Simulation basics

Agenda

- Brief review of simulation basics (we will expand on this later)
- Inverse transform sampling for joint distributions
- Motivation for causal inference (if we have time to start)

Inverse transform sampling

Key result:

- Suppose $U \sim Unif(0, 1)$
- Suppose $F_X(\cdot)$ is a cumulative distribution function (cdf) for (arbitrary) X
  - i.e., $F_X(x) = P(X < x)$
- Suppose $F_X^{-1}(\cdot)$ is the inverse cdf
  - i.e., $F_X^{-1}(u) = \left\{x \ \text{ s.t. } \ P(X < x) = u\right\}$.

Inverse transform sampling

- Define $Z = F_X^{-1}(U)$

  Then:

- $F_Z(x) = P\left(F_X^{-1}(U) < x\right) = F_X(x).$
- i.e., $Z = F_X^{-1}(U)$ is distributed the same as X.

Inverse Transform sampling

Inverse Transform Sampling: Bernoulli(p)

Number of samples (n)

200 5,000

10 510 1,010 2,010 3,010 4,010 5,000

Success probability / threshold (p)

0.01 0.3 0.99

0.01 0.11 0.21 0.31 0.41 0.51 0.61 0.71 0.81 0.91 0.99

Resimulate

Inverse transform for Bernoulli: draw U ~ Unif(0,1), set X = 1{U ≤ p}.

Inverse transform sampling

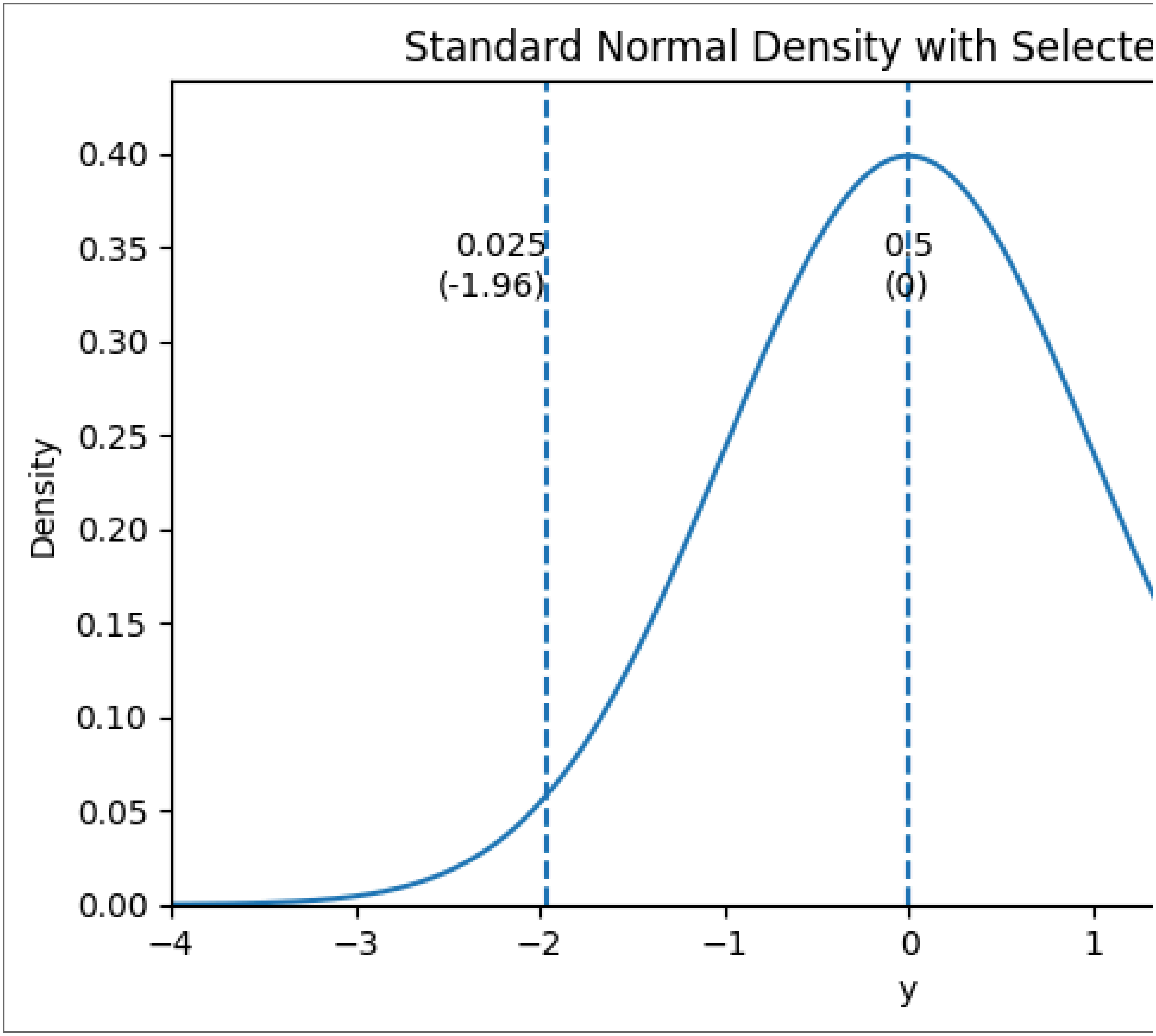
Why does this work?

- The key property of $Unif(0, 1)$ variables (e.g., U) is:
- $P(U < u) = u$ for $u \in [0, 1]$.
- Now consider some r.v. Y with cdf F_Y taking values on the real number line, $\mathbb{R}$.
- Let's say it is a standard normal variable $\mathcal{N}(0, 1)$. (disclaimer, normal variables don't have a closed form cdf).

Inverse transform sampling

-Thus (and remembering that F_Y^{-1} is monotonic increasing):

- $P\left(F_Y^{-1}(U) < y\right) = P\left(U < F_Y(y)\right) = F_Y(y).$
- e.g., $P\left(F_Y^{-1}(U) < -1.96\right) = P(U < 0.025) = 0.025 = P(Y <$
- e.g., $P\left(F_Y^{-1}(U) < 0\right) = P(U < 0.5) = 0.5 = P(Y < 0)$



Standard Normal Density with Selecte
0.025
(-1.96)
0.5
(0)
Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−3
−2
−1
0
1
y

Inverse transform sampling

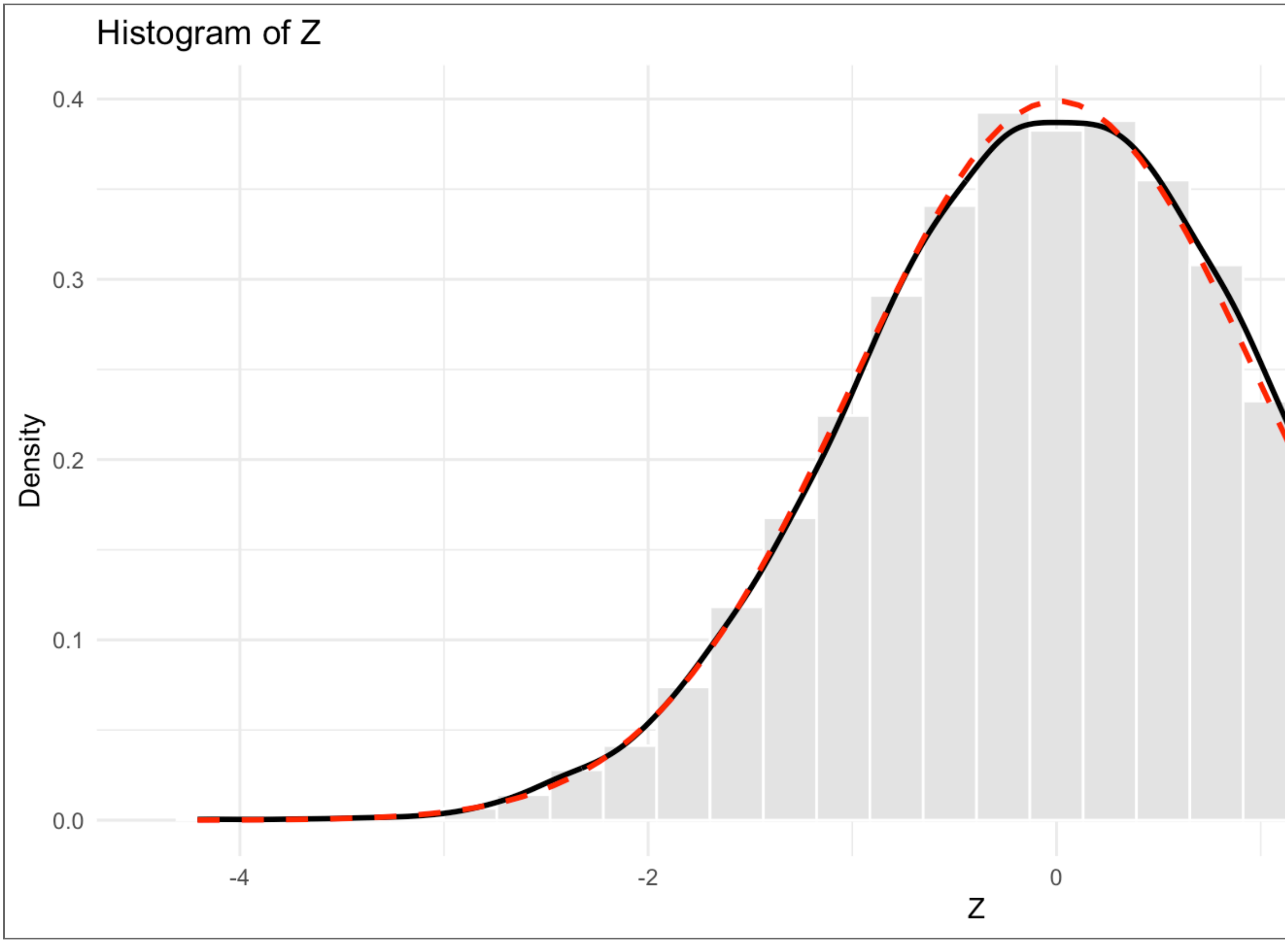
```r
U<-runif(10000, 0,1)
head(U, 25)
```

```
 [1] 0.18164945 0.96184115 0.86116577 0.88099929 0.35504869 0.83269749
 [7] 0.12277281 0.62992643 0.15041318 0.06183721 0.44550188 0.80218493
[13] 0.42490195 0.65054788 0.68081540 0.01277009 0.68145143 0.11766414
[19] 0.29957001 0.38112449 0.16307337 0.49388402 0.80051250 0.33180824
[25] 0.31990931
```

```r
Z<-qnorm(U, 0, 1)
head(Z, 25)
```

```
 [1] -0.9090970  1.7724632  1.0855719  1.1799969 -0.3717253  0.9648798
 [7] -1.1612368  0.3316585 -1.0346629 -1.5395322 -0.1370342  0.8494515
[13] -0.1893686  0.3868000  0.4699802 -2.2331330  0.4717613 -1.1867448
[19] -0.5256376 -0.3025288 -0.9819048 -0.0153311  0.8434533 -0.4349255
[25] -0.4679524
```

Inverse transform sampling



Sampling for joint distributions

- Consider a collection of variables (a vector) $\mathbf{X} \equiv (X_1, \ldots, X_p) \sim P$.
- Suppose we consider a specific P. How do we simulate a vector $\mathbf{Z} \sim P$.

Sampling for joint distributions

- Recall that the distribution P is specified by the joint density function $f_{\mathbf{X}}(x_1, \ldots, x_p)$.
- A joint density function can be factorized as a product of conditional density functions:

$$f_{\mathbf{X}}(x_1, \ldots, x_p) = \prod_{k=1}^{p} f_{X_k | X_{k-1}, \ldots, X_1}(x_k \mid x_{k-1}, \ldots, x_1).$$

- e.g., $f(x_1, x_2) = f(x_1 \mid x_2) f(x_2)$.

Sampling from joint distributions

Suppose:

- $U_k \sim Unif(0,1)$ are mutually independent, for $k = 1, \ldots, p$.
- $F_k(\cdot)$ is a cdf for X_k given $X_1, \ldots, X_{k-1}$.
- i.e., $F_k(x_k \mid x_{k-1}, \ldots, x_1) = P(X_k < x_k \mid x_{k-1}, \ldots, x_1)$.
- $F_k^{-1}(\cdot)$ is the inverse cdf, i.e.,

$$
\begin{aligned}
&F_k^{-1}(u \mid x_{k-1}, \ldots, x_1) \\
=&\Big\{x_k \ \text{ s.t. } \ P(X_k < x_k \mid x_{k-1}, \ldots, x_1) = u\Big\}.
\end{aligned}
$$

Sampling from joint distributions

- Define $Z_1 = F_1^{-1}(U_1)$
- Iteratively define (from $k = 2, \ldots, p$)

$$Z_k = F_k^{-1}(U_k \mid Z_{k-1}, \ldots, Z_1)$$

Then:

- $F_{\mathbf{Z}}(\mathbf{x}) = F_{\mathbf{X}}(\mathbf{x})$.
  - i.e., $\mathbf{Z}$ is distributed the same as $\mathbf{X}$.

Sampling from joint distributions

Example 1: $\mathbf{X} = (X_1, X_2)$, each binary $\{0, 1\}$ variables.

- $P(X_1 = 1) = 0.5$,
- $P(X_2 = 1 \mid X_1 = x_1) = \begin{cases} 0.25 \text{ if } x_1 = 1 \\ 0.75 \text{ if } x_1 = 0 \end{cases}$

Then:

- $Z_1 = \begin{cases} 1 & \text{if } U_1 < 0.5 \\ 0 & \text{if } U_1 \geq 0.5 \end{cases}$
- $Z_2 = \begin{cases} 1 & \text{if } U_2 < 0.25 \text{ and } Z_1 = 1 \\ 0 & \text{if } U_2 \geq 0.25 \text{ and } Z_1 = 1 \\ 1 & \text{if } U_2 < 0.75 \text{ and } Z_1 = 0 \\ 0 & \text{if } U_2 \geq 0.75 \text{ and } Z_1 = 0 \end{cases}.$

Sampling from joint distributions

```
U1<-runif(10000, 0,1)
U2<-runif(10000, 0,1)
Z1 = qbinom(U1, 1, 0.5)
Z2 = qbinom(U2, 1, 0.75 - Z1*0.5)
prop.table(table(Z1))[2]
```

```
     1
0.5048
```

```
prop.table(table(Z1, Z2), margin=1)[2, ]
```

```
        0         1
0.7436609 0.2563391
```

Sampling from joint distributions

Example 2:

- $\mathbf{X} = (X_1, X_2)$, each normally distributed variables.
- Suppose:
  - $X_1 \sim \mathcal{N}(0, 1)$
  - $X_2 \sim \mathcal{N}(\beta_0 + \beta_1 X_1 + \beta_2 X^2, \sigma^2)$
  - $(\beta_0, \beta_1, \beta_2, \sigma^2) = (0, 0.5, 2, 4)$
- Let Φ^{-1} be the inverse cdf for the standard normal.

Sampling from joint distributions

- $(\beta_0, \beta_1, \beta_2, \sigma^2) = (0, 0.5, 2, 4)$

  Then:

- $Z_1 = \Phi^{-1}(U_1)$
- $Z_2 = 2\Phi^{-1}(U_2) + (0.5Z_1 + 2Z_1^2)$

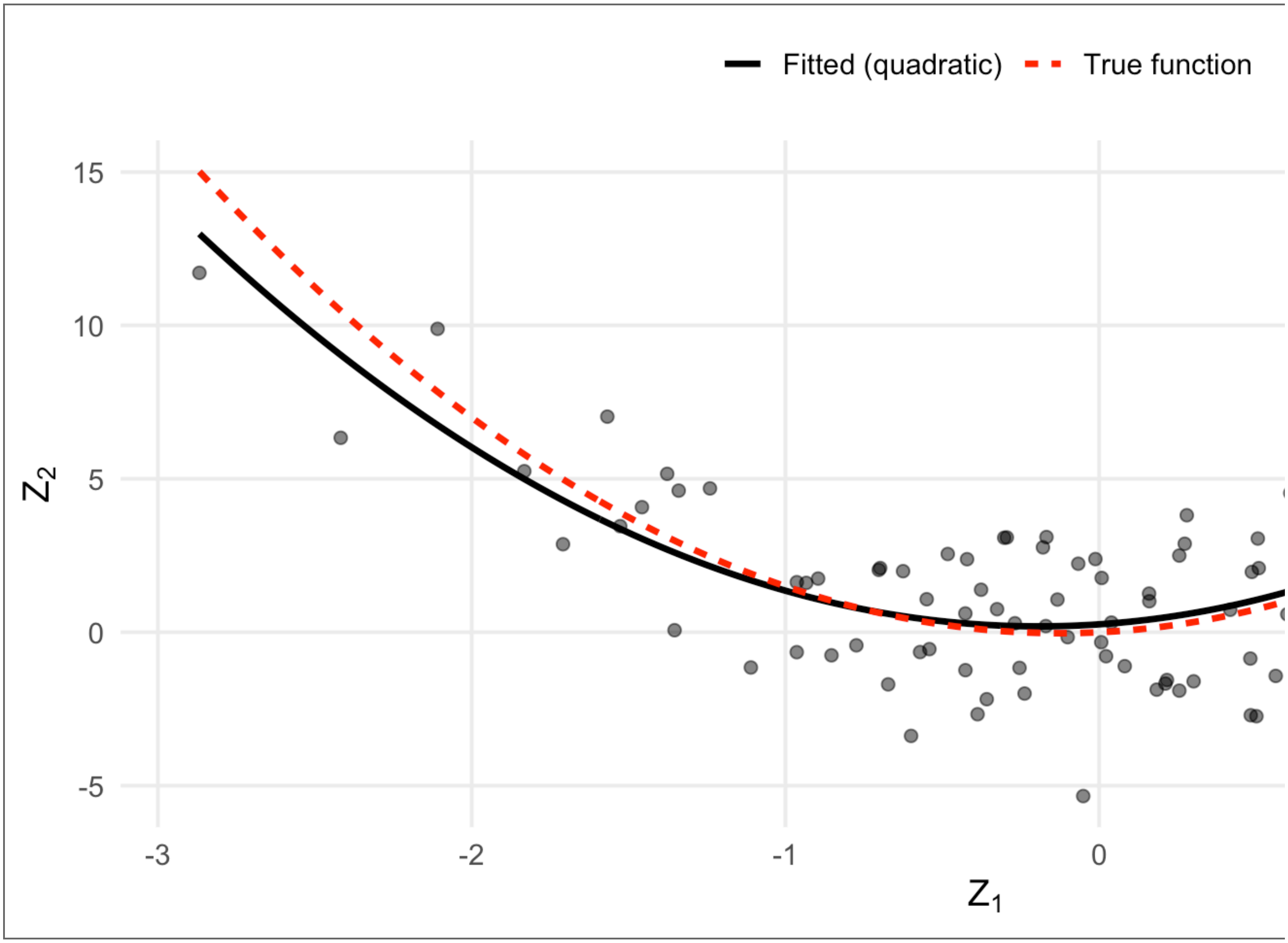
Sampling from joint distributions

```r
U1<-runif(100, 0,1)
U2<-runif(100, 0,1)
Z1 = qnorm(U1, 0, 1)
Z2 = qnorm(U2, 0.5*Z1 + 2*Z1^2, 2)
lm(Z2~Z1 + I(Z1^2))
```

```
Call:
lm(formula = Z2 ~ Z1 + I(Z1^2))

Coefficients:
(Intercept)           Z1      I(Z1^2)
     0.2587       0.6928       1.7890
```

Sampling from joint distributions



Sampling from sampling distributions

- Consider a sample of size $n, \mathbf{X}_n \equiv (X_1, \ldots, X_n)$.
- When data are i.i.d, $f(x_k \mid x_{k-1}, \ldots, x_1) = f(x_k)$.
- This means that simulating samples is easy!

Sampling from sampling distributions

Here we simulate a single sample of size $n = 100$.

```r
n=100
U1<-runif(n)
U2<-runif(n, 0,1)
Z1 = qnorm(U1, 0, 1)
Z2 = qnorm(U2, 0.5*Z1 + 2*Z1^2, 2)
fit <- lm(Z2 ~ Z1 + I(Z1^2), data = df)
coef(fit)["I(Z1^2)"]
```

```
 I(Z1^2)
1.788975
```

```r
summary(fit)$coefficients["I(Z1^2)", "Std. Error"]
```

```
[1] 0.160706
```

Sampling from sampling distributions

Here we simulate a 1000 samples of size $n = 100$.

```
library(dplyr)
n=100
B=10000
U1 = runif(n*B, 0,1)
U2 = runif(n*B, 0,1)
Z1 = qnorm(U1, 0, 1) %>% matrix(nrow=n, ncol=B)
Z2 = qnorm(U2, 0.5*Z1 + 2*Z1^2, 2) %>% matrix(nrow=n, ncol=B)
beta_2_hat = rep(NA, B)
beta_0_hat <- rep(NA, B)
beta_1_hat <- rep(NA, B)
se_beta0 = rep(NA, B)
se_beta1 = rep(NA, B)
se_beta2 = rep(NA, B)
for(i in 1:B){
  df<-data.frame(Z1=Z1[,i], Z2=Z2[,i])
  fit <- lm(Z2 ~ Z1 + I(Z1^2), data=df)
  beta_0_hat[i] <- coef(fit)["(Intercept)"]
  se_beta0[i] <- summary(fit)$coefficients["(Intercept)", "Std. Error"]
  beta_1_hat[i] <- coef(fit)["Z1"]
  se_beta1[i] <- summary(fit)$coefficients["Z1", "Std. Error"]
  beta_2_hat[i] <- coef(fit)["I(Z1^2)"]
  se_beta2[i] <- summary(fit)$coefficients["I(Z1^2)", "Std. Error"]
}
mean(beta_0_hat - 0)
```

```
[1] -0.002375124
```

```
mean(se_beta0-sd(beta_0_hat))
```

```
[1] -0.003851249
```

```
mean(beta_1_hat - 0.5)
```

```
[1] 0.001183913
```

```r
mean(se_beta1-sd(beta_1_hat))
```

```
[1] -0.004070391
```

```r
mean(beta_2_hat - 2)
```

```
[1] 0.0005370023
```

```r
mean(se_beta2-sd(beta_2_hat))
```

```
[1] -0.003863475
```

Sampling from sampling distributions

Sample size (n)

5 30 300

5 35 65 95 125 155 185 215 245 275 300

Repetitions (R)

10 500 5,000

10 510 1,010 1,510 2,010 2,510 3,010 3,510 4,010 4,510 5,000

True mean (μ)

-3 0 3

-3 -2.4 -1.8 -1.2 -0.6 0 0.6 1.2 1.8 2.4 3

Variance (σ^2)

0.1 1 5

0.1 0.6 1.1 1.6 2.1 2.6 3.1 3.6 4.1 4.6 5

- [x] Show 95% t-intervals

Resimulate

Sampling from sampling distributions

Population distribution

Exponential(rate=1)

Sample size (n)

2 30 500

2 52 102 152 202 252 302 352 402 452 500

Repetitions (R)

100 2,000 20,000

100 2,100 4,100 6,100 8,100 10,100 12,100 14,100 16,100 18,100 20,000

- [x] Use theoretical μ and σ when available
- [x] Show QQ plot vs N(0,1)

Resimulate

We simulate R studies of size n, compute Z = sqrt(n)(X̄-μ)/σ, and compare to N(0,1).

Speaker notes