

# SETH L.U.J and SIR M.V COLLEGE

## PRAC 11 12 13 14 15

AIM:- 11 Reshaping data using pivot\_longer()/pivot\_wider() (R).

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Displays R code for reshaping student marks data. The code includes reading a CSV file, creating a wide matrix, and then pivoting it into long format.
- Environment View:** Shows objects created during the session: category\_df, df, long\_df, marks\_df, and wide\_df.
- File Explorer:** Shows the project structure with files like "Data", "Data.RData", "Data.csv", "Data.Rmd", and "Data.Rproj".
- Task View:** Shows various packages loaded: tidyverse, dplyr, purrr, readr, rlang, tibble, and haven.
- Help:** Shows the help page for the "pivot\_longer" function.

```
R> # --- 1. long Format (pivot_longer) ----
R> print(head(long_df, 10)) # show first 10 rows of marks per student
R>
R> # 2. PIVOT_WIDER: spread back to wide format
R> wide_df = long_df %>%
R>   pivot_wider(
R>     names_from = Metric,
R>     values_from = Value
R>   )
R>
R> # --- 3. long format (pivot_longer) ----
R> print(head(wide_df, 10))
R>
R> # --- 4. PIVOT_WIDER: stack study metrics and marks into long format
R> library(dplyr)
R> pivot_longer(
R>   cols = c(maths_courses, time_study, marks),
R>   names_to = "Metric",
R>   values_to = "Value"
R> )
R>
R> # --- 5. long format (pivot_longer) ---
```

# **SETH L.U.J and SIR M.V COLLEGE**

## **PRAC 11 12 13 14 15**

# **SETH L.U.J and SIR M.V COLLEGE**

## **PRAC 11 12 13 14 15**

The screenshot shows the RStudio interface with the following details:

- Top Bar:** Studio, File, Edit, Code, View, Plots, Session, Build, Debug, Tools, Help.
- Left Panel:** Environment, Global Options, Help, Address.
- Code Editor:** Contains R code for generating a long format dataset, creating a student ID, and pivoting data by number\_courses and Marks.
- Console:** Displays the output of the R code, including the generated dataset and the pivoted data frame.
- Data View:** Shows the structure of the data frames: category\_id, id, long\_df, marks\_df, student\_id, student\_marks, and while\_df.
- File Explorer:** Shows the project structure with files like Admissions, Whistler, and various RDS files.

## 12 Combining datasets vertically (concatenation) using rbind() (R).

**AIM:- 13** Identifying and handling duplicates using distinct() (R).

## **OUTPUT:-**

# **SETH L.U.J and SIR M.V COLLEGE**

## **PRAC 11 12 13 14 15**

# **SETH L.U.J and SIR M.V COLLEGE**

## **PRAC 11 12 13 14 15**

## 14 Extracting date components using lubridate:: functions (R).

## **OUTPUT:-**

# **SETH L.U.J and SIR M.V COLLEGE**

## **PRAC 11 12 13 14 15**

The screenshot shows a Jupyter Notebook interface with several cells of Python code and their corresponding outputs.

```
# Data Extraction
# Import required libraries
import pandas as pd
from datetime import datetime, timedelta
from dateutil import relativedelta

# Set the date range
start_date = "2023-01-01"
end_date = "2024-12-31"

# Create a date range
date_range = pd.date_range(start=start_date, end=end_date)

# Create a DataFrame
dates_df = pd.DataFrame(date_range, columns=['Date'])

# Extract year, month, day, and weekday
dates_df['Year'] = dates_df['Date'].dt.year
dates_df['Month'] = dates_df['Date'].dt.month
dates_df['Day'] = dates_df['Date'].dt.day
dates_df['Weekday'] = dates_df['Date'].dt.weekday_name

# Add quarter and day of year
dates_df['Quarter'] = dates_df['Date'].dt.quarter
dates_df['Day_of_Year'] = dates_df['Date'].dt.dayofyear

# Print extracted data components
print("---- Data with Extracted Date Components ----")
print(dates_df[['Year', 'Month', 'Day', 'Weekday', 'Quarter', 'Day_of_Year']])

# Processed Data
# Create a copy of the dates DataFrame
processed_data = dates_df.copy()

# Extract year, month, day, and weekday
processed_data['ActualDate'] = pd.to_datetime(processed_data['Date'])
processed_data['Year'] = year(processed_data['ActualDate'])
processed_data['Month'] = month(processed_data['ActualDate'])
processed_data['Day'] = day(processed_data['ActualDate'])
processed_data['Weekday'] = weekday(processed_data['ActualDate'], label=True)
processed_data['Quarter'] = quarter(processed_data['ActualDate'])
processed_data['Day_of_Year'] = dayofyear(processed_data['ActualDate'])

# Print processed data components
print("---- Data with Processed Date Components ----")
print(processed_data[['ActualDate', 'Year', 'Month', 'Day', 'Weekday', 'Quarter', 'Day_of_Year']])

# Current Time Extraction
# Get current time
current_time = datetime.now()

# Print current time extraction
print("---- Current Time Extraction ----")
print(f"Current Year: {current_time.year}")
print(f"Current Month: {current_time.month}")
print(f"Current Day: {current_time.day}")
print(f"Current Weekday: {current_time.strftime('%A')}")
print(f"Current Quarter: {current_time.quarter}")
print(f"Current Day of Year: {current_time.dayofyear}")

# Current Time Correction
# Print current time correction
print("---- Current Time Correction ----")
print(f"Current Year: {current_time.year}")
print(f"Current Month: {current_time.month}")
print(f"Current Day: {current_time.day}")
print(f"Current Weekday: {current_time.strftime('%A')}")
print(f"Current Quarter: {current_time.quarter}")
print(f"Current Day of Year: {current_time.dayofyear}")

# Current Time
# Print current time
print("---- Current Time ----")
print(f"Current Year: {current_time.year}")
print(f"Current Month: {current_time.month}")
print(f"Current Day: {current_time.day}")
print(f"Current Weekday: {current_time.strftime('%A')}")
print(f"Current Quarter: {current_time.quarter}")
print(f"Current Day of Year: {current_time.dayofyear}")
print(f"Current Hour: {current_time.hour}")
print(f"Current Minute: {current_time.minute}")
print(f"Current Second: {current_time.second}
```

## 15 Generating basic summaries using str() or summary() (R).

## **OUTPUT:-**

# **SETH L.U.J and SIR M.V COLLEGE**

## **PRAC 11 12 13 14 15**

The screenshot shows a Windows desktop environment with several open windows. In the foreground, a RStudio session is active, displaying a Shiny application. The application's UI includes a sidebar with 'Home', 'About', and 'Background App' buttons, and a main panel with a 'Help' button and a 'Dependencies' section. The code editor shows R code for package loading and a 'dependencies' function. The console output shows the creation of a 'data' object and its contents. The bottom pane displays a table of data with columns 'id', 'Category', 'Min', 'Mean', 'Q1', 'Median', 'Max', and 'Rating'. A second RStudio window is visible in the background, showing a file tree with various projects and files.

```
knitr -- RStudio Session
```

The R session window shows the following code and its output:

```
library(tidyverse)
# Load data
airline_fares <- read_csv("airline_fares.csv")
# Summary statistics
summary(airline_fares)
```

The output shows the following summary statistics for the airline\_fares dataset:

Category	Min	Q1	Median	Q3	Max
origin	ATL	ATL	ATL	ATL	ATL
destination	BNA	BNA	BNA	BNA	BNA
airline	AirTran	AirTran	AirTran	AirTran	AirTran
class	Y	Y	Y	Y	Y
month	1	1	1	1	12
year	2010	2010	2010	2010	2010
length	1	1	1	1	1
price	100	100	100	100	1000
origin_d	ATL	ATL	ATL	ATL	ATL
destination_d	BNA	BNA	BNA	BNA	BNA
airline_d	AirTran	AirTran	AirTran	AirTran	AirTran
class_d	Y	Y	Y	Y	Y
month_d	1	1	1	1	12
year_d	2010	2010	2010	2010	2010
length_d	1	1	1	1	1
price_d	100	100	100	100	1000

Next, the user performs a group\_by operation on the 'origin' column and calculates the average price:

```
# Group by origin and calculate average price
airline_fares %>% group_by(origin) %>% summarise(price = mean(price))
```

The output shows the average price for each origin:

origin	price
ATL	100
BNA	100

Finally, the user performs a group\_by operation on both 'origin' and 'destination' columns and calculates the average price:

```
# Group by origin and destination and calculate average price
airline_fares %>% group_by(origin, destination) %>% summarise(price = mean(price))
```

The output shows the average price for each combination of origin and destination:

origin	destination	price
ATL	BNA	100