

SHETH LUJ AND SIR MV COLLEGE  
Subject: Data Analysis with SAS / SPSS /R

Practical No: 9

Aim: Performing text manipulation using `str_sub()`, `str_split()` (R). import dataset.

Code:

```
# R Script: Text Manipulation with stringr (Adapted for ESGCountry.csv)
library(stringr)
library(tidyr)
library(dplyr)

# 1. IMPORT DATASET

country_df <- read.csv("ESGCountry.csv")

country_df <- country_df %>%
  rename(alpha_2_code = X2.alpha.code)

print("--- Original Dataset (Key Columns) ---")
print(head(country_df, c("Country.Code", "Long.Name", "Latest.household.survey")), 5))
```

# 2. USING `str_sub()` (Substring)

```
country_df$Prefix_5 <- str_sub(country_df$Long.Name, 1, 5)

country_df$Suffix_5 <- str_sub(country_df$Long.Name, -5, -1)

print("--- Data after str_sub() ---")
print(country_df %>% select(Long.Name, Prefix_5, Suffix_5) %>% head(5))
```

# 3. USING `str_split()` (Split String)

```
# Method B: Split Fixed (Returns a matrix, easier to assign to columns)
split_matrix <- str_split(country_df$Latest.household.survey, ", ", simplify = TRUE)

country_df$Survey_Type <- split_matrix[, 1] # Text before the comma
country_df$Survey_Detail <- split_matrix[, 2] # Text after the comma

print("--- Data after str_split() (Manual Assignment) ---")
print(country_df %>% select(Latest.household.survey, Survey_Type, Survey_Detail) %>%
  head(5))
```

**SHETH LUJ AND SIR MV COLLEGE**  
**Subject: Data Analysis with SAS / SPSS / R**

## # 4. BONUS: The "Tidy" Way (separate)

```
tidy_data <- country_df %>%
  separate(Long.Name,
    into = c("Status_1", "Status_2", "Rest_of_Name"),
    sep = " ",
    extra = "merge", # Merges all remaining parts into the last column
    remove = FALSE) # Keep the original Long.Name column

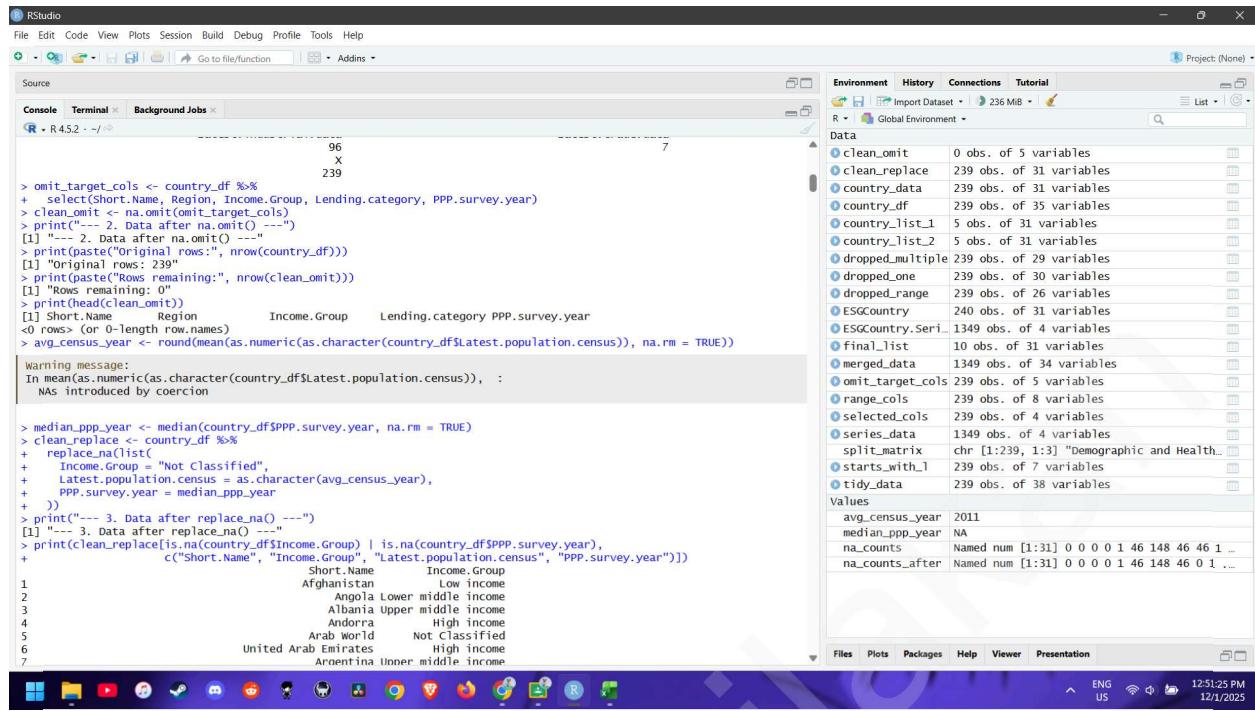
print("--- Bonus: The 'separate' function (easier splitting) ---")
print(tidy_data %>% select(Long.Name, Status_1, Status_2, Rest_of_Name) %>% head(5))
```

## Output:

The screenshot shows an RStudio interface with several open windows and tabs.

- File**, **Edit**, **Code**, **View**, **Plots**, **Session**, **Build**, **Debug**, **Profile**, **Tools**, **Help**
- Console** tab: Displays R code and its output. The session starts with `library(dplyr)` and `library(tidyverse)`. It reads a CSV file named "ESGCountry.csv" and renames columns. The `country\_df` data frame is then printed, showing data for countries like Afghanistan, Angola, and Albania across various regions and income groups. A note indicates the data is from administrative registers. The session also prints the count of missing values per column.
- Source** tab: Shows the R code used in the console.
- Environment** pane: Lists global variables and their details, such as `alpha\_2\_code` (1 obs. of 2 variables), `Special.Notes` (148 obs. of 1 variables), and `Income.Group` (46 obs. of 1 variables). It also lists data frames like `National.accounts.base.year` (192 obs. of 1 variables) and `Alternative.conversion.factor` (192 obs. of 1 variables).
- History** pane: Shows a history of previous R commands entered.
- Connections** pane: Shows connections to databases or other systems.
- Tutorial** pane: Provides access to R documentation and help resources.
- Project** pane: Shows the current project structure, including "Global Environment".
- Files**, **Plots**, **Packages**, **Help**, **Viewer**, **Presentation** tabs at the bottom.

**SHETH LUJ AND SIR MV COLLEGE**  
**Subject: Data Analysis with SAS / SPSS / R**



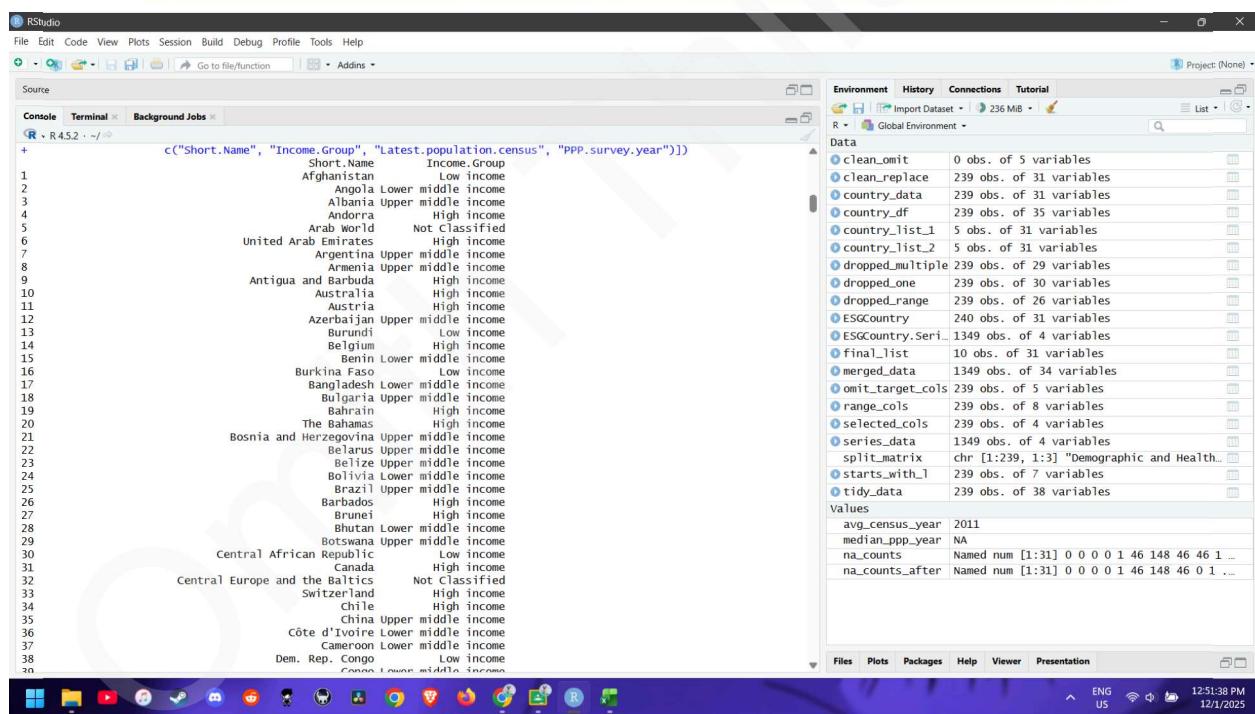
```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console Terminal Background Jobs
R > 4.5.2 ~/ ...
96
X
239
> omit_target_cols <- country_df %>%
+   select(Short.Name, Region, Income.Group, Lending.category, PPP.survey.year)
> clean.omit <- na.omit(omit_target_cols)
> print("--- 2. Data after na.omit() ---")
[1] "--- 2. Data after na.omit() ---"
> print(paste("Original rows:", nrow(country_df)))
[1] "Original rows: 239"
> print(paste("Rows remaining:", nrow(clean.omit)))
[1] "Rows remaining: 0"
> print(head(clean.omit))
[1] Short.Name      Region      Income.Group      Lending.category PPP.survey.year
<0 rows> (or 0-length row.names)
> avg_census_year <- round(mean(as.numeric(as.character(country_df$Latest.population.census))), na.rm = TRUE))
Warning message:
In mean(as.numeric(as.character(country_df$Latest.population.census)), :
  NA introduced by coercion

> median_ppp_year <- median(country_df$PPP.survey.year, na.rm = TRUE)
> clean.replace <- country_df %>%
+   replace_na(list(
+     Income.Group = "Not Classified",
+     Latest.population.census = as.character(avg_census_year),
+     PPP.survey.year = median_ppp_year
+   ))
> print("--- 3. Data after replace_na() ---")
[1] "--- 3. Data after replace_na() ---"
> print(clean.replace[is.na(country_df$Income.Group) | is.na(country_df$Latest.population.census), c("Short.Name", "Income.Group", "Latest.population.census", "PPP.survey.year")])
Short.Name      Income.Group
1 Afghanistan      Low income
2 Angola          Lower middle income
3 Albania          Upper middle income
4 Andorra          High income
5 Arab World        Not Classified
6 United Arab Emirates      High income
7 Argentina         Upper middle income
8 Armenia          Upper middle income
9 Antigua and Barbuda      High income
10 Australia         High income
11 Austria          High income
12 Azerbaijan        Upper middle income
13 Burundi           Low income
14 Belgium           High income
15 Benin             Lower middle income
16 Burkina Faso      Low income
17 Bangladesh        Lower middle income
18 Bulgaria          Upper middle income
19 Bulgaria          Upper middle income
20 The Bahamas       High income
21 Bosnia and Herzegovina Upper middle income
22 Belarus           Upper middle income
23 Belize            Upper middle income
24 Bolivia           Lower middle income
25 Brazil            Upper middle income
26 Barbados          High income
27 Brunei            High income
28 Bhutan            Lower middle income
29 Botswana          Upper middle income
30 Central African Republic      Low income
31 Canada            High income
32 Central Europe and the Balkans      Not Classified
33 Switzerland        High income
34 Chile              High income
35 China              Upper middle income
36 Côte d'Ivoire       Lower middle income
37 Cameroon          Lower middle income
38 Dem. Rep. Congo      Low income
39 Congo              Lower middle income

```

The screenshot shows the RStudio interface. The Source pane contains R code for data cleaning and summarization. The right pane displays the 'Data' environment, showing various objects like 'clean.omit', 'clean.replace', and 'avg\_census\_year'. The status bar at the bottom indicates the system is ENG US and the date is 12/1/2025.



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console Terminal Background Jobs
R > 4.5.2 ~/ ...
c("Short.Name", "Income.Group", "Latest.population.census", "PPP.survey.year"))
Short.Name      Income.Group
1 Afghanistan      Low income
2 Angola          Lower middle income
3 Albania          Upper middle income
4 Andorra          High income
5 Arab World        Not Classified
6 United Arab Emirates      High income
7 Argentina         Upper middle income
8 Armenia          Upper middle income
9 Antigua and Barbuda      High income
10 Australia         High income
11 Austria          High income
12 Azerbaijan        Upper middle income
13 Burundi           Low income
14 Belgium           High income
15 Benin             Lower middle income
16 Burkina Faso      Low income
17 Bangladesh        Lower middle income
18 Bulgaria          Upper middle income
19 Bulgaria          Upper middle income
20 The Bahamas       High income
21 Bosnia and Herzegovina Upper middle income
22 Belarus           Upper middle income
23 Belize            Upper middle income
24 Bolivia           Lower middle income
25 Brazil            Upper middle income
26 Barbados          High income
27 Brunei            High income
28 Bhutan            Lower middle income
29 Botswana          Upper middle income
30 Central African Republic      Low income
31 Canada            High income
32 Central Europe and the Balkans      Not Classified
33 Switzerland        High income
34 Chile              High income
35 China              Upper middle income
36 Côte d'Ivoire       Lower middle income
37 Cameroon          Lower middle income
38 Dem. Rep. Congo      Low income
39 Congo              Lower middle income

```

This screenshot is identical to the one above, showing the same RStudio session with the Source and Data panes visible. The status bar at the bottom indicates the system is ENG US and the date is 12/1/2025.

**SHETH LUJ AND SIR MV COLLEGE**  
**Subject: Data Analysis with SAS / SPSS /R**

```

Source
Console Terminal Background Jobs
R 4.5.2 - / ...
239
> print(head(clean_replace[, c("Short_Name", "Income_Group", "Latest.population.census", "PPP.survey.year")]))
  ...                                     Latest.population.census
1   NA
2 Afghanistan Income_Group Low income          1979
3   Angola Lower middle income           2014
4  Albania Upper middle income          2011
5 Andorra High income 2011. Population data compiled from administrative registers.
6 Arab World Not Classified            2011
7 United Arab Emirates High income          2010
  PPP.survey.year
1   NA
2   NA
3   NA
4   NA
5   NA
6   NA
> print("--- Remaining NAs after replacement ---")
[1] "--- Remaining NAs after replacement ---"
> na_counts <- colSums(is.na(clean_replace))
> print(na_counts[c("Income_Group", "Latest.population.census", "PPP.survey.year")])
  Income_Group Latest.population.census PPP.survey.year
0                  0                   239
>
>
> # R Script: Text Manipulation with stringr (Adapted for ESGCountry.csv)
> library(stringr)
> library(dplyr)
> library(dplyr)
> country_df <- read.csv("ESGCountry.csv")
> country_df %>% 
+   rename(alpha_2_code = X2.alpha.code)
> print("--- Original Dataset (Key Columns) ---")
[1] "--- Original Dataset (Key Columns) ---"
> print(head(country_df, n = 5))
  Country.Code Long_Name           Latest.household.survey
1   AFG Islamic State of Afghanistan Demographic and Health Survey, 2015
2   AGO People's Republic of Angola Demographic and Health Survey, 2015/16
3   ALB       Republic of Albania Demographic and Health Survey, 2017/18
4   AND Principality of Andorra
  Arab_World Arab_World

```

```

Source
Console Terminal Background Jobs
R 4.5.2 - / ...
239
> print(str_split(country_df$Latest.household.survey, ", ", simplify = TRUE))
  ... 
1 People's Republic of Angola Peopl1 ngola
2   Republic of Albania Repub1 bania
3 Principality of Andorra Princ1 dorra
4   Arab World Arab1 world
> # Method B: Split Fixed (Returns a matrix, easier to assign to columns)
> split_matrix <- str_split(country_df$Latest.household.survey, ", ", simplify = TRUE)
> country_df$Survey_Type <- split_matrix[, 1] # Text before the comma
> country_df$Survey_Detail <- split_matrix[, 2] # Text after the comma
> print("--- Data after str_split() (Manual Assignment) ---")
[1] "--- Data after str_split() (Manual Assignment) ---"
> print(country_df %>% select(Latest.household.survey, Survey_Type, Survey_Detail) %>% head(5))
  ... 
1 Demographic and Health Survey, 2015 Demographic and Health Survey, 2015
2 Demographic and Health Survey, 2015/16 Demographic and Health Survey, 2015/16
3 Demographic and Health Survey, 2017/18 Demographic and Health Survey, 2017/18
4 
5
> tidy_data <- country_df %>% 
+   separate(Long_Name,
+     into = c("Status_1", "Status_2", "Rest_of_Name"),
+     sep = ",",
+     extra = "merge", # Merges all remaining parts into the last column
+     remove = FALSE) # Keep the original Long_Name column
Warning message:
Expected 3 pieces. Missing pieces filled with `NA` in 56 rows [5, 7, 16, 23, 26, 27, 31, 33, 47, 52, 55, 61, 66, 70, 72, 74, 80, 81, 84, 89, ...].
> print("--- Bonus: The 'separate' function (easier splitting) ---")
[1] "--- Bonus: The 'separate' function (easier splitting) ---"
> print(tidy_data %>% select(Long_Name, Status_1, Status_2, Rest_of_Name) %>% head(5))
  ... 
1 Islamic State of Afghanistan Islamic State of Afghanistan
2 People's Republic of Angola People's Republic of Angola
3   Republic of Albania Republic of Albania
4 Principality of Andorra Principality of Andorra
5   Arab World Arab World
  <NA>
>
> 

```