Session 4: R Practice 1

Wali Reheman

2024-09-24

```
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
```

```
library(readr)
library(haven)
```

```
data <- haven::read_dta("nyc_schools.dta")
# Source: New York City Department of Education records, assembled by Nathan Favero
```

Renaming Variables

```
head(data)
```

```
## # A tibble: 6 x 19
##   dbn    schoolname          schooltype overallscore overallgrade percentilerank
##   <chr>  <chr>               <chr>             <dbl> <chr>                 <dbl>
## 1 01M015 P.S. 015 Roberto C~ Elementary         39   C                        15
## 2 01M019 P.S. 019 Asher Levy Elementary         55.9 B                        56
## 3 01M020 P.S. 020 Anna Silv~ Elementary         40.2 C                        17
## 4 01M034 P.S. 034 Franklin ~ K-8                67.5 A                        83
## 5 01M063 P.S. 063 William M~ Elementary         59.3 B                        63
## 6 01M064 P.S. 064 Robert Si~ Elementary         48.9 C                        37
## # i 13 more variables: progressgrade <chr>, performancegrade <chr>,
## #   environmentgrade <chr>, closingtheachievementgappoints <dbl>,
## #   principal <chr>, enrollment <dbl>, district <dbl>, iep <chr>,
## #   economicneedindex <dbl>, blackhispanic <chr>, ell <chr>, thgrmathela <chr>,
## #   peerindex <dbl>
```

```
# Rename a variable
data <- data %>% rename(school = schoolname)

# See if changes made
head(data)
```

```
## # A tibble: 6 x 19
##   dbn   school schooltype overallscore overallgrade percentilerank progressgrade
##   <chr> <chr>  <chr>             <dbl> <chr>                 <dbl> <chr>
## 1 01M0~ P.S. ~ Elementary         39   C                        15 F
## 2 01M0~ P.S. ~ Elementary         55.9 B                        56 B
## 3 01M0~ P.S. ~ Elementary         40.2 C                        17 D
## 4 01M0~ P.S. ~ K-8                67.5 A                        83 B
## 5 01M0~ P.S. ~ Elementary         59.3 B                        63 B
## 6 01M0~ P.S. ~ Elementary         48.9 C                        37 C
## # i 12 more variables: performancegrade <chr>, environmentgrade <chr>,
## #   closingtheachievementgappoints <dbl>, principal <chr>, enrollment <dbl>,
## #   district <dbl>, iep <chr>, economicneedindex <dbl>, blackhispanic <chr>,
## #   ell <chr>, thgrmathela <chr>, peerindex <dbl>
```

Summary Statistics for Subsets

Method 1

```
summary(data$overallscore[data$schooltype == "Elementary"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   12.70   44.83   53.80   54.54   64.70  102.20       8
```

```
summary(data$overallscore[data$schooltype == "Middle"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   13.40   45.90   55.30   55.94   65.47  102.20      25
```

```
summary(data$overallscore[data$schooltype == "K-8"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   26.20   44.23   53.95   54.98   64.28   91.90       1
```

```
#Guess what will following codes generate

data$schooltype == "Elementary"
```

```
##    [1]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
##   [13] FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE
##   [25] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##   [37]  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
##   [49] FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE
##   [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
##   [73] FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE
##   [85]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [97] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
##  [109]  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE
##  [121] FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
##  [133]  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE
##  [145]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
##  [157]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
##  [169]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [181]  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
##  [193]  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
##  [205]  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE
##  [217] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
##  [229]  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##  [241] FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [253] FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [265]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##  [277]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE
##  [301] FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
##  [313]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##  [325] FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
##  [337] FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE
##  [349]  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
##  [361]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
##  [373]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE
##  [385] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [397] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [409]  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE
##  [421] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [433] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [445]  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE
##  [457]  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [469]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
##  [481] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE
##  [493] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##  [505]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
##  [517]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [529] FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
##  [541] FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
##  [553]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE
##  [565]  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE
##  [577] FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
##  [589] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
##  [601]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
##  [613]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE
##  [625]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [637]  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE
##  [649]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
##  [661] FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
##  [673]  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE
##  [685]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE
##  [697] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
##  [709] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##  [721] FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [733] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE
##  [745] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [757] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
##  [769] FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
##  [781] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
##  [793]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [805]  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE
##  [817]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
##  [829]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
##  [841]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
##  [853] FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [865]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
##  [877]  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
##  [889] FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
##  [901]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [913]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
##  [925] FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [937]  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [949] FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [961] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
##  [973]  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [985]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE
##  [997] FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
## [1009]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
## [1021]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [1033]  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [1045]  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE
## [1057]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
## [1069] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1081] FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
## [1093]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
## [1105] FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
## [1117] FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
## [1129]  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## [1141] FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
## [1153]  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## [1165]  TRUE  TRUE
```

Breaking Down the Command:

1. **`data$overallscore`:** This selects the `overallscore` column from the `data` dataframe.
2. **`data$schooltype == "Elementary"`:** This creates a logical vector (TRUE/FALSE) that is `TRUE` for rows where the `schooltype` column equals "Elementary" and `FALSE` otherwise.
3. **`data$overallscore[data$schooltype == "Elementary"]`:** This uses the logical vector to subset `overallscore`, selecting only those values where `schooltype` is "Elementary."
4. **`summary()`:** The `summary()` function then computes summary statistics (such as the minimum, 1st quartile, median, mean, 3rd quartile, and maximum) for the selected subset of `overallscore`.

Method 2

```
data%>%
  filter(schooltype=="Elementary")%>%
  select(overallscore)%>%
  summary()
```

```
##   overallscore
##  Min.   : 12.70
##  1st Qu.: 44.83
##  Median : 53.80
##  Mean   : 54.54
##  3rd Qu.: 64.70
##  Max.   :102.20
##  NA’s   :8
```

```
data%>%
  filter(schooltype=="Middle")%>%
  .$overallscore%>% # see the break-downs below
  summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA’s
##   13.40   45.90   55.30   55.94   65.47  102.20      25
```

Breaking Down the Command:

1. **`data %>%`:** Starts a pipeline where the `data` dataframe is passed into the next function.
2. **`filter(schooltype == "Middle")`:** Filters the `data` dataframe to include only rows where the `schooltype`column is equal to “Middle.” The result is a smaller dataframe containing only middle schools.
3. **`.$overallscore`:** Extracts the `overallscore` column from the filtered dataframe. The `.` refers to the dataframe that results from the previous step in the pipeline.
4. **`summary()`:** Applies the `summary()` function to the extracted `overallscore` column, generating summary statistics such as the minimum, 1st quartile, median, mean, 3rd quartile, and maximum for middle schools.

Rescaling Variables

```
head(data)
```

```
## # A tibble: 6 x 19
##   dbn   school schooltype overallscore overallgrade percentilerank progressgrade
##   <chr> <chr>  <chr>             <dbl> <chr>                 <dbl> <chr>
## 1 01M0~ P.S. ~ Elementary         39   C                        15 F
## 2 01M0~ P.S. ~ Elementary         55.9 B                        56 B
## 3 01M0~ P.S. ~ Elementary         40.2 C                        17 D
## 4 01M0~ P.S. ~ K-8                67.5 A                        83 B
## 5 01M0~ P.S. ~ Elementary         59.3 B                        63 B
## 6 01M0~ P.S. ~ Elementary         48.9 C                        37 C
## # i 12 more variables: performancegrade <chr>, environmentgrade <chr>,
## #   closingtheachievementgappoints <dbl>, principal <chr>, enrollment <dbl>,
## #   district <dbl>, iep <chr>, economicneedindex <dbl>, blackhispanic <chr>,
## #   ell <chr>, thgrmathela <chr>, peerindex <dbl>
```

```
# Rescale the overall score to range from 0 to 1
data <- data %>%
  mutate(overallscore = overallscore / 100)

summary(data$overallscore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.1270  0.4497  0.5415  0.5505  0.6490  1.0220      34
```

Creating Dummy Variables for Grades

```
table(data$overallgrade)
```

```
##
##       A   B   C   D   F
##  34 287 399 348  76  22
```

```
# Create dummy variables for letter grades
data <- data %>% mutate(
  gradeA = ifelse(overallgrade == "A", 1, ifelse(overallgrade == "", NA, 0)),
  gradeB = ifelse(overallgrade == "B", 1, ifelse(overallgrade == "", NA, 0)),
  gradeC = ifelse(overallgrade == "C", 1, ifelse(overallgrade == "", NA, 0)),
  gradeD = ifelse(overallgrade == "D", 1, ifelse(overallgrade == "", NA, 0)),
  gradeF = ifelse(overallgrade == "F", 1, ifelse(overallgrade == "", NA, 0)),
  grade_NA = ifelse(overallgrade == "", 1, 0)
)

head(data)
```

```
## # A tibble: 6 x 25
##   dbn   school schooltype overallscore overallgrade percentilerank progressgrade
##   <chr> <chr>  <chr>             <dbl> <chr>                 <dbl> <chr>
## 1 01M0~ P.S. ~ Elementary        0.39  C                        15 F
## 2 01M0~ P.S. ~ Elementary        0.559 B                        56 B
## 3 01M0~ P.S. ~ Elementary        0.402 C                        17 D
## 4 01M0~ P.S. ~ K-8               0.675 A                        83 B
## 5 01M0~ P.S. ~ Elementary        0.593 B                        63 B
## 6 01M0~ P.S. ~ Elementary        0.489 C                        37 C
## # i 18 more variables: performancegrade <chr>, environmentgrade <chr>,
## #   closingtheachievementgappoints <dbl>, principal <chr>, enrollment <dbl>,
## #   district <dbl>, iep <chr>, economicneedindex <dbl>, blackhispanic <chr>,
## #   ell <chr>, thgrmathela <chr>, peerindex <dbl>, gradeA <dbl>, gradeB <dbl>,
## #   gradeC <dbl>, gradeD <dbl>, gradeF <dbl>, grade_NA <dbl>
```

```r
table(data$gradeA,data$overallgrade)
```

```
##
##           A   B   C   D   F
##   0   0   0 399 348  76  22
##   1   0 287   0   0   0   0
```

Breaking Down the Command:

- **`gradeA = ifelse(overallgrade == "A", 1, ifelse(overallgrade == "", NA, 0)):`**
  - **`ifelse(overallgrade == "A", 1, ifelse(overallgrade == "", NA, 0)):`**
    * Checks if `overallgrade` is "A". If true, assigns `1` to `gradeA`.
    * If `overallgrade` is missing (empty string), assigns `NA` to `gradeA`.
    * If neither condition is true, assigns `0` to `gradeA`.
  - **This pattern is repeated for `gradeB`, `gradeC`, `gradeD`, and `gradeF`,** where the check is for "B", "C", "D", and "F", respectively.
- **`grade_missing = ifelse(overallgrade == "", 1, 0):`**
  - This creates a binary indicator that assigns `1` if `overallgrade` is missing (empty string) and `0` otherwise.

Creating an Index Variable

Now, let's create an index of the progress grade and the performance grade. We first convert the grades to numeric variables.

We assign a score of 4 to schools with an A, 3 for a B, etc.

```r
# Creating an index of progress grade and performance grade
data <- data %>% mutate(
  progress = case_when(
    progressgrade == "A" ~ 4,
    progressgrade == "B" ~ 3,
    progressgrade == "C" ~ 2,
    progressgrade == "D" ~ 1,
    progressgrade == "F" ~ 0,
    TRUE ~ NA_real_
  ),
  performance = case_when(
    performancegrade == "A" ~ 4,
    performancegrade == "B" ~ 3,
    performancegrade == "C" ~ 2,
    performancegrade == "D" ~ 1,
    performancegrade == "F" ~ 0,
    TRUE ~ NA_real_
  ),
  index = (progress + performance)/2
)

head(data)
```

```
## # A tibble: 6 x 28
##   dbn   school schooltype overallscore overallgrade percentilerank progressgrade
##   <chr> <chr>  <chr>             <dbl> <chr>                 <dbl> <chr>
## 1 01M0~ P.S. ~ Elementary        0.39  C                        15 F
## 2 01M0~ P.S. ~ Elementary        0.559 B                        56 B
## 3 01M0~ P.S. ~ Elementary        0.402 C                        17 D
## 4 01M0~ P.S. ~ K-8               0.675 A                        83 B
## 5 01M0~ P.S. ~ Elementary        0.593 B                        63 B
## 6 01M0~ P.S. ~ Elementary        0.489 C                        37 C
## # i 21 more variables: performancegrade <chr>, environmentgrade <chr>,
## #   closingtheachievementgappoints <dbl>, principal <chr>, enrollment <dbl>,
## #   district <dbl>, iep <chr>, economicneedindex <dbl>, blackhispanic <chr>,
## #   ell <chr>, thgrmathela <chr>, peerindex <dbl>, gradeA <dbl>, gradeB <dbl>,
## #   gradeC <dbl>, gradeD <dbl>, gradeF <dbl>, grade_NA <dbl>, progress <dbl>,
## #   performance <dbl>, index <dbl>
```

```
# Note: NA + 10 = ?

table(data$progressgrade,data$progress)
```

```
##
##       0   1   2   3   4
##       0   0   0   0   0
##   A   0   0   0   0 185
##   B   0   0   0 316   0
##   C   0   0 367   0   0
##   D   0 162   0   0   0
##   F 102   0   0   0   0
```

```
table(data$performancegrade,data$performance)
```

```
##
##       0   1   2   3   4
##       0   0   0   0   0
##   A   0   0   0   0 439
##   B   0   0   0 302   0
##   C   0   0 247   0   0
##   D   0  95   0   0   0
##   F  49   0   0   0   0
```

NA and NA_real_

`NA`

- **Type:** Generic missing value.
- **Behavior:** When used in expressions, `NA` can adapt to the expected type of the output (integer, numeric, character, etc.). For example, if you're working with a numeric vector and use `NA`, it will automatically be treated as `NA_real_`.
- **Usage:** `NA` is flexible and can be used in various contexts, including vectors of different types.

`NA_real_`

- **Type:** Specifically a missing value of type `double` (real numbers).
- **Behavior:** Explicitly indicates that the missing value is numeric and of type `double`. This is important when you need to ensure that the data type remains consistent, especially in functions like `mutate()` where type consistency is crucial.
- **Usage:** Typically used in numeric calculations or when creating variables where the type must be explicitly `double`.

Why Use `NA_real_`?

- **Type Consistency:** By explicitly using `NA_real_`, you ensure that the `elementary` variable is always treated as a numeric vector. If you used `NA` instead, R would still work correctly in this case, but using `NA_real_` makes the intent clear and avoids potential issues if the type needs to be consistent, especially in more complex operations.
- **Prevents Implicit Type Conversion:** If the context changes or if additional types are introduced, `NA_real_` ensures that R does not implicitly convert the vector to another type, which might happen with `NA`.

Handling String Variables

Convert Percentage Variables to Numeric

```r
# We want to know how many black or hispanic students those schools have
summary(data$blackhispanic)
```

```
##    Length     Class      Mode
##      1166 character character
```

```r
#We need to convert this variable to just numbers.
# Convert blackhispanic and ell variables to numeric

data <- data %>% mutate(
  blackhispanic = as.numeric(gsub("%", "", blackhispanic))
)

data <- data %>% mutate(
  blackhispanic = gsub("%", "", blackhispanic),
  blackhispanic = as.numeric(blackhispanic)
)

summary(data$blackhispanic)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3.60   52.52   92.35   74.99   97.40  100.00
```

Breaking Down the Command:

`mutate(blackhispanic = as.numeric(gsub("%", "", blackhispanic))):`

- **`gsub("%", "", blackhispanic):`** This function removes the percentage signs (`%`) from the `blackhispanic`variable. The `gsub` function replaces each occurrence of `%` with an empty string (`""`).
- **`as.numeric(...):`** Converts the cleaned `blackhispanic` values (now just numbers in string form) into numeric data.
- **`mutate(...):`** Creates a new version of the `blackhispanic` variable within the `data` dataframe, replacing the original values with the cleaned numeric values.

Extracting Substrings

The variable `dbn` contains the district, borough, and school number. The first 2 digits are the district number. The third digit is the borough. And the fourth through sixth digits are the school number.

```
# Extract district, borough, and school number from the dbn variable
data <- data %>% mutate(
  distnum = substr(dbn, 1, 2),
  borough = substr(dbn, 3, 3),
  schoolnum = substr(dbn, 4, 6)
)

substr("abcdef", 2, 5)
```

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

Save the data

```
# Save the cleaned data
saveRDS(data, "nyc_schools_cleaned.RDS")
```