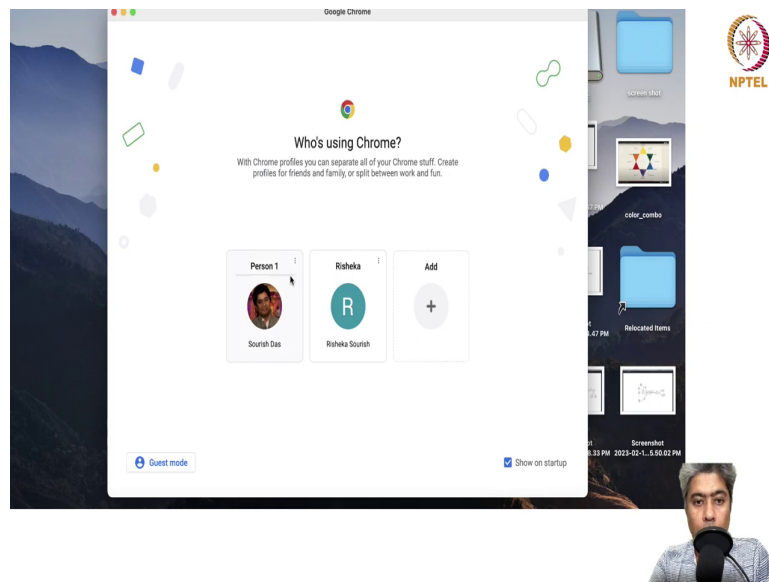


Predictive Analytics - Regression and Classification
Prof. Sourish Das
Department of Mathematics
Chennai Mathematical Institute

Lecture - 48
Hands on with R : Poisson Regression with Football Data

Hello, all. In this video, we are going to do some hands on with R.

(Refer Slide Time: 00:26)



So, first we are going to go to Internet.

(Refer Slide Time: 00:28)

The screenshot shows the Football-Data.co.uk website. The browser address bar displays 'football-data.co.uk/englandm.php'. The website features a navigation menu with links: Home, Free Bets, Livescores, Books on Betting, Casino, Poker, Tennis, Contact, and Like. A large banner for bet365 is visible, with a 'Join' button. Below the banner, there's a section titled 'Data Files: England' with a sub-header 'Last updated: 21/02/23'. This section includes a paragraph about registering with advertised bookmakers for free access to historical results and betting odds data files. It also mentions that CSV data files are available for download for use in spreadsheet applications like Excel. To the right of this section, there's a 'SITE RESOURCES' sidebar with links: Historical Data, Learn to Bet, Free Bets, Books, and Other Sites. Below this, there's a 'BETTING SYSTEMS' sidebar with links: Football Ratings, Wisdom of Crowds, and Contrarian Betting. On the left side, there's a 'WORLD'S FAVOURITE BOOKMAKER' section featuring bet365, and an 'OTHERS' section listing BoyleSports, William Hill, and Betfred. A small video inset in the bottom right corner shows a person speaking into a microphone.

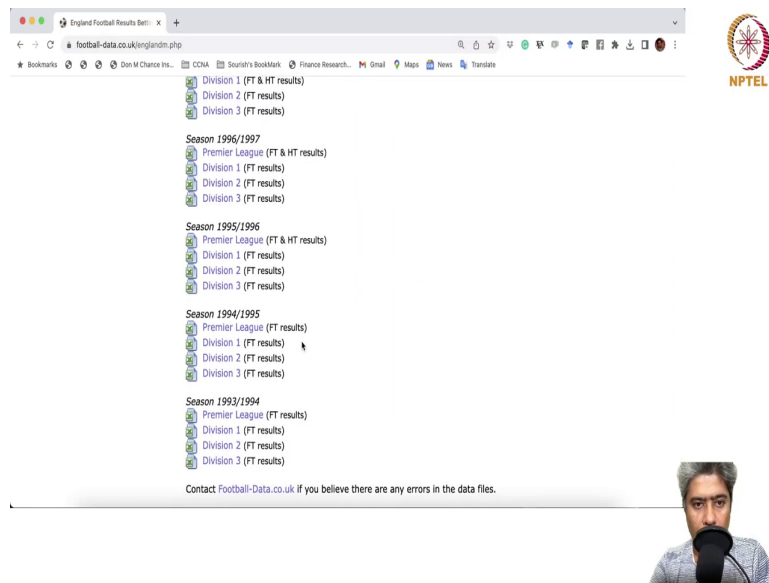
And, go to Google and we are going to choose this football data set football. I think I have it.
England football data set.

(Refer Slide Time: 01:00)

The screenshot displays the website football-data.co.uk/englandm.php. The page features a sidebar on the left with a 'bet365' logo and match details for 'England Premier Le...' including 'Everton v Aston Villa' and 'Leeds v Southampton'. The main content area lists match data for various seasons (2022/2023, 2021/2022, 2020/2021, 2019/2020) and leagues (Premier League, Championship, League 1, League 2, Conference). A right sidebar contains links for 'Contrarian Betting', 'Pinnacle Odds Drop', 'BETTING ARTICLES', 'BET CALCULATORS', and 'ODDS & RESULTS: MAIN LEAGUES'. An NPTEL logo is visible in the top right corner. A small video feed of a person is located in the bottom right corner of the browser window.

Yeah, this is the football data set and of English Premier League and then here is the you know English Premier all years of English Premier League data set from I think 1994.

(Refer Slide Time: 01:17)



You have English Premier League data sets for each year's Premier League data sets you will get. So, we are going to work with you can have lower division data sets also there championships League 1, League 2 conference. So, all these data sets are available from the English Football League.

(Refer Slide Time: 01:48)

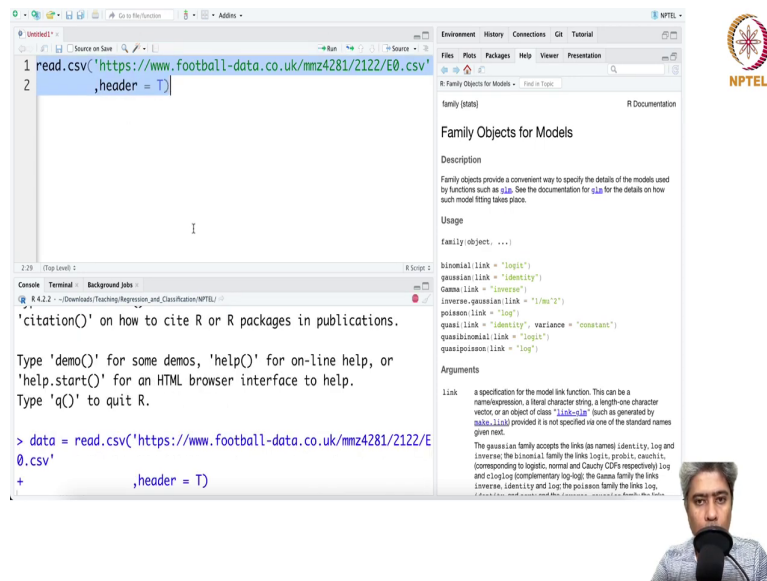
The screenshot shows the RStudio interface. The left pane displays file explorer with folders like Quarto Document..., Quarto Presentation..., Notebook, Markdown..., Shiny Web App..., Plumber API..., Text File, C++ Files, Python Script, R Script, Stan File, D3 Script, R Sweave, R HTML, and R Documentation... The top editor window contains R code:

```
.R (2022-10-31) -- "Innocent and Trusting"  
2022 The R Foundation for Statistical Computing  
ch64-apple-darwin20 (64-bit)  
  
tware comes with ABSOLUTELY NO WARRANTY.  
me redistribute it under certain conditions.  
) or 'licence()' for distribution details.
```

The bottom editor window shows the R documentation page for Family Objects for Models, which includes sections for Description, Usage, family(object,...), binomial link = logit(), gaussian link = identity(), gamma link = inverse(), poisson link = log(), quasi.link = constant(), quasibinomial link = logistic(), quasinegative link = log(), Arguments, Link, and Log. A small inset video of a person speaking is visible in the bottom right corner.

So, I am going to start my R going to start a R script and if I go there and maybe I will just take 21, 22. So, copy the link address.

(Refer Slide Time: 02:03)



The screenshot displays the RStudio interface. The script editor at the top contains the following R code:

```
1 read.csv('https://www.football-data.co.uk/mmz4281/2122/E0.csv')
2 ,header = T)
```

The console at the bottom shows the execution of the code and some help text:

```
> data = read.csv('https://www.football-data.co.uk/mmz4281/2122/E0.csv')
+                                     ,header = T)
```

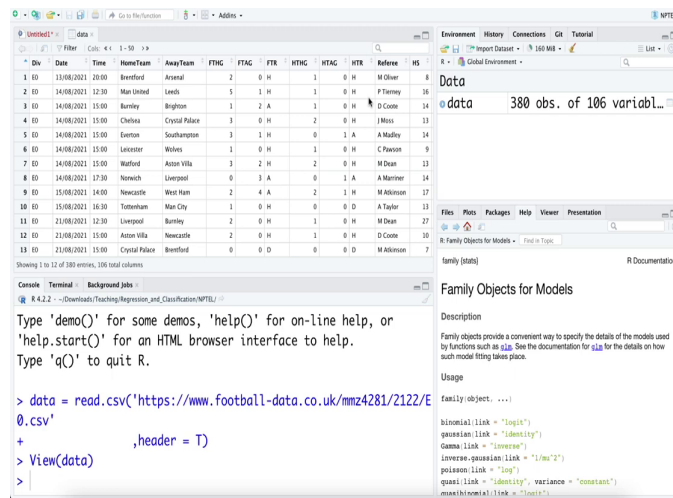
The help window on the right is titled 'Family Objects for Models' and provides information about the 'family' argument in the 'glm' function. It lists various families and their corresponding link functions:

- binomial: link = "logit"
- gaussian: link = "identity"
- gamma: link = "inverse"
- inverse.gaussian: link = "1/mu^2"
- poisson: link = "log"
- quasipoisson: link = "log"
- quasibinomial: link = "logit"
- quasipoisson: link = "log"

The NPTEL logo is visible in the top right corner of the slide.

And, which is a read dot csv give the path header equals to true and run this.

(Refer Slide Time: 02:21)



The screenshot displays the RStudio environment. The top-left pane shows a data table with columns: Div, Date, Time, HomeTeam, AwayTeam, FTHG, FTAG, FTR, HTNG, HTAG, HTR, Referee, and HS. The table contains 15 rows of match data. The top-right pane shows the 'Environment' tab with 'data' listed as a data object containing 380 observations of 106 variables. The bottom-left pane shows the console with the following R code and output:

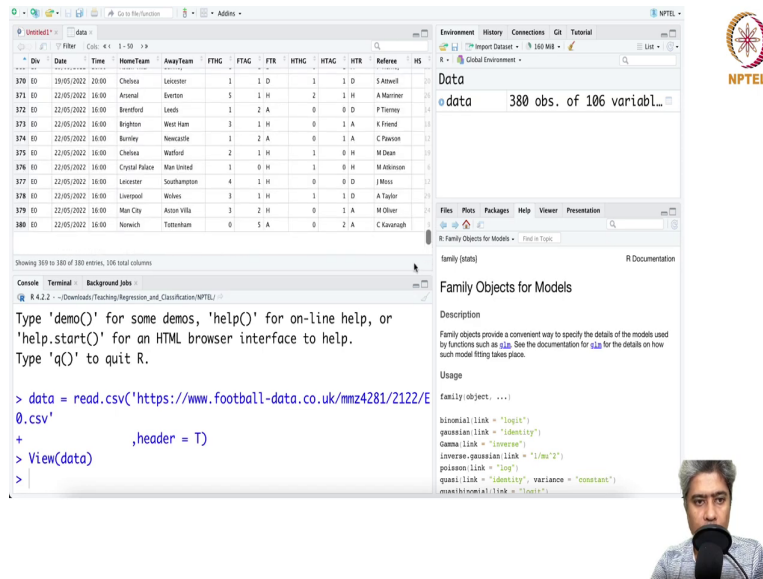
```
> data = read.csv('https://www.football-data.co.uk/mmz4281/2122/E0.csv')
+                                     ,header = T)
> View(data)
>
```

The bottom-right pane shows the 'Family Objects for Models' documentation, which describes how to specify model details using family objects.



And, the data just downloaded it just downloaded the data from the internet and it has 380 observations with 106 variables. So, there are thing is observation about 380 matches.



(Refer Slide Time: 02:42)



The screenshot displays the RStudio environment. The main window shows a data table with columns: Div, Date, Time, HomeTeam, AwayTeam, FTHG, FTAG, FTR, HTAG, HTR, Referee, and HS. The table lists 380 matches from 19/05/2022 to 22/05/2022, involving teams like Chelsea, Arsenal, Liverpool, and Manchester United. The console on the left shows the following R code and output:

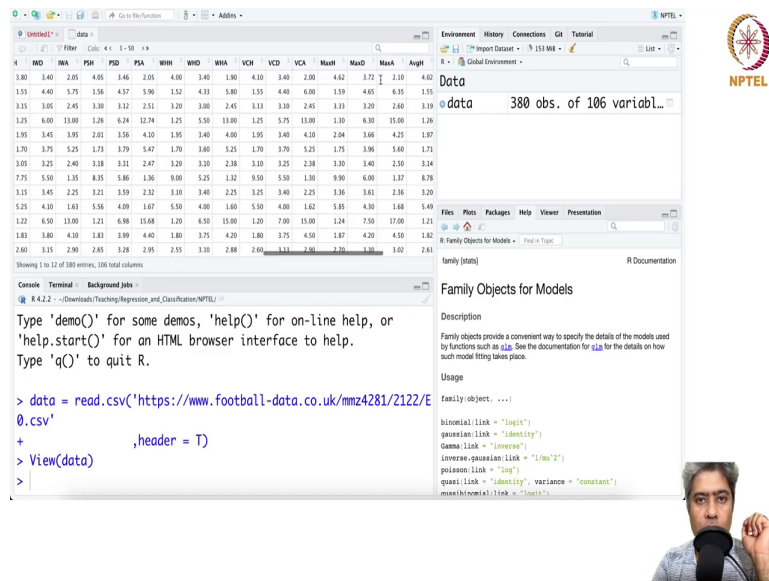
```
> data = read.csv('https://www.football-data.co.uk/mmz4281/2122/E0.csv', header = T)
> View(data)
```

The right-hand pane shows the 'Data' tab with the message 'data 380 obs. of 106 variabl...' and the 'Family Objects for Models' section, which includes a description and usage of family objects.



There are observations about 380 matches, ok, and then their divisions and everything is there. Now, it is about home time HomeTeam and AwayTeam FTHG stands for how many goal were scored by the HomeTeam, how many goal were scored FTAG stands for how many goal was scored by the AwayTeam.

(Refer Slide Time: 03:17)



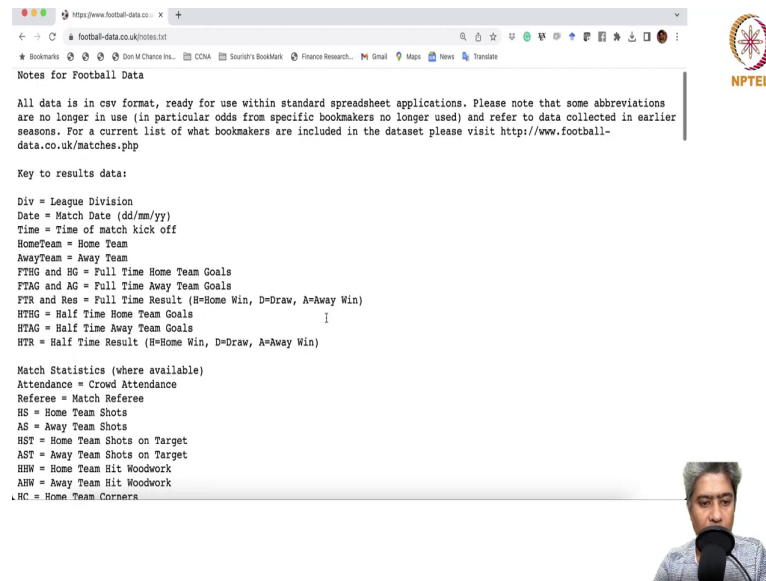
The screenshot displays the RStudio IDE. The top-left pane shows a data frame 'data' with 380 observations and 106 variables. The top-right pane shows the 'Environment' tab with 'data' listed. The bottom-left pane shows the console with the following code and output:

```
> data = read.csv('https://www.football-data.co.uk/mmz4281/2122/E0.csv', header = T)
> View(data)
>
```

The bottom-right pane shows the 'Family Objects for Models' documentation, which includes a description and usage examples for various link functions.

Then there are other observations and then there are different baiting ratios by different baiting houses and lot of other things are available.

(Refer Slide Time: 03:39)



Notes for Football Data

All data is in csv format, ready for use within standard spreadsheet applications. Please note that some abbreviations are no longer in use (in particular odds from specific bookmakers no longer used) and refer to data collected in earlier seasons. For a current list of what bookmakers are included in the dataset please visit <http://www.football-data.co.uk/matches.php>

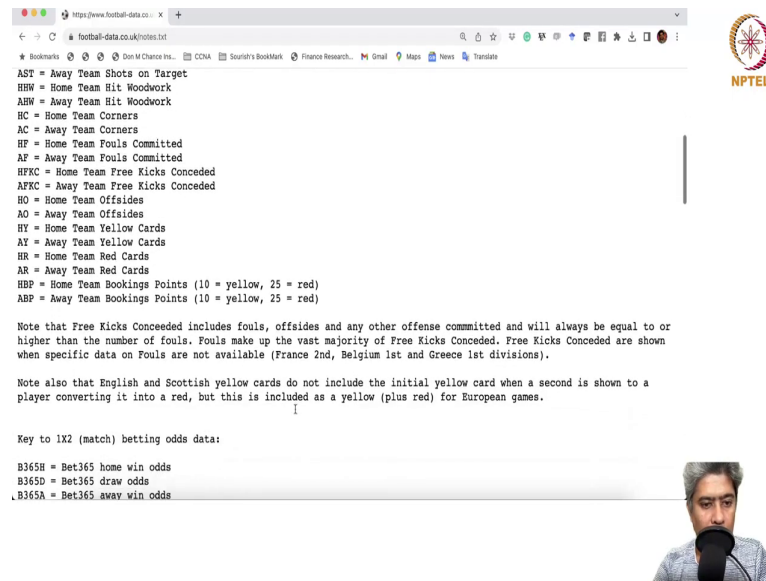
Key to results data:

Div = League Division
Date = Match Date (dd/mm/yy)
Time = Time of match kick off
HomeTeam = Home Team
AwayTeam = Away Team
FTHG and HG = Full Time Home Team Goals
FTAG and AG = Full Time Away Team Goals
FTR and Res = Full Time Result (H=Home Win, D=Draw, A=Away Win)
HTHG = Half Time Home Team Goals
HTAG = Half Time Away Team Goals
HTR = Half Time Result (H=Home Win, D=Draw, A=Away Win)

Match Statistics (where available)
Attendance = Crowd Attendance
Referee = Match Referee
HS = Home Team Shots
AS = Away Team Shots
HST = Home Team Shots on Target
AST = Away Team Shots on Target
HHW = Home Team Hit Woodwork
AHW = Away Team Hit Woodwork
HC = Home Team Corners

And, if you want to know each of those variables so, here is a note they have provided. So, FTHG and HG is Full Time Home Team Goal, FTR and Result is Full Time Result; H means Home Team has Won, D means Draw, A means AwayTeam.

(Refer Slide Time: 03:55)



The screenshot shows a web browser window with the address bar displaying "https://www.football-data.co.uk/notes.txt". The page content lists various football statistics abbreviations and their meanings:

- AST = Away Team Shots on Target
- HHW = Home Team Hit Woodwork
- AHW = Away Team Hit Woodwork
- RC = Home Team Corners
- AC = Away Team Corners
- HF = Home Team Fouls Committed
- AF = Away Team Fouls Committed
- HFKC = Home Team Free Kicks Conceded
- AFKC = Away Team Free Kicks Conceded
- HO = Home Team Offsides
- AO = Away Team Offsides
- HY = Home Team Yellow Cards
- AY = Away Team Yellow Cards
- HR = Home Team Red Cards
- AR = Away Team Red Cards
- HBP = Home Team Bookings Points (10 = yellow, 25 = red)
- ABP = Away Team Bookings Points (10 = yellow, 25 = red)

Below the list, there are two explanatory notes:

Note that Free Kicks Conceded includes fouls, offsides and any other offense committed and will always be equal to or higher than the number of fouls. Fouls make up the vast majority of Free Kicks Conceded. Free Kicks Conceded are shown when specific data on Fouls are not available (France 2nd, Belgium 1st and Greece 1st divisions).

Note also that English and Scottish yellow cards do not include the initial yellow card when a second is shown to a player converting it into a red, but this is included as a yellow (plus red) for European games.

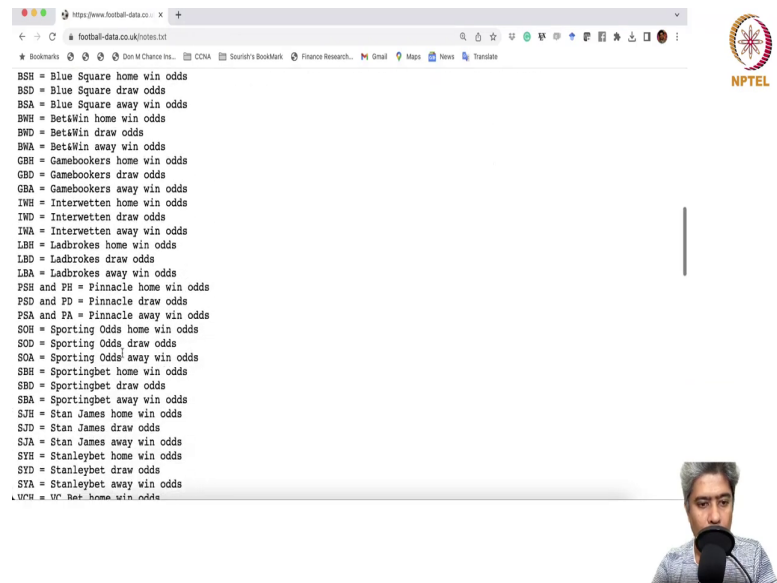
Key to 1X2 (match) betting odds data:

- B365H = Bet365 home win odds
- B365D = Bet365 draw odds
- B365A = Bet365 away win odds

In the bottom right corner, there is a video feed of a man with grey hair, wearing a blue shirt, speaking into a black microphone.

Then HST means Home Team how many Shots on Target; AST - Away Teams how many Shots are on Target; Home HHW means Home Team Hit Woodwork Home Team Shots Home Away Team Shots all these things how many Offsides, Yellow Card, Red Cards, Home Team Booking Points 10 for yellow cards, 25 for red cards.

(Refer Slide Time: 04:28)



The screenshot shows a web browser window with the address bar displaying "https://www.football-data.co.uk/notes.txt". The page content is a list of abbreviations and their corresponding betting odds, such as "BSH = Blue Square home win odds", "BSD = Blue Square draw odds", "BSA = Blue Square away win odds", "BWH = BetWin home win odds", "BWD = BetWin draw odds", "BWA = BetWin away win odds", "GBH = Gamebookers home win odds", "GBD = Gamebookers draw odds", "GBA = Gamebookers away win odds", "IWH = Interwetten home win odds", "IWD = Interwetten draw odds", "IWA = Interwetten away win odds", "LBH = Ladbrokes home win odds", "LBD = Ladbrokes draw odds", "LBA = Ladbrokes away win odds", "PSH and PS = Pinnacle home win odds", "PSD and PD = Pinnacle draw odds", "PSA and PA = Pinnacle away win odds", "SOH = Sporting Odds home win odds", "SOD = Sporting Odds draw odds", "SOA = Sporting Odds away win odds", "SBH = Sportingbet home win odds", "SBD = Sportingbet draw odds", "SBA = Sportingbet away win odds", "SJH = Stan James home win odds", "SJD = Stan James draw odds", "SJA = Stan James away win odds", "SYH = Stanleybet home win odds", "SYD = Stanleybet draw odds", "SYA = Stanleybet away win odds", and "UWH = Uff... home win odds".

In the bottom right corner, there is a small video inset showing a man speaking into a microphone. To the right of the browser window, there is a logo for NPTEL (National Programme on Technology Enhanced Learning) featuring a stylized flower or star shape.

So, all these things are already there.

(Refer Slide Time: 04:39)

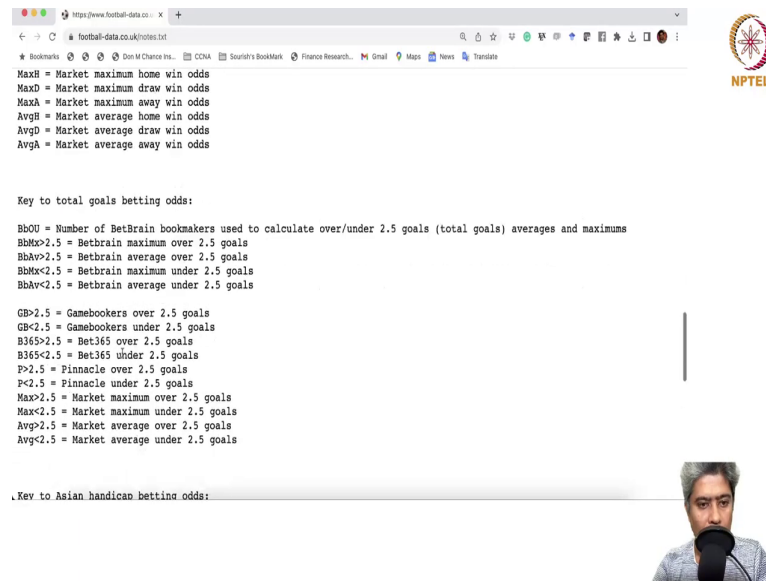
https://www.football-data.co.uk/notes.txt

LBH = Ladbrokes home win odds
LBD = Ladbrokes draw odds
LBA = Ladbrokes away win odds
PSH and PH = Pinnacle home win odds
PSD and PD = Pinnacle draw odds
PSA and PA = Pinnacle away win odds
SOH = Sporting Odds home win odds
SOD = Sporting Odds draw odds
SOA = Sporting Odds away win odds
SBH = Sportingbet home win odds
SBD = Sportingbet draw odds
SBA = Sportingbet away win odds
SJH = Stan James home win odds
SJD = Stan James draw odds
SJA = Stan James away win odds
SYH = Stanleybet home win odds
SYD = Stanleybet draw odds
SYA = Stanleybet away win odds
VCH = VC Bet home win odds
VCD = VC Bet draw odds
VCA = VC Bet away win odds
WHH = William Hill home win odds
WHD = William Hill draw odds
WHA = William Hill away win odds

BbIX2 = Number of BetBrain bookmakers used to calculate match odds averages and maximums
BbMxH = Betbrain maximum home win odds
BbAvH = Betbrain average home win odds
BbMxD = Betbrain maximum draw odds
BbAvD = Betbrain average draw win odds
BbMxA = Betbrain maximum away win odds



(Refer Slide Time: 04:41)



The screenshot shows a web browser window with the address bar displaying "https://www.football-data.co.uk/notes.txt". The page content lists various betting odds abbreviations and their meanings:

- MaxH = Market maximum home win odds
- MaxD = Market maximum draw win odds
- MaxA = Market maximum away win odds
- AvgH = Market average home win odds
- AvgD = Market average draw win odds
- AvgA = Market average away win odds

Key to total goals betting odds:

- BbOU = Number of BetBrain bookmakers used to calculate over/under 2.5 goals (total goals) averages and maximums
- BbOx>2.5 = Betbrain maximum over 2.5 goals
- BbAv>2.5 = Betbrain average over 2.5 goals
- BbOx<2.5 = Betbrain maximum under 2.5 goals
- BbAv<2.5 = Betbrain average under 2.5 goals

GB>2.5 = Gamebookers over 2.5 goals

GB<2.5 = Gamebookers under 2.5 goals

B365>2.5 = Bet365 over 2.5 goals

B365<2.5 = Bet365 under 2.5 goals

P>2.5 = Pinnacle over 2.5 goals

P<2.5 = Pinnacle under 2.5 goals

Max>2.5 = Market maximum over 2.5 goals

Max<2.5 = Market maximum under 2.5 goals

Avg>2.5 = Market average over 2.5 goals

Avg<2.5 = Market average under 2.5 goals

Key to Asian handicap betting odds:

In the bottom right corner, there is a small video feed showing a man speaking into a microphone.

And, then in addition there are some other where you saw there like odds different odds are also there winning odds and you know.

(Refer Slide Time: 04:45)

https://www.football-data.co.uk/notes.txt



Bookmarks Don M Chance Ins... CNNA Sports's BookMark Finance Research... Mail Maps News Translate

Avg<2.5 = Market average under 2.5 goals

Key to Asian handicap betting odds:

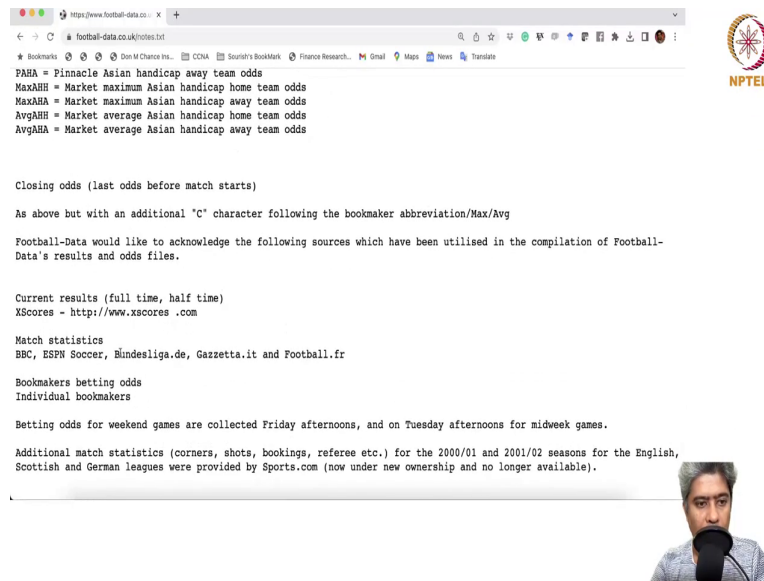
BbAH = Number of BetBrain bookmakers used to Asian handicap averages and maximums
BbAHH = Betbrain size of handicap (home team)
AHH = Market size of handicap (home team) (since 2019/2020)
BbMxAHH = Betbrain maximum Asian handicap home team odds
BbAvAHH = Betbrain average Asian handicap home team odds
BbMxAHA = Betbrain maximum Asian handicap away team odds
BbAvAHA = Betbrain average Asian handicap away team odds

GBAHH = Gamebookers Asian handicap home team odds
GBAHA = Gamebookers Asian handicap away team odds
GBAHH = Gamebookers size of handicap (home team)
LBAHH = Ladbrokes Asian handicap home team odds
LBAHA = Ladbrokes Asian handicap away team odds
LBAHH = Ladbrokes size of handicap (home team)
B365AHH = Bet365 Asian handicap home team odds
B365AHA = Bet365 Asian handicap away team odds
B365AHH = Bet365 size of handicap (home team)
PAHH = Pinnacle Asian handicap home team odds
PAHA = Pinnacle Asian handicap away team odds
MaxAHH = Market maximum Asian handicap home team odds
MaxAHA = Market maximum Asian handicap away team odds
AvgAHH = Market average Asian handicap home team odds
AvgAHA = Market average Asian handicap away team odds



All these odds are there.

(Refer Slide Time: 04:46)



https://www.football-data.co.uk/notes.txt

PAHA = Pinnacle Asian handicap away team odds
MaxHH = Market maximum Asian handicap home team odds
MaxAHA = Market maximum Asian handicap away team odds
AvgHH = Market average Asian handicap home team odds
AvgAHA = Market average Asian handicap away team odds

Closing odds (last odds before match starts)

As above but with an additional "C" character following the bookmaker abbreviation/Max/Avg

Football-Data would like to acknowledge the following sources which have been utilised in the compilation of Football-Data's results and odds files.



Current results (full time, half time)
XScores - <http://www.xscores.com>

Match statistics
BBC, ESPN Soccer, Bundesliga.de, Gazzetta.it and Football.fr

Bookmakers betting odds
Individual bookmakers

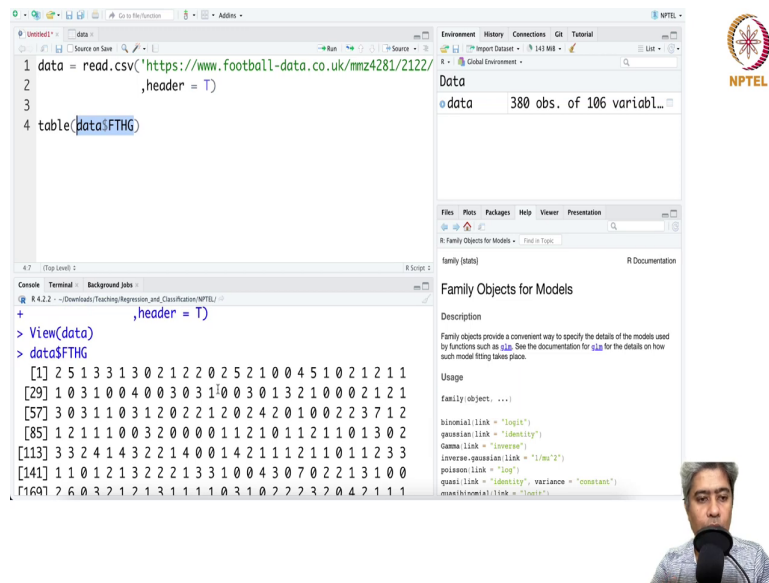
Betting odds for weekend games are collected Friday afternoons, and on Tuesday afternoons for midweek games.

Additional match statistics (corners, shots, bookings, referee etc.) for the 2000/01 and 2001/02 seasons for the English, Scottish and German leagues were provided by Sports.com (now under new ownership and no longer available).



Many many you know betting houses gives odds before the win team wins or not before the team match starts and these odds sometimes may have some predictive power for home team.

(Refer Slide Time: 05:01)



The screenshot displays the RStudio environment. The script editor on the left contains the following R code:

```
1 data = read.csv('https://www.football-data.co.uk/mmz4281/2122/
2 ,header = T)
3
4 table(data$FTHG)
```

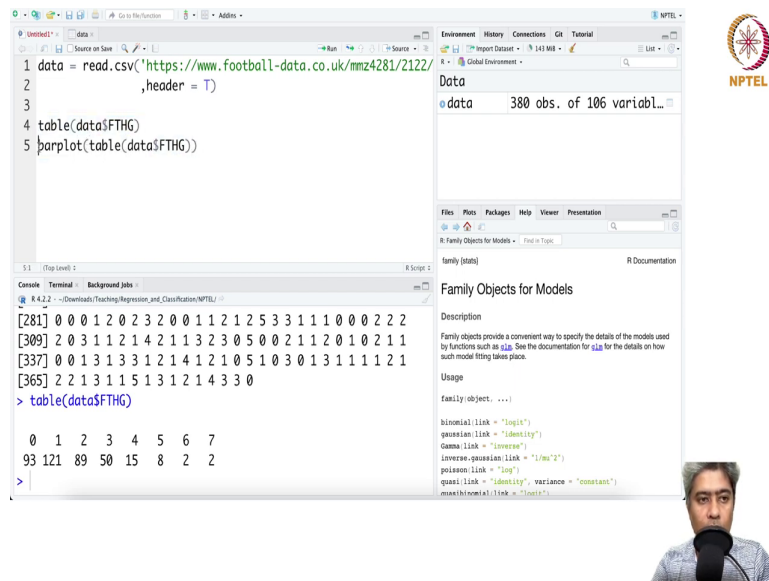
The console on the bottom left shows the execution of the code, with the output of the `table(data$FTHG)` command displayed as a frequency table of goals scored by the home team. The output is as follows:

```
> View(data)
> data$FTHG
[1] 2 5 1 3 3 1 3 0 2 1 2 2 0 2 5 2 1 0 0 4 5 1 0 2 1 2 1 1
[29] 1 0 3 1 0 0 4 0 0 3 0 3 1 0 0 3 0 1 3 2 1 0 0 0 2 1 2 1
[57] 3 0 3 1 1 0 3 1 2 0 2 2 1 2 0 2 4 2 0 1 0 0 2 2 3 7 1 2
[85] 1 2 1 1 1 0 0 3 2 0 0 0 0 1 1 2 1 0 1 1 2 1 1 0 1 3 0 2
[113] 3 3 2 4 1 4 3 2 2 1 4 0 0 1 4 2 1 1 1 2 1 1 0 1 1 2 3 3
[141] 1 1 0 1 2 1 3 2 2 2 1 3 3 1 0 0 4 3 0 7 0 2 2 1 3 1 0 0
[169] 2 6 0 3 2 1 2 1 3 1 1 1 1 0 3 1 0 2 2 2 3 2 0 4 2 1 1 1
```

The Environment pane on the right shows the `data` object as a data frame with 380 observations and 106 variables. The R Documentation pane on the right shows the 'Family Objects for Models' section.

So, I will do some analysis, ok. So, let us see data dollar. So, maybe I will just say FTHG, this one I will do table. So, how many goals are scored by? So, this is FTHG number of goals scored by the home team is essentially see it is a count variable, right.

(Refer Slide Time: 05:34)




The screenshot shows an RStudio interface. The script editor on the left contains the following code:


```
1 data = read.csv('https://www.football-data.co.uk/mmz4281/2122/
2               ', header = T)
3
4 table(data$FTHG)
5 barplot(table(data$FTHG))
```

The console on the bottom left shows the output of the `table(data$FTHG)` command:

```
[281] 0 0 1 2 0 2 3 2 0 0 1 1 2 1 2 5 3 3 1 1 1 0 0 0 2 2 2
[309] 2 0 3 1 1 2 1 4 2 1 1 3 2 3 0 5 0 0 2 1 1 2 0 1 0 2 1 1
[337] 0 0 1 3 1 3 3 1 2 1 4 1 2 1 0 5 1 0 3 0 1 3 1 1 1 1 2 1
[365] 2 2 1 3 1 1 5 1 3 1 2 1 4 3 3 0
> table(data$FTHG)
 0  1  2  3  4  5  6  7
93 121 89 50 15  8  2  2
```

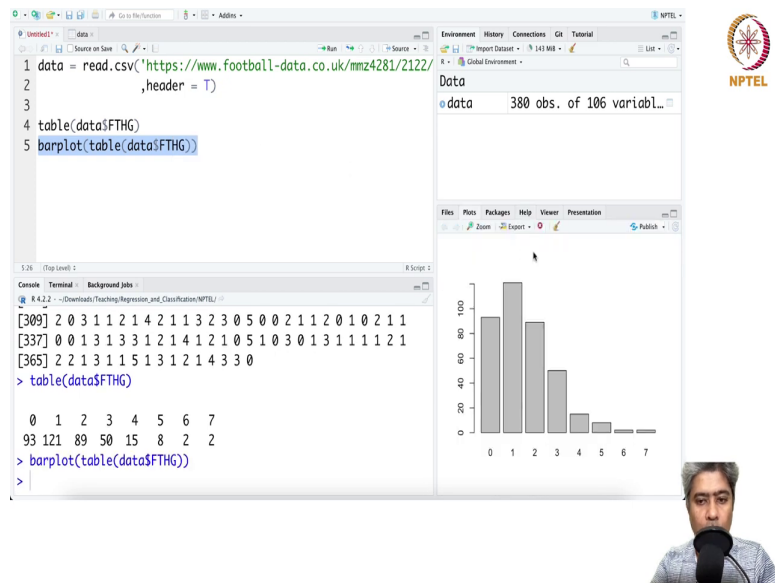
The environment pane on the right shows a variable `data` with 380 observations and 106 variables. The R Documentation pane on the right shows the 'Family Objects for Models' section.



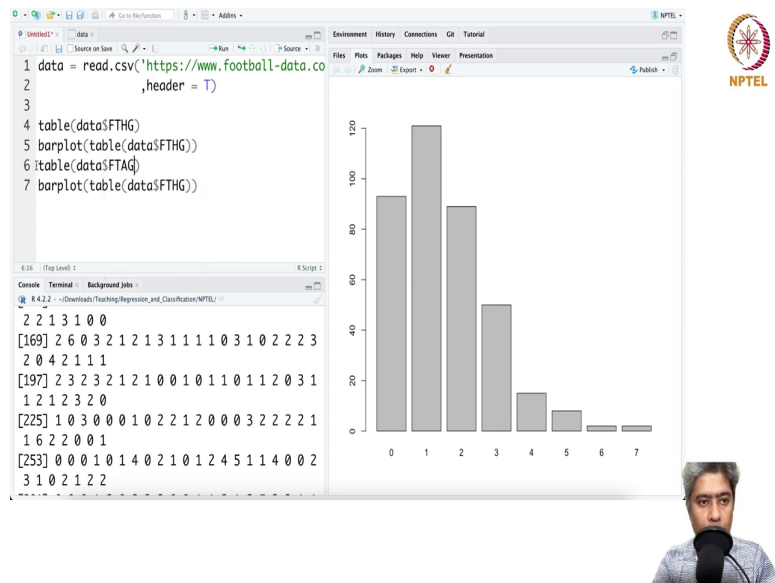


So, if you just do a table ok then there are out of 380 matches there are 93 matches where home team did not score single goal. There are 121 matches where home team scored 1 goal; there 89 matches there are home teams scored 2 goals; there are 50 matches that home team scored 3 goals. So, like this you can do a bar plot of this you can do a bar plot ok. So, this is a nice bar plot, ok.

(Refer Slide Time: 06:09)

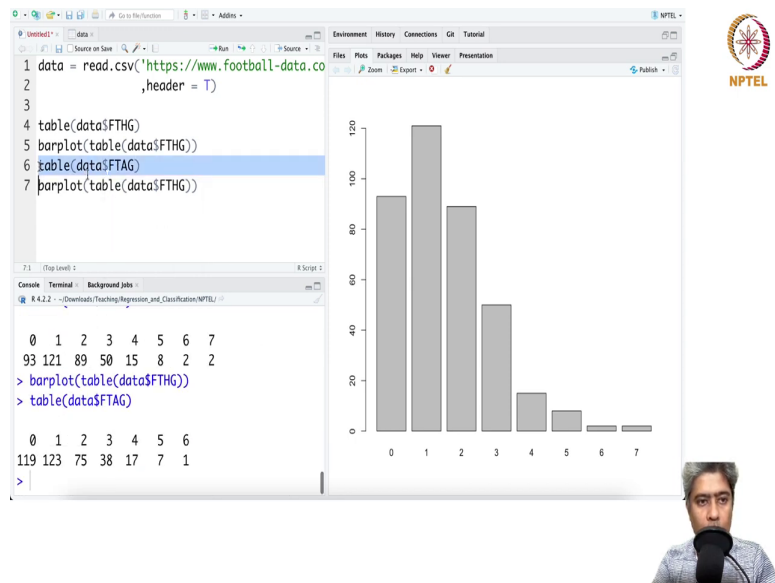


(Refer Slide Time: 06:12)



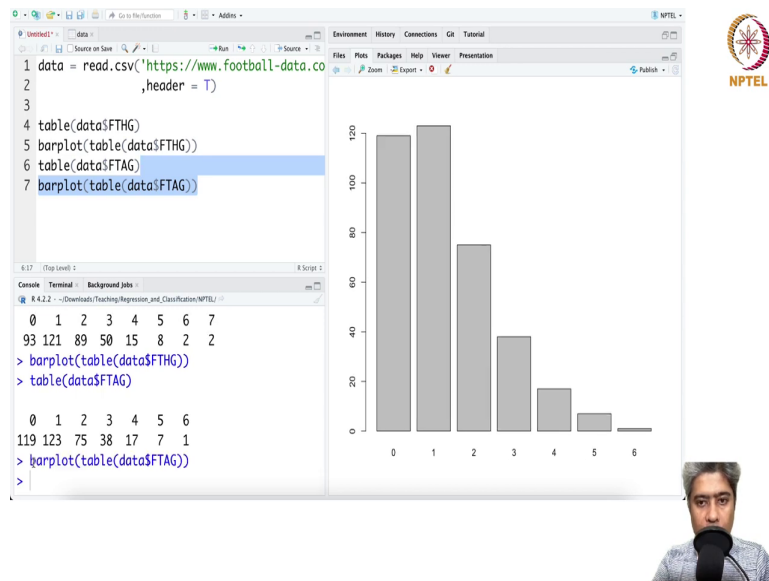
And, then you can do similarly instead of home team you can do a away team, ok.

(Refer Slide Time: 06:28)



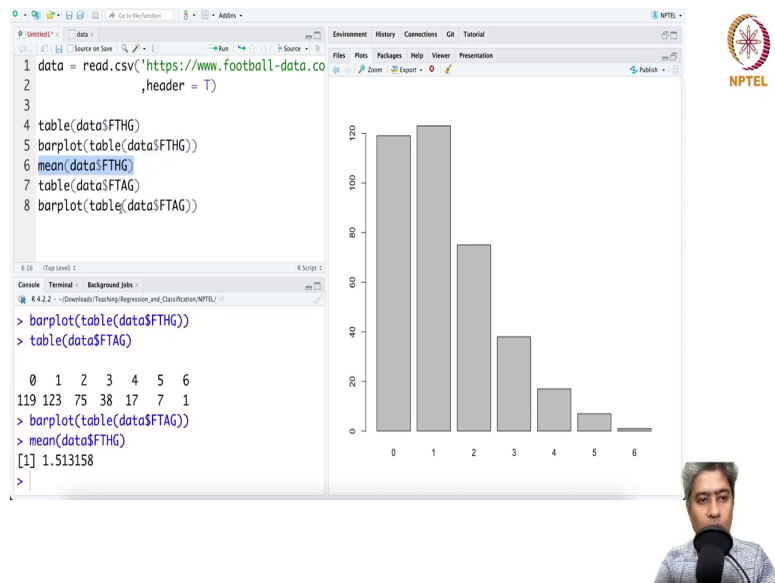
And, can you plot we can do a bar plot and you can see the away team's distribution is slightly different. Away team's distributions is slightly different.

(Refer Slide Time: 06:46)



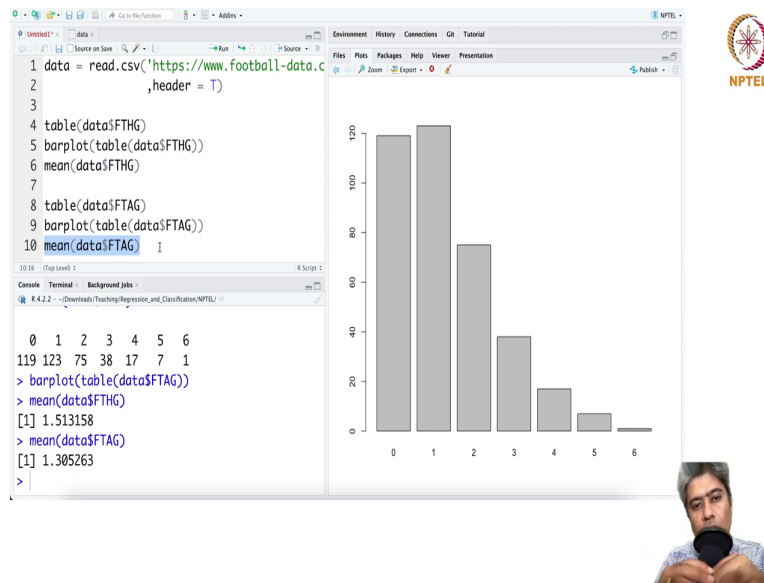
So, away teams cannot goal score any goal in 119 matches, 123 matches they go score exactly one goal; there are 75 matches where they score 20 89 goal; 38 matches they score 3 goals. So, it is like looks like in away teams scoreless number of goals in on an average this score less number of goals in away matches a team.

(Refer Slide Time: 07:24)



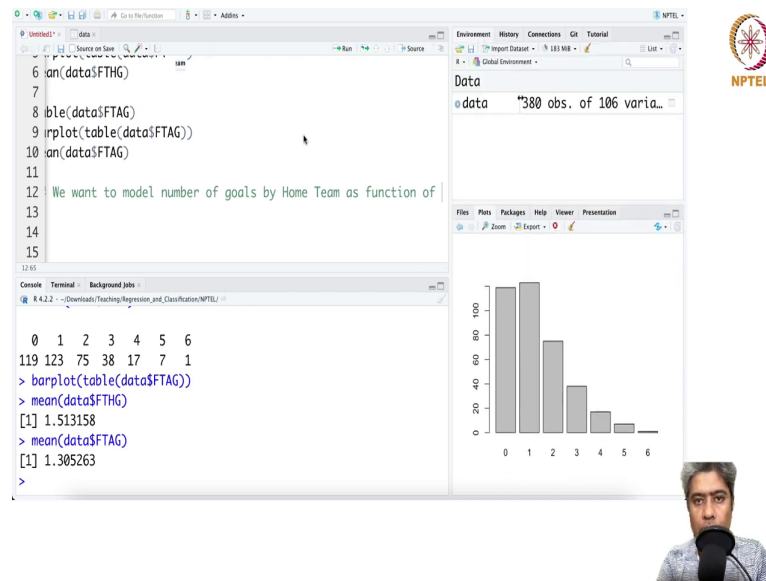
So, we can take mean of the home team score ok.

(Refer Slide Time: 07:42)



So, 1.51 and we can take the mean of the home team score on an average 1.5 many goals whereas, away teams score looks like maybe less little 1.3. So, this is our typical basic summary.

(Refer Slide Time: 08:02)

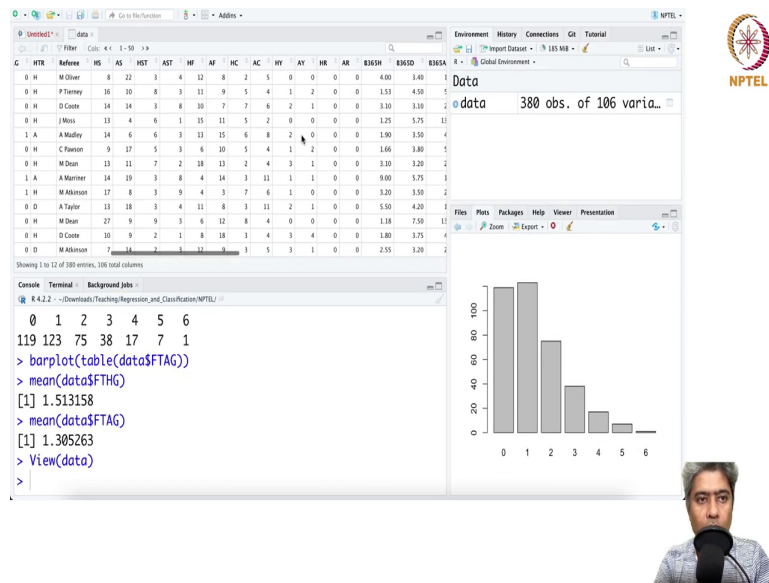


Now, we want to if you want to model, we want to model the number of goals we want to model if you want we want to model number of goals ok. So, how do you want to model number of goals? So, goals by say home team suppose ok as a function of; as function of odds ratio say bait 365 odd ratio say there are there were some, ok.

(Refer Slide Time: 08:40)

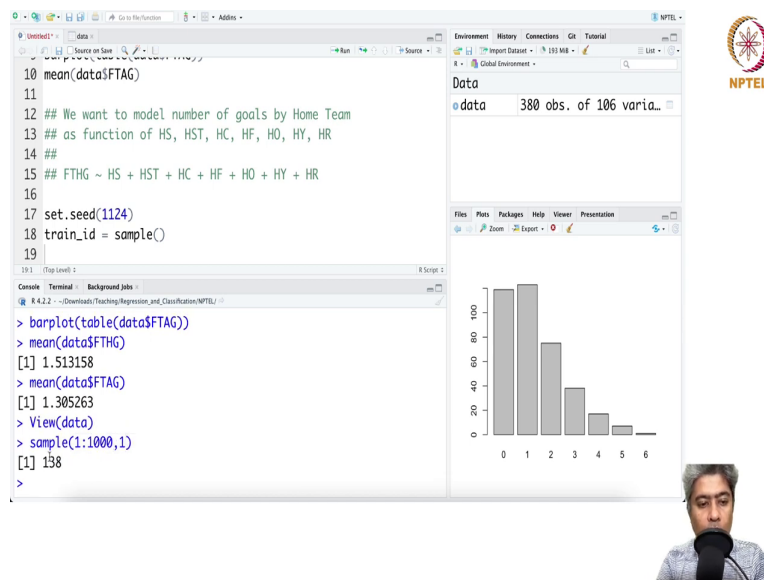


(Refer Slide Time: 08:41)



If you go out or maybe HS, HS stands for shots or HST. I think if you go there HS or HST, HS is the home team shots and home team shots on target.

(Refer Slide Time: 09:01)



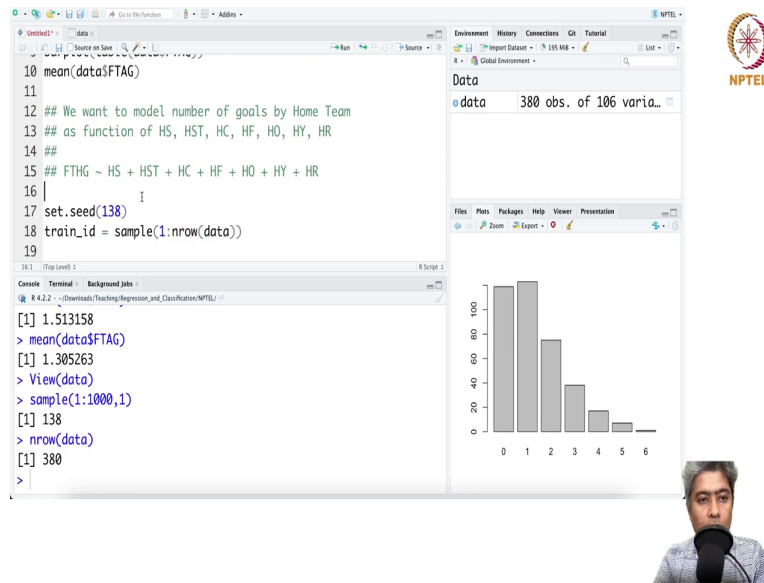
So, we will as a function of HST home team shot on the target and home team corners HC, ok and home team fouls committed HF. These are the functions I clear and home team offside HO. So, these are the functions these are the thing and home team yellow card and home team red card, ok.

So, if it says as a function of home team yellow card and home team red card. So, we want to function we want to model FTHG. So, FT definitely we want to model FTHG as a function of all these variable home team shots, home team shots on target, home team how many corners they get, HF is home team how many fouls they committed, how many off sides, they did home team how many yellow card they found and how many red card they saw, ok.

So, what I am going to do, I am going to say split the data let us split the data into train and test ok. So, train sorry train id will be say sample ok I need to before sampling I need to set up

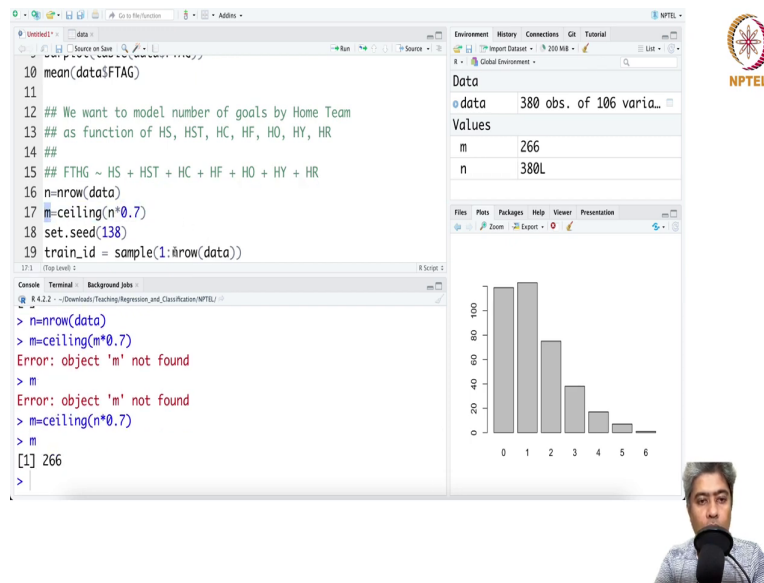
a seed dot seed same 1124 or I will just do a sampling some instead of some 1 is to 1000 comma 138, ok.

(Refer Slide Time: 11:22)



I will take this number as my seed. If you use the same seed, we will get the exact same result that I am getting. So, 1 H 2 ok, 1 H 2 nrow of data. So, nrow of data is 380 and I think what I am going to do is nrow of data and m is ceiling of m times 0.7. So, m is ok first I have to sorry about that m is 266.

(Refer Slide Time: 11:41)



The image shows a screenshot of the RStudio interface. The script editor on the left contains the following R code:

```
10 mean(data$FTAG)
11
12 ## We want to model number of goals by Home Team
13 ## as function of HS, HST, HC, HF, HO, HY, HR
14 ##
15 ## FTHG ~ HS + HST + HC + HF + HO + HY + HR
16 n=nrow(data)
17 m=ceiling(n*0.7)
18 set.seed(138)
19 train_id = sample(1:nrow(data))
```



The console on the bottom left shows the execution of the code, with errors for the `m` variable:

```
> n=nrow(data)
> m=ceiling(m*0.7)
Error: object 'm' not found
> m
Error: object 'm' not found
> m=ceiling(n*0.7)
> m
[1] 266
>
```

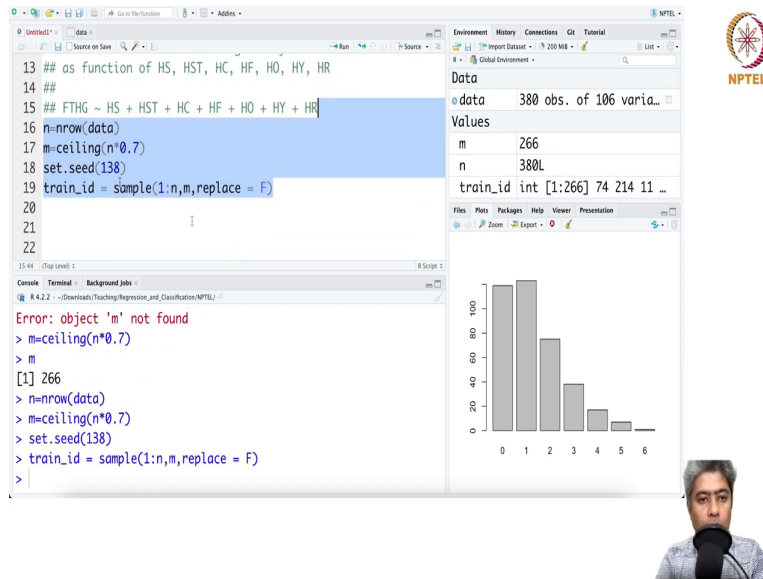
The data viewer on the right shows the 'data' object with 380 observations and 106 variables. The 'Values' section shows the distribution of the 'm' variable:

Values	m	n
	266	380L

A histogram on the right shows the distribution of the 'm' variable, with the x-axis ranging from 0 to 6 and the y-axis ranging from 0 to 100. The distribution is skewed to the right, with a peak at 0.



(Refer Slide Time: 12:12)



The image shows the RStudio interface. The script editor on the left contains the following code:

```
13 ## as function of HS, HST, HC, HF, HO, HY, HR
14 ##
15 ## FTHG ~ HS + HST + HC + HF + HO + HY + HR
16 n=nrow(data)
17 m=ceiling(n*0.7)
18 set.seed(138)
19 train_id = sample(1:n,m,replace = F)
20
21
22
```



The console on the bottom left shows the execution of the code, with an error message:

```
Error: object 'm' not found
> m=ceiling(n*0.7)
> m
[1] 266
> n=nrow(data)
> m=ceiling(n*0.7)
> set.seed(138)
> train_id = sample(1:n,m,replace = F)
>
```

The environment pane on the right shows the following variables:

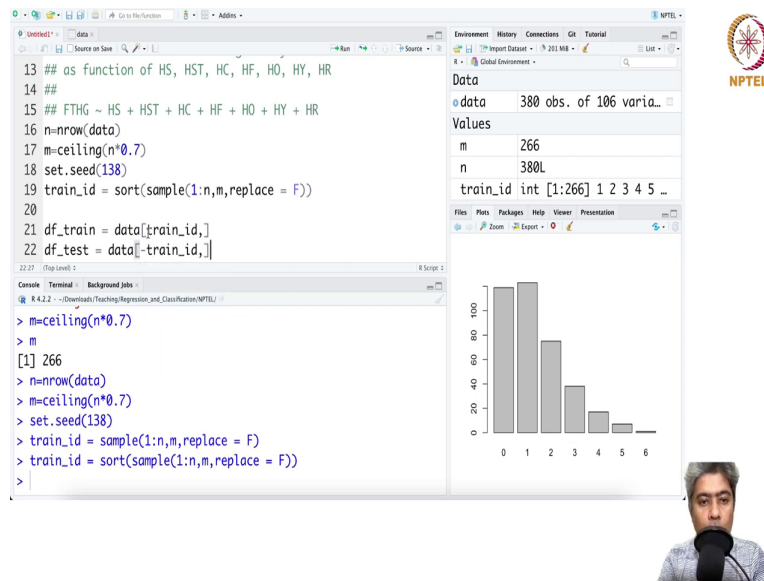
Variable	Value
m	266
n	380L
train_id	int [1:266] 74 214 11 ...

A histogram is displayed on the right side of the environment pane, showing the distribution of the 'train_id' variable. The x-axis represents the 'train_id' values (0 to 6), and the y-axis represents the frequency (0 to 100). The distribution is skewed to the right, with the highest frequency at 0.



So, what I am going to do is n comma m replace equal to false. I do not want to replace this, alright.

(Refer Slide Time: 12:26)



The image shows the RStudio interface. The script editor on the left contains the following code:

```
13 ## as function of HS, HST, HC, HF, HO, HY, HR
14 ##
15 ## FTHG ~ HS + HST + HC + HF + HO + HY + HR
16 n=nrow(data)
17 m=ceiling(n*0.7)
18 set.seed(138)
19 train_id = sort(sample(1:n,m,replace = F))
20
21 df_train = data[train_id,]
22 df_test = data[-train_id,]
```



The console on the bottom left shows the execution of the code:

```
> m=ceiling(n*0.7)
> m
[1] 266
> n=nrow(data)
> m=ceiling(n*0.7)
> set.seed(138)
> train_id = sample(1:n,m,replace = F)
> train_id = sort(sample(1:n,m,replace = F))
>
```

The environment pane on the right shows the data object with 380 observations and 106 variables. The values pane shows the distribution of the train_id variable:

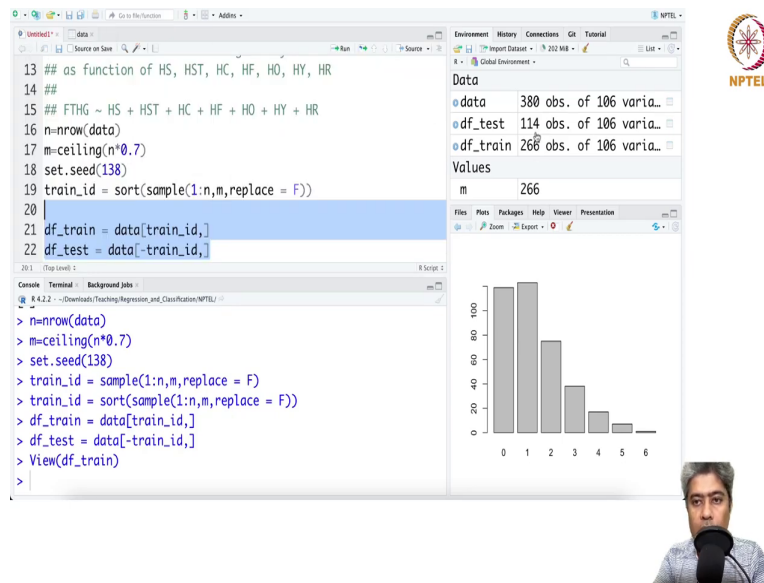
train_id	int
1	266
2	1
3	2
4	3
5	4

A histogram is displayed on the right, showing the frequency of train_id values. The x-axis represents the train_id values (0 to 6), and the y-axis represents the frequency (0 to 100).



Now, let me write this ok sort I would like to sort it. So, one time sort, so, it is not a problem and it is a small data to be honest and then I have df train equal to data train id comma and df test equal to data minus train id comma let me have a run.

(Refer Slide Time: 13:03)



The image shows an RStudio session. The source editor contains the following R code:

```
13 ## as function of HS, HST, HC, HF, HO, HY, HR
14 ##
15 ## FTHG ~ HS + HST + HC + HF + HO + HY + HR
16 n=nrow(data)
17 m=ceiling(n*0.7)
18 set.seed(138)
19 train_id = sort(sample(1:n,m,replace = F))
20
21 df_train = data[train_id,]
22 df_test = data[-train_id,]
```

The console shows the execution of the following commands:

```
> n=nrow(data)
> m=ceiling(n*0.7)
> set.seed(138)
> train_id = sample(1:n,m,replace = F)
> train_id = sort(sample(1:n,m,replace = F))
> df_train = data[train_id,]
> df_test = data[-train_id,]
> View(df_train)
```

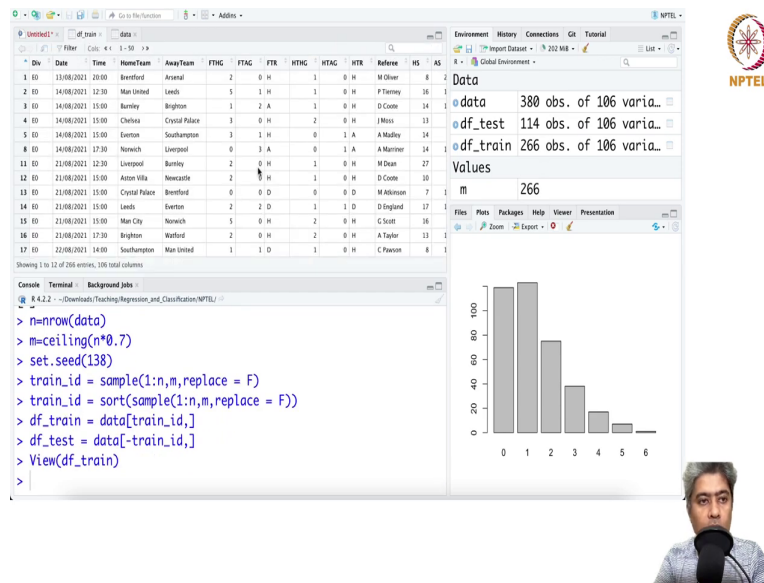
The Environment pane on the right shows the following objects:

Object	Value
data	380 obs. of 106 varia...
df_test	114 obs. of 106 varia...
df_train	266 obs. of 106 varia...
Values	m = 266

A histogram of the variable 'm' is displayed in the bottom right pane. The x-axis represents the values of 'm' (0 to 6), and the y-axis represents the frequency (0 to 100). The distribution is right-skewed, with the highest frequency at 0.

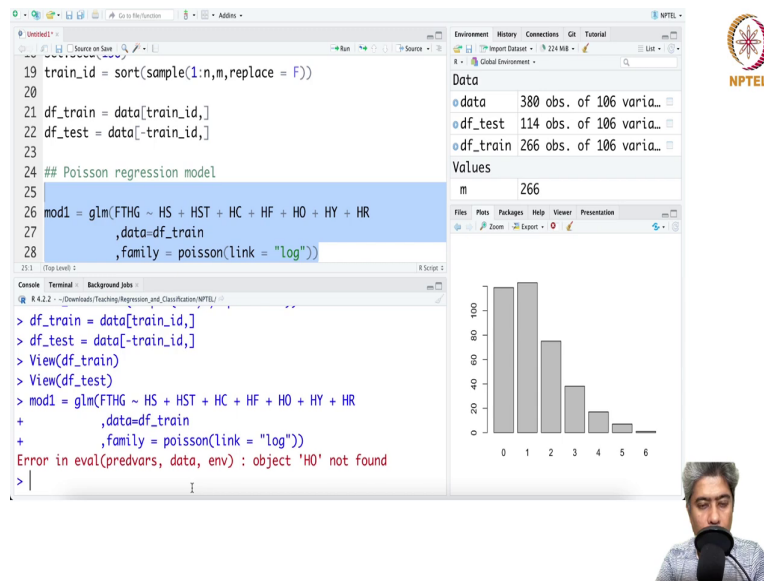
NPTEL

(Refer Slide Time: 13:05)



So, there are the data set that we got this is the training data set and this is the test data set that I got alright, nice.

(Refer Slide Time: 13:20)



The image shows the RStudio interface. The script editor on the left contains the following code:

```
19 train_id = sort(sample(1:n,m,replace = F))
20
21 df_train = data[train_id,]
22 df_test = data[-train_id,]
23
24 ## Poisson regression model
25
26 mod1 = glm(FTHG ~ HS + HST + HC + HF + HO + HY + HR
27           ,data=df_train
28           ,family = poisson(link = "log"))
```

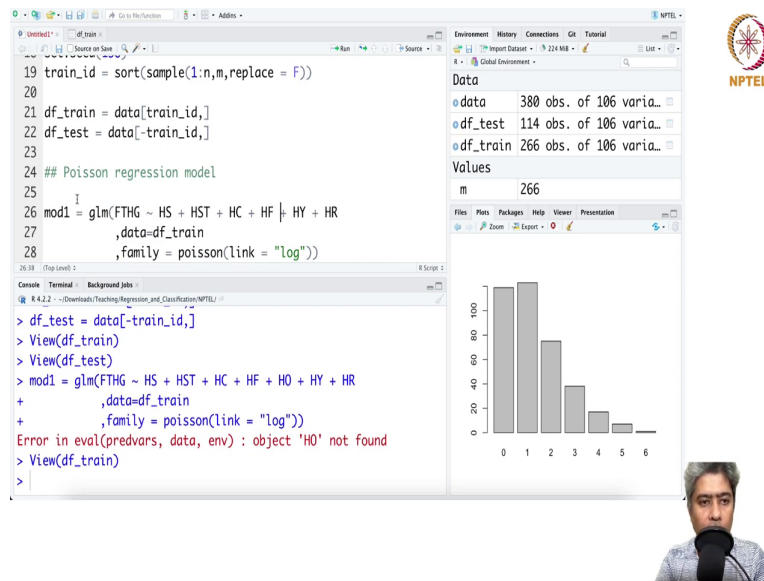
The console on the bottom left shows the execution of the code, including the model fit and a warning message: "Error in eval(predvars, data, env) : object 'HO' not found".

The environment pane on the right shows the data objects: data (380 obs. of 106 variables), df_test (114 obs. of 106 variables), and df_train (266 obs. of 106 variables). The values pane shows the value of 'm' as 266.

A histogram of the response variable 'FTHG' is displayed on the right, showing a distribution with a peak at 0 and a long tail extending to 6.

Now, what I am going to do I am going to fit a Poisson regression model fit a Poisson regression model ok. So, let me try model1 as glm, I am going to call glm and then; obviously, this model that we are talking about with the data equal to df train comma family equal to poisson link equal to log l o g. Let me try this ok it is saying not found ok. So, let me see there is a I think there is a HO HR, ok.

(Refer Slide Time: 14:40)



The image shows the RStudio interface. The script editor contains the following code:

```
19 train_id = sort(sample(1:n,m,replace = F))
20
21 df_train = data[train_id,]
22 df_test = data[-train_id,]
23
24 ## Poisson regression model
25
26 mod1 = glm(FTHG ~ HS + HST + HC + HF | HY + HR
27            ,data=df_train
28            ,family = poisson(link = "log"))
```

The console shows the following commands and output:

```
> df_test = data[-train_id,]
> View(df_train)
> View(df_test)
> mod1 = glm(FTHG ~ HS + HST + HC + HF + HO + HY + HR
+            ,data=df_train
+            ,family = poisson(link = "log"))
Error in eval(predvars, data, env) : object 'HO' not found
> View(df_train)
```

The Environment pane shows the following data objects:

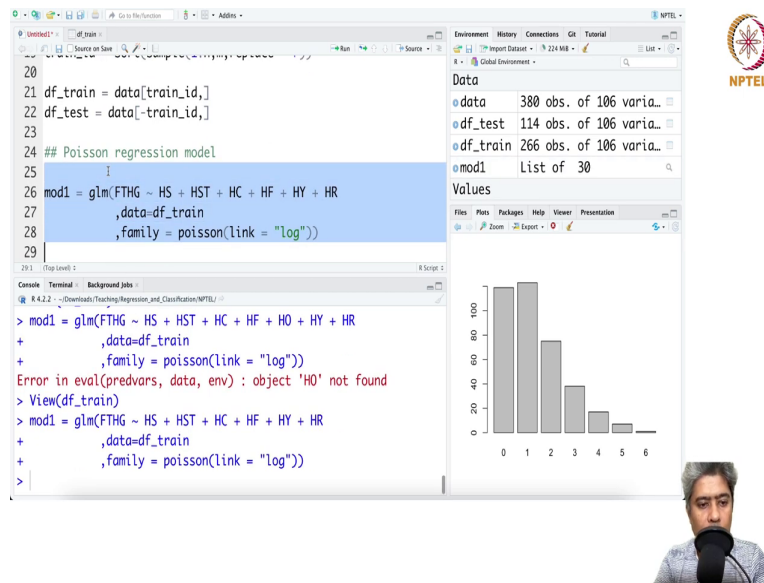
Object	Size	Class
data	380 obs. of 106 varia...	data.frame
df_test	114 obs. of 106 varia...	data.frame
df_train	266 obs. of 106 varia...	data.frame

The Values pane shows the value of 'm' as 266.

A histogram is displayed on the right, showing the distribution of a variable. The x-axis ranges from 0 to 6, and the y-axis ranges from 0 to 100. The distribution is right-skewed, with the highest frequency at 0 and 1.

So, though they are saying they have a off side. But I do not think they have provided the HO value the off side. So, ok we will see. So, let me actually what I can do? I can just make a home team off sides, but I cannot see this home team off side value is actually available.

(Refer Slide Time: 14:55)



The image shows an RStudio session. The script editor contains the following code:

```
20
21 df_train = data[train_id,]
22 df_test = data[-train_id,]
23
24 ## Poisson regression model
25
26 mod1 = glm(FTHG ~ HS + HST + HC + HF + HY + HR
27           ,data=df_train
28           ,family = poisson(link = "log"))
29
```

The console shows the execution of the model and an error message:

```
> mod1 = glm(FTHG ~ HS + HST + HC + HF + HO + HY + HR
+           ,data=df_train
+           ,family = poisson(link = "log"))
Error in eval(predvars, data, env) : object 'HO' not found
> View(df_train)
> mod1 = glm(FTHG ~ HS + HST + HC + HF + HY + HR
+           ,data=df_train
+           ,family = poisson(link = "log"))
>
```

The Environment pane on the right shows the following objects:

Object	Description
data	380 obs. of 106 varia...
df_test	114 obs. of 106 varia...
df_train	266 obs. of 106 varia...
mod1	List of 30

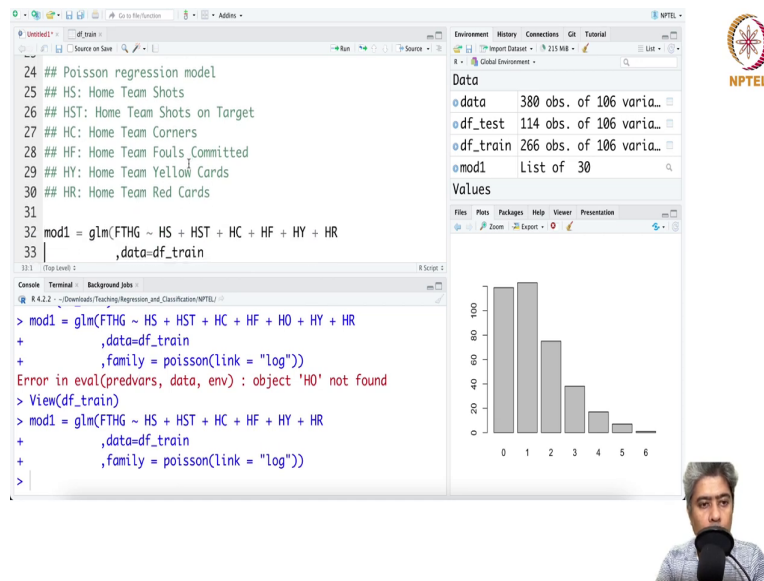
The Plots pane on the right shows a histogram of the variable FTHG (Goals per Game) with the following approximate data:

Goals per Game	Frequency
0	100
1	100
2	75
3	40
4	20
5	10
6	5

An NPTEL logo is visible in the top right corner. A small video feed of a person is visible in the bottom right corner.

So, let me just run this yes now it is being run. So, what I can do? In fact, I can just write down this name of this variable HS is Home Team Shots.

(Refer Slide Time: 15:09)



The image shows the RStudio interface. The source editor on the left contains R code for a Poisson regression model. The console on the bottom left shows the execution of the model, with an error message indicating that the variable 'HO' is not found. The environment pane on the right shows the objects in the global environment, including 'data', 'df_test', 'df_train', and 'mod1'. A histogram is displayed on the right side of the interface, showing the distribution of a variable with values from 0 to 6.

```
24 ## Poisson regression model
25 ## HS: Home Team Shots
26 ## HST: Home Team Shots on Target
27 ## HC: Home Team Corners
28 ## HF: Home Team Fouls Committed
29 ## HY: Home Team Yellow Cards
30 ## HR: Home Team Red Cards
31
32 mod1 = glm(FTHG ~ HS + HST + HC + HF + HY + HR
33            ,data=df_train
34            )
35
36 > mod1 = glm(FTHG ~ HS + HST + HC + HF + HO + HY + HR
37            ,data=df_train
38            ,family = poisson(link = "log"))
39 Error in eval(predvars, data, env) : object 'HO' not found
40 > View(df_train)
41 > mod1 = glm(FTHG ~ HS + HST + HC + HF + HY + HR
42            ,data=df_train
43            ,family = poisson(link = "log"))
44 >
```

Environment: Global Environment

Data

- data: 380 obs. of 106 varia...
- df_test: 114 obs. of 106 varia...
- df_train: 266 obs. of 106 varia...
- mod1: List of 30

Values

Files: Plots Packages Help Viewer Presentation

13.1 (Top Level) R Script

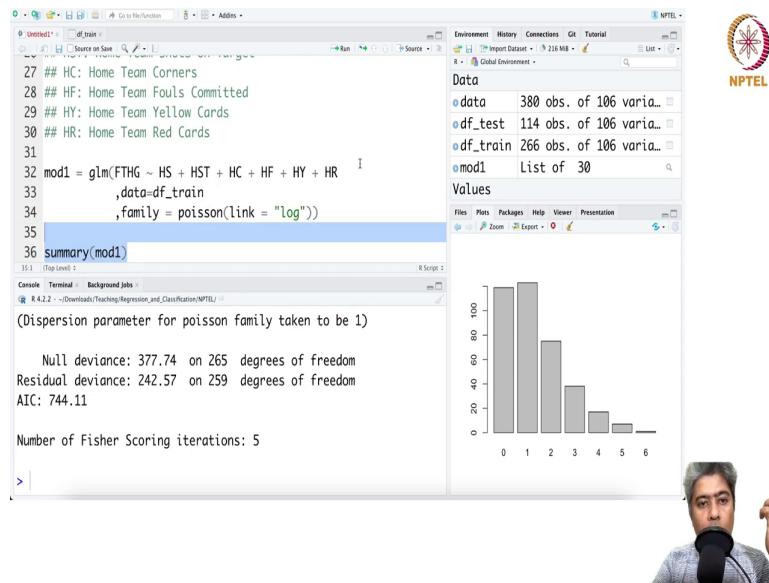
R 4.2.2 - Downloads/Teaching/Regression_and_Classification/NPTEL/

100
80
60
40
20
0

0 1 2 3 4 5 6

And, then HST, HST stands for Home Team Shots on Target. Home Team Shots on Target and then Home Team Corners HC is Home Team Corners ok and then HF is Home Teams Foul Committed, how many HF is how many fouls committed and away teams oh no sorry HY is Home Teams Yellow Card, HY is Home Teams Yellow Card and HR is Home Teams Red Cards team's red cards, alright.

(Refer Slide Time: 16:35)



The image shows the RStudio interface. The script editor on the left contains the following code:

```
27 ## HC: Home Team Corners
28 ## HF: Home Team Fouls Committed
29 ## HY: Home Team Yellow Cards
30 ## HR: Home Team Red Cards
31
32 mod1 = glm(FTHG ~ HS + HST + HC + HF + HY + HR
33           ,data=df_train
34           ,family = poisson(link = "log"))
35
36 summary(mod1)
```

The console on the bottom left displays the output of the `summary(mod1)` command:

```
(Dispersion parameter for poisson family taken to be 1)


Null deviance: 377.74  on 265  degrees of freedom
Residual deviance: 242.57  on 259  degrees of freedom
AIC: 744.11

Number of Fisher Scoring iterations: 5
```

The environment pane on the top right shows the following objects:

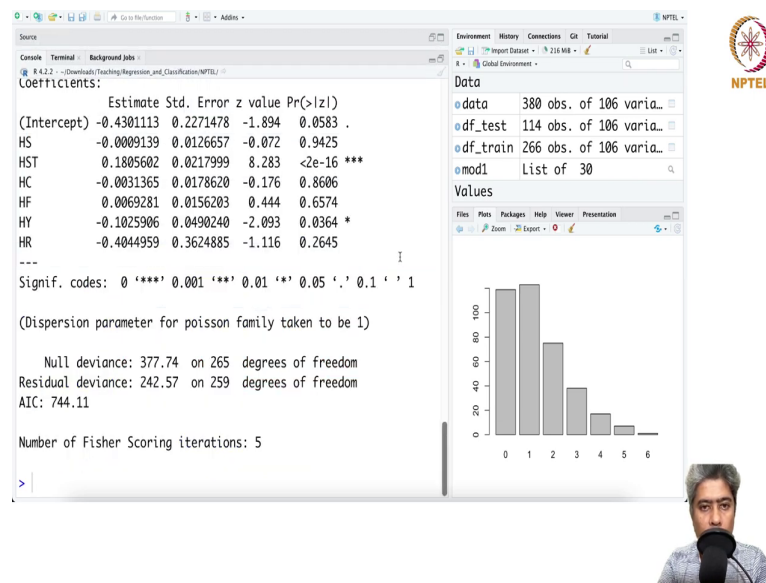
- `data`: 380 obs. of 106 varia...
- `df_test`: 114 obs. of 106 varia...
- `df_train`: 266 obs. of 106 varia...
- `mod1`: List of 30

The plot pane on the bottom right shows a histogram of the residuals, with the x-axis ranging from 0 to 6 and the y-axis ranging from 0 to 100.



So, we have done that and summary, let us run the summary.

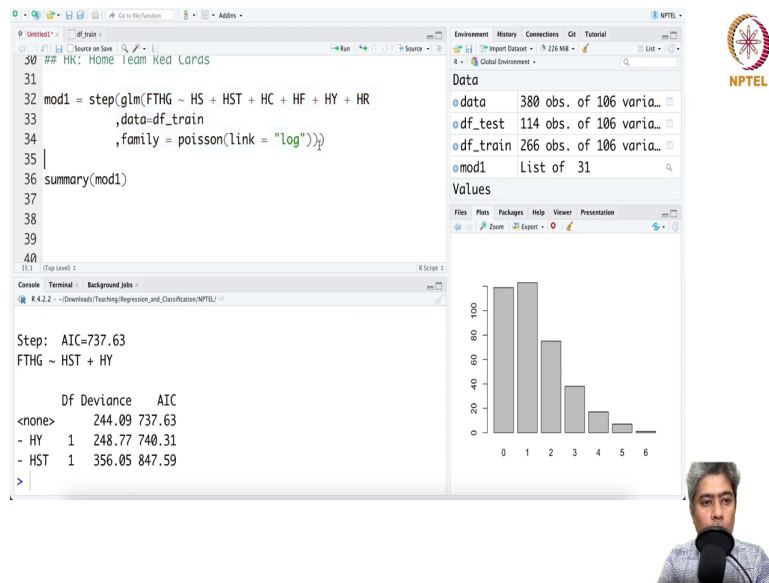
(Refer Slide Time: 16:44)



And, so, let us see how it is. So, HST it says HST has a significant effect on the number of goals that you scored obviously, and then they are saying that home team number of yellow cards that you have seen has a negative impact rate does not have any effect ok probably there will be 0 in any way.

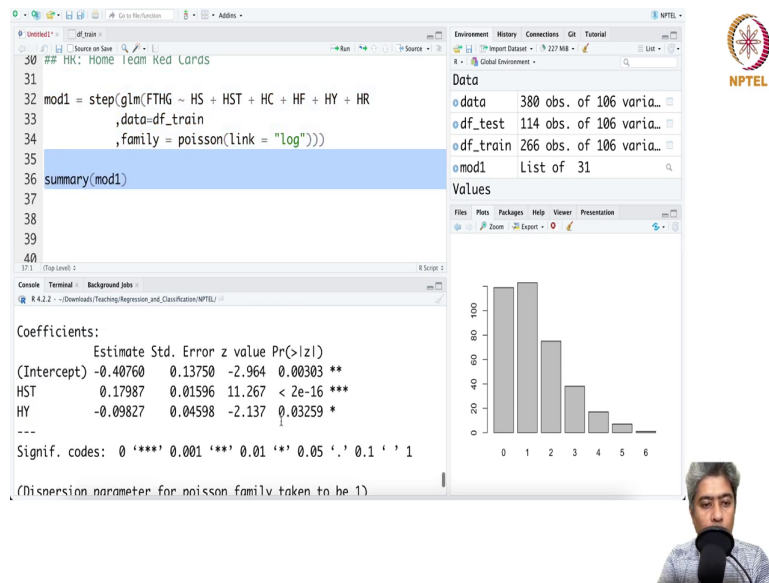
So, anyway so, this is an interesting thing, but the other things like HS how many shots you are taking does not going to help you out, but how many shots on target that is going to be a extremely important number of goals scored by the home team ok.

(Refer Slide Time: 17:35)



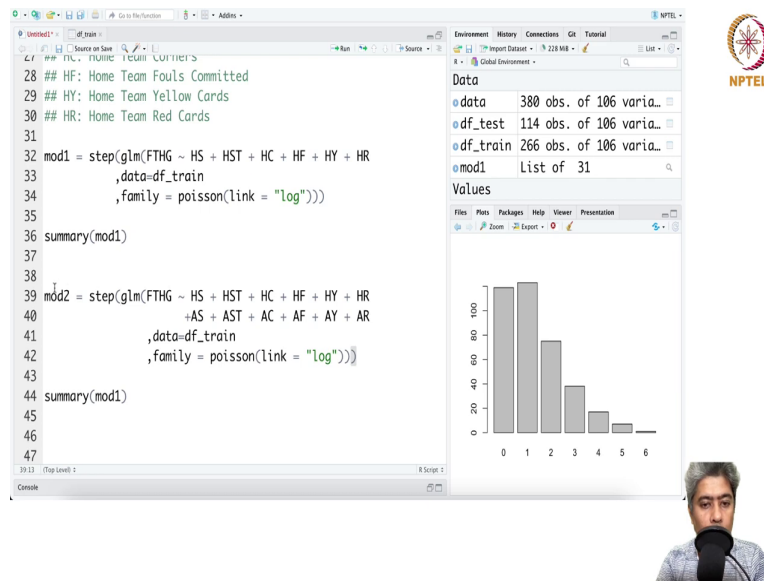
So, what we can do? We can also put a step function here.

(Refer Slide Time: 17:43)



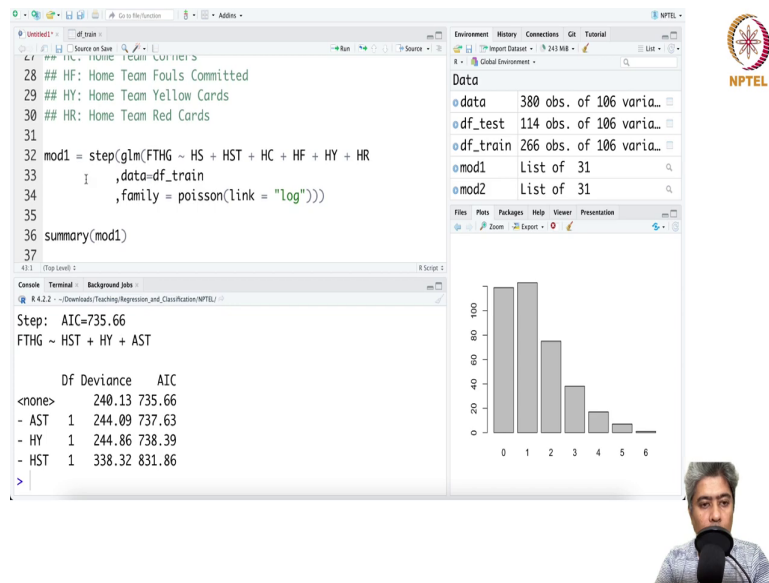
And, if you do that the simple model will be built by HST and a HY rest of the variables are being kind of dropped off.

(Refer Slide Time: 18:08)



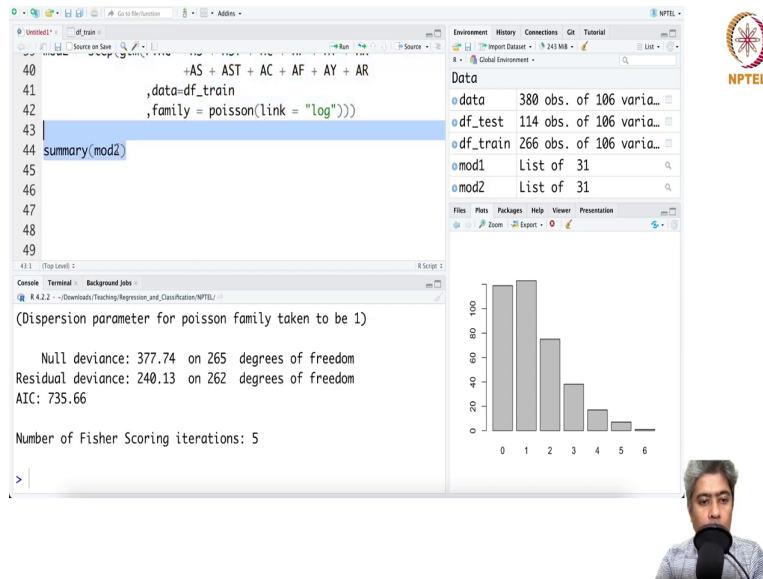
Now, I can build the same model similar models with the away teams goals also. So, that will be a second model, right maybe I will just along with these variables but we will only modeling the home teams goals only, away team how many shots they have taken, away team how many shots on target on their target, away teams AC, AF, fouls committed, I think what was the HC is a home teams corner AY and AR, alright.

(Refer Slide Time: 18:44)



Now, if you do run step by selection.

(Refer Slide Time: 18:47)



The image shows the RStudio interface. The script editor on the left contains the following code:

```
40 +AS + AST + AC + AF + AY + AR
41 ,data=df_train
42 ,family = poisson(link = "log"))
43
44 summary(mod2)
45
46
47
48
49
```

The console on the bottom left displays the output of the model fit:

```
(Dispersion parameter for poisson family taken to be 1)


Null deviance: 377.74 on 265 degrees of freedom
Residual deviance: 240.13 on 262 degrees of freedom
AIC: 735.66

Number of Fisher Scoring iterations: 5
```

The Environment pane on the right shows the following objects:

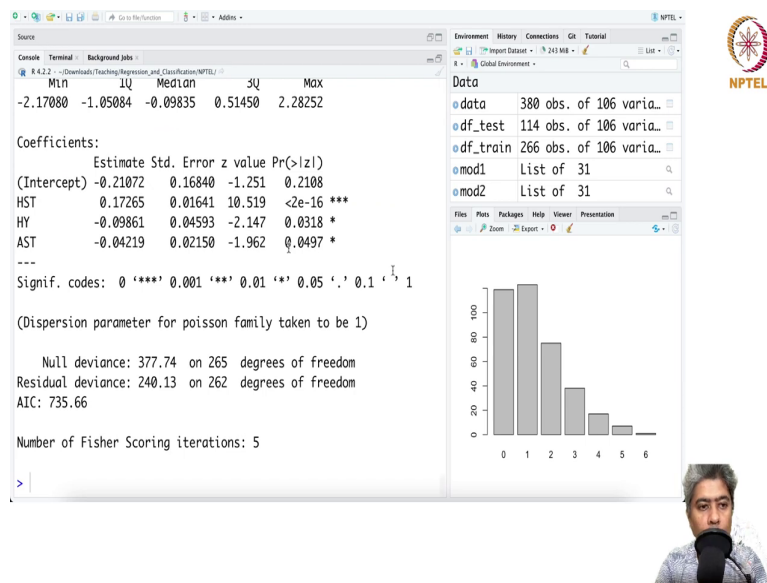
Object	Description
data	380 obs. of 106 varia...
df_test	114 obs. of 106 varia...
df_train	266 obs. of 106 varia...
mod1	List of 31
mod2	List of 31

A histogram of the residuals is displayed on the right side of the console, showing the distribution of the residuals. The x-axis represents the residuals, ranging from -6 to 6, and the y-axis represents the frequency, ranging from 0 to 100.



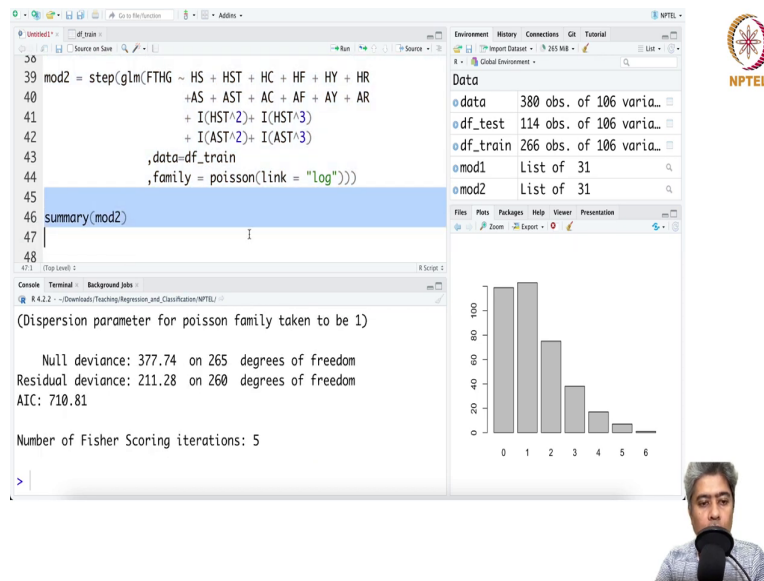
And, ask your model 2 to run ok.

(Refer Slide Time: 18:50)



How many shots on target has a positive effect and how many shots on target by the away team has obviously, a negative effect, but still, it says a very strong effect. So, it might even happen that it has some kind of you know quadratic or some kind of non-linear effect.

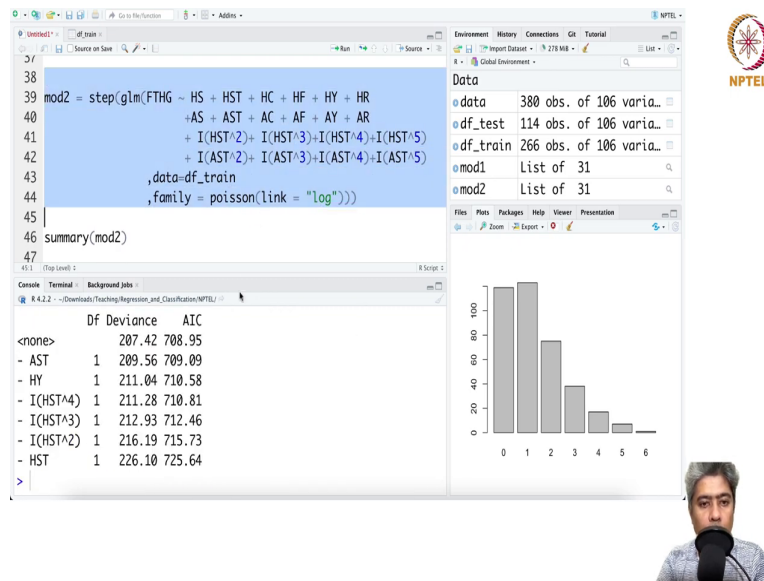
(Refer Slide Time: 19:18)



So, in case it has some we would like to have a square maybe cube we do not know and then maybe I into AST square effect square effect and I AST cubic effect. Let me just run this with.

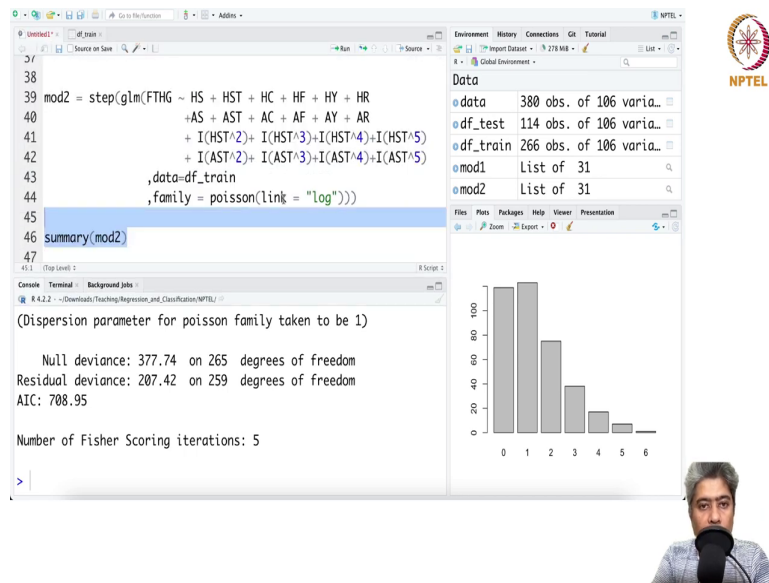


(Refer Slide Time: 20:26)



So, this is really very interesting that I am getting I would even want mine to put a few more engineering effect just T to the power 4 IHST to the power 5 ok. This is AST, this is AST.

(Refer Slide Time: 20:55)



The image shows a screenshot of the RStudio interface. The main editor window displays R code for fitting a Poisson regression model. The code defines a model 'mod2' using the 'glm' function with the 'poisson' family and a log link. The model is then summarized using the 'summary' function. The console window shows the output of the summary function, including the null deviance, residual deviance, AIC, and the number of Fisher Scoring iterations. The environment pane on the right shows the objects in the global environment, including 'data', 'df_test', 'df_train', 'mod1', and 'mod2'. A histogram is also visible in the bottom right corner of the RStudio window.

```
37  
38  
39 mod2 = stepAIC(glm(FTHG ~ HS + HST + HC + HF + HY + HR  
40 + AS + AST + AC + AF + AY + AR  
41 + I(HST^2)+ I(HST^3)+I(HST^4)+I(HST^5)  
42 + I(AST^2)+ I(AST^3)+I(AST^4)+I(AST^5)  
43 ,data=df_train  
44 ,family = poisson(link = "log"))  
45  
46 summary(mod2)  
47
```

Console Output:

```
45.1 (Top Level) > R Script 5  
R 4.2.2 - ~/Downloads/Teaching/Regression_and_Classification/NPTEL/ >  
(Dispersion parameter for poisson family taken to be 1)  
  
Null deviance: 377.74 on 265 degrees of freedom  
Residual deviance: 207.42 on 259 degrees of freedom  
AIC: 708.95  
  
Number of Fisher Scoring iterations: 5  
>
```

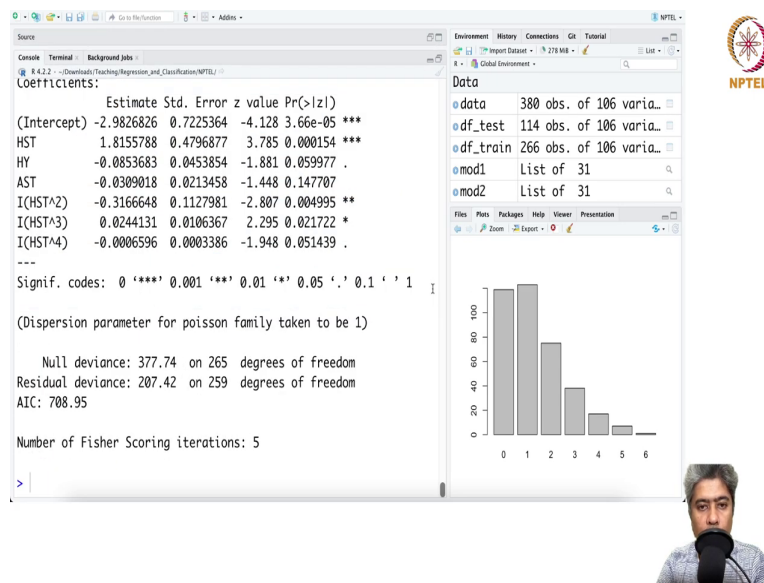
Environment:

Object	Value
data	380 obs. of 106 varia...
df_test	114 obs. of 106 varia...
df_train	266 obs. of 106 varia...
mod1	List of 31
mod2	List of 31

Histogram Data:

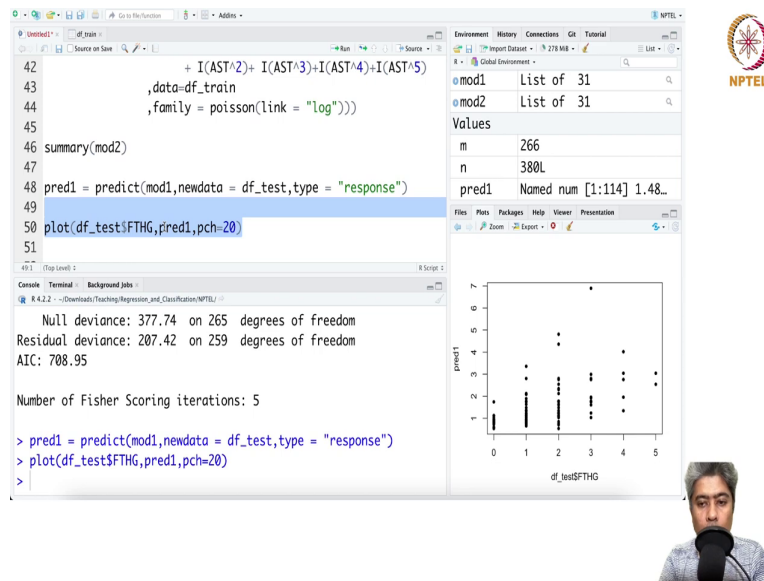
Bin Range	Frequency
0 - 1	100
1 - 2	100
2 - 3	75
3 - 4	40
4 - 5	20
5 - 6	10

(Refer Slide Time: 20:56)



Now, if I run it and if I just this see this ok fifth power does not have any effect, but up to fourth power home team shots on target has a very significant effect. So, this is a very interesting phenomena that we just found. In fact, you can try few more engineering things, but I am just stopping here for now.

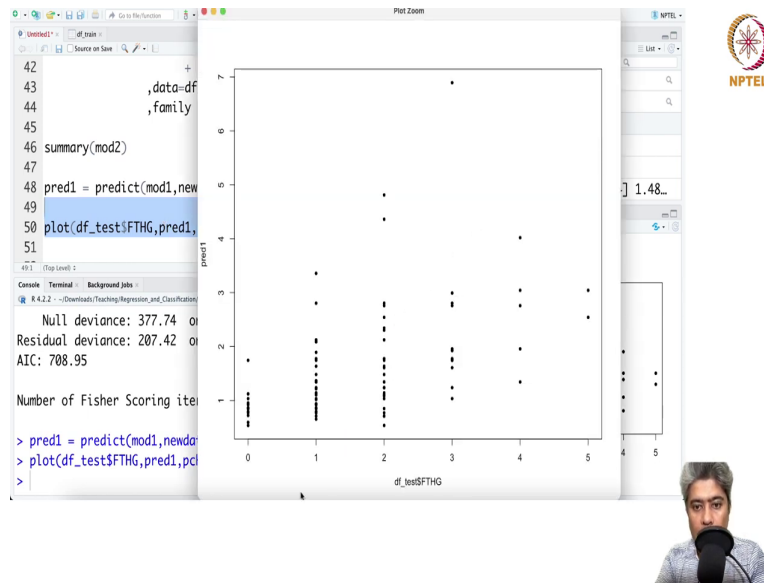
(Refer Slide Time: 21:36)



And, what do you want? I want to predict the number of goals scored by the home team and number of goals scored by the home team say predict1 from the model1 we would like to predict from the model1 and new data equal to df test df test and type equal to you have to say response. So, this is predict, these are the predicted values.

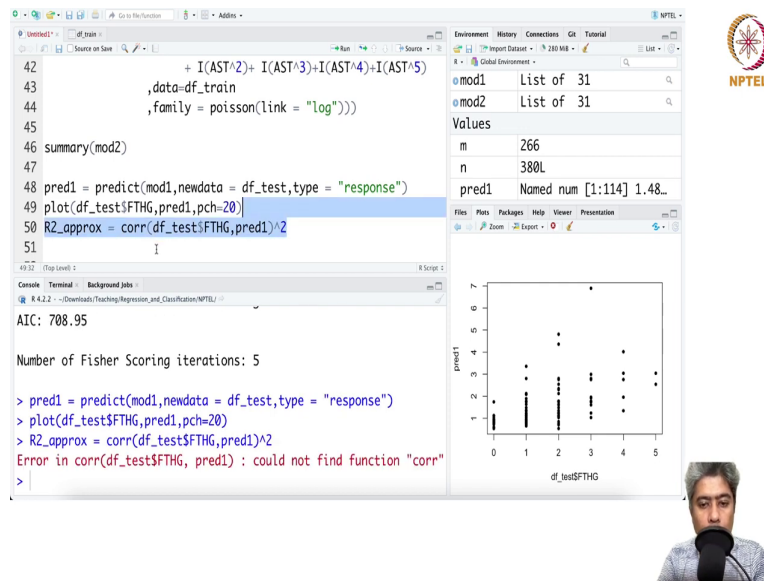
And, now what I have to just do? I want to do I want to just plot the from the df test dollar home team score actually the test data set how many goals scored by the full time home teams goal yeah FTHG and the predicted values if I just plot them pch is equal to 20.

(Refer Slide Time: 22:50)



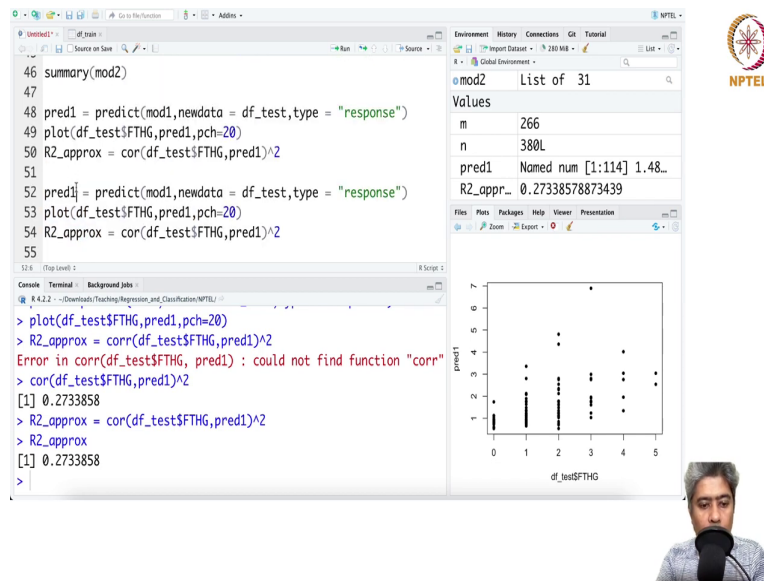
So, we see that so, on the x-axis this is the actual number of goals whereas, these are the predicted number of goals. So, there is some kind of you can do predict to an extent not great, but you can do that.

(Refer Slide Time: 23:08)



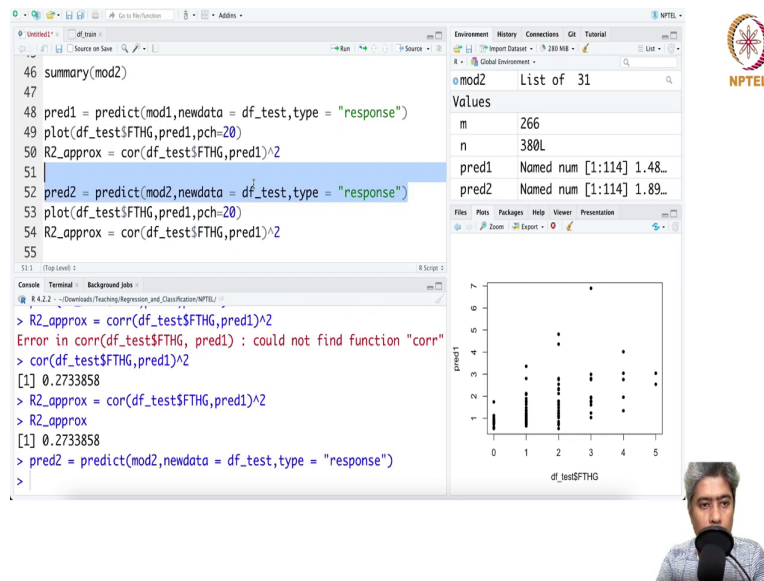
So, I can compute R square approximately by just taking the correlation between the two and then square them up, then square them up.

(Refer Slide Time: 23:30)



Yeah. So, we can see this is not very strong R square, but let us try the other model.

(Refer Slide Time: 23:42)



The image shows an RStudio session. The script editor contains the following code:

```
46 summary(mod2)
47
48 pred1 = predict(mod1,newdata = df_test,type = "response")
49 plot(df_test$FTHG,pred1,pch=20)
50 R2_approx = cor(df_test$FTHG,pred1)^2
51
52 pred2 = predict(mod2,newdata = df_test,type = "response")
53 plot(df_test$FTHG,pred2,pch=20)
54 R2_approx = cor(df_test$FTHG,pred1)^2
55
```

The console shows the following output:

```
> R2_approx = cor(df_test$FTHG,pred1)^2
Error in cor(df_test$FTHG, pred1) : could not find function "cor"
> cor(df_test$FTHG,pred1)^2
[1] 0.2733858
> R2_approx = cor(df_test$FTHG,pred1)^2
> R2_approx
[1] 0.2733858
> pred2 = predict(mod2,newdata = df_test,type = "response")
>
```

The Environment pane shows the following variables:

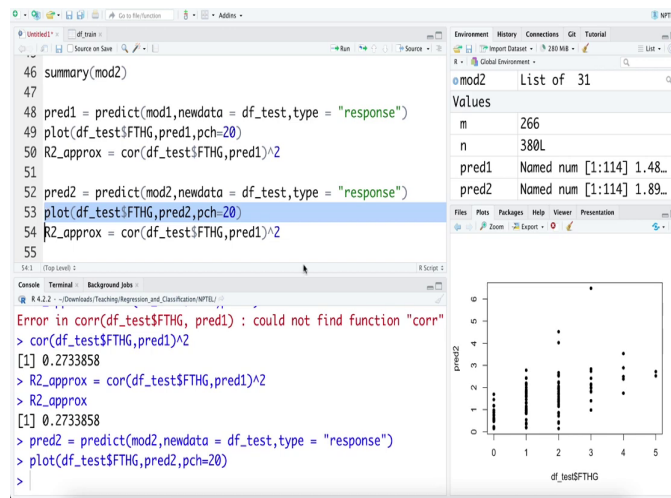
Variable	Value
m	266
n	380L
pred1	Named num [1:114] 1.48...
pred2	Named num [1:114] 1.89...

The plot pane shows a scatter plot of pred1 vs df_test\$FTHG. The x-axis is labeled 'df_test\$FTHG' and ranges from 0 to 5. The y-axis is labeled 'pred1' and ranges from 0 to 10. The plot shows a positive correlation between the two variables.

The NPTEL logo is visible in the top right corner.

So, predict2 mod 2 model 2 and if you predict the second model.

(Refer Slide Time: 23:54)

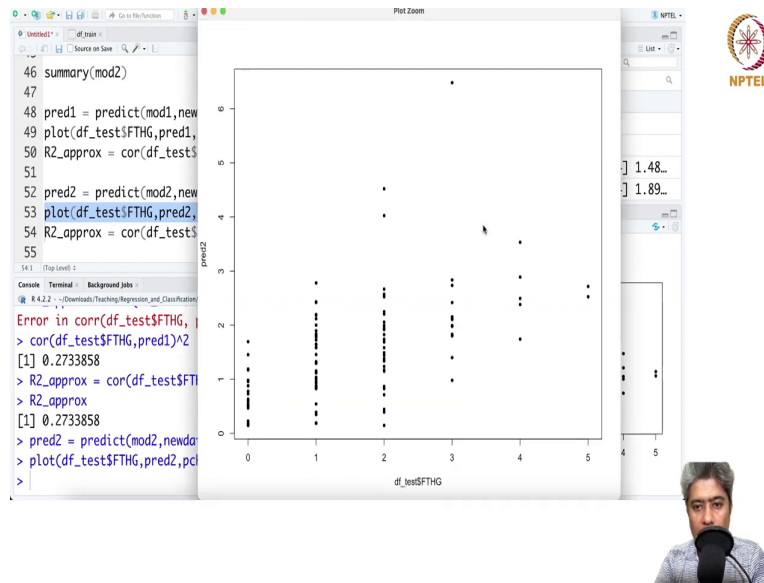


The image shows the RStudio interface with the following components:

- Source Editor:** Contains R code for model evaluation. Line 53, `plot(df_test$FTHG, pred2, pch=20)`, is highlighted in blue.
- Environment:** Shows a list of objects including `mod2` (List of 31), `pred1` (Named num [1:114] 1.48...), and `pred2` (Named num [1:114] 1.89...).
- Console:** Displays the execution of the code. It shows an error message: `Error in cor(df_test$FTHG, pred1) : could not find function "cor"`. Subsequent lines show the calculation of `R2_approx` as 0.2733858 and the prediction of `pred2`.
- Plots:** A scatter plot titled `pred2` vs `df_test$FTHG` is displayed. The x-axis ranges from 0 to 5, and the y-axis ranges from 0 to 10. Data points are represented by black circles (pch=20).

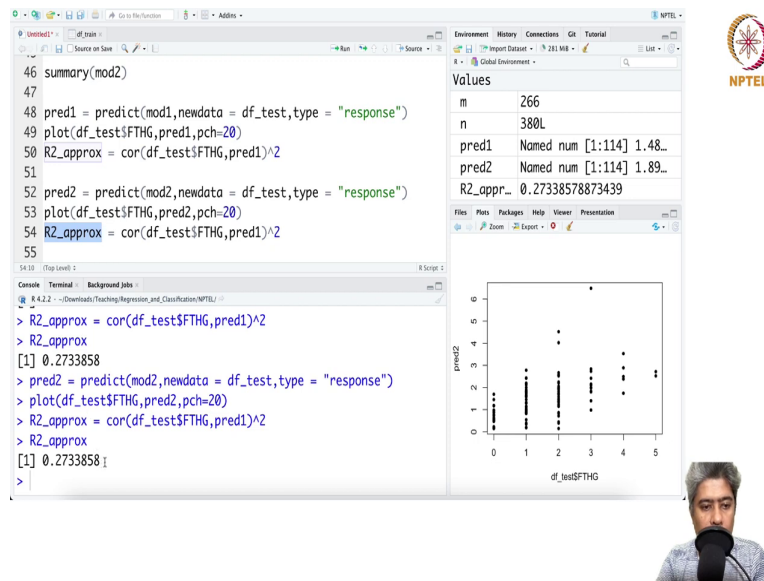


(Refer Slide Time: 23:58)



Second model is slightly maybe better in terms of its prediction on the higher cases.

(Refer Slide Time: 24:06)



The image shows an RStudio session. The source editor on the left contains R code for fitting a model and calculating R-squared values. The console on the bottom left shows the execution of these commands. The environment pane on the right displays the values of the objects created. A scatter plot of predicted values (pred2) against observed values (df_test\$FTHG) is shown on the right.

```
46 summary(mod2)
47
48 pred1 = predict(mod1,newdata = df_test,type = "response")
49 plot(df_test$FTHG,pred1,pch=20)
50 R2_approx = cor(df_test$FTHG,pred1)^2
51
52 pred2 = predict(mod2,newdata = df_test,type = "response")
53 plot(df_test$FTHG,pred2,pch=20)
54 R2_approx = cor(df_test$FTHG,pred1)^2
55
```

Console output:

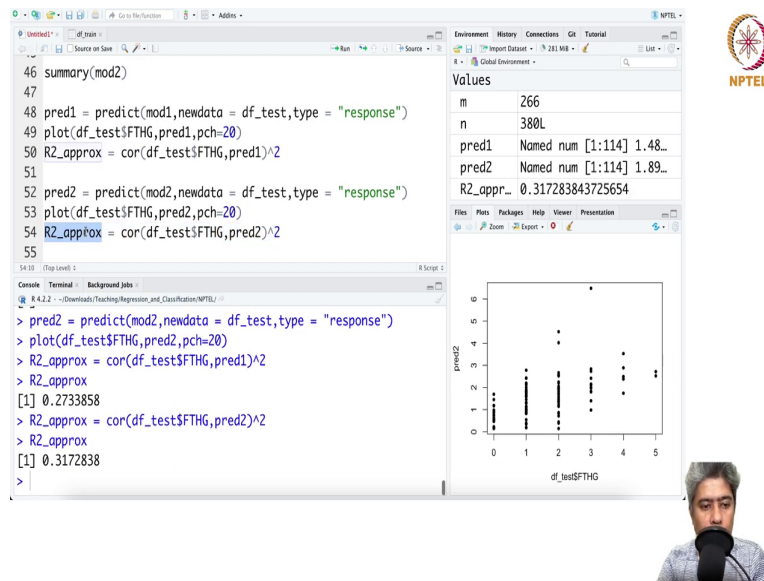
```
> R2_approx = cor(df_test$FTHG,pred1)^2
> R2_approx
[1] 0.2733858
> pred2 = predict(mod2,newdata = df_test,type = "response")
> plot(df_test$FTHG,pred2,pch=20)
> R2_approx = cor(df_test$FTHG,pred1)^2
> R2_approx
[1] 0.2733858
>
```

Environment pane values:

Object	Value
m	266
n	380L
pred1	Named num [1:114] 1.48..
pred2	Named num [1:114] 1.89..
R2_appr...	0.27338578873439

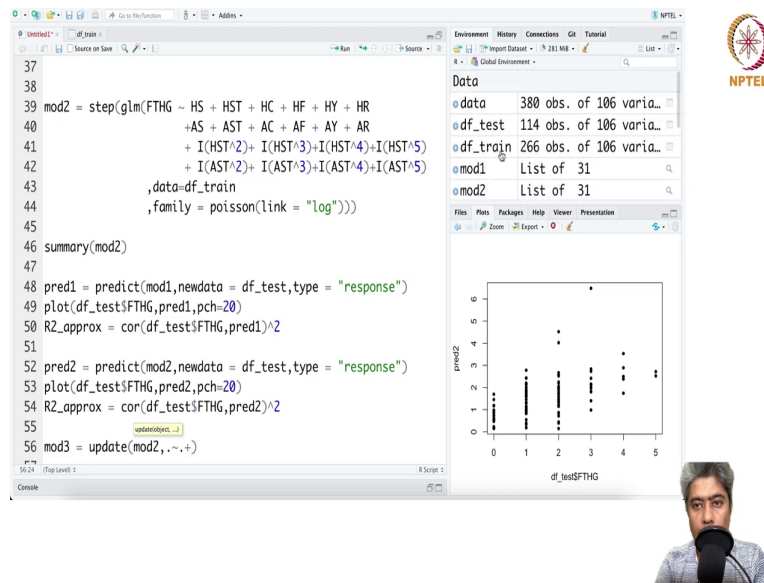
Scatter plot showing predicted values (pred2) on the y-axis versus observed values (df_test\$FTHG) on the x-axis. The plot shows a positive correlation between the two variables.

(Refer Slide Time: 24:14)



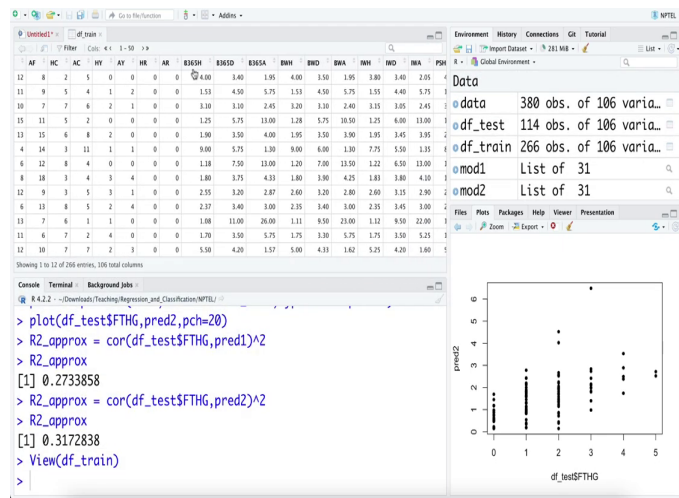
And, then it is actually pred 2. So, now, it is about 4 point increase that we are seeing in the second cases. Now, if we consider this second model as sort of a base model, then we want to see if the any of these baiting odds has any predictive power or not.

(Refer Slide Time: 24:45)

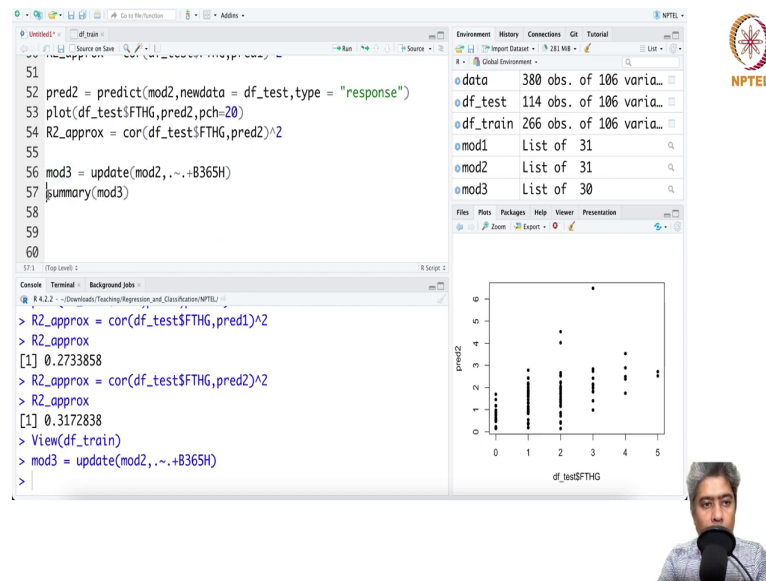


So, what I am going to do I am going to develop a third model mod3 which is essentially update the model2 where I will do just add a few more variable.

(Refer Slide Time: 25:06)

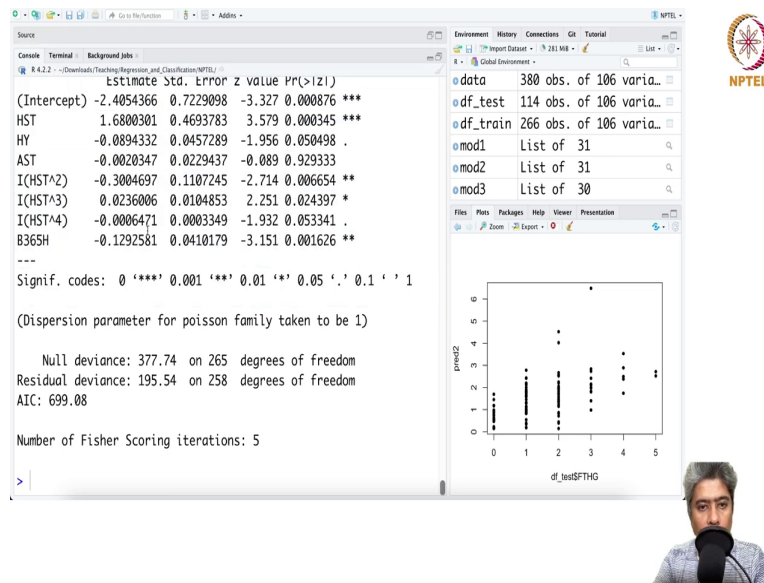


(Refer Slide Time: 25:14)



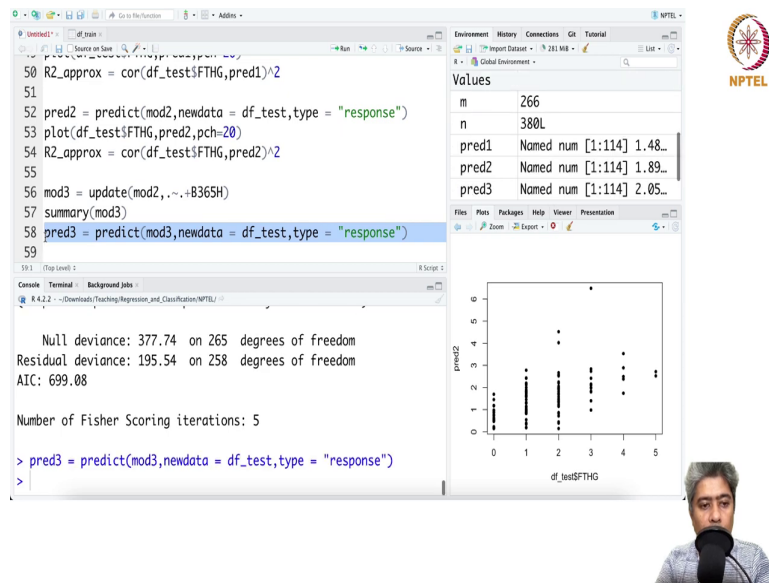
So, I will just go and say B365H. Am I will I be B365H if I do that can I get a better model? So, summary. So, fit the model I am saying basically in the update; update is a built-in function in R. You can fit the same model whatever the model second model you have here along with that you take another predictor B365H on the same data set and everything. Now, if you fit a summary with say mod 3.

(Refer Slide Time: 25:51)



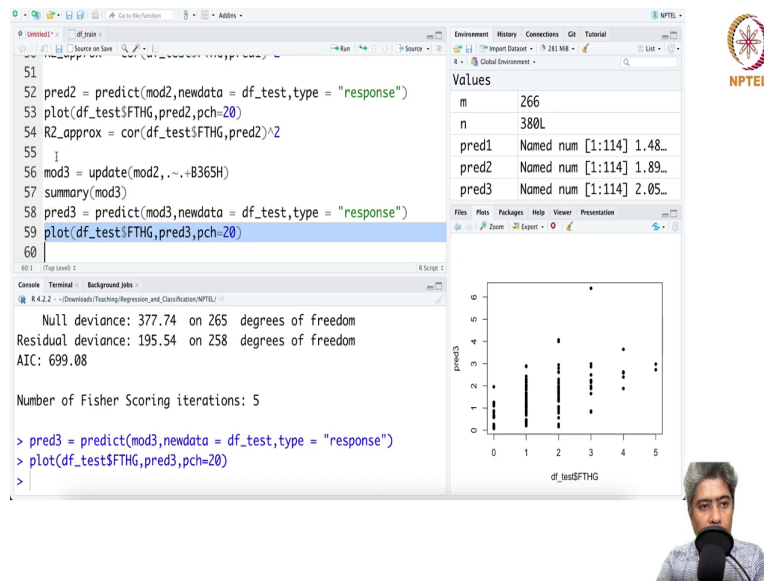
And, this B365H has a predictive power as its significant looks like it does have a significant effect on the number of goals its course. So, what it does is then can we predict from this.

(Refer Slide Time: 26:15)



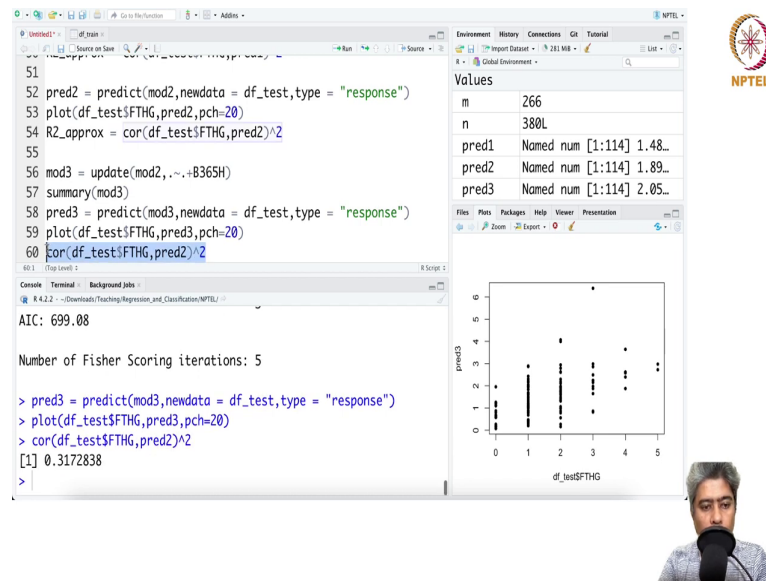
And, let us do that and take the second model and do the third prediction and if we do that.

(Refer Slide Time: 26:32)



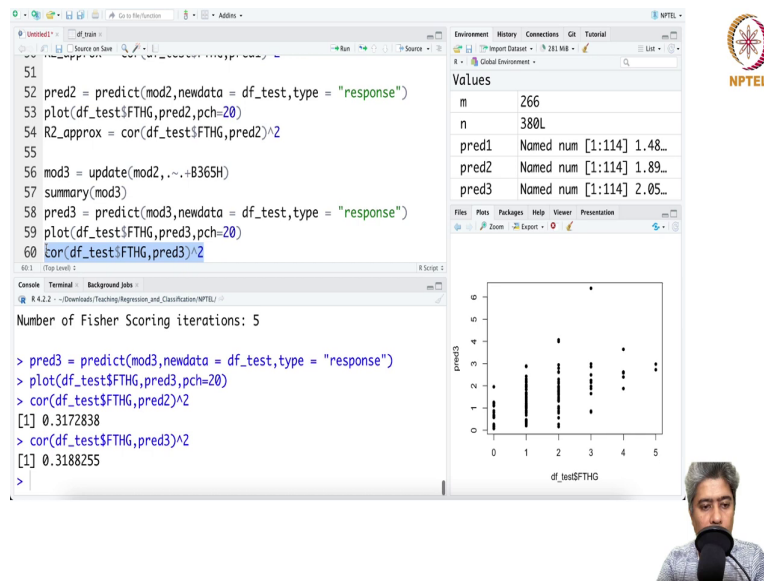
And, then what is the; what is the plot? How does it improve significantly in the out of the sample.

(Refer Slide Time: 26:45)



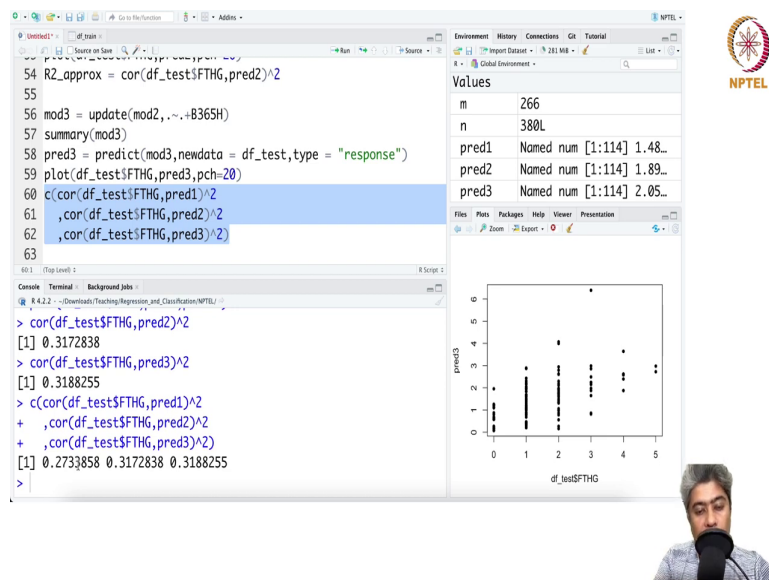
So, maybe a little we do not know, let us compute the correlations let it compute the correlations. So, it is 3 1 almost same.

(Refer Slide Time: 26:53)



Sorry I have forgot to update the third prediction here. So, a little bit a little bit. So, we can what we can do we can just take the first three prediction in the in a vector.

(Refer Slide Time: 27:13)



And, this is the predict 1, this is predict 2 and this is predict 3 and we can just. So, this is the first model 27 percent approximately 27 percent accuracy that we are seeing, this is the second model 31.7 and the third model with bait 365 we are getting 31.8 percent predictive accuracy in terms of you know these models are. It is not very great, but at the same time this model this shows that sometimes your predictive model it has some predictive power which you cannot ignore, but at the same time it is not great.

So, there are a lot of rooms to improve. I would recommend why do not you try yourself that I will share this code on the NPTEL platform, but at the same time I will highly encourage you that you should try this try yourself and you know and tell me what is your what predictive out of the sample predictive accuracy this is my out of the sample predictive accuracy.

Not good, very bad actually. I would not say actually very bad because you know these are very difficult problem and it is a predictive accuracy in the out of the sample you cannot ignore it is not like 0 or 5 percent or 10 percent, it is like 30 percent 32 percent not bad, but you can improve a lot there is a lot of room to improve. So, why do not you try yourself? There are lots of variables I think 100 plus variables are there try yourself and maybe you can push it to 60, 70 or 80 percent, why not?

So, try yourself and. So, we will and write me in the YouTube videos link or you just e-mail me and what is an what is your accuracy you got in out of the sample and what model works for you the best.

Thank you very much. See you in the next video, next lecture.