SPS 2467 Statistical Model Building

Generalized Linear Models

Dr. Mutua Kilai

Kirinyaga University

Review and Introduction

Let $y_1, ..., y_n$ denote n independent observations on a response.

Treat y_i as a realization of a random variable Y_i

In the general linear model we assume that

$$Y_i \sim N(\mu_i, \sigma^2)$$

And we further assume that the expected value μ_i is a linear function

$$\mu_i = X_i' \beta$$

The generalized linear model generalizes both the random and systematic component.

Components of Generalized Linear Models

All generalized linear models have three components:

- Random component
- Systematic component
- Link function

Random Component

The random component of a GLM identifies the response variable Y and selects a probability distribution for it.

Denote the observations on Y by $(Y_1, Y_2, ..., Y_n)$. Standard GLMs treat $Y_1, Y_2, ..., Y_n$ as independent.

If the observations on Y are binary then we assume a *binomial distribution* for Y

In some applications, each observation is a count. Then we have *Poisson or Negative Binomial*

If each observation is continuous, we might assume a normal distribution for Y.

Systematic Component

The systematic component of a GLM specifies the explanatory variables.

These enter linearly as predictors on the right-hand side of the model equation.

The systematic component specifies the variables that are the $\{x_j\}$ in the formula

$$\alpha + \beta_1 x_1 + ... + \beta_k x_k$$

Link Function

Denote the expected value of Y the mean of the probability distribution by $\mu = E(Y)$

The link function specifies a function $g(.)$ that relates μ to the linear predictors as

$$g(\mu) = \alpha + \beta_1 x_1 + ... + \beta_k x_k$$

The function $g(\mu)$ the link function connects the random and the systematic components.

The Exponential Family

We assume that observations come from a distribution in the exponential family with the following probability density function:

$$f(y_i; \theta_i, \phi) = exp\Big\{\frac{y_i\theta_i}{a(\phi)} + c(y_i, \phi)\Big\} \tag{1}$$

Here θ_i, ϕ are parameters and $a(.), b(.)$ and $c(.)$ are known functions.

The θ_i and ϕ are location and scale parameters respectively.

Normal Distribution

The normal distribution is given as:

$$f(y_i, \theta_i, \phi) = \frac{1}{\sqrt{2\pi\sigma}} exp\{-\frac{1}{2\sigma^2}(y_i - \mu)^2\}$$

Which can be expressed as:

$$f(y_i, \theta_i, \phi) = \exp\Big[-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y_i^2 - 2y_i\mu + \mu^2)\Big]$$

We can re-factor and have:

$$f(y_i, \theta_i, \phi) = \Big(\frac{2\mu y_i - \mu^2}{2\sigma^2}\Big) - \frac{1}{2}\Big(\frac{y_i^2}{\sigma^2} + \log(2\pi\sigma^2)\Big)$$

$\theta_i = \mu, \phi = \sigma^2, a_i(\phi) = \phi, b(\theta_i) = \frac{\theta_i^2}{2}, c(y_i, \phi) = \frac{1}{2}\Big(\frac{y_i^2}{\sigma^2} + \log(2\pi\sigma^2)$

The mean is given as $E(y_i) = b'(\theta_i)$

The variance $Var(y_i) = b''(\theta_i)a(\phi)$

Exercises

Exercise 1

The PMF of the Poisson distribution is given as:

$$f(y|\mu) = \frac{e^{-\mu}\mu^{y}}{y!}$$

Show that the Poisson Distribution can be expressed as a member of exponential family and derive the mean and variance.

Exercise 2

The PMF of the Binomial distribution is given as:

$$f(y|n,p) = \binom{n}{y} p^{y}(1-p)^{n-y}$$

Show that the binomial Distribution can be expressed as a member of exponential family and derive the mean and variance.

Exercise 3

The PMF of the Negative Binomial distribution is given as:

$$f(y|r,p) = \binom{r+y-1}{y} p^{r}(1-p)^{y}$$

Show that the negative binomial Distribution can be expressed as a member of exponential family and derive the mean and variance.

Maximum Likelihood Estimation of GLM

Unlike for the general linear model, there is no closed form expression for the MLE of β in general for GLMs.

However all the GLMs can be fit using the same algorithm a form of iteratively re-weighted least squares

Given an initial value for $\hat{\beta}$ calculate the estimated linear predictor $\hat{\eta_i} = x_i'\beta$ and use that to obtain the fitted values $\hat{\mu_i} = g^{-1}(\hat{\eta_i})$. Calculate the adjusted dependent variable

$$z_i = \hat{\eta_i} + (y_i - \hat{\mu_i})(\frac{d\eta_i}{d\mu_i})_0$$

Calculate the iterative weights

$$W_i^{-1} = (\frac{d\eta_i}{d\mu_i})V_i$$

where V_i is the variance function evaluated at $\hat{\mu_i}$

Regress z_i on x_i with weight W_i to give the new estimate of β

Logistic Regression

In logistic problems we are modeling binary data. The usual coding is that

$$Y \in \{1 = "Success\ \ or\ \ 0 = "Failure\}$$

The *Binomial* distribution is a good way to represent this kind of data.

The systematic component in our logistic regression model will be the binomial distribution.

We show that the binomial distribution belongs to the exponential family of distributions

$$\begin{aligned} f(y; \theta, \phi) &= \binom{n}{y} \pi^{y} (1-\pi)^{n-y} \\ &= \exp\left[y \log(\frac{\pi}{1-\pi}) + n \log(1-\pi) + \log \binom{n}{y} \right] \end{aligned} \tag{2}$$

Here

$$\theta = \log(\frac{\pi}{1-\pi})$$

$$b(\theta) = -\log(1-\pi) = \log(1+\exp(\theta))$$

$$\mu = b'(\theta) = \frac{\partial}{\partial \theta} \log[1+\exp(\theta)] = \frac{\theta}{1+\exp(\theta)} = \pi$$

$$g(\mu) = \log[\frac{\pi}{1-\pi}] = \theta$$

You can easily show that

$$E[Y_i] = \mu_i = n_i \pi_i$$

and

$$Var(Y_i) = \sigma_i^2 = n_i \pi_i (1-\pi_i)$$

In logistic regression the outcome is binary example

- Alive or dead
- Pass or fail
- Pay or Default

Logit Transformation

We would like to have the probabilities π_i depend on a vector of observed covariates X_i

The idea is to let π_i be a linear function of the covariates say

$$\pi_i = X_i'\beta$$

where β is a vector of regression coefficients.

We transform the probability π_i to have the odds defined as:

$$odds_i = \frac{\pi_i}{1-\pi_i}$$

Taking the natural logarithm of the odds that is *logit* or log-odds we have:

$$\eta_i = logit\pi_i = \log \frac{\pi_i}{1-\pi_i}$$

Solving for π_i we have:

$$\pi_i = logit^{-1}(\eta_i) = \frac{e^{\eta_i}}{1+e^{\eta_i}}$$

We are now in a position to define the logistic regression model by assuming that the *logit* of the probability π_i rather than the probability itself follows a ,linear model.

Logistic Regression Model

Suppose that we have k independent observations $y_1, ..., y_k$ and that the $i-th$ observation can be treated as a realization of the random variable Y_i.

We assume that Y_i has a binomial distribution

$$Y_i \sim B(n_i, \pi_i)$$

The above equation defines the stochastic structure of the model.

Suppose further that the *logit* of the underlying probability π_i is a linear function of the predictors

$$logit(\pi_i) = x_i'\beta$$

Where x_i is a vector of covariates and β is a vector of regression coefficients. This defines the systematic structure of the model.

The models defined above is a generalized linear model with binomial response and link logit.

The interpretation of β_j represents the change in the *logit* of the probability associated with a unit change in the $j-th$ predictor holding all other predictors constant.

Exponetiating equation above we find the odds for the $i-th$ unit given by

$$\frac{\pi_i}{1-\pi_i} = \exp\{x_i'\beta\}$$

This expression defines a multiplicative model for the odds.

Exponentiating we get $\exp\{x_i'\beta\}$ times $\exp\{\beta_j\}$.

The exponentiated $\exp\{\beta_j\}$ represents the *odds ratio*

Solving for the probability π_i in the logit model gives the more complicated model

$$\pi_i = \frac{\exp\{x_i'\beta\}}{1+\exp\{x_i'\beta\}}$$

Estimation and Hypothesis Testing

Maximum Likelihood Estimation

The likelihood function for n independent binomial observations is a product of densities.

Taking logs, we find that the log-likelihood function

$$\log L(\beta) = \sum \{y_i \log(\pi_i) + (n_i - y_i)\log(1-\pi_i)\}$$

where π_i depends on the covariates x_i and a vector of p parameters β through the logit transformation.

The working dependent variable z_i which has elements

$$z_i = \hat{\eta}_i + \frac{y_i - \hat{\mu}_i}{\hat{\mu}_i(\eta_i - \hat{\mu}_i)} n_i$$

Where n_i are the binomial denominators. We then regress z on the covariates calculating the weighted least squares estimate

$$\hat{\beta} = (X'WX)^{-1}X'Wz$$

Where W is a diagonal matrix of weights with entries

$$w_{ii} = \hat{\mu}_i(n_i - \hat{\mu}_i)/n_i$$

The variance is given by:

$$var(\hat{\beta}) = (X'WX)^{-1}$$

Goodness of Fit Statistic

Suppose we have just fitted a model and want to assess how well it fits the data.

A measure of discrepancy between observed and fitted values is the deviance statistic, which is given by

$$D = 2\sum\{y_i \log(\frac{y_i}{\hat{\mu}_i}) + (n_i - y_i)\log\left(\frac{n_i - y_i}{n_i - \hat{\mu}_i}\right)\} \tag{3}$$

where y_i is the observed and $\hat{\mu}_i$ is the fitted value for the $i-th$ observation.

An alternative measure of goodness of fit is *Pearson chi-squared statistic* which for binomial data can be written as

$$\chi^2_P = \sum_i \frac{n_i(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i(n_i - \hat{\mu}_i)} \tag{4}$$

Tests of Hypothesis

As usual, we can calculate Wald tests based on the large-sample distribution of the m.l.e., which is approximately normal with mean β and variance-covariance matrix.

In particular we can test the hypothesis,

$$H_0 : \beta_j = 0$$

Concerning the significance of a single coefficient by calculating the ratio of the estimate to its standard error

$$z = \frac{\hat{\beta}_j}{\sqrt{V\hat{ar}(\hat{\beta}_j)}}$$

This statistic has approximately a standard normal distribution in large samples.

The wald test can be use to calculate a confidence interval for β_j

The $100(1-\alpha)\%$ confidence that the true parameter lies in the interval with boundaries

$$\hat{\beta}_j \pm z_{1-\frac{\alpha}{2}} \sqrt{V\hat{ar}(\hat{\beta}_j)}$$

Confidence intervals for effects in the logit scale can be translated into confidence intervals for odds ratios by exponentiating the boundaries.

Example 1

A researcher is interested in how variables, such as GRE (Graduate Record Exam scores), GPA (grade point average) and prestige of the undergraduate institution, effect admission into graduate school. The response variable, admit/don't admit, is a binary variable.

```r
mydata <- read.csv("admit.csv")
knitr::kable(head(mydata))
```

admit	gre	gpa	rank
0	380	3.61	3
1	660	3.67	3
1	800	4.00	1
1	640	3.19	4
0	520	2.93	4
1	760	3.00	2

The code below estimates a logistic regression model using the glm (generalized linear model) function. First, we convert rank to a factor to indicate that rank should be treated as a categorical variable.

```r
# convert rank to factor
mydata$rank <- factor(mydata$rank)

# fit the logistic regression model
mylogit <- glm(admit ~ gre + gpa + rank, data = mydata, family = "binomial")

# output a summary table neatly
```

```r
library(gtsummary)

# output without odds ratio

tbl_regression(mylogit)
```

Characteristic	**log(OR)**	**95% CI**	**p-value**
gre	0.00	0.00, 0.00	0.038
gpa	0.80	0.16, 1.5	0.015
rank			
1	—	—	
2	-0.68	-1.3, -0.06	0.033
3	-1.3	-2.0, -0.67	<0.001
4	-1.6	-2.4, -0.75	<0.001

The logistic regression coefficients give the change in the log odds of the outcome for a one unit increase in the predictor variable.

Both gre and gpa are statistically significant, as are the three terms for rank.

- For every one unit change in gre, the log odds of admission (versus non-admission) increases by 0.002
- For a one unit increase in gpa, the log odds of being admitted to graduate school increases by 0.804
- The indicator variables for rank have a slightly different interpretation. For example, having attended an undergraduate institution with rank of 2, versus an institution with a rank of 1, changes the log odds of admission by -0.675.

We can test for an **overall effect** of rank using the **wald.test** function.

```r
library(aod)
wald.test(b = coef(mylogit), Sigma = vcov(mylogit), Terms = 4:6)
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 20.9, df = 3, P(> X2) = 0.00011
```

The chi-squared test statistic of 20.9, with three degrees of freedom is associated with a p-value of 0.00011 indicating that the overall effect of rank is statistically significant.

The odds ratio with their respective CI is given as

```r
# table with odds ratio
library(gtsummary)

tbl_regression(mylogit, exponentiate = TRUE)
```

Characteristic	**OR**	**95% CI**	**p-value**
gre	1.00	1.00, 1.00	0.038
gpa	2.23	1.17, 4.32	0.015
rank			
1	—	—	
2	0.51	0.27, 0.94	0.033

Characteristic	**OR**	**95% CI**	**p-value**
3	0.26	0.13, 0.51	<0.001
4	0.21	0.09, 0.47	<0.001

Now we can say that for a one unit increase in gpa, the odds of being admitted to graduate school (versus not being admitted) increase by a factor of 2.23.

The fitted model is given by:

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_{i2} + \beta_{4X_{i3}} + \beta_{5X_{i4}}$$

$$\log\left(\frac{\pi}{1-\pi}\right) = -3.98 + 0.002X_1 + 0.80X_2 - 0.68X_3 - 1.3X_4 - 1.6X_5$$

We can predict a new variable log of the odds and have:

If the gpa score is 3.8, gre score is 4.0 and the rank of the student is 2

Then:

$$\log\left(\frac{\pi}{1-\pi}\right) = -3.98 + 0.002 \times 4 + 0.80 \times 3.8 - 0.68 \times 1 - 1.3 \times 0 - 1.6 \times 0$$

The odds is the exponentiate of the log-odds as follows:

$$\pi = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_{i2} + \beta_{4X_{i3}} + \beta_{5X_{i4}}}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_{i2} + \beta_{4X_{i3}} + \beta_{5X_{i4}}}}$$

Likelihood Ratio Test

The **likelihood ratio test** is used to test the null hypothesis that any subset of $\beta^{'s}$ is equal to zero.

The likelihood ratio test statistic is given as

$$\Lambda^* = -2(L(\beta^{\hat{(0)}}) - L(\hat{\beta}))$$

where $l(\hat{\beta})$ is the log-likelihood of the fitted model $l(\beta^{\hat{(0)}})$ is the log-likelihood of the reduced model specified by the null hypothesis evaluated at the maximum likelihood estimate of that reduced model.

The test statistic has a χ^2 distribution with $k - r$ degrees of freedom.

Statistical software often presents results for this test in terms of deviance, which is defined as -2 times log-likelihood.

We can compare the two models as:

- Fit one model without the rank variable
- Fit another model with the rank variable

```r
# model 1

model1 <- glm(admit ~ gre + gpa, data = mydata, family = "binomial")
summary(model1)
```

```
##
## Call:
## glm(formula = admit ~ gre + gpa, family = "binomial", data = mydata)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.949378   1.075093  -4.604 4.15e-06 ***
## gre          0.002691   0.001057   2.544   0.0109 *
## gpa          0.754687   0.319586   2.361   0.0182 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 480.34  on 397  degrees of freedom
## AIC: 486.34
##
## Number of Fisher Scoring iterations: 4
```

```r
# Model 2

# convert rank to factor
mydata$rank <- factor(mydata$rank)

# fit the logistic regression model
model2 <- glm(admit ~ gre + gpa + rank, data = mydata, family = "binomial")
summary(model2)
```

```
##
## Call:
## glm(formula = admit ~ gre + gpa + rank, family = "binomial",
##     data = mydata)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.989979   1.139951  -3.500 0.000465 ***
## gre          0.002264   0.001094   2.070 0.038465 *
## gpa          0.804038   0.331819   2.423 0.015388 *
## rank2       -0.675443   0.316490  -2.134 0.032829 *
## rank3       -1.340204   0.345306  -3.881 0.000104 ***
## rank4       -1.551464   0.417832  -3.713 0.000205 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 458.52  on 394  degrees of freedom
## AIC: 470.52
##
## Number of Fisher Scoring iterations: 4
```

The deviance statistic is 480.34 - 458.52 = 21.82. The $\chi^2_{1,0.05} = 3.84$ thus we conclude that the full model is better than the reduced model.

Poisson Regression Models

In this lecture we study log-linear models for count data under the assumption of a Poisson error structure.

Poisson Distribution

A random variable Y is said to have a Poisson distribution with parameter μ if it takes integer values $y = 0, 1, 2, ...$ with probability

$$Pr\{Y = y\} = \frac{e^{-\mu}\mu^y}{y!} \tag{5}$$

We show that the Poisson Model belongs to the exponential family of distribution as:

$$\begin{aligned} f(y) &= \frac{e^{-\mu}\mu^y}{y!} \\ &= \exp(y \ln \mu - \mu - \ln(\mu!)) \end{aligned} \tag{6}$$

$$\theta = \ln \mu, c(y, \phi) = \ln y!$$

$$b(\theta) = \exp(\theta), b'(\theta) = e^\theta \Rightarrow E(Y) = \mu, Var(Y) = b''(\theta) = e^\theta = \mu$$

The canonical link function is $g(\mu) = \ln(\mu)$

Log-Linear Models

Suppose we have a sample of n observations $y_1, y_2, ..., y_n$ which can be treated as realizations of independent Poisson random variables with $Y_i \sim P(\mu_i)$ and suppose that we want to let mean μ_i depend on the vector of explanatory variables x_i

A simple approach is to take logs calculating $\eta_i = \log(\mu_i)$ and assume that the transformed mean follows a linear model $\eta_i = x_i'\beta$

Thus we consider a generalized linear model with link log. Combining these two steps we write the log-linear model as

$$\log(\mu_i) = x_i'\beta$$

In the model, the regression coefficient β_j represents the expected change in the log of the mean per unit change in the predictor x_j

Maximum Likelihood Estimation

The likelihood function for n independent Poisson observations is a product of the probabilities.

Taking logs and ignoring a constant involving $\log(y_i!)$ we find that the log-likelihood function is

$$\log L(\beta) = \sum_i \{y_i \log(\mu_i) - \mu_i\}$$

where μ_i depends on the covariates x_i and a vector of p parameters β through the log link

It is interesting to note that the log is the canonical link function for the Poisson distribution.

Goodness of Fit

A measure of discrepancy between observed and fitted values is the deviance

$$D = 2\sum\{y_i \log(\frac{y_i}{\hat{\mu_i}}) - (y_i - \hat{\mu_i})\}$$

For large samples the distribution of the deviance is approximately a chi-squared with $n - 1$ degrees of freedom.

Thus the deviance can be used directly to test the goodness of fit of the model.

Tests of hypothesis

Likelihood ratio tests for log-linear models can easily be constructed in terms of deviances, just as we did in logistic regression models.

In general, the difference in deviances between two nested models has approximately in large samples a chi-squared distribution with degrees of freedom equal to the difference in the number of parameters between the models, under the assumption that the smaller model is correct.

One can also construct Wald tests aswe have done before based on the fact that the MLE $\hat{\beta}$ has approximately in large samples a multivariate normal distribution with mean equal to the large parameter value β and variance-covariance matrix $Var(\hat{\beta}) = X'WX$

Dispersion Tests

The adjusted deviance is defined as the deviance divided by the degrees of freedom.

A value closer to 1 indicates that there is a satisfactory goodness-of-fit.

Usually a value of greater than 1 indicates signs of over-dispersion.

Overdispersion means that the variance of the response Y is greater than what's assumed by the model.

Over-dispersion can occur as a result of:

- Clustering: Different units or observations may have different levels of inherent variability leading to overdispersion
- Heterogeneity: Correlation or dependence between observations within clusters can contribute to overdispersion

In some cases, there may be under-dispersion, that is where the conditional variance is less than the conditional mean.

Example

```
dataset <- read.csv("poisson.csv")
dataset$prog <- factor(dataset$prog,
  levels = 1:3,
  labels = c("General", "Academic", "Vocational")
)
m1 <- glm(num_awards ~ prog + math, family = poisson, data = dataset)

tbl_regression(m1)
```

Characteristic	**log(IRR)**	**95% CI**	**p-value**
prog			

Characteristic	log(IRR)	95% CI	p-value
General	—	—	
Academic	1.1	0.44, 1.9	0.002
Vocational	0.37	-0.49, 1.3	0.4
math	0.07	0.05, 0.09	<0.001

Evaluating the model

The goodness-of fit is assessed via the chi-square test for goodness of fit.

```r
# Check if the model fits the data, this is the null hypothesis.
with(
  m1,
  cbind(
    res.deviance = deviance,
    df = df.residual,
    p = pchisq(deviance, df.residual, lower.tail = FALSE)
  )
)
```

```
##      res.deviance  df         p
## [1,]     189.4496 196 0.6182274
```

We can compare two models using the ANOVA table and get the chisquare value.

Here we fit a second model as:

```r
m2 <- glm(num_awards ~ math, family = poisson, data = dataset)
```

We compare

```r
# Attention on the output: Model 1 is actually m2
anova(m2, m1, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: num_awards ~ math
## Model 2: num_awards ~ prog + math
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       198     204.02
## 2       196     189.45  2   14.572 0.0006852 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Here we see that the fitted model that includes the prog variable is a significantly better predictor of num_awards.

Ridge Regression

Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity.

When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value.

Ridge regression is carried out on the linear regression model

$$Y = X\beta + \epsilon$$

where

Y is the $n \times 1$ vector of observations of the dependent variable

X is the $N \times K$ matrix of regressors

β is the $k \times 1$ vector of regression coefficients

ϵ is the $n \times 1$ vector of errors

Ridge Estimator

The objective function is given by

$$f(\beta) = (y - X\beta)'(y - X\beta) + \lambda\beta'\beta$$

We differentiate the function with respect to β and set the result equal to zero and have:

$$\frac{\partial f(\beta)}{\partial \beta} = -2X^T(y - X\beta) + 2\lambda\beta = 0$$

Solving for β

$$X^TX\beta + \lambda I\beta = X^Ty$$

Then

$$\hat{\beta_{Ridge}} = (X^TX + \lambda I)X^Ty$$

Where λ is a positive constant

Bias and Variance of Ridge Estimator

We derive the bias and the variance of the ridge estimator under the commonly made assumption that conditional on X, the errors have a zero mean and a constant variance σ^2 and are uncorrelated.

$$E[\epsilon|X] = 0$$

$$Var[\epsilon|X] = \sigma^2 I$$

where σ^2 is a positive constant and I is the $n \times n$ identity matrix.

Bias

The conditional expected value of the ridge estimator $\hat{\beta_\lambda}$ is

$$E[\hat{\beta_\lambda}|X] = (X^TX + \lambda I)^{-1}X^TX\beta$$

which is different from β unless the $\lambda = 0$

The bias of the estimator is

$$E[\hat{\beta_\lambda}|X] - \beta = \Big[(X^TX + \lambda I)^{-1} - (X^TX)^{-1}\Big]X^TX\beta$$

Proof

We can write the ridge estimator as

$$\begin{aligned} \hat{\beta_\lambda} &= (X^TX + \lambda I)^{-1}X^Ty \\ &= (X^TX + \lambda I)^{-1}X^T(X\beta) + \epsilon \\ &= (X^TX + \lambda I)^{-1}X^TX\beta + (X^TX + \lambda I)^{-1}X^T\epsilon \end{aligned} \tag{7}$$

Therefore

$$\begin{aligned} E[\hat{\beta_\lambda}] &= (X^TX + \lambda I)^{-1}X^TX\beta + (X^TX + \lambda I)^{-1}X^TE[\epsilon|X] \\ &= (X^TX + \lambda I)^{-1}X^TX\beta + (X^TX + \lambda I)^{-1}X^T \times 0 \\ &= (X^TX + \lambda I)^{-1}X^TX\beta \end{aligned} \tag{8}$$

The ridge estimator is unbiased if and only if

$$(X^TX + \lambda I)^{-1}X^TX = I$$

Variance

The covariance of the ridge estimator is given by:

$$Var[\hat{\beta_\lambda}|X] = \sigma^2(X^TX + \lambda I)^{-1}X^TX(X^TX + \lambda I)^{-1}$$

Proof

Remember that the OLS estimator $\hat{\beta}$ has conditional variance

$$Var[\hat{\beta}] = \sigma^2(X^TX)^{-1}$$

We can write the ridge estimator as a function of the OLS estimator

$$\begin{aligned} \hat{\beta_\lambda} &= (X^TX + \lambda I)^{-1}X^Ty \\ &= (X^TX + \lambda I)^{-1}X^TX(X^TX)^{-1}X^Ty \\ &= (X^TX + \lambda I)^{-1}X^TX\hat{\beta} \end{aligned} \tag{9}$$

Therefore:

$$\begin{aligned} Var[\hat{\beta_\lambda}] &= (X^TX+\lambda I)^{-1}X^TXVar[\hat{\beta_\lambda}][(X^TX+\lambda I)^{-1}X^TX]^T \\ &= (X^TX+\lambda I)^{-1}X^TXVar[\hat{\beta_\lambda}X^TX(X^TX+\lambda I)^{-1}] \\ &= (X^TX+\lambda I)^{-1}X^TX\sigma^2(X^TX)^{-1}X^TX(X^TX+\lambda I)^{-1} \\ &= \sigma^2(X^TX+\lambda I)^{-1}X^TX(X^TX+\lambda I)^{-1} \end{aligned} \tag{10}$$

How to choose λ

The most common way to find the best λ is by using leave-one-out cross-validation.

The steps are as follows:

- We choose a grid of p possible values of $\lambda_1, \lambda_2, ..., \lambda_p$ for the penalty parameter
- for $i = 1, ..., N$ we exclude the $i-th$ observation (y_i, x_i) from the sample and use the remaining $n-1$ observations to compute p ridge estimates of β denoted by $\beta_{\widehat{\lambda p,i}}$ and compute p out-of-sample predictions of the excluded observation
- We compute the MSE of the predictions

$$MSE_\lambda = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_{\lambda p})^2$$

- We choose as the optimal penalty parameter λ the one that minimizes the MSE of the predictions

Example

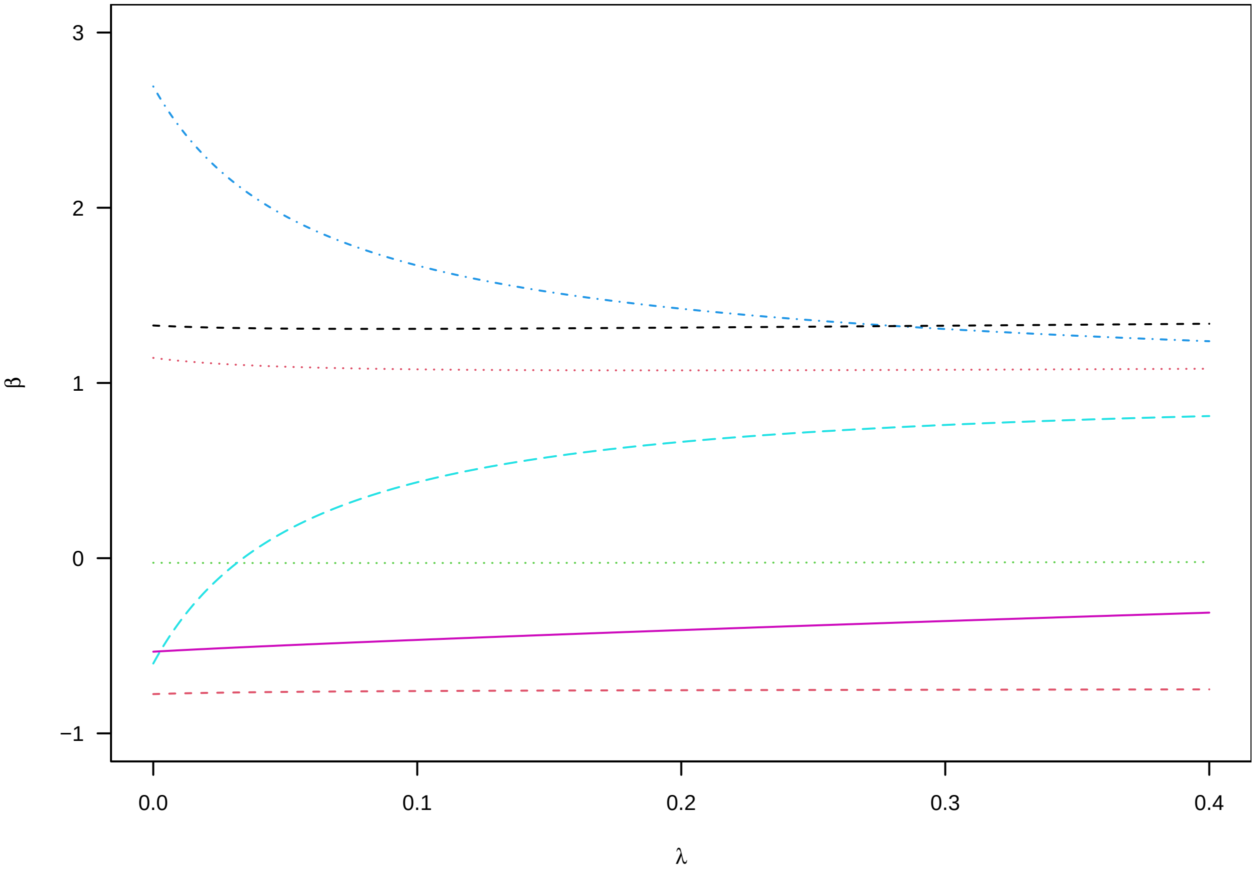
As the beginning of ridge regression, it is recommended to standardize the predictors. You can still carry out ridge regression without doing so, but standardization would improve the effect of ridge regression, as it makes the shrinking fair to each coefficients. Luckily, the function that we are going to use here automatically standardizes the data, so we don't need to do the standardization by ourselves.

We use the `MASS` package in R

```r
# loading the data

data <- read.csv("ridge.csv")

# package to use

library(MASS)

# model with a range of lambdas

fit = lm.ridge(hipcenter ~ ., data, lambda = seq(0, .4, 1e-3))
```

We can observe how the coefficients shrink as λ grows larger:

```r
par(mar = c(4, 4, 0, 0), cex = 0.7, las = 1)
matplot(fit$lambda, coef(fit), type = "l", ylim = c(-1, 3),
        xlab = expression(lambda), ylab = expression(hat(beta)))
```



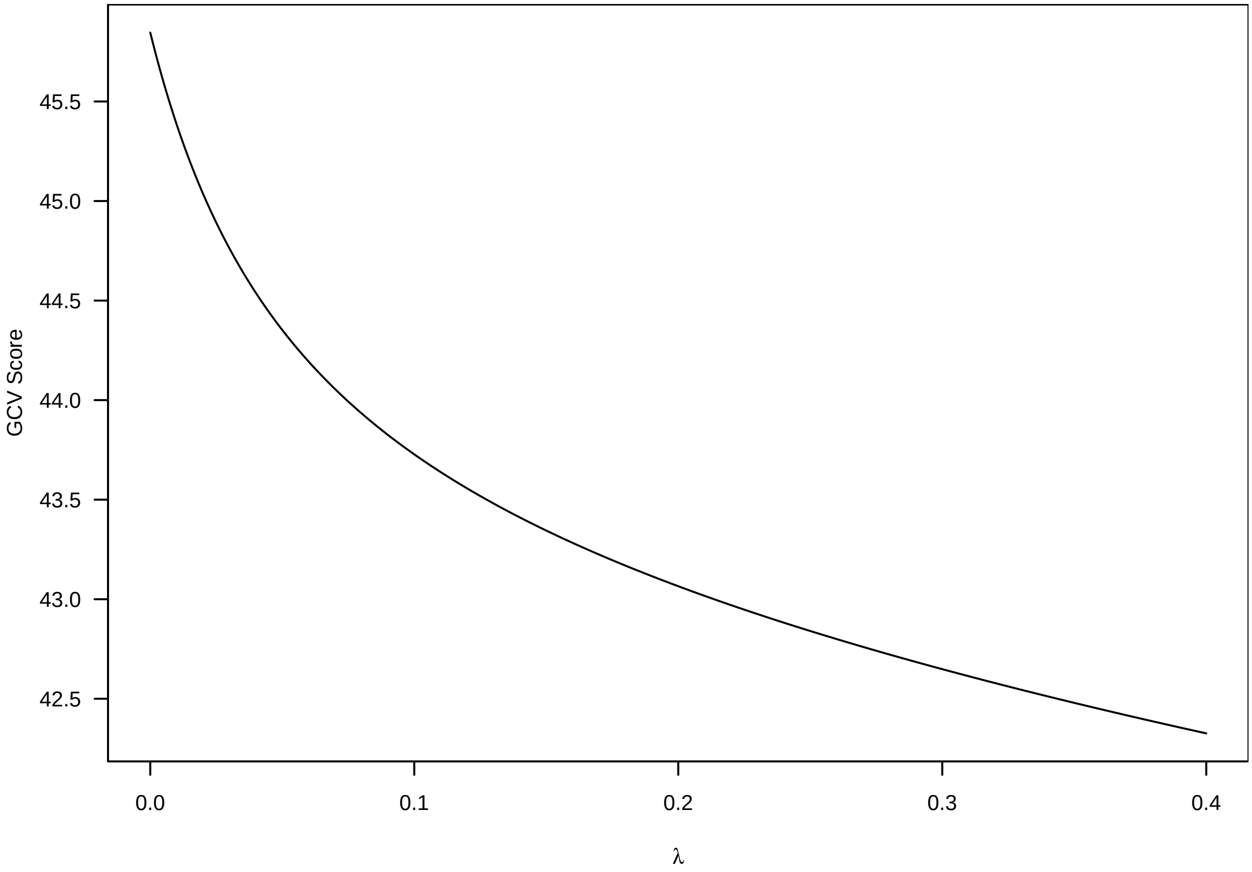
To select the optimal value of λ we use `select` function

```r
select(fit)
```

```
## modified HKB estimator is 5.425415
## modified L-W estimator is 3.589434
## smallest value of GCV  at 0.4
```

So the optimal value of λ is at 0.4

```r
par(mar = c(4, 4, 0, 0), cex = 0.7, las = 1)
plot(names(fit$GCV), fit$GCV, type = 'l',
     xlab = expression(lambda), ylab = "GCV Score")
```



Detecting Outliers in Regression Models

Outliers are observations that appear inconsistent with the rest of dataset.

A more precise definition, they are observations that are distinct from most of the data points in the sample.

There are many methods of detecting outliers in regression models. They include:

- Graphical methods
- Analytical methods

Graphical Methods

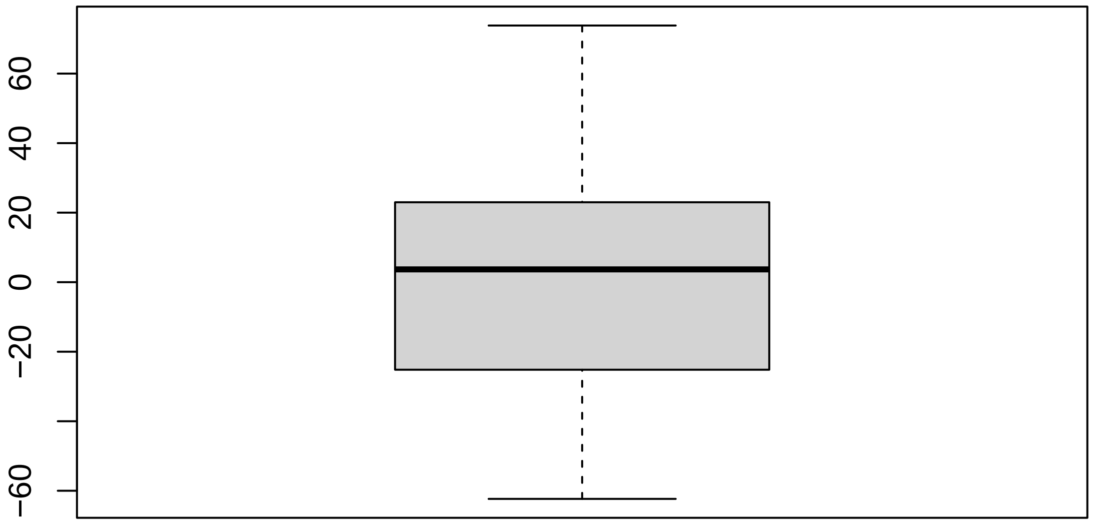
The graphical methods include scatter graph, boxplot, williams graph, Q-Q plot and graph of predicted residuals

Scatter and Box plot

Scatter plot is a line of best fit (alternatively called "trendline") drawn in order to study the relationship between the variables measured. For a set of data variables (dimensions) $X_1, X_2, ..., X_k$the scatter plot matrix shows all the pairwise scatter plots of the variables on the dependent variable.

A box plot is a method for graphically depicting groups of numerical data through their quartiles (i.e. Mean, Median Mode, quartiles). Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles. It is also called box-and-whisker plot and box-and-whisker diagram. Outliers may be plotted as individual points and it can be used for outlier detection in regression model, where the primary aim here is not to fit a regression model but find out outliers using regression and to improve a regression model by removing the outliers.

```r
# loading the data

data <- read.csv("ridge.csv")

# model

fit = lm(hipcenter ~ ., data)

# extract the residuals

res <- fit$residuals

# boxplot of the residuals

boxplot(res)
```



Analytical Methods

The analytical methods include:

- predicted residuals
- Standardized residuals
- Jack-nife residuals
- Cook's distance
- Atkinsons measure

Studentized and Standardized Residuals

The Standardized residuals are given by:

$$\epsilon_{S.i} = \frac{\hat{\epsilon_i}}{\hat{\sigma}\sqrt{1-h_i}} = \frac{\hat{\epsilon_i}}{\sqrt{\hat{\sigma^2}(1-h_i)}}$$

Studentized residuals with large absolute values are considered large. If the regression model is appropriate, with no outlying observations, each Studentized residual follows a t distribution with n-k-1 degrees of freedom.

If the Studentized residual is divided by the estimates of its standard error so that the outcome is a residual with zero mean and standard deviation one, it becomes standardized residual denoted by

$$\epsilon_{ST.i} = \frac{\hat{\epsilon_i}}{sd(\sigma)}$$

The standardized residuals, $d_i > 3$ potentially indicate outlier

Response Surface Methodology

Response Surface Methodology is a collection of statistical and mathematical techniques used for the purpose of:

- Setting up a series of experiments (design) for adequate predictions of a response y.
- Fitting a hypothesized (empirical) model to data obtained under the chosen design.
- Determining optimum conditions on the model's input (control) variables that lead to maximum or minimum response within a region of interest

The *response surface* $\phi(.)$ relates the expected response to the experimental variables

$$\mathbb{E}(y) = \phi(x_1, ..., x_k)$$

The goal of a response surface design is to define n design points $x_{1,j}..., x_{k,j}, j = 1....n$ and use a reasonably simple yet flexible regression function $f(.)$ to approximate the true response surface $\phi(.)$ from the resulting measurements y_j so that $f(x_1, ..., x_k) \approx \phi(x_1, ..., x_k)$.

We discuss two commonly used models for approximating the response surface:

- *first-order model*: requires only a simple experimental design but does not account for curvature of the surface and predicts ever-increasing responses along the path of steepest ascent.

- *second-order model*: allows for curvature and has a defined stationary point (a maximum, minimum, or saddle-point), but requires a more complex design for estimating its parameters.

First Order Model

The first-order model for k quantitative factors (without interactions) is

$$y = f(x_1, ..., x_k) + e = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k + e = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + e$$

We estimate its parameters using standard linear regression; the parameter β_i gives the amount by which the expected response increases if we increase the $i - th$ factor from x_i to $x_i + 1$ keeping all other factors fixed.

Without interactions, the predicted change in the response is independent of the values of the other factors and interactions could be added if necessary.

The predicted response is:

$$\hat{y} = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i x_i$$

Example 1

Using the data named *first_order.csv* fit a first order model.

```
data <- read.csv("first_order.csv")

# loading the package

library(rsm)
```

```
# fit the first-order model

rsm.model <- rsm(y ~ FO(x1, x2), data = data)
summary(rsm.model)
```

```
##
## Call:
## rsm(formula = y ~ FO(x1, x2), data = data)
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 70.41667    3.32793 21.1593 5.521e-09 ***
## x1           1.75301    0.89477  1.9592   0.08175 .
## x2           0.61446    0.89477  0.6867   0.50956
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.3238, Adjusted R-squared:  0.1736
## F-statistic: 2.155 on 2 and 9 DF,  p-value: 0.1719
##
## Analysis of Variance Table
##
## Response: y
##             Df  Sum Sq Mean Sq F value  Pr(>F)
## FO(x1, x2)   2  572.80 286.401   2.155 0.17191
## Residuals    9 1196.12 132.902
## Lack of fit  6 1169.37 194.894  21.857 0.01424
## Pure error   3   26.75   8.917
##
## Direction of steepest ascent (at radius 1):
##        x1        x2
## 0.9437066 0.3307837
##
## Corresponding increment in original units:
##        x1        x2
## 0.9437066 0.3307837
```

We can fit a first-order with two-way interaction model as:

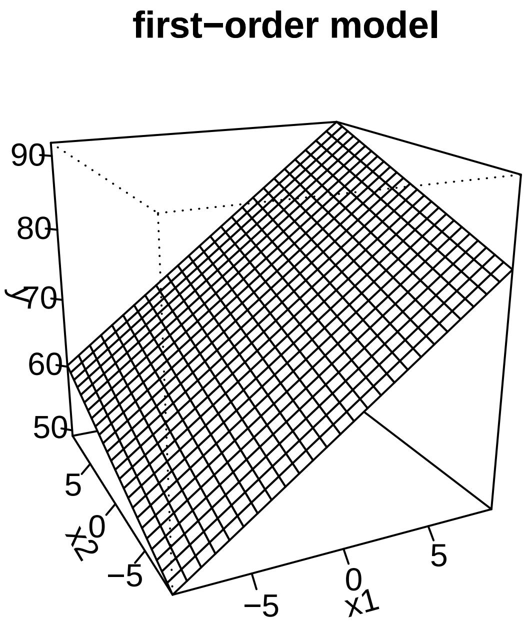
```
# fit the first-order with two-way interaction model.

rsm_twoway <- rsm(y ~ FO(x1, x2) + TWI(x1, x2), data = data)
summary(rsm_twoway)
```

```
##
## Call:
## rsm(formula = y ~ FO(x1, x2) + TWI(x1, x2), data = data)
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 70.41667    3.15546 22.3158 1.719e-08 ***
## x1           1.75301    0.84840  2.0663   0.07265 .
## x2           0.61446    0.84840  0.7243   0.48954
## x1:x2       -7.75000    5.46542 -1.4180   0.19394
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Multiple R-squared:  0.4596, Adjusted R-squared:  0.257 
## F-statistic: 2.268 on 3 and 8 DF,  p-value: 0.1576
## 
## Analysis of Variance Table
## 
## Response: y
##             Df Sum Sq Mean Sq F value Pr(>F)
## FO(x1, x2)   2 572.80 286.401  2.3970 0.1529
## TWI(x1, x2)  1 240.25 240.250  2.0107 0.1939
## Residuals    8 955.87 119.483               
## Lack of fit  5 929.12 185.823 20.8400 0.0155
## Pure error   3  26.75   8.917               
## 
## Stationary point of response surface:
##         x1         x2 
## 0.07928488 0.22619510 
## 
## Eigenanalysis:
## eigen() decomposition
## $values
## [1]  3.875 -3.875
## 
## $vectors
##          [,1]       [,2]
## x1 -0.7071068 -0.7071068
## x2  0.7071068 -0.7071068
```

A plot of the model

```r
persp(rsm.model, x2 ~ x1, zlab = "y", main="first-order model")
```



The Second-Order Model

The second-order model adds purely quadratic (PQ) terms x_i^2 and two-way interaction terms $x_i.x_j$ to the first-order model such that the regression model equation becomes

$$y = f(x_1, ..., x_k) + e = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i<j}^{k} \beta_{ij} x_i x_j + e$$

and we assume a constant error variance $Var(e) = \sigma^2$ for all points.

For example the two-factor second-order response surface approximation is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{1,1} x_1^2 + \beta_{2,2} x_2^2 + \beta_{1,2} x_1 x_2 + e$$

only requires estimation of six parameters to describe the true response surface locally.

The second-order model allows curvature in all directions and interactions between factors which provides more information about the shape of the response surface.

```r
library(rsm)

# fit second-order (SO) model

rsm_so <- rsm(y ~ SO(x1, x2), data = data)
summary(rsm_so)
```

```
## 
## Call:
## rsm(formula = y ~ SO(x1, x2), data = data)
## 
##              Estimate Std. Error t value  Pr(>|t|)    
## (Intercept) 72.927104   4.026565 18.1115 1.824e-06 ***
## x1           1.753012   0.872972  2.0081    0.0914 .  
## x2           0.614458   0.872972  0.7039    0.5079    
## x1:x2       -7.750000   5.623726 -1.3781    0.2174    
## x1^2        -0.137035   0.110586 -1.2392    0.2616    
## x2^2        -0.044442   0.110586 -0.4019    0.7017    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Multiple R-squared:  0.5709, Adjusted R-squared:  0.2133 
## F-statistic: 1.597 on 5 and 6 DF,  p-value: 0.2912
## 
## Analysis of Variance Table
## 
## Response: y
##             Df Sum Sq Mean Sq F value  Pr(>F)
## FO(x1, x2)   2 572.80 286.401  2.2639 0.18511
## TWI(x1, x2)  1 240.25 240.250  1.8991 0.21736
## PQ(x1, x2)   2 196.83  98.417  0.7780 0.50071
## Residuals    6 759.03 126.505                
## Lack of fit  3 732.28 244.094 27.3750 0.01111
## Pure error   3  26.75   8.917                
## 
## Stationary point of response surface:
##         x1         x2 
## 0.07672176 0.22348192 
## 
## Eigenanalysis:
## eigen() decomposition
## $values
## [1]  3.784538 -3.966015
## 
## $vectors
##          [,1]       [,2]
## x1  0.7028703 -0.7113180
## x2 -0.7113180 -0.7028703
```

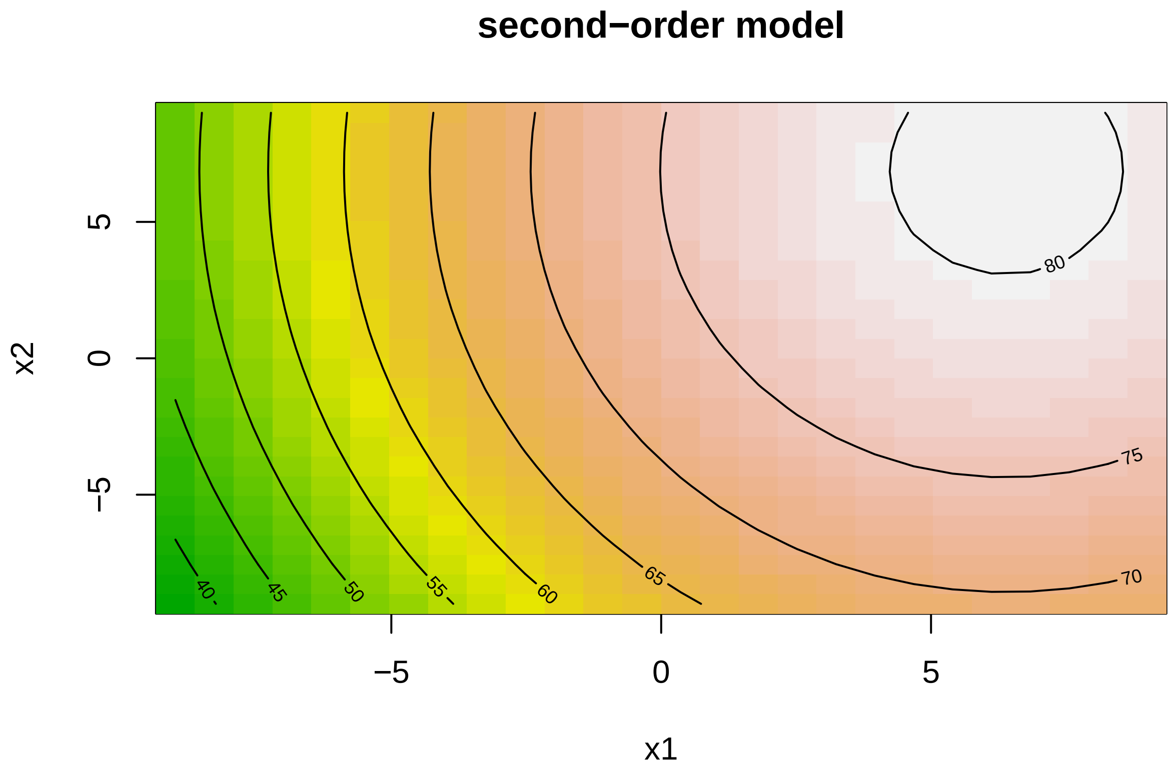
A second-order model without interactions can be fitted in R as

```r
# Fit the second-order without interactions model
rsm_som <- rsm(y ~ FO(x1, x2) + PQ(x1, x2), data = data)
summary(rsm_som)
```

```
## 
## Call:
## rsm(formula = y ~ FO(x1, x2) + PQ(x1, x2), data = data)
## 
##              Estimate Std. Error t value  Pr(>|t|)    
## (Intercept) 72.927104   4.277356 17.0496 5.856e-07 ***
## x1           1.753012   0.927344  1.8904    0.1006    
## x2           0.614458   0.927344  0.6626    0.5288
## x1^2        -0.137035   0.117474 -1.1665    0.2816
## x2^2        -0.044442   0.117474 -0.3783    0.7164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.4351, Adjusted R-squared:  0.1123
## F-statistic: 1.348 on 4 and 7 DF,  p-value: 0.3418
##
## Analysis of Variance Table
##
## Response: y
##             Df Sum Sq Mean Sq F value  Pr(>F)
## FO(x1, x2)   2 572.80 286.401  2.0062 0.20476
## PQ(x1, x2)   2 196.83  98.417  0.6894 0.53297
## Residuals    7 999.28 142.754
## Lack of fit  4 972.53 243.133 27.2672 0.01077
## Pure error   3  26.75   8.917
##
## Stationary point of response surface:
##       x1       x2
## 6.396220 6.912965
##
## Eigenanalysis:
## eigen() decomposition
## $values
## [1] -0.04444242 -0.13703501
##
## $vectors
##    [,1] [,2]
## x1    0   -1
## x2   -1    0
```

A countour plot is given as

```
# second-order model
contour(rsm_som, ~ x1 + x2, image = TRUE, main="second-order model")
```



Choice of a response

Some important properties of a response surface design include:

- Generation of a satisfactory distribution of information throughout the region of interest
- Closeness of $\hat{y}$ to y over R
- Good detectibility of lack of fit
- Insensitivity (robustness) to extreme observations and to violations of the usual normal theory assumptions.
- Ability to perform experiments in blocks.
- Extendibility to a higher-order design.
- Requiring a small number of experimental runs.