Data 606 Final Project: Pokemon Card Pulls from Set Sword and Shield - Silver Tempest

Melissa Bowman

Abstract

Pokémon cards are a highly collectable card game that was first created back in the late 90s. Since then, Pokémon cards have been sought after for collectable purposes by enthusiasts. Due to their popularity, people have attempted to determine the best methods of how to secure the rarest cards in each Pokémon set that is released every several months. In this project, I will be looking at the pull percentages from the most recent Pokémon set titled "Sword and Shield - Silver Tempest". Pull percentages refer to the percentage of various rarities that can be found in a pack of cards. See Figure 4 for a list of all card rarities in this set. The probability rates for this project came from sampling a standard Pokémon booster box (Figure 1) which consists of 36 booster packs (Figure 2) with each booster pack containing 10 cards, 360 cards total. 6 of the booster packs were placed aside for confirmation of the probability. This set consists of 245 different cards that can be pulled.

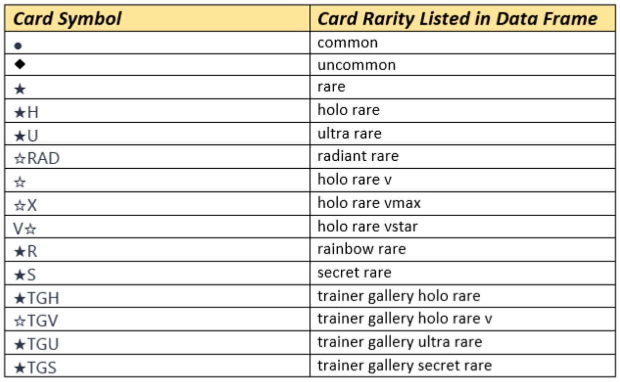
The data frame was constructed by designating each booster pack as a pack number, then followed with each cards unique ID which can be found at the bottom left corner of a Pokémon card, and finally the card rarity was labeled by using the player's guide booklet (Figure 3) which details the card listings. In addition to the booster box probability pulls, I also added an additional 30 booster pack, additional 300 card sample, to determine whether the probabilities lined up with the booster box pull percentages.

Figure 1: Booster Box

Figure 2: Booster Packs

Figure 3: Player's Guide Booklet

Figure 4: Player's Guide Booklet Card Rarity List



Card Symbol	Card Rarity Listed in Data Frame
●	common
◆	uncommon
★	rare
★H	holo rare
★U	ultra rare
☆RAD	radiant rare
☆	holo rare v
☆X	holo rare vmax
V☆	holo rare vstar
★R	rainbow rare
★S	secret rare
★TGH	trainer gallery holo rare
☆TGV	trainer gallery holo rare v
★TGU	trainer gallery ultra rare
★TGS	trainer gallery secret rare

Figure 5: Player's Guide Booklet Card Description

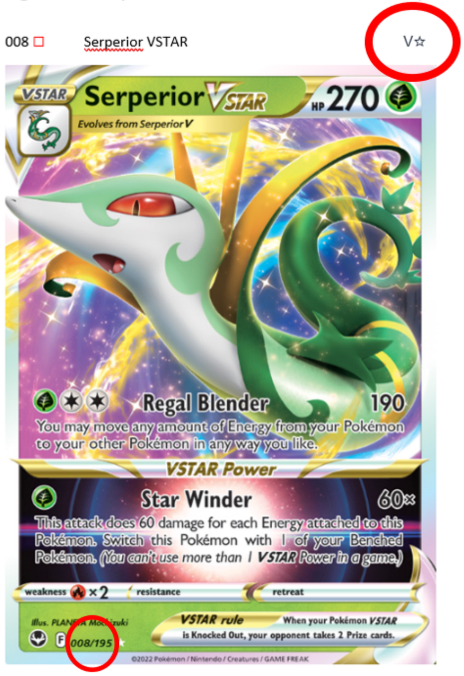
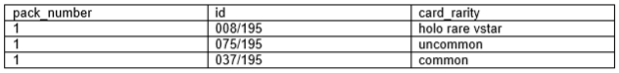

008 □
Serperior VSTAR
V☆
VSTAR
Serperior VSTAR
HP 270
Evolves from Serperior V
Regal Blender 190
You may move any amount of Energy from your Pokémon to your other Pokémon in any way you like.
VSTAR Power
Star Winder 60×
This attack does 60 damage for each Energy attached to this Pokémon. Switch this Pokémon with 1 of your Benched Pokémon. (You can't use more than 1 VSTAR Power in a game.)
weakness ×2
resistance
retreat
Illus. PLANETA Mochizuki
F 008/195
VSTAR rule When your Pokémon VSTAR is Knocked Out, your opponent takes 2 Prize cards.
©2022 Pokémon / Nintendo / Creatures / GAME FREAK

Figure 6: Data Frame Created



pack_number	id	card_rarity
1	008/195	holo rare vstar
1	075/195	uncommon
1	037/195	common

Overview

Research question

There are two research questions that generate the following null hypothesizes:

1. Is the probabilities of the 300 sample cards in the booster box equal to the 60-card sample from the booster box set aside?

   Null Hypothesis: 300 sample size probabilities is equal to 60 sample size probabilities

   Alternative Hypothesis: 300 sample size probabilities not equal to 60 sample size probabilities

2. Are the booster box pull percentages greater than the percentages of a booster pack?

   Null Hypothesis: Booster box probabilities greater than booster pack probabilities

   Alternative Hypothesis: Booster box probabilities less than booster pack probabilities

Context on the data collection

The data was collected by placing in the pack number, the card id, and the cards rarity into 3 different csv files. The first csv file has 300 cards sampled from the booster box. The second has 60 cards from the booster box to confirm the 300 cards sampled. The final csv file has 300 cards sampled from 30 different booster packs.

Description of the dependent variable and the independent variables

The dependent variable is the amount of a given card rarity pulled in the experiment and the independent variables are the card ID and the card rarity type.

Load libraries.

```
library(dplyr)
library(tidyverse)
library(infer)
set.seed(242424)
```

Data Frames for Pulls

```
df_box = read.csv('https://raw.githubusercontent.com/melbow2424/Data-606-Final-Project/main/sword_shiel

df_confirm = read.csv('https://raw.githubusercontent.com/melbow2424/Data-606-Final-Project/main/sword_s

df_packs = read.csv('https://raw.githubusercontent.com/melbow2424/Data-606-Final-Project/main/sword_shi

# Showing data frame of booster box
head(df_box, 11)
```

```
##    pack_number      id card_rarity
## 1            1 080/195    uncommon
## 2            1 155/195    uncommon
## 3            1 075/195    uncommon
## 4            1 037/195      common
## 5            1 106/195      common
## 6            1 054/195      common
## 7            1 072/195      common
## 8            1 013/195      common
## 9            1 047/195      common
## 10           1 126/195        rare
## 11           2 138/195 holo rare V
```

Tidying of data frames before statistical analysis.

```
df_box <- df_box %>%
  #lowercase of card rarity
  mutate(card_rarity = str_to_lower(card_rarity)) %>%
  #remove all trailing whitespace of card rarity
  mutate(card_rarity = str_trim(card_rarity ,"both")) %>%
  #remove all whitespace in id
  mutate(id = str_remove_all(id," "))

df_confirm <- df_confirm %>%
  #lowercase of card rarity
  mutate(card_rarity = str_to_lower(card_rarity)) %>%
  #remove all trailing whitespace of card rarity
  mutate(card_rarity = str_trim(card_rarity ,"both")) %>%
  #replace holo v rare with holo rare v
  mutate(card_rarity = str_replace_all(card_rarity ,"holo v rare", "holo rare v")) %>%
  #remove all whitespace in id
  mutate(id = str_remove_all(id," "))

df_packs <- df_packs %>%
  #lowercase of card rarity
  mutate(card_rarity = str_to_lower(card_rarity)) %>%
  #remove all trailing whitespace of card rarity
  mutate(card_rarity = str_trim(card_rarity ,"both")) %>%
  #remove all whitespace in id
  mutate(id = str_remove_all(id," "))

#Removed a column from the data frame of the booster packs
df_packs <- subset(df_packs, select = -c(X))
```

Summary statistics: Research question 1

Research question 1: Is the probabilities of the 300 sample cards in the booster box is equal to the 60-card sample from the booster box set aside?

Here we try to answer if the following point estimate percentages of the 300 sample cards in the booster box is equal to the 60-card sample from the booster box. Because this is a sample size and not a true population size, we designate the percentage as $\hat{p}$.

```
df_box_merge<- df_box %>%
  count(card_rarity) %>%
  mutate(p_hat = n /sum(n))

df_confirm_merge<- df_confirm %>%
  count(card_rarity) %>%
  mutate(p_hat = n /sum(n))

df_merge <- merge(df_box_merge, df_confirm_merge, by = "card_rarity", all.x = TRUE, all.y = TRUE)
colnames(df_merge) <- c('card_rarity','n_box','p_hat_box','n_confirm','p_hat_confirm')
print(df_merge)
```

```
##                    card_rarity n_box   p_hat_box n_confirm p_hat_confirm
## 1                       common   159 0.530000000        32    0.53333333
## 2                    holo rare     6 0.020000000        NA            NA
## 3                  holo rare v     4 0.013333333         1    0.01666667
## 4              holo rare vstar     1 0.003333333         1    0.01666667
## 5                 radiant rare     2 0.006666667        NA            NA
## 6                         rare    25 0.083333333         4    0.06666667
## 7                  secret rare    NA          NA         1    0.01666667
## 8    trainer gallery holo rare     3 0.010000000        NA            NA
## 9  trainer gallery holo rare v     1 0.003333333        NA            NA
## 10                  ultra rare     1 0.003333333        NA            NA
## 11                    uncommon    98 0.326666667        21    0.35000000
```

Statistical Output: Research question 1

Testing Null Hypothesis: 300 sample size probabilities equal to 60 sample size probabilities

```
df_merge_na0 <- replace(df_merge,is.na(df_merge),0)

df_merge_na0$round_p_hat_box <- round(df_merge_na0$p_hat_box, digits = 2)
df_merge_na0$round_p_hat_confirm <- round(df_merge_na0$p_hat_confirm, digits = 2)

null_box_comfirm <- df_merge_na0 %>%
  mutate(prob_box_confirm = ifelse(df_merge_na0$round_p_hat_box == df_merge_na0$round_p_hat_confirm , "

obs_diff <- null_box_comfirm %>%
specify(round_p_hat_box ~ prob_box_confirm) %>%
calculate(stat = "diff in means", order = c("yes", "no"))

null_dist <- null_box_comfirm %>%
  specify(round_p_hat_box ~ prob_box_confirm) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("yes", "no"))

null_dist %>%
  get_p_value(obs_stat = obs_diff, direction = "two_sided")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.532
```

The p_value is 0.532. This value is too large to reject the Null Hypothesis, but it does not tell us if the Alternative Hypothesis is true or false. This may indicate a Type 2 error because when comparing the 60 cards with the 300 cards point estimate there are missing values and inconsistencies in card type probabilities and card types in general. Consistence probability card types between the df_box and df_comfirm were common, holo rare v, rare, and uncommon. However, some inconsistence card type pulled were holo rare, radiant rare, trainer gallery holo rare, trainer gallery holo rare v, ultra rare, secret rare because they were only present in one of the samplings.

Summary statistics: Research question 2

Research question 2: Is the booster box pull percentages greater than the percentage of a booster packs?

To test if the booster box pulls percentages greater than the percentage of a booster packs, I first had to merge the two data frames df_box and df_comfirm together and this new data frame is names df_full_box. The point estimates will be used to test the Null Hypothesis. Because this is a sample size and not a true population size, we designate the percentage as $\hat{p}$.

```
# Merging the two data frame together to get full booster box sample
df_full_box = rbind(df_box, df_confirm)
```

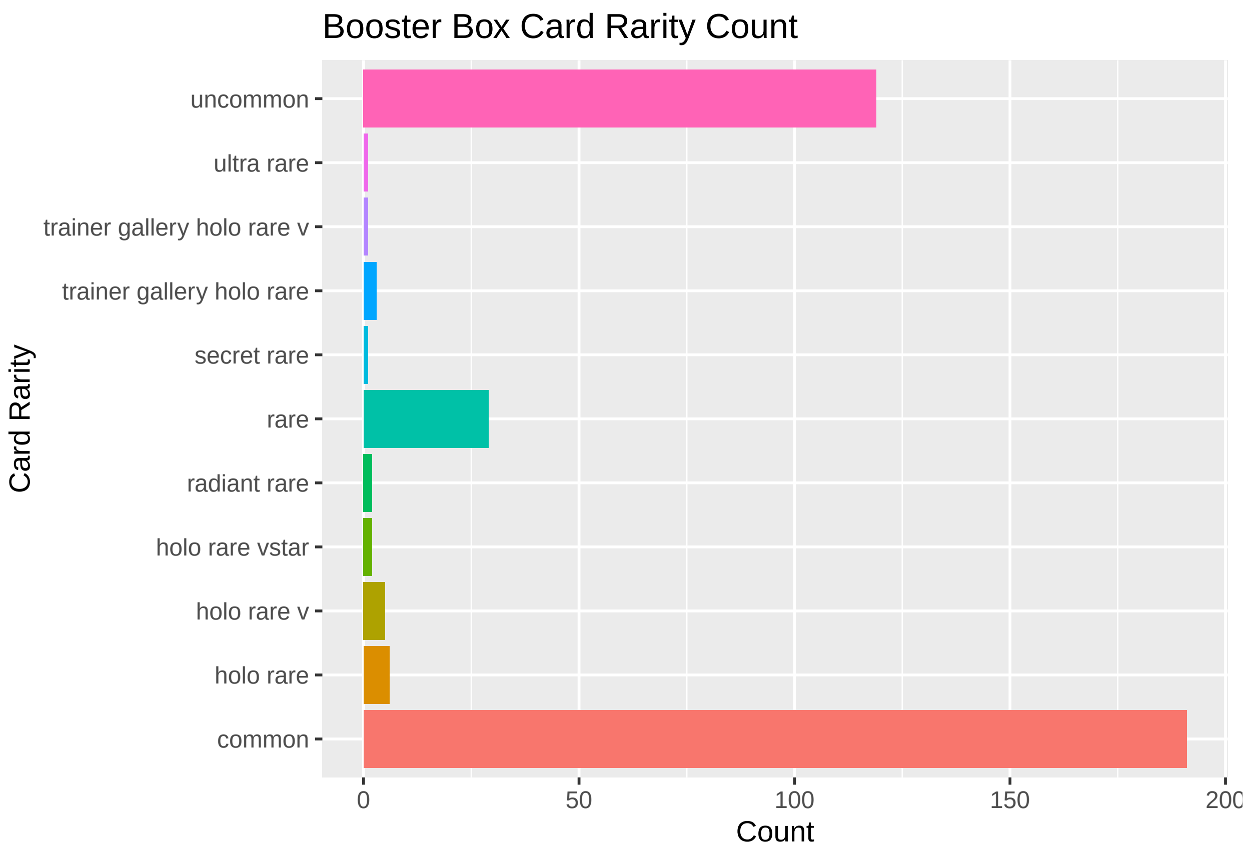
```
df_full_box_merge <- df_full_box %>%
  count(card_rarity) %>%
  mutate(p_hat = n /sum(n))

df_packs_merge <- df_packs %>%
  count(card_rarity) %>%
  mutate(p_hat = n /sum(n))

df_merge_box_pack <- merge(df_full_box_merge, df_packs_merge, by = "card_rarity", all.x = TRUE)
colnames(df_merge_box_pack) <- c('card_rarity','n_box','p_hat_box','n_pack','p_hat_pack')
print(df_merge_box_pack)
```

```
##                     card_rarity n_box   p_hat_box n_pack  p_hat_pack
## 1                        common   191 0.530555556    158 0.526666667
## 2                     holo rare     6 0.016666667      4 0.013333333
## 3                   holo rare v     5 0.013888889      4 0.013333333
## 4               holo rare vstar     2 0.005555556     NA          NA
## 5                  radiant rare     2 0.005555556      2 0.006666667
## 6                          rare    29 0.080555556     27 0.090000000
## 7                   secret rare     1 0.002777778     NA          NA
## 8     trainer gallery holo rare     3 0.008333333      2 0.006666667
## 9   trainer gallery holo rare v     1 0.002777778      3 0.010000000
## 10                   ultra rare     1 0.002777778      2 0.006666667
## 11                     uncommon   119 0.330555556     98 0.326666667
```

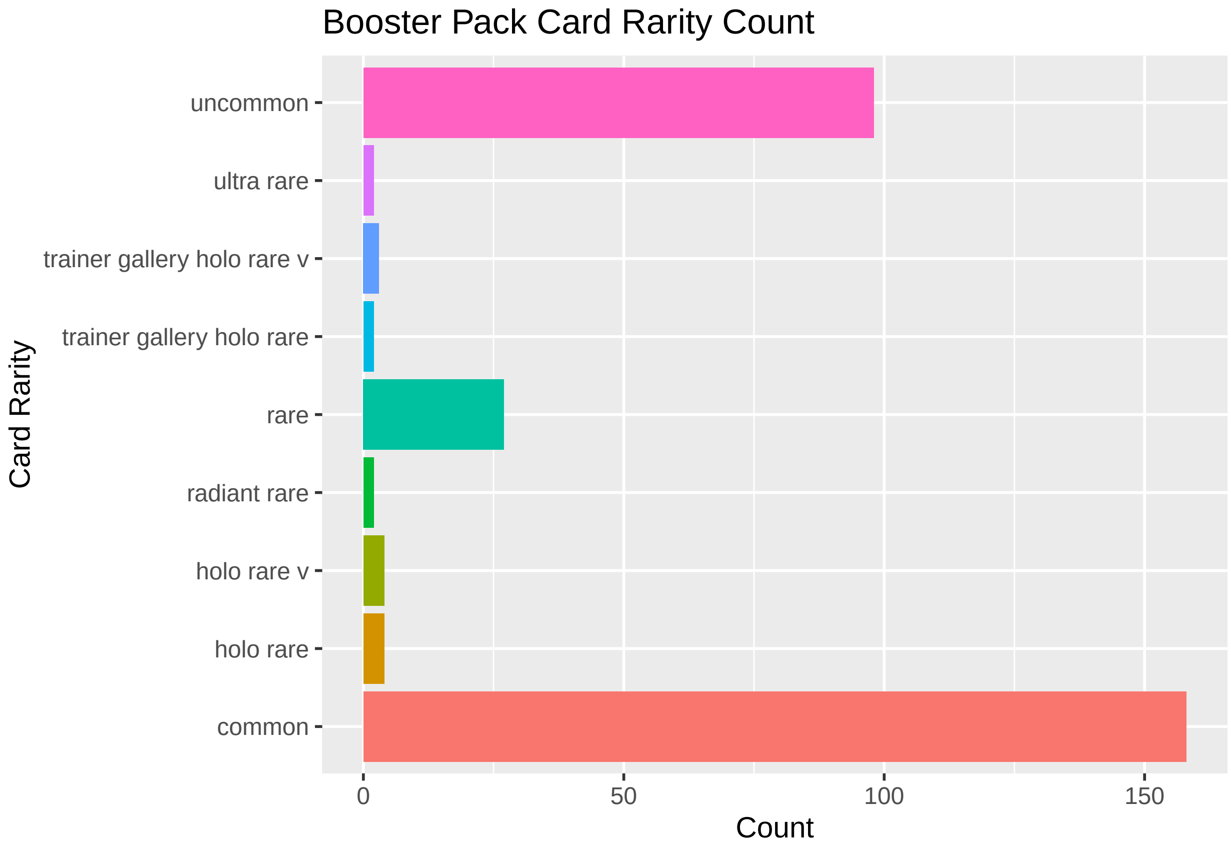
Data Visualizations

Box plots of booster box and booster pack card rarity point estimate counts. Give an overview of which cards are pulled most often and which are harder to pull.

```
ggplot(data = df_full_box, aes(x = card_rarity, fill = card_rarity)) +
  geom_bar()+
  coord_flip()+
  labs(x ="Card Rarity", y = "Count",title = "Booster Box Card Rarity Count")+
  theme(legend.position="none")
```



```
ggplot(data = df_packs, aes(x = card_rarity, fill = card_rarity)) +
  geom_bar()+
  coord_flip()+
  labs(x ="Card Rarity", y = "Count",title = "Booster Pack Card Rarity Count") +
  theme(legend.position="none")
```



Statistical Output: Research question 2

#Testing Null Hypothesis: Booster box probabilities greater than booster pack probabilities

```
df_merge_box_pack_na0 <- replace(df_merge_box_pack,is.na(df_merge_box_pack),0)

null_box_pack <- df_merge_box_pack_na0 %>%
  mutate(prob_box_pack = ifelse(df_merge_box_pack_na0$p_hat_box > df_merge_box_pack_na0$p_hat_pack , "ye

obs_diff_box_pack <- null_box_pack %>%
specify(p_hat_box ~ prob_box_pack) %>%
calculate(stat = "diff in means", order = c("yes", "no"))

null_dist_box_pack <- null_box_pack %>%
  specify(p_hat_box ~ prob_box_pack) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("yes", "no"))


null_dist_box_pack %>%
  get_p_value(obs_stat = obs_diff_box_pack, direction = "two_sided")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1    0.43
```

The p_value is 0.43. Just like from research question1, this value is too large to reject the Null Hypothesis, but it does not tell us if the Alternative Hypothesis is true or false.

Confidence Intervals of Booster Box and Booster Pack

```
df_full_box_1 <- df_full_box %>%
  mutate(ultra_rare = ifelse(card_rarity == "ultra rare", "yes", "no")) %>%
  mutate(rare = ifelse(card_rarity == "rare", "yes", "no"))%>%
  mutate(radiant_rare = ifelse(card_rarity == "radiant rare", "yes", "no"))%>%
  mutate(common = ifelse(card_rarity == "common", "yes", "no"))%>%
  mutate(holo_rare = ifelse(card_rarity == "holo rare", "yes", "no"))%>%
  mutate(holo_rare_v = ifelse(card_rarity == "holo rare v", "yes", "no"))%>%
  mutate(uncommon = ifelse(card_rarity == "uncommon", "yes", "no"))%>%
  mutate(trainer_gallery_holo_rare = ifelse(card_rarity == "trainer gallery holo rare", "yes", "no"))%>
  mutate(trainer_gallery_holo_rare_v = ifelse(card_rarity == "trainer gallery holo rare v", "yes", "no"
  mutate(secret_rare = ifelse(card_rarity == "secret rare", "yes", "no"))%>%
  mutate(holo_rare_vstar = ifelse(card_rarity == "holo rare vstar", "yes", "no"))

a <- df_full_box_1 %>%
  specify(response = ultra_rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

b <- df_full_box_1 %>%
  specify(response = rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

c <- df_full_box_1 %>%
  specify(response = radiant_rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

d <- df_full_box_1 %>%
  specify(response = common, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

e <- df_full_box_1 %>%
  specify(response = holo_rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
f <- df_full_box_1 %>%
  specify(response = holo_rare_v, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

g <- df_full_box_1 %>%
  specify(response = uncommon, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

h <- df_full_box_1 %>%
  specify(response = trainer_gallery_holo_rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

i <- df_full_box_1 %>%
  specify(response = trainer_gallery_holo_rare_v, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

j <- df_full_box_1 %>%
  specify(response = secret_rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

k <- df_full_box_1 %>%
  specify(response = holo_rare_vstar, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

confidence_box <- rbind(a, b, c, d, e, f, g, h, i, j, k)
confidence_box$card_rarity <- c('ultra_rare','rare','radiant_rare','common','holo_rare', 'holo_rare_v',

#Reorder Columns of Data Frame by Index
confidence_box <- confidence_box[ , c(3, 1, 2)]
#print(confidence_box)

df_packs_1 <- df_packs %>%
  mutate(ultra_rare = ifelse(card_rarity == "ultra rare", "yes", "no")) %>%
  mutate(rare = ifelse(card_rarity == "rare", "yes", "no"))%>%
  mutate(radiant_rare = ifelse(card_rarity == "radiant rare", "yes", "no"))%>%
  mutate(common = ifelse(card_rarity == "common", "yes", "no"))%>%
  mutate(holo_rare = ifelse(card_rarity == "holo rare", "yes", "no"))%>%
  mutate(holo_rare_v = ifelse(card_rarity == "holo rare v", "yes", "no"))%>%
  mutate(uncommon = ifelse(card_rarity == "uncommon", "yes", "no"))%>%
  mutate(trainer_gallery_holo_rare = ifelse(card_rarity == "trainer gallery holo rare", "yes", "no"))%>
  mutate(trainer_gallery_holo_rare_v = ifelse(card_rarity == "trainer gallery holo rare v", "yes", "no"
l <- df_packs_1 %>%
  specify(response = ultra_rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

m <- df_packs_1 %>%
  specify(response = rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

n <- df_packs_1 %>%
  specify(response = radiant_rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

o <- df_packs_1 %>%
  specify(response = common, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

p <- df_packs_1 %>%
  specify(response = holo_rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

q <- df_packs_1 %>%
  specify(response = holo_rare_v, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

r <- df_packs_1 %>%
  specify(response = uncommon, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

s <- df_packs_1 %>%
  specify(response = trainer_gallery_holo_rare, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)

t <- df_packs_1 %>%
  specify(response = trainer_gallery_holo_rare_v, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
confidence_pack  <- rbind(l, m, n, o, p, q, r, s, t)
confidence_pack $card_rarity <- c('ultra_rare','rare','radiant_rare','common','holo_rare', 'holo_rare_v

#Reorder Columns of Data Frame by Index
confidence_pack  <- confidence_pack [ , c(3, 1, 2)]
#print(confidence_pack )

confidence_merge_box_pack <- merge(confidence_box, confidence_pack, by = "card_rarity", all.x = TRUE)
colnames(confidence_merge_box_pack) <- c('card_rarity','lower_box','upper_box','lower_pack','upper_pack
print(confidence_merge_box_pack)
```

```
##                     card_rarity   lower_box   upper_box  lower_pack upper_pack
## 1                         common 0.483333333 0.580555556 0.470000000 0.58333333
## 2                      holo_rare 0.005555556 0.030555556 0.003333333 0.02666667
## 3                    holo_rare_v 0.002777778 0.027777778 0.003333333 0.02666667
## 4                holo_rare_vstar 0.000000000 0.013888889          NA         NA
## 5                   radiant_rare 0.000000000 0.013888889 0.000000000 0.01666667
## 6                           rare 0.055555556 0.111111111 0.059916667 0.12333333
## 7                    secret_rare 0.000000000 0.008333333          NA         NA
## 8      trainer_gallery_holo_rare 0.000000000 0.019444444 0.000000000 0.01666667
## 9    trainer_gallery_holo_rare_v 0.000000000 0.008333333 0.000000000 0.02333333
## 10                    ultra_rare 0.000000000 0.008333333 0.000000000 0.01666667
## 11                      uncommon 0.283333333 0.377847222 0.273333333 0.37675000
```

Conclusion

Both research questions yielded a nonrejection of the Null Hypothesis. Therefore, for the first research question the probabilities of the 300 sample cards in the booster box was confirmed by to the 60 card sample from the booster box set aside. For the second research question, the booster box pull percentages are greater than the percentages of a booster pack. However, there are some inconsistencies noticed from the analysis. For the first and second research question, there were missing card rarity variables and for the first question not all probabilities matched between card types.

The limitation on the analysis were cost. To great this sample size it cost approximately 250 dollars. To generate a bigger sample size, the price would increase. Also not begin able to collect all the cards 245 cards in the set again could change the probability of card rarities and exploration on single card probabilities placed limitations on the analysis.