Introduction

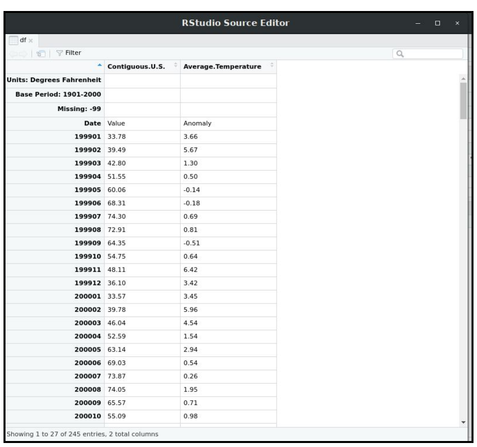
The data that is used based on the roll number is from the year 1999 to the year 2019 for all months taking the average temperature as the parameter. The following sections show the demo code and graphical proof for each section.

Task

Your student ID, your chosen period and the main characteristics of the weather for this period. Pre-process, prepare and clean the data and convert them to the CSV format.

Data Used - January 1999 to January 2019, if the last digit of your ID is 8 or 9.

Source - NOAA National Centers for Environmental Information, Climate at a Glance: National Time Series, published December 2019, Retrieved on December 16, 2019, from https://www.ncdc.noaa.gov/cag/



RStudio Source Editor

df

Filter

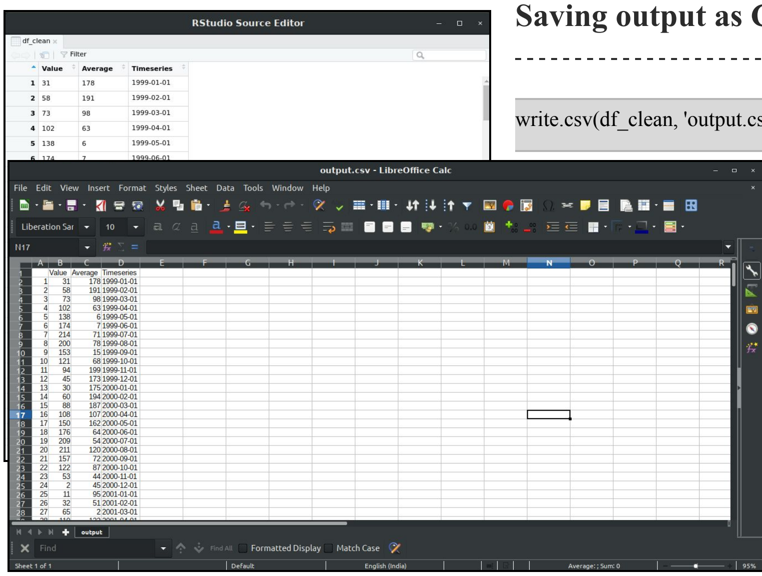
	Contiguous.U.S.	Average.Temperature
Units: Degrees Fahrenheit		
Base Period: 1901-2000		
Missing: -99		
Date	Value	Anomaly
199901	33.78	3.66
199902	39.49	5.67
199903	42.80	1.30
199904	51.55	0.50
199905	60.06	-0.14
199906	68.31	-0.18
199907	74.30	0.69
199908	72.91	0.81
199909	64.35	-0.51
199910	54.75	0.64
199911	48.11	6.42
199912	36.10	3.42
200001	33.57	3.45
200002	39.78	5.96
200003	46.04	4.54
200004	52.59	1.54
200005	63.14	2.94
200006	69.03	0.54
200007	73.87	0.26
200008	74.05	1.95
200009	65.57	0.71
200010	55.09	0.98

Showing 1 to 27 of 245 entries, 2 total columns

Pre Processing

```
library(ggplot2)


df = read.csv('110-tavg-all-1-1999-2019.txt')
df_clean = df[-c(1:4),]
timeseries = rownames(df_clean)
colnames(df_clean)[1] <- format(df[4,1])
colnames(df_clean)[2] <- 'Average'
timeseries = paste(substr(timeseries,1,4), substr(timeseries,5,6), '01', sep = "-")
df_clean = data.frame(as.integer(df_clean$Value), as.integer(df_clean$Average), ts(timeseries))
colnames(df_clean)[1] <- format(df[4,1])
colnames(df_clean)[2] <- 'Average'
colnames(df_clean)[3] <- 'Timeseries'
rownames(df_clean) = c(1:nrow(df_clean))
#rownames(df_clean) = ts(timeseries)
```



Saving output as CSV

```
write.csv(df_clean, 'output.csv')
```

Visualize weather data for this period. Describe the general trend of the data for this period. Produce your own graph. Print out the mean and variance of the data, and describe the distribution by using the histogram.

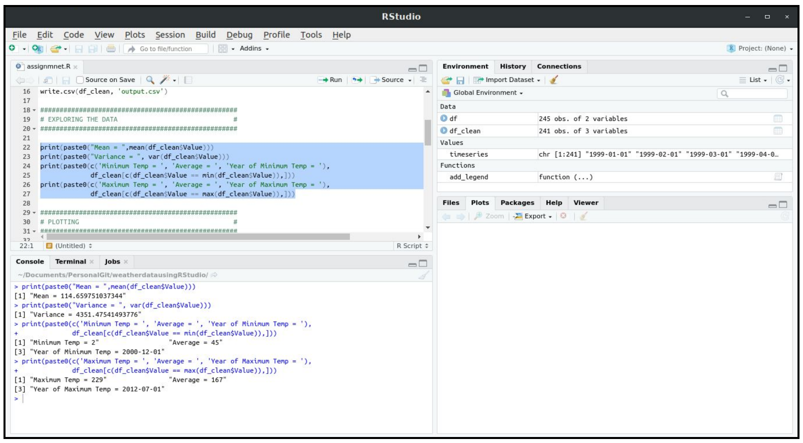
Exploratory data analysis

Exploring the data for:

1. Mean
2. Variance

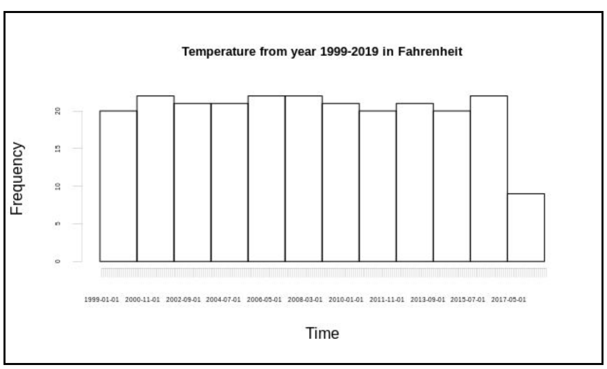
3. Minimum Value
4. Maximum Value

```
print(paste0("Mean = ",mean(df_clean$Value)))
print(paste0("Variance = ", var(df_clean$Value)))
print(paste0(c('Minimum Temp = ', 'Average = ', 'Year of Minimum Temp = '),
         df_clean[c(df_clean$Value == min(df_clean$Value)),]))
print(paste0(c('Maximum Temp = ', 'Average = ', 'Year of Maximum Temp = '),
         df_clean[c(df_clean$Value == max(df_clean$Value)),]))
```

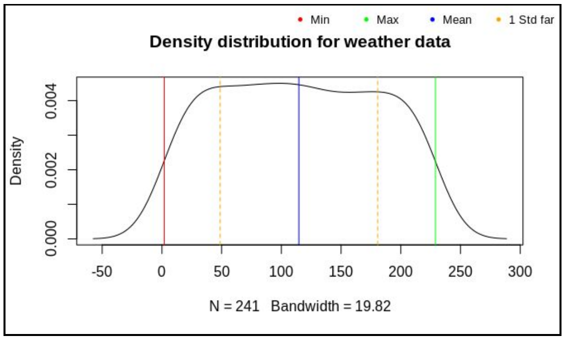


Plotting

```
### HISTOGRAM ###
hist(df_clean$Value,
   xlab = 'Time',
   xaxt='n',
   yaxt='n',
      main = 'Temperature from year 1999-2019 in Fahrenheit',
   cex.main=0.8)
axis(side=1, at=c(1:nrow(df_clean)), labels=c(df_clean$Timeseries),
   lwd=0.1,
   line = NA,
   cex.axis=0.4)
axis(side=2, lwd=0.1,
   cex.axis=0.4)
```



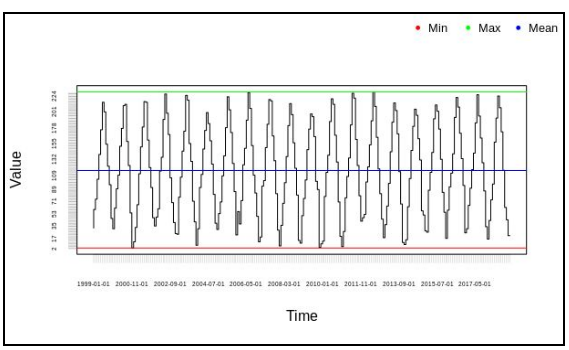

```
### Desnity Plot ###
plot(density(df_clean$Value),
   main = "Density distribution for weather data")
abline(v=mean(df_clean$Value), col='blue')
abline(v=min(df_clean$Value), col='red')
abline(v=max(df_clean$Value), col='green')
abline(v=mean(df_clean$Value) - sd(df_clean$Value), col='orange', lty=2)
abline(v=mean(df_clean$Value) + sd(df_clean$Value), col='orange', lty=2)
add_legend("topright", legend=c("Min", "Max", "Mean", "1 Std far"), pch=20,
      col=c("red", "green", "blue", "orange"),
      horiz=TRUE, bty='n', cex=0.8)
```




```
### Weather Plot ###
plot(c(1:nrow(df_clean)),
    df_clean$Value,
    xlab = 'Time',
    ylab = 'Value',
    type = 'S',
    xaxt = 'n',
    yaxt = 'n')
axis(side=1, at=c(1:nrow(df_clean)), labels=c(df_clean$Timeseries),
    lwd=0.1,
    cex.axis=0.4)
axis(side=2, at=c(1:228), labels=seq(min(df_clean$Value), max(df_clean$Value), 1),
    lwd=0.1,
    cex.axis=0.4)
abline(h=mean(df_clean$Value), col='blue')
abline(h=min(df_clean$Value), col='red')
abline(h=max(df_clean$Value), col='green')

add_legend <- function(...) {
      opar <- par(fig=c(0, 1, 0, 1), oma=c(0, 0, 0, 0),
                  mar=c(0, 0, 0, 0), new=TRUE)
      on.exit(par(opar))
      plot(0, 0, type='n', bty='n', xaxt='n', yaxt='n')
      legend(...)
} ## Reference - https://stackoverflow.com/questions/3932038/plot-a-legend-outside-of-the-plotting-area-in-base-graphics

add_legend("topright", legend=c("Min", "Max", "Mean"), pch=20,
        col=c("red", "green", "blue"),
        horiz=TRUE, bty='n', cex=0.8)
```



One model for modeling weather data is a stochastic model based on the Geometric Brownian motion. You can find more details about this model online.

Mu(μ) - Mean of the drift value i.e the change in the current and previous data divided by the current value and can be calculated using the formula.

$$\frac{1}{n}\sum_{k=1}^{n}\frac{S_K - S_{K=1}}{S_{K+1}}$$

Mu affects the long term movement of the positive mu means positive increment and negative mu indicated negative increment.

Sigma(σ) - It is the standard deviation of the drift value and is given by the formula

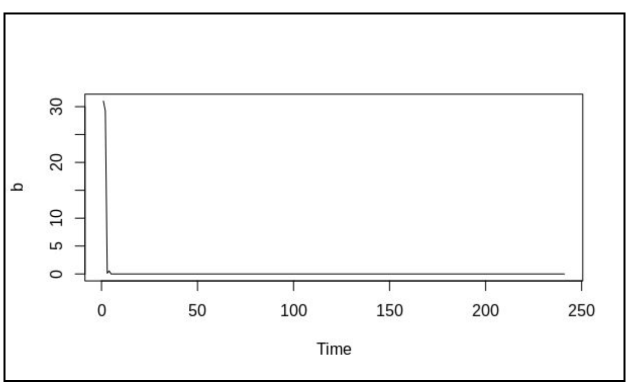
$$\sqrt{\sum_{k=i}^{n}\frac{S_k-\mu^2}{n-1}}$$

Sigma helps us in determining the magnitude of the movement.

```
totaldrift = 0
for (value in c(1:nrow(df_clean)-1)){
   drift = (df_clean[value, 1] - df_clean[value+1, 1])/df_clean[value+1, 1]
   totaldrift = append(totaldrift, drift)
}
totaldrift = totaldrift[2:length(totaldrift)]



output=0
mu = mean(totaldrift)
sigma = sd(totaldrift)
tend = nrow(df_clean)
S1 = df_clean[1,1]
for (t in c(1:tend)){
St = S1 * exp((mu + (sigma**2)/2 ) * t)
output <- append(output, St)
print(paste0(c('Time = ', 'Value = '), c(t, St)))
}
output = output[2:length(output)]
```



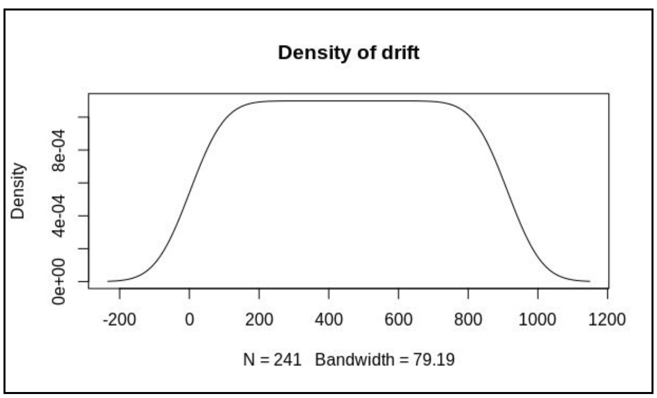

```
> mean(totaldrift)
[1] 0.4086649
> sd(totaldrift)
[1] 2.596726
```

State the condition that weather data have to satisfy in order to be represented by a Geometric Brownian motion.

1. The normality of the log-ratios has a constant mean and variance.
2. log-ratios independent of their past values.
3. Should follow a normal distribution for

$$\text{Log}\left(\frac{S_K}{S_{K+1}}\right)$$

```
logoutput = 0
for(t in c(1:tend)){
  logStS1 = (mu + (sigma**2)/2) * t
  logoutput <- append(logoutput, logStS1)
  print(paste0(c('Time = ', 'Value = '), c(t, logStS1)))
}
logoutput = logoutput[2:length(logoutput)]
```



It does not follow a normal distribution.

Compute values σ and μ for your selected period. Present the formulas used and the results. What is the effect of time on σ and μ? Are the parameters constants (with time)? How could this affect model performance?

1. Mu = 0.4086649
2. Standard deviation = 2.596726

No, the parameters are not constant.

1. Mu affects the long term movement of the positive mu means positive increment and negative mu indicated negative increment.
2. Sigma helps us in determining the magnitude of the movement.

Estimate the expected values of the weather data.

Actual

201902	31.82°F
201903	40.28°F
201904	52.83°F
201905	59.49°F
201906	68.70°F

Predicted

1. 69.08
2. 22.94
3. 11.23
4. 26.96
5. 93.60

Conclusion

Geometric Brownian Motion is useful to plot time series data and it takes into account the changes in data based on time. It uses historical data to judge and predict the next.

References

[1] http://www.sthda.com/english/wiki/abline-r-function-an-easy-way-to-add-straight-lines-to-a-plot-using-r-software

[2] https://stackoverflow.com/questions/3932038/plot-a-legend-outside-of-the-plotting-area-in-base-graphics

[3] https://www.reddit.com/r/AskStatistics/comments/3r7fye/how_can_variance_be_larger_than_a_number_in_the/

[4] http://www.r-tutor.com/elementary-statistics/numerical-measures/standard-deviation

[5] https://www.dummies.com/programming/r/how-to-change-plot-options-in-r/

[6] https://www.datanovia.com/en/lessons/subset-data-frame-rows-in-r/

[7] https://towardsdatascience.com/simulating-stock-prices-in-python-using-geometric-brownian-motion-8dfd6e8c6b18

[8] https://cran.r-project.org/web/packages/somebm/somebm.pdf

[9] http://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=1025&context=imse_pubs