Udemy Course Analysis

Victor Nyakako

2022-12-24

Introduction

Udemy is an online learning platform that offers broad range of courses. It is one of the go to platforms among learners seeking to upskill and broaden their horizons.

In this analysis, a dataset containing udemy courses up to 2017 is used. The datasets only focus on business, music, design, and web development courses.

Loading the Packages

The packages used are:

- tidyverse
- janitor
- lubridate
- skimr

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.4.0      ✔ purrr   0.3.5
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.5.0
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
##
## Attaching package: 'janitor'
##
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
##
##
## Loading required package: timechange
##
##
## Attaching package: 'lubridate'
##
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Loading the Dataset

```
web_courses<- read_csv("Web development Courses.csv")
```

```
## Rows: 1205 Columns: 12
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): course_title, url, level, subject
## dbl  (7): course_id, price, num_subscribers, num_reviews, num_lectures, Rati...
## dttm (1): published_timestamp
##
## 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.
```

```
music_courses<- read_csv("Music Courses.csv")
```

```
## Rows: 680 Columns: 12
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): course_title, url, level, subject
## dbl  (7): course_id, price, num_subscribers, num_reviews, num_lectures, Rati...
## dttm (1): published_timestamp
##
## 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.
```

```
business_courses<- read_csv("Business Courses.csv")
```

```
## Rows: 1192 Columns: 12
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): course_title, url, level, subject
## dbl  (7): course_id, price, num_subscribers, num_reviews, num_lectures, Rati...
## dttm (1): published_timestamp
##
## 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.
```

```
design_courses <- read_csv("Design Courses.csv")
```

```
## Rows: 604 Columns: 12
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): course_title, url, level, subject
## dbl  (7): course_id, price, num_subscribers, num_reviews, num_lectures, Rati...
## dttm (1): published_timestamp
##
## 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.
```

Merging the Data

Let's combine the datasets into a single dataset named "udemy_courses"

```
udemy_courses <- rbind(business_courses, music_courses, web_courses, design_courses)
```

Exploratory Data Analysis

Let us explore the data, and get to find its structure.

```
skim(udemy_courses)
```

Data summary

Name	udemy_courses
Number of rows	3681
Number of columns	12
_____	
Column type frequency:	
character	4
numeric	7
POSIXct	1
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
course_title	5	1	6	243	0	3663	0
url	5	1	29	93	0	3672	0
level	5	1	10	18	0	4	0
subject	4	1	14	19	0	4	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
course_id	5	1	675753.50	343130.44	8324	407474.00	687692.00	960814.00	1282064.0	
price	5	1	66.12	61.06	0	20.00	45.00	95.00	200.0	
num_subscribers	5	1	3199.26	9486.58	0	112.00	912.50	2558.00	268923.0	
num_reviews	5	1	156.31	935.67	0	4.00	18.00	67.00	27445.0	
num_lectures	5	1	40.13	50.40	0	15.00	25.00	46.00	779.0	
Rating	4	1	0.61	0.33	0	0.28	0.76	0.93	1.0	
content_duration	5	1	4.10	6.05	0	1.00	2.00	4.50	78.5	

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
published_timestamp	5	1	2011-07-09 05:43:31	2017-07-06 21:46:30	2016-01-27 17:58:18	3672

```
udemy_courses[is.na(udemy_courses$course_id),] #checking the subset with missing 'course_id'
```

```
## # A tibble: 5 × 12
##   course_id course_ti…¹ url   price num_s…² num_r…³ num_l…⁴ level Rating conte…⁵
##       <dbl> <chr>       <chr> <dbl>   <dbl>   <dbl>   <dbl> <chr>  <dbl>   <dbl>
## 1        NA <NA>        <NA>     NA      NA      NA      NA <NA>   0.690      NA
## 2        NA <NA>        <NA>     NA      NA      NA      NA <NA>  NA          NA
## 3        NA <NA>        <NA>     NA      NA      NA      NA <NA>  NA          NA
## 4        NA <NA>        <NA>     NA      NA      NA      NA <NA>  NA          NA
## 5        NA <NA>        <NA>     NA      NA      NA      NA <NA>  NA          NA
## # … with 2 more variables: published_timestamp <dttm>, subject <chr>, and
## #   abbreviated variable names ¹course_title, ²num_subscribers, ³num_reviews,
## #   ⁴num_lectures, ⁵content_duration
```

Missing Values

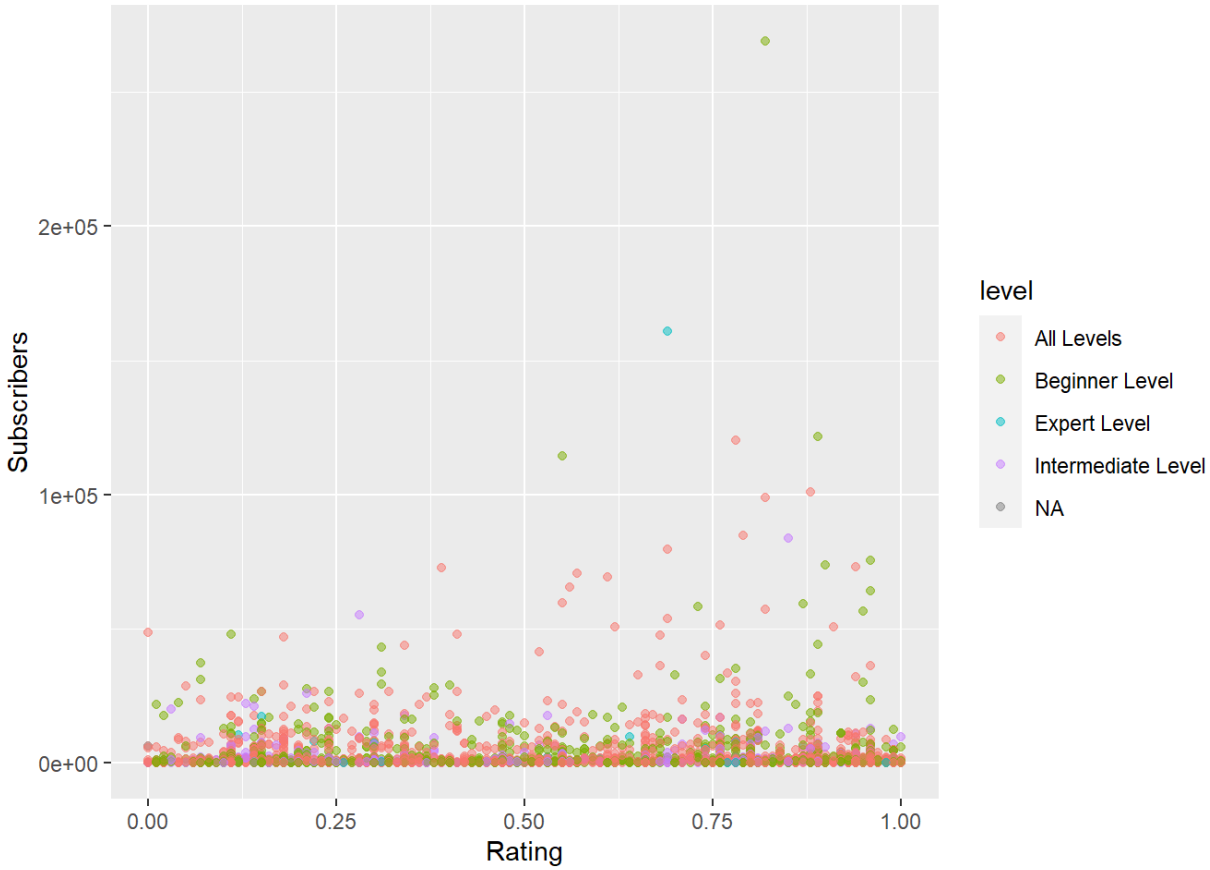
From the exploration using skim function, we learn that there are some columns with missing values. It turns out that the missing values stems from the null records of “course_id” column. The 3 rows with missing ‘course_id’ will thus be expunged.

Further Exploration

The data can be explored further. We can check on the correlation between subscription (num_subscribers) and rating. The business question would be, “does the subscription affect rating?”

From the scatter point below, it is evident that increased subscription does no affect ratings in any way across all levels.

```
ggplot(data= udemy_courses, aes(x= Rating , y = num_subscribers ))+
  geom_point(alpha = 0.5, aes(color = level))+
  labs(y= "Subscribers")
```

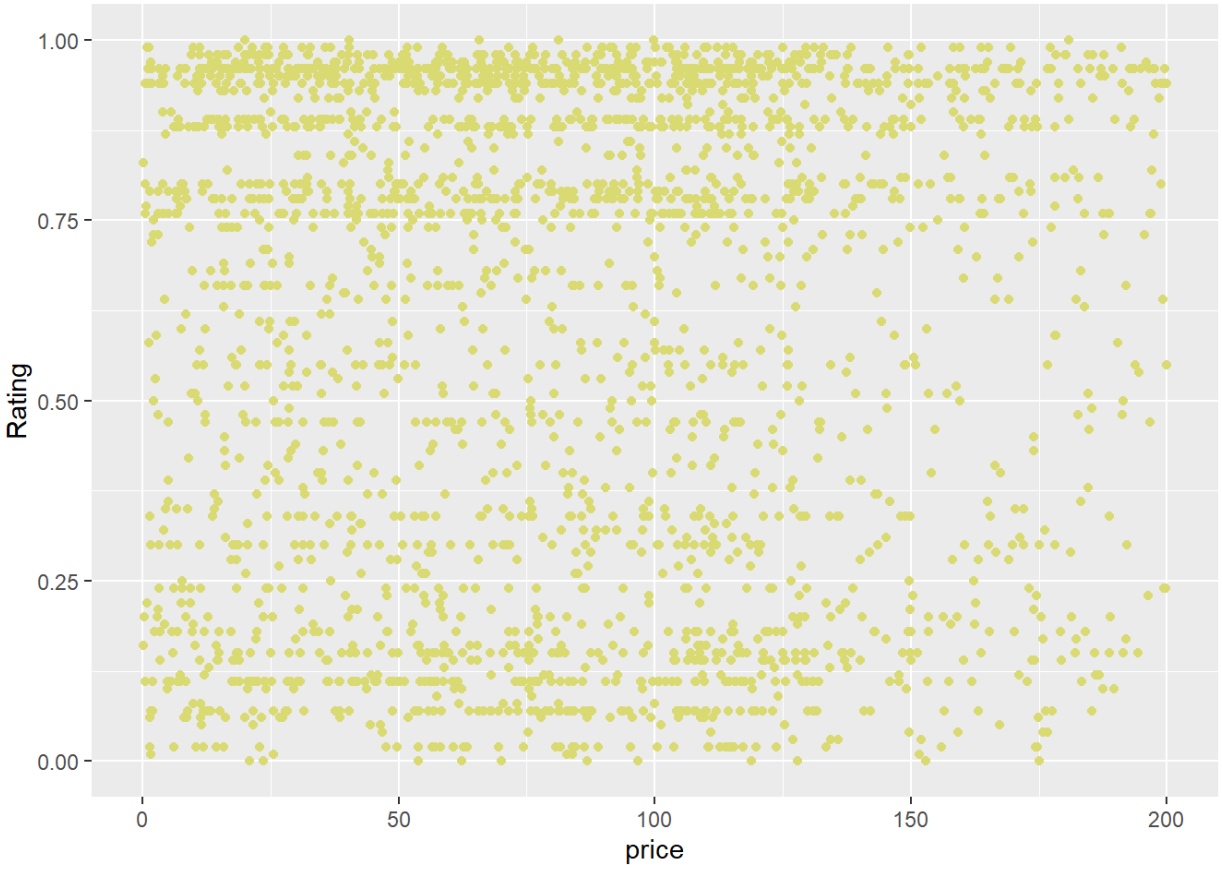


Price Versus Rating

Do highly priced courses receive higher ratings? It is often presumed that expensive courses are of great quality thus attracting an upward rating.

Evidently, there is no clear relationship between the price of the courses and their ratings. Surprisingly, many free courses have very high ratings.

```
ggplot(data= udemy_courses, aes(x= price, y = Rating))+
 geom_point(position = position_jitter(w=100, h=0), color= "#dada72")+
   scale_x_continuous(limits = c(0,200))
```



Data Cleaning

We discovered missing values in our dataset, let's remove them. Additionally, we do not need the url column, so we got to drop it off. We should also ensure that the columns are consistent, and clean_names from janitor package will be handy. Furthermore, we will remove the strings 'level' and 'levels' from the records in the level column. 'Beginner', 'Intermediate', etc., can suffice in place of 'Beginner level', 'All levels,' etc. str_replace_all function from the stringr package will be useful here.

```
udemy_courses <- udemy_courses[!is.na(udemy_courses$course_id),] #only return the rows where course_id is
 not null.
sum(is.na(udemy_courses))
```

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

```
udemy_courses<- udemy_courses %>%
  select(-url)# return all columns except "url"

udemy_courses<- clean_names(udemy_courses) # For consistent casing and naming of the columns

udemy_courses$level <- str_replace_all(udemy_courses$level, " Level", "")
udemy_courses$level <- str_replace_all(udemy_courses$level, " Levels", "")
udemy_courses$level <- ifelse(udemy_courses$level== "Alls", "All",udemy_courses$level)
```

Feature Engineering

There is need for the creation of some new features like, ‘year’ and hour. This will help in knowing the years and the hours the courses were posted.

```
udemy_courses$year <- year(udemy_courses$published_timestamp) # creating the 'year' column

udemy_courses$posting_hour <- hour(udemy_courses$published_timestamp) # creating the 'hour' column
udemy_courses$free_paid <- ifelse(udemy_courses$price == 0, "Free", "Paid") # creating 'free_paid' column
```

Analysis

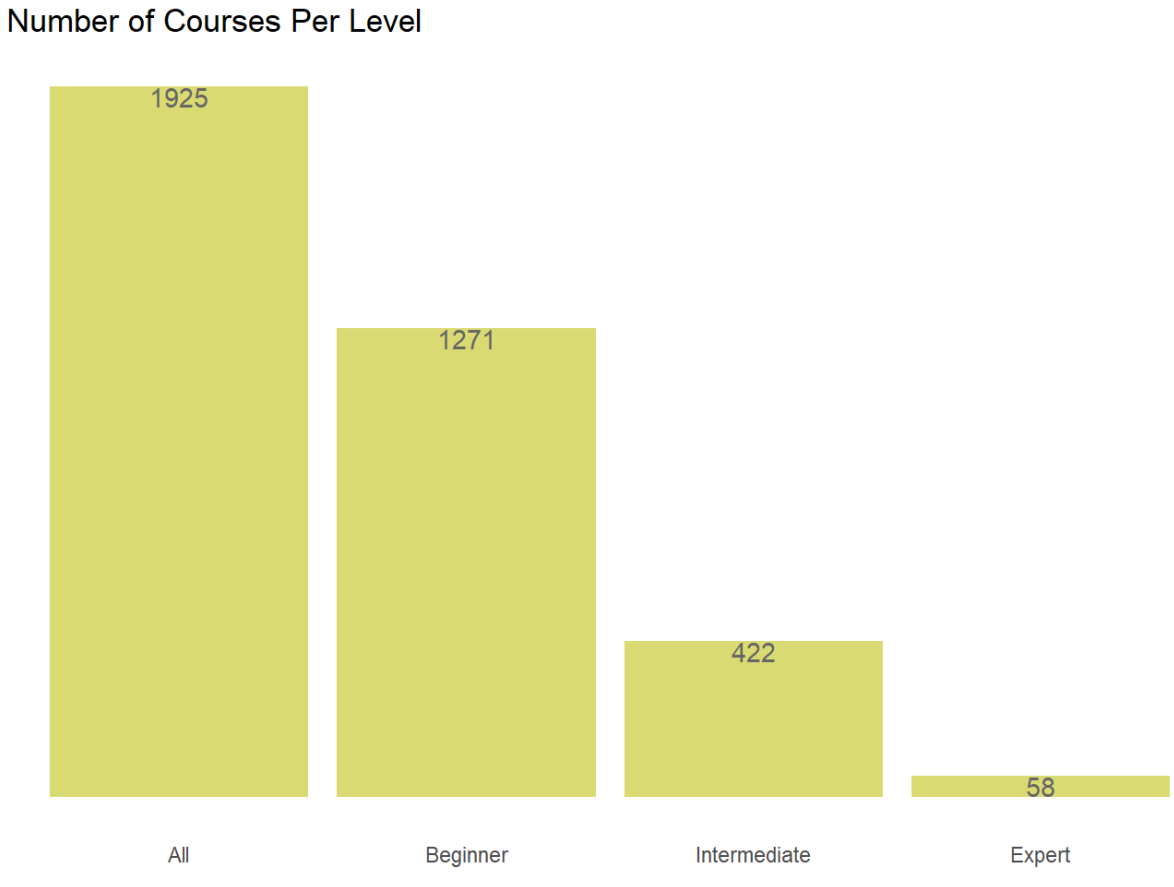
Our data has been explored and some cleaning done. Let us now conduct some analysis.

Number of Courses Per Level

We can see that the courses of all levels take the lead. Expert level courses are the least published during the period.

```
level_courses <- udemy_courses%>%
  group_by(level) %>%
  summarise(number_of_courses = n()) %>%
  arrange(desc(number_of_courses))

ggplot(level_courses, aes(x= reorder(level,-number_of_courses), y = number_of_courses))+
  geom_col(fill = "#dada72")+
   geom_text(aes(label = number_of_courses), color= "#666666", vjust = 1)+
  theme_classic()+
   theme(axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        axis.title.x = element_blank()
   )+
  labs(title = "Number of Courses Per Level")
```

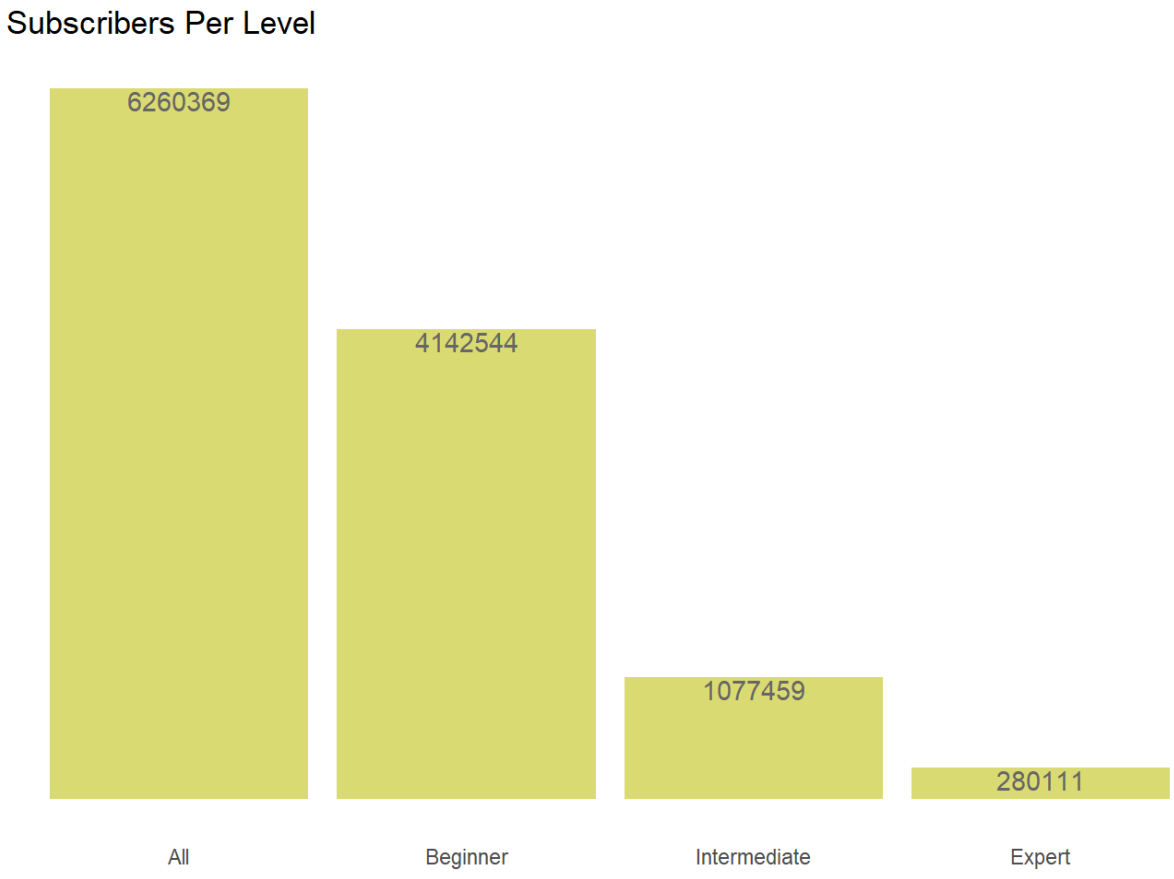


Number of Subscribers Per Level

Expert level subscribers are very few. All level subscribers still take the lead.

```
level_subscribers<- udemy_courses %>%
  group_by(level) %>%            # Creating a subset of the level subscribers
  summarise(subscribers = sum(num_subscribers)) %>%
arrange(desc(subscribers))

ggplot(level_subscribers, aes(x= reorder(level, -subscribers), y = subscribers ))+
  geom_col(fill= "#dada72" )+
  geom_text(aes(label = subscribers),color= "#666666", vjust = 1.1)+
  theme_classic()+
  theme(axis.text.y = element_blank(),
        axis.title.y = element_blank(),
        axis.title = element_blank(),
        axis.ticks = element_blank(),
        axis.line = element_blank()
        )+
  labs(title = "Subscribers Per Level")
```



The Number of Subscribers Per Subject

How many subscribers do we have per subject? Business Finance, All levels, are the courses with the highest number of subscribers. Conversely, graphic design expert level courses have the least number of subscribers.

```
subject_subscribers <- udemy_courses %>%
  group_by(subject, level) %>%
  summarise(subscribers = n()) %>%
  arrange(subject, -subscribers)
```

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

```
print(subject_subscribers)
```

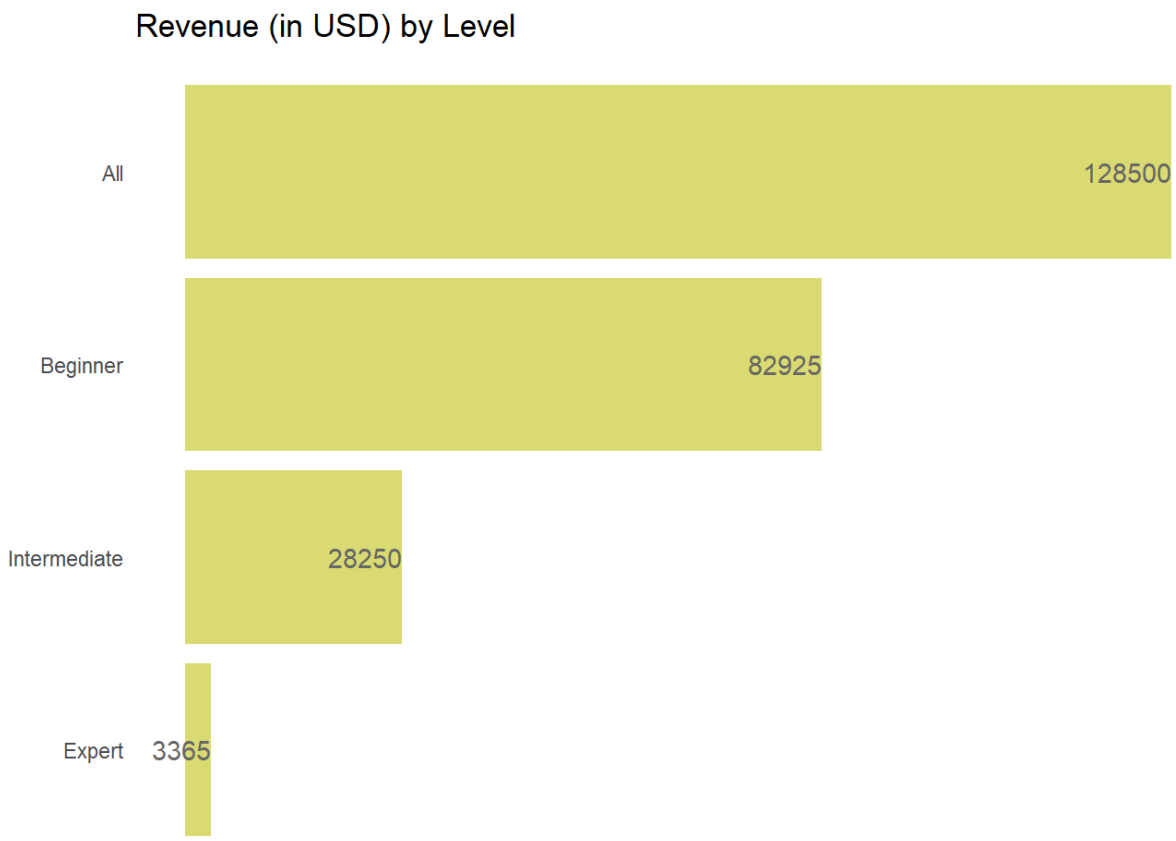
```
## # A tibble: 16 × 3
## # Groups:   subject [4]
##    subject             level        subscribers
##    <chr>               <chr>              <int>
##  1 Business Finance    All                  633
##  2 Business Finance    Beginner             399
##  3 Business Finance    Intermediate         134
##  4 Business Finance    Expert                25
##  5 Graphic Design      All                  335
##  6 Graphic Design      Beginner             184
##  7 Graphic Design      Intermediate          76
##  8 Graphic Design      Expert                 7
##  9 Musical Instruments All                  324
## 10 Musical Instruments Beginner             266
## 11 Musical Instruments Intermediate          78
## 12 Musical Instruments Expert                12
## 13 Web Development     All                  633
## 14 Web Development     Beginner             422
## 15 Web Development     Intermediate         134
## 16 Web Development     Expert                14
```

Revenue Per Level

What are the most priced course levels ? Which ones are charged cheaply? As expected, All level courses generated high revenue for udemy, while expert level courses generating the least amount of revenue.

```
level_revenue <- udemy_courses %>%
  group_by(level) %>%
  summarise(revenue = sum(price)) %>%
  arrange(-revenue)


ggplot(level_revenue, aes(x= reorder(level,revenue), y= revenue))+geom_col(fill ="#dada72" )+
  geom_text(aes(label= revenue) ,color= "#666666", hjust = 1)+
  labs(title = "Revenue (in USD) by Level")+
    theme_classic()+
  theme(axis.title = element_blank(),
        axis.text.x  = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank())+
  coord_flip()
```

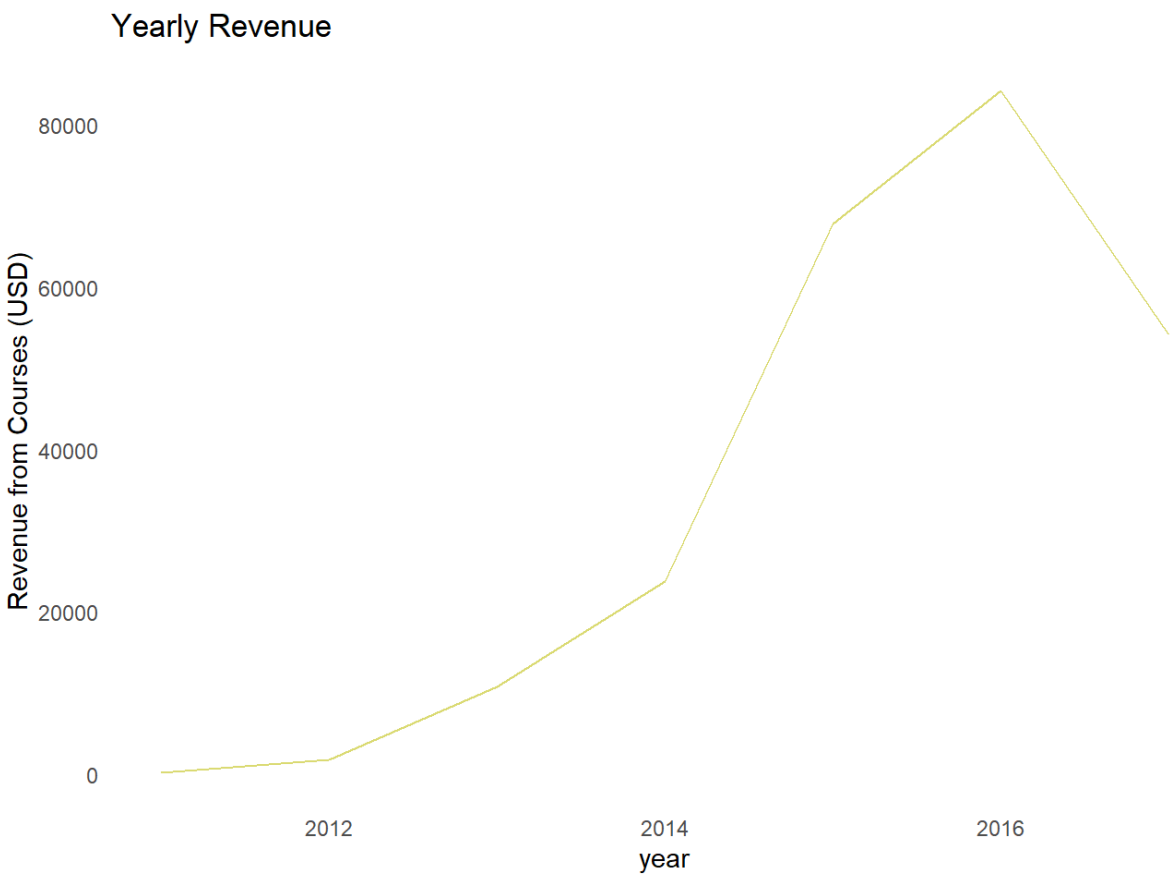


The Revenue Trend Across Years

There has been a steady increase in course revenues for udemy since 2011 to 2016. However, there was a sharp decline in revenue in 2016 through to 2017.

```
year_revenue <- udemy_courses %>%
  group_by(year) %>%
  summarise(revenue = sum(price))

ggplot(year_revenue, aes(x=year, y= revenue))+
  geom_line(color = "#dada72")+
  theme_classic()+
  theme(axis.line = element_blank(),
        axis.ticks = element_blank()
        )+
labs(title = "Yearly Revenue", y= "Revenue from Courses (USD)")
```

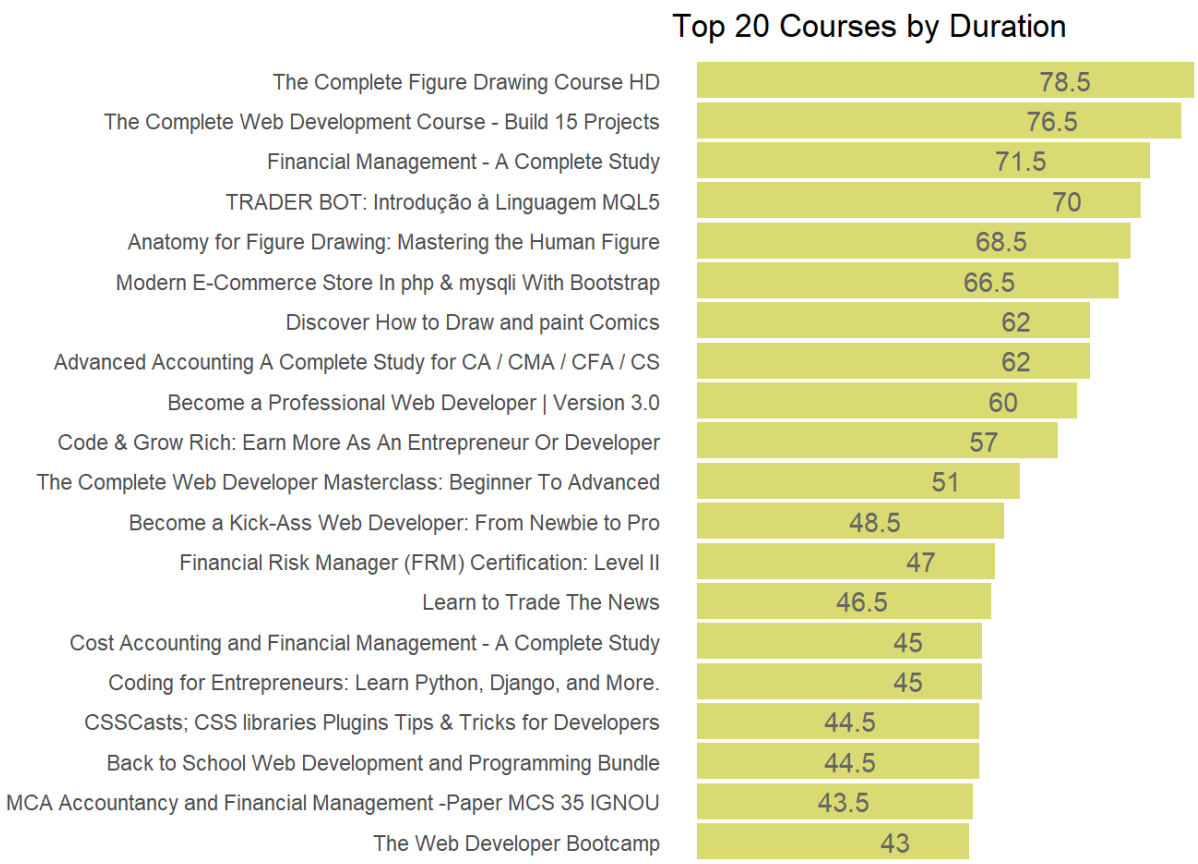


Top 20 Courses in Terms of Content Duration

“The Complete figure Drawing Course HD” is the longest course in terms of content duration. The course is 78.5 hours long.

```
top_20 <- udemy_courses %>%
  select(course_title, content_duration) %>%
  top_n(20, wt= content_duration) %>%
  arrange(-content_duration)

ggplot(top_20, aes(x= reorder(course_title, content_duration), y = content_duration))+
  geom_col(fill = "#dada72" )+
  theme_classic()+
  theme(axis.title.y = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        axis.title.x = element_blank(),
        axis.text.x = element_blank())+
   geom_text(aes(x = course_title, y = content_duration,
                 label = content_duration),color= "#666666", hjust = 3)+
  labs(title = "Top 20 Courses by Duration")+
  coord_flip()
```



The Preferred Posting Hours

For the period, the content creators posted their courses between 3PM and midnight, with most courses uploaded at 6 PM. This makes sense because most content creators are also full time employees, and they only find time for udemy project during off-work hours.

```
udemy_courses %>%
  group_by(posting_hour) %>%
  summarise(number_of_times = n()) %>%
  arrange(-number_of_times)
```

```
## # A tibble: 24 × 2
##    posting_hour number_of_times
##           <int>           <int>
##  1           18             397
##  2           17             377
##  3           21             360
##  4           22             320
##  5           16             291
##  6           20             278
##  7           19             255
##  8           23             227
##  9            0             219
## 10           15             187
## # … with 14 more rows
```

Free Courses and Paid Courses

Generally, there are many paid courses compared to free courses.

```
udemy_courses %>%
  group_by(free_paid) %>%
  summarise(number_of_courses = n())
```

```
## # A tibble: 2 × 2
##   free_paid number_of_courses
##   <chr>                <int>
## 1 Free                   311
## 2 Paid                  3365
```

```
free.paid <- udemy_courses %>%
  group_by(level, free_paid) %>%
  summarise(number_of_courses = n()) %>%
  arrange(-number_of_courses)
```

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

```
print(free.paid)
```

```
## # A tibble: 8 × 3
## # Groups:   level [4]
##   level        free_paid number_of_courses
##   <chr>        <chr>                <int>
## 1 All          Paid                  1756
## 2 Beginner     Paid                  1171
## 3 Intermediate Paid                   387
## 4 All          Free                   169
## 5 Beginner     Free                   100
## 6 Expert       Paid                    51
## 7 Intermediate Free                    35
## 8 Expert       Free                     7
```