A. Understanding variable types (numeric, string, etc.)

- Numeric variables

  Numeric variables store numbers and can be used for mathematical operations.

  Types:

  - **Integer:** Whole numbers without decimal points.
  - **Float:** Numbers with decimal points. These are approximate representations of real numbers.

```
* Clear any existing data in memory
clear
* Set seed for reproducibility
set seed 12345

* Create a dataset with 100 observations
set obs 100

* Generate an integer variable (age)
gen int age = round(runiform() * 52 + 18)

* Generate a float variable (income)
gen float income = runiform() * 80000 + 20000
```

- String Variables

  String variables store text and are used for non-numeric data.

```
* Generate a short string variable (name)
gen str18 name = ""

* Assign values to the string variable
replace name = "John Doe" in 1
replace name = "Jane Smith" in 2
```

- Factor/Categorical variables

  Categorical variables take on a limited number of distinct values, representing different categories.

```
* Generate a categorical variable (education)
* if not explicitly defined, by default numerical variables take float
type.
gen education = 1
replace education = 2 in 21/40
replace education = 3 in 41/60
replace education = 4 in 61/80

* Label the categorical variable
label define edu_labels 1 "High School" 2 "Bachelor's" 3 "Master's" 4
"PhD"

label list edu_labels

label values education edu_labels
```

Exercise:

Generate a factor/categorical variable named age_group based on the following rule.
age < 20 → Teen, 20 <= age < 65 → Adult, age >=65 → Senior.

```
* Generate a categorical Age Group variable
gen age_group = .
replace age_group = 1 if age < 20
replace age_group = 2 if age >= 20 & age < 65
replace age_group = 3 if age >= 65

* Label the Age Group variable
label define agegrp_labels 1 "Teen" 2 "Adult" 3 "Senior"
label values age_group agegrp_labels
```

- Date and Time variable
  Date and time variables store dates, times, and date-time combinations. They require special formats to perform calculations and manipulations.

```
* Generate a date variable
gen date = mdy(12, 25, 2024)

* Format the date variable
format date %td

* Format the date in YYYY-MM-DD format
format date %tdCCYY-NN-DD

* Format the date in MM/DD/YYYY format
format date %tdNN/DD/CCYY
```

B. Converting variable types using destring and tostring

Converting a string variable to numeric

```
* Clear any existing data in memory
clear
set obs 3

* Create a string variable with numeric values
gen str_var = "123"
replace str_var = "456" in 2
replace str_var = "789" in 3

* Convert the string variable to numeric
destring str_var, replace //destring str_var, gen(num_var)
```

Converting a numeric variable to string

```
* Clear any existing data in memory
clear
set obs 3

* Create a numeric variable
gen num_var = 123
replace num_var = 456 in 2
replace num_var = 789 in 3

* Convert the numeric variable to string
tostring num_var, replace //tostring num_var, gen(str_var)
```

C. Encoding a string variable to a factor/categorical variable

```
* Clear any existing data in memory
clear

* set dataset size to 4 observations
set obs 40

* Create a string variable with categorical values
gen education_level = "PhD"
replace education_level = "Bachelor's" in 11/20
replace education_level = "High School" in 21/30
replace education_level = "Master's" in 31/40

* Define a label with a specific order
label define edu_labels 1 "High School" 2 "Bachelor's" 3 "Master's" 4
"PhD"
```

```
* Encode the string variable into a numeric variable using the defined
label
encode education_level, gen(education_encoded) label(edu_labels)

* Encode the string variable without defined labels
encode education_level, gen(education_encoded1)
```

D. Re-ordering variables

```
* Load the built-in dataset 'auto'
sysuse auto, clear

* Original order of variables
describe

* Reorder variables
order price mpg rep78 make headroom

* Verify the new order of variables
describe

* move a variable to the first position
order mpg

* Verify the new order of variables
describe

* move a variable to the first position
order mpg, last

* Verify the new order of variables
describe
```