposal distribution**이 더 필요합니다. 이름에서도 알 수 있듯이 이 분포는 알고리즘을 설계하는 개발자가 직접 제안하는 확률 분포입니다.(주로 정규분포를 이용합니다) MCMC기법은 다음 샘플을 뽑을 때 이전 샘플을 참고한다고 했습니다. 이전 샘플의 참고를 도와줄 함수가 바로 q(xt|x) 입니다. 이전 상태 x로 부터 다음 상태 xt를 예측하려는 목적입니다.  **3.** 그런데 transition distribution T(xt|x)가 있는데 왜 굳이 q(xt|x)가 필요할까요? 그 이유는 바로 T(xt|x)값을 정확히 알 수도 없고 샘플링 할 수도 없기 때문입니다. Detailed balance를 만족하는 T(xt|x)의 존재를 추상적으로 인식할 뿐, 직접 값을 설정할 수는 없습니다. 즉, T를 통해서는 다음 샘플을 뽑을 수가 없습니다. 그래서 q(xt|x)라는 보조 함수를 통해 현재 샘플 Xi로 부터 임시 샘플 Xt을 뽑고 q(Xt|Xi)와 T(Xt|Xi)의 차이를 통해 Xt을 다음 샘플로 인정할지 기각할지를 결정하게 됩니다.  **4.** T(Xt|Xi)가 q(Xt|Xi)보다 크다면 재빨리 Xt를 인정하는게 이득입니다. 원래 T로는 잘 샘플되는 Xt인데 보조수단인 q로는 그렇지 않기 때문에 기회가 왔을 때 인정하는게 좋습니다. T(Xt|Xi)가 q(Xt|Xi)보다 작다면, 얼마나 작은지에 따라 Xt를 인정해야 하므로 T(Xt|Xi)/q(Xt|Xi)의 확률로 인정하면 합리적일 것입니다. 따라서 **acceptance ratio**를 다음과 같이 정의합시다.  A(xt|x)=T(xt|x) / q(xt|x)  **5.** 하지만 식에 계산이 불가능한 T가 있으므로 detailed balance 식을 이용해 다음과 같이 T를 소거하고 계산 가능한 f로 대체해봅니다.    A(xt|x) / A(x|xt) = {T(xt|x)/q(xt|x)} / {T(x|xt)/q(x|xt)} = {p(xt)q(x|xt)} / {p(x)q(xt|x)} = {f(xt)q(x|xt)} / {f(x)q(xt|x)}  이 식을 만족하는 A(xt|x), A(x|xt) 쌍은 많이 있지만, Metropolis는 다음과 같이 설정했습니다.    이러면 위 식을 완벽히 만족할 뿐더러 더이상 ratio가 아닌 probability로서 기능하게 됩니다. 따라서 다음 샘플 Xi+1은 A(Xt|Xi)의 확률로 Xt로 인정하고 1−A(Xt|Xi)의 확률로 Xi로 유지합니다.  **6.** 하지만 알고리즘에 의심스러운 부분이 있습니다. T만 있었을 때와는 달리 q가 등장하고 A가 등장하니, 과연 처음에 의도했던 Markov Chain을 따라서 움직일지 확신하기가 어렵습니다. Metropolis-Hastings는 이러한 의문을 다음과 같은 논리로 해결합니다.  ① MH 알고리즘은 T가 만드는 Markov Chain 1과는 별개로 또 다른 (stationary distribution을 가지는) Markov Chain 2를 만든다.  ② 수학적으로 Markov Chain 1의 stationary distribution과 Markov Chain 2의 stationary distribution이 같음을 증명할 수 있다. (증명의 개요가 궁금하신 분들은 아래 참조.)   | <https://math.stackexchange.com/questions/2923979/what-is-the-proof-guarantee-that-metropolis-hastings-converge-to-our-required>  [**What is the (proof) guarantee that Metropolis Hastings converge to our required distribution**](https://math.stackexchange.com/questions/2923979/what-is-the-proof-guarantee-that-metropolis-hastings-converge-to-our-required)  I am trying to understand the proof behind why Metropolis Hastings (MH) will result in a stationary distribution which is proportional to the distribution from which we wish to sample from.  Here is my understanding so far:  We can easily verify that MH algorithm is an ergodic Markov Chain, under certian regularity conditions. Let say, we wish to sample from a distribution P(X), which we know upto a normalization constant. We can use a proposal distribution that Q(X) to generate samples in each run and accept them based on the condition  min(1,P(X′)/P(X) ∗ Q(X|X′)/Q(X′|X))  Also we know that, all ergodic Markov Chains have a unique stationary distribution. Let us call this stationary distribution which we can observe towards the end of this markov chain as Π(X). Now, the aim to show that Π∝P.  [One](http://wiki.helsinki.fi/download/attachments/59060372/MH_explained.pdf) of the proofs I have read on the internet starts by assuming that we converged to P, then show that once the MH algorithm converges to P, it satisfies Detailed Balance equation and hence P is stationary. I feel it is not correct to start by assuming that Π∝P, then it is stationary.  [Another proof](http://people.duke.edu/~kh269/teaching/notes/MetropolisExplanation.pdf) says that ***Detailed Balance is a necessary condition for a stationary distribution.*** Then, they were able to show - using the detailed balance equation - that if at all our MH algorithm (designed using this condition ) converges to a stationary distribution, then Π∝P. **The only weak link in this proof is the necessary condition. Wikipedia says that Detailed balance is not a necessary condition for stationary.**  Can someone please give a rigorous proof why the stationary distribution will be P(X) | | --- |   장점, 단점 및 구현  즉, 요약하면 MH는 target distribution p(x)에 비례하는 f(x)와 p(x)와 domain을 공유하는 임의의 proposal distribution q(xt|x)이 주어졌을 때, p(x)에 대한 MCMC샘플링을 하는 알고리즘입니다. 다음 그림은 실제로 proposal distribution q(xt|x)를 “평균을 x, 분산을 100으로 하는 정규분포”로 설정하고 target distribution을 p(x) ∝ 0.3e−0.2x^2+0.7e−0.2(x−10)^2로 설정했을 때 MH가 샘플링하는 과정을 나타낸 것입니다.  MH의 샘플링 과정  이렇게 보니까 MH가 정말 완벽한 알고리즘처럼 보입니다. Proposal distribution을 아무거나 골라도 항상 수렴하니까요. **하지만 MH의 한 가지 결점은 proposal distribution에 따라 수렴속도의 차이가 심하다는 것**입니다. 예를 들어 proposal이 정규분포일 때를 봅시다. 정규분포의 분산이 매우 낮으면, 자주 accept되는 대신 상태 공간에서 샘플간의 이동 거리가 매우 작을 것입니다. 그래서 만약 target distribution에 봉우리가 여러개 있다면 가장 높은 봉우리에서만 샘플링이 되는 것처럼 보이게 됩니다. 사실은 수렴속도가 느리다고 말하는게 정확합니다. 반대로 분산이 매우 높으면, 상태 공간에서 샘플 간의 이동 거리가 큰 대신 적게 accept 될 것입니다. 결과적으로 샘플들이 넓게 퍼져있지만 정돈되지 않은 형태가 보이게 됩니다. 역시 수렴속도가 느린 경우에 해당합니다. **그렇다면 수렴속도를 빠르게 하려면 자연스럽게 너무 작지도 않고 크지도 않은 분산을 설정하는게 중요**해집니다. 다음 그림이 이 논의를 시각적으로 단번에 보여줍니다.    서로 다른 proposal에 따른 MH의 결과  앞에서 언급했듯이 MH는 원리의 복잡성에 비해 구현은 굉장히 간단합니다.    MH의 pseudo code  적절한 proposal을 선택한다면 여러분도 MH알고리즘으로 MCMC 샘플링을 충분히 할 수 있습니다. 이 글이 도움이 되었으면 좋겠습니다.  참고문헌   1. [**An Introduction to MCMC for Machine Learning**](https://www.cs.ubc.ca/~arnaud/andrieu_defreitas_doucet_jordan_intromontecarlomachinelearning.pdf), C.Andrieu, 2003. 2. [**MCMC-Wikipedia**](https://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo) 3. [**Metropolis-Hastings-Wikipedia**](https://en.wikipedia.org/wiki/Metropolis%E2%80%93Hastings_algorithm) |
| --- | --- |

| <https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=sw4r&logNo=221934259566>  [참고]  **[베이지안 추론 :: 샘플링(Sampling)] MCMC 란?**  이번 포스팅에서는 **MCMC(Markov Chain Monte Carlo :: 마코프체인 몬테카를로)** 샘플링 방법에 대해서 간략히 알아보겠다.  MCMC가 무엇인지에 대해서 알아보기 이전에, 우리 모두는 베타 분포의 존재를 알고 있다. 그리고 그것의 확률 밀도 함수도 이미 잘 정립되어 있다. 하지만 이 베타 분포로부터 샘플링을 할 수 있나?  ​  MCMC는 어떠한 확률 분포로부터 샘플링이 가능하도록 하는 방법을 제공해준다. 우리가 사후 분포로부터 샘플을 원할 때, MCMC가 가장 필요하다.    ​  위에 보이는 공식은 베이즈 정리이다. 때때로, 우리는 사후 분포로부터 샘플링을 할 필요가 있다. 하지만, 우항에 분모에 있는 표준화 상수(또는 증거라고 불리는)와 함께 사후 확률을 계산하는 것이 쉬운가? 대부분의 경우, 가능도와 사전확률은 구할 수 있다.  ​  하지만, 우리는 표준화 상수 값은 구할 수 없다. 왜 인가? 이 상수 값에 대해서 확장해보자. 만약에 H가 3개의 값을 가진다고 하자.  ​    ​  P(D)는 쉽게 계산할 수 있다. 만약에 H가 연속적인 값이라면? 이제 H는 무한대의 값을 취할 것인데, 어떻게 계산할 것인가? 적분하여 푸는 것이 어려울 것이다.  ​  우리는 사후 분포로부터 샘플링을 하기를 원한다. 하지만 우리는 P(D)를 상수값으로 다루길 원한다. MCMC는 위키에 따르면,  ​  **MCMC 방법은 확률분포로부터 샘플링을 하기 위한 알고리즘의 클래스로, 마코프 체인을 만드는 것을 기반으로 한다. 이 마코프 체인은 바라는 분포를 정상 분포로써 가진다. 체인의 상태는 많은 단계 이후에 바라는 분포의 샘플로써 사용된다. 샘플의 퀄러티는 단계가 많아질 수록 향상된다.**  ​  예제와 함께 알아보자.  ​  아까 말했던 베타 분포로부터 샘플링을 하고자 한다고 해보자. 확률 밀도 함수는 아래와 같다.  ​    ​  여기서 C는 표준화 상수값이다. 그것은 실제로 알파와 베타의 함수이다. 하지만 우리는 어떻게 우리가 실제로 그것이 필요 없는지를 보이고자 한다. 그래서 우리는 그것을 상수값으로 다루고 있다.  ​  이것은 다소 복잡한 문제인데, 베타 분포에서, 만약에 풀기 어렵지 않으면 말이다. 실제로, 우리는 이것 보다 훨씬 더 복잡한 분포 함수와 함께 샘플링을 해야 할지도 모른다. 때때로 우리는 표준화 상수값을 모를 것이다.  ​  MCMC 방법들은 마코프 체인을 생성할 수 있는 알고리즘을 제공하는데, 이 체인은 이것의 정상 분포로써 베타 분포를 가지고, 균일 분포로부터 샘플링을 할 수 있다는 것이 주어져야 한다. 균일 분포는 상대적으로 샘플링 하기가 쉽다.  ​  만약에 우리가 랜덤 상태에서 시작하고, 몇몇 알고리즘을 반복하여 다음 상태로 이동한다면, 우리는 결국 마코프 체인을 생성하게 되고, 시간이 지나면 정상 분포는 베타 분포가 될 것이다. 아주 오랜 시간이 지난 체인들은 결국 베타 분포에서 샘플링을 하는 것과 같은 효과를 낼 것이다. 이것이 MCMC의 파워풀한 부분이다.  ​  이러한 MCMC 알고리즘의 하나는 Metropolis-Hastings 알고리즘이다. 또한, 조건부 확률 밀도 함수를 구할 수 있다면 Gibbs(깁스) 샘플링을 사용할 수도 있다.  ​  ​  %참고자료: <https://towardsdatascience.com/mcmc-intuition-for-everyone-5ae79fff22b1> |
| --- |

**9.6 Key Points**

We covered a lot of material in this chapter. Following are some important lessons.

**Collinearity**

• Collinearity occurs when two or more variables are highly associated. Including them in a linear model can result in confusing, nonsensical, or misleading results, because the model cannot differentiate the contribution from each of them (Sect. 9.1).

• The variance inflation factor (VIF) provides a measure of shared variance among variables in a model. A rule of thumb is that collinearity should be addressed for a variable when VIF > 5 (Sect. 9.1.2).

• Common approaches to fixing collinearity include omitting highly-correlated variables, and using principal components or factor scores (see Chap. 8) instead of individual items (Sect. 9.1.2).

**Logistic Regression**

• Logistic regression relates a binary outcome such as purchase to predictors that may include continuous and factor variables, by modeling the variables’ association with the probability of the outcome (Sect. 9.2.1).

• A logistic regression model, also known as a logit model, is a member of the generalized linear models family, and is fit using **glm( , family=binomial)** (Sect. 9.2.6).

• Coefficients in a logit model can be interpreted in terms of odds ratios, the degree to which they are associated with the increased or decreased likelihood of an outcome. This is done simply by exponentiating the coefficients with exp() (Sect. 9.2.6).

• A statistically significant result does not always mean that the model is appropriate. It is important to explore data thoroughly and to construct models on the basis of careful consideration (Sect. 9.2.7).

**Hierarchical Linear Models**

• In common marketing discussion, a hierarchical model estimates both group level effects and individual differences in effects. Such models are popular in marketing because they provide insight into differences among customers (heterogeneity) and distribution of preference. Hierarchical linear models (HLM) are exemplified when we estimate the importance of effects for individuals as well as for an overall population (Sect. 9.3).

• Effects that are associated with all observations are known as **fixed effects**, and those that differ across various grouping levels are known as **random effects** (Sect. 9.3.1).

• These models are also known as **mixed effect models**, because the total effect for each person is composed of the effect for the overall population (the fixed effect) plus the per-individual (random) effect. We estimated an **HLM using lmer() from the lme4 package** (Sect. 9.3.5).

• The difference between estimating hierarchical effects, as opposed to including the grouping variable as a factor in a standard linear model, is that a hierarchical model estimates every specified effect for each individual or group, not only a single adjustment term.

• The formula for a mixed effect model includes a **grouping term, + ( ... |group)**. Common models have a different intercept by group using (1 | group) or different intercepts and slopes for predictors within each group using (predictor | group) (Sects. 9.3.5, 9.3.6). To estimate an individual level model, the grouping term is typically the respondent identifier.

• Hierarchical models can be used to group observations **at other levels** than the individual level. For example, we might wish to group by store, advertising campaign, salesperson, or some other factor, if we want to estimate effects that are specific to such a grouping (Sect. 9.3.7).

• A common marketing application of HLM is conjoint analysis, to estimate **both overall preference and individual differences in preference**. In this chapter, we demonstrated ratings-based, or metric conjoint analysis (Sect. 9.3.2).

**Bayesian Methods for Hierarchical Linear Models**

• Hierarchical models in marketing are often estimated with Bayesian methods that are able to pool information and produce best estimates of both group and individual effects using potentially sparse data (Sect. 9.4.2).

• A Bayesian hierarchical linear model can be estimated using **MCMChregress()** in the MCMCpack package (Sect. 9.4.2).

• Model coefficients from a hierarchical model are inspected using summaries of the many estimates that are collected in an mcmc object (Sects. 9.4.2, 9.4.3).

**9.7 Data Sources**

In this final section, we offer some advice for where you might find similar data within your own organization.

In Sect. 9.1, we reanalyzed the data from Chap. 4, which describes customers online and offline transactions. In Sect. 9.2, we analyze data describing what promotional offers have been made to customers and whether they have redeemed those offers. This type of data is typically pulled from the customer relationship management (CRM) system or “customer 360” database, which is a central repository for all data describing interactions between the company and its customers. The CRM system draws this data from other systems such as the point-of-sale (cash register) system, the online retail system, the digital analytics platform, the email automation, records of direct mail, etc. If your company doesn’t have a central CRM system, then you may need to pull this data together by contacting the owners of individual systems and then using tools like **merge() in R**. This is tedious and time-consuming work, so be thankful if your company has invested in a central database for customer analytics!

The direct ratings of product profiles analyze in Sects. 9.3 and 9.4, are nearly always collected by surveying customers. They are typically collected online, using a general web survey platform such as Qualtrics, Google Forms, or SurveyMonkey. Depending on the goals of the study, this survey can be emailed to existing customers or sent through a survey research panel that specializes in finding broader samples of consumers (including non-customers). In-person surveys are often executed by having a researcher intercept respondents in a public place and hand each resopndent a tablet to complete the survey online. Collecting this data is a very easy project for a first-time analyst; students can easily collect conjoint data as part of a term project.

An alternative to metric conjoint is choice-based conjoint where respondents chose from set of product profiles. We introduce choice-based conjoint in Chap. 13. Choice surveys require a more complex design and are often collected using or a more specialized platform, such as Sawtooth Software or Conjoint.ly.

One consideration in collecting conjoint data is to ensure that respondents understand the product options and are able to rate their preferences reliably. Commonly, we would educate respondents about the product features within the survey, or would collect data during an in-person, study where the product concepts are presented using images or prototypes and discussed by a group of consumers. This latter option is similar to a focus group with an additional data collection exercise. That format has the advantage of simultaneously collecting qualitative data that help us to understand the quantitatively-measured preferences. Some drawbacks are that in-person samples

are expensive, potentially less representative, and subject to group-influence effects. **We also caution against providing customers with too much information about product options (i.e. more than they would typically get when shopping.**) Elea once participated in a study where the engineer who developed a new feature presented it to respondents. After this appealing “sales pitch,” nearly all the respondents indicated unreasonably strong preferences for this feature.

9.8 Learning More\*

The topics in this chapter are drawn from the vast range of topics related to linear modeling, and the best general recommendation is to learn about those topics broadly, as in Harrell (2015) on strategies and issues for effective regression modeling [92] and Dobson and Barnett (2018) on general linear models [43]. The following notes provide further guidance on specific topics.

**Collinearity.** The best way to learn more about collinearity and how to detect and address it is to become more fluent in linear modeling in general. Good texts for learning broadly about regression modeling are Harrell [92], and Fox and Weisberg [62].

**Logistic regression**. Logistic regression models are especially common in health sciences (modeling improvement after treatment, for instance), and much of that literature is approachable for marketers with modest translation. Hosmer et al. [99] is a standard text on such models and demonstrates the importance of model building and assessment. Binary outcomes are also often the subject of models in the machine learning community. We consider machine learning models in the context of classification in Chap. 11. A general text on those methods is Kuhn and Johnson [123].

**Hierarchical models.** The best overall didactic text on hierarchical models is Gelman and Hill [72], which provides outstanding conceptual explanation and a breadth of models with detailed code in R. The one, comparatively minor limitation of Gelman and Hill is that its level of detail and discussion can make it difficult to determine what to do when confronted with an immediate modeling need.

Support for hierarchical models (also known as **mixed effects models**) is an evolving area in R. Besides the **lme4 package** that we used, another common package is **nlme**, which has a somewhat dated companion book, Pinheiro and Bates [153]. A more up-to-date and didactic text is Galecki and Burzykowski [69].

**Bayesian hierarchical models**. We have provided only an introduction to hierarchical Bayes models and their importance, and have not covered the implementation issues and problems that may arise. To learn more about such models, there are technical introductions at varying levels of mathematical sophistication from Kruschke [120], Gelman et al. [73], and Rossi, Allenby and McCullough [168]. Gelman and Hill [72] discusses hierarchical models from both Bayesian and non-Bayesian perspectives, with examples in R.

Many Bayesian texts, including several of those noted above, discuss the implementation of MCMC samplers (as in MCMCpack). There is a caveat: they show how to write an MCMC sampler in detail, such as the internal workings of MCMChregress(). That is a valuable and reusable skill but a very technical one. For some readers, it may be similar to having an automotive engineer teach you how to drive a sedan; it is highly informative but occasionally overwhelming.

MCMCpack includes functions for several other families of Bayesian models. A general framework that handles both mixed effects and multiple response data, using the MCMC approach, is available in the MCMCglmm package [86]. If you want to do hierarchical logistic regression in a Bayesian framework, you could consider MCMCglmm.

9.9 Exercises