# Hypothesis test for Olympics Data Set

```
library(broom)
library(infer)
library(knitr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cowplot)
library(digest)
library(palmerpenguins)
library(testthat)
```

```
##
## Attaching package: 'testthat'
```

```
## The following object is masked from 'package:dplyr':
##
##     matches
```

```
## The following object is masked from 'package:purrr':
##
##     is_null
```

```
## The following objects are masked from 'package:readr':
##
##     edition_get, local_edition
```

```
## The following object is masked from 'package:tidyr':
##
##     matches
```

```
library(tidyverse)
library(infer)
options(repr.matrix.max.rows = 6)
```

Importing the data:

```
olympics <- read.csv("olympics.csv")
```

Removing NA values from the 'age' column, and wrangling the data for the medal column.

```
age_threshold <- 25
olympics_clean <- olympics |>
```

```
    filter(is.na(age) == FALSE,
           #year >= 1984
           ) |>
    select(name, age, event, medal) |>
    mutate(medal = case_when(is.na(medal) ~ "0", TRUE ~ medal),
           medal = case_when(medal != "0" ~ "1", TRUE ~ medal),
           medal = as.double(medal),
           over_under = case_when(age >= age_threshold ~ "over", TRUE ~ "under"))
olympics_clean <- olympics_clean |> select(over_under, medal)
head(olympics_clean)
```

```
##   over_under medal
## 1      under     0
## 2      under     0
## 3      under     0
## 4       over     1
## 5      under     0
## 6      under     0
```

Calculating the observed proportion:

```
olympics_prop <- olympics_clean |>
    group_by(over_under) |>
    summarise(sum = sum(medal),
              count = n(),
              prop = sum(medal) / n())
olympics_prop
```

```
## # A tibble: 2 x 4
##   over_under   sum  count  prop
##   <chr>      <dbl>  <int> <dbl>
## 1 over       21112 130508 0.162
## 2 under      17939 131134 0.137
```

First, we'll define H0 and HA: $H_0$ : the proportion of athletes under 25 that win a medal is equal to the proportion of athletes 25 and older that win a medal. $H_A$ : the proportion of athletes under 25 that win a medal is greater to the proportion of athletes 25 and older that win a medal.

Let's compute the observed test statistic:

```
delta_star <- diff(olympics_prop$prop)
round(delta_star, 5)
```

```
## [1] -0.02497
```

Now let's use simulation to create the null distribution sample:

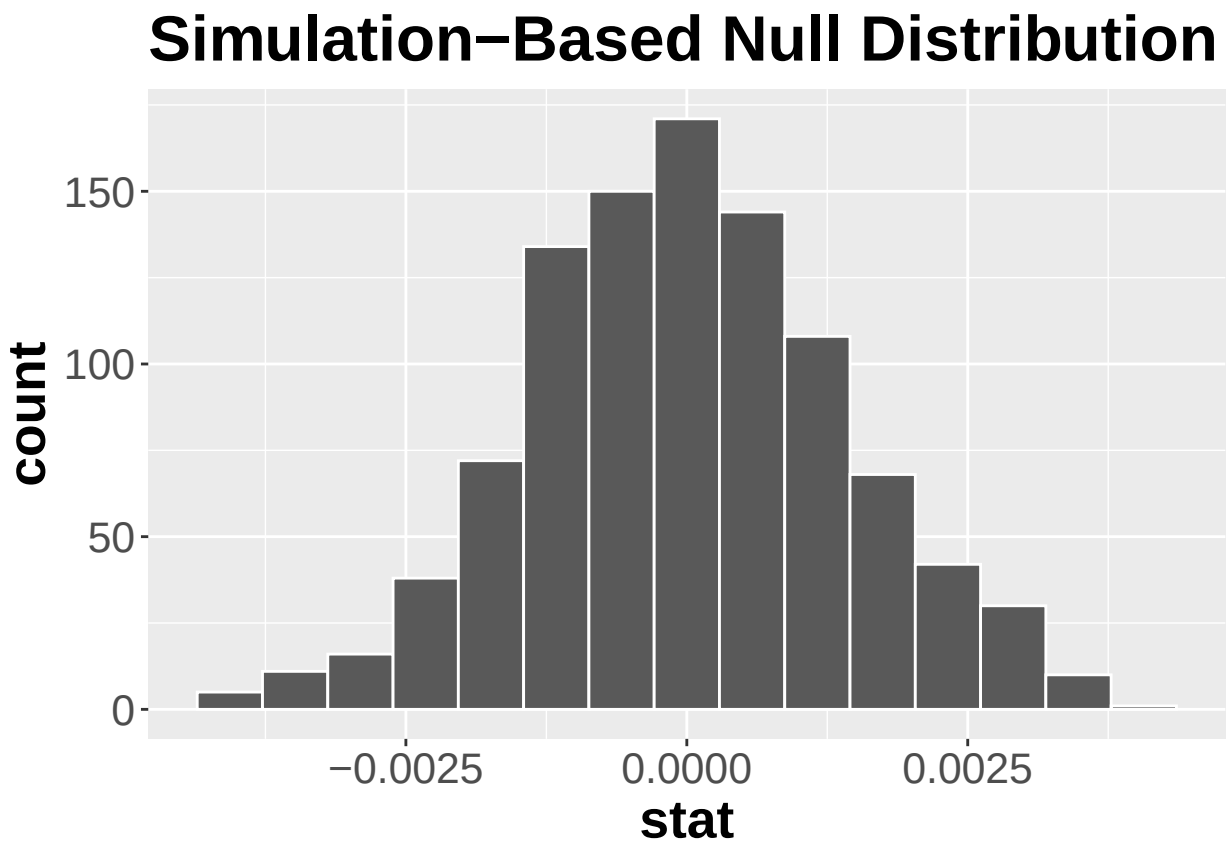
```
olympics_clean <- olympics_clean |>
           mutate(medal = as.factor(medal),
           over_under = as.factor(over_under))

set.seed(1234) # For reproducibility.
null_distribution_olympics <- olympics_clean |>
  specify(formula = medal ~ over_under, success = "1") |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in props", order = c("under", "over"))
head(null_distribution_olympics)
```

```
## Response: medal (factor)
## Explanatory: over_under (factor)
## Null Hypothesis: independence
## # A tibble: 6 x 2
##   replicate      stat
##       <int>     <dbl>
## 1         1 -0.00141
## 2         2  0.000302
## 3         3 -0.000584
## 4         4  0.00137
## 5         5 -0.000217
## 6         6 -0.00139
```

Let's visualize the null distribution now:

```
options(repr.plot.width = 10, repr.plot.height = 10)
h0_dist <- null_distribution_olympics %>%
    visualize() +
    theme(text = element_text(size=20)) +
  theme(
    text = element_text(size = 20),
    plot.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold"),
    legend.title = element_text(face = "bold"),
  )
h0_dist
```

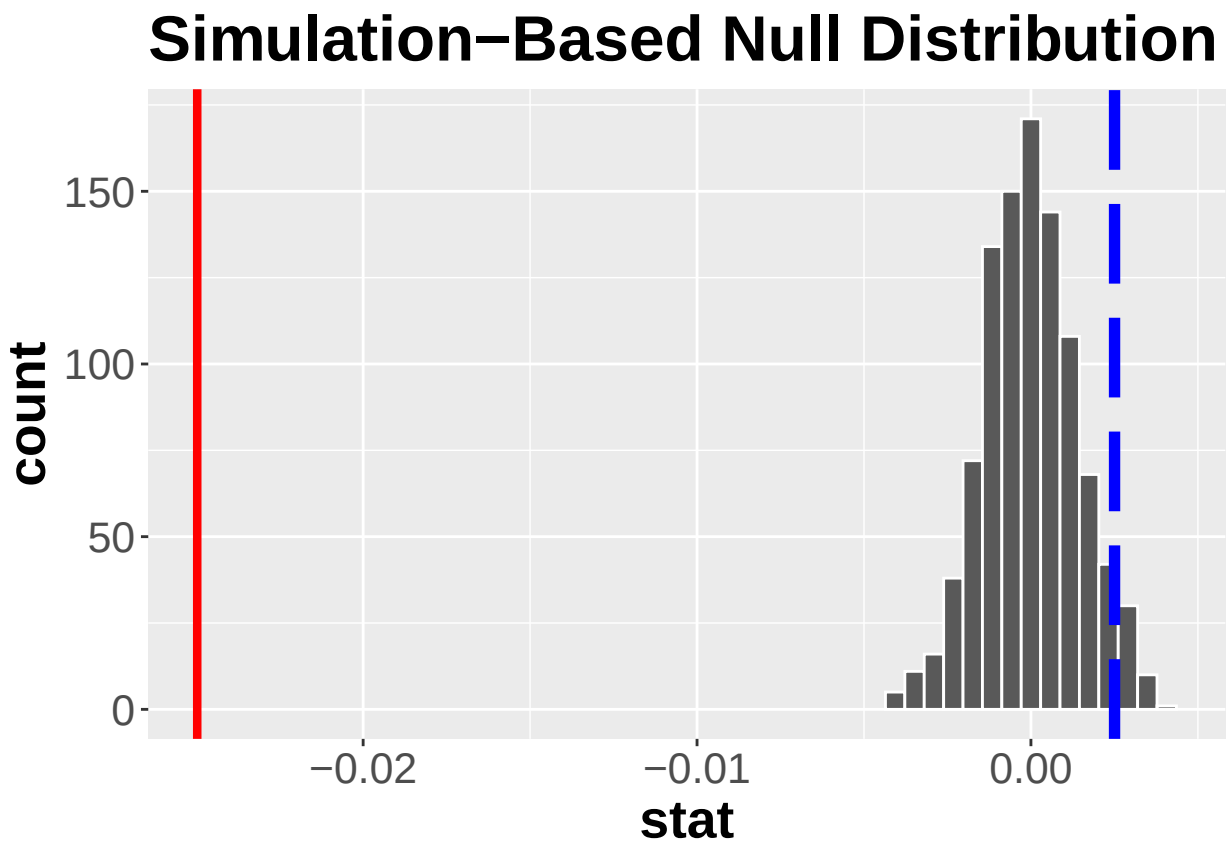



Check where the observed statistic falls:

```
alpha_threshold <- quantile(null_distribution_olympics$stat, 0.95)

h0_dist <- h0_dist + geom_vline(
  xintercept = alpha_threshold,
  color = "blue", lty = 5, size = 2) +
  geom_vline(xintercept = delta_star, color = "red", size = 1.5)

h0_dist
```




Let's calculate our p-value:

```
null_distribution_olympics %>%
  get_pvalue(obs_stat = delta_star, direction = "greater")
```

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