Modelling techniques for Timeseries data, Logistic regression and Principal Component analysis

Rohan Narayan Koli
MSc. Data Analytics
19224842

***Abstract*—This research paper encapsulates three different modelling techniques related to time series analysis, logistic regression and principal component analysis. For time-series analysis, we incorporated ARIMA as the best fit model to forecast Denmark's supply of electricity. Next, a person's life satisfaction is predicted using a survey dataset using logistic regression. Lastly, in another survey dataset, using principal component analysis we reduced the number of variables from thirty to six principal components with sixty-two percent variability explained.**

***Index Terms*—Holt-Winter's, ARIMA, Logistic regression, Binary regression, PCA analysis**

I. Introduction

Now-a-days forecasting data points in the future has become a de facto process to optimize the future resource utilization. It allows us to be prepared for the future events or plan to mitigate the effects of over resource utilization. In the first part, we analyze Denmark's electricity supply trends and analyze various timeseries models for forecasting the future values.

Data accumulated from surveys and records from medical facilities can be successfully understood or analyzed using Binary logistic regression where linear regression fails. In the second part, we analyze a persons satisfaction in life, by using logistic regression on various related variables.

In the third part, we use Principal Component Analysis (PCA) to analyze a dataset with 30 variables and simplify or converge into 6 principal components which would accommodate most of the variability in the components. Lastly, we try to make reasonable judgements on the labels of the identified principal components.

II. Time Series Analysis

Any variable which is measured in a sequential fashion over a series of fixed intervals (sampling intervals) results in the formation of a *time series* [1]. When a time series is dissected, we get two basic components:

1) Systematic components: Level, Trend, Seasonality and cyclic variations
2) Non-Systematic component: Noise

A long term increase or decrease in the data symbolizes a trend. Seasonality is the level of time series affected by seasonal factors along a known and fixed frequency. Cyclical variations are periodic oscillations which typically occur with a period greater than a year and which are not seasonal [2]. Random variations or fluctuations from causes such as measurement errors and other unknown causes constitutes to noise.

A. Dataset and Initial Observations

For our analysis, we are selecting the monthly dataset of the total net electricity supplied in Denmark, from January 2008 to December 2019. Below is the link to the dataset: https://ec.europa.eu/eurostat/web/products-datasets/-/nrg$_1$05m

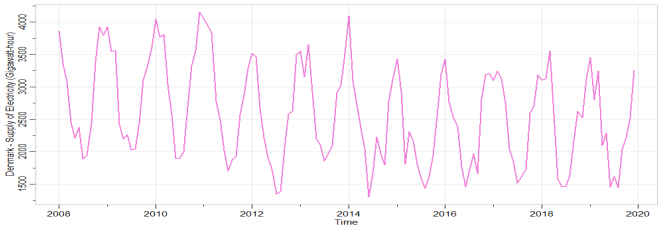

Fig. 1. Denmark - Monthly Electricity Supply

Fig.1 displays the line chart of the time series. Since our dataset has roughly the same size of peaks and troughs, we can safely assume our data to be an additive time-series and therefore we use the "stl" function in R to decompose the time-series. The stl function uses Loess smoothing method for decomposition and it has many advantages over classical and x11 decomposition methods such as handling different types of seasonality and robustness to outliers [3].

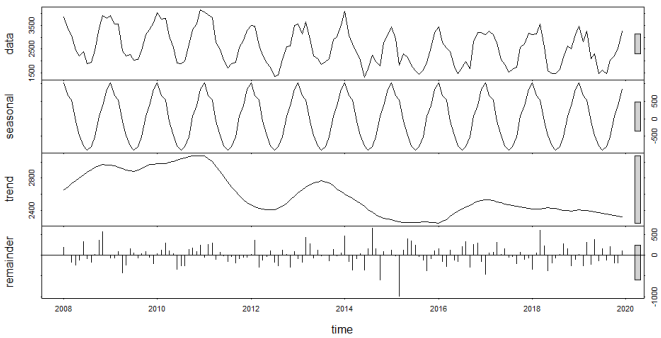

Fig. 2. Timeseries Decomposition

On decomposition using stl function from fig.2, we get the three components trend, seasonality and the remainder (residuals). Here, we observe a downward trend and existence of the seasonal component along with a random plot of residuals.

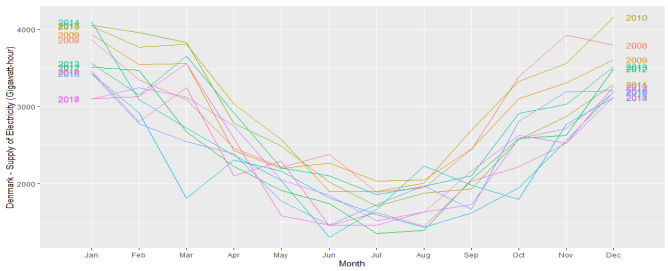

Fig. 3. Seasonal Plot of Timeseries

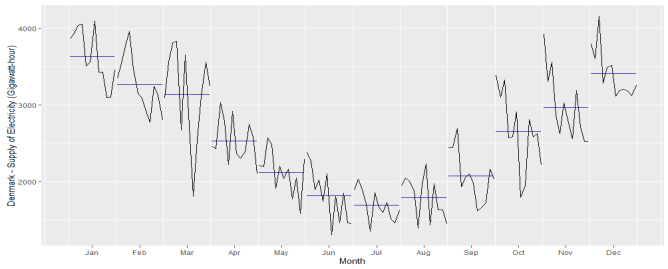

Fig. 4. Subseries Plot of Timeseries

Using the "ggplot2" library in R, we plot the seasonal plots and the subseries plots as observed in fig.3 and fig.4 respectively. The horizontal lines in fig.4 represent the means for each month. We notice, the supply of electricity gradually drops in the months of June, July and August whereas it peaks in the months of January and December.

B. Modelling and Forecasting

To evaluate the models, we will split the data into train and test using the "window" function in base R. The train dataset contains the first 140 observations whereas the test dataset contains the last 4 observations to compare the forecasts.

1) Seasonal Naive Model: The Seasonal Naive model forecasts the value from the most recent identical season. For our timeseries, to predict the value of Denmark's Supply of Electricity for the months of September 2019 to December 2019, the model assumes the values from the months of September 2018 to December 2018 respectively. Below is the general form of the Seasonal Naive model:

$$F_{t+k} = y_{t-M+k}$$

where, M is the season and k represents the number of naive steps ahead [4]. Naive models act as a baseline to evaluate the performance of other models. We use the "snaive" function in R to forecast for four periods as observed in fig.5.

2) Holt-Winters Model: Holt-Winters forecast is a type of forecast obtained using exponential smoothing of the past observations and assigning decaying exponential weights (Recent observations are given more weight) [2]. This model comprises of three smoothing equations for level(l_t), trend(b_t) and seasonal(s_t) components with corresponding smoothing parameters α,β and γ. Since the seasonal component of our dataset are roughly constant, we use the "additive" variation. The general form is given as below:

$$\hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)}$$

where,

$$l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$$

We use "hw" function and set seasonal factor as "additive" in R to use the holt-winters forecast method.

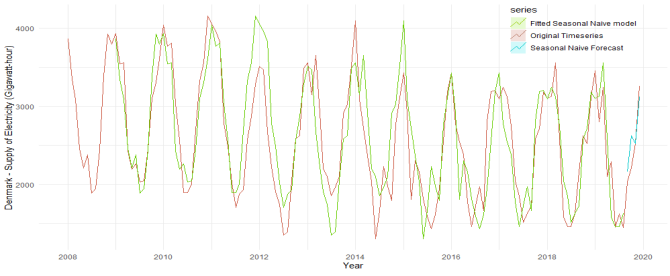

Fig. 5. Seasonal Naive Forecast

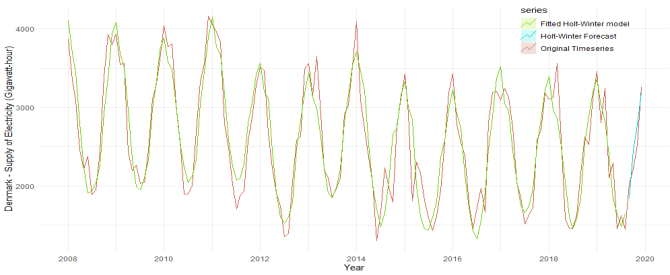

Fig. 6. Holt-Winter's Forecast

3) Using ets() function: The "ets()" function allows us to estimate the parameters by maximizing the likelihood that maximizing the probability of data from the model using Akaike's Information Criterion(AIC) which is defined as-

$$AIC = -2log(L) + 2k$$

where L is the model likelihood and k is the total number of parameters [5]. The ets() function auto-estimates α,βandγ when "ZZZ" is specified in the parameters by minimizing the AIC [6]. For our timeseries the ets() function estimated "ANA" as the best fit model and we get a forecast as in fig.7.

4) Seasonal ARIMA Model (p,d,q) [P,D,Q]: Before modelling a Seasonal ARIMA Model, we check if the timeseries in focus is *stationary*, that is the mean, variance and autocorrelations are all constant over time. An Autoregressive Integrated Moving Average (ARIMA) (p,d,q) integrates three major modelling techniques viz:

- Auto-regressive (AR) (p): Modelling according to the auto-correlation of the series values.

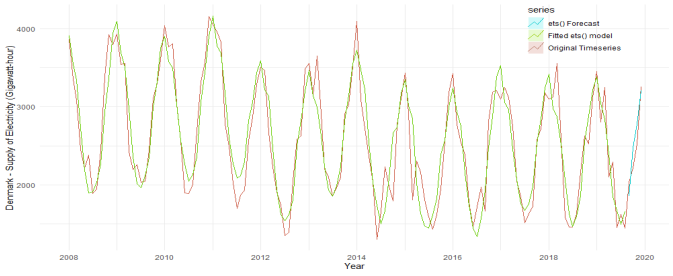

Fig. 7. Ets() Forecast

- Moving Average (MA) (q): Modelling according to the auto-correlation of the forecast errors.
- Integrated (I) (d): Integration refers to the order of differencing in order to remove trend time series "stationary".

Similar to non-seasonal ARIMA, a seasonal ARIMA has components ('P' and 'Q') backshifted to the seasonal periods [2]. To make the timeseries seasonally stationary, we need to difference (D) the time series by subtracting the Y(t)-Y(t-k), where k refers to the seasonal period.

For the timeseries in focus, to make the timeseries stationary, we use 'ndiff()' and 'nsdiff()' in R to compute the number of differencing required for the non-seasonal and seasonal components respectively. We get a value of 1 for both the functions suggesting a differencing of order 1 and back-shifting a period of 12 months which is achieved using "diff(diff(timesries),12)" in R. Finally, to check for stationarity of the timeseries, we use the Augmented Dickey-Fuller (ADF) test which tests the null hypothesis for a presence of unit root in timeseries and an alternative hypothesis of timeseries suggesting the presence of no unit root, i.e timeseries being stationary [7]. Using adf.test() on our differenced timeseries in R, we get a p-value of less than 0.01 and therefore we reject the null hypothesis and accept the alternate hypothesis of timeseries being stationary.

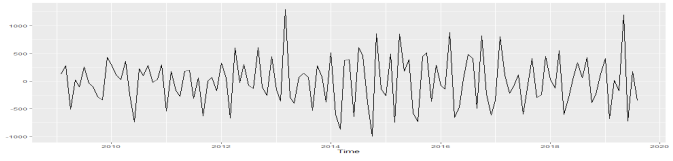

Fig. 8. Stationary Timeseries after adjusting for trend and seasonality

The visual patterns in autocorrelations (ACF) and partial autocorrelation (PACF) help us determine the values of p,P,q and Q. ACF plots measure the relationship between y_t and $y_{(t-k)}$.The PACF measures the relationship between y_t and $y_{(t-k)}$ after removing the lag effects at k=1,2,3,..,k-1. On inspecting the fig.9, initially we observe significant negative spikes at lags 1 and 2 in PACF and ACF plots and gradually decaying to 0, suggesting values for p=0 and q=1 or 2 that is inclusion of a moving average(MA) term [8]. Next, we see a significant spike at lag 12 in ACF plot and gradual decay towards 0 suggesting a seasonal MA term. Collating the above findings, we can estimate the optimum model

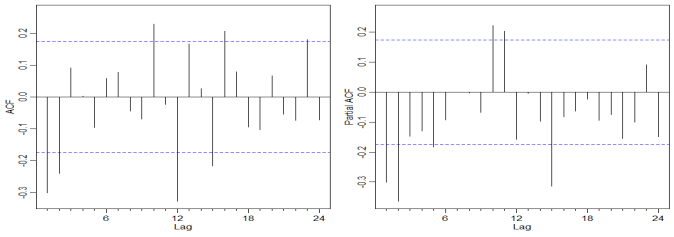

Fig. 9. ACF and PACF plots

to be ARIMA(0,1,1)(0,1,1)$_{12}$ or ARIMA(0,1,2)(0,1,1)$_{12}$. We select the model ARIMA(0,1,2)(0,1,1)$_{12}$ as we get a better AICc of 1834 compared to AICc of 1837 for the model ARIMA(0,1,1)(0,1,1)$_{12}$.

Using the 'auto.arima' function in R from the "fpp2" package and setting parameters "stepwise" and "approximation" as "false" for better estimation of parameters(p,d,q)(P,D,Q) in R, we get a model ARIMA(1,0,1)(0,1,1)$_{12}$ which has an AICc of 1849.

ARIMA Model Diagnostic and Forecasting: Firstly, the residual plot in fig.10 appears to have no trend over time and appears to be random in nature. Next, the ACF plot of the residuals show no significant autocorrelation between the residuals following a normal distribution and they are within the threshold limits indicating that the residuals are *white noise*. Using the "Box-Ljung" test on the residuals, we get a p-value of 0.81 (not significant); hence we accept the null hypothesis that the residuals are independently distributed and no autocorrelation exists between them. Therefore, ARIMA(0,1,2)(0,1,1)$_{12}$ model appears to fit the data well and the same is depicted in Fig.11.

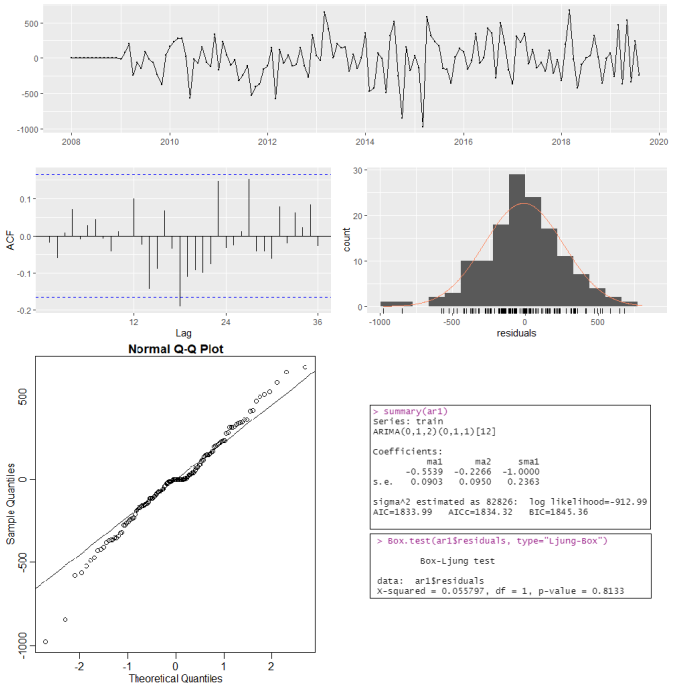

Fig. 10. ARIMA(0,1,2)(0,1,1)[12] Residual Plots

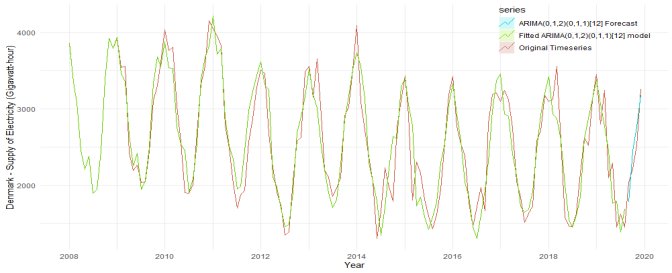

Fig. 11. ARIMA(0,1,2)(0,1,1)[12] forecast

C. Model Comparison and Selection

Using the "accuracy()" function in R, and comparing the forecasts with the training dataset, we get the root mean square errors(RMSE) for each models as shown in fig.12. On comparison, we find ARIMA(0,1,2)$(0,1,1)_{12}$ to be the most optimum model for forecasting Denmark's monthly supply of electricity with an AICc of 1834, RMSE with training dataset of 271 and RMSE of 216 after comparing with the test dataset(forecasting four periods ahead for the months of September to December 2020).

	Model	AICc	RMSE_train	RMSE_test
1	ARIMA(012)(011)[12]	1834.32	270.85	216.43
2	Auto-ARIMA	1849.02	287.91	208.98
3	Ets Model	2301.49	278.05	205.01
4	Holt-Winter	2308.97	280.38	202.51
5	Seasonal Naive	-	416.53	224.92

Fig. 12. Comparison of Timeseries Models

III. Logistic Regression

In linear regression model, there is a linear relationship between the dependent and independent variables. But when the dependent variable is dichotomous, the model fails to generate a sensible relationship between the outcome variable and predictors. The predicted value of the dependent variable is continuous (not probabilistic) and the model is very sensitive to outliers (it may generate probability values less than zero or greater than one). To address this problem, we use logarithmic transformation (logit) to express non-linear relationships in a linear manner [9] allowing us to predict dichotomous variable. The logistic regression with dependent variable (Y) and independent variables X_1,X_2,X_3...X_n can be expressed as-

$$P(Y) = \frac{1}{1 + e^{-(\beta_0 + \Sigma\beta_i X_i)}}$$

The above equation can also be written as -

$$log\left(\frac{P(Y)}{1 - P(Y)}\right) = \beta_0 + \Sigma\beta_i X_i$$

where, the left hand side of the equation represents the log-odds (logit).

A. Dataset Description

For the purpose of our study, we use the dataset acquired from Pew Research Center through a survey of U.S adults about aging, longevity and end-of-life issues in the year 2013. The survey includes a questionnaire about the views on aging and quality of life, medical advances, end-of-life decisions and radical life extension. Below is the link to the dataset- https://www.pewforum.org/dataset/survey-of-aging-and-longevity

1) Dependent variable: For the dependent variable we select the question, *"Overall, are you satisfied or dissatisfied with the way things are going in your life today?"*. We remove the responses with answers "Don't know/Refuse" to make the outcome variable dichotomous. In the survey, there are total 3,246 people answering they are satisfied (coded as 1) where as 630 people answering they are dissatisfied (coded as 0 in R).

2) Independent variable: For the independent variables, we are considering the following questions:

1) X_1: Rate: your personal financial situation?
2) X_2: Rate: your employment situation?
3) X_3: Rate: the number of friends you have?
4) X_4: Rate: your health?
5) X_5: How much do you think "Engineers" contribute to the well being of our society?
6) X_6: How much do you think "Medical doctors" contribute to the well being of our society?
7) X_7: Trend good/bad for American society: "More gay and lesbian couples raising children"
8) X_8: Trend good/bad for American society:"More people of different races marrying each other"
9) X_9: Age of the respondent?
10) X_{10}: Employment status of the respondent

Variables X_1,X_2,X_3 and X_4 are categorical variables and hence coded as factors with 6 levels (responses as: Excellent, Good, Fair, Poor, Doesn't Apply, Refused). Variables X_5 and X_6 are categorical variables with factors of 5 levels (responses as: a lot, some, not much, nothing, refused) whereas variable X_7 and X_8 are categorical variables with 4 factor levels(responses as: good, bad, no difference, refused). Variables X_9 is a continuous variable and variable X_{10} is a categorical variable with 4 factor levels (responses as: Full-time, Part-time, Not employed, Refused). From the independent categorical variables we remove the responses "Refused" and "Doesn't Apply" from our dataset and we are left with 412 people answering "dissatisfied" and 2274 people answering "satisfied" from the dependent variable.

B. Model Building and Diagnostics

To build a logistic regression model in R, we use the "glm" function which is the acronym for 'Generalized linear model' and select family as "Binomial". Initially we include all the predictors in our model and get the summary of our model 'm1' as in fig.13. We note that deviance of the residuals which is given by $2(LL(SaturatedModel) -$

```
> summary(m1)

Call:
glm(formula = Y ~ ., family = "binomial", data = df1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6985   0.2350   0.3163   0.4852   2.0394

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.248164   0.355857  11.938  < 2e-16 ***
X12         -0.033925   0.301096  -0.113 0.910292
X13         -1.081534   0.301219  -3.591 0.000330 ***
X14         -2.052986   0.316486  -6.487 8.77e-11 ***
X22         -0.265267   0.207807  -1.277 0.201777
X23         -0.753675   0.221400  -3.404 0.000664 ***
X24         -1.506902   0.243459  -6.190 6.03e-10 ***
X32         -0.114228   0.156859  -0.728 0.466477
X33         -0.361470   0.184286  -1.961 0.049825 *
X34         -1.028857   0.265401  -3.877 0.000106 ***
X42         -0.240468   0.176158  -1.365 0.172232
X43         -0.341726   0.202900  -1.684 0.092141 .
X44         -0.897250   0.269953  -3.324 0.000888 ***
X52         -0.096278   0.148071  -0.650 0.515554
X53         -0.565055   0.275610  -2.050 0.040345 *
X54         -0.381228   0.439890  -0.867 0.386137
X62         -0.162175   0.151087  -1.073 0.283096
X63         -0.444010   0.255433  -1.738 0.082164 .
X64         -0.351838   0.436835  -0.805 0.420574
X72         -0.331443   0.183627  -1.805 0.071078 .
X73         -0.046387   0.181584  -0.255 0.798369
X82         -0.058781   0.248799  -0.236 0.813232
X83          0.144507   0.142711   1.013 0.311260
X9          -0.013727   0.004261  -3.222 0.001273 **
X102         0.343771   0.185517   1.853 0.063876 .
X103         0.348161   0.161258   2.159 0.030847 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2302.1  on 2685  degrees of freedom
Residual deviance: 1730.2  on 2660  degrees of freedom
AIC: 1782.2

Number of Fisher Scoring iterations: 5
```

Fig. 13. Summary of Model 'm1'

$LL(ProposedModel))$is centered closely at 0 (Median=0.31). Next, we get a null deviance of 2302 which states how well the dependent variable is predicted when only the intercept is considered in model building. On the other hand, we get a Residual deviance of 1730 which states, our model has improved predictions when we added all the 10 independent predictors by minimizing the deviance.

To further optimize the model, we use the "step" function with default parameter as "both" that is backward and forward method to generate model 'm2'. This method simultaneously adds and drops independent variables to minimize the Akaike information Criteria (AIC) of the model and finally selecting the best model with minimum AIC as shown in fig.14.

```
> summary(m2)

Call:
glm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X7 + X9 + X10, family = "binomial",
    data = df1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7590   0.2394   0.3170   0.4829   2.0885

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.228420   0.355318  11.900  < 2e-16 ***
X12         -0.040967   0.300603  -0.136 0.891597
X13         -1.071800   0.300862  -3.562 0.000367 ***
X14         -2.062903   0.316191  -6.524 6.84e-11 ***
X22         -0.267591   0.207025  -1.293 0.196166
X23         -0.774478   0.220557  -3.511 0.000446 ***
X24         -1.514498   0.241907  -6.261 3.83e-10 ***
X32         -0.108373   0.156411  -0.693 0.488390
X33         -0.356961   0.183578  -1.944 0.051840 .
X34         -1.033071   0.264022  -3.913 9.12e-05 ***
X42         -0.240955   0.175897  -1.370 0.170729
X43         -0.357556   0.202062  -1.770 0.076804 .
X44         -0.929754   0.268687  -3.460 0.000539 ***
X52         -0.140807   0.140893  -0.999 0.317605
X53         -0.722282   0.261256  -2.765 0.005698 **
X54         -0.597298   0.400953  -1.490 0.136304
X72         -0.306365   0.172789  -1.773 0.076219 .
X73          0.008048   0.169002   0.048 0.962019
X9          -0.013544   0.004215  -3.214 0.001311 **
X102         0.351206   0.184636   1.902 0.057151 .
X103         0.356873   0.160803   2.219 0.026465 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2302.1  on 2685  degrees of freedom
Residual deviance: 1735.5  on 2665  degrees of freedom
AIC: 1777.5

Number of Fisher Scoring iterations: 5
```

Fig. 14. Summary of Model 'm2'

The created model 'm2' has dropped variables X6 and X8 as these variables were not significantly contributing to the prediction of the model. The same is evident from the Wald's test (analogous to t-statistic in linear regression) in fig.15., which tests the hypothesis that the co-efficient for the predictor is significantly different from zero [9]. To obtain the odds ratio

```
> Anova(m1, type="II", test="Wald")
Analysis of Deviance Table (Type II tests)

Response: Y
    Df    Chisq Pr(>Chisq)
X1   3 112.2381  < 2.2e-16 ***
X2   3  48.5893  1.595e-10 ***
X3   3  16.7339  0.0008016 ***
X4   3  11.2284  0.0105529 *
X5   3   4.6053  0.2030897
X6   3   3.8275  0.2807091
X7   2   5.0181  0.0813444 .
X8   2   1.4042  0.4955420
X9   1  10.3808  0.0012733 **
X10  2   6.1242  0.0467892 *
```

```
> Anova(m2, type="II", test="Wald")
Analysis of Deviance Table (Type II tests)

Response: Y
    Df    Chisq Pr(>Chisq)
X1   3 113.1039  < 2.2e-16 ***
X2   3  49.7959   8.83e-11 ***
X3   3  17.0866  0.0006783 ***
X4   3  12.2684  0.0065180 **
X5   3   9.4507  0.0238617 *
X7   2   5.7222  0.0572057 .
X9   1  10.3267  0.0013112 **
X10  2   6.4529  0.0396991 *
```

Fig. 15. Wald's test of Model 'm1' and 'm2'

we get the exponential value of each coefficients as in fig.16.

For the purpose of interpretation, the odds ratio of 0.96 for variable X12 suggests that the odds of being "satisfied

```
> round(exp(coef(m2)),2)
(Intercept)         X12         X13         X14         X22         X23         X24
      68.61        0.96        0.34        0.13        0.77        0.46        0.22
         X32         X33         X34         X42         X43         X44         X52
        0.90        0.70        0.36        0.79        0.70        0.39        0.87
         X53         X54         X72         X73          X9        X102        X103
        0.49        0.55        0.74        1.01        0.99        1.42        1.43
```

Fig. 16. Odds ratio: Model 'm2'

in life"(Y=1) are 0.96 times higher if the respondent has answered "financial situation" as "Good" in the survey compared to the ones who have answered otherwise (keeping all other factors equal). Similarly, the odds ratio of 0.90 for variable X32 suggests that the odds of being "satisfied"(Y=1) are 0.90 time higher if the respondent has selected the current "Health" aspect of life as "Good" compared to the person who has answered "Poor" or "Fair".

To check for influential data points in our model, we apply the max(cooks.distance()) function to our model and find the the maximum cook's distance to be 0.009 which is well under the threshold of 1.

C. Performance Evaluation

Unlike linear regression, in multivariate logistic regression we utilize pseudo R^2 measure to test amount of variation in dependent variable predicted by the independent variables. To get the values for pseudo R^2 we use the "nagelkerke()" function from the "rcompanion" package in R and create a dataframe to compare both models m1 and m2. The Cox and Snell R^2 is based on log likelihood in comparison with the baseline model whereas the Nagelkerke's model is expansion to Cox and Snell's R^2 adjusting the scale between 1 and 0. McFadden's R^2 is based on the log-likelihood kernels comparing the intercept-only and full estimated model [10]. We get a similar pseudo R^2 for both the models. A confu-

```
> ic
     Model McFadden_R2 Cox_Snell_R2 Nagelkerke_R2
1 Model m1       0.248        0.192         0.333
2 Model m2       0.246        0.190         0.330
```

Fig. 17. Pseudo R^2

sion matrix is a two dimensional matrix which summarizes the classification performance of a model with respect to the original dataset output categorizing the output into true positive(TP), true negative(TN), false positive(FP) and false negative(FN). We create a confusion matrix by calling the function "confusionMatrix()" from the "caret" package in R to compare the models as depicted in fig.18. The Accuracy of the model is defined by (TP+TN)/Total predictions. It states how overall the model accurately classifies. We get an accuracy of 86.3% for the model m1 and 86.4% for model m2. The sensitivity of a model is defined as TP/(TP+FN). Sensitivity defines how many true positives were correctly classified. We get a sensitivity of 30% for both the models which states that 30% of the people who are "Satisfied with their lives" are correctly classified. The specificity of the model is defined as TN/(TN+FP). Sensitivity defines how many true negatives were correctly classified. We get a sensitivity of 96% for both

```
> confusionMatrix(res,df1$Y)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  123   79
         1  289 2195

               Accuracy : 0.863
                 95% CI : (0.8494, 0.8758)
    No Information Rate : 0.8466
    P-Value [Acc > NIR] : 0.009185

                  Kappa : 0.3334

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.29854
            Specificity : 0.96526
         Pos Pred Value : 0.60891
         Neg Pred Value : 0.88366
             Prevalence : 0.15339
         Detection Rate : 0.04579
   Detection Prevalence : 0.07520
      Balanced Accuracy : 0.63190

       'Positive' Class : 0
```

```
> confusionMatrix(res1,df1$Y)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  124   77
         1  288 2197

               Accuracy : 0.8641
                 95% CI : (0.8506, 0.8769)
    No Information Rate : 0.8466
    P-Value [Acc > NIR] : 0.005804

                  Kappa : 0.338

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.30097
            Specificity : 0.96614
         Pos Pred Value : 0.61692
         Neg Pred Value : 0.88410
             Prevalence : 0.15339
         Detection Rate : 0.04617
   Detection Prevalence : 0.07483
      Balanced Accuracy : 0.63355

       'Positive' Class : 0
```

Fig. 18. Confusion Matrix for model m1 and m2

the models which states that 96% of the people who are "Dissatisfied with their lives" were correctly classified [11]. The Receiver operating characteristic (ROC)depicts the performance of the model by plotting the Sensitivity on vertical axis against the Specificity on the horizontal x-axis.To plot the curves, the predictions are sorted by the estimated probabilities in ascending order. A ROC curve of the model m2 is shown in Fig.19. Lastly, to test the goodness of fit, we use the Hosmer-

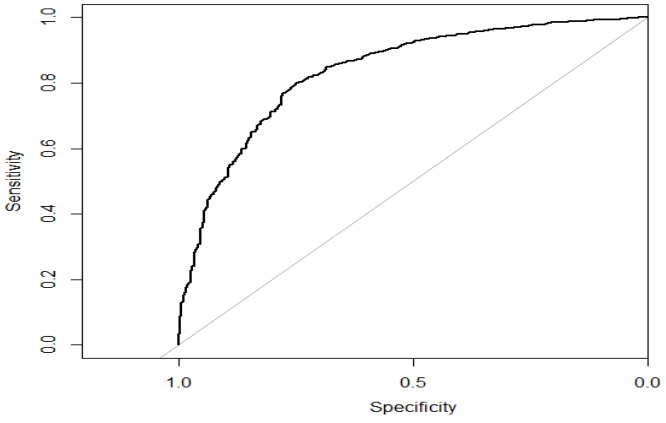

Fig. 19. ROC curve

Lemeshow test which tests if the observed and expected event rates match in the given population subgroups. We use the hoslem.test() function from the "ResourceSelection" package and set default group as 10 (thumb rule: group ¿number of predictors+1). A p-value of 0.12 is achieved indicating no evidence of poor fit.

Hence, we select the model 'm2' as the best fit model and the equation of the model is as follows:

$$log\left(\frac{P(Y)}{1-P(Y)}\right) = 4.23 - 0.04X12 - 1.07X13 - 2.06X14$$
$$-0.27X22 - 0.77X23 - 1.51X24 - 0.11X32 - 0.36X33$$
$$-1.03X34 - 0.24X42 - 0.36X43 - 0.93X44 - 0.14X52$$
$$-0.72X53 - 0.60X54 - 0.31X72 + 0.01X73 - 0.01X9$$
$$+0.35X102 + 0.36X103$$

IV. Principal Components Analysis

Principal Component Analysis (PCA) helps us to summarize large set of correlated variables in a dataset into a smaller number of representative components (principal components) which inturn explain most of the variability compared to the original dataset [12]. PCA is an unsupervised approach to reduce the dimensionality of the and plot the data in low dimensional space. The core objective of PCA is to calculate Eigen vectors(factor weights) and extract the maximum variance from all the variables by successive factoring until most of the variance is explained by the variables in picture.

PCA is a three step process which includes assessing the suitability of data, factor extraction and lastly factor rotation. To illustrate the process we consider a dataset on a survey done on the life goals containing 30 questions and 310 respondents. The link to the survey can be found as below-https://study.sagepub.com/sites/default/files/ch$_1$0$b_l eanne.sav$

A. Assessing the suitability of data

For efficient factor extraction, we need a ratio of 10:1 for sample size v/s the factors or variables. In our dataset, we have a sample of 310 respondents and 30 variables which suffices the requirement for PCA [13].Next, we need to check the correlations of between some of the variables to be greater than 0.3.

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	X25	X26	X27	X28	X29	X30
X1	1.00	0.19	0.18	0.14	0.06	0.13	0.33	0.16	0.11	0.01	-0.06	0.16	0.17	0.11	0.10	0.01	-0.06	-0.12	0.32	0.07	0.17	-0.04	0.01	0.01	0.19	0.19	0.11	-0.07	-0.01	0.05
X2	0.19	1.00	0.24	0.12	0.08	0.21	0.33	0.34	0.12	0.06	0.03	0.21	0.19	0.57	0.22	0.08	-0.03	-0.03	0.27	0.34	0.24	0.01	-0.02	0.05	0.23	0.36	0.27	-0.02	0.06	0.03
X3	0.18	0.24	1.00	0.20	0.00	0.07	0.15	0.27	0.37	0.11	0.00	0.09	0.19	0.20	0.39	0.06	0.02	-0.07	0.32	0.13	0.50	-0.03	-0.04	-0.09	0.35	0.16	0.50	-0.04	-0.05	-0.09
X4	0.14	0.12	0.20	1.00	0.39	0.32	0.14	0.15	0.23	0.69	0.35	0.30	0.19	0.20	0.23	0.61	0.28	0.34	0.14	0.20	0.18	0.42	0.29	0.31	0.17	0.21	0.23	0.36	0.27	0.16
X5	0.06	0.08	0.00	0.39	1.00	0.33	0.11	0.13	0.13	0.41	0.49	0.30	0.15	0.07	0.16	0.38	0.46	0.36	0.03	0.07	0.08	0.30	0.40	0.28	-0.02	0.16	0.12	0.26	0.28	0.28
X6	0.13	0.21	0.07	0.32	0.33	1.00	0.31	0.22	0.09	0.33	0.31	0.81	0.21	0.25	0.06	0.33	0.29	0.40	0.17	0.11	0.03	0.18	0.24	0.54	0.15	0.14	0.01	0.13	0.23	0.47
X7	0.33	0.33	0.15	0.14	0.11	0.31	1.00	0.38	0.19	0.12	-0.01	0.37	0.21	0.34	0.16	0.14	-0.09	-0.01	0.34	0.27	0.19	-0.01	0.03	0.08	0.19	0.34	0.14	-0.06	0.01	0.14
X8	0.16	0.34	0.27	0.15	0.13	0.22	0.38	1.00	0.26	0.12	0.12	0.18	0.15	0.33	0.24	0.16	0.08	0.05	0.33	0.25	0.27	0.02	0.06	0.10	0.22	0.34	0.27	-0.07	0.05	0.15
X9	0.11	0.12	0.37	0.23	0.13	0.09	0.19	0.26	1.00	0.20	0.10	0.16	0.27	0.14	0.70	0.23	0.11	-0.02	0.23	0.13	0.53	0.07	0.10	-0.05	0.36	0.24	0.53	0.02	0.03	0.00
X10	0.01	0.06	0.11	0.69	0.41	0.33	0.12	0.12	0.20	1.00	0.52	0.32	0.21	0.17	0.17	0.69	0.37	0.43	0.10	0.20	0.15	0.60	0.43	0.41	0.12	0.13	0.16	0.52	0.41	0.34
X11	-0.06	0.03	0.00	0.35	0.49	0.31	-0.01	0.12	0.10	0.52	1.00	0.22	0.20	0.06	0.07	0.50	0.67	0.60	0.07	0.13	0.04	0.49	0.64	0.41	0.02	0.09	0.07	0.44	0.65	0.41
X12	0.16	0.21	0.09	0.30	0.30	0.81	0.37	0.18	0.16	0.32	0.22	1.00	0.25	0.24	0.13	0.32	0.20	0.34	0.20	0.05	0.02	0.17	0.24	0.53	0.19	0.12	0.05	0.12	0.18	0.48
X13	0.17	0.19	0.19	0.19	0.15	0.21	0.21	0.15	0.27	0.21	0.20	0.25	1.00	0.25	0.32	0.25	0.06	0.07	0.47	0.29	0.32	0.11	0.17	0.10	0.33	0.24	0.24	0.07	0.19	0.14
X14	0.11	0.57	0.20	0.20	0.07	0.25	0.34	0.33	0.14	0.17	0.06	0.24	0.25	1.00	0.30	0.16	0.06	0.03	0.27	0.54	0.26	0.04	0.05	0.05	0.19	0.54	0.23	-0.03	0.10	0.00
X15	0.10	0.22	0.39	0.23	0.16	0.06	0.16	0.24	0.70	0.17	0.07	0.13	0.32	0.30	1.00	0.24	0.07	-0.07	0.27	0.18	0.61	0.07	0.07	-0.12	0.41	0.26	0.59	0.03	0.05	-0.11
X16	0.01	0.08	0.06	0.61	0.38	0.33	0.14	0.16	0.23	0.69	0.50	0.32	0.25	0.16	0.24	1.00	0.39	0.46	0.18	0.11	0.16	0.49	0.39	0.34	0.21	0.16	0.16	0.39	0.42	0.34
X17	-0.06	-0.03	0.02	0.28	0.46	0.29	-0.09	0.08	0.11	0.37	0.67	0.20	0.06	0.06	0.07	0.39	1.00	0.66	0.05	0.11	0.06	0.42	0.63	0.35	0.00	0.09	0.12	0.37	0.53	0.35
X18	-0.12	-0.03	-0.07	0.34	0.36	0.40	-0.01	0.05	-0.02	0.43	0.60	0.34	0.07	0.03	-0.07	0.46	0.66	1.00	0.03	0.10	-0.05	0.49	0.53	0.58	-0.02	-0.01	0.01	0.45	0.42	0.48
X19	0.32	0.27	0.32	0.14	0.03	0.17	0.34	0.33	0.23	0.10	0.07	0.20	0.47	0.27	0.27	0.18	0.05	0.03	1.00	0.28	0.34	-0.04	0.01	0.04	0.36	0.29	0.27	-0.05	0.08	0.06
X20	0.07	0.34	0.13	0.20	0.07	0.11	0.27	0.25	0.13	0.20	0.13	0.05	0.29	0.54	0.18	0.11	0.11	0.10	0.28	1.00	0.28	0.09	0.07	0.12	0.13	0.61	0.25	0.08	0.13	0.06
X21	0.17	0.24	0.50	0.18	0.08	0.03	0.19	0.27	0.53	0.15	0.04	0.02	0.32	0.26	0.61	0.16	0.06	-0.05	0.34	0.28	1.00	0.05	-0.07	-0.10	0.33	0.33	0.71	0.00	0.06	-0.05
X22	-0.04	0.01	-0.03	0.42	0.30	0.18	-0.01	0.02	0.07	0.60	0.49	0.17	0.11	0.04	0.07	0.49	0.42	0.49	-0.04	0.09	0.05	1.00	0.51	0.45	0.08	0.09	0.06	0.78	0.47	0.31
X23	0.01	-0.02	-0.04	0.29	0.40	0.24	0.03	0.06	0.10	0.43	0.64	0.24	0.17	0.05	0.07	0.39	0.63	0.53	0.01	0.07	0.07	0.51	1.00	0.44	0.03	0.07	0.10	0.47	0.56	0.40
X24	0.01	0.05	-0.09	0.31	0.28	0.54	0.08	0.10	-0.05	0.41	0.41	0.53	0.10	0.05	-0.12	0.34	0.35	0.58	0.04	0.12	-0.10	0.45	0.44	1.00	0.12	0.10	-0.01	0.37	0.31	0.65
X25	0.19	0.23	0.35	0.17	-0.02	0.15	0.19	0.22	0.36	0.12	0.02	0.19	0.33	0.19	0.41	0.21	0.00	-0.02	0.36	0.13	0.33	0.08	0.03	0.12	1.00	0.33	0.34	0.04	0.10	0.08
X26	0.19	0.36	0.16	0.21	0.16	0.14	0.34	0.34	0.24	0.13	0.09	0.12	0.24	0.54	0.26	0.16	0.09	-0.01	0.29	0.61	0.33	0.09	0.07	0.10	0.33	1.00	0.41	0.04	0.10	0.05
X27	0.11	0.27	0.50	0.23	0.12	0.01	0.14	0.27	0.53	0.16	0.07	0.05	0.24	0.23	0.59	0.16	0.12	0.01	0.27	0.25	0.71	0.06	0.10	-0.01	0.34	0.41	1.00	0.05	0.08	-0.08
X28	-0.07	-0.02	-0.04	0.36	0.26	0.13	-0.06	-0.07	0.02	0.52	0.44	0.12	0.07	-0.03	0.03	0.39	0.37	0.45	-0.05	0.08	0.00	0.78	0.47	0.37	0.04	0.04	0.05	1.00	0.47	0.24
X29	-0.01	0.06	-0.05	0.27	0.28	0.23	0.01	0.05	0.03	0.41	0.65	0.18	0.19	0.10	0.05	0.42	0.53	0.42	0.08	0.13	0.06	0.47	0.56	0.31	0.10	0.10	0.08	0.47	1.00	0.34
X30	0.05	0.03	-0.09	0.16	0.28	0.47	0.14	0.15	0.00	0.34	0.41	0.48	0.14	0.00	-0.11	0.34	0.35	0.48	0.06	0.06	-0.05	0.31	0.40	0.65	0.08	0.05	-0.08	0.24	0.34	1.00

Fig. 20. Correlation matrix of variables

On inspecting fig.20., we note that out of 870 correlations (excluding correlations with self), we have 258 correlations above 0.3. Further, we test the sampling adequacy of the variables by performing Kaiser-Meyer-Olkin's (KMO) test to check if there are widespread correlations between variables. Also, to check the homogeneity of variances, we perform Bartlett's test for sphericity [14]. We get a test statistic of 0.85

KMO and Bartlett's Test		
Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.853
Bartlett's Test of Sphericity	Approx. Chi-Square	4751.733
	df	435
	Sig.	000

Fig. 21. Pseudo R square

(greater than 0.6) suggesting meritoriously adequate sampling.

We get a significant statistic (p¡0.0) for Bartlett's test and hence we reject the null hypothesis of equal variance.

B. Factor extraction

For factor extraction. PCA computes eigenvectors called factor weights by generating new set of factors which has a linear relationship to the original variables. Further, these factor weights extract maximum possible variance (varimax) explained by successive factoring(spss). By analyzing the

Total Variance Explained

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	7.422	24.739	24.739	7.422	24.739	24.739	4.283	14.276	14.276
2	4.819	16.062	40.801	4.819	16.062	40.801	3.569	11.895	26.171
3	2.356	7.854	48.655	2.356	7.854	48.655	3.059	10.198	36.369
4	1.654	5.514	54.169	1.654	5.514	54.169	2.969	9.896	46.265
5	1.276	4.254	58.423	1.276	4.254	58.423	2.847	9.490	55.754
6	1.161	3.870	62.293	1.161	3.870	62.293	1.962	6.539	62.293
7	1.039	3.465	65.758						
8	.937	3.125	68.883						
9	.843	2.810	71.693						
10	.808	2.693	74.386						
⋮	⋮	⋮	⋮						
28	.175	.583	98.930						
29	.170	.567	99.497						
30	.151	.503	100.000						

Extraction Method: Principal Component Analysis.

Fig. 22. Factor Extraction

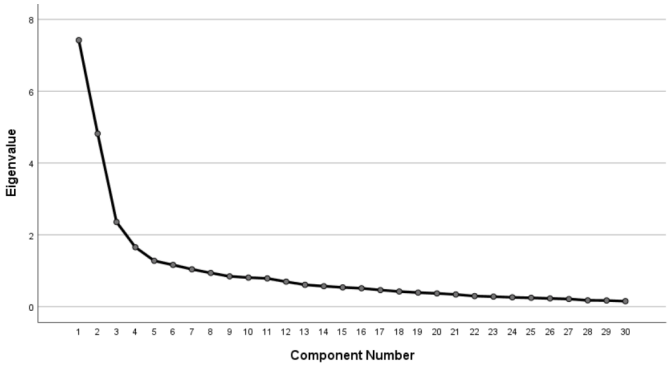

Fig. 23. Scree plot

output from SPSS using "dimension reduction" and "factor analysis", we get a table of all the eigenvalues and a scree plot for the variables. On inspecting we find, 62.3% of the variability is explained by 6 components. Further, we see 6 components above the elbow of the scree plot before the plot becomes parallel to x-axis suggesting 6 components to be extracted.

C. Factor Rotation

To understand which variables are more related to the extracted factors, we examine the factor loadings which are correlations among factors and variables. High factor loading indicate higher correlation and more variance explained by the variable for the particular factor. We use "varimax" rotation which is a type of orthogonal factor rotation technique to rotate the axes of factors, such that a set of variables are found which are highly loaded towards one specific factor. The varimax rotation technique maximizes the dispersion of loadings within factors to get few variables of high loadings, hence the name "varimax". Next, we re-run the factor rotation process, but this time we explicitly specify the extracted number of components as 6 in SPSS and get a sorted table as shown in fig.24.

	Component					
	1	2	3	4	5	6
I will successfully hide the signs of aging.	0.428					
I will have many expensive possessions.	0.683					
My image will be one others find appealing.	0.716					
I will achieve the "look" I've been after.	0.768					
I will have people comment often about how attractive I look.	0.803					
I will keep up with fashions in hair and clothing.	0.831					
I will assist people who need it, asking nothing in return.		0.625				
I will help others improve their lives.		0.766				
I will help people in need.		0.781				
I will work for the betterment of society.		0.795				
I will work to make the world a better place.		0.818				
I will be recognized by lots of different people.			0.744			
My name will be known by many people.			0.781			
I will be admired by many people.			0.649			
My name will appear frequently in the media.	0.437		0.675			
I will be famous.	0.457		0.712			
I will feel that there are people who really love me, and whom I love.				0.653		
At the end of my life, I will look back on my life as meaningful and complete.				0.431		
I will have good friends that I can count on.				0.443		
I will share my life with someone I love.				0.814		
I will have committed, intimate relationships.				0.785		
I will have deep, enduring relationships.				0.751		
I will have a job that pays very well.					0.852	
I will be financially successful.					0.834	
I will be rich.	0.431				0.566	
I will have enough money to buy everything I want.	0.469				0.573	
I will gain increasing insight into why I do the things I do.						0.617
I will choose what I do, instead of being pushed along by life.						0.597
I will know and accept who I really am.						0.705
I will continue to grow and learn new things.		0.402				0.524
Extraction Method: Principal Component Analysis.						
Rotation Method: Varimax with Kaiser Normalization.						
a. Rotation converged in 7 iterations.						

Fig. 24. Rotated Component Matrix

D. Interpreting and understanding factors

We interpret each factor created by analyzing them with respect to each variable to the factor weights and try to assign a commonality to the created factor.

1) Factor 1: On analyzing the questions of the survey categorized as factor 1, we can see a pattern of questions towards a goal of "self beautification". Descriptives such as looking better, fashion and clothing, attractiveness point towards a common factor of "self beautification".

2) Factor 2: The second factor on examining, we find a commonality as "to be helpful". Features such as helping people in need, improving other lives, making world a better place converge towards helpful nature of an individual.

3) Factor 3: The third factor, suggests to be converging towards "recognition by society". Ratings towards questions such as known by many people, admired by people, being famous can be categorized as recognition by the society. We note that the weights of variable "I will be famous" and "My name will appear frequently in media" is shared by component 1 too, but in this case more weightage is given for factor 3 by varimax rotation, and under reasonable understanding, these 2 variables should be recognized under factor 3.

4) Factor 4: The fourth factor converges towards a common factor of having "good relationship goals". Variables such as having good friends, being committed and having enduring relationships explain the factor 4 to be more leaned towards relationship goals in life.

5) Factor 5: The fifth factor depicts the goal of "being financially independent". Variables such as to be rich, to have money to buy everything, to have a financially successful job points towards one's financial stability. We note that the

variables being rich and having money to buy everything are also shared with factor 1, but less weights are given whereas they are highly weighted under factor 5. Hence, through common understanding and high weights we classify them under factor 5.

6) Factor 6: Lastly, the sixth factor talks more about "self-development goal or knowledge expansion". The variables such as growing and leaning new things, gaining increasing insights and learning new things converge towards attaining new knowledge and understanding.

In this way using PCA, we were able to trim down 30 variables to 6 logically related variables. PCA helped in reducing the complexity in analyzing the dataset. The 6 extracted principal components depict 62.3% of the variability of the original dataset variables.

REFERENCES

[1] P. Cowpertwait and A. Metcalfe, *Introductory Time Series with R*, ser. Use R! Springer New York, 2009. [Online]. Available: https://books.google.ie/books?id=QFiZGQmvRUQC

[2] R. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018. [Online]. Available: https://books.google.ie/books?id=_bBhDwAAQBAJ

[3] "6.6 stl decomposition — forecasting: Principles and practice," Otexts.com, 2020. [Online]. Available: https://otexts.com/fpp2/stl.html

[4] G. Shmueli and K. Lichtendahl, *Practical Time Series Forecasting with R: A Hands-On Guide [2nd Edition]*, ser. Practical Analytics. Axelrod Schnall Publishers, 2016. [Online]. Available: https://books.google.ie/books?id=qxWXDwAAQBAJ

[5] "R: Akaike's an information criterion," Ethz.ch, 2020. [Online]. Available: https://stat.ethz.ch/R-manual/R-patched/library/stats/html/AIC.html

[6] https://www.facebook.com/peterstats, "Error, trend, seasonality - ets and its forecast model friends," free range statistics, 2020. [Online]. Available: http://freerangestats.info/blog/2016/11/27/ets-friends

[7] W. Contributors, "Augmented dickey–fuller test," Wikipedia, 07 2020.

[8] R. Nau, "Lecture notes on forecasting introduction to arima models." [Online]. Available: https://people.duke.edu/ rnau/Slides$_o n_A RIMA_m odels --Robert_N au.pdf$

[9] A. Field, *Discovering Statistics Using IBM SPSS Statistics*. SAGE Publications, 2013. [Online]. Available: https://books.google.ie/books?id=AlNdBAAAQBAJ

[10] "Ibm knowledge center," Ibm.com, 2018.

[11] C. Sammut and G. Webb, *Encyclopedia of Machine Learning*, ser. Encyclopedia of Machine Learning. Springer US, 2011. [Online]. Available: https://books.google.ie/books?id=i8hQhp1a62UC

[12] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, ser. Springer Texts in Statistics. Springer New York, 2013. [Online]. Available: https://books.google.ie/books?id=qcI_AAAAQBAJ

[13] R. M. Thorndike, "Book review: Psychometric theory by jum nunnally and ira bernstein new york: Mcgraw-hill, 1994, xxiv+ 752 pp," *Applied Psychological Measurement*, vol. 19, no. 3, pp. 303–305, 1995.

[14] "1.3.5.7. bartlett's test," Nist.gov, 2021. [Online]. Available: https://www.itl.nist.gov/div898/handbook/eda/section3/eda357.htm