Unit1_Paper1_Version2

youngwoo Kwon

2021 2 20

#Overview description
The analysis used Wilson Confidence interval to check the 95% confidence interval of positivity rate for each week.
The test population and positivity rate showed little negative relationship with correlation -0.1843
The test population before two weeks and positivity rate showed more strong negative relationship with correlation -0.4270.

Separted the data into two parts, which test population is less than 5000 and larger than 5000, we cannot reject the hypothesis "Two populations have different positivity rate" since average positivity rate for both samples were in the 95% confidence interval of the total positivity rate. However, we can reject the hypothesis "Two population have different positivity rate" for the data which test population before two weeks were less than 5000 and larger than 5000, since the mean positivity rate for the sample 'population larger than 5000' was out of the 95% confidence interval of the total positivity rate.

Seperated the data into four parts, divided 19 weeks into 5/5/5/4 weeks, all the intervals failed the log likelihood test for 95% confidence level. Also, using chi square test, we can check the p value for each test was less than 2.2e-16.

Merging the intervals into two intervals, 19 weeks into 10/9 weeks, two intervals also failed the log likelihood test for 95% confidence level. Also, using chi square test, we can check the p value fore each test was less than 2.2e-16.

#0.Header

```r
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
mydata = read.csv("http://dept.stat.lsa.umich.edu/~bbh/s485/data/covidTestsFA2020.csv")
mydata
```

```
##          week pos_rate n_tests
## 1  2020-08-30    0.019    1913
## 2  2020-09-06    0.028    2216
## 3  2020-09-13    0.036    3417
## 4  2020-09-20    0.076    3824
## 5  2020-09-27    0.034    4178
## 6  2020-10-04    0.050    4367
## 7  2020-10-11    0.067    6151
## 8  2020-10-18    0.047    6994
## 9  2020-10-25    0.027    6575
## 10 2020-11-01    0.040    6488
## 11 2020-11-08    0.031    7374
## 12 2020-11-15    0.021   12444
## 13 2020-11-22    0.029    3951
## 14 2020-11-29    0.020    6654
## 15 2020-12-06    0.022    6170
## 16 2020-12-13    0.014    7124
## 17 2020-12-20    0.042    1598
## 18 2020-12-27    0.033    2135
## 19 2021-01-03    0.027    4461
```

#1.Wilson Confidence Interval for each week
#First column is the lowerbound for pos_rate and Second column is the upperbound for pos_rate for each week

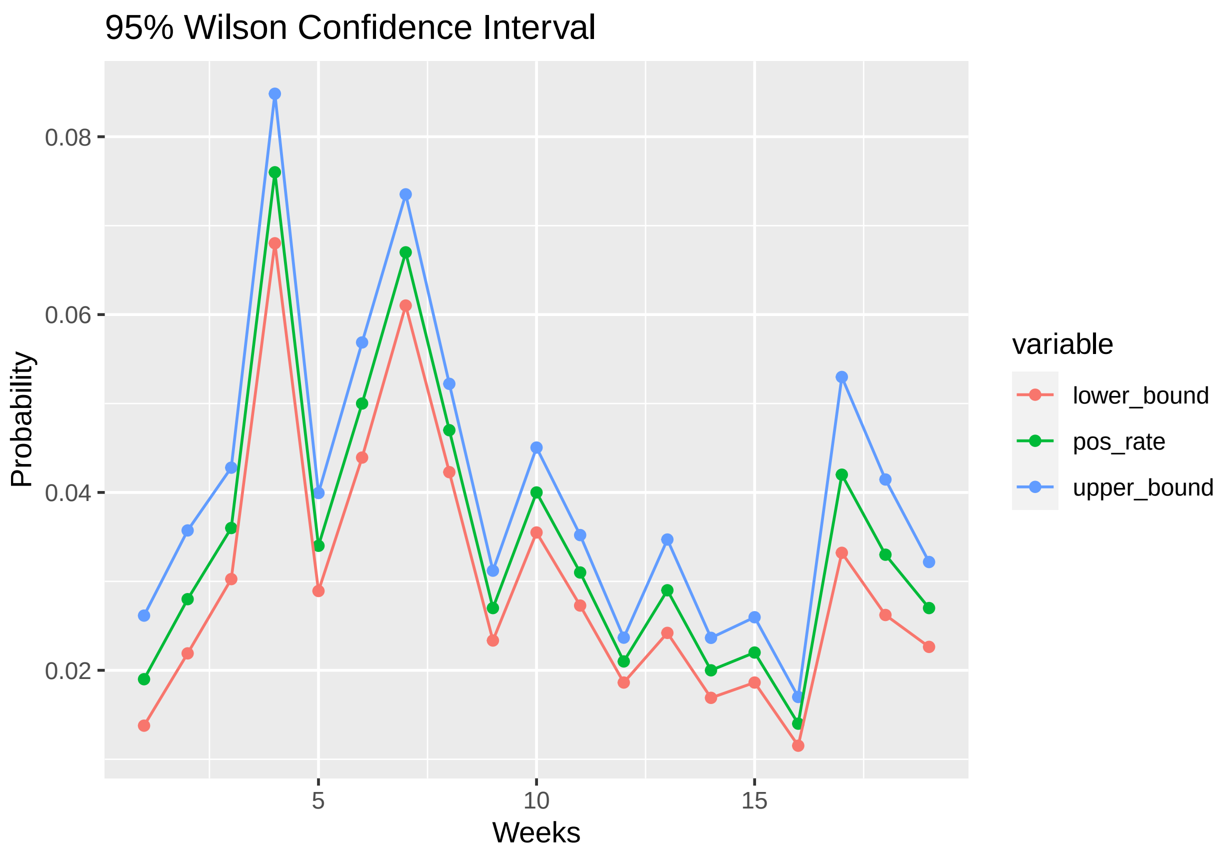
```
Weeks = 1:19
k = 1.96
n = mydata[,3]
phat = mydata[,2]
CI_Wilson = c((n*phat+k^2/2)/(n+k^2) - k*sqrt(n)/(n+k^2)*sqrt(phat*(1-phat)+k^2/(4*n)),(n*phat+k^2/2)/(
CI_Wilson_matrix = matrix(data = CI_Wilson, ncol = 2)
CI_Wilson_matrix
```

```
##             [,1]       [,2]
##  [1,] 0.01377656 0.02615141
##  [2,] 0.02190550 0.03572816
##  [3,] 0.03025656 0.04278558
##  [4,] 0.06801972 0.08483132
##  [5,] 0.02891855 0.03993762
##  [6,] 0.04392211 0.05686891
##  [7,] 0.06101807 0.07352245
##  [8,] 0.04228373 0.05221363
##  [9,] 0.02334978 0.03120262
## [10,] 0.03549752 0.04504689
## [11,] 0.02728179 0.03520663
## [12,] 0.01862460 0.02367106
## [13,] 0.02420756 0.03470747
## [14,] 0.01690265 0.02365127
## [15,] 0.01862639 0.02596847
## [16,] 0.01152179 0.01700207
## [17,] 0.03321396 0.05298283
## [18,] 0.02622174 0.04145582
## [19,] 0.02263524 0.03217871
```

#Graph of the 95% Wilson Confidence Interval

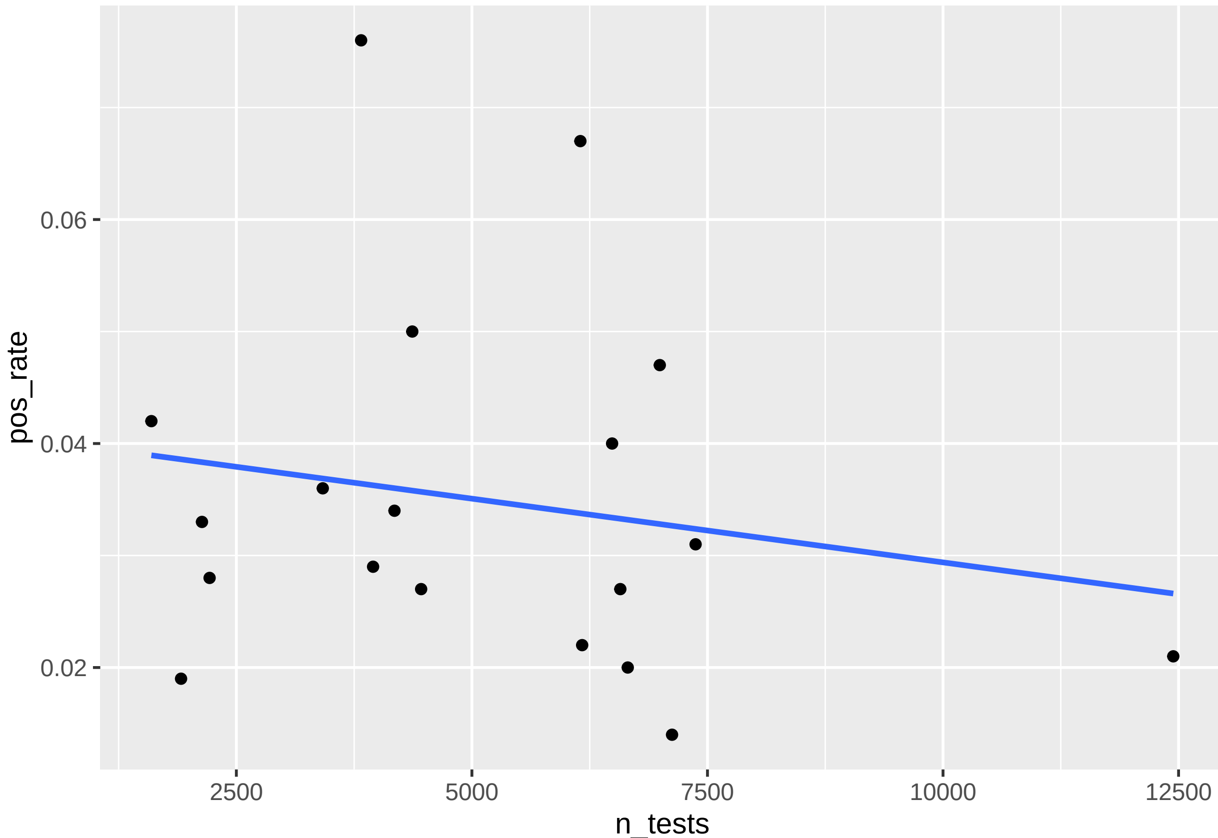
```
Final_data = data.frame(label = 1:19, lower_bound = CI_Wilson_matrix[,1], pos_rate = phat, upper_bound =
mdf_Fianl_data = reshape2::melt(Final_data, id.var = "label")
```

```
ggplot(data = mdf_Fianl_data , aes(label, value, colour = variable)) +
  geom_point() +
  geom_line() +
  labs(title = "95% Wilson Confidence Interval", x = "Weeks", y = "Probability")
```



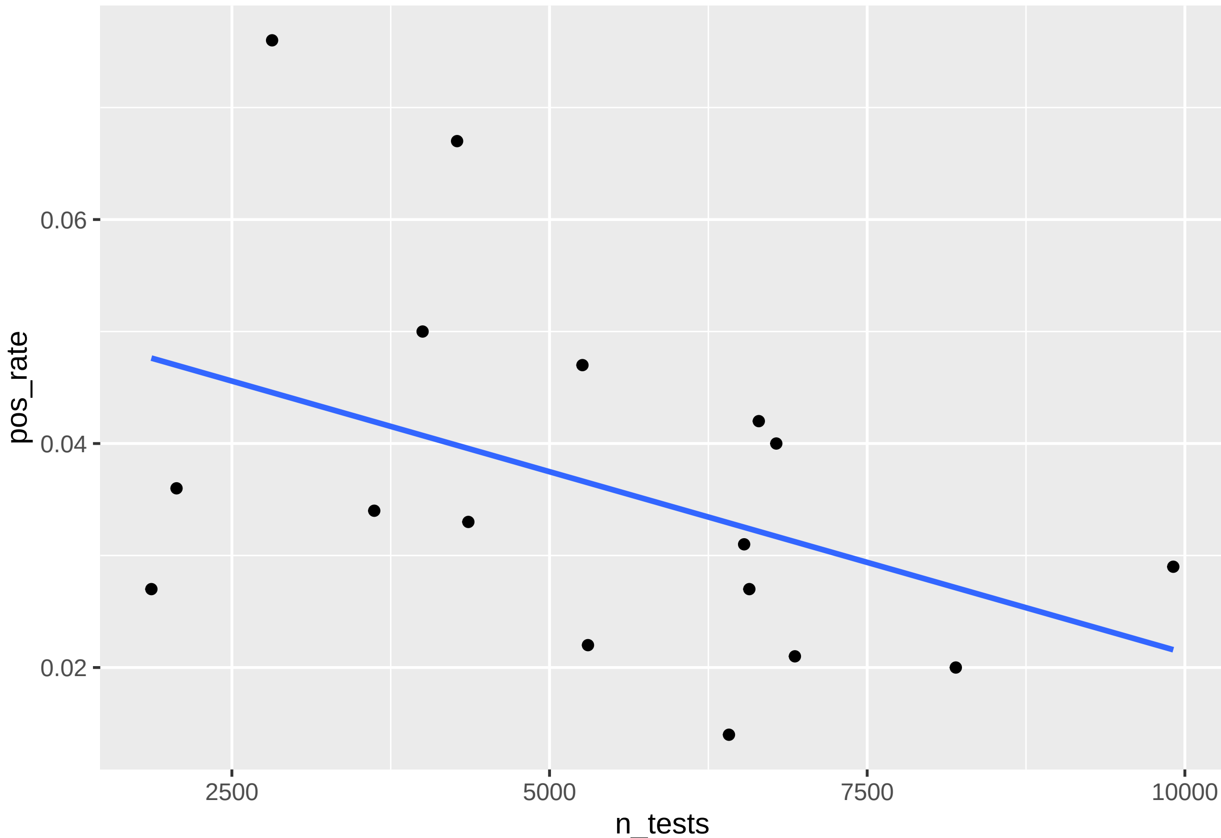
#2.relationship between n_tests and pos_rate

```
ggplot(data = mydata, mapping = aes(x = n_tests, y = pos_rate)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y~x, se = FALSE)
```



#relationship between n_tests of last two weeks and pos_rate

```
n_2week_ago = (n[-c(18,19)]+n[-c(1,19)])/2
phat_new = phat[-c(1,2)]
twoweekdata = data.frame(pos_rate = phat_new, n_tests = n_2week_ago)
ggplot(data = twoweekdata, mapping = aes(x = n_tests, y = pos_rate)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y~x, se = FALSE)
```



#Slope and correlation of the pos_rate and n_tests

```r
lm(pos_rate~n_tests, data = mydata)
```

```
##
## Call:
## lm(formula = pos_rate ~ n_tests, data = mydata)
##
## Coefficients:
## (Intercept)      n_tests
##   4.077e-02   -1.138e-06
```

```r
lm(pos_rate~n_tests, data = twoweekdata)
```

```
##
## Call:
## lm(formula = pos_rate ~ n_tests, data = twoweekdata)
##
## Coefficients:
## (Intercept)      n_tests
##   5.367e-02   -3.238e-06
```

```r
cor(mydata$pos_rate, mydata$n_tests)
```

```
## [1] -0.1843524
```

```r
cor(twoweekdata$pos_rate, twoweekdata$n_tests)
```

```
## [1] -0.4269501
```

#Quantile and 95% Confidence Interval of phat

```r
confint(lm(phat~1), level = 0.95)
```

```
##                  2.5 %     97.5 %
## (Intercept) 0.02713145 0.04265802
```

#Difference between phat for (n_tests) <= 5000 and (n_tests) > 5000

```r
nhigh_data = mydata %>%
  filter(n_tests > 5000)
nlow_data = mydata %>%
  filter(n_tests <= 5000)
mean(nhigh_data[,2])
```

```
## [1] 0.03211111
```

```r
mean(nlow_data[,2])
```

```
## [1] 0.0374
```

#Difference between phat for (past two weeks n_tests) <= 5000 and (past two weeks n_tests) > 5000

```r
n_2week_high_data = twoweekdata %>%
  filter(n_tests > 5000)
n_2week_low_data = twoweekdata %>%
  filter(n_tests <= 5000)
mean(n_2week_high_data[,1])
```

```
## [1] 0.0293
```

```r
mean(n_2week_low_data[,1])
```

```
## [1] 0.04614286
```

#3.LR test - positivity rate was constant within each 4 time intervals
#H_0 : p = p0 where p0 is mean(pos_rate) for the interval
#Checks the hypothesis for each week

```r
Firstmydata = mydata[1:5, ]
Secondmydata = mydata[6:10, ]
Thirdmydata = mydata[11:15, ]
Fourthmydata = mydata[16:19, ]
```

```r
First_phat = rep(mean(Firstmydata[,2]), length(Firstmydata[,2]))
Second_phat = rep(mean(Secondmydata[,2]), length(Secondmydata[,2]))
Third_phat = rep(mean(Thirdmydata[,2]), length(Thirdmydata[,2]))
Fourth_phat = rep(mean(Fourthmydata[,2]), length(Fourthmydata[,2]))

log_lik = function(n,p, phat){
  Logic = (2*(dbinom(round(n*phat), n, phat, log=TRUE) - dbinom(round(n*phat), n, p, log=TRUE)) - qchis
  return(Logic)
}
log_lik(Firstmydata[,3], Firstmydata[,2], First_phat)
```

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

```r
log_lik(Secondmydata[,3], Secondmydata[,2], Second_phat)
```

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

```r
log_lik(Thirdmydata[,3], Thirdmydata[,2], Third_phat)
```

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

```r
log_lik(Fourthmydata[,3], Fourthmydata[,2], Fourth_phat)
```

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

#LR test - positivity rate was different across at least two of these intervals
#Merge the interval and check the null hypothesis again.

```r
FirstMergemydata = mydata[1:10, ]
FirstMerge_phat = rep(mean(FirstMergemydata[,2]), length(FirstMergemydata[,2]))
log_lik(FirstMergemydata[,3],FirstMergemydata[,2], FirstMerge_phat)
```

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

```r
SecondMergemydata = mydata[11:19, ]
SecondMerge_phat = rep(mean(SecondMergemydata[,2]), length(SecondMergemydata[,2]))
log_lik(SecondMergemydata[,3],SecondMergemydata[,2], SecondMerge_phat)
```

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

#p-value from chi-square test for each interval

```r
chi_test = function(n,p, phat){
  pval = chisq.test(n, p = p, rescale.p = TRUE)
  return(pval)
}

chi_test(Firstmydata[,3], Firstmydata[,2], First_phat)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  n
## X-squared = 1807.2, df = 4, p-value < 2.2e-16
```

```r
chi_test(Secondmydata[,3], Secondmydata[,2], Second_phat)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  n
## X-squared = 4483.9, df = 4, p-value < 2.2e-16
```

```r
chi_test(Thirdmydata[,3], Thirdmydata[,2], Third_phat)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  n
## X-squared = 9156, df = 4, p-value < 2.2e-16
```

```r
chi_test(Fourthmydata[,3], Fourthmydata[,2], Fourth_phat)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  n
## X-squared = 19222, df = 3, p-value < 2.2e-16
```

```r
chi_test(FirstMergemydata[,3],FirstMergemydata[,2], FirstMerge_phat)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  n
## X-squared = 9387.8, df = 9, p-value < 2.2e-16
```

```r
chi_test(SecondMergemydata[,3],SecondMergemydata[,2], SecondMerge_phat)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  n
## X-squared = 31752, df = 8, p-value < 2.2e-16
```