Introducción al aprendizaje automático

Christian Guanoquiza, Esteban Narea

Uidad 1

Introducción al aprendizaje automático

El aprendizaje automático se centra en aplicaciones prácticas. La tarea de enseñar a una computadora a aprender implica un problema específico, que podría ser una computadora que pueda jugar un juego, pensar en filosofía o responder preguntas. El aprendizaje automático proporciona un conjunto de herramientas que las computadoras pueden usar para convertir los datos en conocimiento procesable.

Los orígenes del aprendizaje automático

Desde el nacimiento los sensores de nuestro cuerpo son expuestos a datos sin procesar que nuestro cerebro traduce en imágenes, sonidos, olores, sabores y texturas.

Las primeras bases de datos registraron información del entorno observable. Entre bases de datos y sensores, cada aspecto de nuestras vidas está siendo registrado. Gobiernos, empresas y particulares registran y reportan todo tipo de información: sensores meteorológicos registran datos de temperatura y presión, cámaras de vigilancia monitorean aceras y túneles del metro, y se monitorea todo tipo de comportamiento electrónico: transacciones, comunicaciones, amistades, etc.

El campo de investigación que desarrolla algoritmos informáticos para transformar los datos en un comportamiento inteligente se conoce como aprendizaje automático, se originó en un lugar en el que los datos disponibles, los métodos estadísticos y la potencia informática se desarrollaban simultáneamente con rapidez. El aumento en la cantidad de datos requiere potencia informática adicional, lo que ha fomentado el desarrollo de métodos estadísticos para el análisis de grandes conjuntos de datos.

La minería de datos se ocupa de la generación de información novedosa a partir de grandes bases de datos. La distinción entre el aprendizaje automático y la minería de datos es que el aprendizaje tiende a centrarse en realizar una tarea conocida, mientras que la minería de datos se trata de la búsqueda de pepitas de información ocultas.

Nota: Los algoritmos de aprendizaje automático son prácticamente un requisito previo para la minería de datos, pero no ocurre lo contrario.

Usos y abusos del aprendizaje automático

El aprendizaje automático puede dar sentido a datos complejos y se ha utilizado para predecir resultados electorales, identificar y filtrar spam, predecir actividades delictivas y construir aviones y automóviles autónomos.

En cada uno de estos contextos un algoritmo de aprendizaje automático toma datos e identifica patrones que se pueden usar para la acción.

Consideraciones éticas

Debido a la novedad y la rapidez del aprendizaje automático como disciplina, los problemas legales y las normas sociales relacionadas son muy inciertos, por lo que se debe tener cuidado al recopilar o analizar datos para evitar infringir la ley, el uso inapropiado de los datos puede perjudicar sus resultados. Los clientes pueden sentirse incómodos o intimidados cuando se hacen públicos aspectos de sus vidas que consideran privados.

Nota: El hecho de que pueda usar los datos para un fin particular no siempre significa que deba hacerlo.

¿Cómo aprenden las máquinas?

Tom M. Mitchell, dice que se dice que una máquina aprende si es capaz de tomar experiencia y utilizarla de tal manera que su rendimiento mejore en experiencias similares en el futuro. El proceso básico de aprendizaje se puede dividir en tres componentes de la siguiente manera: Entrada de datos: utiliza la observación, el almacenamiento de memoria y el recuerdo para proporcionar una base fáctica para un razonamiento posterior. Abstracción: Implica la traducción de datos en representaciones más amplias. Generalización: utiliza datos abstractos para formar una base para la acción.

Figure 1: Proceso básico de aprendizaje

Abstracción y representación del conocimiento

La representación de datos de entrada sin procesar en un formato estructurado es la tarea para un algoritmo de aprendizaje, antes de esto los datos son simplemente unos y ceros en un disco

o en la memoria; no tienen significado. El trabajo de asignar un significado a los datos ocurre durante el proceso de abstracción.

Durante la representación del conocimiento, las computadoras ensamblan la entrada sin procesar en modelos como ecuaciones, diagramas, reglas lógicas if/else y grupos de datos llamados clústeres. La selección del modelo generalmente no se deja en manos de la máquina. En cambio, el modelo está determinado por la tarea de aprendizaje y el tipo de datos que se analizarán.

El proceso de ajustar un modelo particular a un conjunto de datos se conoce como entrenamiento, el entrenamiento describe con mayor precisión el proceso real que se lleva a cabo cuando el modelo se ajusta a los datos. El aprendizaje implica razonamiento inductivo de abajo hacia arriba. El entrenamiento connota mejor el hecho de que el maestro humano impone el modelo de aprendizaje automático al estudiante máquina, proporcionando a la computadora una estructura que intenta modelar.

Cuando el modelo ha sido entrenado, los datos se han transformado en una forma abstracta que resume la información original.

Generalización

La generalización describe el proceso de transformar el conocimiento abstracto en una forma procesable. Tradicionalmente, esto se considera como una búsqueda en todo el conjunto de modelos que pueden haber sido adquiridos durante el entrenamiento.

Los algoritmos de aprendizaje automático a menudo usan atajos que desglosan conjuntos de conceptos más rápido. Para hacer esto, el algoritmo utilizará heurística, o conjeturas informadas, para determinar dónde se pueden encontrar los conceptos más importantes.

Las heurísticas empleadas por los algoritmos de aprendizaje automático también dan lugar a veces a conclusiones erróneas. Si las conclusiones son sistemáticamente imprecisas, se dice que el algoritmo tiene un sesgo.

Evaluar el éxito del aprendizaje

El sesgo es un mal necesario asociado con el proceso de abstracción y generalización inherente a cualquier tarea de aprendizaje automático. El paso final en el proceso de generalización es determinar el éxito del modelo a pesar de sus sesgos.

Una vez que un modelo ha sido entrenado en un conjunto de datos inicial, el modelo se prueba en un nuevo conjunto de datos y se juzga en qué medida su caracterización de los datos de entrenamiento se generaliza a los nuevos datos.

El hecho de que el modelo no genere perfectamente se debe a problemas de ruido o variaciones de datos inexplicables, como errores de medición debido a sensores imprecisos que a veces suman o restan un poco de las lecturas, problemas de datos informados, como encuestas aleatorias informadas por los encuestados y preguntas para una finalización más rápida.

Pasos para aplicar el aprendizaje automático a sus datos

• Recopilación de datos: Deberá recopilarlos en un formato electrónico adecuado para el análisis. Estos datos servirán como material de aprendizaje que utiliza un algoritmo para generar conocimiento procesable.

• Explorar y preparar los datos: requiere una gran cantidad de intervención humana. Se dedica a aprender más sobre los datos y sus matices durante una práctica llamada exploración de datos.

• Entrenamiento de un modelo sobre los datos: Para cualquier tarea de aprendizaje automático se puede dividir en una serie de pasos más manejables que son: análisis, es probable que tenga una idea de lo que espera aprender de los datos, la tarea específica de aprendizaje automático informará la selección de un algoritmo apropiado, y el algoritmo representará los datos en forma de modelo.

• Evaluación del rendimiento del modelo: Es importante evaluar qué tan bien aprendió el algoritmo a partir de su experiencia. Según el tipo de modelo utilizado, es posible que pueda evaluar la precisión del modelo mediante un conjunto de datos de prueba o que necesite desarrollar medidas de rendimiento específicas para la aplicación prevista.

• Mejora del rendimiento del modelo: si se necesita un mejor rendimiento, puede ser necesario cambiar a un tipo de modelo completamente diferente o complementar sus datos con datos adicionales o realizar un trabajo preparatorio adicional como en el paso dos de este proceso. Después de completar estos pasos, si el modelo parece estar funcionando satisfactoriamente, se puede implementar para la tarea prevista.

Elegir un algoritmo de aprendizaje automático

El proceso de elegir un algoritmo de aprendizaje automático implica hacer coincidir las características de los datos que se van a aprender con los sesgos de los enfoques disponibles.

Datos de entrada:

Todos los algoritmos de aprendizaje automático requieren datos de entrenamiento de entrada. El formato exacto puede diferir, pero en su forma más básica, los datos de entrada toman la forma de ejemplos y características.

Las características vienen en varias formas también. Si una característica representa una característica medida en números, como era de esperar, se llama numérica. Alternativamente, si mide un atributo que está representado por un conjunto de categorías, la característica se denomina categórica o nominal. Un caso especial de variables categóricas se llama ordinal, que designa una variable nominal con categorías que caen en una lista ordenada.

Un modelo predictivo se usa para tareas que involucran, como su nombre lo indica, la predicción de un valor usando otros valores en el conjunto de datos. El algoritmo de aprendizaje intenta descubrir y modelar la relación entre la característica objetivo (la característica que se predice) y las otras características. Debido a que los modelos predictivos reciben instrucciones claras sobre lo que necesitan aprender y cómo deben aprenderlo, el proceso de entrenamiento de un modelo predictivo se conoce como aprendizaje supervisado.

Se utiliza un modelo descriptivo para tareas que se beneficiarían de la información obtenida al resumir datos de formas nuevas e interesantes. ¡A diferencia de los modelos predictivos que predicen un objetivo de interés; en un modelo descriptivo, ninguna característica es más importante que otra. La tarea de modelado descriptivo de dividir un conjunto de datos en grupos homogéneos se denomina agrupación. Esto a veces se usa para el análisis de segmentación que identifica grupos de personas con información similar de compras, donaciones o demográfica para que las campañas publicitarias se puedan adaptar a audiencias particulares.

Coincidir sus datos con un algoritmo apropiado. Para hacer coincidir una tarea de aprendizaje con un enfoque de aprendizaje automático, deberá comenzar con uno de los cuatro tipos de tareas: clasificación, predicción numérica, detección de patrones o agrupación.

Para la clasificación, se necesita más pensamiento para hacer coincidir un problema de aprendizaje con un clasificador apropiado. En estos casos, es útil considerar las diversas distinciones entre los algoritmos.

Uso de R para el aprendizaje autónomo.

Gracias a que R es un software gratuito de código abierto, no hay cargo adicional por esta funcionalidad. Una gran comunidad de expertos que contribuyeron al software agregó los algoritmos necesarios para el aprendizaje automático a la base R. Existen paquetes gratuitos para cada uno de los algoritmos de aprendizaje automático que se tratan en este libro. De hecho, este libro solo cubre una pequeña parte de los paquetes de aprendizaje automático más populares. **Instalación y carga de paquetes de R**

Instalación: Las opciones de instalación predeterminadas son apropiadas para la mayoría de los sistemas. Sin embargo, en algunos casos, es posible que desee instalar un paquete en otra ubicación. Por ejemplo, si no tiene privilegios de raíz o administrador en su sistema, es posible que deba especificar una ruta de instalación alternativa.

En un sistema Microsoft Windows, se puede acceder a esto desde el elemento de comando Instalar paquete(s) en el menú Paquetes, como se muestra en la siguiente capEn Windows, después de iniciar el instalador del paquete (y elegir una ubicación espejo CRAN si aún no lo ha hecho), aparecerá una gran lista de paquetes. Simplemente desplácese hasta el paquete RWeka y haga clic en el botón Aceptar para instalar el paquete y todas las dependencias en la ubicación predeterminada.tura de pantalla.

Carga de paquetes: Para conservar la memoria, R no carga todos los paquetes instalados de forma predeterminada. En su lugar, los usuarios cargan los paquetes a medida que los necesitan mediante la función library() .

Unidad 2

Gestión y comprensión de datos

Vectores:

La estructura de datos fundamental de R es el vector, que almacena un conjunto ordenado de valores llamados elementos. Un vector puede contener cualquier número de elementos. Sin embargo, ¡todos los elementos deben ser del mismo tipo; por ejemplo, un vector no puede contener tanto números como texto.

Hay varios tipos de vectores comúnmente utilizados en el aprendizaje automático: enteros (números sin decimales), numéricos (números con decimales), caracteres (datos de texto) o lógicos (valores VERDADERO o FALSO).

Debido a que los vectores R están inherentemente ordenados, se puede acceder a los registros contando el número del elemento en el conjunto, comenzando en 1, y rodeando este número con corchetes (por ejemplo, [y]) después del nombre del vector.

¿Por qué no usar vectores de caracteres? Una ventaja de usar factores es que generalmente son más eficientes que los vectores de caracteres porque las etiquetas de categoría se almacenan solo una vez. En lugar de almacenar MASCULINO, MASCULINO, FEMENINO, la computadora puede almacenar 1, 1, 2. Esto puede ahorrar memoria. Además, ciertos algoritmos de aprendizaje automático usan rutinas especiales para manejar variables categóricas.

Listas:

Otro tipo especial de vector, una lista, se utiliza para almacenar un conjunto ordenado de valores. Debido a esta flexibilidad, las listas a menudo se usan para almacenar varios tipos de datos de entrada y salida y conjuntos de parámetros de configuración para modelos de aprendizaje automático.

Marcos de datos:

La estructura de datos R más importante utilizada en el aprendizaje automático es el marco de datos, una estructura análoga a una hoja de cálculo o base de datos, ya que tiene filas y columnas de datos. En términos de R, un marco de datos puede entenderse como una lista de vectores o factores, cada uno de los cuales tiene exactamente el mismo número de valores.

En comparación con los vectores, factores y listas unidimensionales, un marco de datos tiene dos dimensiones y, por lo tanto, se muestra en formato de matriz. El marco de datos tiene una columna para cada vector de datos de pacientes y una fila para cada paciente. En términos de aprendizaje automático, las columnas son las características o atributos y las filas son los ejemplos.

Para extraer valores en el marco de datos, ¡podemos usar métodos como los que aprendimos para acceder a valores en vectores, con una excepción importante; debido a que el marco de datos es bidimensional, deberá especificar la posición de las filas y las columnas que desea extraer.

Si desea más de una fila o columna de datos, puede hacerlo especificando vectores para los números de fila y columna que desea. La siguiente declaración extraerá datos de las filas 1 y 3, y de las columnas 2 y 4.

Los métodos que hemos aprendido para acceder a valores en listas y vectores también se pueden usar para recuperar filas y columnas de marcos de datos. Por ejemplo, se puede acceder a las columnas por nombre en lugar de por posición, y se pueden usar signos negativos para excluir filas o columnas de datos.

Matrices y arreglos:

Una matriz es una estructura de datos que representa una tabla bidimensional, con filas y columnas de datos. Las matrices R pueden contener cualquier tipo de datos, aunque se usan con mayor frecuencia para operaciones matemáticas y, por lo tanto, normalmente almacenan solo datos numéricos.

Para crear una matriz, simplemente proporcione un vector de datos a la función matrix() , junto con un parámetro que especifique el número de filas (nrow) o el número de columnas (ncol).

Estrechamente relacionado con la estructura matricial está el arreglo, que es una tabla de datos multidimensional. Donde una matriz tiene filas y columnas de valores, una matriz tiene filas, columnas y cualquier cantidad de capas adicionales de valores. Aunque ocasionalmente usaremos matrices en capítulos posteriores.

Guardar y cargar estructuras de datos R

Para guardar una estructura de datos en particular en un archivo que pueda volver a cargarse más tarde o transferirse a otro sistema, puede usar la función save() . La función save() escribe estructuras de datos R en la ubicación especificada por el parámetro de archivo .

El comando load () recreará cualquier estructura de datos ya guardada que estuviera en un archivo .RData . Para cargar el archivo mydata.RData que guardamos en el código anterior, simplemente escriba:

cargar("misdatos.RData")

Todas las estructuras de datos almacenadas en el archivo que está importando con el comando load() se agregarán a su espacio de trabajo, incluso si sobrescriben algo más en lo que está trabajando.

Importar y guardar datos de archivos CSV

Los archivos de texto se pueden leer en prácticamente cualquier computadora o sistema operativo, lo que hace que el formato sea casi universal. También se pueden exportar e importar desde/hacia programas como Microsoft Excel, lo que brinda una manera rápida y fácil de trabajar con datos de hojas de cálculo.

El formato de archivo de texto tabular más común es el archivo de valores separados por comas (CSV) , que, como sugiere el nombre, usa la coma como delimitador. Los archivos CSV se pueden importar y exportar desde muchas aplicaciones comunes.

Para cargar este archivo CSV en R, se usa read.csv () de la siguiente manera:

pt_data < read.csv(“pt_data.csv”, stringsAsFactors = FALSO)

Si sus datos residen fuera del directorio de trabajo de R, puede especificar la ruta al archivo CSV especificando la ruta completa, por ejemplo, /ruta/a/misdatos.csv al llamar a la función read.csv () .

La función read.csv () es un caso especial de la función read.table () , que puede leer datos tabulares en muchas formas diferentes, incluidos otros formatos delimitados, como el valor separado por tabuladores (TSV).

Importación de datos de bases de datos SQL:

ODBC es un protocolo estándar para conectarse a bases de datos independientemente del sistema operativo o DBMS (Sistema de gestión de bases de datos). Si se ha conectado con anterioridad a una base de datos a través de ODBC, lo más probable es que se haya referido a ella a través de su DSN (Nombre de fuente de datos). Necesitará el DSN, además de un nombre de usuario y contraseña (si su base de datos lo requiere) para usar RODBC.

instalar.paquetes(“RODBC”)

biblioteca (RODBC)

Exploración y comprensión de datos:

Es durante este paso que comenzará a explorar las características y los ejemplos de los datos, y se dará cuenta de las peculiaridades que hacen que sus datos sean únicos. Cuanto mejor comprenda sus datos, mejor podrá hacer coincidir un modelo de aprendizaje automático con su problema de aprendizaje.

Dado el marco de datos de usedcars, ahora asumiremos el rol de un científico de datos, que tiene la tarea de comprender los datos de los autos usados. Aunque la exploración de datos es un proceso fluido, los pasos pueden imaginarse como una especie de investigación en la que se responden preguntas sobre los datos.

Explorando la estructura de los datos:

La función str() proporciona un método para mostrar la estructura de un marco de datos o cualquier estructura de datos R, incluidos vectores y listas. Se puede utilizar para crear el esquema básico de nuestro diccionario de datos:

str(autosusados)

Después del nombre de la variable, chr nos dice que la función es de tipo carácter. En este conjunto de datos, tres de las variables son de carácter, mientras que tres se indican como int, lo que indica un tipo de número entero.

Aunque nuestros datos parecen haber recibido nombres de variables significativos, no siempre es así. A veces, los conjuntos de datos tienen características con nombres y códigos sin sentido o simplemente un número como V1

Explorando variables numéricas:

Emplearemos un conjunto de medidas de uso común para describir valores conocidos como estadísticas de resumen. La función summary () muestra varias estadísticas de resumen comunes.

Medición de la tendencia central: media y mediana Las medidas de tendencia central son una clase de estadísticas que se utilizan para identificar un valor que se encuentra en el medio de un conjunto de datos. Lo más probable es que ya estés familiarizado con una medida común del centro: el promedio. En el uso común, cuando algo se considera promedio, cae en algún lugar entre los extremos de la escala.

En estadística, el promedio también se conoce como la media, una medida definida como la suma de todos los valores dividida por el número de valores.

Aunque la media es, con mucho, la estadística más comúnmente citada para medir el centro de un conjunto de datos no siempre es la más adecuada. Otra medida de tendencia central de uso común es la mediana, que es el valor que se encuentra en la mitad de una lista ordenada de valores. Al igual que con la media, R proporciona una función mediana ().

Recuerde nuevamente los valores medianos informados en la salida de resumen () para el conjunto de datos de automóviles usados. Aunque la media y la mediana del precio son bastante similares (difieren en aproximadamente un 5 por ciento).

Como era de esperar, el mínimo y el máximo son los valores más extremos que se encuentran en el conjunto de datos, lo que indica los valores más pequeños y grandes respectivamente. R proporciona las funciones min () y Max () para calcular estos valores en un vector de datos.

Visualización de variables numéricas:

Diagramas de caja La visualización de variables numéricas puede ser útil para diagnosticar muchos problemas con los datos.

Una visualización común del resumen de cinco números es un diagrama de caja o un diagrama de caja y bigotes. El gráfico de caja muestra el centro y la dispersión de una variable numérica en un formato que le permite obtener rápidamente una idea del rango y el sesgo de una variable, o compararla con otras variables.

boxplot(usedcars$price, main = "Boxplot de Precios de Autos Usados", ylab = "Precio()“)

boxplot(usedcars$mileage, main=“Boxplot de kilometraje de autos usados”, ylab=“Cuentakilómetros (millas)”)

El diagrama de caja y patillas representa los valores de resumen de cinco números usando líneas horizontales. Las líneas horizontales que forman el cuadro en el medio de cada figura representan Q1, Q2 (la mediana) y Q3 cuando se lee el gráfico de abajo hacia arriba.

Es similar a un diagrama de caja en el sentido de que divide los valores de la variable en un número predefinido de porciones o contenedores que actúan como contenedores de valores. Un diagrama de caja requiere que cada una de las cuatro porciones de datos contenga la misma cantidad de valores, y amplía o reduce los intervalos según sea necesario. Por el contrario, un histograma utiliza cualquier número de contenedores de idéntico ancho, pero permite que los contenedores contengan diferentes cantidades de valores.

El histograma se compone de una serie de barras con alturas que indican el recuento o la frecuencia de los valores que se encuentran dentro de cada uno de los contenedores de igual tamaño que dividen los valores.

Ejemplo de histograma

También puede notar que la forma de los dos histogramas es algo diferente. Parece que los precios de los autos usados tienden a dividirse equitativamente a ambos lados del medio, mientras que el kilometraje de los autos se extiende más hacia la derecha.

Es importante tener en cuenta que no todos los eventos aleatorios son uniformes. Por ejemplo, lanzar un dado de truco de seis caras con peso daría como resultado que algunos números salieran con más frecuencia que otros. Si bien cada tirada del dado da como resultado un número seleccionado al azar, no son igualmente probables.

La desviación estándar se puede usar para estimar rápidamente qué tan extremo es un valor dado bajo el supuesto de que proviene de una distribución normal. La regla 689599,7 establece que el 68 % de los valores en una distribución normal se encuentran dentro de una desviación estándar de la media, mientras que el 95 % y el 99,7 % de los valores se encuentran dentro de dos y tres desviaciones estándar, respectivamente.

Explorando variables categóricas:

Debido a que usamos el parámetro stringsAsFactors = FALSE al cargar los datos, R los ha dejado como variables de caracteres (chr) en lugar de convertirlos automáticamente en factores. ¡Además, podríamos considerar tratar el año como categórico; aunque es como un valor numérico (int), el valor de cada año es una categoría que podría aplicarse a varios autos.

Los modos se utilizan en un sentido cualitativo para obtener una comprensión de los valores importantes en un conjunto de datos. Sin embargo, sería peligroso poner demasiado énfasis en la moda ya que el valor más común no es necesariamente una mayoría.

Pensar en las modas como valores comunes nos permite aplicar el concepto de moda estadística a los datos numéricos. Estrictamente hablando, sería poco probable tener una moda para una variable continua, ya que es probable que no se repitan dos valores. Sin embargo, si pensamos en las modas como las barras más altas de un histograma, podemos analizar las modas de variables como el precio y el kilometraje.

Explorando relaciones entre variables:

Visualización de relaciones: Diagramas de dispersión Un diagrama de dispersión es un diagrama que visualiza una relación bivariada. Es una figura bidimensional en la que se dibujan puntos en un plano de coordenadas utilizando los valores de una característica para proporcionar las coordenadas x horizontales y los valores de otra característica para proporcionar las coordenadas verticales y.

Para usar plot (), necesitamos especificar los vectores x e y que contienen los valores usados para colocar los puntos en la figura. Aunque las conclusiones serían las mismas independientemente de qué variable se utilice para proporcionar las coordenadas x e y, la convención dicta que la variable y es la que se supone que depende de la otra (y por lo tanto se la conoce como la variable dependiente).

Ahora, veamos una tabulación cruzada para ver cómo la proporción de autos de colores conservadores varía según el modelo. Dado que asumimos que el modelo de automóvil dicta la elección del color, trataremos a la conservadora como la variable dependiente (y). Por lo tanto, el comando CrossTable() es:

$$\text{CrossTable}(\text{x} = \text{autosusados}\mathit{modelo}, y = \mathit{autosusados}\text{conservador})$$

Hay una gran cantidad de datos en la salida de CrossTable() . La leyenda en la parte superior (con la etiqueta Contenido de la celda) indica cómo interpretar cada valor. Las filas de la tabla indican los tres modelos de autos usados: SE, SEL y SES (más una fila adicional para el total de todos los modelos). Las columnas indican si el color del automóvil es conservador o no (más una columna que totaliza ambos tipos de color).

Puede obtener los resultados de la prueba de chicuadrado agregando un parámetro adicional que especifique chisq = TRUE al llamar a la función CrossTable() . En nuestro caso, la probabilidad es de alrededor del 93 por ciento, lo que sugiere que es muy probable que las variaciones en el recuento de células se deban únicamente al azar y no a una verdadera asociación entre el modelo y el color.

Codigo

```
usedcars <- read.csv("usedcars.csv", stringsAsFactors = FALSE)
str(usedcars)
```

```
'data.frame':   150 obs. of  6 variables:
 $ year        : int  2011 2011 2011 2011 2012 2010 2011 2010 2011 2010 ...
 $ model       : chr  "SEL" "SEL" "SEL" "SEL" ...
 $ price       : int  21992 20995 19995 17809 17500 17495 17000 16995 16995 16995 ...
 $ mileage     : int  7413 10926 7351 11613 8367 25125 27393 21026 32655 36116 ...
 $ color       : chr  "Yellow" "Gray" "Silver" "Gray" ...
 $ transmission: chr  "AUTO" "AUTO" "AUTO" "AUTO" ...
```

```r
summary(usedcars$year)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2000    2008    2009    2009    2010    2012
```

```r
summary(usedcars[c("price","mileage")])
```

```
     price          mileage
 Min.   : 3800   Min.   :  4867
 1st Qu.:10995   1st Qu.: 27200
 Median :13592   Median : 36385
 Mean   :12962   Mean   : 44261
 3rd Qu.:14904   3rd Qu.: 55125
 Max.   :21992   Max.   :151479
```

```r
(36000+44000+56000)/3
```

```
[1] 45333.33
```

```r
mean(c(36000,44000,56000))
```

```
[1] 45333.33
```

```r
median(c(36000,44000,56000))
```

```
[1] 44000
```

```r
range(usedcars$price)##rango de precios más bajo y alto
```

```
[1]  3800 21992
```

```r
diff(range(usedcars$price))##diferencia entre el valor más alto y más bajo
```

```
[1] 18192
```

```r
IQR(usedcars$price)
```

```
[1] 3909.5
```

```r
quantile(usedcars$price)
```

```
    0%     25%     50%     75%    100%
3800.0 10995.0 13591.5 14904.5 21992.0
```

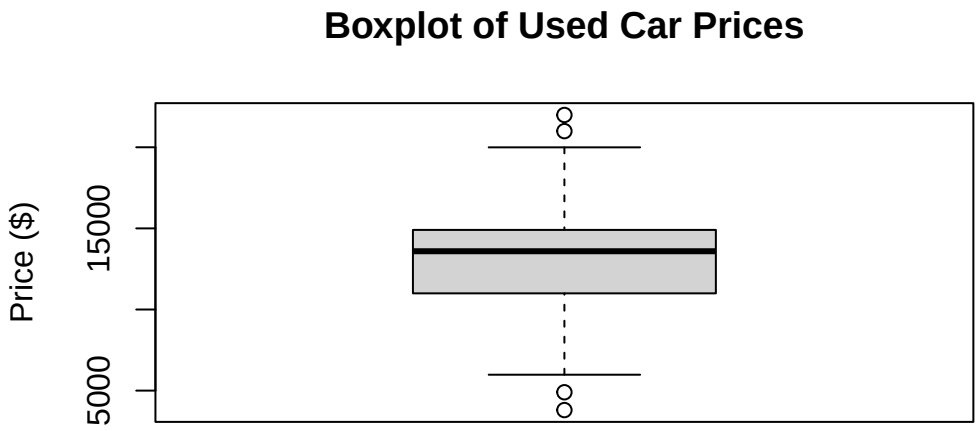
```r
quantile(usedcars$price,probs = c(0.01,0.99))
```

```
     1%      99%
5428.69 20505.00
```

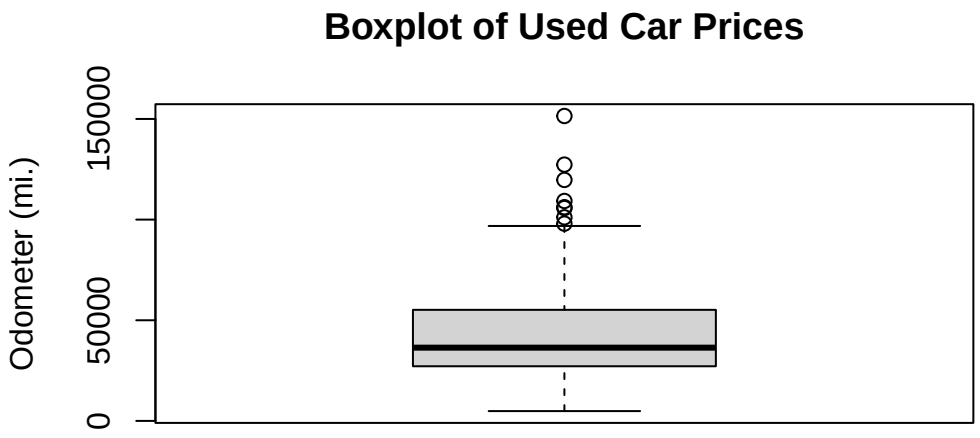
```r
quantile(usedcars$price, seq(from=0, to=1,by=0.20))
```

```
    0%     20%     40%     60%     80%    100%
3800.0 10759.4 12993.8 13992.0 14999.0 21992.0
```

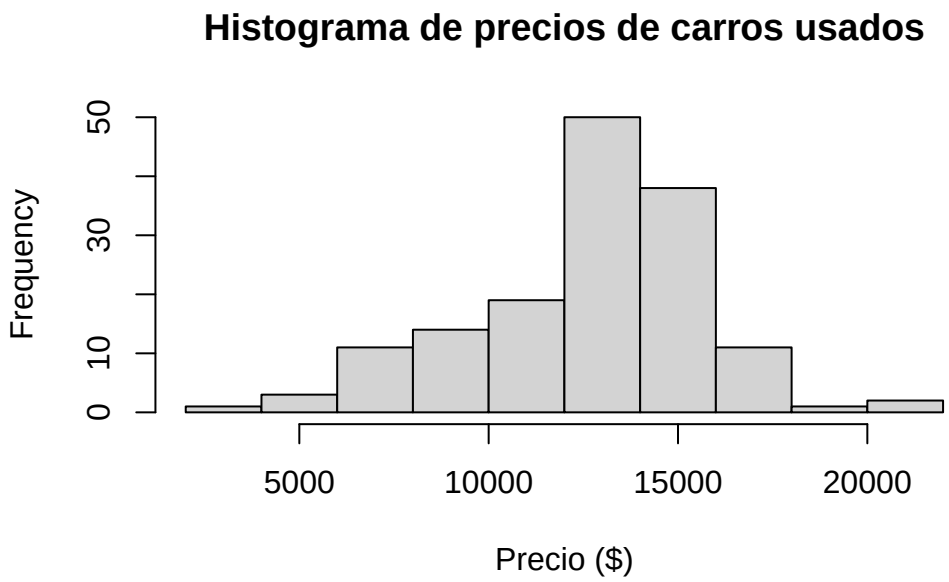
```r
boxplot(usedcars$price,main="Boxplot of Used Car Prices",ylab="Price ($)")
```



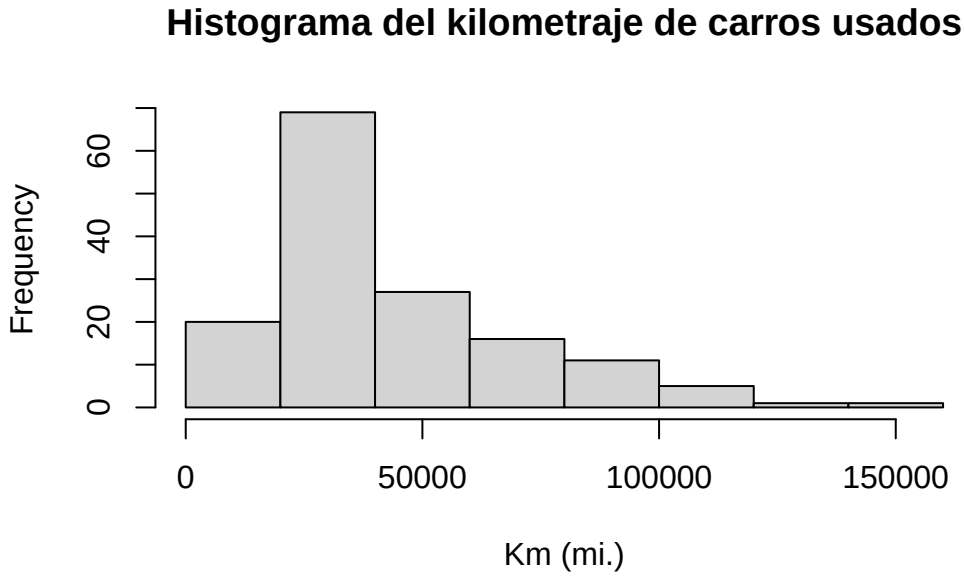
```r
boxplot(usedcars$mileage,main="Boxplot of Used Car Prices",ylab="Odometer (mi.)")
```




```r
hist(usedcars$price,main="Histograma de precios de carros usados",xlab="Precio ($)")
```




```r
hist(usedcars$mileage,main="Histograma del kilometraje de carros usados",xlab="Km (mi.)")
```



```r
var(usedcars$price)##varianza
```

```
[1] 9749892
```

```r
sd(usedcars$price)##desviación estandar
```

```
[1] 3122.482
```

```r
var(usedcars$mileage)
```

```
[1] 728033954
```

```r
sd(usedcars$mileage)
```

```
[1] 26982.1
```

```r
table(usedcars$year)##cantidad de carros por año
```

```
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
   3    1    1    1    3    2    6   11   14   42   49   16    1
```

```r
table(usedcars$model)##cantidad de carros por modelo
```

```
 SE SEL SES
 78  23  49
```

```r
table(usedcars$color)##cantidad de carros por color
```

```
 Black   Blue   Gold   Gray  Green    Red Silver  White Yellow
    35     17      1     16      5     25     32     16      3
```

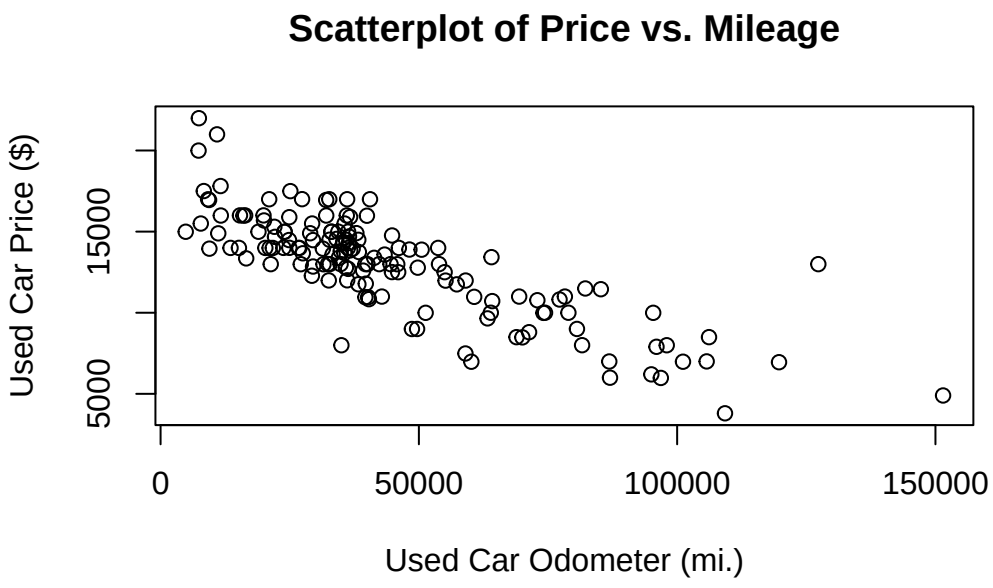
```r
model_table<-table(usedcars$model)
prop.table(model_table)##el 52 por ciento de los autos son del tipo SE.
```

```
      SE       SEL       SES
0.5200000 0.1533333 0.3266667
```

```r
color_table<-table(usedcars$color)
color_pct<-prop.table(color_table)*100
round(color_pct,digits=1)## Los resultados muestran que el negro es el color más común, ya
```

```
 Black   Blue   Gold   Gray  Green    Red Silver  White Yellow
  23.3   11.3    0.7   10.7    3.3   16.7   21.3   10.7    2.0
```

```r
##casi una cuarta parte (23,3 por ciento) de todos los automóviles anunciados son negros.
##con 21.3 por ciento y el rojo es tercero con 16.7 por ciento.
plot(x=usedcars$mileage,y=usedcars$price,main="Scatterplot of Price vs. Mileage", xlab="Us
```



```
##el diagrama de dispersión, notamos una clara relación entre el precio de un usado coche
```