Sampling Distribution of the Sample Proportion

R Simulation Across Sample Sizes and Population Proportions

1. Objective

This report uses R to simulate and visualize the sampling distribution of the sample proportion (p■, read "p-hat") for every combination of sample size $n \in \{1, 10, 30, 50, 100\}$ and true population proportion $p \in \{0.5, 0.7, 0.1\}$. For each of the 15 combinations, 10,000 random samples were drawn from a Bernoulli(p) population and the resulting proportion of successes was recorded, producing an empirical sampling distribution that is then plotted as a histogram.

2. R Code

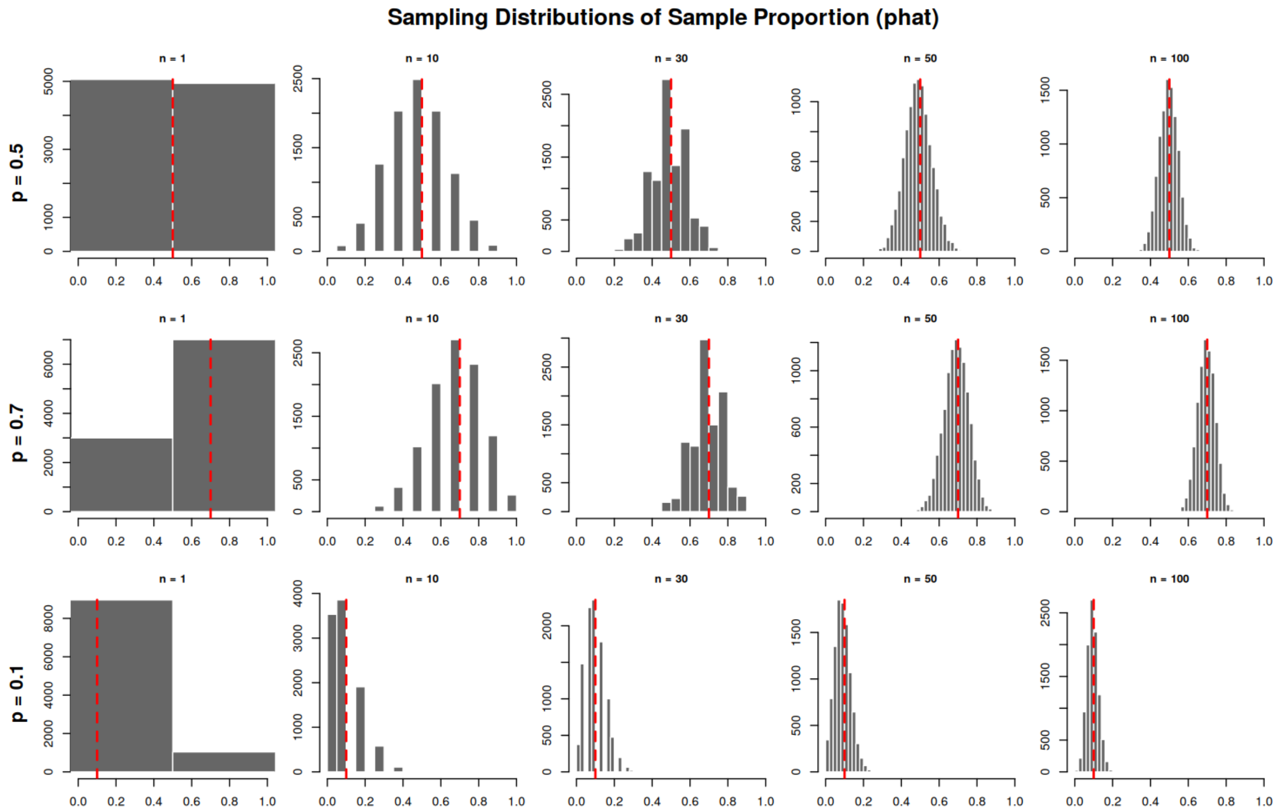
The full commented R script (proportion_distribution.R) is reproduced below.

```
## ==========================================================================
## Sampling Distribution of the Sample Proportion (p-hat)
## --------------------------------------------------------------------------
## Goal:
##   For every combination of sample size n and true population proportion p,
##   simulate many random samples of size n from a Bernoulli(p) population,
##   compute the sample proportion (phat = number of successes / n) for each
##   simulated sample, and plot the resulting distribution of phat.
##
##   n : 1, 10, 30, 50, 100
##   p : 0.5, 0.7, 0.1
##
## This recreates, via simulation, the classic result that the sampling
## distribution of phat is centered at p, has standard deviation
## sqrt(p*(1-p)/n) (the "standard error"), and becomes progressively more
## symmetric / bell-shaped (Normal-like) as n grows -- the Central Limit
## Theorem in action.
## ==========================================================================

set.seed(123)                 # reproducibility: same "random" draws every run

n_values &lt;- c(1, 10, 30, 50, 100)   # sample sizes to examine (columns)
p_values &lt;- c(0.5, 0.7, 0.1)        # true proportions to examine (rows)
n_sim    &lt;- 10000                   # number of simulated samples per (n,p) cell

## --------------------------------------------------------------------------
## Function: simulate the sampling distribution of phat for one (n, p) pair
## --------------------------------------------------------------------------
simulate_phat &lt;- function(n, p, n_sim = 10000) {
  # rbinom(n_sim, size = n, prob = p) draws n_sim binomial(n,p) counts,
  # i.e. for each of the n_sim "experiments" we take a sample of n Bernoulli
  # trials and count the successes. Dividing by n converts counts -&gt; proportions.
  successes &lt;- rbinom(n_sim, size = n, prob = p)
  phat &lt;- successes / n
  return(phat)
}

## --------------------------------------------------------------------------
## Build the 3 (p) x 5 (n) grid of histograms, matching the requested layout:
##   rows    = p = 0.5, 0.7, 0.1
##   columns = n = 1, 10, 30, 50, 100
## --------------------------------------------------------------------------

# Save the figure to a PNG file (also displays if run interactively)
png("proportion_distribution_grid.png", width = 1500, height = 950, res = 130)

# 3 rows x 5 columns of plots; outer margins reserved for row/column labels
par(mfrow = c(length(p_values), length(n_values)),
    mar   = c(3, 2, 2, 1),     # inner margins per panel
```

```
    oma   = c(0, 4, 3, 0))     # outer margins for shared row/column labels

for (p in p_values) {
  for (n in n_values) {

    phat &lt;- simulate_phat(n, p, n_sim)

    hist(phat,
         breaks = if (n == 1) c(-0.1, 0.5, 1.1) else 20,  # n=1 is just 0/1
         main   = paste0("n = ", n),
         xlab   = "sample proportion",
         xlim   = c(0, 1),
         col    = "grey40",
         border = "white",
         cex.main = 0.9)

    # reference line at the true proportion p
    abline(v = p, col = "red", lwd = 2, lty = 2)
  }
}

# Row labels (p = ...) and overall title, placed in the outer margin
mtext("Sampling Distributions of Sample Proportion (phat)",
      outer = TRUE, side = 3, line = 1, cex = 1.2, font = 2)

for (i in seq_along(p_values)) {
  mtext(paste0("p = ", p_values[i]),
        outer = TRUE, side = 2, line = 1,
        at = 1 - (i - 0.5) / length(p_values),
        cex = 1, font = 2)
}

dev.off()

## -----------------------------------------------------------------------
## Numeric summary: mean and sd of phat for each (n, p), compared with theory
## Theory: E[phat] = p ,  SD[phat] = sqrt(p * (1-p) / n)   (the standard error)
## -----------------------------------------------------------------------
results &lt;- data.frame()
for (p in p_values) {
  for (n in n_values) {
    phat &lt;- simulate_phat(n, p, n_sim)
    results &lt;- rbind(results, data.frame(
      p = p,
      n = n,
      simulated_mean = round(mean(phat), 4),
      theoretical_mean = p,
      simulated_sd   = round(sd(phat), 4),
      theoretical_sd = round(sqrt(p * (1 - p) / n), 4)
    ))
  }
}

cat("\n===== Simulated vs Theoretical Mean/SD of Sample Proportion =====\n")
print(results, row.names = FALSE)

write.csv(results, "proportion_distribution_summary.csv", row.names = FALSE)
```

3. Output: Simulated Sampling Distributions

The figure below shows the 3 × 5 grid of histograms (rows = p, columns = n). The red dashed line in each panel marks the true population proportion p.



4. Numeric Summary: Simulated vs. Theoretical Mean and SD

Theory says E[p■] = p and SD[p■] = √(p(1–p)/n) (the standard error). The table compares these theoretical values to what the simulation actually produced.

p	n	Simulated Mean	Theoretical Mean	Simulated SD	Theoretical SD
0.5	1	0.4958	0.5	0.5	0.5
0.5	10	0.502	0.5	0.1609	0.1581
0.5	30	0.501	0.5	0.091	0.0913
0.5	50	0.5004	0.5	0.07	0.0707
0.5	100	0.4996	0.5	0.0497	0.05
0.7	1	0.7007	0.7	0.458	0.4583
0.7	10	0.7004	0.7	0.1457	0.1449
0.7	30	0.6995	0.7	0.0838	0.0837
0.7	50	0.701	0.7	0.0656	0.0648
0.7	100	0.7	0.7	0.0458	0.0458

0.1	1	0.1013	0.1	0.3017	0.3
0.1	10	0.0993	0.1	0.094	0.0949
0.1	30	0.0996	0.1	0.0546	0.0548
0.1	50	0.1004	0.1	0.0429	0.0424
0.1	100	0.1	0.1	0.0296	0.03

5. Discussion

Effect of increasing n. For every value of p, the histograms become narrower and more symmetric as n increases from 1 to 100. This is exactly what the standard error formula predicts: SD(p■) = √(p(1–p)/n) shrinks as n grows, so the sample proportion clusters more tightly around the true p. The simulated standard deviations in the summary table track the theoretical values closely at every n, confirming the formula empirically.

n = 1 is degenerate. When n = 1, a sample is a single Bernoulli trial, so p■ can only take the value 0 or 1. The histogram is therefore just two spikes (at 0 and 1) whose relative heights equal (1 - p) and p. This is not yet a bell-shaped distribution -- it is the underlying Bernoulli distribution itself, before any averaging has occurred.

Central Limit Theorem in action. As n grows (30, 50, 100), the histograms increasingly resemble a Normal curve. This illustrates the Central Limit Theorem: the distribution of a sample proportion (an average of 0/1 outcomes) approaches Normality as the sample size increases, regardless of the underlying Bernoulli shape.

Effect of p (skewness at small n). For p = 0.5, the distribution is symmetric at every n because successes and failures are equally likely. For p = 0.7 and especially p = 0.1, the distribution is visibly skewed at small n (10, 30) -- right-skewed for p = 0.1, left-skewed for p = 0.7 -- because the distribution is bounded at 0 and 1 and p sits closer to one boundary. As n grows, the skewness fades and all three rows converge toward symmetric, bell-shaped distributions centered on their respective p.

Practical rule of thumb. The visible skew for p = 0.1 at small n is the simulation-based version of the common textbook guideline that the Normal approximation to a sample proportion is only reliable when both n*p and n*(1-p) are reasonably large (commonly, at least 5-10). For p = 0.1 and n = 10, n*p = 1, which explains why that particular panel still looks noticeably skewed.

6. Conclusion

The simulation confirms the theoretical behavior of the sampling distribution of a proportion: it is centered at the true p, its spread shrinks proportionally to 1/√n, and its shape converges to a Normal distribution as n increases, with the rate of that convergence depending on how close p is to the boundaries 0 and 1.