Casterbridge Bank Case

Emma Holt, Daniel Kuzjak, Jing Li and Jonny Mills

December 5, 2017

Why Hiring the "Right" Number of Analysts is Important

Hiring the "right" number of analysts is accomplished by trying to match the monthly supply of analysts with the monthly demand of analysts. Matching the supply and demand of analysts helps to maximize the monthly contribution to earnings. If there are too few analysts, then Casterbridge will have to bring in less productive analysts from other areas, and if there are too many analysts, then they will have analysts without projects to work on. Having either too few or too many analysts would result in having a lower monthly contribution to earnings.

Two Different Hiring Strategies

Tom's Strategy

Tom Hardy's hiring strategy is a more simple version than the one Susan used. His model calculates the number of offers the bank should extend while assuming a few variables in order to help formulate the problem in a way that is easily understood and calculated. Tom's model also fails to take estimated contribution to earnings into account and only focuses on finding the optimal value for Q, which is not the overall goal. His assumptions are based on historical information on the hiring process but do not include any variability. Tom's model is weak, however, because he doesn't take into account the fluctuation of his assumptions. First, demand is different throughout certain periods of the year. For instance, historical averages indicate that more analysts are required in September and March, but business is fairly slow in the summer and winter while bankers and clients are away on vacation (summer month's slow down is known as the "beach effect"). Further, demand will fluctuate based on the status of the market. In periods of recession, demand will be significantly lower than if the economy is performing well. Second, Tom assumes analyst retention is 95% in all months when retention rates are frequently higher than this average in January (once year end bonuses were received) and September (after a relaxing summer and as business schools kick off a new term). Finally, Tom assumes that exactly 70% of the graduates that were extended offers will accept their position with the bank. This is an oversimplification as well, as the number could change based on a given year and could better be calculated by a binomial distribution with probability of success of p = 0.07. It is clear that while Tom's model finds a good answer for a theoretical approach, it wouldn't be the best solution in a real-world setting where variability is everywhere.

Tom's Assumptions:

1. The bank will employ 70% of the graduates that were extended offers.
2. Analyst retention is 95% in all months.
3. Average demand is 90 in any month

Tom uses the following equation to find Q, the number of offers to make for July:

$$AverageDemand = (AnalystsJuly1 + AcceptedOffers) \cdot (AvgAnnualRetentionRate)$$

$$90 = (63 \cdot (0.95)^3 + 0.70Q) \cdot (.95)^6$$

$$Q = 98$$

Based on Tom's assumptions, the optimal number of offers to extend is 98.

Susan's Strategy

Susan's approach provides a model that fits the problem and the variability of the situation with better accuracy than Tom's basic model. Susan first decides that she needs to gather historic numbers in order to compile the average number of analysts that are required per month for the past five years. From this data, she found that the averages showed a wide variation in both month to month levels and between years for demand of analysts. Between years, in periods of recession, many fewer analysts were required because business was slow. On the other hand, when the economic climate was favorable, a higher number of analysts are demanded. Susan estimates that that economic growth would be extremely unlikely to be more than 10% different than the actual outcome, and therefore uses a random variable that follows Normal distribution with mean = 0% and standard deviation of 5% to account for economic change. Next, she includes in her model the variability in analyst demand over different months due to increase in work in the fall and spring and a decrease in summer and winter. Demand fluctuated more here, so she used a Normal distribution with mean of 0% but a standard deviation of 10%. Given these patterns in demand, Susan can now estimate the number of analysts that would be required in the coming year from a demand perspective.

The second part of her model is the supply of analysts. The bank had 63 currently and would retain around 95% per month over the whole year. In January and September the bank would most likely (based on historical data) experience lower retention rates, and in summer higher retention rates. To model this, Susan used a uniform distribution between 80% and 100% for January and September, 95% to 100% for summer months and 90% to 100% for the rest of the year. Next, the bank (historically) only has a 70% acceptance rate of their offers, which is displayed in Susan's model with a binomial distribution with a probability of success of p = 0.07 and the number of trials is the number of offers extended. The final part of her model is in the calculation of contribution to earnings, where she found that it appeared to be marginally better for the bank to have too few rather than too many analysts. The strength of Susan's model is that she accounts for variability in every step of the process because she knows that none of her numbers are going to exactly correct. Because of this, she can build a simulation and then predict the best number of offers to extend to analysts.

Susan's Assumptions:

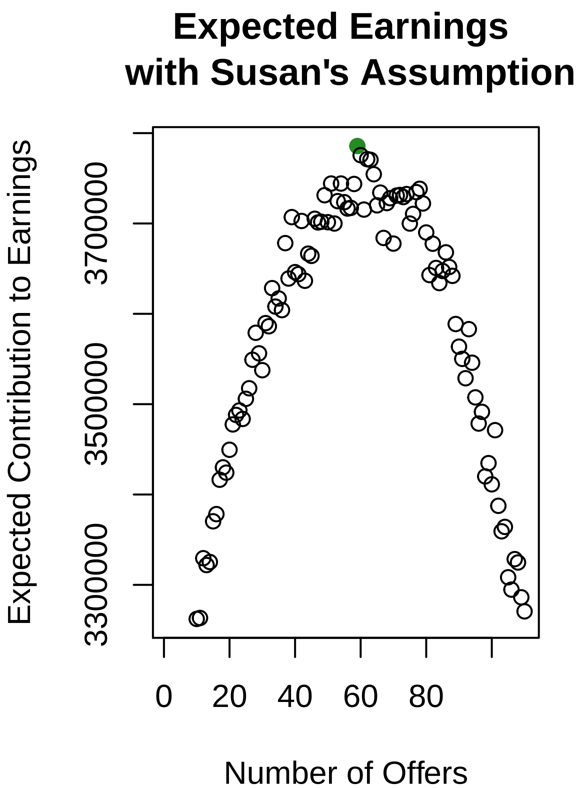
1. The bank will employ 70% of the graduates that were extended offers.
2. Change in analyst demand due to unexpected economic conditions obeys a Normal distribution with mu = 0% and sigma = 5%.
3. Variability in analyst demand each month obeys a Normal distribution with mu = 0% and sigma = 10%.
4. Uncertainty in the retention rate follows a uniform distribution of 80% to 100% in January and September, 95% to 100% in summer months, and 90% to 100% for the rest of the year.
5. Revenue that an analyst earns for the bank in a given month is $10,000.
6. Monthly contribution to earnings of a productive analyst is $4,000.
7. Monthly contribution to earnings of a transferred worker is $400.

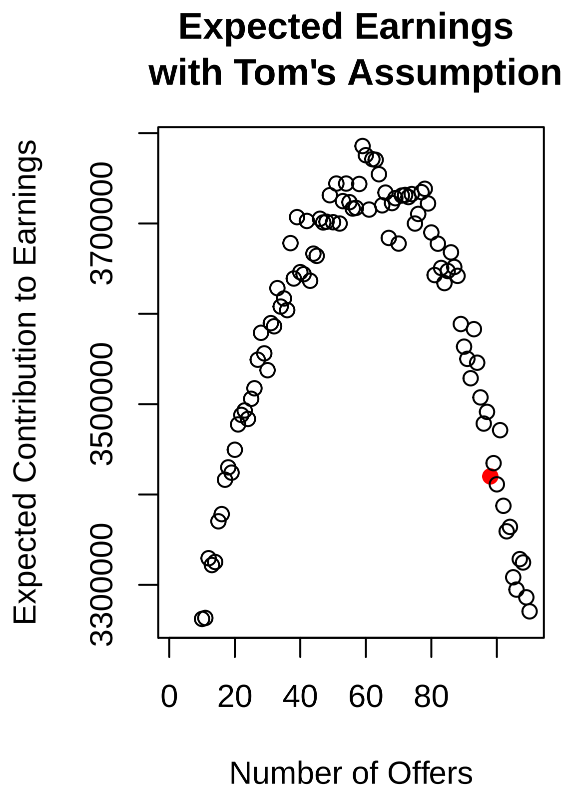
How Many Analysts to Hire?

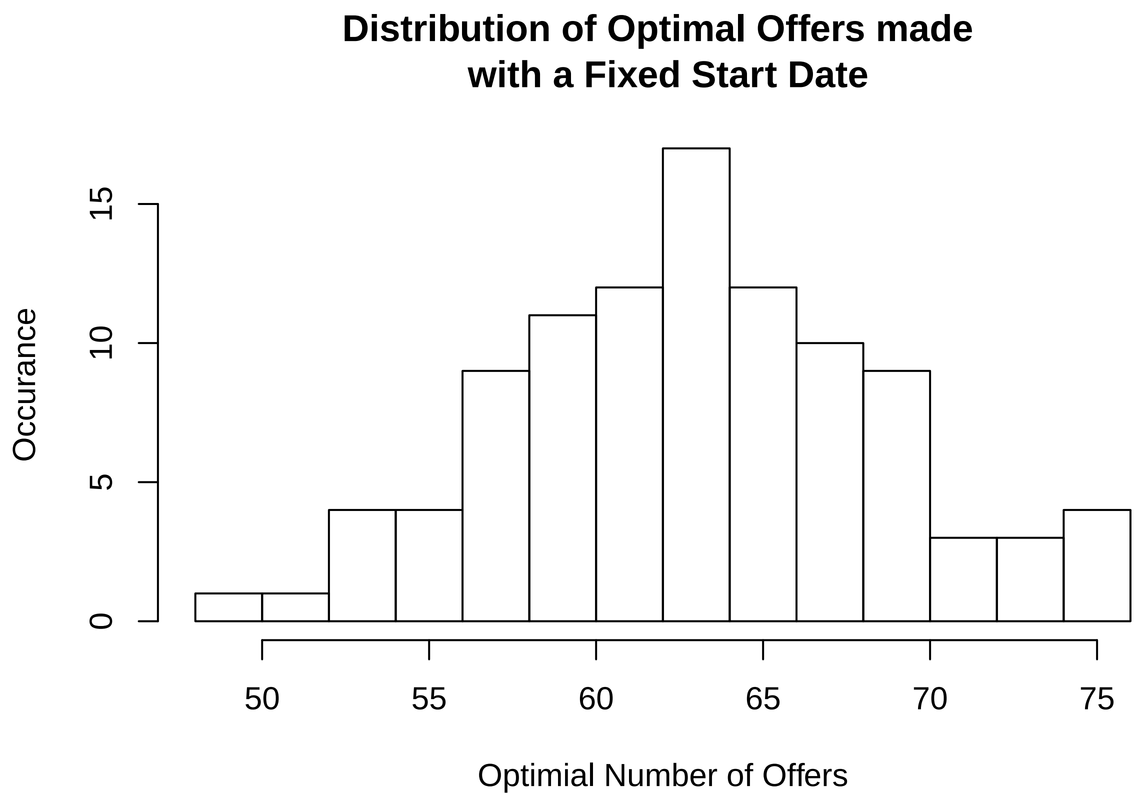
Susan wants to know how many analyst offers should be made in order to maximize the analysts' estimated contributions to earnings.

Fixed Start Date

Our objective was first to attempt to find an optimal solution for the number of offers to make in a given year of recruiting that maximized the estimated contributions to earnings, given that we are only making offers once a year and all analysts will start at the same time in July. In order to find the number of offers to make to analysts in the fixed-start model, we simply took the inputs from Susan's model and ran a simulation of different values for Q. Q in this case represents the fixed-start number of offers. This first round of simulations produced averages of expected contribution to earnings for a sequence of Q = [10:110].



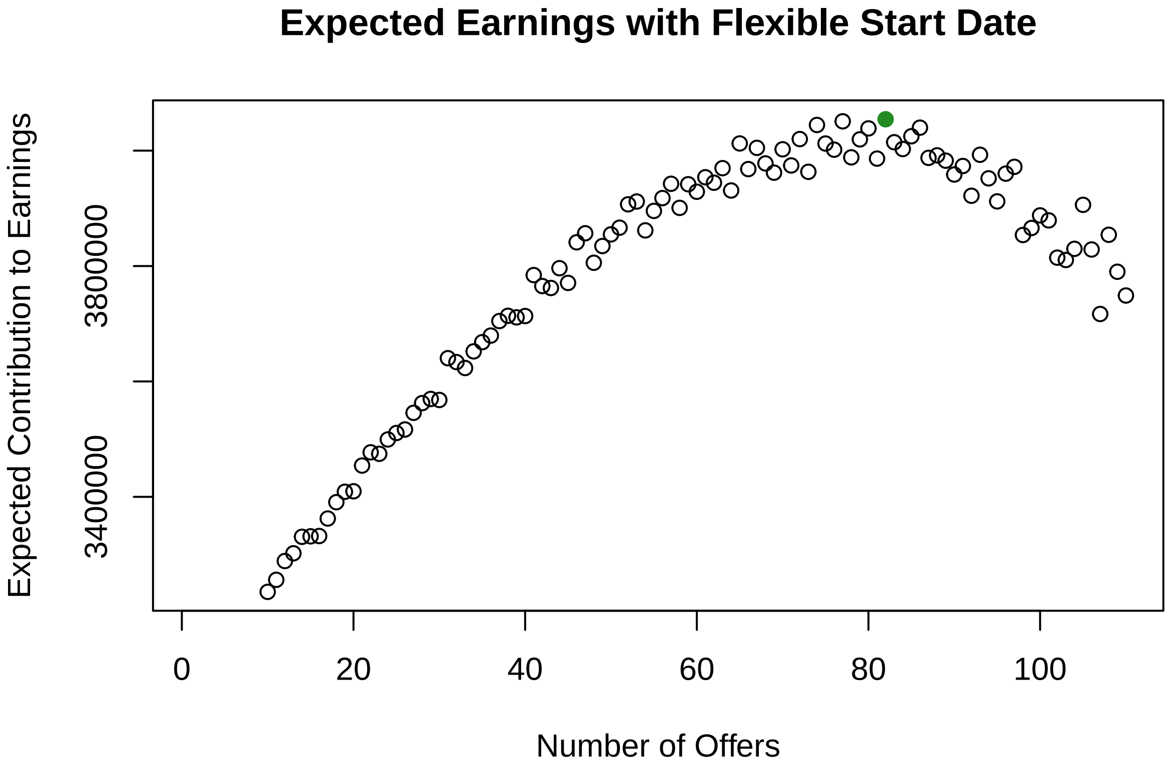


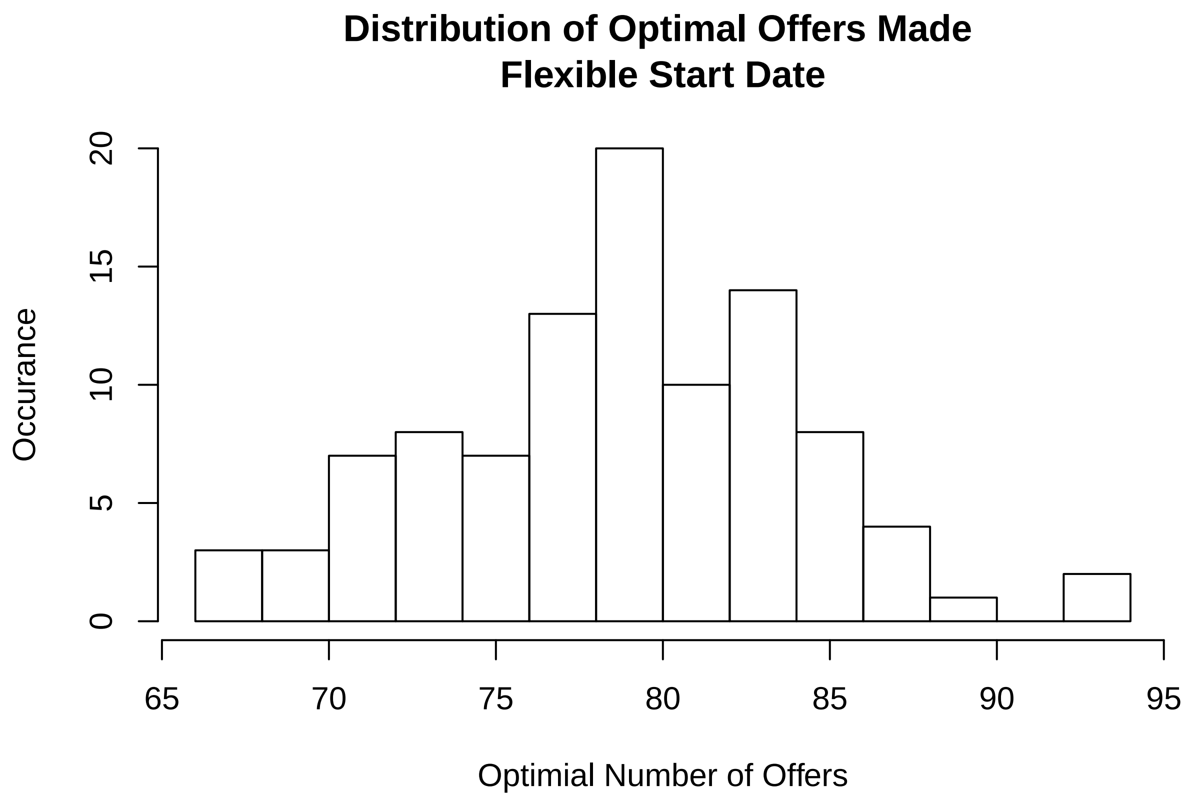


We then found the ideal optimal number of offers to be 54, offers, which produced a much higher contribution to earnings than Tom's optimal solution of 98 offers. This is displayed in the graph above, where the left graph looks at expected earnings using Susan's assumptions and the right graph looks at earnings with Tom's assumptions. Both graphs contain the number of offers (Q) to make in June on the x-axis, and the expected contribution to earnings on the y-axis. The green dot on Susan's plot represents her choice for Q, which appears to be at the top of the function (maximizing contribution to earnings). The red dot on Tom's plot depicts where Q = 98 offers would be, clearly less profitable than Susan's. The difference in expected contribution to earnings between the two strategies is around 0.3656707 million dollars per year, a significant difference! Finally, in order to confirm our optimal number of offers we produced 100 simulations of our model and found the Q which maximized contribution to earnings each time. Using this information, we then produced a frequency histogram to show which value was optimal. The second graph above displays our frequency histogram, with the optimal number of offers on the x-axis and the number of occurrences on the y-axis. This graph depicts a simulation of a normal distribution and confirmed our previous findings of 61-66 being the number of offers that produced the highest expected contribution to earnings when assuming fixed-start.

Flexible (July or September) Start Date

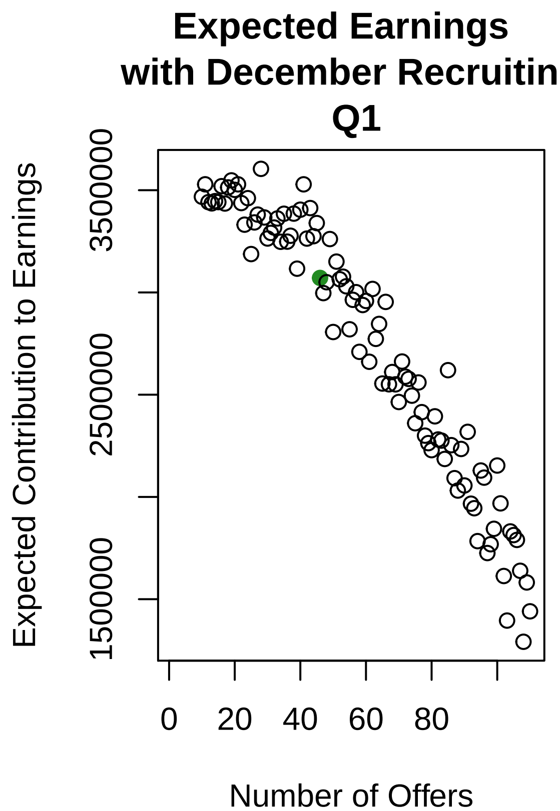
After seeing the results of her fixed start model, Susan was also interested in determining the maximum contribution to earnings that could be made if new analysts had the option to begin work in either July or September.

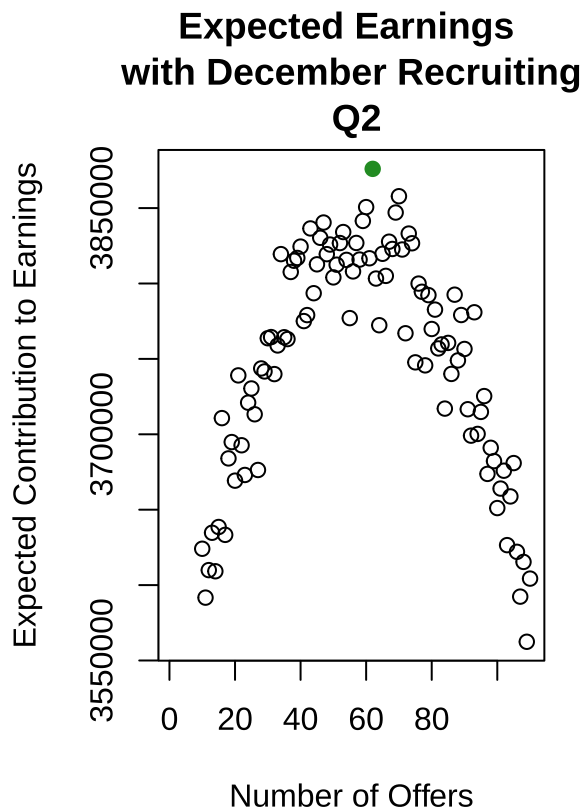

Expected Earnings with Flexible Start Date
Expected Contribution to Earnings
3400000
3800000
0
20
40
60
80
100
Number of Offers

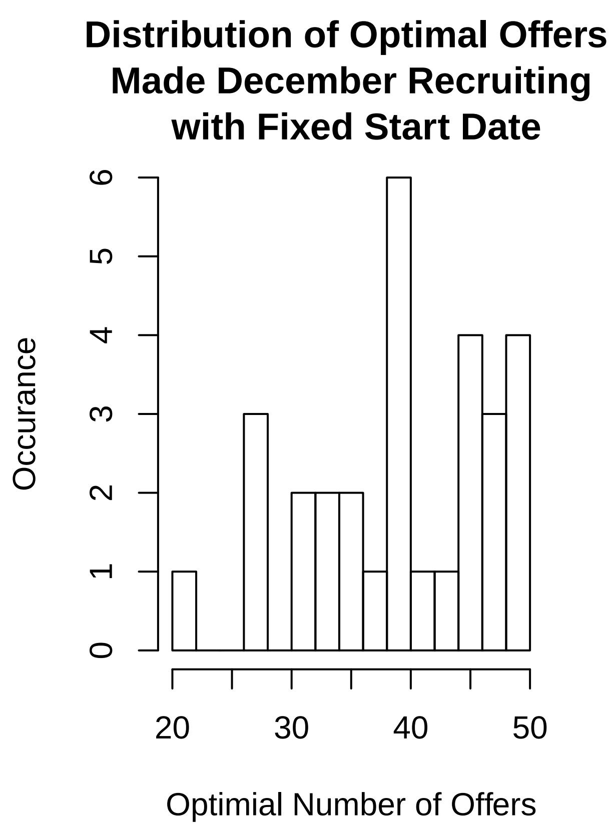


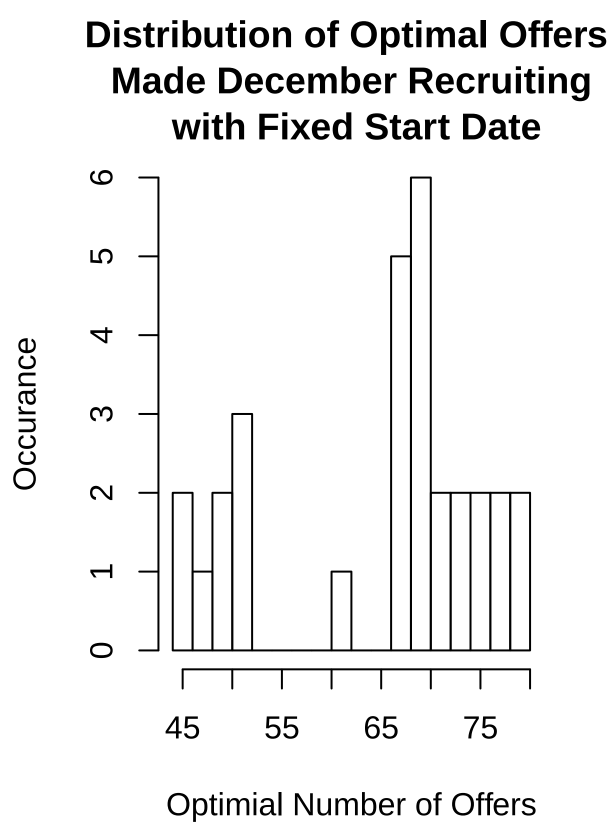
Based on her classmate's experience, Susan assumed that 50% of the students who accepted offers would start in July and the remaining half would start in September. However, only between 70% and 100% of the analysts that delayed their start date would actually show up for work in Autumn. As seen in the plot above, the bank should extend 82 offers given the flexible-start strategy, which results in the highest estimated contribution to earnings of 4.0543218 million dollars . The flexible-start policy results in an expected profit that is 0.2686138 million dollars higher than the largest estimated contribution to earnings based on the fixed-start strategy, and thus, the flexible-start policy is more profitable for Casterbridge Bank. This policy is more profitable because of the low demand for analysts in the summer. By allowing the new hires to choose between the two possible start dates, they don't have as many idle analysts over the summer months. As seen in the histogram above, based on 100 simulations of the best Q-value, we see that the best number of offers to give exend is most likely to fall between 75 and 83. This is due to the fact that of the 70% of offers accepted in May, it is unlikely that all analysts who choose to start in September will actuall show up for work.

Fixed Start Date with April and December Offers

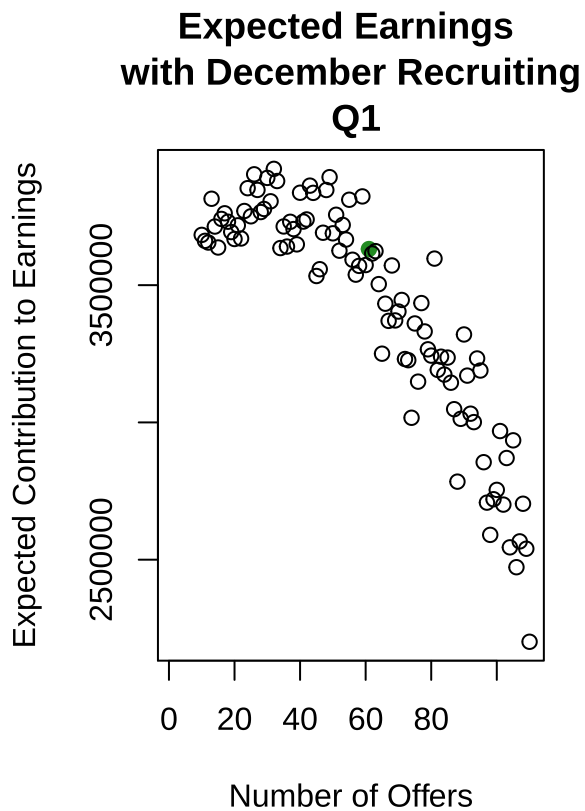


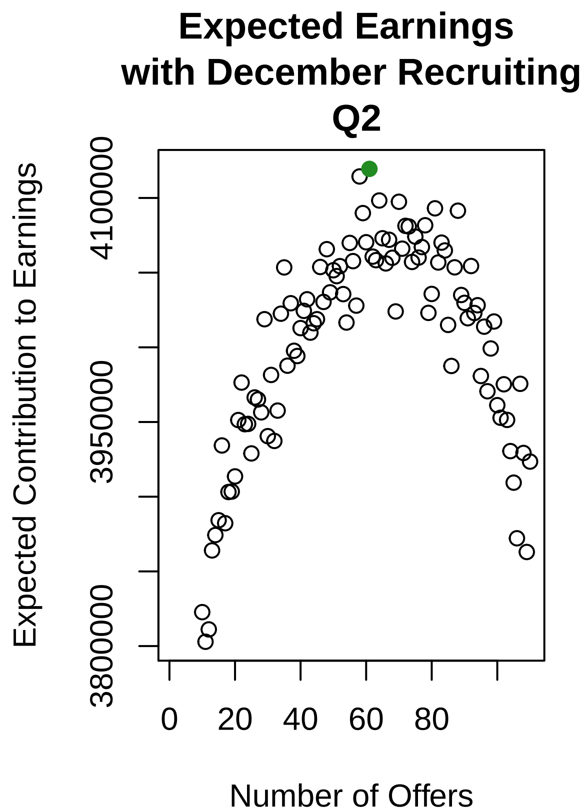


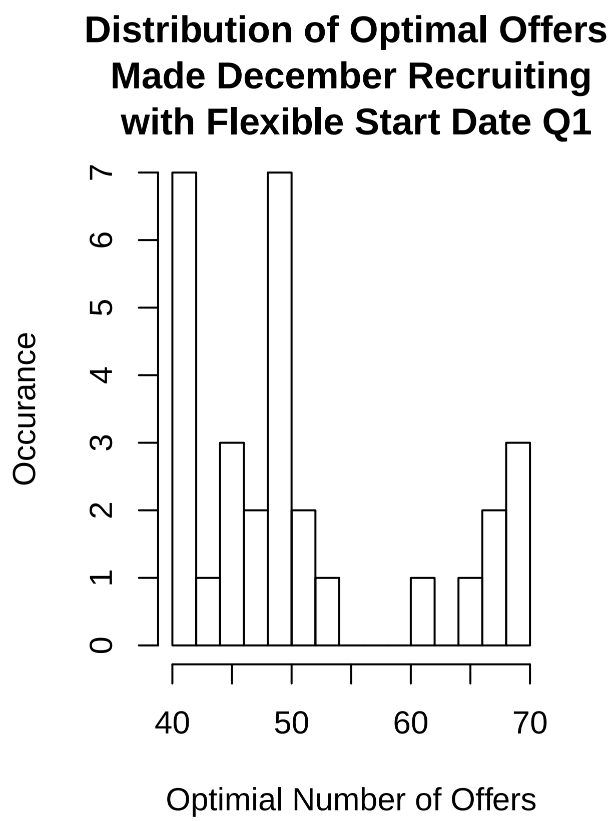


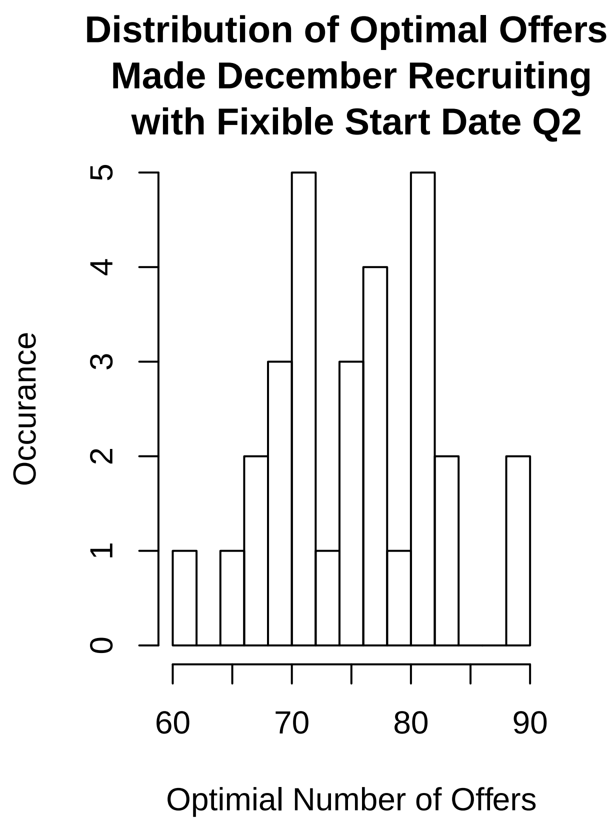


Fixed Start Date with April and December Offers









Appendix

Models

```
#### Susan's Model for Fixed Start Date #####

contributions = function(q) {
    offersmade = q
    new_analyst = rbinom(1, offersmade, 0.7)
    retention = c(runif(1, 0.9, 1), runif(4, 0.95, 1), runif(1,
        0.8, 1), runif(3, 0.9, 1), runif(1, 0.8, 1), runif(3,
        0.9, 1), runif(2, 0.95, 1))
    new_analyst1 = c(0, 0, new_analyst, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0)
    demand_hist = c(105, 95, 75, 70, 70, 110, 105, 90, 65, 80,
        90, 120, 105, 95, 75)
    unexpected = rnorm(1, mean = 0, sd = 0.05)
    noise = rnorm(15, mean = 0, sd = 0.1)
    actual = c()
    for (i in 1:15) {
        actual[i] = demand_hist[i] * (1 + unexpected) * (1 +
            noise[i])
    }
    analyst_next = c(63)
    for (i in 1:14) {
        analyst_next[i + 1] = analyst_next[i] * retention[i] +
            new_analyst1[i]
    }
    short = c()
    for (i in 1:15) {
        if (analyst_next[i] - actual[i] < 0) {
            short[i] = -1 * (analyst_next[i] - actual[i])
        } else short[i] = 0
    }
    excess = c()
    for (i in 1:15) {
        if (analyst_next[i] - actual[i] > 0) {
            excess[i] = analyst_next[i] - actual[i]
        } else excess[i] = 0
    }
    excess_cost = 6000
    short_cost = 3600
    base_contribution = 4000
    earnings = c()
    for (i in 1:15) {
        earnings[i] = base_contribution * actual[i] - (short[i] *
            short_cost + excess[i] * excess_cost)
    }
    sum(earnings)
}
```

```
#### Optimum Offers Made for Fixed Start Date #####

earnings_total = c()
bestq = c()
for (i in 10:110) {
    etmp <- c()
    for (j in 1:100) {
        etmp[j] <- contributions(i)
    }
    earnings_total[i - 9] <- mean(etmp)
}
(bestq = which.max(earnings_total) + 9)


## Shift earnings_total's index backwards so that it starts
## from 10
new_earnings_total = c()
for (i in 1:length(earnings_total)) {
    new_earnings_total[i + 9] = earnings_total[i]
}
par(mfrow = c(1, 2))
index = seq(1, 110, 1)
plot(index, new_earnings_total, col = ifelse(index == bestq,
    "forestgreen", "black"), pch = ifelse(index == bestq, 19,
    1), xlab = "Number of Offers", ylab = "expected contribution to earnings",
    main = "Expected Earnings with Susan's Assumption")
plot(index, new_earnings_total, col = ifelse(index == 98, "red",
    "black"), pch = ifelse(index == 98, 19, 1), xlab = "Number of Offers",
    ylab = "expected contribution to
     earnings", main = "Expected Earnings with Tom's Assumption")
diff_earnings = new_earnings_total[bestq] - new_earnings_total[98]
max_fixed_earnings = new_earnings_total[bestq]


## Simulate the best q multiple times so that we could draw a
## distribution
best_bestq = c()
for (a in 1:10) {
    for (i in 10:110) {
        etmp <- c()
        for (j in 1:100) {
            etmp[j] <- contributions(i)
        }
        earnings_total[i - 9] <- mean(etmp)
    }
    bestq = which.max(earnings_total) + 9
    best_bestq[a] = bestq
}
par(mfrow = c(1, 1))
hist(best_bestq, breaks = 10, main = "Distribution of Optimal Offers made
     \n Fixed Start Date",
    xlab = "Optimial Number of Offers", ylab = "Occurance")
```

```
#### Susan's Model for Flexible Start Date #####

flex = function(q) {
    offersmade = q
    new_analyst = rbinom(1, offersmade, 0.7)
    new_analyst_sep = runif(1, 0.7, 1) * (new_analyst/2)
    retention = c(runif(1, 0.9, 1), runif(4, 0.95, 1), runif(1,
        0.8, 1), runif(3, 0.9, 1), runif(1, 0.8, 1), runif(3,
        0.9, 1), runif(2, 0.95, 1))
    new_analyst1 = c(0, 0, new_analyst/2, 0, new_analyst_sep,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    demand_hist = c(105, 95, 75, 70, 70, 110, 105, 90, 65, 80,
        90, 120, 105, 95, 75)
    unexpected = rnorm(1, mean = 0, sd = 0.05)
    noise = rnorm(15, mean = 0, sd = 0.1)
    actual = c()
    for (i in 1:15) {
        actual[i] = demand_hist[i] * (1 + unexpected) * (1 +
            noise[i])
    }
    analyst_next = c(63)
    for (i in 1:14) {
        analyst_next[i + 1] = analyst_next[i] * retention[i] +
            new_analyst1[i]
    }
    short = c()
    for (i in 1:15) {
        if (analyst_next[i] - actual[i] < 0) {
            short[i] = -1 * (analyst_next[i] - actual[i])
        } else short[i] = 0
    }
    excess = c()
    for (i in 1:15) {
        if (analyst_next[i] - actual[i] > 0) {
            excess[i] = analyst_next[i] - actual[i]
        } else excess[i] = 0
    }
    excess_cost = 6000
    short_cost = 3600
    base_contribution = 4000
    earnings = c()
    for (i in 1:15) {
        earnings[i] = base_contribution * actual[i] - (short[i] *
            short_cost + excess[i] * excess_cost)
    }
    sum(earnings)
}


#### Optimum Offers Made for Flexible Start Date #####

earnings_total = c()
bestq = c()
```

```r
for (i in 10:110) {
    etmp <- c()
    for (j in 1:100) {
        etmp[j] <- flex(i)
    }
    earnings_total[i - 9] <- mean(etmp)
}
(bestq = which.max(earnings_total) + 9)


new_earnings_total = c()
for (i in 1:length(earnings_total)) {
    new_earnings_total[i + 9] = earnings_total[i]
}
par(mfrow = c(1, 1))
index = seq(1, 110, 1)
plot(index, new_earnings_total, col = ifelse(index == bestq,
    "forestgreen", "black"), pch = ifelse(index == bestq, 19,
    1), xlab = "Number of Offers", ylab = "expected
     contribution to earnings",
    main = "Expected Earnings with Flexible Start Date")


best_bestq = c()
for (a in 1:10) {
    for (i in 10:110) {
        etmp <- c()
        for (j in 1:100) {
            etmp[j] <- flex(i)
        }
        earnings_total[i - 9] <- mean(etmp)
    }
    bestq = which.max(earnings_total) + 9
    best_bestq[a] = bestq
}
par(mfrow = c(1, 1))
hist(best_bestq, breaks = 10, main = "Distribution of Optimal Offers
     Made\nFlexible Start Date ",
    xlab = "Optimial Number of Offers", ylab = "Occurance")
max_flex_earning = new_earnings_total[bestq]
(diff_earnings_starts = max_flex_earning - max_fixed_earnings)




#### Susan's Model for Mid-Year Recruiting Fixed Start Date
#### #####

mid_fixed = function(q1, q2) {
    offersmade_jun = q1
    offersmade_dec = q2
    new_analyst_jun = rbinom(1, offersmade_jun, 0.7)
    new_analyst_dec = rbinom(1, offersmade_dec, 0.7)
```

```
    retention = c(runif(1, 0.9, 1), runif(4, 0.95, 1), runif(1,
        0.8, 1), runif(3, 0.9, 1), runif(1, 0.8, 1), runif(3,
        0.9, 1), runif(2, 0.95, 1))
    new_analyst1 = c(0, 0, new_analyst_jun, 0, 0, 0, 0, 0, new_analyst_dec,
        0, 0, 0, 0, 0, 0)
    demand_hist = c(105, 95, 75, 70, 70, 110, 105, 90, 65, 80,
        90, 120, 105, 95, 75)
    unexpected = rnorm(1, mean = 0, sd = 0.05)
    noise = rnorm(15, mean = 0, sd = 0.1)
    actual = c()
    for (i in 1:15) {
        actual[i] = demand_hist[i] * (1 + unexpected) * (1 +
            noise[i])
    }
    analyst_next = c(63)
    for (i in 1:14) {
        analyst_next[i + 1] = analyst_next[i] * retention[i] +
            new_analyst1[i]
    }
    short = c()
    for (i in 1:15) {
        if (analyst_next[i] - actual[i] < 0) {
            short[i] = -1 * (analyst_next[i] - actual[i])
        } else short[i] = 0
    }
    excess = c()
    for (i in 1:15) {
        if (analyst_next[i] - actual[i] > 0) {
            excess[i] = analyst_next[i] - actual[i]
        } else excess[i] = 0
    }
    excess_cost = 6000
    short_cost = 3600
    base_contribution = 4000
    dec_cost = 22000
    earnings = c()
    for (i in 1:15) {
        earnings[i] = base_contribution * actual[i] - (short[i] *
            short_cost + excess[i] * excess_cost) - dec_cost
    }
    sum(earnings)
}

#### Optimum Offers Made for December Recruiting Fixed Start
#### Date #####

earnings_total1 = c()
earnings_total2 = c()
bestq1 = c()
bestq2 = c()
for (a in 10:110) {
    for (i in 10:110) {
        etmp <- c()
```

```
        for (j in 1:20) {
            etmp[j] <- mid_fixed(i, a)
        }
        ## compute mean for the 100 simulations of each q1
        earnings_total1[i - 9] <- mean(etmp)
    }
    ## we inspect each q1 and only pick the value that will yield
    ## highest expected earnings we repeat this process for each
    ## value of q2
    earnings_total2[a - 9] = max(earnings_total1)
    ## we record the index of maximum earnings since we are
    ## looping from 10:110, this index will be the best value of
    ## q1 now, we are writing those values to a vector called
    ## bestq1 notice that the first element of this vector starts
    ## at the 10th spot this is indicating for each q2, that there
    ## is a corresponding best q1
    bestq1[a] = which.max(earnings_total1) + 9
}
## now, our job is to find the best q2
(bestq2 = which.max(earnings_total2) + 9)
## find the best q1 according to the value of best q2
(bestq1 = bestq1[bestq2])


new_earnings_total1 = c()
new_earnings_total2 = c()
for (i in 1:length(earnings_total1)) {
    new_earnings_total1[i + 9] = earnings_total1[i]
}
for (i in 1:length(earnings_total2)) {
    new_earnings_total2[i + 9] = earnings_total2[i]
}
par(mfrow = c(1, 2))
index = seq(1, 110, 1)
plot(index, new_earnings_total1, col = ifelse(index == bestq1,
    "forestgreen", "black"), pch = ifelse(index == bestq1, 19,
    1), xlab = "Number of Offers", ylab = "expected contribution to earnings",
    main = "Expected Earnings with December Recruiting\nQ1")
plot(index, new_earnings_total2, col = ifelse(index == bestq2,
    "forestgreen", "black"), pch = ifelse(index == bestq2, 19,
    1), xlab = "Number of Offers", ylab = "expected contr
     ibution to earnings",
    main = "Expected Earnings with December Recruiting\nQ2")


## from the last plot, we observe there is a range on the
## x-axis we can select to represent the best region where q1
## and q2 are most likly to be chosen from we would like to
## take a sample of 100 from each region as a pool of q1 and
## q2 values
q1 = sample(20:50, 100, replace = T)
q2 = sample(45:80, 100, replace = T)
location = c()
```

```
for (a in 1:30) {
    best_earnings = c()
    for (i in 1:100) {
        ## note that these q1's and q2's are already very likely to be
        ## the best among all q1 and q2 therefore, the vector
        ## best_earnings represents a group of possible highest
        ## expected ## earnings
        best_earnings[i] = mid_fixed(q1[i], q2[i])
    }
    ## we want to determine which of those 100 randomy generated
    ## q1's and q2's return the best earning among all other best
    ## earnings then we record the location corresponding to the
    ## best of the best earnings
    location[a] = which.max(best_earnings)
}
par(mfrow = c(1, 2))
## use the location to find which q1 is chosen from the
## sampled group of q1
hist(q1[location], breaks = 15, main = "Distribution of Optimal Offers Made
     \nDecember Recruiting with Fixed Start Date",
    xlab = "Optimial Number of Offers", ylab = "Occurance")
hist(q2[location], breaks = 15, main = "Distribution of Optimal Offers Made
     \nDecember Recruiting with Fixed Start Date",
    xlab = "Optimial Number of Offers", ylab = "Occurance")

#### Susan's Model for Mid-Year Recruiting Flexible Start Date
#### #####

mid_flex = function(q1, q2) {
    offersmade_jun = q1
    offersmade_dec = q2
    new_analyst_jun = rbinom(1, offersmade_jun, 0.7)
    new_analyst_aug = runif(1, 0.7, 1) * (new_analyst_jun/2)
    new_analyst_dec = rbinom(1, offersmade_dec, 0.7)
    new_analyst_feb = runif(1, 0.7, 1) * (new_analyst_dec/2)
    retention = c(runif(1, 0.9, 1), runif(4, 0.95, 1), runif(1,
        0.8, 1), runif(3, 0.9, 1), runif(1, 0.8, 1), runif(3,
        0.9, 1), runif(2, 0.95, 1))
    new_analyst1 = c(0, 0, new_analyst_jun/2, 0, new_analyst_aug,
        0, 0, 0, new_analyst_dec/2, 0, new_analyst_feb, 0, 0,
        0, 0)
    demand_hist = c(105, 95, 75, 70, 70, 110, 105, 90, 65, 80,
        90, 120, 105, 95, 75)
    unexpected = rnorm(1, mean = 0, sd = 0.05)
    noise = rnorm(15, mean = 0, sd = 0.1)
    actual = c()
    for (i in 1:15) {
        actual[i] = demand_hist[i] * (1 + unexpected) * (1 +
            noise[i])
    }
    analyst_next = c(63)
    for (i in 1:14) {
        analyst_next[i + 1] = analyst_next[i] * retention[i] +
```

```
            new_analyst1[i]
    }
    short = c()
    for (i in 1:15) {
        if (analyst_next[i] - actual[i] < 0) {
            short[i] = -1 * (analyst_next[i] - actual[i])
        } else short[i] = 0
    }
    excess = c()
    for (i in 1:15) {
        if (analyst_next[i] - actual[i] > 0) {
            excess[i] = analyst_next[i] - actual[i]
        } else excess[i] = 0
    }
    excess_cost = 6000
    short_cost = 3600
    base_contribution = 4000
    dec_cost = 22000
    earnings = c()
    for (i in 1:15) {
        earnings[i] = base_contribution * actual[i] - (short[i] *
            short_cost + excess[i] * excess_cost) - dec_cost
    }
    sum(earnings)
}

#### Optimum Offers Made for December Recruiting Flexiable Start
#### Date #####

earnings_total1 = c()
earnings_total2 = c()
bestq1 = c()
bestq2 = c()
for (a in 10:110) {
    for (i in 10:110) {
        etmp <- c()
        for (j in 1:20) {
            etmp[j] <- mid_flex(i, a)
        }
        earnings_total1[i - 9] <- mean(etmp)
    }
    earnings_total2[a - 9] = max(earnings_total1)
    bestq1[a] = which.max(earnings_total1) + 9
}
(bestq2 = which.max(earnings_total2) + 9)
(bestq1 = bestq1[bestq2])


new_earnings_total1 = c()
new_earnings_total2 = c()
for (i in 1:length(earnings_total1)) {
    new_earnings_total1[i + 9] = earnings_total1[i]
}
```

```r
for (i in 1:length(earnings_total2)) {
    new_earnings_total2[i + 9] = earnings_total2[i]
}
par(mfrow = c(1, 2))
index = seq(1, 110, 1)
plot(index, new_earnings_total1, col = ifelse(index == bestq1,
    "forestgreen", "black"), pch = ifelse(index == bestq1, 19,
    1), xlab = "Number of Offers", ylab = "expected contribution to earnings",
    main = "Expected Earnings with December Recruiting\nQ1")
plot(index, new_earnings_total2, col = ifelse(index == bestq2,
    "forestgreen", "black"), pch = ifelse(index == bestq2, 19,
    1), xlab = "Number of Offers", ylab = "expected contribution to earnings",
    main = "Expected Earnings with December Recruiting\nQ2")


q1 = sample(40:70, 100, replace = T)
q2 = sample(60:90, 100, replace = T)
for (a in 1:30) {
    best_earnings = c()
    for (i in 1:100) {
        best_earnings[i] = mid_fixed(q1[i], q2[i])
    }
    location[a] = which.max(best_earnings)
}
par(mfrow = c(1, 2))
hist(q1[location], breaks = 15, main = "Distribution of Optimal Offers Made
     \nDecember Recruiting with Flexible Start Date Q1",
    xlab = "Optimial Number of Offers", ylab = "Occurance")
hist(q2[location], breaks = 15, main = "Distribution of Optimal Offers Made
     \nDecember Recruiting with Fixible Start Date Q2",
    xlab = "Optimial Number of Offers", ylab = "Occurance")
```