Course Project

Teagan Jenner

12/08/2020

Bootstrapping

We use bootstrap methods to resample from our single sample when we would like to know a characteristic (which in this case is a confidence interval) of our data set.

In bootstrapping, the idea is that an observed sample should contain all of the information about the population. So, the observed sample is considered to be the population. Therefore, the distribution of our test statistic can be simulated by generating random samples from the original sample.

Bootstrap t Confidence Intervals

Let's suppose that we want to estimate $\theta = T(F)$ by $\hat{\theta} = T(\hat{F})$. We can estimate the variance of $\hat{\theta}$ by the value $V(\hat{F})$. For Bootstrap t Confidence Intervals, we identify that $R(X, F) = \frac{T(\hat{F})-T(F)}{\sqrt{V(\hat{F})}} = \frac{\hat{\theta}-\theta}{\sqrt{V(\hat{F})}}$ is roughly pivotal. So, we bootstrap $R(X, F)$ and get a collection of $R(X^*, \hat{F})$.

Let $\hat{G}$ be the distribution of $R(X, F)$ and $\hat{G}^*$ be the distribution of $R(X^*, \hat{F})$.

Then, our confidence interval for θ is given by

$$1 - \alpha = P[\hat{\theta} - \sqrt{V(\hat{F})} * \epsilon_{1-\alpha/2}(\hat{G}) \leq \theta \leq \hat{\theta} - \sqrt{V(\hat{F})} * \epsilon_{\alpha/2}(\hat{G})]$$

where $\epsilon_{\alpha/2}(\hat{G})$ is the quantile of $\hat{G}$.

So, we use the principles of bootstrapping to decide that $\hat{G}$ is approximately equal to $\hat{G}^*$ and therefore, $\epsilon_\alpha(\hat{G}) \approx \epsilon_\alpha(\hat{G}^*) \forall \alpha$. Since we can calculate $\epsilon_\alpha(\hat{G}^*)$ from the histogram of bootstrap values $R(X^*, \hat{F})$, we find that the bootstrap confidence interval for θ is:

$$(T(\hat{F}) - \sqrt{V(\hat{F})} * \epsilon_{1-\alpha/2}(\hat{G}^*), T(\hat{F}) - \sqrt{V(\hat{F})} * \epsilon_{\alpha/2}(\hat{G}^*)) \ .$$

An advantage of Bootstrap t confidence intervals provide dependable confidence intervals and standard error. However, this method is sensitive to outliers and performs poorly when the underlying distribution is heavily tailed.

Permutation Tests

Permutation tests (also known as randomization tests) are another important technique aside from bootstrapping that resamples the observed data to build a sampling distribution. These tests rely on the exchangeability of the data; the Givens and Hoeting text tells us that "the data are exchangeable if the probability of any particular joint outcome is the same regardless of the order in which the observations are considered (318)."

The two main advantages of permutation tests over the bootstrap:

(1) If the permutations are randomly assigned, then the p-value is exact (given that all possible permutations are considered).
(2) These tests are frequently more powerful than bootstrap methods.

Some disadvantages of permutation tests are that the observed standard deviation is not a reliable estimate of standard error and that they are require more assumptions and have less flexibility than bootstrap methods.

The Data Set

The data set used in this application was obtained from the Curbside Express Department of a Giant Eagle store located in Johnstown, Pennsylvania on November 5, 2020.

The data set contains the following information:

- Date
- Day Name (Monday-Sunday)
- Sales (in dollars)
- Transaction Count
- Average Transaction (in dollars)
- Average Items Per Transaction

The focus of this application is to determine if certain days of the week have more Curbside transactions than other days of the week. The possible implications of these results would be that the manager of the department could better understand which days are the busiest and therefore, schedule more hours to those identified busiest days.

It is worth a note that the Average Items Per Transaction also heavily influences the time needed to shop orders and that Transaction Count is not the only feature in this data set that would influence scheduling. However, management does identify the number of orders as the most significant measure to identify.

Finally, it is also important to note that the data comes between January 10, 2019 (which was the first day that the Curbside Department was open at this Giant Eagle) and November 4, 2020 which is in the middle of the COVID-19 pandemic. So, we will need to determine what time intervals to explore in this data set. I have chosen to focus on the transactions in 2020 that occur after the peak COVID, since the department has grown and changed so much due to the pandemic. So, we are focusing on the months of July, August, September, and October of 2020.

Before we continue, we should also note that there is also a daily limit of a maximum of 100 orders in a day. So, we will never have more than 100 orders in a day.

Application

First, we need to read in the data set from the exported excel file:

```
# set working directory
getwd()
```

```
## [1] "/Users/teaganjenner/Desktop/Fall 2020 - Computational Statistics/Course Project"
```

```
setwd("/Users/teaganjenner/Desktop/Fall 2020 - Computational Statistics/Course Project")

# read in excel data file
library(readxl)
data <- read_excel("Curbside_11052020.xlsx", col_names = TRUE)

# rename columns
colnames(data) <- c("date", "day_name", "sales", "trans_count", "avg_trans_dollars", "avg_items_per_tra
```

Summary statistics:

```
summary(data)
```

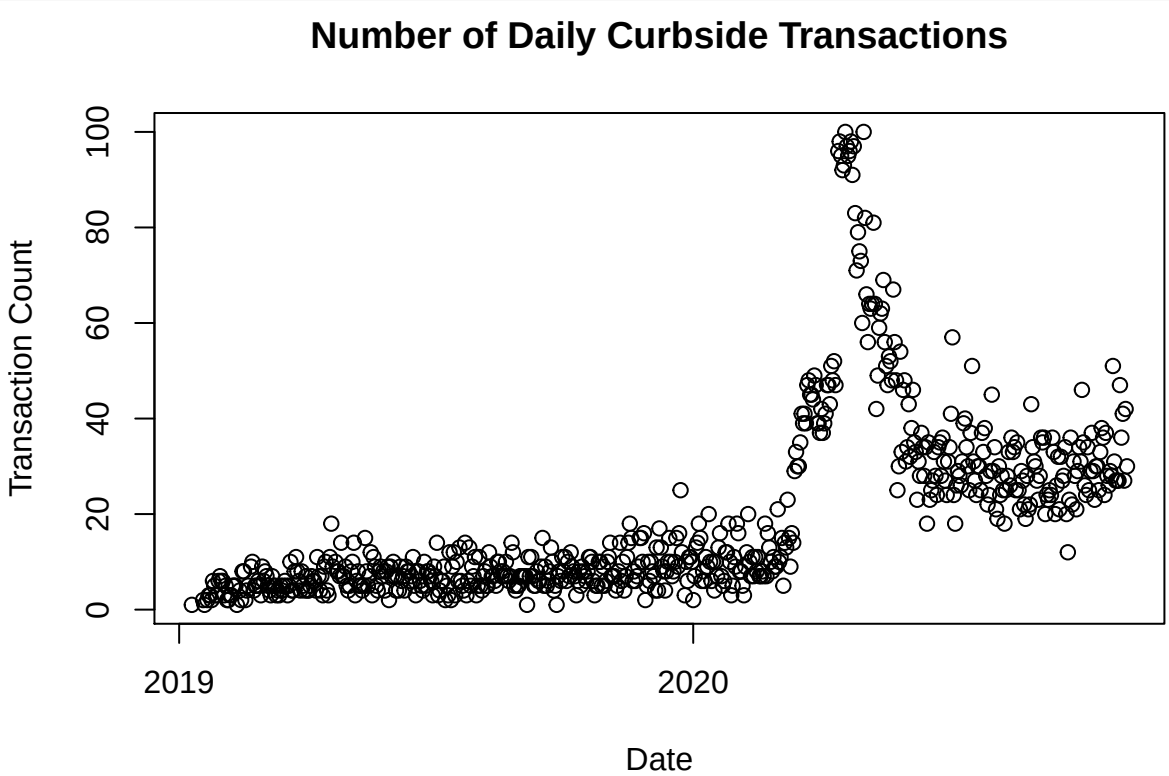
```
##       date                          day_name             sales
##  Min.   :2019-01-10 00:00:00   Length:652         Min.   :    1.67
##  1st Qu.:2019-07-02 18:00:00   Class :character   1st Qu.:  709.48
##  Median :2019-12-12 12:00:00   Mode  :character   Median : 1408.95
##  Mean   :2019-12-13 01:19:30                      Mean   : 2279.57
##  3rd Qu.:2020-05-25 06:00:00                      3rd Qu.: 3437.30
##  Max.   :2020-11-04 00:00:00                      Max.   :12926.85
##   trans_count     avg_trans_dollars avg_items_per_trans
##  Min.   :  1.00   Min.   :  1.67    Min.   :  1.00
##  1st Qu.:  6.00   1st Qu.:102.44    1st Qu.: 31.75
##  Median : 10.00   Median :119.66    Median : 37.30
##  Mean   : 19.18   Mean   :121.54    Mean   : 38.11
##  3rd Qu.: 28.00   3rd Qu.:136.22    3rd Qu.: 43.18
##  Max.   :100.00   Max.   :369.08    Max.   :128.68
```

Change `trans_count` to an integer data type, `day_name` to a factor, and `date` to a date:

```
data$trans_count <- as.integer(data$trans_count)
data$day_name <- as.factor(data$day_name)
data$date <- as.Date(data$date)
```

Before we focus on 2020, let's take a look at the plot of `trans_count` since the start of the Curbside Department in 2019:

```
plot(data$date, data$trans_count, xlab = "Date", ylab = "Transaction Count", main = "Number of Daily Cu
```

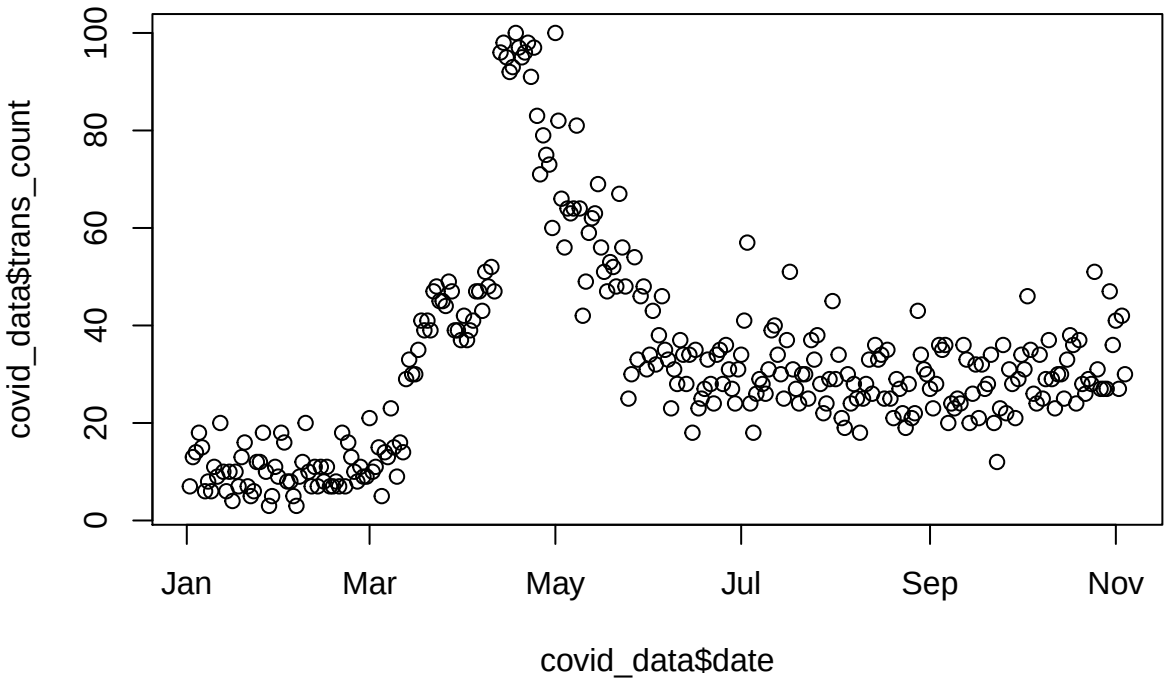


We can see from the plot above that until we entered 2020, the Curbside department didn't tend to have more than 20 orders in a day. There is a clear spike when the virus hits, and there appears to be more orders post-spike than there was pre-COVID. Based on this scatterplot, it appears that the time frame after the

COVID-19 spike would be the time frame to evaluate.

So, let's isolate the data we are most concerned about. Begin by looking at all of the days in 2020:

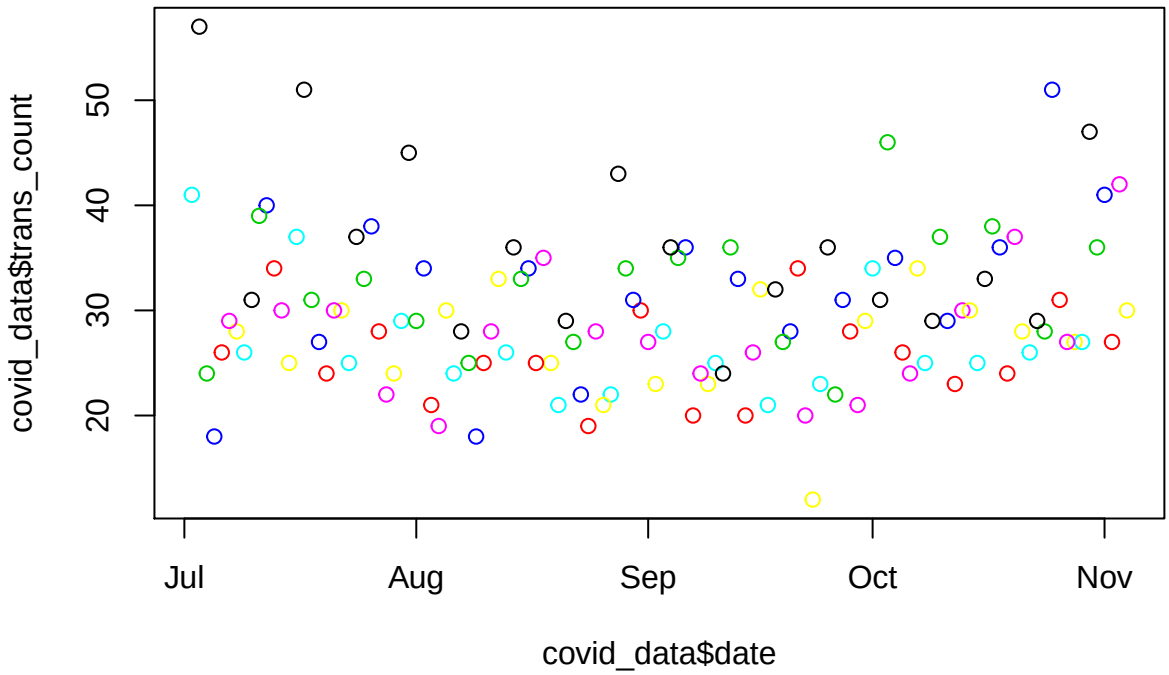
```
covid_data <- subset(data, date > as.Date("2020-01-01"))
plot(covid_data$date, covid_data$trans_count)
```



Now, we have isolated the spike in orders due to COVID-19. It appears that from July-present would be an appropriate time frame to consider in our application.

Let's consider one final plot of the `trans_counts` in this date range that are also colored by `day_name`:

```
covid_data <- subset(data, date > as.Date("2020-07-01"))
plot(covid_data$date, covid_data$trans_count, col = covid_data$day_name)
```



We see that there may be certain days of the week that tend to have more orders (black, blue, green) than others (red, pink) but it is very difficult to tell based on a scatterplot alone. So, we are ready to turn to other methods to help distinguish between `trans_count` and `day_name`.

Bootstrap Confidence Intervals

Define a function to implement the bootstrap t method:

```
bootstrap_t <- function(x, B){
      MuHatStar <- c()

      for(i in 1:B){
            MuHatStar[i] <- mean( sample(x, length(x), replace = TRUE) )
      }
      MuHatStar

      mu_hat <- mean(MuHatStar)

      varMuHatStar <- sum((MuHatStar - mu_hat)^2) / (B-1)

      GHatStar <- c()

      for(i in 1:length(MuHatStar)){
         GHatStar[i] <- (MuHatStar[i] - mu_hat) / sqrt(varMuHatStar)
      }

      lower <- mu_hat - sqrt(varMuHatStar)*quantile(GHatStar, 0.975)
      upper <- mu_hat - sqrt(varMuHatStar)*quantile(GHatStar, 0.025)

      return( c(lower, upper) )
}
```

Create the Bootstrap t Confidence Intervals for each day of the week:

```r
set.seed(120220)

bootstrap_t(covid_data$trans_count[which(covid_data$day_name == "Monday")], 500)
```

```
##    97.5%     2.5%
## 23.79111 27.82028
```

```r
bootstrap_t(covid_data$trans_count[which(covid_data$day_name == "Tuesday")], 500)
```

```
##    97.5%     2.5%
## 24.54389 30.21611
```

```r
bootstrap_t(covid_data$trans_count[which(covid_data$day_name == "Wednesday")], 500)
```

```
##    97.5%     2.5%
## 24.52372 29.60844
```

```r
bootstrap_t(covid_data$trans_count[which(covid_data$day_name == "Thursday")], 500)
```

```
##    97.5%     2.5%
## 24.30400 29.02622
```

```r
bootstrap_t(covid_data$trans_count[which(covid_data$day_name == "Friday")], 500)
```

```
##    97.5%     2.5%
## 32.13011 39.77317
```

```r
bootstrap_t(covid_data$trans_count[which(covid_data$day_name == "Saturday")], 500)
```

```
##    97.5%     2.5%
## 29.51061 34.76200
```

```r
bootstrap_t(covid_data$trans_count[which(covid_data$day_name == "Sunday")], 500)
```

```
##    97.5%     2.5%
## 28.38594 35.97067
```

We get the following confidence intervals after implementing this method for each day of the week:

- Monday: (23.79111, 27.82028)
- Tuesday: (24.54389, 30.21611)
- Wednesday: (24.52372, 29.60844)
- Thursday: (24.30400, 29.02622)
- Friday: (32.13011, 39.77317)
- Saturday: (29.51061, 34.76200)
- Sunday: (28.38594, 35.97067)

We quickly notice that Friday, Saturday, and Sunday appear to have more orders than Monday, Tuesday, Wednesday, and Thursdays. We don't know if this different is significant, but we can begin to see the "busier" days from these 95% confidence intervals.

Interpretation of Bootstrap t Confidence Intervals (** given the data and current trend in Curbside orders **):

-We are 95% confident that the true mean number of transactions on Mondays is between 23 and 28 orders.
-We are 95% confident that the true mean number of transactions on Tuesdays is between 25 and 31 orders.
-We are 95% confident that the true mean number of transactions on Wednesdays is between 25 and 30 orders.
-We are 95% confident that the true mean number of transactions on Thursdays is between 25 and 30 orders.
-We are 95% confident that the true mean number of transactions on Fridays is between 33 and 41 orders.
-We are 95% confident that the true mean number of transactions on Saturdays is between 30 and 35 orders.
-We are 95% confident that the true mean number of transactions on Sundays is between 29 and 37 orders.

Permutation Tests

We will define our test staistic, t, to be the difference between group mean outcomes.

Define a function to implement a Permutation Test:

```
permutationTest <- function(x, y, M){
   t <- c()

   for(i in 1:(M-1)){
      temp_x <- x
      temp_y <- y

      random_x <- sample(1:length(x), 1, replace = FALSE)
      random_y <- sample(1:length(y), 1, replace = FALSE)

      temp_x[random_x] <- y[random_y]
      temp_y[random_y] <- x[random_x]

      t[i] <- mean(temp_x) - mean(temp_y)
   }
   return(t)
}
```

Implement Permutation Tests for each pairing of day of the week:

```
set.seed(2020)
results <- matrix(data = NA, ncol = 2, nrow = 21)

t <- mean(covid_data$trans_count[which(covid_data$day_name == "Monday")]) - mean(covid_data$trans_count
result <- c()
result <- permutationTest(covid_data$trans_count[which(covid_data$day_name == "Monday")], covid_data$tra
result <- sort(result, decreasing = FALSE)
results[1,1] <- "Monday - Tuesday"
results[1,2] <- round(sum(result <= t) / length(result), digits = 4)
```

Performing permutation tests on our subset of the data gives the following results:

```
print(results)
```

```
##       [,1]                    [,2]
##  [1,] "Monday - Tuesday"      "0.4272"
##  [2,] "Monday - Wednesday"    "0.4392"
##  [3,] "Monday - Thursday"     "0.4817"
##  [4,] "Monday - Friday"       "0.1336"
##  [5,] "Monday - Saturday"     "0.2166"
##  [6,] "Monday - Sunday"       "0.2491"
##  [7,] "Tuesday - Wednesday"   "0.5228"
##  [8,] "Tuesday - Thursday"    "0.5998"
##  [9,] "Tuesday - Friday"      "0.1906"
## [10,] "Tuesday - Saturday"    "0.3397"
## [11,] "Tuesday - Saturday"    "0.3277"
## [12,] "Wednesday - Thursday"  "0.5958"
## [13,] "Wednesday - Friday"    "0.1801"
## [14,] "Wednesday - Saturday"  "0.2791"
## [15,] "Wednesday - Sunday"    "0.2871"
## [16,] "Thursday - Friday"     "0.1541"
```

```
## [17,] "Thursday - Saturday"  "0.2796"
## [18,] "Thursday - Sunday"    "0.2786"
## [19,] "Friday - Saturday"    "0.6568"
## [20,] "Friday - Sunday"      "0.6253"
## [21,] "Saturday - Sunday"    "0.5218"
```

With a confidence level of 90%, we fail to conclude that any of the permutation tests yield significant results. Therefore, we conclude that there is no significant different between day and number of transactions for each pairwise comparison.

Summary

While the Bootstrap t Confidence Intervals appeared to support our hypothesis that in particular, Friday/Saturday/Sunday tend to have more orders than the other weekdays, the Permutation Tests failed to conclude that these differences were significant with a 90% level of confidence.

Sources

[1] Givens, Geof H, and Jennifer A Hoeting. Computational Statistics. 2nd ed., John Wiley & Sons, Inc., 2013.

[2] Class Lectures and Instructor Notes