UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ÀREA DE SISTEMES D'INFORMACIÓ I COMUNICACIONS

Institut de Ciències de l'Educació

Clasificación de nubes de puntos mediante análisis discriminante y el software R

Apellidos, nombre	Balaguer Beser, Ángel (abalague@mat.upv.es), Piedra Mora, Adolfo Enrique (aepiemor@topo.upv.es)
Departamento	Departamento de Matemática Aplicada
Centro	E.T.S.I. Geodésica, Cartográfica y Topográfica Universitat Politècnica de València

Resumen de las ideas clave

En este objeto docente se muestra el proceso de cálculo del porcentaje de datos bien clasificados usando la técnica del análisis discriminante lineal con la información proporcionada por datos LiDAR y el software estadístico R en el entorno de RStudio. Se ha aplicado con datos reales obtenidos a través del Plan Nacional de Fotografía Aérea (PNOA) en una zona con vegetación de la provincia de Valencia (España). Se han analizado las funciones discriminantes para ver las variables que separan mejor los datos, elaborando gráficos de dispersión y gráficos de funciones discriminantes.

1. Objetivos

Después de leer con detenimiento este documento, el lector será capaz de:

- Calcular funciones discriminantes para clasificar datos LiDAR.
- Calcular el porcentaje de datos bien clasificados usando el análisis discriminante.
- Usar probabilidades a priori en la clasificación por análisis discriminante.
- Analizar la separación por categorías con gráficos de funciones discriminantes.
- Aplicar el software R para obtener diagramas y resultados del análisis discriminante para ver las diferencias entre puntos de diferentes categorías y clasificar datos.

2. Introducción

El PNOA (Plan Nacional de Fotografía Aérea) proporciona datos LiDAR obtenidos mediante sensores aerotransportados instalados en aviones (escáner láser, receptor GPS y sistema de navegación inercial) que emiten pulsos de luz hacia el suelo. Al medir el tiempo de retorno del pulso se calcula la distancia exacta, generando una densa nube de puntos con coordenadas tridimensionales del terreno. Cuando el avión emite los pulsos láser hacia el suelo, si a su paso encuentra elementos con huecos, como por ejemplo la copa de un árbol, puede generar múltiples retornos que regresan al sensor en distintos momentos. Por ejemplo, el primer retorno es la primera superficie que impacta el láser y suele corresponder a la parte más alta de los árboles. Los retornos intermedios pueden corresponder con los rebotes que ocurren entre las ramas del árbol o diferentes capas de vegetación densa y, el último retorno es la última superficie alcanzada, que, en la gran mayoría de los casos, corresponde al suelo real. Por eso, los datos LiDAR del PNOA incluyen la información del número de retorno.

Los datos del PNOA se proporcionan en mosaicos de 2 km × 2 km de puntos de datos brutos en un archivo binario LAZ (archivo LAS comprimido) (véase [4]), que contiene coordenadas X e Y (EPSG:25830 ETRS89/UTM, Zona 30N) y la cota ortométrica Z. Para este trabajo consideramos un recorte de puntos de uno de estos mosaicos en una zona de vegetación de la Comunitat Valenciana. Cualquiera de los archivos descargados en formato LAZ viene con los puntos

etiquetados numéricamente bajo el estándar internacional de la ASPRS (Sociedad Americana de Fotogrametría y Teledetección), que divide los puntos en categorías esenciales: Suelo (2), Vegetación (3,4,5), Edificios (6), Ruido (7) y Solape (12). En este trabajo hemos considerado únicamente las clases de suelo y vegetación.

La clasificación de nubes de puntos mediante Análisis Discriminante lineal (ADL) es una técnica estadística supervisada que asigna puntos individuales a categorías predefinidas (p. ej., vegetación y suelo). Utiliza características geométricas para maximizar la separación entre clases y minimizar la variación interna (véase [3]). Su punto de partida es una tabla de datos con varias variables continuas y una variable categórica que indica el grupo al que pertenece cada individuo de la muestra. Esta técnica clasifica un nuevo punto en un grupo de dicha variable categórica con la información del conjunto de variables continuas. Para entrenar el modelo clasificación se usa una muestra de puntos que se sabe que están bien clasificados en uno de los grupos. El ADL también se puede usar para entender las diferencias entre los puntos que pertenecen a distintos grupos y ver qué tienen en común los puntos de un mismo grupo.

3. Desarrollo

Para trabajar con el software R en el entorno de R-Studio (véase [5]) usaremos la librería lidR, diseñada específicamente para la lectura, manipulación, procesamiento y visualización de nubes de puntos LiDAR. Su principal fortaleza es que está optimizada para manejar archivos de gran tamaño (formatos .las y .laz) de forma eficiente, permitiendo automatizar flujos de trabajo científicos y cartográficos complejos. Además, usaremos lidRviewer, que es un visor 3D complementario desarrollado específicamente para dar soporte visual al paquete lidR, que permite renderizar nubes de puntos masivas directamente desde la pantalla del ordenador.

Este último se puede instalar desde este repositorio:

install.packages('lidRviewer', repos = c('https://r-lidar.r-universe.dev'))

y a continuación, tras instalar el paquete lidR, procedemos a cargar las dos librerías.

library(lidR)

library(lidRviewer)

Los controles de LidRviewer permiten entre otras cosas rotar con el botón izquierdo del ratón, hacer zoom con la rueda del ratón y desplazar con el botón derecho del ratón. Si los puntos se ven muy pequeños y dispersos, o si están tan juntos que saturan la pantalla, se puede ajustar su escala visual utilizando: Tecla + (Símbolo de sumar) para aumentar el tamaño de los puntos y la Tecla - (Símbolo de restar) para hacerlos más pequeños. Para pintar la nube según las categorías asignadas automáticamente por el PNOA (por ejemplo, suelo en un color y vegetación en otro) se debe pulsar la Tecla c (de clasificación) y si se desea volver a ver la nube coloreada por su altura (coordenada Z) se puede pulsar la tecla z. Si se desea verla por la fuerza del retorno del láser, se tiene que pulsar la tecla i (de Intensity).

La imagen 1 muestra un recorte de los datos originales del PNOA del año 2023 con los que trabajamos, usando las siguientes sentencias de R

las = readLAS(ruta_las_PNOA_recortada)

lidRviewer::view(filter_poi(las, Classification %in% c(2,3,4,5)))

Visualizamos sólo las categorías de clasificación números 2, 3, 4 y 5, que corresponde a zonas de suelo y vegetación.

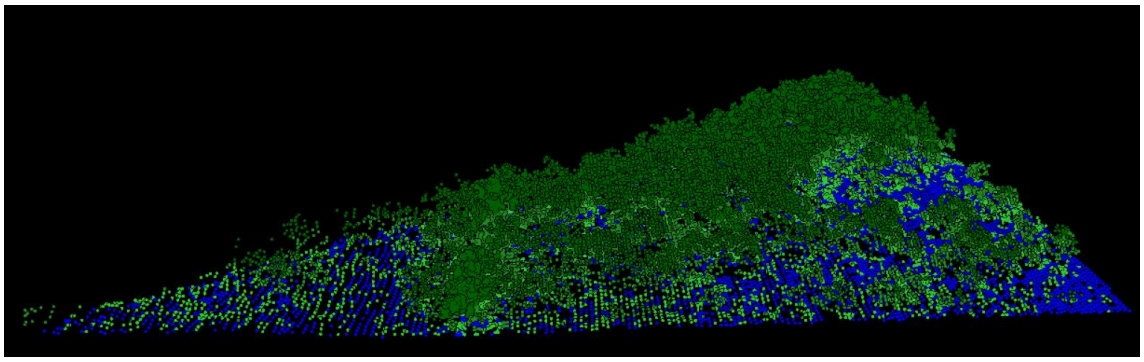

Imagen 1. Imagen obtenida con el visor LidRviewer, pulsando la tecla c para pintar los puntos según la categoría en la que están clasificados.

Para poder ver la cantidad de puntos con los que trabajamos y los puntos que tiene cada categoría de vegetación podemos usar, respectivamente, estas sentencias de R.

> npoints(las)

[1] 42804

> table(las$Classification)

1	2	3	4	5	7
1293	9540	6815	8801	16319	36

La clase 7 corresponde a puntos con ruido y la clase 1 son puntos no clasificados. Por eso nos quedamos solo con las clases 2, 3, 4 y 5, visualizando los datos en formato tabla y definiendo las variables con las que vamos a trabajar. Estas son las sentencias de R para ello y para visualizar los resultados con un diagrama de caja y bigotes, como el que se observa en la imagen 2.

las = filter_poi(las, Classification %in% c(2,3,4,5))

tabla_las = data.frame(X = las$X, Y = las$Y, Z = las$Z, RETURN_NUMBER = las$ReturnNumber, CLASSIFICATION = las$Classification)

boxplot(tabla_las[,4:5])

El suelo (categoría 2) suele tener el último número de retorno de un pulso láser, el cual habitualmente corresponde al retorno número 2, 3 o 4 (dependiendo de la densidad de la vegetación que haya encima) o al retorno número 1 si el terreno está completamente despejado.

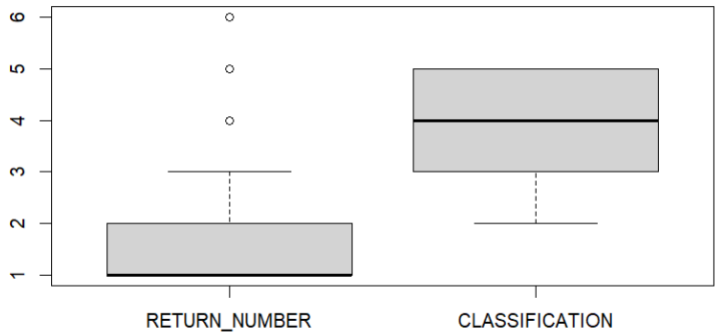

Imagen 2 Diagrama de caja y bigotes para el número de retorno y la categoría de clasificación de los puntos.

Con el siguiente procedimiento de R podemos ver cómo se distribuye el número de retorno (columnas) entre los puntos que pertenecen a cada categoría de clasificación (filas).

```
table(tabla_las$CLASSIFICATION,tabla_las$RETURN_NUMBER)
```

```
      1    2    3   4   5  6
2  3856 2821 2066 672 116  9
3  3297 2032 1169 290  26  1
4  4068 3303 1219 196  15  0
5 11734 3974  580  31   0  0
```

Es importante verificar que nuestras variables son capaces de separar los grupos de clasificación. Para reflexionar sobre ello, conviene mirar la imagen 3 que muestra las diferencias en la coordenada Z original en función de la posición Y de los datos. ¿Piensas que existe alguna transformación de los datos que podría mejorar la separación entre clases? Se propone al lector efectuar otros diagramas de dispersión cambiando las variables de los ejes de coordenadas.

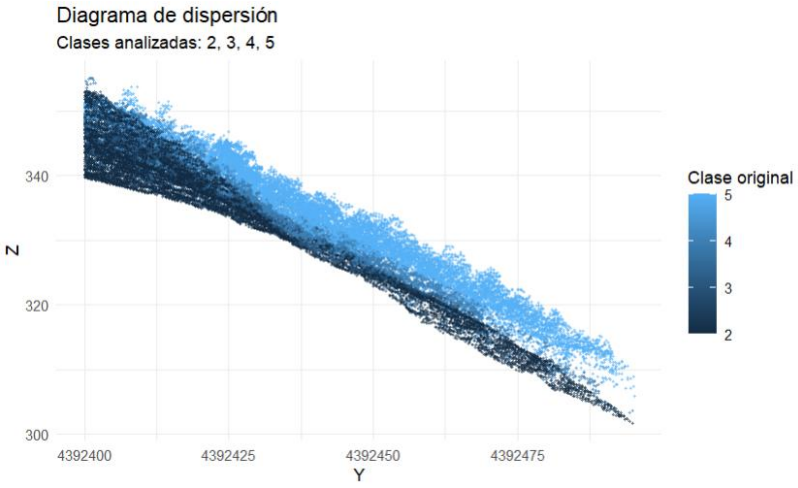

Imagen 3 Diagrama de dispersión para las coordenadas X y Z, para las categorías de clasificación de suelo y vegetación.

Para ello, se debe tener en cuenta que para hacer el diagrama de dispersión de la imagen 3 se han usado estas sentencias con R.

```
library(ggplot2)

clases_usar <- c(2, 3, 4, 5)

ggplot( tabla_las,aes(x = Y, y = Z, color = CLASSIFICATION)) + geom_point(size = 0.5, alpha = 0.5) + theme_minimal() + labs(title = "Diagrama de dispersión", subtitle = paste("Clases analizadas:", paste(clases_usar, collapse = ", ")), x = "Y",y = "Z",color = "Clase original")
```

Los datos originales con los que trabajamos tienen como variables continuas la X e Y, que son las coordenadas geográficas planimétricas de cada punto y, la variable Z, que es la altitud ortométrica, conocida frecuentemente como altura sobre el nivel del mar. Probamos cambiar la variable Z para trabajar con alturas normalizadas, calculadas restando a cada dato el modelo digital del terreno básico. Para ello usaremos estas sentencias de R, entre las cuales usamos la función plot() para observar el resultado obtenido.

```
las_norm <- normalize_height(las, tin())

plot(las_norm, color = "Z") # Ahora Z = 0 será estrictamente el suelo plano

las=las_norm
```

Se pueden efectuar algunos resúmenes estadísticos por grupos como muestran estos estadísticos básicos (mínimo, máximo y media) usando los datos con alturas normalizadas.

```
aggregate(. ~ CLASSIFICATION, data = tabla_las, FUN = min)
CLASSIFICATION        X       Y      Z RETURN_NUMBER
             2 728087.9 4392400  0.000             1
             3 728087.3 4392400 -0.981             1
             4 728087.8 4392400  0.562             1
             5 728101.5 4392400  2.217             1
aggregate(. ~ CLASSIFICATION, data = tabla_las, FUN = max)
CLASSIFICATION      X       Y      Z RETURN_NUMBER
             2 728200 4392495  0.020             6
             3 728200 4392494  1.106             6
             4 728200 4392495  3.021             5
             5 728200 4392495 10.718             4
aggregate(. ~ CLASSIFICATION, data = tabla_las, FUN = mean)
CLASSIFICATION        X       Y             Z RETURN_NUMBER
             2 728159.4 4392429 5.241090e-06      1.993501
             3 728147.6 4392424 3.940726e-01      1.784886
             4 728149.1 4392428 1.903520e+00      1.725940
             5 728154.8 4392438 5.291035e+00      1.320301
```

Si nos fijamos en la coordenada Z normalizada, existe cierta confusión entre los grupos 2 y 3. Por este motivo las categorías números 2 y 3 las vamos a agrupar en una misma clase denominada con el número 2. El resto de las categorías (4 y 5) se mantienen con la clasificación original. También filtramos los posibles valores faltantes y consideramos la variable

"Classification" como un factor para el modelo usando el procedimiento que se describe a continuación.

```
variables <- c("X", "Y", "Z", "ReturnNumber", "Classification")

df_clases_sel <- las@data[, ..variables]

df_clases_sel <- na.omit(df_clases_sel)

df_clases_sel$Classification[df_clases_sel$Classification %in% c(3) ] <-2

df_clases_sel$Classification <- as.factor(df_clases_sel$Classification)
```

Para aplicar la técnica del análisis discriminante usamos el 80% de los datos de la muestra para entrenar el modelo y el 20% restante para testeo, y para ello hacemos la siguiente selección aleatoria de datos:

```
set.seed(123)

idx <- sample(1:nrow(df_clases_sel), 0.8 * nrow(df_clases_sel))

df_train <- df_clases_sel[idx, ]

df_test  <- df_clases_sel[-idx, ]
```

Calculamos el modelo de Análisis Discriminante Lineal usando la función lda() del paquete MASS en R usando dichos datos de entrenamiento. El modelo trabaja por defecto con probabilidades previas calculadas basándose en la proporción de datos de cada clase en la base de datos de entrenamiento.

```
> library(MASS)
> modelo_lda <- lda(Classification ~ X + Y + Z + ReturnNumber,data = df_train)
> print(modelo_lda)
Call:
lda(Classification ~ X + Y + Z + ReturnNumber, data = df_train)

Prior probabilities of groups:
        2         4         5
0.3951175 0.2127788 0.3921037

Group means:
          X       Y         Z ReturnNumber
2 728154.4 4392427 0.1632544     1.910984
4 728149.6 4392428 1.9109516     1.729745
5 728154.6 4392438 5.2954878     1.319139

Coefficients of linear discriminants:
                     LD1         LD2
X             0.002035153  0.02296463
Y             0.008237918  0.02162291
Z            -1.009410525 -0.09913391
ReturnNumber -0.028403674 -0.41574656

Proportion of trace:
   LD1    LD2
0.9989 0.0011
```

Los coeficientes de las funciones discriminantes lineales (que R muestra bajo el encabezado Coefficients of linear discriminants) representan los ponderadores o pesos matemáticos asignados a cada variable predictora para construir las funciones discriminantes (véase [3]). Se tienen tantas funciones discriminantes como grupos menos uno, salvo cuando el número de variables predictoras es menor que el número de grupos menos uno. Geométricamente, estos coeficientes definen las direcciones de los vectores que logran separar de forma óptima los centros de los tres grupos. Los valores que da la función lda() son coeficientes No Estandarizados, que indican cómo multiplicar cada variable original para calcular la puntuación discriminante (score) de cada observación, que mostraremos luego en la imagen 4. ¿Por qué no se usan estos coeficientes directamente para ver la importancia de las variables en la clasificación? Para medir la importancia de cada variable se debería estandarizar los coeficientes multiplicándolos por la desviación estándar ponderada dentro de los grupos.

sds <- apply(df_train[, c("X", "Y", "Z", "ReturnNumber")], 2, sd)

coef_estandarizados <- modelo_lda$scaling * sds

Con ello se obtiene la tabla que se muestra a continuación, la cual muestra que los pesos que las variables tienen en cada función discriminante.

```
> print(coef_estandarizados)
                    LD1        LD2
X             0.05748040  0.6486077
Y             0.18521083  0.4861419
Z            -2.54647360 -0.2500884
ReturnNumber -0.02377653 -0.3480187
```

Con este último resultado, ¿qué variables tienen una mayor importancia en la clasificación? También se debe tener en cuenta que al final de la página anterior la "proportion of trace" indica el porcentaje de la separación total entre grupos que logra explicar cada función discriminante, la cual representa la importancia relativa de cada dimensión en la clasificación de los datos.

Para cambiar las probabilidades previas en un modelo de ADL y hacer que todos los grupos sean tratados con la misma probabilidad, se puede utilizar el argumento prior dentro de la función lda() del paquete MASS. Para igualarlas, debes pasarle un vector numérico donde la suma de todos sus elementos sea exactamente 1 como se muestra a continuación.

n_clases <- nlevels(df_train$Classification)

prob_iguales <- rep(1 / n_clases, n_clases)

modelo_lda1 <- lda(Classification ~ X + Y + Z + ReturnNumber,data = df_train, prior = prob_iguales)

Aplicamos función predict() al modelo con probabilidades a priori que tienen en cuenta los tamaños de cada clase. Dicha función devuelve una lista con tres componentes básicas. La primera (class) contiene la categoría predicha con mayor probabilidad para cada observación. La segunda (posterior) muestra la probabilidad a posteriori de que una observación pertenezca

a cada grupo. La última (x) proporciona las puntuaciones discriminantes (scores), que son las coordenadas de las observaciones en el nuevo espacio generado por las funciones discriminantes lineales. ¿Para qué sirven estas puntuaciones discriminantes?

```
> pred_train <- predict(modelo_lda, newdata = df_train)
> head(pred_train$class)
[1] 5 2 5 4 5 5
Levels: 2 4 5
> head(pred_train$posterior)
             2            4            5
1 4.872947e-05 0.0118144547 9.881368e-01
2 8.790828e-01 0.1209147802 2.378424e-06
3 1.578176e-04 0.0394088530 9.604333e-01
4 3.286392e-01 0.6684942593 2.866497e-03
5 1.726832e-07 0.0002680176 9.997318e-01
6 5.624133e-05 0.0190656760 9.808781e-01
> head(pred_train$x)
         LD1        LD2
1 -2.1341835  1.3509859
2  2.3572787 -0.9623506
3 -1.8858978 -1.0725909
4  0.7624694 -0.3474095
5 -3.2511524  1.9381049
6 -2.0948748 -0.7467054
```

Para ver la separación entre los puntos de entrenamiento en el nuevo espacio discriminante, se deja para el lector que elabore el diagrama de dispersión de la imagen 4. El procedimiento es similar al desarrollado anteriormente para obtener la imagen 3.

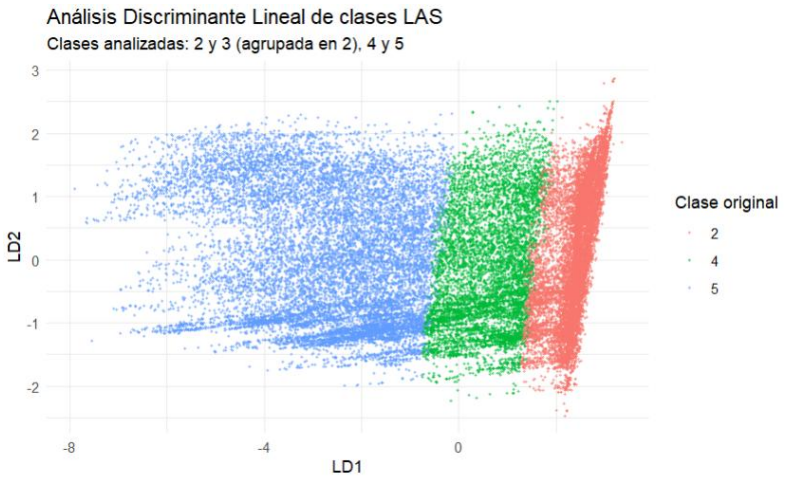

Imagen 4 Diagrama de dispersión para las puntuaciones discriminantes de los datos, indicando la clase original con colores.

El modelo ADL que hemos calculado lo podemos aplicar para clasificar nuevos datos. Por eso hemos reservado unos datos (el 20% del total) para testear el modelo. Con ellos podemos obtener la siguiente matriz de confusión y calcular el porcentaje de datos clasificados correctamente, que nos indicarán la exactitud del modelo.

```
> pred_test <- predict(modelo_lda, newdata = df_test)
> matriz_confusion<-table(df_test$Classification,pred_test$class)
> print(matriz_confusion)

     2    4    5
  2 3241    4    0
  4  457 1284    0
  5    0  321 2988
> exactitud <- sum(diag(matriz_confusion)) / sum(matriz_confusion)
> cat("\nExactitud global (Accuracy):", round(exactitud * 100, 2), "%\n")

Exactitud global (Accuracy): 90.57 %
```

El porcentaje de datos bien clasificados es igual al 90.57%.

4. Conclusiones y cierre

Hemos aplicado una técnica de análisis discriminante lineal para clasificar nuevos datos usando las variables obtenidas a través de la información del LiDAR PNOA. El software R, usado con el entorno de RStudio, ha permitido calcular el porcentaje de datos correctamente clasificados en una muestra tipo test y representar gráficamente los puntos de la muestra de entrenamiento en el espacio discriminante para observar la separación entre las categorías. También se han obtenido las variables que tienen una mayor influencia en la clasificación.

Se propone al lector, repetir el procedimiento de análisis discriminante lineal descargando unos datos LiDAR del PNOA de otra zona forestal, pero usando como factor de clasificación los grupos obtenidos mediante un análisis clúster con el software R. Para obtener dichas categorías se podría usar el procedimiento descrito en la referencia [2]. También se propone comparar los resultados con los obtenidos tras realizar una selección de variables para el análisis discriminante. Para ello se pueden consultar los procedimientos que tiene el programa Statgrapchics Centurion visualizando el objeto de aprendizaje descrito en la referencia [1].

Bibliografía

[1] Balaguer-Beser, ÁA. (2017). Clasificación mediante análisis discriminante con Statgraphics. https://riunet.upv.es/handle/10251/84057

[2] Balaguer-Beser, Ángel (2025). Agrupación de datos espaciales usando análisis clúster y la información de índices espectrales. https://riunet.upv.es/handle/10251/221828

[3] Cuadras, C. M. (2020). Nuevos métodos de análisis multivariante (Vol. 20). Barcelona: CMC editions. https://www.ub.edu/biost3/cuadras/metodos.pdf

[4] Novo, A., Fariñas-Álvarez, N., Martínez-Sánchez, J., González-Jorge, H., & Lorenzo, H. (2020). Automatic processing of aerial LiDAR data to detect vegetation continuity in the surroundings of roads. Remote Sensing, 12(10), 1677.

[5] Posit team (2025). RStudio: Integrated Development Environment for R. Posit Software, PBC, Boston, MA. URL http://www.posit.co/