Logistic Regression: Script Lecture 2

Michaela Kreyenfeld (in collaboration with Cristina Samper)

Fall 2019

1 Multivariate Analysis

1.1 OLS

Regression techniques with micro level data have been dominating social science research since decades. Some researchers even argue that we live in the “age of regression“. The growing availability of large scale micro level data sets together with growing power of PCs has surely fueled this development.

One reason why we estimate a regression model is that we seek to reduce complexity. If we were interested, for example, how age, gender, region and education influences preferences for government spending, we could estimate various cross-tables. A single regression model will synthesize all analysis into one step. The major reason why we estimate multivariate models is that we seek causal effects. We will later see that it is a very difficult business to identify causal effects. However, regression analysis is an important step in this direction.

One of the main reasons for conducting regression models is to control for confounders. We can also phrase it a bit differently: we want to reduce omitted variable bias. Another way of putting it: We fear that our results are biased because of unobserved heterogeneity that we did not account for in the model. Our correlation may be spurious because we omitted central variables. All these formulations mean fairly similar things: They suggest that our model generates a pattern that is due to the fact that we forgot to control for an important variable. A classic example of this kind is the correlation that exists between the prevalence of storks in a region and the birth rate. This example does not hold water anymore. However, in former times, there used to be a strong correlation between the amount of storks and the birth rate. Although that the correlation was high, we would usually not follow from this that the storks brought the babies. Instead, it is clear that we forgot to account for an important “confounder”, namely urbanity. In rural areas, we find both a high prevalence of storks and high birth rates. Subsequently, it is not that the storks “cause” the high birth rates, but rather the family friendly living

conditions common in rural areas. Conventional wisdom in this context shows that "correlation is not causation".

In order to establish causality, a standard procedure is to control for "relevant" covariates that may bias your estimation. We will later evaluate more critically the notion of "relevant" covariates. However, for the time being, we accept this notion. The basic idea is to "partial out" the relevant correlation. Take the example of the storks, we would like to understand the effect of "stork prevalence" on the birth rate, net of urbanity. If we hold urbanity constant, do we still see an influence of "stork prevalence" on the birth rate?

We control for covariates, because we assume that they are both correlated with our dependent variable as well as with our independent variable of interest. Thus, correlation between covariates is only "normal". Please note, however, that there may be too much correlation between covariates. A strong correlation may create problems which are usually titled as multi-collinearity. If the correlation is very strong, R may not be able to estimate the parameters. Even if R is able to estimate the model, it may be difficult to isolate the "true" effect. Let me nevertheless emphasize that there is a lot of dispute among econometricians whether multi-collinearity is a problem or not (see e.g. Gujarati, 1992).

The principle of a regression model is always the same. On the left hand is the dependent variable (also called the outcome variable). In case of OLS-regression, it is a continuous variable, such as income or height. On the right hand side of the equation, there is a constant term, the independent variables (exogenous variables, predictors, covariates), the beta-coefficients (or parameters) and the error term.

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + e_i$$

The linear regression that you will know from STATS I has a lot of advantages. A very appealing one (compared to the logistic regression) is that the model results are fairly easy to interpret. Imagine, for example, that you would be able to survey preferences for government spending with a continuous variable (with a mean around 5). You are interested in the effect of duration of public sector employment on preferences for government spending. In Table 1, you find the results from such an investigation and Figure 1 plots the results.

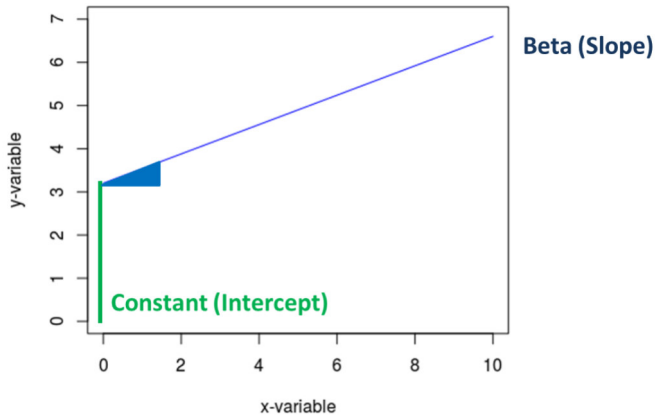
The interpretation of the results is very straightforward: A person who is new to public sector employment (duration=0) has a preference for government spending of 3.20. With each year that he/she stays in public sector employment, his/her preference increases by 0.34 each year.

Table 1: Results from OLS-Regression Dependent Variable: Preferences for Government Spending (continuos variable)[1]

```
linearmodel <- lm(VAR1 ~ DURATION, data=df)
as.data.frame(coef(linearmodel))
```

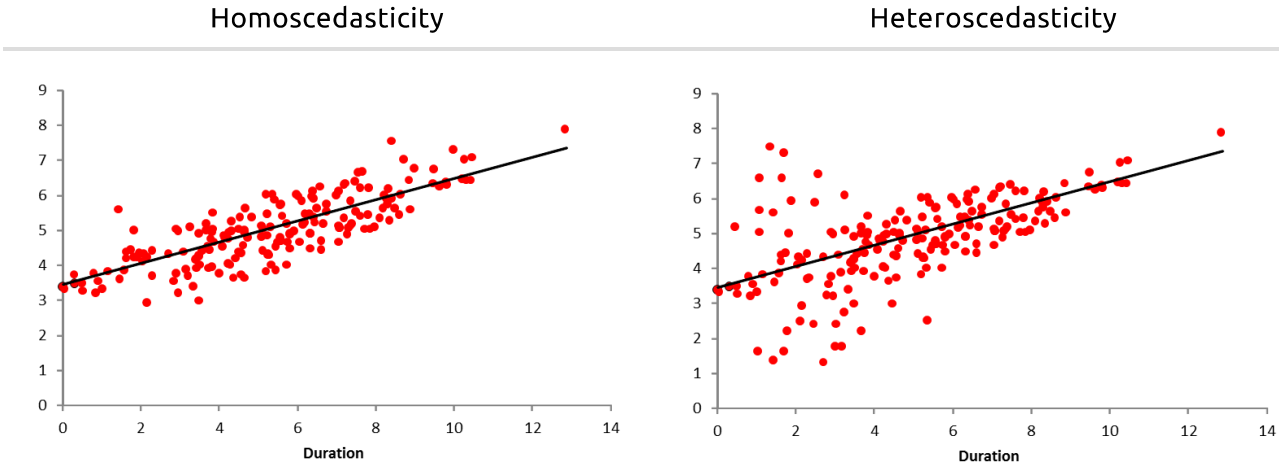
```
##             coef(linearmodel)
## (Intercept)         3.2024222
## DURATION            0.3368069
```

Figure 1: Plot of Results from Table 1



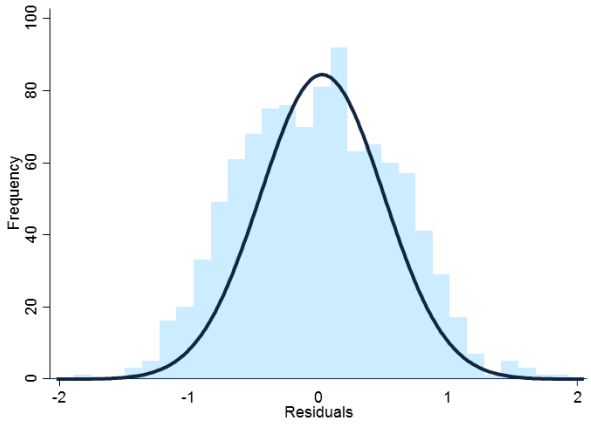
OLS is the common method to estimate a linear regression model. OLS is often referred to as BLUE, which means that it is the Best Linear Unbiased Estimator (e.g. Gujarati, 1992). Compared to other types of estimators, OLS provides estimates with the smallest variance, which means that the difference between the predicted Y and the actual Y is minimized. However, OLS relies on a couple of assumptions. One assumption is homoscedasticity which means that the error term is constant across all observations. This can be best shown graphically (see Figure 2).

Figure 2: Homoscedasticity and Heteroscedasticity



An additional assumption for OLS-regression is often that the error term is normally distributed with a mean zero (called the Zero Conditional Mean). In Figure 3, please find the distribution of the error term from the model above (see Table 1). Please note that it nicely matches a normal distribution that is plotted on top of it.

Figure 3: A Normally-Distributed Error Term



1.2 Linear Regression with a Binary Variable

How do the results change when the outcome variable is no longer continuous, but binary? In order to find out, we take the same data that we used above (see Table 1). However, we recode the dependent variable, which is currently continuous, into a dichotomous variable that equals 0 for persons with preferences for government spending lower than 6 (<6) and 1 for persons with preferences for government spending with a value of 6 or higher (≥6). The results from this investigation are listed in Table 2.

In principle, we could interpret the model results in terms of probabilities. The model shows that with each additional year of public sector employment, the probability that one opts for government spending increase by 8 percent.

Table 2: Results from OLS-Regression Dependent Variable: Preference for Government Spending (binary variable 0/1)

```
df$DUMMY <- -1
df$DUMMY[df$VAR1<6] <-0
df$DUMMY[df$VAR1>=6] <- 1

linearmodelD <- lm(DUMMY ~ DURATION, data=df)
as.data.frame(coef(linearmodelD))
```

```
##             coef(linearmodelD)
## (Intercept)        -0.26892887
## DURATION            0.08089921
```

Albeit that this interpretation makes sense, the model violates important assumptions that are commonly made for OLS-regression models. First of all, let's have a look into the regression line and how it runs through our real data (see Figure 4, left panel). The dependent variable can only take the value 0 or 1. Thus, the observations are clustered at two points of the y-axis, namely at 0 and 1. More importantly, however, the regression line does not run through the "cloud" of observations as it did with the OLS with the continuous variable (see above). What is very obvious from this figure is that the assumption of homoscedasticity is violated. A further aspect that is visible from this figure is that the predicted values from this model may be "out of the range" of possible values. We predict with our regression line negative preferences. Pampel (2000) summarizes these issues as follows:

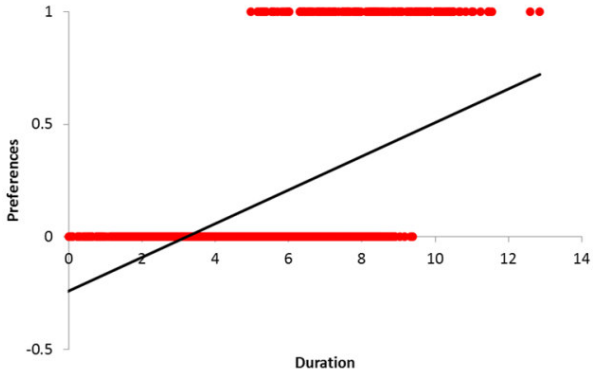
"... the linear regression line can extend upward toward positive infinity as the value of the independent variable increase indefinitely, and extend downwards toward negative infinity ... Thus, the model can give

predicted values for the dependent variable above 1 and below 0. Such values make no sense" (ibid.: 3).

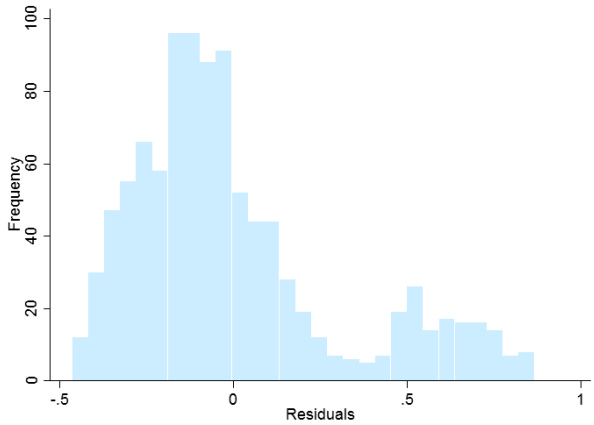
Figure 4 (right panel) plots the residuals from the model. Here it can be easily seen furthermore that the residuals are not normally distributed.

Figure 4: Results from OLS-Regression with a Binary Outcome

Regression Line and Observations



Residuals



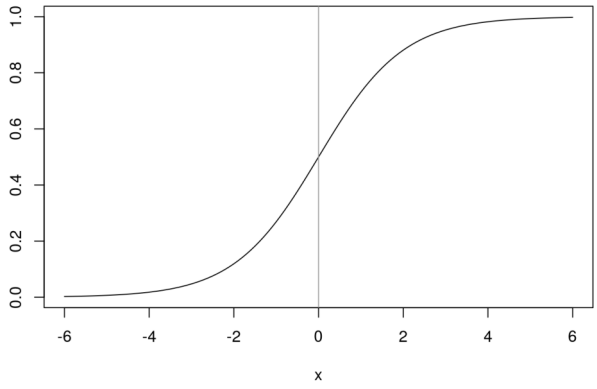
2 (Binary) Logistic Regression

2.1 The Dependent Variable: LN(Odds)

The shortcomings of the OLS-regression for analysing binary variables make the logistic regression an appealing framework. The logistic regression is a method designed for "qualitative" outcomes. The outcome may be binary, like in the binary logistic regression. However, we will later move on to other techniques such as the multinomial model, where the dependent variable has more than one outcome. For the time being, the binary logistic regression is at the heart of our interest. This means that the dependent variable is dummy coded (e.g., 0=no, 1=yes). The independent variables in the model may be either binary, categorical or continuous. However, the interpretation of continuous variables in a logistic regression is difficult and we will touch upon this topic later on. For the time being, we are interested merely in a binary outcome with categorical independent variables.

Let's take a step back and assume that our dependent variable is no longer 0 or 1. Instead, for each individual we want to estimate the probability that the outcome is positive, which is denoted with $P(x)$. We now need to find a functional form that "squeezes" our dependent variable into the right shape. "Right shape" means that the predicted values from the model would no longer be out of range, but that they would be between 0 and 1. Here the logistic function comes into play. Figure 5 plots the logistic function and the corresponding formula. The nice feature of a logistic function is that it transfers a variable in a way that it is bound between 0 and 1 (it takes x values from $(-\infty, \infty)$ and transforms them continuously into the $(0, 1)$ interval.

Figure 5: Functional Form of the Logistic Function



$$f(x) = \frac{1}{1+e^{-x}}$$

The logistic function is a "link function". It transforms the dependent variable in a way that the predicted values from the model are in range. In the "logistic formula", we may replace $f(x)$ by $P(x)$ and x by "xb". Thus, we arrive at the following relationship:

$$P(x) = \frac{1}{1 + e^{-x}}$$

Using some algebra, we may solve for bx and get a term on the left sight of the equation that we already know, the $LN(Odds)$: [2]

$$\ln \frac{P(x)}{1 - P(x)} = bx$$

2.2 Logistic Regression with R

2.2.1 Fitting a logistic regression

Base R offers the `glm` function to fit *generalized linear models* (GLMs). These models are flexible generalizations of the linear model. They allow the dependent variable to be non-normal. The logistic regression is of the most important subtypes of GLMs.

The form of the `glm` function is `glm(formula, family=familytype(link=linkfunction), data=)`

The `formula` is the expression that we want to calculate. Like in the linear model `lm` the formula is expressed as something like `DEP ~ IND1 + IND2` . In the case of the logistic regression `DEP` is a two outcome variable and `IND1` and `IND2` are the independent predictors. So for the `glm` function that we will use in this class to work smoothly, the dependent variable should be either coded (0,1) or be a two level factor. In case it is a two-level factor (which you can check with the function `class(DEP)`) the reference category (0 in the dummy coding) will be the first category to appear when you call the function `levels(DEP)` .

Since we are interested in estimating a model that will take a binomial dependent variable (a variable with two outcomes) we will insert `binomial` as the family type. [Fun fact: if you insert `gausian()` as the family type instead you would obtain the same `lm` model we have calculated before]. Since we are estimating the logistic regression we will use the `logit` *link function* for which the logistic is the inverse. The `logit` is the default link function for the `binomial` family, so specifying `family=binomial` in the `glm` function will suffice to estimate a logistic regression. If you would like to read about other possible link functions within the binomial families you can refer to (Fox & Weisberg, 2018, pg. 233)

The `data=` refers to the dataframe where you have the variables you are refering to in your `formula`.

Since the link function that we are using is the `logit`, the results we obtain from function are in the form of the beta-coefficients. Beta-coefficients can be complicated to interpret, for this reason we will learn to transform them into odds ratios (OR). In order to transfer these beta coefficients into OR we just need to elevate e^{b_0} to the power of coefficient b_0. In a similar manner if we then take the natural logarithm (the `log()` function in r) of the OR we again obtain the beta coefficient: $log(e^{b_0}) = b_0$.

```
logitmodel <- glm(DUMMY ~ AGE_DUMMY ,family=binomial, data=issp2016_Germany)
summary(logitmodel)
```

```
##
## Call:
## glm(formula = DUMMY ~ AGE_DUMMY, family = binomial, data = issp2016_Germany)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5434  -1.5224   0.8512   0.8679   0.8679
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.82888    0.07955  10.419   <2e-16 ***
## AGE_DUMMY1  -0.04663    0.10748  -0.434    0.664
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2023.8  on 1635  degrees of freedom
## Residual deviance: 2023.6  on 1634  degrees of freedom
## AIC: 2027.6
##
## Number of Fisher Scoring iterations: 4
```

In this example, exp(.83) is 2.29. In order to get from the OR (2.29) to the beta coeffient again you can take the natural logarithm of 2.29 Thus, ln(2.29) is .83.

```
coef(logitmodel)
```

```
## (Intercept)  AGE_DUMMY1
##  0.82887879 -0.04663162
```

```
exp(coef(logitmodel))
```

```
## (Intercept)  AGE_DUMMY1
##   2.2907489   0.9544389
```

```
log(exp(coef(logitmodel)))
```

```
## (Intercept)  AGE_DUMMY1
##  0.82887879 -0.04663162
```

2.2.2 Controlling for Categorical Variables

To fit a `glm` model with categorical variables it is a good habit to have them in `factor` class. This will allow you later to change reference category if you find it necessary. To change a variable into factor form you can use the function `as.factor(IND1)` . As stated above, when you have a factor, the reference category for that variable will be the first level to appear when you use the function `levels(IND1)` . Internally, R breaks down a categorical variable into several dummy variables. In the model, it leaves out one of the dummy variables, namely the one that is specified to be the reference category and uses it to compare with the other categories independently.

In the example below, there is a categorical factor variable AGE_CAT with three realizations (`1_young, 2_middle, 3_old`). Due to the numbering used, unless you tell it otherwise R will order the levels of the variable from least to greatest. So when we insert AGE_CAT into the model one tells R, `1_young` is the reference category, meaning that it will omit this category from the model, and take it as a baseline. The model results show that the odds of medium old are 20 percent lower than the odds for the reference category. For the old, the odds are reduced by 22 percent compared to the reference category. The constant (2.648) gives the odds when all other covariates are zero. Thus, it refers to the odds of young respondents.

```
levels(issp2016_Germany$AGE_CAT)
```

```
## [1] "1_young"  "2_middle" "3_old"
```

```
summary(logitmodel)
```

```
##
## Call:
## glm(formula = DUMMY ~ AGE_CAT, family = binomial, data = issp2016_Germany)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6088  -1.4983   0.8005   0.8792   0.8874
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.9738     0.1050   9.276   <2e-16 ***
## AGE_CAT2_middle   -0.2227     0.1354  -1.644   0.1002
## AGE_CAT3_old      -0.2452     0.1388  -1.767   0.0773 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2023.8  on 1635  degrees of freedom
## Residual deviance: 2020.1  on 1633  degrees of freedom
## AIC: 2026.1
##
## Number of Fisher Scoring iterations: 4
```

```
exp(coef(logitmodel))
```

```
##     (Intercept) AGE_CAT2_middle    AGE_CAT3_old
##       2.6480000       0.8003788       0.7825613
```

If you would like to change the reference category with a factor you can use the `relevel()` function. It works if you insert it directly into the model.

Now the results must be read with respect to the `2_middle` category. The odds of the young are 25% higher than for middle respondents.

```
levels(relevel(issp2016_Germany$AGE_CAT, ref = "2_middle"))
```

```
## [1] "2_middle" "1_young"  "3_old"
```

```
logitmodel <- glm(DUMMY ~ relevel(AGE_CAT, ref="2_middle") ,family=binomial, data=
issp2016_Germany)
summary(logitmodel)
```

```
##
## Call:
## glm(formula = DUMMY ~ relevel(AGE_CAT, ref = "2_middle"), family = binomial,
##     data = issp2016_Germany)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6088  -1.4983   0.8005   0.8792   0.8874
##
## Coefficients:
##                                           Estimate Std. Error z value
## (Intercept)                                0.75113    0.08557   8.778
## relevel(AGE_CAT, ref = "2_middle")1_young  0.22267    0.13544   1.644
## relevel(AGE_CAT, ref = "2_middle")3_old   -0.02251    0.12474  -0.180
##                                           Pr(>|z|)
## (Intercept)                                 <2e-16 ***
## relevel(AGE_CAT, ref = "2_middle")1_young    0.100
## relevel(AGE_CAT, ref = "2_middle")3_old      0.857
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2023.8  on 1635  degrees of freedom
## Residual deviance: 2020.1  on 1633  degrees of freedom
## AIC: 2026.1
##
## Number of Fisher Scoring iterations: 4
```

```
exp(coef(logitmodel))
```

```
##                                 (Intercept)
##                                   2.1194030
## relevel(AGE_CAT, ref = "2_middle")1_young
##                                   1.2494085
##   relevel(AGE_CAT, ref = "2_middle")3_old
##                                   0.9777387
```

Please make sure that your categorical variables are factor variables. You can use the option `relevel()` *in R to specify a reference category. Imagine, for example, that you are interested in the effect of gender (coded with 1 and 2) on a y-variable. If you insert your "gender-variable" into the model without specifying that it is a factor/categorical variable, R assumes that it is a continuous variable. You will get correct estimates of your beta-coefficient, but the constant will be meaningless. It gives the odds of a person with "0" gender.*

2.3 Interpretation of the Model Results: Odds and Odds Ratios

2.3.1 The Constant

In general, one does not interpret the beta-coefficients in a logistic regression model. Instead, one re-

arranges the formula, so that the dependent variable is no longer the LN(Odds), but the Odds. Let's first take a simple case, where we only have a constant term:

$$\ln(\frac{P(x)}{1-P(x)}) = b_0$$

We can re-arrange the formula and one gets the exp(b) on the right side of the equation. Thus exp(b) will give you the "effect" on the odds of a positive outcome.

$$\frac{P(x)}{1-P(x)} = e^{b_0}$$

The exp(b) of the constant gives the "base level odds" (Kleinbaum & Klein, 2012). It is the odds when all other covariates are zero. In our case, there are no other covariates in the model, thus, e^{b_0} is just the odds of a positive outcome in the respective sample.

2.3.2 Binary covariates

How does the story change when you have other covariates? The formula is now as follows:

$$\ln(\frac{P(x)}{1-P(x)}) = b_0 + b_1 x$$

Let's take the example of preferences for government spending (coded with 0 and 1). Age is the only other covariate. It can take two values young (x=0) and old (x=1). Table 3 gives the results from the ISSP 2016. In principle, we can interpret the model just like any other regression model. A unit-change in our covariate (age) changes the dependent variable by the same unit, thus by -0.0466. The problem is that the dependent variable is the $LN(Odds)$. A unit change in the $LN(Odds)$ is difficult to conceptualize. For that reason, we again re-arrange the formula and "remove" the logarithm from the odds in two ways: transforming it into a probabilities (for the formula, see under "mathematics" above) or raising e to the power of the coefficients to extract the Odds. Euler's number, 'e' raised to the power of coefficient b, gives the effect that one unit change of x has on the odds. A one unit change of x means than we compare the odds of old (x=1) and the odds of young (x=0) respondents. Thus e^b also written as "Exp(b)" is an OR that we are already familiar with.

Thus, the important "take-home message" is here: For binary variables e raised to the power of the coefficients may be interpreted as Odds Ratios.

Table 3: Results from Logistic-Regression Dependent Variable: Preferences for Government Spending (binary variable 0/1)

```
table
```

```
## # A tibble: 3 x 3
##   ` `                        b `Exp(b)`
##   <chr>                  <dbl>    <dbl>
## 1 Constant(Intercept)  0.829      2.29
## 2 Age_young            0          1
## 3 Age_old             -0.0466     0.954
```

3 References

Fox, J., & Weisberg, S. 2018. *An R companion to applied regression.* Sage Publications.

Gujarati, Damodar. 1992. *Essentials of Econometrics.* New York et al: McGraw-Hill.

Hosmer, David W.Jr., Stanley Lemeshow, and Rodney X. Sturdivant. 2013. *Applied Logistic Regression.* Vol. 3. Wiley.

Kleinbaum, David G. , and Mitchel Klein. 2012. *Logistic Regression: A Self-Learning Text.* Vol. 3. Springer.

Pampel, F.C. 2000. *Logistic Regression: A Primer.* Thousand Oaks: Sage.

4 Acknowledgment

For many helpful comments, extensions and revisions, I thank Cristina Samper. For helpful comments and language editing, I thank Lena Klein. I also thank Florien Kruse who commented on an earlier version of this manuscript. All remaining errors are my own. (I am always happy to receive comments and suggestions on this document: kreyenfeld@hertie-school.org (mailto:kreyenfeld@hertie-school.org)).

5 Appendix

5.1 Code for simulation

```
set.seed(1409)
VAR1 <- rnorm(1000, mean=5, sd=1)
set.seed(1409)
DURATION <-((VAR1-2)/0.5)-(runif(1000)*5)+1.8
df <-as.data.frame(cbind(VAR1, DURATION))
linearmodel <- lm(VAR1 ~ DURATION, data=df)
```

5.2 Algebra to transform LN(Odds) to P(x)

Below, you find the different steps that you need, if you want to re-arrange the probability formula $P(x)$ to the $LN(Odds)$ equation we already know.

Note: in step 2 for simplification purposes we substitute $b_0 + b_1x_1 + b_2x_2$ with bx, but the algebra does not change if you take the longer expression

1. $\ln(\frac{P(x)}{1-P(x)}) = b_0 + b_1x_1 + b_2x_2$
2. $\ln(\frac{P(x)}{1-P(x)}) = bx$
3. $\ln(P(x)) - \ln(1 - P(x)) = bx$
4. $\ln(P(x)) = bx + \ln(1 - P(x))$
5. $P(x) = e^{bx+\ln(1-P(x))}$
6. $P(x) = e^{bx} \times e^{\ln(1-P(x))}$
7. $P(x) = e^{bx} \times (1 - P(x))$
8. $P(x) = e^{bx} - e^{bx} \times P(x)$
9. $P(x) + e^{bx} \times P(x) = e^{bx}$

10. $P(x)(1+e^{bx}) = e^{bx}$
11. $P(x) = \frac{e^{bx}}{(1+e^{bx})}$
12. $P(x) = \frac{e^{bx}}{(1+e^{bx})} \times \frac{e^{-bx}}{e^{-bx}}$
13. $P(x) = \frac{1}{e^{-bx} \times (1+e^{bx})}$
14. $P(x) = \frac{1}{e^{-bx} \times (1+e^{bx})}$
15. $P(x) = \frac{1}{e^{-bx} + e^{bx-bx})}$
16. $P(x) = \frac{1}{e^{-bx}+1}$
17. $P(x) = \frac{1}{1+e^{-bx}}$

---

1. This result is based on simulated data. To simulate the data with R refer to the code for simulation in the appendix.↩
2. In the appendix you can find the different steps that you need to re-arrange the formula above↩