NHANES Data Cleaning & Preparation

Exercise 1: Writing a report in R Markdown

Data Import and Preliminary Exploration

The analysis began by importing six NHANES datasets: three blood pressure files and three demographic files covering the 2013–2018 cycles. Initial exploration showed a mix of continuous variables (e.g., blood pressure, age) and categorical codes (e.g., gender, marital status). Some categorical variables were stored as numeric, such as RIAGENDR (gender) and DMDMARTL (marital status). These were later recoded into factors to ensure proper interpretation.

Merging and Preparing Datasets

Each wave of data was trimmed to retain only relevant columns and a Wave_Id identifier was added. The blood pressure files were stacked into a single dataset, as were the demographic files. These two master datasets were then merged using a full join on SEQN. The merged dataset contained 29,400 rows and 19 columns, which is larger than the standalone blood pressure dataset because the join preserved all participants with demographic data, even if no blood pressure readings were available.

Data Cleaning and Recoding

Next, variable names were standardized and several key variables were recoded into meaningful categories. Gender, ethnicity, education, marital status, family size, family income, and blood pressure validity were converted from numeric codes into descriptive labels. Factors were used for categorical variables, while age and family size were ensured to be numeric. An age group variable was also created to categorize participants into children, adolescents, young adults, middle-aged adults, and older adults. Duplicate rows were removed, and the dataset was exported for further use.

Filtering and Reshaping

For blood pressure analysis, the dataset was restricted to adults aged 21 and older. Only patient ID, age, gender, and the four systolic/diastolic readings were retained. At this point, the dataset was in wide format because multiple readings were stored across separate columns. To facilitate repeated-measures analysis, the data were reshaped into long format, giving each measurement its own row with an indicator for the reading number.

Exercise 2: Data import and preliminary exploration

The blood pressure datasets for 2013–2014, 2015–2016, and 2017–2018 have 9,813 * 23, 9,544 * 21, and 8,704 * 21 dimensions respectively, while the demographic datasets have 10,175 * 47, 9,971 * 47, and 9,254 * 46 dimensions. Across both types of files, the variables cover a mix of numeric health measures (such as systolic and diastolic blood pressure readings), categorical indicators (such as gender, race, and marital status), and administrative identifiers (such as respondent ID). Taken together, these datasets include continuous variables like age and blood pressure, as well as coded categorical variables that describe demographic and health characteristics.

```
## Dimensions of Blood Pressure 17-18:  8704 21
```

```
##
## Dimensions of Blood Pressure 15-16:  9544 21
```

```
##
## Dimensions of Blood Pressure 13-14:  9813 23
```

```
##
##
## Dimensions of Demographics 17-18:  9254 46
```

```
##
## Dimensions of Demographics 15-16:  9971 47
```

```
##
## Dimensions of Demographics 13-14:  10175 47
```

Not all variables are stored in the most appropriate format. For example, the gender variable (RIAGENDR) is coded as numeric but actually represents categories (male/female), so it would be better stored as a factor variable for interpretability and to prevent mistaken numeric operations. Similarly, marital status (DMDMARTL) is also recorded as numeric codes even though it denotes distinct categories such as married, single, or divorced. Treating these as numeric risks misleading statistical summaries, whereas representing them as categorical factors would align with their intended meaning and facilitate clearer analysis.

Exercise 3: Merging and preparing datasets

Selecting Columns From Blood Pressure Data & Adding Wave ID

```r
b1 <- b1 %>%
  select(SEQN, PEASCCT1, BPXSY1, BPXDI1, BPXSY2, BPXDI2, BPXSY3,
         BPXDI3, BPXSY4, BPXDI4) %>%
  mutate(Wave_Id = "17-18")

b2 <- b2 %>%
  select(SEQN, PEASCCT1, BPXSY1, BPXDI1, BPXSY2, BPXDI2, BPXSY3,
         BPXDI3, BPXSY4, BPXDI4)  %>%
  mutate(Wave_Id = "15-16")

b3 <- b3 %>%
  select(SEQN, PEASCCT1, BPXSY1, BPXDI1, BPXSY2, BPXDI2, BPXSY3,
         BPXDI3, BPXSY4, BPXDI4)  %>%
  mutate(Wave_Id = "13-14")
```

Selecting Columns From Demographic Data & Adding Wave ID

```
d1 <- d1 %>%
  select(SEQN, RIAGENDR, RIDAGEYR, RIDRETH1, RIDRETH3, DMDEDUC2,
         DMDMARTL, DMDFMSIZ, INDFMIN2)  %>%
  mutate(Wave_Id = "17-18")

d2 <- d2 %>%
  select(SEQN, RIAGENDR, RIDAGEYR, RIDRETH1, RIDRETH3, DMDEDUC2,
         DMDMARTL, DMDFMSIZ, INDFMIN2)  %>%
  mutate(Wave_Id = "15-16")

d3 <- d3 %>%
  select(SEQN, RIAGENDR, RIDAGEYR, RIDRETH1, RIDRETH3, DMDEDUC2,
         DMDMARTL, DMDFMSIZ, INDFMIN2)  %>%
  mutate(Wave_Id = "13-14")
```

Merging Dataframes

```
# Now bind all together, filling NAs for missing columns
master_b <- bind_rows(b1, b2, b3)
master_d <- bind_rows(d1, d2, d3)


dim(master_b)
```

```
## [1] 28061    11
```

```
dim(master_d)
```

```
## [1] 29400    10
```

Joining Dataframes

```
final_merged <- full_join(master_b, master_d, by = "SEQN")
final_merged <- final_merged %>%
  select(-Wave_Id.y)
dim(final_merged)
```

```
## [1] 29400    19
```

The final merged dataset contains 29,400 rows and 19 columns, representing the combined information from both the blood pressure and demographic datasets. The blood pressure dataset alone had 28,061 rows and 11 columns, while the demographic dataset had 29,400 rows and 10 columns. The number of rows in the final dataset matches the demographic dataset because a full join was used, meaning all individuals from both datasets were retained, even if they were only present in one. This explains why the final dataset has more rows than the blood pressure dataset: it includes participants who had demographic information but no blood pressure data. Additionally, the number of columns increased to 19 because it combines variables from both datasets while ensuring all patient IDs are represented.

Exercise 4: Data cleaning and variable recoding

Using Skim To Review Data

```
# Run skim safely
#skim_summary <- skimr::skim(final_merged)

# Just print in console (no Unicode)
#skim_summary
```

Cleaning Column Names

```
final_merged <- final_merged %>%
  janitor::clean_names()

final_merged <- final_merged %>%
  rename(
    seqn = seqn,
    gender = riagendr,
    age_years = ridageyr,
    ethnicity_primary = ridreth1,
    ethnicity_detailed = ridreth3,
    education_level = dmdeduc2,
    marital_status = dmdmartl,
    family_size = dmdfmsiz,
    family_income = indfmin2,
    bp_validity = peascct1,   # validity comments
    sbp1 = bpxsy1, dbp1 = bpxdi1,
    sbp2 = bpxsy2, dbp2 = bpxdi2,
    sbp3 = bpxsy3, dbp3 = bpxdi3,
    sbp4 = bpxsy4, dbp4 = bpxdi4
  )
```

Recoding Variables To Meaningful Values

```
final_merged <- final_merged %>%
  mutate(
    # Gender
    gender = recode(as.character(gender),
      "1" = "Male",
      "2" = "Female"
    ),

    # Ethnicity - primary (RIDRETH1)
    ethnicity_primary = recode(as.character(ethnicity_primary),
      "1" = "Mexican American",
      "2" = "Other Hispanic",
      "3" = "Non-Hispanic White",
      "4" = "Non-Hispanic Black",
      "5" = "Other Race - Including Multi-Racial"
    ),

    # Ethnicity - detailed (RIDRETH3)
    ethnicity_detailed = recode(as.character(ethnicity_detailed),
      "1" = "Mexican American",
      "2" = "Other Hispanic",
      "3" = "Non-Hispanic White",
      "4" = "Non-Hispanic Black",
      "6" = "Non-Hispanic Asian",
      "7" = "Other Race - Including Multi-Racial"
    ),

    # Education (DMDEDUC2)
    education_level = recode(as.character(education_level),
      "1" = "Less than 9th grade",
      "2" = "9-11th grade (Includes 12th grade with no diploma)",
      "3" = "High school graduate/GED or equivalent",
      "4" = "Some college or AA degree",
      "5" = "College graduate or above",
      "7" = "Refused",
      "9" = "Don't Know"
    ),

    # Marital status (DMDMARTL)
    marital_status = recode(as.character(marital_status),
      "1" = "Married",
      "2" = "Widowed",
      "3" = "Divorced",
      "4" = "Separated",
      "5" = "Never married",
      "6" = "Living with partner",
      "77" = "Refused",
      "99" = "Don't Know"
    ),

    # Family size (DMDFMSIZ)
    family_size = recode(as.character(family_size),
      "1" = "1",
      "2" = "2",
      "3" = "3",
      "4" = "4",
      "5" = "5",
      "6" = "6",
      "7" = "7 or more people in the Family"
    ),

    # Family income (INDFMIN2)
    family_income = recode(as.character(family_income),
      "1"  = "$0 to $4,999",
      "2"  = "$5,000 to $9,999",
      "3"  = "$10,000 to $14,999",
      "4"  = "$15,000 to $19,999",
      "5"  = "$20,000 to $24,999",
      "6"  = "$25,000 to $34,999",
      "7"  = "$35,000 to $44,999",
      "8"  = "$45,000 to $54,999",
      "9"  = "$55,000 to $64,999",
      "10" = "$65,000 to $74,999",
      "12" = "$20,000 and Over",
      "13" = "Under $20,000",
      "14" = "$75,000 to $99,999",
      "15" = "$100,000 and Over",
      "77" = "Refused",
      "99" = "Don't Know"
    ),

    # BP validity (PEASCCT1)
    bp_validity = recode(as.character(bp_validity),
      "1" = "Safety exclusion",
      "2" = "SP refusal",
      "3" = "Time constraints",
      "4" = "Other"
    )
  )

final_merged <- final_merged %>%
  mutate(
    gender = factor(gender),
    ethnicity_primary = factor(ethnicity_primary),
    ethnicity_detailed = factor(ethnicity_detailed),
    education_level = factor(education_level),
    marital_status = factor(marital_status),
    family_income = factor(family_income),
    family_size = as.numeric(family_size),
    age_years = as.numeric(age_years)
  )
```

Creating Age Group Column

```
final_merged <- final_merged %>%
  mutate(age_group = case_when(
    age_years < 10 ~ "Child (0-9)",
    age_years >= 10 & age_years < 20 ~ "Adolescent (10-19)",
    age_years >= 20 & age_years < 40 ~ "Young Adult (20-39)",
    age_years >= 40 & age_years < 60 ~ "Middle-aged (40-59)",
    age_years >= 60 ~ "Older Adult (60+)",
    TRUE ~ NA_character_
  ))
```

Removing Duplicates

```
final_merged <- final_merged %>%
  distinct()
```

Cleaning Bp Validity Column

```
final_merged <- final_merged %>%
  # Clean up bp_validity first
  mutate(
    bp_validity_clean = case_when(
      bp_validity %in% c("1", "Safety exclusion") ~ "Safety exclusion",
      bp_validity %in% c("2", "SP refusal") ~ "SP refusal",
      bp_validity %in% c("3", "Time constraints") ~ "Time constraints",
      bp_validity %in% c("4", "Other") ~ "Other",
      TRUE ~ NA_character_
    )
  )
```

Exporting Cleaned Dataset

```
export(final_merged, "cleaned_nhanes.csv")
```

Exercise 5: Filtering and reshaping a dataset

Using Filter To Keep Adults Only

```
adult_bp <- final_merged %>%
  filter(!is.na(age_years), age_years >= 21)
```

Removing All Other Columns Except Selected Ones

```
adult_bp <- adult_bp %>%
  select(seqn, age_years, gender, sbp1, dbp1, sbp2,
         dbp2, sbp3, dbp3, sbp4, dbp4)
```

Converting To Long Format

Right now, the dataset has multiple readings (sbp1–sbp4, dbp1–dbp4) in separate columns for each person. This is wide format, because repeated measures (blood pressure readings) are spread across columns. Wide format = each subject has one row, with repeated measures stored in multiple columns. Long format = each subject has multiple rows, one per measurement, with a column indicating measurement number.

```
adult_bp_long <- adult_bp %>%
  tidyr::pivot_longer(
    cols = starts_with("sbp") | starts_with("dbp"),
    names_to = c(".value", "reading"),
    names_pattern = "([a-z]+)([0-9]+)"
  )
```

Long Format

```
head(adult_bp_long)
```

```
## # A tibble: 6 x 6
##    seqn age_years gender reading   sbp   dbp
##   <dbl>     <dbl> <fct>  <chr>   <dbl> <dbl>
## 1 93705        66 Female 1          NA    NA
## 2 93705        66 Female 2          NA    NA
## 3 93705        66 Female 3         202    62
## 4 93705        66 Female 4         198    74
## 5 93708        66 Female 1          NA    NA
## 6 93708        66 Female 2         138    78
```