Problem Set 2

Marc Eskew

4/6/2022

Problem 1

We first calculate $Cov(e^{\frac{T}{2}}, e^{\frac{T}{3}})$ using integration:

$$
\begin{aligned}
E[X] &= \int_0^\infty e^{\frac{t}{2}} e^{-t} dt \\
&= 2 \\
E[Y] &= \int_0^\infty e^{\frac{t}{3}} e^{-t} dt \\
&= 1.5 \\
Cov(e^{\frac{T}{2}}, e^{\frac{T}{3}}) &= \int_0^\infty (x - E[X])(y - E[Y] f_T(t) dt \\
&= \int_0^\infty (e^{\frac{t}{2}} - 2)(e^{\frac{t}{3}} - 1.5)(e^{-t}) dt \\
&= 3
\end{aligned}
$$

```r
reps <- c(100,1000,10000,100000)

for(i in reps) {

  t_vec <- rexp(i)

  df <- data.frame(t = t_vec) %>%
    mutate(t_1 = exp(t/2),
           t_2 = exp(t/3), t_3 = t_1*t_2)

  cov_est <- mean(df$t_3) - mean(df$t_1)*mean(df$t_2)

  print(paste(i,":",cov_est))

}
```

```
## [1] "100 : 1.73827704359974"
## [1] "1000 : 1.29516948130332"
## [1] "10000 : 2.30797425244795"
## [1] "1e+05 : 2.68740957870294"
```

Based on these experiments, I do not think this converges at a speed proportional to $\frac{1}{\sqrt{N}}$. This does not contradict our discussion, but what fails here is the consistency of determining the variance of one replication of the simulation.

Problem 2

```
x <- 2*runif(10000,0,1)
y <- 2*runif(10000,0,1)

df <- data.frame(x=x, y=y) %>%
  mutate(i = ifelse(y<=x,1,0)
  ) %>%
  mutate(g = sin(x*y)*i)

est <- 4*mean(df$g)
print(est)
```

```
## [1] 1.021677
```

The MC estimation over this simulation run was **1.0216771**. This is near the solution of ~1.05 determined through calculus.

Problem 3

With a $\lambda(t)$ that increases linearly with time, I would expect the component modeled as an exponential would last longer.

Using the inversion method to sample T with $\lambda(t) = 2t + 1$:

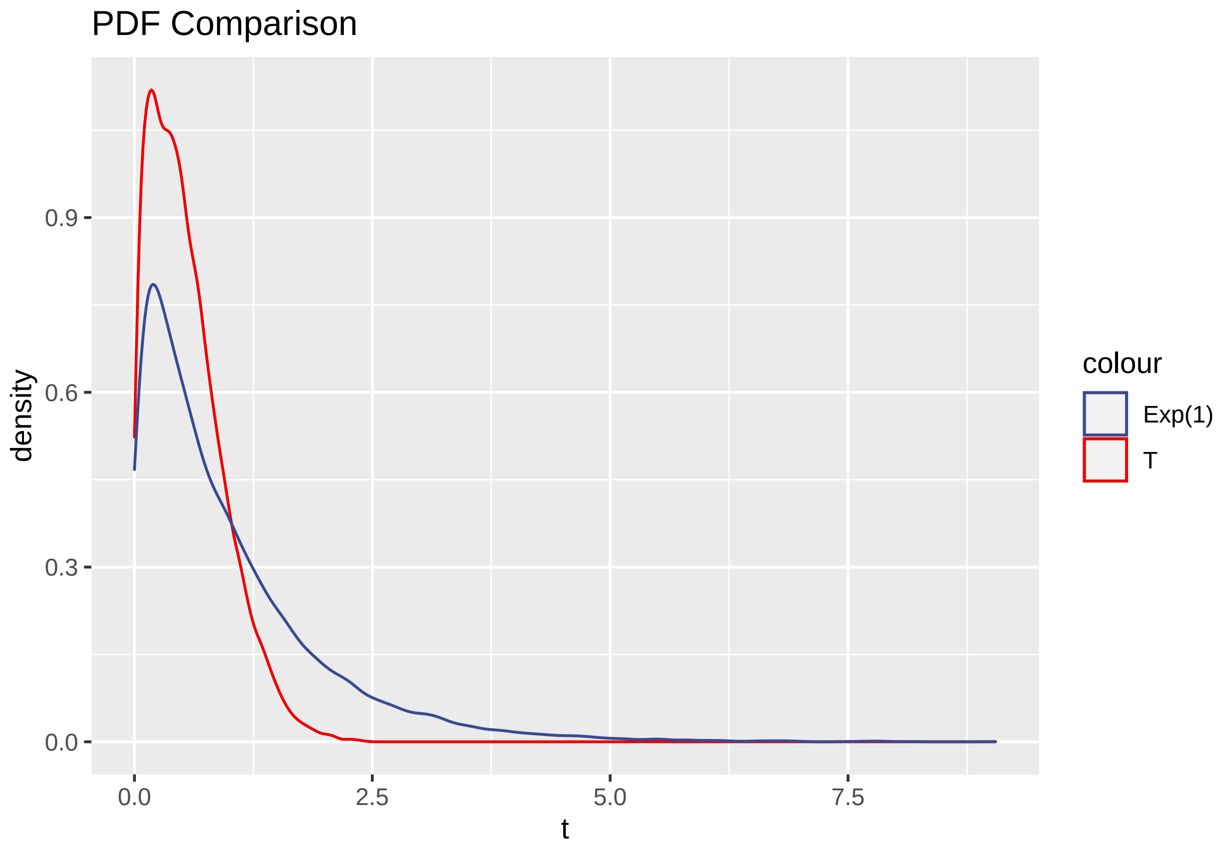
$$\begin{aligned}
\Lambda(t) &= \int_0^t 2t + 1\, dt \\
&= t^2 + t \\
F_Y(y) &= u \\
u &= 1 - e^{-t^2-t} \\
\frac{1}{1-u} &= e^{t^2+t} \\
log(\frac{1}{1-u}) &= t^2 + t \\
log(\frac{1}{1-u}) + \frac{1}{4} &= (t + \frac{1}{2})^2 \\
\sqrt{log(\frac{1}{1-u}) + \frac{1}{4}} &= t + \frac{1}{2} \\
\sqrt{log(\frac{1}{1-u}) + \frac{1}{4}} - \frac{1}{2} &= t
\end{aligned}$$

With the cumulative function inverted, we can sample from a $U(0, 1)$ distribution to represent T.

```
exp_life <- rexp(10000) # Exponential comparison

T_life <- data.frame(u = runif(10000)) %>% # Sample from U(0,1)
  mutate(t = sqrt(.25 + log(1/(1-u)))-.5) # Apply inverse function

ggplot(T_life) +
  geom_density(aes(x = t, color = "T")) +
  geom_density(data = data.frame(x=exp_life),aes(x=x,color = "Exp(1)")) +
  ggsci::scale_color_aaas() +
  ggtitle("PDF Comparison")
```



The mean lifetime when modeled by T is **0.5405838**, which is lower than the exponential model.

Problem 4

a. $\mathbb{E}[N^2]$

```
df_out <- data.frame()

for(i in 1:10000) {
  n <- -1
  p <- 1

  while(p > exp(-9)) {
    p <- p*runif(1)^3
    n <- n+1
  }

  df_out <- rbind(df_out,n)
}

colnames(df_out) <- "N"
mean(df_out$N^2)
```

```
## [1] 11.8709
```

b.$\mathbb{P}[N = i]$

```r
for(i in 0:6) {
  df_out1 <- df_out %>%
    filter(N == i)

  print(paste(i,":",nrow(df_out1)/nrow(df_out)))
}
```

```
## [1] "0 : 0.051"
## [1] "1 : 0.1459"
## [1] "2 : 0.2279"
## [1] "3 : 0.2239"
## [1] "4 : 0.1637"
## [1] "5 : 0.1078"
## [1] "6 : 0.0495"
```