Coursework 3

36071280

02/11/2021

# Question 1

Q1. Read in the data and print the dimensions of the Storm Events data frame

storm\_events <- read.csv("Australia\_severe\_storms\_1975-2015.csv")  
print(dim(storm\_events))

## [1] 14457 14

# Question 2

Q2. Clean the data by removing the variable ID and also Waterspout events from the database. Print the dimensions of the cleaned data frame. Also print the first few rows without the 6 columns of comments, without creating an intermediate data frame.

# Removing variable ID and Waterspout event  
storm\_events\_clean <- storm\_events %>% select(-(ID)) %>%   
 filter(storm\_events$Database != "Waterspout")  
  
print(dim(storm\_events\_clean))

## [1] 14417 13

column\_names <- colnames(storm\_events\_clean)  
list\_of\_comment\_columns <- column\_names[8:13]  
head(storm\_events\_clean %>% select(-all\_of(list\_of\_comment\_columns)))

## Event.ID Database Date.Time Nearest.town State Latitude Longitude  
## 1 20812 Wind 23/11/1975 07:00 SYDNEY NSW -33.8834 151.2167  
## 2 20813 Tornado 02/12/1975 14:00 BARHAM NSW -35.6333 144.1333  
## 3 20814 Wind 09/01/1976 08:50 COFF'S HARBOUR NSW -30.3167 153.1167  
## 4 20815 Hail 16/02/1976 14:00 BANKSTOWN NSW -33.8834 151.2167  
## 5 20816 Rain 25/10/1976 14:00 BOOMI NSW -28.4333 152.6167  
## 6 20817 Hail 08/11/1976 14:00 YOUNG NSW -34.3167 148.3000

# Question 3

Q3. Add a column to your data frame containing the time zone of each event using the following OlsonNames() classifications.

#declaring variables for timezone mapping  
  
list\_of\_relevant\_australian\_tz <- c("QLD" = "Australia/Queensland",  
 "NSW" = "Australia/NSW",  
 "NSW\_BrokenHill" = "Australia/Broken\_Hill",  
 "VIC" = "Australia/Victoria",  
 "SA"="Australia/South",  
 "WA" = "Australia/West",  
 "TAS"= "Australia/Tasmania",  
 "NT" = "Australia/North",  
 "ACT" = "Australia/ACT")  
  
new\_south\_wales <- "NSW"  
aus\_central\_time <- "ACT"  
broken\_hill <- "NSW\_BrokenHill"  
broken\_hill\_expr <- "broken hill"

allocate\_tz <-function(state, nearest\_town) {  
   
 if (length(state) && !is.na(state))  
 {  
 if (state != new\_south\_wales)  
 return (list\_of\_relevant\_australian\_tz[state])  
 else if (state == new\_south\_wales)  
 {  
 if (length(nearest\_town) && !is.na(nearest\_town) &&   
 str\_detect(tolower(nearest\_town), broken\_hill\_expr))  
 return (list\_of\_relevant\_australian\_tz[broken\_hill])  
 else  
 return (list\_of\_relevant\_australian\_tz[new\_south\_wales])  
 }  
 }  
 return (list\_of\_relevant\_australian\_tz[aus\_central\_time])  
}

storm\_events\_tz <- storm\_events\_clean %>% mutate(AustralianTimeZone = NA)  
  
for ( i in 1:nrow(storm\_events\_tz))  
{  
 storm\_events\_tz$AustralianTimeZone[i]<-allocate\_tz(storm\_events\_clean$State[i],  
 storm\_events\_clean$Nearest.town[i])  
}

# Question 4

Q4. Parse the date, time and time zones from the necessary columns to create a new variable in the data frame which converts the time into UTC. You may need the function lubridate::as\_datetime() and/or the use of loops. Print the first few rows of the resultant data frame, without the 6 columns of comments, again without creating an intermediate data frame.

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

storm\_events\_utc <- data.frame()  
for ( s in list\_of\_relevant\_australian\_tz)  
{  
 storm\_events\_utc <- rbind(storm\_events\_utc, storm\_events\_tz %>%   
 filter(storm\_events\_tz$AustralianTimeZone == s) %>%  
 mutate(UTCDateTime = (Date.Time %>% dmy\_hm(tz = s)   
 %>% as\_datetime())))  
}  
  
storm\_events\_utc <- storm\_events\_utc %>% arrange(storm\_events\_utc$Event.ID)  
head(storm\_events\_utc %>% select(-all\_of(list\_of\_comment\_columns)))

## Event.ID Database Date.Time Nearest.town State Latitude Longitude  
## 1 20812 Wind 23/11/1975 07:00 SYDNEY NSW -33.8834 151.2167  
## 2 20813 Tornado 02/12/1975 14:00 BARHAM NSW -35.6333 144.1333  
## 3 20814 Wind 09/01/1976 08:50 COFF'S HARBOUR NSW -30.3167 153.1167  
## 4 20815 Hail 16/02/1976 14:00 BANKSTOWN NSW -33.8834 151.2167  
## 5 20816 Rain 25/10/1976 14:00 BOOMI NSW -28.4333 152.6167  
## 6 20817 Hail 08/11/1976 14:00 YOUNG NSW -34.3167 148.3000  
## AustralianTimeZone UTCDateTime  
## 1 Australia/NSW 1975-11-22 20:00:00  
## 2 Australia/NSW 1975-12-02 03:00:00  
## 3 Australia/NSW 1976-01-08 21:50:00  
## 4 Australia/NSW 1976-02-16 03:00:00  
## 5 Australia/NSW 1976-10-25 04:00:00  
## 6 Australia/NSW 1976-11-08 03:00:00

# Question 5

Q5. Create new variables for the month and year of each event. Print the first few rows of the resultant data frame, without the 6 columns of comments and without creating an intermediate data frame.

#Adding month and year columns   
storm\_events\_with\_cols <- storm\_events\_utc %>%   
 mutate(month\_storm = month(storm\_events\_utc$UTCDateTime),   
 year\_storm = year(storm\_events\_utc$UTCDateTime))  
print(head(storm\_events\_with\_cols %>% select(-list\_of\_comment\_columns)))

## Note: Using an external vector in selections is ambiguous.  
## i Use `all\_of(list\_of\_comment\_columns)` instead of `list\_of\_comment\_columns` to silence this message.  
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.  
## This message is displayed once per session.

## Event.ID Database Date.Time Nearest.town State Latitude Longitude  
## 1 20812 Wind 23/11/1975 07:00 SYDNEY NSW -33.8834 151.2167  
## 2 20813 Tornado 02/12/1975 14:00 BARHAM NSW -35.6333 144.1333  
## 3 20814 Wind 09/01/1976 08:50 COFF'S HARBOUR NSW -30.3167 153.1167  
## 4 20815 Hail 16/02/1976 14:00 BANKSTOWN NSW -33.8834 151.2167  
## 5 20816 Rain 25/10/1976 14:00 BOOMI NSW -28.4333 152.6167  
## 6 20817 Hail 08/11/1976 14:00 YOUNG NSW -34.3167 148.3000  
## AustralianTimeZone UTCDateTime month\_storm year\_storm  
## 1 Australia/NSW 1975-11-22 20:00:00 11 1975  
## 2 Australia/NSW 1975-12-02 03:00:00 12 1975  
## 3 Australia/NSW 1976-01-08 21:50:00 1 1976  
## 4 Australia/NSW 1976-02-16 03:00:00 2 1976  
## 5 Australia/NSW 1976-10-25 04:00:00 10 1976  
## 6 Australia/NSW 1976-11-08 03:00:00 11 1976

# Queston 6

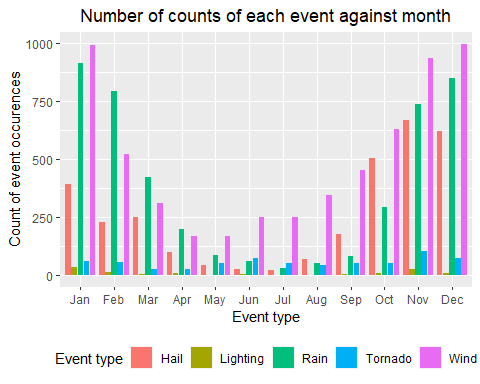
Q6. After discarding Waterspout events there are five types of events left in the data; Rain, Hail, Lighting, Wind, and Tornado.

## i) Create a new data frame which contains the total number of counts for each of the above type of events for each of the twelve months over the forty year period.

storm\_cols\_grouped <-storm\_events\_with\_cols %>% group\_by(Database, month\_storm)  
storm\_counts\_df <- data.frame(count(storm\_cols\_grouped))  
  
# 'storm\_counts\_df' dataframe contains the count of each event against each month

## ii)On a single plot, plot the total number of counts of each event against month. Use the R object month.abb for the labels of the months in the plot.

#Labeling months  
storm\_counts\_df$month\_storm <- factor(storm\_counts\_df$month\_storm, labels=month.abb)  
  
# Bar plot   
ggplot(storm\_counts\_df,aes(month\_storm, n, fill=factor(Database))) +   
 geom\_bar(position="dodge2", stat="identity") +  
 labs(x = "Event type", y="Count of event occurences", fill="Event type",  
 title = "Number of counts of each event against month") +  
 theme(legend.position = "bottom", plot.title = element\_text(hjust = 0.5))



# Using points  
#ggplot(storm\_counts\_df, aes(month\_storm, n, color= Database))+geom\_point()

# Question 7

Q7. From the answer to Question 5, the 6 columns titled Comments, X, X.1, X.2, X.3, X.4 consist of comments.

## i) Combine the comments from these columns into a single column, named All.comments.

# storm\_events\_with\_cols is the data frame from Q5.  
  
storm\_events\_with\_cols[is.na(storm\_events\_with\_cols)] <- ''  
  
column\_names\_comments <- colnames(storm\_events\_with\_cols)  
list\_of\_comment\_columns\_names <- column\_names[8:13]  
  
storm\_counts\_all\_comments <- storm\_events\_with\_cols %>% mutate(All.Comments = str\_trim(paste(storm\_events\_with\_cols$X, storm\_events\_with\_cols$X.1,  
 storm\_events\_with\_cols$X.2, storm\_events\_with\_cols$X.3,  
 storm\_events\_with\_cols$X.4, storm\_events\_with\_cols$Comments)))  
  
#glimpse(storm\_counts\_all\_comments)

## ii) Select the following columns to keep for further analysis, Event.ID, Database, State, All.comments, and the year variable you created.

storm\_analysis <- storm\_counts\_all\_comments %>% select(c(Event.ID, Database, State, All.Comments, year\_storm))

## iii) After which you should add the following command to your script: print(sapply(DF, class)) where DF is the name of the data frame.

print(sapply(storm\_analysis, class))

## Event.ID Database State All.Comments year\_storm   
## "integer" "character" "character" "character" "numeric"

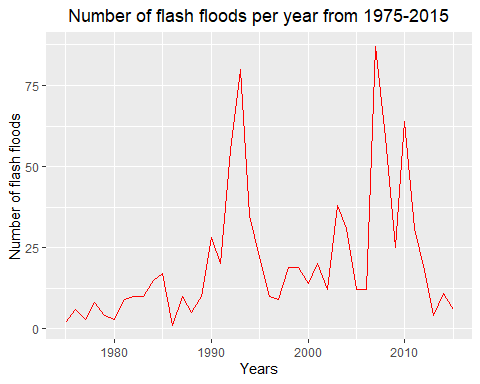
# Q8. Now we use the answer to Question 7(ii) for further analysis

## i) Create an indicator variable which states whether or not a storm event has resulted in a flash flood. Make sure you sort out all terms relating to flash floods.

flash\_flood\_expr <- "flash.\*?flood\*" # flash - any? - flood   
flash\_flood\_expr\_space\_mul <- "flash\\s\*?flood\*" # flash - space(s)? - flood   
flash\_flood\_expr\_mul <- "flash(.|\\s\*)?flood\*" #final  
  
storm\_analysis <- storm\_analysis %>% mutate(flash\_flood\_present = str\_detect(storm\_analysis$All.Comments, regex(flash\_flood\_expr\_mul, ignore\_case = TRUE)))

## ii) Print a plot of the number of flash floods per year from 1975-2015. You may need to first create a vector or data frame to contain the number of flash floods per year.

storm\_analysis\_with\_floods <- storm\_analysis %>% filter(flash\_flood\_present==TRUE)  
storm\_analysis\_grouped <- storm\_analysis\_with\_floods %>% group\_by(year\_storm)  
  
flash\_flood\_counts <- count(storm\_analysis\_grouped)  
  
ggplot(flash\_flood\_counts) + geom\_line(aes(year\_storm, n), color="red") +  
 labs(x = "Years", y="Number of flash floods",  
 title = "Number of flash floods per year from 1975-2015") +  
 theme(plot.title = element\_text(hjust = 0.5))



# Q9. For severe wind events often the wind speed is given. The wind speed is given in knots or km/h.

## i) Extract all wind speeds both those in knots and km/h. Hint: Knots can be abbreviated by kts or kt. Also note that wind speed can be a single, double or triple digit number

wind\_speed\_expr <- "(\\d{1,3})\\s?(knot|kt(s)?|km\\s?\\/\\s?h(r)?)"  
  
storm\_analysis\_speeds <- str\_extract\_all(storm\_analysis,regex(wind\_speed\_expr))

## Warning in stri\_extract\_all\_regex(string, pattern, simplify = simplify, :  
## argument is not an atomic vector; coercing

storm\_analysis\_speeds <- storm\_analysis %>% mutate(wind\_speed = str\_extract(  
 storm\_analysis$All.Comments, wind\_speed\_expr))  
  
storm\_analysis\_speeds <- storm\_analysis\_speeds %>% filter(!is.na(wind\_speed))

## ii) Convert km/h wind speeds to knots (1 knot = 1.852 km/h) rounding the wind/speed to the nearest knot. Hint: It is helpful to work with a reduced data frame which includes only those observations with a wind speed recorded.

knot\_to\_kmh = 1.852  
wind\_speed\_expr\_km <- "(\\d{1,3})\\s?(km\\s?\\/\\s?h(r)?)"  
extract\_number\_expr <- "\\d+"  
  
get\_wind\_value <- function (wind\_speed) {  
 wind\_numeric\_value = as.numeric(str\_extract(wind\_speed, extract\_number\_expr))  
 if (str\_detect(wind\_speed, regex(wind\_speed\_expr\_km, ignore\_case = TRUE)))  
 {  
 return (as.integer(wind\_numeric\_value/knot\_to\_kmh))  
 }  
 else  
 {  
 return (wind\_numeric\_value)  
 }  
}

storm\_analysis\_speed\_numbers <- storm\_analysis\_speeds %>% mutate(wind\_values = NA)  
storm\_analysis\_speed\_numbers$wind\_values <- sapply(storm\_analysis\_speed\_numbers$wind\_speed, FUN = get\_wind\_value)

## iii) Print a boxplot of the wind speeds recorded per state.

ggplot(storm\_analysis\_speed\_numbers, aes(State, wind\_values)) +   
 geom\_boxplot(position="dodge2", color ="black") +   
 labs(x="State", y="Wind speeds", title="Plot of Wind speeds by State") +  
 theme(plot.title = element\_text(hjust = 0.5))

