

# Problem Set 6 – MATH 392

*Theodore Dounias*

*3/17/2018*

## 1.1

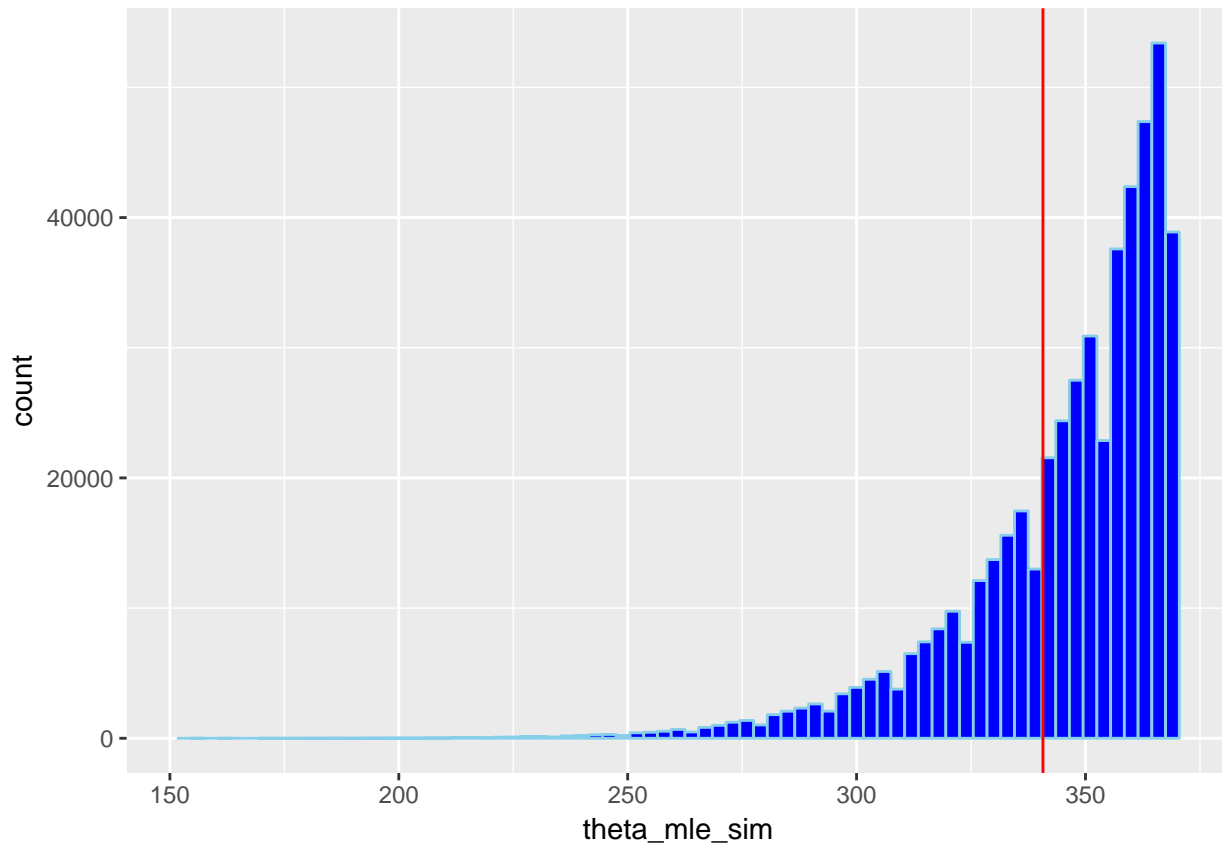
I will use simulation to first find the sampling distribution of  $\hat{\theta}_{MLE,corr} = \frac{n+1}{n}X_{max}$ ; I will draw samples of size 14 from a  $\text{Unif}(0, 345)$  and calculate the test statistic. I will be rounding the samples up, assuming there is no panzer tank with a serial number of 0. In this way each serial number has an equal probability of being selected.

```
theta_mle <- ceiling((15/14)*max(serial))

it <- 500000
theta_mle_sim <- rep(0, it)
x <- rep(0, 14)
for(i in 1:it){
  x <- ceiling(runif(14, 0, 345))
  theta_mle_sim[i] <- (15/14)*max(x)
}

data <- data.frame(theta_mle_sim)

ggplot(data) +
  geom_histogram(aes(theta_mle_sim), binwidth = 3, col = "skyblue", fill = "blue") +
  geom_vline(aes(xintercept = (15/14)*max(serial)), col = "red")
```



```
p_value <- (sum(theta_mle_sim >= theta_mle)+1)/(it + 1)
p_value
```

```
## [1] 0.6798866
```

## 1.2

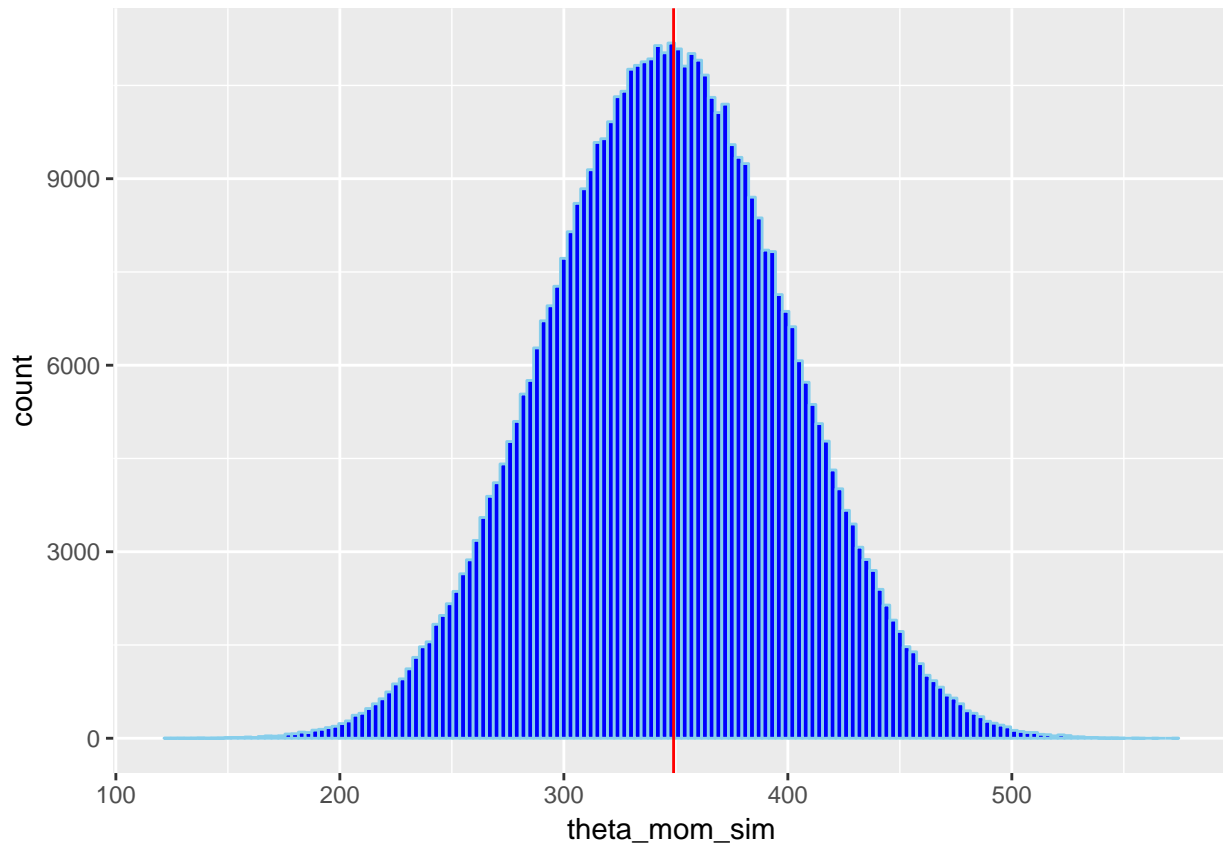
$$\hat{\theta}_{MOM} = 2\bar{X}$$

```
theta_mom <- ceiling(2*mean(serial))

it <- 500000
theta_mom_sim <- rep(0, it)
x <- rep(0, 14)
for(i in 1:it){
  x <- ceiling(runif(14, 0, 345))
  theta_mom_sim[i] <- 2*mean(x)
}

data <- data.frame(theta_mom_sim)

ggplot(data) +
  geom_histogram(aes(theta_mom_sim), binwidth = 3, col = "skyblue", fill = "blue") +
  geom_vline(aes(xintercept = ceiling(2*mean(serial))), col = "red")
```



```
p_value <- (sum(theta_mom_sim >= theta_mom)+1)/(it + 1)
p_value
```

```
## [1] 0.478905
```

### 1.3

Hypothesis testing under this framework is susceptible to type I and type II errors. A type I error would occur if we rejected the null incorrectly. In this specific example, if we rejected the null, we would risk Germany regrouping and increasing their resources while we stand by, despite our capability of dealing a deathblow.

A Type II error occurs when we accept the null hypothesis, and the alternative is true. In the scenario before us, we would overstep into Germany and would likely suffer a critical defeat.

### 1.4

I will again use a simulation approach to make these calculations. I know that the following two equations connect power and alpha in this one sided test:

$$P(\hat{\theta} > q) = 1 - \alpha, \quad 1 - P(\theta > q) = \beta$$

Here theta is the parameter from the true distribution, and theta hat is assuming that the null is true.

With that in mind, I need to simulate the distributions and calculate the probabilities above at a 99% confidence level. I choose .99 and not .95 due to the increased need for certainty, as a miscalculation might lead to a large military defeat. I will write this as a function to save space:

```

mom <- function(x){
  ceiling(2*mean(x))
}

mle <- function(x){
  ceiling((18/17)*max(x))
}

power <- function(true, funct){
  it <- 10000

  alpha <- .99

  #Distribution under the null
  h0_dist <- rep(0, it)
  for(i in 1:it){
    x <- ceiling(runif(17, 0, 345))
    h0_dist[i] <- funct(x)
  }

  #True distribution
  true_dist <- rep(0, it)
  for(i in 1:it){
    x <- ceiling(runif(17, 0, true))
    true_dist[i] <- funct(x)
  }

  #Calculate value
  q <- quantile(h0_dist, probs = alpha)

  (sum(true_dist >= q) + 1)/(it+1)
}

mle_power <- c(power(325, mle), power(335, mle), power(345, mle),
               power(355, mle), power(365, mle))

mom_power <- c(power(325, mom), power(335, mom), power(345, mom),
               power(355, mom), power(365, mom))

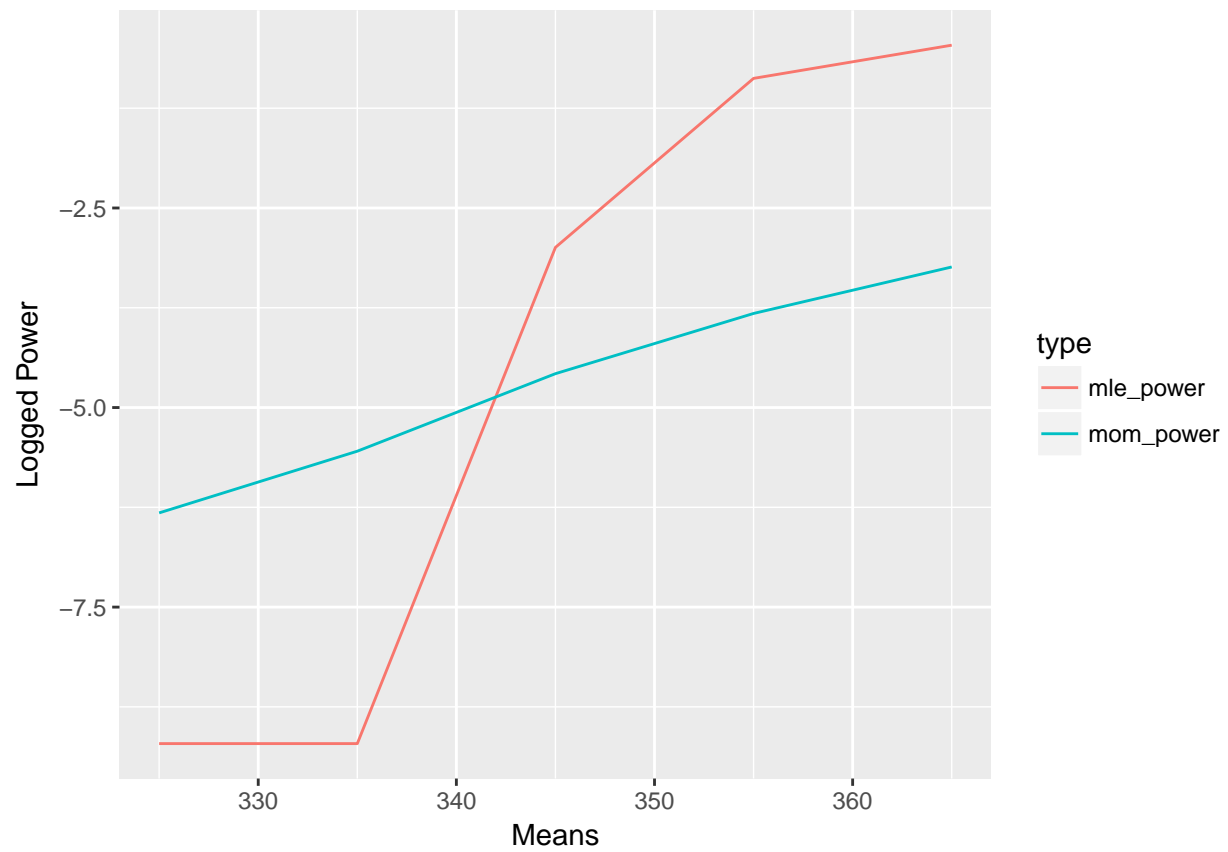
diffs <- seq(325, 365, 10)

data <- data.frame(mle_power, mom_power, diffs)

data <- data %>%
  gather("type", "Value", 1:2)

#I graph logged values because of large divergence in values
ggplot(data, aes(x = diffs)) +
  geom_line(aes(y = log(Value), col = type)) +
  xlab("Means") +
  ylab("Logged Power")

```



I would use the MLE estimator; the power of MOM tests seem relatively static, while with some tweaks to the mean I use for the test it seems I could get a significantly more statistically potent result.