Capitulo 9

Modelos de probabilidad contra datos

Como ejemplo específico, si tenemos una secuencia de observaciones de datos numéricos, podemos calcular fácilmente los valores mínimos y máximos observados, pero no podemos decir qué tan probable sería observar, a partir de un nuevo lote de datos recopilados en condiciones similares, valores menor que el mínimo observado o mayor que el máximo observado.

Si ajustamos una distribución de probabilidad a nuestros datos, una distribución gaussiana basada en la media y la desviación estándar calculada a partir de los datos, podemos responder a estas preguntas, asumiendo que nuestro modelo de probabilidad es adecuado.

Como ejemplo específico, recuerde de la discusión en el Capítulo 3 que los valores de peso del pollito de 18 semanas en el marco de datos ChickWeight del paquete de conjuntos de datos R parecían estar bien aproximados por una distribución gaussiana. La desviación mínima, máxima, media y estándar se calculan fácilmente a partir de esta secuencia de datos

```
wts18 <- ChickWeight$weight[which(ChickWeight$Time == 18)]
min(wts18)
```

```
## [1] 72
```

```
max(wts18)
```

```
## [1] 332
```

```
mean(wts18)
```

```
## [1] 190.1915
```

```
sd(wts18)
```

```
## [1] 57.39476
```

Distribuciones cuantiles

La base para esta caracterización gráfica de los datos son cinco cuantiles de los datos: el mínimo de la muestra, el cuartil inferior, la mediana, el cuartil superior y el máximo de la muestra. Estos valores se pueden calcular a partir de cualquier secuencia de datos numéricos en R utilizando la función cuantil especificando solo el vector de datos.

Por lo tanto, para los datos de peso de pollito de 18 semanas del ejemplo anterior, estos valores son:

```
wts18 <- ChickWeight$weight[which(ChickWeight$Time == 18)]
quantile(wts18)
```

```
##    0%   25%   50%   75%  100%
##  72.0 152.5 187.0 230.5 332.0
```

```
quantile(wts18, probs = seq(0, 1, 0.1))
```

```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
##  72.0 116.8 146.4 159.4 178.0 187.0 203.6 216.6 233.6 261.4 332.0
```

Confidencia de Intervalos y Significancia

La función cuantil Q (x) es esencialmente la inversa de P (z), que nos dice qué El valor de z satisface la condición P (z) = x: Q (x) = {z tal que P (z) = x}

```
minWt <- min(wts18)
maxWt <- max(wts18)
meanWt <- mean(wts18)
sdWt <- sd(wts18)
pMin <- pnorm(q = minWt, mean = meanWt, sd = sdWt)
pMin
```

```
## [1] 0.01973403
```

```
qnorm(mean = meanWt, sd = sdWt, p = seq(0, 1, 0.1))
```

```
##  [1]     -Inf 116.6371 141.8868 160.0936 175.6507 190.1915 204.7323
##  [8] 220.2893 238.4961 263.7458      Inf
```

R admite muchas distribuciones diferentes, varias de las cuales se analizan brevemente más adelante en este capítulo, y para la mayoría de estas distribuciones, este soporte incluye las siguientes cuatro funciones: 1. una función de densidad como dnorm que calcula los valores de la densidad de probabilidad p(x) para un rango de valores de datos x, dados los parámetros de distribución requeridos (por ejemplo, la media y la desviación estándar para los datos gaussianos).

2. una función de distribución acumulativa como pnorm que calcula la probabilidad acumulativa P (x) para esta distribución, dados los mismos parámetros.
3. una función cuantil como qnorm que calcula los cuantiles de la distribución para un conjunto dado de probabilidades y los parámetros de distribución.
4. un generador de números aleatorios como rnorm que genera muestras aleatorias de la distribución, dada la cantidad de muestras deseadas y los parámetros de distribución.

Intervalos Confidenciales

```
qnorm(p = 0.025, mean = 0, sd = 1)
```

```
## [1] -1.959964
```

```
qnorm(p = 0.975, mean = 0, sd = 1)
```

```
## [1] 1.959964
```

Significado estadistico y valores P

El concepto de significado estadístico está estrechamente relacionado con los intervalos de confianza, y los valores p son la medida numérica estándar de significado estadístico, quizás la forma más fácil de introducir el concepto de significado estadístico es mirar un ejemplo común donde surge.

```
library(MASS)
whiteModel <- lm(Gas ~ Temp * Insul, data = whiteside)
summary(whiteModel)
```

```
##
## Call:
## lm(formula = Gas ~ Temp * Insul, data = whiteside)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97802 -0.18011  0.03757  0.20930  0.63803
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.85383    0.13596  50.409  < 2e-16 ***
```

```
## Temp            -0.39324    0.02249 -17.487  < 2e-16 ***
## InsulAfter      -2.12998    0.18009 -11.827 2.32e-16 ***
## Temp:InsulAfter  0.11530    0.03211   3.591 0.000731 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.323 on 52 degrees of freedom
## Multiple R-squared:  0.9277, Adjusted R-squared:  0.9235
## F-statistic: 222.3 on 3 and 52 DF,  p-value: < 2.2e-16
```

Los resultados numéricos en la sección "Coeficientes" de este resumen incluyen los valores de los coeficientes mismos (la columna denominada "Estimación"), el error estándar, que es una estimación de la desviación estándar del valor del coeficiente, el valor t asociado, análogo a lo definido en la ecuación. Para la media, y un valor p discutido en detalle a continuación (la columna etiquetada "P r (> | t |)").

Caracterizacion de variables binarias

Las variables binarias son aquellas que solo pueden asumir uno de dos valores, como la variable Insul en el marco de datos en el lado blanco del paquete MASS que se discutió muchas veces en capítulos anteriores de este libro. Como se señaló en el Capítulo 5, incluso si estas variables son categóricas, como la variable Insul con los valores "Antes" y "Después", es muy sencillo volver a expresarlas como variables numéricas equivalentes con valores 0 y

1. Esto significa que las variables binarias son en cierto sentido "intermedias" entre las variables categóricas y las variables numéricas (discretas) como los datos de recuento de valores enteros considerados. Por lo tanto, las variables binarias son susceptibles de caracterización por ambas técnicas, como las tablas de contingencia

Proporciones impares

Un caso particularmente importante sería valores faltantes que se asociaron fuertemente con la variable de diabetes. El odds ratio y su intervalo de confianza nos permiten examinar esta pregunta. La base para este examen es la siguiente función R, que calcula d y su intervalo de confianza en el nivel cLevel: el estimador de odds ratio. ORproc <- function(tbl, cLevel = 0.95){ # n11 <- tbl[1,1] n12 <- tbl[1,2] n21 <- tbl[2,1] n22 <- tbl[2,2] # OR <- (n11/n12) * (n22/n21) # sigmaLog <- sqrt(1/n11 + 1/n12 + 1/n21 + 1/n22) alpha <- 1 - cLevel zalpha2 <- qnorm(1 - alpha/2) logOR <- log(OR) logLo <- logOR - zalpha2 * sigmaLog logHi <- logOR + zalpha2 * sigmaLog loCI <- exp(logLo) upCI <- exp(logHi) # outFrame <- data.frame(OR = OR, confLevel = cLevel, loCI = loCI, upCI = upCI) return(outFrame) }

```
#zeroInsulin <- as.numeric(PimaIndiansDiabetes$insulin == 0)
#diabetic <- as.numeric(PimaIndiansDiabetes$diabetes == "pos")
#cTable <- table(zeroInsulin, diabetic)
#cTable
```

Caracterizacion de datos contables

La distribucion de Poison DE la misma manera que la distribución gaussiana es la aproximación más popular aplicada a los datos de valor continuo, la distribución de Poisson es la distribución más utilizada para aproximar los datos de conteo. Esta distribución asigna una probabilidad P k a cada valor entero i = 0, 1, 2,.

Donde = 1 · 2 · · · i para i >= 1 y 0! = 1 por convención. Aquí, una constante positiva que representa el único parámetro que define la distribución de Poisson y determina todas las características de distribución. De hecho, tanto la media como la varianza de la distribución de Limitaciones de la distribucion Gausiana

El supuesto de distribución gaussiano es insostenible, incluida la siguientes tipos de datos numéricos: 1. distribuciones de datos fuertemente asimétricas como las que se ven a menudo para las variables eso solo puede asumir valores positivos; 2. variables que están restringidas a un rango acotado (por ejemplo, probabilidades o fracciones que deben estar entre 0 y 1); 3. variables que exhiben "comportamiento de cola pesada" como el ruido del destello 4. distribuciones de datos multimodales como los datos de erupción del géiser Old Faithful

Algunas Alternativas

Limitaciones gaussianas descritas allí: 1. la distribución gamma, una distribución asimétrica apropiada para datos numéricos positivos; 2. la distribución beta, una distribución extremadamente flexible apropiada para datos numéricos acotados; 3. La distribución t de Student, una familia de distribuciones de datos simétricas.

Productos momentos correlacionales

El coeficiente de correlación producto-momento fue desarrollado por Karl Pearson a fines del siglo XIX, y por esa razón a veces se le llama coeficiente de correlación de Pearson. Esta medida sigue siendo extremadamente popular más de un siglo después porque tiene una serie de características extremadamente útiles, incluidas las tres siguientes:

1. facilidad de cálculo: esta medida se programa fácilmente en cualquier lenguaje que admita operaciones aritméticas básicas y está disponible como una función incorporada en cualquier entorno de software que admita cálculos estadísticos básicos, incluidos Microsoft Excel, R, Python y muchos otros.
2. facilidad de interpretación: el coeficiente de correlación producto-momento es una medida de asociación lineal, que cuantifica la tendencia de una variable a variar linealmente.
3. Completitud para datos Gaussianos conjuntos: si dos variables, x e y, tienen una distribución Gaussiana conjunta (un concepto discutido brevemente a continuación), se caracterizan por completo por sus medias individuales y desviaciones estándar y su coeficiente de correlación momento-producto.

```
library(MASS)

cor(whiteside$Temp, whiteside$Gas)
```

```
## [1] -0.6832545
```

```
whitesideLinearModel <- lm(Gas ~ Temp, data = whiteside)
coef(whitesideLinearModel)
```

```
## (Intercept)        Temp
##   5.4861933  -0.2902082
```

Principales componenetes del Analysis PCA

El análisis de componentes principales (PCA) es un método de análisis para conjuntos de datos numéricos multivariados que busca definir nuevas variables llamadas proyecciones o componentes principales que son combinaciones lineales de las variables originales, satisfaciendo las siguientes dos restricciones: 1. Cada componente principal no está correlacionado con todos los demás; 2. La varianza del componente principal es lo más grande posible, sujeta a la primera restricción.

```
str(crabs)
```

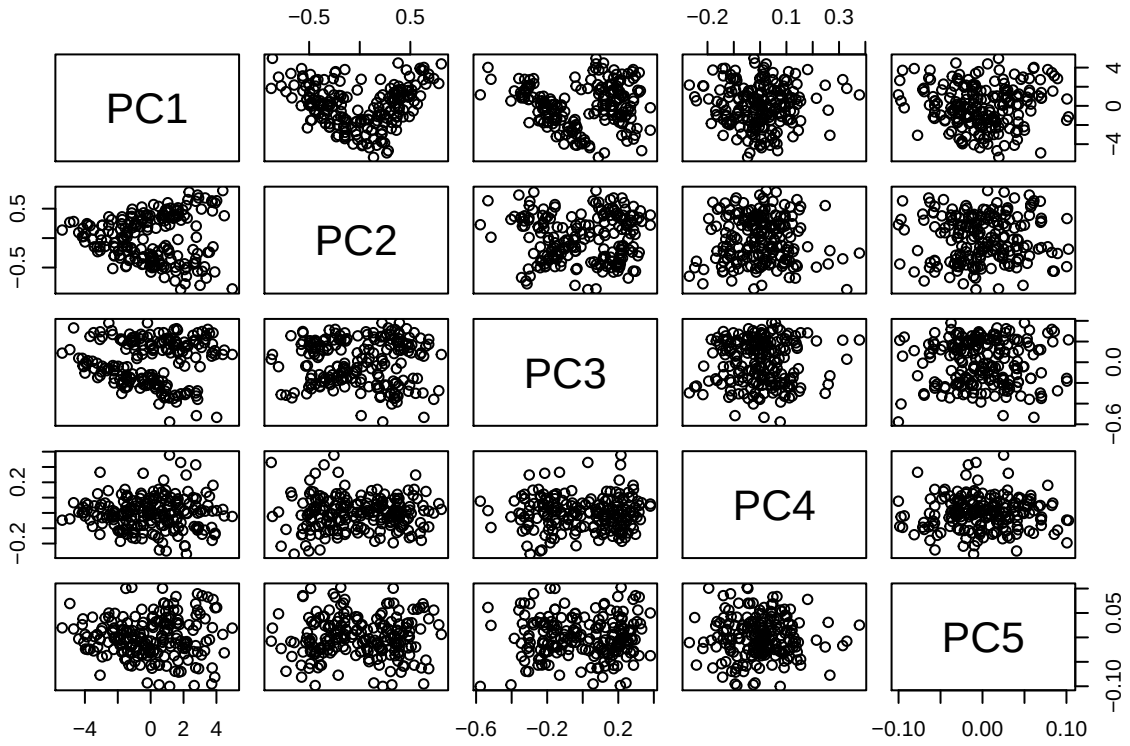
```
## 'data.frame':    200 obs. of  8 variables:
##  $ sp   : Factor w/ 2 levels "B","O": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex  : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ index: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ FL   : num  8.1 8.8 9.2 9.6 9.8 10.8 11.1 11.6 11.8 11.8 ...
##  $ RW   : num  6.7 7.7 7.8 7.9 8 9 9.9 9.1 9.6 10.5 ...
##  $ CL   : num  16.1 18.1 19 20.1 20.3 23 23.8 24.5 24.2 25.2 ...
##  $ CW   : num  19 20.8 22.4 23.1 23 26.5 27.1 28.4 27.8 29.3 ...
##  $ BD   : num  7 7.4 7.7 8.2 8.2 9.8 9.8 10.4 9.7 10.3 ...
```

```
prinComp <- prcomp(crabs[, 4:8], center = TRUE, scale = TRUE)
summary(prinComp)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5
```

```
## Standard deviation     2.1883 0.38947 0.21595 0.10552 0.04137
## Proportion of Variance 0.9578 0.03034 0.00933 0.00223 0.00034
## Cumulative Proportion  0.9578 0.98810 0.99743 0.99966 1.00000
```

```
pcaDF <- as.data.frame(prinComp$x)
plot(pcaDF)
```




```
sex <- crabs$sex
sp <- crabs$sp
crabTypeMB <- as.numeric((sex == "M") & (sp == "B"))
crabTypeFO <- as.numeric((sex == "F") & (sp == "O"))
crabTypeMO <- as.numeric((sex == "M") & (sp == "O"))
dummyDF <- data.frame(crabTypeMB = crabTypeMB,
crabTypeFO = crabTypeFO,
crabTypeMO = crabTypeMO)
cor(dummyDF)
```

```
##            crabTypeMB crabTypeFO crabTypeMO
## crabTypeMB  1.0000000 -0.3333333 -0.3333333
## crabTypeFO -0.3333333  1.0000000 -0.3333333
## crabTypeMO -0.3333333 -0.3333333  1.0000000
```

```
crabs2 <- crabs[, 4:8]
crabs2 <- cbind.data.frame(crabs2, dummyDF)
prinComp2 <- prcomp(crabs2, center = TRUE, scale = TRUE)
```

Trabajando con variables de tiempo

Base R admite una clase de objeto Date, junto con una serie de herramientas útiles para trabajar con estos objetos. Como un simple ejemplo, considere:

```
dayInYear <- seq(1, 365, 1)
dates <- as.Date("2017-01-01") + dayInYear - 1
```

```r
dates[1:10]
```

```
##  [1] "2017-01-01" "2017-01-02" "2017-01-03" "2017-01-04" "2017-01-05"
##  [6] "2017-01-06" "2017-01-07" "2017-01-08" "2017-01-09" "2017-01-10"
```

```r
dates[360:365]
```

```
## [1] "2017-12-26" "2017-12-27" "2017-12-28" "2017-12-29" "2017-12-30"
## [6] "2017-12-31"
```

```r
as.Date("2017-05-23") - as.Date("2017-02-19")
```

```
## Time difference of 93 days
```

```r
min(dates)
```

```
## [1] "2017-01-01"
```

```r
max(dates)
```

```
## [1] "2017-12-31"
```

```r
median(dates)
```

```
## [1] "2017-07-02"
```

```r
scheduleFrame <- data.frame(dayInYear = dayInYear, date = dates)
scheduleFrame$dayOfWeek <- weekdays(scheduleFrame$date)
scheduleFrame$month <- months(scheduleFrame$date)
scheduleFrame$quarter <- quarters(scheduleFrame$date)
str(scheduleFrame)
```

```
## 'data.frame':    365 obs. of  5 variables:
##  $ dayInYear: num  1 2 3 4 5 6 7 8 9 10 ...
##  $ date     : Date, format: "2017-01-01" "2017-01-02" ...
##  $ dayOfWeek: chr  "domingo" "lunes" "martes" "miércoles" ...
##  $ month    : chr  "enero" "enero" "enero" "enero" ...
##  $ quarter  : chr  "Q1" "Q1" "Q1" "Q1" ...
```

```r
write.csv(scheduleFrame, "scheduleFrame.csv", row.names = FALSE)
scheduleFrame2 <- read.csv("scheduleFrame.csv")
str(scheduleFrame2, vec.len = 2)
```

```
## 'data.frame':    365 obs. of  5 variables:
##  $ dayInYear: int  1 2 3 4 5 ...
##  $ date     : Factor w/ 365 levels "2017-01-01","2017-01-02",..: 1 2 3 4 5 ...
##  $ dayOfWeek: Factor w/ 7 levels "domingo","jueves",..: 1 3 4 5 2 ...
##  $ month    : Factor w/ 12 levels "abril","agosto",..: 4 4 4 4 4 ...
##  $ quarter  : Factor w/ 4 levels "Q1","Q2","Q3",..: 1 1 1 1 1 ...
```