Healthcare data analysis

Marisa Lagoa

2025-11-06

Introduction

This project utilizes a synthetic healthcare dataset from Kaggle to develop and demonstrate practical data analysis skills relevant to the American healthcare context. While working with artificial data, this exercise provides valuable experience in identifying data quality issues, performing exploratory analysis, and understanding healthcare data structures that mirror real-world scenarios.

The analysis explicitly addresses the synthetic nature of the dataset, treating the identification of artificial patterns and limitations as a core component of the data quality assessment process.

Data Processing Pipeline

This analysis employed a hybrid data preparation approach that combined the accessibility of spreadsheet software with the reproducibility of programming tools:

1. **Initial data cleaning in Excel**:
    - Removal of clearly erroneous negative billing values
    - Basic formatting standardization
    - Documentation of initial data quality issues
2. **Data processing in R**:
    - Data type conversion and validation
    - Advanced cleaning procedures
    - Systematic quality assessment
    - Exploratory data analysis and visualization

This combined approach leveraged Excel's accessibility for initial inspection and R's reproducibility for systematic data processing.

Data Loading and Setup

```
library(tidyverse)
library(janitor)
library(skimr)
library(lubridate)
library(moments)
library(ggplot2)
```

```r
library(readxl)
## Load dataset
setwd("C:/Users/mflag/data analysis/datasets")
healthcare_df <- read_excel("healthcare_data_cleaned.xlsx")
```

Summary of the dataset

This section provides an initial overview of the dataset structure, variables, and basic characteristics to establish foundational understanding before proceeding with detailed analysis.

```r
head(healthcare_df, 10)
```

```
## # A tibble: 10 x 16
##    names       Age   Gender `Blood Type` `Medical Condition` `Date of Admission`
##    <chr>       <chr> <chr>  <chr>        <chr>               <chr>
##  1 Bobby Jack~ 52    Female O-           Obesity             2019-05-08
##  2 Leslie Ter~ 24    Male   B+           Cancer              2019-05-08
##  3 Danny Smith 49    Male   B+           Obesity             2019-05-08
##  4 Andrew Wat~ 71    Female B-           Hypertension        2019-05-08
##  5 Adrienne B~ 38    Female O-           Obesity             2019-05-08
##  6 Emily John~ 39    Female AB+          Obesity             2019-05-08
##  7 Edward Edw~ 27    Male   A-           Obesity             2019-05-08
##  8 Christina ~ 28    Male   B+           Cancer              2019-05-08
##  9 Jasmine Ag~ 67    Female B-           Cancer              2019-05-08
## 10 Christophe~ 85    Female B-           Cancer              2019-05-08
## # i 10 more variables: Doctor <chr>, Hospital <chr>,
## #   `Insurance Provider` <chr>, `Billing Amount` <chr>, `Room Number` <chr>,
## #   `Admission Type` <chr>, `Discharge Date` <chr>, `days in` <dbl>,
## #   Medication <chr>, `Test Results` <chr>
```

```r
str(healthcare_df)
```

```
## tibble [1,048,575 x 16] (S3: tbl_df/tbl/data.frame)
##  $ names             : chr [1:1048575] "Bobby Jackson" "Leslie Terry" "Danny Smith" "Andrew Watts" .
##  $ Age               : chr [1:1048575] "52" "24" "49" "71" ...
##  $ Gender            : chr [1:1048575] "Female" "Male" "Male" "Female" ...
##  $ Blood Type        : chr [1:1048575] "O-" "B+" "B+" "B-" ...
##  $ Medical Condition : chr [1:1048575] "Obesity" "Cancer" "Obesity" "Hypertension" ...
##  $ Date of Admission : chr [1:1048575] "2019-05-08" "2019-05-08" "2019-05-08" "2019-05-08" ...
##  $ Doctor            : chr [1:1048575] "Corey Webb" "Jennifer Singleton" "Stephanie Tran" "Mark Brow
##  $ Hospital          : chr [1:1048575] "Moss-Mathews" "Wood PLC" "PLC Mathis" "Smith and Lewis Ross"
##  $ Insurance Provider: chr [1:1048575] "Cigna" "Aetna" "Medicare" "Blue Cross" ...
##  $ Billing Amount    : chr [1:1048575] "26451.7511459464" "3602.6618653778401" "35344.934202039702"
##  $ Room Number       : chr [1:1048575] "169" "124" "285" "138" ...
##  $ Admission Type    : chr [1:1048575] "Emergency" "Elective" "Elective" "Emergency" ...
##  $ Discharge Date    : chr [1:1048575] "2019-05-13" "2019-05-28" "2019-06-01" "2019-05-14" ...
##  $ days in           : num [1:1048575] 5 20 24 6 27 20 1 28 26 12 ...
##  $ Medication        : chr [1:1048575] "Ibuprofen" "Aspirin" "Lipitor" "Penicillin" ...
##  $ Test Results      : chr [1:1048575] "Abnormal" "Normal" "Normal" "Abnormal" ...
```

Initial observations:

- Mixed data types with several numeric variables stored as character
- Contains demographic, clinical, and administrative variables
- Shows diverse medical conditions across different age groups

```
skim_without_charts(healthcare_df)
```

Table 1: Data summary

Name	healthcare_df
Number of rows	1048575
Number of columns	16
Column type frequency:	
character	15
numeric	1
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
names	993075	0.05	6	24	0	39959	0
Age	993075	0.05	2	2	0	77	0
Gender	993075	0.05	4	6	0	2	0
Blood Type	993075	0.05	2	3	0	8	0
Medical Condition	993075	0.05	6	12	0	6	0
Date of Admission	993075	0.05	10	10	0	1827	0
Doctor	993075	0.05	6	27	0	39939	0
Hospital	993075	0.05	6	35	0	39639	0
Insurance Provider	993075	0.05	5	16	0	5	0
Billing Amount	993075	0.05	2	18	0	49897	0
Room Number	993075	0.05	3	3	0	400	0
Admission Type	993075	0.05	6	9	0	3	0
Discharge Date	993075	0.05	10	10	0	1856	0
Medication	993075	0.05	7	11	0	5	0
Test Results	993075	0.05	6	12	0	3	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
days in	993075	0.05	15.51	8.66	1	8	15	23	30

- 15 character variables including identifiers, dates, and categorical data
- 1 numeric variable (`days_in`) representing length of hospital stay
- `Billing Amount`, `Room Number, Age` and dates require type conversion for proper analysis
- Incorrect min and max in `Billing Amount`, `Room Number, Age,` because these variables are categorized as characters and skim displays string lengths rather than actual values.

- Extensive missing data: 993075 empty rows (95% of dataset)
- Huge amount of unique patient names (39959) and hospital names (39639)
- Only 6 medical conditions and 5 medication types
- Admissions span approximately 5 years (1827 unique dates)

Data Quality Assessment

Issue Documentation

Data quality issues were systematically documented throughout the cleaning process, covering several distinct problems across multiple variables. The documentation of the cleaning process made in excel is available in the 'Issue_log' worksheet of `healthcare_data_cleaned.xlsx`, tracking 14 quality issues with scope, magnitude, and resolution methods for each.

Unresolved Issues for R Analysis

The following data quality issues, identified as synthetic data artifacts, were preserved for further analytical exploration in R:

- **Patients and hospital naming inconsistencies**: Artificial variation in hospital names (e.g., random commas between names) that does not reflect real-world naming conventions.
- **Medication-condition misalignment**: Illogical drug-prescription patterns that are inconsistent with typical medical practice.
- **Unexpected billing values**: Few low-value billing amounts and a concentration of high-value amounts, which deviate from the expected distribution of healthcare costs.

These unresolved issues will be investigated further during the exploratory data analysis to understand their impact on the analysis and to practice identifying such patterns in synthetic data.

Data Cleaning

Cleaning name columns

Renaming columns for a standard pattern (snake_case)

```
#Column name standardization
healthcare_df <- clean_names(healthcare_df)
```

Removing empty rows

```
#Number of rows before cleaning
print(paste("Number of rows before cleaning:", nrow(healthcare_df)))
```

```
## [1] "Number of rows before cleaning: 1048575"
```

```r
# removing empty rows
healthcare_df <- healthcare_df %>%
filter(if_any(everything(), ~ !is.na(.)))

print(paste("Number of rows after removing empty rows", nrow(healthcare_df)))
```

```
## [1] "Number of rows after removing empty rows 55500"
```

```r
print(paste("Reduction of", round((1 - nrow(healthcare_df)/1048575)*100, 1), "%"))
```

```
## [1] "Reduction of 94.7 %"
```

The dataset imported from excel had 993075 out of 1 048 575 empty rows which indicate an artifact of excel export. After removing them, the dataset now contains 55 500 rows (5% of the original) for analysis.

Correcting data types

Several variables were imported as characters and need to be converted to appropriate types.

- age: converted to numeric
- billing_amount: converted to numeric
- room_number: converted to numeric
- date_of_admission and discharge_date: converted to date

```r
#Data type conversion
healthcare_df <- healthcare_df %>%
  mutate(
    age = as.numeric(age),
    billing_amount = as.numeric(billing_amount),
    room_number = as.numeric(room_number),
    date_of_admission = as.Date(date_of_admission),
    discharge_date = as.Date(discharge_date)
  )
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `billing_amount = as.numeric(billing_amount)`.
## Caused by warning:
## ! NAs introduced by coercion
```

When converting the variable `billing_amount` to numeric, a warning appeared: "NAs introduced by coercion". This happens because some numeric values were manually coded as "NA" in the Excel file. This warning is expected and acceptable because R converted them to proper missing values (NA), which ensures consistent handling of null entries.

Rounding numeric values

I rounded the `billing_amount` variable to two decimal places to improve readability and consistency across records.

```r
#round to 2 decimals
healthcare_df <- healthcare_df %>%
  mutate(billing_amount = round(billing_amount, 2))
```

Summary after cleaning

I verified that all data cleaning steps were completed successfully by examining the dataset structure and key variables.

```r
# Basic dataset dimensions
cat("Dataset dimensions after cleaning:\n")
```

```
## Dataset dimensions after cleaning:
```

```r
cat("- Rows:", nrow(healthcare_df), "\n")
```

```
## - Rows: 55500
```

```r
cat("- Columns:", ncol(healthcare_df), "\n")
```

```
## - Columns: 16
```

```r
#  Check if data types were correctly converted
str(healthcare_df[c("age", "billing_amount", "date_of_admission", "discharge_date", "room_number")])
```

```
## tibble [55,500 x 5] (S3: tbl_df/tbl/data.frame)
##  $ age              : num [1:55500] 52 24 49 71 38 39 27 28 67 85 ...
##  $ billing_amount   : num [1:55500] 26452 3603 35345 33787 30663 ...
##  $ date_of_admission: Date[1:55500], format: "2019-05-08" "2019-05-08" ...
##  $ discharge_date   : Date[1:55500], format: "2019-05-13" "2019-05-28" ...
##  $ room_number      : num [1:55500] 169 124 285 138 253 487 294 395 194 120 ...
```

```r
# missing values per variable
healthcare_df %>%
  summarise(across(everything(), ~sum(is.na(.)))) %>%
  pivot_longer(everything(), names_to = "variable", values_to = "missing_count")
```

```
## # A tibble: 16 x 2
##    variable          missing_count
##    <chr>                     <int>
##  1 names                         0
##  2 age                           0
##  3 gender                        0
##  4 blood_type                    0
##  5 medical_condition             0
##  6 date_of_admission             0
##  7 doctor                        0
##  8 hospital                      0
```

```
##  9 insurance_provider             0
## 10 billing_amount               116
## 11 room_number                    0
## 12 admission_type                 0
## 13 discharge_date                 0
## 14 days_in                        0
## 15 medication                     0
## 16 test_results                   0
```

```r
# Confirm if missing billing amounts are marked as NA
cat("Missing billing amounts:", sum(is.na(healthcare_df$billing_amount)), "\n")
```

```
## Missing billing amounts: 116
```

The cleaning process worked correctly. All variables have the proper data types, only the variable billing_amount has missing values and they are properly marked as NA. The dataset is now ready for analysis.

Exploratory checks

Investigating Billing Anomalies

As noted in the data quality assessment, the billing distribution shows unexpected patterns.

```r
ggplot(healthcare_df, aes(x = billing_amount)) +
  geom_histogram(bins = 50, fill = "brown", alpha = 0.3, color = "brown",linewidth = 0.7) +
  labs(title = "Distribution of Billing Amounts") +
  scale_x_continuous(breaks = seq(0, 52000, by = 5000)) +
  theme_minimal()
```

```
## Warning: Removed 116 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

The histogram in figure 1 shows a near-uniform distribution of billing amounts across most of the range (0–45000), with similar frequencies in each bin.
However, a noticeable drop-off in frequency occurs in the upper billing range (above ~45000), suggesting a truncated or capped distribution.
This pattern reinforces the synthetic nature of the dataset, as real-world billing data is typically right-skewed and not uniformly distributed.

```r
healthcare_df %>%
  filter(billing_amount > 45000) %>%
  ggplot(aes(x = billing_amount)) +
  geom_density(color = "brown", linewidth = 1.5, fill = "brown", alpha = 0.2) +
  labs( title = "Distribution of High Billing Amounts (>$45000)", subtitle = "Shows artificial clusteri
    x = "Billing Amount ($)", y = "Density" ) +
  theme_minimal()
```

The density scale represents the relative probability of observing values within specific ranges, with the total area under the curve equaling 1 (100% of the distribution). The plot in figure 2 helps put in prespective how small the relative probability density is in the extreme values (0.00000–0.00020). It reveals a compressed distribution with limited variance in high cost cases (>45000), contrasting sharply with real healthcare data, where extreme costs would show much greater dispersion across the billing spectrum.

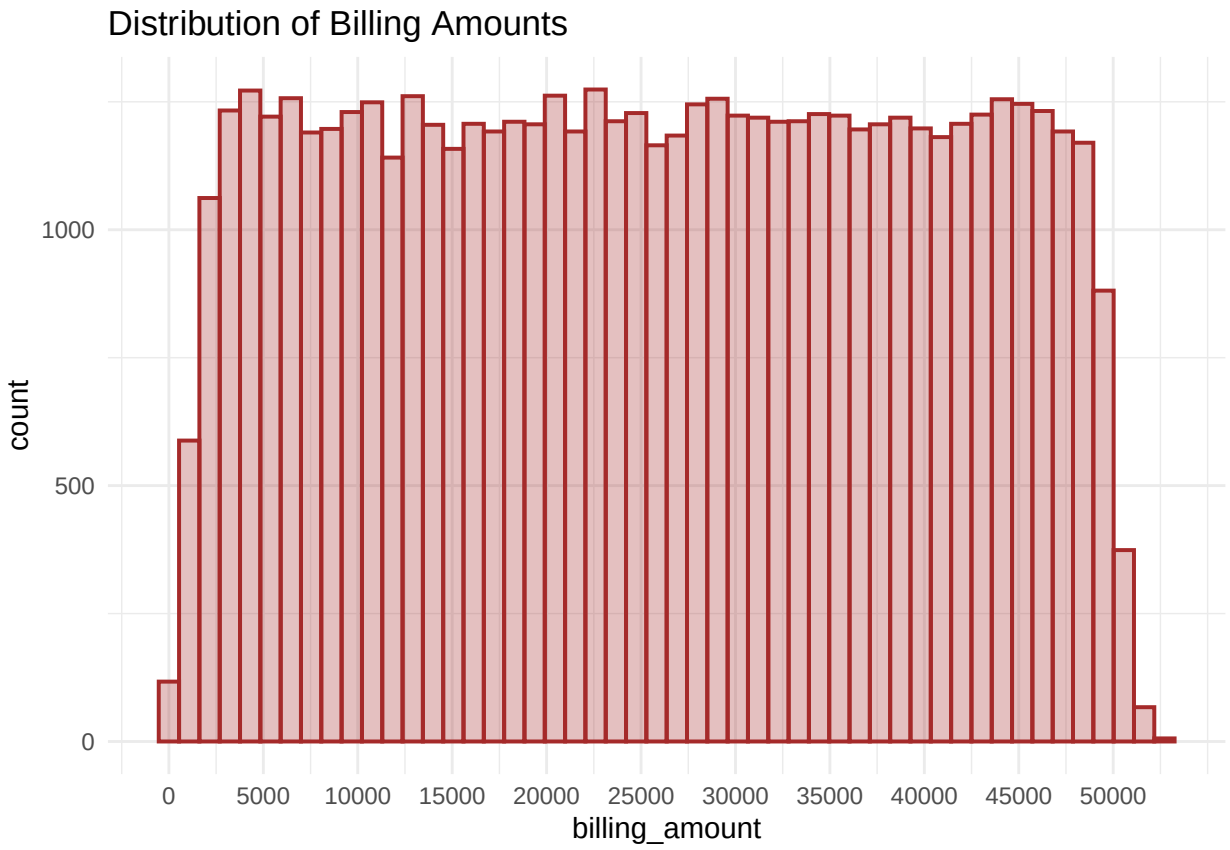

Figure 1: Figure 1 - Histogram with billing_amount distribution

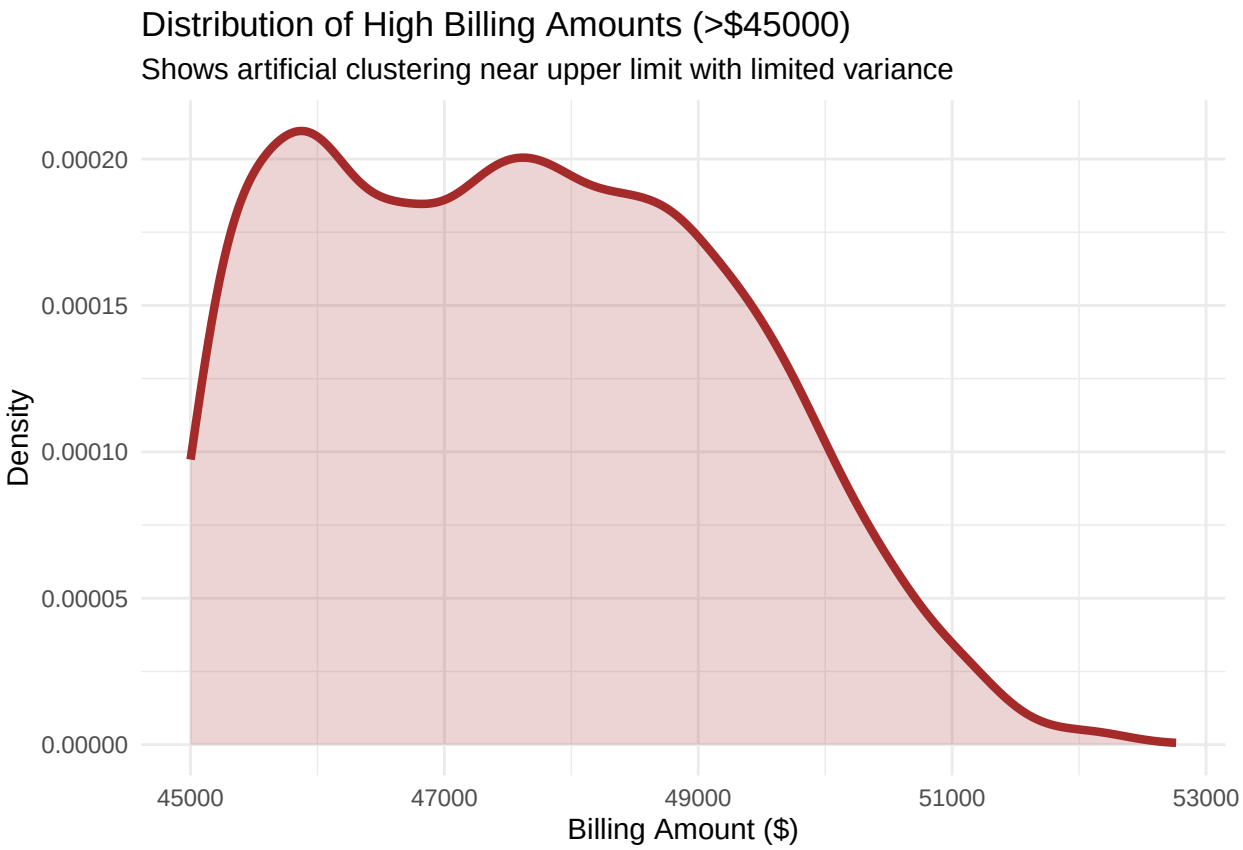

Figure 2: Figure 2 - Distribution of High Billing Amounts (>$45 000)

Billing summary statistics

```
healthcare_df %>%
  summarize(
  billing_amount_mean = mean(billing_amount, na.rm = TRUE),
  billing_amount_max = max(billing_amount, na.rm = TRUE),
  billing_amount_min = min(billing_amount, na.rm = TRUE),
  negative_billing = sum(billing_amount < 0, na.rm = TRUE),
  zero_billing = sum(billing_amount == 0, na.rm = TRUE),
  low_billing_for_admissions = sum(billing_amount < 1000 & days_in > 1, na.rm = TRUE),
  high_billing_count = sum(billing_amount > 50000, na.rm = TRUE),
  missing_billing = sum(is.na(billing_amount))
  ) %>%
  pivot_longer(
    cols = everything(),
    names_to = "quality_issue",
    values_to = "count"
  )
```

```
## # A tibble: 8 x 2
##   quality_issue                count
##   <chr>                        <dbl>
## 1 billing_amount_mean        25594.
## 2 billing_amount_max         52764.
## 3 billing_amount_min            51.3
## 4 negative_billing               0
## 5 zero_billing                   0
## 6 low_billing_for_admissions   302
## 7 high_billing_count           458
## 8 missing_billing              116
```

The quality checks above reveal several indicators of synthetic data generation:

Key Findings:

- **116 missing values** remain after initial cleaning
- **302 cases** show unrealistically low billing (<$1000) for multi-day hospital admissions in american hospitals
- **458 cases** at the upper billing range (>$50000)
- **No zero or negative billing** (addressed during Excel preprocessing)

Critical Analysis:

The concentration of 458 cases near the maximum billing amount (~$52764) and 302 cases with suspiciously low billing for extended hospital stays strongly suggests artificial value generation.

In authentic healthcare data, billing amounts are influenced by multiple clinical and operational factors including procedure complexity, length of stay, medication regimens, and hospital pricing structures. Real-world hospital costs typically exhibit high variability with a pronounced right-skewed distribution—characterized by many low-to-moderate cost admissions and a long tail of increasingly expensive cases admin (2016). This pattern reflects the Pareto principle in healthcare spending, where a small percentage

of complex cases account for the majority of costs. In genuine datasets, extreme values would demonstrate substantial variance, with outliers potentially reaching $100000+, $500000+, or even millions for specialized treatments and prolonged hospitalizations.

However, this synthetic dataset reveals distinctly artificial patterns:

- **Unrealistic distribution**: Figure 1 shows approximately uniform distribution across mid-to-high ranges, contrasting sharply with expected right-skewed patterns
- **Artificial upper boundary**: The abrupt cutoff at ~$52764 suggests predefined constraints rather than natural variance
- **Limited extreme value dispersion**: Figure 2 reveals minimal variation among high-cost cases (>$50000), with only 458 observations clustering near the maximum
- **Data quality artifacts**: The initial presence of 116 negative billing values (converted to NA during excel processing) represents impossible real-world scenarios

These collective anomalies confirm the synthetic nature of this dataset, while providing valuable practice in identifying data quality issues that would be critical when analyzing genuine healthcare financial data.

Examining Naming Inconsistencies

This section investigates naming pattern anomalies identified during data quality assessment, focusing on patient names, doctor names, and hospital naming conventions.

Patient name inconsistencies

```
# Examine patient name inconsistencies
pacient_name_analysis <- healthcare_df %>%
  summarise(
    total_unique_names = n_distinct(names),
    total_records = n(),
    repeated_names = total_records - total_unique_names,
    repeated_names_rate = round(repeated_names / total_records * 100, 2)
  )

cat("Patient Names Analysis:\n",
    "- Total unique names:", pacient_name_analysis$total_unique_names, "\n",
    "- Total records:", pacient_name_analysis$total_records, "\n",
    "- Repeated names:", pacient_name_analysis$repeated_names, "\n",
    "- Repeated names rate:", pacient_name_analysis$repeated_names_rate, "%\n")
```

```
## Patient Names Analysis:
##  - Total unique names: 39959
##  - Total records: 55500
##  - Repeated names: 15541
##  - Repeated names rate: 28 %
```

The naming pattern analysis above reveals several artificial distributions characteristic of synthetic data generation:

- **High repetition rate** of patient names : 28% ; 15541 repeated records

- **Unrealistic distribution** :39959 unique names for 55500 records suggests artificial name assignment

The 28% name duplication rate significantly exceeds expected patterns in authentic healthcare data. In real hospital systems, typical duplication rates range from 8% to 16% in well-managed databases (Just et al. 2016). The observed rate of 28% suggests algorithmic name assignment rather than organic patient registration patterns. Authentic healthcare data would show a more skewed distribution with most names appearing once or twice, and a small number of frequent patients accounting for multiple visits.

Hospital name inconsistencies

```
# Analyze hospital naming patterns
hospital_analysis <- healthcare_df %>%
  group_by(hospital) %>%
  summarise(
    n_patients = n(),
    percentage = round(n() / nrow(healthcare_df) * 100, 2)
  ) %>%
  arrange(desc(n_patients))

#Top 10 Hospitals by Patient Count
print(hospital_analysis %>% head(10))
```

```
## # A tibble: 10 x 3
##    hospital    n_patients percentage
##    <chr>            <int>      <dbl>
##  1 LLC Smith           44       0.08
##  2 Ltd Smith           39       0.07
##  3 Johnson PLC         38       0.07
##  4 Smith Ltd           37       0.07
##  5 Smith Group         36       0.06
##  6 Smith PLC           36       0.06
##  7 Johnson Inc         35       0.06
##  8 Smith Inc           34       0.06
##  9 Group Smith         32       0.06
## 10 Smith LLC           32       0.06
```

```
#Hospital Naming Observations:
cat("- Total unique hospitals:", n_distinct(healthcare_df$hospital), "\n")
```

```
## - Total unique hospitals: 39639
```

```
cat("- Most common hospital:", hospital_analysis$hospital[1],
    "(", hospital_analysis$n_patients[1], "patients )\n")
```

```
## - Most common hospital: LLC Smith ( 44 patients )
```

```
# Plot hospital size distribution
hospital_analysis %>%
  head(10) %>% # Shows top 10 hospitals
  ggplot(aes(x = reorder(hospital, n_patients), y = n_patients)) +
```

```r
  geom_col(fill = "brown", alpha = 0.7) +
  coord_flip() +
  labs( title = "Top 10 Hospitals by Patient Count", subtitle = "Even the 'largest' hospitals serve very
  theme_minimal()
```

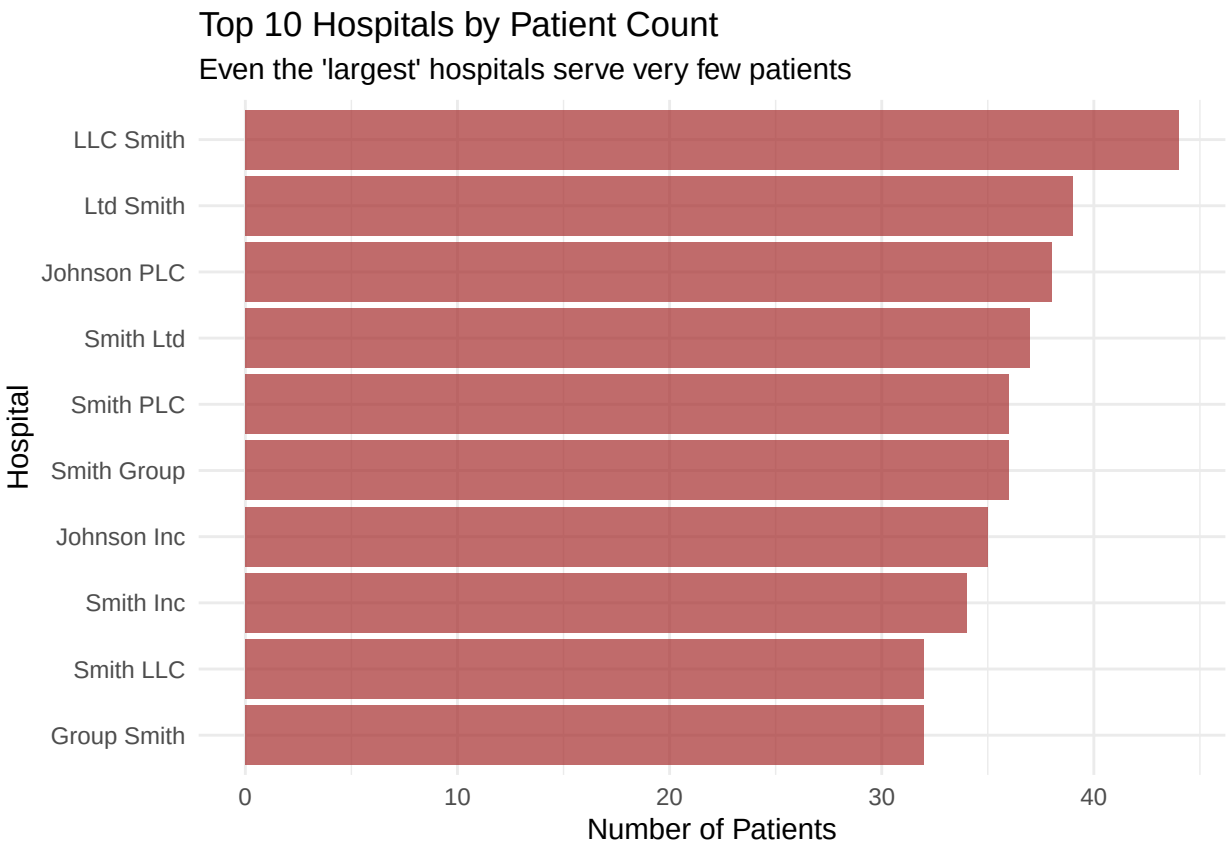

Figure 3: Figure 3 - Top 10 largest hospitals

The hospital name distribution patterns reveal:

Extreme fragmentation- 39639 unique hospitals represents an implausibly high number for serving only 55500 patients. For comparison, the United States healthcare system comprises approximately 6000 hospitals serving over 34 million annual admissions ("Fast Facts on U.S. Hospitals, 2025 | AHA" 2025).

Unrealistic patient distribution- The "largest" hospital (LLC Smith) serves only 44 patients (0.08% of total). The top 10 hospitals in the dataset all maintain artificially similar patient volumes (32-44 patients), whereas authentic hospitals typically serve hundreds to thousands of patients each.

Artificial naming conventions- The dataset shows systematic application of business suffixes (LLC, Ltd, PLC) to common surnames like "Smith" and "Johnson." Genuine hospital names typically reflect geographic locations, saint names, or descriptive clinical specialties (e.g., "Massachusetts General Hospital," "St. Jude Children's Research Hospital," "Memorial Cancer Center").

In conclusion, the hospital naming and distribution analysis provides additional evidence of synthetic data artifacts within the dataset.

Doctor Distribution Patterns

```
# Doctor distribution patterns
doctor_analysis <- healthcare_df %>%
  group_by(doctor) %>%
  summarise(
    n_patients = n()
  ) %>%
  arrange(desc(n_patients))

cat("Doctor distribution patterns analysis:\n",
    "- Total unique doctors:", n_distinct(healthcare_df$doctor), "\n",
    "- Average patients per doctor:", round(mean(doctor_analysis$n_patients), 1), "\n",
    "- Most patients per doctor:", max(doctor_analysis$n_patients), "\n",
    "- Doctors with only 1 patient:", sum(doctor_analysis$n_patients == 1), "\n",
    "- Percentage of doctors with only 1 patient:",
    round((sum(doctor_analysis$n_patients == 1) / nrow(doctor_analysis)) * 100, 1), "%\n")
```

```
## Doctor distribution patterns analysis:
##  - Total unique doctors: 39939
##  - Average patients per doctor: 1.4
##  - Most patients per doctor: 28
##  - Doctors with only 1 patient: 30040
##  - Percentage of doctors with only 1 patient: 75.2 %
```

The doctor distribution shows artificial patterns with 39939 doctors serving 55500 patients—a 1.4 patient-per-doctor ratio that contrasts sharply with real healthcare data because doctors usually have hundreds to thousands of patients. With 75% of doctors treating only one patient and the busiest doctor handling just 28 cases, the data reflects algorithmic assignment rather than clinical practice patterns.

Medication-condition misalignment

While cleaning the dataset in excel, i noticed that often the medical condition of the patient would not make sense with the medication given.

```
# Cmbinations between medication and condition ordered by most often
medication_condition_analysis <- healthcare_df %>%
  count(medical_condition, medication, sort = TRUE) %>%
  head(10)
#Top 10 Medication-Condition Combinations
medication_condition_analysis
```

```
## # A tibble: 10 x 3
##    medical_condition medication      n
##    <chr>             <chr>       <int>
##  1 Cancer            Lipitor      1922
##  2 Arthritis         Aspirin      1918
##  3 Diabetes          Lipitor      1893
##  4 Hypertension      Ibuprofen    1893
##  5 Obesity           Penicillin   1893
```

```
##  6 Asthma            Paracetamol  1888
##  7 Diabetes          Penicillin   1881
##  8 Arthritis         Paracetamol  1877
##  9 Cancer            Ibuprofen    1873
## 10 Arthritis         Penicillin   1866
```

The medication-condition analysis reveals artificially uniform distribution patterns, with illogical drug-prescription combinations (e.g., Lipitor for cancer, Penicillin for obesity) all appearing in nearly identical patient counts (~1900 each). This uniform distribution across clinically unrelated pairs confirms synthetic data generation, as real healthcare data would show specialized treatment patterns with significant variation in prescription frequencies.

Key Insights

- Initial data contained 116 negative billing values, representing impossible real-world scenarios
- Billing distribution shows artificial clustering with 458 cases concentrated near the maximum value (~$52764) and limited variance in high-cost cases
- Hospital fragmentation shows 39639 unique hospitals serving only 55500 patients - an implausible ratio compared to real healthcare systems
- 28% of the patients names repeat which exceeds typical real-world healthcare database rates (8-16%)
- Hospital naming conventions use artificial business suffixes (LLC, Ltd) rather than authentic geographic or descriptive naming
- Doctor distribution shows unrealistic 1.4 patient-per-doctor ratio with 75% of doctors treating only one patient
- Medication-condition misalignment reveals illogical prescription patterns (e.g., Lipitor for cancer) with unnaturally uniform distribution

Recommendations

1. **Implement automated validation rules for medication-condition clinical alignment** - In real healthcare settings, it could be useful to implement a clinical decision support system that automatically checks if a prescribed medication is clinically appropriate for the patient's condition. By developing a database that maps medications to their appropriate medical conditions, the system could flag a prescription if it doesn't align with the diagnosed condition (for example, flagging insulin for a fracture). This would serve as a safety net against prescription errors while significantly improving data quality for analytical purposes
2. **Establish standardized naming conventions for healthcare facilities** -To address inconsistent hospital naming patterns, I would propose creating a national registry of healthcare institutions with standardized official names, where each facility receives a unique ID. Furthermore, an automated name-matching software could be implemented to correct variations (e.g., standardizing 'Hospital St. Mary', 'St Mary Hospital', and 'St. Mary's Medical Center' to one official name). This standardization would enable more accurate performance analysis and reporting by institution.
3. **Implement systems with pre-configured pricing**- Rather than manual entry I recommend automated price selection based on service type and insurance provider. Each insurance company would have their rate tables and public healthcare services would also have standardized pricing. This approach would eliminate the artificial billing patterns observed in this synthetic dataset, where amounts appear randomly generated rather than reflecting real healthcare pricing structures.

Conclusion

This analytical exercise provided me invaluable practice in healthcare data quality assessment, despite using synthetic data. By systematically identifying artificial patterns—such as uniform billing distributions, illogical medication-condition pairings, and implausible doctor-patient ratios—this project developed my critical skills in detecting data quality issues that are directly transferable to real-world healthcare analytics. The process emphasized that rigorous data validation is as crucial as analytical techniques themselves, especially in healthcare contexts where data integrity directly impacts patient outcomes and operational decisions.

Future Explorations: While this analysis covered fundamental dimensions of the data, I recognize there are multiple additional directions that would merit future investigation, such as:

- Analysis of temporal patterns in hospital admissions
- Relationship between length of stay and costs
- Patient segmentation by chronic conditions
- Predictive modeling of hospital costs

As a first project, the focus centered on establishing a solid foundation for understanding the data and its limitations—an essential competency for any analyst in the healthcare field. This experience enhances my ability to distinguish authentic clinical patterns from data artifacts, a fundamental competency for any healthcare data analyst working with electronic health records or hospital administrative data.

Bibliography

admin. 2016. "The Taming of the Skew." https://www.claimsjournal.com/news/national/2016/12/01/275271.htm.

"Fast Facts on U.S. Hospitals, 2025 | AHA." 2025. https://www.aha.org/statistics/fast-facts-us-hospitals.

Hardtstock, F., R. Knapp, U. Maywald, and T. Wilke. 2020. "PMU8 Health Care Spending And The Pareto Principle - The Minority That Causes The Majority." *Value in Health* 23 (December): S604–5. https://doi.org/10.1016/j.jval.2020.08.1220.

Just, Beth Haenke, David Marc, Megan Munns, and Ryan Sandefer. 2016. "Why Patient Matching Is a Challenge: Research on Master Patient Index (MPI) Data Discrepancies in Key Identifying Fields." *Perspectives in Health Information Management* 13 (Spring): 1e. https://pmc.ncbi.nlm.nih.gov/articles/PMC4832129/.