Midterm Project

Ethan Alteza, Barron Bronson, Holden Ellis, Charlotte Huang

2026-02-09

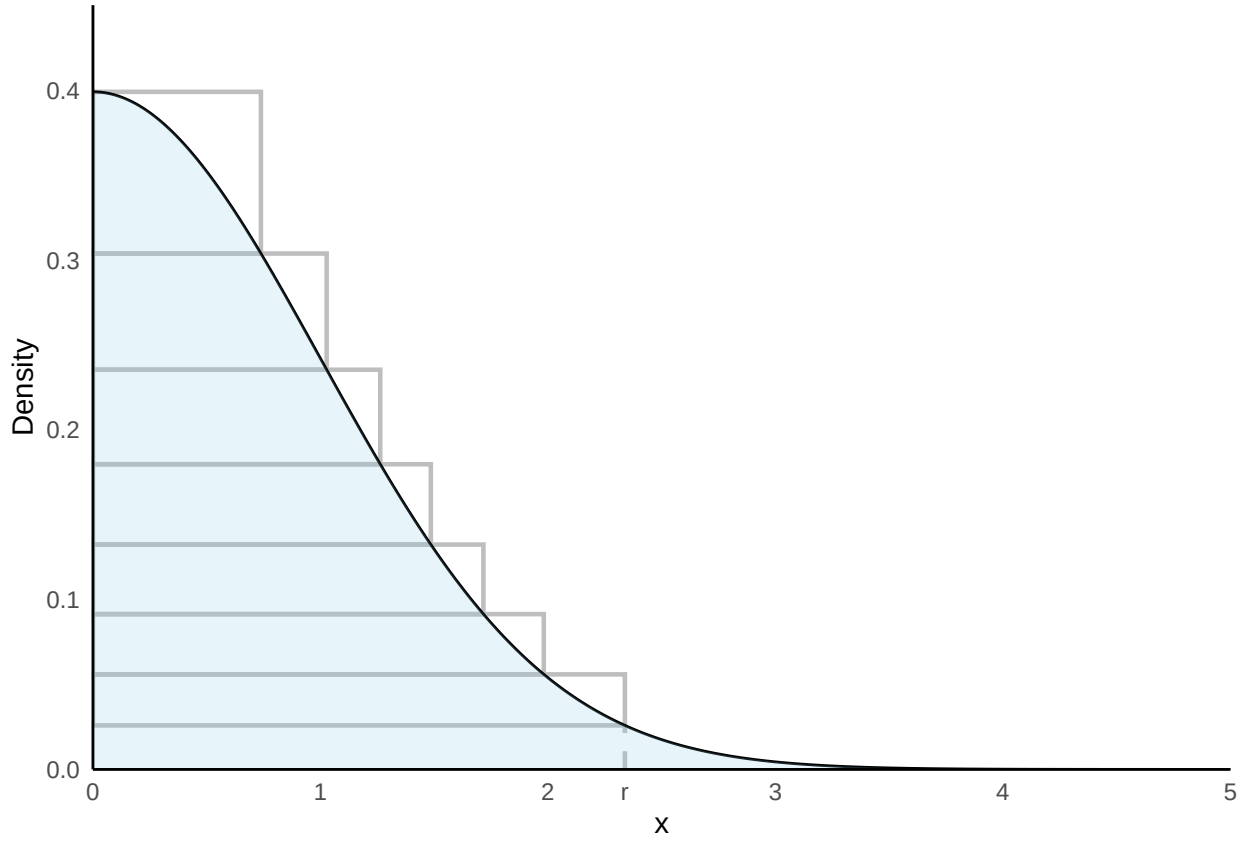
Write Up

https://docs.google.com/document/d/1HZTEfuxbhKEsMSGoFqDvulxrU5hy9rIVp31Wf9el38g/edit?usp=sharing

```
source("Ziggurat.R")
source("BoxMuller.R") # if anyone wants to compare, much faster because of how R runs code
```

```
# build the table for the Ziggurat
x <- zigtable(function(x) {1/sqrt(2*pi) * exp(-x**2/2)}, function(y) {sqrt(-2*log(y*sqrt(2*pi)))},8)$x
y <- dnorm(x)
r_val <- tail(x, 1)

# Dataframe to draw rectangles
df_rect <- data.frame(
  xmin = 0,
  xmax = x[-1],          # x[2]...x[8]
  ymin = y[-1],          # y[2]...y[8]
  ymax = y[-length(y)]   # y[1]...y[7]
)

ggplot() +
  # draw rectangles
  geom_rect(data = df_rect,
            aes(xmin = xmin, xmax = xmax, ymin = ymin, ymax = ymax),
            fill = NA, color = "grey", linewidth = 0.8) +

  # dotted line at x=r
  geom_segment(aes(x = r_val, xend = r_val, y = 0, yend = tail(y, 1)),
               linetype = "dashed", color = "grey", linewidth = 0.8) +

  # draw normal distribution
  stat_function(fun = dnorm, geom = "line", color = "black", n = 1000) +
    # highlight area under the curve
  stat_function(fun = dnorm, geom = "area", fill = "skyblue", alpha = 0.2, xlim = c(0, 5)) +

  # style axes
  scale_x_continuous(
    limits = c(0, 5),
    expand = c(0, 0),
    breaks = sort(c(0:5, r_val)),
```

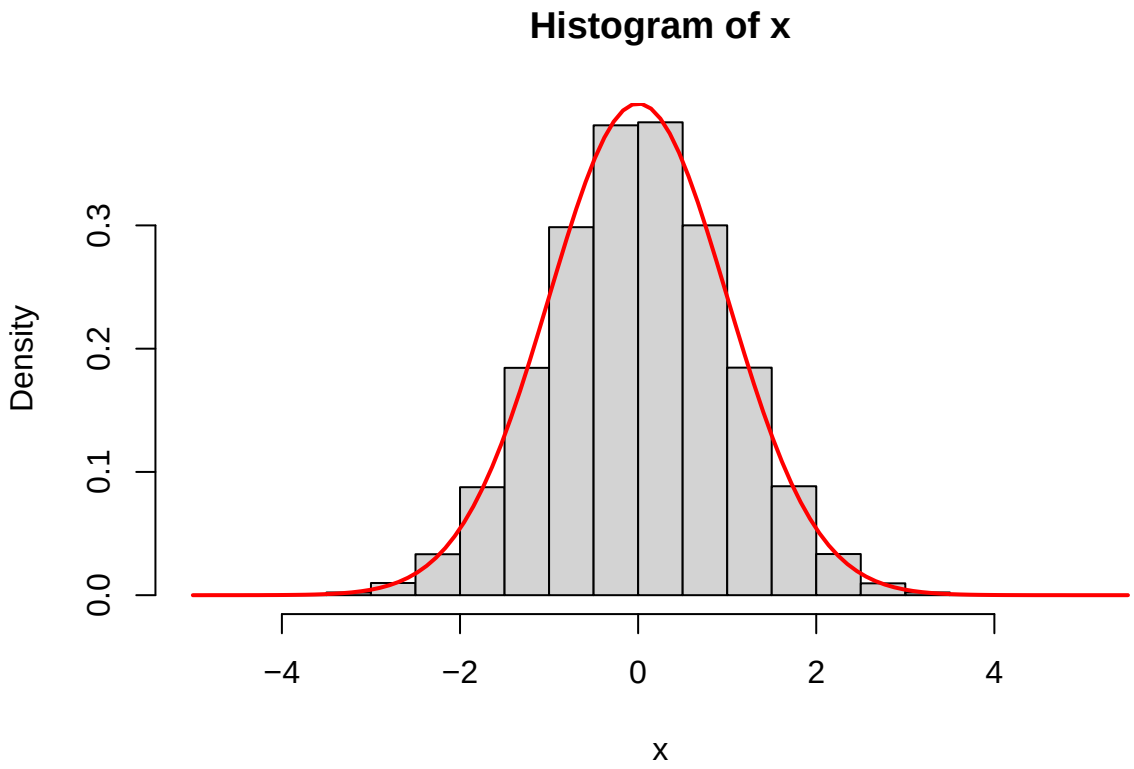
```
    labels = function(b) {
      # Use a small epsilon for float comparison
      ifelse(abs(b - r_val) < 1e-6, "r", b)
    }
  ) +
  scale_y_continuous(limits = c(0, 0.45), expand = c(0, 0)) +

  # labels and plot styles
  labs(x = "x", y = "Density") +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),
    axis.line = element_line(color = "black")
  )
```




```
zigtable(function(x) { exp(-x) }, function(y) { -log(y) },8)
```

```
## $x
## [1] 0.0000000 0.4338930 0.7888599 1.1387148 1.5163812 1.9560291 2.5166676
## [8] 3.3489677
##
## $v
## [1] 0.1527383
```

```r
x <- rnormzig(1000000)
hist(x, freq=FALSE)
curve(dnorm(x), add=TRUE, col="red", lwd=2) # looks good
```




```r
set.seed(777)

#testing: alpha = 0.01
samples1 <- rnormzig(4999)
print("4999 Samples")
```

```
## [1] "4999 Samples"
```

```r
shapiro.test(samples1) # no evidence against normal; p-value > alpha = 0.01
```

```
##
##  Shapiro-Wilk normality test
##
## data:  samples1
## W = 0.99954, p-value = 0.2852
```

```r
ks.test(samples1, "pnorm", mean=mean(samples1), sd=sd(samples1)) # same
```

```
##
```

```
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  samples1
## D = 0.0083523, p-value = 0.8766
## alternative hypothesis: two-sided
```

```
print("1,000,000 ziggurat samples")
```

```
## [1] "1,000,000 ziggurat samples"
```

```
# testing 1 mil samples...
ks.test(x, "pnorm", mean=mean(x), sd=sd(x)) # passed
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.0006743, p-value = 0.7535
## alternative hypothesis: two-sided
```

```
mean(x) # close to 0
```

```
## [1] 0.001408725
```

```
sd(x)   # close to 1
```

```
## [1] 1.000688
```

```
print("1,000,000 rnorm samples")
```

```
## [1] "1,000,000 rnorm samples"
```

```
# additional test:
x2 <- rnorm(1000000)
ks.test(x2, "pnorm", mean=mean(x2), sd=sd(x2)) # passed
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  x2
## D = 0.00069352, p-value = 0.722
## alternative hypothesis: two-sided
```

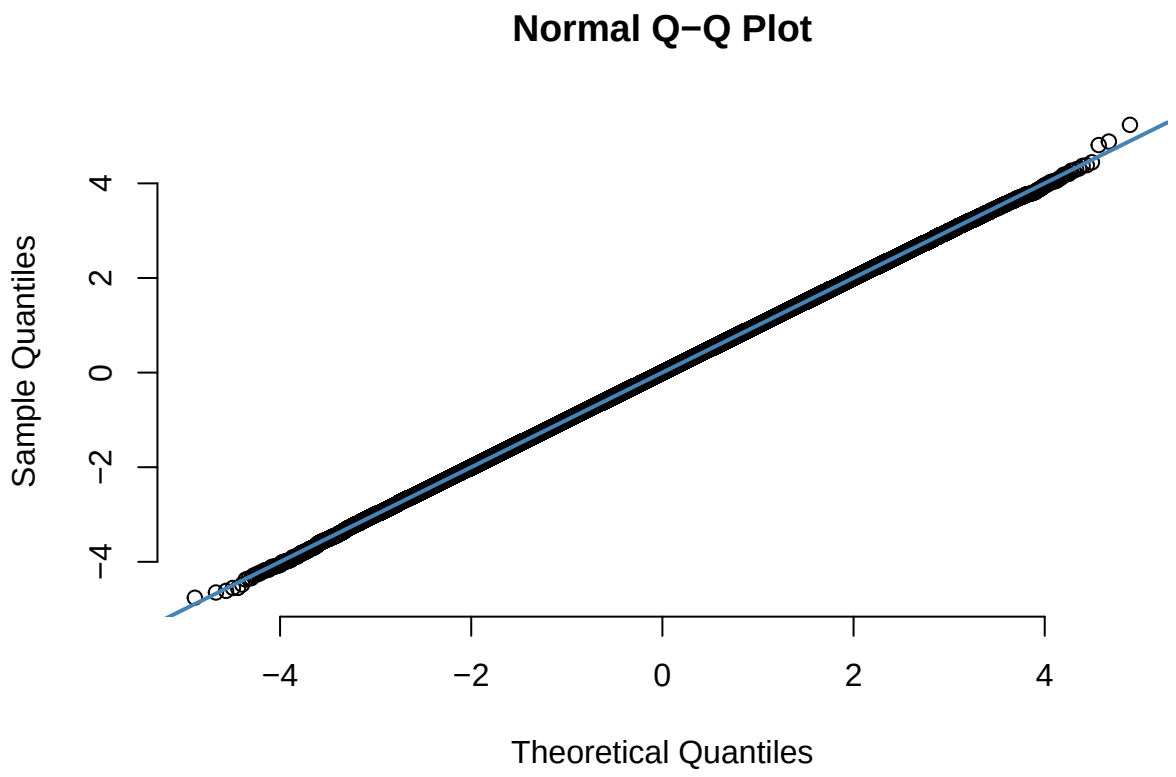
```
mean(x2)  # close to 0
```

```
## [1] 0.000669734
```

```r
sd(x2)    # close to 1
```

```
## [1] 1.001078
```

```r
# QQ plot
qqnorm(x, pch = 1, frame = FALSE)
qqline(x, col = "steelblue", lwd = 2) # basically normal
```



Assume we generate 100000 with Ziggurat method and they're actually normal, using rnorm() for now

A chi-squared goodness of fit test (Knuth 1997) is used to check the distribution of the random variables generated. This test quantizes the horizontal axis of the probability density function into k bins and derives a single value as a quality metric from the determined actual and expected number of samples in each bin.

$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(Y_i - tp_i)^2}{tp_i}$

Where t = number of observation, p_i = probability that each observation falls into the the category i Y_i = the number of observation that actually do fall into the category i.

```r
set.seed(999)

n = 100000
samples <- rnorm(n)

chi_squared_test <- function(x, k){
  t <- length(x) # number of observations
  bins <- seq(-7, 7, length.out = k+1) #200 bins spaced uniformly over data edge of min and max

  Yi <- hist(x, breaks = bins, plot = FALSE)$counts # observed counts
  # expected probabilites (p_i) of standard normal
  p_i <- numeric(k)
  for (i in 1:k) {
    p_i[i] <- pnorm(bins[i+1]) - pnorm(bins[i])
}
  # using computing chi square using equation from the paper
  chi_sq <- sum((Yi - t * p_i)^2 / (t * p_i))
  df <- k - 1
  return(data.frame(chi_square = chi_sq,
                    df = df))
}
```

```r
set.seed(777)

k <- 200
results <- chi_squared_test(x, k)
results2 <- chi_squared_test(x2, k)

results$Method <- "Ziggurat"
results2$Method <- "Rnorm"

combined <- rbind(results, results2)
combined <- combined[, c("Method", setdiff(names(combined), "Method"))]
kable(combined, caption = "Chi-squared value: Ziggurat vs Rnorm")
```

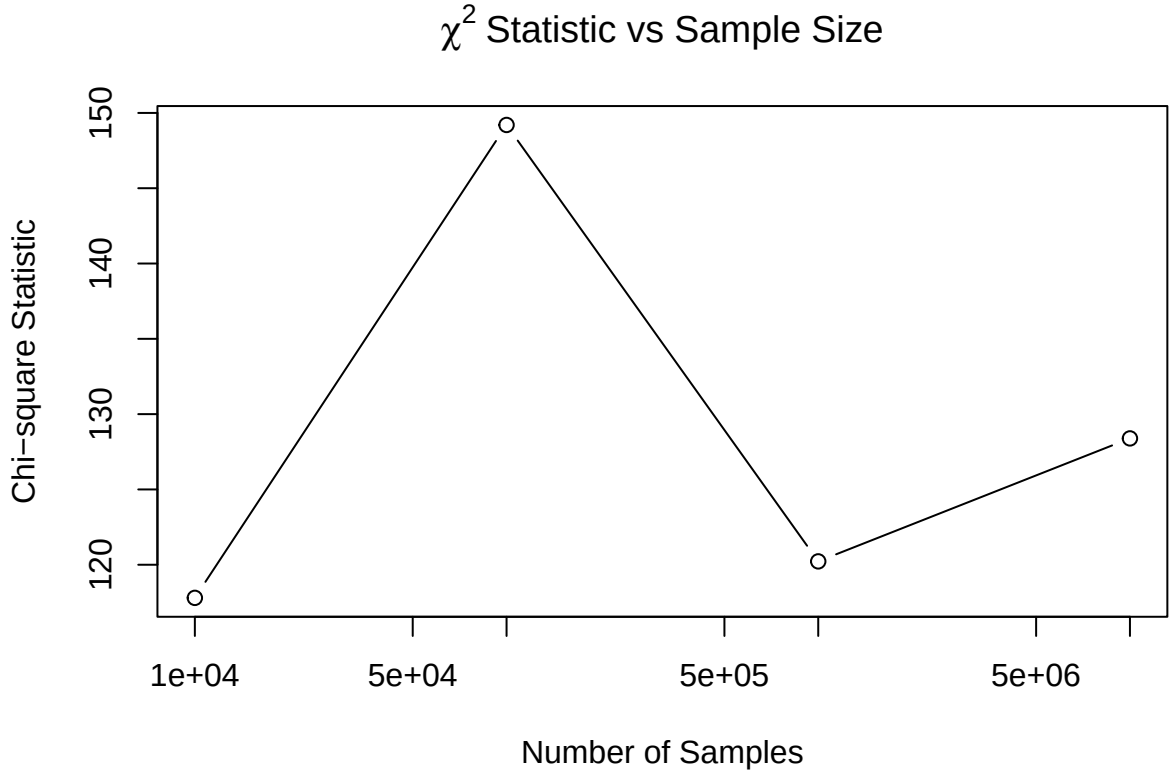
Table 1: Chi-squared value: Ziggurat vs Rnorm

Method	chi_square	df
Ziggurat	174.2026	199
Rnorm	184.4201	199

The critical value for a chi-square test with 199 degrees of freedom at 95% confidence level ($\alpha = 0.05$) is 232.912. (using a look up table on https://www.medcalc.org/en/manual/chi-square-table.php)

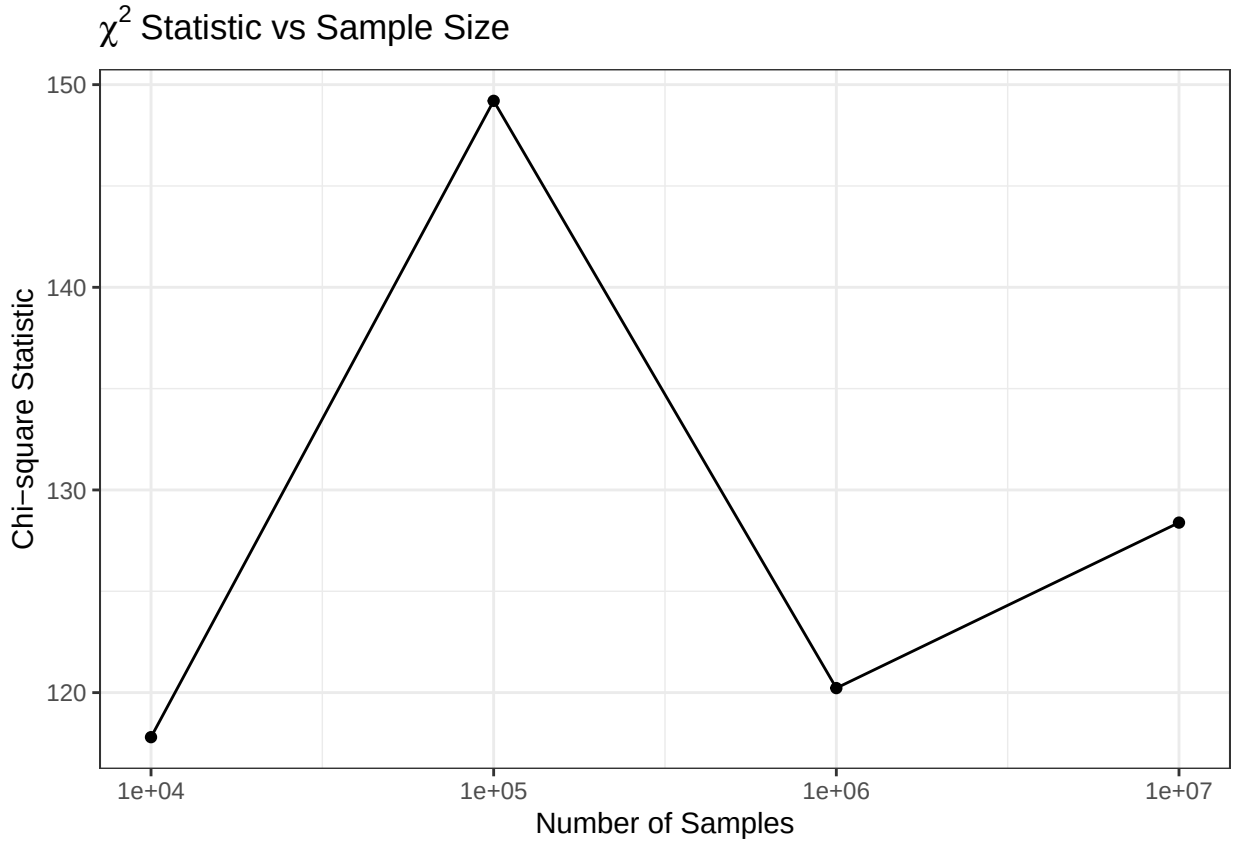
```r
n <- c(10000, 100000, 1000000, 10000000)
set.seed(123)
k <- 200
value <- numeric(length(n))
for(i in 1:length(n)){
  vectors <- rnormzig(n[i])
  chi_sq <- chi_squared_test(vectors, k)
```

```
  value[i] <- chi_sq$chi_square

}

plot(n, value,
     type = "b",
     xlab = "Number of Samples",
     ylab = "Chi-square Statistic",
     main = expression(chi^2 ~ "Statistic vs Sample Size"),
     log = "x")
```




```
df <- data.frame(
  n = n,
  chi_square = value
)

ggplot(df, aes(x = n, y = chi_square)) +
  geom_line() +
  geom_point() +
  scale_x_log10() +
  labs(
    x = "Number of Samples",
    y = "Chi-square Statistic",
    title = expression(chi^2 ~ "Statistic vs Sample Size")
  ) +
```

```
  theme_bw()
```




```
set.seed(123)
test <- rnormzig(1000000)
test_res <- chi_squared_test(test, k)
test_res
```

```
##   chi_square  df
## 1   119.9745 199
```