Imputation on Time Series Missing Data

IDS.506: Fargo Health Group Assignment Technical Appendix Rmd Notebook

Robert Duc Bui - mbui7@uic.edu - 660809303

```
library(tidyverse)
library(tidymodels)
library(lubridate)
library(zoo)
library(Cairo)
library(tsibble)
library(fable)
library(feasts)
library(imputeTS)
```

```
raw_ts <- read_csv("data/raw_ts.csv",
                   show_col_types = FALSE) %>%
  transmute(
    y = `Incoming Examinations`,
    datetime = paste(Year,Month,"01",sep="-") %>% ymd()
  )
```

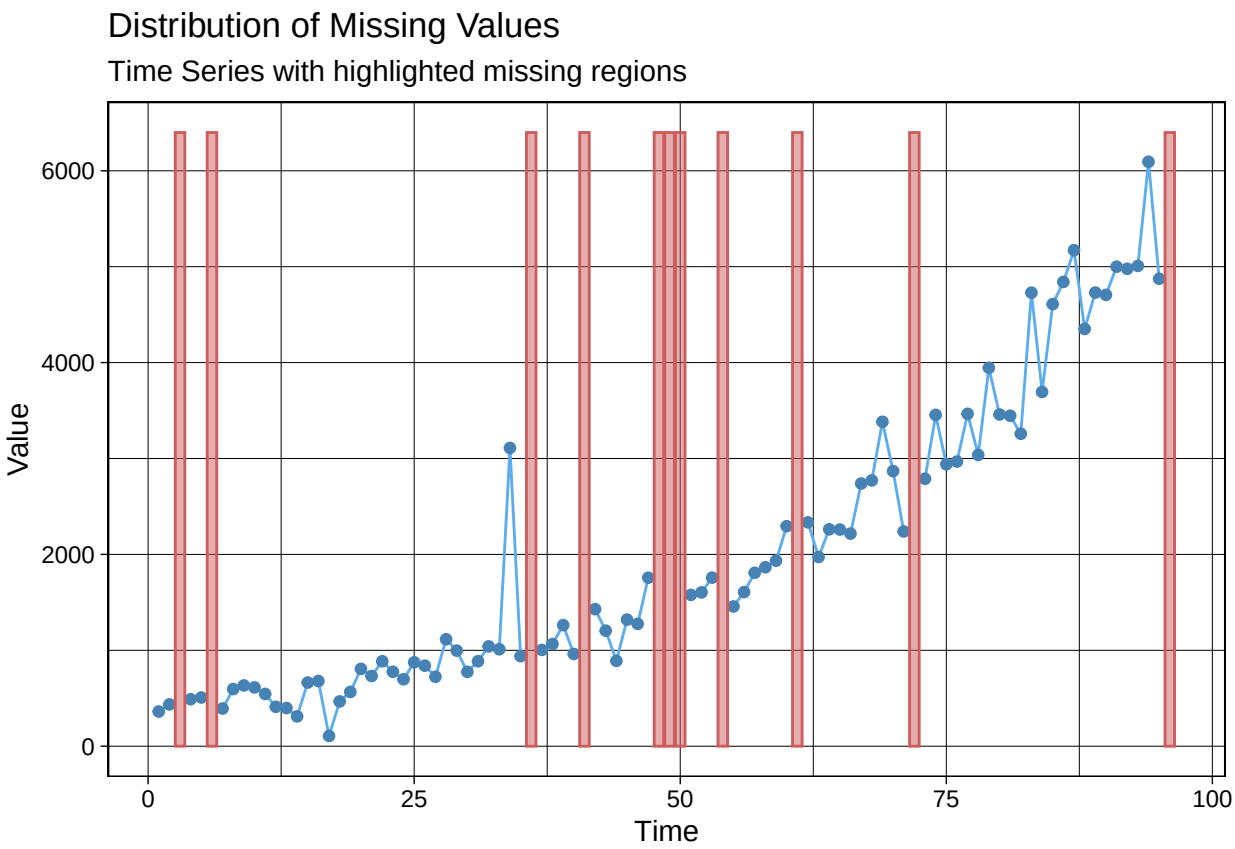
```
# tsibble ecosystem format
tsib <- raw_ts %>% as_tsibble(index = datetime)

# traditional ts format
ts <- raw_ts %>% select(y) %>%
  ts(start = c(2006,1),
     end   = c(2013,12),
     frequency = 12)

# printing time series as matrix
print(ts)
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 2006  362  436   NA  490  508   NA  393  596  634  613  545  411
## 2007  398  311  664  680  107  467  566  806  732  886  776  698
## 2008  875  840  724 1115  997  775  886 1041 1011 3110  939   NA
## 2009 1004 1065 1263  962   NA 1429 1205  890 1320 1276 1757   NA
## 2010   NA   NA 1578 1604 1758   NA 1457 1607 1808 1866 1934 2294
## 2011   NA 2334 1973 2262 2259 2217 2739 2772 3383 2869 2239   NA
## 2012 2789 3455 2940 2968 3466 3037 3946 3459 3446 3258 4729 3694
## 2013 4610 4841 5172 4351 4730 4706 5000 4978 5008 6094 4874   NA
```

```
# Summary Statistics of missing values
statsNA(ts)
```

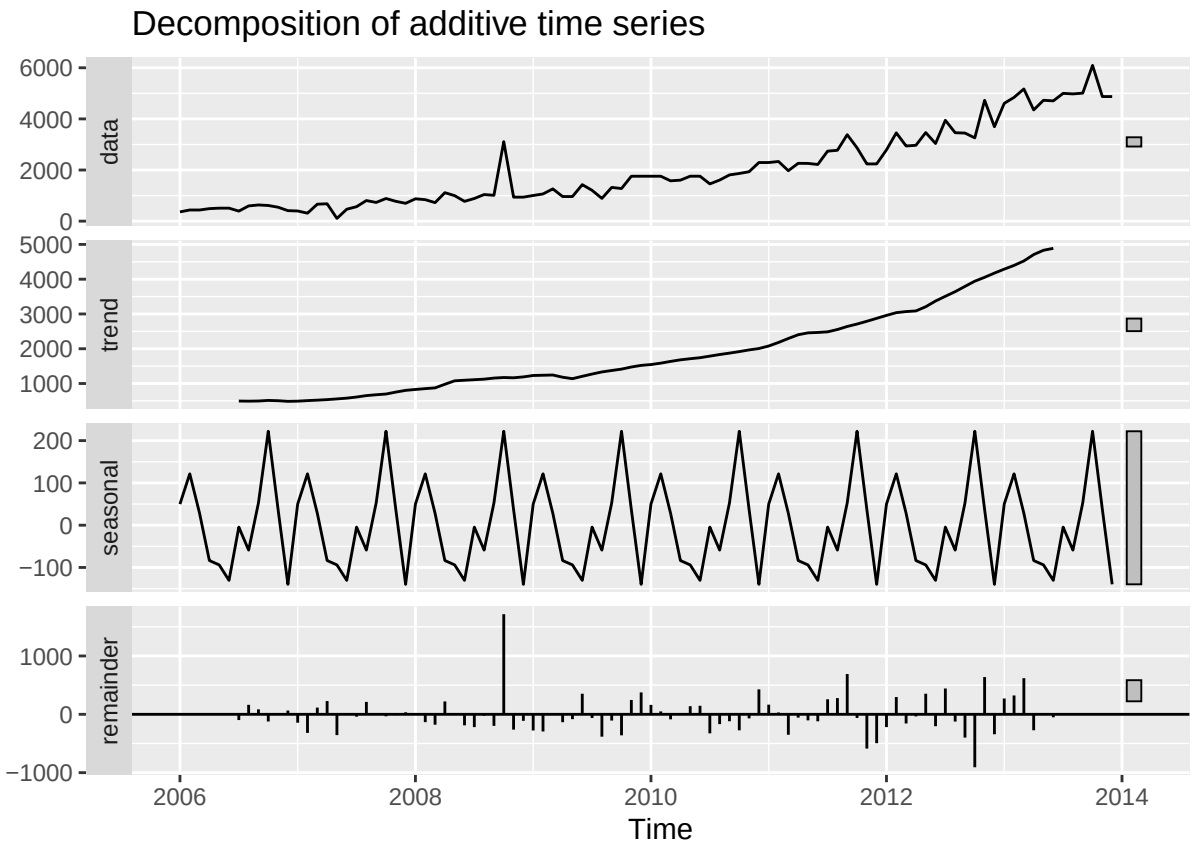
```
## [1] "Length of time series:"
## [1] 96
## [1] "-------------------------"
## [1] "Number of Missing Values:"
## [1] 11
## [1] "-------------------------"
## [1] "Percentage of Missing Values:"
## [1] "11.5%"
## [1] "-------------------------"
## [1] "Number of Gaps:"
## [1] 9
## [1] "-------------------------"
## [1] "Average Gap Size:"
## [1] 1.222222
## [1] "-------------------------"
## [1] "Stats for Bins"
## [1] "  Bin 1 (24 values from 1 to 24) :      2 NAs (8.33%)"
## [1] "  Bin 2 (24 values from 25 to 48) :      3 NAs (12.5%)"
## [1] "  Bin 3 (24 values from 49 to 72) :      5 NAs (20.8%)"
## [1] "  Bin 4 (24 values from 73 to 96) :      1 NAs (4.17%)"
## [1] "-------------------------"
## [1] "Longest NA gap (series of consecutive NAs)"
## [1] "3 in a row"
## [1] "-------------------------"
## [1] "Most frequent gap size (series of consecutive NA series)"
## [1] "1 NA in a row (occurring 8 times)"
## [1] "-------------------------"
## [1] "Gap size accounting for most NAs"
## [1] "1 NA in a row (occurring 8 times, making up for overall 8 NAs)"
## [1] "-------------------------"
## [1] "Overview NA series"
## [1] "  1 NA in a row: 8 times"
## [1] "  3 NA in a row: 1 times"
```

```
# plotting missing periods
ggplot_na_distribution(ts)
```



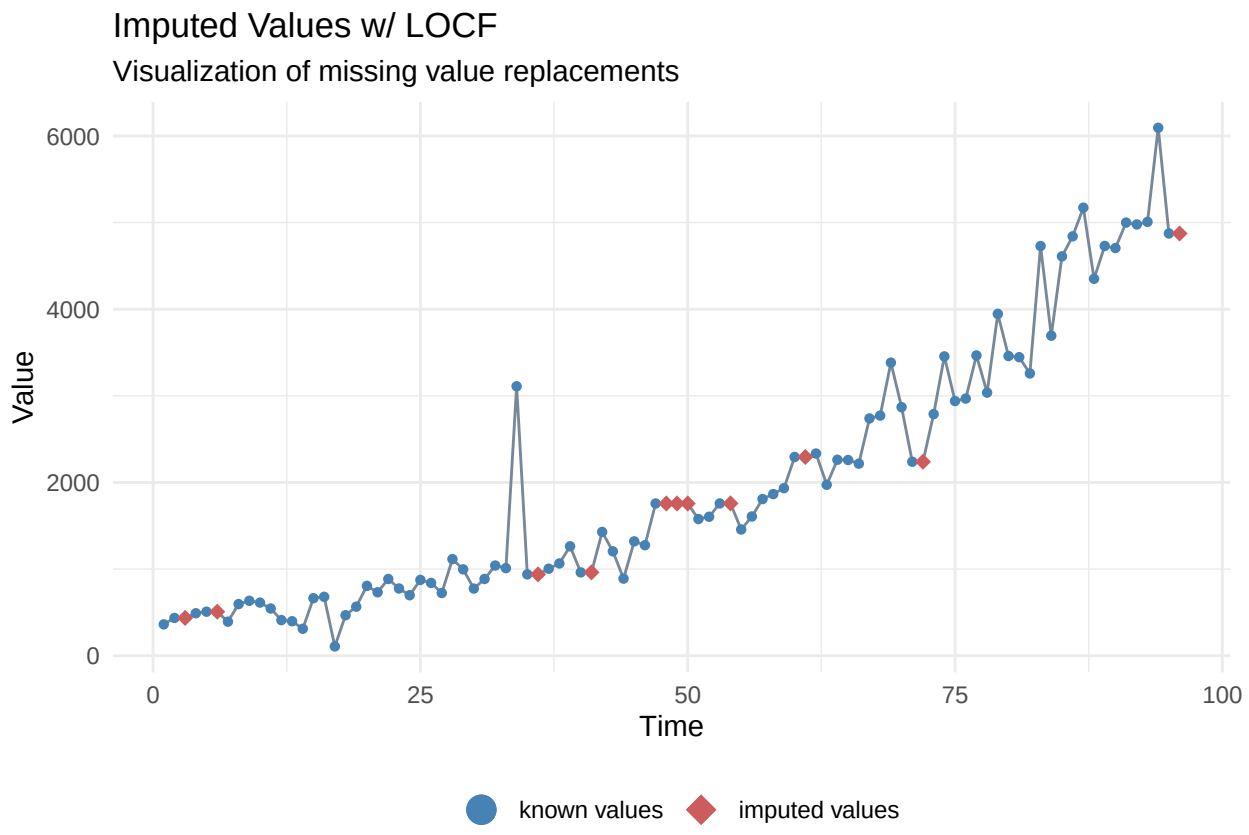
Sanity Check: Is there seasonality?

```
# STL decomposition with basic quick imputed LOCF
ts %>%
  na_locf() %>%
  decompose() %>%
  autoplot(s.window = 'periodic')
```

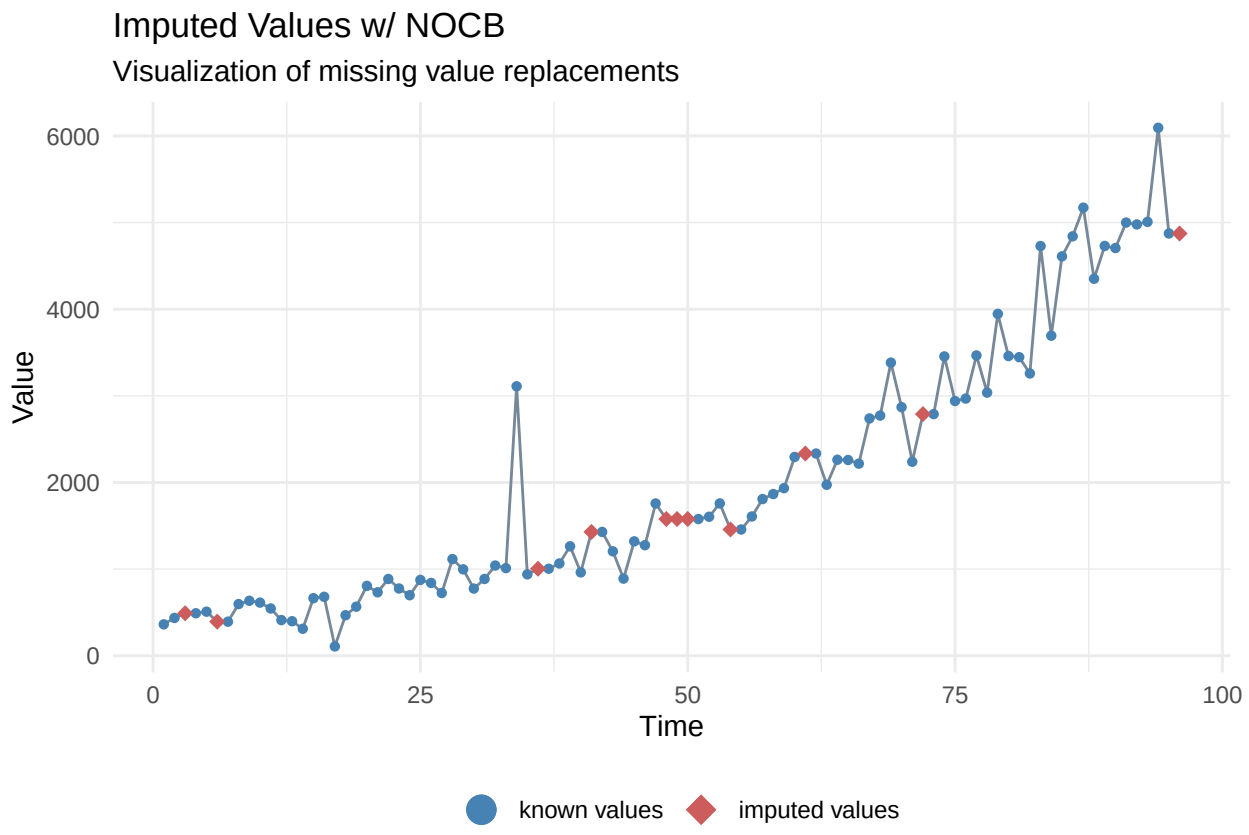


Basic Methods

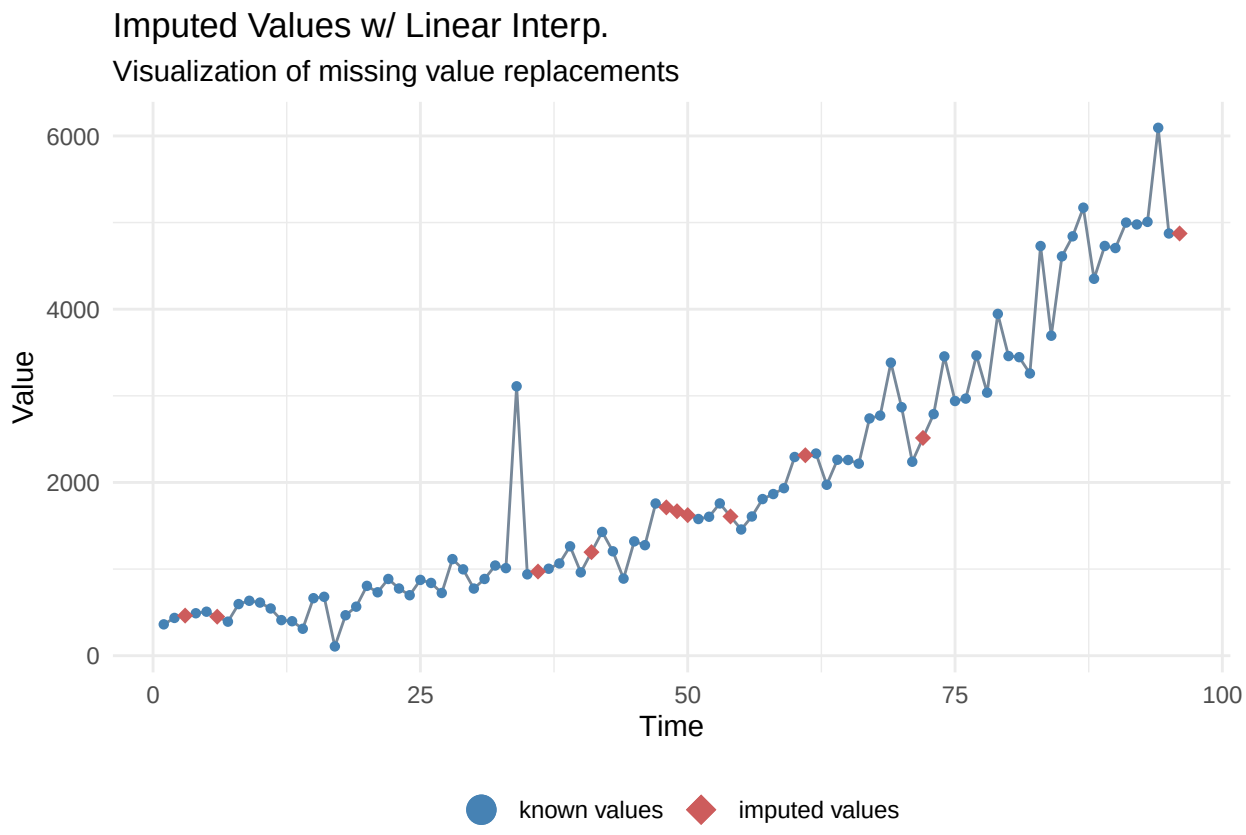
```
# Last-Observation-Carried-Forward imputation
ts_locf <-
  ts %>% na_locf(option = "locf",
                 # For exceptions with no observation from same direction:
                 # reverse direction of method
                 na_remaining = "rev",
                 maxgap = Inf)

ggplot_na_imputations(x_with_na = ts,
                      x_with_imputations = ts_locf,
                      title = "Imputed Values w/ LOCF",
                      theme = ggplot2::theme_minimal())
```




```
# Next-Observation-Carried-Backward imputation
ts_nocb <-
  ts %>% na_locf(option = "nocb",
                 # For exceptions with no observation from same direction:
                 # reverse direction of method
                 na_remaining = "rev",
                 maxgap = Inf)

ggplot_na_imputations(x_with_na = ts,
                      x_with_imputations = ts_nocb,
                      title = "Imputed Values w/ NOCB",
                      theme = ggplot2::theme_minimal())
```

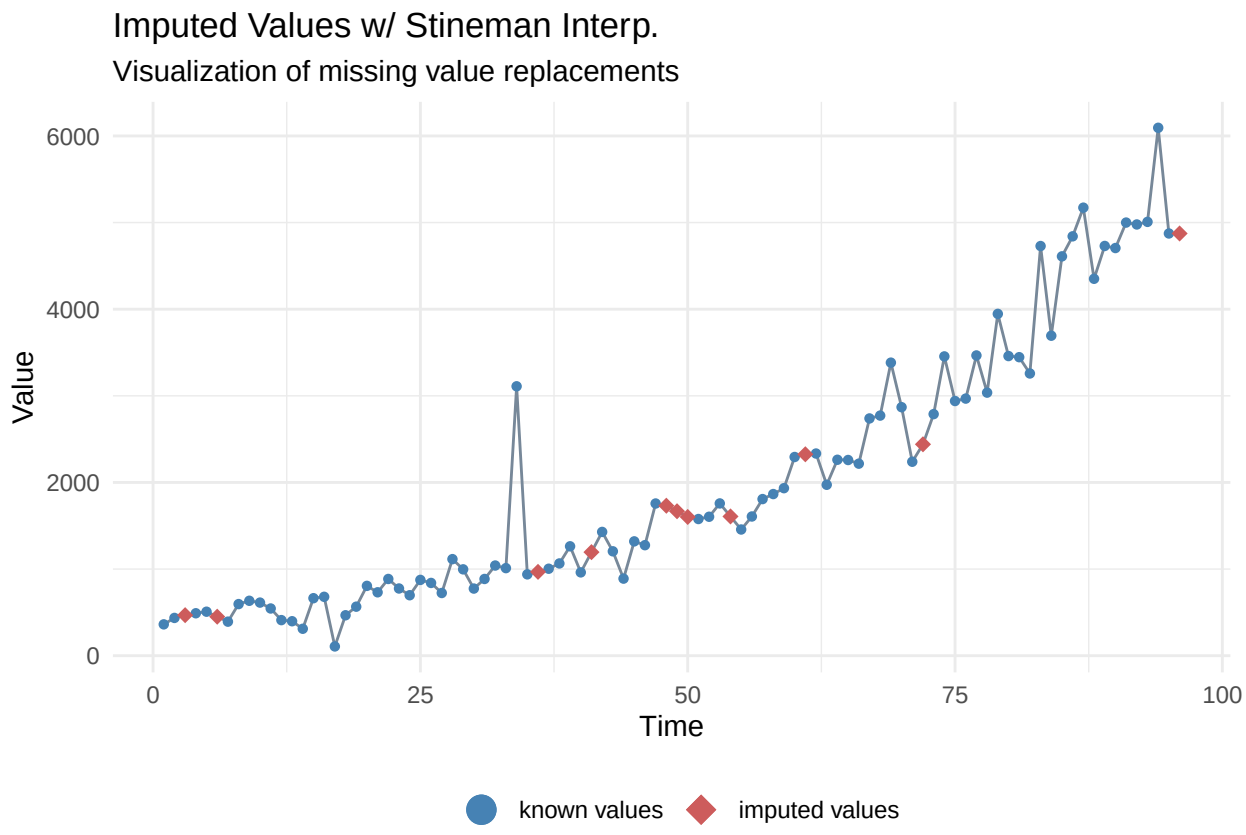



```
# Linear Interpolation
ts_linear <-
  ts %>% na_interpolation(option = "linear",
                          maxgap = Inf)

ggplot_na_imputations(x_with_na = ts,
                      x_with_imputations = ts_linear,
                      title = "Imputed Values w/ Linear Interp.",
                      theme = ggplot2::theme_minimal())
```

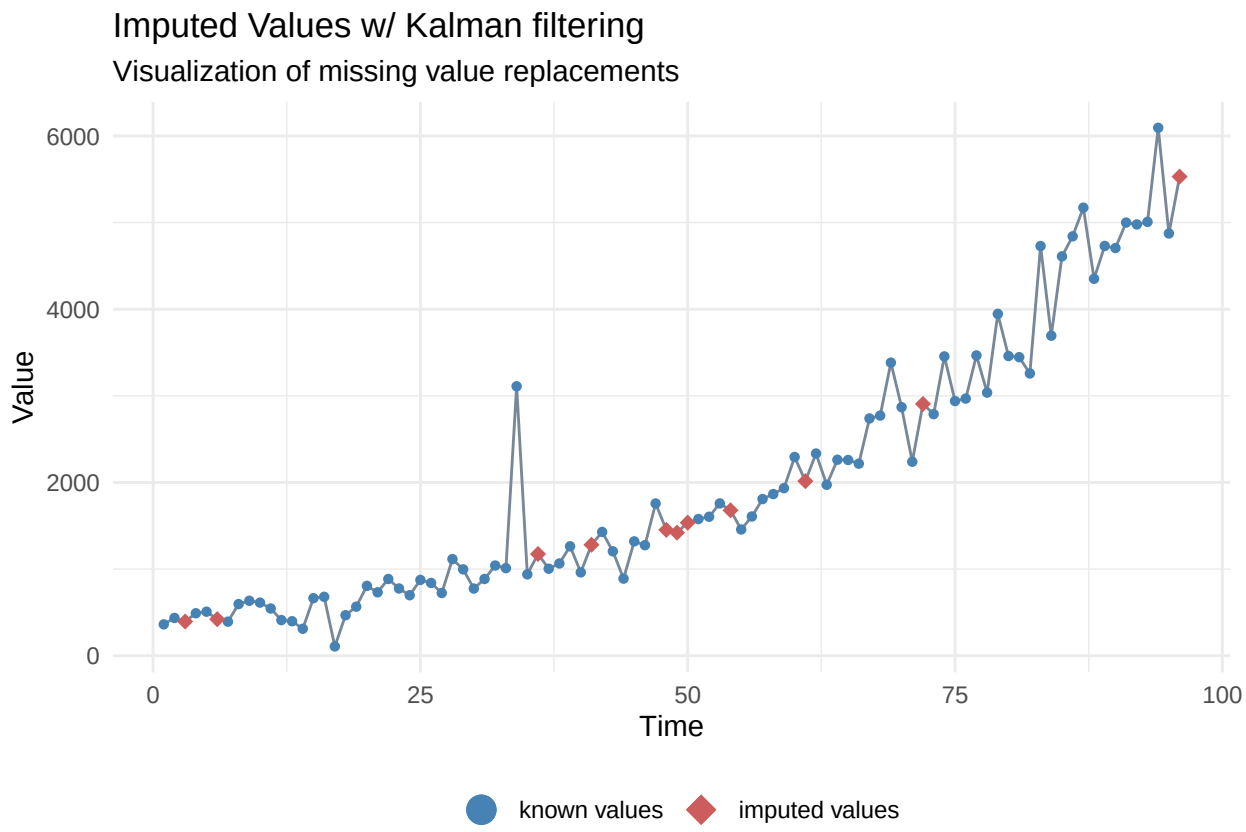


Intermediate Methods

```
# Stineman Interpolation
ts_stineman <-
  ts %>% na_interpolation(option = "stine",
                          maxgap = Inf)

ggplot_na_imputations(x_with_na = ts,
                      x_with_imputations = ts_stineman,
                      title = "Imputed Values w/ Stineman Interp.",
                      theme = ggplot2::theme_minimal())
```

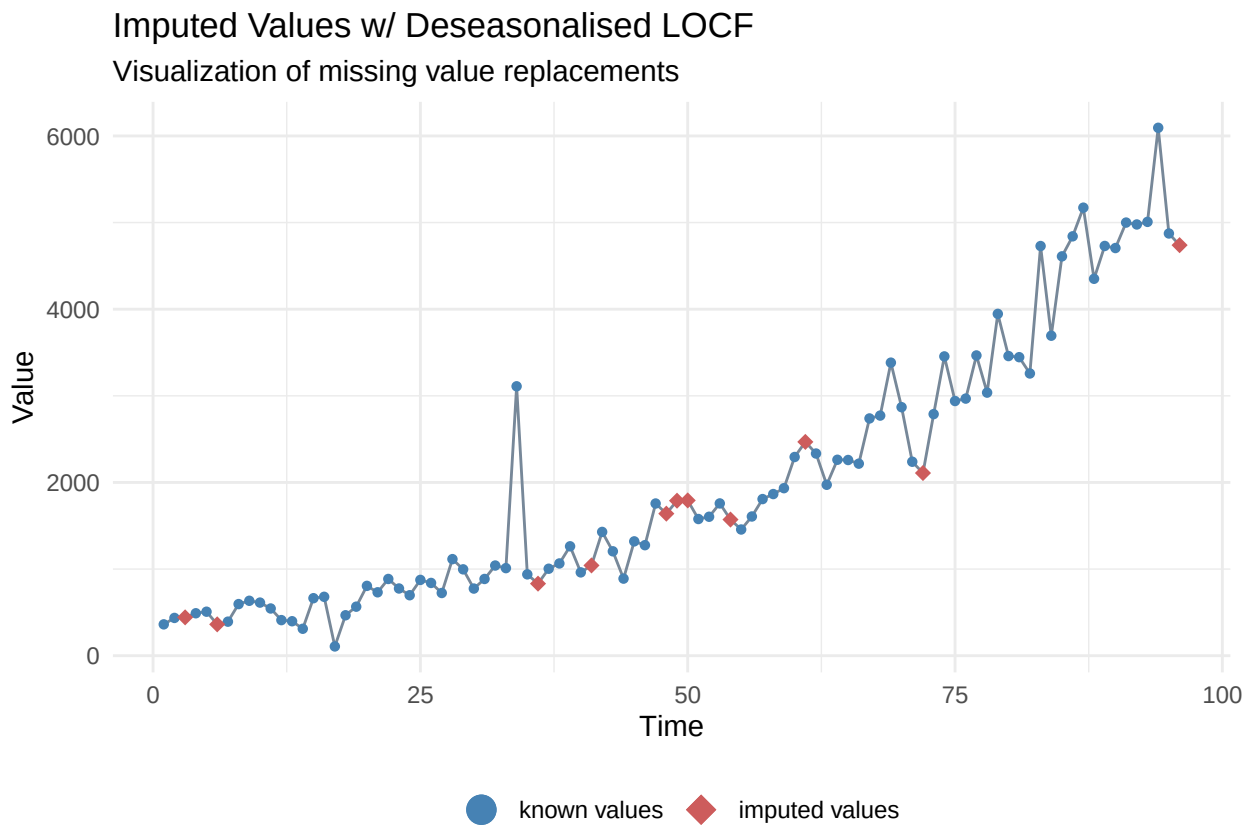



```
# Kalman filtering imputation
ts_kalman <-
  ts %>% na_kalman(model = "StructTS",
                   smooth = T,
                   maxgap = Inf)

ggplot_na_imputations(x_with_na = ts,
                      x_with_imputations = ts_kalman,
                      title = "Imputed Values w/ Kalman filtering",
                      theme = ggplot2::theme_minimal())
```

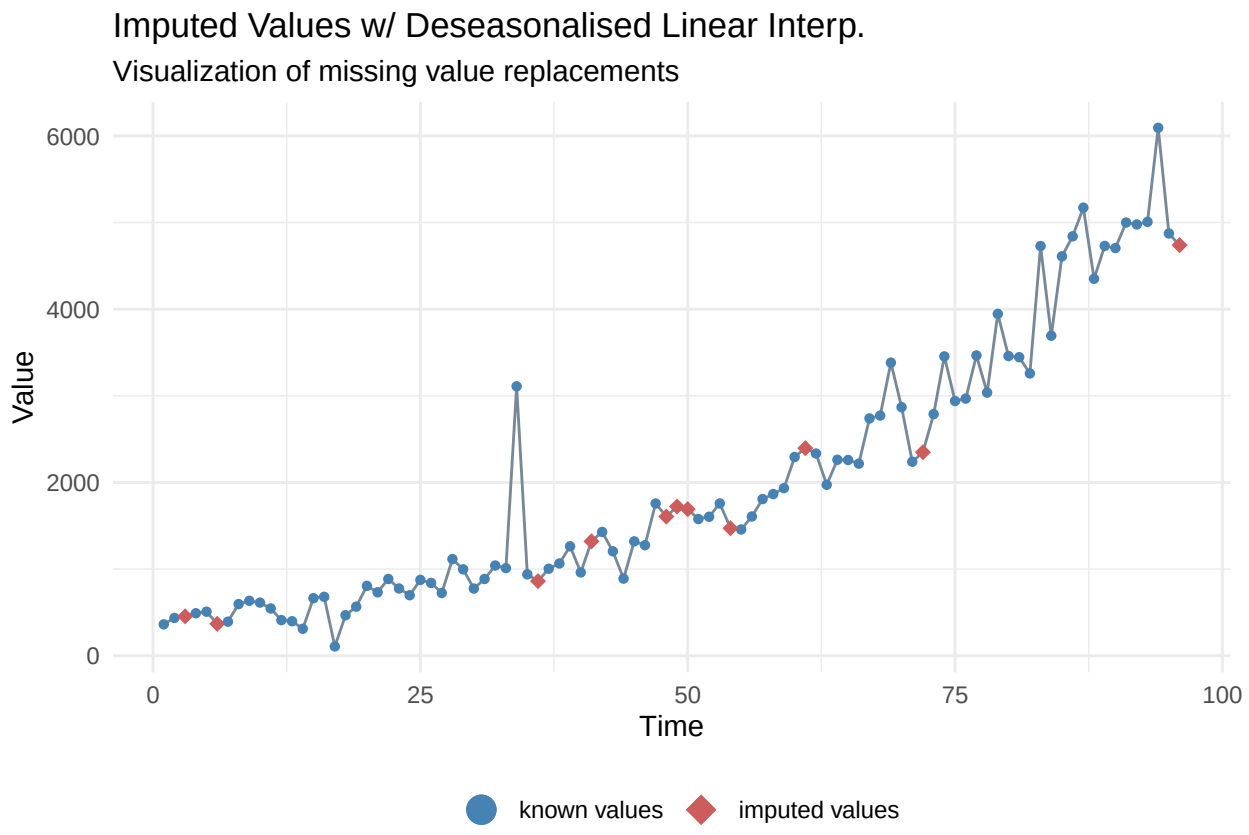


Deseasonalised Methods

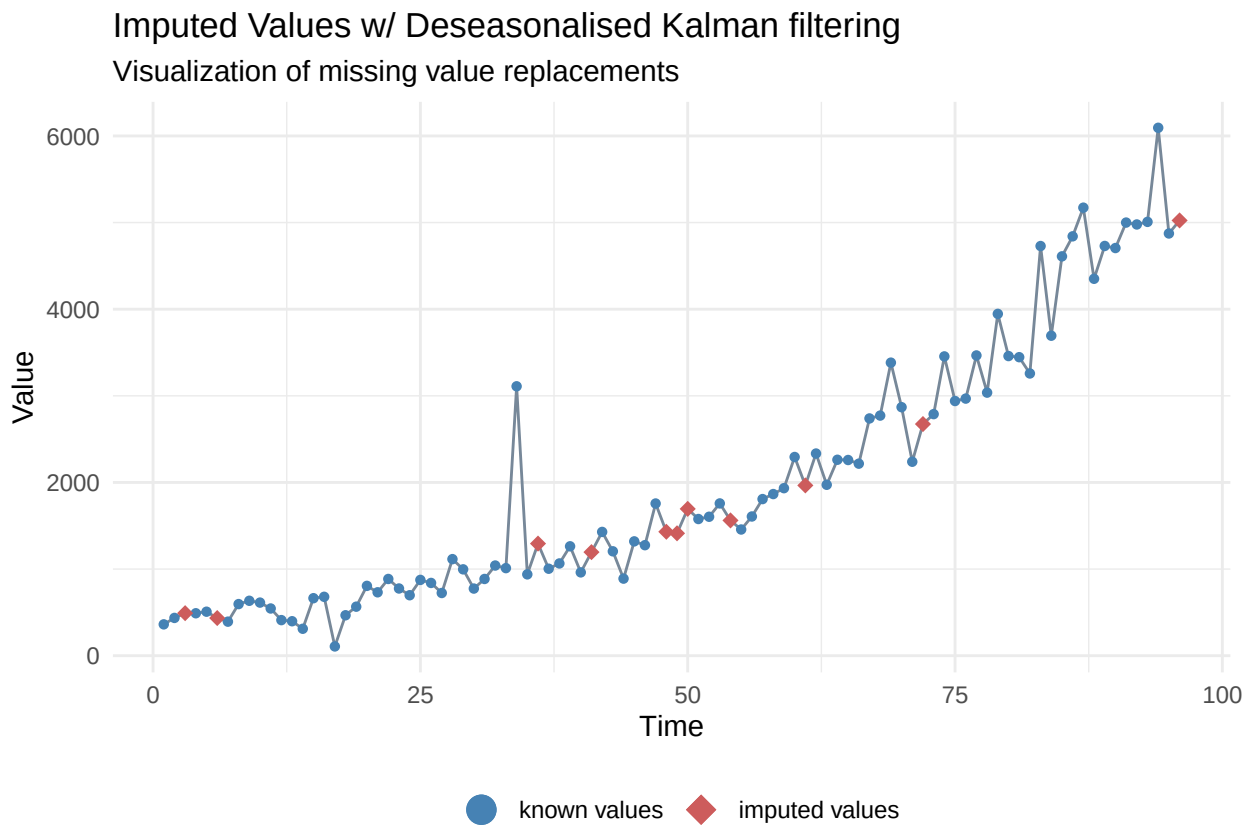
```
# Deseasonalised LOCF
ts_dsea_locf <- ts %>%
  na_seadec(algorithm = "locf",
            maxgap = Inf)

ggplot_na_imputations(x_with_na = ts,
                      x_with_imputations = ts_dsea_locf,
                      title = "Imputed Values w/ Deseasonalised LOCF",
                      theme = ggplot2::theme_minimal())
```




```
# Deseasonalised linear interp
ts_dsea_linear <-ts %>%
  na_seadec(algorithm = "interpolation",
            maxgap = Inf)

ggplot_na_imputations(x_with_na = ts,
                      x_with_imputations = ts_dsea_linear,
                      title = "Imputed Values w/ Deseasonalised Linear Interp.",
                      theme = ggplot2::theme_minimal())
```




```
# Deseasonalised Kalman filtering
ts_dsea_kalman <- ts %>%
  na_seadec(algorithm = "kalman",
            maxgap = Inf)

ggplot_na_imputations(x_with_na = ts,
                      x_with_imputations = ts_dsea_kalman,
                      title = "Imputed Values w/ Deseasonalised Kalman filtering",
                      theme = ggplot2::theme_minimal())
```



Exporting imputed time series objects

```
# Helper function to parse ts obj
to_df <- function(input_ts){
  output_df <- input_ts %>%
    as_tsibble() %>%
    as_tibble() %>%
    transmute(
      `Incoming Examinations` = round(value),
      Year = substr(index %>% as.character(), 1,4),
      Month = substr(index %>% as.character(), 6,8) %>% match(month.abb)
    )
  return(output_df)
}

# Writing to csv (modified from Dr. Ron J Hyndman's "Saving ts objects as csv files")
# Citation (Chicago 17th):
# "Saving Ts Objects as Csv Files | Rob J Hyndman."
# Accessed April 12, 2022. https://robjhyndman.com/hyndsight/ts2csv/.
ts_to_csv <- function(x) {
  fname <- paste0("data/",deparse(substitute(x)), ".csv")
  readr::write_csv(to_df(x), fname)
}
```

```
ts_to_csv(ts_locf)
ts_to_csv(ts_nocb)
ts_to_csv(ts_linear)
ts_to_csv(ts_stineman)
ts_to_csv(ts_kalman)
ts_to_csv(ts_dsea_locf)
ts_to_csv(ts_dsea_linear)
ts_to_csv(ts_dsea_kalman)
```