AIT580 Final Project

Anil Kumar Mureboina

05/12/2022

```r
library(tidyverse)
library(readr)
library(ggplot2)
library(gridExtra)
library(scales)
```

```r
#import data set
udemy_dataset = read_csv("https://raw.githubusercontent.com/wyctorfogos/Text-multiclassification---PySpark/main/udemy_dataset.csv")
```

```r
# Cleaning and transforming
udemy_dataset = subset(udemy_dataset, select = -c(1,4,14))
colnames(udemy_dataset)
```

```
##  [1] "course_id"           "course_title"        "is_paid"
##  [4] "price"               "num_subscribers"     "num_reviews"
##  [7] "num_lectures"        "level"               "content_duration"
## [10] "published_timestamp" "subject"
```

```r
udemy_dataset$price[udemy_dataset$price == 'Free'] = 0
udemy_dataset$price = as.numeric(as.character(udemy_dataset$price))
na_values = udemy_dataset[rowSums(is.na(udemy_dataset)) > 0,]
udemy_dataset = subset(udemy_dataset,course_id!=na_values$course_id )
sum(is.na(udemy_dataset))
```

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

```r
#set working directory
setwd("/Users/Anil/Desktop/First_Semester/AIT580/Final project")
# Write data set to a new csv file
write.csv(udemy_dataset,"udemy_dataset.csv",row.names = FALSE)
```

```r
#Import the final clean data set that has been done by Python
udemy_dataset = read_csv("clean_udemy_dataset.csv")
udemy_dataset = subset(udemy_dataset, select = -c(12:16))
sum(is.na(udemy_dataset))
```

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

```r
summary(udemy_dataset)
```

```
##    course_id       course_title        is_paid            price
##  Min.   :   8324   Length:3677        Mode :logical   Min.   :  0.00
##  1st Qu.: 407658   Class :character   FALSE:310       1st Qu.: 20.00
##  Median : 688092   Mode  :character   TRUE :3367      Median : 45.00
##  Mean   : 675985                                      Mean   : 66.06
##  3rd Qu.: 961508                                      3rd Qu.: 95.00
##  Max.   :1282064                                      Max.   :200.00
##  num_subscribers   num_reviews       num_lectures       level
##  Min.   :     0   Min.   :    0.0   Min.   :  4.00   Length:3677
##  1st Qu.:   111   1st Qu.:    4.0   1st Qu.: 15.00   Class :character
##  Median :   912   Median :   18.0   Median : 25.00   Mode  :character
##  Mean   :  3198   Mean   :  156.3   Mean   : 40.12
##  3rd Qu.:  2547   3rd Qu.:   67.0   3rd Qu.: 46.00
##  Max.   :268923   Max.   :27445.0   Max.   :779.00
##  content_duration  published_timestamp             subject
##  Min.   : 0.1333   Min.   :2011-07-09 05:43:31   Length:3677
##  1st Qu.: 1.0000   1st Qu.:2015-03-16 18:36:17   Class :character
##  Median : 2.0000   Median :2016-01-27 18:16:28   Mode  :character
##  Mean   : 4.0956   Mean   :2015-11-26 14:09:11
##  3rd Qu.: 4.5000   3rd Qu.:2016-10-29 21:56:38
##  Max.   :78.5000   Max.   :2017-07-06 21:46:30
##     category
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :2.492
##  3rd Qu.:4.000
##  Max.   :4.000
```

```r
# Write data set to a new csv file
write.csv(udemy_dataset,"clean_udemy_dataset.csv",row.names = FALSE)
```

```r
# Average price of the courses for all the categories?
Business_Finance = subset(udemy_dataset, subject == 'Business Finance')
Graphic_Design = subset(udemy_dataset, subject == 'Graphic Design')
Musical_Instruments = subset(udemy_dataset, subject == 'Musical Instruments')
Web_Development = subset(udemy_dataset, subject == 'Web Development')

summary(Business_Finance$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   20.00   45.00   68.59   95.00  200.00
```

```r
summary(Graphic_Design$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   20.00   30.00   57.83   80.00  200.00
```

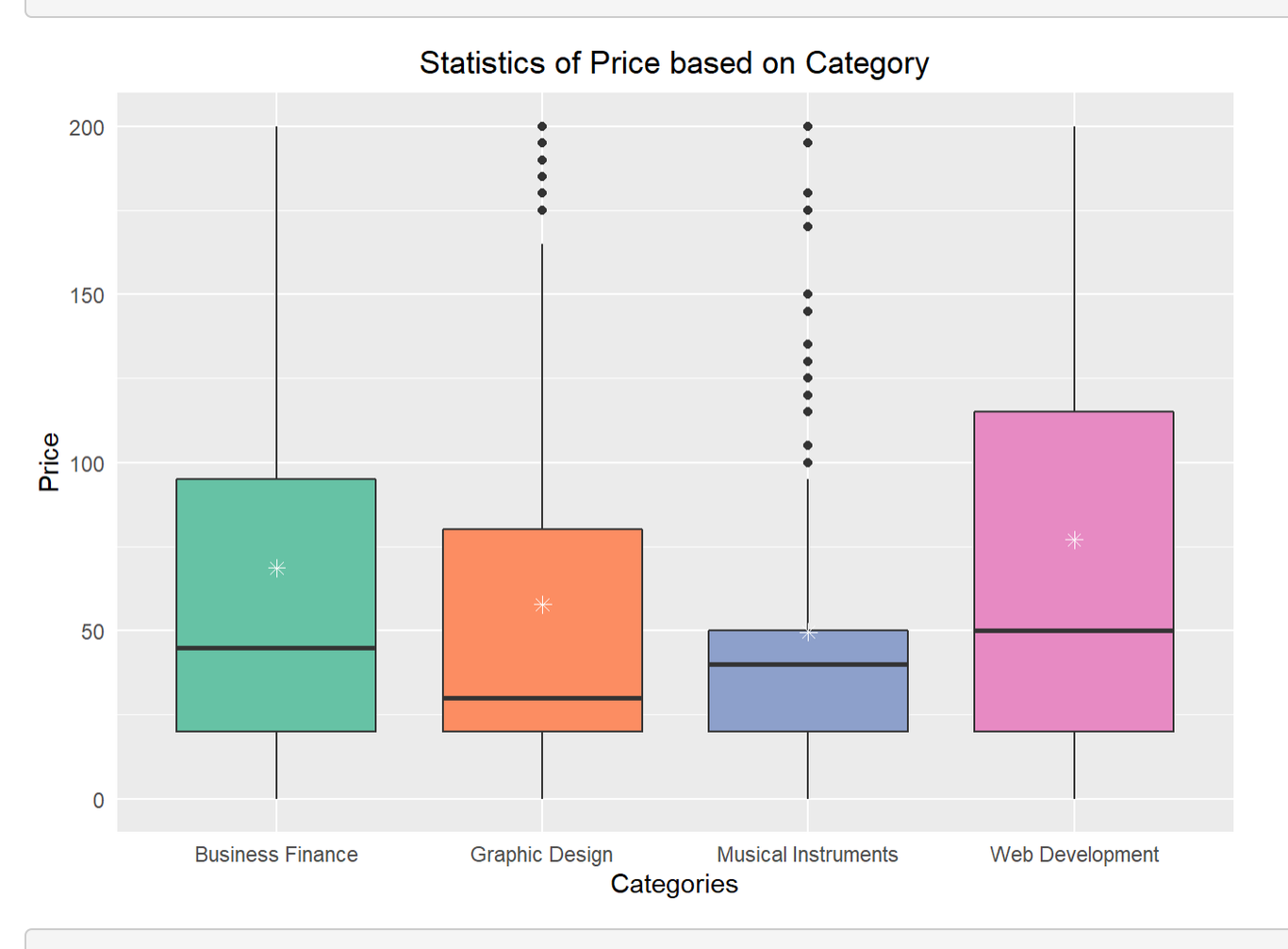
```r
summary(Musical_Instruments$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   20.00   40.00   49.56   50.00  200.00
```

```r
summary(Web_Development$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   20.00   50.00   77.03  115.00  200.00
```

```r
ggplot(udemy_dataset,aes(x = subject,y = price, fill = subject)) +
  geom_boxplot() +
  stat_summary(fun = "mean", geom = "point", shape = 8, size = 2, color = "white") +
  labs(x="Categories", y = "Price", title = "Statistics of Price based on Category") +
  scale_fill_brewer(palette = "Set2") +
  theme(legend.position="none",
        plot.title=element_text(hjust=0.5),
        axis.ticks = element_blank())
```




```r
# Average duration of the course based on the proficiency level?
all_levels = subset(udemy_dataset, level == 'All Levels')
beginner_level = subset(udemy_dataset, level == 'Beginner Level')
expert_level = subset(udemy_dataset, level == 'Expert Level')
intermediate_level = subset(udemy_dataset, level == 'Intermediate Level')

summary(all_levels$content_duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1333  1.5000  2.5000  4.8722  5.5000 76.5000
```

```r
summary(beginner_level$content_duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.450   1.000   2.000   3.091   3.500  78.500
```

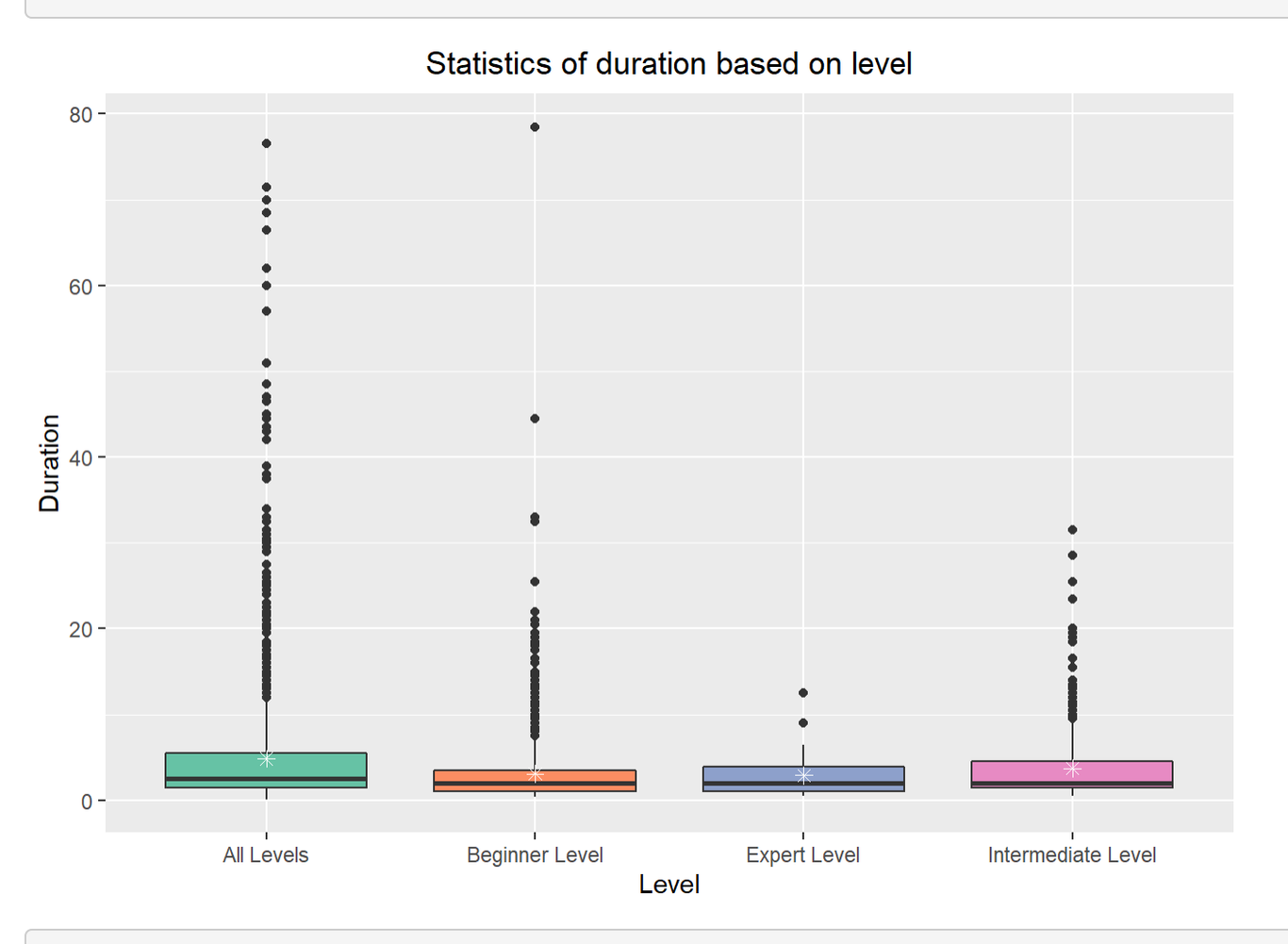
```r
summary(expert_level$content_duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5167  1.0000  2.0000  2.9055  3.8750 12.5000
```

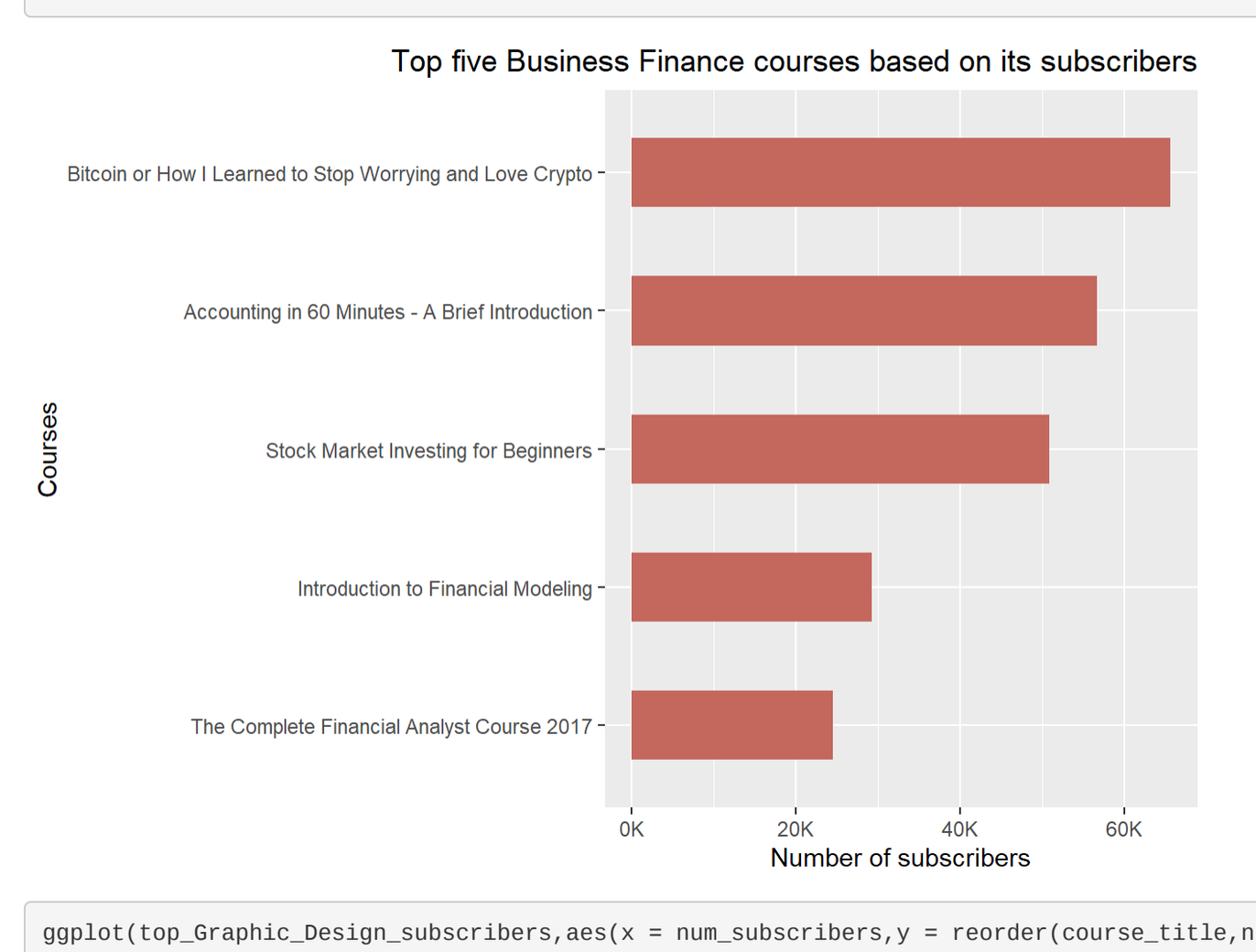
```r
summary(intermediate_level$content_duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.500   1.500   2.000   3.733   4.500  31.500
```

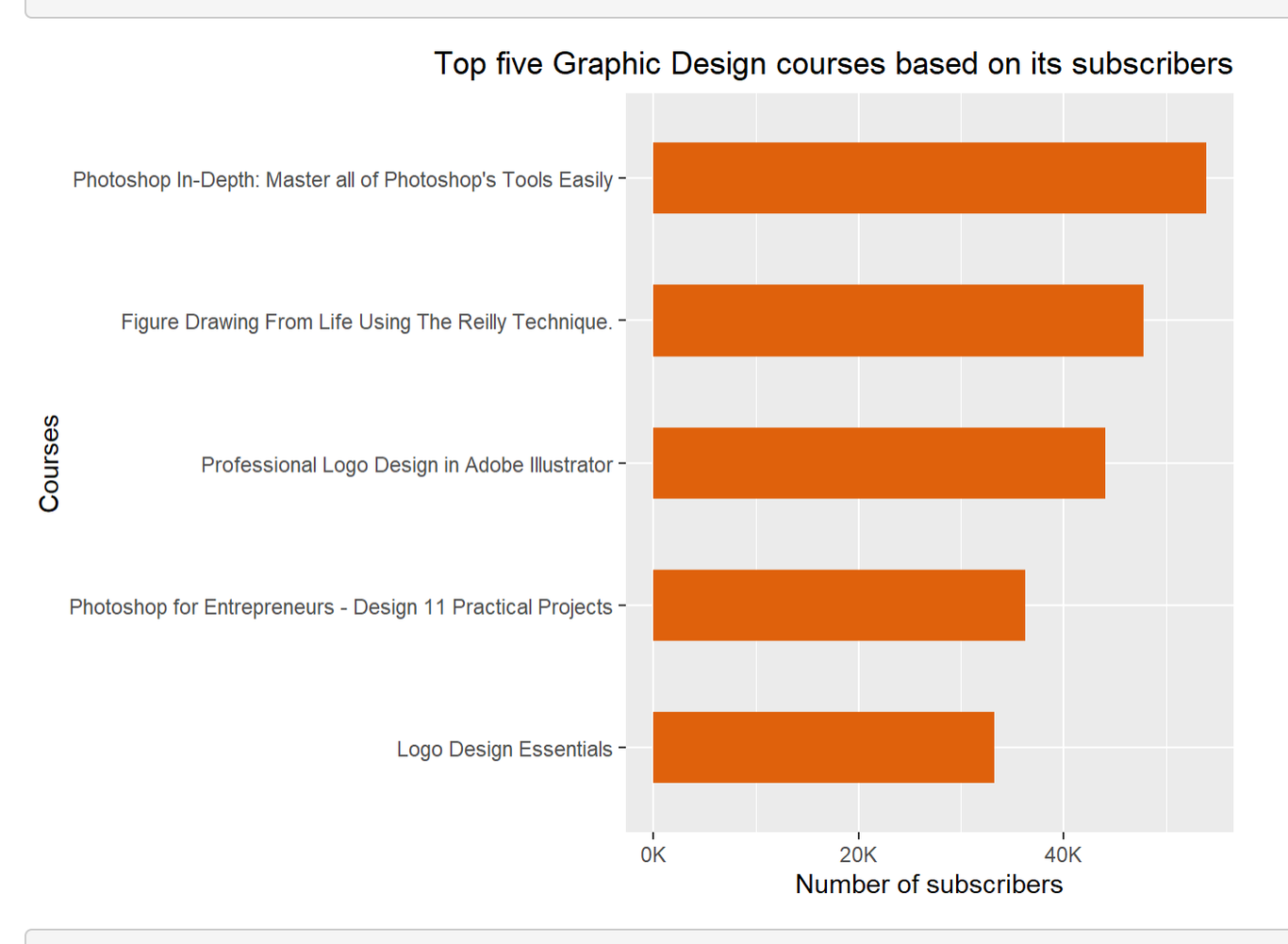
```r
ggplot(udemy_dataset,aes(x = level,y = content_duration, fill = level)) +
  geom_boxplot() +
  stat_summary(fun = "mean", geom = "point", shape = 8, size = 2, color = "white") +
  labs(x="Level", y = "Duration", title = "Statistics of duration based on level") +
  scale_fill_brewer(palette = "Set2") +
  theme(legend.position="none",
        plot.title=element_text(hjust=0.5))
```



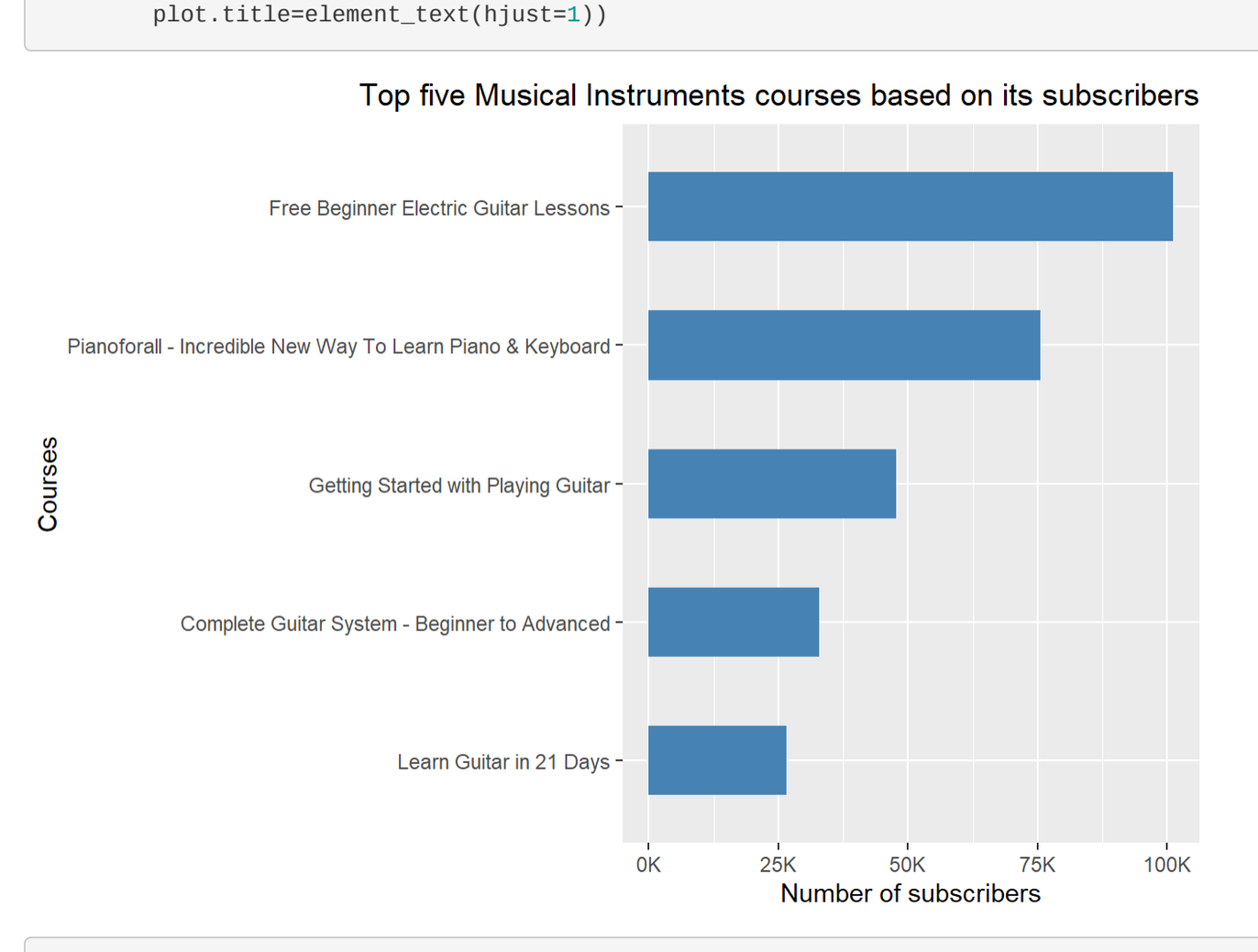

```r
# Courses that have the most subscribers in each category?
top_Business_Finance_subscribers = head(Business_Finance[order(Business_Finance$num_subscribers, decreasing = TRUE), ],5)
top_Graphic_Design_subscribers = head(Graphic_Design[order(Graphic_Design$num_subscribers, decreasing = TRUE), ],5)
top_Musical_Instruments_subscribers = head(Musical_Instruments[order(Musical_Instruments$num_subscribers, decreasing = TRUE), ],5)
top_Web_Development_subscribers = head(Web_Development[order(Web_Development$num_subscribers, decreasing = TRUE), ],5)

ggplot(top_Business_Finance_subscribers,aes(x = num_subscribers,y = reorder(course_title,num_subscribers))) +
  geom_bar(stat = "identity",width = 0.5, fill = "#C4675D") +
  labs(x="Number of subscribers", y = "Courses", title = "Top five Business Finance courses based on its subscribers") +
  scale_x_continuous(labels = label_number(suffix = "K", scale = 1e-3)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=1))
```



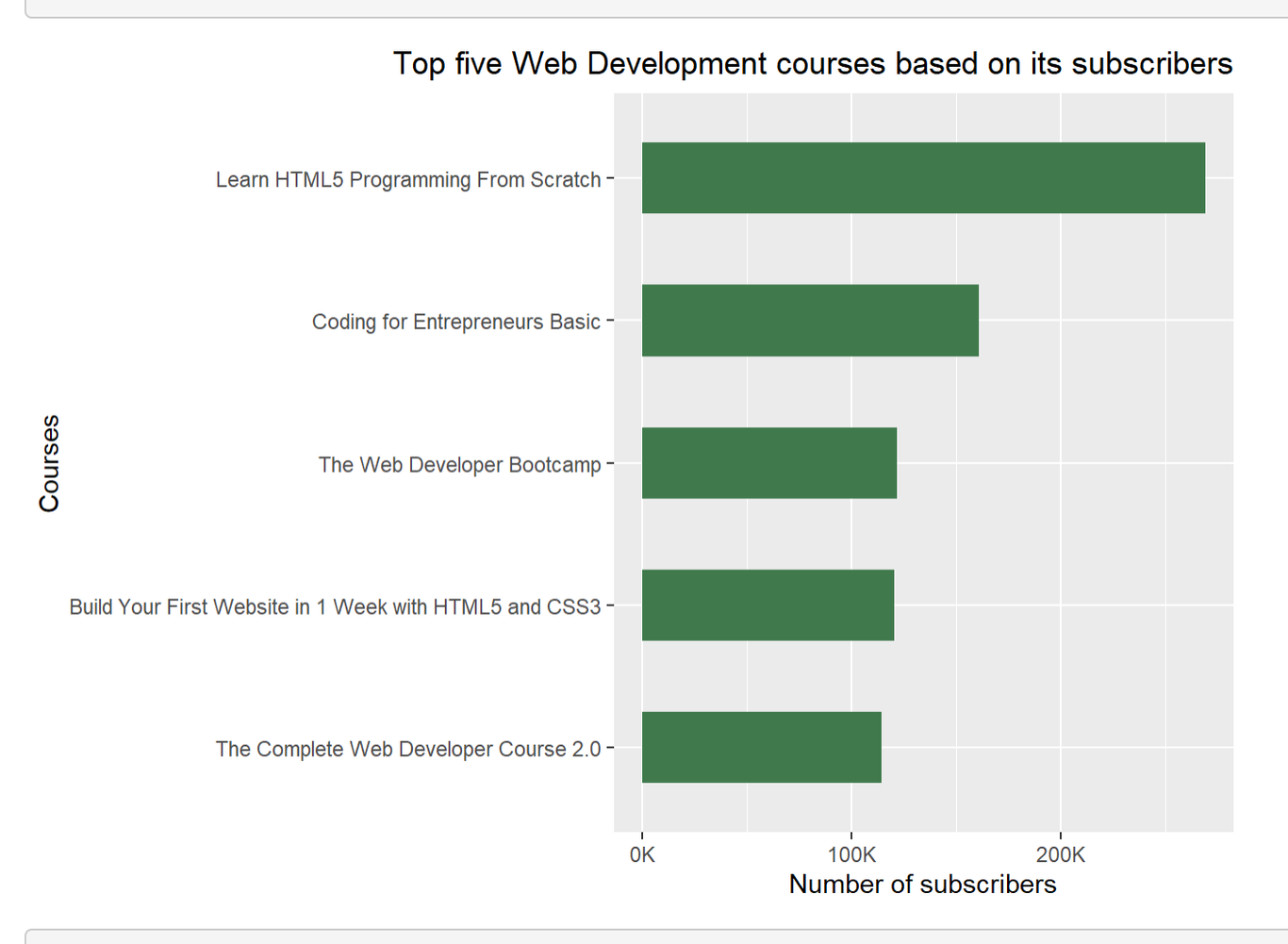

```r
ggplot(top_Graphic_Design_subscribers,aes(x = num_subscribers,y = reorder(course_title,num_subscribers))) +
  geom_bar(stat = "identity",width = 0.5, fill = "#DF610C") +
  labs(x="Number of subscribers", y = "Courses", title = "Top five Graphic Design courses based on its subscribers") +
  scale_x_continuous(labels = label_number(suffix = "K", scale = 1e-3)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=1))
```



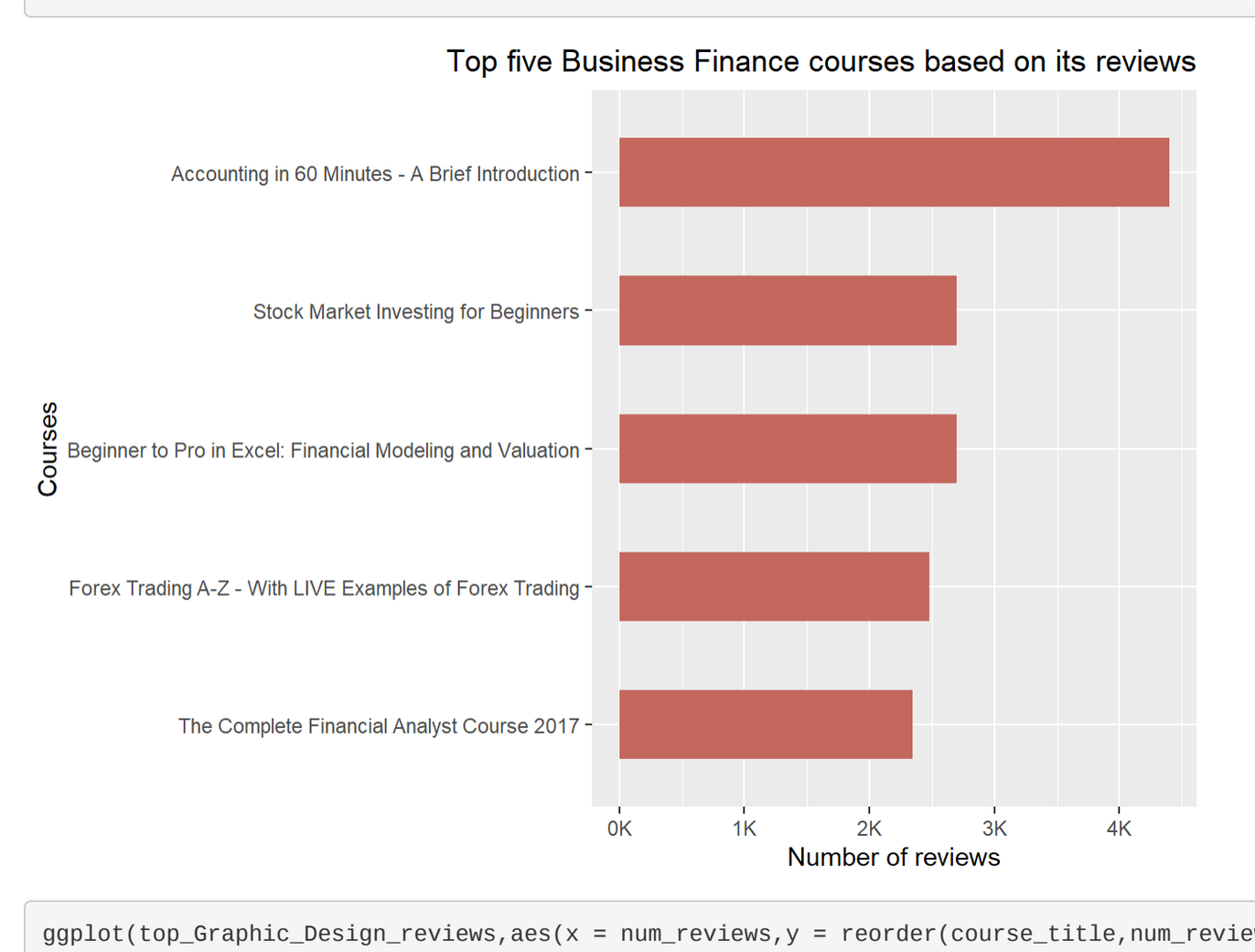

```r
ggplot(top_Musical_Instruments_subscribers,aes(x = num_subscribers,y = reorder(course_title,num_subscribers))) +
  geom_bar(stat = "identity",width = 0.5, fill = "steelblue") +
  labs(x="Number of subscribers", y = "Courses", title = "Top five Musical Instruments courses based on its subscribers") +
  scale_x_continuous(labels = label_number(suffix = "K", scale = 1e-3)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=1))
```



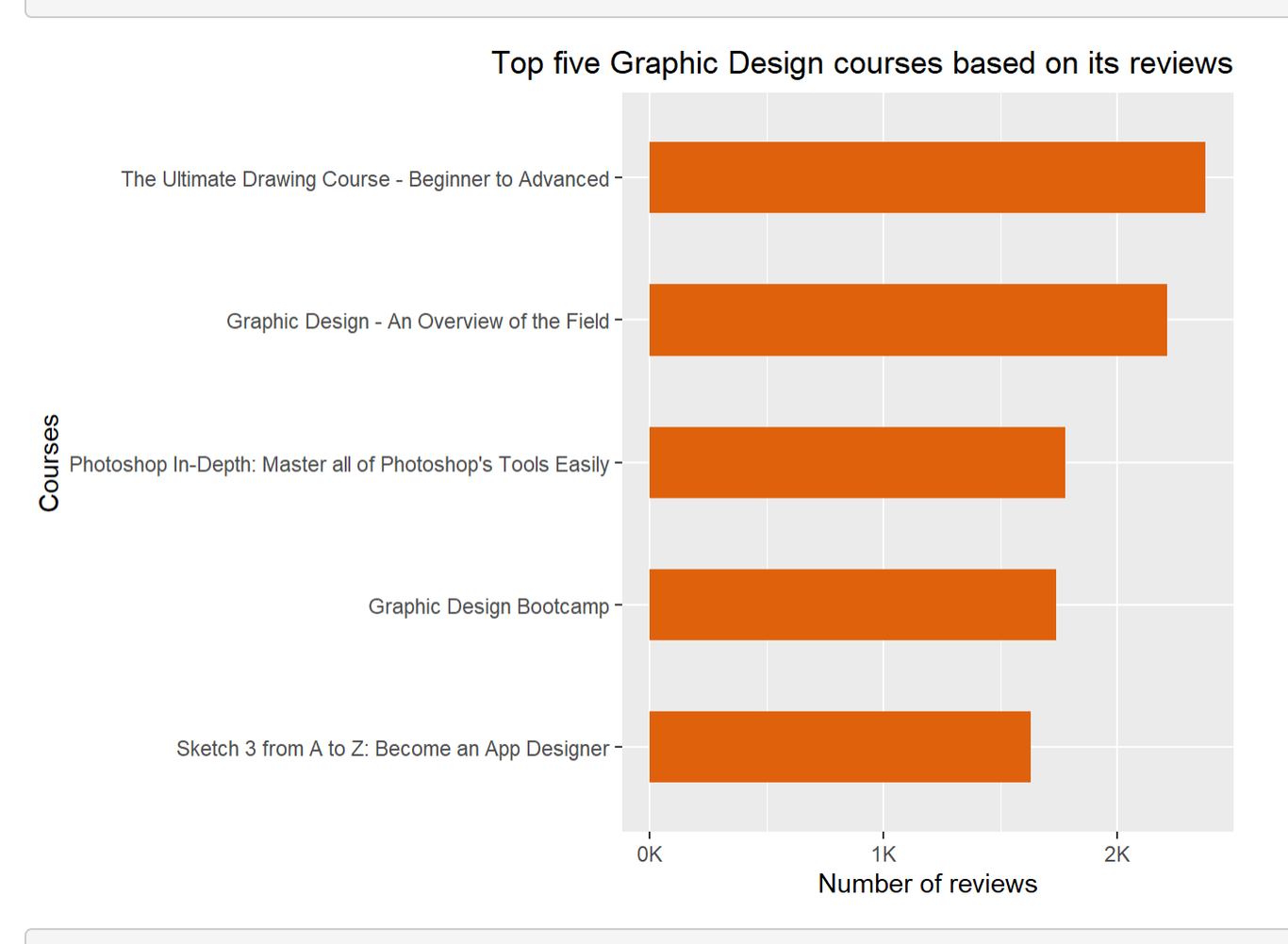

```r
ggplot(top_Web_Development_subscribers,aes(x = num_subscribers,y = reorder(course_title,num_subscribers))) +
  geom_bar(stat = "identity",width = 0.5, fill = "#407A4C") +
  labs(x="Number of subscribers", y = "Courses", title = "Top five Web Development courses based on its subscribers") +
  scale_x_continuous(labels = label_number(suffix = "K", scale = 1e-3)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=1))
```



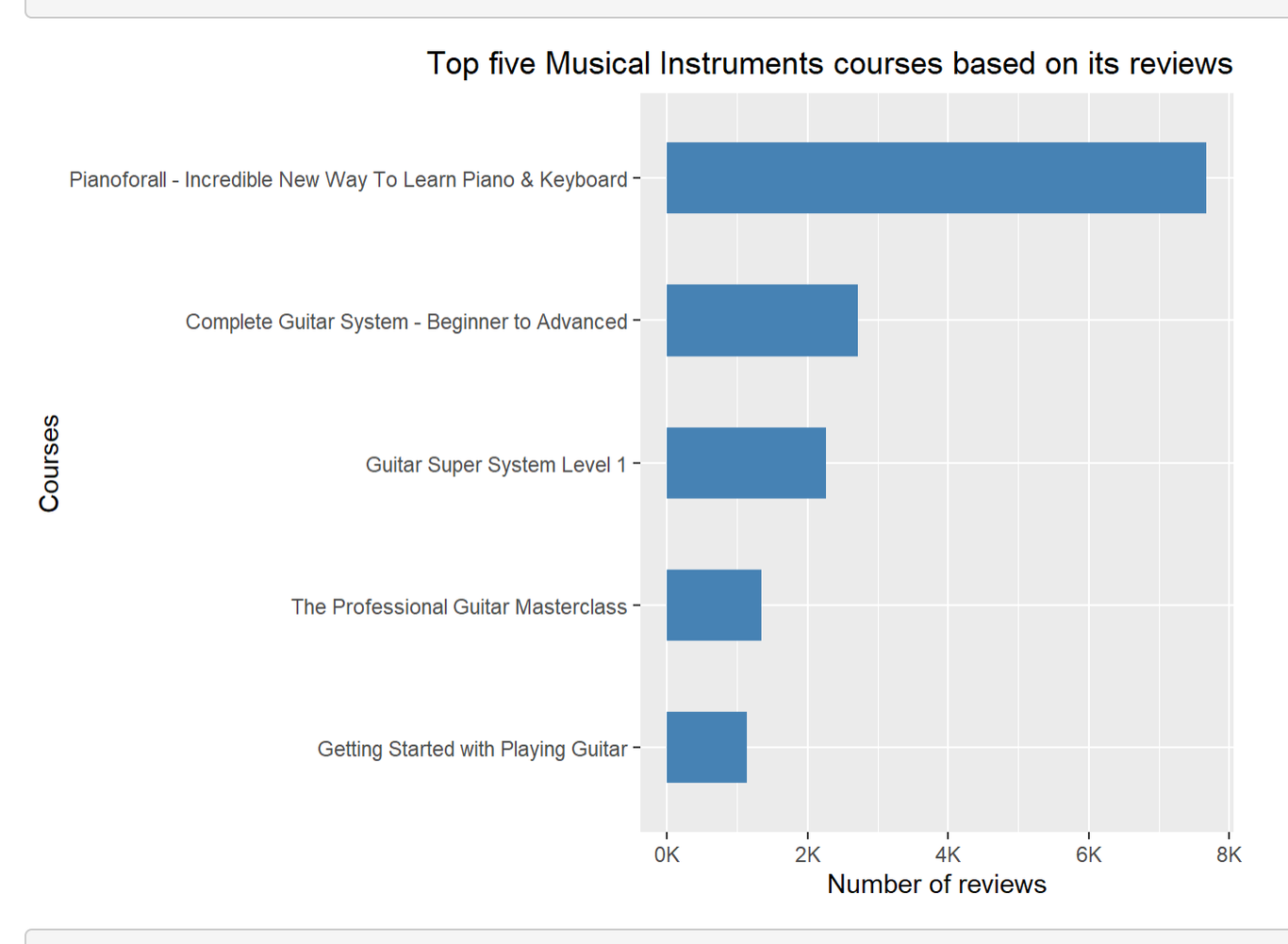

```r
# Courses that have the most reviews in each category?
top_Business_Finance_reviews = head(Business_Finance[order(Business_Finance$num_reviews, decreasing = TRUE), ],5)
top_Graphic_Design_reviews = head(Graphic_Design[order(Graphic_Design$num_reviews, decreasing = TRUE), ],5)
top_Musical_Instruments_reviews = head(Musical_Instruments[order(Musical_Instruments$num_reviews, decreasing = TRUE), ],5)
top_Web_Development_reviews = head(Web_Development[order(Web_Development$num_reviews, decreasing = TRUE), ],5)

ggplot(top_Business_Finance_reviews,aes(x = num_reviews,y = reorder(course_title,num_reviews))) +
  geom_bar(stat = "identity",width = 0.5, fill = "#C4675D") +
  labs(x="Number of reviews", y = "Courses", title = "Top five Business Finance courses based on its reviews") +
  scale_x_continuous(labels = label_number(suffix = "K", scale = 1e-3)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=1))
```



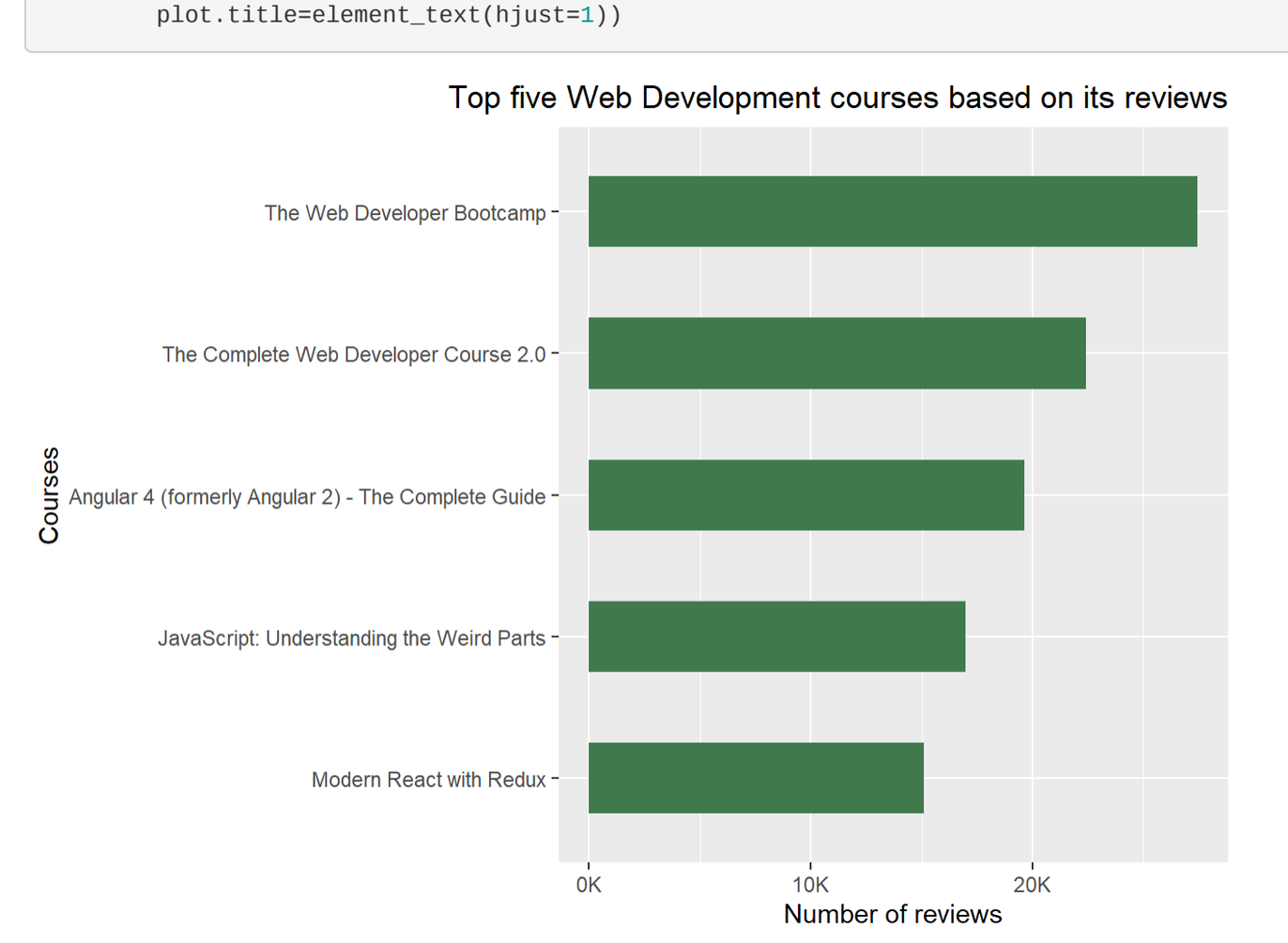

```r
ggplot(top_Graphic_Design_reviews,aes(x = num_reviews,y = reorder(course_title,num_reviews))) +
  geom_bar(stat = "identity",width = 0.5, fill = "#DF610C") +
  labs(x="Number of reviews", y = "Courses", title = "Top five Graphic Design courses based on its reviews") +
  scale_x_continuous(breaks = seq(0,3000,1000),labels = label_number(suffix = "K", scale = 1e-3)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=1))
```



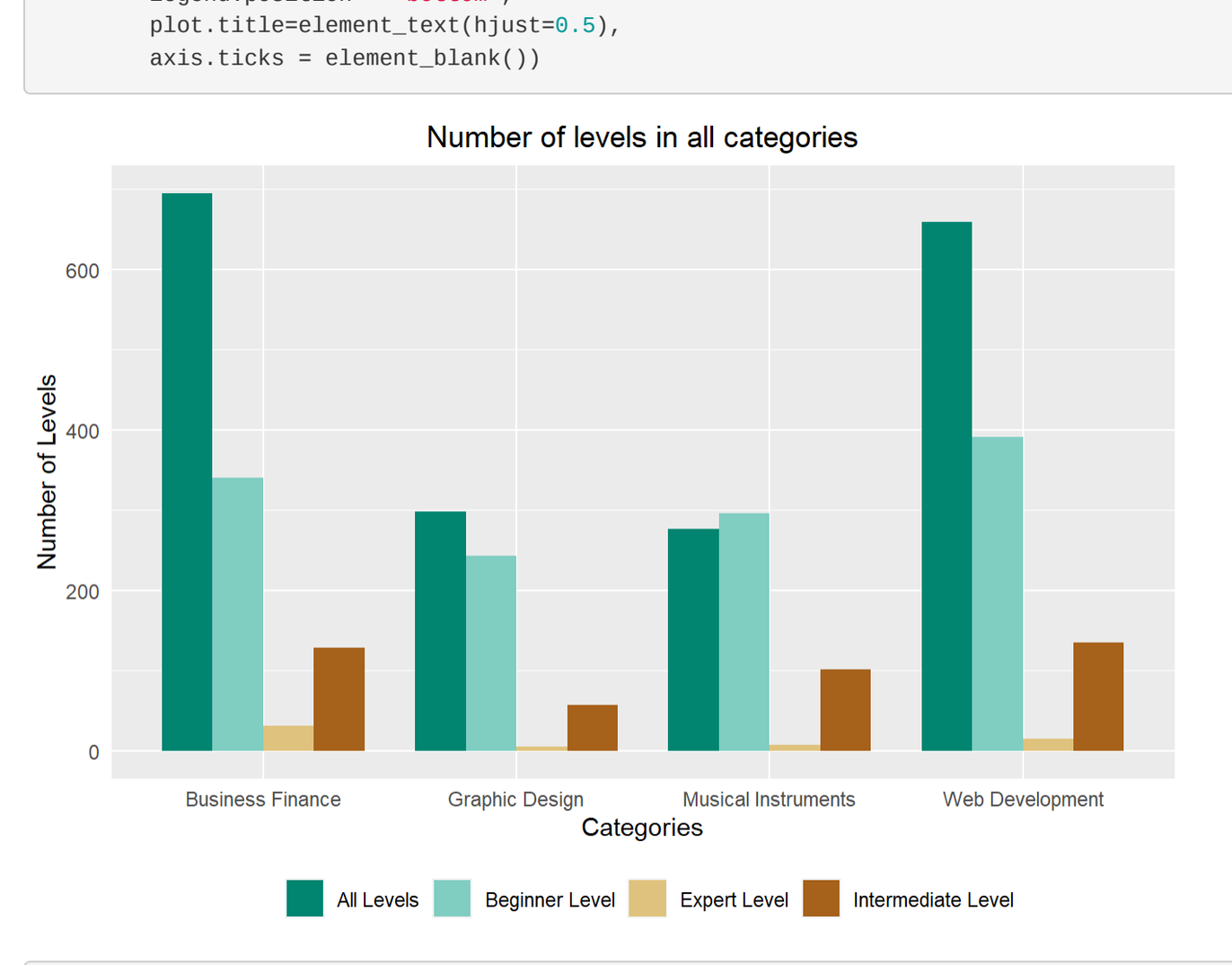

```r
ggplot(top_Musical_Instruments_reviews,aes(x = num_reviews,y = reorder(course_title,num_reviews))) +
  geom_bar(stat = "identity",width = 0.5, fill = "steelblue") +
  labs(x="Number of reviews", y = "Courses", title = "Top five Musical Instruments courses based on its reviews")
+
  scale_x_continuous(labels = label_number(suffix = "K", scale = 1e-3)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=1))
```



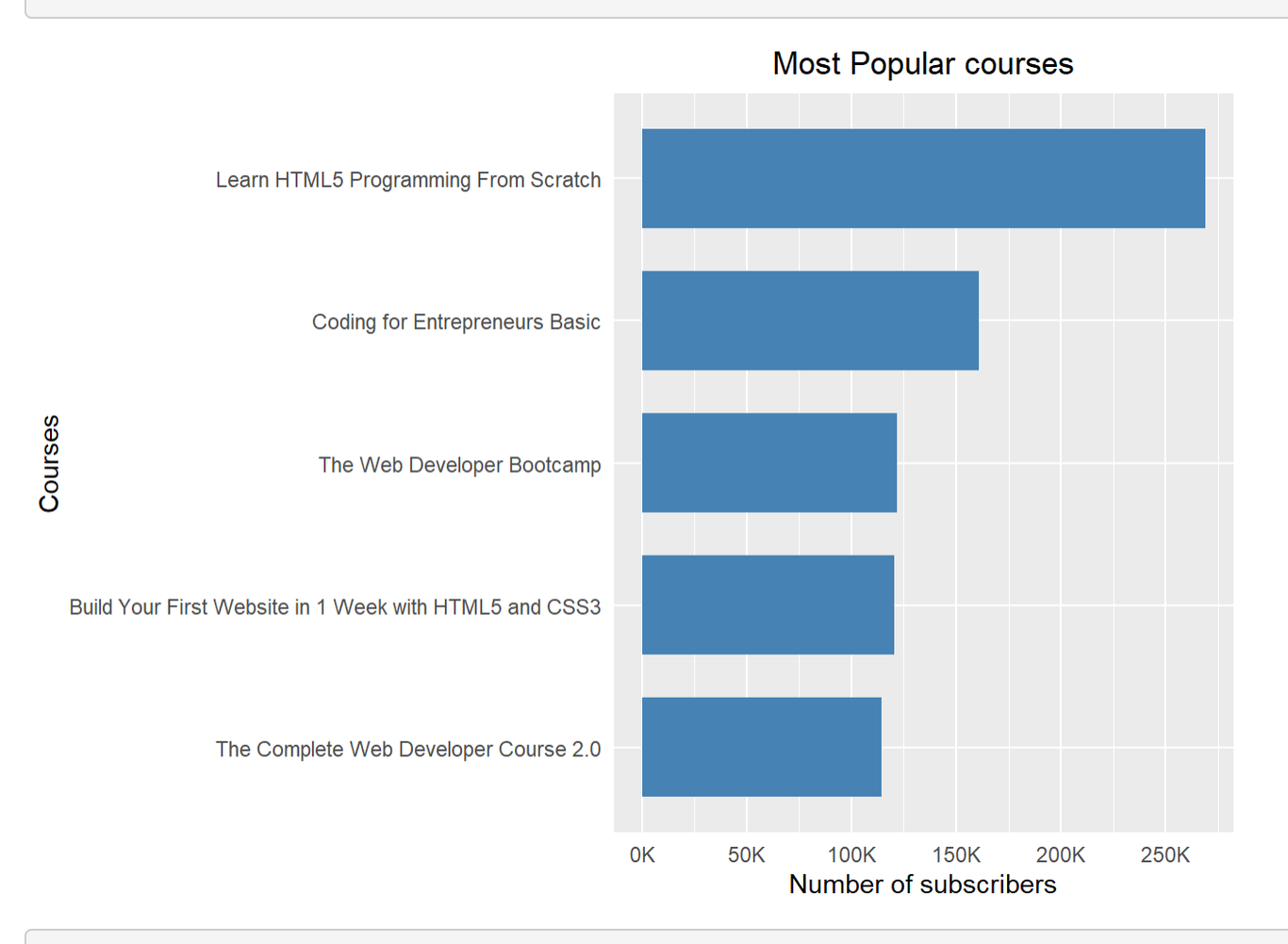

```r
ggplot(top_Web_Development_reviews,aes(x = num_reviews,y = reorder(course_title,num_reviews))) +
  geom_bar(stat = "identity",width = 0.5, fill = "#407A4C") +
  labs(x="Number of reviews", y = "Courses", title = "Top five Web Development courses based on its reviews") +
  scale_x_continuous(labels = label_number(suffix = "K", scale = 1e-3)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=1))
```



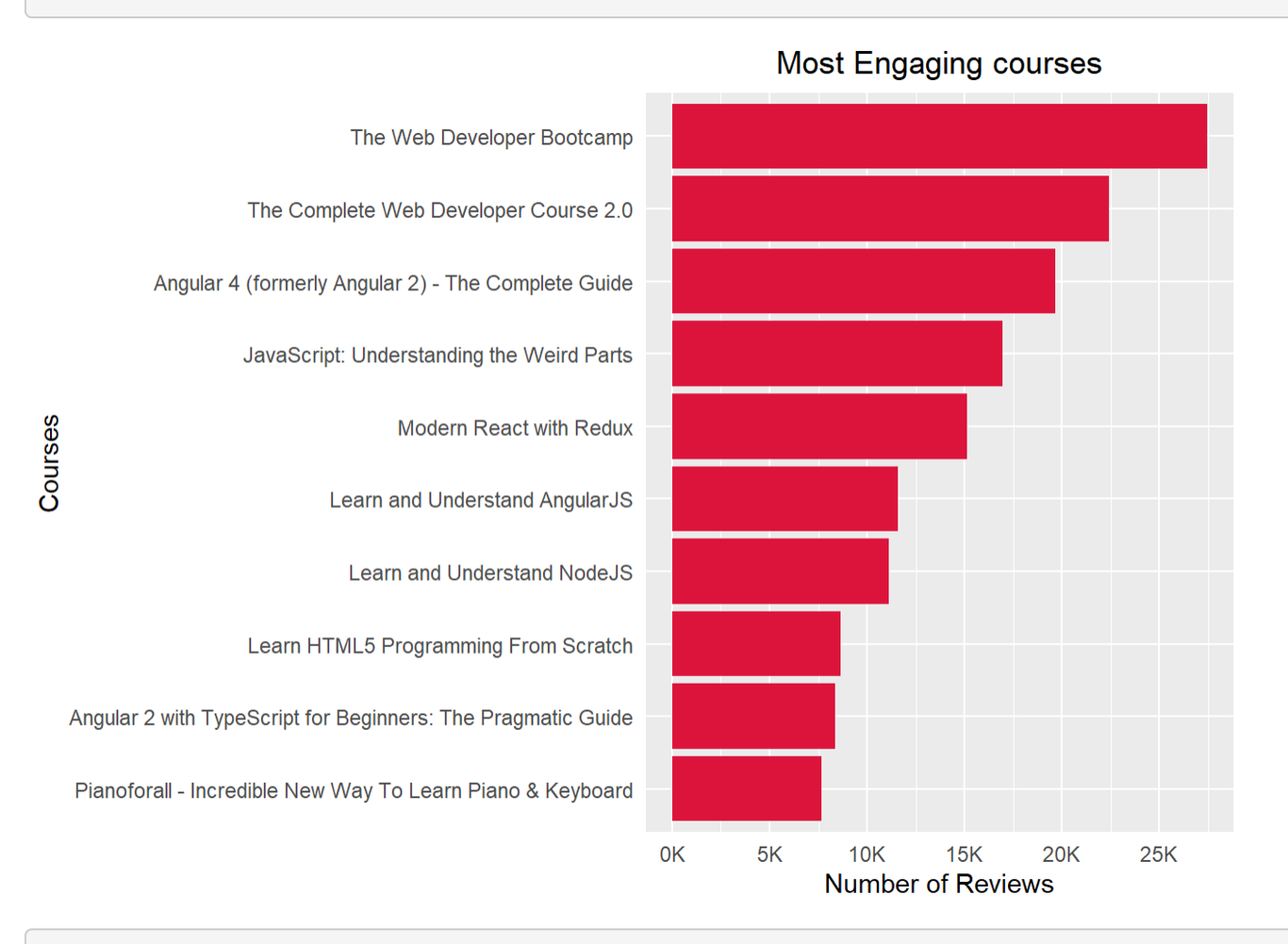

```r
# Level of proficiency count for all the courses in each category
ggplot(udemy_dataset,aes(x = subject, fill = level)) +
  geom_bar(position = position_dodge(), width = 0.8) +
  labs(x="Categories", y = "Number of Levels", title = "Number of levels in all categories") +
  scale_fill_brewer(palette = "BrBG", direction = -1) +
  theme(legend.title = element_blank(),
        legend.position = "bottom",
        plot.title = element_text(hjust=0.5),
        axis.ticks = element_blank())
```



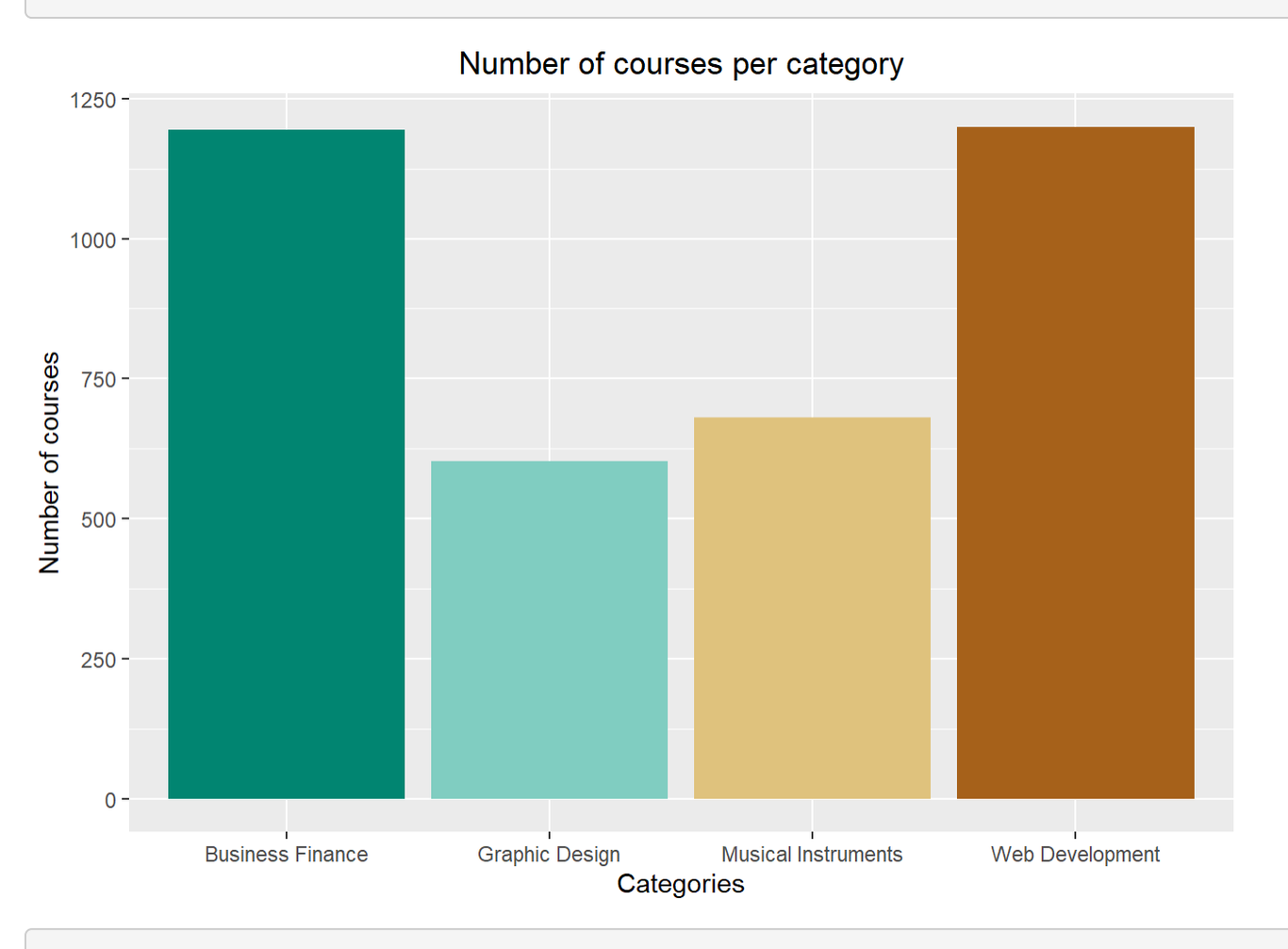

```r
# Most popular courses and when they are published
popular_courses = head(udemy_dataset[order(udemy_dataset$num_subscribers, decreasing = TRUE), ],5)

ggplot(popular_courses,aes(x = num_subscribers,y = reorder(course_title,num_subscribers), fill = course_title)) +
  geom_bar(stat = "identity", fill = "steelblue", width = 0.7) +
  labs(x="Number of subscribers", y = "Courses", title = "Most Popular courses") +
  scale_x_continuous(breaks = seq(0,300000,50000),labels = label_number(suffix = "K", scale = 1e-3)) +
   theme(legend.position="none",
        plot.title=element_text(hjust=0.5),
        axis.ticks=element_blank())
```



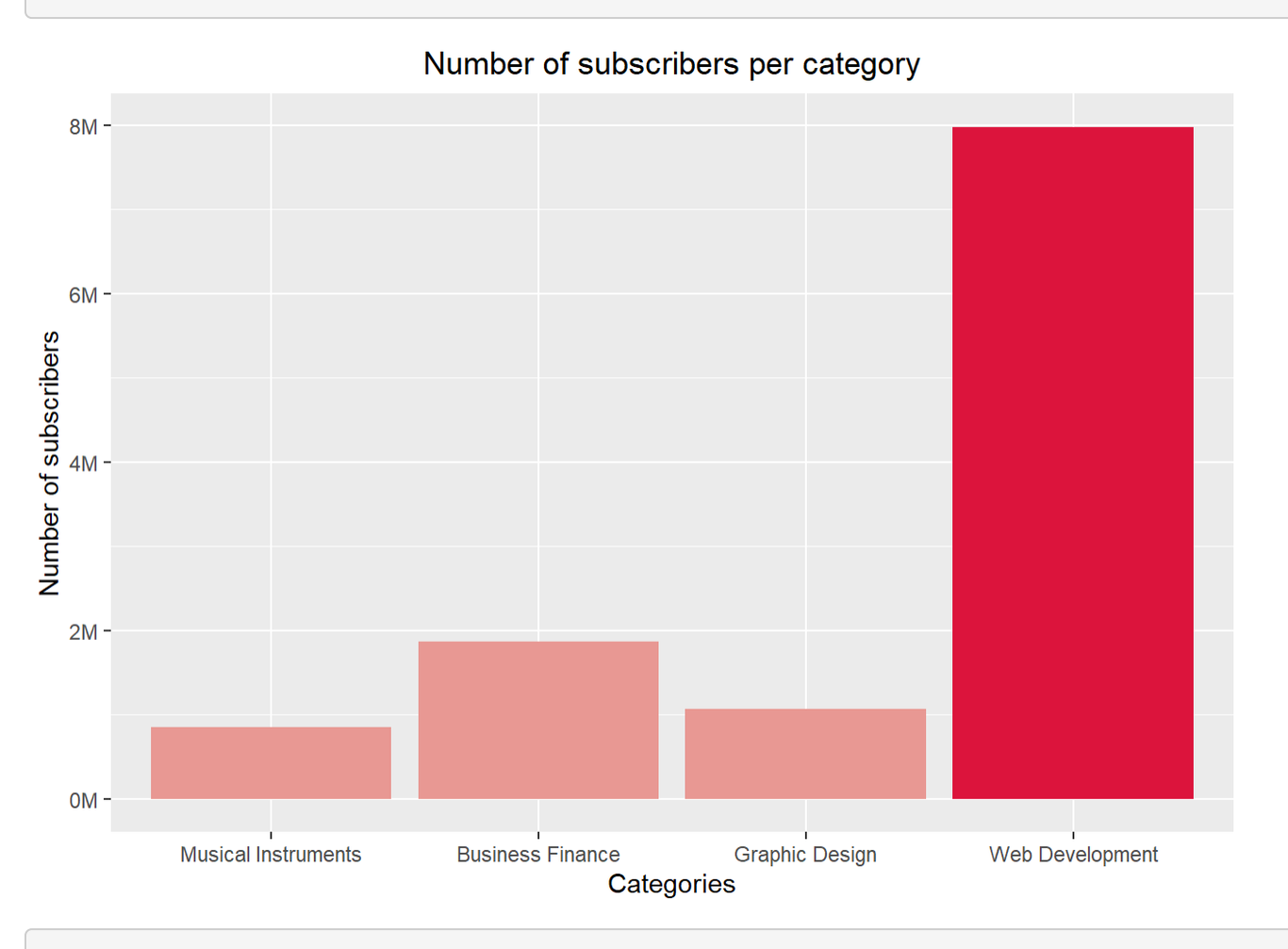

```r
#Most engaging courses
engaging_courses = head(udemy_dataset[order(udemy_dataset$num_reviews, decreasing = TRUE), ],10)

ggplot(engaging_courses,aes(x = num_reviews,y = reorder(course_title,num_reviews), fill = course_title)) +
  geom_bar(stat = "identity", fill = "#DC143C") +
  labs(x="Number of Reviews", y = "Courses", title = "Most Engaging courses") +
  scale_x_continuous(breaks = seq(0,30000,5000),labels = label_number(suffix = "K", scale = 1e-3)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=0.5),
        axis.ticks = element_blank())
```




```r
# Number of courses per category
ggplot(udemy_dataset,aes(x = subject,y = frequency(subject), fill = subject)) +
  geom_bar(stat = "identity", width = 0.9) +
  scale_fill_brewer(direction = -1, palette = "BrBG") +
  labs(x="Categories", y = "Number of courses", title = "Number of courses per category") +
  theme(legend.position="none",
        plot.title=element_text(hjust=0.5))
```




```r
# Number of subscribers per category
ggplot(udemy_dataset,aes(x = subject,y = num_subscribers)) +
  geom_bar(stat = "identity",aes(x = reorder(subject,num_subscribers),y = num_subscribers, fill = subject)) +
  scale_fill_manual(values = c("#E89893","#E89893","#E89893","#DC143C")) +
  labs(x="Categories", y = "Number of subscribers", title = "Number of subscribers per category") +
  scale_y_continuous(breaks = seq(0,8000000,2000000),labels = label_number(suffix = "M", scale = 1e-6)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=0.5))
```




```r
# Number of subscribers per level
subscribers_per_level = ggplot(udemy_dataset,aes(x = reorder(level,-num_subscribers),y = num_subscribers, fill =
 level)) +
  geom_bar(stat = "identity") +
  scale_fill_brewer(palette = "RdBu") +
  labs(x="Level", y = "Number of subscribers") +
  scale_y_continuous(breaks = seq(0,7000000,2000000),labels = label_number(suffix = "M", scale = 1e-6)) +
  theme(legend.position="none",
        plot.title=element_text(hjust=0.5),
        axis.ticks=element_blank())

# Number of courses per level
coursess_per_level = ggplot(udemy_dataset,aes(x = reorder(level,-num_subscribers),y = frequency(course_id), fill
 = level)) +
  geom_bar(stat = "identity") +
  scale_fill_brewer(palette = "RdBu") +
  labs(x="", y = "Number of courses") +
  theme(legend.position="none",
        plot.title=element_text(hjust=0.5),
        axis.text.x = element_blank(),
        axis.ticks=element_blank())

grid.arrange(coursess_per_level,subscribers_per_level, top="Number of courses and their subscribers per level")
```