Programming Assignment 3

R Programming

**Introducción**

* Descargue el archivo ProgAssignment3-data.zip que contiene los datos para la Tarea de programación 3 del sitio web de Coursera.
* Descomprima el archivo en un directorio que servirá como su directorio de trabajo.
* Cuando inicie R, asegúrese de cambiar su directorio de trabajo al directorio donde descomprimió los datos.

Los datos para esta asignación provienen del sitio web de Hospital Compare (<http://hospitalcompare.hhs.gov>) dirigido por el Departamento de Salud y Servicios Humanos de EE. UU.

El propósito del sitio web es proporcionar datos e información sobre la calidad de la atención en más de 4,000 hospitales certificados por Medicare en los EE.UU. Este conjunto de datos cubre esencialmente todos los principales hospitales de EE. UU. Este conjunto de datos se utiliza para una variedad de propósitos, incluida la determinación de si los hospitales deben ser multados por no brindar atención de alta calidad a los pacientes (ver <http://goo.gl/jAXFX> para algunos antecedentes sobre este tema en particular).

El sitio web de Hospital Compare contiene muchos datos y solo analizaremos un pequeño subconjunto para esta tarea. El archivo zip para esta tarea contiene tres archivos:

* outcome-of-care-measures.csv: Contiene información sobre la mortalidad a los 30 días y las tasas de reingreso por ataques cardíacos, insuficiencia cardíaca y neumonía en más de 4,000 hospitales.
* hospital-data.csv: Contiene información sobre cada hospital.
* Hospital\_Revised\_Flatfiles.pdf: Descripciones de las variables en cada archivo (es decir, el libro de códigos).

Una descripción de las variables en cada uno de los archivos está en el archivo PDF incluido llamado Hospital\_Revised\_Flatfiles.pdf.

Este documento contiene información sobre muchos otros archivos que no están incluidos con esta asignación de programación. Deberá centrarse en las variables para el Número 19 (“Resultado de las medidas de atención.csv”) y el Número 11 (“Datos del hospital.csv”). Puede resultarle útil imprimir este documento (al menos las páginas de las Tablas 19 y 11) para tener a su lado mientras trabaja en esta tarea. En particular, los números de las variables para cada tabla indican índices de columna en cada tabla (es decir, "Nombre del hospital" es la columna 2 en el archivo outcome-of-care-measures.csv file).

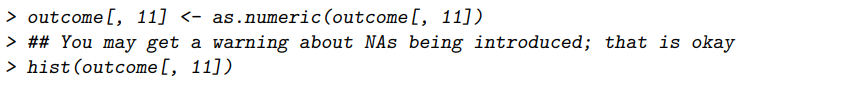
1. **Graficar las tasas de mortalidad a 30 días por ataque cardíaco**

Lea los datos de outcome en R a través de la función read.csv y observe las primeras filas.



Hay muchas columnas en este conjunto de datos. Puede ver cuántos escribiendo **ncol(outcome)** (puede ver el número de filas con la función nrow). Además, puede ver los nombres de cada columna escribiendo **names(outcome)** (los nombres también están en el documento PDF.

Para hacer un histograma simple de las tasas de mortalidad a los 30 días por ataque cardíaco (columna 11 en el conjunto de datos de outcome), ejecute:



Debido a que originalmente leímos los datos como caracteres, al especificar

**colClasses = "character"** necesitamos obligar a la columna a ser numérica. Puede recibir una advertencia sobre la introducción de NA pero eso está bien.

1. **Encontrar el mejor hospital en un estado**

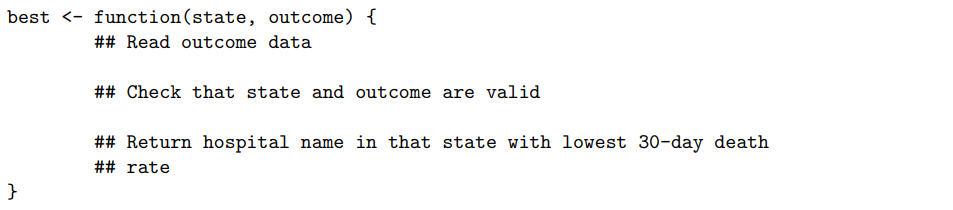
Escriba una función llamada **best** que tome dos argumentos: el nombre abreviado de 2 caracteres de un estado y el nombre outcome

La función lee el archivo outcome-of-care-measures.csv y devuelve un vector de caracteres con el nombre del hospital que tiene la mejor mortalidad (es decir, la más baja) de 30 días para el resultado especificado en ese estado.

El nombre del hospital es el nombre proporcionado en la variable Hospital.Name. Los resultados pueden ser uno de "heart attack", "heart failure" o "pneumonía". Los hospitales que no tienen datos sobre un resultado particular deben ser excluidos del conjunto de hospitales al decidir las clasificaciones.

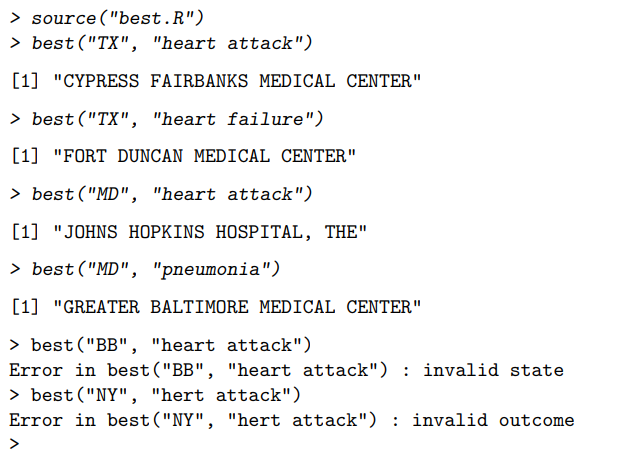
**Handling Ties (Manejo de empates):** Si hay un empate para el mejor hospital para un resultado dado, entonces los nombres de los hospitales se deben ordenar en orden alfabético y se debe elegir el primer hospital de ese conjunto (es decir, si los hospitales "b", "c" y "f" están empatados para mejor, entonces el hospital "b" debe ser devuelto).

La función debe usar la siguiente plantilla:



La función debe verificar la validez de sus argumentos. Si se pasa un valor de **state** no válido a **best**, la función debería arrojar un error a través de la función de **stop** con el mensaje exacto **"invalide state"**.Si se pasa un valor de **outcome** no válido a **best**, la función debería arrojar un error a través de la función de **stop** con el mensaje exacto **"invalide outcome"**.

Aquí hay algunos resultados de muestra de la función:



Guarde su código para esta función en un archivo llamado **best.R**.

1. **Clasificación de hospitales por resultado en un estado**

Escriba una función llamada **rankhospital** que tome tres argumentos: el nombre abreviado de 2 caracteres de un estado (**state**), un resultado (**outcome**) y la clasificación de un hospital en ese estado para ese resultado (**num**).

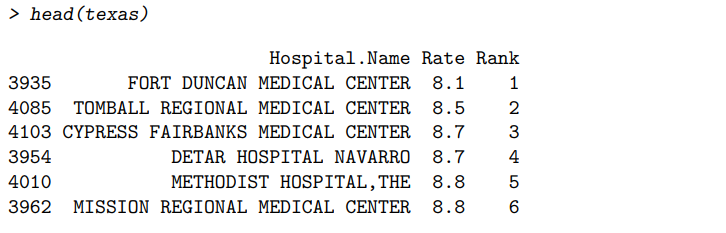
La función lee el archivo **outcome-of-care-measures.csv** y devuelve un vector de caracteres con el nombre del hospital que tiene la clasificación especificada por el argumento **num**. Por ejemplo, la llamada:



devolvería un vector de caracteres que contiene el nombre del hospital con la quinta tasa de mortalidad más baja de 30 días por heart failure. El argumento **num** puede tomar los valores **"best", "worst"** o un número entero que indica la clasificación (los números más pequeños son mejores). Si el número dado por **num** es mayor que el número de hospitales en ese estado, entonces la función debería devolver **NA**. Los hospitales que no tienen datos sobre un outcome particular deben ser excluidos del conjunto de hospitales al decidir las clasificaciones.

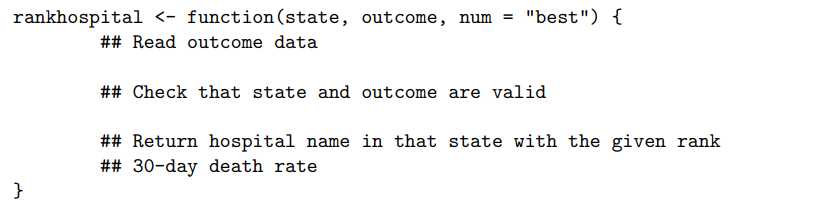
**Handling Ties (Manejo de empates):** Puede ocurrir que varios hospitales tengan la misma tasa de mortalidad de 30 días por una causa determinada de la muerte. En esos casos, los lazos deben romperse utilizando el nombre del hospital. Por ejemplo, en Texas ("TX"),

Aquí se muestran los hospitales con la tasa de mortalidad por insuficiencia cardíaca más baja en 30 días.



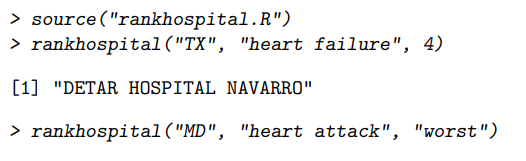
Tenga en cuenta que Cypress Fairbanks Medical Center y Detar Hospital Navarro tienen la misma tasa de 30 días (8.7). Sin embargo, debido a que Cypress viene antes de Detar alfabéticamente, Cypress ocupa el puesto número 3 en este esquema y Detar está clasificado como número 4. Uno puede usar la función de orden para ordenar múltiples vectores de esta manera (es decir, donde un vector se usa para romper empates en otro vector )

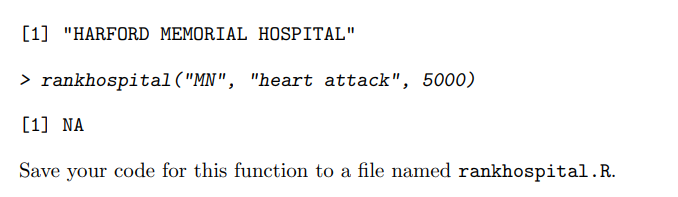
La función debe usar la siguiente plantilla:



La función debe verificar la validez de sus argumentos. Si se pasa un valor de **state** no válido a **rankhospital**, la función debería arrojar un error a través de la función de **stop** con el mensaje exacto "**invalide state**". Si se pasa un valor de **outcome** no válido a **rankhospital**, la función debería arrojar un error a través de la función de **stop** con el mensaje exacto **"invalide outcome"**.

Aquí hay algunos resultados de muestra de la función:





Guarda tu código para esta función con un archivo llamado **rankhospital.R**

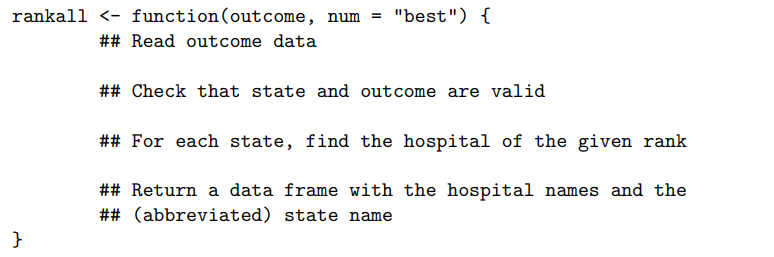
1. **Clasificación de hospitales en todos los estados**

Escriba una función llamada **rankall** que tome dos argumentos: un nombre de resultado (**outcome**) y una clasificación de hospital (**num**). La función lee el archivo **outcome-of-care-measures.csv** y devuelve un marco de datos de 2 columnas que contiene el hospital en cada state que tiene la clasificación especificada en **num**. Por ejemplo, la función llamada

devolvería un marco de datos que contiene los nombres de los hospitales que son los mejores en sus respectivos estados para las tasas de mortalidad por ataque cardíaco a 30 días. La función debe devolver un valor para cada estado (algunos pueden ser **NA**). La primera columna en el marco de datos se llama **hospital**, que contiene el nombre del hospital, y la segunda columna se llama **state**, que contiene la abreviatura de 2 caracteres para el nombre del estado. Los hospitales que no tienen datos sobre un resultado particular deben ser excluidos del conjunto de hospitales al decidir las clasificaciones.

**Handling Ties (Manejo de empates):** La función de **rankall** debe manejar los empates en las tasas de mortalidad de 30 días de la misma manera que la función **rankhospital** maneja los empates.

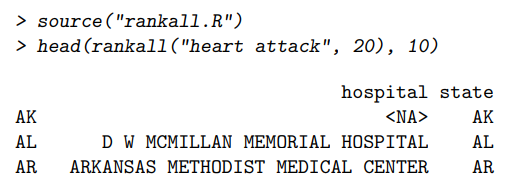
La función debe usar la siguiente plantilla:

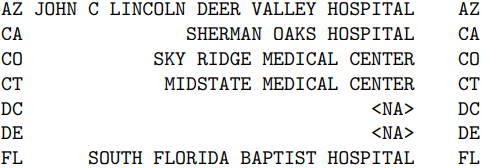


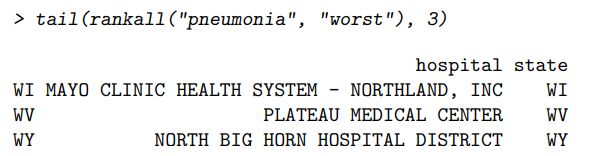
**NOTA:** Para el propósito de esta parte de la tarea (y para mayor eficiencia), su función NO debe llamar a la función **rankhospital** de la sección anterior.

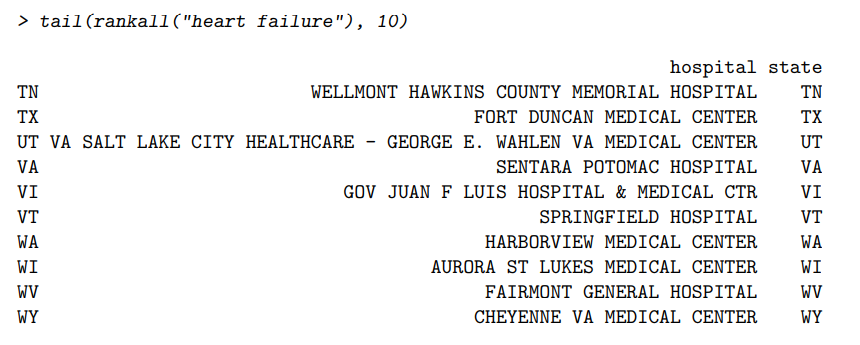
La función debe verificar la validez de sus argumentos. Si se pasa un valor de **outcome** no válido a **rankall**, la función debería arrojar un error a través de la función de **stop** con el mensaje exacto "invalid outcome". La variable **num** puede tomar los valores **"best", "worst"** o un número entero que indica la clasificación (los números más pequeños son mejores). Si el número dado por **num** es mayor que el número de hospitales en ese estado, entonces la función debería devolver **NA.**

Aquí hay algunos resultados de muestra de la función.









Guarda tu código para esta función con un archivo llamado rankall.R