Introducción a R y Tidyverse

Es esencial que, para progresar adecuadamente en la ruta de aprendizaje DS, instalen el lenguaje de programación R y el entorno de desarrollo integrado RStudio en sus computadoras. Estas herramientas son fundamentales para el desarrollo de las habilidades y prácticas que se abordarán en la ruta.

Si tienen dudas no duden en escribirme al mail: camila@datasketch.co
Subir avances a https://github.com/datasketch/R-Data-Science-Tasks

Conceptos Básicos de R:

1. Sintaxis básica de R.
2. Estructuras de datos en R (vectores, matrices, listas, data frames).
3. Formato de datos abiertos
4. Creación de proyecto en R
5. Instalación y manejo de paquetes.

Recursos:
Sintaxis básica y estructura de datos:

- https://rsanchezs.gitbooks.io/ciencia-de-datos-con-r/content/sintaxis/expresiones/expresiones.html
- https://cran.r-project.org/doc/contrib/rdebuts_es.pdf

Proyecto:

- https://epirhandbook.com/es/r-projects.html

Formato de datos:

- https://www.youtube.com/watch?v=zTq1clni3z8
- https://libguides.uprm.edu/datamanagement/format

Instalación y manejo de paquetes:

- https://r-pkgs.org/introduction.html
- https://www.youtube.com/watch?v=4ljTJQFu3po

Nota:
Para garantizar un avance estructurado y organizado en su aprendizaje de R, es crucial que cada actividad se gestione como un proyecto de R independiente. Recomiendo crear una carpeta principal en su computadora, nombrada ‘yoestudioR’ (o el nombre que deseen), como su espacio de trabajo dedicado. Dentro de esta carpeta, organice el contenido por temas, creando subcarpetas específicas para cada uno. Por ejemplo, para el tema de estructuras de datos, cree una subcarpeta llamada ‘estructura_datos’ dentro de ‘yoestudioR’. En esta, deberá guardar todos los archivos relevantes, incluyendo su proyecto de R, que se nombrará ‘estructura_datos.Rproj’. Esta estructura le ayudará a mantener sus archivos y proyectos bien ordenados y fácilmente accesibles a medida que avanza en el curso. (vea el recurso de proyecto https://epirhandbook.com/es/r-projects.html)

Actividad:
Crear un paquete en R cuya principal funcionalidad será recibir un data.frame o conjunto de datos externo (use las librerias pertinentes segun el formato de entrada de la base read_csv, openxlsx, rio) como entrada y devolver un resumen detallado de este. El resumen incluirá la clase de cada columna, junto con advertencias sobre datos faltantes (NA) o irregularidades en los datos que puedan sugerir 'suciedad' en la información.

Pasos de la Actividad:
Diseño del Paquete:

- Discusión sobre la estructura y funciones clave del paquete.
- Planificación de cómo el paquete procesará y analizará los datos.
- Desarrollo de Funciones Específicas:

Creación de una función que acepte un data.frame o conjunto de datos externo como entrada.

- Implementación de lógica para identificar la clase de cada columna.
- Desarrollo de algoritmos para detectar y advertir sobre NA y anomalías en los datos.

Documentación Adecuada:

- Uso de roxygen2 para documentar la funcionalidad del paquete y cada función.
- Inclusión de ejemplos y casos de uso en la documentación.

Incorporación de Mensajes de Advertencia:

- Programación de mensajes de advertencia informativos y claros para el usuario cuando se detecten datos faltantes o inconsistencias.

Pruebas y Validación:

- Escritura de pruebas con testthat para validar la funcionalidad.
- Comprobación de que el paquete maneja correctamente diferentes tipos de data.frames.

Elaboración de Vignettes:

- Creación de una o más vignettes demostrando el uso práctico del paquete.
- Inclusión de ejemplos de data.frames con distintos tipos de 'suciedad' en los datos.

Revisión de Pares y Mejoras:

- Revisión y mejora del paquete basada en los comentarios recibidos.

Recomiendo ver el siguiente video para depuración de errores:
https://www.youtube.com/watch?v=9vABzGCQeqU

Introducción a Tidyverse:

1. Filosofía y componentes del tidyverse.
2. tidyr para limpieza de datos (gather, spread, separate, unite, etc).
3. dplyr para manipulación de datos (select, filter, mutate, summarise, etc).

Recursos:
Introducción a tidyverse:

- https://ucodemy.github.io/rbioqadv/tidyverse/
- https://www.r-bloggers.com/2021/04/tidyverse-in-r-complete-tutorial/

tidyr:

- https://rpubs.com/jaortega/151936
- https://www.youtube.com/watch?v=4_7MDLTWflA
- https://www.youtube.com/watch?v=KYYtvX1WXn4

dplyr:

- https://rsanchezs.gitbooks.io/rprogramming/content/chapter9/index.html
- https://anderfernandez.com/blog/tutorial-dplyr/
- https://www.google.com/search?client=firefox-b-d&q=tutorial+de+dplyr#fpstate=ive&vld=cid:93960aa3,vid:InqoYLIc62M,st:0
- https://www.youtube.com/watch?v=raKdO4klDhg

Actividad:
En esta actividad, trabajará con un conjunto de datos reales proporcionado por el DANE (Departamento Administrativo Nacional de Estadística) de Colombia. Descargue el archivo "Anexo - Información base para el seguimiento al exceso de mortalidad (departamento y sexo)" el cual contiene datos sobre defunciones, que deberán limpiar y organizar utilizando tidyr.

Pasos de la Actividad:

Descarga y Exploración de Datos:

- Descargar el archivo de datos desde el sitio web del DANE.
- Realizar una exploración inicial para entender la estructura y el contenido del dataset.

Identificación de Problemas de Limpieza:

- Identificar problemas comunes de limpieza en los datos, como valores faltantes, inconsistencias, formatos de datos incorrectos, etc.

Aplicación de tidyr para Limpieza de Datos:

- Usar funciones de tidyr para abordar los problemas identificados.
- Transformar los datos para mejorar su organización y legibilidad.

Documentación del Proceso:

- Documentar cada paso del proceso de limpieza, explicando qué se hizo y por qué.
- Crear un reporte que muestre el antes y después de la limpieza.

Análisis Exploratorio Post-Limpieza:

- Realizar un análisis exploratorio básico de los datos limpios para verificar su integridad y prepararlos para análisis posteriores.

Reflexión y Discusión:

- Reflexionar sobre los desafíos encontrados durante la limpieza de datos.
- Discutir cómo estas habilidades pueden aplicarse en otros contextos de análisis de datos.

Limpieza y Manipulación Avanzada de Datos

Manipulación Avanzada con dplyr:

1. Joins, operaciones entre conjuntos.
2. Group_by y summarise para agregación de datos.
3. Manejo de datos faltantes.

Recursos:
Joins, operaciones entre conjuntos:

- https://r4ds.hadley.nz/joins.html
- https://dplyr.tidyverse.org/reference/setops.html
- https://dplyr.tidyverse.org/reference/mutate-joins.html

Group_by y summarise para agregación de datos.

- https://dplyr.tidyverse.org/reference/summarise.html
- https://american-stat-412612.netlify.app/material/1-06-lecture/
- https://www.youtube.com/watch?v=HVQomfQicz8

Datos faltantes:

- https://r4ds.hadley.nz/missing-values

Actividad:
En esta tarea, trabajará con los datos de la Encuesta Nacional de Lectura - ENLEC 2017, disponible en el sitio web del DANE. El conjunto de datos está en formato ZIP e incluye información sobre actividades de menores, uso de bibliotecas, prácticas de escritura y hábitos de lectura. Cada archivo XLSX contiene múltiples tablas en una sola hoja, las cuales deberá unificar y analizar, por lo tanto al final del proceso tendrá una base de datos consolidada con las diferentes actividades y la información que hay en cada excel.

Pasos de la Actividad:

Descarga y Exploración de Datos:

- Descargar el archivo ZIP de la ENLEC 2017 desde el sitio web proporcionado.
- Explorar los archivos XLSX para comprender la estructura y el contenido de las tablas.

Preparación y Unificación de Datos:

- Extraer y unificar las tablas de cada archivo XLSX, manteniendo la coherencia en la estructura de los datos.
- Agregar una nueva columna para identificar cada tabla (por ejemplo: cabeceras, centros poblados), excluyendo los totales nacionales.

Estructuración de Datos en Formato Tidy:

- Asegurarse de que la unión de tablas siga los principios del formato tidy (cada variable forma una columna, cada observación forma una fila).

Exploración de la base limpia:

- Haga una lista de las preguntas que podría responder con esta base de datos que consolidó.

Análisis Agrupado por Género:

- Agrupar los datos por género y realizar un análisis detallado de cada variable.
- Agrupar los datos por actividad y realizar un análisis detallado de cada variable.
- Agrupar los datos por actividad y género y realizar un análisis detallado de cada variable.
- Identificar patrones, tendencias y posibles correlaciones en las actividades de lectura y escritura.

Documentación y Reporte de Hallazgos:

- Documentar meticulosamente el proceso de manipulación y análisis de datos.
- Elaborar un informe que resuma los hallazgos clave, con gráficos y tablas para ilustrar los resultados. (Preferiblemente en Quarto document)

Limpieza de Datos:

1. Técnicas avanzadas de transformación de datos.
2. Uso de purrr para aplicar funciones a listas y vectores.

Recursos:
Transformación de datos:

- https://r4ds.had.co.nz/transform.html
- https://www.youtube.com/watch?v=DiY8EqZDwoI

Purrr:

- https://www.rebeccabarter.com/blog/2019-08-19_purrr
- https://sanderwuyts.com/en/blog/purrr-tutorial/

Actividad
Esta actividad involucra el uso del paquete nycflights13 para analizar el rendimiento de diferentes aerolíneas con base en el conjunto de datos flights (data(flights)). Utilizarán purrr para aplicar funciones de análisis y agregación a los datos.

Pasos de la Actividad:

Instalación y Carga de Paquetes:

- Instalar y cargar el paquete nycflights13 y purrr.
- Cargar el conjunto de datos flights y familiarizarse con su estructura.

Creación de Subconjuntos de Datos por Aerolínea:

- Dividir el conjunto de datos flights en subconjuntos, uno por cada aerolínea.
- Crear una lista en R donde cada elemento sea un subconjunto de datos correspondiente a una aerolínea.

Análisis de Datos con purrr:

- Utilizar purrr para aplicar funciones de análisis a cada subconjunto de datos. Por ejemplo, calcular el promedio de retrasos de salida y llegada, tasa de cancelaciones, etc., para cada aerolínea.
- Emplear funciones como map o map_df para realizar estas operaciones.

Programación Funcional y Visualización con ggplot2

Programación Funcional con purrr:

1. Conceptos básicos de programación funcional.
2. Map, walk y funciones relacionadas.
3. Integración de purrr con dplyr.
4. Visualización de Datos con ggplot2:

Recursos:
Purrr avanzado:

- http://modern-rstats.eu/functional-programming.html
- https://www.youtube.com/watch?v=EGAs7zuRutY
- https://www.youtube.com/watch?v=b5Mt4Kzp3BU
- https://www.tidyverse.org/blog/2023/05/purrr-walk-this-way/
- https://www.youtube.com/watch?v=-zZNrGM5aZ4

Introducción ggplot2:

- https://ggplot2-book.org/introduction
- https://rpubs.com/arvindpdmn/ggplot2-basics

Gramática de gráficos.

1. Creación de diferentes tipos de gráficos (barras, líneas, puntos).
2. Personalización de gráficos (escalas, temas, leyendas).

Recursos:

- https://www.youtube.com/watch?v=HPJn1CMvtmI&list=PLtL57Fdbwb_C6RS0JtBojTNOMVlgpeJkS
- https://ggplot2-book.org/getting-started

Actividad

En esta actividad usará el conjunto de datos mtcars para realizar un análisis detallado del rendimiento de los automóviles. La actividad se centrará en el uso de purrr para aplicar funciones de dplyr de manera eficiente a diferentes subconjuntos de datos.

Pasos de la Actividad:

Exploración del Conjunto de Datos mtcars:

- Cargar el conjunto de datos mtcars y realizar una exploración inicial para familiarizarse con sus variables.

Preparación de Subconjuntos de Datos:

- Crear subconjuntos de datos basados en características seleccionadas (por ejemplo, agrupar por número de cilindros o por tipo de transmisión).

Análisis de Datos con purrr y dplyr:

- Utilizar purrr para aplicar funciones de dplyr a cada subconjunto de datos. Por ejemplo, calcular el promedio de millas por galón (mpg), potencia (hp) y otras métricas por grupo.
- Emplear funciones como map o map_df para realizar estas operaciones de manera eficiente.

Visualización de Resultados con ggplot2:

- Crear visualizaciones comparativas utilizando ggplot2 para representar los resultados del análisis, como gráficos de barras o diagramas de cajas.

Interpretación y Discusión:

- Analizar e interpretar los resultados obtenidos.
- Discutir cómo la combinación de purrr y dplyr facilita el análisis de datos y la generación de insights.

Documentación y Presentación:

- Documentar el proceso de análisis y los códigos utilizados.
- Preparar un informe o presentación que resuma los hallazgos y las técnicas utilizadas.

Visualización Avanzada y Espacial

Visualización Avanzada con highcharter:

1. Introducción a highcharter.
2. Creación de gráficos interactivos.

Recursos:

- https://rstudio-pubs-static.s3.amazonaws.com/320413_6ab300527e8548b1a3cbd0d4c6200fcc.html
- https://jkunst.com/highcharter/articles/first-steps.html
- https://jkunst.com/highcharter/articles/showcase.html

Actividad:
En esta actividad examinará y replicará un gráfico de Highcharts creado en JavaScript, luego lo replicarán en R utilizando highcharter. Posteriormente, ampliarán esta comparación creando otros tipos de gráficos en ambos lenguajes.

Ejemplo integración de código JS de pie en highcharts :
Código js:

```javascript
// Data retrieved from https://netmarketshare.com/
// Make monochrome colors
const colors = Highcharts.getOptions().colors.map((c, i) =>
        // Start out with a darkened base color (negative brighten), and end
        // up with a much brighter color
        Highcharts.color(Highcharts.getOptions().colors[0])
        .brighten((i - 3) / 7)
        .get()
);

// Build the chart
Highcharts.chart('container', {
        chart: {
        plotBackgroundColor: null,
        plotBorderWidth: null,
        plotShadow: false,
        type: 'pie'
        },
        title: {
        text: 'Browser market shares in February, 2022',
        align: 'left'
        },
        tooltip: {
        pointFormat: '{series.name}: <b>{point.percentage:.1f}%</b>'
        },
        accessibility: {
        point: {
        valueSuffix: '%'
        }
        },
        plotOptions: {
        pie: {
        allowPointSelect: true,
        cursor: 'pointer',
```

```
        colors,
        borderRadius: 5,
        dataLabels: {
        enabled: true,
        format: '<b>{point.name}</b><br>{point.percentage:.1f} %',
        distance: -50,
        filter: {
               property: 'percentage',
               operator: '>',
               value: 4
        }
        }
        }
        },
        series: [{
        name: 'Share',
        data: [
        { name: 'Chrome', y: 74.03 },
        { name: 'Edge', y: 12.66 },
        { name: 'Firefox', y: 4.96 },
        { name: 'Safari', y: 2.49 },
        { name: 'Internet Explorer', y: 2.31 },
        { name: 'Other', y: 3.398 }
        ]
        }]
});
```

Cómo se construye en Código R:

```Python
library(highcharter)

highchart() |>
 hc_chart(
        plotBackgroundColor = NULL,
        plotBorderWidth = NULL,
        plotShadow = FALSE,
        type = 'pie'
 ) |>
 hc_title(
        text = 'Browser market shares in February, 2022',
        align = 'left'
 ) |>
 hc_tooltip(
```

```
      pointFormat = '{series.name}: <b>{point.percentage:.1f}%</b>'
) |>
hc_plotOptions(
      pie = list(
      allowPointSelect = TRUE,
      cursor = 'pointer',
      borderRadius = 5,
      dataLabels = list(
      enabled = TRUE,
      format = '<b>{point.name}</b><br>{point.percentage:.1f} %',
      distance = -50,
      filter = list(
      property = 'percentage',
      operator = '>',
      value = 4
      )
      )
      )
) |>
hc_add_series(
      name = 'Share',
      data = list(
      list(name = 'Chrome', y = 74.03),
      list(name ='Edge', y = 12.66),
      list(name ='Firefox', y = 4.96),
      list(name ='Safari', y = 2.49),
      list(name ='Internet Explorer', y = 2.31),
      list(name ='Other', y = 3.398 )
      )
)
```

Pasos de la Actividad:

Análisis del Código en JavaScript y R:

- Examinar y entender los códigos de ejemplo proporcionados, centrándose en la estructura y sintaxis de cada uno.

Replicación y Comparación:

- Replicar el gráfico de Highcharts en R usando highcharter.
- Comparar los procesos de creación de gráficos en ambos lenguajes. Puede hacerlo en docs de google, en Rmarkdown o en Quarto.

Creación de Otros Gráficos:

- Extender la actividad creando gráficos de barras, líneas y treemap en ambos JavaScript y R. Para esto ingrese a los demos de highcharts, seleccione el gráfico que desea y en la parte inferior de este hay una serie de opciones para extraer el código, diríjase a “Edit in:” jsfiddle.

- Compare las diferencias en la implementación y visualización.

Documentación y Discusión:
- Documentar el proceso y las observaciones.
- Discutir las ventajas y desafíos de cada enfoque.

Visualización de Datos Espaciales con leaflet:

1. Fundamentos de datos geoespaciales.
2. Creación de mapas interactivos con leaflet.

Recursos:
- https://rubenfcasal.github.io/estadistica_espacial/intro-estesp.html
- https://r-spatial.github.io/sf/articles/sf1.html
- https://rstudio.github.io/leaflet/
- https://bookdown.org/nicohahn/making_maps_with_r5/docs/leaflet.html

Actividad
En esta actividad, utilizará un archivo topojson de la división departamental de Colombia disponible en la carpeta de anexos del repositorio. Deberán generar o simular un conjunto de datos relevante y luego crear visualizaciones geoespaciales utilizando diferentes técnicas y estilos con el paquete leaflet.

Pasos de la Actividad:

Exploración del Archivo Topojson:
- Acceder y explorar el archivo topojson de la división departamental de Colombia disponible en la carpeta de anexos.

Preparación de Datos:
- Generar, buscar o simular un conjunto de datos que pueda ser georreferenciado a las divisiones departamentales de Colombia. Por ejemplo, datos demográficos, ambientales o económicos.

Creación de un Mapa Coroplético:
- Utilizar leaflet para crear un mapa coroplético que represente los datos seleccionados, asignando diferentes colores a las divisiones departamentales según los valores de los datos.
- Experimentar con distintas paletas de colores y estilos para mejorar la visualización.

Creación de Mapas de Puntos:
- Elaborar mapas de puntos para representar los datos, incluyendo una versión con clustering y otra sin él.
- Ajustar opciones como el tamaño y color de los puntos, y añadir tooltips informativos.

Personalización y Experimentación:

- Jugar con diferentes opciones que ofrece leaflet, como tiles (fondos de mapa), tipografías, y controles interactivos.
- Crear varias versiones estéticas de los mapas, explorando diferentes estilos y configuraciones.

Documentación y Presentación:

- Documentar el proceso de creación de los mapas, incluyendo las decisiones tomadas en cuanto a diseño y funcionalidad.
- Preparar una presentación o galería que muestre las diferentes versiones de los mapas creados, explicando las técnicas y estilos utilizados en cada uno.

Introducción a Shiny para Aplicaciones Web Interactivas

Fundamentos de Shiny:

1. Conceptos básicos de Shiny y su arquitectura (UI y server).
2. Creación de una aplicación Shiny simple (input, output, reactivity).

Recursos:

- https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html
- https://www.youtube.com/watch?v=JgQGuWrWcF8&t=2s
- https://www.youtube.com/watch?v=glHoaeNzzTU
- https://www.youtube.com/watch?v=cqOUpnF-Lco

Construcción de Interfaces de Usuario en Shiny:

1. Personalización de la interfaz de usuario.
2. Uso de widgets y controles (sliders, botones, dropdowns).

Recursos:

- https://mastering-shiny.org/action-dynamic.html
- https://shiny.posit.co/r/articles/build/css/
- https://www.youtube.com/watch?v=GZmzj8a4liY

Actividad:

Para esta actividad se utilizará un conjunto de datos proporcionado en la carpeta de anexos del repositorio, que contiene información sobre casos de feminicidios ocurridos en Colombia

en 2017. La tarea consiste en desarrollar una aplicación Shiny que permita a los usuarios filtrar, explorar y visualizar estos datos de manera interactiva.

Pasos de la Actividad:

Exploración y Preparación de los Datos:

- Acceder y explorar la base de datos de feminicidios proporcionada.
- Realizar cualquier limpieza o transformación de datos necesaria para facilitar su análisis y visualización.

Planificación de la Aplicación Shiny:

- Diseñar un esquema para la aplicación, definiendo las funcionalidades clave, como filtros y opciones de visualización. (subir foto, esquema o imagen que se pueda ver en formato jpg, pdf o png)
- Decidir qué tipos de gráficos y tablas serán más efectivos para presentar los datos.

Desarrollo de la Interfaz de Usuario (UI):

- Crear una interfaz de usuario en Shiny que sea intuitiva y fácil de navegar.
- Incluir elementos de UI como barras de navegación, sliders, botones de selección y campos de entrada para filtros.

Implementación del Servidor Shiny:

- Programar el servidor Shiny para manejar la lógica de la aplicación, asegurando que los filtros y visualizaciones respondan adecuadamente a las interacciones del usuario.

Creación de Visualizaciones Interactivas:

- Integrar visualizaciones dinámicas y reactivas que cambien en función de los filtros aplicados por los usuarios.
- Utilizar paquetes como ggplot2, highcharter o leaflet para gráficos y mapas.

Pruebas y Mejoras:

- Probar la aplicación exhaustivamente para identificar y corregir errores o problemas de usabilidad.
- Ajustar y mejorar la aplicación basándose en el feedback de las pruebas.

Documentación y Presentación:

- Documentar el proceso de desarrollo, incluyendo decisiones de diseño y problemas encontrados.
- Preparar una presentación final de la aplicación, destacando sus características y funcionalidades.

Eso es todo por ahora, exitos 🙂