```r
library(DBI)
library(RSQLite)
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
```

It is very easy to read all the data from a SQLite database, which can be accomplished with just one line of code:

```r
conn <- dbConnect(RSQLite::SQLite(), "diabetes.db")
```

However, it is often quicker to read from CSV files, which can be done using the following command:

```r
csv_data <- read.csv("diabetes_dataset.csv")
```

If we have multiple tables instead of just a single table, SQLite becomes significantly more useful. This ability stems from the fact that SQL allows for straightforward joining of tables. While packages like Pandas (in Python) can perform similar tasks, they often require learning additional syntax and nuances, making it cumbersome for those more accustomed to SQL.

CSV files are inherently flat and not designed for relational operations like joins, making SQL databases superior for complex datasets where relationships among multiple tables are involved. For example, querying a specific table in SQLite can be very easy:

```r
#query = "SELECT * FROM diabetes"
#data <- dbGetQuery(conn, query)
```

Put data in right form for a model.

```r
csv_data$diagnosed_diabetes <- as.factor(csv_data$diagnosed_diabetes)
```

Model with all the predictors. Notice the only predictor that seems to be useful in this model is hba1c, with genderOther, education_levelNo formal, employment_statusStudent, and diastolic_bp being slightly significant.

```r
full_model <- glm(diagnosed_diabetes ~ ., data = csv_data, family = binomial)
summary(full_model)
```

```
##
## Call:
## glm(formula = diagnosed_diabetes ~ ., family = binomial, data = csv_data)
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        -3.169e+01  6.820e+00  -4.647 3.37e-06 ***
## age                                -1.078e-02  7.057e-02  -0.153   0.8786
## genderMale                          4.240e-01  6.458e-01   0.657   0.5114
## genderOther                        -4.445e+00  1.779e+00  -2.498   0.0125 *
## ethnicityBlack                      1.469e-02  7.036e-01   0.021   0.9833
## ethnicityHispanic                  -4.112e-01  6.943e-01  -0.592   0.5536
## ethnicityOther                       -8.991e-01  9.359e-01  -0.961   0.3367
## ethnicityWhite                       -3.557e-01  6.134e-01  -0.580   0.5620
## education_levelHighschool             5.833e-01  4.013e-01   1.453   0.1461
## education_levelNo formal              1.622e+00  7.376e-01   2.199   0.0279 *
## education_levelPostgraduate           7.060e-01  5.207e-01   1.356   0.1752
## income_levelLow                      -4.702e-01  9.166e-01  -0.513   0.6080
## income_levelLower-Middle             -6.408e-01  8.733e-01  -0.734   0.4631
## income_levelMiddle                   -6.771e-01  7.952e-01  -0.851   0.3945
## income_levelUpper-Middle             -3.976e-01  8.277e-01  -0.480   0.6310
## employment_statusRetired              4.795e-01  4.487e-01   1.069   0.2853
## employment_statusStudent              1.827e+00  7.909e-01   2.310   0.0209 *
## employment_statusUnemployed           3.932e-01  5.305e-01   0.741   0.4586
## smoking_statusFormer                  4.770e-01  5.781e-01   0.825   0.4094
## smoking_statusNever                   3.626e-03  4.482e-01   0.008   0.9935
## alcohol_consumption_per_week          1.510e-01  1.359e-01   1.111   0.2666
## physical_activity_minutes_per_week   5.110e-03  8.472e-03   0.603   0.5464
## diet_score                            1.242e-01  1.480e-01   0.839   0.4013
## sleep_hours_per_day                   2.768e-02  1.623e-01   0.171   0.8646
## screen_time_hours_per_day             1.253e-02  9.719e-02   0.129   0.8974
## family_history_diabetes              -2.299e+00  3.773e+00  -0.609   0.5423
## hypertension_history                  8.128e-01  5.294e-01   1.535   0.1247
## cardiovascular_history               -2.833e-02  9.717e-01  -0.029   0.9767
## bmi                                  -1.952e-01  1.252e-01  -1.559   0.1189
## waist_to_hip_ratio                    6.520e+00  5.953e+00   1.095   0.2734
## systolic_bp                           1.938e-02  1.662e-02   1.166   0.2435
## diastolic_bp                         -4.715e-02  2.232e-02  -2.112   0.0347 *
## heart_rate                           -1.517e-02  2.071e-02  -0.733   0.4637
## cholesterol_total                     8.529e-04  1.581e-02   0.054   0.9570
## hdl_cholesterol                       4.028e-02  3.252e-02   1.239   0.2155
## ldl_cholesterol                      -1.700e-03  1.713e-02  -0.099   0.9210
## triglycerides                        -1.821e-04  5.434e-03  -0.034   0.9733
## glucose_fasting                      -1.955e-03  2.044e-02  -0.096   0.9238
## glucose_postprandial                 -2.782e-03  1.554e-02  -0.179   0.8579
## insulin_level                        -6.629e-03  3.661e-02  -0.181   0.8563
## hba1c                                 4.066e+00  7.464e-01   5.448 5.08e-08 ***
## diabetes_risk_score                   2.065e-01  2.341e-01   0.882   0.3776
## diabetes_stageNo Diabetes            -2.429e+01  3.533e+03  -0.007   0.9945
## diabetes_stagePre-Diabetes           -2.775e+01  1.774e+03  -0.016   0.9875
## diabetes_stageType 1                  2.850e-01  5.051e-01   0.564   0.5726
## diabetes_stageType 2                  2.980e+01  8.569e+02   0.035   0.9723
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 134603.96  on 99999  degrees of freedom
## Residual deviance:    255.68  on 99954  degrees of freedom
## AIC: 347.68
##
## Number of Fisher Scoring iterations: 25
```

```r
null_model <- glm(diagnosed_diabetes ~ 1, data = csv_data, family = binomial)
stepwise_model <- step(null_model,
                       direction = "both",
                                    scope = list(lower = null_model, upper = full_model))
```

Querying certain predictors from the dataset using SQL is straightforward and provides flexibility. Below is an example of selecting specific predictors from the diabetes table:

```
query = "SELECT diagnosed_diabetes, diabetes_stage, hba1c, diabetes_risk_score,
systolic_bp, bmi, diastolic_bp, gender, employment_status, education_level,
hypertension_history FROM diabetes"
sql_data <- dbGetQuery(conn, query)
sql_data$diagnosed_diabetes <- as.factor(sql_data$diagnosed_diabetes)
```

Similarly, we can select specific columns from a CSV dataset using R's native indexing and similar in Pandas to get the same data:

```
csv_sub_data = csv_data[c("diagnosed_diabetes", "diabetes_stage", "hba1c",
"diabetes_risk_score", "systolic_bp", "bmi", "diastolic_bp", "gender",
"employment_status", "education_level", "hypertension_history")]
```

Model only containing the significant predictors from last test.

```
simple_model <- glm(diagnosed_diabetes ~ ., data = sql_data, family = binomial)
summary(simple_model)
```

```
##
## Call:
## glm(formula = diagnosed_diabetes ~ ., family = binomial, data = sql_data)
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -23.04294    3.10743  -7.415 1.21e-13 ***
## diabetes_stageNo Diabetes      -23.92338 3683.44020  -0.006  0.99482
## diabetes_stagePre-Diabetes     -27.11015 1841.94371  -0.015  0.98826
## diabetes_stageType 1             0.09058    0.46906   0.193  0.84687
## diabetes_stageType 2            28.42382  962.72621   0.030  0.97645
## hba1c                            3.69109    0.38451   9.599  < 2e-16 ***
## diabetes_risk_score              0.06146    0.02367   2.597  0.00941 **
## systolic_bp                      0.02388    0.01428   1.673  0.09440 .
## bmi                             -0.08819    0.04542  -1.942  0.05218 .
## diastolic_bp                    -0.03930    0.02046  -1.921  0.05469 .
## genderMale                       0.54014    0.57666   0.937  0.34893
## genderOther                     -3.64918    1.57050  -2.324  0.02015 *
## employment_statusRetired         0.59261    0.41788   1.418  0.15615
## employment_statusStudent         1.78642    0.69670   2.564  0.01034 *
## employment_statusUnemployed      0.40068    0.49242   0.814  0.41582
## education_levelHighschool        0.52041    0.36287   1.434  0.15153
## education_levelNo formal         1.56636    0.69984   2.238  0.02521 *
## education_levelPostgraduate      0.74029    0.46508   1.592  0.11144
## hypertension_history             0.68444    0.47219   1.450  0.14720
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 134603.96  on 99999  degrees of freedom
## Residual deviance:    267.14  on 99981  degrees of freedom
## AIC: 305.14
```

```
##
## Number of Fisher Scoring iterations: 25
```

```r
anova(simple_model, full_model)
```

```
## Analysis of Deviance Table
##
## Model 1: diagnosed_diabetes ~ diabetes_stage + hba1c + diabetes_risk_score +
##     systolic_bp + bmi + diastolic_bp + gender + employment_status +
##     education_level + hypertension_history
## Model 2: diagnosed_diabetes ~ age + gender + ethnicity + education_level +
##     income_level + employment_status + smoking_status + alcohol_consumption_per_week +
##     physical_activity_minutes_per_week + diet_score + sleep_hours_per_day +
##     screen_time_hours_per_day + family_history_diabetes + hypertension_history +
##     cardiovascular_history + bmi + waist_to_hip_ratio + systolic_bp +
##     diastolic_bp + heart_rate + cholesterol_total + hdl_cholesterol +
##     ldl_cholesterol + triglycerides + glucose_fasting + glucose_postprandial +
##     insulin_level + hba1c + diabetes_risk_score + diabetes_stage
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1     99981     267.14
## 2     99954     255.68 27   11.458   0.9961
```

This shows that most of the predictors seem to have no correlation with the outcome and the model can be hust as accurate with fewer predictors. Notice how the accuracy, precision, and recall results for the simplified model are comparable to the accuracy, precision, and recall results for the full model, further cementing the hypothesis that many of the predictors are not correlated.

```r
# Load necessary libraries
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
# Make predictions
predicted_probs <- predict(simple_model, type = "response")  # Get probabilities
predicted_classes <- ifelse(predicted_probs > 0.5, 1, 0)     # Classify based on a threshold of 0.5

# Create a confusion matrix
confusion_matrix <- confusionMatrix(as.factor(predicted_classes), as.factor(sql_data$diagnosed_diabetes

# Extract Accuracy, Precision, and Recall from the confusion matrix
accuracy <- confusion_matrix$overall['Accuracy']
precision <- confusion_matrix$byClass['Precision'][1]
recall <- confusion_matrix$byClass['Recall'][1]

# Print results
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.99939
```

```r
cat("Precision:", precision, "\n")
```

```
## Precision: 0.9992999
```

```r
cat("Recall:", recall, "\n")
```

```
## Recall: 0.999175
```

While both SQLite and CSV files provide easy access to data, SQLite offers significant advantages when dealing with multiple tables, thanks to its ability to perform efficient joins using SQL queries.