# OLS Asymptotics

AECN 396/896-002

# Before we start

## Learning objectives

Understand the consequences of the violation of the homoskedasticity assumption and how to deal with the problem

## Table of contents

# OLS Asymptotics

**Large Sample Properties of OLS**

- Properties of OLS that hold only when the sample size is infinite very large
- (loosely put) How OLS estimators behave when the number of observations goes infinite (really large)

**Small Sample Properties of OLS**

Under certain conditions:

- OLS estimators are unbiased
- OLS estimators are efficient

These hold whatever the sample size is.

# Consistency

## Consistency

**Verbally (and very loosely)**

An estimator is consistent if the probability that the estimator produces the true parameter is 1 when sample size is infinite.

## MC simulation: consistency of OLS estimators

OLS estimator of the coefficient on $x$ in the following model with all $MLR.1$ through $MLR.4$ satisfied:

$$y_i = \beta_0 + \beta_1 x_i + u_i$$

with all the conditions necessary for the unbiasedness property of OLS satisfied.

## Conceptual steps of the MC simulations

- Generate data according to $y_i = \beta_0 + \beta_1 x_i + u_i$
- Estimate the coefficients and store them
- Repeat the above experiment 1000 times
- Examine how the coefficient estimates are distributed

## Conceptual steps of the MC simulations

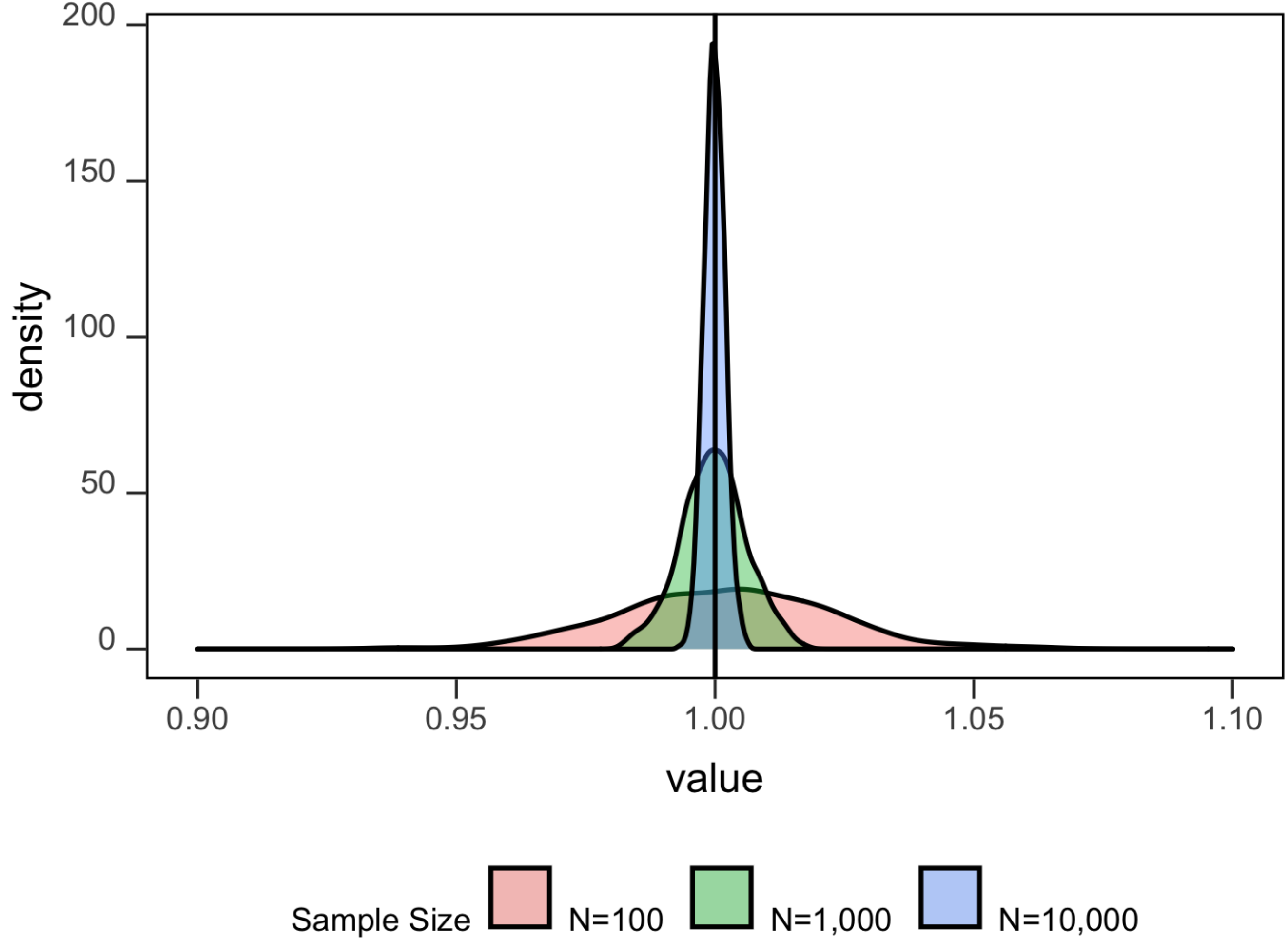
- Generate data according to $y_i = \beta_0 + \beta_1 x_i + u_i$
- Estimate the coefficients and store them
- Repeat the above experiment 1000 times
- Examine how the coefficient estimates are distributed

**What you should see is**

As $N$ gets larger (more observations), the distribution of $\hat{\beta}_1$ get more tightly centered around its true value (here, $1$)

```r
#--- Preparation ---#
B <- 1000 # the number of iterations
N_list <- c(100, 1000, 10000) # sample size
N_len <- length(N_list)
estimate_storage <- matrix(0, B, 3) # estimates storage

for (j in 1:N_len) {
  temp_N <- N_list[j]
  for (i in 1:B) {
    #--- generate data ---#
    x <- rnorm(temp_N) # indep var 1
    u <- rnorm(temp_N) * 0.2 # error
    y <- 1 + x + u # dependent variable 1
    data <- data.frame(y = y, x = x)

    #--- OLS ---#
    reg <- lm(y ~ x, data = data) # OLS

    #--- store coef estimates ---#
    estimate_storage[i, j] <- reg$coef[2]
  }
}
```

```r
plot_data <- melt(data.frame(estimate_storage))

#--- create a figure ---#
g_co_ex <- ggplot(data = plot_data) +
  geom_density(aes(x = value, fill = variable), alpha = 0.4) +
  geom_vline(xintercept = 1) +
  xlim(0.9, 1.1) +
  scale_fill_discrete(
    name = "Sample Size",
    labels = c("N=100 ", "N=1,000 ", "N=10,000")
  ) +
  theme(
    legend.position = "bottom"
  )
```


200
150
100
50
0
density
0.90
0.95
1.00
1.05
1.10
value
Sample Size
N=100
N=1,000
N=10,000

**Consistency of OLS estimators**

Under $MLR.1$ through $MLR.4$, OLS estimators are consistent

## MC simulations: Inconsistency of OLS estimators

**Conceptual steps of MC simulations**

- generate data ($N$ observations) according to $y_i = \beta_0 + \beta_1 x_i + u_i$ with $E[u_i|x_i] \neq 0$
- Estimate the coefficients and store them
- repeat the above experiment 1000 times
- examine how the coefficient estimates are distributed

## MC simulations: Inconsistency of OLS estimators

**Conceptual steps of MC simulations**

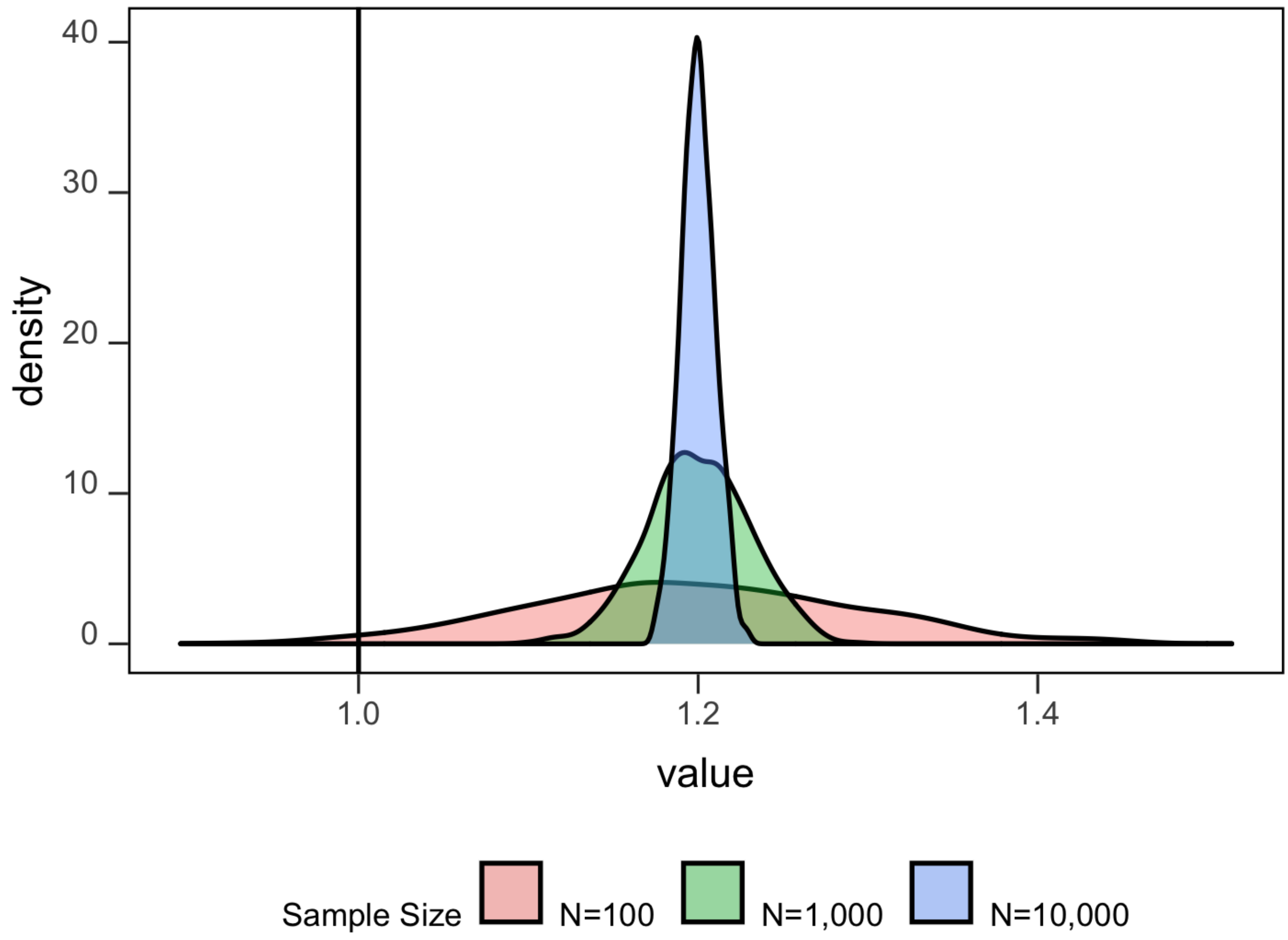
- generate data ($N$ observations) according to $y_i = \beta_0 + \beta_1 x_i + u_i$ with $E[u_i|x_i] \neq 0$
- Estimate the coefficients and store them
- repeat the above experiment 1000 times
- examine how the coefficient estimates are distributed

**Question**

Would the bias disappear as N gets larger?

```r
#--- Preparation ---#
N_list <- c(100, 1000, 10000) # sample size
N_len <- length(N_list)
estimate_storage <- matrix(0, B, 3) # estimates storage

for (j in 1:N_len) {
  temp_N <- N_list[j]
  for (i in 1:B) {
    #--- generate data ---#
    mu <- rnorm(temp_N) # shared term between x and u
    x <- rnorm(temp_N) + 0.5 * mu # <<
    u <- rnorm(temp_N) + 0.5 * mu # <<
    y <- 1 + x + u # dependent variable
    data <- data.frame(y = y, x = x)

    #--- OLS ---#
    reg <- lm(y ~ x, data = data) # OLS

    #--- store coef estimates ---#
    estimate_storage[i, j] <- reg$coef[2]
  }
}
```

```r
#--- wide to long format ---#
plot_data <- melt(data.frame(estimate_storage))

#--- create a figure ---#
g_inco_ex <- ggplot(data = plot_data) +
  geom_density(aes(x = value, fill = variable), alpha = 0.4) +
  geom_vline(xintercept = 1) +
  scale_fill_discrete(
    name = "Sample Size",
    labels = c("N=100 ", "N=1,000 ", "N=10,000")
  ) +
  theme(
    legend.position = "bottom"
  )
```


density
value
0
10
20
30
40
1.0
1.2
1.4
Sample Size
N=100
N=1,000
N=10,000

# Asymptotic Normality

## Testing

When we talked about hypothesis testing, we made the following assumption:

**Normality assumption**

The population error $u$ is independent of the explanatory variables $x_1, \ldots, x_k$ and is normally distributed with zero mean and variance $\sigma^2$:

$u \sim Normal(0, \sigma^2)$

## Testing

When we talked about hypothesis testing, we made the following assumption:

**Normality assumption**

The population error $u$ is independent of the explanatory variables $x_1, \ldots, x_k$ and is normally distributed with zero mean and variance $\sigma^2$:

$$u \sim Normal(0, \sigma^2)$$

**Remember**

- If the normality assumption is violated, t-statistic and F-statistic we constructed before are no longer distributed as t-distribution and F-distribution, respectively
- So, whenever $MLR.6$ is violated, our t- and F-tests are invalid

**Fortunately**

You can continue to use t- and F-tests because (slightly transformed) OLS estimators are approximately normally distributed when the sample size is large enough.

## Central Limit Theorem (CLT)

Suppose $\{x_1, x_2, ...\}$ is a sequence of ideticaly independently distributed random variables with $E[x_i] = \mu$ and $Var[x_i] = \sigma^2 < \infty$. Then, as $n$ approaches infinity,

$$\sqrt{n}(\frac{1}{n}\sum_{i=1}^{n} x_i - \mu) \xrightarrow{d} N(0, \sigma^2)$$

## Central Limit Theorem (CLT)

Suppose $\{x_1, x_2, ...\}$ is a sequence of idetically independently distributed random variables with $E[x_i] = \mu$ and $Var[x_i] = \sigma^2 < \infty$. Then, as $n$ approaches infinity,

$$\sqrt{n}(\frac{1}{n}\sum_{i=1}^{n} x_i - \mu) \xrightarrow{d} N(0, \sigma^2)$$

**Verbally**

Sample mean less its expected value multiplied by $\sqrt{n}$ (square root of the sample size) is going to be distributed as Normal distribution as $n$ goes infinity where its expected value is 0 and variance is the variance of $x$.

**Example**

$x_i \sim Bern[p = 0.3]$

1 with probability $p$ and 0 with probability $1 - p$.

- $E[x_i] = p = 0.3$
- $Var[x_i](\sigma^2) = p(1-p) = 0.21$

**Example**

$x_i \sim Bern[p = 0.3]$

1 with probability $p$ and 0 with probability $1 - p$.

- $E[x_i] = p = 0.3$
- $Var[x_i](\sigma^2) = p(1 - p) = 0.21$

**According to CLT**

$\left(\sqrt{n}(\frac{1}{n}\sum_{i=1}^{n} x_i - \mu) \xrightarrow{d} N(0, \sigma^2)\right)$

$\sqrt{n}(\frac{1}{n}\sum_{i=1}^{n} x_i - 0.3) \xrightarrow{d} N(0, 0.21)$

## MC simulations: CLT

**Conceptual steps of the MC simulation**

- draw $n$ observations from $x_i \sim Bern(0.3)$
- find its mean, subtract the expected value (here, $E[x_i] = 0.3$), multiply by $\sqrt{n}$ ($\sqrt{n}(\frac{1}{n}\sum_{i=1}^{n} x_i - \mu)$
- store the calculated value
- repeat the above experiment 1000 times
- examine how the calculated values are distributed

## MC simulations: CLT

**Conceptual steps of the MC simulation**

- draw $n$ observations from $x_i \sim Bern(0.3)$
- find its mean, subtract the expected value (here, $E[x_i] = 0.3$), multiply by $\sqrt{n}$ ($\sqrt{n}(\frac{1}{n}\sum_{i=1}^{n} x_i - \mu)$
- store the calculated value
- repeat the above experiment 1000 times
- examine how the calculated values are distributed

**What you should see is**

As $N$ gets larger (more observations), the distribution of $\sqrt{n}(\frac{1}{n}\sum_{i=1}^{n} x_i - 0.3)$ looks more and more like $N(0, 0.21)$

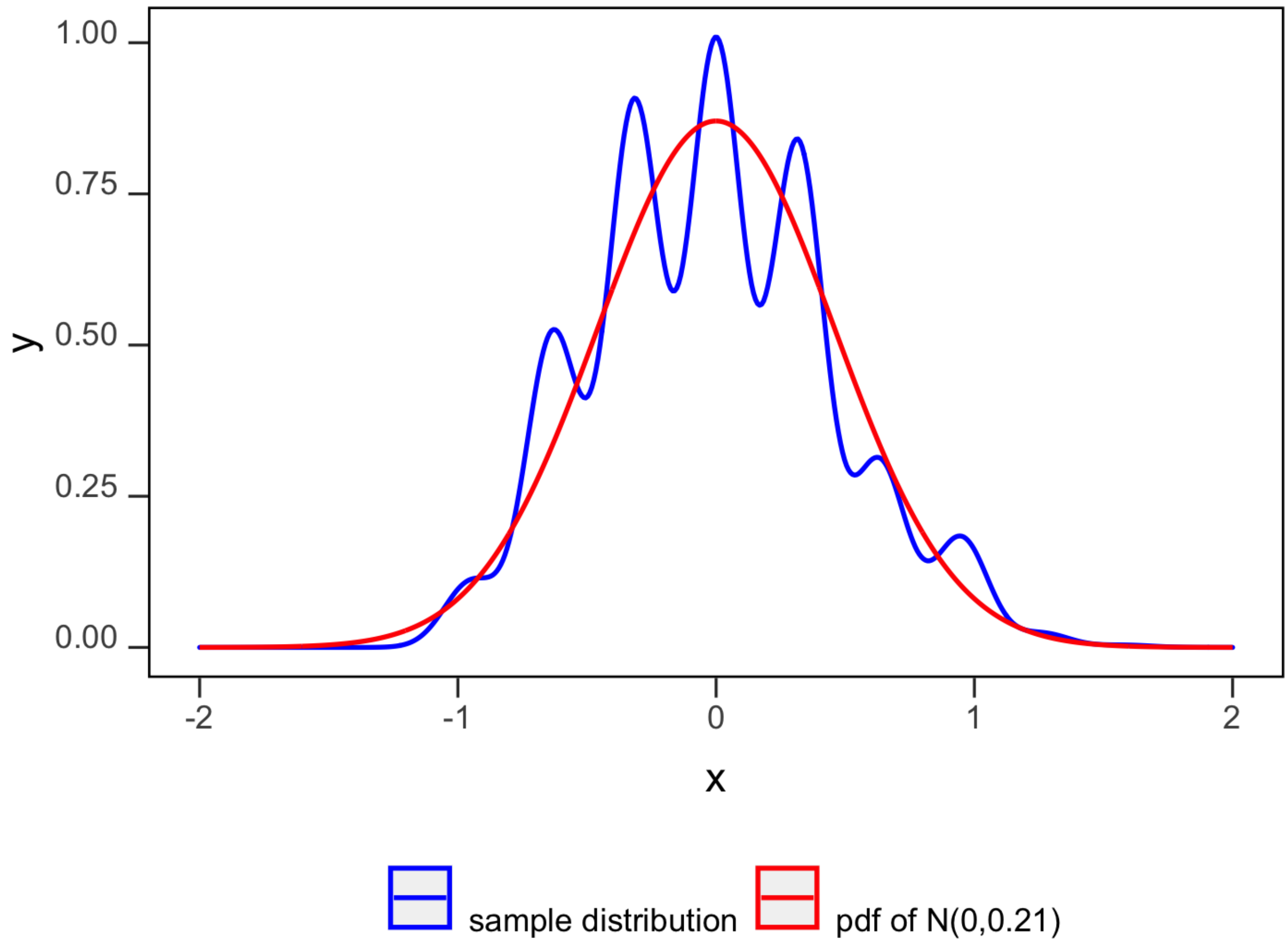
## MC simulations (N = 10)

```
set.seed(893269)
#--- the number of observations ---#
# this is what we change
N <- 10 # number of observations
B <- 1000 # number of iterations
p <- 0.3 # mean of the Bernoulli distribution
storage <- rep(0, B)

for (i in 1:B) {
  #--- draw from Bern[0.3] (x distributed as Bern[0.3]) ---#
  x_seq <- runif(N) <= p

  #--- sample mean ---#
  x_mean <- mean(x_seq)

  #--- normalize ---#
  lhs <- sqrt(N) * (x_mean - p)

  #--- save lhs to storage ---#
  storage[i] <- lhs
}
```

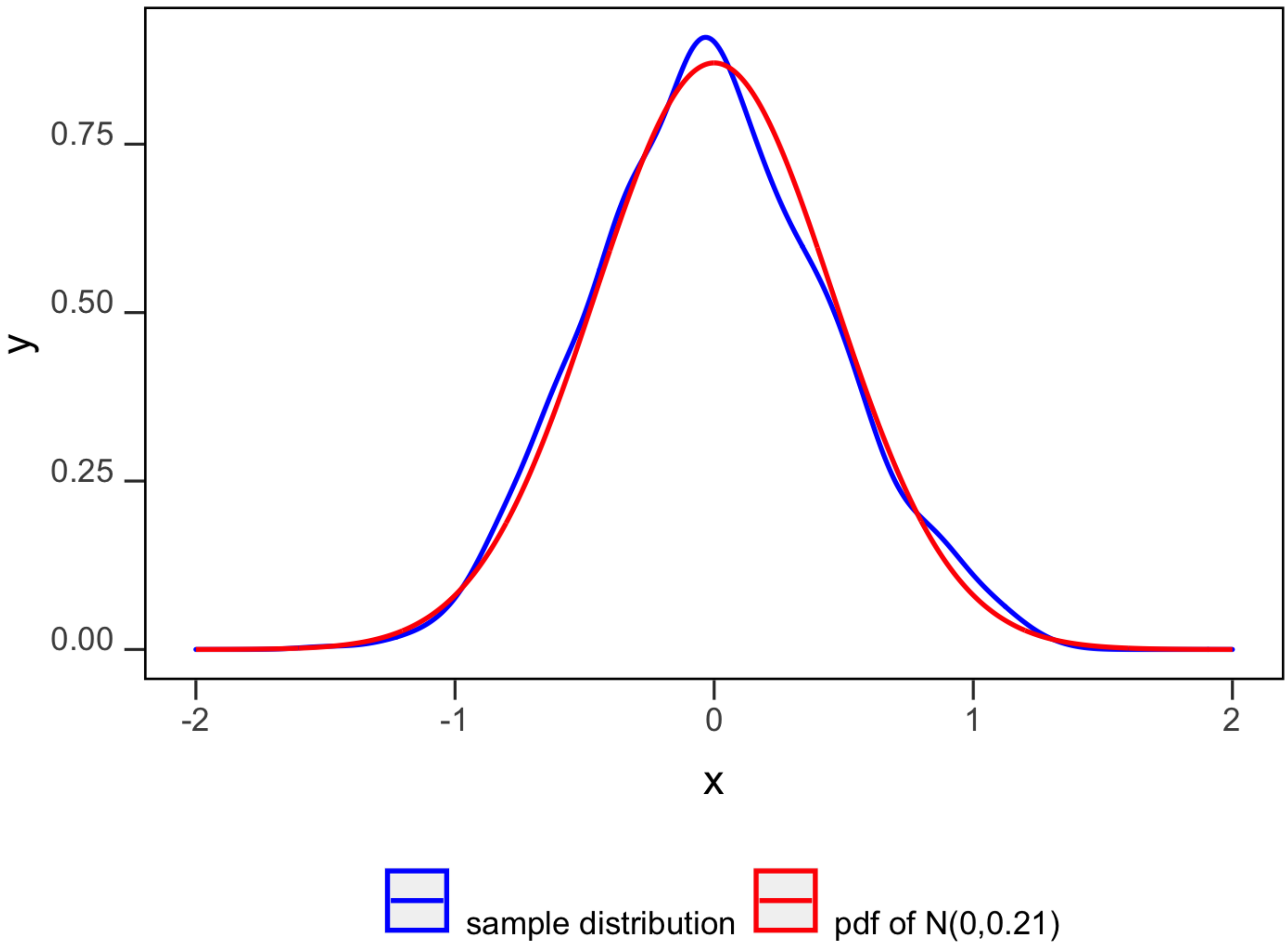
## Visualization

```
data_pdf <- data.frame(
  x = seq(-2, 2, length = 1000),
  y = dnorm(seq(-2, 2, length = 1000), sd = sqrt(p * (1 - p)))
)

g_N_10 <-
  ggplot() +
  geom_density(
    data = data.frame(x = storage),
    aes(x = x, color = "sample distribution")
  ) +
  geom_line(
    data = data_pdf,
    aes(y = y, x = x, color = "pdf of N(0,0.21)")
  ) +
  scale_color_manual(
    values = c("sample distribution" = "blue", "pdf of N(0,0.21)" = "red"),
    name = ""
  ) +
  theme(
    legend.position = "bottom"
  )
```

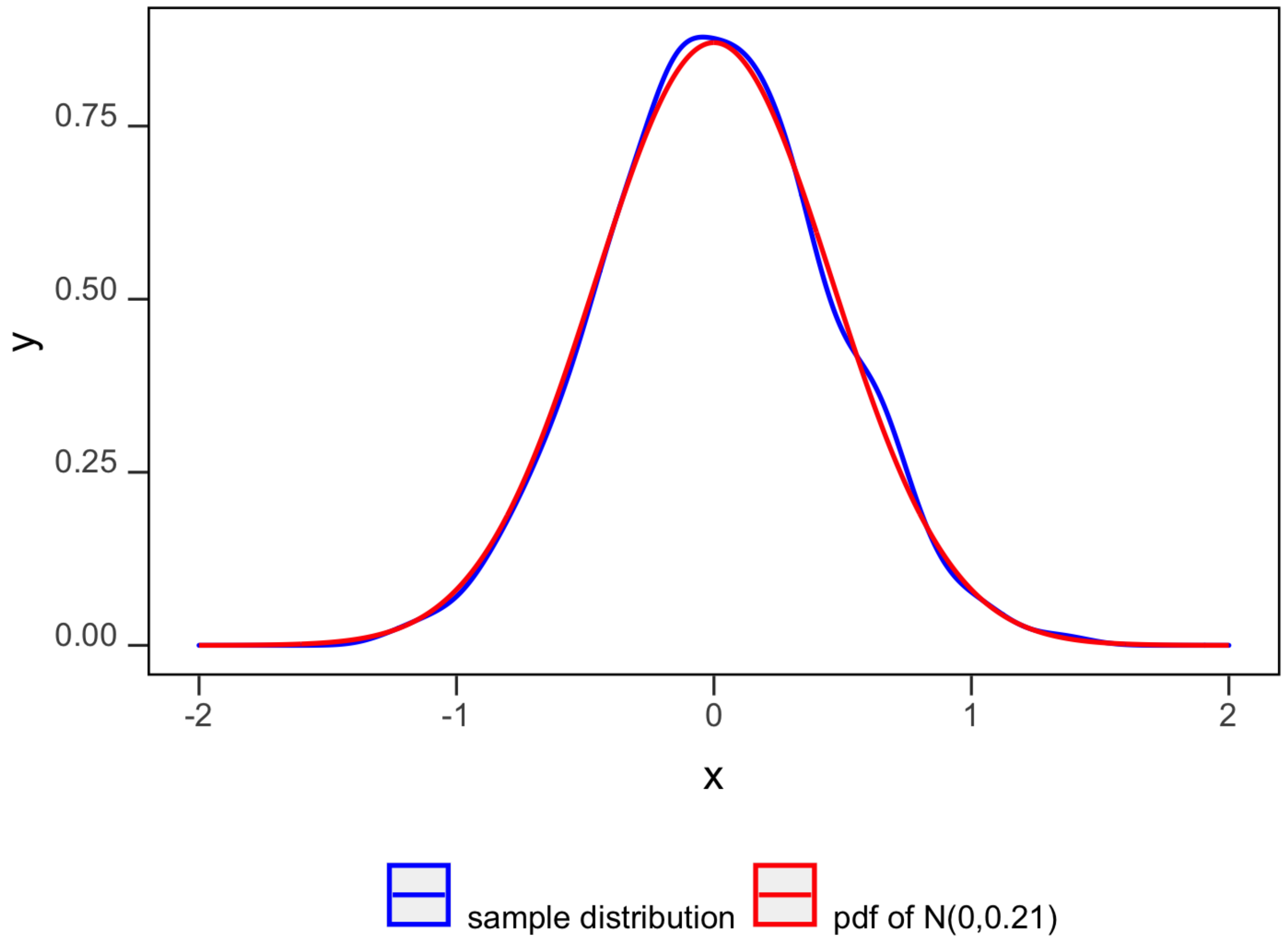
## MC simulations (N = 10)



MC simulations (N = 100)



## MC simulations (N = 10000)



**Important**

CLT holds for any distribution of $x_i$ as long as it has a finite expected value and variance.

Under assumptions $MLR.1$ through $MLR.5$ (MLR.6 not necessary!!),

**Asymptotic Normality of OLS**

$\sqrt{n}(\hat{\beta}_j - \beta_j) \xrightarrow{a} N(0, \sigma^2/\alpha_j^2)$

where $\alpha_j^2 = plim(\frac{1}{n}\sum_{i=1}^{n} r_{i,j}^2)$, where $r_{i,j}^2$ are the residuals from regressing $x_j$ on the other independent variables.

**Consistency**

$\hat{\sigma}^2 \equiv \frac{1}{n-k-1}\sum_{i=1}^{n} \hat{u}_i^2$ is a consistent estimator of of $\sigma^2$ $(Var(u))$

**Further**

- $(\hat{\beta}_j - \beta_j)/se(\hat{\beta}_j) \xrightarrow{a} N(0,1)$
- $(\hat{\beta}_j - \beta_j)/\widehat{se(\hat{\beta}_j)} \xrightarrow{a} N(0,1)$, where $\widehat{se(\hat{\beta}_j)} = \sqrt{\frac{\hat{\sigma}^2}{SST_j(1-R_j^2)}}$

**Small sample (any sample size)**

Under $MLR.1$ through $MLR.5$ and $MLR.6$ $(u_i \sim N(0, \sigma^2))$,

- $(\hat{\beta}_j - \beta_j)/sd(\hat{\beta}_j) \sim N(0,1)$
- $(\hat{\beta}_j - \beta_j)/se(\hat{\beta}_j) \sim t_{n-k-1}$

**Large sample (when（n）goes infinity)**

Under $MLR.1$ through $MLR.5$ without $MLR.6$,

- $(\hat{\beta}_j - \beta_j)/sd(\hat{\beta}_j) \xrightarrow{a} N(0,1)$
- $(\hat{\beta}_j - \beta_j)/se(\hat{\beta}_j) \xrightarrow{a} N(0,1)$

## Testing under large sample

**It turns out,**

You can proceed exactly the same way as you did before (practically speaking)!!

- calculate $(\hat{\beta}_j - \beta_j)/\widehat{se(\hat{\beta}_j)}$
- check if the obtained value is greater than (in magnitude) the critical value for the specified significance level under $t_{n-k-1}$

## Testing under large sample

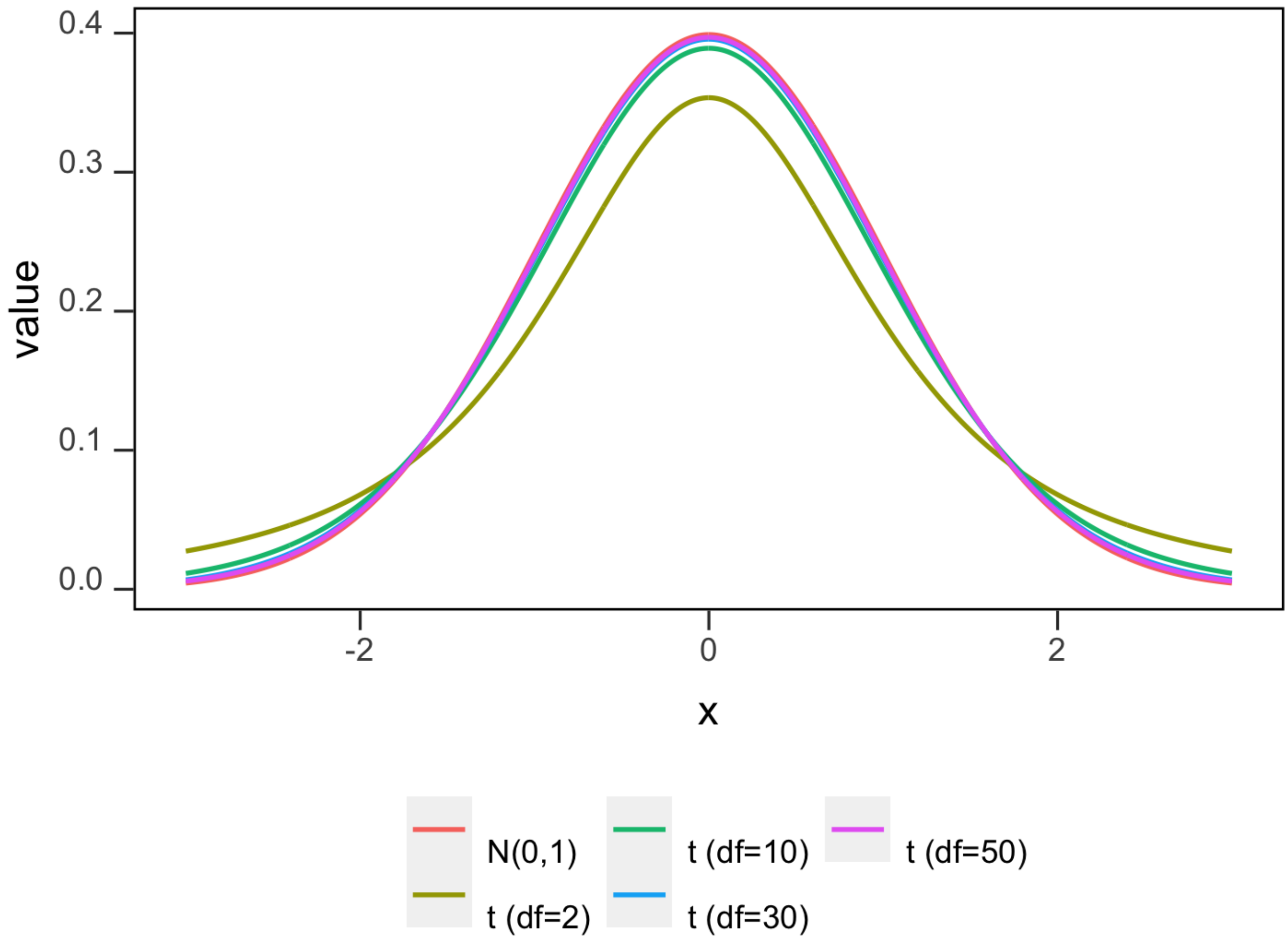
**It turns out,**

You can proceed exactly the same way as you did before (practically speaking)!!

- calculate $(\hat{\beta}_j - \beta_j)/\widehat{se(\hat{\beta}_j)}$
- check if the obtained value is greater than (in magnitude) the critical value for the specified significance level under $t_{n-k-1}$

**But,**

Shouldn't we use $N(0,1)$ when you find the critical value?



0.4
0.3
0.2
0.1
0.0
value
-2
0
2
x
N(0,1)
t (df=2)
t (df=10)
t (df=30)
t (df=50)

**Testing under large sample**

Since $t_{n-k-1}$ and $N(0,1)$ are almost identical when $n$ is large, there is very little error in using $t_{n-k-1}$ instead of $N(0,1)$ to find the critical value.

## When the homoskedasticity is violated (as almost always the case)

**Important**

The consistency of the estimation of $\widehat{Var(\hat{\beta})}$ DOES require the homoskedasticity assumption (MLR.5)!!

- the usual t-statistics and confidence intervals are invalid no matter how large the sample size is if error is heteroskedastic
- so, we should use heteroskedasticity-robust or cluster-robust standard error estimators even when the sample size is large