

Problem 1 (6 percent)

Workday 1

Danny Moncada and monca016

February 06, 2020

```
suppressWarnings(suppressPackageStartupMessages({  
  library(TSA)  
  library(ggplot2)  
  library(dplyr)  
  library(forecast)  
}))
```

Simulations

Question 1

Please simulate one sample path of a white noise process of length $T = 200$ and save it into a time series object

- Please use name `e1` for the object that would contain the data

Hints:

1. **Always** specify a seed before generating random numbers. This way your results are reproducible!
2. Use `ts` to convert a vector to a Time-Series

```
# your white noise sample path should be called "e1"  
set.seed(42)  
  
T <- 200L  
  
y <- rnorm(T)  
e1 <- ts(y)
```

Question 2

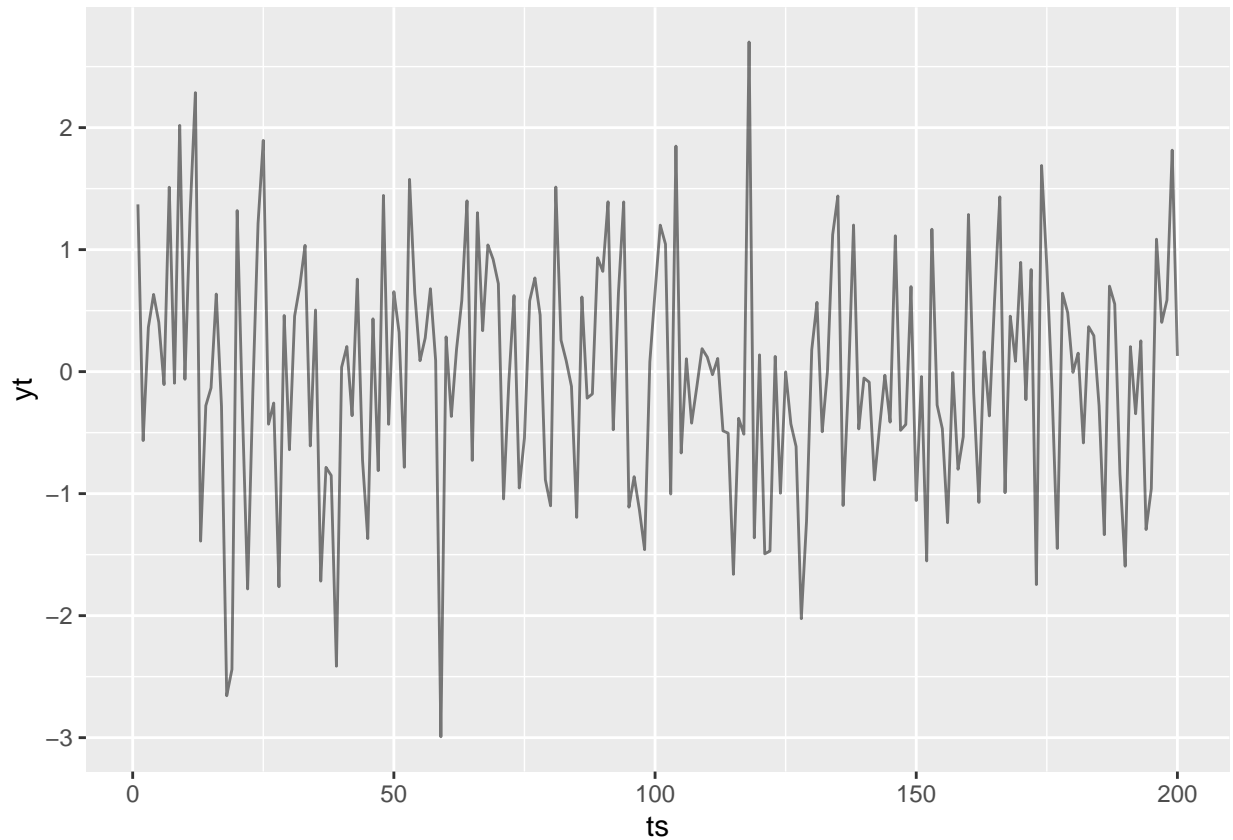
Please plot the sample path e_t

Hints:

1. You can use `time(x)` to extract the time component of time series
2. Use `geom_line` if you are using `ggplot`

```
etplot <- ggplot(data.frame(ts = time(e1), yt = e1), aes(x = ts, y = yt)) + geom_line(alpha = 0.5)
etplot
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```



Question 3

Please compute the mean and standard deviation of that process, add them as horizontal lines to the plot (red for the mean and blue for 2 standard deviations around the mean)

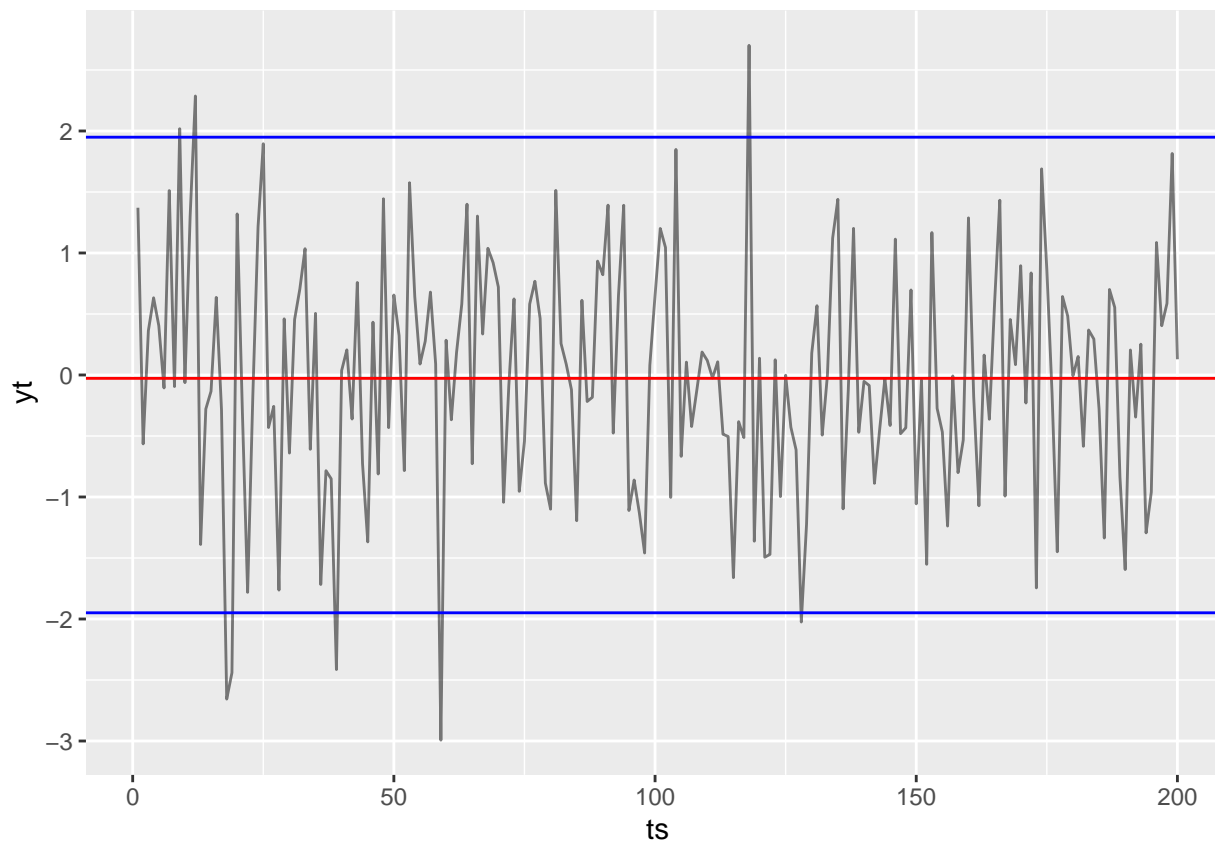
Hints:

- `geom_hline` function may be helpful

```
# please write your code below
mu = mean(e1)
sd = sd(e1)

etplot + geom_hline(yintercept = mu, color = "red") + geom_hline(yintercept = 2*sd, color = "blue") +
  geom_hline(yintercept = -2*sd, color = "blue")
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```



Question 4

Please generate a sample path from a random walk with the length of $T = 200$ and plot it.

- Please use the variable name `Y4` for the array that would contain the sample path.
- Please **do not overwrite** variables created in the previous chunks (such as `e1` that you created earlier)

Hints:

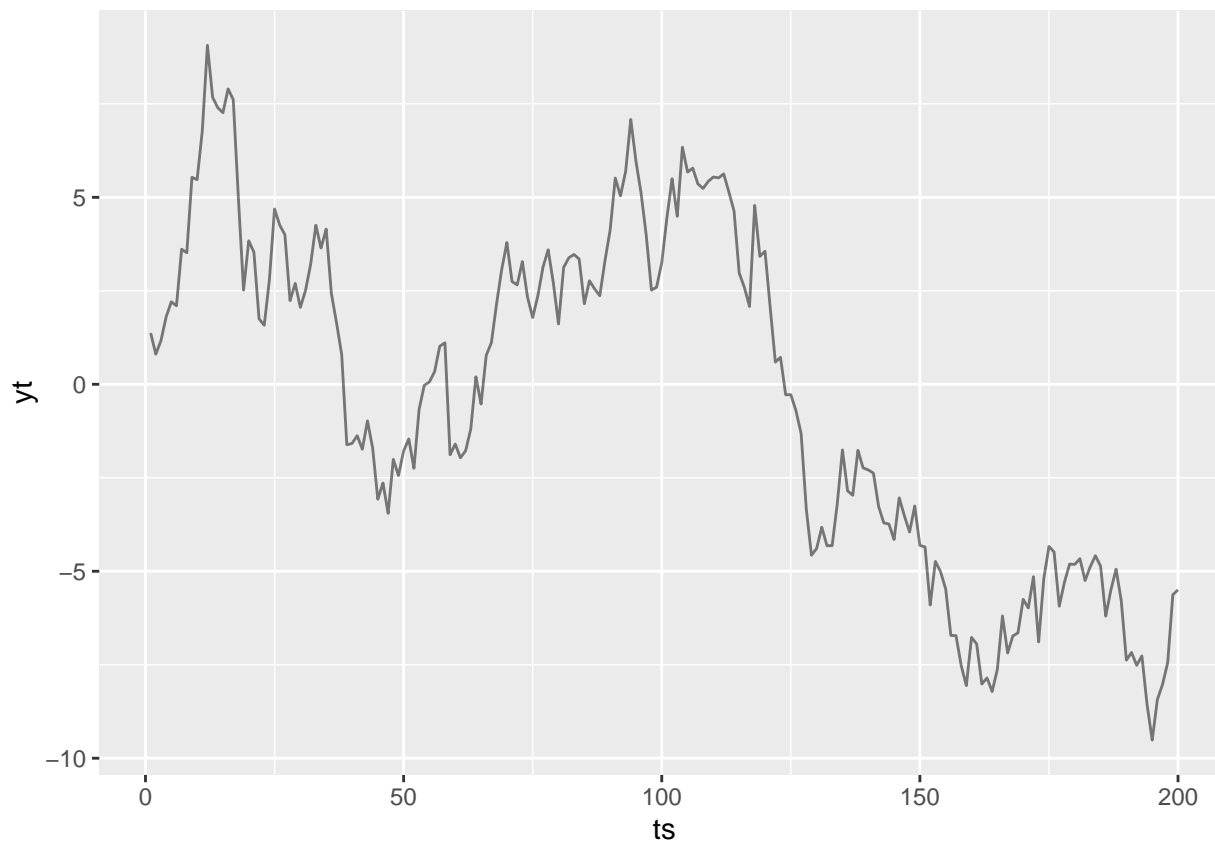
1. Try avoiding using loops. You don't really need a loop here.

```
set.seed(42)

T <- 200L

# please write your code below
Y4 <- cumsum(rnorm(T))

y4plot <- ggplot(data.frame(ts = 1:T, yt = Y4), aes(x = ts, y = yt)) + geom_line(alpha = 0.5)
y4plot
```



Question 5

Please generate 50 sample paths from a random walk with the length of $T = 200$ and plot all of them using different colors.

- Please use the data.frame named **randwalk** that would contain all the sample paths in one long line called column Y. (That is, **randwalk\$Y**)

Hints:

1. Use **dplyr** chains! Try avoiding using for-loops.
2. At this point, you should be using **ggplot**
3. Create the data in the *long* format:
 - I would prefer if you first generate one very long sequence of white noise first, put into a data.frame and then add columns for sample path id and time

```
set.seed(42)

N <- 50L # 50 sample paths
T <- 200L

# please write your code below
randwalk <- data.frame(Y = rep(rnorm(N*T)),
```

```

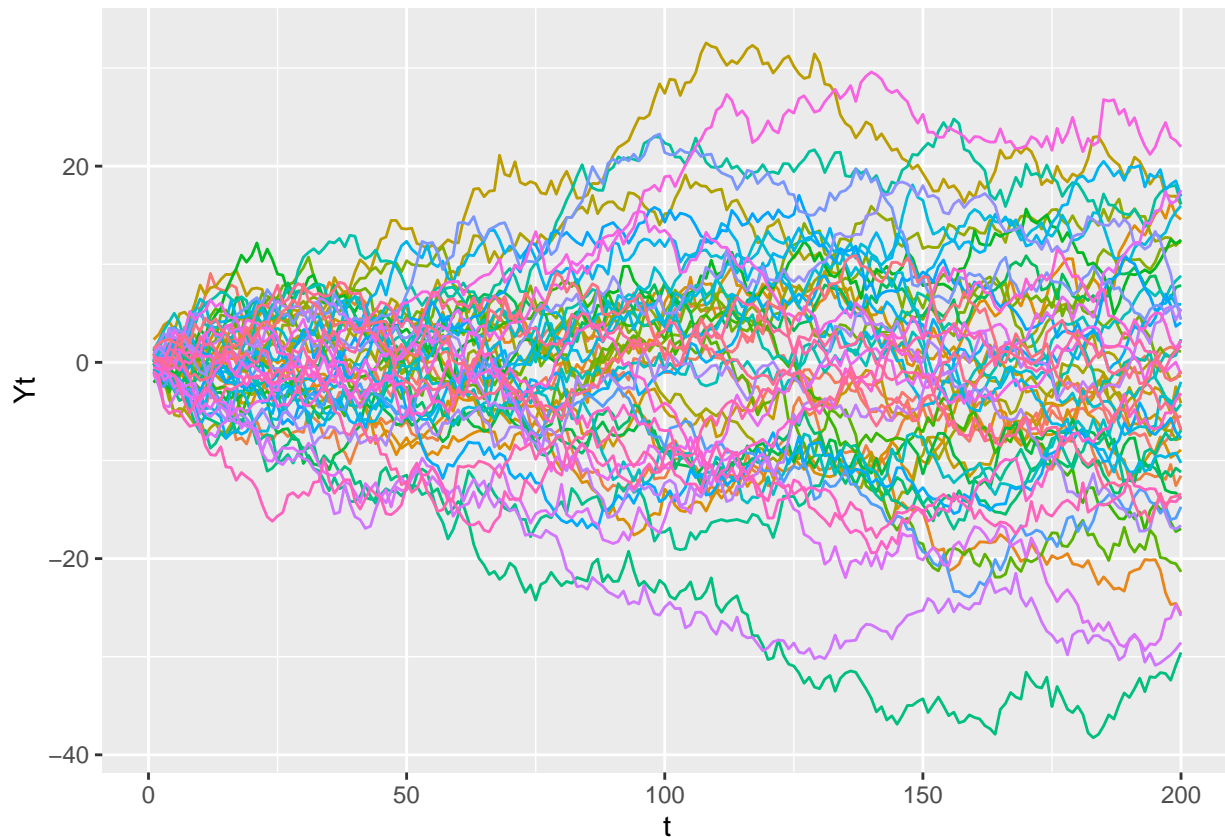
      id = rep(1:N, each = T),
      t = rep(1:T, N))

randwalk$Yt = do.call(c, tapply(randwalk$Y, randwalk$id, FUN = cumsum))

randwalk_plot <- ggplot(data = randwalk, aes(x = t, y = Yt, color = as.factor(id))) +
  geom_line() + theme(legend.position = "none")

randwalk_plot

```



Question 6

Compute the mean function $E[Y_t]$ for every t and add to your plot as well as the variance function $\text{Var}[Y_t]$ for every t and add (standard deviation) to your plot too

Hints:

1. Use `dplyr` chains! Try avoiding using for-loops.
2. Create a separate data.frame with means and standard deviations that will help with extra layers in `ggplot`
3. At this point, you should be using `ggplot`

```

randwalk2 <- randwalk %>%
  group_by(t) %>%

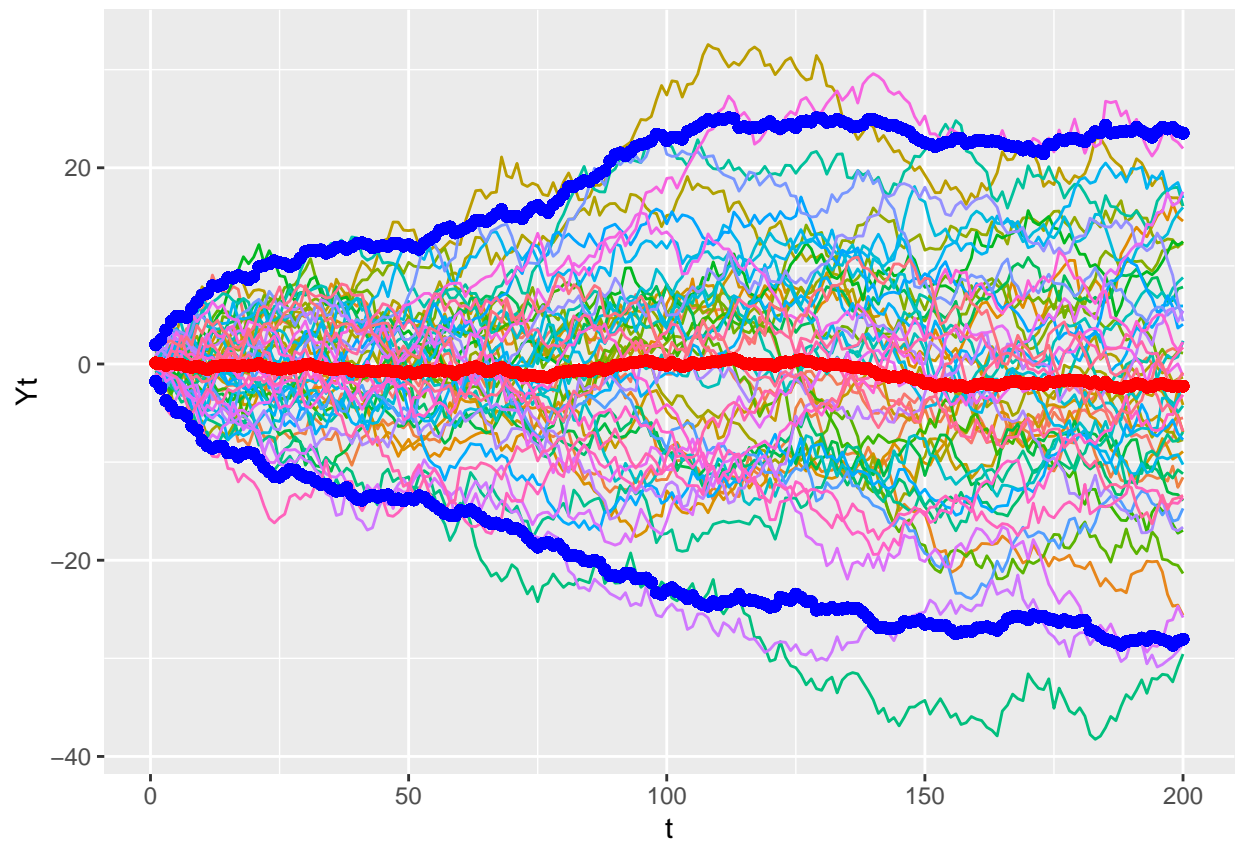
```

```

      mutate(mu = mean(Yt),
             upper = mu + 2*sd(Yt),
             lower = mu - 2*sd(Yt))

randwalk_plot + geom_point(aes(y = randwalk2$mu), color = "red") + geom_point(aes(y = randwalk2$upper),
                                     geom_point(aes(y = randwalk2$lower), color = "blue")

```



Done!

Congratulations!