

**Applied Statistics and Econometrics**  
**Professor Deep Mukherjee**  
**Department of Economic Sciences**  
**Indian Institute of Technology, Kanpur**  
**Lecture – 41**

**Regression Involving Dummy Variable in R**

Hello friends. Welcome back to the lecture series on Applied Statistics and Econometrics. And this is the second last lecture in the course. In today's lecture. We are again going to visit the use of R software for statistical and econometric computation and model estimation. So, before we discuss the use of R, let us have a look at today's agenda items.

(Refer Slide Time: 00:39)



So, here in today's lecture, we are going to take care of the Dummy variable analysis. So, in the last lecture, we have dealt with regression of continuous variables, mostly, but here the entire lecture of today is devoted to handling of Dummy or indicator variables in our platform. So, the lecture can be thought of half of split between two main concepts. One is basically how to make use of Dummy or indicator variable in regression where Y is continuous. So, the Dummy variable is basically there as an independent explanatory variable. And the second half of the lecture will be devoted to the special case when Dummy variable becomes the dependent variable. Yes, we are going to talk about the discrete choice models in this part.

So, let me start by telling this thing again, so the objective of these R lecture sessions is not to actually teach you how to handle R in live sessions. So, the objective behind these three lectures is very simple. I want to actually give you a very brief recap of the major concepts

that we have covered in the course. So, towards the end of the course, you get a feel that, these are the most important things that we know, we have learned in the course. And then, when it comes to statistical computation or estimation of different models and drawing statistical inference out of the raw data, how R software can be used.

So, here in this lecture also, I am not going to take live session on R, rather I will show you some pasted screenshots from the actual R environment when you are conducting some data analysis in R platform. And, there of course you are going to see the code or command that you need to write. And then, I will actually try to help you to interpret the results, what you see on screen.

(Refer Slide Time: 03:05)

**Dummy as independent variable**

$\pi$

- In R environment, we can introduce a dummy variable in regression model as follow
  - Step 1: Generate our dummy variable in R and keep MH as base category

```
MHMP4$d.state<- 0
MHMP4$d.state[MHMP4$state=="MP"]<-1
```
  - Step 2: Introduce the dummy as independent variable in regression model
    - For change in only intercept term:

```
Mod3<- lm(prodha~fertha+workha+nia+d.state,MHMP4)
```
    - For change in only slope coefficient:

```
Mod4<- lm(prodha~fertha+workha+nia+I((d.state*fertha)),MHMP4)
```
    - For change in both intercept & slope coefficient:

```
Mod5<- lm(prodha~fertha+workha+nia+d.state+I((d.state*fertha)),MHMP4)
```
- To see the regression outcomes, we have to use summary command. However, in order to compared the results with initial regression model we use "stargazer" option

2

So, let us start today's discussion with a Dummy as an independent explanatory variable. So, first you have to introduce or define a Dummy variable in the regression model in the R platform. And in the last lecture only I have talked about how to create dummy variables in the R set up, but it is not a bad idea to repeat the thing again so that you do not have to go back to the previous lecture and find out what was the code.

So, for this exercise, we are going to make use of the datasets that we have used in lecture number 40, that was the last lecture where we have used the agricultural data and at district level from two States, Maharashtra and Madhya Pradesh. So, if you remember in that dataset, we had one variable which was qualitative in nature. And the name of the variable was state. And, it has two labels, MP for Madhya Pradesh and MH for Maharashtra.

Now, I want to create a dummy variable and I want to declare that if the observation is from other state, then I am going to assign a value one to the dummy variable or indicator variable. And that is what you know I am going to do. So, here you see, I have declared a dummy variable here and I have kept MH the label for Maharashtra as the base category. So, basically now I have these dummy variable defined as d dot state. So, d here comes for dummy. You can give another name. It does not matter, but here at least in this lecture, I am introducing dummy variable as d dot operation.

Now, in step two, you have to introduce the dummy variable as the independent variable in the linear regression model. Now, there could be three different ways to make use of an indicator or a dummy variable in a linear regression setup. If you remember the discussions that we had, the theoretical discussions that we had in the course, a dummy can play three types of roles. The simplest possible case is that the dummy will act as a shift pair of the intercept term in the regression. The second is that the dummy variable can impact the regression coefficient of one of the explanatory variables. So, basically then there we are talking about slope change and finally the most complicated and comprehensive category of dummy analysis is that it says that the dummy variable can impact both the intercept and the slope coefficients of the regression model.

So, we are going to look at these three categories one by one. So, here, we are going to work with three different models. So, first let us focus on the model where we are expecting a change only in the intercept term. So, here I am defining model by declaring or calling it a model three. And the code, I am showing here and it starts with lm. So, let me remind you again. So, lm is basically stands for linear model because we are running a linear regression model and here the dependent variable is prodha so that is the monetary value of production per hectare of agricultural land.

And there are three inputs, and they are fertilizer use per hectare of land. Then labor input, which is represented by these variable workha. So, this is labor unit used per hectare of land. Then nia is basically net indicated area. And finally, now we incorporate the dummy variable d dot state and here, as you see that this dummy variable is not interacting with any of the continuous regressors here in the model, then we are talking about basically an expected change in the intercept term only.

Now, we move on to the second model and that we denote by model four. So, in model four, I am going to show you how dummy variable can be useful to model slope changes. So, we start with hypothesis where we assume that productivity of fertilizer per hectare of land varies across states because of change in agro-climate conditions, soil conditions, etcetera. So, here, we expect that the beta coefficient, the regression coefficient for the variable, fertha may vary from Madhya Pradesh to Maharashtra.

Now, this is a hypothesis. In reality, this could be the case or this could not be the case. So, we have to statistically test whether these average product of fertilizer indeed varies from one state to the other. So, the best way to do that is basically by introducing an interaction term in the linear regression model. So, you already have the dummy variable in the linear regression model three.

Now, we talk about model four where, we also incorporate the interaction term between the continuous variable fertha and the dummy variable  $d \cdot state$ . So, here I am showing you the code or command that you need to write for estimation of model four. So, here you see, I am incorporating the interaction term between the dummy variables  $d \cdot state$  and fertha.

And then finally we are going to come to model five, where we are going to talk about both intercept and slope coefficient changes due to the presence of dummy variable. So, now concentrate on the model for code or command that I have shown here. So, note that I have taken out the dummy variable,  $d \cdot state$  as it was there in model three. And now I am replacing that variable by this interaction variable,  $d \cdot state \cdot fartha$ . So, here we are talking about only the change in the slope coefficient, but not in the intercept.

Now, we come to the most general case. There we have model five and in model five, you see that I have both the dummy variable  $d \cdot state$  and the interaction between the dummy variable and the continuous variable fertha. So, model five the most general model that you can think in this context. So, basically we are now going to estimate three different models in R environment. And of course we know we would like to have a comparative picture.

Now, we know as I have shown you in the last lecture, when you get the output from our R, it is not very much presentable because, you know numbers are not presented in form of a nice table but does it mean that you cannot produce a regression table like in other software? Yes, you can. But for that, you have to make use of a package and I am showing you now how you can make use of one package to produce nice looking regression tables so that you can

compare more than two models side by side so that you can have a better idea how the coefficients are changing from one model to the other, or the model goodness of fit statistics is going to change from one model to the other. There are many packages, but in this lecture, I am going to show you the use of stargazer package option. So, of course you need to download and install this particular package option before you want to execute the next step, which is basically preparation of regression tables.

(Refer Slide Time: 11:35)

**Presenting estimated models in one single table**

```

> library(stargazer)
> stargazer(mod1,mod2,mod3,mod4, column.labels = c("Initial", "Intercept", "slope", "both"),
+ type = "text",
+ keep.stat=c("R", "rsq"))

```

	dependent variable:			
	Initial (1)	Intercept (2)	slope (3)	both (4)
fertha	100.415*** (24.037)	80.337*** (27.373)	104.335*** (24.509)	45.803 (45.270)
workha	2.872422 (2.156286)	3.479251 (2.362743)	3.088649 (2.573093)	3.725328 (2.377487)
nfa	95.588** (39.854)	131.108*** (54.384)	132.872** (58.498)	132.428** (57.809)
d.state		-1.044348 (2.053524)		-6.864537 (4.483324)
I((d.state * fertha))			-23.133 (26.688)	55.117 (37.322)
constant	3.951942 (4.042432)	4.821733 (4.042410)	3.198381 (4.142337)	7.708646 (3.044029)
observations	62	62	62	62
R2	0.332	0.357	0.341	0.388

note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

So, you see here, I am showing you a screenshot from R and you see the first line shows that in library, I have this package downloaded and installed. So, the stargazer is shown within parentheses. And then basically you write the command that you see in the second line. It is a little bit lengthy, but it shows you how to represent a different model. So, here, you have model one, which is basically the initial OLS model that we have estimated during the last lecture.

And I have shown that regression output there also, and now we have three new models with dummy variables or dummy plus interaction variables. And these are basically models, three, four, and five. And I am going to have a comparative picture of these four models. So, these will become now my columns with different levels and you can actually write the labels for each of these models.

So, say for mod one, you want to give a label of initial. So, that is basically our base model to start with. And then, you may wish to call model three as intercept model and model four as the slope model and model five as both. So, you just need to type these name of the models

within inverted quotation without inverted quotation, it is not going to work. And then basically you finish the quote by mentioning that what are the other things that you want to see in the table.

So, of course you want to show how many observations with which you have worked. So, that is given by small n within inverted quotation and then you want to get an idea about the R square from these regression so again within inverted quotation, you have to write rsq. So, now you run this code or command in R and you get this nice looking table presented to you. So, here you see for four columns, you are seeing the regression results for the dependent variable prodha and for the initial model you have coefficients for fertha, workha and nia only because of course that is the base model, which we talked about in the last lecture.

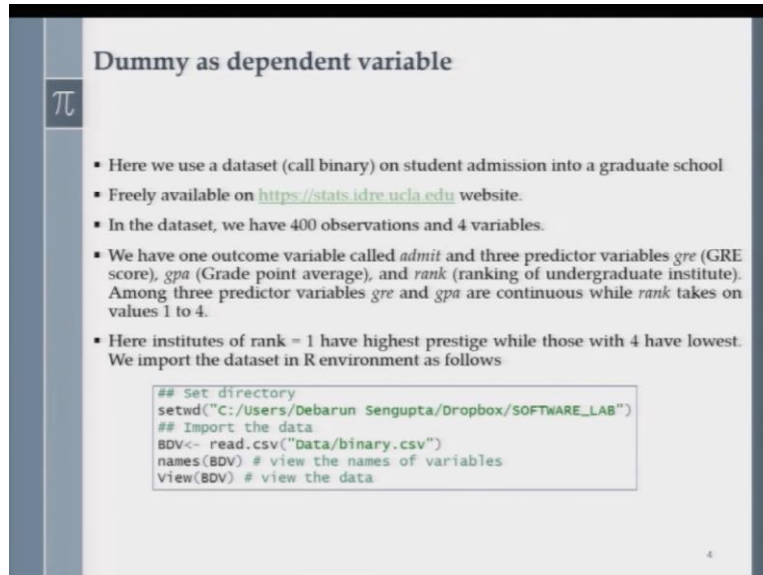
Now, you see, the models that we spoke about in the last slide involving dummy variables are presented in columns two, three, and four. And here you see that, unfortunately, the dummy variable is not turning out to be significant in any of these models neither is the interaction between the dummy variable and a continuous variable. Now, you cannot say that if you introduce a dummy variable and an interaction variable of the dummy with one of the existing continuous variables in the regression equation always you are going to get significant results. Now, it is experiment. I just wanted to show you that how to handle a dummy in R environment.

So, here in this simplified example, you see that dummy variable is not significant. So, it is not impacting the intercept parameter of the regression equation. It is also not impacting the slope, parameter of the fertilizer input use variable. So, that is why you see, we throw in a new variables, but they are not explaining the variation in Y that much. So, R square is also not changing by that much actually. So, you see the difference between the R squared is pretty low from one model to the other. So, you can say here that the state factor which is basically a fixed factor is not affecting the average productivity of fertilizer use on agricultural production.

Now, we move on to the exciting case of dummy as the dependent variable. So, we have had lengthy discussion on the logit models previously. So, we are going to talk about the estimation of logit model and interpretation of the results in R platform. So, for that of course we need a dataset to start with and we have got a dataset from one existing data source and

first we are going to explain the dataset to you, and then we are going to take you to R environment.

(Refer Slide Time: 16:45)



$\pi$

### Dummy as dependent variable

- Here we use a dataset (call binary) on student admission into a graduate school
- Freely available on <https://stats.idre.ucla.edu> website.
- In the dataset, we have 400 observations and 4 variables.
- We have one outcome variable called *admit* and three predictor variables *gre* (GRE score), *gpa* (Grade point average), and *rank* (ranking of undergraduate institute). Among three predictor variables *gre* and *gpa* are continuous while *rank* takes on values 1 to 4.
- Here institutes of rank = 1 have highest prestige while those with 4 have lowest. We import the dataset in R environment as follows

```
## Set directory
setwd("C:/Users/Debarun Sengupta/Dropbox/SOFTWARE_LAB")
## Import the data
BDV<- read.csv("Data/binary.csv")
names(BDV) # view the names of variables
View(BDV) # view the data
```

4

So, we went to the website that we are showing here and that is basically housed at university of California, Los Angeles. And there are some free datasets available for students. In fact, I suggest you to go and visit this particular website and download freely available datasets and try your hands on them because now in this course, you are going to have some idea about how to handle datasets in R. So, why do not you go there and download datasets and run some R codes that you have learned in these lectures just to have a feel. So, from that data source, we have downloaded a dataset and let us call it binary.

And it is basically on student admission into graduate school in the United States. So, in that dataset, we have a 400 observations. So, these are basically 400 student application cases. And there are four variables for each of the student cases. So, from the dataset we can say that there is one outcome variable which we can use for logit estimation and that is called *admit*. And the, there are three other predictor variables.

So, *admit* means that whether a student got admission into graduate school or not, and then there are three predictor variables or explanatory variables. One is the *gre* that is basically the GRE score. And then the second one is titled *gpa*. So, that is basically the Grade Point Average. So, that is basically, you can say the performance in the last educational program that the student was enrolled in. And then finally we have a variable called *rank* and that actually gives me the ranking of the undergraduate institution.

So, now, among these three predictor variables, gre and gpa are definitely continuous because they are dealing with the numbers, but the rank variable actually is qualitative in nature. And there are four levels, one to four. And here the institutes of rank one have the highest prestige while those with four have the lowest. And now we import the dataset in R environment as follows. So, you have to first set your directory as we know, we have spoken about that in the last lecture also. So, here I am showing you again how to make use of the state WD command, and then you need to input the data and read the data.

So, as I said that the data is saved in your computer as binary dot CSV. So, you have to use the file name you see and you can give it a different file name. It does not matter. The best way to start doing data analysis even in the context of discrete Y is to generate and look at the summary statistics figures because summary statistics tables actually tell you a lot about the variation in these variables. And it is just good to have a feel of the Y variable actually, especially which you are going to model.

(Refer Slide Time: 20:28)

**Descriptive statistics**

$\pi$

- In R, we compute the descriptive statistics for whole dataset as follow

```
> summary(BDV)# For whole dataset
```

admit	gre	gpa	rank
Min. :0.0000	Min. :220.0	Min. :2.260	Min. :1.000
1st Qu.:0.0000	1st Qu.:520.0	1st Qu.:3.130	1st Qu.:2.000
Median :0.0000	Median :580.0	Median :3.395	Median :2.000
Mean :0.3175	Mean :587.7	Mean :3.390	Mean :2.485
3rd Qu.:1.0000	3rd Qu.:660.0	3rd Qu.:3.670	3rd Qu.:3.000
Max. :1.0000	Max. :800.0	Max. :4.000	Max. :4.000

- In our dataset we have two categorical variables (admit & rank) and two continuous variables (gre & gpa).
- In case of categorical variables, frequency distribution is more important than its summary statistics. In R we can see the frequency distribution of admit variable as follow

```
> table(BDV$admit)
```

0	1
273	127

5

So, we actually get the summary statistics for the whole dataset by writing this simple command. And here, I am showing you all these measures that you have seen these measures several times in this course, like mean, median, quartiles and all, but we have, two categorical variables in our dataset, admit and rank. And it is not a very good idea to look at the discrete variables by looking at the values of media and, and quartiles and all because they are not continuous. But let us have a look at the admit variable. One that we are going to model.



So, you see the mean is going to say us something. So, mean says, a value is there 0.3175. So, it says that 31.75 percentage of the students in the dataset got enrolled in the graduate school. Okay. So, now, if we want to get the exact frequency counts of students who got admission in the graduate school and who did not get admission in the graduate, you can actually make use of one command call table for the dummy variable. And that is basically I am showing here in the box.

So, you see for zero value, I have 273 observations and for our one value of the dummy variable or indicator variable. I have 127 observations. So, now if compute the ratio 127 divided by 400. So, that should give you the proportion of the students who got admitted into the grad school. Right. And if you actually compute the ratio, you will get that number that I have talked about a couple of minutes back. And the number is 0.3175. So, that is basically the mean that is showing here.

(Refer Slide Time: 22:30)

**Contingency table**

$\pi$

- Contingency table (or cross tabulation) is a two-way frequency distribution table. For example, we want to know the frequency distribution of admit variable across different levels of rank variable then we use contingency table.
- Note that by-default R consider rank variable as continuous variable; in order to declare it categorical variable we use following command

```
> BDV$rank<-factor(BDV$rank)
```
- Once we declare rank as categorical, we generate contingency table as follows

```
> xtabs(~rank+admit, data = BDV)
```

rank	0	1
1	28	33
2	97	54
3	93	28
4	55	12

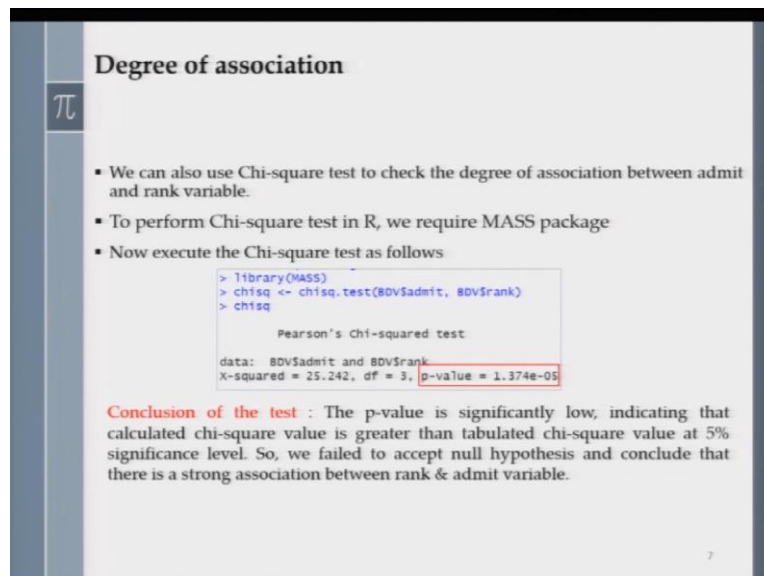
So, now, we are going to talk about contingency tables. Now, why do we want to talk about contingency tables? So, if you remember, in the first part of the lecture, we spent sufficient time on contingency tables because they are the best way to represent the interactions of qualitative variables and we can make use of these tables to conduct hypothesis testing so that we can see whether there is any association between two qualitative variables or attributes or not. And, although our major focus is to conduct a logistic regression in today's lecture, but it is not a bad idea to learn how one can conduct degrees of association tests between two qualitative variables, because many times in many disciplines, degrees of

association, Pearson's chi-square test are very useful. So, we are going to show you next how we are going to conduct a Pearson's chi-square test in R environment.

So, note that by default R consider rank variable as the continuous variable. So, we have to declare that the variable rank although it is taking a numbers like 1, 2, 3, 4, but it is not a continuous variable. It is basically a qualitative variable. So, in order to declare it as a categorical variable, you have to make use of a command. And that command, I am showing here in this box.

Now, once you have declared the rank variable as a categorical one, you can generate the contingency table by writing a simple command and that command is xtabs. So, basically this will produce the cross tabulation of two qualitative variables, admit and rank. And here you see in the rows, you have four different levels of the qualitative variable rank. And in the columns, you have two labels of the qualitative variable admit. And in the cells, you are going to see the frequencies.

(Refer Slide Time: 24:48)



**Degree of association**

$\pi$

- We can also use Chi-square test to check the degree of association between admit and rank variable.
- To perform Chi-square test in R, we require MASS package
- Now execute the Chi-square test as follows

```
> library(MASS)
> chisq <- chisq.test(BDV$admit, BDV$rank)
> chisq
```

Pearson's Chi-squared test

data: BDV\$admit and BDV\$rank  
X-squared = 25.242, df = 3, p-value = 1.374e-05

**Conclusion of the test** : The p-value is significantly low, indicating that calculated chi-square value is greater than tabulated chi-square value at 5% significance level. So, we failed to accept null hypothesis and conclude that there is a strong association between rank & admit variable.

7

Now, once the frequencies are opt-in, you can now perform the Pearson's chi-square test to check the degree of association between the two variables, admit and rank. Now, why are we interested in Pearson's chi-square test in this context? Because we can have a hypothesis that the rank of the undergraduate in a institution where from the student got his or her undergraduate degree may have an effect on the decision-making while the student case was being discussed and a decision has to be taken whether the student shall be permitted in a graduate school or not.

So, the prestige of the educational Institute, the ranking of that educational Institute from where you are coming as a student, undergraduate student may have some impact on the decision variable which is represented here by `admit`. So, the best way to look at whether there is any association between these two qualitative variables are not is by conducting hypothesis testing via Pearson's chi-square test.

Note that as usual chi-square test in R is not automatically done. So, you require a package and you require MASS package. So, you have to download and install that package in your machine. And then here you can follow the code. So, again, I am showing you the library command that you write to first check whether that MASS package is properly registered in R environment or not. And once you have a look at it then you can go for the chi-square test and chi-square test is conducted by writing the simple `chisq` command. And here is the outcome.

So, you see the p value is extremely small. And then how do you conclude your test? So, as p is very, very low, then you can say that the calculated Chi-square value is greater than the critical tabulated Chi-square value at 5 percent significance level. So, we reject the null hypothesis with which we started. And the null hypothesis says that there is no degree of association. So, finally we conclude that there is a strong association between the qualitative variables `rank` and `admit`. So, now we are going to finally move on to the discussion of logit models and I am going to make use of the same dataset that we used from UCLA data source to explain you the working of operational side of the logit model in R environment.

(Refer Slide Time: 27:46)

**Logit Model**

$\pi$

- The Code below shows how to estimate a logistic model in R.

```
##### logit model
logit<-glm(admit~gre+gpa+rank,family = binomial(link="logit"),BDV )
```

- Here `glm` stands for generalised linear model. In this `glm` function `admit` is the dependent variable and `gre`, `gpa`, and `rank` are independent variables.
- Once we specify both dependent and independent variables, then we define the argument for logistic regression. For the `family`, use `binomial` option.
- Note that by default R fits logistic model with `family` argument with `binomial` option.
- We save our `glm` function as `logit` and use `summary(logit)` option to get summary of the logistic model & all estimates.

8

So, here, I am showing you the code that you must write in order to estimate a logit model. Of course note that admit is the dummy dependent variable so that we are going to model. So, that is our Y, and that is discrete. And we are going to make use of two continuous variables, gre and gpa and one qualitative variable say rank.

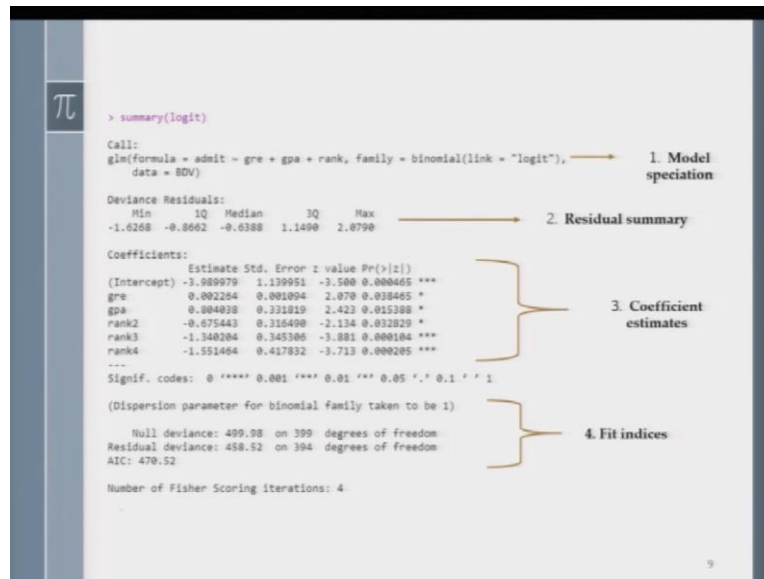
So, here you see that we have three independent explanatory variables, gre, gpa and rank. Now, you look at the code and then you see some jargons are there, which probably you have not seen before, like, this thing glm. So, there is a big difference between the code that we ran for the Willis case and what we are going to run now. Note that in the case of ordinary least squares estimation in R environment, we start the code by writing lm. So, that is basically a linear model, but now we are writing glm. So, we are adding an extra g in front of l and what does that mean?

So, here glm stands for Generalized Linear Model or General Linear Model. So, it is basically general regression set up, which can also handle non-linearities. So, you can say that the glm is a unification of both linear and non-linear regression models that also allows the incorporation of non-normal response distributions. So, if you remember in the context of logit model, I said that our response variable does not have a normal distribution. So, the response variable under a glm must be a member of the exponential family.

So, that is the only requirement. So, it can be normal, it can be Poisson, or it can be binomial or exponential or gamma distributions. So, here in this context, you see here, this word family that actually talks about the family of distributions from which you need to choose one to run your general linear model. And here, I am choosing binomial as the distribution from the exponential family of distributions. And I am actually using the logit link function to run the logit model. So, what is the logit link function?

I hope that you remember that discussion how we know made a logit transformation for the discrete dependent variable in the logit regression model lecture. Now, we know we save our glm function as logit, and then we can use the summary logit option to get the summary of the logistic model and you can get all the estimates.

(Refer Slide Time: 30:53)



The image shows a screenshot of an R console window with the following content:

```
> summary(logit)

Call:
glm(formula = admit ~ gre + gpa + rank, family = binomial(link = "logit"),
     data = BDV)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6268 -0.8662 -0.0388  1.1490  2.0790

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979    1.139951  -3.508 0.000465 ***
gre           0.902204    0.001094   2.078 0.038405 *
gpa           0.804828    0.232819   2.423 0.015308 *
rank2        -0.675443    0.316490  -2.134 0.032829 *
rank3        -1.348284    0.345386  -3.881 0.000184 ***
rank4        -1.551464    0.417832  -3.713 0.000205 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 458.52  on 394  degrees of freedom
AIC: 478.52

Number of Fisher Scoring iterations: 4
```

Annotations on the right side of the screenshot:

- 1. Model specification: points to the `Call:` line.
- 2. Residual summary: points to the `Deviance Residuals:` section.
- 3. Coefficient estimates: points to the `Coefficients:` table.
- 4. Fit indices: points to the `Null deviance`, `Residual deviance`, and `AIC` lines.

Now, we know this slide is showing you the screenshot of the R environment here, as we are estimating only one model. I am not showing you how to construct the regression table, but you can also make use of that stargazer command or package that I have discussed this lecture only. So, let us have a look at the regression outcome.

So, the first component of the regression outcome is basically the model specification. So, R actually tells you what model you have estimated then in the second block of results, you see the residual summary is given and median and quartile values are reported. And then in the third block of the regression results, you see the coefficients are reported.

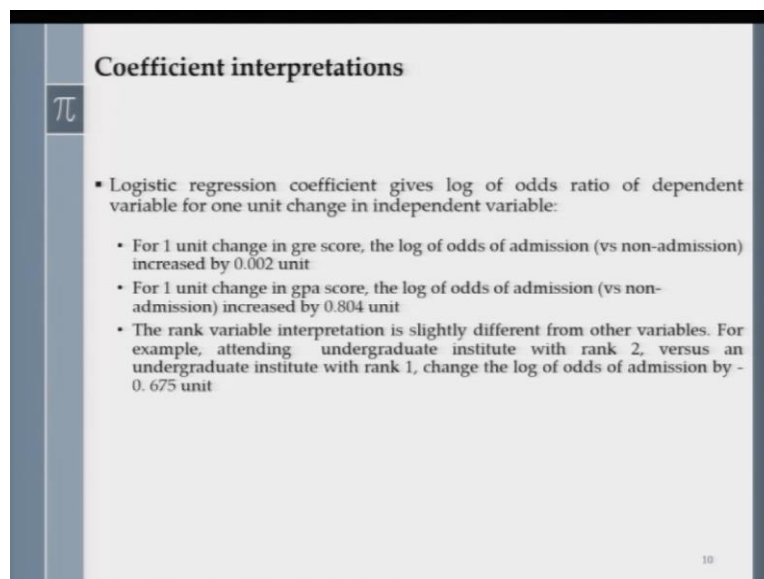
So, the coefficient estimates are reported here. And standard error for the regression coefficients are reported. The Z values are reported and the p-values are also reported. So, here you see that we are so lucky that we have all the explanatory variables turning out to be significant although at different levels, but at least three of them are significant at one percent level.

Now, the last block of the regression outcome shows you the fit indices. So, here you see, we are getting a measure of AIC, the Akaike information criterion. And finally, you are getting the number of Fisher Scoring Iterations four. Now, what do we actually mean by these number of iterations and these Akaike information criterion? Why are not we getting the familiar measure R square?

Note here, we are not running a wireless regression, so you shall not expect R square to be derived here. Here, if you remember the discussion on the logistic regression models, we have to add up the maximum likelihood estimation procedure to estimate these non-linear models. And generally, for non-linear models, R squares are not reported when (())(33:09) is the method to estimate. And for that the way one can choose between models is basically by looking at the Akaike information criterion or the Bayesian Information Criteria. So, here R reports the AIC criteria.

Now, as it involves the estimation of logit models, involves maximum likelihood estimations. So, there is an objective function which is basically the likelihood function which needs to be maximized in order to derive the regression coefficients. So, there has to be like iterations. So, numerically the software actually does iteration to find the maximize value of the log-likelihood function. So, here you see that after four iterations only the software claims that it has found a global maxima for the log-likelihood function. So, it is optimized after four iterations.

(Refer Slide Time: 34:13)



**Coefficient interpretations**

$\pi$

- Logistic regression coefficient gives log of odds ratio of dependent variable for one unit change in independent variable:
  - For 1 unit change in gre score, the log of odds of admission (vs non-admission) increased by 0.002 unit
  - For 1 unit change in gpa score, the log of odds of admission (vs non-admission) increased by 0.804 unit
  - The rank variable interpretation is slightly different from other variables. For example, attending undergraduate institute with rank 2, versus an undergraduate institute with rank 1, change the log of odds of admission by -0.675 unit

10

```

> summary(logit)

Call:
glm(formula = admit ~ gre + gpa + rank, family = binomial(link = "logit"),
     data = BDV)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0268  -0.8662  -0.0388   1.1490   2.0790

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.389979    1.139951  -3.560 0.000405 ***
gre          0.002204    0.001094   2.078 0.038465 *
gpa         -0.004038    0.331819  -2.422 0.015288 *
rank2      -0.675443    0.316408  -2.134 0.032829 *
rank3      -1.348284    0.345366  -3.881 0.000104 ***
rank4      -1.551464    0.417832  -3.713 0.000205 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 458.52  on 394  degrees of freedom
AIC: 470.52

Number of Fisher Scoring iterations: 4

```

So, now we are going to spend a couple of minutes on the interpretation of the coefficients. Now, the coefficient interpretation is a bit tricky in the context of logit model, because we are not talking about a linear link function. So, how do we actually interpret the coefficients coming out of our logit model? So, before we get into the interpretation of the Obtain regression coefficients in the last slide. Let us look at a simplest possible case where the logit model has only one regressor and there is no other explanatory variable involved.

So, basically the fitted value of the logit model at a particular value of  $X$ , say  $X_i$  would be obtained as  $B_0 + B_1 X_i$ . Now, the fitted value at another value of  $X$ , say  $X_i + 1$  then will be  $B_0 + B_1 (X_i + 1)$ . So, the difference in the two predicted values will be now  $B_1$  only.

So, the fitted value, when you are using the specific value  $X_i$  talks about the log odds and when you use the value  $X_i + 1$  and you get a fitted value that fitted value talks about the log odds of the regressor for specific value of  $X_i + 1$ . So, now you can take anti logs and then you get the odds ratio back and we have discussed the case of odds ratio in the theoretical lecture. So, you have odds ratio defined as the odds of the number  $X_i + 1$  divided by odds of the number  $X_i$  and these ratio actually is given by exponential of the  $B_1$  slope coefficient.

So, now we are going to look at the case that we have. So, here, let us look at the case of the gre variable. So, let us see what coefficient value it has. So, it has a 0.002 as the coefficient value. So, we can say that for one unit change in continuous variables gre score, the log of

odds of admission into the graduate program versus non-admission in the program increases by 0.002 units.

So, now we are going to look at the gpa variable. So, let us have a look at the coefficient value and that is 0.8040. So, how can I interpret this number? So, for one unit change in the gpa score, the log of odds of admission versus non-admission increases by 0.804 units. Now, the rank variable is a bit tricky because you know, it is not a continuous variable.

So, it only changes by one unit. So, first, let us look at what are the rank variables we have used. So, here in the regression you see, I have actually use the dummy variables for the ranks and I have three dummy variables. Rank one is basically the base category. So, I have three dummy variables, rank two, rank three, and rank four. Each of them are actually indicating the level or rank of the institution.

So, rank two is a dummy variable, which takes value one, when the institution's rank is found to be two. And similarly, you can interpret rank three and rank four variables. So, now let us focus on the case of rank two. And if we can interpret the coefficient for a rank 2, then it can be followed for rank 3 and rank 4 variables also.

So, now note that the rank variable rank 2 has a coefficient of minus 0.675. So, what does that mean? That means attending an undergraduate Institute with rank 2 versus and undergraduate institution of rank 1 changes the log of odds ratio of admission by this many units, minus 0.675 units. So, you can see that as the student is graduating from a lower rank institution then the chance of getting admission into the graduate program is actually lower. And we get a magnitude of the log odds ratio by looking at the coefficient for this variable rank 2.



(Refer Slide Time: 39:20)

### McFadden R<sup>2</sup>

$\pi$

- In the case of logistic regression, we use Pseudo R<sup>2</sup> technique to evaluate the goodness-of-fit of our estimated model.
- By default R does not provide the McFadden's R<sup>2</sup>. In order to obtain it we have installed the "pscl" package.

```
> library(pscl)
> p02(logit)
Fitting null model for pseudo-r2
      lln      llnnull      G2      McFadden      r2M      r2CJ
-229.25874624 -249.98825878 41.45925588 0.08292194 0.08845782 0.11799588
> |
```

- You can also compute this as follows

```
> logit1<-glm(admit~1,family = binomial(link="logit"),DV )
> loglik(logit1)
[log lik.] -249.9883 (df=1)
> logit<-glm(admit~gre+gpa+rank,family = binomial(link="logit"),DV )
> loglik(logit)
[log lik.] -229.2587 (df=6)
> 1-loglik(logit)/loglik(logit1)
[log lik.] 0.08292194 (df=6)
> |
```

Now, in this slide, we are going to talk about the goodness-of-fit of our logistic regression model. So, if you remember, we have discussed the case of McFadden's R square in our theoretical discussion for logistic regression. So, here, I am going to show you how in R environment you can get the value of that Pseudo R squared. So, here note that, R does not provide the McFadden's R square automatically. So, for that again, you have to download and install a package and the name of the packages, pscl. So, I am showing you here, the code and the outcome. Once you run this pscl command to obtain the McFadden's R square.

So, here you see that you have this McFadden's R square number reported, and that is marked with this red box. And it is quite low, but as I told you in the theoretical discussion that generally in discrete choice models, you do not expect you a very high Pseudo R square value, 0.08. I do not know whether it is good or bad, but as the regression coefficients are all significant, significant, you can say that these variables have explained some degree of variation in the discrete Y variable.

Now, note that, if this pscl package is not available or you have some problem with these pscl package how can you get a measure for McFadden's R square. Now, if you jog your memory probably you will remember that we have spoken about that formula of McFadden's Pseudo R square. And that is basically a comparison of two log-likelihood numbers actually. And these two log-likelihood numbers are coming from two different models. One is basically the full model, and one is basically the intercept only model.

So, in the full model, you have all the explanatory variables, explaining the variation of Y and in the intercept only model you just do not have any explanatory variable in the model. So, basically you put exclusion restrictions on the coefficients for the explanatory variables. So, if you now run these two logit models then you are going to get two different log-likelihood values, and then you can actually compare them and then you can derive McFadden's R square by computing a few things in R environment. So, you do not need actually that pscl package to help you.

So, in the second box of this slide, I am showing you how you can actually compute the McFadden's R square. So, first you define a logit model and that you can call logit one. And that is basically an intercept only model and you get the log-likelihood value as minus 249.98, and the degrees of freedom is 1 because you have only the intercept in the regression model.

And now you run your full model where you bring in those three explanatory variables in the regression equation. And then if you run the model, then you actually have 6 degrees of freedom because you are estimating 6 parameter values. One intercept and 5 explanatory variables. Two of them are continuous in nature, gre and gpa. And then you have three dummy variables, rank 2, rank 3, and rank 4.

So, if you run that model, then you get a log-likelihood value minus 229.25. So, now you can actually make use of these two log-likelihood numbers and here, I am showing you the last command, which is actually computing the McFadden's R square formula's value for you. So, the software computer value 0.0829. And you note that you get back the exact same number in the first step when you actually implemented the pscl package to compute the McFadden's R square.

(Refer Slide Time: 43:47)

### Odds ratio

$\pi$

- In the case of logistic regression, estimated coefficients provides log odds ratio, which is very difficult to understand sometimes. So, we exponentiate those estimated coefficient and interpret them as odds ratio.

```
> exp(cbind(OR=coef(logit), confint(logit)))
Waiting for profiling to be done...
              OR      2.5 %    97.5 %
(Intercept) 0.0185001 0.001889165 0.1665354
gre          1.0022670 1.000127602 1.0044457
gpa          2.2345448 1.173858216 4.3238349
rank2       0.5089310 0.272289674 0.9448343
rank3       0.2617923 0.131641717 0.5115181
rank4       0.2119375 0.090715546 0.4706961
> |
```

- Now we can say that for a unit increase in GRE score odds of being admitted to graduate school (vs not being admitted) increase by a factor of 1.002

12

Now, in this slide, we're going to talk about odds ratio. Odds ratios play a big role in interpreting the logistic regression models and estimated coefficients actually provide us the log odds ratio. Right? So, it is somewhat difficult to understand. So, we exponentiate these estimated coefficients and then we get the odds ratio.

So, this is the way generally in empirical researchers discuss the interpretation of the coefficients in the context of logistic regression model. So, here I am showing you how you can get the odds ratio by taking the exponentiation of the estimated regression coefficients. The code is given and you see for each of my variables, continuous and discrete, now I have odds ratio.

So, if we concentrate on say the first variable in the list, the gre continuous variable, then we see the value of odds ratio is 1.0022. So, one can say that for an unit increase in the gre score, the odds of being admitted to a graduate school versus not being admitted in the graduate school increases by a factor of 1.002.

(Refer Slide Time: 45:10)

$\pi$

### Predicted probability

- Here we calculate the predicted probability of being admitted into graduate school for different ranking of institutions, holding other variables constant at their means. In order to create predicted probabilities we first create a new dataset with the values we want independent variables to take on to make prediction.

```
> # we create a new dataset holding gre & gpa constant at their mean
> SDV1 <- with(SDV, data.frame(gre = mean(gre), gpa = mean(gpa), rank = factor(1:4)))
> # Predicted probability from Logit
> SDV1P_logit <- predict(logit, newdata = SDV1, type = "response")
> SDV1
```

gre	gpa	rank	P_logit
1 587.7	3.3899	1	0.5160816
2 587.7	3.3899	2	0.3522846
3 587.7	3.3899	3	0.2186128
4 587.7	3.3899	4	0.1846684

- From above result we see that, predicted probability of being admitted into graduate school is 0.516 for students from the highest ranking institute (i.e. rank=1) and 0.184 for students from the lowest ranking institute (i.e. rank=4), holding GRE score & Grade point at their mean level.

13

In some empirical applications or projects, the scholars and the analysts may be interested to find out the exact probabilities associated with success and failure for a particular observational unit. So, here, if student is the observational unit and we are talking about success being measured by getting an admission in the graduate or not.

We maybe interested to actually calculate what probability a student has of getting admission into the graduate school. If we observe certain values for the covariates or the explanatory variables for that particular student. So, for that, we have to now compute the probability of success from the estimated regression equation. So, in the next slide, I am going to show you how we are going to compute such probabilities from logistic regression equation.

So, here you see, I am showing you the codes in the box. So, note that to create the predicted probabilities, we have to create a new dataset and that dataset should have the values we want independent variables to take on to make the prediction. So, here in this slide, I am going to show you how you can predict the probability of one fictional unit.

And for that fictional student, let me assume that I set my data at the mean. So, basically for the gre explanatory variable, the score is the mean gre score. For the gpa variable, I am assuming that the student has the mean gpa score. And then I am going to run this particular command to generate the predicted probabilities. And note that here, there is a problem because the rank variable is qualitative in nature, and we have defined different dummy variables for it in the logistics regression model.

So, now actually here, when you are predicting probability, you can actually use four different levels because when the numbers are limited, only 1, 2, 3, 4, four cases are there, it is not wise to calculate a mean, because mean of these, these grades 1, 2, 3, 4 does not make any sense. So, you compute the predicted probability for different values of rank. So, that is what we know we have done. And the result is seen in the box.

So, here you see that for the same student. Now, we know I have this different combination of variable values and the first row tells me that the mean gre score is 587.7 and mean gpa is 3.3899. And now we are assuming that the rank variable is taking value one. So, the estimated logistic regression actually now offers me an estimated probability value for these values of the explanatory variable. And the calculated number is 0.5166. So, similarly, you can now change the value for rank. From 1 to 2, 2 to 3, 3 to 4, and you can generate the probability scores.

So, we are done with our discussion on how to conduct a discrete variable data analysis in R environment. So, in the last lecture of the course, I am going to now talk about the time series data analysis. So, please join me for one more last time. See you then. Thank you.