

# STAT 8010 R Session 3: Normal Distribution, Sampling Distributions, Inference for One Population Mean

Whitney Huang

6/2/2023

## Contents

<b>Session Objective</b>	<b>2</b>
Normal distribution . . . . .	2
Normal density curve . . . . .	2
Standard Normal: $Z \sim N(0, 1)$ . . . . .	3
Example: STAT 8020 Exam Score . . . . .	4
Standard normal percentiles . . . . .	4
General normal percentiles . . . . .	4
Normal approximation to Binomial Distribution . . . . .	4
Sampling Distribution . . . . .	5
Sample Mean . . . . .	5
Sample Median . . . . .	6
Sample Variance . . . . .	7
CLT . . . . .	8
Animation . . . . .	8
Sample Size and the Normal Approximation . . . . .	8
Chi-Squared Distribution . . . . .	10
Student's t-Distribution . . . . .	12
F-Distribution . . . . .	12
Inference for One Population Mean . . . . .	13
Understanding Confidence Intervals . . . . .	13
Average Height Example . . . . .	14
Sample Size Calculation . . . . .	15
Average Height Example Revisited . . . . .	15
Hypothesis Testing: Cereal Weight Example . . . . .	15
Blood Test Example . . . . .	16
Type I and Type II Errors . . . . .	17
Power analysis . . . . .	18

## Session Objective

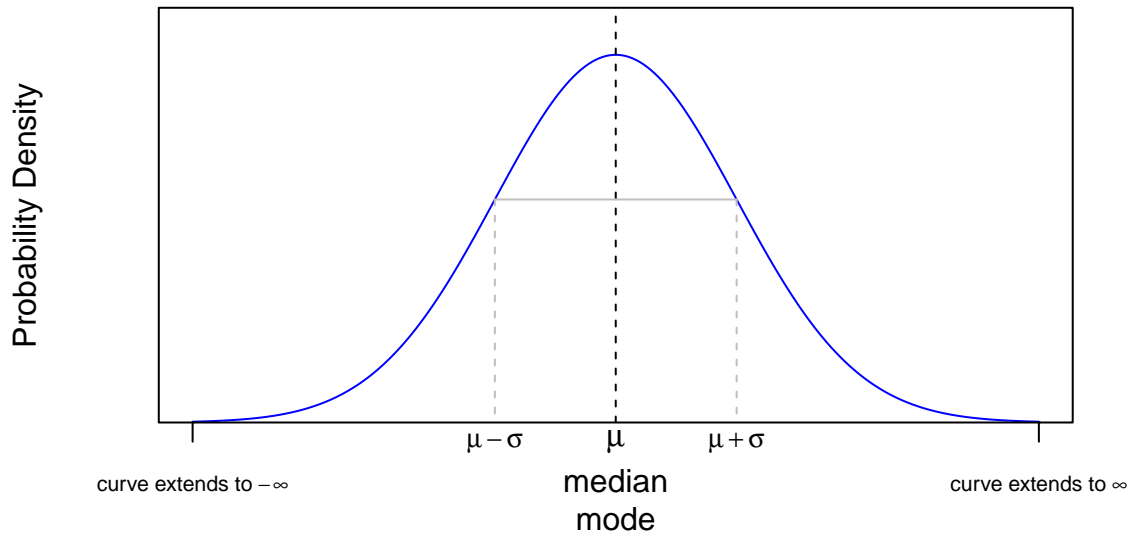
- To gain experience with R, a programming language and free software environment for statistical computing and graphics.
- To be able to compute probabilities/quantiles with any normal distribution
- To demonstrate the concept of sampling distribution and CLT
- To be able to conduct inference on mean for one numerical variable

## Normal distribution

### Normal density curve

```
xg <- seq(-3.5, 3.5, 0.01)
yg <- dnorm(xg)

par(las = 1)
plot(xg, yg, type = "l", xlab = "",
     ylab = "Probability Density",
     col = "blue", xaxt = "n",
     yaxt = "n", yaxs = "i",
     ylim = c(0, 0.45))
abline(v = 0, lty = 2)
axis(1, at = 0, label = expression(mu),
     tick = F, line = -1)
axis(1, at = 0, label = "median", line = 0,
     tick = F)
axis(1, at = 0, label = "mode", line = 1,
     tick = F)
axis(1, at = -3.5, labels = expression(paste("curve extends to ", -infinity)),
     cex.axis = 0.7)
axis(1, at = 3.5, labels = expression(paste("curve extends to ", infinity)),
     cex.axis = 0.7)
segments(0, dnorm(1), -1, col = "gray")
segments(0, dnorm(1), 1, col = "gray")
segments(-1, dnorm(1), -1, 0, lty = 2,
     col = "gray")
segments(1, dnorm(1), 1, 0, lty = 2,
     col = "gray")
axis(1, at = -1, label = expression(mu - sigma), tick = F, cex.axis = 0.8, line = -1)
axis(1, at = 1, label = expression(mu + sigma), tick = F, cex.axis = 0.8, line = -1)
```



**Standard Normal:**  $Z \sim N(0, 1)$

We use  $\Phi(\cdot)$  to denote the cdf of the standard normal distribution

1.  $\Phi(0) = .50 \Rightarrow$  Mean and Median ( $50_{th}$  percentile) for standard normal are both 0

```
pnorm(0)
```

```
## [1] 0.5
```

2.  $\Phi(-z) = 1 - \Phi(z)$

```
(z <- rnorm(1))
```

```
## [1] -0.723239
```

```
pnorm(-z)
```

```
## [1] 0.7652335
```

```
1 - pnorm(z)
```

```
## [1] 0.7652335
```

3.  $\mathbb{P}(Z > z) = 1 - \Phi(z) = \Phi(-z)$

```
pnorm(1, lower.tail = F)
```

```
## [1] 0.1586553
```

```
pnorm(-1, lower.tail = T)
```

```
## [1] 0.1586553
```

### Example: STAT 8020 Exam Score

```
mu = 78; sigma2 = 36
```

- What is the probability that a randomly chosen test taker got a score greater than 84?

```
1 - pnorm(84, mu, sqrt(sigma2))
```

```
## [1] 0.1586553
```

- Suppose the passing score for this exam is 75. What is the probability that a randomly chosen test taker got a score greater than 84 given that she/he pass the exam?

```
(1 - pnorm(84, mu, sqrt(sigma2))) / (1 - pnorm(75, mu, sqrt(sigma2)))
```

```
## [1] 0.2294488
```

- Find the 84<sub>th</sub> percentile

```
qnorm(0.84, mu, sqrt(sigma2))
```

```
## [1] 83.96675
```

### Standard normal percentiles

```
qnorm(c(.1, .55, .9))
```

```
## [1] -1.2815516 0.1256613 1.2815516
```

### General normal percentiles

```
qnorm(.8, 20, 7)
```

```
## [1] 25.89135
```

```
20 + 7 * qnorm(.8)
```

```
## [1] 25.89135
```

### Normal approximation to Binomial Distribution

```
n = 400; p = 0.93
sum(dbinom(370:373, n, p))
```

```
## [1] 0.3009909
```

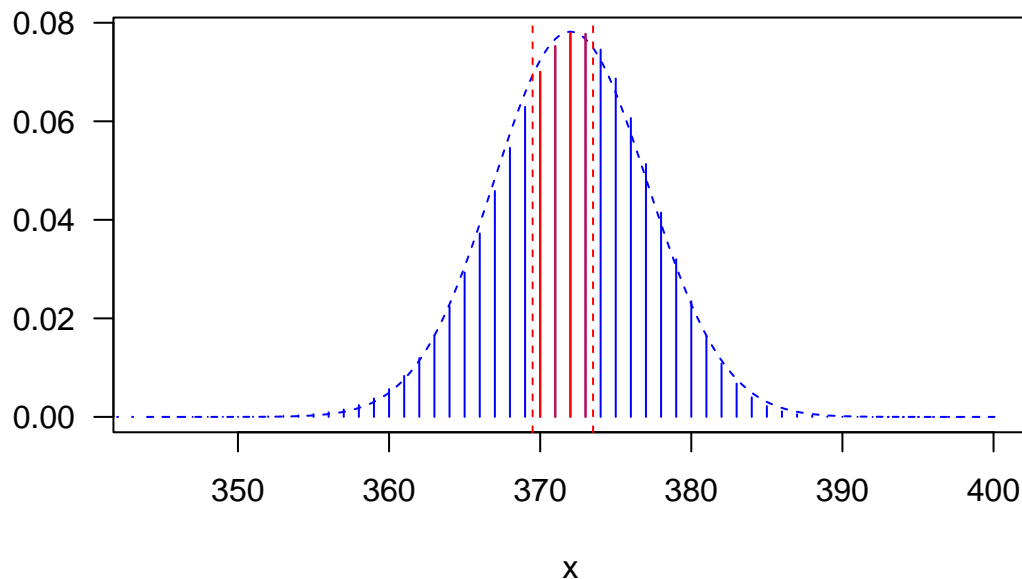
```
pbinom(373, n, p) - pbinom(369, n, p)
```

```
## [1] 0.3009909
```

```
mu = n * p; sigma = sqrt(n * p * (1 - p))
pnorm(373.5, mu, sigma) - pnorm(369.5, mu, sigma)
```

```
## [1] 0.3035037
```

```
plot(0:400, dbinom(0:400, n, p), type = "h", las = 1,
     xlim = c(344, 400), xlab = "x", ylab = "", col = "blue")
xg <- seq(344, 400, 0.1); yg <- dnorm(xg, mu, sigma)
lines(xg, yg, col = "blue", lty = 2)
abline(v = c(369.5, 373.5), lty = 2, col = "red")
lines(370:373, dbinom(370:373, n, p), type = "h", col = "red")
```



## Sampling Distribution

### Sample Mean

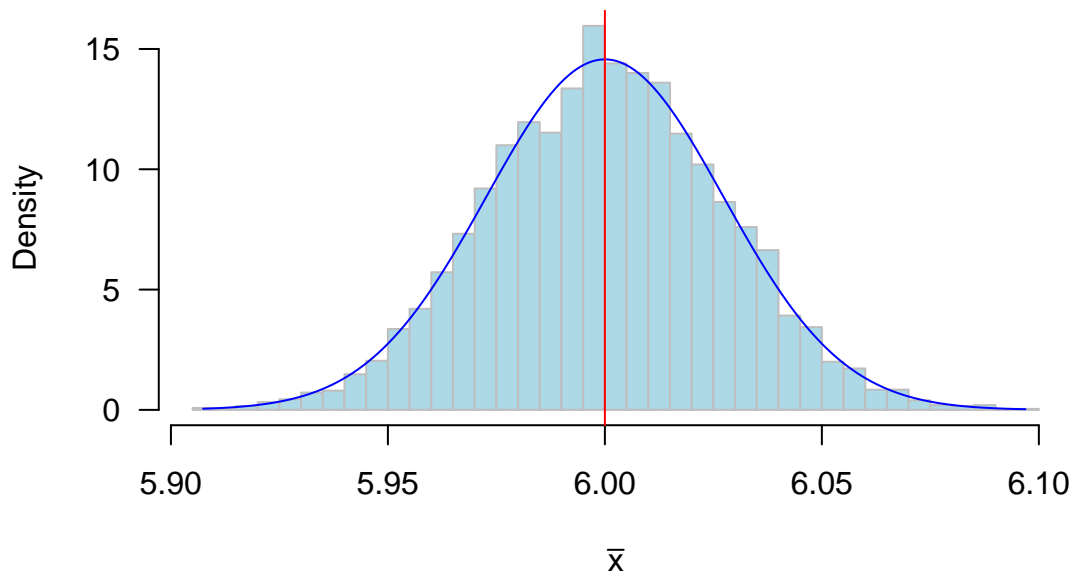
```
M = 5000; n = 3000
set.seed(123)
x <- replicate(M, rnorm(n, 6, 1.5))
xbar <- apply(x, 2, mean)
```

```

hist(xbar, 60, col = "lightblue", border = "gray", prob = T,
     las = 1, xlab = expression(bar(x)))
xg <- seq(min(xbar), max(xbar), len = 1001); yg = dnorm(xg, 6, 1.5 / sqrt(n))
lines(xg, yg, col = "blue")
abline(v = 6, col = "red")

```

### Histogram of $\bar{x}$



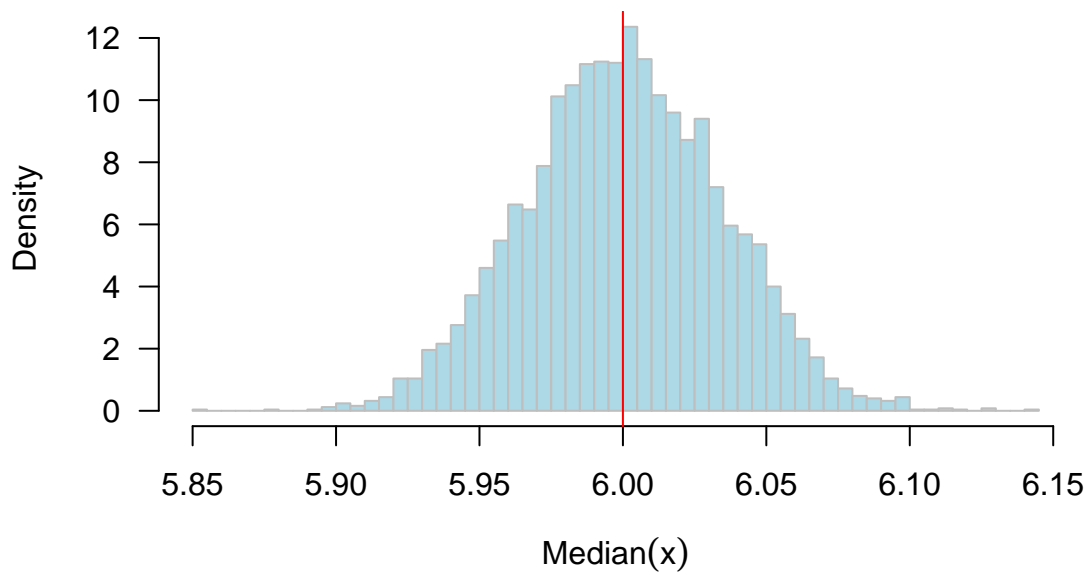
### Sample Median

```

xmed <- apply(x, 2, median)
hist(xmed, 60, col = "lightblue", border = "gray", prob = T,
     las = 1, xlab = expression("Median"(x)))
abline(v = 6, col = "red")

```

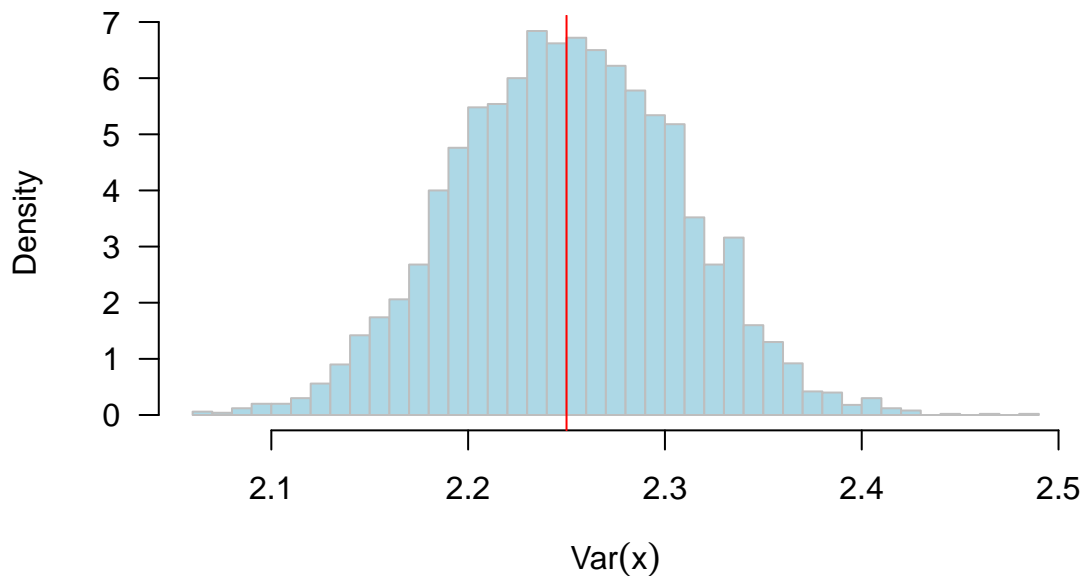
## Histogram of xmed



## Sample Variance

```
xvar <- apply(x, 2, var)
hist(xvar, 40, col = "lightblue", border = "gray", prob = T,
     las = 1, xlab = expression("Var"(x)))
abline(v = 1.5^2, col = "red")
```

## Histogram of xvar



## CLT

### Animation

```
set.seed(123)
sim <- replicate(120, rexp(n = 100, 0.2))
library(animation)
saveLatex({
  for (i in 1:120){
    par(mfrow = c(1, 2), mar = c(5, 4, 1, 1), las = 1)
    brk <- seq(0, max(sim), length.out = 25)
    hist(sim[, i], breaks = brk, prob = T, border = "gray", main = " ",
         xlab = "x", xlim = range(sim), ylim = c(0, 0.25), cex.lab = 1.25,
         cex.axis = 1.25)
    rug(sim[, i])

    xg <- seq(0, range(sim)[2], length.out = 1000)
    lines(xg, dexp(xg, 0.2), lwd = 1.5)
    abline(v = mean(sim[, i]), col = "blue", lwd = 1.5)
    x_bar_1 <- apply(sim, 2, mean)

    library(MASS)
    norm_mle1 <- fitdistr(x_bar_1[1:120], "normal")$estimate
    hist(x_bar_1[1:120], 12, prob = T, border = "white", main = " ",
         xlab = expression(bar(x)[100]), xlim = c(3.2, 6.8), ylim = c(0, 1),
         cex.lab = 1.25, cex.axis = 1.25)
    rug(x_bar_1[1:i], col = "blue", lwd = 1.25)
    xg1 <- seq(3.2, 6.8, length.out = 1000)

  }
  hist(x_bar_1[1:120], 12, prob = T, border = "gray", main = " ",
       xlab = " ", xlim = c(3.2, 6.8), ylim = c(0, 1), add = T)
  lines(xg1, dnorm(xg1, mean = norm_mle1[1], sd = norm_mle1[2]),
       col = "blue", lty = 2, lwd = 1.8)
  lines(xg1, 100 * dgamma(xg1 * 100, 100, 0.2), lwd = 1.5)
  abline(v = mean(x_bar_1[1:120]), col = "blue")
  abline(v = 5, col = "red")

}, img.name = "CLT", ani.opts = "controls,width=0.95\\textwidth",
  latex.filename = ifelse(interactive(), "CTL_demos.tex", ""),
  nmax = 120, ani.dev = "pdf", ani.type = "pdf", ani.width = 10,
  interval = 0.3,
  ani.height = 5,
  documentclass = paste("\\documentclass{article}",
                        "\\usepackage[papersize={10in,5in},margin=0.1in]{geometry}",
                        sep = "\\n"))
```

### Sample Size and the Normal Approximation

```
set.seed(123)
simExp <- replicate(5000, rexp(n = 5000, 0.2))
```



```

xbarEX5 <- apply(simExp[1:5,], 2, mean)
xbarEX30 <- apply(simExp[1:30,], 2, mean)
xbarEX100 <- apply(simExp[1:100,], 2, mean)
xbarEX500 <- apply(simExp[1:500,], 2, mean)
xbarEX5000 <- apply(simExp, 2, mean)

par(mfrow = c(2, 3), las = 1)
hist(xbarEX5, 24, prob = T, main = "", xlab = expression(bar(x)[5]),
     ylab = "Density", border = "gray", col = "lightblue", cex.lab = 1.25)
abline(v = 5, lwd = 1.5)
xgEX5 <- seq(0, 20, length.out = 1000)
lines(xgEX5, dnorm(xgEX5, mean = 5, sd = 5 / sqrt(5)), lty = 2, lwd = 1.3)
hist(xbarEX30, 24, prob = T, main = "Exponential", xlab = expression(bar(x)[30]),
     ylab = "Density", border = "gray", col = "lightblue", cex.lab = 1.25)
abline(v = 5, lwd = 1.5)
xgEX30 <- seq(2, 10, length.out = 1000)
lines(xgEX30, dnorm(xgEX30, mean = 5, sd = 5 / sqrt(30)), lty = 2, lwd = 1.3)
hist(xbarEX500, 24, prob = T, main = "", xlab = expression(bar(x)[500]),
     ylab = "Density", border = "gray", col = "lightblue", cex.lab = 1.25)
abline(v = 5, lwd = 1.5)
xgEX500 <- seq(4, 6, length.out = 1000)
lines(xgEX500, dnorm(xgEX500, mean = 5, sd = 5 / sqrt(500)), lty = 2, lwd = 1.3)

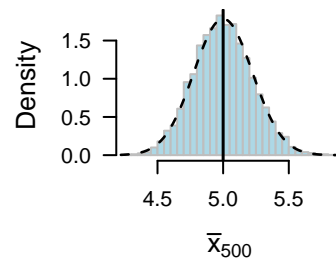
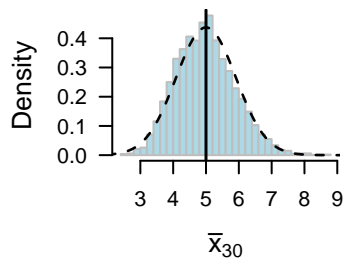
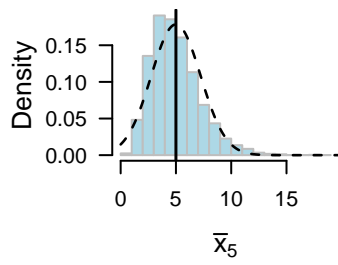
set.seed(123)
simN <- replicate(5000, rnorm(n = 500))

xbarN5 <- apply(simN[1:5,], 2, mean)
xbarN30 <- apply(simN[1:30,], 2, mean)
xbarN500 <- apply(simN[1:500,], 2, mean)

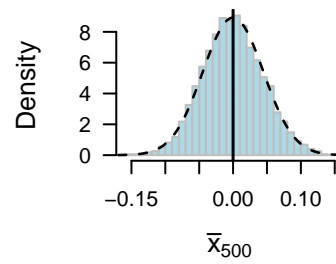
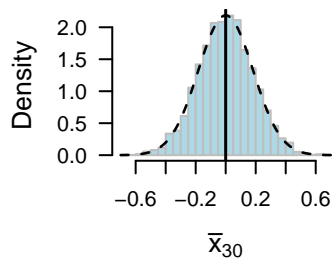
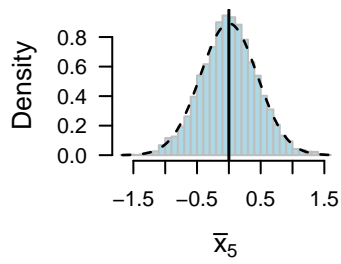
hist(xbarN5, 24, prob = T, main = "", xlab = expression(bar(x)[5]),
     ylab = "Density", border = "gray", col = "lightblue", cex.lab = 1.25)
abline(v = 0, lwd = 1.5)
xgN5 <- seq(-2, 2, length.out = 1000)
lines(xgN5, dnorm(xgN5, mean = 0, sd = 1 / sqrt(5)), lty = 2, lwd = 1.3)
hist(xbarN30, 24, prob = T, main = "Normal", xlab = expression(bar(x)[30]),
     ylab = "Density", border = "gray", col = "lightblue", cex.lab = 1.25)
abline(v = 0, lwd = 1.5)
xgN30 <- seq(-0.7, 0.7, length.out = 1000)
lines(xgN30, dnorm(xgN30, mean = 0, sd = 1 / sqrt(30)), lty = 2, lwd = 1.3)
hist(xbarN500, 24, prob = T, main = "", xlab = expression(bar(x)[500]),
     ylab = "Density", border = "gray", col = "lightblue", cex.lab = 1.25)
abline(v = 0, lwd = 1.5)
xgN500 <- seq(-0.2, 0.2, length.out = 1000)
lines(xgN500, dnorm(xgN500, mean = 0, sd = 1 / sqrt(500)), lty = 2, lwd = 1.3)

```

## Exponential

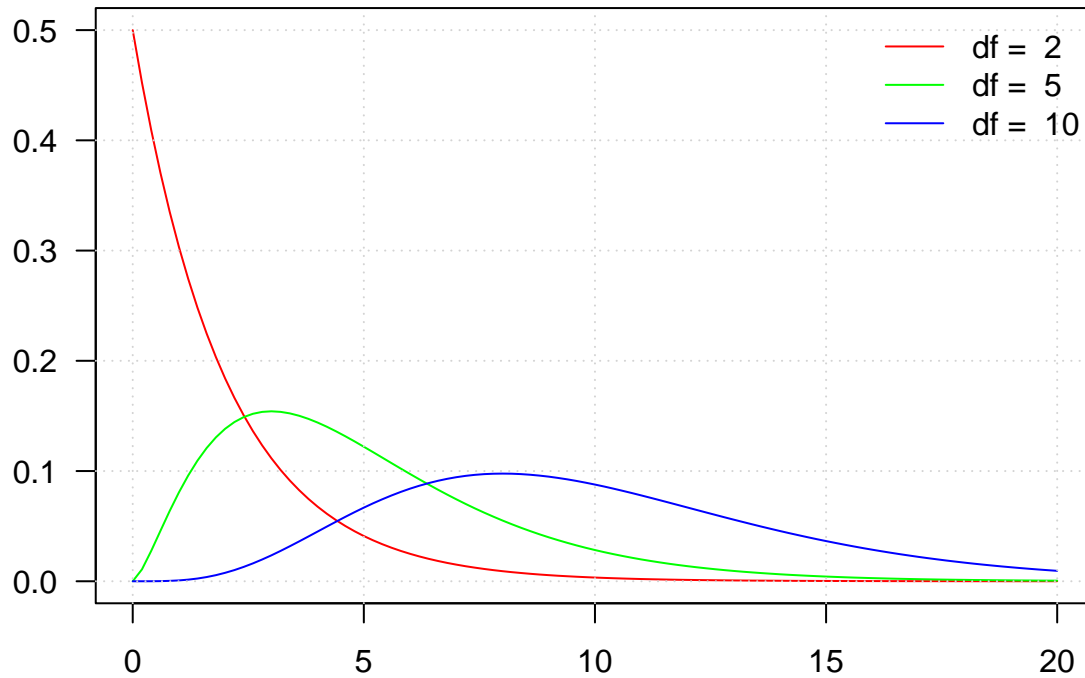


## Normal



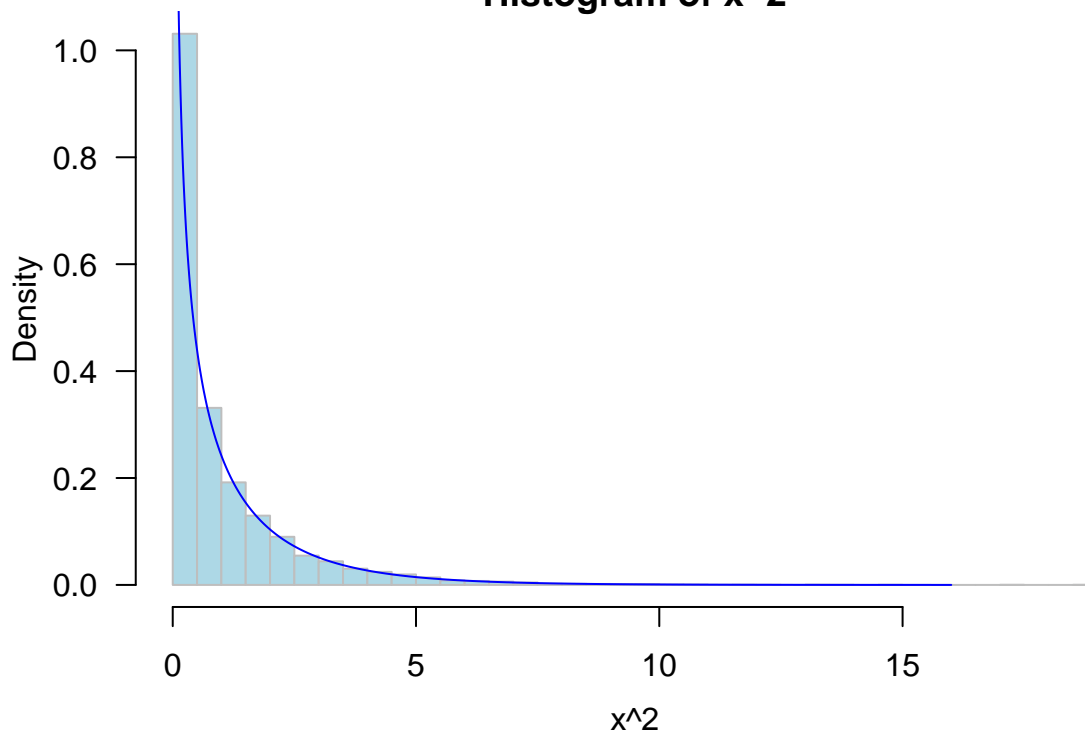
## Chi-Squared Distribution

```
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.4, 1, 0))
curve(dchisq(x, df = 2), from = 0, to = 20, col = "red", xlab = "", ylab = "")
curve(dchisq(x, df = 5), from = 0, to = 20, col = "green", add = T)
curve(dchisq(x, df = 10), from = 0, to = 20, col = "blue", add = T)
grid()
legend("topright", legend = paste("df = ", c(2, 5, 10)), bty = "n", lty = 1,
      col = c("red", "green", "blue"))
```



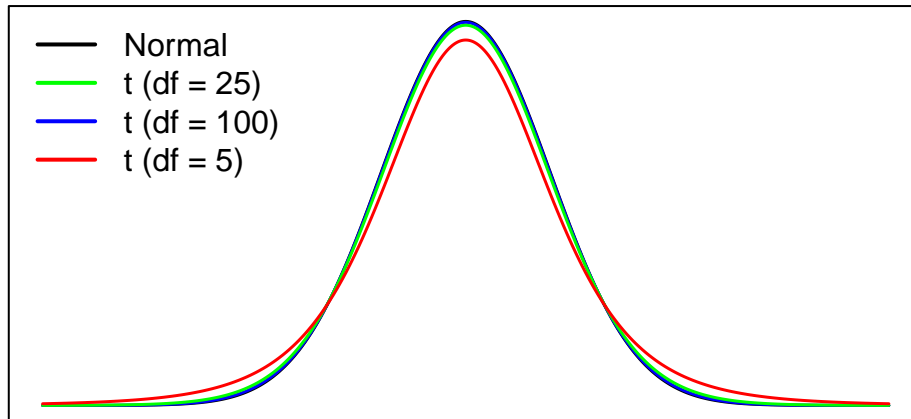
```
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.4, 1, 0))
x <- rnorm(10000)
hist(x^2, 50, prob = T, col = "lightblue", border = "gray")
curve(dchisq(x, df = 1), from = 0, to = 16, n = 1001, col = "blue", add = T)
```

**Histogram of  $x^2$**



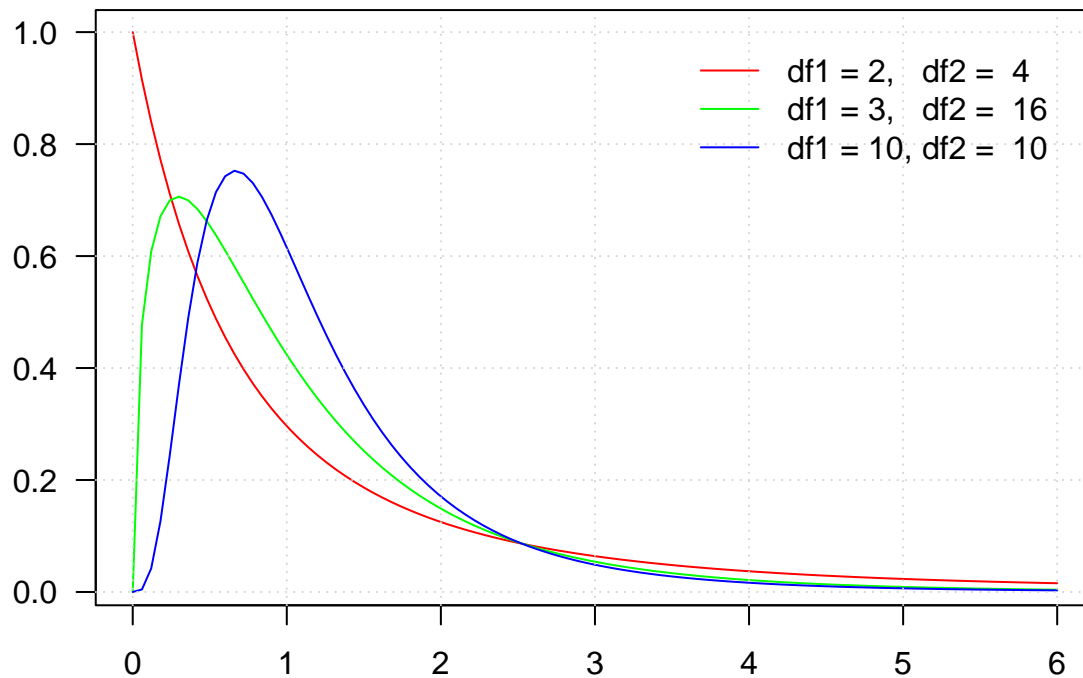
## Student's t-Distribution

```
curve(dnorm(x, sd = 1), from = -5, to = 5,
      xlab = "", ylab = "", lwd = 1.5, n = 1001, xaxt = "n", yaxt = "n")
curve(dt(x, 100), from = -5, to = 5, n = 1001, lwd = 1.5, add = T, col = "blue")
curve(dt(x, 25), from = -5, to = 5, n = 1001, lwd = 1.5, add = T, col = "green")
curve(dt(x, 5), from = -5, to = 5, n = 1001, lwd = 1.5, add = T, col = "red")
legend("topleft", legend = c("Normal", "t (df = 25)", "t (df = 100)", "t (df = 5)"),
      col = c("black", "green", "blue", "red"), lty = 1, lwd = 2, bty = "n")
```



## F-Distribution

```
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.4, 1, 0))
curve(df(x, 2, 4), from = 0, to = 6, col = "red", xlab = "", ylab = "")
curve(df(x, 3, 16), from = 0, to = 6, col = "green", add = T)
curve(df(x, 10, 10), from = 0, to = 6, col = "blue", add = T)
grid()
legend(3.5, 1, legend = paste("df1 = ", c(2, 3, 10), ",", sep = ""), bty = "n", lty = 1,
      col = c("red", "green", "blue"))
legend(4.75, 1, legend = paste("df2 = ", c(4, 16, 10)), bty = "n")
```



## Inference for One Population Mean

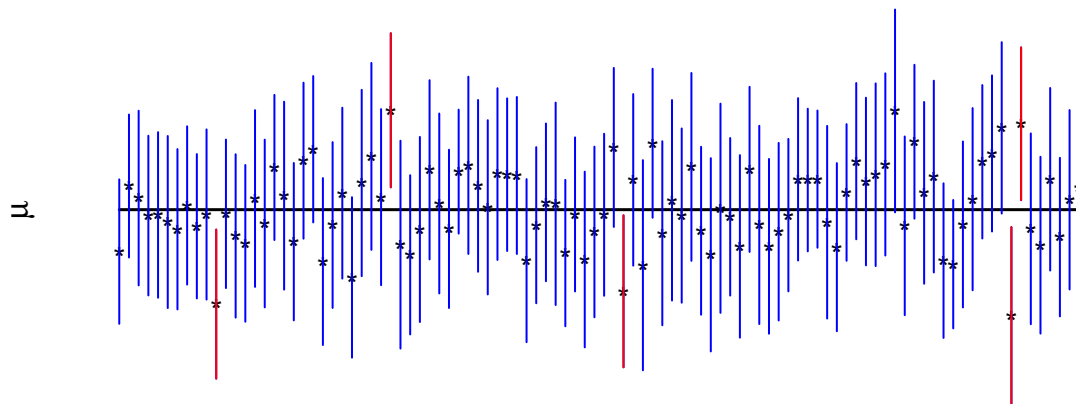
### Understanding Confidence Intervals

```
par(las = 1, mar = c(3, 3, 0.5, 0), mgp = c(1, 1, 0))
set.seed(12)
M = 100; n = 50
x <- replicate(M, rnorm(n))
xbar <- apply(x, 2, mean)
sd <- apply(x, 2, sd)
t <- qt(1 - 0.05 / 2, n - 1)
ME <- t * (sd / sqrt(n))
CI <- array(c(xbar - ME, xbar + ME), dim = c(100, 2))
(miss <- which(apply(CI, 1, prod) > 0))
```

```
## [1] 11 29 53 93 94
```

```
par(las = 1)
plot(1:100, rep(0, 100), type = "l", bty = "n", xaxt = "n", yaxt = "n", xlab = "", lwd = 1.5,
     ylab = expression(mu))
points(1:100, xbar, pch = "*")
for (i in 1:100){
  segments(i, xbar[i] - ME[i], i,
          xbar[i] + ME[i], col = "blue")
}

for (i in miss){
  segments(i, xbar[i] - ME[i], i,
          xbar[i] + ME[i], col = "red")
}
```



### Average Height Example

We measure the heights of 40 randomly chosen men, and get a mean height of 5'9" ( $\sim 175\text{cm}$ ). Suppose we know the standard deviation of men's heights is 4" ( $\sim 10\text{cm}$ ). Find the 95% confidence interval of the true mean height of ALL men.

```
xbar = 5 * 12 + 9; sd = 4
# 95% CI
alpha = 0.05
n = 40
ME <- qnorm(1 - alpha / 2) * (sd / sqrt(n))
CI <- c(xbar - ME, xbar + ME)
CI
```

```
## [1] 67.76041 70.23959
```

```
n = 400
ME <- qnorm(1 - alpha / 2) * (sd / sqrt(n))
CI <- c(xbar - ME, xbar + ME)
CI
```

```
## [1] 68.60801 69.39199
```

```
alpha = 0.01
ME <- qnorm(1 - alpha / 2) * (sd / sqrt(n))
CI <- c(xbar - ME, xbar + ME)
CI
```

```
## [1] 68.48483 69.51517
```

```
sd = 6
ME <- qnorm(1 - alpha / 2) * (sd / sqrt(n))
CI <- c(xbar - ME, xbar + ME)
CI
```

```
## [1] 68.22725 69.77275
```

```

CI <- function(xbar, sd, n, alpha){
  ME <- qnorm(1 - alpha / 2) * (sd / sqrt(n))
  return(c(xbar - ME, xbar + ME))
}

```

```
CI(69, 4, 400, 0.05)
```

```
## [1] 68.60801 69.39199
```

### Sample Size Calculation

Compute the sample size needed in order to estimate the true mean height of All men such that the 95% CI to be 0.5 inches in width

```

sd = 4; alpha = 0.05; ME = 0.5 / 2
n = (qnorm(1 - alpha / 2) * 4 / ME)^2
n

```

```
## [1] 983.4135
```

### Average Height Example Revisited

We measure the heights of 40 randomly chosen men, and get a mean height of 5'9" ( $\sim 175\text{cm}$ ), and a standard deviation of 4.5" ( $\sim 11.4\text{cm}$ ). Find the 95% confidence interval of the true mean height of ALL men.

```

n = 40; alpha = 0.05; sdEst = 4.5; xbar = 69

ME <- qt(1 - alpha / 2, n - 1) * (sdEst / sqrt(n))
CI <- c(xbar - ME, xbar + ME)
CI

```

```
## [1] 67.56083 70.43917
```

### Hypothesis Testing: Cereal Weight Example

New Age Granola Inc claims that average weight of its cereal boxes is 16 oz. The Genuine Grocery Corporation will send back a shipment if the average weight is any less. Suppose Genuine Grocery Corporation takes a random sample of 49 boxes, weight each one, and compute the sample mean  $\bar{X} = 15.9$  oz and sample standard deviation  $s = 0.35$  oz. Perform a hypothesis test at 0.05 significant level to determine if they would reject  $H_0$ , and therefore, this shipment

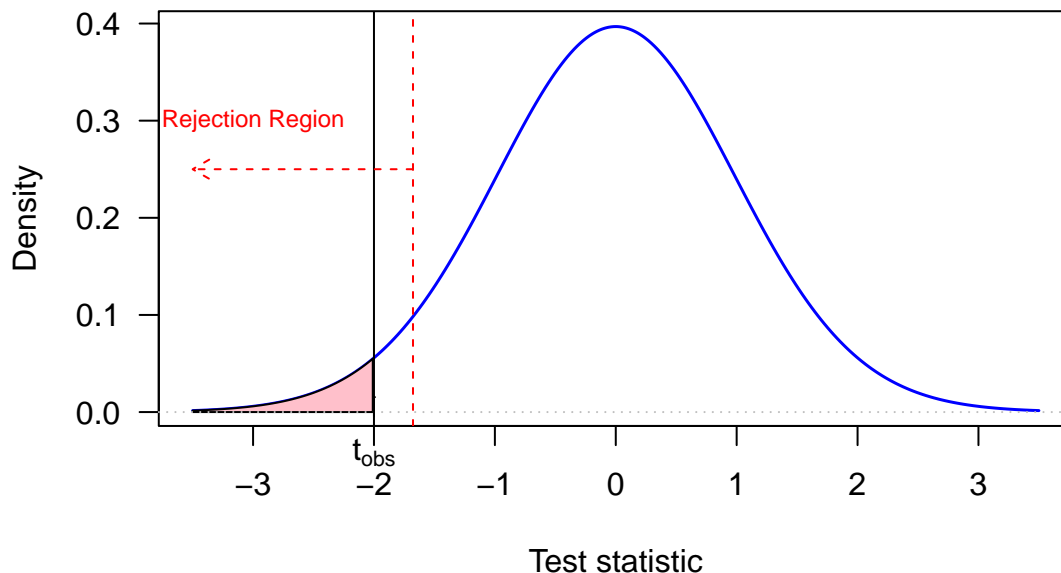
1.  $H_0 : \mu = 16$  vs.  $H_a : \mu < 16$
2. Test Statistic:  $t_{obs} = \frac{15.9-16}{0.35/\sqrt{49}} = -2$
3. Rejection Region Method:  $-t_{0.05,48} = -1.68 \Rightarrow$  Rejection Region is  $(-\infty, -1.68]$ . Since  $t_{obs}$  is in rejection region, we reject  $H_0$
4. P-Value Method:  $\mathbb{P}(t^* \leq -2) = 0.0256 < \alpha = 0.05 \Rightarrow$  reject  $H_0$

5. Draw a Conclusion: We do have enough statistical evidence to conclude that the average weight is less than 16 oz at 0.05 significant level

```
# P-value
pt(-2, 48)
```

```
## [1] 0.02558797
```

```
curve(dt(x, 48), from = -3.5, to = 3.5, n = 1001, lwd = 1.5, col = "blue",
      xlab = "Test statistic",
      ylab = "Density", las = 1)
abline(v = -2)
x_grid <- seq(-3.5, 3.5, 0.01)
y_grid <- dt(x_grid, 48)
polygon(c(x_grid[x_grid < -2], rev(x_grid[x_grid < -2])),
        c(y_grid[x_grid < -2], rep(0, length(y_grid[x_grid < -2]))), col = "pink")
abline(h = 0, lty = 3, col = "gray")
axis(1, at = -2, labels = expression(t["obs"]),
     line = -0.75)
abline(v = -qt(0.95, 48), lty = 2, col = "red")
arrows(-1.68, 0.25, x1 = -3.5, length = 0.1,
       lty = 2, col = "red")
text(-3, 0.3, "Rejection Region", col = "red", cex = 0.75)
```



### Blood Test Example

A series of blood tests were run on a particular patient over five days. It is of interest to determine if the mean blood protein for this patient differs from 7.25, the value for healthy adults. Suppose the sample mean ( $n = 20$ ) is 7.35 and sample standard deviation is 0.5. Perform a hypothesis test using significance level of 0.05

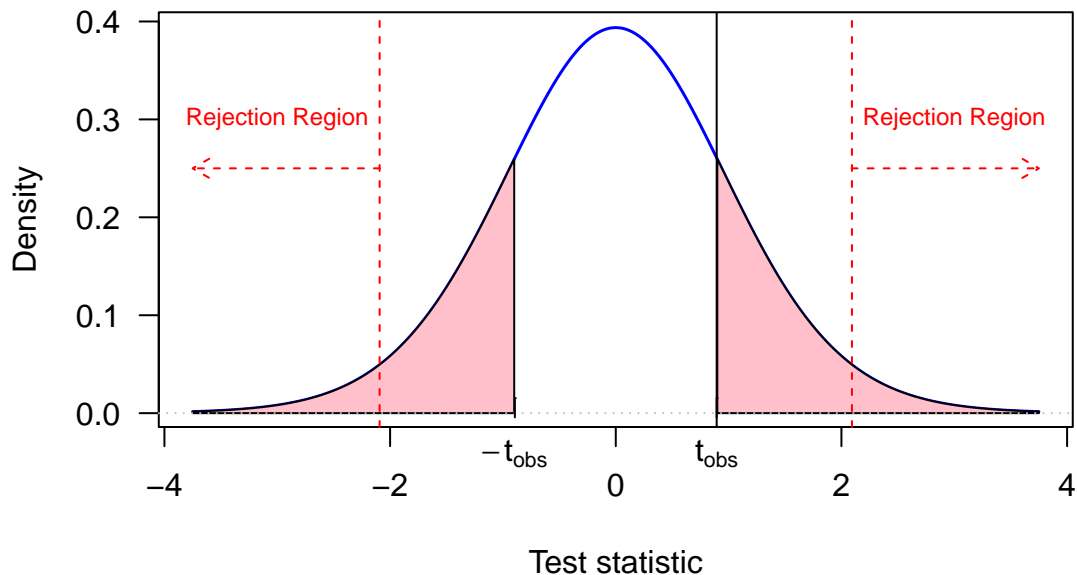
1.  $H_0 : \mu = 7.25$  vs.  $H_a : \mu \neq 7.25$
2.  $t_{obs} = \frac{7.35 - 7.25}{0.5/\sqrt{20}} = 0.8944$



3. P-value:  $2 \times \mathbb{P}(t^* \geq 0.8944) = 0.3823 > 0.05$

4. We do not have enough statistical evidence to conclude that the mean blood protein is different from 7.25 at 5% significant level

```
curve(dt(x, 19), from = -3.75, to = 3.75, n = 1001, lwd = 1.5, col = "blue",
      xlab = "Test statistic",
      ylab = "Density", las = 1)
x_grid <- seq(-3.75, 3.75, 0.01)
y_grid <- dt(x_grid, 19)
t_star = 0.8944
polygon(c(x_grid[x_grid < -t_star], rev(x_grid[x_grid < -t_star])),
       c(y_grid[x_grid < -t_star], rep(0, length(y_grid[x_grid < -t_star]))),
       col = "pink")
polygon(c(x_grid[x_grid > t_star], rev(x_grid[x_grid > t_star])),
       c(y_grid[x_grid > t_star], rep(0, length(y_grid[x_grid > t_star]))),
       col = "pink")
abline(h = 0, lty = 3, col = "gray")
axis(1, at = 0.8944, labels = expression(t["obs"]), line = -0.75)
axis(1, at = -0.8944, labels = expression(-t["obs"]), line = -0.75)
abline(v = -qt(0.975, 19), lty = 2, col = "red")
abline(v = qt(0.975, 19), lty = 2, col = "red")
abline(v = 0.8944)
arrows(-2.09, 0.25, x1 = -3.75, length = 0.1,
       lty = 2, col = "red")
arrows(2.09, 0.25, x1 = 3.75, length = 0.1,
       lty = 2, col = "red")
text(-3, 0.3, "Rejection Region", col = "red", cex = 0.75)
text(3, 0.3, "Rejection Region", col = "red", cex = 0.75)
```

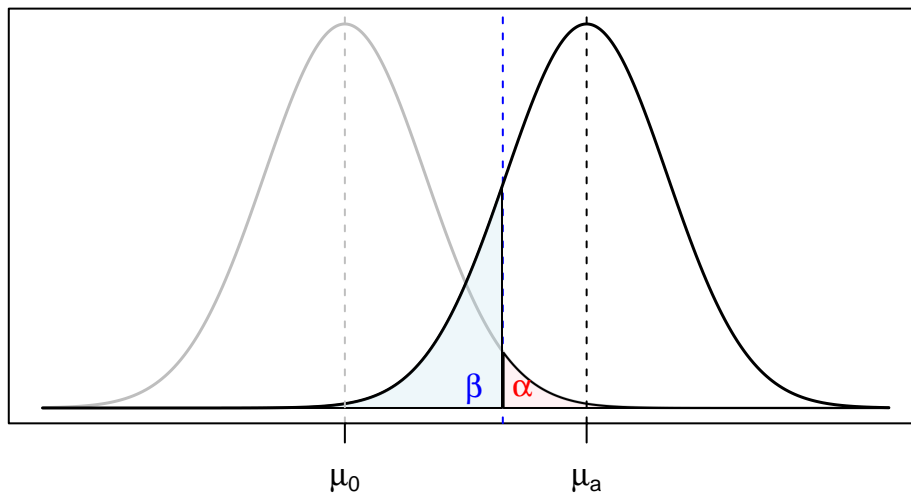


Type I and Type II Errors

```

library(scales)
curve(dnorm(x), from = -3.75, to = 6.75, n = 1001, lwd = 1.5, col = "gray",
      xlab = "",
      ylab = "", las = 1,
      xaxt = "n", yaxt = "n")
abline(v = 1.96, lty = 2, col = "blue")
x_grid <- seq(-3.75, 6.75, 0.01)
y_grid <- dnorm(x_grid)
polygon(c(x_grid[x_grid > 1.96], rev(x_grid[x_grid > 1.96])),
        c(y_grid[x_grid > 1.96], rep(0, length(y_grid[x_grid > 1.96]))),
        col = alpha("pink", 0.2))
curve(dnorm(x, mean = 3), from = -3.75, to = 6.75, n = 1001, lwd = 1.5, add = T,
      xlab = "",
      ylab = "", las = 1)
x_grid <- seq(-3.75, 6.75, 0.01)
y_grid <- dnorm(x_grid, mean = 3)
polygon(c(x_grid[x_grid < 1.96], rev(x_grid[x_grid < 1.96])),
        c(y_grid[x_grid < 1.96], rep(0, length(y_grid[x_grid < 1.96]))),
        col = alpha("lightblue", 0.2))
abline(v = 0, lty = 2, col = "gray")
abline(v = 3, lty = 2)
axis(1, at = 0, labels = expression(mu[0]))
axis(1, at = 3, labels = expression(mu[a]))
text(2.2, 0.02, expression(alpha), col = "red")
text(1.6, 0.02, expression(beta), col = "blue")

```



## Power analysis

```
library(asbio)
```

```
## Loading required package: tcltk
```

```
power.z.test(sigma = 10, n = 25, power = NULL, alpha = 0.05,
             effect = 4, test = c("one.tail"))
```

```
## $sigma
## [1] 10
##
## $n
## [1] 25
##
## $power
## [1] 0.63876
##
## $alpha
## [1] 0.05
##
## $effect
## [1] 4
##
## $test
## [1] "one.tail"
```

```
power.z.test(sigma = 10, n = NULL, power = 0.8, alpha = 0.05,
             effect = 4, test = c("one.tail"))
```

```
## $sigma
## [1] 10
##
## $n
## [1] 38.64098
##
## $power
## [1] 0.8
##
## $alpha
## [1] 0.05
##
## $effect
## [1] 4
##
## $test
## [1] "one.tail"
```

```
power.t.test(n = 25, delta = 4, sd = 10, sig.level = 0.05,
             power = NULL, type = "one.sample", alternative = "one.sided")
```

```
##
##      One-sample t test power calculation
##
##              n = 25
##             delta = 4
##              sd = 10
##             sig.level = 0.05
```

```
##           power = 0.617259
## alternative = one.sided
```

```
power.t.test(n = NULL, delta = 4, sd = 10, sig.level = 0.05,
             power = 0.8, type = "one.sample", alternative = "one.sided")
```

```
##
## One-sample t test power calculation
##
##           n = 40.02908
##           delta = 4
##           sd = 10
##           sig.level = 0.05
##           power = 0.8
## alternative = one.sided
```