In this practice assignment we'll be preparing our data "census_income.csv" in order to build a prediction model for outcome Y present in the data. We'll be building a logistic regression model for this binary outcome. Don't worry about model building process , we'll get into that once we have covered required modules. In this assignment we'll be preparing our data.

Dummy Variables

Note : Go through linear regression material first to do this.

Biggest requirement of any data prep process is to deal with categorical variables. As we have discussed earlier in brief we'll be making dummy variables out of categorical variables.

Find out number of categories in each categorical variable. Your output should look like this:

```
## [1] "Number of categories in  workclass  : 9"
## [1] "Number of categories in  education  : 16"
## [1] "Number of categories in  marital.status  : 7"
## [1] "Number of categories in  occupation  : 15"
## [1] "Number of categories in  relationship  : 6"
## [1] "Number of categories in  race  : 5"
## [1] "Number of categories in  sex  : 2"
## [1] "Number of categories in  native.country  : 42"
```

Select variables which have 6 or less number of categories and make (n-1) dummy variables for them. We are giving an example code for creating dummy variables for variable race.

```
table(d$race)
```

```
##
##  Amer-Indian-Eskimo  Asian-Pac-Islander               Black
##                 311                1039                3124
##               Other               White
##                 271               27816
```

```
d=d%>%
  mutate(race_AIE=as.numeric(race==" Amer-Indian-Eskimo"),
         race_API=as.numeric(race==" Asican-Pac-Islander"),
         race_Black=as.numeric(race==" Black"),
         race_White=as.numeric(race==" White")) %>%
  select(-race)
# we ignored the category which had least frequency
```

Create dummy variables for rest of the variables in a similar fashion [having 6 or less categories : sex , relationship].

Combining Similar Categories

There were many categorical variables which had many more distinct categories and making `n-1` dummy variables might not be a very good approach. What we can do instead is to bring down number of categories by combining similar categories. We saw this in one of the earlier practice assignment we'll build on top of that.

Here is example code for creating dummy variables with grouping categories for variable workclass.

Note: Grouping is done on the basis of similar behaviour across classes of target [which is Y in this case]

```
round(prop.table(table(d$workclass,d$Y),1),1)
```

```
##
##                        <=50K  >50K
##    ?                     0.9   0.1
##    Federal-gov           0.6   0.4
##    Local-gov             0.7   0.3
##    Never-worked          1.0   0.0
##    Private               0.8   0.2
##    Self-emp-inc          0.4   0.6
##    Self-emp-not-inc      0.7   0.3
##    State-gov             0.7   0.3
##    Without-pay           1.0   0.0
```

```
# you can take any category [ after grouping ] as base [the one to ignore]
d=d %>%
  mutate(wc_1=as.numeric(workclass==" Self-emp-inc"),
         wc_2=as.numeric(workclass==" Federal-gov"),
         wc_3=as.numeric(workclass %in% c(" Local-gov"," Self-emp-not-inc"," State-gov")),
         wc_4=as.numeric(workclass==" Private"),
         wc_5=as.numeric(workclass==" ?")) %>%
  select(-workclass)
```

You can use similar methods to create dummy variables for variables [education , marital.status, occupation , native.country].Create dummy variables for rest of the categorical variables .

Flag variables

Although we should generaly leave numeric variables as is [discussion on monotonic transformations will happen in regression modules], but sometimes there is particular value which occurs too many times in a variable, in such cases you should make flag variables which take value 0,1 in accordance when that variable takes that particular values. Here is an example of creating flag variable for variable capital.gain:

```
# this will give % of observations where capital.gain is 0
sum(d$capital.gain==0)/nrow(d)
```

```
## [1] 0.9167102
```

More than 90% values are 0 , lets go ahead create a flag variable for this

```
d=d %>%
  mutate(cg_flag0=as.numeric(capital.gain==0))
```

In the same manner check for what % of data capital.loss is zero and create a flag variable if this comes out to be high

Converting the target

For running logistic regression your target needs to take values 0,1. Right now Y takes two categorical values . convert them so that Y takes value when it is " >50K" and 0 otherwise

Save this code for data prep that you have written . We'll be using this prepared data in our exercise in logistic regression module.