Lecture 3 - R code

WILD 502- Jay Rotella

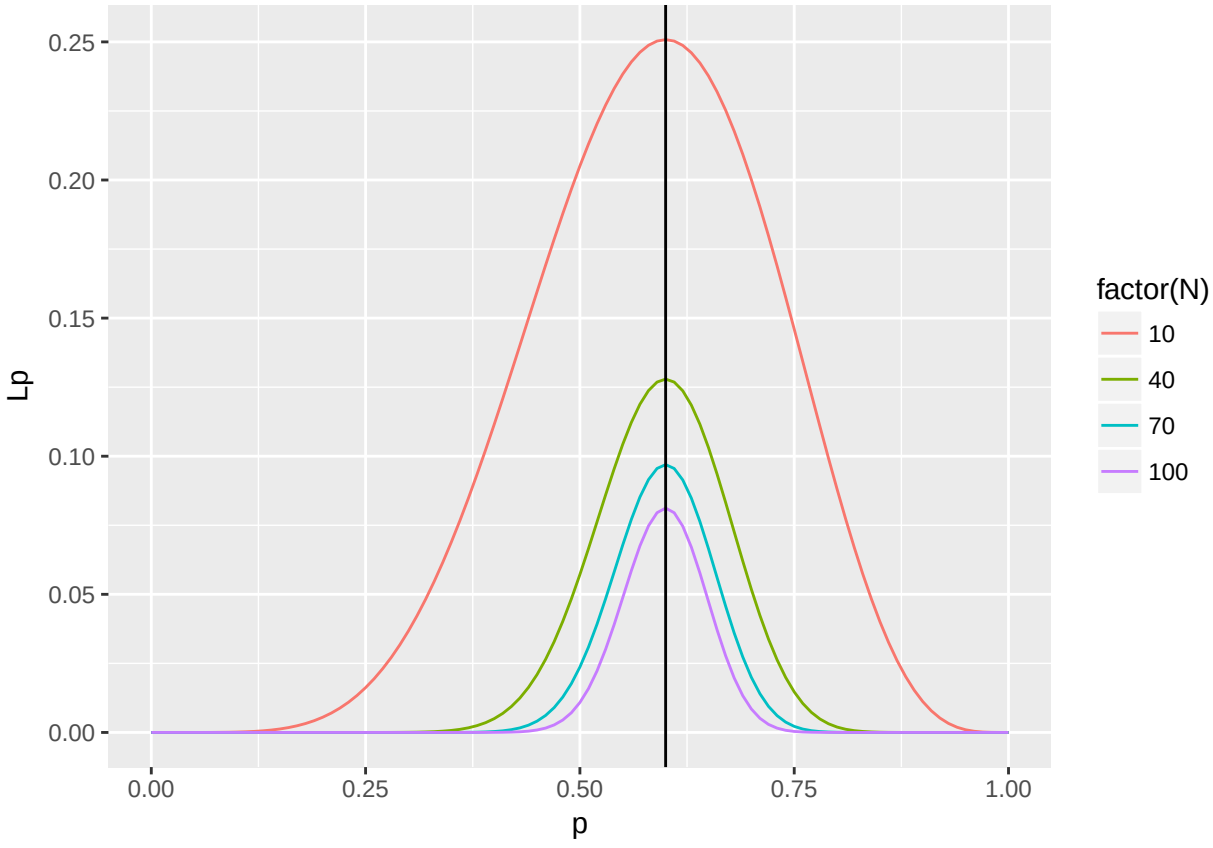
Likelihood Values and Sample Size

First, let's take a look at how the likehood values for various values of p vary as we change the sample size but keep the proportion of successes the same.

```
library(tidyverse) # for dplyr and ggplot2
library(binom) # for binomial confidence intervals
library(knitr) # for 'kable' function for printing nice table
true.p = 0.6
p=seq(from = 0, to = 1, by = 0.01)

# make a dataframe with combinations of p and N
out <- expand.grid(p = seq(0, 1, 0.01), N = seq(10, 100, 30))
# add y and likelihood values for each value of p
out <- out %>%
  mutate(y = true.p * N,
         Lp = choose(N, y) * p^(y) * (1-p)^(N-y),
         lnLik = log(Lp))

# plot likelihoods for each value of N
ggplot(out, aes(x = p, y = Lp, color = factor(N))) +
          geom_line() + geom_vline(xintercept = true.p)
```



If we calculate confidence intervals based on the likelihood profile, we can see how the uncertainty in $\hat{p}$ changes as sample size changes.

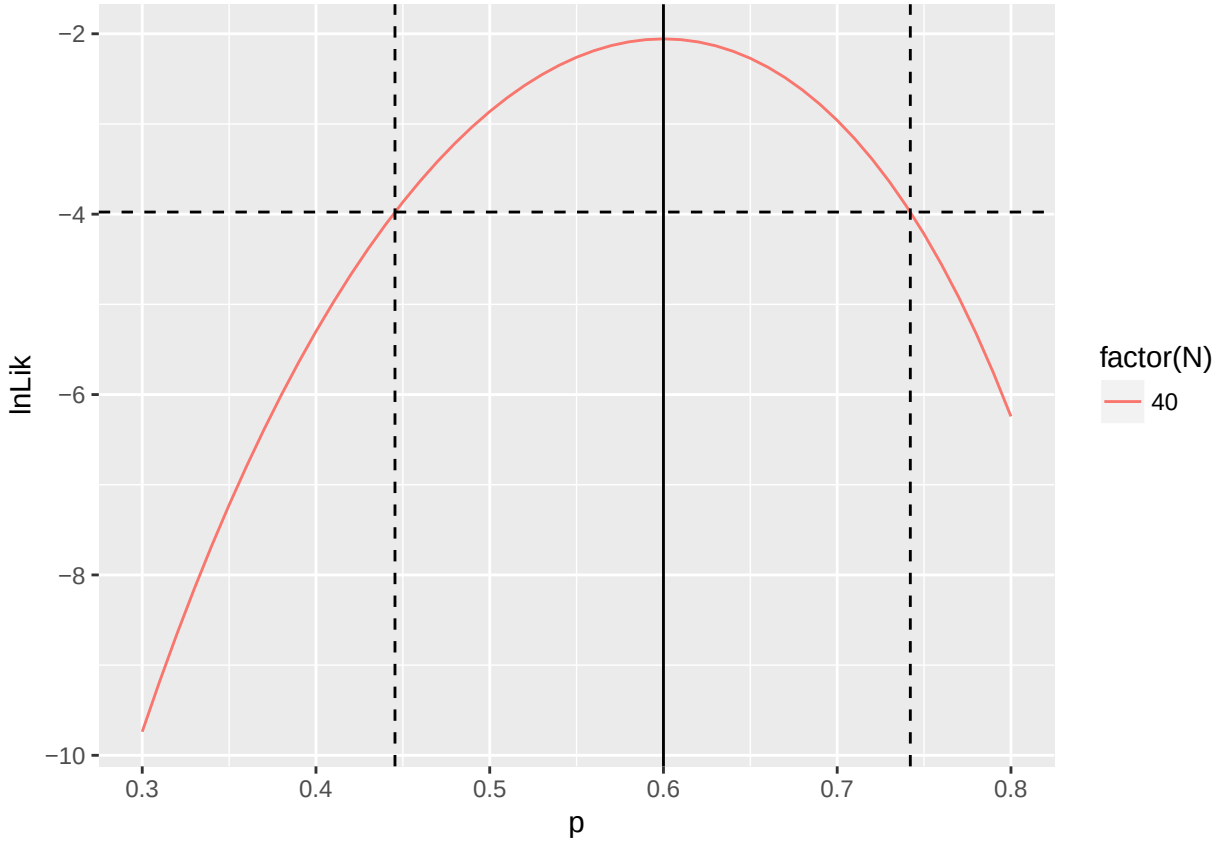
```
CIs = binom.profile(x = 0.6 * seq(10, 100, 30),
              n = seq(10, 100, 30),
              conf.level = 0.95)
kable(CIs)
```

method	x	n	mean	lower	upper
profile	6	10	0.6	0.3000155	0.8542937
profile	24	40	0.6	0.4454692	0.7420869
profile	42	70	0.6	0.4831658	0.7096343
profile	60	100	0.6	0.5023482	0.6925867

As explained in Chapter 1 of CW(*see page 1-22*), "... for a single parameter, likelihood theory shows that the 2 points 1.92 units down from the maximum of the log likelihood function provide a 95% confidence interval when there is no extra-binomial variation ...". The graphic below shows this for the scenario where N = 40 and y = 24.

```
out40 <- filter(out, N == 40, between(p, 0.3, 0.8))
max_ln_lik <- max(out40$lnLik)

ggplot(out40, aes(x = p, y = lnLik, color = factor(N))) +
  geom_line() + geom_vline(xintercept = true.p) +
  geom_hline(yintercept = max_ln_lik - 1.92, linetype = "dashed") +
  geom_vline(xintercept = c(CIs[2, 5], CIs[2, 6]), linetype = "dashed")
```



Converting between Probability and Log-odds

This simple exercise is intended to give you a better understanding of the connections between the log-odds of an outcome and the probability of an outcome. Recall that if the probability of an event is 0.25, that the value for the log-odds is calculated as $ln(\frac{0.25}{0.75})$. In R, you can use the `qlogis` function to obtain the log-odds for a given probability, e.g., `qlogis(0.25)` = -1.0986123.

If you know the log-odds, then you can calculate the probability by using $\frac{exp(x)}{1+exp(x)}$, where x represents the log-odds value. The In R, you can use the `plogis` function to obtain the probability for a given log-odds value, e.g., `plogis(-1.098612)` = 0.25.

Probability values range from 0 to 1. It turns out that for $\frac{exp(x)}{1+exp(x)}$, values of x ranging from -5 to +5 create probabilities that range from just above 0 to very close to 1. Values of x ranging from -1 to +1 create probabilities that range from about 0.25 to 0.75. The material below will let you explore the relationships for yourself.

```
log_odds = seq(from = -5, to = 5, by = 0.25)
# use 'plogis' function to calculate exp(x)/(1 + exp(x))
y = plogis(log_odds)
# store log_odds and y in data frame for use with ggplot
d = data.frame(log_odds, y)
head(d, 4)
```

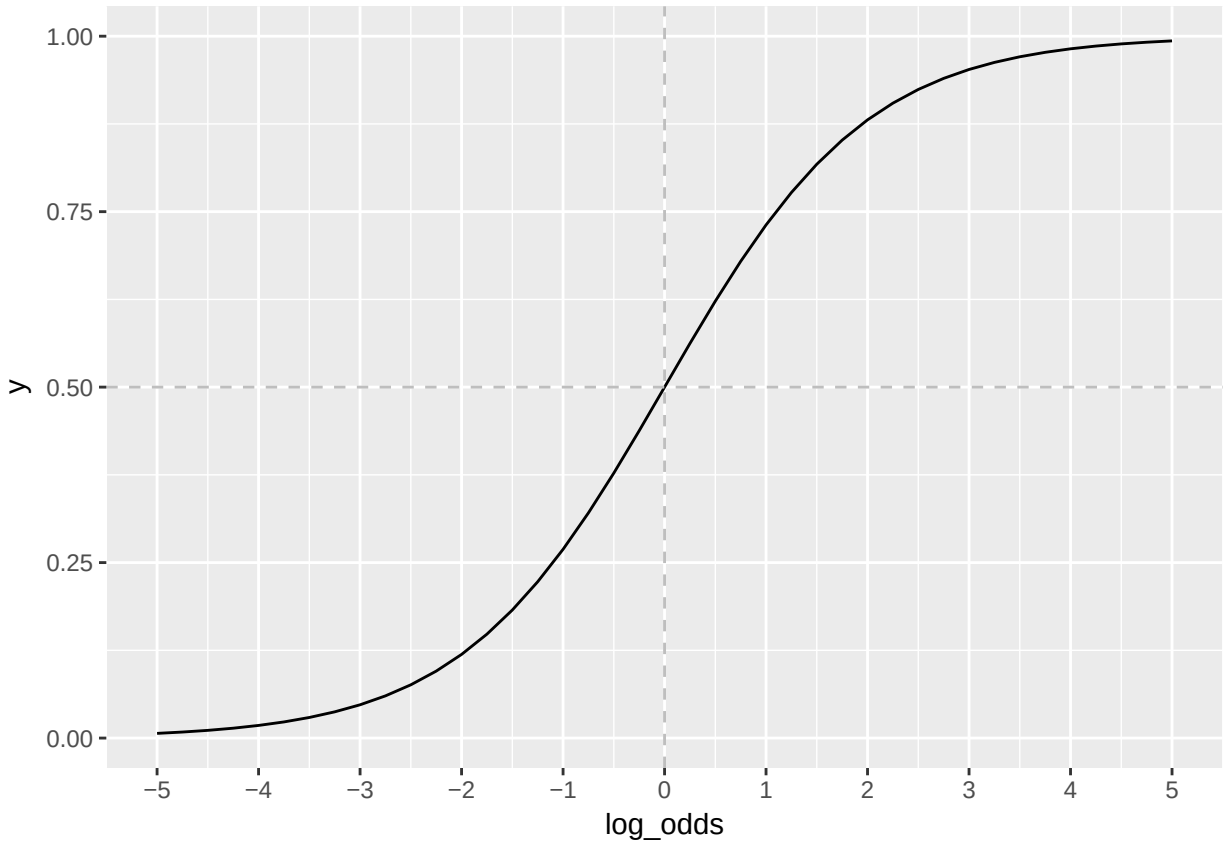
```
##   log_odds           y
## 1    -5.00 0.006692851
## 2    -4.75 0.008577485
## 3    -4.50 0.010986943
## 4    -4.25 0.014063627
```

```
tail(d, 4)
```

```
##    log_odds         y
## 38     4.25 0.9859364
## 39     4.50 0.9890131
## 40     4.75 0.9914225
## 41     5.00 0.9933071
```

Below, we plot the relationship, so you can see the pattern among the values for log-odds and associated probabilities. You might wonder what happens if you get log-odds values that are very very small (e.g., -24, -147, or -2421) or very big (e.g.,14, 250, or 1250). You should use the `plogis` function on such values (no commas in your numbers, e.g., `plogis(-2421)`) to find out for yourself.

```
ggplot(d, aes(x = log_odds, y = y)) +
  geom_line() +
  geom_hline(aes(yintercept = 0.5),
             colour = "gray",
             linetype = "dashed") +
             geom_vline(aes(xintercept = 0.0),
             colour = "gray",
             linetype = "dashed") +
  scale_x_continuous(breaks = seq(-5, 5, by = 1))
```


1.00
0.75
0.50
0.25
0.00
y
−5
−4
−3
−2
−1
0
1
2
3
4
5
log_odds