Categorical data

Introduktion

Denna föreläsningar handlar om hur man analyserar och skapar modellen när den beroende variabeln är kategorisk. Det finns två typer av skalor på kategoriska variabler:

- **Nominalskala:** Kategoriska variabler som följer en nominalskala kan inte rangordnas. Detta handlar bland annat om *ja* och *nej* frågor, yrkesgrupptillhörighet, eller val av tvättmedel.
- **Ordinalskala:** När en variabel vars skala följer en ordning. Säg, betyg från 1 - 5 kallas för ordinalskala.

Yttligare pratas det om **sambandanalys:**

- Sambandsanalys handlar om hur man undersöker samband mellan kategoriska variabler. För att undersöka detta kan man använda χ^2 test för korstabeller eller Goodness of fit test.

Det diskuteras även **multinominalfördelning** och **länkfunktioner** som används för att modellera och analysera kategoriska variabler. Bland annat:

- Logit.
- Probit.
- Logit modeller.
- Probit modeller.
- Multinominal logit modeller
- Ordnad logit och probit modeller för ordinala responsvariabler.

Analysmetoder

Analys av korstabeller

När man vill undersöka om det finns samband mellan två kategoriska variabler (säg kön och resereservation) använder man ett så kallas **test of independence, χ^2 (SV: test för oberoende).** Detta typ av test har sjävklart antaganden som måste uppfyllas vilka är:

- Slumpmässigt urval.
- Oberoende observationer.
- Helst $E_{ij} \geq 5$
- Minst 80% av $E_{ij} \geq 5$ och ingen $E_{ij} < 1$

Det finns såklart fall när dessa antaganden inte kan uppfyllas. Då kan **Fisher's Exact test** användas istället.

Vad undersöker man (hypoteser)

Man skriver upp nollhypotes och alternativ hypotesen:

$$H_0 : \text{Finns inget samband mellan kön och resereservation}$$
$$H_A : \text{Finns samband mellan kön och resereservation}$$

Korstabellen

Metod	Kvinna	Man
Resebyrå	256	74
Internet	41	42
Telefon	66	34

Vi har vår korstabell och genom denna kan vi nu (givet att antaganden uppfylls) genomföra testet.

Statistikan

Under nollhypotesen beräknas de förväntade frekvenserna som:

$$E_{ij} = \frac{R_i \cdot C_j}{n}$$

Här är R_i och C_i rad och kolumntotalen för rad i eller kolumn j. Vidare kan följande statistikan beräknas:

$$\chi^2_{test} = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

Här är då O_{ij} de observerade frekvenserna från korstabellen med r rader och c kolumner. Denna statistika är χ^2 fördelad där nollhypotesen förkastas om testvariabeln är större än det kritiska värden med angiven signifikansnivå:

$$\chi^2 \sim \chi^2_{df=(r-1)(c-1)}$$

Reslutat och förväntade frekvenser

```r
# Indexerar de två frekvenskolumnerna
result <- chisq.test(obs[,2:3])

result$expected |>
  kable(caption = "Förväntade frekvenser")
```

Förväntade frekvenser

Kvinna	Man
233.50877	96.49123
58.73099	24.26901
70.76023	29.23977

```r
result
```

```
    Pearson's Chi-squared test

data:  obs[, 2:3]
X-squared = 26.811, df = 2, p-value = 1.507e-06
```

I bilden ovan ses de förväntade frekvenserna tillsammans med resultaten för statistikan med tillhörande p-värde (istället för kritiska värdet). Man kan notera att p-värdet är mindre än 5% därför förkastas nollhypotesen. Slutsatsen kan dras att variablerna har ett signifikant samband.

- Tolka förväntade frekvensen (expected frequency).

De förväntade frekvenserna kan tolkas som att för resebyrå förväntas det vara 234 kvinnor och 97 män.

Multinomialfördelningen

Multinomialfördelningen är en **generalisering av binomalfördelningen** (dikotom/binär variabel) när man har fler än 2 kategorier. Givet att man har

kända (known) sannolikheter för k kategorier $(p_1, ..., p_k)$ ges den gemensamma sannolikheten som:

$$P(x_1, x_2, ..., x_k) = \binom{n}{x_1, x_2, ..., x_k} \cdot p_1^{x_1} \cdot p_2^{x_2} \cdot ... \cdot p_k^{x_k}$$

för alla **icke negativa heltal för** $x_1, ..., x_k$.

Regressionsmodeller (för icke-kontinuerliga responsvariabler)

Binär responsvariabel

När man har en binär responsvariabel är inte linjär regression lämplig utan man behöver en annan länkfunktion, $F(x)$. Vi måste även definera om det förväntade värdet av den stokatiska variabeln, vilket blir:

$$E[Y] = P(Y = 1) = F(\boldsymbol{x}\boldsymbol{\beta})$$

Alltså vi beräknar det förväntade värdet av den stokastiska variabeln, Y, vars värde vi antar är 1. Här är följande parametrar:

$$\boldsymbol{x} = [1, x_1, x_2, ..., x_k] \text{ och } \boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2, ..., \beta_k]^t$$

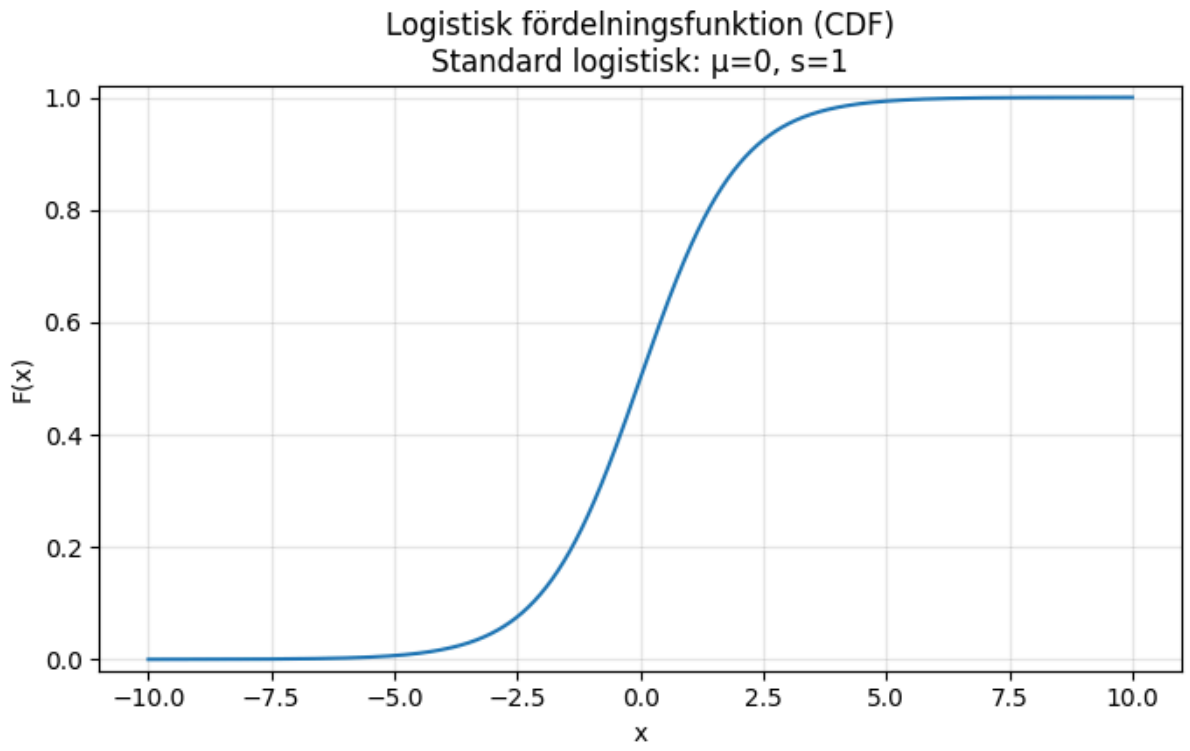
Logit länken och länkfunktionen

Det som faktiskt gör denna typ av regression till binär regression är logit länken som förvandlar vår linjära regression (eftersom vi har ursprungligen en linjär regression) till en sannolikhet mellan 0 och 1. Logit innebär log-oddset enligt:

$$\ln\left(\frac{P(Y = 1)}{P(Y = 0)}\right) = \boldsymbol{x}\boldsymbol{\beta}$$

$$P(Y = 1) = \frac{e^{\boldsymbol{x}\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}\boldsymbol{\beta}}} = \Lambda(\boldsymbol{x}\boldsymbol{\beta})$$

Här är då Λ den logistiska fördelningsfunktionen.



Probit modellen

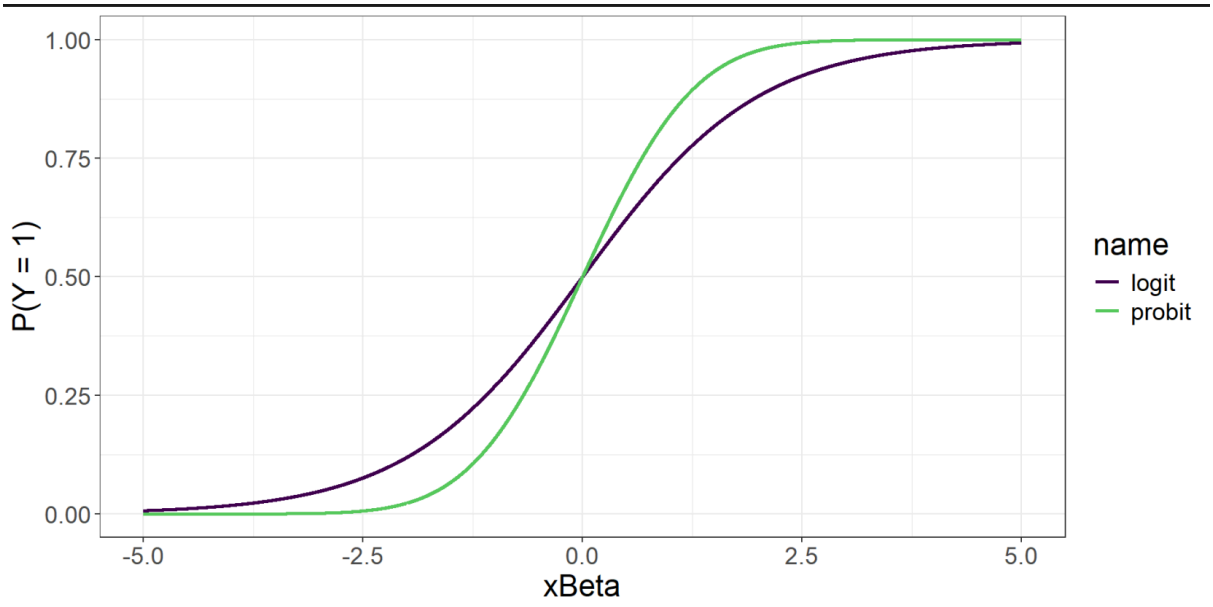
En probit modell är en binär regressionsmodell (med en annan länkfunktion än logit) där man kopplar sannolikheten till en linjär responsvariabel visa standardnormalfördelningen CDF. Det vill säga genom att använda normalfördelningen får man också en begränsning inom intervallet $[0, 1]$ genom följande integral:

$$P(Y = 1) \int_{-\infty}^{\boldsymbol{x\beta}} \phi(t)dt = \boldsymbol{\Phi(x\beta)}$$

Som man då kan notera är $\Phi(\cdot)$ fördelningsfunktionen (CDF) för en standardnormal, $Z \sim N(0, 1)$.

Probit vs logit (länkfunktioner)

Skillnaden mellan probit och logit som länkfunktioner är inte jätte annorlunda — de modellerar samma sak. Hur som helst har probit en snävare sigmoid kurva än vad logit har.



Maximum likelihoodskattning (för parameter estimering)

Bernoulli modell

Problemet vi möter när man går vidare för att modellera en binär responsvariabel är att vanligt OLS inte längre fungerar för att uppskatta parametrarna i modellen. Därför måste man använda en annan metod, och den bästa för detta är maximum likelihood estimation. Det positiva med MLE är att man kan skatta parametrar från en mängd olika fördelningar som Weibull, Pareto, Normal, osv.

Anledningen till varför vi kallar det Bernoulli modell är eftersom vi modellerar ett utfall som är binärt (0/1). MLE skattningen för Bernoulli modellen för **en observation är:**

$$Y_i|\theta \sim^{iid} \text{Bernoulli}(\theta = P(Y_i = 1) = F(\boldsymbol{x_i}\boldsymbol{\beta}))$$

Men oftast kommer vi aldrig att ha **endast en observation** i vår datamängd utan en mängd observationer. Därför behöver vi gå från föregående uttryckt till att skapa en **gemensam sannolikhet (eng: joint probablity).** I praktiken har vi n observationer. Man modellerar därför den gemensamma fördelningen för slumpvektorn $\boldsymbol{Y} = (Y_1, ..., Y_n)$(stokatiska vektorn) givet β. Om vi nu antar att

observationerna är oberoende givet β och att $Y_i|\beta \sim \text{Bernoulli}(p_i)$ med $p_i = F(x_i^t\beta)$. Då faktoriserar den gemensamma sannolikheten som:

$$\begin{aligned} L &= P(Y_1 = y_1, Y_2 = y_2, ..., Y_n = y_n|\theta) \\ &= \prod_{y_i=0} P(Y_i = 0) \cdot \prod_{y_i=1} P(Y_i = 1) \\ &\prod_{y_i=0} [1 - F(\boldsymbol{x_i\beta})] \cdot \prod_{y_i=1} F(\boldsymbol{x_i\beta}) \\ &= \prod_{i=1}^{n} [1 - F(\boldsymbol{x_i\beta})]^{1-y_i} \cdot [F(\boldsymbol{x_i\beta})]^{y_i} \end{aligned}$$

I uttrycket ovan har vi kommit fram till **likelihoodfunktionen** men den är lite komplicerad att derivera i nuläget. Därför tar vi $\ln$ av likelihoodfunktionen och får **loglikelihoodfunktionen**. Vi kommer nu att arbeta med summor istället vilket är något enklare:

$$\ln L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \Big[y_i \ln F(\mathbf{x}_i\boldsymbol{\beta}) + (1 - y_i) \ln \big(1 - F(\mathbf{x}_i\boldsymbol{\beta})\big) \Big].$$

Sedan får skattningen genom att **maximera loglikelihoodfunktionen med avseende på β.** Alltså fås det parameter värde som gör den observerade datan mest sannolik enlitg modellen.

Optimeringen

MLE är ett **optimeringsproblem**. Det vill säga vi skriver en algoritm som kommer att upprepa beräkningar tills den kommer fram till konvergens. Det vill säga att den faktiskt funnit maximum. Detta går såklart när man beräknar för hand att kontrollera att andra derivatan är negativt ⇒ man har nått maximum. Men man måste tänka på att det optimum som fås är lokalt, men det lokala optimum kan vara det globala optimum. Detta är förstås svårt att visualisera när man har en stor datamängd eftersom att man inte kan visualisera efter tredje dimensionen.

Här är kod för ett exempel på hur man skriver en optimeringsalgoritm för att genomföra en maximumlikelihood skattning:

```
# Läser in relevant data
load("Spector_Mazzeo_Data.RData")
```

```
n ← length(GRADE)
y ← matrix(GRADE, nrow = 1, ncol = n)

Ones ← matrix(1, nrow = n, ncol = 1)

# Skapar designmatrisen
x ← matrix(c(Ones,PSI,GPA,TUCE), nrow = n, ncol = 4)

# Skapar loglikelihoodfunktionen
LogLikFunc ← function(Beta){

  Logistic_x_Beta ← exp(x%*%Beta)/(1+exp(x%*%Beta))

  LogLiks ← y%*%log(Logistic_x_Beta) + (1-y)%*%log(1-Logistic_x_Beta)

  return(-LogLiks)
}

# Initialiserar värden på parametrarna
InitVal ← matrix(c(0.5,0.5,0.5,0.5), nrow = 4, ncol = 1)
# Genomför numerisk optimering
OptRes ← optim(par = InitVal, fn = LogLikFunc)
```

Tänk på här att man inte ska välja startvärden som är 0 på grund av att man kan få problemet med derivatorna. Sedan kan man notera konvergens ska vara 0. Detta innebär att algortimen har konvergerat. Reslutatet som fås blir:

```
OptRes

$par
            [,1]
[1,] -13.02250520
[2,]   2.37764325
[3,]   2.82698857
[4,]   0.09506494
```

```
$value
[1] 12.88964

$counts
function gradient
     419       NA
```

- par = de skattade parametrarna som fås av MLE. Det vill säga:
  - Intercept
  - b1
  - b2
  - b3
- value = 12.88 vilket är $\log L(\hat{\beta} = -12.88)$
- count visar:
  - function = 419. Det vill säga att funktionen utvärdera målfunktionen 419 gånger.
  - gradient = NA betyder att man inte gav någon gradientfunktion så den utvärderas ej.

Parametertolkning

I en **logit modell** visar parametrarna förändringen av **log-oddset.** För att göra det mer tolkningsbart behöver man transformera om parametern e^{β_j} så att **oddset** kan tolkas. Vi säger då:

> När x_i ökar med en enhet, förändras **oddset** att tillhöra klass 1 med faktor e^{β_j} , givet att alla andra variabler hålls konstanta.

I ett exempel: **Spector och Mazzeo (1980)** analyserade effekten av en ny undervisningsmetod i nationalekonomi. Datamaterialet bestod av:

- Beroende variabel **GRADE:** en indikator med värde 1 och students betyg förbättrades och 0 annars.
- Förklaringsvariabel **PSI:** antar värdet 1 om den nya undervisningsmetoden användes och 0 annars.
- Förklaringsvariabel **GPA:** genomsnittsbetyg för varje individ i urvalet.

- Förklaringsvariabel **TUCE:** poäng på ett test om förkunskap.

När den nya undervisningsmetoden används **(PSI = 1)** ökar oddset att få bättre betyg med $e^{2.378} = 10.779$ gånger jämfört med när metoden inte användes givet alla andra variabler hålls konstanta.

Anpassning av logit och probit modeller i R

```
modelLogit ← glm(GRADE ~ PSI + GPA + TUCE, family = "binomial")

summary(modelLogit)

Call:
glm(formula = GRADE ~ PSI + GPA + TUCE, family = "binomial")

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -13.02135    4.93127  -2.641  0.00828 **
PSI          2.37869    1.06456   2.234  0.02545 *
GPA          2.82611    1.26293   2.238  0.02524 *
TUCE         0.09516    0.14155   0.672  0.50143
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 41.183  on 31  degrees of freedom
Residual deviance: 25.779  on 28  d
```

```
modelProbit ← glm(GRADE ~ PSI + GPA + TUCE, family = "binomial"(link = "probit"))

summary(modelProbit)

Call:
glm(formula = GRADE ~ PSI + GPA + TUCE, family = binomial(link = "probit"))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.45231    2.57152  -2.898  0.00376 **
PSI          1.42633    0.58695   2.430  0.01510 *
GPA          1.62581    0.68973   2.357  0.01841 *
TUCE         0.05173    0.08119   0.637  0.52406
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 41.183  on 31  degr
```

egrees of freedom
AIC: 33.779

Number of Fisher Scoring iteration
s: 5

ees of freedom
Residual deviance: 25.638 on 28
degrees of freedom
AIC: 33.638

Number of Fisher Scoring iteration
s: 6

Hur jämför man modellerna?

I outputs kan man se att modellerna är mycket lika i passform:

- Residual deviance: 25.638 (probit) vs 25.779 (logit)
- AIC: 33.638 (probit) vs 33.779 (logit)

skillnaderna är så små vilket innebär att de i princip ger samma fit på datan.

Vilken ska man välja?

- Välj **logit** om du vill prata om **odds ratio** (vanligt i tillämpningar).
- Välj **probit** om du gillar latent normal tolkning (vanligt i ekonomi/ekonometri) eller om det passar en teoretisk härledning.

Statistisk inferens

När man genomför statistisk inferens kan man genomföra **z-test** för enskilda parametrar där s_{β_j} tas från den skattade kovariansmatrisen $S_{\hat{\beta}}$. Wald test kan användas för generella restriktioner flexibelt genom formen:

$$H_0 : \boldsymbol{R\beta} = \boldsymbol{q}$$

Men notera här att när vi genomföra detta **wald test** testar man flera (linjära) restriktioner samtidigt. Här är $\boldsymbol{R}$ är en $r \times k$ matris där r = antal restriktioner och k = antal parametrar. **Restriktioner innebär** att man skapar en matris över parametrarna (begränsar våra parametrar till vissa värden) och gör **ETT** ett. Alltså ett gemensamt test istället för flera individuella. **Wald statistikan** är:

$$W = (\boldsymbol{R\hat{\beta}} - \boldsymbol{q})^t [\boldsymbol{R}(\boldsymbol{S_{\hat{\beta}}})\boldsymbol{R^t}]^{-1} (\boldsymbol{R\hat{\beta}} - \boldsymbol{q})$$
$$W \sim \chi^2_r$$

Wald testet är approximiativt χ^2 fördelad med antalet restritkioner under nollhypotesen.

Här är då:

$$\underbrace{\begin{bmatrix} r_{10} & r_{11} & \cdots & r_{1k} \\ r_{20} & r_{21} & \cdots & r_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n0} & r_{n1} & \cdots & r_{nk} \end{bmatrix}}_{\mathbf{R}} \boldsymbol{\beta} = \mathbf{q}, \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}, \qquad \mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix}.$$

med r_{ij} och q_i är reella tal och n är antalet restriktioner under nollhypotesen.

Wald test

Wald testet kan skrivas i ett ekvationsystem som en linjärkombination:

$$\mathbf{R}\boldsymbol{\beta} = \mathbf{q} \iff \begin{cases} r_{10}\beta_0 + r_{11}\beta_1 + \cdots + r_{1k}\beta_k = q_1, \\ r_{20}\beta_0 + r_{21}\beta_1 + \cdots + r_{2k}\beta_k = q_2, \\ \qquad \vdots \\ r_{n0}\beta_0 + r_{n1}\beta_1 + \cdots + r_{nk}\beta_k = q_n. \end{cases}$$

Skattade koefficienter och kovariansmatrisen fås (i R) genom **coef()** och **vcov()**. För att undersöka GPA och TUCE samtidigt i logitmodellen gör man följande:

```
require(aod)

wald.test(
  b = coef(modelLogit),
  Sigma = vcov(modelLogit),
  Terms = 3:4
)

Wald test:
----------

Chi-squared test:
X2 = 6.4, df = 2, P(> X2) = 0.042
```

I koden ovan säger vi då att i nollhypotesen är GPA och TUCEs skattningar lika med 0 eller i alternativhypotesen att minst en skiljer sig från noll. Vi ser i utskriften att p värdet är 0.042 vilket förkastar nollhypotesen. GPA och TUCE är statistiskt signifikanta.

Vad är problemet i detta?

Det som ofta händer är att TUCE har högt pvärde ensam men tillsammans med GPA blir det gemensamma testet signifikant.

Likelihoodkvot-test

Likelihoodkvot testet för **allmänna restriktioner** undersöker relationen mellan två modeller genom:

$$LR_{test} = -2 \cdot [\ln \hat{L}_r - \ln \hat{L}]$$

Här är $\ln \hat{L}_r$ är log-likelihood av den restrikterade (restricted) modellen och $\hat{L}$ är då log-likelihood av den icke-restrikerade (unrestricted) modellen.

Teststatistikan är approximativt χ^2 fördelad med antalet restriktioner under nollhypotesen. Om LRT är liten får vi konsekvent en liten skillnad — men om den är stor så är skillnaden stor. Detta innebär i sin tur att om vi har en modell som är betydligt bättre än den andra kommer LRT också vara hög. Att förkasta nollhypotesen i detta fall innebär att man inte behöver begränsa modellen. LRT motsvarar i linjär regresson det partiella F testet.

Hur gör man i R?

```
require(lmtest)

modelUnRestricted ←
  glm(GRADE ~ PSI + GPA + TUCE, family = "binomial")

modelRestricted ← glm(GRADE ~ PSI, family = "binomial")

lrtest(modelUnRestricted, modelRestricted)

Likelihood ratio test

Model 1: GRADE ~ PSI + GPA + TUCE
Model 2: GRADE ~ PSI
```

```
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   4 -12.890
2   2 -17.671 -2 9.5624   0.008386 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I exemplet ovan görs ett LRT jämfört den fulla modellen (PSI+GPA+TUCE) mot en restrikterad modell (bara PSI). Vi ser att p värdet är mycket lågt vilket innebär att man kan förkasta nollhypotesen alltså den fulla modellen passar signifikant bättre än en modell med endast PSI.

Likelihoodkvot-test vs Wald test

För att faktiskt gå ner till botten av vad det är vi testar så måste man fråga sig själv vad är det vi jämför?

Vad är det vi jämför?

Båda testerna (Wald och LRT) testar samma typ av hypotes (säg):

$$H_0 : \beta_{\text{GPA}} = 0 \text{ och } \beta_{\text{TUCE}} = 0$$

vi undersöker alltså den restrikterade modellen mot alternativet att minst en av dem inte är noll.

LRT

LRT bygger på skillnaden i maximerad log-likelihood mellan två modeller:

$$LR = 2(\ell(\hat{\beta}_{\text{full}} - \ell(\hat{\beta}_{\text{restricted}})) \approx^{H0} \chi^2_r$$

detta test kräver alltså att man skattar två modeller.

Wald testet

Wald testet bygger på hur långt $\hat{\beta}$ ligger från restriktionen, vägt med kovariansmatrisen:

$$W = (\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{q})^\top \left(\mathbf{R}\, \widehat{\mathrm{Var}}(\hat{\boldsymbol{\beta}})\, \mathbf{R}^\top \right)^{-1} (\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{q}) \overset{H_0}{\approx} \chi^2_r.$$

Detta krävetr **endast en modell (den orestrikterade).**

Ger testerna faktiskt samma reslutat?

Testerna ger inte nödvändigtvis samma resultat. De är asymptotiskt ekvivalenta (de brukar närma sig varandra när $n \to \infty$ men för ändliga stickprov kan de ge olika p-värden och ibland olika beslut.

Det vill säga **WALD och LRT testar samma restriktioner** men kan ge olika p-värden i ändliga stickprov eftersom att Wald bygger på en lokal normal/kvadratapproximation via standardfel, medan LRT bygger på skillnaden i log likelihood mellan två modeller. **Men testerna är asymptotiskt ekvivaltenta** när n går mot oändligheten och då ger dem samma resultat.

Utvärderingsmått för logit och probit modellen

För att utvärdera logit och probit modeller kan man anvönda **likelihoodkvotindex** vilket är *ett mått mellan 0 och 1* på hur bra modellen anpassar data:

$$LRI = 1 - \frac{\ln L}{\ln L_0}$$

Här är:

- $\ln L_0 = n \cdot [P \cdot \ln P + (1-P) \cdot \ln(1-P)]$ och P är andelen 1:or (eller 0:or) för responsvariabeln.

Om $LRI = 1$ är modellen perfekt i anpassning men om $LRI = 0$ bidrar ingen förklarande variabel till modellen. Man bör dock vara informerad att måttet inte är tolkningsbart i procent endast för att det kan vara mellan noll och ett.

Mått mellan 0 och 1 hur bra modellen anpassar till data. Jämfört med en tom modell och den man har nu. Detta får inte tolkas som procent

---

Modeller för k > 2 (fler än två kategorier)

> Hur utökar vi den logistiska modellen för att hantera en responsvariabel som har fler än två kategorier då man inte kan använda logit eller probit länken.

Ordningen spelar roll

Först måste man kontrollera vilken kategorisk variabel vi hanterar då ordningen spelar roll.

- Nominal skala
- Ordinal skala

Multinomial logit modell

Finns det ett problem med en multinomial probit modell?

Det finns ett stort problem när det kommer till den multinomiala probit modellen. Man antar att feltermerna i modellen är **multivariat normafördelade** och kan vara korrelerade. Detta motsvarar en sannolikhet för att en multivariat normalvariabel hamnar i ett visst område. Det blir en intergral i dimension $k-1$ (men större om man har korrelationer). Denna intergral har oftast ingen enkel sluten form vilket innebär att man kräver numerisk intergration eller simulering och det blir mycket beräkningstungt när k växer.

Men varför blir multinomial logit enkel?

I en multinomial logit antar man istället en annan fördelning för feltermerna (typiskt i extremvärdesfamiljen/Gumbel). Då råkar det finnas en snygg sluten form för val-sannolikheterna:

$$P(Y=k) = \frac{e^{\boldsymbol{x}\boldsymbol{\beta_k}}}{1+\sum_{k=2}^{K} e^{\boldsymbol{x}\boldsymbol{\beta_k}}}, \quad k=1,...,K$$

Här är $\boldsymbol{\beta}_k$ parametervektorn för kategori k av total K kategorier och $\boldsymbol{x}$ är radvektorn av förklaringsvariabler.

Specialfall för referenskategori $k=1$, där $\beta_1 = 0$, ger:

$$P(Y=1) = \frac{e^{\boldsymbol{x}\boldsymbol{\beta_k}}}{1+\sum_{k=2}^{K} e^{\boldsymbol{x}\boldsymbol{\beta_k}}}$$

Loglikelihooden för multinomiala logit modellen

Loglikelihooden för multinomiala logit modellen generaliserar den binomiala logit modellen:

$$\ln L = \sum_{i=1}^{n} \sum_{k=1}^{K}) d_{ik} \ln P(Y_i = k)$$

Där $d_{ik} = 1$ om alternativ k väljs av enhet i, och 0 annars. Loglikelihooden då alla koefficeinter är lika med noll blir:

$$\ln L_0 = \sum_{k=1}^{K} n_k \ln P_k$$

> $\ln L_0$ är en tom modell (denna formel som visar är loglikelihooden för den tomma modellen för att utvärdera efter att deb blivit anpassad).

Modellen ger oddset för kategorierna j och k enligt:

$$\frac{P(Y=j)}{P(Y=k)} = \exp[\boldsymbol{x}(\boldsymbol{\beta_j} - \boldsymbol{\beta_k})]$$

Ordinal logit modell

Vad är problemet som ordinal logit modell löser?

Problemet med logit och probit modellerna är att de inte kan hantera när kategorierna i responsvariabeln har en specifikt ordning, säg rankning från 1-5. Det går inte heller att använda linjär regression i detta syftet. För att lösa detta kan man använda en **ordinal logit modell**

Latent regression (ordnad logit)

Latent regression betyder att man tänkre sig att det finns en underliggande kontinuerlig variabel y^* som följer en vanlig regressionsmodell, men att vi **inte observerar** y^* **direkt.**

I en ordnad logit modell antar man att:

$$y^* = \boldsymbol{x}\boldsymbol{\beta} + \epsilon$$

Det man **observerar** är bara en ordnad kategori $y \in \{1, ..., K\}$ som skapas genom att jämföra y^* med trösklar (cutpoints)

- y^* är en latent (icke-observerat) representation av y, för representationen gäller:

$$y = \begin{cases} 1, & y^* \leq \mu_1, \\ 2, & \mu_1 < y^* \leq \mu_2, \\ \vdots & \vdots \\ K, & y^* > \mu_{K-1}. \end{cases}$$

Denna beskriver "kopplingen" mellan en **latent** kontinuerligt variabel y^* och den observerade ordnade kategorivariabeln $y \in \{1, .., K\}$:

- $\mu_1, ..., \mu_{K-1}$ är tröskelvärdet (cutpoints) som delar in y^* i K intervall.
- Om y^* hamnar i ett visst intervall så observerar vi motsvarande kategori y.
    - Det vill säga y är en disktretiserad version av y^*

Ordinal logit modell

Antag den logistiska fördelningsfunktionen för feltermen ϵ, dvs:

$$P(\epsilon \leq x\beta) = \Lambda(x\beta)$$

Då följer det att:

$$\begin{aligned} P(y = 1) &= P(y^* \leq \mu_1) = P(x^\top \beta + \varepsilon \leq \mu_1) = \Lambda(\mu_1 - x^\top \beta), \\ P(y = 2) &= P(\mu_1 < y^* \leq \mu_2) = \Lambda(\mu_2 - x^\top \beta) - \Lambda(\mu_1 - x^\top \beta), \\ P(y = 3) &= P(\mu_2 < y^* \leq \mu_3) = \Lambda(\mu_3 - x^\top \beta) - \Lambda(\mu_2 - x^\top \beta), \\ &\vdots \\ P(y = K) &= P(y^* > \mu_{K-1}) = 1 - \Lambda(\mu_{K-1} - x^\top \beta), \\ &\text{där } \mu_1 < \mu_2 < \cdots < \mu_{K-1}, \qquad \Lambda(t) = \frac{1}{1 + e^{-t}}. \end{aligned}$$

Här är loglikelihoodfunktionen $\ln L = \sum_{i=1}^{n} \sum_{k=1}^{K} d_{ik} \cdot \ln P(Y_i = k)$ är en generalisering av logit modellen och maximering av loglikelihoodfunktionen ger skattade värden på $\beta, \mu_1, ..., \mu_{K-1}$.