Data_exploration_Tao

Tao Tao

11/23/2019

Load the cleaned data.

```
library(readr)
policy <- read_csv("policies.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Quote_dt = col_date(format = ""),
##   discount = col_character(),
##   Home_policy_ind = col_character(),
##   state_id = col_character(),
##   county_name = col_character(),
##   quoted_amt = col_character(),
##   Prior_carrier_grp = col_character(),
##   Cov_package_type = col_character(),
##   policy_id = col_character(),
##   split = col_character(),
##   primary_parking = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v ggplot2 3.2.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
train <- policy %>%
  filter(split == 'Train')
```

```
nrow(train)
```

```
## [1] 36871
```

```
ncol(train)
```

```
## [1] 22
```

There are 36871 observations in the train dataset. And 20 variables when excluding convert_ind and split.

convert_ind

```
summary(factor(train$convert_ind))
```

```
##     0     1
## 32751  4120
```

```
4120/32751
```

```
## [1] 0.1257977
```

The overall convert rate is 0.1258.

quoted_amt

```
train$quoted_amt <- as.numeric(gsub('[$,]', '', train$quoted_amt))
summary(train$quoted_amt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      15    2246    3744    5849    6522  108608      87
```

```
cor.test(train$convert_ind, train$quoted_amt)
```

```
##
##  Pearson's product-moment correlation
##
## data:  train$convert_ind and train$quoted_amt
## t = -14.648, df = 36782, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08630522 -0.06598513
## sample estimates:
##         cor
## -0.07615308
```

A negative correlation between quoted_amt and convert_ind. This may be a important feature. Higher quoted_amt introduces lower convert_ind, which is interesting.

discount

```
train <- train %>%
  mutate(discount = if_else( discount == 'Yes', 1, 0))
summary(factor(train$discount))
```

```
##     0     1
## 27180  9691
```

```
cor.test(train$convert_ind, train$discount)
```

```
##
##  Pearson's product-moment correlation
##
## data:  train$convert_ind and train$discount
## t = 8.2361, df = 36869, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.03266133 0.05303832
```

```
## sample estimates:
##        cor
## 0.04285428
```

There is a positive correlation between discount and convert_ind. The policy of discount introduces higher convert_ind, which makes sense.

Home_policy_ind

```
train <- train %>%
  mutate(Home_policy_ind = if_else( Home_policy_ind == 'Y', 1, 0))
summary(factor(train$Home_policy_ind))
```

```
##     0     1
## 29954  6917
```

```
cor.test(train$convert_ind, train$Home_policy_ind)
```

```
##
##  Pearson's product-moment correlation
##
## data:  train$convert_ind and train$Home_policy_ind
## t = 6.5705, df = 36869, p-value = 5.08e-11
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.02400033 0.04439093
## sample estimates:
##        cor
## 0.03419919
```

There is a positive correlation between discount and Home_policy_ind. If the customer has bought the home insurance in this company, then the customer is more likely to buy car insurance in the same company.

credit_score

```
summary(train$credit_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   369.0   583.0   642.0   641.5   697.0   850.0     224
```

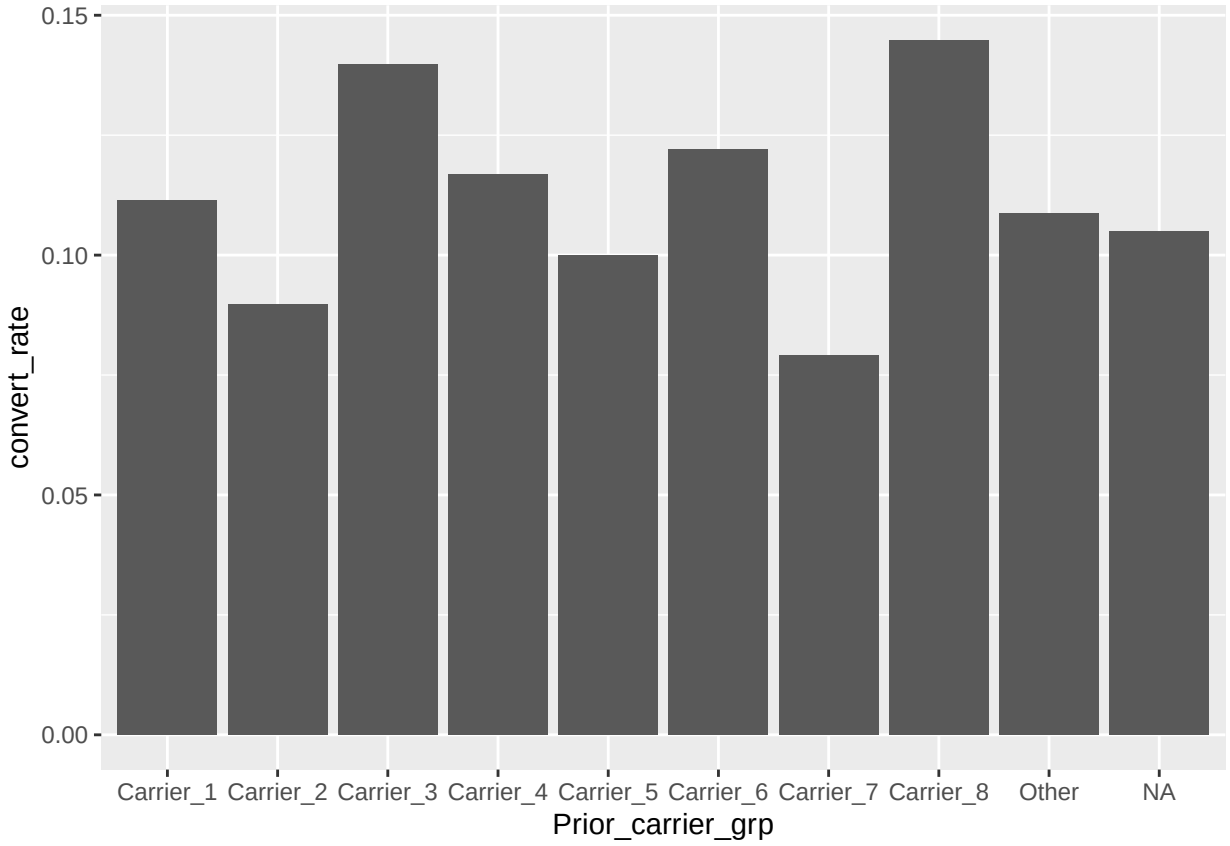
```
cor.test(train$convert_ind, train$credit_score)
```

```
##
##  Pearson's product-moment correlation
##
## data:  train$convert_ind and train$credit_score
## t = 13.89, df = 36645, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.06217785 0.08254739
## sample estimates:
##        cor
## 0.07237016
```

There is a positive correlation between discount and credit_score. If the customer has a higher credit score, the customer intends to convert.

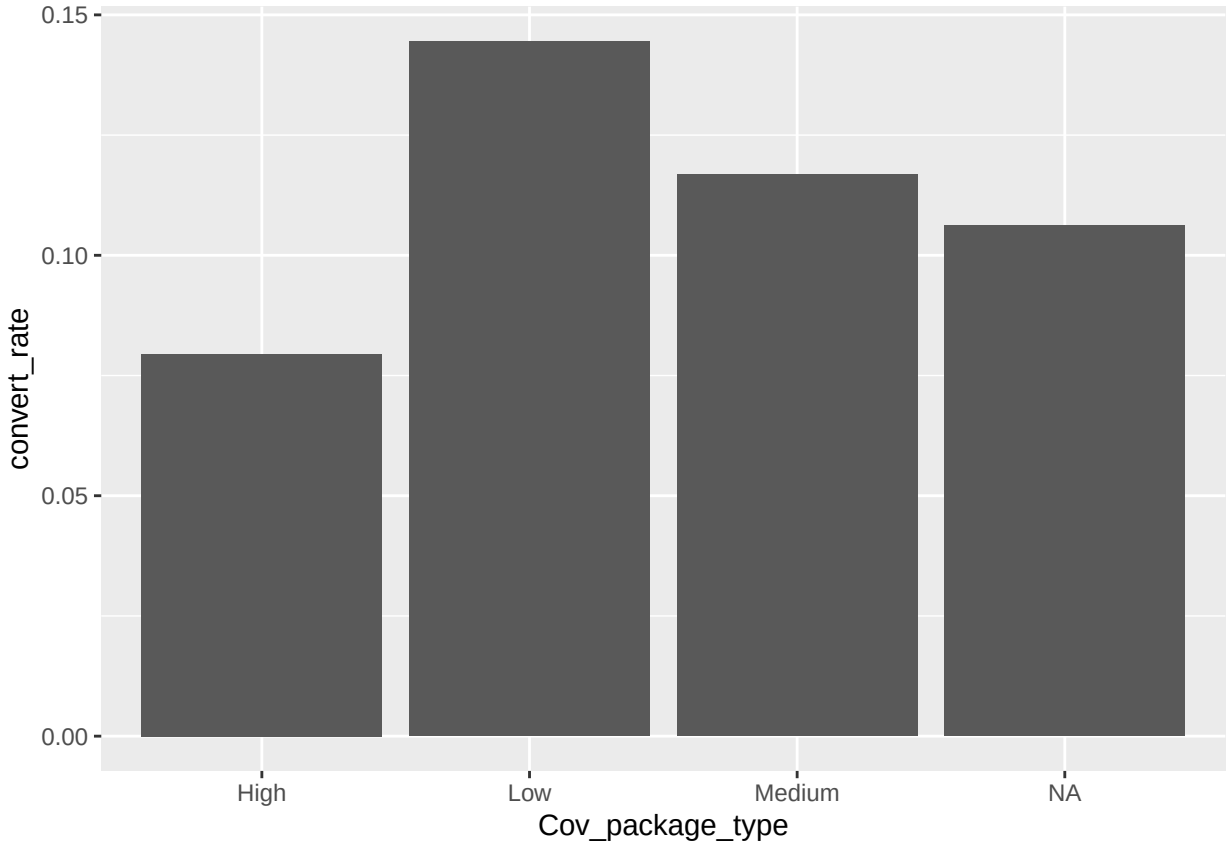
Prior_carrier_grp

```
train %>%
  group_by(Prior_carrier_grp) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = Prior_carrier_grp, y = convert_rate)) +
  geom_col()
```



cov_package_type

```
train %>%
  group_by(Cov_package_type) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = Cov_package_type, y = convert_rate)) +
  geom_col()
```



quote_dt

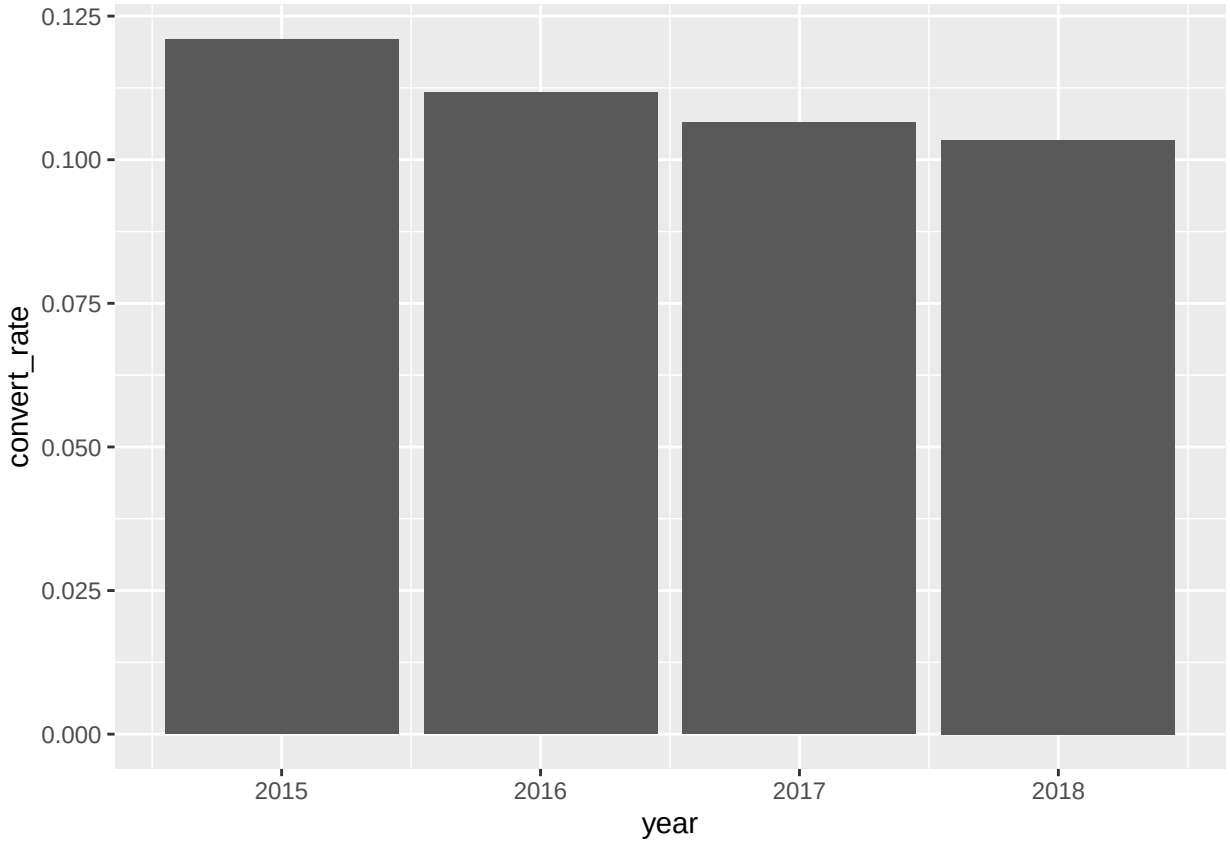
year

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

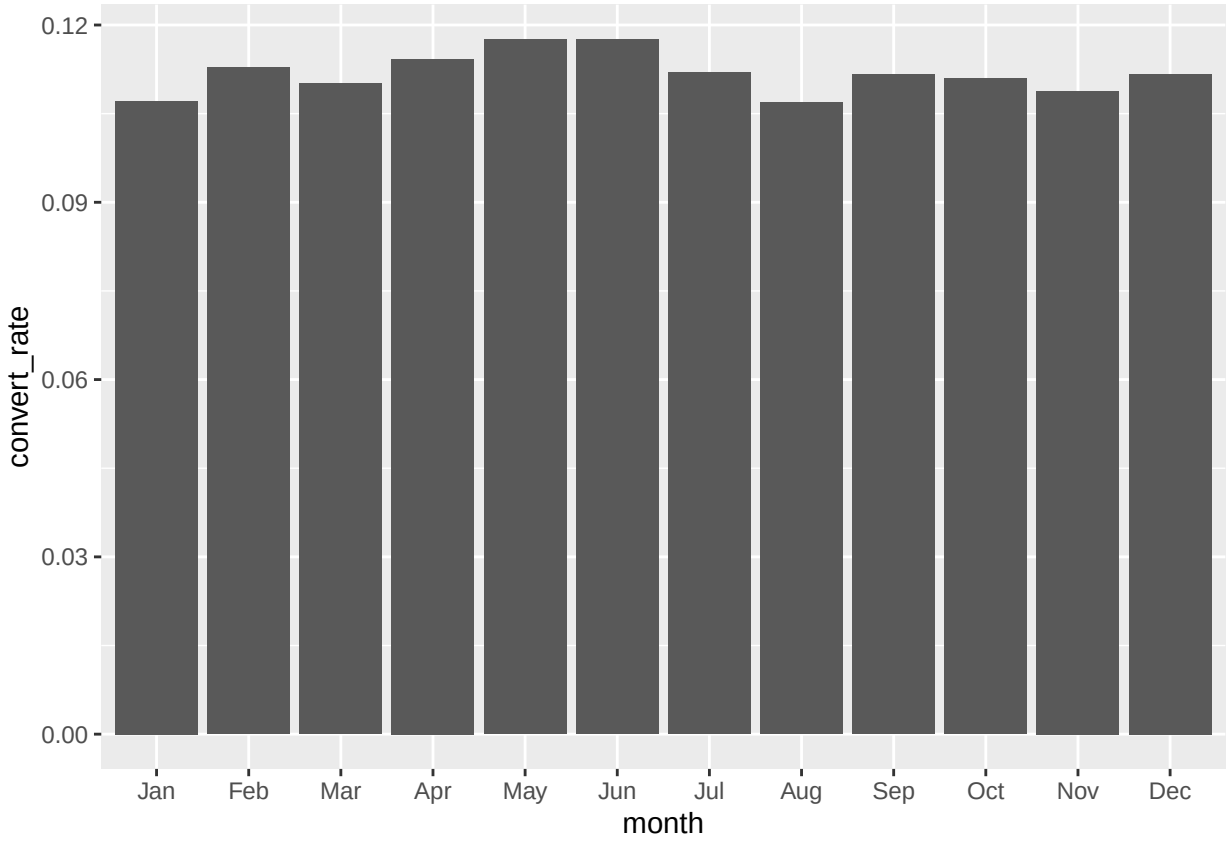
```
train %>%
  mutate(year = year(Quote_dt)) %>%
  group_by(year) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = year, y = convert_rate)) +
  geom_col()
```



Bad news for the company.

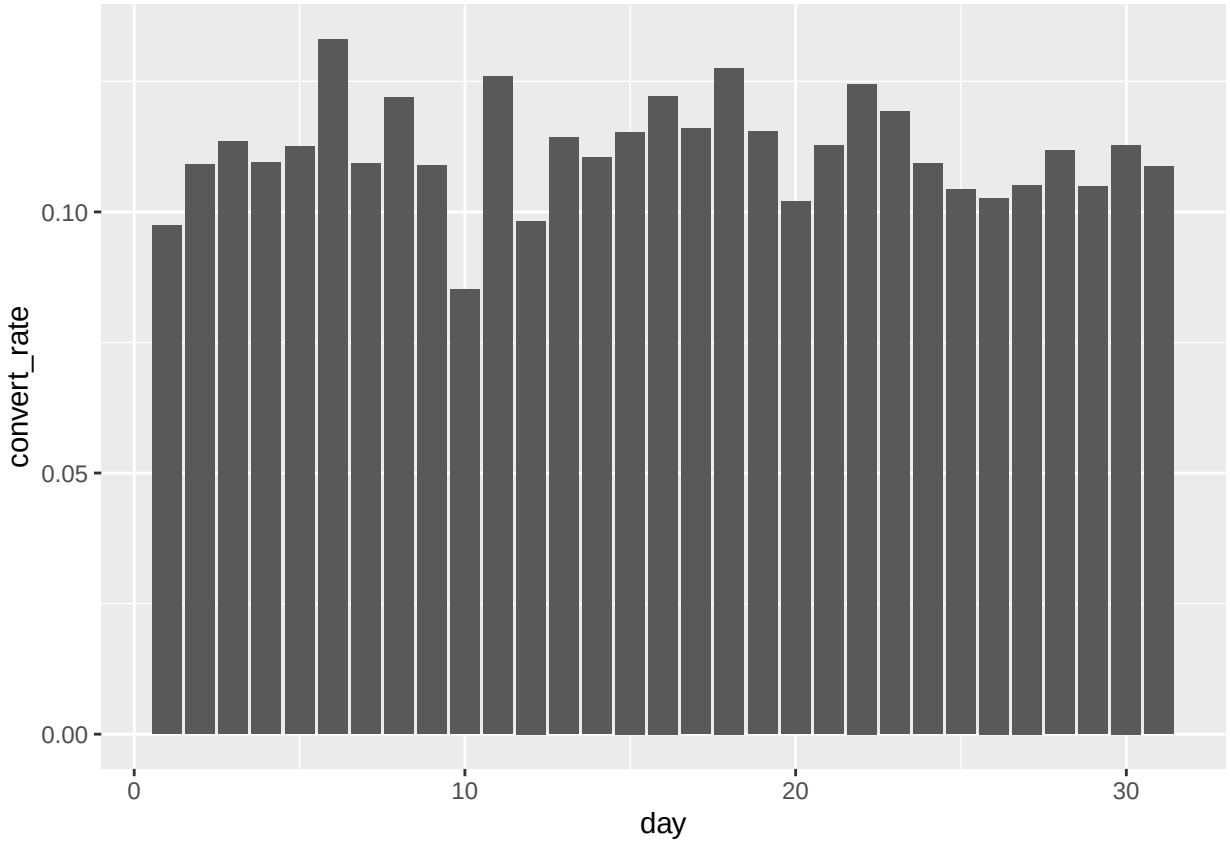
month

```
train %>%
  mutate(month = month(Quote_dt, label = T)) %>%
  group_by(month) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = month, y = convert_rate)) +
  geom_col()
```



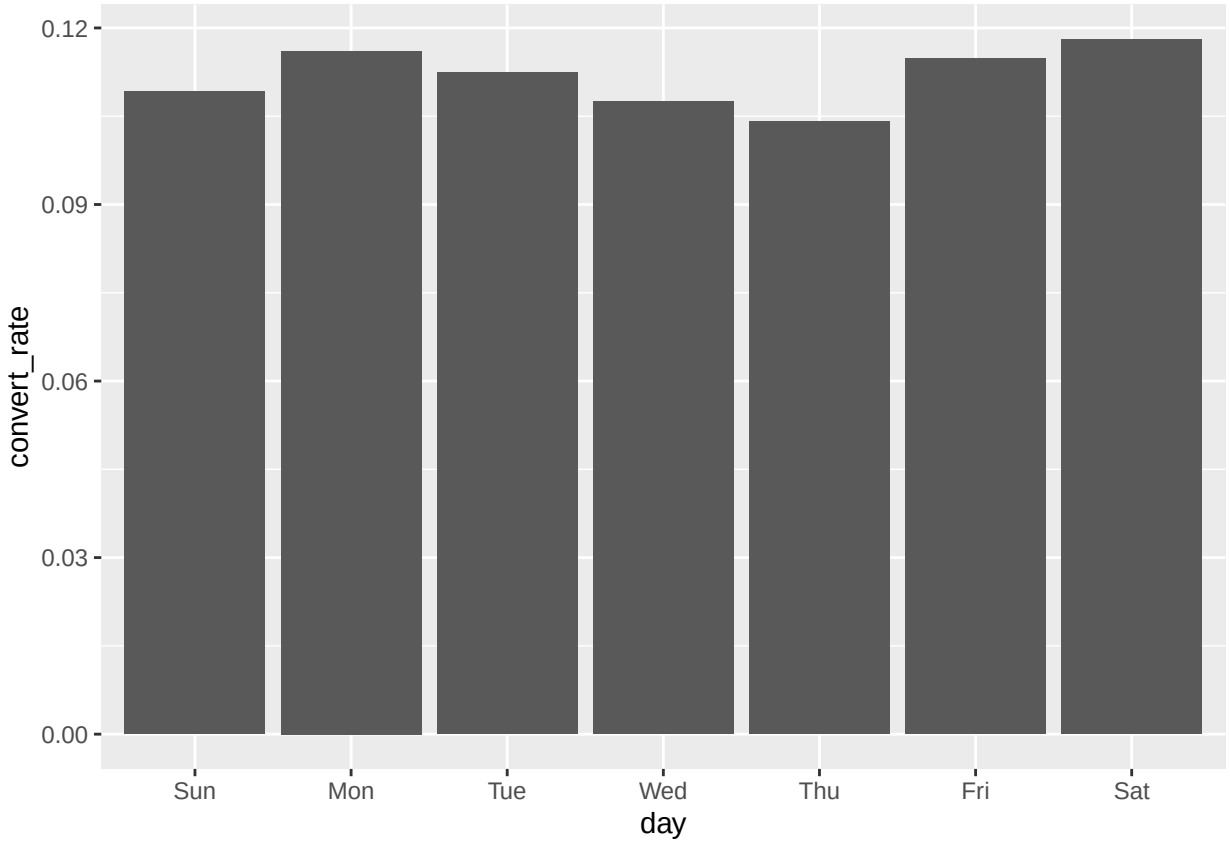
day of month

```
train %>%
  mutate(day = day(Quote_dt)) %>%
  group_by(day) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = day, y = convert_rate)) +
  geom_col()
```



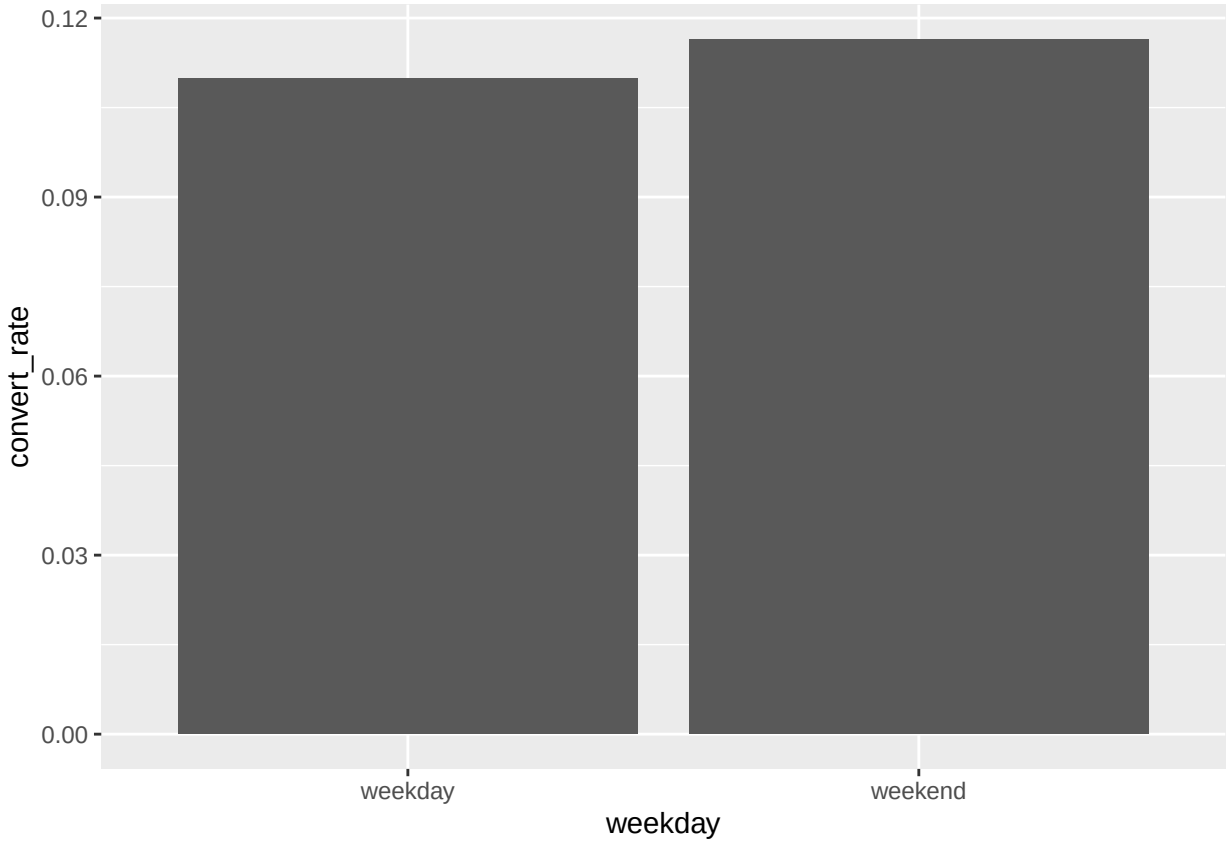
day of week

```
train %>%
  mutate(day = wday(Quote_dt, label = T)) %>%
  group_by(day) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = day, y = convert_rate)) +
  geom_col()
```



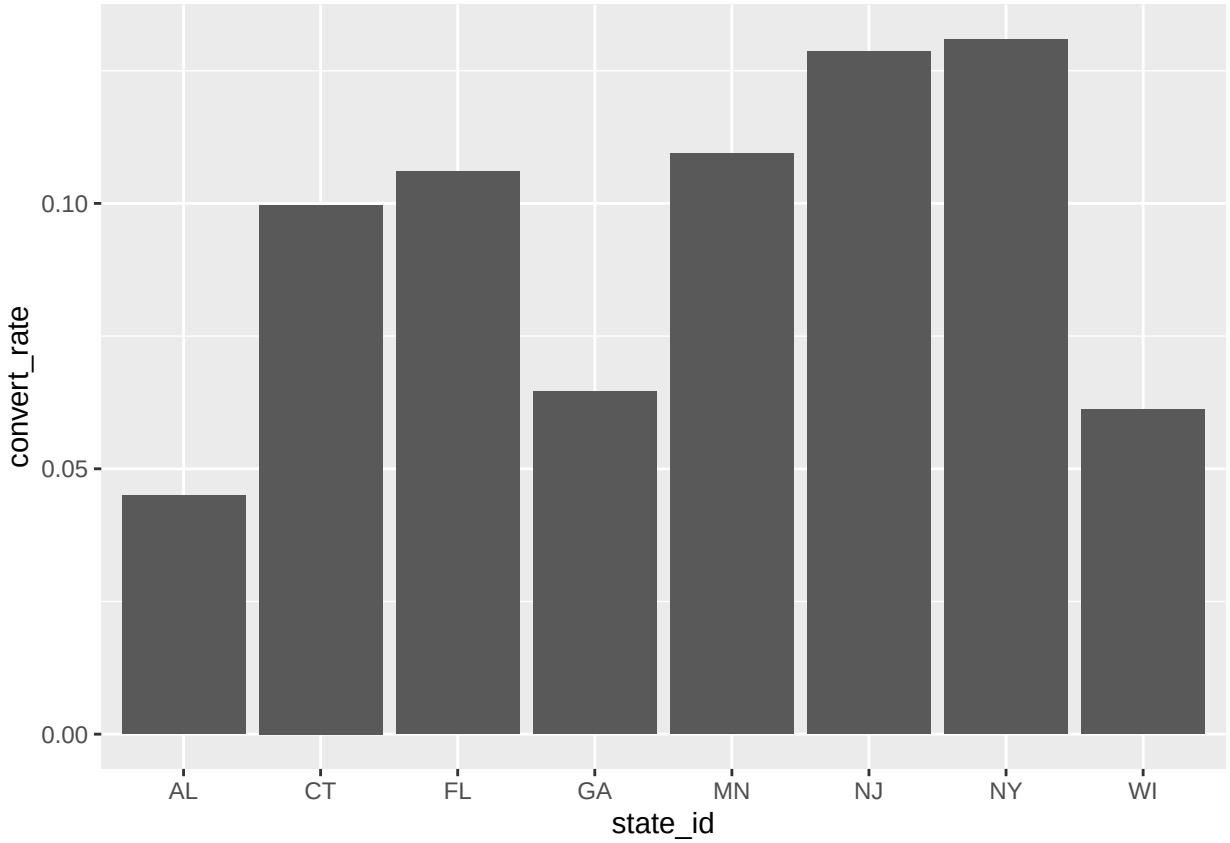
weekday/weekend

```
train %>%
  mutate(day = wday(Quote_dt)) %>%
  mutate(weekday = if_else(day <=5, 'weekday', 'weekend')) %>%
  group_by(weekday) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = weekday, y = convert_rate)) +
  geom_col()
```



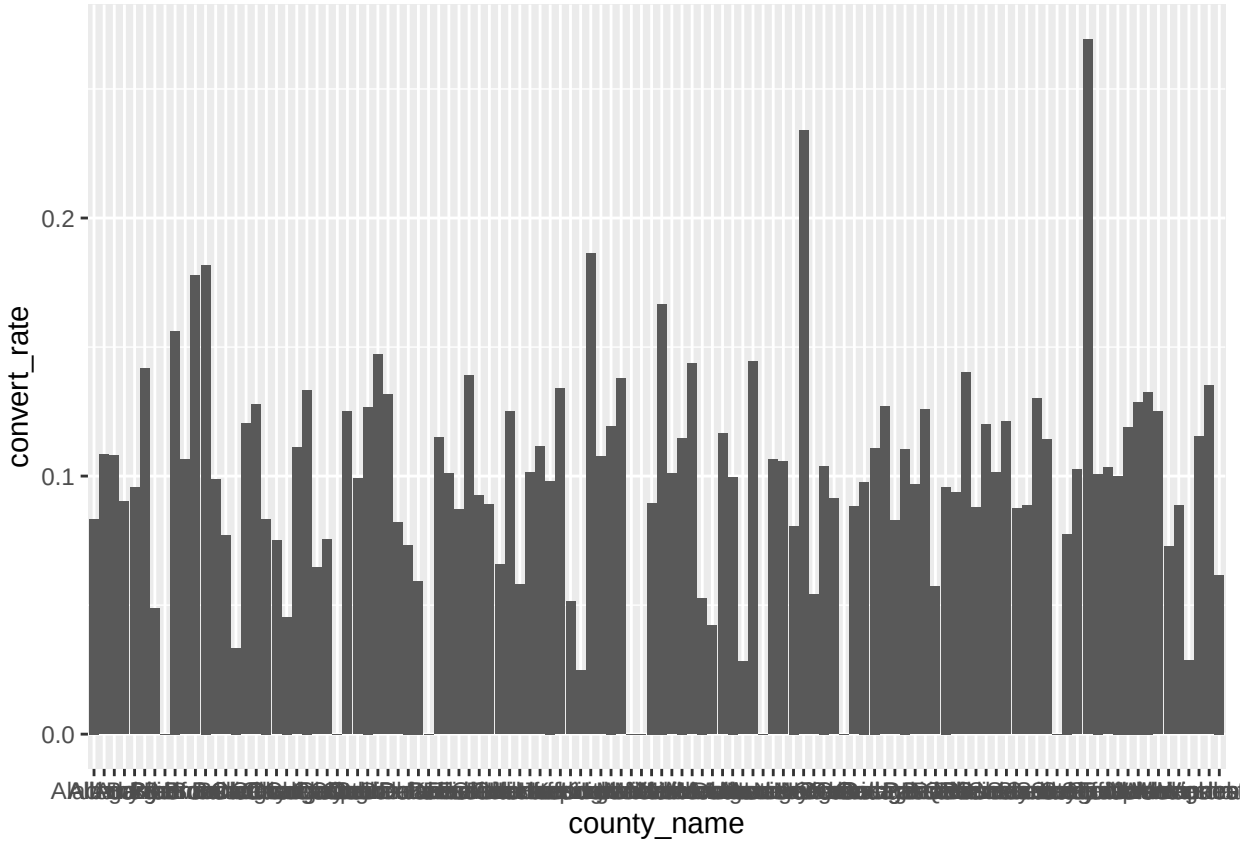
state_id

```
train %>%
  group_by(state_id) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = state_id, y = convert_rate)) +
  geom_col()
```



county_name

```
train %>%
  group_by(county_name) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = county_name, y = convert_rate)) +
  geom_col()
```



zip

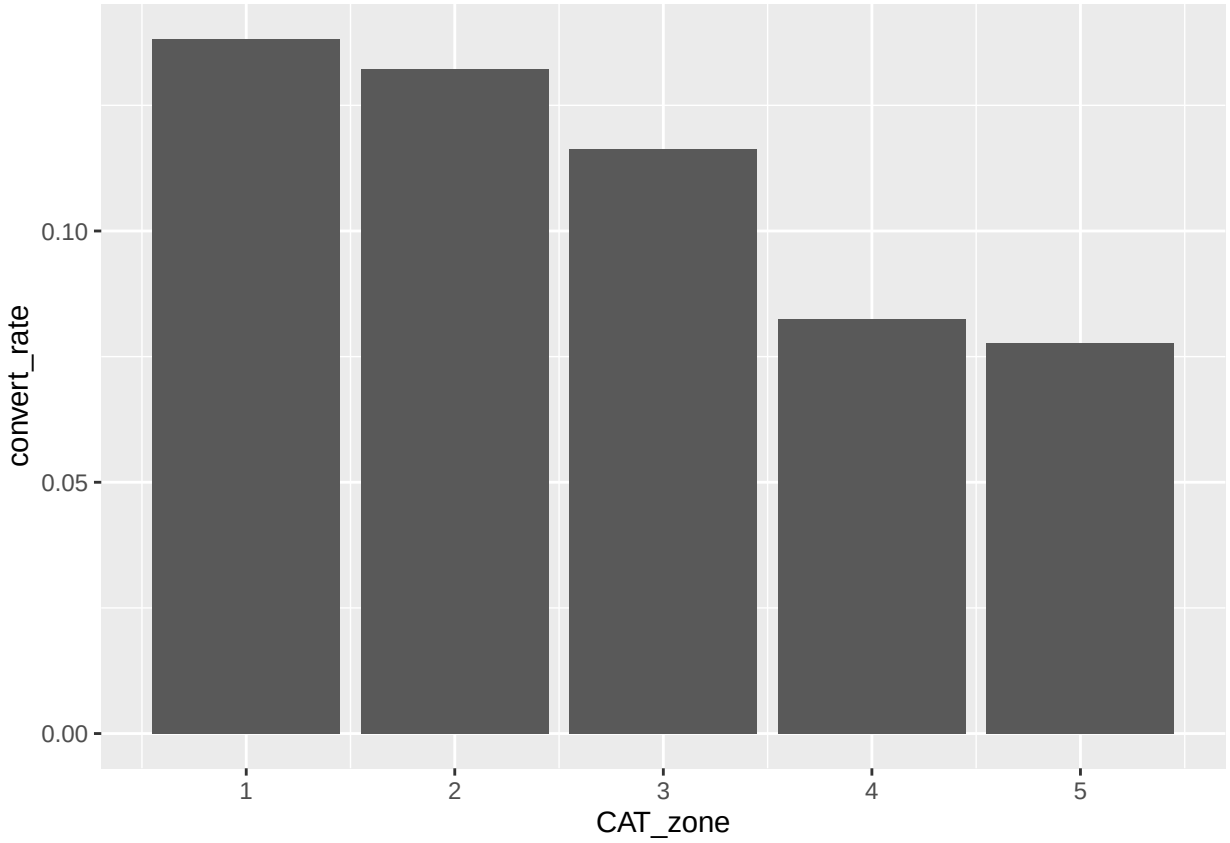
```
train %>%
  group_by(zip) %>%
  summarise(
    num = n(),
    convert_rate = sum(convert_ind)/n()
    ) %>%
  arrange(desc(convert_rate))
```

```
## # A tibble: 1,169 x 3
##       zip   num convert_rate
##     <dbl> <int>        <dbl>
##  1 10028    17        0.471
##  2 10075    17        0.471
##  3 10016    17        0.412
##  4 10031    17        0.412
##  5 11549    17        0.412
##  6 10023    10        0.4
##  7 10039    23        0.391
##  8 10009    18        0.389
##  9 10019    26        0.385
## 10 10452    26        0.385
## # ... with 1,159 more rows
```

CAT_zone

```
train %>%
  group_by(CAT_zone) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = CAT_zone, y = convert_rate)) +
  geom_col()
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```



number_drivers

```
summary(train$number_drivers)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.159   3.000   6.000
```

```
cor.test(train$convert_ind, train$number_drivers)
```

```
##
##  Pearson's product-moment correlation
##
## data:  train$convert_ind and train$number_drivers
## t = -12.235, df = 36869, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07374950 -0.05341757
```

```
## sample estimates:
##         cor
## -0.06359014
```

primary_parking

```
train %>%
  group_by(primary_parking) %>%
  summarise(convert_rate = sum(convert_ind)/n()) %>%
  ggplot(aes(x = primary_parking, y = convert_rate)) +
  geom_col()
```