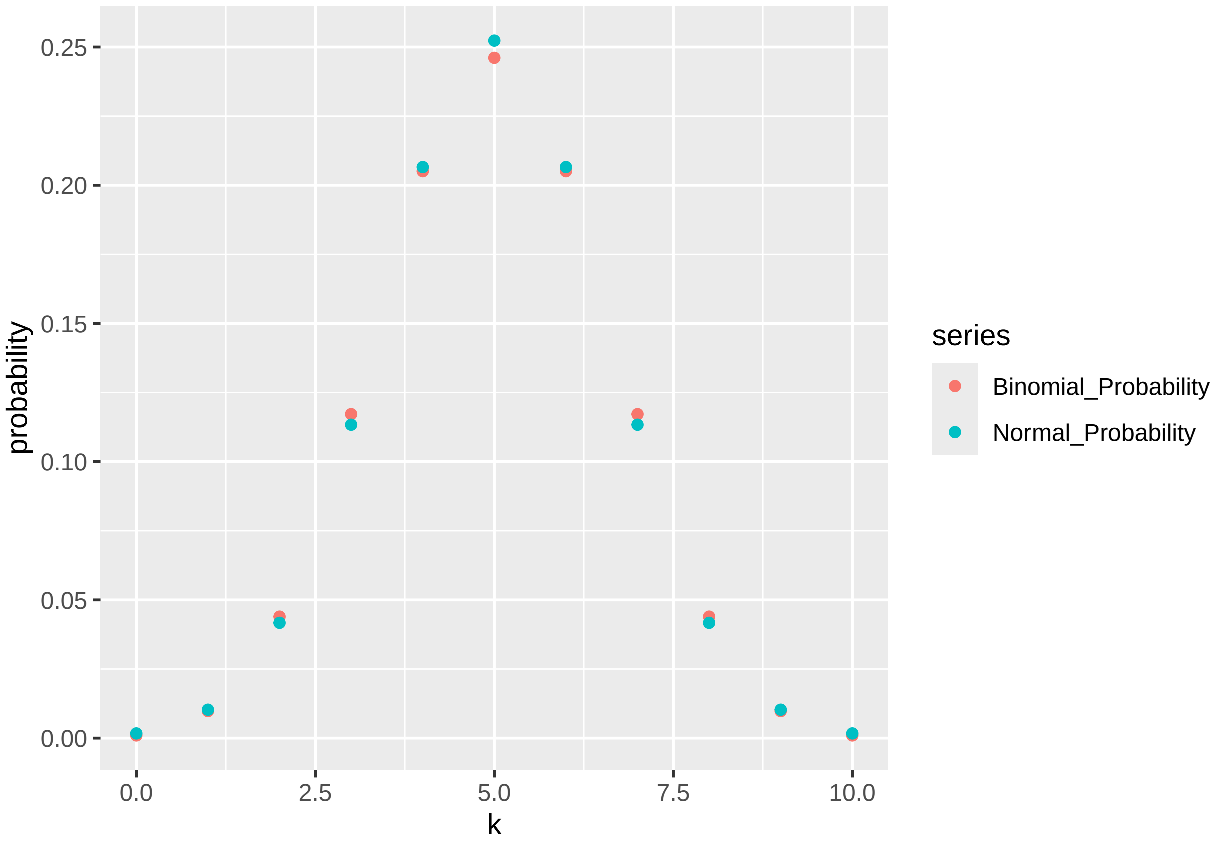
computer_proj1

Harrison Halesworth

2025-09-08

```r
# Library imports for experiment
library(ggplot2)
library(tidyr)
```

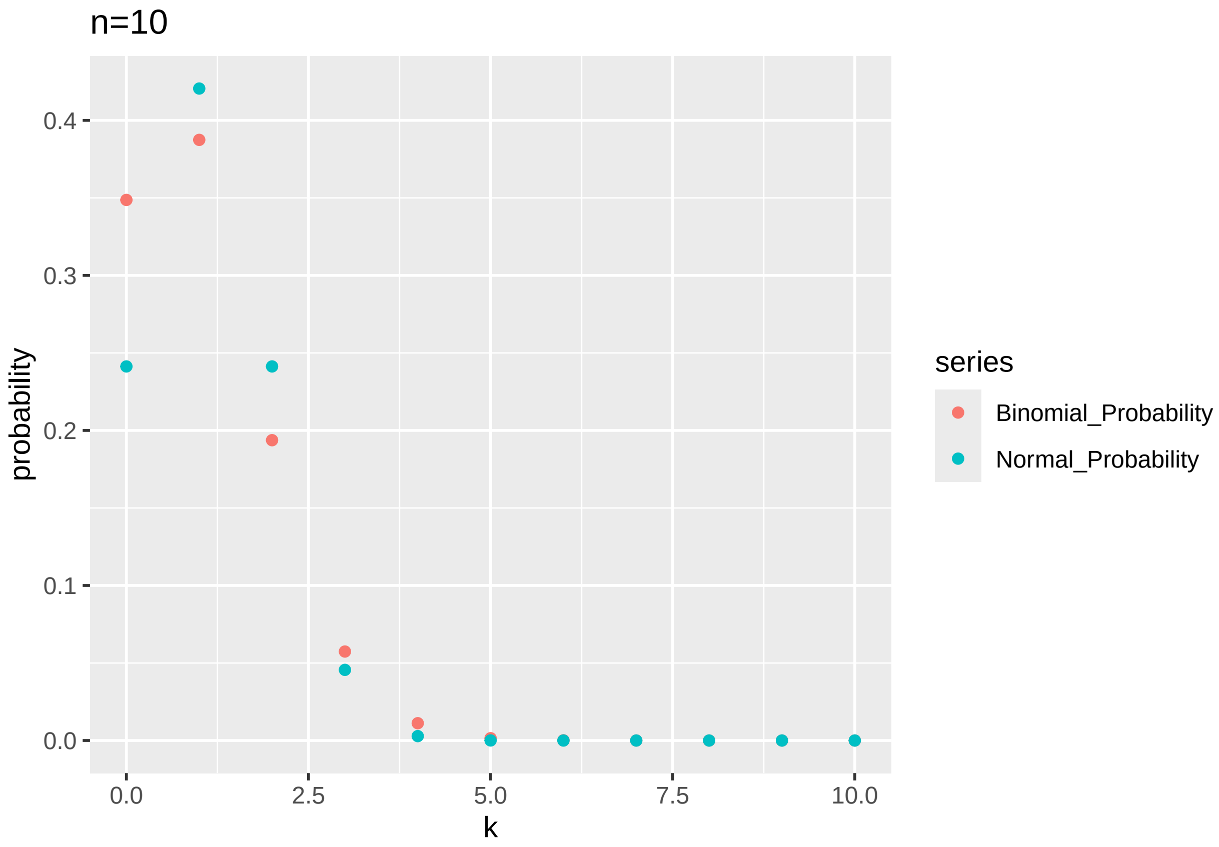
```r
# Begin by defining n number of trials, as well as p for success probability
n <- 10
p <- 0.5
q <- 1 - p
k <- 0:n

# Determine probabilities for different numbers of successes between [0,n] for binomial distribution
probs_bin <- dbinom(k, size=n, prob=p)

# Determine normal approximation of probabilities with n and p
normal_approx <- dnorm(k, mean=n*p, sd=sqrt(n*p*q))

# Establish dataframe for binomial probs and normal approx to make simpler use of ggplot
df <- data.frame(
  x = 0:(length(normal_approx)-1),
  Binomial_Probability = probs_bin,
  Normal_Probability = normal_approx
)

# Pivot dataframe to allow for plotting y1, and y2 (binomial and normal probabilities)
df2 <- pivot_longer(df, cols = c(Binomial_Probability, Normal_Probability), names_to = "series", values

# Plot results
ggplot2::ggplot(data=df2, aes(x = x, y = value, color = series)) +
  geom_point() +
  labs(x = "k", y = "probability")
```



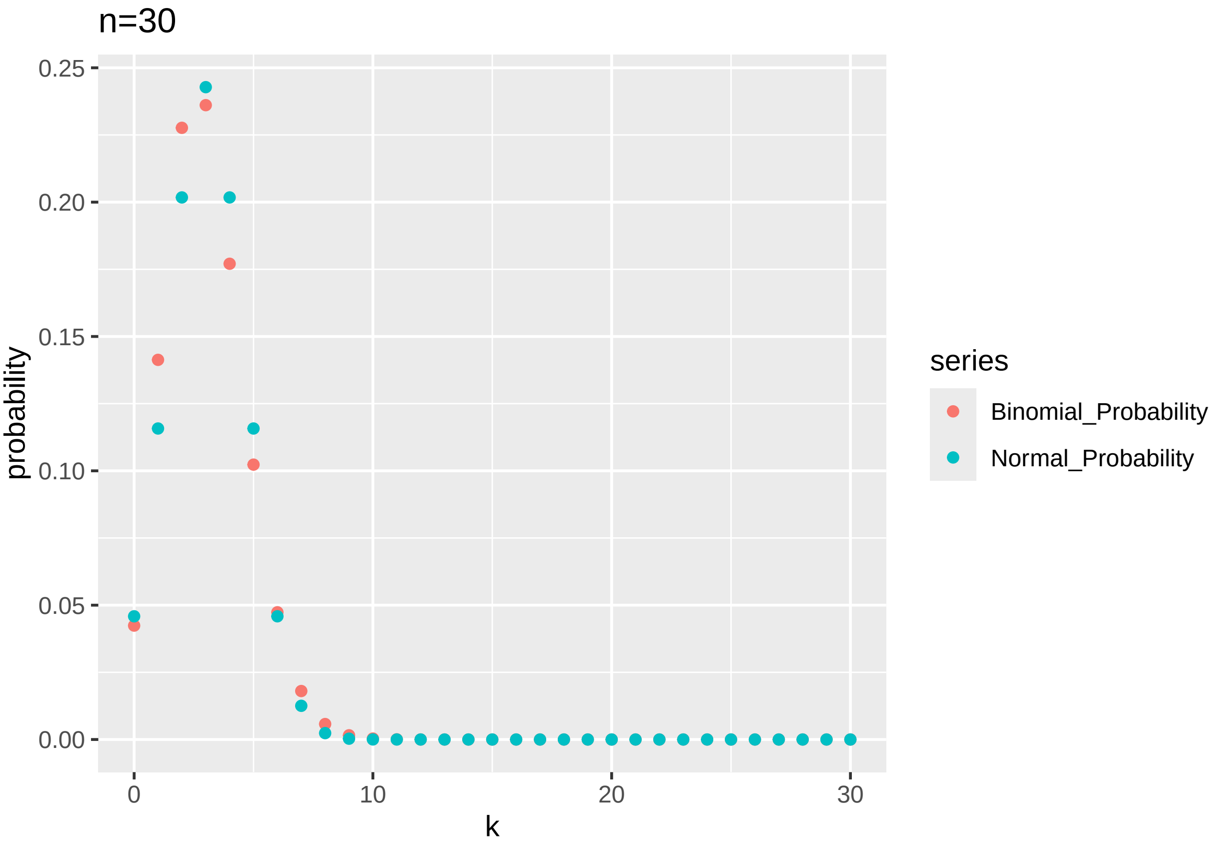
### Remarks From this graphic it is evident that with parameters $n = 10$ and $p = 0.5$, that the binomial probability mass function bears a distribution glaringly similar to the normal distribution with a mean of np, which is to be expected with a p value so close to 0.5. We can now look to shift around values and see how closely the shape of the binomial resembles the normal distribution.

```
# Begin by defining n number of trials, as well as p for success probability
n <- 10; n2 <- 30; n3 <- 60
p <- 0.1
q <- 1 - p
k <- 0:n; k2 <- 0:n2; k3 <- 0:n3

# Determine probabilities for different numbers of successes between [0,n], [0,n2], and [0,n3] for bino
probs_bin <- dbinom(k, size=n, prob=p)
probs_bin2 <- dbinom(k2, size=n2, prob=p)
probs_bin3 <- dbinom(k3, size=n3, prob=p)

# Determine normal approximation of probabilities with n and p (n2 and n3 as well)
normal_approx <- dnorm(k, mean=n*p, sd = sqrt(n*p*q))
normal_approx2 <- dnorm(k2, mean=n2*p, sd = sqrt(n2*p*q))
normal_approx3 <- dnorm(k3, mean=n3*p, sd = sqrt(n3*p*q))

# Establish dataframe for binomial probs and normal approx to make simpler use of ggplot for each n val
df <- data.frame(
  x = 0:(length(normal_approx)-1),
  Binomial_Probability = probs_bin,
  Normal_Probability = normal_approx
)
```

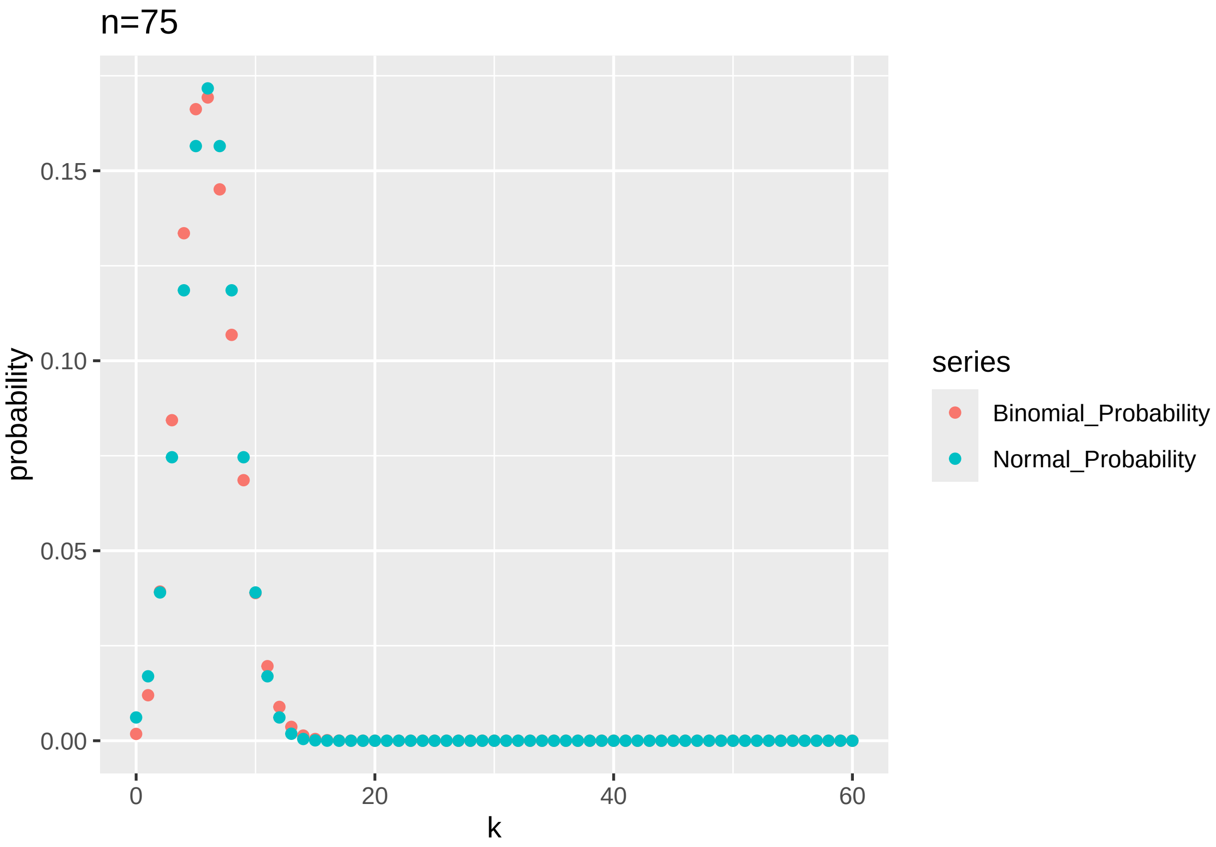
```
df2 <- data.frame(
  x = 0:(length(normal_approx2)-1),
  Binomial_Probability = probs_bin2,
  Normal_Probability = normal_approx2
)

df3 <- data.frame(
  x = 0:(length(normal_approx3)-1),
  Binomial_Probability = probs_bin3,
  Normal_Probability = normal_approx3
)

# Pivot dataframe to allow for plotting y1, and y2 (binomial and normal probabilities)
df4 <- pivot_longer(df, cols = c(Binomial_Probability, Normal_Probability), names_to = "series", values_
df5 <- pivot_longer(df2, cols = c(Binomial_Probability, Normal_Probability), names_to = "series", values
df6 <- pivot_longer(df3, cols = c(Binomial_Probability, Normal_Probability), names_to = "series", values

# Plot results n=10
ggplot2::ggplot(data=df4, aes(x = x, y = value, color = series)) +
  geom_point() +
  labs(title = "n=10", x = "k", y = "probability")
```



```
# Plot results n=30
ggplot2::ggplot(data=df5, aes(x = x, y = value, color = series)) +
  geom_point() +
  labs(title = "n=30", x = "k", y = "probability")
```



```
# Plot results n=75
ggplot2::ggplot(data=df6, aes(x = x, y = value, color = series)) +
  geom_point() +
  labs(title = "n=75", x = "k", y = "probability")
```

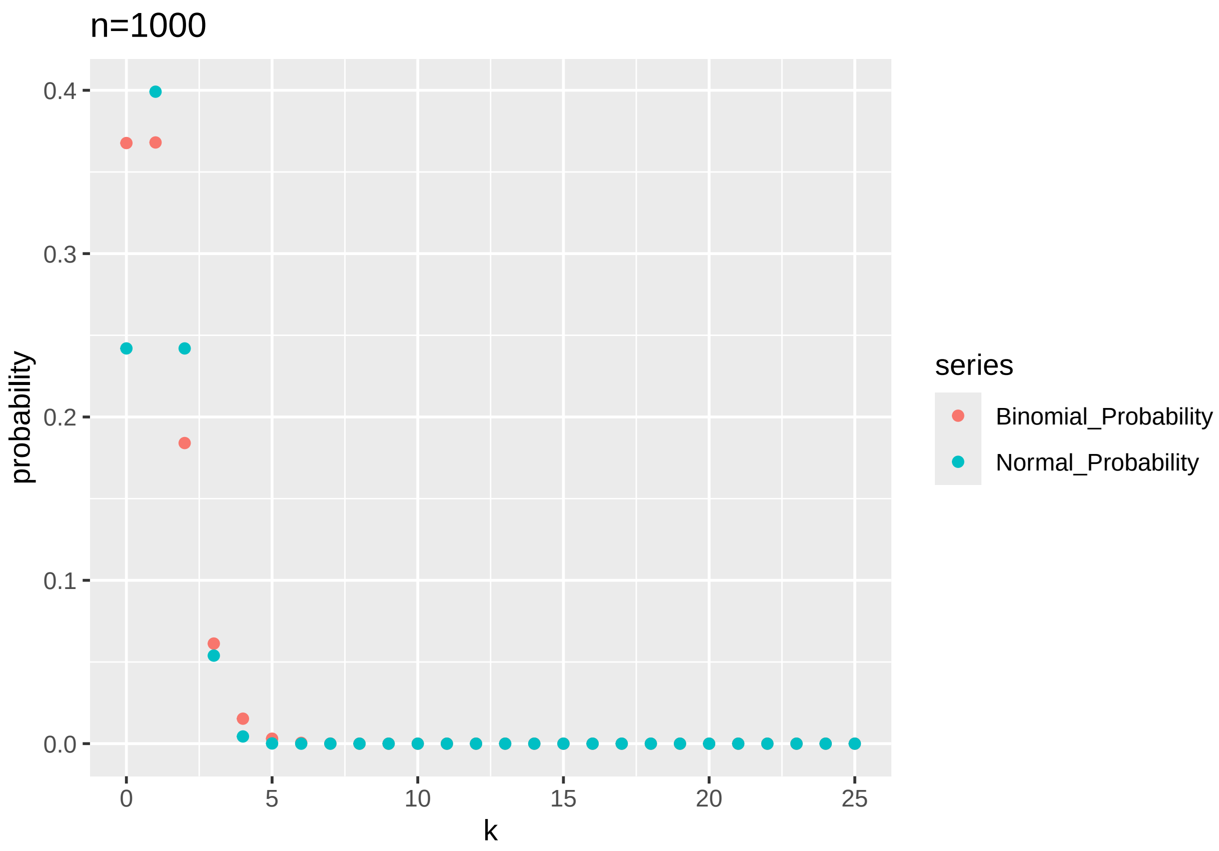


Remarks

We can see here that with parameter $p = 0.1$, that even with n taking on a value of 10, the distributions still bear resemblance, and increasing n to values like 30 and 60 makes this even more clear, which tells us that n need not be exceedingly large in order for the normal distribution to properly estimate the binomial distribution. Let's now consider an extreme case for p where it is very close to 0, which I presume will cause the accuracy of the estimate to falter, due to the extremity, or require an infeasibly large n to be sufficient.

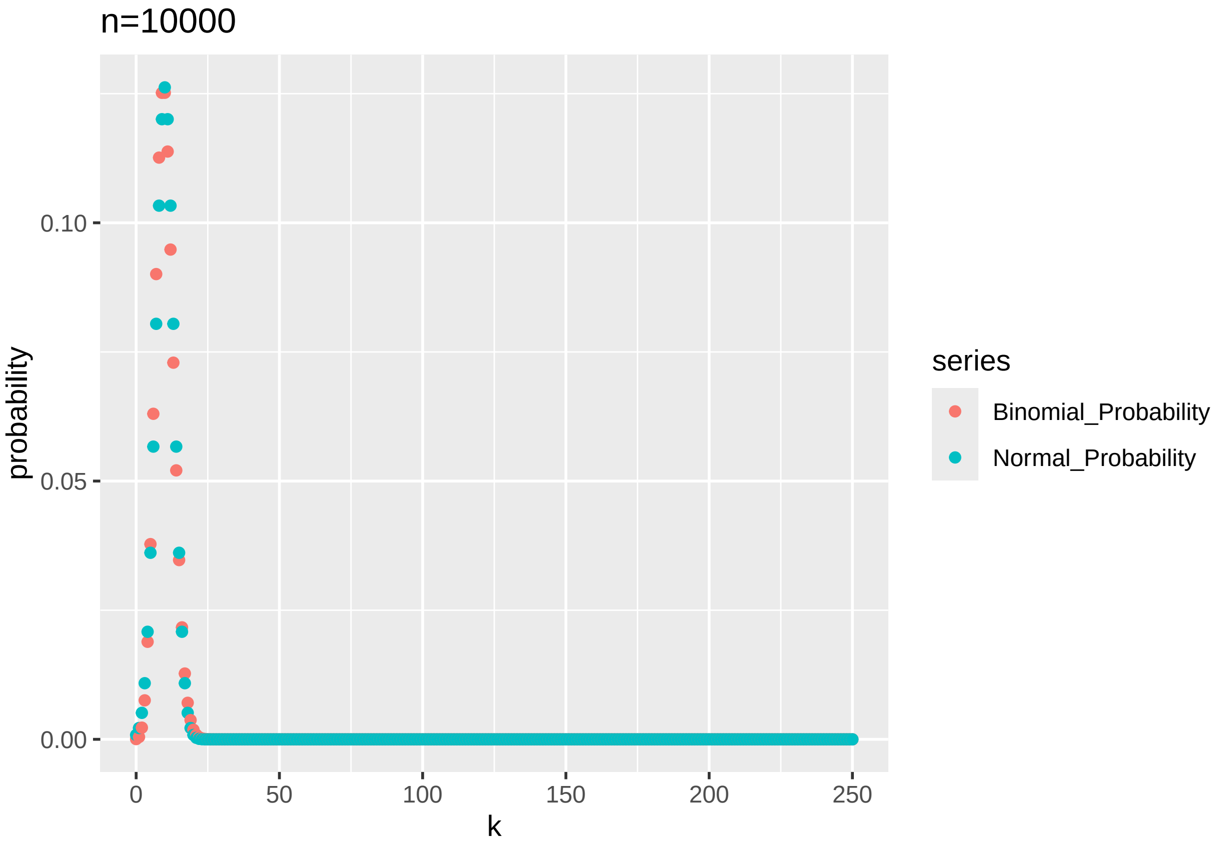
```
# Begin by defining n number of trials, as well as p for success probability
n <- 1000; n2 <- 10000
p <- 0.001
q <- 1 - p
k <- 0:n; k2 <- 0:n2

# Determine probabilities for different numbers of successes between [0,n] for binomial distribution
probs_bin <- dbinom(k, size=n, prob=p)
probs_bin2 <- dbinom(k2, size=n2, prob=p)

# Determine normal approximation of probabilities with n and p
normal_approx <- dnorm(k, mean=n*p, sd = sqrt(n*p*q))
normal_approx2 <- dnorm(k2, mean=n2*p, sd = sqrt(n2*p*q))

# Establish dataframe for binomial probs and normal approx to make simpler use of ggplot
df <- data.frame(
  x = 0:(length(normal_approx)-1),
  Binomial_Probability = probs_bin,
  Normal_Probability = normal_approx
)

df2 <- data.frame(
  x = 0:(length(normal_approx2)-1),
  Binomial_Probability = probs_bin2,
  Normal_Probability = normal_approx2
)

# Pivot dataframe to allow for plotting y1, and y2 (binomial and normal probabilities)
df3 <- pivot_longer(df, cols = c(Binomial_Probability, Normal_Probability), names_to = "series", values_
df4 <- pivot_longer(df2, cols = c(Binomial_Probability, Normal_Probability), names_to = "series", value

# Plot results n=1000
ggplot2::ggplot(data=df3, aes(x = x, y = value, color = series)) +
  geom_point() +
  labs(title = "n=1000", x = "k", y = "probability") +
  xlim(0, n/40)
```

```
## Warning: Removed 1950 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



```
# Plot results n=10000
ggplot2::ggplot(data=df4, aes(x = x, y = value, color = series)) +
  geom_point() +
  labs(title = "n=10000",x = "k", y = "probability") +
   xlim(0, n2/40)
```

```
## Warning: Removed 19500 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



Remarks

In this case we let p take on a value close to 0 and see that even after $n = 1000$, the resemblance is there but not sufficient, until $n = 10000$, and we can obviously see that with p values nearing 0 and 1 that the size of n actually does need to be exceedingly large for the approximation to hold.

Overall

From this experiment we can conclude that for medial values of p in the binomial distribution, n need not be exceedingly large before it closely resembles a normal distribution with mean np and variance npq, but the theorem’s asymptotic nature rears its head as you approach extreme values of p close to 0 or 1, then n is required to be relatively large in order for the resemblance to show.