If you search for the phrase “fastest man on earth” in Google, chances are that it will return the answer “Usain Bolt”. It certainly does so for me, even though the results might be different if Google decides to personalize the results for you. This is because currently he holds the world record for being the quickest (9.58s) to run a 100 metres race. Justin Gatlin once did it in 9.45s, but that does not count as the record because he was assisted by the wind.

However, as statisticians, we know that a single data point does not necessarily prove anything and we should look at the distribution of timings for the top runners to check if Usain Bolt is truly the fastest man on earth. In this article, we will try to find an answer to this question using data.

**Data**

Our first step is to find data on the 100m race timings for all top runners. Unfortunately, I could not find any public source of this data. The best option we have is to use data from the excellent website [Alltime Athletics](http://www.alltime-athletics.com/) maintained. This provides the details of the all-time men’s best 100m race timings. We can answer a slightly different question using this data. Out of all runners who appear in this list, is there any evidence to suggest that there is a statistical difference among the top runners?

Our first step will be to extract and store the data in a R data frame. We use the *rvest* and *readr* R packages to do this.

library(rvest)

library(readr)

male\_100\_html <- read\_html("http://www.alltime-athletics.com/m\_100ok.htm")

An examination of the HTML source reveals that the data has been stored within HTML pre-tags. There are also seven tables available as part of the page, out of which we are interested in the table “All-time men’s best 100m”. We use the function *html\_nodes* from the *rvest* package using an XPath selector. The next step is to extract the HTML text using the function *html\_text*, and restrict to the first element.

male\_100\_pres <- male\_100\_html %>%

html\_nodes(xpath = "//pre")

male\_100\_htext <- male\_100\_pres %>%

html\_text()

male\_100\_htext <- male\_100\_htext[[1]]

The data is stored in fixed-width format in the variable *male\_100\_htext*. The *read\_fwf* function from the *readr* package can be used to store the data in a data frame. The exact arguments to the function were decided using a bit of trial and error.

male\_100 <- readr::read\_fwf(male\_100\_htext, skip = 1, n\_max = 3178,

col\_types = cols(.default = col\_character()),

col\_positions = fwf\_positions(

c(1, 16, 27, 35, 66, 74, 86, 93, 123),

c(15, 26, 34, 65, 73, 85, 92, 122, 132)

))

str(male\_100)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 3178 obs. of 9 variables:

## $ X1: chr "1" "2" "3" "3" ...

## $ X2: chr "9.58" "9.63" "9.69" "9.69" ...

## $ X3: chr "+0.9" "+1.5" "±0.0" "+2.0" ...

## $ X4: chr "Usain Bolt" "Usain Bolt" "Usain Bolt" "Tyson Gay" ...

## $ X5: chr "JAM" "JAM" "JAM" "USA" ...

## $ X6: chr "21.08.86" "21.08.86" "21.08.86" "09.08.82" ...

## $ X7: chr "1" "1" "1" "1" ...

## $ X8: chr "Berlin" "London" "Beijing" "Shanghai" ...

## $ X9: chr "16.08.2009" "05.08.2012" "16.08.200" "20.09.2009" ...

## - attr(\*, "spec")=

## .. cols(

## .. .default = col\_character(),

## .. X1 = col\_character(),

## .. X2 = col\_character(),

## .. X3 = col\_character(),

## .. X4 = col\_character(),

## .. X5 = col\_character(),

## .. X6 = col\_character(),

## .. X7 = col\_character(),

## .. X8 = col\_character(),

## .. X9 = col\_character()

## .. )

For this analysis, we are only interested in the name of the runner and the race timing. So we restrict the data to these two columns, and give each column an appropriate name.

library(dplyr)

male\_100 <- male\_100 %>%

select(X2, X4)

colnames(male\_100) <- c("timing", "runner")

We observe that some of the timings have an “A” against the time. The Wikipedia page [Athletics abbreviations](https://en.wikipedia.org/wiki/Athletics_abbreviations) informs us that this indicates a mark set at altitude. We simply remove the “A” and also convert the variable *timing* into a numeric variable.

male\_100 <- male\_100 %>%

mutate(timing = gsub("A", "", timing),

timing = as.numeric(timing))

male\_100

## # A tibble: 3,178 x 2

## timing runner

##

## 1 9.58 Usain Bolt

## 2 9.63 Usain Bolt

## 3 9.69 Usain Bolt

## 4 9.69 Tyson Gay

## 5 9.69 Yohan Blake

## 6 9.71 Tyson Gay

## 7 9.72 Usain Bolt

## 8 9.72 Asafa Powell

## 9 9.74 Asafa Powell

## 10 9.74 Justin Gatlin

## # ... with 3,168 more rows

**Exploration**

As with any statistical analysis, the first step is to explore and understand the data we are working with. We see that there are 318 unique runners in the data.

n\_distinct(male\_100$runner)

## [1] 328

runner\_cnt <- male\_100 %>%

group\_by(runner) %>%

summarise(n\_rec = n()) %>%

arrange(desc(n\_rec))

runner\_cnt

## # A tibble: 328 x 2

## runner n\_rec

##

## 1 Asafa Powell 144

## 2 Mike Rodgers 101

## 3 Justin Gatlin 98

## 4 Maurice Greene 83

## 5 Nesta Carter 73

## 6 Usain Bolt 71

## 7 Kim Collins 68

## 8 Tyson Gay 67

## 9 Ato Boldon 65

## 10 Frank Fredericks 62

## # ... with 318 more rows

As expected, some of the well-known names dominate the list with the maximum number of records. Our next step is to look at the distribution of the number of records.

library(ggplot2)

ggplot(runner\_cnt, aes(n\_rec)) +

geom\_histogram(binwidth = 5, fill = "lightblue", colour = "black") +

xlab(NULL) +

ylab(NULL) +

ggtitle("Distribution of number of records by runner") +

theme\_bw() +

theme(

panel.border = element\_blank(),

plot.title = element\_text(hjust = 0.5, vjust = 0.5)

)

As expected, this distribution is heavily skewed as the top runners will tend to dominate the list. We first restrict our data to those who have at least 50 records each.

male\_100\_2 <- male\_100 %>%

inner\_join(

select(filter(runner\_cnt, n\_rec >= 50), runner),

by = c("runner" = "runner")

)

There are 16 runners left in this data. Let us have a look at the distribution of their race timings as well as the mean and median race timings for these runners.

ggplot(male\_100\_2, aes(timing)) +

geom\_density() +

facet\_wrap(~runner, nrow = 4, ncol = 4) +

xlab(NULL) +

ylab(NULL) +

ggtitle("Distribution of race timings by runner") +

theme\_bw() +

theme(

panel.border = element\_blank(),

plot.title = element\_text(hjust = 0.5, vjust = 0.5)

)

male\_100\_means <- male\_100\_2 %>%

group\_by(runner) %>%

summarise(mean\_timing = mean(timing)) %>%

arrange(mean\_timing)

male\_100\_means

## # A tibble: 16 x 2

## runner mean\_timing

##

## 1 Usain Bolt 9.90

## 2 Asafa Powell 9.94

## 3 Tyson Gay 9.95

## 4 Yohan Blake 9.96

## 5 Justin Gatlin 9.96

## 6 Maurice Greene 9.97

## 7 Ato Boldon 10.00

## 8 Jimmy Vicaut 10.0

## 9 Frank Fredericks 10.0

## 10 Mike Rodgers 10.0

## 11 Nesta Carter 10.0

## 12 Carl Lewis 10.0

## 13 Linford Christie 10.0

## 14 Dennis Mitchell 10.0

## 15 Kim Collins 10.0

## 16 Michael Frater 10.0

male\_100\_medians <- male\_100\_2 %>%

group\_by(runner) %>%

summarise(median\_timing = median(timing)) %>%

arrange(median\_timing)

male\_100\_medians

## # A tibble: 16 x 2

## runner median\_timing

##

## 1 Usain Bolt 9.9

## 2 Asafa Powell 9.95

## 3 Yohan Blake 9.96

## 4 Justin Gatlin 9.97

## 5 Maurice Greene 9.97

## 6 Tyson Gay 9.97

## 7 Ato Boldon 10

## 8 Mike Rodgers 10

## 9 Frank Fredericks 10.0

## 10 Jimmy Vicaut 10.0

## 11 Nesta Carter 10.0

## 12 Linford Christie 10.0

## 13 Dennis Mitchell 10.0

## 14 Carl Lewis 10.0

## 15 Kim Collins 10.0

## 16 Michael Frater 10.0

There are 6 runners who are at the top using both the mean or the median. Also, for all of them, the mean and the median timings are less than 10 seconds. So we will further restrict the data to these 6 runners, and see if their race timings are significantly different.

male\_100\_3 <- male\_100\_2 %>%

filter(runner %in% c("Usain Bolt", "Asafa Powell", "Yohan Blake",

"Justin Gatlin", "Maurice Greene", "Tyson Gay"))

**Analysis**

Now that we have the data ready, our next step is to try and answer the original question. We will follow two different approaches. Our first approach will be the non-parametric Kruskall-Wallis rank sum test. We will also do a multiple comparison test for the Kruskall-Wallis. The next approach will be a pairwise comparison of the runners using simulations. Another article which implements the same idea, and provides code using the *infer* package mentioned below:

This heavily commented code will load and clean and rearrange the WWBI data into something we can analyze and plot.

# Load libraries

library(tidyverse)

library(ggridges)

library(scales)

library(infer)

set.seed(1234) # Make all random draws reproducible

# https://datacatalog.worldbank.org/dataset/worldwide-bureaucracy-indicators

wwbi <- read\_csv("WWBIData.csv")

# Create a list of indicators we want to work with

indicators <- c(

"BI.PWK.PRVS.FE.ZS", # females as share of private paid employees

"BI.PWK.PUBS.FE.ZS" # females as share of public paid employees

)

# Make a small, cleaner subset of the WWBI data

wwbi\_small <- wwbi %>%

# Only select the columns we care about

select(country = `Country Name`, country\_code = `Country Code`,

indicator = `Indicator Code`, starts\_with("20")) %>%

# Keep only the indicators we care about

filter(indicator %in% indicators) %>%

# Gather all the year-based columns into two long columns

gather(year, value, starts\_with("20")) %>%

# Spread the data back out so that there are columns for each indicator

spread(indicator, value) %>%

# Make these indicator names human readable

rename(share\_female\_private = `BI.PWK.PRVS.FE.ZS`,

share\_female\_public = `BI.PWK.PUBS.FE.ZS`) %>%

# Amid all the gathering and spreading, every column has become a character.

# This converts the year and all the share\_\* variables back to numbers

mutate\_at(vars(year, starts\_with("share")), as.numeric)

wwbi\_2012 <- wwbi\_small %>%

filter(year == 2012) %>%

# Get rid of rows that are missing data in the share\_\* columns

drop\_na(starts\_with("share")) %>%

# Make this tidy and long, with a column for private or public

gather(sector, proportion, starts\_with("share")) %>%

# Make these values even nicer

mutate(sector = recode(sector,

share\_female\_private = "Women in private sector",

share\_female\_public = "Women in public sector"))

First, we’ll use a ridge plot (or overlapping density plots) to see if there’s a difference in the distribution of employment across these two sectors. We include quantile\_lines = TRUE and quantiles = 2 to draw the median of each distribution.

ggplot(wwbi\_2012, aes(x = proportion, y = sector, fill = sector)) +

stat\_density\_ridges(quantile\_lines = TRUE, quantiles = 2, scale = 3, color = "white") +

scale\_fill\_manual(values = c("#98823c", "#9a5ea1"), guide = FALSE) +

scale\_x\_continuous(labels = percent\_format(accuracy = 1)) +

labs(x = "Percent of women employed in the sector", y = NULL) +

theme\_minimal() +

theme(panel.grid.minor = element\_blank())

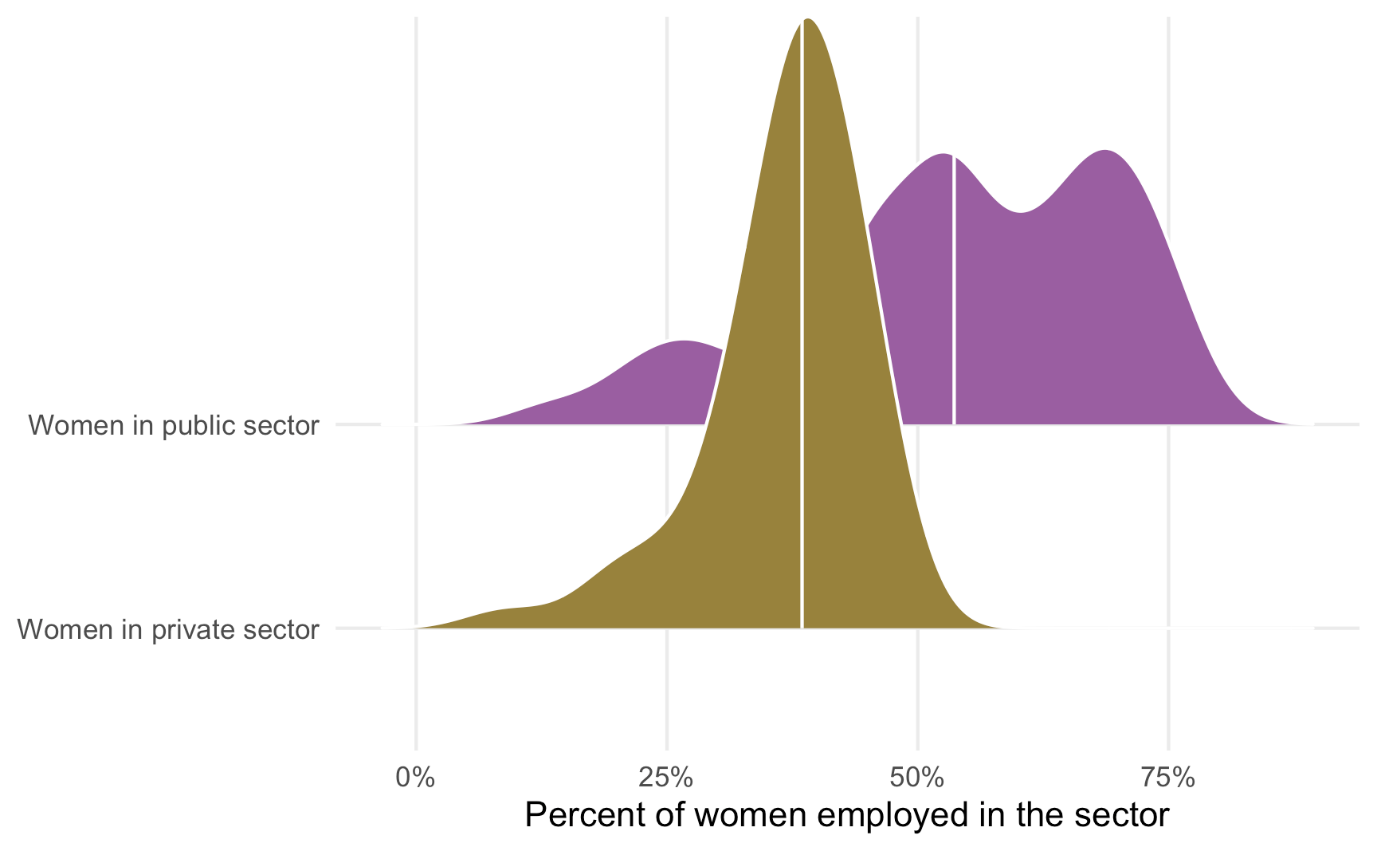


Figure 1: Distribution of female employment in public and private sectors

It looks like there’s a difference here—the median percent of women in the public sector looks like it’s between 40–50%, while the median percent of women in the private sector is 50ish%.

We can use the infer package to determine the exact difference in the medians of these distributions. We use a formula in specify() to indicate which variables we want to be our response (or y) and which we want to be our explanatory (or x) variables, and then we calculate() the difference in median values. The order argument tells infer to subtract the public values from the private values.

diff\_prop <- wwbi\_2012 %>%

specify(proportion ~ sector) %>%

calculate("diff in medians",

order = c("Women in private sector", "Women in public sector"))

diff\_prop

## # A tibble: 1 x 1

## stat

## <dbl>

## 1 -0.152

The difference in the median proportions here is 15.2%. More women appear to be employed in the public sector. But because of sampling issues, is there a possibility that this 15% difference could potentially be zero? Could it be positive instead of negative?

To test this, we simulate a world where the actual difference in medians between these two sectors is zero. We then plot that null distribution, place the observed 15.2% difference in it, and see how well it fits. Here we follow the same specify() %>% calculate() pattern, but introduce two new steps. We first hypothesize() that the two sectors are independent of each other and that there’s no difference between the two. We then generate() a null distribution based on permutation (essentially shuffling the private/public labels within the existing data 5,000 times), and then calculate the difference in medians for the two groups.

pub\_private\_null <- wwbi\_2012 %>%

specify(proportion ~ sector) %>%

hypothesize(null = "independence") %>%

generate(reps = 5000, type = "permute") %>%

calculate("diff in medians",

order = c("Women in private sector", "Women in public sector"))

# Conveniently, the visualize() function that comes with infer returns a ggplot

# object, so we can continue to add ggplot layers to enhance and clean the plot

pub\_private\_null %>%

visualize() +

geom\_vline(xintercept = diff\_prop$stat, color = "#FF4136", size = 1) +

labs(x = "Difference in median proportion\n(Women in private sector − women in public sector)",

y = "Count",

subtitle = "Red line shows observed difference in median proportions") +

scale\_x\_continuous(labels = percent\_format(accuracy = 1)) +

theme\_minimal() +

theme(panel.grid.minor = element\_blank())

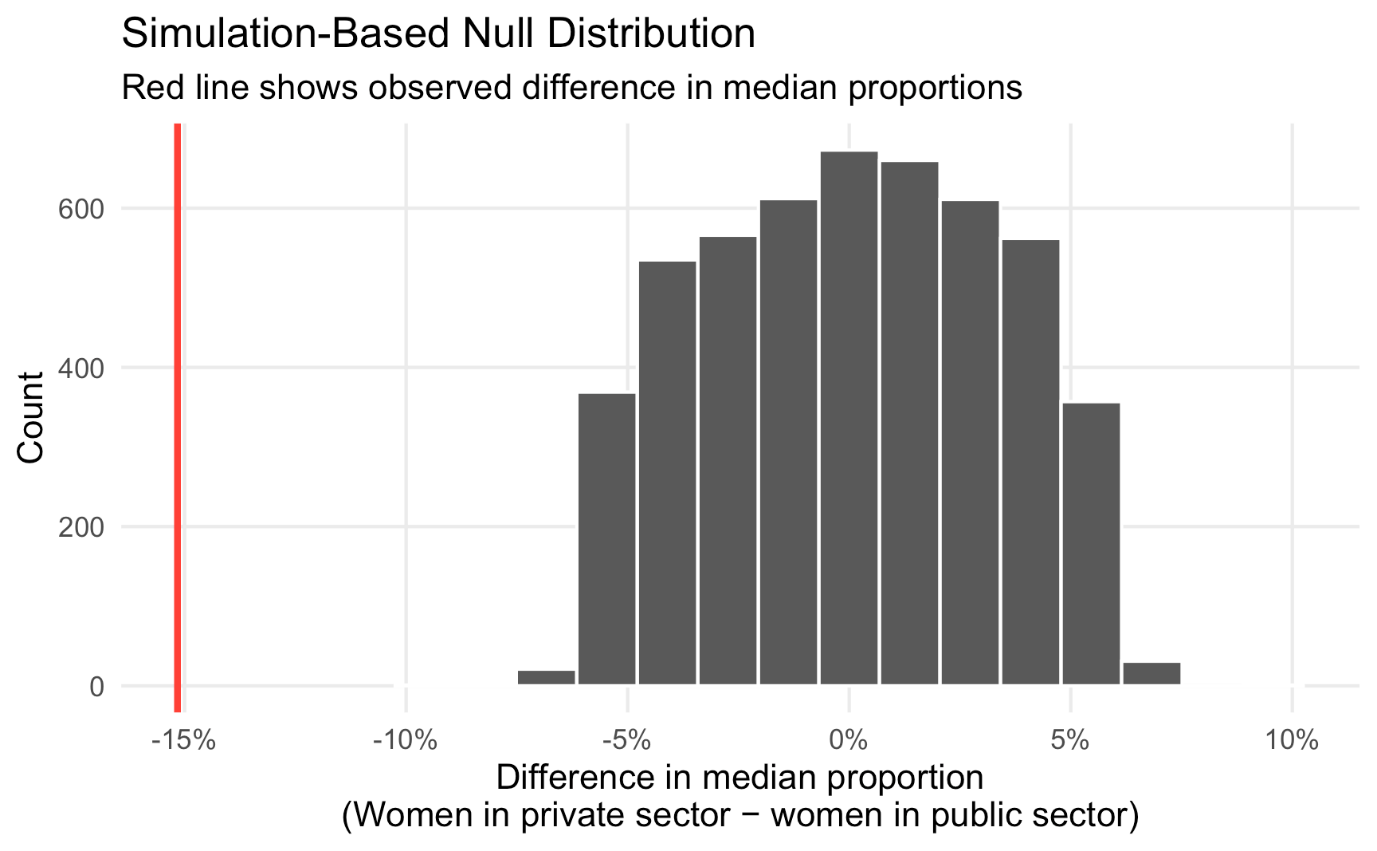


Figure 2: Observed difference in a hypothetical null world

This red line is pretty far in the left tail of the distribution and seems atypical. We can calculate the probability of seeing a difference as big as 15.2% with the get\_pvalue() function. Because we care about differences in means or medians (and because the difference could be positive if we flipped the order of subtracting the two groups—public−private instead of private−public), we specify direction = "both" to get a two-tailed p-value.

pub\_private\_null %>%

get\_pvalue(obs\_stat = diff\_prop, direction = "both")

## 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. See `?get\_p\_value()` for

## more information.

## # A tibble: 1 x 1

## p\_value

## <dbl>

## 1 0

The p-value is basically 0 (it’s not actually 0—it’s something like 0.0000001—but because it is based on simulations, it’ll report 0 and R will show a warning), which means that there’s a 0% chance of seeing a difference at least as large as 15.2% in a world where there’s no difference. That’s pretty strong evidence, and I’d feel confident declaring that there’s a statistically significant difference between sectoral employment patterns for women, with more women working in public sector jobs than in the private sector.

And that’s it! No flow charts. No complex assumption checking. Using the infer package in R, we brute forced-ly simulated our way to a p-value that is actually interpretable, and now we know that women tend to work in the public sector more than in the private sector. Magic!

kt <- kruskal.test(timing ~ runner, data = male\_100\_3)

kt

##

## Kruskal-Wallis rank sum test

##

## data: timing by runner

## Kruskal-Wallis chi-squared = 19.202, df = 5, p-value = 0.001762

Since the p-value is significant at the 99% level, it indicates that at least one of the runners is different from the others.

library(pgirmess)

ktph <- kruskalmc(male\_100\_3$timing, male\_100\_3$runner)

ktph$dif.com

## obs.dif critical.dif difference

## Asafa Powell-Justin Gatlin 32.046910 57.86481 FALSE

## Asafa Powell-Maurice Greene 49.073712 60.89670 FALSE

## Asafa Powell-Tyson Gay 24.278711 65.34670 FALSE

## Asafa Powell-Usain Bolt 45.633314 64.07814 FALSE

## Asafa Powell-Yohan Blake 28.887692 68.71975 FALSE

## Justin Gatlin-Maurice Greene 17.026801 65.91562 FALSE

## Justin Gatlin-Tyson Gay 7.768200 70.04750 FALSE

## Justin Gatlin-Usain Bolt 77.680224 68.86559 TRUE

## Justin Gatlin-Yohan Blake 3.159219 73.20426 FALSE

## Maurice Greene-Tyson Gay 24.795001 72.57220 FALSE

## Maurice Greene-Usain Bolt 94.707025 71.43207 TRUE

## Maurice Greene-Yohan Blake 20.186020 75.62365 FALSE

## Tyson Gay-Usain Bolt 69.912024 75.26171 FALSE

## Tyson Gay-Yohan Blake 4.608981 79.25099 FALSE

## Usain Bolt-Yohan Blake 74.521005 78.20829 FALSE

The only two cases where there seems to be a difference are – “Justin Gatlin-Usain Bolt” and “Maurice Green-Usain Bolt”.

We now do the pairwise comparison of athletes using simulations. To do that, we will write a function to compare two individual runners. The first step will be to calculate the difference in medians in the observed data. This is accomplished using the *specify* and *calculate* functions in the *infer* package. *specify* is used to specify the response and explanatory variables. *calculate* is used to calculate the test statistic, which is the difference in medians in this case. The second step is similar, except that we include a null hypothesis that there is no difference in the median timings between the runners. We also perform 10,000 simulations – where each simulation permutes the runners in the existing data, and then calculate the difference in medians between the two groups. The *infer* package also provides a *visualize* function to visualize the distribution of the simulated test statistic.

library(infer)

set.seed(2154)

compare\_runners <- function(runner1, runner2, visualize = FALSE) {

diff\_med <- male\_100\_3 %>%

filter(runner %in% c(runner1, runner2)) %>%

arrange(runner) %>%

specify(timing ~ runner) %>%

calculate("diff in medians",

order = c(runner1, runner2))

diff\_med\_null <- male\_100\_3 %>%

filter(runner %in% c(runner1, runner2)) %>%

arrange(runner) %>%

specify(timing ~ runner) %>%

hypothesize(null = "independence") %>%

generate(reps = 10000, type = "permute") %>%

calculate("diff in medians",

order = c(runner1, runner2))

if (visualize) {

diff\_med\_plot <- diff\_med\_null %>%

visualize(method = "simulation") +

shade\_p\_value(obs\_stat = diff\_med, direction = "less") +

xlab("Difference in median run timings") +

ylab(NULL) +

ggtitle(paste0("Comparing ", runner1, " and ", runner2)) +

scale\_x\_continuous(labels = scales::number\_format(accuracy = 0.01)) +

theme\_bw() +

theme(

panel.border = element\_blank(),

plot.title = element\_text(hjust = 0.5, vjust = 0.5)

)

print(diff\_med\_plot)

}

diff\_med\_null %>%

get\_pvalue(obs\_stat = diff\_med, direction = "less")

}

The function is applied to each pair of runners. Rather than looking at the results for all the 15 pairs of runners, we do it based on their rankings by median times. There is only one exception, where we also look at the pair Usain Bolt-Yohan Blake, who have the 1st and 3rd least median times respectively. The results for each pair, as well as the p-value generated by the simulation, are shown in the graphs and tables below.

compare\_runners("Usain Bolt", "Asafa Powell", TRUE)

## # A tibble: 1 x 1

## p\_value

##

## 1 0.0185

compare\_runners("Asafa Powell", "Yohan Blake", TRUE)

## # A tibble: 1 x 1

## p\_value

##

## 1 0.200

compare\_runners("Yohan Blake", "Justin Gatlin", TRUE)

## # A tibble: 1 x 1

## p\_value

##

## 1 0.473

compare\_runners("Justin Gatlin", "Maurice Greene", TRUE)

## # A tibble: 1 x 1

## p\_value

##

## 1 0.715

compare\_runners("Maurice Greene", "Tyson Gay", TRUE)

## # A tibble: 1 x 1

## p\_value

##

## 1 0.702

compare\_runners("Usain Bolt", "Yohan Blake", TRUE)

## # A tibble: 1 x 1

## p\_value

##

## 1 0.0027

There is only a 1.85% chance of seeing a difference as large as the observed difference if there is actually no difference between the timings of Usain Bolt and Asafa Powell. The chance for the pair of Usain Bolt and Yohan Blake is even lower at 0.27%. For all the remaining pairs, we fail to reject the null hypothesis and do not find sufficient evidence to conclude that the timings for these pairs are statistically different. It appears that Usain Bolt is better than the next best in the list which is Asafa Powell and Yohan Blake. Given the data we have, we do have reason to believe that Usain Bolt is the fastest man on earth (as measured by 100m race timings).