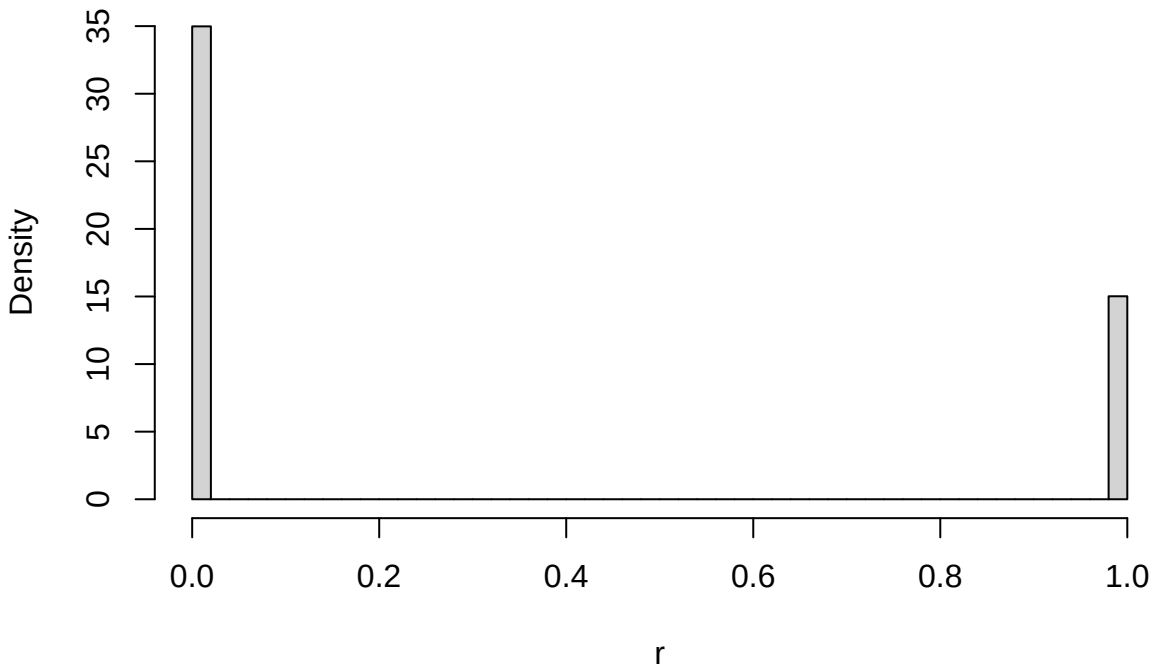
Lecture_3_MACSS

Auffhammer

2024-09-12

There was a question after last class, which suggested doubt about the fact that the proportion has a normal distribution. Three ways to show this. One is pages and pages of theory. Two is relying on the CLT (which does not make assumptions on the pdf of the underlying random variable, other than finite variance, which a binomial has). Three is firing up R and running a little Monte Carlo. Also, we could do this for a decent size probability of success and decent sample sizes. Then we could go with a setting, where the probability of success is small as is the smaple size (and things break down - hence the sample size condition).

```
rm(list = ls()) # clear memory

set.seed(22092008) # set random number generator seed

N<-1000000 # population size

n <- 1000 # sample size (for calculation of mean)
numloop <- 10000 # number of draws

# Placeholder
g <- integer(numloop) # vector to hold sample mean for each iteration

# # For normally distributed data
# sig2 <- 1 # variance for population
# mu <- 0 # mean of population
# r <- rnorm(N, mean=mu,sd=sqrt(sig2))

# For uniformly distributed data
# ub <- 10
# lb <- -10
# r <- runif(N, min = lb, max = ub)
# sig2 <- (lb-ub)^2/12 # variance for uniform (SCIENCE!)

# For a binomial!
 p <- 0.3
 r <- rbinom(n=N, size=1, prob=p)
 sig2 <- (p)*(1-p)
# lb <- -10
# r <- runif(N, min = lb, max = ub)
# sig2 <- (lb-ub)^2/12 # variance for uniform (SCIENCE!)


# Plot population
hist(r,prob=TRUE,breaks = 50,main = "Raw data")
```

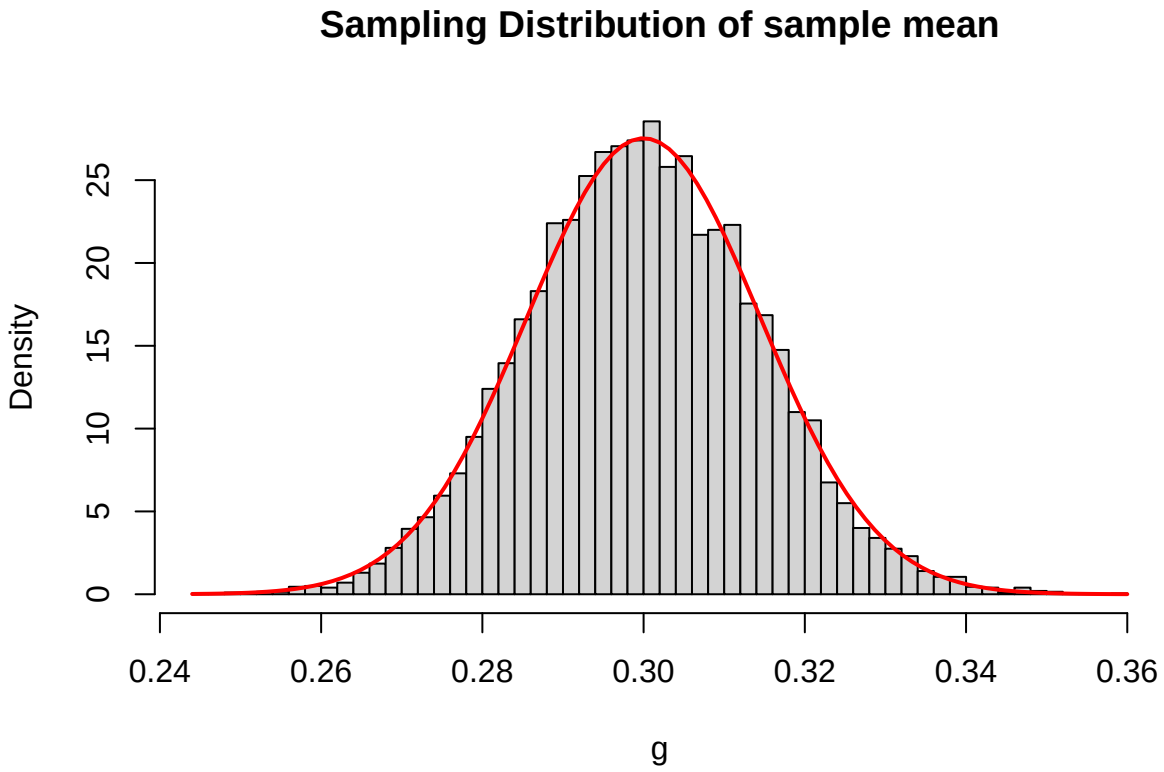



```
readline(prompt="Press [enter] to continue")
```

```
## Press [enter] to continue
```

```
## [1] ""
```

```
# Draw `numloop' samples of size n and calculate mean each time.

for(i in 1:numloop) {
  tmp <-sample(r, n, replace = FALSE, prob = NULL)
  g[i] <- mean(tmp)
}

# Calculate minimum and maximum of sample means (cleaner plot)
a <-min(g)
b <-max(g)


# Make histogram of sample means with 50 bins
hist(g,prob=TRUE,breaks = 50,main = "Sampling Distribution of sample mean")

# Overlay normal distribution with predicted mean and variance
curve(dnorm(x,mean=p,sd=sqrt(sig2/n)),a,b,add=TRUE,lwd=2,col="red")
```



Now let's get ot the second question about Confidence Intervals. Someone in class said `Max, your interpretation of what a confidence interval is, is not consistent with what I learned elsewhere.` Max said `We are 95% confident that the true consumption of participants lies between 23.78 and 28.23 kWh per day.` I think the issue was with the use of the word `confident`. This is standard usage found everywhere, but often gets confused with the word `probability`. This is a discussion in Statistics that is old and still a sore spot for many. But the best way to think about a confidence interval in your mind is what the asker of the question proposed. `If I calculated a CI 1000 times on different random samples from the population, 95% of the time, the Confidence Interval would contain the true parameter.`

```
rm(list = ls()) # clear memory
library("dplyr")
```

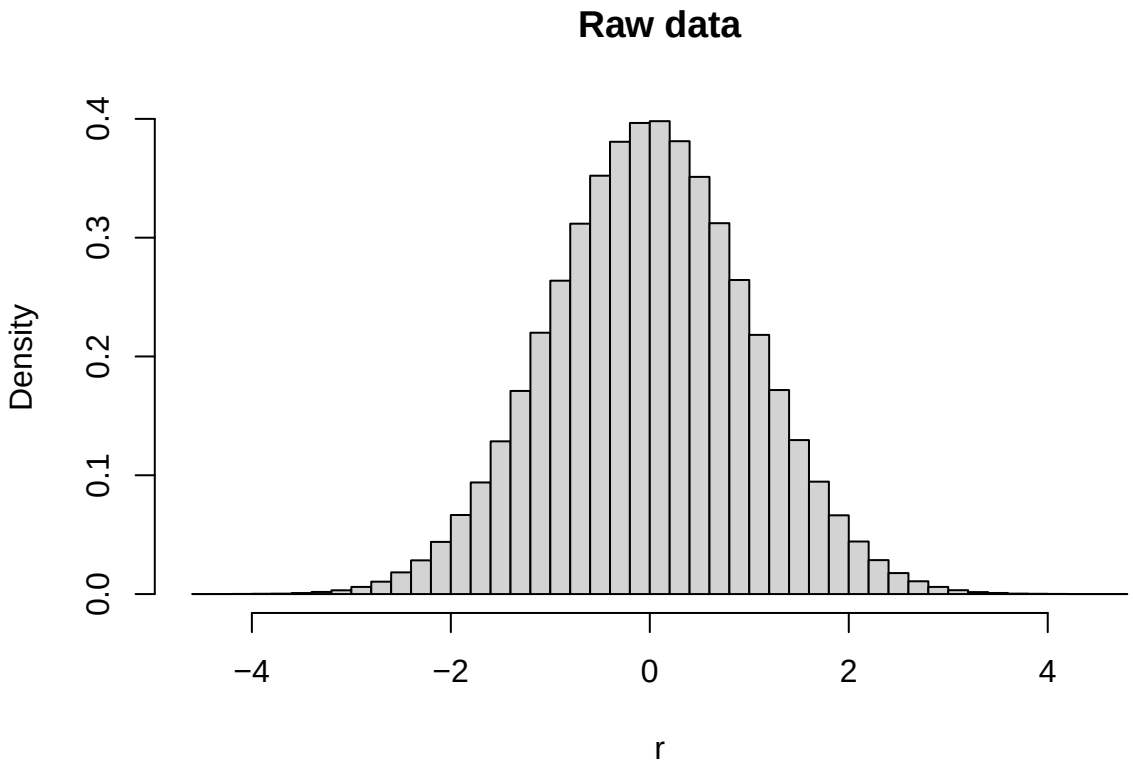
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
set.seed(22092008) # set random number generator seed

N<-1000000 # population size
n <- 1000 # sample size (for calculation of mean)
df = n-1

# We will first do this once. Then repeat it 1000 times.
numloop <- 10000 # number of draws
```

```r
# Placeholder
g <- integer(numloop)
gtmp <-integer(numloop)
se <-integer(numloop)
cil <-integer(numloop)
ciu <-integer(numloop)
cheese <-integer(numloop)
# Set Type 1 Error Probability
alpha <- 0.05
a2 <-0.5*alpha
# # Let's use normally distributed data
 sig2 <- 1 # variance for population
 mu <- 0 # mean of population (Max's birthday)
 r <- rnorm(N, mean=mu,sd=sqrt(sig2))



# Plot population
hist(r,prob=TRUE,breaks = 50,main = "Raw data")
```




```r
# tmp <-sample(r, n, replace = FALSE, prob = NULL)
# gtmp <- mean(tmp)
# se <-sqrt(var(tmp)/n)
# cil <- gtmp+qt(0.5*alpha,n-1)*se
# ciu <- gtmp-qt(0.5*alpha,n-1)*se
# cheese <- between(mu,cil,ciu)

# Now let's do this numloop times
for(i in 1:numloop) {
```

```
tmp <-sample(r, n, replace = TRUE, prob = NULL)
gtmp[i] <- mean(tmp)
se[i] <-sqrt(var(tmp)/n)
cil[i] <- gtmp[i]+qt(a2,df)*se[i]
ciu[i] <- gtmp[i]-qt(a2,df)*se[i]
cheese[i] <- mu<cil[i] | mu>ciu[i]
}
print(round(mean(cheese),2))
```

```
## [1] 0.05
```

Now let's get ambitious. We are going to generate some data on health status of some made up folks and play random sampling and random assignment.

```
rm(list = ls()) # clear memory
library("dplyr")
library('MASS')
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library('crosstable')
library('flextable')
set.seed(22092008) # set random number generator seed
N <- 10000 #Population Size
n <- 1000 # Sample Size
mu <- c(0, 0,0)
# a <- 0.5 #Gender Income Covariance
# b <- 0.1 #Gender Insurance Covariance
# c <- 0.8 #Income insurance

# If insurance were randomized
a <- 0.5 # Set to 0.5 as default
b <- 0.0 # Set to 0.1 as default
c <- 0.0 # Set to 0.8 as default

# Some betas for later
b1 <-1 #Gender Beta
b2 <-5 # Income Beta
b3 <-3 #Insurance
shifter <- 30
Sigma <- matrix(c(1, a, b, a, 1, c,b, c, 1), nrow=3)
data = mvrnorm(N, mu, Sigma, empirical=FALSE)
Gender = data[, 1]  # standard normal (mu=0, sd=1)
Income = data[, 2]  # standard normal (mu=0, sd=1)
Insurance= data[, 3]  # standard normal (mu=0, sd=1)

# Gender and income should be binary
Ins <- Insurance>0
Gend <- Gender>0
cor(Ins,Gend)
```

```
## [1] -0.003263571
```

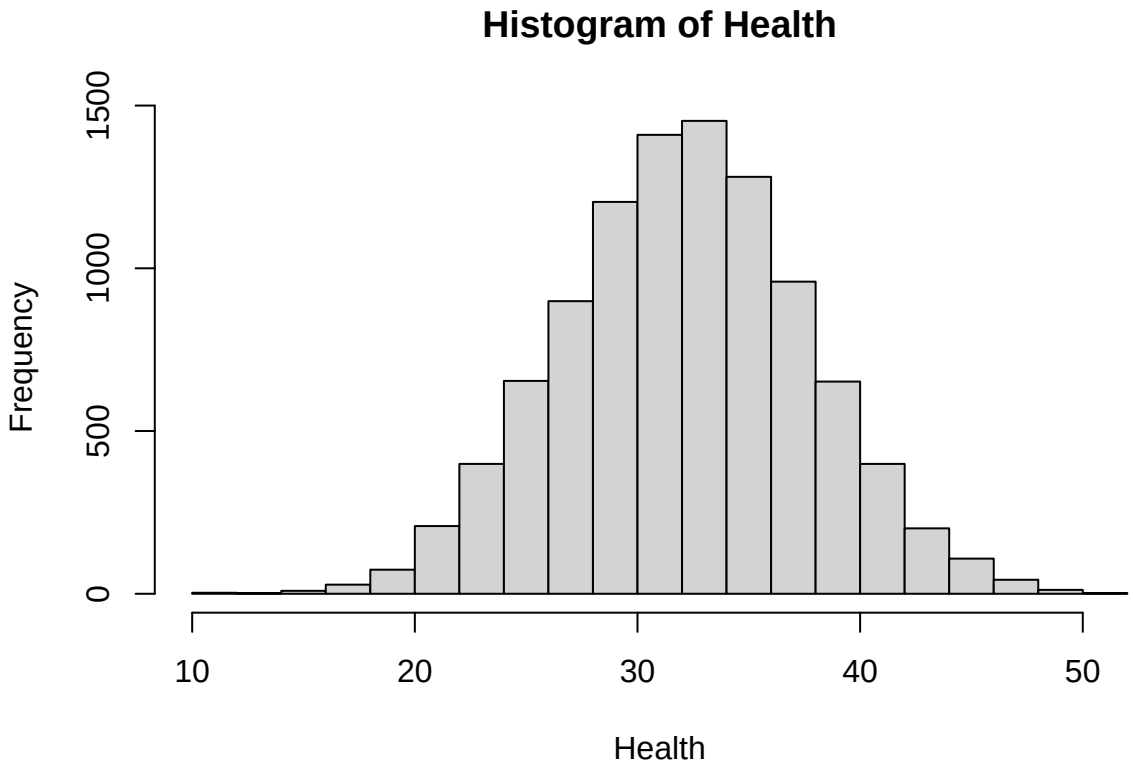
```
cor(Ins,Income)
```

```
## [1] -0.007736086
```

```
cor(Gend,Income)
```

```
## [1] 0.4073727
```

```
# We are going to generate some arbitrary Health Index
Health <- shifter +  rnorm(N,mean=0,sd=1) + b1*Gend + b2*Income + b3* Ins
# Plot my Outcome
hist(Health)
```




```
# Let's make a nice table comparing groups across insurance
mydata <- data.frame(Income, Gend, Health, Ins)
ft1 <- crosstable(mydata,by="Ins", test=TRUE, funs=c(mean=mean, "std error"=sd)) %>%
  as_flextable()
```

```
## Warning in crosstable(mydata, by = "Ins", test = TRUE, funs = c(mean = mean, : Be aware that automati
## context, as it would cause extensive alpha inflation otherwise.
## This warning is displayed once every 8 hours.
```

```
print (ft1)
```

```
## a flextable object.
## col_keys: `label`, `variable`, `FALSE`, `TRUE`, `test`
## header has 2 row(s)
## body has 6 row(s)
## original dataset sample:
##      .id  label  variable         FALSE          TRUE
## 1 Income Income      mean          0.03          0.01
```

```
## 2 Income Income std error           1.0           1.0
## 3   Gend   Gend     FALSE 2452 (49.40%) 2512 (50.60%)
## 4   Gend   Gend      TRUE 2504 (49.72%) 2532 (50.28%)
## 5 Health Health      mean          30.6          33.6
##                                                 test
## 1          p value: 0.4392 \n(Two Sample t-test)
## 2          p value: 0.4392 \n(Two Sample t-test)
## 3 p value: 0.7442 \n(Pearson's Chi-squared test)
## 4 p value: 0.7442 \n(Pearson's Chi-squared test)
## 5    p value: <0.0001 \n(Wilcoxon rank sum test)
```

We will play with this more next time.