Assignment1

Tian Walker

2024-10-25

The purpose of this assignment is to get you working with the `recipes` package and preprocessing the variables in two different datasets. You will use the same datasets with processed variables to build models in the next assignments.

There are alternative ways to submit your assignment depending on your preference.

1. You can Copy/Edit this notebook and complete it with your responses. Then, you can save and run the completed Kaggle notebook and submit the link through Canvas. If you keep your notebook Private, do not forget to share it with 'UOCOEEDS' so I can access it.
2. You copy/paste the questions and download the datasets from the notebook to your computer. Then, complete the assignment as an R markdown document. Then, you can knit the R Markdown document to a PDF and submit both the .Rmd and PDF files by uploading them on Canvas.
3. You knit the R Markdown document to an HTML document and host it on your website/blog or any publicly available platform. Then, you can submit the .Rmd file by uploading it on Canvas and putting the link for the HTML document as a comment.
4. If you have a GitHub repo and store all your work for this class in a GitHub repo, you can create a folder for this assignment in that repo and put the.Rmd file and PDF document under a specific folder. Then, you can submit the link for the GitHub repo on Canvas.

To receive full credit, you must complete the following tasks. Please make sure that all the R code you wrote for completing these tasks and any associated output are explicitly printed in your submitted document. If the task asks you to submit the data files you created, please upload these datasets along with your submission.

If you have any questions, please do not hesitate to reach out to me.

Task 1: Preprocessing Text Data

Description

For this part of the assignment, you will work with a Twitter dataset which is randomly sampled from a larger dataset on the Kaggle platform (see this link for the original data). In this subset data, there are 1,500 tweets and three variables. A description of the three variables in the dataset follows:

- **sentiment**: a character string variable with two values (Positive and Negative) for the outcome variable to predict.
- **time**: a character string variable indicating time of a tweet (e.g.,Thu Jun 18 07:35:01 PDT 2009)
- **tweet**: a character string variable that provides the full text of a tweet.

This subset data is available as an input data in this R notebook (**'../input/tweets/tweet_sub.csv'**).

Our ultimate goal is to build a model to predict whether or not a tweet has a positive sentiment by using the information from time of the tweet and text of the tweet. We will do this in the following assignments. For this assignment, we will only engineer features to use them later for building our models and prepare the dataset for model development.

Please complete the following tasks. Provide the R code you wrote and any associated output for each task.

Tasks

Task 1.1 Import the tweet data into the R environment. You can give any name to this data object. Print the structure of this data object using the `str` function.

Task 1.1 Import the tweet data into the R environment. You can give any name to this data object. Print the structure of this data object using the `str` function.

```r
data <- rio::import(here("assignment_1/data/tweet_sub.csv"))
data2 <- data
```

```r
str(data)
```

```
## 'data.frame':    1500 obs. of  3 variables:
##  $ sentiment: chr  "Negative" "Positive" "Positive" "Positive" ...
##  $ time     : chr  "Thu Jun 18 07:35:01 PDT 2009" "Sun May 10 00:31:52 PDT 2009" "Sun May 31 09:15:1
##  $ tweet    : chr  "I think my twitter is attackd by a kind of worm" "@ddlovato demi if you can i thi
```

Task 1.2 The `time` variable in this dataset is a character string such as *Thu Jun 18 07:35:01 PDT 2009*. Create four new columns in the dataset using this time variable to show the day, date, month, and hour of a tweet. The table below provides some examples of how these four new columns would look like given time as a character string.

time	day	month	date	hour
Thu Jun 18 07:35:01 PDT 2009	4	Jun	18	7
Sun May 10 00:31:52 PDT 2009	7	May	10	0
Sun May 31 09:15:19 PDT 2009	7	May	31	9
Fri May 22 07:25:52 PDT 2009	5	May	22	7
Sun May 31 02:09:52 PDT 2009	7	May	31	2
Sun Jun 07 09:13:08 PDT 2009	7	Jun	7	9

Make sure that `day` column is a numeric variable from 1 to 7 (Monday = 1, Sunday =7), `date` column is a numeric variable from 1 to 31, and `hour` column is a numeric variable from 0 to 23, and `month` column is a factor variable.

Calculate and print the frequencies for each new column (day, month, date, and hour) you created from the original `time`.

DATES

```r
data2 <- data
data2$wkday <- gsub( " .*$", "", data2$time )

data2$month <- word(data2$time, 2)
data2$month <- as.factor(data2$month)

data2$date  <- word(data2$time, 3) #daynum
data2$date <- as.numeric(data2$date)

data2$hms  <- word(data2$time, 4)
data2$hour <- substr(data2$hms, start = 1, stop = 2)
data2$hour <- as.numeric(data2$hour)
#data2$hms <-lubridate::hms(data2$hms) this does a cool thing where it pulls the numbers and labels the
#data2$hms <- word(data2$hms, 1) this pulls mostly just the hour, but in cases where hour is 0 the abov
```

```
#data2$hms <- gsub("[^[:digit:]]", "", data2$hms) This separates the letter part of the variable
```

Frequency of posts on days of the week:

```
##    Monday   Tuesday Wednesday  Thursday    Friday  Saturday    Sunday 
##     "20%"     "10%"      "6%"      "8%"     "15%"     "20%"     "21%" 
```

Descriptives for Date variable

```
##    vars    n  mean  sd median trimmed   mad min max range skew kurtosis   se
## X1    1 1500 15.05 9.8     16   14.82 13.34   1  31    30 0.11    -1.21 0.25
```

```
data3 <- data2
#changing variables from numberes to characters so that the frequencies can be calculated based on each

data3$date <- as.character(data3$date)
data3$hour <- as.character(data3$hour)

library(gtsummary)
library(flextable)
```

```
## 
## Attaching package: 'flextable'

## The following object is masked from 'package:gtsummary':
## 
##     continuous_summary
```

```
table1 <- data3 |>
select(day, month, date, hour ) |>
  tbl_summary()

table1 |>
  as_flex_table()
```

Characteristic	N = 1,500[1]
day	
1	297 (20%)
2	156 (10%)
3	92 (6.1%)
4	115 (7.7%)
5	223 (15%)
6	300 (20%)
7	317 (21%)
month	
Apr	87 (5.8%)
Jun	877 (58%)
May	536 (36%)
date	
1	115 (7.7%)

Characteristic	**N = 1,500**[1]
10	22 (1.5%)
11	4 (0.3%)
13	2 (0.1%)
14	20 (1.3%)
15	99 (6.6%)
16	66 (4.4%)
17	87 (5.8%)
18	95 (6.3%)
19	67 (4.5%)
2	86 (5.7%)
20	75 (5.0%)
21	27 (1.8%)
22	57 (3.8%)
23	14 (0.9%)
24	6 (0.4%)
25	26 (1.7%)
26	10 (0.7%)
27	1 (<0.1%)
28	18 (1.2%)
29	65 (4.3%)
3	63 (4.2%)
30	86 (5.7%)
31	99 (6.6%)
4	24 (1.6%)
5	60 (4.0%)
6	98 (6.5%)
7	98 (6.5%)
9	10 (0.7%)
hour	
0	67 (4.5%)
1	74 (4.9%)
10	47 (3.1%)
11	64 (4.3%)
12	53 (3.5%)
13	44 (2.9%)

Characteristic	**N = 1,500**[1]
14	34 (2.3%)
15	46 (3.1%)
16	64 (4.3%)
17	42 (2.8%)
18	51 (3.4%)
19	62 (4.1%)
2	62 (4.1%)
20	51 (3.4%)
21	71 (4.7%)
22	67 (4.5%)
23	78 (5.2%)
3	70 (4.7%)
4	65 (4.3%)
5	73 (4.9%)
6	90 (6.0%)
7	85 (5.7%)
8	75 (5.0%)
9	65 (4.3%)

[1] n (%)

Task 1.3 Recode the outcome variable (`sentiment`) into a binary variable such that Positive is equal to 1 and Negative is equal to 0. Calculate and print the frequencies for tweets with positive and negative sentiments.

```
data2 <- data2 |>
  mutate(sentiment = ifelse(sentiment == "Positive", 1, 0))
```

Task 1.4 Load the `reticulate` package and Python library `sentence_transformers`. Then, generate tweet embeddings for each tweet in this dataset using the`allenai/longformer-base-4096` model. Tweet embeddings for each tweet should be a vector of numbers with length 768. Append these embeddings to the original data.

```
library(reticulate)
```

```
## Warning: package 'reticulate' was built under R version 4.4.1

##
## Attaching package: 'reticulate'

## The following object is masked from 'package:rio':
##
##     import
```

```
st <- import('sentence_transformers')
```

```
model.name <- 'allenai/longformer-base-4096'

longformer      <- st$models$Transformer(model.name)
pooling_model   <- st$models$Pooling(longformer$get_word_embedding_dimension())
LFmodel         <- st$SentenceTransformer(modules = list(longformer,pooling_model))
```

```
LFmodel$get_max_seq_length()
```

```
## [1] 4098
```

```
LFmodel$get_sentence_embedding_dimension()
```

```
## [1] 768
```

```
read.embeddings <- LFmodel$encode(data2$tweet,
                                  show_progress_bar = TRUE)
```

```
read.embeddings <- LFmodel$encode(data2$tweet,
                                  show_progress_bar = TRUE)
```

Task 1.5 Remove the two columns `time` and `tweet` from the dataset as you do not need them anymore.

```
 data2 <- data2 |>
  select(-time, -tweet, -wkday, -hms)
```

create ID variable

```
data2 <- tibble::rowid_to_column(data2, "ID")
```

last data preparation

```
outcome <- c('sentiment')

  id      <- c('ID')

  categorical <- c('month')

  cyclic   <- c(
                 'date',
                 'hour',
                 'day')

  embed_number <- paste0('x', 1:768)

# 3) Convert all nominal, ordinal, and binary variables to factors
  # Leave the rest as is

  for(i in categorical){

    data2[,i] <- as.factor(data2[,i])

  }

  #Repeat with numeric just to be sure that they are all numbers
```

```
# for(i in numeric){
#
#   data2[,i] <- as.numeric(data2[,i])
#
# }
```

put the two data frames together

```
data4 <- cbind(data2, read.embeddings)
library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
data4 <- clean_names(data4)

data4 <- data4 |>
  rename( "ID" = "id")
```

Task 1.6 Prepare a recipe using the `recipe()` and `prep()` functions from the `recipes` package for final transformation of the variables in this dataset.

First, make sure you have the most recent developer version of the `recipes` package from Github. If not, install it from Github.

```
#devtools::install_github("tidymodels/recipes")
```

Your recipe should have the following specifications:

- each cyclic variable (`day`, `date`, and `hour`) is recoded into two new variables of sin and cos terms (`?step_harmonic()`).
- `month` variable is recoded into dummy variables using one-hot encoding (`?step_dummy`)
- all numerical embeddings (Dim1 - Dim768) are standardized (`?step_normalize`)

Print the blueprint.

```
blueprint <- recipe(x  = data4,
                    vars  = c(id, outcome, categorical,cyclic,embed_number ),
                    roles = c('ID', 'outcome',
                              rep('predictor',772))) |>
step_harmonic('date', cycle_size = 1, frequency = 1/31) |>

step_harmonic('hour', cycle_size = 1, frequency = 1/24) |>

step_harmonic('day', cycle_size = 1, frequency = 1/7) |>

step_dummy(all_of(categorical),one_hot=TRUE) |>

step_normalize(x1:x768)
```

```r
blueprint
```

```
##

## -- Recipe ----------------------------------------------------------------------

##

## -- Inputs

## Number of variables by role

## outcome:     1
## predictor: 772
## ID:          1

##

## -- Operations

## * Harmonic numeric variables for: "date"

## * Harmonic numeric variables for: "hour"

## * Harmonic numeric variables for: "day"

## * Dummy variables from: all_of(categorical)

## * Centering and scaling for: x1:x768
```

Task 1.7 Finally, apply this recipe to the whole dataset and obtain the final version of the dataset with transformed variables. The final dataset should have 1500 rows and 778 columns as the following:

- 768 columns for tweet embeddings,
- three columns for dummy variables representing the variable `month`,
- two columns for the sin and cos terms representing the variable `day`,
- two columns for the sin and cos terms representing the variable `date`,

- two columns for the sin and cos terms representing the variable `hour`.

```r
prepare <- prep(blueprint,
                training = data4)
prepare
```

```
##

## -- Recipe ----------------------------------------------------------------------

##

## -- Inputs

## Number of variables by role

## outcome:     1
## predictor: 772
## ID:          1

##

## -- Training information

## Training data contained 1500 data points and no incomplete rows.

##
```

```
## -- Operations

## * Harmonic numeric variables for: date | Trained

## * Harmonic numeric variables for: hour | Trained

## * Harmonic numeric variables for: day | Trained

## * Dummy variables from: month | Trained

## * Centering and scaling for: x1, x2, x3, x4, x5, x6, x7, x8, ... | Trained
```

```
baked_data <- bake(prepare, new_data = data4)

baked_data
```

```
## # A tibble: 1,500 x 779
##       ID sentiment      x1       x2      x3      x4      x5     x6      x7
##    <int>     <dbl>   <dbl>    <dbl>   <dbl>   <dbl>   <dbl>  <dbl>   <dbl>
##  1     1         0 -0.509   0.483    0.881   0.578   0.374   0.123 -1.37
##  2     2         1  0.0167 -1.14    -0.585   0.941  -1.84   -0.571  0.0760
##  3     3         1 -0.0711 -1.19    -0.144   0.536  -0.504   0.152  0.178
##  4     4         1 -0.952   1.28     0.659  -0.584   0.590   0.651 -0.376
##  5     5         1 -0.546  -0.622    1.05   -1.47   -0.118   1.12  -0.467
##  6     6         1 -1.16   -1.28    -0.661   1.53    0.149   0.139 -1.05
##  7     7         0  0.561   0.00783 -1.27   -0.785  -1.78   -1.57   1.31
##  8     8         0  1.25   -1.12     0.0905 -0.614   0.0733 -1.05  -1.19
##  9     9         1 -0.0983  0.332   -0.718   0.0439  0.940  -0.400 -0.337
## 10    10         1 -0.0604  0.115   -0.511   1.57   -0.290   0.466  0.721
## # i 1,490 more rows
## # i 770 more variables: x8 <dbl>, x9 <dbl>, x10 <dbl>, x11 <dbl>, x12 <dbl>,
## #   x13 <dbl>, x14 <dbl>, x15 <dbl>, x16 <dbl>, x17 <dbl>, x18 <dbl>,
## #   x19 <dbl>, x20 <dbl>, x21 <dbl>, x22 <dbl>, x23 <dbl>, x24 <dbl>,
## #   x25 <dbl>, x26 <dbl>, x27 <dbl>, x28 <dbl>, x29 <dbl>, x30 <dbl>,
## #   x31 <dbl>, x32 <dbl>, x33 <dbl>, x34 <dbl>, x35 <dbl>, x36 <dbl>,
## #   x37 <dbl>, x38 <dbl>, x39 <dbl>, x40 <dbl>, x41 <dbl>, x42 <dbl>, ...
```

Task 1.8 Export the final dataset (1500 x 778) as a .csv file.

```
write.csv(baked_data,"~/Documents/Everything/PhD_harddrive/EDLD_654/assignment_1/final_dataset1")
```

Task 2: Preprocessing Continuous and Categorical Variables

Description

For the second part of the assignment, we are going to use a student performance dataset. The data attributes include student grades, demographic, social and school related features, and it was collected by using school reports and questionnaires. The dataset has 649 observations and 31 variables. This data is available as an input data in this R notebook (**'../input/student-performance/student.csv'**).

Below is a table of data dictionary for the variables in this dataset.

Variable	Name	Description
1	school	student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
2	sex	student's sex assigned at birth (binary: 'F' - female or 'M' - male)
3	age	student's age (numeric: from 15 to 22)
4	address	student's home address type (binary: 'U' - urban or 'R' - rural)
5	famsize	family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
6	Pstatus	parent's cohabitation status (binary: 'T' - living together or 'A' - apart)

Variable	Name	Description
7	Medu	mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education, 4 - higher education
8	Fedu	father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education, 4 - higher education
9	Mjob	mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
10	Fjob	father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
11	reason	reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
12	guardian	student's guardian (nominal: 'mother', 'father' or 'other')
13	traveltime	home to school travel time (numeric: 1 - <15 min, 2 - 15 to 30 min, 3 - 30min to 1 hour, or 4 >1hour)
14	studytime	weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15	failures	number of past class failures (numeric: n if 1<=n<3, else 4)
16	schoolsup	extra educational support (binary: yes or no)
17	famsup	family educational support (binary: yes or no)
18	paid	extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
19	activities	extra-curricular activities (binary: yes or no)
20	nursery	attended nursery school (binary: yes or no)
21	higher	wants to take higher education (binary: yes or no)
22	internet	Internet access at home (binary: yes or no)
23	romantic	with a romantic relationship (binary: yes or no)
24	famrel	quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
25	freetime	free time after school (numeric: from 1 - very low to 5 - very high)
26	goout	going out with friends (numeric: from 1 - very low to 5 - very high)
27	Dalc	workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
28	Walc	weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
29	health	current health status (numeric: from 1 - very bad to 5 - very good)
30	absences	number of school absences (numeric: from 0 to 93)
31	G3	final grade (numeric: from 0 to 20, output target)

Tasks

Task 2.1 Import the student performance data into the R environment. You can give any name to this data object. Print the structure of this data object using the `str` function.

```r
dat <- rio::import(here("assignment_1/data/student2.csv"))

str(dat)
```

```
## 'data.frame':    649 obs. of  32 variables:
##  $ student_id: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ school    : chr  "GP" "GP" "GP" "GP" ...
##  $ sex       : chr  "F" "F" "F" "F" ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : chr  "U" "U" "U" "U" ...
##  $ famsize   : chr  "GT3" "GT3" "LE3" "GT3" ...
##  $ Pstatus   : chr  "A" "T" "T" "T" ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 NA 2 4 2 4 ...
##  $ Mjob      : chr  "at_home" "at_home" "at_home" "health" ...
##  $ Fjob      : chr  "teacher" "other" "other" "services" ...
##  $ reason    : chr  "course" "course" "other" "home" ...
##  $ guardian  : chr  "mother" "father" "mother" "mother" ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ schoolsup : chr  "yes" "no" "yes" "no" ...
##  $ famsup    : chr  "no" "yes" "no" "yes" ...
##  $ paid      : chr  "no" "no" "no" "no" ...
##  $ activities: chr  "no" "no" "no" "yes" ...
##  $ nursery   : chr  "yes" "no" "yes" "yes" ...
##  $ higher    : chr  "yes" "yes" "yes" "yes" ...
##  $ internet  : chr  "no" "yes" "yes" "yes" ...
##  $ romantic  : chr  "no" "no" "no" "yes" ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 NA 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 NA 1 ...
##  $ health    : int  3 3 3 5 5 5 3 NA NA 5 ...
##  $ absences  : int  4 2 6 0 0 6 0 2 0 0 ...
##  $ G3        : int  11 11 12 14 13 13 13 13 17 13 ...
```

Task 2.2 Using the `ff_glimpse()` function from the `finalfit` package, provide a snapshot of missingness in this dataset. This function also returns the number of levels for categorical variables. If there is any variable with large amount of missingness (e.g. more than 75%), remove this variable from the dataset.

```
library(finalfit)

ff_glimpse(dat)
```

```
## $Continuous
##                 label var_type   n missing_n missing_percent  mean    sd  min
## student_id student_id    <int> 649         0             0.0 325.0 187.5  1.0
## age               age    <int> 597        52             8.0  16.8   1.2 15.0
## Medu             Medu    <int> 630        19             2.9   2.5   1.1  0.0
## Fedu             Fedu    <int> 630        19             2.9   2.3   1.1  0.0
## traveltime traveltime    <int> 636        13             2.0   1.6   0.8  1.0
## studytime   studytime    <int> 623        26             4.0   1.9   0.8  1.0
## failures     failures    <int> 630        19             2.9   0.2   0.6  0.0
## famrel         famrel    <int> 643         6             0.9   3.9   0.9  1.0
## freetime     freetime    <int> 623        26             4.0   3.2   1.0  1.0
## goout           goout    <int> 630        19             2.9   3.2   1.2  1.0
## Dalc             Dalc    <int> 604        45             6.9   1.5   0.9  1.0
## Walc             Walc    <int> 591        58             8.9   2.3   1.3  1.0
## health         health    <int> 604        45             6.9   3.5   1.5  1.0
## absences     absences    <int> 636        13             2.0   3.7   4.7  0.0
## G3                 G3    <int> 649         0             0.0  11.9   3.2  0.0
##            quartile_25 median quartile_75   max
## student_id       163.0  325.0       487.0 649.0
## age               16.0   17.0        18.0  22.0
## Medu               2.0    2.0         4.0   4.0
## Fedu               1.0    2.0         3.0   4.0
## traveltime         1.0    1.0         2.0   4.0
## studytime          1.0    2.0         2.0   4.0
## failures           0.0    0.0         0.0   3.0
## famrel             4.0    4.0         5.0   5.0
## freetime           3.0    3.0         4.0   5.0
## goout              2.0    3.0         4.0   5.0
## Dalc               1.0    1.0         2.0   5.0
## Walc               1.0    2.0         3.0   5.0
## health             2.0    4.0         5.0   5.0
## absences           0.0    2.0         6.0  32.0
## G3                10.0   12.0        14.0  19.0
## 
## $Categorical
##                  label var_type   n missing_n missing_percent levels_n levels
## school         school    <chr> 649         0             0.0        3      -
## sex               sex    <chr> 649         0             0.0        3      -
## address       address    <chr> 649         0             0.0        3      -
## famsize       famsize    <chr> 649         0             0.0        3      -
## Pstatus       Pstatus    <chr> 649         0             0.0        3      -
## Mjob             Mjob    <chr> 649         0             0.0        6      -
## Fjob             Fjob    <chr> 649         0             0.0        6      -
## reason         reason    <chr> 649         0             0.0        5      -
## guardian     guardian    <chr> 649         0             0.0        4      -
## schoolsup   schoolsup    <chr> 649         0             0.0        3      -
## famsup         famsup    <chr> 649         0             0.0        3      -
## paid             paid    <chr> 649         0             0.0        3      -
## activities activities    <chr> 649         0             0.0        3      -
## nursery       nursery    <chr> 649         0             0.0        3      -
## higher         higher    <chr> 649         0             0.0        3      -
## internet     internet    <chr> 649         0             0.0        3      -
## romantic     romantic    <chr> 649         0             0.0        3      -
##            levels_count levels_percent
## school                -              -
## sex                   -              -
## address               -              -
## famsize               -              -
## Pstatus               -              -
## Mjob                  -              -
## Fjob                  -              -
## reason                -              -
## guardian              -              -
## schoolsup             -              -
## famsup                -              -
## paid                  -              -
## activities            -              -
## nursery               -              -
## higher                -              -
## internet              -              -
## romantic              -              -
```

Note: Each variable except *student_id* and *G3* has at least one missing value in this dataset.

Task 2.3 Most of the variables in this dataset are categorical, and particularly a binary variable with a Yes and No response. Check the frequency of unique values for all categorical variables. If there is any inconsistency (e.g., Yes is coded as both 'y' and 'Y') for any of these variables in terms of how values are coded, fix them. Also, check the distribution of numeric variables and make sure there is no anomaly.

```
dat[dat == ''] <- NA
```

```
table(dat$school)
```

```
## 
##  GP  MS 
## 400 217
```

```
table(dat$sex)
```

```
## 
##   F   M 
## 341 243
```

```
describe(dat$age)
```

```
##    vars   n  mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 597 16.76 1.22     17   16.72 1.48  15  22     7 0.43     0.09 0.05
```

```
table(dat$address)
```

```
## 
##   R   U 
## 180 411
```

```
table(dat$famsize)
```

```
## 
## GT3 LE3 
## 413 171
```

```
table(dat$Pstatus)
```

```
## 
##   A   T 
##  79 538
```

```
table(dat$Medu)
```

```
## 
##   0   1   2   3   4 
##   5 138 183 133 171
```

```
table(dat$Fedu)
```

```
## 
##   0   1   2   3   4 
##   6 172 203 127 122
```

```
table(dat$Mjob)
```

```
## 
##  at_home   health    other services  teacher 
##      133       47      255      131       70
```

```
table(dat$Fjob)
```

```
## 
##  at_home   health    other services  teacher 
##       41       23      359      178       35
```

```
table(dat$reason)
```

```
##
##     course       home      other reputation
##        283        149         72        139
```

```
table(dat$guardian)
```

```
##
## father mother  other
##    134    423     40
```

```
table(dat$traveltime)
```

```
##
##   1   2   3   4
## 360 207  53  16
```

```
table(dat$studytime)
```

```
##
##   1   2   3   4
## 206 291  93  33
```

```
table(dat$failures) #there are no 4 values, seems like this may be incorrect
```

```
##
##   0   1   2   3
## 531  70  16  13
```

```
table(dat$schoolsup)
```

```
##
##  no yes
## 564  66
```

```
table(dat$famsup)
```

```
##
##  no yes
## 243 380
```

```
table(dat$paid)
```

```
##
##  no yes
## 599  37
```

```
table(dat$activities)
```

```
##
##  no yes
## 326 310
```

```
table(dat$nursery)
```

```
##
##  no yes
## 121 502
```

```
table(dat$higher)
```

```
##
##  no yes
##  65 519
```

```
table(dat$internet)
```

```
##
##  no yes
## 136 448
```

```
table(dat$romantic)
```

```
##
##  no yes
## 374 217
```

```
table(dat$famrel)
```

```
##
##   1   2   3   4   5
##  20  29 100 316 178
```

```
table(dat$freetime)
```

```
##
##   1   2   3   4   5
##  45 105 244 169  60
```

```
table(dat$goout)
```

```
##
##   1   2   3   4   5
##  47 141 200 133 109
```

```
table(dat$Dalc)
```

```
##
##   1   2   3   4   5
## 418 113  43  16  14
```

```
table(dat$Walc)
```

```
##
##   1   2   3   4   5
## 223 138 107  80  43
```

```
table(dat$health)
```

```
##
##   1   2   3   4   5
##  85  72 112 100 235
```

```
table(dat$absences)
```

```
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  18  21  22
## 236  12 110   7  91  12  48   3  40   7  21   5  12   1   8   2  10   3   2   2
##  24  26  30  32
##   1   1   1   1
```

```r
describe(dat$absences)
```

```
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 636 3.69 4.66      2    2.82 2.97   0  32    32 2.01     5.66 0.18
```

```r
table(dat$G3)
```

```
## 
##   0   1   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19 
##  15   1   1   3  10  35  35  97 104  72  82  63  49  36  29  15   2 
```

```r
describe(dat$G3)
```

```
##    vars   n  mean   sd median trimmed  mad min max range  skew kurtosis   se
## X1    1 649 11.91 3.23     12   12.04 2.97   0  19    19 -0.91     2.66 0.13
```

Below is a list of variables that should be treated as categorical (with a nominal scale):

- school
- sex
- address
- famsize
- Pstatus
- Mjob
- Fjob
- reason
- guardian
- schoolsup
- famsup
- paid
- activities
- nursery
- higher
- internet
- romantic

Below is a list of variables that should be treated as numeric (with an ordinal or continuous scale):

- Medu
- Fedu
- traveltime
- studytime
- failures
- famrel
- freetime
- goout
- Dalc
- Walc
- health
- age
- absences

The purpose is to check the observed data and make sure the observed data is aligned with the data dictionary and there are no unexpected or unusual values that are not consistent with what expect from data. If there is such data points, they have to be fixed or issued should be resolved before moving forward.

Task 2.4 Prepare a recipe using the `recipe()` and `prep()` functions from the `recipes` package for final transformation of the variables in this dataset.

Suppose that we categorize the variables in this datasets as the following:

- `student_id` is the ID variable
- `G3` is the outcome variable
- `Medu`,`Fedu`,`traveltime`,`studytime`,`failures`,`famrel`,`freetime`,`goout`,`Dalc`,`Walc`,`health`,`age`,`absences` are numeric predictors
- `school`,`sex`,`address`,`famsize`,`Pstatus`,`Mjob`,`Fjob`,`reason`,`guardian`,`schoolsup`,`famsup`, `paid`,`activities`,`nursery`,`hig` are all categorical predictors.

```
id <- c('student_id')
outcome <- c('G3')
numeric <- c('Medu','Fedu','traveltime','studytime','failures','famrel','freetime','goout','Dalc','Walc
categorical <- c('school','sex','address','famsize','Pstatus','Mjob','Fjob','reason','guardian','school
```

```
for(i in categorical){

    dat[,i] <- as.factor(dat[,i])

  }

  #Repeat with numeric just to be sure that they are all numbers

   for(i in numeric){

    dat[,i] <- as.numeric(dat[,i])

   }
```

```
str(dat)
```

```
## 'data.frame':    649 obs. of  32 variables:
##  $ student_id: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ school    : Factor w/ 2 levels "GP","MS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex       : Factor w/ 2 levels "F","M": 1 1 1 1 1 2 2 1 2 2 ...
##  $ age       : num  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : Factor w/ 2 levels "R","U": 2 2 2 2 2 2 2 2 NA 2 ...
##  $ famsize   : Factor w/ 2 levels "GT3","LE3": 1 1 2 1 1 2 2 1 2 1 ...
##  $ Pstatus   : Factor w/ 2 levels "A","T": 1 2 2 2 2 2 2 1 1 2 ...
##  $ Medu      : num  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : num  4 1 1 2 3 NA 2 4 2 4 ...
##  $ Mjob      : Factor w/ 5 levels "at_home","health",..: 1 1 1 2 3 4 3 3 4 3 ...
##  $ Fjob      : Factor w/ 5 levels "at_home","health",..: 5 3 3 4 3 NA 3 5 3 3 ...
##  $ reason    : Factor w/ 4 levels "course","home",..: 1 1 3 2 2 4 2 2 2 2 ...
##  $ guardian  : Factor w/ 3 levels "father","mother",..: 2 1 2 2 1 2 2 2 2 2 ...
##  $ traveltime: num  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : num  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ schoolsup : Factor w/ 2 levels "no","yes": 2 1 2 1 1 1 1 2 1 1 ...
##  $ famsup    : Factor w/ 2 levels "no","yes": 1 2 1 2 2 2 1 2 2 2 ...
##  $ paid      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ activities: Factor w/ 2 levels "no","yes": 1 1 1 2 1 2 1 1 1 2 ...
##  $ nursery   : Factor w/ 2 levels "no","yes": 2 1 2 2 2 2 2 2 2 2 ...
##  $ higher    : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ internet  : Factor w/ 2 levels "no","yes": 1 2 2 2 1 2 2 1 2 NA ...
##  $ romantic  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ famrel    : num  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : num  3 3 NA 2 3 4 4 1 2 5 ...
##  $ goout     : num  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : num  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : num  1 1 3 1 2 2 1 1 NA 1 ...
##  $ health    : num  3 3 3 5 5 5 3 NA NA 5 ...
##  $ absences  : num  4 2 6 0 0 6 0 2 0 0 ...
##  $ G3        : int  11 11 12 14 13 13 13 13 17 13 ...
```

Your recipe should have the following specifications in the order below:

```
blueprint <- recipe(x  = dat,
                    vars  = c(id, outcome, categorical, numeric ),
                    roles = c('id', 'outcome', rep('predictor',30))) |>
step_indicate_na(all_of(categorical),all_of(numeric)) |>
step_zv(all_predictors()) |>
step_impute_mean(all_of(numeric)) |>
step_impute_mode(all_of(categorical)) |>
step_poly(all_of(numeric),degree=3) |>
step_normalize(paste0(numeric,'_poly_1'),
               paste0(numeric,'_poly_2'),
               paste0(numeric,'_poly_3')) |>

  # One-hot encoding for all categorical variables

  step_dummy(all_of(categorical),one_hot=TRUE)
```

- create an indicator variable for missingness for all predictors,
- remove the numeric predictors with zero variance,
- replace missing values with mean for numeric predictors,
- replace missing values with mode for categorical predictors,
- expand numeric predictors using using polynomial basis functions with three degrees of freedom and standardize,
- recode categorical predictors into dummy variables using one-hot encoding.

Print the blueprint.

```
blueprint
```

```
##

## -- Recipe ----------------------------------------------------------------------

##

## -- Inputs

## Number of variables by role

## outcome:    1
## predictor: 30
## id:         1

##

## -- Operations

## * Creating missing data variable indicators for: all_of(categorical), ...

## * Zero variance filter on: all_predictors()

## * Mean imputation for: all_of(numeric)
## * Mode imputation for: all_of(categorical)

## * Orthogonal polynomials on: all_of(numeric)

## * Centering and scaling for: paste0(numeric, "_poly_1"), ...

## * Dummy variables from: all_of(categorical)
```

Task 2.5 Finally, apply this recipe to the whole dataset and obtain the final version of the dataset with transformed variables. The final dataset should have 649 rows and 114 columns as the following:

- one column representing the ID variable, `id`,
- one column representing the outcome variable, `score`,
- 30 columns representing missing indicator variables,
- three columns for polynomial terms for each of the numeric variables,
- two columns for dummy variables representing `school`,
- two columns for dummy variables representing `sex`,
- two columns for dummy variables representing `address`,
- two columns for dummy variables representing `famsize`,
- two columns for dummy variables representing `Pstatus`,
- five columns for dummy variables representing `Mjob`,
- five columns for dummy variables representing `Fjob`,
- four columns for dummy variables representing `reason`,
- three columns for dummy variables representing `guardian`,
- two columns for dummy variables representing `schoolsup`,
- two columns for dummy variables representing `famsup`,
- two columns for dummy variables representing `paid`,
- two columns for dummy variables representing `activities`,
- two columns for dummy variables representing `nursery`,
- two columns for dummy variables representing `higher`,
- two columns for dummy variables representing `internet`,
- two columns for dummy variables representing `romantic`

Task 2.7 Export the final dataset (649 x 114) as a .csv file.

```r
write.csv(baked_data, "~/Documents/Everything/PhD_harddrive/EDLD_654/assignment_1/final_dataset2")
```