STAT 961: Homework 4

Jiahang Sha

Due Friday, November 19 at 11:59pm

1 Instructions

Setup. Pull the latest version of this assignment from Github and set your working directory to `stat-961-fall-2021/homework/homework-4`. Consult the getting started guide if you need to brush up on `R`, `LaTeX`, or `Git`.

Collaboration. The collaboration policy is as stated on the Syllabus:

> "Students are permitted to work together on homework assignments, but solutions must be written up and submitted individually. Students must disclose any sources of assistance they received; furthermore, they are prohibited from verbatim copying from any source and from consulting solutions to problems that may be available online and/or from past iterations of the course."

In accordance with this policy,

Please list anyone you discussed this homework with: Professor Katsevich, Wenshuo Liu, Sam Rosenberg

Please list what external references you consulted (e.g. articles, books, or websites): uniroot, ggplot documentation

Writeup. Use this document as a starting point for your writeup, adding your solutions between `\begin{sol}` and `\end{sol}`. See the preparing reports guide for guidance on compilation, creation of figures and tables, and presentation quality. Show all the code you wrote to produce your numerical results, and include complete derivations typeset in LaTeX for the mathematical questions.

Programming. The `tidyverse` paradigm for data manipulation (`dplyr`) and plotting (`ggplot2`) are strongly encouraged, but points will not be deducted for using base `R`.

```
library(tidyverse)
```

Grading. Each sub-part of each problem will be worth 3 points: 0 points for no solution or completely wrong solution; 1 point for some progress; 2 points for a mostly correct solution; 3 points for a complete and correct solution modulo small flaws. The presentation quality of the solution for each problem (as exemplified by the guidelines in Section 3 of the preparing reports guide) will be evaluated out of an additional 3 points.

Submission. Compile your writeup to PDF and submit to Gradescope.

Problem 1. Inverting the Wald, likelihood ratio, and score tests for a Poisson GLM.

You have two email accounts: your personal one and your academic one. Last month, you received y_1 and y_2 emails in your personal and academic inboxes, respectively. Interested in the extent to which you receive more (or less) email in your academic inbox, you set up the following Poisson regression model:

$$y_i \overset{\text{ind}}{\sim} \text{Poi}(\mu_i); \quad \log \mu_i = \beta_0 + \beta_1 x_i; \quad i \in \{1, 2\},$$

where $x_i \in \{0, 1\}$ is an indicator for your academic inbox. Your goal is to build a level-α confidence interval for e^{β_1} (the factor by which the expected number of emails in your academic inbox exceeds that in your personal inbox), and to this end you will invert the Wald, likelihood ratio, and score tests.

(a) What is the unrestricted maximum likelihood estimate $(\widehat{\beta}_0, \widehat{\beta}_1)$? What are the corresponding fitted means $(\widehat{\mu}_1, \widehat{\mu}_2)$? What is the maximum likelihood estimate for β_0 if β_1 is fixed at some value $\beta_1^0 \in \mathbb{R}$? What are the corresponding fitted means? What do the fitted means reduce to when $\beta_1^0 = 0$, and why does this make sense?

(b) What is the large-sample normal approximation to the sampling distribution of $\widehat{\boldsymbol{\beta}}$? What is the resulting level-α Wald confidence interval for e^{β_1} (defined by transforming the endpoints of the Wald confidence interval for β_1)? Express your answer explicitly.

(c) Given some $\beta_1^0 \in \mathbb{R}$, what is the likelihood ratio test statistic for $H_0 : \beta_1 = \beta_1^0$? What is the level-$\alpha$ confidence interval for e^{β_1} that results from inverting this test? The endpoints of your interval may be specified as solutions to a nonlinear equation.

(d) Formulate the test $H_0 : \beta_1 = \beta_1^0$ as a goodness of fit test. What is the corresponding score test statistic? What is the level-α confidence interval for e^{β_1} that results from inverting this test? Express your answer explicitly.

Solution 1.

a. $\log \mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^n (\theta_i y_i - \psi(\theta_i)) + \sum_{i=1}^n \log h(y_i) = \beta_0 y_1 - e^{\beta_0} + (\beta_0 + \beta_1) y_2 - e^{\beta_0+\beta_1} - \log(y_1!) - \log(y_2!)$. Taking gradients, we get $\nabla \log \mathcal{L}(\boldsymbol{\beta}) = \begin{bmatrix} y_1 - e^{\beta_0} + y_2 - e^{\beta_0+\beta_1} \\ y_2 - e^{\beta_0+\beta_1} \end{bmatrix}$. Setting this to be 0, we get $\hat{\beta}_0 = \log(y_1), \hat{\beta}_1 = \log(\frac{y_2}{y_1})$. The corresponding fitted values means $(\hat{\mu}_1, \hat{\mu}_2) = e^{X\beta} = (y_1, y_2)$.

Suppose β_1^0 is fixed. Then, we compute $\nabla_{\beta_0} \log \mathcal{L}(\boldsymbol{\beta}) = y_1 - e^{\beta_0} + y_2 - e^{\beta_0+\beta_1^0}$. Setting this to be 0, we get $\hat{\beta}_0 = \log(\frac{y_1+y_2}{1+e^{\beta_1^0}})$. In this case, the corresponding fitted means $(\hat{\mu}_1, \hat{\mu}_2) = (\frac{y_1+y_2}{1+e^{\beta_1^0}}, e^{\beta_1^0} \frac{y_1+y_2}{1+e^{\beta_1^0}})$.

When β_1^0 is fixed to be 0, we compute $(\hat{\mu}_1, \hat{\mu}_2) = (\frac{y_1+y_2}{1+e^0}, e^0 \frac{y_1+y_2}{1+e^0}) = (\frac{y_1+y_2}{2}, \frac{y_1+y_2}{2})$. This assumption means that the indicator is assumed to not work, and therefore no difference in the number of incoming emails between two mailboxes is assumed. Two email boxes have the same probability to receive an email, so the fitted means should of course be the average of the total number of emails I received.

(b) $I(\beta) = -E[\nabla^2 \log \mathcal{L}(\boldsymbol{\beta})] = \begin{bmatrix} e^{\beta_0} + e^{\beta_0+\beta_1} & e^{\beta_0+\beta_1} \\ e^{\beta_0+\beta_1} & e^{\beta_0+\beta_1} \end{bmatrix}$. Plugging in $\hat{\beta}$, we derive $I(\hat{\beta}) = \begin{bmatrix} y_1 + y_2 & y_2 \\ y_2 & y_2 \end{bmatrix}$. Therefore, $I(\hat{\beta})^{-1} = \begin{bmatrix} \frac{1}{y_1} & -\frac{1}{y_1} \\ -\frac{1}{y_1} & \frac{y_1+y_2}{y_1 y_2} \end{bmatrix}$. We use (39) and (40) $\hat{\beta} \sim N(\beta, \begin{bmatrix} \frac{1}{y_1} & -\frac{1}{y_1} \\ -\frac{1}{y_1} & \frac{y_1+y_2}{y_1 y_2} \end{bmatrix})$, where

$\hat{\beta}_1 \sim N(\beta_1, \frac{y_1+y_2}{y_1 y_2})$. We use (41) to construct

$$\text{CI}(\hat{\beta}_1) = \left[\log(\frac{y_2}{y_1}) - z_{1-\alpha/2}\sqrt{\frac{y_1+y_2}{y_1 y_2}}, \log(\frac{y_2}{y_1}) + z_{1-\alpha/2}\sqrt{\frac{y_1+y_2}{y_1 y_2}}\right]$$

Applying e^x to the obtained confidence interval, we find

$$\text{CI}(e^{\hat{\beta}_1}) = \left[\frac{y_2}{y_1} e^{-z_{1-\alpha/2}\sqrt{\frac{y_1+y_2}{y_1 y_2}}}, \frac{y_2}{y_1} e^{z_{1-\alpha/2}\sqrt{\frac{y_1+y_2}{y_1 y_2}}}\right]$$

all at a confidence level α.

(c) By (37) we know the deviance for Poisson GLMs with log-link $D(y;\hat{\mu}) = 2(y_1 \log(y_1/\hat{\mu}_1) + y_2 \log(y_2/\hat{\mu}_2)) = 0$
and $D(y;\hat{\mu}_{-S}) = 2(y_1 \log(y_1/\hat{\mu}_1^0) + y_2 \log(y_2/\hat{\mu}_2^0)) = 2\left(y_1 \log\left(\frac{(e^{\beta_1^0}+1)y_1}{y_1+y_2}\right) + y_2 \log\left(\frac{(e^{\beta_1^0}+1)y_2}{e^{\beta_1^0}(y_1+y_2)}\right)\right)$.
In this model, $|S| = 1$. Therefore, we use LRT test statistics in terms of difference in deviances

$$T^{LRT} = D(y;\hat{\mu}_{-S}) - D(y;\hat{\mu}) = 2\left(y_1 \log\left(\frac{(e^{\beta_1^0}+1)y_1}{y_1+y_2}\right) + y_2 \log\left(\frac{(e^{\beta_1^0}+1)y_2}{e^{\beta_1^0}(y_1+y_2)}\right)\right) \sim \chi_1^2$$

.
Therefore, $\text{CI}(e^{\beta_1}) = \{x : 2\left(y_1 \log\left(\frac{(x+1)y_1}{y_1+y_2}\right) + y_2 \log\left(\frac{(x+1)y_2}{x(y_1+y_2)}\right)\right) \leq \chi^2_{1,1-\alpha}\}$
$= \{x : (y_1+y_2)\log(1+x) - y_2 \log(x) \leq \frac{\chi^2_{1,1-\alpha}}{2} - y_1 \log(\frac{y_1}{y_1+y_2}) - y_2 \log(\frac{y_2}{y_1+y_2})\}$.

(d) Under H_0, we have $(\hat{\mu}_1, \hat{\mu}_2) = (\frac{y_1+y_2}{1+e^{\beta_1^0}}, e^{\beta_1^0}\frac{y_1+y_2}{1+e^{\beta_1^0}})$. Therefore, the score test statistic is

$$\frac{(y_1-\hat{\mu}_1)^2}{\hat{\mu}_1} + \frac{(y_2-\hat{\mu}_2)^2}{\hat{\mu}_2} = \frac{(y_1 - \frac{y_1+y_2}{1+e^{\beta_1^0}})^2}{\frac{y_1+y_2}{1+e^{\beta_1^0}}} + \frac{(y_2 - e^{\beta_1^0}\frac{y_1+y_2}{1+e^{\beta_1^0}})^2}{e^{\beta_1^0}\frac{y_1+y_2}{1+e^{\beta_1^0}}} = \frac{(y_2 - y_1 e^{\beta_1^0})^2}{(y_1+y_2)e^{\beta_1^0}} \sim \chi_1^2$$

Inverting this test, we solve the quadratic inequality and obtain the confidence interval as

$$\text{CI}(e^{\beta_1}) = \left[\frac{\chi^2_{1,(1-\alpha)}y_1 + \chi^2_{1,(1-\alpha)}y_2 + 2y_1y_2 - \sqrt{c_1}}{2y_1^2}, \frac{\chi^2_{1,(1-\alpha)}y_1 + \chi^2_{1,(1-\alpha)}y_2 + 2y_1y_2 + \sqrt{c_1}}{2y_1^2}\right]$$

where $c_1 = {\chi^2_{1,(1-\alpha)}}^2 y_1^2 + 2{\chi^2_{1,(1-\alpha)}}^2 y_1y_2 + {\chi^2_{1,(1-\alpha)}}^2 y_2^2 + 4\chi^2_{1,(1-\alpha)}y_1^2y_2 + 4\chi^2_{1,(1-\alpha)}y_1y_2^2$.

Problem 2. Comparing the three confidence interval constructions from Problem 1.

Let's use a numerical simulation to compare the three confidence interval constructions from Problem 1 in finite samples.

(a) Write functions called `get_ci_wald`, `get_ci_lrt`, and `get_ci_score` that take as arguments (`y_1`, `y_2`, `alpha`) and return the corresponding confidence intervals for e^{β_1}. If the confidence interval is undefined for a given pair (y_1, y_2), your function should return $(-\infty, \infty)$.

(b) To get a first sense of how the three intervals compare, compute level $\alpha = 0.05$ intervals for $(y_1, y_2) = (10^1, 10^1), (10^{1.5}, 10^{1.5}), \ldots, (10^5, 10^5)$. Plot the lower and upper endpoints of these intervals as functions of y_1 (you should arrive at a plot containing six curves, corresponding to the lower and upper endpoints of the three methods). Add a dashed horizontal line at the MLE for e^{β_1} (which is the same for each given pair (y_1, y_2)). How do the interval widths compare, both across methods and across (y_1, y_2) values?

(c) Next, calculate the average length and coverage of the three level $\alpha = 0.05$ confidence intervals for e^{β_1} in the following simulation setting. Set $(\mu_1, \mu_2) = (10^1, 10^1), (10^{1.5}, 10^{1.5}), \ldots, (10^5, 10^5)$. For each pair (μ_1, μ_2), generate 5000 realizations of (y_1, y_2) and compute the three confidence intervals for each realization. Plot the average length and coverage for each of the three interval constructions as a function of μ_1 (please omit the undefined/infinite-length intervals from the calculations of length and coverage). Compare and contrast the average lengths and coverages of the three constructions, both across methods and across (μ_1, μ_2) values.

(d) Last month you received 60 emails in your personal inbox and 90 in your academic inbox. Pick one of the three confidence interval constructions above that you feel has good coverage and small width. According to this construction, what is the confidence interval for e^{β_1}? Can you reject the null hypothesis that the two inboxes receive emails at the same rate?

Solution 2.

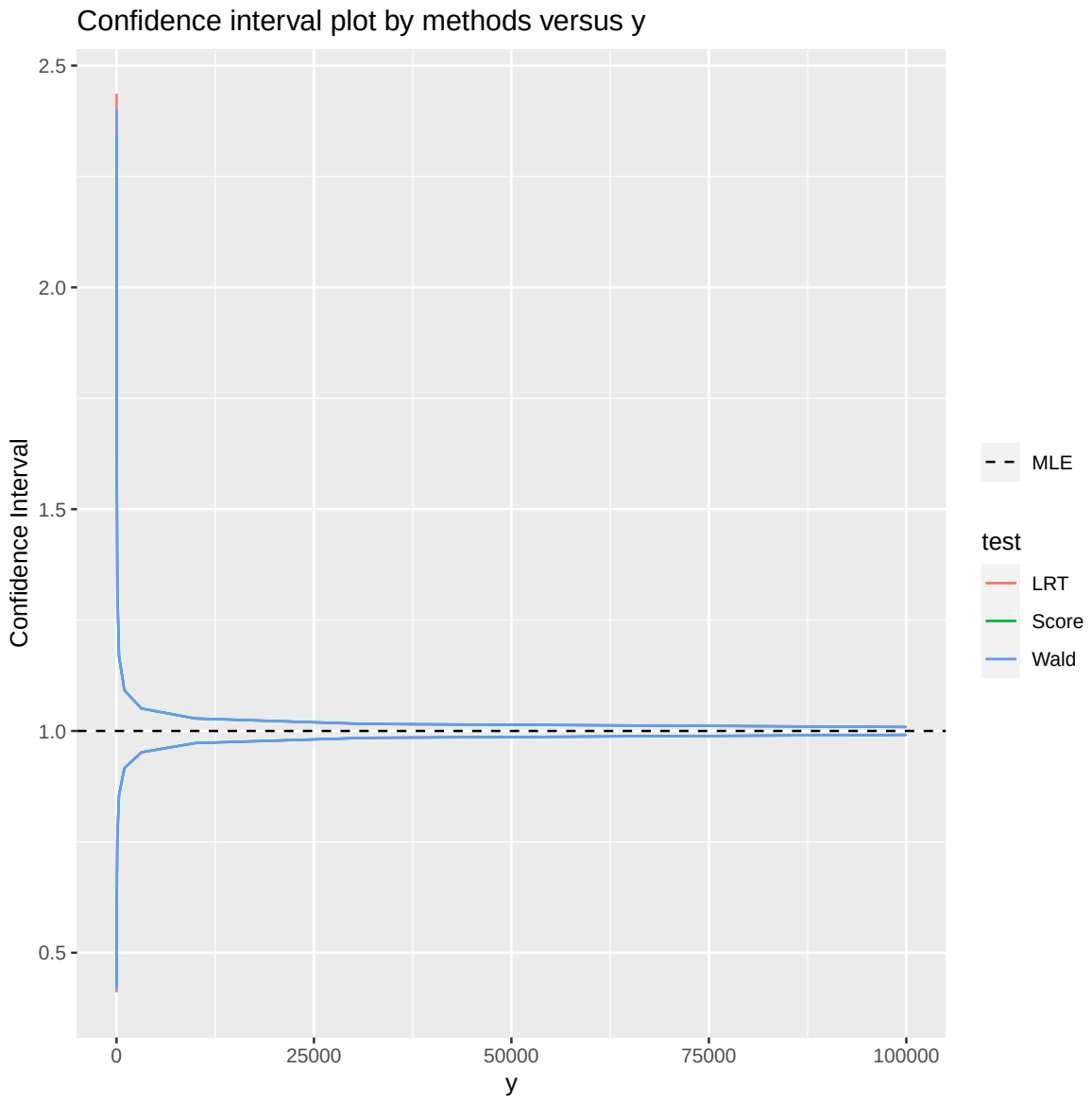
(a)
```r
#function to return return CI by wald test
get_ci_wald <- function(y_1, y_2, alpha){
  df = data.frame('lower' = -Inf, 'upper' = Inf)
  if (y_2 < 0 || y_1 < 0 || alpha > 1 || alpha < 0){
    return(df)
  }
  coef = y_2/y_1
  lowerexp = exp( -qnorm(1-alpha/2) * sqrt(1/y_2 + 1/y_1) )
  upperexp = exp( qnorm(1-alpha/2) * sqrt(1/y_2 + 1/y_1) )
  df$lower = coef*lowerexp
  df$upper = coef*upperexp
  return(df)
}


require(rootSolve)
#function to return return CI by lrt test
get_ci_lrt <- function(y_1, y_2, alpha){
  c = qchisq(1-alpha, df = 1)
```

```
  df = data.frame('lower' = -Inf, 'upper' = Inf)
  if (y_2 < 0 || y_1 < 0 || alpha > 1 || alpha < 0){
    return(df)
  }
  abc <- function(x){
    -c/2 + y_1 * log((1+x) *y_1/ (y_1+y_2)) +y_2 * log((x+1)/x*y_2/(y_1+y_2))
  }
  df$lower = uniroot(abc, lower = 0, upper = y_2/y_1, tol=1e-12)$root
  df$upper = uniroot(abc, lower = y_2/y_1, upper = exp(10), tol=1e-12)$root
  return(df)
}

#function to return return CI by score test
get_ci_score <- function(y_1, y_2, alpha){
  df = data.frame('lower' = -Inf, 'upper' = Inf)
  if (y_2 < 0 || y_1 < 0 || alpha > 1 || alpha < 0){
    return(df)
  }
  c = qchisq(1-alpha, df = 1)
  discri = (c/y_1)^2 + 2*c/y_1*c*y_2/y_1/y_1 + 4/y_1*c/y_1*y_2 +
    (c/y_1*y_2/y_1)^2 + 4/y_1*c/y_1*y_2/y_1*y_2
  lowerbd = c/2/y_1 + c*y_2/2/y_1/y_1 + y_2/y_1 -sqrt(discri)/2
  upperbd = c/2/y_1 + c*y_2/2/y_1/y_1 + y_2/y_1 +sqrt(discri)/2
  df$lower = lowerbd
  df$upper = upperbd
  return(df)
}
```

(b)

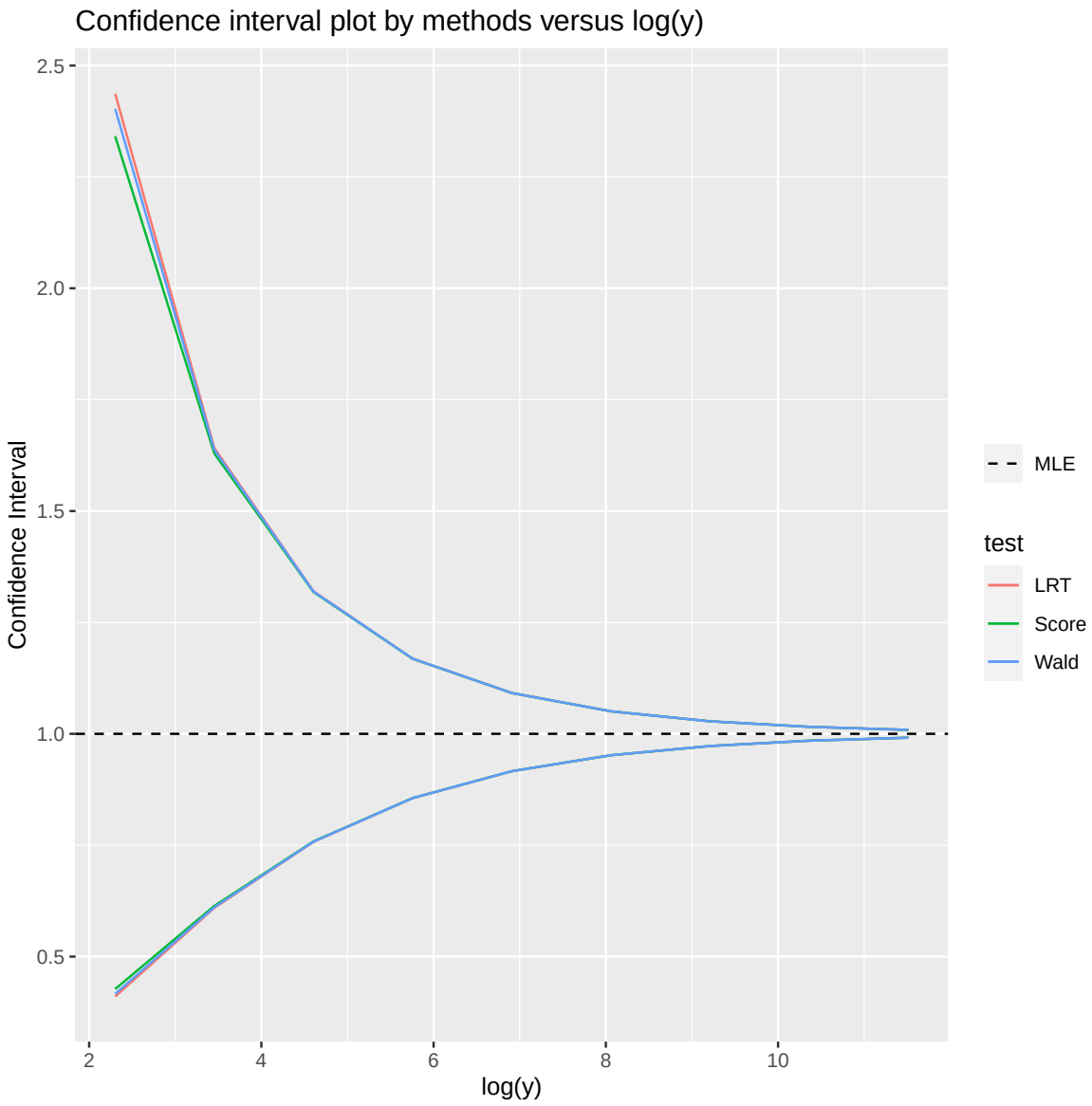
```
alpha = 0.05
yval = round(c(10^seq(1,5,0.5)))
bdf = data.frame()
for (i in 1:length(yval)){
  y = yval[i]
  #wald
  dftemp = get_ci_wald(y,y,alpha)
  dftemp$y = y
  dftemp$test = "Wald"
  bdf = rbind(bdf, dftemp)
  #score
  dftemp = get_ci_score(y,y,alpha)
  dftemp$y = y
  dftemp$test = "Score"
  bdf = rbind(bdf, dftemp)
  #lrt
  dftemp = get_ci_lrt(y,y,alpha)
  dftemp$y = y
  dftemp$test = "LRT"
```

```
  bdf = rbind(bdf, dftemp)
}
#plot the confidence interval
bdf %>% ggplot(aes(color=test)) +
  geom_line(aes(x=y, y=lower))+
  geom_line(aes(x=y, y=upper))+
  geom_hline(aes(yintercept=1, linetype="MLE"))+
  scale_linetype_manual(name = "", values=2)+
  ggtitle("Confidence interval plot by methods versus y")+
  labs(y="Confidence Interval")
```



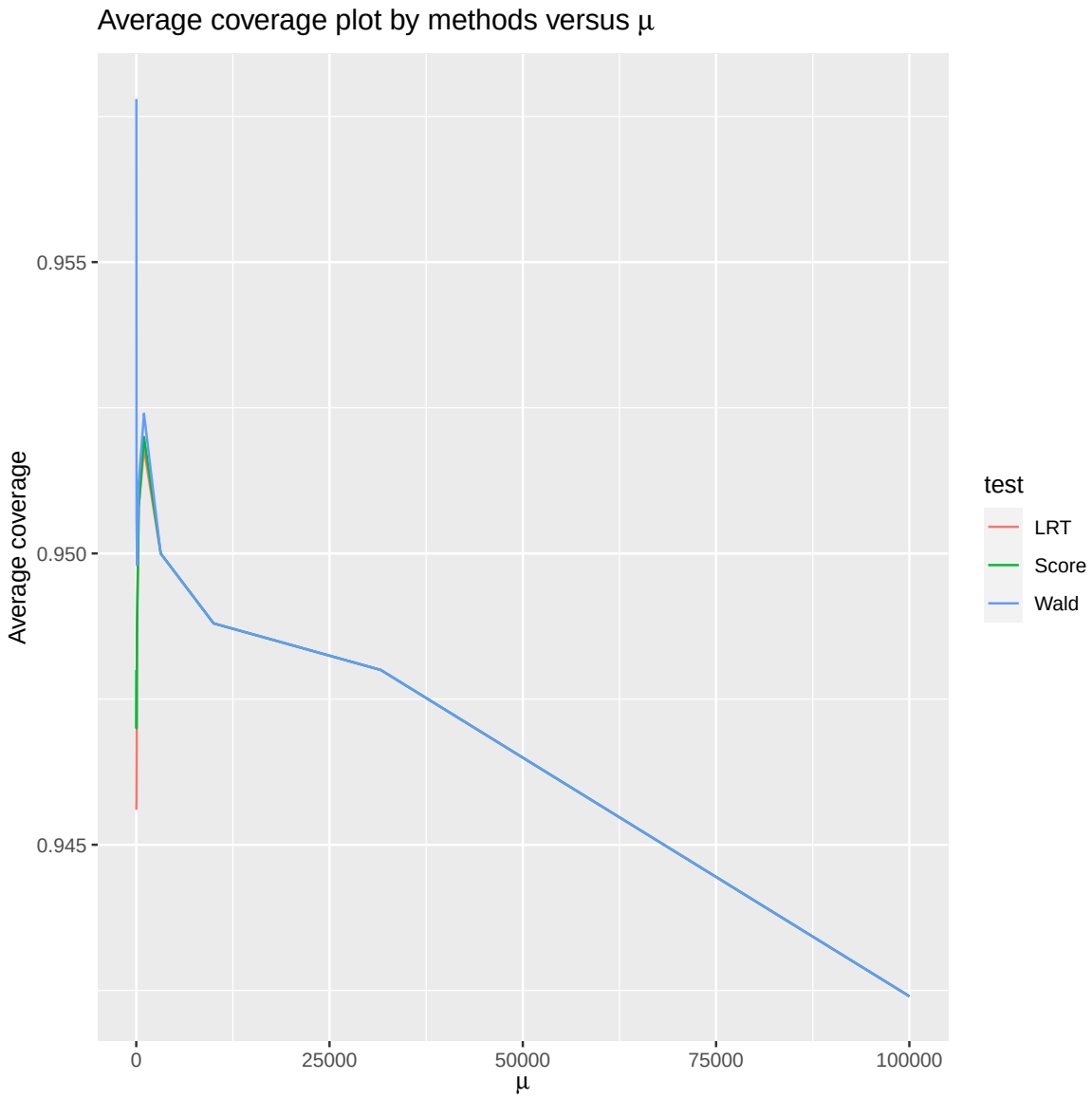

```
#try to plot on log(y) scale
bdf$logy = log(bdf$y)
```

```
bdf %>% ggplot(aes(color=test)) +
  geom_line(aes(x=logy, y=lower))+
  geom_line(aes(x=logy, y=upper))+
  geom_hline(aes(yintercept=1, linetype="MLE"))+
  scale_linetype_manual(name = "", values=2)+
  ggtitle("Confidence interval plot by methods versus log(y)")+
  labs(x="log(y)", y="Confidence Interval")
```

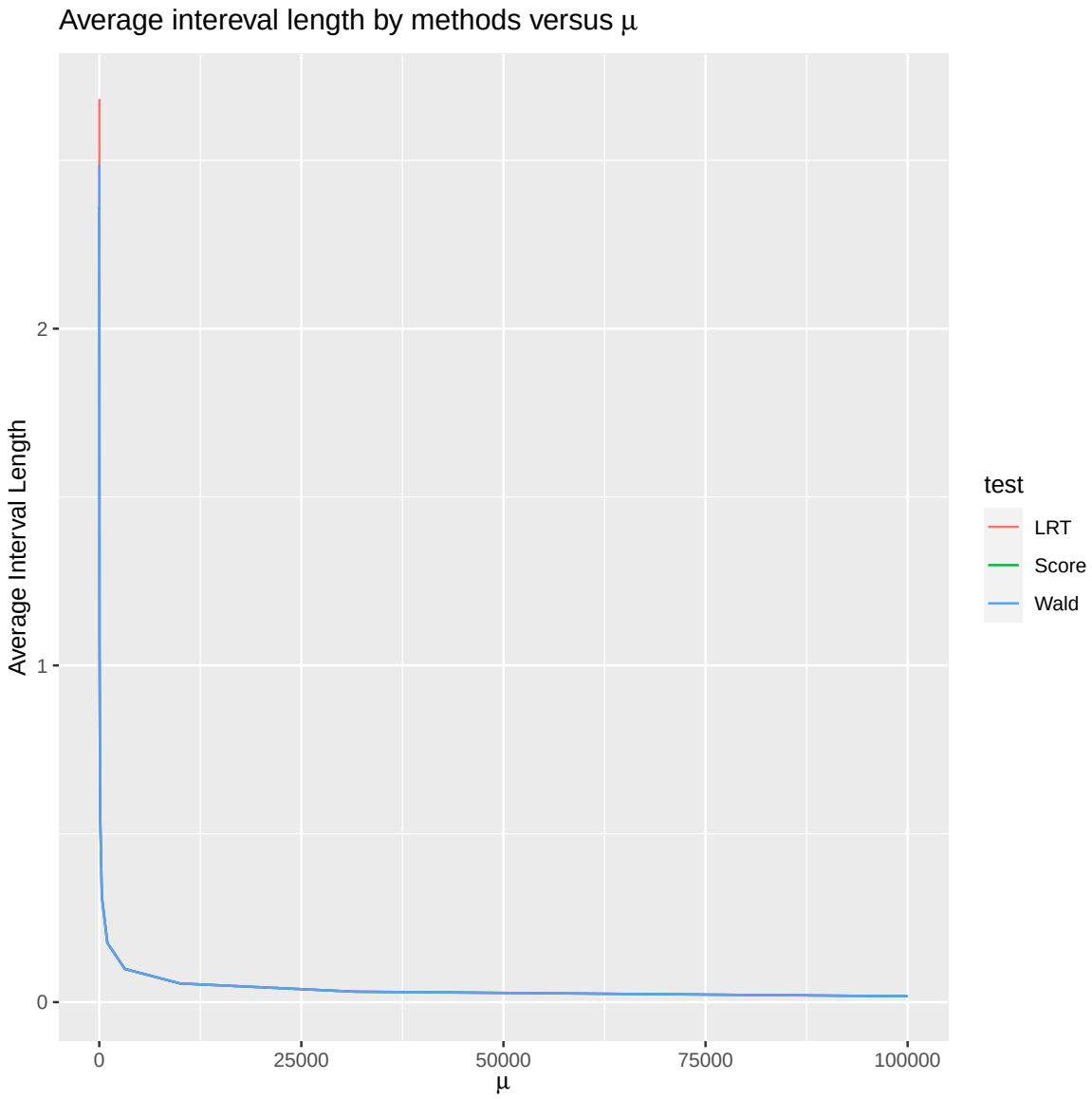


Due to the nature of exponential function, the lower endpoints of three intervals differ only extremely slightly. The upper endpoints of three intervals also differ quite slightly, with LRT giving the widest confidence interval and score test inverted CI giving the most narrowest. As y increases, three confidence intervals all become narrow converging to 1, which is the MLE for e^{β_1}.

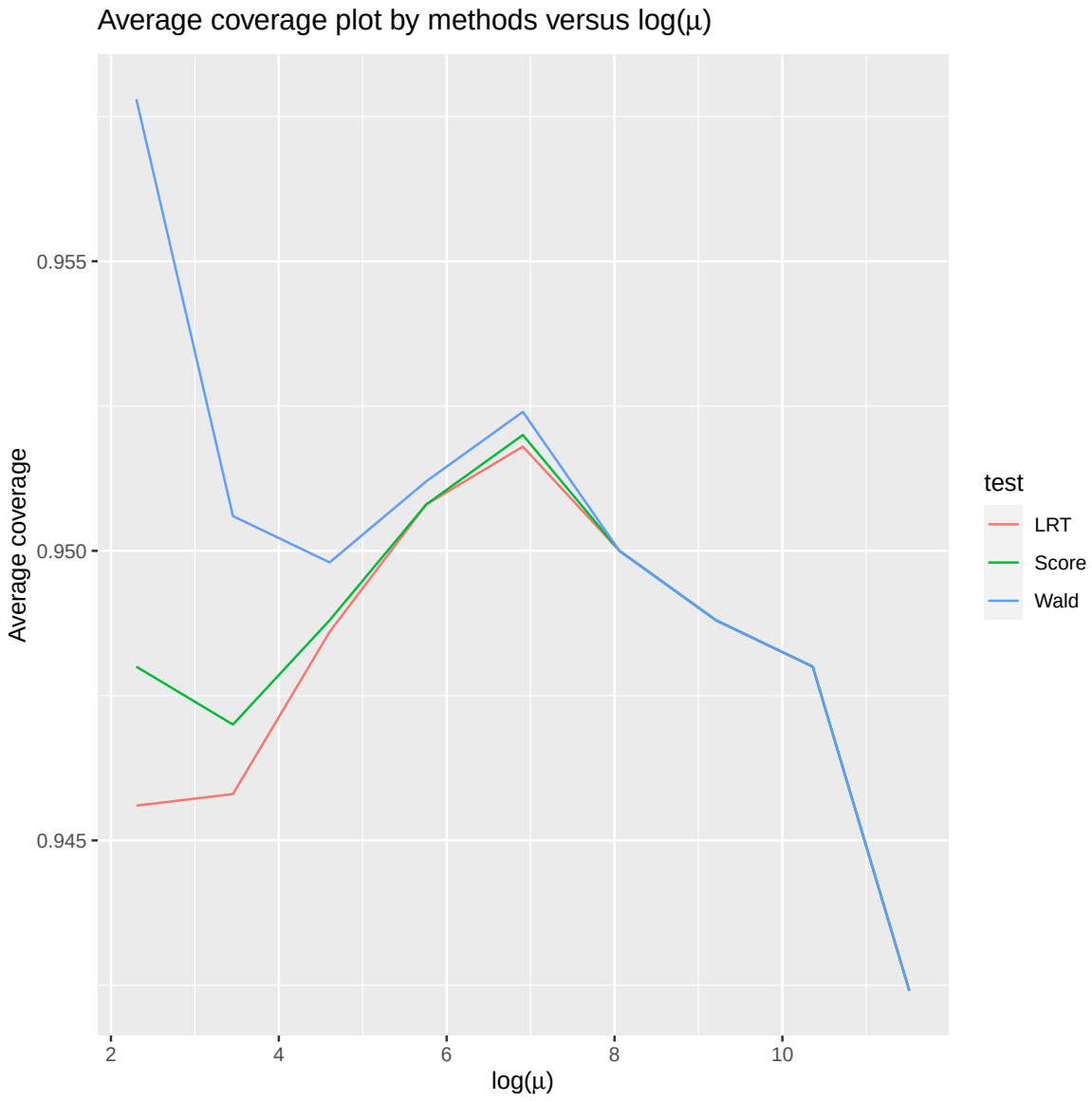
(c)
```r
alpha = 0.05
muval = c(10^seq(1,5,0.5))
cmudf = data.frame()
for (i in 1:length(muval)){
  mu = muval[i]
  totalsample = rpois(10000,mu)
  #get random (y1,y2)
  yone = totalsample[1:5000]
  ytwo = totalsample[5001:10000]
  dfloop = data.frame()
  for (j in 1:5000){
    #wald
    dftemp = get_ci_wald(yone[j],ytwo[j],alpha)
    dftemp$test = "Wald"
    dftemp$mu = mu
    dftemp$contained = (dftemp$lower <= 1 && dftemp$upper >=1)
    dftemp$length = dftemp$upper - dftemp$lower
    dfloop = rbind(dfloop, dftemp)
    #score
    dftemp = get_ci_score(yone[j],ytwo[j],alpha)
    dftemp$test = "Score"
    dftemp$mu = mu
    dftemp$contained = (dftemp$lower <= 1 && dftemp$upper >=1)
    dftemp$length = dftemp$upper - dftemp$lower
    dfloop = rbind(dfloop, dftemp)
    #lrt
    dftemp = get_ci_lrt(yone[j],ytwo[j],alpha)
    dftemp$test = "LRT"
    dftemp$mu = mu
    dftemp$contained = (dftemp$lower <= 1 && dftemp$upper >=1)
    dftemp$length = dftemp$upper - dftemp$lower
    dfloop = rbind(dfloop, dftemp)
  }
  #get rid of the inifnity
  dfloop = dfloop[is.finite(dfloop$lower),]
  cmudf = rbind(cmudf,
                data.frame(dfloop %>% group_by(test,mu) %>%
                           summarise_at(vars(lower, upper, length, contained),
                                        list(name=mean))))

}
require(latex2exp)
#plot the average coverage
cmudf %>% ggplot(aes(fill=test, color=test))+
  geom_line(aes(x=mu, y=contained_name))+
  ggtitle(TeX("Average coverage plot by methods versus $\\mu$"))+
  labs(x=TeX("$\\mu$"), y="Average coverage")
```



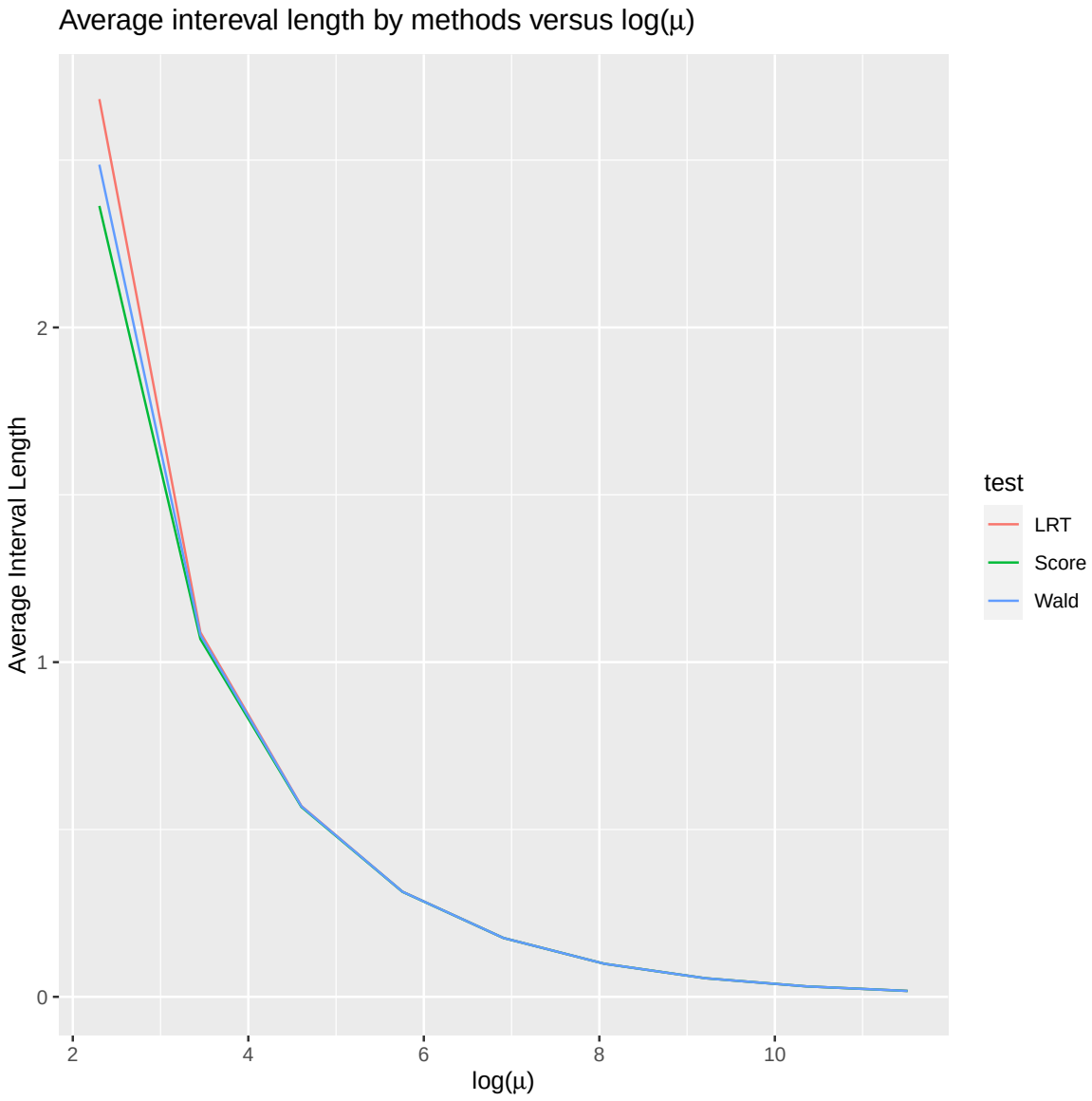

```
#plot the average length
cmudf %>% ggplot(aes(fill=test, color= test)) +
  geom_line(aes(x=mu, y=length_name))+
  ggtitle(TeX("Average intereval length by methods versus $\\mu$"))+
  labs(x=TeX("$\\mu$"), y="Average Interval Length")
```




```
#try to plot the average coverage on log(mu) scale
cmudf$logmu = log(cmudf$mu)
cmudf %>% ggplot(aes(fill=test, color= test)) +
  geom_line(aes(x=logmu, y=contained_name))+
  ggtitle(TeX("Average coverage plot by methods versus $\\log(\\mu)$"))+
  labs(x=TeX("$\\log(\\mu)$"), y="Average coverage")
```




```
#plot the average length on log(mu) scale
cmudf %>% ggplot(aes(fill=test, color= test)) +
  geom_line(aes(x=logmu, y=length_name))+
  ggtitle(TeX("Average intereval length by methods versus $\\log(\\mu)$"))+
  labs(x=TeX("$\\log(\\mu)$"), y="Average Interval Length")
```



LRT has the worst coverage while Walds has the best among all three methods when μ is not very large. However, also notice that the difference in coverage between three methods are not significant (at most less than 2%). It can be said that the difference in the coverage between three methods is negligible. LRT yields the highest average confidence length while score test yields the lowest among all three methods, while this difference, as can be seen in the graph, is also negligible when μ_1, μ_2 is not too small. As expected, when μ_1, μ_2 increases, the coverage of three confidence intervals become increasingly similar and the confidence interval lengths all become narrower.

(d) I think score test is good.

```
#use score test
get_ci_score(60,90,.05)

##      lower    upper
```

```
## 1 1.083547 2.076514
```

The confidence interval for e^{β_1} does not contain 1, which means $\beta_1 = 0$ is not included. We reject the null hypothesis $\beta_1 = 0$, meaning that it is not very possible that two inboxes receive emails at the same rate.

Problem 3. Case study: Child development.

Children were asked to build towers as high as they could out of cubical and cylindrical blocks.[1] The number of blocks used and the time taken were recorded (see `blocks_data` below). In this problem, only consider the number of blocks used and the age of the child.

```
blocks_data = read_tsv("../../data/blocks.tsv")
print(blocks_data, n = 5)

## # A tibble: 100 x 6
##   Child Number  Time Trial Shape   Age
##   <chr>  <dbl> <dbl> <dbl> <chr> <dbl>
## 1 A         11  30       1 Cube   4.67
## 2 B          9  19       1 Cube   5
## 3 C          8  18.6     1 Cube   4.42
## 4 D          9  23       1 Cube   4.33
## 5 E         10  29       1 Cube   4.33
## # ... with 95 more rows
```

(a) Create a scatter plot of blocks used versus age; since there are exact duplicates of `(Number, Age)` in the data, use `geom_count()` instead of `geom_point()`. Propose a GLM to model the number of blocks used as a function of age.

(b) Fit this GLM using R, and write down the fitted model. Determine the standard error for each regression parameter, and find the 95% Wald confidence intervals for the regression coefficients.

(c) Use Wald, score, and likelihood ratio tests to determine if age seems necessary in the model. Compare the results and comment.

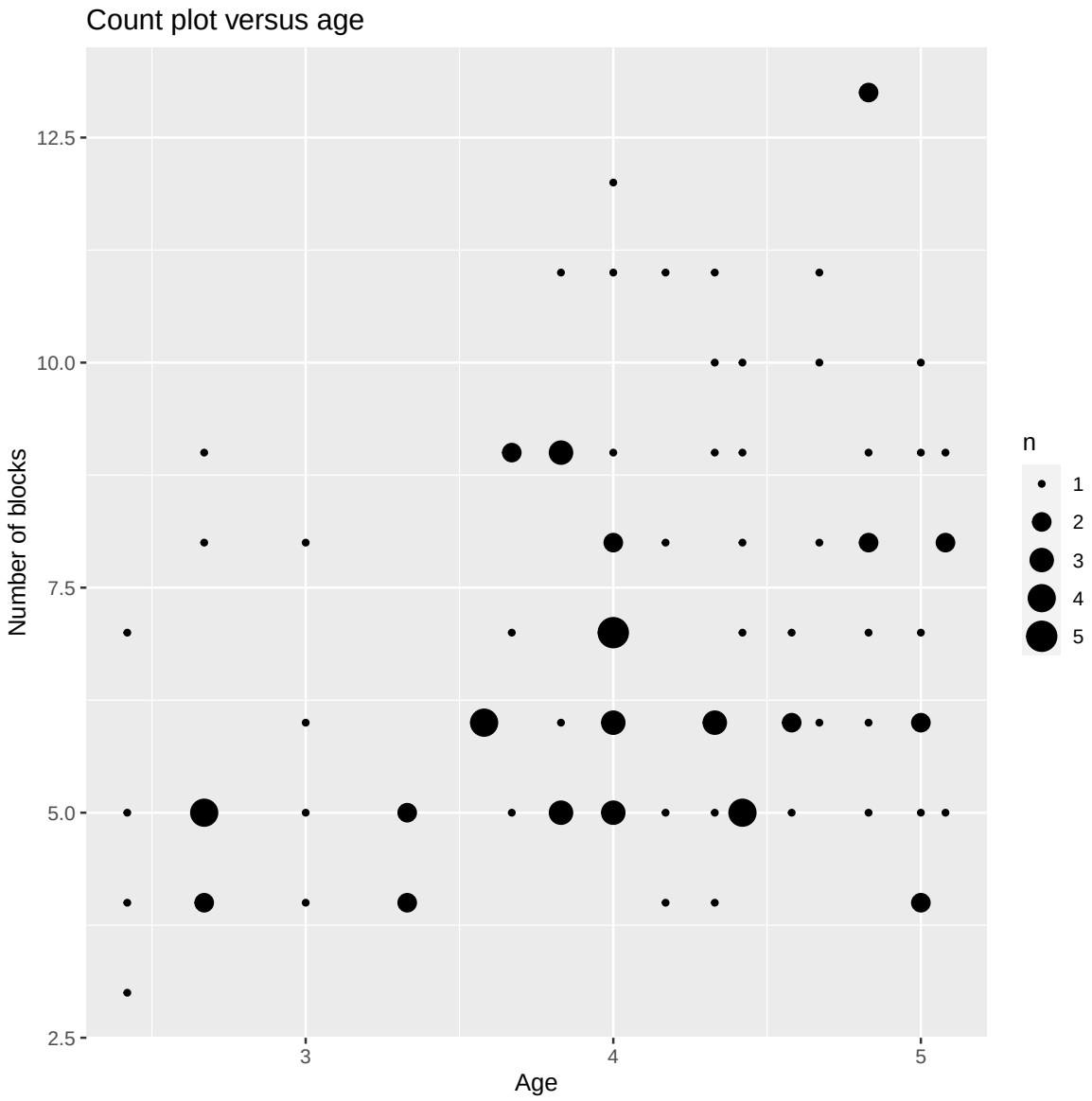
(d) Plot the number of blocks used against age as in part (a), adding the relationship described by the fitted model as well as lines indicating the lower and upper 95% confidence intervals for these fitted values.

Acknowledgment: This problem was drawn from "Generalized Linear Models With Examples in R" (Dunn and Smyth, 2018).

Solution 3.

(a)
```
#plot the data
ggplot(blocks_data,aes(y=Number,x=Age)) +
  geom_count()+
  ggtitle("Count plot versus age")+
  labs(y="Number of blocks")
```

[1] Johnson, B., Courtney, D.M.: Tower building. Child Development 2(2), 161-162 (1931).



(b)

```
#run glm
require(MASS)
glmvar = blocks_data %>%
  glm(formula = Number~Age, family = poisson(link = "log"))
sumglmvar = summary(glmvar)
sumglmvar

##
## Call:
## glm(formula = Number ~ Age, family = poisson(link = "log"), data = .)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.4977  -0.6542  -0.1984   0.4714   1.8155
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.3447     0.2223   6.048 1.47e-09 ***
## Age           0.1415     0.0534   2.650  0.00805 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 69.429  on 99  degrees of freedom
## Residual deviance: 62.244  on 98  degrees of freedom
## AIC: 439.64
##
## Number of Fisher Scoring iterations: 4

#print look at the std. error in coef
coef(sumglmvar)

##              Estimate Std. Error  z value     Pr(>|z|)
## (Intercept) 1.3446992 0.22235364 6.047570 1.470468e-09
## Age         0.1415096 0.05340039 2.649974 8.049803e-03
```

```
#get Wald CI
confint.default(glmvar)

##                  2.5 %    97.5 %
## (Intercept) 0.90889409 1.7805044
## Age         0.03684679 0.2461725
```

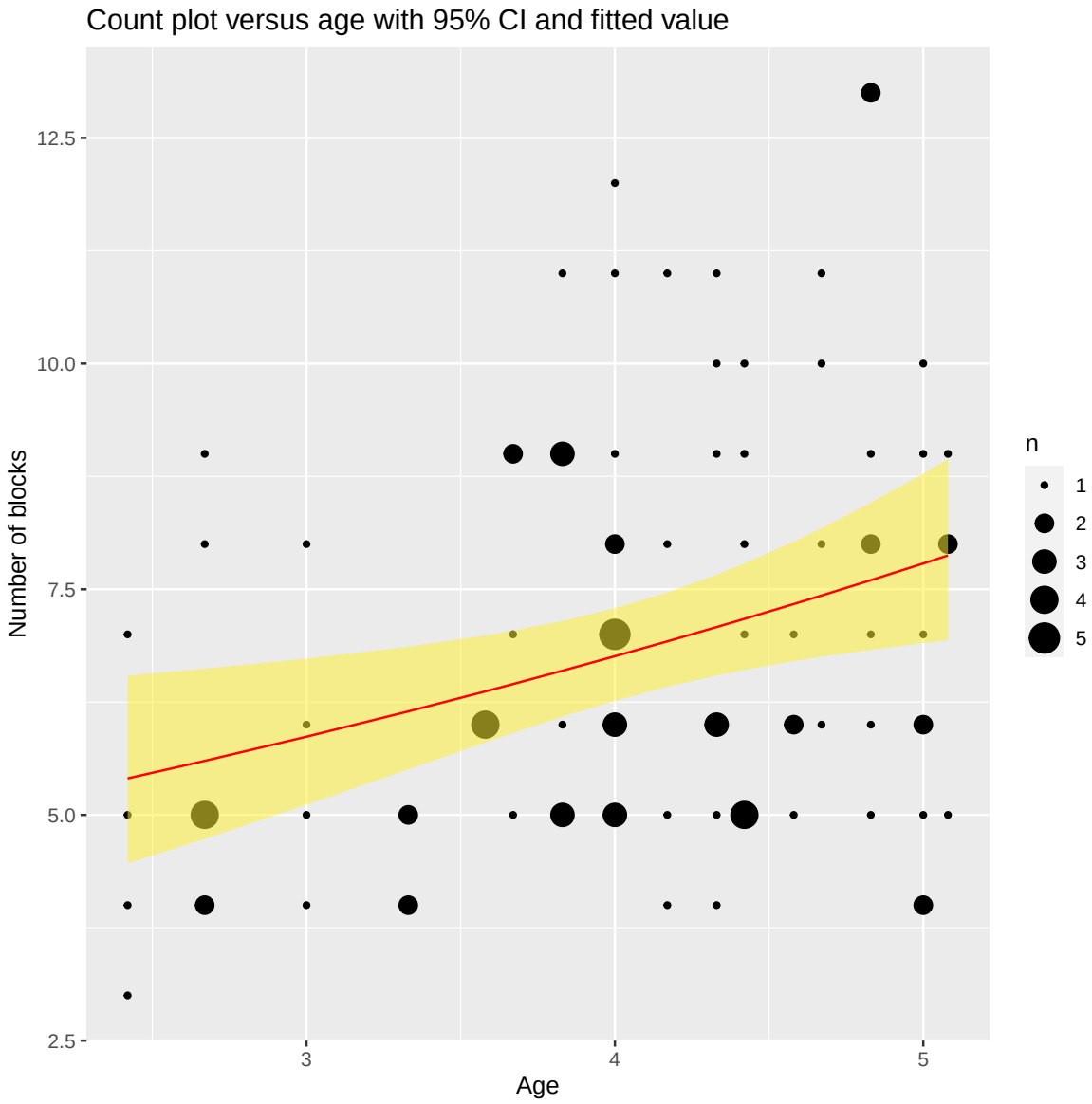
(c)

```
#create a null model
qqglmvar = glm(Number~1, data = blocks_data, family = poisson(link = "log"))
#score test
anova(qqglmvar,glmvar,test="Rao")

## Analysis of Deviance Table
##
## Model 1: Number ~ 1
## Model 2: Number ~ Age
##   Resid. Df Resid. Dev Df Deviance    Rao Pr(>Chi)
## 1        99     69.429
## 2        98     62.244  1   7.1854 7.0346 0.007995 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#likelihood ratio test
anova(qqglmvar,glmvar,test="LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: Number ~ 1
## Model 2: Number ~ Age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        99     69.429
## 2        98     62.244  1   7.1854  0.00735 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#wald test
coef(sumglmvar)

##              Estimate Std. Error  z value     Pr(>|z|)
## (Intercept) 1.3446992 0.22235364 6.047570 1.470468e-09
## Age         0.1415096 0.05340039 2.649974 8.049803e-03
```

We observe that age is necessary with very small p values throughout all three tests. Wald test gives a slightly higher p value, which makes sense because 100 is not a very large number while Wald test performs slightly worse for small samples. On the other hand, the difference in p value is indeed not very large from other two tests because 100 is not too small either. LRT's p-value is smaller than Wald's and score test's even though the difference is insignificant too.

(d)
```
#plot count + fitted value + 95% CI
newdata = data.frame(Age=blocks_data$Age)
preds = predict(glmvar, newdata, se.fit=TRUE,interval='confidence')
blocks_data$lowerCI = exp(preds$fit - 1.96*preds$se.fit)
blocks_data$upperCI = exp(preds$fit + 1.96*preds$se.fit)
blocks_data$fitdata = exp(preds$fit)

blocks_data %>% ggplot(aes(y=Number,x=Age)) +
  geom_count()+
  geom_ribbon(aes(ymin = lowerCI, ymax = upperCI),fill="#FFF03388")+
  geom_line(aes(y=fitdata),color="red")+
  ggtitle("Count plot versus age with 95% CI and fitted value")+
  labs(y="Number of blocks")
```


Count plot versus age with 95% CI and fitted value
Number of blocks
12.5
10.0
7.5
5.0
2.5
3
4
5
Age
n
1
2
3
4
5