Data Wrangling (Data Preprocessing)

Practical assessment 1

Jason Wang

11/08/2024

Setup

```
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(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(magrittr)
library(rmarkdown)
library(rvest)
library(gdata)
```

```
##
## Attaching package: 'gdata'
```

```
## The following objects are masked from 'package:dplyr':
##
##     combine, first, last, starts_with
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following object is masked from 'package:utils':
##
##     object.size
```

```
## The following object is masked from 'package:base':
##
##     startsWith
```

```
library(readr)
```

```
##
## Attaching package: 'readr'
```

```
## The following object is masked from 'package:rvest':
##
##     guess_encoding
```

```
library(openxlsx)
```

Student names, numbers and percentage of contributions

Group information

Student name	Student number	Percentage of contribution
Jason Wang	s4134626	100%

Data Description

The title is “Overseas migrant arrivals by country of birth, state/territory - financial years, 2004-05 to 2022-23.”

We are looking at the migration of people from different countries of birth to Australia.

The data is https://www.abs.gov.au/statistics/people/population/overseas-migration/2022-23-financial-year/34070DO002_202223.xlsx (https://www.abs.gov.au/statistics/people/population/overseas-migration/2022-23-financial-year/34070DO002_202223.xlsx).

It is sourced from the Australian Bureau of Statistics. It has 21 variables and 248 observations.It includes 1 numerical and 2 categorical variables.

Standard Australian Classification of Countries (SACC) refers to the code a country has that is classified by Australia. It incorporates numbers, however these numbers are classed as categorical because these numbers categorise countries and do not signify any numeric value, similarly to mobile phone number. More specifically, it is a nominal variable because there is no real order in these numbers, thus cannot be ordinal.

Country of birth is also a nominal categorical variable. Nominal because you cannot order them. It refers to the country of birth migrants are coming from.

The specific year is a discrete numerical variable because it is increasing every year by 1, and it is not precise measurement. Separate values.

Read/Import Data

```
migration.url <- "https://www.abs.gov.au/statistics/people/population/overseas-migration/2022
-23-financial-year/34070DO002_202223.xlsx"

migration <- read.xlsx(migration.url, sheet = "Table 2.1", startRow = 15)
migration <- migration[1:(nrow(migration) - 6), ]

head(migration)
```

	SACC.code(e)	Country.of.birth(e)	2004-05	2005-06	2006-07	2007-08	2008-09	2009-10
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1101	Australia	47290	50150	52920	53190	56470	54530
2	1102	Norfolk Island(g)	30	30	40	40	30	30
3	1199	Aust E T, nec	0	0	0	0	0	0
4	1201	New Zealand	32030	32770	36300	42070	38090	31580
5	1301	New Caledonia	200	150	160	190	190	220
6	1302	PNG	1370	1620	1610	1820	1710	1550

6 rows | 1-10 of 22 columns

I defined migration as the url for the online Excel file. I then used read.xlsx from the openxlsx package to read the excel file. The file has many sheets, I chose the first sheet which was named “Table 2.1” and started at row 15 because that is where the data starts.

I proceeded to delete the last 6 rows because they included unnecessary totals and spacings between rows. nrow(migration) totaled to 254, and subtracted 6 to get 248 observations. In summary, i subsetted my data by removing the first 15 and last 6 rows to incorporate only the data. To read the data, i used the head function.

Inspect and Understand

```
dim(migration)
```

```
## [1] 248  21
```

```
colnames(migration)
```

```
##  [1] "SACC.code(e)"        "Country.of.birth(e)" "2004-05"
##  [4] "2005-06"             "2006-07"             "2007-08"
##  [7] "2008-09"             "2009-10"             "2010-11"
## [10] "2011-12"             "2012-13"             "2013-14"
## [13] "2014-15"             "2015-16"             "2016-17"
## [16] "2017-18"             "2018-19"             "2019-20"
## [19] "2020-21"             "2021-22"             "2022-23(f)"
```

```
str(migration)
```

```
## 'data.frame':    248 obs. of  21 variables:
##  $ SACC.code(e)       : chr  "1101" "1102" "1199" "1201" ...
##  $ Country.of.birth(e): chr  "Australia" "Norfolk Island(g)" "Aust E T, nec" "New Zealand"
...
##  $ 2004-05            : num  47290 30 0 32030 200 ...
##  $ 2005-06            : num  50150 30 0 32770 150 ...
##  $ 2006-07            : num  52920 40 0 36300 160 ...
##  $ 2007-08            : num  53190 40 0 42070 190 ...
##  $ 2008-09            : num  56470 30 0 38090 190 ...
##  $ 2009-10            : num  54530 30 0 31580 220 ...
##  $ 2010-11            : num  53150 30 0 43430 170 ...
##  $ 2011-12            : num  50610 30 0 48220 130 ...
##  $ 2012-13            : num  50870 40 0 41500 130 ...
##  $ 2013-14            : num  46320 30 0 28440 110 ...
##  $ 2014-15            : num  46350 20 0 23300 150 ...
##  $ 2015-16            : num  48400 70 0 24220 130 ...
##  $ 2016-17            : num  50810 80 0 22660 160 ...
##  $ 2017-18            : num  49810 10 0 21270 120 ...
##  $ 2018-19            : num  50420 10 0 20960 130 ...
##  $ 2019-20            : num  60710 0 0 15640 100 ...
##  $ 2020-21            : num  37020 0 0 12000 40 ...
##  $ 2021-22            : num  34910 0 0 16280 100 ...
##  $ 2022-23(f)         : num  33630 10 0 27510 120 ...
```

```
migration$`Country.of.birth(e)` <- as.factor(migration$`Country.of.birth(e)`)
str(migration)
```

```
## 'data.frame':    248 obs. of  21 variables:
##  $ SACC.code(e)       : chr  "1101" "1102" "1199" "1201" ...
##  $ Country.of.birth(e): Factor w/ 248 levels "Adelie Land",..: 15 162 14 157 156 173 199 2
39 91 114 ...
##  $ 2004-05            : num  47290 30 0 32030 200 ...
##  $ 2005-06            : num  50150 30 0 32770 150 ...
##  $ 2006-07            : num  52920 40 0 36300 160 ...
##  $ 2007-08            : num  53190 40 0 42070 190 ...
##  $ 2008-09            : num  56470 30 0 38090 190 ...
##  $ 2009-10            : num  54530 30 0 31580 220 ...
##  $ 2010-11            : num  53150 30 0 43430 170 ...
##  $ 2011-12            : num  50610 30 0 48220 130 ...
##  $ 2012-13            : num  50870 40 0 41500 130 ...
##  $ 2013-14            : num  46320 30 0 28440 110 ...
##  $ 2014-15            : num  46350 20 0 23300 150 ...
##  $ 2015-16            : num  48400 70 0 24220 130 ...
##  $ 2016-17            : num  50810 80 0 22660 160 ...
##  $ 2017-18            : num  49810 10 0 21270 120 ...
##  $ 2018-19            : num  50420 10 0 20960 130 ...
##  $ 2019-20            : num  60710 0 0 15640 100 ...
##  $ 2020-21            : num  37020 0 0 12000 40 ...
##  $ 2021-22            : num  34910 0 0 16280 100 ...
##  $ 2022-23(f)         : num  33630 10 0 27510 120 ...
```

```
levels(migration$`Country.of.birth(e)`)
```

```
##   [1] "Adelie Land"       "Afghanistan"       "Aland Islands"
##   [4] "Albania"           "Algeria"           "Andorra"
##   [7] "Angola"            "Anguilla"          "Antigua/Barbuda"
##  [10] "Argentina"         "Argentinian A T"   "Armenia"
##  [13] "Aruba"             "Aust E T, nec"     "Australia"
##  [16] "Australian A T"    "Austria"           "Azerbaijan"
##  [19] "Bahamas"           "Bahrain"           "Bangladesh"
##  [22] "Barbados"          "Belarus"           "Belgium"
##  [25] "Belize"            "Benin"             "Bermuda"
##  [28] "Bhutan"            "Bolivia"           "Bonaire/SE/Saba"
##  [31] "Bosnia/Herzegov"   "Botswana"          "Brazil"
##  [34] "British A T"       "Brunei"            "Bulgaria"
##  [37] "Burkina Faso"      "Burundi"           "Cabo Verde"
##  [40] "Cambodia"          "Cameroon"          "Canada"
##  [43] "Cayman Islands"    "Cent Africa Rep"   "Chad"
##  [46] "Chile"             "Chilean A T"       "China"
##  [49] "Colombia"          "Comoros"           "Congo, Dem Rep"
##  [52] "Congo, Rep"        "Cook Islands"      "Costa Rica"
##  [55] "Cote d'Ivoire"     "Croatia"           "Cuba"
##  [58] "Curacao"           "Cyprus"            "Czechia"
##  [61] "Denmark"           "Djibouti"          "Dominica"
##  [64] "Dominican Rep"     "Ecuador"           "Egypt"
##  [67] "El Salvador"       "Equator Guinea"    "Eritrea"
##  [70] "Estonia"           "Eswatini"          "Ethiopia"
##  [73] "Falkland Is"       "Faroe Islands"     "Fiji"
##  [76] "Finland"           "France"            "French Guiana"
##  [79] "French Poly"       "Gabon"             "Gambia"
##  [82] "Gaza Str/W Bank"   "Georgia"           "Germany"
##  [85] "Ghana"             "Gibraltar"         "Greece"
##  [88] "Greenland"         "Grenada"           "Guadeloupe"
##  [91] "Guam"              "Guatemala"         "Guinea"
##  [94] "Guinea-Bissau"     "Guyana"            "Haiti"
##  [97] "Holy See"          "Honduras"          "Hong Kong"
## [100] "Hungary"           "Iceland"           "India"
## [103] "Indonesia"         "Iran"              "Iraq"
## [106] "Ireland"           "Israel"            "Italy"
## [109] "Jamaica"           "Japan"             "Jordan"
## [112] "Kazakhstan"        "Kenya"             "Kiribati"
## [115] "Korea, North"      "Korea, South"      "Kosovo"
## [118] "Kuwait"            "Kyrgyzstan"        "Laos"
## [121] "Latvia"            "Lebanon"           "Lesotho"
## [124] "Liberia"           "Libya"             "Liechtenstein"
## [127] "Lithuania"         "Luxembourg"        "Macau"
## [130] "Madagascar"        "Malawi"            "Malaysia"
## [133] "Maldives"          "Mali"              "Malta"
## [136] "Marshall Is"       "Martinique"        "Mauritania"
## [139] "Mauritius"         "Mayotte"           "Mexico"
## [142] "Micronesia, F S"   "Moldova"           "Monaco"
## [145] "Mongolia"          "Montenegro"        "Montserrat"
## [148] "Morocco"           "Mozambique"        "Myanmar"
## [151] "N Mariana Is"      "Namibia"           "Nauru"
## [154] "Nepal"             "Netherlands"       "New Caledonia"
## [157] "New Zealand"       "Nicaragua"         "Niger"
## [160] "Nigeria"           "Niue"              "Norfolk Island(g)"
## [163] "North Macedonia"   "Norway"            "Oman"
## [166] "Pakistan"          "Palau"             "Panama"
## [169] "Paraguay"          "Peru"              "Philippines"
## [172] "Pitcairn Is"       "PNG"               "Poland"
## [175] "Polynesia, nec"    "Portugal"          "Puerto Rico"
## [178] "Qatar"             "Queen Maud Land"   "Reunion"
## [181] "Romania"           "Ross Dependency"   "Russia"
## [184] "Rwanda"            "S America, nec"    "Samoa"
## [187] "Samoa American"    "San Marino"        "Sao Tome/Princ"
## [190] "Saudi Arabia"      "Senegal"           "Serbia"
## [193] "Seychelles"        "Sierra Leone"      "Singapore"
## [196] "Sint Maarten Dp"   "Slovakia"          "Slovenia"
## [199] "Solomon Islands"   "Somalia"           "South Africa"
## [202] "South Sudan"       "Sp North Africa"   "Spain"
## [205] "Sri Lanka"         "St Barthelemy"     "St Helena"
## [208] "St Kitts/Nevis"    "St Lucia"          "St Martin (Fr)"
## [211] "St Pierre/Mique"   "St Vinc/Grenad"    "Sudan"
## [214] "Suriname"          "Sweden"            "Switzerland"
## [217] "Syria"             "Taiwan"            "Tajikistan"
## [220] "Tanzania"          "Thailand"          "Timor-Leste"
## [223] "Togo"              "Tokelau"           "Tonga"
## [226] "Trinidad/Tobago"   "Tunisia"           "Turkey"
## [229] "Turkmenistan"      "Turks/Caicos Is"   "Tuvalu"
## [232] "Uganda"            "UK, CIs & IOM"     "Ukraine"
## [235] "Unit Arab Emir"    "Uruguay"           "USA"
## [238] "Uzbekistan"        "Vanuatu"           "Venezuela"
## [241] "Vietnam"           "Virgin Is, Brit"   "Virgin Is, US"
## [244] "Wallis/Futuna"     "Western Sahara"    "Yemen"
## [247] "Zambia"            "Zimbabwe"
```

dim(migration) checks the number of observations and variables respectively. colnames(migration) states the names of each variable. To summarise the types of variables, we want to use str() and not summary(). str(migration) shows the structure of the dataframe, listing the variables and its data type, and the obseverations in each variable.

I converted country of birth to a factor form from character by using as.factor because we want to read it as factor form. It represents distinct categories so factor form is much preferred, and it is easier for R to read. SACC should be left as character variable because it functions as an identifier as mentioned prior. I then used levels() to check the levels of migration.

Subsetting

```
migration_subset <- migration[1:10, ]
migration_matrix <- as.matrix(migration_subset)
print(migration_matrix)
```

```
##    SACC.code(e) Country.of.birth(e) 2004-05 2005-06 2006-07 2007-08 2008-09
## 1  "1101"       "Australia"         "47290" "50150" "52920" "53190" "56470"
## 2  "1102"       "Norfolk Island(g)" "   30" "   30" "   40" "   40" "   30"
## 3  "1199"       "Aust E T, nec"     "    0" "    0" "    0" "    0" "    0"
## 4  "1201"       "New Zealand"       "32030" "32770" "36300" "42070" "38090"
## 5  "1301"       "New Caledonia"     "  200" "  150" "  160" "  190" "  190"
## 6  "1302"       "PNG"               " 1370" " 1620" " 1610" " 1820" " 1710"
## 7  "1303"       "Solomon Islands"   "  280" "  190" "  190" "  200" "  190"
## 8  "1304"       "Vanuatu"           "  100" "  120" "   90" "  100" "  100"
## 9  "1401"       "Guam"              "    0" "   10" "   10" "   10" "   10"
## 10 "1402"       "Kiribati"          "   80" "   70" "   70" "   70" "   80"
##    2009-10 2010-11 2011-12 2012-13 2013-14 2014-15 2015-16 2016-17 2017-18
## 1  "54530" "53150" "50610" "50870" "46320" "46350" "48400" "50810" "49810"
## 2  "   30" "   30" "   30" "   40" "   30" "   20" "   70" "   80" "   10"
## 3  "    0" "    0" "    0" "    0" "    0" "    0" "    0" "    0" "    0"
## 4  "31580" "43430" "48220" "41500" "28440" "23300" "24220" "22660" "21270"
## 5  "  220" "  170" "  130" "  130" "  110" "  150" "  130" "  160" "  120"
## 6  " 1550" " 1560" " 1640" " 1580" " 1270" " 1240" " 1490" " 1320" " 1140"
## 7  "  170" "  220" "  220" "  230" "  150" "  180" "  170" "  170" "  200"
## 8  "  140" "  110" "  130" "  140" "   80" "   80" "  110" "  110" "  110"
## 9  "   10" "   10" "   10" "   10" "   10" "    0" "   10" "    0" "   10"
## 10 "   80" "   40" "   50" "   50" "   40" "   40" "   40" "   70" "  120"
##    2018-19 2019-20 2020-21 2021-22 2022-23(f)
## 1  "50420" "60710" "37020" "34910" "33630"
## 2  "   10" "    0" "    0" "    0" "   10"
## 3  "    0" "    0" "    0" "    0" "    0"
## 4  "20960" "15640" "12000" "16280" "27510"
## 5  "  130" "  100" "   40" "  100" "  120"
## 6  " 1220" " 1400" "  350" " 1320" " 2220"
## 7  "  190" "  400" " 1070" " 2310" " 1660"
## 8  "  300" " 2580" " 2810" " 2630" " 2750"
## 9  "   10" "   10" "    0" "   10" "   10"
## 10 "   90" "  180" "  220" "  240" "  690"
```

```
str(migration_matrix)
```

```
##  chr [1:10, 1:21] "1101" "1102" "1199" "1201" "1301" "1302" "1303" "1304" ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:10] "1" "2" "3" "4" ...
##   ..$ : chr [1:21] "SACC.code(e)" "Country.of.birth(e)" "2004-05" "2005-06" ...
```

I subsetted the dataset. The syntax when subsetting goes as rows then columns which is separated by a comma. Thus 1:10 means the first 10 observations, and the blank space after the comma means keep all variables, which is what we want. To turn it into a matrix, i used as.matrix to convert the subsetted dataframe into a matrix. Printed to see and it had correctly turned into a matrix.

str(migration_matrix) to check the structure of the matrix, and it returned as a character matrix. It becomes a character matrix because when combining different types of elements, they will be coerced into the most flexible type possible. Since characters and numeric variables are present, the matrix turns into character because they are most flexible.

The ordering for coercion is logical < integer < numeric < character, where character is most flexible.

Create a new Data Frame

```
age <- c(16,17,18,22,26,28,30,34,35,41)
satisfaction_levels <- factor(c("Very Dissatisfied", "Dissatisfied", "Dissatisfied","Neutra
l",
                                "Neutral", "Neutral", "Neutral", "Satisfied", "Satisfied", "V
ery Satisfied" ),
                              levels = c("Very Dissatisfied", "Dissatisfied", "Neutral", "Sat
isfied", "Very Satisfied"),
                              ordered = TRUE)

age <- as.integer(age)

df <- data.frame(age, satisfaction_levels)

str(df)
```

```
## 'data.frame':    10 obs. of  2 variables:
##  $ age                : int  16 17 18 22 26 28 30 34 35 41
##  $ satisfaction_levels: Ord.factor w/ 5 levels "Very Dissatisfied"<..: 1 2 2 3 3 3 3 4 4 5
```

```
levels(df$satisfaction_levels)
```

```
## [1] "Very Dissatisfied" "Dissatisfied"      "Neutral"
## [4] "Satisfied"         "Very Satisfied"
```

```
pay_per_hour <- c(10,14,15,26,28,32,35,40,56,65)

df <- cbind(df,pay_per_hour)

print(df)
```

```
##    age satisfaction_levels pay_per_hour
## 1   16   Very Dissatisfied           10
## 2   17        Dissatisfied           14
## 3   18        Dissatisfied           15
## 4   22             Neutral           26
## 5   26             Neutral           28
## 6   28             Neutral           32
## 7   30             Neutral           35
## 8   34           Satisfied           40
## 9   35           Satisfied           56
## 10  41      Very Satisfied           65
```

I created a vector of 10 elements called age. This defaults as numeric, hence I converted by using as.integer. This will represent the integer variable.

I also created a vector called satisfaction_level that ranges from very dissatisfied to very satisfied.Simultaneously, I factored it using the factor function and ordered it. This will represent the ordinal variable and was factorised and ordered. I then created a dataframe with age and satisfaction level.

This was then inputted into str(). Here it shows that it was correctly formatted with 2 variables and 10 observations. The variables returned as numeric and ordered factor with levels as we wanted. I use levels() to depict the range of satisfaction I had created. I then create another vector regarding pay per hour. This would then be combined into the dataframe using cbind.

Printing the dataframe shows that there are 3 variables and they correspond to a made up dataframe that could somewhat be realistic.

References

Australian Bureau of Statistics. (2022-23-financial-year). Overseas Migration. ABS. https://www.abs.gov.au/statistics/people/population/overseas-migration/latest-release (https://www.abs.gov.au/statistics/people/population/overseas-migration/latest-release)