

Operation	Tidyverse: library(tidyverse)	Pandas: import pandas as pd	SQL
Read CSV	<pre>df <- read_csv("path/to/file.csv")</pre>	<pre>df = pd.read_csv("path/to/file.csv")</pre>	<pre>LOAD DATA INFILE 'path/to/file.csv' INTO TABLE table_name FIELDS TERMINATED BY ',' ENCLOSED BY '' LINES TERMINATED BY '\n' IGNORE 1 ROWS;</pre>
Print first k rows	<pre>head(df, k)</pre>	<pre>df.head(k)</pre>	<pre>SELECT * FROM table_name LIMIT k;</pre>
Dimensions	<pre>dim(df)</pre>	<pre>df.shape</pre>	<pre>SELECT COUNT(*), COUNT(DISTINCT column_name) FROM table_name;</pre>
Datatype	<pre>str(df)</pre>	<pre>df.dtypes</pre>	<pre>DESCRIBE table_name;</pre>
Filter Data	<pre>df_filtered <- df %>% filter(condition)</pre>	<pre>df_filtered = df[df["column_name"] condition]</pre>	<pre>SELECT * FROM table_name WHERE condition;</pre>
Select columns	<pre>df_selected <- df %>% select(column1, column2, column3)</pre>	<pre>df_selected = df[["column1", "column2", "column3"]]</pre>	<pre>SELECT column1, column2, column3 FROM table_name;</pre>
Sort	<pre>df_sorted <- df %>% arrange(column1, desc(column2))</pre>	<pre>df_sorted = df.sort_values(["column1", "column2"], ascending=[True, False])</pre>	<pre>SELECT * FROM table_name ORDER BY column1 ASC, column2 DESC;</pre>
Fill NaN	<pre>df_filled <- df %>% mutate(column = ifelse(is.na(column), replacement_value, column))</pre>	<pre>df_filled = df.fillna(replacement_value)</pre>	<pre>UPDATE table_name SET column = replacement_value WHERE column IS NULL;</pre>
Join	<pre>df_joined <- df1 %>% Inner_join(df2, by = "common_column")</pre>	<pre>df_joined = pd.merge(df1, df2, on="common_column", how="inner")</pre>	<pre>SELECT *FROM table1 INNER JOIN table2 ON table1.common_column = table2.common_column;</pre>
Concatenate	<pre>df_concatenated <- bind_rows(df1, df2)</pre>	<pre>df_concatenated = pd.concat([df1, df2], axis=0)</pre>	<pre>SELECT * FROM table1 UNION ALL SELECT * FROM table2;</pre>
Group	<pre>df_grouped <- df %>% group_by(column1, column2) %>% summarize(mean_column3=mean(column3))</pre>	<pre>df_grouped = df.groupby(["column1", "column2"]).agg({ "column3": "mean" }).reset_index()</pre>	<pre>SELECT column1, column2, AVG(column3) AS mean_column3 FROM table_name GROUP BY column1, column2;</pre>
Unique values	<pre>unique(df\$column)</pre>	<pre>df["column"].unique()</pre>	<pre>SELECT DISTINCT column FROM table_name;</pre>
Rename Column	<pre>df_renamed <- df %>% rename(new_column_name = old_column_name)</pre>	<pre>df_renamed = df.rename(columns= { "old_column_name": "new_column_name" })</pre>	<pre>ALTER TABLE table_name CHANGE old_column_name new_column_name data_type;</pre>
Delete column	<pre>df_deleted <- df %>% select(-column_to_delete)</pre>	<pre>df_deleted = df.drop("column_to_delete", axis=1)</pre>	<pre>ALTER TABLE table_name DROP COLUMN column_to_delete;</pre>