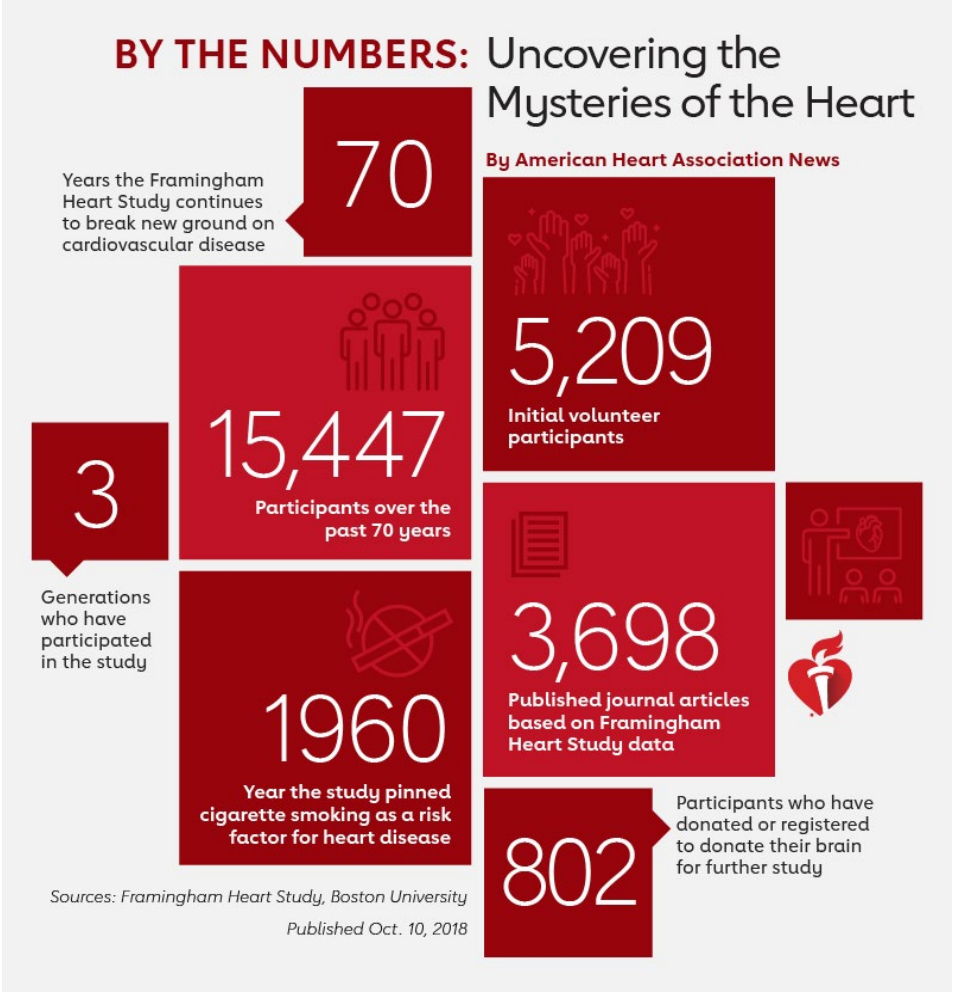
Lab 3

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!]

   Boxplot of sysBP:
   Based on the boxplot we can say that possibility of the systolic blood pressure being higher is more in people suffering from CHD. Thus, we can say that the risk of suffering from CHD is higher if a person has higher systolic blood pressure. This is also true as per medical theories.

   Boxplot of diaBP:
   We observe the similar pattern as systolic blood pressure for diastolic blood pressure.

   Boxplot of total cholesterol:
   Based on the boxplot we can say that people suffering from CHD have a possibility of having higher cholesterol levels. Thus, having high cholesterol could be associated with suffering from CHD.

   Thus, we can say that blood pressure levels and cholesterol are associated with heart disease.

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?

   The ratio of males to females in the training set is 44.59:55.408 while the ratio in the test set is 43.84:56.15 which is quite similar proportions. This means that the sampling bias is negligible between the test and train sets. Also, though the split is not exactly 50% in both the datasets its pretty close to a 50% split. Which means that data is not highly biased towards a particular gender. A 50% split would have been ideal and would have helped make stronger conclusions.

In terms of age, the distribution is not uniform in the dataset, it follows a near normal distribution. But the distribution of age groups in the test and the training data sets are quite similar. This means that there is small or negligible sampling bias introduced into the data.

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!

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.

   Based on the collinearity diagnostic test using VIF Values we see that cigsperday and currentsmoker are correlated. Similarly, sysbp and diabp are correlated. We can take cigsperday and drop current smoker to reduce multicollinearity as they are related to each other. We do not have to delete one of the sysbp and diabp as they are independently related to CHD in medical terms.

   The variables that are significant at 95% are those variables which have a p-value < 0.05.

   These variables are,

   Gender, age, cigsperday, prevalentstroke, prevalenthyp, sysbp, heartrate and glucose.

[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

This model says that the risk factors to heart disease are generally smoking cigarettes, having a precondition like hypertension or stroke, irregularities in heartrate and glucose. Gender and age could also be risk factors. But in medical terms, being diabetic or having high BMI, having high cholesterol could also be potential risk factors to heart disease. This model fails to capture that relation.

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.

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.

On inspection of the test dataset we find that 172 out of 1097 people have heart disease. But out model was able to predict only 10 people as having heart disease. On direct comparison we can say that the model is not a good fit as it is not able to predict accurately. Also, if we look closely, we still must verify whether these cases predicted to have heart disease by our model are actually people with heart disease. But, irrespective of that based on our initial results we can say that the model is not a good fit to predict the heart disease occurrence.

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.

[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.

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

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.

   The variables that are statistically significant at 95% are gender, age, cigsperday, prevalenthyp, totchol, sysbp, heartrate, glucose.

   While it covers most of the statistically significant variables from Q3(apart from prevalentstroke) it also includes totchol (total cholesterol) as a significant variable.

   This should improve the model as the variable totchol should be associated with heart disease as per the medical terms.

   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 1-year increase in age is associated with an increase in the odds of having CHD by a factor of 1.069 (about a 6.9% increase), holding everything else constant.

     Gender:

     The odds of a male having CHD is about 52.53% higher than the odds of a female having CHD, holding everything else constant.

     Diabetes:

     The odds of a diabetic person having CHD is 0.05% lower than the odds of a non-diabetic person, 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?

   The logistic model predicts that 19 people will have heart disease. This is a bit higher than the value predicted by linear regression model. On comparison with the actual number of people suffering from heart disease as per the test dataset (172) this predicted number if much smaller. We can also not be sure based on this output that whether these predictions are true positives or false positives. But still, based on our initial observation we can say that the model is still not the best fit though it is an improvement from the linear model. This improvement is because the logistic model output consists of probabilities that fall within the range of 0 and 1 unlike linear model and thus performs better. Also the logistic model performs better in case of categorical DV like tenyearchd.

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

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**.

   From the tables in Q6 we learnt that the model predicted 19 people to have heart disease while 1078 people were predicted to not have the disease. The matrix provides us more information on this, it provides a breakdown on how many of these predictions are correct or incorrect (false positives and false negatives).

   From the matrix we can say that, 13 people were correctly predicted to have heart disease while 919 were correctly predicted to not have heart disease. It also tells us that 159 people who had heart disease were incorrectly predicted to not have it while 6 people who did not have the disease were predicted to have it.

   Thus, though it may look like the model predicted 19 cases as having CHD from the results in the matrix we know that the model is predicting only 13 of them correctly.

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.

   Plot 1: Predicted Probability of CHD vs Age (by Gender)

   We see that the probability of CHD is higher is males than is females in any age category. The difference in probability decreases as the age increases beyond 65 years. This is inline with the medical theories.

   Plot 2: Predicted Probability of CHD vs Ciggarettes per day (by education)

   We see an increasing trend in the probability of CHD as the number of cigarettes per day increases. On splitting the data based on education we can observe that more educated the person, less the number of cigarettes smoked. Also, the probability of CHD increases at a lower rate for more educated people. This could be due to more exposure and awareness to heart health in educated people.

   Plot 3: Predicted Probability of CHD vs Total Cholesterol (by prevalent hypertension)

   We see that higher cholesterol is associated with higher probability of CHD. At the same time the cholesterol levels in people with prevalent hypertension are higher which makes them more prone to CHD than people who do not have hypertension. This is inline with the medical theories.

   Plot 4: Predicted Probability of CHD vs Glucose Level (by BpMeds)

   We see a steep increase in the probability of CHD as the glucose levels increases. We see that the probability is a lot higher in males than females in case of increasing glucose levels. This is inline with the medical theories.

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!

   resultsLogCaret and resultsLog have the same values and outputs as they are both logistic models and working on the same dataset. The difference is that the predictedClass variable is already vectorized in case of resultsLogCaret unlike resultsLog where we have to add an additional step for it. This is a feature provided by the Caret library we are using.

   The confusion matrix created using caret not only shares the matrix but it also provides the details on accuracy, confidence interval, sensitivity and precision among other values. This output is much more detailed and enables deeper inference and conclusion than the output in Q7. This is again an additional feature of the caret library.

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

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!

    Model 1:

    Included all the variables except,

    Default: It has only one value "yes" rest is all "no" in the training set. This will introduce bias

    NewCustomer: As the poutcome = success covers the information provided by new customer variable

    Duration: As mentioned in the dataset description, the variable is highly volatile, does not explain anything new and should not be used in a predictive model

    Model 2:

    After checking for multicollinearity removed the following variables.

    Month – As the information is better explained by pdays

    Agegroup – As it is derived from age

    Model 3:

    Removed the statistically insignificant variables education, marital, euribor3m

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

Comparison of the models:

Model	*Accuracy*	*Sensitivity*	*Positive Prediction*	*Negative Prediction*
Model 1	*0.8898*	*0.2574*	*0.689*	*0.899*
Model 2	*0.8892*	*0.2497*	*0.687*	*0.899*
Model 3	*0.8898*	*0.2506*	*0.696*	*0.899*

Accuracy cannot be a trusted metric as the dataset itself is biased with ~87% of the customers not opening an account.

Sensitivity shows out of total actual customers who opened the account what proportion did we predict correctly to be opening an account. All three models show almost similar sensitivity.

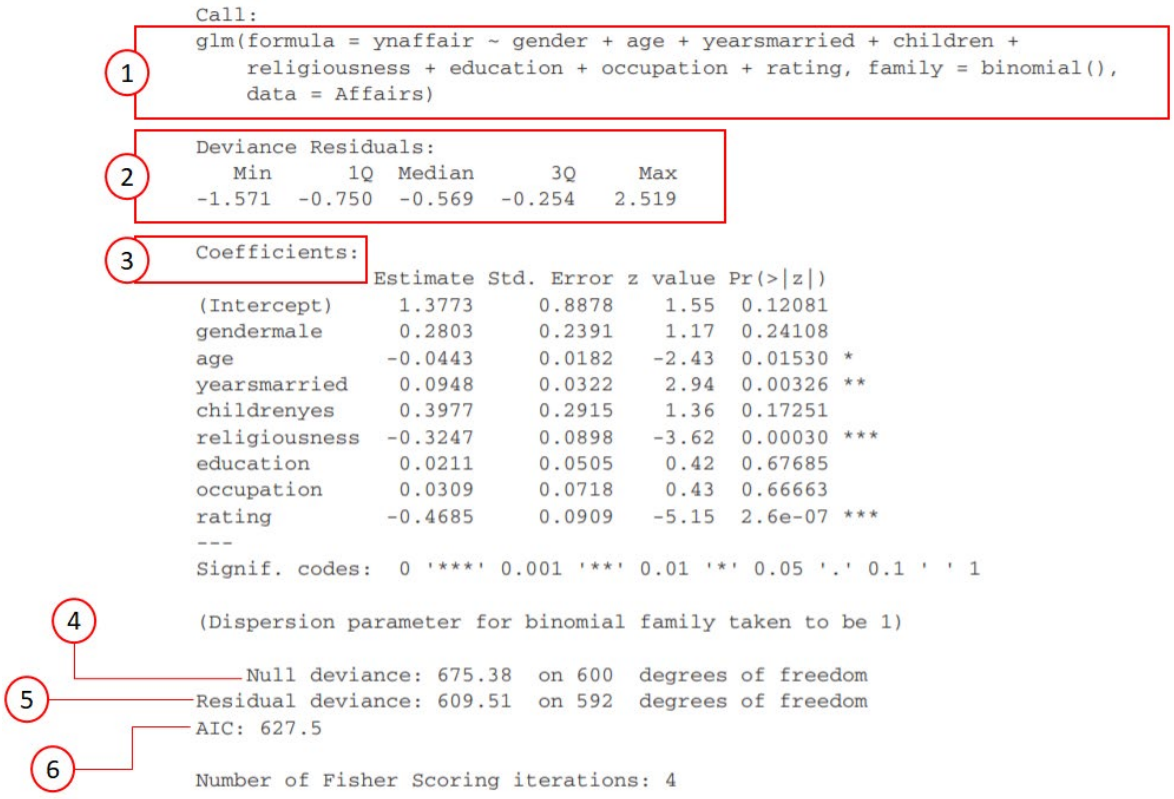
Negative Prediction shows out of the total customers predicted to not open an account what proportion actually did not open an account. All three models have same performance in terms of negative prediction.

Positive Prediction shows out of the total customers predicted to open an account what proportion actually did open an account. This value is the highest (slightly) for Model 3.

On using any of these models the bank would spend money on the customers who are predicted to open an account while targeting them. They want the conversion rate of these customers to be the highest. This makes Positive Prediction an important metric for this model for measuring performance. Thus, the bank could choose model 3 for its use.

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:

```
1  ```{r}
2  resultsLog <-
3    glm(openedAccount ~ ., family='binomial', data=dfTrain) %>%
4    predict(dfTest, type='response') %>%
5    bind_cols(dfTest, predictedProb=.) %>%
6    mutate(predictedClass = as.factor(ifelse(predictedProb > 0.5, 1, 0)))
7
8  resultsLog %>%
9    conf_mat(truth=openedAccount, estimate=predictedClass)
10 ```
11
12 ```{r}
13 resultsLogCaret <-
14   train(openedAccount ~ ., family='binomial', data=dfTrain, method='glm') %>%
15   predict(dfTest, type='raw') %>%
16   bind_cols(dfTest, predictedClass=.)
17
18 resultsLogCaret %>%
19   xtabs(~predictedClass+openedAccount, .) %>%
20   confusionMatrix(positive = '1')
21 ```
```

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