Homework 6

Winter 2024

KEY

2024-02-28

Exercises

1.

a. Find the values for $\bar{x}$, $\bar{y}$, σ_1 and σ_2.

```
n1 <- 25
n2 <- 36

xbar <- (1.37 + 1.53)/2
ybar <- (1.17 + 1.29)/2

sigma1 <- sqrt(n1)*((1.53-1.37)/2)/qnorm(p=0.975)
sigma2 <- sqrt(n2)*( (1.29 - 1.17)/2)/qnorm(p=0.975)

cat("xbar:", xbar, "ybar:", ybar, "sigma1", round(sigma1,4), "sigma2", round(sigma2,4) )
```

```
## xbar: 1.45 ybar: 1.23 sigma1 0.2041 sigma2 0.1837
```

b. The expected value and variance of $\bar{X} - \bar{Y}$ is shown below:

$$\begin{aligned}
E\left[\bar{X} - \bar{Y}\right] &= E\left[\bar{X}\right] - E\left[\bar{Y}\right], \\
&= \mu_1 - \mu_2. \\
Var\left[\bar{X} - \bar{Y}\right] &= Var\left[\bar{X}\right] + Var\left[\bar{Y}\right] \qquad \text{independence} \\
&= \frac{\sigma_1^2}{25} + \frac{\sigma_2^2}{36} \\
SD(\bar{X} - \bar{Y}) &= \sqrt{\frac{\sigma_1^2}{25} + \frac{\sigma_2^2}{36}}
\end{aligned}$$

Therefore

$$\bar{X} - \bar{Y} \approx Norm\left(\mu_1 - \mu_2, \sqrt{\frac{\sigma_1^2}{25} + \frac{\sigma_2^2}{36}}\right)$$

c. Just as we did with a single mean, we can use the distribution of $\bar{X} - \bar{Y}$ to construct a confidence interval for $\mu_1 - \mu_2$. We simply consider the probability lying under this distribution between -1.96 and

1.96. Specifically

$$P\left(-1.96 \leq \frac{\bar{X}-\bar{Y}-(\mu_1-\mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}}} \leq 1.96\right) = 0.95.$$

We then invert the event inside the probability to keep $\mu_1 - \mu_2$ in the middle of the inequalities. In other words:

$$P\left(\bar{X}-\bar{Y}-1.96\times\sqrt{\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}} \leq (\mu_1-\mu_2) \leq (\bar{X}-\bar{Y})+1.96\times\sqrt{\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}}\right) = 0.95.$$

Therefore a 95% confidence interval for $\mu_1 - \mu_2$ is

$$\bar{X}-\bar{Y} \pm 1.96\sqrt{\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}}.$$

For this data, we can use the answers from part a to get the interval.}

```
xbar - ybar + c(-1,1)*1.96*sqrt(sigma1^2/n1 + sigma2^2/n2)
```

```
## [1] 0.1199982 0.3200018
```

2. A large sample $100(1-\alpha/2)\%$ confidence interval estimator for π_0 is given by

$$\hat{\pi}_0 \pm z_{\alpha/2}\sqrt{\frac{\hat{\pi}_0 \times (1-\hat{\pi}_0}{n}}$$

where $z_{\alpha/2}$ is the critical value which will ensure the desired level of confidence. For a 90% confidence interval, $z_{\alpha/2} = 1.645$.

For this data, we have $\hat{\pi}_0 = \frac{54}{83}$. The interval is calculated below.

```
n <- 83
pihat <- (23+15+6+8)/n



ci <- pihat +c(-1,1)*1.645*sqrt( pihat*(1-pihat)/n)
```

With 90% confidence, the probability that X exceeds 2 is in the range 0.539, 0.714

3. The formula for a large sample $100(1-\alpha)\%$ confidence interval for π_0 is

$$\hat{\pi}_0 \pm z_{\alpha/2}\sqrt{\frac{\hat{\pi}_0 \times (1-\hat{\pi}_0)}{n}}$$

where $z_{\alpha/2}$ is the number such that there is an area of $1-\alpha$ between $\pm z_{\alpha/2}$.

For a 96% confidence interval, $\alpha = 0.04$ and therefore $z_{\alpha/2} = qnorm(p = 0.02) = -2.0537$.

In order to ensure with 96% confidence that $\hat{\pi}_0$ is no further from π_0 than 0.05, we need

$$n \geq \frac{2.0537^2 \times \frac{1}{4}}{0.05^2} = 422$$

.

Similarly, we can show that in order to ensure with 92% confidence that $\hat{\pi}_0$ is no further from π_0 than 0.04, we need

$$n \geq \frac{1.7507^2 \times \frac{1}{4}}{0.04^2} = 479$$

Therefore, we need a larger sample size for requiring 92% confidence that $\hat{\pi}_0$ is no further from π_0 than 0.04.

4.

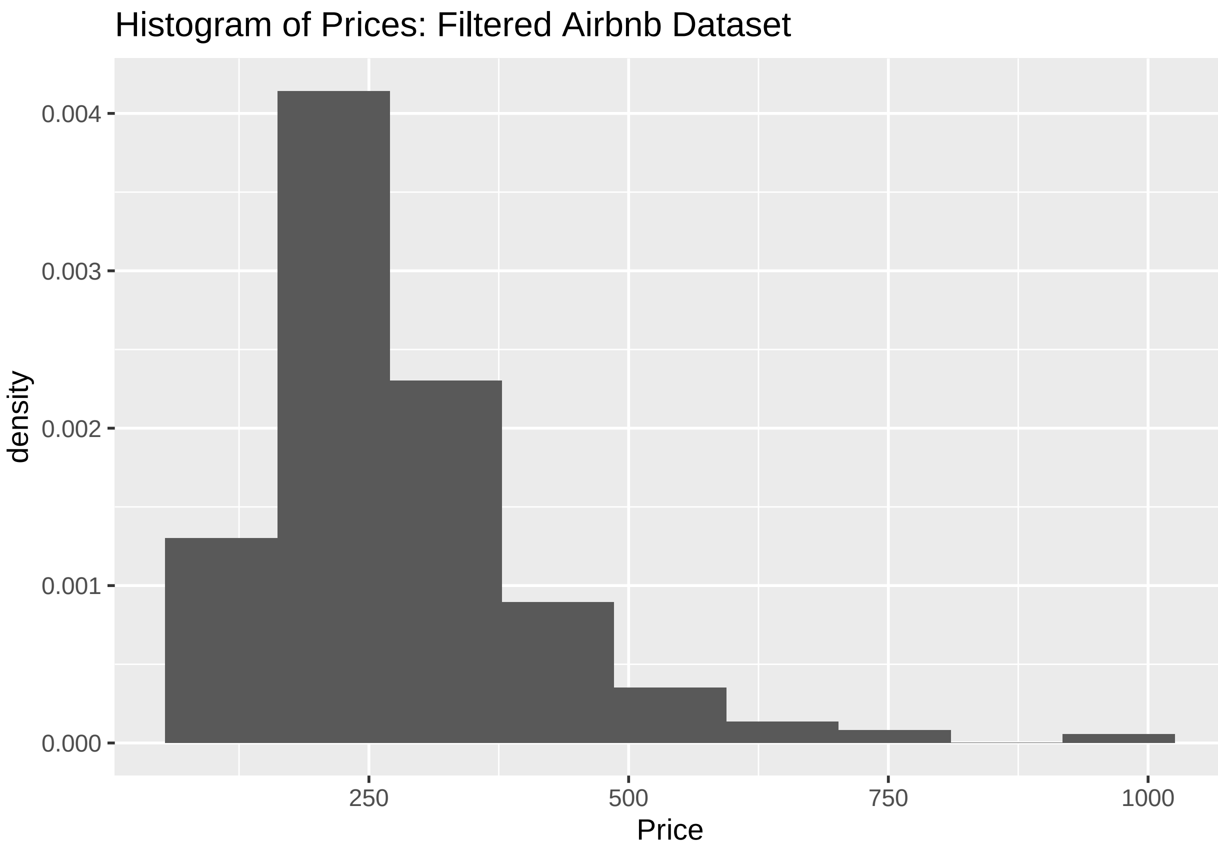
a.

```
#create airbnb_3bed

airbnb <- read_csv("listings.csv")
airbnb_3bed <- airbnb %>% filter(property_type == "House",
                                 room_type == "Entire home/apt",
                                 bedrooms >= 3) %>%
   mutate(price = parse_number(price)) %>%
   select(price)

# glimpse
airbnb_3bed %>% glimpse()
```

```
## Rows: 342
## Columns: 1
## $ price <dbl> 975, 450, 461, 700, 450, 600, 450, 325, 175, 222, 348, 400, 170,~
```

b.

```
#make histogram of price
ggplot(data=airbnb_3bed,
       mapping = aes(x=price,
                     y=after_stat(density)))+
   #binwidth ~ diff(range(x))/log2(nrow(airbnb_new))
   geom_histogram(binwidth = 108)+
   labs(title = "Histogram of Prices: Filtered Airbnb Dataset",
        x = 'Price',
        y = 'density')
```




```
#calculate median price

airbnb_3bed %>%
                 summarise(pop_median = median(price))
```

```
## # A tibble: 1 x 1
##   pop_median
##        <dbl>
## 1        250
```

The shape of the population distribution of price is skewed to the right. The median price in the population is $250

c.

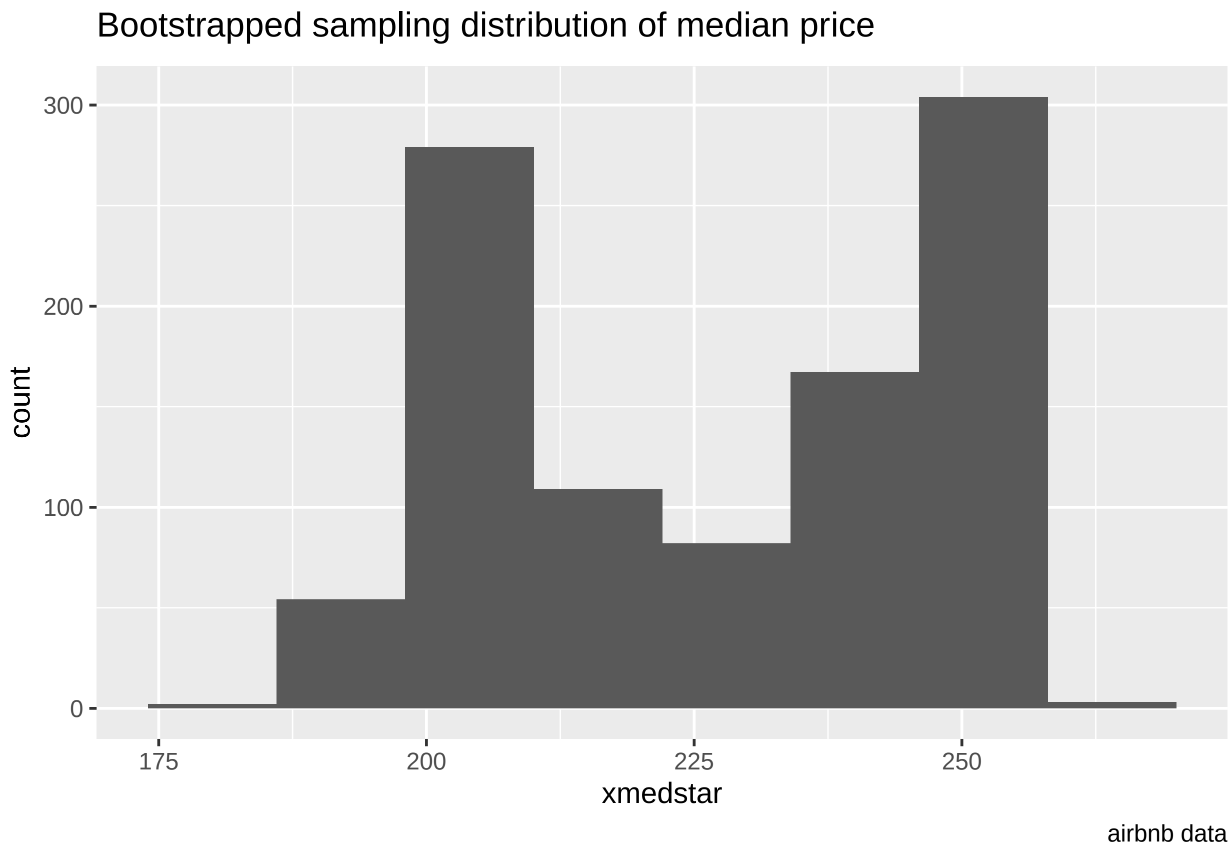
```
set.seed(1512974)

airbnb_3bed_sample <- airbnb_3bed %>% slice_sample(n=50)

obs_median <- airbnb_3bed_sample %>%
             summarise(muhat =median(price)) %>% pull()

cat("Median price in sample", obs_median)
```

```
## Median price in sample 230
```

The median price in the sample is $230

d.

```r
#generate sample of n = 50 from airbnb_3bed_sample
#with replacement and then calculate median.
#Repeat B times

set.seed(14141)
B = 1000

boot_df <- tibble(
   xmedstar = replicate(n = B,
                        expr = median(sample(
                          x =airbnb_3bed_sample$price,
                          size = nrow(airbnb_3bed_sample),
                          replace = TRUE))))



#make a histogram of the bootstrap estimates

ggplot(data = boot_df,
       mapping = aes(x = xmedstar) ) +
  geom_histogram(binwidth = 12) +
  labs(title = expression(paste("Bootstrapped sampling distribution of median price"  )),
       caption="airbnb data")
```



The bootstrapped sampling distribution of the median of price appears bi-modal. But we must keep in mind that this is only based on $B = 1,000$ replications.

```
#calculated bias-corrected point estimate and 95% percentile-based confidence interval
boot_results <- boot_df %>%
                        summarise( boot_mean = mean(xmedstar),
                                   boot_se = sd(xmedstar),
                                   bias = boot_mean - obs_median,
                                   bias_corrected_est = obs_median - bias,
                                   lower = quantile(xmedstar, 0.01),
                                   upper = quantile(xmedstar, 0.99))

boot_results
```

```
## # A tibble: 1 x 6
##   boot_mean boot_se     bias bias_corrected_est lower upper
##       <dbl>   <dbl>    <dbl>              <dbl> <dbl> <dbl>
## 1   225.258 21.5102 -4.74250            234.742   195   250
```

The bias corrected estimate and lower and upper limits of the 98% bootstrap percentile interval is shown above. I chose the percentile-based interval since the sampling distribution is not symmetric.

In one sentence, the (bias-corrected) estimate of the population median price based on the sample of 50 listings is \$234.7425. With 98% confidence, the smallest the population median price can be is \$195 and the largest it can be \$250