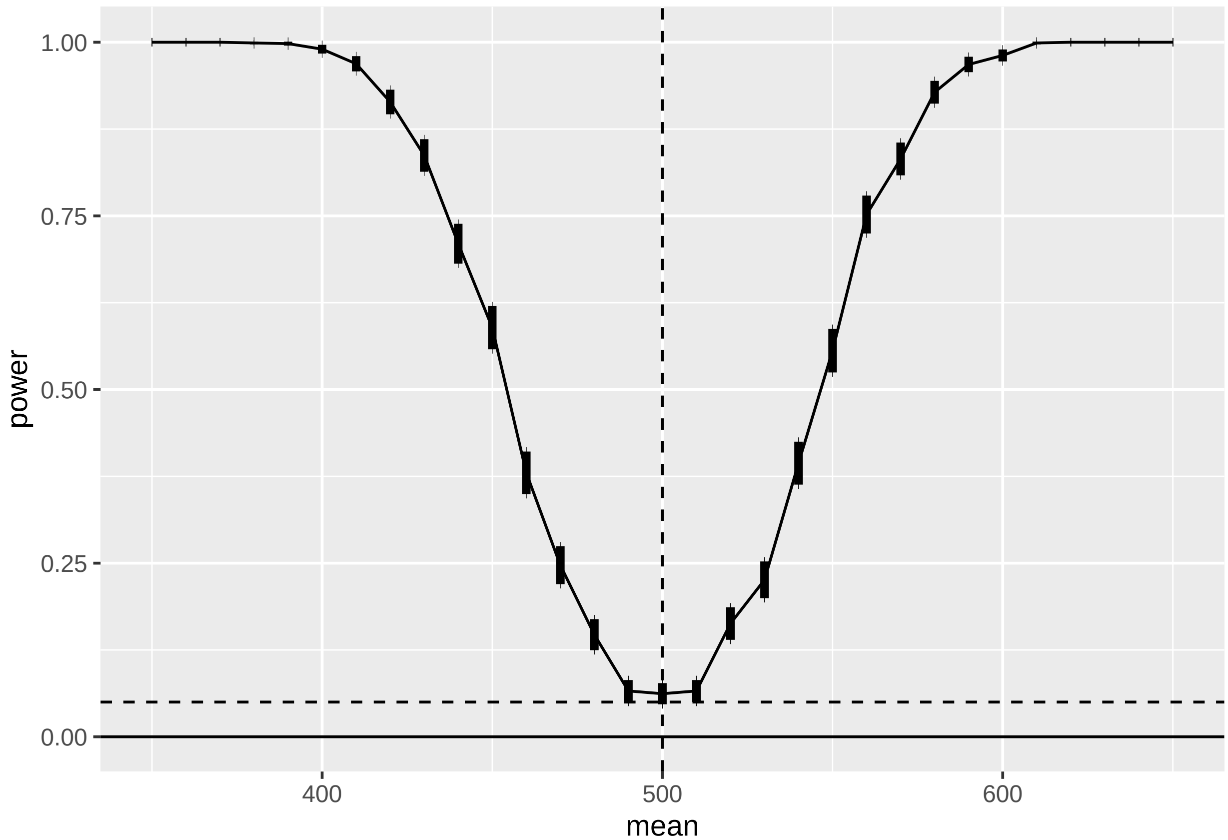
Homework 8

11/05/2021

Question 1 (7.2)

Plot the empirical power curve for the $t - test$ in *Example* 7.9, changing the alternative hypothesis to $H_1 : \mu \neq 500$, and keeping the significance level $\alpha = 0.05$.

```
library(ggplot2)

n <- 20
m <- 1000
mu0 <- 500
sigma <- 100
mu <- seq(350,650,10) #alternatives
M <- length(mu)
power <- numeric(M)
for( i in 1:M ){
  mu1 <- mu[i]
  pvalues <- replicate(m, expr = {
    #simulate under alternative mu1
    x <- rnorm(n,mean=mu1,sd=sigma)
    ttest <- t.test(x,alternative="two.sided",mu=mu0)
    ttest$p.value
  })
  power[i] <- mean(pvalues <= .05)
}
se <- sqrt(power*(1-power)/m)

# plot the empirical power curve
# adding vertical error bars at pi(theta) +/- 2se(pi(theta))

df <- data.frame(mean=mu, power=power, upper=power+2*se, lower=power-2*se)
ggplot(df,aes(x=mean,y=power)) +
  geom_line()+
  geom_vline(xintercept=500,lty=2) +
  geom_hline(yintercept=c(0,.05),lty=1:2) +
  geom_errorbar(aes(ymin=lower, ymax=upper), width=0.2, lwd=1.5)
```



Question 2 (7.3)

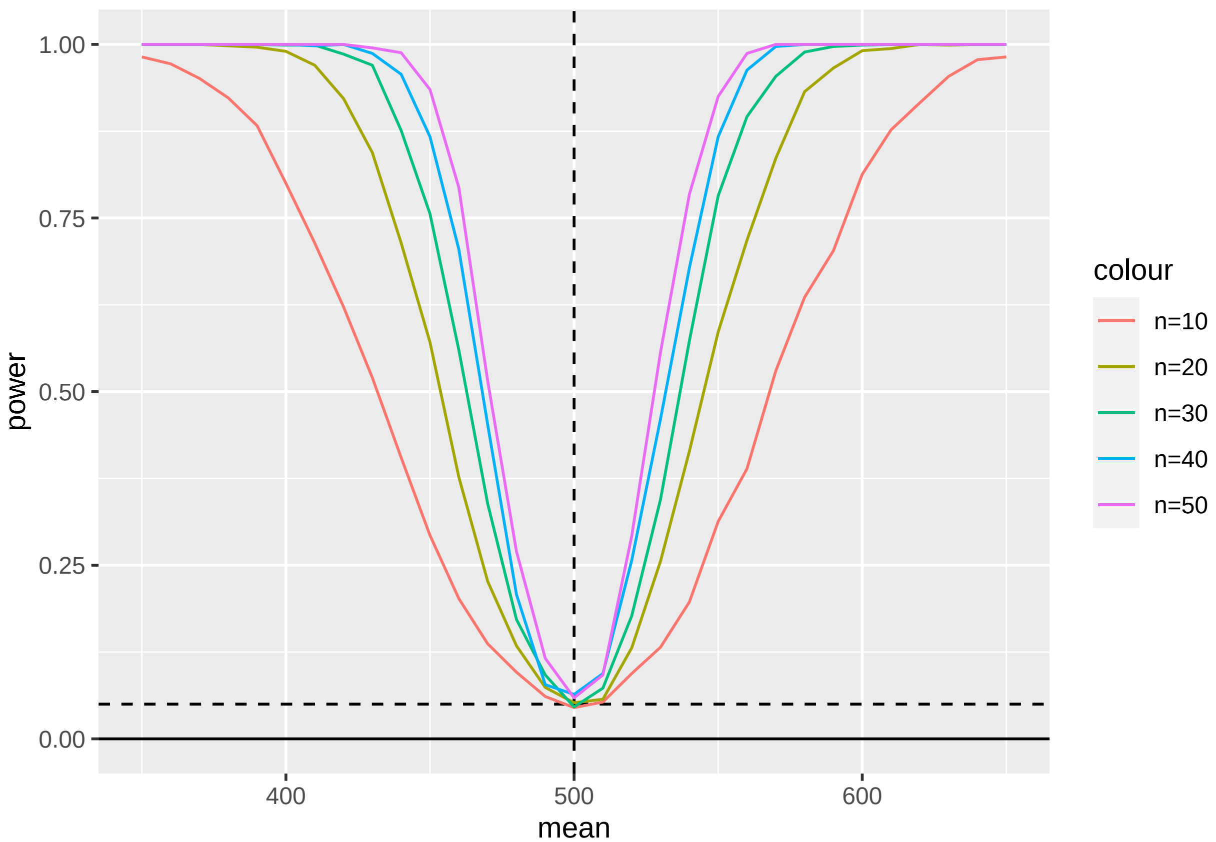
Plot the power curves for the $t-test$ in *Example* 7.9 for sample sizes 10, 20, 30, 40, and 50, but omit the standard error bars. Plot the curves on the same graph, each in a different color or different line type, and include a legend. Comment on the relation between power and sample size.

```
m <- 1000
mu0 <- 500
sigma <- 100
mu <- seq(350,650,10)
M <- length(mu)

n <- 10
power1 <- numeric(M)
for( i in 1:M ){
  mu1 <- mu[i]
  pvalues <- replicate(m, expr = {
    x <- rnorm(n,mean=mu1,sd=sigma)
    ttest <- t.test(x,alternative="two.sided",mu=mu0)
    ttest$p.value
  })
  power1[i] <- mean(pvalues <= .05)
}
df1 <- data.frame(mean=mu, power=power1)
```

```r
n <- 20
power2 <- numeric(M)
for( i in 1:M ){
  mu1 <- mu[i]
  pvalues <- replicate(m, expr = {
    x <- rnorm(n,mean=mu1,sd=sigma)
    ttest <- t.test(x,alternative="two.sided",mu=mu0)
    ttest$p.value
  })
  power2[i] <- mean(pvalues <= .05)
}
df2 <- data.frame(mean=mu, power=power2)


n <- 30
power3 <- numeric(M)
for( i in 1:M ){
  mu1 <- mu[i]
  pvalues <- replicate(m, expr = {
    x <- rnorm(n,mean=mu1,sd=sigma)
    ttest <- t.test(x,alternative="two.sided",mu=mu0)
    ttest$p.value
  })
  power3[i] <- mean(pvalues <= .05)
}
df3 <- data.frame(mean=mu, power=power3)


n <- 40
power4 <- numeric(M)
for( i in 1:M ){
  mu1 <- mu[i]
  pvalues <- replicate(m, expr = {
    x <- rnorm(n,mean=mu1,sd=sigma)
    ttest <- t.test(x,alternative="two.sided",mu=mu0)
    ttest$p.value
  })
  power4[i] <- mean(pvalues <= .05)
}
df4 <- data.frame(mean=mu, power=power4)


n <- 50
power5 <- numeric(M)
for( i in 1:M ){
  mu1 <- mu[i]
  pvalues <- replicate(m, expr = {
    x <- rnorm(n,mean=mu1,sd=sigma)
    ttest <- t.test(x,alternative="two.sided",mu=mu0)
    ttest$p.value
  })
  power5[i] <- mean(pvalues <= .05)
}
```

```r
df5 <- data.frame(mean=mu, power=power5)
```

```r
ggplot() +
  geom_vline(xintercept=500,lty=2) +
  geom_hline(yintercept=c(0,.05),lty=1:2) +
  geom_line(data=df1, aes(x=mean, y=power, color='n=10'))+
  geom_line(data=df2, aes(x=mean, y=power, color='n=20'))+
  geom_line(data=df3, aes(x=mean, y=power, color='n=30'))+
  geom_line(data=df4, aes(x=mean, y=power, color='n=40'))+
  geom_line(data=df5, aes(x=mean, y=power, color='n=50'))
```



-Ans: As sample size increases, the power of the test increases closer to the mean value and approaches 1 at a faster rate.

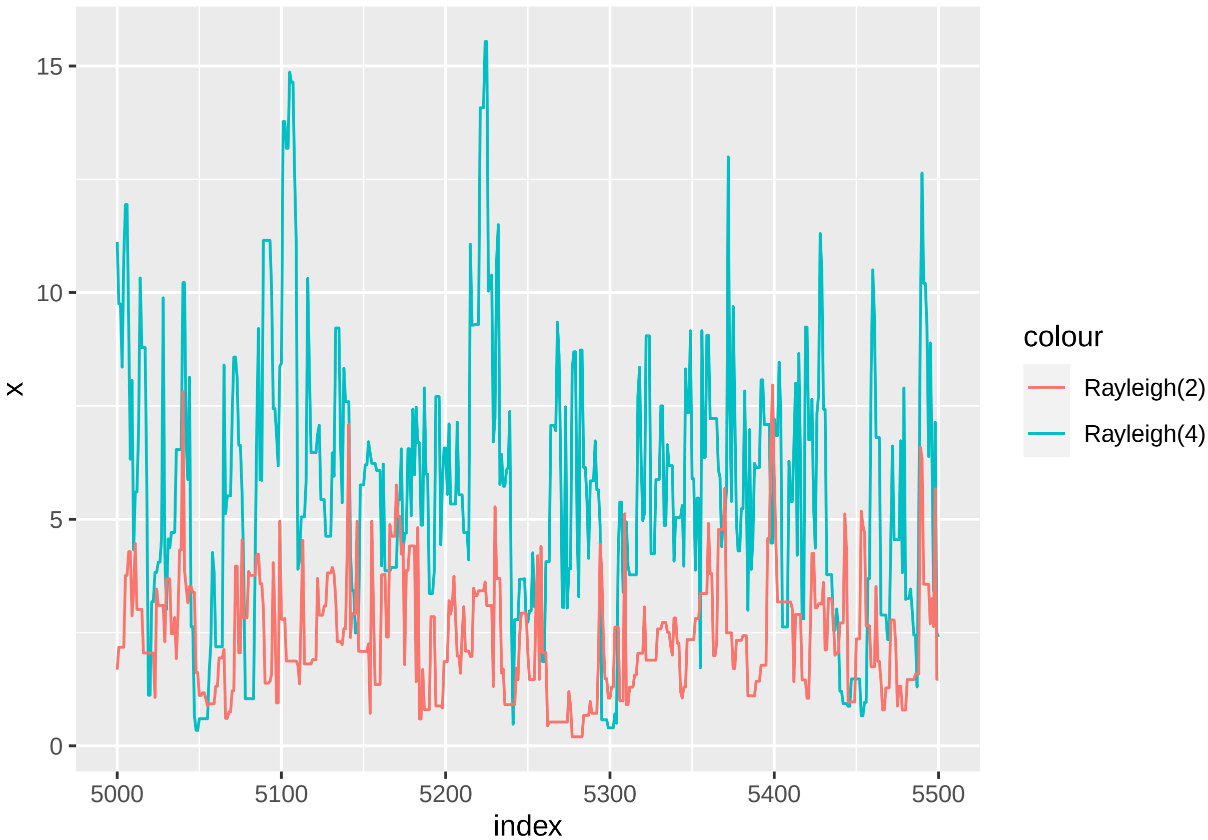
Question 3 (11.1)

Repeat *Example* 11.1 for the target distribution $Rayleigh(\sigma = 2)$. Compare the performance of the $Metropolis - Hastings$ sampler for *Example* 11.1 and this problem. In particular, what differences are obvious from the plot corresponding to *Figure* 11.1?

```r
set.seed(888)
f <- function(x,sigma){
  if(any(x<0)) return(0)
  stopifnot(sigma >0 )
```

```r
  return((x/sigma^2)*exp(-x^2/(2*sigma^2)))
}

m <- 10000
sigma1 <- 2
x1 <- numeric(m)
x1[1] <- rchisq(1,df=1)
k1 <- 0
u <- runif(m)
for(i in 2:m){
  xt <- x1[i-1]
  y <- rchisq(1,df=xt)
  num <- f(y,sigma1)*dchisq(xt,df=y)
  den <- f(xt,sigma1)*dchisq(y,df=xt)
  if (u[i]<= num/den) x1[i] <- y
  else {
    x1[i] <- xt
    k1 <- k1+1 # y is rejected
  }
}

#EXAMPLE 11.1
sigma2 <- 4
x2 <- numeric(m)
x2[1] <- rchisq(1,df=1)
k2 <- 0
u <- runif(m)
for(i in 2:m){
  xt <- x2[i-1]
  y <- rchisq(1,df=xt)
  num <- f(y,sigma2)*dchisq(xt,df=y)
  den <- f(xt,sigma2)*dchisq(y,df=xt)
  if (u[i]<= num/den) x2[i] <- y
  else {
    x2[i] <- xt
    k2 <- k2+1 # y is rejected
  }
}
```

```r
index <- 5000:5500
y1 <- x1[index]
y2 <- x2[index]
df_y <- data.frame(index, y1,y2)
ggplot() +
  geom_line(data=df_y, aes(x=index, y=y2, color='Rayleigh(4)'))+
  geom_line(data=df_y, aes(x=index, y=y1, color='Rayleigh(2)'))+
  ylab("x")
```



-Ans: We can see right away that in this exercise (red) the values appear to be lower and with significantly less variability.

Question 4 (11.2)

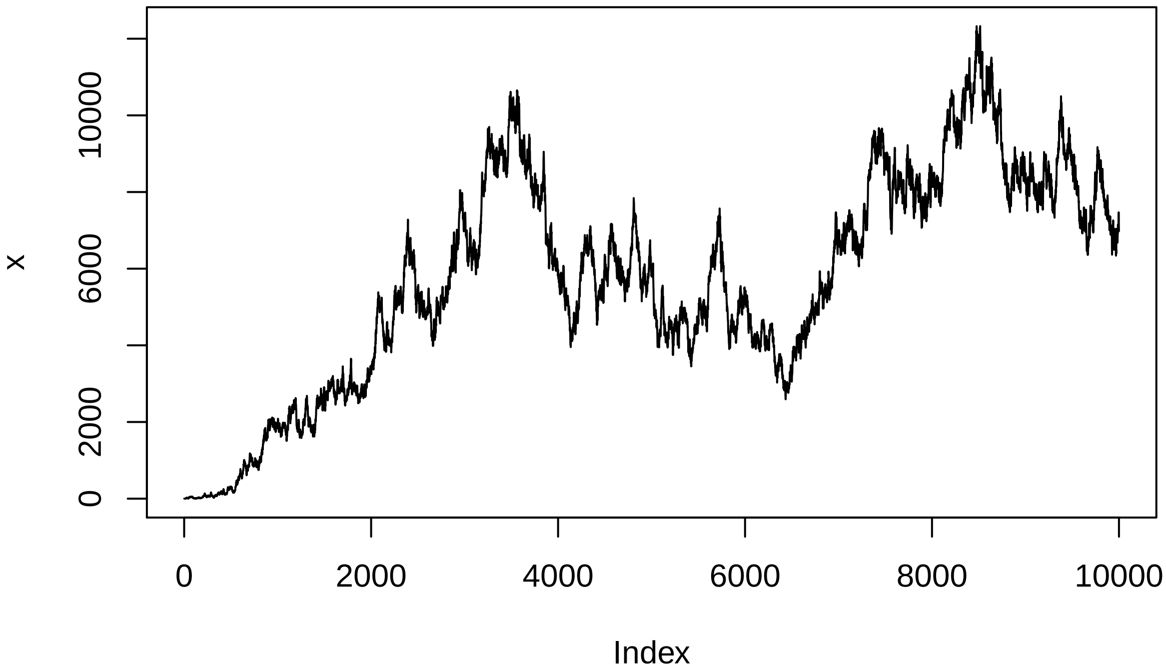
Repeat *Example* 11.1 using the proposal distribution $Y \sim Gamma(X_t, 1)$ (shape parameter X_t and rate parameter 1).

```
#EXAMPLE 11.1
m <- 10000
rate <- 1
x <- numeric(m)
x[1] <- rchisq(1,df=1)
k <- 0
u <- runif(m)

for(i in 2:m){
  xt <- x[i-1]
  y <- rchisq(1,df=xt)
  num <- rgamma(1,y,rate)*dchisq(xt,df=y)
  den <- rgamma(1,xt,rate)*dchisq(y,df=xt)
  if (u[i]<= num/den) x[i] <- y
  else {
    x[i] <- xt
    k <- k+1 # y is rejected
  }
```

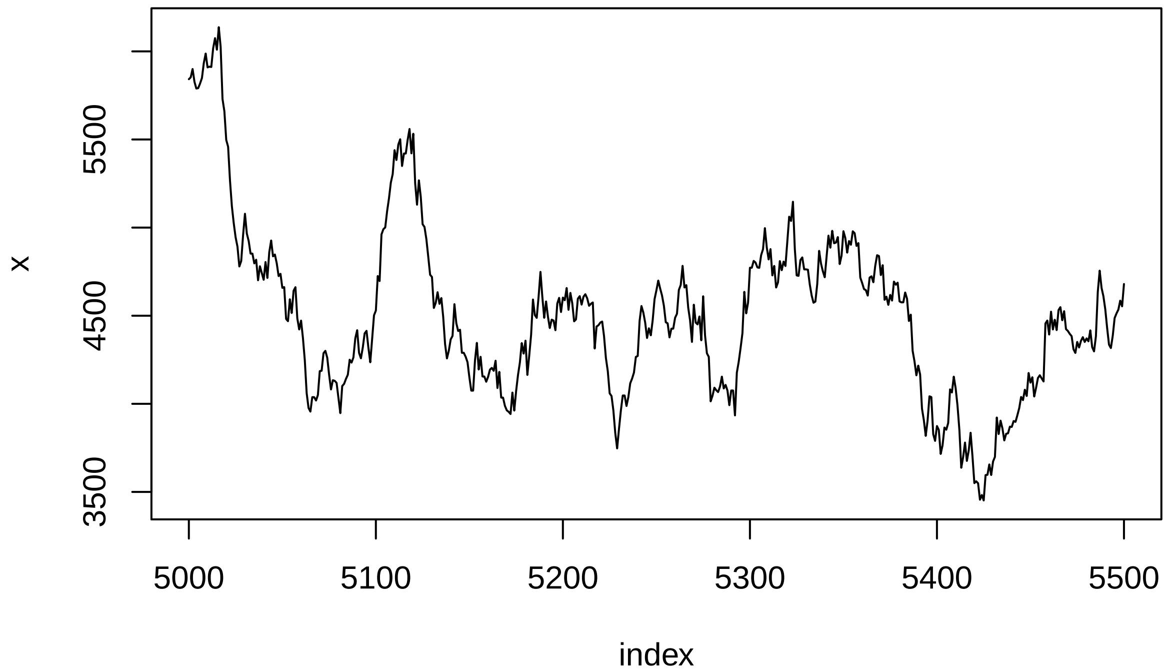
```r
}

# Check how many y rejected
print(k)
```

```
## [1] 253
```

```r
# Plot the Markov chain x
plot(x, type="l")
```




```r
# display a partial plot starting at index 5000:5500
index <- 5000:5500
y1 <- x[index]
plot(index,y1,type="l", main="", ylab="x")
```


x
3500
4500
5500
5000
5100
5200
5300
5400
5500
index