R Coursework 3

36306330

2022-11-09

R Markdown

Task1 : print dimension

```
Aus_storm <- read.csv(file = 'Australia_severe_storms_1975-2015.csv')
dim(Aus_storm)
```

```
## [1] 14457    15
```

Task 2: clean the data

```
# remove ID & waterspout in Database
Aus_storm <- Aus_storm %>% select(-ID) %>%
  filter(Database != 'Waterspout')
# print dimension
dim(Aus_storm)
```

```
## [1] 14417    14
```

```
# print first few rows without the 7 columns of comments
head(select(Aus_storm, -Comments, -X, -X.1, -X.2, -X.3, -X.4, -X.5))
```

```
##   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
```

Task 3: Add a column containing time zone of each event

```
#if_else to check assign timezone
assign_timezone <- function(state, town){
  if (state == 'QLD'){
    timezone = 'Australia/Queensland'
  }
  if (state == 'NSW'){
    if (str_detect(town, '([Bb][Rr][Oo][Kk][Ee][Nn] [Hh][Ii][Ll][Ll])')){
        timezone = 'Australia/Broken_Hill'
    }
    else {
      timezone = 'Australia/NSW'
    }
  }
  if (state == 'VIC'){
    timezone = 'Australia/Victoria'
  }
  if (state == 'SA'){
    timezone = 'Australia/South'
  }
  if (state == 'WA'){
    timezone = 'Australia/West'
  }
  if (state == 'TAS'){
    timezone = 'Australia/Tasmania'
  }
  if (state == 'NT'){
    timezone = 'Australia/North'
  }
  if (state == 'ACT'){
    timezone = 'Australia/ACT'
  }
  return(timezone)
}

#for loop for dataframe
for (i in 1:nrow(Aus_storm)){
  Aus_storm$Timezone[i] <- assign_timezone(Aus_storm$State[i],Aus_storm$Nearest.town[i])
}
```

Task 4: Converts the time into UTC

```
#for loop to convert the data to date time format according timezone
results <- POSIXct()
for (i in 1:nrow(Aus_storm)){
  results[i] <- dmy_hm(Aus_storm$Date.Time[i], tz = Aus_storm$Timezone[i])
}
Aus_storm$Date.Time.UTC <- results

#Print the first few rows without the 7 columns of comments
head(select(Aus_storm, -Comments, -X, -X.1, -X.2, -X.3, -X.4, -X.5))
```

```
##   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
##         Timezone       Date.Time.UTC
## 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
```

Task 5: Create new variables for the month and year of each event

```
#create variable month & year
Aus_storm$Date.Time <- dmy_hm(Aus_storm$Date.Time)
Aus_storm$Month <- month(Aus_storm$Date.Time)
Aus_storm$Year <- year(Aus_storm$Date.Time)

# #Print the first few rows without the 7 columns of comments
head(select(Aus_storm, -Comments, -X, -X.1, -X.2, -X.3, -X.4, -X.5))
```

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

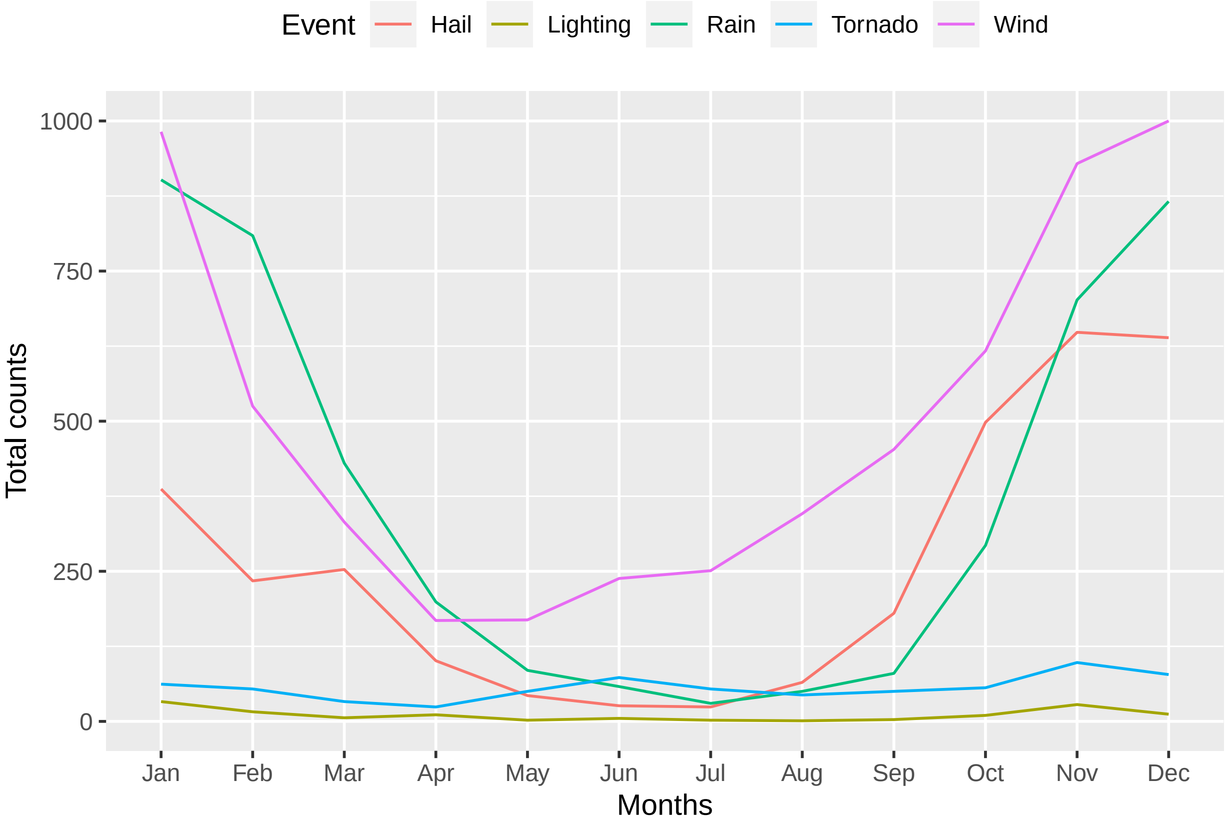
Task 6:

i) Create a new data frame: total number of counts of each events for each twelve months over the forty year period

```
Events_month <- Aus_storm %>% count(Month, Event = Database)
```

ii) Plot total number of counts of each event against month

```
ggplot(Events_month) +
  geom_line(aes(x = Month, y = n, color = Event)) +
  theme(legend.position = 'top') +
  ylab('Total counts') +
  xlab('Months') +
  scale_x_discrete(limits = month.abb)
```



Task 7: Modify Comments columns

i) Combine the comments into a single column: All.comments

```
Aus_storm <- Aus_storm %>% unite('All.comments', Comments:X.5, sep = " ")
```

ii) Select Event.ID, Database, State, All.comments, and the year variable created.

```
Aus_storm <- Aus_storm %>% select(Event.ID, Database, State, All.comments, Year)
```

iii) print(sapply(DF, class))

```
print(sapply(Aus_storm, class))
```

```
##     Event.ID     Database        State All.comments         Year
##    "integer"  "character"  "character"  "character"    "numeric"
```

Task 8:

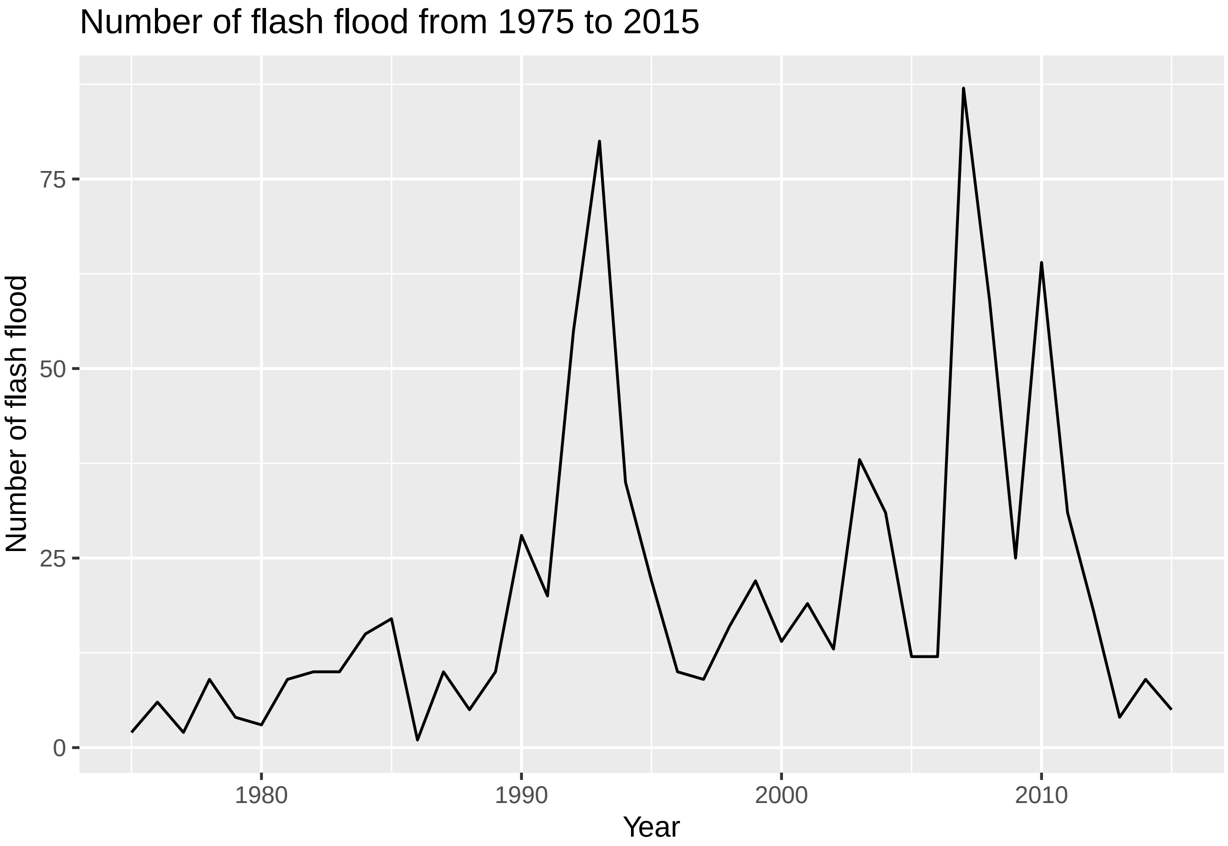
i) Create an indicator variable which states whether or not a storm event has resulted in a flash flood.

```
expr <- '([Ff]lash\\s*[Ff]lood\\w*)'
Aus_storm$flash_flood <- str_detect(Aus_storm$All.comments, expr)
```

ii) Plot of the number of flash floods per year from 1975-2015. [3 marks]

```
#create a data frame to store number of flash floods and year
#filter true only for flash flood
flash_flood_by_year <- filter(Aus_storm, flash_flood == T)
flash_flood_by_year <- count(flash_flood_by_year, Year, flash_flood)
```

```
ggplot(flash_flood_by_year) +
  geom_line(aes(x = Year, y = n)) +
  xlab('Year') +
  ylab('Number of flash flood') +
  ggtitle('Number of flash flood from 1975 to 2015')
```



Task 9:

i) Extract all wind speeds (knots and km/h)

```
# km/h, km/hr, Km/hr, km.hr
# kmph, kmh, kph,
# kt, kts, KT
# KNOTS, knot, knots
# kn

expr <- '\\b\\d{1,3}\\s*[Kk][Tt][sS]?|\\d+\\s*kn|\\d+\\s*[Kk][Nn][Oo][Tt][Ss]?
|\\d{1,3}\\s*[kK]m[./]hr?|\\d+\\s*k[mp]p?h'
Aus_storm$wind.speed <- str_extract(Aus_storm$All.comments, expr)

# \\b\\d{1,3} to capture 1 to 3 digits of wind speed, some of the datas
# have extremely large values (not possible for wind speed)
```

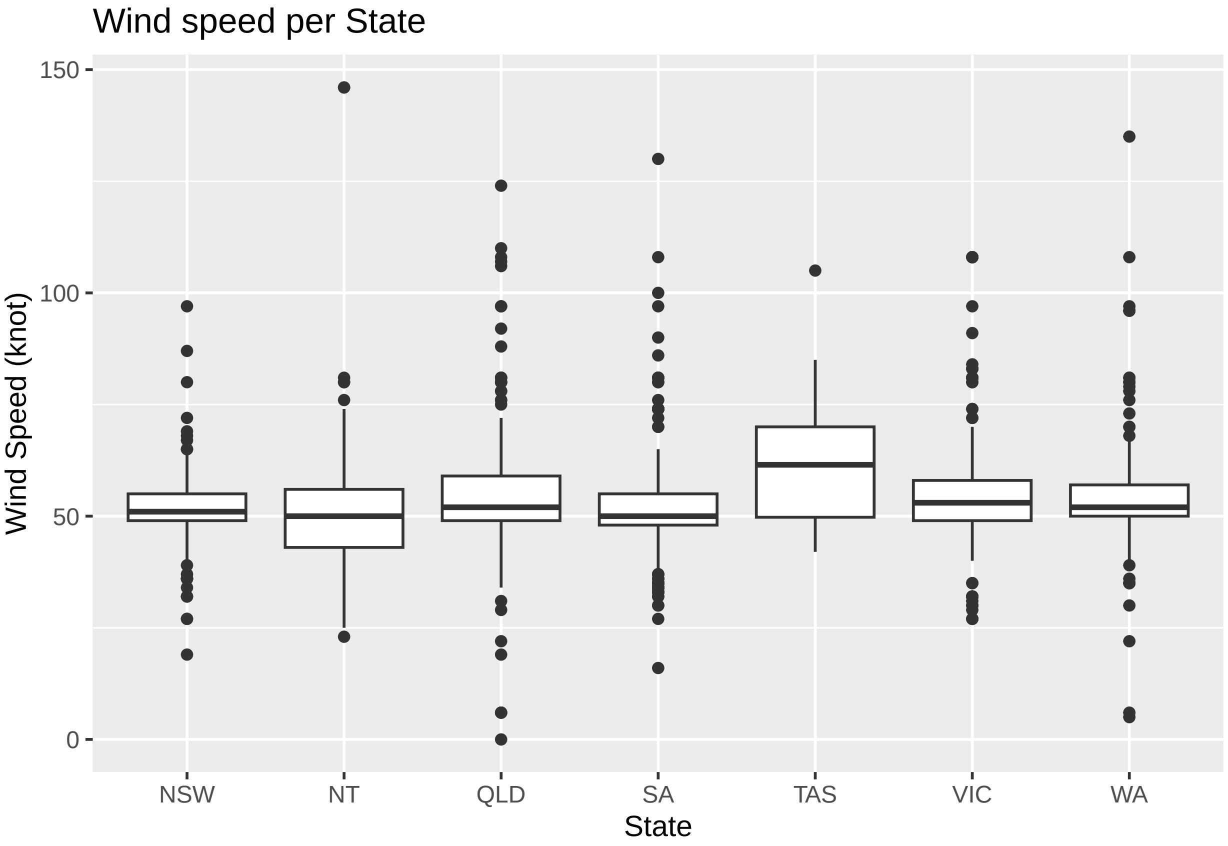
ii) Convert km/h wind speeds to knots (1 knot = 1.852 km/h) rounding the wind/speed to the nearest knot.

```
#filter dataframe to exclude wind speed NA
Aus_storm <- Aus_storm %>% filter(!is.na(wind.speed))
```

```
#extract to a vector
ws_knot <- as.numeric(str_extract(Aus_storm$wind.speed, '\\d+'))

for (i in 1:nrow(Aus_storm)){
if (str_detect(Aus_storm$wind.speed[i], '\\d{1,3}\\s*[kK]m[./]hr?|\\d+\\s*k[mp]p?h'))
  {
    Aus_storm$wind.speed[i] <- round(ws_knot[i] / 1.852, digit = 0)
  }
else {
Aus_storm$wind.speed[i] <- ws_knot[i]
}
  }

#rename col to specifiy unit knot
Aus_storm <- Aus_storm %>% rename(wind.speed.knot = wind.speed)
```

iii) Boxplot of the wind speeds recorded per state.

```
#change col class to numeric
Aus_storm$wind.speed.knot <- as.numeric(Aus_storm$wind.speed.knot)

#Boxplot
ggplot(Aus_storm) +
  geom_boxplot(aes(x = State, y = wind.speed.knot)) +
  ylab('Wind Speed (knot)') +
  ggtitle('Wind speed per State')
```


Wind speed per State
Wind Speed (knot)
NSW
NT
QLD
SA
TAS
VIC
WA
State