

PSWR

Sucheta

30/11/2021

```
#Solution 1.a)

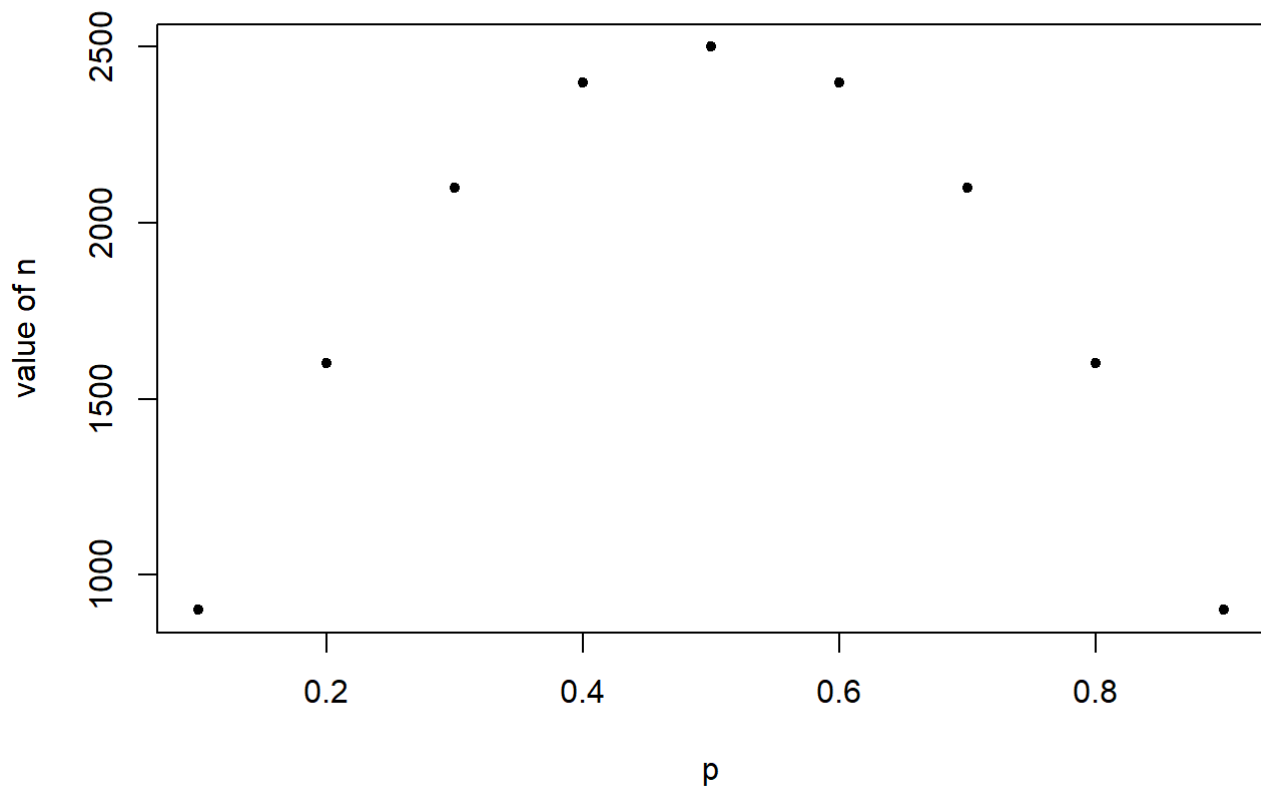
var_phat = function(p,n){
  return(p*(1-p)/n)
}

sd_phat = function(p,n){
  return((p*(1-p)/n)^0.5)
}

#solution 1.b)

#SD(phat) = 0.01 = (p(1-p)/n)^0.5
#So, (0.01)^2 = p(1-p)/n
#Therefore, n = p(1-p)/(0.01)^2

sam_n = function(p){
  return(p*(1-p)/(0.01)^2)
}
prob = c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9)
n_prob = rep(NA,length(prob))
for (i in 1:length(prob)){
  n_prob[i]=sam_n(prob[i])
}
plot(prob,n_prob,ylab='value of n', xlab='p',pch=20)
```



```
#solution 1.c)
```

#phat depends on the occurrence and non occurrence of an event A. Let 1 denote the occurrence(probability = p) and 0 denote the non occurrence(probability = 1-p).

```
x = c(0,1)
```

```
phat = function(sim_size,p,n){
  phat_sam = rep(NA,sim_size)
  for (i in 1:sim_size){
    z = sample(x,n,replace=TRUE,prob=c(1-p,p))
    count = sum(z==1)
    phat_sam[i]=count/n
  }
  return(phat_sam)
}
```

```
#Solution c.i)
```

```
n=500
sim_size = 1000
p=c(0.01,0.1, 0.25, 0.5, 0.75, 0.9, 0.99)
sd1 = rep(NA,length(p))
for (i in 1:length(p)){
  phat1 = phat(sim_size,p[i],n)
  sd1[i]=sd(phat1)
}
sd1
```

```
## [1] 0.004598524 0.013361047 0.019493896 0.022220503 0.019420617 0.013556824  
## [7] 0.004459932
```

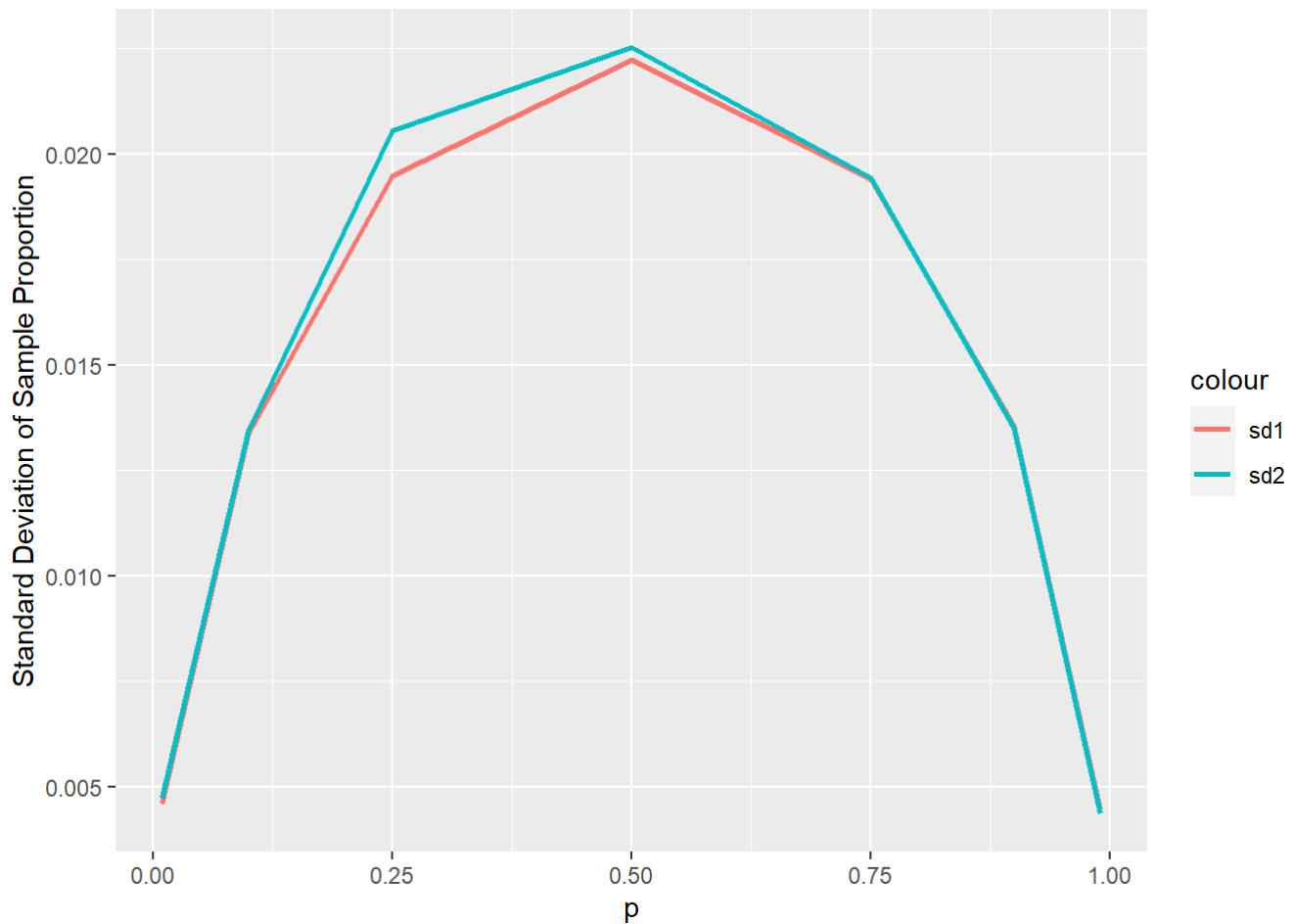
```
#Solution1.c.ii)  
n_sd = function(p,sd){  
  return(p*(1-p)/(sd)^2)  
}  
  
sd2 = rep(NA,length(p))  
for (i in 1:length(p)){  
  n2 = n_sd(p[i],sd1[i])  
  phat2 = phat(sim_size,p[i],n2)  
  sd2[i]=sd(phat2)  
}  
sd2
```

```
## [1] 0.004742179 0.013436129 0.020559681 0.022527167 0.019427393 0.013510451  
## [7] 0.004351914
```

```
df = data.frame(p,sd1)  
cbind(df,sd2)
```

```
##      p      sd1      sd2  
## 1 0.01 0.004598524 0.004742179  
## 2 0.10 0.013361047 0.013436129  
## 3 0.25 0.019493896 0.020559681  
## 4 0.50 0.022220503 0.022527167  
## 5 0.75 0.019420617 0.019427393  
## 6 0.90 0.013556824 0.013510451  
## 7 0.99 0.004459932 0.004351914
```

```
library(ggplot2)  
ggplot(df,aes(x=p))+geom_line(aes(y=sd1,colour='sd1'),size=1)+  
  geom_line(aes(y=sd2,colour='sd2'),size=1)+  
  ylab("Standard Deviation of Sample Proportion")
```



#Solution 2.b and 2.c)

```
n = c(100,500,1000)
mean=c()
var = c()
diff_mean = c()
diff_var=c()
set.seed(11)
df_pois = function(lambda){
  for (j in n){
    pois = rpois(j,lambda)
    mean = append(mean,mean(pois))
    var = append(var,var(pois))
    diff_mean = append(diff_mean,lambda-mean(pois))
    diff_var=append(diff_var,lambda-var(pois))
  }
  return(data.frame(n,mean,var,diff_mean,diff_var))
}

#lambda = 10
df_pois(10)
```

```
##      n  mean      var diff_mean  diff_var
## 1  100  9.540  8.473131    0.460  1.5268687
## 2  500 10.116 10.736016   -0.116 -0.7360160
## 3 1000  9.943  9.577328    0.057  0.4226717
```

```
#As the value of n is increasing the difference between the sample mean and true
#mean and the sample variance and true variance is decreasing.
#we can see as we are double the value of n, our error is getting almost halved.
true_mean = 10
true_var = 10
mean_sam = mean(df_pois(10)$mean)
mean_var = mean(df_pois(10)$var)
print(paste("The expectation of sample mean is : ",mean_sam))
```

```
## [1] "The expectation of sample mean is : 9.90966666666667"
```

```
print(paste("The expectation of sample variance is : ",mean_var))
```

```
## [1] "The expectation of sample variance is : 10.3539952196078"
```

```
#The expectation of sample mean and variance, differ by 0.1 and
#0.35 respectively from the true mean = 10
```

```
#Lambda = 20
df_pois(20)
```

```
##      n   mean      var diff_mean  diff_var
## 1  100 19.680 20.74505    0.320 -0.7450505
## 2   500 19.834 20.19884    0.166 -0.1988417
## 3 1000 20.134 20.67272   -0.134 -0.6727167
```

```
#As the value of n increases the difference between the sample mean and true
#mean and the sample variance and true variance decreases at first step but then
#again increases. But overall the difference is reduced on increasing n.
```

```
true_mean = 20
true_var = 20
mean_sam = mean(df_pois(20)$mean)
mean_var = mean(df_pois(20)$var)
print(paste("The expectation of sample mean is : ",mean_sam))
```

```
## [1] "The expectation of sample mean is : 19.9036666666667"
```

```
print(paste("The expectation of sample variance is : ",mean_var))
```

```
## [1] "The expectation of sample variance is : 19.737532153639"
```

```
#The expectation of sample mean and variance, differ by 0.1 and
#0.36 respectively from the true mean = 20
```

```
#Lambda = 50
df_pois(50)
```

```
##      n    mean      var diff_mean  diff_var
## 1  100 49.530 45.94859    0.470  4.0514141
## 2   500 50.392 49.87408   -0.392  0.1259158
## 3  1000 50.002 51.04304   -0.002 -1.0430390
```

#As the value of n increases the difference between the sample mean and true mean and the sample variance and true variance decreases initially. But, then it increases very largely, indicating not to increase n for lambda=50.

```
true_mean = 50
true_var = 50
mean_sam = mean(df_pois(50)$mean)
mean_var = mean(df_pois(50)$var)
print(paste("The expectation of sample mean is : ",mean_sam))
```

```
## [1] "The expectation of sample mean is : 49.7466666666667"
```

```
print(paste("The expectation of sample variance is : ",mean_var))
```

```
## [1] "The expectation of sample variance is : 53.1424211962615"
```

#The expectation of sample mean and variance, differ by 0.26 and 3.14 respectively from the true mean = 50

#We can conclude that for larger values of lambda, increasing the value of n increases the difference between the sample mean and true mean and the sample variance and true variance. We also observe, The expectation of sample mean and variance are close to lambda.