Predicting Revenue from Google Advertising Data

MATH2319 - Machine Learning
Course Project

Ben Cole - s3412349

Print Date: 27/05/2019

Contents

1 Phase 1 - Introduction, Cleaning, and Exploration **2**
1.1 Outline . . . 2
1.1.1 Nature of the Data . . . 2
1.2 Data Processing . . . 3
1.2.1 Libraries . . . 3
1.2.2 Loading Data . . . 4
1.2.3 Classifying Data . . . 4
1.2.4 Descriptive Statistics . . . 5
1.2.5 Density Plots . . . 9
1.3 References . . . 22

1 Phase 1 - Introduction, Cleaning, and Exploration

1.1 Outline

The prescribed data set contained advertising metrics provided by a prominent search engine. The data contained severald descriptive features pertaining to a range of information. Finally, the target feature was a measure of revenue associated with each of the observations.

1.1.1 Nature of the Data

The below is an exerpt from accompanying documentation about the dataset.

Features in this data set are as follows:

- companyId: Company ID of record (categorical)
- countryId: Country ID of record (categorical)
- deviceType: Device type of record (categorical corresponding to desktop, mobile, tablet)
- day: Day of record (integer between 1 (oldest) and 30 for train, 31 and 35 (most recent) for test)
- dow: Day of week of the record (categorical)
- price1, price2, price3: Price combination for the record set by the company (numeric)
- ad_area: area of advertisement (numeric)
- ad_ratio: ratio of advertisement's length to its width (numeric)
- requests, impression, cpc, ctr, viewability: Various metrics related to the record (numeric)
- ratio1, …, ratio5: Ratio characteristics related to the record (numeric)
- y (target feature): revenue-related metric (numeric)

1.1.1.1 Target Feature

The column/variable **y** was selected as the target feature in the dataset.

1.1.1.2 Descriptive Features

All other columns/variables in the dataset, as outlined above, were chosen as descriptive features.

1.2 Data Processing

1.2.1 Libraries

```
library(pacman)                         ## for loading multiple packages

suppressMessages(p_load(character.only = T,
                        install = F,
                        c("tidyverse",  ## thanks Hadley
                          "lubridate",  ## for handling dates
                          "forcats",    ## for categorial variables, not for felines
                          "zoo",        ## some data cleaning capabilities
                          "lemon",      ## add ons for ggplot
                          "rvest",      ## scraping web pages
                          "knitr",      ## knitting to RMarkdown
                          "kableExtra", ## add ons for knitr tables
                          "scales",     ## quick and easy formatting prettynums
                          "grid",       ## for stacking ggplots
                          "gridExtra",  ## also for stacking ggplots
                          "e1071",      ## for skew and kurtosis
                          "janitor",    ## cleaning colnames
                          "beepr")))    ## plays a beep tone
```

Table 1: Sample of Advertising Data Frame

case_id	companyId	countryId	deviceType	day	dow	price1	price2	price3	ad_area	ad_ratio
155814	159	140	2	22	Saturday	0.12	0.27	0.5446	0.0001	1.00000
73956	159	144	1	12	Wednesday	0.15	0.34	0.6651	7.5000	0.83333
127206	43	77	1	19	Wednesday	0.21	0.58	1.1509	18.0000	2.00000
156770	43	100	3	22	Saturday	0.00	0.00	0.0000	0.0001	1.00000
20006	43	12	1	4	Tuesday	0.16	1.05	2.0970	7.5000	0.83333
113091	43	108	1	17	Monday	0.00	0.00	0.0000	7.5000	0.83333
207612	43	249	1	30	Sunday	0.22	0.66	1.3131	3.2000	0.31250
32514	95	38	2	6	Thursday	0.84	1.62	3.2400	18.0000	2.00000
95605	43	10	2	14	Friday	0.09	0.17	0.3283	6.5520	0.12363
211484	43	59	2	30	Sunday	0.00	0.00	0.0000	0.0001	1.00000
5867	43	139	2	1	Saturday	0.57	0.57	0.5703	0.0001	1.00000
81327	159	140	1	13	Thursday	0.00	0.00	0.0000	7.5000	0.83333
117931	43	234	1	17	Monday	0.00	0.00	0.0000	7.5000	0.83333
171271	43	43	2	24	Monday	3.40	3.40	3.4035	0.0001	1.00000
12830	43	234	1	3	Monday	0.00	0.00	0.0000	18.0000	2.00000
100071	43	101	1	15	Saturday	0.80	0.80	0.7964	0.0001	1.00000
20857	43	77	2	4	Tuesday	1.76	3.05	6.1200	7.5000	0.83333
188188	159	57	2	27	Thursday	0.68	1.26	2.5144	0.0001	1.00000
209180	159	167	1	30	Sunday	0.06	0.36	0.7358	7.5000	0.83333
197464	43	178	2	28	Friday	2.72	3.21	6.4193	8.7300	0.09278

1.2.2 Loading Data

```
advertising_train <- read_csv("advertising_train.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   dow = col_character()
## )
```

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

```
sample_adv <- sample_n(advertising_train, 20)

kable_styling(kable(sample_adv[ , 1:(ncol(sample_adv)/2)],
                    caption = "Sample of Advertising Data Frame"),
              font_size = 8.5, latex_options = c("striped"),
              full_width = F)
```

```
kable_styling(kable(sample_adv[ , c(1, ((ncol(sample_adv)/2)+1):ncol(sample_adv))],
                    caption = "Sample of Advertising Data Frame (cont)"),
              font_size = 8.5, latex_options = c("striped"),
              full_width = F)
```

1.2.3 Classifying Data

Per the above feature definitions, the data was classified.

```
advertising_train$companyId <- as.factor(advertising_train$companyId)

advertising_train$countryId <- as.factor(advertising_train$countryId)

advertising_train$deviceType <- as.factor(advertising_train$deviceType)
```

Table 2: Sample of Advertising Data Frame (cont)

case_id	requests	impression	cpc	ctr	viewability	ratio1	ratio2	ratio3	ratio4	ratio5	y
155814	948	595	0.1381	0.0017	0.7439	0.7866	0.9748	1.0000	0.0000	0.0000	0.1261654
73956	2614	424	0.0400	0.0142	0.5481	0.7241	0.9835	0.0920	0.1203	0.7877	0.0914939
127206	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.6029821
156770	36	26	0.0087	0.0385	0.5000	0.9615	0.6538	0.0000	0.6923	0.3077	0.0459459
20006	5244	1787	0.9114	0.0006	0.5788	0.3783	0.9720	0.0470	0.2440	0.7096	0.2453885
113091	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	3.3793103
207612	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	2.4932432
32514	7449	3392	0.3996	0.0032	0.1980	0.6763	0.7810	1.0000	0.0000	0.0000	0.5470354
95605	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0324607
211484	102	99	0.0323	0.0303	0.8718	1.0000	0.8081	1.0000	0.0000	0.0000	1.1343066
5867	45	35	0.0019	0.4286	0.9333	0.9714	0.9143	1.0000	0.0000	0.0000	0.3745098
81327	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1081081
117931	1924	1922	1.0130	0.0068	0.8015	1.0000	0.3632	0.2856	0.0713	0.6431	5.6790493
171271	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.8250000
12830	539	431	0.2545	0.0023	0.0667	1.0000	0.6497	0.0371	0.7471	0.2158	0.4520782
100071	509	147	0.0819	0.0544	1.0000	0.9932	0.1088	0.2313	0.0000	0.7687	1.6350951
20857	1277	557	0.9146	0.0036	0.9282	0.5871	0.9838	1.0000	0.0000	0.0000	1.3565114
188188	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.2346608
209180	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.4916724
197464	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5023333

```
advertising_train$dow <- as.factor(advertising_train$dow)

sapply(advertising_train, class)
```

```
##     case_id   companyId   countryId  deviceType         day         dow
##   "numeric"    "factor"    "factor"    "factor"   "numeric"    "factor"
##      price1      price2      price3     ad_area    ad_ratio    requests
##   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
##  impression         cpc         ctr viewability      ratio1      ratio2
##   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
##      ratio3      ratio4      ratio5           y
##   "numeric"   "numeric"   "numeric"   "numeric"
```

1.2.4 Descriptive Statistics

1.2.4.1 Numeric Variables

```
advertising_train_long_num <- select(advertising_train,
                                     colnames(advertising_train),
                                     -case_id, -countryId,
                                     -companyId, -deviceType,
                                     -dow)

advertising_train_long_num <- gather(advertising_train_long_num,
                                     key = "Variable",
                                     value = "Value")

summary_adv_num <- summarise(group_by(advertising_train_long_num,
                                      Variable),
                             "Mean" = mean(Value, na.rm = T),
                             "Std Dev" = sd(Value, na.rm = T),
                             "Min" = min(Value, na.rm = T),
                             "Q1" = quantile(Value, 0.25, na.rm = T),
```

Table 3: Summary Statistics of Numeric Variables

Variable	Mean	Std Dev	Min	Q1	Median	Q3	Max	Number of NA
ad_area	4.724	6.273	0.000	0.000	0.000	7.500	36.000	0.000
ad_ratio	0.923	0.482	0.083	0.833	1.000	1.000	5.000	0.000
cpc	0.178	0.707	0.000	0.000	0.016	0.125	132.534	0.000
ctr	0.033	0.093	0.000	0.000	0.002	0.012	2.000	0.000
day	15.791	8.386	1.000	9.000	16.000	23.000	30.000	0.000
impression	5,585.714	98,713.340	0.000	0.000	99.000	1,058.000	6,100,324.000	0.000
price1	0.438	1.281	0.000	0.000	0.010	0.190	14.690	0.000
price2	0.630	1.482	0.000	0.000	0.090	0.570	63.120	0.000
price3	0.932	1.840	0.000	0.000	0.295	0.986	78.900	0.000
ratio1	0.558	0.447	0.000	0.000	0.750	1.000	1.000	0.000
ratio2	0.491	0.414	0.000	0.000	0.627	0.896	1.027	0.000
ratio3	0.312	0.444	0.000	0.000	0.028	1.000	1.500	0.000
ratio4	0.131	0.240	0.000	0.000	0.000	0.164	1.077	0.000
ratio5	0.188	0.297	0.000	0.000	0.000	0.385	1.200	0.000
requests	8,678.997	122,347.229	0.000	0.000	147.000	1,633.000	6,701,924.000	0.000
viewability	0.378	0.366	0.000	0.000	0.332	0.716	7.000	0.000
y	0.847	1.391	0.000	0.150	0.419	0.959	47.060	0.000

```
                            "Median" = median(Value, na.rm = T),
                            "Q3" = quantile(Value, 0.75, na.rm = T),
                            "Max" = max(Value, na.rm = T),
                            "Number of NA" = sum(is.na(Value)))

kable_styling(kable(summary_adv_num,
                    digits = 3, format.args = list(nsmall = 3,
                                                   scientific = F,
                                                   big.mark = ","),
                    caption = "Summary Statistics of Numeric Variables"),
              font_size = 8.5, latex_options = c("striped"),
              full_width = F)
```

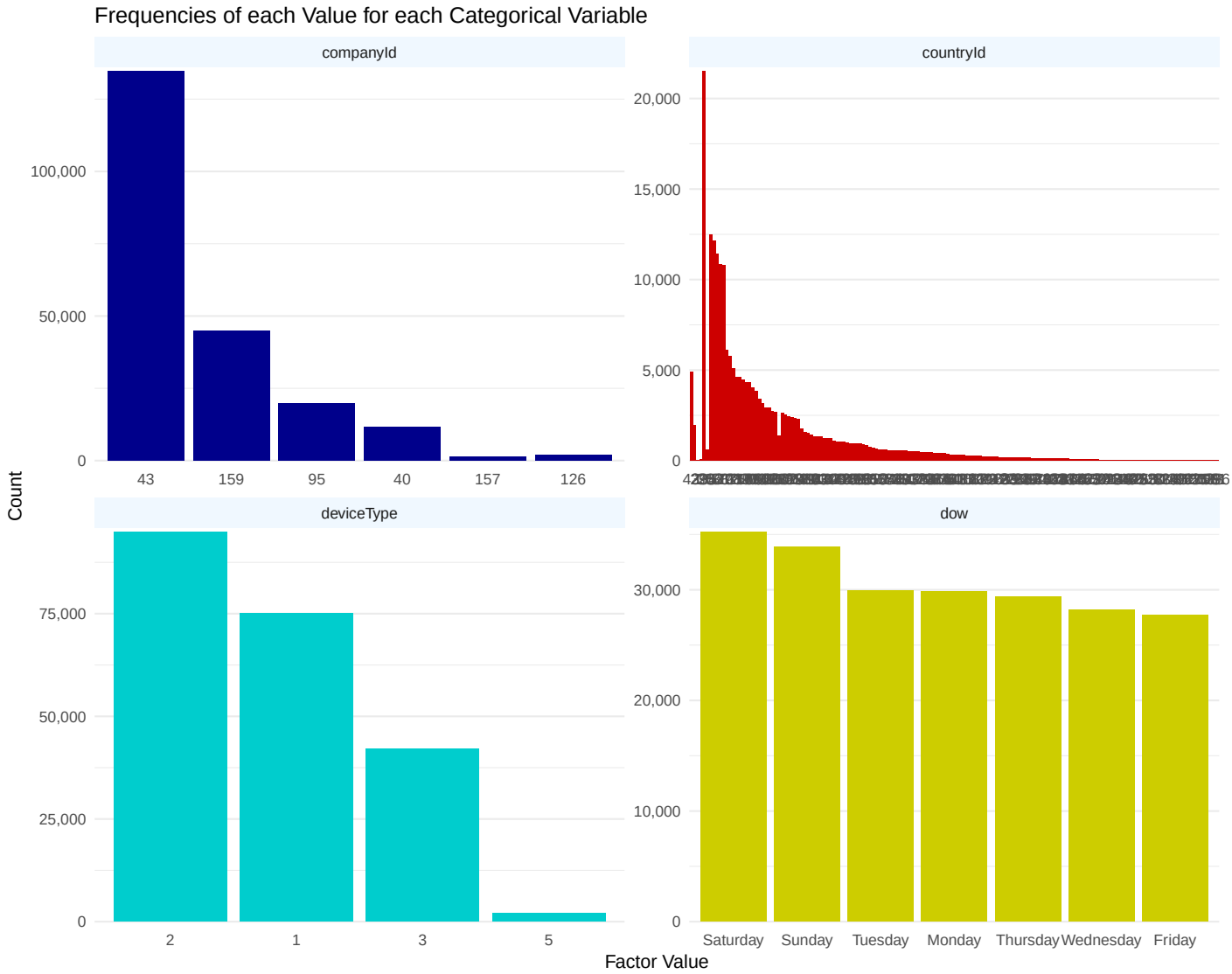
1.2.4.2 Categorical and Non-Numeric Variables

```
advertising_train_long_cat <- select(advertising_train,
                                     countryId,
                                     companyId, deviceType,
                                     dow)

advertising_train_long_cat <- gather(advertising_train_long_cat,
                                     key = "Variable",
                                     value = "Value")
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

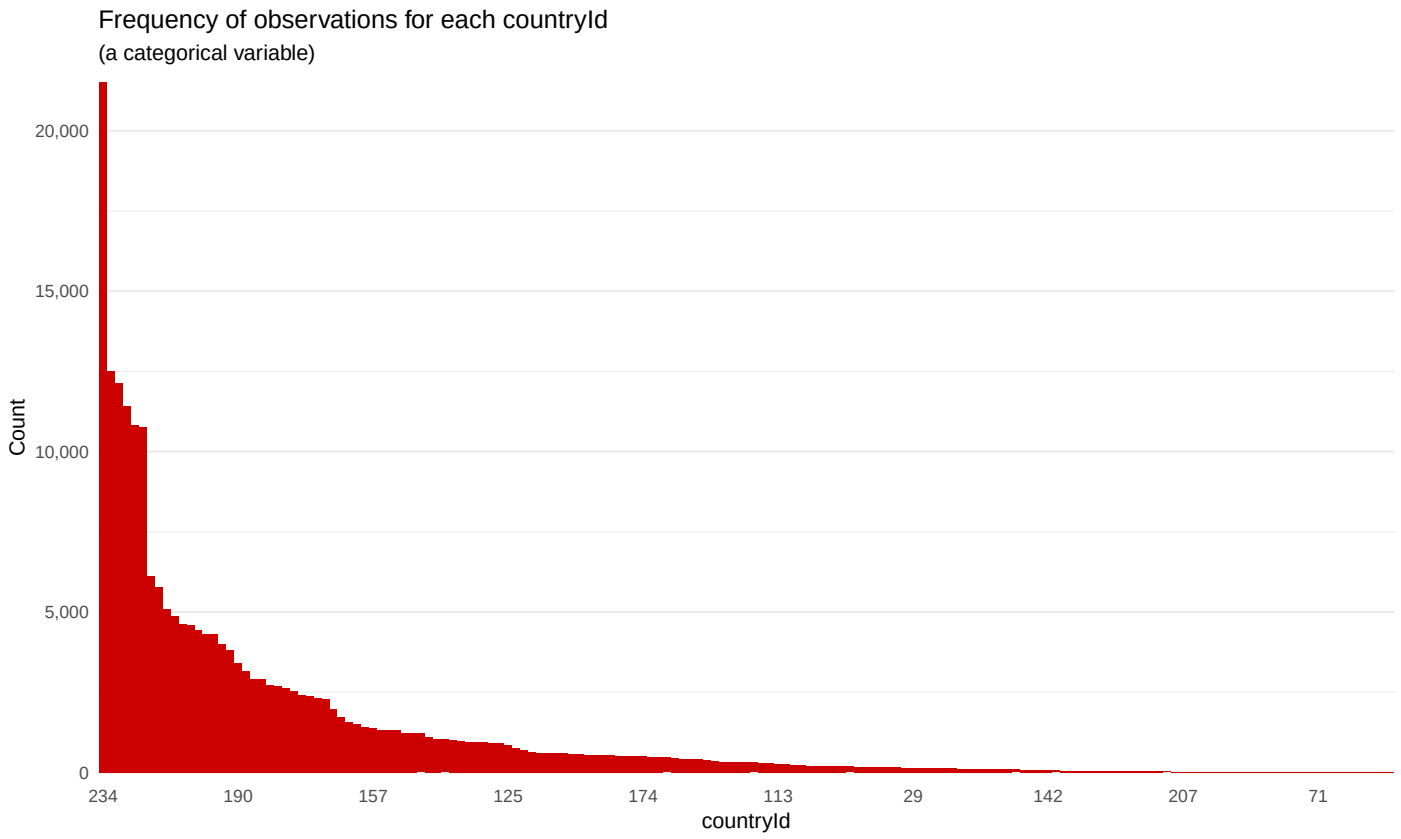
```
advertising_train_long_cat$Variable <- as.factor(advertising_train_long_cat$Variable)

advertising_train_long_cat$Value <- as.factor(advertising_train_long_cat$Value)

ggplot(advertising_train_long_cat) +
  geom_bar(aes(x = fct_infreq(Value),
               fill = Variable),
           show.legend = F) +
```

```
  facet_rep_wrap(~Variable,
                 repeat.tick.labels = T,
                 scales = "free") +
  scale_y_continuous(labels = comma,
                     expand = c(0.01, 0),
                     "Count") +
  scale_x_discrete("Factor Value") +
  scale_fill_manual(values = c("blue4", "red3", "cyan3", "yellow3")) +
  labs(title = "Frequencies of each Value for each Categorical Variable") +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        strip.background = element_rect(fill = "aliceblue",
                                        colour = NA))
```




```
country_labels <- levels(fct_infreq(advertising_train$countryId))[c(seq(1,
                                                                        length(levels(fct_infreq(adv
                                                                        ceiling(length(levels(fct_ir


ggplot(advertising_train) +
```

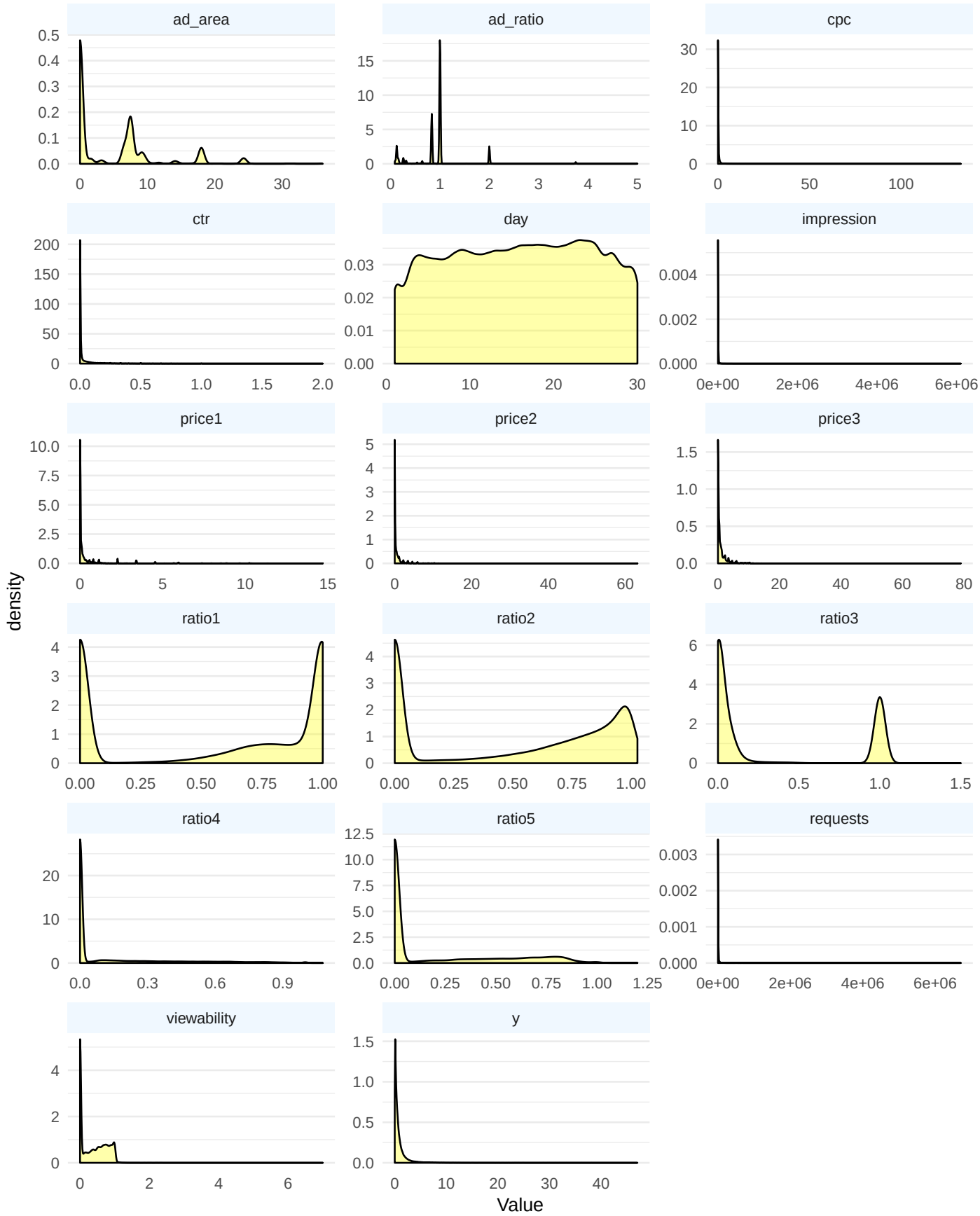
```
  geom_bar(aes(x = fct_infreq(countryId)),
           fill = "red3") +
  scale_y_continuous(labels = comma,
                     expand = c(0.01, 0),
                     "Count") +
  scale_x_discrete(breaks = country_labels,
                   "countryId") +
  labs(title = "Frequency of observations for each countryId",
       subtitle = "(a categorical variable)") +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())
```



1.2.5 Density Plots

1.2.5.1 Numeric Variables

```
ggplot(advertising_train_long_num) +
  geom_density(aes(x = Value),
               fill = "yellow",
               alpha = 1/3) +
  facet_rep_wrap(~Variable,
                 repeat.tick.labels = T,
                 scales = "free",
                 ncol = 3) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        strip.background = element_rect(fill = "aliceblue",
                                        colour = NA))
```

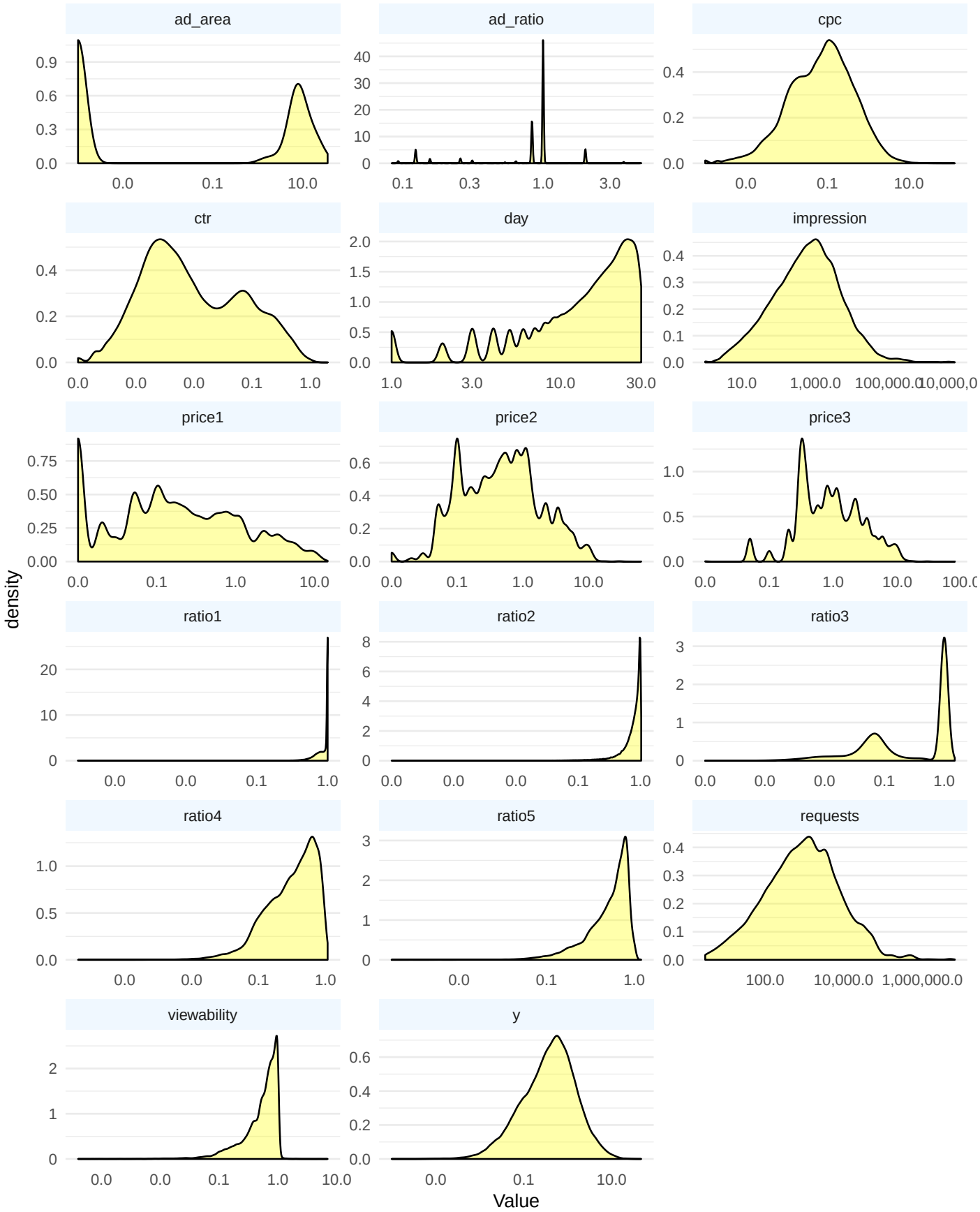



```
ggplot(advertising_train_long_num) +
  geom_density(aes(x = Value),
               fill = "yellow",
```

```
              alpha = 1/3) +
  facet_rep_wrap(~Variable,
                 repeat.tick.labels = T,
                 scales = "free",
                 ncol = 3) +
  scale_x_log10(labels = comma_format(accuracy = 0.1)) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        strip.background = element_rect(fill = "aliceblue",
                                        colour = NA))
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Removed 1213004 rows containing non-finite values (stat_density).
```

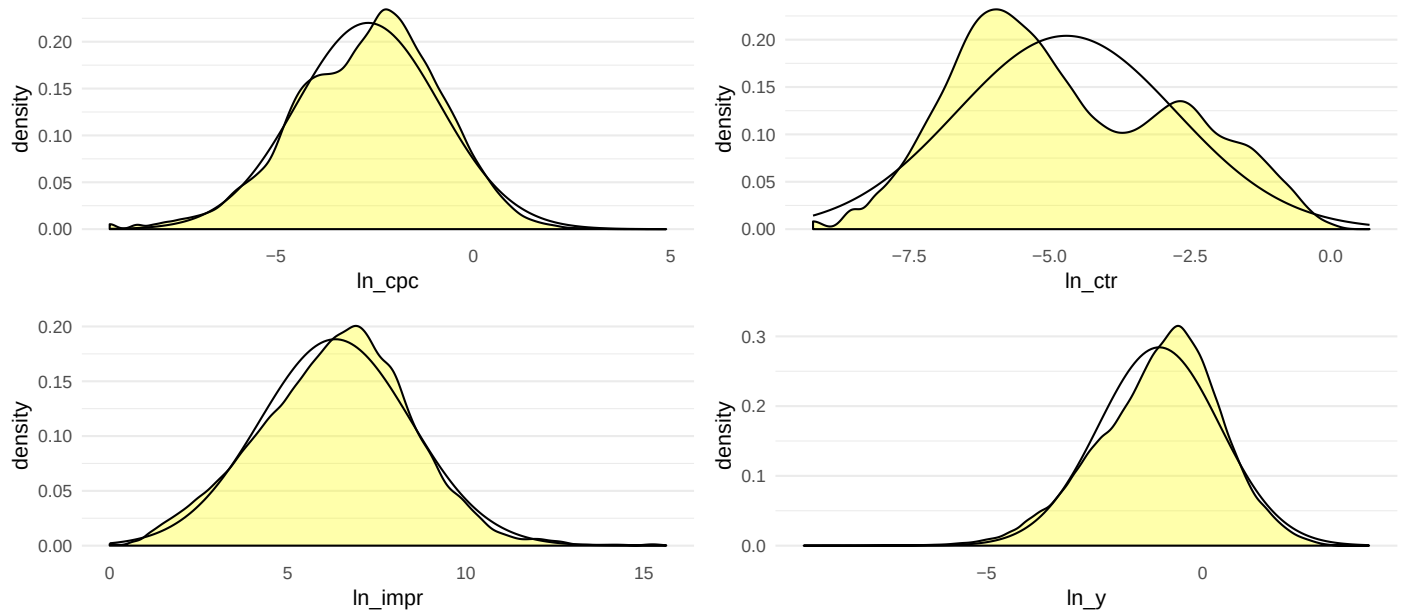



```
advertising_train <- mutate(advertising_train,
                            "ln_cpc" = log(cpc),
                            "ln_ctr" = log(ctr),
```

```
                            "ln_impr" = log(impression),
                            "ln_y" = log(y))
```

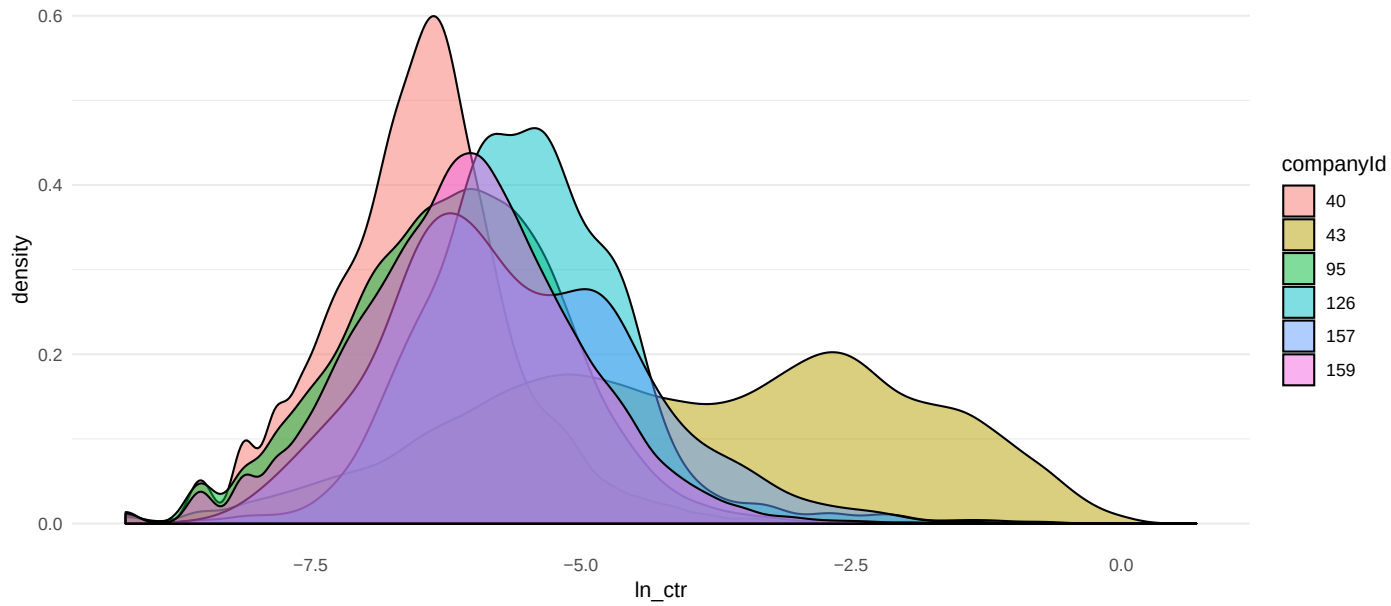
```
finite_cpc <- filter(advertising_train,
                     is.finite(ln_cpc))

p_cpc <- ggplot(finite_cpc) +
  geom_density(aes(x = ln_cpc),
               fill = "yellow", alpha = 1/3) +
  stat_function(geom = "path", fun = dnorm,
                n = 201,
                args = list(mean(finite_cpc$ln_cpc),
                            sd(finite_cpc$ln_cpc))) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())

finite_ctr <- filter(advertising_train,
                     is.finite(ln_ctr))

p_ctr <- ggplot(finite_ctr) +
  geom_density(aes(x = ln_ctr),
               fill = "yellow", alpha = 1/3) +
  stat_function(geom = "path", fun = dnorm,
                n = 201,
                args = list(mean(finite_ctr$ln_ctr),
                            sd(finite_ctr$ln_ctr))) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())

finite_impr <- filter(advertising_train,
                      is.finite(ln_impr))

p_impr <- ggplot(finite_impr) +
  geom_density(aes(x = ln_impr),
               fill = "yellow", alpha = 1/3) +
  stat_function(geom = "path", fun = dnorm,
                n = 201,
                args = list(mean(finite_impr$ln_impr),
                            sd(finite_impr$ln_impr))) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())

finite_y <- filter(advertising_train,
                   is.finite(ln_y))

p_y <- ggplot(finite_y) +
  geom_density(aes(x = ln_y),
               fill = "yellow", alpha = 1/3) +
  stat_function(geom = "path", fun = dnorm,
```

```
                n = 201,
                args = list(mean(finite_y$ln_y),
                            sd(finite_y$ln_y))) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())

grid.arrange(p_cpc, p_ctr,
             p_impr, p_y)
```



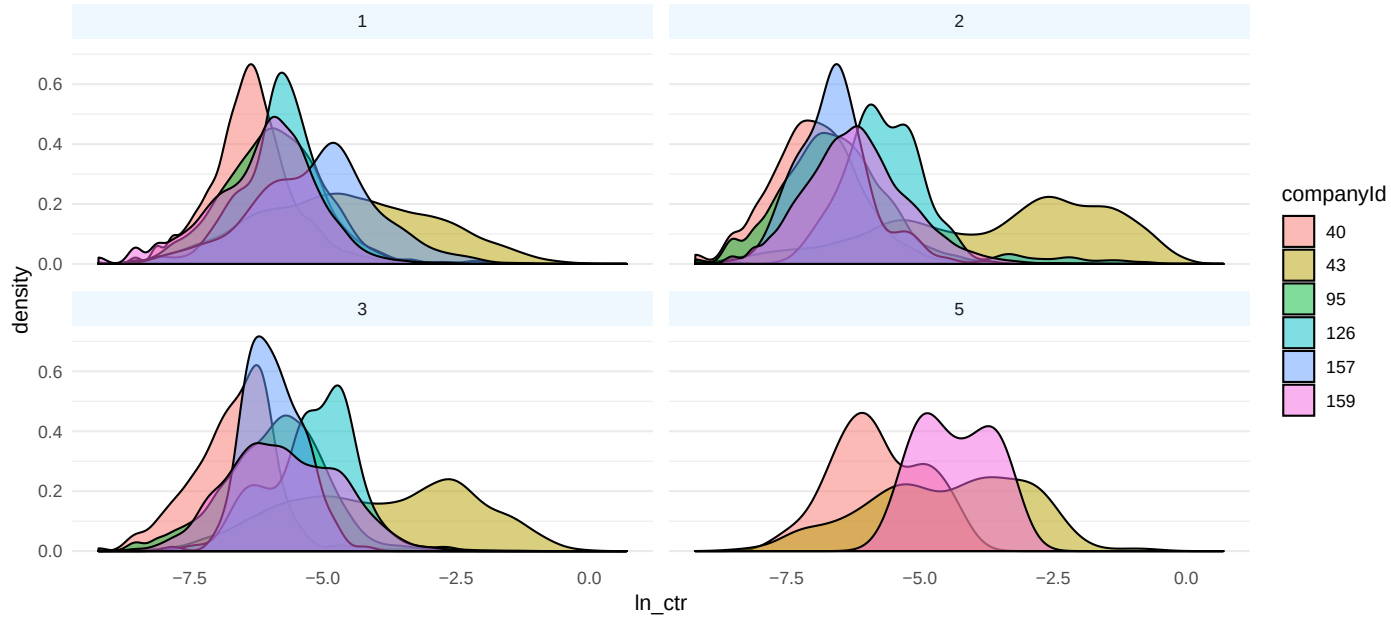

```
ggplot(advertising_train) +
  geom_density(aes(x = ln_ctr, fill = companyId),
               alpha = 1/2) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())
```

```
## Warning: Removed 78957 rows containing non-finite values (stat_density).
```




```
ggplot(advertising_train) +
  geom_density(aes(x = ln_ctr, fill = companyId),
               alpha = 1/2) +
  facet_rep_wrap(~deviceType) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        strip.background = element_rect(fill = "aliceblue",
                                        colour = NA))
```

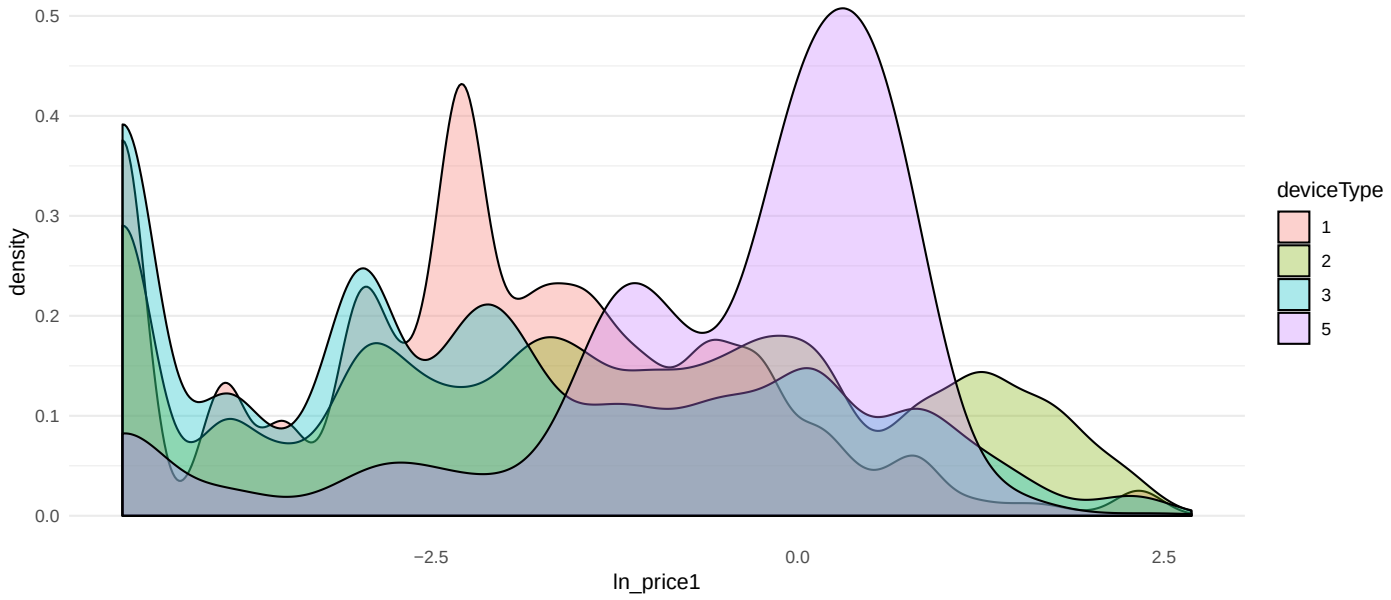
```
## Warning: Removed 78957 rows containing non-finite values (stat_density).
```




```
price_trans <- mutate(advertising_train,
                      "ln_price1" = log(price1),
                      "ln_price2" = log(price2),
                      "ln_price3" = log(price3))
```

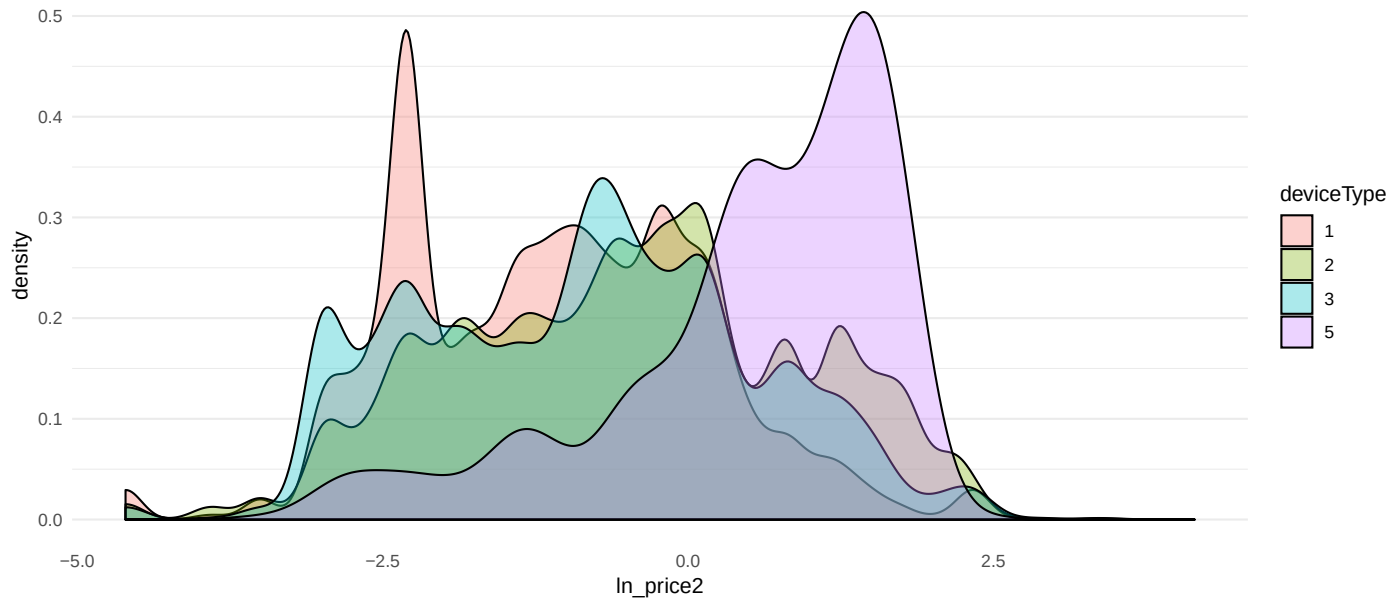
```
ggplot(price_trans) +
  geom_density(aes(x = ln_price1, fill = deviceType),
               alpha = 1/3) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())
```

```
## Warning: Removed 92892 rows containing non-finite values (stat_density).
```



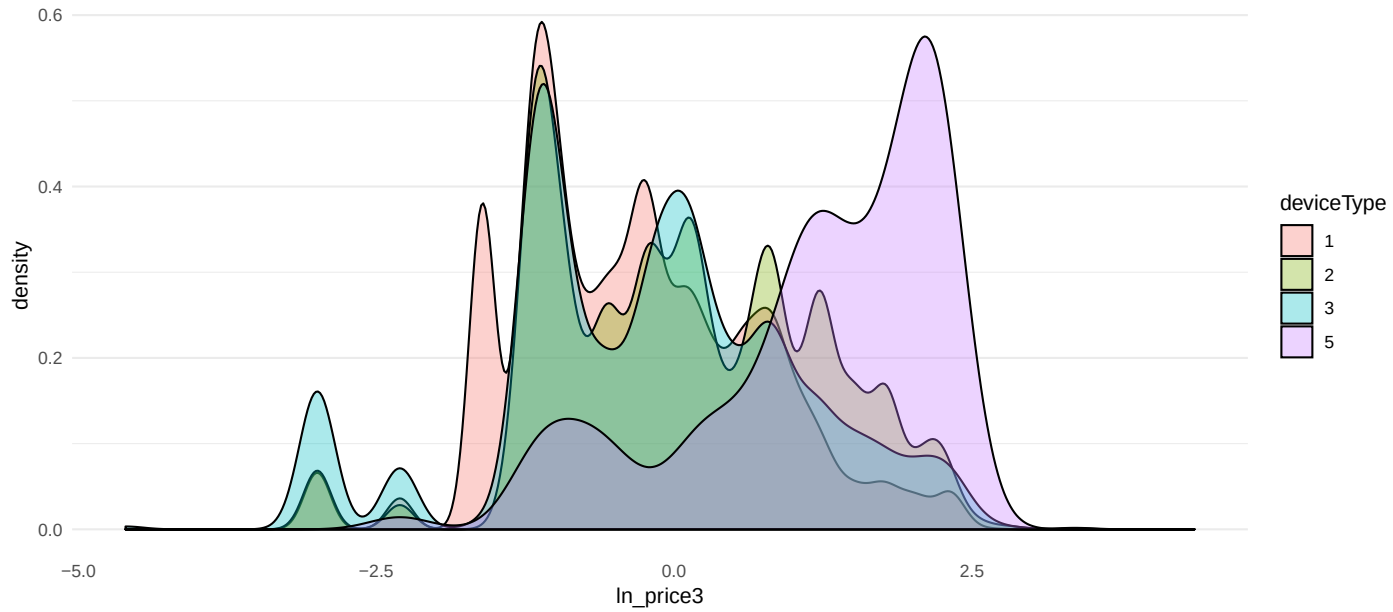

```
ggplot(price_trans) +
  geom_density(aes(x = ln_price2, fill = deviceType),
               alpha = 1/3) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())
```

```
## Warning: Removed 92804 rows containing non-finite values (stat_density).
```



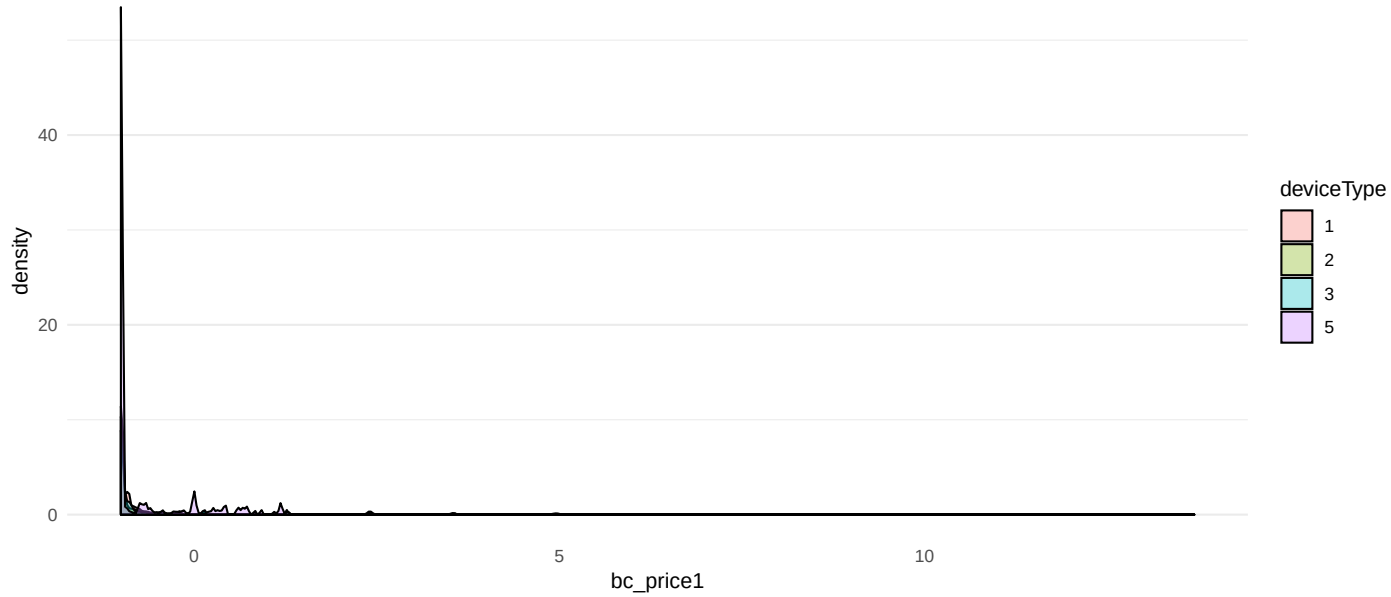

```
ggplot(price_trans) +
  geom_density(aes(x = ln_price3, fill = deviceType),
               alpha = 1/3) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())
```

```
## Warning: Removed 92804 rows containing non-finite values (stat_density).
```




```
boxcox <- function (x, lambda = 1) {

  (x^{lambda} - 1) /
  {lambda}

}
```

```r
boxcox_pos_price <- mutate(advertising_train,
                           "bc_price1" = boxcox(price1),
                           "bc_price2" = boxcox(price2),
                           "bc_price3" = boxcox(price3))

ggplot(boxcox_pos_price) +
  geom_density(aes(x = bc_price1, fill = deviceType),
               alpha = 1/3) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())
```

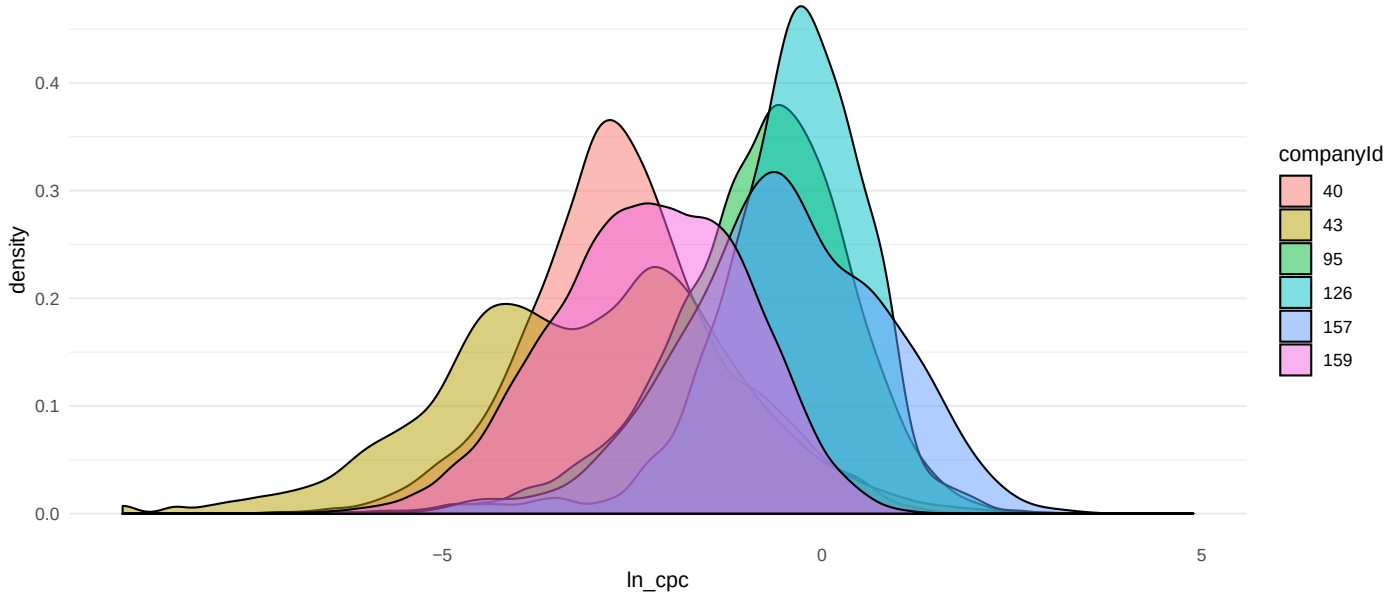



```r
box_grobs <- list()

for (i in 1:length(seq(0.1, 0.3, 0.025))) {

  j <- seq(0.1, 0.3, 0.025)[i]

  boxcox_price <- mutate(advertising_train,
                         "bc_price1" = boxcox(x = price1,
                                              lambda = j),
                         "bc_price2" = boxcox(x = price2,
                                              lambda = j),
                         "bc_price3" = boxcox(x = price3,
                                              lambda = j))

  box_grobs[[i]] <- grob(ggplot(boxcox_pos_price) +
    geom_density(aes(x = bc_price1, fill = deviceType),
                 alpha = 1/3) +
    ggtitle(paste("Lambda = ", j)) +
    theme_minimal() +
    theme(panel.grid.major.x = element_blank(),
          panel.grid.minor.x = element_blank())
```

```
  )

}

do.call(grid.arrange, box_grobs)
```

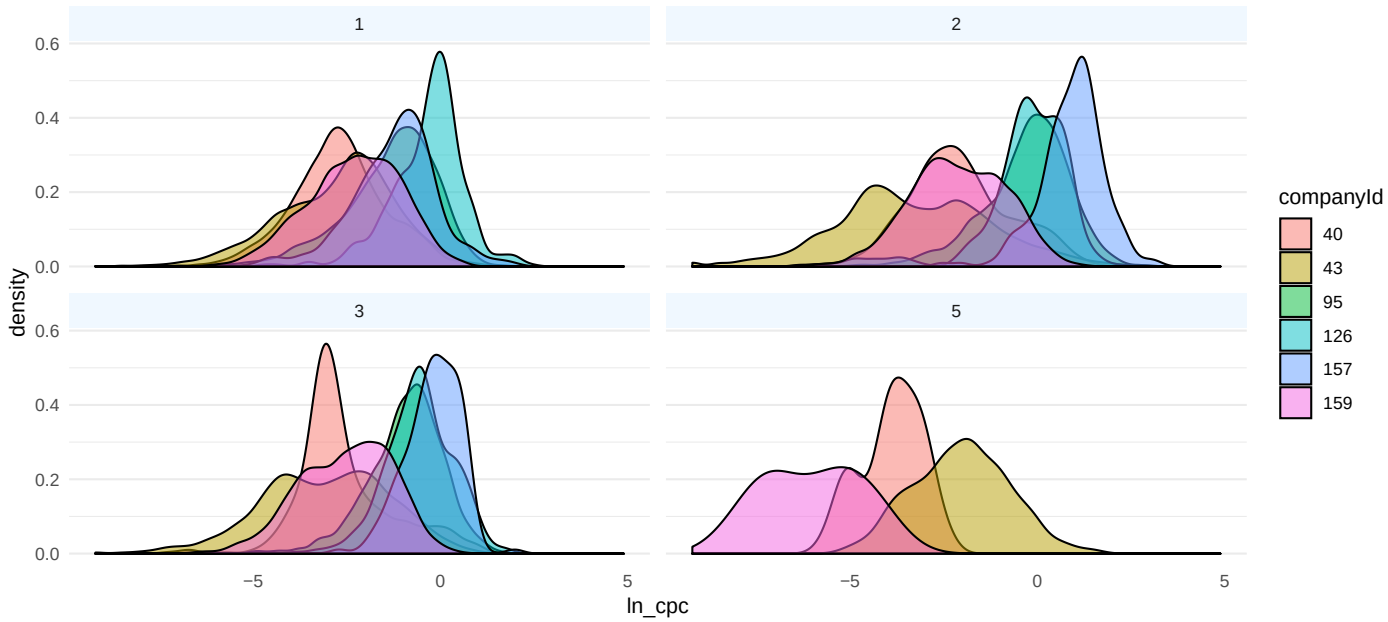
```
ggplot(advertising_train) +
  geom_density(aes(x = ln_cpc, fill = companyId),
               alpha = 1/2) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())
```

```
## Warning: Removed 78913 rows containing non-finite values (stat_density).
```



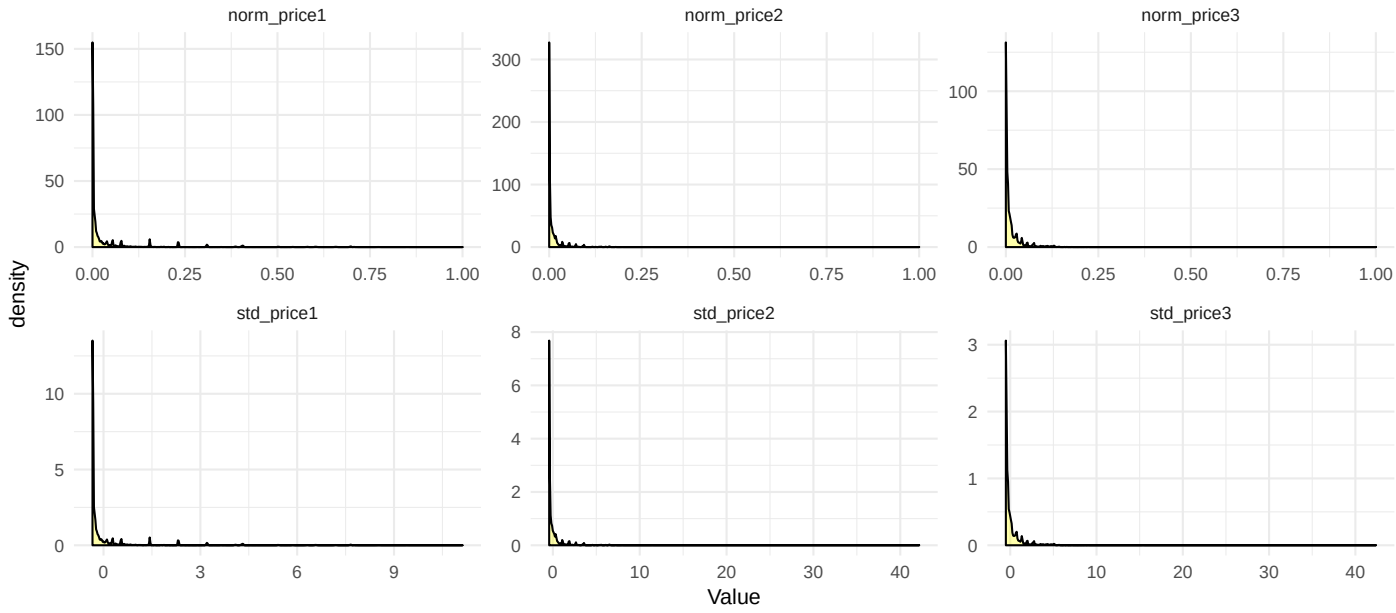
```
ggplot(advertising_train) +
  geom_density(aes(x = ln_cpc, fill = companyId),
               alpha = 1/2) +
  facet_rep_wrap(~deviceType) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        strip.background = element_rect(fill = "aliceblue",
                                        colour = NA))
```

```
## Warning: Removed 78913 rows containing non-finite values (stat_density).
```




```
normalise <- function(x) {

  (x - min(x)) /
    (max(x) - min(x)) *(1 - 0) +0

}

standardise <- function(x) {

  (x - mean(x)) /
    sd(x)

}

price_normal <- mutate(advertising_train,
                       norm_price1 = normalise(price1),
                       norm_price2 = normalise(price2),
                       norm_price3 = normalise(price3),
                       std_price1 = standardise(price1),
                       std_price2 = standardise(price2),
                       std_price3 = standardise(price3))
```

```
normal_price_long <- gather(price_normal,
                            norm_price1, norm_price2, norm_price3,
                            std_price1, std_price2, std_price3,
                            key = "Normalisation",
                            value = "Value")

ggplot(normal_price_long) +
  geom_density(aes(x = Value),
               fill = "yellow",
               alpha = 1/3) +
  facet_rep_wrap(~Normalisation,
                 ncol = 3,
                 scales = "free",
                 repeat.tick.labels = T) +
  theme_minimal()
```



1.3 References