Spatial Experiments Doc 1

Antony Sikorski

```
suppressMessages(library( fields))
suppressMessages(library(ggplot2))
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
suppressMessages(library(latex2exp))
```

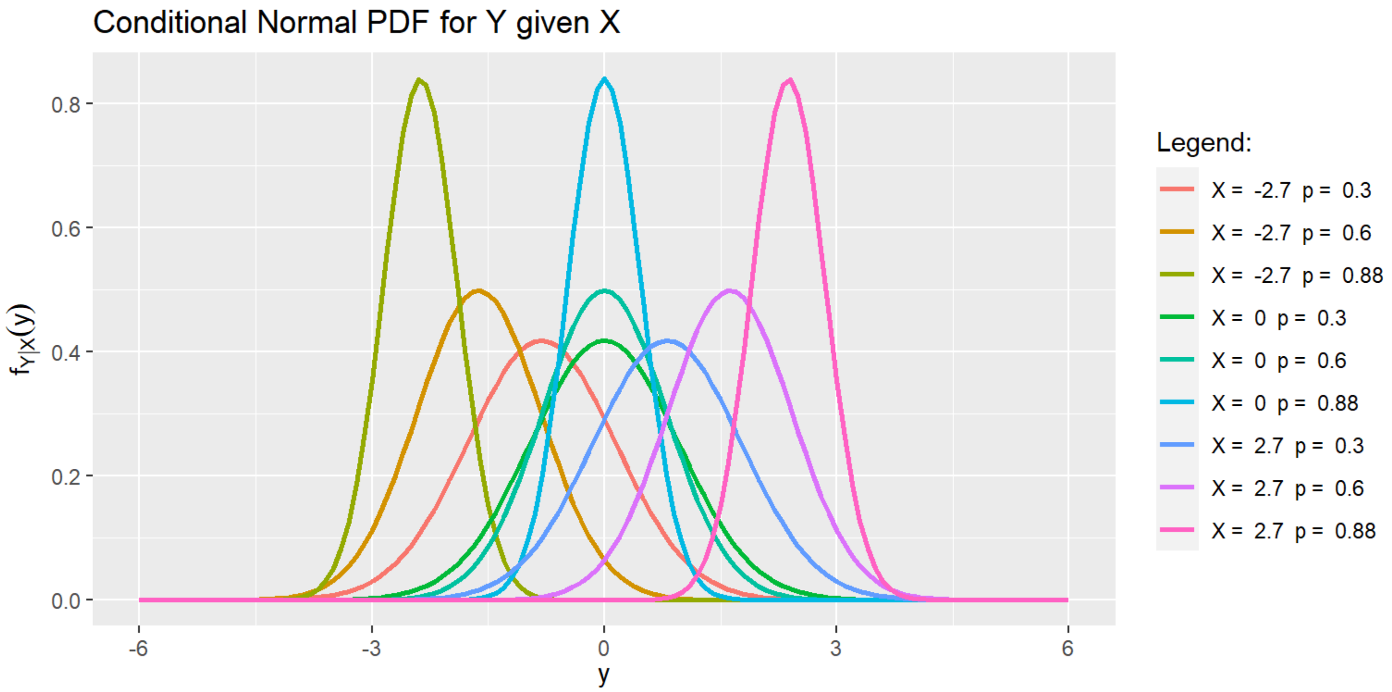
Conditional Gaussian Distributions

We need to determine the conditional pdf for Y given X. First, we find that $\mu = \frac{\rho(x-0)}{1} + 0 = \rho x$, and $\sigma^2 = (1-\rho^2)(1) = 1-\rho^2$. We find that

$$f_{Y|X}(y) = \frac{1}{\sqrt{2\pi(1-\rho^2)}} e^{\frac{-1}{2}\frac{(y-\rho x)^2}{(1-\rho^2)}}$$

Here is the first plot, with each combination of X and ρ as a separate color.

```
#creating data frame with our different ro and x values
df <- expand.grid(y = seq(-6,6,0.1), roVals = c(0.3, 0.6, 0.88), xVals = c(-2.7, 0, 2.7))

#writing in the pdf function
df$func <- (1/(sqrt(2*pi*(1-(df$roVals^2)))))*exp((-(1/2)*(df$y-(df$roVals*df$xVals))^2*(1/(1-(d
f$roVals^2)))))

#implementing a way to color the columns
df$colorcode <- paste("X = ", df$xVals, " p = ", df$roVals)



#plotting
gg <- ggplot(df, aes(y, func, colour = factor(colorcode))) +
  geom_path(linewidth = 1) +
  xlab(TeX("$y$")) +
  ylab(TeX("$f_{Y|X}(y)")) +
  labs(colour = "Legend:") +
  ggtitle("Conditional Normal PDF for Y given X")
gg
```

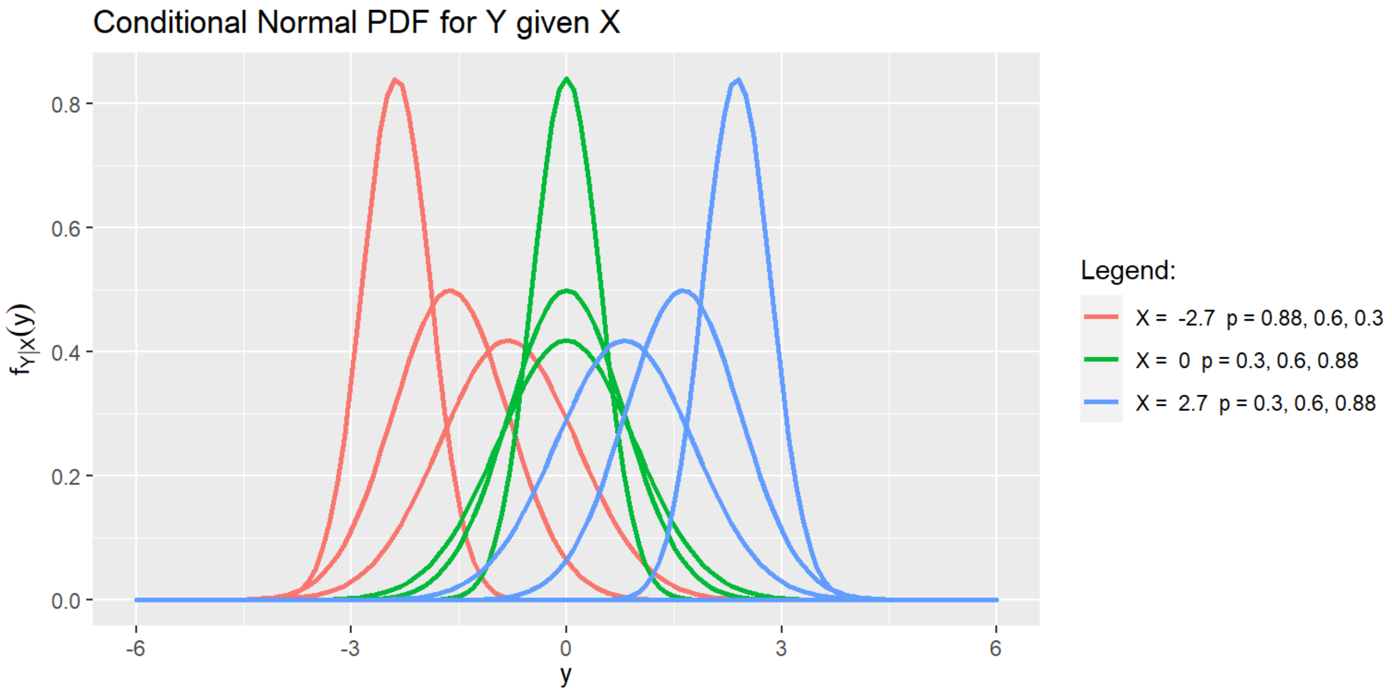


Here is the same plot, but this time it is color coded simply by the X value, to get a more clear idea of how the pdf changes when the correlation between X and Y changes.

```
#nearly identical to code above
df <- expand.grid(y = seq(-6,6,0.1), roVals = c(0.3, 0.6, 0.88), xVals = c(-2.7, 0, 2.7))
df$func <- (1/(sqrt(2*pi*(1-(df$roVals^2)))))*exp((-(1/2)*(df$y-(df$roVals*df$xVals))^2*(1/(1-(d
f$roVals^2)))))

#difference lies here, where we color code only based on X vals
df$colorcode <- paste("X = ", df$xVals, " p = 0.3, 0.6, 0.88")

#replacing the order to be consistent with what is shown on the graph from left to right
df$colorcode[df$colorcode == "X =  -2.7  p = 0.3, 0.6, 0.88"] <- "X =  -2.7  p = 0.88, 0.6, 0.3"


#plotting
gg <- ggplot(df, aes(y, func, colour = factor(colorcode))) +
  geom_path(size = 1) +
  xlab(TeX("$y$")) +
  ylab(TeX("$f_{Y|X}(y)")) +
  labs(colour = "Legend:") +
  ggtitle("Conditional Normal PDF for Y given X")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
```

```
gg
```



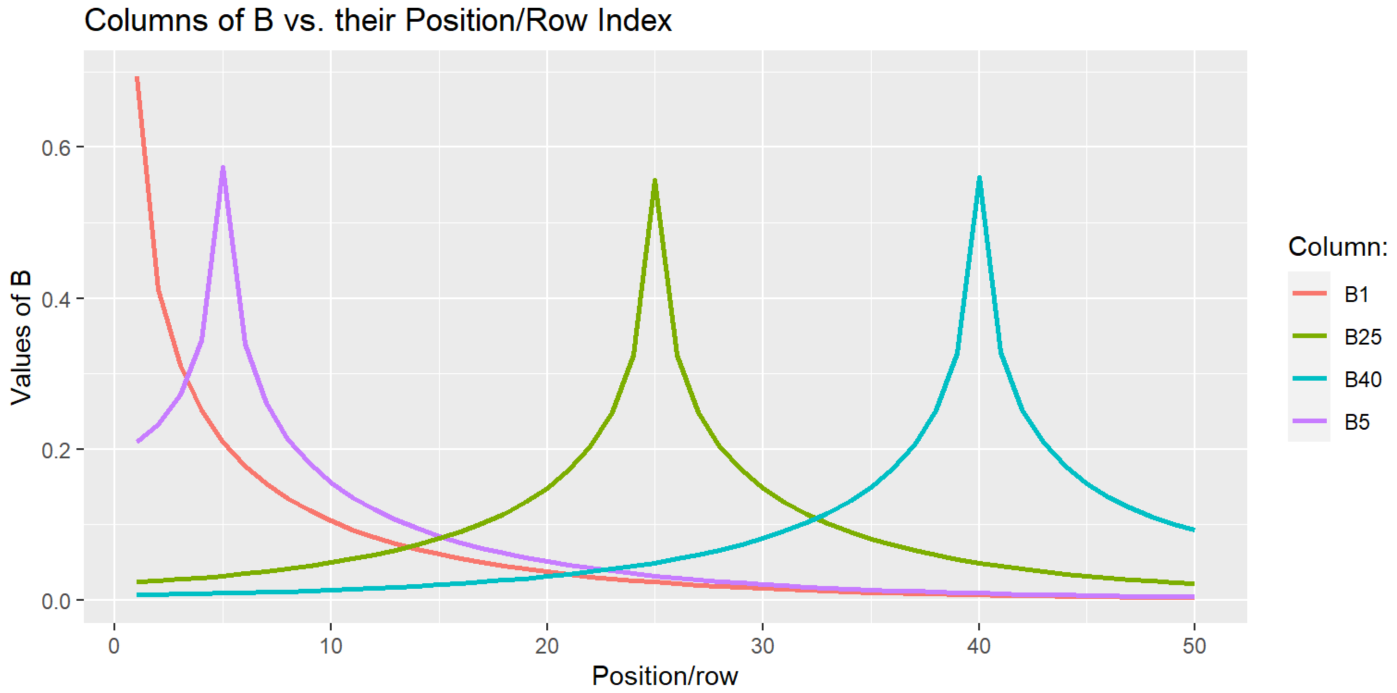
Understanding covariance matrices

We create our covariance matrix, and then we find the symmetric square root, B.

```
#creating the covariance matrix
Cov <- matrix(nrow = 50, ncol = 50)
for (i in 1:50) {
  for (j in 1:50) {
    Cov[i,j] <- exp(-abs(i-j)/15)
  }
}

#symmetric square root code given by Dr. Nychka
hold<- eigen(Cov, symmetric=TRUE)
D.5<- diag( sqrt( hold$values))
B <- hold$vectors %*%D.5 %*% t(hold$vectors )
```

Next, we graph the columns 1, 5, 25, and 40 of B against their locations.

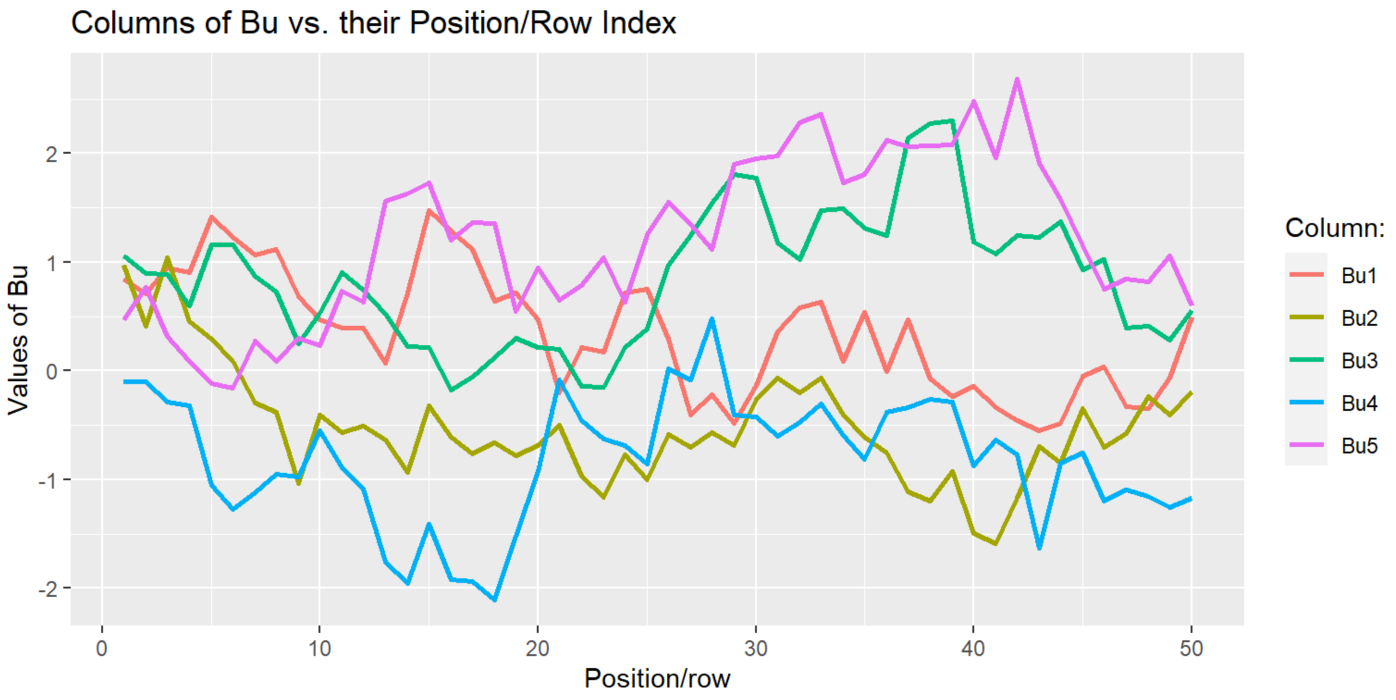
```
#combining the columns of Y into a data frame in order to plot
X <- 1:50
Y <- cbind(B[,1], B[,5], B[,25], B[,40])
df_B <- data.frame(X = X, B1 = B[,1], B5 = B[,5], B25 = B[,25], B40 = B[,40] )

#plotting
bb <- ggplot(df_B, aes(X)) +
  geom_path(aes(y = B1, colour = "B1"), size = 1) +
  geom_line(aes(y = B5, colour = "B5"), size = 1) +
  geom_line(aes(y = B25, colour = "B25"), size = 1) +
  geom_line(aes(y = B40, colour = "B40"), size = 1) +
  xlab("Position/row") +
  ylab("Values of B") +
  labs(colour = "Column:") +
  ggtitle("Columns of B vs. their Position/Row Index")
bb
```



The curves seem to peak at the their entries that are a part of the diagonal of the matrix. Mathematically speaking, their peaks are of height B_{ij}, where $i = j$ (Ex: the curve for B25 peaks at index 25). This makes sense, as the matrix is symmetric, and it's diagonal entries will be the greatest since it is the symmetric square root of a covariance matrix who's diagonal entries are also it's largest entries.

Now we will plot 5 realizations of the X vector where $X = B\mathbf{u}$, and $\mathbf{u}$ are iid N(0,1). We will plot each column of X together to get an idea of the variation of the features.

```r
set.seed(777)

#5 realizations of the X vector, stapled together in a matrix
Xnew <- matrix(nrow = 50, ncol = 5)
for (i in 1:5){
  u <- rnorm(50, 0, 1)
  Xnew[,i] <- B%*%u
}

#data frame for ggplot
df_u <- data.frame(X = X, U1 = Xnew[,1], U2 = Xnew[,2], U3 = Xnew[,3],
                   U4 = Xnew[,4], U5 = Xnew[,5])

#plotting
bb_u <- ggplot(df_u, aes(X)) +
  geom_path(aes(y = U1, colour = "Bu1"), size = 1) +
  geom_path(aes(y = U2, colour = "Bu2"), size = 1) +
  geom_path(aes(y = U3, colour = "Bu3"), size = 1) +
  geom_path(aes(y = U4, colour = "Bu4"), size = 1) +
  geom_path(aes(y = U5, colour = "Bu5"), size = 1) +
  xlab("Position/row") +
  ylab("Values of Bu") +
  labs(colour = "Column:") +
  ggtitle("Columns of Bu vs. their Position/Row Index")
bb_u
```



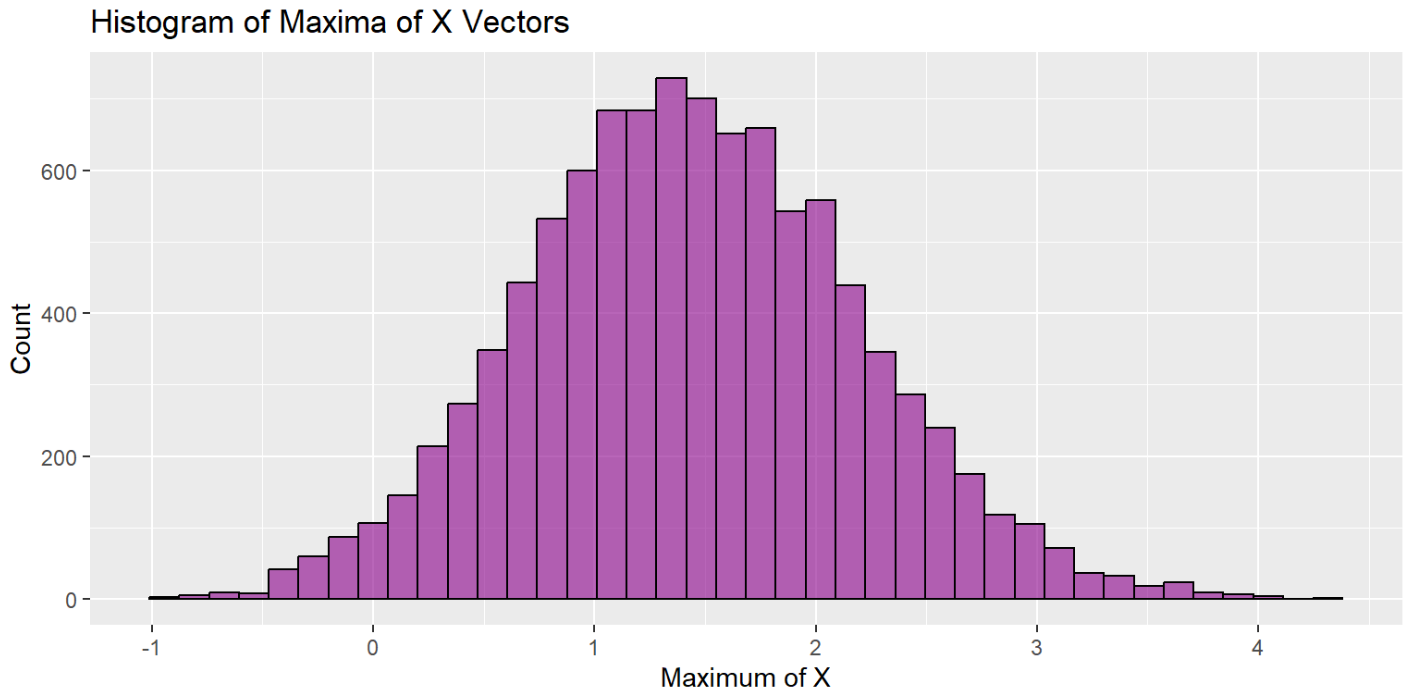
Since X follows a multivariate normal distribution, we know that each of it's components follow a univariate normal distribution. A linear combination of two of those components will also follow a univariate distribution. For $\frac{X_{26}+X_{25}}{2}$, we find that $E(\frac{X_{26}+X_{25}}{2}) = \frac{1}{2}(E(X_{25}) + E(X_{26})) = 0$. Since we have our covariance matrix Σ as well, we can find that the variance is

$$Var(\frac{X_{26}+25}{2}) = \frac{1}{4}(Var(X_{25}) + Var(X_{26} + 2\Sigma[25,26])) = \frac{1}{4}(1+1+2(0.9355070)) = 0.9677$$

. Thus, we know that $\frac{X_{26}+X_{25}}{2}$ follows a normal distribution $N(0, 0.9677)$.

Let’s upt the stakes. This time, we will do something similar to part 3(b), and we will generate significantly more (10,000) realizations of the X vector.

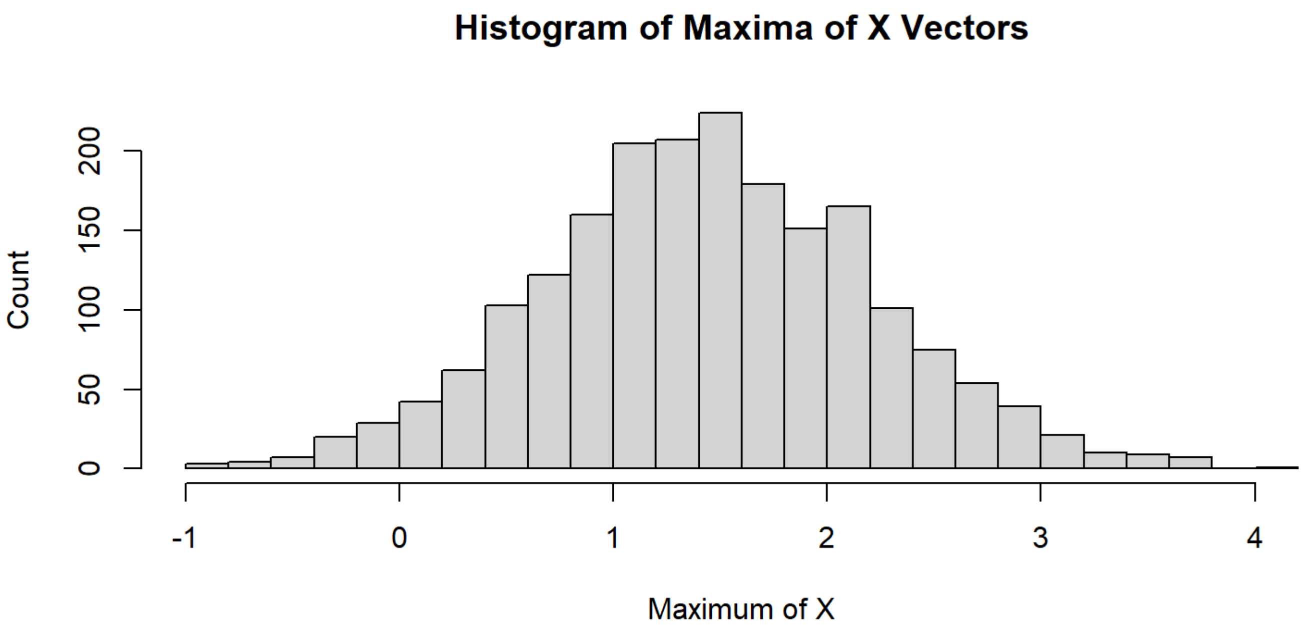
```
n = 10000
Y2 <- {}
set.seed(777)
for (i in 1:n){
  u <- rnorm(50, 0, 1)
  X2 <- B%*%u
  Y2 <- append(Y2, max(X2))
}

dfFinal <- data.frame(X = 1:10000, Y2)

ggplot(dfFinal, aes(Y2)) +
  geom_histogram(bins = 40, color = "black", fill = "darkmagenta", alpha = 0.6) +
  xlab("Maximum of X") +
  ylab("Count") +
  ggtitle("Histogram of Maxima of X Vectors")
```



The distribution of the maxima of the X vectors does seem to look Normal.

Now we practice our efficiency in R by generating a matrix that is 2000 realizations of this process, and generating a histogram for the maxima, all in 2 lines.

```r
set.seed(777)
X_speedy <- B%*%matrix(rnorm(100000, 0, 1), nrow=50, ncol = 2000)
hist(apply(X_speedy, 2, max), breaks = 30, main = "Histogram of Maxima of X Vectors", xlab = "Maximum of X", ylab = "Count")
```


Histogram of Maxima of X Vectors
Count
Maximum of X