Coursework 3

36071280

02/11/2021

```
library(tidyverse)
```

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", show_col_types = FALSE)
```

```
## New names:
## * `` -> ...10
## * `` -> ...11
## * `` -> ...12
## * `` -> ...13
## * `` -> ...14
```

```
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 Waterspount event
storm_events_clean <- storm_events %>% select(-(ID)) %>% filter(storm_events$Database != "Waterspout")
print(dim(storm_events_clean))
```

```
## [1] 14417    13
```

#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"
```

```
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_e
      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_clean))
{
  storm_events_tz$AustralianTimeZone[i] <- allocate_tz(storm_events_clean$State[i], storm_events_clean$
}

#storm_events_tz2 = lapply(storm_events_tz, 2, FUN = allocate_tz)
```

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()
#storm_events_utc <- storm_events_tz %>% mutate(UTCDateTime = NA)

for ( s in  list_of_relevant_australian_tz)
```

```
{
  #print(s)
  storm_events_utc <- rbind(storm_events_utc, storm_events_tz %>% filter(storm_events_tz$AustralianTime
    #storm_events_utc$UTCDateTime

}
storm_events_utc %>% arrange(storm_events_utc$`Event ID`)
```

```
## # A tibble: 14,417 x 15
##    `Event ID` Database `Date/Time`    `Nearest town`     State Latitude Longitude
##         <dbl> <chr>    <chr>          <chr>              <chr>    <dbl>     <dbl>
##  1      20812 Wind     23/11/1975 0~ SYDNEY             NSW      -33.9      151.
##  2      20813 Tornado  02/12/1975 1~ BARHAM             NSW      -35.6      144.
##  3      20814 Wind     09/01/1976 0~ COFF'S HARBOUR     NSW      -30.3      153.
##  4      20815 Hail     16/02/1976 1~ BANKSTOWN          NSW      -33.9      151.
##  5      20816 Rain     25/10/1976 1~ BOOMI              NSW      -28.4      153.
##  6      20817 Hail     08/11/1976 1~ YOUNG              NSW      -34.3      148.
##  7      20818 Hail     09/11/1976 1~ SYDNEY (WESTERN S~ NSW      -33.7      151
##  8      20818 Rain     09/11/1976 1~ WESTERN SUBURBS    NSW      -33.7      151
##  9      20819 Wind     09/11/1976 1~ SYDNEY             NSW      -33.9      151.
## 10      20820 Wind     10/11/1976 1~ MOREE/TRANGIE      NSW      -29.5      150.
## # ... with 14,407 more rows, and 8 more variables: Comments <chr>, ...10 <chr>,
## #   ...11 <chr>, ...12 <lgl>, ...13 <lgl>, ...14 <lgl>,
## #   AustralianTimeZone <chr>, UTCDateTime <dttm>
```

```
storm_events_utc2 <- storm_events_tz %>% mutate(UTCDateTime = NA)

for ( j in 1: nrow(storm_events_utc2)) {
  storm_events_utc2$UTCDateTime[j] <- (dmy_hm(storm_events_utc2$`Date/Time`[j], tz=storm_events_utc2$Aus
}
storm_events_utc2$UTCDateTime <- storm_events_utc2$UTCDateTime %>% as_datetime()

#comment_columns = c(...10,...11, ...12,...13, ...14, Comments)
# filter all the tz by the name,
# mass apply dmy_hm  -> loop should be 1:10
head(storm_events_utc2 %>% select(-c(...10,...11, ...12,...13, ...14, Comments)))
```

```
## # A tibble: 6 x 9
##   `Event ID` Database `Date/Time`      `Nearest town` State Latitude Longitude
##        <dbl> <chr>    <chr>            <chr>          <chr>    <dbl>     <dbl>
## 1      20812 Wind     23/11/1975 07:00 SYDNEY         NSW      -33.9      151.
## 2      20813 Tornado  02/12/1975 14:00 BARHAM         NSW      -35.6      144.
## 3      20814 Wind     09/01/1976 08:50 COFF'S HARBOUR NSW      -30.3      153.
## 4      20815 Hail     16/02/1976 14:00 BANKSTOWN      NSW      -33.9      151.
## 5      20816 Rain     25/10/1976 14:00 BOOMI          NSW      -28.4      153.
## 6      20817 Hail     08/11/1976 14:00 YOUNG          NSW      -34.3      148.
## # ... with 2 more variables: AustralianTimeZone <chr>, UTCDateTime <dttm>
```

```
storm_events_utc = storm_events_utc2
```

#Q5. 5. 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.

```
storm_events_with_cols = storm_events_utc %>% mutate(month_storm = month(storm_events_utc$UTCDateTime),
                                                     year_storm = year(storm_events_utc$UTCDateTime))
```

```
#...10,...11, ...12,...13, ...14, Comments
print(storm_events_with_cols[-c(8:13)])
```

```
## # A tibble: 14,417 x 11
##    `Event ID` Database `Date/Time`   `Nearest town`     State Latitude Longitude
##         <dbl> <chr>    <chr>         <chr>              <chr>    <dbl>     <dbl>
##  1      20812 Wind     23/11/1975 0~ SYDNEY             NSW      -33.9      151.
##  2      20813 Tornado  02/12/1975 1~ BARHAM             NSW      -35.6      144.
##  3      20814 Wind     09/01/1976 0~ COFF'S HARBOUR     NSW      -30.3      153.
##  4      20815 Hail     16/02/1976 1~ BANKSTOWN          NSW      -33.9      151.
##  5      20816 Rain     25/10/1976 1~ BOOMI              NSW      -28.4      153.
##  6      20817 Hail     08/11/1976 1~ YOUNG              NSW      -34.3      148.
##  7      20818 Hail     09/11/1976 1~ SYDNEY (WESTERN S~ NSW      -33.7      151
##  8      20818 Rain     09/11/1976 1~ WESTERN SUBURBS    NSW      -33.7      151
##  9      20819 Wind     09/11/1976 1~ SYDNEY             NSW      -33.9      151.
## 10      20820 Wind     10/11/1976 1~ MOREE/TRANGIE      NSW      -29.5      150.
## # ... with 14,407 more rows, and 4 more variables: AustralianTimeZone <chr>,
## #   UTCDateTime <dttm>, month_storm <dbl>, year_storm <dbl>
```

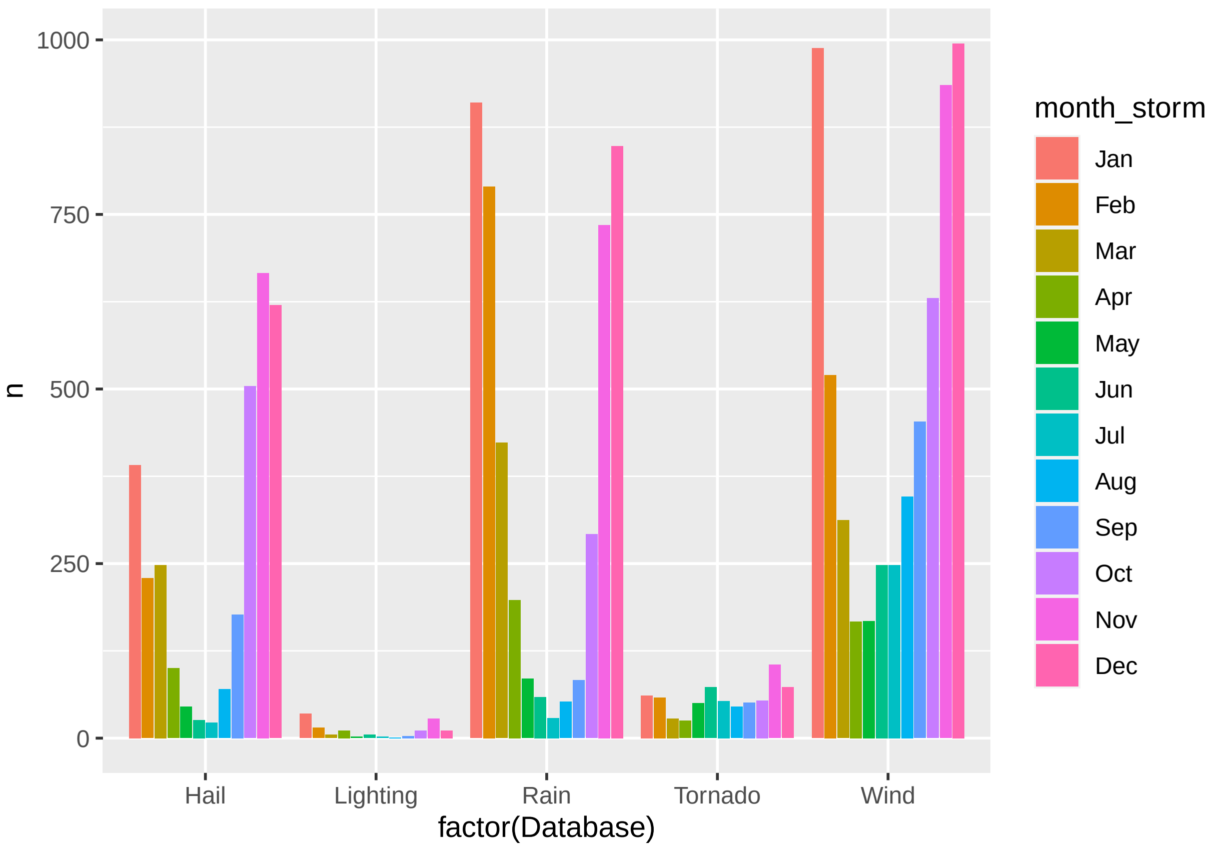
Q6. After discarding Waterspout events there are five types of events left in the data; Rain, Hail, Lighting, Wind, and Tornado.

##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))
```

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.

```
storm_counts_df$month_storm <- factor(storm_counts_df$month_storm, labels=month.abb)

ggplot(storm_counts_df,aes(factor(Database), n, fill=month_storm )) +
  geom_bar(position="dodge2", stat="identity")
```



```
#ggplot(storm_counts_df,aes(factor(Database), n, fill=month_storm )) +
 # geom_line()
```

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.na(storm_events_with_cols)] <- ''
storm_counts_all_comments <- storm_events_with_cols %>% mutate(AllComments = str_trim(paste(storm_event
                                                                                  storm_events_with_co
                                                                                  storm_events_with_co

#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_counts_all_comments$`Event ID`
#select(storm_counts_all_comments, storm_counts_all_comments$`Event ID`)
storm_analysis <-  storm_counts_all_comments %>% select(c(`Event ID`, Database, State, AllComments, yea

print(sapply(storm_analysis, class))
```

```
##    Event ID    Database       State AllComments  year_storm
##   "numeric" "character" "character" "character"   "numeric"
```

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

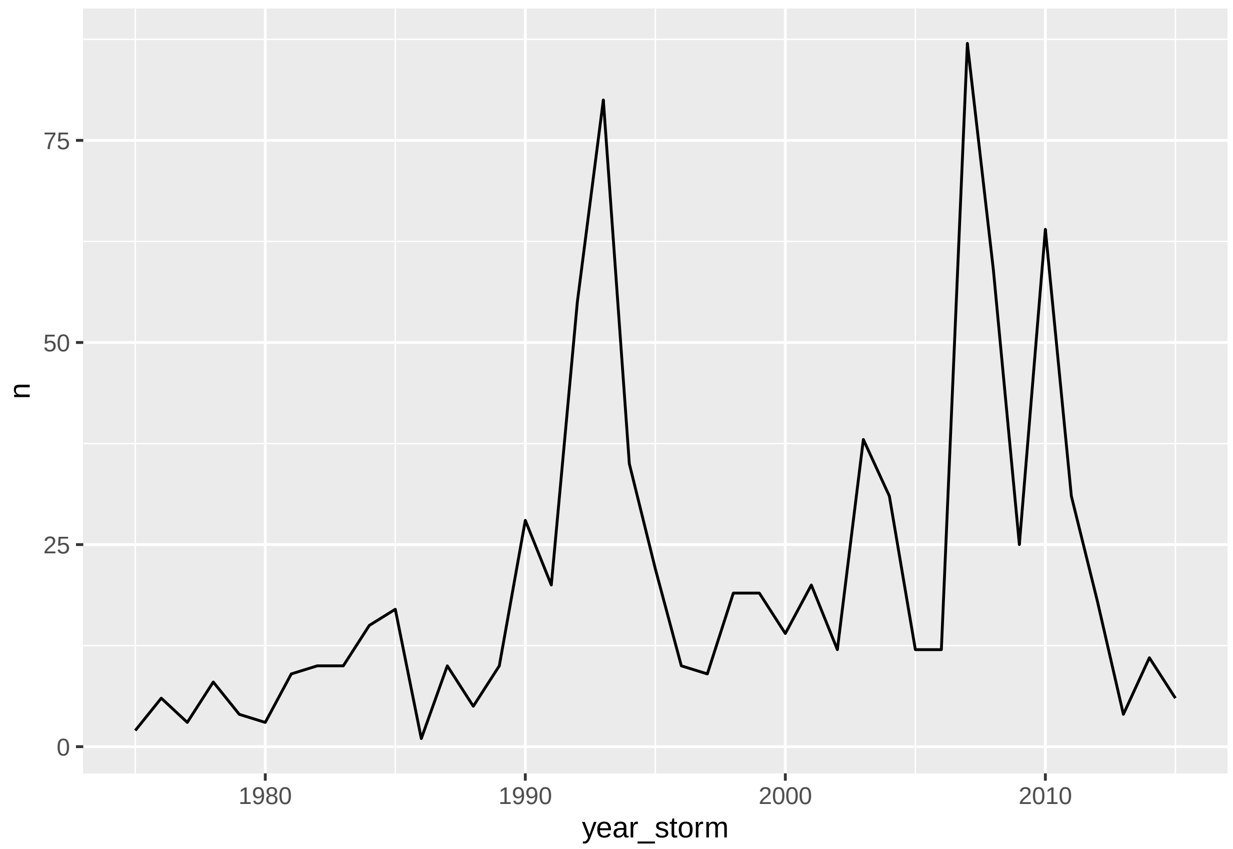
i) Create an indicator variable which states whether or not a storm event has resulted in a flash flood. Make sure you sort out all terms relating to flash floods.

```
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$AllComments

storm_analysis_with_floods <- storm_analysis %>% filter(flash_flood_present==TRUE)
storm_analysis_grouped <- storm_analysis_with_floods %>% group_by(year_storm)

dim(storm_analysis_grouped)
```

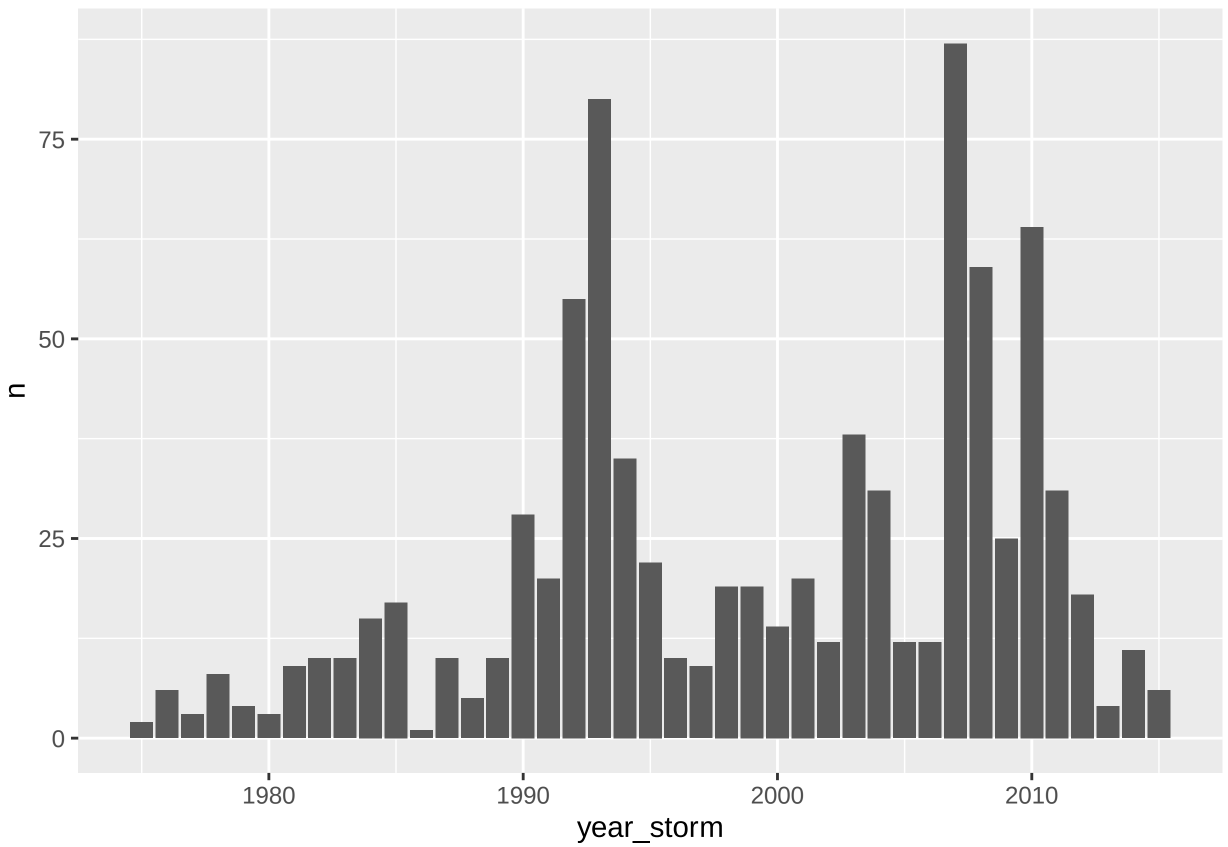
```
## [1] 854   6
```

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.

```
flash_flood_counts <- count(storm_analysis_grouped)

ggplot(flash_flood_counts) + geom_line(aes(year_storm, n))
```



```
ggplot(flash_flood_counts,aes(year_storm, n)) + geom_bar(stat="identity")
```



```
#View(flash_flood_counts)
```

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$AllComments

storm_analysis_speeds <- storm_analysis_speeds %>% filter(!is.na(wind_speed))

#write.csv(storm_analysis_speeds, "storm_analysis_speeds.csv")
```

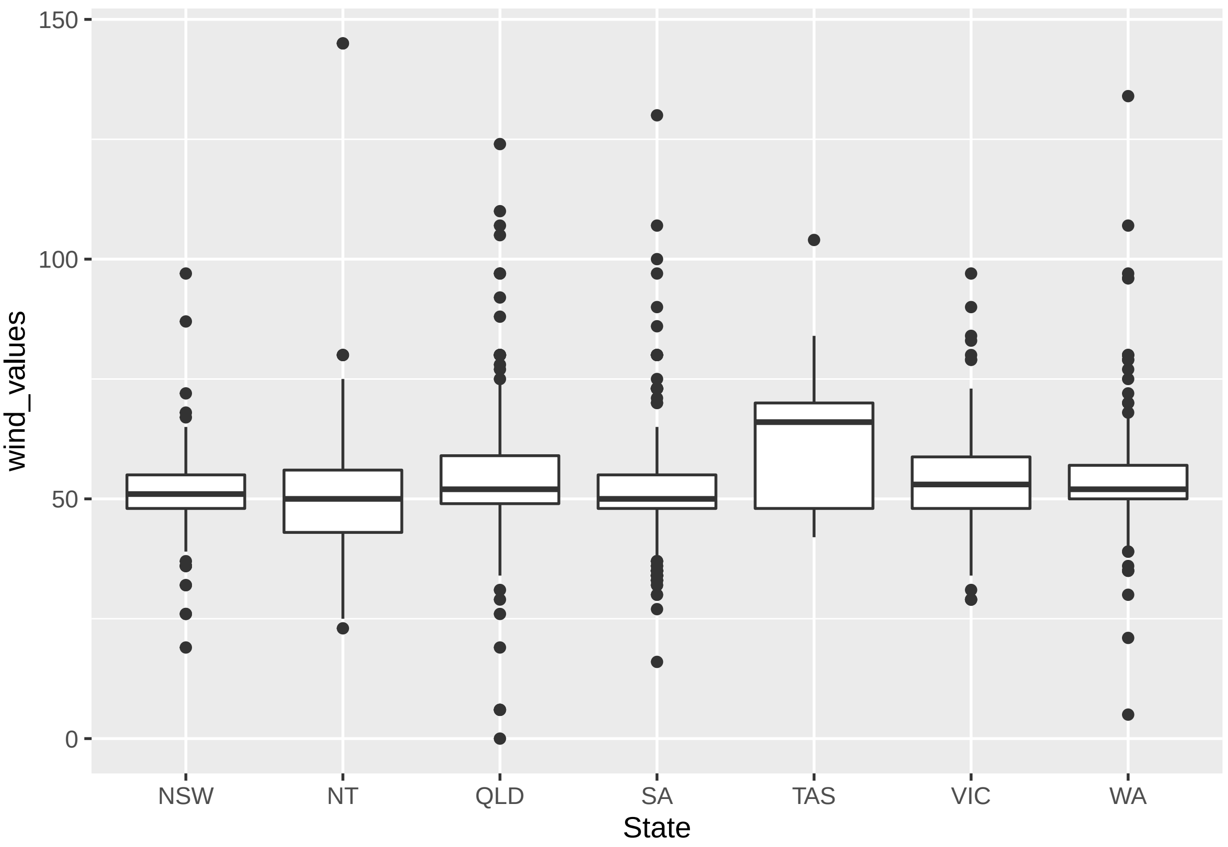
ii) Convert km/h wind speeds to knots (1 knot = 1.852 km/h) rounding the wind/speed to the nearest knot. Hint: It is helpful to work with a reduced data frame which includes only those observations with a wind speed recorded.

```
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_wi

#as.numeric(str_extract(storm_analysis_speeds$wind_speed[27], extract_number_expr))
#storm_analysis_speed_numbers[1:15,]
```

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

```
ggplot(storm_analysis_speed_numbers, aes(State, wind_values)) + geom_boxplot(position="dodge2")
```


150
100
50
0
wind_values
NSW
NT
QLD
SA
TAS
VIC
WA
State