1. Reporting a binomial test

The null hypothesis assumes that both dative constructions are equiprobable. If that's the case, then the chance of seeing one dative over the other, or our p = .5. The sample size is equal to the sum of the occurrences of both dative types, n = 501 + 1859 = 2360. Our x will be the number of the occurrences of prepositional dative (though it could also be the double object dative), so that x = 501.

The above null hypothesis can be tested in R using the following expressions:

```
x <- 501 #prepositional dative
n <- 501 + 1859 #all dative occurrences
p <- .5 #50/50 probability null hypothesis

binom.test(x, n, p)
```

The test yields the following result:

```
Exact binomial test

data:  x and n
number of successes = 501, number of trials = 2360, p-value < 2.2e-16
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.1959431 0.2293504
sample estimates:
probability of success
             0.2122881
```

The p-value below 2.2e-16 is less than the alpha coefficient of .05, suggesting that the null hypothesis should be rejected, i.e. the two dative types are not equally likely to occur. At alpha = .05 the test was significant. The 95% confidence interval is from 0.196 (lower bracket) to 0.224 (upper bracket).

2. McNemar's test

Listed below are the commands I used in this assignment:

```
#set working directory
setwd("/Users/zub/Google Drive/CUNY/Stats for Ling Research/Homework/HW2")

#read file and save it as ptb variable, enable header row, and disable '#' as a comment character
ptb <- read.csv("/Users/zub/Google Drive/CUNY/Stats for Ling Research/Homework/HW2/PTB.csv", header = TRUE, comment.char = "")

#compute a boolean vector of correct Stanford predictions
stanford.correct <- ptb$gold.tag == ptb$Stanford.tag
#compute a boolean vector of correct NLP4J predictions
```

```
nlp4j.correct <- ptb$gold.tag == ptb$NLP4J.tag

#compute the number of wins of Stanford over NLP4J and store it in x1
x1 <- sum(stanford.correct & !nlp4j.correct)

#compute the number of wins of NLP4J over Stanford and store it in x2
x2 <- sum(nlp4j.correct & !stanford.correct)

x1 #print the value of x1
x2 #print the value of x2

#run a test to confirm or disprove the null hypothesis

x <- min(x1,x2)
n <- x1 + x2
p <- .5

binom.test(x, n, p)
```

Here is the result of the binomial test

```
Exact binomial test

data:  x and n
number of successes = 943, number of trials = 1959, p-value = 0.1038
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.459029 0.503763
sample estimates:
probability of success
              0.481368
```

Here are the answers:

1. The number of wins (x1) of the stanford tagger over the NLP4J tagger is 943.
2. The number of wins (x2) of the NLP4J tagger over the Stanford taggers is 1016.
3. At alpha = .05 the p-value is .1038, which is around twice the size of alpha. This indicates that the alternative hypothesis is weak. We reject the alternative hypothesis but do not reject the null hypothesis, which suggests that neither tagger is significantly better than the other one.