Labbrapport i Statistik

Laboration 1

732G34

Hampus Beijer

Avdelningen för Statistik och maskininlärning
Institutionen för datavetenskap
Linköpings universitet

2026-1-18

Innehåll

1 R Paket

```
require(dplyr)
require(kableExtra)
require(lmtest)
require(aod)
require(MASS)
library(nnet)
```

2 Uppgift 1

I följande uppgift är ett datamaterial bestående av tusen autktioner av förpackningar av samlarmynt från eBay , vilket består av följande variabler:

- `UnOpen`: en dikotom variabel där 1 innebär att förpackningen är oöppnad och 0 om den är öppnad.
- `PowerSeller`: en diktorom variabel som visar om en säljare på eBay är rankad bland de mest lyckosamma. Om en individ är har den värdet 1, annars 0.
- `LogBookValue`: en kontinuerlig variabel som är den naturliga logartimen av priset på myntet i dess förpackning (taget från Golden Eagle Coins).
- `StartBidRatio`: en kontinuerlig variabel som mäter kvoten mellan det lägsta budpriset och priset på myntförpackningen (från Golden Eagle Coins).

De tjugo första observationerna av datamaterialet presenteras i tabell 1.

```
eBay <- read.csv2("/Users/hampusbeijer/Downloads/eBay.csv")
kable(head(eBay,20), caption ="Datamaterial över tusen samlarmynt sålda på eBay.")
```

Tabell 1: Datamaterial över tusen samlarmynt sålda på eBay.

UnOpen	PowerSeller	LogBookValue	StartBidRatio
0	0	2.941804	0.3688654
0	1	3.772761	0.2298851
0	0	4.600158	0.1672362
0	0	2.251292	1.2094737
0	0	2.917771	0.8729730
0	0	2.251292	1.2094737
0	0	2.708050	0.3653333
0	0	4.051785	0.0520000
0	0	3.540959	0.1736232
0	0	2.803360	0.2569697
0	0	2.803360	0.2569697
1	1	4.085976	0.2504202
1	1	3.384390	0.5050847
1	1	4.051785	0.2591304

UnOpen	PowerSeller	LogBookValue	StartBidRatio
1	1	3.384390	0.5050847
1	1	3.113515	0.6622222
0	0	5.164786	0.0439429
0	0	2.014903	1.1320000
0	0	4.241327	0.0933813
0	0	4.600158	0.0652261

2.0.1 Grundläggande teori för deluppgift 1a)

Maximum Likelihood Estimation (MLE)

Innan denna uppgift kan genomföras behöver den teoretiska aspekten diskuteras. Maximum likelihood skattning (MLE) är en metod för att uppskatta värden på parametrar utan att behöva använda priorfördelningar eller en förlustfunktion. Enkelt förklarat finner man den skattning, θ, som maximerar en given likelihoodfunktion (s. 417). Det bra med denna metod är att man kan uppskatta parametrar för olika likelihoodfunktioner (t.ex: Binomial, Normal, Pareto, Weibull, m.m).

I fallet med denna uppgift är responsvariabeln dikotom, alltså kan anta värden $Y_i \in [0, 1]$, vilket är definitionen av Bernoullifördelningen. Så responsvaraibeln är:

$$Y \sim \text{Bernoulli}(\theta), \quad \text{Där } P(Y = 1) = \theta \text{ och } P(Y = 0) = 1 - \theta$$

Bernoullifördelningens likelihoodfunktion defineras som:

$$f(\boldsymbol{x}|\theta) = \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i} = \theta^{\sum_{i=1}^{n} x_i}(1-\theta)^{n-\sum_{i=1}^{n} x_i} = \ell(\theta)$$

För att få fram maximum krävs det att man deriverar likelihoodfunktionen och sätter den till noll, och sedan löser den. Men problemet med detta är att derivera produkter blir snabbt komplicerat. Därför kan man istället ta logartimen av likelihoodfunktionen, eftersom att man få kommer jobba med summor (DeGroot & Schervish, 2013, s. 420).

$$\log \ell(\theta) = \log f(\boldsymbol{x}|\theta) = \sum_{i=1}^{n} \left[x_i \log \theta + (1 - x_i) \log(1 - \theta)\right]$$

Nu deriverar man log-likelihoodfunktionen med avseende på θ:

$$\frac{d\ell(\theta)}{d\theta} = \frac{\sum x_i}{\theta} - \frac{n - \sum x_i}{1 - \theta}$$

Sätt nu derivatan lika med noll och lös ut θ:

$$\frac{\sum x_i}{\theta} - \frac{n - \sum x_i}{1 - \theta} = 0$$
$$\frac{\sum x_i}{\theta} = \frac{n - \sum x_i}{1 - \theta}$$

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

På detta sätt har man alltså nu kommit fram till MLE för en Bernoulli likelihoodfunktion.

2.0.2 Bakomliggande teori för deluppgift 1e)

Likelihoodkvot-test

Ett likelihoodkvot-test används för att genomföra en hypotesprövning om huruvida det finns en signifikant skillnad mellan en full och restrikterad modell. Statistikan formuleras enligt:

$$\text{LR} = -2 \cdot [\ln \ell_r - \ln \ell_f]$$

Här defineras $\ln \ell_R$ som log-likelihooden för den restrikterade modellen och $\ln \ell_f$ som log-likelihooden av den fulla modellen. När man beräknat likelihood-kvot testet kan man direkt tolka det, där en stor skillnad indikerar på ett en full modell är lämpligast. Hur som helst, bör man beräkna det kritiska värdet (eller p-värdet) för att göra en korrekt bedömning. Likelihood-kvot testes anses vara chi^2 fördelat enligt:

$$\text{LR} \sim \chi^2_k$$

Där k är antalet restriktioner, bättre sagt, antal variabler som tagits bort från den fulla modellen. Om noll-hypotesen inte kan förkastas tyder detta på att den fulla modellen är bättre än den restrikterade modellen (UCLA Institute for Digital Research and Education, n.d.).

2.0.3 Bakomliggande teori inför uppgift 1f)

Wald test

Ett Wald test är ett alternativ till ett likelihoodkvot-test, i princip, ger samma slutsats som likelihood-kvot testet. Men ett Wald test kräver endast en modell vilket är den fulla modellen (UCLA Institute for Digital Research and Education, n.d.).

Wald testet är dock inte detsamma som ett z-test, eftersom att ett z-test undersöker en skattad parameter i taget. Skillnaden mellan z-testet och Wald testet är att man undersöker flera skattade parametrar samtidigt. Man vill undersöka hypoteserna:

$$H_0 : R\beta = qH_a : R\beta \neq q$$

Där statistikan defineras enligt:

$$W = (\boldsymbol{R}\hat{\boldsymbol{\beta}} - \boldsymbol{q})^{\mathrm{t}} \left[\boldsymbol{R}\, \widehat{S}_{\beta}\, \boldsymbol{R}^{\mathrm{t}}\right]^{-1} (\boldsymbol{R}\hat{\boldsymbol{\beta}} - \boldsymbol{q})$$

$$\boldsymbol{R} = \begin{pmatrix} r_{10} & r_{11} & \cdots & r_{1k} \\ r_{20} & r_{21} & \cdots & r_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n0} & r_{n1} & \cdots & r_{nk} \end{pmatrix} \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} \quad \boldsymbol{R\beta} = \boldsymbol{q} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix}$$

Där $\boldsymbol{\beta}$ är en kolonnvektor bestående av modellens parametrar. R är restriktionsmatrisen, bättre förklarat, den väljer ut vilka parametrar som ska restrikteras. q är ett reelt tal med de värdena som testar i nollhypotesen (Hietala, 2026, s.18).

2.1 Deluppgift 1a)

Beräkna maximum likelihoodskattningarna för parametrarna i en probitmodell genom att på egen hand skriva (och kommentera) kod för att maximera likelihoodfunktionen. Jämför sedan om skattningarna från din kod stämmer överens med de skattningar som fås från funktionen glm(). (2p)

I denna uppgift genomfördes MLE för hand med hjälp av en algortim som togs up i kursen gång. Självklart skrevs denna om av mig. Algortimen skrevs lite annorlunda än den som presenteras i föreläsningen på grund av att den optimerade för en logit modell medan denna gör för en probit modell.

I tabell 2 presenteras parameterskattningarna för den skattade probit modellen med glm() funktionen. Man kan notera att $LogBookValue$ och $StartBidRatio$ är icke-signifikanta. Parameterskattningen för Interceptet är -1.8903, 0.4317 för PowerSellers, 0.1459 för LokBookValue, och -0.1015 för StartBidRatio.

I tabell 3 presenteras parameterskattningarna för min optimeringsalgoritm och glm() funktionens parameterskattningar. Dessa skattningar är, i princip, exakt samma men skiljer sig med en liten decimal. Se kod nedan:

```
### Deluppgift 1a) ------------------------------------------------------------#

# I följande del ska MLE genomföras för parametrarna i en probitmodell genom
# att själva skriva en function som kan finna optimum. Sedan ska detta jämföras
# med skattningat från glm() funktionen

# FÖRBEREDELSE INFÖR ALGORTIMEN ----------------------------------------------#

# Först behövs längden av responsvektorn för att bilda en matris
# dvs att vi får en kolonnvektor (i matrisform) av den beroende variabeln

antal <- length(eBay$UnOpen)
y <- matrix(eBay$UnOpen,nrow = 1,n = antal) # responsvektor

# Nu behövs en designmatris skapas som inkluderar interceptet, men observera att
# denna endast ska bestå av 1:or, sedan kommer resterande variabler
designmatris <- cbind(1,eBay$PowerSeller,eBay$LogBookValue,eBay$StartBidRatio)

# Undersöker så att designmatrisen faktiskt blev en matris: OK
is.matrix(designmatris)
```

```
## [1] TRUE
```

```
# Nu är allt klart och algortimen kan skapas; dvs funktionen för optimeringen!
#-------------------------------------------------------------------------------#

# Jag tänker använda samma variabelnamn som från föreläsningen som Isak Hietala
# har skapat på grund av att det visar en god förståelse att det faktiskt är
# parametervektorn för modellen som multipliceras med designmatrisen.
LogLikelihoodFunktionen <- function(beta){
```

```
  # Detta är designmatrisen multiplicerat med den skattade parametervektorn
  xBeta <- designmatris %*% beta

  # PROBIT: ------------------------------------------------------------------#
  # Det skiljer sig någorlunda från den vanliga logit modellen eftersom att
  # eftersom man inte använda logit länken utan man använder normalfördelningens
  # täthetsfunction som oftast beskrivs med stora Phi.

  # Skillnaden blir nu att p_i i modellen växlad med
  # Phi(deisgnmatrisen * parametervektorn), teorin visar detta. Så konsekvent
  # så måste probitsannolikheterna fås från Phi(xBeta).

  probit <- pnorm(xBeta)
  # ------------------------------------------------------------------------#
  # Nu kan log likelihoodfunktionen skrivas för hand. Detta är som visades tidigare
  # den log likelihoodfunktion som jag uppvisade i min teori del i uppgiften.
  # Men som sagt, p_i BYTS NU UT MOT SANNOLIKHETERNA FRÅN NORMALFÖRDELNINGEN.

  # y i koden står för responsvektorn; det vill säga den oberoende variabelns
  # observationer.
  log_likelihood_funktionen <- y %*% log(probit) + (1-y) %*% log(1-probit)

  # SISTA STEG: retunera

  # Notera att vi tar - log likelihood funktionen här. Detta har inget med teorin
  # att göra. Det behövs eftersom att funktionen optim() i R används för att
  # minimera en funktion. Men genom att sätta negativ så byter vi håll, alltså
  # vi kommer maximera istället för minimera!
  return(-log_likelihood_funktionen)
  }

# Nu behövs startvärden för att köra optimeringen. Det rekommenderas att inte
# använda startvärden som är nollor eftersom att detta kan skapa problem
# med derivatorna. Let's put it to test och se om det faktiskt gör så!

start <- matrix(c(0,0,0,0),nrow = 4,ncol = 1)

# OPTIMERINGEN SKER NU:

  # Det är nu optimeringen sker. Jag använder funktionen optim() som kommer
  # att maximera så jag får MLE för varje parameter.

optimering <- optim(
  # Mina startvärden för optimeringen
  par =start,
  # Min funktion för optimeringen
  fn = LogLikelihoodFunktionen)

# Skapar en tabell över utskrifterna
```

```
  # OBS: kan inte använda piping pga select inte kan hantera listor.

# Utskrift för optimeringen
optimering
```

```
## $par
##             [,1]
## [1,] -1.8900385
## [2,]  0.4315618
## [3,]  0.1458181
## [4,] -0.1015801
##
## $value
## [1] 336.5006
##
## $counts
## function gradient
##      313       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```
probit_modell <- glm(
  # Börjar med att definera modellen
  UnOpen ~ PowerSeller + LogBookValue + StartBidRatio,
  # Hänvisar till rätt DF
  data = eBay,
  # Säger åt vilken familj av fördelningar jag vill använda, där jag specificerar
  # att jag vill använda probit som länkfunktion.
  family = binomial(link="probit")
)

summary(probit_modell) %>%
  coef() %>% kable(caption ="Parameterskattningar för probit modellen.")
```

Tabell 2: Parameterskattningar för probit modellen.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.8902619	0.3433683	-5.505056	0.0000000
PowerSeller	0.4317366	0.1176166	3.670713	0.0002419
LogBookValue	0.1458797	0.0850544	1.715134	0.0863207
StartBidRatio	-0.1015077	0.1756366	-0.577942	0.5633032

```r
# Jämför mellan optimeringsalgortimen och glm funktionens MLE
DF <- data.frame(
  Optimering = round(optimering$par,4),
  glm = round(coef(probit_modell),4)
)

kable(DF, caption ="Skillnaden i parameterskattningarna mellan optimeringsalgortimen och glm MLE.")
```

Tabell 3: Skillnaden i parameterskattningarna mellan optimeringsalgortimen och glm MLE.

	Optimering	glm
(Intercept)	-1.8900	-1.8903
PowerSeller	0.4316	0.4317
LogBookValue	0.1458	0.1459
StartBidRatio	-0.1016	-0.1015

2.2 Deluppgift 1b)

Beräkna den skattade sannolikheten för UnOpen = 0 i auktion nr 200 (rad 200 i data). Jämför den skattade sannolikheten med det observerade värdet på responsvariabeln och kommentera rimligheten i den skattade sannolikheten. (1p)

Nu beräknas sannolikheten att för att myntförpackningen är öppnad, alltså, P(Y=0). För beräkningen använder jag koden i deluppgift 1a) där jag skapade (bl.a) designmatrisen. Innan jag förklarar med går jag bara igenom teorin kort. Teorin säger att för att få fram sannolikheten att $P(Y = 1)$ kan man beräkna:

$$P(Y = 1) = \Phi(\boldsymbol{x\beta})$$

alltså standard normalfördelningens fördelningsfunktion. Vidare plockas rad 200 ut ur designmatrisen och gångras ihop med koefficienterna från optimeringen (obs: glm coefs() är samma som min optimering så reslutatet är densamma). Det som plockades ut är $\boldsymbol{x\beta}$. Nu kan sannolikheten beräknas genom att ta $1 - \Phi(\boldsymbol{x\beta})$. Observera att $1 - P(Y = 1) = P(Y = 0)$ vilket är det man söker i denna uppgiften. I tabell 4 presenteras sannolikheten tillsammans med det riktiga observera värdet på responsvariabeln. Sannolikheten att myntförpackningen på rad 200 är öppnad är 84.5% och det observerade värdet är 0. Den skattade sannolikheten säger att myntförpackningen med stor sannolikhet bör vara öppnad, vilket den också är. Det vill säga, svaret är mycket rimligt.

Först plockar jag ut alla observationer från designmatrisen på rad 200 och gångrar ihop dessa med parametrarna från min optimering (optimeringen gav samma som glm). Nu har jag allt som behövs för att beräkna sannolikheten, och det är nu som normalfördelningens täthet kommer till användning (se kod). Eftersom att man från $\phi()$ har gett sannolikheten

```
### Deluppgift 1b) ------------------------------------------------------------#
# Det är givet från defintionen av modellen att sannolikheten att ett observerat
# värde är lika med 0, är P(Y=0) = 1- p.

# Som nämnades tidigare är p = Phi(xBeta). Det vill säga,

# Låt oss plocka ut observationen från designmatrisen:
  # - Vi ska kolla på rad 200:

observation_200 <- designmatris[200,]

# Men eftersom vi är intresserade av den skattade sannolikheten kan måste man ta
# observationerna på rad 200 gånger parameterskattningarna.
observerad_skattning <- observation_200 %*% optimering$par

# Nu använder vi normalfördelningen för att få fram sannolikheten att en
# myntförpackning är oöppnad.

# Men som nämnades tigiare har vi inte tagit sannolikheten att det sannolikheten
# för de observationerna på rad 200, utan just nu har vi \hat{P}(Y=1). Så vi
# 1 minus observerad_skattning för att få rätt

sannolikheten_öppen <- 1 - pnorm(observerad_skattning)
```

```r
# Nu skapar jag en tabell som blir som en jämförelse mellan sannolikheten
# och det faktiska observerade värdet på responsvariabeln i rad 200.

df_jämförelse <- data.frame(
  sannolikhet_öppen = sannolikheten_öppen,
  Observerad_y = y[1,200]
)

kable(df_jämförelse, caption ="Jämförelse av den beräknade sannolikheten och
      det observerade värdet på responsvariabeln. ")
```

Tabell 4: Jämförelse av den beräknade sannolikheten och det observerade värdet på responsvariabeln.

sannolikhet_öppen	Observerad_y
0.8479546	0

2.3 Deluppgift 1c)

Beräkna den förväntade förändringen av oddset då variabeln LogBookValue ökar med en enhet, givet att de övriga variablerna hålls konstanta. Tolka resultatet i ord. (1p)

I denna uppgift har nu en logit modell anpassats istället som ser ut som följande:

$$\log\left(\frac{P(Y=1)}{P(Y=0)}\right) = \beta_0 + \beta_1 \text{PowerSeller} + \beta_2 \text{LogBookValue} + \beta_3 \text{StartBidRatio}$$

för att lösa denna uppgiften tar man parameterskattningen och transformerar tillbaka den för att kunna tolka den. Från tabell 3, noteras det att LogBookValue är 0.2644, alltså $\exp(0.2644) = 1.3027$. Detta kan tolkas som att oddset ökar med 30.27% för varje ökning i LogBookValue, givet att resterande variabeler hålls konstanta.

```
### Deluppgift 1c) ------------------------------------------------------------#

# Skattar en logit modell ist

logit_modell <- glm(UnOpen ~ PowerSeller + LogBookValue + StartBidRatio,
                    data = eBay,
                    family = binomial(link="logit"))

# Kort sagt, vi vill få fram oddskvoten för LogBookValue. Detta kan enkelt
# göras genom att skatta modellen (klart) och ta fram skattningen för denna
# variabel sedan transformera tillbaka den.

skattningar <- coef(logit_modell)
oddskvoten_LogBookValue <- exp(skattningar[3])
kable(skattningar, caption ="Parameterskattningar för logit modellen.")
```

Tabell 5: Parameterskattningar för logit modellen.

	x
(Intercept)	-3.3229681
PowerSeller	0.8585447
LogBookValue	0.2644950
StartBidRatio	-0.2356561

```
kable(oddskvoten_LogBookValue, caption ="Oddskvoten för LogBookValue")
```

Tabell 6: Oddskvoten för LogBookValue

	x
LogBookValue	1.302773

2.4 Deluppgift 1d)

Givet medelvärdet av de övriga variablerna i modellen, beräkna hur många gånger mer sannolikt det är för UnOpen = 0 om säljaren inte är lyckosam (PowerSeller = 0) jämfört med om säljaren är lyckosam (PowerSeller= 1). Tolka resultatet i ord. (1p)

I denna deluppgift beräknas hur många gånger mer sannolikt det är att UnOpen = 0 om säljaren inte är lyckosam (PowerSeller = 0) jämfört med att säljaren är lyckosam (PowerSeller = 1). Man vill beräkna följande:

$$\frac{P(\text{UnOpen=0}|\ \text{PowerSeller} = 0)}{P(\text{UnOpen} = 0|\ \text{PowerSeller} = 1)}$$

I min kod börjar jag med att beräkna sannolikheten när $PowerSeller = 0$. Detta är variabeln som heter "PowerSeller0" som man kan notera lägger jag även till medelvärdena för resterande kovariater som angivet i uppgiften. I nästa del av koden skapar jag variabeln "PowerSeller1". Här skapar jag nu $P(PowerSeller = 0)$. Men det som jag har beräknat tidigare är förstås log odds, för att omvandla dessa till sannolikheter behöver man omvandla tillbaka detta görs i variablerna "sannolikhet_0" och "sannolikhet_1". Detta blir då sannolikheterna som efterfrågas i uppgiften. Nu när sannolikheterna har fåtts kan jag beräkna sannolikheterna:

$$\frac{P(\text{UnOpen=0}|\ \text{PowerSeller} = 0)}{P(\text{UnOpen} = 0|\ \text{PowerSeller} = 1)} = \frac{0.9319134}{0.8529482} = 1.092579$$

I tabellen nedan presenterar jag sannolikheterna och oddskovten. Sannolikheten att $UnOpen = 0$ när $PowerSeller = 0$ är 0.9319. Sannolikheten att $UnOpen = 0$ när $PowerSeller = 1$ är 0.8629. Kvoten blir här 1.09. I formlen ovanför ser man hur det ska tolkas. Alltså det är 1.09 gånger mer sannolikt att en myntförpackning är öppnad när en säljare på eBay inte är lyckosam jämfört med en säljare som är lyckosam.

```
### Deluppgift 1d) ------------------------------------------------------------#

# Börjar med att dra ut koefficienterna
koef<- coef(logit_modell)

# Det jag beräknat här under är att PowerSeller blir noll. Därför är den inte med
# pga om x= 1, så x * beta = 0 * beta = 0!!
# OBSERVERA: frågan vill ha medelvärden av kovariaten drf tar jag mean()
PowerSeller0<- koef[1] + koef[3] * mean(eBay$LogBookValue) + koef[4]*mean(eBay$StartBidRatio)

# Nu vänder vi på det och tar med att PowerSeller = 1
# Tar PowerSeller0 + koef[1] alltså jag lägger till PowerSeller = 1, och eftersom
# att PowerSeller = 0 i tidigare så är det samma modell fast nu med PowerSeller
# inkluderad. Gick bara snabbare att skriva, kan vara svårt att förstå direkt.
PowerSeller1 <- PowerSeller0 + koef[2]

# Nu använder jag den logistiska fördelningsfunktionen för beräkningen!

sannolikhet_0 <- 1/(1 + exp(PowerSeller0))
sannolikhet_1 <- 1/(1 + exp(PowerSeller1))

# Nu är det lite tydligare att jag faktiskt beräknade oddskvoten för y = 0 mot 1
oddskvoten <- sannolikhet_0 / sannolikhet_1
```

```r
# Skapar en DF
df_odd <- data.frame(oddskvot = oddskvoten,
                     Sannolikhet0 = sannolikhet_0,
                     Sannolikhet1 = sannolikhet_1)
kable(df_odd, caption ="Oddskvoten med sannolikheterna.")
```

Tabell 7: Oddskvoten med sannolikheterna.

	oddskvot	Sannolikhet0	Sannolikhet1
(Intercept)	1.092579	0.9319134	0.8529482

2.5 Deluppgift 1e)

Utför ett likelihoodkvottest om minst en av de variablerna LogBookValue och StartBidRatio bidrar till att förklara sannolikheten för att myntförpackningen lever- eras sluten och oöppnad i dess originalförpackning, givet att PowerSeller redan finns med i modellen. Redovisa lösningen som en fullständig hypotesprövning med relevanta formler och beräkningar. Ni får använda utskrifter som stöd i beräkningarna men var tydlig med varifrån värden som används kan avläsas. (1p)

Givet att man kan teorin är denna uppgiften mycket simpel, man utnyttjar log likelihooden för två modeller. En reducerad mdoell och en full modell. Man är intresserad av att dra en slutsats om LogBookValue och StartBidRatio faktiskt bidrar till modellen. Därför skapas en full modell med alla variabler och en reducerad modell utan variablerna. Hypoteserna blir därför:

$$H_0 : \beta_2 = \beta_3 = 0 \quad H_A : \text{Minst en } \beta_i \neq 0$$

Där LR testet blir

$$LR = -2(-339.0260 - (-336.3368)) = 5.3784 \quad \sim \chi^2_2 = 5.99$$

Det vill säga att med 5% signifkans kan man inte bekräfta att LookBookValue och StartBidRatio bidrar signifikant till modellen. Svaret tyder på att en modell med endast PowerSeller är att föredra. Både det kritiska värdet och statistikan, tillsammans med p-värde visar att man är mycket nära på att förkasta nollhypotesen, därför kan det vara av intresse att undersöka detta djupare. Till exempel genomföra parvisa tester för respektive koefficienter.

```
### Deluppgift 1e) ------------------------------------------------------------#

# Jag slutar nu beskriva teorin i koden för det tar för långt tid. Jag fokuserar
# på att göra detta i dokumentet ist.

# Skapar de fulla och restrikterade modellerna:
full_modell <- glm(UnOpen ~ PowerSeller + LogBookValue + StartBidRatio,
                   data = eBay,
                   family = binomial(link="logit"))

restrikterad_modell <- glm(UnOpen ~ PowerSeller,
                   data = eBay,
                   family = binomial(link="logit"))


# Utför log-likelihoodkvot testet
logkvot_test <- lrtest(restrikterad_modell,full_modell)
kable(logkvot_test, caption = "Likelihoodkvot-test för full och
      restrikterad modell.")
```

Tabell 8: Likelihoodkvot-test för full och restrikterad modell.

#Df	LogLik	Df	Chisq	Pr(>Chisq)
2	-339.0260	NA	NA	NA
4	-336.3368	2	5.378535	0.0679307

```
# Beräknar kritiska värdet
qchisq(0.95, df = 2)
```

```
## [1] 5.991465
```

2.6 Deluppgift 1f)

Utför ett motsvarande Waldtest för det som undersöktes i e) genom att för hand (utan att använda inbyggda funktioner) beräkna värdet på Walds teststatistika med hjälp av skattningarna från logit-modellen. Avgör sedan om minst en av förklaringsvariablerna LogBookValue och StartBidRatio bidrar till att förklara sannolikheten för att myntförpackningen levereras sluten och oöppnad i dess originalförpackning, givet att PowerSeller redan finns med i modellen. (2p)

I denna deluppgift beräknar jag Wald testet manuellt med hjälp av coef() och vcov() i R. Det vart mycket simpelt att genomföra denna beräkning, jag valde att använda samma namn på variablerna som det automatiska Wald testet. I R, vilket är restriktionsmatrisen presenterar jag de restriktioner jag vill sätta. Jag har skrivit koden på så sätt att man kan se tydligt vilken variabel jag restrikterar, alltså LogBook. Det komplicerade i denna kod är att förstå hur restriktionerna faktiskt sätts. Anledningen till varför det finns 2 rader i R bind är eftersom jag vill undersöka två variabler. Om man hade velat undersöka tre variabler skulle man lagt till en till rad (alltså tre restriktioner). Det viktiga här är att det måste finnas ett linjärt oberoende; alltså en rad inte kan återskapas av en annan. Enklare förklarat innebär detta att man inte kan ha t.ex två rader som är (0,0,0,1) och (0,0,0,1) för då finns ett linjärt beroende.

Vidare sätts matris q upp vilket visar de nollhypoteser som ska testas, där man vill se i nollhypotesen om parametrarna bidrar med någon effekt (=0). Statistikan kan sedan beräknar med hjälp av ginv(), det provades en annan funktion solve() men det gick inte att beräkna W med den. Se här:

$$\boldsymbol{R} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_3 \end{pmatrix} \quad \boldsymbol{R\beta} = \boldsymbol{q} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Med följande hypoteser:

$$H_0 : \beta_2 \text{ och } \beta_3 = 0 \quad H_A : \text{Minst en parameter} \neq 0$$

Wald testet blev 5.45 och statistikan beräknade och blev 5.99. Nollhypotesen kan inte förkastas på 5% signifkans, och därför kan man inte se att LogBookValue och StartBid bidrar med någon effekt givet att PowerSeller redan är med i modellen.

```
### Deluppgift 1f) ------------------------------------------------------------#

# I denna del av uppgiften ska W statistikan beräknas för hand. Jag väljer
# därför att kolla på wald.test() funktionen och tar variabelnamn utifrån
# vad den har valt för namn för de!

b <- coef(full_modell)
sigma <- vcov(full_modell)

# Jag måste själv definera matris R och q

# Istället matrix() använder jag cbind() pga att man ser vad respektive rad
# faktiskt innebär:

R <- rbind(
  # Rad 1: Först vill vi undersöka om LokBookValue = 0 drf 1 på dens plats!
  c(0,0,1,0), # ORDER: Intercept, PowerSeller, LogBook, StartBid
  # Rad 2: Sist vill vi undersöka om StartBid = 0, drf 1 på sista platsen
```

```
  c(0,0,0,1)) # ORDER: Intercept, PowerSeller, LogBook, StartBid

# Nollhypotesen: dvs 2 restriktioner
q <- c(0,0)

# Statistikan:
# W <- t(R %*% b - q) %*% solve(R %*% sigma %*%t(R)) %*% (R %*% b - q)
# OBS: måste använda #ginv() ist för solve() annars kan inte statistikan beräknas
# detta är pga systemet blir singulärt.
W <- t(R %*% b - q) %*%  ginv(R %*% sigma %*% t(R)) %*% (R %*% b - q)

# Beräknar kritiska värdet med 2 restriktioner = k:
chi2 <- qchisq(0.95,df = 2)

df_wald <- data.frame(
  W = W,
  chi2 = chi2
)

kable(df_wald, caption ="Wald statistika med tillhörande kritiskt värde.")
```

Tabell 9: Wald statistika med tillhörande kritiskt värde.

W	chi2
5.45435	5.991465

2.7 Deluppgift 1g)

Jämför resultaten av likelihood kvottestet i uppgift e) och Waldtestet i uppgift f) utifrån olika val av signifikansnivåer och tolka dem i ord. Ger testen samma resultat? Varför eller varför inte? (1p)

I denna del av uppgiften jämförs nu Wald och LRT testerna med olika signifkansnivåer. Jag har valt att undersöka om nollhypoteserna för respektive tester kan förkastas på 5% signifikansnivå och 10% signifkansnivå.

Reslutatet blev att för Wald testet och LRT testet kunde inte nollhypoteserna förkastas på 5% signifkansnivå, vilket innebär attman kan ej kan dra slutsaten om att LogBookValue och StartBidRatio bidrar till modellen på 5% signifkans givet att PowerSeller redan är med. Hur som helst kunde man se att på 10% signifkans kan nollhypoteserna förkastas. Det kan tolkas som att med 10% signifkans bidrar parametrarna signifkant till modellen, givet att PowerSeller redan är med.

Självklart bör det diskuteras om testet på 10% signifkans faktiskt är tillförlitligt, eftersom att man vet att nollhypoteserna inte kunde förkastas på 5% signifkans. Anledningen till detta är mest troligt att effekten blir stor eftersom att man testar två variabler samtidigt, effekten är inte tillräckligt stor vid 5% signfikansnivå men är vid 10%. Det vill säga att ensamt avviker inte parametrarna något märkvärdigt, men de gör dem tillsammans. Ibland kan dock en signifkansnivå på över 5% vara lämplig, detta diskuteras i boken Statistical Rethinking: A Bayesian Course with Examples in R and Stan av Richard McElreath (2020). Men jag kommer dessvärre inte ihåg vilken sidan. Men att om man vet att en specifik parameter inte kommer att finna vid en

viss gräns (5% signfikans), kan det vara ett bättre alternativ att ta en högre gräns. Men för att kunna göra sådant påstående skulle jag behöva känna till datamaterialet väl, vilket jag inte gör. Avslutningsvis, vid detta tillfälle kan man undersöka parametrarna separat för att se om, till exempel, en parameter bidrar signifkant mer än den andra.

```
### Deluppgift 1f) ------------------------------------------------------------#

# Nu återanvänder jag chi2 värdet på LRT och kollar olika signifikans nivåer:

df_signifikans_RT <- data.frame("Chi2" = round(logkvot_test$`Pr(>Chisq)`[2],5),
           alpha = c(0.05,0.1),
           "Val" = c("Förkasta ej"," Förkasta")

)

df_signifikans_Wald <- data.frame("Chi2" = c(qchisq(0.95,df=2),qchisq(0.90,df=2)),
                                  alpha = c(0.05,0.1),
                                  "Val" = c("Förkasta ej"," Förkasta"))

# Skriver ut
kable(df_signifikans_RT, caption ="Reslutat för LRT vid olika signifikansnivåer.")
```

Tabell 10: Reslutat för LRT vid olika signifikansnivåer.

Chi2	alpha	Val
0.06793	0.05	Förkasta ej
0.06793	0.10	Förkasta

```
kable(df_signifikans_Wald,caption ="Reslutat för Wald test vid olika signifikansnivåer")
```

Tabell 11: Reslutat för Wald test vid olika signifikansnivåer

Chi2	alpha	Val
5.991465	0.05	Förkasta ej
4.605170	0.10	Förkasta

3 Uppgift 2

I uppgift två presenteras ett datamaterial över olika typer av viner, tillsammans med deras alkoholhalt och styrkan på dess färg (tabell 11).

```
### UPPGIFT 2 ------------------------------------------------------------###
vin <- read.csv2("/Users/hampusbeijer/Downloads/vin.csv")

kable(head(vin,10), caption ="Datamaterialet över tre sorters vin,
      alkohol i procent, och styrkan på dess färg.")
```

Tabell 12: Datamaterialet över tre sorters vin, alkohol i procent, och styrkan på dess färg.

Type	Alcohol	Color
1	14.23	5.64
1	13.20	4.38
1	13.16	5.68
1	14.37	7.80
1	13.24	4.32
1	14.20	6.75
1	14.39	5.25
1	14.06	5.05
1	14.83	5.20
1	13.86	7.22

3.1 Bakomliggande teori för deluppgift 2a)

Multinomial logit modell

När man vill modellera en beroende variabel med fler än två nivåer fungera inte en vanlig logit eller probit modell längre. Lösningen till detta är multinomial logit modelll. Tyvärr är här slutet för probit modellen, eftersom att en multinomial probit modell skulle innebära att man behöver beräkna multipla integraler. Det specifika problemet är att de avancerade integralerna är svåra att beräkna, och är därför inte lämplig. Men som tur finns den multinomiala logit modellen, där man modellerar en beroende variabel Y med K nivåer. Modellen defineras enligt:

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

Anledningen till varför summeringen börjar vid $k = 2$ är på grund av att en kategori används som referenskategori. Om man väljer referenskategorin till kategori 1 innebär detta konsekvent att $\beta_1 = 0$. Detta innebär att modellformuleringen kommer att förändras eftersom att $e^{\boldsymbol{x}\boldsymbol{\beta}}$ kommer att bli e^0 när $\beta_1 = 0$. Alltså fås (Hietala, 2026, s.24).:

$$P(Y = 1) = \frac{e^{\mathbf{0}}}{1 + \sum_{k=2}^{K} e^{\boldsymbol{x}\boldsymbol{\beta}_k}} = \frac{1}{1 + \sum_{k=2}^{K} e^{\boldsymbol{x}\boldsymbol{\beta}_k}}$$

Loglikelihood funktionen för multinomial logit modellen

Modellens loglikelihood funktion defineras enligt:

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

här defineras d_{ik} som en dummyvariabel, det vill säga, d_{ik} kan antingen anta värdet 0 eller 1 (s.25):

$$d_{ik} = \begin{cases} 1, & \text{om vin } i \text{ tillhör kategori } k, \\ 0, & \text{annars.} \end{cases}$$

Oddset för den multinomiala logit modellen

För att jämföra kategorier används odds. Oddset för den multinomiala modellen defineras som:

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

Här är j och k kategorier man jämför, till exempel: $\frac{P(Y=1)}{P(Y=2)}$ kan tolkas som hur mer sannolikt det är att ett vin är av nivå 1 än nivå 2 (s.25).

Likelihoodkvotindex

Likelihoodkvotindexx är ett ett mått på hur bra den anpassade modellen är jämfört med en med endast ett intercept. LRI använder loglikelihooden för att beräkna värdet enligt:

$$\text{LRI} = 1 - \frac{\ln \ell}{\ln \ell_0}$$

Där ett $LRI = 1$ indikerar att modellen anpassar mycket bra till datan, men mycket dåligt om $LRI = 0$. Det vill säga att om LRI värdet är mycket nära noll är en modell utan förklarande variabler bättre än en modell som inkluderar dem (s.22).

3.2 Deluppgift 2a)

Skatta sannolikheten för att ett vin med 13% alkoholhalt och färgintensitet 5 är av typ 3. (2p)

Man vill beräkna sannolikheten att $Y = 3$ givet att Alkohol $= 13$ och Color $= 5$, alltså:

$$P(Y = 3|\text{Alkohol} = 13, \text{Color} = 5) = \frac{e^{(\hat{\beta}_{30}+13\hat{\cdot}\beta_{31}+5\cdot\hat{\beta}_{32})}}{1 + e^{(\hat{\beta}_{20}+13\hat{\cdot}\beta_{21}+5\cdot\hat{\beta}_{22})} + e^{(\hat{\beta}_{30}+13\hat{\cdot}\beta_{31}+5\cdot\hat{\beta}_{32})}}$$

För att genomföra detta skattas modellen i R med hjälp av funktionen multinom() där typen av vin är responsvariabeln med alkoholhalt och färg som kovariater. Det kan noteras att modellen når konvergens i utskriften vilket innebär att modellen lyckades skattas.

Formeln för att beräkna denna sannolikhet visades tidigare, där man kan notera att koefficienterna för modellen behövs för beräkning. Dessa plockas ut med hjälp av functionen coef(), sedan indexeras rätt koefficienter ut och formeln återskapas i kod för att kunna beräknas. I tabell 12 presenteras reslutatet att sannolikheten att vinet är av typ 3 givet att alkholhalten är 13% med en färgstyrka på 5 enheter är 44.6%.

```
### Deluppgift 2a) ------------------------------------------------------------#

# För denna uppgift vill man modellera den den beroende variabeln, type,
# har tre kategorier drf kan man inte använda en vanlig logit eller probit
# -modell utan vi får använda en multinomial modell.

# Nu skattar jag en mutlinomial logit modell där jag automatiskt kommer bli
# tilldelad en referenskategori, och detta blev nivå 1.

vin$Type <- factor(vin$Type) # Gjorde ingen skillnad (man behöver inte factor())
multinomial_logit <- multinom(Type ~ Alcohol + Color,
                              data = vin)
```

```
## # weights:  12 (6 variable)
## initial  value 195.552987
## iter  10 value 84.226348
## iter  20 value 71.316253
## iter  30 value 70.912467
## iter  40 value 70.907690
## iter  40 value 70.907690
## final  value 70.907690
## converged
```

```
# DET VIKTIGA:
# Vi vill alltså beräkna sannolikheten då Y = 3 givet att alcohol = 13 och color
# = 5. För detta behöver vi koefficienterna från modellen.

coefs <- coef(multinomial_logit)

# FÖR ATT FÖRSTÅ: se formel i uppgiften
```

```r
# Nivå 2 parameterskattningar
beta20 <- coefs[1,1]
beta21 <-  coefs[1,2]
beta22 <- coefs[1,3]

# Nivå 3 parameterskattningar
beta30 <- coefs[2,1]
beta31 <-  coefs[2,2]
beta32 <- coefs[2,3]

nivå2 <- (beta20 + 13 * beta21 + 5 * beta22)
nivå3 <- (beta30 + 13 * beta31 + 5 * beta32)

sannolikhet <-  exp(nivå3) / (1 + exp(nivå2) + exp(nivå3))

kable(sannolikhet, caption ="Sannolikheten att ett vin med 13 procent
      alkoholhalt, 5 på färgintensitet är av kategori 3.")
```

Tabell 13: Sannolikheten att ett vin med 13 procent alkoholhalt, 5 på färgintensitet är av kategori 3.

x
0.4460262

3.3 Deluppgift 2b)

Beräkna oddset för att ett vin är av typ 3 gentemot typ 2 för ett vin med 14% alkoholhalt och färgintensitet 6. Motivera dina beräkningar och tolka oddset i ord. (2p)

Nu vill man beräkna hur mycket mer sannolikt (odds) att ett vin av nivå 3 är jämfört med nivå 2. Tidigare visades det i teorin visades definitonen av odds för multinomial modellen. Det blir nu:

$$\frac{P(Y = 3|\text{Alkohol} = 14, \text{Color} = 6)}{P(Y = 2|\text{Alkohol} = 14, \text{Color} = 6))} = \exp[X(\beta_3 - \beta_2)]$$

Observera att β_2 och β_3 står för alla parametrarna för nivå 3 respektive nivå 2. Det beräknas nu enligt:

```r
### Deluppgift 2b) ------------------------------------------------------------#
# Nu ska oddset beräknas för ett vin av nivå 3 jämfört med 2 med alkohol = 14
# och färg = 6.
# Koden som användes i tidigare uppgift kan nu återanvändas måste bara ändra
# siffrorna något

nivå2 <- (beta20 + 14 * beta21 + 6 * beta22)
nivå3 <- (beta30 + 14 * beta31 + 6 * beta32)

odds <- exp(nivå3 - nivå2)
```

Reslutatet presenteras i tabell 13, vilket är 82. Detta kan tolkas som att oddset för att ett vin med alkoholhalt på 14% och färgstyrka på 6 är av typ 3 jämfört med typ 2 är drygt 82. Detta är ett mycket högt odds, som visar att oddset för att det faktiska vinet är av typ 3 jämfört med typ 2 är markant.

```
kable(odds, caption ="Oddset för att ett vin är av nivå 3 jämfört med nivå 2
      om det har 14 procent alkoholnivå och färg 6.")
```

Tabell 14: Oddset för att ett vin är av nivå 3 jämfört med nivå 2 om det har 14 procent alkoholnivå och färg 6.

x
82.27808

3.4 Deluppgift 2c)

Beräkna värdet på likelihoodkvotindexet (LRI) för modellen. (1p)

I denna deluppgift beräknas ett likelihoodkvotindex, även känt som ett LRI test. Detta test jämför om en full modell är bättre än en tom modell. Med en full modell kan inkludera en eller fler kovariater. Den tomma modellen har inga kovariater, utan modellerar endast ett intercept. För beräkningen av LRI behövs loglikelihood från respektive modeller och detta fås genom att skatta modellerna och använda funktionen logLik() som plockar ut modellernas loglikelihood. Testet är:

$$\text{LRI} = 1 - \frac{\ln \ell}{\ln \ell_0}$$

där $\ell = -70.91$ är log likelihood för den fulla modellen och $\ell_0 = -193.31$ är loglikelihood för den tomma modellen. Då beräknas LRI enligt:

$$\text{LRI} = 1 - \left(\frac{-70.93}{-193.31}\right) = 0.63$$

det vill säga att $LRI = 0.63$. Gränsen för detta test säger att ett LRI som ligger nära noll indikerar på att en modell utan kovariater är att föredra, men en modell nära 1 är den mest ideala. LRI testet är relativt hög och man kan dra slutsatsen att en tom modell inte hade varit bättre. Det bör även nämnas att LRI testet får inte tolkas som procent.

```
### Deluppgift 2c) ------------------------------------------------------------#

# För att beräkna LRI behövs en full modell och en modell med utan
# förklarande variabler.

full_modell <- multinom(Type ~ Alcohol + Color,
                        data = vin)
```

```
## # weights:  12 (6 variable)
## initial  value 195.552987
## iter  10 value 84.226348
## iter  20 value 71.316253
```

```
## iter  30 value 70.912467
## iter  40 value 70.907690
## iter  40 value 70.907690
## final  value 70.907690
## converged
```

```
noll_modell <- multinom(Type~1,
                        data = vin)
```

```
## # weights:  6 (2 variable)
## initial  value 195.552987
## final  value 193.314843
## converged
```

```
# Beräknar nu LRI
# OBS: måste ha as.numeric annars blir uträkningen ingen siffra.
full <- logLik(full_modell)
noll <- logLik(noll_modell)

kvot <- as.numeric(full/noll)
LRI <- 1 - kvot

df_lri <- data.frame(LRI = LRI)

kable(LRI, caption="Loglikelihoodkvotindex för multinomial logitmodellen.")
```

Tabell 15: Loglikelihoodkvotindex för multinomial logitmodellen.

x
0.633201

4 Uppgift 3

```
### UPPGIFT 3 ----------------------------------------------------------------###
fattigdom <- read.csv2("/Users/hampusbeijer/Downloads/fattigdom.csv")
kable(head(fattigdom,n=10), caption="Datamaterial över fattigdom kopplad med
      religion, utbildning, ålder och, och kön.")
```

Tabell 16: Datamaterial över fattigdom kopplad med religion, utbildning, ålder och, och kön.

poverty	religion	degree	age	gender
Too Little	yes	no	44	male
About Right	yes	no	40	female
Too Little	yes	no	36	female
Too Much	yes	yes	25	female
Too Little	yes	yes	39	male
About Right	yes	no	80	female
Too Much	yes	no	48	female
Too Little	yes	no	32	male
Too Little	yes	no	74	female
Too Little	yes	no	30	male

4.1 Bakomliggande teori för uppgifterna

Grunden till den ordinala logit modellen (latent regression)

När man vill modellera en beroende variabel vars värden följer en specifik ordning är inte längre en vanlig logit modell lämplig. Lösningen är en ordnad logit modell som kan ta hänsyn till ordningen i responsvariabeln. En ordinal logit modell bygger på latent regression, där man har en latent variabel som vars skala är kontinuerlig men uppdelade i kategorier. Latent regression defineras enligt:

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

Här är y^* den latenta variabeln, $\boldsymbol{x}\boldsymbol{\beta}$ är de förklarande variablerna med dess parametrar, och ϵ är feltermen.

Det viktiga med y^* är att den inte är observerbar, och därför måste man utgå från y som finns i ens eget datamaterial, enligt:

$$y = \begin{cases} 1, & \text{om } y^* \leq \mu_1, \\ 2, & \text{om } \mu_1 < y^* \leq \mu_2, \\ 3, & \text{om } \mu_2 < y^* \leq \mu_3, \\ \vdots & \\ K, & \text{om } y^* > \mu_{K-1} \end{cases}$$

För denna uppgiften (som exempel) blir $y = poverty$ som har tre kategorier Too Little, About Right, och Too Much. Sedan måste denna variabel delas in i intervall på y^* vilket kommer se ut som (Hietala, 2026, s. 28):

$$y = \begin{cases} 1, & \text{om } y^* \leq \mu_1, \\ 2, & \text{om } \mu_1 < y^* \leq \mu_2, \\ 3, & \text{om } y^* > \mu_2 \end{cases}$$

Ordinal logit modell

Nu antar man att den logistiska fördelningsfunktionen (CDF) för ϵ är (s.29):

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

Som man kan notera är detta samma definition som pratades om i tidigare delar (s. 11):

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

Med hjälp av fördelningsfunktionen kan man därför beräkna följande sannolikheter:

$$P(y = 1) = \Lambda(\mu_1 - x\beta)$$

$$P(y = 2) = \Lambda(\mu_2 - x\beta) - \Lambda(\mu_1 - x\beta)$$

$$P(y = 3) = \Lambda(\mu_3 - x\beta) - \Lambda(\mu_2 - x\beta)$$

$$P(y = K) = 1 - \Lambda(\mu_{K-1} - x\beta)$$

I denna uppgiften, på grund av att det finns $K = 3$ nivåer blir det:

$$P(y = 1) = \Lambda(\mu_1 - x\beta)$$

$$P(y = 2) = \Lambda(\mu_2 - x\beta) - \Lambda(\mu_1 - x\beta)$$

$$P(y = 3) = 1 - \Lambda(\mu_2 - x\beta)$$

Loglikelihoodfunktionen för den ordinala logit modellen

För att skatta den ordinala logit modellen används MLE (Maximum Likelihood Estimation) för att skatta β, och μ_1, och μ_2 (obs $K = 3$ drf endast $3 - 1 = 2$). Loglikelihoodfunktionen för ordinala logit modellen är (s.29):

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

4.2 Deluppgift 3a)

Skatta sannolikheten att en 35-årig kvinna som tillhör en religion oc ej har universitetsexamen har åsikten "About Right". (1p)

Vi definerar ordinal logit modellen som ska skattas enligt:

$$y^* = x\beta + \epsilon$$

Och eftersom det finns tre nivåer är $K = 3$, vilket konskevent innebär att följande sannolikhet ska beräknas:

$$P(Y = 2) = \Lambda(\mu_2 - x\beta) - \Lambda(\mu_1 - x\beta)$$

Det som är viktigt att förstå här är att $\Lambda()$ är den logistiska fördelningsfunktionen vilket innebär att beräkningen kommer att bli:

$$P(Y = 2) = \Lambda(\mu_2 - x\beta) - \Lambda(\mu_1 - x\beta) = \frac{e^{(\mu_2 - x\beta)}}{1 + e^{(\mu_2 - x\beta)}} - \frac{e^{(\mu_2 - x\beta)}}{1 + e^{(\mu_2 - x\beta)}} = \frac{e^{1.9871}}{1 + e^{1.9871}} = 0.3172$$

Det vill säga, min modell säger att sannolikheten att en 35 årig kvinna som är religös, saknar universitetsexamen, och har åsikten "About Right" är 31.72%. OBS: jag kommentera inte mycket hur jag gick tillväga för detta eftersom jag redan presenterat teorin och kommenterar min kod noggrant för att visa det praktiska utförandet.

```
### Deluppgift 3a) ------------------------------------------------------------#
# beroende variabel har ragordning så vi kan inte använda en vanlig mdoell utan
# vi måste använda en ordinal logit modell.

# Responsvariabeln msåte vara en faktor:
fattigdom$poverty <- factor(fattigdom$poverty,
                            levels = c("Too Little","About Right","Too Much"),
                            ordered = TRUE)
fattigdom$religion <- factor(fattigdom$religion)
fattigdom$degree <- factor(fattigdom$degree)
fattigdom$gender <- factor(fattigdom$gender)

# Skattar modellen mha polr
modell <- polr(poverty ~ religion + degree + age + gender,
               data = fattigdom,
               Hess = TRUE)

# coef() tar ut koefficienterna för modellen
skatt <- coef(modell)
religion_ja <- skatt[1]
degree_ja <- skatt[2]
age <- skatt[3]
gender_male <- skatt[4]

# zeta är här de skattade medelvärdena mu1 respektive mu2
medel <- modell$zeta
```

```r
mu1 <- medel[1]
mu2 <- medel[2]

# Från formel ser vi att man behöver xbeta vilket där:
# age: 35
# kön: kvinna
# religion: tillhör
# utbildning: nej
# åsikt: About Right vilket då blir P(Y=2)

xbeta <- c(age*35 + gender_male*0 + degree_ja*0 + religion_ja*1)

# LOGITISKA FUNKTIONEN:
# Det viktiga att komma ihåg i denna uppgiften är att vi arbetar med den
# logistiska funktionen så nu måste jag manuellt få fram de specifika värdena

sannolikheten_2 <- (
(exp(mu2-xbeta) / (1+exp(mu2-xbeta))) - (exp(mu1-xbeta) / (1+exp(mu1-xbeta)))
)
```

4.3 Deluppgift 3b)

Beräkna oddset för att samma person från a) har åsikten "Too Little" gentemot "Too Much". Motivera dina beräkningar och tolka oddset i ord. (2p)

I denna deluppgift beräknas oddset för att samma person från deluppgift 3a) har åsikten "Too Little" ($Y = 1$) jämfört med "Too Much" ($Y = 3$). Tidigare (i teori delen) presenteras att:

$$P(Y = 1) = \Lambda(\mu_1 - x\beta)$$

och

$$P(Y = 3) = 1 - \Lambda(\mu_{K-1} - x\beta) = 1 - \Lambda(\mu_2 - x\beta)$$

I frågan vill man beräkna oddset för $Y = 1$ jämfört med $Y = 3$, enligt definitonen för ett odds blir detta då:

$$\frac{P(Y=1)}{P(Y=3)} = \frac{\Lambda(\mu_1 - x\beta)}{1 - \Lambda(\mu_2 - x\beta)} = \frac{\frac{e^{(\mu_1 - x\beta)}}{1+e^{(\mu_1 - x\beta)}}}{1 - \frac{e^{(\mu_2 - x\beta)}}{1+e^{(\mu_2 - x\beta)}}} = \frac{\frac{e^{(0.6618-0.4114)}}{1+e^{(0.6618-0.4114)}}}{1 - \frac{e^{(2.3986-0.4114)}}{1+e^{(2.3986-0.4114)}}} = 4.6640$$

Detta tolkas som att oddset att en kvinna är religös, utan universitetsexamen, och har åsikten "Too Litte" jämfört med "Too Much" är 4.66, alltså mer än 4.5 gånger sannolikt.

```r
### Deluppgift 3b) ------------------------------------------------------------#

sannolikheten_1 <- (
  (exp(mu1-xbeta) / (1+exp(mu1-xbeta)))
)
```

```r
sannolikheten_3 <- (
  1 -   (exp(mu2-xbeta) / (1+exp(mu2-xbeta)))
)

ODDS_1_vs_3 <- sannolikheten_1 / sannolikheten_3
```

5 Uppgift 4

5.1 Bakomliggande teori för uppgift 4

Trunkering

Tänk att man har mätt en population, där man studerat en slumpvariabel X som är individers hjärtslag per minut. Men en forskare är endast intresserad av att studera de individer vars hjärtslag är 70 slag per minut. Så forskaren tar bort alla individer vars värden överstiger 70. Detta kallas för trunkering och innebär att man tar bort observationer från en population och konsekvent får en subpopulation.

Täthetsfunktionen för en stokatiskt trunkerad variabel

Täthetsfunktionen för en trunkerad slumpvariabel defineras enligt:

$$f(x|X > a) = \frac{f(x)}{P(X > a)}$$

Där $f(x)$ är en täthetsfunktion och a är en konstant. Anledningen till varför man delar med $P(X > a)$ är eftersom att en täthet måste integreras till 1, viket $P(X > a)$ bidrar med (Hietala, 2026, s.6).

Den trunkerade normalfördelningen När man har en stokatiskt normalfördelad variabel kan man använda standardnormalfördelningen för att beräkans $P(X > a)$ enligt:

$$P(X > a) = 1 - \Phi(\alpha), \quad \alpha = \frac{a - \mu}{\sigma}$$

Vidare kan definera slumpvariabelns täthetsfunktion som:

$$f(x) = \frac{dF(x)}{dx} = \frac{d\Phi(\frac{x-\mu}{\sigma})}{dx} = \frac{1}{\sigma}\Phi(\frac{x-\mu}{\sigma})$$

Nu nör denna täthet är defineras kan den trunkerade normalfördelningen defineras enligt:

$$f(x|X > a) = \frac{f(x)}{1 - \Phi(\alpha)} = \frac{\frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})}{1 - \Phi(\alpha)}$$

Där $\Phi(\alpha)$ är CDF för en standard normalfördelad slumpvariabel och $\phi(\frac{x-\mu}{\sigma})$ är tätheten för en standard normalfördelad variabel (Hietala, s.7).

Förväntat värde och varians på en stokastisk trunkerad variabel

För att beräkna det förväntade värdet av en trunkerade normalfördelad slumpvariabel används följande formel:

$$\mathbb{E}[X|X > a] = \mu + \sigma\lambda(\alpha)$$

Och variansen beräknas genom:

$$\text{Var}[X|X > a] = \sigma^2(1 - \delta(\alpha)) = \text{Var}[X|X > a] = \sigma^2 \left(1 - \left(\frac{\phi(\frac{a-\mu}{\sigma})}{1 - \Phi(\frac{a-\mu}{\sigma})} \cdot \left(\frac{\phi(\frac{a-\mu}{\sigma})}{1 - \Phi(\frac{a-\mu}{\sigma})} - \frac{a-\mu}{\sigma}\right)\right)\right)$$

Där $\lambda(\alpha) = \frac{\phi(\alpha)}{1-\Phi(\alpha)}$, om $x > a$ annars om $x < a$ beräknas $\lambda(\alpha) = -\frac{\phi(\alpha)}{\Phi(\alpha)}$. Sedan kan $\delta(\alpha)$ beräknas genom $\delta(\alpha) = \lambda(\alpha)(\lambda(\alpha) - \alpha)$ där $\alpha = \frac{a-\mu}{\sigma}$.

5.2 Uppgift 4:

I denna uppgift har man en Normalfördelning med medelvärde $\mu = 44$ men variansen är okänd. Hur som helst nämns det att ett värde över 55 uppnås av 9% av populationen. Vi känner alltså till att $P(X > 55) = 0.09$ och därmed vet vi att $P(X \leq 55) = 0.91$. Anledningen varför $P(X \leq 55) = 0.91$ beräknades fram är att den ska användas för att finna variansen i populationen. Varför denna används och inte mindre än femtiofem är att tabellvärden i Normalfördelningen är från vänster sida så det blir enklare att få fram. Vi använder den standardiserade normalfördelningen för att utnyttja värdena som har framtagits:

$$P(Z \leq z) = P(Z \leq \frac{X - \mu}{\sigma})$$

Vi vet om högerledet eftersom att $P(X \leq 55) = 0.91$ och så kan nu σ nu finnas:

$$P(Z \leq \frac{55 - 44}{\sigma}) = 0.91$$

$$P(Z \leq \frac{11}{\sigma}) = 0.91$$

Vi vet att det finns ett värde från den standardiserade normalfördelningen som ger $z = 0.91$. Om man kollar i en bok finner man att det z som ger 0.91 är $z = 1.34$. Detta innebär att $\frac{11}{\sigma} = 1.34$. Multiplicera med σ på bägge sidor för att ändra på ekvationen $(11 = \sigma \cdot 1.34)$ och dela med 1.34 så fås svaret:

$$\sigma = \frac{11}{1.34} = 8.21$$

Nu kan det slumpmässiga urvalet göras med fördelningen $N(44, 8.21)$. Observera att datan har inte blivit trunkerad än.

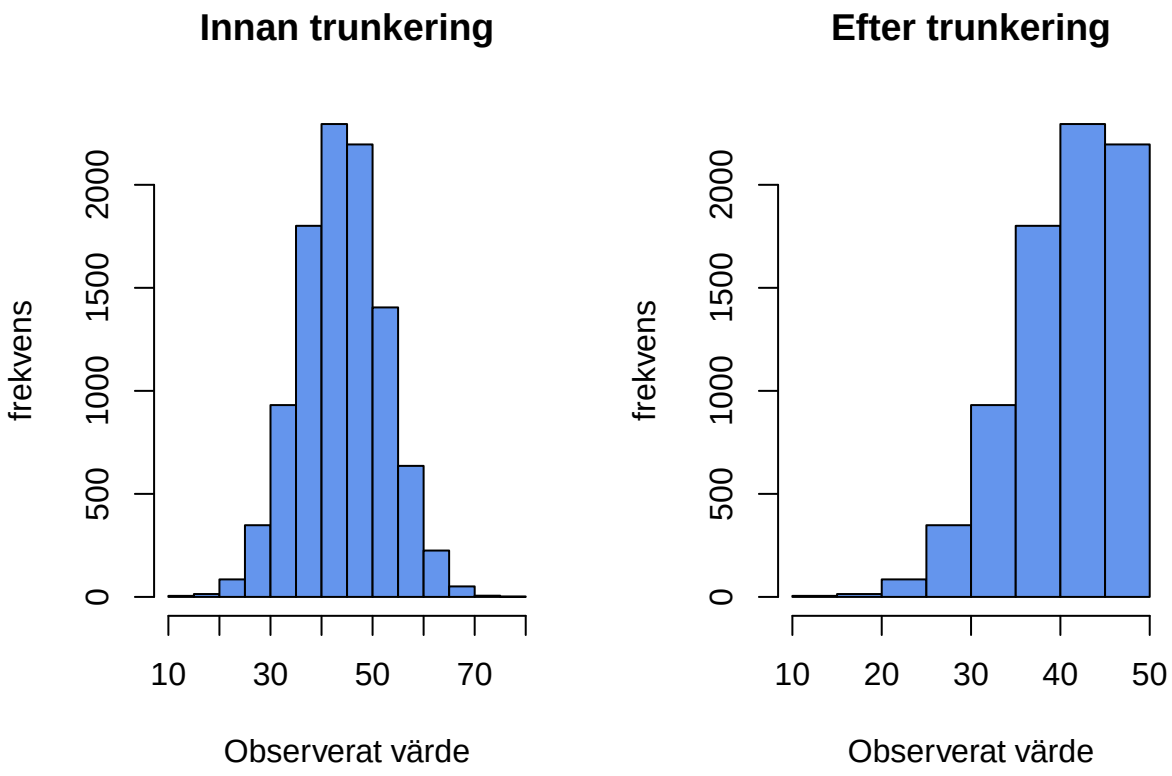
```
### UPPGIFT 4 ----------------------------------------------------------------#
# Drar ett 1e4
sample <- rnorm(1e4,44,8.21)
```

5.3 Deluppgift 4a)

Skapa ett histogram som speglar fördelningen för den del av populationen som har ett mätvärde under 50. Var noga med att redovisa hur du tog fram histogrammet. (1p)

I uppgift 4, ovan, drogs ett sample på tiotusen observationer från populationen. Men i denna deluppgift vill man trunkera datan, det vill säga, man vill ta restriktera vilka mätvärden som ska tillåtas att inkluderas i stickprovet. Man är intresserad av alla värden som är under 50. För att få dessa använder jag indexering och plockar ut alla värden i stickprovet (sample) som har värden under 50. I graferna nedan presenteras det dragna stickprovet innan trunkering och efter trunkering. Här kan man notera att trunkeringen fungerade, alla värden över 50 plockades bort. Men det viktiga att komma ihåg nu är att formler för beräkning av väntevärde och varians i denna subpopulation förändras.

```
### Deluppgift 4a ------------------------------------------------------------#

# TRUNKERING:
# Nu trunkerar jag datamaterialet, dvs, jag tar bort alla observationer som inte
# är under 50.

sample_50 <- sample[sample < 50]


par(mfrow=c(1,2))

hist(sample, main ="Innan trunkering", col ="cornflowerblue"
,xlab ="Observerat värde", ylab ="frekvens"
     )

hist(
  sample_50, main ="Efter trunkering", col="cornflowerblue",
  xlab ="Observerat värde", ylab ="frekvens"
)
```



5.4 Deluppgift 4b)

Beräkna det teoretiska väntevärdet och variansen för den del av populationen som har ett mätvärde högre än 53. Vad illustrerar väntevärdet och variansen i subpopulationen "mätvärden högre än 53" jämfört med väntevärdet och variansen för mätvärdena i hela populationen? Motivera utförligt! (2p)

Det teoretiska medelvärdet presenteras i min teori del, men för denna uppgift blir det:

$$\mathbb{E}[X|X > 53] = \mu + \sigma\lambda(\alpha) = 44 + 8.21 \cdot \frac{\phi(\frac{53-44}{8.21})}{1 - \Phi(\frac{53-44}{8.21})} = 44 + 8.21 \cdot \frac{0.2188}{1 - 0.8635} = 57.16$$

Till sist beräknas variansen:

$$\text{Var}[X|X > 53] = \sigma^2 \left(1 - \left(\frac{\phi(\frac{a-\mu}{\sigma})}{1 - \Phi(\frac{a-\mu}{\sigma})} \cdot \left(\frac{\phi(\frac{a-\mu}{\sigma})}{1 - \Phi(\frac{a-\mu}{\sigma})} - \frac{a-\mu}{\sigma}\right)\right)\right) =$$

$$8.21^2[1 - \left(\frac{\phi(\frac{53-44}{8.21})}{-1 - \Phi(\frac{53-44}{8.21})}\right)\left(\frac{\phi(\frac{53-44}{8.21})}{1 - \Phi(\frac{53-44}{8.21})} - \frac{53-44}{8.21}\right)]$$

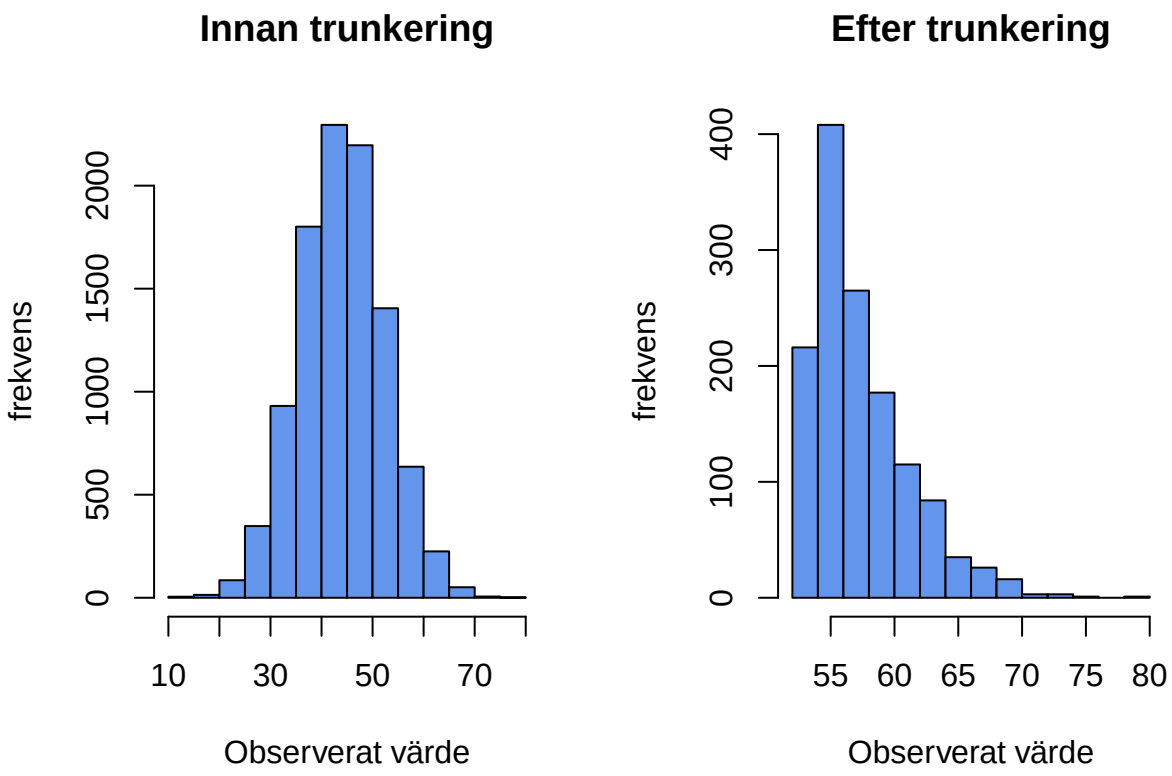
$$= 8.21^2 \left(1 - \left(\frac{\phi(\frac{53-44}{8.21})}{1-\Phi(\frac{53-44}{8.21})} \cdot \left(\frac{\phi(\frac{53-44}{8.21})}{1-\Phi(\frac{53-44}{8.21})} - \frac{53-44}{8.21}\right)\right)\right) = 12.67$$

Reslutatet blev $\mathbb{E}[X|X > 53] = 57$ och $Var[X|X > 53] = 12.56$. Detta väntevärde beskriver det genomsnittliga värden för en population där alla mätvärden är över 53. Alltså, när data trunkeras ökar det förväntade värdet till 57. Men tvärtom för variansen, den minskar till 12.67. Kort sagt beskriver väntevärdet (57) den subpopulation vars värden är högre än 53.

Det tidigare medelvärdet för populationen var 44 med en varians på $8.21^2 = 67.4$. Anledningen till varför medelväret ökat är på grund av att observationer med mindre värden exkluderats. Variansen när datan trunkeras, minskar, vilket visar att spridningen minskar när datamaterialet trunkeras. Detta kan även ses, jag presenterar två grafer nedan över trunkeringen ($X > 53$).

```
### Deluppgift 4b ------------------------------------------------------------#
# Nu gör jag beräkningar i R för väntevärdet:

# Beräknar lilla phi:
phi <- dnorm(((53-44)/8.21))
# Beräknar stora Phi:
Phi <- pnorm(((53-44)/8.21))

# Värden som behövs yttligare
mu_x <- 44
sigma <- 8.21
väntevärde <- (
  mu_x + sigma * (phi/ (1-Phi))
  )

# Nu kan jag beräkna variansen ----------------------------------------------#

alpha <- (53-44)/8.21

varians <- c(
  sigma*sigma*(1 - ((phi/ (1-Phi))*((phi/ (1-Phi))- alpha)))
)

# Gör en df för kable men tror jag skriver det i en formel i RMarkdown ist.
df_E_var <- data.frame(
  Expected = väntevärde,
  Variance = varians
)

# Plottar trunkeringen. X >53 för att visa hur medel och varians ändras
par(mfrow=c(1,2))

hist(sample, main ="Innan trunkering", col ="cornflowerblue"
,xlab ="Observerat värde", ylab ="frekvens"
     )

hist(
```

```
  sample[sample > 53], main ="Efter trunkering", col="cornflowerblue",
  xlab ="Observerat värde", ylab ="frekvens"
)
```



6 References

DeGroot, M. H., & Schervish, M. J. (2013). Probability and statistics (4th ed., Pearson New International Edition). Pearson Education Limited.

Hietala, I. (2026). Kategoriska data (732G34). Linköpings universitet.

UCLA Institute for Digital Research and Education. (n.d.). How are the likelihood ratio, Wald, and Lagrange multiplier (score) tests different and/or similar? https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faqhow-are-the-likelihood-ratio-wald-and-lagrange-multiplier-score-tests-different-andor-similar/