

## Análisis de Resultados al Ejecutar las Instrucciones del Tidyverse

El análisis se basa en los datos de uso de ciclovías en la Ciudad de México durante el mes de febrero de 2020. El objetivo principal es procesar, limpiar y analizar estos datos utilizando el conjunto de herramientas del paquete tidyverse en R. A continuación, se presenta un análisis detallado de los resultados obtenidos en cada etapa del procesamiento de datos.

### 1. Extracción de Datos

```
# Establecer el directorio de trabajo  
setwd("c:/data")  
getwd()
```

```
# Leer el archivo CSV  
raw_data <- read_csv(str_c(getwd(), "/2020-02.csv"))
```

Análisis:

- Carga de Datos: Se establece el directorio de trabajo y se carga el archivo CSV que contiene los registros de uso de ciclovías.
- Formato de Datos: La función read\_csv del paquete readr se utiliza para leer el archivo CSV de manera eficiente, importando los datos en un formato adecuado para su posterior manipulación.

### 2. Transformación de Datos

```
# Convertir columnas de fecha y hora a formato datetime  
viajes_diarios <- raw_data %>%  
  mutate(fecha_hora = dmy_hms(paste(Fecha_Retiro, Hora_Retiro, sep=" ")))
```

Análisis:

- Conversión de Fechas: Utilizando la función mutate de dplyr, se crea una nueva columna fecha\_hora combinando las columnas de fecha y hora en un solo objeto datetime. La función dmy\_hms de lubridate asegura que las fechas y horas se interpreten correctamente.
- Facilidad de Manipulación: Este paso es crucial para facilitar el análisis temporal de los datos, permitiendo filtrar, agrupar y resumir los datos por periodos específicos de tiempo.

### 3. Filtrado de Datos

```
# Filtrar datos entre el 24 y 27 de febrero
viajes_diarios <- viajes_diarios %>%
  filter(fecha_hora >= as.Date('2020-02-24') & fecha_hora <= as.Date('2020-02-27'))
```

Análisis:

- Selección de Periodo: Se filtran los datos para enfocarse en un periodo específico (24-27 de febrero), lo que permite realizar un análisis más detallado y enfocado en días específicos.
- Relevancia Temporal: Este tipo de filtrado es esencial para estudiar comportamientos en periodos determinados y para análisis comparativos con otros periodos.

### 4. Agrupación y Resumen de Datos

```
# Agrupar y resumir datos por hora
viajes_diarios <- viajes_diarios %>%
  group_by(horas = floor_date(fecha_hora, unit = 'hour')) %>%
  summarise(conteo = n())
```

Análisis:

- Agrupación Temporal: Los datos se agrupan por hora utilizando floor\_date de lubridate, lo que permite sumarizar los viajes por cada hora.
- Conteo de Viajes: La función summarise de dplyr se utiliza para contar el número de registros (viajes) en cada grupo (hora).
- Identificación de Patrones: Este paso facilita la identificación de patrones horarios en el uso de las ciclovías, proporcionando información valiosa sobre los picos de uso y las horas de menor actividad.

### 5. Compleción de Datos y Manejo de NAs

```
# Crear estructura de datos con horario completo
horas_completas <- data.frame(
  horas = seq(
    floor_date(min(viajes_diarios$horas), unit='hour'),
    floor_date(max(viajes_diarios$horas), unit='hour'),
    by = 'hour'
  )
)
```

```
# Completar los datos con conteo y asignar ceros donde no hay registros
viajes_horas <- horas_completas %>%
  left_join(viajes_diarios, by = c("horas" = "horas")) %>%
  mutate(conteo = ifelse(is.na(conteo), 0, conteo))
```

Análisis:

- **Compleción de Secuencia:** Se crea una secuencia completa de horas dentro del periodo de interés, asegurando que no haya huecos en la serie temporal.
- **Asignación de Ceros:** Mediante `left_join` y `mutate`, se completan los datos faltantes asignando un conteo de cero a las horas sin registros. Esto es crucial para la consistencia del análisis de series temporales.

## 6. Visualización de Datos

```
# Graficar el patrón generado para los días seleccionados
ggplot(data = viajes_horas, aes(x = horas, y = conteo)) +
  geom_line() +
  geom_point() +
  labs(title = "Patrón de Uso de Ciclovías del 24 al 27 de Febrero de 2020",
       x = "Horas", y = "Conteo de Viajes") +
  theme_minimal()
```

Análisis:

- **Visualización de Patrones:** Utilizando `ggplot2`, se crea una gráfica que muestra el conteo de viajes por hora. Esta visualización ayuda a identificar claramente los picos y valles en el uso de ciclovías.
- **Comunicación Efectiva:** Las gráficas permiten una comunicación más efectiva de los resultados, haciendo evidente los patrones de uso y facilitando la toma de decisiones informadas.

## 7. Modelado y Predicción

```
# Crear una serie temporal de los datos
conteo_ts <- ts(viajes_horas$conteo, start = 1, frequency = 24)

# Aplicar el modelo ARIMA
ajuste <- auto.arima(y = conteo_ts)
```

```
# Graficar la predicción del modelo
predicciones <- forecast(ajuste)
autoplot(predicciones) +
  labs(title = "Predicción del Modelo ARIMA",
        x = "Horas", y = "Conteo de Viajes") +
  theme_minimal()
```

#### Análisis:

- **Modelado Temporal:** La creación de una serie temporal con ts y el ajuste de un modelo ARIMA con auto.arima de forecast permite realizar predicciones basadas en los datos históricos.
- **Predicción Visual:** La gráfica resultante de la predicción muestra cómo se espera que varíe el uso de las ciclovías, proporcionando una herramienta valiosa para la planificación futura.
- **Validación del Modelo:** Comparando las predicciones con los datos reales se puede evaluar la precisión del modelo, permitiendo ajustes y refinamientos.

## Conclusiones

- **Eficiencia del Tidyverse:** Las herramientas del tidyverse (dplyr, ggplot2, lubridate, readr) ofrecen un conjunto potente y cohesivo para la manipulación, limpieza y análisis de datos.
- **Transformaciones Claras:** Cada paso de transformación es claro y transparente, facilitando el seguimiento y la replicación del análisis.
- **Datos Completos y Limpios:** La capacidad de completar y limpiar datos faltantes asegura la integridad y consistencia del análisis de series temporales.
- **Visualización y Comunicación:** Las capacidades de visualización de ggplot2 permiten comunicar de manera efectiva los resultados del análisis, apoyando la toma de decisiones basada en datos.
- **Modelado Predictivo:** La integración con paquetes como forecast permite no solo analizar datos históricos sino también realizar predicciones futuras, aportando un valor añadido significativo al análisis de datos.