Data 606 Lab 6

Joao De Oliveira

2025-10-19

Overview

In this lab, we will explore and visualize the data using the tidyverse suite of packages, and perform statistical inference using infer. The data can be found in the companion package for OpenIntro resources, openintro.

Load Packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(openintro)
```

```
## Loading required package: airports
## Loading required package: cherryblossom
## Loading required package: usdata
```

```
library(infer)
```

Exercise 1

What are the counts within each category for the amount of days these students have texted while driving within the past 30 days?

```
data(yrbss)

glimpse(yrbss)
```

```
## Rows: 13,583
## Columns: 13
## $ age                      <int> 14, 14, 15, 15, 15, 15, 15, 14, 15, 15, 15, 1~
## $ gender                   <chr> "female", "female", "female", "female", "fema~
## $ grade                    <chr> "9", "9", "9", "9", "9", "9", "9", "9", "9", ~
## $ hispanic                 <chr> "not", "not", "hispanic", "not", "not", "not"~
## $ race                     <chr> "Black or African American", "Black or Africa~
## $ height                   <dbl> NA, NA, 1.73, 1.60, 1.50, 1.57, 1.65, 1.88, 1~
## $ weight                   <dbl> NA, NA, 84.37, 55.79, 46.72, 67.13, 131.54, 7~
## $ helmet_12m               <chr> "never", "never", "never", "never", "did not ~
## $ text_while_driving_30d   <chr> "0", NA, "30", "0", "did not drive", "did not~
## $ physically_active_7d     <int> 4, 2, 7, 0, 2, 1, 4, 4, 5, 0, 0, 0, 4, 7, 7, ~
## $ hours_tv_per_school_day  <chr> "5+", "5+", "5+", "2", "3", "5+", "5+", "5+",~
## $ strength_training_7d     <int> 0, 0, 0, 0, 1, 0, 2, 0, 3, 0, 3, 0, 0, 7, 7, ~
## $ school_night_hours_sleep <chr> "8", "6", "<5", "6", "9", "8", "9", "6", "<5"~
```

```r
yrbss %>%
  count(text_while_driving_30d) %>%
  arrange(desc(n))
```

```
## # A tibble: 9 x 2
##   text_while_driving_30d     n
##   <chr>                  <int>
## 1 0                       4792
## 2 did not drive           4646
## 3 1-2                      925
## 4 <NA>                     918
## 5 30                       827
## 6 3-5                      493
## 7 10-19                    373
## 8 6-9                      311
## 9 20-29                    298
```

Answer 1

As we can see in the output, the count of high school students that claimed to never text while driving (0) in the past 30 days is 4792, 925 claimed to have done it between 1 and times, 493 did it between 3 and 5 times, 311 between 6 and 9 times, 373 between 10 and 19 times, 298 between 20 and 29 times, 827 claimed to have done it every day in the last 30 days, 4646 didn't drive in the past month, and 918 didn't answer or didn't know. This shows that the great majority claimed to never text while driving or they just didn't drive in that period. However, there are still high students who text while drive (827 every single day!).

Exercise 2

What is the proportion of people who have texted while driving every day in the past 30 days and never wear helmets?

```r
no_helmet <- yrbss %>%
  filter(helmet_12m == "never")

no_helmet <- no_helmet %>%
  mutate(text_ind = ifelse(text_while_driving_30d == "30", "yes", "no"))
```

```r
no_helmet %>%
  count(text_ind) %>%
  mutate(proportion = 100 * (n / sum(n)))
```

```
## # A tibble: 3 x 3
##   text_ind     n proportion
##   <chr>    <int>      <dbl>
## 1 no        6040      86.6
## 2 yes        463       6.64
## 3 <NA>       474       6.79
```

Answer 2

By looking at the output table, we can see that 86.6% who never wear helmets claim to not have texted, while 6.64 % did not wear helmet and texted while driving.

Exercise 3

What is the margin of error for the estimate of the proportion of non-helmet wearers that have texted while driving each day for the past 30 days based on this survey?

```r
no_helmet %>%
  filter(text_ind %in% c("yes", "no")) %>%
  specify(response = text_ind, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95) %>%
  mutate(ME = (upper_ci - lower_ci) / 2)
```

```
## # A tibble: 1 x 3
##   lower_ci upper_ci      ME
##      <dbl>    <dbl>   <dbl>
## 1   0.0655   0.0781 0.00631
```

Answer 3

The margin of error, as we can see in the output, is 0.00654 or 0.654 percentage points (comparing both percentages).

Exercise 4

Using the infer package, calculate confidence intervals for two other categorical variables (you'll need to decide which level to call "success", and report the associated margins of error. Interpet the interval in context of the data. It may be helpful to create new data sets for each of the two countries first, and then use these data sets to construct the confidence intervals.

```r
unique(yrbss$hours_tv_per_school_day)
```

```
## [1] "5+"           "2"            "3"            "do not watch" "<1"
## [6] "4"            "1"            NA
```

```
unique(yrbss$school_night_hours_sleep)
```

```
## [1] "8"   "6"   "<5"  "9"   "10+" "7"   "5"   NA
```

```
# success is watching less than 2h of tv
tv_hours_df <- yrbss %>%
  filter(!is.na(hours_tv_per_school_day)) %>%
  mutate(success = if_else(hours_tv_per_school_day %in% c("do not watch","<1","1","2"), "yes", "no")) %
  mutate(success = factor(success, levels = c("no","yes")))

table(tv_hours_df$success)
```

```
##
##   no  yes
## 4782 8463
```

```
prop.table(table(tv_hours_df$success))
```

```
##
##        no       yes
## 0.3610419 0.6389581
```

```
# Manual bootstrap for TV
set.seed(1234)
n_reps <- 1000
bootstrap_props <- replicate(n_reps, {
  sample_data <- sample(tv_hours_df$success, size = nrow(tv_hours_df), replace = TRUE)
  mean(sample_data == "yes")
})

ci_tv_low <- data.frame(
  lower_ci = quantile(bootstrap_props, 0.025),
  upper_ci = quantile(bootstrap_props, 0.975)
) %>%
  mutate(ME = (upper_ci - lower_ci) / 2)

p_hat_tv <- mean(tv_hours_df$success == "yes")

ci_tv_low
```

```
##       lower_ci  upper_ci          ME
## 2.5% 0.6304247 0.6474896 0.008532465
```

```
p_hat_tv
```

```
## [1] 0.6389581
```

```r
# success is sleeping 8h or more
sleep_hours_df <- yrbss %>%
  filter(!is.na(school_night_hours_sleep)) %>%
  mutate(success = if_else(school_night_hours_sleep %in% c("8","9","10+"), "yes", "no")) %>%
  mutate(success = factor(success, levels = c("no","yes")))

table(sleep_hours_df$success)
```

```
##
##   no  yes
## 8564 3771
```

```r
prop.table(table(sleep_hours_df$success))
```

```
##
##        no       yes
## 0.6942846 0.3057154
```

```r
# Manual bootstrap for sleep
set.seed(1234)
bootstrap_props_sleep <- replicate(n_reps, {
  sample_data <- sample(sleep_hours_df$success, size = nrow(sleep_hours_df), replace = TRUE)
  mean(sample_data == "yes")
})

ci_sleep_enough <- data.frame(
  lower_ci = quantile(bootstrap_props_sleep, 0.025),
  upper_ci = quantile(bootstrap_props_sleep, 0.975)
) %>%
  mutate(ME = (upper_ci - lower_ci) / 2)

p_hat_sleep <- mean(sleep_hours_df$success == "yes")

ci_sleep_enough
```

```
##       lower_ci  upper_ci          ME
## 2.5% 0.2975253 0.3135813 0.008027969
```

```r
p_hat_sleep
```

```
## [1] 0.3057154
```

```r
# Check proportions in each dataset
table(tv_hours_df$success)
```

```
##
##   no  yes
## 4782 8463
```

```r
table(sleep_hours_df$success)
```

```
##
##   no  yes
## 8564 3771
```

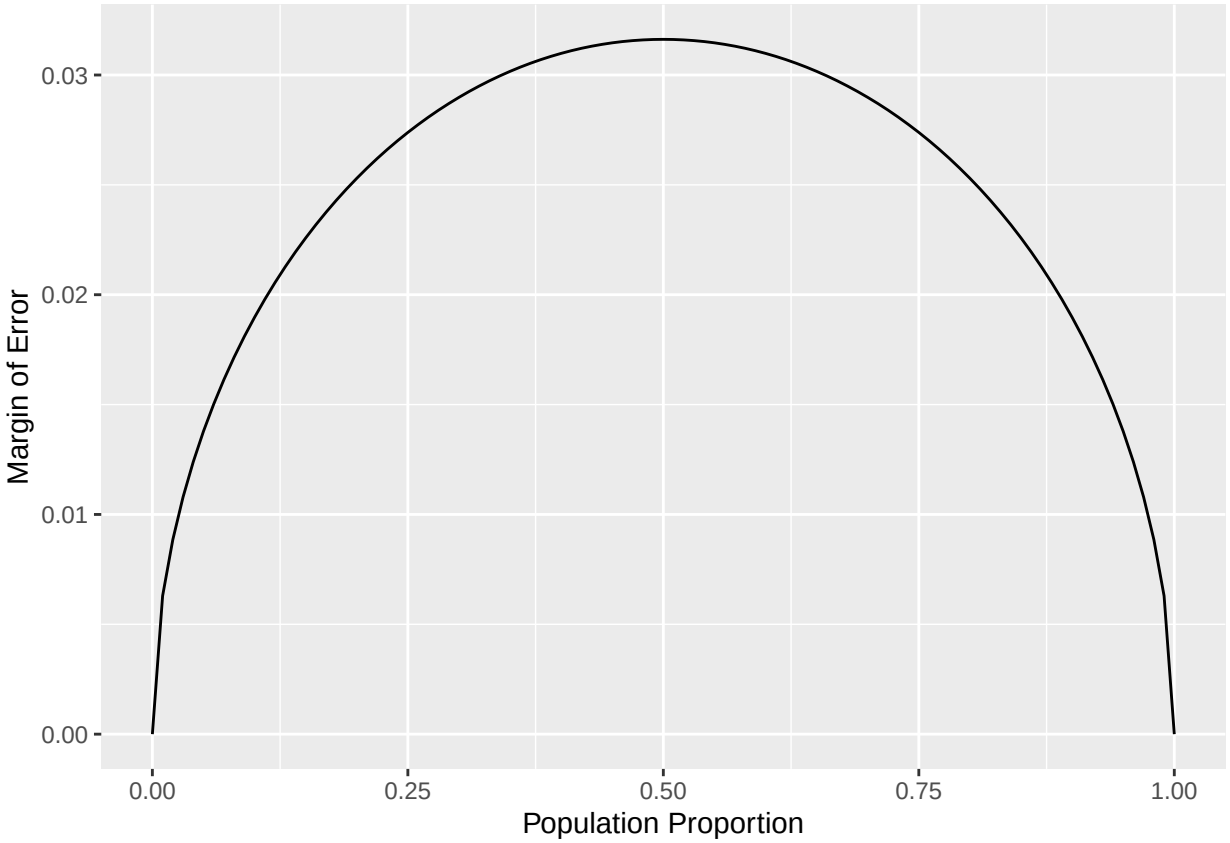
Answer 4

In the selected sample, the proportion of high school students who watch 2 or less hours of tv per day is 0.6390 or 63.90%. We can also say that we are 95% confident that around 63 to 64.75% (with a margin of error of 0.85 percentage points)of students watch 2 or less hours of tv per day, which suggests that most students don't watch more than 2h o tv per day. In the selected sample, the proportion of students who sleep 8 or more hours per day is 0.3057 or 30.57%. We are also 95% confident that around 29.75% and 31.36% (with a margin of error of 0.8 percentage points)of students sleep at least 8 hours of sleep per day.

Exercise 5

Describe the relationship between p and me. Include the margin of error vs. population proportion plot you constructed in your answer. For a given sample size, for which value of p is margin of error maximized?

```r
n <- 1000

# variable p and margin of error
p <- seq(from = 0, to = 1, by = 0.01)
me <- 2 * sqrt(p * (1 - p)/n)

# define data frame and plot
dd <- data.frame(p = p, me = me)
ggplot(data = dd, aes(x = p, y = me)) +
  geom_line() +
  labs(x = "Population Proportion", y = "Margin of Error")
```



Answer 5

By analyzing the plot, we can see that the relationship between p and me has an parabolic (curve) shape and is symmetric around p = 0.5. The margin of error is maximized at p = 0.5, decreasing as it gets closer to 0 and 1.

Exercise 6

Describe the sampling distribution of sample proportions at n=300 and p=0.1. Be sure to note the center, spread, and shape.

```
n <- 300
p <- 0.10
reps <- 1000

simulation <- tibble(p_hat = rbinom(reps, size = n, prob = p) / n)

summary <- simulation %>%
  summarise(
    mean_p_hat = mean(p_hat),
    np = n * p,
    np_2 = n * (1-p),
    sd_p_hat = sd(p_hat),
    q25 = quantile(p_hat, 0.025),
```

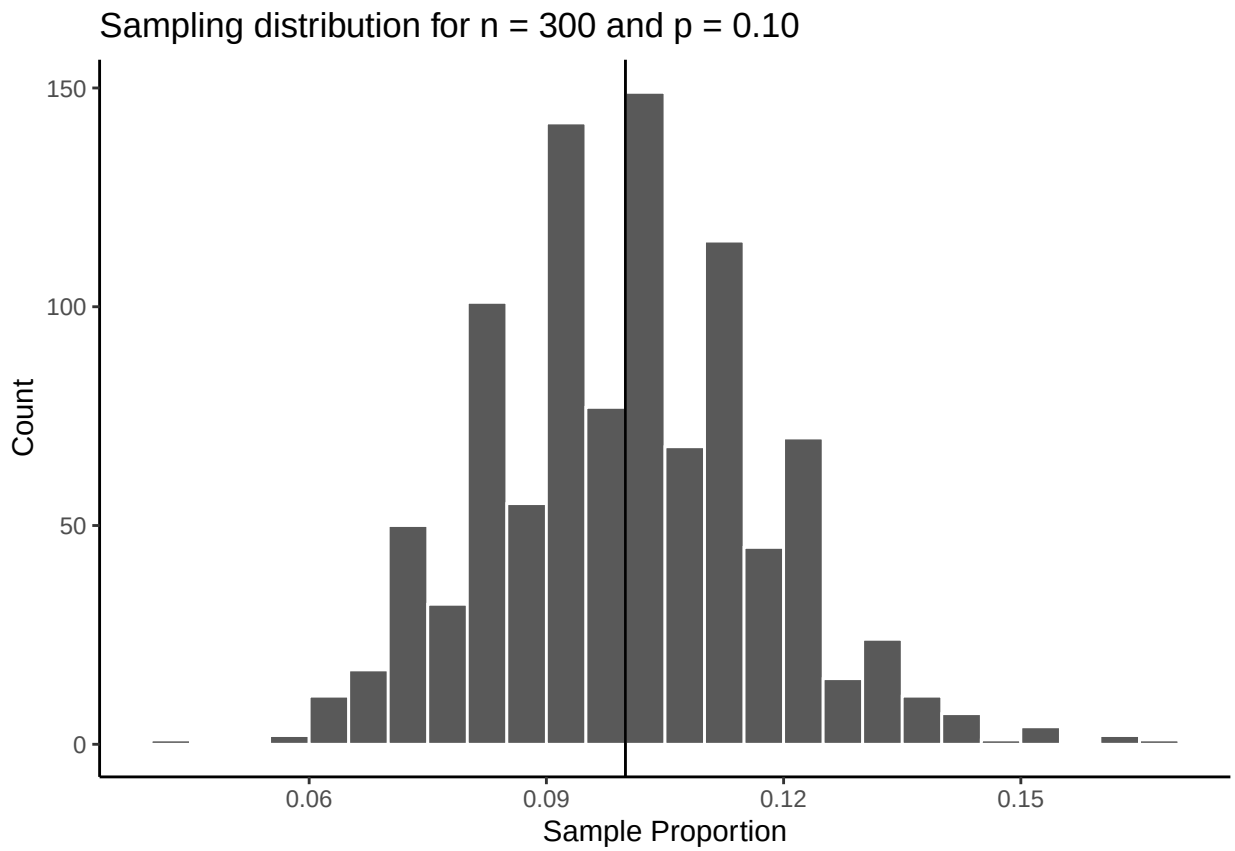
```
    q975 = quantile(p_hat, 0.975)
  )


summary
```

```
## # A tibble: 1 x 6
##   mean_p_hat    np  np_2 sd_p_hat    q25  q975
##        <dbl> <dbl> <dbl>    <dbl>  <dbl> <dbl>
## 1     0.0993    30   270   0.0174 0.0667 0.137
```

```
se <- 2 * sqrt(p * (1 - p)/n)
se
```

```
## [1] 0.03464102
```

```
ggplot(data = simulation, aes(x = p_hat)) +
  geom_histogram(binwidth = 0.005, boundary = 0, closed = "left", color = "white") +
  geom_vline(xintercept = p) +
  labs(
    title = "Sampling distribution for n = 300 and p = 0.10",
    x = "Sample Proportion",
    y = "Count"
    ) +
  theme_classic()
```



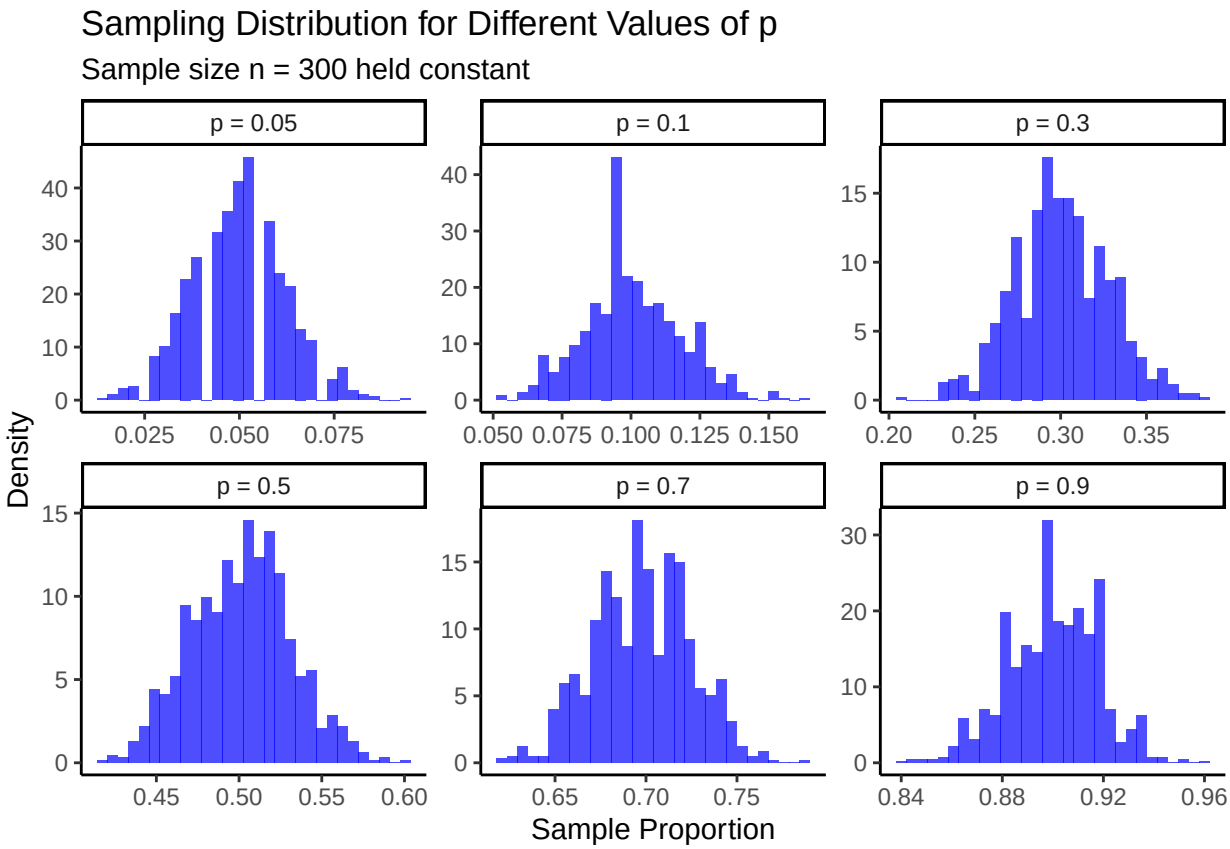
Answer 6

By looking at the graph, we can see that the sampling distribution of sample proportions has a center equal to 0.0993, which is very close to 0.1 (the true population proportion), a standard error of 0.0346, and approximately normal (bell_shaped) since it passes the success fail test because np = 30 and n(1-p) = 270 (both above 10).

Exercise 7

Keep n constant and change p. How does the shape, center, and spread of the sampling distribution vary as p changes. You might want to adjust min and max for the x-axis for a better view of the distribution.

```
n <- 300
p_values <- c(0.05, 0.1, 0.3, 0.5, 0.7, 0.9)
reps <- 1000

comparison_data <- data.frame()

for (p in p_values) {
  sample_props <- replicate(reps, {
    sample_data <- rbinom(n, 1, p)
    mean(sample_data)
  })

  temp_data <- data.frame(
    p_hat = sample_props,
    p_value = paste("p =", p)
  )
  comparison_data <- rbind(comparison_data, temp_data)
}

ggplot(comparison_data, aes(x = p_hat)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30, fill = "blue", alpha = 0.7) +
  facet_wrap(~p_value, scales = "free") +
  labs(
    title = "Sampling Distribution for Different Values of p",
    subtitle = paste("Sample size n =", n, "held constant"),
    x = "Sample Proportion",
    y = "Density"
  ) +
  theme_classic()
```




```
# Calculate stats for each p
stats_summary <- comparison_data %>%
  group_by(p_value) %>%
  reframe(
    mean = mean(p_hat),
    sd = sd(p_hat),
    theoretical_se = sqrt(unique(parse_number(p_value)) * (1 - unique(parse_number(p_value))) / n),
    np = n * p_hat,
    np_2 = n * (1-p_hat)
  )

print(stats_summary)
```

```
## # A tibble: 6,000 x 6
##    p_value    mean     sd theoretical_se    np  np_2
##    <chr>     <dbl>  <dbl>          <dbl> <dbl> <dbl>
##  1 p = 0.05 0.0500 0.0123         0.0126    13   287
##  2 p = 0.05 0.0500 0.0123         0.0126    12   288
##  3 p = 0.05 0.0500 0.0123         0.0126    16   284
##  4 p = 0.05 0.0500 0.0123         0.0126    14   286
##  5 p = 0.05 0.0500 0.0123         0.0126    21   279
##  6 p = 0.05 0.0500 0.0123         0.0126    14   286
##  7 p = 0.05 0.0500 0.0123         0.0126    17   283
##  8 p = 0.05 0.0500 0.0123         0.0126    17   283
##  9 p = 0.05 0.0500 0.0123         0.0126    11   289
```

```
## 10 p = 0.05 0.0500 0.0123         0.0126    14   286
## # i 5,990 more rows
```

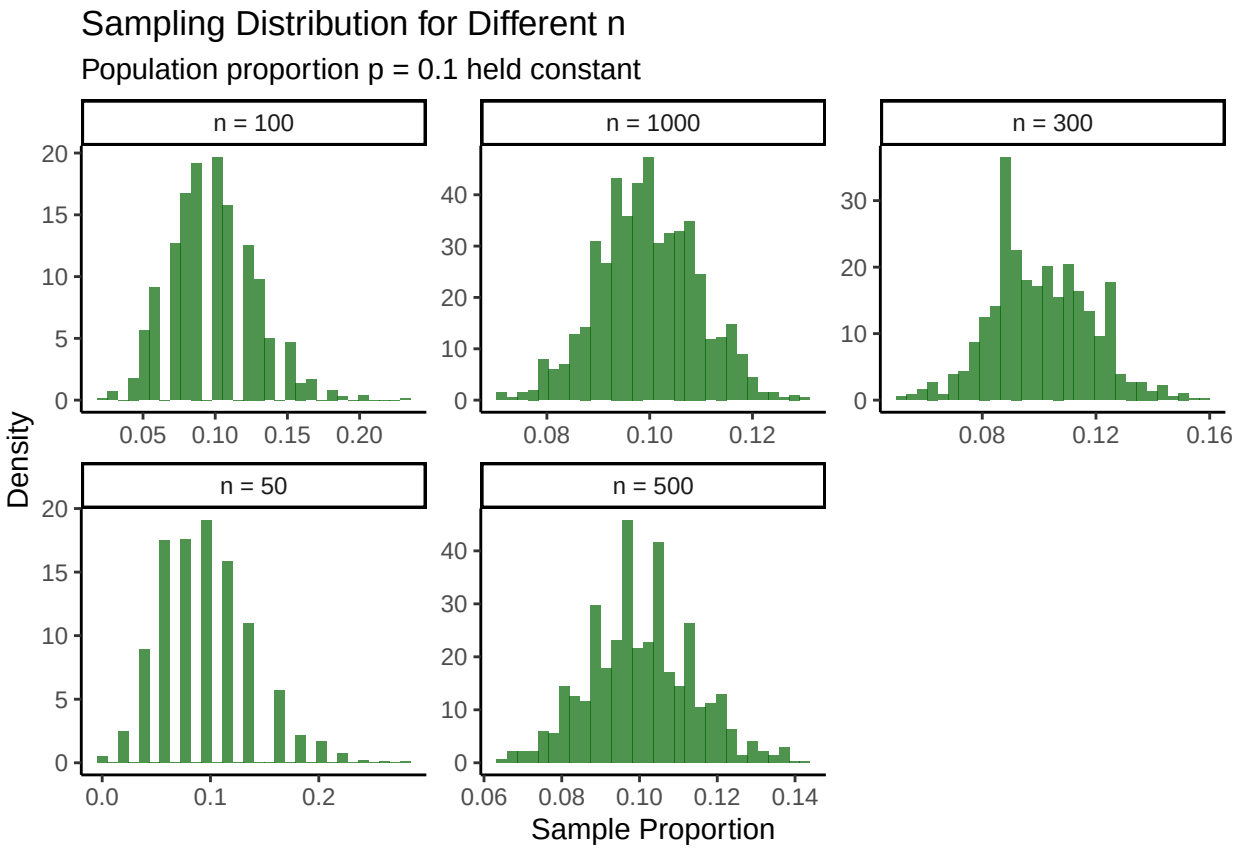
Answer 7

We can see in the output table of statistics summary that the mean is always equal (or very close) to the population proportion. The standard error, as expected, has its maximum value when p = 0.5 and decreases for p closer to 0 or 1. The distribution is still normal since the it passes both success fail tests for normal distribution for all values of p.

Exercise 8

Now also change n. How does n appear to affect the distribution of p^?

```r
p <- 0.1
n_values <- c(50, 100, 300, 500, 1000)
reps <- 1000

# Create comparison plot
comparison_data_n <- data.frame()

for (n in n_values) {
  sample_props <- replicate(reps, {
    sample_data <- rbinom(n, 1, p)
    mean(sample_data)
  })

  temp_data <- data.frame(
    p_hat = sample_props,
    n_value = paste("n =", n)
  )
  comparison_data_n <- rbind(comparison_data_n, temp_data)
}

ggplot(comparison_data_n, aes(x = p_hat)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30, fill = "darkgreen", alpha = 0.7) +
  facet_wrap(~n_value, scales = "free") +
  labs(
    title = "Sampling Distribution for Different n",
    subtitle = paste("Population proportion p =", p, "held constant"),
    x = "Sample Proportion",
    y = "Density"
  ) +
  theme_classic()
```




```
# Calculate statistics for each n
stats_summary_n <- comparison_data_n %>%
  group_by(n_value) %>%
  reframe(
    mean = mean(p_hat),
    sd = sd(p_hat),
    theoretical_se = sqrt(p * (1 - p) / unique(parse_number(n_value))),
    np = n * p_hat,
    np_2 = n * (1-p_hat)
  )

print(stats_summary_n)
```

```
## # A tibble: 5,000 x 6
##    n_value   mean     sd theoretical_se    np  np_2
##    <chr>    <dbl>  <dbl>          <dbl> <dbl> <dbl>
##  1 n = 100 0.0983 0.0297           0.03    80   920
##  2 n = 100 0.0983 0.0297           0.03    20   980
##  3 n = 100 0.0983 0.0297           0.03    70   930
##  4 n = 100 0.0983 0.0297           0.03    70   930
##  5 n = 100 0.0983 0.0297           0.03    90   910
##  6 n = 100 0.0983 0.0297           0.03    90   910
##  7 n = 100 0.0983 0.0297           0.03   110   890
##  8 n = 100 0.0983 0.0297           0.03    80   920
##  9 n = 100 0.0983 0.0297           0.03   180   820
```

```
## 10 n = 100 0.0983 0.0297           0.03   130   870
## # i 4,990 more rows
```

Answer 8

WE can see that the mean continues to be equal or very similar to the population proportion regardless of the sample size. The standard error decreases when n increases, so a larger sample size will have a more concentrated distribution (less variability) as it is obvious by comparing the plots for all different sample sizes.

Exercise 9

Is there convincing evidence that those who sleep 10+ hours per day are more likely to strength train every day of the week? As always, write out the hypotheses for any tests you conduct and outline the status of the conditions for inference. If you find a significant difference, also quantify this difference with a confidence interval.

```
df <- yrbss %>%
  filter(!is.na(school_night_hours_sleep),
         !is.na(strength_training_7d)) %>%
  mutate(
    sleep_grp   = if_else(school_night_hours_sleep == "10+", "10+", "other"),
    train_daily = if_else(strength_training_7d == "7", "yes", "no")
  ) %>%
  mutate(
    sleep_grp   = factor(sleep_grp,   levels = c("other","10+")),
    train_daily = factor(train_daily, levels = c("no","yes"))
  ) %>%
  select(sleep_grp, train_daily)

# sample p
desc <- df %>%
  count(sleep_grp, train_daily) %>%
  group_by(sleep_grp) %>%
  mutate(p_hat = n / sum(n)) %>%
  ungroup()

desc
```

```
## # A tibble: 4 x 4
##   sleep_grp train_daily     n p_hat
##   <fct>     <fct>       <int> <dbl>
## 1 other     no           9949 0.836
## 2 other     yes          1958 0.164
## 3 10+       no            228 0.731
## 4 10+       yes            84 0.269
```

```
p_hats <- df %>%
  group_by(sleep_grp) %>%
  summarize(p_hat = mean(train_daily == "yes"), n = n(), .groups = "drop")

p_hats
```

```
## # A tibble: 2 x 3
##   sleep_grp p_hat     n
##   <fct>     <dbl> <int>
## 1 other     0.164 11907
## 2 10+       0.269   312
```

```
obs_diff <- df %>%
  specify(response = train_daily, explanatory = sleep_grp, success = "yes") %>%
  calculate(stat = "diff in props", order = c("10+","other"))
obs_diff
```

```
## Response: train_daily (factor)
## Explanatory: sleep_grp (factor)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1 0.105
```

```
# hypotheses test
set.seed(1234)
null_dist <- df %>%
  specify(response = train_daily, explanatory = sleep_grp, success = "yes") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("10+","other"))

p_val <- get_p_value(null_dist, obs_stat = obs_diff, direction = "greater")
```

```
## Warning: Please be cautious in reporting a p-value of 0. This result is an approximation
## based on the number of `reps` chosen in the `generate()` step.
## i See `get_p_value()` (`?infer::get_p_value()`) for more information.
```

```
p_val
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

```
# bootstrap CI
set.seed(1234)
boot_dist <- df %>%
  specify(response = train_daily, explanatory = sleep_grp, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "diff in props", order = c("10+","other"))

ci_95 <- get_ci(boot_dist, level = 0.95, type = "percentile")
ci_95
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   0.0579    0.154
```

Answer 9

The goal was to test if students who sleep 10+ hours per night are more likely to strength train every day when compared with students who sleep less. The hypotheses being: H0: p10+ = p_other vs HA: p10+ > pother. The estimated difference in proportions p10+ and pother is 0.105 and the p value is 0, which means that, under the null hypothesis of no relationship, we wouldnt oberseve a difference. So, we can say that there is enough evidence that students who sleep 10+ hours are more likely to strength train every day. A 95% bootstrap CI for the difference in proportions wont include zero, confirming that it is a meaningful difference in the population.

Exercise 10

Let's say there has been no difference in likeliness to strength train every day of the week for those who sleep 10+ hours. What is the probability that you could detect a change (at a significance level of 0.05) simply by chance? Hint: Review the definition of the Type 1 error.

Answer 10

If there is really no difference between the groups, the prob of detecting a change (rejecting the null hypothesis) simply by chance is exactly the significance level = 0.05. This is the definition of a Type 1 error - rejecting a true null hypothesis. The significance level is set to control this probability. In this case, there would be about 5% probability that our sample results would appear significant (randomness).

Exercise 11

Suppose you're hired by the local government to estimate the proportion of residents that attend a religious service on a weekly basis. According to the guidelines, the estimate must have a margin of error no greater than 1% with 95% confidence. You have no idea what to expect for p. How many people would you have to sample to ensure that you are within the guidelines? Hint: Refer to your plot of the relationship between p and margin of error. This question does not require using a dataset.

```
# p = 0.5 since that is max variance
# z score for 95% CI is 1.96
p <- 0.5
z <- 1.96
me <- 0.01

n_needed <- (z^2 * p * (1 - p)) / (me^2)
n_needed
```

```
## [1] 9604
```

Answer 11

I the formula for margin of error is ME = z * sqrt(p(1-p)/n), so we only have to isolate n, giving us n = z^2 * p * (1 - p)) / me^2. I pick p = 0.5 because that is the probability that maximizes the sample size. As we can see by looking at the output, the sample would have to have 9604 or more residents to ensure it is between the guidelines.