LECTURE 5 – PART A – CHI-SQUARE TESTS FOR CATEGORICAL MODELS

Book Chapter 11

Types of Variables and Basic Types of Analysis

Dependent	Outcome	Predictor	Analysis	Estimation Method
Ratio (Continuous)	Value Prediction	Categorical Only	ANOVA (Regression)	Least Squares
Ratio (Continuous)	Value Prediction	Categorical and/or Continuous	Regression	Least Squares
Nominal (Categorical)	Association (Dependence)	Categorical Only	Contingency Table	Chi-Square
Nominal (Categorical)	Category Value Probability, Classification	Categorical and/or Continuous	Logistic Regression	Maximum Likelihood

Note that many of the techniques such as regression and logistic regression can be specialized to *predict ranks* when the *dependent data* is *ordinal*. Other techniques also take advantage of ordinality of predictors (relative to nominall).

Functions of Random Variables

- Recall that we saw:
  - In a random sample, if each sample point is an *independent* draw from an *Normal distribution* N(μ,σ), then and the sample mean $\overline{X} = (\Sigma_{i=1,\ldots,n} X_i)/n \sim N(\mu, \sigma/\sqrt{n})$

- In the same manner:

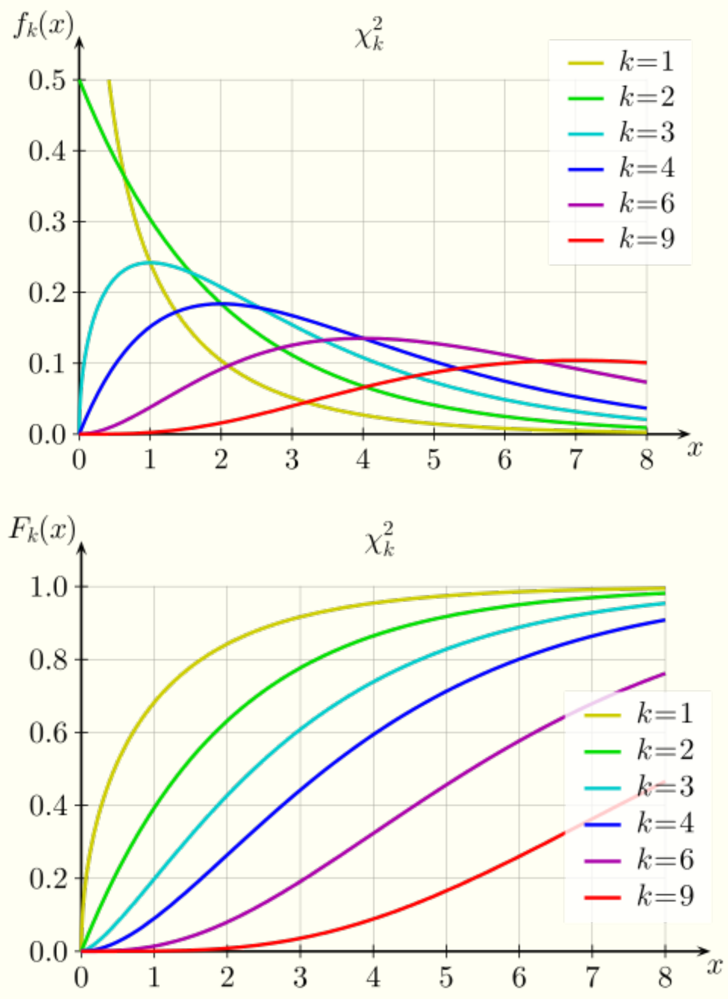
  - If $Z \sim N(0,1)$ is a standard normal random variable, then Z^2 has a **Chi-Square** distribution with 1 degree of freedom.

  - The sum of independent Chi-Square random variables also has a Chi-square distribution. That is, If X is the sum of the squares of standard *k* normal random variables $(Z_1)^2 + (Z_2)^2 + \ldots + (Z_k)^2$, then X has a Chi-square distribution χ^2_k, where k is degrees of freedom.

  - The *ratio* of two Chi-square random variables has an **F-distribution**.

The Chi-Square Distribution

- For the χ^2_k distribution, the population mean is $\mu = k$ and the population standard deviation is $\sigma = \sqrt{2k}$.
  - 1. The curve is nonsymmetrical and skewed to the right.
  - 2. There is a different chi-square curve for each *df*.
- The degrees of freedom k depends on how the Chi-square is being used.
- We will look at 3 applications of the Chi-square distribution
  - *Test for Independence* (whether two random variables are independent or not)
  - *Goodness of Fit* (whether a population fits a given distribution)
  - *Test for Homogeneity* (whether two populations have the same distribution)



The Chi-Square Distribution – Test for Independence

- A **contingency** table is a table showing the distribution of one discrete random variable in rows and another discrete random variable in columns, and used to study the *association* between the two discrete random variables.

	$Y = y_1$	$Y = y_2$	$Y = y_3$	$Y = y_4$	X - marginal
$X = x_1$	$P(X = x_1, Y = y_1)$ +	$P(X = x_1, Y = y_2)$ +	$P(X = x_1, Y = y_3)$ +	$P(X = x_1, Y = y_4)$ +	**= $P(X = x_1)$ +**
$X = x_2$	$P(X = x_2, Y = y_1)$ =	$P(X = x_2, Y = y_3)$	$P(X = x_2, Y = y_3)$	$P(X = x_2, Y = y_4)$	**$P(X = x_2)$ =**
Y - marginal	**$P(Y = y_1)$**	**$P(Y = y_2)$**	**$P(Y = y_3)$**	**$P(Y = y_4)$**	**1**

- The ***probabilities*** *of X and Y*, respectively, because they are on the margins of the contingency table.

- The entries inside the cells of the table (such as $P(X = x_2, Y = y_3)$) are called the ***joint probabilities***.

- It is easy to see that we can get ***conditional probabilities*** such as:

The Chi-Square Distribution – Test for Independence

- Recall that if two random variables X and Y are independent, If X and Y are **independent** P(X|Y) = P(X) or alternatively P(X,Y) = P(X).P(Y) i.e., the joint probabilities will be the product of the marginal probabilities
- We show a contingency table where X and Y are independent. X and Y are not independent if even one cell in the table does not obey P(X,Y) = P(X).P(Y)

	$Y = y_1$	$Y = y_2$	$Y = y_3$	$Y = y_4$	X - marginal
$X = x_1$	$P(X = x_1, Y = y_1) = P(X = x_1)*P(Y = y_1)$	$P(X = x_1, Y = y_2) = P(X = x_1)*P(Y = y_2)$	$P(X = x_1, Y = y_3) = P(X = x_1)*P(Y = y_3)$	$P(X = x_1, Y = y_4) = P(X = x_1)*P(Y = y_4)$	$P(X = x_1)$
$X = x_2$	$P(X = x_2, Y = y_2) = P(X = x_2)*P(Y = y_1)$	$P(X = x_2, Y = y_2) = P(X = x_2)*P(Y = y_3)$	$P(X = x_2, Y = y_3) = P(X = x_2)*P(Y = y_3)$	$P(X = x_2, Y = y_3) = P(X = x_2)*P(Y = y_4)$	$P(X = x_2)$
Y - marginal	$P(Y = y_1)$	$P(Y = y_2)$	$P(Y = y_3)$	$P(Y = y_4)$	1

The Chi-Square Distribution – Test for Independence (Book Example 11.6)

Practice with calculator

- In a volunteer group, adults 21 and older volunteer from one to nine hours each week to spend time with a disabled senior citizen. The program recruits among community college students, four-year college students, and nonstudents.
- Table A is an ***observed sample*** of the adult volunteers and the number of hours they volunteer per week.
- *Is the number of hours volunteered **independent** of the type of volunteer?*
- Here X is the discrete random variable *number of hours volunteered* and Y is the discrete random variable *type of volunteer* (technically Y is not a random variable because it is categorical, but we can make it a discrete random variable by mapping each type to a number)
- If we take the relative frequency approach to probability, then we can easily convert the observed table into marginal and joint probabilities as shown in **Table B** by dividing every cell value (including margin values) in **Table A** by 839.

Table A – Observed Counts

Type\Hours	**1–3**	**4–6**	**7–9**	**Total**
Community College	111	96	48	*255*
Four-Year College	96	133	61	*290*
Nonstudents	91	150	53	*294*
Total	*298*	*379*	*162*	*839*

Table B – Observed Probabilities

Type\Hours	**1–3**	**4–6**	**7–9**	**Total**
Community College	0.132	0.114	0.057	0.304
Four-Year College	0.114	0.159	0.073	0.346
Nonstudents	0.108	0.179	0.063	0.350
Total	0.355	0.452	0.193	1.000

Practice with calculator

The Chi-Square Distribution – Test for Independence

- Now, *if* X and Y were **independent** our **Expected** table of probabilities (C) and **Expected** table of counts (D) are shown. In **Table C**, every joint probability is the product of the corresponding marginals and in **Table D**, every cell count inside the margins is the joint probability multiplies by 839.

- That is,
  Expected Count for cell (i, j) =

$$\frac{\textbf{Margin count for row i} * \textbf{Margin Count for Column j}}{\textbf{Total Count}}$$

- So in practice, we do not need to convert to probabilities, but simply use the above Expected Count formulas for the contingency table to get expected counts

Table C – Expected Probabilities

Type\Hours	1–3	4–6	7–9	Total
Community College	0.108	0.137	0.058	0.304
Four-Year College	0.123	0.156	0.067	0.346
Nonstudents	0.124	0.159	0.068	0.350
Total	0.355	0.452	0.193	1.000

Table D – Expected Counts

Type\Hours	1–3	4–6	7–9	Total
Community College	91	115	49	*255*
Four-Year College	103	131	56	*290*
Nonstudents	104	133	57	*294*
Total	*298*	*379*	*162*	*839*

Practice with calculator

The Chi-Square Distribution – Test for Independence

Table D - Expected Counts

Type\Hours	1–3	4–6	7–9	Total
Community College	91	115	49	*255*
Four-Year College	103	131	56	*290*
Nonstudents	104	133	57	*294*
Total	*298*	*379*	*162*	*839*

Table C - Expected Probabilities

Type\Hours	1–3	4–6	7–9	Total
Community College	0.108	0.137	0.058	0.304
Four-Year College	0.123	0.156	0.067	0.346
Nonstudents	0.124	0.159	0.068	0.350
Total	0.355	0.452	0.193	1.000

Table A - Observed Counts

Type\Hours	1–3	4–6	7–9	Total
Community College	111	96	48	*255*
Four-Year College	96	133	61	*290*
Nonstudents	91	150	53	*294*
Total	*298*	*379*	*162*	*839*

Table B - Observed Probabilities

Type\Hours	1–3	4–6	7–9	Total
Community College	0.132	0.114	0.057	0.304
Four-Year College	0.114	0.159	0.073	0.346
Nonstudents	0.108	0.179	0.063	0.350
Total	0.355	0.452	0.193	1.000

Practice with calculator

The Chi-Square Distribution – Test for Independence

- We are now in a position to test whether the random variables X and Y (number of hours volunteered and type of volunteer) are **independent.**
- If Table A of *observed* counts is not different from Table D of *expected* counts **in the population**, then we can say that X and Y are independent. Otherwise, we reject the null hypothesis.
  - $\mathbf{H_0}$: X and Y are independent; $\mathbf{H_a}$: X and Y are not independent
  - (OR)
  - $\mathbf{H_0}$: Every observed cell count in Table A = Every expected cell count in Table D; $\mathbf{H_a}$: At least one cell count is different
  - (OR)
  - $\mathbf{H_0}$: (Every observed cell count - Every expected cell count) = **0**; $\mathbf{H_a}$: At least one cell count difference ≠ **0**
  - (OR)
  - $\mathbf{H_0}$: Sum of (observed cell count - expected cell count)^2 = **0**; $\mathbf{H_a}$: Sum of squared differences ≠ **0**

Table D - Expected Counts

Type\Hours	1–3	4–6	7–9	Total
Community College	91	115	49	*255*
Four-Year College	103	131	56	*290*
Nonstudents	104	133	57	*294*
Total	*298*	*379*	*162*	*839*

Table A - Observed Counts

Type\Hours	1–3	4–6	7–9	Total
Community College	111	96	48	*255*
Four-Year College	96	133	61	*290*
Nonstudents	91	150	53	*294*
Total	*298*	*379*	*162*	*839*

The Chi-Square Distribution – Test for Independence

- Just as in regression, where we look at squared residuals between observed and predicted (expected) Y values for each X, we square the differences to remove the canceling effect of one residual difference from the other, and we sum the differences to get a single-number that we can check against **0**.

- We can further normalize the squared differences in each cell count through dividing the difference by the expected cell count, and then squaring so that:

  - $\mathbf{H_0}$: Sum of ((observed cell count - expected cell count) ^{2} /(expected cell count)) = **0**;
    $\mathbf{H_a}$: Sum of ((observed cell count - expected cell count) ^{2} /(expected cell count)) ≠ **0**

The Chi-Square Distribution – Test for Independence

- It is now clear that our *test-statistic* will be Sum of (Every observed cell count - Every expected cell count)^2 **from a sample,** where expected cell counts is np.
- Further, if np $\geq$ 5, it can be shown that $\frac{(\text{observed} - \text{expected})}{\sqrt{\text{expected}}}$ will be a *standard normal random variable* especially as n increases (this is called "asymptotically")
- Hence, $\frac{(\text{observed} - \text{expected})^2}{\text{expected}}$ will have a Chi-square distribution wit one degree of freedom $\boldsymbol{\chi}^{\mathbf{2}}_1$
- It can therefore be shown that $\sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$ has a Chi-square distribution with *k* degrees of freedom, $\boldsymbol{\chi}^{\mathbf{2}}_k$
- *k = (number of rows – 1) * (number of columns – 1).*
- Even though we are adding up the normalized squared differences of *(number of rows) * (number of columns)*, the degrees of freedom is lower because the sum of counts for each row and column is a known fixed number (marginal totals) that reduces the degrees of freedom.

Table D - Expected Counts

Type\Hours	1–3	4–6	7–9	Total
Community College	91	115	49	*255*
Four-Year College	103	131	56	*290*
Nonstudents	104	133	57	*294*
Total	*298*	*379*	*162*	*839*

Table A - Observed Counts

Type\Hours	1–3	4–6	7–9	Total
Community College	111	96	48	*255*
Four-Year College	96	133	61	*290*
Nonstudents	91	150	53	*294*
Total	*298*	*379*	*162*	*839*

Practice with calculator

The Chi-Square Distribution – Test for Independence

- *Going back to our problem*: In a volunteer group, adults 21 and older volunteer from one to nine hours each week to spend time with a disabled senior citizen. The program recruits among community college students, four-year college students, and nonstudents. Table A is an ***observed*** sample of the adult volunteers and the number of hours they volunteer per week. Is the number of hours volunteered **independent** of the type of volunteer?

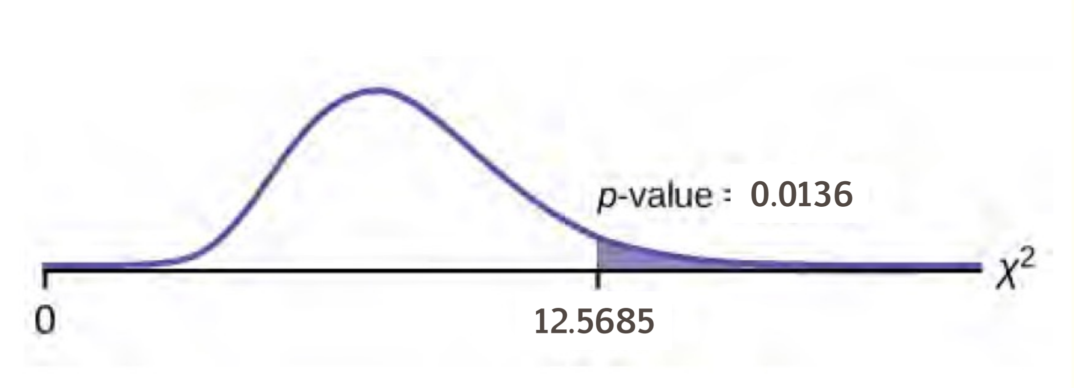
- The table below shows the standardized squared cell differences as the grand total, which will be our test statistic.

- So we can get the p-value from a Chi-square table with (3-1)*(3-1) = 4 degrees of freedom =

```
> print(paste("The probability value > 12.5685,in a Chi-square distribution with (4) df is:"
+             ,round(1 - pchisq(12.5685,4),4)))
[1] "The probability value > 12.5685,in a Chi-square distribution with (4) df is: 0.0136"
```

- Consequently, we will reject the null hypothesis that number of hours volunteered **independent** of the type of volunteer at $\alpha = 0.05$.

Community College	4.3956	3.1391	0.0204	7.5551
Four-Year College	0.4757	0.0305	0.4464	0.9527
Nonstudents	1.6250	2.1729	0.2807	4.0786
Total	6.4963	5.3426	0.7475	12.5865



Practice with calculator

The Chi-Square Distribution – Test for Independence (**Chi-Sq.R**)

- In **R**, we can use *chisq.test()* function.
- The difference in results between expected counts in **R** and our hand calculations is due to rounding.
- X-squared = 12.991, df = 4, p-value = 0.01132

Table D - Expected Counts

Type\Hours	1–3	4–6	7–9	Total
Community College	91	115	49	*255*
Four-Year College	103	131	56	*290*
Nonstudents	104	133	57	*294*
Total	*298*	*379*	*162*	*839*

```
> #
> college <- c("Community College", "Four_Year College", "Non-Students")
> hours <- c("1-3","4-6", "7-9")
> m <- cbind(c(111, 96, 91), c(96, 133, 150), c(48, 61, 53))
> rownames(m) <- college
> colnames(m) <- hours
> print(m)
                  1-3 4-6 7-9
Community College 111  96  48
Four_Year College  96 133  61
Non-Students       91 150  53
> #
> tbl <- as.table(m)
> print(tbl)
                  1-3 4-6 7-9
Community College 111  96  48
Four_Year College  96 133  61
Non-Students       91 150  53
> ch <- chisq.test(tbl)
> print(ch$residuals)
                         1-3        4-6        7-9
Community College  2.1464772 -1.7880604 -0.1763148
Four_Year College -0.6900708  0.1746359  0.6688187
Non-Students      -1.3136852  1.4918030 -0.5000487
> print(ch$expected)
                        1-3      4-6      7-9
Community College  90.57211 115.1907 49.23719
Four_Year College 103.00358 131.0012 55.99523
Non-Students      104.42431 132.8081 56.76758
```

The Chi-Square Distribution – Goodness of Fit

- The idea of $\sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$ to perform hypothesis tests is suitable to many contexts.
- We can therefore use this concept to check how "good" (well) the observed fits the expected, in so-called "**Goodness of Fit**" tests.
- One such test would be to *compare the observed and expected marginal distributions of any discrete random variable* to test the null hypothesis that the observed sample marginal fits the expected population marginal.
- Consider the table of expected "student absence from classes" shown for 100 students.
- We can convert this to an arbitrary distribution a discrete random variable by simply dividing the expected values by 100 (because there are 100 students) to a probability to show the distribution itself, but we don't have to.

Number of absences per term	**Expected number of students**
0–2	50
3–5	30
6–8	12
9–11	6
12+	2

Practice with calculator

The Chi-Square Distribution – Goodness of Fit

- The table of actual (*observed*) number of absences for a group of 100 students is shown, along with the expected.

- The sum $\sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}} = 22.33$ has a Chi-square distribution with (k-1) = 4 degrees of freedom, and can be used to test:
  - **H_0**: Student absenteeism fits faculty perception.
  - **H_a**: Student absenteeism does not fit faculty perception.

- The **p-value** for our Chi-square test statistic with 4 degrees of freedom is: = 0.0002

```
> print(round(1 - pchisq(22.33,4),4))
[1] 2e-04
```

- This leads to a **rejecting** the null hypothesis "Student absenteeism fits faculty perception".

- In **R**, the chisq.test() function performs a goodness of fit test by comparing the relative frequency of actual counts against expected probabilities.

- **R** gives a warning when the expected cell counts are too small (np must be > 5 in normal approximation to binomial), and therefore its Chi-square statistic may not be accurate.

Number of absences per term	Actual number of students	Expected number of students
0–2	35	50
3–5	40	30
6–8	20	12
9–11	1	6
12+	4	2

Number of absences per term	Actual number of students	Probability
0–2	35	0.5
3–5	40	0.3
6–8	20	0.12
9–11	1	0.06
12+	4	0.02

```
> # The Chi-square Goodness of Fit
> #
> absences <- c("0-2", "3-5", "6-8","9-11", "12+")
> expected <- c(50,30,12,6,2)
> probab <- expected/sum(expected)
> print(probab)
[1] 0.50 0.30 0.12 0.06 0.02
> actual <- c(35, 40, 20, 1, 4)
> chisq.test(actual,p = probab)

        Chi-squared test for given probabilities

data:  actual
X-squared = 19.333, df = 4, p-value = 0.0006758

Warning message:
In chisq.test(actual, p = probab) :
  Chi-squared approximation may be incorrect
>
```

The Chi-Square Distribution – Goodness of Fit (Book Example 11.4)

Practice with calculator

- *Suppose you flip two coins 100 times. The results are 20 HH, 27 HT, 30 TH, and 23 TT. Are the coins fair? Test at a 5% significance level.*

- Let X = the **number of heads** in one flip of the two coins. X takes on the values 0, 1, 2. (There are 0, 1, or 2 heads in the flip of two coins.) Therefore, the **number of cells is three**.

- Since X = the number of heads, the *observed* frequencies are 20 (for two heads), 57 (for one head), and 23 (for zero heads or both tails).

- The *expected* frequencies are 25 (for two heads), 50 (for one head), and 25 (for zero heads or both tails) under the null hypothesis that both coins are fair.

- So, our test statistic = $(20\text{-}25)^2/25 + (57\text{-}50)^2/50 + (23\text{-}25)^2/25$ = **2.14** has a Chi-square distribution with (k-1) = 2 degrees of freedom, and can be used to test:
  - **H_0**: Both the coins are fair.
  - **H_a**: At least one coin is not fair.

- The **p-value** for our Chi-square test statistic with 2 degrees of freedom is: = 0.3430 (because the Chi-square test is upper-

	2H	1H	0H
Observed	20	57	23
Expected	25	50	25

```
> observed <- c(20, 57, 23)
> expected_prob <- c(.25, .5, .25)
> chisq.test(observed,p = expected_prob)

        Chi-squared test for given probabilities

data:  observed
X-squared = 2.14, df = 2, p-value = 0.343

> print(paste("The probability value > 2.14,in a Chi-square distribution with (2) df is:"
+             ,round(1 - pchisq(2.14,2),4)))
[1] "The probability value > 2.14,in a Chi-square distribution with (2) df is: 0.343"
> 
```

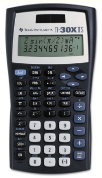
```
> print(paste("The probability value > 2.14,in a Chi-square distribution with (2) df is:"
+             ,round(1 - pchisq(2.14,2),4)))
[1] "The probability value > 2.14,in a Chi-square distribution with (2) df is: 0.343"
```

The Chi-Square Distribution – Test for Homogeneity

- The **test for homogeneity**, can be used to draw a conclusion about whether *two populations have the same distribution*.
- To calculate the test statistic for a test for homogeneity, follow the same procedure as with the test of independence.
- **Note:** The expected value for each cell needs to be at least five in order for you to use this test.
- The *degrees of freedom* = (number of columns – 1)

The Chi-Square Distribution – Test for Homogeneity (Book Example 11.8)

Practice with calculator



- Do male and female college students have the same distribution of living arrangements? Use a level of significance of 0.05. Suppose that 250 randomly selected male college students and 300 randomly selected female college students were asked about their living arrangements: dormitory, apartment, with parents, other. The results are shown in **Table**. Do male and female college students have the same distribution of living arrangements?
  - **H_0**: The distribution of living arrangements for male college students is the same as the distribution of living arrangements for female college students.
  - **H_a**: The distribution of living arrangements for male college students is not the same as the distribution of living arrangements for female college students.
- If we view it exactly like a contingency table we can convert it to a test for **independence between Gender and Living Arrangement** so that:
  - **H_0**: Living arrangement *is* independent of Gender for college
  - **H_a**: Living arrangement is *not* independent of Gender for college
- df = (number of rows – 1)*(number of columns – 1) = 3

	Dormitory	Apartment	With Parents	Other
Males	72	84	49	45
Females	91	86	88	35

Practice with calculator

The Chi-Square Distribution – Test for Homogeneity

- If we view it exactly like a contingency table we can convert it to a test for **independence between Gender and Living Arrangement** so that:
  - H_0: Living arrangement *is* independent of Gender for college students
  - H_a: Living arrangement is *dependent on* Gender for college students
- *df* = (number of rows – 1)*(number of columns – 1) = 3
- The **p-value** for our test statistic of 10.1287 with 3 degrees of freedom = 0.0175
- At 0.05 level of significance we **reject** the null hypothesis and conclude Living arrangement is *dependent on* Gender for college students

Observed

	Dorm	Parent	Apartment	Other	
Male	72	84	49	45	**250**
Female	91	86	88	35	**300**
	163	**170**	**137**	**80**	**550**

Expected

	Dorm	Parent	Apartment	Other	
Male	74.09091	77.27273	62.27273	36.36364	**250**
Female	88.90909	92.72727	74.72727	43.63636	**300**
	163	**170**	**137**	**80**	**550**

(Obs-Exp)^2/Exp

	Dorm	Parent	Apartment	Other	
Male	0.059007	0.585668	2.828932	2.051136	**5.524744**
Female	0.049173	0.488057	2.357443	1.70928	**4.603953**
					10.1287

```
> #
> # Chi-Square Test for Homogeneity
> #
> male <- c(72, 84, 49, 45)
> female <- c(91, 86, 88, 35)
> tbl <- cbind(male, female)
> chisq.test(tbl)

        Pearson's Chi-squared test

data:  tbl
X-squared = 10.129, df = 3, p-value = 0.0175
```

LECTURE 5 – PART B – BINOMIAL (DICHOTOMOUS) LOGISTIC REGRESSION

Types of Variables and Basic Types of Analysis

Dependent	Outcome	Predictor	Analysis	Estimation Method
Ratio (Continuous)	Value Prediction	Categorical Only	ANOVA (Regression)	Least Squares
Ratio (Continuous)	Value Prediction	Categorical and/or Continuous	Regression	Least Squares
Nominal (Categorical)	Association (Dependence)	Categorical Only	Contingency Table	Chi-Square
Nominal (Categorical)	Category Value Probability, Classification	Categorical and/or Continuous	Logistic Regression	Maximum Likelihood

Logistic Regression (**LogisticReg.R**)

- The technique of logistic regression is used when the dependent variable is dichotomous, ordinal or multinomial, rather than continuous.

- **Example**: A researcher is interested in how variables, such as GRE (Graduate Record Exam scores), GPA (grade point average) and *prestige of the undergraduate institution*, effect **admission into graduate school**. Rank = 1 indicates undergraduate institution of the highest rank or prestige.

- The response or dependent variable, *admit/don't admit*, is a ***binary*** variable.

- We note right away that
  1. Y is not normally distributed, since it is *not continuous*, violating an important regression assumption.
  2. Heteroscedasticity may be common in this type of data set.
  3. Using the usual multiple regression may not guarantee valid values for Y.

- Let us fit a linear regression model to the data set Admit-Reg.csv in Canvas under Lecture 5.

  (Source: https://stats.idre.ucla.edu/r/dae/logit-regression/)

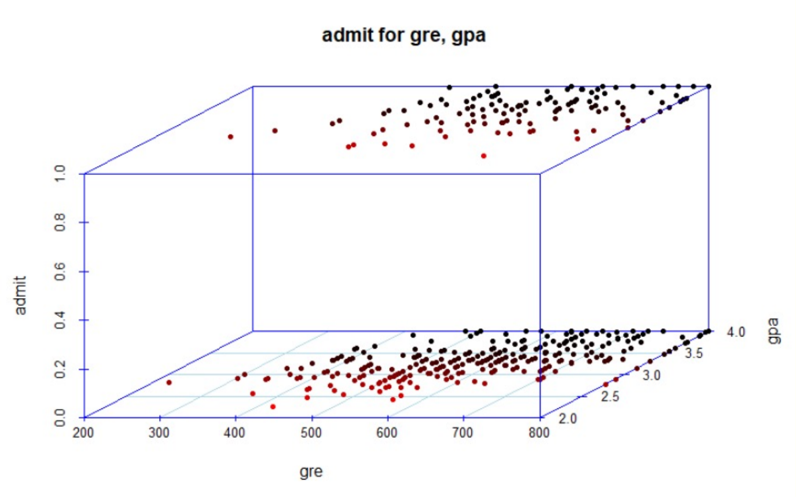
- Our equation becomes:
  - $\widehat{\text{admit}} = -.182 + 0.000\text{gre} + 0.151\text{gpa} - .110\text{rank}$

```
  admit gre  gpa rank
1     0 380 3.61    3
2     1 660 3.67    3
3     1 800 4.00    1
4     1 640 3.19    4
5     0 520 2.93    4
6     1 760 3.00    2
```

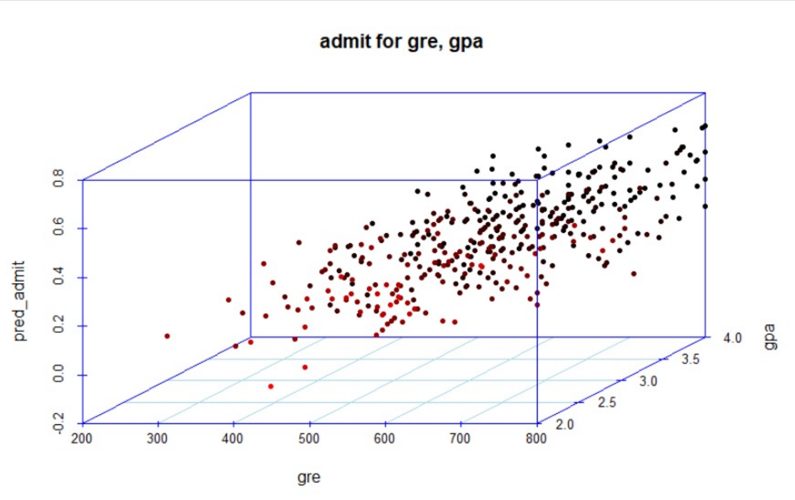
```
> admit <- df$admit
> gre <- df$gre
> gpa <- df$gpa
> rank <- df$rank
> mod1 <- lm(admit ~ gre+gpa+rank)
> summary(mod1)

Call:
lm(formula = admit ~ gre + gpa + rank)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.6617 -0.3417 -0.1947  0.5061  0.9556 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.1824127  0.2169695  -0.841   0.4010    
gre          0.0004424  0.0002101   2.106   0.0358 *  
gpa          0.1510402  0.0633854   2.383   0.0176 *  
rank        -0.1095019  0.0237617  -4.608 5.48e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4448 on 396 degrees of freedom
Multiple R-squared:  0.09601,	Adjusted R-squared:  0.08916 
F-statistic: 14.02 on 3 and 396 DF,  p-value: 1.054e-08
```

Logistic Regression (**LogisticReg.R**)

```
pred_admit <- predict(mod1)
#
# install.packages("scatterplot3d")
library(scatterplot3d)

scatterplot3d(gre, gpa, admit, highlight.3d=TRUE, col.axis="blue",
              col.grid="lightblue", main="admit for gre, gpa", pch=20)
scatterplot3d(gre, gpa, pred_admit, highlight.3d=TRUE, col.axis="blue",
              col.grid="lightblue", main="admit for gre, gpa", pch=20)
```



- The scatterplot of actual admit vs gre and gpa (we did not put in rank, but that does not matter) shows two separate planes corresponding to admit = 1 and admit = 0.
- The scatterplot of *predicted* admit vs gra and gpa includes values for admit between 0 and 1
- It is clear, that using multiple regression, while violating assumptions, also will produce predicted values that may make no sense, because values of admit between 0 and 1 are meaningless.



Logistic Regression

- If we re-think our problem we realize that we are not so much interested in the values of admit (Y variable) as the *probability* of admit or not.
- Thus, given values of the predictors (such as gpa gre and rank) we want to predict the *probability* that a student will be admitted or not.
- In other words, *we need a transformation* of the dependent variable from values to probabilities (of "success" or "failure").
- But, multiple regression can only predict values of a variable that is assumed normally distributed and can range from –infinity to +infinity.
- Because, probabilities can only range from 0 to 1, we instead use a *double transformation*. One, we predict something called *odds ratio* (that can range from –infinity to +infinity) and then transform odds ratios to probabilities.

Logistic Regression

- The odds of success are defined as the ratio of:
  - $\frac{\text{Probability of Success}}{\text{Probability of Failure}}$.
  - Let's say that the probability of success of some event is .8. Then the probability of failure is 1- .8 = .2. In our example, the odds of success are .8/.2 = 4. That is to say that the odds of success are 4 to 1.
  - If the probability of success is .5, i.e., 50-50 percent chance, then the odds of success is 1 to 1.

- The transformation from probability to odds is a *monotonic transformation*, meaning the odds increase as the probability increases or vice versa. Probability ranges from 0 and 1. Odds range from 0 and positive infinity.

- However, we can do even better, by using the logarithm of the odds ratio, because for probabilities between 0 and 1 we get a transformation that provides values between –infinity and +infinity

- The other advantage is that log odds provides easy interpretations. Positive log odds means that the probability of success is greater than then the probability of failure.

- Thus, the transformation function we are looking for is the *logit link function* that uses a linear regression model to predict log odds.

p	odds	logodds
.001	.001001	-6.906755
.01	.010101	-4.59512
.15	.1764706	-1.734601
.2	.25	-1.386294
.25	.3333333	-1.098612
.3	.4285714	-.8472978
.35	.5384616	-.6190392
.4	.6666667	-.4054651
.45	.8181818	-.2006707
.5	1	0
.55	1.222222	.2006707
.6	1.5	.4054651
.65	1.857143	.6190392
.7	2.333333	.8472978
.75	3	1.098612
.8	4	1.386294
.85	5.666667	1.734601
.9	9	2.197225
.999	999	6.906755
.9999	9999	9.21024

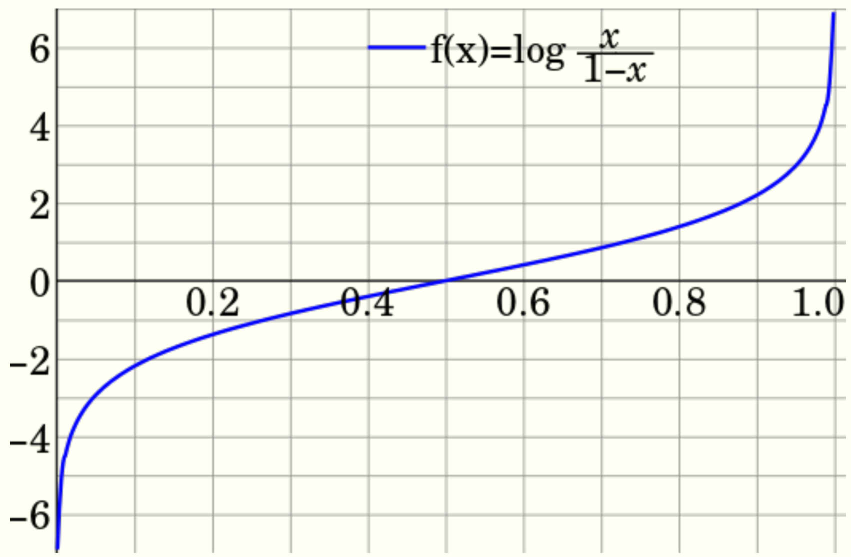
https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-how-do-i-interpret-odds-ratios-in-logistic-regression/

Logistic Regression

- The *logarithm of odds (of success)* = $\alpha + \sum_i^k \beta_i X_i$ and is called the *logit link function.*
- That is, $\log\left(\frac{\text{Probability of Success}}{\text{Probability of Failure}}\right) = \alpha + \sum_i^k \beta_i X_i$ or $\log\left(\frac{\text{Probability of Success}}{1 - \text{Probability of Success}}\right) = \alpha + \sum_i^k \beta_i X_i$.
- We now have a regular multiple regression linear model where the dependent variable, instead of being the Probability of Success is the Log Odds of Success, but can range from –infinity to +infinity, is interpretable, and can easily be translated back to valid probabilities.
- We can see that: $P(X = x) = \frac{e^{\alpha + \sum_i^k \beta_i x_i}}{1+e^{\alpha + \sum_i^k \beta_i x_i}}$, where k is the number of independent variables; α, β_ι are unknown parameters.
- This function $(P(X = x))$ is called the *logistic function* and is the inverse of the *logit link function.*
- Probability of "failure" = 1 – P(X=x) = $\frac{1}{1+e^{\alpha + \sum_i^k \beta_i x_i}}$
- Odds = P("Success")/P("Failure") = $e^{\alpha + \sum_i^k \beta_i x_i}$

Logistic Regression

- The logit link function has the shape shown in the figure to the right.
- The logit link function provides a nice interpretation for the betas.
- We have the logit link function $\ln \frac{P(X=x)}{1-P(X=x)} = \alpha + \sum_{i}^{k} \beta_i X_i$.
- Then, $\beta_i = \ln(\frac{P(X=x+1)}{1- F(x+1)})$ - $\ln(\frac{P(X=x)}{1-P(X=x)})$
- Thus, β is the change in log odds ratio for (Y="success") for a unit increase in X_i holding all other X's constant (controlling for them).
- When β is positive, increases in that variable will increase the log-odds ratio for (Y="success") and therefore increase the probability of success, controlling for other variables (i.e., the correlation of other X variables with that X).



Logistic Regression (**LogisticReg.R**)

- Let us carry out Logistic Regression for our example of admit/non-admit and interpret the solution.

- In **R**, we run logistic regression using the **general linear model** *glm()* function rather than the lm() function. The family parameter is set to *family="binomial"* indicating that the dependent variable admit is a binary variable.

- **Interpretation**:

  Because their betas are positive, increasing gre scores or gpa, improves the probability for success (admit). With rank, increase in rank decreases probability of success (admit). This makes sense, because low rank values actually mean the student was a better student than one with a higher value.

- The equation for **<u>log-odds</u>** is:
  - -3.450 + 0.0023*gre + 0.7770*gpa – 0.5600*rank
  - When gre, rank and gpa are 0, then the log odds of admit is the intercept -3.450. This corresponds to a base P(admit) of about 0.52
  - The change in the log odds ratio of admit, for unit change in gre, is 0.002, controlling for gpa and rank.

```
> # Logistic Regression
> #
> logimod1 <- glm(admit ~ gre + gpa + rank, data = df, family = "binomial")
> summary(logimod1)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5802  -0.8848  -0.6382   1.1575   2.1732

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.449549   1.132846  -3.045  0.00233 **
gre          0.002294   0.001092   2.101  0.03564 *
gpa          0.777014   0.327484   2.373  0.01766 *
rank        -0.560031   0.127137  -4.405 1.06e-05 ***
---
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: 459.44  on 396  degrees of freedom
AIC: 467.44

Number of Fisher Scoring iterations: 4
```

Practice with calculator

Logistic Regression (LogisticReg.R)

- **Prediction**:
- The equation for **log-odds** is:
  - -3.450 + 0.0023*gre + 0.7770*gpa – 0.5600*rank
  - Suppose GRE was 800, GPA was 4 and rank was 1, then our prediction for **log-odds** would be:
    - -3.449 + 0.0023*(800) + (0.777)*4 – (0.56)*1 = 0.939
- Hence, predicted **log-odds** = 0.9336
- Hence, predicted **odds-ratio** = exp(log-odds) = exp(0.9336) = 2.54
- Hence, **probability of admit** ("success") = 2.54/3.54 = 0.718
- This is called "Predicted Probability" and given by:
- $\frac{e^{-3.45+0.002gre+0.777gpa-0.560rank}}{1+e^{-3.45+0.002gre+0.777gpa-0.560rank}}$

```
> # Logistic Regression
> #
> logimod1 <- glm(admit ~ gre + gpa + rank, data = df, family = "binomial")
> summary(logimod1)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5802  -0.8848  -0.6382   1.1575   2.1732

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.449549   1.132846  -3.045  0.00233 **
gre          0.002294   0.001092   2.101  0.03564 *
gpa          0.777014   0.327484   2.373  0.01766 *
rank        -0.560031   0.127137  -4.405 1.06e-05 ***
---
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: 459.44  on 396  degrees of freedom
AIC: 467.44

Number of Fisher Scoring iterations: 4
```

```
> df_a <- data.frame(gre=800, gpa=4, rank=1)
> p_log_odds <- predict(logimod1, df_a)
> p_odds <- exp(p_log_odds)
> p_prob <- (p_odds/(1+p_odds))
> #
> print(paste("Predicted Log Odds = ", round(p_log_odds,4),
+             "Predicted Odds Ratio = ", round(p_odds,4),
+             "Predicted Probability of Success = ",round(p_prob,4)))
[1] "Predicted Log Odds =  0.9336 Predicted Odds Ratio =  2.5438 Predicted Probability of Success =  0.7178"
```

Logistic Regression (**LogisticReg.R**)

- We can develop predicted probabilities of admit for all the observations.

- We show the predicted probabilities of admit for the first 6 observations

- In **R**, you can use the *glm.probs()* function to get the predicted probabilities

```
> #
> p_log_odds <- predict(logimod1)
> p_odds <- exp(p_log_odds)
> prob_admit <- (p_odds/(1+p_odds))
> df_pred <- data.frame(gre, gpa, rank, admit, prob_admit)
> print(head(df_pred))
  gre  gpa rank admit prob_admit
1 380 3.61    3     0 0.18955274
2 660 3.67    3     1 0.31778076
3 800 4.00    1     1 0.71781361
4 640 3.19    4     1 0.14894920
5 520 2.93    4     0 0.09795421
6 760 3.00    2     1 0.37867846
```

```
> #
> glm.probs <- predict(logimod1,type = "response")
> glm.probs[1:5]
         1          2          3          4          5 
0.18955274 0.31778076 0.71781361 0.14894920 0.09795421 
```

Logistic Regression

- Hypothesis Testing of Logistic Regression Coefficients
  - Large-Sample test (**Wald Test**):
  - $\mathbf{H_0}$: $\beta_i = 0$; $\mathbf{H_a}$: $\beta_i \neq 0$
- The Wald test is a standard normal distribution where the test statistic is given by:
  - $z = \frac{\hat{\beta}_i}{\sigma_{\hat{\beta}_i}}$, where $\hat{\beta}_i$ is the parameter estimate shown in the printout and $\sigma_{\hat{\beta}_i}$ is the standard error of the estimate, see printout.
  - The df and **p-values** are also shown
- In our case, we see that at $\alpha = 0.05$, all the predictors are significant.

```
> # Logistic Regression
> #
> logimod1 <- glm(admit ~ gre + gpa + rank, data = df, family = "binomial")
> summary(logimod1)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5802  -0.8848  -0.6382   1.1575   2.1732

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.449549   1.132846  -3.045  0.00233 **
gre          0.002294   0.001092   2.101  0.03564 *
gpa          0.777014   0.327484   2.373  0.01766 *
rank        -0.560031   0.127137  -4.405 1.06e-05 ***
---
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: 459.44  on 396  degrees of freedom
AIC: 467.44

Number of Fisher Scoring iterations: 4
```

Using Logistic Regression for Classification

- Earlier we saw how to obtain the Predicted Probability (of Admit)
- We can convert this to a classification problem by calculating the Predicted Value for each case. The predicted value is obtained by assuming that if predicted P(Admit) ≥ 0.5, we will assign a value of 1, else 0.
- This reduces it to a classification problem.
- In **R**, we first use the *ifelse()* function to create the predicted admit class with P(Admit) >=0.5
- Then, we use the *xtabs()* function to create a cross-classification table to actual admit (0,1) vs. predicted admit (0,1). The counts are then converted to percentages.
- In our case the correct classification rate of the model is 70.5%
- Clearly, this is one measure of the predictive ability of our model

```
> # Classification Using Logistic Regression
> #
> pred_admit <- ifelse(glm.probs >= 0.5,1,0)
> df_class <- cbind(admit, pred_admit)
> print(head(df_class))
  admit pred_admit
1     0          0
2     1          0
3     1          1
4     1          0
5     0          0
6     1          0
> class_tbl <- xtabs(~ admit + pred_admit, data = df_class)
> class_pct <- class_tbl/length(admit)
> print(class_tbl)
     pred_admit
admit   0   1
    0 253  20
    1  98  29
> print(class_pct)
     pred_admit
admit      0      1
    0 0.6325 0.0500
    1 0.2450 0.0725
> print(paste("Correct classification Rate (percent): ",
+             (class_pct[1,1] + class_pct[2,2])*100))
[1] "Correct classification Rate (percent):  70.5"
>
```

Logistic Regression

- How were the model coefficients estimated? Unlike ordinary least squares (OLS) regression, the method of estimation used in Logistic Regression is *Maximum Likelihood Estimation (MLE)*.
- The theoretical underpinnings of MLE are beyond the scope of this class. However, here are some important characteristics of MLE.
  - MLE also assumed that the error term is normally distributed in a manner similar to OLS
  - MLE is also sensitive to outliers and influential observations
  - MLE also assumes that there is no serious Multicollinearity problem
  - MLE estimates all the coefficients simultaneously and *iteratively*, beginning with a starting solution.
  - The iteration stops, giving us the estimates for the coefficients, when a certain *tolerance* is reached
  - Standard errors of the estimates may be obtained by bootstrapping.
  - Unlike Least Squares estimation, where a solution to parameter estimation is always guaranteed, in rare cases the estimation process may not converge and we may not get solutions. This could also be because of poor start values.

Practice with calculator

Logistic Regression with Continuous and Categorical Predictors

- In the previous example, rank which was an ordinal variable, was treated as a continuous variable.
- We can re-run the logistic regression by treating rank as a <u>categorical variable</u>.
- In **R**, we convert rank into a factor, frank.
- The equation for <u>**log-odds**</u> of predicted admit is:
  - -3.999 + 0.0023*gre + 0.804*gpa – 0.675*(frank="2")
    -1.340*(frank="3") -1.551*(frank="4")
  - Suppose GRE was 800, GPA was 4 and frank was 1,
    then our prediction for log-odds would be:
    - -3.999 + 0.0023*(800) + (0.804)*4 (Since frank = 1)
      all the other rank variables will be 0
- All other calculations will be as before.

```
> #
> # Using rank as a categorical variable or factor
>
> df$frank <- factor(df$rank)
> logimod2 <- glm(admit ~ gre + gpa + frank, data = df, family = "binomial")
> summary(logimod2)

Call:
glm(formula = admit ~ gre + gpa + frank, family = "binomial",
    data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6268  -0.8662  -0.6388   1.1490   2.0790

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979   1.139951  -3.500 0.000465 ***
gre          0.002264   0.001094   2.070 0.038465 *
gpa          0.804038   0.331819   2.423 0.015388 *
frank2      -0.675443   0.316490  -2.134 0.032829 *
frank3      -1.340204   0.345306  -3.881 0.000104 ***
frank4      -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
```

Logistic Regression with Continuous and Categorical Predictors

- **Interpreting Coefficients**:
- The output shows the coefficients, their standard errors, the z-statistic (sometimes called a Wald z-statistic), and the associated p-values.
- Both gre and gpa are statistically significant, as are the three terms for rank.
- The logistic regression coefficients give the change in the **log odds** of the outcome for a one unit increase in the predictor variable.
- For every one unit change in gre, the log odds of admission (versus non-admission) increases by 0.002 (with the gpa in the model, for all of the values of rank)
- For a one unit increase in gpa, the log odds of being admitted to graduate school increases by 0.804 (with the gre in the model, for all of the values of rank).
- The indicator variables for rank have a slightly different interpretation. For example, having attended an undergraduate institution with rank of 2, versus an institution with a rank of 1, changes the log odds of admission by -0.675.

```
> #
> # Using rank as a categorical variable or factor
>
> df$frank <- factor(df$rank)
> logimod2 <- glm(admit ~ gre + gpa + frank, data = df, family = "binomial")
> summary(logimod2)

Call:
glm(formula = admit ~ gre + gpa + frank, family = "binomial",
    data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6268  -0.8662  -0.6388   1.1490   2.0790

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979   1.139951  -3.500 0.000465 ***
gre          0.002264   0.001094   2.070 0.038465 *
gpa          0.804038   0.331819   2.423 0.015388 *
frank2      -0.675443   0.316490  -2.134 0.032829 *
frank3      -1.340204   0.345306  -3.881 0.000104 ***
frank4      -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
```

Practice with calculator

Logistic Regression with Continuous and Categorical Predictors

- Predicted Log-odds of Admit = -3.999 + 0.0023*gre + 0.804*gpa – 0.675*(frank="2") -1.340*(frank="3") -1.551*(frank="4")
- Equations for **<u>log-odds</u>** of predicted admit :
  - -3.999 + 0.0023*gre + 0.804*gpa (schools with rank of 1)
  - -4.674 + 0.0023*gre + 0.804*gpa (schools with rank of 2 i.e., frank2)
  - -5339 + 0.0023*gre + 0.804*gpa (schools with rank of 3 i.e., frank3)
  - -5.55 + 0.0023*gre + 0.804*gpa (schools with rank of 4 i.e., frank4)
- Given a gre score of 800 and a gpa of 3.5, let us calculate the predicted probabilities of being admitted to graduate school from under-graduate institutions of each rank (lower value of rank, higher prestige).

```
> df_s <- with(df, data.frame(gre =800, gpa = 3.5, frank = factor(1:4)))
> prob_admit <- predict(logimod2,df_s, type = "response")
> print(paste("Probability of admit in school with different ranks ", round(prob_admit,4)))
[1] "Probability of admit in school with different ranks  0.6538"
[2] "Probability of admit in school with different ranks  0.4901"
[3] "Probability of admit in school with different ranks  0.3308"
[4] "Probability of admit in school with different ranks  0.2858"
```

Logistic Regression Model Fit Concepts

- We may also wish to see measures of how well our model fits.
- This can be particularly useful when comparing competing models.
- The output produced by *summary(logimod2)* includes indices of fit (shown below the coefficients), including the *null and deviance residuals* and the *AIC*.
- These are badness of fit measures in that we want the numbers to be smaller…smaller numbers mean a better model.
- One measure of model fit is the significance of the overall model. This test asks whether the model with predictors fits significantly better than a model with *just an intercept* (i.e., a *null model*).

```
> #
> # Using rank as a categorical variable or factor
>
> df$frank <- factor(df$rank)
> logimod2 <- glm(admit ~ gre + gpa + frank, data = df, family = "binomial")
> summary(logimod2)

Call:
glm(formula = admit ~ gre + gpa + frank, family = "binomial",
    data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6268  -0.8662  -0.6388   1.1490   2.0790

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979   1.139951  -3.500 0.000465 ***
gre          0.002264   0.001094   2.070 0.038465 *
gpa          0.804038   0.331819   2.423 0.015388 *
frank2      -0.675443   0.316490  -2.134 0.032829 *
frank3      -1.340204   0.345306  -3.881 0.000104 ***
frank4      -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
```

Logistic Regression Model Fit Concepts

- The test statistic is the *difference between the residual deviance for the model with predictors and the null model.*

- The test statistic is distributed **chi-squared** with *degrees of freedom* = degrees of freedom between the current model – degrees of freedom for the null model (i.e., the number of predictor variables in the model).

- To perform the test of the difference in deviance for the two models (i.e., the test statistic) we can use the command:

```
> # Overall Model Test using Chi-Square Test
> #
> with(logimod2, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 7.57819e-08
```

- The p-value near 0 tells us that our model with 5 predictors is significantly better (at α = 0.05) than the null model with no predictors except the intercept.

```
> #
> # Using rank as a categorical variable or factor
>
> df$frank <- factor(df$rank)
> logimod2 <- glm(admit ~ gre + gpa + frank, data = df, family = "binomial")
> summary(logimod2)

Call:
glm(formula = admit ~ gre + gpa + frank, family = "binomial",
    data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6268  -0.8662  -0.6388   1.1490   2.0790

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979   1.139951  -3.500 0.000465 ***
gre          0.002264   0.001094   2.070 0.038465 *
gpa          0.804038   0.331819   2.423 0.015388 *
frank2      -0.675443   0.316490  -2.134 0.032829 *
frank3      -1.340204   0.345306  -3.881 0.000104 ***
frank4      -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
```

LECTURE 5 – PART C – MULTINOMIAL (POLYTOMOUS) LOGISTIC REGRESSION

Logistic Multinomial (Polytomous) Regression

- In our previous example, our dependent variable was dichotomous (two categories; admit/do not admit) and assumed to have a Binomial distribution for the number of cases in each of two categories. It is also called *dichotomous* logistic regression.
- A logical extension is to the case where the dependent variable is *polytomous* (more than two categories) and assumed to have a multinomial distribution for the number of cases in each category.
- The underlying principles and transformations are the same and are extensions of the dichotomous case.
- The interpretations require more care.

https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/

Logistic Multinomial (Polytomous) Regression

- We will take the example of a data set (Look for the data set hsdemo.csv in Canvas under Lecture 5)
  - We are trying to predict/classify cases into 3 program (prog) categories: general, academic, vocational, using socioeconomic status (ses) of the student and writing scores (write).
  - ses is a categorical variable, but write is a continuous predictor.
- Thus, our model has a polytomous dependent variable, (prog) a categorical predictor (ses) and a continuous predictor (write) (sometimes called *covariate*)

	id	female	ses	schtyp	prog	read	write	math	science	socst	honors	awards	cid
1	45	female	low	public	vocation	34	35	41	29	26	not enrolled	0	1
2	108	male	middle	public	general	34	33	41	36	36	not enrolled	0	1
3	15	male	high	public	vocation	39	39	44	26	42	not enrolled	0	1
4	67	male	low	public	vocation	37	37	42	33	32	not enrolled	0	1
5	153	male	middle	public	vocation	39	31	40	39	51	not enrolled	0	1
6	51	female	high	public	general	42	36	42	31	39	not enrolled	0	1

https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/

Logistic Multinomial (Polytomous) Regression

- As a preliminary analysis let us use cross-tabulate to obtain a contingency table for prog and ses.
- The contingency table Chi-square test shows that prog is not independent (that is, it *is* dependent) on ses. The program that a student enters issignificantly dependent on their socio-economic status.
- However, **write** is a continuous variable and we want to be able to predict classify a program that a student would go into, based on ses and write score.

```
> tbl <- with(dfm, table(ses, prog))
> chisq.test(tbl)

        Pearson's Chi-squared test

data:  tbl
X-squared = 16.604, df = 4, p-value = 0.002307
```

Logistic Multinomial (Polytomous) Regression

- In **R**, install.packages("nnet")
- Use the *multinom()* function from library(nnet)
- Multinomial regression requires one of the categories of the independent variable prog to be the *reference level*.
- We will choose academic as the reference level for **prog** and "low" as the reference level for **ses**.
- The log-odds of the other categories (general and vocation) are expressed relative to entering a academic school.
- The relevel() function in library(nnet) is used to establish the reference category.
- The levels of **prog** are re-ordered so that vocation is first and the others are moved down.
- The levels of **ses** are re-ordered so that low is first and the others are moved down.

```
> # Set academic as reference category for multinomial dependent variable prog
> #
> prog <- relevel(prog, ref = "academic")
> #
> # Set low as reference category for predictor variable ses
> #
> ses <- relevel(ses, ref = "low")
```

Practice with calculator

Logistic Multinomial (Polytomous) Regression

- We will start with parameter estimates.
- For equation 1, program=general is taken as "success" and prog = academic is taken as "failure" and the prediction model is fitted for the log odds:
- $\mathbf{ln}\left(\frac{\mathbf{P(general)}}{\mathbf{P(academic)}}\right) = \mathbf{2.8522 - 0.0579write - 0.5333\ (ses = middle) - 1.1628(ses = high)}$
- For equation 2, program= vocation is taken as "success" and prog = academic is taken as "failure" and the prediction model is fitted for the log odds:
- $\mathbf{ln}\left(\frac{\mathbf{P(vocation)}}{\mathbf{P(academic)}}\right) = \mathbf{5.2183 - 0.1136write + 0.2914(ses = middle) - 0.9827(ses = high)}$
- Controlling for **ses**, one unit increase in writing score (**write)** results in
  - a .0579 **decrease** in the *log odds* of being in general versus academic program
  - a 0.1136 **decrease** in the *log odds* of being in vocation versus academic program
- Controlling for **write** score,
  - The relative log odds of being in general versus academic program will **decrease** by 0.5333, if moving from (ses = low) to (ses = medium) and **decrease** by 1.1628 if moving from (ses = low) to (ses = high)
  - The relative log odds of being in vocation versus academic program will **increase** by 0.2914, if moving from (ses = low) to (ses = medium) and

```
> # Specify the polytomous logistic regression model
> #
> mmod1 <- multinom(prog ~ ses + write)
# weights:  15 (8 variable)
initial  value 219.722458
iter  10 value 179.982880
final  value 179.981726
converged
> #
> # Model summary
> summary(mmod1)
Call:
multinom(formula = prog ~ ses + write)

Coefficients:
         (Intercept)    seshigh  sesmiddle       write
general     2.852198 -1.1628226 -0.5332810 -0.0579287
vocation    5.218260 -0.9826649  0.2913859 -0.1136037

Std. Errors:
         (Intercept)   seshigh sesmiddle      write
general     1.166441 0.5142196 0.4437323 0.02141097
vocation    1.163552 0.5955665 0.4763739 0.02221996

Residual Deviance: 359.9635
AIC: 375.9635
```

Logistic Multinomial (Polytomous) Regression

- We can test the significance of the coefficients by extracting the coefficient estimates and standard errors, and then using the Wald test (z-test).

- The output tells us that at $\alpha = 0.05$,

- (the slope parameter estimate of) **write** is significant for *both* prog = general and prog = vocation, relative to academic.

- for prog = general relative to academic, ses=(from low to) middle is not significant but ses= (from low to) high are both significant. That is, only moving from ses=low to ses=high makes a significant impact on the log-odds of being in general vs being in academic program (it decreases it by -2.261).

- for prog = vocation relative to academic, neither ses=(from low to) middle nor ses= (from low to) high are significant. That is moving up in the ses category levels has no significant impact on the log-odds of being in vocation vs being in an academic program.

```
> # Wald's Test of Coefficients
> #
> z_value <- summary(mmod1)$coefficients/summary(mmod1)$standard.errors
> print(z_value)
         (Intercept)   seshigh  sesmiddle     write
general     2.445214 -2.261334 -1.2018081 -2.705562
vocation    4.484769 -1.649967  0.6116747 -5.112689
> p_value <- (1 - pnorm(abs(z), 0, 1)) * 2
> print(p_value)
          (Intercept) dfm$ses2high dfm$ses2middle        write
general  0.0144766100   0.02373856      0.2294379 6.818902e-03
vocation 0.0000072993   0.09894976      0.5407530 3.176045e-07
> #
```

```
> # Model summary
> summary(mmod1)
Call:
multinom(formula = prog ~ ses + write)

Coefficients:
         (Intercept)    seshigh  sesmiddle      write
general     2.852198 -1.1628226 -0.5332810 -0.0579287
vocation    5.218260 -0.9826649  0.2913859 -0.1136037

Std. Errors:
         (Intercept)   seshigh sesmiddle      write
general     1.166441 0.5142196 0.4437323 0.02141097
vocation    1.163552 0.5955665 0.4763739 0.02221996

Residual Deviance: 359.9635
AIC: 375.9635
```

Practice with calculator

Predicted Probabilities

- The predicted probability of being in the three different programs when write = 40, ses = low.

- **Calculating by Hand:**
  - $\ln\left(\frac{P(\text{general})}{P(\text{academic})}\right) = 2.8522 - 0.0579\text{write}$
    - ~~0.5333 (ses = middle) – 1.1628(ses = high)~~ (because ses=low)
  - $\ln\left(\frac{P(\text{general})}{P(\text{academic})}\right)$ = 2.8522 – 0.0579(40) = 0.5362
  - $\frac{P(\text{general})}{P(\text{academic})}$ = exp(0.5362) = 1.7095
  - P(general) = 1.7095*P(academic)

- Similarly, using
  - $\ln\left(\frac{P(\text{vocation})}{P(\text{academic})}\right) = 5.2183 - 0.1136\text{write}$
  - P(vocation) = 1.9635*P(academic)

- But, P(general) + P(vocation) = 1- P(academic)
  - 3.673*P(academic) = 1- P(academic)
  - So, P(academic) = 1/0.4673 = 0.214

```
> #  Probability of being in the three different programs when write = 40, ses = low;
> #
> dses <- data.frame( write = 40,ses = "low")
> print(dses)
  write ses
1    40 low
> predict(mmod1, newdata = dses, "probs")
academic  general vocation
0.214141 0.365653 0.420206
```

$$\ln\left(\frac{\mathbf{P(general)}}{\mathbf{P(academic)}}\right)$$
$$= \mathbf{2.8522 - 0.0579write - 0.5333\ (ses = middle) - 1.1628(ses = high)}$$

$$\ln\left(\frac{\mathbf{P(vocation)}}{\mathbf{P(academic)}}\right)$$
$$= \mathbf{5.2183 - 0.1136write + 0.2914(ses = middle) - 0.9827(ses = high)}$$

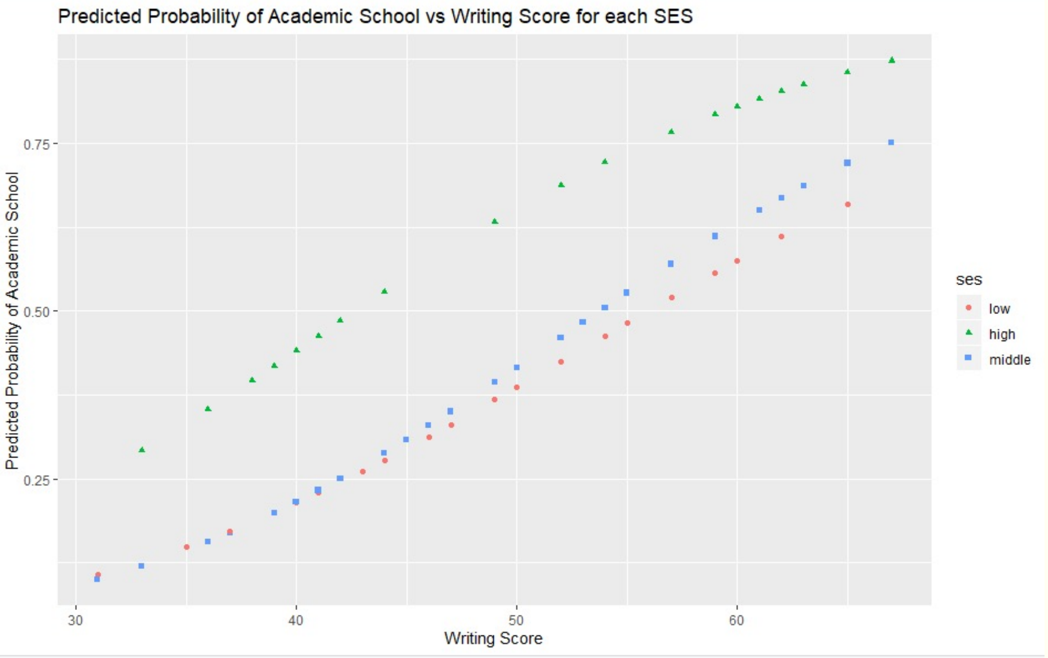
Predicted Probabilities

- We can calculate the predicted probabilities for the entire data set.
- We create a new data frame (dfp) with the original ses, write and prog observations.
- We then obtain the predicted probability for each prog category using the *fitted()* function and append it to dfp.
- For observation 1, the person had ses=low and write = 35. They went to vocation school. The model has the highest predicted probability for vocation as well.
- The plot clearly shows that for over the range of all writing scores, the predicted probability of being in an academic institution is always highest when ses=high.
- It also shows that as write score increases, the predicted probability of being in academic increases for all ses.

```
> print(p_value)
         (Intercept) dfm$ses2high dfm$ses2middle        write
general  0.0144766100   0.02373856      0.2294379 6.818902e-03
vocation 0.0000072993   0.09894976      0.5407530 3.176045e-07
> #
```

```
> dfp <- data.frame(ses, write, prog)
> dfp$pp <- fitted(mmod1)
> print(head(dfp))
     ses write     prog pp.academic pp.general pp.vocation
1    low    35 vocation   0.1482764  0.3382454   0.5134781
2 middle    33  general   0.1202017  0.1806283   0.6991700
3   high    39 vocation   0.4186747  0.2368082   0.3445171
4    low    37 vocation   0.1726885  0.3508384   0.4764731
5 middle    31 vocation   0.1001231  0.1689374   0.7309395
6   high    36  general   0.3533566  0.2377976   0.4088458
```

```
# Predicted Probabilities for whole data set
#
dfp <- data.frame(ses, write, prog)
dfp$pp <- fitted(mmod1)
print(head(dfp))
#
pp_academic <- c(dfp$pp[,1])
print(head(pp_academic))
#
library(ggplot2)
#
#
ggplot(dfp, aes(x=write, y=pp_academic, shape=ses, color=ses)) +
  geom_point() +
  ggtitle("Predicted Probability of Academic School vs Writing Score for each SES") +
  xlab("Writing Score") +
  ylab("Predicted Probability of Academic School")
#
```



Classification Table

- We can develop a classification table using a classification rule that each case will be classified into the program with the highest predicted probability.
- First, we find the column name from the predicted probabilities data frame which has the maximum predicted probability value.
- Then, if this name matches the actual value of prog, we assign a 1.
- We develop a classification table and obtain classification percents.
- The overall correct classification percent for our model is 53%

```
> head(dfp$pp)
   academic   general  vocation
1 0.1482764 0.3382454 0.5134781
2 0.1202017 0.1806283 0.6991700
3 0.4186747 0.2368082 0.3445171
4 0.1726885 0.3508384 0.4764731
5 0.1001231 0.1689374 0.7309395
6 0.3533566 0.2377976 0.4088458
> #
> # Choose COLUMN NAME (academic, general, vocation) from df$pp with the highest probability
> #
> dfp$p_class <- colnames(dfp$pp)[apply(dfp$pp,1,which.max)]
> #
> # Classify as 1 if this column name matches the actual prog value
> #
> dfp$classif <- ifelse(dfp$prog == dfp$p_class, 1, 0)
> print(head(dfp))
     ses write     prog pp.academic pp.general pp.vocation  p_class classif
1    low    35 vocation   0.1482764  0.3382454   0.5134781 vocation       1
2 middle    33  general   0.1202017  0.1806283   0.6991700 vocation       0
3   high    39 vocation   0.4186747  0.2368082   0.3445171 academic       0
4    low    37 vocation   0.1726885  0.3508384   0.4764731 vocation       1
5 middle    31 vocation   0.1001231  0.1689374   0.7309395 vocation       1
6   high    36  general   0.3533566  0.2377976   0.4088458 vocation       0
> #
> # Develop Classification Table
> #
> classif_tbl <- xtabs(~ prog + p_class, data = dfp)
> print(classif_tbl)
          p_class
prog       academic general vocation
  academic       92       4        9
  general        27       7       11
  vocation       23       4       23
> #
> classif_pct <- classif_tbl/length(prog)
> print(classif_pct)
          p_class
prog       academic general vocation
  academic    0.460   0.020    0.045
  general     0.135   0.035    0.055
  vocation    0.115   0.020    0.115
> print(paste("Correct classification Rate (percent): ",
+             (classif_pct[1,1] + classif_pct[2,2] + classif_pct[3,3])*100))
[1] "Correct classification Rate (percent):  61"
```

Conclusion

- When it comes to categorical dependent variables there are many techniques that can be used. Logistic regression, with the logit transformation that we have used, is only one of them.
- Here are some alternatives:
  - Multinomial probit regression: similar to multinomial logistic regression but with independent normal error terms.
  - Multiple-group discriminant function analysis: A multivariate method for multinomial outcome variables
  - Multiple logistic regression analyses, one for each pair of outcomes: One problem with this approach is that each analysis is potentially run on a different sample. The other problem is that without constraining the logistic models, we can end up with the probability of choosing all possible outcome categories greater than 1.
  - Collapsing number of categories to two and then doing a logistic regression: This approach suffers from loss of information and changes the original research questions to very different ones.
  - Ordinal logistic regression: *If the outcome variable is truly ordered* and if it also satisfies the assumption of proportional odds, then switching to *ordinal logistic regression* will make the model more parsimonious.
  - Alternative-specific multinomial probit regression and Nested logit model