# Time Series Project

Jacob Clinton George Smith-Kolff

2024-10-13

# Contents

# Chapter 1

# Introduction

(Probably work on this section last)

## 1.1 What is missing data

## 1.2 Issues caused by missing data

## 1.3 Structure of the report

# Chapter 2

# Theory

## 2.1 Time series

**Missing at random. This basically means that the missingness in the data is not imformative**

We begin by providing definitions for both uni variate and multivariate time series:

A univariate time series $Y = \{y_1, y_2, ..., y_t\} \in \mathbb{R}^t$ is a sequence $t$ observations on a single variable.
This can be extended to multivariate time series:
$X = \{x_1, x_2, ..., x_t\} \in \mathbb{R}^{t \times d}$ where each $x_i$ is a $d$ dimensional vector, with observed values $y = \{y_1, y_2, ..., y_t\} \in \mathbb{R}^t$.

## 2.2 Missing data

remove the subsubsection numbers later

Data missingness is a common occurrence that is common in working with real world data. Missing data can arise from device failures such as measureing equipment failing, or from data censoring (such as from governments) [1]. Missing data typically falls into one of three categories:

### 2.2.1 Missing Completely at Random (MCAR)

Data is said to be missing completely at random if the distribution that describes how missingness occurs is independent of both the observed and unobserved values in the time series.

### 2.2.2 Missing at Random (MAR)

Missing at random is when missingness is related to the observed data, but is independent of the unobserved data. This means that there is some external factor that is causing the missingness. An example given in [2] is that data from a sensor is more likely to be missing on weekends since on some weekends the sensor is shutdown.

### 2.2.3 Not Missing at Random (NMAR)

Not missing at random means that the missingness is related to the value of the observation itself. An example of this is a sensor that will return a missing value if the recorded value is above 100°.

Come back and include probability notation

## 2.3 Missing data imputation

Most missing data imputation methods require MCAR or MAR, this is because these systems of missingness are not informative towards the missing values themselves.

Can extend this writeup later.

Idea is to start with the most basic techniques, then more the more advanced ones

The most simple approach to dealing with missing data is to simply delete the missing observation. However, this approach can lead to a biased result as we have lost information. Consider real world situations, where if we have a multivariate time series, there most likely exists some dependencies between the variables that we can exploit to aid in imputation if we have missing values in either $x_i$ or $x_i$.

further discussion on exploting the dependencies of X to help input missing values within X or y

Can also mention the differences between inputing a missing target variable vs a missing predictior variable in a multivariate

### 2.3.1 Univariate data imputation

#### 2.3.1.1 The struggle with univariate data imputation

Unlike with a multivariate time series, a univariate time series is only a single sequence of real numbers $\{y_1, y_2, ..., y_n\}$. This simplier form actually leads to increased difficulty when it comes to imputing a missing value of $y$, since we can no longer exploit any dependencies between the predictor variables in the series. With univariate imputation we can only rely on the previous (or future) values for $y$ along with the implicit time variable [2].

#### 2.3.1.2 Last Observed Carried Forward (LOCF) Next Observation Carried Backward (NOCB)

One of the most basic imputation techniques for a univariate series is LOCF. In this method we simply replace the missing value with the previous (non missing) observed value. NOCB is similar but we take the next observed value and back fill the missing value. For a small univariate time series with strong autocorrelation this method can work just fine. However, using these methods does introduce flat lines into the series, which can distort the temportal pattern in the series and introduce bias in the series. This approach is also not viable for large gaps. [3]

#### 2.3.1.3 Mean/Median imputation

Mean or median imputation is another very simple technique to quickly fill missing values. This approach is simple, but like LOCF and NOCB it ignores any temporal aspect of the series. For mean imputation we calculate the mean of the observed values for $Y$ thus $\bar{y}$ is used as $\overline{y}_i$, for the missing value at $y_i$. The approach is the same for median imputation, but $y_i$ is just the middle value of $Y$. Mean and median imputation can be a quick and easy solution, but this method assumes a stationary time series, that is, if there is a trend in the series using a mean or median imputation technique can lead to very biased or rubbish results for the imputation. This method also ignores any temportal dependencies in the series, and lasty is not suitable if there is a large number of missing values present in the series.

#### 2.3.1.4 ARIMA-based Imputation

Missing data can be imputed using an ARIMA model

$$\left(1 + \sum_{i=1}^{q} \beta_i B^i\right) y(t) W(t)$$

Where $p$ is the AR part, $d$ is the degree of differencing, $q$ is order of the moving average part. In ARIMA based imputation the model will use the rest of the series to impute the missing value, if there are several missing values then each missing value is calculated one at a time and the model will use the previously imputed values as well as the data in the series to predict the next missing values

#### 2.3.1.5 Kalman filtering

### 2.3.2 Multivariate data imputation techniques

#### 2.3.2.1 K-nearest neighbours

K-Nearest neighbors is a simple non-parametric machine learning algorithm, it can be used for either classification or regression. It can be applied in a time series data imputation context to estimate missing values in a time series.

$$\frac{1}{K}\sum_{j=1}^{K} Y_j$$

Consider a missing value $x_i$ in a time series, k-nearest neighbors will impute a value for the missing value by calculating the mean of the datapoints in the neighborhood determined by a distance metric such as Euclidean distance. The algorithm is simple and is shown to be quite effective [3]. K-nearest neighbors algorithm has a hyper-parameter K, the number of nearest (non missing) data points to consider. A choice of K that is too large may lead to smoothing over temporal patters (under fitting) this has the effect of ignoring potentially important seasonal patterns. On the other hand a value of K that is too small could lead to being overly sensitive to the noise. The value for K needs to be carefully chosen, typically by cross validation, but in a time series context it could be chosen using sliding windows and forward chaining. The curse of dimensionality is an issue for K-nearest neighbors. With higher dimension it becomes harder to find data points that are closer, this can lead to an increased sensitivity to noise and misleading imputations. The computational cost also increases rapidly with more dimensons in the data making the algorithm less efficient. Another issue related to computational cost, K-nearest neighbors is a lazy learning algorithm so the complete dataset ends up being stored for large datasets this can make the algorithm slow or even infeasible

Come back to this section later

#### 2.3.2.2 Multivariate Imputation by Chained Equations (MICE)

#### 2.3.2.3 General Adversial Networks (GAM)

# Chapter 3

# Illustration of the method application

Using the methods of missing data in practice.

## 3.1 The data

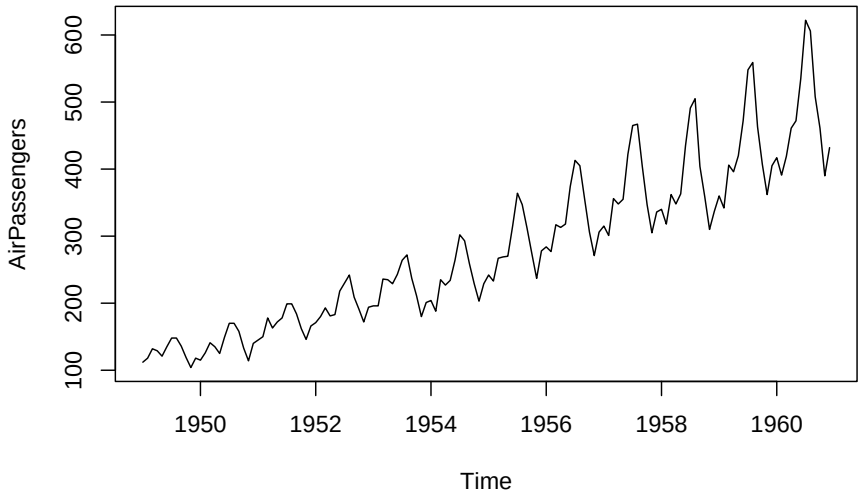
### 3.1.1 AirPassengers

The first data set is AirPassengers from [4]. . . ..

## 3.2 The methods

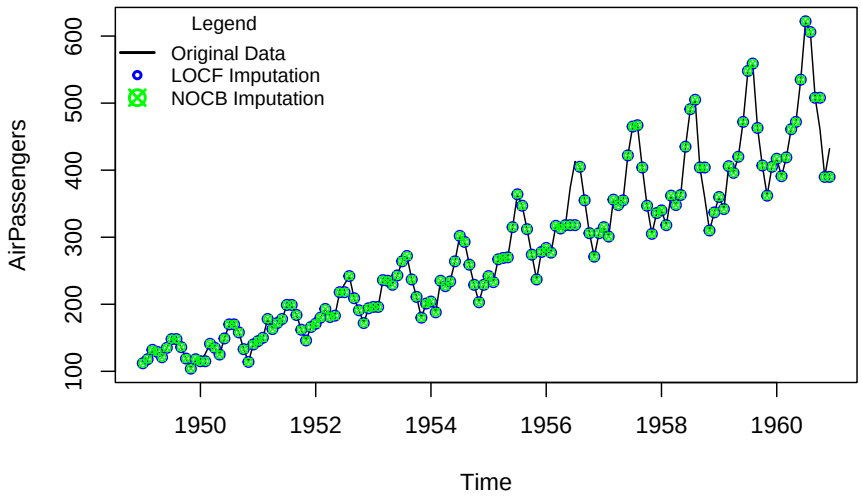
### 3.2.1 LOCF and NOCB

The simple methods LOCF and NOCB are analysed first. Consider the plot of the




series
The series shows not only a strong increasing trend, but also strong seasonality. It is common place for a time series to have both of these ...

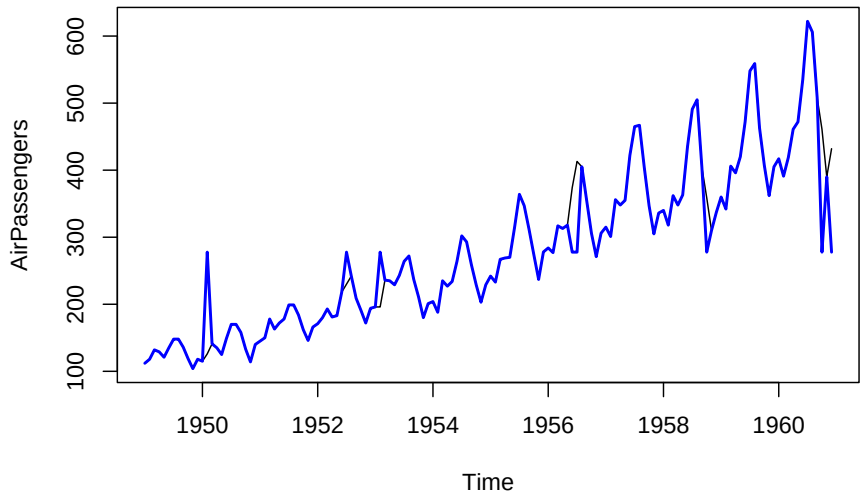
Now to simulate MCAR randomness I randomly set 6% of the observations in the data to be NA. Which corresponds to 8 missing values in this dataset.




Discussion of how it looks

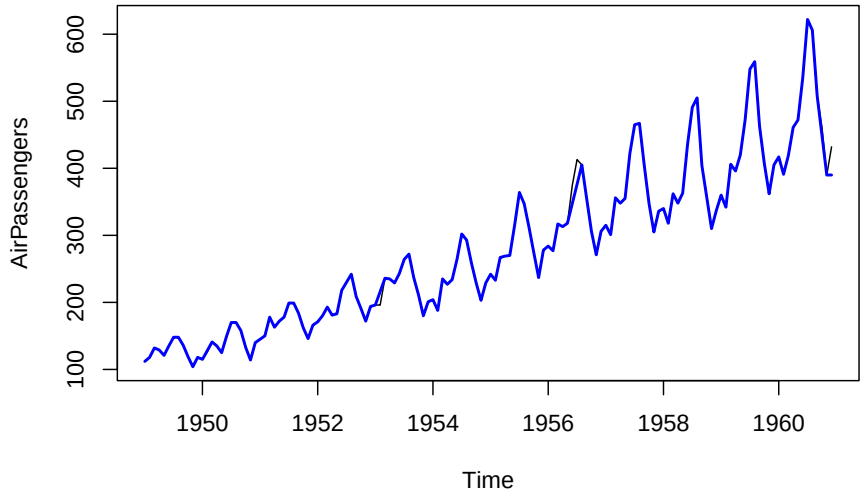
Next I show off the mean imputation (and median maybe)

```
plot(AirPassengers)
Air.mean <- Air.missing.6 |> na_mean()
lines(Air.mean, type = "l", col = 'blue', lwd =2)
```

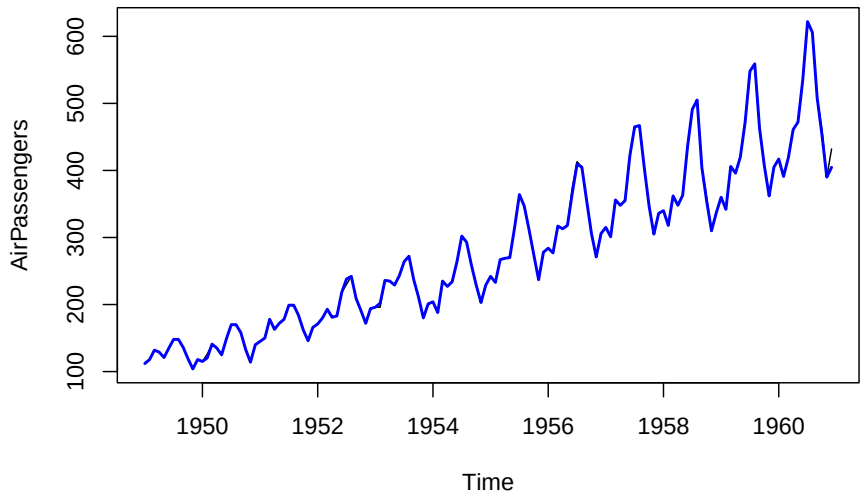



Discuss how this performs

```
Air.interp <- Air.missing.6 |> na_interpolation(option = "linear")
plot(AirPassengers)
lines(Air.interp, type = "l", col = 'blue', lwd = 2)
```



```r
Air.interp <- Air.missing.6 |> na_kalman()
plot(AirPassengers)
lines(Air.interp, type = "l", col = 'blue', lwd = 2)
```




Discussion about kalman kicked actual ass

Come back and make the plots nice. Can also remove some plots and put them in the appdix

# Chapter 4

# Conclusion


[1] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 793. John Wiley & Sons, 2019.

[2] S. Moritz, A. Sardá, T. Bartz-Beielstein, M. Zaefferer, and J. Stork, "Comparison of different methods for univariate time series imputation in r," *arXiv preprint arXiv:1510.03924*, 2015.

[3] H. Ahn, K. Sun, K. P. Kim, *et al.*, "Comparison of missing data imputation methods in time series forecasting," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 767–779, 2022.

[4] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: Forecasting and control*. John Wiley & Sons, 2015.

# Chapter 5

# Appendix