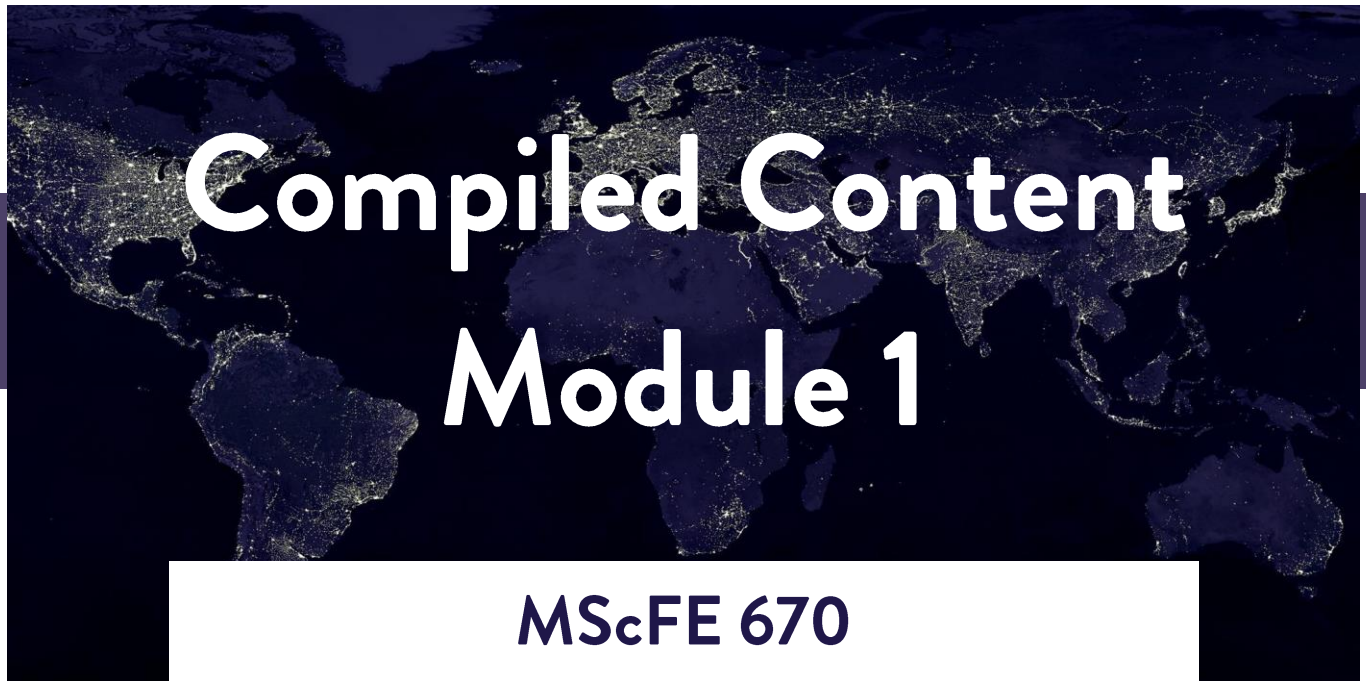
WORLDQUANT UNIVERSITY



Compiled Content Module 1

MScFE 670

Data Feeds and Technology

Revised: 09/07/2019

Table of Contents

Module 1: R for Data Science

Module 1 begins by introducing data science and examining the workflow of a typical data science project. The processes of data cleaning and data analysis using R are discussed and the principles to build a basic data model introduced. The module concludes by discussing techniques to communicate results and presenting tools such as Shiny to visualize data in an organized and intuitive manner.

Unit 1: Introduction to Data Science

Welcome to the R for Data Science module. In this module, we are going to work through a real-life example from start to finish to completely immerse you in the data science process. Once you have completed this module, you will understand what exactly data scientists do and the full workflow of a typical data science project. You should also start to get a good idea of what part of the data science process you enjoy most.

First things first: what data are we going to work with? Under course resources, you will find an attached file – CountryData.xlsx – which contains information on 192 countries supplied by the World Bank in 2010. This information contains 19 different variables, from population growth to GDP per capita, among others. For now, start by simply looking at the data. Explore it and try to understand what the different variables are. Let your mind wander and think of the possibilities of what we can do with it. Keep the data in mind as we start the next section where we explore the roles of a data scientist in more detail.

Roles of a data scientist and the data science process

The diagram below provides an overview of the data science process.

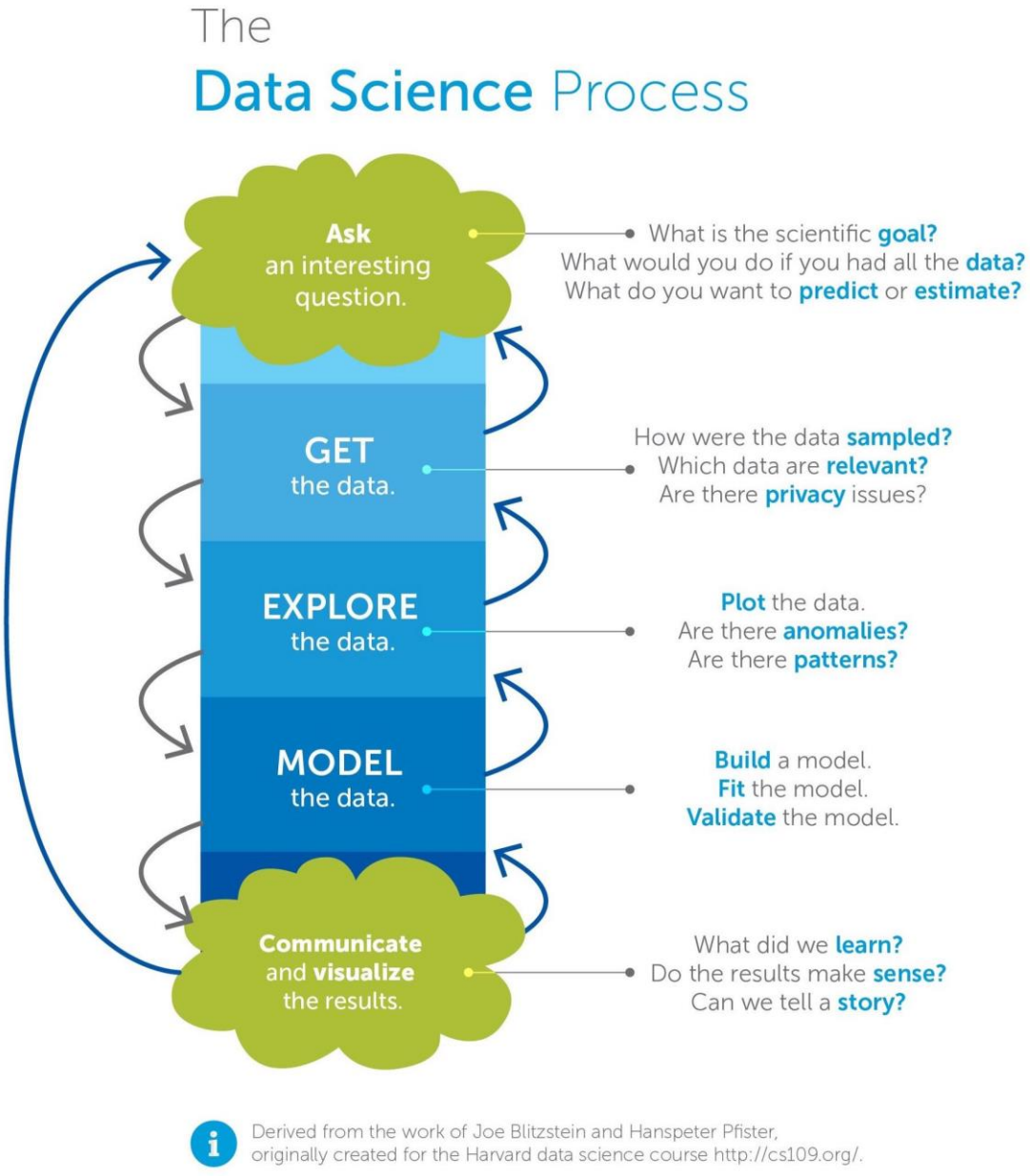


Figure 1: The Data Science Process

As you can see, data science is not simply just about building a neural network or some other complicated model. Rather, it is an intricate process that looks to use data to answer important questions in a statistically valid way.

Generally, a well-rounded data scientist should be able to successfully perform all of the roles detailed in Figure 1, or they should at least have a basic understanding of each of them. Often, however, when a data scientist is contracted for a specific job, they may have a specific role and simply focus on one specific part of the process. This module aims to help you work through the entire data science process so that, more than just gaining knowledge of each role, you can start to identify which role you enjoy most. Let's now talk a bit more about each step of the process.

Ask an interesting question

The first part involves getting curious. You can imagine you had the exact data you wanted and then think of what you could do with it. You could also imagine the type of problem you are trying to solve. Before we get into some examples, though, let's make a distinction between the two main types of data science projects we would generally encounter.

The **supervised learning problem** involves using data with features or variables and given labels to make predictions. If we imagined data used to predict fraudulent credit card transactions, features would be the value of a transaction, the time it took place, and so on. The label, on the other hand, would be whether the transaction was fraudulent or not.

The **unsupervised learning problem** involves using data with just features or variables for inference, and trying to find some meaningful structure in the data.

Now on to some examples of asking interesting questions.

Supervised learning

- Could I predict which insurance claims were fraudulent?
- Could I predict handwritten digits successfully?
- Could I predict who would have survived the Titanic sinking?

Kaggle.com has many more engaging challenges where you can build models to predict these exact things and build your skills.

Unsupervised learning

- Are there different groups of customers at my store?
- What different groups of soccer players are there?
- Are there meaningful clusters of countries?

We are going to focus in on this last question to help us explore the nature of data science.

Getting the data

This is possibly the most challenging part of the entire data science process. Without data, there is no project. It is as simple as that. In this module, we are not going to focus on how to get the data, as entire books are written on this topic, but it is important to briefly highlight the challenges of this process.

Data comes in many different forms, and sometimes it does not actually exist yet. Imagine you begin working for insurance company that is concerned with predicting fraudulent claims. In order to make accurate predictions, start to think of the exact data you would need and how you would capture this.

As a data scientist, this process of determining what data you require and the means by which it will be captured should always be in your mind. This is because you won't have a project without data.

When dealing with large-scale corporations, sifting through data that is stored in different places and using different standards becomes a very difficult task. Generally, the more legacy attached to a company, the more you will struggle to easily interface and extract data necessary for your project.

Exploring the data (and cleaning it)

Assuming we now have our data, we need first to explore and clean it before building any models. This is since we will likely have missing data, duplicate values, redundant data frames, and other errors that will make it difficult for us to simply start building models. We first need to take care of all these issues, so our data is the cleanest and simplest possible form, thereby giving ourselves the best possible chance of producing a statistically meaningful model.

We are going to focus on this important part of the data science process in the second set of notes.

Model building

Model building involves using appropriate statistical techniques to build some statistical model that utilizes your cleaned data from the previous step in order to predict an outcome or make some inference. Some examples of the statistical techniques used to build these models include: linear regression, random forests, gradient boosting, support vector machines, neural networks, clustering, self-organizing maps, and many more.

As well as being perhaps the most exciting aspect of data science, model building is also one of the most complicated parts of the process and requires rigorous statistical knowledge to ensure a valid and accurate model.

Luckily for us, programming languages like R and Python have many statistical packages that allow us to build many of these complex models in just a few lines of simple code. It is important to note, however, that these statistical techniques should never be treated as a black box. Even a basic understanding of the underlying statistical technique will allow you to better understand whether this technique is suitable for your data and how you could perhaps tweak the hyperparameters involved in the relevant statistical technique in order to improve the performance of your model.

In the third set of notes, we are going to focus on building a model for the country data.

Data visualization

The reality is, even if you manage to get high quality data and build an effective model, if you cannot communicate the results to users or effectively visualize the data, people will often be hesitant to use your model.

A core component of the data science process is that of taking complex, multi-dimensional data and creating a visualization that can easily and efficiently communicate the data and/or model to a wide audience.

To give you an example, here is a visual created from JSON Cryptokitty data to help effectively visualize the Cryptokitties genome mapping and how traits are passed on from different generations.

```
        "blockNumber": 4606120,
        "timeStamp": 1511428737,
        "gasPrice": 1100000000,
        "gasUsed": 138904,
        "logIndex": 26,
        "transactionHash": "0xae1d64d6d1e35652f1c88760bafaf800efe886917518787d4c500ffe36ab587f",
        "transactionIndex": 35,
        "txLogs": [
        {
        "type": "Birth",
        "owner": "0xba52c75764d6f594735dc735be7f1830cdf58ddf",
        "kittyId": "994",
        "matronId": "64256",
        "sireId": "426",
        "genes":
"100101011010010100100111001000010100101010010001100001011100100001000011110100100011001010001100000001100011000100001000111001000011100111100000100110000000011000110001100010100110000100011000101000100100001110010010011100110001110011100110101110"
        },
```

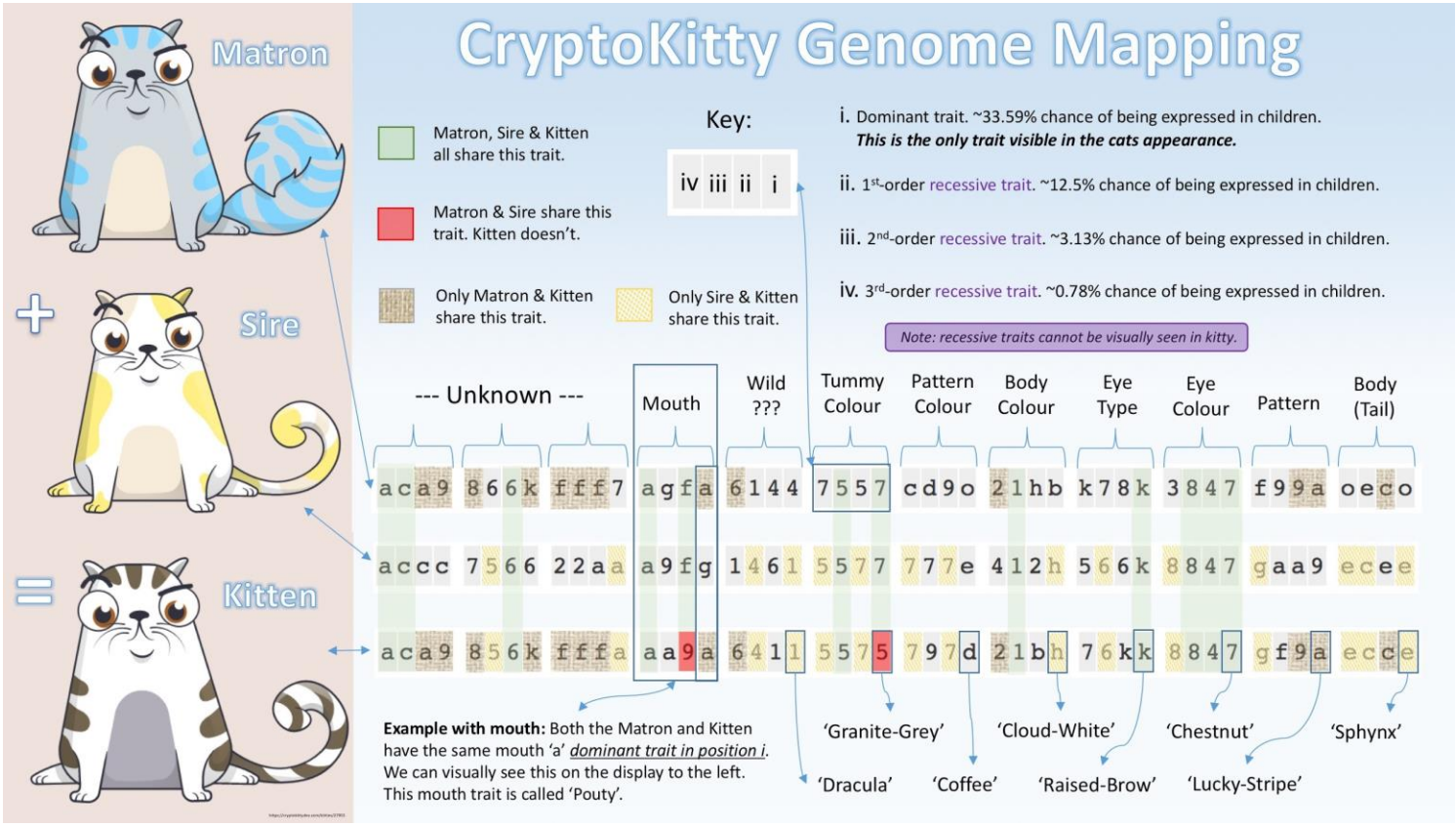



Figure 2: CryptoKitty Genome Mapping

Summary

In this first module, we have gone through the data science process. First, we imagine the problem we would like to solve or what we would like to explore. Then we aim to get the data that will help us to achieve this goal. We then explore the data to ascertain its applicability and gain initial insights. We then clean the data and use this to build a certain model using the data. Once we are sure of the statistical validity of our model's results, it is up to us to effectively communicate these results to stakeholders. This is the data science process in a nutshell.[1]

[1] If you are still interested in learning more about this, take a look at the full article I wrote https://medium.com/newtown-partners/cryptokitties-genome-mapping-6412136c0ae4'

Unit 2: Data Cleaning and Analysis

Introduction

In this module, we are going to use the statistical programming language R for our data science project. R is well-suited for getting up and running quickly. In the Econometrics module, you will have done an introduction to R, so we are going to assume basic familiarity with the language. If at any point you are feeling stuck, there is a wealth of free online R resources available to help you. Particularly, I have found https://www.r-bloggers.com/ to provide great material.

You should have R and R Studio installed on your laptop from the Econometrics module. If not, be sure to refer back to that module and follow the simple download and installation instructions.

Getting our data into R

First things first: let's get our data into our R environment. As previously mentioned, we will be working with the file, CountryData.xlsx, which you will find in resources. We first need to download this file and ensure that it is in the correct directory for R to use it. The easiest way to do this is to place this file in the same directory as your R script. Then you can easily click:

Session > Set Working Directory > To Source File Location

In order to read the Excel file, you may use a different package depending on what operating system (OS) your machine is running – i.e. Mac, Windows, or Ubuntu. On a Mac, for instance, the package *gdata* plays well with my system. You might need to do a little Googling to find what works for your system. Remember, to install any package use the command:

```
install.packages("gdata")
```

Then, to get the data in, ensure you load the required package. This can be done nicely using the following command:

```
require(gdata)
```

Now we are finally ready to import our data.

Here it is useful to note that whenever you are looking for help on a function available in the package, type, `?package_name`, into the console. The help window will pop up and you can explore the relevant functions and the exact syntax and input required. On a Mac, the function below is used in order to get the data:

```
# Getting the excel data in
df = read.xls (("CountryData.xlsx"), sheet = 1, header = TRUE)
# Removing blank rows generated
df<- df[c(1:21)]
```

If at any point you are unsure, the R script currently being used will be made available to you. The data is now housed in the data frame, called *df*, and we are ready to see what we have got.

The command we'll use is:

```
head(df,6)
```

This shows the first six entries of our data and helps us to get a preview or high-level view of what our data set looks like.

	Country <fctr>	CountryName <fctr>	ForeignInvestment <dbl>	ElectricityAccess <dbl>	RenewableEnergy <dbl>	CO2Emission <dbl>	Inflation <dbl>	MobileSubscriptions <dbl>	InternetUse <dbl>	Exports <dbl>
1	ABW	Aruba	7.5681598	93.35629	5.4647159	24.670529	2.0777390	129.63637	62.0	61.50245
2	AFG	Afghanistan	0.3400968	42.70000	14.8398060	0.293837	0.8925369	35.46777	4.0	10.02385
3	AGO	Angola	-3.9131508	33.51495	54.1938366	1.243406	14.4705412	40.23840	2.8	62.38774
4	ALB	Albania	9.1377905	100.00000	37.1153298	1.578574	3.5522674	91.56093	45.0	27.97947
5	ARE	United Arab Emirates	3.0346200	100.00000	0.1081219	19.443690	0.8789368	132.10539	68.0	77.71304
6	ARG	Argentina	2.6751617	98.82000	8.9614460	4.558500	10.7801154	138.46898	45.0	18.93382

Figure 3: First 6 rows of our dataset

Now that we have data, let's explore it.

Data analysis and cleaning

Given variables

The first thing we are generally interested in doing is ascertaining the size and depth of our data. Through understanding the number of observations we have, we can begin to understand what insights we might be able to draw from the data. Our data size will inform whether our results can be statistically valid and what models might be best to use. We'll use the following command to get this insight:

```
str(df)
```

You will see we have 192 observations – i.e. the countries – of 21 variables. The first two variables are the country names, and the remaining 19 variables are the things we are interested in. You will see that they currently have special abbreviated names.

If you go to the third sheet in the Excel file, you will see what these names correspond to. These names can also be found in Figure 4 below.

Indicator Code	Description	Units
BX.KLT.DINV.WD.GD.ZS	Foreign direct investment, net inflows (% of GDP)	%
NE.EXP.GNFS.ZS	Exports of goods and services (% of GDP)	%
NE.IMP.GNFS.ZS	Imports of goods and services (% of GDP)	%
NY.GDP.PCAP.CD	GDP per capita (current US$)	$
FP.CPI.TOTL.ZG	Inflation, consumer prices (annual %)	%
EG.ELC.ACCS.ZS	Access to electricity (% of population)	%
EG.FEC.RNEW.ZS	Renewable energy consumption (% of total final energy consumption)	%
EN.ATM.CO2E.PC	CO2 emissions (metric tons per capita)	tons
IT.CEL.SETS.P2	Mobile cellular subscriptions (per 100 people)	#
IT.NET.USER.ZS	Individuals using the Internet (% of population)	%
SP.DYN.AMRT.FE	Mortality rate, adult, female (per 1,000 female adults)	#
SP.DYN.AMRT.MA	Mortality rate, adult, male (per 1,000 male adults)	#
SP.DYN.CBRT.IN	Birth rate, crude (per 1,000 people)	#
SP.DYN.CDRT.IN	Death rate, crude (per 1,000 people)	#
SP.DYN.IMRT.IN	Mortality rate, infant (per 1,000 live births)	#
SP.DYN.LE00.IN	Life expectancy at birth, total (years)	#
SP.DYN.TFRT.IN	Fertility rate, total (births per woman)	#
SP.POP.GROW	Population growth (annual %)	%
SP.URB.TOTL.IN.ZS	Urban population (% of total)	%

Figure 4: Variable description

Variable measurement scales

As you can see, variables are measured on different scales. This is very important to take into consideration. If left as is, the different measurement scales would greatly affect our clustering model that we will build in the next section. Let's examine this more by examining a basic boxplot of the variables.

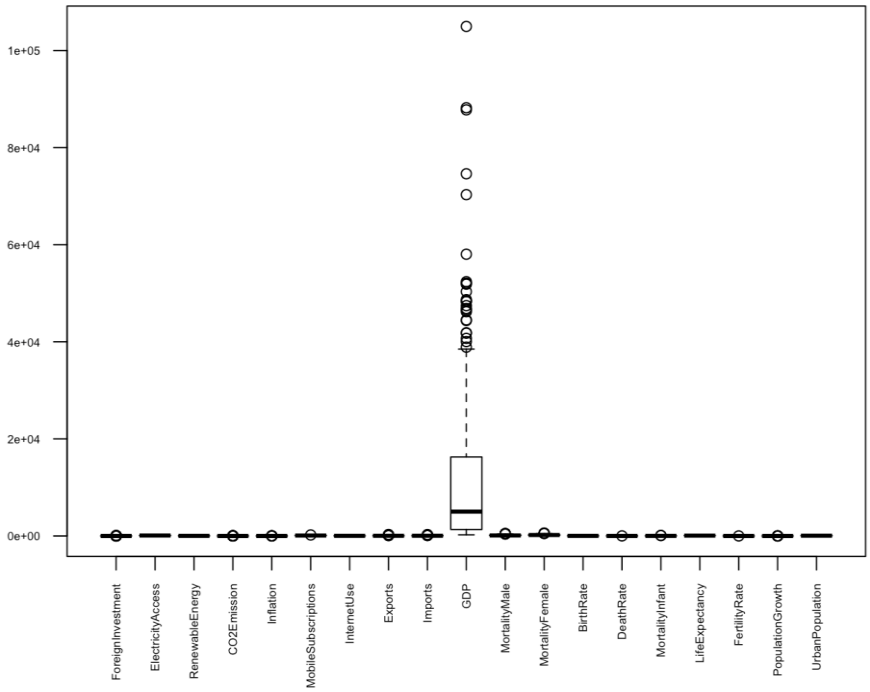


Figure 5: GDP

As can be seen, the GDP figure absolutely dwarfs all the other figures. This is because GDP per capita is measured in dollars, whereas many other figures are measured as percentages. Let's ignore GDP for now and examine a boxplot of the remaining variables.

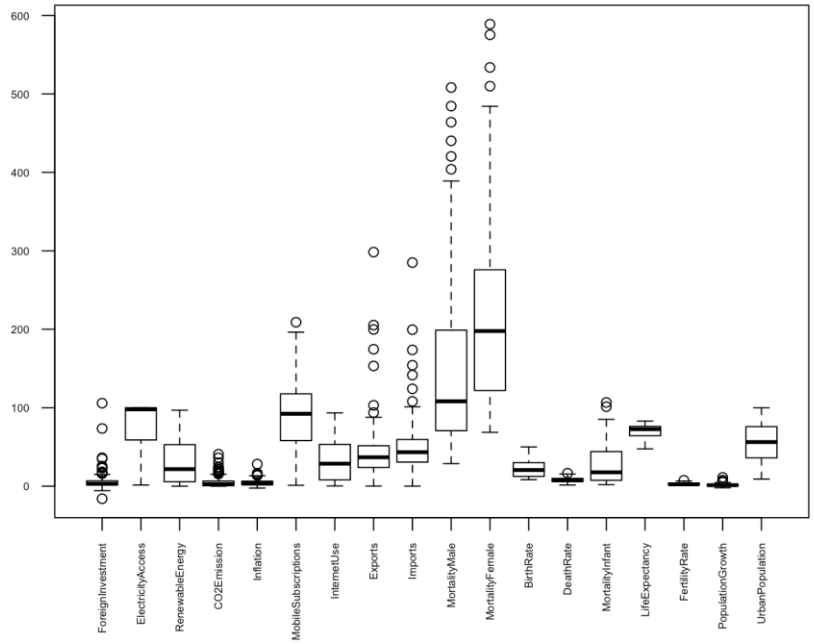


Figure 6: Remaining variables boxplot

Here we can see that, even when excluding GDP, the various measurement scales used to measure the variables differ widely. Why is this important? Later on, when clustering our data, we will want to find groups of countries that are similar. Similar in the sense of the above-mentioned characteristics, such as electricity access, inflation rates, and mortality rates. While this will be discussed at greater length in the model building section, for now it will be helpful to consider that for us to figure out how similar countries are we will need to introduce some notion of distance – i.e. how close (similar) or far (dissimilar) these countries are to each other. If we have one variable with an extreme variance, all our clusters would simply be based solely on this variable as the other variables, comparatively, provide very little variance. Based on the above finding, it is necessary for us to scale our data before model building. This essentially means subtracting the mean and dividing by the standard deviation for each respective variable. Here, scaling ensures our variables are all represented on a comparable scale suitable for our clustering at a later stage. See Figure 7 below for the boxplot of how our variables look after we have scaled our data.

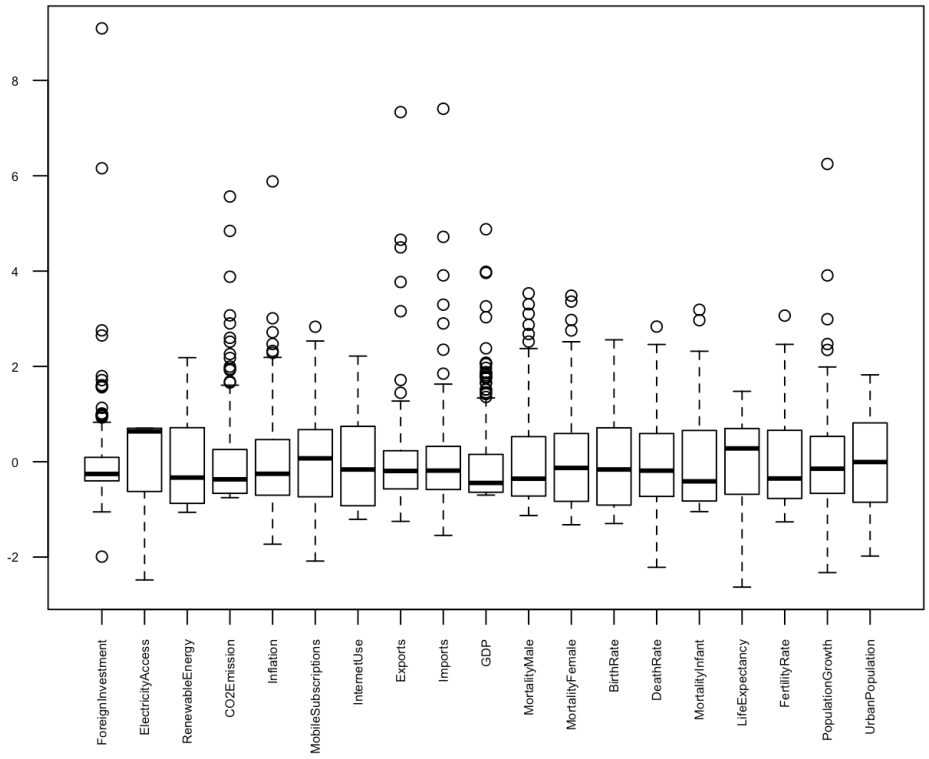


Figure 7: Boxplot after scaling

In any data science task, this step of checking the different scales on which your variables are measured is very important. This is a very subtle but important point, which we need to take into consideration in order to present statistically valid results.

Correlations

Now we can briefly look into the correlations of the various variables. It is often very interesting to understand whether – and which of – your variables are correlated.

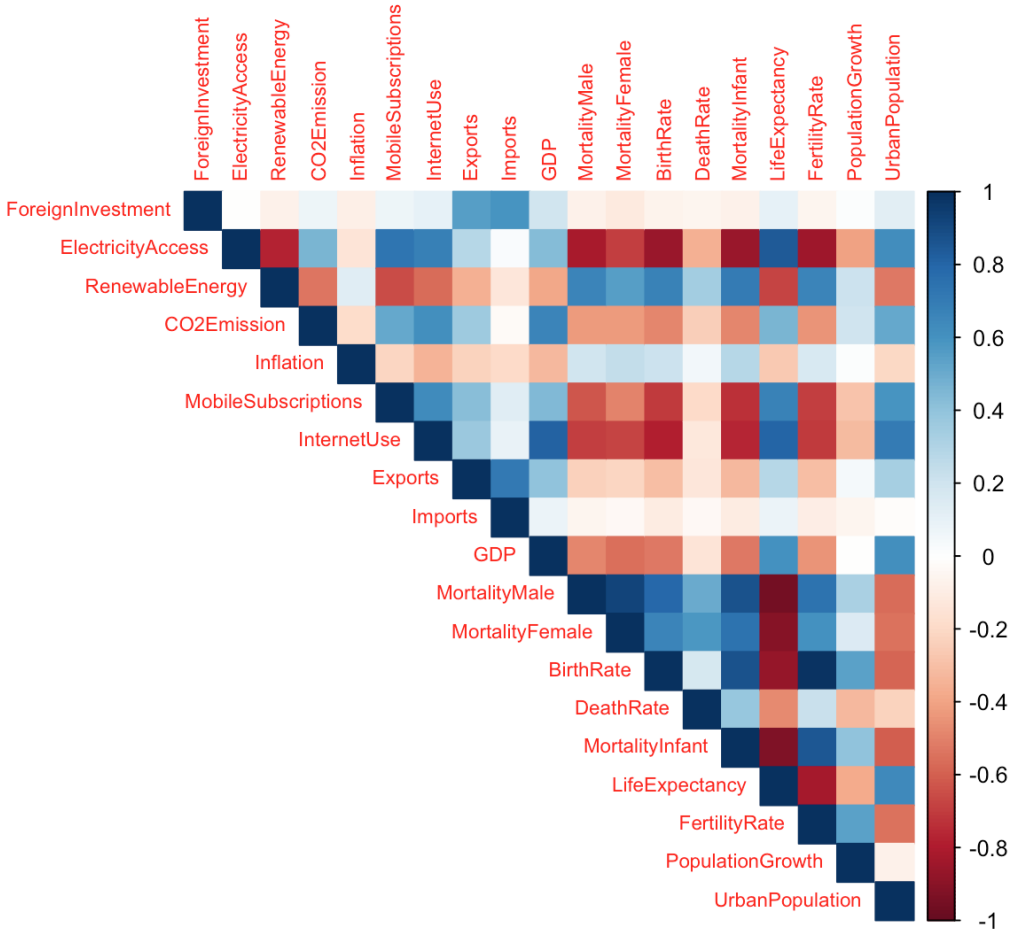


Figure 8: Correlations

We can see, for example, that there is a strong negative correlation between electricity access and the infant mortality rate. This makes sense as we would expect increased electricity access to substantially reduce the infant mortality rate, since, for instance, medical apparatus powered by electricity can be used in the prevention of infant mortality.

Besides helping us identify interesting trends in the data, as seen above, correlation analysis will also help us identify possible problems present in the dataset. **Multicollinearity** – i.e. the presence of highly correlated variables – is a term used to describe a problem that affects the validity of a statistical method, such as linear regression.

Depending on the model you are building, correlations may be of interest to you for different reasons. Since we are going to be building a clustering model, the correlation of variables is not of utmost importance beyond the interesting trends it highlights for us. We will explain a clustering model soon, to give you more context, but, to give you a basic idea, clustering models seek to group observations – in our case, countries – that have similar characteristics.

Missing values

Let's now check for missing values, or **nulls**, in our data set. Missing values are common in many data sets, and it is important to assess the frequency of these nulls and how you are going to deal with them early on in the project. It is important to deal with null values since, if left undealt-with, they may be used as simple zero values in your model, leading to very misleading results. Take a look at Figure 9 below, where we check exactly how many missing values are present.

Table 2: Nulls in data set

Indicator	# of nulls
FP.CPI.TOTL.ZG	10
NE.EXP.GNFS.ZS	7
NE.IMP.GNFS.ZS	7
SP.DYN.IMRT.IN	7
BX.KLT.DINV.WD.GD.ZS	6
EN.ATM.CO2E.PC	4
SP.DYN.AMRT.FE	4
SP.DYN.AMRT.MA	4
SP.DYN.LE00.IN	3
SP.DYN.TFRT.IN	3
IT.CEL.SETS.P2	2
IT.NET.USER.ZS	2
EG.FEC.RNEW.ZS	1
NY.GDP.PCAP.CD	1
SP.DYN.CBRT.IN	1
SP.DYN.CDRT.IN	1
EG.ELC.ACCS.ZS	0
SP.POP.GROW	0
SP.URB.TOTL.IN.ZS	0
Totals	**63**

Region	# of nulls
St. Kitts and Nevis	6
Palau	6
Guam	5
Virgin Islands (U.S.)	5
Bermuda	4
Dominica	4
Puerto Rico	4
South Sudan	4
Syrian Arab Republic	4
Cuba	2
Djibouti	2
Ethiopia	2
Guinea	2
Jordan	2
Papua New Guinea	2
Sao Tome and Principe	2
Aruba	1
Eritrea	1
Hong Kong SAR, China	1
Kiribati	1
Macau SAR, China	1
Turkmenistan	1
Uzbekistan	1
Totals	**63**

Figure 9: Nulls in dataset

One strategy is to not include any observations which have null entries. The issue with this is that we would be excluding many countries that may only have one or two missing values. We could decide to exclude countries that are missing four or more values.

Another strategy would be to use the second sheet of data in our .xlsx file, which contains certain country groups, and use the average for the group to which the country belongs. We could write our own algorithm to do this. Consider furthermore that countries may be part of multiple groups – i.e. one country may be part of the Low-income group *and* the Sub-Saharan Africa group. At this point it is up to the discretion of the data scientist whether they average those two values or

perform another operation. Another strategy is to use various kinds of imputation. These can easily be done with the Caret package in R.

We are going to use bagging for the missing values. Imputation via bagging fits a bagged tree model for each variable (as a function of all the others). This method is simple, accurate, and accepts missing values, but it has a fairly high computational cost. However, because our data set is small this is not an issue.

We are now left with a complete data set and are ready to build our model. Remember: whatever the project, always make sure to state all your assumptions. For example, in this case we would have explained that, firstly, there were 63 missing values in total and, secondly, we used imputation via bagging in order to fill in these null values. Doing this allows a stakeholder in our project to understand the complete context of what has been done.

Summary

Let's recap all we have done in this module.

First, we have managed to import our dataset into the R environment. We noted that data may come in different forms (from .xlsx to .csv, amongst others), and that different packages may be needed in order to import different data sets.

Once we imported our data, we began by noting the size of our dataset (192 observations, 19 variables). We then examined how the variables were measured on different scales and how this could obviously affect our analysis and model building at a later stage. We decided that we should scale our data before building our model in the next section.

We then briefly explored the correlation matrix of the variables to unearth some basic trends present in the data. So far, all of the above has helped us to get acquainted with the dataset and help build some familiarity with the task at hand.

Finally, we dealt with missing values in the dataset. We noted that missing data is a frequent occurrence. One should always check for, and decide on how to deal with, missing data values. In our case, we decided to use imputation via bagging to fill in the 63 null values in our dataset.

All of this will help us when, in the next section, we begin building our clustering model for our data.

Unit 3: Basic Model Building

Now it's time to build a model. Remember our goal: to build a model that allows us to discover meaningful structure within our country dataset. Accordingly, we be using an unsupervised model. For more details on the subtle nuances of choosing what models to use, visit this great resource.

Some unsupervised models we could use include clustering models, such as k-means or hierarchical, or self-organizing maps. In this module, we are just going to focus on building a k-means clustering model in order to try and discover possible groups of similar countries.

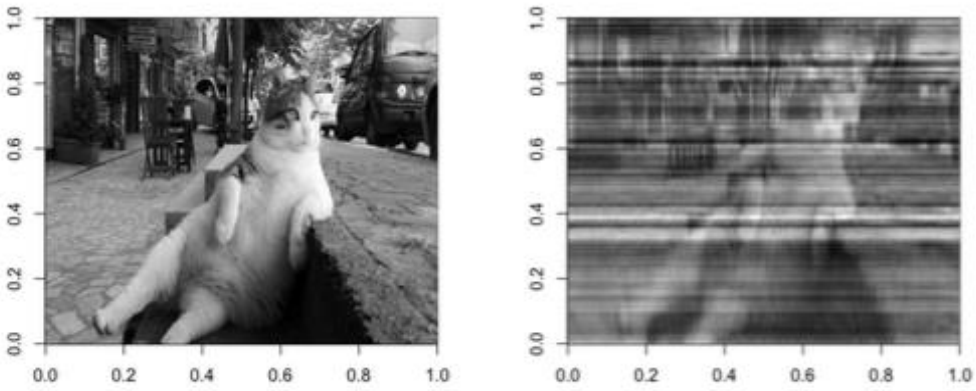
It is always beneficial to know and understand many different types of statistical approaches (from simple linear regression to deep neural nets) when engaging in a data science project. That way, you can develop robust intuitions regarding which model may work best for the problem at hand. This will also give you the advantage of being able to implement multiple different models and use the best performing model, or to create an ensemble that utilizes several models.

Scaling

As noted previously, when we measure the distance between any two data points, this measure can be greatly affected by the different scales with which our variables are measured. As our data points are countries, we are going to standardize our data accordingly before we continue, using the standard scale function.

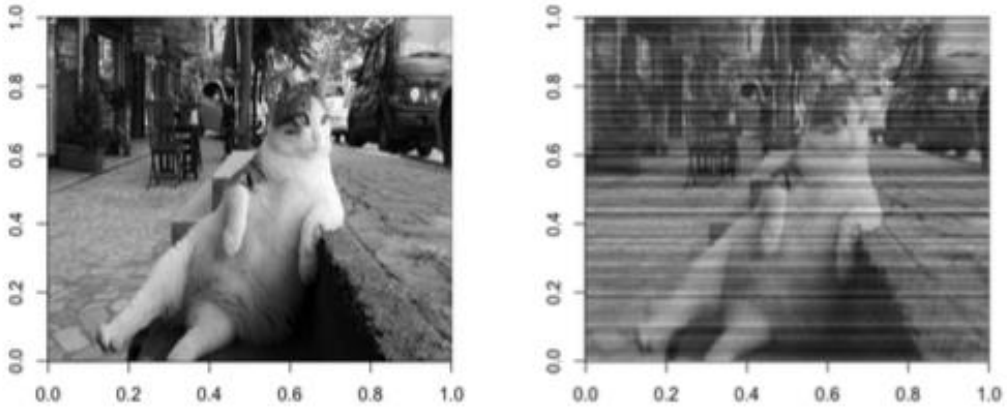
Principal components analysis (PCA)

What is principal components analysis (PCA) and why are we introducing it here? **PCA** is a dimensionality reduction technique that allows you to shrink your number of variables while still capturing as much of the information, or variance, as possible. We will not go into the details here, but we will use some cat photos to explain the concept a little further. Take a look at the cat photos below. Each is constructed of just a small subset of principal components, as opposed to all the available pixels.



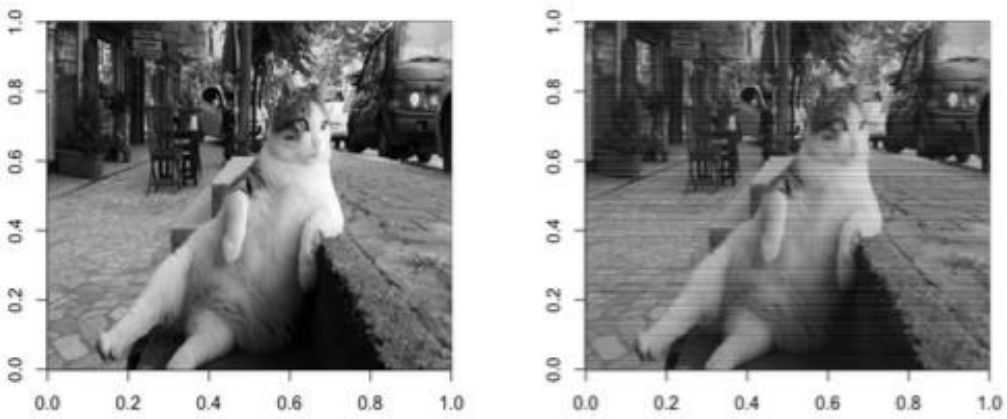
Original picture on the left. Picture on the right is a reconstruction with 20 out of 420 components.

Figure 10: Reconstructed cat photo (20/420)



Original picture on the left. Picture on the right is a reconstruction with 50 out of 420 components.

Figure 11: Reconstructed cat photo (50/420)



Original picture on the left. Picture on the right is a reconstruction with 150 out of 420 components.

Figure 12: Reconstructed cat photo (150/420)

Figures 10 through 12 illustrate how PCA allows you to capture as much information as possible, in this case a picture of a cat, with a reduced number of variables (termed **principal components**).

The reason we are interested in PCA is that it is a handy visual technique. We want to eventually plot our data points and visualize our clusters in a two-dimensional plot; a 19-dimensional plot just does not exist. Before we continue to build our clustering model and plot our results, let's demonstrate the power of PCA with a simple biplot.

In general, a **biplot** is two-dimensional scatter plot where the data points are plotted as points and the variables are displayed as vectors. A biplot is very effective to use for data exploration. If you are wondering which two variables we are going to use for the biplot (since we have 19), then you are asking the right question. Thanks to principal components analysis, we are going to use the first and second principal component as each respective axis. Doing so produced the biplot below. At a later stage, we will use these same two principal components as the axes for plotting our clustering results.

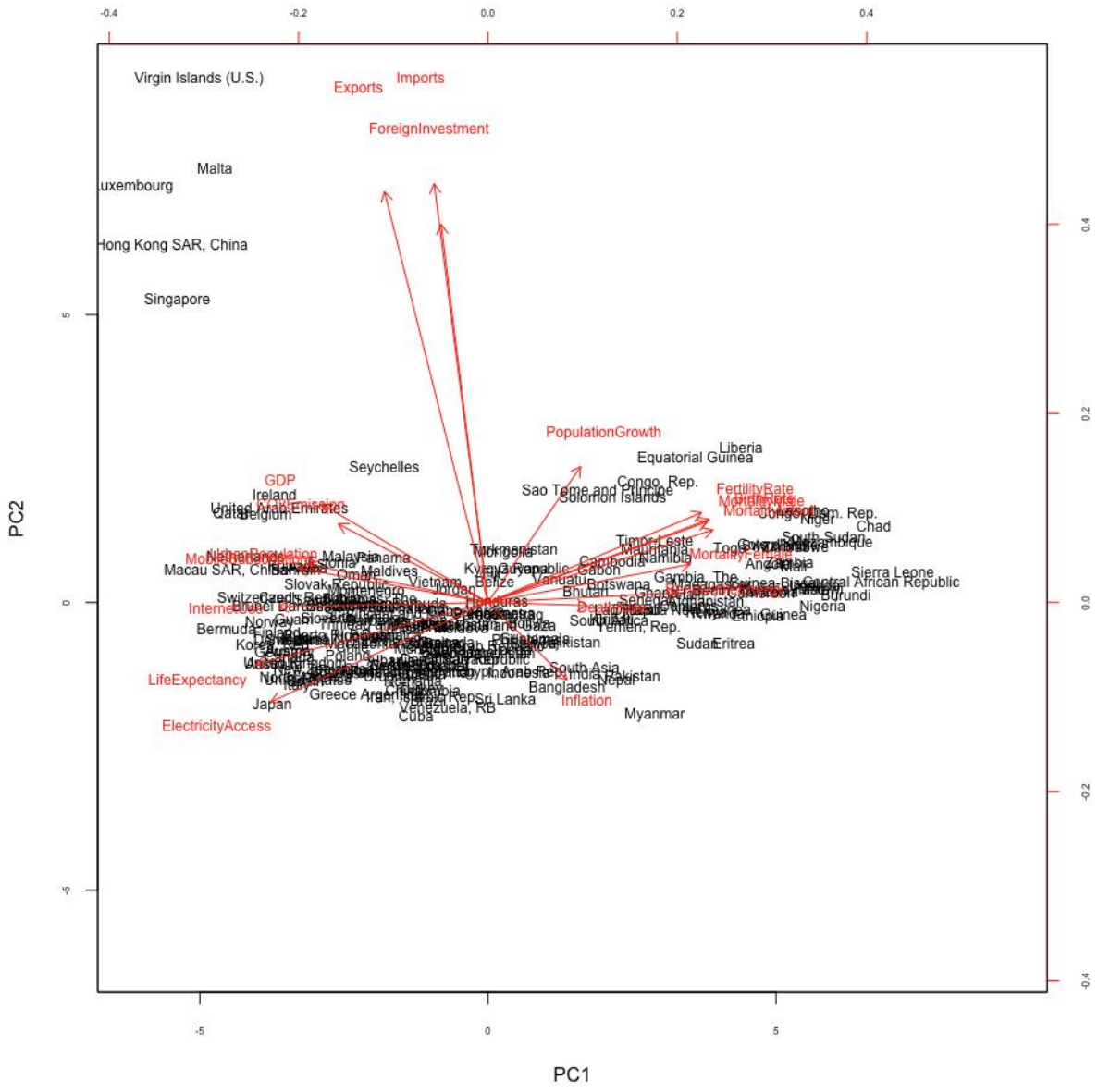


Figure 13: Two-dimensional biplot

There are at least two questions we should consider at this stage:

1. How much of the information or variance do these first two principal components capture? (This is a bit like asking how much of the cat picture can we see with the first two principal components.)
2. Which of our 19 variables are contributing toward first and second principal components (PC1 and PC2)?

The first question requires a bit more knowledge about PCA. Have a look at the code written in the R script below. You can see from the table below that approximately 63% of the variance or information is captured in PC1 and PC2, which we are using for our plot. If you are interested in exploring this in more detail, then you can search online.

```
# Creating a datatable to store and plot the
# No of Principal Components vs Cumulative Variance Explained
vexplained <- as.data.frame(pca.out$sdev^2/sum(pca.out$sdev^2))
vexplained <- cbind(c(1:19),vexplained,cumsum(vexplained[,1]))
colnames(vexplained) <- c("No_of_Principal_Components","Individual_Variance_Explained",
                          "Cumulative_Variance_Explained")

# Table showing the amount of variance explained by the principle components
vexplained
```

No_of_Principal_Components	Individual_Variance_Explained	Cumulative_Variance_Explained
1	0.5108106883	0.5108107
2	0.1179946170	0.6288053
3	0.0849015653	0.7137069
4	0.0693497415	0.7830566
5	0.0483412027	0.8313978
6	0.0375646727	0.8689625
7	0.0252211769	0.8941837
8	0.0234089805	0.9175926
9	0.0188722005	0.9364648
10	0.0151439631	0.9516088
11	0.0116159464	0.9632248
12	0.0095145048	0.9727393
13	0.0076479097	0.9803872
14	0.0066615622	0.9870487
15	0.0056406965	0.9926894
16	0.0047114054	0.9974008
17	0.0016225440	0.9990234
18	0.0006398665	0.9996632
19	0.0003367559	1.0000000

Figure 14: Variance percentages

Our answer to the next question is important as it is necessary for interpretation in our analysis. We will use a package called “factoextra” throughout this module. The package allows us to easily visualize the contributions of our variables to these two dimensions. See the visuals below, which show the various variable contributions to PC1 and PC2.

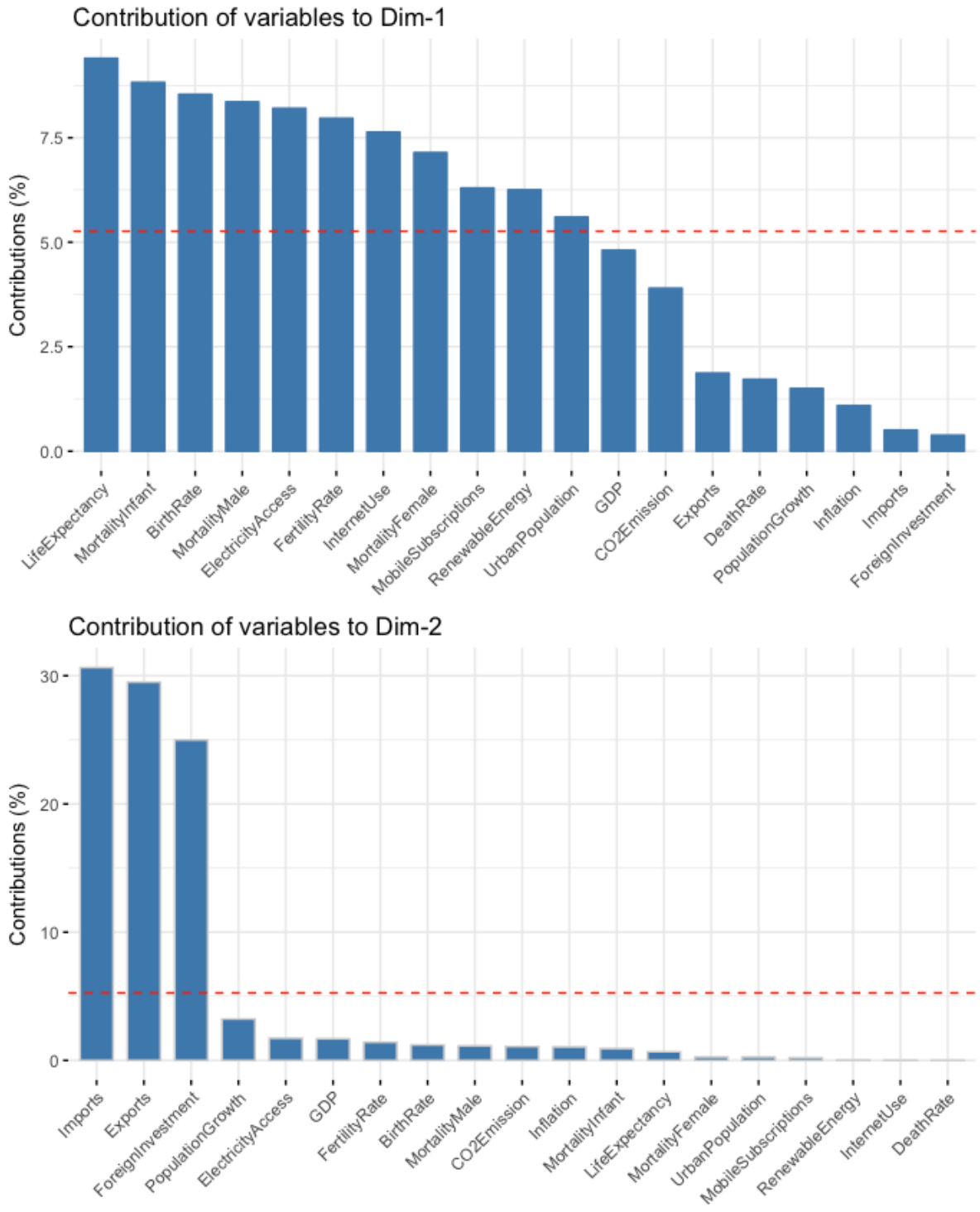


Figure 15: Contributions to PCs

Now that we have a good understanding of PC1 and PC2 (the dimensions which we will use in our cluster plots), let's get to building our actual clustering model.

Clustering model (K-means)

Measuring distance

Distance measurement is a key part of our clustering analysis. Since the distance between any two data points – say, Sweden and South Africa – can be understood as representing the similarity of these two countries, the choice of how we measure "distance" between these two data points will consequently affect our clustering model. There are both distance- and correlation-based distance metrics.

Classic distance-based measurements include:

- Euclidean distance
- Manhattan distance.

Correlation distance-based measurements include:

- Spearman
- Pearson.

Depending on the data science project and the nature of the data, different distance measurements may be utilized. **Euclidean distance** is how we intuitively think of measuring the distance between two objects – i.e. drawing a straight line between two points and measuring the distance of that line. In contrast, **correlation-based distance** measurements instead consider two objects to have a small distance between each other if there is a high correlation between their respective values, even if their values are far apart in terms of Euclidean distance.

While there are further intricacies concerning distance measurement that will not be covered here, it is important to note that different types exist, and each data scientist should carefully consider what measurement best suits the project at hand. For the purposes of this project, we are going to simply use Euclidean distance. Here is a great resource to better understand distance measurements.

Can we cluster?

The next question we generally want to answer is whether it is appropriate to cluster our data. Although many different techniques exist, we are going to simply visually determine whether clusters are present by computing and displaying the dissimilarity matrix of our data – using Euclidean distance. The dissimilarity matrix helps us to visualize the distance between different observations (countries in our case), and hence determine how dissimilar observations may be. A

value of 0 indicates zero dissimilarity. Hence the diagonal line is red, as each country understandably has zero dissimilarity to itself.

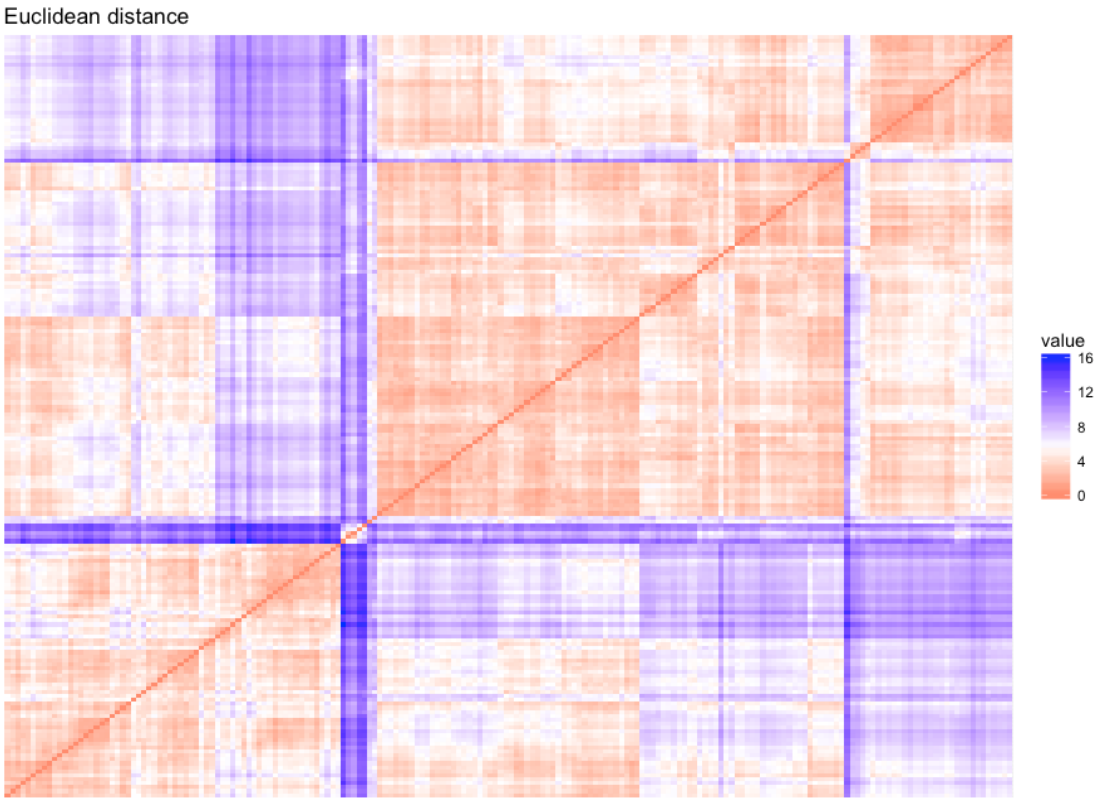


Figure 16: Clusters in Euclidean distance

This plot helps us to visually confirm that distinct clusters seem apparent in the dataset.

K-means clustering

K-means clustering is an algorithm that creates clusters such that the total within-cluster variation is minimized. The algorithm begins by selecting k different data points as the centers for the cluster. The remaining data points are then assigned to the various cluster centers to which they are closest. Iteratively, the algorithm computes the new mean for each cluster and then continues to update and assign data points to the different clusters until convergence is reached.

Do not let the description fool you: the algorithm is actually very simple. In the lecture video, there is a visual demonstration that will help you to easily understand exactly what is happening.

Choosing K

One drawback of the k-means clustering algorithm is that k, the number of clusters, has to be specified before the algorithm begins. Deciding the value of k is a non-trivial task that has no *de facto* solution. R has a sufficient package, NbClust, that will allow us to harness 30 different tests in order for us to try and ascertain the most appropriate number of clusters. See this additional resource for more clarity on choosing a suitable value for k.

Summary

We first noted that we have an unsupervised learning problem and decided that we wanted to build a clustering model. We then scaled our data to ensure that all variables were measured on a standard scale.

Thereafter we explored the idea of principal components analysis, which will be particularly useful for plotting our data points in two dimensions using the first and second principal component (PC1 and PC2) as the axes.

We noted how differing distance measurements can affect the analysis before proceeding to use Euclidean distance and a dissimilarity matrix plot to visually confirm that our data contained meaningful clusters. Finally, we examined the k-means clustering algorithm we are going to use, and we briefly talked about choosing the value of k.

In the next set of notes, we are going to finally take a look at what we have and start to think about how we are going to effectively draw insights and communicate these results to relevant stakeholders.[2]

---

[2] *An Introduction to Statistical Learning with applications in R* gives an excellent introduction to some of the available statistical methods. This free online resource can be found at https://www-bcf.usc.edu/~gareth/ISL/

Further reading:

https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/

Unit 4: Shiny

Introduction

It's time to start thinking about how we are going to effectively draw insights and visually communicate the results of our data science project. Remember that visualization is one of the most important parts of a project. Effectively visualizing your results will help your audience to better understand exactly what you have achieved during the project.

There are many principles to keep in mind for effective data visualization. We are going to revert back to the CryptoKitty Genome Mapping visual I created to walk you through some of the important visualization considerations to keep in mind while designing an effective visual.

Visualizing results

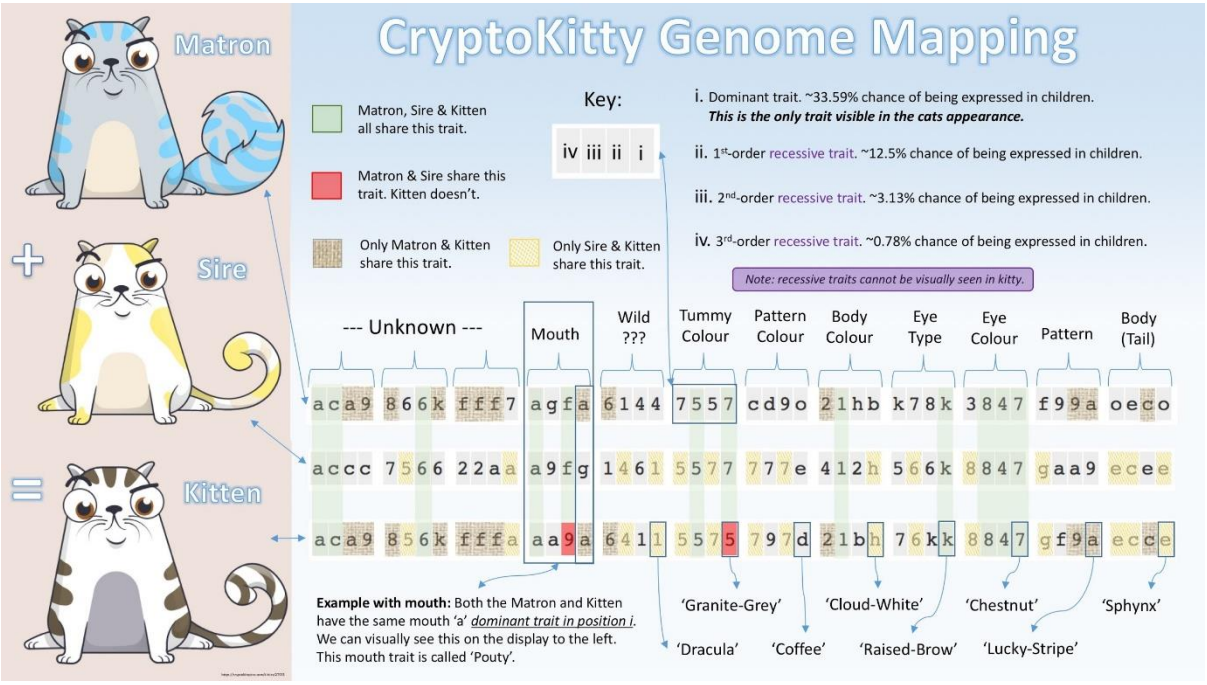


Figure 19: CryptoKitty Genome Mapping - Updated

Here is a list of the various elements that were considered in order to make the visual more effective:

- DNA bars of cats were brought much closer together. Originally each DNA bar was in line with the cat it corresponded to.

- The DNA bars being closer together helps users to more easily examine differences between the DNA of the respective cats.
- Plus (+) and equals (=) symbols were added to the cats on the left in order to signify that the first two cats combine to make the third.
- The parent kittens were named Matron and Sire so that they could be easily referred to in the key.
- Colors were used to help users easily distinguish different and shared traits between the cats in the DNA bars. The green color extends across all DNA bars to further help users.
- Texture was used to better highlight information in DNA bars.
- The three curly brackets indicating the unknown segment of the genome was added so that users would not mistakenly try to guess what is happening there.
- A gradient background was used to provide some contrast and to add more visual appeal.
- The cats on the left were given their own background color to further distinguish them.
- Thick line edges were utilized around the boxes in the key to improve readability.
- A box was put around a quadruplet of DNA pointing toward the key to allow the user to more easily understand the DNA.

The following all pertain to making it easier for the user to understand how the DNA corresponds to the kitty visual:

- Bold and italic text was used in the key to emphasize that only the trait in position i can be visually seen.
- Awkward-looking and eye-grabbing purple text color and purple text box were used to again state that recessive traits cannot be visualized.
- A detailed example referring to the kitty's mouth was incorporated in order to demonstrate how the DNA visually determines the mouth.
- A box outline was used around the mouth DNA segment in order for the user to more easily understand the above point.

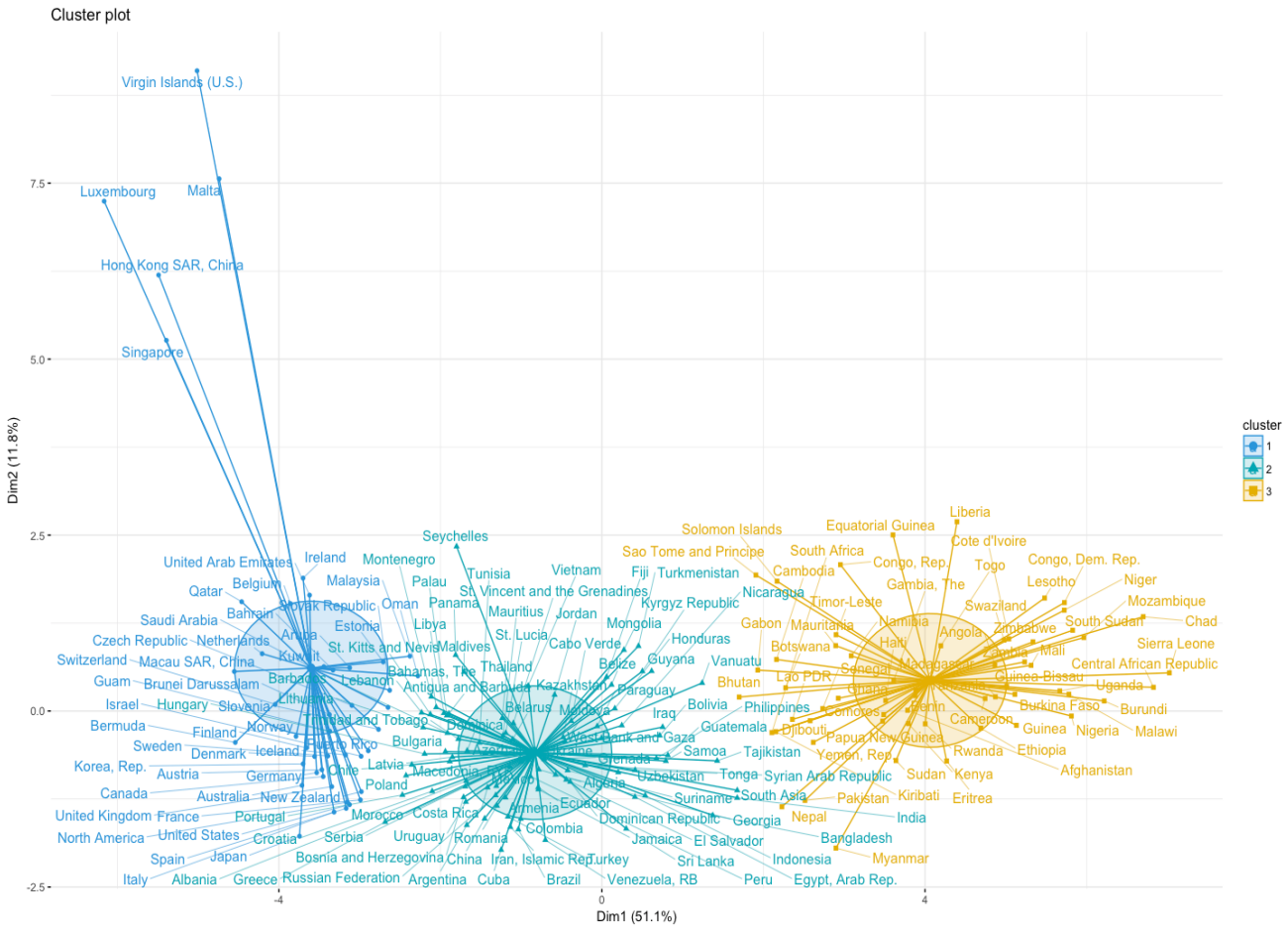
As you can see, there are many considerations that go into making a visual effective for the end user. One of the best strategies is iteration in your design. Once you have created a basic visual for your model, ask for feedback from friends or colleagues; their opinions can act as a guide to help you improve your visuals. With this in mind, let's look at some of the visuals we can employ for our ongoing data science project.

Below are a few key visuals: the respective variable averages for each cluster, the cluster plot, and the clusters plotted on a world map.

cluster	ForeignInvestment	ElectricityAccess	RenewableEnergy	CO2Emission	Inflation	MobileSubscriptions	InternetUse	Exports	Imports
1	1	9.556497	99.83657	11.90415	12.7512100	2.011545	122.92465	70.621026	68.49036	59.82697
2	2	4.735364	94.38806	20.28216	3.9744206	4.996811	101.46718	31.529656	38.64856	45.87540
3	3	4.978734	36.70759	65.39138	0.6922908	5.880514	45.64165	5.947933	30.86129	45.58313

	GDP	MortalityMale	MortalityFemale	BirthRate	DeathRate	MortalityInfant	LifeExpectancy	FertilityRate	PopulationGrowth	UrbanPopulation
1	41218.311	55.5217	100.6637	12.95984	6.961805	5.442285	79.29608	1.834613	1.4065335	82.32484
2	6775.059	102.7179	191.4813	18.30169	7.483860	17.475281	72.93427	2.272341	0.9343943	56.40761
3	1718.404	273.9878	325.6874	35.68736	9.870879	58.931034	59.28474	4.746259	2.4932587	36.30353

Figure 20: Respective variable averages for each cluster



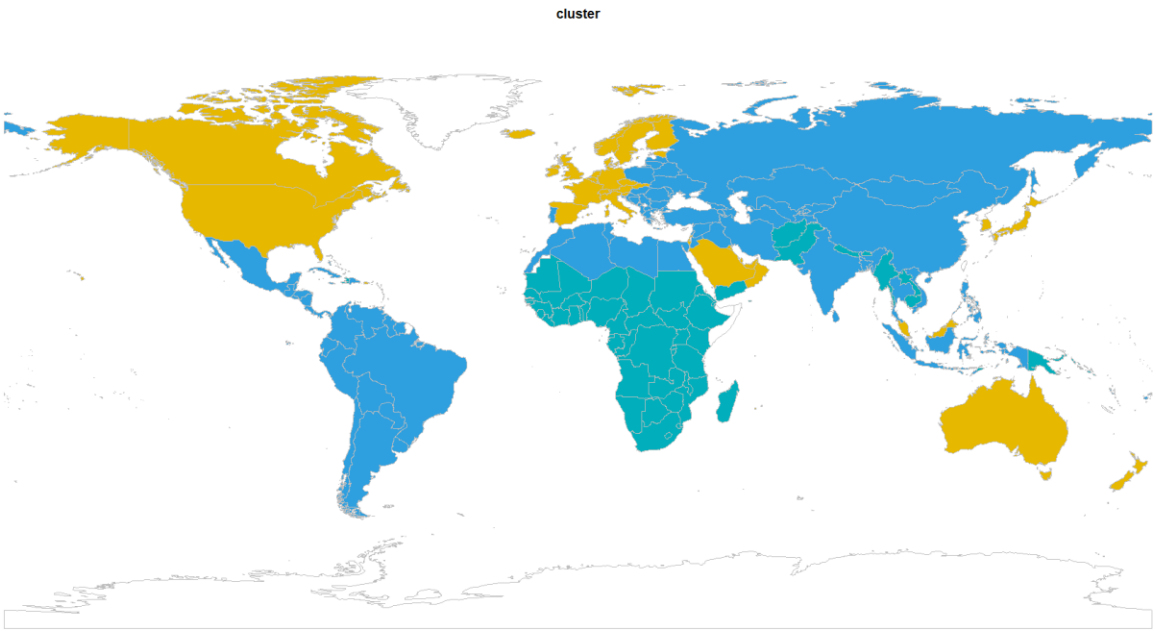


Figure 21: Clusters shown on world map

These visuals are a great starting point. Along with these visuals, we would likely want to highlight some of the key findings of our project – examples could include:

- There seem to be three distinct clusters: upper developed (North America, Europe, and Australasia), middle developed (South America and Asia), lower developed (Africa).
- The average infant mortality rate is 10x higher in the lower developed cluster compared to the upper developed cluster.

We could simply present our results in a static report with these visuals, but we are going to take it a step further. We are going to build and deploy an interactive webpage that can be used to effectively showcase our results.

As an aside, what we have just done looks quite similar to the process of how we categorize first-, second-, and third-world countries. What this shows is that the skills and competencies being learned now can facilitate the tackling of tasks of significant magnitude.

Why Shiny

The goal of this module is to learn the Shiny framework to help you effectively communicate your results throughout your organization. What, then, is Shiny? Shiny is an R package that allows you to build interactive web applications straight from R. Let's consider why this is so helpful.

Imagine you are in the process of building your model, and you want to share some of your progress with your colleagues. Or maybe you simply want to share your final results. The reality is, many people in your organization probably won't have R installed on their computer, nor will they know how to run the scripts you want to send to them.

With Shiny, you can easily build an interactive web application that essentially houses your code. If you then send the URL to your stakeholder, they can interact with it simply and easily. Another reason is that it makes it much easier to explore and visualize the data, particularly if the app is well designed. This is because you could build in functionality that would allow you to filter certain datasets, allowing your user to easily contrast information.

Take a look at the screenshot below of a Shiny app built for this data science project.

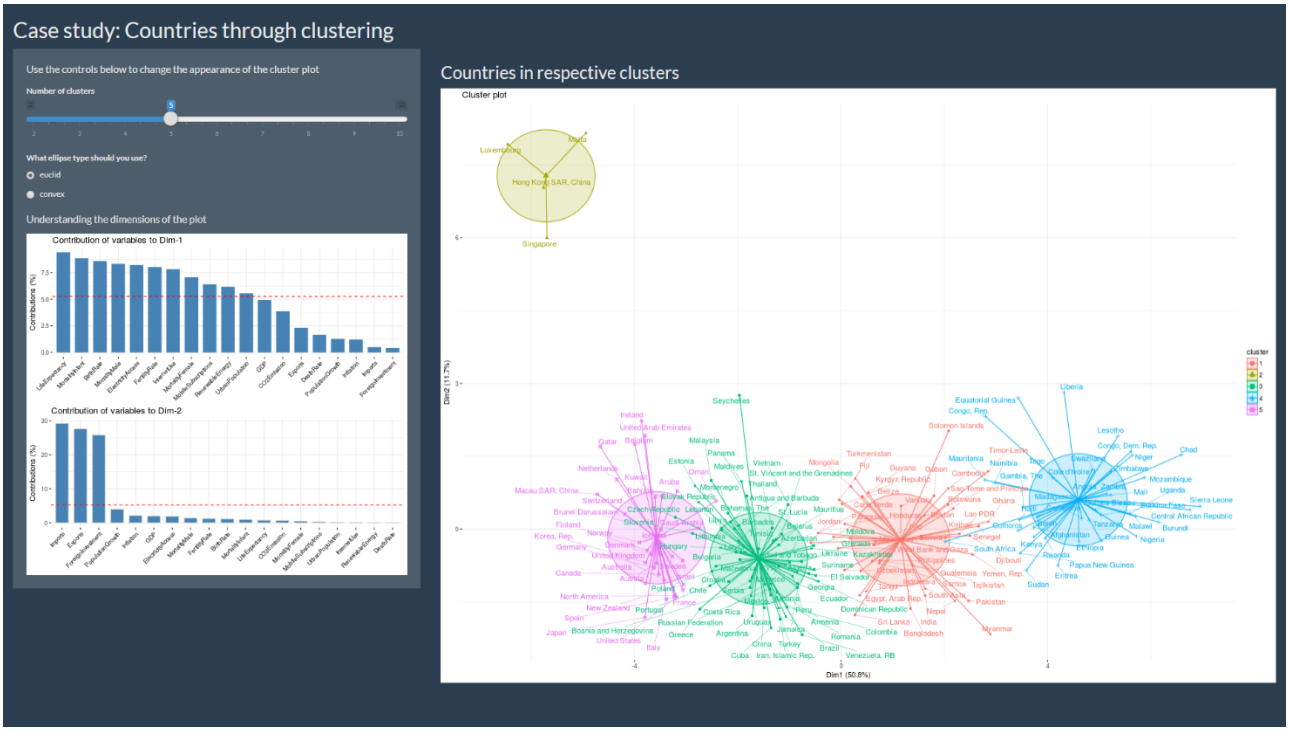


Figure 22: Screenshot of basic Shiny App with five clusters

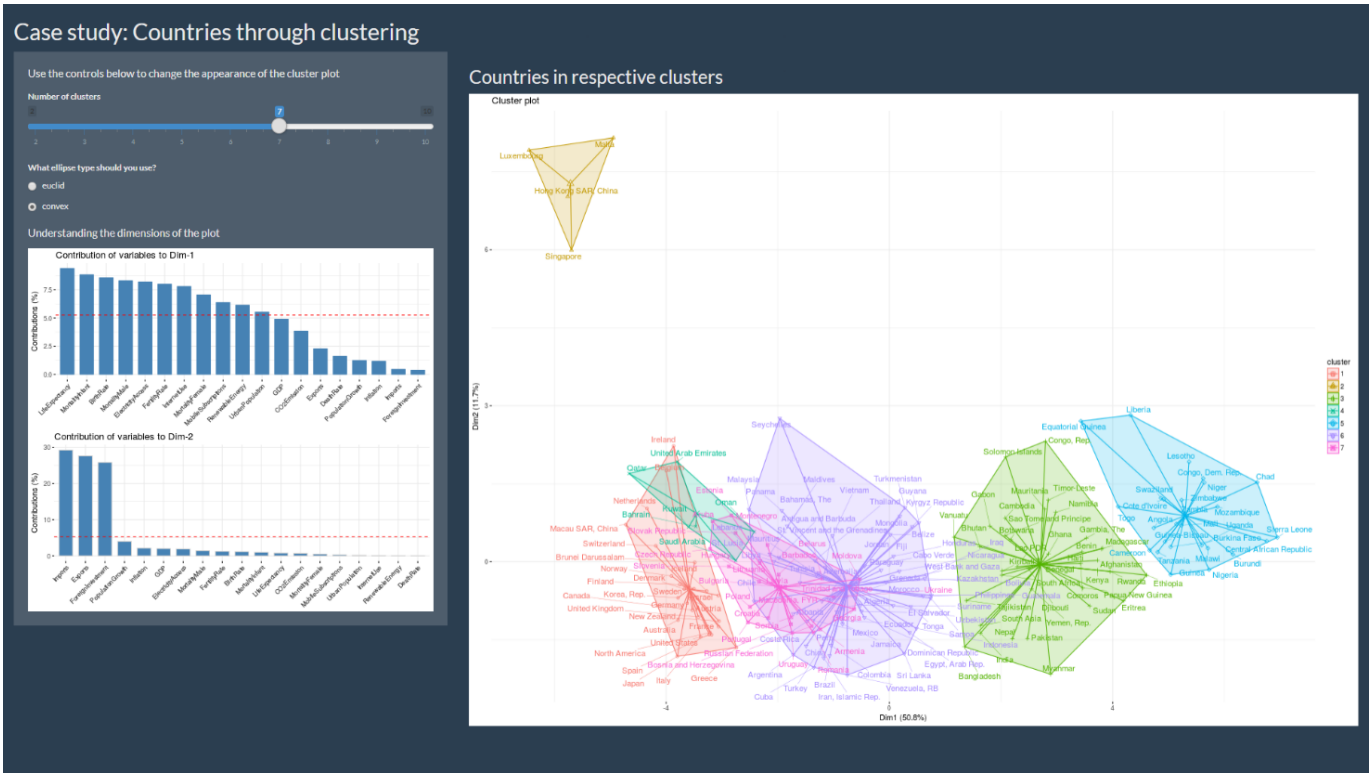


Figure 23: Screenshot of alternate view of Shiny App with seven clusters

In the built application, we have a slider that allows the user to change the number of clusters. This then renders a new cluster visualization. It also allows you to choose the ellipse type you would like to use for building your cluster. In the lecture we will go through some basics to get you and your first Shiny App underway.

Summary

We started off emphasizing the importance of data visualization; in order to successfully showcase your data science project, it is critical to present it in a visually effective manner.

We then explored an example of a CryptoKitty genome mapping visual. We considered many of the design justifications to showcase the numerous thoughts that go into making an effective visualization.

Thereafter we explored possible visuals presented by our data science project. We motivated that not all stakeholders will be able to run R scripts sent to them, making building an interactive web application presenting your data science project results or progress a very effective means of communication. Furthermore, we realized that interactive data visualizations can be more helpful

to understand and explore the data. Accordingly, we introduced Shiny, a package for R that allows us to do the above.

Bibliography

Analytics Vidhya (2016). Practical Guide to Principal Component Analysis (PCA) in R & Python. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/

Beautiful, I. (2019). Information is Beautiful. [online] Information is Beautiful. Available at: https://informationisbeautiful.net/

Clark, J. (2018). CryptoKitties Genome Mapping. [online] Medium. Available at: https://medium.com/newtown-partners/cryptokitties-genome-mapping-6412136c0ae4

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). An introduction to statistical learning. 1st ed. New York: Springer.

R-bloggers. (2019). R news and tutorials contributed by (750) R bloggers. [online] Available at: https://www.r-bloggers.com/ [Accessed 13 Feb. 2019].

RStudio (2017). Learn Shiny. [online] Shiny.rstudio.com. Available at: https://shiny.rstudio.com/tutorial/

Munzner, T. (2016). Visualization Analysis & Design. [online] Cs.ubc.ca. Available at: https://www.cs.ubc.ca/~tmm/talks/minicourse14/vad16infoplus-4x4.pdf

Collaborative Review Task

In this module, you are required to complete a collaborative review task, which is designed to test your ability to apply and analyze the knowledge you have learned during the week.

Brief

Time to finally put everything together.

Your task is to conduct a clustering analysis on the *CountryData.xlsx* file. The end goal is to build an interactive web application with Shiny and deploy this online using ShinyApps. Anyone will then be able to simply access your interactive data project at the click of a URL.

It is up to you to decide how you can best represent the results of your project. Approach this project as if the United Nations had just given you all the data and then asked for your best effort in presenting a summarized version of the results of your analysis on the countries reflected in the data.

To get an idea as to some possibilities, you can refer to the screenshots in the fourth section of notes.

Be sure to include the following in your Collaborative Review Task:

- Relevant information on any assumptions you may have made.
- The most prominent insights from the analysis.

Once you have completed your Shiny app, visit http://www.shinyapps.io/

Here you can sign up and deploy your app for free by following the simple steps. A unique URL with to your project will be produced, which will contain your interactive web application. Be sure to submit this URL.

For guidance and reference, you can refer to the R script used throughout the tutorial to conduct the analysis and clean the data.

Hint: Many of the functions used in the R script have certain parameters you can change if you are looking to do things slightly differently.