Exercise 17 – Logistic regression

Zoltan Kekecs

15 May 2021

Table of content

Exercise 17 - Logistic regression

In this exercise we will learn how to make predictive models on binomial outcome variables. We will mainly discuss using logistic regression.

Data management and descriptive statistics

Our dataset

We will use the Heart Disease dataset, a well-known dataset used to demonstrate classifications problems. The dataset contains different information about patients who were screened for the presence of heart disease. The following variables are included in the dataset:

- age - age in years
- sex - (1 = male; 0 = female)
- cp - chest pain type (0 = asymptomatic, 1 = typical angina, 2 = atypical angina, 3 = non-anginal pain)
- trestbps - resting systolic blood pressure (in mm Hg on admission to the hospital)
- chol - serum cholestoral in mg/dl
- fbs - (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
- restecg - resting electrocardiographic results: (0 = normal; 1 = having ST-T wave abnormality; 2 = showing probable or definite left ventricular hypertrophy)
- thalach - maximum heart rate achieved during the exercise test
- exang - exercise induced angina (1 = yes; 0 = no)
- oldpeak - ST depression induced by exercise relative to rest
- slope - the slope of the peak exercise ST segment
- ca - number of major vessels (0-3) colored by flourosopy
- thal - 3 = normal; 6 = fixed defect; 7 = reversable defect
- disease_status - have disease or not (heart_disease vs. no_heart_disease)

The .sav file can be downloaded from:

https://github.com/kekecsz/SIMM32/blob/master/2021/Lab_3/Heart%20disease.sav

See more information about the dataset here: https://www.kaggle.com/ronitf/heart-disease-uci; https://archive.ics.uci.edu/ml/datasets/Heart+Disease

Exploratory data analysis

You should always check the dataset for coding errors or data that does not make sense, and explore the dataset to get a basic feel of what type of data you are dealing with.

View data in the data editor and display simple descriptive statistics and plots. You can find the commands for data exploration in the **Analyze > Descriptive Statistics tab**, such as:

Analyze > Descriptive Statistics tab > Frequencies

> Frequencies tables will allow you to inspect what kind of values are there in each variable. These values should inform you about whether the data take realistic values.

Analyze > Descriptive Statistics tab > Descriptives

> Descriptives can give you information about the mean and SD, minimum and maximum values, and the skewness and kurtosis of the distribution of the variables.

Analyze > Descriptive Statistics tab > Explore

> Explore gives you similar information, but it also includes confidence intervals around the mean, and gives you the option to display a histogram to visually inspect the distribution of the data.

Research question

As we have seen previously, the main purpose of regression models is to do prediction. In this study we would like to predict from the test results of the person whether they have heart disease or not. So the outcome variable in our models will be heart disease status.

Recoding variables

One issue that we have to face is that the outcome variable is a string, and not a numerical variable. We can solve this by recoding the outcome variable so that it becomes a numerical variable.

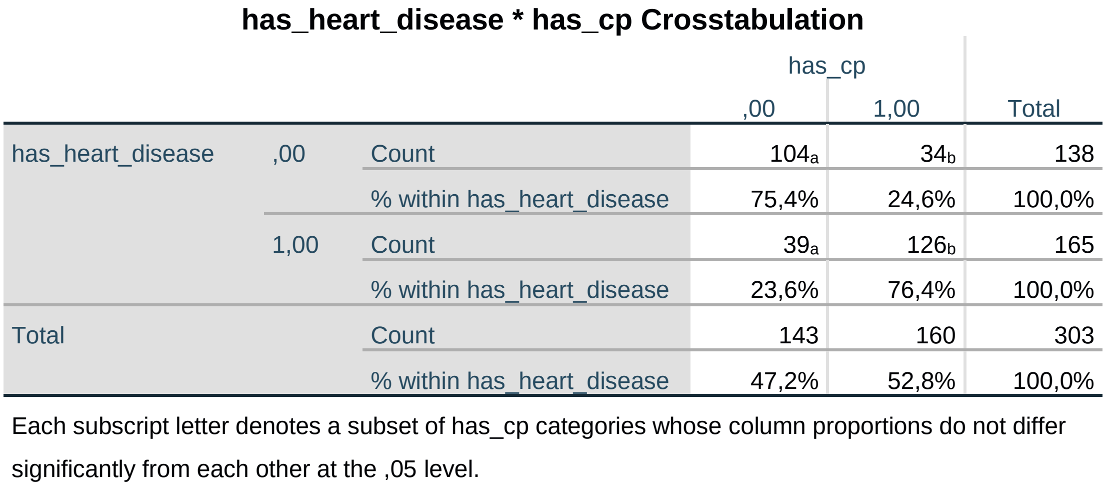
Using the **Transform>Recode into different variables…** function we can do this. Here we can designate the new variable name by entering anything into the **Name** text box (in this example I named the new variable “has_heart_disease”), and clincking “change”. Then we also need to click “Old and new variables” to be abel to specify what old values to recode to what new values. We need to specify these one-by-one for each value we want to change. For example we will change the value “heart_disease” to 1and 'no_heart_disease' to 0. After specifying each, you need to press “Add”, and it will appear in the Old → New list.

For the exercise we will also need to recode cp (chest pain) into a new variable. cp represents different types of chest pain the person has (1-3), or if the person does not experience chest pain. We will create a new variable with Recode into different variable. The new variable will be called has_cp, and it will represent if the person experiences chest pain (any type)(coded as 1) or not (coded as 0).

```
*Recode string into binary numeric variable


RECODE disease_status ('heart_disease'=1) ('no_heart_disease'=0) INTO
has_heart_disease.

EXECUTE.


*Recode cp into has_cp


RECODE cp (0=0) (MISSING=SYSMIS) (ELSE=1) INTO has_cp.

EXECUTE.
```

We usually start with exploratory data analysis. We look at the relationship between the variables in our model using tables and graphs.

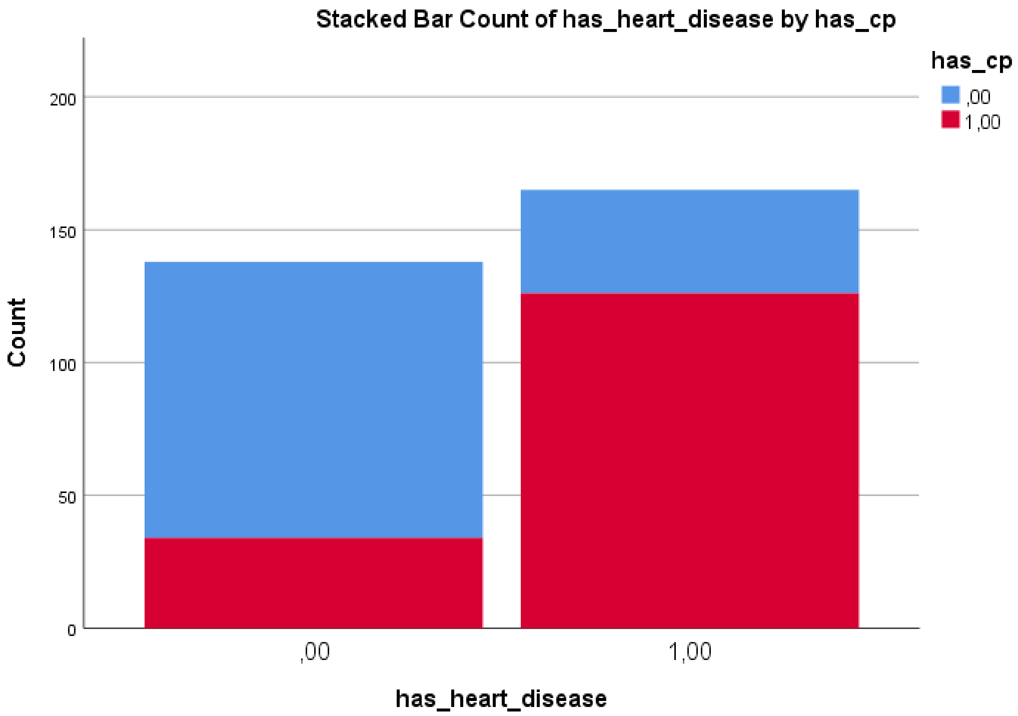
We can explore the relationship of two categorical variables using **Analyse > Descriptives > Crosstabs** (we can ask for percentages in the **Cells…** menu), and using the **Chart builder** by building a **stacked bar chart** with one categorical variable on the x axis and the other as the color of the bars.



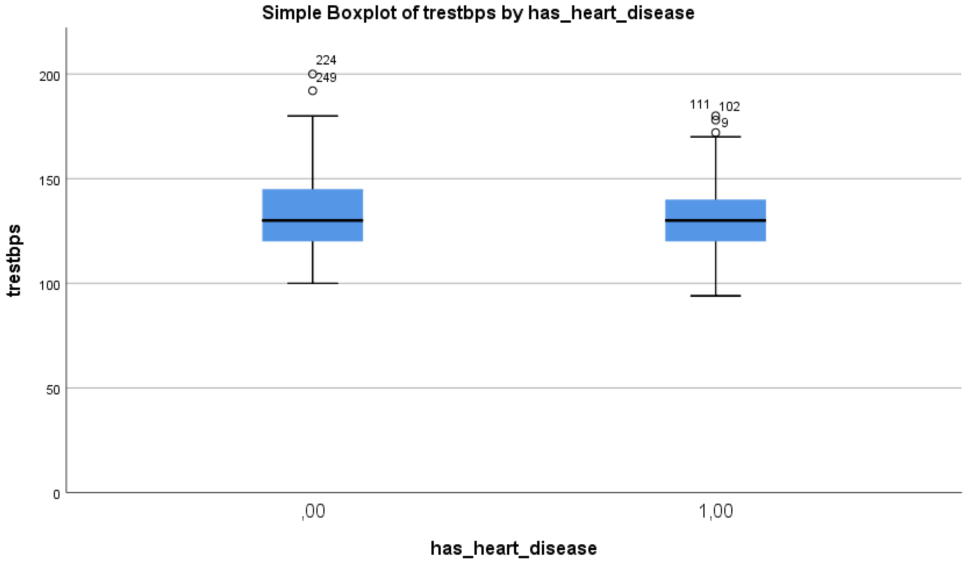
has_heart_disease * has_cp Crosstabulation

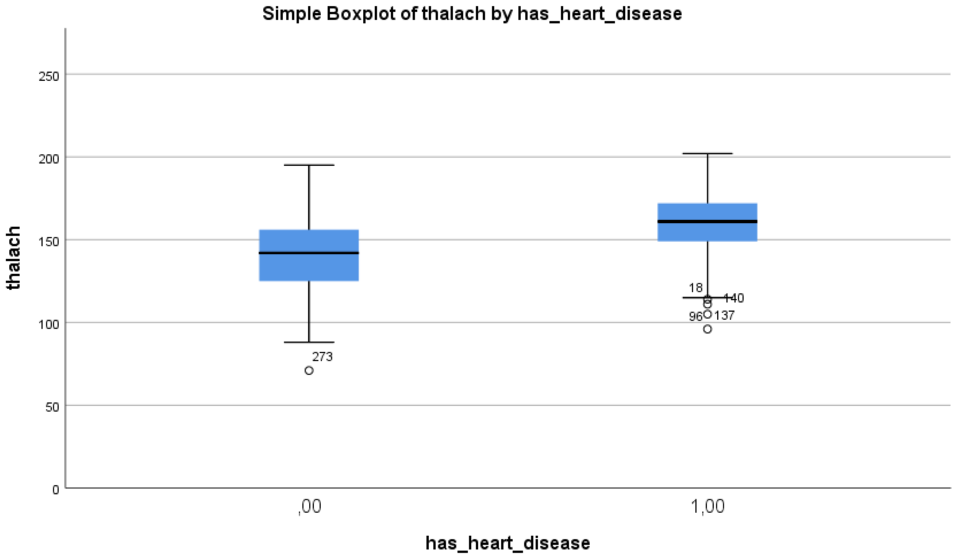
			has_cp		
			,00	1,00	Total
has_heart_disease	,00	Count	104_a	34_b	138
		% within has_heart_disease	75,4%	24,6%	100,0%
	1,00	Count	39_a	126_b	165
		% within has_heart_disease	23,6%	76,4%	100,0%
Total		Count	143	160	303
		% within has_heart_disease	47,2%	52,8%	100,0%

Each subscript letter denotes a subset of has_cp categories whose column proportions do not differ significantly from each other at the ,05 level.



You can use the **Analyse > Descriptives > Explore** procedure to explore the distribution of continuous variables by groups, by specifying the grouping variable as a Factor. We can also use the **Chart builder** to explore this relationship by building a **boxplot** with the continuous variable on the Y axis and the categorical variable on the X axis.





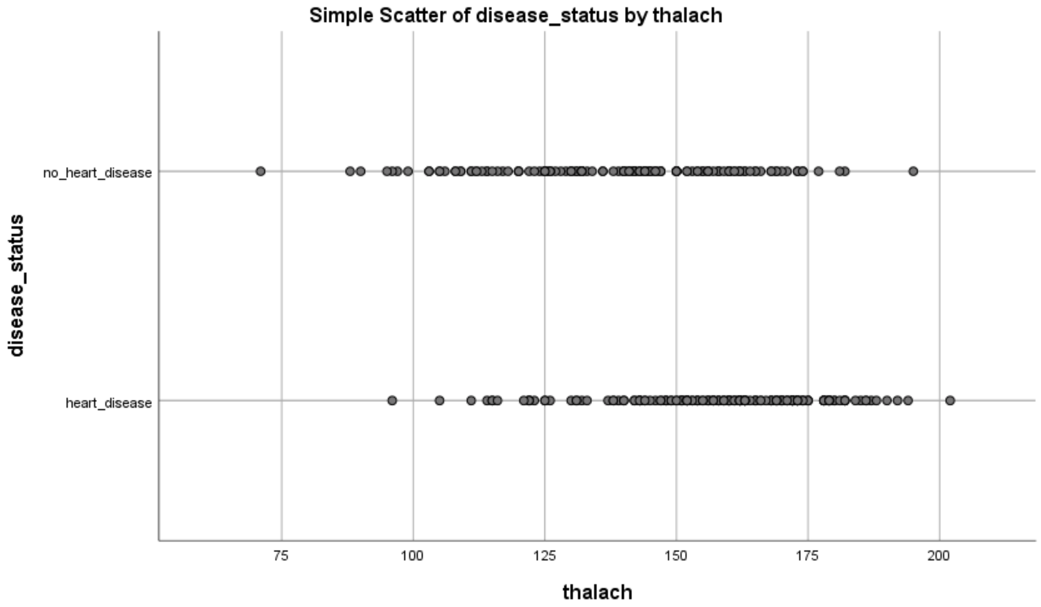
This exploratory analysis tells us that chest pain and higher post-exercise heart rate (thalach) could potentially be risk factors for heart disease. The relationship is not that clear from the graphs and descriptives when it comes to heart disease and resting blood pressure.

Binary logistic regression

Fitting regular regression (not recommended)

Based on the previous exercises you might think that fitting a regression model on the data would be a good solution here, so let's start with looking at what would be the result of a regular linear regression with this data.

Let's see the relationship of a potential predictor and the predicted outcome. Here we plot the scatterplot of disease status and thalach (the maximum heart rate reached during exercise).



It is immediately apparent that a single linear regression line that we could fit on this data would not fit well.

If we fit a regular regression model with the just created has_heart_disease variable as an outcome and thalach as the independent predictor variable, we get a regression coefficient of 0.009 for thalach. This means that for every 1 points of heart rate increase, there is a 0.009 point increase in the outcome variable.

Coefficients[a]

				Standardized		
		Unstandardized Coefficients		Coefficients		
Model		B	Std. Error	Beta	t	Sig.
1	(Constant)	-,830	,172		-4,817	,000
	thalach	,009	,001	,422	8,070	,000

a. Dependent Variable: has_heart_disease

If we now compute the expected outcome value for a person with 120 as the highest heart rate achieved. We would get -0.83 + 0.009*120 = 0.25. This prediction does not really make sense when the outcome can only be 0 or 1.

We might think of this predicted value as the probability of having a value of 1 instead of 0 on the outcome variable. But we can also see that the model could easily return negative numbers as predictions as well, while probabilities can only take a value of 0 or 1. For example a very sporty person might only produce 90 as the highest heart rate on the exercise test that was used in this study. The predicted “probability” for this person having a heart disease would be -0.02, but there is no such thing as a negative probability for an event. So we need another approach that would return realistic predictions.

Logistic regression basic idea

The solution is to use Generalized Linear Models (GLMs) for prediction instead of regular Linear Models. GLMs are designed to model outcome variables that are not normally distributed. One member of this family of GLMs is Logistic regression, which is specifically designed to model binary outcomes (categorical outcome variables with only two levels).

The basic idea in GLMs is to use a link function that transforms the predicted outcome variable to a scale that would put the results of the regression on a realistic scale. The link function used by logistic regression is the logit function (which is the natural logarithm of the odds of the predicted event). So in logistic regression instead of predicting the actual outcome value (1 or 0), we predict the log-odds of the outcome value. This now can be treated in the regular linear regression framework, since ln(odds) can take any value between negative infinity to infinity. So after we computed the regression equation for the ln(odds) for any case, we can derive the odds or the probability of the outcome event from the regression equation as well.

Running the analysis in SPSS

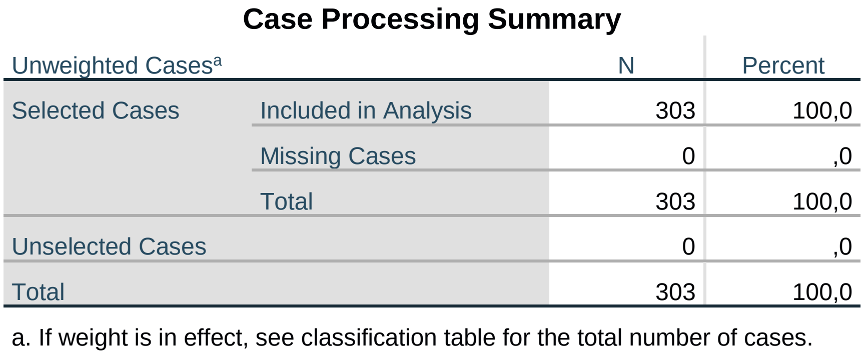
Now let's try this in action. Let's say that we have good reason to believe that having chest pain (has_cp), resting systolic blood pressure (trestbps) and maximum heart rate achieved during the exercise test (thalach) should predict if someone has heart disease (has_heart_disease).

We can build a logistic regression model in the **Analyse > Regression > Binary logistic** menu. There we can specify the dependent and the independent variables just like we did when building a regular linear regression model. In the **Options** menu you should also ask for the CI for Exp(B).

```
*Logistic regression analysis


LOGISTIC REGRESSION VARIABLES has_heart_disease
  /METHOD=ENTER has_cp trestbps thalach
  /PRINT=CI(95)
  /CRITERIA=PIN(0.05) POUT(0.10) ITERATE(20) CUT(0.5).
```

Interpreting the output

We get the following output from the analysis above:

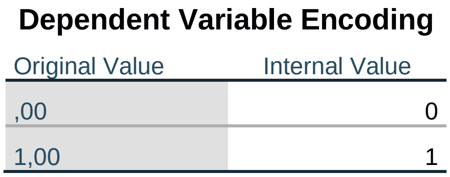


Case Processing Summary

Unweighted Cases[a]		N	Percent
Selected Cases	Included in Analysis	303	100,0
	Missing Cases	0	,0
	Total	303	100,0
Unselected Cases		0	,0
Total		303	100,0

a. If weight is in effect, see classification table for the total number of cases.

The case processing summary gives you basic information about which cases in the dataset were used in the analysis. In our example we used all cases in the dataset, nothing was excluded and there were no missing values.



Dependent Variable Encoding

Original Value	Internal Value
,00	0
1,00	1

If you choose logistic regression, SPSS transforms our outcome variable to values with 0 or 1 even if it was a string variable. The Dependent Variable Encoding gives you information about which original values is represented by 0 and which is represented by 1 for the purpose of our analysis. In our example we already used a variable with values of 0 and 1, so nothing was recoded, but you could try using the string variable as a dependent here and see which value will be recoded into which number automatically.

Block 0: Beginning Block

SPSS conducts a hierarchical regression analysis here. In “Block 0” we check the predictive efficiency of the “null model”. Remember that in a regular linear regression the null model uses the mean for predicting the outcome value. In our case the mean of the outcome is 0.54. But a value of 0.54 makes no sense as a predicted value, since the value of the outcome can only take the value of 0 or 1. In a logistic regression we use the “most likely value” as a predicted value for all cases in the null model. So if more than 50% of the cases have a value of 1, the null model will use 1 as a prediction for *every* case. Conversely, if the probability is lower than 50%, the null model will use 0 as a prediction for *every* case.

Classification Table[a,b]

			Predicted		
			has_heart_disease		Percentage
	Observed		,00	1,00	Correct
Step 0	has_heart_disease	,00	0	138	,0
		1,00	0	165	100,0
	Overall Percentage				54,5

a. Constant is included in the model.

b. The cut value is ,500

This also makes the prediction performance of the null model immediately apparent from the classification table: if 54.5% of the cases had a value of 1 to begin with, and if we use 1 as a predicted value for everyone, we will naturally have a 54.5% correct prediction rate. Our goal with our model containing the predictors is to improve our prediction efficiency from using the most likely value as a prediction (the null model).

Interpreting model coefficients

Now let’s look at the output of the null model (Block 0) to better understand it:

The null model only has one element, the intercept (in other words the constant), and we only use this to make a prediction.

In a regular linear regression, the coefficient for the intercept would be interpreted as the expected value of the outcome if all of the predictors have the value of 0 (zero), and in the null model this would fall on the mean of the outcome variable. The B value (regression coefficient) of the intercept in binary logistic regression can be interpreted the same way. However, this does not represent the expected value of the outcome, instead, it represents the log(Odds) of the outcome. This log(Odds) is hard to interpret in its raw form, so we usually convert this to Odds, or probabilities. For this to make sense, we need to understand the meaning of Odds and probabilities and how to convert one into the other.

The odds of some event reflect the likelihood that the event will take place. The odds is usually represented as a pair of numbers. If the event/outcome of interest is surviving an airplane crash, and if the odds for survival are 1 to 4, this means that on average for every 1 person surviving, there are 4 people who do not survive. This can be also written up as the odds being 0.25 since 1/4 = 0.25. If the odds for surviving the plane crash are 2 to 1, this means that on average for every 2 survivors there is 1 person who dies. The odds here can also be represented as 2, since 2/1 = 2. We can compute the odds from ln(Odds) by using the exp() function: Odds = exp(ln(Odds)). It is important to realize that the value of odds **depends on our perspective**, on what is our event of interest. In this example our event of interest is surviving, and if we switch the event of interest to not surviving (dying) in the plane crash (the other possible outcome) the odds would be the inverse of the odds of survival. So if the odds of survival is 0.25, the odds of dying is 1/0.25 = 4. If the odds of survival is 2, the odds of dying would be 1/2 = 0.5.

Odds ratio: Another important concept related to odds is that of odds ratio. This is an effect size measure, which can be used to assess the effect of being a member of a certain group vs. not being the member of that group on the odds of the event of interest. The **odds ratio** is the ratio of the odds of the event of interest between two groups. It is computed by dividing the odds of the event of interest in the group by the odds of the event in outside of this group. It basically represents how much higher or lower the odds (risk) for an event is in one group over another group. The easiest example is if we compare the odds of an event in two groups, so let's compute the **odds ratio** related to biological sex for dying in a plane accident. Let's imagine that for every 1 male who survives a plane crash there are 6 who die in the plane crash on average, making the odds of death in a plane accident 6 for males. Let's suppose that for some reason, women are less at risk of dying in a plane crash: for every 1 female who survives the crash there are 2 females who die, so the odds of death in a plane crash for females is 2. Now we can calculate the odds ration by dividing the two odds: 6/2 = 3. So the odds ratio of males dying in a plane accident over females is 3, because the odds for males to die in a plane crash is 3 times as that of females.

Just like with odds, the odds ratio is also dependent on our perspective: on what is the group of interest. So if the odds ratio of males (vs. females) dying in a plane crash is 3, the odds ratio of females (vs. males) for dying in a plane crash is 1/3 = 0.3333. The odds ratio of 0.33 meaning that the odds of die in a plane accident is 0.33 times as large of females compared to males. In other words, the odds of dying is 33% among females compared to males.

Probability is the fraction of times you expect to see the event of interest in many trials. If the probability is 1/4 we expect to see that event 1 out of 4 times on average. So if the probability of surviving a plane crash is 1/4 = 0.25 = 25%, we expect that out of 4 people 1 will survive and 3 will die. On the other hand if the probability is 3/4 = 0.75 = 75%, we would expect that on average 3 will survive and 1 will die out of 4 people. We can compute the probability from log(Odds) by the following formula: p = exp(log(Odds))/ (1 + exp(log(Odds))), in other words Odds / (1 + Odds).

Now we can convert the intercept's coefficient that we got in our null model (B = 0.179) to Odds or probabilities.

Odds = Exp(0.179) = 1.196. (This is displayed in the variables in the Equation table in the Exp(B) column in SPSS.)
In other words: for every 1.196 people who has heart disease, there is 1 person who does not have heart disease in our target population.

p = Exp(0.179)/ (1 + Exp(0.179)) = 0.5446

In other words: for every 100 people in our target population there are 54 who has heart disease. This is of course exactly the same probability as we observed in our dataset, since in this model we only use this observed probability for prediction as noted above.

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 0	Constant	,179	,115	2,400	1	,121	1,196

The variables not in the equation table gives us a list of which variables are not yet included in this null model that will be included in later blocks, and also about whether adding any of these predictors individually to the model would improve predictive power significantly or not.

Variables not in the Equation

			Score	df	Sig.
Step 0	Variables	has_cp	80,680	1	,000
		trestbps	6,365	1	,012
		thalach	53,893	1	,000
	Overall Statistics		103,583	3	,000

Now that we understand odds and probabilities, and how we can calculate them from the regression coefficient, we can use the same knowledge when interpreting the coefficients listed in the Variables in the Equation table in Block 1.

Variables in the Equation

								95% C.I.for EXP(B)	
		B	S.E.	Wald	df	Sig.	Exp(B)	Lower	Upper
Step 1[a]	has_cp	2,014	,291	48,019	1	,000	7,494	4,240	13,247
	trestbps	-,022	,008	6,776	1	,009	,978	,962	,995
	thalach	,033	,007	22,125	1	,000	1,034	1,020	1,048
	Constant	-2,923	1,480	3,902	1	,048	,054		

a. Variable(s) entered on step 1: has_cp, trestbps, thalach.

Here we can see the model coefficients in the final model, including the intercept, and the predictors:

The coefficient for has_cp is 2.014. We can already see from the sign of this number (positive) that holding everything else constant, having chest pain (has_cp = 1) in the model is associated with increased risk of heart disease. However, because of the log(Odds) scale, it is hard to interpret the size of the effect, so let’s move on to the interpreting the Exp(B) column.

The coefficient is converted to the odds scale in the Exp(B) column: 7.494. The Exp(B) of the predictors is interpreted as **odds ratio** (rather than raw odds). In order to interpret this odds ratio, we need to remember that in the has_cp variable 0 means that the person did not report chest pain, and 1 means that the person reported chest pain. The **Exp(B) represents to odds ratio of being the member of the group where the variable takes the value of 1** (having chest pain) **vs. 0** (not having chest pain). This means that if every other variable is unchanged, being a member of the chest pain group is associated with 7.49 times higher odds for having hear disease compared to being the member of the no-chest-pain group.

We can interpret the coefficients of the continuous predictors in a similar way. One helpful thing is to imagine that we are comparing two groups again. For example the odds ratio of trestbps is exp(B) = 0.962. This can be interpreted as follows: being the member of the group that has 1 point higher value on trestbps is associated with 0.962 times the odds for having heart disease compared to people with 1 point lower values on this variable (if every other predictor’s value is held constant). In other words, higher values of trestbps are associated with lower risk for heart disease, specifically, 1 point increase in trestbps decreases the odds to 96.2% that of the odds if the value of trestbps was unchanged.

We can also use the regression coefficient to make predictions about the likelihood of the outcome or event of interest being true for any individual case in the same way as we use the regression equation to predict outcomes in linear regression. For example, let’s say that we would like to use this statistical model to predict the likelihood of a new patient in the hospital having heart disease based on his or her test results.

Importantly, we need to use the regression coefficient, log(odds) in the regression equation (instead of the odds ratio). The regression equation for this model is: -2,923 + 2,014*has_cp - 0,022*trestbps + 0.033*thalach.

Let’s say that a new patient has the following test results:
The patient does not report chest pain : has_cp = 0
Resting systolic blood pressure: trestbps = 110
Heart rate achieved during exercise: thalach = 190

Applying the above equation to this particular patient: -2.923 + 2.014*0 – 0.022*110 + 0.033*190 = 0.927
This is the log(odds) for heart disease for this patient.
This can be converted to odds: Exp(0.927) = 2.526, So for patients with such test results, for every one who does not have heart disease, there is 2.5 patients on average who does have heart disease.
Probability: 2.526 / (1+2.526) = 0.716. So there is a 71.6% probability that the patient has heart disease.

Assessing goodness of fit

The binary logistic regression procedure in SPSS does not provide many of the important indexes that are needed to accurately assess model fit. So if assessing model fit is important (it usually is), we need to use another procedure. Fortunately the multinomial regression procedure can produce all the important outputs provided by the binary regression procedure, and it also provides the right model fit parameters. So in most cases it is better to run our logistic regression analysis with this procedure.

This can be accessed in **Analyse > Regression > Multinomial logistic…** Here the dependent is set to has_heart_disease.
Reference levels for categorical variables

Importantly, in multinomial logistic regression the default reference level is the last level of the factor (which is determined by alphabetical order if it is not set otherwise). So in order to make the interpretation similar to the regular and the binomial logistic output, we need to specify in the **Reference Category** menu (right below the dependent box), that we would like to use the First level of the factor as the reference level (instead of the last). This way the reference group will be set to has_heart_disease = 0, just like in the binary logistic regression.

Now we have to specify the predictors of the model. We can specify two types of predictors: factors (for categorical predictors) and covariates (for continuous predictors). We have one categorical predictor, has_cp, which we normally would put into the factors box, but as I mentioned above in multinomial regression the reference category by default is the last level of the factor again. In the regular linear regression and the binomial logistic regression procedure so far we were used to using 0 as a **reference level** for everything, that is, the intercept represented the predicted value of the outcome if every predictor's value is 0. However, if we entered has_cp as a factor predictor, we would use has_cp=1 as a reference level, which would mean that the intercept would represent people who have chest pain (has_cp=1) instead of people who do not have chest pain (has_cp=0) which was the case so far in the previously used regression analyses. We have two options to change this behavior and force SPSS to produce an output where the reference group is has_cp=0. Option 1 is to create a new variable where has_cp is reverse-coded (we could call that has_no_cp), where those people would get a value of 1 who do not report chest pain, and those get 0 who do report chest pain. We could use this predictors as a factor predictor then, and get the same regression coefficients as in the binary logistic regression. An easier option is to just enter has_cp as a covariate predictor instead of a factor predictor. This should be fine as long as we do not have any interaction terms in our model. I have chosen this latter option.

Whatever option you chose in your analysis, **please take extra care to note which are the reference levels of your categorical variables in your model**, both in the case of the dependent variable and the predictors.

Entering continuous predictors

Continuous predictors should be entered as covariates into the model.

Setting other options

By default, this procedure enters only the main effects of all predictors into the model. If you would like to enter interactions, you could specify this in the Model… menu, either by selecting full factorial and entering factor predictors which you want to enter the interaction of, or by manually specifying which predictors should interact by selecting the Custom/Stepwise option in the Model… menu. In our example we have no interactions in the model, so we will leave this at its default setting.

In the Statistics… menu we should ask for Information criteria to get the AIC, and for Classification table aside from the boxes checked by default.

Now we can run our analysis.

```
*Logistic regression analysis through multinomial regression procedure

NOMREG has_heart_disease (BASE=FIRST ORDER=ASCENDING) WITH has_cp
trestbps thalach
  /CRITERIA CIN(95) DELTA(0) MXITER(100) MXSTEP(5) CHKSEP(20)
LCONVERGE(0) PCONVERGE(0.000001)
    SINGULAR(0.00000001)
  /MODEL
  /STEPWISE=PIN(.05) POUT(0.1) MINEFFECT(0) RULE(SINGLE)
ENTRYMETHOD(LR) REMOVALMETHOD(LR)
  /INTERCEPT=INCLUDE
  /PRINT=CLASSTABLE PARAMETER SUMMARY LRT CPS STEP MFI IC.
```

Interpreting goodness of fit

When we look at the Parameter Estimates table in the output, now we can see that all of the regression coefficients match with the coefficients returned by the binomial logistic procedure. If this is not the case, the issue is probably with the reference levels, and you should make sure that you interpret the results according to the correct reference levels, or redo the analysis to match the reference levels used in the binomial logistic procedure.

Parameter Estimates

has_heart_disease a		B	Std. Error	Wald	df	Sig.	Exp(B)	95% Confidence Interval for Exp(B) Lower Bound	 Upper Bound
1,00	Intercept	-2,923	1,480	3,902	1	,048			
	thalach	,033	,007	22,125	1	,000	1,034	1,020	1,048
	trestbps	-,022	,008	6,776	1	,009	,978	,962	,995
	has_cp	2,014	,291	48,019	1	,000	7,494	4,240	13,247

a. The reference category is: ,00.

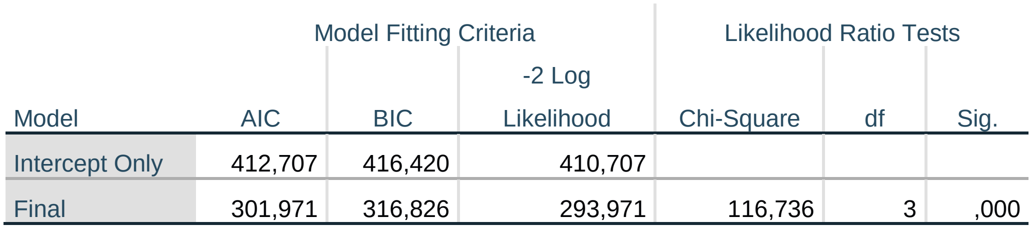
Is the model significantly better than the null model?

Just like in a regular linear regression we could be interested in whether it is useful to take into account our predictors in the first place, so whether the model as a whole is significantly better at predicting the outcome than the null model (which has no predictors).

To determine whether the model is significantly better than the null model, we should check the AIC value of the null model (Intercept only), and the Final model, using the previously learned criteria: if the difference is at least 2 points in **AIC** between the two models, the model with the lower AIC has a significantly better fit.

We can also check the **Chi-squared test** result, which should return the exact same values as we found earlier in the Omnibus test table in the binary logistic procedure. This test can be interpreted the same way as the F test (ANOVA) in regular linear regression. If this test returns a significant result, it means that the model has a significantly better fit compared to the other models it is compared to (in this case, the null model).

Model Fitting Information



	Model Fitting Criteria			Likelihood Ratio Tests		
			-2 Log			
Model	AIC	BIC	Likelihood	Chi-Square	df	Sig.
Intercept Only	412,707	416,420	410,707			
Final	301,971	316,826	293,971	116,736	3	,000

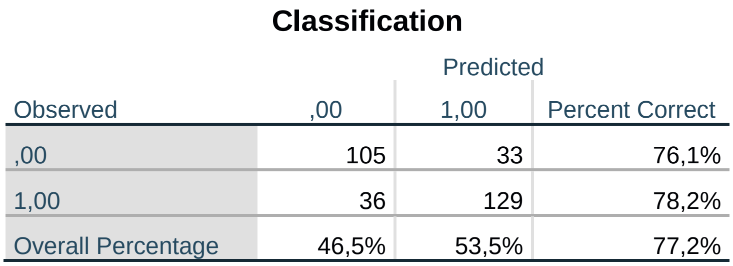
How good is our model? What is the predictive power of the model?

Another relevant question might be how effective our model is in predicting the outcome value. The Model Fitting Information table also shows us the **-2 Log likelihood** (-2LL), which is also called “deviance” in the literature. This has the same general meaning as the residual sum of squares (RSS) in regular regression. It compares the difference in probability between the predicted outcome and the actual outcome for each case and sums these differences together to provide a measure of the total error in the model. Just like RSS, and AIC, -2LL is hard to interpret outside of the context of the specific model, since its value depends on sample size, the number of parameters in the model and the goodness of fit. What is important to understand is that it shows the amount of error left after accounting for all of the variance explained by the predictors in our model, and that the lower this number the better the model fit is.

Another familiar indicator of the model’s predictive power is the R^2 index. However, there is no exact R^2 index for logistic regression. Instead, the proportion of explained variance is estimated using some procedure, they are called pseudo R squared. There are multiple estimates produced by SPSS, the binary logistic procedure only produces Cox and Snell R^2 and Nagelkerke R^2. Neither of these are well accepted in the literature. The issue with Cox and Snell R^2 is that it has an upper limit which is lower than 1. For those of us who are used to the good old R^2 index which can take any value between 0 and 1 and thus represents the proportion of explained variance, this might make the Cox and Snell R^2 hard to interpret. The Nagelkerke R^2 was devised to counteract this by expanding the scale of the Cox and Snell R^2 so that it is able to extend to 1. However, the correction used for this is often seen as an overcompensation and this can return unrealistically large R^2 values. Currently one of the more accepted R^2 indexes out there is the **McFadden R^2**. This is not produced by the binomial regression procedure, that is one of the reasons we chose to run the analysis through the multinomial regression procedure, which produces the McFadden R^2 by default.

One other important indicator of model performance could be the **number of cases correctly classified** by the model in the dataset. This information can be seen in the Classification table. The classification table shows that if we used our model to predict the outcome in our sample, how many of the times would the prediction be correct. The prediction is based on the probability of the event of interest predicted for each case by the model, of the probability is at least 50%, the model would predict an outcome value of 1, if it is lover than 50%, it would predict an outcome value of 0. (In some real-life scenarios it would make sense to use a different classification cutoff value than 50%. You can specify this if you want in the Options menu in the binary logistic regression procedure.)

In the Classification table you can check how many of the cases that actually have a value of 0 on the dependent variable had a predicted value of 0 and a predicted value of 1, and the table contains the same information about people who actually have a value of 1 on the dependent variable. The percentage correct column tells us the proportion of correctly predicted cases by the full model. This should be interpreted in the light of the actual occurrence of the more likely outcome of in the sample, because this is the correct prediction performance of the null model. So the null model is able to correctly predict the outcome 54.5% of the times, since this is the observed occurance of the more likely outcome in the sample: has_heart_disease = 1. By contrast, the final model was able to correctly predict the outcome in 77.2% of the cases overall.



Classification

	Predicted		
Observed	,00	1,00	Percent Correct
,00	105	33	76,1%
1,00	36	129	78,2%
Overall Percentage	46,5%	53,5%	77,2%

If you are interested in what is the estimated probability for each case and what is the predicted outcome value for each case, you can ask for these in the **Save menu** within the multinomial regression procedure: check the “predicted category” and the “actual category probability” boxes. The probabilities saved in the dataset this way should match the predicted probabilities that you can calculate by hand using the regression equation formula, and converting the log(odds) to probabilities.

What to report

Statistical analysis:

In our case, we would report the following about the methods we used:

“We have conducted a binomial logistic regression analysis where the presence of heart disease was the dependent variable (“no heart disease” was the reference level) and the model included three predictors: 1) whether or not the patient reported chest pain (has_cp, categorical variable with “no chest pain” as the reference level), 2) resting systolic blood pressure (trestbps), 3) heart rate achieved during exercise (thalach).”

Results:

In the results section we should essentially report the same things as for linear regression.

First of all, we should report about the **overall model fit and classification performance**:

“The final logistic regression model with all three predictors had a significantly better model fit than the null model (Chi^2 = 116.74, df = 3, p < 0.001, AIC of final model = 301.97, AIC of null model = 412.71). The model explained 28% of the variance (McFadden R^2 = 0.28). Heart disease occurred in 54.5% of the cases in our sample (165 out of 303 individuals). The final model correctly predicts the presence of heart disease in 78.2% of the cases, and the absence of heart disease in 76.1% of the cases in our sample, the overall correct prediction rate was 77.2%.”

Second, you would usually report the **information about the regression coefficients of each predictor**, and information about the Chi^2 test of added predictive power of each predictor. You can report these in a table format, and you can find these data in the Parameter estimates table and the Likelihood Ratio Test table in the Multinomial logistic procedure output. The reduced model AIC is informative because it can show the relative importance of the predictors in the model (the value shows how much would the AIC of the model be if that predictor would be excluded from the model). The Chi^2 test shows whether the predictor has a significant unique added value to the predictive performance of the model. If the confidence interval (CI) of the odds ratio (OR) for the predictor includes 1, it also indicates that we are not confident enough to say that the predictor has a significant added value to the model. So the report would go something like this:

“Table 1. contains the information about the predictors in the model. All predictors had a significant added value to the model. As shown in the table, the most influential predictor was the presence of chest pain (reduced model AIC = 352.69, OR = 7.49 [4.24 - 13.25]).”

Table 1. Model coefficients and predictor statistics.

	b	Odds Ratio (OR)	95% CI lb of OR	95% CI ub of OR	AIC of reduced model	Chi^2	p-value
Intercept	-2.92				303.97	4.00	.045
has_cp	2.01	7.49	4.24	13.25	352.69	52.72	<.001
trestbps	-.02	.98	.96	.995	307.03	7.06	.008
thalach	.033	1.03	1.02	1.05	324.67	24.70	<.001

Note: All Chi^2 test dfs = 1. AIC of reduced model represents the AIC of the model excluding that model component/predictor.

Model diagnostics for logistic regression

Model diagnostics of Logistic regression can be done using guidance from the following link:

The following section is written up based on the following guides:
http://www.restore.ac.uk/srme/www/fac/soc/wie/research-new/srme/modules/mod4/9/index.html
https://www.statisticssolutions.com/assumptions-of-logistic-regression/

Influential cases

Before assumption checks, it is always a good idea to explore potential influential cases. We can do this the same way as for linear regression: In the Save menu of the Binary Logistic Procedure we can ask for Cook’s distances. Cook’s distances can be plotted on a scatterplot against participant ID as we have seen for linear regression, and the raw values can be explored using the descriptive statistics as well. Remember that just because a Cook’s distance is high, that alone does not warrant excluding a case from the data. You should consider whether the case is a realistic one, that you would like your model to apply to, because throwing out a case will make your model “blind” to that case.

Logistic regression has similar assumptions to linear regression, but the error terms (residuals) do not need to be normally distributed, homoscedasticity is not required, and the dependent variable in logistic regression is not measured on an interval or ratio scale.

The following assumptions apply:

Binary dependent

1) binary logistic regression requires the **dependent** variable to be **binary**
- This assumption is obvious to check.

Observations independent

2) logistic regression requires the **observations** to be **independent** of each other
- This assumption should be checked based on our knowledge about the research design and the sampling procedure. For example if some participants come from the same school, class, institution, etc. this might make the observations related to each other, violating this assumption.

No multicollinearity

3) logistic regression requires there to be little or **no multicollinearity** among the independent variables.

- We can check this by requesting a correlation table of all of the predictors in our model, and we can look for very high correlations (0.8 or higher might cause problems related to multicollinearity)
- We do not get a VIF statistic in the logistic regression procedures, but if we want to get VIF statistics, it is possible to run our model using the linear regression procedure and ask for the collinearity statistics, to further explore potential multicollinearity.

Linearity

4) logistic regression assumes linearity of independent variables and log odds. Although this analysis does not require the dependent and independent variables to be related linearly, it requires that the continuous **independent variables are linearly related to the log odds**.
- This assumption can be checked by using the Box-Tidwell procedure (Box & Tidwell, 1962), which was developed for linear regression, but is also appropriate for logistic regression models. (Full reference: Box, G. E. P., & Tidwell, P. W. (1962). Transformation of the independent variables. Technometrics, 4, 531-550.) This assumption should only be checked for continuous predictors. The basic idea is to create a new variable that is the natural logarithm of the predictor, and enter the interaction term of this variable and the original predictor into the model. If the interaction term is significant in the coefficient's table, it may indicate a violation of the linearity assumption. Here is a detailed description of the use of this procedure in SPSS: https://event-mohs.gov.mm/wp-content/uploads/2019/12/Logistics-regression.pdf

Sample size

5) Finally, logistic regression typically requires a **large sample size**. A general guideline is that you need at minimum of 10 cases with the least frequent outcome for each independent variable in your model. For example, if you have 5 independent variables and the expected probability of your least frequent outcome is .10, then you would need a minimum sample size of 500 (10*5 / .10).