

TPS Analysis Technical Appendix

Andrew Nalundasan, Lynna Tran, Rosalie Sherry, Vishaal Diwan

Load Libraries

```
library(readxl) # to read in data
library(tidyverse)
library(jtools) # for export_summs
library(vtable) # for vtable
library(dplyr)
library(lubridate) # to cast date-time datatypes
library(gridExtra) # for grid arrange
library(janitor) # for tabyl
library(writexl)
library(lubridate)
library(sqldf)
```

Load data

Read-in Ledger Reports

```
ledger_balance <- read_excel("../02_raw_data/ledger/Ledger Beginning and Ending Balances.xlsx")
ledger_01 <- read_excel("../02_raw_data/ledger/CLEAN 54445 - MAILINGS AND NEWSLETTER.xlsx")
ledger_02 <- read_excel("../02_raw_data/ledger/CLEAN 54410 - MARKETING.xlsx")
ledger_03 <- read_excel("../02_raw_data/ledger/CLEAN 54425 - WEBSITE EXPENSE.xlsx")
ledger_04 <- read_excel("../02_raw_data/ledger/CLEAN 54430 - BRANDING EXPENSE.xlsx")
ledger_05 <- read_excel("../02_raw_data/ledger/CLEAN 54435 - PUBLIC RELATIONS.xlsx")
ledger_06 <- read_excel("../02_raw_data/ledger/CLEAN 54455 - TENANT RELATIONS.xlsx")
ledger_07 <- read_excel("../02_raw_data/ledger/CLEAN 54465 - BROCHURE AND BUSINESS CARDS.xlsx")
ledger_08 <- read_excel("../02_raw_data/ledger/CLEAN 54475 - DUES MEMBERSHIPS SUBSCRIPTIONS.xlsx")
ledger_09 <- read_excel("../02_raw_data/ledger/CLEAN 54480 - INTERNET ADVERTISING.xlsx")
ledger_10 <- read_excel("../02_raw_data/ledger/CLEAN 54481 - INTERNET LISTING SERVICES.xlsx")
ledger_11 <- read_excel("../02_raw_data/ledger/CLEAN 54482 - SEO SERVICES.xlsx")
ledger_12 <- read_excel("../02_raw_data/ledger/CLEAN 54485 - SIGN POSTERS OTHER.xlsx")
ledger_13 <- read_excel("../02_raw_data/ledger/CLEAN 54490 - MODEL EXPENSE.xlsx")
ledger_14 <- read_excel("../02_raw_data/ledger/CLEAN 54495 - REFERRAL COMMISSIONS.xlsx")
```

```
ledger_01 <- ledger_01 %>% mutate(ledgerAccount = "Mailings")
ledger_02 <- ledger_02 %>% mutate(ledgerAccount = "Marketing")
ledger_03 <- ledger_03 %>% mutate(ledgerAccount = "Website Expense")
ledger_04 <- ledger_04 %>% mutate(ledgerAccount = "Branding Expense")
ledger_05 <- ledger_05 %>% mutate(ledgerAccount = "Public Relations")
ledger_06 <- ledger_06 %>% mutate(ledgerAccount = "Tenant Relations")
ledger_07 <- ledger_07 %>% mutate(ledgerAccount = "Brochure")
ledger_08 <- ledger_08 %>% mutate(ledgerAccount = "Dues Memberships")
ledger_09 <- ledger_09 %>% mutate(ledgerAccount = "Internet Advertising")
ledger_10 <- ledger_10 %>% mutate(ledgerAccount = "Internet Listing")
ledger_11 <- ledger_11 %>% mutate(ledgerAccount = "SEO Services")
ledger_12 <- ledger_12 %>% mutate(ledgerAccount = "Signs Posters")
ledger_13 <- ledger_13 %>% mutate(ledgerAccount = "Model Expense")
ledger_14 <- ledger_14 %>% mutate(ledgerAccount = "Referral Comissions")
```

Read-in Traffic Reports

- True spreadsheet names are omitted and updated with pseudonyms from final technical appendix to provide anonymity of TPS' properties

```

bluejay <- read_excel("../02_raw_data/trafficPseudonym/bluejay.xlsx")
cardinal <- read_excel("../02_raw_data/trafficPseudonym/cardinal.xlsx")
chicken <- read_excel("../02_raw_data/trafficPseudonym/chicken.xlsx")
condor <- read_excel("../02_raw_data/trafficPseudonym/condor.xlsx")
crow <- read_excel("../02_raw_data/trafficPseudonym/crow.xlsx")
eagle <- read_excel("../02_raw_data/trafficPseudonym/eagle.xlsx")
goldfinch <- read_excel("../02_raw_data/trafficPseudonym/goldfinch.xlsx")
falcon <- read_excel("../02_raw_data/trafficPseudonym/falcon.xlsx")
flamingo <- read_excel("../02_raw_data/trafficPseudonym/flamingo.xlsx")
hummingbird <- read_excel("../02_raw_data/trafficPseudonym/hummingbird.xlsx")
mockingbird <- read_excel("../02_raw_data/trafficPseudonym/mockingbird.xlsx")
osprey <- read_excel("../02_raw_data/trafficPseudonym/osprey.xlsx")
parrot <- read_excel("../02_raw_data/trafficPseudonym/parrot.xlsx")
peacock <- read_excel("../02_raw_data/trafficPseudonym/peacock.xlsx")
pelican <- read_excel("../02_raw_data/trafficPseudonym/pelican.xlsx")
penguin <- read_excel("../02_raw_data/trafficPseudonym/penguin.xlsx")
raven <- read_excel("../02_raw_data/trafficPseudonym/raven.xlsx")
redhawk <- read_excel("../02_raw_data/trafficPseudonym/redhawk.xlsx")
robin <- read_excel("../02_raw_data/trafficPseudonym/robin.xlsx")
seagull <- read_excel("../02_raw_data/trafficPseudonym/seagull.xlsx")
sparrow <- read_excel("../02_raw_data/trafficPseudonym/sparrow.xlsx")
swan <- read_excel("../02_raw_data/trafficPseudonym/swan.xlsx")
vulture <- read_excel("../02_raw_data/trafficPseudonym/vulture.xlsx")

```

Joining Data

Combine Ledger Reports

```

# pipeline to join all reports for base_1 datasets
ledger_base <- bind_rows(x = ledger_01, y = ledger_02)
ledger_base <- bind_rows(x = ledger_base, y = ledger_03)
ledger_base <- bind_rows(x = ledger_base, y = ledger_04)
ledger_base <- bind_rows(x = ledger_base, y = ledger_05)
ledger_base <- bind_rows(x = ledger_base, y = ledger_06)
ledger_base <- bind_rows(x = ledger_base, y = ledger_07)
ledger_base <- bind_rows(x = ledger_base, y = ledger_08)
ledger_base <- bind_rows(x = ledger_base, y = ledger_09)
ledger_base <- bind_rows(x = ledger_base, y = ledger_10)
ledger_base <- bind_rows(x = ledger_base, y = ledger_11)
ledger_base <- bind_rows(x = ledger_base, y = ledger_12)
ledger_base <- bind_rows(x = ledger_base, y = ledger_13)
ledger_base <- bind_rows(x = ledger_base, y = ledger_14)

```

Combine Traffic Reports

```

trafficReportList <- bind_rows(bluejay, cardinal, chicken, condor, crow, eagle,
                               goldfinch, falcon, flamingo, hummingbird, mockingbird, osprey,
                               parrot, peacock, pelican, penguin, raven, redhawk, robin,
                               seagull, sparrow, swan, vulture)

```

Data Cleanup

Rename Sources in TrafficReportList

```

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% c("Apartment list", "Apartment List") ~ "Apartmentlist.com"
    ,TRUE ~ Source
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% "apartments.com" ~ "Apartments.com"
    ,TRUE ~ Source
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% "Craigslist-Basic Ad" ~ "Craigslist"
    ,TRUE ~ Source
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% c("Drive By/ Signage", "Driveby") ~ "Drive-by/Signage"
    ,TRUE ~ Source
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% "apartments.com" ~ "Apartments.com"
    ,TRUE ~ Source
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% c("Google Ads", "Google My Business", "Google PayPerClick (PPC)",
      "Google PayPerClick(PPC)", "Google Plus") ~ "Google"
    ,TRUE ~ Source
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% c("Other", "Other/Online",
      "Other/Walk in", "Other/Walk-in") ~ "Other"
    ,TRUE ~ Source
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% c("Zillow Network", "Zillow/Hotpads") ~ "Zillow"
    ,TRUE ~ Source
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% c("Referral - Resident", "Referral- Resident",
      "Referrals", "Resident Referral", "Employee Referral",
      "Referral - Family Member") ~ "Referral"
    ,TRUE ~ Source
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% c("Property website", "portal") ~ "Property Website"
    ,TRUE ~ Source
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Source = case_when(
    Source %in% c("Other tarragon property", "Tarragon Website", "Corporate Website") ~ "Corporate Website"
    ,TRUE ~ Source
  )
)

```

```

)
)
table(trafficReportList$Source)

```

```

##
##      A-Frame Apartmentlist.com Apartments.com Applicant Grouping
##      116      24573      50      5
##      Banner and Sign      Brochure Corporate Website      CoStar Group
##      171      17      153      5598
##      Craigslist Drive-by/Signage      Facebook Googel My Business
##      382      1274      102      3
##      Google      MobileApp      Office      Other
##      24857      3      579      1034
##      Outreach Flyer      Palermo      Portal Promote Roommate
##      3      2      1      17
##      Property Website      Referral      Rent.com RENTCafe.com ILS
##      9951      376      12      636
##      RentPath      Reply      TPS website      Transfer Unit
##      64      327      2      6
##      Zillow Zumper/PadMapper
##      18723      13

```

Rename Status and Result/Reason in TrafficReportList

```

trafficReportList <- trafficReportList %>% rename("Result" = "Result/ Reason")

trafficReportList <- trafficReportList %>%
  mutate(Status = case_when(
    Status %in% c("Canceled Guest", "CanceledGuest") ~ "Canceled"
    ,TRUE ~ Status
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Result = case_when(
    Result %in% "Apartment list" ~ "Apartment List",
    TRUE ~ Result
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Result = case_when(
    Result %in% "Applicant grouping" ~ "Applicant Grouping",
    TRUE ~ Result
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Result = case_when(
    Result %in% c("Created by Lead2Lease (Unqualified)",
      "Criminal History (Unqualified)",
      "Does not credit qualify (Unqualified)",
      "Does not income qualify (Unqualified)",
      "Found another apartment (Unqualified)",
      "Not moving (Unqualified)", "Personal (Unqualified)",
      "Pet (Unqualified)", "Screening process (Unqualified)",
      "Screening Process (Unqualified)",
      "Still looking / Undecided (Unqualified)",
      "Timeframe Change (Unqualified)",
      "Too many occupants (Unqualified)",
      "Unqualified (Unqualified)") ~ "Unqualified",
    TRUE ~ Result
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Result = case_when(
    Result %in% c("Desired floorplan not avail", "Desired floorplan not availabl",
      "Desired Floorplan not availabl") ~ "Desired Floorplan not avail"
    ,TRUE ~ Result
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Result = case_when(
    Result %in% c("Desired mone-in date not avail",
      "Desired move in date not avail",
      "Desired move in date unavail",
      "Desired Move in date unavail",
      "Desired move-in date not avail",
      "Desired Move-in date not avail",
      "Desired Move-in Date not avail") ~ "Desired Move in date not avail"
    ,TRUE ~ Result
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Result = case_when(
    Result %in% "Found another apartment" ~ "Found another Apartment"
    ,TRUE ~ Result
  )
)

trafficReportList <- trafficReportList %>%
  mutate(Result = case_when(
    Result %in% c("Future / 30 days out", "Future / 30 Days out",
      "Future / 30-days out", "Future/30 days out") ~ "Future / 30-Days Out"
    ,TRUE ~ Result
  )
)

trafficReportList <- trafficReportList %>%

```

```

mutate(Result = case_when(
  Result %in% "Leased an Apartment" ~ "Leased"
,TRUE ~ Result
)
)

trafficReportList <- trafficReportList %>%
mutate(Result = case_when(
  Result %in% "Not moving" ~ "Not Moving"
,TRUE ~ Result
)
)

trafficReportList <- trafficReportList %>%
mutate(Result = case_when(
  Result %in% "Promote Roommate" ~ "Roommate"
,TRUE ~ Result
)
)

trafficReportList <- trafficReportList %>%
mutate(Result = case_when(
  Result %in% "Screening process" ~ "Screening Process"
,TRUE ~ Result
)
)

trafficReportList <- trafficReportList %>%
mutate(Result = case_when(
  Result %in% c("Still looking / Undecided", "Still looking/ Undecided",
    "Still Looking/ Undecided", "Still Looking/Undecided ",
    "StillLooking/ Undecided", "Still Looking/Undecided") ~ "Still Looking / Undecided"
,TRUE ~ Result
)
)

trafficReportList <- trafficReportList %>%
mutate(Result = case_when(
  Result %in% "Timeframe change" ~ "Timeframe Change"
,TRUE ~ Result
)
)

trafficReportList$Result[is.na(trafficReportList$Result)] <- "Unknown"

table(trafficReportList$Result)

```

```

##
##           Apartment List           Applicant Grouping
##           11717           113
##   Desired Floorplan not avail   Desired Move in date not avail
##           823           1395
##           Duplicate           Found another Apartment
##           1843           1816
##   Future / 30-Days Out           House did not close
##           79           1
##           Invalid MFTE           Leased
##           52           2060
##   Level One Notes           Marriage/Divorce
##           670           1
##           Not Moving           Nothing
##           1383           3181
##   Online Lead           Personal
##           2547           2243
##           Price           Roommate
##           292           26
##   Screening Process           Set Appointment
##           7           5
##   Still Looking / Undecided           Timeframe Change
##           1615           32
##           Took Application           Unknown
##           756           55725
##           Unqualified           Will Return
##           377           455

```

Wrangle data

```
trafficReportList <- trafficReportList %>%
  mutate("Status" = as.factor(Status)) %>%
  mutate("Source" = as.factor(Source)) %>%
  mutate("Result" = as.factor(Result))

# this is resulting in NA's
# Can't change 'Event Date' to date type. Need help on this.
# as.Date("Event Date", format = "%mm/%dd/%YYYY")

# recode Status vector
trafficReportList <- trafficReportList %>%
  mutate(num_status = case_when(trafficReportList$Status == 'Resident' ~ 1,
                                trafficReportList$Status == 'Approved' ~ 1,
                                trafficReportList$Status == 'Prospect' ~ 0,
                                trafficReportList$Status == 'Denied' ~ 0,
                                trafficReportList$Status == 'CanceledGuest' ~ 0,
                                trafficReportList$Status == 'Canceled Guest' ~ 0,
                                trafficReportList$Status == 'Canceled' ~ 0,
                                trafficReportList$Status == 'Applied' ~ 0))

# filter 'Source' to top hits
trafficReportList_top_source_result <- trafficReportList %>%
  filter(Source %in% c("Apartmentlist.com", "CoStar Group", 'Drive-by/Signage',
                      'Google', 'Office', 'Property Website',
                      'RENTCAFE.com ILS', 'Zillow')) %>%
  filter(Result %in% c("Apartment List", "Desired move-in date not avail",
                      "Duplicate", "Found another apartment",
                      "Leased", "Not Moving", "Nothing", "Online Lead",
                      "Personal", "Still looking / Undecided", "Took Application", "Unknown"))
```

##Adding Sources to the Ledger_base

```
write_xlsx(ledger_base, "../02_raw_data/LedgerSources/ledger_base")

sourceList <- unique(trafficReportList$Source)

source_df <- data_frame(sourceList)
write_xlsx(source_df, "../02_raw_data/LedgerSources/sourceList")

ledger_baseSource <- read.csv("../02_raw_data/LedgerSources/ledger_base_withsourcesCSV.csv")
```

```
ledger_baseDates <- ledger_baseSource

ledger_baseDates <- ledger_baseDates %>%
  mutate('Year' = (substring(ledger_baseDates$'Date', 0, 4))) %>%
  mutate('Month' = (substring(ledger_baseDates$'Date', 6, 7))) ##

ledger_baseDates <- ledger_baseDates %>%
  mutate('Season' = case_when(ledger_baseDates$Month == '01' ~ 'Q1-Winter',
                              ledger_baseDates$Month == '02' ~ 'Q1-Winter',
                              ledger_baseDates$Month == '03' ~ 'Q1-Winter',
                              ledger_baseDates$Month == '04' ~ 'Q2-Spring',
                              ledger_baseDates$Month == '05' ~ 'Q2-Spring',
                              ledger_baseDates$Month == '06' ~ 'Q2-Spring',
                              ledger_baseDates$Month == '07' ~ 'Q3-Summer',
                              ledger_baseDates$Month == '08' ~ 'Q3-Summer',
                              ledger_baseDates$Month == '09' ~ 'Q3-Summer',
                              ledger_baseDates$Month == '10' ~ 'Q4-Fall',
                              ledger_baseDates$Month == '11' ~ 'Q4-Fall',
                              ledger_baseDates$Month == '12' ~ 'Q4-Fall'
                              ))
```

```

trafficReportDates <- trafficReportList %>%
  mutate('Year' = (substring(trafficReportList$'Event Date', 7, 11))) %>%
  mutate('Month' = (substring(trafficReportList$'Event Date', 1, 2)))

trafficReportDates <- trafficReportDates %>%
  mutate('Season' = case_when(trafficReportDates$Month == '01' ~ 'Q1-Winter',
                              trafficReportDates$Month == '02' ~ 'Q1-Winter',
                              trafficReportDates$Month == '03' ~ 'Q1-Winter',
                              trafficReportDates$Month == '04' ~ 'Q2-Spring',
                              trafficReportDates$Month == '05' ~ 'Q2-Spring',
                              trafficReportDates$Month == '06' ~ 'Q2-Spring',
                              trafficReportDates$Month == '07' ~ 'Q3-Summer',
                              trafficReportDates$Month == '08' ~ 'Q3-Summer',
                              trafficReportDates$Month == '09' ~ 'Q3-Summer',
                              trafficReportDates$Month == '10' ~ 'Q4-Fall',
                              trafficReportDates$Month == '11' ~ 'Q4-Fall',
                              trafficReportDates$Month == '12' ~ 'Q4-Fall'
  ))

```

Base EDA Step 1: Uni-variate non-graphical EDA

```

# top of the dataset and look each subset
head(trafficReportList)

```

```

## # A tibble: 6 x 9
##   `Property Name` `Unit type` Unit   `Applicant/Pros...` `Event Date` Source Status
##   <chr>          <chr>      <chr> <chr>          <chr>      <fct>  <fct>
## 1 aspen         asplxlre <NA>  p0249995      07/29/2021 Google Resid...
## 2 aspen         asplxlre <NA>  p0118408      07/24/2021 Prope... Cance...
## 3 aspen         asplxlre A-03   p0250359      08/18/2021 Google Resid...
## 4 aspen         asplxlre B-07   p0247479      07/14/2021 Google Cance...
## 5 aspen         asplxlre B-07   p0247479      07/17/2021 Google Cance...
## 6 aspen         asplxlre D-05   p0272657      04/17/2022 Google Resid...
## # ... with 2 more variables: Result <fct>, num_status <dbl>

```

```

# check variable table
vtable(trafficReportList)

```

trafficReportList

Name	Class	Values
Property Name	character	
Unit type	character	
Unit	character	
Applicant/Prospect	character	
Event Date	character	
Source	factor	'A-Frame' 'Apartmentlist.com' 'Apartments.com' 'Applicant Grouping' 'Banner and Sign' and more
Status	factor	'Applied' 'Approved' 'Canceled' 'Denied' 'Prospect' and more
Result	factor	'Apartment List' 'Applicant Grouping' 'Desired Floorplan not avail' 'Desired Move in date not avail' 'Duplicate' and more
num_status	numeric	Num: 0 to 1

```
summary(trafficReportList)
```



```
## Property Name      Unit type      Unit      Applicant/Prospect
## Length:89214      Length:89214      Length:89214      Length:89214
## Class :character   Class :character   Class :character   Class :character
## Mode :character    Mode :character    Mode :character    Mode :character
##
##
##
## Event Date          Source          Status
## Length:89214        Google          :24857      Applied : 303
## Class :character     Apartmentlist.com:24573      Approved: 371
## Mode :character      Zillow          :18723      Canceled:58738
##                      Property Website : 9951      Denied : 1677
##                      CoStar Group   : 5598      Prospect:16339
##                      (Other)        : 5348      Resident:11757
##                      NA's          : 164       NA's : 29
##
## Result      num_status
## Unknown      :55725      Min. :0.000
## Apartment List:11717      1st Qu.:0.000
## Nothing      : 3181      Median :0.000
## Online Lead   : 2547      Mean :0.136
## Personal      : 2243      3rd Qu.:0.000
## Leased        : 2060      Max. :1.000
## (Other)       :11741      NA's :29
```

Base EDA Step 2: Uni-variate graphical EDA - Categorical

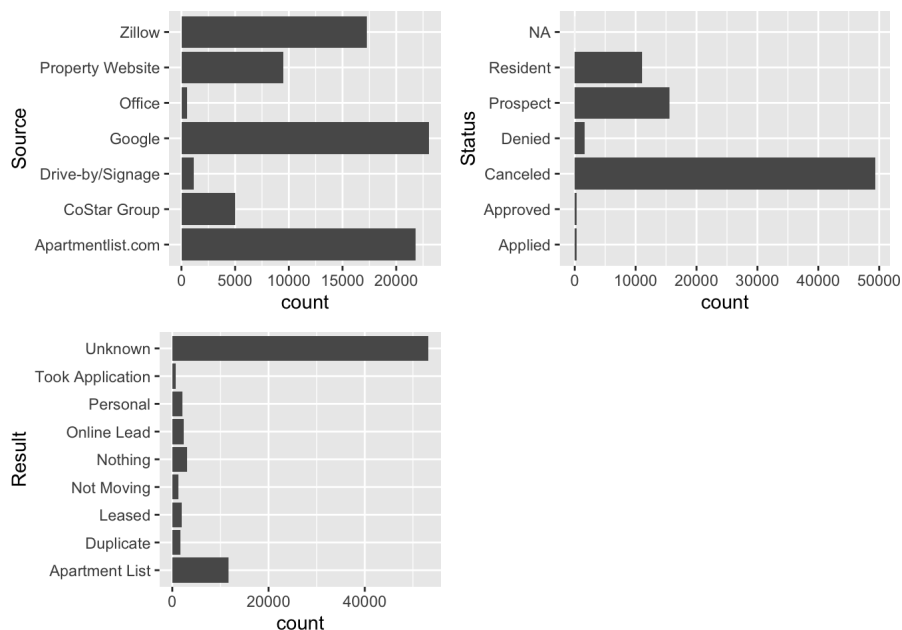
```
# Only use bar graphs (like histogram) because nothing like boxplot for categorical variables
grid.arrange(

# Source
ggplot(data = trafficReportList_top_source_result, mapping = aes(x = Source)) +
  geom_bar() +
  coord_flip(),

# Status
ggplot(data = trafficReportList_top_source_result, mapping = aes(x = Status)) +
  geom_bar() +
  coord_flip(),

# Result/ Reason
ggplot(data = trafficReportList_top_source_result, mapping = aes(x = Result)) +
  geom_bar() +
  coord_flip(),

ncol = 2 )
```



```

trafficReportList %>%
  tabyl(Source) %>% # creates table of counts
  adorn_totals(where = c("row", "col")) %>% # Total margins
  adorn_percentages(denominator = "all") %>% # creates proportions
  adorn_rounding(2)

```

```

##           Source      n percent valid_percent Total
##           A-Frame 0.00      0              0 0.00
## Apartmentlist.com 0.28      0              0 0.28
## Apartments.com 0.00      0              0 0.00
## Applicant Grouping 0.00      0              0 0.00
## Banner and Sign 0.00      0              0 0.00
## Brochure 0.00      0              0 0.00
## Corporate Website 0.00      0              0 0.00
## CoStar Group 0.06      0              0 0.06
## Craigslist 0.00      0              0 0.00
## Drive-by/Signage 0.01      0              0 0.01
## Facebook 0.00      0              0 0.00
## Googel My Business 0.00      0              0 0.00
## Google 0.28      0              0 0.28
## MobileApp 0.00      0              0 0.00
## Office 0.01      0              0 0.01
## Other 0.01      0              0 0.01
## Outreach Flyer 0.00      0              0 0.00
## Palermo 0.00      0              0 0.00
## Portal 0.00      0              0 0.00
## Promote Roommate 0.00      0              0 0.00
## Property Website 0.11      0              0 0.11
## Referral 0.00      0              0 0.00
## Rent.com 0.00      0              0 0.00
## RENTCafe.com ILS 0.01      0              0 0.01
## RentPath 0.00      0              0 0.00
## Reply 0.00      0              0 0.00
## TPS website 0.00      0              0 0.00
## Transfer Unit 0.00      0              0 0.00
## Zillow 0.21      0              0 0.21
## Zumper/PadMapper 0.00      0              0 0.00
## <NA> 0.00      0              NA 0.00
## Total 1.00      0              0 1.00

```

```

# find highest factors in 'Source' and 'Result/Reason'

```

```

trafficReportList %>%
  tabyl(Result) %>% # creates table of counts
  adorn_totals(where = c("row", "col")) %>% # Total margins
  adorn_percentages(denominator = "all") %>% # creates proportions
  adorn_rounding(2)

```

```

##           Result      n percent Total
## Apartment List 0.13      0 0.13
## Applicant Grouping 0.00      0 0.00
## Desired Floorplan not avail 0.01      0 0.01
## Desired Move in date not avail 0.02      0 0.02
## Duplicate 0.02      0 0.02
## Found another Apartment 0.02      0 0.02
## Future / 30-Days Out 0.00      0 0.00
## House did not close 0.00      0 0.00
## Invalid MFTE 0.00      0 0.00
## Leased 0.02      0 0.02
## Level One Notes 0.01      0 0.01
## Marriage/Divorce 0.00      0 0.00
## Not Moving 0.02      0 0.02
## Nothing 0.04      0 0.04
## Online Lead 0.03      0 0.03
## Personal 0.03      0 0.03
## Price 0.00      0 0.00
## Roommate 0.00      0 0.00
## Screening Process 0.00      0 0.00
## Set Appointment 0.00      0 0.00
## Still Looking / Undecided 0.02      0 0.02
## Timeframe Change 0.00      0 0.00
## Took Application 0.01      0 0.01
## Unknown 0.62      0 0.62
## Unqualified 0.00      0 0.00
## Will Return 0.01      0 0.01
## Total 1.00      0 1.00

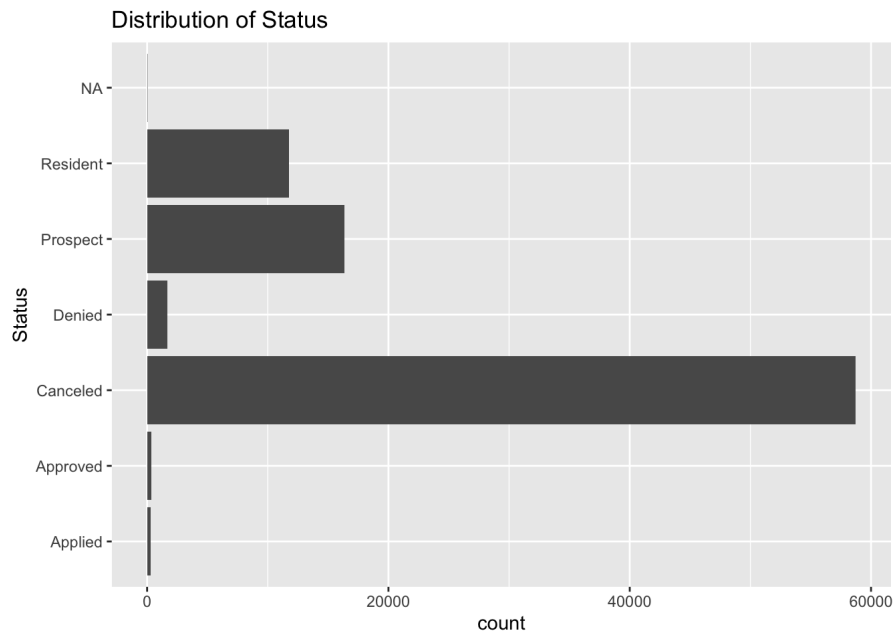
```

```
cols <- c("Joined" = "#37c871", "Not Joined" = "#ff0000")

p <- trafficReportList %>%
  ggplot(mapping = aes(x = `Status`))

p + geom_histogram(bins = 10, position = "dodge", stat = "count") +
  scale_fill_manual(values = cols) +

  #aesthetics = "colour") +
  #scale_fill_manual(c("green", "red")) +
  labs(title = "Distribution of Status") +
  theme(legend.position = "right") +
  coord_flip()
```



Marketing Metrics

Grouping on Ledger by Source

```
ledgerSourceGroups <- ledger_baseSource %>%
  select(Source) %>%
  tabyl(Source)
```

Grouping on the Traffic reports by source

```
trafficSourceGroups <- trafficReportList %>%
  select(Source) %>%
  group_by(Source) %>%
  count(Source)

summary(trafficSourceGroups)
```

```
##           Source           n
## A-Frame           : 1   Min.    :    1.0
## Apartmentlist.com : 1   1st Qu.:    9.0
## Apartments.com     : 1   Median : 116.0
## Applicant Grouping: 1   Mean    : 2877.9
## Banner and Sign    : 1   3rd Qu.: 607.5
## (Other)            :25   Max.    :24857.0
## NA's              : 1
```

Grouping on the Traffic reports data by Source and Status

```
trafficReportList %>%
  select(Source, Status) %>%
  tabyl(Source, Status)
```

##	Source	Applied	Approved	Canceled	Denied	Prospect	Resident	NA
##	A-Frame	0	0	62	14	3	37	0
##	Apartmentlist.com	17	12	18965	74	5001	504	0
##	Apartments.com	0	0	43	0	3	1	3
##	Applicant Grouping	0	0	0	0	0	5	0
##	Banner and Sign	0	3	119	11	15	23	0
##	Brochure	0	0	16	0	1	0	0
##	Corporate Website	0	0	76	0	28	49	0
##	CoStar Group	6	25	3430	115	1263	759	0
##	Craigslist	0	3	245	5	60	69	0
##	Drive-by/Signage	3	0	918	23	126	204	0
##	Facebook	0	0	93	0	2	7	0
##	Googel My Business	0	0	3	0	0	0	0
##	Google	132	166	14181	783	4213	5370	12
##	MobileApp	0	0	3	0	0	0	0
##	Office	0	0	319	37	115	108	0
##	Other	10	5	609	0	186	224	0
##	Outreach Flyer	0	0	3	0	0	0	0
##	Palermo	0	0	2	0	0	0	0
##	Portal	0	0	1	0	0	0	0
##	Promote Roommate	0	0	1	0	0	16	0
##	Property Website	94	127	5245	403	1081	3001	0
##	Referral	0	0	218	6	49	103	0
##	Rent.com	0	0	12	0	0	0	0
##	RENTCafe.com ILS	0	0	339	0	257	40	0
##	RentPath	0	0	61	0	1	2	0
##	Reply	0	7	177	12	43	88	0
##	TPS website	0	0	0	0	2	0	0
##	Transfer Unit	0	0	1	0	0	5	0
##	Zillow	41	23	13440	194	3882	1142	1
##	Zumper/PadMapper	0	0	5	0	8	0	0
##	<NA>	0	0	151	0	0	0	13

Grouping on the Traffic reports data by Year and Season

```
trafficReportDates %>%
  select(Year) %>%
  tabyl(Year)
```

```
## Year      n    percent
## 2021 37237 0.4173896
## 2022 51977 0.5826104
```

```
trafficReportDates %>%
  select(Season) %>%
  tabyl(Season)
```

```
## Season      n    percent
## Q1-Winter 20873 0.2339655
## Q2-Spring 31104 0.3486448
## Q3-Summer 20273 0.2272401
## Q4-Fall 16964 0.1901495
```

Grouping on the Traffic Property data

```
## Joining Traffic Reports Data with Property List
```

```
trafficProperty <- trafficReportList %>%
  left_join(property_list, by = c('Property Name' = 'Yardi Id Code'))
```

```
trafficProperty %>%
  select('PROPERTY TYPE:') %>%
  tabyl('PROPERTY TYPE:')
```

```
## PROPERTY TYPE:      n    percent valid_percent
## Garden 65768 0.73719371      0.7719974
## High-rise 10558 0.11834465      0.1239318
## Mid-rise 8866 0.09937902      0.1040708
## <NA> 4022 0.04508261      NA
```

```
trafficProperty %>%
  select('LOCATION:') %>%
  tabyl('LOCATION:')
```

```
## LOCATION:      n      percent valid_percent
##      Rural  4925  0.05520434      0.05781059
##      Suburban 60923 0.68288609      0.71512583
##      Urban  19344 0.21682696      0.22706357
##      <NA>   4022  0.04508261           NA
```

```
trafficProperty %>%
  select('COUNTY:') %>%
  tabyl('COUNTY:')
```

```
## COUNTY:      n      percent valid_percent
##      KING 30963 0.34706436      0.3634496
##      PIERCE 42314 0.47429776      0.4966898
##      THURSTON 11915 0.13355527      0.1398606
##      <NA>   4022  0.04508261           NA
```

Cost per Lead Analysis

```
GLSourceGrouping <- sqldf("
  select Source, count(Source) countSource,
  sum(Debit) sumDebit, sum(Credit) sumCredit,
  (sum(Debit) - sum(Credit)) TotalSpending
  from ledger_baseSource
  group by Source
  ")
```

```
kable(GLSourceGrouping)
```

Source	countSource	sumDebit	sumCredit	TotalSpending
	2682369	585.21	110399.3	259185.91
Apartmentlist.com	178	74931.50	0.0	74931.50
Apartments.com	26411	6612.62	0.0	116612.62
Comcast	2	640.14	0.0	640.14
Drive-by/Signage	8	2047.02	0.0	2047.02
Elise	6	24076.74	0.0	24076.74
Google	126	65232.64	0.0	65232.64
Marketing	325	85528.06	13891.5	71636.56
Matterport	100	519.24	0.0	519.24
Referral	68	11595.00	100.0	11495.00
SightMap	3	346.50	0.0	346.50
Sightmap	1	115.50	0.0	115.50
The Daily	1	223.50	0.0	223.50
Zillow	94	69500.00	0.0	69500.00

```
TRSourceGrouping <- sqldf("
  select *, (case when sq.ResidentStatusCount = 0 then 0
    else cast(sq.ResidentStatusCount as float)/
    cast(sq.CountLead as float)
  end
  ) ResidentRate
  from
    (select Source, count(Source) countLead,
    (select count(trp2.num_status)
    from trafficReportList trp2
    where trp2.num_status = 1 and trp2.Source = trp1.Source )
    from trafficReportList trp1
    group by Source) sq
  ")
kable(TRSourceGrouping)
```

Source	countLead	ResidentStatusCount	ResidentRate
NA	0	0	0.0000000
A-Frame	116	37	0.3189655
Apartmentlist.com	24573	516	0.0209987
Apartments.com	50	1	0.0200000
Applicant Grouping	5	5	1.0000000
Banner and Sign	171	26	0.1520468
Brochure	17	0	0.0000000

Source	countLead	ResidentStatusCount	ResidentRate
CoStar Group	5598	784	0.1400500
Corporate Website	153	49	0.3202614
Craigslist	382	72	0.1884817
Drive-by/Signage	1274	204	0.1601256
Facebook	102	7	0.0686275
Googel My Business	3	0	0.0000000
Google	24857	5536	0.2227139
MobileApp	3	0	0.0000000
Office	579	108	0.1865285
Other	1034	229	0.2214700
Outreach Flyer	3	0	0.0000000
Palermo	2	0	0.0000000
Portal	1	0	0.0000000
Promote Roommate	17	16	0.9411765
Property Website	9951	3128	0.3143403
RENTCafe.com ILS	636	40	0.0628931
Referral	376	103	0.2739362
Rent.com	12	0	0.0000000
RentPath	64	2	0.0312500
Reply	327	95	0.2905199
TPS website	2	0	0.0000000
Transfer Unit	6	5	0.8333333
Zillow	18723	1165	0.0622229
Zumper/PadMapper	13	0	0.0000000

Getting Cost per Lead

```
costPerLead <- GLSourceGrouping %>%
  inner_join(TRSourceGrouping, "Source") %>%
  mutate(costPerLead = TotalSpending/countLead,
         costForAllResidents = costPerLead * ResidentStatusCount,
         unallocatedBudget = TotalSpending - costForAllResidents) %>%
  select(Source, TotalSpending, countLead, ResidentStatusCount, ResidentRate, costPerLead, costForAllResidents, unallocatedBudget)

kable(costPerLead)
```

Source	TotalSpending	countLead	ResidentStatusCount	ResidentRate	costPerLead	costForAllResidents	unallocatedBudget
Apartmentlist.com	74931.50	24573	516	0.0209987	3.049343	1573.4609	73358.039
Apartments.com	116612.62	50	1	0.0200000	2332.252400	2332.2524	114280.368
Drive-by/Signage	2047.02	1274	204	0.1601256	1.606766	327.7803	1719.240
Google	65232.64	24857	5536	0.2227139	2.624317	14528.2172	50704.423
Referral	11495.00	376	103	0.2739362	30.571808	3148.8963	8346.104
Zillow	69500.00	18723	1165	0.0622229	3.712012	4324.4939	65175.506

Getting Cost per Lead by Year

```

GLYearGrouping <- sqldf("
  select Source, Year, count(Source) countSource,
  sum(Debit) sumDebit, sum(Credit) sumCredit,
  (sum(Debit) - sum(Credit)) DifferenceDebitCredit
  from ledger_baseDates
  group by Source, Year
")

TRYearGrouping <- sqldf("
  select *, (case when sq.ResidentStatusCount = 0 then 0
    else cast(sq.ResidentStatusCount as float)/
      cast(sq.CountLead as float)
    end
    ) ResidentRate
  from
    (select Source, Year, count(Source) countLead,
      (select count(trp2.num_status)
        from trafficReportDates trp2
        where trp2.num_status = 1 and trp2.Source = trp1.Source and trp2.Year = trp1.Year )
      ResidentStatusCount
      from trafficReportDates trp1
      group by Source, Year) sq
")

costPerYear <- GLYearGrouping %>%
  inner_join(TRYearGrouping, c("Source", "Year")) %>%
  mutate(costPerLead = DifferenceDebitCredit/countLead,
    costForAllResidents = costPerLead * ResidentStatusCount,
    unallocatedBudget = DifferenceDebitCredit - costForAllResidents) %>%
  rename(TotalSpending = DifferenceDebitCredit) %>%
  select(Source, Year, TotalSpending, countLead, ResidentStatusCount, ResidentRate, costPerLead, costForAllResidents, unallocatedBudget)

kable(costPerYear)

```

Source	Year	TotalSpending	countLead	ResidentStatusCount	ResidentRate	costPerLead	costForAllResidents	unallocatedBudget
Apartmentlist.com	2021	40914.00	8819	218	0.0247194	4.6393015	1011.36773	39902.6323
Apartmentlist.com	2022	34017.50	15754	298	0.0189158	2.1592929	643.46928	33374.0307
Apartments.com	2021	54340.01	21	0	0.00000002587	6.195238	0.00000	54340.0100
Apartments.com	2022	62272.61	29	1	0.03448282147	3.313793	2147.33138	60125.2786
Drive-by/Signage	2021	385.44	695	90	0.1294964	0.5545899	49.91309	335.5269
Drive-by/Signage	2022	1661.58	579	114	0.1968912	2.8697409	327.15047	1334.4295
Google	2021	40988.52	11308	2562	0.2265653	3.6247365	9286.57484	31701.9452
Google	2022	24244.12	13549	2974	0.2194996	1.7893660	5321.57450	18922.5455
Referral	2021	4400.00	222	64	0.2882883	19.8198198	1268.46847	3131.5315
Referral	2022	7095.00	154	39	0.2532468	46.0714286	1796.78571	5298.2143
Zillow	2021	39500.00	7707	506	0.0656546	5.1252108	2593.35669	36906.6433
Zillow	2022	30000.00	11016	659	0.0598221	2.7233115	1794.66231	28205.3377

Getting Cost per Lead by Season

```

GLSeasonGrouping <- sqldf("
  select Source, Season, count(Source) countSource,
  sum(Debit) sumDebit, sum(Credit) sumCredit,
  (sum(Debit) - sum(Credit)) DifferenceDebitCredit
  from ledger_baseDates
  group by Source, Season
")

TRSeasonGrouping <- sqldf("
  select *, (case when sq.ResidentStatusCount = 0 then 0
    else cast(sq.ResidentStatusCount as float)/
    cast(sq.CountLead as float)
    end
    ) ResidentRate
  from
    (select Source, Season, count(Source) countLead,
    (select count(trp2.num_status)
    from trafficReportDates trp2
    where trp2.num_status = 1 and trp2.Source = trp1.Source and trp2.Season =
    ResidentStatusCount
    from trafficReportDates trp1
    group by Source, Season) sq
    )

costPerSeason <- GLSeasonGrouping %>%
  inner_join(TRSeasonGrouping, c("Source", "Season")) %>%
  mutate(costPerLead = DifferenceDebitCredit/countLead,
    costForAllResidents = costPerLead * ResidentStatusCount,
    unallocatedBudget = DifferenceDebitCredit - costForAllResidents) %>%
  rename(TotalSpending = DifferenceDebitCredit) %>%
  select(Source, Season, TotalSpending, countLead, ResidentStatusCount, ResidentRate, costPerLead, costForAllResidents, unallocatedBudget)

kable(costPerSeason)

```

Source	Season	TotalSpending	countLead	ResidentStatusCount	ResidentRate	costPerLead	costForAllResidents	unallocatedBudget
Apartmentlist.com	Q1-Winter	15701.75	4236	178	0.0420208	3.706740	659.79969	15041.9503
Apartmentlist.com	Q2-Spring	21814.75	11518	120	0.0104185	1.893970	227.27644	21587.4736
Apartmentlist.com	Q3-Summer	23171.00	4870	108	0.0221766	4.757905	513.85380	22657.1462
Apartmentlist.com	Q4-Fall	14244.00	3949	110	0.0278552	3.606989	396.76880	13847.2312
Apartments.com	Q1-Winter	31374.62	10	0	0.0000000	3137.462000	0.00000	31374.6200
Apartments.com	Q2-Spring	30897.99	19	1	0.0526316	1626.210000	1626.21000	29271.7800
Apartments.com	Q3-Summer	27988.01	7	0	0.0000000	3998.287143	0.00000	27988.0100
Apartments.com	Q4-Fall	26352.00	14	0	0.0000000	1882.285714	0.00000	26352.0000
Drive-by/Signage	Q1-Winter	1287.09	269	61	0.2267658	4.784721	291.86799	995.2220
Drive-by/Signage	Q2-Spring	374.49	310	53	0.1709677	1.208032	64.02571	310.4643
Drive-by/Signage	Q3-Summer	385.44	375	48	0.1280000	1.027840	49.33632	336.1037
Google	Q1-Winter	18297.92	6571	1639	0.2494293	2.784648	4564.03757	13733.8824
Google	Q2-Spring	8687.57	6978	1335	0.1913156	1.244994	1662.06735	7025.5027
Google	Q3-Summer	8864.69	6067	1358	0.2238339	1.461132	1984.21774	6880.4723
Google	Q4-Fall	29382.46	5241	1204	0.2297272	5.606270	6749.94883	22632.5112
Referral	Q1-Winter	3699.00	81	26	0.3209877	45.666667	1187.33333	2511.6667
Referral	Q2-Spring	3396.00	73	13	0.1780822	46.520548	604.76712	2791.2329
Referral	Q3-Summer	3100.00	96	24	0.2500000	32.291667	775.00000	2325.0000
Referral	Q4-Fall	1300.00	126	40	0.3174603	10.317460	412.69841	887.3016
Zillow	Q1-Winter	13000.00	5054	398	0.0787495	2.572220	1023.74357	11976.2564
Zillow	Q2-Spring	24000.00	5962	261	0.0437773	4.025495	1050.65414	22949.3459

Source	Season	TotalSpending	countLeadResident	StatusCountResident	RatecostPerLead	costForAllResidents	unallocatedBudget
Zillow	Q3-Summer	14500.00	3952	252	0.0637652	3.669028	924.59514
Zillow	Q4-Fall	18000.00	3755	254	0.0676431	4.793609	1217.57656
							16782.4234

Getting Cost per Lead by Property Type

```

GLPropertyType <- left_join(ledger_baseSource, property_list, by = c("Property"= "Yardi Id Code"))

GLPropertyTypeGrouping <- sqldf("
  select [Property Type:] PropertyType ,count([Property Type:]) countPropertyType,
  sum(Debit) sumDebit, sum(Credit) sumCredit,
  (sum(Debit) - sum(Credit)) DifferenceDebitCredit
  from GLPropertyType
  group by [Property Type:]
")

TRPropertyTypeGrouping <- sqldf("
  select *, (case when sq.ResidentStatusCount = 0 then 0
    else cast(sq.ResidentStatusCount as float)/
    cast(sq.countLeadPropertyType as float)
  end
  ) ResidentRate
  from
    (select [Property Type:] PropertyType, count([Property Type:])
    countLeadPropertyType,
    (select count(tp2.num_status)
    from trafficProperty tp2
    where tp2.num_status = 1 and tp2.[Property Type:] = tp1.[Property Type:])
    ResidentStatusCount
    from trafficProperty tp1
    group by [Property Type:] sq
  )
")

costPerPropertyType <- GLPropertyTypeGrouping %>%
  inner_join(TRPropertyTypeGrouping, "PropertyType") %>%
  mutate(costPerLead = DifferenceDebitCredit/countLeadPropertyType,
    costForAllResidents = costPerLead * ResidentStatusCount,
    unallocatedBudget = DifferenceDebitCredit - costForAllResidents) %>%
  rename(TotalSpending = DifferenceDebitCredit, countLead = countLeadPropertyType) %>%
  select(PropertyType, TotalSpending, countLead, ResidentStatusCount, ResidentRate, costPerLead, costForAllResidents, unallocatedBudget)

kable(costPerPropertyType)

```

PropertyType	TotalSpending	countLeadResident	StatusCountResident	RatecostPerLead	costForAllResidents	unallocatedBudget
Garden	490577.65	65768	9994	0.1519584	7.459215	74547.394
High-rise	71404.44	10558	835	0.0790869	6.763065	5647.159
Mid-rise	134580.78	8866	799	0.0901196	15.179425	12128.360
						122452.42

Getting Cost per Lead by County

```

GLCountyGrouping <- sqldf("
  select [County:] County ,count([County:]) countCounty,
  sum(Debit) sumDebit, sum(Credit) sumCredit,
  (sum(Debit) - sum(Credit)) DifferenceDebitCredit
  from GLPropertyType
  group by [County:]
")

TRCountyGrouping <- sqldf("
  select *, (case when sq.ResidentStatusCount = 0 then 0
    else cast(sq.ResidentStatusCount as float)/
    cast(sq.countLeadCounty as float)
    end
    ) ResidentRate
  from
    (select [County:] County , count([County:]) countLeadCounty,
    (select count(tp2.num_status)
    from trafficProperty tp2
    where tp2.num_status = 1 and tp2.[County:] = tp1.[County:])
    from trafficProperty tp1
    group by [County:]) sq
  ResidentStatusCount
")

costPerCounty <- GLCountyGrouping %>%
  inner_join(TRCountyGrouping, "County") %>%
  mutate(costPerLead = DifferenceDebitCredit/countLeadCounty,
    costForAllResidents = costPerLead * ResidentStatusCount,
    unallocatedBudget = DifferenceDebitCredit - costForAllResidents)

kable(costPerCounty)

```

County	countCounty	sumDebit	sumCredit	DifferenceDebitCredit	countLeadCounty	ResidentStatusCount	ResidentRate	costPerLead	costForAllResidents
KING	1239291716.32	43737.55	247978.8	30963	3113	0.1005394	8.008874	24931.62	
PIERCE	2447476629.58	74662.98	401966.6	42314	7370	0.1741740	9.499612	70012.14	
THURSTON	172 52607.77	5990.27	46617.5	11915	1145	0.0960974	3.912505	4479.81	

Getting Cost per Lead by Location Type

```

GLLocationGrouping <- sqldf("
  select [Location:] Location ,count([Location:]) countLocation,
  sum(Debit) sumDebit, sum(Credit) sumCredit,
  (sum(Debit) - sum(Credit)) DifferenceDebitCredit
  from GLPropertyType
  group by [Location:]
")

TRLocationGrouping <- sqldf("
  select *, (case when sq.ResidentStatusCount = 0 then 0
    else cast(sq.ResidentStatusCount as float)/
    cast(sq.countLeadLocation as float)
    end
    ) ResidentRate
  from
    (select [Location:] Location , count([Location:]) countLeadLocation,
    (select count(tp2.num_status)
    from trafficProperty tp2
    where tp2.num_status = 1 and tp2.[Location:] = tp1.[Location:])
    from trafficProperty tp1
    group by [Location:]) sq
  ResidentStatusCount
")

costPerLocation <- GLLocationGrouping %>%
  inner_join(TRLocationGrouping, "Location") %>%
  mutate(costPerLead = DifferenceDebitCredit/countLeadLocation,
    costForAllResidents = costPerLead * ResidentStatusCount,
    unallocatedBudget = DifferenceDebitCredit - costForAllResidents) %>%
  rename(TotalSpending = DifferenceDebitCredit, countLead = countLeadLocation) %>%
  select(Location, TotalSpending, countLead, ResidentStatusCount, ResidentRate, costPerLead, costForAllResidents, unallocatedBudget)

kable(costPerLocation)

```

Location	TotalSpending	countLead	ResidentStatusCount	ResidentRate	costPerLead	costForAllResidents	unallocatedBudget
Rural	29242.0	4925	639	0.1297462	5.937462	3794.038	25447.96
Suburban	507202.0	60923	9383	0.1540141	8.325297	78116.259	429085.79
Urban	160118.8	19344	1606	0.0830232	8.277441	13293.570	146825.25

Cost per Lead Analysis by Property Characteristics

Getting Cost per Lead by Property Type and Source

```

GLPropertySourceGrouping <- sqldf("
  select [Property Type:] PropertyType, Source, ([Property Type:] || Source) PS,
  count(*) countPropertySource,
  sum(Debit) sumDebit, sum(Credit) sumCredit,
  (sum(Debit) - sum(Credit)) DifferenceDebitCredit
  from GLPropertyType
  group by [Property Type:], Source
")

TRPropertySourceGrouping <- sqldf("
  select *, (sq.PropertyType || sq.Source) PS,
  (case when sq.ResidentStatusCount = 0 then 0
    else cast(sq.ResidentStatusCount as float)/
      cast(sq.countLeadPropertySource as float)
    end
  ) ResidentRate
  from
    (select [Property Type:] PropertyType, Source,
    count(*) countLeadPropertySource,
    (select count(tp2.num_status)
    from trafficProperty tp2
    where tp2.num_status = 1 and tp2.[Property Type:] = tp1.[Property Type:] and tp2.Source = tp1.Source)
  ResidentStatusCount
    from trafficProperty tp1
    group by [Property Type:], Source) sq
")
costPerPropertySource <- GLPropertySourceGrouping %>%
  inner_join(TRPropertySourceGrouping, "PS") %>%
  mutate(costPerLead = DifferenceDebitCredit/countLeadPropertySource,
  costForAllResidents = costPerLead * ResidentStatusCount,
  unallocatedBudget = DifferenceDebitCredit - costForAllResidents) %>%
  rename(TotalSpending = DifferenceDebitCredit, countLead = countLeadPropertySource,
  PropertyType = PropertyType.x, Source = Source.x) %>%
  select(PropertyType, Source, TotalSpending, countLead, ResidentStatusCount, ResidentRate, costPerLead, costFor
  AllResidents, unallocatedBudget)

kable(costPerPropertySource)

```

PropertyType	Source	TotalSpending	countLead	ResidentStatusCount	ResidentRate	costPerLead	costForAllResidents	unallocatedBudget
Garden	Apartmentlist.com	59447.00	17232	373	0.0216458	3.449803	1286.77640	58160.2236
Garden	Apartments.com	62934.72	39	1	0.0256410	1613.710769	1613.71077	61321.0092
Garden	Drive-by/Signage	2047.02	1033	179	0.1732817	1.981626	354.71111	1692.3089
Garden	Google	48449.45	17957	4494	0.2502645	2.698081	12125.17839	36324.2716
Garden	Referral	10395.00	328	94	0.2865854	31.692073	2979.05488	7415.9451
Garden	Zillow	50000.00	14639	944	0.0644853	3.415534	3224.26395	46775.7360
High-rise	Apartmentlist.com	3150.75	2687	24	0.0089319	1.172590	28.14217	3122.6078
High-rise	Google	12837.15	3587	449	0.1251742	3.578798	1606.88050	11230.2695
High-rise	Referral	300.00	25	0	0.0000000	12.000000	0.00000	300.0000
High-rise	Zillow	5500.00	1828	39	0.0213348	3.008753	117.34136	5382.6586
Mid-rise	Apartmentlist.com	12333.75	3126	81	0.0259117	3.945537	319.58853	12014.1615
Mid-rise	Apartments.com	33817.90	9	0	0.0000000	3757.544444	0.00000	33817.9000
Mid-rise	Google	3946.04	1950	312	0.1600000	2.023610	631.36640	3314.6736
Mid-rise	Referral	800.00	12	5	0.4166667	66.666667	333.33333	466.6667
Mid-rise	Zillow	14000.00	1748	131	0.0749428	8.009153	1049.19908	12950.8009

Getting Cost per Lead by County and Source

```

GLCountySourceGrouping <- sqldf("
  select [County:] County, Source, ([County:] || Source) CS,
  count(*) countCountySource,
  sum(Debit) sumDebit, sum(Credit) sumCredit,
  (sum(Debit) - sum(Credit)) DifferenceDebitCredit
  from GLPropertyType
  group by [County:], Source
")
TRCountySourceGrouping <- sqldf("
  select *, (sq.County || sq.Source) CS,
  (case when sq.ResidentStatusCount = 0 then 0
    else cast(sq.ResidentStatusCount as float)/
      cast(sq.countLeadCountySource as float)
    end
  ) ResidentRate
  from
  (select [County:] County, Source,
  count(*) countLeadCountySource,
  (select count(tp2.num_status)
  from trafficProperty tp2
  where tp2.num_status = 1 and tp2.[County:] = tp1.[County:] and tp2.Source = tp1.Source)
  ResidentStatusCount
  from trafficProperty tp1
  group by [County:], Source) sq
")
costPerCountySource <- GLCountySourceGrouping %>%
  inner_join(TRCountySourceGrouping, "CS") %>%
  mutate(costPerLead = DifferenceDebitCredit/countLeadCountySource,
  costForAllResidents = costPerLead * ResidentStatusCount,
  unallocatedBudget = DifferenceDebitCredit - costForAllResidents) %>%
  rename(TotalSpending = DifferenceDebitCredit, countLead = countLeadCountySource,
  County = County.x, Source = Source.x) %>%
  select(County, Source, TotalSpending, countLead, ResidentStatusCount, ResidentRate, costPerLead, costForAllResidents, unallocatedBudget)
kable(costPerCountySource)

```

County	Source	TotalSpending	countLead	ResidentStatusCount	ResidentRate	costPerLead	costForAllResidents	unallocatedBudget
KING	Apartmentlist.com	15824.50	8090	120	0.0148331	1.956057	234.72682	15589.773
KING	Apartments.com	62509.20	7	0	0.0000000	8929.885714	0.00000	62509.200
KING	Google	20477.51	9270	1553	0.1675297	2.209009	3430.59040	17046.920
KING	Referral	2400.00	154	31	0.2012987	15.584416	483.11688	1916.883
KING	Zillow	20500.00	6064	285	0.0469987	3.380607	963.47296	19536.527
PIERCE	Apartmentlist.com	51007.00	10406	310	0.0297905	4.901691	1519.52431	49487.476
PIERCE	Apartments.com	48526.42	40	1	0.0250000	1213.160500	1213.16050	47313.260
PIERCE	Drive-by/Signage	2047.02	749	142	0.1895861	2.733004	388.08657	1658.933
PIERCE	Google	38585.25	11439	3193	0.2791328	3.373131	10770.40854	27814.841
PIERCE	Referral	8295.00	191	55	0.2879581	43.429319	2388.61257	5906.387
PIERCE	Zillow	45000.00	9702	676	0.0696764	4.638219	3135.43599	41864.564
THURSTON	Apartmentlist.com	8100.00	4549	48	0.0105518	1.780611	85.46933	8014.531
THURSTON	Apartments.com	5577.00	1	0	0.0000000	5577.000000	0.00000	5577.000
THURSTON	Google	6169.88	2785	509	0.1827648	2.215397	1127.63696	5042.243
THURSTON	Referral	800.00	20	13	0.6500000	40.000000	520.00000	280.000
THURSTON	Zillow	4000.00	2449	153	0.0624745	1.633320	249.89792	3750.102

Getting Cost per Lead by Location Type and Source

```

GLLocationSourceGrouping <- sqldf("
  select [Location:] Location, Source, ([Location:] || Source) LS,
  count(*) countLocationSource,
  sum(Debit) sumDebit, sum(Credit) sumCredit,
  (sum(Debit) - sum(Credit)) DifferenceDebitCredit
  from GLPropertyType
  group by [Location:], Source
")
TRLocationSourceGrouping <- sqldf("
  select *, (sq.Location || sq.Source) LS,
  (case when sq.ResidentStatusCount = 0 then 0
    else cast(sq.ResidentStatusCount as float)/
      cast(sq.countLeadLocationSource as float)
    end
  ) ResidentRate
  from
  (select [Location:] Location, Source,
  count(*) countLeadLocationSource,
  (select count(tp2.num_status)
  from trafficProperty tp2
  where tp2.num_status = 1 and tp2.[Location:] = tp1.[Location:] and tp2.Source = tp1.Source)
  ResidentStatusCount
  from trafficProperty tp1
  group by [Location:], Source) sq
")
costPerLocationSource <- GLLocationSourceGrouping %>%
  inner_join(TRLocationSourceGrouping, "LS") %>%
  mutate(costPerLead = DifferenceDebitCredit/countLeadLocationSource,
  costForAllResidents = costPerLead * ResidentStatusCount,
  unallocatedBudget = DifferenceDebitCredit - costForAllResidents) %>%
  rename(TotalSpending = DifferenceDebitCredit, countLead = countLeadLocationSource,
  Location = Location.x, Source = Source.x) %>%
  select(Location, Source, TotalSpending, countLead, ResidentStatusCount, ResidentRate, costPerLead, costForAllR
esidents, unallocatedBudget)
kable(costPerLocationSource)

```

Location	Source	TotalSpending	countLead	ResidentStatusCount	ResidentRate	costPerLead	costForAllResidents	unallocatedBudget
Rural	Apartmentlist.com	1396.00	332	0	0.00000004	2.04819e+00	0.0000	1396.0000
Rural	Apartments.com	5577.00	3	0	0.00000001	8.59000e+03	0.0000	5577.0000
Rural	Google	1360.33	1835	335	0.1825613	7.413243e-01	248.3436	1111.9864
Rural	Referral	500.00	43	0	0.00000001	1.162791e+01	0.0000	500.0000
Rural	Zillow	3000.00	1259	76	0.06036542	3.82843e+00	181.0961	2818.9039
Suburban	Apartmentlist.com	61201.00	16900	373	0.02207103	6.21361e+00	1350.7676	59850.2324
Suburban	Apartments.com	69677.42	41	1	0.02439021	6.99449e+03	1699.4493	67977.9707
Suburban	Drive-by/Signage	2047.02	891	153	0.17171722	2.97441e+00	351.5085	1695.5115
Suburban	Google	47508.31	16164	4170	0.25798072	9.39143e+00	12256.2270	35252.0830
Suburban	Referral	9895.00	285	94	0.32982463	4.71930e+01	3263.6140	6631.3860
Suburban	Zillow	52000.00	13386	868	0.06484393	8.84656e+00	3371.8811	48628.1189
Urban	Apartmentlist.com	12334.50	5813	105	0.01806302	1.21882e+00	222.7976	12111.7024
Urban	Apartments.com	41358.20	4	0	0.00000001	0.033955e+04	0.0000	41358.2000
Urban	Google	16364.00	5495	750	0.13648772	9.77980e+00	2233.4850	14130.5150
Urban	Referral	1100.00	37	5	0.13513512	9.72973e+01	148.6486	951.3514
Urban	Zillow	14500.00	3570	170	0.04761904	0.061625e+00	690.4762	13809.5238

Ratio of Status by Property Type

```

RatioStatus <- sqldf("
select *, cast(sq.CountCanceledLeads as float)/cast(sq.CountLeads as float) RatioStatus from
  (select count(*) CountLeads,
  (select count(*)
  from trafficProperty tp2
  where tp2.Status = 'Canceled'
  ) CountCanceledLeads
  from trafficProperty tp1
  ) sq
")
kable(RatioStatus)

```

CountLeads	CountCanceledLeads	RatioStatus
89214	58738	0.6583944

```
RatioStatusbyPropertyType <- sqldf("
select *, cast(sq.CountCanceledLeads as float)/cast(sq.CountLeads as float) RatioStatus from
  (select [Property Type:], count(*) CountLeads,
    (select count(*)
     from trafficProperty tp2
     where tp2.Status = 'Canceled' and tp2.[Property Type:] = tp1.[Property Type:]
     group by [Property Type:] ) CountCanceledLeads
   from trafficProperty tp1
   group by [Property Type:]) sq
")

kable(RatioStatusbyPropertyType)
```

Property Type:	CountLeads	CountCanceledLeads	RatioStatus
NA	4022	NA	NA
Garden	65768	48483	0.7371822
High-rise	10558	4317	0.4088843
Mid-rise	8866	4224	0.4764268

Regression Analysis

- Independent variable: num_status (if they signed a lease or not)
- Dependent variables: total spending (overall spending, by source, etc.)

What Source drives Status? Ho: some activity does NOT impact the variable of interest Ha: some activity DOES impact the variable of interest

Variable of interest <- Status

Ho: Source does not impact Status Ha: Source impacts Status

What County drive Status?? Ho: County does not impact Status Ha: County impacts Status

Join dataframes

```
traffic_source_grouping <- trafficReportList %>%
  right_join(GLSourceGrouping) %>%
  drop_na()
```

```
## Joining, by = "Source"
```

```
reg_source <- glm(formula = num_status ~ Source, data = traffic_source_grouping)

export_summs(reg_source)
```

	Model 1
(Intercept)	0.37 ***
	(0.02)
SourceApartments.com	-0.21
	(0.20)
SourceDrive-by/Signage	0.04
	(0.03)
SourceGoogle	0.05 **
	(0.02)
SourceReferral	0.10 *
	(0.04)
SourceZillow	0.04 *
	(0.02)
N	13777
AIC	19666.43
BIC	19719.14

Pseudo R2 0.00

*** p < 0.001; ** p < 0.01; * p < 0.05.

```
summary(reg_source)
```

```
##
## Call:
## glm(formula = num_status ~ Source, data = traffic_source_grouping)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.4788  -0.4282  -0.4158   0.5718   0.8333
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.37397    0.01587   23.559 < 2e-16 ***
## SourceApartments.com -0.20730    0.20224  -1.025  0.30538
## SourceDrive-by/Signage  0.04047    0.03007   1.346  0.17833
## SourceGoogle      0.05428    0.01659   3.272  0.00107 **
## SourceReferral     0.10482    0.04160   2.520  0.01175 *
## SourceZillow      0.04185    0.01974   2.120  0.03403 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2439053)
##
##      Null deviance: 3362.5  on 13776  degrees of freedom
## Residual deviance: 3358.8  on 13771  degrees of freedom
## AIC: 19666
##
## Number of Fisher Scoring iterations: 2
```

Comments

- Ho: Source does not impact Status
 - For every 1 unit increase in tenants signing a lease, there is a 0.11 unit increase in leases signed by Referral
 - This is significant at the 1% level
 - For every 1 unit increase in tenants signing a lease, there is a 0.04 unit increase in leases signed via Google
 - This is significant at the 5% level
 - We reject the null hypothesis due to a low p-value < 0.05
- Ha: Source impacts Status
 - We fail to reject the alternative hypothesis due to low p-value < 0.05

```
reg_county <- glm(formula = num_status ~ `COUNTY:`, data = trafficProperty)
export_summs(reg_county)
```

	Model 1
(Intercept)	0.10 ***
	(0.00)
`COUNTY:PIERCE`	0.07 ***
	(0.00)
`COUNTY:THURSTON`	-0.00
	(0.00)
N	85167
AIC	58592.10
BIC	58629.51
Pseudo R2	0.02

*** p < 0.001; ** p < 0.01; * p < 0.05.

```
summary(reg_county)
```

```
##
## Call:
## glm(formula = num_status ~ `COUNTY`, data = trafficProperty)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.1743  -0.1743  -0.1005  -0.0961   0.9039
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.100539    0.001940   51.834 <2e-16 ***
## `COUNTY`PIERCE    0.073738    0.002553   28.885 <2e-16 ***
## `COUNTY`THURSTON -0.004442    0.003680   -1.207   0.227
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1164878)
##
##      Null deviance: 10040.4  on 85166  degrees of freedom
## Residual deviance:  9920.6  on 85164  degrees of freedom
##   (4047 observations deleted due to missingness)
## AIC: 58592
##
## Number of Fisher Scoring iterations: 2
```

Comments

- Ho: County does not impact Status
 - For every 1 unit increase in a person signing a lease, there is a 0.08 unit association with leases being signed in Pierce County (compared to King county)
 - This is statistically significant at the 0.1 level
 - We reject the null hypothesis due to very low p-value (<2e-16)
 - There appears to be no effect in Thurston county compared to King County with p-value 0.509
- Ha: County impacts Status
 - We fail to reject the alternative hypothesis because (<2e-16) > 0.05

Export Dfs

```
write_xlsx(traffic_source_grouping, "../02_raw_data/output/traffic_source_grouping.csv")

write_xlsx(trafficProperty, "../02_raw_data/output/trafficProperty.csv")

##Cost Per Lead Analysis

write_xlsx(costPerLead, "../02_raw_data/output/costPerLead.csv")

write_xlsx(costPerYear, "../02_raw_data/output/costPerYear.csv")

write_xlsx(costPerSeason, "../02_raw_data/output/costPerSeason.csv")

write_xlsx(costPerPropertyType, "../02_raw_data/output/costPerPropertyType.csv")

write_xlsx(costPerCounty, "../02_raw_data/output/costPerCounty.csv")

write_xlsx(costPerLocation, "../02_raw_data/output/costPerLocation.csv")

write_xlsx(costPerPropertySource, "../02_raw_data/output/costPerPropertySource.csv")

write_xlsx(costPerCountySource, "../02_raw_data/output/costPerCountySource.csv")

write_xlsx(costPerLocationSource, "../02_raw_data/output/costPerLocationSource.csv")

write_xlsx(RatioStatusbyPropertyType, "../02_raw_data/output/RatioStatusbyPropertyType.csv")
```



```
TotalLeads <- sqldf('
  select *, (cast(sq.ResidentCount as float)/
             cast(sq.TotalLeads as float)) ResidentRate
  from
  (select cast(count(*) as float) TotalLeads,
   (select cast(count(*) as float) from trafficReportList
    where num_status = 1) ResidentCount
   from trafficReportList) sq
')
```