Simulated Data: Nurse Staffing and Quality of Care

Nikhil Haas

May 13, 2015

Introduction

This simulated data set is based on the journal article "Is More Better? The Relationship between Nurse Staffing and the Quality of Nursing Care in Hospitals" in Medical Care (Sochalski, 2004). In this study, Sochalski analyzed data obtained from a 1999 survey of 8,670 inpatient nurses from Pennsylvania, U.S.A. hospitals. The survey asked nurses to rate the quality of care they delivered to patients during their last shift and estimate the number of patients they cared for, the number of tasks that went undone, and the number of patient safety problems that occurred during the same shift. The author found several correlations between the responses to the questions. The questions and response data from survey are the inspiration for this simulated data set.

Method

Here, we generate artificial responses to the questions that were in the 1999 survey. In this simulated scenario, the responses are entered into a mock hospital administrative database over several years by its nurses in an effort to better understand what the hospital can do to improve patient care. The hospital has no knowledge of what factors correlate with patient care and is hoping to use the data obtained to discover relationships that will help them achieve their goal of delivering better care.

The relationships discovered by Sochalski will drive the correlations between survey responses in this simulated data set. However, to attempt to represent the difficulties and variabilities in analyzing such a data set in the real world, staffing ratios during weekdays and the weekend will be unequal and both noise and confounding factors will be introduced into the data.

Questions and Response Ranges

Nurses have been asked to complete the survey following each shift. Below are the columns of data that will hold the response data, as well as the expected ranges for each response type.

1. Employee ID: Between 0 and 9000
2. Quality of care as rated by nurses: Poor, Fair, Good, Excellent
3. Number of registered nurses on-staff during last shift: 1 - 30
4. Number of patients cared for during last shift: 0 - 20
5. Number of tasks that went undone during last shift: 0 - 7
6. Number of patient safety problems during last shift: 0 - 4
7. Date this survey was taken: YYYY/MM/DD

Code

A single function, `generateDataSet()`, will be used to create the simulated data set. This function shall take two inputs, a `startDate` and an `endDate`, and generates survey responses for numerous ficticious employees for each day between those two dates. `generateDataSet()` will run several sub-functions that perform the various logical steps in generating the data. These sub-functions are:

- `genDaysfromRange`: generate a table of dates between `startDate` and `endDate` to make data for
- `expandRows`: for each date, make a row to hold a survey response for each nurse who worked that day
- `setQualOfCareRating`: create a Quality of Care rating for each survey response
- `relateQualOfCareToNumPatients`: create each nurse's Patient Workload
- `addEmployeeID`: create employee IDs for each survey response
- `convertQual`: convert Quality of Care from integers to strings 'Poor', 'Fair', etc.
- `filterCols`: remove the tempoary columns that were use during data generation

With these sub-functions, we can define a single function to generate and return the data set:

```
generateDataSet <- function(startDate, endDate) {
  eachDay <- genDaysFromRange(startDate, endDate)
  dataSet <- expandRows(eachDay)
  dataSet <- setQualOfCareRating(dataSet)
  dataSet <- relateQualOfCareToNumPatients(dataSet)
  dataSet <- addEmployeeID(dataSet)
  dataSet <- convertQual(dataSet)
  dataSet <- filterCols(dataSet)
  )
  dataSet
}
```

Then run the function with:

```
dataSet <- generateDataSet(startDate="1998/03/08", endDate="2007/12/01")
```

Of particular note is the linear relationship between Quality of Care and Patient Workload, created by `relateQualOfCareToNumPatients`. This relationship is built after `setQualOfCareRating` is run. We could have instead first produced the number of patients the hospital intakes each day and have different probabilities of intake during weekday and weekend days; following this, Quality of Care ratings that depend on *numNurses* and and patient intake could have been produced. However, since we know a linear relationship will be created between Quality of Care and patient workload, it was chosen to produce Quality of Care first and then create patient workload from that. This approach makes it easier to simulate the results of the 1999 survey better. Because the total number of "Poor", "Fair", "Good", and "Excellent" ratings from the survey is known, we could use those numbers to create probabilities when simulating Quality of Care to better represent the survey results. Below is the code to generate the linear relationship between Quality of Care and Patient Workload.

```
relateQualOfCareToNumPatients <- function(dataSet) {
  # Creates a column 'numPatientsCaredFor' which is linearly related to
  # qualOfCare. In the paper, the relationship between the two variables is
  # argued to be in the reverse (i.e. that Quality of Care rating depends on
  # Number of Patients). This is not being refuted, here, we are simply using
  # a reverse relationship to generate the data set. The linearly relationship
  # will exist no matter which variable we have depend on which, so it is up
  # to the analyst to interpret the direction of the relationship.

  linRelate <- function(x) {
    # The linear relationship to be applied to qualOfCare
    -1.76*x + 9 + rpois(1, lambda=2)
    # Adds some variability using Poisson distribution. With lambda=2, the
    # distribution should have mode=1, min=0, max=8, mean=2. The idea is that
    # on a given day a nurse could get overloaded by caring for more patients,
    # but the likelihood of getting overloaded by N patients decreases as N
    # increases.
    }

  dataSet$patientWorkload <- NA
  for (w in c('weekday', 'weekend')) {
    rowsOfType <- dataSet$qualOfCare[dataSet$typeOfDay == w]
    dataSet$patientWorkload[dataSet$typeOfDay == w] <- vapply(
      rowsOfType,
      FUN=linRelate,
      FUN.VALUE=double(1)
      )
    }
  # Set the Patient Workload as an integer instead of a decimal value
  dataSet$patientWorkload <- as.integer(
    round(dataSet$patientWorkload, digits=0)
    )
  dataSet
}
```

You can see that a bias of 9 is introduced to increase the Patient Workload to a range between 0 and 20, which reflects the true survey results. You will also see that variability in the linear relationship (i.e. noise) is introduced using a Poisson distribution, seen below, to arbitrarily increase the Patient Workload. The idea behind introducing variability in this way is that a hospital might plan to have a specific number of nursing staff available to cover an estimated intake of patients, but there is a probability that the number of patients might surge. It might also decrease, but in this simulation we have a baseline number of patients plus a varying likelood of Patient Workload surge.

The full codeused to generate the dataset code can be seen in Appendix A.

Results

By creating the `eachDay` data frame, we have a list of days and the number of nurses that worked every day. The *numNurses* working each day is generated from a random normal distribution that is different for weekday and weekend days. The effects of that difference can be seen in Table 1, which shows that only 7 nurses worked the first Sunday, compared to a typical 9-12 nurses during the weekdays.

Table 1: Head of `eachDay` data.frame from genDaysFromRange()

date	dayOfWeek	typeOfDay	numNurses
1998-03-08	Sunday	weekend	6
1998-03-09	Monday	weekday	11
1998-03-10	Tuesday	weekday	10
1998-03-11	Wednesday	weekday	13
1998-03-12	Thursday	weekday	11
1998-03-13	Friday	weekday	11

This difference in staffing during weekday and weekend days seen here is reflected across the entire dataset, as shown in Table 2.

Table 2: Means, Min. and Max. Values from `eachDay`

	Value
Mean number of nurses	9.55
Mean number of nurses (weekday)	10.97
Mean number of nurses (weekend)	6.01
Min number of nurses (weekday)	7.00
Min number of nurses (weekend)	2.00
Max number of nurses (weekday)	15.00
Max number of nurses (weekend)	9.00

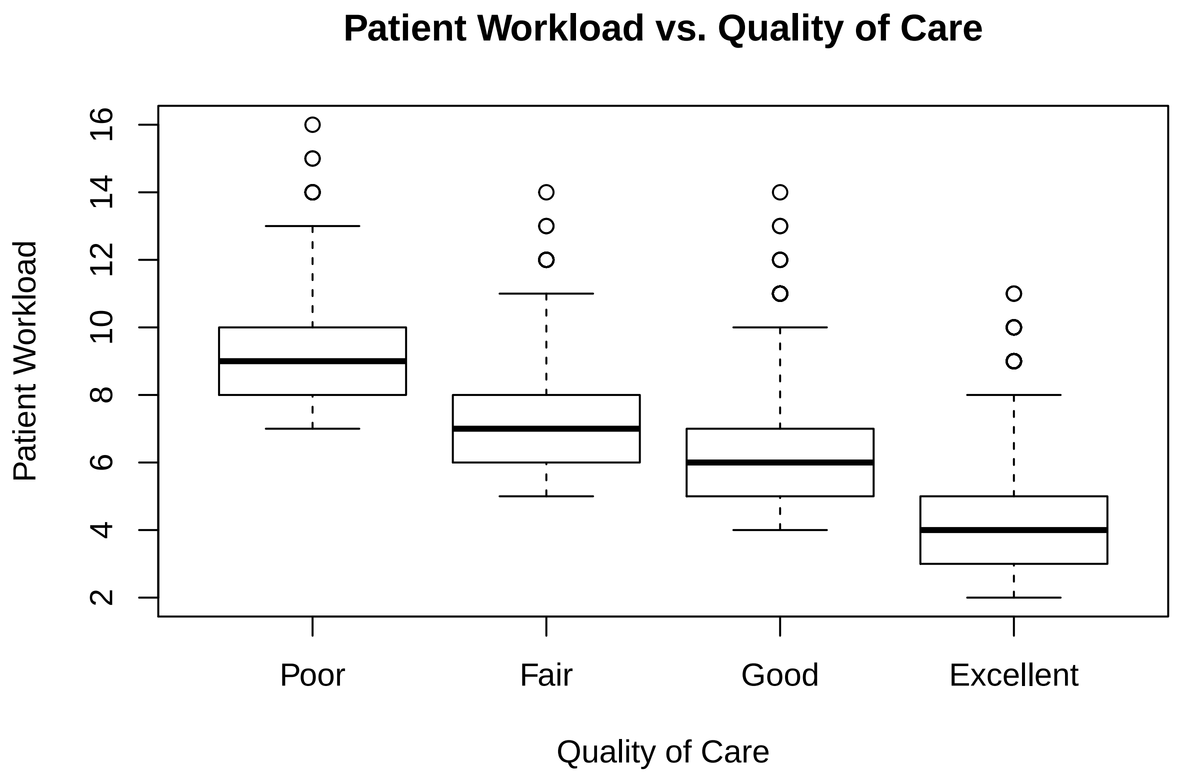
Next, we look at the Quality of Care data produced. Below we see that Quality of Care ratings during the weekday are far more likely to be 3 ("Good") or better. In contrast, during the weekend quality of care is rarely 4 ("Excellent") and is typically 1 ("Poor"). This is consistent with the weekday and weekend probabilities used to generate Quality of Care data.

We also can see the differences between weekday and weekend Quality of Care in Table 3 below.

Table 3: Quality of Care Ratings During Weekdays and Weekends

	Value
Mean quality of care (weekday)	3.56
Mean quality of care (weekend)	1.91
STD of quality of care (weekday)	0.67
STD of quality of care (weekend)	0.92

Patient workload was created through a linear relationship with Quality of Care ratings. That linear relationship can be observed in the plot below.



We see that as patient workload increases the Quality of Care rating decreases. What this graph cannot tell you, however, is that patient workload is very high during the weekends. This is a key component of the data simulation that should be discovered during the analysis phase.

For added realism, certain nurses are more likely to be weekend workers than others. The histogram below shows the Employee IDs (numeric values between 0 and 9000) separated by type of day and binned. Employees with IDs between approximately 4000 and 6000 appear to not work weekends, and certain other employees work weekends more frequently than weekdays.

Finally, we can see the first several rows of the resulting data set in Table 4.

Table 4: Head of final dataSet

date	employeeID	qualOfCare	patientWorkload	numDailyStaffedNurses
1998-03-08	006225	Fair	8	6
1998-03-08	006595	Good	5	6
1998-03-08	002677	Good	6	6
1998-03-08	003184	Good	5	6
1998-03-08	008279	Poor	13	6
1998-03-08	006773	Excellent	5	6
1998-03-09	00293	Excellent	4	11
1998-03-09	0056	Excellent	5	11
1998-03-09	00854	Excellent	4	11
1998-03-09	001221	Excellent	6	11
1998-03-09	003369	Excellent	5	11
1998-03-09	004293	Good	7	11
1998-03-09	008164	Excellent	4	11
1998-03-09	006773	Excellent	7	11
1998-03-09	004297	Good	5	11
1998-03-09	004326	Excellent	4	11
1998-03-09	004347	Good	5	11
1998-03-10	00293	Excellent	3	10
1998-03-10	00854	Excellent	5	10
1998-03-10	005828	Good	6	10

Analysis

In particular, we would like to recover the linear relationship betwen Quality of Care and Patient Workload. This can easily be achieved using the `lm` function, which fits linear models. We first need to convert Quality of Care to numeric values, then fit a model using `dataSet$patientWorkload` and `dataSet$qualOfCare`.

```r
convertQualToNum <- function(dataSet) {
  # Convert qualOfCare to numeric
  dataSet$qualOfCareNum <- NA
  dataSet$qualOfCareNum[dataSet$qualOfCare == "Poor"] <- 1
  dataSet$qualOfCareNum[dataSet$qualOfCare == "Fair"] <- 2
  dataSet$qualOfCareNum[dataSet$qualOfCare == "Good"] <- 3
  dataSet$qualOfCareNum[dataSet$qualOfCare == "Excellent"] <- 4
  dataSet
}
dataSet <- convertQualToNum(dataSet)
```

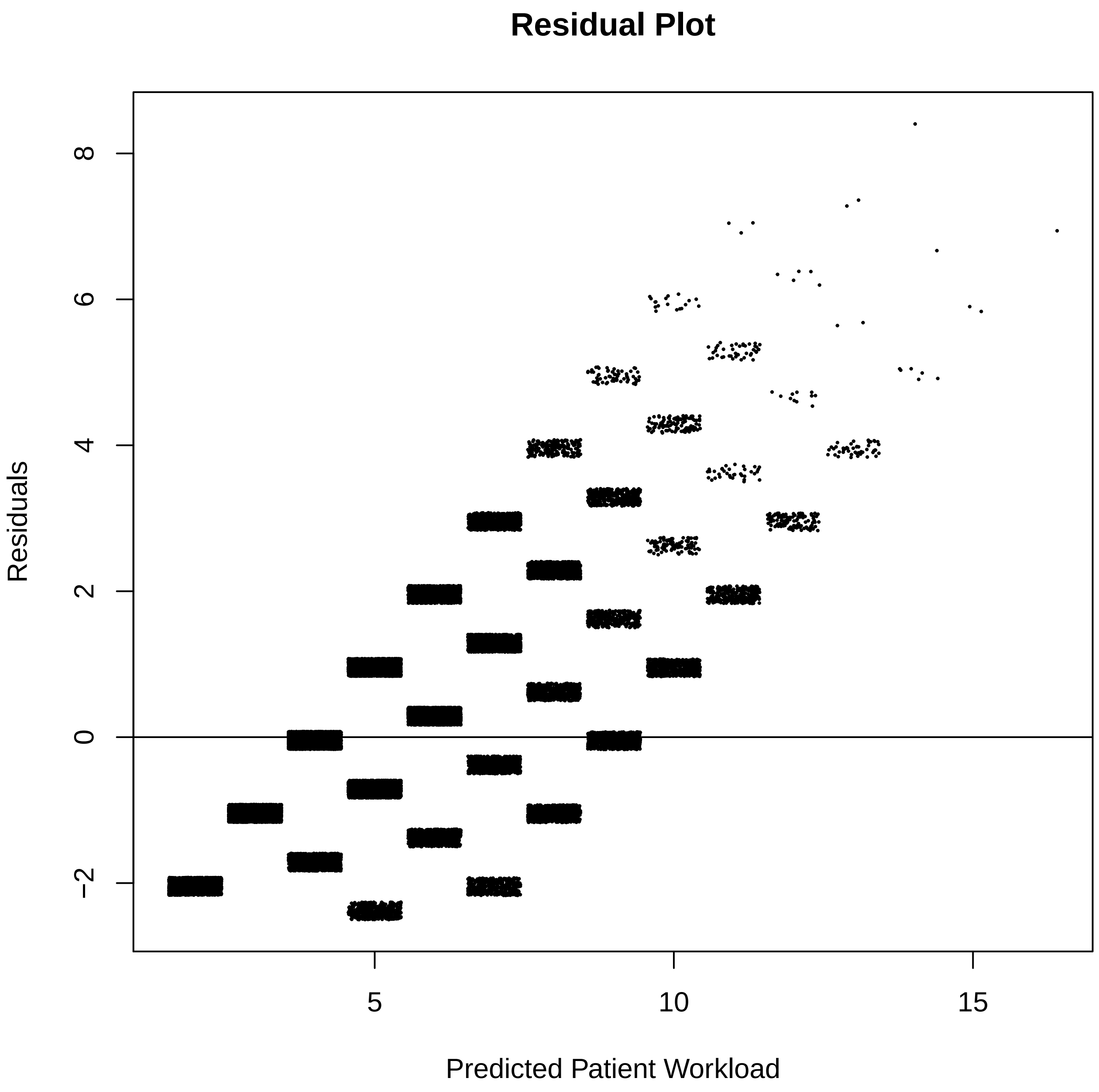
```r
analyzeDataSet <- function(dataSet) {
  lm(dataSet$patientWorkload ~ dataSet$qualOfCareNum)
}
m <- analyzeDataSet(dataSet)
print(summary(m))
```

```
##
## Call:
## lm(formula = dataSet$patientWorkload ~ dataSet$qualOfCareNum)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3807 -1.0442 -0.0442  0.9558  8.2876
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           10.717312   0.027564   388.8   <2e-16 ***
## dataSet$qualOfCareNum -1.668288   0.008097  -206.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.429 on 33956 degrees of freedom
## Multiple R-squared:  0.5556, Adjusted R-squared:  0.5556
## F-statistic: 4.245e+04 on 1 and 33956 DF,  p-value: < 2.2e-16
```

We get back the equation `-1.66x + 10.7`, with a probability of this . Recall that we biased the linear model by 9 then added noise with a Poisson distribution that is most dense around 1 and 2. This would make our true bias very close to intercept found by the fitted linear model, 10.7. The coefficient we chose when building the data was -1.76, and the linear model has a coefficient of -1.66, which is approximately 10% less.

The p-value of <2e-16 is nearly 0, therefore we can conclude that there is significance to the coefficient being non-zero. In other words, there appears to be a linear relationship between Quality of Care and Patient Workload, although the fit of that line to the data should always be verified with a plot since a linear model can be (incorrectly) fitted to data following a quadratic curve to yield a coefficient.

We can also plot the residuals, which shows the difference between the fitted values and the simulated data. Because we have ordinal data, we add jitter to better visualize the residual plot.



Looking at the residual plot, we can see a very linear pattern. Typically we would be happy seeing random points centered around the y=0 line. Instead, above we see a very non-random pattern of points. Remember, however, that our Quality of Care variable that Patient Workload is based on is ordinal data ("Poor", "Fair", etc.) that was transformed into something numeric (0, 1, etc.). Because of this, we might consider ignoring this less-than-ideal residual plot and only consider the other information acquired to evaluate our model.

References

Sochalski, J. (2004). Is more better?: the relationship between nurse staffing and the quality of nursing care in hospitals. Medical care, 42(2), II-67.

Appendix A

```r
# Set constraints
qualOfCareOptions <- c("Poor", "Fair", "Good", "Excellent")
qualOfCareProb <- list(weekday=c(.02, .04, .30, .64),
                       weekend=c(.4, .33, .21, .06))
numNurses <- c(weekday=c(mean=11, sd=1.2),
               weekend=c(mean=6, sd=1))

numTasksUndone <- c(0:7)
numTasksUndoneMean <- c(weekday=1.9, weekend=4.1)
numSafetyProbs <- c(0:4)
numSafetyProbsMean <- c(weekday=0.4, weekend=1.1)

startDate="1998/03/08"
endDate="2007/12/01"
```

```r
genDaysFromRange <- function(startDate, endDate) {
  # This function creates a data frame where each row represents a day
  # (starting from 'startDate' and ending on 'endDate') containing the day
  # of the week, whether or not the day is a weekday or a weekend day, and
  # the number of nurses working that day. Using the value for number of
  # nurses that worked each day, later we will create a data frame to
  # represent each nurse's daily responses to the questions asked by the
  # survey, which we will ultimately return as the final data set.
  #
  # |    date    | dayOfWeek | typeOfDay | numNurses |
  # | 1998/03/08 |   Sunday  |  weekend  |     7     |
  # | 1998/03/09 |   Monday  |  weekday  |     9     |
  #                         etc...

  # Create column of dates
  eachDay <- data.frame(date=seq.Date(as.Date(startDate),
                                      as.Date(endDate),
                                      by='day'))
  # Create column identifying each date as 'Sunday', 'Monday', etc.
  eachDay$dayOfWeek <- weekdays(as.Date(eachDay$date))
  # Create column that identifies each day as 'weekday' or 'weekend', using
  # the value in the 'dayOfWeek' column to lookup whether or not the day is a
  # 'weekday' or 'weekend' day.
  typeOfDay <- c(Sunday='weekend',
                 Monday='weekday',
                 Tuesday='weekday',
                 Wednesday='weekday',
                 Thursday='weekday',
                 Friday='weekday',
                 Saturday='weekend')
  eachDay$typeOfDay <- typeOfDay[eachDay$dayOfWeek]

  # Create column containing number of nurses that worked that day, which
  # depends on whether or not the day is a 'weekday' or 'weekend' day.
  eachDay$numNurses <- NA
  for (w in c('weekday', 'weekend')) {
    # You need to get mean and STD of number of nurses using the variable
    # 'numNurses' and index it like: numNurses['weekday.mean'] or
    # numNurses['weekend.sd']. Build those index values first:
    mean <- paste(w, 'mean', sep='.')  # 'weekday.mean' or 'weekend.mean'
    sd <- paste(w, 'sd', sep='.')  # 'weekday.sd' or 'weekend.sd'
    # Pull out all 'weekday' or 'weekend' days, depending on which 'w' we
    # are on in the for loop, and add the number of nurses from a normal
    # distribution whose parameters have already been set in 'numNurses'.
    numDaysOfType <- sum(eachDay$typeOfDay == w)
    sampleNurses <- rnorm(n=numDaysOfType,
                          mean=numNurses[mean],
                          sd=numNurses[sd])
    sampleNurses <- as.integer(round(sampleNurses), digits=0)
    eachDay$numNurses[eachDay$typeOfDay == w] <- sampleNurses
    }
  eachDay
}
```

```
expandRows <- function(eachDay) {
  # This function takes the data frame generated earlier (which contains
  # the number of nurses working per day) and expands those rows so there is a
  # row for *each nurse* that worked *each day*. In other words, this
  # function will expand each day in N times, where N is the numNurses
  # that worked that day. These expanded rows will contain the daily responses
  # nurses enter into the system after they finish a shift.

  # Repeate each row "number_of_times=numNurses", taking columns 1, 3, & 4 and
  # setting row.names to NULL:
  dataSet <- data.frame(eachDay[rep(c(1:nrow(eachDay)), eachDay$numNurses),
                                c(1, 3, 4)],
                        row.names=NULL)
  # Set column headers of the expanded data frame
  names(dataSet) <- c('date', 'typeOfDay', 'numDailyStaffedNurses')
  dataSet
}
```

```
setQualOfCareRating <- function(dataSet) {
  # Creates a column containg Quality of Care ratings for 'weekday' and
  # 'weekend' days. Quality of Care on the weekend will generally be lower
  # because the number of nurses is lower. Quality of Care is rated by nurses
  # as Poor, Fair, Good, or Excellent, but here we will use 1-4 to represent
  # these values during data generation.

  dataSet$qualOfCare = NA
  for (w in c('weekday', 'weekend')) {
    rowsOfType <- dataSet$typeOfDay[dataSet$typeOfDay == w]
    qualRating <- sample(x=c(1:length(qualOfCareOptions)),
                         size=length(rowsOfType),
                         prob=as.numeric(unlist(qualOfCareProb[w])),
                         replace=T)
    dataSet$qualOfCare[dataSet$typeOfDay == w] <- qualRating
    }
  dataSet
}
```

```
relateQualOfCareToNumPatients <- function(dataSet) {
  # Creates a column 'numPatientsCaredFor' which is linearly related to
  # qualOfCare. In the paper, the relationship between the two variables is
  # argued to be in the reverse (i.e. that Quality of Care rating depends on
  # Number of Patients). This is not being refuted, here, we are simply using
  # a reverse relationship to generate the data set. The linearly relationship
  # will exist no matter which variable we have depend on which, so it is up
  # to the analyst to interpret the direction of the relationship.

  linRelate <- function(x) {
    # The linear relationship to be applied to qualOfCare
    -1.76*x + 9 + rpois(1, lambda=2)
    # Adds some variability using Poisson distribution. With lambda=2, the
    # distribution should have mode=1, min=0, max=8, mean=2. The idea is that
    # on a given day a nurse could get overloaded by caring for more patients,
    # but the likelihood of getting overloaded by N patients decreases as N
    # increases.
    }

  dataSet$patientWorkload <- NA
  for (w in c('weekday', 'weekend')) {
    rowsOfType <- dataSet$qualOfCare[dataSet$typeOfDay == w]
    dataSet$patientWorkload[dataSet$typeOfDay == w] <- vapply(
      rowsOfType,
      FUN=linRelate,
      FUN.VALUE=double(1)
      )
    }
  # Set the Patient Workload as an integer instead of a decimal value
  dataSet$patientWorkload <- as.integer(
    round(dataSet$patientWorkload, digits=0)
    )
  dataSet
}
```

```
addEmployeeID <- function(dataSet){
    # Adds a column of employee IDs, where on the weekend certain employees are
    # more likely to work.

    # Number of employees should be greater than their maximum daily staff
    ids <- runif(max(dataSet$numDailyStaffedNurses) + 15, 0, 9000)
    ids <- paste0('00', as.integer(ids))  # Add 00s before ID for realism

    dataSet$employeeID <- NA

    # For every day, need to get a sample of employee IDs
    for (d in unique(dataSet$date)) {
      dailyResponses <- subset(dataSet, date == d)
      numResponses <- nrow(dailyResponses)
      typeOfDay <- dailyResponses$typeOfDay[1]
```

```r
      # Sample the employee IDs using a custom probability distrubition on
      # the IDs. This makes some employees more likely to work the weekends, and
      # other employees never work the weekends:
      p <- rep(0, length(ids))
      p[1:2] <- 0.22
      p[3] <- 0.01
      p[4:6] <- 0.12
      p[7:9] <- 0.05
      p[10:11] <- 0.02
      if (typeOfDay == 'weekend') {
        idsList <- sample(x=ids,
                          size=numResponses,
                          replace=F,
                          prob=p)
      } else {
        # Otherwise, if it's a weekday get a random employee ID
        idsList <- sample(x=ids,
                          size=numResponses,
                          replace=F)
      }
      dataSet$employeeID[dataSet$date == d] <- idsList
    }
    dataSet
  }
```

```r
convertQual <- function(dataSet) {
  # Convert Quality of Care to "Poor", "Fair", etc.
  dataSet$qualOfCare <- cut(as.numeric(dataSet$qualOfCare), 4, c("Poor",
                                                                 "Fair",
                                                                 "Good",
                                                                 "Excellent"))
  dataSet
}
```

```r
filterCols <- function(dataSet) {
  # Return the dataSet without typeOfDay
  data.frame(subset(dataSet, select=(c("date",
                                       "employeeID",
                                       "qualOfCare",
                                       "patientWorkload",
                                       "numDailyStaffedNurses"))))
}
```

```r
generateDataSet <- function(startDate, endDate) {
  eachDay <- genDaysFromRange(startDate, endDate)
  dataSet <- expandRows(eachDay)
  dataSet <- setQualOfCareRating(dataSet)
  dataSet <- relateQualOfCareToNumPatients(dataSet)
  dataSet <- addEmployeeID(dataSet)
  dataSet <- convertQual(dataSet)
  dataSet <- filterCols(dataSet)
  )
  dataSet
```

```
}
```

```
dataSet <- generateDataSet(startDate="1998/03/08", endDate="2007/12/01")
```