Ve406 Lecture 3

Jing Liu

UM-SJTU Joint Institute

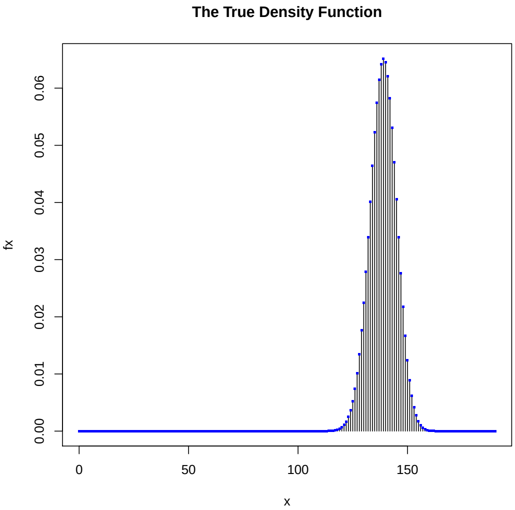
May 21, 2018

Q: Can you understand what the following piece of R code is about?

```
> rm(list = ls())
> # Simulation study on CI, estimation, and testing
> num = 1e3        # number of repetition
> n = 190          # number of trial
> # Generate a true parameter use uniform(0,1)
> p = runif(1)
> p
```

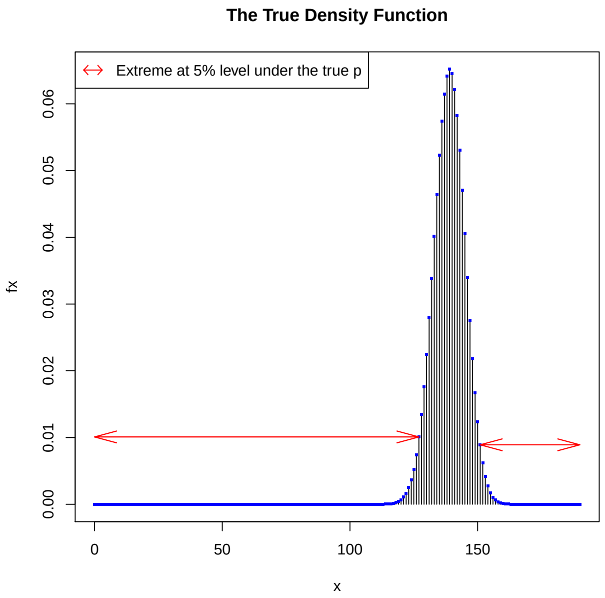
```
[1] 0.730913
```

```
> x = 0:190
> fx = dbinom(x, size = n, prob = p)
>
> plot(x, fx, type = "h",
+      main = "The True Density Function")
>
> points(x, fx, pch = ".", col = "blue", cex = 3)
```



- Let us indicate unusual events under the true p on the graph.

```
> # Indicate unlikely events at 5% level under true
> x_unlikely_lower =
+   qbinom(0.025, size = n, prob = p)

> yL = dbinom(x_unlikely_lower,
+              size = n, prob = p)

> arrows(x0 = 0, y0 = yL,
+        x1 = x_unlikely_lower, y1 = yL,
+        angle = 15, col = 2, code = 3, lwd = 1.2)

> x_unlikely_upper =
+   qbinom(0.025, size = n, prob = p,
+          lower.tail = FALSE)
> yU = dbinom(x_unlikely_upper, size = n, prob = p)
> arrows(x0 = x_unlikely_upper, y0 = yU,
+        x1 = n, y1 = yU,
+        angle = 15, col = 2, code = 3, lwd = 1.2)
```

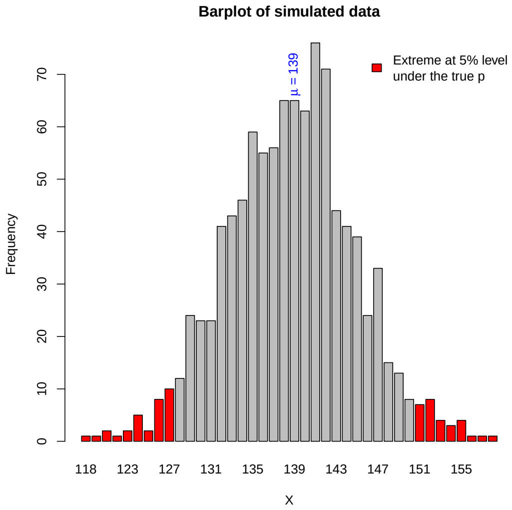


The True Density Function
Extreme at 5% level under the true p
fx
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0
50
100
150
x

```
> legend("topleft", lwd = NA, lty = NA, legend =
+          "Extreme at 5% level under the true p",
+          x.intersp = 0)

> par(font = 5) #change font to get arrows

> legend("topleft", legend = NA,
+          lwd = 1, lty = NA,
+          pch = 171,
+          col = 2,
+          bty = "n",
+          pt.cex=1.3)

> par(font = 1) #change back to common font
```

Q: A confidence interval is a type of interval estimate that may fail with a given probability, what exactly is this probability? What is actually to be repeated?

```
> X = rbinom(num, size = n, prob = p)
```



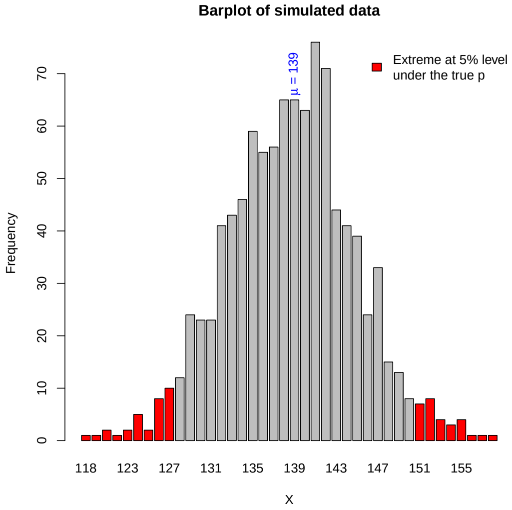
```r
> # Sorting the data into a table of counts
> counts = table(X)

> nonzero.counts = names(counts)
> head(nonzero.counts)
```

```
[1] "118" "120" "121" "122" "123" "124"
```

```r
> tmp = as.character(x_unlikely_lower)
> xL.index = which(nonzero.counts == tmp)

> tmp = as.character(x_unlikely_upper)
> xU.index = which(nonzero.counts == tmp)

> tmp = length(counts)
> index.vec = rep(1, tmp)

> index.vec[1:xL.index] = 2
> index.vec[xU.index:tmp] = 2
```

```r
> cols.vec = c("grey", "red")[index.vec]

> x.mean = round(n*p)
> xm.index = which(nonzero.counts == x.mean)

> barpos =
+   barplot(counts, col = cols.vec,
+           xlab = "X", ylab = "Frequency",
+           main = "Barplot of simulated data")

> text(barpos[xm.index],
+      counts[[xm.index]] + 5,
+      bquote(mu~"="~.(x.mean)),
+      col = 4, srt = 90)

> legend("topright", legend =
+          "Extreme at 5% level\nunder the true p",
+        fill = 2, bty = "n")
```

Q: What do you think will happen to the C.I. based on those x in the red bars?



```
> res = binom.test(X[1], n = n, p = p)
>
> res; p
```

```

        Exact binomial test

data:  X[1] and n
number of successes = 124, number of trials = 190, p-value = 0.01733
alternative hypothesis: true probability of success is not equal to 0.7315841
95 percent confidence interval:
 0.5803205 0.7200957
sample estimates:
probability of success
             0.6526316

[1] 0.7315841
```

```
> names(res)
```

```
[1] "statistic"   "parameter"   "p.value"
[4] "conf.int"    "estimate"    "null.value"
[7] "alternative" "method"      "data.name"
```

```
> res$p.value
```

```
[1] 0.01732913
```

```
> res$conf.int[1:2]
```

```
[1] 0.5803205 0.7200957
```

```
> res.df = data.frame(
+   CIL = double(), CIU = double())

> for (i in 1:num){
+   res = binom.test(X[i], n = n, p = p)
+   res.df[i,1:2] = res$conf.int[1:2]
+ }
```

```
> head(res.df, 2)
```

```
        CIL       CIU
1 0.5803205 0.7200957
2 0.6184566 0.7544657
```

```
> res.df[1,1] < p
```

```
[1] TRUE
```

```
> res.df[1,1] < p & res.df[1,2] > p
```

```
[1] FALSE
```

```
> n_ci_contain_true =
+   sum(res.df[, 1] < p & res.df[, 2] > p)
```

```
> (rate = 1 - n_ci_contain_true/num)
```

```
[1] 0.044
```

- What does the law of large numbers say?

Law of Large Numbers

Suppose that X_1, X_2, ..., X_n all have the same expected value

$$\mathbb{E}\left[X_i\right] = \mu,$$

the same variance

$$\mathrm{Var}\left[X_i\right] = \sigma^2,$$

and zero covariance with each other

$$\mathrm{Cov}\left[X_i, X_j\right] = 0 \qquad \text{where} \quad i \neq j$$

In particular, if the X_i are i.i.d., then the following holds

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \to \mu \qquad \text{when} \quad n \to \infty$$

- The law of large number is the justification for using the following earlier

$$\hat{\beta}_0 = \bar{Y}_n - \hat{\beta}_1 \bar{X}_n$$

- It is easy to understand intuitively, but it was often misunderstood.
- Consider the following pieces of R code.

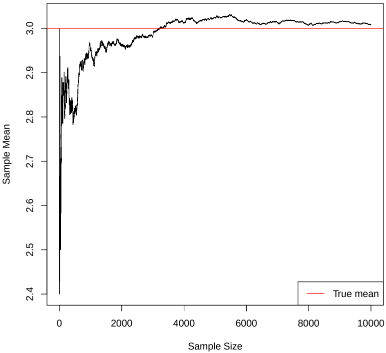
```
> # LLN ------------------------------------
> rm(list=ls())
> n = 1e4 # final sample size
> lambda = 3

> # Consider Poisson random variables
> xpois = rpois(n, lambda)
> xexp = lambda # True mean for Poisson
```

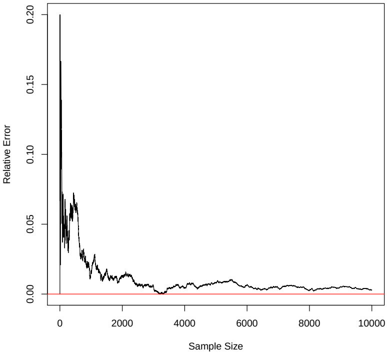
- Let us investigate $\bar{X}$ as n increases

```
> # sample size at various stages
> nseq = 1:n
```

```r
> # sample mean at various stage
> xcbar = cumsum(xpois) / nseq

> # Relative error
> error = abs(xcbar-xexp) / abs(xexp)

> plot(nseq, xcbar, type = "l",
+      xlab = "Sample Size", ylab = "Sample Mean")
>
> abline(h = xexp, col = 2)
>
> legend("bottomright", legend = "True mean",
+        lty = 1, col = 2)

> plot(nseq, error, type = "l",
+      xlab = "Sample Size", ylab = "Relative Error")
>
> abline(h = 0, col = 2)
```

Q: What do you expect to see?

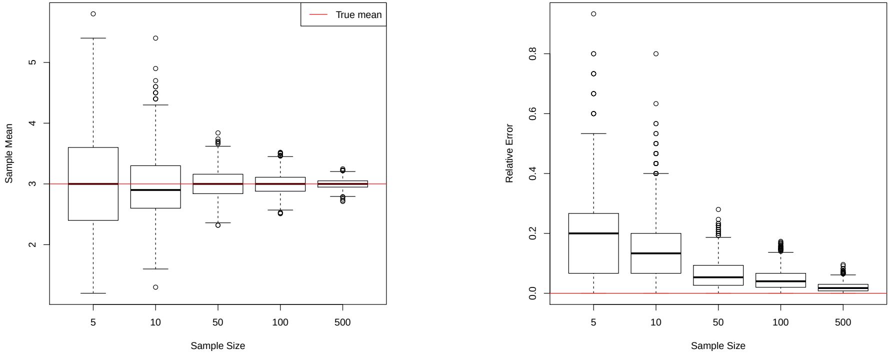




Q: Do you think it is due to not having a large enough sample size?

```
> # A better look at LLN -----------------------------
> sample.size.vec = c(5, 10, 50, 100, 500)
> ncases = length(sample.size.vec)              # 5 cases
> num = 1000                           # number of repetitons
> xbar.vec = double()      # x bar for all simulations
> error.vec = double()     # error for all simulations
> n.vec = integer()        # sample size for each case
```

```
> for(j in 1:ncases){
+   # Current sample size
+   n = sample.size.vec[j]
+   s.vec = double(num)  # all x bar for this n
+   e.vec = double(num)  # all error for this n
+   # Repeat num number of times
+   for (i in 1:num){
+       x = rpois(n, lambda)
+       s.vec[i] = sum(x) / n
+       e.vec[i] = abs(s.vec[i] - xexp) / abs(xexp)
+   }
+   # Store simulation results
+   xbar.vec = c(xbar.vec, s.vec)
+   error.vec = c(error.vec, e.vec)
+   n.vec = c(n.vec, rep(sample.size.vec[j], num))
+
+ }
```



- Note how we mange to reduce the spread of the distribution to 0 effectively.

$$\lim_{n\to\infty} \Pr\left(\left|\bar{X}_n - \mu\right| > \varepsilon\right) = 0 \qquad \text{for any} \quad \varepsilon > 0$$

```r
> x.df =
+   data.frame(xbar = xbar.vec,
+              error = error.vec, n = n.vec)
>
> boxplot(xbar~n, data = x.df,
+         xlab = "Sample Size",
+         ylab = "Sample Mean")
>
> abline(h = xexp, col = 2)
>
> legend("topright", legend = "True mean",
+        lty = 1, col = 2)
>
> boxplot(error~n, data = x.df,
+         xlab = "Sample Size",
+         ylab = "Relative Error")
>
> abline(h = 0, col = 2)
```

Q: What does the central limit theorem say?

Central Limit Theorem

Suppose X_1, X_2, ..., X_n are i.i.d. with mean μ and variance σ^2, then

$$\sqrt{n}\frac{\bar{X}_n - \mu}{\sigma}$$

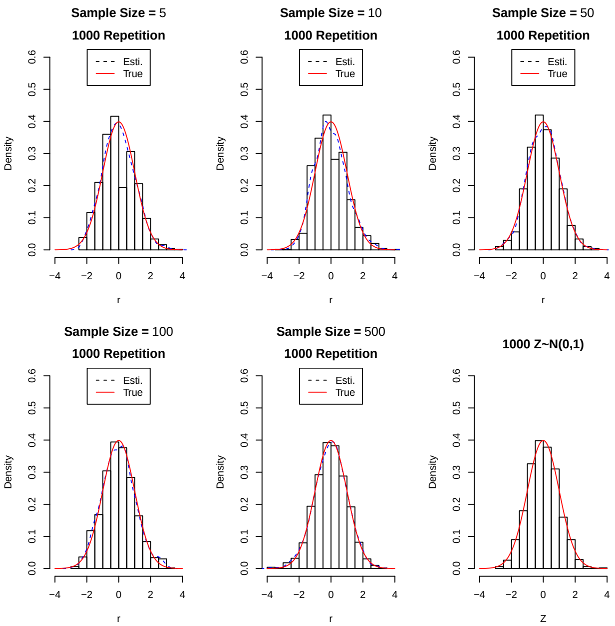
becomes more and more likely a standard normal random variable as $n \to \infty$

$$Z \sim \text{Normal}(0, 1)$$

in the sense that for every real number z

$$\lim_{n\to\infty} \Pr\left(\sqrt{n}\frac{\bar{X}_n - \mu}{\sigma} \leq z\right) = \Pr\left(Z \leq z\right)$$

- Intuitively, it means the sequence of distributions can be better and better described by a normal distribution as n becomes bigger and bigger.


Sample Size = 5
1000 Repetition
Esti.
True
Density
r
Sample Size = 10
1000 Repetition
Esti.
True
Density
r
Sample Size = 50
1000 Repetition
Esti.
True
Density
r
Sample Size = 100
1000 Repetition
Esti.
True
Density
r
Sample Size = 500
1000 Repetition
Esti.
True
Density
r
1000 Z~N(0,1)
Density
Z

```
> # CLT ----------------------------------------
> sample.size.vec
```

```
[1]   5  10  50 100 500
```

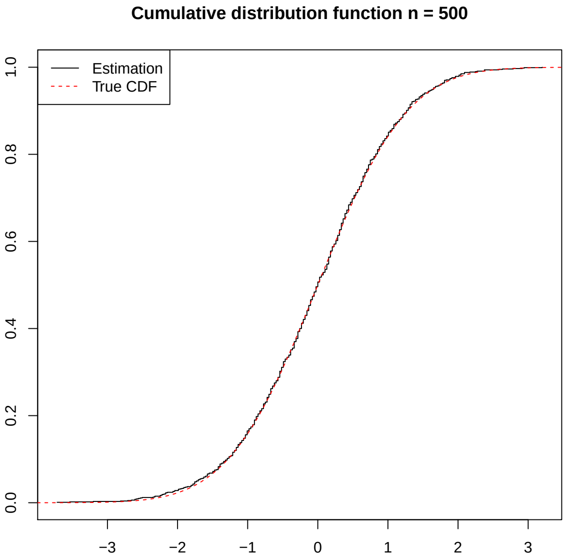
```
> lambda; xexp
```

```
[1] 3
[1] 3
```

```
> xvar = lambda # True variance for Poisson

> sqrt.n.vec = sqrt(n.vec) # last for loop

> top = xbar.vec - xexp
> bottom = sqrt(xvar)
> r.vec = sqrt.n.vec * top / bottom

> x.hist = seq(-4, 4, length.out = 100)
> par(mfrow = c(2,3))
```

```r
> for (eps in sample.size.vec){
+   ss = r.vec[n.vec == eps]   # subset by n
+
+   tname =                    # title
+     bquote(bold(atop(
+       "Sample Size ="~.(eps), "1000 Repetition")))
+
+   hist(ss, freq = FALSE, xlab = "r", main = tname,
+        ylim = c(0, 0.6), xlim = c(-4, 4))
+
+   # Kernel Density Estimation
+   lines(density(ss), col = 4, lty = 2)
+   # True Density function
+   lines(x.hist, dnorm(x.hist), col = 2, lty = 1)
+
+   legend("top", col = c(1,2), lty = c(2,1),
+          legend = c("Estimation","True "))
+ }
```

- We can ask R to use sample quantiles to estimate the distribution function.



```
> true.norm = rnorm(num, mean = 0, sd = 1)
>
> hist(true.norm, freq = FALSE,
+       ylim = c(0, 0.6), xlim = c(-4, 4),
+       xlab = "Z", main = "1000 Z~N(0,1)")
>
> lines(x.hist, dnorm(x.hist), col = 2, lty = 1)
>
> par(mfrow = c(1,1))

> sample_quantile = sort(r.vec[n.vec == 500])
> sample_cdf = (1:num) / num
>
> plot(sample_quantile, sample_cdf, type = "s",
+       xlab = "", ylab = "", main =
+         "Cumulative distribution function n = 500")
> lines(x.hist, pnorm(x.hist), col = 2, lty = 2)
> legend("topleft", lty = c(1, 2), col = c(1, 2),
+         legend = c("Estimation", "True CDF"))
```

Definition

An estimator is consistent if

$$\hat{\theta}_n \to \theta \qquad \text{when} \quad n \to \infty$$

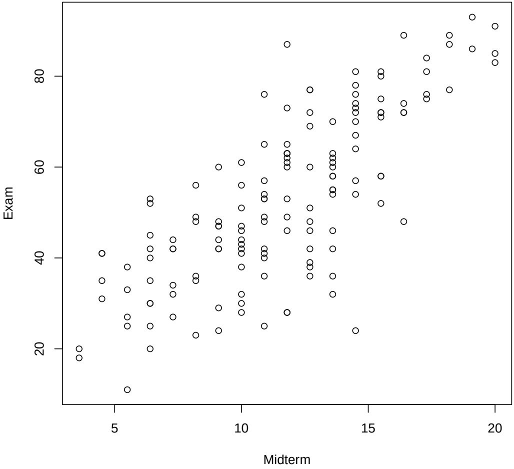
- The law of large numbers states $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n}$ is a consistent estimator of

  $$\mathbb{E}\left[X_i\right] \qquad \text{where } X_i \text{ are i.i.d.}$$

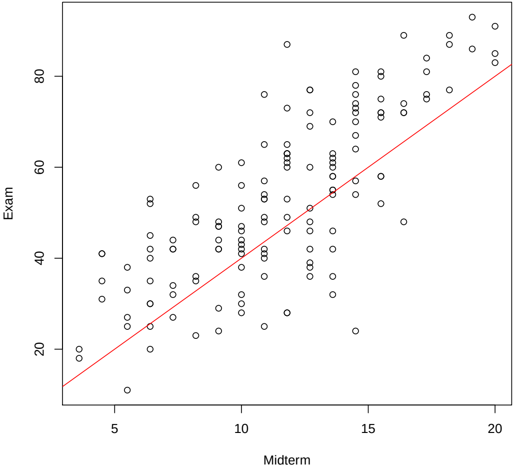
  as well as being an unbiased estimator of it.
- The central limit theorem states the sampling distribution is asymptotically normal with the variance $\frac{\sigma^2}{n}$, thus the standard error of $\hat{\theta} = \bar{X}$,

  $$\mathrm{SE} = \frac{\sigma}{\sqrt{n}} \to 0 \qquad \text{when} \quad n \to \infty$$

Q: What do you think the relationship between Midterm and Final Exam is?



- The straight line $y = 4x$ seems to be reasonable to my eye.

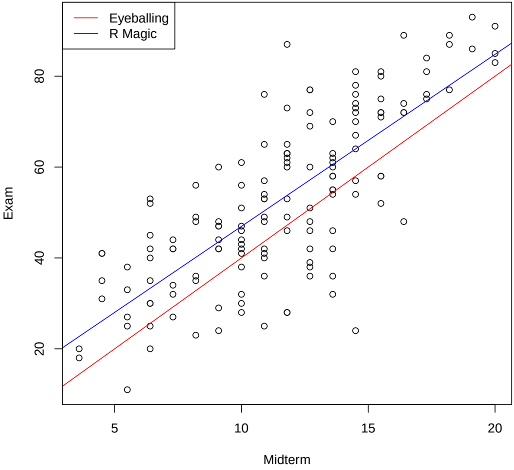


```
> # Load data locally
> course.df = read.table("~/Desktop/course.txt",
+                                 header = TRUE)
>
> plot(Exam~Midterm, data = course.df)
>
> abline(a = 0, b = 4, col = "red")
```

- Of course, R can "automatically decide" the best line for us

```
> course.lm = lm(Exam~Midterm, data = course.df)
>
> abline(course.lm, col = "blue")
>
> legend("topleft", legend =
+              c("Eyeballing", "R Magic"),
+           lty = 1, col = c(2, 4))
```

- Statistics is far more nature to out eyes than Calculus!



Q: Any idea what the followings are about?

```r
> summary(course.lm)
```

```
Call:
lm(formula = Exam ~ Midterm, data = course.df)

Residuals:
    Min      1Q  Median      3Q     Max
-39.980  -6.471   0.826   8.575  33.242

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)   9.0845     3.2204   2.821  0.00547 **
Midterm       3.7859     0.2647  14.301  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.05 on 144 degrees of freedom
Multiple R-squared:  0.5868,    Adjusted R-squared:  0.5839
F-statistic: 204.5 on 1 and 144 DF,  p-value: < 2.2e-16
```

```r
> predict(course.lm, data.frame(Midterm = 18))
```

```
       1
77.23109
```

Q: Under what conditions are the above outcomes valid?

- There are three important steps in statistical modelling:

1. Formulate a statistical model
2. Check the model assumptions
3. Inference and prediction

- Three common model assumptions for simple linear regression:

1. The conditional mean of the response is linear in terms of β_0, β_1, x_i

$$\mathbb{E}\left[Y_i \mid X_i = x_i\right] = \beta_0 + \beta_1 x_i$$

2. The errors have zero mean and constant variance

$$\mathbb{E}\left[\varepsilon_i \mid X_i\right] = 0 \quad \text{and} \quad \mathrm{Var}\left[\varepsilon_i \mid X_i\right] = \sigma^2 \qquad \text{where} \quad \varepsilon_i = Y_i - \beta_0 - \beta_1 X_i$$

3. The errors are uncorrelated with X_i, and uncorrelated with each other.

- Recall we said the following is natural choice

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \qquad \text{and} \qquad \hat{\beta}_1 = \frac{c_{xy}}{s_x^2}$$

where

$$\bar{x} = \frac{1}{n} \sum_i^n x_i; \qquad \bar{y} = \frac{1}{n} \sum_i^n y_i$$

and

$$s_x^2 = \frac{1}{n-1} \sum_i^n (x_i - \bar{x})^2; \qquad c_{xy} = \frac{1}{n-1} \sum_i^n (x_i - \bar{x})(y_i - \bar{y})$$

```
> x = course.df$Midterm; y = course.df$Exam
> beta_1_hat = cov(x,y ) / var(x)
> xbar = mean(x); ybar = mean(y)
> beta_0_hat = ybar - beta_1_hat * xbar
```

```
> beta_0_hat; beta_1_hat
```

```
[1] 9.084463
[1] 3.785924
```

```
> summary(course.lm)
```

```
Call:
lm(formula = Exam ~ Midterm, data = course.df)

Residuals:
    Min      1Q  Median      3Q     Max
-39.980  -6.471   0.826   8.575  33.242

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)   9.0845     3.2204   2.821  0.00547 **
Midterm       3.7859     0.2647  14.301  < 2e-16 ***
---
Signif. codes:  0 ‘***‘ 0.001 ‘**‘ 0.01 ‘*‘ 0.05 ‘.‘ 0.1 ‘ ‘ 1

Residual standard error: 12.05 on 144 degrees of freedom
Multiple R-squared:  0.5868,    Adjusted R-squared:  0.5839
F-statistic: 204.5 on 1 and 144 DF,  p-value: < 2.2e-16
```

Q: Do you think the estimator $\hat{\beta}_1 = \dfrac{c_{xy}}{s_x^2}$ is consistent?