ECON312 Problem Set 2B: question 1

Futing Chen, Hongfan Chen, Will Parker

05/20/2020

Contents

A. For each of the K independent normal samples, test if the population mean is zero using "t" tests: **2**
for $K = 2$ . . . 2
for $K = 5$ . . . 2

B: Can you reject the null? **3**
F-statistic . . . 3
Order-statistic: $max_k\{X_k\}_{k=1}^K$ . . . 4
Step down method . . . 7

C: For each sample, characterize the test statistic for the Order statistic formally, and plot a Monte Carlo simulation of its distribution. **7**

D: Can you reject $\mu_k = 0$ or not? **8**

E: How would you conduct a meta-analysis across samples? Compare your meta-nalysis with each sample in B. **8**

F. How should you pool information over samples? **8**

```
library(tidyverse)
library(knitr)
library(readxl)
```

Load in data

```
sheets <- excel_sheets("PS2_Q1_Data.xlsx")


data_k2 <- readxl::read_excel("PS2_Q1_Data.xlsx", sheet = 1) %>%
  select(sample_1 = `0`, sample_2 = `1`) %>%
  pivot_longer(cols = everything(),
               names_prefix = "sample_",
               names_to = "k")


haven::write_dta(data_k2, "data_k2.dta")


data_k5 <- readxl::read_excel("PS2_Q1_Data.xlsx", sheet = 2) %>%
  select(sample_1 = `0`,
         sample_2 = `1`,
```

```
         sample_3 = `2`,
         sample_4 = `3`,
         sample_5 = `4`) %>%
  pivot_longer(cols = everything(),
               names_prefix = "sample_",
               names_to = "k")

haven::write_dta(data_k5, "data_k5.dta")
```

A. For each of the K independent normal samples, test if the population mean is zero using "t" tests:

for $K = 2$

```
summary_k2 <- data_k2 %>%
  group_by(k) %>%
  summarise(mean = mean(value),
            sd = sd(value),
            N = n()) %>%
  mutate(std_error = sqrt(sd/N),
         t_stat = mean/std_error,
          p_value = 2*pt(abs(t_stat), df = N -1, lower = FALSE)) %>%
  select(-N, -sd)


summary_k2 %>%
  kable(digits = 3, col.names = c("k", "$\\hat{\\mu}_k$", "$se(\\mu_k)$","$t_k$", "p-value"))
```

k	$\hat{\mu}_k$	$se(\mu_k)$	t_k	p-value
1	-0.069	0.103	-0.666	0.507
2	-0.016	0.102	-0.161	0.872

for $K = 5$

```
summary_k5 <- data_k5 %>%
  group_by(k) %>%
  summarise(mean = mean(value),
            sd = sd(value),
            N = n()) %>%
  mutate(std_error = sqrt(sd/N),
         t_stat = mean/std_error,
          p_value = 2*pt(abs(t_stat), df = N -1, lower = FALSE)) %>%
  select(-N, -sd)

summary_k5 %>%
  kable(digits = 3, col.names = c("k", "$\\hat{\\mu}_k$", "$se(\\mu_k)$","$t_k$", "p-value"))
```

k	$\hat{\mu}_k$	$se(\mu_k)$	t_k	p-value
1	-0.025	0.092	-0.265	0.791
2	-0.124	0.095	-1.312	0.193
3	0.083	0.101	0.820	0.414
4	-0.077	0.100	-0.767	0.445
5	0.085	0.101	0.838	0.404

B: Can you reject the null?

F-statistic

Under the null hypothesis by the CLT we have

$$\sqrt{N} * \begin{pmatrix} \hat{\mu}_1 \\ ... \\ \hat{\mu}_k \end{pmatrix} \xrightarrow{d} N(0, \Sigma)$$

So the logic of the F-test is that we want to test the linear restriction that all the sample means are zero

$$\mu_1 = 0, \mu_2 = 0, ... = \mu_k = 0$$

So we set

$$R = I_k$$

Denoting $\begin{pmatrix} \hat{\mu}_1 \\ ... \\ \hat{\mu}_k \end{pmatrix} = \hat{\mu}$, by the continuous mapping theorem we have

$$\sqrt{n} * (R\hat{\mu}) \xrightarrow{d} N(0, R\hat{\Sigma}R')$$

Our test statistic is then

$$T_n = N(R\hat{\mu})(R\hat{\Sigma}R')^{-1}(R\hat{\mu})'$$

Where $N = k * 100$. Since $R = I_k$, this reduces down to

$$T_n = N(\hat{\mu}\hat{\Sigma}^{-1}\hat{\mu}')$$

This statistic is distributed as χ^2_k in the limit, so the F-statistic is

$$\frac{100 * k * (\mu\Sigma^{-1}\mu')}{k} = 100 * (\mu\Sigma^{-1}\mu') \sim F(k, N - k)$$

```
fstat_calc <- function(data){

  summary_data <- data %>%
    group_by(k) %>%
  summarise(mean = mean(value))
```

```
  mu <- summary_data$mean

  sigma <- data %>%
            pivot_wider(names_from = k, values_from = value) %>%
            unnest() %>%
            as.matrix() %>%
            cov()

  k <- length(mu)

  n <- k*100

  t(mu)%*%mu

  F_stat <- as.numeric(100*(t(mu)%*%solve(sigma)%*%mu))

  p_value <- as.numeric(pf(F_stat, df1 = k, df2 = n - k, lower.tail = FALSE))

  return(tibble(k, n, F_stat, p_value))
}

fstat_calc(data_k2) %>%
  rbind(fstat_calc(data_k5)) %>%
  kable(digits = 3, col.names =  c("K", "N", "F", "p-value"))
```

K	N	F	p-value
2	200	0.422	0.656
5	500	5.924	0.000

For the $K = 5$ sample, we reject the joint hypothesis that $\mu_1 = \mu_2 = ... = \mu_k = 0$ by the F-test at level $\alpha = 0.05$.

Order-statistic: $max_k\{X_k\}_{k=1}^K$

Under the null hypothesis that $\mu_k = 0$ by the CLT we know the distribution of each sample mean $\bar{X_k}$

$$\sqrt{n}(\bar{X_k} - 0) \xrightarrow{d} N(0, 1)$$

let $Y = max\{\bar{X_1}, .., \bar{X_K}\}$. Since the $\bar{X_k}$ are iid r.v. from the same distribution, we know that the CDF of Y can be written as

$$F(Y) = P(Y \leq x) = P(\bar{X_1} \leq x, ..., \bar{X_K} \leq x) = F(\bar{X_k})^K$$

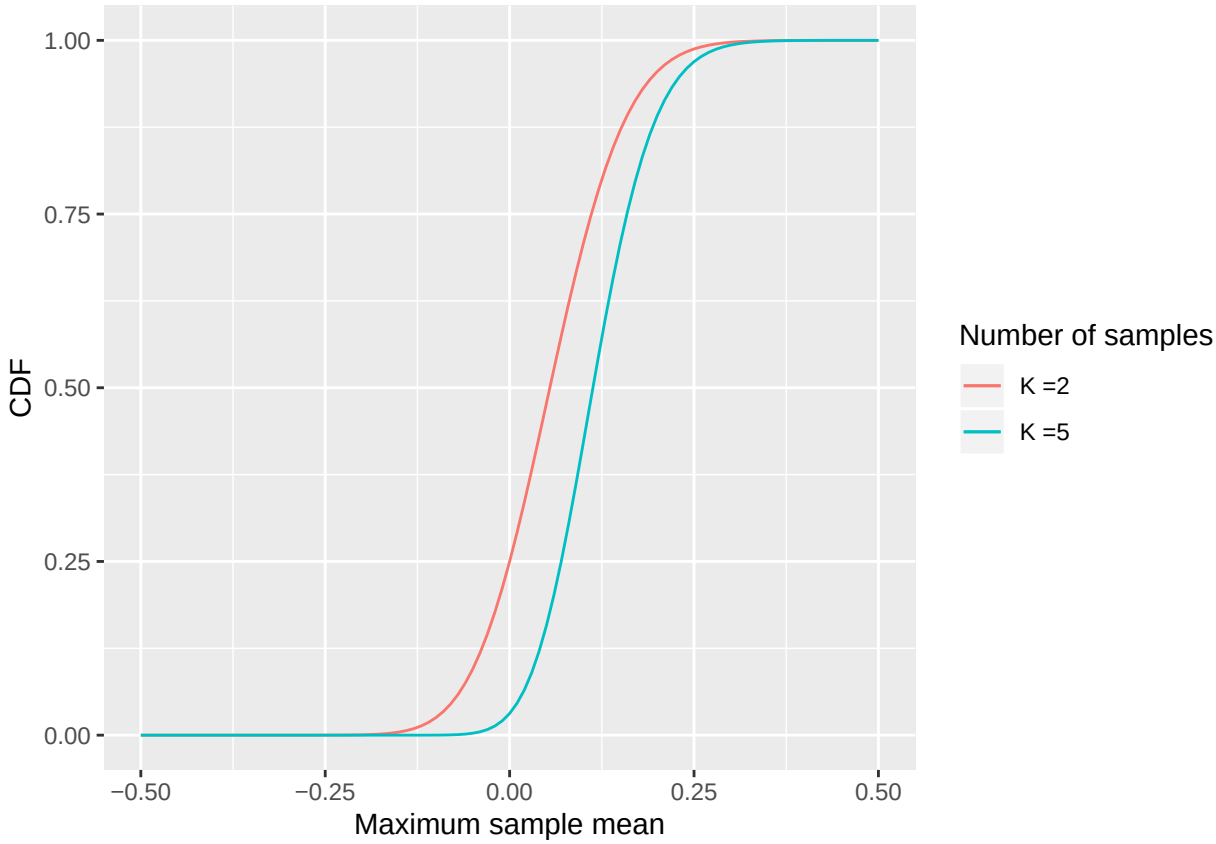
We know that $F(\bar{X}) = \frac{1}{\sqrt{n}}\Phi(x)$. So we can write the distribution of the order statistic as

$$F(max\{\bar{X_1}, .., \bar{X_K}\}) = \Phi(\sqrt{n}x)^K$$

Or more specifically:

$$F(max\{\bar{X_1}, .., \bar{X_K}\}) = (\frac{1}{2} * (1 + erf(\frac{\sqrt{n} * x}{\sqrt{2}}))^K$$

```
CDF_Xbar_max <- function(x, K, n = 100){
  return((pnorm(sqrt(n)*x))^K)
}


tibble(x = seq(-0.5, 0.5, 0.01)) %>%
  mutate(CDF_k2 = CDF_Xbar_max(x, K=2),
         CDF_k5 = CDF_Xbar_max(x, K=5)) %>%
  ggplot(aes(x =x)) +
    geom_line(aes(y = CDF_k2, color = "K =2")) +
    geom_line(aes(y = CDF_k5, color = "K =5")) +
  lims(x = c(-0.5, 0.5)) + labs(x = "Maximum sample mean", y = "CDF", color = "Number of samples")
```



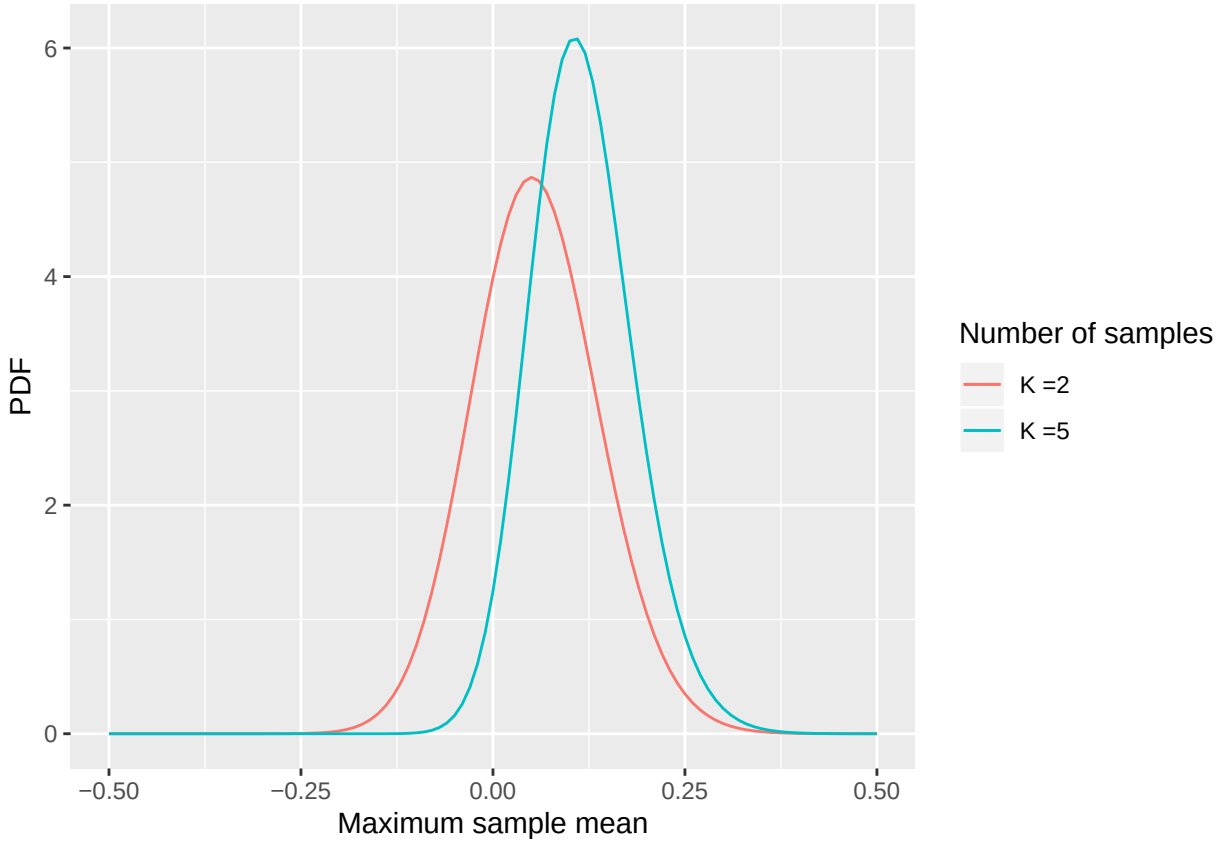
CDF of $Y = max\{\bar{X_1}, .., \bar{X_K}\}$ where each sample is a mean of $n = 100$ draws from $X \sim N(0, 1)$

Differentiating the CDF with respect to X we get

$$f(max\{\bar{X_1}, .., \bar{X_K}\}) = \sqrt{n} * K * \Phi(\sqrt{n} * x)^{K-1} * \phi(\sqrt{n} * x)$$

```
pdf_Xbar_max <-  function(x, K, n = 100){
  return(sqrt(n)*K*(pnorm(sqrt(n)*x)^(K-1))*dnorm(sqrt(n)*x))
}

tibble(x = seq(-0.5, 0.5, 0.01)) %>%
```

```
  mutate(pdf_k2 = pdf_Xbar_max(x, K=2),
         pdf_k5 = pdf_Xbar_max(x, K=5)) %>%
  ggplot(aes(x =x)) +
    geom_line(aes(y = pdf_k2, color = "K =2")) +
    geom_line(aes(y = pdf_k5, color = "K =5")) +
  lims(x = c(-0.5, 0.5)) + labs(x = "Maximum sample mean", y = "PDF", color = "Number of samples")
```




```
pvalue_order <- function(summary_data){

  mu <- summary_data$mean

  max_Xbar <- max(mu)

  k <- length(mu)

  grid_search <- tibble(x = seq(-0.5, 0.5, 0.01)) %>%
    mutate(CDF = CDF_Xbar_max(x, K = k),
           observed_value = max_Xbar,
           diff = abs(x- observed_value))

  CDF_observed_value <- filter(grid_search, diff == min(grid_search$diff))$CDF

  p_value <- 1- CDF_observed_value

  return(tibble(k, max_Xbar, p_value))
}
```

```r
pvalue_order(summary_k2) %>%
  rbind(pvalue_order(summary_k5)) %>%
  kable(col.names = c("K", "$max(\\bar{X})$", "p-value"), digits = 3)
```

K	$max(\bar{X})$	p-value
2	-0.016	0.823
5	0.085	0.696

Here the p-values represent the probability of observing a maximum sample mean greater than the observed maximum in the data under the null hypothesis that

$$\mu_1 = 0, \mu_2 = 0, ... = \mu_k = 0$$

Step down method

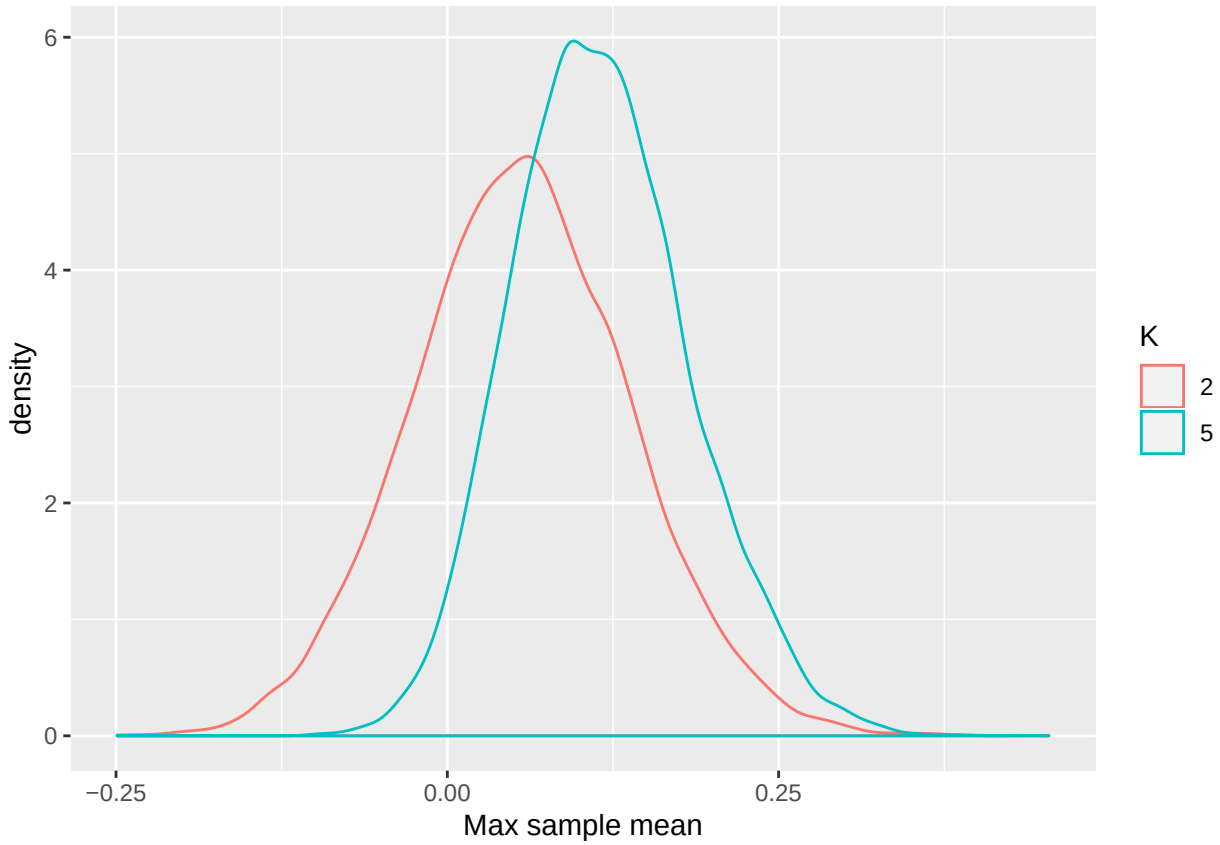
$K = 2$

$K = 5$

C: For each sample, characterize the test statistic for the Order statistic formally, and plot a Monte Carlo simulation of its distribution.

```r
mc_sim_order_dist <- function(K, n = 100, N = 10000){
  output <- tibble(K = numeric(), max_Xbar  = numeric())

  for (i in seq(1:N)){
    Xbars <- rnorm(K, sd = 1/sqrt(n))

    Y <- max(Xbars)

    output <- output %>%
      rbind(tibble(K, Y))

  }

  return(output)
}

mc_samples <- mc_sim_order_dist(2) %>%
  rbind(mc_sim_order_dist(5))
```

```r
mc_samples %>%
  ggplot(aes(x = Y, color = factor(K))) +
  geom_density() +
  labs(x = "Max sample mean", y = "density", color = "K")
```



Result of 10,000 monte carlo simulations for the order statistic under the null hypothesis

D: Can you reject $\mu_k = 0$ or not?

E: How would you conduct a meta-analysis across samples? Compare your meta-nalysis with each sample in B.

F. How should you pool information over samples?