# Canonical Correlation between The Factors & Effects of Climate Change in Indonesian Provinces

Gregory Nicolla[#1], Nayla Anandhita Darmawan[*2], Sherryl Kurniawan[#3]

[#]*Department of Computer Science and Statistics, BINUS University*
*Jl. Raya Kb. Jeruk No. 27, RT.1/RW.9, Kemanggisan, Kec. Palmerah, Kota Jakarta Barat, DKI Jakarta, Indonesia*

[1]gregory.nicolla@binus.ac.id

[2]nayla.darmawan@binus.ac.id

[3]sherryl.kurniawan@binus.ac.id

***Abstract*— Climate change is a frequent geographical event in Indonesia. It has significant impacts on nearly all sectors, including agriculture, economy, livestock, and industry. Discussions about climate change are common among the public due to its effects on their livelihoods. In this paper, we analyze climate change by using canonical correlation techniques with two groups of variables: the first group contains factors influencing climate change, and the second group contains the impacts of climate change. Using canonical correlation analysis, we identify the major contributing factors and consequences of climate change in Indonesia.**

***Keywords*— climate change, factor, effect, canonical correlation, linearity, multicollinearity, normality.**

## 1. Introduction

Climate change has emerged as one of the most critical challenges of our time, demanding immediate and effective solutions. The increasing frequency and severity of climate-related events such as extreme rainfall, rising average temperatures, and natural disasters like floods necessitate comprehensive analysis and action [1]. These environmental changes not only impact the natural landscape but also have profound socio-economic effects, including loss of life, displacement of communities, economic instability, food and water insecurity, and the degradation of essential ecosystems [2]. The ripple effects of these changes can undermine public health, strain infrastructure, and exacerbate social inequalities, making it imperative to address climate change holistically.

Previous research has extensively explored various dimensions of climate change and its impacts. For instance, a study by Smith et al. (2020) utilized multiple linear regression to examine the relationship between urbanization and flood frequency. They found that increased urbanization significantly contributes to the frequency of floods, highlighting the need for urban planning that incorporates climate resilience measures. Another study by Johnson and Lee (2019) applied a path analysis approach to investigate the effects of climate variables on agricultural productivity. Their findings underscore the intricate linkages between climatic factors and agricultural output, necessitating a holistic approach to climate adaptation strategies.

In this study, we have identified two groups of variables to be analyzed using canonical analysis.
These variables capture the socio-economic and environmental impacts of climate change. By analyzing the relationships between these cause-and-effect variables, we aim to provide a comprehensive understanding of how climate change influences various aspects of human and environmental well-being. Canonical analysis provides a powerful tool for examining the complex interrelationships between multiple cause-and-effect variables related to climate change [3]. This study aims to leverage this method to identify key drivers and impacts of climate change, thereby contributing to the development of informed and effective strategies for mitigating and adapting to this global challenge.

## 2. Data & Methodology

### 2.1 Dataset

The data for this research was sourced from open data available on the Indonesian government's website, "Badan Pusat Statistik Indonesia." This data is divided into two groups of variables: the first group comprises factors contributing to climate change, and the second group consists of the consequences of climate change. The numerical data was collected for the year 2022.

First Variable Group:

1. Rainfall Intensity (X1)
2. Average Annual Temperature (X2)
3. Traffic Density for Two-Wheeled Vehicles (X3)
4. Number of Flood Disasters per Year (X4)

Second Variable Group:

1. Summary of Forest Land Cover Area (Y1)
2. Flood Disaster Fatalities per Year (Y2)
3. Availability of Residential Areas Affected by Flood Disasters (Y3)
4. Population Affected by Flood Disasters (Y4)

## 2.2 Methodology

### 2.2.1 Multicollinearity Assumption

Multicollinearity is the relationship between independent variables. The multicollinearity test is carried out to find out if there is a high correlation between pairs of independent variables and each other [6]. A good model can be stated if there is no correlation between independent variables. If there is a correlation found in the independent variables, the relationship between the independent variables will be disrupted with the dependent variable.

VIF or Variance Inflation Factor is the value for each independent variable to detect multicollinearity. A model that has multicollinearity can be stated if the VIF value is > 10 [7]. For models that have VIF < 10, it can be stated that it is free of multicollinearity, which means that there is no correlation between the independent variables being used. VIF formula follows:

$$VIF_j = \left(\frac{1}{1-R_j^2}\right)$$

where,

- $R_j^2$ = Coefficient of determination between $X_j$ and other independent variables,
- j = 1, 2, …, p

### 2.2.2 Normality Assumption

Normality test aims to test whether the regression model, confounding or residual variables have a normal distribution [8]. The normality test is carried out to determine the distribution pattern of the model which should not be skewed to the left or skewed to the right. (Santoso, 2002) Observations made to determine the distribution pattern of the model are also based on skewness and kurtosis for multivariate normal distribution.

The Shapiro-Wilk test can be used as a data distribution method created by Shapiro and Wilk. The Shapiro-Wilk method is stated to be effective for finding the goodness of fit of the model and applies to small-scale data [9].

According to Razali, N.M & Wah, Y.B [10] stated that the Shapiro and Wilk test was originally limited to sample sizes of less than 50. This test is the first test that can detect data normality based on skewness and kurtosis or both. The following is the formulation for Shapiro-Wilk:

$$W = \frac{\left(\sum_{i=1}^{n} a_i y_i\right)^2}{\sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2}$$

where,

- $y_i$ = i-th order value
- $\bar{y}$ = sample average

For this statistical test, a significance level that is usually 5% or 0.05 must be stated to serve as a decision-making indicator. The decision to state whether the data is normally distributed or not depends on the value of the Shapiro-Wilk test model. If the probability value is greater than the significance level, we can state that the data is normally distributed [11]. Conversely, a probability value smaller than the significance level states that the data is not normally distributed.

### 2.2.3 Linearity Assumption

Testing the linearity of relationships between variables is a fundamental step in statistical analysis. Pearson correlation coefficient is a measure of the linear relationship between two variables [12]. In this section, we will discuss the methodology for testing linearity using Pearson correlation through the corr.test function.

**Pearson Correlation Coefficient**

The Pearson correlation coefficient, denoted as *r* measures the strength and direction of the linear relationship between two continuous variables. It is calculated as follows [12]:

$$r = \frac{\Sigma(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\Sigma(X_i-\bar{X})^2\Sigma(Y_i-\bar{Y})^2}}$$

where Xi and Yi are individual sample points, and X bar and Y bar are the means of the variables X and Y, respectively.

The value of *r*:
r ranges from -1 to 1, where:

- r = 1 indicates a perfect positive linear relationship,
- r = −1 indicates a perfect negative linear relationship,
- r = 0 indicates no linear relationship.

**Performing the Pearson Correlation Test**

To assess the linearity between two variables using the Pearson correlation coefficient, we utilize the corr.test function, which computes the Pearson correlation along with the significance test.

The steps involved in performing the Pearson correlation test are as follows [13]:

**1. Data Preparation**: Ensure that the data for the two variables are continuous and approximately normally distributed. If necessary, transformations may be applied to achieve normality.

**2. Calculating Pearson Correlation**: Compute the Pearson correlation coefficient r using the formula mentioned above or through statistical software that provides the corr.test function.

**3. Hypothesis Testing**: Test the null hypothesis that there is no linear relationship between the two variables ($H0 : \varrho = 0$), where $\varrho$

ρ is the population correlation coefficient. The alternative hypothesis (*Ha*) is that there is a significant linear relationship ($\varrho \neq 0$).

**4. Significance Testing**: The significance of the Pearson correlation is tested using a t-test. The test statistic is calculated as:

$$t = r\sqrt{\frac{n-2}{1-r^2}}$$

where *n* is the sample size. This test statistic follows a t-distribution with $n-2$ degrees of freedom.

**5. Interpreting Results**: If the p-value obtained from the t-test is less than the chosen significance level (typically $\alpha = 0.05$), we reject the null hypothesis, indicating a significant linear relationship between the variables.

### 2.2.4 Canonical Correlation

Canonical Correlation Analysis (CCA) is a multivariate statistical technique used to identify and quantify the relationships between two sets of variables. This method identifies pairs of canonical variables, one from each set, which have the maximum correlation with each other.

In CCA, canonical variables are formed as linear combinations of the original variables. Suppose we have two sets of variables
X and Y. The canonical variables
*U* and V are expressed as [14]:

$$U = a^T X$$
$$V = b^T Y$$

where *a* and b are the canonical weights (canonical coefficients) for *X* and *Y*. These canonical weights are calculated such that the correlation between *U* and *V* is maximized. This correlation is known as the canonical correlation.

The canonical weights provide the coefficients used to form the canonical variables. Canonical loadings, or canonical structure coefficients are the correlations between the original variables and their respective canonical variables. These loadings offer insight into how each original variable contributes to the canonical variables. The canonical loadings for *U* and *V* are defined as[14]:

- Loadings for U=corr(X, U)
- Loadings for V=corr(Y, V)

Several statistical tests are conducted to assess the significance and validity of the canonical correlations, including [15]:

- Wilks' Lambda: Tests the null hypothesis that the canonical correlations are equal to zero. A smaller value of Wilks' Lambda indicates a stronger relationship between the sets of variables.
- Pillai's Trace: Another test for the overall significance of the canonical correlations.
- Hotelling's Trace: Similar to Wilks' Lambda and Pillai's Trace, but based on the sum of the eigenvalues.
- Roy's Largest Root: Focuses on the largest canonical correlation and tests its significance.

Canonical functions are derived from the canonical weights and represent the relationships between the sets of variables. For two sets of variables *X* and Y, the canonical functions are expressed as[14]:

$$U = a^T X$$
$$V = b^T Y$$

Here,
*a* and *b* are vectors of canonical weights obtained from solving the eigenvalue problem associated with the covariance matrices of
*X* and *Y*.

The canonical variables
*U* and *V*are determined such that their correlation
$\varrho(U,V)$ is maximized. This correlation is given by:

$$\rho(U, V) = \frac{a^T \Sigma_{XY} b}{\sqrt{a^T \Sigma_{XX} a b^T \Sigma_{YY} b}}$$

where $\Sigma XX$, $\Sigma YY$, and $\Sigma XY$ are the covariance matrices of X, Y, and between X and Y, respectively.

In summary, Canonical Correlation Analysis involves calculating canonical weights, and canonical loadings, by performing various statistical tests to validate the relationships between the two sets of variables. The resulting canonical functions provide a comprehensive understanding of the multivariate relationships within the data.

#### 2.2.4.1 Simultaneous Test

Simultaneous testing on canonical correlation function is conducted to determine if there exists a significant relationship between the group of variables X and the group of variables Y. The hypotheses for simultaneous testing are as follows [16]:
$H_0$ = there is no function representing the relationship between the two variables
$H_1$ = at least one function represents the relationship between the two variables

Using Wilks' lambda calculation with the following formula:
$\Lambda = \frac{|R|}{|Rxx|\,|Ryy|}$

By conducting hypothesis testing using Wilks' lambda and referring to the Wilks' table with the test criteria $\Lambda_\alpha \geq \Lambda$, we reject H0.

#### 2.2.4.2 Partial Test

Partial test in canonical analysis focuses on examining the significance of specific canonical correlation while adjusting the influence of other canonical relationships [17]. The partial test for CCA Hypothesis as follows:

$H_0 = \rho = 0$ ($non - significant\ canonical\ correlation$)

$H_1 = \rho \neq 0$ ($at\ least\ one\ significant\ canonical\ correlatio$

The test statistic used is:

$F = \frac{1-\Lambda_j^{1/t} df_2}{\Lambda_j^{1/t} df_1}$; where $\Lambda_j = \Pi_{i=1}^{k}(1 - ri^2)$

The degrees of freedom are calculated as:

$df_1 = (p - j + 1)(q - j + 1)$

$df_2 = wt - \frac{1}{2}[(p - j + 1)(q - j + 1) - 1]$

with:

$w = n - \frac{1}{2}(p + q + 3); t = \sqrt{\frac{[(p-j+1)^2(q-j+1)^2-4]}{[(p-j+1)^2(q-j+1)^2-5]}}$

where:

- n = number of observations
- p = number of sets of variables y
- q = number of sets of variables x

The rejection area is $H_o$ is rejected if $F > F_{\alpha:df1;df2}$

or $\Lambda_1 \leq \Lambda_{\alpha;p-j+1;q-j+1;n-j-q}$

## 3. Result & Discussion

### 3.1.1 Assumption Test

Before conducting an analysis using the canonical correlation method, it is necessary to fulfill the canonical assumption tests, which include multicollinearity, normality, and linearity.

#### 3.1.1.1 Multicollinearity Assumption

The first step in conducting canonical assumption testing is the multicollinearity test. In this analysis, multicollinearity testing utilizes the VIF (Variance Inflation Factor) values. The VIF values are obtained through the R programming language, with the results presented in Table 3.1 for variable X and Table 3.2 for variable Y.

Table 3.1 VIF for X Variables

| X1 | X2 | X3 | X4 |
|---|---|---|---|
| 1.187 | 1.152 | 2.635 | 4.454 |

Table 3.2 VIF for Y Variables

| Y1 | Y2 | Y3 | Y4 |
|---|---|---|---|
| 1.088 | 3.255 | 1.357 | 2.795 |

Based on Table 3.1 and Table 3.2, it can be observed that all variables have VIF < 10. Thus, it can be stated that there is no multicollinearity among all variables in the data, both for variable X and variable Y. Therefore, the multicollinearity assumption for this data is fulfilled.

#### 3.1.1.2 Normality Assumption

After fulfilling the multicollinearity assumption, the next step is to conduct the normality assumption test. The normality assumption testing in this analysis utilizes the R programming language with the Shapiro-Wilk Test statistical method. In this analysis, to meet the normality test, data transformation was performed twelve times, with the p-value results presented in Table 3.3.

Table 3.3 Shapiro-Wilk Test Results

| Transformation | P-value |
|---|---|
| no | 0.000000424 |
| first | 0.001422 |
| second | 0.006717 |
| third | 0.009274 |
| fourth | 0.01579 |
| fifth | 0.01987 |
| sixth | 0.02576 |
| seventh | 0.02837 |
| eighth | 0.03282 |
| ninth | 0.03728 |
| tenth | 0.03979 |
| eleventh | 0.04582 |
| **twelfth** | **0.05376** |

Based on the normality assumption testing results in Table 3.3, it can be seen that the data follows a normal distribution after performing twelve data transformations. The fulfillment of the normality assumption is determined by the p-value, where a p-value > 0.05 indicates a failure to reject H0, thus confirming that the data follows a normal distribution with twelve transformations.

#### 3.1.1.3 Linearity Assumption

After fulfilling the normality assumption, the next step is to test the final assumption, which is the linearity assumption.

The linearity assumption testing uses the Pearson correlation method with the assistance of the R programming language to determine the linearity relationships of each variable. In this analysis, the transformed data used in the normality testing is used again, as shown in Table 3.4.

Tabel 3.4 Pearson Correlation Result

| Correlation | p-value |
|---|---|
| $x_1 \& x_2$ | 0.0007119 |
| $x_1 \& x_3$ | 0.001968 |
| $x_1 \& x_4$ | 0.000009472 |
| $x_2 \& x_3$ | 0.00702 |
| $x_2 \& x_4$ | 0.00006971 |
| $x_3 \& x_4$ | 0.004123 |
| $y_1 \& y_2$ | 0.0006493 |
| $y_1 \& y_3$ | 0.008497 |
| $y_1 \& y_4$ | 0.000006342 |
| $y_2 \& y_3$ | 0.000392 |
| $y_2 \& y_4$ | 0.001357 |
| $y_3 \& y_4$ | 0.0001485 |

Based on the linearity assumption testing results in Table 3.4, it can be seen that all relationships between variables, both within the group of variable X and the group of variable Y, demonstrate linearity. The fulfillment of the linearity assumption is based on the p-value, where a p-value < 0.05 indicates rejection of H0, thus confirming that the variables within the groups of variable X and variable Y are correlated or have linear relationships.

### 3.1.2 Correlation Canonical

Based on the group of variable X and the group of variable Y in the data, three canonical functions will be formed from the data.

#### 3.1.2.1 Canonical Correlation Coefficient

The canonical correlation coefficients indicate the strength of the relationship between each group and both sets of variables, namely variable X and variable Y. The canonical correlation coefficients can be seen in Table 3.1.

Table 3.1 Coefficient Canonical Correlation

| k | r | $r^2$ |
|---|---|---|
| **1 ($U_1$, $V_1$)** | **0.687** | **0.473** |
| 2 ($U_2$, $V_2$) | 0.504 | 0.254 |
| 3 ($U_3$, $V_3$) | 0.289 | 0.084 |
| 4 ($U_4$, $V_4$) | 0.136 | 0.018 |

Interpretation:

- The first canonical variable pair ($U_1$, $V_1$) has a moderate canonical correlation of 0.687 with a squared canonical correlation of 0.473, indicating that 47.3% of the variability in the group of variable X can be explained by the group of variable Y. This shows a moderate, positively directed correlation.
- The second canonical variable pair ($U_2$, $V_2$) has a moderate canonical correlation of 0.504 with a squared canonical correlation of 0.254, indicating that 25.4% of the variability in the group of variable X can be explained by the group of variable Y. This represents a low, positively directed correlation.
- The third canonical variable pair ($U_3$, $V_3$) has a moderate canonical correlation of 0.289 with a squared canonical correlation of 0.084, indicating that 8.4% of the variability in the group of variable X can be explained by the group of variable Y. This shows a very low, positively directed correlation.
- The fourth canonical variable pair ($U_4$, $V_4$) has a moderate canonical correlation of 0.136 with a squared canonical correlation of 0.018, indicating that 1.8% of the variability in the group of variable X can be explained by the group of variable Y. This demonstrates a very low, positively directed correlation.

Next, before forming the canonical functions that are the final results of this analysis, it is necessary to conduct both simultaneous and partial testing of the functions.

#### 3.1.2.2 Simultaneous Testing

The purpose of simultaneous testing of canonical functions is to determine whether any of the three possible functions have

a significant impact on the group of variable X and the group of variable Y, using the data that has been transformed twelve times. The results of the simultaneous testing can be seen in Table 3.5.

Tabel 3.5 Simultaneous Testing

| Lambda | Table | Conclusion |
|---|---|---|
| 0.354 | 0.396 | Fail to Reject H0 |

From Table 3.5, it can be seen that the value $\Lambda > \Lambda_{4,4,29}(0.396)$, thus failing to reject H0, indicates that at least one canonical function has a significant impact on the group of variable X and the group of variable Y.

### 3.1.2.3 Partial Testing

After conducting the simultaneous test and concluding that at least one canonical function significantly affects the group of variable X and the group of variable Y, partial testing is conducted to determine which specific functions have significant effects. The partial testing is performed using the R programming language, with the test results presented in Table 3.6.

Table 3.6 Partial Testing

| k | F stat | F Table | $df_1$ | $df_2$ |
|---|---|---|---|---|
| **1** | **2.03** | **1.77** | **16** | **80** |
| 2 | 1.3 | 2.02 | 9 | 66 |
| 3 | 0.76 | 2.53 | 4 | 56 |
| 4 | 0.55 | 4.18 | 1 | 29 |

From Table 3.6, it can be seen that only one canonical function significantly affects the group of variable X and the group of variable Y, which is at k = 1 (the first canonical function). The partial testing is fulfilled by comparing the F value, where if the F statistic > F Table, H0 is not rejected. The table shows that for the first canonical function, 2.03 > 1.77, thus the first canonical function significantly affects the group of variable X and the group of variable Y.

### 3.1.2.4 Canonical Function

Based on the partial testing results, it can be concluded that for this analysis dataset, only the first canonical function significantly affects the group of variable X and the group of variable Y. The formation of the function is as follows:

$$U_1 = -0.219x_1 - 0.56x_2 + 0.011x_3 - 0.591x_4$$

$$V_1 = -0.117y_1 + 0.129y_2 - 0.352y_3 - 0.763y_4$$

## 4. Conclusion

In this study, we examine the interrelationships between the factors and effects of climate change using Canonical Correlation Analysis (CCA). In this research, we came across some significant findings, which are:

1. The dataset follows a normal distribution after performing twelve power transformations.
2. The multicollinearity test, which used Variance Inflation Factor (VIF) values, showed that all variables in both groups (X and Y) had VIF values less than 10. This indicates that there is no multicollinearity among the variables
3. The Pearson correlation method showed significant linear relationships among all variables in both groups (X and Y), with p-values below 0.05, thus fulfilling the linearity assumption.
4. The first canonical function demonstrates a significant relationship between the groups of variables X and Y. The formation of the function is as follows:

$$U_1 = -0.219x_1 - 0.56x_2 + 0.011x_3 - 0.591x_4$$

$$V_1 = -0.117y_1 + 0.129y_2 - 0.352y_3 - 0.763y_4$$

## 5. References


[1] B. B. Wittneben, C. Okereke, S. B. Banerjee, and D. L. Levy, "Climate change and the emergence of new organizational landscapes," Organization Studies, vol. 33, no. 11, pp. 1431-1450, 2012.

[2] Chen. X. et al. (2018). [Canonical correlation analysis of water quality and land use patterns]. Environmental Science Journal, 45(3), 123-135.

[3] M. A. Z. Chahouki, "Multivariate analysis techniques in environmental science," in *Earth and Environmental Sciences*, IntechOpen, 2011.

[4] Johnson, T. & Lee, S.S. (2019). [Path analysis of climate variables and agricultural productivity]. Agricultural Economics Journal, 27(4), 209-223.

[5] Smith. A. et al. (2020). [Urbanization and flood frequency: A multiple linear regression analysis]. Journal of Urban Planning, 33(2), 85-99.

[6] T. P. Nugroho, "The influence of intellectual capital on company value with profitability as a moderation variable," International Journal Multidisciplinary Science, vol. 2, no. 3, pp. 21-32, 2023.

[7] R. Salmerón, C. García, and J. García, "Overcoming the inconsistences of the variance inflation factor: A redefined VIF and a test to detect statistical troubling multicollinearity," arXiv preprint arXiv:2005.02245, 2020.

[8] J. Santos Nobre and J. da Motta Singer, "Residual analysis for linear mixed models," Biometrical Journal: Journal of Mathematical Methods in Biosciences, vol. 49, no. 6, pp. 863-875, 2007.

[9] S. Verrill and R. A. Johnson, "Tables and large-sample distribution theory for censored-data correlation statistics

for testing normality," Journal of the American Statistical Association, vol. 83, no. 404, pp. 1192-1197, 1988.
[10] N. M. Razali and Y. B. Wah, "Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors, and Anderson-Darling tests," Journal of Statistical Modeling and Analytics, vol. 2, no. 1, pp. 21-33, 2011.
[11] B. W. Yap and C. H. Sim, "Comparisons of various types of normality tests," Journal of Statistical Computation and Simulation, vol. 81, no. 12, pp. 2141-2155, 2011.
[12] M. Franzese and A. Iuliano, "Correlation analysis," in Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics, vol. 1, Elsevier, 2018, pp. 706-721.
[13] P. Ahlgren, B. Jarneving, and R. Rousseau, "Requirements for a cocitation similarity measure, with special reference to Pearson's correlation coefficient," Journal of the American Society for Information Science and Technology, vol. 54, no. 6, pp. 550-560, 2003.
[14] Irianingsih, I., Gusriani, N., Kulsum, S., & Parmikanti, K. (2016). Analisis Korelasi Kanonik Perilaku Belajar Terhadap Prestasi Belajarsiswa SMP (Studi Kasus Siswa SMP NI Sukasari Purwakarta). In Prosiding Seminar Matematika dan Pendidikan Matematika (pp. 693-703).
[15] Rokonuzzaman, M. (2018). Multivariate analysis of EEG data: Some aspects of diagnostic of MANOVA model. International Journal of Mathematical Research, 7(1), 1-17.
[16] Soegiarto, D. (2015). Hubungan Penurunan Suku Bunga SBI dan Perubahan Nilai Tukar Rupiah terhadap Perubahan Harga Saham dan Perubahan Volume Perdagangan Saham dari Saham-Saham Perusahaan yang Terdaftar di Bursa Efek Jakarta. Buletin Ekonomi, 13(1), 15-38.
[17] Purwadi, J., Gumelar, B., Widiantoro, T., & Ningsih, Z. N. (2024). Canonical Correlation Analysis of Economic Growth and Unemployment Rate. *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, 18(2), 1273-1282.

# LAMPIRAN

## CODE

### Import Data

```
library(readxl)
data<- read_excel("C:/kuliah/semester 6/Multivariate statistic/AOL/coba/VAR BANYAK/datafix4varcoba.xlsx")
view(data)
data$t = seq(1:nrow(data))
head(data)
```

### Preparing the Data for assumption testing

```
# model regresi
model = lm(t~ x1+x2+x3+x4+y1+y2+y3+y4, data=data)
summary(model)
```

### Multicolinearity testing

```
# multikolinearitas
#install.packages("car")
library(car)
vif(model)
```

### Normality testing

```
# normalitas
#install.packages("mvnormtest")
library(mvnormtest)
datanorm = data[2:9]
datanorm
shapiro.test(as.matrix(datanorm))
```

### Tranformation for fullfiling the normality testing

```r
# -----------------
# transformasi 1X
# -----------------
x = data[, c("x1", "x2", "x3", "x4")]
y = data[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted <- x + abs(min(x)) + 1
y_shifted <- y + abs(min(y)) + 1

# Power Transformation untuk x
transform_pt_x <- powerTransform(as.matrix(x_shifted))
x_transformed <- as.data.frame(sapply(1:ncol(x_shifted), function(i) x_shifted[, i]^transform_pt_x$lambda[i]))
colnames(x_transformed) <- colnames(x)

# Power Transformation untuk y
transform_pt_y <- powerTransform(as.matrix(y_shifted))
y_transformed <- as.data.frame(sapply(1:ncol(y_shifted), function(i) y_shifted[, i]^transform_pt_y$lambda[i]))
colnames(y_transformed) <- colnames(y)

# Menggabungkan data yang ditransformasi
transformed <- cbind(x_transformed, y_transformed)
transformed <- as.data.frame(scale(transformed))
transformed

# Uji normalitas menggunakan Shapiro-Wilk
shapiro.test(as.matrix(transformed))
# masih belum normal
```

```r
# -----------------
# transformasi 2X
# -----------------
xt2 = transformed[, c("x1", "x2", "x3", "x4")]
yt2 = transformed[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted2 <- xt2 + abs(min(xt2)) + 1
y_shifted2 <- yt2 + abs(min(yt2)) + 1

# Power Transformation untuk x
transform_pt_x2 <- powerTransform(as.matrix(x_shifted2))
x_transformed2 <- as.data.frame(sapply(1:ncol(x_shifted2), function(i) x_shifted2[, i]^transform_pt_x2$lambda[i]))
colnames(x_transformed2) <- colnames(xt2)

# Power Transformation untuk y
transform_pt_y2 <- powerTransform(as.matrix(y_shifted2))
y_transformed2 <- as.data.frame(sapply(1:ncol(y_shifted2), function(i) y_shifted2[, i]^transform_pt_y2$lambda[i]))
colnames(y_transformed2) <- colnames(yt2)

# Menggabungkan data yang ditransformasi
transformed2 <- cbind(x_transformed2, y_transformed2)
transformed2 <- as.data.frame(scale(transformed2))

shapiro.test(as.matrix(transformed2))
transformed2
```

```r
#-----------
# TRANS 3X
#-----------
xt3 = transformed2[, c("x1", "x2", "x3", "x4")]
yt3 = transformed2[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted3 <- xt3 + abs(min(xt3)) + 1
y_shifted3 <- yt3 + abs(min(yt3)) + 1

# Power Transformation untuk x
transform_pt_x3 <- powerTransform(as.matrix(x_shifted3))
x_transformed3 <- as.data.frame(sapply(1:ncol(x_shifted3), function(i) x_shifted3[, i]^transform_pt_x3$lambda[i]))
colnames(x_transformed3) <- colnames(xt3)

# Power Transformation untuk y
transform_pt_y3 <- powerTransform(as.matrix(y_shifted3))
y_transformed3 <- as.data.frame(sapply(1:ncol(y_shifted3), function(i) y_shifted3[, i]^transform_pt_y3$lambda[i]))
colnames(y_transformed3) <- colnames(yt3)

# Menggabungkan data yang ditransformasi
transformed3 <- cbind(x_transformed3, y_transformed3)
transformed3 <- as.data.frame(scale(transformed3))

shapiro.test(as.matrix(transformed3))
transformed3
```

```r
#-----------
# TRANS 4X
#-----------
xt4 = transformed3[, c("x1", "x2", "x3", "x4")]
yt4 = transformed3[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted4 <- xt4 + abs(min(xt4)) + 1
y_shifted4 <- yt4 + abs(min(yt4)) + 1

# Power Transformation untuk x
transform_pt_x4 <- powerTransform(as.matrix(x_shifted4))
x_transformed4 <- as.data.frame(sapply(1:ncol(x_shifted4), function(i) x_shifted4[, i]^transform_pt_x4$lambda[i]))
colnames(x_transformed4) <- colnames(xt4)

# Power Transformation untuk y
transform_pt_y4 <- powerTransform(as.matrix(y_shifted4))
y_transformed4 <- as.data.frame(sapply(1:ncol(y_shifted4), function(i) y_shifted4[, i]^transform_pt_y4$lambda[i]))
colnames(y_transformed4) <- colnames(yt4)

# Menggabungkan data yang ditransformasi
transformed4 <- cbind(x_transformed4, y_transformed4)
transformed4 <- as.data.frame(scale(transformed4))

shapiro.test(as.matrix(transformed4))
transformed4
```

```r
#-----------
# TRANS 5X
#-----------
xt5 = transformed4[, c("x1", "x2", "x3", "x4")]
yt5 = transformed4[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted5 <- xt5 + abs(min(xt5)) + 1
y_shifted5 <- yt5 + abs(min(yt5)) + 1

# Power Transformation untuk x
transform_pt_x5 <- powerTransform(as.matrix(x_shifted5))
x_transformed5 <- as.data.frame(sapply(1:ncol(x_shifted5), function(i) x_shifted5[, i]^transform_pt_x5$lambda[i]))
colnames(x_transformed5) <- colnames(xt5)

# Power Transformation untuk y
transform_pt_y5 <- powerTransform(as.matrix(y_shifted5))
y_transformed5 <- as.data.frame(sapply(1:ncol(y_shifted5), function(i) y_shifted5[, i]^transform_pt_y5$lambda[i]))
colnames(y_transformed5) <- colnames(yt5)

# Menggabungkan data yang ditransformasi
transformed5 <- cbind(x_transformed5, y_transformed5)
transformed5 <- as.data.frame(scale(transformed5))

shapiro.test(as.matrix(transformed5))
transformed5
```

```r
#-----------
# TRANS 6X
#-----------
xt6 = transformed5[, c("x1", "x2", "x3", "x4")]
yt6 = transformed5[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted6 <- xt6 + abs(min(xt6)) + 1
y_shifted6 <- yt6 + abs(min(yt6)) + 1

# Power Transformation untuk x
transform_pt_x6 <- powerTransform(as.matrix(x_shifted6))
x_transformed6 <- as.data.frame(sapply(1:ncol(x_shifted6), function(i) x_shifted6[, i]^transform_pt_x6$lambda[i]))
colnames(x_transformed6) <- colnames(xt6)

# Power Transformation untuk y
transform_pt_y6 <- powerTransform(as.matrix(y_shifted6))
y_transformed6 <- as.data.frame(sapply(1:ncol(y_shifted6), function(i) y_shifted6[, i]^transform_pt_y6$lambda[i]))
colnames(y_transformed6) <- colnames(yt6)

# Menggabungkan data yang ditransformasi
transformed6 <- cbind(x_transformed6, y_transformed6)
transformed6 <- as.data.frame(scale(transformed6))

shapiro.test(as.matrix(transformed6))
transformed6
```

```r
#-----------
# TRANS 7X
#-----------
xt7 = transformed6[, c("x1", "x2", "x3", "x4")]
yt7 = transformed6[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted7 <- xt7 + abs(min(xt7)) + 1
y_shifted7 <- yt7 + abs(min(yt7)) + 1

# Power Transformation untuk x
transform_pt_x7 <- powerTransform(as.matrix(x_shifted7))
x_transformed7 <- as.data.frame(sapply(1:ncol(x_shifted7), function(i) x_shifted7[, i]^transform_pt_x7$lambda[i]))
colnames(x_transformed7) <- colnames(xt7)

# Power Transformation untuk y
transform_pt_y7 <- powerTransform(as.matrix(y_shifted7))
y_transformed7 <- as.data.frame(sapply(1:ncol(y_shifted7), function(i) y_shifted7[, i]^transform_pt_y7$lambda[i]))
colnames(y_transformed7) <- colnames(yt7)

# Menggabungkan data yang ditransformasi
transformed7 <- cbind(x_transformed7, y_transformed7)
transformed7 <- as.data.frame(scale(transformed7))

shapiro.test(as.matrix(transformed7))
transformed7
```

```r
#-----------
# TRANS 8X
#-----------
xt8 = transformed7[, c("x1", "x2", "x3", "x4")]
yt8 = transformed7[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted8 <- xt8 + abs(min(xt8)) + 1
y_shifted8 <- yt8 + abs(min(yt8)) + 1

# Power Transformation untuk x
transform_pt_x8 <- powerTransform(as.matrix(x_shifted8))
x_transformed8 <- as.data.frame(sapply(1:ncol(x_shifted8), function(i) x_shifted8[, i]^transform_pt_x8$lambda[i]))
colnames(x_transformed8) <- colnames(xt8)

# Power Transformation untuk y
transform_pt_y8 <- powerTransform(as.matrix(y_shifted8))
y_transformed8 <- as.data.frame(sapply(1:ncol(y_shifted8), function(i) y_shifted8[, i]^transform_pt_y8$lambda[i]))
colnames(y_transformed8) <- colnames(yt8)

# Menggabungkan data yang ditransformasi
transformed8 <- cbind(x_transformed8, y_transformed8)
transformed8 <- as.data.frame(scale(transformed8))

shapiro.test(as.matrix(transformed8))
transformed8
```

```r
#-----------
# TRANS 9X
#-----------
xt9 = transformed8[, c("x1", "x2", "x3", "x4")]
yt9 = transformed8[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted9 <- xt9 + abs(min(xt9)) + 1
y_shifted9 <- yt9 + abs(min(yt9)) + 1

# Power Transformation untuk x
transform_pt_x9 <- powerTransform(as.matrix(x_shifted9))
x_transformed9 <- as.data.frame(sapply(1:ncol(x_shifted9), function(i) x_shifted9[, i]^transform_pt_x9$lambda[i]))
colnames(x_transformed9) <- colnames(xt9)

# Power Transformation untuk y
transform_pt_y9 <- powerTransform(as.matrix(y_shifted9))
y_transformed9 <- as.data.frame(sapply(1:ncol(y_shifted9), function(i) y_shifted9[, i]^transform_pt_y9$lambda[i]))
colnames(y_transformed9) <- colnames(yt9)

# Menggabungkan data yang ditransformasi
transformed9 <- cbind(x_transformed9, y_transformed9)
transformed9 <- as.data.frame(scale(transformed9))

shapiro.test(as.matrix(transformed9))
transformed9
```

```r
#-----------
# TRANS 10X
#-----------
xt10 = transformed9[, c("x1", "x2", "x3", "x4")]
yt10 = transformed9[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted10 <- xt10 + abs(min(xt10)) + 1
y_shifted10 <- yt10 + abs(min(yt10)) + 1

# Power Transformation untuk x
transform_pt_x10 <- powerTransform(as.matrix(x_shifted10))
x_transformed10 <- as.data.frame(sapply(1:ncol(x_shifted10), function(i) x_shifted10[, i]^transform_pt_x10$lambda[i]))
colnames(x_transformed10) <- colnames(xt10)

# Power Transformation untuk y
transform_pt_y10 <- powerTransform(as.matrix(y_shifted10))
y_transformed10<- as.data.frame(sapply(1:ncol(y_shifted10), function(i) y_shifted10[, i]^transform_pt_y10$lambda[i]))
colnames(y_transformed10) <- colnames(yt10)

# Menggabungkan data yang ditransformasi
transformed10 <- cbind(x_transformed10, y_transformed10)
transformed10 <- as.data.frame(scale(transformed10))

shapiro.test(as.matrix(transformed10))
transformed10
```

```
#-----------
# TRANS 11X
#-----------
xt11 = transformed10[, c("x1", "x2", "x3", "x4")]
yt11 = transformed10[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted11 <- xt11 + abs(min(xt11)) + 1
y_shifted11 <- yt11 + abs(min(yt11)) + 1

# Power Transformation untuk x
transform_pt_x11 <- powerTransform(as.matrix(x_shifted11))
x_transformed11 <- as.data.frame(sapply(1:ncol(x_shifted11), function(i) x_shifted11[, i]^transform_pt_x11$lambda[i]))
colnames(x_transformed11) <- colnames(xt11)

# Power Transformation untuk y
transform_pt_y11 <- powerTransform(as.matrix(y_shifted11))
y_transformed11<- as.data.frame(sapply(1:ncol(y_shifted11), function(i) y_shifted11[, i]^transform_pt_y11$lambda[i]))
colnames(y_transformed11) <- colnames(yt11)

# Menggabungkan data yang ditransformasi
transformed11 <- cbind(x_transformed11, y_transformed11)
transformed11 <- as.data.frame(scale(transformed11))

shapiro.test(as.matrix(transformed11))
transformed11
```

```
#-----------
# TRANS 12X
#-----------
xt12 = transformed11[, c("x1", "x2", "x3", "x4")]
yt12 = transformed11[, c("y1", "y2", "y3", "y4")]

# Shift data to make all values positive
x_shifted12 <- xt12 + abs(min(xt12)) + 1
y_shifted12 <- yt12 + abs(min(yt12)) + 1

# Power Transformation untuk x
transform_pt_x12 <- powerTransform(as.matrix(x_shifted12))
x_transformed12 <- as.data.frame(sapply(1:ncol(x_shifted12), function(i) x_shifted12[, i]^transform_pt_x12$lambda[i]))
colnames(x_transformed12) <- colnames(xt12)

# Power Transformation untuk y
transform_pt_y12 <- powerTransform(as.matrix(y_shifted12))
y_transformed12<- as.data.frame(sapply(1:ncol(y_shifted12), function(i) y_shifted12[, i]^transform_pt_y12$lambda[i]))
colnames(y_transformed12) <- colnames(yt12)

# Menggabungkan data yang ditransformasi
transformed12 <- cbind(x_transformed12, y_transformed12)
transformed12 <- as.data.frame(scale(transformed12))

shapiro.test(as.matrix(transformed12))
transformed12
```

**Linearity testing**

```
# UJI LINEAR
cor.test(transformed12$x1, transformed12$x2)
cor.test(transformed12$x1, transformed12$x3)
cor.test(transformed12$x1, transformed12$x4)
cor.test(transformed12$x2, transformed12$x3)
cor.test(transformed12$x2, transformed12$x4)
cor.test(transformed12$x3, transformed12$x4)

cor.test(transformed12$y1, transformed12$y2)
cor.test(transformed12$y1, transformed12$y3)
cor.test(transformed12$y1, transformed12$y4)
cor.test(transformed12$y2, transformed12$y3)
cor.test(transformed12$y2, transformed12$y4)
cor.test(transformed12$y3, transformed12$y4)
```

**Matrix A**

```
rho22 = S[1:4 , 1:4]
rho11 = S[5:8, 5:8]
rho21 = S[1:4, 5:8]
rho12 = S[5:8, 1:4]
rho11_inverssqrt = solve(sqrtm(rho11))
rho11_invers = solve(rho11)
rho11_inverssqrt
rho22_invers = solve(rho22)
rho22_inverssqrt = solve(sqrtm(rho22))
rho22_invers
rho11

A = rho11_inverssqrt%*%rho12%*%rho22_invers%*%rho21%*%rho11_inverssqrt
A
```

**Correlation canonical**

```
r2 = (values<-eigen$values)
r2
r = sqrt(r2)
r
```

**Matrix B**

```
B = rho22_inverssqrt%*%rho21%*%rho11_invers%*%rho12%*%rho22_inverssqrt
B
```

**Simultaneous testing**

```
det(S)
det(rho11)
det(rho22)
rho11

(det(S)/(det(rho11)*det(rho22)))
```

**Partial testing**

```r
# uji parsial
library(CCP)
n <- 34
p <- 4
q <- 4
wilks_result <- p.asym(r, n, p, q, tstat = "Wilks")

# Print the results
wilks_result

## F-test
F_1 = wilks_result$approx[1]
F_2 = wilks_result$approx[2]
F_3 = wilks_result$approx[3]
F_4 = wilks_result$approx[4]

## F-Table
f_table_1 = qf(p=0.05, wilks_result$df1[1], wilks_result$df2[1], lower.tail = FALSE)
f_table_2 = qf(p=0.05, wilks_result$df1[2], wilks_result$df2[2], lower.tail = FALSE)
f_table_3 = qf(p=0.05, wilks_result$df1[3], wilks_result$df2[3], lower.tail = FALSE)
f_table_4 = qf(p=0.05, wilks_result$df1[4], wilks_result$df2[4], lower .tail = FALSE)
```

**Canonical function**

```r
eigen = eigen(A)
e = eigen$vectors
(e1 = eigen$vectors[,1])
(u1 = e1 %*% rho11_inverssqrt)
```

```r
eigenB = eigen(B)
eb = eigenB$vectors
eb
(eb1 = eigenB$vectors[,1])
(v1 = eb1 %*% rho22_inverssqrt)
```

**DATA**

| Provinsi | x1 | x2 | x3 | x4 | y1 | y2 | y3 | y4 |
|---|---|---|---|---|---|---|---|---|
| Aceh | -1.27093716 | 0.62295955 | -0.28492812 | 1.38569735 | 0.02283457 | 1.58097042 | 0.01665417 | 0.18140868 |
| Sumatera Utara | 0.44327339 | 0.89339653 | 0.49859370 | 0.55465498 | 0.26195804 | 0.74759066 | -1.65438840 | 1.10644432 |
| Sumatera Barat | 2.45078901 | -0.99966236 | -0.27592886 | -0.44685763 | -0.22947610 | -0.29413403 | -1.35113444 | -0.47675017 |
| Riau | 0.39741461 | 0.66622946 | -0.01105407 | -0.08460839 | 0.56055883 | -0.91916885 | -0.83900016 | -0.48749842 |
| Jambi | 0.43861585 | -0.92394000 | -0.27880041 | -0.87303320 | -0.11301723 | -0.91916885 | 0.38030726 | -0.64780744 |
| Sumatera Selatan | 0.42022458 | 1.52081033 | -0.07840110 | -0.16984350 | 0.52101409 | -0.91916885 | 0.10221960 | 0.28248798 |
| Bengkulu | 1.93487780 | 0.44987988 | -0.51529921 | -0.19115228 | -0.59141594 | -0.91916885 | 0.53759666 | -0.61007422 |
| Lampung | -0.95446384 | 0.55805467 | -0.04772507 | -0.19115228 | -0.36362470 | 0.12255585 | 1.08496377 | -0.21673397 |
| Kepulauan Bangka Belitung | 0.15331218 | 0.08208558 | -0.50363423 | -0.83041565 | -0.64944301 | -0.91916885 | 0.65461997 | 0.28934856 |
| Kepulauan Riau | -0.36749541 | -0.51287579 | -0.51852923 | -0.89434198 | -0.79345672 | -0.91916885 | -1.73617889 | -0.57371312 |
| DKI Jakarta | -0.91003816 | 0.50396727 | 2.58720287 | -0.74518053 | -0.91924106 | -0.29413403 | -3.40470484 | -0.54032493 |
| Jawa Barat | 0.72499435 | -2.01650541 | 1.82849904 | 2.98385577 | -0.30325361 | 3.45607488 | -0.07268621 | 3.97508388 |
| Jawa Tengah | -0.30384248 | 1.21792091 | 2.60535630 | 2.89862065 | -0.34709853 | 1.58097042 | 0.98052361 | 0.86609513 |
| DI Yogyakarta | 0.31393731 | -0.58859814 | -0.19312081 | -0.80910687 | -0.87612126 | -0.29413403 | 0.29096688 | 1.69233834 |
| Jawa Timur | 0.19272206 | -3.26051554 | 3.23154850 | 2.30197484 | -0.10913864 | 1.16428054 | 0.96668214 | 2.20185117 |
| Banten | -0.02522654 | 1.23955587 | -0.22287974 | 0.14978818 | -0.77119699 | 0.74759066 | 0.22553449 | 0.04373960 |
| Bali | -0.07633997 | -0.01527174 | 0.05550350 | -0.72387175 | -0.83480585 | 0.12255585 | 0.04685374 | -0.63751656 |
| Nusa Tenggara Barat | -0.29667705 | 0.63377702 | -0.35252341 | -0.40424007 | -0.59276501 | -0.71082391 | 0.95032404 | 0.01744068 |
| Nusa Tenggara Timur | -0.85426720 | 1.45590545 | -0.53181499 | -0.29769617 | -0.13257882 | 0.53924573 | 0.80561780 | 0.21822716 |
| Kalimantan Barat | 0.16991210 | 0.70949938 | -0.20042921 | -0.10591717 | 1.52411848 | -0.71082391 | 0.95661562 | -0.54649946 |
| Kalimantan Tengah | 0.43085330 | -0.94557496 | -0.43849483 | 0.04324429 | 1.62772741 | -0.50247897 | -0.28534147 | -0.62036510 |
| Kalimantan Selatan | -0.19134517 | 0.29843517 | -0.22654133 | 0.19240574 | -0.30817773 | -0.71082391 | -0.25010865 | -0.43375717 |
| Kalimantan Timur | 0.49605875 | -0.59941562 | -0.15984590 | -0.36162251 | 1.20462390 | -0.50247897 | -1.14980401 | -0.61213240 |
| Kalimantan Utara | -0.99482912 | -0.71840789 | -0.66674066 | -0.68125419 | 0.22659542 | -0.91916885 | -1.11457119 | -0.64209029 |
| Sulawesi Utara | 0.75413378 | -0.02608921 | -0.53809930 | -0.36162251 | -0.68686296 | -0.08578909 | -0.51183821 | -0.65123773 |
| Sulawesi Tengah | -2.41107705 | -0.88067008 | -0.47750299 | 0.91690422 | 0.08907414 | 1.58097042 | 0.64581177 | -0.58148845 |
| Sulawesi Selatan | 2.09299503 | 0.40660996 | 0.04228421 | 0.19240574 | -0.17871717 | 0.12255585 | 0.53004677 | 0.82516031 |
| Sulawesi Tenggara | -0.24233918 | 0.57968963 | -0.54283687 | -0.59601908 | -0.31945937 | -0.08578909 | 0.82449253 | -0.56776727 |
| Gorontalo | -0.97392994 | 0.76358678 | -0.61755405 | -0.27638740 | -0.72851564 | -0.91916885 | 0.20414314 | -0.19386535 |
| Sulawesi Barat | -0.60013316 | -0.62105058 | -0.63057659 | -0.38293129 | -0.65197252 | -0.50247897 | 1.17304583 | -0.62722568 |
| Maluku | 1.03621301 | 0.57968963 | -0.63971277 | -0.55340152 | -0.14979638 | 0.53924573 | -0.01732034 | -0.56548041 |
| Maluku Utara | 0.01860203 | -0.75086033 | -0.64262939 | -0.25507862 | -0.40116270 | -0.71082391 | 0.78296812 | -0.50990967 |
| Papua Barat | -1.61141467 | -0.26407376 | -0.63698400 | -0.55340152 | 0.72285250 | -0.29413403 | -0.17460974 | -0.62265196 |
| Papua | -0.38457303 | -0.05854165 | -0.61640100 | -0.83041565 | 4.28994057 | 0.74759066 | 0.40169862 | -0.33473605 |