**Module 8 Portfolio Project**

**Prepared by: I**ris Bradford

**Subject:** Sales Volume Comparisons by Lodging

**Prepared for:** Dr. Mills

**Course:** Foundations of Data Analytics

**Date:** 3/8/2020

The preliminary work I conducted to obtain this data began with harvesting the data set that would be used for this project. I obtained consent from the Director of Marketing to use my company’s sales history from 2018-2019 with many rich and pertinent variables. Our data tables are stored in our enterprise database management system. We use Oracle which is in SQL. My SQL skills are minimal, but as I gather from conversations with our Director of Business Intelligence we are on an SSL Server with an Oracle Client that is used as a conduit to Tableau. We would store this data table on our SQL server, accessed and parsed via Oracle into Tableau where we are able to perform these queries.

Accessing Tableau server, I was able pull in the required fields from Enterprise and then make the necessary adjustments to generate a crosstab which was saved as a .csv file to prep the data that I would reference in R.

There were quite a few steps taken to prepare the data before importing the data into R. This included scrubbing for dirty data, which seems to be common for most organizations, and ours is no exception. The issues of the data set I was working with were easily resolved. There were record duplicates, there were data entry errors, as well as some records with missing information.

To give a background on my thoughts of planning for this analysis and expected results, I’ll explain how the alternative hypothesis came to fruition within our organization. One of the departments that works very closely with my department has a couple of gift options that a guest will receive in exchange for attending a presentation. The more common gift offered by this sister department was Visa Gift Cards and there was speculation that these prospects didn’t perform as well as those guests who received lodging as their promotional gift.

When these questions began to surface in the summer of 2018, we started experimenting with the different gifts and tracking their performance. The suspicion was confirmed after various efforts and trials. Hence my expected results going into this study were that guests who have lodging have a higher sales volume than those who don’t.

The analytics question I chose to address was to evaluate the above to determine if there was a positive, negative or neutral relationship between tour guests with and without lodging. I will be evaluating whether a prospect who has lodging spends more on their purchase than a guest who doesn’t have lodging.

In stating this out:

HO: Lodging has no impact on the volume per guest

HA: Lodging has an impact on the volume per guest

What I found to be most profound in this exercise was just how small the P-value calculated out to be. At a 95% confidence interval, a statistically significant P < 0.05 would be adequate to reject the null hypothesis in favor of the alternative. What I found when performing the t.test was P < 0.00896, which would be interpreted as highly statistically significant as P < 0.001. Given the former, finding a P < 0.00896 reassured me with absolute confidence that there is indeed a strong connection between a prospects tour gift, in this case- lodging and the volume spent.

The Data set I chose to work with is the total sales volume from January 2018 through December 2019.

* Process the data using R:

> #dplyr, ggplot, magrittr, stringr

> library(dplyr)

> library(ggplot2)

> library(magrittr)

> library(stringr)

> #Import dataset

> Volume\_Per\_Guest\_Property2<-read.csv("VPG\_by\_TourDateContractDate\_(Parameters)\_crosstab (2).csv")

> Volume\_Per\_Guest\_Property2

ï..TourID\_Dimension TourHasLodging TourAccommodationName

1 116827 TRUE DoubleTree

2 195675 TRUE Grand Timber

3 200328 TRUE Breck Inn

4 216620 TRUE Breck Inn

5 220640 TRUE Beaver Run

6 222119 TRUE Beaver Run

7 222580 TRUE Grand Timber

8 228923 TRUE Breck Inn

9 230234 TRUE Breck Inn

10 230739 TRUE Breck Inn...

66 295962 TRUE GL7

67 296017 TRUE Grand Timber

68 296132 TRUE GL7

69 296401 TRUE Breck Inn

70 298140 TRUE Breck Inn

71 299500 TRUE Beaver Run

ProgramsBu Close.Rate.Curr Average.Sale.Curr VPGnn.Curr

1 Outbound 1 $14,730 $14,730

2 Outbound 0 $0 $0

3 Outbound 0 $0 $0

4 Outbound 0 $0 $0

5 Outbound 1 $15,490 $15,490

6 Outbound 0 $0 $0

7 Outbound 0 $0 $0

8 Outbound 1 $15,600 $15,600

9 Outbound 0 $0 $0

10 Outbound 0 $0 $0...

66 Outbound 0 $0 $0

67 Summit Co 0 $0 $0

68 Outbound 0 $0 $0

69 Outbound 0 $0 $0

70 Outbound 0 $0 $0

71 Outbound 0 $0 $0

Rescind.Vol Rescind..Sales. X X.1 X.2 X.3 X.4

1 $0 0 NA NA NA NA

2 $0 0 NA NA NA NA

3 $0 0 NA NA NA NA

4 $0 0 NA NA NA NA

5 $0 0 NA NA NA NA

6 $0 0 NA NA NA NA

7 $22,584 1 NA NA NA NA

8 $0 0 NA NA NA NA

9 $0 0 NA NA NA NA

10 $0 0 NA NA NA NA

65 $0 0 NA NA NA NA

66 $10,385 1 NA NA NA NA

67 $0 0 NA NA NA NA

68 $0 0 NA NA NA NA

69 $0 0 NA NA NA NA

70 $0 0 NA NA NA NA

71 $0 0 NA NA NA NA

[ reached 'max' / getOption("max.print") -- omitted 33689 rows ]

> #Review beginning of data set

> head(Volume\_Per\_Guest\_Property2)

ï..TourID\_Dimension TourHasLodging TourAccommodationName

1 116827 TRUE DoubleTree

2 195675 TRUE Grand Timber

3 200328 TRUE Breck Inn

4 216620 TRUE Breck Inn

5 220640 TRUE Beaver Run

6 222119 TRUE Beaver Run

ProgramsBu Close.Rate.Curr Average.Sale.Curr VPGnn.Curr

1 Outbound 1 $14,730 $14,730

2 Outbound 0 $0 $0

3 Outbound 0 $0 $0

4 Outbound 0 $0 $0

5 Outbound 1 $15,490 $15,490

6 Outbound 0 $0 $0

Rescind.Vol Rescind..Sales

1 $0 0

2 $0 0

3 $0 0

4 $0 0

5 $0 0

6 $0 0

> VPG2<-str\_replace\_all(Volume\_Per\_Guest\_Property2$VPGnn.Curr, "[^[:alnum:]]", "")

> VPG2<-as.integer(VPG2)

> VPG2

[1] 14730 0 0 0 15490 0 0 15600 0

[10] 0 0 0 0 0 18670 0 0 0

[19] 0 0 18990 0 0 0 0 0 0

[28] 0 0 0 0 0 0 0 0 0

[37] 0 0 0 0 0 0 0 12190 0

[46] 0 0 0 0 0 0 0 0 0

[55] 0 0 0 0 27233 0 0 0 0

[64] 0 0 0 0 0 0 0 0 0

[73] 0 0 0 0 0 0 0 0 0

[82] 0 22539 0 0 0 0 0 0 0

[91] 0 0 0 0 0 0 0 17990 0

[100] 0 0 0 43595 0 0 0 0 9990

[109] 0 0 0 0 0 0 0 0 20490

[118] 0 0 0 21790 0 0 0 0 0

[127] 0 16990 0 0 0 0 0 0 0

...

[298] 0 0 0 93806 0 16990 0 0 0

[307] 19415 0 46993 0 0 0 0 0 18990

[316] 36000 21990 0 0 0 0 0 0 0

[325] 0 13990 0 0 9990 18990 0 0 0

[334] 0 20590 0 0 0 0 0 0 0

[343] 19990 0 0 0 0 0 0 0 71052

[352] 0 0 0 0 0 0 0 0 0

[361] 0 0 36910 0 0 0 0 0 10400

[370] 0 0 0 0 0 0 0 0 0

[379] 0 0 0 0 0 0 0 0 0

[388] 0 0 0 0 0 0 0 0 0

[397] 0 0 0 0 0 0 0 0 0

[406] 0 0 0 0 0 0 0 0 0

[415] 0 0 0 0 0 0 0 0 0

[424] 0 0 0 0 0 0 0 0 0

[433] 0 0 12990 0 0 0 0 17630 0

[442] 0 96700 0 0 16905 16630 0 0 0

...

[919] 0 30000 0 0 0 0 0 0 0

[928] 19590 0 0 0 0 0 0 0 0

[937] 19905 0 0 0 0 0 0 0 0

[946] 0 38275 0 10050 0 0 0 0 15996

[955] 0 0 0 0 0 0 0 20385 0

[964] 0 0 10995 18270 0 0 0 10464 0

[973] 39900 0 21250 0 0 0 24990 47700 11990

[982] 0 0 0 0 16090 0 0 0 0

[991] 14595 18120 0 0 0 24500 0 0 0

[1000] 0

[ reached getOption("max.print") -- omitted 32760 entries ]

> #Read Class

> #Validate as Int

> #Summarize

> #Print Vector

> class(VPG2)

[1] "integer"

> is.integer(VPG2)

[1] TRUE

> summary(VPG2)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0 0 0 4831 0 575495

> round(mean(VPG2), digits = 0)

[1] 4831

> #Define Property as categorical character vector

> Property2<- as.character(Volume\_Per\_Guest\_Property2$TourAccommodationName)

> Property2<- na.omit(Property2)

> Property2

[1] "DoubleTree" "Grand Timber"

[3] "Breck Inn" "Breck Inn"

[5] "Beaver Run" "Beaver Run"

[7] "Grand Timber" "Breck Inn"

[9] "Breck Inn" "Breck Inn"

[11] "Grand Timber" "Breck Inn"

[225] "Ski Village" "Breck Inn"

[227] "Breck Inn" "Grand Timber"

[229] "Beaver Run" "Grand Timber"

[231] "Breck Inn" "GL7"

[233] "Breck Inn" "Ski Village"

[235] "Grand Timber" "Breck Inn"

[237] "Grand Timber" "Beaver Run"

[239] "DoubleTree" "Breck Inn"

...

[821] "Grand Timber" "GC8"

[823] "Breck Inn" "Beaver Run"

[825] "Grand Timber" "Grand Timber"

[827] "" "Grand Timber"

[829] "" "Grand Timber"

[975] "GL7" "Ski Village"

[977] "" ""

[979] "GL7" "GL7"

[981] "Grand Timber" ""

[983] "Beaver Run" "Breck Inn"

[985] "" "Grand Timber"

[987] "Beaver Run" ""

[989] "Grand Timber" "Grand Timber"

[991] "" "Grand Timber"

[993] "" "GL7"

[995] "GC8" "Breck Inn"

[997] "Grand Timber" "GL7"

[999] "" ""

[ reached getOption("max.print") -- omitted 32760 entries ]

> #Define each property vector

> #Calcuate Occupancy Percentage

> GP <- filter(Volume\_Per\_Guest\_Property2, Property2 == "Gold Point")

> GP<- as.integer(count(GP))

> GP

[1] 261

> GTL <- filter(Volume\_Per\_Guest\_Property2, Property2 == "Grand Timber")

> GTL<- as.integer(count(GTL))

> GTL

[1] 5346

> GL7 <- filter(Volume\_Per\_Guest\_Property2, Property2 == "GL7")

> GL7 <- as.integer(count(GL7))

> GL7

[1] 3596

> GC8 <- filter(Volume\_Per\_Guest\_Property2, Property2 == "GC8")

> GC8<- as.integer(count(GC8))

> GC8

[1] 1420

> BI <- filter(Volume\_Per\_Guest\_Property2, Property2 == "Breck Inn")

> BI <-as.integer(count(BI))

> BI

[1] 3483

> SV <- filter (Volume\_Per\_Guest\_Property2, Property2 == "Ski Village")

> SV<-as.integer(count(SV))

> SV

[1] 731

> BR <- filter (Volume\_Per\_Guest\_Property2, Property2 == "Beaver Run")

> BR<-as.integer(count(BR))

> BR

[1] 994

> DT <- filter (Volume\_Per\_Guest\_Property2, Property2 == "DoubleTree")

> DT<-as.integer(count(DT))

> DT

[1] 841

> IMI <- filter (Volume\_Per\_Guest\_Property2, Property2 == "IMI")

> IMI<-as.integer(count(IMI))

> IMI

[1] 49

> BS <- filter (Volume\_Per\_Guest\_Property2, Property2 == "Baymont Suites")

> BS<-as.integer(count(BS))

> BS

[1] 0

> MTT <- filter (Volume\_Per\_Guest\_Property2, Property2 == "Mt. Thunder Lodge")

> MTT<-as.integer(count(MTT))

> MTT

[1] 0

> RI <- filter (Volume\_Per\_Guest\_Property2, Property2 == "Residence Inn")

> RI<-as.integer(count(RI))

> RI

[1] 20

> RQ <- filter (Volume\_Per\_Guest\_Property2, Property2 == "RQ")

> RQ<-as.integer(count(RQ))

> RQ

[1] 118

> Total\_Units<- (sum(GC8, GL7, GTL,GP,BI, Partners))

> Total\_Units

[1] 17853

> Partners<- (sum(SV, BR, DT, BR, MTT, IMI,RI, RQ))

> Partners

[1] 3747

> OccupancyPP<- Partners/Total\_Units

> OccupancyPP<-round(OccupancyPP, digits = 2)

> OccupancyPP

[1] 0.21

> OccupancyGTL<- (GTL/Total\_Units)

> OccupancyGTL<-round (OccupancyGTL, digits=2)

> OccupancyGTL

[1] 0.3

> OccupancyGL7<- (GL7/Total\_Units)

> OccupancyGL7<- round (OccupancyGL7, digits=2)

> OccupancyGL7

[1] 0.2

> OccupancyGC8<- (GC8/Total\_Units)

> OccupancyGC8<- round (OccupancyGC8, digits=2)

> OccupancyGC8

[1] 0.08

> OccupancyBI<- (BI/Total\_Units)

> OccupancyBI<- round (OccupancyBI, digits=2)

> OccupancyBI

[1] 0.2

> #Set Show\_Count Vector

> #Remove NA's from variable

> #Define as boolean

> #Print

> #Verify is properly classed, verify is logical

> Show\_Count2<- (Volume\_Per\_Guest\_Property2$Close.Rate.Curr)

> Show\_Count2<- (as.logical(Show\_Count2))

> Show\_Count2

[1] TRUE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE

[11] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE

[21] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[31] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[41] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE

[51] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE

[61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[71] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[81] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[861] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE

[871] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[881] TRUE FALSE TRUE FALSE FALSE TRUE TRUE FALSE TRUE FALSE

[891] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE

[901] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[911] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE

[921] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE

[931] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE

[941] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE

[951] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE

[961] FALSE TRUE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE

[971] TRUE FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE TRUE

[981] TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE

[991] TRUE TRUE FALSE FALSE FALSE TRUE NA FALSE FALSE FALSE

[ reached getOption("max.print") -- omitted 32760 entries ]

> class(Show\_Count2)

[1] "logical"

> is.logical(Show\_Count2)

[1] TRUE

> Avg\_Sale<-(Volume\_Per\_Guest\_Property2$Average.Sale.Curr)

> Avg\_Sale<-str\_replace\_all(Avg\_Sale, "[^[:alnum:]]", "")

> Avg\_Sale<-as.numeric(Avg\_Sale)

> Avg\_Sale<-round(Avg\_Sale, digits=0)

> Avg\_Sale

[1] 14730 0 0 0 15490 0 0 15600 0

[10] 0 0 0 0 0 18670 0 0 0

[19] 0 0 18990 0 0 0 0 0 0

[28] 0 0 0 0 0 0 0 0 0

[37] 0 0 0 0 0 0 0 12190 0

[46] 0 0 0 0 0 0 0 0 0

[55] 0 0 0 0 27233 0 0 0 0

[64] 0 0 0 0 0 0 0 0 0

[73] 0 0 0 0 0 0 0 0 0

[82] 0 22539 0 0 0 0 0 0 0

[91] 0 0 0 0 0 0 0 17990 0

[100] 0 0 0 43595 0 0 0 0 9990

[109] 0 0 0 0 0 0 0 0 20490

[118] 0 0 0 21790 0 0 0 0 0

[127] 0 16990 0 0 0 0 0 0 0

[136] 0 0 97669 0 16400 0 13990 0 0

[145] 0 0 0 0 0 0 0 0 0

[154] 0 0 0 0 0 0 22800 0 11990

[163] 9990 0 0 15990 0 0 0 0 0

[172] 0 0 0 0 0 0 0 0 0

[181] 0 20224 0 24990 0 0 0 0 0

[190] 0 0 0 0 0 33690 0 0 0

[199] 0 0 0 0 19290 0 15490 0 0

[208] 0 0 0 0 0 0 0 42300 0

[217] 0 19916 38000 0 0 0 0 0 0

[226] 0 0 0 0 13990 0 36400 14490 0

[235] 0 0 0 0 0 0 0 0 0

[244] 0 0 0 0 0 0 0 0 0

[253] 0 0 0 0 0 0 0 0 0

[262] 0 0 0 0 0 0 0 35640 0

[271] 0 9540 0 0 0 0 0 13682 0

[280] 0 0 15070 0 0 0 0 0 45200

[289] 0 0 0 0 0 0 0 0 0

[298] 0 0 0 93806 0 16990 0 0 0

[307] 19415 0 46993 0 0 0 0 0 18990

[316] 36000 21990 0 0 0 0 0 0 0

[325] 0 13990 0 0 9990 18990 0 0 0

[334] 0 20590 0 0 0 0 0 0 0

[343] 19990 0 0 0 0 0 0 0 71052

[352] 0 0 0 0 0 0 0 0 0

[361] 0 0 36910 0 0 0 0 0 10400

[370] 0 0 0 0 0 0 0 0 0

[379] 0 0 0 0 0 0 0 0 0

[388] 0 0 0 0 0 0 0 0 0

[397] 0 0 0 0 0 0 0 0 0

[406] 0 0 0 0 0 0 0 0 0

[415] 0 0 0 0 0 0 0 0 0

[424] 0 0 0 0 0 0 0 0 0

[433] 0 0 12990 0 0 0 0 17630 0

[442] 0 96700 0 0 16905 16630 0 0 0

[451] 0 18997 0 0 0 0 0 0 0

[460] 0 0 0 0 0 0 0 15000 15495

[469] 0 0 0 0 0 19517 0 10500 0

[478] 0 0 0 0 0 0 0 0 0

[487] 0 0 0 0 0 0 22170 0 10834

[496] 0 0 0 0 0 60400 25396 0 0

[505] 0 0 18330 79038 0 0 0 0 0

[514] 0 0 0 0 0 0 0 0 0

[523] 0 0 0 0 16690 0 0 0 0

[532] 0 0 0 0 0 14061 0 0 0

[541] 0 0 0 0 0 0 0 0 0

[550] 0 0 0 0 0 0 0 0 0

[559] 0 0 39620 0 0 0 0 12690 0

[568] 21790 0 0 0 0 23390 0 0 0

[577] 0 0 0 0 0 0 0 0 0

[586] 0 20290 0 0 0 23350 0 0 0

[595] 0 0 0 18570 0 0 0 0 0

[604] 0 0 32160 0 0 0 0 0 14113

[613] 0 0 0 22790 0 0 0 0 0

[622] 0 0 0 0 0 0 0 0 0

[631] 0 0 0 16000 0 0 0 0 0

[640] 24007 0 26100 0 50300 0 0 0 0

[649] 0 0 0 0 0 0 0 0 0

[658] 0 31660 0 0 0 0 0 0 0

[667] 0 27234 0 0 0 0 0 0 0

[676] 0 0 0 15990 0 0 18110 0 0

[685] 0 0 0 152452 0 0 0 0 0

[694] 0 0 0 0 0 0 0 0 0

[703] 0 0 0 17990 0 0 0 55000 0

[712] 0 0 16100 0 0 0 0 0 0

[721] 0 0 0 0 0 0 0 0 0

[730] 31527 0 0 0 0 0 0 61800 0

[739] 0 0 25600 0 0 46400 0 0 0

[748] 0 22148 0 0 0 0 0 0 0

[757] 13945 0 0 0 0 0 0 18290 0

[766] 0 0 0 0 16990 0 0 0 0

[775] 69514 0 14690 0 0 0 31750 0 0

[784] 0 0 0 0 0 0 0 0 0

[793] 0 17900 0 0 0 0 0 0 0

[802] 0 0 0 0 0 0 0 0 22175

[811] 0 0 0 21600 0 0 0 0 0

[820] 0 18535 9990 0 0 0 0 0 0

[829] 0 0 0 0 0 0 0 0 0

[838] 0 0 0 0 0 19890 0 0 0

[847] 0 0 0 0 0 0 0 10990 0

[856] 13990 0 0 0 0 0 0 0 0

[865] 15990 0 0 0 0 0 0 0 0

[874] 0 0 0 0 0 0 0 15990 0

[883] 58000 0 0 10990 10400 0 56043 0 0

[892] 0 19990 0 0 0 0 0 0 16775

[901] 0 0 0 0 0 0 0 0 0

[910] 0 0 36995 0 0 0 0 0 0

[919] 0 30000 0 0 0 0 0 0 0

[928] 19590 0 0 0 0 0 0 0 0

[937] 19905 0 0 0 0 0 0 0 0

[946] 0 38275 0 10050 0 0 0 0 15996

[955] 0 0 0 0 0 0 0 20385 0

[964] 0 0 10995 18270 0 0 0 10464 0

[973] 39900 0 21250 0 0 0 24990 47700 11990

[982] 0 0 0 0 16090 0 0 0 0

[991] 14595 18120 0 0 0 24500 9760 0 0

[1000] 0

[ reached getOption("max.print") -- omitted 32760 entries ]

> class(Avg\_Sale)

[1] "numeric"

> is.numeric(Avg\_Sale)

[1] TRUE

> Volume2<-str\_replace\_all(Volume\_Per\_Guest\_Property2$Average.Sale.Curr, "[$,]", "")

> #Volume2<- na.omit(Volume2)

> Volume2<-(as.integer(Volume2))

> Volume2

[1] 14730 0 0 0 15490 0 0 15600 0

[10] 0 0 0 0 0 18670 0 0 0

[19] 0 0 18990 0 0 0 0 0 0

[28] 0 0 0 0 0 0 0 0 0

[37] 0 0 0 0 0 0 0 12190 0

[46] 0 0 0 0 0 0 0 0 0

[55] 0 0 0 0 27233 0 0 0 0

[64] 0 0 0 0 0 0 0 0 0

[73] 0 0 0 0 0 0 0 0 0

[82] 0 22539 0 0 0 0 0 0 0

[91] 0 0 0 0 0 0 0 17990 0

[100] 0 0 0 43595 0 0 0 0 9990

[109] 0 0 0 0 0 0 0 0 20490

[118] 0 0 0 21790 0 0 0 0 0

[127] 0 16990 0 0 0 0 0 0 0

[136] 0 0 97669 0 16400 0 13990 0 0

[145] 0 0 0 0 0 0 0 0 0

[154] 0 0 0 0 0 0 22800 0 11990

[163] 9990 0 0 15990 0 0 0 0 0

[172] 0 0 0 0 0 0 0 0 0

[181] 0 20224 0 24990 0 0 0 0 0

[703] 0 0 0 17990 0 0 0 55000 0

[712] 0 0 16100 0 0 0 0 0 0

[721] 0 0 0 0 0 0 0 0 0

[730] 31527 0 0 0 0 0 0 61800 0

[739] 0 0 25600 0 0 46400 0 0 0

[748] 0 22148 0 0 0 0 0 0 0

[757] 13945 0 0 0 0 0 0 18290 0

[766] 0 0 0 0 16990 0 0 0 0

[775] 69514 0 14690 0 0 0 31750 0 0

[784] 0 0 0 0 0 0 0 0 0

[793] 0 17900 0 0 0 0 0 0 0

[802] 0 0 0 0 0 0 0 0 22175

[811] 0 0 0 21600 0 0 0 0 0

[820] 0 18535 9990 0 0 0 0 0 0

[829] 0 0 0 0 0 0 0 0 0

[838] 0 0 0 0 0 19890 0 0 0

[847] 0 0 0 0 0 0 0 10990 0

[856] 13990 0 0 0 0 0 0 0 0

[865] 15990 0 0 0 0 0 0 0 0

[874] 0 0 0 0 0 0 0 15990 0

[883] 58000 0 0 10990 10400 0 56043 0 0

[892] 0 19990 0 0 0 0 0 0 16775

[901] 0 0 0 0 0 0 0 0 0

[910] 0 0 36995 0 0 0 0 0 0

[919] 0 30000 0 0 0 0 0 0 0

[928] 19590 0 0 0 0 0 0 0 0

[937] 19905 0 0 0 0 0 0 0 0

[946] 0 38275 0 10050 0 0 0 0 15996

[955] 0 0 0 0 0 0 0 20385 0

[964] 0 0 10995 18270 0 0 0 10464 0

[973] 39900 0 21250 0 0 0 24990 47700 11990

[982] 0 0 0 0 16090 0 0 0 0

[991] 14595 18120 0 0 0 24500 9760 0 0

[1000] 0

[ reached getOption("max.print") -- omitted 32760 entries ]

> HasLodging<-(Volume\_Per\_Guest\_Property2$TourHasLodging)

> HasLodging<-as.logical(HasLodging)

> class(HasLodging)

[1] "logical"

> is.logical(HasLodging)

[1] TRUE

> as.num = function(Volume2, na.strings = "NA") {

+ stopifnot(is.character(x))

+ na = Volume2 %in% na.strings

+ Volume2[na] = 0

+ Volume2 = as.numeric(Volume2)

+ Volume2[na] = NA\_real\_

+ summary(Volume2)

+ Volume2

+ }

> as.numeric(Volume2, na.strings="Volume 2")

> Volume2

[1] 14730 0 0 0 15490 0 0 15600 0

[10] 0 0 0 0 0 18670 0 0 0

[19] 0 0 18990 0 0 0 0 0 0

[28] 0 0 0 0 0 0 0 0 0

[37] 0 0 0 0 0 0 0 12190 0

[46] 0 0 0 0 0 0 0 0 0

[55] 0 0 0 0 27233 0 0 0 0

[64] 0 0 0 0 0 0 0 0 0

[73] 0 0 0 0 0 0 0 0 0

[82] 0 22539 0 0 0 0 0 0 0

[91] 0 0 0 0 0 0 0 17990 0

[100] 0 0 0 43595 0 0 0 0 9990

[109] 0 0 0 0 0 0 0 0 20490

[118] 0 0 0 21790 0 0 0 0 0

[127] 0 16990 0 0 0 0 0 0 0

[136] 0 0 97669 0 16400 0 13990 0 0

[145] 0 0 0 0 0 0 0 0 0

[154] 0 0 0 0 0 0 22800 0 11990

[163] 9990 0 0 15990 0 0 0 0 0

[

[775] 69514 0 14690 0 0 0 31750 0 0

[784] 0 0 0 0 0 0 0 0 0

[793] 0 17900 0 0 0 0 0 0 0

[802] 0 0 0 0 0 0 0 0 22175

[811] 0 0 0 21600 0 0 0 0 0

[820] 0 18535 9990 0 0 0 0 0 0

[829] 0 0 0 0 0 0 0 0 0

[838] 0 0 0 0 0 19890 0 0 0

[847] 0 0 0 0 0 0 0 10990 0

[856] 13990 0 0 0 0 0 0 0 0

[865] 15990 0 0 0 0 0 0 0 0

[874] 0 0 0 0 0 0 0 15990 0

[883] 58000 0 0 10990 10400 0 56043 0 0

[892] 0 19990 0 0 0 0 0 0 16775

[901] 0 0 0 0 0 0 0 0 0

[910] 0 0 36995 0 0 0 0 0 0

[919] 0 30000 0 0 0 0 0 0 0

[928] 19590 0 0 0 0 0 0 0 0

[937] 19905 0 0 0 0 0 0 0 0

[946] 0 38275 0 10050 0 0 0 0 15996

[955] 0 0 0 0 0 0 0 20385 0

[964] 0 0 10995 18270 0 0 0 10464 0

[973] 39900 0 21250 0 0 0 24990 47700 11990

[982] 0 0 0 0 16090 0 0 0 0

[991] 14595 18120 0 0 0 24500 9760 0 0

[1000] 0

[ reached getOption("max.print") -- omitted 32760 entries ]

> ggplot(Volume\_Per\_Guest\_Property2, aes(x=Property2, y=VPG2)) +

+ geom\_boxplot()+

+ theme(axis.text.x=element\_text(angle = 90, vjust = .05))

> str(Volume\_Per\_Guest\_Property2)

'data.frame': 33760 obs. of 14 variables:

$ ï..TourID\_Dimension : Factor w/ 33736 levels "116827","195675",..: 1 2 3 4 5 6 7 8 9 10 ...

$ TourHasLodging : Factor w/ 3 levels "FALSE","Total",..: 3 3 3 3 3 3 3 3 3 3 ...

$ TourAccommodationName: Factor w/ 20 levels ""," GL7"," Grand Timber",..: 8 12 7 7 6 6 12 7 7 7 ...

$ ProgramsBu : Factor w/ 16 levels "Broker Re","Front Ran",..: 13 13 13 13 13 13 13 13 13 13 ...

$ Close.Rate.Curr : int 1 0 0 0 1 0 0 1 0 0 ...

$ Average.Sale.Curr : Factor w/ 3693 levels "$0 ","$1,000 ",..: 929 1 1 1 1078 1 1 1097 1 1 ...

$ VPGnn.Curr : Factor w/ 3681 levels "$0 ","$1,000 ",..: 939 1 1 1 1083 1 1 1103 1 1 ...

$ Rescind.Vol : Factor w/ 534 levels "$0 ","$10,000 ",..: 1 1 1 1 1 1 281 1 1 1 ...

$ Rescind..Sales. : int 0 0 0 0 0 0 1 0 0 0 ...

$ X : logi NA NA NA NA NA NA ...

$ X.1 : logi NA NA NA NA NA NA ...

$ X.2 : logi NA NA NA NA NA NA ...

$ X.3 : logi NA NA NA NA NA NA ...

$ X.4 : Factor w/ 2 levels "","z": 1 1 1 1 1 1 1 1 2 1 ...

> sd(Avg\_Sale, na.rm= FALSE)

[1] 17019.46

> #boxplot(VPG2)

> #geom\_boxplot aes (x VPG2 y Property2)

> #Ho- Mean VPG of

> ?boxplot

> t.test(VPG2~HasLodging, mu= 0, alt = "two.sided", conf = 0.95, var.eq=F, paired=F)

Welch Two Sample t-test

data: VPG2 by HasLodging

t = 2.6146, df = 5042.9, p-value = 0.00896

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

226.7155 1585.6051

sample estimates:

mean in group FALSE mean in group TRUE

5621.412 4715.252

> #t.test(VPG, data = Volume\_Per\_Guest\_Property, mu= 0, alt = "greater", conf = 0.95, var.eq = FALSE, paired = FALSE)

> mod <- lm(VPG2 ~ HasLodging, data=Volume\_Per\_Guest\_Property2)

> anova(mod)

Analysis of Variance Table

Response: VPG2

Df Sum Sq Mean Sq F value Pr(>F)

HasLodging 1 3.0982e+09 3098223767 10.677 0.001086 \*\*

Residuals 33757 9.7957e+12 290182987

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> ggplot(Volume\_Per\_Guest\_Property2, aes(x=Property2, y=VPG2)) +

+ geom\_boxplot()+

+ theme(axis.text.x=element\_text(angle = 90, vjust = .05))

> total\_sales<-sum(Avg\_Sale)

> total\_sales

[1] 160804323

> TS<- total\_sales/Total\_Units

> TS

[1] 9007.132

Calculate R summary statistics of Volume Per Guest

> summary(VPG2)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0 0 0 4831 0 575495

* Hypotheses for a two-sample t test
* Welch Two Sample t-test
* data: VPG2 by HasLodging
* t = 2.6146, df = 5042.9, p-value = 0.00896
* alternative hypothesis: true difference in means is not equal to 0
* 95 percent confidence interval:
* 226.7155 1585.6051
* sample estimates:
* mean in group FALSE mean in group TRUE
* 5621.412 4715.252

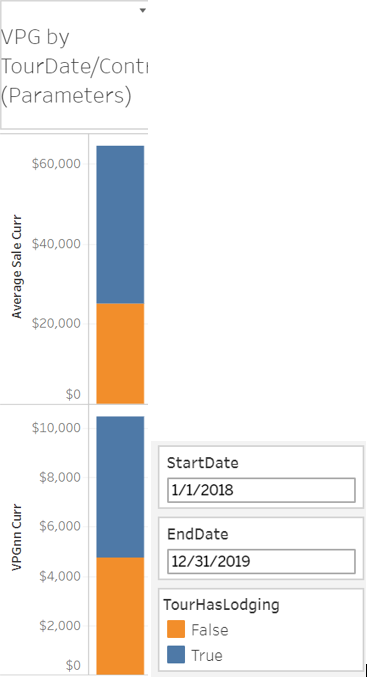
The ggPlot below depicts the difference in VPG by Lodging Property:

A screenshot of a cell phone

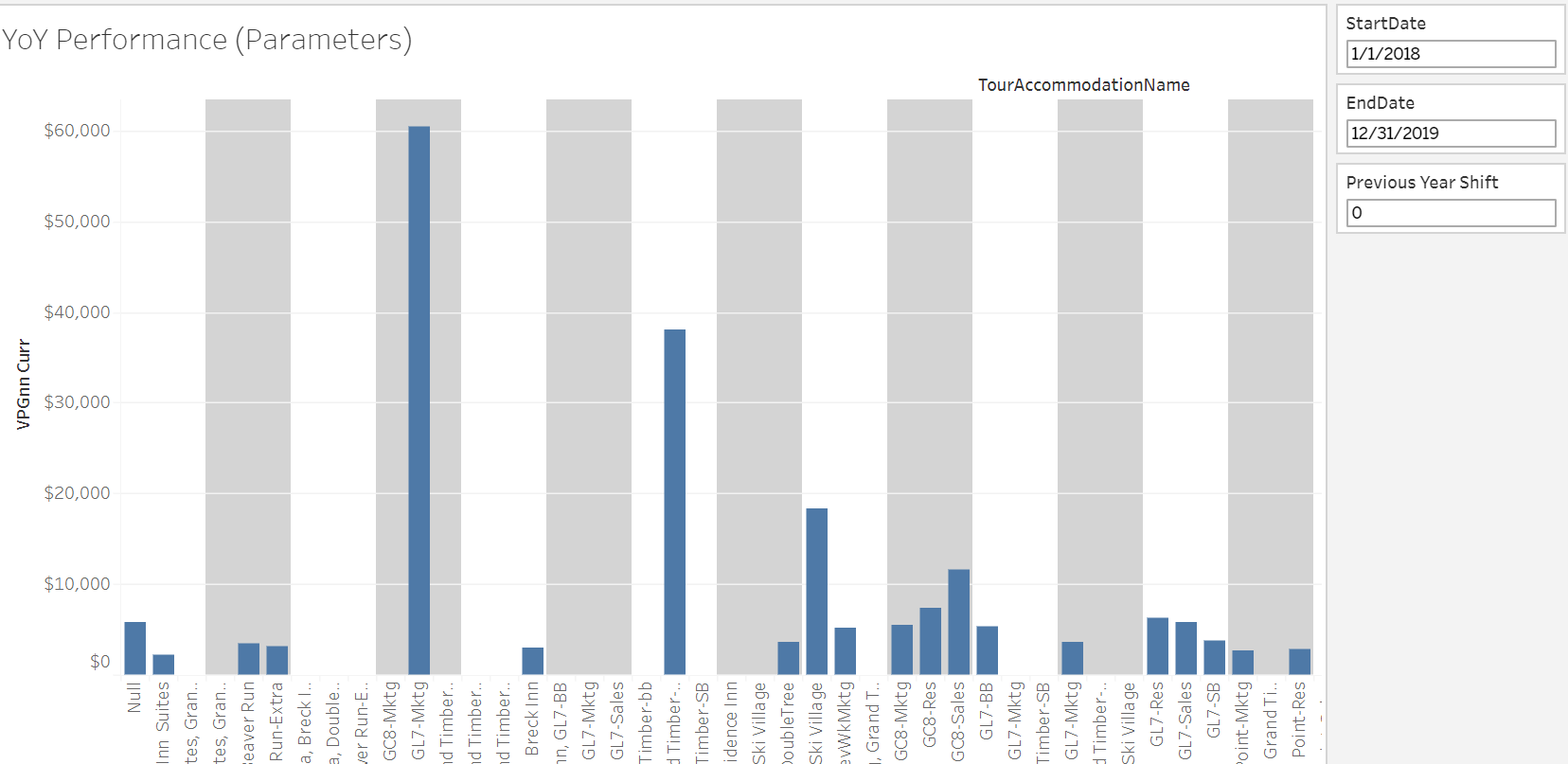
Description automatically generated

* Following the GitHub Upload instruction sheet, upload your finished project to GitHub
* Determine how to interpret results and reach the right conclusions
* Determine the question(s) you are addressing
* Plan and build a system for storing, querying, or otherwise managing the data
* They system used to store this data will live on the company SQL server, accessed
* Produce some graphs to support your analysis
* If you like, create other interesting visualizations

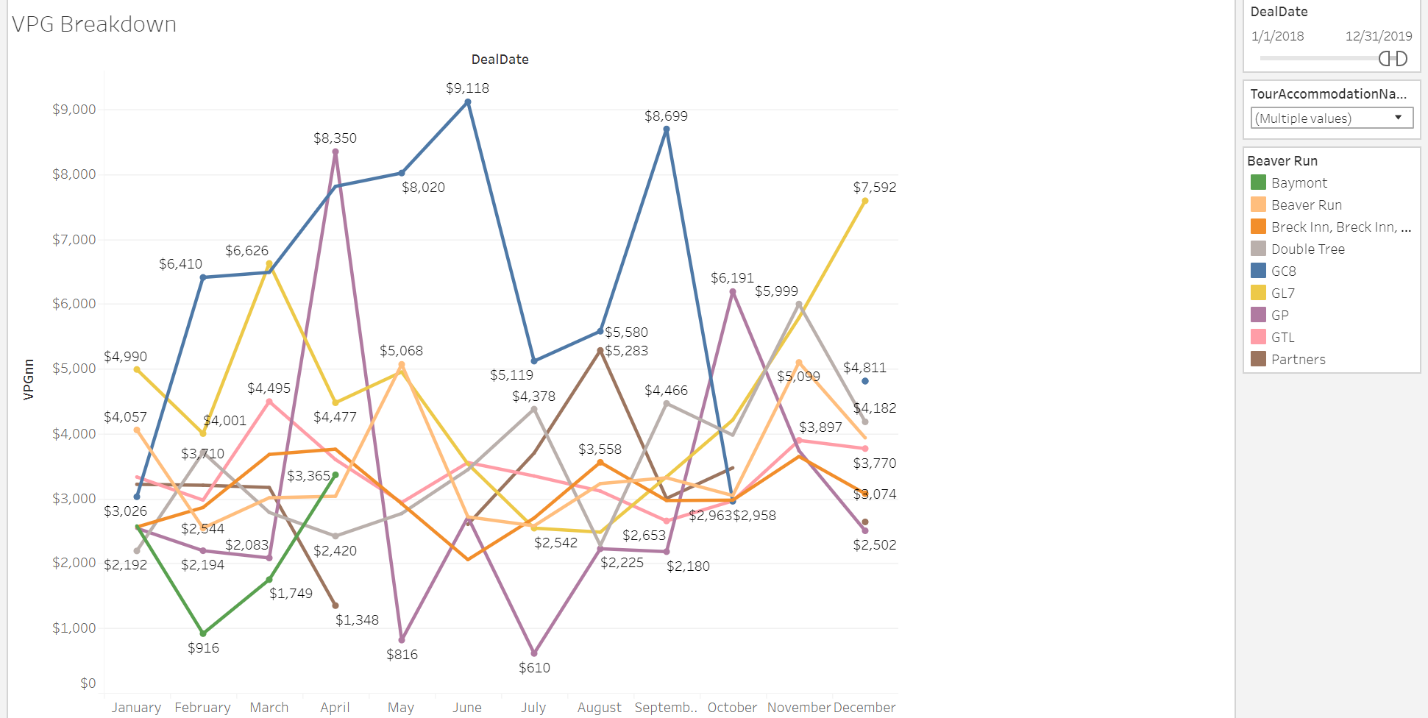
The below visualization demonstrates that tour that have lodging have a greater VPG and a greater average sale. Again, confirming the work in R studio.



This visualization demonstrates, just as we saw in the ggplot above- of the lodging accommodations in this data set, GL7 has the highest VPG .



Below we see the fluctuation of VPG by month Jan 2018 – Dec 2019. This essentially combines the values from the month of each year in illustrating the plot line of each property.



<https://www.oracle.com/database/what-is-database.html>

* [https://help.tableau.com/current/pro/desktop/en-us/examples\_oracle.htm#](https://help.tableau.com/current/pro/desktop/en-us/examples_oracle.htm)
* <https://www.statsdirect.com/help/basics/p_values.htm>