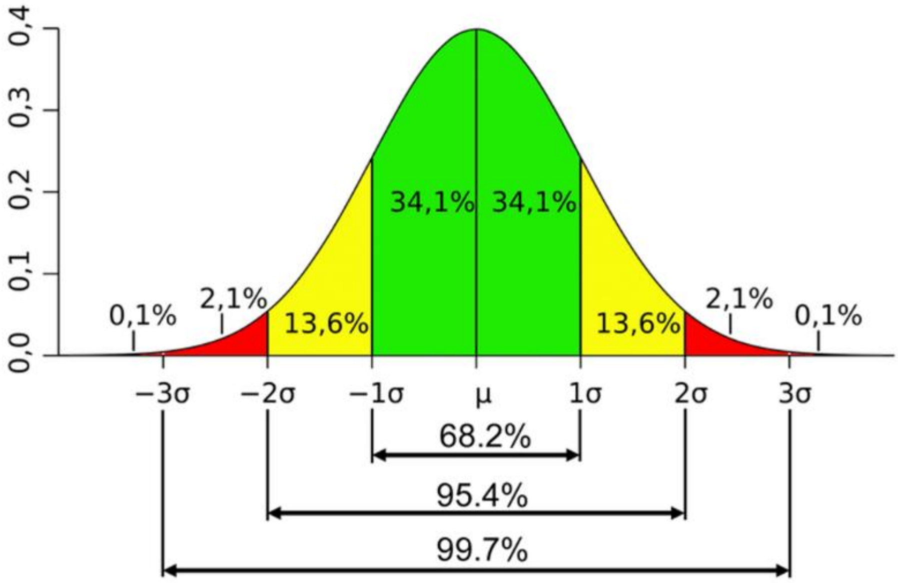
Session 9: Hypothesis testing

9.1. The concept of normal distribution

a. What is a Normal Distribution?

- **Shape:** The normal distribution looks like a bell-shaped curve.
- **Symmetry:** It is perfectly symmetrical around the center.



b. Key Characteristics:

- **Mean (Average):** The center of the curve.
- **Standard Deviation:** Measures the spread of the data.
  - 68.2% of the data falls within 1 standard deviation of the mean.
  - 95.4% falls within 2 standard deviations.
  - 99.7% falls within 3 standard deviations.

c. Why is it Important?

- **Natural Occurrences:** Many natural phenomena follow this distribution (e.g., heights, test scores). For example, most students score around the average in a class, fewer scoring very high or very low.
- **Central Limit Theorem:** In large samples, the samples' mean tend to be normally distributed. (Video)
- **Statistical Inferences:** Helps in making predictions and decisions based on data.

9.2. Hypothesis testing

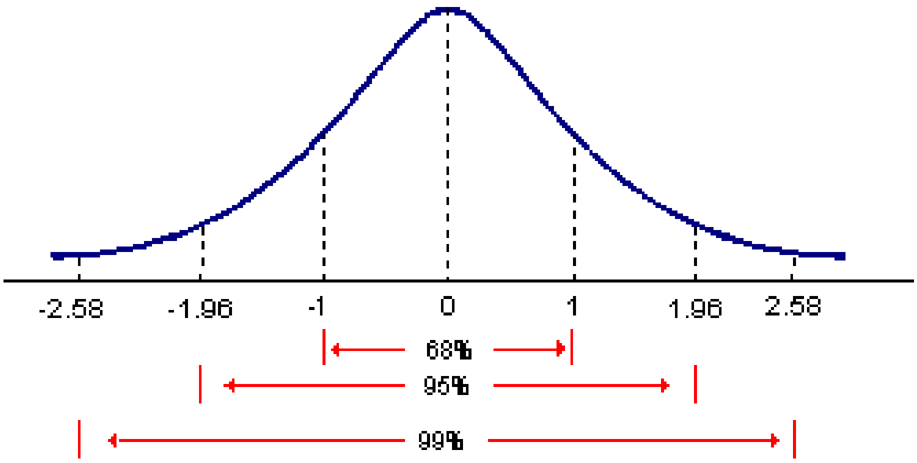
a. What is Hypothesis Testing?

- Hypothesis testing is a method used to decide whether there is enough evidence to support a particular claim about a population based on a sample of data.
- **Null Hypothesis (H_0):** This is the default statement that there is no effect or no difference. It assumes that any observed differences are due to random chance. Example: "The average age is equal to 20."

- **Alternative Hypothesis** (H_1): This is what you want to prove, stating there is an effect or a difference.
  Example: "The average age is not equal to 20."

b. Procedure of hypothesis testing

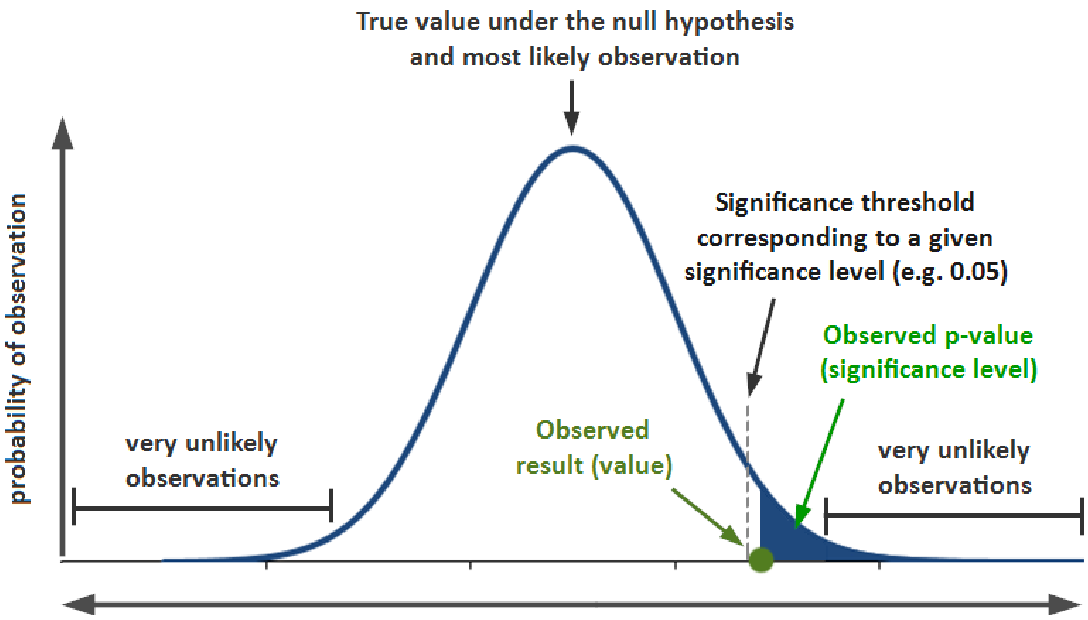
- State the null and alternative hypothesis. (e.g. $H_0: \mu = 0, H_1: \mu \neq 0$)
- Collect sample data.
- Calculate sample mean and stadard error ($\frac{s}{\sqrt{n}}$).
- Calculate t-statistics ($t = \frac{\bar{X}-\mu}{Standar\ Error}$).
- Compare absolute value of t-statistics |t| with critical values for given level of significance (α). [1.65 (10% significance level), 1.96 (5%), 2.58 (1%)]



- Decision: reject null hypothesis if |t| exceeds critical value, otherwise fail to reject null hypothesis.

c. Hypothesis testing with p-value

- p-value : probability (area under normal distribution) beyond |t|.



- **Decision :** reject null hypothesis if p-value is lower than the significance level, otherwise fail to reject null hypothesis.
- Easier to conduct hypothesis testing with p-value. No need to calculate t-statistics and remember different critical values.

9.3. Hypothesis testing in Stata

```
* Clear existing data
clear

* Create a dummy dataset
set seed 12345
set obs 100
gen group = mod(_n, 2)
gen score = 50 + group * 10 + rnormal(0, 10)

*conducting hypothesis testing
ttest score = 50 //H0: pop_mean = 50
ttest score = 55 //H0: pop_mean = 55
ttest score = 60 //H0: pop_mean = 60

* conducting two-sample t-test
ttest score, by(group) //H0: pop_mean_group1 = pop_mean_group2
                       //OR H0: pop_mean_group1 - pop_mean_group2 = 0

*Same answer can be obtained from regression
reg score group
```

Exercise:
Using NMICS6 data (hl.sav), conduct a hypothesis test whether average age between male and female is statistically different.

```
import spss "https://gitlab.com/misc.a/referenced/-/raw/main/NMICS6/hl.sav",
clear

* HL6 -> Age, HL4 -> Sex
sum HL6 if HL4 == 1 //male : average age is 28.263
sum HL6 if HL4 == 2 //female : average age is 28.827

*Looks like the population means for male and female are not statistically
different.
*Let's conduct the hypothesis testing

ttest HL6, by(HL4)
*Alternatively

reg HL6 HL4
```

9.4. Hypothesis testing using non-parametric approach (bootstraping)

Bootstrap : generating distribution of statistics of interest by resampling the sample with replacement. Using Bootstrap, we can calculate standard errors, confidence intervals, and other statistical measures.

```
clear
set seed 1
set obs 100
gen score = round(runiform() * 100)

* Bootstrap the median and test against a specified value (e.g., 50)
bootstrap r(p50), reps(1000): summarize score, detail

* Testing whether median is equal to 50 or not
test _bs_1 = 50
```

Exercise:

Using NMICS6 data (hl.sav), conduct a hypothesis test whether medeian age between male and female is statistically different.

```
import spss "https://gitlab.com/misc.a/referenced/-/raw/main/NMICS6/hl.sav",
clear

set seed 12345
* Define a program to calculate the difference in medians
program define diff_medians, rclass
    summarize HL6 if HL4 == 1, detail
    local med0 = r(p50)
    summarize HL6 if HL4 == 2, detail
    local med1 = r(p50)
    return scalar diff = `med1' - `med0'
end

* Bootstrap the difference in medians
bootstrap r(diff), reps(100): diff_medians
```

Session 10: Regression analysis

10.1. Simple regression analysis

```
clear

set seed 12345
set obs 100
gen study_hours = round(runiform() * 10)
gen score = 50 + 5 * study_hours + rnormal(0, 5)

reg score study_hours
```

```
. reg score study_hours

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(1, 98)        =    811.79
       Model |  23506.7755         1  23506.7755   Prob > F        =    0.0000
    Residual |  2837.77267        98   28.956864   R-squared       =    0.8923
-------------+----------------------------------   Adj R-squared   =    0.8912
       Total |  26344.5482        99  266.106547   Root MSE        =    5.3812

------------------------------------------------------------------------------
       score | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
 study_hours |   4.962436   .1741703    28.49   0.000     4.616801    5.308071
       _cons |   49.85214   1.005975    49.56   0.000     47.85581    51.84846
------------------------------------------------------------------------------
```

10.2. Multiple regression and diagnostics

```
clear
set obs 200

gen age = mod(_n,52) + 18
gen educ_year = mod(_n,18)

* Generate Income variable with a positive relationship with Age and Education
gen income = 20000 + 800 * age + 3000 * educ_year + rnormal(0, 2000)

* Regression with omitted variable
reg income age

*******************************
* Residual diagnostics
```

```
*******************************
* Residual visual inspection
rvfplot

* Histogram plot for residual's distribution visualization
predict resid, residuals
hist resid

*Formal test of residuals normality
swilk resid
drop resid

* Multiple regression with correct specification
reg income age educ_year

*******************************
* residual diagnostics
*******************************
* Residual visual inspection
rvfplot

* Histogram plot for residual's distribution visualization
predict resid, residuals
hist resid

*Formal test of residuals normality
swilk resid
```

Session 11: Advance regression with binary dependent variables (logit/probit)

```
import spss "https://gitlab.com/misc.a/referenced/-/raw/main/NMICS6/hh.sav",
clear

* dropping missing values
drop if missing(HHSEX)

* checking levels of HHSEX (Household Head Sex)
codebook HHSEX
label list labels410

gen hh_size = HH48 //HH member size variable
gen urb_rur = HH6 //1=Urban 2=Rural
gen province = HH7 //province number

* generating binary dependent variable separately
gen hhsex_male = 1
replace hhsex_male = 0 if HHSEX == 2 //1=Male 2=Female

*running logistic regression
logit hhsex_male hh_size ib1.urb_rur ib3.province
margins, dydx(hh_size urb_rur province)

* Similar results can be obtained using probit
* Running probit regression
probit hhsex_male hh_size ib1.urb_rur ib3.province
margins, dydx(hh_size urb_rur province)
```

Session 12: Time series analysis

12.1. Stationarity concept

- Stationarity refers to a time series whose statistical properties, such as mean, variance, and autocorrelation, remain constant over time.
- Non-stationary series are prone to spurious relationships.

12.2. Spurious relationship

```
clear
set seed 1
set obs 100

gen year = 1900 + _n
tsset year
gen ice_cream_sales = year*10 + rnormal(0, 50)
gen shark_attacks = year*5 + rnormal(0, 20)

* visual inspection for stationarity
twoway line ice_cream_sales year, name(ice_cream_sales, replace)
twoway line shark_attacks year, name(shark_attacks, replace)

dfuller ice_cream_sales //H0 : Non-stationary
dfuller shark_attacks //H0 : Non-stationary

* Run the initial regression (spurious relationship)
reg shark_attacks ice_cream_sales
```

12.3. Making series stationary to avoid spurious relationship

```
*****************************
* Making Series Stationary
*****************************
* Differencing variable makes series stationary
* If a variable is stationary at first difference, then its called
* I(1). I(0) means the variable is stationary at level.
twoway line D.ice_cream_sales year, name(ice_cream_sales, replace)
twoway line D.shark_attacks year, name(shark_attacks, replace)

dfuller D.ice_cream_sales //H0 : Non-stationary
dfuller D.shark_attacks //H0 : Non-stationary

*no relationship observed after differencing
reg D.shark_attacks D.ice_cream_sales

** log difference is preferred over simple difference as
** interpretation of coefficient becomes easier.
gen lshark_attacks = log(shark_attacks)
gen lice_cream_sales = log(ice_cream_sales)

twoway line D.lice_cream_sales year, name(ice_cream_sales, replace)
twoway line D.lshark_attacks year, name(shark_attacks, replace)

dfuller D.lice_cream_sales //H0 : Non-stationary
dfuller D.lshark_attacks //H0 : Non-stationary

reg D.lshark_attacks D.lice_cream_sales
```

12.4. Example of non-stationary series with actual relationship

```
clear
set seed 1
set obs 100

gen year = 1900 + _n
tsset year
gen income = year*10 + rnormal(0, 50)
gen expenditure = income*0.5 + rnormal(0, 20)

* visual inspection for stationarity
twoway line income year, name(income, replace)
```

```
twoway line expenditure year, name(expenditure, replace)

dfuller income //H0 : Non-stationary
dfuller expenditure //H0 : Non-stationary

* Run the initial regression
reg expenditure income

*****************************
* Making Series Stationary
*****************************
gen lincome = log(income)
gen lexpenditure = log(expenditure)

* visual inspection for stationarity
twoway line D.lincome year, name(income, replace)
twoway line D.lexpenditure year, name(expenditure, replace)

dfuller D.lincome //H0 : Non-stationary
dfuller D.lexpenditure //H0 : Non-stationary

* Run the regression at first difference
reg D.lexpenditure D.lincome
```