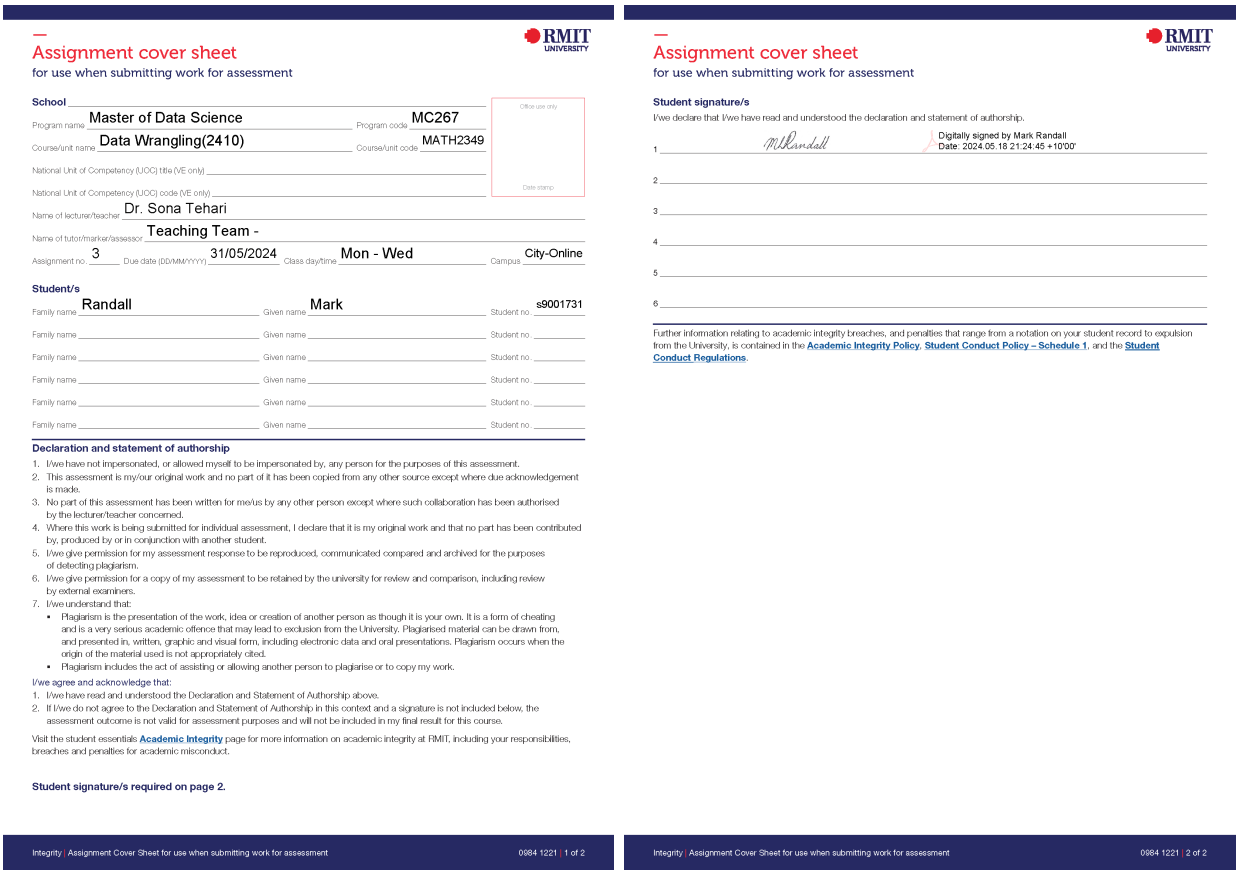
Data Wrangling (Data Preprocessing)

Practical Assessment 2

s9001731 Mark randall

2024-06-19



Assignment cover sheet

for use when submitting work for assessment

RMIT UNIVERSITY

School

Program name: Master of Data Science — Program code: MC267

Course/unit name: Data Wrangling(2410) — Course/unit code: MATH2349

National Unit of Competency (UOC) title (VE only)

National Unit of Competency (UOC) code (VE only)

Office use only — Date stamp

Name of lecturer/teacher: Dr. Sona Tehari

Name of tutor/marker/assessor: Teaching Team -

Assignment no.: 3 — Due date (DD/MM/YYYY): 31/05/2024 — Class day/time: Mon - Wed — Campus: City-Online

Student/s

Family name: Randall — Given name: Mark — Student no.: s9001731

Family name — Given name — Student no.

Family name — Given name — Student no.

Family name — Given name — Student no.

Family name — Given name — Student no.

Family name — Given name — Student no.

Declaration and statement of authorship

1. I/we have not impersonated, or allowed myself to be impersonated by, any person for the purposes of this assessment.
2. This assessment is my/our original work and no part of it has been copied from any other source except where due acknowledgement is made.
3. No part of this assessment has been written for me/us by any other person except where such collaboration has been authorised by the lecturer/teacher concerned.
4. Where this work is being submitted for individual assessment, I declare that it is my original work and that no part has been contributed by, produced by or in conjunction with another student.
5. I/we give permission for my assessment response to be reproduced, communicated compared and archived for the purposes of detecting plagiarism.
6. I/we give permission for a copy of my assessment to be retained by the university for review and comparison, including review by external examiners.
7. I/we understand that:
   - Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
   - Plagiarism includes the act of assisting or allowing another person to plagiarise or to copy my work.

I/we agree and acknowledge that:

1. I/we have read and understood the Declaration and Statement of Authorship above.
2. If I/we do not agree to the Declaration and Statement of Authorship in this context and a signature is not included below, the assessment outcome is not valid for assessment purposes and will not be included in my final result for this course.

Visit the student essentials **Academic Integrity** page for more information on academic integrity at RMIT, including your responsibilities, breaches and penalties for academic misconduct.

Student signature/s required on page 2.

Integrity | Assignment Cover Sheet for use when submitting work for assessment — 0984 1221 | 1 of 2

Assignment cover sheet

for use when submitting work for assessment

RMIT UNIVERSITY

Student signature/s

I/we declare that I/we have read and understood the declaration and statement of authorship.

1 MRandall — Digitally signed by Mark Randall Date: 2024.05.18 21:24:45 +10'00'

2

3

4

5

6

Further information relating to academic integrity breaches, and penalties that range from a notation on your student record to expulsion from the University, is contained in the **Academic Integrity Policy**, **Student Conduct Policy – Schedule 1**, and the **Student Conduct Regulations**.

Integrity | Assignment Cover Sheet for use when submitting work for assessment — 0984 1221 | 2 of 2

Student names, numbers and percentage of contributions

Table 1: Group information

Student name	Student number	Percentage of contribution
Mark Randall	s9001731	100

Library Load

Package 1 : RColorBrewer (Neuwirth 2022)
Package 2 : ggsci (Xiao 2024)
Package 3 : forecast (Hyndman et al. 2024)
Package 4 : MVN (Korkmaz, Goksuluk & Zararsiz 2021)
Package 5 : outliers (Komsta 2022)
Package 6 : egg (Auguie 2019)
Package 7 : gridExtra (Auguie 2017)
Package 8 : scales (Wickham, Pedersen & Seidel 2023)
Package 9 : geosphere (Hijmans 2022)
Package 10 : OpenStreetMap (Fellows & JMapViewer library by Jan Peter Stotz 2023)
Package 11 : tidyterra (Hernangómez 2024)
Package 12 : maptiles (Giraud 2024)
Package 13 : sf (Pebesma 2024)
Package 14 : sn (Azzalini 2023)
Package 15 : stats4 (R Core Team 2024a)
Package 16 : moments (Komsta & Novomestky 2022)
Package 17 : ggnewscale (Campitelli 2024)
Package 18 : Hmisc (Harrell Jr 2024)
Package 19 : validate (van der Loo & de Jonge 2024)
Package 20 : deducorrect (van der Loo, de Jonge & Scholtus 2015)
Package 21 : editrules (de Jonge & van der Loo 2024)
Package 22 : igraph (Csárdi et al. 2024)
Package 23 : deductive (van der Loo & de Jonge 2021)
Package 24 : tidyselect (Henry & Wickham 2024)
Package 25 : rvest (Wickham 2024)
Package 26 : here (Müller 2020)
Package 27 : glue (Hester & Bryan 2024)
Package 28 : magrittr (Bache & Wickham 2022)
Package 29 : lubridate (Spinu, Grolemund & Wickham 2023)
Package 30 : forcats (Wickham 2023a)
Package 31 : stringr (Wickham 2023b)
Package 32 : purrr (Wickham & Henry 2023)
Package 33 : tibble (Müller & Wickham 2023)
Package 34 : ggplot2 (Wickham et al. 2024)
Package 35 : tidyverse (Wickham 2023c)
Package 36 : kableExtra (Zhu 2024)
Package 37 : knitr (Xie 2024)
Package 38 : readxl (Wickham & Bryan 2023)
Package 39 : readr (Wickham, Hester & Bryan 2024)
Package 40 : dplyr (Wickham et al. 2023)
Package 41 : tidyr (Wickham, Vaughan & Girlich 2024)
Package 42 : openxlsx (Schauberger & Walker 2023)
Package 43 : stats (R Core Team 2024b)
Package 44 : graphics (R Core Team 2024c)
Package 45 : grDevices (R Core Team 2024d)
Package 46 : utils (R Core Team 2024e)
Package 47 : datasets (R Core Team 2024f)
Package 48 : methods (R Core Team 2024g)
Package 49 : base (R Core Team 2024h)

Abstract

"Most vehicular accidents in Victoria involve a male driver between 18 to 30 years of Age and a high powered car."

This project will use some empirical data collected by the Victorian State Government(Vic 2024) to examine this statement.

Executive Summary

Information that is absolutely needed is age and sex of the driver, power of the vehicle of all vehicles and drivers in an accident incident. The date, time, day, location of the accident and make of vehicle , if available, would assist in providing correlated causes.

At annex A is a preliminary exploration of the data files. Examining the tables results in the following:

- From Accident.csv get date, time, day.
- From Node.csv get latitude, longitude, and local government authority(LGA).
- From Person.csv get sex, age group and road user. Also foreign key of vehicle id
- From Vehicle.csv get year, make, model, and power.

These modified data frames are then scrutinised for:

- Missing values, NAN, values.
- Opportunities factors and ordered factors.
- Conversion to a time-stamp if applicable.
- Other anomalies.

After the initial inspection conduct a merge of the sub-data frames to one data frame and examine for "tidy" state. Then check to see if within ranges and final NA/NAN/INF and impute missing/out of range data..

Outliers were checked for certain attributes and dealt with.

Transformation of skewed attributes was then conducted.

Data

```
# Global Utility functions could be removed to source
# file.
checkNAInCols <- function(checkDF, checkDFVarNames) {
    chkedAllDFList <- list()
    for (posn in 1:length(checkDF)) {
        chkIndDFList <- list()
        checkCol <- colnames(checkDF[[posn]])
        count <- 1
        for (inputCol in checkCol) {
            sumNA <- checkDF[[1]][inputCol] %>%
                summarise(numNa = sum(is.na(.)))
            if (is.numeric(checkDF[[1]][inputCol][[1]]) |
                is.integer(checkDF[[1]][inputCol][[1]])) {
                sumNAN <- sum(is.nan(checkDF[[1]][[inputCol]]))
                sumINF <- sum(is.infinite(checkDF[[1]][[inputCol]]))
            } else {
                sumNAN <- 0
                sumINF <- 0
            }
            tmpSums <- list(sumNA = sumNA[[1]], sumNAN = sumNAN[[1]],
                sumINF = sumINF[[1]])
            tmpSumsByCol <- list(tmpColName = c(tmpSums))
            names(tmpSumsByCol) <- inputCol
            chkIndDFList <- append(chkIndDFList, tmpSumsByCol)
            count <- count + 1
        }
        tmpDFSUMCOLS <- list(tmpDFName = c(chkIndDFList))
        names(tmpDFSUMCOLS) <- checkDFVarNames[[posn]]
        chkedAllDFList <- append(chkedAllDFList, tmpDFSUMCOLS)
    }
    return(tmpDFSUMCOLS)
}

matrixNANANINF <- function(inputListsSums) {
    matrixDF <- matrix(NA, nrow = length(inputListsSums[[1]]),
        ncol = length(inputListsSums[[1]][[1]]))
    rownames(matrixDF) <- names(inputListsSums[[1]])
    colnames(matrixDF) <- names(inputListsSums[[1]][[1]])
    for (row in 1:length(inputListsSums[[1]])) {
        for (col in 1:length(inputListsSums[[1]][[1]])) {
            matrixDF[c(row), c(col)] <- inputListsSums[[1]][[row]][[col]]
        }
    }
    return(matrixDF)
}

dispHead <- function(chkAttrDf) {
    if (dim(chkAttrDf)[[1]] > 0) {
        res <- head(chkAttrDf, 5)
    } else {
        res <- cat(paste(deparse(substitute(chkAttrDf)),
            "is empty.\n"))
    }
    return(res)
}

lat <- c(-39.2, -34)
long <- c(140, 150)

VICLATLONG <- c(lat, long)
```

Accident.csv Data Frame

```
accDateTimeDay <- vicRoadsDFList$ACCIDENT.csv %>%
    select(ACCIDENT_NO, ACCIDENT_DATE, ACCIDENT_TIME, DAY_OF_WEEK,
        DAY_WEEK_DESC)
str(accDateTimeDay)
tibble [168,470 x 5] (S3: tbl_df/tbl/data.frame)
 $ ACCIDENT_NO  : chr [1:168470] "T20120000062" "T20120000063" "T20120000064" "T20120000065" ...
 $ ACCIDENT_DATE: Date[1:168470], format: "2012-01-01" "2012-01-01" ...
 $ ACCIDENT_TIME: 'hms' num [1:168470] 18:00:00 16:45:00 21:40:00 17:50:00 ...
  ..- attr(*, "units")= chr "secs"
 $ DAY_OF_WEEK  : num [1:168470] 1 1 1 1 1 1 1 1 1 1 ...
 $ DAY_WEEK_DESC: chr [1:168470] "Sunday" "Sunday" "Sunday" "Sunday" ...
```

```
# Utility Function hidden to save page space. Code
# available on request.
chkDF1 <- checkNAInCols(list(accDateTimeDay), c("accDateTimeDay"))
# Utility Function hidden to save page space. Code
# available on request.
matrixDF1 <- matrixNANANINF(chkDF1)
matrixDF1 %>%
    kable(caption = glue("Table of Sums of NA,NAN, and Inf for {names(chkDF1)}"),
        longtable = TRUE, format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 8)
```

Table 2: Table of Sums of NA,NAN, and Inf for accDateTimeDay

	sumNA	sumNAN	sumINF
ACCIDENT_NO	0	0	0
ACCIDENT_DATE	0	0	0
ACCIDENT_TIME	0	0	0
DAY_OF_WEEK	0	0	0
DAY_WEEK_DESC	0	0	0

```
dayLabelsLevels <- c("Monday", "Tuesday", "Wednesday", "Thursday",
    "Friday", "Saturday", "Sunday")
accDateTimeDay %<>%
    mutate(Timestamp = lubridate::ymd_hms(paste(ACCIDENT_DATE,
        ACCIDENT_TIME), tz = "Australia/Melbourne"), Full_Day = factor(DAY_WEEK_DESC,
        ordered = TRUE, levels = dayLabelsLevels, labels = dayLabelsLevels),
        Time_As_Float = round((as.numeric(format(Timestamp,
            "%H")) + (as.numeric(format(Timestamp, "%M")))/60),
            digits = 2)) %>%
    rename(Date_char = ACCIDENT_DATE, Time_char = ACCIDENT_TIME) %>%
    select(-c("DAY_OF_WEEK", "DAY_WEEK_DESC"))
str(accDateTimeDay)
tibble [168,470 x 6] (S3: tbl_df/tbl/data.frame)
 $ ACCIDENT_NO  : chr [1:168470] "T20120000062" "T20120000063" "T20120000064" "T20120000065" ...
 $ Date_char    : Date[1:168470], format: "2012-01-01" "2012-01-01" ...
 $ Time_char    : 'hms' num [1:168470] 18:00:00 16:45:00 21:40:00 17:50:00 ...
  ..- attr(*, "units")= chr "secs"
 $ Timestamp    : POSIXct[1:168470], format: "2012-01-01 18:00:00" "2012-01-01 16:45:00" ...
 $ Full_Day     : Ord.factor w/ 7 levels "Monday"<"Tuesday"<..: 7 7 7 7 7 7 7 7 7 7 ...
 $ Time_As_Float: num [1:168470] 18 16.8 21.7 17.8 20 ...
```

```
n_distinct(accDateTimeDay$ACCIDENT_NO)
[1] 168470
```

```
rm(matrixDF1, chkDF1)
```

The date, time and day observations are extracted from the ACCIDENT.csv file. There are 168470 priamry accident identifiers.

The table returned shows that there are NA, NAN, Inf values in the original data frame. Mutate without problem.

The date and time are character representations. Mutate to add a posix object with local time zone of Australia/Melbourne.

Mutate to add an float value of time.

Change `DAY_WEEK_DESC` to an ordered factored list. Make the start day of the order as Monday to align with Federal government standards. Reduce column names to a more human readable and manageable form.

Node.csv Data Frame

```
accNode <- vicRoadsDFList$NODE.csv %>%
    select(ACCIDENT_NO, LGA_NAME, LATITUDE, LONGITUDE)
str(accNode)
tibble [177,007 x 4] (S3: tbl_df/tbl/data.frame)
 $ ACCIDENT_NO: chr [1:177007] "T20220012597" "T20220019076" "T20220019043" "T20220019019" ...
 $ LGA_NAME   : chr [1:177007] "GEELONG" "HUME" "DANDENONG" "GLEN EIRA" ...
 $ LATITUDE   : num [1:177007] -38.1 -37.7 -38 -37.9 -37.7 ...
 $ LONGITUDE  : num [1:177007] 144 145 145 145 145 ...
```

```
# Utility Function hidden to save page space. Code
# available on request.
chkDF2 <- checkNAInCols(list(accNode), c("accNode"))
# Utility Function hidden to save page space. Code
# available on request.
matrixDF2 <- matrixNANANINF(chkDF2)
matrixDF2 %>%
    kable(caption = glue("Table of Sums of NA,NAN, and Inf for {names(chkDF2)}"),
        longtable = TRUE, format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 8)
```

Table 3: Table of Sums of NA,NAN, and Inf for accNode

	sumNA	sumNAN	sumINF
ACCIDENT_NO	0	0	0
LGA_NAME	37	0	0
LATITUDE	0	0	0
LONGITUDE	0	0	0

```
str(accNode)
tibble [177,007 x 4] (S3: tbl_df/tbl/data.frame)
 $ ACCIDENT_NO: chr [1:177007] "T20220012597" "T20220019076" "T20220019043" "T20220019019" ...
 $ LGA_NAME   : chr [1:177007] "GEELONG" "HUME" "DANDENONG" "GLEN EIRA" ...
 $ LATITUDE   : num [1:177007] -38.1 -37.7 -38 -37.9 -37.7 ...
 $ LONGITUDE  : num [1:177007] 144 145 145 145 145 ...
```

```
n_distinct(accNode$ACCIDENT_NO)
[1] 174242
```

```
rm(matrixDF2, chkDF2)
```

There was no mutating of this data frame. There are 172242 unique accident identifiers.

Person.csv Data Frame

```
accPerson <- vicRoadsDFList$PERSON.csv %>%
    select(ACCIDENT_NO, PERSON_ID, VEHICLE_ID, SEX, AGE_GROUP,
        ROAD_USER_TYPE_DESC)
str(accPerson)
tibble [393,661 x 6] (S3: tbl_df/tbl/data.frame)
$ ACCIDENT_NO : chr [1:393661] "T20220027723" "T20210018500" "T20210026317" "T20220000143" ...
$ PERSON_ID : chr [1:393661] "B" "A" "01" "A" ...
$ VEHICLE_ID : chr [1:393661] "B" "A" "B" "A" ...
$ SEX : chr [1:393661] "M" "M" "M" "F" ...
$ AGE_GROUP : chr [1:393661] "50-59" "30-39" "70+" "65-69" ...
$ ROAD_USER_TYPE_DESC: chr [1:393661] "Drivers" "Drivers" "Passengers" "Drivers" ...
```

```
# Utility Function hidden to save page space. Code
# available on request.
chkDF3 <- checkNAInCols(list(accPerson), c("accPerson"))
# Utility Function hidden to save page space. Code
# available on request.
matrixDF3 <- matrixNANANINF(chkDF3)
matrixDF3 %>%
    kable(caption = glue("Table of Sums of NA,NAN, and Inf for {names(chkDF3)}"),
        longtable = TRUE, format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 8)
```

Table 4: Table of Sums of NA,NAN, and Inf for accPerson

	sumNA	sumNAN	sumINF
ACCIDENT_NO	0	0	0
PERSON_ID	0	0	0
VEHICLE_ID	15850	0	0
SEX	31	0	0
AGE_GROUP	0	0	0
ROAD_USER_TYPE_DESC	0	0	0

```
str(accPerson)
tibble [393,661 x 6] (S3: tbl_df/tbl/data.frame)
$ ACCIDENT_NO : chr [1:393661] "T20220027723" "T20210018500" "T20210026317" "T20220000143" ...
$ PERSON_ID : chr [1:393661] "B" "A" "01" "A" ...
$ VEHICLE_ID : chr [1:393661] "B" "A" "B" "A" ...
$ SEX : chr [1:393661] "M" "M" "M" "F" ...
$ AGE_GROUP : chr [1:393661] "50-59" "30-39" "70+" "65-69" ...
$ ROAD_USER_TYPE_DESC: chr [1:393661] "Drivers" "Drivers" "Passengers" "Drivers" ...
```

```
n_distinct(accPerson$ACCIDENT_NO)
[1] 168470
```

```
unique(accPerson$AGE_GROUP)
 [1] "50-59"   "30-39"   "70+"     "65-69"   "40-49"   "26-29"   "18-21"
 [8] "22-25"   "60-64"   "Unknown" "5-12"    "13-15"   "16-17"   "0-4"
```

```
accPerson %<>%
    mutate(AGE_GROUP = factor(AGE_GROUP, ordered = TRUE,
        levels = c("0-4", "5-12", "13-15", "16-17", "18-21",
            "22-25", "26-29", "30-39", "40-49", "50-59",
            "60-64", "65-69", "70+", "Unknown"))) %>%
    filter(ROAD_USER_TYPE_DESC %in% c("Drivers", "Motorcyclists")) %>%
    group_by(by = ACCIDENT_NO) %>%
    pivot_wider(names_from = PERSON_ID, values_from = AGE_GROUP,
        names_prefix = "persId", names_sort = TRUE) %>%
    ungroup()
# Utility Function hidden to save page space. Code
# available on request.
chkDF4 <- checkNAInCols(list(accPerson), c("accPerson"))
# Utility Function hidden to save page space. Code
# available on request.
matrixDF4 <- abs(-t(matrixNANANINF(chkDF4)))
matrixDF4[, 1:10] %>%
    kable(caption = glue("Table of Sums of NA,NAN, and Inf for {names(chkDF4)} Wider cols 1 to 10 "),
        longtable = TRUE, format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 6)
```

Table 5: Table of Sums of NA,NAN, and Inf for accPerson Wider cols 1 to 10

	ACCIDENT_NO	VEHICLE_ID	SEX	ROAD_USER_TYPE_DESC	by	persId01	persId02	persId03	persId07	persIdA
sumNA	0	0	30	0	0	272991	273091	273108	273112	114843
sumNAN	0	0	0	0	0	0	0	0	0	0
sumINF	0	0	0	0	0	0	0	0	0	0

```
matrixDF4[, 11:20] %>%
    kable(caption = glue("Table of Sums of NA,NAN, and Inf for {names(chkDF4)} cols 11 to 20"),
        longtable = TRUE, format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 6)
```

Table 6: Table of Sums of NA,NAN, and Inf for accPerson cols 11 to 20

	persIdB	persIdC	persIdD	persIdE	persIdF	persIdG	persIdH	persIdI	persIdJ	persIdK
sumNA	180088	256967	269188	272076	272806	272998	273063	273086	273095	273101
sumNAN	0	0	0	0	0	0	0	0	0	0
sumINF	0	0	0	0	0	0	0	0	0	0

```
matrixDF4[, 21:30] %>%
    kable(caption = glue("Table of Sums of NA,NAN, and Inf for {names(chkDF4)} cols 21 to 30"),
        longtable = TRUE, format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 6)
```

Table 7: Table of Sums of NA,NAN, and Inf for accPerson cols 21 to 30

	persIdL	persIdM	persIdN	persIdO	persIdP	persIdQ	persIdR	persIdS	persIdT	persIdU
sumNA	273104	273107	273110	273111	273110	273111	273111	273111	273111	273112
sumNAN	0	0	0	0	0	0	0	0	0	0
sumINF	0	0	0	0	0	0	0	0	0	0

```
rm(matrixDF3, matrixDF4, chkDF4, chkDF3)
```

The initial data frame contained repetitive values for the person Id and the vehicle Id. This is suggestive that the data is stored in one table or a multitude of tables under column partition with the accident number as the primary key identifier. This limits the ability to interrogate the data for person and vehicles involved in many accidents.

The age group was set to an ordered factor. As per specification this data frame was set to "untidy" via a pivot wider. That is the observational row will contain multiple driver identifiers. The resultant NA/NAN/Inf interrogation table displays a large number of NA but not one column full of NA.

Vehicle.csv Data Frame

```
accVehicle <- vicRoadsDFList$VEHICLE.csv %>%
    select(ACCIDENT_NO, VEHICLE_ID, VEHICLE_YEAR_MANUF,
        VEHICLE_MAKE, VEHICLE_MODEL, VEHICLE_POWER, VEHICLE_TYPE_DESC,
        NO_OF_CYLINDERS) %>%
    mutate(NO_OF_CYLINDERS = as.integer(NO_OF_CYLINDERS),
        VEHICLE_YEAR_MANUF = as.integer(VEHICLE_YEAR_MANUF))
str(accVehicle)
tibble [307,149 x 8] (S3: tbl_df/tbl/data.frame)
$ ACCIDENT_NO : chr [1:307149] "T20120000009" "T20120000012" "T20120000012" "T20120000013" ...
$ VEHICLE_ID : chr [1:307149] "A" "A" "B" "A" ...
$ VEHICLE_YEAR_MANUF: int [1:307149] 1996 2002 1988 1997 2010 1983 1998 1996 2008 2006 ...
$ VEHICLE_MAKE : chr [1:307149] "HOLDEN" "HOLDEN" "TOYOTA" "MITSUB" ...
$ VEHICLE_MODEL : chr [1:307149] "ACCLAI" "MONARO" NA "MAGNA" ...
$ VEHICLE_POWER : logi [1:307149] NA NA NA NA NA NA ...
$ VEHICLE_TYPE_DESC : chr [1:307149] "Car" "Car" "Car" "Car" ...
$ NO_OF_CYLINDERS : int [1:307149] 6 8 4 4 6 4 4 8 6 6 ...
```

```
# Utility Function hidden to save page space. Code
# available on request.
chkDF5 <- checkNAInCols(list(accVehicle), c("accVehicle"))
# Utility Function hidden to save page space. Code
# available on request.
matrixDF5 <- matrixNANANINF(chkDF5)
matrixDF5 %>%
    kable(caption = glue("Table of Sums of NA,NAN, and Inf for {names(chkDF5)}"),
        longtable = TRUE, format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 8)
```

Table 8: Table of Sums of NA,NAN, and Inf for accVehicle

	sumNA	sumNAN	sumINF
ACCIDENT_NO	0	0	0
VEHICLE_ID	0	0	0
VEHICLE_YEAR_MANUF	6918	0	0
VEHICLE_MAKE	17076	0	0
VEHICLE_MODEL	28622	0	0
VEHICLE_POWER	307149	0	0
VEHICLE_TYPE_DESC	0	0	0
NO_OF_CYLINDERS	31488	0	0

```
vehMakes <- unique(accVehicle$VEHICLE_MAKE)
vehModel <- unique(accVehicle$VEHICLE_MODEL)
n_distinct(accVehicle$ACCIDENT_NO)
[1] 168469
```

```
accVehicle %<>%
    select(-(VEHICLE_POWER))
rm(matrixDF5, chkDF5, vehMakes, vehModel)
```

The power attribute is non-existent in all observations, the only method to find this would be by brute force. It has been decided that this attribute is dropped. To achieve the analysis it is noted that the number of cylinders is only missing in 10.25% and there is a correlation between cylinders and power.

The year and number of cylinders have been converted to an integer value. There are missing or NA values resultant from the parse. An imputation of cylinders and year can be done from manufacturer and model. However `vehMakes` and `vehModel` on this data suggests that these input fields were not scrutinised to a data dictionary.

Merge the Data Frames

Phase One Merge Person with Vehicle The `accPerson` data frame forms the basis of the left side of the join argument. Pre-filtering for drivers(motorcyclists) has already occurred. The data frame will then be returned to a "tidy" state. That is the observational row will now only contain one person.

As part of the "tidy" the vehicle , person and join by identifiers will be removed. They are no longer of benefit and could jeopardise any modelling.

```
accCombDF <- left_join(accPerson, accVehicle, by = c("ACCIDENT_NO",
    "VEHICLE_ID")) %>%
    pivot_longer(cols = c(6:30), names_to = "Dvr_Id", values_to = "Age_Group") %>%
    filter(!is.na(Age_Group)) %>%
    select(-c("VEHICLE_ID", "Dvr_Id", "by"))
str(accCombDF)
tibble [273,114 x 9] (S3: tbl_df/tbl/data.frame)
$ ACCIDENT_NO : chr [1:273114] "T20220027723" "T20210018500" "T20220000143" "T20170015017" ...
$ SEX : chr [1:273114] "M" "M" "F" "M" ...
$ ROAD_USER_TYPE_DESC: chr [1:273114] "Drivers" "Drivers" "Drivers" "Drivers" ...
$ VEHICLE_YEAR_MANUF : int [1:273114] 2012 0 2013 2008 2012 2019 2011 2010 2008 2002 ...
$ VEHICLE_MAKE : chr [1:273114] "FORD" "HYNDAI" "HYNDAI" "TOYOTA" ...
$ VEHICLE_MODEL : chr [1:273114] "FALCON" NA "IX35" "LAND06" ...
$ VEHICLE_TYPE_DESC : chr [1:273114] "Car" "Car" "Station Wagon" "Station Wagon" ...
$ NO_OF_CYLINDERS : int [1:273114] 6 NA 4 4 4 4 4 4 6 6 ...
$ Age_Group : Ord.factor w/ 14 levels "0-4"<"5-12"<"13-15"<..: 10 8 12 9 7 8 13 10 9 6 ...
```

Merge Phase One with Temporal and Location Data Frames Temporal join was conducted with first followed by location. The location when first attempted threw an error of many to many relationship. This was examined and duplicate rows were found. It may have been prudent to pass the primary data frames threw `distinct()`, however given the "disjointed untidy" recording of data it is discounted. As it was unclear from the source data whether an accident could have two locations. That is if a car accident occured at X and the driver left the scene and caused an accident at Y is that still recorded with the same accident number.

Duplicated observations caused by location and duplicated location attribute rows were discarded.

Column names moved to a shorter more human readable form.

```
accCombDF <- left_join(accCombDF, accDateTimeDay, by = c("ACCIDENT_NO"))
accCombDF <- left_join(accCombDF, accNode, by = c("ACCIDENT_NO"),
    relationship = "many-to-many") %>%
    distinct()
colnames(accCombDF) <- c("Id", "Gender", "Dvr/Cyclist",
```

```
    "Veh_Year", "Veh_Make", "Veh_Model", "Veh_Type", "Cylinders",
    "Age_Group", "Date_char", "Time_char", "Timestamp",
    "Full_Day", "Time_Float", "LGA_Name", "Latitude", "Longitude")

str(accCombDF)
tibble [273,069 x 17] (S3: tbl_df/tbl/data.frame)
 $ Id          : chr [1:273069] "T20220027723" "T20210018500" "T20220000143" "T20170015017" ...
 $ Gender      : chr [1:273069] "M" "M" "F" "M" ...
 $ Dvr/Cyclist: chr [1:273069] "Drivers" "Drivers" "Drivers" "Drivers" ...
 $ Veh_Year    : int [1:273069] 2012 0 2013 2008 2012 2019 2011 2010 2008 2002 ...
 $ Veh_Make    : chr [1:273069] "FORD" "HYNDAI" "HYNDAI" "TOYOTA" ...
 $ Veh_Model   : chr [1:273069] "FALCON" NA "IX35" "LAND06" ...
 $ Veh_Type    : chr [1:273069] "Car" "Car" "Station Wagon" "Station Wagon" ...
 $ Cylinders   : int [1:273069] 6 NA 4 4 4 4 4 4 6 6 ...
 $ Age_Group   : Ord.factor w/ 14 levels "0-4"<"5-12"<"13-15"<..: 10 8 12 9 7 8 13 10 9 6 ...
 $ Date_char   : Date[1:273069], format: "2022-12-06" "2021-08-28" ...
 $ Time_char   : 'hms' num [1:273069] 16:32:00 13:20:00 19:50:00 14:40:00 ...
  ..- attr(*, "units")= chr "secs"
 $ Timestamp   : POSIXct[1:273069], format: "2022-12-06 16:32:00" "2021-08-28 13:20:00" ...
 $ Full_Day    : Ord.factor w/ 7 levels "Monday"<"Tuesday"<..: 2 6 1 4 5 3 4 4 7 7 ...
 $ Time_Float  : num [1:273069] 16.5 13.3 19.8 14.7 10.8 ...
 $ LGA_Name    : chr [1:273069] "GEELONG" "BRIMBANK" "BALLARAT" "MILDURA" ...
 $ Latitude    : num [1:273069] -38.2 -37.8 -37.6 -34.2 -38.4 ...
 $ Longitude   : num [1:273069] 144 145 144 142 145 ...
```

Add a further Correlated Field of Distance from a Point The distance from the Melbourne GPO will be added. Here an input could be established to determine accidents within a known black spot. The return from the check gave that all values longitude and latitude where within the earth spherical coordinate system, however NA were not examined. To add distances NA values filtered out and distance calculated via "geosphere" package.

```
distToPointAcc <- function(fromLongLat, toLongLat) {

    calcDist <- geosphere::distVincentyEllipsoid(fromLongLat,
        toLongLat)
    return(calcDist)
}
# First Check If Long and Lat in -180 to 180, -90 to
# 90

chkLongLatEarth <- accCombDF %>%
    filter((Longitude <= -180 | Longitude >= 180) | (Latitude <=
        -90 | Latitude >= 90))
dispHead(chkLongLatEarth)
chkLongLatEarth is empty.
NULL
```

```
gpoMelbLatLong <- c(144.9627, -37.125)

accCombDF %<>%
    mutate(Dist_GPO = case_when(!((is.na(Longitude) | is.na(Latitude))) ~
        mapply(function(inLong, inLat) distToPointAcc(gpoMelbLatLong,
            c(inLong, inLat)), Longitude, Latitude)))
n_distinct(accCombDF$Id)
[1] 163048
```

```
# Utility Function hidden to save page space. Code
# available on request.
chkDF6 <- checkNAInCols(list(accCombDF), c("accCombDF"))
# Utility Function hidden to save page space. Code
# available on request.
matrixDF6 <- abs(-t(matrixNANANINF(chkDF6)))
matrixDF6[, 1:9] %>%
    kable(caption = glue("Table of Sums of NA,NAN, and Inf for {names(chkDF6)} cols 1 to 9"),
        longtable = TRUE, format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 6)
```

Table 9: Table of Sums of NA,NAN, and Inf for accCombDF cols 1 to 9

	Id	Gender	Dvr/Cyclist	Veh_Year	Veh_Make	Veh_Model	Veh_Type	Cylinders	Age_Group
sumNA	0	30	0	1764	26	6029	0	7462	0
sumNAN	0	0	0	0	0	0	0	0	0
sumINF	0	0	0	0	0	0	0	0	0

```
matrixDF6[, 10:17] %>%
    kable(caption = glue("Table of Sums of NA,NAN, and Inf for {names(chkDF6)} cols 10 to 18"),
        longtable = TRUE, format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 6)
```

Table 10: Table of Sums of NA,NAN, and Inf for accCombDF cols 10 to 18

	Date_char	Time_char	Timestamp	Full_Day	Time_Float	LGA_Name	Latitude	Longitude
sumNA	0	0	0	0	0	197	146	146
sumNAN	0	0	0	0	0	0	0	0
sumINF	0	0	0	0	0	0	0	0

```
missId <- setdiff(unique(accPerson$ACCIDENT_NO), unique(accNode$ACCIDENT_NO))
numMissing <- length(missId)

missingAccNo <- accCombDF %>%
    filter(Id %in% missId)
rm(matrixDF6, accDateTimeDay, accNode, accPerson, accVehicle,
    chkDF6)
```

The NA values in cylinders and year are accounted for by missing data in the `accVehicle` data frame. Imputation by brute force may add values; and goes beyond the time frame of this report.

The merged data frame,`accCombDF`, displays 146 observations without location, yet the `accNode` data frame displays

all observations with values. By executing `setdiff(unique(accPerson$ACCIDENT_NO), unique(accNode$ACCIDENT_NO))` and then conducting `length()` there are 103 values of `ACCIDENT_NO` that are different. Sub-setting those observations to `missingAccNo` allows an examination that concludes that the information to conduct the analysis is there; however any geo-positional plotting or analysis will require that these observations are removed.

Data Dictionary Merged Data Sets

```
accDFAttr <- colnames(accCombDF)
accDFType <- c("Character", "Character", "Character", "Integer",
    "Character", "Character", "Character", "Integer", "Ordered Factor Character",
    "Date Format Y-m-d", "hms num", "POSIXct", "Ordered Factor Character",
    "Double", "Character", "Double", "Double", "Double")
accDFDesc <- c("Unique Accident Identifier Starts with T followed by Year (T2024...).",
    "Gender of driver in accident.  Range M or F", "Whether Drivers or Motorcyclists.",
    "Year of Vehicle Manufacture.  Range 1908 to present",
    "Vehicle Manufacturer.", "Vehicle Model.", "Vehicle Body Type.",
    "Numbers of Cylinders of the vehicle. Range 1 to 12 (Special case for train, tram, horse, trailers = 0)",
    "Age Group of the driver.", "Date of Accident. Range 2012-01-01 to present",
    "Time of Accident. Range 0 to 12", "Timestamp. Range 2012-01-01 00:00:00.00 to present",
    "Day of Week. range Monday to Sunday", "Decimal time of Accident. Range 0 to 24",
    "Name of Local Government Authority.  See Victorian Government Source",
    "Latitude. Range -34 to -39.2", "Longitude. Range 140 to 150",
    "Distance Meters to Melb GPO. Greater or equal to 0")
accDFDataDict <- data.frame(cbind(accDFAttr, accDFType,
    accDFDesc))
colnames(accDFDataDict) <- c("Attribute", "Data Type", "Description")


accDFDataDict %>%
    kbl(caption = "Data Dictionary Victorian Road Accidents Distilled") %>%
    kable_styling(font_size = 8) %>%
    column_spec(1, width = "2.5cm") %>%
    column_spec(2, width = "2cm") %>%
    column_spec(3, width = "10cm") %>%
    kable_classic(full_width = F)
```

```
rm(accDFAttr, accDFType, accDFDesc, accDFDataDict)
```

Table 11: Data Dictionary Victorian Road Accidents Distilled

Attribute	Data Type	Description
Id	Character	Unique Accident Identifier Starts with T followed by Year (T2024...).
Gender	Character	Gender of driver in accident. Range M or F
Dvr/Cyclist	Character	Whether Drivers or Motorcyclists.
Veh_Year	Integer	Year of Vehicle Manufacture. Range 1908 to present
Veh_Make	Character	Vehicle Manufacturer.
Veh_Model	Character	Vehicle Model.
Veh_Type	Character	Vehicle Body Type.
Cylinders	Integer	Numbers of Cylinders of the vehicle. Range 1 to 12 (Special case for train, tram, horse, trailers = 0)
Age_Group	Ordered Factor Character	Age Group of the driver.
Date_char	Date Format Y-m-d	Date of Accident. Range 2012-01-01 to present
Time_char	hms num	Time of Accident. Range 0 to 12
Timestamp	POSIXct	Timestamp. Range 2012-01-01 00:00:00.00 to present
Full_Day	Ordered Factor Character	Day of Week. range Monday to Sunday
Time_Float	Double	Decimal time of Accident. Range 0 to 24
LGA_Name	Character	Name of Local Government Authority. See Victorian Government Source
Latitude	Double	Latitude. Range -34 to -39.2
Longitude	Double	Longitude. Range 140 to 150
Dist_GPO	Double	Distance Meters to Melb GPO. Greater or equal to 0

Check Ranges of Distilled Data Observations and Impute

The following checks were done on the attributes of the data frame.

```
chkId <- accCombDF %>%
    filter(!str_detect(accCombDF$Id, "^T2"))
dispHead(chkId)
chkId is empty.
NULL
```

```
chkGender <- accCombDF %>%
    filter(!accCombDF$Gender %in% c("M", "F"))
dispHead(chkGender)  # 1.93 percent so deleted
# A tibble: 5 x 18
  Id         Gender `Dvr/Cyclist` Veh_Year Veh_Make Veh_Model Veh_Type Cylinders
  <chr>      <chr>  <chr>            <int> <chr>    <chr>     <chr>        <int>
1 T20220006- U      Drivers              0 FORD     XR6       Car             NA
2 T20210016- U      Drivers           2014 HOLDEN   CAPTIV    Station-         4
3 T20200023- U      Drivers           2019 HYNDAI   SANTA     Station-         4
4 T20220026- U      Drivers              0 UNKN     <NA>      Car             NA
5 T20230013- U      Drivers              0 UNKN     UTE       Utility         NA
# i 10 more variables: Age_Group <ord>, Date_char <date>, Time_char <time>,
#   Timestamp <dttm>, Full_Day <ord>, Time_Float <dbl>, LGA_Name <chr>,
#   Latitude <dbl>, Longitude <dbl>, Dist_GPO <dbl>
```

```
accCombDF %<>%
    filter(accCombDF$Gender %in% c("M", "F"))
chkDvr <- accCombDF %>%
    filter(!`Dvr/Cyclist` %in% c("Drivers", "Motorcyclists"))
dispHead(chkDvr)
chkDvr is empty.
NULL
```

```
chkVehYear <- accCombDF %>%
    filter(!(between(Veh_Year, 1908, as.numeric(format(Sys.Date(),
        "%Y")))))

dispHead(chkVehYear)  # Replace with the mean easy to impute. But if time brute force.
# A tibble: 5 x 18
  Id         Gender `Dvr/Cyclist` Veh_Year Veh_Make Veh_Model Veh_Type Cylinders
  <chr>      <chr>  <chr>            <int> <chr>    <chr>     <chr>        <int>
1 T20210018- M      Drivers              0 HYNDAI   <NA>      Car             NA
2 T20220001- M      Motorcyclists        0 K T M    SX        Motor C-        NA
3 T20230003- M      Motorcyclists        0 HUSQVA   701 SE    Motor C-         1
4 T20230006- M      Motorcyclists        0 B.M.W.   K50       Motor C-         2
5 T20230003- M      Motorcyclists        0 YAMAHA   FZ6-S     Motor C-         4
```

```
# i 10 more variables: Age_Group <ord>, Date_char <date>, Time_char <time>,
#   Timestamp <dttm>, Full_Day <ord>, Time_Float <dbl>, LGA_Name <chr>,
#   Latitude <dbl>, Longitude <dbl>, Dist_GPO <dbl>
```

```
vehYearMean <- round(mean(accCombDF$Veh_Year, na.rm = TRUE),
    digits = 0)
accCombDF %<>%
    mutate(Veh_Year = case_when(!(between(Veh_Year, 1908,
        as.numeric(format(Sys.Date(), "%Y")))) ~ vehYearMean,
        TRUE ~ Veh_Year))
chkCyl1 <- accCombDF %>%
    filter(!(Cylinders >= 1 & Cylinders <= 12))
dispHead(chkCyl1)  # Best done by brute force intuition.
# A tibble: 5 x 18
  Id          Gender `Dvr/Cyclist` Veh_Year Veh_Make Veh_Model Veh_Type Cylinders
  <chr>       <chr>  <chr>            <dbl> <chr>    <chr>     <chr>        <int>
1 T20120017~ M      Drivers           1998 MITSUB   TRITON    Utility         66
2 T20120019~ M      Drivers           1985 FORD     COURIE    Utility         64
3 T20130012~ M      Drivers           1993 TOYOTA   HILUX     Utility         64
4 T20130013~ M      Drivers           2008 FORD     FALCON    Utility         63
5 T20140003~ M      Drivers           2004 HOLDEN   COMMOD    Utility         64
# i 10 more variables: Age_Group <ord>, Date_char <date>, Time_char <time>,
#   Timestamp <dttm>, Full_Day <ord>, Time_Float <dbl>, LGA_Name <chr>,
#   Latitude <dbl>, Longitude <dbl>, Dist_GPO <dbl>
```

```
accCombDF %<>%
    mutate(Cylinders = case_when((Cylinders >= 60 & Cylinders <
        70) ~ 6, (Cylinders == 88) ~ 8, (Id == "T20140005298" |
        Id == "T20210010866") ~ 4, (Id == "T20140015002" |
        Id == "T20180015095" | Id == "T20190004933" | Id ==
        "T20230015484") ~ 6, (Id == "T20170016461") ~ 4,
        (Id == "T20150002025" | Id == "T20160020971" | Id ==
            "T20180001536" | Id == "T20230019609") ~ 2,
        TRUE ~ Cylinders))
chkCyl2 <- accCombDF %>%
    filter(is.na(Cylinders))
dispHead(chkCyl2)
# A tibble: 5 x 18
  Id          Gender `Dvr/Cyclist` Veh_Year Veh_Make Veh_Model Veh_Type Cylinders
  <chr>       <chr>  <chr>            <dbl> <chr>    <chr>     <chr>        <dbl>
1 T20210018~ M      Drivers           1971 HYNDAI   <NA>      Car             NA
2 T20210023~ M      Drivers           1990 FORD     <NA>      Car             NA
3 T20220001~ M      Motorcyclists     1971 K T M    SX        Motor C~        NA
4 T20220008~ M      Drivers           2018 BARKER   TRIAXL    Prime M~        NA
5 T20220023~ M      Motorcyclists     2016 YAMAHA   R3        Motor C~        NA
# i 10 more variables: Age_Group <ord>, Date_char <date>, Time_char <time>,
#   Timestamp <dttm>, Full_Day <ord>, Time_Float <dbl>, LGA_Name <chr>,
#   Latitude <dbl>, Longitude <dbl>, Dist_GPO <dbl>
```

```
# unique(chkCyl2$Veh_Type)
accCombDF %<>%
    mutate(Cylinders = case_when((str_detect(Veh_Type, "Prime|Heavy|Truck") &
        is.na(Cylinders)) ~ 8, str_detect(Veh_Type, "Train|Tram|Horse|Parked") ~
        0, (str_detect(Veh_Type, "Car|Taxi|Light") & is.na(Cylinders)) ~
        4, (str_detect(Veh_Type, "Panel|Station|Utility|Other") &
        is.na(Cylinders)) ~ 6, (str_detect(Veh_Type, "Bus|Plant") &
        is.na(Cylinders)) ~ 6, (str_detect(Veh_Type, "Moped|Cycle|Scooter|Quad") &
        is.na(Cylinders)) ~ 2, (str_detect(Veh_Type, "Car|Not|") &
        is.na(Cylinders)) ~ 4, TRUE ~ Cylinders))
# Age Group showing 0 NA not checked. Would show NA as
# factor.  Next temporal attributes show no na, nan if
# check for range
chkDateChar <- accCombDF %>%
    filter(!(between(Date_char, ymd("2012-01-01"), now())))
dispHead(chkDateChar)
chkDateChar is empty.
NULL
```

```
ChkTimeChar <- accCombDF %>%
    filter(!(between(hour(Time_char), 0, 24)))
dispHead(ChkTimeChar)
ChkTimeChar is empty.
NULL
```

```
# Timestamp created as local Melbourne
chkTimestamp <- accCombDF %>%
    filter(!(between(Timestamp, ymd_hms("2011-12-31 24:00:0",
        tz = "Australia/Melbourne"), now())))
dispHead(chkTimestamp)
chkTimestamp is empty.
NULL
```

```
chkDay <- accCombDF %>%
    filter(!(Full_Day %in% dayLabelsLevels))
dispHead((chkDay))
(chkDay) is empty.
NULL
```

```
chkTMFLoat <- accCombDF %>%
    filter(!(between(Time_Float, 0, 24)))
dispHead((chkTMFLoat))
(chkTMFLoat) is empty.
NULL
```

```
chkLGAName <- accCombDF %>%
    filter(!(LGA_Name %in% str_to_upper(namesLGAVic)))
dispHead((chkLGAName))
# A tibble: 5 x 18
  Id         Gender `Dvr/Cyclist` Veh_Year Veh_Make Veh_Model Veh_Type Cylinders
  <chr>      <chr>  <chr>            <dbl> <chr>    <chr>     <chr>        <dbl>
1 T20220027~ M      Drivers           2012 FORD     FALCON    Car              6
2 T20210017~ M      Motorcyclists     2019 SUZUKI   DRZ4SM    Motor C~         1
3 T20220016~ M      Motorcyclists     2017 YAMAHA   WR450F    Motor C~         1
4 T20220001~ M      Motorcyclists     1971 K T M    SX        Motor C~         2
5 T20210016~ F      Drivers           2003 MAZDA    3MAXX     Car              4
# i 10 more variables: Age_Group <ord>, Date_char <date>, Time_char <time>,
#   Timestamp <dttm>, Full_Day <ord>, Time_Float <dbl>, LGA_Name <chr>,
#   Latitude <dbl>, Longitude <dbl>, Dist_GPO <dbl>
```

```
unique(chkLGAName$LGA_Name)
 [1] "GEELONG"                   "BENDIGO"
 [3] "DANDENONG"                 NA
 [5] "SHEPPARTON"                "(MOUNT HOTHAM)"
 [7] "(FRENCH ISLAND)"           "(FALLS CREEK)"
 [9] "(MOUNT BULLER)"            "(MOUNT BAW BAW)"
[11] "(LAKE MOUNTAIN)"           "MOUNT BULLER ALPINE RESOR"
[13] "(MOUNT STIRLING)"
```

```
apineLGA <- c("(MOUNT HOTHAM)", "(FALLS CREEK)", "(MOUNT BAW BAW)")
mansLGA <- c("MOUNT BULLER ALPINE RESOR", "(MOUNT BULLER)",
    "(MOUNT STIRLING)")
accCombDF %<>%
    mutate(LGA_Name = case_when(LGA_Name == "GEELONG" ~
        "GREATER GEELONG", LGA_Name == "BENDIGO" ~ "GREATER BENDIGO",
        LGA_Name == "DANDENONG" ~ "GREATER DANDENONG", LGA_Name ==
            "SHEPPARTON" ~ "GREATER SHEPPARTON", LGA_Name %in%
            apineLGA ~ "ALPINE", LGA_Name %in% mansLGA ~
            "MANSFIELD", LGA_Name == "(LAKE MOUNTAIN)" ~
            "MURRINDINDI", LGA_Name == "(MOUNT BAW BAW)" ~
            "BAW BAW", LGA_Name == "(FRENCH ISLAND)" ~ "UNINCORPORATED VIC",
        TRUE ~ LGA_Name))
chkLat <- accCombDF %>%
    filter(is.na(Latitude) | !(between(Latitude, VICLATLONG[[1]],
        VICLATLONG[[2]])))
dispHead((chkLat))
# A tibble: 5 x 18
  Id         Gender `Dvr/Cyclist` Veh_Year Veh_Make Veh_Model Veh_Type Cylinders
  <chr>      <chr>  <chr>            <dbl> <chr>    <chr>     <chr>        <dbl>
1 T20210012~ M      Drivers           2004 KIA      CERATO    Car              4
2 T20220010~ M      Drivers           2010 TOYOTA   COROLL    Car              4
3 T20210012~ F      Drivers           2019 MITSUB   OUTLAN    Station~         4
4 T20210015~ M      Drivers           2014 NISSAN   PULSAR    Car              4
5 T20200021~ M      Motorcyclists     2017 HSQVRN   FE501     Motor C~         1
# i 10 more variables: Age_Group <ord>, Date_char <date>, Time_char <time>,
#   Timestamp <dttm>, Full_Day <ord>, Time_Float <dbl>, LGA_Name <chr>,
#   Latitude <dbl>, Longitude <dbl>, Dist_GPO <dbl>
```

```
chkLong <- accCombDF %>%
    filter(is.na(Longitude) | !(between(Longitude, VICLATLONG[[3]],
        VICLATLONG[[4]])))
dispHead((chkLong))
# A tibble: 5 x 18
  Id         Gender `Dvr/Cyclist` Veh_Year Veh_Make Veh_Model Veh_Type Cylinders
  <chr>      <chr>  <chr>            <dbl> <chr>    <chr>     <chr>        <dbl>
1 T20210012~ M      Drivers           2004 KIA      CERATO    Car              4
2 T20220010~ M      Drivers           2010 TOYOTA   COROLL    Car              4
3 T20210012~ F      Drivers           2019 MITSUB   OUTLAN    Station~         4
4 T20210015~ M      Drivers           2014 NISSAN   PULSAR    Car              4
5 T20200021~ M      Motorcyclists     2017 HSQVRN   FE501     Motor C~         1
# i 10 more variables: Age_Group <ord>, Date_char <date>, Time_char <time>,
#   Timestamp <dttm>, Full_Day <ord>, Time_Float <dbl>, LGA_Name <chr>,
#   Latitude <dbl>, Longitude <dbl>, Dist_GPO <dbl>
```

```
rm(chkCyl1, chkCyl2, chkDateChar, chkDay, chkDvr, chkGender,
    chkId, chkLat, chkLong, chkLongLatEarth, chkLGAName,
    chkTimestamp, chkTMFLoat, ChkTimeChar, chkVehYear)
```

The following was conducted on the data frame:

- Missing gender omitted as 1.93% of observations.Difficult to impute.

- Year of Manufacture of vehicle set to mean. Trivial but a lot quicker than brute force.

- Cylinders added by intuition of examination of observations, brute force. For those outside range. For those NA an intuitive type approach was taken.

- LGA Names were given the correct LGA nomenclature , imputed from location or left NA. Those left NA were for geo-spatial omission.

- Latitude and Longitude were within range except for NA which were again left NA for geo-spatial omission.

Check Outliers

Outliers will be checked but not omitted. The reason for this is because the observations are what I classify as contained. That is ranges have been checked and deleting any outlier would negate the veracity of the analysis. The meaningful attributes would be cylinders and range. Most other numerical values are temporal. `Time_Float` is numerical but cyclic, ie. 23:99 is only 0.02 away from 00.01 on consecutive days, or 23.98. Similarly with days as they are cyclic.

```
basePlot <- ggplot(accCombDF) + theme(axis.title.y = element_text(angle = 0,
    vjust = 0.5, size = 6)) + theme(axis.title.x = element_text(angle = 0,
    vjust = 0.5, size = 6)) + theme(plot.title = element_text(size = 6)) +
    theme(axis.text.y = element_text(face = "bold", colour = "blue",
        angle = 0, size = 6)) + theme(axis.text.x = element_text(face = "bold",
    colour = "blue", angle = 45, size = 6))

barAgeGp <- basePlot + geom_bar(aes(x = Age_Group)) + ggtitle("Bar Plot Age Group")
barAgeGp2 <- accCombDF %>%
    filter(Age_Group %in% c("0-4", "5-12", "13-15", "16-17")) %>%
    ggplot() + geom_bar(aes(x = Age_Group)) + ggtitle("Bar Plot Age \nGroup <=17") +
    theme(axis.title.y = element_text(angle = 0, vjust = 0.5,
        size = 6)) + theme(axis.title.x = element_text(angle = 0,
    vjust = 0.5, size = 6)) + theme(plot.title = element_text(size = 6)) +
    theme(axis.text.y = element_text(face = "bold", colour = "blue",
        angle = 90, size = 6)) + theme(axis.text.x = element_text(face = "bold",
    colour = "blue", angle = 45, size = 6))
histTime <- basePlot + ggtitle("Histogram Time of day") +
    geom_histogram(aes(x = Time_Float))
histDist <- ggplot(accCombDF, aes(x = Dist_GPO, y = after_stat(density))) +
    geom_histogram(binwidth = 10) + geom_density(colour = "green",
    linewidth = 0.3) + ggtitle("Histogram Distance to GPO") +
    scale_x_continuous(name = "Distance Kms", labels = function(x) x/1000) +
    theme(axis.title.y = element_text(angle = 0, vjust = 0.5,
        size = 6)) + theme(axis.title.x = element_text(angle = 0,
    vjust = 0.5, size = 6)) + theme(plot.title = element_text(size = 6)) +
    theme(axis.text.y = element_text(face = "bold", colour = "blue",
        angle = 90, size = 6)) + theme(axis.text.x = element_text(face = "bold",
    colour = "blue", angle = 45, size = 6))
histCyl <- basePlot + ggtitle("Histogram Cylinders") + geom_histogram(aes(x = Cylinders))
outDist <- basePlot + geom_boxplot(aes(y = Dist_GPO)) +
    ggtitle("Box Plot Distance to GPO") + scale_y_continuous(name = "Distance Kms",
    labels = function(x) x/1000)
outDist2 <- basePlot + geom_boxplot(aes(x = Age_Group, y = Dist_GPO,
    colour = Gender)) + ggtitle("Box Plot Age/ Distance to GPO") +
    theme(axis.text.y = element_text(face = "bold", colour = "blue",
        angle = 45)) + scale_y_continuous(name = "Distance Kms",
    labels = function(x) x/1000)
figure1 <- grid.arrange(outDist2, barAgeGp, barAgeGp2, histTime,
    histCyl, histDist, outDist, nrow = 6, ncol = 3, layout_matrix = rbind(rbind(c(1,
        1, 1), c(1, 1, 1)), rbind(c(2, 3, 4), c(2, 3, 4)),
        rbind(c(5, 6, 7), c(5, 6, 7))))
```

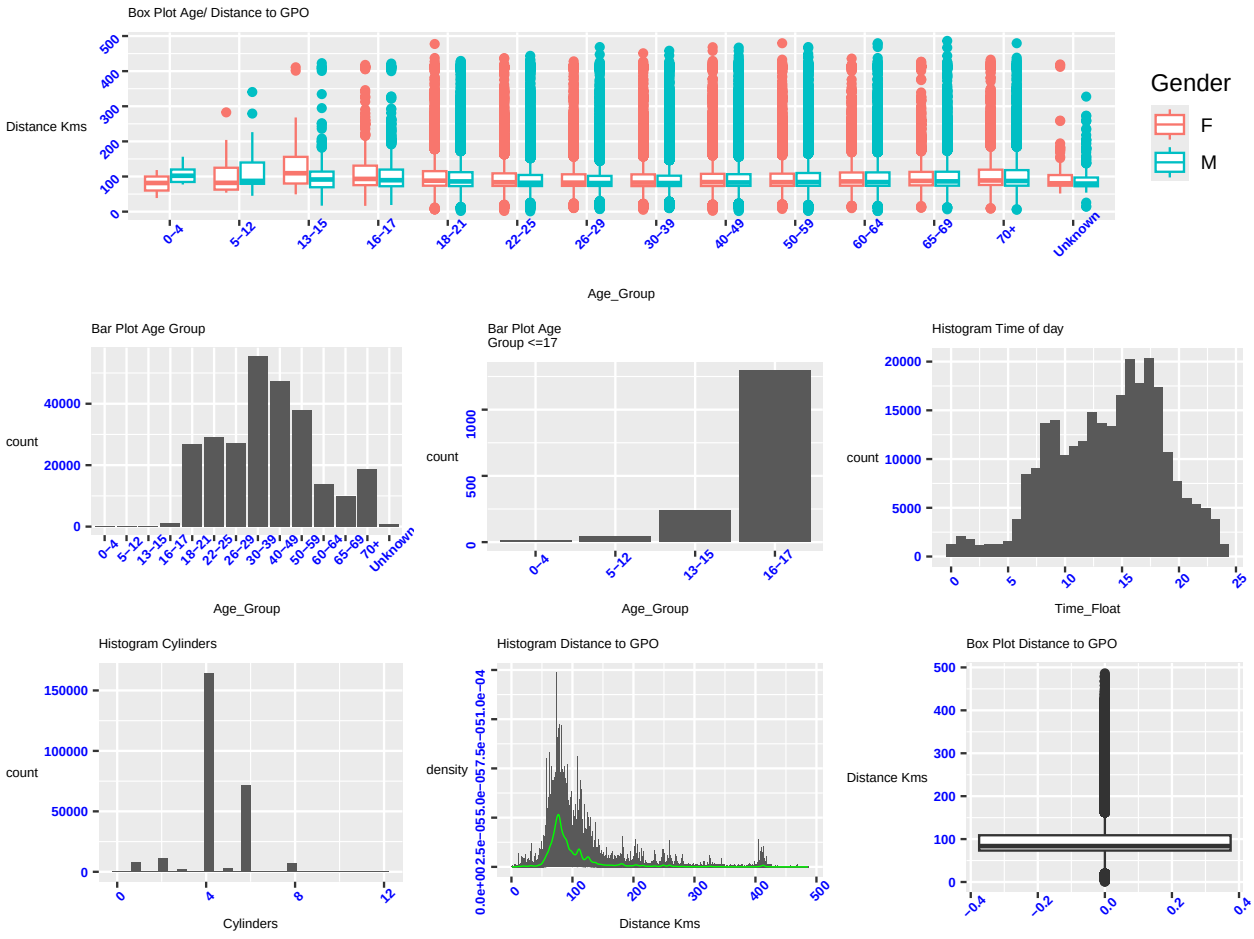


Figure 1: Plots showing distribution of attributes Of the DataFrame

The second bar plot of the age group is given as it shows observations in the 0-4 and 5-12 grouping. These values are not omitted as there is a plausible case for there existence. That is a child may have been left alone in a vehicle and caused the accident.

Most attributes/observations in this data frame could be converted to a categorical type vector; except for distance from the GPO. And as can be seen from the plot it is heavily right skewed. If we wanted to use this in an analysis the smaller represented higher numbers would impact greatly on any out come. The next section will deal with attempting to regain a normal distribution for the attribute.

The following code `maleAcc <- accCombDF %>% filter(Gender == "M" & !(is.na(Dist_GPO))) %>% select(Time_Float,Dist_GPO) results <- mvn(data = maleAcc, multivariateOutlierMetho = "quan", showOutliers = TRUE)` was run but in a windows system a vector of more than 2 Gb causes problems. Investigation to remedial action is longer than time frame to submit report.

Transform

```
baseDistGPODF <- accCombDF %>%
    filter(!(is.na(Dist_GPO))) %>%
    select(Dist_GPO)
sqrtDist <- sqrt(baseDistGPODF)
cubertDist <- baseDistGPODF^(1/3)
recipDist <- 1/baseDistGPODF
bCDist <- BoxCox(baseDistGPODF, lambda = "auto")
log10Dist <- log10(baseDistGPODF)
distZScores <- sqrtDist %>%
    select(Dist_GPO) %>%
    outliers::scores(type = c("z"))
distZScores1 <- sqrtDist %>%
    select(Dist_GPO) %>%
    outliers::scores(type = c("z"))
distZScores2 <- cubertDist %>%
    select(Dist_GPO) %>%
    outliers::scores(type = c("z"))
distZScores3 <- recipDist %>%
    select(Dist_GPO) %>%
    outliers::scores(type = c("z"))
distZScores4 <- bCDist %>%
    select(Dist_GPO) %>%
    outliers::scores(type = c("z"))
distZScores5 <- log10Dist %>%
    select(Dist_GPO) %>%
    outliers::scores(type = c("z"))
sumMatrix <- as.matrix(cbind(summary(distZScores1), summary(distZScores2),
    summary(distZScores3), summary(distZScores4), summary(distZScores5),
    summary(distZScores)))
colnames(sumMatrix) <- c("Sqare Root", "Cube Root", "Reciprocal",
    "BoxCox", "log10", "Original")
rownames(sumMatrix) <- NULL
kable(sumMatrix, caption = "Summary of Z Scores")
```

Table 12: Summary of Z Scores

Sqare Root	Cube Root	Reciprocal	BoxCox	log10	0riginal
Min. :-4.1273	Min. :-5.6489	Min. : -1.49599	Min. :-1.9220	Min. :-11.8368	Min. :-4.1273
1st Qu.:-0.5495	1st Qu.:-0.5546	1st Qu.: -0.41047	1st Qu.:-0.5081	1st Qu.: -0.5460	1st Qu.:-0.5495
Median :-0.2762	Median :-0.2587	Median : 0.01293	Median :-0.3064	Median : -0.2122	Median :-0.2762
Mean : 0.0000	Mean : 0.0000	Mean : 0.00000	Mean : 0.0000	Mean : 0.0000	Mean : 0.0000
3rd Qu.: 0.3290	3rd Qu.: 0.3750	3rd Qu.: 0.26994	3rd Qu.: 0.1876	3rd Qu.: 0.4569	3rd Qu.: 0.3290
Max. : 5.7656	Max. : 5.2302	Max. :178.37813	Max. : 7.5498	Max. : 4.2418	Max. : 5.7656

```
meanAccDistTrans <- c(mean(sqrtDist$Dist_GPO, na.rm = TRUE),
    mean(cubertDist$Dist_GPO, na.rm = TRUE), mean(recipDist$Dist_GPO,
        na.rm = TRUE), mean(bCDist$Dist_GPO, na.rm = TRUE),
    mean(log10Dist$Dist_GPO, na.rm = TRUE), mean(baseDistGPODF$Dist_GPO,
        na.rm = TRUE))
medianAccDistTrans <- c(stats::median(sqrtDist$Dist_GPO,
    na.rm = TRUE), stats::median(cubertDist$Dist_GPO, na.rm = TRUE),
    stats::median(recipDist$Dist_GPO, na.rm = TRUE), stats::median(bCDist$Dist_GPO,
        na.rm = TRUE), stats::median(log10Dist$Dist_GPO,
        na.rm = TRUE), stats::median(baseDistGPODF$Dist_GPO,
        na.rm = TRUE))
sdAccDistTrans <- c(stats::sd(sqrtDist$Dist_GPO, na.rm = TRUE),
    stats::sd(cubertDist$Dist_GPO, na.rm = TRUE), stats::sd(recipDist$Dist_GPO,
        na.rm = TRUE), stats::sd(bCDist$Dist_GPO, na.rm = TRUE),
    stats::sd(log10Dist$Dist_GPO, na.rm = TRUE), stats::sd(baseDistGPODF$Dist_GPO,
        na.rm = TRUE))
skewAccDistTrans <- c(moments::skewness(sqrtDist$Dist_GPO,
    na.rm = TRUE), moments::skewness(cubertDist$Dist_GPO,
    na.rm = TRUE), moments::skewness(recipDist$Dist_GPO,
    na.rm = TRUE), moments::skewness(bCDist$Dist_GPO, na.rm = TRUE),
    moments::skewness(log10Dist$Dist_GPO, na.rm = TRUE),
    moments::skewness(baseDistGPODF$Dist_GPO, na.rm = TRUE))
numOutAccDistTrans <- c(length(which(abs(distZScores1) >
    3)), length(which(abs(distZScores2) > 3)), length(which(abs(distZScores3) >
    3)), length(which(abs(distZScores4) > 3)), length(which(abs(distZScores5) >
    3)), length(which(abs(distZScores) > 3)))
transAccDF <- data.frame(transFun = c("Square root", "Cube Root",
    "Reciprocal", "BoxCox", "Log10", "Original"), mean = meanAccDistTrans,
    median = medianAccDistTrans, std_dev = sdAccDistTrans,
    skew = skewAccDistTrans, Outlier_Num = numOutAccDistTrans)
kable(transAccDF, caption = "Comarison of Transformations")
```

Table 13: Comarison of Transformations

transFun	mean	median	std_dev	skew	Outlier_Num
Square root	3.078377e+02	289.184236	6.753995e+01	1.9374352	5289
Cube Root	4.537508e+01	43.730362	6.358435e+00	1.5543663	5299
Reciprocal	1.190000e-05	0.000012	6.600000e-06	39.5932376	2010
BoxCox	9.932471e+04	83626.522515	5.123708e+04	3.0417536	6563
Log10	4.958768e+00	4.922349	1.716256e-01	0.6422555	5084
Original	9.932571e+04	83627.522515	5.123708e+04	3.0417536	5289

At figure 1 the histogram plot of Distance Kms shows an extreme right or positive skew. Accordingly only those transformation methods for positive skew were examined. At table 12 the transformations have reduced the maximums and minimums. It appears that the BoxCox method reduced this range the most but has a high differential between the third quartile and maximum. This is followed by the square root method slightly larger maximum -minimum range , but lower third quartile to maximum lower. The cube root has approximately the same maximum and minimum range but the lowest third quartile to maximum range.

Table 13 gives a different picture. The value of skew was calculated using the moments package skewness method(Agostino Test). The higher the number the greater the skew. The reciprocal method seems to be the best fit in term mean approaching median, and number of outliers but has a large skew number. The next best is the log10 method in mean approaching median, and number of outliers. It also has the lower skew number.

Given more time it would be propitious to understand the Agostino Test as it may weigh against the reciprocal solution. For this data set the log 10 method of transformation will be done on distance from the GPO.

```
accCombDF %<>%
    mutate(Dist_GPO = log10(Dist_GPO))
histDist2 <- ggplot(accCombDF, aes(x = Dist_GPO, y = after_stat(density))) +
    geom_histogram(binwidth = 0.05) + geom_density(colour = "green",
    linewidth = 0.3) + ggtitle("Histogram Distance to GPO") +
    scale_x_continuous(name = "Distance Kms", labels = function(x) x/1000) +
    theme(axis.title.y = element_text(angle = 0, vjust = 0.5,
        size = 6)) + theme(axis.title.x = element_text(angle = 0,
    vjust = 0.5, size = 6)) + theme(plot.title = element_text(size = 6)) +
    theme(axis.text.y = element_text(face = "bold", colour = "blue",
        angle = 90, size = 6)) + theme(axis.text.x = element_text(face = "bold",
    colour = "blue", angle = 45, size = 6))
figure2 <- grid.arrange(histDist, histDist2, nrow = 1, ncol = 2)
```

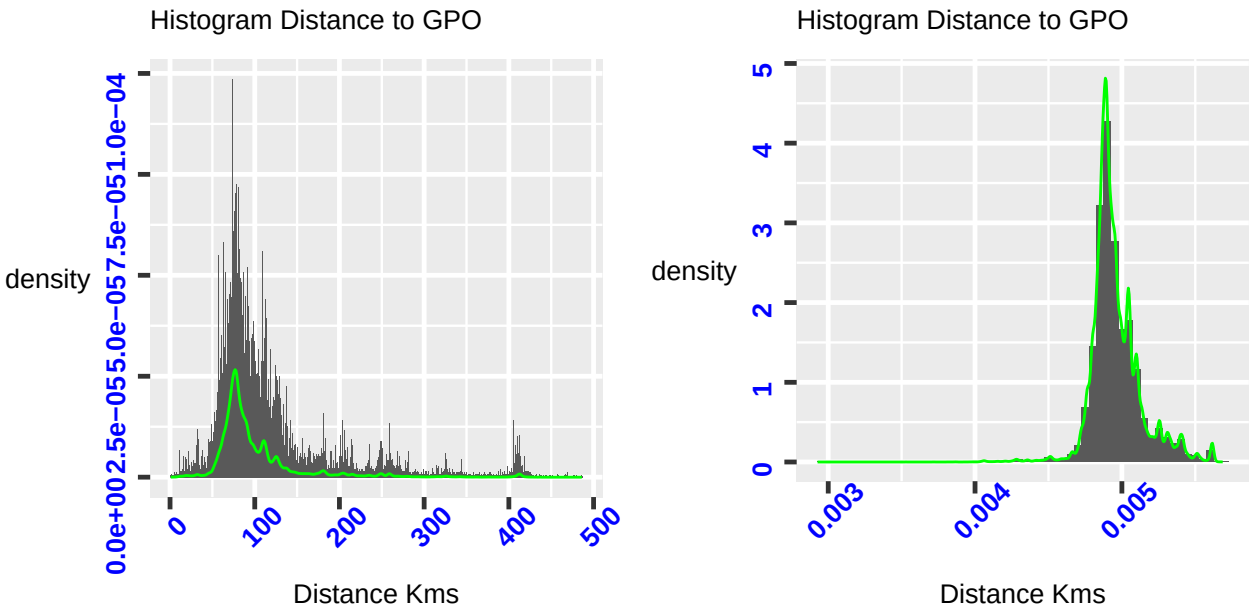


Figure 2: Comparison Histograms

Conclusion

While the abstract has not been answered. This report to date has developed a data frame that will be able to provide the answer. It is also believed that this data frame could be able to to develop a model to predict accident probabilities given a set of attributes. It is believed by the author that the key features of this data frame would be:

- gender
- age group
- cylinders
- full day
- time as float
- distance from GPO(or any other point)
- LGA

A possible usage of such a model might be in the insurance industry to apply to premiums.

Attached at Annex B is an example of a geo-spatial plot that can be generated from this data frame.

The author acknowledges the effort in producing the data, but has a criticism in the missing values and seemingly lack of control in some entries. It is also suggested that some form of relational map is developed, and data bases created. Again it is acknowledged that to fix twelve years would be an exhaustive task.

INITIAL EXAMINATION OF THE VICTORIAN ROAD CRASH DATA FILES

(Vic 2024)
| The Victoria Road Crash Data URL(Vic 2024) contains nine comma-separated value (csv) and one geo spatial java script object notation (GeoJSON) file. These were downloaded to a Data folder for examination. The files are:

```
csvFileNames <- list.files("../../Data",pattern = "*.csv", full.names = TRUE)
fullFileNames <- list.files("../../Data", full.names = TRUE)

for (file in fullFileNames) {
  cat(paste("-\t", file,"\n"))
}
```

- ../../Data/ACCIDENT.csv
- ../../Data/ACCIDENT_EVENT.csv
- ../../Data/ACCIDENT_LOCATION.csv
- ../../Data/ATMOSPHERIC_COND.csv
- ../../Data/NODE.csv
- ../../Data/PERSON.csv
- ../../Data/ROAD_SURFACE_COND.csv
- ../../Data/SUB_DCA.csv
- ../../Data/VEHICLE.csv
- ../../Data/VICTORIAN_ROAD_CRASH_DATA.geojson

```
#Create data frames fro csv files
vicRoadsDFList <- sapply(csvFileNames, read.csv)
#Change Key of list to mor HR form
names(vicRoadsDFList) <- c(gsub("../../Data/","", names(vicRoadsDFList)))
```

The URL indicates that the metadata was updated 29 April 2024, data observations updated as at 27 November 2024 and that observations temporal start was 1 January 2012.

```
posnOfDF <- 1
cat("These files have duplicated accident number keys in the data frame:")
```

These files have duplicated accident number keys in the data frame:

```
for (tmpDF in vicRoadsDFList) {
  if (n_distinct(tmpDF$ACCIDENT_NO) < dim(tmpDF)[[1]]) {
    cat(paste("-\t",names(vicRoadsDFList)[[posnOfDF]],"\n"))
  }
  posnOfDF <- posnOfDF + 1
}
```

- ACCIDENT_EVENT.csv
- ATMOSPHERIC_COND.csv
- NODE.csv
- PERSON.csv
- ROAD_SURFACE_COND.csv
- SUB_DCA.csv
- VEHICLE.csv

This would suggest that the `ACCIDENT_NO` key/attribute is a foreign key in these files.
The following is a lst of the attributes/column names by data frame.

```
vecColNamesDF <- list()
posnCN <- 1
for (df in vicRoadsDFList) {
    vecColNamesDF[[names(vicRoadsDFList)[[posnCN]]]] <- colnames(df)
    posnCN <- posnCN + 1
}
# https://stackoverflow.com/questions/60199801/how-to-view-a-list-like-table-style-in-r
max_len <- max(lengths(vecColNamesDF))
df <- purrr::map_df(vecColNamesDF, ~c(., rep("", max_len -
    length(.))))
df[, 1:5] %>%
    kable(caption = "Attributes Files 1-5", longtable = TRUE,
        format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 5)
```

Table 14: Attributes Files 1-5

ACCIDENT.csv	ACCIDENT_EVENT.csv	ACCIDENT_LOCATION.csv	ATMOSPHERIC_COND.csv	NODE.csv
ACCIDENT_NO	ACCIDENT_NO	ACCIDENT_NO	ACCIDENT_NO	ACCIDENT_NO
ACCIDENT_DATE	EVENT_SEQ_NO	NODE_ID	ATMOSPH_COND	NODE_ID
ACCIDENT_TIME	EVENT_TYPE	ROAD_ROUTE_1	ATMOSPH_COND_SEQ	NODE_TYPE
ACCIDENT_TYPE	EVENT_TYPE_DESC	ROAD_NAME	ATMOSPH_COND_DESC	AMG_X
ACCIDENT_TYPE_DESC	VEHICLE_1_ID	ROAD_TYPE		AMG_Y
DAY_OF_WEEK	VEHICLE_1_COLL_PT	ROAD_NAME_INT		LGA_NAME
DAY_WEEK_DESC	VEHICLE.1.COLL.PT.DESC	ROAD_TYPE_INT		DEG_URBAN_NAME
DCA_CODE	VEHICLE_2_ID	DISTANCE_LOCATION		LATITUDE
DCA_DESC	VEHICLE_2_COLL_PT	DIRECTION_LOCATION		LONGITUDE
LIGHT_CONDITION	VEHICLE.2.COLL.PT.DESC			POSTCODE_CRASH
NODE_ID	PERSON_ID			
NO_OF_VEHICLES	OBJECT_TYPE			
NO_PERSONS_KILLED	OBJECT_TYPE_DESC			
NO_PERSONS_INJ_2				
NO_PERSONS_INJ_3				
NO_PERSONS_NOT_INJ				
NO_PERSONS				
POLICE_ATTEND				
ROAD_GEOMETRY				
ROAD_GEOMETRY_DESC				
SEVERITY				
SPEED_ZONE				
RMA				

```
df[, 6:9] %>%
    kable(caption = "Attributes Files 6-9", longtable = TRUE,
        format = "latex", booktabs = TRUE) %>%
    kable_styling(font_size = 5)
```

Table 15: Attributes Files 6-9

PERSON.csv	ROAD_SURFACE_COND.csv	SUB_DCA.csv	VEHICLE.csv
ACCIDENT_NO	ACCIDENT_NO	ACCIDENT_NO	ACCIDENT_NO
PERSON_ID	SURFACE_COND	SUB_DCA_CODE	VEHICLE_ID
VEHICLE_ID	SURFACE_COND_DESC	SUB_DCA_SEQ	VEHICLE_YEAR_MANUF
SEX	SURFACE_COND_SEQ	SUB_DCA_CODE_DESC	VEHICLE_DCA_CODE
AGE_GROUP			INITIAL_DIRECTION
INJ_LEVEL			ROAD_SURFACE_TYPE
INJ_LEVEL_DESC			ROAD_SURFACE_TYPE_DESC
SEATING_POSITION			REG_STATE
HELMET_BELT_WORN			VEHICLE_BODY_STYLE
ROAD_USER_TYPE			VEHICLE_MAKE
ROAD_USER_TYPE_DESC			VEHICLE_MODEL
LICENCE_STATE			VEHICLE_POWER
TAKEN_HOSPITAL			VEHICLE_TYPE
EJECTED_CODE			VEHICLE_TYPE_DESC
			VEHICLE_WEIGHT
			CONSTRUCTION_TYPE
			FUEL_TYPE
			NO_OF_WHEELS
			NO_OF_CYLINDERS
			SEATING_CAPACITY
			TARE_WEIGHT
			TOTAL_NO_OCCUPANTS
			CARRY_CAPACITY
			CUBIC_CAPACITY
			FINAL_DIRECTION
			DRIVER_INTENT

VEHICLE_MOVEMENT
TRAILER_TYPE
VEHICLE_COLOUR_1
VEHICLE_COLOUR_2

CAUGHT_FIRE
INITIAL_IMPACT
LAMPS
LEVEL_OF_DAMAGE
TOWED_AWAY_FLAG

TRAFFIC_CONTROL
TRAFFIC_CONTROL_DESC

The following details those tables with common attribute names and what those names are.

```
namesOfFile <- names(vecColNamesDF)
for (posnOne in 1:length(namesOfFile)) {
    cat(namesOfFile[[posnOne]], "at list no.", posnOne,
        "intersects with the following file:")
    cat("  \n")
    if (posnOne == length(namesOfFile)) {
        break
    }
    for (posnTwo in (posnOne + 1):length(namesOfFile)) {
        cat("-\t", namesOfFile[[posnTwo]], "at list no.",
            posnTwo, " with these attributes:")
        cat("  \n")
        cat("\t\t-\t", intersect(vecColNamesDF[[posnOne]],
            vecColNamesDF[[posnTwo]]))
        cat("  \n")

        posnTwo <- posnTwo + 1
    }
    cat("  \n")
    posnOne <- posnOne + 1

}
```

ACCIDENT.csv at list no. 1 intersects with the following file:
- ACCIDENT_EVENT.csv at list no. 2 with these attributes:
- ACCIDENT_NO
- ACCIDENT_LOCATION.csv at list no. 3 with these attributes:
- ACCIDENT_NO NODE_ID
- ATMOSPHERIC_COND.csv at list no. 4 with these attributes:
- ACCIDENT_NO
- NODE.csv at list no. 5 with these attributes:
- ACCIDENT_NO NODE_ID
- PERSON.csv at list no. 6 with these attributes:
- ACCIDENT_NO
- ROAD_SURFACE_COND.csv at list no. 7 with these attributes:
- ACCIDENT_NO
- SUB_DCA.csv at list no. 8 with these attributes:
- ACCIDENT_NO
- VEHICLE.csv at list no. 9 with these attributes:
- ACCIDENT_NO

ACCIDENT_EVENT.csv at list no. 2 intersects with the following file:
- ACCIDENT_LOCATION.csv at list no. 3 with these attributes:
- ACCIDENT_NO
- ATMOSPHERIC_COND.csv at list no. 4 with these attributes:
- ACCIDENT_NO
- NODE.csv at list no. 5 with these attributes:
- ACCIDENT_NO
- PERSON.csv at list no. 6 with these attributes:
- ACCIDENT_NO PERSON_ID
- ROAD_SURFACE_COND.csv at list no. 7 with these attributes:
- ACCIDENT_NO
- SUB_DCA.csv at list no. 8 with these attributes:
- ACCIDENT_NO
- VEHICLE.csv at list no. 9 with these attributes:
- ACCIDENT_NO

ACCIDENT_LOCATION.csv at list no. 3 intersects with the following file:
- ATMOSPHERIC_COND.csv at list no. 4 with these attributes:
- ACCIDENT_NO
- NODE.csv at list no. 5 with these attributes:
- ACCIDENT_NO NODE_ID
- PERSON.csv at list no. 6 with these attributes:
- ACCIDENT_NO
- ROAD_SURFACE_COND.csv at list no. 7 with these attributes:
- ACCIDENT_NO
- SUB_DCA.csv at list no. 8 with these attributes:
- ACCIDENT_NO
- VEHICLE.csv at list no. 9 with these attributes:
- ACCIDENT_NO

ATMOSPHERIC_COND.csv at list no. 4 intersects with the following file:
- NODE.csv at list no. 5 with these attributes:
- ACCIDENT_NO
- PERSON.csv at list no. 6 with these attributes:
- ACCIDENT_NO
- ROAD_SURFACE_COND.csv at list no. 7 with these attributes:
- ACCIDENT_NO
- SUB_DCA.csv at list no. 8 with these attributes:
- ACCIDENT_NO
- VEHICLE.csv at list no. 9 with these attributes:
- ACCIDENT_NO

NODE.csv at list no. 5 intersects with the following file:
- PERSON.csv at list no. 6 with these attributes:
- ACCIDENT_NO
- ROAD_SURFACE_COND.csv at list no. 7 with these attributes:
- ACCIDENT_NO
- SUB_DCA.csv at list no. 8 with these attributes:
- ACCIDENT_NO
- VEHICLE.csv at list no. 9 with these attributes:
- ACCIDENT_NO

PERSON.csv at list no. 6 intersects with the following file:
- ROAD_SURFACE_COND.csv at list no. 7 with these attributes:
- ACCIDENT_NO
- SUB_DCA.csv at list no. 8 with these attributes:
- ACCIDENT_NO
- VEHICLE.csv at list no. 9 with these attributes:
- ACCIDENT_NO VEHICLE_ID

ROAD_SURFACE_COND.csv at list no. 7 intersects with the following file:
- SUB_DCA.csv at list no. 8 with these attributes:
- ACCIDENT_NO
- VEHICLE.csv at list no. 9 with these attributes:
- ACCIDENT_NO

SUB_DCA.csv at list no. 8 intersects with the following file:
- VEHICLE.csv at list no. 9 with these attributes:
- ACCIDENT_NO

VEHICLE.csv at list no. 9 intersects with the following file:

Local Government Areas as at 2021 were found at OPENDATASOFT(OPENDATASOFT n.d.).

```
councilsVic_sf <- read_sf("../../Councils/georef-australia-local-government-area.geojson") %>%
    filter(ste_name == "Victoria")
namesLGAVic <- unique(councilsVic_sf$lga_name) %>%
    append(., "Merri-bek")
```

```
councilsVic_sf["Longitude"] <- sapply(councilsVic_sf$geo_point_2d,
    function(in2d) parse_number(strsplit(in2d, split = ":")[[1]][[2]]))
councilsVic_sf["Latitude"] <- sapply(councilsVic_sf$geo_point_2d,
    function(in2d) parse_number(strsplit(in2d, split = ":")[[1]][[3]]))
testOSMPlus1 <- councilsVic_sf[, c("Latitude", "Longitude")]
df1 <- st_as_sf(testOSMPlus1, crs = "+proj=lonlat")
df_merc1 <- st_transform(df1, 4326)
dc1 <- get_tiles(df_merc1, provider = "OpenStreetMap", zoom = 8)
maleAccDf <- accCombDF %>%
    filter(Gender == "M" & !(is.na(Dist_GPO))) %>%
    select(Dist_GPO, Age_Group, Longitude, Latitude)
mycolours = c(brewer.pal(name = "BuPu", n = 9)[4:7], brewer.pal(name = "Blues",
    n = 9)[5:9], brewer.pal(name = "Oranges", n = 9)[4:9])
vicMaleMap <- ggplot() + geom_spatraster_rgb(data = dc1) +
    geom_sf(data = df1, aes(geometry = geometry), size = 0.3,
        alpha = 0, color = "red") + geom_point(data = maleAccDf,
    aes(x = Longitude, y = Latitude, colour = Age_Group),
    size = 0.1) + scale_color_manual(values = mycolours) +
    coord_sf(crs = 4326)
ggsave("vicmap.png", plot = vicMaleMap, width = 20, height = 15,
    units = "cm")
```

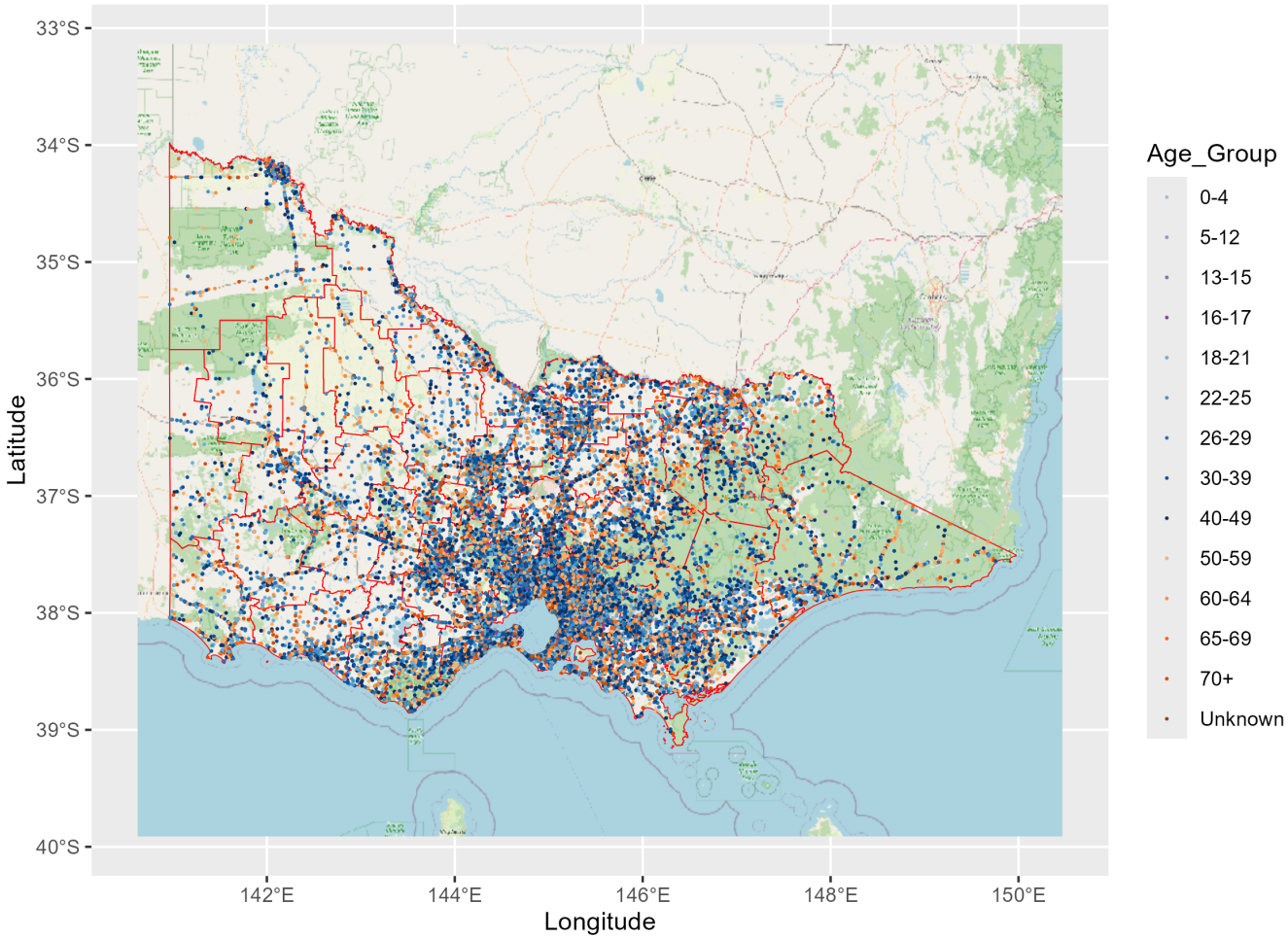


Figure 3: Accident Locations by Age Group Male

Bibliography

Auguie, B 2017, *gridExtra: Miscellaneous functions for "grid" graphics*,.

Auguie, B 2019, *Egg: Extensions for 'ggplot2': Custom geom, custom themes, plot alignment, labelled panels, symmetric scales, and fixed panel size*,.

Azzalini, A 2023, *Sn: The skew-normal and related distributions such as the skew-t and the SUN*,.

Bache, SM & Wickham, H 2022, *Magrittr: A forward-pipe operator for r*,.

Campitelli, E 2024, *Ggnewscale: Multiple fill and colour scales in 'ggplot2'*,.

Csárdi, G, Nepusz, T, Traag, V, Horvát, S, Zanini, F, Noom, D & Müller, K 2024, *Igraph: Network analysis and visualization*,.

de Jonge, E & van der Loo, M 2024, *Editrules: Parsing, applying, and manipulating data cleaning rules*,.

Fellows, I & JMapViewer library by Jan Peter Stotz, using the 2023, *OpenStreetMap: Access to open street map raster images*,.

Giraud, T 2024, *Maptiles: Download and display map tiles*,.

Harrell Jr, FE 2024, *Hmisc: Harrell miscellaneous*,.

Henry, L & Wickham, H 2024, *Tidyselect: Select from a set of strings*,.

Hernangómez, D 2024, *Tidyterra: 'Tidyverse' methods and 'ggplot2' helpers for 'terra' objects*,.

Hester, J & Bryan, J 2024, *Glue: Interpreted string literals*,.

Hijmans, RJ 2022, *Geosphere: Spherical trigonometry*,.

Hyndman, R, Athanasopoulos, G, Bergmeir, C, Caceres, G, Chhay, L, Kuroptev, K, O'Hara-Wild, M, Petropoulos, F, Razbash, S, Wang, E & Yasmeen, F 2024, *Forecast: Forecasting functions for time series and linear models*,.

Komsta, L 2022, *Outliers: Tests for outliers*,.

Komsta, L & Novomestky, F 2022, *Moments: Moments, cumulants, skewness, kurtosis and related tests*,.

Korkmaz, S, Goksuluk, D & Zararsiz, G 2021, *MVN: Multivariate normality tests*,.

Müller, K 2020, *Here: A simpler way to find your files*,.

Müller, K & Wickham, H 2023, *Tibble: Simple data frames*,.

Neuwirth, E 2022, *RColorBrewer: ColorBrewer palettes*,.

OPENDATASOFT, vol. 2024.

Pebesma, E 2024, *Sf: Simple features for r*,.

R Core Team 2024a, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

R Core Team 2024b, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

R Core Team 2024c, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

R Core Team 2024d, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

R Core Team 2024e, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

R Core Team 2024f, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

R Core Team 2024g, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

R Core Team 2024h, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

Schauberger, P & Walker, A 2023, *Openxlsx: Read, write and edit xlsx files*,.

Spinu, V, Grolemund, G & Wickham, H 2023, *Lubridate: Make dealing with dates a little easier*,.

van der Loo, M & de Jonge, E 2021, *Deductive: Data correction and imputation using deductive methods*,.

van der Loo, M & de Jonge, E 2024, *Validate: Data validation infrastructure*,.

van der Loo, M, de Jonge, E & Scholtus, S 2015, *Deducorrect: Deductive correction, deductive imputation, and deterministic correction*,.

Vic, D 2024, 'Victoria road crash data', vol. 2024.

Wickham, H 2023a, *Forcats: Tools for working with categorical variables (factors)*,.

Wickham, H 2023b, *Stringr: Simple, consistent wrappers for common string operations*,.

Wickham, H 2023c, *Tidyverse: Easily install and load the 'tidyverse'*,.

Wickham, H 2024, *Rvest: Easily harvest (scrape) web pages*,.

Wickham, H & Bryan, J 2023, *Readxl: Read excel files*,.

Wickham, H, Chang, W, Henry, L, Pedersen, TL, Takahashi, K, Wilke, C, Woo, K, Yutani, H, Dunnington, D & van den Brand, T 2024, *ggplot2: Create elegant data visualisations using the grammar of graphics*,.

Wickham, H, François, R, Henry, L, Müller, K & Vaughan, D 2023, *Dplyr: A grammar of data manipulation*,.

Wickham, H & Henry, L 2023, *Purrr: Functional programming tools*,.

Wickham, H, Hester, J & Bryan, J 2024, *Readr: Read rectangular text data*,.

Wickham, H, Pedersen, TL & Seidel, D 2023, *Scales: Scale functions for visualization*,.

Wickham, H, Vaughan, D & Girlich, M 2024, *Tidyr: Tidy messy data*,.

Xiao, N 2024, *Ggsci: Scientific journal and sci-fi themed color palettes for 'ggplot2'*,.

Xie, Y 2024, *Knitr: A general-purpose package for dynamic report generation in r*,.

Zhu, H 2024, *kableExtra: Construct complex table with 'kable' and pipe syntax*,.