R Project

Krishna Hemant, Snehal Rajwar, Ashok

01/03/2022

install.packages("webshot") webshot::install_phantomjs() tinytex::install_tinytex()

```
sampled <- read.csv("E:\\Comp&Viz\\sampled.csv")
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(ggplot2)
```

```
## Warning in register(): Can't find generic 'scale_type' in package ggplot2 to
## register S3 method.
```

```
library(dplyr)
library(tidyr)
library(gridExtra)
library(lubridate)
library(plotly)
library(treemapify)
library(scales)
library(ggalluvial)
library(RColorBrewer)
library(forcats)
library(treemap)
library(hrbrthemes)
```

```
#Forming the sampled data frame
df <- sampled %>% select(country,price) %>% group_by(country) %>% summarise(total_price = sum(price)) %
df1<- sampled %>% select(month,country,price) %>% group_by(month,country) %>% summarise(total_price = su
```

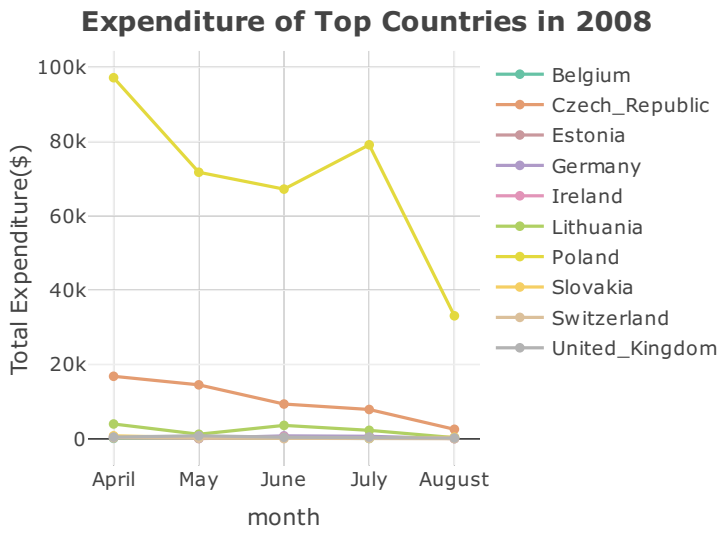
```
## 'summarise()' has grouped output by 'month'. You can override using the
## '.groups' argument.
```

```r
#Filtering domain names from countries and separating both
domains <- c('net(*.net)','com(*.com)','int(*.int)','org(*.org)','biz(*.biz)')

domain_df <- df %>% filter((df$country %in% domains))

country_total <- df %>% filter(!(df$country %in% domains))
```

```r
#line plot for various countries over four months of 2008 and the price spent
h <- head(country_total,10)
vec <- as.vector(h$country)

#sorting months in order
dd<- ungroup(df1)
x <- c("April","May",'June','July','August')
dd <- dd %>% filter(dd$country %in% vec) %>% mutate(month =  factor(month, levels = x)) %>% arrange(mon


#A line plot to show the total amount spent by the top 10 countries through the months of 2008

#This data in the plot shows that Poland has the highest expenditure since it's a Polish E-commerce com
plot_ly(
        data = dd,
        x = ~month,
        y = ~total_price,
        color = ~country,
        type = "scatter",
        mode = "lines+markers"
) %>% layout(title = "<b>Expenditure of Top Countries in 2008</b>",yaxis = list(title = ' Total Expendi
```

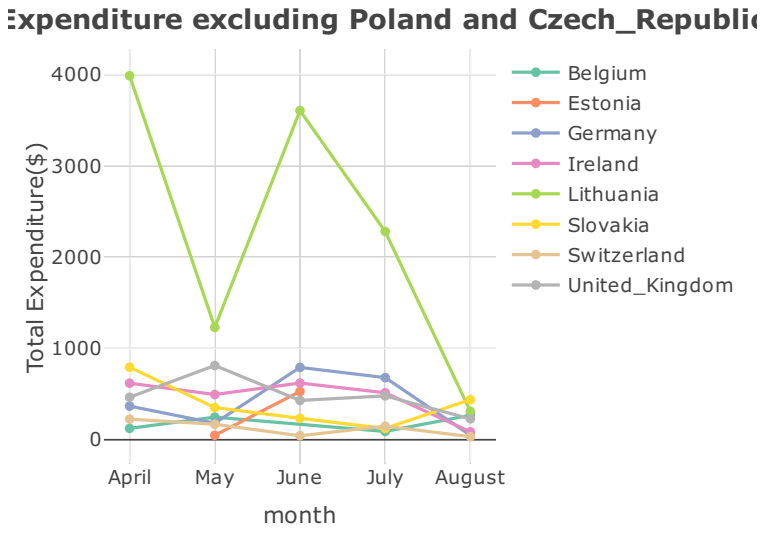
```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors
```

```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors
```




```
#Without Poland and Czech_Republic
# This plot shows a more zoomed in version of the previous plot excluding the top two countries.Since t

dd1 <- dd %>% filter(!(dd$country %in% c('Poland','Czech_Republic')))

plot_ly(
        data = dd1,
        x = ~month,
        y = ~total_price,
        color = ~country,
        type = "scatter",
        mode = "lines+markers"
) %>% layout(title = "<b>Expenditure excluding Poland and Czech_Republic</b>", yaxis = list(title = ' T
```

```
## Warning: 'layout' objects don't have these attributes: 'Legend'
## Valid attributes include:
## '_deprecated', 'activeshape', 'annotations', 'autosize', 'autotypenumbers', 'calendar', 'clickmode',
```

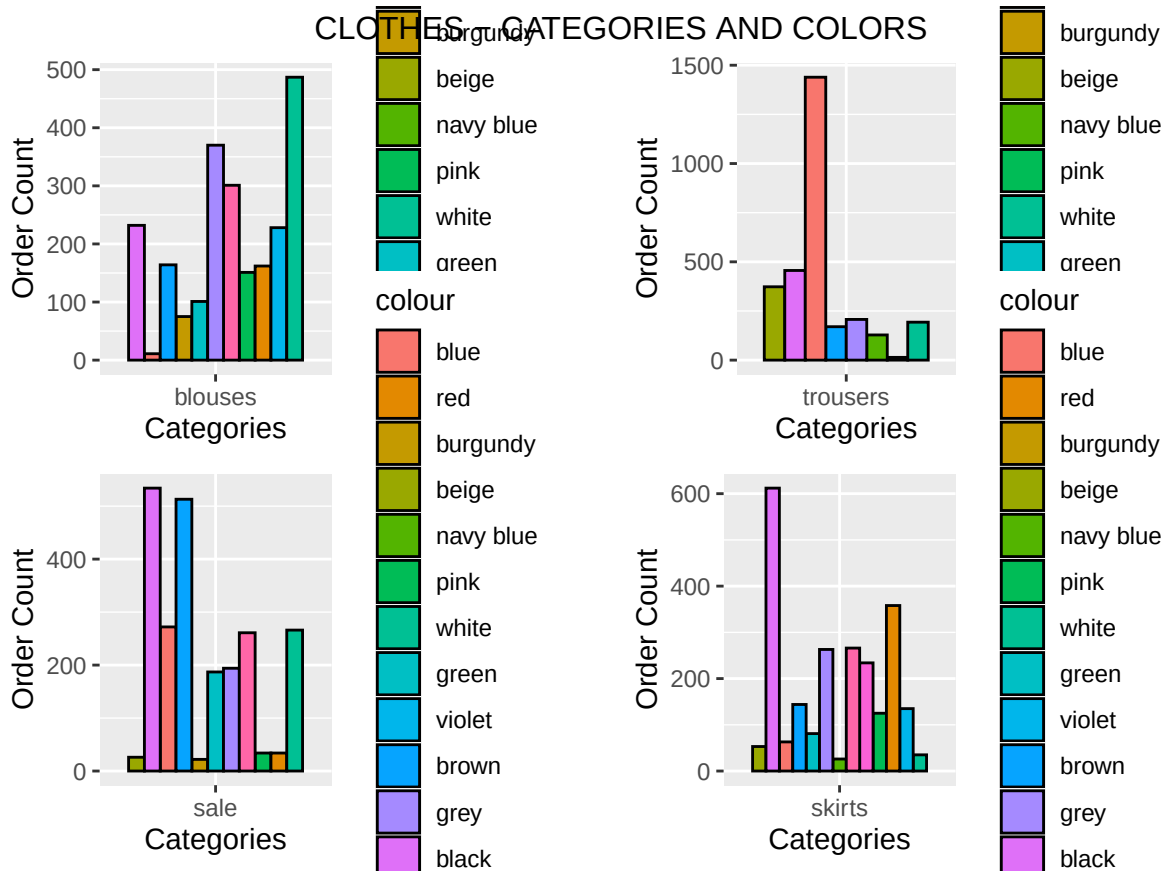



```
#Bar plot, categories vs colors
#Data frame for bar



def <- sampled %>% group_by(page.1..main.category.,colour) %>% summarise(count = n()) %>% arrange(count)
```

```
## `summarise()` has grouped output by 'page.1..main.category.'. You can override
## using the `.groups` argument.
```

```
def<- def %>% group_by(page.1..main.category.) %>% arrange(count)

gg_color_hue <- function(n) {
  hues = seq(15, 375, length = n + 1)
  hcl(h = hues, l = 65, c = 100)[1:n]
}

s = unique(def$colour)
cols = setNames(gg_color_hue(length(s)), s)


blouse_df <- filter(def, page.1..main.category. == 'blouses') %>% arrange(count)
sale_df <- filter(def, page.1..main.category. == 'trousers')%>% arrange(count)
skirt_df <- filter(def, page.1..main.category. == 'sale')%>% arrange(count)
trouser_df <- filter(def, page.1..main.category. == 'skirts')%>% arrange(count)
```

```
#White blouses seem to have the most orders, followed by Grey
blouse<- ggplot(blouse_df, aes(x = page.1..main.category., y = count, fill = colour )) +
        geom_bar(stat="identity", colour = 'black', position = 'dodge')  +xlab("Categories") + ylab("Or

#Brown stands out for sale than other categories with a lot of orders
sale <- ggplot(sale_df, aes(x = page.1..main.category., y = count, fill = colour)) +
  geom_bar(stat="identity", colour = 'black', position = 'dodge')  +xlab("Categories") + ylab("Order Co

#Black seems to be a preferred color for skirts, and red too has a high preference
skirt <- ggplot(skirt_df, aes(x = page.1..main.category., y = count, fill = colour )) +
  geom_bar(stat="identity", colour = 'black', position = 'dodge') +xlab("Categories") + ylab("Order Cou

#Blue trousers has the highest sales by a huge margin as it would have mostly been Jeans, followed by b
trouser <- ggplot(trouser_df, aes(x = page.1..main.category., y = count, fill = colour )) +
  geom_bar(stat="identity", colour = 'black', position = 'dodge') +xlab("Categories") + ylab("Order Cou

grid.arrange(blouse,sale,skirt,trouser, nrow = 2, ncol = 2, top = "CLOTHES - CATEGORIES AND COLORS")
```




```
#The code block doesn't display plot properly, can be viewed outside codeblock for full result
```

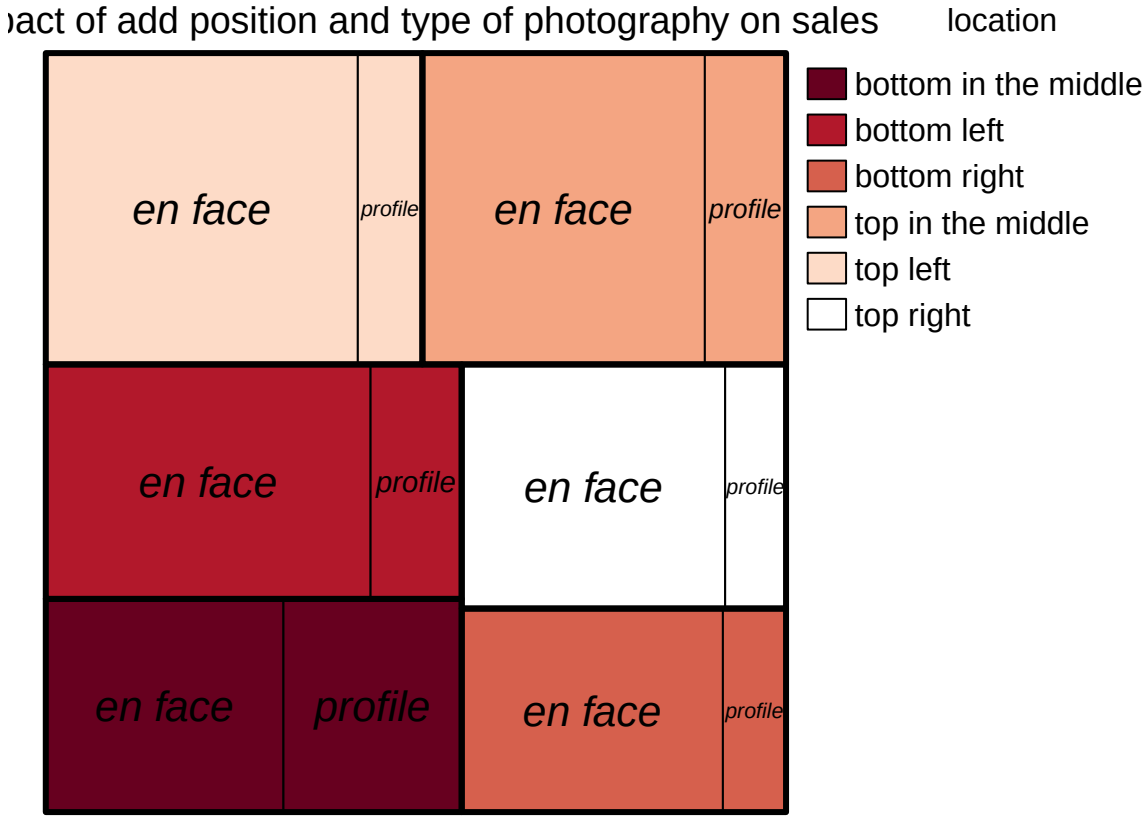
```
#The placement of advertisement along with type of photograph used is effecting the sales of the produc
#en face which is a head shot or face-focused photography get's the maximum attention of the user
#The larger box tells us the total sales for that location of ad on the webpage and the division classi
#based on the type of photography
```

```r
#Top left seems to be the best position of the photo with the worst position being bottom right
data<-sampled%>%group_by(location,model.photography)%>%
  summarise (total_value=sum(price))
```

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

```r
  treemap(data, #Your data frame object
          index=c("location","model.photography"),
          vSize = "total_value",
          vColor= "location",
          type="categorical",
          fontsize.labels=c(0,16),
          fontcolor.labels=c("white","Black"),
          fontface.labels=c(2,3),
          bg.labels=c("transparent"),
          align.labels=list(
            c("center", "center"),
            c("center", "center")
          ),
          overlap.labels=0.5,
          inflate.labels=F,


          palette = "RdGy",
          title="Impact of add position and type of photography on sales",
          fontsize.title = 14

  )
```

```
## Warning in if (class(try(col2rgb(bg.labels), silent = TRUE)) == "try-error")
## stop("Invalid bg.labels"): the condition has length > 1 and only the first
## element will be used
```

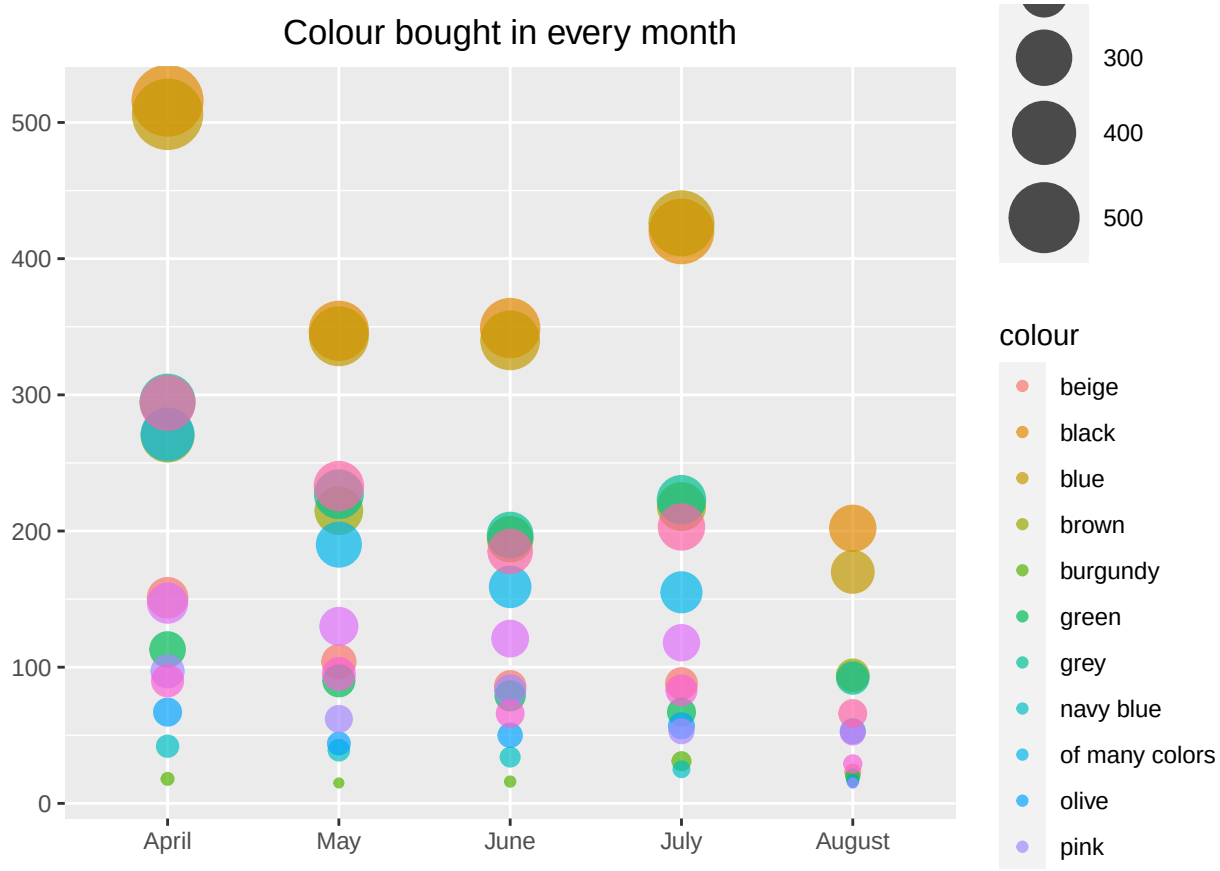



```
#Black and blue are consistently most bought by all the customers throughout the months which means lau
#in the color have higher chances of increasing sales.
#Colors like white,beige, are more popular in summer probably because of being cooler and absorbing les
#Besides some consistent fashion favorite must haves like brown,black,blue .The change climate affects
#which the company can keep in mind in launching products every month and have higher chances of succes

data_df<-sampled%>%group_by(month,colour)%>% summarise(total=n())
```

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

```
  x<- c("April","May",'June','July','August')
data_df <- data_df %>%  mutate(month = factor(month, levels = x)) %>% arrange(month)

ggplot(data_df, aes(x = data_df$month , y = data_df$total,
                    size = data_df$total,
                    color = colour))+
  theme(axis.title.y = element_blank()) + theme(axis.title.x=element_blank())+

  geom_point(alpha = 0.7)+


  scale_size(range = c(0.8, 12), name = "Total clothes ")+
  ggtitle("Colour bought in every month")+
```

```
  # code to center the title which is left aligned
  # by default
  theme(plot.title = element_text(hjust = 0.5))
```

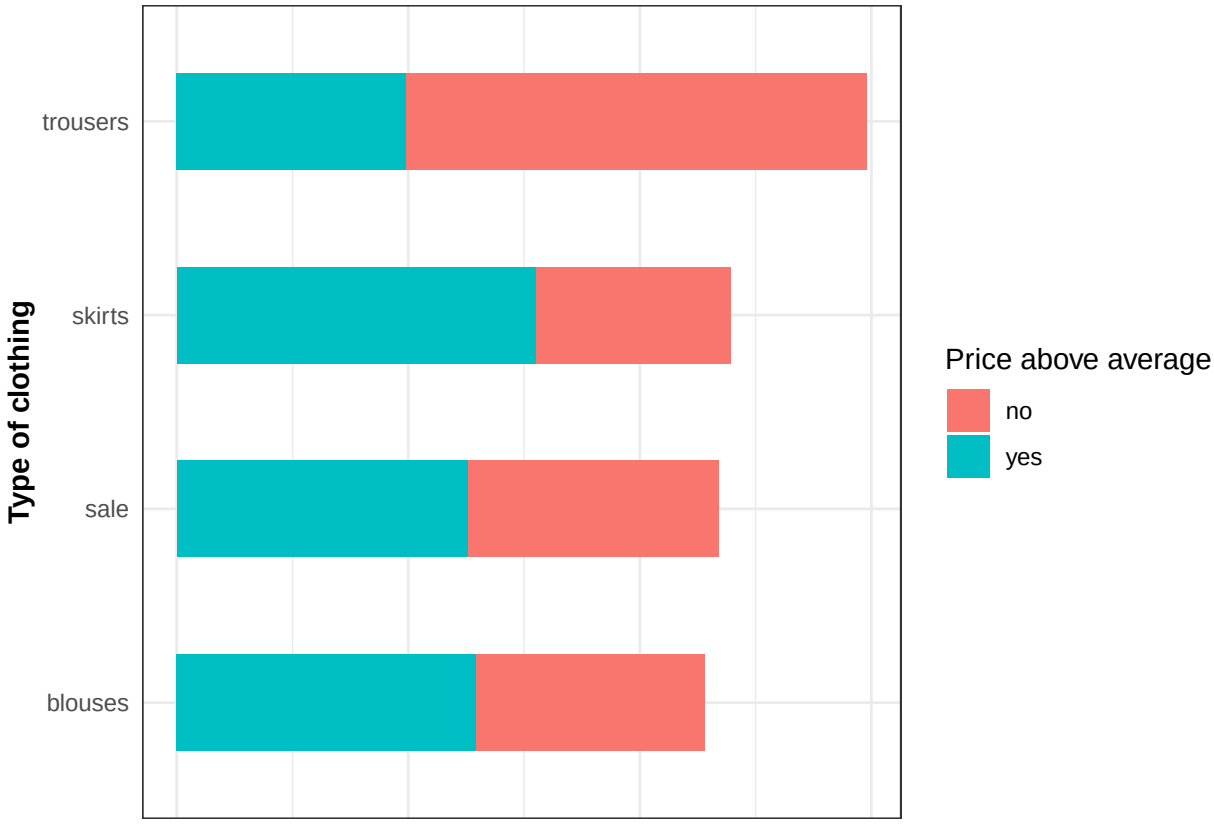



```
theme_set(theme_bw())
#67% of products in the category-trousers are sold below the average category price being a widely sold
#the individual profit margins are lower because of the demand increasing the overall profit
#skirts have the highest number of product above average price
#The surprising observation about e-commerce market manipulation is that the prices during "sale" are a
#into purchasing them.
# Data Prep
  # load data
data2<-sampled %>% group_by(page.1..main.category.,price.2) %>% summarise(total=n())
```

```
## `summarise()` has grouped output by 'page.1..main.category.'. You can override
## using the `.groups` argument.
```

```
data2<-data2%>%  group_by(page.1..main.category.) %>%
  mutate(Percentage_above_average_price = 100*total/sum(total))
  data2$Percentage_above_average_price<- format(round(data2$Percentage_above_average_price, 2), nsmall =
  data2$Percentage_above_average_price<-lapply(data2$Percentage_above_average_price, function(x)  paste


# Diverging Barcharts
```

```
ggplot(data2, mapping =aes(x=data2$page.1..main.category., y=data2$total, label=Percentage_above_averag
  geom_bar(stat='identity', mapping = aes(fill=price.2), width=.5) + xlab('Type of clothing')+ylab(elem
                                                                    axis.ticks =element_blank())+
```

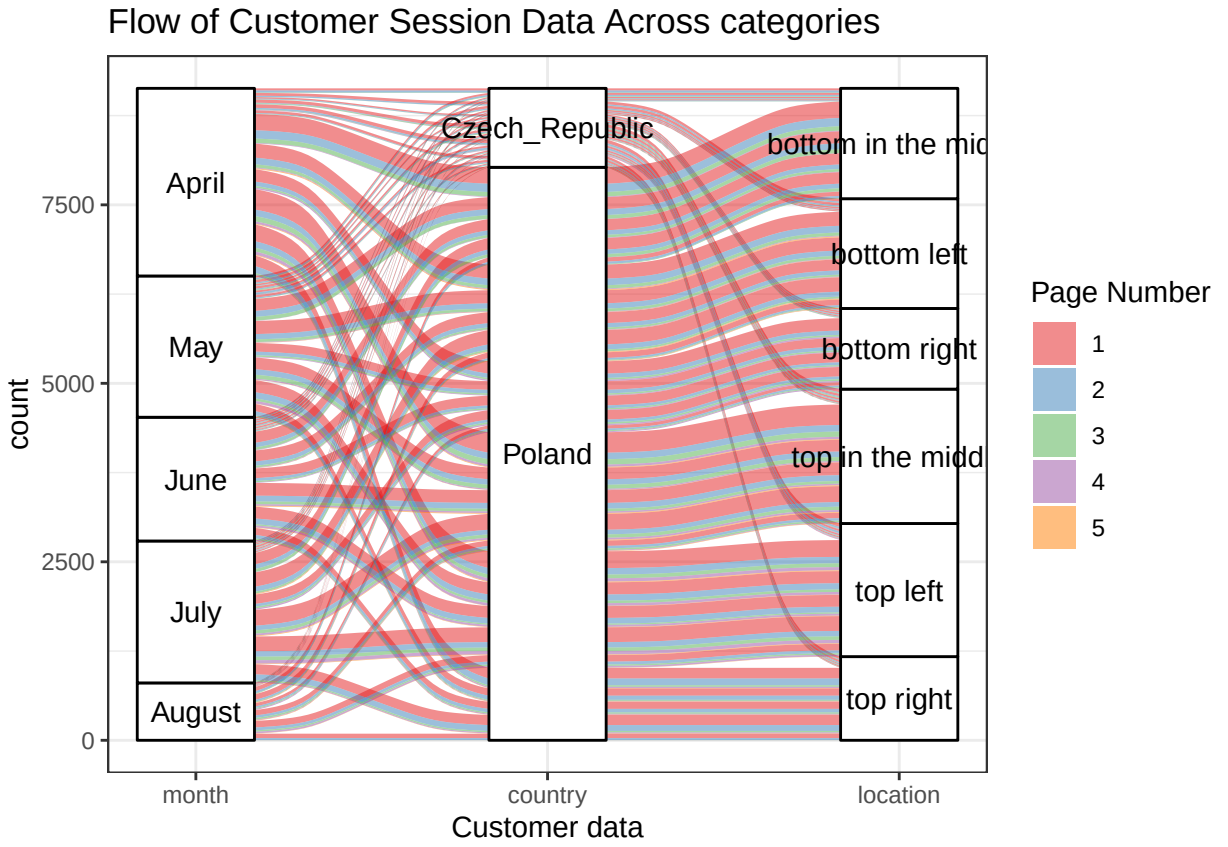



```
#Refer to the table data2 in environment section for exact percentage values for 'Above average price'
```

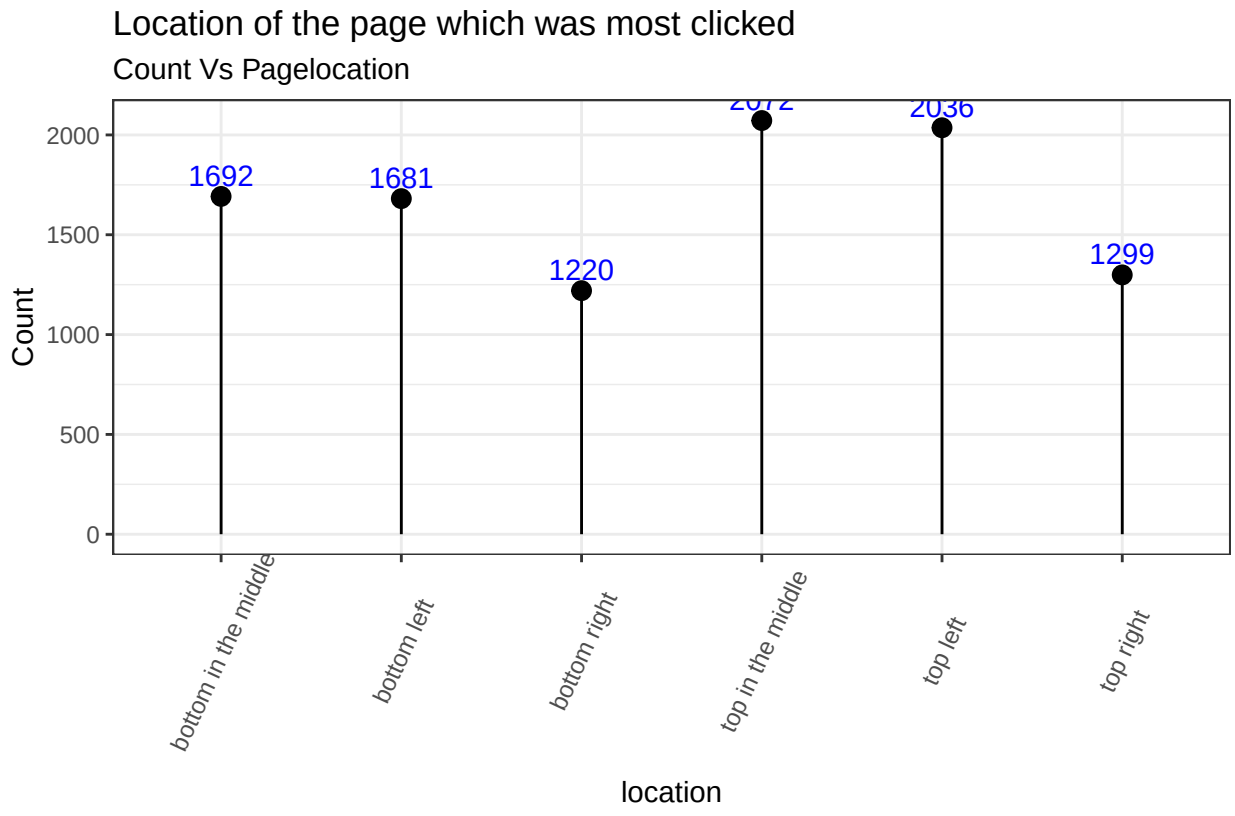
```
#Plot 3 Alluvial chart of order flow through various categories

al_df <-  sampled %>% filter(sampled$country %in% c('Poland','Czech_Republic')) %>% mutate(month =  fac
```

```
## `summarise()` has grouped output by 'month', 'country', 'location'. You can
## override using the `.groups` argument.
```

```
#This chart shows that most of the data has flown during Q2 of the year(Financial crisis reducing sales
#The highst clicks seem to be on the images top in the middle and top left of the webpage
#The image in the top right of page 2 seems to be of interest to people as it has large traffic
ggplot(data = al_df,
       aes(axis1 = month, axis2 = country, axis3 = location,
           y = count)) +
        scale_x_discrete(limits = c('month','country','location'), expand = c(.1, .05)) +
        xlab("Customer data") +
        geom_alluvium(aes(fill = as.factor(page))) +
        geom_stratum() + geom_text(stat = "stratum", aes(label = after_stat(stratum))) +   scale_fill_b
```




```
lol_chart <- sampled %>%
group_by(location) %>%
summarise(Count = n())



ggplot(lol_chart, aes(x=location, y=Count)) +
geom_point(size=3) +
geom_segment(aes(x=location,
xend=location,
y=0,
yend=Count)) +
labs(title="Location of the page which was most clicked",
subtitle="Count Vs Pagelocation",
caption="source: mpg") +
theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
geom_text(aes(label = Count), vjust = -0.5, colour = "Blue")
```

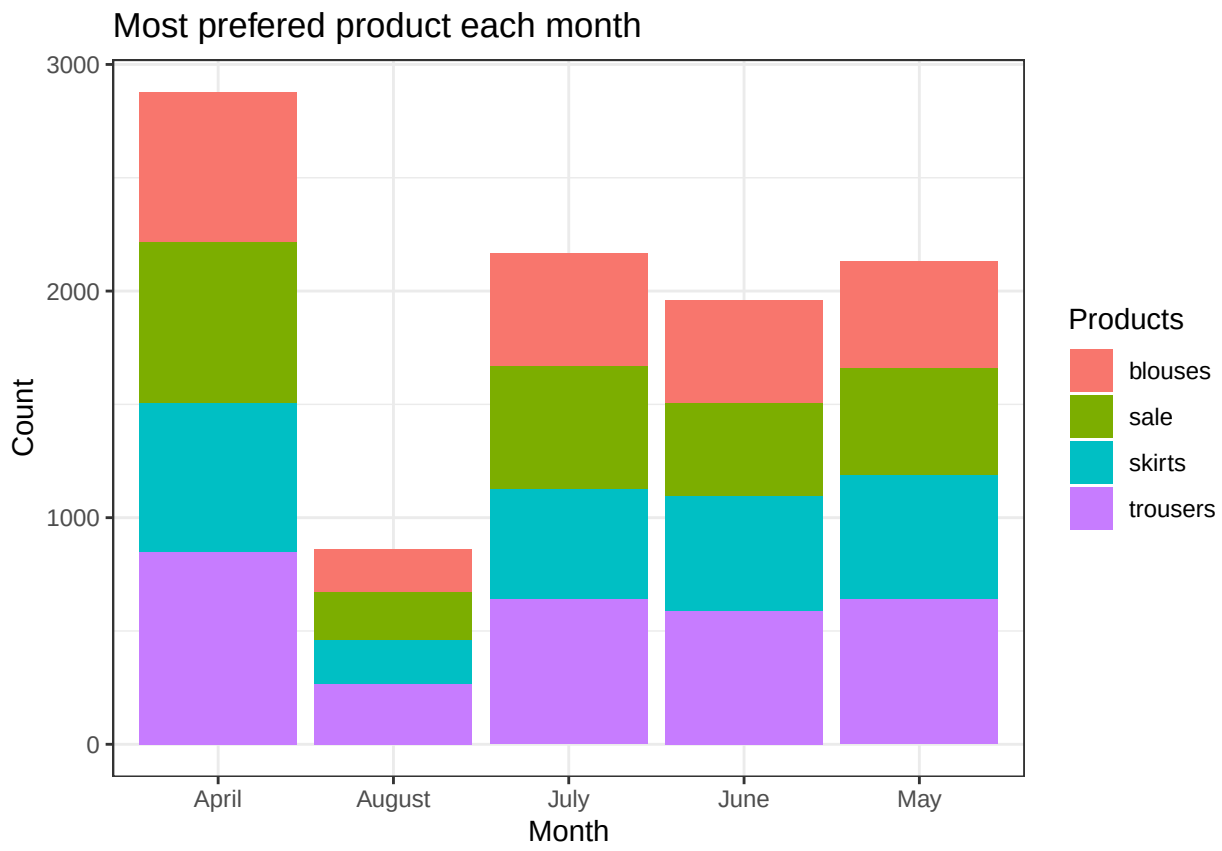


source: mpg

```
#this plot shows the part of the webpage which was most clicked



############################



stacked_bar <- sampled %>%
group_by(month, page.1..main.category.) %>%
summarise(Count = n())
```

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

```
ggplot(stacked_bar, aes(fill=page.1..main.category., y=Count, x=month)) +
geom_bar(position="stack", stat="identity") +
xlab("Month") +
ylab("Count") +
labs(fill='Products') +
ggtitle("Most prefered product each month")
```




```
#this plot depicts the most purched product of the month
```