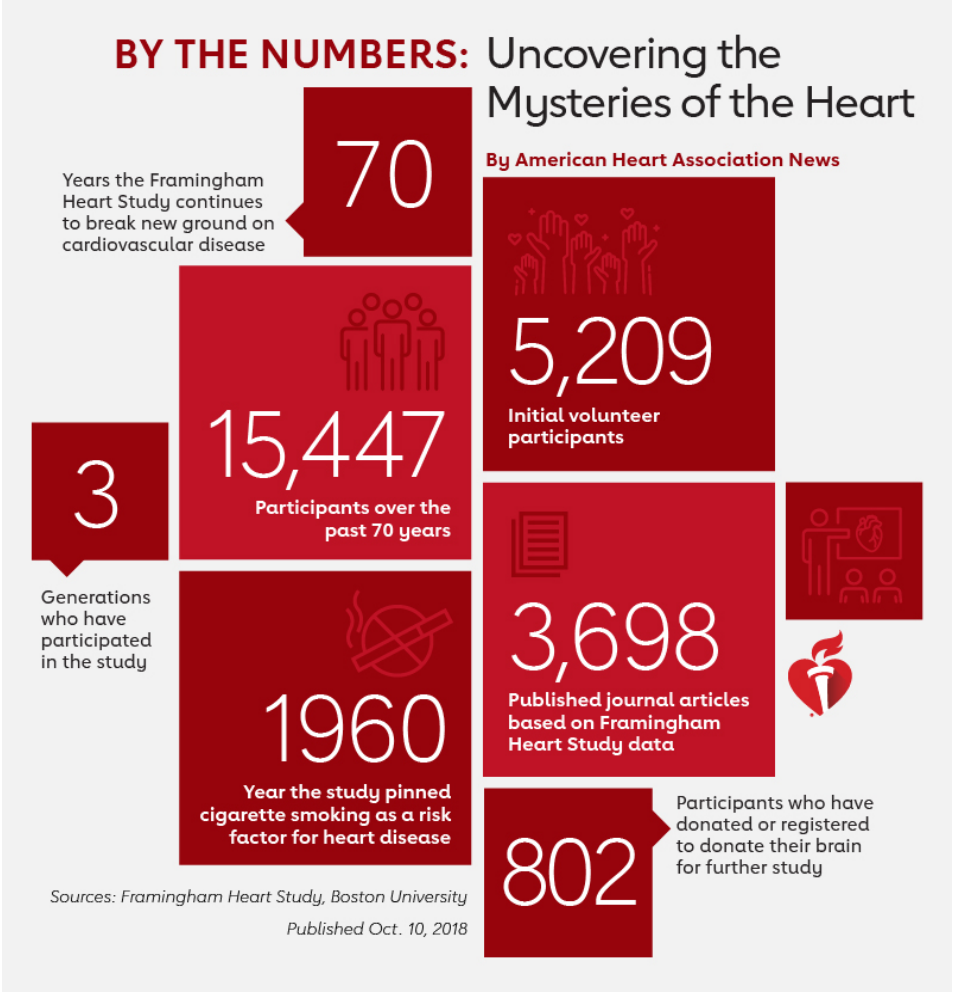
Lab 3

The Framingham Heart Study is a long term prospective study of cardiovascular disease among a population of subjects in the community of Framingham, Massachusetts. The Framingham Heart Study was a landmark study in epidemiology in that it was the first prospective study of cardiovascular disease and identified the concept of risk factors and their joint effects over the course of three generations. The study began in 1948 and 5,209 subjects were initially enrolled in the study. Participants have been examined biennially since the inception of the study and all subjects are continuously followed through regular surveillance for cardiovascular outcomes.

You will find the data file ***framinghamHeart.csv***, which you can load as **`dff`**. This is a subset of the data collected as part of the Framingham study. Participant clinic data was collected during three examination periods, approximately 6 years apart, from roughly 1956 to 1968. Each participant was followed for a total of 24 years for the outcome of a specified set of adverse health events. The dependent variable is **TenYearCHD**, specifying whether a subset of events associated with chronic heart disease occurred within 10 years of follow up. The variables are defined below. The purpose of the study is to determine the risk factors of heart disease.



Data Dictionary

Variable	Description	Coding
gender	Male or Female	0 = Female; 1 = Male
age	Age of the patient	
education	Highest level of education achieved	1 = High School; 2 = High School Diploma or GED; 3 = Some college or vocational School; 4 = College degree
currentSmoker	Indicates if the person is currently a smoker or not	0 = Not a smoker; 1 = Is a smoker
cigsPerDay	The number of cigarettes the person smoked on average in one day	
BPMeds	Whether the patient was on blood pressure medication	0 = Not on BP meds; 1 = On BP meds
prevalentStroke	Whether the patient previously had a stroke	0 = Free of disease; 1 = Stroke
prevalentHyp	Whether the patient has hypertension (high blood pressure)	0 = Free of disease; 1 = Hypertension
diabetes	Whether the patient has diabetes	0 = Free of disease; 1 = Diabetes
totChol	Total cholesterol level	mg/dL
sysBP	Systolic blood pressure	mmHg
diaBP	Diastolic blood pressure	mmHg
BMI	Body Mass Index	Weight (kg) / Height (meter-squared)
heartRate	Heart rate	Beats/Min (Ventricular)
glucose	Glucose level	mg/dL
TenYearCHD	Coronary heart disease	‘0' indicates the event did not occur during the 10-year follow

		up, and ‘1' indicates an event did occur during the follow up

Data Analysis

Before you start, **load the “caret” library** in addition to the usual four libraries we always load.

In addition, pay attention to what R reports after you load the dataset:

```
Parsed with column specification:
cols(
  gender = col_double(),
  age = col_double(),
  education = col_double(),
  currentSmoker = col_double(),
  cigsPerDay = col_double(),
  BPMeds = col_double(),
  prevalentStroke = col_double(),
  prevalentHyp = col_double(),
  diabetes = col_double(),
  totChol = col_double(),
  sysBP = col_double(),
  diaBP = col_double(),
  BMI = col_double(),
  heartRate = col_double(),
  glucose = col_double(),
  TenYearCHD = col_double()
)
```

Notice that R reads all the columns as numbers. You know from the data dictionary that some variables are supposed to be factors. You need to ask R to convert them into factors:

i. Create a list of columns that are supposed to be factors:

```
colsToFactor <- c('gender', 'education', 'currentSmoker', 'BPMeds',
'prevalentStroke', 'prevalentHyp', 'diabetes')
```

ii. Ask R to replace (overwrite) selected variables with their factor conversions:

```
dff <- dff %>%
  mutate_at(colsToFactor, ~factor(.))      => What do you think mutate_at does?
```

Now, if you run `str(dff)`, you will see that the variables in your data are correctly identified:

```
Classes ‘spec_tbl_df’, ‘tbl_df’, ‘tbl’ and 'data.frame':       3658 obs. of  16 variables:
 $ gender         : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 1 2 2 ...
 $ age            : num  39 46 48 61 46 43 63 45 52 43 ...
 $ education      : Factor w/ 4 levels "1","2","3","4": 4 2 1 3 3 2 1 2 1 1 ...
 $ currentSmoker  : Factor w/ 2 levels "0","1": 1 1 2 2 2 1 1 2 1 2 ...
 $ cigsPerDay     : num  0 0 20 30 23 0 0 20 0 30 ...
 $ BPMeds         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ prevalentStroke: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ prevalentHyp   : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 1 2 2 ...
 $ diabetes       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ totChol        : num  195 250 245 225 285 228 205 313 260 225 ...
 $ sysBP          : num  106 121 128 150 130 ...
 $ diaBP          : num  70 81 80 95 84 110 71 71 89 107 ...
 $ BMI            : num  27 28.7 25.3 28.6 23.1 ...
 $ heartRate      : num  80 95 75 65 85 77 60 79 76 93 ...
 $ glucose        : num  77 76 70 103 85 99 85 78 79 88 ...
 $ TenYearCHD     : num  0 0 0 1 0 0 1 0 0 0 ...
```

1. **Data exploration:** To explore visually whether blood pressure levels and total cholesterol levels are associated with heart disease, create boxplots of *sysBP*, *diaBP*, and *totChol*, broken up by the levels of *TenYearCHD*. [**Hint:** Dynamic plots may help understanding!]

Ans - sysBP, diaBP and totChol are all associated with heart disease as “no heart disease” group (represented by 0) is associated with lower value of median for sysBP, diaBP and totalChol. No heart disease group also has lower values for first and third quartile, further consolidating the association between these parameters and TenYearCHD.

2. **Data preprocessing:**

   (i) Read the data file into R. Set the seed to **123** and split the data into `dffTrain` and `dffTest`. Randomly sample 70% of the data for training, and use the rest as test dataset.

   (ii) What are the proportions by gender in your training vs. test set? How does the distribution of age look? Looking at these, do you observe any signs of a sampling bias?

   Hints:

   [A] It’s time to use R like a pro! You can pipe your *dffTrain* into the `group_by(`*`variable`*`)` function and then into **`tally()`** *-no arguments-* to get the counts across a group.

   - ❏ To add percentages, pipe one more step into `mutate(pct = 100*n/sum(n))`

   [B] For a continuous variable like age, there are so many groups, right? Each age is practically a different group. In such cases, you may want to create your own groups.

   - ❏ You can use *`ageGroup=cut_interval(age, length=10)`* in group_by()

   [C] You can also create a histogram for age, which probably makes more sense.

   - ❏ After creating the histogram, try adding `fill=gender` into aes() of ggplot(), and see what happens. In addition, define `color='black'` inside the histogram!

Ans - Comparing the percentages of gender and age groups, the distribution seems unbiased as the percentages are roughly similar across test and train groups.

Age distribution is normal, with maximum number of people in the observations in the range of

35-66 years.

3. **Linear probability model:** Build a linear probability model `fitLPM` using all variables in `dffTrain`. Make sure to check for collinearity[1] by both thinking about the variables, and using VIF values as guiding signals, and take necessary precautions. You know how to mitigate collinearity (if not, please ask during the lab!). After finalizing the model, which of the variables are statistically significant at the 95% level? What does this model tell you about the risk factors of heart disease? Do you have any reservations? Discuss.

   Hints:

   [A] To include all the variables, use a full stop **.** To exclude a variable, use a negative **-**

   [B] Run diagnostics to see whether this model violates the linear regression assumptions.

Ans - The model that includes all variables has high multicollinearity, as indicated by the vif values. The vif values are similar for cigsPerDay and currentSmoker, indicating collinearity between these. This is because, it is possible to deduct the value of currentSmoker from cigsPerDay as a value of "0" cigsPerDay indicates the person is not a currentSmoker and any other value of cigsPerDay indicates the person is a currentSmoker. Thus, we remove currentSmoker from the model.

At 95% significance level, following variables are statistically significant:

Gender, age, cigsPerDay, prevalentStroke, prevalentHyp, sysBP, heartRate and glucose.

This model indicates that these above factors have a major effect on the risk of heart disease.

All these factors look relevant, and thus it makes sense to include only these factors in the model. Other variables such as education, BPMeds etc, which do not have much effect on the heart disease risk, should be removed.

From the diagnostic plots, it can be concluded that the model violates all linear regression assumptions - Linearity, Homoscedasticity, Independence & Normality. This indicates it is not a linear model.

[1] Likely multicollinearity. If "multicollinearity problem is extreme: any variable in the model can be written as a linear combination of all of the other variables in the model. Essentially, this means that we can never know exactly which variables (if any) truly are predictive of the outcome, and we can never identify the best coefficients for use in the regression. At most, we can hope to assign large regression coefficients to variables that are correlated with the variables that truly are predictive of the outcome." ISLR p. 243

4. Speaking of using R like a pro, a better way to run a model and create a results table with predictions is as follows. Please run this code to make predictions using the LPM model and store them into *resultsLPM*[2]

```
resultsLPM <-
   lm( …fill in here… ) %>%
   predict( …fill in here… ) %>%    => Use the option type='response' for probabilities
   bind_cols(dffTest, predictedProb=.) %>%     => The dot marks where to pipe into
   mutate(predictedClass = …fill in here… )    => Use 50% as cutoff for classification
```

   Inspect `resultsLPM`. Then, **copy and paste your code from Q2-ii** and check the prevalence of *TenYearCHD* in the *test dataset* this time. How many people have heart disease in reality (in the test dataset)? Run the same code for *predictedClass* in the *resultsLPM*. How many people did the model predict having heart disease? Compare and report your observations.

   Before you continue:

   You may have noticed that we did not convert TenYearCHD into a factor yet, even though it is a factor. This is because we wanted to use it in a linear model. It is time to make it a factor.

   ❏ Use `mutate()` to convert TenYearCHD to a factor both in *dffTrain* and *dffTest* datasets.

Ans - In the test dataset, the count of people with heart disease in reality is 172.

The model predicts the count of people with heart disease as 10.

This shows the model is not a very good fit as it is unable to predict the count of people with heart diseases properly and the count is actually very less. This implies it has not captured all the variables and their interactions properly.

5. **Logistic regression:** Build a logistic regression using the predictor variables you decided to keep in the model you built in Q3. Which variables are statistically significant at the 95% level? Compare your results with the results you obtained from the model in Q3.

   Hint: See the appendix for an annotated logistic regression output in R with the definitions.

Ans - According to this model, at 95% significance level, following variables are statistically significant:

Gender, age, cigsPerDay, prevalentHyp, totChol, sysBP, heartRate and glucose.

---

[2] You can replicate this idea for any other model to make predictions -including the ones you did last week. When you are using this chunk of code for a linear regression, you don't need the last line because you don't need a conversion into classes. Instead, I would change *bind_cols(dffTest, **predictedProb**=.)* into *bind_cols(dffTest, **predictedValues**=.)* for a better understanding in a linear model.

Comparing this logistic model with the model in Q3, prevalentStroke is no longer statistically significant in the logistic model where it was in the linear model.

totChol was not statistically significant in the linear model, but it is in the logistic model.

There has been an improvement in the predicted count of people with heart disease in logistic model, which is now 19, as compared to 10 in the linear model. Although the accuracy is still very low as only 19 out of 172 cases of people having a heart disease have been predicted correctly by the logistic model.

Interpret the following variables: *age*, *gender*, and *diabetes* (whether significant or not):

- ❏ **Hint:** You can run **`exp(coef(fit))`** after a logistic model to exponentiate the coefficients of all variables at once, and use them in your interpretations.
- ❏ **Type these interpretations AFTER completing the lab unless you have any questions.**

Age: A one-year increase in the age leads to an increase in the odds of a heart disease by a factor of 1.069 (about 6.9% increase), holding everything else constant.

Gender: Odds ratio is 1.5253, which indicates the odds of male having a heart disease is about 52.5% more as compared to female having a heart disease, holding everything else constant.

Diabetes: Odds ratio is 0.99, which indicates the odds of a diabetic person having a heart disease is about 1% less as compared to non-diabetic person having a heart disease, holding everything else constant.

6. Create a new results table ***resultsLog*** by using the logistic model. Let's continue like a pro.

   Hint: You will follow the same steps you took in Q4 but this time for logistic regression. This means, **your predictedClass will need to be defined as a factor** (you know how to do this!).

   How many people did the logistic model predict having heart disease? Report your observations and compare them with the actual values, and the predictions of the linear probability model from Q4. Do you think the logistic model is an improvement? Why?

   Hint: For now, continue to use your code from Q2-ii to create the tables for comparison.

Ans - Logistic model predicts 19 people having a heart disease, whereas the actual count of people having heart disease is 172.

There has been an improvement in the predicted count of people with heart disease in logistic model, which is now 19, as compared to 10 in the linear model. Although the accuracy is still very low as only 19 out of 172 cases of people having a heart disease have been predicted correctly by the logistic model.

7. It is time to create a confusion matrix, a final step before evaluating performance (which we will cover next week). As you're using R like a pro, it is so easy to create a confusion matrix.

   ❏ Pipe the *resultsLog* dataframe you created in **Q6** into the function `conf_mat(truth = ..., estimate = ...)`
   ❏ **Optional:** Pipe one more step into `autoplot(type = 'heatmap')` to color code. This is useful when more than two classes are involved. For now, this is just a learning point.

   Explain what the matrix tells you in addition to what you learned from the tables in **Q6**.

Ans - Confusion matrix also tells us about the false positives, false negatives, true positives and true negatives whereas the tables created above only show total positives and total negatives.

In the above table in Q6, we only got to know that the total predicted number of people having a heart disease was 19 and not having a heart disease was 19. We then compare this with the actual number of people having heart disease not having heart disease from the test set.

Here, the confusion matrix gives following values:

1. True positives - Count of people actually having heart disease and model predicted heart disease - 19
2. True negatives - Count of people actually not having heart disease and model predicted no heart disease - 919
3. False positives - Count of people actually not having heart disease and model predicted heart disease - 6
4. False negatives - Count of people actually having heart disease and model predicted no heart disease - 159

Our aim in this case should be to reduce false negatives as we would like to increase the accuracy of predicting people having heart disease.

8. No analysis is complete without a visualization. Plot the relationship between the statistically significant variables (*age*, *cigsPerDay*, *totChol*, *glucose*) and the probability of heart disease:

   ❏ Note that you stored the predicted probabilities as *predictedProb* in the *resultsLog* in Q6.

- ❏ Use `geom_point()` and `geom_smooth()` after `ggplot()`, without adding any parameters
- ❏ Be creative. For example, add `color=currentSmoker` (or `=gender`) into the `aes()`
- ❏ Add a title for the plots, and label both axes [**Hint:** You can use the labs() function]

Discuss your observations.

Ans -

1. Age vs Probability of Heart Disease, broken down by gender:

There is a clearly increasing slope, indicating that as the age increases, the probability of heart disease increases.

Further, the rate of increase is almost the same for both males and females but the overall probability values are higher for males as compared to females, as the smoothing line for males is clearly above the line for females.

2. Cigarettes Per Day vs Probability of Heart Disease, broken down by gender:

The rate of change of probability is almost constant with an increase in the number of cigarettes till a particular point, around 42 cigarettes per day and then starts increasing linearly.

Further, the probability of heart disease is more for males as compared to females smoking the same number of cigarettes per day. Also, for females there are no values higher than approximately 42, thus the smoothing line is almost flat. Interestingly, the probability drops a bit for females as the number of cigarettes increases from 0 to 10.

3. Total Cholesterol vs Probability of Heart Disease, broken down by hypertension:

The rate of change of probability increases with an increase in the total cholesterol level.

Further, as expected, the probability of heart disease is higher in people with hypertension (high BP) as compared to those without hypertension. The rate of change is approximately same for both groups.

4. Glucose vs Probability of Heart Disease:

Most people have glucose levels below 100 mg/dL and there is an increase in the probability of heart disease with increase in the glucose level. The rate of change tends to increase as the value becomes higher.

At the same glucose level, females have a higher probability of having a heart disease as compared to males.

There are not many males with very high glucose levels (above 270 approx.) whereas there are few females with very high glucose levels.

Switching to a new framework "Caret" we will continue to use in this course from now on:

9. You already loaded the "caret" library at the beginning. If not, load it now. Replicate the analysis in Question 6, this time using the caret library. Use Appendix II[3] for guidance.

   - ❏ Name the results table resultsLogCaret and create it using the train function.
   - ❏ Inspect resultsLogCaret carefully, compare it with resultsLog from Q6 and discuss.
   - ❏ Create the confusion matrix using caret, and compare it with the one in Q7. Discuss.
   - ❏ Don't worry about the rest of the output after the matrix. We will discuss it next week!

Ans - resultsLogCaret and resultsLog have the same values for probability and thus the confusion matrix also gives the same values. Basically, the model and thus output is the same in both.

10. Now that you have learned how to use logistic regression for classification, and how to do so **using the caret library**, you can solve another business problem for *Banco Portugal*. See Appendix III for the details of the dataset. The bank runs a telemarketing campaign for a savings account. Have you ever received one of those promotions by the way? "Open a savings account today and get XXX$ bonus!" See this month's promotions by clicking here.

    Banco Portugal hires you to predict whether a customer will open an account. The bank will use your model to develop promotional campaigns with higher conversion rates. Load the data, make conversions of variables as you see fit, and build logistic regression models using the caret library. Explore at least three alternative models[4], compare their performance, and pick a final model. Show your full work in the R Notebook. Below, discuss only your findings, your final decision, and explain how your final model helps Banco Portugal with its purpose.

    Now that we have discussed the performance measures, you can decide on a performance metric (or two) beyond just accuracy to compare the models and explain your reasoning. Because the caret library already reports the values of performance measures by default, you don't need to do any coding -This part is pretty much a thinking and reflecting exercise!

Ans -

About the model:

[3] If you made it to this point, ask me for the handout that includes Appendix II and III.

[4] These models can all be logistic regressions with a different set of independent variables.

After exploring the relationship between openedAccount and all other independent variables, factors that have a significant effect on the probability of a customer opening a bank account were found to be as follows:

1. Contact
2. Euribor3m
3. cons.conf.idx
4. cons.price.idx
5. emp.var.rate
6. poutcome
7. pdays
8. day_of_week
9. month
10. job

It doesn't make sense to use newcustomer - as this is highly correlated with the dependent variable itself and duration - as this data will only be available after the customer has been contacted.

All other variables were retained/removed based on the accuracy level of the model. (If adding a variable increased accuracy, it was retained and if not was removed.)

Comparing models based on sensitivity and specificity as well, the last model seems to be best with highest values.

Performance Measures:

This model has an accuracy of 89.11%, predicting 7839 true negatives and 307 true positives.

High specificity or predicting true negatives more accurately is more important here as it can lead to a lot of cost savings, by not wasting money on customers that are less likely to convert. Model does this best and thus has been selected as the final model. This also has overall highest accuracy.

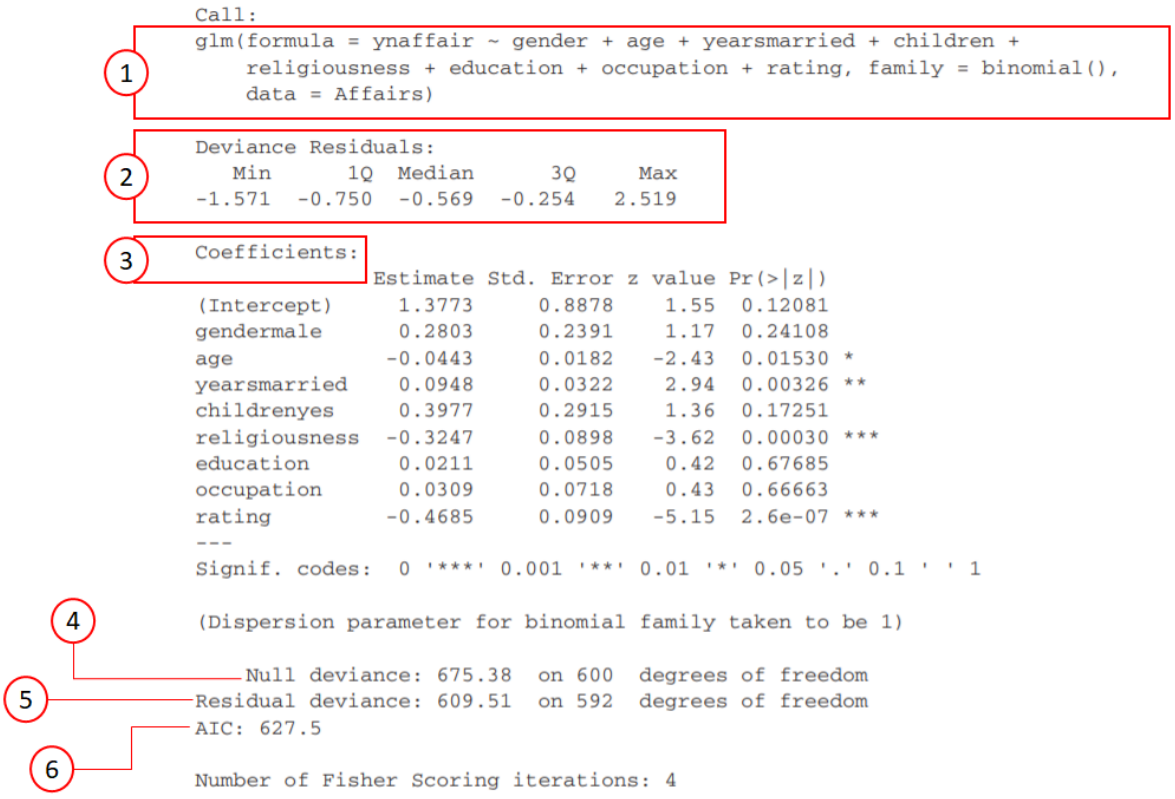
Model's value add to company:

This model will help Banco Portugal in prioritizing customer base according to these parameters and target those customers that meet the criteria according to the model, as this will help in driving maximum conversions. When running the target customer base data through the model, only those customers that have the probability of conversion can be contacted. The marketing budget can be effectively utilized by spending money in a more targeted way with high conversion rates.

Based on this model, data collection can also be more focussed as only parameters required by the model could be collected, if other information is not particularly useful elsewhere, saving time and effort.

Appendix I: How to run logistic regression in R and read the regression output

The output from summary() may seem overwhelming at first, so let's break it down one item at a time:



#	Item	Description
1	Formula	Like it was in the linear regression, the *glm()* formula describes the relationship between the dependent and independent variables. Note that you need to include *family = 'binomial'* as an argument.
2	Deviance Residuals	Because the difference between the observed and the fitted values are not very informative in a logistic regression, R reports the deviance residuals, which are the signed square roots of the ith observation to the overall deviance, calculated as follows: $$d_i = \text{sgn}(y_i - \hat{y}_i)\left\{2y_i \log\left(\frac{y_i}{\hat{y}_i}\right) + 2(n_i - y_i)\log\left(\frac{n_i - y_i}{n_i - \hat{y}_i}\right)\right\}^{(1/2)}$$

3	Coefficients	The regression coefficients show the change in log(odds) in the dependent variable for a unit change in the predictor variable, holding all other predictor variables constant. Because log(odds) are difficult to interpret, we usually exponentiate the coefficients and convert them into the odds scale: exp(the coefficient of yearsmarried) = exp(0.0948) = 1.10, which means a 1-year increase in the number of years married is associated with an increase in the odds of an affair by a factor of 1.10 (about a 10% increase), holding everything else constant. *What about a 10-year increase in the number of years married?* If you interpret a categorical variable like gendermale, exp(0.2803)=1.32 becomes the odds ratio. Therefore, the odds of a male having an affair are about 32% higher than the odds of a female doing so, holding everything else constant. You can exponentiate all coefficients by running `exp(coef(fit))`
4-5	Null Deviance, and Residual Deviance	The *null deviance* shows how well the dependent variable is explained by a model that includes only the intercept. The *residual deviance* shows how well the dependent variable is explained by a model that includes all the independent variables.
6	AIC	The Akaike Information Criterion (AIC) provides a method for assessing the quality of your model through comparison of related models. It's based on the Deviance measure, but includes a penalty for including additional independent variables. Much like adjusted R-squared, it intends to help you leave irrelevant predictors out. However, unlike adjusted R-squared, the reported number itself is not meaningful. When you compare nested models[5], you should select the model that has the smallest AIC. For BIC, run `BIC(fit)` after a regression, where *fit* is the model name, and R will report the BIC score. All of this also applies to BIC.
7	Fisher Scoring	This is just showing the number of iterations the model went through before it converged to this solution (not really useful).

[5] AIC can also be used in non-nested models, but using it requires caution. The data must be exactly the same.

Appendix II: Modeling using native way vs. the Caret way

Line by line comparison of making predictions using a logistic regression native way vs. caret way:

Note that the dependent variable is openedAccount in the example below:

```r
resultsLog <-
  glm(openedAccount ~ ., family='binomial', data=dfTrain) %>%
  predict(dfTest, type='response') %>%
  bind_cols(dfTest, predictedProb=.) %>%
  mutate(predictedClass = as.factor(ifelse(predictedProb > 0.5, 1, 0)))

resultsLog %>%
  conf_mat(truth=openedAccount, estimate=predictedClass)
```

```r
resultsLogCaret <-
  train(openedAccount ~ ., family='binomial', data=dfTrain, method='glm') %>%
  predict(dfTest, type='raw') %>%
  bind_cols(dfTest, predictedClass=.)

resultsLogCaret %>%
  xtabs(~predictedClass+openedAccount, .) %>%
  confusionMatrix(positive = '1')
```

Line 14 vs. Line 3: Use train() function instead of glm() and define the method in the method argument.
Line 15 vs. Line 4: Use predict() with type='raw' to get the predicted classes instead of probabilities.
Line 16 vs. Line 5: Name the column as predictedClass instead of predictedProb for this reason.
N/A vs. Line 6: No need to use a mutate() function to convert probabilities into classes.

Line 19 vs. N/A: Use the xtabs() function only because confusionMatrix() needs one.
Line 20 vs. Line 9: Use confusionMatrix() rather than conf_mat() and define the positive class.

Appendix III: Details of the Banco Portugal savings account dataset

Relevant Information:
The bank's customer-level data is extended by the addition of five social and economic features/predictors (at the end of data dictionary, national-wide indicators from Portugal), published by the Banco de Portugal and publicly available at bportugal.pt/estatisticasweb

Source:
Sérgio Moro (ISCTE-IUL), Paulo Cortez (Univ. Minho) and Paulo Rita (ISCTE-IUL) @ 2014

Past Usage:
The full dataset was described and analyzed in:
S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems (2014), doi:10.1016/j.dss.2014.03.001.

Objective:
The classification goal is to predict if a customer will open a savings account (*accountOpened*).

Data Summary:
Number of observations: 41188 Number of variables: 20+

Data Dictionary:
For more information, you can refer to Moro et al. (2014) cited above.

Variable	Data type	Description
openedAccount	categorical	Has the customer opened a savings account? ("yes","no")
newcustomer	categorical	If the customer is a new customer or not (yes = 1, no=0)
age	numeric	Age of the customer
agegroup	categorical	The age group that the customer belongs to ("Teenagers", "Young Adults", "Adults", "Senior Citizens")
job	categorical	Type of job ("admin", "blue-collar", "entrepreneur", "housemaid", "management", "retired", "self-employed", "services", "student", "technician", "unemployed", unknown)
marital	categorical	Marital status ("divorced", "married", "single", "unknown"; note: "divorced" means divorced or widowed)
education	categorical	Educational qualification ("basic.4y", "basic.6y", "basic.9y", "high.school", "illiterate", "professional.course",

		"university.degree", "unknown")
default	categorical	Has credit in default? ("no", "yes", "unknown")
housing	categorical	Has a housing loan? ("no", "yes", "unknown")
loan	categorical	Has a personal loan? ("no", "yes", "unknown")
contact	categorical	Contact communication type ("cellular", "telephone")
month	categorical	Last contact month of year ("jan", "feb", ..., "nov", "dec")
day_of_week	categorical	Last contact day of the week ("mon","tue","wed","thu","fri")
duration	numeric	Last contact duration, in seconds **Important**: This attribute highly affects the outcome (e.g., if duration=0 then y="no"). Yet, the duration is not known before a call is performed. Also, after the end of the call the outcome is obviously known. So, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
campaign	numeric	Number of contacts performed during this campaign and for this client (includes the last contact)
pdays	numeric	Number of days passed by after the client was last contacted from a previous campaign **("999" means client was not previously contacted)**
previous	numeric	Number of contacts performed before this campaign
poutcome	categorical	Outcome of the previous marketing campaign ("failure", "nonexistent" ,"success")
emp.var.rate	numeric	Employment variation rate - quarterly indicator
cons.price.idx	numeric	Consumer price index - monthly indicator
cons.conf.idx	numeric	Consumer confidence index - monthly indicator
euribor3m	numeric	Euribor 3 month rate - daily indicator
nr.employed	numeric	Number of total employment - quarterly indicator