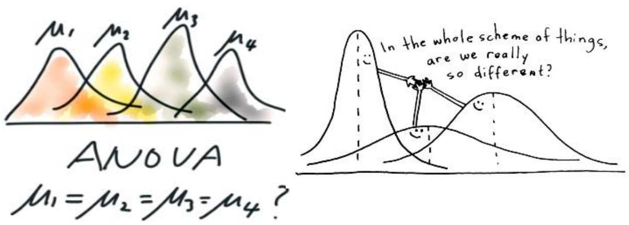
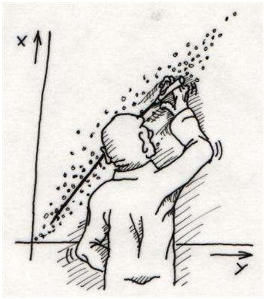
Advanced Statistics

Individual Assignment



ANOVA (Analysis of Variance)



Linear Regression

PCA
(Principal Component Analysis)

By:

Nikhil Panchal

Submission:

11th January, 2021

GREAT LAKES
INSTITUTE OF MANAGEMENT

greatlearning
Power Ahead

Introduction:

Nikhil Panchal

- Overall **13+ years of cross functional experience** in FMCG and Alcobev industry.
- Working with **Diageo PLC** since 2017, looking after **Data Analytics CoE for Global Audit & Risk department**.
- Before to that worked with **Coca-Cola for 10 years** and completed stint in operations, supply chain (direct & indirect), Master data maintenance (SAP cross functional role) and Data analytics (Internal Audit).

Table of Contents:

Problem1:

A research laboratory was developing a new compound for the relief of severe cases of hay fever. In an experiment with 36 volunteers, the amounts of the two active ingredients (A & B) in the compound was varied at three levels each. Randomization was used in assigning four volunteers to each of the nine treatments.

[Assume all of the ANOVA assumptions are satisfied]

Remarks: All the questions of problem1 which explained here, are also performed in python as well. Please refer python notebook "AS_Individual_Assignment_Problem1".

1.1 State the Null and Alternate Hypothesis for conducting one-way ANOVA for both the variables 'A' and 'B' individually

ANOVA Background: Analysis of Variance (ANOVA) is a **parametric statistical method** used to compare datasets. This technique was invented by **R.A. Fisher**, and is thus often referred to as **Fisher's ANOVA**, as well. This technique is the part of the domain called "Experimental Designs". It helps in establishing in a precise fashion the Cause Effect relation amongst variables. From the statistical inference point of view, ANOVA is an extension of independent t- test for testing the equality of two population means. When we have to **compare more than two population means, we use ANOVA**.

ANOVA Key Assumptions: There are mainly 3 assumptions which needs to be considered for ANOVA test, are stated below:

Independence of case: Independence of case assumption means that the case of the dependent variable should **be independent or the sample should be selected randomly**.

Normality: Distribution of each group should be normal. The **Kolmogorov Smirnov or the Shapiro Wilk test may be used to confirm normality of the group**.

Homogeneity: Homogeneity means **variance between the groups should be the same. Levene's test is used to test the homogeneity between groups.**

In this problem, we have been already given that ***all the ANOVA assumptions are satisfied***. Hence, we will not perform these assumptions while solving the same.

Dataset Background: In this dataset, the research laboratory is developing new compound for reducing pain in *hay fever*. In this compound, they are having two active ingredients (i.e., A & B) with three levels each. Further, there are 36 volunteers used randomly *(where in four volunteers assigned to total nine treatments)*.

Let's pan out the NULL (**H_0**) and ALTERNATE (**H_a**) hypothesis for ingredient A & ingredient B with respect to Relief time.

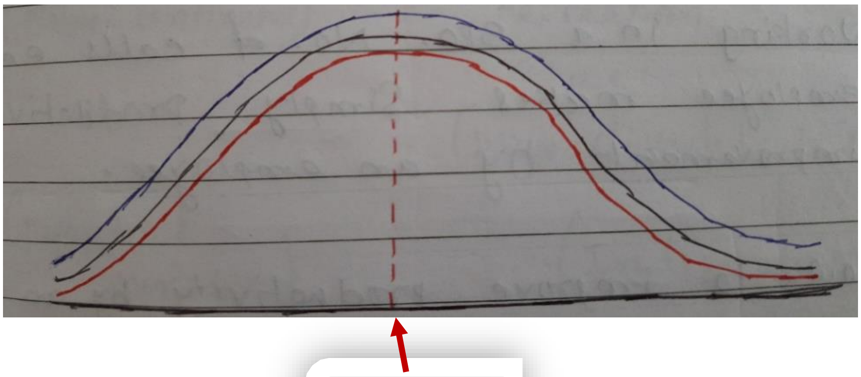
Hypothesis for ingredient "A":

Assuming that the 3 levels represents populations, whose values are randomly and independently selected, follow a normal distribution, and have equal variances.

Null Hypothesis (H_0):

$\mu_{A1} = \mu_{A2} = \mu_{A3}$

(Average relief time for all the three levels in ingredient A are equal)

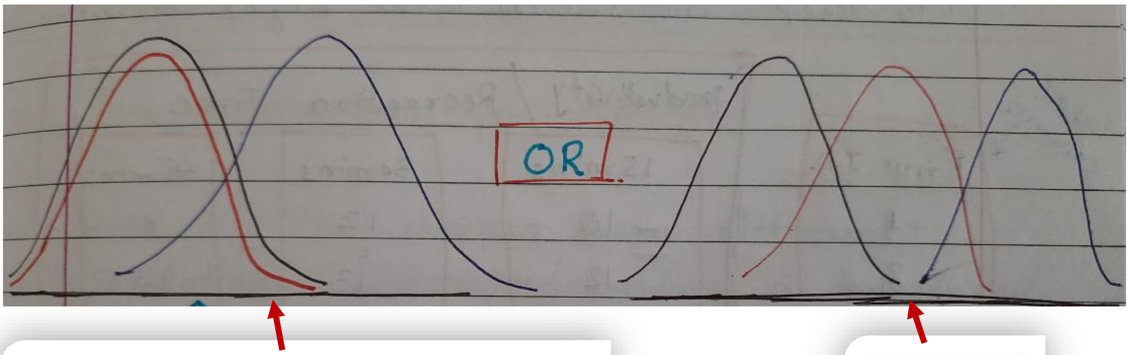


$\mu_{A1} = \mu_{A2} = \mu_{A3}$

Alternate Hypothesis (H_a):

$\mu_{A1} \neq \mu_{A2} \neq \mu_{A3}$ **OR** $\mu_{A1} = \mu_{A2} \neq \mu_{A3}$ **OR** $\mu_{A1} \neq \mu_{A2} = \mu_{A3}$ **OR** $\mu_{A1} = \mu_{A3} \neq \mu_{A2}$

(Not all of the average relief time for all the three levels in ingredient A are equal. In other words, at least one pair has average relief time in ingredient A are unequal)



$\mu_{A1} = \mu_{A2} \neq \mu_{A3}$ **OR** $\mu_{A1} \neq \mu_{A2} = \mu_{A3}$ **OR** $\mu_{A1} = \mu_{A3} \neq \mu_{A2}$

$\mu_{A1} \neq \mu_{A2} \neq \mu_{A3}$

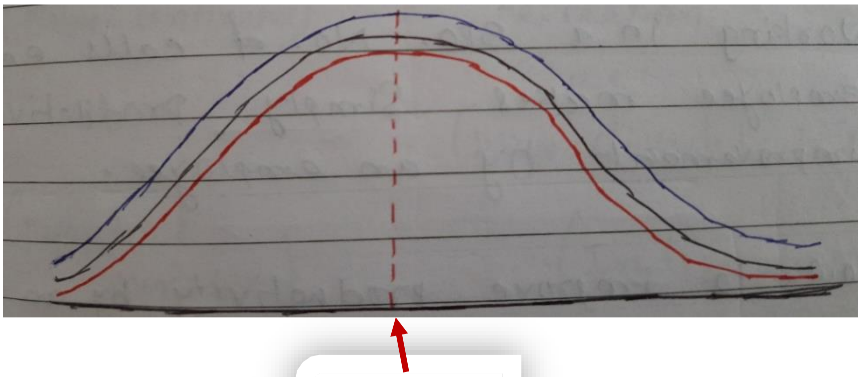
Hypothesis for ingredient "**B**":

Assuming that the 3 levels represents populations, whose values are randomly and independently selected, follow a normal distribution, and have equal variances.

Null Hypothesis (H_0):

$\mu_{B1} = \mu_{B2} = \mu_{B3}$

(Average relief time for all the three levels in ingredient B are equal)

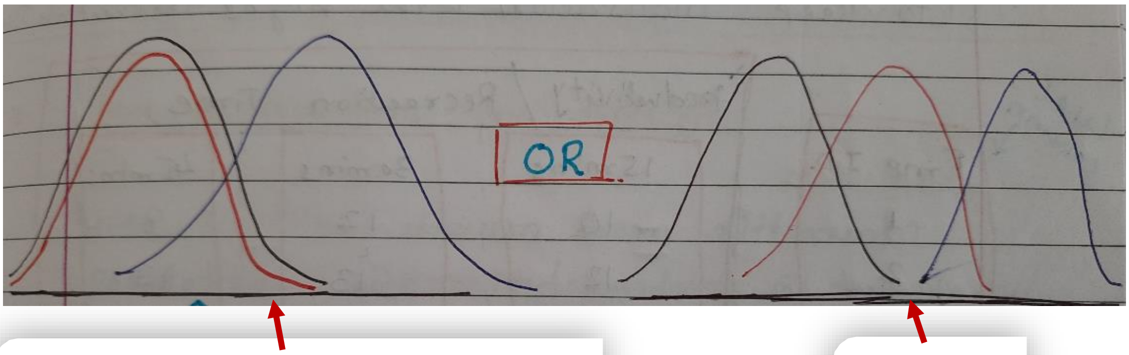


$\mu_{B1} = \mu_{B2} = \mu_{B3}$

Alternate Hypothesis (H_a):

$\mu_{B1} \neq \mu_{B2} \neq \mu_{B3}$ **OR** $\mu_{B1} = \mu_{B2} \neq \mu_{B3}$ **OR** $\mu_{B1} \neq \mu_{B2} = \mu_{B3}$ **OR** $\mu_{B1} = \mu_{B3} \neq \mu_{B2}$

(Not all of the average relief time for all the three levels in ingredient B are equal, which means at least one pair has average relief time in ingredient B are unequal)



$\mu_{B1} = \mu_{B2} \neq \mu_{B3}$ **OR** $\mu_{B1} \neq \mu_{B2} = \mu_{B3}$ **OR** $\mu_{B1} = \mu_{B3} \neq \mu_{B2}$

$\mu_{B1} \neq \mu_{B2} \neq \mu_{B3}$

1.2 Perform one-way ANOVA for variable ‘A’ with respect to the variable ‘Relief’. State whether the Null Hypothesis is accepted or rejected based on the ANOVA results.

Background:

In one-way ANOVA, we partition the **total variations (SST)** into variation that is due to *“differences between the groups”* and variation that is due to *“differences within the groups”*. The **between group variation (SSB)** measures the differences from group to group. The **within group (SSW)** measures the differences within the groups.

Total Variations (SST) = **Between Group Variations (SSB)** + **Within Group Variations (SSW)**

Total Variations (SST): The total variation is represented by the **sum of square total (SST)**. Because the population mean of the groups are assumed to be equal under the null hypothesis, we compute the total variation among all the values by summing the squared differences between each individual value and the **grand mean**($\bar{\bar{X}}$). The grand mean is the mean of all the values in all the groups combined. Below equation shows the computation of the total variation.

no. of groups

no. of values in group j

$$SST = \sum_{j=1}^{C} \sum_{i=1}^{n_j} \left(x_{ij} - \bar{\bar{x}}\right)^2$$

ith value in group j

grand mean

Between Group Variations (SSB): Between group variation, usually called the **sum of squares between the groups (SSB)**, by summing the squared differences between the sample mean of each group, $\bar{X}_j$, and the grand mean $\bar{\bar{X}}$, weighted by sample size n_j in each group. Below equation shows the computation of the between the group variation.

no. of groups

sample mean of group j

$$SSB = \sum_{j=1}^{C} n_j \left(\bar{x}_j - \bar{\bar{x}}\right)^2$$

no. of values in group j

grand mean

Within Group Variations (SSW): Within group variation, usually called the **sum of squares within the groups (SSW)**, measures the difference between each value and the mean of its own group and sums the squares of these differences over all groups. Below equation shows the computation of the within the group variation.

$$SSW = \sum_{j=1}^{\overset{\text{no. of groups}}{c}} \sum_{i=1}^{\overset{\text{no. of values in group j}}{n_j}} \left(\underset{i^{th}\text{ value in group j}}{x_{ij}} - \underset{\text{sample mean of group j}}{\bar{\bar{x}}_j}\right)^2$$

Mean Squares in one-way ANOVA: Because we are comparing c groups, there are c – 1 degrees of freedom associated with the sum of squares between groups. The same way, each of the c groups contributes n_j – 1 degrees of freedom, there are n – c degrees of freedom associated with the sum of squares within the groups. Further, there are n – 1 degrees of freedom associated with the sum of squares total.

If we divide each of the sums of squares by its respective degrees of freedom, we have three variances. In ANOVA, these three variances are called the **mean squares** and the three mean squares are defined as MSB (mean square between the groups), MSW (mean squares within the group) and MST (mean square total).

$$MSB = \frac{SSB}{c - 1}$$

$$MSW = \frac{SSW}{n - c}$$

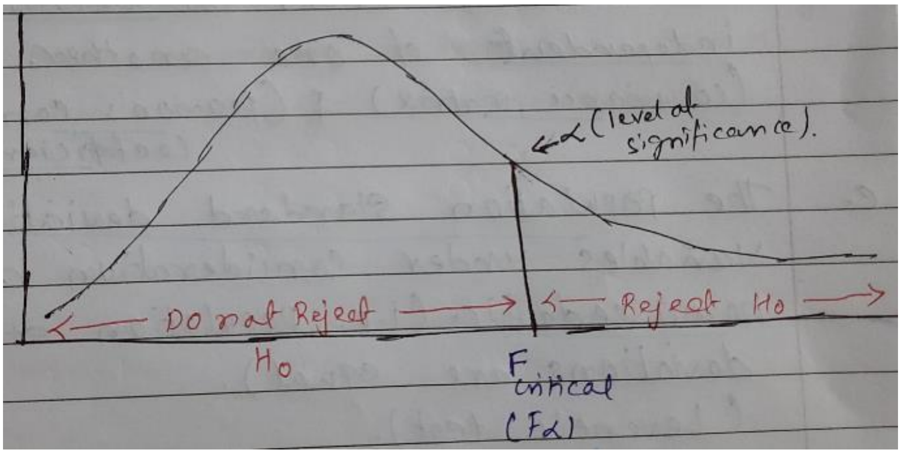
$$MST = \frac{SST}{n - 1}$$

F test in one-way ANOVA: To determine if there is a significant difference between the group means, we use the F test for differences between more than two means. If the null hypothesis is true and there is no differences between the c group means, MSA, MSW and MST, will provide estimates of the overall variance in the population.

$$F_{STAT} = \frac{MSB}{MSW}$$

For a given level of significance (α), we reject the null hypothesis if the F_{STAT} test statistic value is greater than the upper-tail value (F_{α}), from the F distribution with c – 1 degrees of freedom in numerator and n – c in the denominator. So, the decision rule is:

Reject H_0 (null hypothesis) if $F_{STAT} > F_{\alpha}$, otherwise, do not reject H_0.



ANOVA Summary Table: This summary table is typically used to summarise the result of a one-way ANOVA.

Variation	Degree of Freedom	Sum of Squares	Mean Square (Variance)	F_{STAT}
Between Groups	c – 1	SSB	MSB	$F_{STAT} = \frac{MSB}{MSW}$
Within Groups	n – c	SSW	MSW	
Total	n – 1	SST	MST	

Based on above brief explanation, let's perform the same steps for the given hay fever dataset, specific to ingredient 'A' with 3 levels and variable 'Relief' to check whether to accept or reject the null hypothesis.

As stated in problem1.1, mentioned below is the null and alternate hypothesis.

Null Hypothesis (H_0):

$\mu_{A1} = \mu_{A2} = \mu_{A3}$

(Average relief time for all the three levels in ingredient A are equal)

Alternate Hypothesis (H_a):

$\mu_{A1} \neq \mu_{A2} \neq \mu_{A3}$ **OR** $\mu_{A1} = \mu_{A2} \neq \mu_{A3}$ **OR** $\mu_{A1} \neq \mu_{A2} = \mu_{A3}$ **OR** $\mu_{A1} = \mu_{A3} \neq \mu_{A2}$

(Not all of the average relief time for all the three levels in ingredient A are equal. In other words, at least one pair has average relief time in ingredient A are unequal)

Data table summary: based on data table summary let's calculate variations and mean squares, F value and draw out the ANOVA summary table.

	Ingredient "A"		
	A1	A2	A3
Relief	2.4	5.8	6.1
	2.7	5.2	5.7
	2.3	5.5	5.9
	2.5	5.3	6.2
	4.6	8.9	9.9
	4.2	9.1	10.5
	4.9	8.7	10.6
	4.7	9	10.1
	4.8	9.1	13.5
	4.5	9.3	13
	4.4	8.7	13.3
	4.6	9.4	13.2
Sample Mean	**3.883333**	**7.833333**	**9.833333**
Grand Mean	**7.183333333**		
Std dev.	**1.059016**	**1.777298**	**3.127977**
Variance	**1.121515**	**3.158788**	**9.784242**

Between Group Variation:

no. of groups

sample mean of group j

$$SSB = \sum_{j=1}^{c} n_j(\bar{x}_j - \bar{\bar{x}})^2$$

no. of values in group j

grand mean

$$SSB = 12 * (3.88 - 7.18)^2 + 12 * (7.83 - 7.18)^2 + 12 * (9.83 - 7.18)^2$$

$$SSB = 220.02$$

Within Group Variation:

no. of groups

no. of values in group j

$$SSW = \sum_{j=1}^{C} \sum_{i=1}^{n_j} (x_{ij} - \bar{\bar{x}}_j)^2$$

i^th value in group j

sample mean of group j

$$SSW = (2.4 - 3.88)^2 + (2.7 - 3.88)^2 + ... + (13.2 - 9.83)^2$$

$$SSW = 154.71$$

Mean Square between the Group:

Here, c (no. of groups) = 3 (i.e., A1, A2 & A3)

$$MSB = \frac{SSB}{c-1}$$

$$= \frac{220.02}{3-1}$$

$$\boxed{MSB = 110.01}$$

Mean Square within the Group:

Here, c (no. of groups) = 3 (i.e., A1, A2 & A3)

& n = 36

$$MSW = \frac{SSW}{n-c}$$

$$= \frac{154.71}{36-3}$$

$$\boxed{MSW = 4.688}$$

F_{STAT} & Fα Value:

Here, c (no. of groups) = 3 (i.e., A1, A2 & A3)

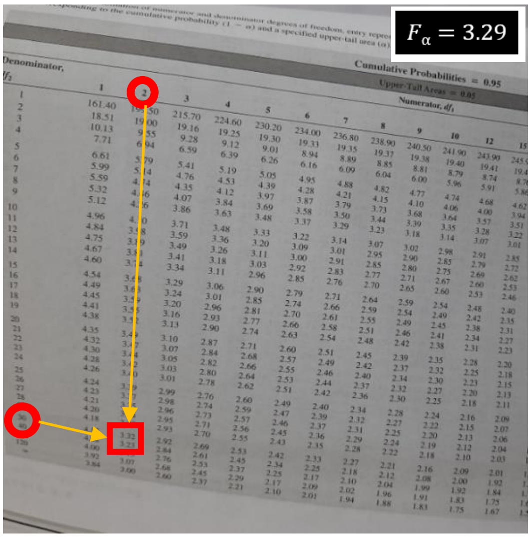
& n = 36

$$F_{STAT} = \frac{MSB}{MSW}$$

$$= \frac{110.01}{4.688}$$

$$\boxed{F_{STAT} = 23.465}$$

For **Fα (critical) value**, we need to follow f distribution table, where numerator df1 = (c - 1) = **2** and denominator **df2** = (n – c) = (36 – 3) = **33** and assuming the *level of significance as 95%*.



$F_\alpha = 3.29$

Cumulative Probabilities = 0.95

Upper-Tail Areas = 0.05

Numerator, df_1

Denominator, df_2

Because **F_{STAT} = 23.465** is greater than **Fα = 3.29**,

$$\boxed{F_{STAT} = 23.465} > \boxed{F_\alpha = 3.29}$$

We Reject the null hypothesis (H_0). Hence, we conclude that "there is a significant difference in the mean relief time for the three levels of ingredient A".

1.3 Perform one-way ANOVA for variable ‘B’ with respect to the variable ‘Relief’. State whether the Null Hypothesis is accepted or rejected based on the ANOVA results.

Based on brief explanation in 1.2, let’s perform the same steps for the given hay fever dataset, specific to ingredient ‘B’ with 3 levels and variable ‘Relief’ to check whether to accept or reject the null hypothesis.

As stated in problem1.1, mentioned below is the null and alternate hypothesis.

Null Hypothesis (H_0):

$\mu_{B1} = \mu_{B2} = \mu_{B3}$

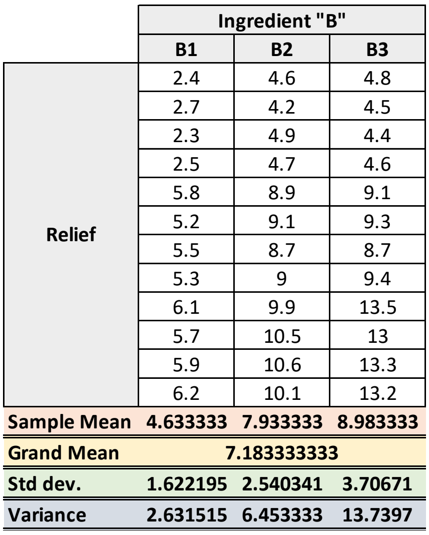
(Average relief time for all the three levels in ingredient B are equal)

Alternate Hypothesis (H_a):

$\mu_{B1} \neq \mu_{B2} \neq \mu_{B3}$ **OR** $\mu_{B1} = \mu_{B2} \neq \mu_{B3}$ **OR** $\mu_{B1} \neq \mu_{B2} = \mu_{B3}$ **OR** $\mu_{B1} = \mu_{B3} \neq \mu_{B2}$

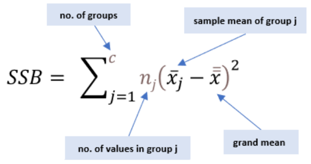
(Not all of the average relief time for all the three levels in ingredient B are equal, which means at least one pair has average relief time in ingredient B are unequal)

Data table summary: based on data table summary let’s calculate variations and mean squares, F value and draw out the ANOVA summary table.



	Ingredient "B"		
	B1	B2	B3
Relief	2.4	4.6	4.8
	2.7	4.2	4.5
	2.3	4.9	4.4
	2.5	4.7	4.6
	5.8	8.9	9.1
	5.2	9.1	9.3
	5.5	8.7	8.7
	5.3	9	9.4
	6.1	9.9	13.5
	5.7	10.5	13
	5.9	10.6	13.3
	6.2	10.1	13.2
Sample Mean	4.633333	7.933333	8.983333
Grand Mean	7.183333333		
Std dev.	1.622195	2.540341	3.70671
Variance	2.631515	6.453333	13.7397

Between Group Variation:



$$SSB = \sum_{j=1}^{c} n_j(\bar{x}_j - \bar{\bar{x}})^2$$

$$SSB = 12 * (4.63 - 7.18)^2 + 12 * (7.93 - 7.18)^2 + 12 * (8.98 - 7.18)^2$$

$$\boxed{SSB = 123.66}$$

Within Group Variation:

$$SSW = \overset{\text{no. of groups}}{\sum_{j=1}^{c}} \overset{\text{no. of values in group j}}{\sum_{i=1}^{n_j}} (\underset{\text{i}^{\text{th}}\text{ value in group j}}{x_{ij}} - \underset{\text{sample mean of group j}}{\bar{\bar{x}}_j})^2$$

$$SSW = (2.4 - 4.63)^2 + (2.7 - 4.63)^2 + ... + (13.2 - 8.98)^2$$

$$\boxed{SSW = 251.07}$$

Mean Square between the Group:

Here, c (no. of groups) = 3 (i.e., B1, B2 & B3)

$$MSB = \frac{SSB}{c - 1}$$

$$= \frac{123.66}{3 - 1}$$

$$\boxed{MSB = 61.83}$$

Mean Square within the Group:

Here, c (no. of groups) = 3 (i.e., B1, B2 & B3)

& n = 36

$$MSW = \frac{SSW}{n - c}$$

$$= \frac{251.07}{36 - 3}$$

$$\boxed{MSW = 7.608}$$

F_{STAT} & Fα Value:

Here, c (no. of groups) = 3 (i.e., B1, B2 & B3)

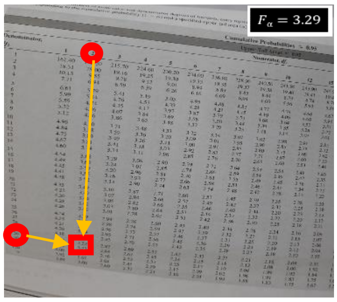
& n = 36

$$F_{STAT} = \frac{MSB}{MSW}$$

$$= \frac{61.83}{7.608}$$

$$\boxed{F_{STAT} = 8.127}$$

For **Fα (critical) value**, we need to follow f distribution table, where numerator df1 = (c - 1) = **2** and denominator **df2** = (n – c) (36 – 3) = **33** and assuming the *level of significance as 95%*.



=

Because F_{STAT} = **8.127** is greater than **Fα = 3.29**,

$$\boxed{F_{STAT} = 8.127} > \boxed{F_\alpha = 3.29}$$

We Reject the null hypothesis (H_0). Hence, we conclude that "there is a significant difference in the mean relief time for the three levels of ingredient B".

1.4 Analyse the effects of one variable on another with the help of an interaction plot. What is the interaction between the two treatments? [hint: use the 'point plot' function from the 'seaborn' function].

Background:

An interaction plot shows how **the relationship between one categorical factor (here ingredient "A") and a continuous response variable (here "Relief time") depends on the value of the second categorical factor (here ingredient "B")**. This plot displays means for *the levels of one factor on the x-axis (ingredient "A") and a separate line for each level of another factor (ingredient "B").* Evaluate the lines to understand how the interactions affect the relationship between the factors (ingredient "A" and "B") and the response variable ("Relief time").

In other words, Interaction effects occur when the effect of one variable depends on the value of another variable. Interaction effects are common in regression analysis, ANOVA, etc. Interaction effects indicate that a third variable influences the relationship between an independent and dependent variable. This type of effect makes the model more complex, but if the real world behaves this way, it is critical to incorporate it in your model.

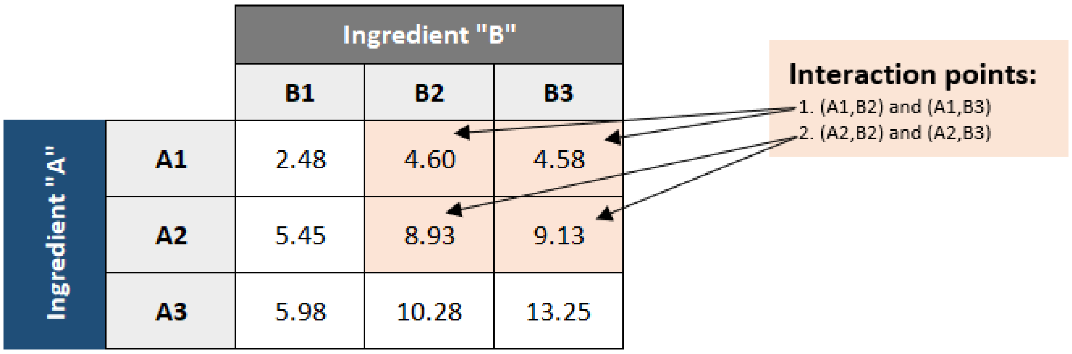
If interaction shows, **Parallel lines then No interaction occurs**.

If interaction shows, **Nonparallel lines then an interaction occurs**. *The more nonparallel the lines are, the greater the strength of the interaction.*

We can use this plot to display the effects, be sure to perform the appropriate ANOVA test and evaluate the statistical significance of the effects. If the interaction effects are significant, we can't interpret the main effects without considering the interaction effects.

Based on the given dataset, mentioned below is the **summary table view** and **an interaction plot view** for categorical variable (between "A" and "B") with continuous variable (relief time).

Summary table view:

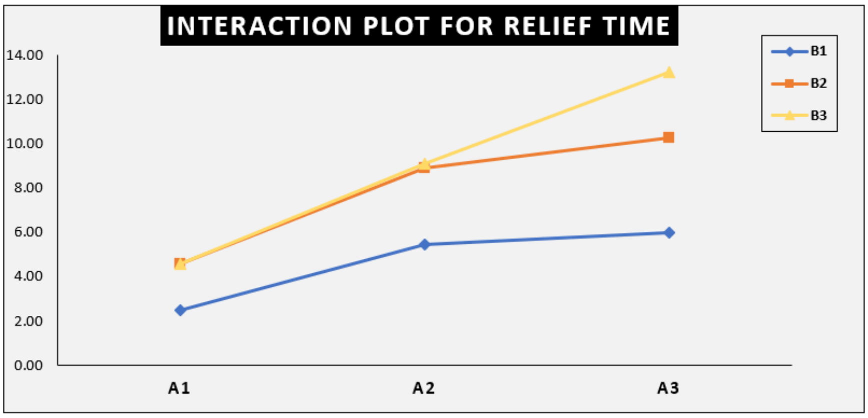


		Ingredient "B"		
		B1	B2	B3
Ingredient "A"	A1	2.48	4.60	4.58
	A2	5.45	8.93	9.13
	A3	5.98	10.28	13.25

Interaction points:

1. (A1,B2) and (A1,B3)
2. (A2,B2) and (A2,B3)

Interaction plot view:



From the above interaction plot, since the lines are interacting each other at points: ***i) (A1,B2 & A1,B3) and ii) (A2,B2 & A2,B3)*** that means there is an interaction effect of these 2 variables. In other words, the **lines are not parallel**. This interaction effect indicates that the relationship between ingredient "A" and "relief time" depends on the value of ingredient "B".

Conclusion: The results indicate that the interaction effects between ingredient "A" and ingredient "B" are significant (very unlikely to be absent).

1.5 Perform a two-way ANOVA based on the different ingredients (variable 'A' & 'B') with the variable 'Relief' and state your results.

Background:

Two-way ANOVA is used to *determine whether or not there is a statistically significant difference between the means of three or more independent groups that have been split on two factors*.

The purpose of a two-way ANOVA is to determine how two factors (here ingredient "A" and "B") impact a response variable (here "Relief"), and to determine whether or not there is an interaction between the two factors on the response variable.

An **important advantage** of the two-way ANOVA is that it is more efficient compared to the one-way. There are two assignable sources of variation – A and B in our dataset – and this helps to reduce error variation thereby making this design more efficient.

For given dataset, we are going to calculate the two-way ANOVA table "by hand".

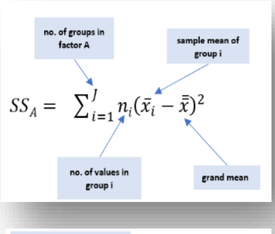
Mentioned below table & formulas are used to calculate

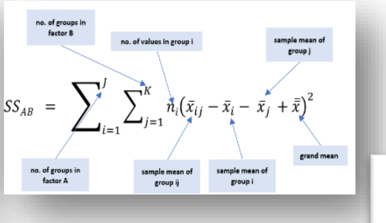
(Without Interaction): $F_{STAT(A)}$, $F_{STAT(B)}$ & (With interaction)$F_{STAT(AB)}$.

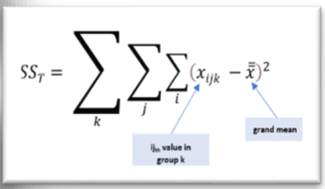
Summary Table:

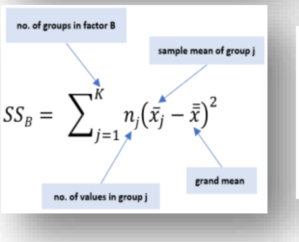
Variation	Degree of Freedom	Sum of Squares	Mean Square (Variance)	F_{STAT}
Factor A	$df_A = J - 1$	SS_A	$MS_A = SS_A / df_A$	$F_A = MS_A / MS_W$
Factor B	$df_B = K - 1$	SS_B	$MS_B = SS_B / df_B$	$F_B = MS_B / MS_W$
Interaction AB	$df_{AB} = (J - 1) * (K - 1)$	SS_{AB}	$MS_{AB} = SS_{AB} / df_{AB}$	$F_{AB} = MS_{AB} / MS_W$
Residual (error / within)	$df_W = N - (J * K)$	SS_W	$MS_W = SS_W / df_W$	
Total	$df_T = N - 1$	SS_T	$MS_T = SS_T / df_T$	

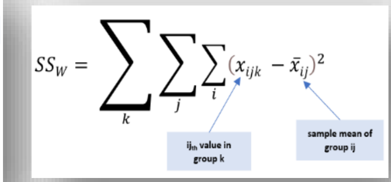
Formulas for Sum of Squares:


$$SS_A = \sum_{i=1}^{J} n_i(\bar{x}_i - \bar{\bar{x}})^2$$


$$SS_{AB} = \sum_{i=1}^{J} \sum_{j=1}^{K} n_i(\bar{x}_{ij} - \bar{x}_i - \bar{x}_j + \bar{\bar{x}})^2$$


$$SS_T = \sum_k \sum_j \sum_i (x_{ijk} - \bar{\bar{x}})^2$$


$$SS_B = \sum_{j=1}^{K} n_j(\bar{x}_j - \bar{\bar{x}})^2$$

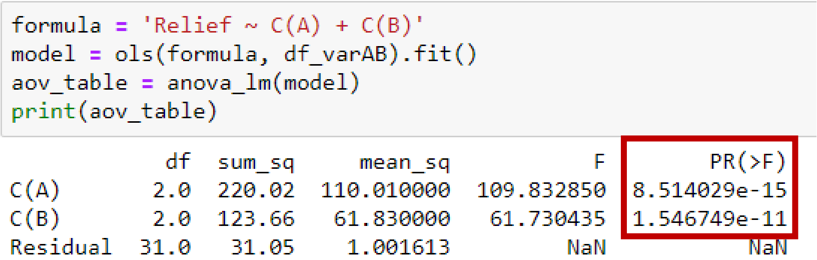

$$SS_W = \sum_k \sum_j \sum_i (x_{ijk} - \bar{x}_{ij})^2$$

After using the formulas, mentioned below is the F value(with interaction).

All the F values **$F_{STAT(A)}$, $F_{STAT(B)}$ & $F_{STAT(AB)}$** values are greater than **Fα = 3.35,3.35 & 2.73** respectively.

Variation	Degree of Freedom	Sum of Squares	Mean Square (Variance)	F_{STAT}
Factor A	$df_A = 3 - 1 = 2$	$SS_A =$ **220.02**	$MS_A =$ **110.01**	$F_A =$ **1827.858**
Factor B	$df_B = 3 - 1 = 2$	$SS_B =$ **123.66**	$MS_B =$ **61.83**	$F_B =$ **1027.329**
Interaction AB	$df_{AB} = 4$	$SS_{AB} =$ **29.42**	$MS_{AB} =$ **7.36**	$F_{AB} =$ **122.226**
Residual (error / within)	$df_W = 36 - (3*3) = 27$	$SS_W =$ **1.62**	$MS_W =$ **0.06**	
Total	$df_T = 35$	$SS_T =$ **374.73**		

Screenshot of python output (without interaction):

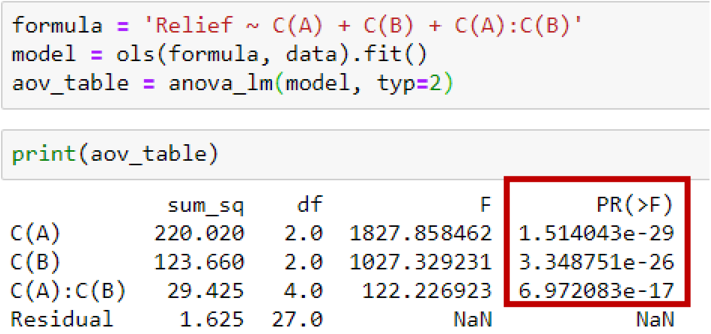


```
formula = 'Relief ~ C(A) + C(B)'
model = ols(formula, df_varAB).fit()
aov_table = anova_lm(model)
print(aov_table)
```

```
            df  sum_sq     mean_sq           F        PR(>F)
C(A)       2.0  220.02  110.010000  109.832850  8.514029e-15
C(B)       2.0  123.66   61.830000   61.730435  1.546749e-11
Residual  31.0   31.05    1.001613         NaN           NaN
```

For both ingredient A & ingredient B, **P value is less than 0.05. hence, ingredient A & B has the significant effect on relief time.**

Screenshot of python output (with interaction):



```
formula = 'Relief ~ C(A) + C(B) + C(A):C(B)'
model = ols(formula, data).fit()
aov_table = anova_lm(model, typ=2)
```

```
print(aov_table)
```

```
            sum_sq    df            F        PR(>F)
C(A)       220.020   2.0  1827.858462  1.514043e-29
C(B)       123.660   2.0  1027.329231  3.348751e-26
C(A):C(B)   29.425   4.0   122.226923  6.972083e-17
Residual     1.625  27.0          NaN           NaN
```

For both ingredient A & ingredient B, **P value is less than 0.05. hence, ingredient A & B has the significant effect on relief time**. Further, **P value of interaction effect between A & B is also less than 0.05. it means it has significant interaction effect**.

Conclusion:

We Reject the null hypothesis (H_0). Hence, we conclude that "both factors (ingredient A & ingredient B) have a statistically significant effect on relief time. Further, there is a significant interaction effect between ingredient A and ingredient B".

1.6 Mention the business implications of performing ANOVA for this particular case study.

ANOVA In the Business Context:

ANOVA is widely used across businesses and industries for a variety of purposes and is a technique that enables companies to identify problems, trends, risks and opportunities that impact both short and long-term viability. Below are some few considerations within given case study:

- Quality and cost comparison
- Product safety tests
- Optimize production

Benefits of ANOVA:

ANOVA has many benefits in both statistical and business contexts. It's often used when measuring financial data or indifferent management scenarios. Companies can create new opportunities, spot potential issues, and learn to understand what is driving behavior.

- **Hypothesis Testing:** Enables the comparison of independent and dependent variables.
- **Understanding Data Sets:** An analyst or statistician can best determine inconsistencies in data sets.
- **Group Comparisons:** Allows multiple groups to be compared at the same time to uncover relationships between data.
- **Sales and Marketing Improvement:** Businesses can answer customer and product research questions to improve advertising and marketing for better sales.
- **Project Management:** Leadership, such as project management, can better align their goals and strategies with business and departmental cost objectives.
- **Industry-Wide Approach:** ANOVA is effective for a wide variety of uses across different industries, including financial services, eCommerce, industrial, R&D, and more.
- **Product Development:** Organizations can better pinpoint and understand what product features to improve or adapt for the best results.

Business Implications:

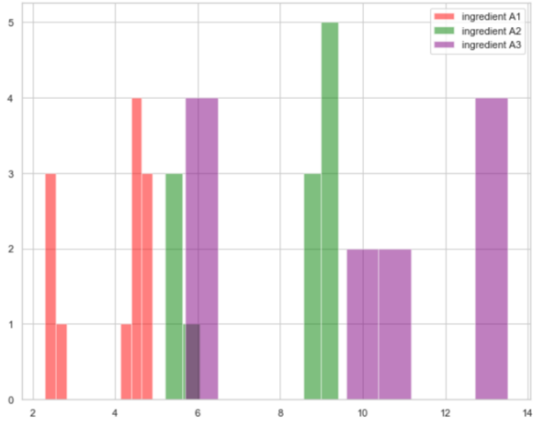
The purpose of this case study (a two-way ANOVA) is to determine how two factors (here ingredient "A" and "B") impact a response variable (here "Relief"), and to determine whether or not there is an interaction between the two factors on the response variable.

Descriptive statistics allowed to determining the mean of the independent variable A and B, and the dependent variable, relief time. Using the mean of the independent variables as reference, and computed a two-way ANOVA to analyze the data.

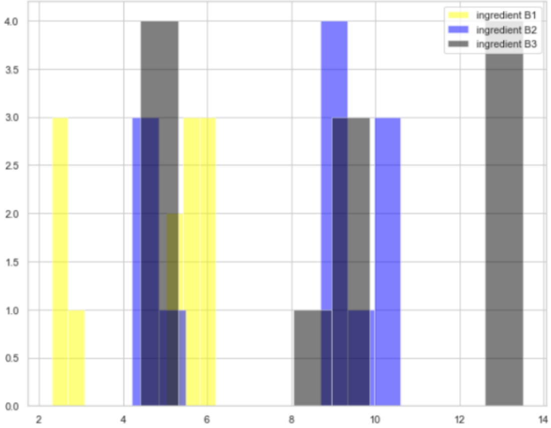
The ANOVA was significant, the effect size was strong, allowing to rejecting the null hypothesis, and indicating that there is a statistically significant relationship with strong effect size between ingredient A and ingredient B and relief time.

Furthermore,

Distribution plot of ingredient A with 3 variants says that, they are quite different then each other. There is only small overlay happening at point 6 otherwise they are quite different. Therefore, we can conclude that the ingredient A with all 3 variants have different treatment.



Distribution plot of ingredient B with 3 variants says that: For variants B2(blue) and B3(black), have somewhat overlay between them. For, the third one B1 (yellow) the distribution seems quite different then the other two. Therefore, we can conclude that the ingredient B with all 3 variants have different treatment.



Final Summary: As these treatments are quite different from each other, look for more samples or other variants which increases product quality and reduces time effort and increases relief time.

Problem2:

A company performed a survey to understand the income of households in various neighborhoods of a country. The data dictionary is also present. You can access the data dictionary from the following file Income_Data Dictionary. Please refer to the following data set to solve the problem Income.csv.

['FamilyIncome' is the target variable]

Remarks: All the questions of problem1 which explained here, are also performed in python as well. Please refer python notebook "AS_Individual_Assignment_Problem2".

2.1 Perform exploratory data analysis on the dataset. Showcase some charts, graphs.

Background:

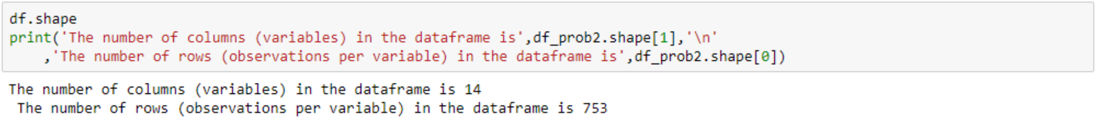
In statistics, exploratory data analysis (EDA) is an approach to analyzing datasets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task. In other words, EDA refers to the critical process of performing initial investigations on data so as:

- to discover patterns,
- to spot anomalies,
- to test hypothesis and
- to check assumptions

with the help of summary statistics and graphical representations.

In this problem, we have been instructed to use Income dataset and perform EDA on the same.

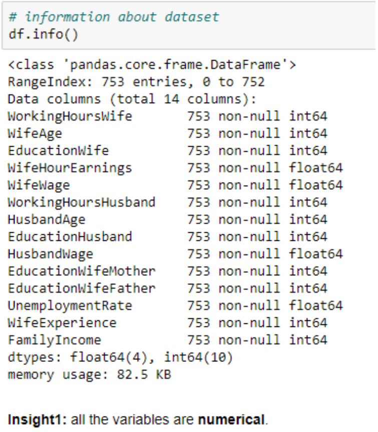
df.shape: Total number of rows and columns in the data set using ".shape". in this dataset we have 753 rows and 14 columns.



```
df.shape
print('The number of columns (variables) in the dataframe is',df_prob2.shape[1],'\n'
    ,'The number of rows (observations per variable) in the dataframe is',df_prob2.shape[0])
```

```
The number of columns (variables) in the dataframe is 14
 The number of rows (observations per variable) in the dataframe is 753
```

df.info(): This function helps to get information about the dataset. in this dataset, we have columns/variables are as Numerical.

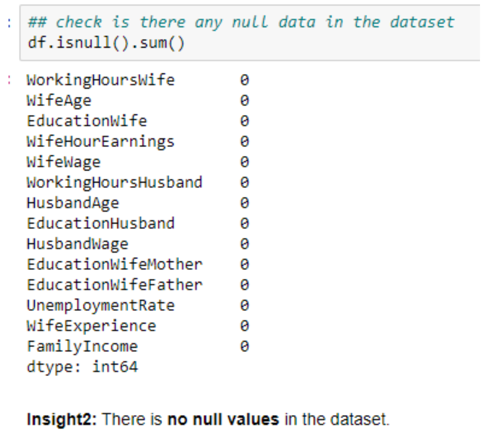


```
# information about dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 753 entries, 0 to 752
Data columns (total 14 columns):
WorkingHoursWife       753 non-null int64
WifeAge                753 non-null int64
EducationWife          753 non-null int64
WifeHourEarnings       753 non-null float64
WifeWage               753 non-null float64
WorkingHoursHusband    753 non-null int64
HusbandAge             753 non-null int64
EducationHusband       753 non-null int64
HusbandWage            753 non-null float64
EducationWifeMother    753 non-null int64
EducationWifeFather    753 non-null int64
UnemploymentRate       753 non-null float64
WifeExperience         753 non-null int64
FamilyIncome           753 non-null int64
dtypes: float64(4), int64(10)
memory usage: 82.5 KB
```

Insight1: all the variables are **numerical**.

df.isnull().sum(): This function helps to see is there any null values in the dataset. In this dataset, we don’t have any null values.

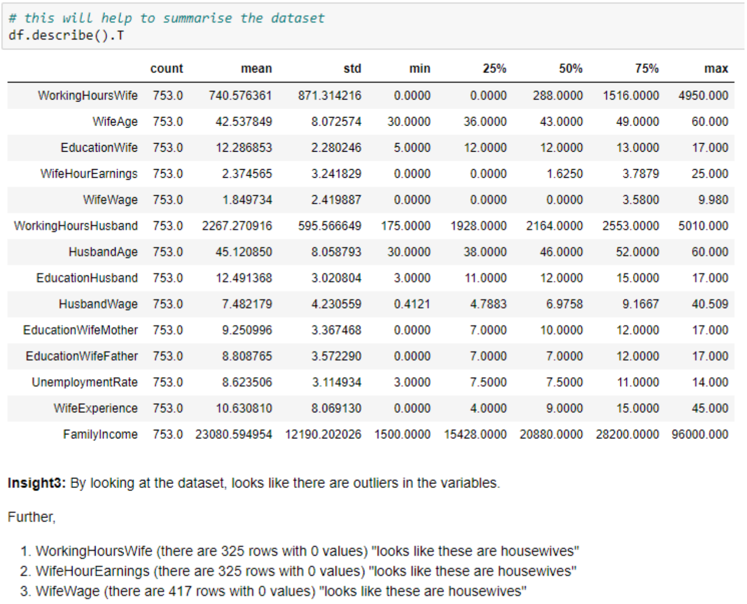


```
## check is there any null data in the dataset
df.isnull().sum()
```

```
WorkingHoursWife        0
WifeAge                 0
EducationWife           0
WifeHourEarnings        0
WifeWage                0
WorkingHoursHusband     0
HusbandAge              0
EducationHusband        0
HusbandWage             0
EducationWifeMother     0
EducationWifeFather     0
UnemploymentRate        0
WifeExperience          0
FamilyIncome            0
dtype: int64
```

Insight2: There is **no null values** in the dataset.

df.describe().T: This function helps to get transpose version on descriptive stats summary which helps to decode the dataset.



```
# this will help to summarise the dataset
df.describe().T
```

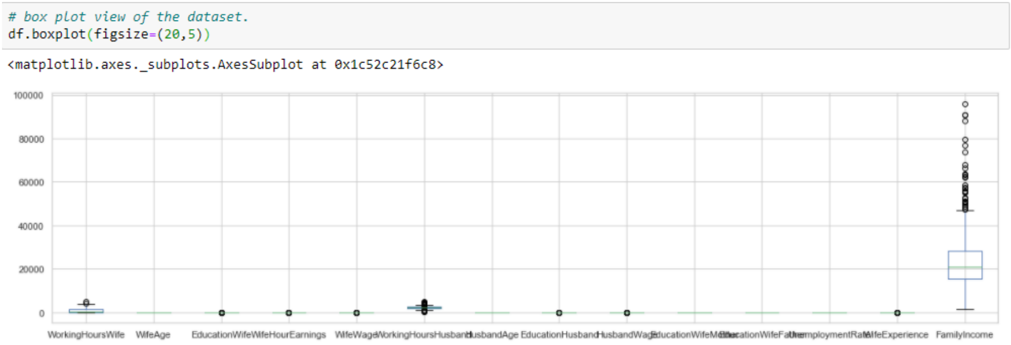
	count	mean	std	min	25%	50%	75%	max
WorkingHoursWife	753.0	740.576361	871.314216	0.0000	0.0000	288.0000	1516.0000	4950.000
WifeAge	753.0	42.537849	8.072574	30.0000	36.0000	43.0000	49.0000	60.000
EducationWife	753.0	12.286853	2.280246	5.0000	12.0000	12.0000	13.0000	17.000
WifeHourEarnings	753.0	2.374565	3.241829	0.0000	0.0000	1.6250	3.7879	25.000
WifeWage	753.0	1.849734	2.419887	0.0000	0.0000	0.0000	3.5800	9.980
WorkingHoursHusband	753.0	2267.270916	595.566649	175.0000	1928.0000	2164.0000	2553.0000	5010.000
HusbandAge	753.0	45.120850	8.058793	30.0000	38.0000	46.0000	52.0000	60.000
EducationHusband	753.0	12.491368	3.020804	3.0000	11.0000	12.0000	15.0000	17.000
HusbandWage	753.0	7.482179	4.230559	0.4121	4.7883	6.9758	9.1667	40.509
EducationWifeMother	753.0	9.250996	3.367468	0.0000	7.0000	10.0000	12.0000	17.000
EducationWifeFather	753.0	8.808765	3.572290	0.0000	7.0000	7.0000	12.0000	17.000
UnemploymentRate	753.0	8.623506	3.114934	3.0000	7.5000	7.5000	11.0000	14.000
WifeExperience	753.0	10.630810	8.069130	0.0000	4.0000	9.0000	15.0000	45.000
FamilyIncome	753.0	23080.594954	12190.202026	1500.0000	15428.0000	20880.0000	28200.0000	96000.000

Insight3: By looking at the dataset, looks like there are outliers in the variables.

Further,

1. WorkingHoursWife (there are 325 rows with 0 values) "looks like these are housewives"
2. WifeHourEarnings (there are 325 rows with 0 values) "looks like these are housewives"
3. WifeWage (there are 417 rows with 0 values) "looks like these are housewives"

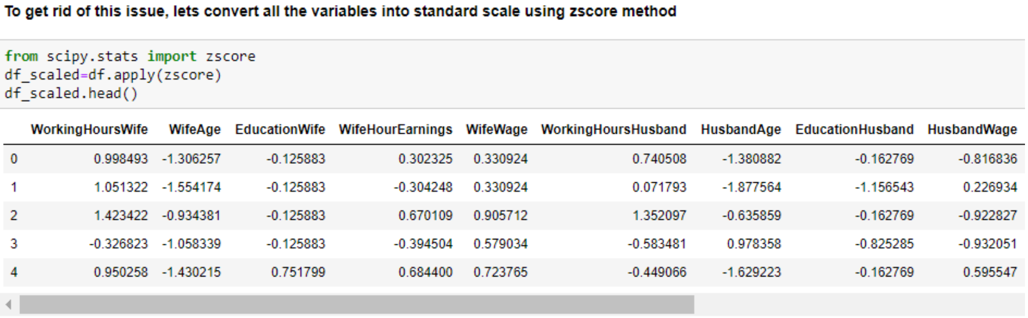
df.boxplot(): This function helps to do the univariate analysis of the dataset. By looking at the view, it looks like there are 9 variables out of total 14 has the outliers. Further, there is a huge scale variation in the dataset hence couldn’t interpret the result. So, let’s do data standardisation using z-score to make them on a same scale.



```
# box plot view of the dataset.
df.boxplot(figsize=(20,5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c52c21f6c8>
```

Data Standardisation (Z score technique): This function helps to do convert all the variables into same scale.

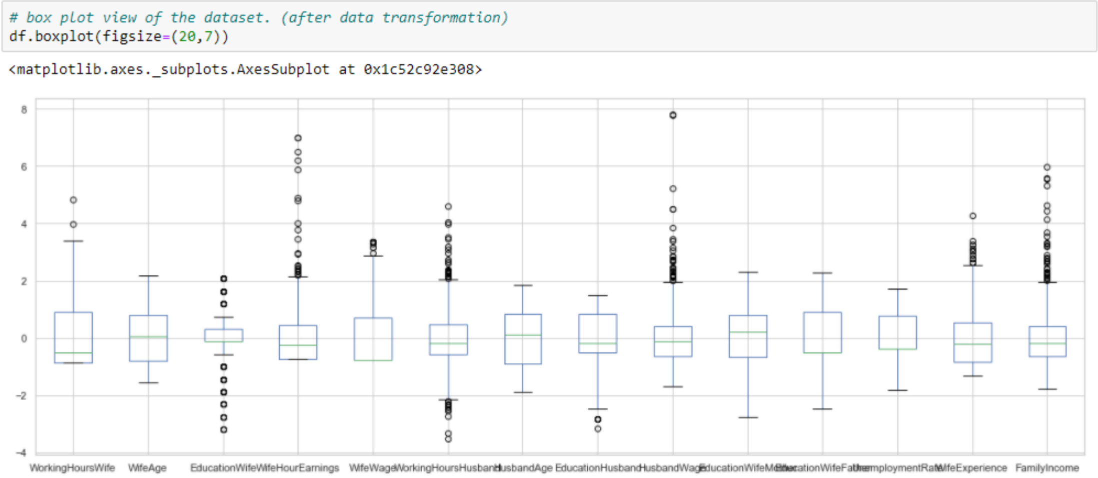


To get rid of this issue, lets convert all the variables into standard scale using zscore method

```
from scipy.stats import zscore
df_scaled=df.apply(zscore)
df_scaled.head()
```

	WorkingHoursWife	WifeAge	EducationWife	WifeHourEarnings	WifeWage	WorkingHoursHusband	HusbandAge	EducationHusband	HusbandWage
0	0.998493	-1.306257	-0.125883	0.302325	0.330924	0.740508	-1.380882	-0.162769	-0.816836
1	1.051322	-1.554174	-0.125883	-0.304248	0.330924	0.071793	-1.877564	-1.156543	0.226934
2	1.423422	-0.934381	-0.125883	0.670109	0.905712	1.352097	-0.635859	-0.162769	-0.922827
3	-0.326823	-1.058339	-0.125883	-0.394504	0.579034	-0.583481	0.978358	-0.825285	-0.932051
4	0.950258	-1.430215	0.751799	0.684400	0.723765	-0.449066	-1.629223	-0.162769	0.595547

Box plot view after Data Standardisation: Outlier are clearly visible after above activity.

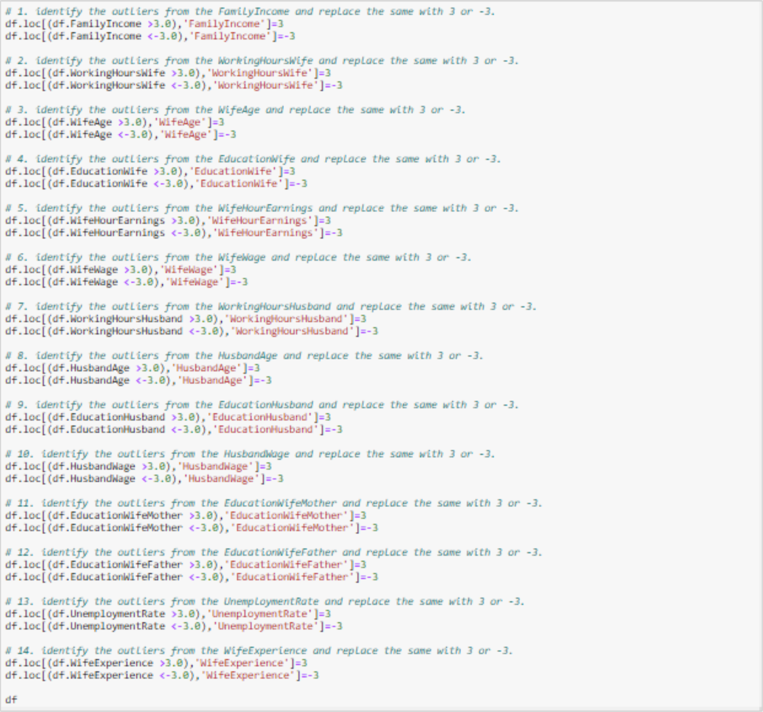


```
# box plot view of the dataset. (after data transformation)
df.boxplot(figsize=(20,7))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c52c92e308>
```

As we can see the outlier in the dataset let's replace the outliers using Data Transformation:

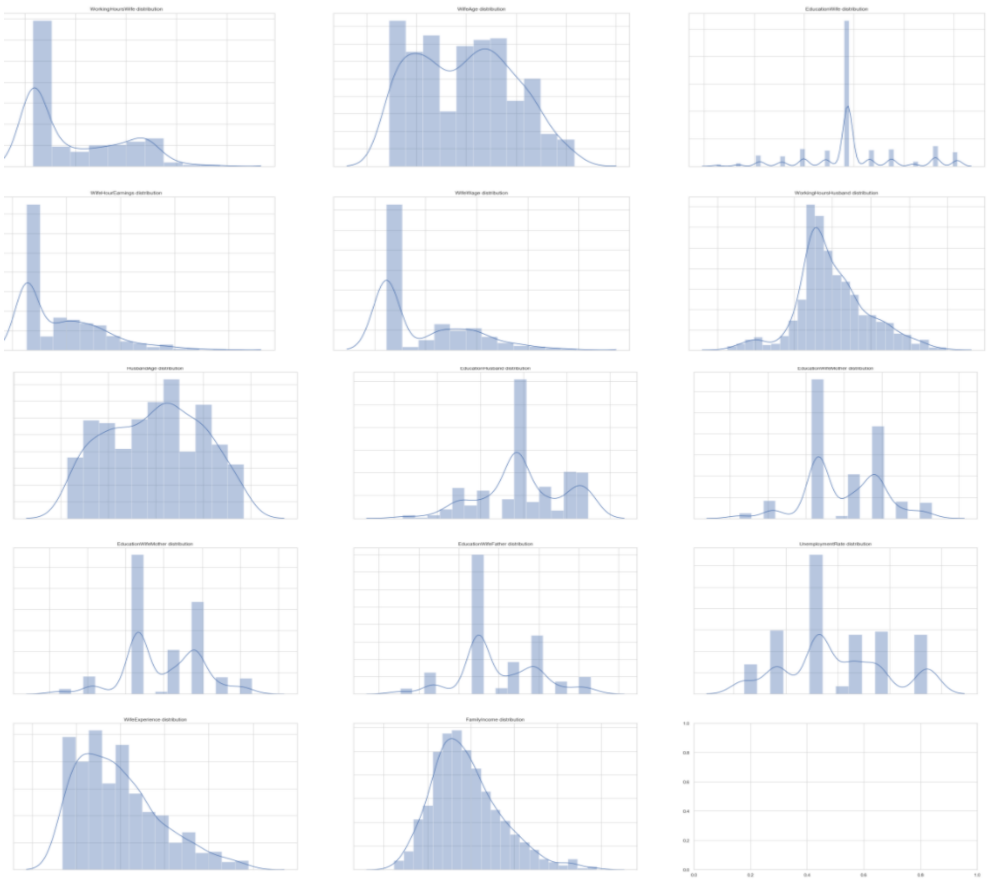
as per imperial rule, z score value between -3 to 3 contains covers 99.7% of the data. So, let's replace the outliers in each variable where zscore value is more then 3 and less than -3 with 3 and -3 respectively.



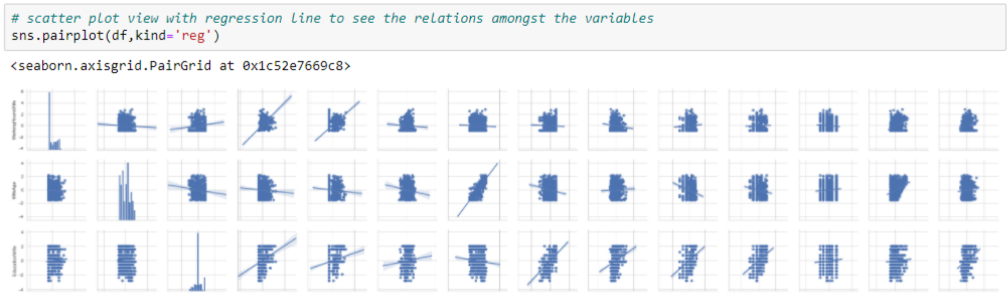
```
# 1. identify the outliers from the FamilyIncome and replace the same with 3 or -3.
df.loc[(df.FamilyIncome >3.0),'FamilyIncome']=3
df.loc[(df.FamilyIncome <-3.0),'FamilyIncome']=-3

# 2. identify the outliers from the WorkingHoursWife and replace the same with 3 or -3.
df.loc[(df.WorkingHoursWife >3.0),'WorkingHoursWife']=3
df.loc[(df.WorkingHoursWife <-3.0),'WorkingHoursWife']=-3

# 3. identify the outliers from the WifeAge and replace the same with 3 or -3.
df.loc[(df.WifeAge >3.0),'WifeAge']=3
df.loc[(df.WifeAge <-3.0),'WifeAge']=-3

# 4. identify the outliers from the EducationWife and replace the same with 3 or -3.
df.loc[(df.EducationWife >3.0),'EducationWife']=3
df.loc[(df.EducationWife <-3.0),'EducationWife']=-3

# 5. identify the outliers from the WifeHourEarnings and replace the same with 3 or -3.
df.loc[(df.WifeHourEarnings >3.0),'WifeHourEarnings']=3
df.loc[(df.WifeHourEarnings <-3.0),'WifeHourEarnings']=-3

# 6. identify the outliers from the WifeWage and replace the same with 3 or -3.
df.loc[(df.WifeWage >3.0),'WifeWage']=3
df.loc[(df.WifeWage <-3.0),'WifeWage']=-3

# 7. identify the outliers from the WorkingHoursHusband and replace the same with 3 or -3.
df.loc[(df.WorkingHoursHusband >3.0),'WorkingHoursHusband']=3
df.loc[(df.WorkingHoursHusband <-3.0),'WorkingHoursHusband']=-3

# 8. identify the outliers from the HusbandAge and replace the same with 3 or -3.
df.loc[(df.HusbandAge >3.0),'HusbandAge']=3
df.loc[(df.HusbandAge <-3.0),'HusbandAge']=-3

# 9. identify the outliers from the EducationHusband and replace the same with 3 or -3.
df.loc[(df.EducationHusband >3.0),'EducationHusband']=3
df.loc[(df.EducationHusband <-3.0),'EducationHusband']=-3

# 10. identify the outliers from the HusbandWage and replace the same with 3 or -3.
df.loc[(df.HusbandWage >3.0),'HusbandWage']=3
df.loc[(df.HusbandWage <-3.0),'HusbandWage']=-3

# 11. identify the outliers from the EducationWifeMother and replace the same with 3 or -3.
df.loc[(df.EducationWifeMother >3.0),'EducationWifeMother']=3
df.loc[(df.EducationWifeMother <-3.0),'EducationWifeMother']=-3

# 12. identify the outliers from the EducationWifeFather and replace the same with 3 or -3.
df.loc[(df.EducationWifeFather >3.0),'EducationWifeFather']=3
df.loc[(df.EducationWifeFather <-3.0),'EducationWifeFather']=-3

# 13. identify the outliers from the UnemploymentRate and replace the same with 3 or -3.
df.loc[(df.UnemploymentRate >3.0),'UnemploymentRate']=3
df.loc[(df.UnemploymentRate <-3.0),'UnemploymentRate']=-3

# 14. identify the outliers from the WifeExperience and replace the same with 3 or -3.
df.loc[(df.WifeExperience >3.0),'WifeExperience']=3
df.loc[(df.WifeExperience <-3.0),'WifeExperience']=-3

df
```

After **replacing the outliers, 8% data got replaced with 3 or -3 with same no of rows in the dataset.**

Histogram/Distribution plot: This plot helps to check the linearity of the variables it is a good practice to plot distribution graph and look for skewness of features. Kernel density estimate (kde) is a quite useful tool for plotting the shape of a distribution



Scatter Plot: The same way scatter plot also helps to visualise the linear relationship between the variables. Glimpse of the scatter plot view from the dataset is as below. However, the full view is already available in the python notebook.



```
# scatter plot view with regression line to see the relations amongst the variables
sns.pairplot(df,kind='reg')
```

```
<seaborn.axisgrid.PairGrid at 0x1c52e7669c8>
```

By this way, I have concluded the EDA portion. There are many insights which came out and would be useful for next set of steps during problem solving.

2.2 Is there evidence of multicollinearity? Showcase your analysis.

Background:

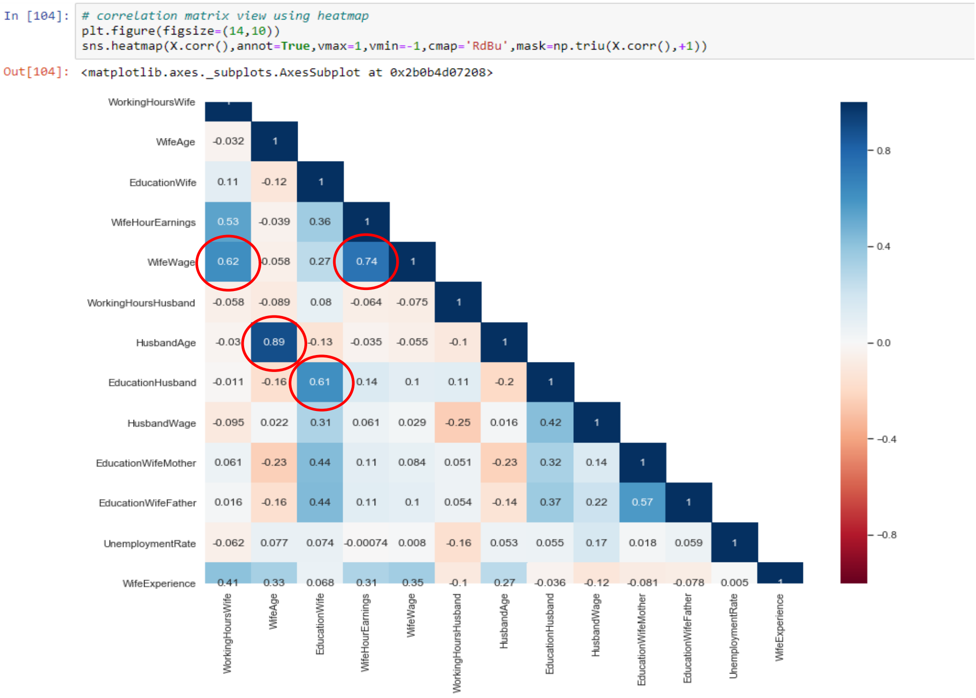
Multicollinearity generally occurs when there are *high correlations between two or more predictor variables*. In other words, one predictor variable can be used to predict the other. This creates redundant information, skewing the results in a regression model.

An easy way to detect multicollinearity is to calculate correlation coefficients for all pairs of predictor variables. If the correlation coefficient, r, is exactly +1 or -1, this is called perfect multicollinearity. If r is close to or exactly -1 or +1, one of the variables should be removed from the model if at all possible.

- If correlation is ZERO (0), it means independent variables are not joining hands to predict dependent variable
- If correlation is ONE (1), it's difficult to make accurate model.

So, anything greater than 0.6 in correlation, considered to be multicollinearity. Some statisticians called "Multicollinearity as a cancer in statistics".

Let's perform the multicollinearity (***Test of Assumption1: The independent variables should not be correlated)***



```
In [104]: # correlation matrix view using heatmap
          plt.figure(figsize=(14,10))
          sns.heatmap(X.corr(),annot=True,vmax=1,vmin=-1,cmap='RdBu',mask=np.triu(X.corr(),+1))

Out[104]: <matplotlib.axes._subplots.AxesSubplot at 0x2b0b4d07208>
```

	WorkingHoursWife	WifeAge	EducationWife	WifeHourEarnings	WifeWage	WorkingHoursHusband	HusbandAge	EducationHusband	HusbandWage	EducationWifeMother	EducationWifeFather	UnemploymentRate	WifeExperience
WorkingHoursWife	1												
WifeAge	-0.032	1											
EducationWife	0.11	-0.12	1										
WifeHourEarnings	0.53	-0.039	0.36	1									
WifeWage	0.62	-0.058	0.27	0.74	1								
WorkingHoursHusband	-0.058	-0.089	0.08	-0.064	-0.075	1							
HusbandAge	-0.03	0.89	-0.13	-0.035	-0.055	-0.1	1						
EducationHusband	-0.011	-0.16	0.61	0.14	0.1	0.11	-0.2	1					
HusbandWage	-0.095	0.022	0.31	0.061	0.029	-0.25	0.016	0.42	1				
EducationWifeMother	0.061	-0.23	0.44	0.11	0.084	0.051	-0.23	0.32	0.14	1			
EducationWifeFather	0.016	-0.16	0.44	0.11	0.1	0.054	-0.14	0.37	0.22	0.57	1		
UnemploymentRate	-0.062	0.077	0.074	-0.00074	0.008	-0.16	0.053	0.055	0.17	0.018	0.059	1	
WifeExperience	0.41	0.33	0.068	0.31	0.35	-0.1	0.27	-0.036	-0.12	-0.081	-0.078	0.005	1

Based on above heat map graph, it is obvious that mentioned below 4 combinations have the high correlation.

- HusbandAge and WifeAge **(0.89)**
- WifeWage and WifeHourEarnings **(0.74)**
- WifeWage and WorkingHoursWife **(0.62)**
- EducationHusband and EducationWife **(0.61)**

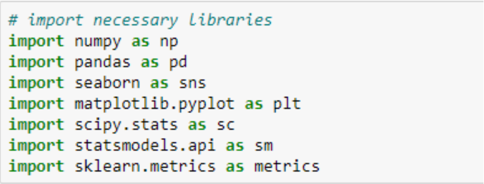
Conclusion: Hence, we can conclude that there is **some degree of correlation (multicollinearity) amongst the variables** in the dataset.

2.3 Perform Multiple Linear Regression (using the 'statsmodels' library) and comment on the model thus built.

As we have already checked assumption1: no or little multicollinearity in problem2.2. let's check other assumptions and build the model accordingly.

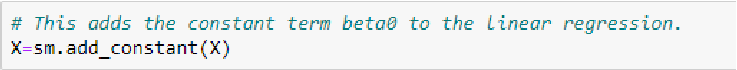
Test of Assumption 2: (Linear Relationship means that the dependent variable should be linearly related with the coefficients)

Import necessary libraries:



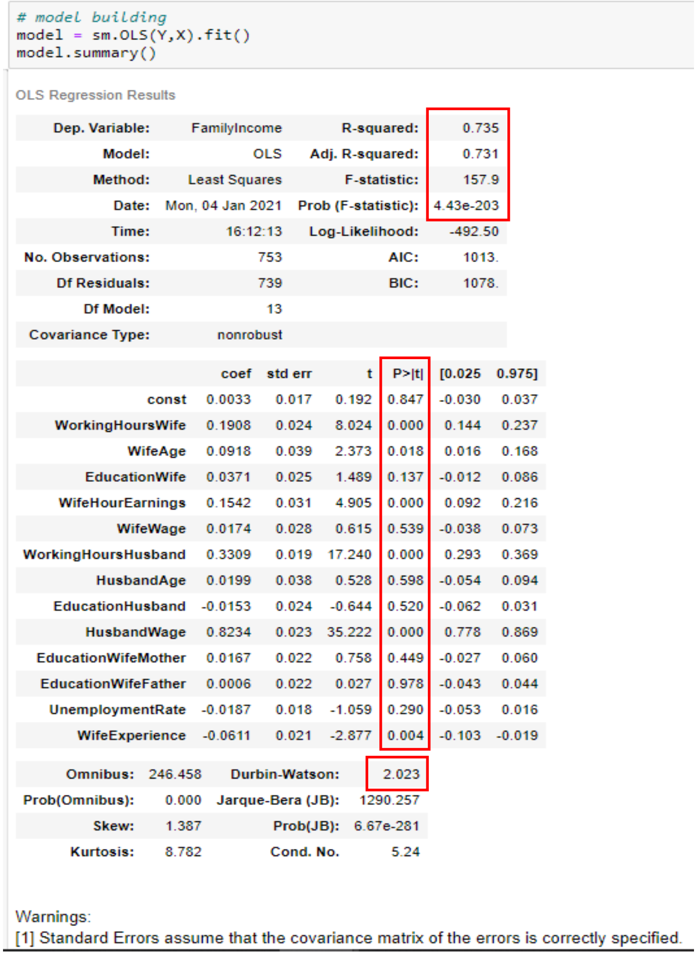
```
# import necessary libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as sc
import statsmodels.api as sm
import sklearn.metrics as metrics
```

Add constant value (i.e., β_0) in the dataset:



```
# This adds the constant term beta0 to the linear regression.
X=sm.add_constant(X)
```

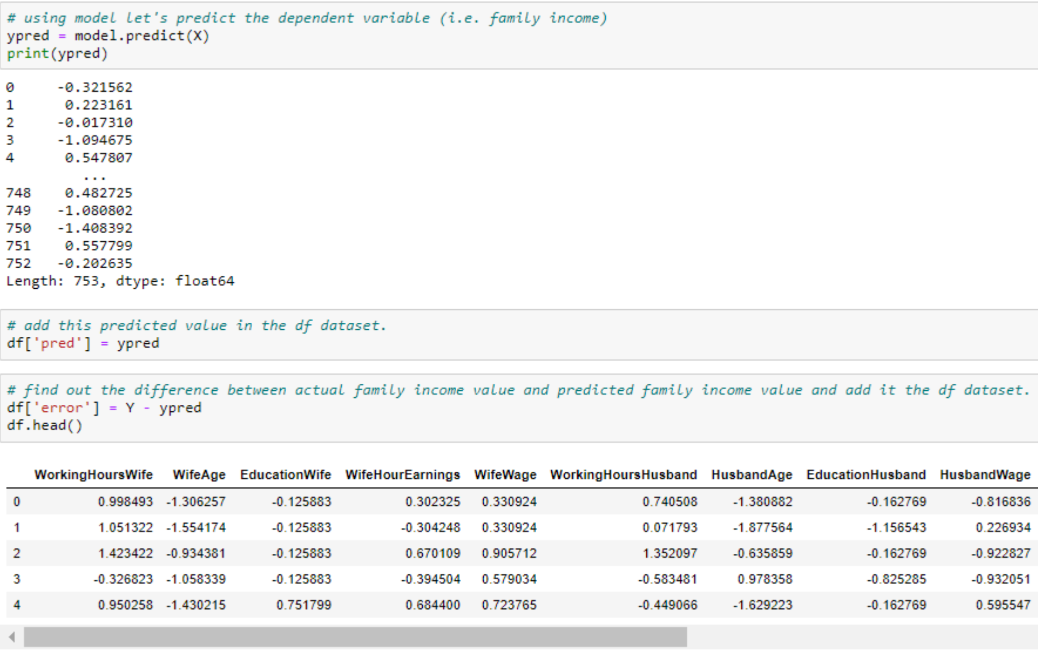
Run the model by executing below code and interpret the results:



```
# model building
model = sm.OLS(Y,X).fit()
model.summary()
```

```
OLS Regression Results
      Dep. Variable:      FamilyIncome          R-squared:      0.735
             Model:                OLS      Adj. R-squared:     0.731
            Method:      Least Squares         F-statistic:     157.9
              Date:   Mon, 04 Jan 2021  Prob (F-statistic):  4.43e-203
              Time:           16:12:13      Log-Likelihood:    -492.50
  No. Observations:                753                 AIC:     1013.
      Df Residuals:                739                 BIC:     1078.
          Df Model:                 13
   Covariance Type:          nonrobust

                           coef  std err       t   P>|t|  [0.025   0.975]
                const    0.0033    0.017   0.192   0.847  -0.030    0.037
     WorkingHoursWife    0.1908    0.024   8.024   0.000   0.144    0.237
              WifeAge    0.0918    0.039   2.373   0.018   0.016    0.168
        EducationWife    0.0371    0.025   1.489   0.137  -0.012    0.086
     WifeHourEarnings    0.1542    0.031   4.905   0.000   0.092    0.216
             WifeWage    0.0174    0.028   0.615   0.539  -0.038    0.073
  WorkingHoursHusband    0.3309    0.019  17.240   0.000   0.293    0.369
          HusbandAge     0.0199    0.038   0.528   0.598  -0.054    0.094
     EducationHusband   -0.0153    0.024  -0.644   0.520  -0.062    0.031
          HusbandWage    0.8234    0.023  35.222   0.000   0.778    0.869
  EducationWifeMother    0.0167    0.022   0.758   0.449  -0.027    0.060
  EducationWifeFather    0.0006    0.022   0.027   0.978  -0.043    0.044
     UnemploymentRate   -0.0187    0.018  -1.059   0.290  -0.053    0.016
       WifeExperience   -0.0611    0.021  -2.877   0.004  -0.103   -0.019

        Omnibus:  246.458     Durbin-Watson:      2.023
  Prob(Omnibus):    0.000  Jarque-Bera (JB):   1290.257
           Skew:    1.387          Prob(JB):  6.67e-281
       Kurtosis:    8.782          Cond. No.       5.24

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Using this model, predict the y value and then find out the errors between actual y value in the dataset and newly predicted y value:



```
# using model let's predict the dependent variable (i.e. family income)
ypred = model.predict(X)
print(ypred)
```

```
0     -0.321562
1      0.223161
2     -0.017310
3     -1.094675
4      0.547807
         ...
748    0.482725
749   -1.080802
750   -1.408392
751    0.557799
752   -0.202635
Length: 753, dtype: float64
```

```
# add this predicted value in the df dataset.
df['pred'] = ypred
```

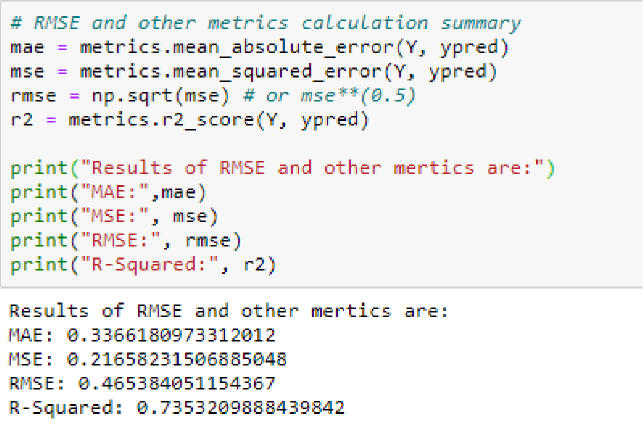
```
# find out the difference between actual family income value and predicted family income value and add it the df dataset.
df['error'] = Y - ypred
df.head()
```

	WorkingHoursWife	WifeAge	EducationWife	WifeHourEarnings	WifeWage	WorkingHoursHusband	HusbandAge	EducationHusband	HusbandWage
0	0.998493	-1.306257	-0.125883	0.302325	0.330924	0.740508	-1.380882	-0.162769	-0.816836
1	1.051322	-1.554174	-0.125883	-0.304248	0.330924	0.071793	-1.877564	-1.156543	0.226934
2	1.423422	-0.934381	-0.125883	0.670109	0.905712	1.352097	-0.635859	-0.162769	-0.922827
3	-0.326823	-1.058339	-0.125883	-0.394504	0.579034	-0.583481	0.978358	-0.825285	-0.932051
4	0.950258	-1.430215	0.751799	0.684400	0.723765	-0.449066	-1.629223	-0.162769	0.595547

Calculate RMSE and other metrics for to check errors in the model:

Background RMSE: The RMSE is the square root of the variance of the residuals. It indicates the **absolute fit of the model to the data–how close the observed data points are to the model's predicted values**. Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit.

As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. **Lower values of RMSE indicate better fit**. RMSE is a good measure of **how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction.**

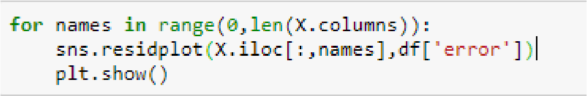


```
# RMSE and other metrics calculation summary
mae = metrics.mean_absolute_error(Y, ypred)
mse = metrics.mean_squared_error(Y, ypred)
rmse = np.sqrt(mse) # or mse**(0.5)
r2 = metrics.r2_score(Y, ypred)

print("Results of RMSE and other mertics are:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
```

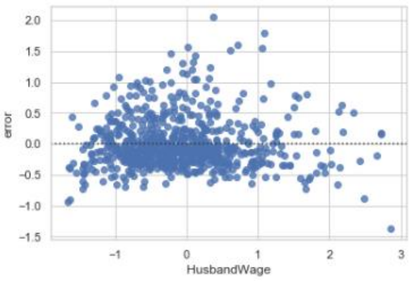
```
Results of RMSE and other mertics are:
MAE: 0.3366180973312012
MSE: 0.21658231506885048
RMSE: 0.465384051154367
R-Squared: 0.7353209888439842
```

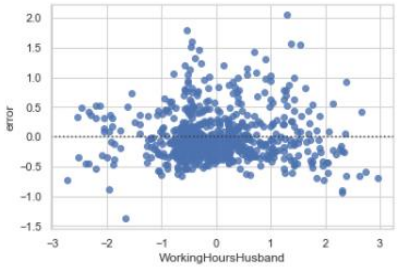
Test of Assumption 3: (The error terms have a constant variance i.e., they are homoscedastic in nature)

Below is the python code helps to get each variable wise relation to the residual value. Glimpse of the scatter plot view from the dataset is as below. However, the full view is already available in the python notebook.



```
for names in range(0,len(X.columns)):
    sns.residplot(X.iloc[:,names],df['error'])
    plt.show()
```



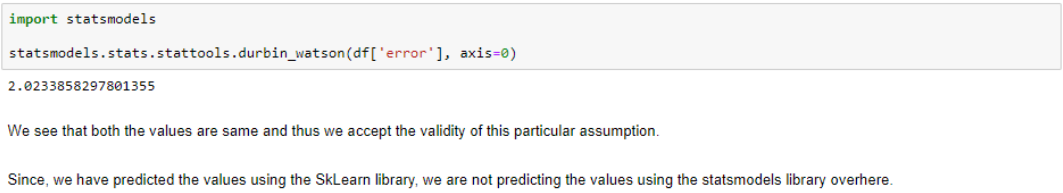


Test of Assumption 4: There should not be any auto-correlation between the error terms. (One value of the error term should not predict the next value of the error term)

Here, we notice that there are mix nature in p-value. There are **6 variables** for which p-values is less than 0.05 but there are **7 variables** for which p-value is greater than 0.05 but we will not be dropping that variables as intuitively that seems like an important variable.

The variation in the dependent variable which is explained by the independent variables is 73.53%

Here, we see that the Durbin-Watson test statistic is close to 2 and thus we can say that this particular assumption of Linear Regression is also verified.



```
import statsmodels

statsmodels.stats.stattools.durbin_watson(df['error'], axis=0)
```

```
2.0233858297801355
```

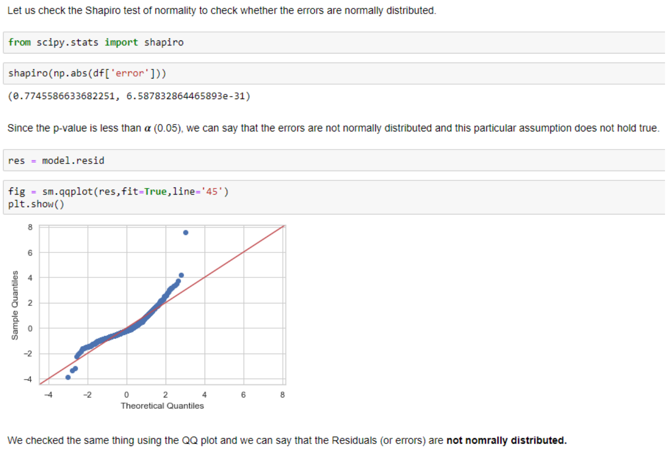
We see that both the values are same and thus we accept the validity of this particular assumption.

Since, we have predicted the values using the SkLearn library, we are not predicting the values using the statsmodels library overhere.

Below is the python code helps to get each variable wise relation to the residual value. Glimpse of the scatter plot view from the dataset is as below. However, the full view is already available in the python notebook.

Test of Assumption 5: The errors are assumed to be normally distributed

Let us check the Shapiro test of normality to check whether the errors are normally distributed.



Let us check the Shapiro test of normality to check whether the errors are normally distributed.

```
from scipy.stats import shapiro
```

```
shapiro(np.abs(df['error']))
```

```
(0.7745586633682251, 6.587832864465893e-31)
```

Since the p-value is less than α (0.05), we can say that the errors are not normally distributed and this particular assumption does not hold true.

```
res = model.resid
```

```
fig = sm.qqplot(res,fit=True,line='45')
plt.show()
```

We checked the same thing using the QQ plot and we can say that the Residuals (or errors) are **not nomrally distributed.**

Final Summary(snapshot) on problem 2.3:

There are **13 independent variables and 753 rows** in the dataset *(after replacing the outliers zscore value more than 3 or less than -3 with 3 and -3 respectively).* I have looked at all 5 assumptions to assess whether model is good to go or not? and found:

3 out of 5 assumptions **are failing to satisfy** the model passing requirement. Further,

I have checked **multicollinearity** and found 4 combinations with high collinearity (correlation values more than 0.6). **(Not Satisfying)**

Also checked **linear relationship** amongst independent variables and dependent variables are found: **(Not Satisfying)**

R^2 (coefficient of determination) value is **0.735** which means the variation in the dependent variable which is explained by the independent variables is only **73.5%** (if the R^2 value is less than 0.75 than model is considered to be weak).

Out of 13 independent variables, **7 variables** having p-value more than (α) 0.05. it means that these variables are not making significant impact to the dependent variable.

We have RMSE (root mean square error) value for this model is **0.465**.

Homoscedasticity amongst the errors (residuals) are following the pattern which is required. **(Satisfying)**

Also checked **auto correlations** amongst the errors (residuals) through Durbin- Watson test and the value were **2.023** which is in-line with suggested value of between 2 to 2.5. **(Satisfying)**

As per assumption **errors should be normally distributed.** But while using Shapiro's test found p-value **6.59e-31** which less than (α) 0.05. hence, errors are not normally distributed. **(Not Satisfying)**

Conclusion: This linear regression model is **not advised to use to predict "family income".** To predict "family income" some variables should be **dropped or merged**. So, **PCA (principal component analysis)** is the next option to go with and then reassess the model.

2.4 Perform Principal Component Analysis (on the predictor variables) and extract the Principal Components. Comment on the reason behind choosing the number of Principal Components.

Background:

Principal Component Analysis(PCA), is a dimension reduction technique that is often used to **reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one** that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

So, to sum up, the idea of PCA is simple — reduce the number of variables of a dataset, while preserving as much information as possible.

Let’s Perform PCA step by step.

Step1: Data Standardisation

Background: The aim of this step is to standardize the range of the continuous variables so that each one of them contributes equally to the analysis.

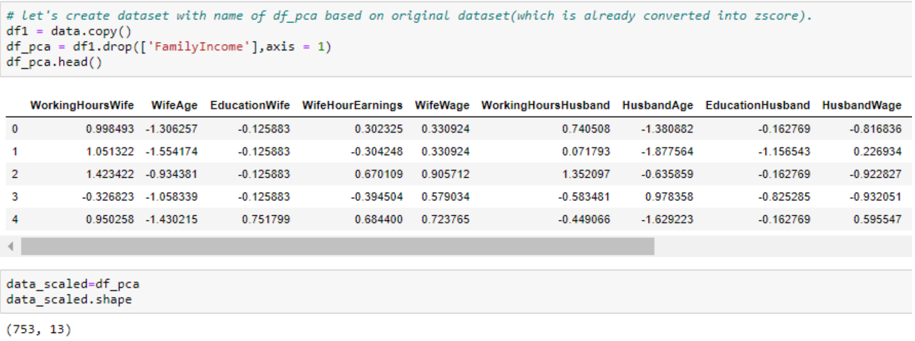
More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables. That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (For example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results. So, transforming the data to comparable scales can prevent this problem.

Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$Z\ score = \frac{(value\ \ - mean)}{std\ dev.}$$

Principal Component Analysis Standardization Once the standardization is done, all the variables will be transformed to the same scale.

For our dataset we have already completed this step after importing the dataset. Hence, we don’t need to perform this again. We will just make a copy of the dataset and complete the other steps.



```
# let's create dataset with name of df_pca based on original dataset(which is already converted into zscore).
df1 = data.copy()
df_pca = df1.drop(['FamilyIncome'],axis = 1)
df_pca.head()
```

	WorkingHoursWife	WifeAge	EducationWife	WifeHourEarnings	WifeWage	WorkingHoursHusband	HusbandAge	EducationHusband	HusbandWage
0	0.998493	-1.306257	-0.125883	0.302325	0.330924	0.740508	-1.380882	-0.162769	-0.816836
1	1.051322	-1.554174	-0.125883	-0.304248	0.330924	0.071793	-1.877564	-1.156543	0.226934
2	1.423422	-0.934381	-0.125883	0.670109	0.905712	1.352097	-0.635859	-0.162769	-0.922827
3	-0.326823	-1.058339	-0.125883	-0.394504	0.579034	-0.583481	0.978358	-0.825285	-0.932051
4	0.950258	-1.430215	0.751799	0.684400	0.723765	-0.449066	-1.629223	-0.162769	0.595547

```
data_scaled=df_pca
data_scaled.shape
```

```
(753, 13)
```

Step2: Covariance Matrix Computation

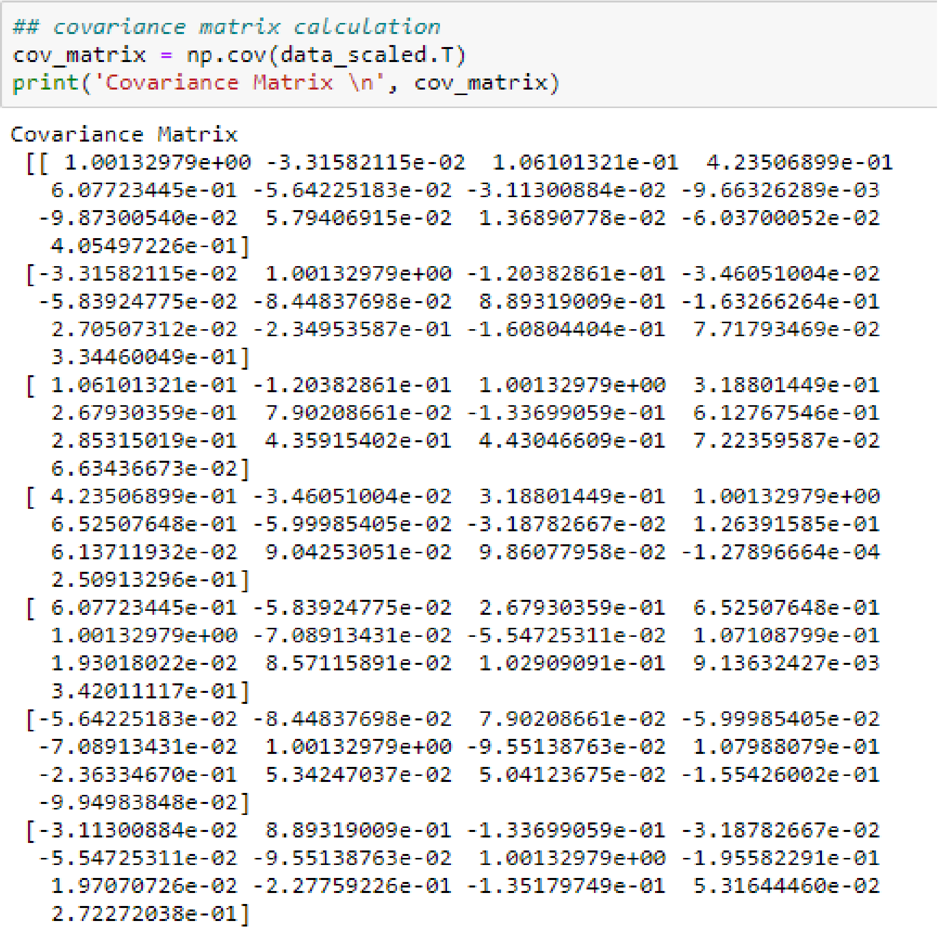
Background: The aim of this step is to understand how the variables of the input dataset are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

The covariance matrix is a p × p symmetric matrix (where p is the number of dimensions) that has as entries the covariances associated with all possible pairs of the initial variables.

It's actually the sign of the covariance that matters:

- if **+ve** then: the two variables increase or decrease together (correlated)
- if **-ve** then: One increases when the other decreases (Inversely correlated)

Now, that we know that the covariance matrix is not more than a table that summaries the correlations between all the possible pairs of variables. let's perform this step.



```
## covariance matrix calculation
cov_matrix = np.cov(data_scaled.T)
print('Covariance Matrix \n', cov_matrix)
```

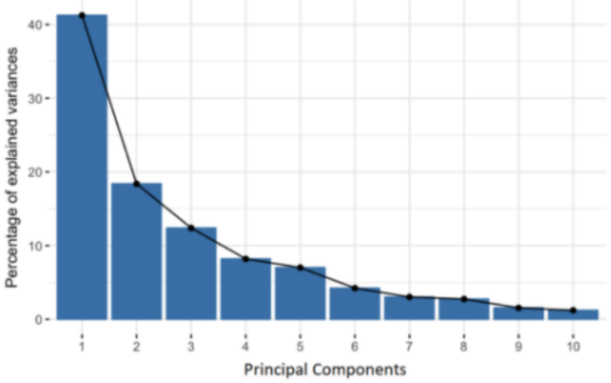
```
Covariance Matrix
 [[ 1.00132979e+00 -3.31582115e-02  1.06101321e-01  4.23506899e-01
   6.07723445e-01 -5.64225183e-02 -3.11300884e-02 -9.66326289e-03
  -9.87300540e-02  5.79406915e-02  1.36890778e-02 -6.03700052e-02
   4.05497226e-01]
 [-3.31582115e-02  1.00132979e+00 -1.20382861e-01 -3.46051004e-02
  -5.83924775e-02 -8.44837698e-02  8.89319009e-01 -1.63266264e-01
   2.70507312e-02 -2.34953587e-01 -1.60804404e-01  7.71793469e-02
   3.34460049e-01]
 [ 1.06101321e-01 -1.20382861e-01  1.00132979e+00  3.18801449e-01
   2.67930359e-01  7.90208661e-02 -1.33699059e-01  6.12767546e-01
   2.85315019e-01  4.35915402e-01  4.43046609e-01  7.22359587e-02
   6.63436673e-02]
 [ 4.23506899e-01 -3.46051004e-02  3.18801449e-01  1.00132979e+00
   6.52507648e-01 -5.99985405e-02 -3.18782667e-02  1.26391585e-01
   6.13711932e-02  9.04253051e-02  9.86077958e-02 -1.27896664e-04
   2.50913296e-01]
 [ 6.07723445e-01 -5.83924775e-02  2.67930359e-01  6.52507648e-01
   1.00132979e+00 -7.08913431e-02 -5.54725311e-02  1.07108799e-01
   1.93018022e-02  8.57115891e-02  1.02909091e-01  9.13632427e-03
   3.42011117e-01]
 [-5.64225183e-02 -8.44837698e-02  7.90208661e-02 -5.99985405e-02
  -7.08913431e-02  1.00132979e+00 -9.55138763e-02  1.07988079e-01
  -2.36334670e-01  5.34247037e-02  5.04123675e-02 -1.55426002e-01
  -9.94983848e-02]
 [-3.11300884e-02  8.89319009e-01 -1.33699059e-01 -3.18782667e-02
  -5.54725311e-02 -9.55138763e-02  1.00132979e+00 -1.95582291e-01
   1.97070726e-02 -2.27759226e-01 -1.35179749e-01  5.31644460e-02
   2.72272038e-01]
```

Step3: Compute Eigen vectors and Eigen values of the Cov. matrix to identify Principal Components

Background: Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the data.

Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components.

So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the scree plot below.



Organizing information in principal components this way, will allow you to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as your new variables.

An important thing to realize here is that, the principal components are less interpretable and don’t have any real meaning since they are constructed as linear combinations of the initial variables.

Principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data.

To put all this simply, just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible.

Eigen Vectors and Eigen Values: Now that we understood what we mean by principal components, let’s go back to eigenvectors and eigenvalues. What we firstly need to know about them is that **they always come in pairs, so that every eigenvector has an eigenvalue.** *Their number is equal to the number of dimensions of the data*. For example, for a 3-dimensional data set, there are 3 variables, therefore there are 3 eigenvectors with 3 corresponding eigenvalues.

Very Important: The eigenvectors of the cov. matrix is actually "**the directions of the axes where there is the most variance(most information) and that we call Principal Components**". The eigenvalues are simply "*the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component*".

By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance.

Example to understand eigen value and eigen vector in simple way:

Take 2-dimensional dataset with 2 variables *x, y* and that the eigenvectors and eigenvalues of the covariance matrix are as follows:

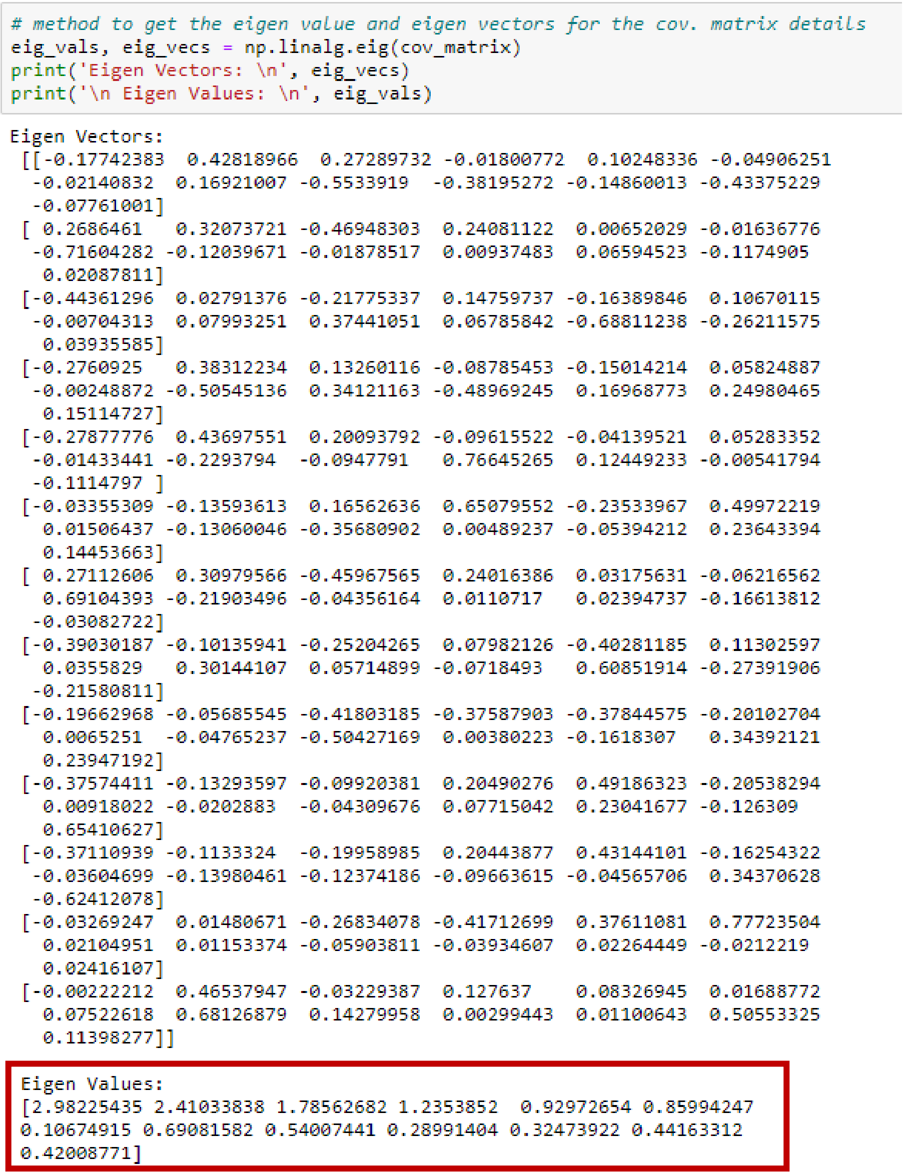
$$v1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix} \qquad \lambda_1 = 1.284028$$

$$v2 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix} \qquad \lambda_2 = 0.04908323$$

Principal Component Analysis Example, if we rank the eigenvalues in descending order, we get λ1>λ2, which means that the eigenvector that corresponds to the first principal component (PC1) is v1 and the one that corresponds to the second component (PC2) is v2.

After having the principal components, to compute the percentage of variance (information) accounted for by each component, we divide the eigenvalue of each component by the sum of eigenvalues. If we apply this on the example above, we find that PC1 and PC2 carry respectively 96% and 4% of the variance of the data.

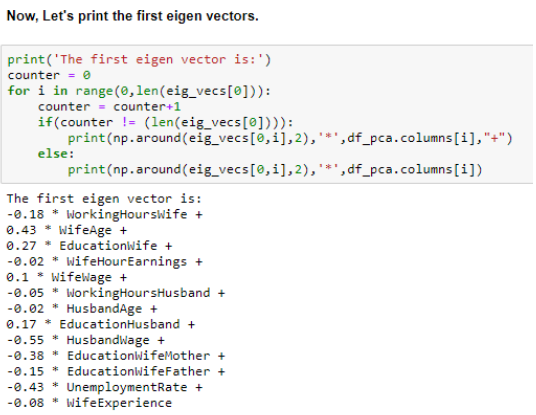
Now, let's put all the information what we understood in the theory above, apply the same in the given dataset.



```
# method to get the eigen value and eigen vectors for the cov. matrix details
eig_vals, eig_vecs = np.linalg.eig(cov_matrix)
print('Eigen Vectors: \n', eig_vecs)
print('\n Eigen Values: \n', eig_vals)
```

```
Eigen Vectors:
 [[-0.17742383  0.42818966  0.27289732 -0.01800772  0.10248336 -0.04906251
  -0.02140832  0.16921007 -0.5533919  -0.38195272 -0.14860013 -0.43375229
  -0.07761001]
 [ 0.2686461   0.32073721 -0.46948303  0.24081122  0.00652029 -0.01636776
  -0.71604282 -0.12039671 -0.01878517  0.00937483  0.06594523 -0.1174905
   0.02087811]
 [-0.44361296  0.02791376 -0.21775337  0.14759737 -0.16389846  0.10670115
  -0.00704313  0.07993251  0.37441051  0.06785842 -0.68811238 -0.26211575
   0.03935585]
 [-0.2760925   0.38312234  0.13260116 -0.08785453 -0.15014214  0.05824887
  -0.00248872 -0.50545136  0.34121163 -0.48969245  0.16968773  0.24980465
   0.15114727]
 [-0.27877776  0.43697551  0.20093792 -0.09615522 -0.04139521  0.05283352
  -0.01433441 -0.2293794  -0.0947791   0.76645265  0.12449233 -0.00541794
  -0.1114797 ]
 [-0.03355309 -0.13593613  0.16562636  0.65079552 -0.23533967  0.49972219
   0.01506437 -0.13060046 -0.35680902  0.00489237 -0.05394212  0.23643394
   0.14453663]
 [ 0.27112606  0.30979566 -0.45967565  0.24016386  0.03175631 -0.06216562
   0.69104393 -0.21903496 -0.04356164  0.0110717   0.02394737 -0.16613812
  -0.03082722]
 [-0.39030187 -0.10135941 -0.25204265  0.07982126 -0.40281185  0.11302597
   0.0355829   0.30144107  0.05714899 -0.0718493   0.60851914 -0.27391906
  -0.21580811]
 [-0.19662968 -0.05685545 -0.41803185 -0.37587903 -0.37844575 -0.20102704
   0.0065251  -0.04765237 -0.50427169  0.00380223 -0.1618307   0.34392121
   0.23947192]
 [-0.37574411 -0.13293597 -0.09920381  0.20490276  0.49186323 -0.20538294
   0.00918022 -0.0202883  -0.04309676  0.07715042  0.23041677 -0.126309
   0.65410627]
 [-0.37110939 -0.1133324  -0.19958985  0.20443877  0.43144101 -0.16254322
  -0.03604699 -0.13980461 -0.12374186 -0.09663615 -0.04565706  0.34370628
  -0.62412078]
 [-0.03269247  0.01480671 -0.26834078 -0.41712699  0.37611081  0.77723504
   0.02104951  0.01153374 -0.05903811 -0.03934607  0.02264449 -0.0212219
   0.02416107]
 [-0.00222212  0.46537947 -0.03229387  0.127637    0.08326945  0.01688772
   0.07522618  0.68126879  0.14279958  0.00299443  0.01100643  0.50553325
   0.11398277]]

 Eigen Values:
 [2.98225435 2.41033838 1.78562682 1.2353852  0.92972654 0.85994247
 0.10674915 0.69081582 0.54007441 0.28991404 0.32473922 0.44163312
 0.42008771]
```

Based on this let’s print the eigen vectors formula:

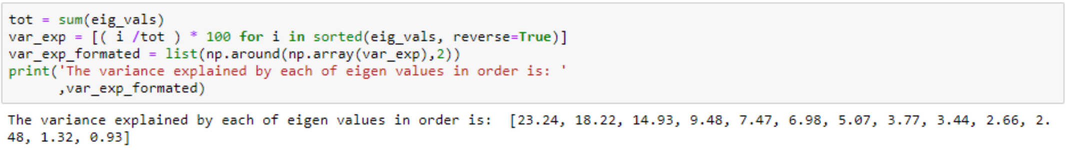


Now, Let's print the first eigen vectors.

```
print('The first eigen vector is:')
counter = 0
for i in range(0,len(eig_vecs[0])):
    counter = counter+1
    if(counter != (len(eig_vecs[0]))):
        print(np.around(eig_vecs[0,i],2),'*',df_pca.columns[i],"+")
    else:
        print(np.around(eig_vecs[0,i],2),'*',df_pca.columns[i])
```

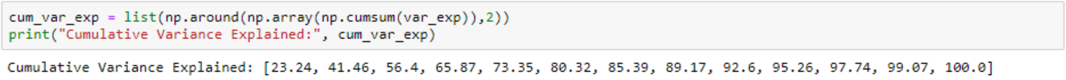
```
The first eigen vector is:
-0.18 * WorkingHoursWife +
0.43 * WifeAge +
0.27 * EducationWife +
-0.02 * WifeHourEarnings +
0.1 * WifeWage +
-0.05 * WorkingHoursHusband +
-0.02 * HusbandAge +
0.17 * EducationHusband +
-0.55 * HusbandWage +
-0.38 * EducationWifeMother +
-0.15 * EducationWifeFather +
-0.43 * UnemploymentRate +
-0.08 * WifeExperience
```

Now, Let's calculate the variance explained by eigen values and the cumulative variance by the eigen values:



```
tot = sum(eig_vals)
var_exp = [( i /tot ) * 100 for i in sorted(eig_vals, reverse=True)]
var_exp_formated = list(np.around(np.array(var_exp),2))
print('The variance explained by each of eigen values in order is: '
      ,var_exp_formated)
```

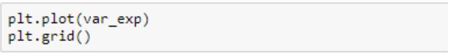
```
The variance explained by each of eigen values in order is:  [23.24, 18.22, 14.93, 9.48, 7.47, 6.98, 5.07, 3.77, 3.44, 2.66, 2.
48, 1.32, 0.93]
```



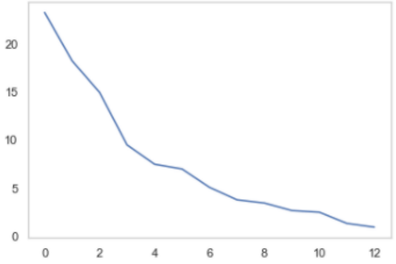
```
cum_var_exp = list(np.around(np.array(np.cumsum(var_exp)),2))
print("Cumulative Variance Explained:", cum_var_exp)
```

```
Cumulative Variance Explained: [23.24, 41.46, 56.4, 65.87, 73.35, 80.32, 85.39, 89.17, 92.6, 95.26, 97.74, 99.07, 100.0]
```

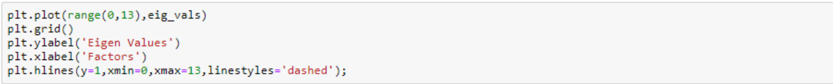
Let's plot the variance explained by each eigen value with the eigen value



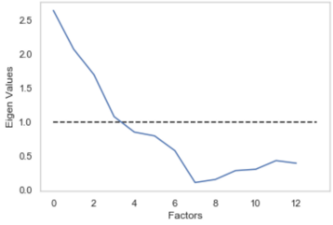
```
plt.plot(var_exp)
plt.grid()
```

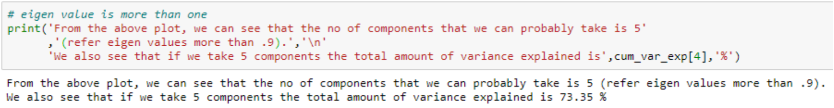


Next step is to plot of eigen values with the number of factors or Principal Components



```
plt.plot(range(0,13),eig_vals)
plt.grid()
plt.ylabel('Eigen Values')
plt.xlabel('Factors')
plt.hlines(y=1,xmin=0,xmax=13,linestyles='dashed');
```

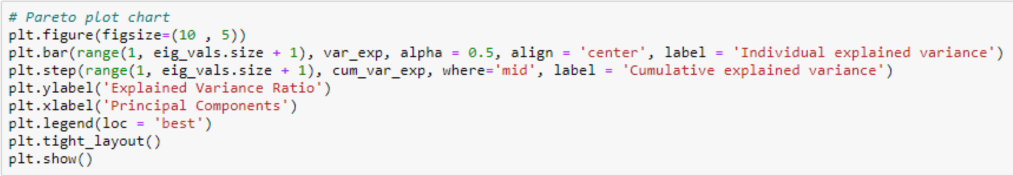




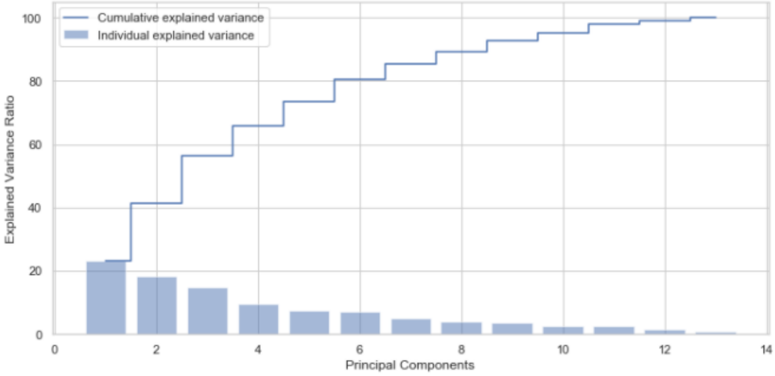
```
# eigen value is more than one
print('From the above plot, we can see that the no of components that we can probably take is 5'
      ,'(refer eigen values more than .9).','\n'
      'We also see that if we take 5 components the total amount of variance explained is',cum_var_exp[4],'%')
```

```
From the above plot, we can see that the no of components that we can probably take is 5 (refer eigen values more than .9).
We also see that if we take 5 components the total amount of variance explained is 73.35 %
```

Let's plot one more graph with both the variance explained by each eigen value and the cumulative variance explained.



```
# Pareto plot chart
plt.figure(figsize=(10 , 5))
plt.bar(range(1, eig_vals.size + 1), var_exp, alpha = 0.5, align = 'center', label = 'Individual explained variance')
plt.step(range(1, eig_vals.size + 1), cum_var_exp, where='mid', label = 'Cumulative explained variance')
plt.ylabel('Explained Variance Ratio')
plt.xlabel('Principal Components')
plt.legend(loc = 'best')
plt.tight_layout()
plt.show()
```

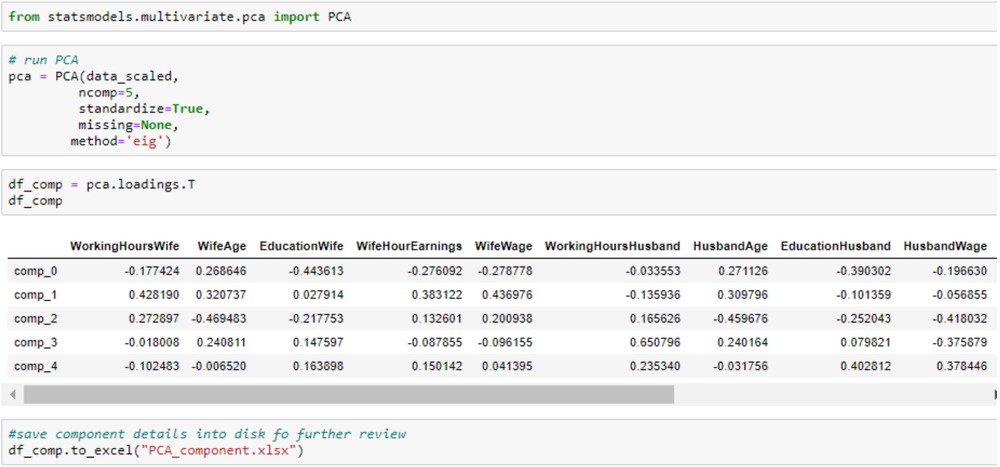


Step4: Calculate Principal Components

Background: In the previous steps, apart from standardization, we do not make any changes on the dataset, we have just selected the principal components and form the feature vector, but the input dataset remains always in terms of the original axes (i.e., in terms of the initial variables).

In this step, which is the last one, the aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components.(hence the name Principal Components Analysis).

Below is the most used method to find principal components.



```
from statsmodels.multivariate.pca import PCA
```

```
# run PCA
pca = PCA(data_scaled,
         ncomp=5,
         standardize=True,
         missing=None,
        method='eig')
```

```
df_comp = pca.loadings.T
df_comp
```

	WorkingHoursWife	WifeAge	EducationWife	WifeHourEarnings	WifeWage	WorkingHoursHusband	HusbandAge	EducationHusband	HusbandWage
comp_0	-0.177424	0.268646	-0.443613	-0.276092	-0.278778	-0.033553	0.271126	-0.390302	-0.196630
comp_1	0.428190	0.320737	0.027914	0.383122	0.436976	-0.135936	0.309796	-0.101359	-0.056855
comp_2	0.272897	-0.469483	-0.217753	0.132601	0.200938	0.165626	-0.459676	-0.252043	-0.418032
comp_3	-0.018008	0.240811	0.147597	-0.087855	-0.096155	0.650796	0.240164	0.079821	-0.375879
comp_4	-0.102483	-0.006520	0.163898	0.150142	0.041395	0.235340	-0.031756	0.402812	0.378446

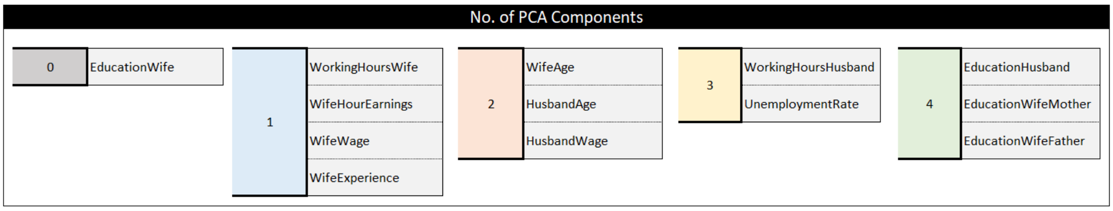
```
#save component details into disk fo further review
df_comp.to_excel("PCA_component.xlsx")
```

Final Summary(snapshot) on problem 2.4:

Based on the **eigen values** greater than **0.9**, there are **5 components** which are considered for the PCA. Further, these 5 components are explaining **~72%** of the total variances.

In nutshell, we have reduced dimension from **13 variables to 5 components**.

Further, the grouping of the components based on the loading values are mentioned below:



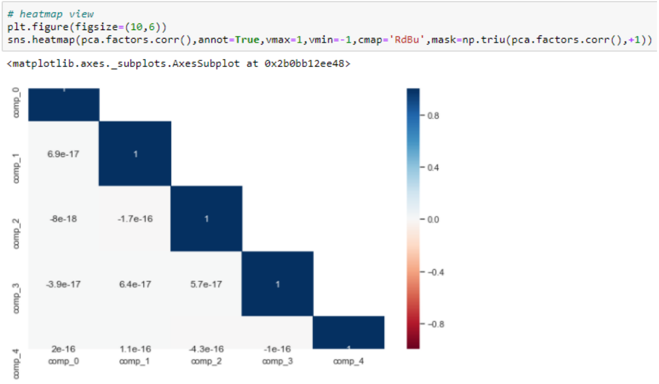
No. of PCA Components

Component	Variables
0	EducationWife
1	WorkingHoursWife WifeHourEarnings WifeWage WifeExperience
2	WifeAge HusbandAge HusbandWage
3	WorkingHoursHusband UnemploymentRate
4	EducationHusband EducationWifeMother EducationWifeFather

2.5 Perform Multiple Linear Regression with 'FamilyIncome' as the dependent variable and the Principal Components extracted as the independent variables.

Now, as we have completed dimension reduction through PCA and found **5 components**. Let's consider these **5 components as an independent variable** and **"FamilyIncome" as dependent variable** to perform multiple linear regression (MLR).

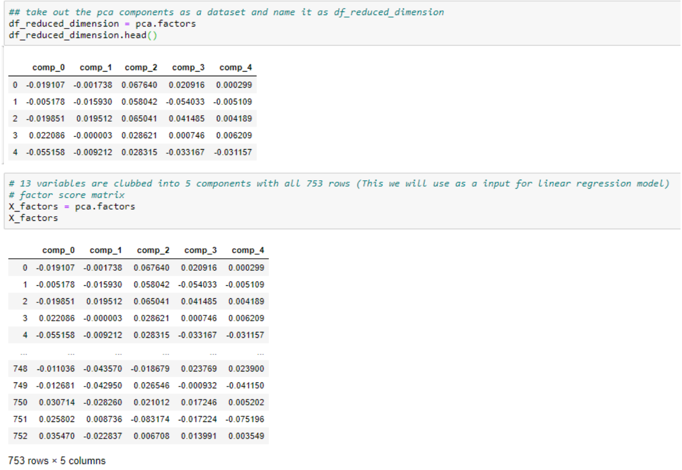
- **Before we proceed to MLR calculation let's see heat map view:**



```
# heatmap view
plt.figure(figsize=(10,6))
sns.heatmap(pca.factors.corr(),annot=True,vmax=1,vmin=-1,cmap='RdBu',mask=np.triu(pca.factors.corr(),+1))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2b0bb12ee48>
```

- **Now, we will use these 5 components in the dataset and see the MLR result:**



```
## take out the pca components as a dataset and name it as df_reduced_dimension
df_reduced_dimension = pca.factors
df_reduced_dimension.head()
```

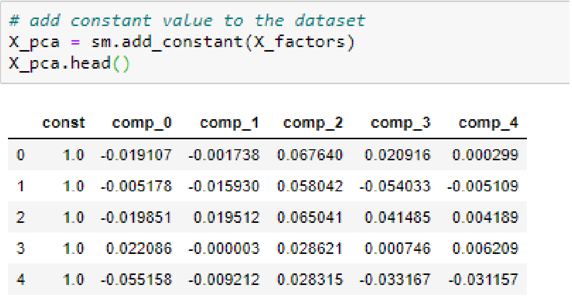
	comp_0	comp_1	comp_2	comp_3	comp_4
0	-0.019107	-0.001738	0.067640	0.020916	0.000299
1	-0.005178	-0.015930	0.058042	-0.054033	-0.005109
2	-0.019851	0.019512	0.065041	0.041485	0.004189
3	0.022086	-0.000003	0.028621	0.000746	0.006209
4	-0.055158	-0.009212	0.028315	-0.033167	-0.031157

```
# 13 variables are clubbed into 5 components with all 753 rows (This we will use as a input for linear regression model)
# factor score matrix
X_factors = pca.factors
X_factors
```

	comp_0	comp_1	comp_2	comp_3	comp_4
0	-0.019107	-0.001738	0.067640	0.020916	0.000299
1	-0.005178	-0.015930	0.058042	-0.054033	-0.005109
2	-0.019851	0.019512	0.065041	0.041485	0.004189
3	0.022086	-0.000003	0.028621	0.000746	0.006209
4	-0.055158	-0.009212	0.028315	-0.033167	-0.031157
...	...	...	...	...	...
748	-0.011036	-0.043570	-0.018679	0.023769	0.023900
749	-0.012681	-0.042950	0.026546	-0.000932	-0.041150
750	0.030714	-0.028260	0.021012	0.017246	0.005202
751	0.025802	0.008736	-0.083174	-0.017224	-0.075196
752	0.035470	-0.022837	0.006708	0.013991	0.003549

753 rows × 5 columns

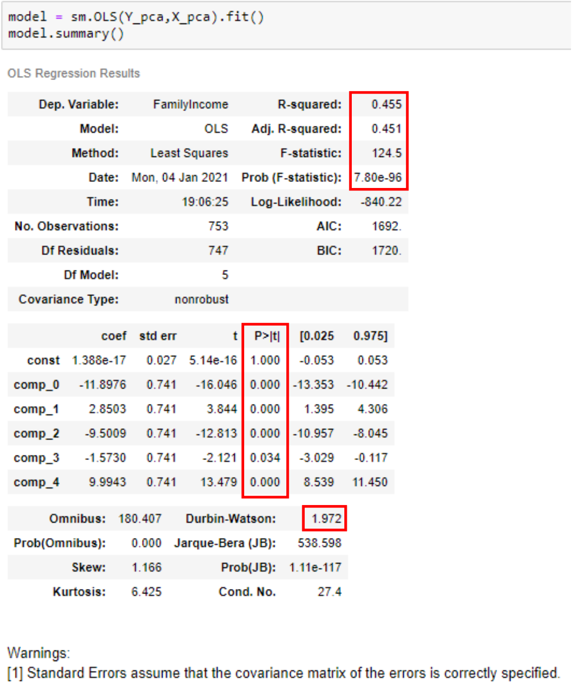
- **Add constant value to the dataset:**



```python
# add constant value to the dataset
X_pca = sm.add_constant(X_factors)
X_pca.head()
```

	const	comp_0	comp_1	comp_2	comp_3	comp_4
0	1.0	-0.019107	-0.001738	0.067640	0.020916	0.000299
1	1.0	-0.005178	-0.015930	0.058042	-0.054033	-0.005109
2	1.0	-0.019851	0.019512	0.065041	0.041485	0.004189
3	1.0	0.022086	-0.000003	0.028621	0.000746	0.006209
4	1.0	-0.055158	-0.009212	0.028315	-0.033167	-0.031157

- **Run the model:**



```python
model = sm.OLS(Y_pca,X_pca).fit()
model.summary()
```

OLS Regression Results

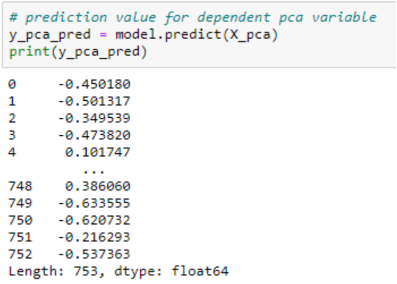
Dep. Variable:	FamilyIncome	R-squared:	0.455
Model:	OLS	Adj. R-squared:	0.451
Method:	Least Squares	F-statistic:	124.5
Date:	Mon, 04 Jan 2021	Prob (F-statistic):	7.80e-96
Time:	19:06:25	Log-Likelihood:	-840.22
No. Observations:	753	AIC:	1692.
Df Residuals:	747	BIC:	1720.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.388e-17	0.027	5.14e-16	1.000	-0.053	0.053
comp_0	-11.8976	0.741	-16.046	0.000	-13.353	-10.442
comp_1	2.8503	0.741	3.844	0.000	1.395	4.306
comp_2	-9.5009	0.741	-12.813	0.000	-10.957	-8.045
comp_3	-1.5730	0.741	-2.121	0.034	-3.029	-0.117
comp_4	9.9943	0.741	13.479	0.000	8.539	11.450

Omnibus:	180.407	Durbin-Watson:	1.972
Prob(Omnibus):	0.000	Jarque-Bera (JB):	538.598
Skew:	1.166	Prob(JB):	1.11e-117
Kurtosis:	6.425	Cond. No.	27.4

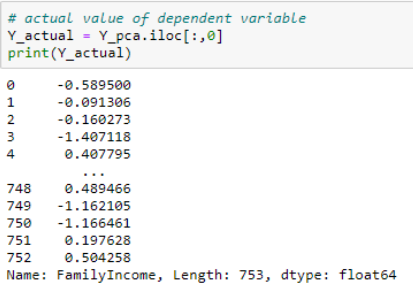
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

- **Predict y value and calculate error/residual:**



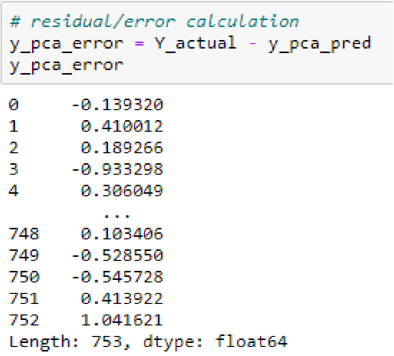
```python
# prediction value for dependent pca variable
y_pca_pred = model.predict(X_pca)
print(y_pca_pred)
```

```
0     -0.450180
1     -0.501317
2     -0.349539
3     -0.473820
4      0.101747
         ...
748    0.386060
749   -0.633555
750   -0.620732
751   -0.216293
752   -0.537363
Length: 753, dtype: float64
```



```python
# actual value of dependent variable
Y_actual = Y_pca.iloc[:,0]
print(Y_actual)
```

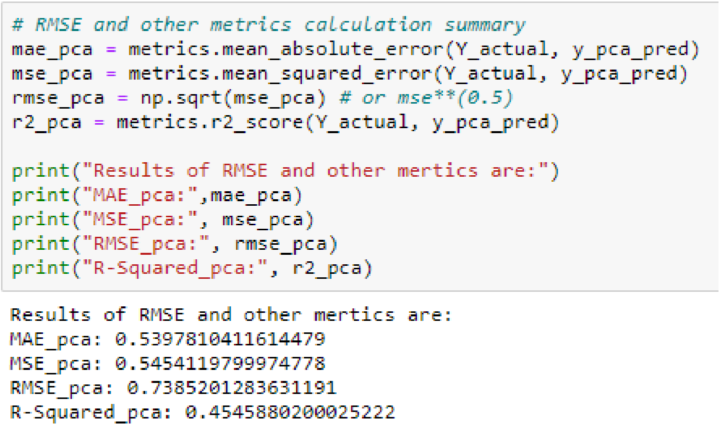
```
0     -0.589500
1     -0.091306
2     -0.160273
3     -1.407118
4      0.407795
         ...
748    0.489466
749   -1.162105
750   -1.166461
751    0.197628
752    0.504258
Name: FamilyIncome, Length: 753, dtype: float64
```



```python
# residual/error calculation
y_pca_error = Y_actual - y_pca_pred
y_pca_error
```

```
0     -0.139320
1      0.410012
2      0.189266
3     -0.933298
4      0.306049
         ...
748    0.103406
749   -0.528550
750   -0.545728
751    0.413922
752    1.041621
Length: 753, dtype: float64
```

- **RMSE and other metrics calculation:**



```
# RMSE and other metrics calculation summary
mae_pca = metrics.mean_absolute_error(Y_actual, y_pca_pred)
mse_pca = metrics.mean_squared_error(Y_actual, y_pca_pred)
rmse_pca = np.sqrt(mse_pca) # or mse**(0.5)
r2_pca = metrics.r2_score(Y_actual, y_pca_pred)

print("Results of RMSE and other mertics are:")
print("MAE_pca:",mae_pca)
print("MSE_pca:", mse_pca)
print("RMSE_pca:", rmse_pca)
print("R-Squared_pca:", r2_pca)
```

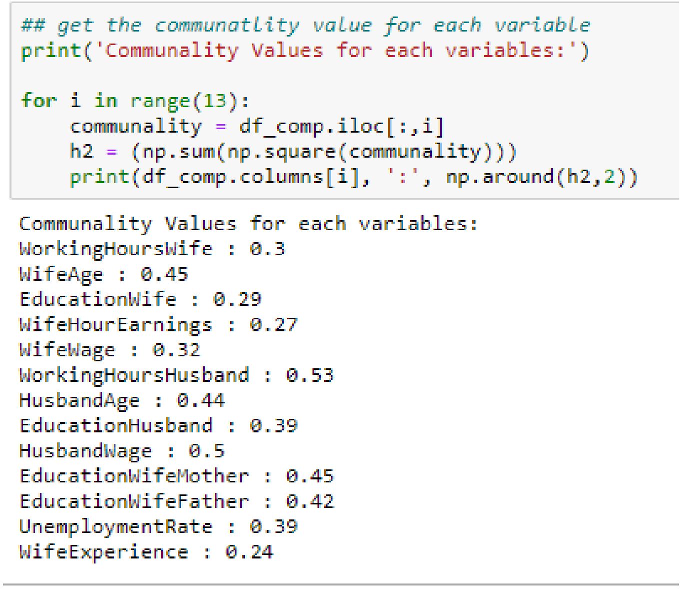
```
Results of RMSE and other mertics are:
MAE_pca: 0.5397810411614479
MSE_pca: 0.5454119799974778
RMSE_pca: 0.7385201283631191
R-Squared_pca: 0.4545880200025222
```

- **Communality Score(h^2):**

Background: In PCA and Factor Analysis, a variable's communality is a useful measure **for predicting the variable's value.** Communality may be denoted as h^2. More specifically, *it tells you what proportion of the variable's variance is a result of either:*

  - The principal components or,
  - The correlations between each variable and individual factors.

A variable's communality ranges from 0 to 1.In general, one way to think of communality is as the proportion of common variance found in a particular variable.



```
## get the communatlity value for each variable
print('Communality Values for each variables:')

for i in range(13):
    communality = df_comp.iloc[:,i]
    h2 = (np.sum(np.square(communality)))
    print(df_comp.columns[i], ':', np.around(h2,2))
```

```
Communality Values for each variables:
WorkingHoursWife : 0.3
WifeAge : 0.45
EducationWife : 0.29
WifeHourEarnings : 0.27
WifeWage : 0.32
WorkingHoursHusband : 0.53
HusbandAge : 0.44
EducationHusband : 0.39
HusbandWage : 0.5
EducationWifeMother : 0.45
EducationWifeFather : 0.42
UnemploymentRate : 0.39
WifeExperience : 0.24
```

2.6 Comment on the Model thus built using the Principal Components and with 'FamilyIncome'.

Final Summary(snapshot) based on problem 2.5:

To run MLR model after PCA, these are the things have been taken into considerations. I have taken no. components identified during problem2.4 **(i.e., 5) as an independent variable.** Took **"FamilyIncome"** as a dependent variable where *performed data standardisation (using z-score) so that independent and dependent variables will be on a same scale* before running the model.

There are **753 rows** in the dataset. *(replaced the outliers with 3 and -3, where zscore value more than 3 and less than -3 respectively)*

After running the model mentioned below are the **observations** sighted:

R^2 value has **drastically reduced** from **73.5%** *(model before PCA)* to **45.5%** *(model after PCA)* which means *before PCA due multicollinearity we had inflated R2 value.*

RMSE score has **increased** from **0.465** *(model before PCA)* to **0.739** *(model after PCA)* which means *after PCA, predicted value and observed value having more residuals.*

Prob (F-statistic) value is **7.80e-96** which is *less than 0.5* means **the model is still significant.**

Durbin-Watson value is **~2(1.972)** which is in line with the benchmark value (i.e., 2 to 2.5) means ***no auto correlation between errors***.

All 5 components have p-value less than (α) 0.05. *It means they are significant and all the components are adding significance to the model.*

Out of 5 components, **3 components** having **-ve coefficient relation**. Which means *if component value increases then the dependent variable decreases*.

Communality (h^2) value for each variable is **very minimal**. *Highest Communality (h2) value is 0.45 (WifeAge) which is still very less.*

Conclusion: After applying all the concepts which i learned in Advanced Statistics class, I come to the conclusion that **this model is WEAK**. We may need to use **other Data Mining techniques to improve the accuracy of the model.**

2.7 Mention the business implication and interpretation of the models.

In this case study, I have performed multiple linear regression (MLR) and assessed the model and found the result is not significant. Hence, went for principal component analysis (PCA) to do the dimension reduction and improve the significance of the independent variables in the dataset. After PCA, I reran the MLR model and found that though the components are significant in the dataset but still R2 value has reduced drastically. Hence, I have concluded the model is WEAK and we should not procced to use it in the production environment. Now the question is

I. if I will not do any further changes in the model and deploy the same in the production environment what will be the implications I would face and,
II. if we want to improve the model accuracy (i.e., improve R2 value) what are the changes I need to be do.

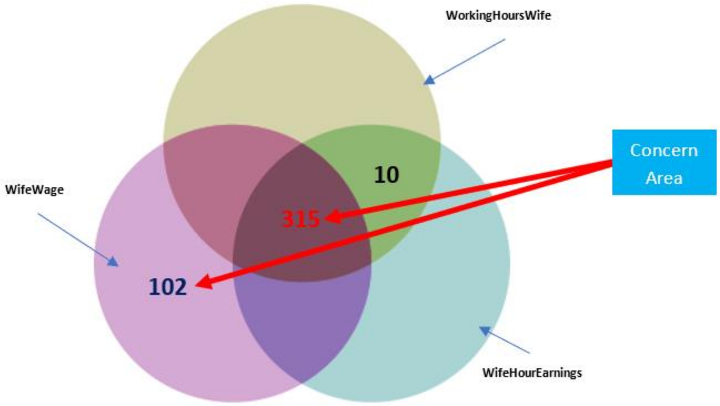
Let’s see both the above scenarios in summary:

Key Implications:

- As model **accuracy is less** predict the “FamilyIncome” as after using PCA, model shows that only 45 percent of the variability can be explained by the independent variables. If we go back and eliminate the PCA part and use the first iteration MLR still, model shows that only 73 percent of the variability can be explained by the independent variables. It will also impact the “**End user performance**” where based on “FamilyIncome” they are planning to cast their advertisement, insurance policy categorisation, family segmentation etc...
- It will impact the “**Time and Cost perspective**” as non-significant variables are getting captured. *(we might need to pay third party to get such variables to perform this research)*
- It will impact on the “Company image” who is doing research (if it is third party) and their agreement with the end users will get impacted.

Key Changes to improve the accuracy of the model:

- In the existing dataset, sighted 3 columns “**WorkingHoursWife, WifeHourEarnings & WifeWage” in which “ZERO” values** found in 325, 325 and 417 rows respectively. Further, **~40% (315 out of 753) of the dataset**, all these three columns have value as “ZERO”. As, these columns are significant we need to relook the samples and ensure all the variables must have data relatively. *(refer below graph)*



- Add “**new variables**” in the dataset which are significant to the dependent variable (i.e., family income) and “**drop non-significant variables**”. There are certain independent variables like “EducationWifeMother, EducationWifeFather” are not making significance impact. Hence, excluding such variables and adding relevant variables into the dataset will improve the accuracy of the model.
- Reperform the all 5 assumptions and assess the accuracy of the model and based on result of the model perform PCA or other data mining techniques to increase the accuracy of the model and then deploy the same in the production environment.