

Length n Probability Generation

Matthew Seguin

Summary

The goals of this document are:

- Explain what an probability vector of length n is and what properties such a vector has.
- Illustrate how to generate a random probability vector of length n .
- Analyze how to bias such a probability vector.
- Examine the distribution of the norms of such vectors.

Importing Libraries

```
library(tidyverse)
library(latex2exp)
```

Generator

```
prob_generator <- function(N, m = 1000, shape = NA){  
  shape <- ifelse(is.na(shape), rep(1, N), shape)  
  gams <- matrix(rgamma(n = N*m, shape = shape), nrow = N)  
  probs <- t(t(gams)/colSums(gams))  
  return(probs)  
}
```

This just uses the Dirichlet distribution for generating the probabilities.

The intuition is that we generate $X_j \sim \text{Gamma}(\alpha_j, \lambda)$ for $j \in \{1, \dots, n\}$. Then we consider the distribution of

$$p = (p_1, \dots, p_n) = \left(\frac{X_1}{S}, \dots, \frac{X_n}{S} \right) \text{ where } S = \sum_{j=1}^n X_j.$$

Note that this is an $n - 1$ dimensional vector because we know $\sum_{j=1}^n p_j = \sum_{j=1}^n \frac{X_j}{S} = \frac{1}{S} \sum_{j=1}^n X_j = \frac{S}{S} = 1$. Quick note:

once we know $n - 1$ of the p_j we know the last one.

Furthermore we also know $X_j \geq 0$ for each $j \in \{1, \dots, n\}$ (since X_j follows a gamma distribution) so

$S = \sum_{j=1}^n X_j > X_j \geq 0$ for each $j \in \{1, \dots, n\}$ as well. Which then tells us that $p_j = \frac{X_j}{S} \geq 0$ and $p_j = \frac{X_j}{S} \leq 1$.

So $p = (p_1, \dots, p_n)$ is indeed a probability vector of length n .

The support of the Dirichlet distribution is the standard $n - 1$ simplex. Which is given by

$$\Delta_{n-1} = \left\{ (p_1, \dots, p_n) \in \mathbb{R}^n : \forall_{j \in \{1, \dots, n\}} 0 \leq p_j \leq 1, \sum_{j=1}^n p_j = 1 \right\}.$$

Furthermore the density can be given by:

$$f_{p_1, \dots, p_n}(p_1, \dots, p_n) = \frac{1}{B(\alpha)} \prod_{j=1}^n x_j^{\alpha_j - 1} \text{ for } (p_1, \dots, p_n) \in \Delta_{n-1}$$

$$\text{Where } B(\alpha) = \frac{\Gamma(\alpha_1) \dots \Gamma(\alpha_n)}{\Gamma(\alpha_1 + \dots + \alpha_n)}.$$

The special case where $\alpha_j = 1$ for each $j \in \{1, \dots, n\}$ is essentially a uniform distribution over the standard $k - 1$ simplex

as the density is given by:

$$f_{p_1, \dots, p_n}(p_1, \dots, p_n) = \frac{1}{B(\alpha)} \prod_{j=1}^n x_j^{\alpha_j - 1} = \frac{1}{B(\alpha)} \prod_{j=1}^n x_j^{1-1} = \frac{1}{B(\alpha)} = \frac{\Gamma(\alpha_1 + \dots + \alpha_n)}{\Gamma(\alpha_1) \dots \Gamma(\alpha_n)}$$

$$= \frac{\Gamma(n)}{(\Gamma(1))^n} = \frac{(n-1)!}{(0!)^n} = (n-1)! \text{ for } (p_1, \dots, p_n) \in \Delta_{n-1}$$

It is uniform since clearly the density does not depend on $p = (p_1, \dots, p_n)$.

The above result uses the fact that $\Gamma(k) = (k-1)!$ for $k \in \{1, 2, \dots\}$ which I will prove below:

$$\Gamma(k) = \int_0^\infty t^{k-1} e^{-t} dt$$

This clearly converges for $k-1 \geq 0$ i.e. for $k \geq 1$.

Now let $u(t) = t^{k-1}$ and $\frac{dv}{dt} = e^{-t}$ so that $\frac{du}{dt} = (k-1)t^{k-2}$ and $v(t) = -e^{-t}$. Then:

$$\begin{aligned} \Gamma(k) &= \int_0^\infty t^{k-1} e^{-t} dt = u(t)v(t) \Big|_0^\infty - \int_0^\infty v(t) \frac{du}{dt} dt = -t^{k-1} e^{-t} \Big|_0^\infty + \int_0^\infty (k-1)t^{k-2} e^{-t} dt \\ &= \lim_{t \rightarrow \infty} -\frac{t^{k-1}}{e^t} + 0 + (k-1) \int_0^\infty t^{(k-1)-1} e^{-t} dt = (k-1)\Gamma(k-1) \end{aligned}$$

Which again converges for $k-1 \geq 1$ or equivalently $k \geq 2$.

Therefore if $k \geq 2$ then $\Gamma(k) = (k-1)\Gamma(k-1)$ which seems like a factorial, now we need our base case:

$$\Gamma(1) = \int_0^\infty t^{1-1} e^{-t} dt = \int_0^\infty e^{-t} dt = \int_{-\infty}^\infty f_T(t) dt = 1 = 0! \quad \text{where } T \sim \text{Exponential}(1)$$

Therefore if $k \geq 1$ we know for

$$\Gamma(k) = (k-1)\Gamma(k-1) = \dots = (k-1)\dots(2)\Gamma(2) = (k-1)\dots(2)(1)\Gamma(1) = (k-1)\dots(2)(1) = (k-1)! \quad \square$$

Testing that this does indeed produce length n probability vectors.

There is always the issue of machine precision for floating point numbers so the sums can not be exactly one but we can make a simple tolerance based on machine epsilon.

```
tol <- 4.75*.Machine$double.eps
tol
```

```
## [1] 1.054712e-15
```

```
for (i in 2:1000){
  test <- prob_generator(N = i)
  stopifnot(
    min(test) >= 0,
    max(abs(colSums(test) - 1)) <= tol
  )
}
```

As there are no errors we can see that all of these have non-negative probabilities and that they sum to 1 (within our tolerance based on machine imprecision) and hence define length n probability vectors.

How to Bias The Generated Probabilities

First recall that the standard $\alpha = (\alpha_1, \dots, \alpha_n) = (1, \dots, 1)$ will produce uniform probabilities over Δ_{n-1} (which is just Ω for length n probabilities). For several values of n let us examine the joint distribution of p_i and p_j under the standard α .

Distribution from the standard $\alpha = (\alpha_1, \dots, \alpha_n) = (1, \dots, 1)$

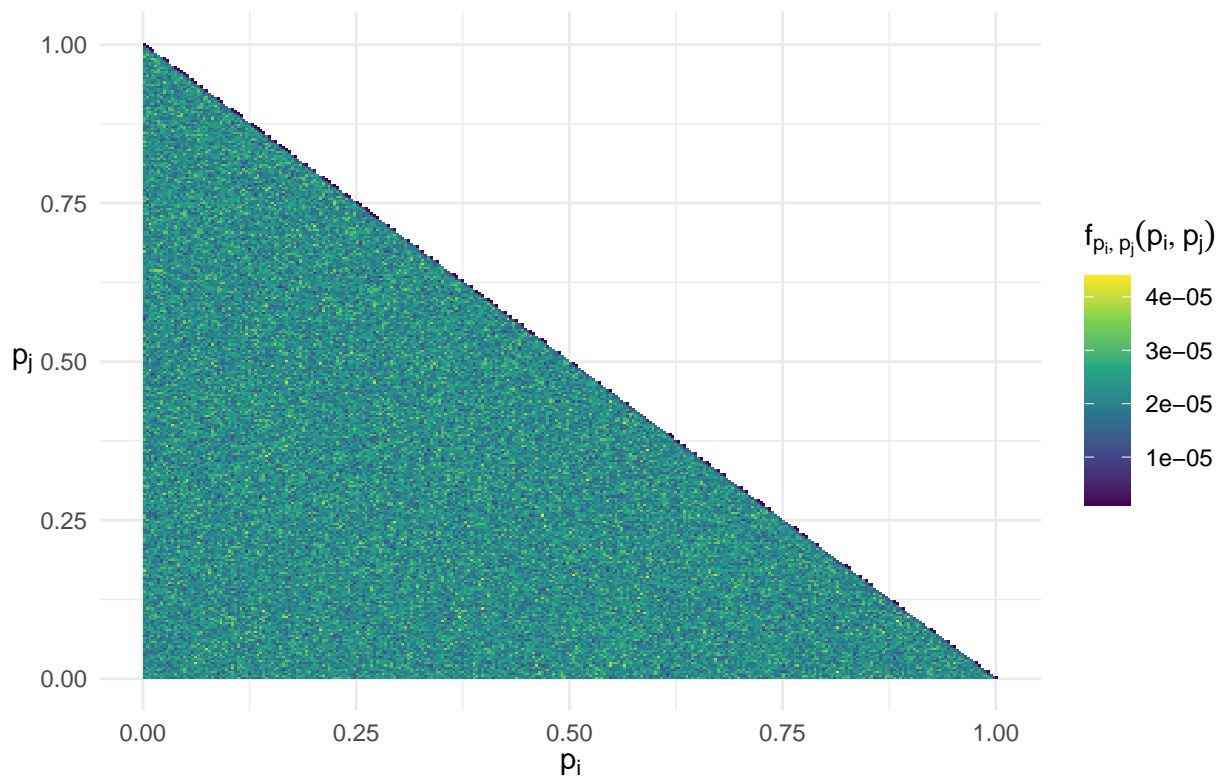
The $n = 2$ case is trivial as we will just have the line $(p, 1 - p)$ so I will skip that case. In the symmetric case we don't need to look at all possible pairs of p_i and p_j we can just use one due to exchangeability.

- For $n = 3$:

```
probs <- prob_generator(N = 3, m = 1000000)
p_comp <- data.frame(pi = probs[1,], pj = probs[3,])

p_comp %>%
  ggplot(aes(x = pi, y = pj)) +
  geom_bin2d(aes(fill = after_stat(density)),
             bins = 300) +
  scale_fill_viridis_c() +
  labs(x = TeX("$p_i$"),
       y = TeX("$p_j$"),
       fill = TeX("$f_{\{p_i, p_j\}}(p_i, p_j)$"),
       title = TeX("$n=3$: Joint Distribution of $(p_i, p_j)$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(color = "black"),
        axis.title.y = element_text(angle = 0, vjust = 0.5))
```

$n = 3$: Joint Distribution of (p_i, p_j)

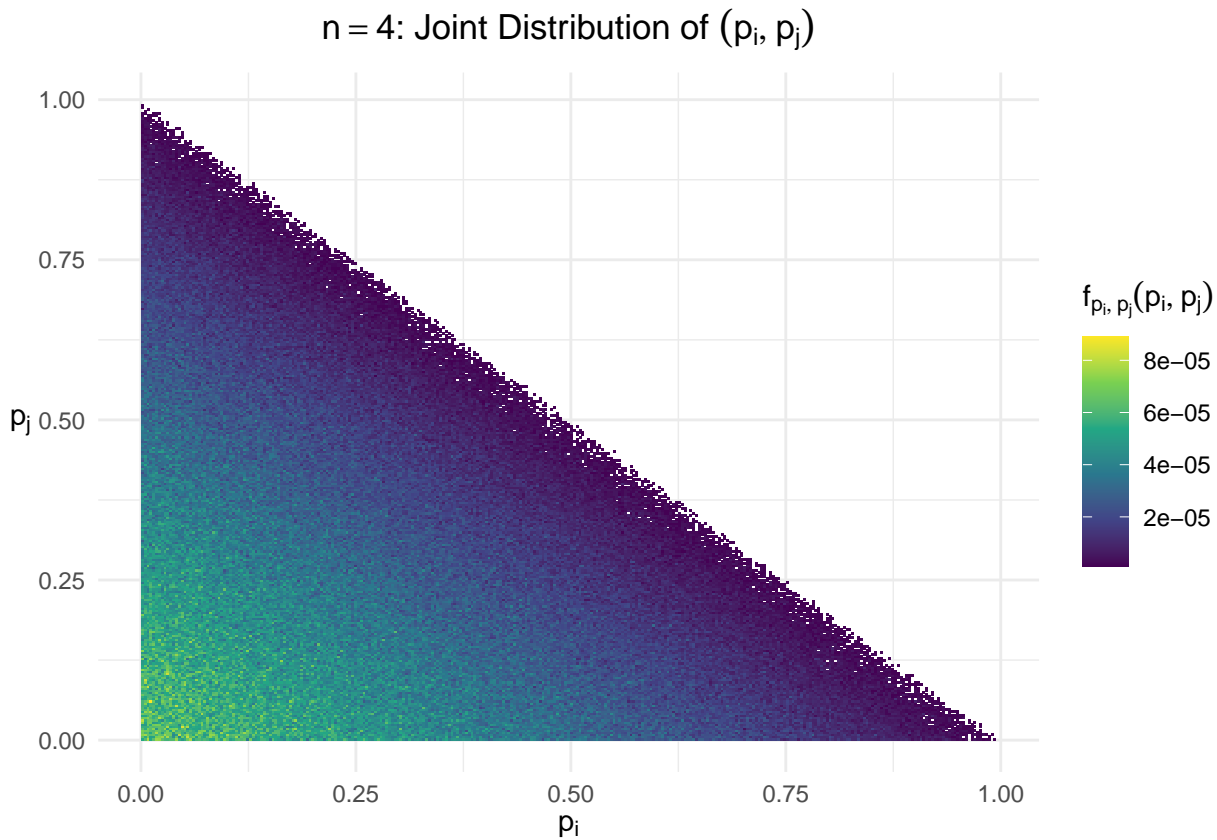


This is exactly what we would expect. Since knowing p_i and p_j entirely determines the final probability and we know $p = (p_1, p_2, p_3)$ is uniform over $\Delta_{n-1} = \Delta_2$ we should see that this is a uniform distribution which it is here.

- For $n = 4$:

```
probs <- prob_generator(N = 4, m = 1000000)
p_comp <- data.frame(pi = probs[1,], pj = probs[4,])

p_comp %>%
  ggplot(aes(x = pi, y = pj)) +
  geom_bin2d(aes(fill = after_stat(density)),
             bins = 300) +
  scale_fill_viridis_c() +
  labs(x = TeX("$p_i$"),
       y = TeX("$p_j$"),
       fill = TeX("$f_{\{p_i, p_j\}}(p_i, p_j)$"),
       title = TeX("$n=4$: Joint Distribution of $(p_i, p_j)$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(color = "black"),
        axis.title.y = element_text(angle = 0, vjust = 0.5))
```

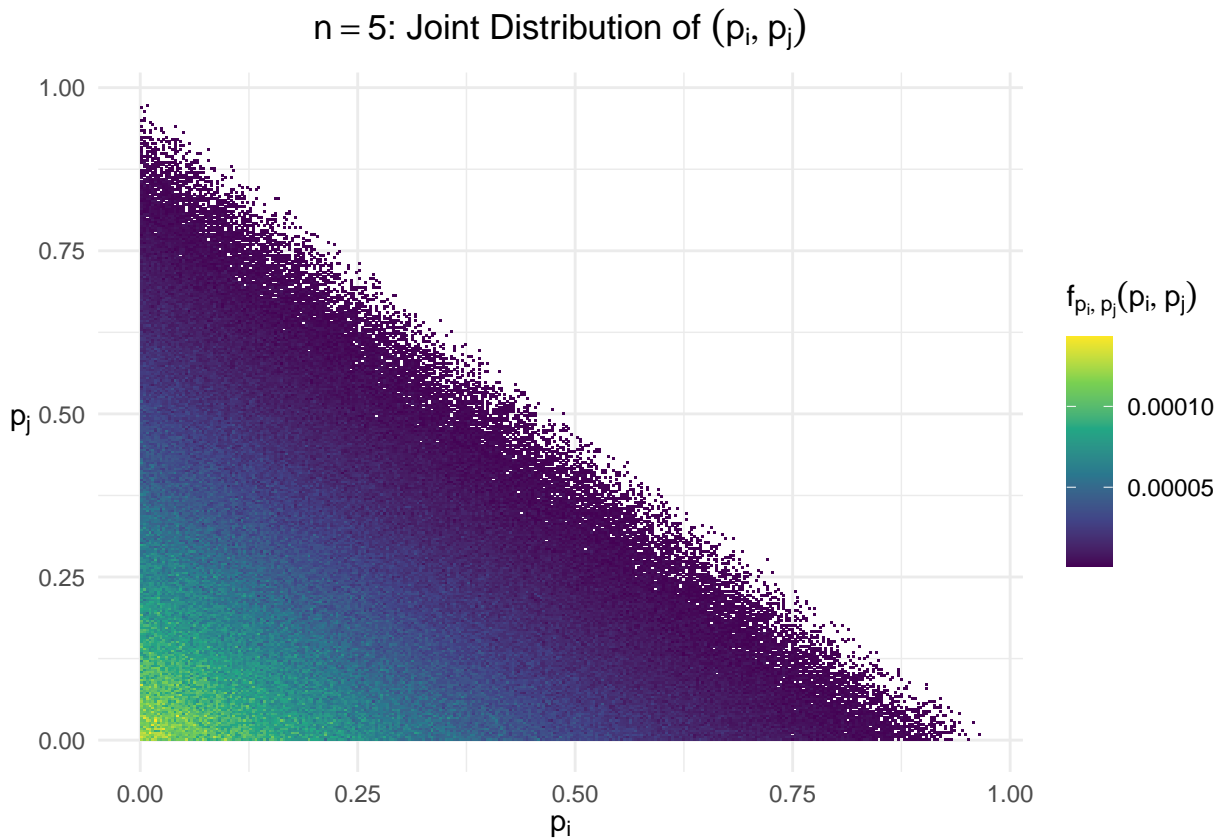


Again this is exactly what we would expect. $p = (p_1, \dots, p_4)$ is now uniform over $\Delta_{n-1} = \Delta_3$. Given Here we are more likely to see smaller values of p_i and p_j since there are now 2 more probabilities we have to account for. So when we assign one of the probabilities the remaining ones must now sum to a value smaller than 1 and so we often see smaller values.

- For $n = 5$:

```
probs <- prob_generator(N = 5, m = 1000000)
p_comp <- data.frame(pi = probs[1,], pj = probs[5,])

p_comp %>%
  ggplot(aes(x = pi, y = pj)) +
  geom_bin2d(aes(fill = after_stat(density)),
             bins = 300) +
  scale_fill_viridis_c() +
  labs(x = TeX("$p_i$"),
       y = TeX("$p_j$"),
       fill = TeX("$f_{\{p_i, p_j\}}(p_i, p_j)$"),
       title = TeX("$n=5$: Joint Distribution of $(p_i, p_j)$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(color = "black"),
        axis.title.y = element_text(angle = 0, vjust = 0.5))
```



Again this is exactly what we would expect. $p = (p_1, \dots, p_5)$ is now uniform over $\Delta_{n-1} = \Delta_4$ and by a similar argument we will see smaller probabilities in this joint graph generally.

- This pattern of smaller probabilities being more likely continues for all $n \in \mathbb{N}$ again by the same reasoning as before.

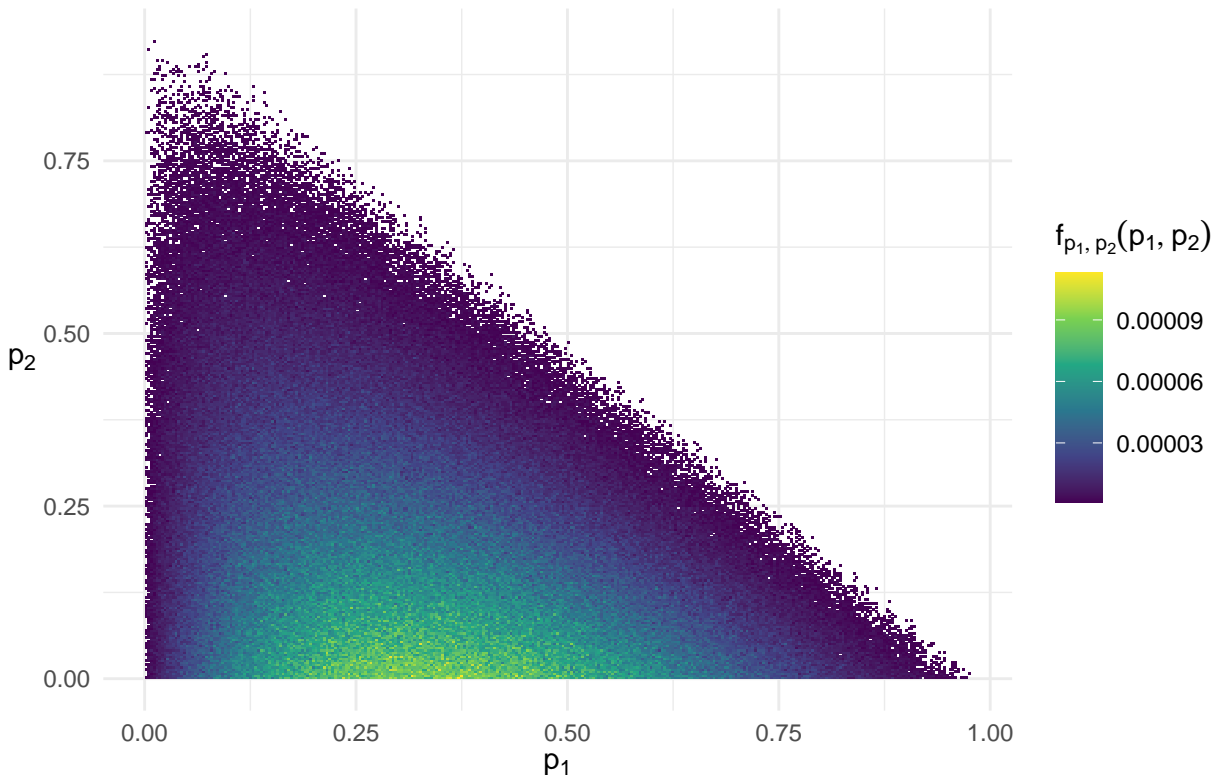
Distribution when we change α

It is a well known fact that the marginal distribution of p_i from a Dirichlet distribution is a $\text{Beta}(\alpha_i, (\alpha_1 + \dots + \alpha_n) - \alpha_i)$ random variable and therefore $\mathbb{E}[p_i] = \frac{\alpha_i}{\alpha_1 + \dots + \alpha_n}$ and hence $\mathbb{E}[p] = \mathbb{E}[(p_1, \dots, p_n)] = (\mathbb{E}[p_1], \dots, \mathbb{E}[p_n]) = (\frac{\alpha_1}{\alpha_1 + \dots + \alpha_n}, \dots, \frac{\alpha_n}{\alpha_1 + \dots + \alpha_n})$. This allows us to appropriately bias the expectation of the probabilities we get by changing the shape parameters for the gamma distributions used. For example if $n = 3$ and we want on average $p_1 = 2p_2$ then we would use $\alpha = (2\lambda, \lambda, \mu)$ to construct our probabilities. Below is a quick example with graphs (using $n = 3$ since before we had a uniform joint distribution there) to illustrate that this works. Now the pairs are not exchangeable so we need to examine each pair.

```
probs <- prob_generator(N = 3, m = 1000000, shape = c(2, 1, 3))
p_comp <- data.frame(p1 = probs[1,],
                    p2 = probs[2,],
                    p3 = probs[3,])

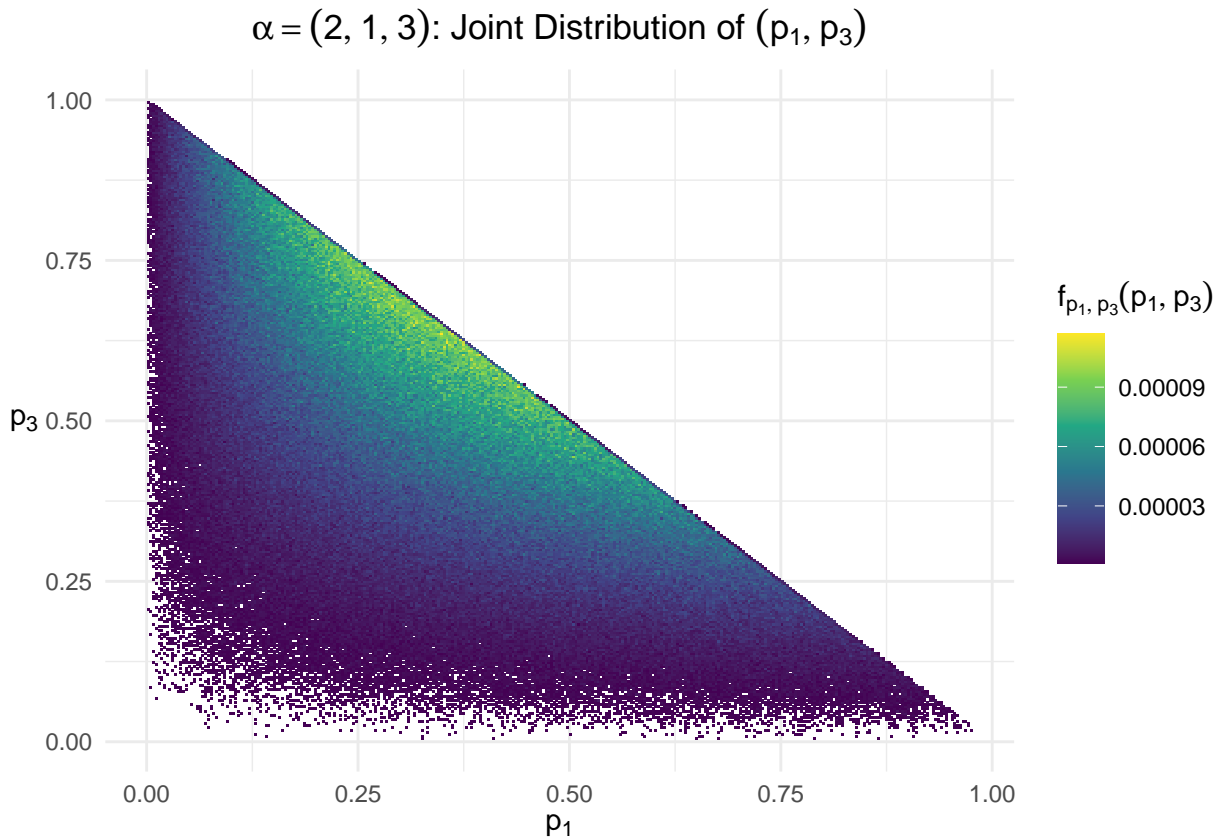
p_comp %>%
  ggplot(aes(x = p1, y = p2)) +
  geom_bin2d(aes(fill = after_stat(density)),
             bins = 300) +
  scale_fill_viridis_c() +
  labs(x = TeX("$p_1$"),
       y = TeX("$p_2$"),
       fill = TeX("$f_{p_1, p_2}(p_1, p_2)$"),
       title = TeX("$\alpha = (2, 1, 3)$: Joint Distribution of $(p_1, p_2)$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(color = "black"),
        axis.title.y = element_text(angle = 0, vjust = 0.5))
```

$\alpha = (2, 1, 3)$: Joint Distribution of (p_1, p_2)



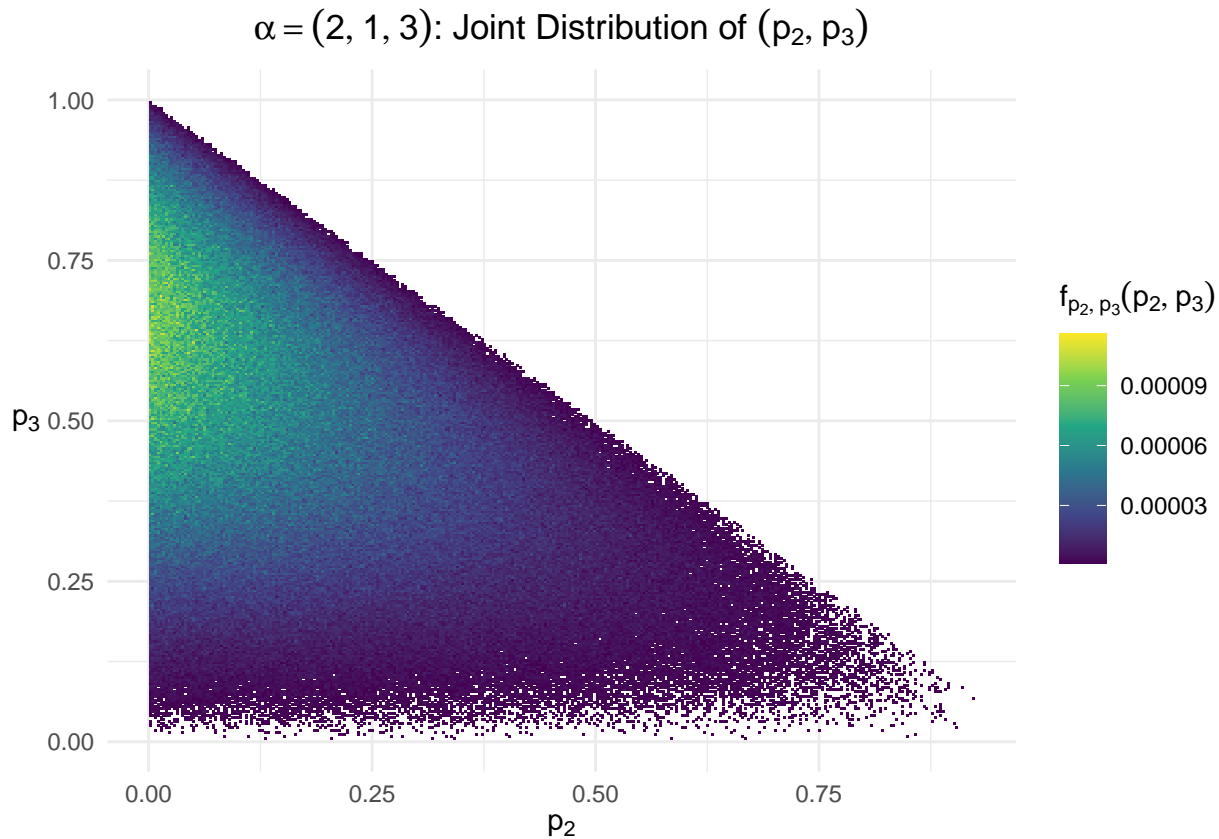
We see that p_1 is more often than not much larger than p_2 as expected.

```
p_comp %>%
  ggplot(aes(x = p1, y = p3)) +
  geom_bin2d(aes(fill = after_stat(density)),
    bins = 300) +
  scale_fill_viridis_c() +
  labs(x = TeX("$p_1$"),
    y = TeX("$p_3$"),
    fill = TeX("$f_{p_1,p_3}(p_1,p_3)$"),
    title = TeX("$\alpha = (2, 1, 3)$: Joint Distribution of $(p_1, p_3)$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
    axis.title = element_text(color = "black"),
    axis.title.y = element_text(angle = 0, vjust = 0.5))
```



We see that p_1 is more often than not slightly smaller than p_3 as expected.


```
p_comp %>%
  ggplot(aes(x = p2, y = p3)) +
  geom_bin2d(aes(fill = after_stat(density)),
    bins = 300) +
  scale_fill_viridis_c() +
  labs(x = TeX("$p_2$"),
    y = TeX("$p_3$"),
    fill = TeX("$f_{p_2,p_3}(p_2,p_3)$"),
    title = TeX("$\\alpha = (2, 1, 3)$: Joint Distribution of $(p_2, p_3)$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
    axis.title = element_text(color = "black"),
    axis.title.y = element_text(angle = 0, vjust = 0.5))
```



We see that p_2 is more often than not significantly smaller than p_3 as expected.

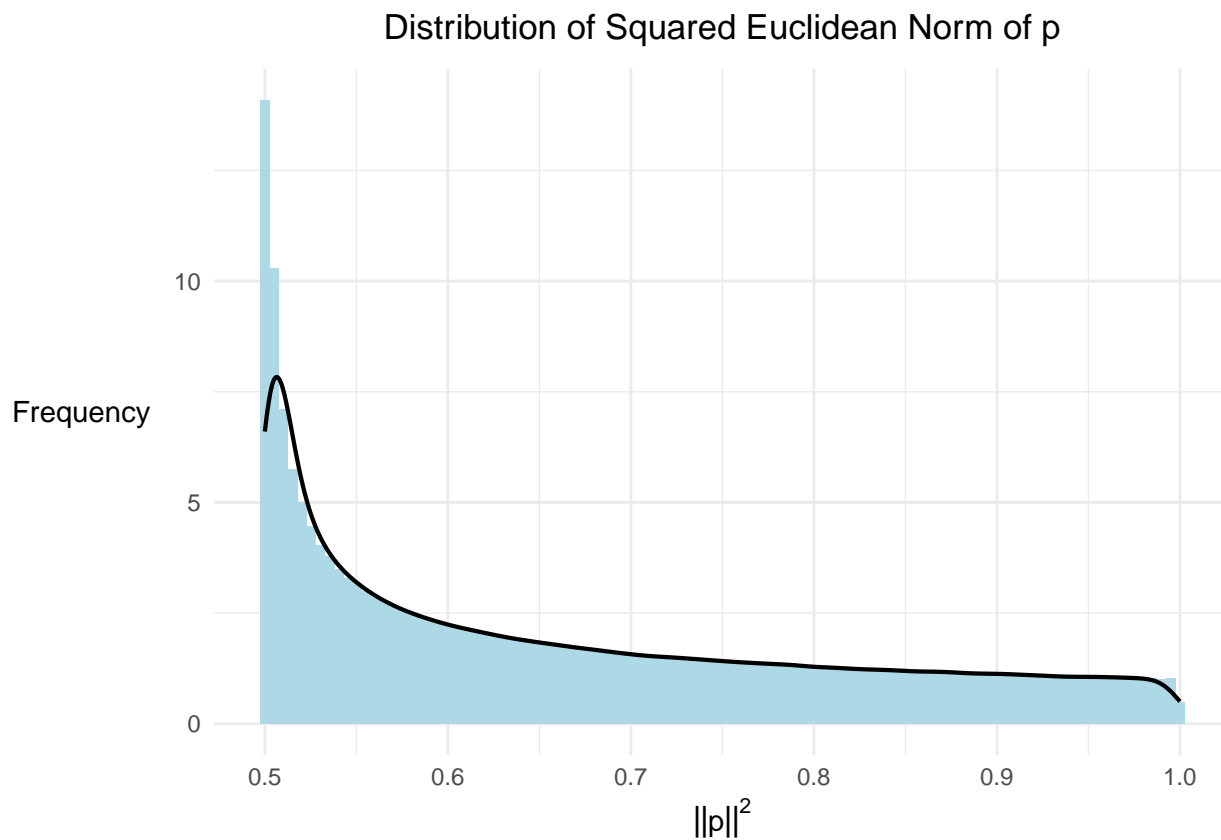
Examining Distribution of the Square of the Euclidean Norm

Below are some graphs to analyze the distribution of $\|p\|^2$.

- First $n = 2$:

```
probs <- prob_generator(N = 2, m = 1000000)
sq_norms <- data.frame(sq_norm = colSums(probs^2))

sq_norms %>%
  ggplot(aes(x = sq_norm)) +
  geom_histogram(aes(y = after_stat(density)),
    fill = "lightblue",
    bins = 100) +
  geom_density(col = "black",
    linewidth = 0.75) +
  labs(x = TeX("$\\|p\\|^2$"),
    y = "Frequency",
    title = TeX("Distribution of Squared Euclidean Norm of $p$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
    axis.title = element_text(color = "black"),
    axis.title.y = element_text(angle = 0, vjust = 0.5))
```

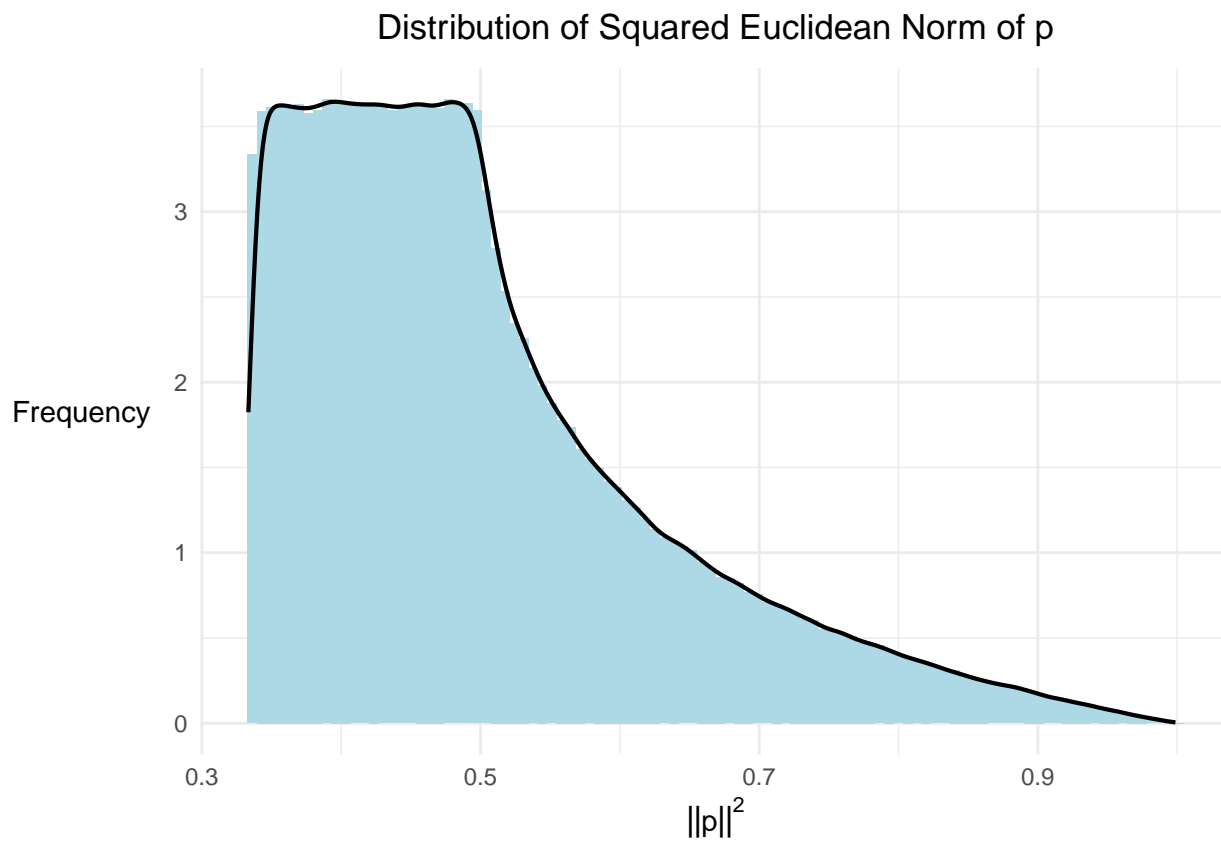


Comment.

- Now $n = 3$:

```
probs <- prob_generator(N = 3, m = 1000000)
sq_norms <- data.frame(sq_norm = colSums(probs^2))

sq_norms %>%
  ggplot(aes(x = sq_norm)) +
  geom_histogram(aes(y = after_stat(density)),
    fill = "lightblue",
    bins = 100) +
  geom_density(col = "black",
    linewidth = 0.75) +
  labs(x = TeX("$||p||^2$"),
    y = "Frequency ",
    title = TeX("Distribution of Squared Euclidean Norm of $p$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
    axis.title = element_text(color = "black"),
    axis.title.y = element_text(angle = 0, vjust = 0.5))
```

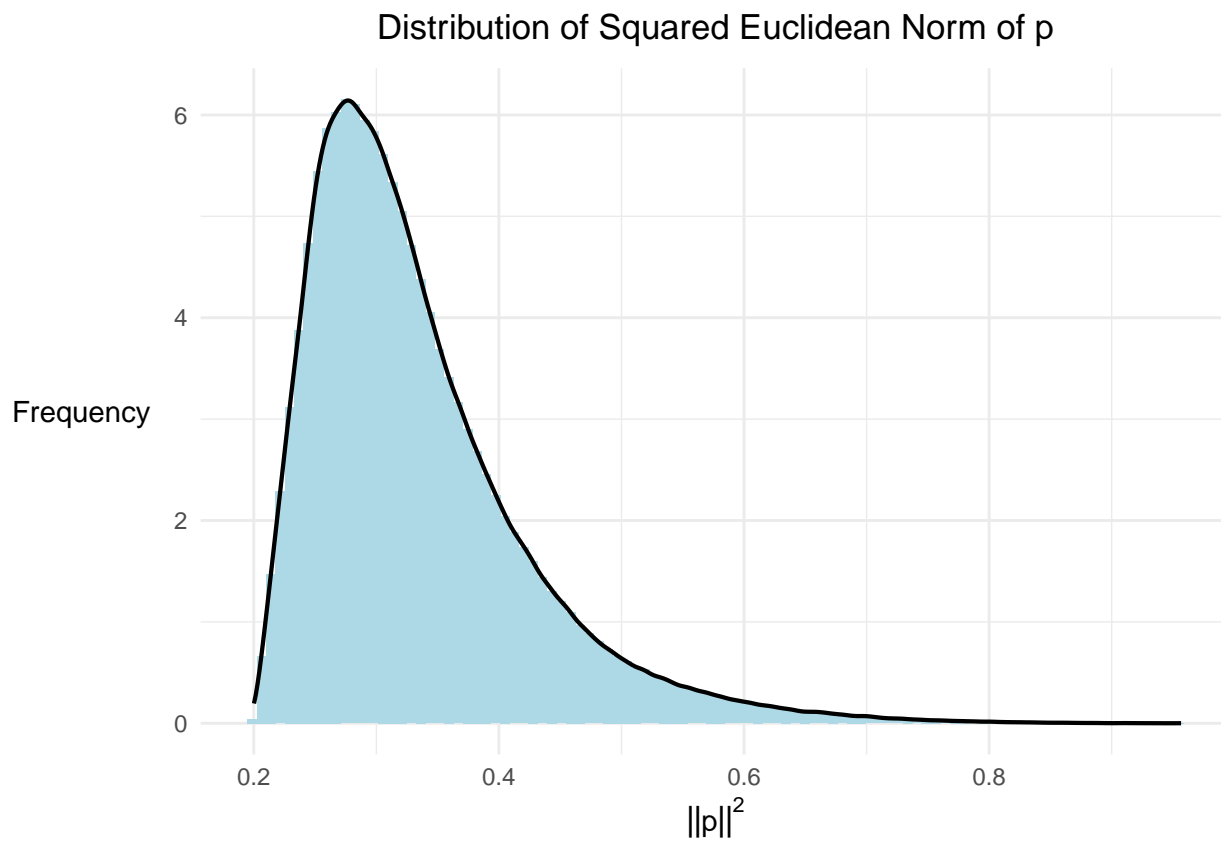


Comment.

- Now $n = 5$:

```
probs <- prob_generator(N = 5, m = 1000000)
sq_norms <- data.frame(sq_norm = colSums(probs^2))

sq_norms %>%
  ggplot(aes(x = sq_norm)) +
  geom_histogram(aes(y = after_stat(density)),
    fill = "lightblue",
    bins = 100) +
  geom_density(col = "black",
    linewidth = 0.75) +
  labs(x = TeX("$||p||^2$"),
    y = "Frequency ",
    title = TeX("Distribution of Squared Euclidean Norm of $p$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
    axis.title = element_text(color = "black"),
    axis.title.y = element_text(angle = 0, vjust = 0.5))
```

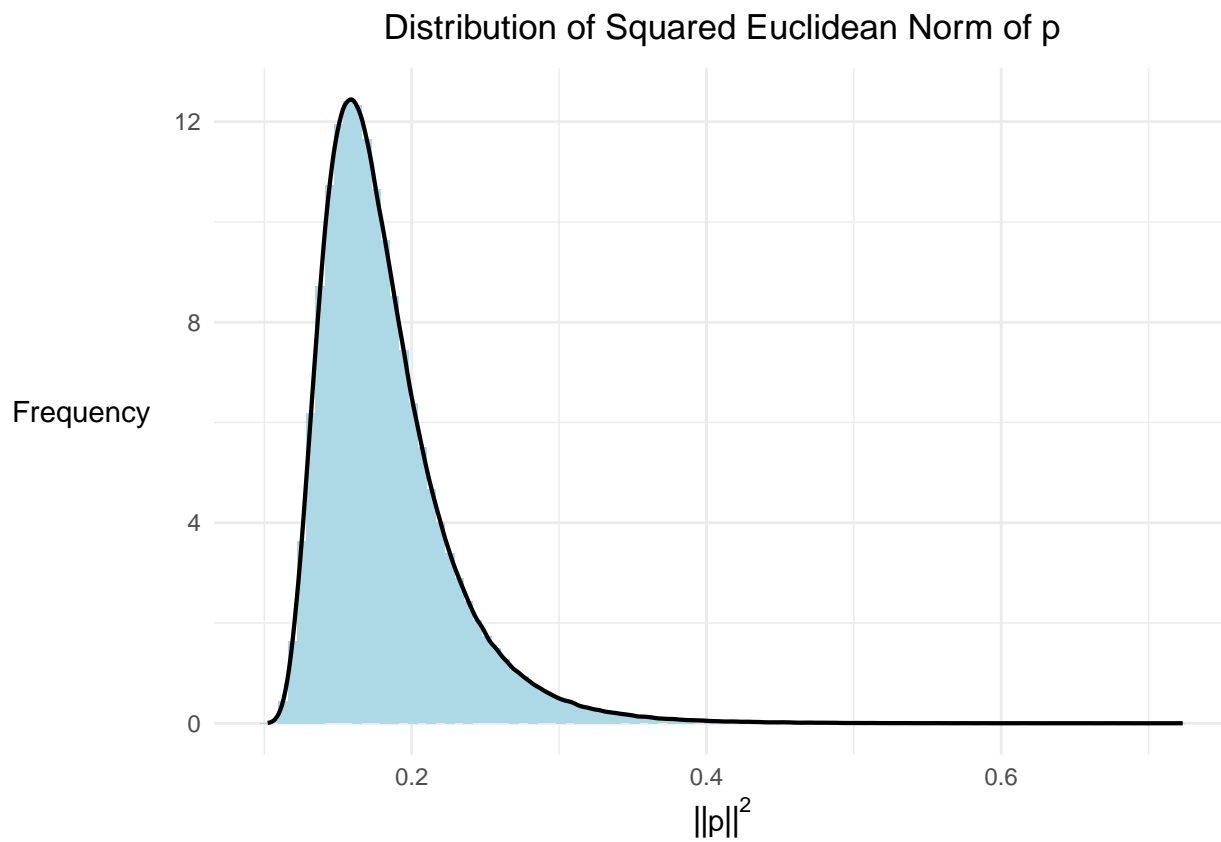


Comment.

- Now $n = 10$:

```
probs <- prob_generator(N = 10, m = 1000000)
sq_norms <- data.frame(sq_norm = colSums(probs^2))

sq_norms %>%
  ggplot(aes(x = sq_norm)) +
  geom_histogram(aes(y = after_stat(density)),
    fill = "lightblue",
    bins = 100) +
  geom_density(col = "black",
    linewidth = 0.75) +
  labs(x = TeX("$||p||^2$"),
    y = "Frequency ",
    title = TeX("Distribution of Squared Euclidean Norm of $p$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
    axis.title = element_text(color = "black"),
    axis.title.y = element_text(angle = 0, vjust = 0.5))
```

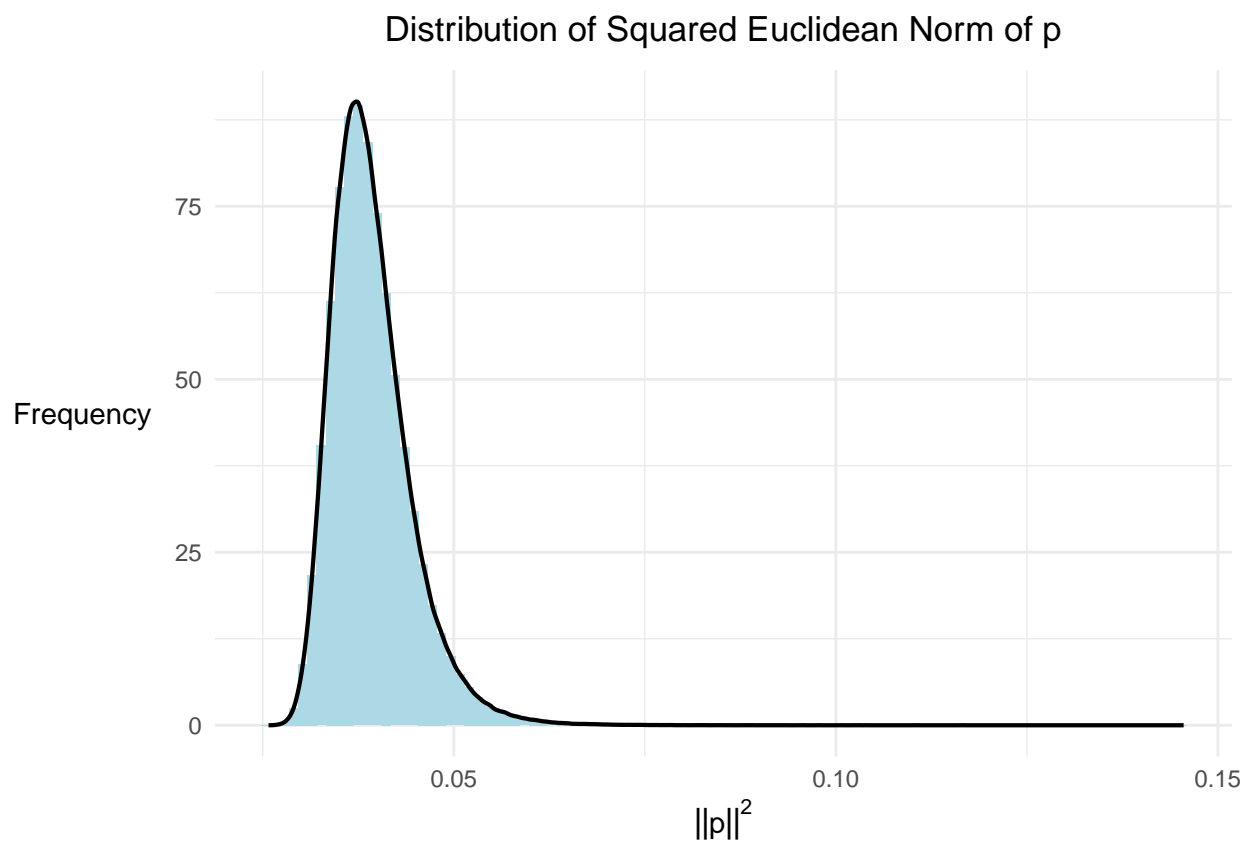


Comment.

- Now $n = 50$:

```
probs <- prob_generator(N = 50, m = 1000000)
sq_norms <- data.frame(sq_norm = colSums(probs^2))

sq_norms %>%
  ggplot(aes(x = sq_norm)) +
  geom_histogram(aes(y = after_stat(density)),
    fill = "lightblue",
    bins = 100) +
  geom_density(col = "black",
    linewidth = 0.75) +
  labs(x = TeX("$||p||^2$"),
    y = "Frequency ",
    title = TeX("Distribution of Squared Euclidean Norm of $p$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
    axis.title = element_text(color = "black"),
    axis.title.y = element_text(angle = 0, vjust = 0.5))
```

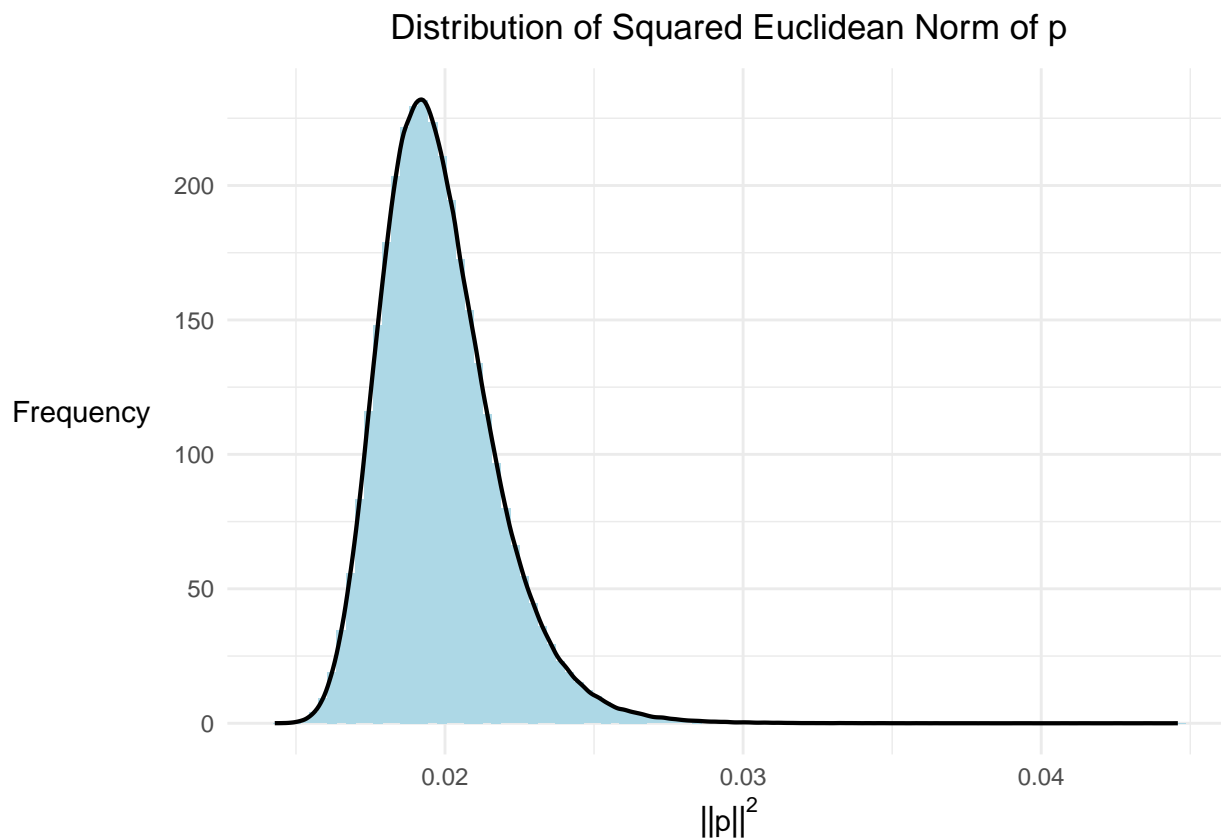


Comment.

- Now $n = 100$:

```
probs <- prob_generator(N = 100, m = 1000000)
sq_norms <- data.frame(sq_norm = colSums(probs^2))

sq_norms %>%
  ggplot(aes(x = sq_norm)) +
  geom_histogram(aes(y = after_stat(density)),
    fill = "lightblue",
    bins = 100) +
  geom_density(col = "black",
    linewidth = 0.75) +
  labs(x = TeX("$||p||^2$"),
    y = "Frequency ",
    title = TeX("Distribution of Squared Euclidean Norm of $p$")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
    axis.title = element_text(color = "black"),
    axis.title.y = element_text(angle = 0, vjust = 0.5))
```

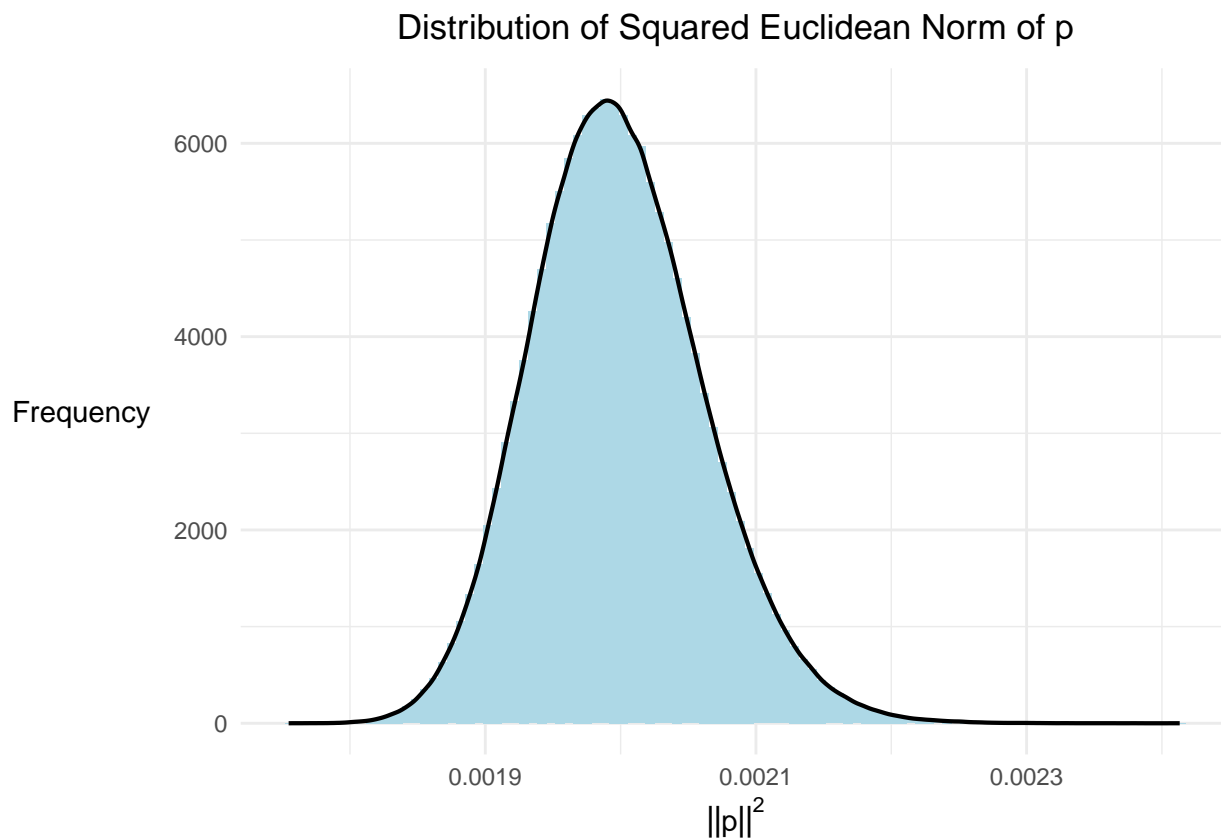


Comment.

- Now $n = 1000$:

```
probs <- prob_generator(N = 1000, m = 1000000)
sq_norms <- data.frame(sq_norm = colSums(probs^2))

sq_norms %>%
  ggplot(aes(x = sq_norm)) +
    geom_histogram(aes(y = after_stat(density)),
                  fill = "lightblue",
                  bins = 100) +
    geom_density(col = "black",
                linewidth = 0.75) +
    labs(x = TeX("$||p||^2$"),
         y = "Frequency ",
         title = TeX("Distribution of Squared Euclidean Norm of $p$")) +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5),
          axis.title = element_text(color = "black"),
          axis.title.y = element_text(angle = 0, vjust = 0.5))
```



Comment.