```
-- STEP 0: CHECK FOR MISSING DATA OR EXTREME VALUES
-- AND HANDLE THEM
-- ======================================================


-- In this case, this dataset is "ideal"


-- PART 1a: EXPLORE THE TYPES OF DATA WE HAVE IN EACH VARIABLE
-- (COLUMN)
-- ======================================================


\d diabetes
-- We have categorical and numerical variables. Although some numeric
-- variables are actually categorical,


Explore categorical variables (even those that appear to be numeric)
SELECT DISTINCT(gender) FROM diabetes_preproc; -- Other, Male, Female
SELECT DISTINCT(hypertension) FROM diabetes_preproc; -- 0, 1
SELECT DISTINCT(heart_disease) FROM diabetes_preproc; -- 0, 1
SELECT DISTINCT(diabetes) FROM diabetes_preproc; -- True/False


-- PART 1b: DELETE IRRELEVANT COLUMNS AND CREATE A NEW TABLE
-- ======================================================


-- Delete irrelevant columns:
-- year and location
CREATE TABLE diabetes_preproc AS
SELECT gender, age, hypertension, heart_disease, bmi, hbA1c_level,
blood_glucose_level, diabetes
FROM diabetes;


\d diabetes_preproc
```

```
-- PART 2: DATA TYPE CONVERSION AND VARIABLE TRANSFORMATION
-- ======================================================

-- Convert the predictor variable (diabetes) from True/False (character) to
-- 1/0 (numeric)

-- Update table
UPDATE diabetes_preproc
SET diabetes =
CASE
WHEN diabetes = 'True' THEN '1.0'
WHEN diabetes = 'False' THEN '0.0'
END;

SELECT diabetes FROM diabetes_preproc;

-- And change the data type: '1.0'/'0.0' from character to float
ALTER TABLE diabetes_preproc
ALTER COLUMN diabetes SET DATA TYPE FLOAT
USING diabetes::FLOAT;

-- Transform the "gender" column to one-hot format and remove the original
-- gender

-- Add new columns
ALTER TABLE diabetes_preproc ADD COLUMN gender_other FLOAT DEFAULT 0;
ALTER TABLE diabetes_preproc ADD COLUMN gender_female FLOAT DEFAULT 0;
ALTER TABLE diabetes_preproc ADD COLUMN gender_male FLOAT DEFAULT 0;

-- And populate columns
UPDATE diabetes_preproc
SET gender_other =
```

```
CASE
WHEN gender = 'Other' THEN 1 ELSE 0 END,
gender_female =
CASE
WHEN gender = 'Female' THEN 1 ELSE 0 END,
gender_male =
CASE
WHEN gender = 'Male' THEN 1 ELSE 0 END;

-- And delete original "gender" column
ALTER TABLE diabetes_preproc
DROP COLUMN gender;

SELECT gender_other, gender_female, gender_male FROM diabetes_preproc;

-- Convert columns from numeric type to FLOAT:
ALTER TABLE diabetes_preproc
ALTER COLUMN age SET DATA TYPE FLOAT USING age::FLOAT,
ALTER COLUMN hypertension SET DATA TYPE FLOAT USING hypertension::FLOAT,
ALTER COLUMN heart_disease SET DATA TYPE FLOAT USING heart_disease::FLOAT,
ALTER COLUMN bmi SET DATA TYPE FLOAT USING bmi::FLOAT,
ALTER COLUMN hba1c_level SET DATA TYPE FLOAT USING hba1c_level::FLOAT,
ALTER COLUMN blood_glucose_level SET DATA TYPE FLOAT USING
blood_glucose_level::FLOAT;

-- PART 3: SCALING NUMERIC VARIABLES
-- TO THE RANGE OF 0 TO 1
-- ======================================================

-- Calculating and Storing Scaling Factors
WITH scaling AS (
```

```
SELECT
MIN(bmi) AS min_bmi, MAX(bmi) as max_bmi,
MIN(hbA1c_level) AS min_hg, MAX(hbA1c_level) AS max_hg,
MIN(blood_glucose_level) as min_gluc, MAX(blood_glucose_level) as max_gluc
FROM diabetes_preproc
)
SELECT * FROM scaling;

CREATE TABLE scaling_factors AS
WITH escalation AS (
SELECT
MIN(age) AS min_age, MAX(age) AS max_age,
MIN(bmi) AS min_bmi, MAX(bmi) AS max_bmi,
MIN(hbA1c_level) AS min_hg, MAX(hbA1c_level) AS max_hg,
MIN(blood_glucose_level) AS min_gluc, MAX(blood_glucose_level) AS max_gluc
FROM diabetes_preproc
)
SELECT * FROM escalation;

-- Scale numeric variables
WITH escalation AS (
SELECT
MIN(age) AS min_age, MAX(age) AS max_age,
MIN(bmi) AS min_bmi, MAX(bmi) AS max_bmi,
MIN(hbA1c_level) AS min_hg, MAX(hbA1c_level) AS max_hg,
MIN(blood_glucose_level) AS min_gluc, MAX(blood_glucose_level) AS max_gluc
FROM diabetes_preproc
)
UPDATE diabetes_preproc
SET
age = (age - min_age)/(max_age - min_age),
```

```
bmi = (bmi - min_bmi)/(max_bmi-min_bmi),
hbA1c_level = (hbA1c_level - min_hg)/(max_hg-min_hg),
blood_glucose_level = (blood_glucose_level-min_gluc)/(max_gluc-min_gluc)
FROM escalation;

SELECT age,
       bmi,
       hbA1c_level,
       blood_glucose_level
FROM   diabetes_preproc;
```