Project 5 - Understanding No-Shows for Physicians' Appointment

Team 13

April 27, 2017

Project Description

A person makes a doctor appointment, receives all the instructions and does not show up. Who to blame? Can data science help provide a solution?

Why did we choose this project?

Healthcare costs in the US took up 17.8% of GDP in 2016. Given this high cost, any form of systemic improvements can drive tremendous savings. To this end, many operations research projects have focused on how to improve processes at healthcare providers (clinics, hospitals).

As OR students, the latest hooha of United forcibly "re-accommodating" a customer was definitely an interesting one on many levels, particularly because it shone a spotlight on a very common revenue maximizing practice - overbooking.

Like airplane seats, doctor appointments are "perishable goods" - once passed, the time lost due to a no-show is irretrievable. According to a study done, the estimated cost to a community hospital per year due to no-show is approximately $3 million. There is definitely an imperative to devise a solution, but would overbooking be feasible in an industry like healthcare? We decided to test our hypothesis on a dataset shared by a hospital from Brazil.

Dataset Description

Those data have been recorded in the state of Espirito Santo, Brazil. Datapoints are only from the public sector, primary care.

Description of the dataset
- Age: Age of the patient
- Gender: Gender of the patient
- AppointmentRegistration: Time and date when the patient took the appointment
- AppointmentData: Date of the appointment
- Diabetes: Is the patient affected by diabetes? (0/1)
- Alcoholism: is the patient alcoolic? (0/1)
- Hypertension: Is the patient affected by hypertension? (0/1)
- Handicap: Level of handicap of the aptient (0/1/2/3/4)
- Smokes: Is the patient smoking? (0/1)
- Scholarship: Is the patient receiving a scholarship? Those scholarships are given by Bolsa Familia to low income families who accept to send their kids to school nd have them vaccinated.
- Tuberculosis: Is the patient affected by tuberculosis? (0/1)
- AwaitingTime: How many days the patient waited between the appointment registration and the date of the appointment
- Status: Did the patient show-up or not? (No-Show/Show-Up)

Source
It's a kaggle dataset which have been taken here. https://www.kaggle.com/joniarroba/noshowappointments

Data Cleaning

Step 0: Load Packages

Step 1: Load Dataset

Step 2: Clean & Organize Dataset

```
## 'data.frame':    300000 obs. of  15 variables:
##  $ Age                   : int  19 24 4 5 38 5 46 4 20 51 ...
##  $ Gender                : Factor w/ 2 levels "F","M": 2 1 1 2 2 1 1 1 1 1 ...
##  $ AppointmentRegistration: Factor w/ 295425 levels "2013-05-29T15:14:11Z",..: 152112 248200 23071 8
##  $ ApointmentData        : Factor w/ 534 levels "2014-01-02T00:00:00Z",..: 290 440 36 164 490 152 4
##  $ DayOfTheWeek          : Factor w/ 7 levels "Friday","Monday",..: 7 7 6 5 6 6 6 1 6 6 ...
##  $ Status                : Factor w/ 2 levels "No-Show","Show-Up": 2 2 2 2 2 1 2 2 2 2 ...
##  $ Diabetes              : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Alcoolism             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HiperTension          : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Handcap               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Smokes                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Scholarship           : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Tuberculosis          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sms_Reminder          : int  0 0 0 1 1 1 1 1 0 1 ...
##  $ AwaitingTime          : int  -29 -1 -1 -15 -6 -35 -18 -14 -14 -4 ...
```

```
##       age         gender             appointment_registration
##  Min.   : -2.00   F:200505   2015-04-15T14:19:34Z:     9
##  1st Qu.: 19.00   M: 99495   2014-03-19T17:10:21Z:     8
##  Median : 38.00              2014-07-03T08:50:50Z:     8
##  Mean   : 37.81              2015-05-27T13:57:09Z:     8
##  3rd Qu.: 56.00              2013-12-23T12:40:00Z:     7
##  Max.   :113.00              2014-01-07T15:42:57Z:     7
##                              (Other)             :299953
##              appointment_date   dayofweek_apptmt    show_up
##  2014-10-22T00:00:00Z:   759   Friday   :52771   No-Show: 90731
##  2014-09-03T00:00:00Z:   756   Monday   :59298   Show-Up:209269
##  2014-11-24T00:00:00Z:   752   Saturday : 1393
##  2014-11-19T00:00:00Z:   742   Sunday   :    6
##  2015-05-25T00:00:00Z:   741   Thursday :60262
##  2014-09-29T00:00:00Z:   739   Tuesday  :62775
##  (Other)             :295511   Wednesday:63495
##  diabetes   alcoholism hypertension handicap   smoker     scholarship
##  0:276610   0:292497   0:235233     0:294403   0:284289   0:270931
##  1: 23390   1:  7503   1: 64767     1:  5098   1: 15711   1: 29069
##                                     2:   449
##                                     3:    39
##                                     4:    11
##
##
##  tuberculosis sms_reminder daydiff_regist_appt
##  0:299865     0:128547     Min.   :-398.00
##  1:   135     1:170654     1st Qu.: -20.00
##               2:   799     Median :  -8.00
##                            Mean   : -13.84
##                            3rd Qu.:  -4.00
##                            Max.   :  -1.00
```

```
##
## [1] "C"
```

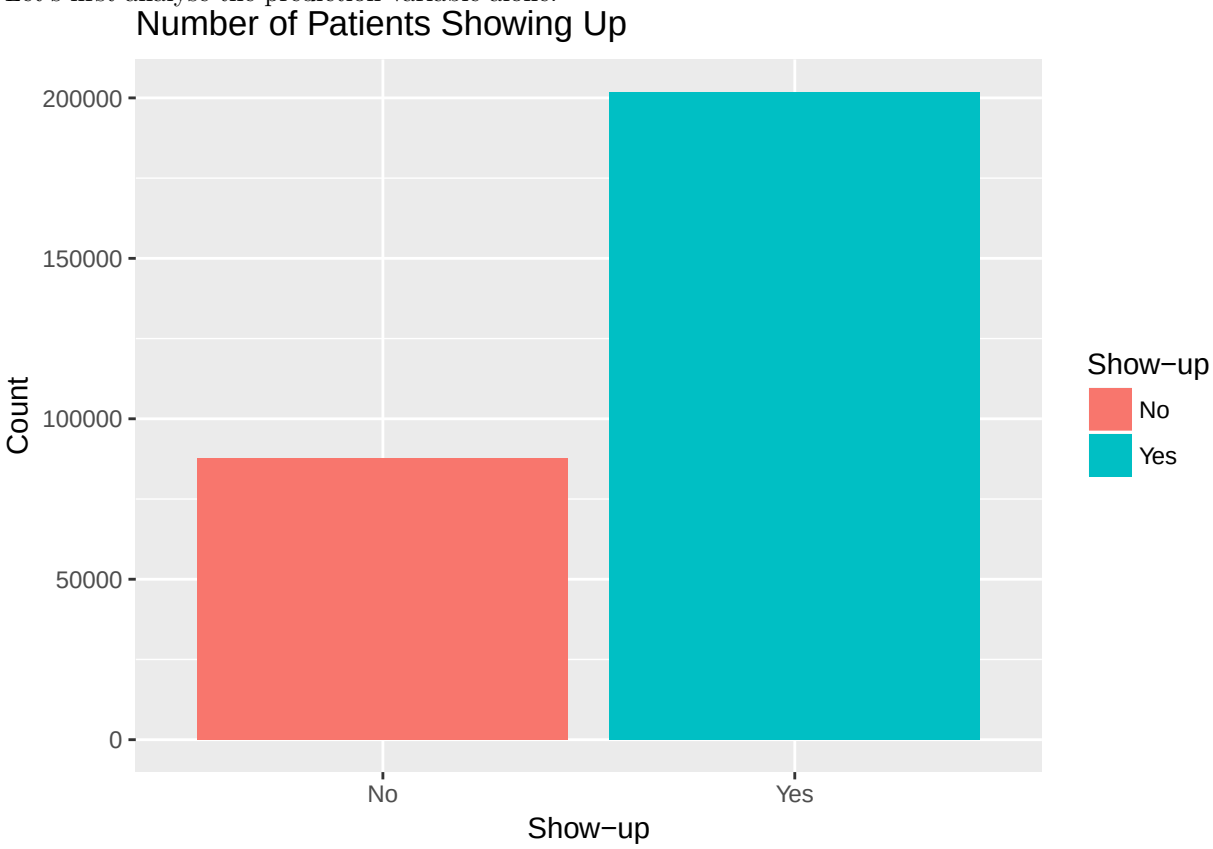
Exploratory Data Analysis

Number of Datapoints

We have access to 300k datapoints.

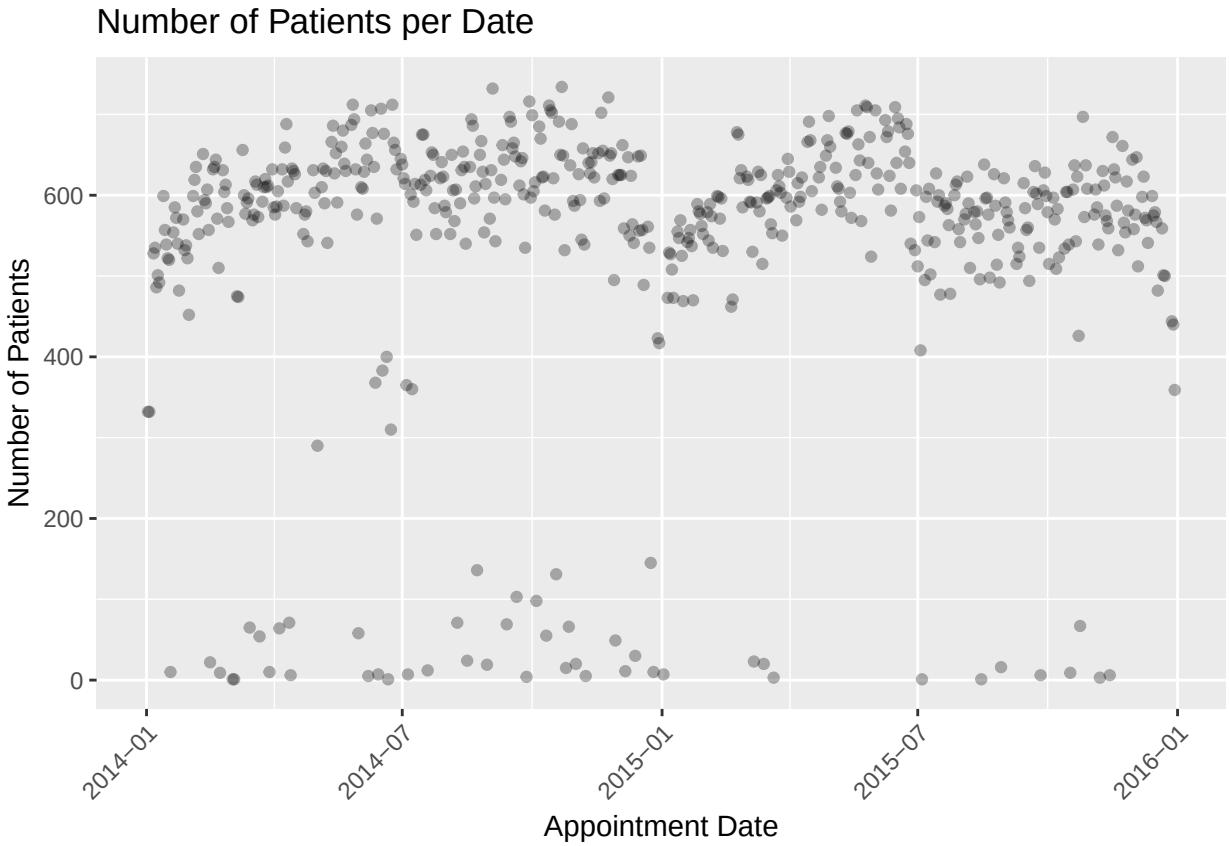
Prediction Variable

Let's first analyse the prediction variable alone.



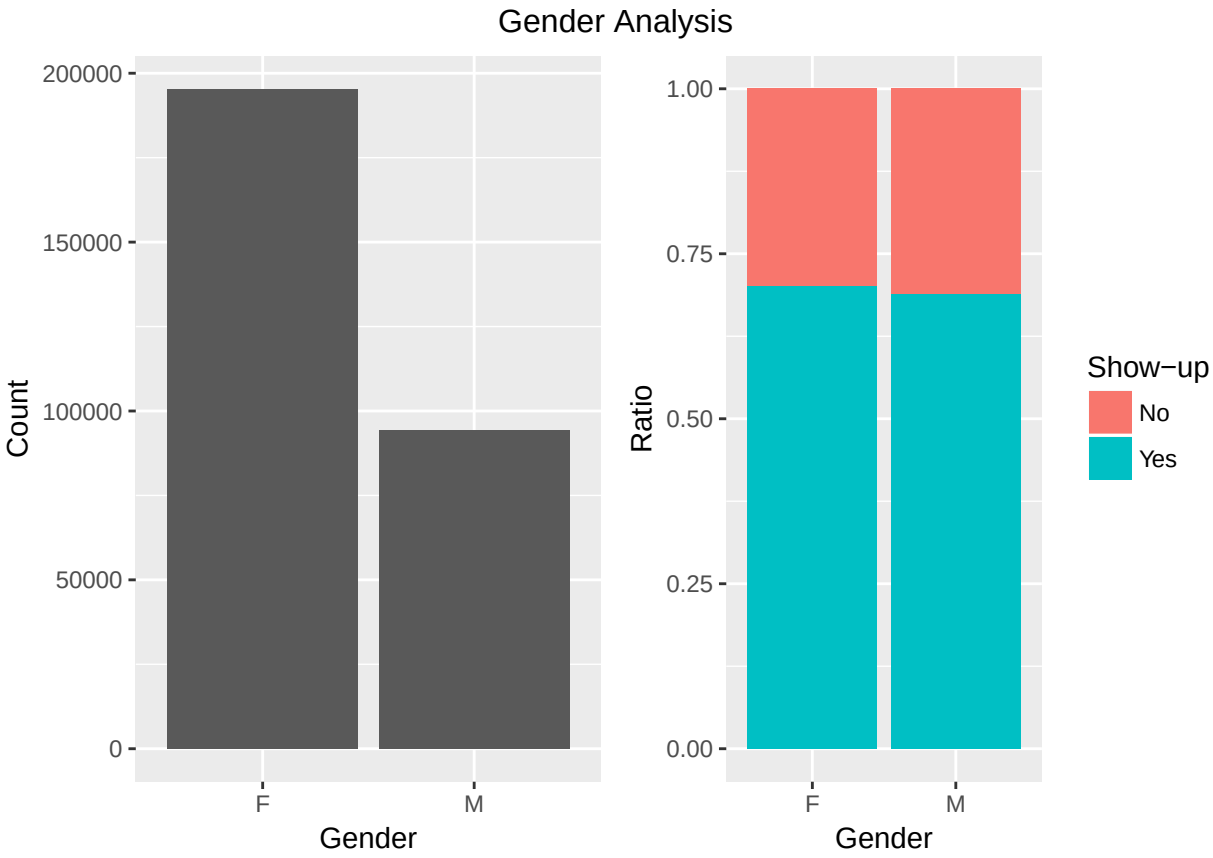
On the 300k datapoints we have, one third of them are No-Show Datapoints.
From a data science point of view, we have unbalanced classes, we will have to deal with it when doing predictions.

Timeframe & Number of Datapoints per Day



We have access to 2 years of data from January 2014 to December 2015. The number of datapoints per date is around constant. Outliers are Sundays.

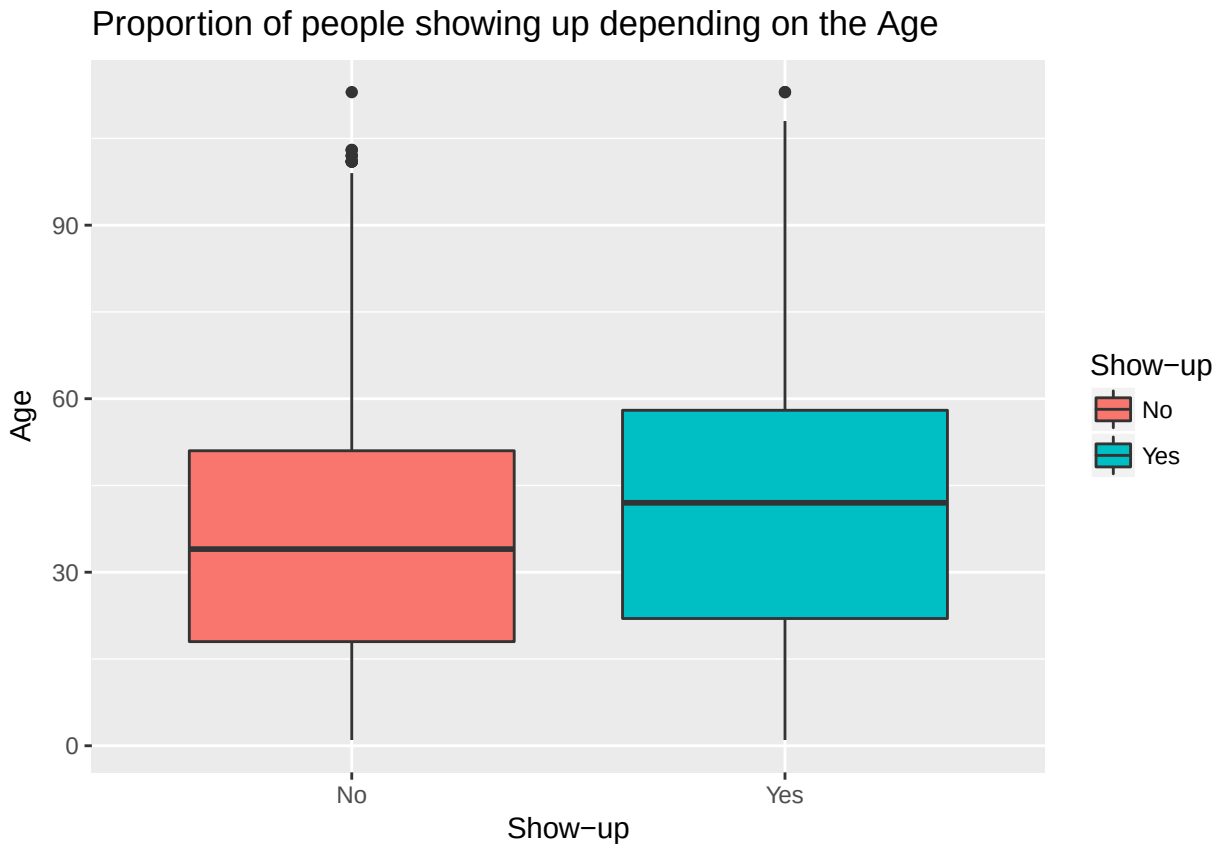
Gender Analysis



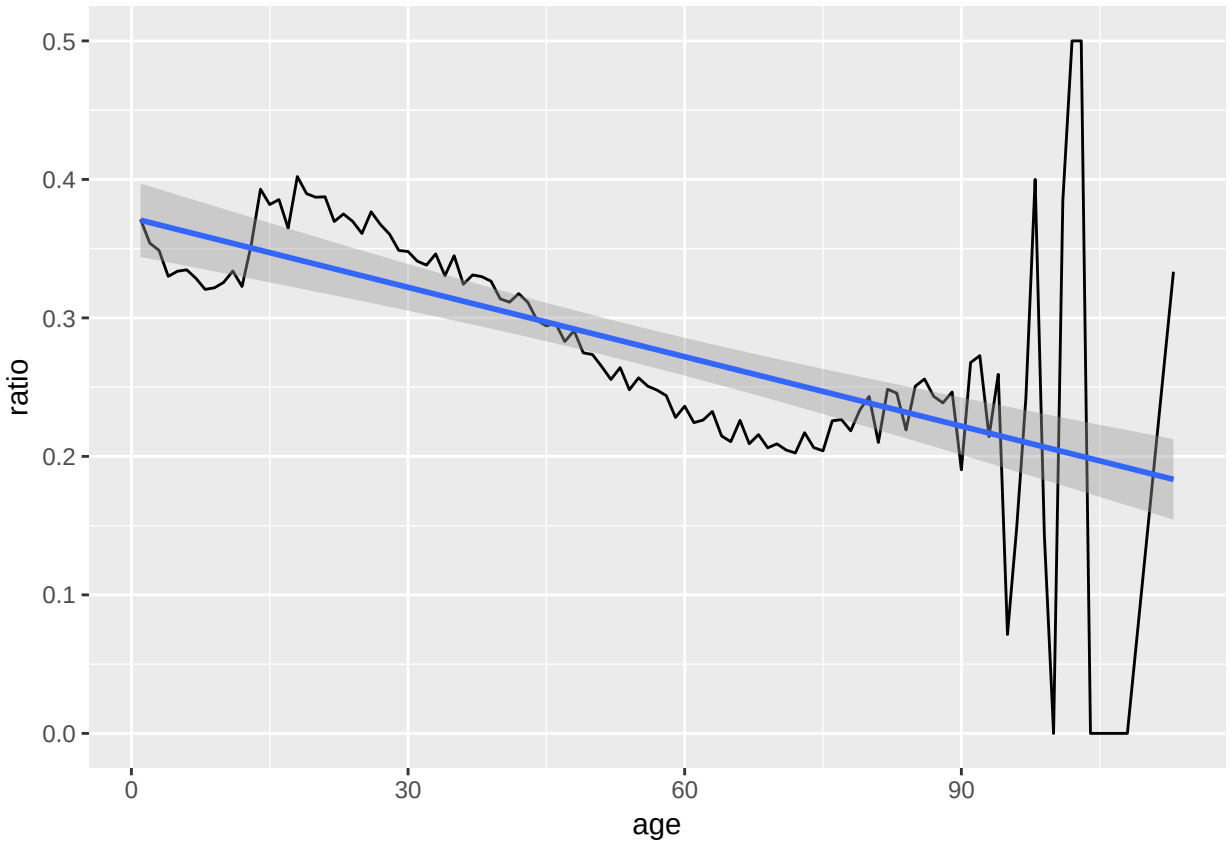
The first insight is that women are more often going to visit a physician. However the ration of no-show/show are really similar. Gender doesn't seem to play an important role.

Age Analysis

On the first graph, we display the number of patient depending on the age to know what the sample we have look like. After 60 years old, people begin to die, and hence there are less patients. We see on the representation of the ife expectancy of people in Brazil whicih is comprised between 60 and 100 years old. (The life expectancy in Brazil is 73.8 years old).

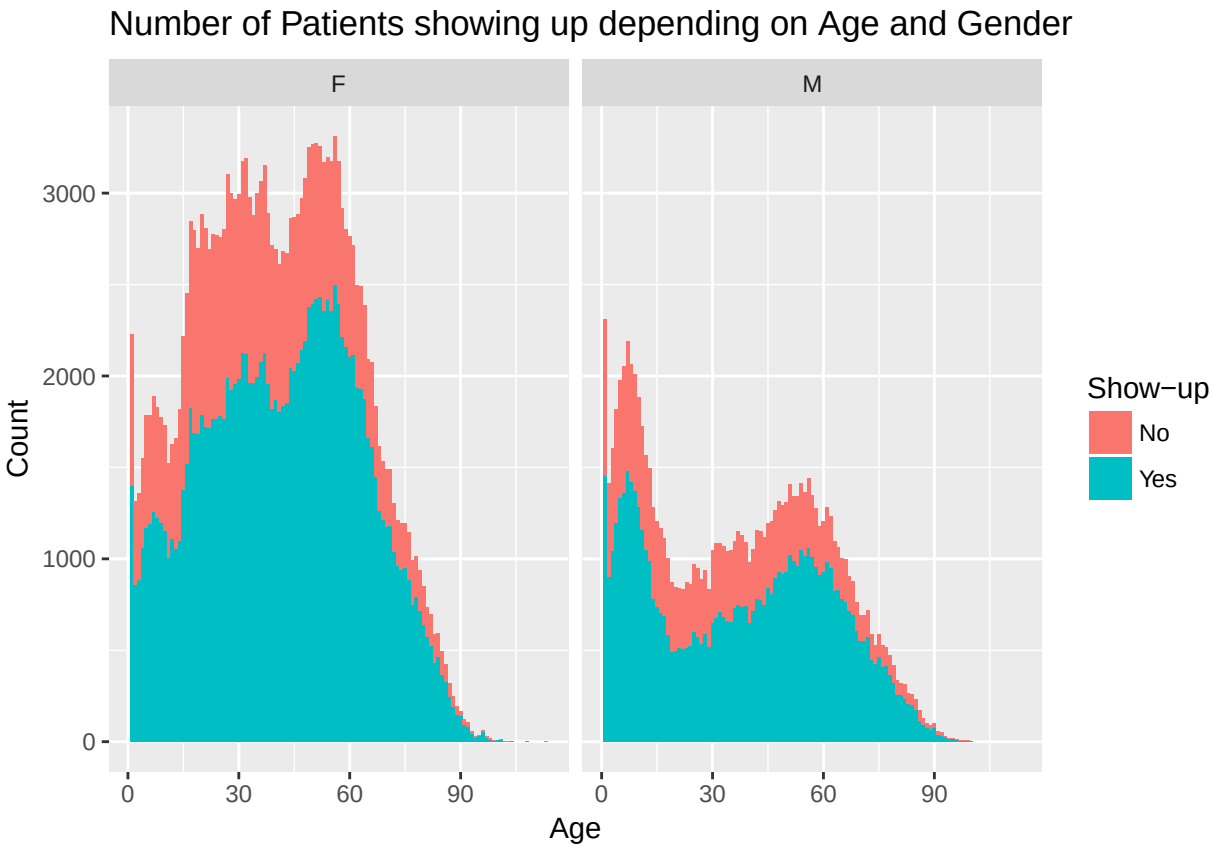


The age seems to be a important factor. The older the people are the more likely they are going to show up.



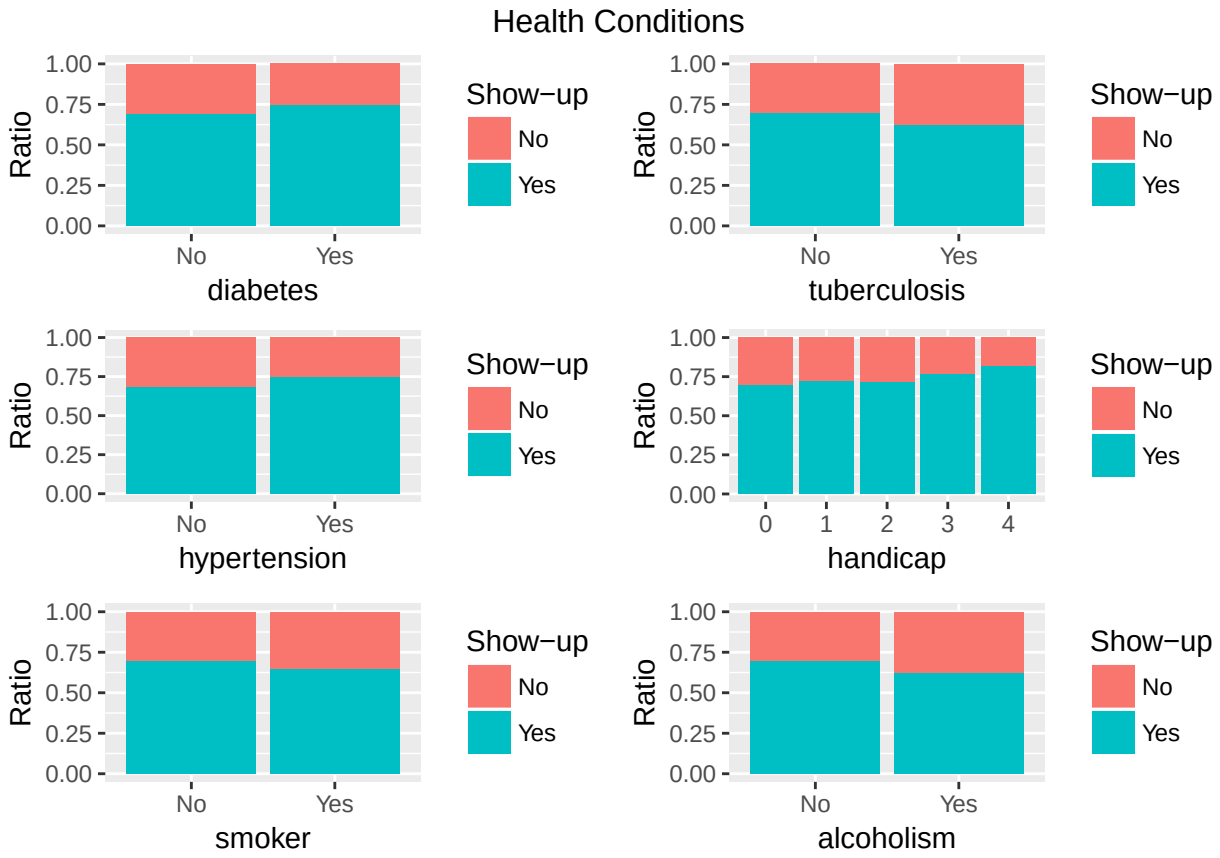
Above, we can see the same trend. After 95 years old the figures are not significant because the sample is too small. Hence only the first part of the graph can be interpreted.

Age & Gender Analysis

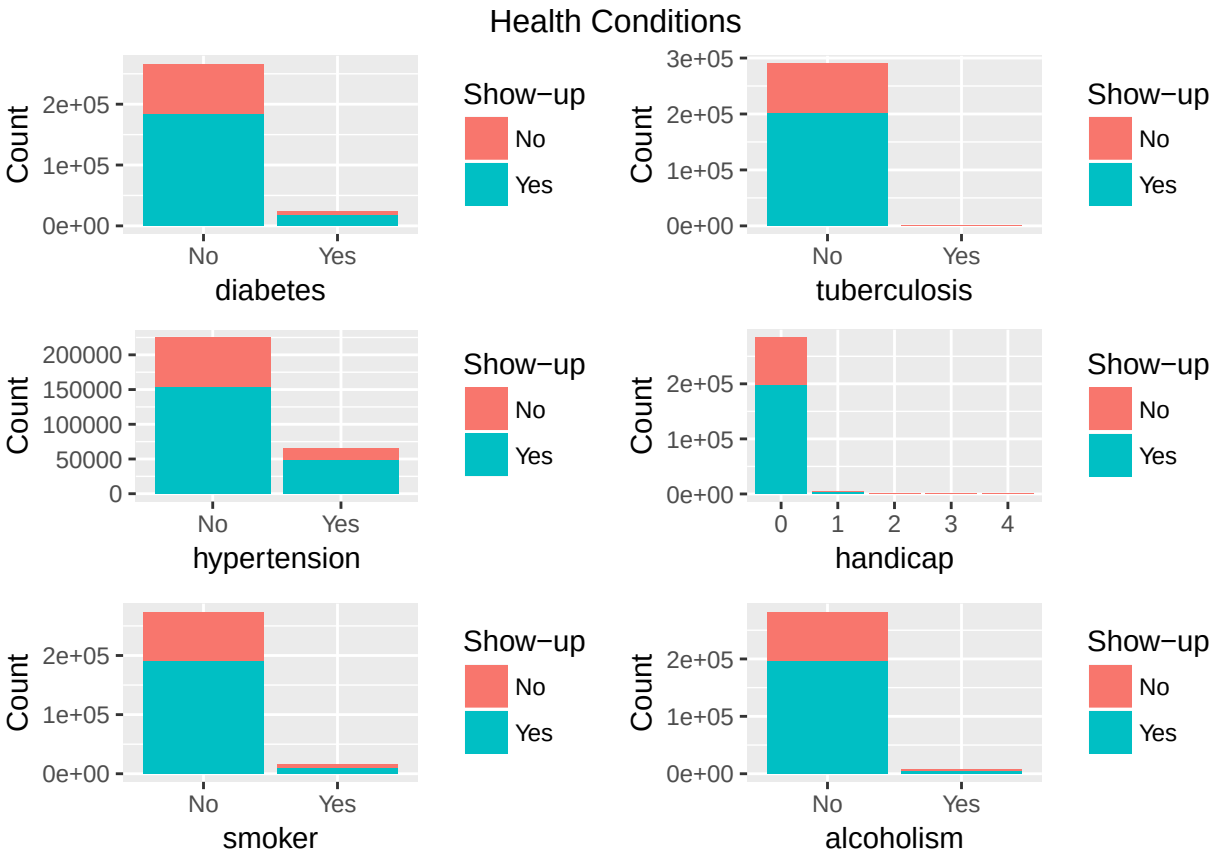


The ratio is similar between woman and man. We see again that women take more appointments. However we see that before 20 years old the number of appointments and the trend is exactly the same.

Health Conditions Analysis

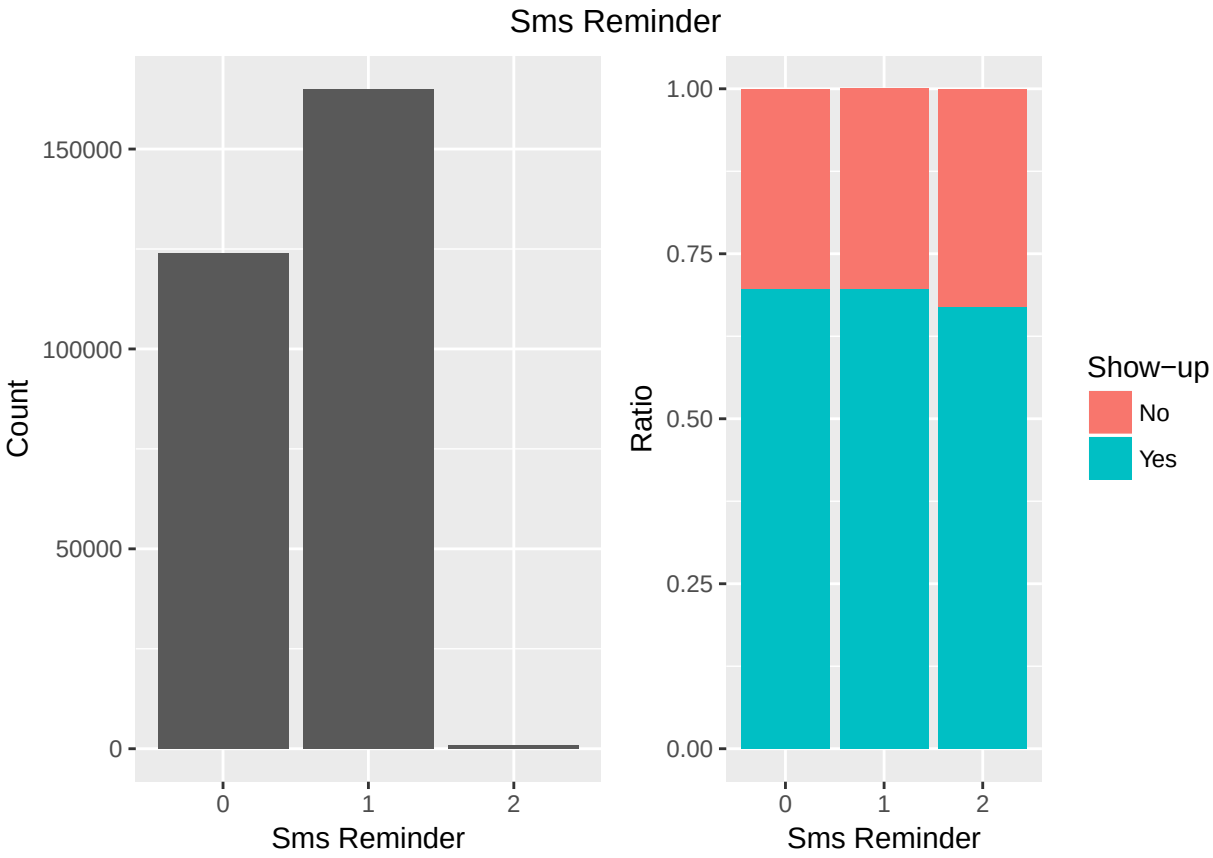


Form this first anaysis, we can conclude that people having a handicap are more likely to show up, going further the more severe is there handicap the more likely they are going to show up. The same trand is observed for people havnig diabete or hypertension.
However for smokers, alcoholic people and people suffering from tuberculosis, the trend is reverse.
To validate this insights, we need to look at the samples size, which we are going to do below.



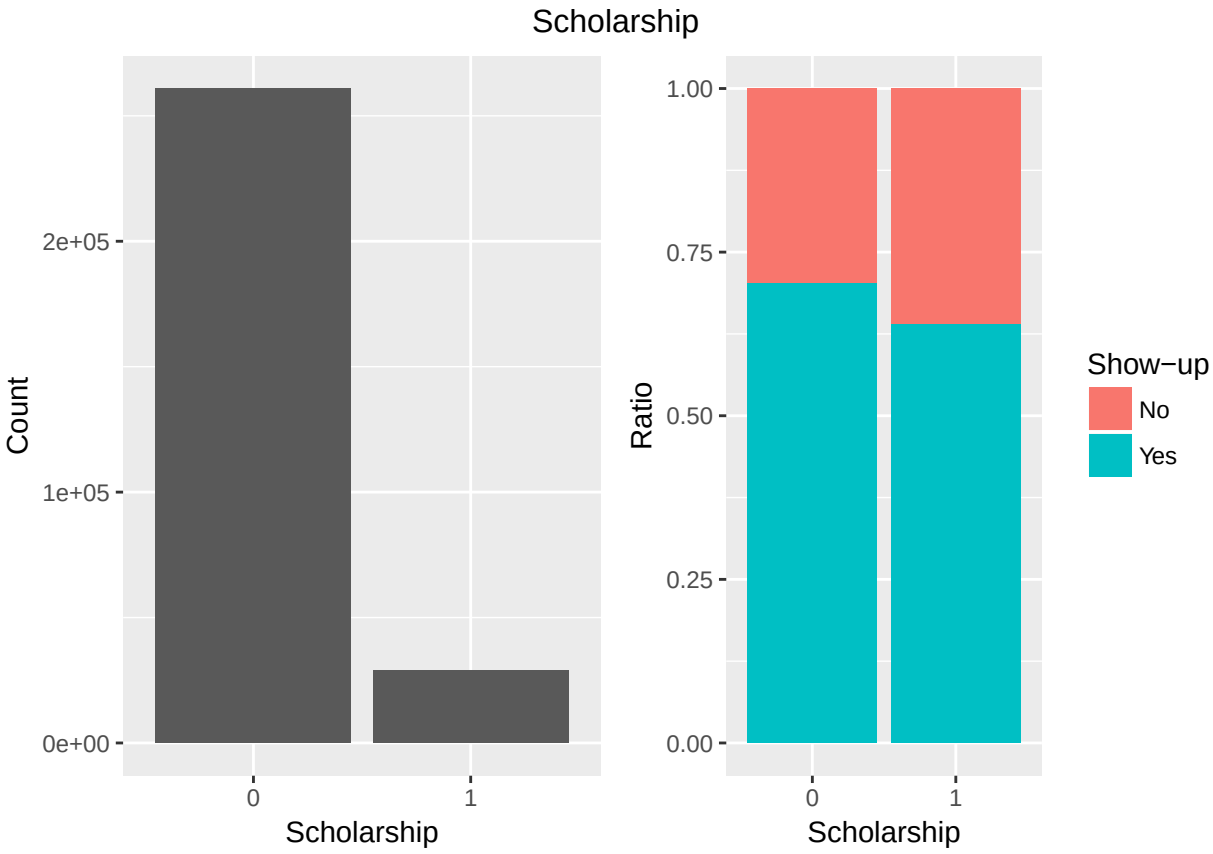
As we were expected, we always have less sick people. The number of people having tuberculosis is so low that the ratio we analyzed earlier should not to be taken into account.

SMS Reminder



There is no strong pattern associated with the sending of a sms reminder. Without taking into account people that received 2 sms (too few people), we notice that the ration is really similar. However we don't know if those sms are sent randomnly or not.

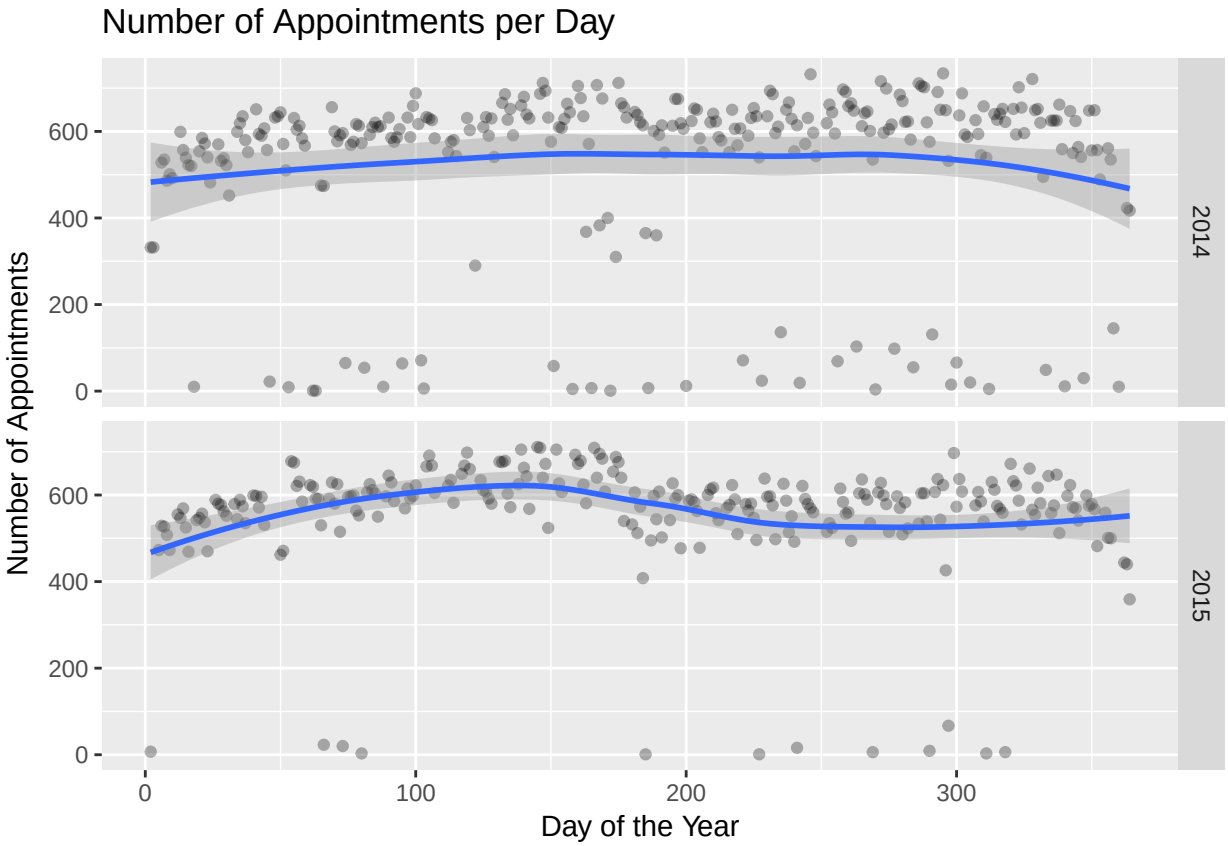
Scholarship Analysis



We did explain at the beginning of this file what scholarship meant, it's basically linked to the level of income of the family. We see that this feature have a strong impact.

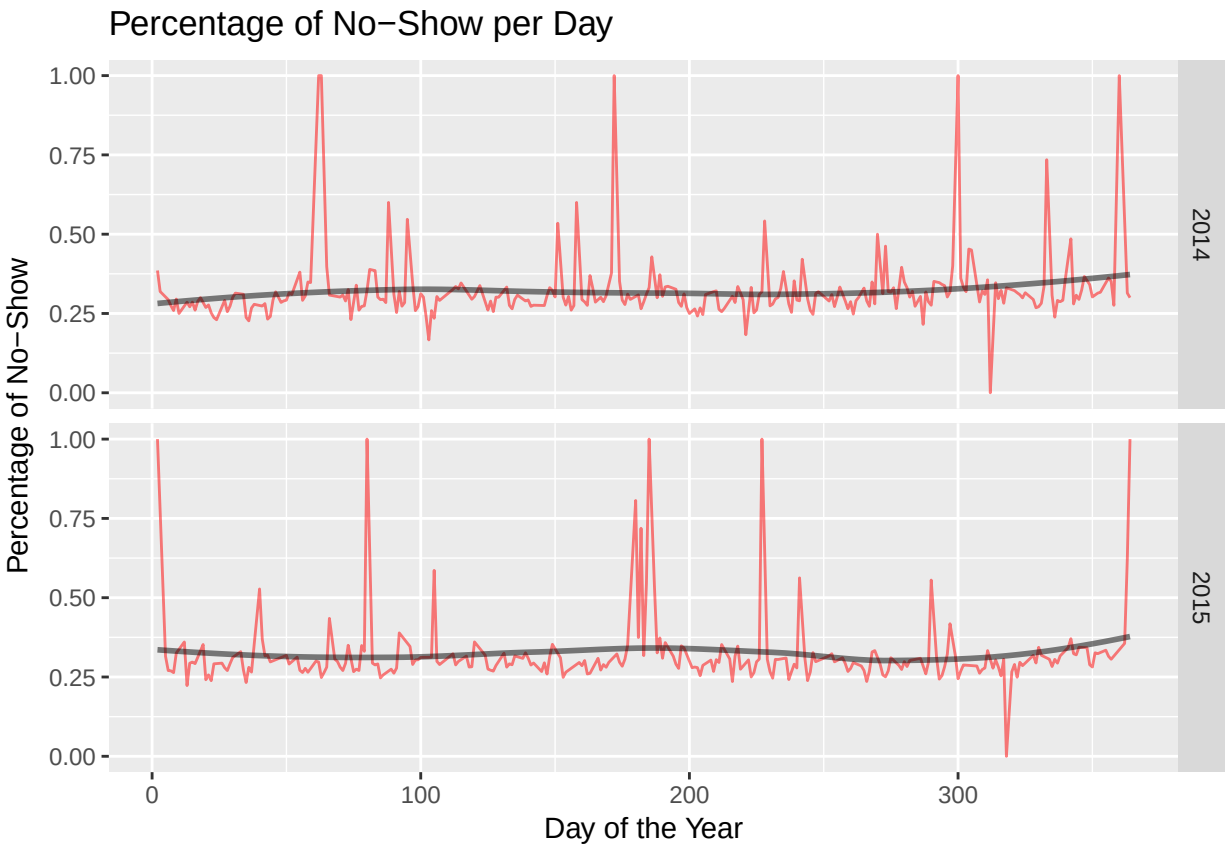
Time Analysis

Overview of the dataset per date.

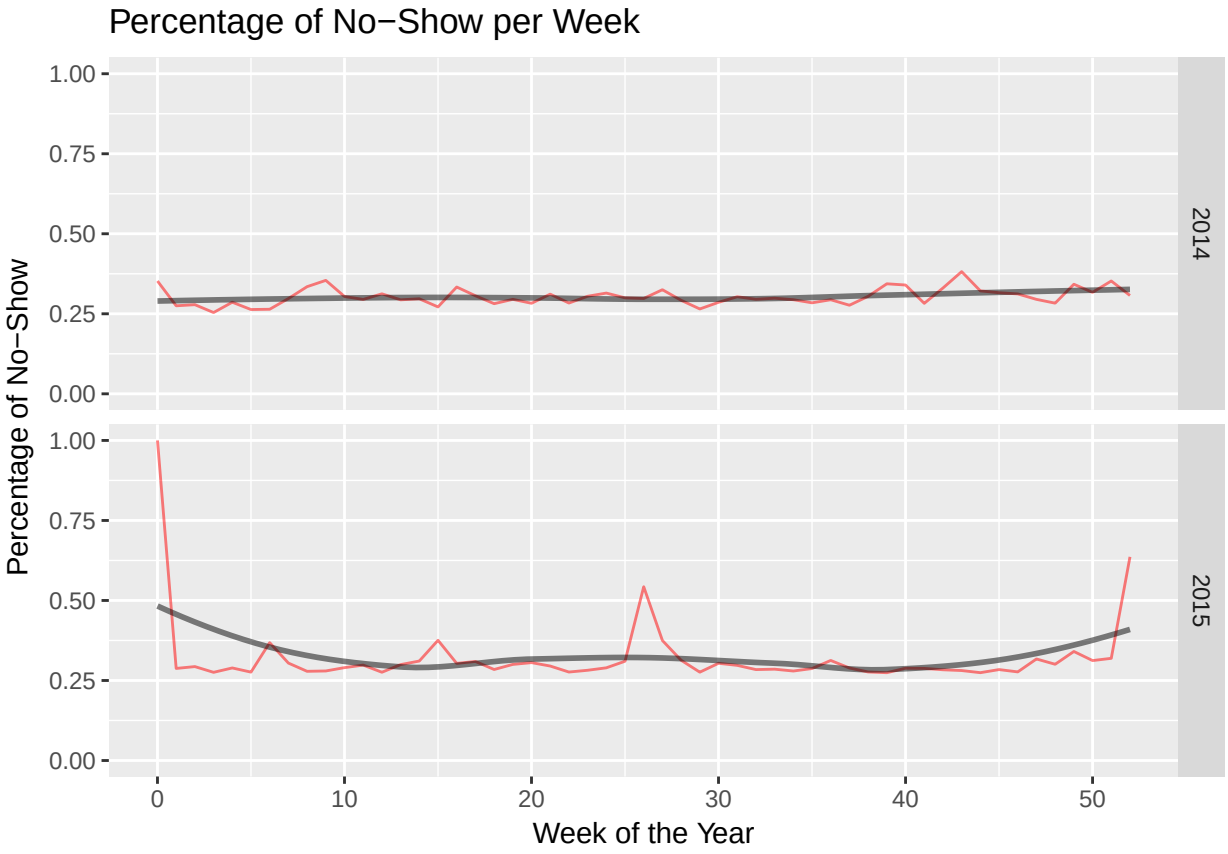


We observe a very small trend: people schedule slightly more medical appointment during the first half of the year.

Now, to explain a little bit how time influences our variable of interest, we show the evolution of the noshow ratio through the period we are studying.

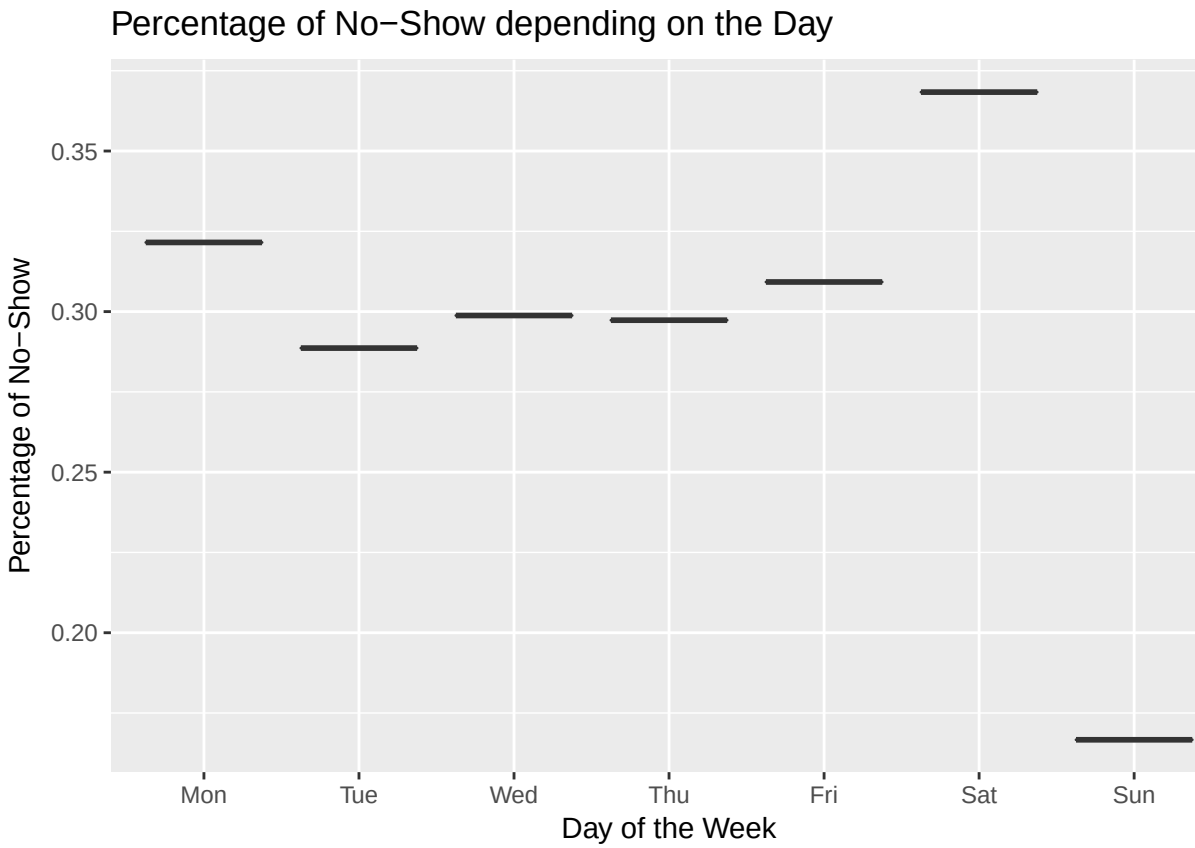


Using a daily granularity, we observe a high variance that makes the graph difficult to interpret. Let's change modify the graph by using a weekly granularity.



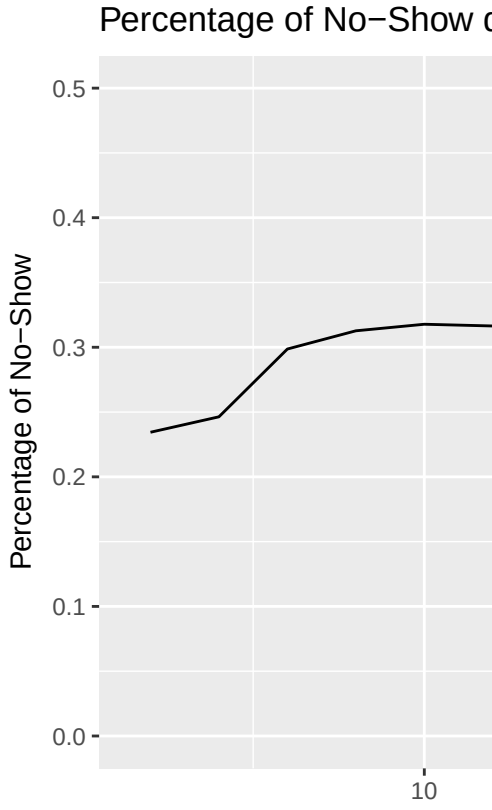
On a weekly basis we have more or less the same observations: we don't see any correlation between the date and the no-show ratio having a year window.

The natural next step is to look at the same ratio vs weekdays. It sounds pretty intuitive that there are some in-



formation to extract.

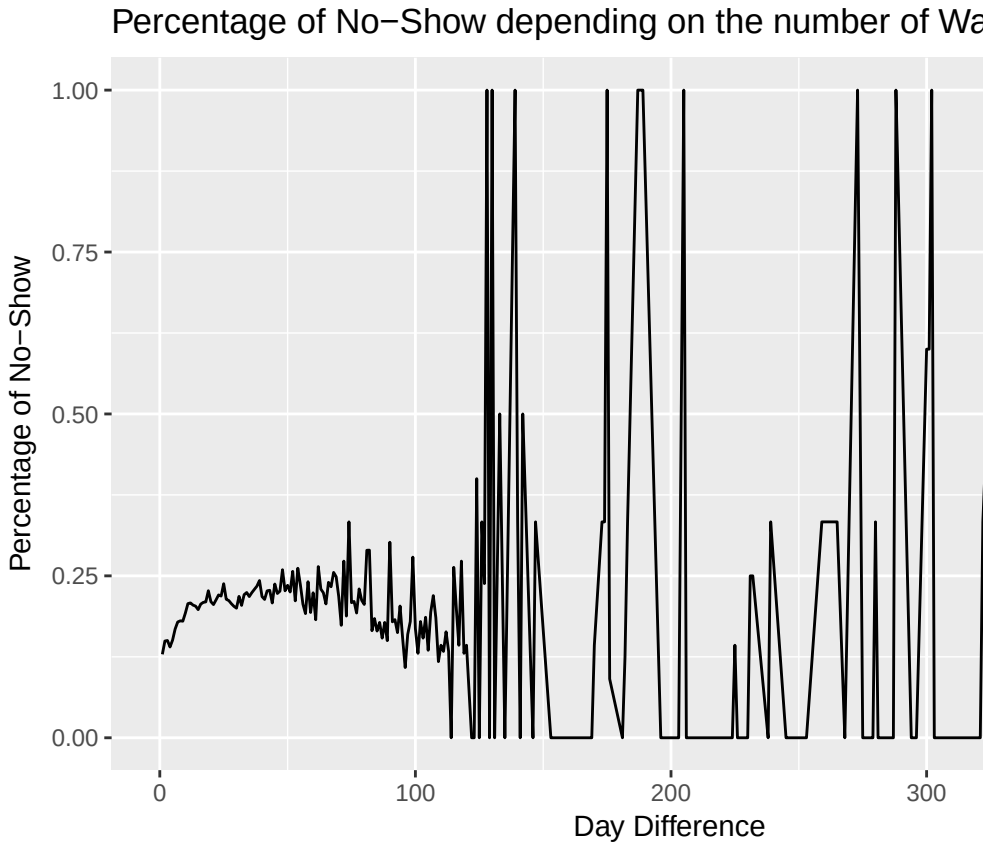
We clearly see a trend here: the ratio is extremely high on Saturday! It jumps from 30 to 37%. Sunday is not significant because very few people have appointment on Sundays.



Let us have a look ot the moment in the day the patient register for the appointment.

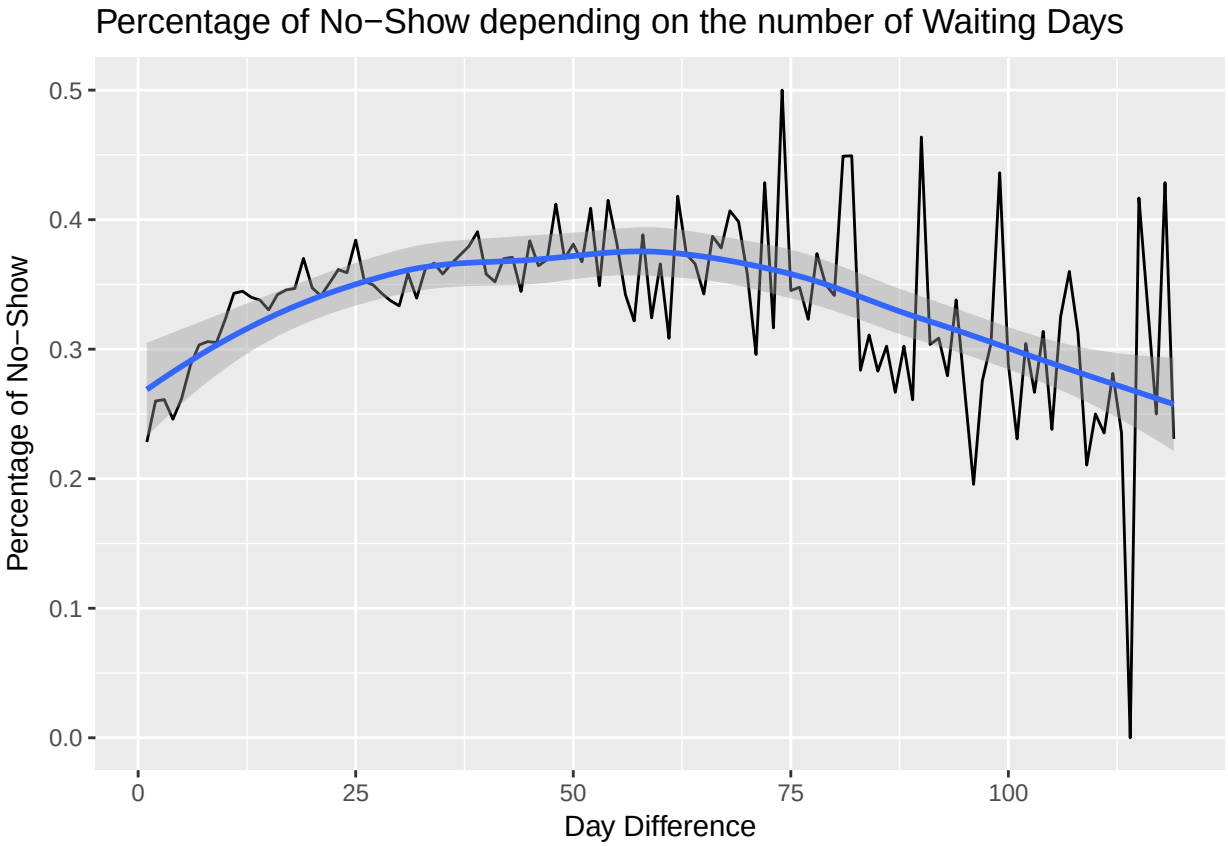
We notice that people calling very early (<6am) in the morning are more likely to show up compared to people calling late at night (>7pm).

Another important aspect in the time dimension is the time difference between the moment they register for the



appointment and the appointment date.

We first notice that the curve is extremely heratic after a certain value due to the number of datapoints. Let us zoom in the 0-120 days window. A waiting time of more than 120 days is really rare, so the second part od the curve isn't significant.



It seems that we have a good model using a polynomial regression as follow:

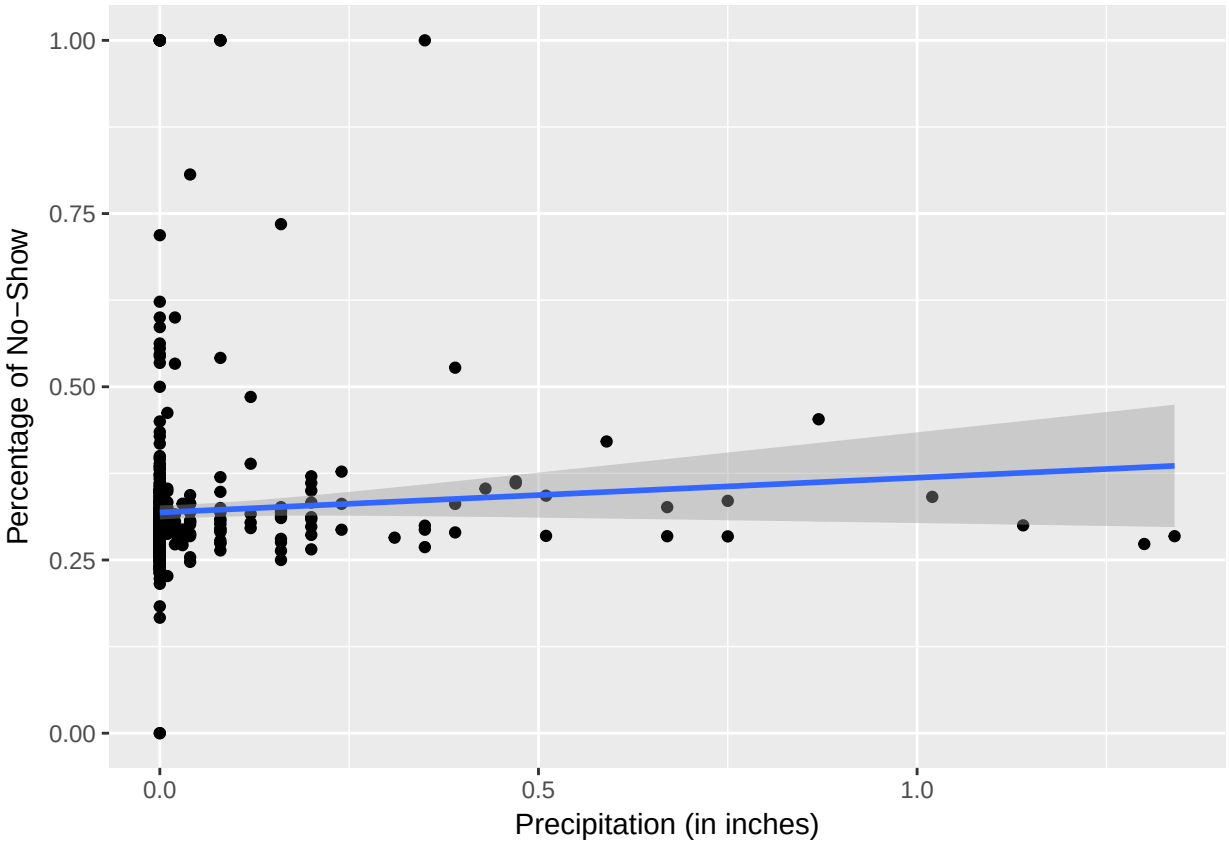
```
##
## Call:
## lm(formula = ratio ~ daydiff_regist_appt + daydiff_regist_appt_squared,
##     data = by_diff[by_diff$daydiff_regist_appt < 120, ])
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.257712 -0.026016 -0.000109  0.016365  0.186608
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  2.807e-01  1.491e-02  18.834  < 2e-16 ***
## daydiff_regist_appt          3.407e-03  5.734e-04   5.941 3.04e-08 ***
## daydiff_regist_appt_squared -3.165e-05  4.629e-06  -6.838 3.99e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05329 on 116 degrees of freedom
## Multiple R-squared:  0.319,  Adjusted R-squared:  0.3073
## F-statistic: 27.17 on 2 and 116 DF,  p-value: 2.093e-10
```

We see that the polynomial model fit pretty well this part of the curve with 2 p-values extremely low, which is good.

Weather Analysis

In this part, we want to assess the impact of the weather on attendance of patients. We webscraped data

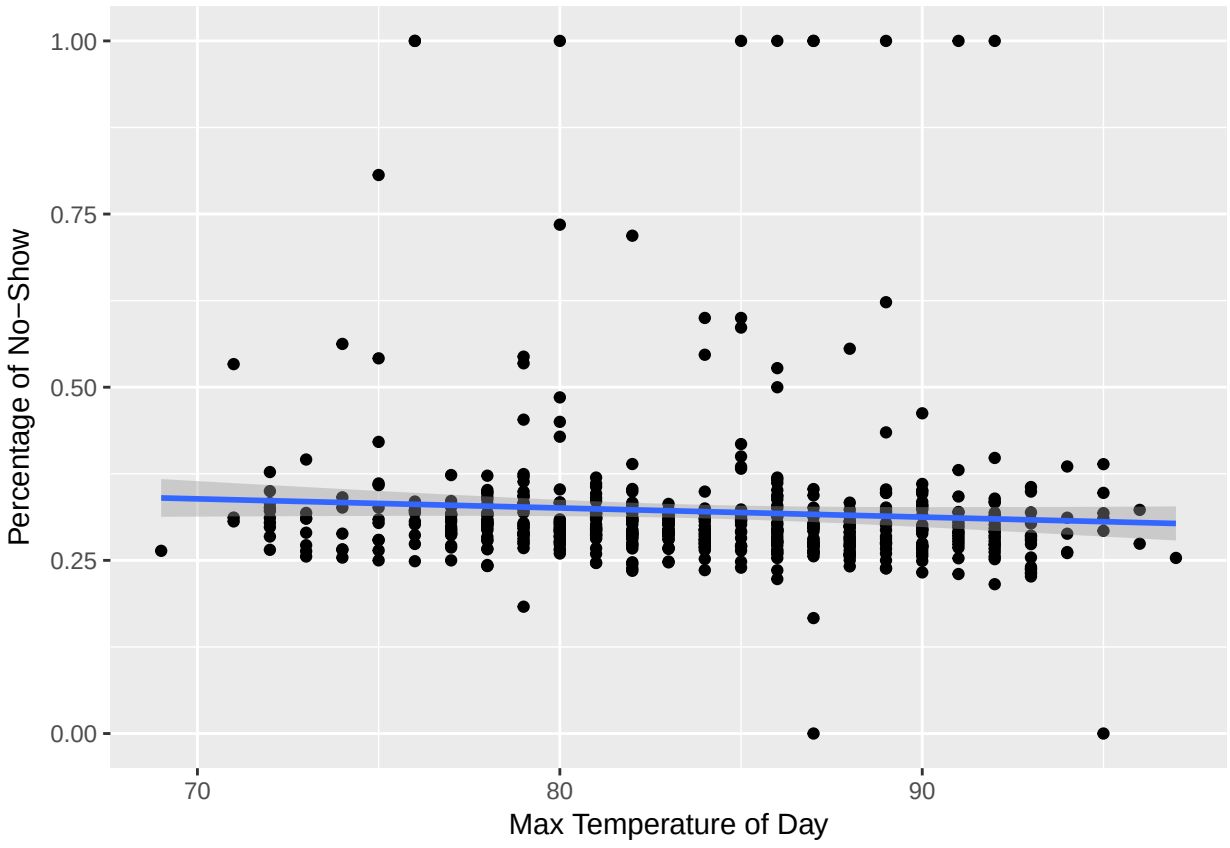
about the weather in this specific city during the period of the analysis; and we look for any correlation between the level of precipitation, the temperature, the visibility, the wind speed and the noshow ratio previously described.
The webscrabing code can be foudn in the lib folder (Weather scrapping.ipynb).



Precipitation level

```
##
## Call:
## lm(formula = ratio ~ max_precipitation_inches, data = by_temp[by_temp$max_precipitation_inches <
##     2, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31854 -0.04438 -0.02180  0.00055  0.68146
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.318544   0.005194  61.332   <2e-16 ***
## max_precipitation_inches 0.050223   0.034358   1.462    0.144
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1154 on 530 degrees of freedom
## Multiple R-squared:  0.004015,   Adjusted R-squared:  0.002136
## F-statistic: 2.137 on 1 and 530 DF,  p-value: 0.1444
```

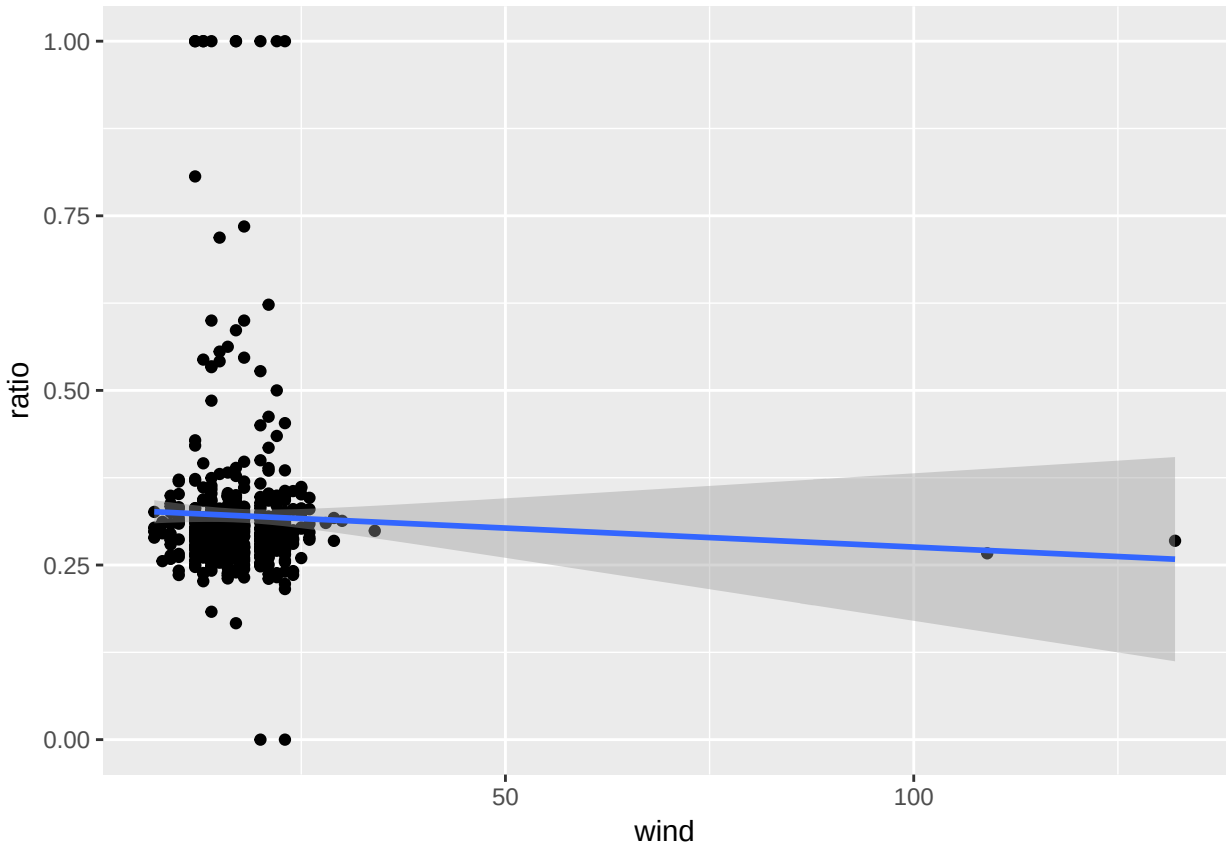
Each point on the graph represent one day. We note that there is no strong correlation between the ratio and the precipitation level.



Temperature

```
##
## Call:
## lm(formula = ratio ~ max_temp, data = by_temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31642 -0.04249 -0.02391  0.00185  0.69016
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.4309307  0.0733169   5.878 7.35e-09 ***
## max_temp    -0.0013162  0.0008726  -1.508    0.132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1151 on 532 degrees of freedom
## Multiple R-squared:  0.004259,   Adjusted R-squared:  0.002387
## F-statistic: 2.275 on 1 and 532 DF,  p-value: 0.132
```

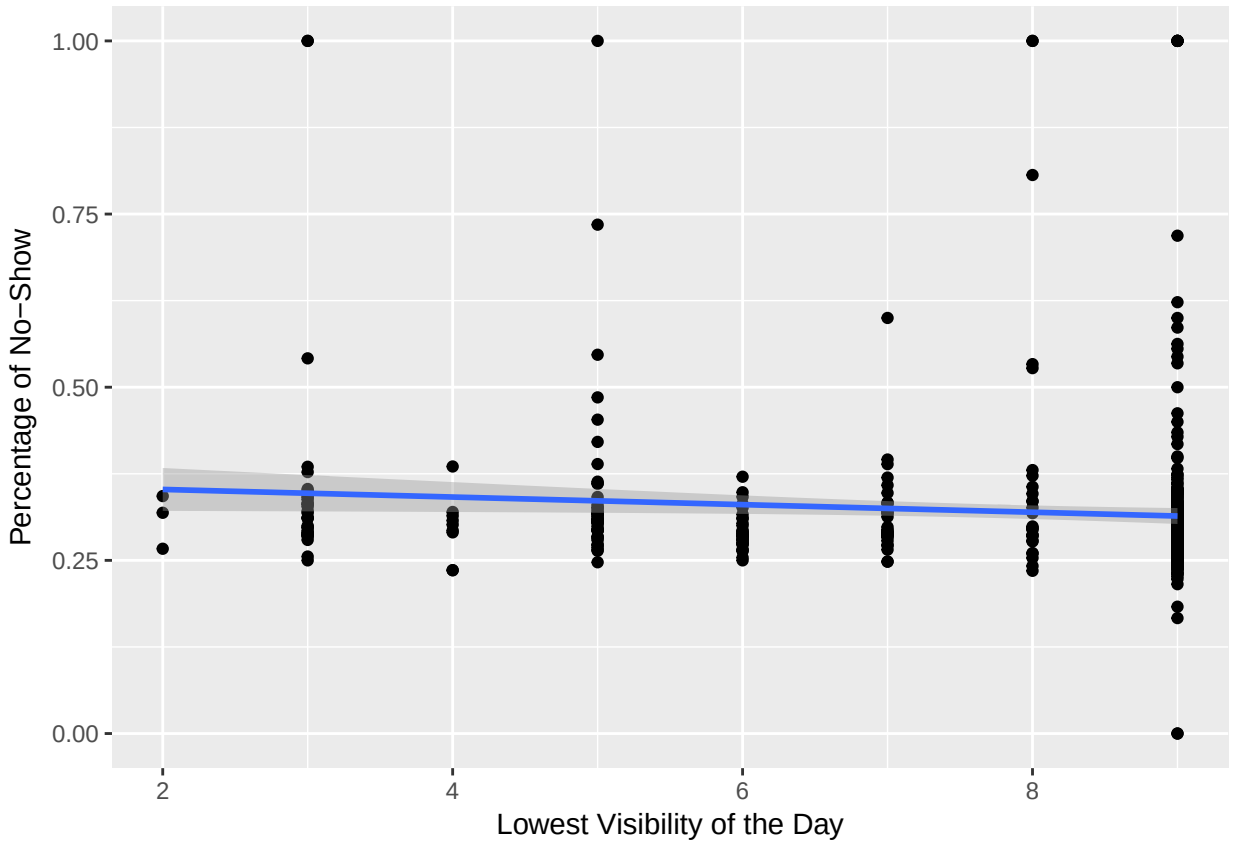
Same result with the temperature.



Wind

```
##
## Call:
## lm(formula = ratio ~ wind, data = by_temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31920 -0.04426 -0.02205  0.00198  0.68243
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.3300648  0.0123503  26.725   <2e-16 ***
## wind        -0.0005432  0.0006481  -0.838    0.402
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1153 on 532 degrees of freedom
## Multiple R-squared:  0.001319,   Adjusted R-squared:  -0.0005585
## F-statistic: 0.7025 on 1 and 532 DF,  p-value: 0.4023
```

Just like the two first variables, the wind speed is not correlated with the No-Show ratio.



Visibility

```
##
## Call:
## lm(formula = ratio ~ lowest_visibility, data = by_temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31402 -0.04358 -0.02190  0.00146  0.68598
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.363375   0.020717  17.540   <2e-16 ***
## lowest_visibility -0.005483   0.002578  -2.127   0.0339 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1149 on 532 degrees of freedom
## Multiple R-squared:  0.008433,   Adjusted R-squared:  0.00657
## F-statistic: 4.525 on 1 and 532 DF,  p-value: 0.03387
```

Visibility IS correlated with the ratio. We observe that the ratio decrease when the lowest visibility of the day increase. We obtain a good p-value of 3% for this correlation.

We conclude from this analysis that among all weather variables we found, the only feature that is more strongly correlated with the ratio is the lowest visibility of the day.

Prediction Part

Goal:
Now that we understand the characteristics of the dataset and the significance of explanatory variables

through plots and regression analysis, we now move on to trying to predict the probability of a given patient to not show up. The motivation behind this is 2-fold:

1) If we know that a patient is very likely to not show-up given some characteristics at the very point he/she books an appointment, we can incentivise the patient to show-up or to perhaps redirect the patient to a certain part of the day to facilitate handling no-shows on an aggregate level.

2) If we know the number of patients who are very likely to not show up, we can come up with several strategies which are similar to overbooking so as to increase utilization of resources.

Feature Creation

Seperation Train/Test set

Imbalanced Classes

Our dataset is imbalanced and we need to keep that in mind when trying to predict the number of no-shows. This imbalanceness is totally normal and was expected (most patient do show-up and that's perfectly normal). However, having an imbalanced dataset will make the model comparison task more difficult.

```
## [1] 69.69669
```

69.70% of the patients do show up. As a result, predicting that all patient will show-up will guve an accuracy of 69.70%. Obviously this model is not satisfactory.

Hence accuracy may not be the best metric to focus on when comparing a model. (See Accuracy Paradox (https://en.wikipedia.org/wiki/Accuracy_paradox)).

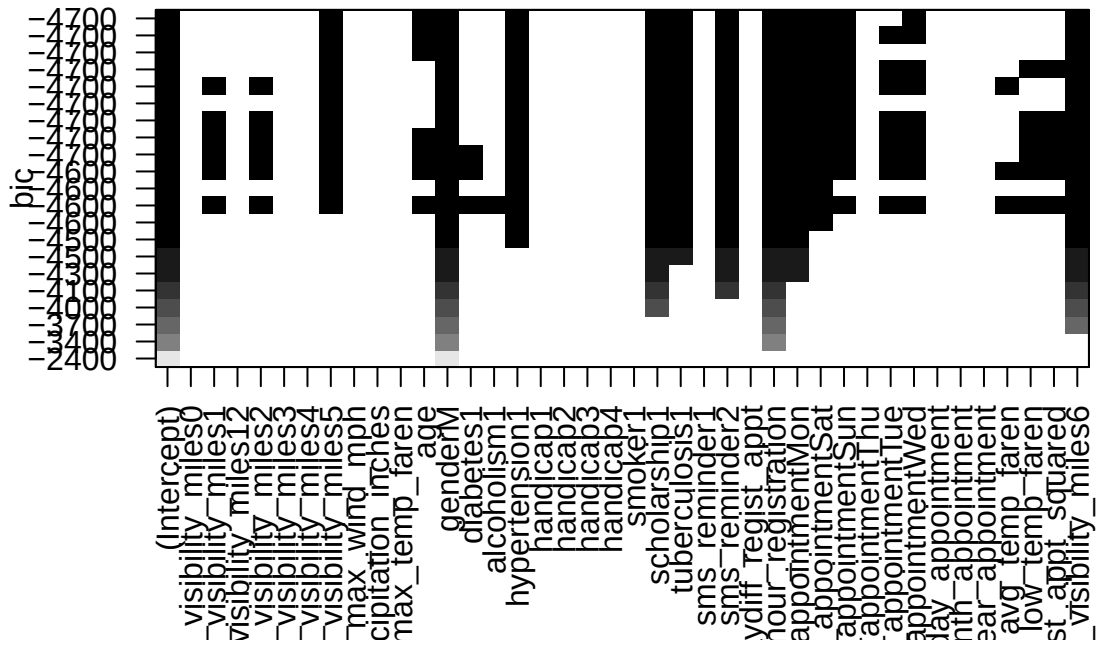
Few metrics or methods we can use are the following:
- Recall
- Precision
- ROC Curve

Feature Selection

In this section, we will try to discover which features are the most important. Hence, we will be able to verify if we get the same conclusions as in the exploratory data analysis.

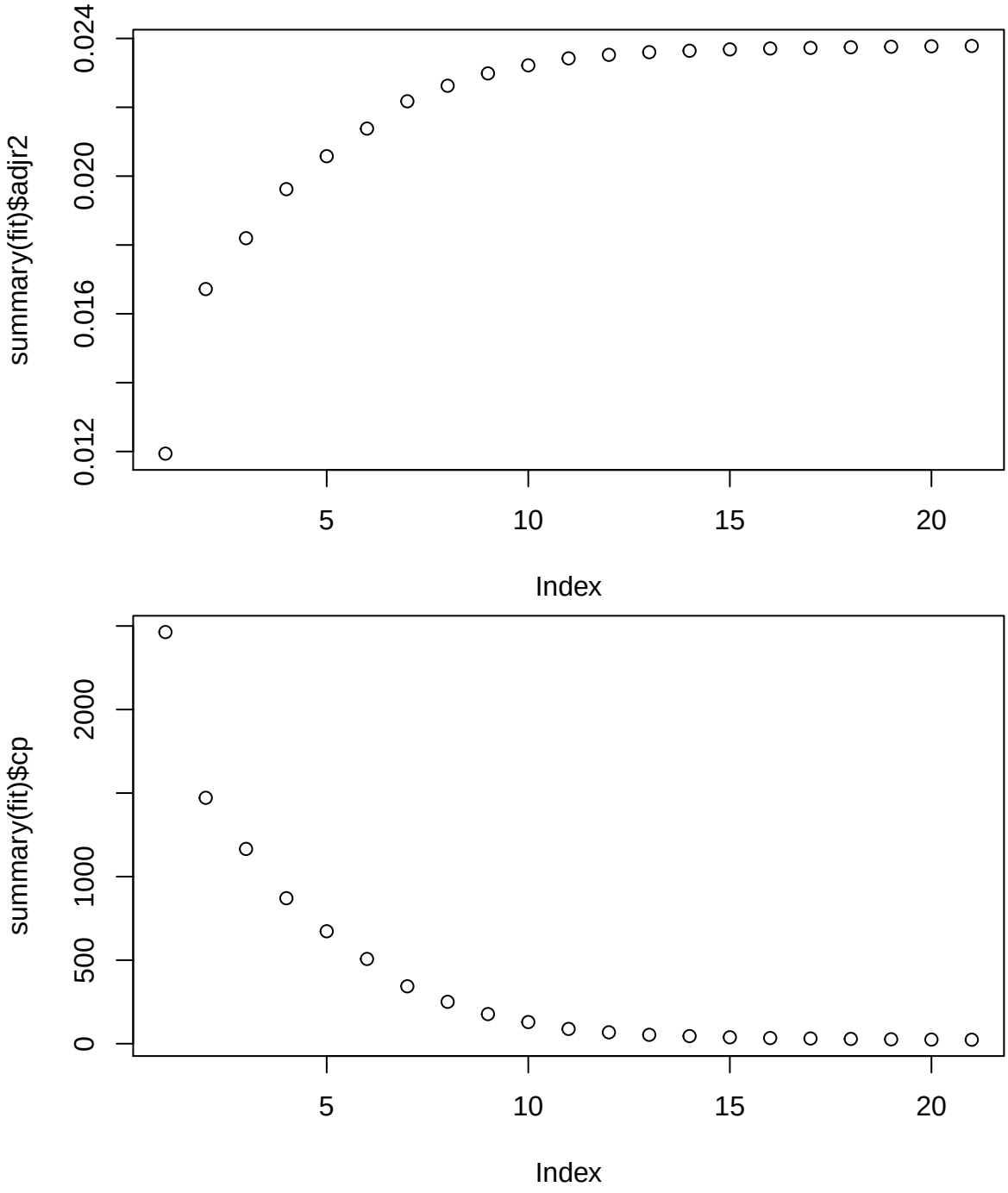
```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

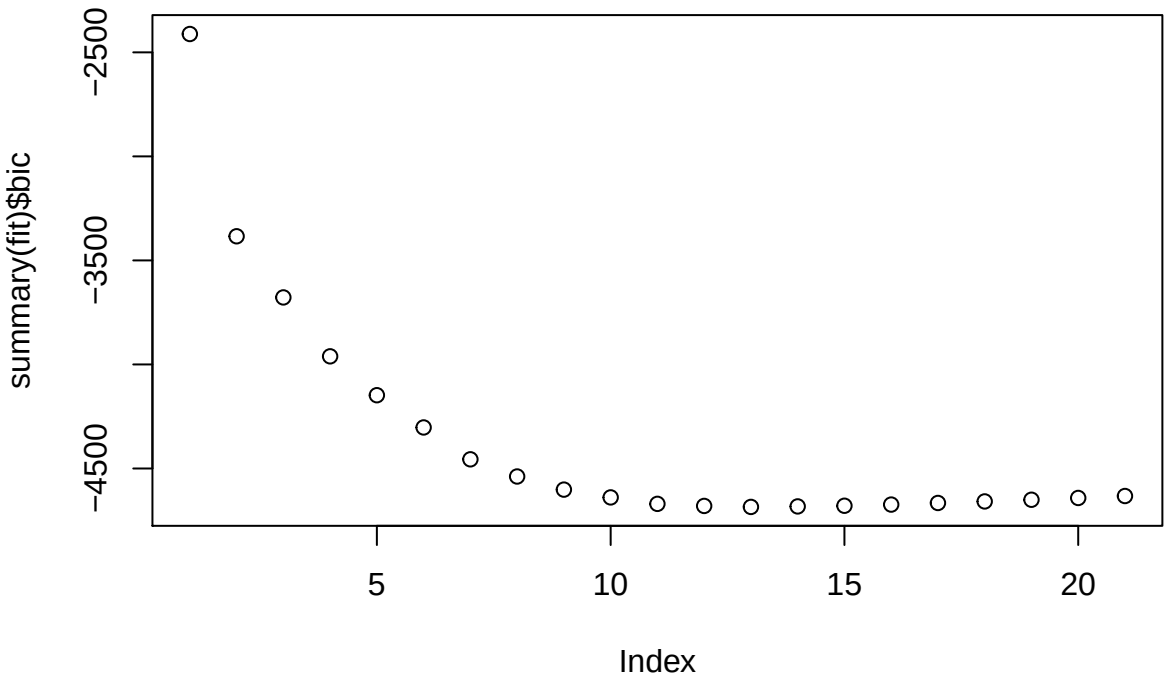
```
## Reordering variables and trying again:
```



Important variables (by decreasing order):
- age
- daydiff_regist_appt
- daydiff_regist_appt_squared
- hour_registration
- smoker1
- scholarship1
- alcoholism1
- sms_reminder1
- appointment_Mon
- appointment_Sat

Those was the one having the more impact in the Exploratory Data Analysis. Those results aren't a surprise.


summary(fit)$adjr2
0.012
0.016
0.020
0.024
5
10
15
20
Index
summary(fit)$cp
0
500
1000
2000
5
10
15
20
Index



We infer from those graphs that there is no optimum number of variables below 15 using regressions; in addition, the 9 first variables listed above seem to extract most of what can be extracted using linear regression. It is not a surprising result considering our previous analysis and the fact that most of our features are factor variables.

However, this analysis straighten some of our previous points: age, daydiff_regist_appt and hour_registration are important features that explain a lot the show/noshow ratio.

Let us now focus on the predictive model. Apparently, after this preliminary analysis, linear correlation don't explain enough to use it as a predictive model. Thus, let us focus on tree-based algorithm.

Tree-base method

```
##
## Call:
##  randomForest(formula = show_up ~ ., data = df.train.sub, importance = TRUE,      mtry = 2, ntree =
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 29.89%
## Confusion matrix:
##      0      1 class.error
## 0 2097  59349 0.965872473
## 1 1258 140065 0.008901594
```

```
##     lowest_visibility_miles                max_wind_mph
##                   831.77828                  1336.81188
##        precipitation_inches               max_temp_faren
##                   518.31860                  1378.98441
##                         age                      gender
##                  4000.89978                   528.50485
##                    diabetes                  alcoholism
##                   263.10770                   194.35787
##                hypertension                    handicap
```

```
##                   408.56032                   211.91503
##                      smoker                 scholarship
##                   267.23701                   326.82274
##                tuberculosis                sms_reminder
##                    12.16787                   497.80451
##         daydiff_regist_appt           hour_registration
##                  2361.60283                  2334.56713
##         weekday_appointment             day_appointment
##                  1054.99827                  1615.84919
##           month_appointment     day_of_year_appointment
##                  1039.42099                  1807.36205
##              avg_temp_faren              low_temp_faren
##                  1196.50129                  1311.29937
## daydiff_regist_appt_squared
##                  2343.47457
```

The more important variables are the same as the on highlighted above:
- Age
- Hour of registration
- The number of days between the registration and the appointment

Let's look at this model on the test set:

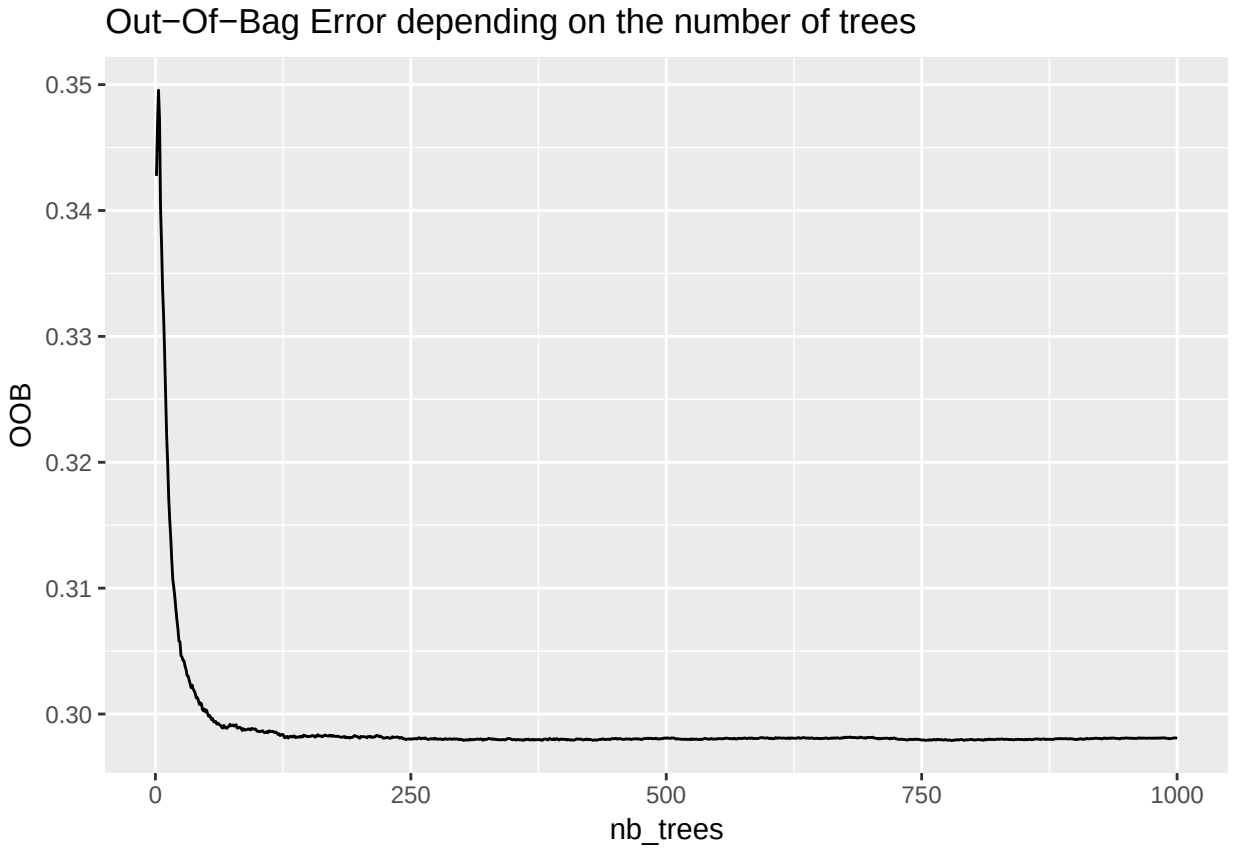
```
##
## rf.pred     0     1
##       0   757   379
##       1 25576 60187
```

```
## [1] 0.2986801
```

The test error rate is 29.8%. Looking at the confusion matrix, two third of the predicted No-Shows are correctly predictedbut overall we predict only 1,000 No-Shows while there are around ~26,000 No-Shows to predict.

At this point we need to wonder which is more important - detecting all no-shows at a lower confidence or detecting less no-shows but at a higher confidence? We will return to this question later.

We now try to find the optimal parameters for the Random Forest model (note that computer might crash with too many trees):



The out-of-bag error is a good approximation of the out of sample event. It allows us to not use cross-validation which necessites a huge computation power.
We see that augmenting the number of trees isn't helping much. The out-of-bag error is bounded to ~29%.

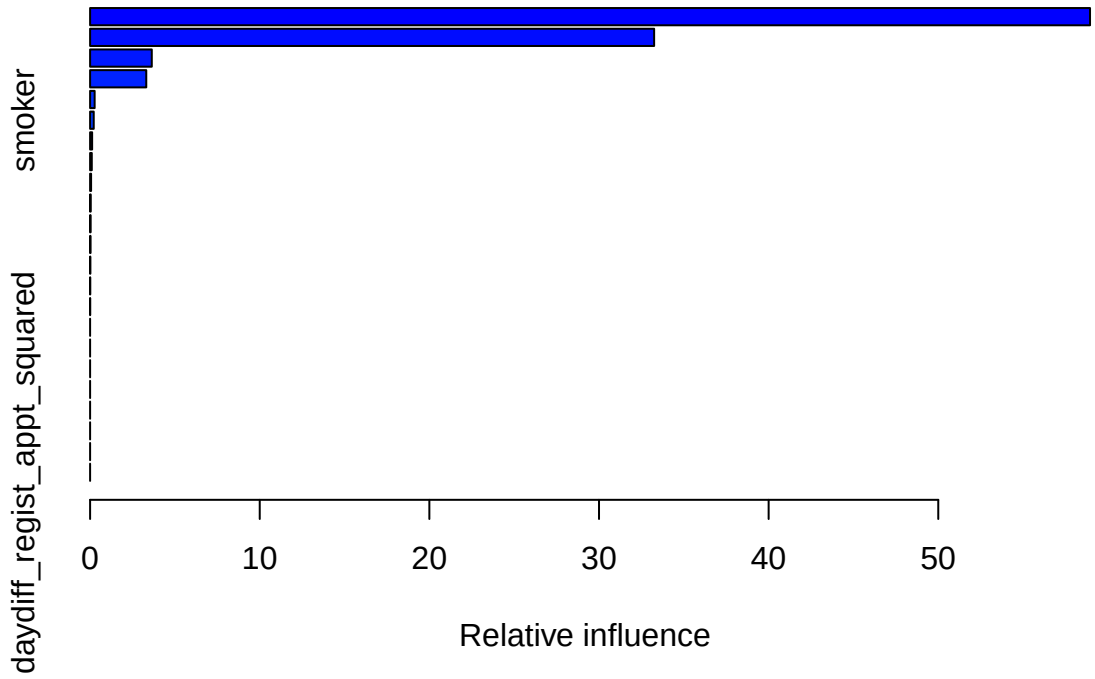
We can also try to use more sub-trees.

```
##
## Call:
##  randomForest(formula = show_up ~ ., data = df.train.sub, importance = TRUE,      mtry = 3, ntree = 
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 30.95%
## Confusion matrix:
##      0      1 class.error
## 0 7253  54193  0.88196140
## 1 8556 132767  0.06054216
```

Increasing the number of splits increases the out-of-bag error which means is it will lead to overfitting.

Boosting
Given that the predictions are not very different.

```
## gbm(formula = show_up ~ ., distribution = "adaboost", data = df.train.sub,
##     n.trees = 100, interaction.depth = 4, shrinkage = 0.01)
## A gradient boosted model with adaboost loss function.
## 100 iterations were performed.
## There were 23 predictors of which 15 had non-zero influence.
```




```
##                                                        var      rel.inf
## age                                                    age 58.946058714
## daydiff_regist_appt                    daydiff_regist_appt 33.249578964
## sms_reminder                                sms_reminder  3.636509017
## hour_registration                      hour_registration  3.311398148
## day_of_year_appointment          day_of_year_appointment  0.267080824
## smoker                                            smoker  0.213495594
## avg_temp_faren                              avg_temp_faren  0.119856475
## weekday_appointment                    weekday_appointment  0.098302758
## scholarship                                  scholarship  0.057753259
## low_temp_faren                            low_temp_faren  0.033064058
## max_wind_mph                                max_wind_mph  0.026748994
## alcoholism                                    alcoholism  0.022688807
## lowest_visibility_miles          lowest_visibility_miles  0.010417829
## day_appointment                            day_appointment  0.004771559
## precipitation_inches                  precipitation_inches  0.002275000
## max_temp_faren                              max_temp_faren  0.000000000
## gender                                              gender  0.000000000
## diabetes                                          diabetes  0.000000000
## hypertension                                hypertension  0.000000000
## handicap                                        handicap  0.000000000
## tuberculosis                                tuberculosis  0.000000000
## month_appointment                      month_appointment  0.000000000
## daydiff_regist_appt_squared daydiff_regist_appt_squared  0.000000000
```

```
## [1] 0.08457943
```

The training error is very high.

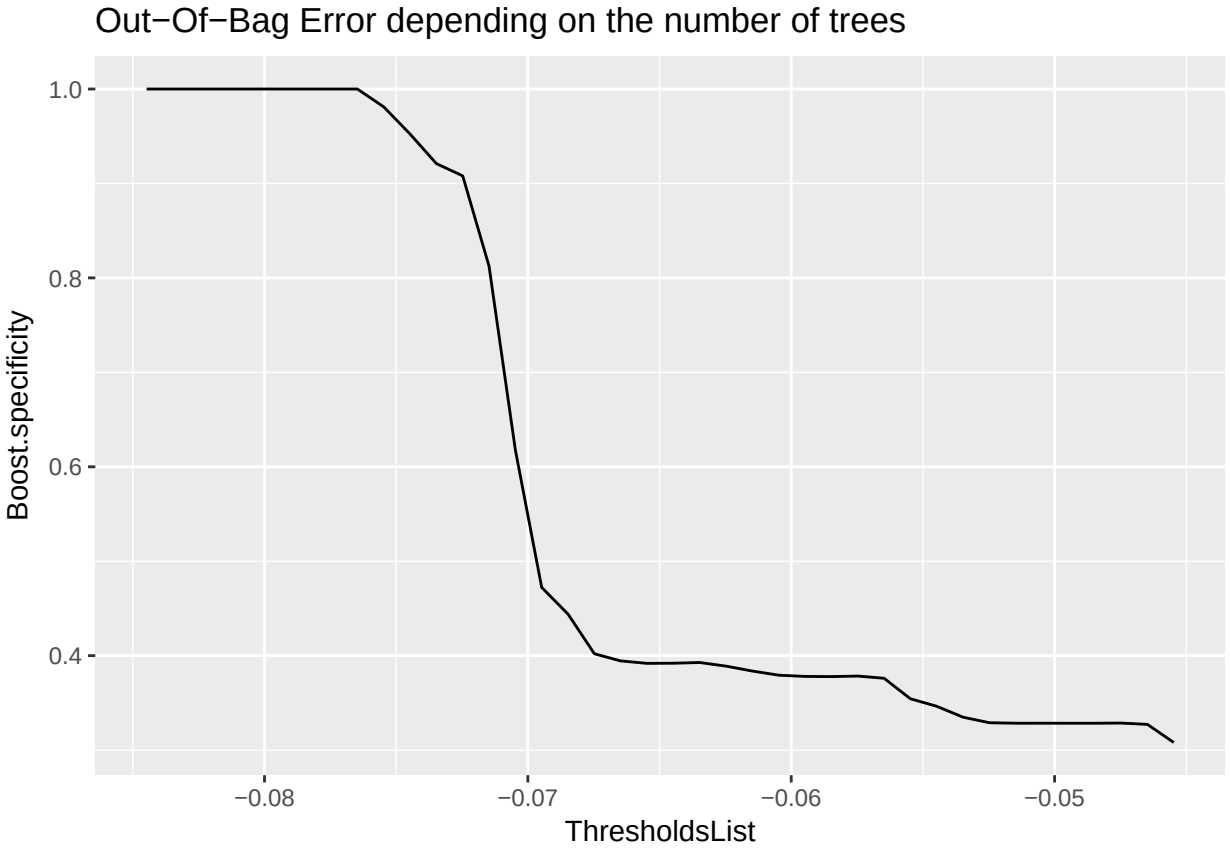
Prediction on the test set.

```
##
## boost.pred     0     1
##          1 26333 60566
```

```
## [1] 0.30303
```

The boosting model doesn't help. It predicts 1 every time because the predicted probabilities are very high. We can try to change the threshold.

Now we try to optimize the threshold to select for high specificity/true negativity rate - ultimately, we want a high confidence of correctly predicting the number of no-shows for any given day.
Hence, we are not looking at accuracy but at specificity.



Some threshods give very good values of specificity. However, each threshold below 0.07 is able to predict less than 50 No-Shows out of ~26,000 No-Shows which is not helpful.

Business Value

To evaluate the value of building an effective prediction model, we can look at use cases at the individual and day-aggregated level.

At the individual level, we can use a prediction model which only includes features that is fully known at the point of appointment booking (no weather in this case) to categorize if a patient is considered higher-risk above a certain threshold. This allows us to do two things: 1. Focus resources on providing more interventions (even more sms and calls) to increase likelihood of patient showing up. Or to confirm if patient is not going to show-up 2 days before appointment. 2. Give dollar incentives to channel likely no-shows into morning/evening appointment slots so that most of the no-shows are clustered around a certain period of the day.

At the day-aggregated level, we can use a prediction model which INCLUDES weather features given by meteorological forecasts, that is able to give us the number of no-shows for a given day at a 95% confidence level. This can be computed 2 days before the actual appointment date.

All this information from above can help us implement a "pseudo-overbooking" policy: an opt-in waitlist.

The opt-in waitlist can be generated for each appointment day, 2 days in advance and the length of the list = # people who rescheduled + (conservative) estimate of # predicted no-shows. Given that we have "consolidated" in advance the likely no-shows into a certain part of the day, we can ask the wait-listers to come at a specific window if they are willing to take the risk to have an earlier appointment.

Even if we are only able to cut the number of no-shows from 30% to 25%, we can already generate an estimated savings of $0.5 million for a community hospital. The savings will be even greater for a larger hospital. Furthermore, we can generate customer satisfaction amongst those wait-listers who manage to get an earlier appointment.