CSC8631 report

Isaac Wheeler

26/11/2020

In this project, I am going to be providing some analysis of the video stats data; asking a few different questions and trying to find any sort of pattern within these. Video stats are only provided for years 3-7 so this is where the my analysis will be based upon. As mentioned in the executive summary, the data in these files is already in a very usable state. As such, data preparation takes place before I make different parts of my analysis and are included in the code chunks and briefly explained.

This report takes the form of asking a few questions about the data. I'll indicate what I am looking into, prepare the data to analyse, plot and analyse, before evaluating in conjunction with my understanding of what the client is looking for in the data.

Video engagement across years

Watch percentages

The first question I investigated is if there is any significant differences between years in terms of engagement with the videos. This was by done by looking into how the percentage watched (up to different percentages of the video) changed, if at all, across the years.

In order to do this, I knocked up a quick function that would work out across all the videos the average percentage of views that made it to differing thresholds of the video. These thresholds were 5%, 10%, 25%, 50%, 75%, 95% and 100%, and the data were all provided in the video stats csv files.

```
helper.avgper <- function(y)
{
  temp = rep(NULL,7)
  for (i in 9:15)
  {
    temp[i-8] = sum(y[i])/13
  }
  return(temp)
}
```

This simple helper function will, when provided with a video stats file of the same format, work out the average percentages watched for the year, and return it as a vector. This function can be found in the helpers file of the repository, along with one other that we're going to use later.

So to carry out the analysis, we simply need to run the function for all of our years and then plot it.

```
y3a=helper.avgper(y3)
y4a=helper.avgper(y4)
y5a=helper.avgper(y5)
```

```
y6a=helper.avgper(y6)
y7a=helper.avgper(y7)
```

We store each vector on its own. For the benefit of readability alone I will put this all into a dataframe also.

```
a5p = c(y3a[1],y4a[1],y5a[1],y6a[1],y7a[1])
a10p = c(y3a[2],y4a[2],y5a[2],y6a[2],y7a[2])
a25p = c(y3a[3],y4a[3],y5a[3],y6a[3],y7a[3])
a50p = c(y3a[4],y4a[4],y5a[4],y6a[4],y7a[4])
a75p = c(y3a[5],y4a[5],y5a[5],y6a[5],y7a[5])
a95p = c(y3a[6],y4a[6],y5a[6],y6a[6],y7a[6])
a100p = c(y3a[7],y4a[7],y5a[7],y6a[7],y7a[7])

labe = c("5%","10%","25%","50%","75%","95%","100%")
(viddf = data.frame(year=3:7,"5%"=a5p,"10%"=a10p,"25%"=a25p,
                    "50%"=a50p,"75%"=a75p,"95%"=a95p,"100%"=a100p))
```

```
##   year      X5.     X10.     X25.     X50.     X75.     X95.    X100.
## 1    3 74.25846 72.95462 71.10846 68.48462 66.63923 64.25231 56.34308
## 2    4 73.45077 71.77538 69.59462 66.64692 64.47923 62.24154 55.43231
## 3    5 78.95000 77.57231 75.00923 72.51154 70.70231 68.69308 60.56000
## 4    6 78.67692 76.73000 73.97385 71.03385 68.99154 66.77308 57.92846
## 5    7 74.65308 72.84308 69.24000 66.28231 64.25308 62.05000 55.65769
```

There are some interesting things to draw already from this data. Year five was a good year for video engagement in terms of overall percentages, whereas year four was actually the worst. We can also see that even for the most engaged years, almost 20% of video views do not even get 5% into the video. Further, there is a consistent drop off of views that overall means videos lose about 15-20% of the views as the video goes on to the finish. In order to visualise this fully, we're going to want to plot our data. Notice how I pointed out that the reason for the above dataframe was readability alone; we are going to use something else for our plot. GGplot requires that you use a data frame when you plot data, so why do we need another data frame when we've got a perfectly good one sitting around already. Three words; grouped bar plot.

```
year = c(rep("year 3",7),rep("year 4",7),rep("year 5",7),
         rep("year 6",7),rep("year 7",7))
perwatched = rep(c("5%","10%","25%","50%","75%","95%","100%"),5)
values = c(y3a,y4a,y5a,y6a,y7a)
(datas = data.frame(year,perwatched,values))
```

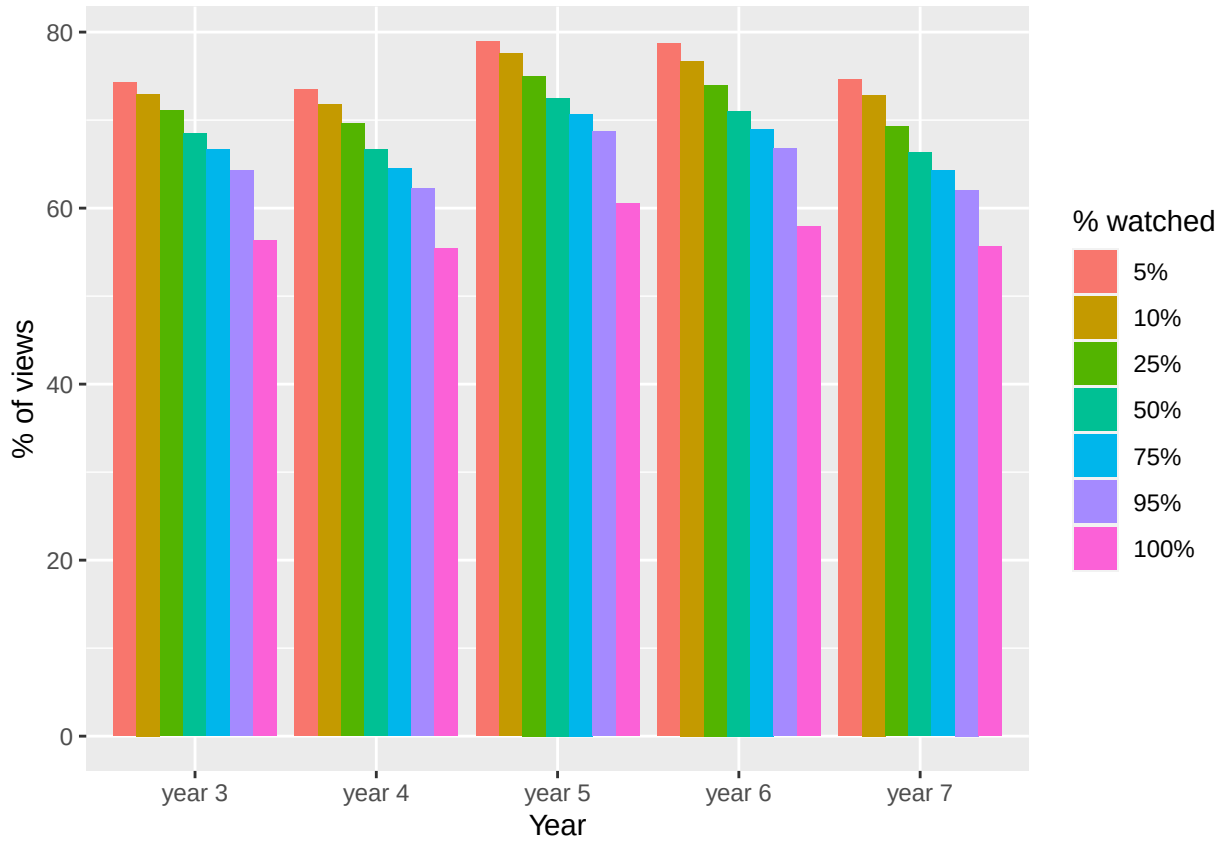
```
##       year perwatched   values
## 1  year 3         5% 74.25846
## 2  year 3        10% 72.95462
## 3  year 3        25% 71.10846
## 4  year 3        50% 68.48462
## 5  year 3        75% 66.63923
## 6  year 3        95% 64.25231
## 7  year 3       100% 56.34308
## 8  year 4         5% 73.45077
## 9  year 4        10% 71.77538
## 10 year 4        25% 69.59462
## 11 year 4        50% 66.64692
## 12 year 4        75% 64.47923
```

```
## 13 year 4        95% 62.24154
## 14 year 4       100% 55.43231
## 15 year 5         5% 78.95000
## 16 year 5        10% 77.57231
## 17 year 5        25% 75.00923
## 18 year 5        50% 72.51154
## 19 year 5        75% 70.70231
## 20 year 5        95% 68.69308
## 21 year 5       100% 60.56000
## 22 year 6         5% 78.67692
## 23 year 6        10% 76.73000
## 24 year 6        25% 73.97385
## 25 year 6        50% 71.03385
## 26 year 6        75% 68.99154
## 27 year 6        95% 66.77308
## 28 year 6       100% 57.92846
## 29 year 7         5% 74.65308
## 30 year 7        10% 72.84308
## 31 year 7        25% 69.24000
## 32 year 7        50% 66.28231
## 33 year 7        75% 64.25308
## 34 year 7        95% 62.05000
## 35 year 7       100% 55.65769
```

```
datas$perwatched = factor(datas$perwatched, levels=c("5%","10%","25%",
                                                     "50%","75%","95%","100%"))
```

Let's explain whats going on here. In order for GGplot to get the grouped bar plot to work, it needs to know effectively two x elements of each y. Here, the year part of this data frame is the group that the each observation is a part of, and the perwatched is identifying our percentage watched. The two dataframes are essentially the same thing, just the second one is very long and not easily readable. However, it enables us to plot a grouped bar plot that allows us to easily visualise the differences in year.

```
ggplot(datas, aes(fill=perwatched, y=values, x=year)) +
  geom_bar(position="dodge", stat="identity") + labs(x="Year",y="% of views") +
  labs(fill = "% watched")
```



This is a very useful plot for our comparison. We can clearly see that although there are slight differences in the percentages across the years, the trend is a very similar one. There is generally a consistent drop in viewers as the video goes on. The most significant drops certainly are from 95% watched to 100% watched, showing that a decent group of students will watch most of the video, but tune out before watching it all. This could possibly indicate that the videos could be slightly shortened. Youtube analytics suggest that in a given video it is normal to see a gradual decline in audience engagement as a video goes on, but sharper decreases like at the end of this video suggest that the something in the video is causing many viewers to stop watching. Source: https://creatoracademy.youtube.com/page/lesson/engagement-analytics?cid=analytics-series&hl=en#strategies-zippy-link-1

Overall views

The analysis that looks simply at the overall percentages for a year is all well and good, but unfortunately in this course not all videos or years were created equal. There are big differences between years in terms of number of students enrolled, and differences between videos in terms of the number of views on each video. Comparisons between videos will be tackled in the next part of this report, but for now lets stay on video engagement across the years.

The percentage analysis we've done thus far was good for investigating if there was a trend across years as it put all the years on a level playing field for comparison. But, to repeat myself, *not all years are created equal*. To see the overall picture in terms of total views, I needed to alter the calculation I was making for the percentage views. The benefit of my approach for the first part is that we can essentially reproduce the analysis we just did for the percentages, but simply alter our calculations to work out views rather than percentages of views.

This was done by making another helper function, almost identical to the avgper one used previously.

```
helper.avgview <- function(y)
{
  temp = rep(NULL,7)
  for (i in 9:15)
  {
    temp[i-8] = sum((y[i]/100)*y[4])/13
  }
  return(temp)
}
```

Avgview simply does the same calculation but using the total views we can work out how many views crossed each threshold per year. We now simply use this function to get our vectors of views.

```
y3a=helper.avgview(y3)
y4a=helper.avgview(y4)
y5a=helper.avgview(y5)
y6a=helper.avgview(y6)
y7a=helper.avgview(y7)
```

We can form the data frame again for readability. This uses the same code as before when forming the data frame so most of it has been committed for brevity.

```
(viddf = data.frame(year=3:7,"5%"=a5p,"10%"=a10p,"25%"=a25p,
                    "50%"=a50p,"75%"=a75p,"95%"=a95p,"100%"=a100p))
```

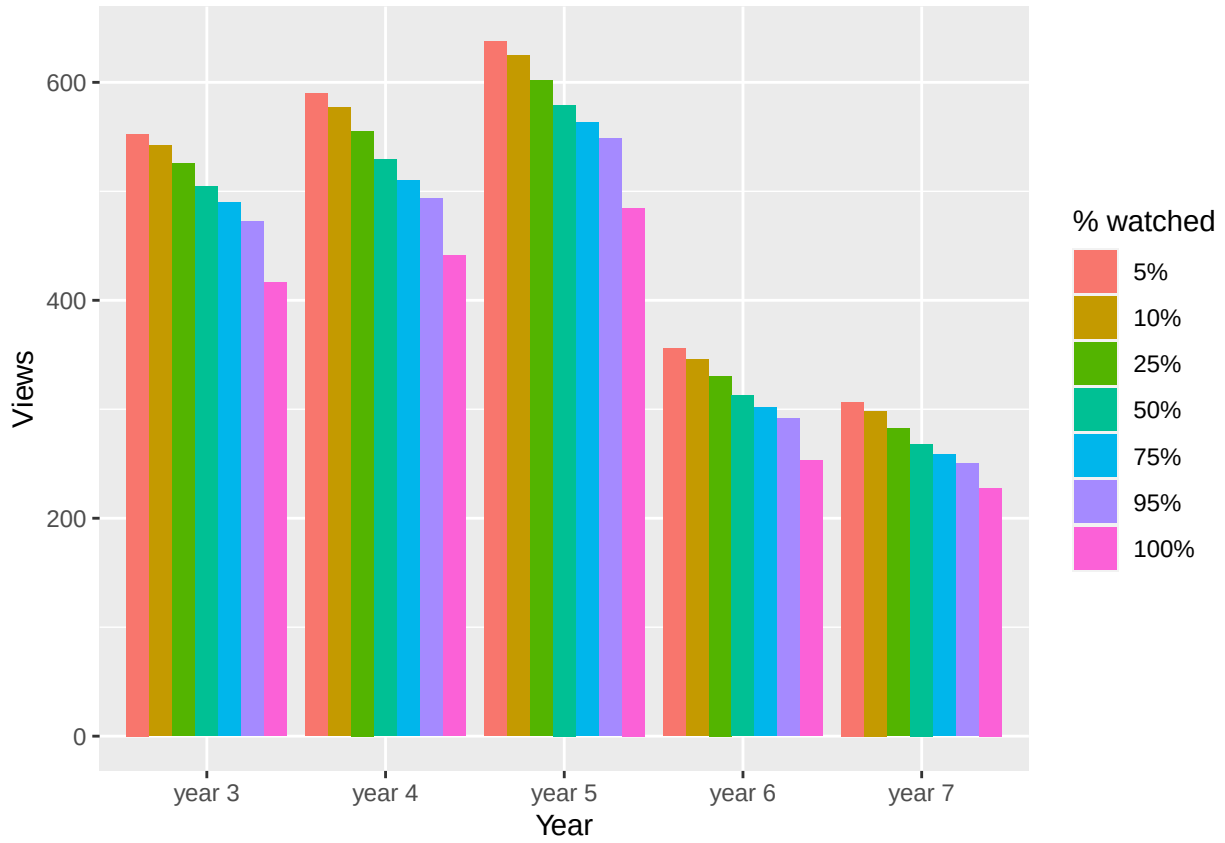
```
##   year      X5.     X10.     X25.     X50.     X75.     X95.    X100.
## 1    3 552.6984 541.9287 525.4658 504.5362 489.8368 472.4660 416.1434
## 2    4 589.6921 576.6123 555.3760 529.4631 509.9937 493.5414 440.9319
## 3    5 637.6308 624.5319 601.8566 579.3067 563.3838 548.3748 484.7548
## 4    6 355.6128 345.4624 330.3835 312.6920 301.3063 291.4648 252.8473
## 5    7 306.5378 298.5349 282.4682 267.9933 258.6968 250.3831 227.6115
```

We see a similar decreasing pattern as with the percentages, but here we see the actual effect on the views.

Now in order to plot we form another data frame for the grouped bar plot. Again, this is done with exactly the same code as before so is omitted.

Once this is done, we plot our data.

```
ggplot(datas, aes(fill=perwatched, y=values, x=year)) +
  geom_bar(position="dodge", stat="identity") + labs(x="Year",y="Views") +
  labs(fill = "% watched")
```



Here we can see a more overall picture of what is happening across the years. It is reflective of the situation of enrollments, which fall from 3544 in year 5 to 2342 in year 7. When you take this into account, the apparent decrease in overall views is probably expected. We see the same pattern as we saw with the overall percentages, the decreasing trend across years as students continue to lose interest the further they get into the video. As a tool to compare across the years this plot is less useful than the percentage of views plot, as the years have a different number of students enrolled. This will obviously have an effect on the number of views the videos get.

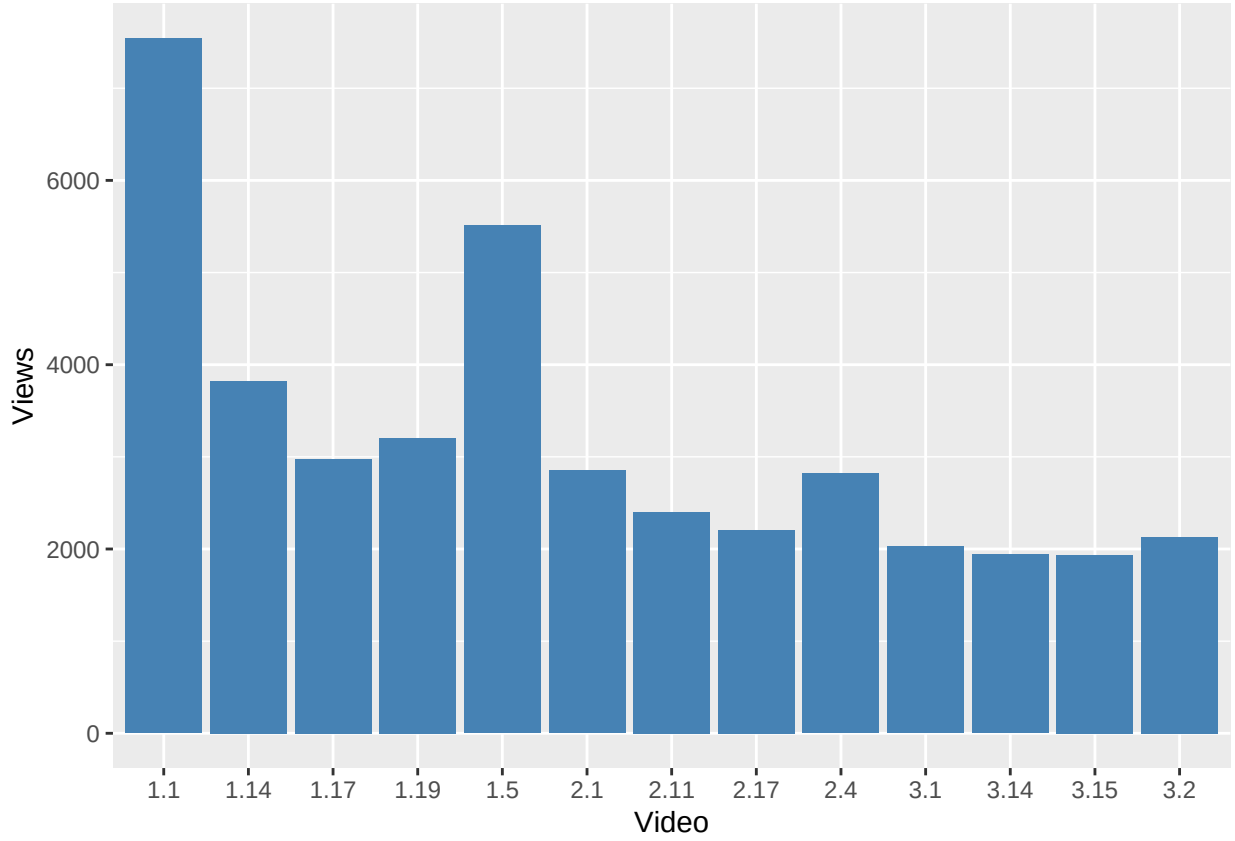
Video comparison

Comparing between the videos presents an interesting discussion about the use of the two plots I have looked at so far. When comparing across years, we are less interested in total views and more looking for trends, and this meant that the percentage graph was of more use to us to fairly compare the years. This is because here I am mainly interested in looking at the video stats, rather than enrollment numbers which define the total view graph. However I think its fair to say the opposite is true when we are looking to compare across videos. The percentage graph will be helpful to see if there is any particular trend with an individual video, but the total views graph should help to paint a picture of how much a video is watched.

First, lets take a step back and simply look at across all the years how many views each video got. This is a simpler undertaking, we just take the values and sum them from each year for each video, then set up our data frame and plot. With a simple bar plot, we don't need to bother with the different levels that we needed for the grouped bar plot. Essentially, this code is more concise and easier to understand.

```
views = rep(NULL,13)
for (j in 1:13)
```

```
{
  #for video with row index j, get the total views (column 4) and add them up
  views[j] = y3[j,4] + y4[j,4] + y5[j,4] + y6[j,4] + y7[j,4]
}
v = c("1.1","1.14","1.17","1.19","1.5","2.1","2.11",
      "2.17","2.4","3.1","3.14","3.15","3.2")
viewdf = data.frame(video=v, views=views)
ggplot(data=viewdf, aes(x=video, y=views)) +
  geom_bar(stat="identity", fill="steelblue") +
  labs(x="Video",y="Views")
```



Here we can see that as would be expected, the first video "Welcome" is far and away the most viewed video. There is a sharp decrease to the next video, with the only other video that gets close in terms of views being video 1.5 "Privacy online and offline". There is a clear tail off towards the end of the course, but it seems that it levels out towards the very end. This is a trend that is often evidenced in many modules of undergraduate maths lecture attendance to my own experience.

Lets break this down with our percentage watched thresholds. This time we'll first consider the total views first.

```
temp = rep(NULL,7)
val = c()
#for each video i
for (i in 1:13)
{
  #for each percentage watched threshold p
```

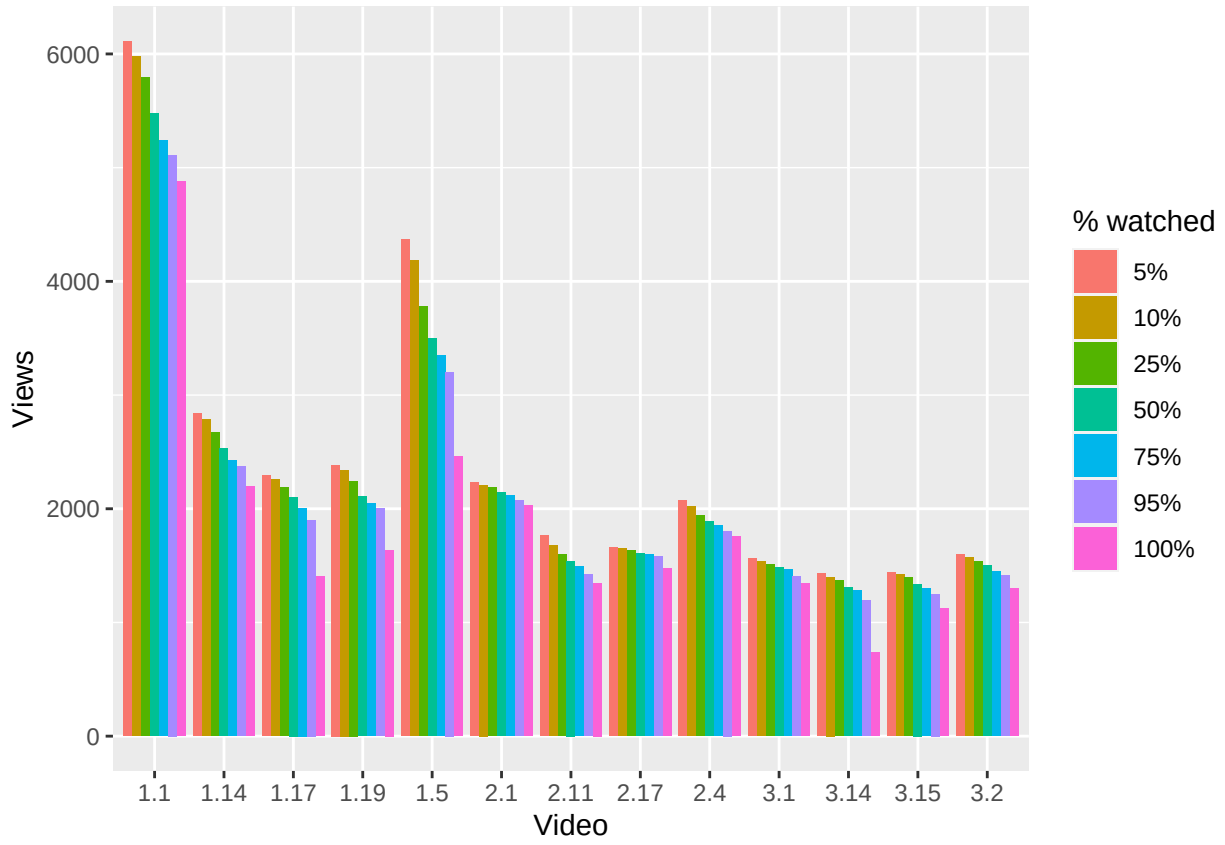
```
  for (p in 1:7)
  {
    #total views are calculated by ((percentaged watched to this point)/100)*total views
    #needs to be /100 as the 50% is stored just as 50 in the data
    temp[p] = (y3[i,p+8]/100)*y3[i,4] + (y4[i,p+8]/100)*y4[i,4] +
      (y5[i,p+8]/100)*y5[i,4] + (y6[i,p+8]/100)*y6[i,4] +
      (y7[i,p+8]/100)*y7[i,4]
  }
  val = append(val,temp)
}
```

In this chunk I've iterated through each video and within that each percentage watched threshold, calculating the number of views for each video that passed each threshold. In the earlier part of my analysis where I compared across years I presented this part in a readable data frame output. However, we have a lot of data to represent here so I've skipped straight to plotting it. Hence why the data is actually stored in one long vector rather than separate ones.

You'll hopefully recognise at least the structure of the following code by now. We are setting up a data frame for a grouped bar plot, building it very long so that the plot knows about the two x elements of every y. We group the percentage watched thresholds by video and plot.

```
videos = c(rep("1.1",7),rep("1.14",7),rep("1.17",7),rep("1.19",7),
           rep("1.5",7),rep("2.1",7),rep("2.11",7),rep("2.17",7),rep("2.4",7),
           rep("3.1",7),rep("3.14",7),rep("3.15",7),rep("3.2",7))
perwatched = rep(c("5%","10%","25%","50%","75%","95%","100%"),13)
values = val
datas = data.frame(videos,perwatched,values)
datas$perwatched = factor(datas$perwatched, levels=c("5%","10%","25%",
                                                     "50%","75%","95%","100%"))

ggplot(datas, aes(fill=perwatched, y=values, x=videos)) +
  geom_bar(position="dodge", stat="identity") +
  labs(x="Video",y="Views") +
  labs(fill = "% watched")
```



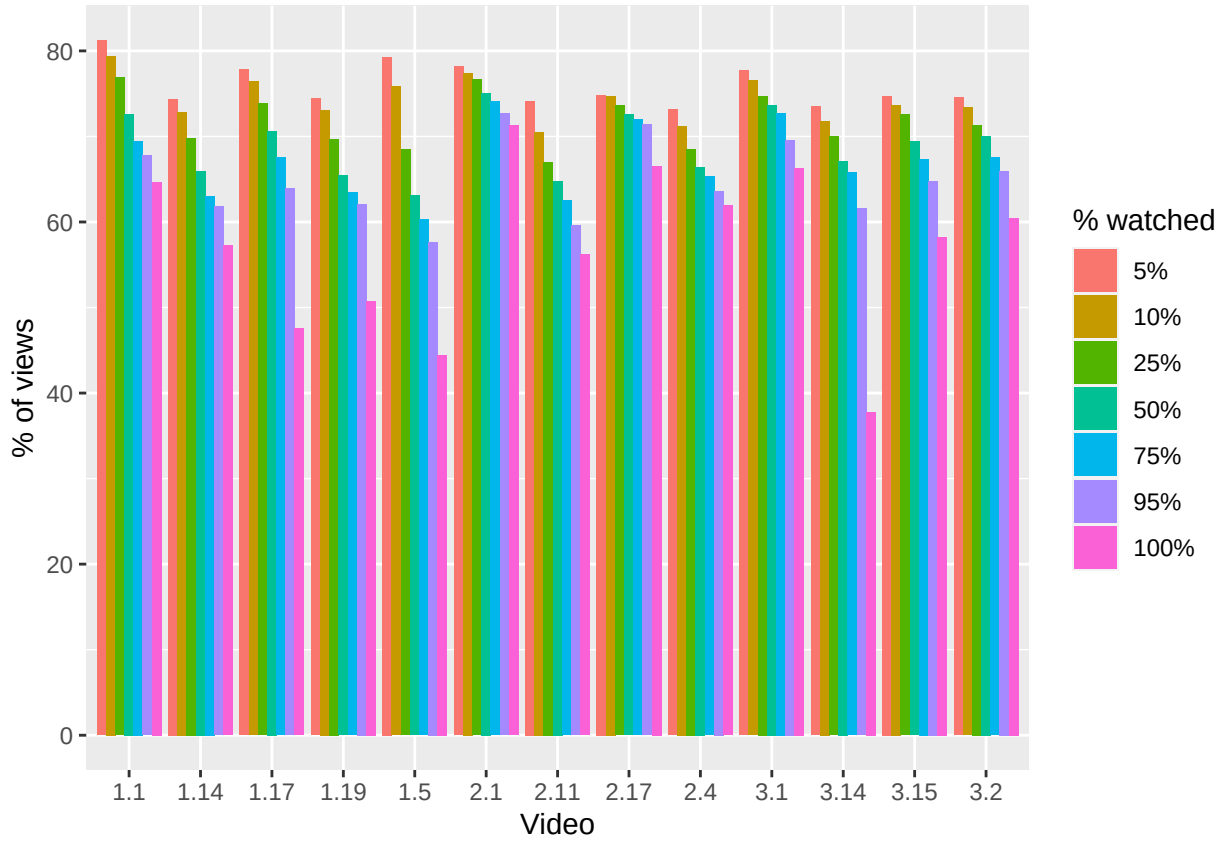
When we looked at the years comparison plot, it was dependent on enrollments which varied year by year. When comparing across videos however, we are not bound by this and we can see some interesting things. We see a familiar pattern across the board in terms of the general decrease in viewers watching further into the video. However what is most interesting to me in this plot is the degree to which this occurs in each video, with many exhibiting fairly distinct behaviour. Take for example 3.14. The videos in the same chapter all represent our typical gradual decline from 5% to 100%. 3.14 however displays an incredibly large drop from 95% to 100% watched, indicating something must have cause students to not watch until the end of that video. 1.17 has a similar behavior, whereas 2.17 and 2.1 is remarkably flat in terms of their shape, meaning they managed to retain many more of those who watched only 5% of the video to the end in comparison with others. We discussed how 1.5 is the video that is most popular when we discount the welcome 1.1 video, and the trend it displays is very interesting too. More than in most other videos, it loses viewers from the 5% to 50% threshold.

In order to gain a closer of understanding of the behaviour of the data around the videos across the years, we plot the percentage of views across videos.

```
temp = rep(NULL,7)
val = c()
for (i in 1:13)
{
  for (p in 1:7)
  {
    #here we can just take the percentages and average them
    temp[p] = (y3[i,p+8] + y4[i,p+8] + y5[i,p+8] + y6[i,p+8] + y7[i,p+8])/5
  }
  val = append(val,temp)
}
```

```
videos = c(rep("1.1",7),rep("1.14",7),rep("1.17",7),rep("1.19",7),
           rep("1.5",7),rep("2.1",7),rep("2.11",7),rep("2.17",7),rep("2.4",7),
           rep("3.1",7),rep("3.14",7),rep("3.15",7),rep("3.2",7))
perwatched = rep(c("5%","10%","25%","50%","75%","95%","100%"),13)
values = val
datas = data.frame(videos,perwatched,values)
datas$perwatched = factor(datas$perwatched, levels=c("5%","10%","25%",
                                                     "50%","75%","95%","100%"))

ggplot(datas, aes(fill=perwatched, y=values, x=videos)) +
  geom_bar(position="dodge", stat="identity") +
  labs(x="Video",y="% of views") +
  labs(fill = "% watched")
```



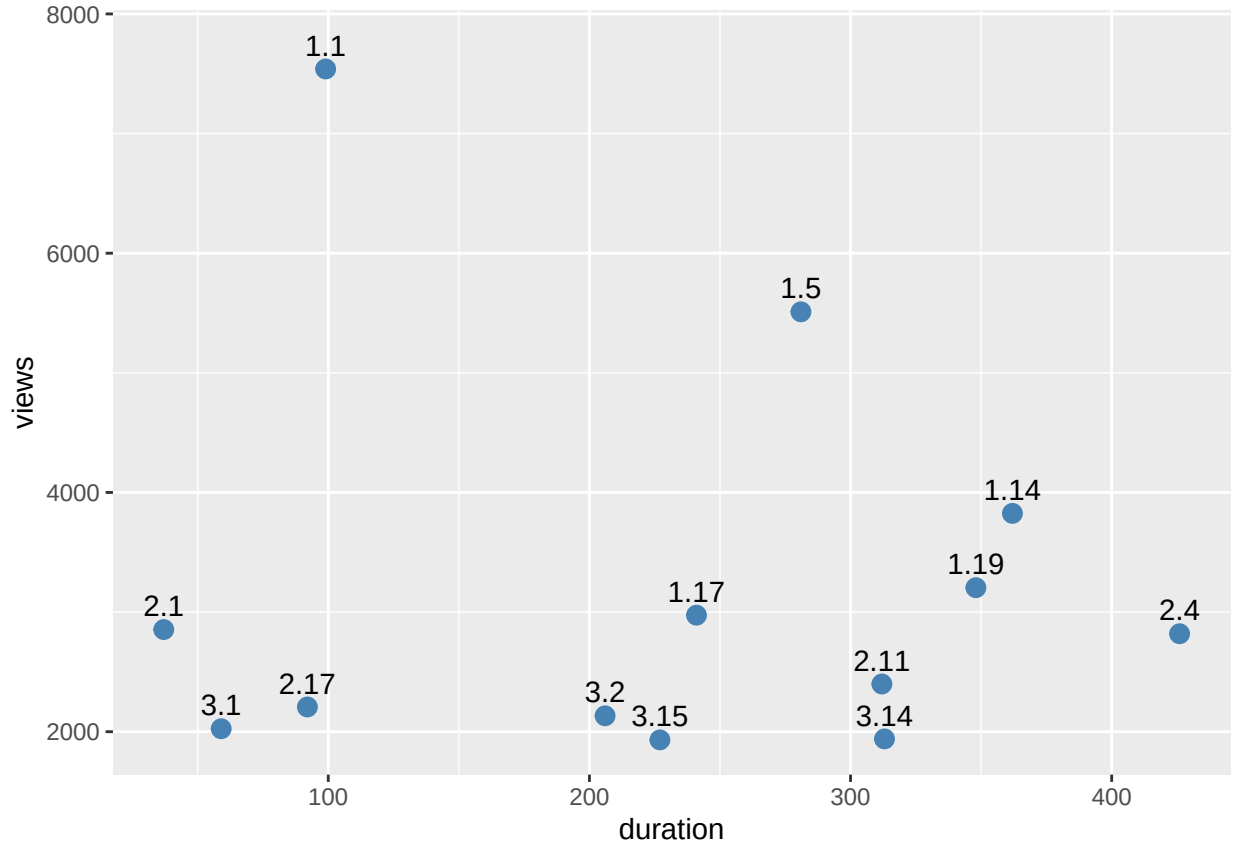
This accentuates the patterns we saw in the previous plot. 3.14 loses almost 20% of its audience that watched 95% of the video before the end. Considering most videos aren't losing that proportion of its audience across the whole video, this further reinforces the implication that something in that video is making users not watch the last 5% of the video.

Effect of duration

Its a reasonable question to ask if duration is having an effect on the "watchability" of a video, as intuitively, longer videos may require more engagement. To investigate this, we'll use a scatter plot.

```
duration = rep(NULL,13)
for (q in 1:13)
{
  #duration of videos is the same across years3-7 so we only need to grab from one year
  duration[q] = y3[q,3]
}
viewdf$duration = duration

ggplot(data=viewdf, aes(x=duration,y=views)) + geom_point(size=3,colour="steelblue") +
  geom_text(label=viewdf$video,nudge_y = 200)
```



This has been done as an addition to the data frame we used for the simple bar plot of videos against views. Here we can see that there's no real overall pattern to the plot, so we cant really say anything about duration having an effect on viewership of videos.

When we observe this in conjunction with the analysis we've conducted before on the videos, we can pull out some interesting ideas. The cluster in the bottom left of the plot represents the shortest videos that are getting a slightly less than average amount of viewers. When we go back and look at the pattern of their viewers engagement, we see that these are the videos with probably the least viewers lost throughout the video. This is insufficient evidence to make any sort of conclusion about shorter videos keeping their audience, as we haven't accounted for the possibility that less viewed videos have a more committed audience. We can see that the longest video, 2.4, has about average views and a normal pattern to viewership engagement. In fact most of the longer videos show the more linear decline in viewers as the videos go on.

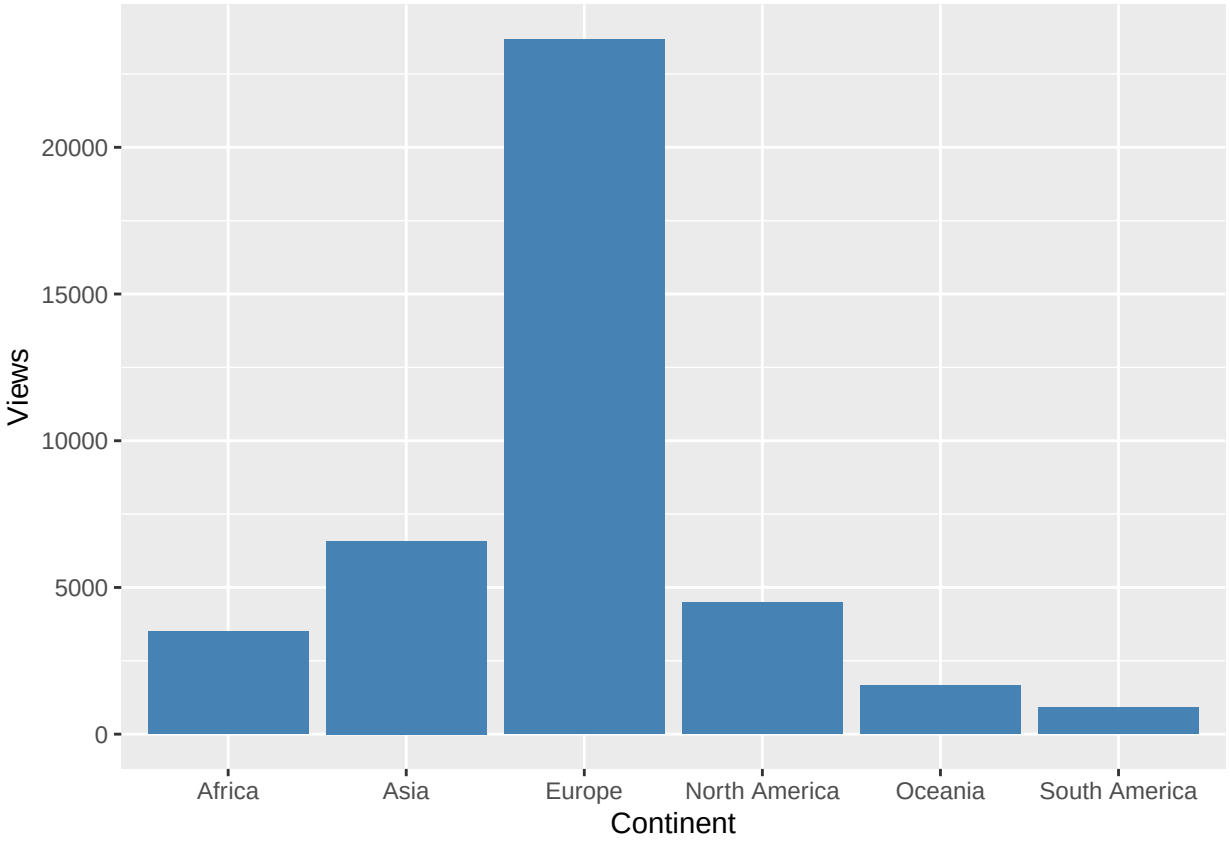
Region analysis

Here I'll investigate the habits of users from different continents when watching videos. First, we'll just have a look at which continents make up for the different shares of views. These figures are pulled from each of the year files and are calculated in a similar fashion as before.

```
#the continent %s are in columns 22 to 27
#performing the same calculations for total views as before
#just this time we're using the continent %s rather than viewed %s
europe = (y3[[22]]/100)*y3[[4]] + (y4[[22]]/100)*y4[[4]] + (y5[[22]]/100)*y5[[4]] +
  (y6[[22]]/100)*y6[[4]] + (y7[[22]]/100)*y7[[4]]
oce = (y3[[23]]/100)*y3[[4]] + (y4[[23]]/100)*y4[[4]] + (y5[[23]]/100)*y5[[4]] +
  (y6[[23]]/100)*y6[[4]] + (y7[[23]]/100)*y7[[4]]
asia = (y3[[24]]/100)*y3[[4]] + (y4[[24]]/100)*y4[[4]] + (y5[[24]]/100)*y5[[4]] +
  (y6[[24]]/100)*y6[[4]] + (y7[[24]]/100)*y7[[4]]
NorthAm = (y3[[25]]/100)*y3[[4]] + (y4[[25]]/100)*y4[[4]] + (y5[[25]]/100)*y5[[4]] +
  (y6[[25]]/100)*y6[[4]] + (y7[[25]]/100)*y7[[4]]
SouthAm = (y3[[26]]/100)*y3[[4]] + (y4[[26]]/100)*y4[[4]] + (y5[[26]]/100)*y5[[4]] +
  (y6[[26]]/100)*y6[[4]] + (y7[[26]]/100)*y7[[4]]
africa = (y3[[27]]/100)*y3[[4]] + (y4[[27]]/100)*y4[[4]] + (y5[[27]]/100)*y5[[4]] +
  (y6[[27]]/100)*y6[[4]] + (y7[[27]]/100)*y7[[4]]
```

These vectors calculate the share each region has of the total views of each video. This will be useful for the next plot we do after this. For now, we need to sum these to get our region bar plot.

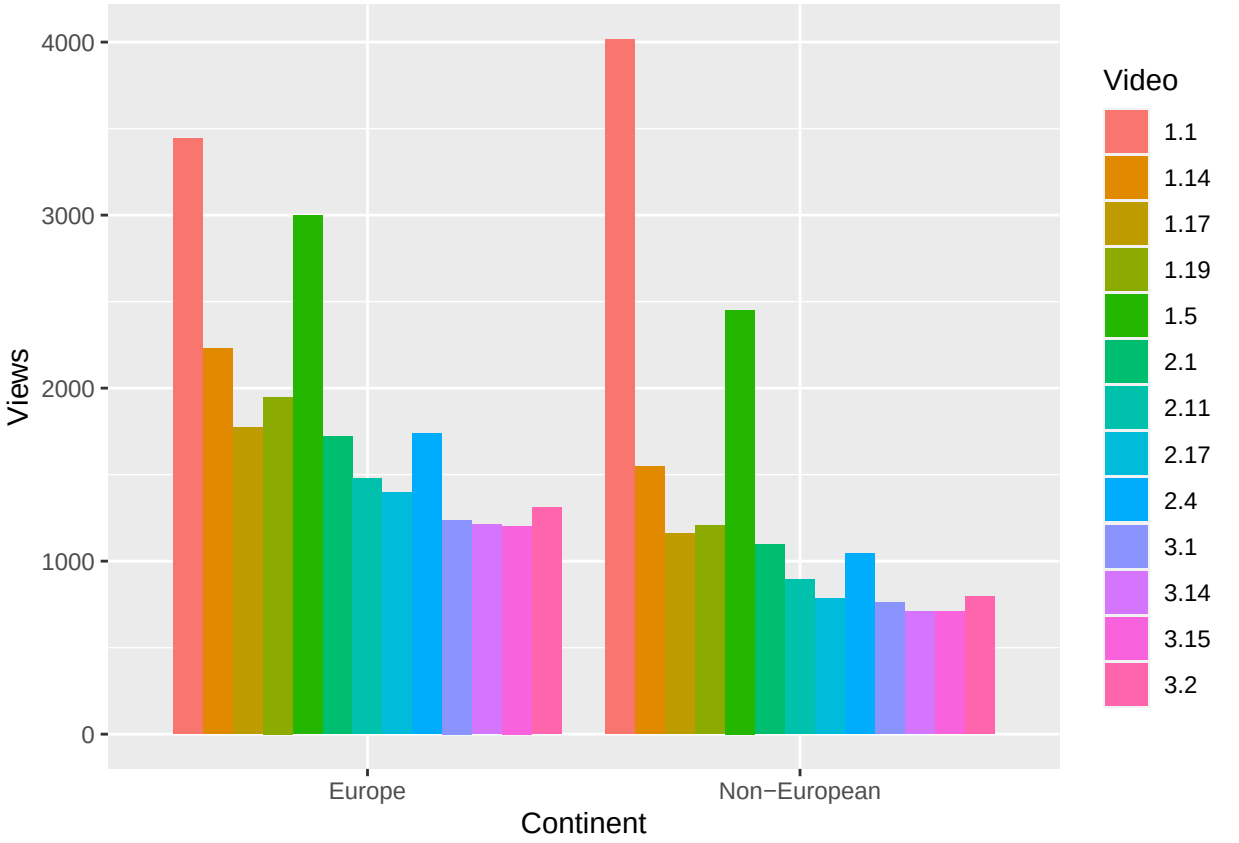
```
values = c(sum(europe),sum(oce),sum(asia),sum(NorthAm),sum(SouthAm),sum(africa))
continents = c("Europe", "Oceania", "Asia",
               "North America", "South America", "Africa")
cont = data.frame(continent = continents, views = values)
ggplot(data=cont, aes(x=continent, y=views)) +
  geom_bar(stat="identity", fill="steelblue") +
  labs(x="Continent",y="Views")
```



Europe clearly dominates the overall views since the course has been collecting video stats, as you would expect from briefly investigating the enrollment numbers. The next most popular continents of Asia, North America and Africa are also not surprising.

To further investigate, let's look into if there are differences between videos in terms of where the views are coming from. Here I am going to simplify our continents into "European" and "Non-European", just to simplify our comparisons. It turns out that these groups roughly represent about a 60:40 split of the total views (Europe represents 58% = 23680 views, Non European represents 42% = 17165). Therefore we total up all of the rest of the continents into one vector and create a grouped bar plot to represent how these two differing groups engage with each video.

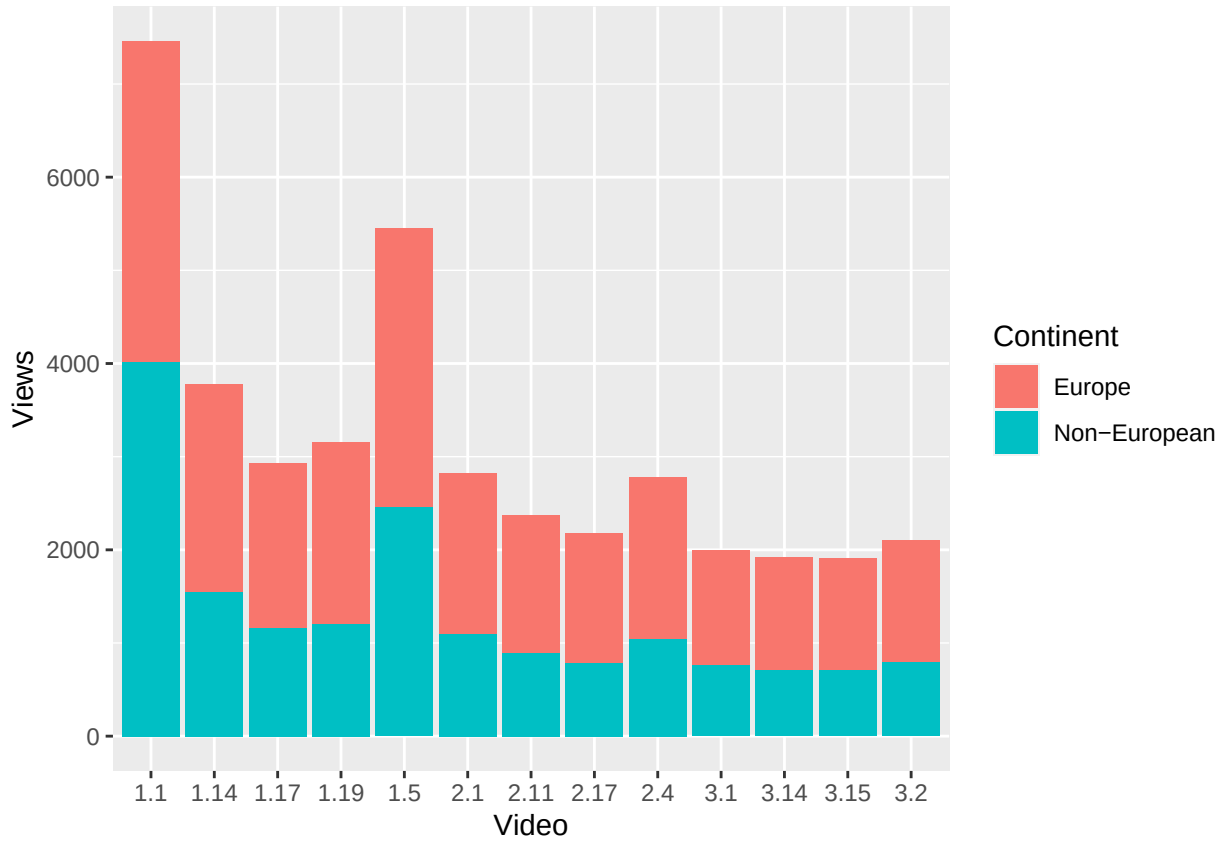
```
noneurope = oce+asia+NorthAm+SouthAm+africa
values = c(europe, noneurope)
video = rep(c("1.1","1.14","1.17","1.19","1.5","2.1","2.11",
              "2.17","2.4","3.1","3.14","3.15","3.2"),2)
european = c(rep("Europe",13),rep("Non-European",13))
dat = data.frame(video,european,values)
ggplot(dat, aes(fill=video, y=values, x=european)) +
  geom_bar(position="dodge", stat="identity") + labs(x="Continent",y="Views") +
  labs(fill = "Video")
```



This plot helps us to separate the trends of the European students and Non-European students. We can see a fairly similar pattern between the groups, with a few noticeable differences. Non-Europeans responsible for a larger proportion of the initial 1.1 welcome video views, and in general less for the other videos. European students comparatively are viewing the welcome video less, but in general engaging with the other videos more.

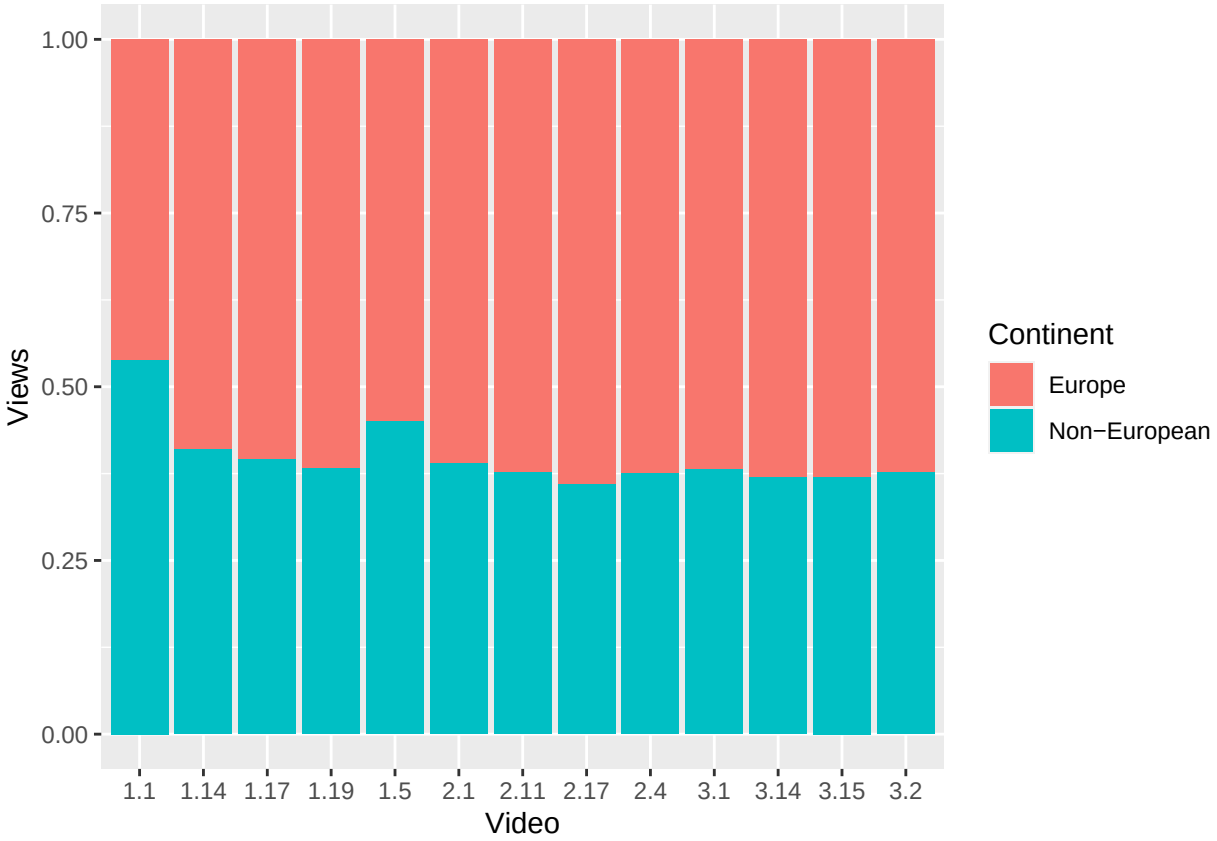
To further investigate this, we can look at this as a stacked bar plot of the video views.

```
ggplot(dat, aes(fill=european, y=values, x=video)) +
  geom_bar(position="stack", stat="identity") + labs(x="Video",y="Views") +
  labs(fill = "Continent")
```



This plot allows us to get the picture when focusing more on individual video views. We can also plot the stacked percentage plot to see a comparative effect.

```
ggplot(dat, aes(fill=european, y=values, x=video)) +
  geom_bar(position="fill", stat="identity") + labs(x="Video",y="Views") +
  labs(fill = "Continent")
```



This shows us without doubt that the Europeans dominate the views of most videos, but in a video like 1.5, Non Europeans make up 45% of views which is slightly more than their overall share of video views. What we can draw from this is that Non-Europeans are much more likely to watch the 1.1 welcome video, but from there the mix is fairly balanced in terms of views compared against number of enrolements.

Overall we can see from the regional analysis that video views are dominated by the European students due to them being the most represented group in the enrolements. Certain videos are more viewed by Non-European students, but generally videos are engaged with proportionally across the board.