Configuration Manual

MSc Research Project
Financial Technology

Tatyana Gricenko
Student ID: X19233027

School of Computing
National College of Ireland

Supervisor: Noel Cosgrave

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Tatyana Gricenko

Student ID: X19233027

Programme: Financial Technology **Year:** 2020

Module: MSc Research Project

Lecturer: Noel Cosgrave

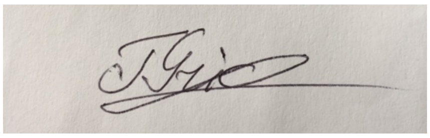
Submission Due Date: 28/09/2020

Project Title: Blockchain Technology: edit or not?

Word Count: **5085** **Page Count: 29**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:



Date: 28/09/2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	v
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	v
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	v

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Tatyana Gricenko
Student ID: X19233027

This configuration manual describes hardware, software specifications and programming steps for the research project “Blockchain technology: edit or not?”

1 Hardware

Processor: 2.8 GHz Intel Core i7
Memory: 8 GB 1067 MHz DDR3
Graphics: ATI Radeon HD 4850 512 MB
Storage: 1 TB SATA Disk

2 Software

macOS High Sierra version 10.13.6
Microsoft Excel for Mac version 16.39
Google Forms [1]
R version 4.0.2 [2]
R Studio version 1.3.1056 [3]

3 Project development

The implementation of this work is done using R programming. R and R Studio were installed. The dataset was downloaded from Google Forms in csv format. Firstly, the answer to question 6 was analysed. Followed by the analysis to answers to questions 16,1,3 and 4. Finally, the answers to questions 16,17,18,19,20. Answers to the web survey downloaded and stored in Excel in csv format. This dataset used for the analysis in R Studio.

---

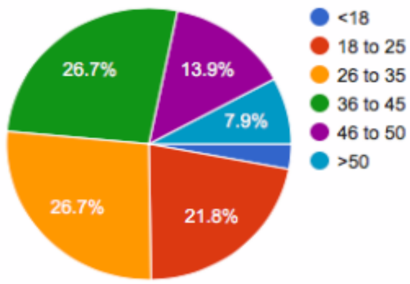
1 https://docs.google.com/forms/d/1hSqx5nyCkFbdZL2VEZrOzi92Rkjw3czbImJp5oU0uhU/edit

2 http://www.r-project.org/

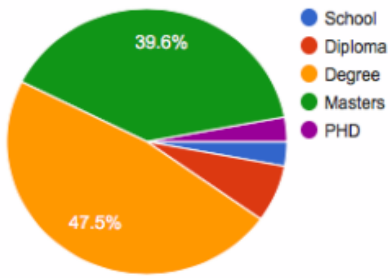
3 http://www.rstudio.com/

4 Answers to web survey

Q1, Figure 5. Age :

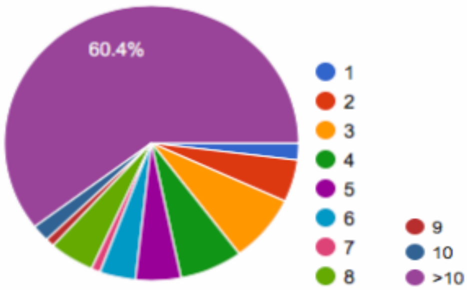


Q2, Figure 6. Education :

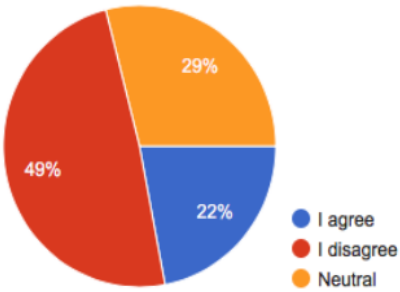


Q 3, Figure 7. Industry :

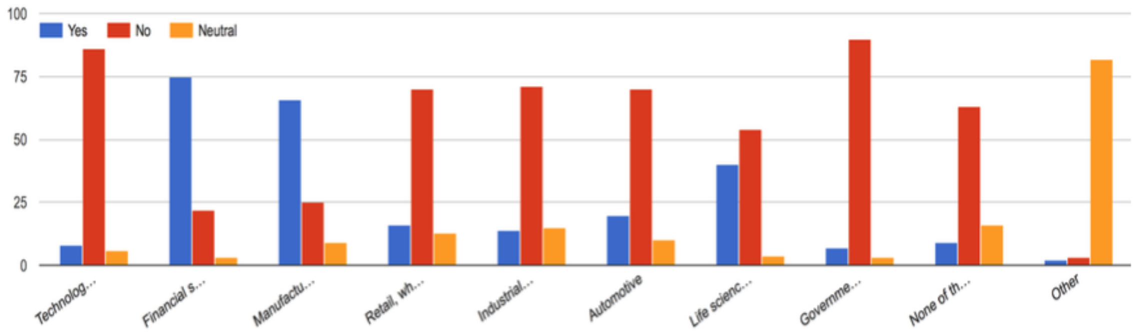
Q 4, Figure 8. Years in profession :



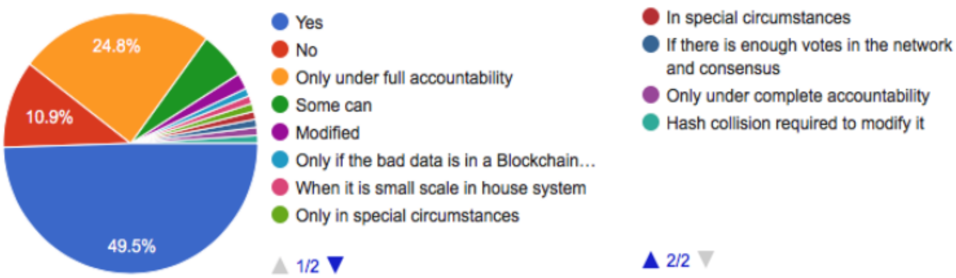
Q 5, Figure 9. If we allow someone to change the records, trust will wear down. The change for the kindest motives, like human error, opens the door for to the loss of trust :



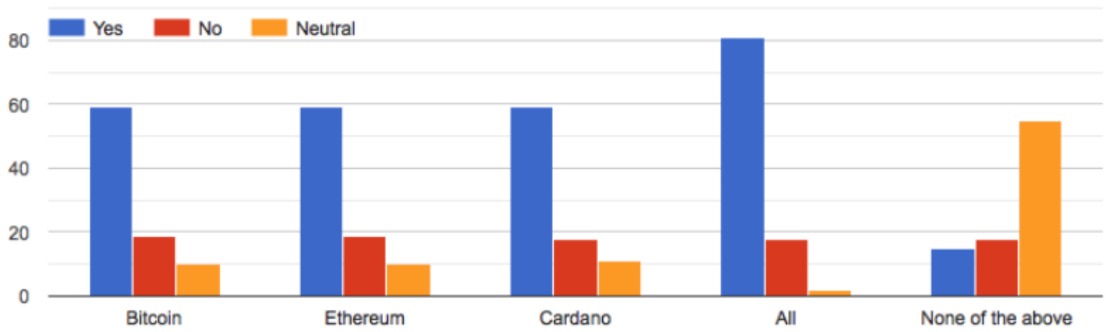
Q 6, Figure 10. Redactable blockchain would suit in the following industries :



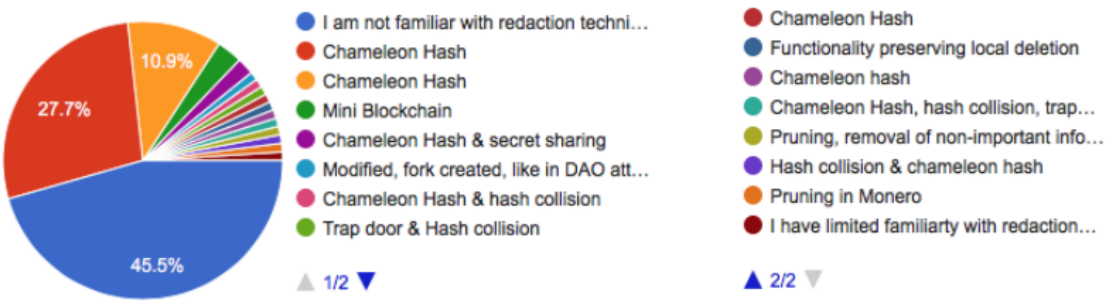
Q 7, Figure 11. Do you think blockchains can be redacted :



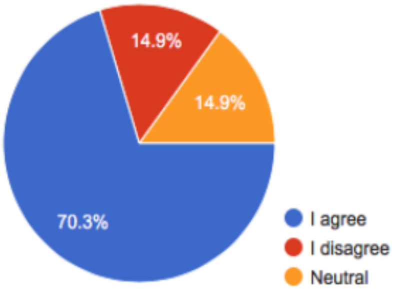
Q 8, Figure 12. Which type of blockchain do you think can be redacted :



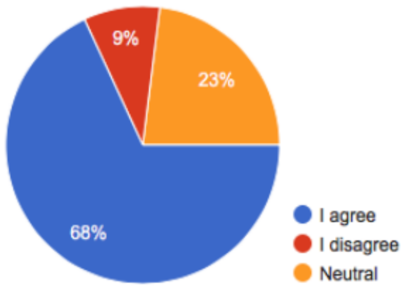
Q 9, Figure 13. Methods of redaction are you familiar with :



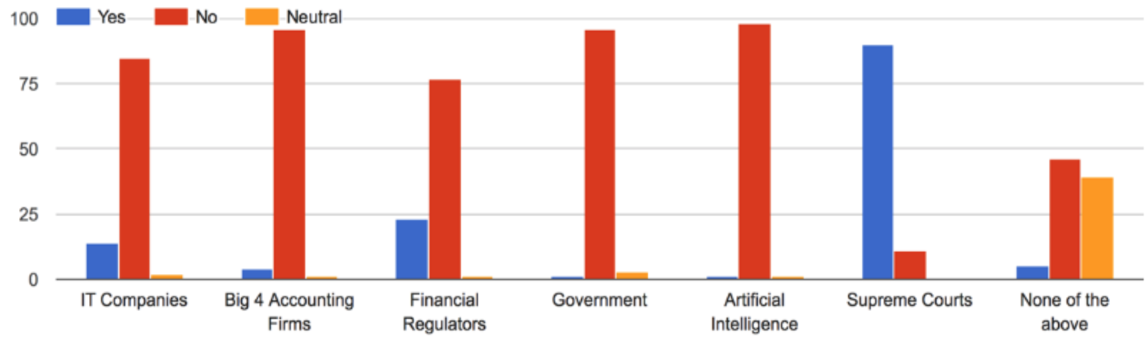
Q 10, Figure 14. Total immutability of the blockchain presently acts as a barrier for their wider adoption :



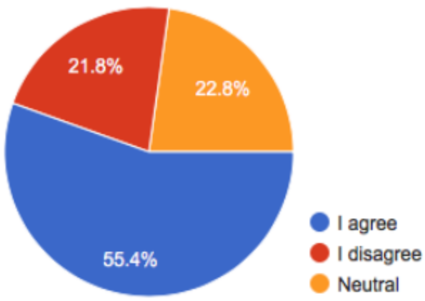
Q 11, Figure 15. Redactable blockchain needs comprehensive and strong governance model :



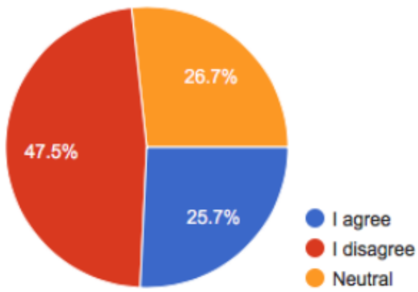
Q 12, Figure 16. In cases like "input errors, bugs in the code, fraudulent/illegal transactions, outdated terms in the smart contract." Do you agree that all parties sharing a blockchain need to comply with certain legislation? If so, do you agree that a representative from the list below should be allowed to edit the blockchain in certain circumstances if a court order exists on a case by case basis :



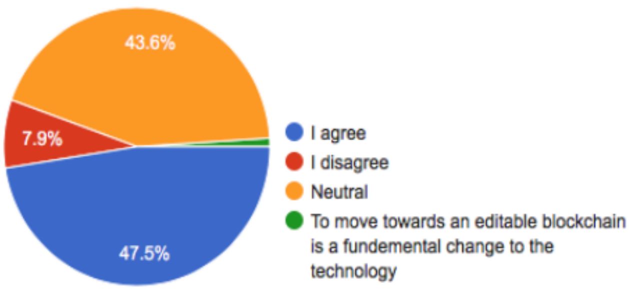
Q 13, Figure 17. There are more risks with redactable blockchains :



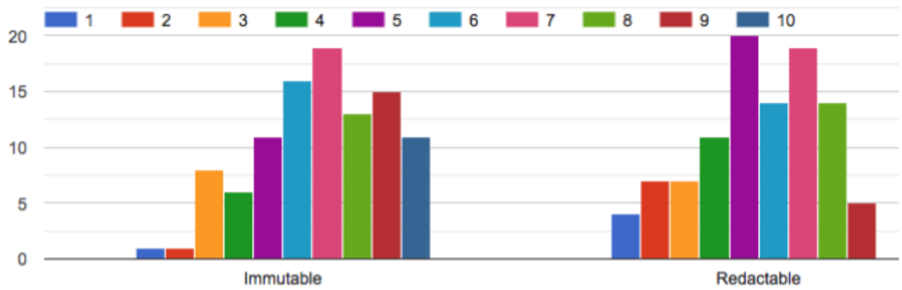
Q 14, Figure 18. Immutability is a core feature and strength of blockchain technology, stops fraudsters from altering the record :



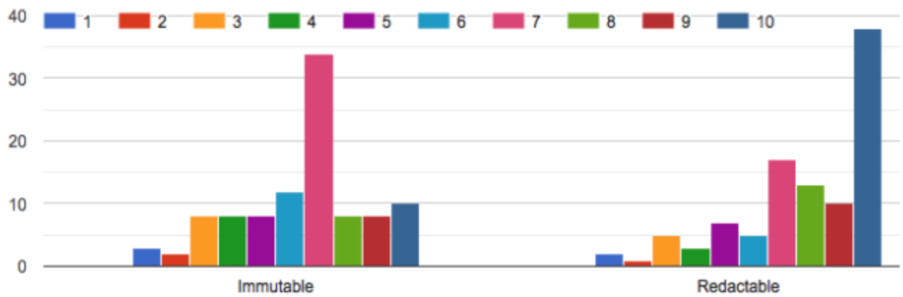
Q 15, Figure 19. Blockchain has already evolved from totally permission-less public Blockchains like Bitcoin to permissioned Blockchains like J.P. Morgan's Quorum, which involves nodes obtaining permission from some higher authority to join the network. To move towards an editable blockchain is just an extra phase in this technology evolution.



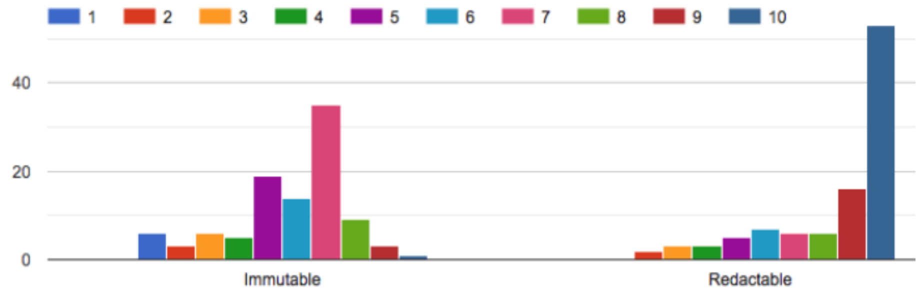
Q 16, Figure 20. Please rate for both types of blockchain technology your knowledge :



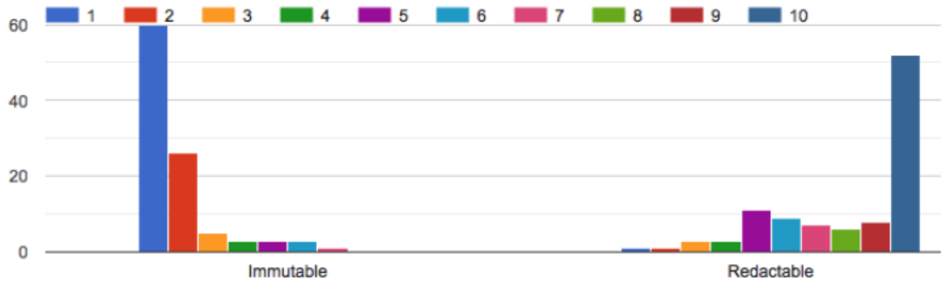
Q 17, Figure 21. Please rate for both types of blockchain technology the effectiveness:



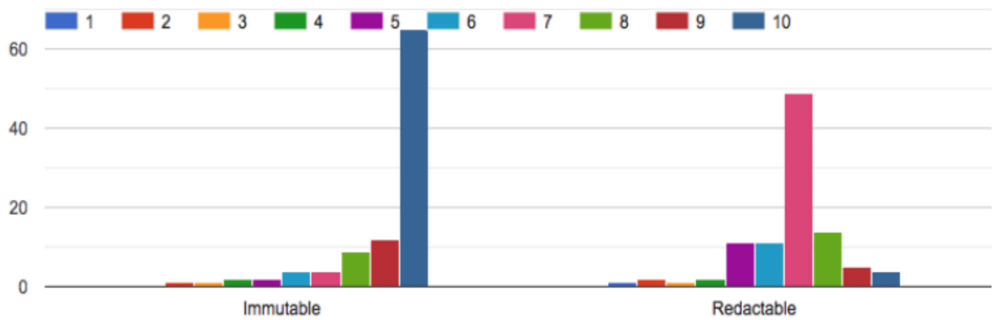
Q 18, Figure 22. Please rate the suitability of immutable or redactable blockchain technology for their stakeholders. For example, data entry staff, solicitors, accountants, auditors and others :



Q 19, Figure 23. Please rate for both types of blockchain technology the compliance with regulations and law :



Q 20, Figure 24. Please rate for both types of blockchain technology the security and resilience to fraud :



5 Preparation for implementation

In R Studio the following libraries were installed:

formattable, version 0.2.0.1: makes presentation richer, simpler, more flexible and deliver more information. Is for formatting on data frames and vectors.

rstatix, version 0.6.0: for execution of fundamental statistical tests, t-test, Wilcoxon test, Cramer V to check the relationship among categorical variables. Helper functions for recognising multivariate and univariate outliers, evaluating homogeneity and normality of variances.

dplyr, version 1.4.4: for work with distant database tables, in case of in-memory data frames.

ggplot, version 2.3.2: graphics creator that will follow the instructions to map variables exactly per the guidelines

coin, version 1.3 – 1: gives provisional interpretation for the overall independence problem including multivariate, correlation, two-sample problems.

pwr, version 1.8.6: calculating effect sizes for all the preceding tests analogous to usual effect sizes (large, medium, small,)

tidyverse, version 1.3.0: gathers multipurpose packages: dplyr, ggplot2, tidyr, etc. The packages work in synchronization to visualize data. Needed for modelling, processing, cleaning purposes.

ggpubr, version 0.4.0: offers simple functions for generating and altering prepared by ggplot2 plots.

PairedData, version 1.1.1: used for statistics, effect sizes and hypothesis tests are provided for analysing paired data, set of graphics (based on ggplot2) and datasets.

na. tools, version 0.3.1: In case of missing values in vectors, this essential instrument gives a reliable basis.

6 Project Implementation

6.1 Average scores

To find out average scores, the answers to question 6 used, the following inputs required:

```
#Finding the average score
test1 <- Data[,c(grep("Q6.*", colnames(Data)))]

library("rstatix")
Technology <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Technology...media...and.telecommunications., na.rm = T)
Financial <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Financial.services., na.rm = T)
Manufacturing <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Manufacturing., na.rm = T)
Retail <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Retail...wholesale...logistics...and.distribution., na.na.rm = T)
Industrial <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Industrial.products.and.construction., na.rm = T)
Automotive <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Automotive., na.rm = T)
Life <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Life.sciences.and.health.care., na.rm = T)
Government <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Government.and.public.services., na.rm = T)

#x <- data.frame("Sector", "Neutral", "No", "Yes")

z <- rbind("Technology, media, and telecommunications" = Technology$n,
           "Financial services" = Financial$n,
           "Manufacturing" = Manufacturing$n,
           "Retail, wholesale, logistics, and distribution" = Retail$n,
           "Industrial products and construction" = Industrial$n,
           "Automotive" = Automotive$n,
           "Life sciences and health care" = Life$n,
           "Government and public services" = Government$n)
z <- as.data.frame(z)

names(z)[names(z) == "V1"] <- "Neutral"
names(z)[names(z) == "V2"] <- "Immutable"
names(z)[names(z) == "V3"] <- "Redactable"
library(formattable)
formattable(z,
            align = c("l",rep("r", NCOL(z) - 1)),
            list(`Neutral` = formatter("span",x ~ percent(x / 100)),
                 `Immutable` = formatter("span",x ~ percent(x / 100)),
                 `Redactable` = formatter("span",x ~ percent(x / 100)))
```

The outputs received:

```
Attaching package: 'rstatix'

The following object is masked from 'package:stats':

    filter

> Technology <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Technology...media...and.telecommunications., na.rm = T)
> Financial <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Financial.services., na.rm = T)
> Manufacturing <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Manufacturing., na.rm = T)
> Retail <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Retail...wholesale...logistics..and.distribution., na.na.rm = T)
> Industrial <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Industrial.products.and.construction., na.rm = T)
> Automotive <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Automotive., na.rm = T)
> Life <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Life.sciences.and.health.care., na.rm = T)
> Government <-
freq_table(test1$Q6.In.your.opinion...redactable.blockchain.would.suit.in.the.follo
wing.industries...Government.and.public.services., na.rm = T)
> #x <- data.frame("Sector", "Neutral", "No", "Yes")
> z <- rbind("Technology, media, and telecommunications" = Technology$n,
+             "Financial services" = Financial$n,
+             "Manufacturing" = Manufacturing$n,
+             "Retail, wholesale, logistics, and distribution" = Retail$n,
+             "Industrial products and construction" = Industrial$n,
+             "Automotive" = Automotive$n,
+             "Life sciences and health care" = Life$n,
+             "Government and public services" = Government$n)
> z <- as.data.frame(z)
>
> names(z)[names(z) == "V1"] <- "Neutral"
> names(z)[names(z) == "V2"] <- "Immutable"
> names(z)[names(z) == "V3"] <- "Redactable"
> library(formattable)
> formattable (z,
+              align = c ("l", rep ("r", NCOL(z) - 1)),
+              list (`Neutral` = formatter("span",x ~ percent(x / 100)),
+                   `Immutable` = formatter ("span",x ~ percent(x / 100)),
+                   `Redactable` = formatter("span",x ~ percent(x / 100)))
+              )
```

The percentages of respondents per industry are calculated by R Studio are below:

	Neutral	Immutable	Redactable
Technology, media, and telecommunications	6.00%	86.00%	8.00%
Financial services	3.00%	22.00%	75.00%
Manufacturing	9.00%	25.00%	66.00%
Retail, wholesale, logistics, and distribution	13.00%	70.00%	16.00%
Industrial products and construction	15.00%	71.00%	14.00%
Automotive	10.00%	70.00%	20.00%
Life sciences and health care	4.00%	54.00%	40.00%
Government and public services	3.00%	90.00%	7.00%

6.2 Correlation between variables

To find out if the correlation exits between knowledge, age, industry and years in profession, the answers to questions 16, 1, 3 and 4 used, the following inputs required:

```
required_packages <- c('MASS', 'rcompanion', 'lsr', 'vcd', 'DescTools')
for (p in required_packages) {
  if(!require(p,character.only = TRUE)) {
    install.packages(p, dep = TRUE)
  }
}

#Read the data. Change the file path here.
Data <-  read.csv("~/Desktop/Reasearch/Tanya-2/Copy of Research Question _ Blockchain
Technology _ Edit or not _.csv/Copy of Research Question _ Blockchain Technology _
Edit or not _.csv")


## If Knowledge >= 6 Then Expert else Novice
#Creating the new catgorical variable for Immutable blockchain knowledge
Blockchain$Immutable_Knowledge_level_cat                                           <-
ifelse(Blockchain$Data.Q16.Please.rate.your.knowledge.about.both.types.of.blockchai
n.technology...No.very.little.knowledge..Some.knowledge..Very.knowledgeable...An.ex
pert......Immutable. >= 6,
                        c("Expert"), c("Novice"))

#Creating the new categorical variable for Redactable blockchain knowledge
Blockchain$Redactable_Knowledge_level_cat                                          <-
ifelse(Blockchain$Data.Q16.Please.rate.your.knowledge.about.both.types.of.blockchai
n.technology...No.very.little.knowledge..Some.knowledge..Very.knowledgeable...An.ex
pert......Redactable. >= 6,
                         c("Expert"), c("Novice"))
```

```
Blockchain$Immutable_Knowledge_level_cat                                    <-
as.factor(Blockchain$Immutable_Knowledge_level_cat)
Blockchain$Redactable_Knowledge_level_cat                                   <-
as.factor(Blockchain$Redactable_Knowledge_level_cat)

#Immutable Blockchain
library('MASS')

#Immutable Blockchain Correlations Crosstab generation
table(Blockchain$Data.Q1.Age.group...,Blockchain$Immutable_Knowledge_level_cat)
table(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Immutable_Knowledge_level_cat)
table(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Immutable_Knowledge_level_cat)

#Redacatble Blockchain Correlations Crosstab generation
table(Blockchain$Data.Q1.Age.group..., Blockchain$Redactable_Knowledge_level_cat)
table(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Redactable_Knowledge_level_cat)
table(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Redactable_Knowledge_level_cat)

#Now we have both the categorical variables. So, we will apply Pearson's Chi Square
statistics and Cramer's V coeeficient to check the correlations.(Immutable
Blockchain)
chisq.test(Blockchain$Data.Q1.Age.group...,
Blockchain$Immutable_Knowledge_level_cat)
#Here p value (0.2467) > 0.05 hence we conclude that two variables are dependent.
chisq.test(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Immutable_Knowledge_level_cat)
#Here p value (0.0553) > 0.05 hence we conclude that two variables are dependent.
chisq.test(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Immutable_Knowledge_level_cat)
#Here p value (0.7208) > 0.05 hence we conclude that two variables are dependent.


#Now we have both the categorical variables. So, we will apply Pearson's Chi Square
statistics and Cramer's V coeeficient to check the correlations.(Redactable
Blockchain)
chisq.test(Blockchain$Data.Q1.Age.group...,
Blockchain$Redactable_Knowledge_level_cat)
#Here p value (0.2143) > 0.05 hence we conclude that two variables are dependent.
chisq.test(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Redactable_Knowledge_level_cat)
#Here p value (0.104) > 0.05 hence we conclude that two variables are dependent.
chisq.test(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Redactable_Knowledge_level_cat)
#Here p value (0.7188) > 0.05 hence we conclude that two variables are dependent.

library('DescTools')
#Calculating (corrected) contintency coefficient
#Original COefficients
#Immutable Blockchain
ContCoef(Blockchain$Data.Q1.Age.group..., Blockchain$Immutable_Knowledge_level_cat,
correct = FALSE) #0.2488
ContCoef(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Immutable_Knowledge_level_cat, correct = FALSE) #0.3888
ContCoef(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Immutable_Knowledge_level_cat, correct = FALSE) #0.2554

#Redactable Blockchain
ContCoef(Blockchain$Data.Q1.Age.group...,
Blockchain$Redactable_Knowledge_level_cat, correct = FALSE) #0.2560
```

```
ContCoef(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Redactable_Knowledge_level_cat, correct = FALSE) #0.3683
ContCoef(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Redactable_Knowledge_level_cat, correct = FALSE) #0.2558

#Corrected Coefficients
#Immutable Blockchain
ContCoef(Blockchain$Data.Q1.Age.group...,  Blockchain$Immutable_Knowledge_level_cat,
correct = TRUE) #0.3518
ContCoef(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Immutable_Knowledge_level_cat, correct = TRUE) #0.5498
ContCoef(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Immutable_Knowledge_level_cat, correct = TRUE) #0.3612

#Redactable Blockchain
ContCoef(Blockchain$Data.Q1.Age.group...,
Blockchain$Redactable_Knowledge_level_cat, correct = TRUE) #0.3621
ContCoef(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Redactable_Knowledge_level_cat, correct = TRUE) #0.5208
ContCoef(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Redactable_Knowledge_level_cat, correct = TRUE) #0.3617

#Calculating Cramer's V coefficient
library('rcompanion')
cramerV(Blockchain$Data.Q1.Age.group...,  Blockchain$Immutable_Knowledge_level_cat,
bias.correct = FALSE) #0.2569
cramerV(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Immutable_Knowledge_level_cat, bias.correct = FALSE) #0.422
cramerV(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Immutable_Knowledge_level_cat, bias.correct = FALSE) #0.2642

cramerV(Blockchain$Data.Q1.Age.group.,   Blockchain$Redactable_Knowledge_level_cat,
bias.correct = FALSE) #0.2649
cramerV(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Redactable_Knowledge_level_cat, bias.correct = FALSE) #0.3961
cramerV(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Redactable_Knowledge_level_cat, bias.correct = FALSE) #0.2646


The outputs received are below:

> #Immutable Blockchain Correlations Crosstab generation
> table(Blockchain$Data.Q1.Age.group...,Blockchain$Immutable_Knowledge_level_cat)


                   Expert Novice
  <18            3    0
  >50            5    3
  18 to 25      14    8
  26 to 35      24    3
  36 to 45      19    8
  46 to 50       9    5
```

```
> table (Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Immutable_Knowledge_level_cat)

                                                Expert  Novice
  Automotive                                         2     5
  Education                                          1     0
  Financial services                                17     2
  Government and public services                     1     1
  Industrial products and construction               1     2
  Law                                                5     5
  Manufacturing sale, logistics, and distribution    7     2
  Student                                            2     0
  Technology, media and telecommunications          24     5
> table(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Immutable_Knowledge_level_cat)

          Expert Novice
  >10      44  17
  1         2  0
  10        2  0
  2         4  1
  3         7  1
  4         4  3
  5         4  1
  6         3  1
  7         1  0
  8         2  3
  9         1  0
> #Redacatble Blockchain Correlations Crosstab generation
> table(Blockchain$Data.Q1.Age.group..., Blockchain$Redactable_Knowledge_level_cat)


     Expert Novice
  <18               1       2
  >50               3       5
  18 to 25         13       9
  26 to 35         18       9
  36 to 45         13       14
  46 to 50          4       10
> table(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Redactable_Knowledge_level_cat)

                                                                     Expert
Novice
  Automotive                                             1  6
  Education                                              1  0
  Financial services                                    11  8
  Government and public services                         0  2
  Industrial products and construction                   1  2
  Law                                                    4  7
  Life sciences and health care                          4  6
  Manufacturing                                          5  3
  Retail, wholesale, logistics, and distribution         3  6
  Student                                                1  1
  Technology, media and telecommunications              21  8
```

```
> table (Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Redactable_Knowledge_level_cat)

      Expert Novice
  >10      28 33
  1          1    1
  10        2     0
  2          3    2
  3          6    2
  4          4    3
  5          3    2
  6          2    2
  7          1    0
  8          2    3
  9          0    1
> #Now we have both the categorical variables. So, we will apply Pearson's Chi Square
statistics and Cramer's V coeeficient to check the correlations.(Immutable
Blockchain)
>chisq.test(Blockchain$Data.Q1.Age.group..,
Blockchain$Immutable_Knowledge_level_cat)

Pearson's Chi-squared test

data:  Blockchain$Data.Q1.Age.group... and Blockchain$Immutable_Knowledge_level_cat
X-squared = 6.6663, df = 5, p-value = 0.2467

Warning message:
Inchisq.test(Blockchain$Data.Q1.Age.group...,
Blockchain$Immutable_Knowledge_level_cat) :
  Chi-squared approximation may be incorrect
> #Here p value (0.2467) > 0.05 hence we conclude that two variables are dependent.
>chisq.test(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Immutable_Knowledge_level_cat)

Pearson's Chi-squared test

data:                 Blockchain$Data.Q3.What.industry.do.you.work.in...         and
Blockchain$Immutable_Knowledge_level_cat
X-squared = 17.983, df = 10, p-value = 0.05526

Warning message:
In chisq.test(Blockchain$Data.Q3.What.industry.do.you.work.in...,   :
  Chi-squared approximation may be incorrect
> #Here p value (0.0553) > 0.05 hence we conclude that two variables are dependent.
>      chisq.test(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Immutable_Knowledge_level_cat)

Pearson's Chi-squared test

data:Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.           and
Blockchain$Immutable_Knowledge_level_cat
X-squared = 7.0488, df = 10, p-value = 0.7208

Warning message:
In chisq.test(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,   :
  Chi-squared approximation may be incorrect
> #Here p value (0.7208) > 0.05 hence we conclude that two variables are dependent.
> #Now we have both the categorical variables. So, we will apply Pearson's Chi Square
statistics and Cramer's V coeeficient to check the correlations.(Redactable
Blockchain)
>                                         chisq.test(Blockchain$Data.Q1.Age.group...,
Blockchain$Redactable_Knowledge_level_cat)

Pearson's Chi-squared test
```

```
data:  Blockchain$Data.Q1.Age.group... and Blockchain$Redactable_Knowledge_level_cat
X-squared = 7.0862, df = 5, p-value = 0.2143

Warning message:
Inchisq.test(Blockchain$Data.Q1.Age.group...,
Blockchain$Redactable_Knowledge_level_cat) :
  Chi-squared approximation may be incorrect
> #Here p value (0.2143) > 0.05 hence we conclude that two variables are dependent.
>chisq.test(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Redactable_Knowledge_level_cat)

Pearson's Chi-squared test

data:                  Blockchain$Data.Q3.What.industry.do.you.work.in...         and
Blockchain$Redactable_Knowledge_level_cat
X-squared = 15.849, df = 10, p-value = 0.104

Warning message:
In chisq.test(Blockchain$Data.Q3.What.industry.do.you.work.in...,   :
  Chi-squared approximation may be incorrect
> #Here p value (0.104) > 0.05 hence we conclude that two variables are dependent.
>      chisq.test(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Redactable_Knowledge_level_cat)

Pearson's Chi-squared test

data:Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.          and
Blockchain$Redactable_Knowledge_level_cat
X-squared = 7.0698, df = 10, p-value = 0.7188

Warning message:
In chisq.test(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,   :
  Chi-squared approximation may be incorrect
> #Here p value (0.7188) > 0.05 hence we conclude that two variables are dependent.
> library('DescTools')
> #Calculating (corrected) contintency coefficient
> #Original COefficients
> #Immutable Blockchain
>ContCoef(Blockchain$Data.Q1.Age.group...,
Blockchain$Immutable_Knowledge_level_cat, correct = FALSE) #0.2488
[1] 0.2488303
>ContCoef(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Immutable_Knowledge_level_cat, correct = FALSE) #0.3888
[1] 0.3887642
>ContCoef(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Immutable_Knowledge_level_cat, correct = FALSE) #0.2554
[1] 0.2554155
> #Redactable Blockchain
>ContCoef(Blockchain$Data.Q1.Age.group...,
Blockchain$Redactable_Knowledge_level_cat, correct = FALSE) #0.2560
[1] 0.2560484
>ContCoef(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Redactable_Knowledge_level_cat, correct = FALSE) #0.3683
[1] 0.3682896
>ContCoef(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Redactable_Knowledge_level_cat, correct = FALSE) #0.2558
[1] 0.2557714
> #Corrected Ceffficients
> #Immutable Blockchain
>ContCoef(Blockchain$Data.Q1.Age.group...,
Blockchain$Immutable_Knowledge_level_cat, correct = TRUE) #0.3518
[1] 0.3518992
>ContCoef(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Immutable_Knowledge_level_cat, correct = TRUE) #0.5498
```

```
[1] 0.5497956
>ContCoef(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Immutable_Knowledge_level_cat, correct = TRUE) #0.3612
[1] 0.361212
> #Redactable Blockchain
>ContCoef(Blockchain$Data.Q1.Age.group...,
Blockchain$Redactable_Knowledge_level_cat, correct = TRUE) #0.3621
[1] 0.3621071
>ContCoef(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Redactable_Knowledge_level_cat, correct = TRUE) #0.5208
[1] 0.5208402
>ContCoef(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Redactable_Knowledge_level_cat, correct = TRUE) #0.3617
[1] 0.3617154
> #Calculating Cramer's V coefficient
> library('rcompanion')
>cramerV(Blockchain$Data.Q1.Age.group...,  Blockchain$Immutable_Knowledge_level_cat,
bias.correct = FALSE) #0.2569
Cramer V
  0.2569
>cramerV(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Immutable_Knowledge_level_cat, bias.correct = FALSE) #0.422
Cramer V
   0.422
>cramerV(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Immutable_Knowledge_level_cat, bias.correct = FALSE) #0.2642
Cramer V
  0.2642
>cramerV(Blockchain$Data.Q1.Age.group...,
Blockchain$Redactable_Knowledge_level_cat, bias.correct = FALSE) #0.2649
Cramer V
  0.2649
>cramerV(Blockchain$Data.Q3.What.industry.do.you.work.in...,
Blockchain$Redactable_Knowledge_level_cat, bias.correct = FALSE) #0.3961
Cramer V
  0.3961
>cramerV(Blockchain$Data.Q4.How.many.years.of.work.experience.do.you.have.,
Blockchain$Redactable_Knowledge_level_cat, bias.correct = FALSE) #0.2646
Cramer V
  0.2646
>
```

6.3 Wilcoxon signed rank test (alternative of t – test)

Shapiro - Wilk test used to check the assumptions if the data is normally distributed. In this case the alternative to t – test, Wilcoxon test was used. The following inputs required:

```
#T-Tests
#1) Regarding Blockchain Knowldge


# Create a data frame
blockchain_knowledge <- data.frame(
  group = rep(c("Immutable", "Redactable"), each = 101),
  weight = c(Immutable_Knowledge,  Redactable_Knowledge)
)
#Computing Group Statistics
library("dplyr")
group_by(blockchain_knowledge, group) %>%
  summarise(
    count = n(),
    mean = mean(weight, na.rm = TRUE),
    sd = sd(weight, na.rm = TRUE)
  )
# Plot weight by group and color by group
library("ggpubr")
ggboxplot(blockchain_knowledge, x = "group", y = "weight",
          color = "group", palette = c("#00AFBB", "#E7B800"),
          order = c("Immutable", "Redactable"),
          ylab = "Weight", xlab = "Groups")

# Plot paired data
library(PairedData)
pd <- paired(Immutable_Knowledge, Redactable_Knowledge)
plot(pd, type = "profile") + theme_bw()

#Checking the Assumptions
#1) Are the two samples paired? YES. As both the observations are from the same
sample.
#2) Is this a large sample? YES n>30
#3) Is the difference between the two groups follows normal distribution? To do this,
we will perform Shapiro-Wilk Test of normality.
#Shapiro-Wilk Test
#Null hypothesis: the data are normally distributed
#Alternative hypothesis: the data are not normally distributed

# compute the difference
d_knowledge <- with(blockchain_knowledge,
          weight[group == "Immutable"] - weight[group == "Redactable"])
# Shapiro-Wilk normality test for the differences
shapiro.test(d_knowledge)

library("ggpubr")
ggdensity(d_knowledge,
          main = "Density plot of difference",
          xlab = "Diference")

#As the P value is Shapiro Wilk test is < 0.05, the differences do not follow the
normal distribution. Hence, we need to perform the non-parametric alternative in this
case.
test_knowledge <- wilcox.test(Immutable_Knowledge,Redactable_Knowledge, paired=TRUE,
exact = FALSE, conf.int = TRUE,conf.level = 0.95)
test_knowledge
Zstat_knowledge<-qnorm(test_knowledge$p.value/2)
```

```
abs(Zstat_knowledge)/sqrt(101) #Total observations: 101
#Effect size is large in this case as value : 0.86
median(Immutable_Knowledge)
median(Redactable_Knowledge)

#Here p value < 0.001 hence there is significant difference between knowledge of immutable and redactable blockchains.
```

```
#2) Regarding Blockchain Effectiveness


# Create a data frame
blockchain_effective <- data.frame(
  group = rep(c("Immutable", "Redactable"), each = 101),
  weight = c(Immutable_Effective,  Redactable_Effective)
)

#Computing Group Statistics
library("dplyr")
group_by(blockchain_effective, group) %>%
  summarise(
    count = n(),
    mean = mean(weight, na.rm = TRUE),
    sd = sd(weight, na.rm = TRUE)
  )

# Plot weight by group and color by group
library("ggpubr")
ggboxplot(blockchain_effective, x = "group", y = "weight",
          color = "group", palette = c("#00AFBB", "#E7B800"),
          order = c("Immutable", "Redactable"),
          ylab = "Weight", xlab = "Groups")

# Plot paired data
library(PairedData)
pd <- paired(Immutable_Effective, Redactable_Effective)
plot(pd, type = "profile") + theme_bw()

#Checking the Assumptions
#1) Are the two samples paired? YES. As both the observations are from the same sample.
#2) Is this a large sample? YES n>30
#3) Is the difference between the two groups follows normal distribution? To do this, we will perform Shapiro-Wilk Test of normality.
#Shapiro-Wilk Test
#Null hypothesis: the data are normally distributed
#Alternative hypothesis: the data are not normally distributed

# compute the difference
d_effective <- with(blockchain_effective,
          weight[group == "Immutable"] - weight[group == "Redactable"])
# Shapiro-Wilk normality test for the differences
shapiro.test(d_effective)

library("ggpubr")
ggdensity(d_effective,
          main = "Density plot of difference",
          xlab = "Diference")

#As the P value in Shapiro Wilk test is < 0.05, the differences do not follow the normal distribution. Hence, we need to perform the non-parametric alternative in this case.
```

```
test_effective <- wilcox.test(Immutable_Effective,Redactable_Effective, paired=TRUE,
exact = FALSE, conf.int = TRUE,conf.level = 0.95)
test_effective
Zstat_effective<-qnorm(test_effective$p.value/2)
abs(Zstat_effective)/sqrt(101) #Total observations: 101
#Effect size is medium in this case as value : 0.44
median(Immutable_Effective)
median(Redactable_Effective)

#Here p value < 0.001 hence there is significant difference between effectiveness of
immutable and redactable blockchains.
```

```
#3) Regarding Blockchain Suitability

# Create a data frame
blockchain_suitability <- data.frame(
  group = rep (c("Immutable", "Redactable"), each = 101),
  weight = c(Immutable_Suitability,  Redactable_Suitability)
)

#Computing Group Statistics
library("dplyr")
group_by(blockchain_suitability, group) %>%
  summarise(
    count = n(),
    mean = mean(weight, na.rm = TRUE),
    sd = sd(weight, na.rm = TRUE)
  )

# Plot weight by group and color by group
library("ggpubr")
ggboxplot(blockchain_suitability, x = "group", y = "weight",
          color = "group", palette = c("#00AFBB", "#E7B800"),
          order = c("Immutable", "Redactable"),
          ylab = "Weight", xlab = "Groups")

# Plot paired data
library(PairedData)
pd <- paired(Immutable_Suitability, Redactable_Suitability)
plot(pd, type = "profile") + theme_bw()


#Checking the Assumptions
#1) Are the two samples paired? YES. As both the observations are from the same
sample.
#2) Is this a large sample? YES n>30
#3) Is the difference between the two groups follows normal distribution? To do this,
we will perform Shapiro-Wilk Test of normality.
#Shapiro-Wilk Test
#Null hypothesis: the data are normally distributed
#Alternative hypothesis: the data are not normally distributed

# compute the difference
d_suitability <- with(blockchain_suitability,
          weight[group == "Immutable"] - weight[group == "Redactable"])
# Shapiro-Wilk normality test for the differences
shapiro.test(d_suitability)

library("ggpubr")
ggdensity(d_suitability,
          main = "Density plot of difference",
          xlab = "Diference")
```

```
#As the P value is Shapiro Wilk test is < 0.05, the differences do not follow the normal distribution. Hence, we need to perform the non-parametric alternative in this case.
test_suitability    <-    wilcox.test(Immutable_Suitability,Redactable_Suitability, paired=TRUE, exact = FALSE, conf.int = TRUE,conf.level = 0.95)
test_suitability
Zstat_suitability<-qnorm(test_suitability$p.value/2)
abs(Zstat_suitability)/sqrt(101) #Total observations: 101
#Effect size is large in this case as value : 0.75
median(Immutable_Suitability)
median(Redactable_Suitability)

#Here p value < 0.001 hence there is significant difference between suitability of immutable and redactable blockchains.
```

```
#4) Regarding Blockchain Compliance

# Create a data frame
blockchain_compliance <- data.frame(
  group = rep(c("Immutable", "Redactable"), each = 101),
  weight = c(Immutable_Compliance,  Redactable_Compliance)
)

#Computing Group Statistics
library("dplyr")
group_by(blockchain_compliance, group) %>%
  summarise(
    count = n(),
    mean = mean(weight, na.rm = TRUE),
    sd = sd(weight, na.rm = TRUE)
  )

# Plot weight by group and color by group
library("ggpubr")
ggboxplot(blockchain_compliance, x = "group", y = "weight",
          color = "group", palette = c("#00AFBB", "#E7B800"),
          order = c("Immutable", "Redactable"),
          ylab = "Weight", xlab = "Groups")

# Plot paired data
library(PairedData)
pd <- paired(Immutable_Compliance, Redactable_Compliance)
plot(pd, type = "profile") + theme_bw()

#Checking the Assumptions
#1) Are the two samples paired? YES. As both the observations are from the same sample.
#2) Is this a large sample? YES n>30
#3) Is the difference between the two groups follows normal distribution? To do this, we will perform Shapiro-Wilk Test of normality.
#Shapiro-Wilk Test
#Null hypothesis: the data are normally distributed
#Alternative hypothesis: the data are not normally distributed

# compute the difference
d_compliance <- with(blockchain_compliance,
          weight[group == "Immutable"] - weight[group == "Redactable"])
# Shapiro-Wilk normality test for the differences
shapiro.test(d_compliance)

library("ggpubr")
ggdensity(d_compliance,
          main = "Density plot of difference",
```

```
          xlab = "Diference")

#As the P value is Shapiro Wilk test is < 0.05, the differences do not follow the normal distribution. Hence, we need to perform the non-parametric alternative in this case.
test_compliance      <-      wilcox.test(Immutable_Knowledge,Redactable_Knowledge, paired=TRUE, exact = FALSE, conf.int = TRUE,conf.level = 0.95)
test_compliance
Zstat_compliance<-qnorm(test_compliance$p.value/2)
abs(Zstat_compliance)/sqrt(101) #Total observations: 101
#Effect size is large in this case as value : 0.80
median(Immutable_Compliance)
median(Redactable_Compliance)

#Here p value < 0.001 hence there is significant difference between compliance of immutable and redactable blockchains.
```

```
#5) Regarding Blockchain Security and Resilience to Fraud

# Create a data frame
blockchain_security <- data.frame(
  group = rep(c("Immutable", "Redactable"), each = 100),
  weight = c(Immutable_Security,  Redactable_Security)
)

#Computing Group Statistics
library("dplyr")
group_by(blockchain_security, group) %>%
  summarise(
    count = n(),
    mean = mean (weight, na.rm = TRUE),
    sd = sd(weight, na.rm = TRUE)
  )

# Plot weight by group and color by group
library("ggpubr")
ggboxplot(blockchain_security, x = "group", y = "weight",
          color = "group", palette = c("#00AFBB", "#E7B800"),
          order = c("Immutable", "Redactable"),
          ylab = "Weight", xlab = "Groups")

# Plot paired data
library(PairedData)
pd <- paired(Immutable_Security, Redactable_Security)
plot(pd, type = "profile") + theme_bw()

#Checking the Assumptions
#1) Are the two samples paired? YES. As both the observations are from the same sample.
#2) Is this a large sample? YES n>30
#3) Is the difference between the two groups follows normal distribution? To do this, we will perform Shapiro-Wilk Test of normality.
#Shapiro-Wilk Test
#Null hypothesis: the data are normally distributed
#Alternative hypothesis: the data are not normally distributed

# compute the difference
d_security <- with(blockchain_security,
                     weight[group == "Immutable"] - weight[group == "Redactable"])
# Shapiro-Wilk normality test for the differences
shapiro.test(d_security)

library("ggpubr")
```

```
ggdensity(d_security,
          main = "Density plot of difference",
          xlab = "Diference")


#As the P value is Shapiro Wilk test is < 0.05, the differences do not follow the
normal distribution. Hence, we need to perform the non-parametric alternative in this
case.
test_security <- wilcox.test(Immutable_Knowledge,Redactable_Knowledge, paired=TRUE,
exact = FALSE, conf.int = TRUE,conf.level = 0.95)
test_security
Zstat_security <-qnorm(test_security$p.value/2)
abs(Zstat_security)/sqrt(100) #Total observations: 10
#Effect size is large in this case as value : 0.81
median(Immutable_Security)
median(Redactable_Security)

#Here p value < 0.001 hence there is significant difference between security of
immutable and redactable blockchains.
```

The following outputs were generated:

```
> #1) Regarding Blockchain Knowldge

`summarise()` ungrouping output (override with `.groups` argument)
# A tibble: 2 x 4
  group      count  mean    sd
  <chr>      <int> <dbl> <dbl>
1 Immutable    101  6.77  2.17
2 Redactable   101  5.51  2.09
> # Plot weight by group and color by group
> library("ggpubr")
> ggboxplot(blockchain_knowledge, x = "group", y = "weight",
+           color = "group", palette = c("#00AFBB", "#E7B800"),
+           order = c("Immutable", "Redactable"),
+           ylab = "Weight", xlab = "Groups")
> # Plot paired data
> library(PairedData)
> pd <- paired(Immutable_Knowledge, Redactable_Knowledge)
> plot(pd, type = "profile") + theme_bw()
> #Checking the Assumptions
> #1) Are the two samples paired? YES. As both the observations are from the same
sample.
> #2) Is this a large sample? YES n>30
> #3) Is the difference between the two groups follows normal distribution? To do
this, we will perform Shapiro-Wilk Test of normality.
> #Shapiro-Wilk Test
> #Null hypothesis: the data are normally distributed
> #Alternative hypothesis: the data are not normally distributed
> # compute the difference
> d_knowledge <- with(blockchain_knowledge,
+           weight[group == "Immutable"] - weight[group == "Redactable"])
> # Shapiro-Wilk normality test for the differences
> shapiro.test(d_knowledge)


Shapiro-Wilk normality test

data:  d_knowledge
W = 0.65782, p-value = 5.253e-14

> library("ggpubr")
```

```
> ggdensity(d_knowledge,
+           main = "Density plot of difference",
+           xlab = "Diference")
> #As the P value is Shapiro Wilk test is < 0.05, the differences do not follow the
normal distribution. Hence, we need to perform the non-parametric alternative in
this case.
> test_knowledge <- wilcox.test(Immutable_Knowledge,Redactable_Knowledge,
paired=TRUE, exact = FALSE, conf.int = TRUE,conf.level = 0.95)
> test_knowledge

Wilcoxon signed rank test with continuity correction

data:  Immutable_Knowledge and Redactable_Knowledge
V = 4150.5, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 1.000025 1.499996
sample estimates:
(pseudo)median
      1.000088

> Zstat_knowledge<-qnorm(test_knowledge$p.value/2)
> abs(Zstat_knowledge)/sqrt(101) #Total observations: 101
[1] 0.8599367
> #Effect size is large in this case as value : 0.86
> median(Immutable_Knowledge)
[1] 7
> median(Redactable_Knowledge)
[1] 6
> #2) Regarding Blockchain Effectivness

`summarise()` ungrouping output (override with `.groups` argument)
# A tibble: 2 x 4
  group      count  mean    sd
  <chr>      <int> <dbl> <dbl>
1 Immutable    101  6.43  2.23
2 Redactable   101  7.81  2.38
> # Plot weight by group and color by group
> library("ggpubr")
> ggboxplot(blockchain_effective, x = "group", y = "weight",
+           color = "group", palette = c("#00AFBB", "#E7B800"),
+           order = c("Immutable", "Redactable"),
+           ylab = "Weight", xlab = "Groups")
> # Plot paired data
> library(PairedData)
> pd <- paired(Immutable_Effective, Redactable_Effective)
> plot(pd, type = "profile") + theme_bw()
> #Checking the Assumptions
> #1) Are the two samples paired? YES. As both the observations are from the same
sample.
> #2) Is this a large sample? YES n>30
> #3) Is the difference between the two groups follows normal distribution? To do
this, we will perform Shapiro-Wilk Test of normality.
> #Shapiro-Wilk Test
> #Null hypothesis: the data are normally distributed
> #Alternative hypothesis: the data are not normally distributed
> # compute the difference
> d_effective <- with(blockchain_effective,
+           weight[group == "Immutable"] - weight[group == "Redactable"])
> # Shapiro-Wilk normality test for the differences
> shapiro.test(d_effective)

Shapiro-Wilk normality test
```

```
data:  d_effective
W = 0.96791, p-value = 0.01458

> library("ggpubr")
> ggdensity(d_effective,
+           main = "Density plot of difference",
+           xlab = "Diference")
> #As the P value in Shapiro Wilk test is < 0.05, the differences do not follow the
normal distribution. Hence, we need to perform the non-parametric alternative in
this case.
> test_effective <- wilcox.test(Immutable_Effective,Redactable_Effective,
paired=TRUE, exact = FALSE, conf.int = TRUE,conf.level = 0.95)
> test_effective

Wilcoxon signed rank test with continuity correction

data:  Immutable_Effective and Redactable_Effective
V = 751.5, p-value = 9.386e-06
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -2.5000586 -0.9999657
sample estimates:
(pseudo)median
     -1.999993

> Zstat_effective<-qnorm(test_effective$p.value/2)
> abs(Zstat_effective)/sqrt(101) #Total observations: 101
[1] 0.4408871
> #Effect size is medium in this case as value : 0.44
> median(Immutable_Effective)
[1] 7
> median(Redactable_Effective)
[1] 8
```

```
> #3) Regarding Blockchain Suitability
> #Computing Group Statistics
> library("dplyr")
> group_by(blockchain_suitability, group) %>%
+   summarise(
+     count = n(),
+     mean = mean(weight, na.rm = TRUE),
+     sd = sd(weight, na.rm = TRUE)
+   )
`summarise()` ungrouping output (override with `.groups` argument)
# A tibble: 2 x 4
  group      count  mean    sd
  <chr>      <int> <dbl> <dbl>
1 Immutable    101  5.77  2.00
2 Redactable   101  8.48  2.19
> # Plot weight by group and color by group
> library("ggpubr")
> ggboxplot(blockchain_suitability, x = "group", y = "weight",
+           color = "group", palette = c("#00AFBB", "#E7B800"),
+           order = c("Immutable", "Redactable"),
+           ylab = "Weight", xlab = "Groups")
> # Plot paired data
> library(PairedData)
> pd <- paired(Immutable_Suitability, Redactable_Suitability)
> plot(pd, type = "profile") + theme_bw()
> #Checking the Assumptions
```

```
> #1) Are the two samples paired? YES. As both the observations are from the same sample.
> #2) Is this a large sample? YES n>30
> #3) Is the difference between the two groups follows normal distribution? To do this, we will perform Shapiro-Wilk Test of normality.
> #Shapiro-Wilk Test
> #Null hypothesis: the data are normally distributed
> #Alternative hypothesis: the data are not normally distributed
> # compute the difference
> d_suitability <- with(blockchain_suitability,
+            weight[group == "Immutable"] - weight[group == "Redactable"])
> # Shapiro-Wilk normality test for the differences
> shapiro.test(d_suitability)

Shapiro-Wilk normality test

data:  d_suitability
W = 0.96388, p-value = 0.007271

> library("ggpubr")
> ggdensity(d_suitability,
+            main = "Density plot of difference",
+            xlab = "Diference")
> #As the P value is Shapiro Wilk test is < 0.05, the differences do not follow the normal distribution. Hence, we need to perform the non-parametric alternative in this case.
> test_suitability <- wilcox.test(Immutable_Suitability,Redactable_Suitability, paired=TRUE, exact = FALSE, conf.int = TRUE,conf.level = 0.95)
> test_suitability

Wilcoxon signed rank test with continuity correction

data:  Immutable_Suitability and Redactable_Suitability
V = 104, p-value = 3.323e-14
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -3.500032 -2.500038
sample estimates:
(pseudo)median
     -3.000001

> Zstat_suitability<-qnorm(test_suitability$p.value/2)
> abs(Zstat_suitability)/sqrt(101) #Total observations: 101
[1] 0.7547441
> #Effect size is large in this case as value : 0.75
> median(Immutable_Suitability)
[1] 6
> median(Redactable_Suitability)
[1] 10
```

```
> #4) Regarding Blockchain Compliance
`summarise()` ungrouping output (override with `.groups` argument)
# A tibble: 2 x 4
  group      count  mean    sd
  <chr>      <int> <dbl> <dbl>
1 Immutable    101  1.77  1.31
2 Redactable   101  8.14  2.37
> # Plot weight by group and color by group
> library("ggpubr")
> ggboxplot(blockchain_compliance, x = "group", y = "weight",
+           color = "group", palette = c("#00AFBB", "#E7B800"),
+           order = c("Immutable", "Redactable"),
+           ylab = "Weight", xlab = "Groups")
> # Plot paired data
> library(PairedData)
> pd <- paired(Immutable_Compliance, Redactable_Compliance)
> plot(pd, type = "profile") + theme_bw()
> #Checking the Assumptions
> #1) Are the two samples paired? YES. As both the observations are from the same
sample.
> #2) Is this a large sample? YES n>30
> #3) Is the difference between the two groups follows normal distribution? To do
this, we will perform Shapiro-Wilk Test of normality.
> #Shapiro-Wilk Test
> #Null hypothesis: the data are normally distributed
> #Alternative hypothesis: the data are not normally distributed
> # compute the difference
> d_compliance <- with(blockchain_compliance,
+           weight[group == "Immutable"] - weight[group == "Redactable"])
> # Shapiro-Wilk normality test for the differences
> shapiro.test(d_compliance)

Shapiro-Wilk normality test

data:  d_compliance
W = 0.81737, p-value = 7.667e-10

> library("ggpubr")
> ggdensity(d_compliance,
+           main = "Density plot of difference",
+           xlab = "Diference")
> #As the P value is Shapiro Wilk test is < 0.05, the differences do not follow the
normal distribution. Hence, we need to perform the non-parametric alternative in
this case.
> test_compliance <- wilcox.test(Immutable_Knowledge,Redactable_Knowledge,
paired=TRUE, exact = FALSE, conf.int = TRUE,conf.level = 0.95)
> test_compliance

Wilcoxon signed rank test with continuity correction

data:  Immutable_Knowledge and Redactable_Knowledge
V = 4150.5, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 1.000025 1.499996
sample estimates:
(pseudo)median
      1.000088

> Zstat_compliance<-qnorm(test_compliance$p.value/2)
> abs(Zstat_compliance)/sqrt(101) #Total observations: 101
[1] 0.8599367
> #Effect size is large in this case as value : 0.80
```

```
> median(Immutable_Compliance)
[1] 1
> median(Redactable_Compliance)
[1] 10
```

```
> #5) Regarding Blockchain Security and Resilience to Fraud

> #Computing Group Statistics
> library("dplyr")
> group_by(blockchain_security, group) %>%
+   summarise(
+     count = n(),
+     mean = mean(weight, na.rm = TRUE),
+     sd = sd(weight, na.rm = TRUE)
+   )
`summarise()` ungrouping output (override with `.groups` argument)
# A tibble: 2 x 4
  group      count  mean    sd
  <chr>      <int> <dbl> <dbl>
1 Immutable    100  9.05  1.72
2 Redactable   100  6.77  1.56
> # Plot weight by group and color by group
> library("ggpubr")
> ggboxplot(blockchain_security, x = "group", y = "weight",
+           color = "group", palette = c("#00AFBB", "#E7B800"),
+           order = c("Immutable", "Redactable"),
+           ylab = "Weight", xlab = "Groups")
> # Plot paired data
> library(PairedData)
> pd <- paired(Immutable_Security, Redactable_Security)
> plot(pd, type = "profile") + theme_bw()
> #Checking the Assumptions
> #1) Are the two samples paired? YES. As both the observations are from the same
sample.
> #2) Is this a large sample? YES n>30
> #3) Is the difference between the two groups follows normal distribution? To do
this, we will perform Shapiro-Wilk Test of normality.
> #Shapiro-Wilk Test
> #Null hypothesis: the data are normally distributed
> #Alternative hypothesis: the data are not normally distributed
> # compute the difference
> d_security <- with(blockchain_security,
+                       weight[group == "Immutable"] - weight[group ==
"Redactable"])
> # Shapiro-Wilk normality test for the differences
> shapiro.test(d_security)

Shapiro-Wilk normality test

data:  d_security
W = 0.89284, p-value = 6.615e-07

> library("ggpubr")
> ggdensity(d_security,
+           main = "Density plot of difference",
+           xlab = "Diference")
> #As the P value is Shapiro Wilk test is < 0.05, the differences do not follow the
normal distribution. Hence, we need to perform the non-parametric alternative in
this case.
> test_security <- wilcox.test(Immutable_Knowledge,Redactable_Knowledge,
paired=TRUE, exact = FALSE, conf.int = TRUE,conf.level = 0.95)
> test_security
```

```
Wilcoxon signed rank test with continuity correction

data:  Immutable_Knowledge and Redactable_Knowledge
V = 4150.5, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 1.000025 1.499996
sample estimates:
(pseudo)median
      1.000088

> Zstat_security <-qnorm(test_security$p.value/2)
> abs(Zstat_security)/sqrt(100) #Total observations: 10
[1] 0.8642256
> #Effect size is large in this case as value : 0.81
> median(Immutable_Security)
[1] 10
> median(Redactable_Security)
>[1] 7
```