Similar Codes

This document contains code that may help you perform the data cleaning steps for Assignment 2. We will use data available in R for the first example, and we will make our own data frame for the second example.

US Rent Income

Let's print the first 8 observations of the `us_rent_income` dataframe to see the data we will work with.

```
pacman::p_load(tidyverse, dplyr)
head(us_rent_income, 8)
```

```
## # A tibble: 8 x 5
##   GEOID NAME     variable estimate   moe
##   <chr> <chr>    <chr>       <dbl> <dbl>
## 1 01    Alabama  income      24476   136
## 2 01    Alabama  rent          747     3
## 3 02    Alaska   income      32940   508
## 4 02    Alaska   rent         1200    13
## 5 04    Arizona  income      27517   148
## 6 04    Arizona  rent          972     4
## 7 05    Arkansas income      23789   165
## 8 05    Arkansas rent          709     5
```

We will create a new dataframe called `us_wide` that

- uses `select()` to drop the `moe` variable
- uses `pivot_wider()` to reshape the data from long to wide

```
us_wide <- us_rent_income %>%
  select(-c(moe)) %>%
  pivot_wider(names_from = variable ,
              values_from = estimate)

# print first 8 rows of the us_wide data frame
head(us_wide, 8)
```

```
## # A tibble: 8 x 4
##   GEOID NAME        income  rent
##   <chr> <chr>        <dbl> <dbl>
## 1 01    Alabama      24476   747
## 2 02    Alaska       32940  1200
## 3 04    Arizona      27517   972
## 4 05    Arkansas     23789   709
```

```
## 5 06    California   29454  1358
## 6 08    Colorado     32401  1125
## 7 09    Connecticut  35326  1123
## 8 10    Delaware     31560  1076
```

Let's do the following transformation in the `us_wide` data frame.

- use `rename()` function to rename `NAME` to `name`
- use `mutate()` and nested `ifelse()` functions to create a variable called `income_cat` that takes the value of
  - 1 if income <= 26000
  - 2 if 26000 < income <= 32000
  - 3 if income > 32000
  - NA otherwise

```
us_wide <- us_wide %>%
  rename(state = NAME) %>%
  mutate(incomecat = ifelse(income <= 26000, 1,
                            ifelse(income > 26000 & income <= 32000, 2,
                                   ifelse(income > 32000, 3, NA))))
```

Finally, we create a new dataframe called `select_states` that uses `filter()` and `%in%` operator to filter observations for California, Oregon, and Washington states only.

```
select_states <- us_wide %>%
  filter(state %in% c("California", "Oregon", "Washington"))

table(select_states$state)
```

```
##
## California     Oregon Washington
##          1          1          1
```

Working with Dates

I will create two vectors containing dates with different formats.

```
customer <- c("customer1", "customer2", "customer3")
dates1 <- c("2020-06-23", "2020-05-15", "2021-01-25")
dates2 <- c("06/23/2020", "05/15/2020", "01/25/2021")
dates <- as.data.frame(cbind(customer, dates1, dates2))
glimpse(dates)
```

```
## Rows: 3
## Columns: 3
## $ customer <chr> "customer1", "customer2", "customer3"
## $ dates1   <chr> "2020-06-23", "2020-05-15", "2021-01-25"
## $ dates2   <chr> "06/23/2020", "05/15/2020", "01/25/2021"
```

You can see from the data that both `dates1` and `dates2` were encoded as character (chr) by R. We use the `mutate()` and `as.Date()` functions to convert these 2 variables to a date format. Take note of the format of the year, month, and date.

```
dates <- dates %>%
  mutate(dates1 = as.Date(dates1, format = c("%Y-%m-%d")),
         dates2 = as.Date(dates2, format = c("%m/%d/%Y")))

glimpse(dates)
```

```
## Rows: 3
## Columns: 3
## $ customer <chr> "customer1", "customer2", "customer3"
## $ dates1   <date> 2020-06-23, 2020-05-15, 2021-01-25
## $ dates2   <date> 2020-06-23, 2020-05-15, 2021-01-25
```

We can use `slice()` to drop the first row.

```
dates %>%
  slice(-1)
```

```
##    customer     dates1     dates2
## 1 customer2 2020-05-15 2020-05-15
## 2 customer3 2021-01-25 2021-01-25
```