KickStarterProject

(Eliza) Zijin Huang, Tian Xia

4/17/2019

1.

**Data Acquisition and Integration

```
data_source_1 <- read.csv("Kickstarter001.csv", header = TRUE, sep = ",")
data_source_2 <- read.csv("Kickstarter002.csv", header = TRUE, sep = ",")

raw_data <- rbind(data_source_1, data_source_2)
```

2.

**Data Cleaning There are 3784 data totally and there are 3680 projects are completed.

```
live_data <- raw_data %>% filter(raw_data$state == "live")
```

To clean the "catagory" column

```
raw_data$category <- raw_data$category %>%
  str_extract("slug\":\".+\",\"") %>%
  str_replace_all("\",\"", "") %>%
  str_replace_all("slug\":\"", "") %>%
  str_replace("/.+", "")
```

To clean the "location" column

```
raw_data$location <- raw_data$location %>%
  str_extract("name\":\".+\",\"") %>%
  str_replace("\",\".+", "") %>%
  str_replace_all("name\":\"", "")
```

To get rid of creator,photo, slug, urls column

```
raw_data$creator <- NULL
raw_data$photo <- NULL
raw_data$slug <- NULL
raw_data$urls <- NULL
```

To add a preparation_duration column

```
raw_data$preparation_duration <- raw_data$launched_at - raw_data$created_at
raw_data$preparation_duration_r <- seconds_to_period(raw_data$preparation_duration)
```

To add a launch_duration column

```
raw_data$launch_duration <- raw_data$deadline - raw_data$launched_at
raw_data$launch_duration_r <- seconds_to_period(raw_data$launch_duration)
raw_data$launch_duration_r <- day(raw_data$launch_duration_r)
```

To convert epoch seconds to readable time

```
raw_data$created_at_readable <- anytime(raw_data$created_at)
raw_data$deadline_readable <- anytime(raw_data$deadline)
raw_data$launched_at_readable <- anytime(raw_data$launched_at)
raw_data$preparation_duration_r <- NULL
```

Transfer raw data into a new variable

```
clean_data <- raw_data
write.csv(clean_data, "./data/data.csv")
head(clean_data)
```

```
##   backers_count
## 1            30
## 2             0
## 3           102
## 4            22
## 5             2
## 6             7
##
## 1 Experience tea and coffee as it should be in our handmade, fine bone china mugs. Made exclusively
## 2    Playing Roles Outside of Basic Education (P.R.O.B.E)\nThe magazine that highlights extracurricul
## 3                                                                                          A pilot for
## 4 A film about suicide. The struggles of our modern world taking people to their limit and how commo
## 5               Fusing the technical qualities and accuracy of photography with a digital process to
## 6                                          A digital, interactive magazine and online community fo
##       category converted_pledged_amount country created_at currency
## 1       crafts                     1547      GB 1515610761      GBP
## 2   publishing                        0      US 1426362805      USD
## 3 film & video                     8101      US 1525106061      USD
## 4 film & video                     1566      GB 1519854040      GBP
## 5          art                       11      US 1407346285      USD
## 6   publishing                      826      US 1411150798      USD
##   currency_symbol currency_trailing_code current_currency   deadline
## 1               £                 false              USD 1521190409
## 2               $                  true              USD 1429112946
## 3               $                  true              USD 1531713540
## 4               £                 false              USD 1522443600
## 5               $                  true              USD 1410484909
## 6               $                  true              USD 1414008752
##   disable_communication friends  fx_rate  goal         id is_backing
## 1                 false         1.308394  1000 1361161119
## 2                 false         1.000000  5000  746509287
## 3                 false         1.000000  6000 1402909261
## 4                 false         1.308394   400  311541751
## 5                 false         1.000000 11000  466957735
## 6                 false         1.000000  2000 1471254290
```

```
##   is_starrable is_starred launched_at       location
## 1        false          1517306009         London
## 2        false          1426520946       Columbus
## 3        false          1529070876 St. Petersburg
## 4        false          1519937886         Dorset
## 5        false          1407892909      Ypsilanti
## 6        false          1411416752  San Francisco
##                                          name permissions pledged
## 1 Fine Bone China Ceramic Mugs, Made in England             1111.0
## 2                           P.R.O.B.E. Magazine                0.0
## 3                     'Merican Wasteland Pilot             8101.0
## 4                         Cliff - Feature Film             1116.5
## 5                            Photo to Artwork                11.0
## 6   Lilah Magazine 1st issue launching Dec 2014              826.0
##
## 1
## 2
## 3
## 4 {"id":3322408,"project_id":3322408,"state":"active","state_changed_at":1523464237,"name":"CLIFF - F
## 5
## 6
##                                                              source_url
## 1                 https://www.kickstarter.com/discover/categories/crafts
## 2 https://www.kickstarter.com/discover/categories/publishing/periodicals
## 3       https://www.kickstarter.com/discover/categories/film%20&%20video
## 4       https://www.kickstarter.com/discover/categories/film%20&%20video
## 5      https://www.kickstarter.com/discover/categories/art/digital%20art
## 6 https://www.kickstarter.com/discover/categories/publishing/periodicals
##   spotlight staff_pick      state state_changed_at static_usd_rate
## 1      true      false successful       1521190409        1.413819
## 2     false      false     failed       1429112947        1.000000
## 3      true      false successful       1531713540        1.000000
## 4      true      false successful       1522443600        1.390235
## 5     false      false     failed       1410484909        1.000000
## 6     false      false     failed       1414008752        1.000000
##   usd_pledged      usd_type preparation_duration launch_duration
## 1    1570.753 international              1695248         3884400
## 2       0.000 international               158141         2592000
## 3    8101.000 international              3964815         2642664
## 4    1552.197 international                83846         2505714
## 5      11.000      domestic               546624         2592000
## 6     826.000 international               265954         2592000
##   launch_duration_r created_at_readable   deadline_readable
## 1                44 2018-01-10 13:59:21 2018-03-16 04:53:29
## 2                30 2015-03-14 15:53:25 2015-04-15 11:49:06
## 3                30 2018-04-30 12:34:21 2018-07-15 23:59:00
## 4                29 2018-02-28 16:40:40 2018-03-30 17:00:00
## 5                30 2014-08-06 13:31:25 2014-09-11 21:21:49
## 6                30 2014-09-19 14:19:58 2014-10-22 16:12:32
##   launched_at_readable
## 1  2018-01-30 04:53:29
## 2  2015-03-16 11:49:06
## 3  2018-06-15 09:54:36
## 4  2018-03-01 15:58:06
```

```
## 5  2014-08-12 21:21:49
## 6  2014-09-22 16:12:32
```

3.

**Data exploration and Visualization

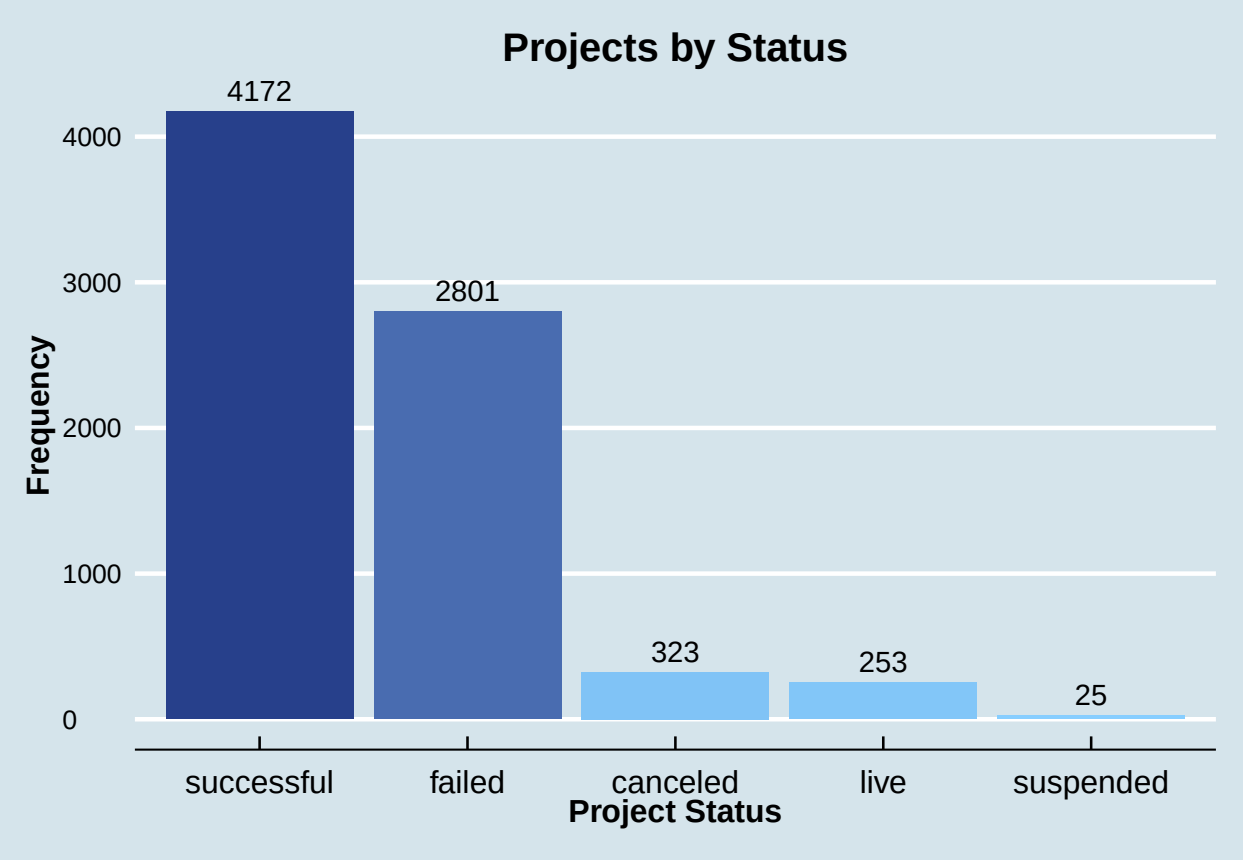
3.1 Summarise the number of projects for each status

```
status_prjects <- clean_data %>%
  group_by(clean_data$state) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
head(status_prjects)
```

```
## # A tibble: 5 x 2
##   `clean_data$state` count
##   <fct>              <int>
## 1 successful          4172
## 2 failed              2801
## 3 canceled             323
## 4 live                 253
## 5 suspended             25
```

Plot the number of projects for each status

```
ggplot(status_prjects, aes(reorder(status_prjects$`clean_data$state`, -count), count, fill=count)) + ge
  ggtitle("Projects by Status") + xlab("Project Status") + ylab("Frequency") +
  geom_text(aes(label=count), vjust=-0.5) + theme_economist() +
  theme(plot.title=element_text(hjust=0.5), axis.title=element_text(size=12, face="bold"),
        axis.text.x=element_text(size=12), legend.position="null") +
  scale_fill_gradient(low="skyblue1", high="royalblue4")
```



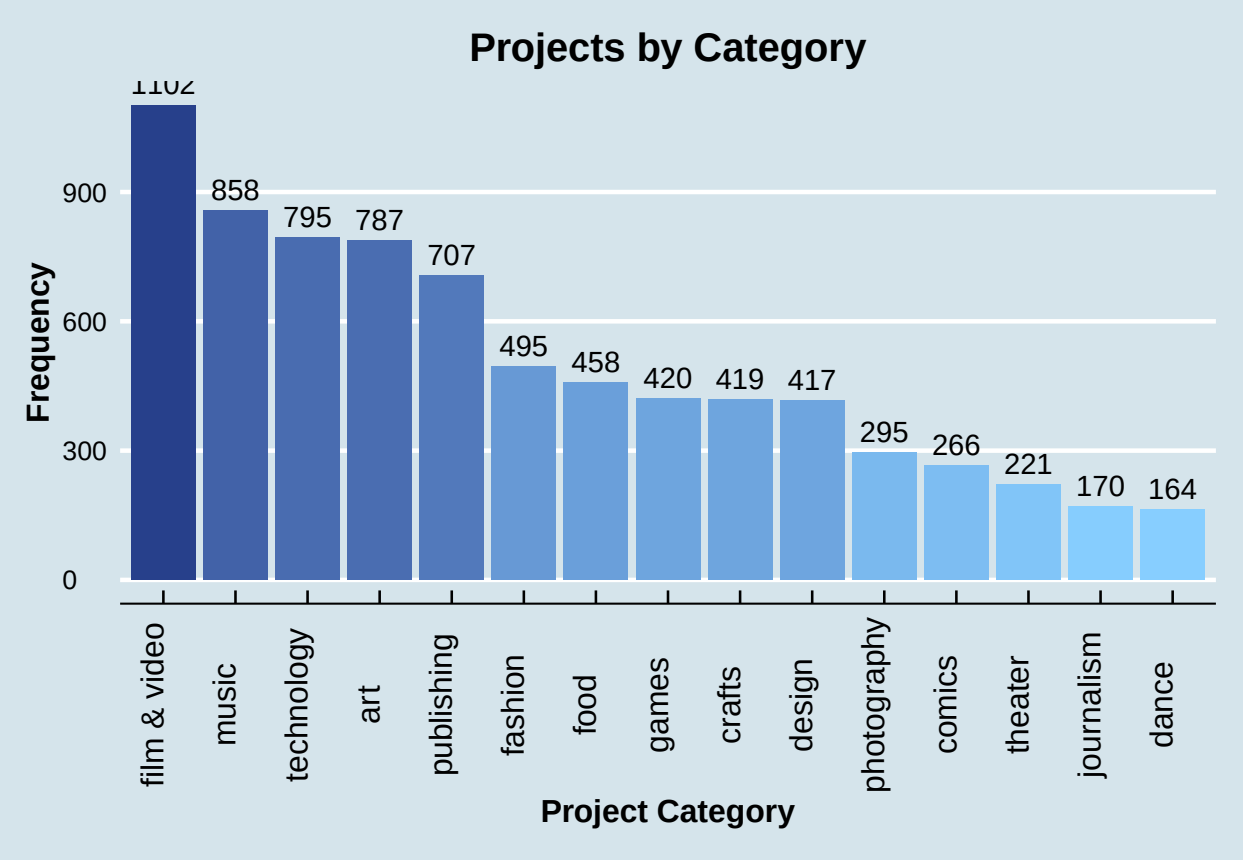
3.2 Summarise the number of projects for each catagory

```
catagory_projects <- clean_data %>%
  group_by(clean_data$category) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
head(catagory_projects)
```

```
## # A tibble: 6 x 2
##   `clean_data$category` count
##   <chr>                <int>
## 1 film & video          1102
## 2 music                  858
## 3 technology             795
## 4 art                    787
## 5 publishing             707
## 6 fashion                495
```

Plot the popularity of each category, which is dertermined by the number of projects

```
ggplot(catagory_projects, aes(reorder(catagory_projects$`clean_data$category`, -count), count, fill=cou
  ggtitle("Projects by Category") + xlab("Project Category") + ylab("Frequency") +
  geom_text(aes(label=count), vjust=-0.5) + theme_economist() +
  theme(plot.title=element_text(hjust=0.5), axis.title=element_text(size=12, face="bold"),
        axis.text.x=element_text(size=12, angle=90), legend.position="null") +
  scale_fill_gradient(low="skyblue1", high="royalblue4")
```



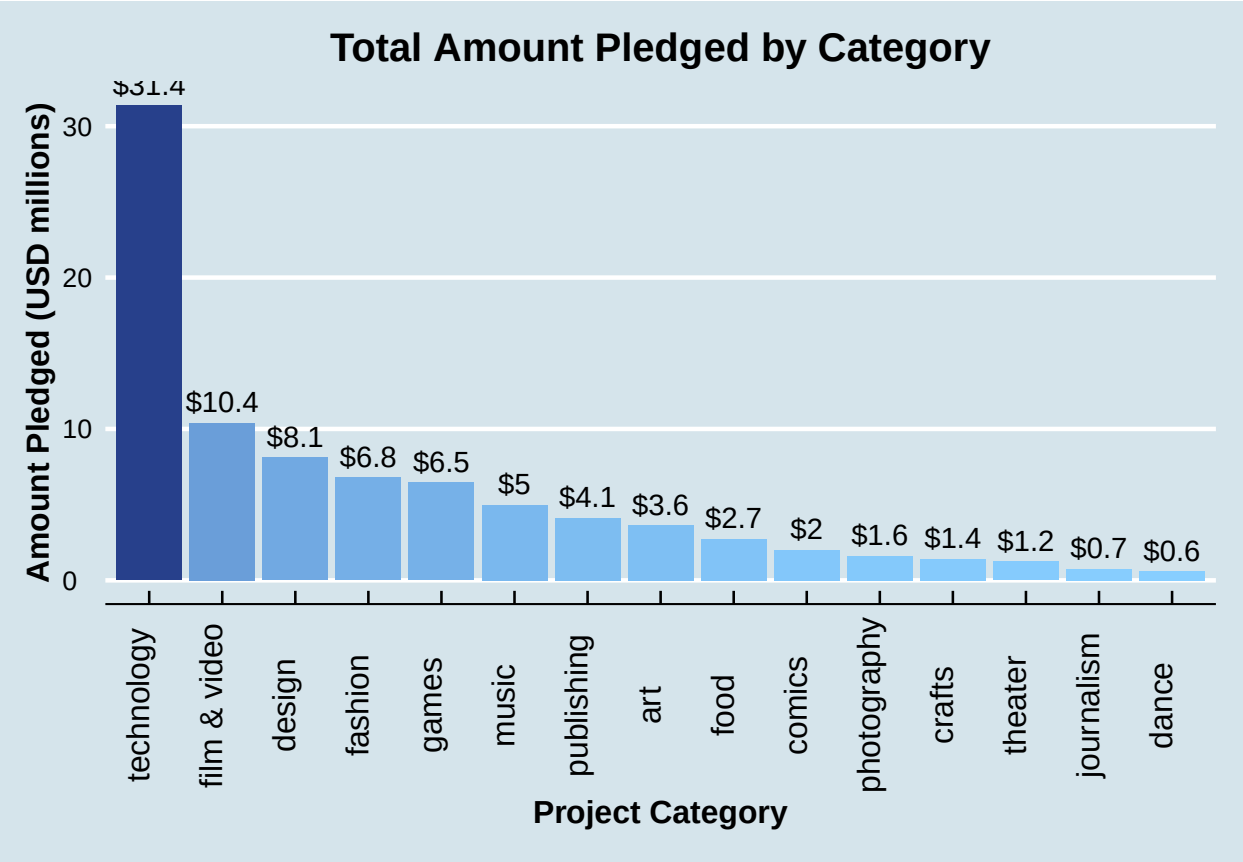
3.3 What types of projects are being funded?

```
pledged_category <- clean_data %>%
  group_by(clean_data$category) %>%
  summarise(total = sum(usd_pledged)) %>%
  arrange(desc(total))
head(pledged_category)
```

```
## # A tibble: 6 x 2
##   `clean_data$category`     total
##   <chr>                     <dbl>
## 1 technology            31373736.
## 2 film & video          10398557.
## 3 design                 8108525.
## 4 fashion                6797765.
## 5 games                  6465490.
## 6 music                  4988363.
```

Plot the amount pledged by each category

```
ggplot(pledged_category, aes(reorder(pledged_category$`clean_data$category`, -total), total/1000000, fi
  ggtitle("Total Amount Pledged by Category") + xlab("Project Category") +
  ylab("Amount Pledged (USD millions)") +
  geom_text(aes(label=paste0("$", round(total/1000000,1))), vjust=-0.5) + theme_economist() +
  theme(plot.title=element_text(hjust=0.5), axis.title=element_text(size=12, face="bold"),
        axis.text.x=element_text(size=12, angle=90), legend.position="null") +
  scale_fill_gradient(low="skyblue1", high="royalblue4")
```



3.4 How much is pledged per backer for each category?

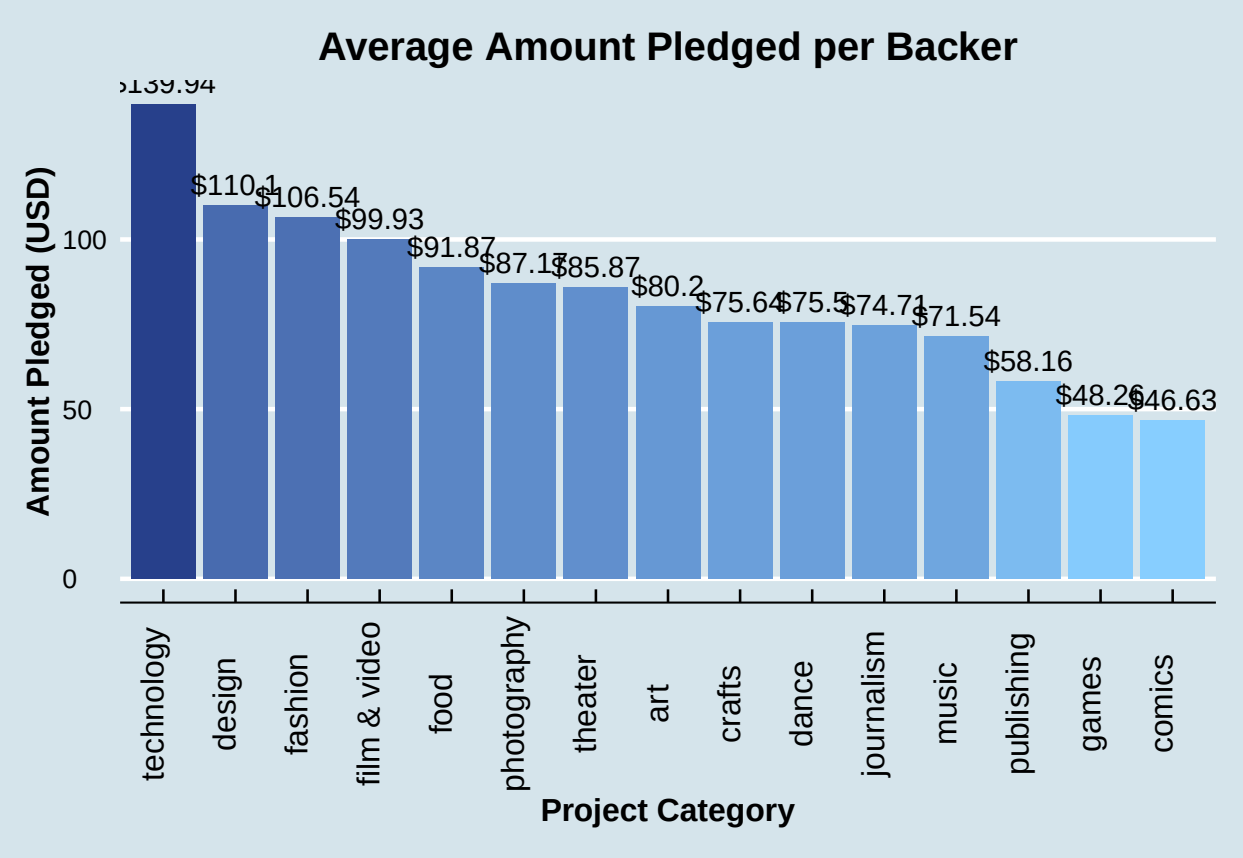
```
pledged_avg_category <- clean_data %>%
  group_by(clean_data$category) %>%
  summarise(pledged = sum(usd_pledged), backers=sum(backers_count)) %>%
  mutate(avg = pledged/backers) %>%
  arrange(desc(avg))
head(pledged_avg_category)
```

```
## # A tibble: 6 x 4
##   `clean_data$category`   pledged backers   avg
##   <chr>                     <dbl>   <int> <dbl>
## 1 technology            31373736.  224188 140.
## 2 design                 8108525.   73647 110.
## 3 fashion                6797765.   63802 107.
## 4 film & video          10398557.  104058  99.9
## 5 food                   2745202.   29880  91.9
## 6 photography            1620027.   18585  87.2
```

Plot the amount pledged per backer for each category

```
ggplot(pledged_avg_category, aes(reorder(pledged_avg_category$`clean_data$category`, -avg), avg, fill=av
  ggtitle("Average Amount Pledged per Backer") + xlab("Project Category") +
  ylab("Amount Pledged (USD)") +
```

```
  geom_text(aes(label=paste0("$", round(avg,2))), vjust=-0.5) + theme_economist() +
  theme(plot.title=element_text(hjust=0.5), axis.title=element_text(size=12, face="bold"),
        axis.text.x=element_text(size=12, angle=90), legend.position="null") +
  scale_fill_gradient(low="skyblue1", high="royalblue4")
```



3.5 Get the 10 highest goal successful projects

```
top_ten_success <- clean_data[clean_data$state == "successful",] %>%
  select("category", "goal", "state") %>%
  arrange(desc(goal))
head(top_ten_success)
```

```
##       category    goal      state
## 1   technology 1500000 successful
## 2   technology  800000 successful
## 3   technology  800000 successful
## 4 photography  700000 successful
## 5   technology  500000 successful
## 6       design  500000 successful
```

3.6 Get the average project goal

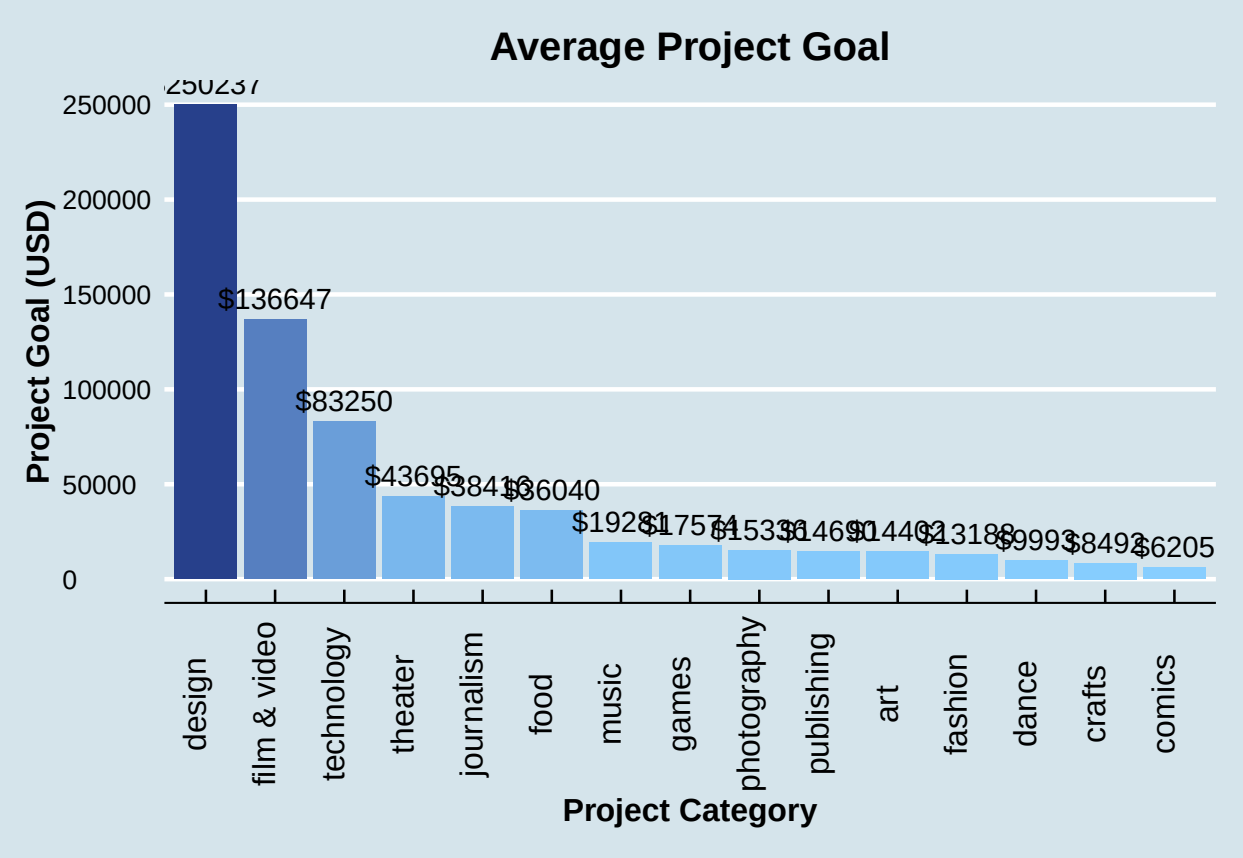
```
goal_avg <- clean_data %>%
  group_by(category) %>%
  summarise(goals = sum(goal), projects = n()) %>%
```

```
  mutate(avg = goals/projects) %>%
  arrange(desc(avg))
head(goal_avg)
```

```
## # A tibble: 6 x 4
##   category          goals projects     avg
##   <chr>             <dbl>    <int>   <dbl>
## 1 design       104348985.       417 250237.
## 2 film & video 150585489.      1102 136647.
## 3 technology    66183820.       795  83250.
## 4 theater        9656489.       221  43695.
## 5 journalism     6530680.       170  38416.
## 6 food          16506111.       458  36040.
```

Plot the average project goal.

```
ggplot(goal_avg, aes(reorder(goal_avg$category, -avg), avg, fill=avg)) + geom_bar(stat="identity") +
  ggtitle("Average Project Goal") + xlab("Project Category") + ylab("Project Goal (USD)") +
  geom_text(aes(label=paste0("$", round(avg,0))), vjust=-0.5) + theme_economist() +
  theme(plot.title=element_text(hjust=0.5), axis.title=element_text(size=12, face="bold"),
        axis.text.x=element_text(size=12, angle=90), legend.position="null") +
  scale_fill_gradient(low="skyblue1", high="royalblue4")
```



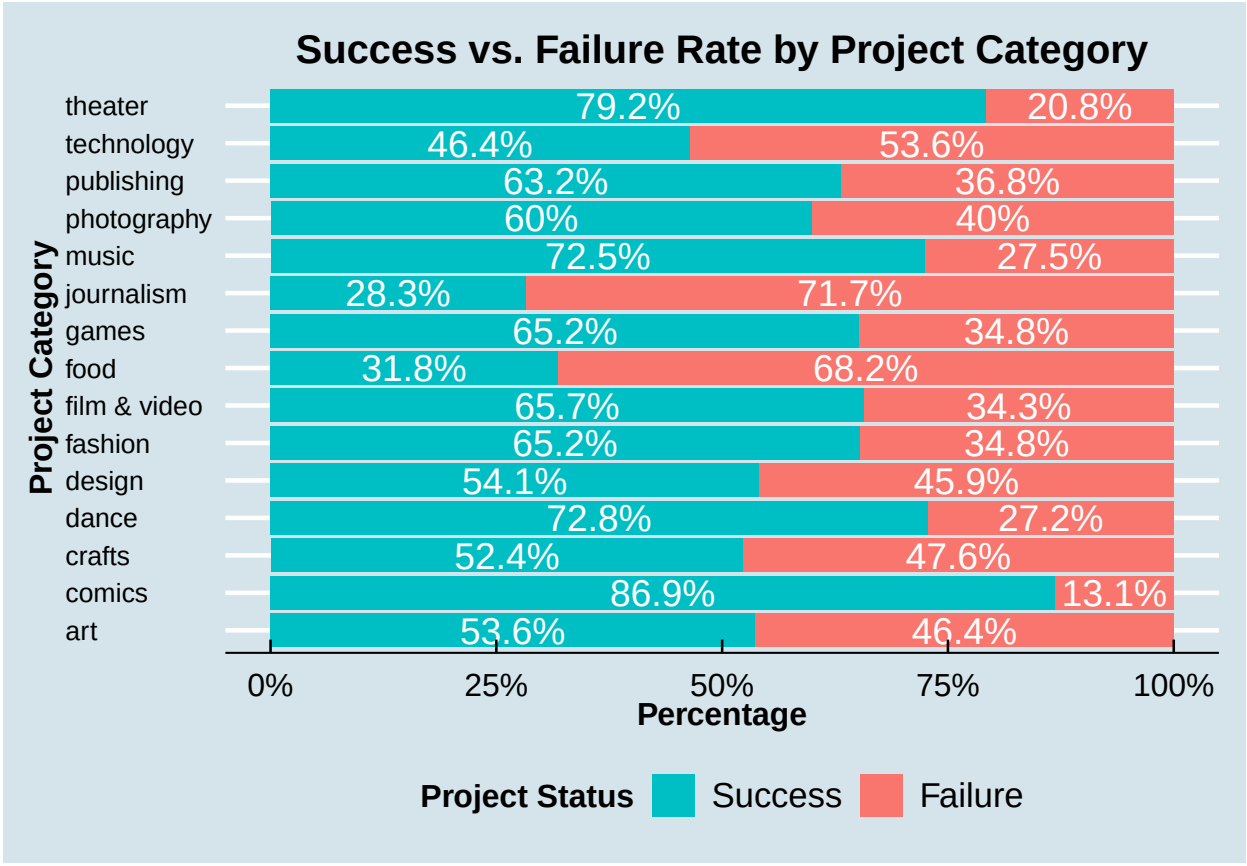
3.7 percentage for projects in each category

```r
perc_projects <- clean_data %>%
  filter(state %in% c("successful", "failed")) %>%
  group_by(category, state) %>%
  summarize(count=n()) %>%
  mutate(pct=count/sum(count)) %>%
  arrange(desc(state), pct)
head(perc_projects)
```

```
## # A tibble: 6 x 4
## # Groups:   category [6]
##   category   state      count   pct
##   <chr>      <fct>      <int> <dbl>
## 1 journalism successful    43 0.283
## 2 food       successful   132 0.318
## 3 technology successful   344 0.464
## 4 crafts     successful   199 0.524
## 5 art        successful   390 0.536
## 6 design     successful   198 0.541
```

Plot the percentage for each category

```r
ggplot(perc_projects, aes(perc_projects$category, pct, fill=state)) + geom_bar(stat="identity") +
  ggtitle("Success vs. Failure Rate by Project Category") +
  xlab("Project Category") + ylab("Percentage") + scale_y_continuous(labels=scales::percent) +
  scale_fill_discrete(name="Project Status", breaks=c("successful", "failed"),
                      labels=c("Success", "Failure")) +
  geom_text(aes(label=paste0(round(pct*100,1),"%")), position=position_stack(vjust=0.5),
            colour="white", size=5) + theme_economist() +
  theme(plot.title=element_text(hjust=0.5), axis.title=element_text(size=12, face="bold"),
        axis.text.x=element_text(size=12), legend.position="bottom",
        legend.title=element_text(size=12, face="bold")) + coord_flip()
```



3.8 Does project length affect success rate?

```
perc_length <- clean_data %>%
  filter(state %in% c("successful", "failed"), launch_duration_r < 61) %>%
  group_by(launch_duration_r, state) %>%
  summarize(count=n()) %>%
  mutate(pct=count/sum(count))
head(perc_length)
```

```
## # A tibble: 6 x 4
## # Groups:   launch_duration_r [4]
##   launch_duration_r state      count   pct
##               <dbl> <fct>      <int> <dbl>
## 1                 1 failed         1 0.5
## 2                 1 successful     1 0.5
## 3                 2 successful     1 1
## 4                 3 failed         6 0.667
## 5                 3 successful     3 0.333
## 6                 4 failed         2 1
```

```
ggplot(perc_length[perc_length$state=="successful",], aes(launch_duration_r, pct)) +
  geom_point(colour="royalblue4", size=2.5) + ggtitle("Success Rate vs. Project Length") +
  xlab("Project Length (Days)") + ylab("Success Rate (%)") +
  scale_x_continuous(breaks=c(0,10,20,30,40,50,60)) + geom_vline(xintercept=30, colour="red") +
  theme_economist() +
  theme(plot.title=element_text(hjust=0.5), axis.title=element_text(size=12, face="bold"))
```

3.9 Where it is coming form?

```
countries_freq <- clean_data %>%
  group_by(country) %>%
  summarize(count=n())

countries.match <- joinCountryData2Map(countries_freq, joinCode="ISO2", nameJoinColumn="country")
```

```
## 22 codes from your data successfully matched countries in the map
## 0 codes from your data failed to match with a country code in the map
## 220 codes from the map weren't represented in your data
```

```
mapCountryData(countries.match, nameColumnToPlot="count",
               mapTitle="Number of Projects by Country", catMethod="logFixedWidth",
               colourPalette="heat")
```

Number of Projects by Country