

Package ‘MissingHandle’

July 21, 2025

Type Package

Title Handles Missing Dates and Data and Converts into Weekly and Monthly from Daily

Version 0.1.1

Author Mr. Sandip Garai [aut, cre]

Maintainer Mr. Sandip Garai <sandipnicksandy@gmail.com>

Description Many times, you will not find data for all dates. After first January, 2011 you may have next data on 20th January, 2011 and so on. Also available dates may have zero values. Try to gather all such kinds of data in different excel sheets of a single excel file. Every sheet will contain two columns (1st one is dates and second one is the data). After loading all the sheets into different elements of a list, using this you can fill the gaps for all the sheets and mark all the corresponding values as zeros. Here I am talking about daily data. Finally, it will combine all the filled results into one data frame (first column is date and other columns will be corresponding values of your sheets) and give one combined data frame. Number of columns in the data frame will be number of sheets plus one. Then imputation will be done. Daily to monthly and weekly conversion is also possible. More details can be found in Garai and others (2023) <doi:10.13140/RG.2.2.11977.42087>.

License GPL-3

Encoding UTF-8

Imports zoo, imputeTS, dplyr

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2023-04-30 04:20:02 UTC

Contents

clean_and_combine	2
impute_combined	5
monthly_from_daily	8
weekly_from_daily	12

clean_and_combine	<i>Fill Missing Dates and Combine Data into a Data Frame</i>
-------------------	--

Description

Many times, you will not find data for all dates. After first January, 2011 you may have next data on 20th January, 2011 and so on. Also available dates may have zero values. Try to gather all such kinds of data in different excel sheets of a single excel file. Every sheet will contain two columns (1st one is dates and second one is the data). Load every sheet to separate elements of a list. Using this you can fill the gaps for all the sheets and mark all the corresponding values as zeros. Here I am talking about daily data. Finally, it will combine all the filled results into one data frame (first column is date and other columns will be corresponding values of your sheets) and give one csv file. Number of columns in the data frame will be number of sheets plus one.

Usage

```
clean_and_combine(
  my_list,
  starting_date = as.Date("2011-01-01"),
  ending_date = as.Date("2022-12-31"),
  date_format = "%d-%m-%y"
)
```

Arguments

my_list	List of elements containing two columns each. First column is data which may have missing dates and second column is corresponding time series values.
starting_date	From which date data is needed
ending_date	Upto which date data is needed
date_format	Specify the date format of your data

Value

- clean_and_combined_df: Data frame of combined data containing multiple columns. First column is complete dates and others are corresponding values of second column of every element of input list. Missing values are denoted as zeros.

References

- Paul, R. K., & Garai, S. (2021). Performance comparison of wavelets-based machine learning technique for forecasting agricultural commodity prices. *Soft Computing*, 25(20), 12857-12873.
- Paul, R. K., & Garai, S. (2022). Wavelets based artificial neural network technique for forecasting agricultural prices. *Journal of the Indian Society for Probability and Statistics*, 23(1), 47-61.

- Garai, S., & Paul, R. K. (2023). Development of MCS based-ensemble models using CEEM-DAN decomposition and machine intelligence. *Intelligent Systems with Applications*, 18, 200202.
- Garai, S., Paul, R. K., Rakshit, D., Yeasin, M., Paul, A. K., Roy, H. S., Barman, S. & Manjunatha, B. (2023). An MRA Based MLR Model for Forecasting Indian Annual Rainfall Using Large Scale Climate Indices. *International Journal of Environment and Climate Change*, 13(5), 137-150.

Examples

```

# # real data
# # reading excel file into list ####
# file_path <- "excel_file.xlsx"
#
# # get sheet names
# sheet_names <- openxlsx::getSheetNames(file_path)
#
# # create an empty list to store the cleaned data frames
# my_list <- list()
#
# # loop through each sheet and apply the cleaning code
# for (sheet_name in sheet_names) {
#
#   column_types <- c('date', 'numeric')
#
#   date_format <- "%d-%m-%y"
#
#   # Read in the sheet as a data frame
#   data <- readxl::read_excel(file_path, sheet = sheet_name, col_types = column_types)
#
#   # add the cleaned data frame to the list
#   my_list[[sheet_name]] <- as.data.frame(data)
# }

# creating example ####

# 1st element ####
# Create a sequence of dates from "2011-01-01" to "2015-12-31"
dates <- seq(as.Date("2011-01-01"), as.Date("2011-03-31"), by="day")

# Generate random prices for each date
price_1 <- runif(length(dates), min=0, max=100)

# Combine the dates and prices into a data frame
df <- data.frame(Dates = dates, Price_a = price_1)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates2 <- seq(as.Date("2011-05-01"), as.Date("2011-12-31"), by="day")

# Generate random prices for each date
price_2 <- runif(length(dates2), min=0, max=100)

```

```
# Combine the dates and prices into a data frame
df2 <- data.frame(Dates = dates2, Price_a = price_2)

# Merge the two data frames row-wise
df <- rbind(df, df2)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates3 <- seq(as.Date("2012-02-01"), as.Date("2012-12-31"), by="day")

# Generate random prices for each date
price_3 <- runif(length(dates3), min=0, max=100)

# Combine the dates and prices into a data frame
df3 <- data.frame(Dates = dates3, Price_a = price_3)

# Merge the two data frames row-wise
df <- rbind(df, df3)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates4 <- seq(as.Date("2013-04-01"), as.Date("2022-12-31"), by="day")

# Generate random prices for each date
price_4 <- runif(length(dates4), min=0, max=100)

# Combine the dates and prices into a data frame
df4 <- data.frame(Dates = dates4, Price_a = price_4)

# Merge the two data frames row-wise
df <- rbind(df, df4)

# Specify column data types
df <- data.frame(Dates = as.Date(df$Dates),
                 price_a = round(as.numeric(df$Price_a)))
# 2nd element #####
# Create a sequence of dates from "2011-01-01" to "2015-12-31"
dates <- seq(as.Date("2011-01-01"), as.Date("2011-05-31"), by="day")

# Generate random prices for each date
price_1 <- runif(length(dates), min=0, max=100)

# Combine the dates and prices into a data frame
df_second <- data.frame(Dates = dates, Price_b = price_1)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates2 <- seq(as.Date("2011-06-01"), as.Date("2011-10-31"), by="day")

# Generate random prices for each date
price_2 <- runif(length(dates2), min=0, max=100)

# Combine the dates and prices into a data frame
df_second2 <- data.frame(Dates = dates2, Price_b = price_2)

# Merge the two data frames row-wise
```

```

df_second <- rbind(df_second, df_second2)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates3 <- seq(as.Date("2012-01-01"), as.Date("2012-12-31"), by="day")

# Generate random prices for each date
price_3 <- runif(length(dates3), min=0, max=100)

# Combine the dates and prices into a data frame
df_second3 <- data.frame(Dates = dates3, Price_b = price_3)

# Merge the two data frames row-wise
df_second <- rbind(df_second, df_second3)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates4 <- seq(as.Date("2013-03-01"), as.Date("2022-12-31"), by="day")

# Generate random prices for each date
price_4 <- runif(length(dates4), min=0, max=100)

# Combine the dates and prices into a data frame
df_second4 <- data.frame(Dates = dates4, Price_b = price_4)

# Merge the two data frames row-wise
df_second <- rbind(df_second, df_second4)

# Specify column data types
df_second <- data.frame(Dates = as.Date(df_second$Dates),
                       price_b = round(as.numeric(df_second$Price_b)))

# my_list ####
# Create a list
my_list <- list()

# Add the data frame to the list
my_list$df <- df

my_list$df_second <- df_second

# getting output ####
my_combined_data <- clean_and_combine(my_list = my_list)
print(head(my_combined_data))

```

impute_combined

Fill Zeros as NA and Impute

Description

Imputation will be done. It will assign dates from start date to end date in the specified format. Finally, imputed data will be provided.

Usage

```
impute_combined(My_df, method_impute = na_kalman)
```

Arguments

My_df	Data frame with 1st column as dates and others containing missing values denoted as zeros
method_impute	Select imputation method from ImputeTS package

Value

- imputed_df: Data frame of combined imputed data

References

- Paul, R. K., & Garai, S. (2021). Performance comparison of wavelets-based machine learning technique for forecasting agricultural commodity prices. *Soft Computing*, 25(20), 12857-12873.
- Paul, R. K., & Garai, S. (2022). Wavelets based artificial neural network technique for forecasting agricultural prices. *Journal of the Indian Society for Probability and Statistics*, 23(1), 47-61.
- Garai, S., & Paul, R. K. (2023). Development of MCS based-ensemble models using CEEM-DAN decomposition and machine intelligence. *Intelligent Systems with Applications*, 18, 200202.
- Garai, S., Paul, R. K., Rakshit, D., Yeasin, M., Paul, A. K., Roy, H. S., Barman, S. & Manjunatha, B. (2023). An MRA Based MLR Model for Forecasting Indian Annual Rainfall Using Large Scale Climate Indices. *International Journal of Environment and Climate Change*, 13(5), 137-150.

Examples

```
# creating example ####

# 1st element ####
# Create a sequence of dates from "2011-01-01" to "2015-12-31"
dates <- seq(as.Date("2011-01-01"), as.Date("2011-03-31"), by="day")

# Generate random prices for each date
price_1 <- runif(length(dates), min=0, max=100)

# Combine the dates and prices into a data frame
df <- data.frame(Dates = dates, Price_a = price_1)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates2 <- seq(as.Date("2011-05-01"), as.Date("2011-12-31"), by="day")

# Generate random prices for each date
price_2 <- runif(length(dates2), min=0, max=100)
```

```
# Combine the dates and prices into a data frame
df2 <- data.frame(Dates = dates2, Price_a = price_2)

# Merge the two data frames row-wise
df <- rbind(df, df2)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates3 <- seq(as.Date("2012-02-01"), as.Date("2012-12-31"), by="day")

# Generate random prices for each date
price_3 <- runif(length(dates3), min=0, max=100)

# Combine the dates and prices into a data frame
df3 <- data.frame(Dates = dates3, Price_a = price_3)

# Merge the two data frames row-wise
df <- rbind(df, df3)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates4 <- seq(as.Date("2013-04-01"), as.Date("2022-12-31"), by="day")

# Generate random prices for each date
price_4 <- runif(length(dates4), min=0, max=100)

# Combine the dates and prices into a data frame
df4 <- data.frame(Dates = dates4, Price_a = price_4)

# Merge the two data frames row-wise
df <- rbind(df, df4)

# Specify column data types
df <- data.frame(Dates = as.Date(df$Dates),
                price_a = round(as.numeric(df$Price_a)))
# 2nd element #####
# Create a sequence of dates from "2011-01-01" to "2015-12-31"
dates <- seq(as.Date("2011-01-01"), as.Date("2011-05-31"), by="day")

# Generate random prices for each date
price_1 <- runif(length(dates), min=0, max=100)

# Combine the dates and prices into a data frame
df_second <- data.frame(Dates = dates, Price_b = price_1)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates2 <- seq(as.Date("2011-06-01"), as.Date("2011-10-31"), by="day")

# Generate random prices for each date
price_2 <- runif(length(dates2), min=0, max=100)

# Combine the dates and prices into a data frame
df_second2 <- data.frame(Dates = dates2, Price_b = price_2)

# Merge the two data frames row-wise
```

```

df_second <- rbind(df_second, df_second2)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates3 <- seq(as.Date("2012-01-01"), as.Date("2012-12-31"), by="day")

# Generate random prices for each date
price_3 <- runif(length(dates3), min=0, max=100)

# Combine the dates and prices into a data frame
df_second3 <- data.frame(Dates = dates3, Price_b = price_3)

# Merge the two data frames row-wise
df_second <- rbind(df_second, df_second3)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates4 <- seq(as.Date("2013-03-01"), as.Date("2022-12-31"), by="day")

# Generate random prices for each date
price_4 <- runif(length(dates4), min=0, max=100)

# Combine the dates and prices into a data frame
df_second4 <- data.frame(Dates = dates4, Price_b = price_4)

# Merge the two data frames row-wise
df_second <- rbind(df_second, df_second4)

# Specify column data types
df_second <- data.frame(Dates = as.Date(df_second$Dates),
                       price_b = round(as.numeric(df_second$Price_b)))

# my_list #####
# Create a list
my_list <- list()

# Add the data frame to the list
my_list$df <- df

my_list$df_second <- df_second

# getting output #####
my_combined_data <- clean_and_combine(my_list = my_list)
print(head(my_combined_data))
my_imputed_data <- impute_combined(my_combined_data)
print(head(my_imputed_data))

```

monthly_from_daily *Convert Daily Data to monthly*

Description

Converts daily data to monthly data. One needs to specify the month format.

Usage

```
monthly_from_daily(
  my_daily_data,
  starting_date = "2011-01-01",
  ending_date = "2022-12-31",
  year_month_format = "%Y-%m",
  month_ending_format = "%Y-%m-%d",
  month_ending_day = "-1",
  year_month = "year_month",
  month_ending_date = "month_ending_date"
)
```

Arguments

my_daily_data	A data frame containing first column as dates and others are columns contains daily data
starting_date	From which date data is present
ending_date	Upto which date data is present
year_month_format	specify the year month format
month_ending_format	specify month ending format
month_ending_day	corresponding days of a month
year_month	this is a variable, leave this as it is
month_ending_date	name of the first column of the output data frame

Value

- my_monthly_data: Data frame containing converted data into monthly one

References

- Paul, R. K., & Garai, S. (2021). Performance comparison of wavelets-based machine learning technique for forecasting agricultural commodity prices. *Soft Computing*, 25(20), 12857-12873.
- Paul, R. K., & Garai, S. (2022). Wavelets based artificial neural network technique for forecasting agricultural prices. *Journal of the Indian Society for Probability and Statistics*, 23(1), 47-61.
- Garai, S., & Paul, R. K. (2023). Development of MCS based-ensemble models using CEEM-DAN decomposition and machine intelligence. *Intelligent Systems with Applications*, 18, 200202.
- Garai, S., Paul, R. K., Rakshit, D., Yeasin, M., Paul, A. K., Roy, H. S., Barman, S. & Manjunatha, B. (2023). An MRA Based MLR Model for Forecasting Indian Annual Rainfall Using Large Scale Climate Indices. *International Journal of Environment and Climate Change*, 13(5), 137-150.

Examples

```
# creating example #####

# 1st element #####
# Create a sequence of dates from "2011-01-01" to "2015-12-31"
dates <- seq(as.Date("2011-01-01"), as.Date("2011-03-31"), by="day")

# Generate random prices for each date
price_1 <- runif(length(dates), min=0, max=100)

# Combine the dates and prices into a data frame
df <- data.frame(Dates = dates, Price_a = price_1)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates2 <- seq(as.Date("2011-05-01"), as.Date("2011-12-31"), by="day")

# Generate random prices for each date
price_2 <- runif(length(dates2), min=0, max=100)

# Combine the dates and prices into a data frame
df2 <- data.frame(Dates = dates2, Price_a = price_2)

# Merge the two data frames row-wise
df <- rbind(df, df2)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates3 <- seq(as.Date("2012-02-01"), as.Date("2012-12-31"), by="day")

# Generate random prices for each date
price_3 <- runif(length(dates3), min=0, max=100)

# Combine the dates and prices into a data frame
df3 <- data.frame(Dates = dates3, Price_a = price_3)

# Merge the two data frames row-wise
df <- rbind(df, df3)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates4 <- seq(as.Date("2013-04-01"), as.Date("2022-12-31"), by="day")

# Generate random prices for each date
price_4 <- runif(length(dates4), min=0, max=100)

# Combine the dates and prices into a data frame
df4 <- data.frame(Dates = dates4, Price_a = price_4)

# Merge the two data frames row-wise
df <- rbind(df, df4)

# Specify column data types
df <- data.frame(Dates = as.Date(df$Dates),
                price_a = round(as.numeric(df$Price_a)))
```

```
# 2nd element #####
# Create a sequence of dates from "2011-01-01" to "2015-12-31"
dates <- seq(as.Date("2011-01-01"), as.Date("2011-05-31"), by="day")

# Generate random prices for each date
price_1 <- runif(length(dates), min=0, max=100)

# Combine the dates and prices into a data frame
df_second <- data.frame(Dates = dates, Price_b = price_1)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates2 <- seq(as.Date("2011-06-01"), as.Date("2011-10-31"), by="day")

# Generate random prices for each date
price_2 <- runif(length(dates2), min=0, max=100)

# Combine the dates and prices into a data frame
df_second2 <- data.frame(Dates = dates2, Price_b = price_2)

# Merge the two data frames row-wise
df_second <- rbind(df_second, df_second2)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates3 <- seq(as.Date("2012-01-01"), as.Date("2012-12-31"), by="day")

# Generate random prices for each date
price_3 <- runif(length(dates3), min=0, max=100)

# Combine the dates and prices into a data frame
df_second3 <- data.frame(Dates = dates3, Price_b = price_3)

# Merge the two data frames row-wise
df_second <- rbind(df_second, df_second3)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates4 <- seq(as.Date("2013-03-01"), as.Date("2022-12-31"), by="day")

# Generate random prices for each date
price_4 <- runif(length(dates4), min=0, max=100)

# Combine the dates and prices into a data frame
df_second4 <- data.frame(Dates = dates4, Price_b = price_4)

# Merge the two data frames row-wise
df_second <- rbind(df_second, df_second4)

# Specify column data types
df_second <- data.frame(Dates = as.Date(df_second$Dates),
                      price_b = round(as.numeric(df_second$Price_b)))

# my_list #####
# Create a list
my_list <- list()
```

```
# Add the data frame to the list
my_list$df <- df

my_list$df_second <- df_second

# getting output ####
my_combined_data <- clean_and_combine(my_list = my_list)
print(head(my_combined_data))
my_imputed_data <- impute_combined(my_combined_data)
print(head(my_imputed_data))
my_monthly_data <- monthly_from_daily(my_imputed_data)
print(head(my_monthly_data))
```

weekly_from_daily *Convert Daily Data to Weekly*

Description

Converts daily data to weekly data. One needs to specify the week format.

Usage

```
weekly_from_daily(
  my_daily_data,
  starting_date = "2011-01-01",
  ending_date = "2022-12-31",
  year_week_format = "%Y-%W",
  week_ending_format = "%Y-%W-%u",
  week_ending_day = "-7",
  year_week = "year_week",
  week_ending_date = "week_ending_date"
)
```

Arguments

my_daily_data	A data frame containing first column as dates and others are columns contains daily data
starting_date	From which date data is present
ending_date	Upto which date data is present
year_week_format	specify the year week format
week_ending_format	specify week ending format
week_ending_day	corresponding days of a week 7 or 6 days
year_week	this is a variable, leave this as it is
week_ending_date	name of the first column of the output data frame

Value

- my_weekly_data: Data frame containing converted data into weekly one

References

- Paul, R. K., & Garai, S. (2021). Performance comparison of wavelets-based machine learning technique for forecasting agricultural commodity prices. *Soft Computing*, 25(20), 12857-12873.
- Paul, R. K., & Garai, S. (2022). Wavelets based artificial neural network technique for forecasting agricultural prices. *Journal of the Indian Society for Probability and Statistics*, 23(1), 47-61.
- Garai, S., & Paul, R. K. (2023). Development of MCS based-ensemble models using CEEM-DAN decomposition and machine intelligence. *Intelligent Systems with Applications*, 18, 200202.
- Garai, S., Paul, R. K., Rakshit, D., Yeasin, M., Paul, A. K., Roy, H. S., Barman, S. & Manjunatha, B. (2023). An MRA Based MLR Model for Forecasting Indian Annual Rainfall Using Large Scale Climate Indices. *International Journal of Environment and Climate Change*, 13(5), 137-150.

Examples

```
# creating example ####

# 1st element ####
# Create a sequence of dates from "2011-01-01" to "2015-12-31"
dates <- seq(as.Date("2011-01-01"), as.Date("2011-03-31"), by="day")

# Generate random prices for each date
price_1 <- runif(length(dates), min=0, max=100)

# Combine the dates and prices into a data frame
df <- data.frame(Dates = dates, Price_a = price_1)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates2 <- seq(as.Date("2011-05-01"), as.Date("2011-12-31"), by="day")

# Generate random prices for each date
price_2 <- runif(length(dates2), min=0, max=100)

# Combine the dates and prices into a data frame
df2 <- data.frame(Dates = dates2, Price_a = price_2)

# Merge the two data frames row-wise
df <- rbind(df, df2)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates3 <- seq(as.Date("2012-02-01"), as.Date("2012-12-31"), by="day")

# Generate random prices for each date
price_3 <- runif(length(dates3), min=0, max=100)
```

```
# Combine the dates and prices into a data frame
df3 <- data.frame(Dates = dates3, Price_a = price_3)

# Merge the two data frames row-wise
df <- rbind(df, df3)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates4 <- seq(as.Date("2013-04-01"), as.Date("2022-12-31"), by="day")

# Generate random prices for each date
price_4 <- runif(length(dates4), min=0, max=100)

# Combine the dates and prices into a data frame
df4 <- data.frame(Dates = dates4, Price_a = price_4)

# Merge the two data frames row-wise
df <- rbind(df, df4)

# Specify column data types
df <- data.frame(Dates = as.Date(df$Dates),
                price_a = round(as.numeric(df$Price_a)))
# 2nd element #####
# Create a sequence of dates from "2011-01-01" to "2015-12-31"
dates <- seq(as.Date("2011-01-01"), as.Date("2011-05-31"), by="day")

# Generate random prices for each date
price_1 <- runif(length(dates), min=0, max=100)

# Combine the dates and prices into a data frame
df_second <- data.frame(Dates = dates, Price_b = price_1)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates2 <- seq(as.Date("2011-06-01"), as.Date("2011-10-31"), by="day")

# Generate random prices for each date
price_2 <- runif(length(dates2), min=0, max=100)

# Combine the dates and prices into a data frame
df_second2 <- data.frame(Dates = dates2, Price_b = price_2)

# Merge the two data frames row-wise
df_second <- rbind(df_second, df_second2)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates3 <- seq(as.Date("2012-01-01"), as.Date("2012-12-31"), by="day")

# Generate random prices for each date
price_3 <- runif(length(dates3), min=0, max=100)

# Combine the dates and prices into a data frame
df_second3 <- data.frame(Dates = dates3, Price_b = price_3)
```

```
# Merge the two data frames row-wise
df_second <- rbind(df_second, df_second3)

# Create a sequence of dates from "2016-02-01" to "2022-12-31"
dates4 <- seq(as.Date("2013-03-01"), as.Date("2022-12-31"), by="day")

# Generate random prices for each date
price_4 <- runif(length(dates4), min=0, max=100)

# Combine the dates and prices into a data frame
df_second4 <- data.frame(Dates = dates4, Price_b = price_4)

# Merge the two data frames row-wise
df_second <- rbind(df_second, df_second4)

# Specify column data types
df_second <- data.frame(Dates = as.Date(df_second$Dates),
                       price_b = round(as.numeric(df_second$Price_b)))

# my_list ####
# Create a list
my_list <- list()

# Add the data frame to the list
my_list$df <- df

my_list$df_second <- df_second

# getting output ####
my_combined_data <- clean_and_combine(my_list = my_list)
print(head(my_combined_data))
my_imputed_data <- impute_combined(my_combined_data)
print(head(my_imputed_data))
my_weekly_data <- weekly_from_daily(my_imputed_data)
print(head(my_weekly_data))
```

Index

`clean_and_combine`, [2](#)

`impute_combined`, [5](#)

`monthly_from_daily`, [8](#)

`weekly_from_daily`, [12](#)