# Finding the best market to advertise in - Coding Content

Bena

2024-01-19

# Background

Let's assume that we're working for an an e-learning company that offers courses on programming. Most of our courses are on web and mobile development, but we also cover many other domains, like data science, game development, etc. We want to promote our product and we'd like to invest some money in advertisement. Our goal in this project is to find out the two best markets to advertise our product in.

The objectives include: -summarizing distributions using the mean, the median, and the mode. -using measures of variability of a distribution using the range, the mean absolute deviation, the variance, and the standard deviation. -locating any value(s) in a distribution using z-scores.

# Understanding the data

We could organize surveys for a couple of different markets but since that's too costly, we'll first explore the use of reliable existing data. One good candidate is the data from freeCodeCamp's New Coder Survey. (https://medium.freecodecamp.org/we-asked-20-000-people-who-they-are-and-how-theyre-learning-to-code-fff5d668969 (https://medium.freecodecamp.org/we-asked-20-000-people-who-they-are-and-how-theyre-learning-to-code-fff5d668969)) https://www.freecodecamp.org/ (https://www.freecodecamp.org/) is a free e-learning platform that offers courses on web development. They run a popular Medium publication (over 400,000 followers), their survey attracted new coders with varying interests (not only web development), which is ideal for the purpose of our analysis. The 2017 survey data is publicly available in this GitHub repository(https://github.com/freeCodeCamp/2017-new-coder-survey/tree/master/clean-data (https://github.com/freeCodeCamp/2017-new-coder-survey/tree/master/clean-data)).

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.4.0     ✓ purrr   1.0.1
## ✓ tibble  3.1.8     ✓ dplyr   1.1.0
## ✓ tidyr   1.3.0     ✓ stringr 1.5.0
## ✓ readr   2.1.3     ✓ forcats 1.0.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
survey_data <- read_csv("2017-fCC-New-Coders-Survey-Data.csv")
```

```
## Rows: 18175 Columns: 136
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr   (27): BootcampName, CityPopulation, CodeEventOther, CommuteTime, Count...
## dbl  (105): Age, AttendedBootcamp, BootcampFinish, BootcampLoanYesNo, Bootca...
## dttm   (4): Part1EndTime, Part1StartTime, Part2EndTime, Part2StartTime
##
## 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.
```

```
head(survey_data)
```

```
## # A tibble: 6 × 136
##     Age Attend…¹ Bootc…² Bootc…³ Bootc…⁴ Bootc…⁵ Child…⁶ CityP…⁷ CodeE…⁸ CodeE…⁹
##   <dbl>    <dbl>   <dbl>   <dbl> <chr>     <dbl>   <dbl> <chr>     <dbl>   <dbl>
## 1    27        0      NA      NA <NA>         NA      NA more t…      NA      NA
## 2    34        0      NA      NA <NA>         NA      NA less t…      NA      NA
## 3    21        0      NA      NA <NA>         NA      NA more t…      NA      NA
## 4    26        0      NA      NA <NA>         NA      NA betwee…      NA      NA
## 5    20        0      NA      NA <NA>         NA      NA betwee…      NA      NA
## 6    28        0      NA      NA <NA>         NA      NA less t…      NA      NA
## # … with 126 more variables: CodeEventFCC <dbl>, CodeEventGameJam <dbl>,
## #   CodeEventGirlDev <dbl>, CodeEventHackathons <dbl>, CodeEventMeetup <dbl>,
## #   CodeEventNodeSchool <dbl>, CodeEventNone <dbl>, CodeEventOther <chr>,
## #   CodeEventRailsBridge <dbl>, CodeEventRailsGirls <dbl>,
## #   CodeEventStartUpWknd <dbl>, CodeEventWkdBootcamps <dbl>,
## #   CodeEventWomenCode <dbl>, CodeEventWorkshops <dbl>, CommuteTime <chr>,
## #   CountryCitizen <chr>, CountryLive <chr>, EmploymentField <chr>, …
```

```
glimpse(survey_data)
```

```
## Rows: 18,175
## Columns: 136
## $ Age                          <dbl> 27, 34, 21, 26, 20, 28, 29, 29, 23, 24, …
## $ AttendedBootcamp             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ BootcampFinish               <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ BootcampLoanYesNo            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ BootcampName                 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ BootcampRecommend            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ ChildrenNumber               <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CityPopulation               <chr> "more than 1 million", "less than 100,00…
## $ CodeEventConferences         <dbl> NA, NA, NA, NA, NA, NA, 1, NA, NA, 1, NA…
## $ CodeEventDjangoGirls         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CodeEventFCC                 <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CodeEventGameJam             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CodeEventGirlDev             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CodeEventHackathons          <dbl> NA, NA, 1, NA, NA, NA, NA, NA, NA, 1, NA…
## $ CodeEventMeetup              <dbl> NA, NA, NA, NA, NA, NA, 1, NA, NA, NA, N…
## $ CodeEventNodeSchool          <dbl> NA, NA, 1, NA, NA, NA, NA, NA, NA, NA, N…
## $ CodeEventNone                <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CodeEventOther               <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CodeEventRailsBridge         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CodeEventRailsGirls          <dbl> NA, NA, NA, NA, NA, NA, 1, NA, NA, NA, N…
## $ CodeEventStartUpWknd         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CodeEventWkdBootcamps        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CodeEventWomenCode           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ CodeEventWorkshops           <dbl> NA, NA, NA, NA, NA, NA, 1, NA, NA, NA, N…
## $ CommuteTime                  <chr> "15 to 29 minutes", NA, "15 to 29 minute…
## $ CountryCitizen               <chr> "Canada", "United States of America", "U…
## $ CountryLive                  <chr> "Canada", "United States of America", "U…
## $ EmploymentField              <chr> "software development and IT", NA, "soft…
## $ EmploymentFieldOther         <chr> NA, NA, NA, NA, NA, NA, "Market research…
## $ EmploymentStatus             <chr> "Employed for wages", "Not working but l…
## $ EmploymentStatusOther        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ ExpectedEarning              <dbl> NA, 35000, 70000, 40000, 140000, NA, 300…
## $ FinanciallySupporting        <dbl> NA, NA, NA, 0, NA, NA, NA, NA, NA, NA, N…
## $ FirstDevJob                  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 1, NA, N…
## $ Gender                       <chr> "female", "male", "male", "male", "femal…
## $ GenderOther                  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ HasChildren                  <dbl> NA, NA, NA, 0, NA, NA, NA, NA, NA, NA, N…
## $ HasDebt                      <dbl> 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, N…
## $ HasFinancialDependents       <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, N…
## $ HasHighSpdInternet           <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1…
## $ HasHomeMortgage              <dbl> 0, 0, NA, 1, NA, NA, 1, NA, NA, NA, 0, N…
## $ HasServedInMilitary          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ HasStudentDebt               <dbl> 0, 1, NA, 0, NA, NA, 1, NA, NA, NA, 1, N…
## $ HomeMortgageOwe              <dbl> NA, NA, NA, 40000, NA, NA, 120000, NA, N…
## $ HoursLearning                <dbl> 15, 10, 25, 14, 10, 12, 16, 15, 5, 2, 15…
## $ ID.x                         <chr> "02d9465b21e8bd09374b0066fb2d5614", "5bf…
## $ ID.y                         <chr> "eb78c1c3ac6cd9052aec557065070fbf", "21d…
## $ Income                       <dbl> NA, NA, 13000, 24000, NA, NA, 40000, NA,…
## $ IsEthnicMinority             <dbl> NA, 0, 1, 0, 0, 0, NA, 1, 0, 0, 0, 1, 0,…
```

```
## $ IsReceiveDisabilitiesBenefits <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0…
## $ IsSoftwareDev                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0…
## $ IsUnderEmployed               <dbl> 0, NA, 0, 1, NA, NA, 0, NA, 0, NA, 1, NA…
## $ JobApplyWhen                  <chr> NA, "Within 7 to 12 months", "Within 7 t…
## $ JobInterestBackEnd            <dbl> NA, NA, 1, 1, 1, NA, NA, NA, NA, 1, NA, …
## $ JobInterestDataEngr           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ JobInterestDataSci            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ JobInterestDevOps             <dbl> NA, NA, 1, NA, NA, NA, NA, NA, NA, NA, N…
## $ JobInterestFrontEnd           <dbl> NA, NA, 1, 1, 1, NA, NA, NA, NA, 1, NA, …
## $ JobInterestFullStack          <dbl> NA, 1, 1, 1, 1, NA, 1, NA, NA, 1, NA, NA…
## $ JobInterestGameDev            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 1, N…
## $ JobInterestInfoSec            <dbl> NA, NA, NA, NA, 1, NA, NA, NA, NA, NA, N…
## $ JobInterestMobile             <dbl> NA, NA, 1, NA, 1, NA, NA, NA, NA, NA, NA…
## $ JobInterestOther              <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ JobInterestProjMngr           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ JobInterestQAEngr             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 1, N…
## $ JobInterestUX                 <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 1, N…
## $ JobPref                       <chr> "start your own business", "work for a n…
## $ JobRelocateYesNo              <dbl> NA, 1, 1, NA, 1, NA, NA, NA, NA, 1, NA, …
## $ JobRoleInterest               <chr> NA, "Full-Stack Web Developer", "Front-E…
## $ JobWherePref                  <chr> NA, "in an office with other developers"…
## $ LanguageAtHome                <chr> "English", "English", "Spanish", "Portug…
## $ MaritalStatus                 <chr> "married or domestic partnership", "sing…
## $ MoneyForLearning              <dbl> 150, 80, 1000, 0, 0, 200, 0, 0, 700, 100…
## $ MonthsProgramming             <dbl> 6, 6, 5, 5, 24, 12, 12, 4, 29, 18, 5, 1,…
## $ NetworkID                     <chr> "6f1fbc6b2b", "f8f8be6910", "2ed189768e"…
## $ Part1EndTime                  <dttm> 2017-03-09 00:36:22, 2017-03-09 00:37:0…
## $ Part1StartTime                <dttm> 2017-03-09 00:32:59, 2017-03-09 00:33:2…
## $ Part2EndTime                  <dttm> 2017-03-09 00:59:46, 2017-03-09 00:38:5…
## $ Part2StartTime                <dttm> 2017-03-09 00:36:26, 2017-03-09 00:37:1…
## $ PodcastChangeLog              <dbl> NA, NA, 1, NA, NA, NA, NA, NA, NA, NA, N…
## $ PodcastCodeNewbie             <dbl> NA, 1, NA, NA, NA, NA, NA, 1, NA, 1, 1, …
## $ PodcastCodePen                <dbl> NA, NA, 1, NA, NA, NA, NA, NA, NA, NA, 1…
## $ PodcastDevTea                 <dbl> 1, NA, NA, NA, NA, NA, NA, NA, 1, NA, NA…
## $ PodcastDotNET                 <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ PodcastGiantRobots            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ PodcastJSAir                  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 1, N…
## $ PodcastJSJabber               <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 1, N…
## $ PodcastNone                   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ PodcastOther                  <chr> NA, NA, "Codenewbie", NA, NA, NA, NA, NA…
## $ PodcastProgThrowdown          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ PodcastRubyRogues             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ PodcastSEDaily                <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ PodcastSERadio                <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ PodcastShopTalk               <dbl> NA, NA, 1, NA, NA, NA, NA, NA, NA, NA, N…
## $ PodcastTalkPython             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ PodcastTheWebAhead            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ ResourceCodecademy            <dbl> 1, 1, 1, NA, NA, NA, 1, NA, NA, NA, 1, N…
## $ ResourceCodeWars              <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ ResourceCoursera              <dbl> NA, NA, NA, NA, NA, NA, NA, 1, NA, NA, N…
## $ ResourceCSS                   <dbl> NA, 1, 1, NA, NA, NA, NA, NA, NA, 1, NA,…
```

```
## $ ResourceEdX                  <dbl> NA, NA, NA, NA, NA, 1, NA, 1, NA, NA, NA…
## $ ResourceEgghead              <dbl> NA, NA, NA, 1, NA, NA, NA, NA, 1, NA, NA…
## $ ResourceFCC                  <dbl> 1, 1, 1, 1, NA, NA, 1, 1, NA, 1, 1, 1, 1…
## $ ResourceHackerRank           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ ResourceKA                   <dbl> NA, NA, NA, NA, NA, 1, NA, NA, NA, 1, 1,…
## $ ResourceLynda                <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ ResourceMDN                  <dbl> 1, NA, 1, 1, NA, NA, NA, NA, 1, NA, NA, …
## $ ResourceOdinProj             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ ResourceOther                <chr> NA, NA, NA, NA, NA, NA, "Sololearn", NA,…
## $ ResourcePluralSight          <dbl> NA, NA, NA, NA, NA, 1, 1, NA, NA, 1, NA,…
## $ ResourceSkillcrush           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ ResourceSO                   <dbl> NA, 1, NA, 1, 1, NA, 1, NA, 1, 1, NA, NA…
## $ ResourceTreehouse            <dbl> NA, NA, NA, NA, NA, 1, NA, NA, NA, NA, N…
## $ ResourceUdacity              <dbl> NA, NA, 1, NA, NA, NA, NA, 1, NA, 1, NA,…
## $ ResourceUdemy                <dbl> 1, 1, 1, NA, NA, NA, NA, NA, NA, NA, 1, …
## $ ResourceW3S                  <dbl> 1, 1, NA, NA, NA, NA, 1, NA, NA, NA, 1, …
## $ SchoolDegree                 <chr> "some college credit, no degree", "some …
## $ SchoolMajor                  <chr> NA, NA, NA, NA, "Information Technology"…
## $ StudentDebtOwe               <dbl> NA, NA, NA, NA, NA, NA, 8000, NA, NA, NA…
## $ YouTubeCodeCourse            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ YouTubeCodingTrain           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 1, N…
## $ YouTubeCodingTut360          <dbl> NA, NA, 1, NA, NA, NA, NA, NA, NA, 1, 1,…
## $ YouTubeComputerphile         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ YouTubeDerekBanas            <dbl> NA, NA, 1, NA, NA, NA, NA, 1, NA, 1, NA,…
## $ YouTubeDevTips               <dbl> NA, NA, 1, 1, NA, NA, NA, 1, NA, 1, 1, 1…
## $ YouTubeEngineeredTruth       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ YouTubeFCC                   <dbl> NA, 1, NA, 1, NA, NA, NA, 1, NA, 1, 1, 1…
## $ YouTubeFunFunFunction        <dbl> NA, NA, NA, 1, NA, NA, NA, 1, NA, 1, NA,…
## $ YouTubeGoogleDev             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ YouTubeLearnCode             <dbl> NA, NA, 1, NA, NA, NA, NA, NA, NA, 1, NA…
## $ YouTubeLevelUpTuts           <dbl> NA, NA, 1, 1, NA, NA, NA, NA, NA, 1, NA,…
## $ YouTubeMIT                   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 1, N…
## $ YouTubeMozillaHacks          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ YouTubeOther                 <chr> NA, NA, NA, NA, NA, "CodingEntrepreneurs…
## $ YouTubeSimplilearn           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ YouTubeTheNewBoston          <dbl> NA, NA, NA, NA, NA, 1, NA, 1, NA, 1, NA,…
```

# Checking for Sample Representativity

Most of the courses we offer are on web and mobile development, but we also cover many other domains, like data science, game development, etc. For the purpose of our analysis, we want to answer questions about a population of new coders that are interested in the subjects we teach. We'd like to know:

> Where are these new coders located. What are the locations with the greatest number of new coders. How much money new coders are willing to spend on learning.

Before starting to analyze the sample data we have, we need to clarify whether it's representative for our

population of interest and it has the right categories of people for our purpose.

The categories would be found in the `JobInterest...` &/or `JobRoleInterest` columns. If a person is interested in working in a certain area, they'll most probably learning more about it.

```
View(survey_data)
```

```
#split-and-combine workflow
library(dplyr)
frequency_table_interests <- survey_data %>%
                             group_by(JobRoleInterest) %>%
                             summarise(freq = n()*100/nrow(survey_data)) %>%
                             arrange(desc(freq))
```

We need to split the responses with more than 1 area of interest. We need to drop NA values

```
split_interests <- survey_data %>%
                    select(JobRoleInterest) %>%
                    drop_na() %>%                  #no. reduced from 18,175 to 6,992
                    rowwise %>%
                    mutate(options = length(str_split(JobRoleInterest, ",")[[1]]))
```

Frequency table for the options variable in the split_interests table

```
no_of_options <- split_interests %>%
                  ungroup() %>%                     #this is needed because we used the rowwi
se() function before
                  group_by(options) %>%
                  summarize(freq = n()*100/nrow(split_interests))
```

31.65% of the participants have clarity on their area of interest. The rest have mixed interests. Let's find out how many chose web or mobile development, since those are our focus areas.

```
web_or_mobile <- str_detect(survey_data$JobRoleInterest, "Web Developer|Mobile Developer")
developers <- table(web_or_mobile)
developers <- developers*100/sum(developers)
```

From the frequency table, about 86% of the population is interested in our focus ares. Therefore the sample is representative of our population of interest. Visualizing the same below:

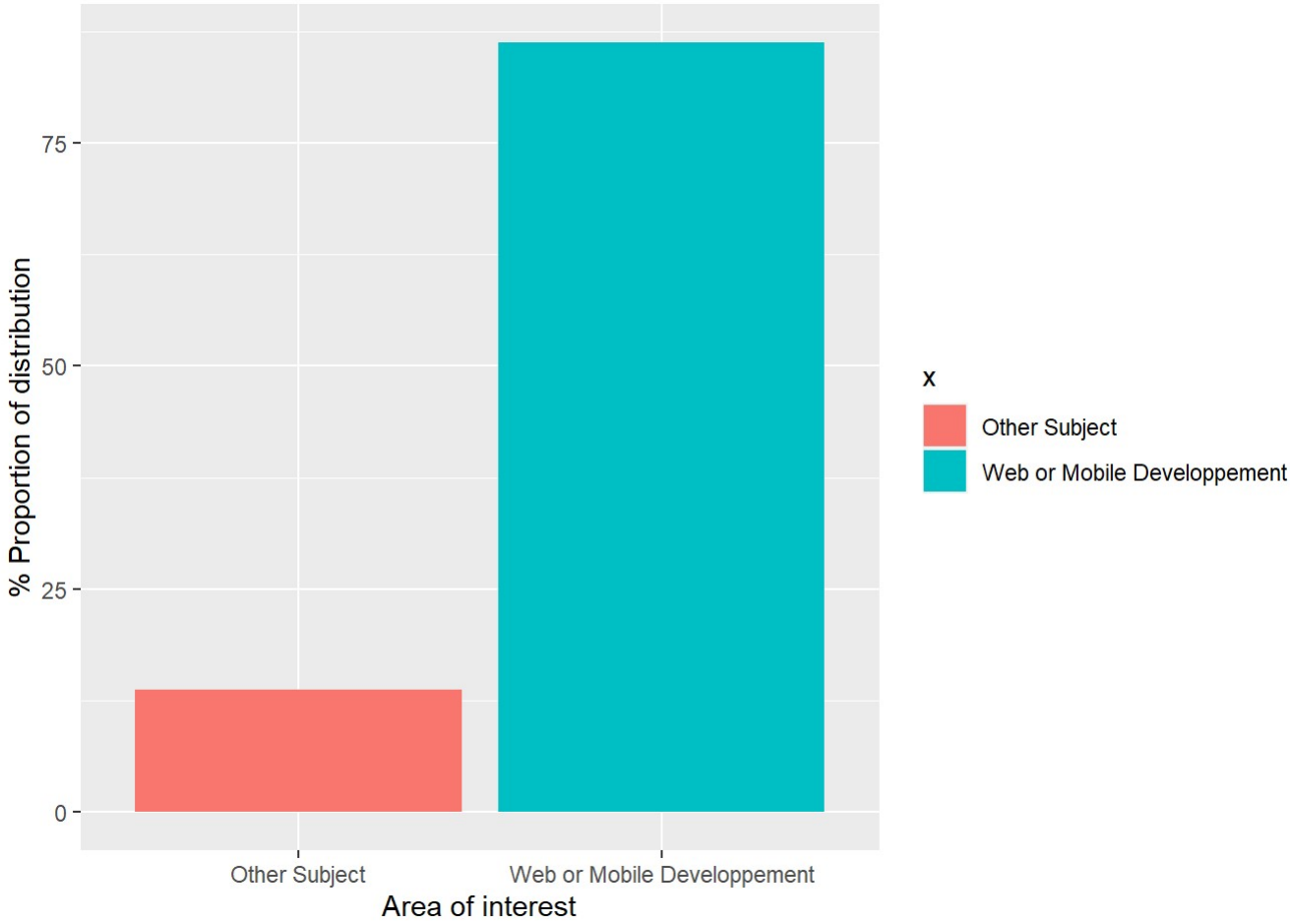
```
df <- tibble::tibble(x = c("Other Subject","Web or Mobile Developpement"),
                     y = developers)

library(ggplot2)

ggplot(data = df, aes(x = x, y = y, fill = x)) +
  geom_histogram(stat = "identity") +
  xlab("Area of interest") +
  ylab("% Proportion of distribution")
```

```
## Warning in geom_histogram(stat = "identity"): Ignoring unknown parameters:
## `binwidth`, `bins`, and `pad`
```

```
## Don't know how to automatically pick scale for object of type <table>.
## Defaulting to continuous.
```




We want to advertise our courses to people interested in all sorts of programming niches but mostly web and mobile development.

# New Coders - Locations and Densities

We'll consider the `CountryLive` variable as the markets we could advertise in. It describes the countries participants are currently located in. We could assume that's where they live and work.

Finding densities in each market. This will increase probability of purchases taking place if we ran ad campaigns in those locations.

```
countries <- survey_data %>%
               filter(JobRoleInterest != "NA") %>%  #alternatively use *tidyr::drop_na(JobRol
eInterest)*
               group_by(CountryLive) %>%
               summarise(frequency = n()) %>%
               arrange(-frequency)
```

```
countries <- countries %>%
              mutate(percentage = frequency*100/sum(frequency))
```

US is the leading market with 44.69% of respondents located therein. It's followed by India, UK & Canada having 7.5, 4.5 & 3.7% respectively. It would be interesting to know how much people will be willing to spend in these top markets. So that we do not make an unworthy investment in a market where people are only willing to learn for free, as this would not be profitable for the company.

# Spending Money for Learning

The `MoneyForLearning` column describes in American dollars the amount of money spent by participants from the moment they started coding until the moment they completed the survey. Our company sells subscriptions at a price of $59 per month, we're therefore interested in finding out how much money each student spends per month. We'll use the `MonthsProgramming` & `MoneyForLearning` variables to find average spent per month. We'll narrow down our analysis to only these 4 countries because: -These are the countries having the highest absolute frequencies in our sample, which means we have a decent amount of data for each. -Our courses are written in English, and English is an official language in all these four countries. The more people that know English, the better our chances to target the right people with our ads.

```
months_learning <- survey_data %>%
                    group_by(MonthsProgramming) %>%
                    summarise(freq = n())
```

The minimum number of months is 0, & there are 1091 responses with NA. Replace 0 with 1 to avoid errors, assuming they just started learning.

```
survey_data2 <- survey_data %>%
                 drop_na(JobRoleInterest)

survey_data2 <- survey_data2 %>%
                mutate(MonthsProgramming = replace(MonthsProgramming, MonthsProgramming==0,
1))

months_learning2 <- survey_data2 %>%
                     group_by(MonthsProgramming) %>%
                     summarise(freq = n())
```

Also remove responses where `MoneyForLearning` is NA

```
survey_data2 <- survey_data2 %>%
                drop_na(MoneyForLearning)
```

Finding average spent per month

```
survey_data2 <- survey_data2 %>%
                mutate(MoneyPerMonth = MoneyForLearning/MonthsProgramming) %>%
                drop_na(MoneyPerMonth)
```

We want to group the data by country, and then measure the average amount of money that students spend per month in each country. First, let's remove the rows having `NA` values for the `CountryLive` column, and check out if we still have enough data for the four countries that interest us.

```
survey_data2 <- survey_data2 %>%
                drop_na(CountryLive)
```

```
countries2 <- survey_data2 %>%
                group_by(CountryLive) %>%
                summarise(freq = n()) %>%
                arrange(desc(freq))
```

Let's compute the average spent per month in each of the 4 countries

```
top4countries <- c("United States of America","India","United Kingdom","Canada")

countries_mean <- survey_data2 %>%
                  filter(CountryLive %in% top4countries) %>%
                  group_by(CountryLive) %>%
                  summarise(country_mean = mean(MoneyPerMonth),
                            sum = sum(MoneyPerMonth),
                            freq = n()) %>%
                  arrange(-country_mean)
```

The results for the United Kingdom and Canada are a bit surprisingly low relative to the values we see for India. If we considered a few socio-economical metrics (like GDP per capita (https://bit.ly/2I3cukh)), we'd intuitively expect people in the UK and Canada to spend more on learning than people in India. It might be that we don't have have enough representative data for the United Kingdom and Canada, or we have some outliers (maybe coming from wrong survey answers) making the mean too large for India, or too low for the UK and Canada. Or it might be that the results are correct.

# Dealing with Extreme Outliers

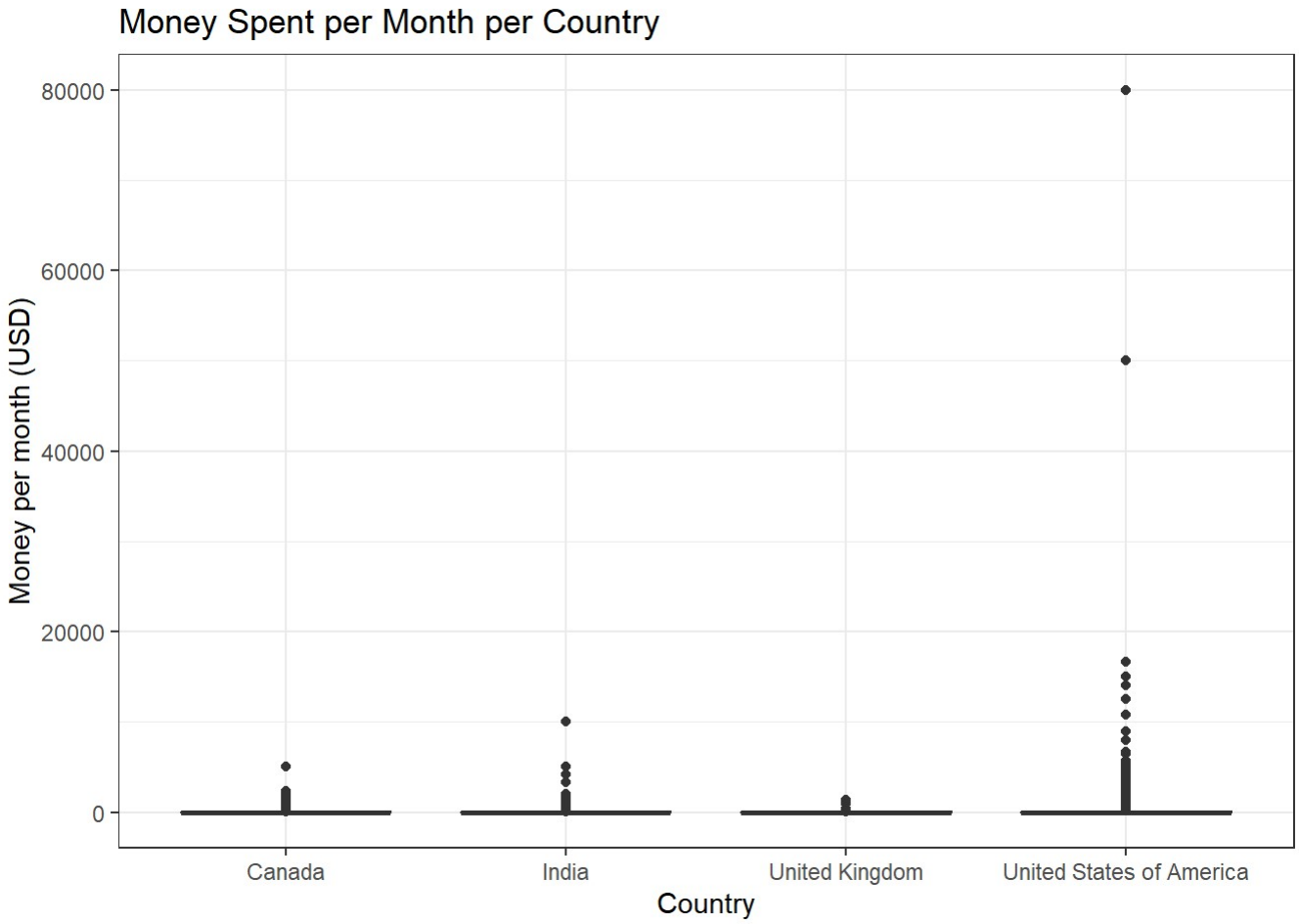
Let's use box plots to visualize the distribution of the `MoneyPerMonth` variable for each country.

Since maybe, we will remove elements from the database, we add an index column containing the number of each row. Hence, we will have a match with the original database in case of some indexes.

```
survey_data_top4 <- survey_data2 %>%
                     filter(CountryLive == "United States of America"|CountryLive == "India"|
CountryLive == "United Kingdom"|CountryLive == "Canada") %>%
                     mutate(index = row_number())
```

```
boxplot1 <- ggplot(data = survey_data_top4,
                   aes(x=CountryLive,y=MoneyPerMonth)) +
              geom_boxplot() +
              ggtitle("Money Spent per Month per Country") +
              xlab("Country") +
              ylab("Money per month (USD)") +
              theme_bw()

boxplot1
```




Seems like there are 2 outliers for US, we'll therefore remove values above $20,000 since it is unlikely that an individual can spend that much in a month.

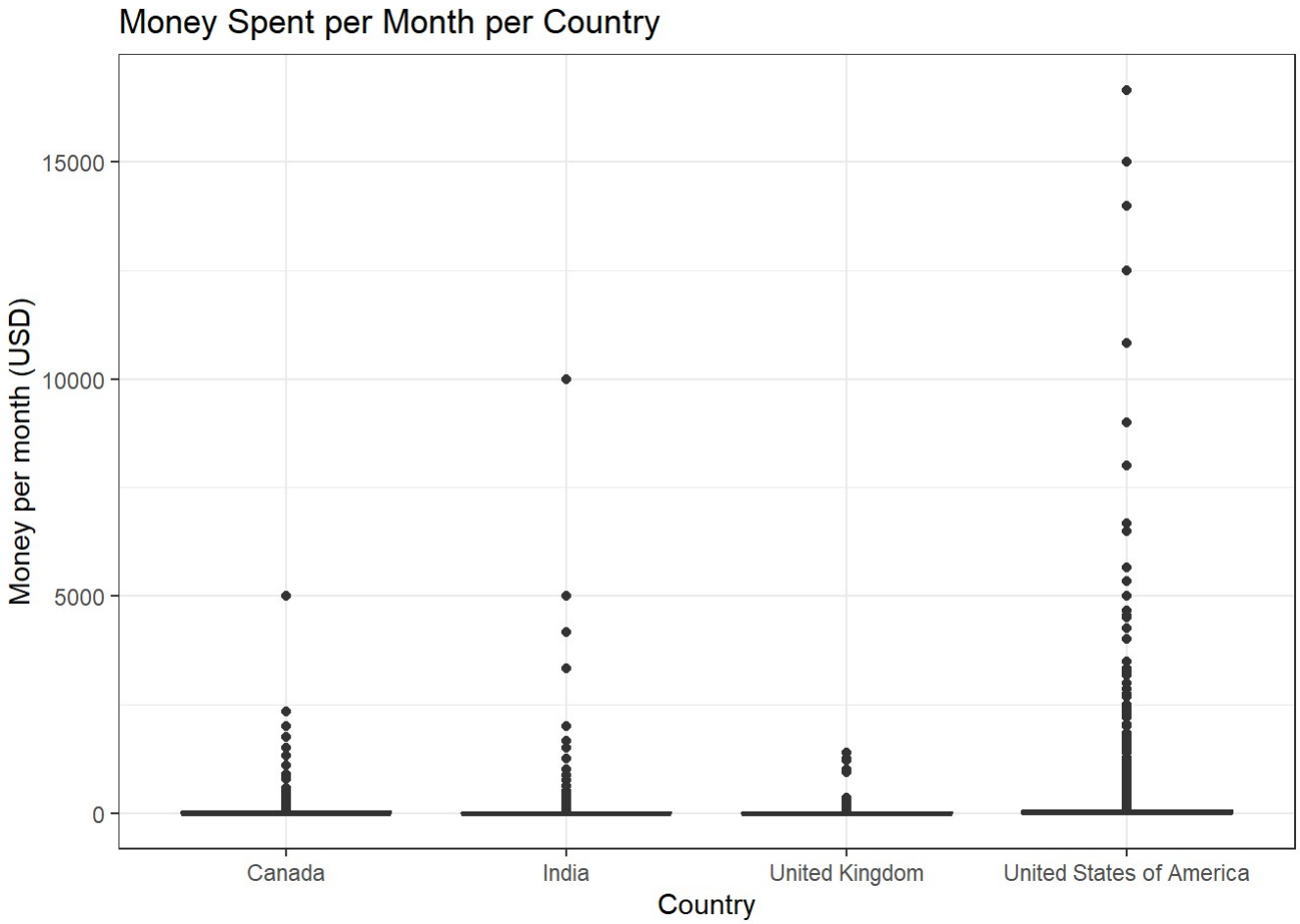
```
survey_data_top4 <- survey_data_top4 %>%
                    filter(MoneyPerMonth < 20000)
```

Let's redo the mean & boxplot

```
countries_mean2 <- survey_data_top4 %>%
                   group_by(CountryLive) %>%
                   summarise(country_mean = mean(MoneyPerMonth),
                             sum = sum(MoneyPerMonth),
                             freq = n()) %>%
                   arrange(-country_mean)
```

```
boxplot2 <- ggplot(data = survey_data_top4,
                   aes(x=CountryLive,y=MoneyPerMonth)) +
             geom_boxplot() +
             ggtitle("Money Spent per Month per Country") +
             xlab("Country") +
             ylab("Money per month (USD)") +
             theme_bw()

boxplot2
```




We can see a few extreme outliers for India (values over $2,500 per month), but it's unclear whether this is good data or not. Maybe these persons attended several bootcamps, which tend to be very expensive. Let's examine these two data points to see if we can find anything relevant.

```
outliers_india <- survey_data_top4 %>%
                   filter(CountryLive == "India", MoneyPerMonth >= 2500)
```

It seems that neither participant attended a bootcamp. Overall, it's really hard to figure out from the data whether these persons really spent that much money with learning. The actual question of the survey was *"Aside from university tuition, about how much money have you spent on learning to code so far (in US dollars)?"*, so they might have misunderstood and thought university tuition is included. It seems safer to remove these six rows.

```
survey_data_top4 <- survey_data_top4 %>%
                     filter(!(index %in% outliers_india$index))
```

The boxplot also revealed more outliers for US, above $6,000. Lets's examine these too.

```
outliers_us <- survey_data_top4 %>%
                filter(CountryLive =="United States of America", MoneyPerMonth >= 6000)
```

6/11 of these extreme outliers, indicate six people attended bootcamps, which justify the large sums of money spent on learning. For the other five, it's hard to figure out from the data where they could have spent that much money on learning. Consequently, we'll remove those rows where participants reported that they spend $6,000 each month, but they have never attended a bootcamp.

Also, the data shows that 8 respondents have `MonthsProgramming` <=3 when they completed the survey. They most likely paid a large sum of money for a bootcamp that was going to last for several months, so the amount of money spent per month is unrealistic and should be significantly lower (because they probably didn't spend anything for the next couple of months after the survey). As a consequence, we'll remove these 8 outliers.

```
no_bootcamp <- survey_data_top4 %>%
                      filter(CountryLive == "United States of America" &
                             MoneyPerMonth >= 6000 &
                             AttendedBootcamp == 0)
```

```
survey_data_top4 <- survey_data_top4 %>%
                     filter(!(index %in% no_bootcamp$index))
```

```
less_than3months <- survey_data_top4 %>%
                     filter(CountryLive == "United States of America" &
                            MoneyPerMonth >= 6000 &
                            MonthsProgramming <= 3)
```

```
survey_data_top4 <- survey_data_top4 %>%
                     filter(!(index %in% less_than3months$index))
```

The boxplot for Canada shows there's 1 outlier, respondent spent $5,000 per month. Let's exammine

```
outliers_canada <- survey_data_top4 %>%
                    filter(CountryLive == "Canada" &
                           MoneyPerMonth >= 5000 &
                           MonthsProgramming <= 3)
```

Here, the situation is similar to some of the US respondents — this participant had been programming for no

more than two months when he completed the survey. He seems to have paid a large sum of money in the beginning to enroll in a bootcamp, and then he probably didn't spend anything for the next couple of months after the survey. We'll take the same approach here as for the US and remove this outlier.

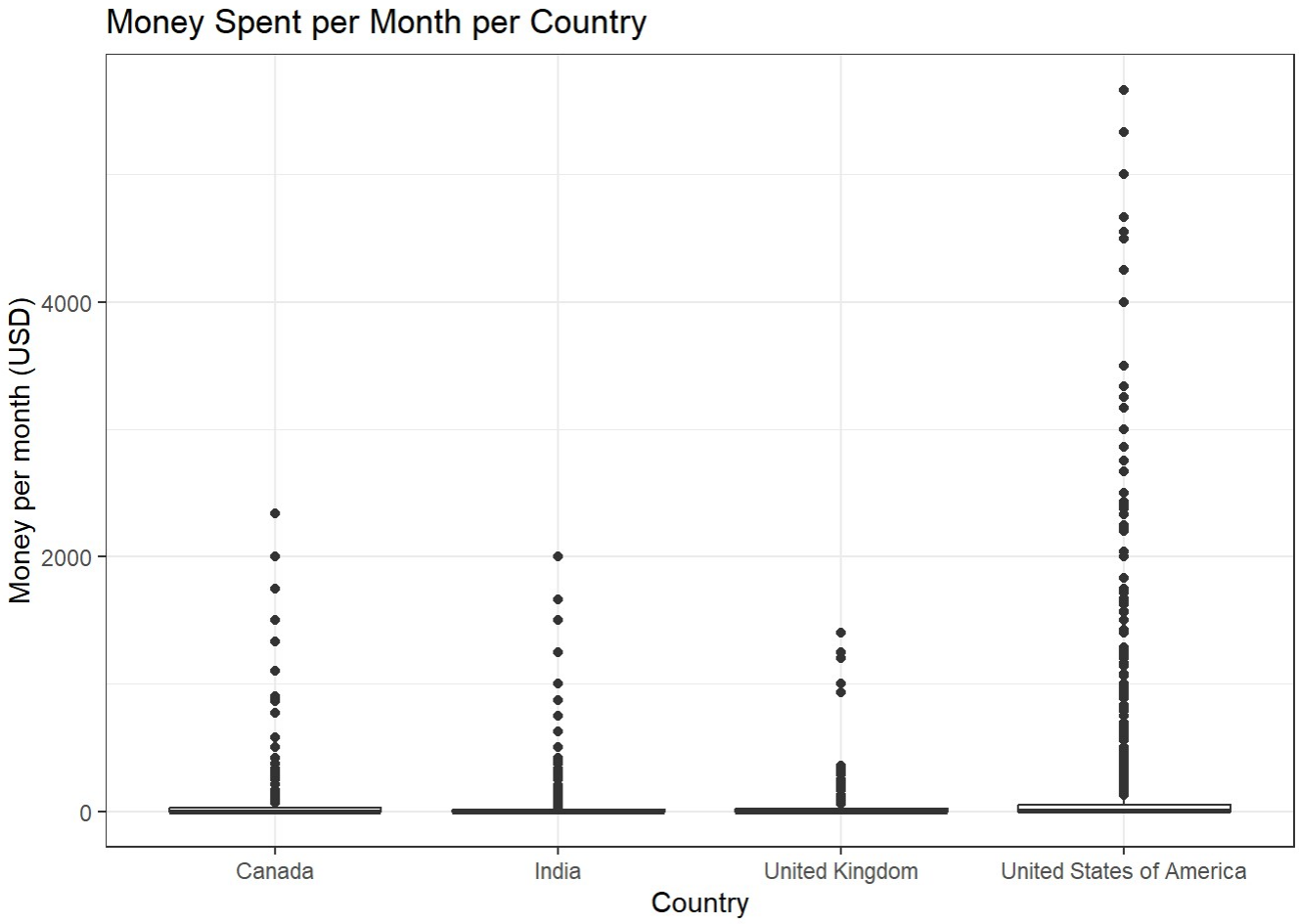
```
survey_data_top4 <- survey_data_top4 %>%
                     filter(!(index %in% outliers_canada$index))
```

Let's recompute the mean values and generate the final box plots.

```
countries_mean3 <- survey_data_top4 %>%
                   group_by(CountryLive) %>%
                   summarise(country_mean = mean(MoneyPerMonth),
                             sum = sum(MoneyPerMonth),
                             freq = n()) %>%
                   arrange(-country_mean)
```

```
boxplot3 <- ggplot(data = survey_data_top4,
                   aes(x=CountryLive,y=MoneyPerMonth)) +
             geom_boxplot() +
             ggtitle("Money Spent per Month per Country") +
             xlab("Country") +
             ylab("Money per month (USD)") +
             theme_bw()

boxplot3
```




# Choosing the Two Best Markets

The country mean results suggest that US is a good market A significant number of respondents live there and coders are willing to spend roughly about $143 per month. Canada seems to be the second best market we could try running our ad campaigns. Coders there are willing to spend roughly about $93 which is above the price of our subscriptions of $59 per month.

The data suggests strongly that we shouldn't advertise in the UK, we might consider taking a closer look at India before deciding to choose Canada as our second best choice. Although it seems more tempting to choose Canada, there are good chances that India might actually be a better choice because of the large number of potential customers.

We have several options: a) We could advertise in US, Canada & India & split the budget in these proportions: 50% for US; 30% for India, 20% for Canada or 50% for US; 25% for India, 25% for Canada

b. We could advertise in US & Canada only or US & India 70% for US; 30% for India or 65% for US; 35% for Canada

c. Advertise in the US only.

It's probably best to send our analysis to the marketing team and let them use their domain knowledge to decide. They might want to do some extra surveys in India and Canada and then get back to us for analyzing the new survey data.

# Conclusion

In this project, we analyzed survey data from new coders to find the best two markets to advertise in. The only solid conclusion we reached is that the US would be a good market to advertise in.

For the second best market, it wasn't clear-cut what to choose between India and Canada. We decided to send the results to the marketing team so they can use their domain knowledge to take the best decision.