PSTAT 10 Worksheet 5

Setup

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
set.seed(1)
```

Step 1

```
roll <- function(nroll)
{
    rolls <- sample(c(1,2,3,4,5,6), nroll, replace = TRUE)
    rolls_df <- as_tibble(table(rolls))
    num_less <- sum((rolls_df |> filter(rolls<4))$n)
    num_odd <- sum((rolls_df |> filter((as.integer(rolls) %% 2) != 0))$n)
    if((num_less > nroll/2) && (num_odd > nroll/2))
    {
        return(TRUE)
    }
    else
    {
        return(FALSE)
    }
}
```

Step 2

```
replicates <- replicate(n = 500, roll(10000))
```

Step 3

```
replicates_df <- as_tibble(table(replicates))
num_true <- ((replicates_df |> filter(replicates == TRUE))$n)
print(num_true / sum(replicates_df$n))
```

```
## [1] 0.31
```

This result was much faster and easier to understand than learning the multivariate central limit theorem.

Step 4

```
running_mean <- function(m)
{
    m_replicates <- head(replicates, m)
    m_replicates_df <- (as_tibble(table(m_replicates)))
    num_true <- ((m_replicates_df |> filter(m_replicates == TRUE))$n)
    return(num_true / sum(replicates_df$n))
}

running <- sapply(1:500, running_mean)

x <- (1:500)
plot(x,running[x],type="l", main="Approximation", xlab='replication', ylab='estimate')
abline(h = ((1/4) + (asin(1/3)/(2*pi))), col = "red", lty = 2, lwd = 2)
```