Fergusson College

(Autonomous), Pune

Insuring the Harvest:

"Statistical Insights into Crop Insurance"

TYBSc Statistics

Fergusson College (Autonomous), Pune

Department of Statistics

TYBSc Statistics Project 2023-24

Insuring the Harvest: Statistical Insights into Crop Insurance

Project Guide: Mr. Shital Gadekar

Project Group 4 A:

Name	Roll No.
Snehal Jha	212201
Ankit Nanaware	212202
Vishal Masirkar	212210
Vaishnav Amrutkar	212147
Tanya Sood	212144
Vijay Jadhav	212139
Anand Kumar	212145

Deccan Education Society's

Fergusson College (Autonomous),Pune

Department of Statistics

T. Y. B. Sc.

Year 2023-24

STS3609: Statistics Practical III

CERTIFICATE

This is to certify that Mr./Ms. _____

Roll no._____ , has satisfactorily completed the project work entitled **"Insuring the Harvest: Statistical Insights into Crop Insurance''** during the academic year 2023 – 24 as per the rules and regulations laid down by FERGUSSON COLLEGE (Autonomous), Pune.

Place : Pune

Date : / / 2024

Signature of Guide
Mr. Shital Gadekar

Signature of HOD
Dr. Subhash Shende
(Department of Statistics)

ACKNOWLEDGEMENT

We would like to extend our deepest gratitude to everyone who contributed to the success of this project. This journey has been one of collaboration, learning, and mutual support, culminating in a project that we are proud to present.

First and foremost, we express our sincere thanks to our project guide, Mr. Shital Gadekar, whose guidance, expertise, and patience were instrumental in steering this project towards its completion.

We are also deeply grateful to the teaching and non-teaching staff of Fergusson College, particularly those in the Department of Statistics, for providing us with the resources and environment conducive to our research and development efforts. Their assistance in navigating academic challenges was crucial.

We would also like to acknowledge the support of our families and friends, who provided encouragement, understanding, and motivation throughout the duration of this project.

Lastly, we extend our gratitude to each other. This project was a collaborative effort that required dedication, compromise, and teamwork. This project is not only a reflection of our hard work but also a testament to the support and guidance we received from all those mentioned above. Thank you for making this journey memorable and our project a success.

Table of Contents

Motivation

Agriculture plays a vital role in the Indian economy. Over 70 per cent of the rural households depend on agriculture as their principal means of livelihood. Agriculture along with fisheries and forestry accounts for one-third of the nation's Gross Domestic Product (GDP) and is its single largest contributor. Agricultural exports constitute a fifth of the total exports of the country.

In a nation with a profound reliance on agriculture, the significance of studying crop insurance becomes unmistakably apparent. The intricacies and challenges inherent in the agricultural sector underscore the pivotal role that crop insurance plays in mitigating risks and ensuring the economic stability of this vital industry.

Introduction

Insurance is a tool to protect you against a small probability of a large, unexpected loss. It is a technique of providing people a means to transfer and share risk where losses suffered by few are met from the funds accumulated through small contributions made by many who are exposed to similar risks.

Crop insurance is critical in lowering farmers' financial risks associated with crop failures caused by bad weather conditions, pests, diseases, or other unforeseen causes. It offers farmers a safety net by compensating them for losses sustained during such disasters, ensuring their economic stability and food security.

Start and End Year	Name of Crop Insurance Scheme	Primary Feature of the Scheme
1972–78	First individual approach crop insurance scheme	First scheme in India after independence
1979–84	Pilot Crop Insurance Scheme	First area index-based scheme
1985–99	Comprehensive Crop Insurance Scheme (CCIS)	Crop Insurance made mandatory for loanee farmers
1997–98	Experimental Crop Insurance Scheme (ECIS)	Fully subsidised scheme
1999–2016	National Agricultural Insurance Scheme (NAIS)	Sharecroppers were included for insurance cover
2003–04	Farm Income Insurance Scheme (FIIS)	First scheme to cover farm income, rather than the cost of cultivation
2007–to date	Weather Based Crop Insurance Scheme (WBCIS)	First scheme to ascertain crop loss based on deviation in rainfall
2010–2016	Modified National Agricultural Insurance Scheme (MNAIS)	Private sector participation encouraged. Immediate partial payment to affected farmers introduced
2016–to date	Pradhan Mantri Fasal Bima Yojana (PMFBY)	Premium rates lowered. Use of technology emphasised

Objectives:

I. To study the influence of socioeconomic factors such as age, income, etc. on willingness of farmers to ensure their crops.

II. To study the influence of macroeconomic factors on the aggregate enrolment of farmers in crop insurance schemes.

III. To determine the best drought index for prediction of crop yield.

IV. To model the number of claims in a crop insurance scheme.

V. To model the number of claims in a crop insurance scheme.

VI. To calculate one-time premium for crop insurance.

Statistical techniques used

1. Logistic regression
2. Subset regression (using bestglm and StepAIC)
3. Generalized Linear Model
4. Lasso Regression
5. Multiple Linear Regression
6. Time Series Analysis (ARIMA model)
7. Actuarial Statistics

Software used

1. RStudio
2. Microsoft Excel
3. EasyFit

(The R-codes and their outputs used in the models in this project are listed in the appendix.)

Willingness to Insure (WTI) Model

Assessing rural farmers' willingness to insure their farms in Maharashtra based on socio-economic factors.

Willingness to Insure (WTI) for crop insurance refers to inclination of farmers or agricultural stakeholders to purchase crop insurance. It is a key factor in determining the demand for and the success of crop insurance programs. Understanding farmers' willingness to pay helps policymakers, insurance providers, and other stakeholders design effective and sustainable insurance schemes.

Data Collected : The data used in this study were obtained randomly through a survey conducted using a google form circulated across the different districts of Maharashtra. A random sample of size 110 was obtained through the google form.

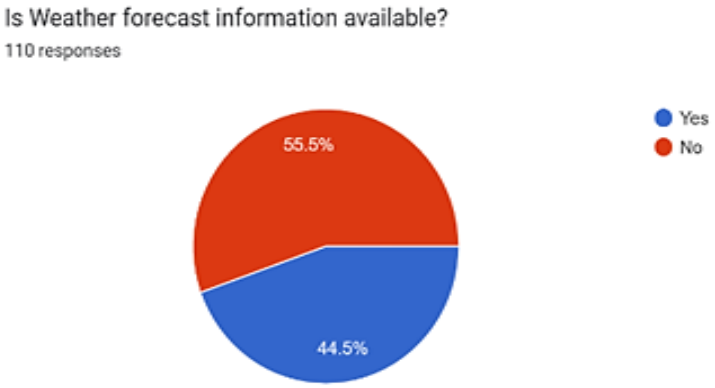
The choice of variables used in this study is made based on previous studies.

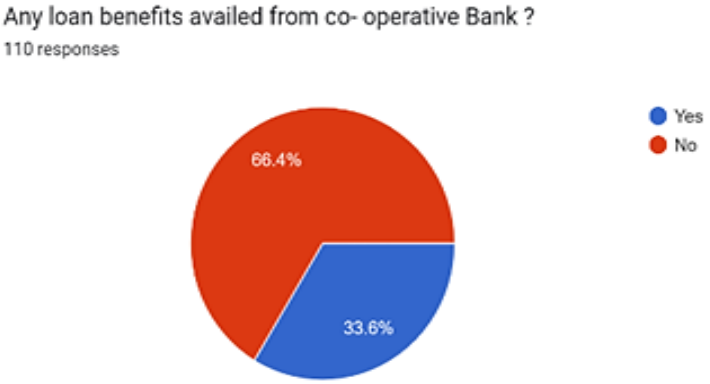
Analysis of data

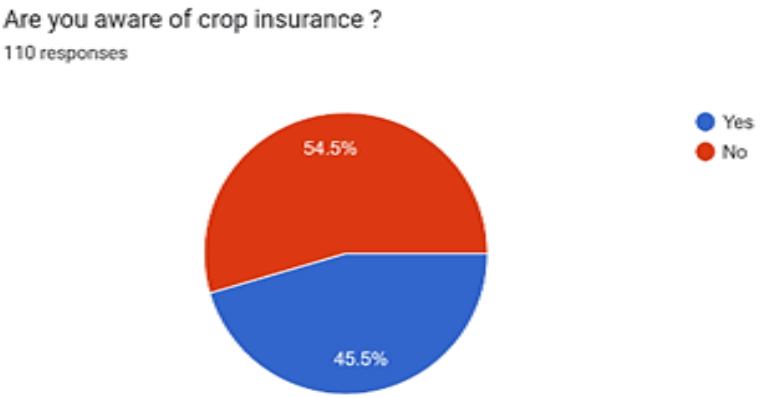
Summary statistics:

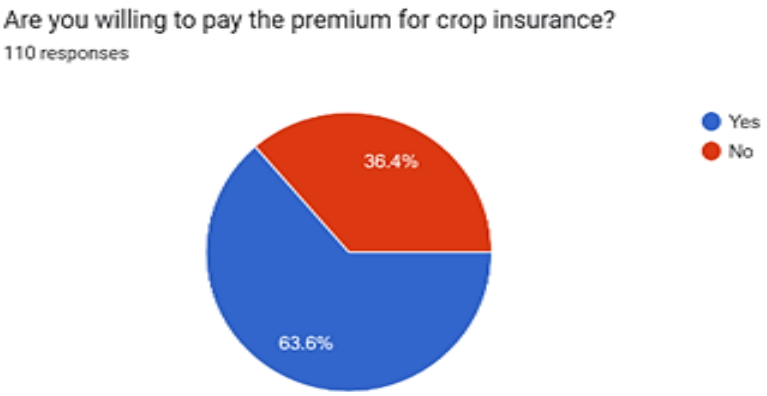
Variable	Description	Sample Mean	Standard deviation
WTI (Willingness to Insure) 'y'	1, if the farmer is willing to pay for crop insurance, 0, otherwise	0.618182	0.488056
Age 'x1'	Age of household head (years)	45.05455	13.49267
Education 'x2'	Number of years of schooling	11.77273	3.582985
Livestock 'x3'	Number of livestock owned	2.309091	4.380651
Income 'x4'	Yearly household income in ('000 Rs)	299.4364	435.2783
Farm size 'x5'	Farmland under production (acre)	9.979545	7.308263
Insurance awareness 'x6'	1 if a farmer is aware of crop insurance, 0 otherwise	0.2	0.98428

Loan access ‘x7’	1 if farmer is loanee, 0 otherwise	-0.12727	0.996407
Household size ‘x8’	Total number of household members	4.790909	1.299933









Above pie diagrams shows that only about **45.5 %** of rural farmers in the sample area are aware of the crop insurance scheme, and **63.6 %** are willing to ensure in crop insurance. Also, it is observed that about **33.6 %** farmers take a loan from co-operative banks for the farming expenditures while **44.5 %** acknowledged having access to weather forecast information.

Analytical framework of Multiple Logistic regression model

Since our objective is to investigate whether the farmer is willing to insure the farm or not, we use a logistic regression model for this study.

Multiple logistic regression is a statistical tool used to model the relationship between a binary response variable and multiple predictor variables.

Key properties of the logistic regression equation

- Logistic regression's dependent variable obeys 'Bernoulli distribution'.
- Estimation/prediction is based on 'maximum likelihood.'

We have dependent variable i.e. (response) in this study –

WTI = Willingness to Insure

And the independent variables are: -

1) Farmer's age; 2) Farmer's years of education; 3)Livestock owned by farmer; 4) Income; 5) Farm size; 6) Weather information available or not; 7) Loans availed or not (whether farmer take loan for farming or not); 8) Household size.

The Multiple logistic regression on the willingness-to-insure (WTI) can be specified as:

$$Yi = \prod(Xi) + \varepsilon i$$

where, $\prod(Xi) = \frac{e^{\beta 0+\beta 1X1+\beta 2X2+\cdots+\beta kXk}}{1+e^{\beta 0+\beta 1X1+\beta 2X2+\cdots+\beta kXk}}$

Note that β0, β1, β2… are the regression coefficients.

Yi = 1 if Yi >0

= 0 if Yi ≤0

In the above equation, Yi is the response variable that takes the value of 1 if a farmer is willing to insure his/her farm and 0 otherwise, Xi represents a vector of independent variables which are hypothesised to influence the farmers' WTI, β is a vector of parameters to be estimated and εi is the error term which is assumed to be independent with mean '0' and variance '∏(Xi)*(1-∏(Xi))' .

Variable Selection

We have to reduce the insignificant predictors for getting best logistic regression model.

We use different statistical tools in R software for getting best fit regression model. We use stepAIC function using library “MASS” and another method using bestglm function using library “bestglm”.

1. `*stepAIC*` *(Stepwise AIC selection):*
`stepAIC` uses a stepwise selection method based on the Akaike Information Criterion (AIC). It iteratively adds or removes predictors to find the model that minimizes the AIC.

2. `*bestglm*` *(Best Subset Selection):*
`bestglm` performs an exhaustive search over all possible combinations of predictors and selects the best subset based on a chosen criterion (e.g., AIC, BIC, or other).

According to stepAIC, the best predictors of WTI are **livestock**, **farm size** and **insurance awareness**.

According to bestglm, **education**, **farm size** and **insurance awareness** are the significant predictors for WTI.

Since in StepAIC function both forward and backward step elimination is involved there might be some error in detecting significant predictors, so it is better to use bestglm function for detecting best predictors for the given model. Also, the bestglm function is more suitable for a smaller number of predictors.

Fitting the best logistic regression model

X2 = Education
X5 = Farm size
X7 = Insurance awareness

$Y = \prod(X) + \varepsilon$

where, $\prod(X)= \frac{e^{\beta 0+\beta 1X1+\beta 2X2+\cdots+\beta kXk}}{1+e^{\beta 0+\beta 1X1+\beta 2X2+\cdots+\beta kXk}}$, where β0, β1, β2 …… are the regression coefficients.

Therefore,

$\hat{Y} = \frac{e^{\beta 0+\beta 2X2+\beta 5X5+\beta 6X6}}{1+e^{\beta 0+\beta 2X2+\beta 5X5+\beta 6X6}}$, i.e.

$$\hat{Y} = \frac{e^{-10.9875+ 0.7079X2 + 0.4291X5 +2.5652X6}}{1 + e^{-10.9875+ 0.7079X2 + 0.4291X5 +2.5652X6}}$$

This is the required equation of multiple logistic regression model for assessment of farmers willingness to insure i.e. to enroll in crop insurance scheme.

Model Validation

1)Testing significance of regressors

We compare the test statistic G with $\chi 2$ (n,0.05), where

G = Null Deviance – Residual Deviance

Here,
n = 3
Null deviance = 146.288
Residual deviance = 23.476
G = 122.812, and $\chi 2$ (3,0.05) = 9.837.

As G > $\chi 2$ (3,0.05), we may conclude that regression coefficients are significant at 5% level of significance.

2) By checking the McFadden's Pseudo R^2

McFadden Pseudo R2 = 1 – (Residual deviance/Null deviance)

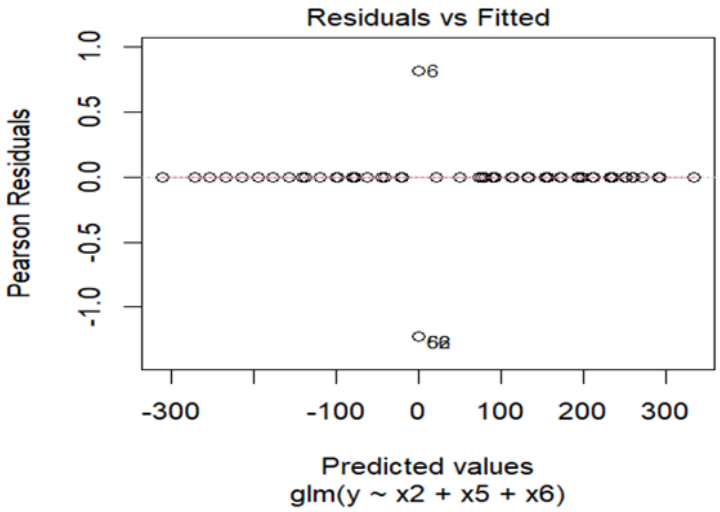
= 1 – (23.476/146.288)

= 0.839523

As McFadden Pseudo R2 is 0.8395, 83.95% of the variation is explained by the model, suggesting a strong goodness-of-fit.

Validating Assumptions

1.Linearity



As in the above residuals vs. fitted plot, we can observe points plotted are close to red line thus the assumption of linearity is satisfied.

2.Multicollinearity

Multicollinearity in regression analysis refers to the situation where two or more predictor variables in a regression model are highly correlated. This high correlation can cause issues in the estimation of the regression coefficients and can affect the reliability and interpretability of the results.
The values of VIF exceeding 5 indicates multicollinearity. Values of VIF close to 1 are desirable.

```
> library(car)
> vif(reg)

      x2    x5    x6

1.556061 1.140413 1.519208
```

Since VIF values are close to 1 there may be very less or no multicollinearity.

3.Independence of residuals

One way to determine if this assumption is met is to perform a Durbin-Watson test, which is used to detect the presence of autocorrelation in the residuals of a regression. This test uses the following hypotheses:

H0 (null hypothesis): There is no correlation among the residuals.

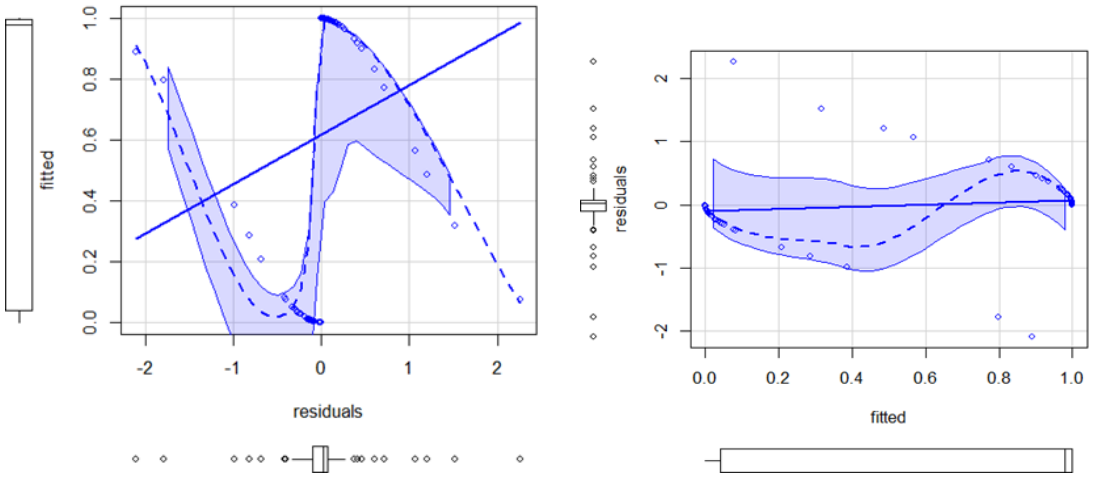
HA (alternative hypothesis): The residuals are autocorrelated.

```
> durbinWatsonTest(reg)
```

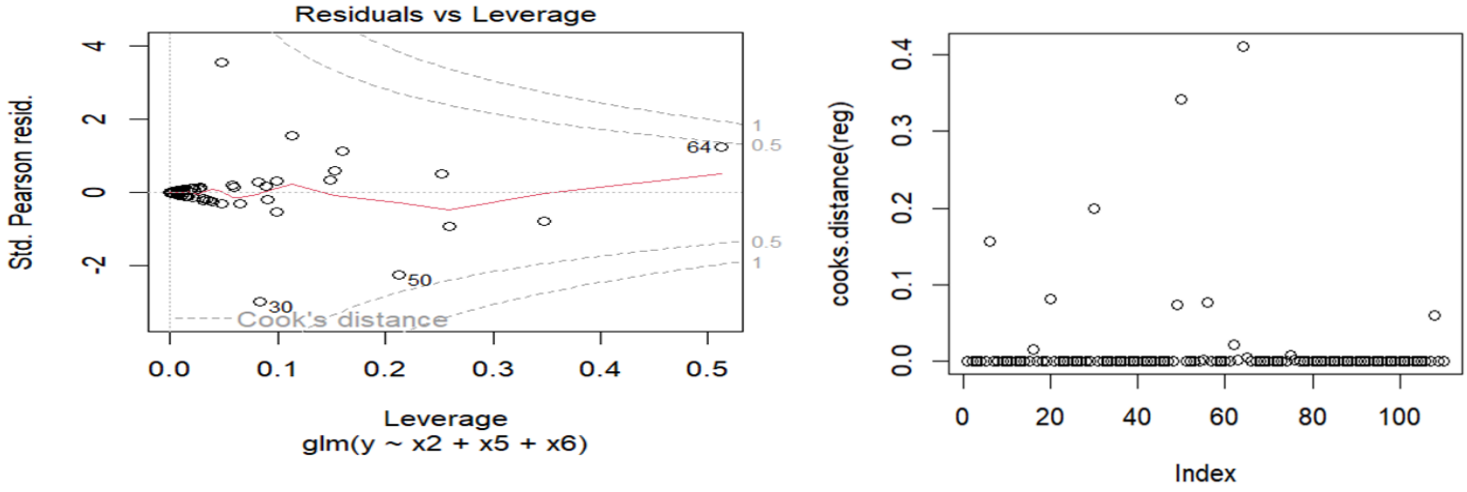
```
lag Autocorrelation D-W Statistic p-value
   1       0.09301242       1.813956   0.316
Alternative hypothesis: rho != 0
```

As p value > 0.05, we accept H0 that the there is no autocorrelation between residuals i.e. residuals are independent.

There is no pattern in graph which implies that residuals are independent.



4. No outlier effects



Observations with Cook's distance close to 0 and under 1 are considered to have low influence on the regression model.
Observations that fall above the threshold line (typically at Cook's distance = 1) are considered potentially influential and may warrant further investigation.
Hence from above plot of cook's distance of regression model, we observe low or no outlier effects.

Interpretation of results

From the R output we observe that,

$$\beta_0 = -10.9875, \beta_2=0.7079, \beta_5=0.4291, \beta_6= 2.5652$$

The results of this model emphasize the significance of education, farm size, insurance awareness variables in influencing the farmers' WTI in crop insurance.

1. **Age:**
   Age does not have any significant effect on farmers willingness to insure.

2. **Education:**
   The farmer's level of education has a positive and statistically significant ($p < 0.05$) relationship with the likelihood of adopting crop insurance, which implies that highly educated farmers are more likely to insure their farms. This result confirmed a prior expectation and is in line with the previous findings. Typically, farmers with a higher level of education are expected to be more aware of the benefits of agricultural insurance products which may also encourage early adoption.

   If the level of education increases by one year, the odds of farmers willing to enroll in crop insurance (i.e. Y=1) increases by factor e^0.7079= 2.029724359. Thus, the probability of farmers willingness to insure increases by 2.0297 times with unit increment in education level.

3. **Livestock:**
   Livestock owned does not have any significant effect on farmers willingness to pay.

4. **Income:**
   Income should have some effect on farmers enrollment as per previous study, but due to unavailability of large sample size here we don't observe any effect of income.

5. **Farm Size:**
   From the results it is found that total farm size owned by farmer has positive effect on WPI. Odds of WPI increases by factor e^0.4291 = 1.5358874614. Probability of farmers willingness to enroll increases by 53.58% with unit increase in farm size i.e. farmer having more farm under production would like to get insured while farmers with less land may be disinclined to insure their crops.

6. **Insurance Awareness:**
   From the results it is found that farmers awareness about insurance scheme has positive effect on WPI at 95% CI. The odds of WPI increases by factor of e^2.5652= 13.00325876. Probability of farmers willingness to enroll increases by 13% if they are aware about crop insurance schemes.

7. **Loan access:**
   We don’t observe any significant effect of loan access to farmers in their willingness to insure.

8. **Household size:**
   We don’t observe any significant effect of household size of farmers in their willingness to insure.

Enrollment Model

Determining influence of different macro-economic factors on aggregate enrollment in crop insurance schemes.

This study aims to determine the influence of macroeconomic factors on enrollment of farmers under crop insurance schemes in India. An understanding of their relationship will provide better insights towards the growth of the crop insurance sector.

Power Analysis

A power analysis is the calculation used to estimate the smallest sample size needed for an experiment, given a required significance level, statistical power, and effect size.
Effect size provides a quantitative measure of the strength or magnitude of the relationship between variables in a statistical analysis.

Here, we use Cohen's d effect size (for continuous variables):

- Small Effect Size: $d \approx 0.2$
- Moderate Effect Size: $0.2<d<0.5$
- Large Effect Size: $d \geq 0.5$

Let us estimate the minimum sample size for glm using R.
Let no. of predictors, k = 2 (including intercept),
level of significance, $\alpha = 0.10$,
power$(1-\beta) = 0.8$, and
effect size, $f 2 = 0.35$, where 0.35 is moderate effect size suggested by Cohen (1988).

A moderate effect size indicates a meaningful or substantial relationship between variables, suggesting that the observed effect is not trivial but is also not extremely large.

Here, $u = k - 1$, $v = n - u - 1$, where n is the no. of observations. Thus, n = 19.
For 10% level of significance, 80% power and 0.35 effect size, **the minimum sample size is 19.**

Descriptive Analysis of Data

The secondary data has been collected year wise from 2001 to 2020 (sample size = 20).

Y	X1	X2	X3	X4	X5
No. of farmers insured (in lakhs) under crop insurance schemes	FDI equity inflows in agriculture services sector (US$ million)	Agricultural Exports (in '000 Rs. Crore)	Area of Foodgrains in Million Hectares	Production of foodgrains in Million Tonnes	Yield of foodgrains in '00 Kg./Hectare

Abb: FDI=Foreign Direct Investment

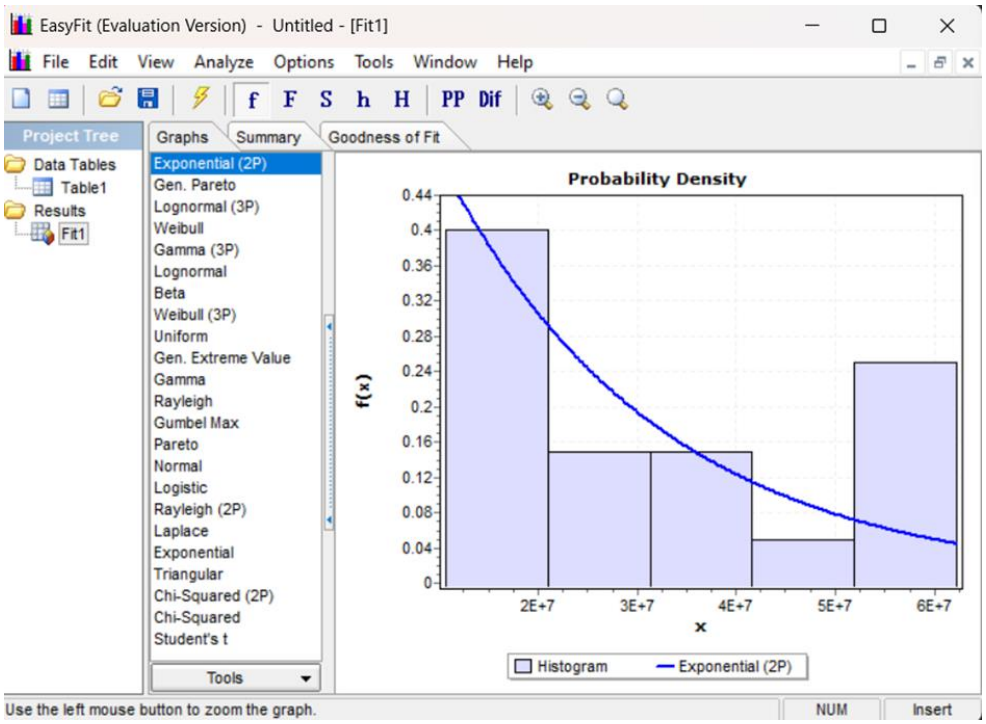
First 10 rows of the data collected:

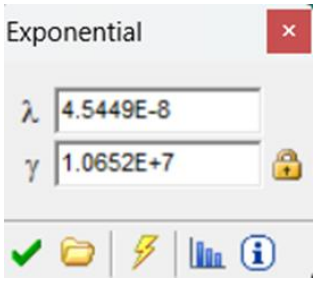
Y	X1	X2	X3	X4	X5
106.5202	14.06	29.72861	122.78	212.85	17.34
120.9552	11.01	34.65394	113.86	174.77	15.35
123.9212	0.59	36.41548	123.45	213.19	17.27
162.1825	3.83	41.60265	120.08	198.36	16.52
167.2236	9.08	45.71097	121.6	208.6	17.15
179.1204	12.53	57.76787	123.71	217.28	17.56
191.138	58.13	74.67348	124.07	230.78	18.6
195.3601	5.35	81.06452	122.85	234.47	19.09
259.4179	1222.22	84.44395	121.34	218.11	17.98
273.0883	43.9	113.0466	126.67	244.5	19.3

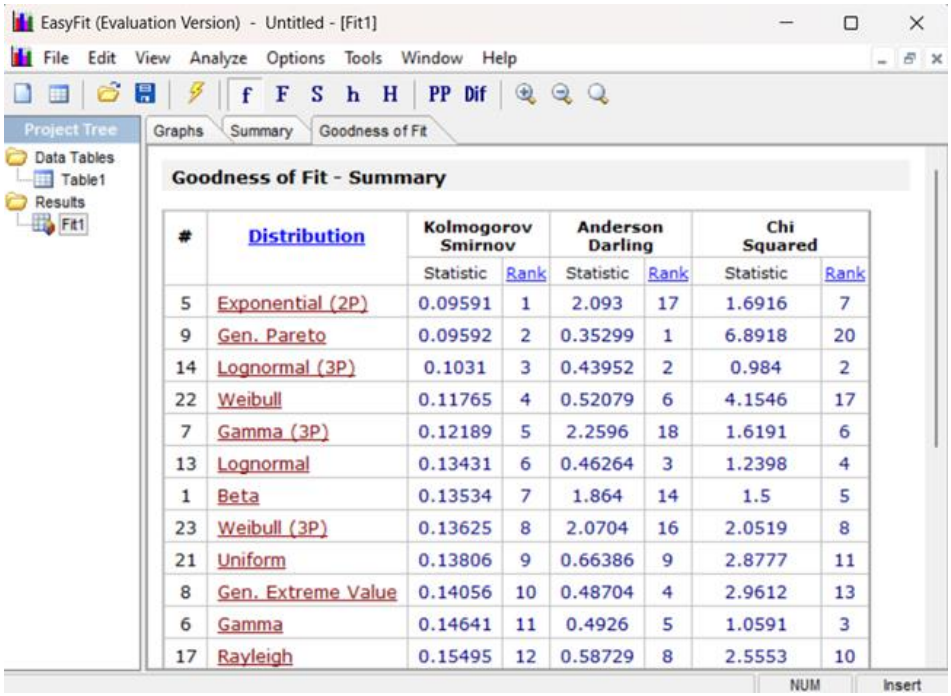
```
> summary(df)
       y                x1                x2               x3       
 Min.   :106.5   Min.   :   0.59   Min.   : 29.73   Min.   :113.9  
 1st Qu.:176.1   1st Qu.:  12.15   1st Qu.: 54.75   1st Qu.:122.5  
 Median :285.0   Median :  55.16   Median :147.92   Median :123.9  
 Mean   :326.5   Mean   : 113.57   Mean   :152.08   Mean   :123.8  
 3rd Qu.:497.3   3rd Qu.:  89.32   3rd Qu.:242.65   3rd Qu.:125.4  
 Max.   :622.4   Max.   :1222.22   Max.   :308.83   Max.   :129.8  
       x4               x5        
 Min.   :174.8   Min.   :15.35  
 1st Qu.:216.3   1st Qu.:17.50  
 Median :248.0   Median :19.79  
 Mean   :244.6   Mean   :19.70  
 3rd Qu.:267.6   3rd Qu.:21.29  
 Max.   :310.7   Max.   :23.94
```

Distribution of Response variable

Using EasyFit to find distribution of response variable







#	Distribution	Kolmogorov Smirnov		Anderson Darling		Chi Squared	
		Statistic	Rank	Statistic	Rank	Statistic	Rank
5	Exponential (2P)	0.09591	1	2.093	17	1.6916	7
9	Gen. Pareto	0.09592	2	0.35299	1	6.8918	20
14	Lognormal (3P)	0.1031	3	0.43952	2	0.984	2
22	Weibull	0.11765	4	0.52079	6	4.1546	17
7	Gamma (3P)	0.12189	5	2.2596	18	1.6191	6
13	Lognormal	0.13431	6	0.46264	3	1.2398	4
1	Beta	0.13534	7	1.864	14	1.5	5
23	Weibull (3P)	0.13625	8	2.0704	16	2.0519	8
21	Uniform	0.13806	9	0.66386	9	2.8777	11
8	Gen. Extreme Value	0.14056	10	0.48704	4	2.9612	13
6	Gamma	0.14641	11	0.4926	5	1.0591	3
17	Rayleigh	0.15495	12	0.58729	8	2.5553	10

The top five distributions of response variable are exponential, gen. pareto, lognormal. Weibull and Gamma. Since, the exponential distribution is a particular case of gamma distribution, **we consider the response variable to follow Gamma distribution. This is supported by continuous and non-negative nature of the response variable, and also by its positive skewness** calculated as follows using the summary statistics:

Bowley's coefficient of skewness = (Q3 + Q1 - 2Med) / (Q3 - Q1)
i.e., SKB = (497.3 + 176.1 – 2*285)/(497.3 – 176.1) = **0.3219**

Since SKB is greater than 0, it implies that Y is positively skewed.

Generalized Linear Model

Generalized linear models (GLMs) relate the response variable that we want to predict to the explanatory variables about which we have information. A GLM takes multiple regression one step further by allowing the data to be non-normally distributed.

A GLM consists of three components:

i. A distribution for the response variable
ii. A 'linear predictor' (a function of the predictors linear in the parameters)
iii. A 'link function' (that connects the mean response to the linear predictor)

If Y is response variable and X is predictor, then,

$$g(\mu) = \eta$$

where $\mu = E(Y)$, $\eta = \beta 0 + \beta 1 * X$, g(.) is the link function.

Syntax: model = glm(Y ~ …, family = … (link = …))

For our model, we use family = Gamma and link = log.

In the context of a gamma GLM, both the log link function and the inverse link function are commonly used, but we will use the log link function to ensure that predicted values are positive.

Choosing the best model

Using AIC

The Akaike Information Criterion (AIC) is a measure of relative quality of statistical models for a given set of data.
The preferred model is the one with minimum AIC value.

Using McFadden's Pseudo R-Squared

We use the following formula to calculate McFadden's R-Squared:

McFadden's R-Squared = 1 – (Residual deviance / Null deviance)

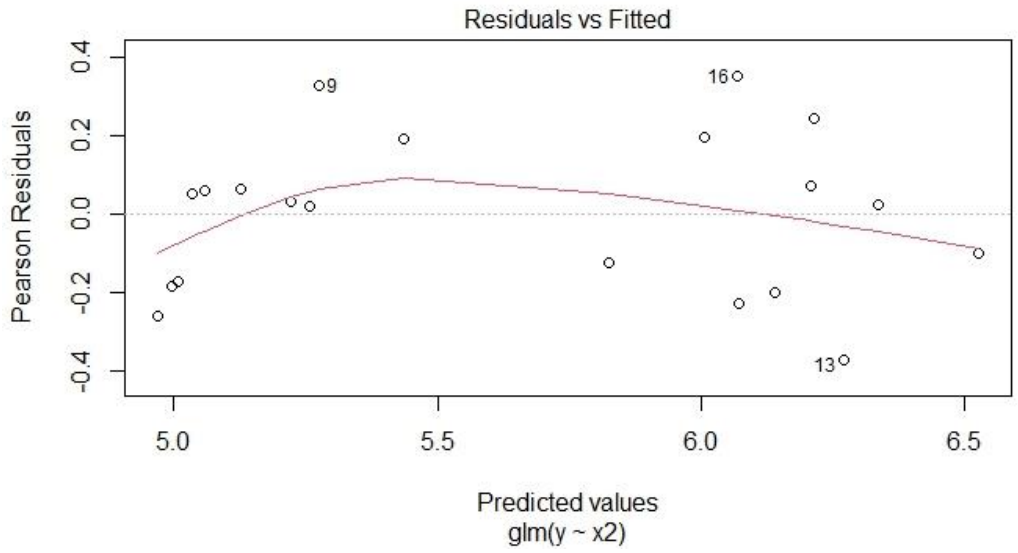
McFadden's R-Squared ranges from 0 to 1, with higher values indicating a better model fit. The deviance statistic plays the same role that the residual sum of squares plays for OLS.

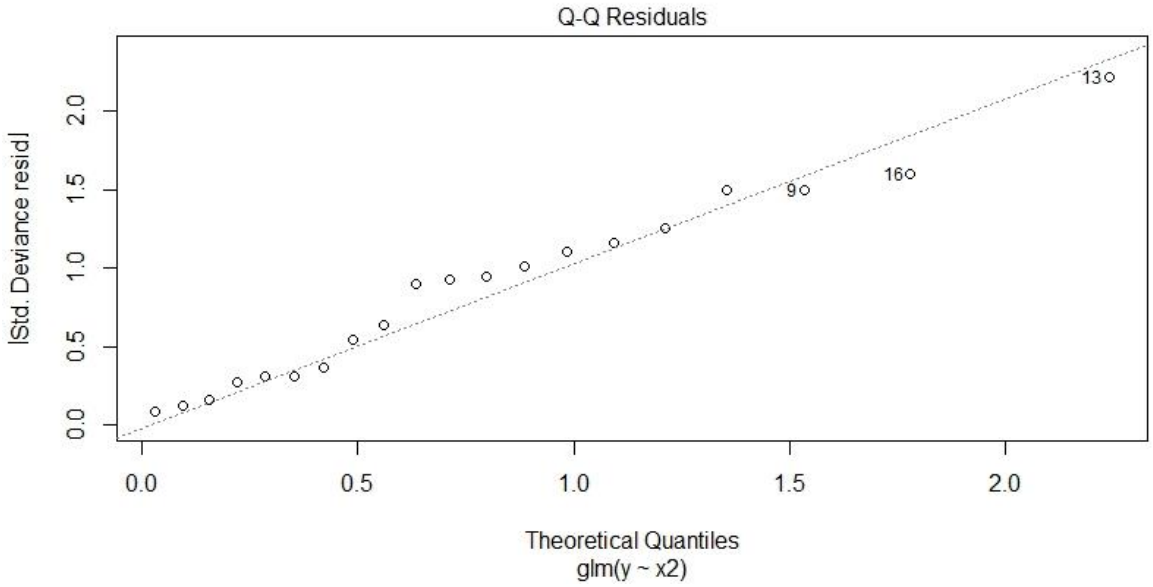
The preferred model is one with maximum R squared value.

Validating assumptions

The assumptions of a GLM require that the residuals should show no patterns.

Since the residuals show no significant pattern in the residuals vs. fitted plot, the assumption of independence of residuals is satisfied. The Q-Q plot also validates the model, but there has been some debate about whether this is appropriate for non-normal distributions.





Conclusion

Here, we have limited our study to only five factors, but many other factors such as GDP, inflation rates, interest rates, government policies, etc. can be included.

We only use one regressor here for given sample size (since sample size cannot be increased due to unavailability of annual data of farmers insured before year 2000 and after year 2020).

The use of log link function here gave smaller AIC values and higher R squared values than those obtained when using the inverse link function, indicating a better fit.

Model No.	Model	AIC	McFadden's R2
1	Y ~ X1	265.7831	0.0012
2	Y ~ X2	223.8027	0.8719
3	Y ~ X3	252.1458	0.4822
4	Y ~ X4	228.3073	0.8398
5	Y ~ X5	226.8151	0.8512

Based on minimum AIC and maximum R squared, **model2 (y ~ x2) is the best model**.

Hence, the annual aggregate enrollment of farmers (in lakhs) under crop insurance schemes is best explained by the **agricultural exports (in '000 Rs. Crore)** in India.

The required equation of GLM is,

$$\log(E(Y)) = 4.8047717 + 0.0055744 * X2$$

Thus, E(Y) = e^(4.8047717 + 0.0055744*X2). The odds ratio for a change of 10 units (i.e., 10,000 crores) is e^(10*0.0055744) = 1.05733.

Hence, an increase in agricultural exports by 10 units will result in 5.733% increase in no. of farmers insured.

Optimizing Drought Index Selection

Optimizing Drought Index Selection with Lasso Regression to predict detrended crop yield.

Lasso Regression Model

Lasso regression, short for **Least Absolute Shrinkage and Selection Operator**, is a linear regression technique that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces.

It achieves this by adding a penalty term to the ordinary least squares objective function, which penalizes the absolute size of the regression coefficients.

The objective function of Lasso regression can be written as:

$$\text{minimize} \left(\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right)$$

Here, y_i is the response variable (detrended yield in your case), x_{ij} are the explanatory variables (SPI, RAI, MZCI, and statistical z-scores), β_j are the coefficients, p is the number of predictors, and λ is the regularization parameter.

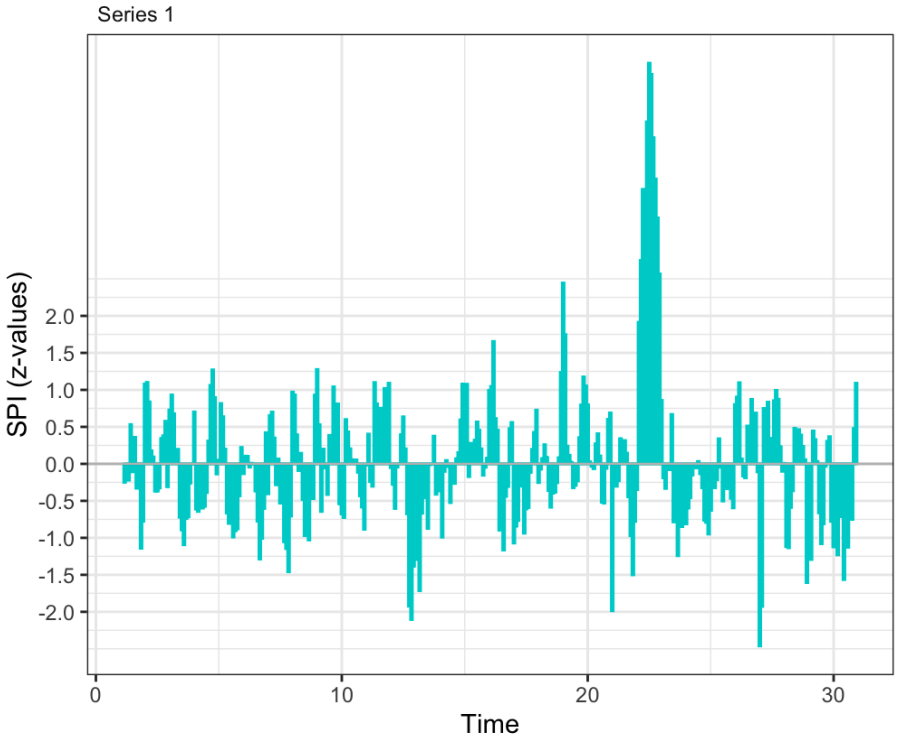
Drought indices (used as predictors)

SPI (Standardised Precipitation Index)

The **SPI (Standardized Precipitation Index)** is a tool that measures the precipitation deficit over various time periods. These time periods help determine the impact of drought on the availability of various water resources. The SPI for a particular location is based on the long-term precipitation record for a specific period. This long-term record is then fitted to a probability distribution, such as the gamma distribution. This probability distribution is then transformed into a normal distribution function, which ensures that the mean SPI for the location and the desired period is zero.

Year	SPI
1981	-0.250843203
1982	-0.301137933
1983	-0.50881358
1984	0.029768173
1985	-0.910197841
1986	-0.855492342
1987	-0.968753874
1988	-0.318219871
1989	0.226440012
1990	0.033462668
1991	0.337364376
1992	-0.741772789
1993	-0.28961099
1994	0.195844929
1995	-0.100036908
1996	-0.535248318
1997	-0.326059356
1998	-0.005252559
1999	0.230609647
2000	-0.162583174
2001	2.317486343
2002	3.26039196
2003	-0.703281052
2004	-0.569687839
2005	-0.127334953
2006	0.536836945
2007	0.463219692
2008	-0.177415259
2009	-0.656185274

SPI	SPI Category
≥ 2.00	Extremely wet
$1.50 - 1.99$	Severely wet
$1.00 - 1.49$	Moderately wet
$0 - 0.99$	Mildly wet
$-0.99 - 0$	Mildly drought
$-1.49 - -1.00$	Moderately drought
$-1.99 - -1.50$	Severely drought
≤ -2.00	Extremely drought



RAI (Rainfall Anomaly Index)

Rainfall Anomaly Index (RAI) is a simple hydro-climatic index used to estimate wetness and dryness conditions associated with climatic change. RAI only requires precipitation data.

RAI	RAI Category
≥ 4.00	Extremely rainy
$3.00 - 3.99$	Highly rainy
$2.00 - 2.99$	Moderately rainy
$0.50 - 1.99$	Low rainfall
$-0.49 - 0.49$	Normal
$-1.99 - -0.50$	Slight reduction in rainfall
$-2.99 - -2.00$	Moderate reduction in rainfall
$-3.99 - -3.00$	Large reduction in rainfall
≤ -4.00	Extreme reduction in rainfall

Statistical z-score

Z-scores are a straightforward way to calculate drought. To get the Z-score, subtract the long-term mean from the monthly rainfall value and then divide the difference by the standard deviation. Z-scores don't require fitting gamma distribution or Pearson type III method. They are widely used in many drought studies due to their simplicity and effectiveness.

Z – SCORE	CONDITION
No Drought	>0.25
Weak Drought	0.25 to -0.25
Slight Drought	-0.25 to - 0.52
Moderately Drought	-0.52 to - 0.84
Severely Drought	-0.84 to - 1.25
Extremely light	< -1.25

Modified China Z-index (MZCI)

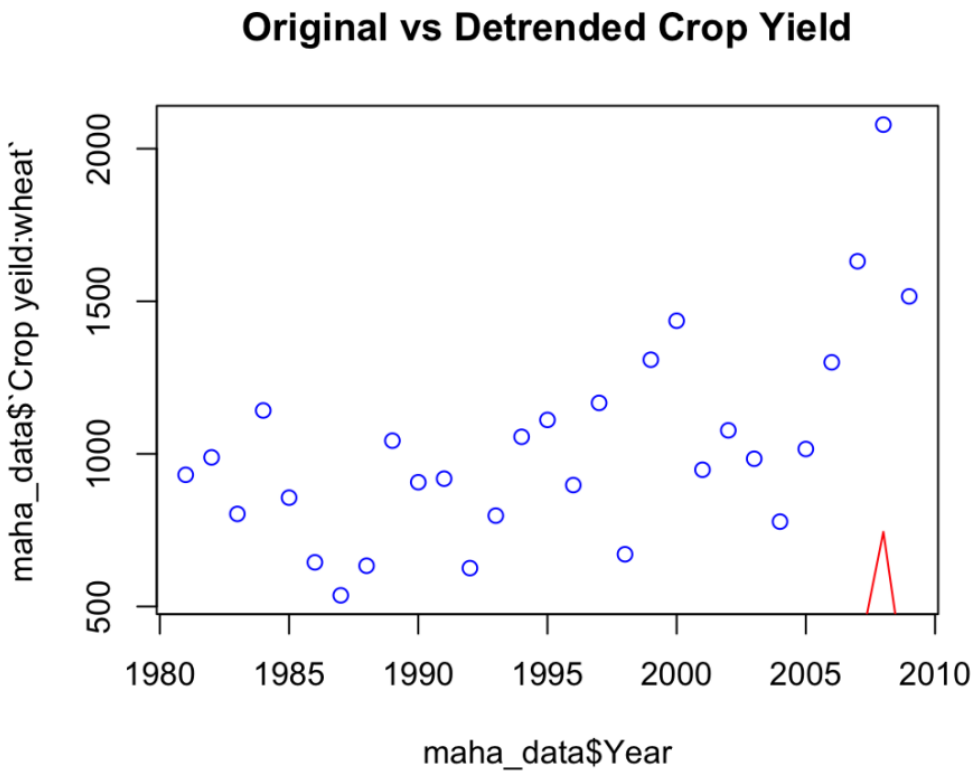
The CZI method utilizes the Wilson-Hilferty cube root transformation, as proposed by Kendall and Stuart in 1977. It is assumed that precipitation data follows the Pearson type III distribution. The primary distinction between CZI and MCZI is that MCZI employs the median as opposed to the mean.

CZI/MCZI	CONDITION
2.0+	Extremely wet
1.5 to 1.99	Very wet
1.0 to 1.49	Moderately wet
-0.99 to 0.99	Near normal
-1.0 to -1.49	Moderately dry
-1.5 to -1.99	Severely dry
-2 and less	Extremely dry

Detrended crop yield (used as response variable)

Annual crop yield data of wheat crops in Maharashtra have been collected for years 1982-2009 from CEIC.

Detrending data involves removing long-term trends or patterns unrelated to the predictors of interest, particularly those associated with time, to focus on the short-term variations and fluctuations in the data.



Conclusion

Using R-software to fit the lasso regression model, we observe that the coefficients for SPI are consistently non-zero across a wide range of lambda values.

This suggests that the **"SPI" (Standardized Precipitation Index)** variable is the most important predictor of detrended crop yield in the model.

Claim Size Modelling Using Multiple Regression

Claim size is the sum that the insurer pays when an insured event occurs. Here, the insured event in crop insurance is crop loss due to drought, flood, pests, etc.

Claim size modelling is important because it can help avoid underestimating the risk of large losses.

Variables

Response Variable : Number of Farmers Insured (Number of Claims)
Regressors : Area Insured, Sum Insured, Rainfall, Farmers Premium

Descriptive Analysis of Data

The secondary data has been collected state-wise since year from 2016 to 2021. Here, sample size = 131.

The first 10 rows of data are given below:

STATE	YEAR	Reported Claims(Rs in Cr.) Y	Farmer Applications Enrolled (In Lakhs) X1	Gross Premium Collected(Rs in Cr.) X2	Paid Claims X3	Rainfall X4	Sum Insured(in Crores) X5	Area Insured(Ha) X6
ANDHRA PRADESH	2016	944.3	17.8	803.6	944.3	760.4	5479.31	785.88328
ASSAM	2016	5.4	0.6	8.6	5.4	2140.5	30.11	4312.95
BIHAR	2016	347.8	27.1	1416	347.8	1158	11721.27	2465249.21
CHHATTISGARH	2016	160	15.5	289.3	160	1315.8	7212.51	2414754.7
GOA	2016	0	0	0.1	0	3065.1	5.8	548.14
GUJARAT	2016	1267.2	19.8	2274.6	1267.2	604.9	12323.75	2841629.65
HARYANA	2016	298.1	13.4	363.4	298.1	392.9	11782.93	2084575.87
HIMACHAL PRADESH	2016	45.3	3.8	71.7	45.3	921.5	371.35	85416.19
JHARKHAND	2016	31.1	8.8	271.4	31.1	1264	2010.49	375725.83
KARNATAKA	2016	2093.8	29.5	1332.7	2093.8	849.9	10410.46	4455065.7

```
> summary(df)
   enrollment      Paid_claims      Reported_Claims    sum_assured        area_insured
 Min.   :  0.00   Min.   :   0.00   Min.   :   0.00   Min.   :    0.05   Min.   :    0.01
 1st Qu.:  0.60   1st Qu.:   1.35   1st Qu.:   3.65   1st Qu.:   75.24   1st Qu.:   11.92
 Median : 13.00   Median : 160.00   Median : 298.10   Median : 4926.47   Median :  674.12
 Mean   : 26.17   Mean   : 908.16   Mean   : 943.68   Mean   : 8605.28   Mean   : 2366.42
 3rd Qu.: 39.55   3rd Qu.:1141.55   3rd Qu.:1170.30   3rd Qu.:12053.34   3rd Qu.: 2370.13
 Max.   :183.80   Max.   :6758.6    Max.   :7494.20   Max.   :50167.84   Max.   :22111.20
    premium          Rainfall
 Min.   :   0.0   Min.   : 351.8
 1st Qu.:   9.8   1st Qu.: 904.6
 Median : 289.3   Median :1253.5
 Mean   :1149.9   Mean   :1536.8
 3rd Qu.:1451.5   3rd Qu.:1793.5
 Max.   :7180.5   Max.   :5649.1
```

Fitting of Multiple Linear regression

Model:

A multiple linear regression model relating p regressors to response variable Y can be expressed as:

$$Y = \beta0 + \beta1*X1 + \beta2*X2 + \ldots + \beta p*Xp + \varepsilon$$

Where, β0, β1, β2, …, βp are regression coefficients and ε is random error.

Assumptions:

i. Errors are independently and normally distributed with E(ε) = 0, Var(ε) = σ^2.
ii. Measurements on regressors are without error or with negligible error.

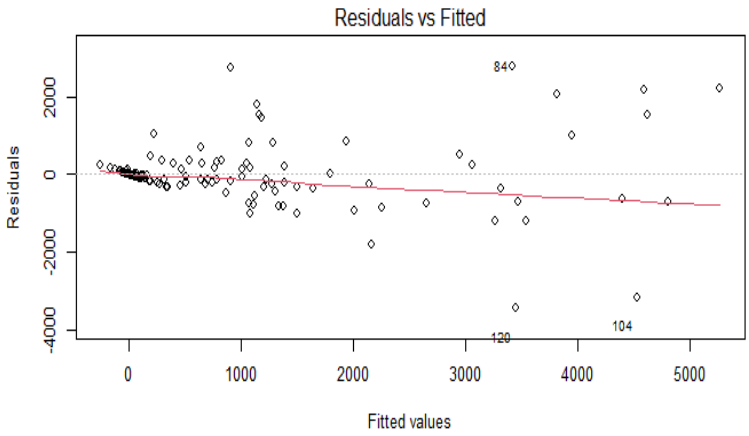
The best fit of the model obtained using R is **Y ~ X1 + X2 + X4 + X5 + X6**, that is, the model equation is given by:

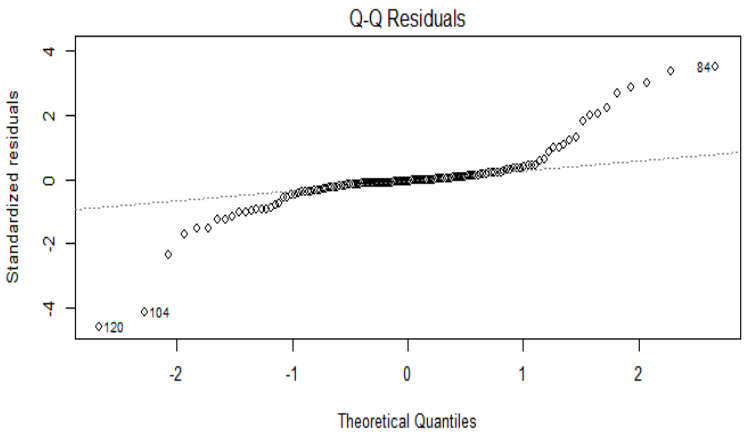
Y = 163.180585 + 3.139060*X1 + 0.554380*X2 - 0.072670*X4 - 0.004316*X5 + 0.088612*X6

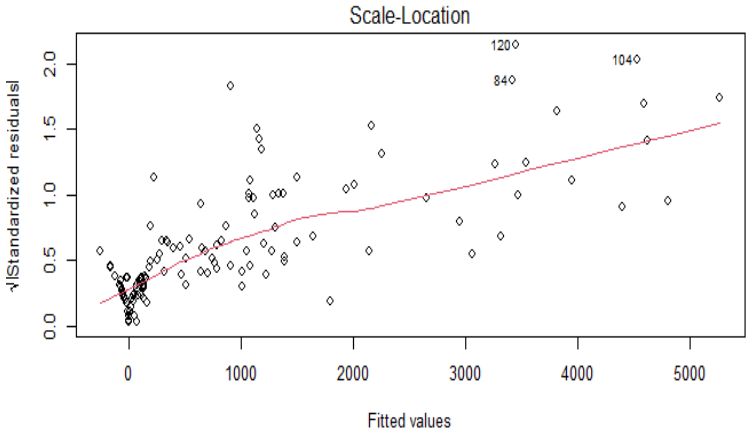
Where Y = Reported claims (in Rs. Crore)
X1 = Farmers enrolled (in lakhs)
X2 = Gross Premium Collected (in Rs. Crore)
X4 = Rainfall
X5 = Sum Insured (in Rs. Crore)
X6 = Area Insured (in Ha)

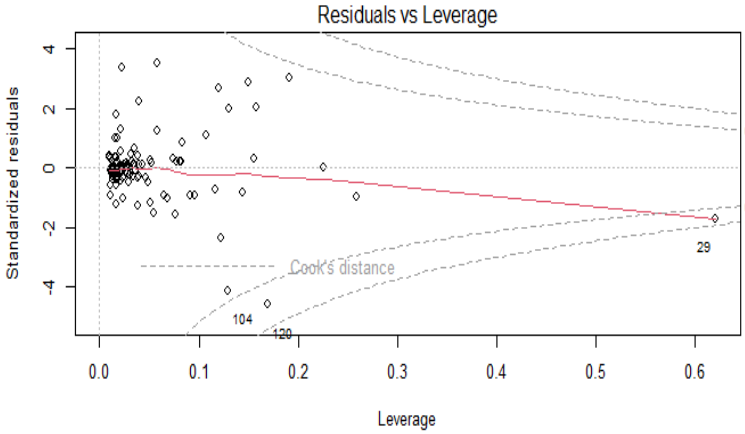
Multiple R-squared = 0.7181

Residual plots:









Conclusions:

- Farmers' enrollment, Gross premium collected, and Area insured have a positive relationship with Claim size.
- Rainfall and sum insured have a negative relationship with claim size. This indicates that the claim size increases with increase in drought conditions.
- 71.81% of variation in the data is explained by the model. Hence, the model can be considered a good fit.
- From the residual plots, the assumption of constant variance is not validated, hence an appropriate transformation may be necessary.

Claim Frequency Modelling using Time Series Analysis

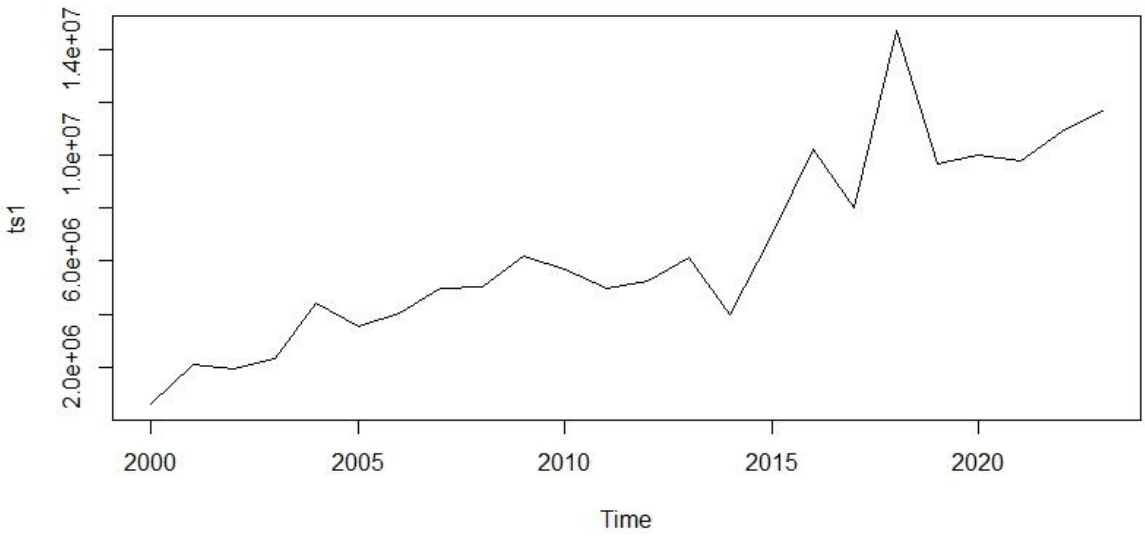
A **time series** is a series of data points indexed or listed in time order. It can be continuous trace or discrete set of observations. Here we are dealing with observations taken at discrete time periods. By appropriate choice of origin and scale we can take the time periods to be 1, 2, and so on.

Analysis of data:

Analysis will be done on the claims data of NAIS and PMFBY schemes from years 2000 to 2023 (rabi season).

The first 10 rows of data are:

year	claim
2000	579940
2001	2091733
2002	1955431
2003	2326811
2004	4421287
2005	3531045
2006	4048524
2007	4977980
2008	5044016
2009	6210648



ARIMA

Here for forecasting we have used the technique: ARIMA Modelling.
ARIMA stands for Autoregressive Integrated Moving Average.

The AR part of ARIMA indicates that the evolving variable of interest is regressed on its own lagged values.
The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past.

ARIMA Models are generally denoted by ARIMA(p,d,q) where parameters p, d and q are non-negative integers:
p= order of autoregressive model
q= order of the moving average model
d= degree of differencing

Stationarity

In general, stationarity implies a type of statistical equilibrium or stability in data. A time series is said to be strictly stationary if its properties are not affected by change in time origin.

In other words, if joint probability distribution of observations Yt , Yt+1, ..., Yt+h is exactly same as the joint probability distribution of Yt+k , Yt+k+1, ..., Yt+h+k.

A time series is said to be weakly stationary if it has the property that the **mean, variance and autocorrelation structure do not change over time**.

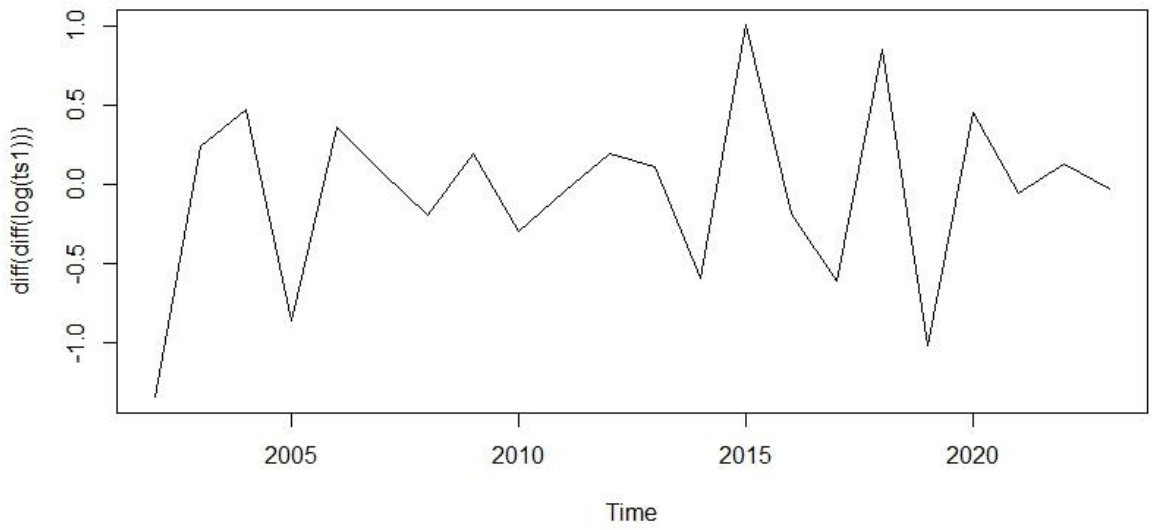
The main assumption for ARIMA is that the time series is stationary. But from the above plot, we can clearly observe a trend in the data, that is, our series is not stationary.

We employ the following methods to make the series stationary:

Log Transformation:
Log Transformation can be used to stabilize the variance of a series with non-constant variance.

Differencing:
Differencing series is the change between consecutive data points in the series.
First order differencing is Yt’ = Yt – Yt-1.
Second order differencing is Yt’’ = Yt’ – Yt-1’ = Yt – 2Yt-1 + Yt-2.

After applying log transformation and second-order differencing, our time series becomes stationary:



Test to check stationarity

ADF (Augmented Dickey Fuller) Test
It can be used to determine the presence of unit root in the series, and hence help us understand if the series is stationary or not. The null and alternate hypothesis of this test are:

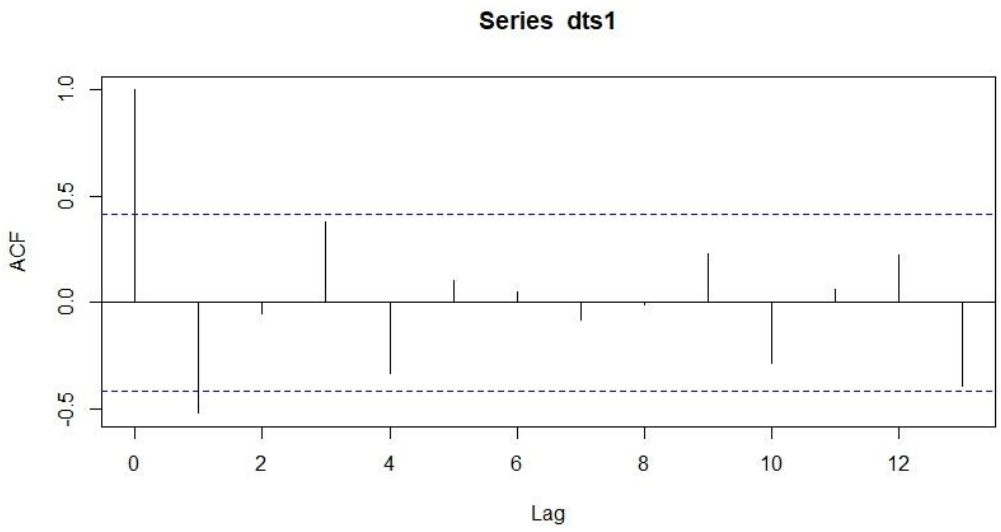
ADF test is conducted with the following assumptions:

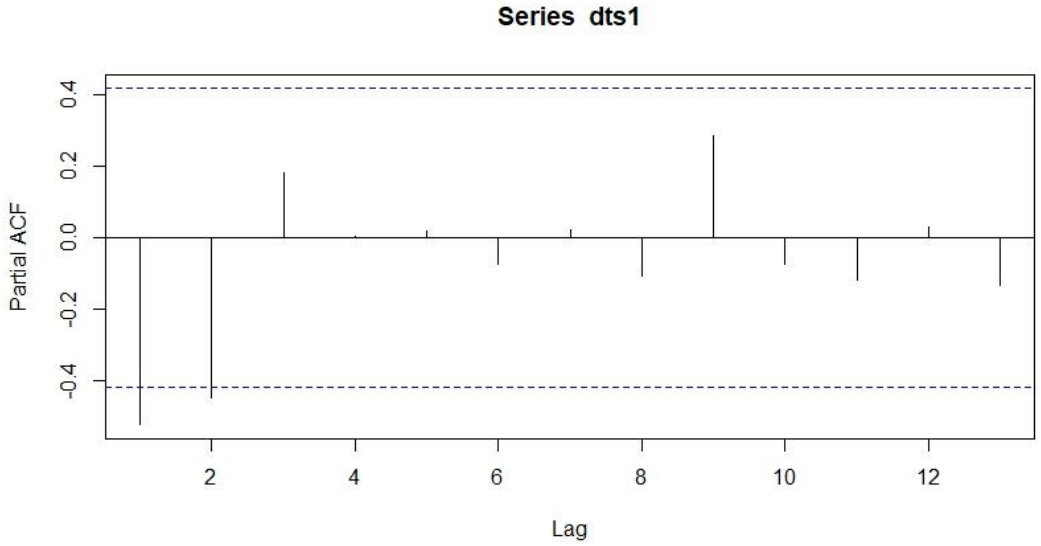
- Null Hypothesis (HO): Series is non-stationary, or series has a unit root.
- Alternate Hypothesis(HA): Series is stationary, or series has no unit root.

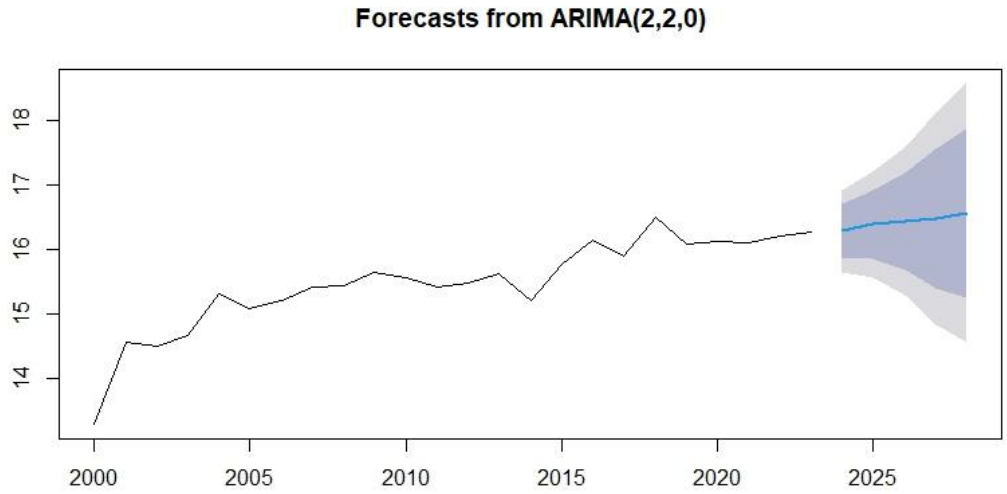
If the null hypothesis is failed to be rejected, this test may provide evidence that the series is non-stationary.

Using adf.test in R, we get the p-value as 0.0406, i.e. the transformed series is stationary. Hence, we continue modelling on this series.

Fitting of ARIMA model







Using the *auto.arima* function in R, we get the best model for our data as **ARIMA(2,2,0) model**. Here, no MA term is required in the model.

We get the next 5 years forecast as:

Year	No. of claims
2024	11825892
2025	13138764
2026	13671374
2027	14336617
2028	15667920

Portmanteau Test

Portmanteau tests (including *Box-Pierce* test or *Ljung-Box* test (more accurate)) are used to see if the model trained is ready for inferencing and forecasting meaning that the model has been able to capture all of the trend and seasonality within a reasonable threshold or not. In other words, they test if the residuals of the trained model is a white noise (i.e. normal distribution (0, 1)).

To test:

H0: The residuals are independently distributed.

HA: The residuals are not independently distributed; they exhibit serial correlation.

Using the **Ljung- Box test** in R, we get the **p-value = 0.3876.** Hence, we accept the null hypothesis at 5% l.o.s.

Conclusion

- The time series is made stationary using log transformation and differencing, which is validated using the ADF test.
- ARIMA(2,2,0) is the best model and we have forecasted the next 5 years no. of claims.
- The residuals have no significant autocorrelation. Thus, we conclude that the model does not exhibit significant lack of fit.

Premium Calculations using Actuarial Statistics

Actuarial statistics is important in the insurance sector, especially when calculating premiums. These premiums are the financial backbone of insurance firms, reflecting the money collected from policyholders to cover potential future claims and costs.

Actuarial statistics provides the mathematical basis and analytical tools required to assess risk, set appropriate premium levels, and assure insurers' financial stability.

Data : The following premium calculations have been done using the figures quoted in the Pradhan Mantri Fasal Bima Yojana (PMFBY) 2016.

Assumptions:
1. The sum assured to each farmer was Rs.30,000.
2. The interest rates have been taken according to the bank rates varying with time.
3. A single premium is charged from the farmers under this scheme.

Formula

The formula used to calculate premiums is:

$$\boldsymbol{PV\ of\ inflow = PV\ of\ outflow}$$

$$\boldsymbol{P = X/\ddot{a}_{\overline{n}|}}$$

where,
P: one time premium
X: sum assured
$\ddot{a}_{\overline{n}|}$: annuity payable in advance for n years

The scheme is functional for 6 months for each type of crop therefore the formula is used as follows:

$$P = 30000/(12 * \ddot{a}^{(12)}_{\overline{0.5}|}) \qquad calculated\ @d^{(12)}$$

$$P = 30000/(12 * \frac{1-v^n}{d^{(12)}})$$

where,

v : present value factor , calculated using the formula $v = (1+i)^{-1}$

$d^{(12)}/12$: monthly discount factor calculated using the formula $(1+i)^{-1} = \left(1 - \frac{d^{12}}{12}\right)^1$

i: bank rate

year	sum assured	interest rate	interest rates	discount rates	present value factor	monthly discount rate	annuity	premium
2016	30000	6.40%	0.064	0.06015038	0.93984962	0.00515628	5.92318555	5064.84218
2017	30000	6.15%	0.0615	0.05793688	0.94206312	0.00496124	5.92607191	5062.37529
2018	30000	6.29%	0.0629	0.05917772	0.94082228	0.00507052	5.92445446	5063.75738
2019	30000	5.62%	0.0562	0.05320962	0.94679038	0.0045461	5.93222046	5057.12831
2020	30000	4.26%	0.0426	0.04085939	0.95914061	0.00347043	5.9481838	5043.55632
2021	30000	4.00%	0.04	0.03846154	0.96153846	0.00326306	5.95126657	5040.94375
2022	30000	4.98%	0.0498	0.04743761	0.95256239	0.00404178	5.939699	5050.76099
2023	30000	6.50%	0.065	0.06103286	0.93896714	0.00523415	5.92203347	5065.8275
2024	30000	6.50%	0.065	0.06103286	0.93896714	0.00523415	5.92203347	5065.8275

However, in actual practice only **2% of this premium is paid by the farmers for kharif crops** and **1.5% of this premium is paid by the farmers for rabi crops** and the remaining percentage is paid by government subsidy.

premium	**premium for kharif crops**	**premium for rabi crops**
5064.842	101.2968436	75.97263273
5062.375	101.2475057	75.9356293
5063.757	101.2751477	75.95636077
5057.128	101.1425662	75.85692463
5043.556	100.8711264	75.65334483
5040.944	100.8188749	75.6141562
5050.761	101.0152199	75.76141491
5065.827	101.3165499	75.98741242
5065.827	101.3165499	75.98741242

Conclusion

- We have employed actuarial techniques to predict the one time premium to be paid by farmers for half-yearly crop insurance.

Limitations

- Obtaining comprehensive historical data on crop yields, weather patterns, pest infestations, and other relevant factors is challenging. Most of the data was available only for certain regions or time periods, limiting the scope and accuracy of the analysis.

- Even when data is available, its quality may vary. Inaccuracies, inconsistencies, and missing values can affect the reliability of the analysis and conclusions drawn from it.

- Due to lack of granularity, especially when it comes to detailed information about localized factors such as soil types, irrigation practices, and specific crop varieties, it can be difficult to assess risk accurately and tailor insurance products accordingly.

Scope

- To improve the adoption rate of crop insurance schemes in Maharashtra, empirical research on farmers' Willingness to Insure (WTI) in crop insurance is needed.
- A larger sample and more variety of macroeconomic factors in addition to the predictors used in the Enrolment model will lead to a more accurate model. The lognormal family can also be used in GLM.
- Ridge regression may be used for selection of best drought index to predict crop yield.
- The Joint Gamma-Poisson Distribution can be used to model average claim size and no. of claims.
- Monthly time series data of no. of claims may be considered for checking if seasonality is present.
- The premium calculations can be improved upon by further study of actuarial statistics.

Appendix (R-codes)

Willingness to Insure (WTI) Model

Logistic Regression model

```
> library(readxl)
> project2024 <- read_excel("C:/Users/Lenovo/Downloads/project2024.xls")
> View(project2024)
> y=project2024$WTP
> x1=project2024$Age
> x2=project2024$`Educational Years`
> x3=project2024$livestock
> x4=project2024$`Income(000's)`
> x5=project2024$`Farm size`
> x6=project2024$Awareness
> x7=project2024$`members in the family`
> x8=project2024$Hosehold size
 > reg=glm(y~x1+x2+x3+x4+x5+x6+x7+x8, family=binomial)

> summary(reg)
Call:
glm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8, family =
binomial)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -20.728266  19.938560  -1.040 0.299
x1          -0.108486   0.146361  -0.741  0.459
x2          2.153401   1.834045   1.174   0.240
x3          5.878977   4.895711   1.201   0.230
x4          0.003756   0.004296   0.874   0.382
x5          0.207924   0.305514   0.681   0.496
x6          3.168e+00  3.548e+03   0.001  0.999
x7                -5.435e+00  4.333e+03  -0.001 0.999
x8          -2.407281   2.725018  -0.883  0.377

(Dispersion parameter for binomial family taken to be 1)

      Null deviance: 146.2884  on 109  degrees of freedom

Residual deviance:   8.4715  on 101  degrees of freedom

AIC: 22.471

Number of Fisher Scoring iterations: 15
```

Using stepAIC function

```
> reg=glm(y~x1+x2+x3+x4+x5+x6+x7+x8, family=binomial)
> library(MASS)
> stepAIC(reg, direction = 'both')
```

```
Start:  AIC=16
y ~ x1 + x2 + x3 + x4 + x5 + x6 +x8
  Df Deviance     AIC
- x2  1     0.000  14.000
- x4  1     0.000  14.000
- x8  1     0.000  14.000
- x1  1     0.000  14.000
- x5  1     0.000  14.000
- x6  1     8.471  22.471
<none>      0.000  16.000
- x3  1  144.175 158.175

Step:  AIC=14
y ~ x1 + x3 + x4 + x5 + x6 + x8

      Df Deviance AIC
- x4  1     0.000 12.000
- x8  1     0.000 12.000
- x1  1     0.000 12.000
- x5  1     0.000 12.000
<none>      0.000 14.000
+ x2  1     0.000 16.000
- x3  1   13.614 25.614
- x6  1   19.125 31.125

Step:  AIC=12
y ~ x1 + x3 + x5 + x6 + x8

      Df Deviance AIC
- x8  1     0.000 10.000
- x1  1     0.000 10.000
- x5  1     0.000 10.000
<none>      0.000 12.000
+ x4  1     0.000 14.000
+ x2  1     0.000 14.000
- x3  1   17.006 27.006
- x6  1   23.166 33.166

Step:  AIC=10
y ~ x1 + x3 + x5 + x6

      Df Deviance AIC
- x1  1   0.0000  8.000
<none>      0.0000 10.000
+ x2  1   0.0000 12.000
+ x8  1   0.0000 12.000
+ x4  1   0.0000 12.000
- x5  1   8.6958 16.696
- x3  1  23.8628 31.863
- x6  1  26.7774 34.777

Step:  AIC=8
y ~ x3 + x5 + x6

      Df Deviance AIC
<none>      0.000  8.000
+ x1  1     0.000 10.000
+ x8  1     0.000 10.000
+ x4  1     0.000 10.000
```

```
+ x2  1     0.000 10.000
- x5  1    12.642 18.642
- x3  1    23.872 29.872
- x6  1    26.813 32.813

Call:  glm(formula = y ~ x3 + x5 + x6, family = binomial(link = "probit"))

Coefficients:
(Intercept)             x3          x5          x6
    -21.069       6.953       1.262       -10.737

Degrees of Freedom: 109 Total (i.e. Null);  106 Residual
Null Deviance:          146.3
Residual Deviance: 6.879e-09              AIC: 8
```

Using bestglm function

```
> library(bestglm)

> df=data.frame(x1,x2,x3,x4,x5,x6,x7,y)

> df

> bestmodel2= bestglm(df, family = binomial, IC = "BIC", t = "default",
CVArgs = "default", qLevel = 0.99, TopModels = 5, method = "exhaustive",
intercept = TRUE, weights = NULL, nvmax = "default",
RequireFullEnumerationQ = FALSE)

> bestmodel2

BIC
BICq equivalent for q in (0.493050055108639, 0.820384122657956)
Best Model:
            Estimate Std. Error   z value Pr(>|z|)

(Intercept) -14.1973228  5.2655267 -2.696278 0.007011916

x2          0.7219016  0.3903038  1.849589 0.064372826
x5          0.3253155  0.1373607  2.368330 0.017868591
x6          5.1556910  1.6425001  3.138929 0.001695667

> summary(bestmodel2)

Fitting algorithm:  BIC-glm
Best Model:
            df  deviance
Null Model 105  18.04562
Full Model 109 146.28838

            likelihood-ratio test - GLM

data:  H0: Null Model vs. H1: Best Fit BIC-glm

X = 128.24, df = 3, p-value < 2.2e-16
```

Fitting model using best predictors

```
> r=glm(y~x2+x5+x6,family=binomial)

> summary(r)

Call:
glm(formula = y ~ x2 + x5 + x6, family = binomial())

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.9875     3.5750  -3.073 0.002116 **
x2            0.7079     0.2767   2.559 0.010510 *
x5            0.4291     0.1479   2.902 0.003707 **
x6            2.5652     0.7000   3.664 0.000248 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

      Null deviance: 146.288  on 109  degrees of freedom
Residual deviance:  23.476  on 106  degrees of freedom
AIC: 31.476

Number of Fisher Scoring iterations: 8
```

Model Assumptions

```
> plot(reg)
> fitted=fitted(reg)

> plot(residuals,fitted)

> cor(residuals,fitted)

[1] 0.165449


> scatterplot(residuals,fitted)
> library(stats)

> cooks.distance(reg)
            1           2           3           4           5           6
      7           8           9
5.524901e-10 5.831478e-05 1.466470e-04 9.234121e-10 3.324695e-04
1.571912e-01 2.1019-09
..... ..... ..... ..... ..... .....
> plot(cooks.distance(reg))
> plot(reg)
```

Enrollment Model

Power Analysis

```
> pwr.f2.test(u = 1,v = NULL,f2 = 0.35,sig.level = 0.1, power=0.8)

     Multiple regression power calculation

              u = 1
```

```
              v = 17.17422
             f2 = 0.35
      sig.level = 0.1
          power = 0.8
```

GLM in R

R-code:

```
> model1 = glm(y~x1, family = Gamma(link = "log"), data = df)
> summary(model1)

Call:
glm(formula = y ~ x1, family = Gamma(link = "log"), data = df)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.779e+00  1.393e-01  41.478   <2e-16 ***
x1          8.623e-05  4.942e-04   0.174    0.863
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 0.3251901)

    Null deviance: 6.2422  on 19  degrees of freedom
Residual deviance: 6.2349  on 18  degrees of freedom
AIC: 265.78

Number of Fisher Scoring iterations: 7
```

We repeat the above code for rest of the predictors and get the following coefficients:

```
> model2$coefficients
(Intercept)          x2
4.804771710 0.005574414
> model3$coefficients
(Intercept)          x3
 -8.1298937   0.1117823
> model4$coefficients
(Intercept)          x4
 2.02017556  0.01487211
> model5$coefficients
(Intercept)          x5
  1.3252131   0.2198292

> model1$aic
[1] 265.7831
> model2$aic
[1] 223.8027
> model3$aic
[1] 252.1458
> model4$aic
[1] 228.3073
> model5$aic
[1] 226.8151
```

```
> with(summary(model1), 1 - deviance/null.deviance)
[1] 0.001166765
> with(summary(model2), 1 - deviance/null.deviance)
[1] 0.8719146
> with(summary(model3), 1 - deviance/null.deviance)
[1] 0.482191
> with(summary(model4), 1 - deviance/null.deviance)
[1] 0.8398264
> with(summary(model5), 1 - deviance/null.deviance)
[1] 0.8512535
```

Optimizing Drought Index Selection

Calculation of Drought Indices

SPI (Standardised Precipitation Index)

```
> #obtaining SPI values
> library(SPEI)
> SPI3=spi(rain_data$precp,3)
[1] "Calculating the Standardized Precipitation Evapotranspiration
Index (SPEI) at a time scale of 3. Using kernel type 'rectangular',
with 0 shift. Fitting the data to a Gamma distribution. Using the
ub-pwm parameter fitting method. Checking for missing values (`NA`):
all the data must be complete. Using the whole time series as
reference period. Input type is vector. No time information
provided, assuming a monthly time series."
> SPI3
  [1]           NA           NA -0.249479631 -0.040434507
  [5] -0.221221343  0.532872887 -0.108806735  0.360390066
  [9] -0.330715103 -0.283735377 -1.142020493 -0.776863826
 [13]  1.080623332  1.101154832  0.838945662  0.178047364
 [17]  0.094631300 -0.369023704 -0.368724094 -0.327657372
 [21]  0.343380413  0.379965474  0.575657453 -0.309111452
 [25]  0.728951545  0.932017298  0.678107085  0.024724004
(…83 more rows, total 360 values.)

> plot(SPI3)
```

RAI (Rainfall Anomaly Index)

```
> #Rainfall anomaly index
> # Load required library
> library(dplyr)
>
> # Calculating average monthly precipitation for each month
> avg_monthly_precip <- rain_data %>%
+   group_by(month) %>%
+   summarise(avg_monthly_prep = mean(precp))
>
> # Calculating average annual precipitation for each year
> avg_annual_precip <- rain_data %>%
+   group_by(year) %>%
```

```
+   summarise(avg_annual_prep = mean(precp))
>
> # Merging average monthly and annual precipitation data
> merged_data <- merge(rain_data, avg_monthly_precip, by = "month",
suffixes = c("", "_monthly")) %>%
+   merge(avg_annual_precip, by = "year", suffixes = c("",
"_annual"))
>
> # Calculate RAI for each month
> merged_data$RAI <- merged_data$avg_annual_prep -
merged_data$avg_monthly_prep
>
> # Calculate average RAI for each year
> avg_RAI_yearly <- merged_data %>%
+   group_by(year) %>%
+   summarise(avg_RAI = mean(RAI))
>
> # Create a data frame to store all the calculated values
> result_df <- data.frame(
+   Year = avg_RAI_yearly$year,
+   Avg_RAI = avg_RAI_yearly$avg_RAI,
+   Avg_Annual_Prep = avg_annual_precip$avg_annual_prep
+ )
> # View the resulting data frame
> print(result_df)
   Year    Avg_RAI Avg_Annual_Prep
1  1980  -9.422500        93.95833
2  1981  -2.730833       100.65000
3  1982 -24.964167        78.41667
4  1983  10.110833       113.49167
5  1984 -27.097500        76.28333
6  1985 -25.039167        78.34167
7  1986 -25.605833        77.77500
8  1987 -21.414167        81.96667
9  1988   9.177500       112.55833
10 1989 -10.230833        93.15000
11 1990  18.452500       121.83333
12 1991 -19.264167        84.11667
13 1992 -12.605833        90.77500
14 1993  -1.655833       101.72500
15 1994   2.269167       105.65000
16 1995 -14.755833        88.62500
17 1996 -12.247500        91.13333
18 1997  -7.047500        96.33333
19 1998   6.177500       109.55833
20 1999  -6.164167        97.21667
21 2000 -15.247500        88.13333
22 2001 250.035833       353.41667
23 2002 -22.747500        80.63333
24 2003 -16.939167        86.44167
25 2004 -15.705833        87.67500
26 2005  12.210833       115.59167
27 2006  16.569167       119.95000
28 2007   1.394167       104.77500
29 2008 -15.647500        87.73333
```

```
30 2009 -19.864167        83.51667
```

Statistical z-score

```
> #obtaining Statistical Z-score
> # Load the required libraries
> library(dplyr)
>
>
> # Step 1: Calculate mean and standard deviation of monthly
rainfall for each month
> monthly_stats <- rain_data %>%
+   group_by(month) %>%
+   summarise(mean_precp = mean(precp, na.rm = TRUE),
+             sd_precp = sd(precp, na.rm = TRUE))
>
> # Step 2: Merge the monthly statistics with the original dataset
> rain_data <- left_join(rain_data, monthly_stats, by = "month")
>
> sd_precp=sd(precp)
>
>
> # Step 3: Calculate z-score for each monthly rainfall value
> rain_data <- rain_data %>%
+   mutate(z_score = (precp - mean_precp) /   sd_precp)
>
> # Step 4: Aggregate z-scores to obtain yearly values
> yearly_z_scores <- rain_data %>%
+   group_by(year) %>%
+   summarise(yearly_z_score = sum(z_score, na.rm = TRUE))
>
> # Print or use yearly_z_scores for further analysis
> print(yearly_z_scores)
# A tibble: 30 × 2
    year yearly_z_score
   <dbl>          <dbl>
 1  1980         -0.771
 2  1981         -0.224
 3  1982         -2.04
 4  1983          0.828
 5  1984         -2.22
 6  1985         -2.05
 7  1986         -2.10
 8  1987         -1.75
 9  1988          0.751
10  1989         -0.837
# i 20 more rows
# i Use `print(n = ...)` to see more rows
>
> # Convert yearly_z_scores to a dataframe
> yearly_z_scores_df <- as.data.frame(yearly_z_scores)
>
> # Print the dataframe
> print(yearly_z_scores_df)
```

```
   year yearly_z_score
1  1980     -0.7711755
2  1981     -0.2235024
3  1982     -2.0431682
4  1983      0.8275114
5  1984     -2.2177688
6  1985     -2.0493065
7  1986     -2.0956848
8  1987     -1.7526219
9  1988      0.7511237
10 1989     -0.8373327
11 1990      1.5102271
12 1991     -1.5766572
13 1992     -1.0317123
14 1993     -0.1355201
15 1994      0.1857178
16 1995     -1.2076770
17 1996     -1.0023849
18 1997     -0.5767959
19 1998      0.5055916
20 1999     -0.5045003
21 2000     -1.2479170
22 2001     20.4639425
23 2002     -1.8617473
24 2003     -1.3863698
25 2004     -1.2854288
26 2005      0.9993839
27 2006      1.3560875
28 2007      0.1141042
29 2008     -1.2806546
30 2009     -1.6257636
```

Modified China Z-index (MZCI)

```
> # Calculate the median and median absolute deviation (MAD) of
monthly rainfall for each month over the entire period
> monthly_median_mad <- rain_data %>%
+   group_by(month) %>%
+   summarise(median_precp = median(precp, na.rm = TRUE),
+             mad_precp = mad(precp, na.rm = TRUE))
>
> # Join the monthly median and MAD with the original data
> rain_data_with_stats <- left_join(rain_data, monthly_median_mad,
by = "month")
>
> # Calculate the deviation of each month's rainfall from the long-
term monthly median
> rain_data_with_deviation <- rain_data_with_stats %>%
+   mutate(deviation = precp - median_precp)
>
> # Calculate the Z-score for each month using MAD instead of
standard deviation
> rain_data_with_zscore <- rain_data_with_deviation %>%
+   mutate(z_score = deviation / mad_precp)
>
```

```
> # Calculate the average Z-score for each year to obtain the MCZI
> yearly_mczi <- rain_data_with_zscore %>%
+   group_by(year) %>%
+   summarise(MCZI = mean(z_score, na.rm = TRUE))
>
> # Print the dataframe with yearly MCZI values
> print(yearly_mczi)
# A tibble: 30 × 2
    year   MCZI
   <dbl>  <dbl>
 1  1980 1.44
 2  1981 1.29
 3  1982 0.566
 4  1983 0.429
 5  1984 0.176
 6  1985 0.0352
 7  1986 1.52
 8  1987 1.67
 9  1988 0.439
10  1989 1.15
# i 20 more rows
# i Use `print(n = ...)` to see more rows
>
> #detrended yield
>
> # dataframe with the years detrended yeild and all indexes
> index_df
   Year detrended_yield          SPI Avg_Annual_prep
2  1982      267.256995 -0.301137933        78.41667
3  1983       58.480197 -0.508813580       113.49167
4  1984      373.703399  0.029768173        76.28333
5  1985       64.526601 -0.910197841        78.34167
6  1986     -171.050197 -0.855492342        77.77500
7  1987     -302.726995 -0.968753874        81.96667
8  1988     -229.303793 -0.318219871       112.55833
9  1989      156.919409  0.226440012        93.15000
10 1990       -2.757389  0.033462668       121.83333
11 1991      -14.734187  0.337364376        84.11667
12 1992     -331.310985 -0.741772789        90.77500
13 1993     -182.987783 -0.289610990       101.72500
14 1994       51.735419  0.195844929       105.65000
15 1995       83.458621 -0.100036908        88.62500
16 1996     -153.618177 -0.535248318        91.13333
17 1997       92.105025 -0.326059356        96.33333
18 1998     -427.471773 -0.005252559       109.55833
19 1999      186.451429  0.230609647        97.21667
20 2000      290.474631 -0.162583174        88.13333
21 2001     -221.202167  2.317486343       353.41667
22 2002     -115.778966  3.260391960        80.63333
23 2003     -232.355764 -0.703281052        86.44167
24 2004     -461.932562 -0.569687839        87.67500
25 2005     -247.509360 -0.127334953       115.59167
26 2006       12.913842  0.536836945       119.95000
27 2007      320.437044  0.463219692       104.77500
```

```
28 2008      744.460246 -0.177415259        87.73333
29 2009      158.183448 -0.656185274        83.51667
      Avg_RAI Statistical_Zscore        MZCI
2  -24.964167        -1.64469497  0.56602205
3   10.110833        -0.92971865  0.42909418
4  -27.097500        -3.45087589  0.17606544
5  -25.039167        -3.78300986  0.03522371
6  -25.605833        -1.71411519  1.52272828
7  -21.414167        -0.01418274  1.66857196
8    9.177500        -1.73603504  0.43935989
9  -10.230833        -1.36788843  1.15283856
10  18.452500         1.89214244  1.62551322
11 -19.264167        -4.81343598 -0.08597717
12 -12.605833        -3.52687799 -0.26092554
13  -1.655833         0.25043302  2.02781506
14   2.269167        -1.40915729  0.56274450
15 -14.755833         1.70945400  1.30126608
16 -12.247500        -3.42250580 -0.13915558
17  -7.047500         4.70788246  4.60652922
18   6.177500         0.95457939  0.81642206
19  -6.164167        -2.37159331  0.34976993
20 -15.247500        -3.90727001  0.23562542
21 250.035833        40.00837724 17.19158313
22 -22.747500        -4.19894724 -0.10518692
23 -16.939167        -4.38643049 -0.11009033
24 -15.705833        -3.25240743  0.17262755
25  12.210833         1.84425044  0.85418316
26  16.569167         2.08860474  1.48396345
27   1.394167        -2.85092657 -0.10977943
28 -15.647500        -2.75029912  0.16161584
29 -19.864167        -1.43117452  0.58758222
```

Detrending crop yield

R code:

```
> # Predict crop yield values using the linear regression model
> lm_model <- lm(`Crop yeild:wheat` ~ Year, data = maha_data)  #
Corrected variable name
> predicted_values <- predict(lm_model)
> detrended_yield <- maha_data$`Crop yeild:wheat` - predicted_values
>
> # Plot original vs detrended crop yield
> plot(maha_data$Year, maha_data$`Crop yeild:wheat`, col = "blue",
main = "Original vs Detrended Crop Yield")
> lines(maha_data$Year, detrended_yield, col = "red")
```

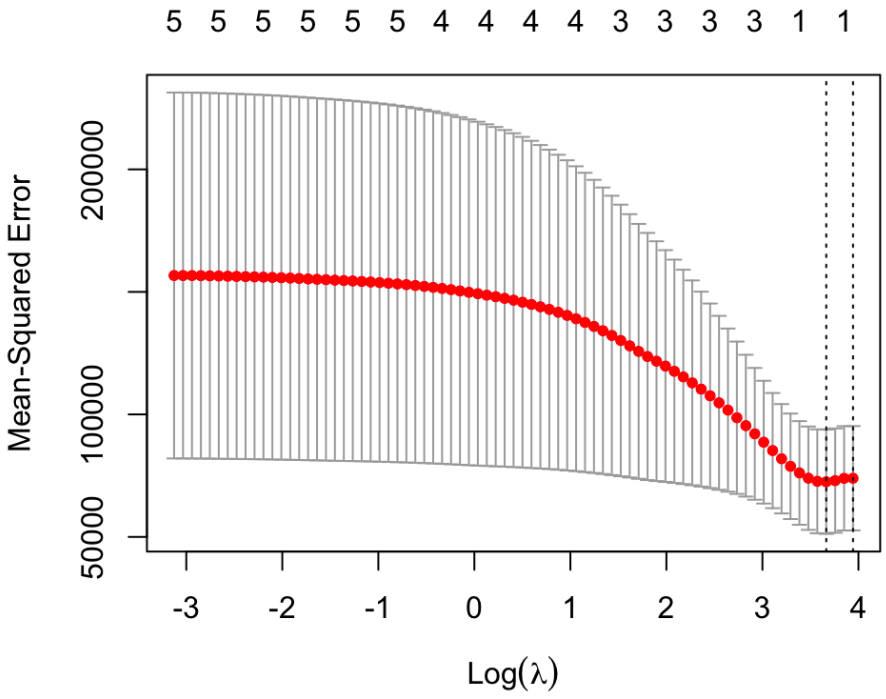
Fitting Lasso regression model

R code:

```
> library(glmnet)
>    # Prepare data
```

```
>    response_var <- index_df$detrended_yield
>  explanatory_vars <- index_df[, c("Avg_Annual_prep", "SPI",
"Avg_RAI", "Statistical_Zscore", "MZCI")]
>
>    # Fit Lasso regression model
>    lasso_model <- glmnet(as.matrix(explanatory_vars),
response_var, alpha = 1)
>
>    summary(lasso_model)
           Length Class     Mode
a0          84    -none-    numeric
beta       420    dgCMatrix S4
df          84    -none-    numeric
dim          2    -none-    numeric
lambda      84    -none-    numeric
dev.ratio   84    -none-    numeric
nulldev      1    -none-    numeric
npasses      1    -none-    numeric
jerr         1    -none-    numeric
offset       1    -none-    logical
call         4    -none-    call
nobs         1    -none-    numeric
>
>    # Extract coefficients
>    coef_matrix <- coef(lasso_model)
>    #Based on our output,the coefficients for "SPI" along with the
other explanatory variables consistently appears with non-zero
coefficients across different values of lambda

 >
> plot(cv.lasso)
```



```
>  # Compute the absolute sum of coefficients for each variable
>  abs_sum_coef <- rowSums(abs(coef_matrix[-1,]))
>
>  # Identify the variable with the highest absolute sum of
coefficients
>  most_important_variable <-
names(abs_sum_coef)[which.max(abs_sum_coef)]
>
>       # Print the most important variable
>       print(most_important_variable)
[1] "SPI"
```

Claim Size Modelling Using Multiple Regression

R-code:

```
> mod4=lm(dataofmodel$`Reported Claims(Rs in Cr.)`~dataofmodel$`Farm
er Applications Enrolled (In Lakhs)`+dataofmodel$`Gross Premium Coll
ected(Rs in Cr.)`+dataofmodel$Rainfall+dataofmodel$`Sum Insured(in C
rores)`+dataofmodel$`Area Insured(Ha)`)
> summary(mod4)

Call:
lm(formula = dataofmodel$`Reported Claims(Rs in Cr.)` ~ dataofmodel$
`Farmer Applications Enrolled (In Lakhs)` +
    dataofmodel$`Gross Premium Collected(Rs in Cr.)` + dataofmodel$R
ainfall +
    dataofmodel$`Sum Insured(in Crores)` + dataofmodel$`Area Insured
(Ha)`)

Residuals:
    Min      1Q  Median      3Q     Max
-3444.7  -211.5   -41.9   130.1  2802.5

Coefficients:
                                                          Estimate Std
. Error t value Pr(>|t|)
(Intercept)                                             163.180585 175
.876907   0.928    0.355
dataofmodel$`Farmer Applications Enrolled (In Lakhs)`     3.139060   3
.705213   0.847    0.399
dataofmodel$`Gross Premium Collected(Rs in Cr.)`          0.554380   0
.083962   6.603 1.04e-09 ***
dataofmodel$Rainfall                                     -0.072670   0
.081742  -0.889    0.376
dataofmodel$`Sum Insured(in Crores)`                     -0.004316   0
.014386  -0.300    0.765
dataofmodel$`Area Insured(Ha)`                            0.088612   0
.031697   2.796    0.006 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 820 on 125 degrees of freedom
Multiple R-squared:  0.7181,       Adjusted R-squared:  0.7068
F-statistic: 63.67 on 5 and 125 DF,  p-value: < 2.2e-16
> par(mfrow=c(2,2))
> plot(mod4)
```

Claim Frequency Modelling using Time Series Analysis

```
> library(readxl)
> data <- read_excel("C:/Users/Snehal Jha/Downloads/project123.xlsx", sheet
= "Sheet1") #importing data
> ts1 = ts(data$claims, start = 2000, frequency = 1)
> plot(ts1)
> dts1 = diff(diff(log(ts1)))
> plot(dts1)
> library(tseries)
> adf.test(dts1, "stationary")

        Augmented Dickey-Fuller Test

data:  dts1
Dickey-Fuller = -3.7316, Lag order = 2, p-value = 0.0406
alternative hypothesis: stationary

> acf(dts1)
> pacf(dts1)
> library(forecast)
> ARIMA = auto.arima(log(ts1), d = 2, ic = "aic", trace = T)

 ARIMA(2,2,2)                    : 23.15081
 ARIMA(0,2,0)                    : 39.1209
 ARIMA(1,2,0)                    : 31.7687
 ARIMA(0,2,1)                    : 26.90721
 ARIMA(1,2,2)                    : 26.0268
 ARIMA(2,2,1)                    : 21.21778
 ARIMA(1,2,1)                    : 25.14917
 ARIMA(2,2,0)                    : 19.22675
 ARIMA(3,2,0)                    : 21.21704
 ARIMA(3,2,1)                    : 22.98866

 Best model: ARIMA(2,2,0)

> forecasted = forecast(ARIMA, h = 5)
> forecasted
     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2024       16.28580 15.86970 16.70190 15.64943 16.92217
2025       16.39108 15.85343 16.92873 15.56881 17.21335
2026       16.43081 15.67901 17.18262 15.28104 17.58059
2027       16.47833 15.39413 17.56252 14.82019 18.13646
2028       16.56713 15.24730 17.88695 14.54862 18.58563
> plot(forecasted)
> Box.test(ARIMA$residuals, type = "Ljung-Box")

        Box-Ljung test

data:  ARIMA$residuals
X-squared = 0.74632, df = 1, p-value = 0.3876

> fr = as.numeric(forecasted$mean)
> forecasted_values = exp(fr)
> forecasted_values
[1] 11825892 13138764 13671374 14336617 15667920
```

References

i. Comparison of four precipitation-based drought indices in Marathwada region of Maharashtra, India. Authors: Mohit Mayoor, Anjani Kumari, Somnath Mahapatra, Prabeer Kumar Parhi, and Harendra Prasad Singh.

ii. Mukherjee, Subhankar & Pal, Parthapratim. (2017). Impediments to the Spread of Crop Insurance in India. Economic and political weekly. 52.

iii. Agriculture Statistics at a Glance (2021, 2022)

iv. Gurdev Singh 'Performance of NAIS' (June 2010)

v. Rainfall Statistics of India (2018 – 2021)

vi. Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

vii. Performance Evaluation of Pradhan Mantri Fasal Bima Yojana (PMFBY), IIM Ahmedabad

viii. https://loksabhadocs.nic.in/Refinput/New_Reference_Notes/English/CROP_INSURANCE_IN_INDIA_2015.pdf

ix. https://www.spotfire.com/glossary/what-is-power-analysis

x. https://www.imdpune.gov.in/library/public/e-book110.pdf

xi. https://www.ceicdata.com/en/india/production-of-foodgrains-in-major-states-wheat/agricultural-production-wheat-maharashtra

xii. https://www.tandfonline.com/doi/full/10.1080/23322039.2022.2104780

xiii. https://tradingeconomics.com/calendar/interest-rate

xiv. https://pib.gov.in/PressReleasePage.aspx?PRID=1843882

xv. https://www.tropmet.res.in/monsoon_workshop/23_pdf/Guhathakurta_pulak.pdf

xvi. https://pmfby.gov.in/adminStatistics/dashboard