# PREDICTION

巫小珍

2022-08-06

## 目录

# 4 预测

预言是一门很好的生意，但却充满风险。—— 马克・吐温《赤道漫游记》

## 4.1 预测选举结果

### 4.1.1 R 的循环语句

```
for( i in X) {

expression

}
```

`cat()` 和 `print()` 函数可以将对象打印出来。

参数：`sep =` “” 表示对象以什么分隔。

```
values <- c(2, 4, 6)
n <- length(values) # number of elementes in "values"
results <- rep(NA, n)
# empty container vector for storing the results
## loop counter "i" will
## take vcalue 1, 2...n in that order
for (i in 1:n){
  ## store the result of multiplication as the
  ## ith element
  results[i] <- values[i] * 2
  cat(values[i], "times 2 is equal to",
      results[i],"\n")
}
```

```
## 2 times 2 is equal to 4
## 4 times 2 is equal to 8
## 6 times 2 is equal to 12
```

```
results
```

```
## [1]  4  8 12
```

### 4.1.2 R 中的一般条件语句

- `if(x){ expression}`
- `if(x){}else{}`
- `if(x){}`

  `else if(y){}`

  `else{}`

```
values <-  1:5
n <- length(values)
results <- rep(NA, n)
for (i in 1:n) {
  ## x and r get overwrittern in each iteration
  x <- values[i]
  r <- x %% 2 ## 计算 x 除于 2 有余数
  if(r == 0){
    cat(x, "is even and I will perform addition",
        x, "+", x, "\n")
    results[i] <- x * x
  }else{
    cat(x, "is odd and I will perform multiplication",
        x, "*", x, "\n")
    results[i] <- x * x
  }

}
```

```
## 1 is odd and I will perform multiplication 1 * 1
## 2 is even and I will perform addition 2 + 2
## 3 is odd and I will perform multiplication 3 * 3
## 4 is even and I will perform addition 4 + 4
## 5 is odd and I will perform multiplication 5 * 5
```

```
results
```

```
## [1]  1  4  9 16 25
```

### 4.1.3 基于民意调查的预测

```
## load election results by statue
pres08 <-  read.csv("Datasets/pres08.csv")
## load ppplling data
polls08 <- read.csv("Datasets/polls08.csv")
## compute Obama's margin
polls08$margin <-  polls08$Obama - polls08$McCain
pres08$margin <-  pres08$Obama - pres08$McCain
```

对于每个州，我们只使用最新一次民意调查结果对奥巴马的获胜幅度进行预测。也就是说，我们计算在大选最接近的一天里所有民意调查的预测平均值。

`as.Date()` 函数可以将数据转换为 Date 类别。

```
x <- as.Date("2008-11-04")
y <- as.Date("2008/9/1")
x - y # number of days between 2008/9/1 and 11/4
```

```
## Time difference of 64 days
```

```
## Time difference of 64 days
```

```
##convert to a Date object
polls08$middate <-  as.Date(polls08$middate)
## compute the number of days to Election Day
polls08$DaysToElection <- as.Date("2008-11-04") - polls08$middate
poll.pred <- rep(NA, 51)
## extract unique state names which the loop will
## iterate through
st.names <- unique(polls08$state)
## add state names as labels for easy interpretation
## later on
names(poll.pred) <- as.character(st.names)
## loop across 50 states olus DC
for(i in 1:51){
  ## subset the ith state
  state.data <- subset(polls08,
                       subset = (state == st.names[i]))
  ## further subset the latest polls within the state
  latest <- subset(state.data,
              DaysToElection == min(DaysToElection))
  ## compute the mean of latest polls and store it
  poll.pred[i] <- mean(latest$margin)
}
```

**预测误差**　**预测误差**被定义为：

$$ = - $$

平均预测误差被称为误差。当偏误为零时，预测结果被认为是无偏的。最后，**预测误差的均方根被称为均方根误差**，表示预测误差的平均大小。

均方根以 `sqrt(mean(error^2))` 进行计算。

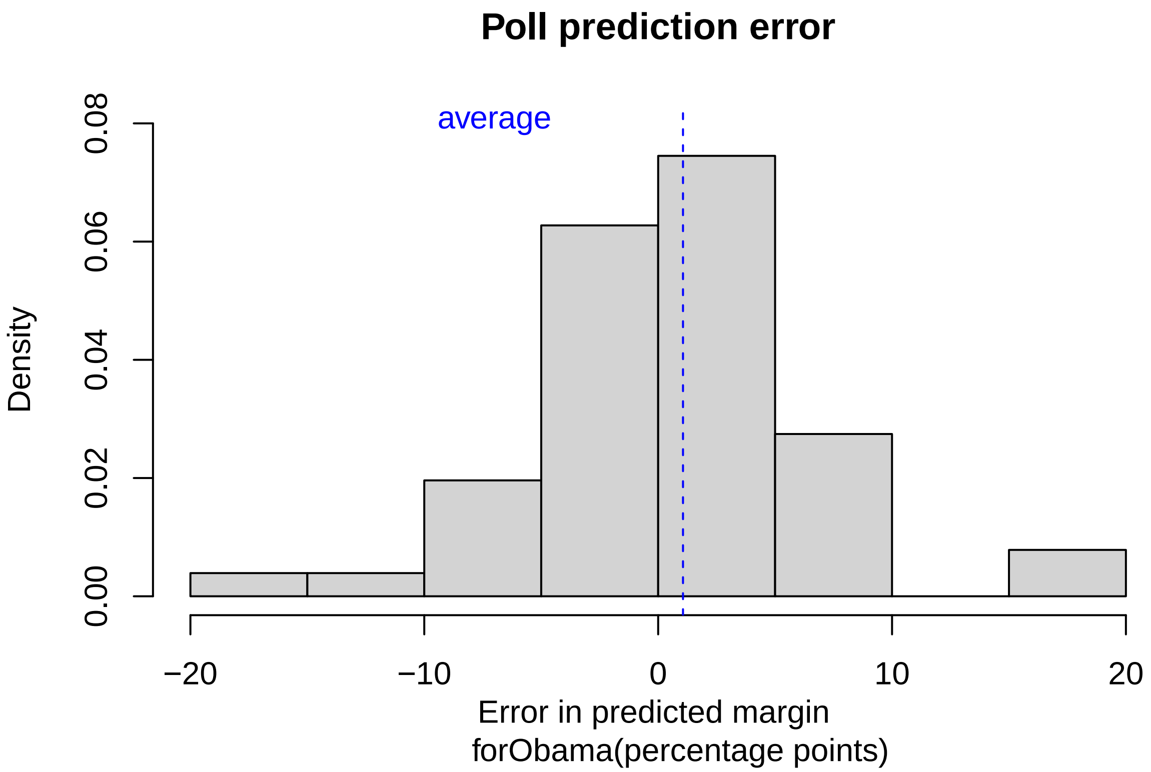
```
errors <-  pres08$margin - poll.pred
names(errors) <- st.names
mean(errors)
```

```
## [1] 1.062092
```

```
sqrt(mean(errors^2))
```

```
## [1] 5.90894
```

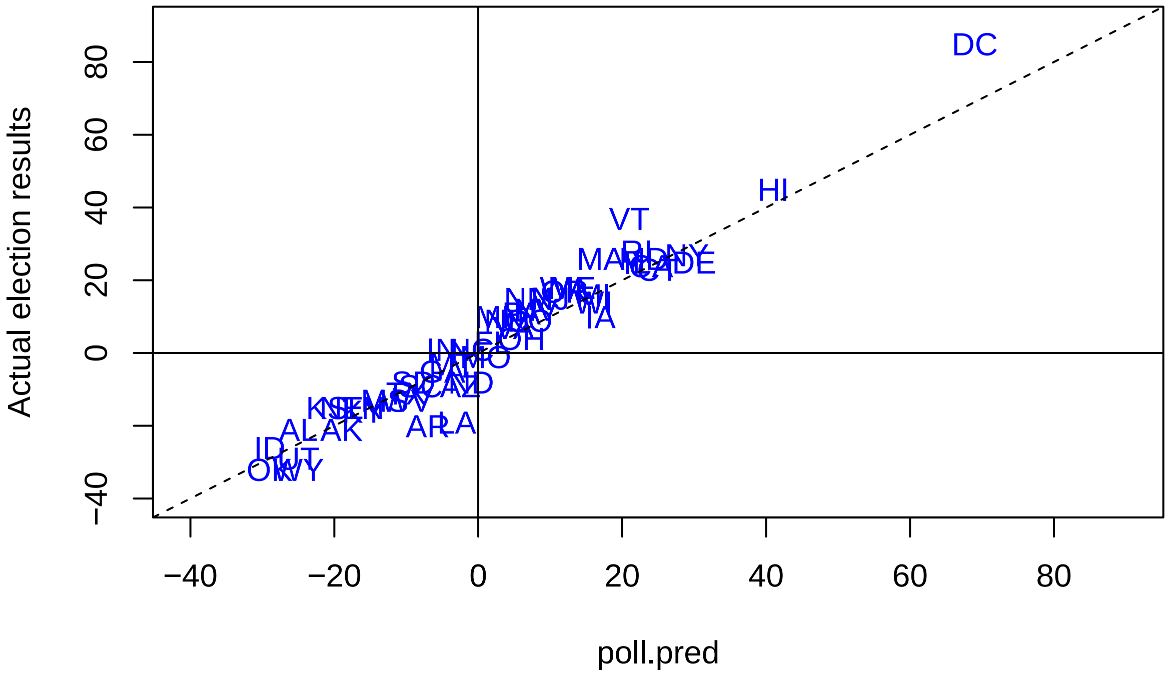
```
## histogram
hist(errors, freq = FALSE, ylim = c(0, 0.08),
     main = "Poll prediction error",
     xlab = "Error in predicted margin
     forObama(percentage points)")

## add mean
abline(v = mean(errors), lty = "dashed", col = "blue")
text(x  = -7, y = 0.08, "average", col = "blue")
```



画图的方式进一步观察每个州的投票预测准确度。预测结果画在横轴上，实际结果在纵轴上，用两个字母的名字缩写来表示每个州。

plot 的 type 设置为”n”，用 labels 去存储文本标签的字符向量。

```r
plot(poll.pred, pres08$margin, type = "n", main = "",
     xlim = c(-40, 90), ylim = c(-40, 90),
     ylab = "Actual election results")
## add state abbreviations
text(x = poll.pred, y = pres08$margin,
     labels = pres08$state, col = "blue")
## lines
abline(a = 0, b = 1, lty = "dashed")
abline(v = 0)
abline(h = 0)
```



**分类** sign() 函数来确实正负号

```
## which state polls called wrongs?
pres08$state[sign(poll.pred) != sign(pres08$margin)]
```

```
## [1] "IN" "MO" "NC"
```

```
## what was the actual margin foe these states?
pres08$margin[sign(poll.pred) != sign(pres08$margin)]
```

```
## [1]  1 -1  1
```

预测结果类别的问题被称为分类问题。这里的错误分类率为 3/51。

**在二元分类问题中**，有两种类型的错误分类。我们可能错误地预测奥巴马是某州的获胜者，也有可能错误地预测他是某州的失利者。

分类指的是预测分类结果的问题。分类仅可能是正确的或错误的。在二元分类问题中，有两种错误分类：假阳性和假阴性，分别代表错误预测出”正”或”负”的结果。

```
## 根据民调预测计算奥巴马的选举团选票数
sum(pres08$EV[pres08$margin > 0])
```

```
## [1] 364
```

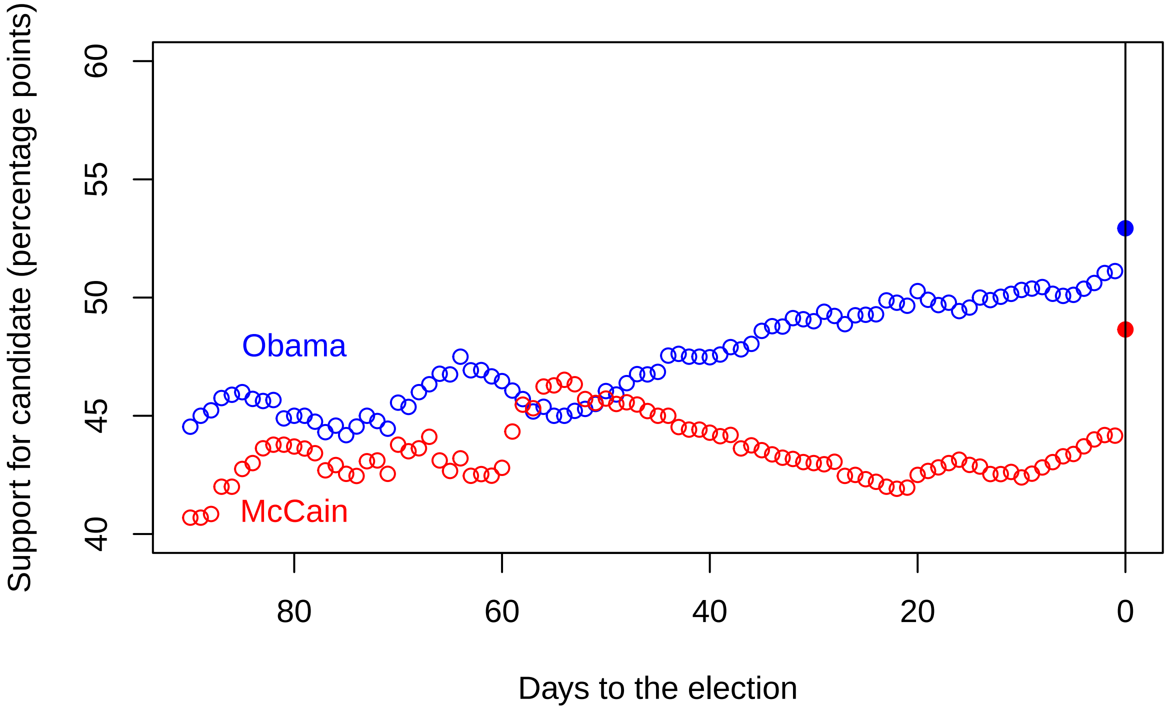
```
## poll population
sum(pres08$EV[poll.pred > 0])
```

```
## [1] 349
```

虽然全国总票数不能决定选举结果，但我们可以检验全国民意调查的准确性以及民意在竞选过程中的变化。

```
## load the data
pollsUS08 <- read.csv("Datasets/pollsUS08.csv")
## compute number of days to the election as before
pollsUS08$middate <- as.Date(pollsUS08$middate)
pollsUS08$DaysToElection <- as.Date("2008-11-04") - pollsUS08$middate
## empty vectors to score predictions
Obama.pred <- McCain.pred <- rep(NA, 90)
for(i in 1:90){
  ## take all polls conducted within the past 7 days
  week.data <- subset(pollsUS08,
                subset = ((DaysToElection <= (90 - i + 7))
                          & (DaysToElection > (90 - i))))
  ## compute support for each candidate using the acerage
  Obama.pred[i] <- mean(week.data$Obama)
  McCain.pred[i] <- mean(week.data$McCain)
}
```

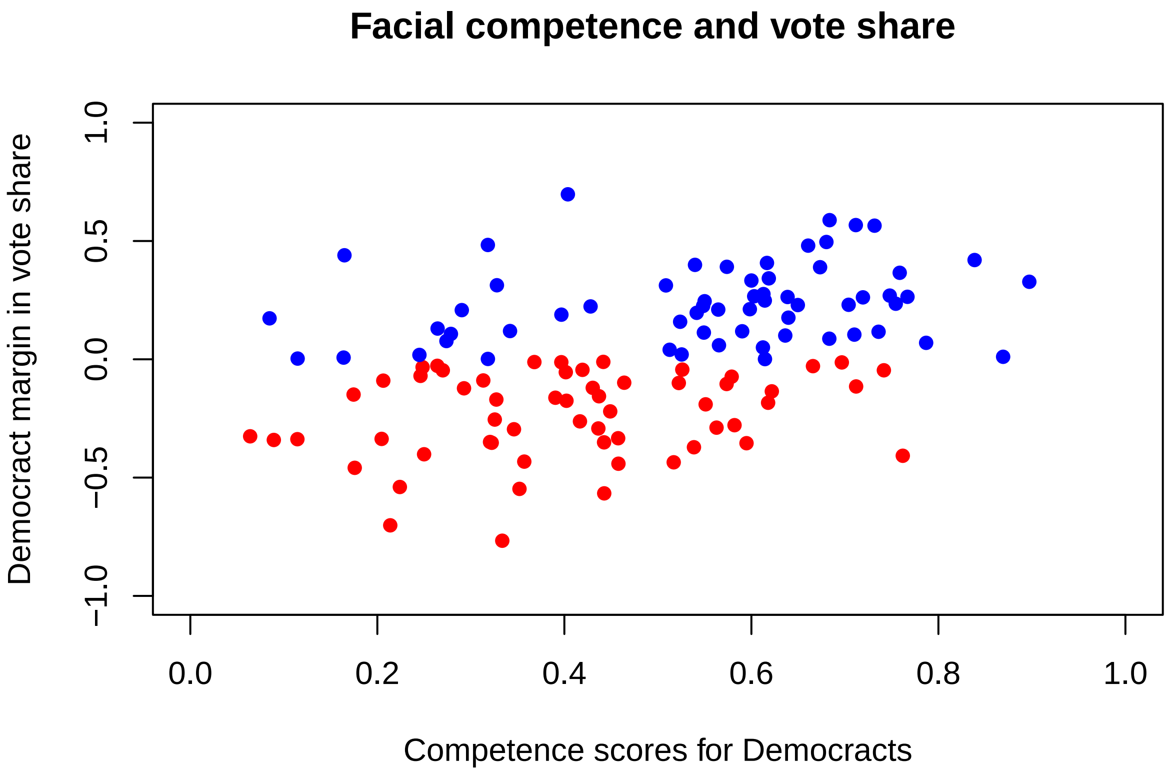
```
## 画图
plot(90:1, Obama.pred, type = "b", xlim = c(90, 0),
     ylim = c(40, 60), col = "blue",
     xlab = "Days to the election",
     ylab = "Support for candidate (percentage points)")
lines(90:1, McCain.pred, type = "b", col = "red")
## actual election results:pch = 19 gives solid circles
points(0, 52.93, pch = 19, col = "blue")
points(0, 48.65, pch = 19, col = "red")
## line indicating Election Day
abline(v = 0)
##labeling candidates
text(80, 48, "Obama", col = "blue")
text(80, 41, "McCain", col = "red")
```



## 4.2 线性回归

### 4.2.1 面部长相与选举结果的联系

```
## load the data
face <- read.csv("Datasets/face.csv")
## two-party vote share for Democracts and Republicans
face$d.share <- face$d.votes / (face$d.votes +face$r.votes)
face$r.share <- face$r.votes / (face$d.votes + face$r.votes)
face$diff.share <- face$d.share - face$r.share
## 画图
plot(face$d.comp, face$diff.share, pch = 16,
     col = ifelse(face$w.party == "R", "red", "blue"),
     xlim = c(0, 1), ylim = c(-1, 1),
     xlab = "Competence scores for Democracts",
     ylab = "Democract margin in vote share",
     main = "Facial competence and vote share ")
```



### 4.2.2 相关性与散点图

```
cor(face$d.comp, face$diff.share)
```

```
## [1] 0.4327743
```

受访者眼中候选人的能力与他/她在选举日真实的获胜幅度之间存在一定的正相关性。

没有相关性并不意味着两个变量没有关系，说不定是存在非线性的关系。

**相关系数**量化了两个变量之间的线性关系。散点图中数据云的上升趋势意味着正相关，而数据云中的向下趋势表示负相关。相关性往往不合适表示非线性关系。

### 4.2.3 最小二乘法

两个变量的线性关系模型：

$Y = \alpha + \beta X + \varepsilon$

是截距， 是斜率， 是误差项。

在 R 中使用 `lm(Y ~ X)` 函数来拟合线性回归模型。

参数：

- Y：结果变量
- X：预测值

使用 `coef()` 可以直接获取估计系数，`fitted()` 可以获取预测（拟合）值。

`resid()` 函数获取残差，`sqrt(mean(x^2))` 获取均方根误差 (RMSE)

我们使用民主党的胜选票数占比作为结果变量，民主党候选人在受访者眼中的能力作为预测因子。

```
fit <- lm(diff.share ~ d.comp, data = face )
fit
```

```
##
## Call:
## lm(formula = diff.share ~ d.comp, data = face)
##
## Coefficients:
## (Intercept)       d.comp
##     -0.3122       0.6604
```

说明预测的斜率为 0.6604，截距为-0.3122,。

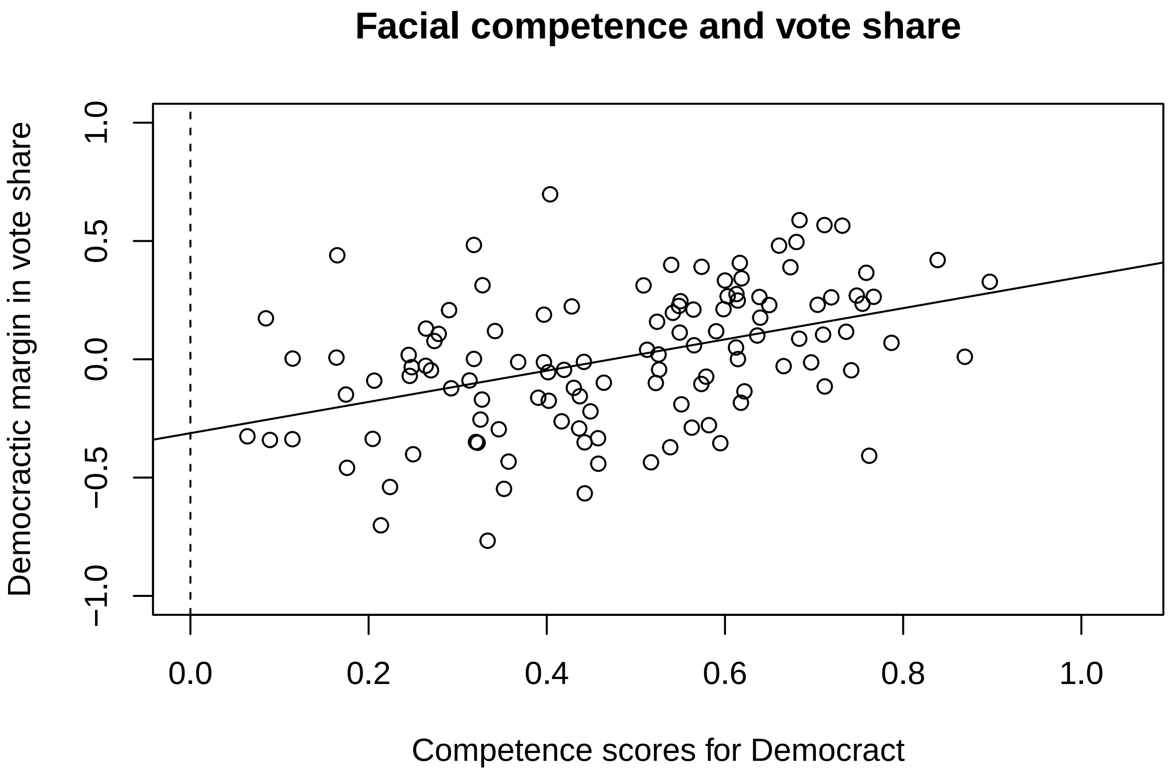
```
coef(fit)
```

```
## (Intercept)      d.comp
##  -0.3122259   0.6603815
```

```
head(fitted(fit))
```

```
##           1           2           3           4           5           6
##  0.06060411 -0.08643340  0.09217061  0.04539236  0.13698690 -0.10057206
```

```
plot(face$d.comp, face$diff.share, xlim = c(0, 1.05),
     ylim = c(-1, 1), xlab = "Competence scores for Democract",
     ylab = "Democractic margin in vote share",
     main = "Facial competence and vote share")
abline(fit)
abline(v = 0, lty = "dashed")
```



```r
epsilon.hat <- resid(fit)
sqrt(mean(epsilon.hat^2))
```

```
## [1] 0.2642361
```

### 4.2.4 趋中回归

趋中回归代表了一种经验现象，其中有着远离分布均值的预测因子的观测值倾向于更具有接近平均数的结果变量。这种趋势可以凭偶然来解释。

### 4.2.5 R 中的合并数据

`merge()` 函数可以在 R 中合并两个数据集。

参数：

- x 和 y：需要合并的数据集
- by：合并的变量名称

合并变量必须都存在于两个数据集中,通常情况下是相同的名称,如果不是同名,可以采用 by.x = "",by.y = ""。

cbind() 函数也可以合并数据，合并多个数据框的列合并。rbind() 函数可以合并多个数据框的行。但是 cbind() 函数会包含相同的列，即使它们包含相同的数据。而且 cbind() 不会自动排序匹配，merge() 是会的。

```
pres12 <- read.csv("Datasets/pres12.csv")
head(pres08)
```

```
##   state.name state Obama McCain EV margin
## 1    Alabama    AL    39     60  9    -21
## 2     Alaska    AK    38     59  3    -21
## 3    Arizona    AZ    45     54 10     -9
## 4   Arkansas    AR    39     59  6    -20
## 5 California    CA    61     37 55     24
## 6   Colorado    CO    54     45  9      9
```

```
head(pres12)
```

```
##   state Obama Romney EV
## 1    AL    38     61  9
## 2    AK    41     55  3
## 3    AZ    45     54 11
## 4    AR    37     61  6
## 5    CA    60     37 55
## 6    CO    51     46  9
```

```
## merge two data frames
pres <- merge(pres08, pres12, by = "state")
summary(pres)
```

```
##     state            state.name           Obama.x          McCain
##  Length:51          Length:51          Min.   :33.00   Min.   : 7.00
##  Class :character   Class :character   1st Qu.:43.00   1st Qu.:40.00
##  Mode  :character   Mode  :character   Median :51.00   Median :47.00
##                                        Mean   :51.37   Mean   :47.06
##                                        3rd Qu.:57.50   3rd Qu.:56.00
##                                        Max.   :92.00   Max.   :66.00
##       EV.x           margin          Obama.y          Romney
##  Min.   : 3.00   Min.   :-32.000   Min.   :25.00   Min.   : 7.00
##  1st Qu.: 4.50   1st Qu.:-13.000   1st Qu.:40.50   1st Qu.:41.00
##  Median : 8.00   Median :  4.000   Median :51.00   Median :48.00
##  Mean   :10.55   Mean   :  4.314   Mean   :49.06   Mean   :49.04
##  3rd Qu.:11.50   3rd Qu.: 17.500   3rd Qu.:56.00   3rd Qu.:58.00
##  Max.   :55.00   Max.   : 85.000   Max.   :91.00   Max.   :73.00
##       EV.y
##  Min.   : 3.00
##  1st Qu.: 4.50
##  Median : 8.00
##  Mean   :10.55
##  3rd Qu.:11.50
##  Max.   :55.00
```

标准化采用 scale() 函数。计算 Z 分数。当我们标准化结果变量和预测因子时，两个样本的均值都为零，估计的截距变为零。我们可以在公式中加入 - 1，在没有截距的情况下拟合模型。

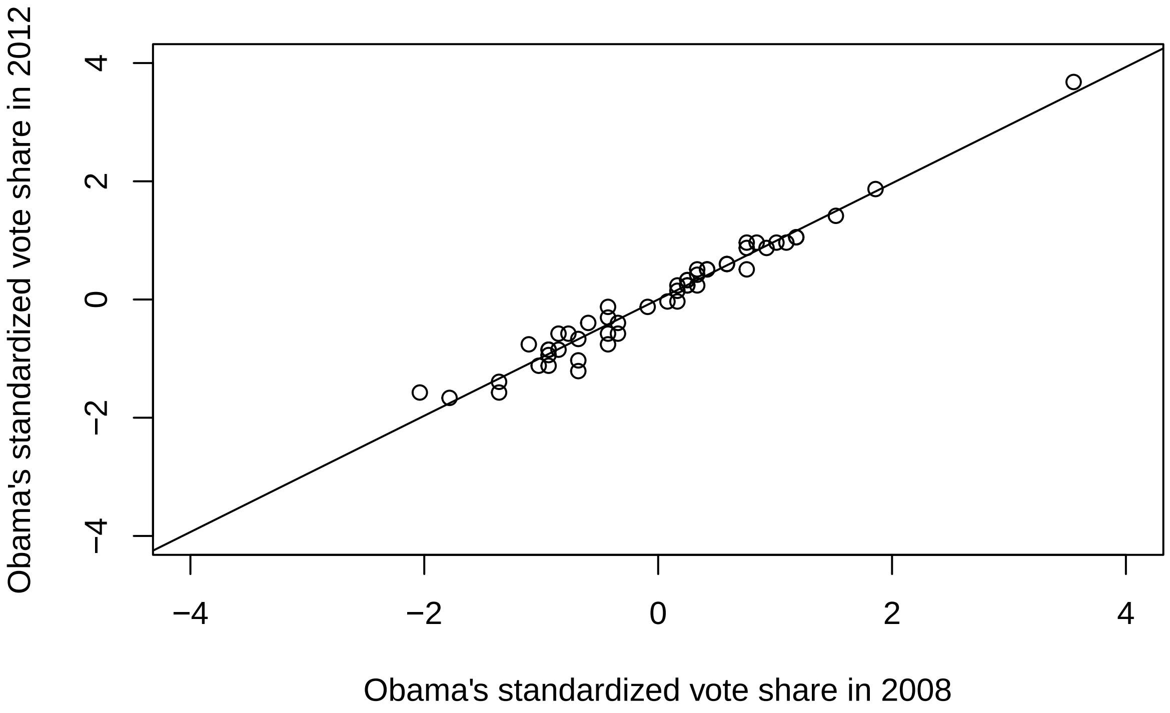
```
pres$Obama2012.z <- scale(pres$Obama.x)
pres$Obama2008.z <- scale(pres$Obama.y)
fit1 <- lm(Obama2012.z ~ Obama2008.z, data = pres)
fit1
```

```
##
## Call:
## lm(formula = Obama2012.z ~ Obama2008.z, data = pres)
##
## Coefficients:
## (Intercept)  Obama2008.z
##   2.646e-17    9.834e-01
```

```
fit1 <- lm(formula = Obama2012.z ~ -1 + Obama2008.z, data = pres)
fit1
```

```
##
## Call:
## lm(formula = Obama2012.z ~ -1 + Obama2008.z, data = pres)
##
## Coefficients:
## Obama2008.z
##      0.9834
```

```
plot(pres$Obama2008.z, pres$Obama2012.z, xlim = c(-4,4),
     ylim = c(-4, 4),
     ylab = "Obama's standardized vote share in 2012",
     xlab = "Obama's standardized vote share in 2008")
abline(fit1)
```



```
## bottom quantile
mean((pres$Obama2012.z > pres$Obama2008.z)[pres$Obama2008.z <= quantile(pres$Obama2008.z, 0.25)])
```

```
## [1] 0.6153846
```

```
## top quantile
mean((pres$Obama2012.z > pres$Obama2008.z)[pres$Obama2008.z >= quantile(pres$Obama2008.z, 0.75)])
```

```
## [1] 0.5
```

### 4.2.6 模型拟合

**R 方** R 方度量模型和数据的拟合程度，也是模型预测观测值的准确程度。**R 方的范围从 0 到 1**。

决定系数是模型拟合的度量，并且用预测值解释的结果变量的变异的比例来表示。它被定义为 1 减去残差平方和（SSR）与总平方和（TSS）之比。

```r
florida <- read.csv("Datasets/florida.csv")
fit2 <- lm(Buchanan00 ~ Perot96, data = florida)
fit2
```

```
##
## Call:
## lm(formula = Buchanan00 ~ Perot96, data = florida)
##
## Coefficients:
## (Intercept)      Perot96
##     1.34575      0.03592
```

```r
TTS2 <- sum((florida$Buchanan00 - mean(florida$Buchanan00))^2)
SSR2 <- sum(resid(fit2)^2)

(TTS2 - SSR2) / TTS2
```

```
## [1] 0.5130333
```

```r
## 定义一个计算 R 方的函数
R2 <- function(fit){
  resid <- resid(fit)
  y <- fitted(fit) + resid
  TTS <- sum((y - mean(y)) ^2)
  SSR <- sum(resid^2)
  R2 <- (TTS - SSR) / TTS
  return(R2)
}
R2(fit2)
```

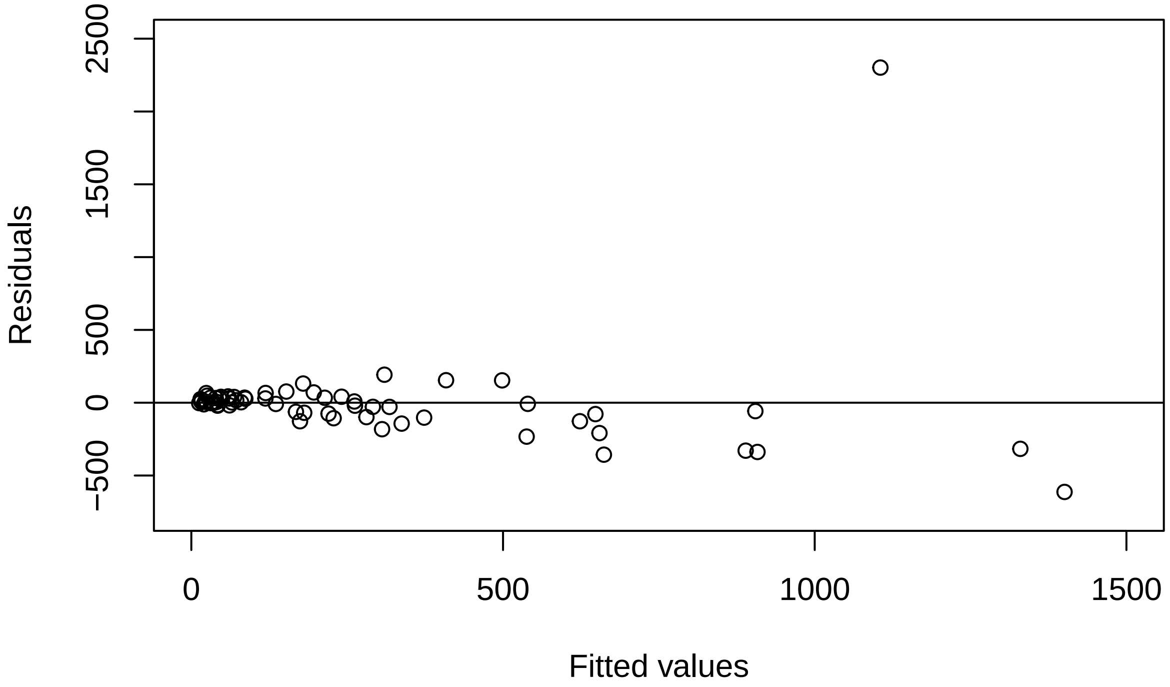
```
## [1] 0.5130333
```

```r
R2(fit1)
```

```
## [1] 0.9671579
## summary() 函数也可以计算 R 方
summary(fit2)$r.squared
```

```
## [1] 0.5130333
```

```
plot(fitted(fit2), resid(fit2), xlim = c(0, 1500),
     ylim = c(-750, 2500), xlab = "Fitted values",
     ylab = "Residuals")
abline(h = 0)
```



残差图

发现存在一个极大的残差或异常值，如果去掉之后观察一下是否改善了模型拟合度。

```
florida$county[resid(fit2) == max(resid(fit2))]
```

```
## [1] "PalmBeach"
```

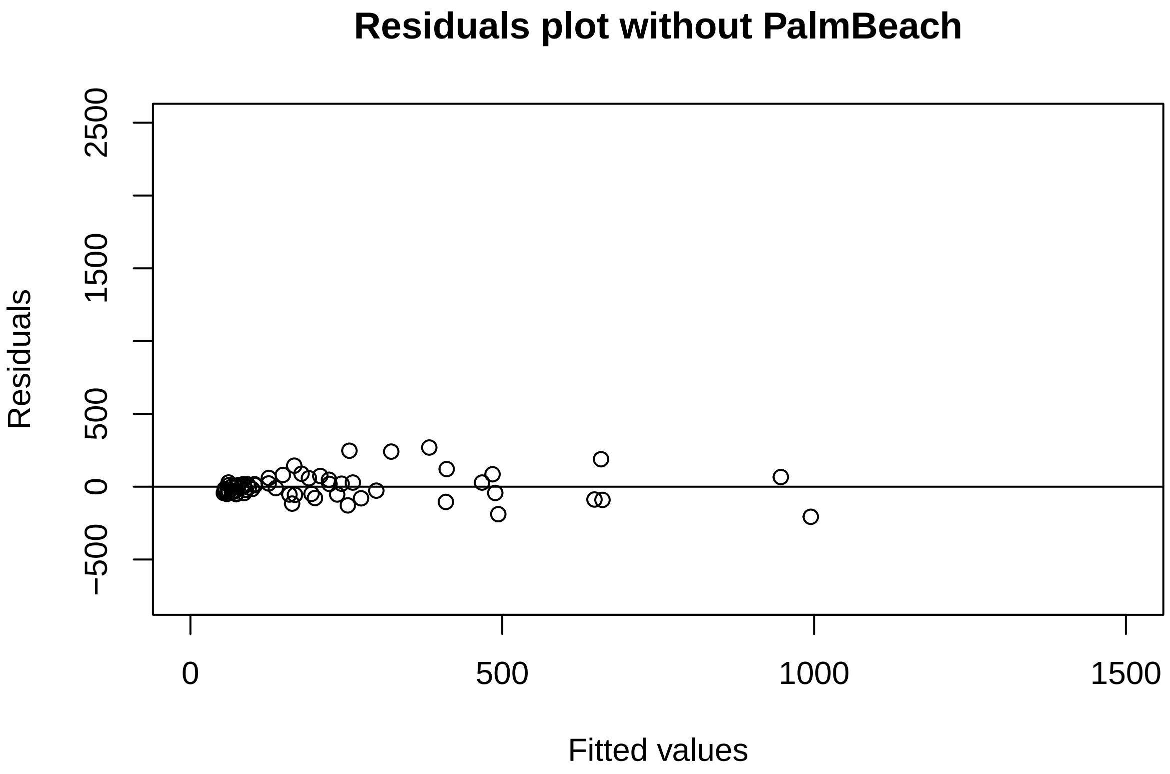
```
florida.pb <- subset(florida, subset = (county != "PalmBeach"))
fit3 <- lm(Buchanan00 ~ Perot96, data = florida.pb)
fit3
```

```
##
## Call:
## lm(formula = Buchanan00 ~ Perot96, data = florida.pb)
##
## Coefficients:
## (Intercept)      Perot96
##    45.84193      0.02435
```

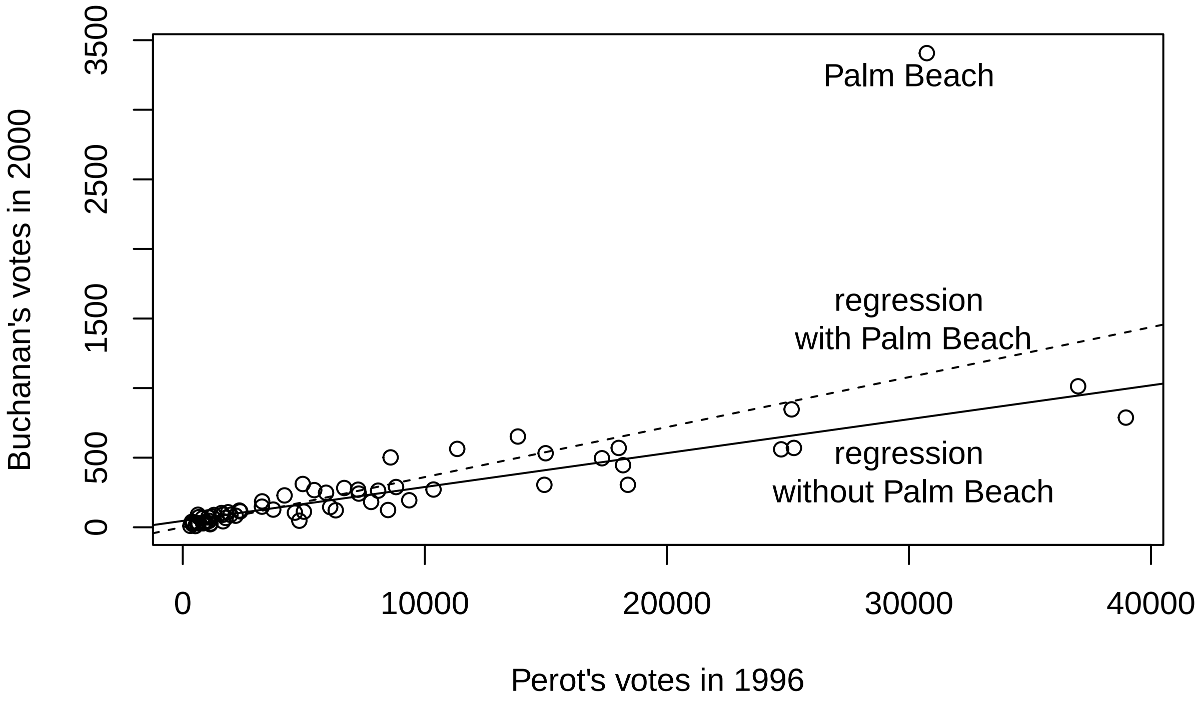
```
R2(fit3)
```

```
## [1] 0.8511675
```

```
plot(fitted(fit3), resid(fit3), xlim = c(0, 1500),
     ylim = c(-750, 2500), xlab = "Fitted values",
     ylab = "Residuals", main = "Residuals plot without PalmBeach")
abline(h = 0)
```



```
plot(florida$Perot96, florida$Buchanan00,
     xlab = "Perot's votes in 1996",
     ylab = "Buchanan's votes in 2000")
abline(fit2, lty = "dashed")
abline(fit3)
text(30000, 3250, "Palm Beach")
text(30000, 1500, "regression\n with Palm Beach")
text(30000, 400, "regression\n without Palm Beach")
```



注意如果对特定样本进行过度调整，称为过度拟合，则该模型可能会在另一个样本中预测得到不太准确。所以我们要避免模型过度拟合到特定样本。

## 4.3 回归与因果关系

因果推断需要反事实结果。对于接受干预的个体，我们希望预测未经干预时的结果变量的值。在某些假设下，回归模型可用于预测反事实结果。

### 4.3.1 随机化实验

某个研究主要观测政府中女政治家对政策结果产生的因果影响。

Q：女性是否会推行与男性不同的政策？

为了克服混是否因为政治家淆性别还是意识形态影响政策差异，利用印度的随机政策实验。

首先我们需要观察在西孟加拉邦中每个村落（领导席位女性）保留政策是否得到妥善的实施。

```
women <- read.csv("Datasets/women.csv")
mean(women$female[women$reserved == 1])
```

```
## [1] 1
```

```r
mean(women$female[women$reserved == 0])
```

```
## [1] 0.07476636
```

发现好像保留政策得到了实施。

继续观察在女性更关注的饮用水质量和男性更关注的灌溉问题上，这些保留政策和没有保留政策的村庄之间饮用水设备和灌溉系统的平均数量差异。

```r
mean(women$water[women$reserved == 1]) -
  mean(women$water[women$reserved == 0])
```

```
## [1] 9.252423
```

```r
mean(women$irrigation[women$reserved == 1]) -
  mean(women$irrigation[women$reserved == 0])
```

```
## [1] -0.3693319
```

当预测变量是二元变量的时候， **可以被解释为估计的平均干预效应**, 并且在数值上**斜率系数  等于相应的均差估计值**。另一方面，估计的截距等于控制条件下的结果变量的平均值。**干预分配的随机化**，允许将在线性回归会模型下定义的相关关系阐释为因果关系。

```r
lm(water ~ reserved, data = women)
```

```
## 
## Call:
## lm(formula = water ~ reserved, data = women)
## 
## Coefficients:
## (Intercept)     reserved  
##      14.738        9.252
```

```r
lm(irrigation ~ reserved, data = women)
```

```
## 
## Call:
## lm(formula = irrigation ~ reserved, data = women)
```

```
##
## Coefficients:
## (Intercept)     reserved
##      3.3879      -0.3693
```

### 4.3.2 多元预测回归

线性回归模型只包含一个预测因子。但是回归模型可以不止有一个预测因子。

$Y = \alpha + \beta 1X1 + \beta 2X2 + \beta 3X3 + \beta 4X4 + ... + \beta pXp + \varepsilon$

其中包含截距、预测因子、系数、误差项、预测因子的数量

最小二乘法是拟合线性的回归模型，但是对于其他的非线性不合适。

lm() 函数会自动创建一组指标或虚拟变量，这些指标变量将用于计算，但不会保存在数据框中。

predict() 函数获得预测的平均结果。可以从 lm() 函数获取输出并计算预测值。predict() 函数与 fitted() 函数不同，可以将新数据框作为 newdate 参数，并对此数据框中的每个观测值进行预测。

新数据框的变量必须**与拟合线性模型的预测因子相匹配**，虽然他们可以具有不同的值。

```
social <- read.csv("Datasets/social.csv")
levels(social$messages)
```

```
## NULL
```

```
fit <- lm(primary2006 ~ messages, data = social)
fit
```

```
##
## Call:
## lm(formula = primary2006 ~ messages, data = social)
##
## Coefficients:
##       (Intercept)    messagesControl  messagesHawthorne  messagesNeighbors
##          0.314538          -0.017899           0.007837           0.063411
```

```
## create indicator variables
social$Control <-  ifelse(social$messages == "Control", 1, 0)
social$Hawthorne <- ifelse(social$messages == "Hawthorne", 1, 0)
social$Neighbors <- ifelse(social$messages == "Neighbors", 1, 0)
lm(primary2006 ~ Control + Hawthorne + Neighbors, data = social)
```

```
##
## Call:
## lm(formula = primary2006 ~ Control + Hawthorne + Neighbors, data = social)
##
## Coefficients:
## (Intercept)      Control    Hawthorne    Neighbors
##    0.314538    -0.017899     0.007837     0.063411
```

```
## create a data frame with unique values of "meassages"
unique.message <- data.frame(messages = unique(social$messages))
unique.message
```

```
##     messages
## 1 Civic Duty
## 2  Hawthorne
## 3    Control
## 4  Neighbors
```

```
predict(fit, newdata = unique.message)
```

```
##         1         2         3         4
## 0.3145377 0.3223746 0.2966383 0.3779482
```

```
## sample average
tapply(social$primary2006, social$messages, mean)
```

```
## Civic Duty    Control  Hawthorne  Neighbors
##  0.3145377  0.2966383  0.3223746  0.3779482
```

为了使线性回归的输出更易于解释，我们可以删除截距并使用所有四个指标变量。这种替代性设计使我们能够直接获得每个组内的平均结果作为相应指标变量的系数。

```
## linear regression without intercept
fit.noint <- lm(primary2006 ~ -1 + messages, data = social)
fit.noint
```

```
##
## Call:
## lm(formula = primary2006 ~ -1 + messages, data = social)
```

```
##
## Coefficients:
## messagesCivic Duty     messagesControl   messagesHawthorne   messagesNeighbors
##             0.3145              0.2966              0.3224              0.3779
```

通过计算干预效应的系数减去控制组的系数来估计相对于控制条件的平均干预效应。

对照组是该模型下的基准组，没有截距。

无论是使用没有截距的模型还是原始模型，任何两组之间估计的因果效应的差异等于相应系数之间的差异。

**平均因果效应估计的计算方法：**

- 因子干预变量的线性回归
- 均值差估计

```
##estimate average effect of "Neighbors" condition
coef(fit)["messagesNeighbors"] - coef(fit)["messagesControl"]
```

```
## messagesNeighbors
##        0.08130991
```

```
## difference-in-means
mean(social$primary2006[social$messages == "Neighbors"]) -
  mean(social$primary2006[social$messages == "Control"])
```

```
## [1] 0.08130991
```

**调整 R 方**　通过调整自由度来调整 R 方。自由度 = n - p -1。

$$R \ = 1 - ((SSR/(n-p-1))/(TTS/(n-1)))$$

```
## adjusted R-squared
adjR2 <- function(fit){
  resid <- resid(fit)
  y <- fitted(fit) + resid
  n <- length(y)
  TTS.adj <- sum((y - mean(y))^2) / (n - 1)
  SSR.adj <- sum(resid^2) / (n - length(coef(fit)))
  R2.adj <- 1 - SSR.adj / TTS.adj
  return(R2.adj)
}
adjR2(fit)
```

```
## [1] 0.003272788
```

```
R2(fit) ## 原来的 R 方
```

```
## [1] 0.003282564
```

```
fitsummary <- summary(fit)
fitsummary$adj.r.squared
```

```
## [1] 0.003272788
```

```
fitsummary$r.squared # 原来的 R 方
```

```
## [1] 0.003282564
```

### 4.3.3 异质干预效应

一个有着交互项的线性回归模型的实例是：

$Y = \alpha + \beta 1X1 + \beta 2X2 + \beta 3X1X2 + \varepsilon$

```
social.neighbor <- subset(social, (messages == "Control") |
                              (messages == "Neighbors"))

## standard way to generate main and interaction effects
fit.int <- lm(primary2006 ~ primary2004 + messages + primary2004:messages,
              data = social.neighbor)
fit.int
```

```
##
## Call:
## lm(formula = primary2006 ~ primary2004 + messages + primary2004:messages,
##     data = social.neighbor)
##
## Coefficients:
##                  (Intercept)                   primary2004
##                      0.23711                       0.14870
##            messagesNeighbors  primary2004:messagesNeighbors
##                      0.06930                       0.02723
```

### 4.3.4 断点回归设计

## 4.4 总结

## 4.5 练习

Exercise1：基于博彩市场的预测

Exercise2：墨西哥的选举和条件现金转移计划

Exercise3：巴西政府转移和减少贫困率