results

Do celebrities have an advantage over KOLs (key opinion leader)? In particular, what's the revenue comparison between the two groups? We have a hypothesis that people come to celebrities' live streaming rooms because they like the person and come to KOLs' live streaming rooms because KOLs offer a higher discount or have expertise in the specific area.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

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

```
library(lubridate)
```

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

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggrepel)

livestream <- read_csv("../processed data/Data_Livestream_General_processed.csv")
```

```
## Rows: 1224 Columns: 22

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (9): Start_time, Duration, Fan Conversion Ratio, Sales_conversion_value...
## dbl (13): Session_id, Peak_viewers, Gifts_from_viewers, Number_of_products, ...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
livestream
```

```
## # A tibble: 1,224 x 22
##    Session_id Start_time    Peak_viewers Gifts_from_viewers Number_of_products
##         <dbl> <chr>                <dbl>              <dbl>              <dbl>
##  1          1 8/31/21 11:53        15042              19843                105
##  2          2 8/30/21 11:49        13639              12125                107
##  3          3 8/29/21 11:49        24133              27657                104
##  4          4 8/28/21 11:50        27482              37051                 89
##  5          5 8/27/21 11:51        17296              15501                 95
```

```
## 6          6 8/26/21 11:49       13674             26106                118
## 7          7 8/25/21 12:00       35163             35461                115
## 8          8 8/25/21 11:48           0                 0                 NA
## 9          9 8/24/21 11:53       21898             38878                111
## 10        10 8/23/21 11:48       13425                 0                 88
## # ... with 1,214 more rows, and 17 more variables: Number_of_goods_sold <dbl>,
## #   Total_sales_amount <dbl>, Duration <chr>, Views <dbl>,
## #   Average_number_of_online_viewers <dbl>, Number_of_likes <dbl>,
## #   Number_of_new_followers <dbl>, Fan Conversion Ratio <chr>,
## #   Number_of_new_fans <dbl>, Per_customer_transaction <dbl>, UV <dbl>,
## #   Sales_conversion_value_ratio <chr>, Average_length_of_stay <chr>,
## #   Name <chr>, Host_category <chr>, Occupation <chr>, Date <chr>
```

```r
#change date
livestream$Date <- as.Date(livestream$Date,format='%m/%d/%y')
livestream$Host_category<-as.factor(livestream$Host_category)



#group revenue by day, host
total_by_day<-livestream %>% group_by(Name, Host_category, Date) %>%
  dplyr::summarize(Sales_amount_total = sum(Total_sales_amount,na.rm=TRUE),n=n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Name', 'Host_category'. You can override using the `.groups` arg
```

```r
average<-total_by_day %>% group_by(Name, Host_category) %>%
  dplyr::summarize(Sales_amount_avg = mean(Sales_amount_total,na.rm=TRUE))
```

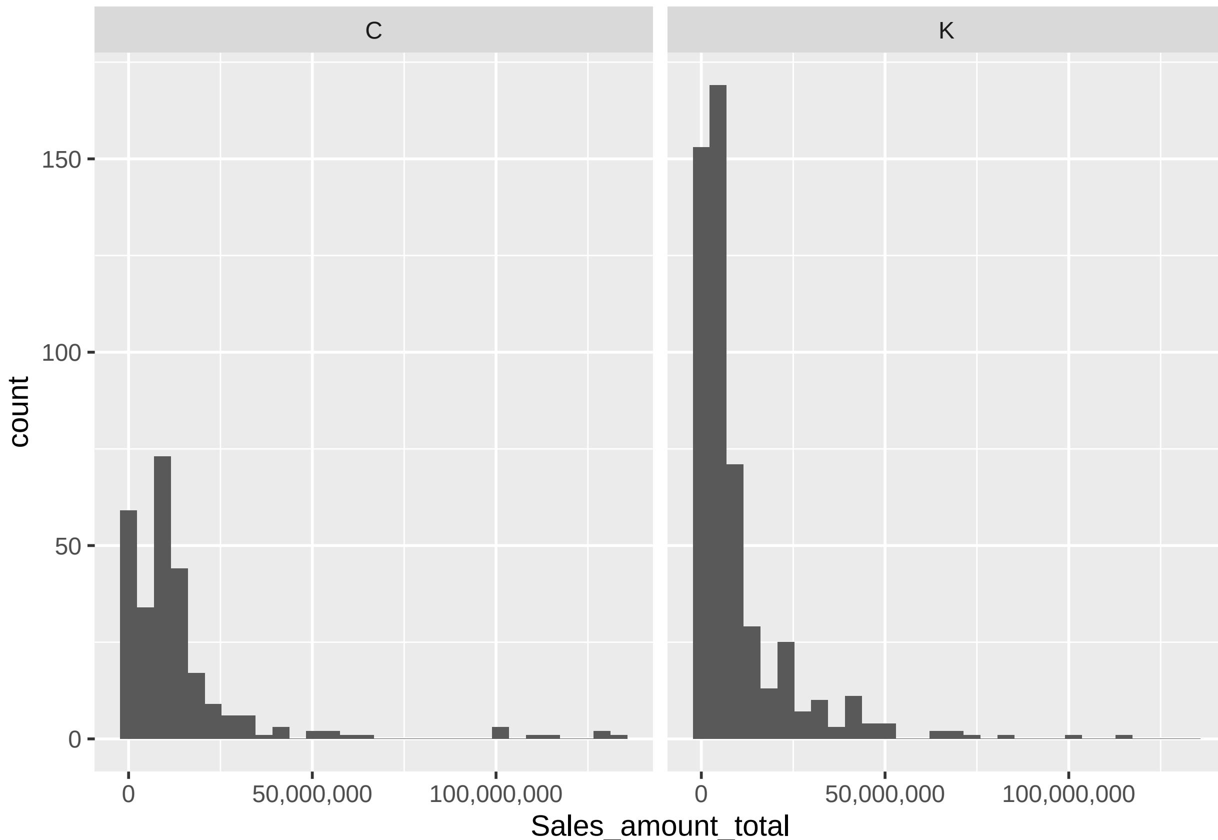
```
## `summarise()` has grouped output by 'Name'. You can override using the `.groups` argument.
```

```r
total<-total_by_day %>% group_by(Name, Host_category) %>%
  dplyr::summarize(Sales_amount_total = sum(Sales_amount_total,na.rm=TRUE),n=n()) %>%
  left_join(average, by=c("Name", "Host_category"))
```

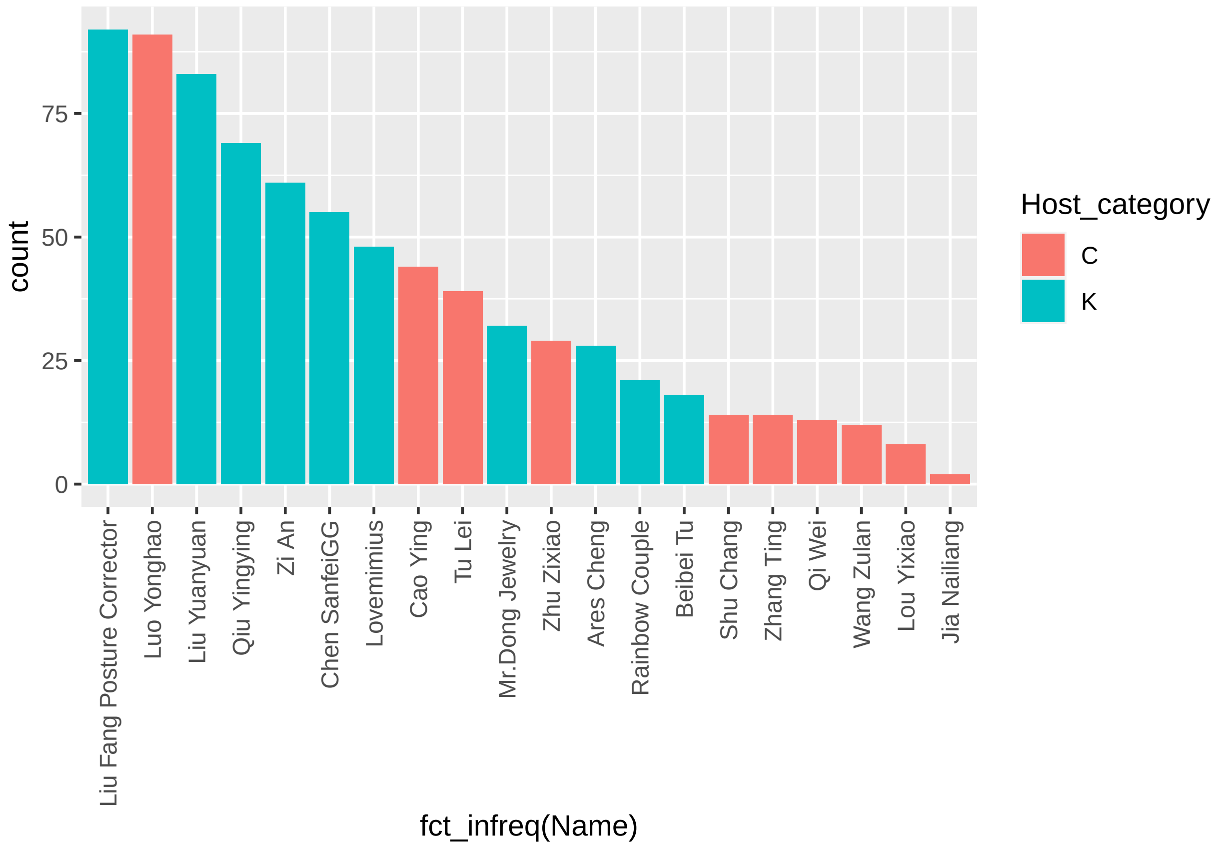
```
## `summarise()` has grouped output by 'Name'. You can override using the `.groups` argument.
```

```r
#sales distribution: KOLs host more streaming than c; KOLs sell more than c
ggplot(total_by_day, aes(Sales_amount_total)) +
geom_histogram()+
facet_wrap(~Host_category)+
scale_x_continuous(labels = scales::comma)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



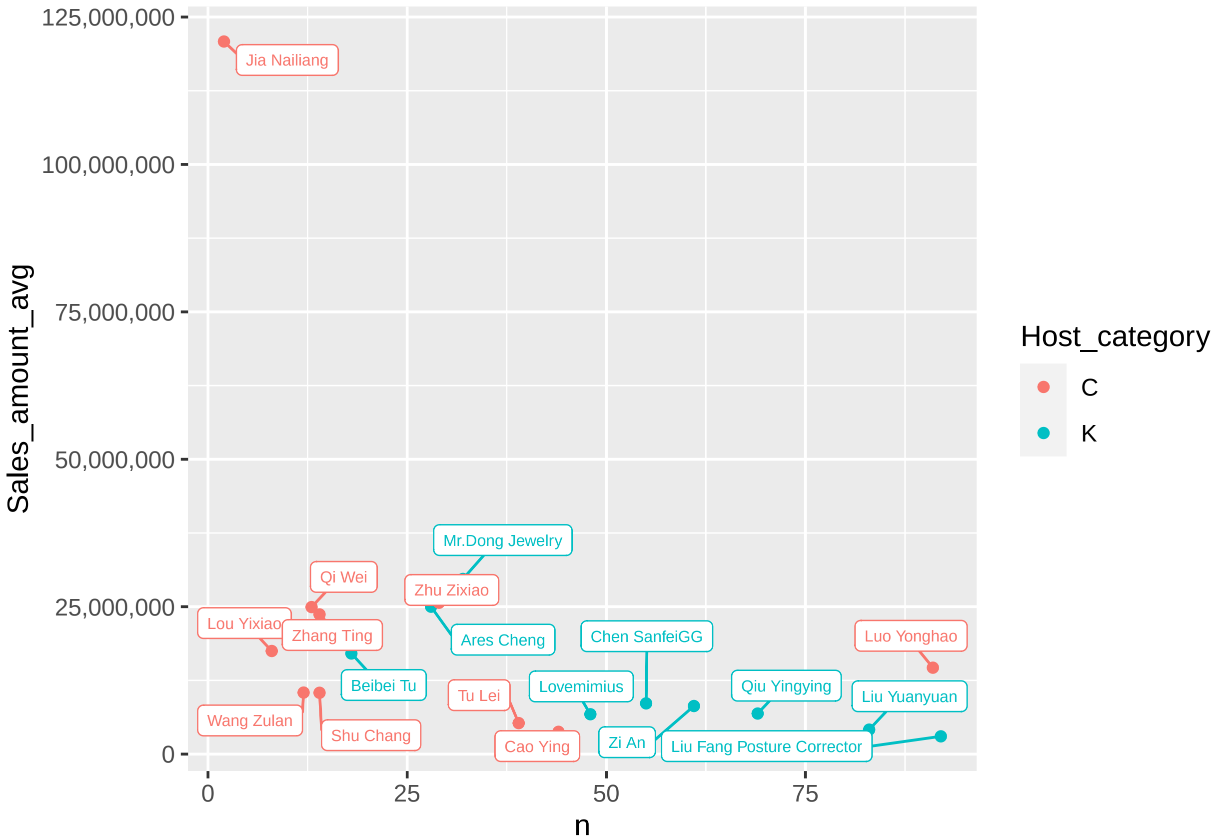

```
#total number of days steaming: KOLs host more streaming than c
ggplot(total_by_day, aes(fct_infreq(Name), fill=Host_category)) +
geom_bar()+
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
#total sales vs days of streaming: KOLs generate more revenue, Luo is an outlier
ggplot(total, aes(n, Sales_amount_total, color=Host_category)) +
geom_point()+
geom_label_repel(aes(label = Name), size = 2, min.segment.length = 0, max.overlaps = 9, show.legend=FALS
```


```
#avg sales vs days of streaming
ggplot(total, aes(n, Sales_amount_avg, color=Host_category)) +
geom_point()+
geom_label_repel(aes(label = Name), size = 2, min.segment.length = 0, max.overlaps = 9, show.legend=FALS
scale_y_continuous(labels = scales::comma)
```

```
## Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
#jia nailiang just starts in mid-aug and only has two days of livestreaming. outlier

#after removing Jia: in general, celebrities hold less streaming and have more avg
total%>%filter(Name!='Jia Nailiang') %>%
ggplot(aes(n, Sales_amount_avg, color=Host_category)) +
geom_point()+
geom_label_repel(aes(label = Name), size = 2, min.segment.length = 0, max.overlaps = 9, show.legend=FALS
scale_y_continuous(labels = scales::comma)
```


```r
#by categories
sales <-read_csv("../processed data/Data_Sales_processed2.csv")
```

```
## Rows: 22091 Columns: 12

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): Goods_category, Commission_rate, Conversion_rate, Name, Host_catego...
## dbl (6): Item_Id, Retailing_price_(YUAN), Sales_volume, Sales_amount, Short_...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
sales
```

```
## # A tibble: 22,091 x 12
##    Item_Id Goods_category      `Retailing_price_(Y~ Commission_rate Sales_volume
##      <dbl> <chr>                              <dbl> <chr>                  <dbl>
##  1       1 3C digital products                 5549 0.00%                   8045
##  2       2 3C digital products                 5546 0.00%                   4061
##  3       3 3C digital products                 6299 0.00%                   2835
##  4       4 3C digital products                 9039 0.00%                   1361
##  5       5 Jewelry Accessories                19000 1.00%                    590
##  6       6 3C digital products                8429. 0.00%                   1000
##  7       7 3C digital products                6699. 0.00%                   1211
##  8       8 Jewelry Accessories               19000. 1.00%                    408
##  9       9 Beauty skin care                    798. 4.00%                   8993
## 10      10 3C digital products                6259. 0.00%                   1000
```

```
## # ... with 22,081 more rows, and 7 more variables: Sales_amount <dbl>,
## #   Conversion_rate <chr>, Short_video <dbl>, Live_streaming <dbl>, Name <chr>,
## #   Host_category <chr>, Good_category_new <chr>
```

```
sum(is.na(sales))
```

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

```
#factor relevel
sales$Name<-as.factor(sales$Name)
name_order<-fct_reorder(total$Name,-total$Sales_amount_total)%>%levels()
sales$Name<-fct_relevel(sales$Name,name_order)
levels(sales$Name)
```

```
##  [1] "Luo Yonghao"                 "Mr.Dong Jewelry"
##  [3] "Zhu Zixiao"                  "Ares Cheng"
##  [5] "Zi An"                       "Qiu Yingying"
##  [7] "Chen SanfeiGG"               "Rainbow Couple"
##  [9] "Liu Yuanyuan"                "Zhang Ting"
## [11] "Lovemimius"                  "Qi Wei"
## [13] "Beibei Tu"                   "Liu Fang Posture Corrector"
## [15] "Jia Nailiang"                "Tu Lei"
## [17] "Cao Ying"                    "Shu Chang"
## [19] "Lou Yixiao"                  "Wang Zulan"
```

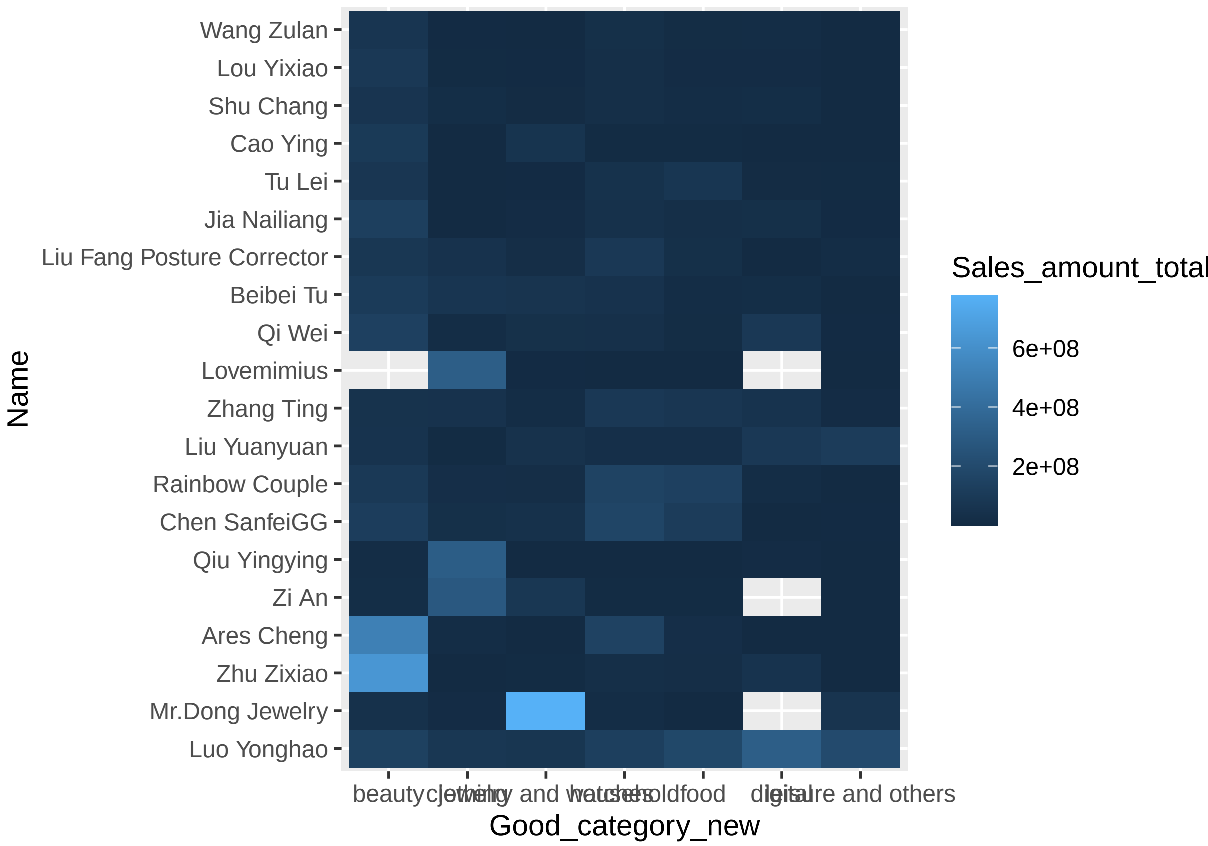
```
sales$Good_category_new<-as.factor(sales$Good_category_new)
sales_total<-sales %>% group_by(Good_category_new) %>%
  summarize(Sales_volume_total = sum(Sales_volume), Sales_amount_total = sum(Sales_amount))
good_order<-fct_reorder(sales_total$Good_category_new, -sales_total$Sales_amount_total)%>%levels()
sales$Good_category_new<-fct_relevel(sales$Good_category_new,good_order)
levels(sales$Good_category_new)
```

```
## [1] "beauty"              "clothing"            "jewelry and watches"
## [4] "household"           "food"                "digital"
## [7] "leisure and others"
```

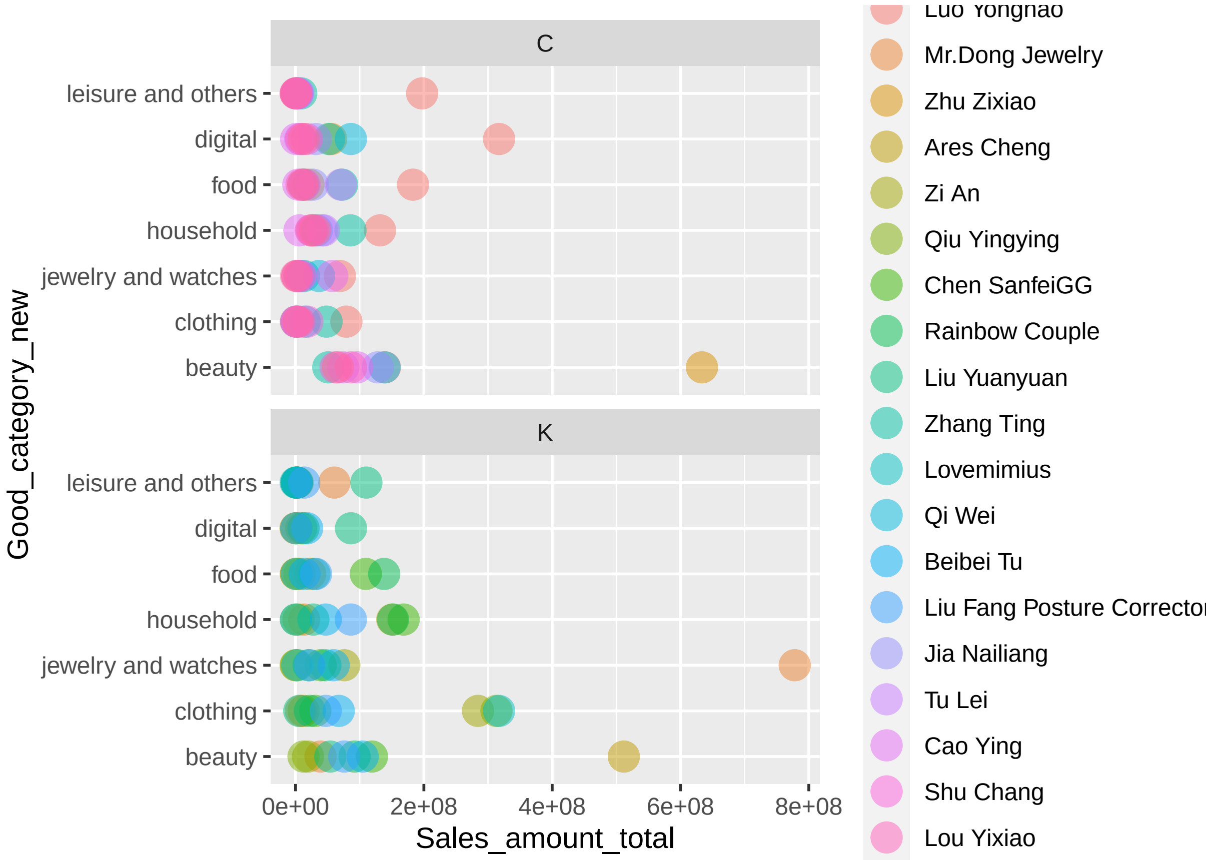
```
#aggregate ccategory total
sales2<-sales %>% group_by(Name,Host_category, Good_category_new) %>%
  summarize(Sales_volume_total = sum(Sales_volume), Sales_amount_total = sum(Sales_amount))
```

```
## `summarise()` has grouped output by 'Name', 'Host_category'. You can override using the `.groups` ar
```

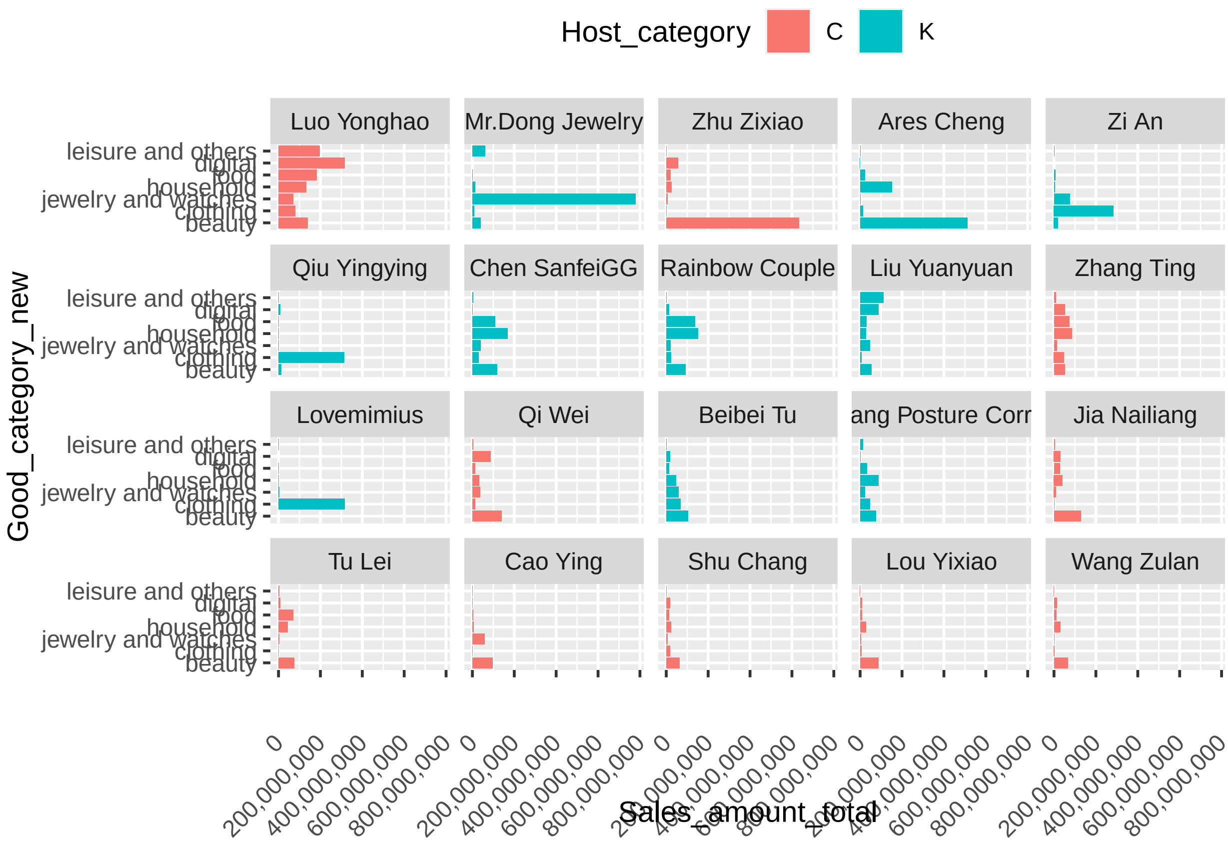
```
#EDAV, not use
sales2%>%
ggplot(aes( Good_category_new,Name, fill=Sales_amount_total)) +
geom_tile()
```



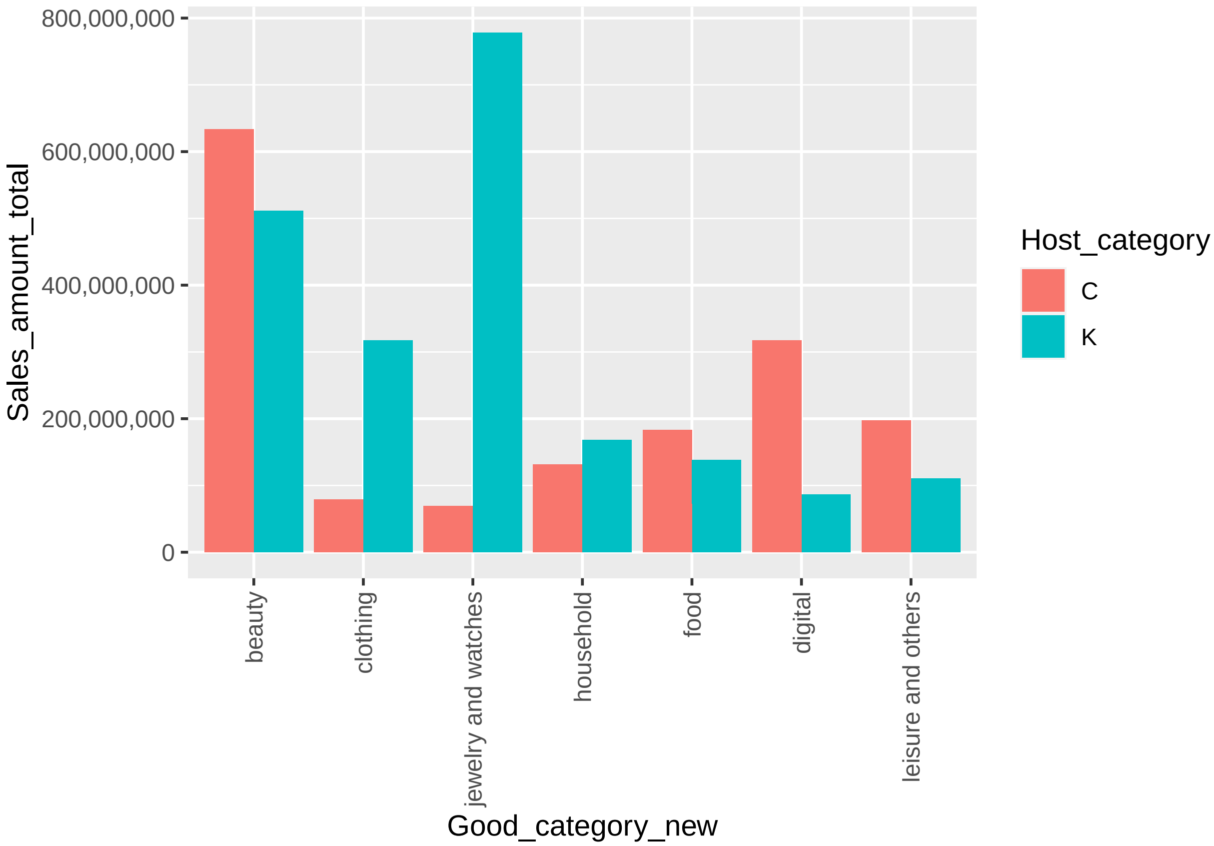

```r
#not use
ggplot(sales2, aes(x = Good_category_new, y = Sales_amount_total, color = Name)) +
  geom_point(size = 5, alpha = .5) +
  coord_flip() +
  facet_wrap(~Host_category, ncol = 1)
```



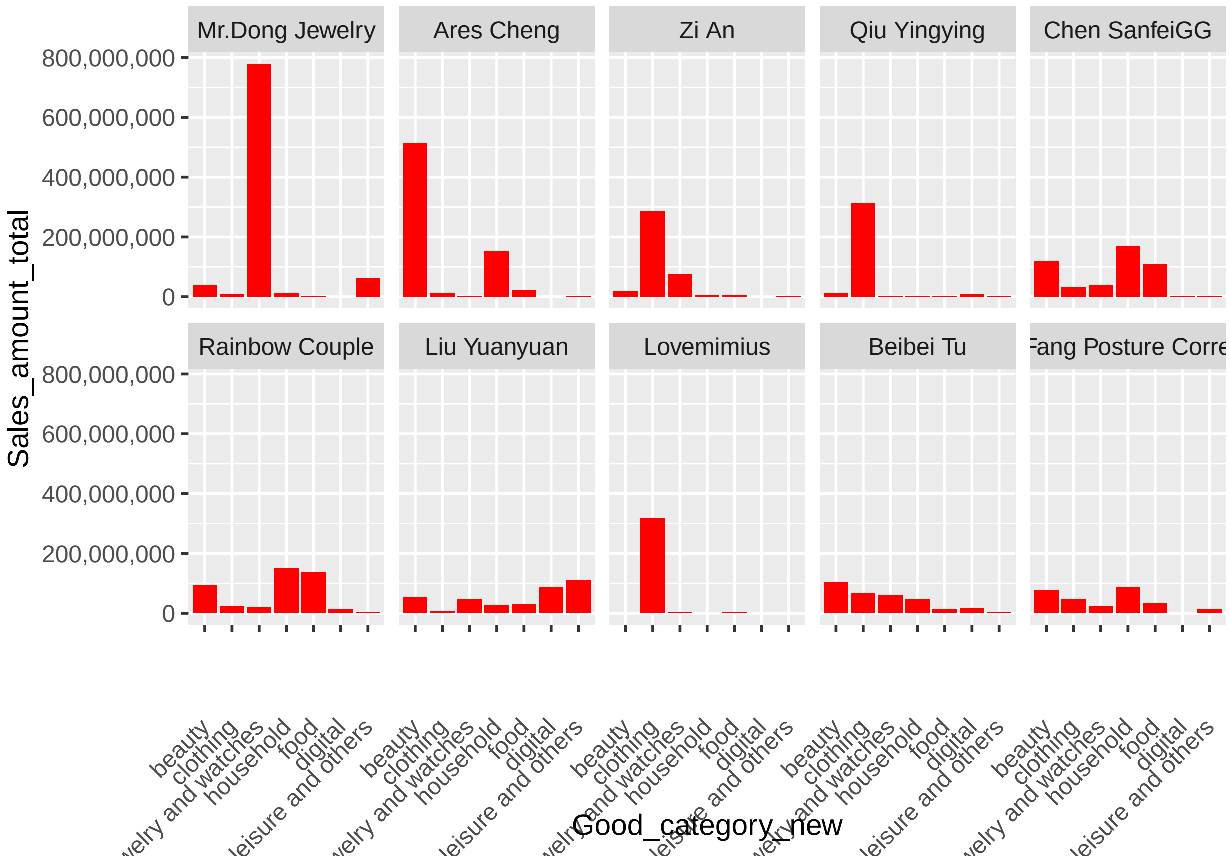

```
 # scale_color_manual(values = icecreamcolors) +
  #theme_dotplot

#revenue by host, good category, not clear info
ggplot(sales2, aes(Sales_amount_total, Good_category_new,fill = Host_category)) +
  geom_col() +
  facet_wrap(~Name, nrow=4)+
  scale_x_continuous(labels = scales::comma)+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1))+
  theme(legend.position = "top")
```



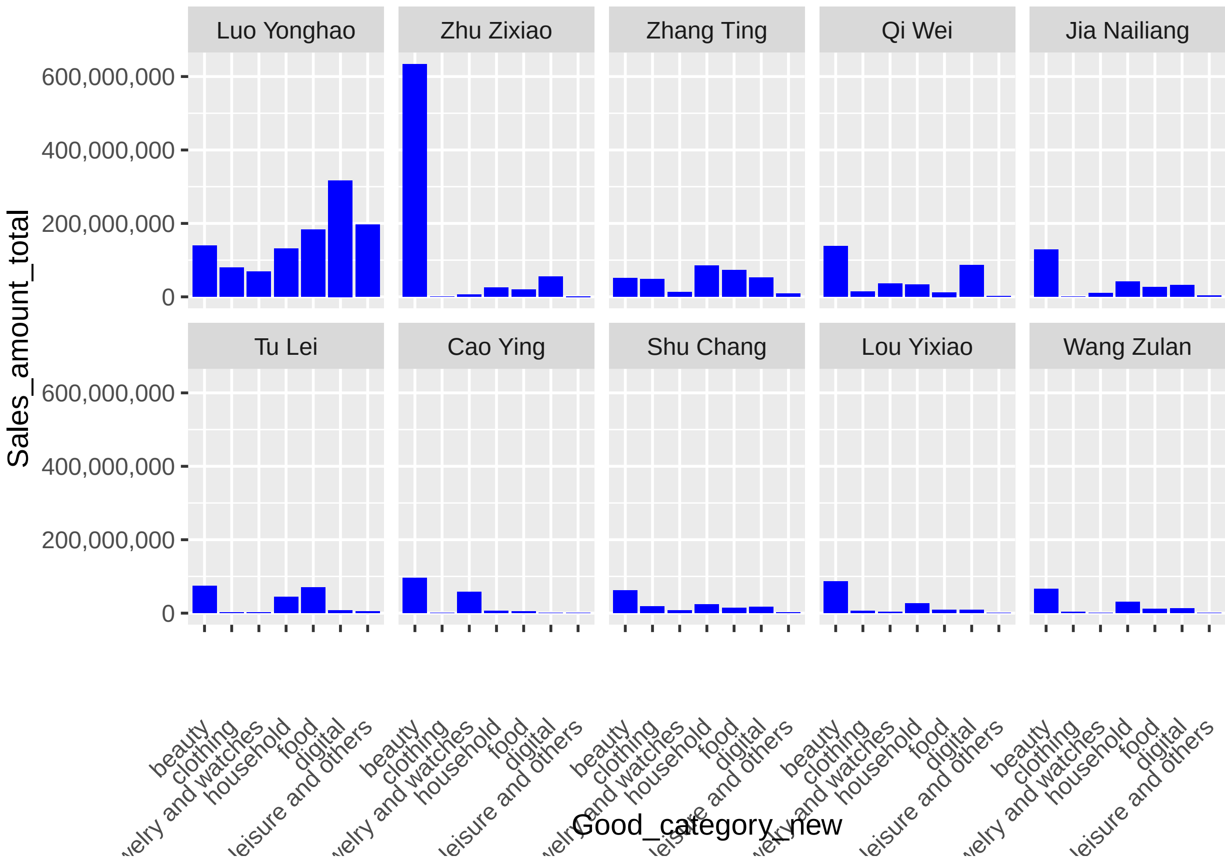

```
sales2%>%
ggplot(aes( Good_category_new, Sales_amount_total, fill= Host_category)) +
geom_col(position = "dodge")+
scale_y_continuous(labels = scales::comma)+
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
#K is generally good at 1 or 2 categories according to their expertise.
sales2%>%
filter(Host_category=='K')%>%
  ggplot(aes(Good_category_new, Sales_amount_total)) +
  geom_col(fill = 'red') +
  facet_wrap(~Name, nrow=2)+
  scale_y_continuous(labels = scales::comma)+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1))
```




```
#beauty is the most popular category in celebraies. Other categories is pretty even
sales2%>%
filter(Host_category=='C')%>%
  ggplot(aes(Good_category_new,Sales_amount_total)) +
  geom_col(fill = 'blue') +
  facet_wrap(~Name, nrow=2)+
  scale_y_continuous(labels = scales::comma)+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1))
```



From 2021/06 to 2021/08, among top20 hosts, KOL generated more revenue than celebrities.

Do celebrities have an advantage over KOLs (key opinion leader)? In particular, what's the revenue comparison between the two groups? We have a hypothesis that people come to celebrities' live streaming rooms because they like the person and come to KOLs' live streaming rooms because KOLs offer a higher discount or have expertise in the specific area. We will focus on the top 20 hosts' sales data in the past quarter to answer the questions. We will first define people to be celebrities if they are famous actors/actresses/singers based on our common judgments. Then we will look at their sales data and test our hypothesis. look at sales in different cateogry