SOCIETY OF ACTUARIES
TECHNOLOGY SECTION

ACTUARIAL TECHNOLOGY TODAY
MAY 2020

# The Use of Advanced Predictive Analytics for Rate Making in Insurance

By Kemi Akinyemi and Ben Leiser

The Society of Actuaries Actuarial Innovation & Technology Program Steering Committee engaged Risk & Regulatory Consulting LLC (RRC) to conduct research on the intersection of U.S. insurance ratemaking and analytics techniques in advanced modern rating systems and associated regulatory considerations in the U.S. RRC performed research on the current regulatory environment and emerging issues concerning the use of advanced analytics techniques for rating and summarized that information within a report located at *https://www.soa.org/globalassets/assets/files/resources/research-report/2019/insurance-regulatory-issues-us.pdf*. The contents of this paper are based on the research within the report.

Companies use basic and complex modeling tools and techniques to set rates for their products within the constraints of applicable regulations, statutes and laws. The complex modeling tools rely on advanced predictive modeling techniques. The property and casualty industry appears to be further along in its use of advanced predictive analytics than the life and health industries.

Actuaries currently rely on Actuarial Standards of Practice (ASOP) and laws and regulations such as state and federal statutes as their guidelines for developing advanced models, but they are looking to regulators to provide additional comprehensive guidance. Companies are looking to understand what documentation regulators need when rates are developed using innovative methods and advanced predictive techniques.

State regulators are tasked with ensuring that the rates for the insurance products are adequate, not excessive, and not unfairly discriminatory, among other responsibilities. Regulators understand that advanced modeling techniques could vary across companies and are looking to companies to appropriately demonstrate that their use of advanced predictive analytics in determining rates are appropriate for the products offered and are not unfairly discriminatory to policyholders. Some companies that are exploring advanced predictive analytics are using them in ratemaking, while others are limiting their reliance on advanced predictive analytics until comprehensive regulatory guidance becomes available.

For any model used for rate making, the company must understand how the data is being used, how the model relies on the data for rate making, and be able to communicate this clearly to all key stakeholders, including policyholders, management and the public.

## MODELING TECHNIQUES

The following are some modeling techniques that are used for rate making. Companies need to assess the risk and reward tradeoff between the tools that they consider or use.

1. Basic modeling techniques such as trending and linear regression, are still in use in some companies, especially within the health industry. In the property and casualty industry, the following are two basic univariate approaches for determining an overall rate level:

    a. **Pure-premium method:** The pure-premium method determines an indicated average rate and involves projecting the average loss and loss adjustment expenses per exposure and the average fixed expenses per exposure to the period that the rates will be in effect. The sum of those two is then adjusted for variable expenses

and the target profit percentage by dividing by one minus the sum of the variable expense provision and target profit percentage to get the indicated average rate. In other words, the average rate is the average premium per exposure. To derive the current premiums, one would multiply the average rate by the current exposures. This method is used with new lines of business where there aren't any current rates to adjust.

b. **Loss-ratio method:** The loss-ratio method compares the estimated percentage of each premium dollar needed to cover future losses, loss adjustment expenses, and other fixed expenses to the amount of each premium dollar that is available to pay for such costs. The sum of the projected loss and the loss adjustment expense ratio, and the fixed expense ratio is divided by one minus the sum of the variable expense provision and the target profit percentage to get the indicated change factor. The change factor represents the indicated adjustment to the current rates. For example, if the change factor is 1.10, this indicates that the current rates are 10 percent too low and need to be increased by a factor of 1.10. The major difference between the pure-premium and loss-ratio approaches is that the loss-ratio approach uses premium as opposed to exposure.

Basic models are still in use primarily because of their ease of design, use and explanation. Univariate methods are limited because they do not account for the effect of other rating variables. There are multivariate classification rate making techniques that consider multiple rating variables simultaneously and automatically adjust for exposure correlations between rating variables and allow for the interaction and interdependency between two or more rating variables.

2. Complex models are increasingly being used and are multivariate in nature. Multivariate methods also attempt to remove unsystematic effects in the data (noise) and capture only the systematic effects in the data (signal) as much as possible. Multivariate techniques also produce model diagnostics and information about the appropriateness of the model fitted. The property and casualty industry appears to be further along in their use of advanced predictive analytics than the life and health industries. The following are some of the more complex models being used:

   a. **Generalized linear models (GLM):** The GLM models are one of the most common tools used because it is easier to design, has a wide range of applicability, and is relatively easier to explain to regulators and other stakeholders. The GLM is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. GLMs are based on a single regression equation whose predictions are easier to explain through regression coefficients. GLMs and other linear models focus on expressing a relationship between an observed response variable and several predictor or explanatory variables. For example, actuaries can use GLMs to determine base rates and proposed rate changes for pricing analyses. As an example, GLMs can be helpful for auto rate making because there are many factors that can be predictive of an adequate rate such as the number of accidents, credit scores, geographic location, vehicle type, age, gender and years of driving experience. GLMs can account for multiple variables at a time and more accurately than more simple models. GLMs can also be used in a reserving capacity to estimate future loss reserves for non-traditional exposures such as loyalty programs for airlines and hotels.

   b. **Machine learning:** Machine learning is an application of artificial intelligence that provides systems with the ability to automatically learn and improve from experience without being explicitly programmed. It focuses on the development of computer programs that can access data and use it to learn for themselves. The results obtained from machine learning are likely to be more accurate than the results from GLM, but the machine learning model is not as easy to explain as the GLM model.

      Machine learning techniques, particularly artificial neural networks (ANNs), are increasingly popular in several disciplines, but not necessarily insurance due to a lack of interpretability. The explosion in the variety and volume of available data, coupled with cheap data storage and fast computing power, have made ANNs a key tool of data scientists. ANNs allow modeling of nonlinear processes and can be a useful tool for solving problems such as retention modeling, fraud detection, claims triage, property/casualty reserving using individual claim detail and traditional pricing models. ANNs attempt to digitally replicate the biological mechanism of the human brain and its neurons. Although ANNs have been on the upswing in a variety of fields, the insurance sector has not utilized these "brain-like" techniques on a large scale. The insurance field is still heavily skewed in favor of the more familiar and traditional data-mining techniques, such as GLMs and rule-based algorithms.

Other tools used include classification and regression trees, random forests, and neural networks, but these tend to be less transparent. Machine learning is increasingly being considered in the insurance industry. GLMs and linear models are very transparent, making it easier to explain to regulators and stakeholders than machine learning. Although machine learning and other advanced modeling techniques could be more accurate than GLMs and linear models, it

could be difficult to determine the extent to which each variable is contributing to the determined rate. As insurers consider the use of predictive analytics in the determination of rates, regulators want to ensure that the determined rates are appropriate and not unfairly discriminatory. Because the regulatory space is evolving, there is limited guidance on how the use of predictive analytics in the determination of rates could inadvertently lead to unfair discrimination and how companies can check their models and demonstrate that their models do not produce unfairly discriminatory results. Situations could arise where machine learning models unfairly discriminate against classes of individuals or promote institutional bias. Companies will need to understand what variables could be unfairly discriminatory and should be able to demonstrate that their models are compliant and not unfairly discriminatory.

## MODEL VALIDATION

Effective insurance rate making heavily relies on models working effectively to capture risk and appropriately account for the key elements of rate determination. Model validation is the process of performing an independent challenge and thorough assessment of the reasonableness and adequacy of a model based on peer review and testing across multiple dimensions, including design, data, assumptions, results, and governance. Proper model validation and governance are necessary to mitigate financial model risk due to the potential negative impact of models failing to appropriately price products as designed. Through model review and validation, companies can confirm that their models are working as designed, identify limitations of their models, and manage the associated risks in the models. An effective model validation process should be periodic and include a review of the following: data, application code, assumptions and methodologies, plan code mapping, product features, controls, and model performance and outcome analysis. Some companies review their model through a "Train/Test/Validate" approach, where three different datasets are used to train the model, test the model, and then validate the model. Through out-of-time validation, the model can be validated based on a different time period to evaluate its robustness.

## DATA SOURCES AND TYPES OF VARIABLES ALLOWED

The types of variables allowed to be used in rating could vary depending on the line of business. The rating variables are used to segment the insured population into different groups of similar risks for rating purposes. The criteria for selecting variables may be summarized into the following criteria categories: actuarial or statistical, operational, social, and legal.

**Actuarial or Statistical Criteria:** Companies typically consider the following actuarial or statistical criteria to help ensure the accuracy and reliability of the potential rating variable:

- Statistical significance,
- homogeneity, and
- credibility.

**Operational Criteria:** Companies also consider practical and operational constraints after identifying the statistical criteria of their variables. Practical rating variables should have the following qualities:

- Objective,
- inexpensive to administer, and
- verifiable.

**Social Criteria:** Companies may want to consider public perception as they identify the rating variables to be used. The following items affect social acceptability of using a particular risk characteristic as a rating variable:

- Affordability,
- causality,
- controllability, and
- privacy concerns.

**Legal Criteria:** The risk classification may be affected by state and federal statutes and regulations. Generally, constitutions govern statutes and statutes govern regulations. The rate classification system must comply with the applicable laws and regulations of each jurisdiction in which a company is writing business. Actuaries need to be familiar with the laws and regulations of each jurisdiction in which their company writes insurance and assure that the classification rating complies with that jurisdiction's laws and regulations. This usually requires working with other professionals, such as lawyers or regulatory compliance experts, in determining what is acceptable and what is not.

## BENEFITS AND DRAWBACKS

Advanced predictive analytics techniques enable insurers to better understand their data. Proper implementation of predictive analytics techniques can improve an insurer's consistency and efficiency in product pricing and product development. The use of predictive analytics in rate making has several benefits:

- Improve pricing by increasing the number of rate segments and price points,
- efficient underwriting and pricing,
- improve competitive advantage, and
- provide insurers with a better understanding of the risks and key drivers.

Some drawbacks of the use of advanced modeling techniques in insurance could include the following:

- Companies could have unrealistic expectations that machine learning and artificial intelligence can deliver solutions to every business problem.
- Internal models could be built by people who do not understand sound actuarial practices. Insufficient understanding of internal or vendor models could lead to setting unfairly discriminatory rates. Companies need to be able to explain their models such that the ability of consumers and regulators to understand the rating is not compromised.
- Some companies may rush to be first, while utilizing poor due diligence in the process. This can lead to unintended consequences of volatility, regulatory compliance issues, or bad headlines. Making too big of a mistake too early in the process affects the stability of the model and lengthens the time to adoption.
- Certain benefits of risk pooling could be diminished if there is over-segmentation of risks.
- Consumers may be unjustifiably penalized in the form of higher rates for irrelevant factors.

## CONCLUDING REMARKS

The insurance industry appears to be trailing other industries in the use of advanced predictive models to set rates, with the property and casualty industry leading the pack. Although these complex techniques have been embraced by other industries and other sections of insurance industries, insurers appear to be cautious in the use of advanced predictive analytics for setting rates. Reasons for this may be due to limited comprehensive regulation on the use of predictive analytics for rate making and how the insurer can demonstrate that their rates are not unfairly discriminatory, as well as the resource constraints of budget, time and staff. Some of the complex models being used by insurance companies include GLMs and machine learning, with the machine learning being more precise than GLM. GLM is generally understood by the industry, but the machine learning models would typically require more explanation. The increased complexity of the models implies that companies need to be able to demonstrate that their rates are not unfairly discriminatory, and regulators need to be prepared to understand and review multiple variations of these advanced modeling techniques. Companies need to validate the models used and ensure that the data used in the model meet certain required criteria. In addition, companies will also decide their risk and reward threshold from using certain data and advanced predictive models and if the potential benefits outweigh the potential costs. ■

Kemi Akinyemi, FSA, EA, MAAA, is an actuarial consultant at Risk and Regulatory Consulting. She can be contacted at *kemi.akinyemi@riskreg.com.*

Benjamin Leiser, FSA, MAAA, is an actuarial manager at Risk and Regulatory Consulting. He can be contacted at *ben.leiser@riskreg.com.*

### REFERENCE

This article is based on the Society of Actuaries report on "Insurance Regulatory Issues in the United States," and source documents are contained within that report. Society of Actuaries. 2019. "Insurance Regulatory issues in the United States" May 2019 *https://www.soa.org/globalassets/assets/files/resources/research-report/2019/insurance-regulatory-issues-us.pdf*

# Guerrilla Automation With R:

## R Automation Despite Resource Constraints

By Timothy Quast

Guerrilla warfare is a type of asymmetric warfare wherein a smaller, less powerful military uses unconventional tactics to fight a stronger, more traditional military. It often involves a higher level of mobility, with nimbleness making up for the military's lack of resources. We can draw an analogy in business. When a problem calls for a traditional technological solution, but no such solution is currently in-place, actuaries may require a low-cost, nimble approach to meeting the needs of their stakeholders.

That's where R comes in. R is a free and open source programming language with a powerful and continually expanding set of libraries and packages. It is highly versatile and compatible with a wide variety of other systems. In this article, I will walk through a hypothetical use-case for R that highlights R's advantages. Example code will be interspersed throughout the article. The full code, complete with example excel files can be accessed via Github at the following URL: *https://github.com/TimothyQuast/Guerrilla_Automation*

If you download the repository and would like to follow along, I highly recommend installing RStudio for free from *Rstudio.com*. A rudimentary understanding of R will be helpful in following along, but it is not essential.

### THE USE-CASE

Let's suppose that our stakeholder has a large number of files in Excel throughout their network drives with financial information. They would like to be able to summarize all the financial information and trace the lineage of each subcomponent. Specifically, let's say that they want to support numerous account balances using the actuarial workpapers that feed them. They want to break each balance into pieces, with each piece corresponding to one actuarial workpaper, which contributes to that balance. Doing so would be extremely valuable for audit purposes and reasonableness testing.

The actuarial workpapers are stored in Excel in a regular format in different places throughout the stakeholder's network drives, with a variety of teams contributing to the same account balances. Moreover, the workpapers represent an aggregated version of the actuarial results, so they aren't overly granular. But they are granular enough to make a manual solution infeasible. How do we get the data we need to support the account balances?

The proper, traditional method is a big fancy subledger containing the supporting balances along with metadata that traces each balance. But let's say the stakeholder doesn't have a big, fancy subledger yet. Further, let's say that the stakeholder wants the balances supported **soon**—sooner than a big, fancy subledger can be developed. We need a temporary solution to solve the problem quickly. In such a conundrum, one might try automating the task with R!

R is suitable for the task for several reasons:

- **It's free!** It can be used for such a task without requiring investment dollars or licensing.
- **It's open source with a broad user base.** Developers are continually producing marvelous new packages that can solve tough problems, making R extremely versatile.
- **It's highly compatible.** The variety of packages and the non-proprietary nature of the system make it uniquely capable of interacting with other systems, such as Excel.
- **It's suitable for rapid prototyping.** R is elegant and fairly high-level, allowing the user to do a lot with just a little code.

R has some disadvantages too:

- **It's less efficient.** R doesn't do as well as other languages in terms of processing efficiency. That's why it's important that the workpapers are already aggregated. If they were extremely granular, with a high volume of data, then R might struggle.
- **There's a learning curve.** Like all programming languages, R must be learned. It might be difficult for new personnel to learn the language, making the process less portable.
- **It has a copyleft license.** R is usable for any commercial purpose, but the license requires that any derivative works be under a compatible open source license. In other words, you cannot distribute proprietary software that uses R: it would have to be open source. This is not a problem as long as we are internal or we are charging for labor (instead of software licensing).

The drawbacks are not overwhelming, and it won't cost us anything but time to try, so let's figure out how to solve our problem with R. I've constructed an example problem (which you can find in the GitHub repository) using four Excel files (see Figure 1). **Ledger Balances.xlsx** contains the "ledger balances" we are trying to support. Three workpaper files contain the "workpapers" that support the "ledger balances." Pretend that the workpaper files are located in disparate places throughout the stakeholder's network drives!

Figure 1
Example Problem

**Excel Files/Ledger Balances.xlsx**

| Account Number | Amount |
|---|---|
| A0000 | 15368 |
| A1111 | 6973 |
| A2222 | 13909 |
| A3333 | 3349 |
| A4444 | 7725 |
| A5555 | 9504 |
| A7777 | 15486 |
| A8888 | 8472 |
| A9999 | 2992 |

**Excel Files/Workpaper 1.xlsx**

| Account Number | Amount |
|---|---|
| A0000 | 4747 |
| A1111 | 0 |
| A2222 | 5228 |
| A3333 | 0 |
| A4444 | 4741 |
| A5555 | 4445 |
| A7777 | 9560 |
| A8888 | 0 |
| A9999 | 0 |

**Excel Files/Workpaper 2.xlsx**

| Account Number | Amount |
|---|---|
| A0000 | 9836 |
| A1111 | 0 |
| A2222 | 8508 |
| A3333 | 3349 |
| A4444 | 0 |
| A5555 | 0 |
| A7777 | 5926 |
| A8888 | 8472 |
| A9999 | 0 |

**Excel Files/Workpaper 3.xlsx**

| Account Number | Amount |
|---|---|
| A0000 | 785 |
| A1111 | 6973 |
| A2222 | 173 |
| A3333 | 0 |
| A4444 | 2984 |
| A5555 | 5059 |
| A7777 | 0 |
| A8888 | 0 |
| A9999 | 2992 |

Figure 2
Input Files Chronicled

**Excel Files/Input Control.xlsx**

| Filepath | Sheet | Range |
|---|---|---|
| Excel Files/Workpaper 1.xlsx | Sheet1 | B4:C13 |
| Excel Files/Workpaper 2.xlsx | Sheet1 | B4:C9001 |
| Excel Files/Workpaper 3.xlsx | An Unusual Sheet Name | C4:D13 |
| Excel Files/Workpaper 3.xlsx | A Mistaken Sheet Name | C4:D13 |
| Excel Files/Workpaper 3 Mistaken Filename.xlsx | An Unusual Sheet Name | C4:D13 |

You can also manually verify that the account number totals in the three workpapers sum to the ledger balance totals, but then what's the point of automating it? Notice that the workpapers keep the data in a regular format, but **Workpaper 3.xlsx** has the data in different cells and it has an unusual sheet name! These differences are intentional, and we will handle them in our solution.

## THE SOLUTION

The first step to solving any problem is simple: get organized! I start by chronicling the input files in the **Excel Files/Input Control.xlsx** spreadsheet (see Figure 2).

I typically prefer to use absolute file paths, but I set this up with relative paths so that it will work on other machines. The first three rows here contain correct information. The last two contain errors in the sheet name and file name, respectively. This can happen if, for example, the process is run on a monthly basis, and the file-naming conventions for a workpaper change from one month to the next. I included them to demonstrate R's ability to deal with these complications. Two things to note:

- The range for Workpaper 2 is unnecessarily large. This is a useful tactic when the numbers of rows changes from month to month and you want to ensure that you capture all the data. We eventually remove the resulting empty rows from our data.
- There are no errors in the Range column. A runtime error in the Range column would be captured in the same way as file/sheet name errors, but a logical error (i.e., if we entered the range incorrectly) could create challenges. It is possible to dynamically determine the correct range for each sheet, but doing so is a bit trickier, and beyond the scope of this article.

Next, we start using R. We rely on three packages: readxl (Wickham & Bryan 2018), writexl (Ooms, 2018), and dplyr (Wickham et al., 2020). If you have Rstudio installed, you can follow along in the GitHub repository by opening the file **Guerrilla Automation.Rproj** and then opening the file **Guerrilla Automation.R** from the same instance of Rstudio. You can run the whole process from start to finish by calling either the **main** or the **output_results** functions. Note that you can install the requisite packages using the following code. You also need to source the R script file:

```
install.packages(c("readxl","writexl",
"dplyr"))
source('Guerrilla Automation.R')
```

Step 1 is to import the Input Control data. We do this in the **import_input_control** function with the following lines:

```
input_control = read_excel(filepath, sheet =
"Input Control", range="B4:D9001")
input_control = data.frame(input_con-
trol[!is.na(input_control$Filepath),])
```

Note that the range parameter is excessively large. This helps if you have to add new files to the input control. The second line converts input_control to a data frame and removes the NA values at the end of the data (which occur because of the excessive range). The input_control data frame now looks as shown in Figure 3:

Figure 3
Input_Control Data

**input_control**

| Filepath | Sheet | Range |
|---|---|---|
| Excel Files/Workpaper 1.xlsx | Sheet1 | B4:C13 |
| Excel Files/Workpaper 2.xlsx | Sheet1 | B4:C9001 |
| Excel Files/Workpaper 3.xlsx | An Unusual Sheet Name | C4:D13 |
| Excel Files/Workpaper 3.xlsx | A Mistaken Sheet Name | C4:D13 |
| Excel Files/Workpaper 3 Mistaken Filename.xlsx | An Unusual Sheet Name | C4:D13 |

Figure 4
Data Frames

**input_list[[“input_control”]]**

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Filepath** | **Sheet** | **Range** | **File_Is_Good** | **Error_Message** | | | | | |
| 2 | Excel Files/\ | Sheet1 | B4:C13 | TRUE | | | | | | |
| 3 | Excel Files/\ | Sheet1 | B4:C9001 | TRUE | | | | | | |
| 4 | Excel Files/\ | An Unusual | C4:D13 | TRUE | | | | | | |
| 5 | Excel Files/\ | A Mistaken | C4:D13 | FALSE | Error: Sheet 'A Mistaken Sheet Name' not found | | | | | |
| 6 | Excel Files/\ | An Unusual | C4:D13 | FALSE | Error: `path` does not exist: ‘Excel Files/Workpaper 3 Mistaken Filename.xlsx’ | | | | | |

Figure 5
Data Frames

**input_list[[“input_data”]]**

| | Filepath | Sheet | Range | Account Number | Amount |
|---|---|---|---|---|---|
| 1 | Excel Files/Input 1.xlsx | Sheet1 | B4:C13 | A0000 | 4747 |
| 2 | Excel Files/Input 1.xlsx | Sheet1 | B4:C13 | A1111 | 0 |
| 3 | Excel Files/Input 1.xlsx | Sheet1 | B4:C13 | A2222 | 5228 |
| 4 | Excel Files/Input 1.xlsx | Sheet1 | B4:C13 | A3333 | 0 |
| 5 | Excel Files/Input 1.xlsx | Sheet1 | B4:C13 | A4444 | 4741 |
| 6 | Excel Files/Input 1.xlsx | Sheet1 | B4:C13 | A5555 | 4445 |
| 7 | Excel Files/Input 1.xlsx | Sheet1 | B4:C13 | A7777 | 9560 |
| 8 | Excel Files/Input 1.xlsx | Sheet1 | B4:C13 | A8888 | 0 |
| 9 | Excel Files/Input 1.xlsx | Sheet1 | B4:C13 | A9999 | 0 |
| 10 | Excel Files/Input 2.xlsx | Sheet1 | B4:C9001 | A0000 | 9836 |

Step 2 is to loop through the input_control data frame and use read_excel on the parameters in each row. We do this in the **gather_input_data** function. We also handle our erroneous example rows in the Input Control using a tryCatch. Similar to how we removed the extraneous rows from the Input Control, we also removed the extraneous rows from Workpaper 2. I’ve omitted the code for Step 2 because it is rather lengthy, but it results in a list of two data frames. One is an augmented version of input_control and the other is called input_data. I assign these data frames to input_list via:

```
input_list = list()
# Code omitted
input_list[["input_control"]] = input_
control
input_list[["input_data"]] = input_data
```

Now they are as shown in Figures 4 and 5.

Note the two extra columns in input_control. We can use these to track which files imported successfully and what went wrong with the files that failed.

Step 3 is to summarize the data. We do this in the **aggregate_input_data** function using elegant dplyr functions. We store our results in a list called aggregate_list. We keep the old input_list along with our new summary.

```
aggregate_list = list()
aggregate_list[["input_control"]] = input_
list[["input_control"]]
aggregate_list[["input_data"]] = input_
list[["input_data"]]
aggregate_list[["summary"]] = data.frame(in-
put_list[["input_data"]] %>%
group_by(`Account Number`) %>% sum-
marise(Amount = sum(Amount)))
```

The summary now appears as shown in Figure 6.

Figure 6
Input_Data Summary

**aggregate_list[["summary"]]**

| Account Number | Amount |
|---|---|
| A0000 | 15368 |
| A1111 | 6973 |
| A2222 | 13909 |
| A3333 | 3349 |
| A4444 | 7725 |
| A5555 | 9504 |
| A7777 | 15486 |
| A8888 | 8472 |
| A9999 | 2992 |

Note that these values are exactly the same as the ones we started with in **Ledger Balances.xlsx**. Huzzah!

Step 4 is to output the results to Excel. We do this in the output_results function using the writexl package as follows:

```
write_xlsx(aggregate_list, "./Excel Files/R
Output/Results.xlsx")
```

Now, we're back in Excel! In the file **Excel Files/R Output/Results.xlsx**, we have a tab for each data frame. I chose to output the results into a new Excel file. This prevents issues that occur when you are trying to write to a file that is currently open. If you open the **Results.xlsx** file and run the process again, you should get an error. One way to prevent this is to include a time stamp in the file name via:

```
write_xlsx(aggregate_list, paste("./Excel
Files/R Output/Results ",
format(Sys.time(), "%Y%m%d %s"), ".xlsx",
sep=""))
```

Now we can easily see which files were imported successfully, and we can trace each ledger balance back to the files that contributed to it. Our approach to automating data gathering has a lot of flexibility. Here's a few other things we could do with our solution:

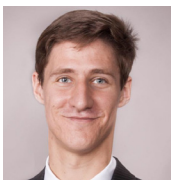
1. Add additional identifier columns to Input Control.xlsx, which can provide helpful splits to our aggregate data.

2. Add functionality that dynamically detects the appropriate range for each input file.

3. Automate the task of comparing the resulting summary to the original ledger balance.

4. Iteratively correct **Input Control.xlsx** using the **input_control** tab in **Results.xlsx** until all of the rows contain correct information.

As you can see, R offers a lot of power and compatibility as a free and open source system. I enjoy using R in both business and as a hobby because of its elegance and versatility. It is a great resource for rapidly developing solutions that can meet the needs of your stakeholder. That's all folks. I hope you enjoyed this "R-ticle" and found it most helpful! ■



Timothy Quast, ASA, is an assistant actuary at AIG. He can be contacted at *timothy.quast@aig.com.*

## REFERENCES

Hadley Wickham and Jennifer Bryan (2018). readxl: Read Excel Files. R package version 1.1.0. *https://CRAN.R-project.org/package=readxl*

Jeroen Ooms (2018). writexl: Export Data Frames to Excel 'xlsx' Format. R package version 1.0. *https://CRAN.R-project.org/package=writexl*

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2020). dplyr: A Grammar of Data Manipulation. R package version 0.8.5. *https://CRAN.R-project.org/package=dplyr*

# Principal Component Analysis Using R

By Soumava Dey

In today's Big Data world, exploratory data analysis has become a stepping stone to discover underlying data patterns with the help of visualization. Due to the rapid growth in data volume, it has become easy to generate large dimensional datasets with multiple variables. However, the growth has also made the computation and visualization process more tedious in the recent era.

The two ways of simplifying the description of large dimensional datasets are the following:

1. Remove redundant dimensions or variables, and
2. retain the most important dimensions/variables.

Principal component analysis (PCA) is the best, widely used technique to perform these two tasks. The purpose of this article is to provide a complete and simplified explanation of principal component analysis, especially to demonstrate how you can perform this analysis using R.

## WHAT IS PCA?

In simple words, PCA is a method of extracting important variables (in the form of components) from a large set of variables available in a data set. PCA is a type of unsupervised linear transformation where we take a dataset with too many variables and untangle the original variables into a smaller set of variables, which we called "principal components." It is especially useful when dealing with three or higher dimensional data. It enables the analysts to explain the variability of that dataset using fewer variables.

## WHY PERFORM PCA?

The goals of PCA are to:

1. Gain an overall structure of the large dimension data,
2. determine key numerical variables based on their contribution to maximum variances in the dataset,
3. compress the size of the data set by keeping only the key variables and removing redundant variables, and
4. find out the correlation among key variables and construct new components for further analysis.

Note that, the PCA method is particularly useful when the variables within the data set are highly correlated and redundant.

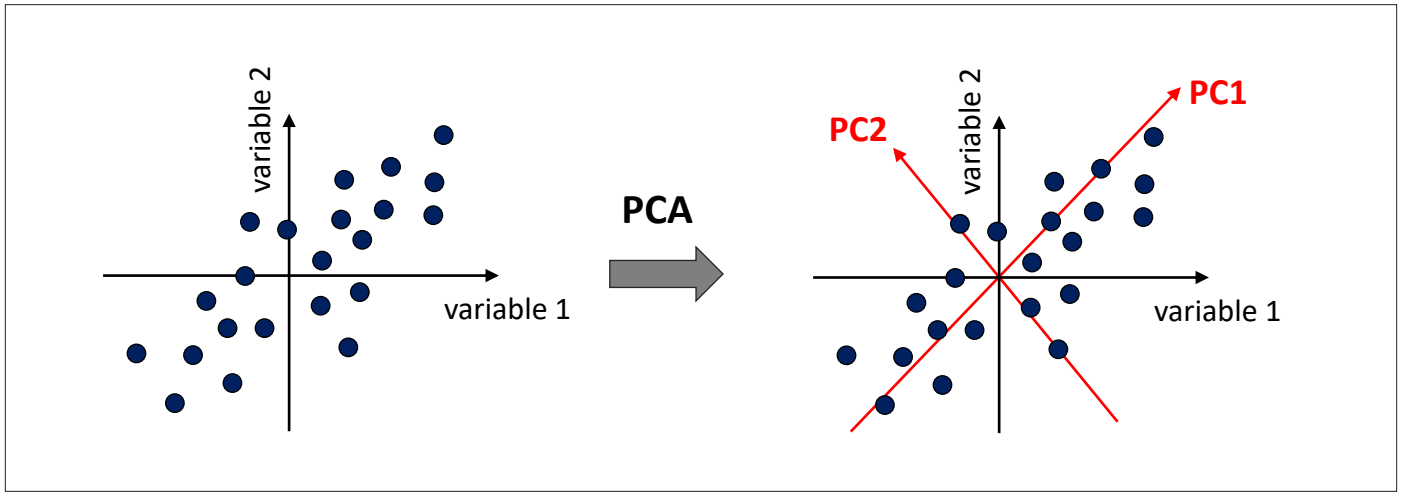
## HOW DO WE PERFORM PCA?

Before I start explaining the PCA steps, I will give you a quick rundown of the mathematical formula and description of the principal components.

### What are Principal Components?

Principal components are the set of new variables that correspond to a linear combination of the original key variables. The number of principal components is less than or equal to the number of original variables.

Figure 1
Principal Components




Source: ourcodingclub.github.io

In Figure 1, the PC1 axis is the first principal direction along which the samples show the largest variation. The PC2 axis is the second most important direction, and it is orthogonal to the PC1 axis.

The first principal component of a data set $X_1,X_2,...,X_p$ is the linear combination of the features

$$Z_1=\phi_{1,1}X_1+\phi_{2,1}X_2+...+\phi_{p,1}X_p$$

$\Phi_{p,1}$ is the loading vector comprising of all the loadings ($\phi_1 \ldots \phi_p$) of the principal components.

The second principal component is the linear combination of $X_1,\ldots, X_p$ that has maximal variance out of all linear combinations that are uncorrelated with $Z_1$. The second principal component scores $z_{1,2},z_{2,2},\ldots,z_{n,2}$ take the form

$$Z_2=\phi_{1,2}X_1+\phi_{2,2}X_2+...+\phi_{p,2}X_p$$

It is necessary to understand the meaning of covariance and eigenvector before we further get into principal components analysis.

## Covariance

Covariance is a measure to find out how much the dimensions may vary from the mean with respect to each other. For example, the covariance between two random variables X and Y can be calculated using the following formula (for population):

$$\text{Cov}(x,y) = \text{SUM } [(xi - xm) * (yi - ym)] / (n - 1)$$

- xi = a given x value in the data set
- xm = the mean, or average, of the x values
- yi = the y value in the data set that corresponds with xi
- ym = the mean, or average, of the y values
- n = the number of data points

Both covariance and correlation indicate whether variables are positively or inversely related. Correlation also tells you the degree to which the variables tend to move together.

## Eigenvectors

Eigenvectors are a special set of vectors that satisfies the linear system equations:

$$Av = \lambda v$$

where A is an (n x n)square matrix, v is the eigenvector, and λ is the eigenvalue. Eigenvalues measure the amount of variances retained by the principal components. For instance, eigenvalues tend to be large for the first component and smaller for the subsequent principal components. The number of eigenvalues and eigenvectors of a given dataset is equal to the number of dimensions that dataset has. Depending upon the variances explained by the eigenvalues, we can determine the most important principal components that can be used for further analysis.

## GENERAL METHODS FOR PRINCIPAL COMPONENT ANALYSIS USING R

Singular value decomposition (SVD) is considered to be a general method for PCA. This method examines the correlations between individuals,

The functions prcomp ()["stats" package] and PCA()["FactoMineR" package] use the SVD.

PCA () function comes from FactoMineR. So, install this package along with another package called Factoextra which will be used to visualize the results of PCA.

In this article, I will demonstrate a sample of SVD method using PCA() function and visualize the variance results.

## Dataset Description

I will explore the principal components of a dataset which is extracted from KEEL-dataset repository.

This dataset was proposed in McDonald, G.C. and Schwing, R.C. (1973) "Instabilities of Regression Estimates Relating Air Pollution to Mortality," *Technometrics*, vol.15, 463-482. It contains 16 attributes describing 60 different pollution scenarios. The attributes are the following:

1. PRECReal: Average annual precipitation in inches
2. JANTReal: Average January temperature in degrees F
3. JULTReal: Same for July
4. OVR65Real: of 1960 SMSA population aged 65 or older
5. POPNReal: Average household size
6. EDUCReal: Median school years completed by those over 22
7. HOUSReal: of housing units which are sound and with all facilities
8. DENSReal: Population per sq. mile in urbanized areas, 1960
9. NONWReal: non-white population in urbanized areas, 1960
10. WWDRKReal: employed in white collar occupations
11. POORReal: of families with income less than $3000
12. HCReal: Relative hydrocarbon pollution potential
13. NOXReal: Same for nitric oxides
14. SO@Real: Same for sulphur dioxide
15. HUMIDReal: Annual average % relative humidity at 1pm
16. MORTReal: Total age-adjusted mortality rate per 100,000

Figure 2
Computer Code for Pollution Scenarios

```
pollution <- read.delim("pollution.dat",
header = FALSE,skip = 19, sep = ",")

colnames(pollution) <- c("PRECReal",
"JANTReal","JULTReal","OVR65Real",
"POPNReal","EDUCReal","HOUSReal",
"DENSReal","NONWReal","WWDRKReal",
"POORReal","HCReal","NOXReal","SO@Real",
"HUMIDReal","MORTReal")

library(dplyr)

pollution <- mutate(pollution,MORTReal_Type
= case_when

  (pollution$MORTReal < 900.0 ~
  "Low Mortality",
  pollution$MORTReal > 900.0 & MORTReal <
  1000.0 ~ "Medium Mortality",
  pollution$MORTReal > 1000.0 ~
  "High Mortality"))
```

The code in Figure 2 loads the dataset to an R data frame and names all 16 variables. In order to define a different range of mortality rate, one extra column named "MORTReal_TYPE" has been created in the R data frame. This extra column will be useful to create data visualization based on mortality rates.

## Compute Principal Components Using PCA ()

PCA () [*FactoMineR* package] function is very useful to identify the principal components and the contributing variables associated with those PCs. A simplified format is:

```
library("FactoMineR")

pollution.PCA <- PCA(pollution[c(-17)],
scale.unit = TRUE, graph = FALSE)
```

- pollution: a data frame. Rows are individuals and columns are numeric variables
- scale.unit: a logical value. If TRUE, the data are scaled to unit variance before the analysis. This standardization to the same scale avoids some variables to become dominant just because of their large measurement units. It makes the variable comparable.
- graph: a logical value. If TRUE a graph is displayed.

The output of the function PCA () is a list that includes the following components:

```
>pollution.pca
""Results for the Principal Component Analysis (PCA)**
The analysis was performed on 60 individuals, described by 16 variables
*The results are available in the following objects:

Name                    description
1  "$eig"               "eigenvalues"
2  "$var"               "results for the variables"
3  "$var$coord"         "coord. for the variables"
4  "$var$cor"           "correlations variables - dimensions"
5  "$var$cos2"          "cos2 for the variables"
6  "$var$contrib"       "contributions of the variables"
7  "$ind"   `           "results for the individuals"
8  "$ind$coord"         "coord.for the individuals"
9  "$ind$cos2"          "cos2 for the individuals"
10 "$ind$contrib"       "contributions of the individuals"
11 "$call"              "summary statistics"
12 "$call$centre"       "mean of the variables"
13 "$call$ecart.type"   "standard error of the variables"
14 "$call$row.w"        "weights for the individuals"
15 "$call$col.w"        "weights for the variables"
```

For better interpretation of PCA, we need to visualize the components using R functions provided in factoextra R package:

get_eigenvalue(): Extract the eigenvalues/variances of principal components
fviz_eig(): Visualize the eigenvalues
fviz_pca_ind(), fviz_pca_var(): Visualize the results individuals and variables, respectively.

## EIGENVALUES

As described in the previous section, eigenvalues are used to measure the variances retained by the principal components. First principal component keeps the largest value of eigenvalues and the subsequent PCs have smaller values. To determine the eigenvalues and proportion of variances held by different PCs of a given data set we need to rely on the R function get_eigenvalue() that can be extracted from the factoextra package.

```
library("factoeextra")

eig.val <- get_eigenvalue(pollution.PCA)
eig.val
```

```
  "eig.val"
  eigenvalue variance.percent cumulative.variance.percent
  Dim.1  4.878595616      30.49122260                    30.49122
  Dim.2  2.766574422      17.29109013                    47.78231
  Dim.3  2.292475683      14.32797302                    62.11029
  Dim.4  1.351660343       8.44787715                    70.55816
  Dim.5  1.223507408       7.64692130                    78.20508
  Dim.6  1.086738477       6.79211548                    84.99720
  Dim.7  0.661476260       4.13422662                    89.13143
  Dim.8  0.479425447       2.99640904                    92.12784
  Dim.9  0.407500850       2.54688031                    94.67472
  Dim.10 0.244819892       1.53012432                    96.20484
  Dim.11 0.194097702       1.21311064                    97.41795
  Dim.12 0.156401959       0.97751224                    98.39546
  Dim.13 0.116810134       0.73006334                    99.12553
  Dim.14 0.089284390       0.55802744                    99.68355
  Dim.15 0.045962000       0.28726250                    99.97082
  Dim.16 0.004669417       0.02918386                   100.00000
```

The sum of all the eigenvalues gives a total variance of 16.

The proportion of all the eigenvalues is demonstrated by the second column "variance.present." For example, if you divide 4.878 by 16 equals to 0.304875, i.e., almost 30.49 percent variance explained by the first component/dimension. Based on the output of eig.val object, we can derive the fact that the first six eigenvalues keep almost 82 percent of total variances existed in the dataset.

As an alternative approach, we can also examine the pattern of variances using a scree plot which showcases the order of eigenvalues from largest to smallest. In order to produce the scree plot (see Figure 3), we will use the function fviz_eig() available in factoextra() package:

```
fviz_eig(pollution.pca, addlabels = TRUE,
hjust = -0.3, ylim = c(0,35))
```

From the scree plot above, we might consider using the first six components for the analysis because 82 percent of the whole dataset information is retained by these principal components.

## VARIABLES CONTRIBUTION GRAPH

The next step is to determine the contribution and the correlation of the variables that have been considered as principal components of the dataset. In order to extract the relationship of the variables from a PCA object we need to use the function get_pca_var () which provides a list of matrices containing all the results for the active variables (coordinates, correlation between variables, squared cosine and contributions).

```
var_pollution <- get_pca_var(pollution.PCA)
var_pollution
```

```
> var_pollution.pca
Principal Component Analysis Results for
varibles

Name          description
1 "$coord"    "Coordinates for the variables"
2 "$cor"      "Correlations between variables
               and dimensions"
3 "$cos2"     "Cos2 for the variables"
4 "$contrib"  "contributions of the variables"
```

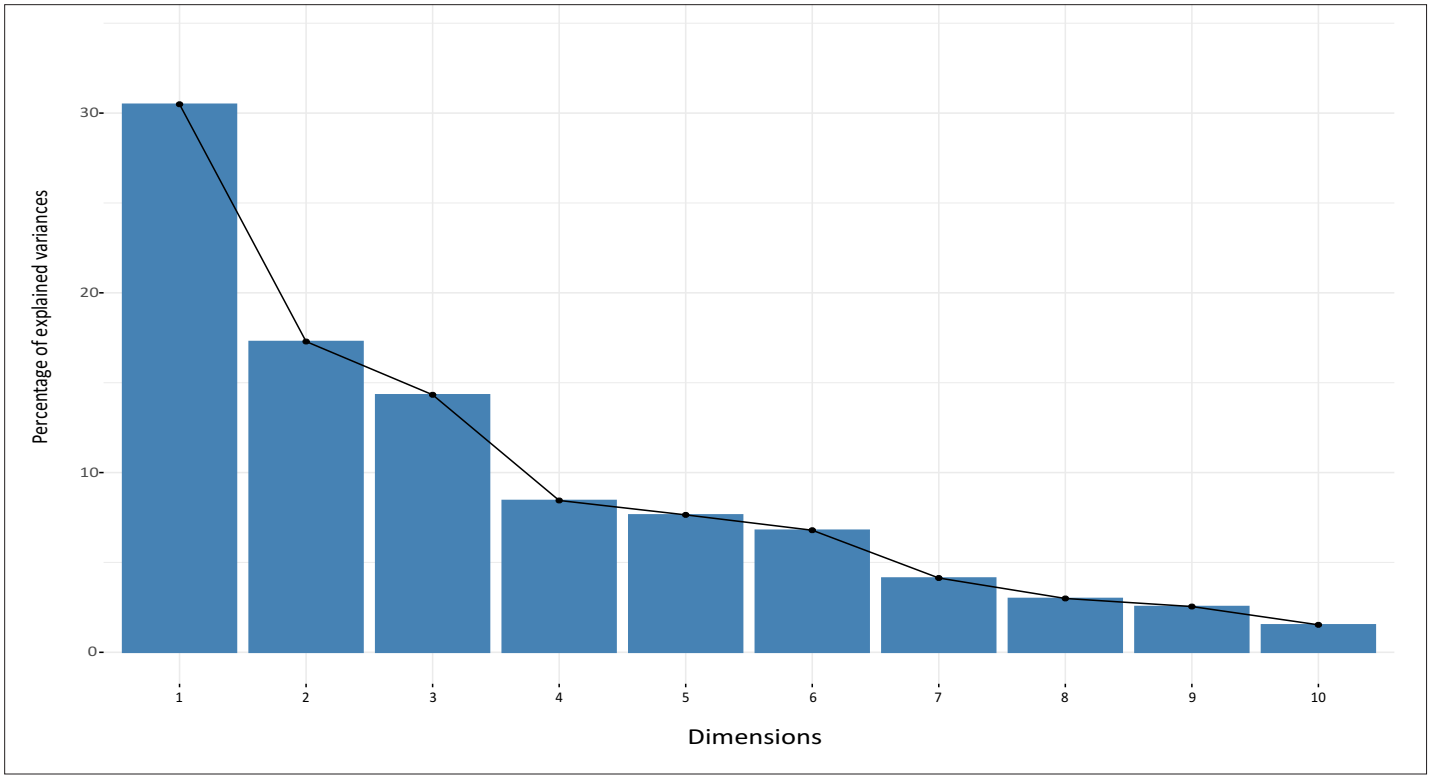
## CORRELATION CIRCLE PLOT

We can apply different methods to visualize the SVD variances in a correlation plot in order to demonstrate the relationship between variables. The correlation between a variable and a principal component (PC) is used as the coordinates of the variable on the PC.

```
# Coordinates of Variables

head(var_pollution$contrib)
```

Figure 3
Scree Plot



```
> head(var_pollution$conrib)
               Dim.1      Dim.2        Dim.3      Dim.4      Dim.5
PRECReal  11.3363777  1.2901207 2.320962e-04 10.5955205  1.6692242
JANTReal   0.3312333 21.4926762 3.234525e+00 10.8905057  0.6850515
JULTReal  10.2768749  2.7936599 2.838199e+00  0.1211819 15.3922039
OVR65Real  2.4116845 11.8740295 3.795171e+00 27.4577926  0.1384696
POPNReal   8.1452038  0.5965791 3.132326e-03 30.1085931  5.6756771
EDUCReal   8.6313043  2.4816964 1.212178e+01  2.0276052
```

To plot all the variables we can use fviz_pca_var() :

```
fviz_pca_var(pollution.PCA,col.var = "black")
```

Figure 4
Relationship Between Variables

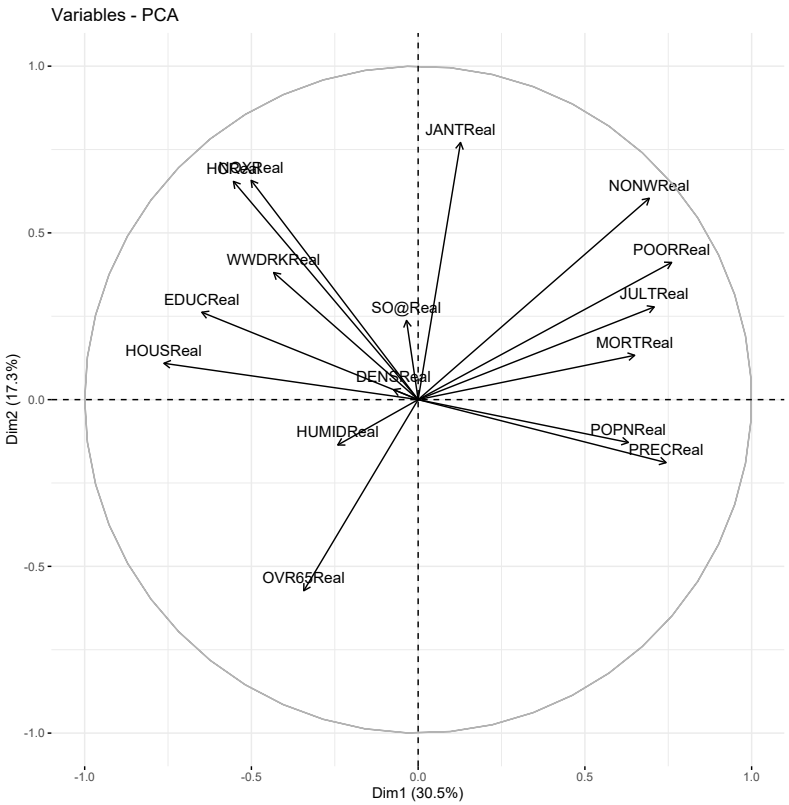



Figure 4 shows the relationship between variables in three different ways:

- Positively correlated variables are grouped together.
- Negatively correlated variables are located on opposite sides of the plot origin
- The distance between variables and the origin measures the quality of the variables on the factor map. Variables that are away from the origin are well represented on the factor map.

## QUALITY OF REPRESENTATION

This shows the quality of representation of the variables on the factor map called cos2, which is multiplication of squared cosine and squared coordinates. The previously created object var_pollution holds cos2 value:

```
head(var_pollution$cos2)
```

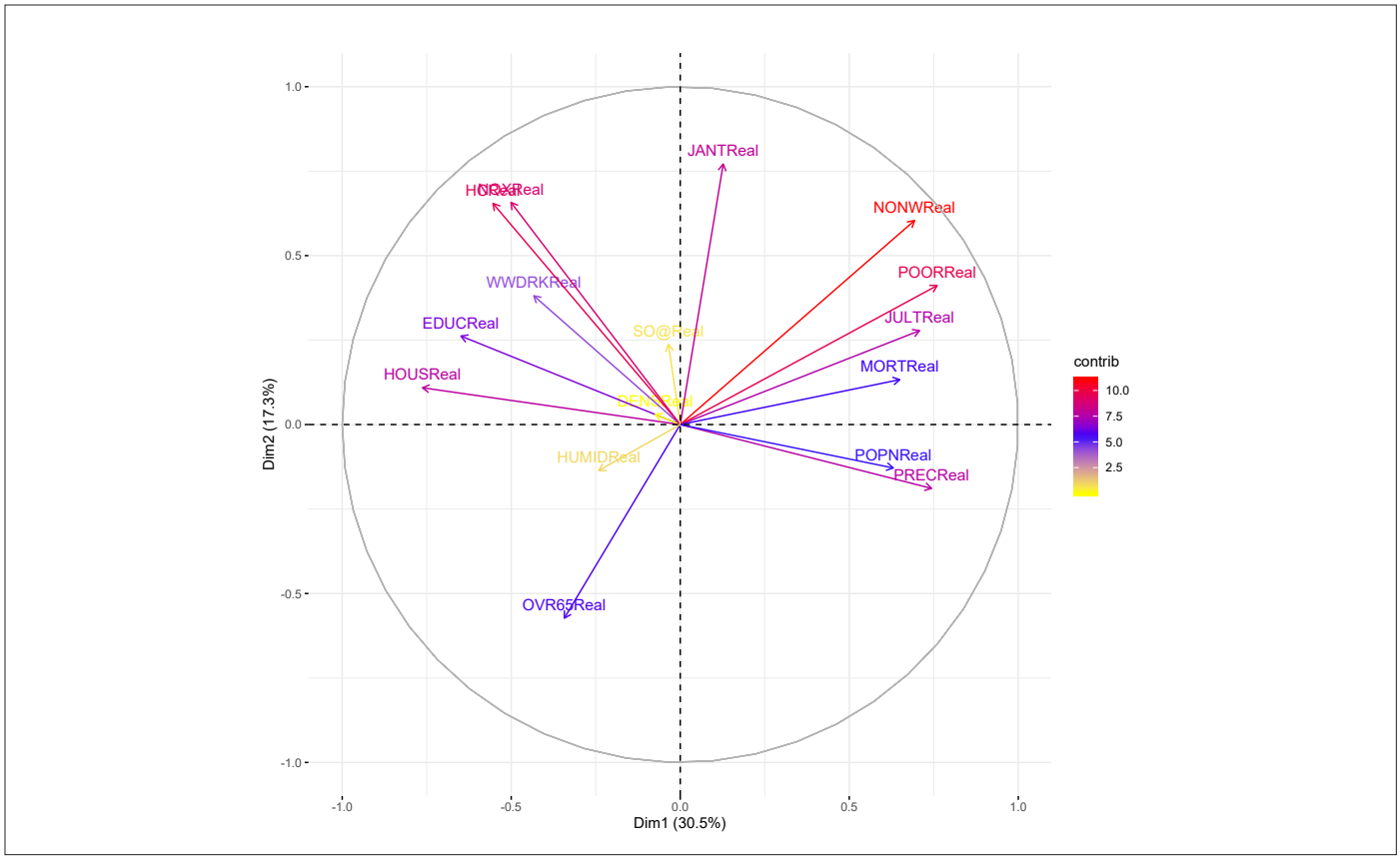
```
head(var_pollution$cos2)
               Dim.1      Dim.2        Dim.3       Dim.4       Dim.5
PRECReal   0.55305602 0.03569215 5.320748e-06 0.143215449 0.020423082
JANTReal   0.01615953 0.59461088 7.415070e-02 0.147202647 0.008381656
JULTReal   0.50136717 0.07728868 6.506502e-02 0.001637968 0.188324755
OVR65Real  0.11765633 0.32850386 8.700336e-02 0.371136093 0.001694186
POPNReal   0.39737155 0.01650481 7.180782e-05 0.406965913 0.069442330
EDUCReal   0.42108643 0.06865798 2.778888e-01 0.027406335 0.025652309
```

A high cos2 indicates a good representation of the variable on a particular dimension or principal component. Whereas, a low cos2 indicates that the variable is not perfectly represented by PCs.

Cos2 values can be well presented using various aesthetic colors in a correlation plot. For instance, we can use three different colors to present the low, mid and high cos2 values of variables that contribute to the principal components.

```
fviz_pca_var(pollution.PCA,col.var = "cos2",
             gradient.cols = c("green","blue","red"),
             repel = TRUE # Avoid text overlapping
```

Figure 5
Variables—PCA



Variables that are closed to circumference (like NONWReal, POORReal and HCReal ) manifest the maximum representation of the principal components. However, variables like HUMIDReal, DENSReal and SO@Real show week representation of the principal components.

## CONTRIBUTION OF VARIABLES TO PCS

After observing the quality of representation, the next step is to explore the contribution of variables to the main PCs. Variable contributions in a given principal component are demonstrated in percentage.

Key points to remember:

- Variables with high contribution rate should be retained as those are the most important components that can explain the variability in the dataset.
- Variables with low contribution rate can be excluded from the dataset in order to reduce the complexity of the data analysis.

The function fviz_contrib() [factoextra package] can be used to draw a bar plot of variable contributions. If your data contains many variables, you can decide to show only the top contributing variables. The R code (see code 1 and Figures 6 and 7) below shows the top 10 variables contributing to the principal components:

Code 1

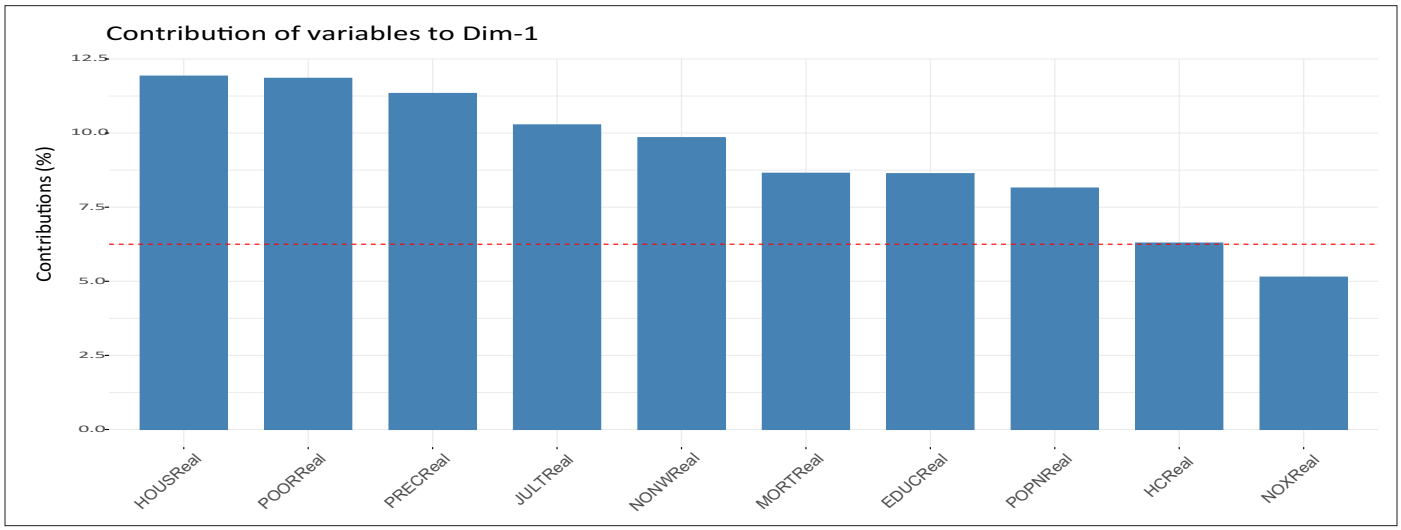
```
#Contributions of variables to PC1
fviz_contrib(pollution.PCA, choice = "var",
axes = 1, top = 10)
#Contribution of variables to PC2
fviz_contrib(pollution.PCA, choice = "var",
axes = 2, top = 10)
```

The most important (or, contributing) variables can be highlighted on the correlation plot as in code 2 and Figure 8.

Code 2

```
fviz_pca_var (pollution.pca, col.var =
"contrib",
      Gradient.cols = c("yellow", "blue",
      "red")
```

Figures 6 and 7
Top 10 Variables Contributing to Principal Components



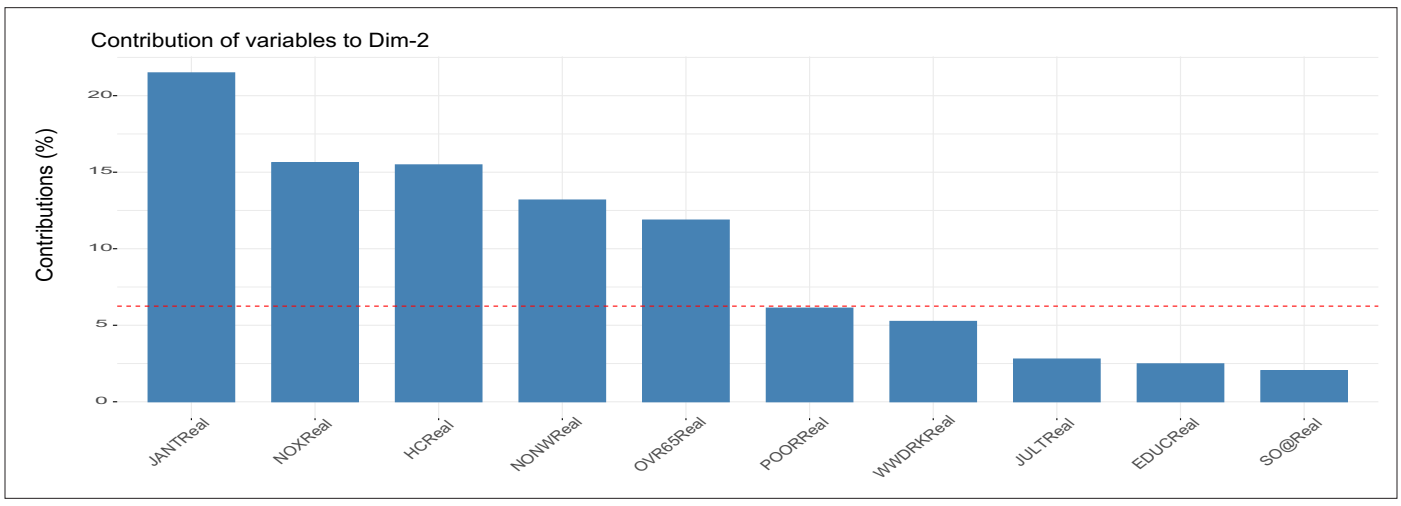
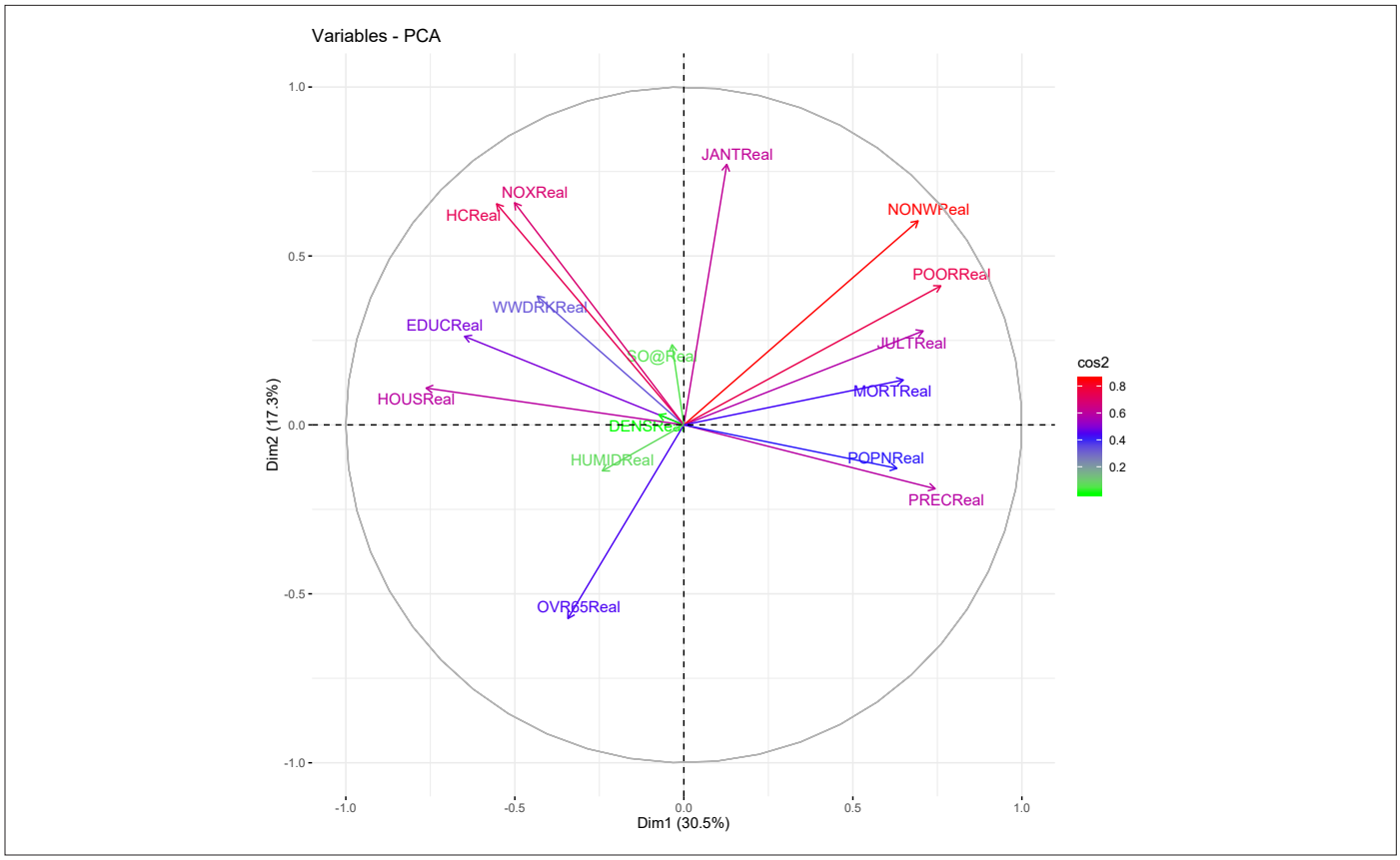



Figure 8
Graphical Display of the Eigen Vector and Their Relative Contribution



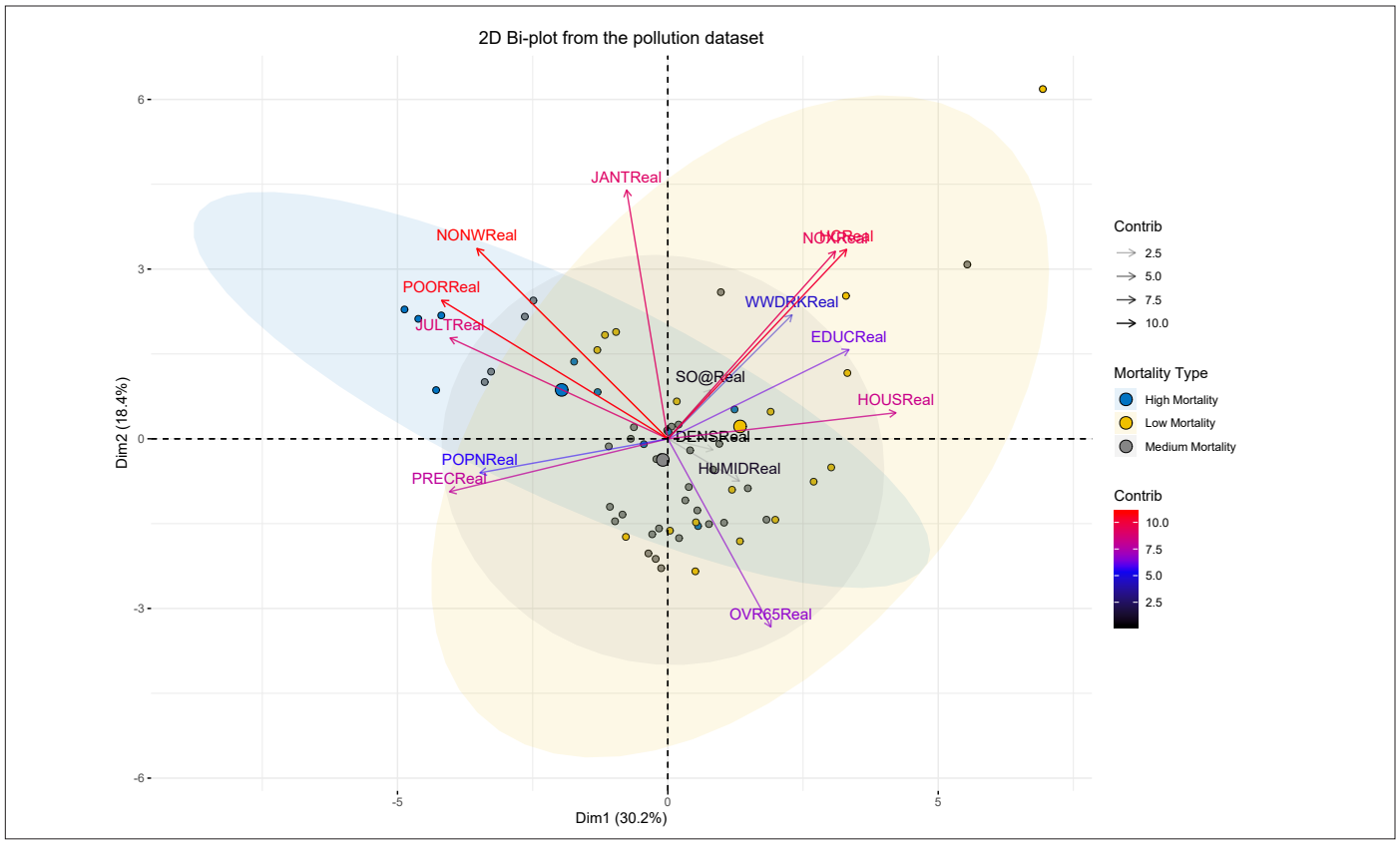

## BIPLOT

To make a simple biplot of individuals and variables, type this:

Code 3

```
fviz_biplot (pollution.pca,
      col.ind = pollution$MORTReal_TYPE, palette = "jco",
      addEllipses = TRUE, label = "var"
      col.var = "black", repel = TRUE,
      legend.title = "Mortality_Range")
```

Figure 9
Mortality Rate Value and Corresponding Key Variables Grouped




In Figure 9, column "MORTReal_TYPE" has been used to group the mortality rate value and corresponding key variables.

## SUMMARY

PCA analysis is unsupervised, so this analysis is not making predictions about pollution rate, rather simply showing the variability of dataset using fewer variables. Key observations derived from the sample PCA described in this article are:

1. Six dimensions demonstrate almost 82 percent variances of the whole data set.

2. The following variables are the key contributors to the variability of the data set:
   NONWReal, POORReal, HCReal, NOXReal, HOUSReal and MORTReal.

3 Correlation plots and Bi-plot help to identify and interpret correlation among the key variables.

### For Python Users

To implement PCA in python, simply import PCA from sklearn library. The code interpretation remains the same as explained for R users above.

## INDUSTRY APPLICATION USE

PCA is a very common mathematical technique for dimension reduction that is applicable in every industry related to STEM (science, technology, engineering and mathematics). Most importantly, this technique has become widely popular in areas of quantitative finance. For instance, fund portfolio managers often use PCA to point out the main mathematical factors that drive the movement of all stocks. Eventually, that helps in forecasting portfolio returns, analyzing the risk of large institutional portfolios and developing asset allocation algorithms for equity portfolios.

PCA has been considered as a multivariate statistical tool which is useful to perform the computer network analysis in order to identify hacking or intrusion activities. Network traffic data is typically high-dimensional making it difficult to analyze and visualize. Dimension reduction technique and Bi-plots are helpful to understand the network activity and provide a summary of possible intrusions statistics. Based on a study conducted by UC Davis, PCA is applied to selected network attacks from the DARPA 1998 intrusion detection datasets namely: Denial-of-Service and Network Probe attacks.

Multidimensional reduction capability was used to build a wide range of PCA applications in the field of medical image processing such as feature extraction, image fusion, image compression, image segmentation, image registration and de-noising of images.

Using the multivariate analysis feature of PCS efficient properties it can identify patterns in data of high dimensions and can serve applications for pattern recognition problems. For example, one type for PCA is the Kernel principal component analysis (KPCA) which can be used for analyzing ultrasound medical images of liver cancer ( Hu and Gui, 2008). Compared with the experiments of wavelets, the experiment of KPCA showed that KPCA

is more effective than wavelets especially in the application of ultrasound medical images.

## CONCLUSION

This tutorial gets you started with using PCA. Many statistical techniques, including regression, classification, and clustering can be easily adapted to using principal components.

PCA helps to produce better visualization of high dimensional data. The sample analysis only helps to identify the key variables that can be used as predictors for building the regression model for estimating the relation of air pollution to mortality. My article does not outline the model building technique, but the six principal components can be used to construct some kind of model for prediction purposes.

### Further Reading

PCA using prcomp() and princomp() (tutorial). *http://www.sthda.com/english/wiki/pca-using-prcomp-and-princomp*

PCA using ade4 and factoextra (tutorial). *http://www.sthda.com/english/wiki/pca-using-ade4-and-factoextra* ■

*Soumava Dey is an actuarial systems analyst at AIG. He can be contacted at soumava.dey@aig.com.*

### REFERENCE

Husson, Francois, Sebastien Le, and Jérôme Pagès. 2017. Exploratory Multivariate Analysis by Example Using R. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. *http://factominer.free.fr/bookV2/index.html.*

Abdi, Hervé, and Lynne J. Williams. 2010. "Principal Component Analysis." John Wiley and Sons, Inc. WIREs Comp Stat 2: 433–59. *http://staff.ustc.edu.cn/~zwp/teach/MVA/abdi-awPCA2010.pdf.*

KEEL-dataset citation paper: J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework." Journal of *Multiple-Valued Logic and Soft Computing* 17:2-3 (2011) 255-287.

Khaled Labib and V. Rao Vemuri."An Application of Principal Component Analysis to the Detection and Visualization of Computer Network Attacks." *https://web.cs.ucdavis.edu/~vemuri/papers/pcaVisualization.pdf*

Libin Yang. 2015. "An Application of Principal Component Analysis to Stock Portfolio Management." *https://ir.canterbury.ac.nz/bitstream/handle/10092/10293/thesis.pdf*

*https://www.researchgate.net/publication/272576742_Principal_Component_Analysis_in_Medical_Image_Processing_A_Study*

*https://rdrr.io/cran/factoextra/man/fviz_pca.html*