Problem 1

a

From common sense, we should observe a positive correlation between sale price and construction year, and between sale price and size of the house/lot. From Figure 1 and Figure 2 one can see indications of this correlation, but these patterns are too noisy and random to be of value on their own. Using predictive analytics we can obtain a much better understanding of what really affects housing prices, and create a model to predict the value of new homes.

```
ggplot(data=ames[ames$LotArea < 100000,]) +
geom_point(aes(x=LotArea,y=SalePrice))
```

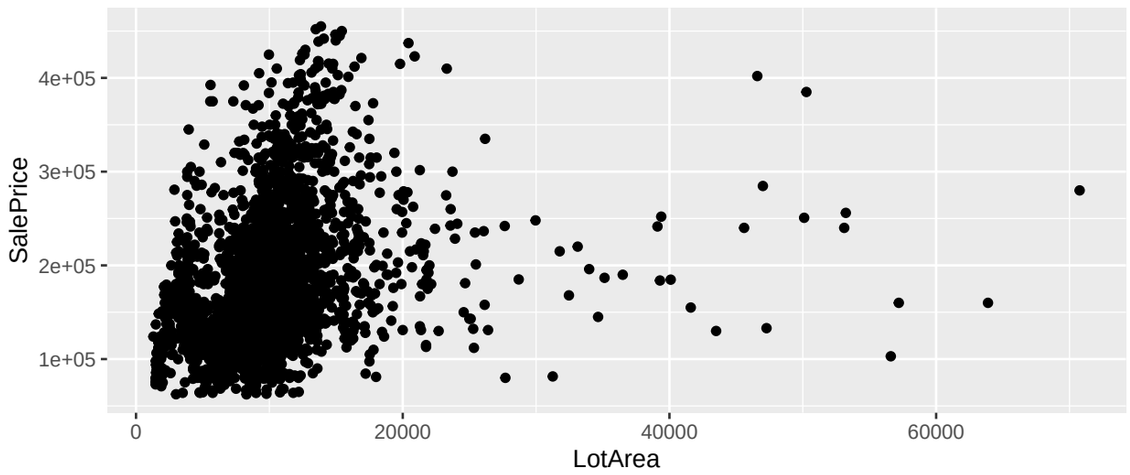

Figure 1: Price vs Lot Area (some outliers removed)

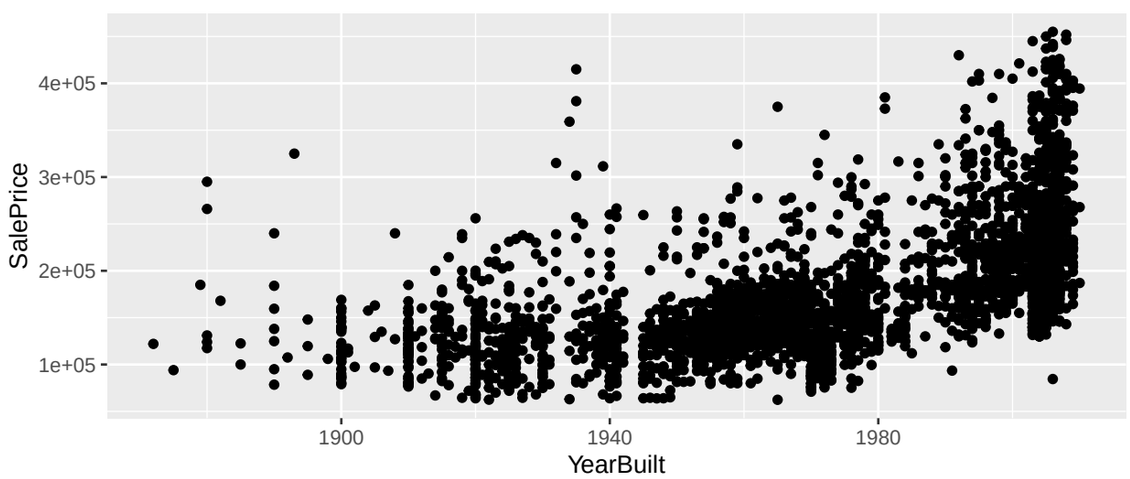

Figure 2: Price vs Construction year

b

The root node of the three in fig. 3 is exterior material quality, so this is the most important variables in the dataset, according to the model. Further down the tree it queries ground living area and neighorhood on both sides, so these seem fairly important aswell.

```
amesTree = rpart(SalePrice ~ ., data=ames.train)
```

c

Since we do not have the air condition predictor in our tree, we can not answer that question. The only thing we can tell her is that central air condition is less important that these other factors, which really does not help our friend at all. This illustrates some of the limitations with the CART approach, we can not really infer anything about things which is not in the tree.

d

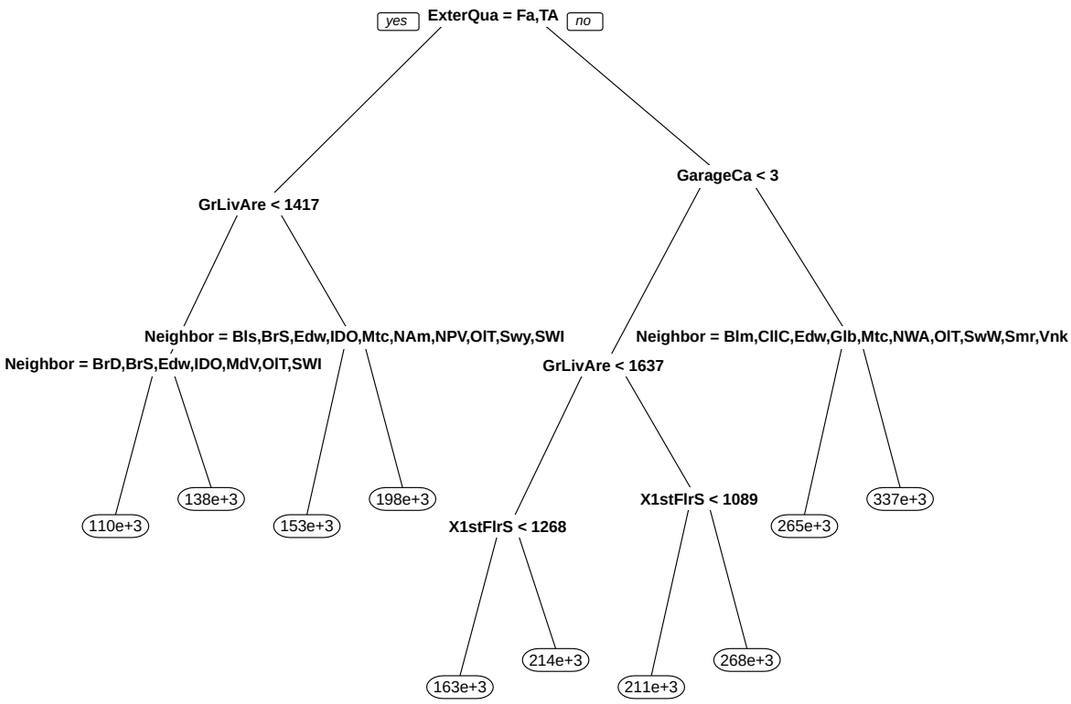

Figure 3: Regression tree generated by the training data

```
cpMax <- 0.0005
cpMin <- 0.000
cpStep <- (cpMax - cpMin)/100
refit = TRUE
if (refit){
  set.seed(123) # Reproduce results in case only this chunk is executed
  cv.amestree = train(SalePrice ~.,
                 data = ames.train,
                 method = "rpart",
                 trControl = trainControl(method="cv"),
                 metric = "Rsquared",
                 tuneGrid = data.frame(.cp=seq(cpMin, cpMax, by = cpStep)))
}

cv.amestree$bestTune

##         cp
## 67 0.00033
```

```
cvmodel = rpart(SalePrice ~., data=ames.train, cp=cv.amestree$bestTune$cp)
```

So the best CP value is 0.00033, and the resulting and much bigger tree can be seen in fig. 4. The root node is once again exterior material quality. In the first few layers down the tree, the size of the ground floor, basement, garage and the construction year are all prominent, supporting the initial theory that they are important.

e

So the extended model greatly outperforms the default model with out of sample R^2 of 0.912 vs 0.755, MAE of 14674 vs 25299 and RMSE of 21092 vs 35198. So we gain a lot of predictive power and accuracy, but we lose the simplicity, speed and interpretability of the first model. Also it was computationally much more expensive to find the extended model because of the cross validation.

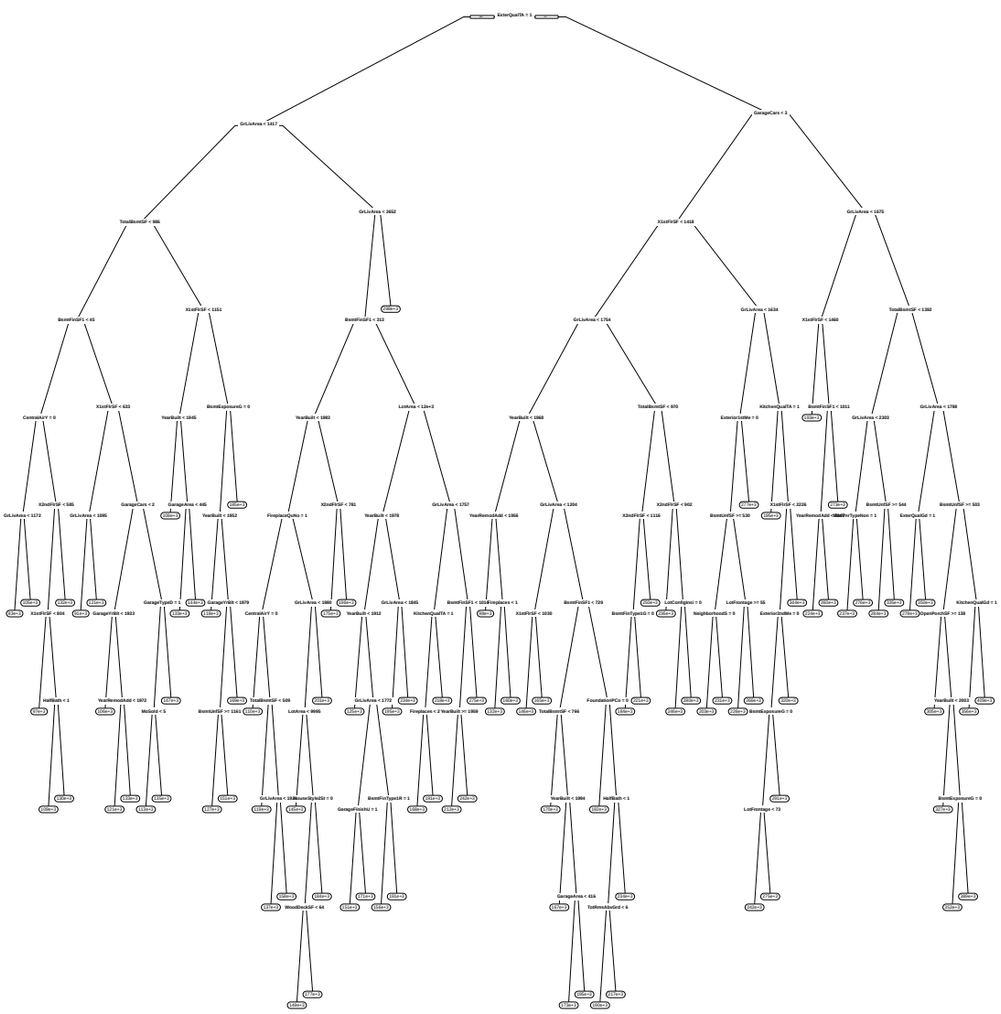

Figure 4: The best tree as chosen by the cross validation

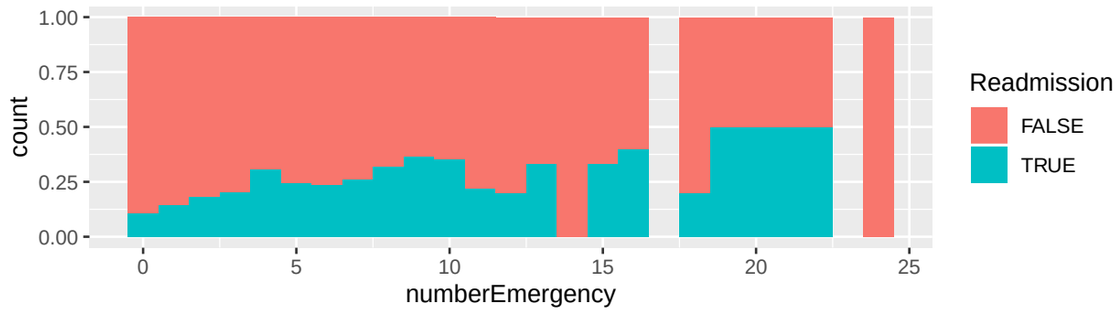

Figure 5: Conditional probability of readmission given the number of emergency visits (some outliers removed)

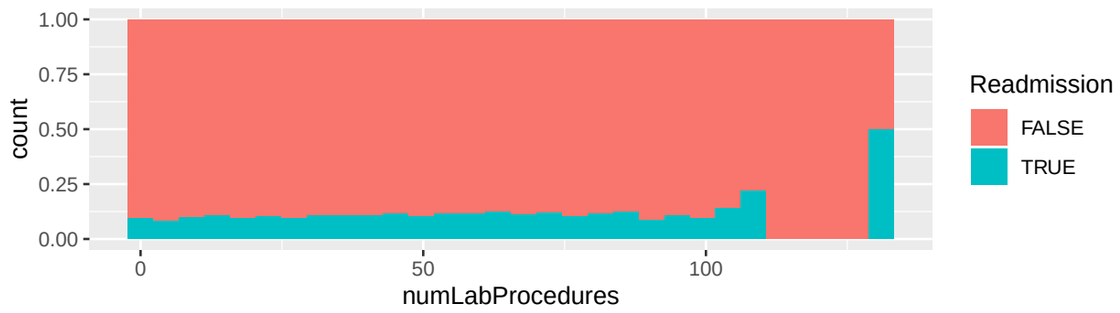

Figure 6: Conditional probability of readmission given the number of lap procedures

Problem 2

a

A reasonable proxy for the event of unplanned readmission might be the number of emergency visits, as this is also an unplanned event. From fig. 5 we can see a small increase in the conditional probability, but we have very little data for the higher number of visits, so we should be careful to generalize too much.

Another possible indicator could be the number of lab procedures, as a patient with more procedures is perhaps likely to require more medical assistance, both planned and unplanned. From fig. 6 however, this predictor does not seem very important. Again, we should be careful to extrapolate from the points with high number of procedures due to low amount of data points. With a more systematic analytical model, we can predict patients with high risk of readmission and use targeted intervention to reduce readmission in these groups.

b

For the true positives, we have the cost of the intervention, and in 75% of the cases, the readmission: $\$1200 + 0.75 \cdot \$35000 = \$27450$. For the true negatives, we pay nothing. For the false positives we only pay \$1200 for the intervention. For the false negatives we pay \$35000 for the readmission. So the added cost of missclassification, ie the loss matrix is

$$\begin{bmatrix} 0 & 1200 \\ 35000 - 27450 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1200 \\ 7550 & 0 \end{bmatrix} \tag{1}$$

c

```
lossMatrix = cbind(c(0,7550), c(1200,0))
readmTree = rpart(readmission ~.,
                  data = readm.train,
                  method = "class",
                  parms=list(loss=lossMatrix),
                  cp = 0.001)
```

In fig. 7 we can se how the tree chose patients for intervention. If you have been admitted more than twice, you get chosen. Alo in some cases if you have only been admitted once, given some other factors. Like if you have been to the emergency room more than once, which supports the initial theory. We also find the number of lab procedures in the tree, altough near the bottom, which I initially though would not be too relevant.

d

```
preds = predict(readmTree, newdata=readm.test, type="class")
CM = table(readm.test$readmission, preds)
CM

##    preds
##         0     1
##   0 18644  4056
##   1  1796   945

acc = sum(diag(CM))/sum(CM)
acc

## [1] 0.7699776

TPR <- CM[2,2]/sum(CM[2,])
TPR

## [1] 0.3447647

FPR <- CM[1,2]/sum(CM[1,])
FPR

## [1] 0.1786784

modelCostMatrix = cbind(c(0, 35000), c(1200, 27450))
modelCosts = sum(CM*modelCostMatrix)
modelCosts

## [1] 93667450

baselineCosts = 35000*sum(CM[2,])
baselineCosts - modelCosts

## [1] 2267550
```

The models accuracy is way lower than simply guessing and choosing no intervention every time, and the true positive rate is only 0.344. However since type II errors is more than 6 times as expensive as type I errors, the resulting costs are lower than just guessing.

Using the CART model for targeted telehealth intervention reduces expected costs by $2,267,550. Of the 2741 pasients who got readmitted in the test dataset, the model correctly identifies 945 of them. This saves $\$7550 \cdot 945 = \$7{,}134{,}750$ in costs, but we are also intervening "needlessly" with 4056 pasients who would not come back anyway, increasing costs by $4,867,200. One could argue that the intervention actually has some value for all the pasients, meaning the actual value of using the model is higher than the number above.

e

```
per = seq(0, 1, 0.01)
prices = seq(500, 2000, 10)

percentageDF = data.frame(
  profit = sapply(per, function(x) getProfit(1200, x)),
  percentageReduction = per
)

interventionDF = data.frame(
  profit = sapply(prices, function(x) getProfit(x, 0.25)),
  interventionPrice = prices
)
```

Let p be the percentage reduction, c be the cost of the intervention and f the resulting profit from using the prediction model as compared to the baseline model. Then the product of the confusion matrix and the cost matrix is

$$\begin{aligned} f(p,c) &= 2741 \cdot 35000 - (4056c + 1796 \cdot 35000 + 945(c + 35000(1-p))) \\ &= 70000 + 33075000p - 5001c \end{aligned} \tag{2}$$

Fixing $p = 0.25$ gives the plot in fig. 9, and the intervention is cost-effective for $c < \$1667$. In the same way, fixing $c = 1200$ gives fig. 8 and the intervention is cost effective for $p > 0.179$.

f

The idea is too choose the ratio in the loss matrix to balance the decision rule. By trial and error I found these values which give an intervention percentage of 4.14 for the training set.

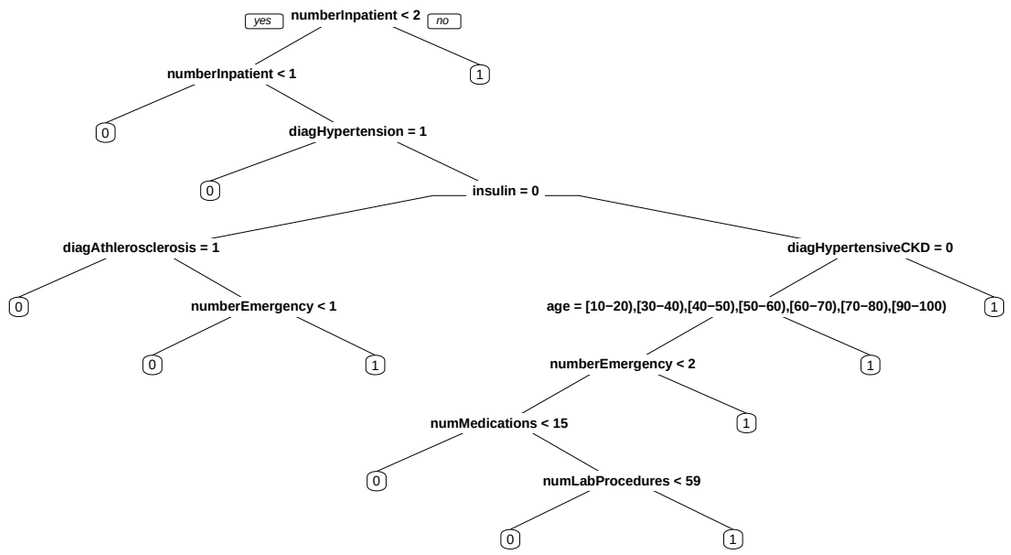

Figure 7: Decision tree for telehealth intervention

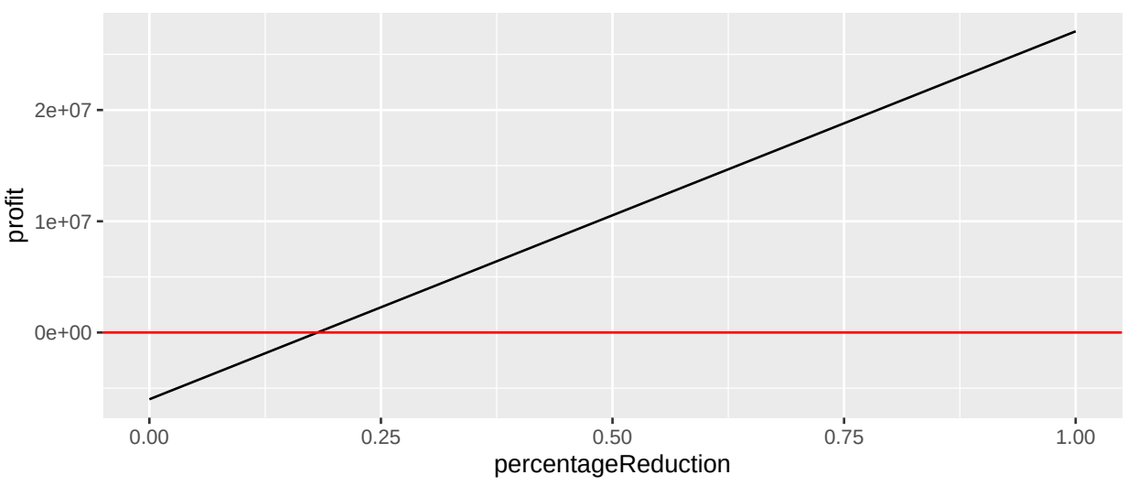

Figure 8: Profit of using the predictive model as a function of the percentage effect of the telehealth intervention

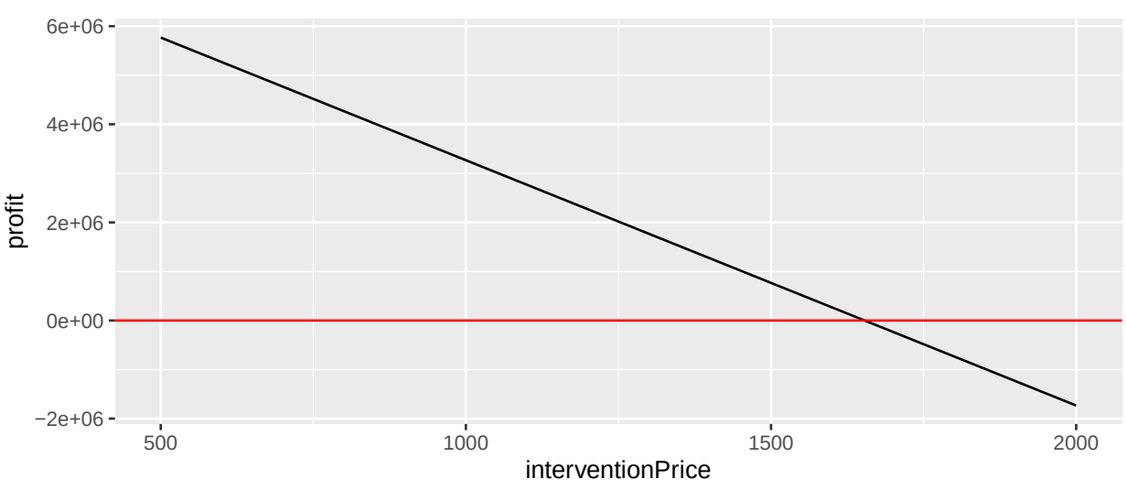

Figure 9: Profit of using the predictive model as a function of the price of the telehealth intervention

```
p = 0.2
c = 1600
falsePosCost = c
falseNegCost = 35000 - (c + 35000*(1-p))
```

The resuling confusion matrix on the test set is

```
testPreds.m = predict(readmTree.m, newdata=readm.test, type="class")
testCM.m = table(readm.test$readmission, testPreds.m)
testCM.m

##    testPreds.m
##         0     1
##   0 21900   800
##   1  2452   289

sum(testCM.m[,2])/sum(testCM.m)

## [1] 0.04280492

testCM.m[2,2]*7550

## [1] 2181950

baselineCosts - sum(testCM.m*modelCostMatrix)

## [1] 1221950

modelCosts - sum(testCM.m*modelCostMatrix)

## [1] -1045600
```

With current budgets, applying the intervention to 1089 (4.28%) pasients identified by the model in the test set has a value of $2,181,950, and gives a net profit of $1,221,950 compared to the baseline. This is lower then our original model, and we therefore have an opportunity cost of (minimum) $1,045,600 as compared to the non constrained case. Based on these findings we should increase budgets for telehealth interventions, as the potential cost reduction outweights the budget increase.