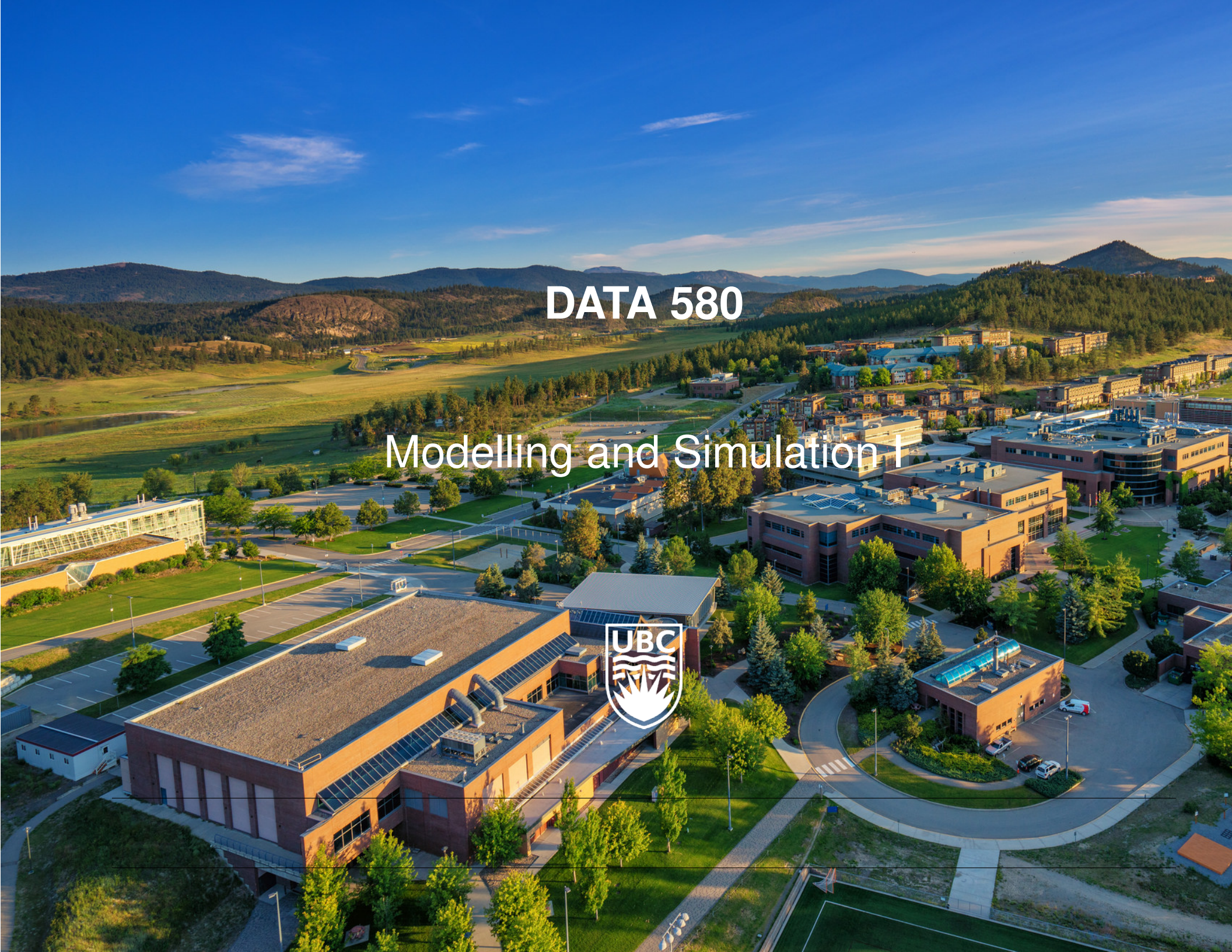

DATA 580
Modelling and Simulation I
UBC

Modelling Continuous Data

Non-uniform Random Variables - Simulation via Inverse cdf

Weibull and Lognormal Random Variables

Normality, and Independence of the Sample Mean and Standard Deviation

Distributions based on the Normal: χ^2, t and F

The Sample Mean and its Standard Error: Confidence Interval

Application to Monte Carlo Integration

Modelling Continuous Data

Examples:

1. **Errors can occur in the production of two-dimensional medical images.**

   A probability model for the proportions of such errors can be of use for quality assurance.

   For example, it is useful to know whether a machine is producing an unusually high proportion of errors.

2. **Probability models are also of use in reliability: what is the probability that an individual or a machine will survive for a given amount of time? Did this component burn out unusually early?**

Modelling Continuous Data

Example:

A probability model for the proportion of impurity in samples of iron ore is

$$f(x) = (\alpha + 1)x^{\alpha}, \quad 0 \leq x \leq 1$$

where α is an *unknown parameter*.

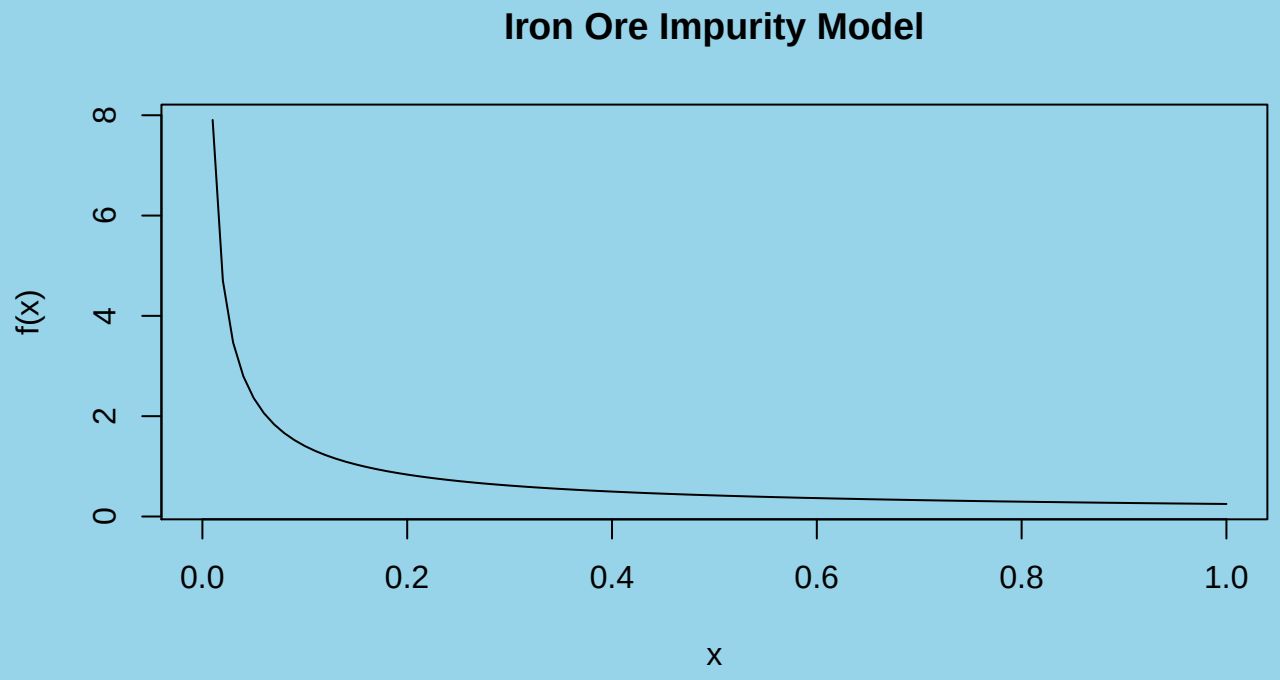
The function $f(x)$ is another example of a *probability density function* (pdf), since it is nonnegative and it integrates to 1.

The pdf is highest at values of x that are most probable.

Visualizing the pdf

The density curve can be plotted using the `curve()` function, which takes a function of x as its first argument.

```
alpha <- -0.75  #  alpha is set to -0.75
curve((alpha+1)*x^alpha, ylab="f(x)",
      main="Iron Ore Impurity Model")
```



Calculation of Probabilities Using the pdf

Recall: the probability that a random variable X with density function $f(x)$ takes a value in an interval $[a, b]$ is calculated as

$$P(a \leq X \leq b) = \int_a^b f(x)dx.$$

The *cumulative distribution function* (cdf) is

$$F(y) = P(X \leq y) = \int_{-\infty}^{y} f(x)dx = \int_0^y (\alpha+1)x^{\alpha}dx = y^{\alpha+1}, \quad y \in [0, 1].$$

Probabilities of Large Proportions of Impurity

For example, we may interested in knowing whether an observed iron ore impurity value y is unusually large.

We can check this by calculating the probability that the proportion of impurity X exceeds y.

$$P(X > y) = 1 - F(y) = 1 - y^{\alpha+1}.$$

Note that we are assuming $y \in (0, 1)$ here. If $y \geq 1$, the probability would be 0.

If we know that the value of α is -0.5, then The cumulative distribution function is

$$F(y) = y^{0.5} \quad \textbf{so } P(X > y) = 1 - y^{0.5}.$$

Simulating from the Model

We can use the same procedure as we used to simulate exponential random variables: invert the cdf at a sequence of random variables U.

First, let's verify that this makes sense. The mathematics was worked out earlier, to show that if U is a uniform random variable, then when $F^{-1}(U)$ is a random variable with cdf $F(x)$.

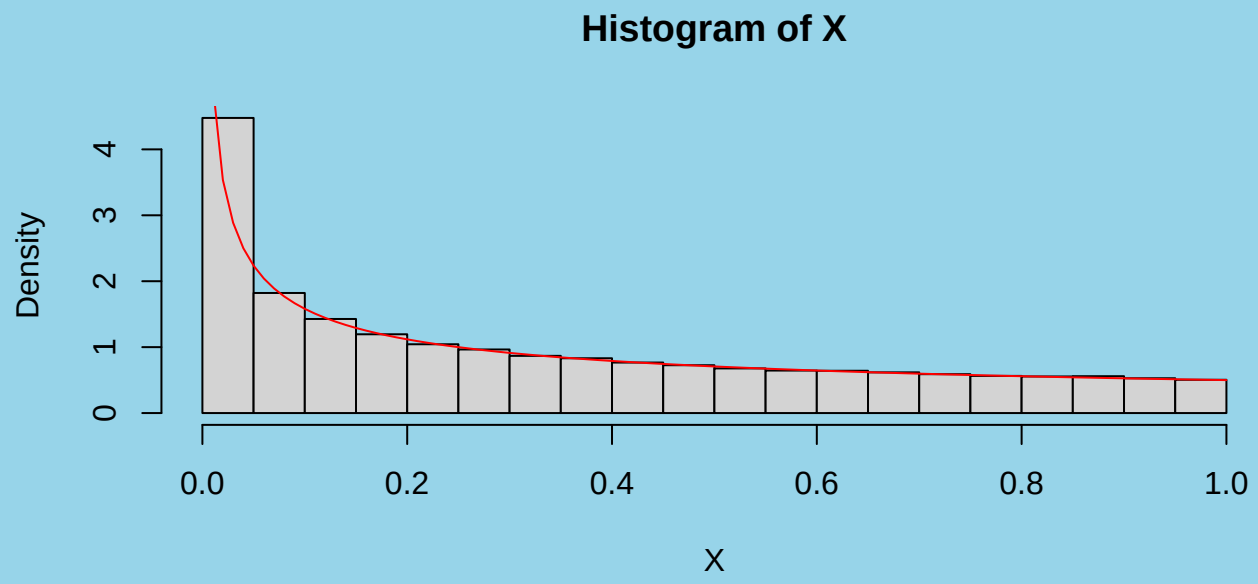
We can also use simulation to show this:

```
U <- runif(100000) # simulate lots of uniforms
X <- U^2 # apply inverse cdf (assume alpha = -.5)
```

Simulating from the Model

Plot the histogram and overlaid density curve

```
hist(X, freq = FALSE)
alpha <- -0.5
curve((alpha + 1)*x^(alpha), 0, 1, add = TRUE, col = "red")
```



The density curve matches the relative frequency histogram closely. Exercise: try this for other values of α.

General Simulation Method: Inverse CDF

If you have a way of calculating the inverse function of the cdf, the following method can be used to convert uniform numbers to the flavour you are targetting:

```
U <- runif(N) # simulate N uniforms
X <- Finv(U) # tranform to the target distribution
                        # using the inverse of the cdf
X
```

Another Example

Suppose X is a random variable with cdf $F(x) = \sin(x)$ for $x \in [0, \pi/2]$.

1. **Is $F(x)$ a true cdf?**

2. **Simulate 10000 random variates from this distribution.**

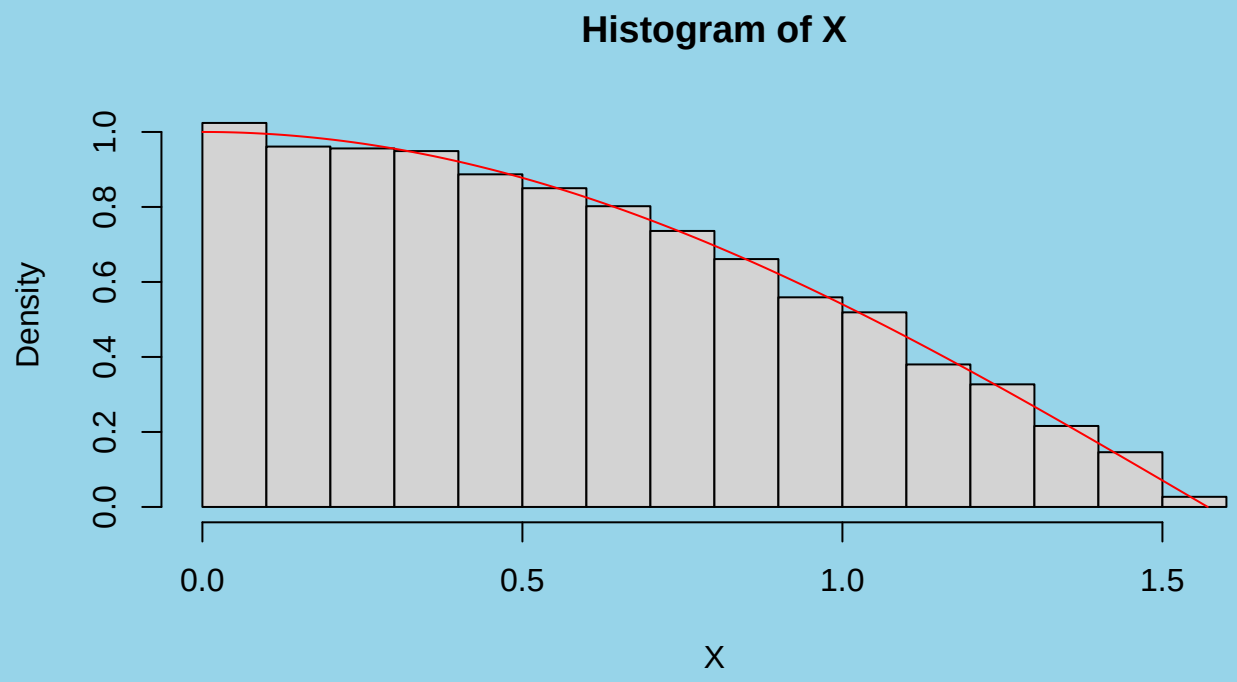
```
U <- runif(10000)
X <- asin(U)
```

Note that the pdf is $f(x) = \cos(x)$ for $x \in [0, \pi/2]$, and 0, otherwise.

Simulating from the Model

Plot the histogram and overlaid density curve

```
hist(X, freq = FALSE)
curve(cos(x), 0, pi/2, add = TRUE, col = "red")
```



The density curve matches the relative frequency histogram closely.

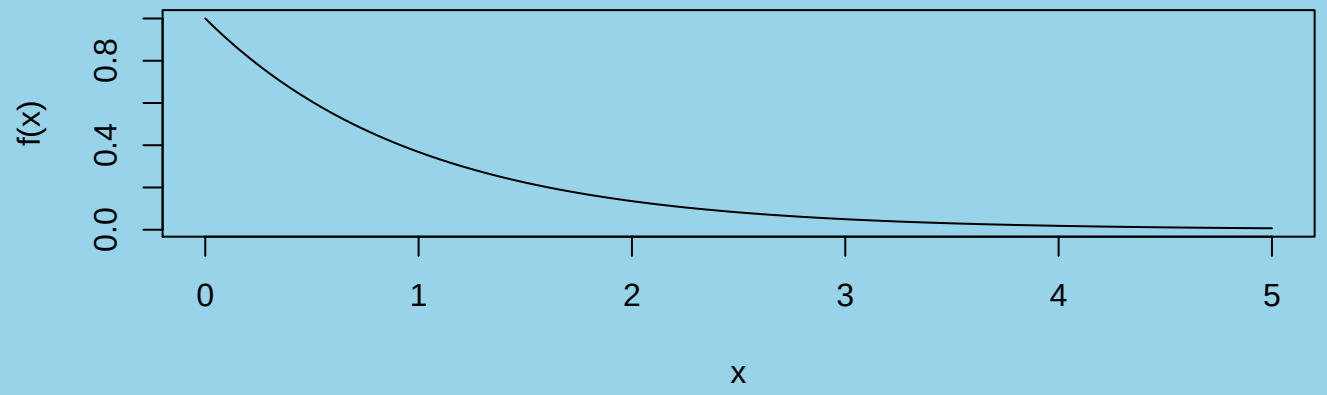
Models for Survival

Recall the exponential distribution, a simple model for a lifetime distribution:

$$f(x) = \lambda e^{-\lambda x}, \quad x > 0$$

and $f(x) = 0$, otherwise.

```
curve(dexp(x), 0, 5, ylab="f(x)")
```



The density is highest near 0. When we simulate from this distribution we get a lot of unrealistically low values:

```
X <- rexp(9); X

## [1] 2.584 1.308 1.329 1.232 0.597 1.266 0.459 1.228 8.319
```

The Weibull Distribution

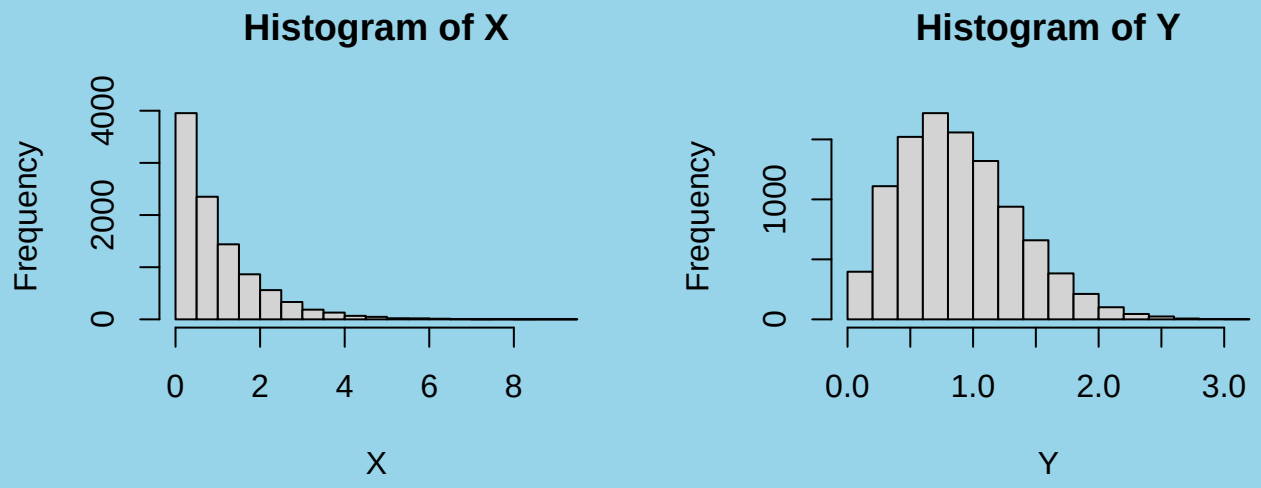
If we take the square root of X, the behaviour is different:

```
sqrt(X)
```

```
## [1] 1.607 1.144 1.153 1.110 0.773 1.125 0.677 1.108 2.884
```

The Weibull Distribution

```
X <- rexp(10000)
Y <- sqrt(X)
par(mfrow=c(1, 2))
hist(X); hist(Y)
```



The Weibull Distribution

In general, a Weibull random variable is defined as a power of an exponential random variable.

That is, if X is exponential, λ, then $Y = X^{1/\beta}$ is Weibull with parameters β and λ. β controls the shape of the distribution and λ controls the scale.

The cdf of Y is

$$F(y) = P(Y \leq y) = P(X^{1/\beta} \leq y) = P(X \leq y^{\beta}) = 1 - e^{-\lambda y^{\beta}}$$

where we used the exponential cdf of X in the middle of the above derivation.

Differentiating $F(y)$ gives you the pdf of the Weibull.

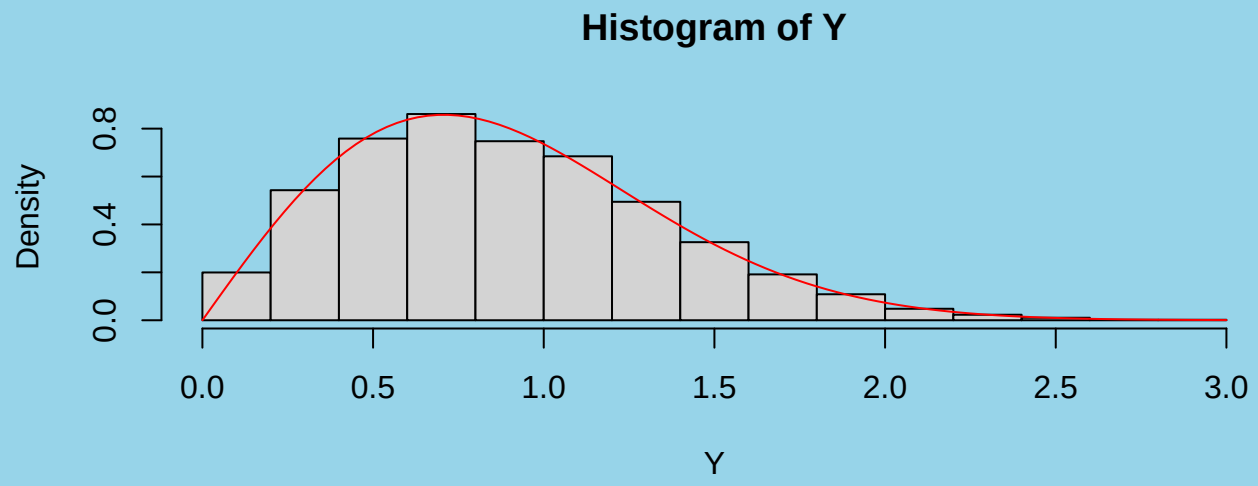
Note:

```
dweibull(x, shape = 2, scale = 1) # Weibull pdf with beta = 2, lambda = 1
pweibull(x, shape = 2, scale = 1) # cdf
rweibull(n, shape = 2, scale = 1) # rng
```

The Weibull Distribution

Simulating and comparing with the density curve:

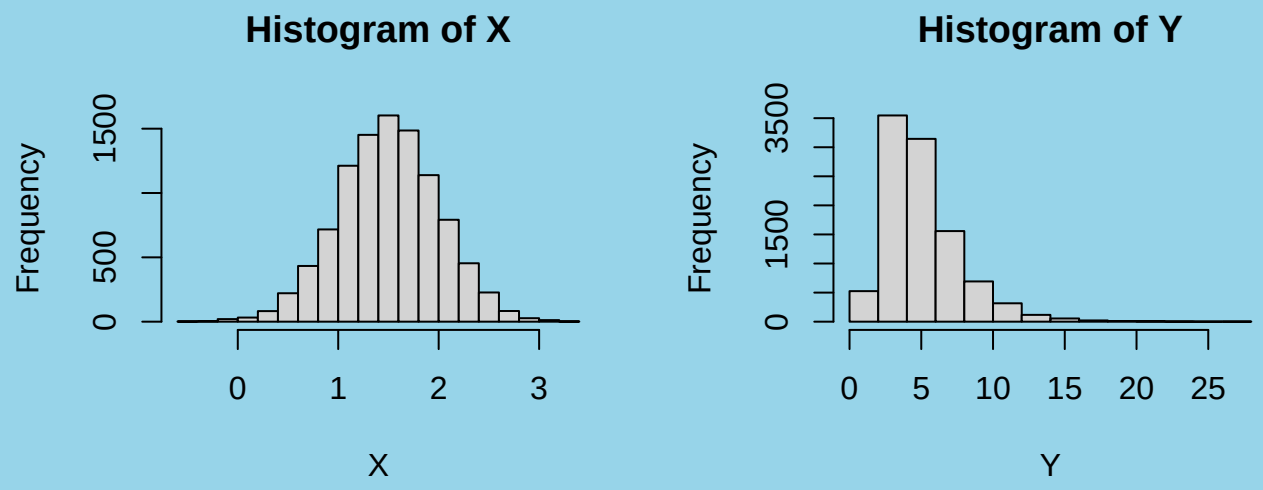
```
Y <- rweibull(10000, shape = 2, scale = 1)
hist(Y, freq = FALSE)
curve(dweibull(x, shape = 2, scale = 1), 0, 3, col = "red", add = TRUE)
```



The Lognormal Distribution

If we take the exponential of X, where X is a normal random variable, we obtain a lognormal random variable, another model for survival times:

```
X <- rnorm(10000, mean = 1.5, sd = 0.5)
Y <- exp(X)
par(mfrow=c(1, 2))
hist(X); hist(Y)
```



Example: Liver Transplant Waiting Times

Data in the `transplant` data slide in the *survival* package relate to waiting times for liver transplant patients.

We consider the males who have type B blood here:

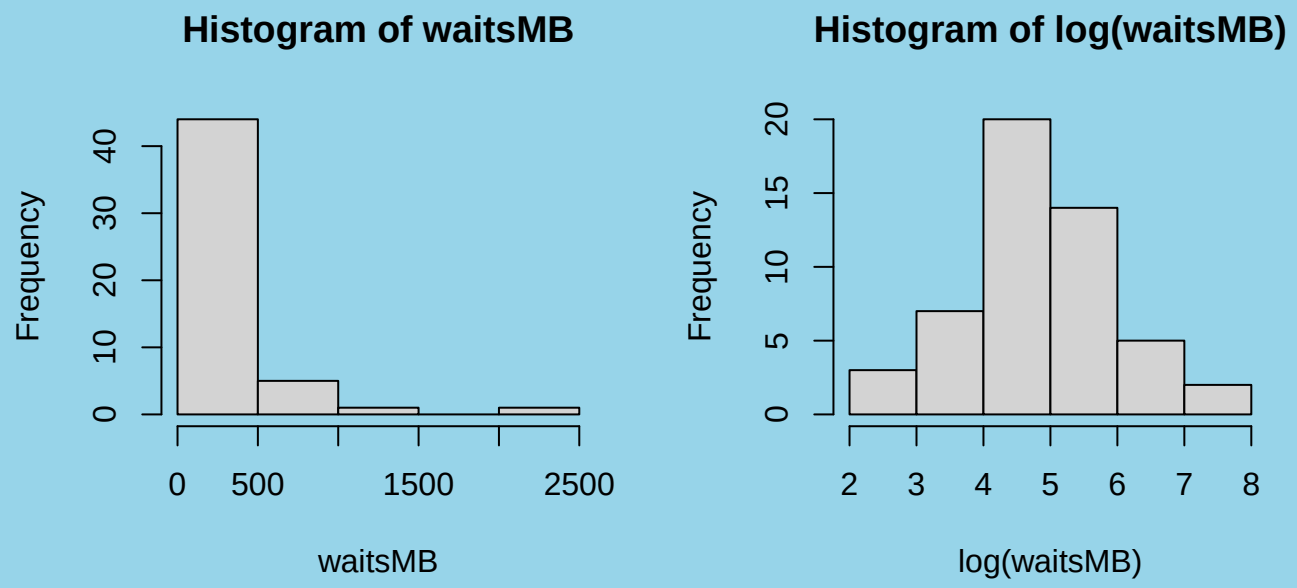
```
library(survival)
waitsMB <-  subset(transplant,
                           sex=="m" & abo=="B" )$futime
```

These data are well approximated by the lognormal distribution.

Example: Liver Transplant Waiting Times

Raw data on left. Log of raw data on right.

```
par(mfrow=c(1, 2))
 hist(waitsMB); hist(log(waitsMB))
```

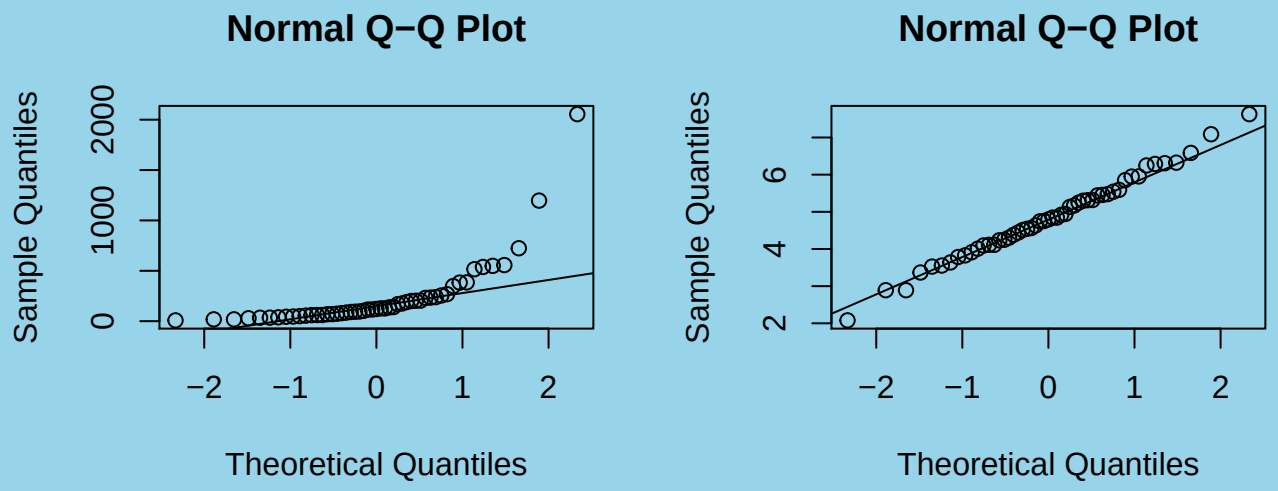


The log plot looks very normal.

QQ-Plot: A Better Way of Checking Normality

Look for a straight line. If you see it, you have normality. If not, you don't.

```
par(mfrow=c(1, 2))
qqnorm(waitsMB); qqline(waitsMB)
qqnorm(log(waitsMB)); qqline(log(waitsMB))
```



The plot on the right looks much straighter than the one on the left.

Random Variables Constructed from Normals

Construction starts with the standard normal random variable

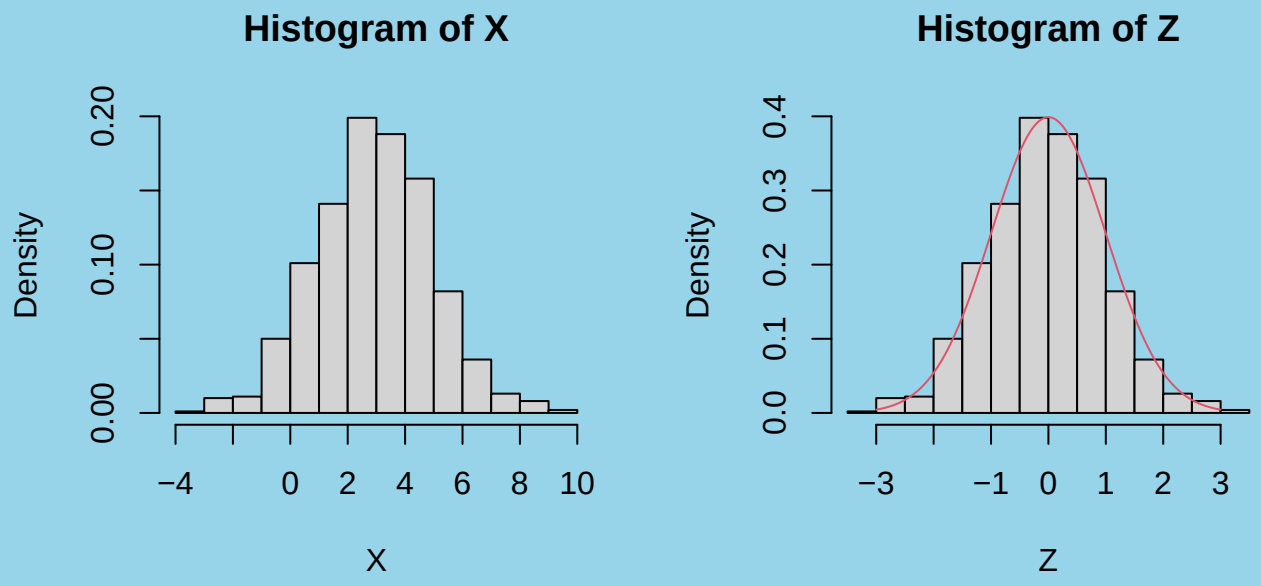
- **Let Y be a normal random variable with mean μ and standard deviation σ**
- $$Z = \frac{Y - \mu}{\sigma} \tag{1}$$

  is a standard normal random variable.

Transforming Normal to Standard Normal

Check standardization by simulation:

```
X <- rnorm(1000, mean =3, sd = 2); Z <- (X-3)/2
par(mfrow=c(1, 2))
hist(X, freq=FALSE); hist(Z, freq=FALSE)
curve(dnorm(x), -3, 3, col=2, add=TRUE)
```



The distribution of `Z` is identical to that of `X`, therefore normal. N(0,1) pdf curve matches.

The χ^2 Random Variables

- **Squaring Z leads to a χ^2 random variable on 1 degree of freedom.**

- **Note that**

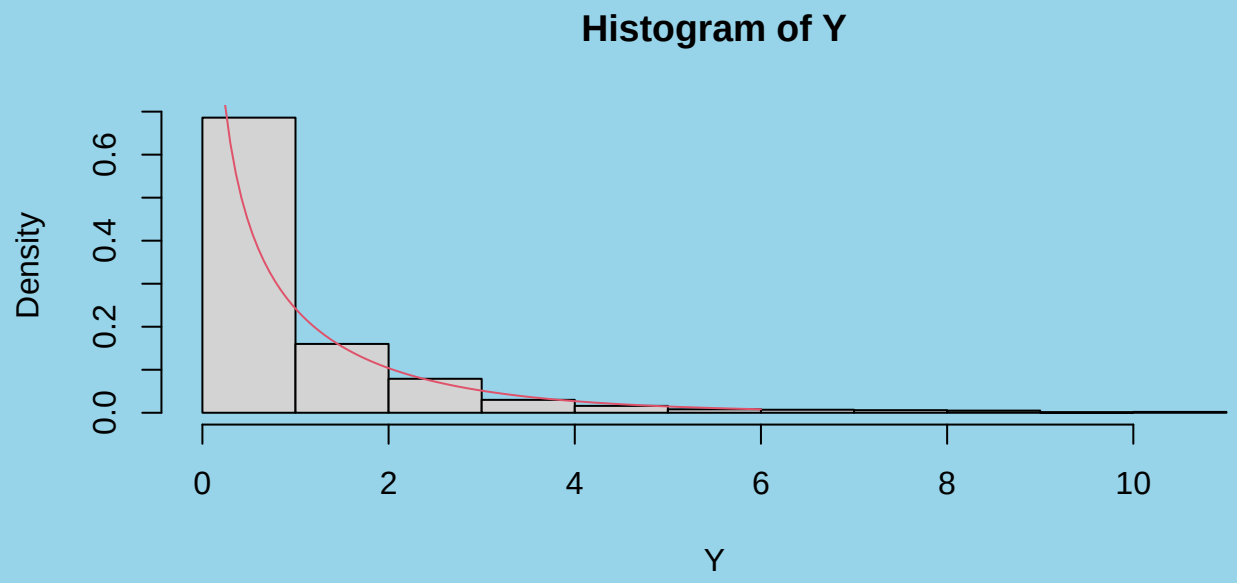
$$E[Z^2] = 1 \tag{2}$$

  $\rightsquigarrow$ a χ^2 random variable on 1 degree of freedom has expected value 1.

On the next slide, we check that Z^2 is χ^2 by simulation, using `dchisq()`.

The χ^2 Random Variables

```
Y <- Z^2
hist(Y, freq=FALSE)
curve(dchisq(x, df = 1), 0, 6, add=TRUE, col=2)
```



χ^2 random variables can be generated using `rchisq()`:

```
rchisq(5, df = 1)

## [1] 0.00385 6.82201 0.00743 0.29101 0.12279
```

The χ^2 Random Variables

- **If $Z_1, \ldots, Z_n$ is a sequence of n independent standard normal random variables, then**

$$X = \sum_{j=1}^{n} Z_j^2 \tag{3}$$

  is a $\chi^2_{(n)}$ random variable on n degrees of freedom.

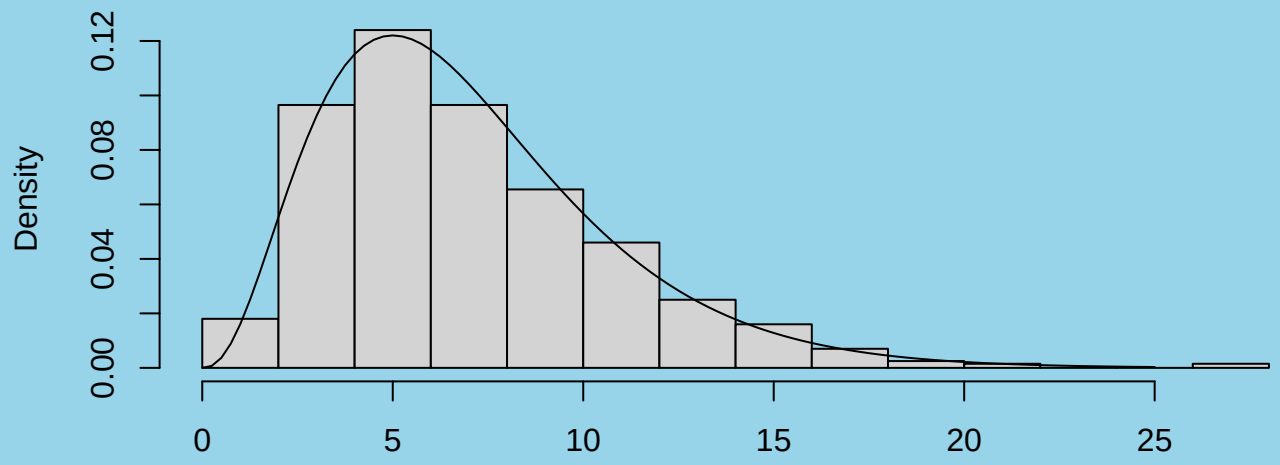
-

$$E[\sum_{j=1}^{n} Z_j^2] = \sum_{j=1}^{n} E[Z_j^2] = n \tag{4}$$

The χ^2 Random variables

1000 simulated values of X for the case where $n = 7$

```
X <-  rchisq(1000, df = 7)
hist(X, freq = FALSE, main = " ")
curve(dchisq(x, df = 7), from = 0, to = 25, add = TRUE)
```



A Connection between S^2 and the χ^2 Distribution

- **if μ was known, but σ^2 was unknown, we could estimate σ^2 from a sample of Y's in an unbiased manner by using the formula**

$$\widehat{\sigma}^2 = \frac{1}{n}\sum_{j=1}^{n}(Y_j - \mu)^2. \tag{5}$$

- **Unbiasedness follows from noting that**

$$E[\widehat{\sigma}^2] = \frac{1}{n}\sum_{j=1}^{n}E[(Y_j - \mu)^2] = \sigma^2. \tag{6}$$

A Connection between S^2 and the χ^2 Distribution

- **Observe that**

$$n\widehat{\sigma^2}/\sigma^2 = n\frac{\sum_{j=1}^{n}(Y_j-\mu)^2/n}{\sigma^2} = \sum_{j=1}^{n}\left(\frac{Y_j-\mu}{\sigma}\right)^2$$

  is a χ^2 random variable on n degrees of freedom.

- **Usually μ is not known. Then, S_Y^2 (which is $\widehat{\sigma^2}$ with μ replaced by $\bar{Y}$) is an unbiased estimator for σ^2, and $(n-1)S_Y^2/\sigma^2$ is a χ^2 random variable on $n-1$ degrees of freedom.**

Demonstration of Connection Between S^2 and χ^2

Let us consider a random samples of $n = 20$ normal random variables, each with mean 3 and standard deviation 2, and let us draw 1000 such samples.

We will show that $(n-1)S^2/\sigma^2$ has a χ^2 distribution on 19 degrees of freedom:

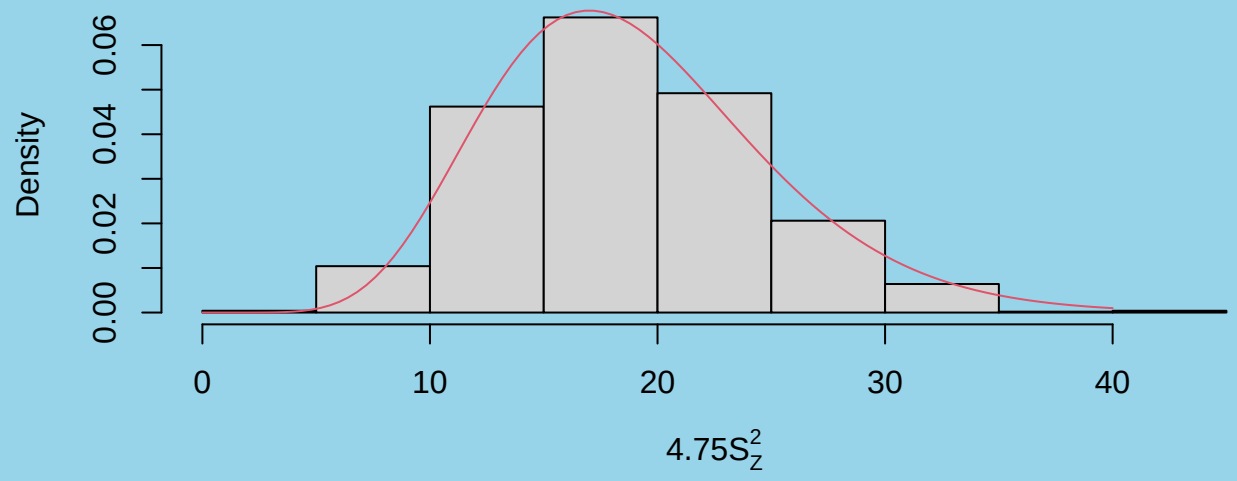
```
m <- 1000; n <- 20; sigma <- 2
# m samples of size n:
Z <- matrix(rnorm(m*n, mean = 3, sd = sigma), nrow=n)
S2z <- apply(Z, 2, var)
sqrt(S2z[1:5]) # the first 5 sample standard deviations


## [1] 1.28 2.41 1.86 1.77 2.27
```

These are scattered about $\sigma = 2$.

Demonstration of Connection Between S^2 and χ^2

Compare the relative frequency histogram of $(n-1)S^2/\sigma^2$ with the χ^2 density curve:

```
hist((n-1)*S2z/sigma^2, freq = FALSE, main = " ")
curve(dchisq(x, df = 19), 0, 40, col = 2, add = TRUE)
```



The histogram approximates the density curve closely. Exercise: check this result for other sample sizes, such as $n = 2, 5, 10, 50$. Try different values of the μ and σ as well.

The F random variable

If X_1 and X_2 are independent χ^2 random variables on m and n degrees of freedom, respectively, then

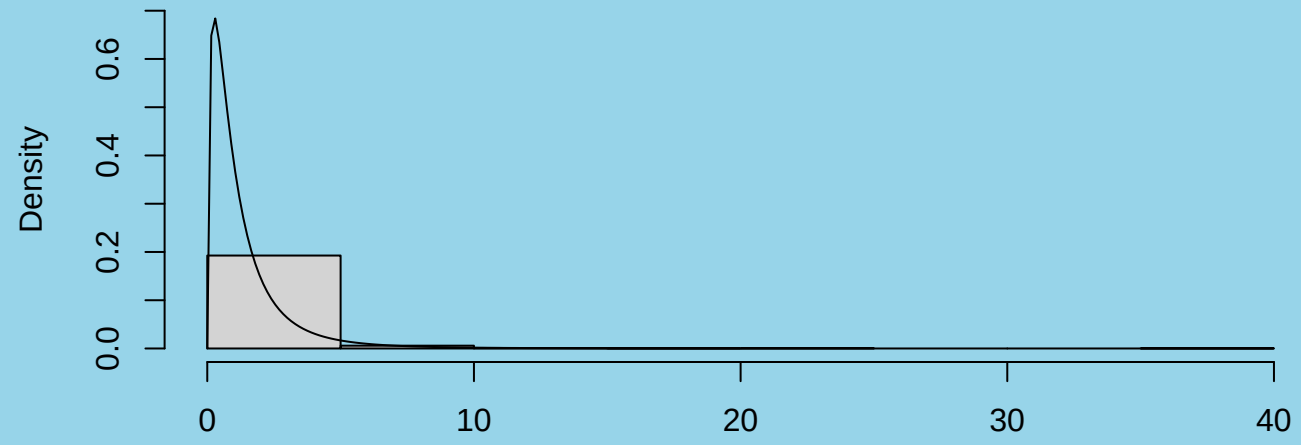
$$F = \frac{X_1/m}{X_2/n} \tag{7}$$

is an F random variable on m and n degrees of freedom. m is sometimes referred to as the numerator degrees of freedom, and n is the denominator degrees of freedom.

The F random variable

1000 simulated values of F for the case where $m = 3$ and $n = 7$, $F_{(3,7)}$.

```
F <-  rf(1000, df1 = 3, df2 = 7)
hist(F, freq = FALSE, ylim = c(0, 0.7), main = " ")
curve(df(x, df1 = 3, df2 = 7), from = 0, to = 15, add = TRUE)
```



The t random variable

Suppose Z is a standard normal random variable and suppose X is a χ^2 random variable on n degrees of freedom, then

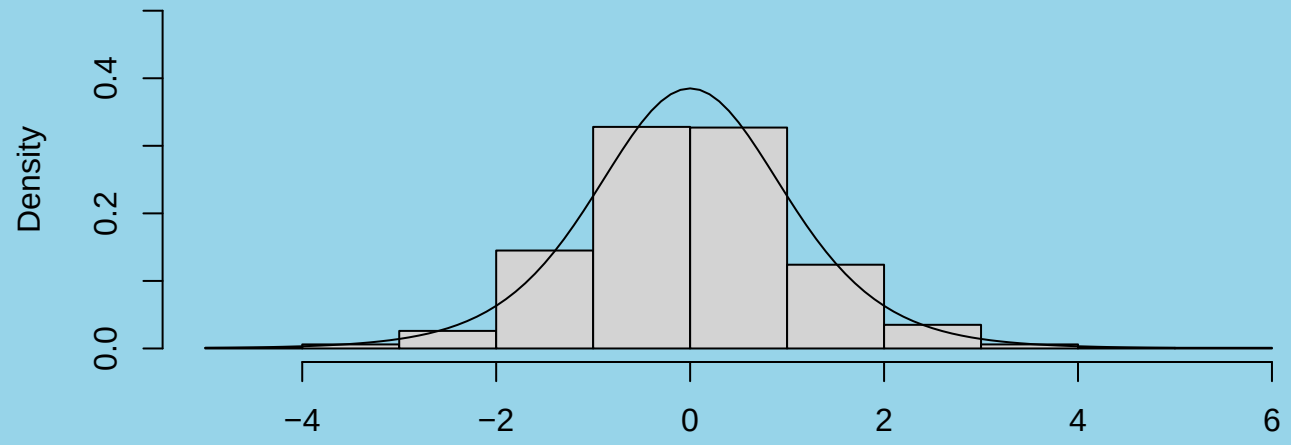
$$T = \frac{Z}{\sqrt{X/n}} \tag{8}$$

is a t random variable on n degrees of freedom, provided that Z and X are independent of each other.

The t random variable

1000 simulated values of t for the case where $n = 7$

```
T <-  rt(1000, df = 7)
hist(T, freq = FALSE, ylim = c(0, .5), main = " ")
curve(dt(x, df = 7), from = -5, to = 5, add = TRUE)
```



Studentizing yields a t random variable

- $\bar{Y}$ **is normally distributed with mean** μ **and variance** σ^2/n**, if the underlying sample consists of** n **uncorrelated normal random variables with common mean** μ **and common variance** σ^2**.**

- **We will demonstrate empirically that** $\bar{Y}$ **and** S_Y^2 **are independent**

- $(n-1)S_Y^2/\sigma^2$ **is a** $\chi^2_{(n-1)}$ **random variable**

- **We will now show by simulation that**

$$\frac{\bar{Y}-\mu}{S_Y/\sqrt{n}} \tag{9}$$

is a t **random variable on** $n-1$ **degrees of freedom.**

Simulation of Distribution of t Statistic

Let us consider a random samples of $n = 20$ normal random variables, each with mean 3 and standard deviation 2, and let us draw 1000 such samples.

We will show that $(\bar{X} - \mu)\sqrt{n}/S$ has a t distribution on 19 degrees of freedom:

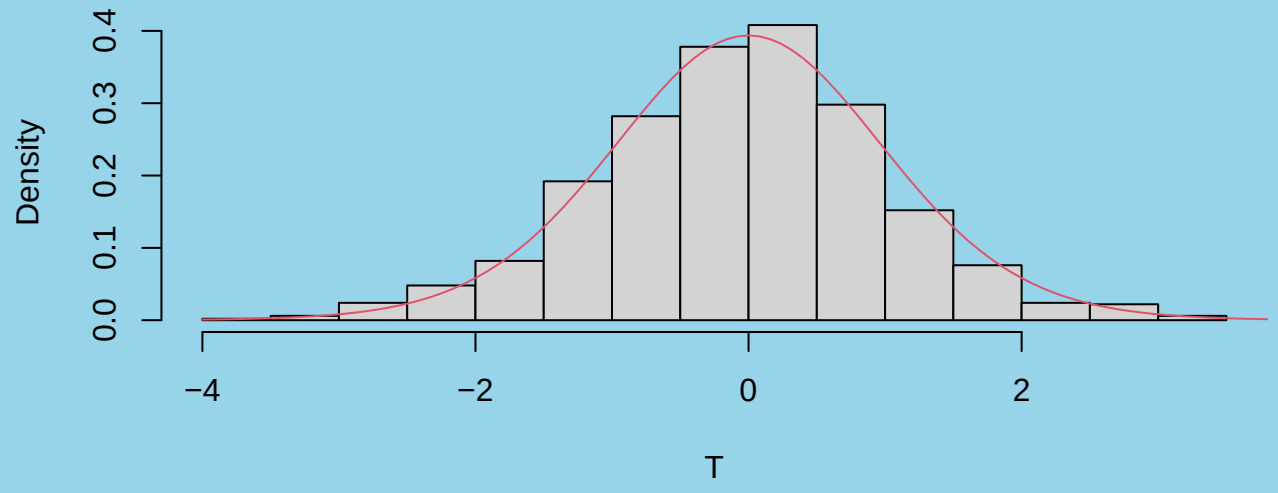
```
m <- 1000; n <- 20; sigma <- 2
# m samples of size n:
Z <- matrix(rnorm(m*n, mean = 3, sd = sigma), nrow=n)
Sz <- apply(Z, 2, sd); xbar <- apply(Z, 2, mean)
T <- sqrt(n)*(xbar - 3)/Sz  # t statistics
T[1:5] # first 5 t statistic values

## [1]  0.8708  0.1152 -0.8730  0.1202 -0.0687
```

These are scattered about 0.0.

Demonstration of Connection Between S^2 and χ^2

Compare the relative frequency histogram of $(\bar{X} - \mu)\sqrt{n}/S$ with the t density curve:

```
hist((xbar - 3)*sqrt(n)/Sz, freq = FALSE, main = " ")
curve(t(x, df = 19), -4, 4, col = 2, add = TRUE)
```



The histogram approximates the density curve closely. Exercise: check this result for other sample sizes, such as $n = 2, 5, 10, 50$. Try different values of the μ and σ as well.

Standard Error of the Mean

The t statistic

$$\frac{\bar{X} - \mu}{S/\sqrt{n}}$$

can be interpreted as the number of $S/\sqrt{n}$ units between $\bar{X}$ and μ.

$S/\sqrt{n}$ is actually an estimate of the standard deviation of $\bar{X}$, for random samples of size n.

$S/\sqrt{n}$ is the estimate of *Standard Error of the Mean*: $\sigma/\sqrt{n}$.

For our simulated samples above, with $\sigma = 2$, $n = 20$, $\sigma/\sqrt{n} = 0.447$, and

```
sd(xbar) # estimated standard error of the mean

## [1] 0.441
```

Independence of the Sample Mean and Standard Deviation

Development of t and F statistics only worked because of independence of the sample mean and standard deviation.

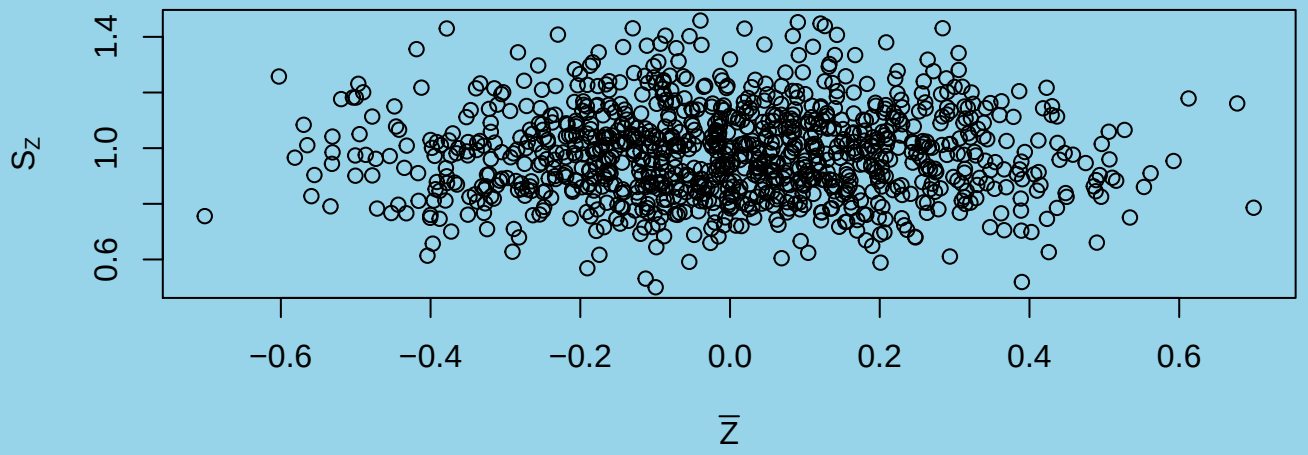
For normally distributed samples, the sample mean and standard deviation are independent.

We can see evidence for this from simulated data. Let us consider a samples of $n = 20$ uncorrelated standard normal random variables, and let us draw 1000 such samples. Here is a way to do this:

```
m <- 1000; n <- 20
Z <- matrix(rnorm(m*n), nrow=n)
zbar <- apply(Z, 2, mean); Sz <- apply(Z, 2, sd)
```

Independence of the Sample Mean and Standard Deviation

```
plot(Sz ~ zbar)
```

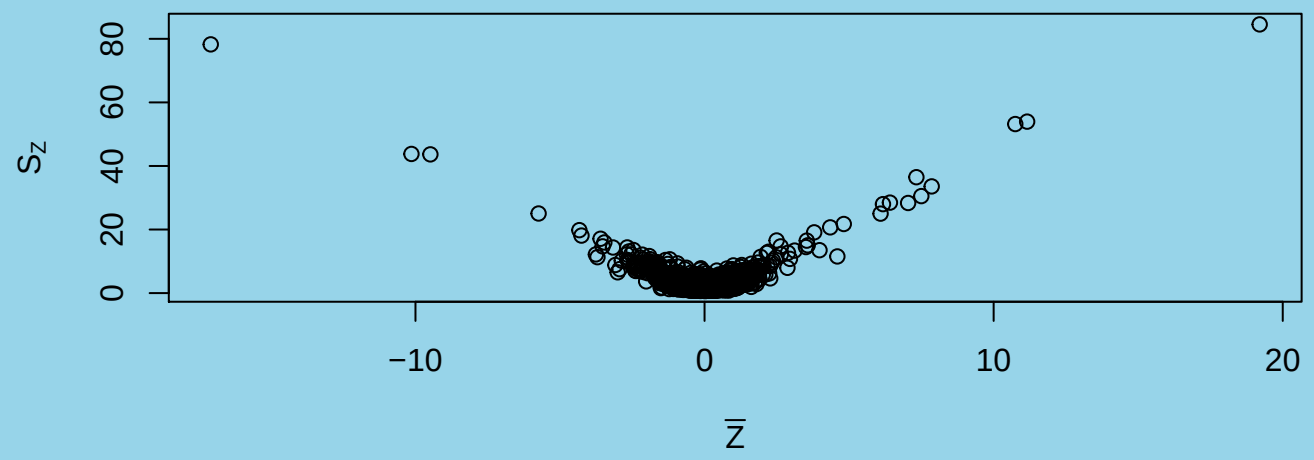


No pattern. It appears to be impossible to predict the standard deviation from the sample mean for normal data.

Dependence of the Sample Mean and Standard Deviation

For non-normal data, the picture is different. The sample mean and standard deviation are no longer independent. t and F statistics will no longer be accurate.

```
m <- 5000
Z <- matrix(rt(m*n, df=2), nrow=n) # t data on 2 df
zbar <- apply(Z, 2, mean); Sz <- apply(Z, 2, sd)
plot(Sz ~ zbar)
```



Clear pattern. The standard deviation is quite predictable from the sample mean for averages of t random variables.

Confidence Intervals for the Mean

Given data of the form $X_1, X_2, \ldots, X_n$ which are a random sample of independent normal random variables from a normal population with mean μ and variance σ^2, we want to estimate μ with a confidence interval.

We will use the usual statistical notation for the sample mean:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

and for the sample variance:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2.$$

Frosted Flakes Example

Two Methods of Measuring Sugar Content:

1. **Lab Analysis - slow, but accurate**

2. **High Performance Liquid Chromatography (HPLC) - fast,**

   ... but is HPLC accurate?

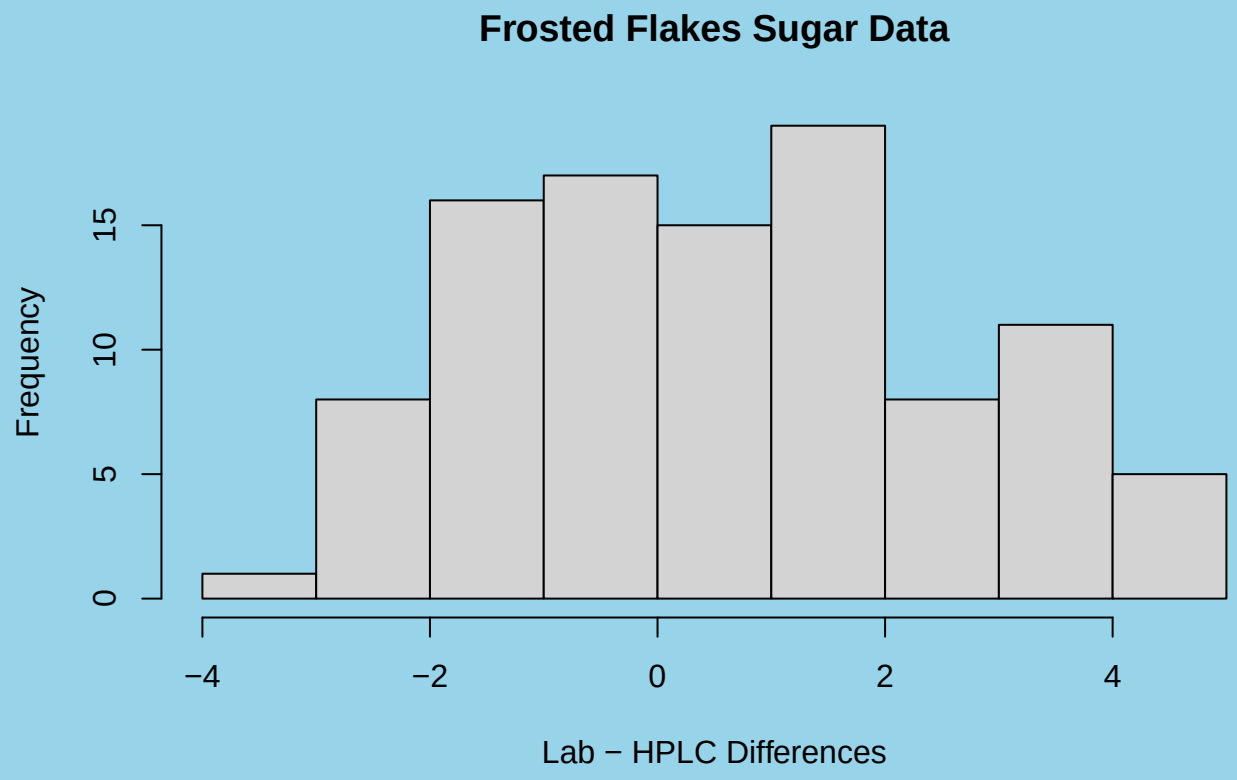
Measurements of each type were taken on 100 frosted flakes samples ...

Frosted Flakes Measurements

```
FFdiff <- scan("FFdiff.txt")
length(FFdiff) # how many sample elements?

## [1] 100

FFdiff[1:10] # first 10 observations

##  [1] -1.2  2.7  1.1 -1.8 -2.8  1.1  2.7  1.9  3.3  3.1
```

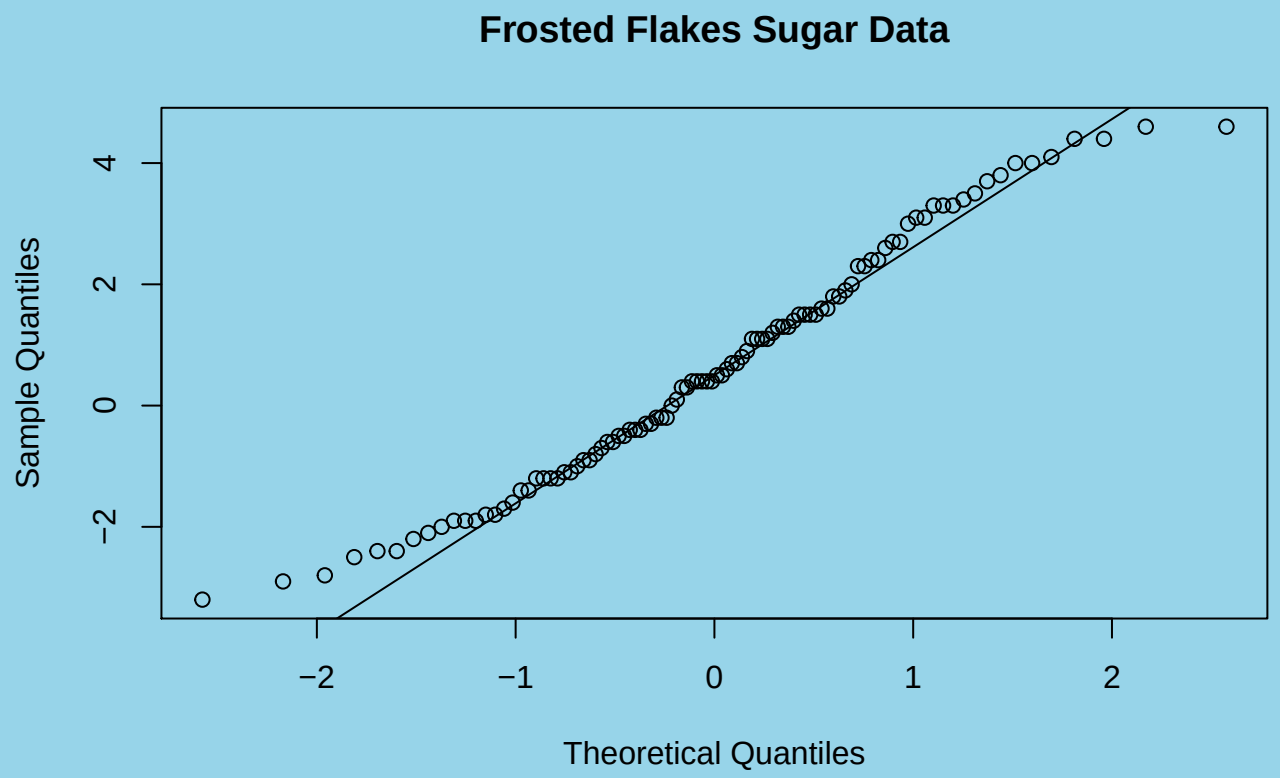
Frosted Flakes Measurements

```
hist(FFdiff, main = "Frosted Flakes Sugar Data", xlab =
  "Lab - HPLC Differences")
```



Frosted Flakes Measurements

```
qqnorm(FFdiff, main = "Frosted Flakes Sugar Data")
qqline(FFdiff)
```



... reasonably normal-looking

What is the expected value of the difference ($\mu = E[X]$)?

Answer: We don't know.

Estimate: $\bar{x} = .622$**.**

How much error is there in this estimate?

Standard Error of Estimator: $\sqrt{\text{Var}(\textbf{Estimator})}$

Standard Error of $\bar{X}$: $\sqrt{\text{Var}(\bar{X})} = \frac{\sigma}{\sqrt{n}}$.

Estimated Standard Error (S.E.): $s/\sqrt{n}$ **= 1.98/10 = .198.**

Example (cont'd)

The approximate probability that $\bar{X}$ differs from μ by less than 2 standard errors is

$$P(-2S.E. < \bar{X} - \mu < 2S.E.) =$$

$$P(-2 < Z < 2) = .9772 - .0228$$

$$= .9544$$

($\bar{X}$ is approximately normal with mean μ and variance σ/n, if n is large enough.)

since

```
pnorm(2) - pnorm(-2)

## [1] 0.954
```

Conclusion

We can be 95.44% confident that the true expected value of the difference in sugar content measurements lies within 2 S.E. of .622:

$$.622 \pm .396.$$

This is an example of a 95.44% confidence interval.
We conclude that HPLC is not accurate. Calibration is required, if HPLC is to be used.

Confidence Interval Formula (Large n)

n independent measurements taken from a population with expected value μ and variance σ^2.

If n is large, then an approximate $100\%(1-\alpha)$ confidence interval for μ is given by

$$\bar{x} \pm z_{\alpha/2}\frac{s}{\sqrt{n}}$$

where $z_{\alpha/2}$ is defined so that

$$P(Z > z_{\alpha/2}) = \alpha/2$$

e.g. $z_{.2/2} = 1.28$ since

```
1 - pnorm(1.28)   #  Obtain 1.28 using    " > 1 - qnorm(.1) "

## [1] 0.1
```

Exercise.

Find a 95% confidence interval for the expected difference in sugar content measurement.

$\alpha = .05$ $z_{.025} = 1.96$ **from**

```
qnorm(1 - .025)

## [1] 1.96
```

The 95% c.i. for μ is given by

$$\bar{x} \pm z_{.025}\textbf{S.E.} = .622 \pm 1.96(.198) = .622 \pm .388$$

Exercise.

Find a 90% confidence interval for the expected difference in sugar content measurement.

$\alpha = .1$

$z_{.05} = 1.645$ **from**

```
qnorm(1 - .05)

## [1] 1.64
```

The 90% c.i. for μ is given by

$$.622 \pm 1.645(.198) =$$

$$.622 \pm .326$$

A Small Sample Confidence Interval for μ (Optional)

Define the upper percentile of the t distribution as $t_{\alpha,\nu}$ in

$$P(T > t_{\alpha,\nu}) = \alpha.$$

Here T has a t distribution on ν degrees of freedom. Use `qt(1-alpha, nu)`**.**

Then we can say that

$$P\left(-t_{\alpha/2,n-1} < \frac{\bar{X} - \mu}{S/\sqrt{n}} < t_{\alpha/2,n-1}\right) = 1 - \alpha.$$

A Small Sample Confidence Interval for μ (Optional)

Therefore,

$$P\left(\bar{X} - t_{\alpha/2,n-1}S/\sqrt{n} < \mu < \bar{X} + t_{\alpha/2,n-1}S/\sqrt{n}\right) = 1 - \alpha.$$

and

$$\left(\bar{X} - t_{\alpha/2,n-1}S/\sqrt{n}, \bar{X} + t_{\alpha/2,n-1}S/\sqrt{n}\right) \tag{10}$$

defines a $1 - \alpha$ confidence interval for μ.

Example: Small Sample Confidence interval for μ (Optional)

Find a 95% confidence interval for the expected value of concentration measurements taken from a chemical process. Sample measurements are

204	190	202	207
204	202	201	195

Example: Small Sample Confidence interval for μ (Optional)

If X denotes a concentration measurement, then $\bar{x} = 201.$, $s = 5.50$, and $n = 8$ so a 95% confidence interval for $\mu = E[X]$ is

$$\bar{x} \pm t_{.025,7}\frac{s}{\sqrt{8}}$$

$$= 201 \pm 2.365(5.5)/\sqrt{8}$$

$$= 201 \pm 4.60$$

since

```
qt(1 - .025, 7)

## [1] 2.36
```

Application to Monte Carlo Integration

Suppose $g(x)$ is any function that is integrable on the interval $[a, b]$.

The integral

$$\int_a^b g(x)dx$$

gives the area of the region with $a < x < b$ and y between 0 and $g(x)$ (where negative values count towards negative areas).

Monte Carlo integration uses simulation to obtain approximations to these integrals. It relies on the law of large numbers.

Monte Carlo Integration

This law says that a sample mean from a large random sample will tend to be close to the expected value of the distribution being sampled.

If we can express an integral as an expected value, we can approximate it by a sample mean.

We can assess the error in the simulation using the standard error and a confidence interval.

Monte Carlo Integration

For example, let $U_1, U_2, \ldots, U_n$ be independent uniform random variables on the interval $[a, b]$. These have density $f(u) = 1/(b-a)$ on that interval. Then

$$E[g(U_i)] = \int_a^b g(u) \frac{1}{b-a} du$$

so the original integral $\int_a^b g(x)dx$ can be approximated by $b-a$ times a sample mean of $g(U_i)$.

Example

To approximate the integral $\int_0^1 x^4 dx$, use the following lines:

```
u <- runif(100000)
mean(u^4)   #  Compare with the exact answer, $0.2$


## [1] 0.201
```

Calculate the standard error.

```
SE <- sd(u^4)/sqrt(100000); SE

## [1] 0.000845
```

A 95% confidence interval for the integral is

```
mean(u^4) + c(-1.96, 1.96)*SE

## [1] 0.199 0.202
```

Example

To approximate the integral $\int_2^5 \sin(x)dx$, use the following lines:

```
u <- runif(100000, min = 2, max = 5)
mean(sin(u))*(5-2)   #  true value can be shown to be -0.700.


## [1] -0.7
```

Calculate the standard error.

```
SE <- sd(sin(u)*(5-2))/sqrt(100000); SE

## [1] 0.00619
```

A 95% confidence interval for the integral is

```
mean(sin(u))*(5-2) + c(-1.96, 1.96)*SE

## [1] -0.712 -0.688
```

Multiple Integration

Now let $V_1, V_2, \ldots, V_n$ be an additional set of independent uniform random variables on the interval $[0, 1]$, and suppose $g(x, y)$ is now an integrable function of the two variables x and y. The law of large numbers says that

$$\lim_{n\to\infty} \sum_{i=1}^{n} g(U_i, V_i)/n = \int_0^1 \int_0^1 g(x, y)dxdy$$

with probability 1.

So we can approximate the integral $\int_0^1 \int_0^1 g(x, y)dxdy$ by generating two sets of independent uniform pseudorandom variates, computing $g(U_i, V_i)$ for each one, and taking the average.

Example

Approximate the integral $\int_3^{10}\int_1^7 \sin(x-y)dxdy$ using the following:

```
U <- runif(100000, min = 1, max = 7)
V <- runif(100000, min = 3, max = 10)
mean(sin(U - V))*42


## [1] 0.116
```

Calculate the standard error.

```
SE <- sd(sin(U-V)*42)/sqrt(100000); SE

## [1] 0.094
```

A 95% confidence interval for the integral is

```
mean(sin(U-V))*42 + c(-1.96, 1.96)*SE

## [1] -0.0679  0.3006
```

The factor of $42 = (7-1)(10-3)$ compensates for the joint density of U and V being $f(u,v) = 1/42$.

Using non-uniform pseudorandom numbers (Optional)

The uniform density is by no means the only density that can be used in Monte Carlo integration.

If the density of X is $f(x)$, then

$$E[g(X)/f(X)] = \int [g(x)/f(x)]f(x)dx = \int g(x)dx$$

so we can approximate the latter by sample averages of $g(X)/f(X)$.

Example (Optional)

To approximate the integral $\int_1^\infty \exp(-x^2)dx$, write it as

$$\int_0^\infty \exp[-(x+1)^2]dx,$$

and use an exponential distribution for X:

```
X <- rexp(100000)
mean( exp( -(X + 1)^2 ) / dexp(X) )

## [1] 0.139
```

The true value of this integral is 0.1394.

Caution

Monte Carlo integration is not always successful: sometimes the ratio $g(X)/f(X)$ varies so much that the sample mean doesn't converge.

Try to choose $f(x)$ so this ratio is roughly constant, and avoid situations where $g(x)/f(x)$ can be arbitrarily large.