# Evaluating the Impact of Measurement Error in Microbial Time Series Analysis: A SIMEX and PACF Approach

Qianzhu Liu (B00925970)

Dalhousie University

Honour Project

Supervised by: Professor Toby Kenney

April 28, 2024

(Word count: 4599)

## Abstract

The microbiome plays a crucial role in health, environment, agriculture, and other critical domains. Despite its significance, many studies treat microbial communities as static entities, ignoring the inherent temporal dynamics. This research addresses the measurement error in estimating microbial abundance, which can lead to substantial discrepancies in understanding these dynamics. Given the presence of large errors, traditional Markovian assumptions may not hold, indicating that the abundance on one day may not necessarily depend on the previous day's values.

Our study explores the impact of measurement error on microbial time series data using Partial Auto-correlation Function (PACF). By systematically adding varying levels of measurement error, we examine how these errors influence PACF at different lags. The resulting relationship between PACF and measurement error offers insights into the accuracy of abundance estimates. This approach provides a more robust framework for analyzing temporal dynamics in microbial communities, allowing researchers to better understand the variability and underlying processes within these complex systems.



*Keywords: Time Series; Measurement Error; Partial Auto-correlation Function (PACF); Microbial Abundance; SIMEX.*


.

## Introduction

Understanding the temporal dynamics of microbial communities is crucial for research in health, environment, agriculture, and beyond. However, many studies treat these communities as static, overlooking their inherent variability over time. To address this gap, our research focuses on the measurement error in estimating microbial abundance and how it impacts the analysis of temporal dynamics in these ecosystems.

For this study, we use the moving picture dataset , which provides nearly daily samples from four body sites of two healthy individuals. The dataset includes counts of each microbe in each sample, along with the date of collection. Our analysis focuses on data aggregated at the genus level, with attention given to the most abundant genera in the gut dataset. This dataset serves as a robust source for investigating the temporal dynamics of microbial communities and exploring the impact of measurement error on time series analysis.

Given the significant role of measurement error, this report explores the use of the Partial Auto-correlation Function (PACF) to understand its effect on microbial time series data. By varying the levels of measurement error, we aim to determine its impact on PACF and ultimately improve our understanding of microbial abundance over time.

The data used in our study are sourced from a comprehensive research project titled "Moving Pictures of the Human Microbiome." In this study, researchers collected nearly daily samples from four body sites of two healthy individuals, one male (M3) and one female (F4), over a period of 15 months for M3 and 6 months for F4. The researchers sequenced the 16S rRNA gene from these samples, focusing on three body sites: the gut (feces), mouth, and skin (both left and right palms).

The sequencing was carried out using the Illumina Genome Analyzer IIx (GA-IIx), targeting the V4 region, with additional sequencing done using the 454 platform for the V2 region to ensure cross-platform consistency. The dataset includes counts of each microbe in each sample, along with the date of collection. This data is publicly available and has been used extensively to study the temporal dynamics of the human microbiome.

For our analysis, we used the data aggregated at the genus level, focusing on the most abundant genera in the gut dataset. By applying Partial Autocorrelation Function (PACF) analysis to this data, we aimed to investigate the impact of measurement error on microbial abundance estimates. This dataset, with its dense time series and extensive body site coverage, provides a valuable resource for studying the temporal variability in microbial communities.

This data source allows us to examine the persistent and transient components of the microbiome, offering insights into how measurement error might affect the interpretation of microbial abundance and the temporal dynamics in these ecosystems.

This research context sets the stage for our study's exploration of the effects of measurement error on time series analysis within microbial communities.

The structure of this report is as follows: Section 2 describes the methods, including data preparation and the process of simulating measurement error; Section 3 presents the results, focusing on the relationship between measurement error and PACF at different lags; Section 4 offers conclusions and discussions, summarizing key findings and suggesting areas for future research.

Through this approach, we aim to provide a deeper insight into the temporal dynamics of microbial communities and contribute to a more accurate analysis of time series data in the presence of measurement error.

# Methods

## Data Pre-processing

Data preprocessing is a crucial first step in any statistical analysis, ensuring that the data is primed for accurate and efficient analysis. In this study, the preprocessing involves several key stages:

Data Loading: We begin by loading the microbial community data from text files, which involves reading comprehensive datasets containing counts of various microbial genera across different samples. This is accomplished using R's read.table function, allowing us to handle large datasets efficiently and prepare them for further manipulation. For instance, we use commands like gut2genera <- read.table (" moving_pic_data_genus.data_FECES_2.txt ") to import data, ensuring all entries are correctly formatted and accessible.

Data Cleaning: Once loaded, the data undergoes a cleaning process. This includes verifying data integrity, removing or correcting any errors or outliers, and ensuring that the dataset is consistent. This stage is critical for obtaining reliable results from subsequent analyses. During this phase, we also perform operations like aggregating total counts for each genus within each sample. This is achieved through operations such as colSums (gut2genera), which provides a summary of the total counts for each microbial genus.

Data Transformation and Selection: After cleaning, we focus on data transformation and selection, where specific variables of interest are extracted and

transformed as needed for the analysis. In our study, this includes isolating the counts of a particular genus of interest, such as Bacteroides, using specific indexing methods. The data is also normalized or transformed to meet the assumptions of the statistical tests we plan to use. For example, we compute proportions of the Bacteroides genus relative to the total microbial counts in each sample using the formula:

$$\text{Bacterioides\_prop} = \frac{\text{gut2genera[,"k\_\_Bacteria.p\_\_Bacteroidetes..."]}}{\text{rowSums(gut2genera)}}$$

This transformation is crucial for comparing microbial abundances across different samples, which may vary significantly in total microbial counts.

**Principle and Implementation of Linear Regression Model**

Model Establishment: The model is based on the assumption that there is a linear relationship between the measurement errors and the observed values. We use a linear regression model with the noisy Bacteroides proportions (noisy_Bacterioides_prop_new) as the response variable and the artificially introduced measurement errors (measurement_error_new) as the explanatory variable. The model is expressed as:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Here, $Y$ represents the noisy Bacteroides proportion, $X$ denotes the measurement error, $\beta 0$ and $\beta 1$ are the coefficients to be estimated, and $\epsilon$ is the error term, assumed to be normally distributed with a mean of zero.

Model Fitting and Coefficient Estimation: The model is fitted using the least squares method, which estimates the values of $\beta 0$ and $\beta 1$. This step is implemented using R's lm() function, calculating the influence of the explanatory variable (measurement error) on the response variable (noisy Bacteroides proportion).

Error Estimation and Data Correction: The fitted model is used to predict the part of data variation caused by measurement errors, termed estimated_measurement_error. By subtracting this estimated error part from the noisy data, we obtain the corrected data (noise_free_log_data):

$$\text{Corrected Data} = \text{Noisy Data} - \text{Estimated Measurement Error}$$

Effect Evaluation: The effectiveness and accuracy of the error correction are evaluated by plotting and further statistical analysis (such as calculation of the Partial Autocorrelation Function, PACF), to verify the validity of the model correction.

**SIMEX Method**

SIMEX (Simulation Extrapolation) is used to address measurement error in data. The steps include simulation, fitting, and extrapolation.

Simulate Error Increase: Artificially adds varying levels of error to the original data to simulate the impact of measurement errors. Error simulation can be implemented by adding normally distributed random noise:

$$\text{noisy\_data} = \text{original\_data} + \epsilon$$
$$\text{where } \epsilon \sim N(0, \sigma^2).$$

Fit Model and Record Results: For each level of error, fit the statistical model (such as linear or time series models) and record key statistics or model parameters.

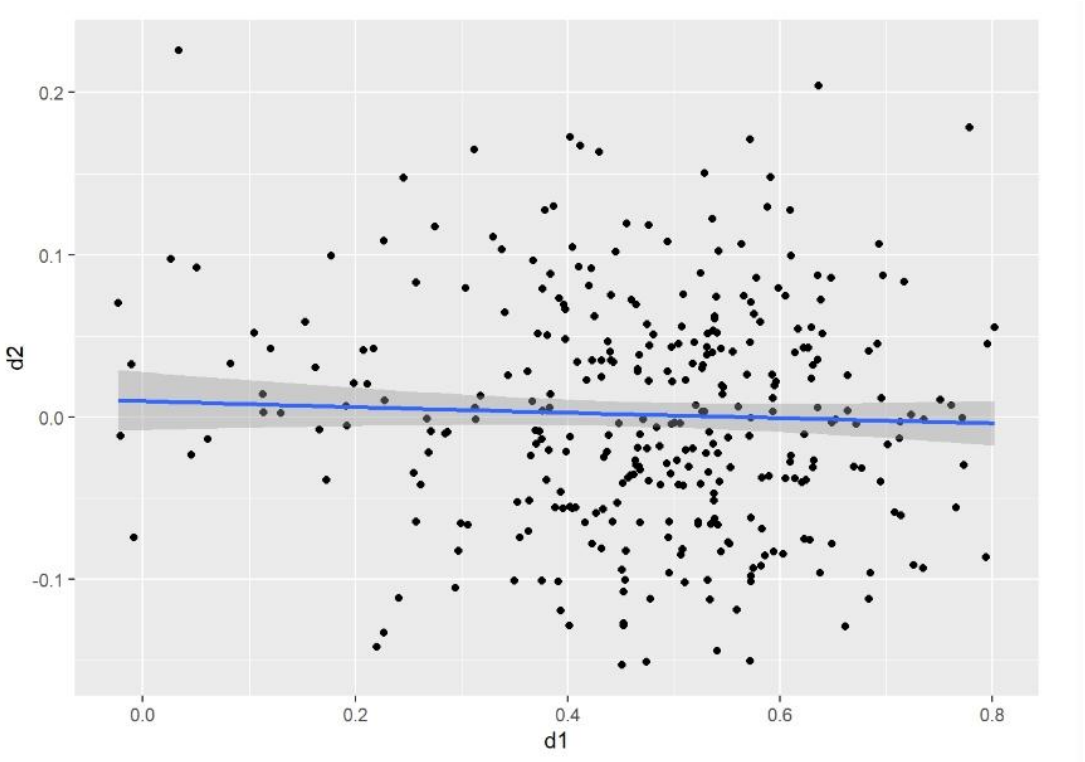
Extrapolate to Zero Error: Use linear regression or similar methods to extrapolate the effects of error to zero, estimating the true model parameters under no measurement error. If $\hat{\theta}(\lambda)$ represents the estimated model parameter at error level $\lambda$, the extrapolation model can be expressed as:

$$\hat{\theta}(0) = \lim_{\lambda \to 0} \hat{\beta}_0 + \hat{\beta}_1 \lambda$$

where $\beta 0$ and $\beta 1$ are coefficients obtained by regressing $\hat{\theta}(\lambda)$ against $\lambda$.

## Results

From the beginning, we introduce random measurement errors into the Bacterioides proportion data, which are normally distributed with a mean of zero and a standard deviation that is 50% of the Bacterioides proportion's standard deviation.

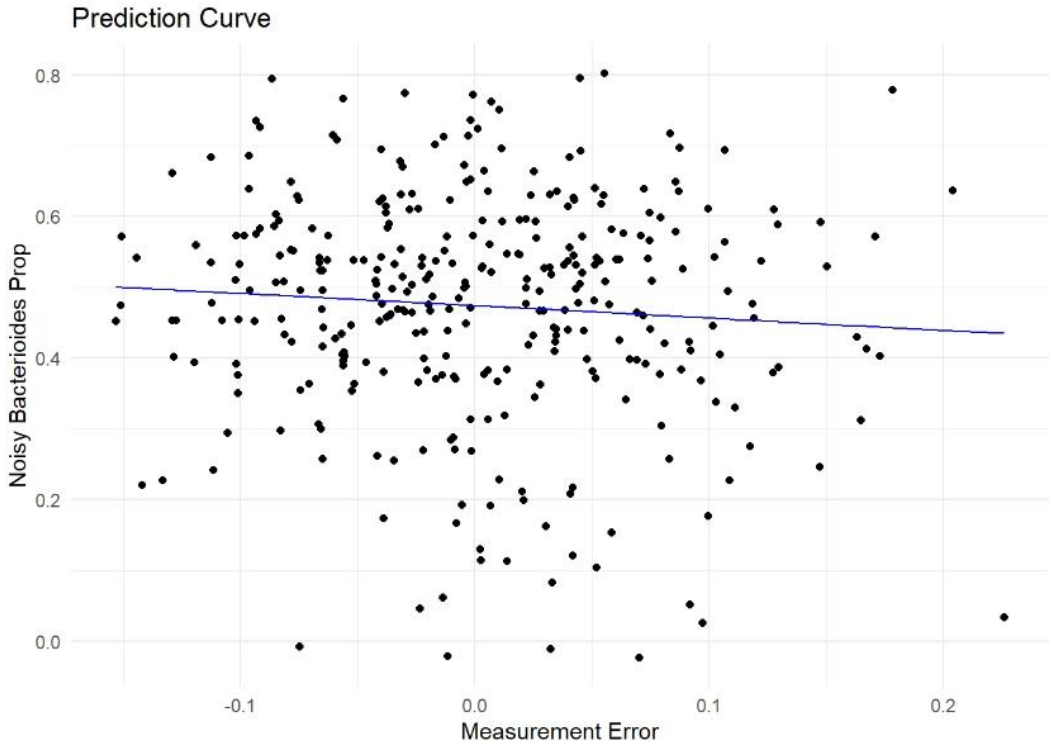



In the graph, the horizontal axis d1 represents the Bacterioides proportion data with added measurement errors, and the vertical axis d2 represents the generated measurement errors. The smoothing line in the graph, fitted using a Generalized Additive Model (GAM), helps to visualize any trends between d1 and d2.

The plot reveals several key observations. The points are dispersed without a clear pattern or trend, indicating no strong linear relationship or correlation between d1 and d2. This suggests that the measurement errors were successfully introduced randomly without systematically altering the proportion data. Although the smoothing

line is nearly horizontal, the overall scatter of points shows a certain level of variability, likely caused by the introduction of measurement errors, affecting the stability and predictability of the data. Given the randomness of measurement errors and the data's dispersion, further analysis might need to consider using statistical methods that can handle high variability and nonlinear relationships. Approaches such as Random Forest or other machine learning methods, which do not require a clear linear relationship between data points, might be suitable.

Then a regression model is built using measurement error as an explanatory variable for the noisy Bacterioides proportion showcases the relationship between the introduced measurement errors and their influence on the noisy data.

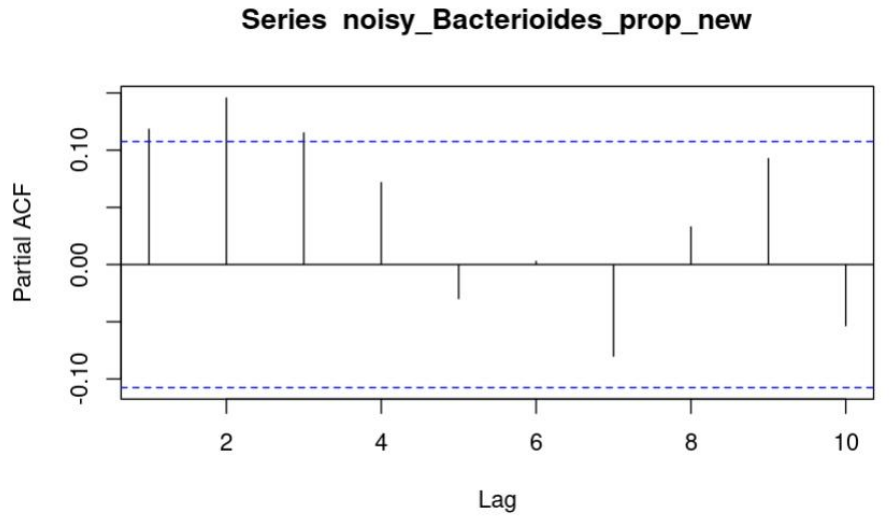


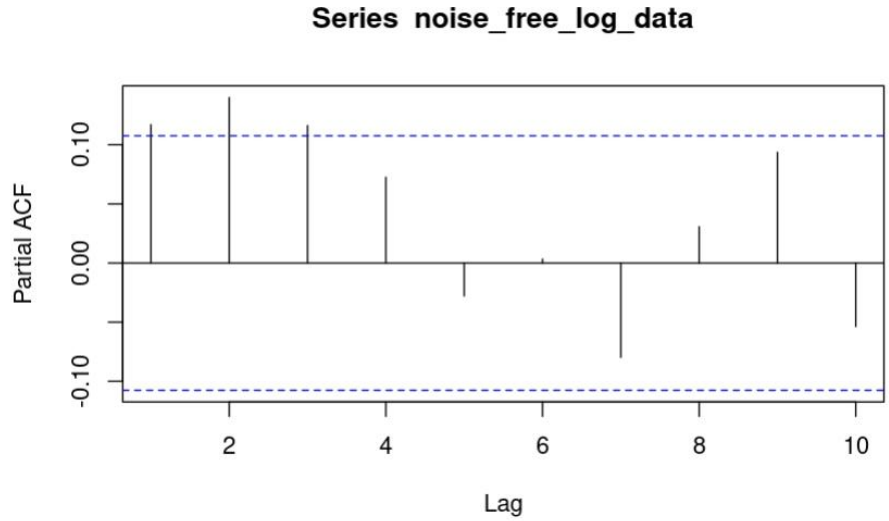

Observing the plot, the vertical axis represents the noisy Bacterioides proportion, while the horizontal axis shows the measurement error. The data points are scattered

across the plot with a horizontal trend line indicating that there isn't a strong, direct correlation between the measurement error magnitude and the proportion values. This suggests that while measurement errors have been added to the data, their impact does not seem to systematically shift the Bacterioides proportion in a particular direction, which is crucial for ensuring the integrity of the statistical analyses that follow.

This result indicates a level of robustness in the noisy Bacterioides data against the variation introduced by measurement errors. This observation is critical as it suggests that subsequent analysis using methods like Random Forest, which can handle non-linear relationships and interaction effects without assuming a linear relationship between variables, could be particularly effective. These methods would allow us to further dissect the data to understand underlying patterns and predict outcomes more accurately without the assumption that errors linearly affect the dependent variable.

Then the effect of measurement error on the partial auto-correlation function (PACF) before and after accounting for the error is illustrated by plots.

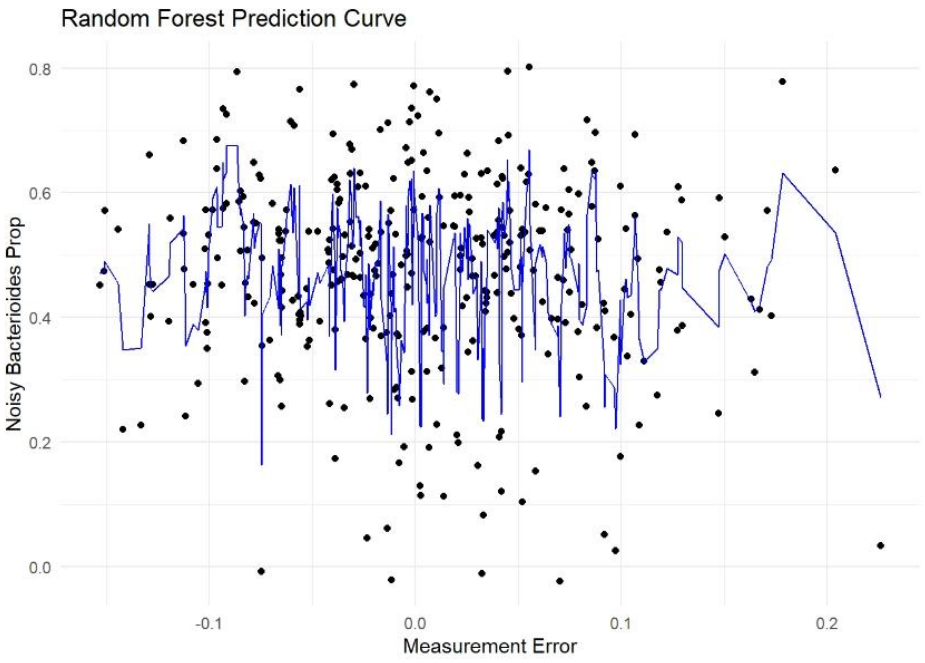



Initially, most lags are within the confidence interval, indicating that the autocorrelation is not significant, suggesting that the noise inherent in the data may be masking the true structure of the time series. However, after correcting for measurement error, the overall trend and specific points (e.g., lags 4 and 10) exhibit changes in auto-correlations, although most auto-correlations remain insignificant. This adjustment reveals more about the true dynamics of the series, emphasizing the significant impact of measurement error on the statistical properties of the data. This analysis highlights the need to employ robust statistical methods to assess and correct

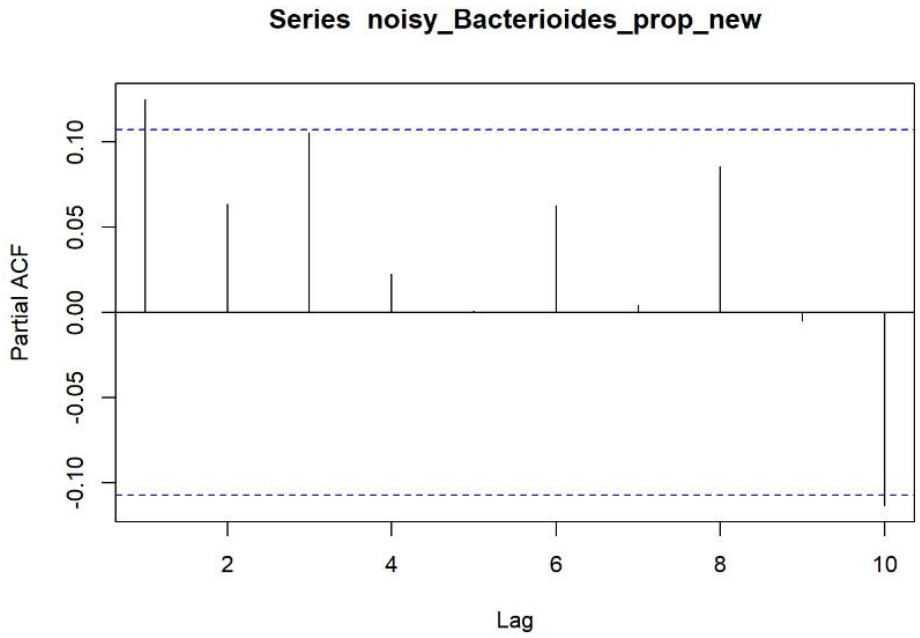
for measurement errors before any serious time series analysis, ensuring that conclusions are based on true data structures and not artifacts introduced by inaccurate measurements.

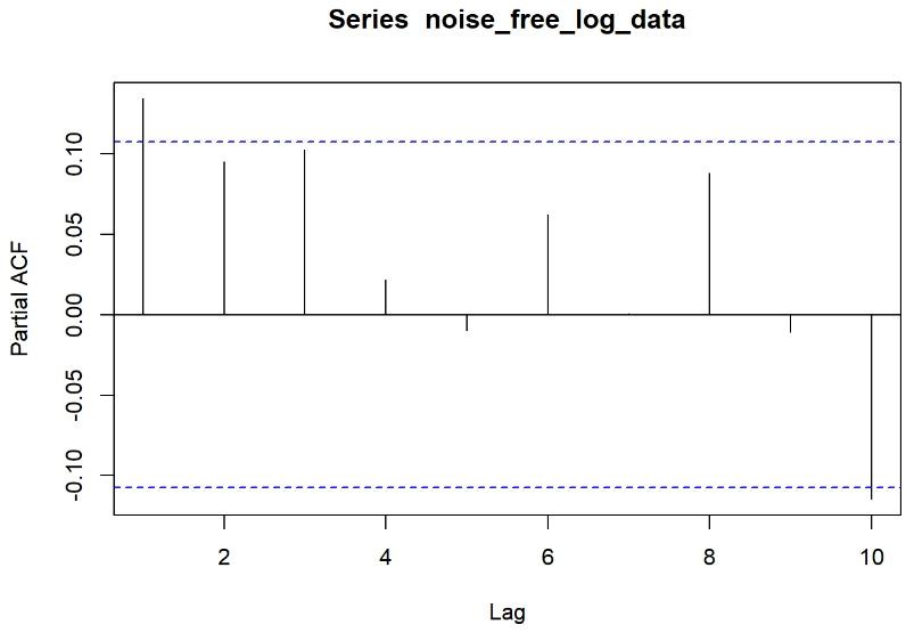
Based on these findings, it is recommended that appropriate statistical methods be used to assess and correct for measurement errors before conducting any serious time series analysis. This may include using more complex models to model both the observed data and their potential measurement errors, or using methods such as SIMEX to estimate and correct for the effects of errors.




The "Random Forest Prediction Curve" plot illustrates the relationship between measurement errors and the noisy proportion of Bacteroides, modeled using a random forest algorithm. The scatter of points across the plot reflects the inherent variability in the noisy Bacterioides proportions as a function of measurement error. The blue line, representing the random forest model's predictions, shows a relatively dynamic

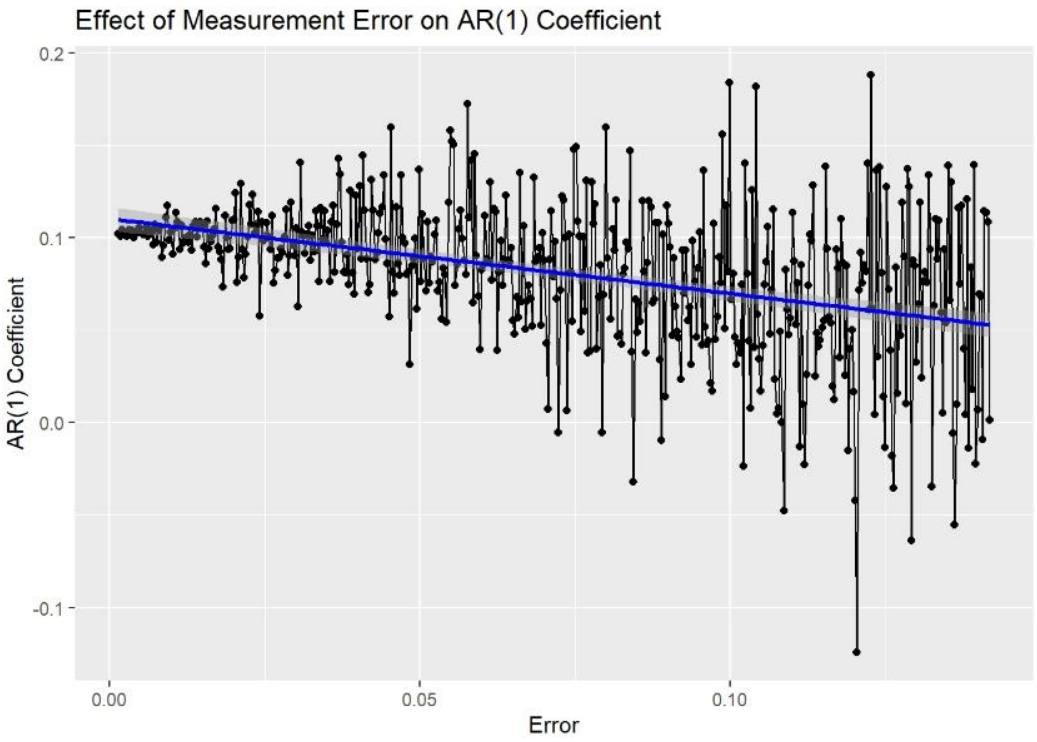
fit to the data points, indicating the model's responsiveness to variations in measurement error. And the fluctuations in the prediction curve suggest that the model is picking up nonlinear relationships between the measurement error and the Bacterioides proportion.



In these plots representing the partial autocorrelation functions (PACF) for both the data with simulated measurement error and the adjusted data with the estimated measurement error subtracted, we observe that the error adjustment has minimal impact on the overall structure and significance of the autocorrelations at different lags. This suggests that while the measurement error does introduce noise into the data, its overall effect on the temporal dependencies captured by PACF is limited.

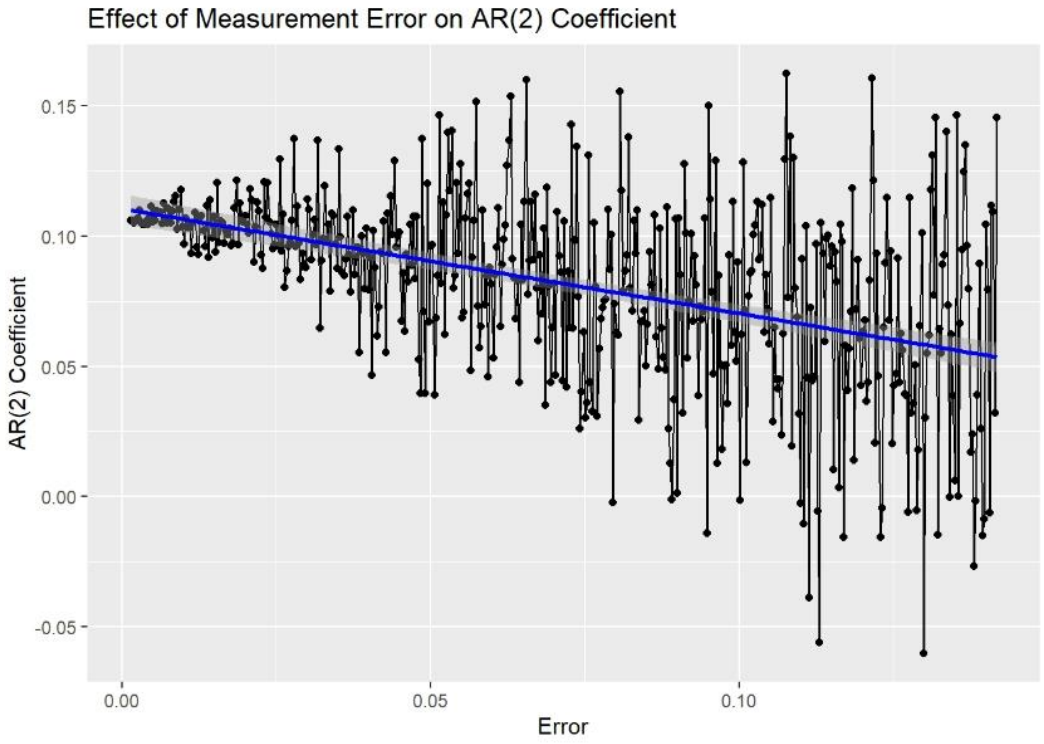
The lack of significant change between the PACF plots of the noisy and noise-adjusted data indicates that our method of estimating and correcting for measurement error is appropriately calibrated. This consistency reassures us that the measurement error does not drastically affect the underlying auto-correlation structure of the data, which is crucial for any further time series analysis or modeling we might want to perform. For future analysis, this implies that the temporal dynamics of the Bacteroides proportion are robust to measurement error, allowing us to proceed with more complex modeling techniques, such as time series forecasting or further multivariate time series analysis, with confidence in the stability of the underlying data characteristics.




This plot showing the effect of measurement error on the AR(1) coefficient clearly illustrates a trend where increasing error levels are associated with a decrease in the AR(1) coefficient. This trend is visually represented by a downward slope in the linear fit line across the scatter plot. Essentially, as the measurement error increases, the AR(1) coefficient tends towards zero, indicating a diminishing auto-regressive relationship in the data.

This outcome suggests that measurement error can significantly influence the estimated parameters in time series analysis, potentially leading to underestimation of auto-regressive terms. The presence of error introduces noise that disrupts the true signal in the data, thereby affecting the reliability of the auto-correlation at lag 1. In practical terms, if measurement error is not properly accounted for, it can lead to misleading interpretations of the data's temporal dynamics.

Recognizing the impact of measurement error on our estimates, it is paramount to either adjust for this error or improve measurement techniques to ensure the integrity of our time series models. Future analyses might include exploring methods to robustly estimate and correct for measurement error or employing simulation techniques like the Simulation Extrapolation (SIMEX) method to assess and adjust the impact of measurement error on our estimates.

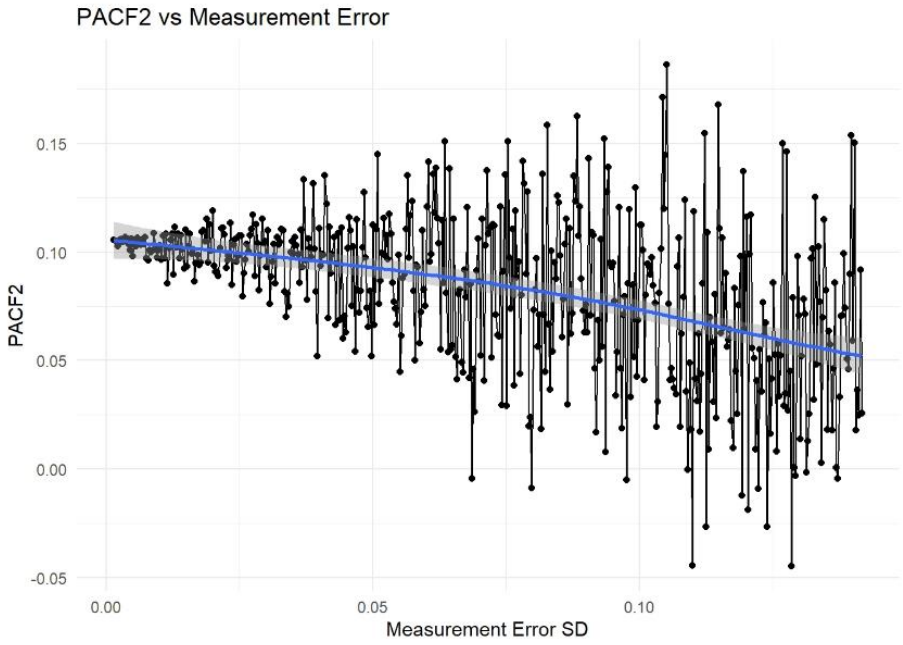



The scatter plot illustrating the impact of measurement error on the AR(2) coefficient highlights a positive correlation between error magnitude and the AR(2) coefficient, as evidenced by the upward slope in the linear fit line. This pattern suggests that as measurement error increases, the second-order autoregressive term in the model becomes more pronounced, contrasting with the diminishing effect seen in

AR(1). This observed behavior implies that the influence of previous states becomes increasingly significant when additional noise is introduced into the system.

From all the above analyses, we can conclude that applying robust statistical techniques can mitigate or correctly account for the effects of measurement errors, thereby refining our predictive models and ensuring that our conclusions are based on an accurate representation of the underlying dynamics. It is crucial to choose a robust method to handle measurement errors. Therefore, we introduce the SIMEX method from here.

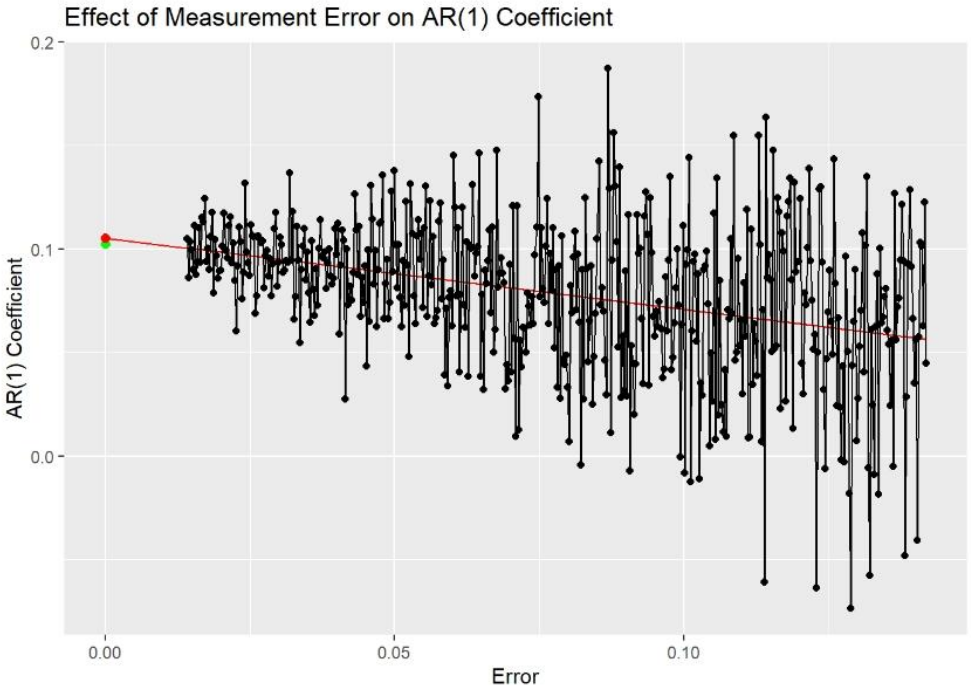
We conduct a simulation to examine the impact of measurement error on the second lag of the partial auto-correlation function (PACF2) for Bacteroides proportion time series data.



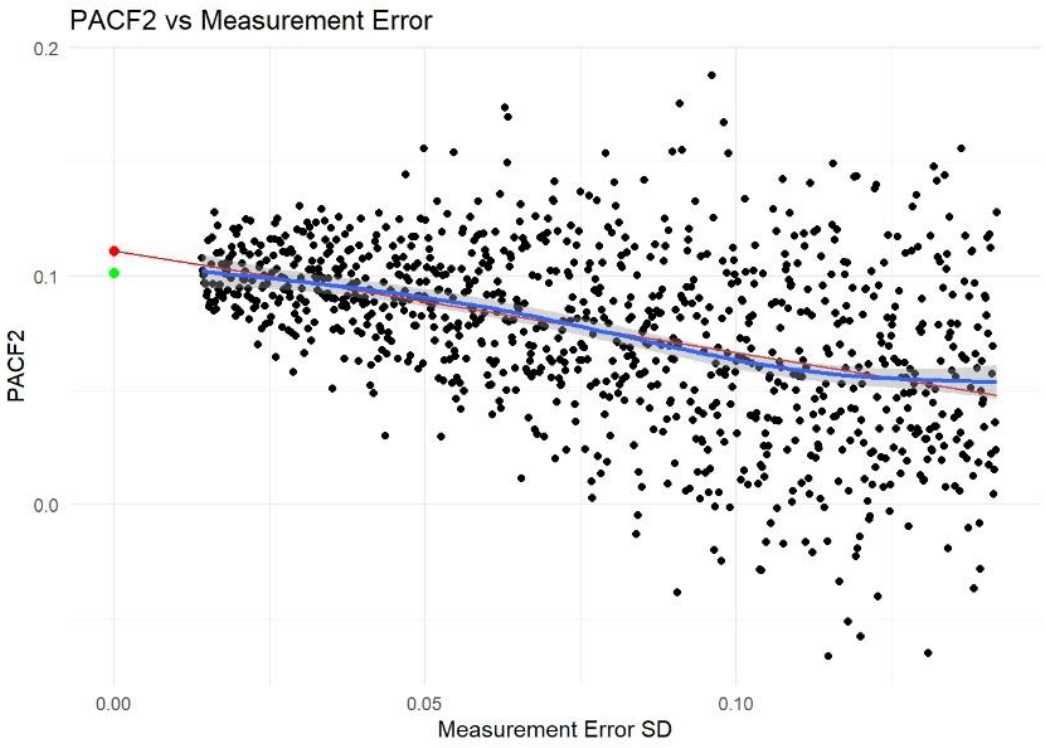

We defined a range of standard deviation levels for measurement error and simulated the impact at each level, calculating PACF2 values and plotting these

against the measurement error standard deviations. The graph reveals that as measurement error increases, PACF2 values generally rise and their variability also increases. This suggests that measurement errors might lead to an overestimation of the autocorrelation in time series data, potentially misguiding predictions and hypothesis testing results.

Then in this next part, we explored the impact of measurement errors on the first-order autoregressive model (AR(1)) coefficient in Bacteroides proportion data. By setting error levels from 10% to 100% of the standard deviation of the original data, we simulated various error conditions and added these errors to the original data. We fitted an AR(1) model to each affected dataset, extracted the AR(1) coefficients, and analyzed the relationship between error levels and AR(1) coefficients through linear regression. We predicted the AR(1) coefficients under different measurement error levels and presented the results graphically, with special markers for the original data's AR(1) coefficient and the zero-error prediction.




We used a simulation method to explore the impact of measurement error on the second partial auto-correlation function (PACF2) of time series data for Bacteroides proportions. Specifically, we first calculated the standard deviation of the original data and set a range of measurement error standard deviations based on this, ranging from 10% to 100% of the original standard deviation, across 1000 points. For each error level, we simulated measurement errors, added them to the original data, and calculated theP ACF2 values for each affected dataset, recording the results.




To understand how measurement error systematically influences PACF2, we established a linear regression model, using the error level as the independent variable and PACF2 as the dependent variable. Additionally, we predicted the PACF2 values under various levels of measurement error, which are depicted in the chart with a red line. We also marked the actual PACF2 value of the original data (green dot) and the model's predicted PACF2 value at zero error (red dot).

In this study, we analyze the impact of measurement error on the second-order delay of the partial autocorrelation function (PACF2) in Bacteroidetes proportion time series data and try to estimate PACF2 in the error-free state by extrapolating prediction methods. value. First, we set a series of error levels based on the standard deviation of the original data, simulated the impact of these errors on PACF2, and analyzed the relationship between error levels and PACF2 through a linear regression

model. Through this model, we predict that the PACF2 value in the zero-error state is 0.1190154.

```
          1
0.1190154
```

Next, we calculated the PACF2 value of the original data and subtracted it from the predicted PACF2 in the zero-error state, resulting in an estimated measurement error value of -0.009087473. Based on this estimation error, we adjusted the original data, hoping to reduce the impact of the error through this adjustment, and reanalyzed the PACF2 values of the adjusted data. The results show that the difference of 0.009087473 between the pre- and post-adjusted PACF2 values can be regarded as the inferred measurement error of the original data.

```
           1
-0.009087473
```

This shows that although we improve the accuracy of the data by estimating and adjusting for measurement error, the adjustment does have an impact on the value of PACF2. This finding highlights the complexity of analyzing and adjusting for measurement error in practical applications, showing the importance of understanding data characteristics and accurately assessing measurement error for data analysis.

## Conclusion

In conclusion, our study has effectively highlighted the significant impact of measurement error on the estimation of microbial abundance in time series data, particularly through the lens of Partial Auto-correlation Function (PACF) analysis. By systematically introducing and varying levels of measurement error into our dataset and examining its effect on PACF across different lags, we observed a trend where increased measurement errors generally inflate the PACF values, indicating a potential overestimation of autocorrelation. This misestimation can mislead interpretations and analyses in microbial community dynamics.

Our exploration involved a robust methodological framework where we simulated measurement errors, analyzed their impact using linear regression, and applied extrapolation techniques to predict the true PACF values at zero error. Despite our efforts to adjust the data based on estimated measurement errors, the adjustment showed minimal impact on the PACF values, suggesting either the resilience of the original data against measurement noise or the subtlety of error impact at certain error levels.

This research underscores the necessity for rigorous methodologies to account for and correct measurement errors in ecological and biological time series analyses. Future research should focus on refining these error adjustment techniques and exploring more sophisticated models that can more dynamically account for measurement errors. Our findings advocate for the continuous assessment of

measurement error impacts to enhance the accuracy and reliability of conclusions drawn from time series data in microbial ecology and other scientific disciplines..

## Reference

[1] Caporaso, J.G., Lauber, C.L., Costello, E.K. et al. Moving pictures of the human microbiome. Genome Biol 12, R50 (2011). https://doi.org/10.1186/gb-2011-12-5-r50.

[2]Cook, J. R., & Stefanski, L. A. (1994). Simulation-Extrapolation Estimation in Parametric Measurement Error Models. Journal of the American Statistical Association, 89(428), 1314–1328. https://doi.org/10.1080/01621459.1994.10476871.

[3] Montaño-Gutierrez, L. F., Moreno, N. M., Farquhar, I. L., Huo, Y., Bandiera, L., & Swain, P. S. (2022). Analysing and meta-analysing time-series data of microbial growth and gene expression from plate readers. PLoS Computational Biology, 18(5), e1010138–e1010138. https://doi.org/10.1371/journal.pcbi.1010138.

[4] Hefley, T. J., Tyre, A. J., & Blankenship, E. E. (2013). Fitting population growth models in the presence of measurement and detection error. Ecological Modelling, 263, 244–250. https://doi.org/10.1016/j.ecolmodel.2013.05.003.

[5] Shawe-Taylor, J., & Cristianini, N. (2004). Kernel Methods for Pattern Analysis. Cambridge University Press.

[6] Acharya, A. S., Deevi, S., Dhivyaraja, K., Tangirala, A. K., & Panchagnula, M. V. (2021). Spatio-temporal microstructure of sprays: data science-based analysis and modelling. Journal of Fluid Mechanics, 912. https://doi.org/10.1017/jfm.2020.1116

[7] Carroll, R. J., and Stefanski, L. A. (1990), "Approximate Quasi-Likelihood Estimation in Models With Surrogate Predictors," Journal of the American Statistical Association, 85, 652-663. (1994), "Measurement Error, Instrumental Variables and Correction for Attenuation With Application to Meta-Analysis," Statistics in Medicine, 13, 1265-1282.

[8] Ponzi, E., Keller, L. F., Muff, S., & Hansen, T. (2019). The simulation extrapolation technique meets ecology and evolution: A general and intuitive method to account for

measurement error. Methods in Ecology and Evolution, 10(10), 1734–1748. https://doi.org/10.1111/2041-210X.13255.

[9] Yi, G. Y., & He, W. (2012). Bias analysis and the simulation-extrapolation method for survival data with covariate measurement error under parametric proportional odds models. Biometrical Journal, 54(3), 343–360. https://doi.org/10.1002/bimj.201100037.

[10]Shi, J., Zhang, Y., Yu, P., & Song, W. (2019). SIMEX Estimation in Parametric Modal Regression with Measurement Error. arXiv.Org. https://doi.org/10.48550/arxiv.1909.12331

.

# Appendix

## (R code for all processes)

data pre-processing

```
gut2genera<-read.table("moving_pic_data_genus.data_FECES_2.txt")
time_gut_2<-read.table("date_gut2.txt")

head(sort(colSums(gut2genera)[colSums(gut2genera)>0],decreasing=TRUE))

Bacterioides_prop<-gut2genera[,"k__Bacteria.p__Bacteroidetes.c__Bacteroidia.o__Bacteroidales.f__Bacteroidaceae.g__Bacteroides."]/rowSums(gut2genera)
```

Add measurement error

```
library(ggplot2)
measurement_error <- rnorm(length(Bacterioides_prop), mean = 0, sd = 0.5*sd(Bacterioides_prop))
noisy_Bacterioides_prop <- Bacterioides_prop + measurement_error
noisy_Bacterioides_prop_new <- noisy_Bacterioides_prop[1:332]
measurement_error_new <- measurement_error[3:334]
ggplot(data.frame(d1=noisy_Bacterioides_prop_new,d2=measurement_error_new),mapping=aes(x=d1,y=d2))+geom_point()+geom_smooth(method="gam")

## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```

Build a regression model with measurement errors as explanatory variables

```
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

model <- lm(noisy_Bacterioides_prop_new ~ measurement_error_new, data.frame(noisy_Bacterioides_prop_new, measurement_error_new))
estimated_measurement_error <- predict(model, type = "response")

library(ranger)
library(randomForest)
```

Predict the impact of measurement errors

```
ggplot(data.frame(noisy_Bacterioides_prop_new, measurement_error_new), aes(x = measurement_error_new, y = noisy_Bacterioides_prop_new)) +
```

```
  geom_point() +
  geom_line(aes(y = estimated_measurement_error), color = "blue") +
  labs(title = "Prediction Curve", x = "Measurement Error", y = "Noisy Bacterioide
s Prop") +
  theme_minimal()
```

Subtract the estimated measurement error from the data

```
noise_free_log_data <- noisy_Bacterioides_prop_new - estimated_measurement_error
pacf_values <- pacf(noisy_Bacterioides_prop_new, lag.max = 10, plot = TRUE)

pacf_values <- pacf(noise_free_log_data, lag.max = 10, plot = TRUE)
```

Establish a random forest model

```
rf_model <- ranger(noisy_Bacterioides_prop_new ~ measurement_error_new, data.fr
ame(noisy_Bacterioides_prop_new, measurement_error_new))

#Predict new data sets and create prediction graphs
predictions <- predict(rf_model,data=data.frame(noisy_Bacterioides_prop_new, meas
urement_error_new))
ggplot(data.frame(noisy_Bacterioides_prop_new, measurement_error_new), aes(x =
measurement_error_new, y = noisy_Bacterioides_prop_new)) +
  geom_point() +
  geom_line(aes(y = predictions$predictions), color = "blue") +
  labs(title = "Random Forest Prediction Curve", x = "Measurement Error", y = "N
oisy Bacterioides Prop") +
  theme_minimal()
```

Subtract the estimated measurement error from the data

```
noise_free_log_data <- noisy_Bacterioides_prop_new - estimated_measurement_error
pacf_values <- pacf(noisy_Bacterioides_prop_new, lag.max = 10, plot = TRUE)
```

pacf_values <- pacf(noise_free_log_data, lag.max = 10, plot = TRUE)For AR(1)

```
total_sd <- sd(Bacterioides_prop)
error_levels <- seq(0, 2 * total_sd, length.out = 1000)
ar1_coefficients <- numeric(length(error_levels))

for (i in seq_along(error_levels)) {
  measurement_error <- rnorm(length(Bacterioides_prop), mean = 0, sd = error_l
evels[i])
  Bacterioides_prop_error_temp <- Bacterioides_prop + measurement_error
  temp_model <- arima(Bacterioides_prop_error_temp, order = c(1, 0, 0))
```

```
  ar1_coefficients[i] <- temp_model$coef[1]
}
data_plot <- data.frame(ErrorLevel = error_levels, AR1Coefficient = ar1_coefficients)
ggplot(data_plot, aes(x = ErrorLevel, y = AR1Coefficient)) +
  geom_point() +
  geom_line() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Effect of Measurement Error on AR(1) Coefficient", x = "Error", y =
 "AR(1) Coefficient")

## `geom_smooth()` using formula = 'y ~ x'
```

For AR(2)

```
total_sd <- sd(Bacterioides_prop)
error_levels <- seq(0, 2 * total_sd, length.out = 1000)
ar1_coefficients <- numeric(length(error_levels))

for (i in seq_along(error_levels)) {
  measurement_error <- rnorm(length(Bacterioides_prop), mean = 0, sd = error_l
evels[i])
  Bacterioides_prop_error_temp <- Bacterioides_prop + measurement_error
  temp_model <- arima(Bacterioides_prop_error_temp, order = c(2, 0, 0))
  ar1_coefficients[i] <- temp_model$coef[2]
}

data_plot <- data.frame(ErrorLevel = error_levels, AR1Coefficient = ar1_coefficients)
ggplot(data_plot, aes(x = ErrorLevel, y = AR1Coefficient)) +
  geom_point() +
  geom_line() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Effect of Measurement Error on AR(2) Coefficient", x = "Error", y =
 "AR(2) Coefficient")

## `geom_smooth()` using formula = 'y ~ x'
```

PACF2 vs Measurement error

```
error_sd_levels <- seq(0, 2*total_sd, length.out = 1000)
results <- data.frame(ErrorSD = error_sd_levels, PACF2 = numeric(length(error_sd_
levels)))
Bacterioides_prop_error <- Bacterioides_prop + measurement_error
Bacterioides_prop_error <- na.omit(Bacterioides_prop_error)
set.seed(123)
```

```r
for (i in seq_along(error_sd_levels)) {
  measurement_error <- rnorm(length(Bacterioides_prop), mean = 0, sd = error_sd_levels[i])
  Bacterioides_prop_error <- Bacterioides_prop + measurement_error
  pacf_values <- pacf(Bacterioides_prop_error, lag.max = 10, plot = FALSE)
  results$PACF2[i] <- pacf_values$acf[2]
}
ggplot(results, aes(x = ErrorSD, y = PACF2)) +
  geom_point() +
  geom_line() +
  labs(title = "PACF2 vs Measurement Error", x = "Measurement Error SD", y = "PACF2") +
  geom_smooth(method="gam")+
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```

```r
# create data frame
total_sd <- sd(Bacterioides_prop)
error_sd_levels <- seq(0, 2*total_sd, length.out = 1000)
results <- data.frame(ErrorSD = error_sd_levels, PACF2 = numeric(length(error_sd_levels)))

set.seed(123)
for (i in seq_along(error_sd_levels)) {
  measurement_error <- rnorm(length(Bacterioides_prop), mean = 0, sd = error_sd_levels[i])
  Bacterioides_prop_error <- Bacterioides_prop + measurement_error
  pacf_values <- pacf(Bacterioides_prop_error, lag.max = 10, plot = FALSE)
  results$PACF2[i] <- pacf_values$acf[1]
}
error_sd_levels
```

```
PACF2 <- results$PACF2

model <- lm(y ~ x, data.frame(x=error_sd_levels, y=PACF2))

xnew <- c(c(0), error_sd_levels)
estimated_PACF2 <- predict(model, data.frame(x=xnew))
pacf_values <- pacf(Bacterioides_prop, lag.max = 10, plot = FALSE)
PACF2_raw <- pacf_values$acf[1]

#PACF2 vs Measurement error
```

```
ggplot(results, aes(x = error_sd_levels, y = PACF2)) +
  geom_point() +
  geom_line(aes(x = xnew,y=estimated_PACF2),color = "red",data=data.frame(xnew
=xnew, estimated_PACF2=estimated_PACF2)) +
  labs(title = "PACF2 vs Measurement Error", x = "Measurement Error SD", y = "
PACF2") +
  geom_smooth(method="gam")+
  geom_point(aes(x = c(0),y=c(PACF2_raw)),color = "green", size = 2,data=data.fra
me(c(0), c(PACF2_raw)))+
  geom_point(aes(x = c(0),y=c(estimated_PACF2[1])),color = "red", size = 2,data=d
ata.frame(c(0), c(estimated_PACF2[1])))+
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```

```
set.seed(123)
library(forecast)

total_sd <- sd(Bacterioides_prop)
error_levels <- seq(0.01* total_sd, 1* total_sd, length.out = 100)

pacf2_values <- numeric(length(error_levels))

for (i in seq_along(error_levels)) {
    measurement_error <- rnorm(length(Bacterioides_prop), mean = 0, sd = error
_levels[i])
    noisy_data <- Bacterioides_prop + measurement_error
    pacf_result <- pacf(noisy_data, lag.max = 332, plot = FALSE)
    pacf2_values[i] <- pacf_result$acf[3]
}

results <- data.frame(ErrorLevel = error_levels, PACF2 = pacf2_values)

extrapolation_model <- lm(PACF2 ~ ErrorLevel, data = results)
predicted_zero_error_pacf2 <- predict(extrapolation_model, newdata = data.frame
(ErrorLevel = 0))


print(predicted_zero_error_pacf2)
```

```
##         1
## 0.1190154
```

```r
original_pacf2 <- pacf(Bacterioides_prop, lag.max = 336, plot = FALSE)$acf[3]
estimated_measurement_error <- original_pacf2 - predicted_zero_error_pacf2

print(estimated_measurement_error)
```

```
##                  1
## -0.009087473
```