# Data Understanding for Mercari Price Suggestion

*Markus Loide*

*10 detsember 2017*

## Reading the data

```
train_data <- fread("data/train.tsv", na.strings="")
```

```
##
Read 25.6% of 1482535 rows
Read 44.5% of 1482535 rows
Read 63.4% of 1482535 rows
Read 74.2% of 1482535 rows
Read 91.1% of 1482535 rows
Read 1482535 rows and 8 (of 8) columns from 0.315 GB file in 00:00:08
```

```
colnames(train_data)
```

```
## [1] "train_id"          "name"              "item_condition_id"
## [4] "category_name"     "brand_name"        "price"
## [7] "shipping"          "item_description"
```

Amount of rows in the training dataset

```
nrow(train_data)
```

```
## [1] 1482535
```

```
str(train_data)
```

```
## Classes 'data.table' and 'data.frame':   1482535 obs. of  8 variables:
##  $ train_id         : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ name             : chr  "MLB Cincinnati Reds T Shirt Size XL" "Razer BlackWidow Chroma Keyboard"
##  $ item_condition_id: int  3 3 1 1 1 3 3 3 3 3 ...
##  $ category_name    : chr  "Men/Tops/T-shirts" "Electronics/Computers & Tablets/Components & Parts"
##  $ brand_name       : chr  NA "Razer" "Target" NA ...
##  $ price            : num  10 52 10 35 44 59 64 6 19 8 ...
##  $ shipping         : int  1 0 1 1 0 0 0 1 0 0 ...
##  $ item_description : chr  "No description yet" "This keyboard is in great condition and works like
##  - attr(*, ".internal.selfref")=<externalptr>
```

### Name

Number of different names in the dataset

```
length(unique(train_data$name))
```

```
## [1] 1225273
```

Number of missing values

```
length(train_data[is.na(train_data$name), ])
```

```
## [1] 8
```

### Item condition

Different values for condition

```
length(unique(train_data$item_condition_id))
```

```
## [1] 5
```

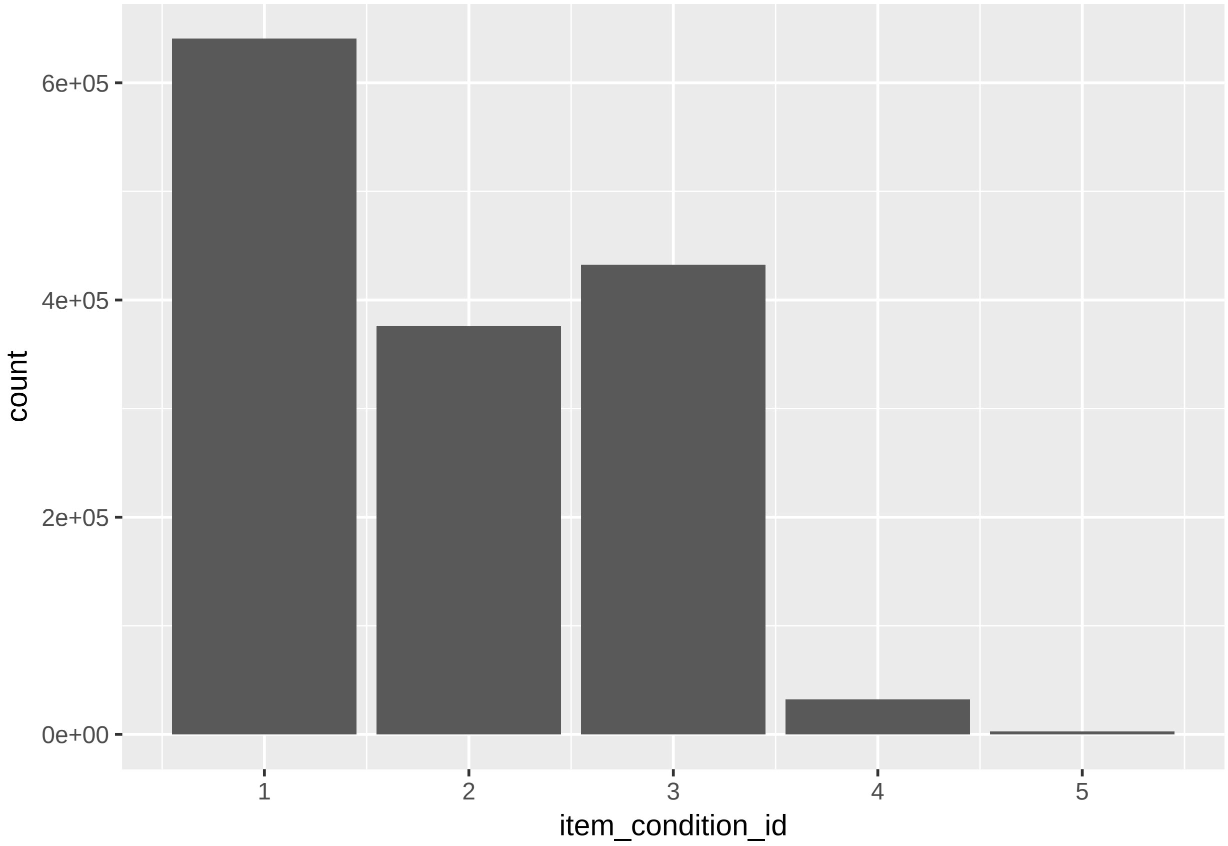
Missing values

```
nrow(train_data[is.na(train_data$item_condition_id), ])
```

```
## [1] 0
```

Distribution of item conditions

```
ggplot(train_data, aes(x = item_condition_id)) + geom_bar()
```



The distributions for the different values of item condition are heavily skewed towards to lower numbers. As such, it might be necessary to use sampling.

### Item category

Unique values

```
length(unique(train_data$category_name))
```

```
## [1] 1288
```

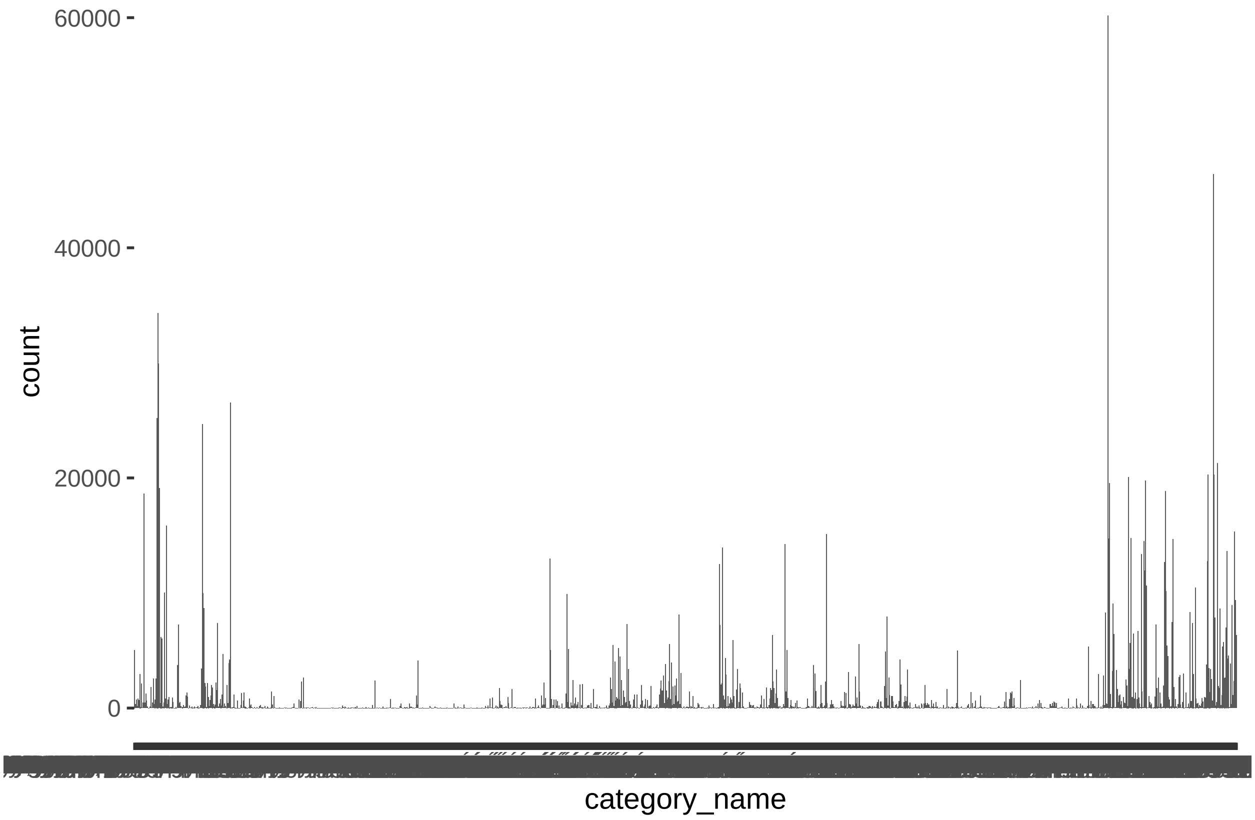
Missing values

```
nrow(train_data[is.na(train_data$category_name), ])
```

```
## [1] 6327
```

Distribution

```
ggplot(train_data, aes(x = category_name)) + geom_bar()
```



As with the last one, here some categories are represented very little. A solution for this can also be found with sampling.

## Brand name

```
length(unique(train_data$brand_name))
```

```
## [1] 4810
```

```
nrow(train_data[is.na(train_data$brand_name), ])
```

```
## [1] 632682
```

Average price depending whether or not brand name exists

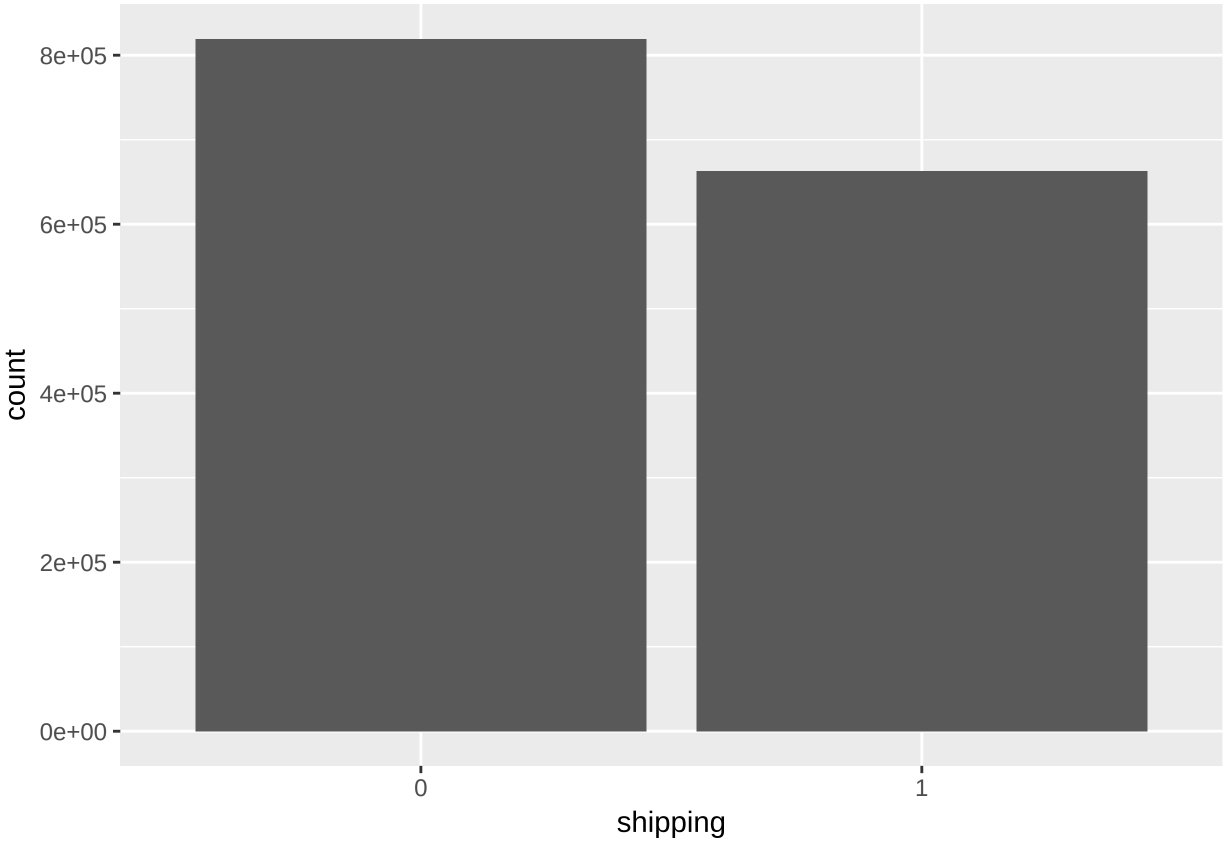
```
train_data %>%
  group_by(is.na(brand_name)) %>%
  summarise(average_price = mean(price), average_price_stdev = sd(price), max_price = max(price), min_pr
```

```
## # A tibble: 2 x 5
##   `is.na(brand_name)` average_price average_price_stdev max_price
##                 <lgl>         <dbl>               <dbl>     <dbl>
## 1               FALSE      30.90952            44.71224      2009
## 2                TRUE      21.13345            27.36126      2000
## # ... with 1 more variables: min_price <dbl>
```

### Shipping info

```
train_data$shipping <- as.factor(train_data$shipping)
```

```
ggplot(train_data, aes(x = shipping)) + geom_bar()
```



Pretty equal - good.

### Item description

Written in free form, no use even trying to see how many are different.

### Price

```
max(train_data$price)
```

```
## [1] 2009
```

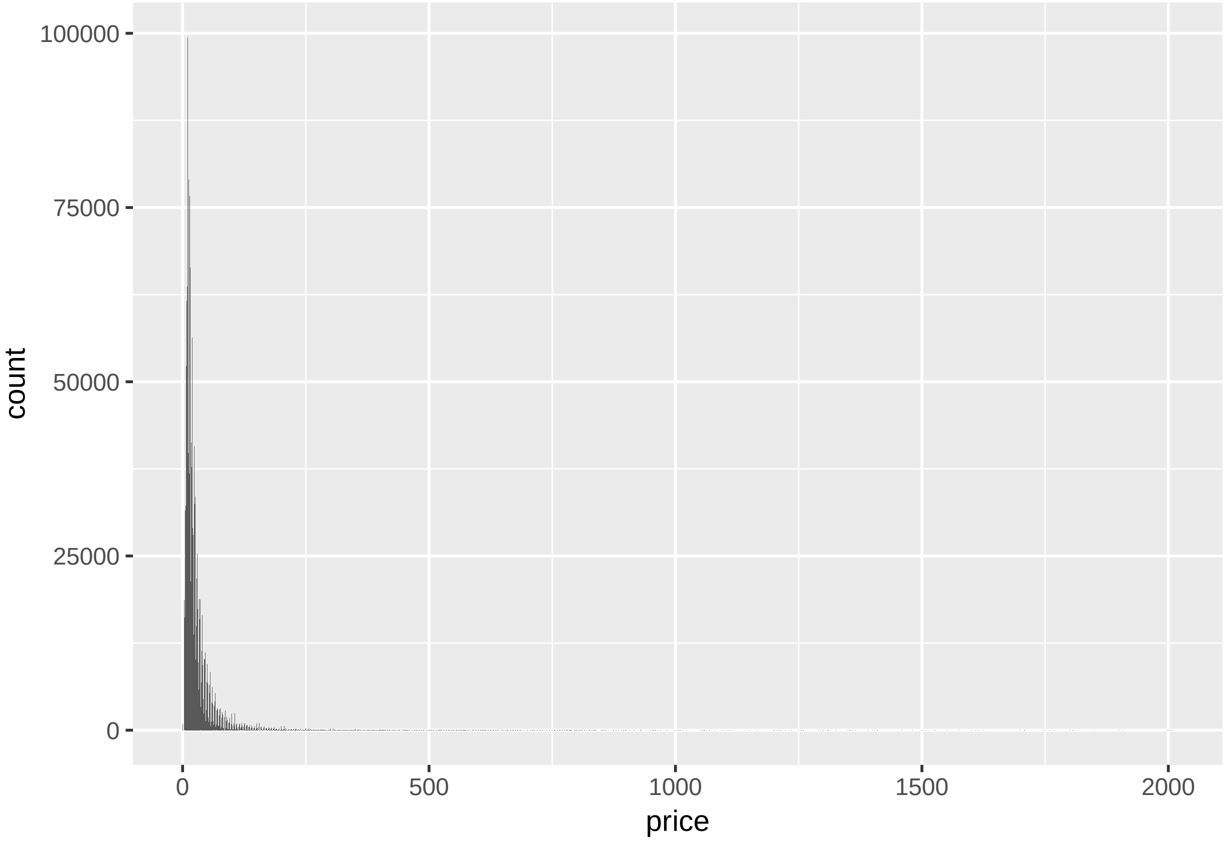
```
min(train_data$price)
```

```
## [1] 0
```

```
mean(train_data$price)
```

```
## [1] 26.73752
```

```
ggplot(train_data, aes(x = price)) + geom_bar()
```



Also a case for sampling perhaps.