

Reserves Forecasting Chain Ladder Analysis

– *Data Understanding & Preparation*

SEPTEMBER 7

Universidad Nacional de Colombia

Authored by: Cristian Ávila



Reserves Forecasting Chain Ladder Analysis

Data Understanding & Preparation

This document contains the results obtained after performing data understanding & preparation analysis to the file medmal_pos.csv, which contains the information of claims for Medical Malpractice LoB.

The analysis was performed in google collab using python programming Language in the following link

<https://colab.research.google.com/drive/1AePIIA1igsPBqIn2hzECRt2AnAYXdW->

In the next sections, the results and analysis will be shown according to guidance and tips provided during the class.

Data Understanding

For understanding the data we perform the following steps:

1. Displaying the first few rows of the dataset, as a quick way to get an overview of what the data looks like:

```
[2] # Display the first few rows of the dataset
print(df.head())
```

| | GRCODE | GRNAME | AccidentYear | DevelopmentYear | DevelopmentLag | \ |
|---|--------|--------------------|--------------|-----------------|----------------|---|
| 0 | 669 | Scpie Indemnity Co | 1988 | 1988 | 1 | |
| 1 | 669 | Scpie Indemnity Co | 1988 | 1989 | 2 | |
| 2 | 669 | Scpie Indemnity Co | 1988 | 1990 | 3 | |
| 3 | 669 | Scpie Indemnity Co | 1988 | 1991 | 4 | |
| 4 | 669 | Scpie Indemnity Co | 1988 | 1992 | 5 | |

| | IncurLoss_F2 | CumPaidLoss_F2 | BulkLoss_F2 | EarnedPremDIR_F2 | \ |
|---|--------------|----------------|-------------|------------------|---|
| 0 | 121905 | 2716 | 97966 | 129104 | |
| 1 | 112211 | 24576 | 64117 | 129104 | |
| 2 | 103226 | 43990 | 39008 | 129104 | |
| 3 | 99599 | 59722 | 20736 | 129104 | |
| 4 | 96006 | 71019 | 13599 | 129104 | |

| | EarnedPremCeded_F2 | EarnedPremNet_F2 | Single | PostedReserve97_F2 |
|---|--------------------|------------------|--------|--------------------|
| 0 | -6214 | 135318 | 0 | 344558 |
| 1 | -6214 | 135318 | 0 | 344558 |
| 2 | -6214 | 135318 | 0 | 344558 |
| 3 | -6214 | 135318 | 0 | 344558 |
| 4 | -6214 | 135318 | 0 | 344558 |

Can be identified that the file has this structure:

- GRCODE NAIC company code (including insurer groups and single insurers)
- GRNAME NAIC company name (including insurer groups and single insurers)
- AccidentYear Accident year(1988 to 1997)
- DevelopmentYear Development year (1988 to 1997)
- DevelopmentLag Development year (AY-1987 + DY-1987 - 1)
- IncurLoss_ Incurred losses and allocated expenses reported at year end
- CumPaidLoss_ Cumulative paid losses and allocated expenses at year end

- BulkLoss_ Bulk and IBNR reserves on net losses and defense and cost containment expenses reported at year end
PostedReserve97_ Posted reserves in year 1997 taken from the Underwriting and Investment Exhibit – Part 2A, including net losses unpaid and unpaid loss adjustment expenses
- EarnedPremDIR_ Premiums earned at incurral year - direct and assumed
- EarnedPremCeded_ Premiums earned at incurral year – ceded
- EarnedPremNet_ Premiums earned at incurral year - net
Single 1 indicates a single entity, 0 indicates a group insurer

2. Check for missing Values

```
# Check for missing values
print(df.isnull().sum())
```

```
GRCODE          0
GRNAME          0
AccidentYear    0
DevelopmentYear  0
DevelopmentLag  0
IncurLoss_F2    0
CumPaidLoss_F2  0
BulkLoss_F2     0
EarnedPremDIR_F2 0
EarnedPremCeded_F2 0
EarnedPremNet_F2 0
Single          0
PostedReserve97_F2 0
dtype: int64
```

`data.isnull()` returns a DataFrame of the same shape as data but with Boolean values indicating whether each element is NaN (null) or not. `.sum()` is used to count the

number of True values (which are equivalent to missing values) in each column., Our data set does not contains NaN elements.

3. Summary statistics:

| | GRCODE | AccidentYear | DevelopmentYear | DevelopmentLag | \ |
|-------|--------------|--------------|-----------------|----------------|---|
| count | 3400.000000 | 3400.000000 | 3400.000000 | 3400.000000 | |
| mean | 22809.764706 | 1992.500000 | 1997.000000 | 5.500000 | |
| std | 14708.377001 | 2.872704 | 4.062617 | 2.872704 | |
| min | 669.000000 | 1988.000000 | 1988.000000 | 1.000000 | |
| 25% | 10341.000000 | 1990.000000 | 1994.000000 | 3.000000 | |
| 50% | 19764.000000 | 1992.500000 | 1997.000000 | 5.500000 | |
| 75% | 36234.000000 | 1995.000000 | 2000.000000 | 8.000000 | |
| max | 44504.000000 | 1997.000000 | 2006.000000 | 10.000000 | |

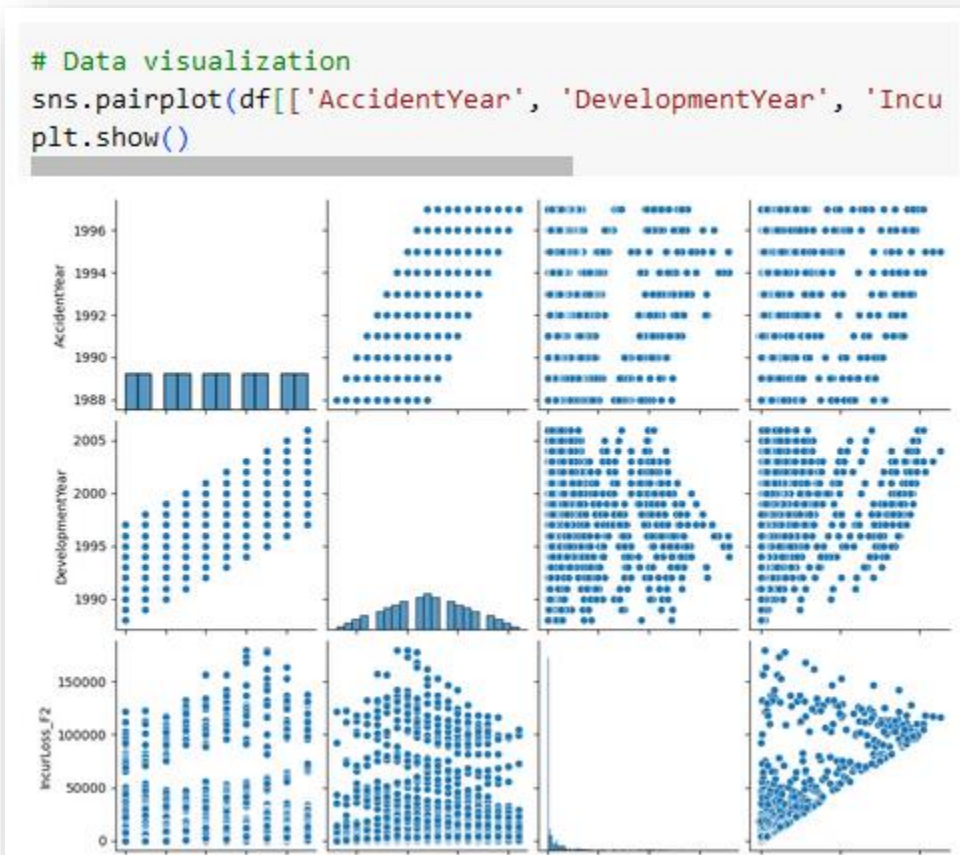
| | IncurLoss_F2 | CumPaidLoss_F2 | BulkLoss_F2 | EarnedPremDIR_F2 | \ |
|-------|---------------|----------------|---------------|------------------|---|
| count | 3.400000e+03 | 3.400000e+03 | 3400.000000 | 3400.000000 | |
| mean | 5.851528e-17 | -5.433562e-17 | 1095.803235 | 14111.605882 | |
| std | 1.000147e+00 | 1.000147e+00 | 7612.672277 | 26399.284476 | |
| min | -4.338370e-01 | -4.612378e-01 | -32101.000000 | -781.000000 | |
| 25% | -4.332026e-01 | -3.917256e-01 | 0.000000 | 0.000000 | |
| 50% | -4.091345e-01 | -3.808023e-01 | 0.000000 | 1500.000000 | |
| 75% | -9.548327e-02 | -1.355527e-01 | 107.250000 | 18094.500000 | |
| max | 6.262040e+00 | 6.220053e+00 | 104402.000000 | 131948.000000 | |

| | EarnedPremCeded_F2 | EarnedPremNet_F2 | Single | PostedReserve97_F2 | |
|-------|--------------------|------------------|-------------|--------------------|--|
| count | 3400.000000 | 3400.000000 | 3400.000000 | 3400.000000 | |
| mean | 1803.497059 | 12308.108824 | 0.852941 | 57065.529412 | |
| std | 3893.424584 | 24824.225795 | 0.354217 | 134355.533990 | |
| min | -6214.000000 | -728.000000 | 0.000000 | 0.000000 | |
| 25% | 0.000000 | 0.000000 | 1.000000 | 629.000000 | |
| 50% | 106.500000 | 1302.000000 | 1.000000 | 5875.000000 | |
| 75% | 1473.500000 | 13490.000000 | 1.000000 | 46762.000000 | |
| max | 25553.000000 | 135318.000000 | 1.000000 | 702246.000000 | |

data.describe() generates summary statistics for numerical columns in the dataset. This includes count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum.

From the dataset we can state, that contains information that makes sense in the regard of claims historical information were incurred losses have been registered.

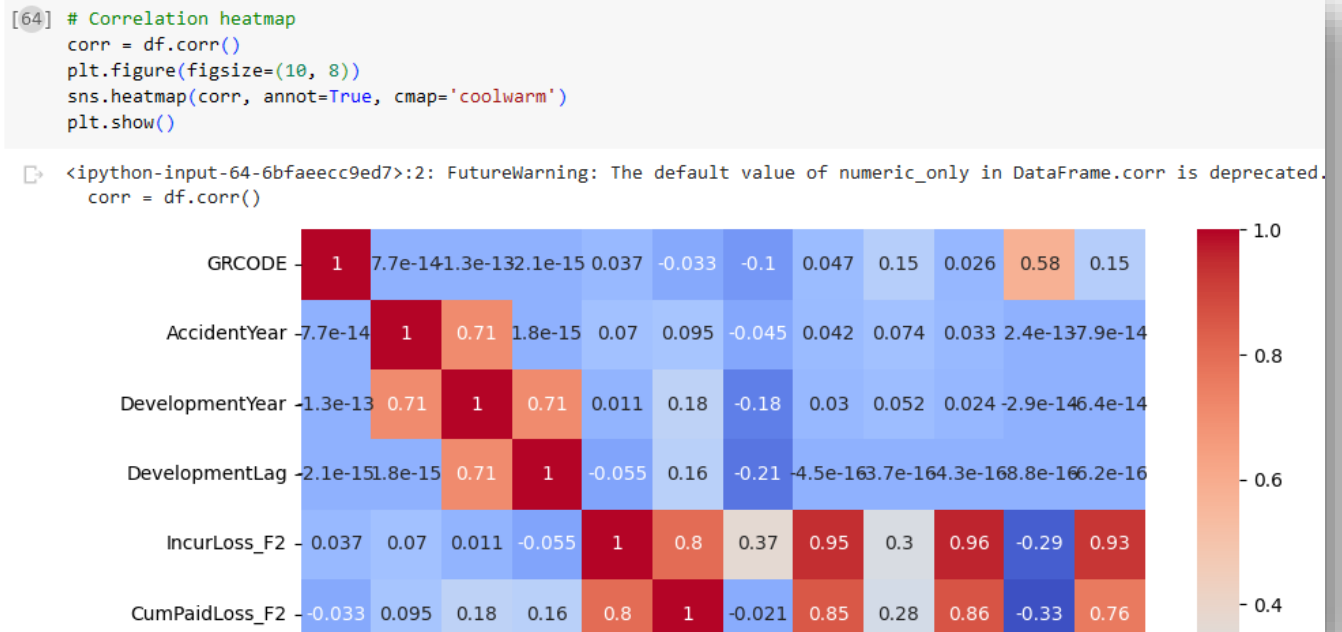
4. Data Visualization



`sns.pairplot()` generates a grid of scatter plots for the specified columns. In this case, it creates scatter plots for AccidentYear, DevelopmentYear, IncurLoss_F2, and CumPaidLoss_F2.

`plt.show()` displays the generated plot.

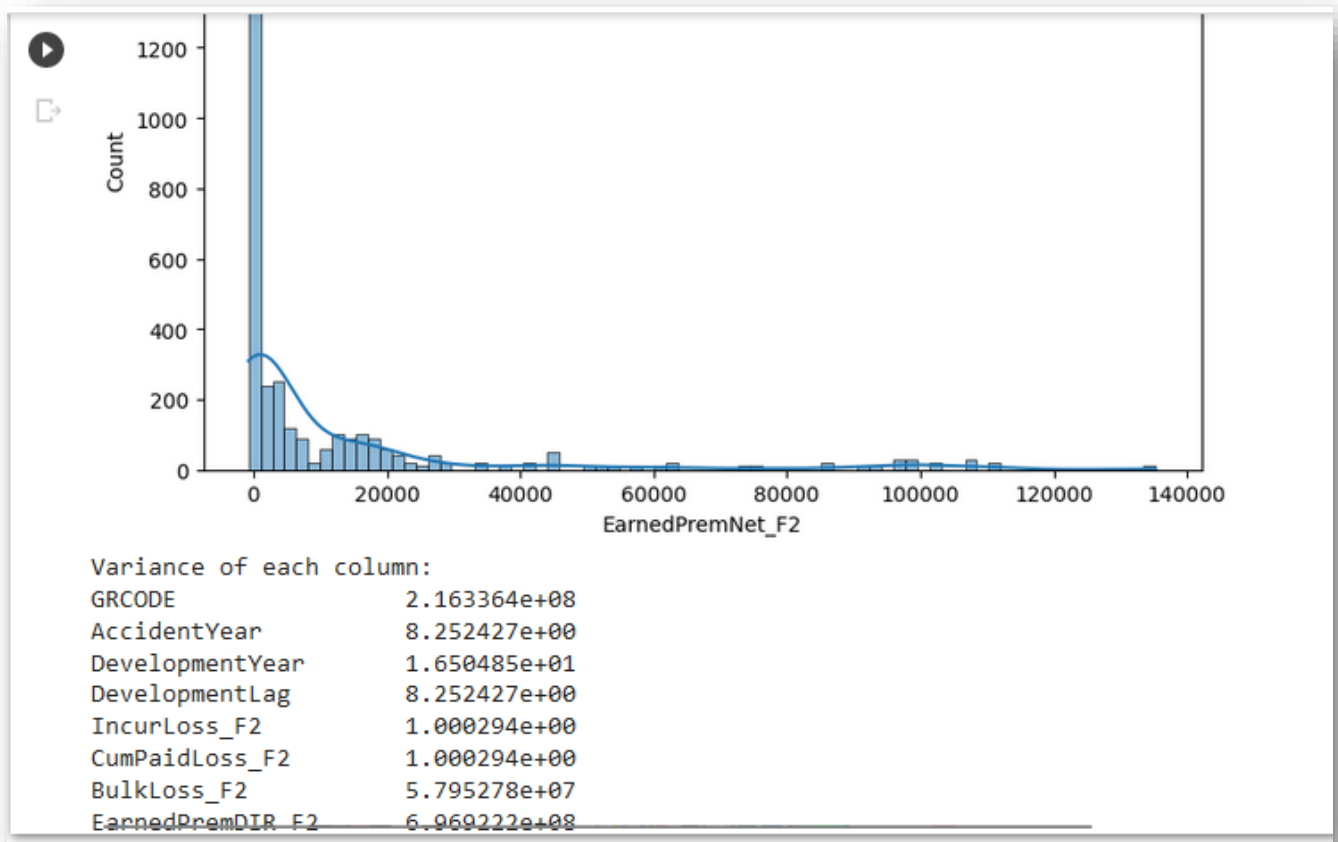
5. Correlation Heatmap



`data.corr()` computes the pairwise correlation of columns in the dataset.
`sns.heatmap()` creates a heatmap to visualize the correlations.
`annot=True` adds the correlation values to the heatmap.
`cmap='coolwarm'` sets the color scheme.

6. Additional Data understanding steps:

In this section we try to match each column of the data set to a distribution, then we calculate the variance on each column.



Data Preparation

This section of the code is focused on data preprocessing, specifically on the rationale for including or excluding certain rows of data. Let's break it down: