Data Analysis for Job Positions

Dec 06 2020

The aim of this project is to explore jobs and try to find the common vocabulary used in the job description. Job type, company, title, location and description were analyzed below. For descriptions, texting manipulation was used to clean and dig the insight. I used two data sets, which the larger one is from Kaggle and another one is from Github.The data set which is from Github Job and, specifically jobs for software developers , data science. Linkis are https://jobs.github.com/api. The larger data set whcih is from Kaggle and jobs are from Australia.https://www.kaggle.com/vipulanand/jobs-data. Due to there are not many jobs posted on the Github(around 150), then I downloaded a larger data set from Kaggle to show the reproducibility of my code. I posted the larger data set in the repo as well. The code which is used to analysis are exactly same except the name of variable and attributes of job. The shiny App show the content from Github.

0. Import Packages

1. Data Ingest

```
p1 = GET("https://jobs.github.com/positions.json?page=1")
p2 = GET("https://jobs.github.com/positions.json?page=2")
p3 = GET("https://jobs.github.com/positions.json?page=3")

p1 <- fromJSON(rawToChar(p1$content))
p2 <- fromJSON(rawToChar(p2$content))
p3 <- fromJSON(rawToChar(p3$content))

data <- rbind(p1, p2, p3)
names(data)
```

```
##  [1] "id"           "type"         "url"          "created_at"   "company"
##  [6] "company_url"  "location"     "title"        "description"  "how_to_apply"
## [11] "company_logo"
```

```
glimpse(data)
```

```
## Rows: 150
## Columns: 11
## $ id           <chr> "46ab84b4-12dc-4a95-ab08-422428554dfc", "bead9090-4ebb-4…
## $ type         <chr> "Full Time", "Full Time", "Full Time", "Full Time", "Ful…
## $ url          <chr> "https://jobs.github.com/positions/46ab84b4-12dc-4a95-ab…
## $ created_at   <chr> "Fri Oct 09 19:42:30 UTC 2020", "Fri Oct 30 03:52:17 UTC…
## $ company      <chr> "Playco", "Agiloft Inc", "Agiloft Inc", "Agiloft, Inc.",…
## $ company_url  <chr> "http://www.play.co", "http://www.agiloft.com", "http://…
## $ location     <chr> "Remote", "Kyiv, Ukraine, Budapest, Hungary, Russia", "K…
## $ title        <chr> "Senior Fullstack Engineer", " Senior EJB Developer (Rem…
## $ description  <chr> "<p>Here at Playco, we make games that bring the world c…
## $ how_to_apply <chr> "<p><a href=\"https://jobs.lever.co/playco/453d2258-7e66…
## $ company_logo <chr> "https://jobs.github.com/rails/active_storage/blobs/eyJf…
```

```
df <- data %>% select(id, type, company, location, title, description)
df <- as.data.frame(df)
df %>% head() %>% as.tibble() %>% view()
```

2. Exploratory Data Analysis

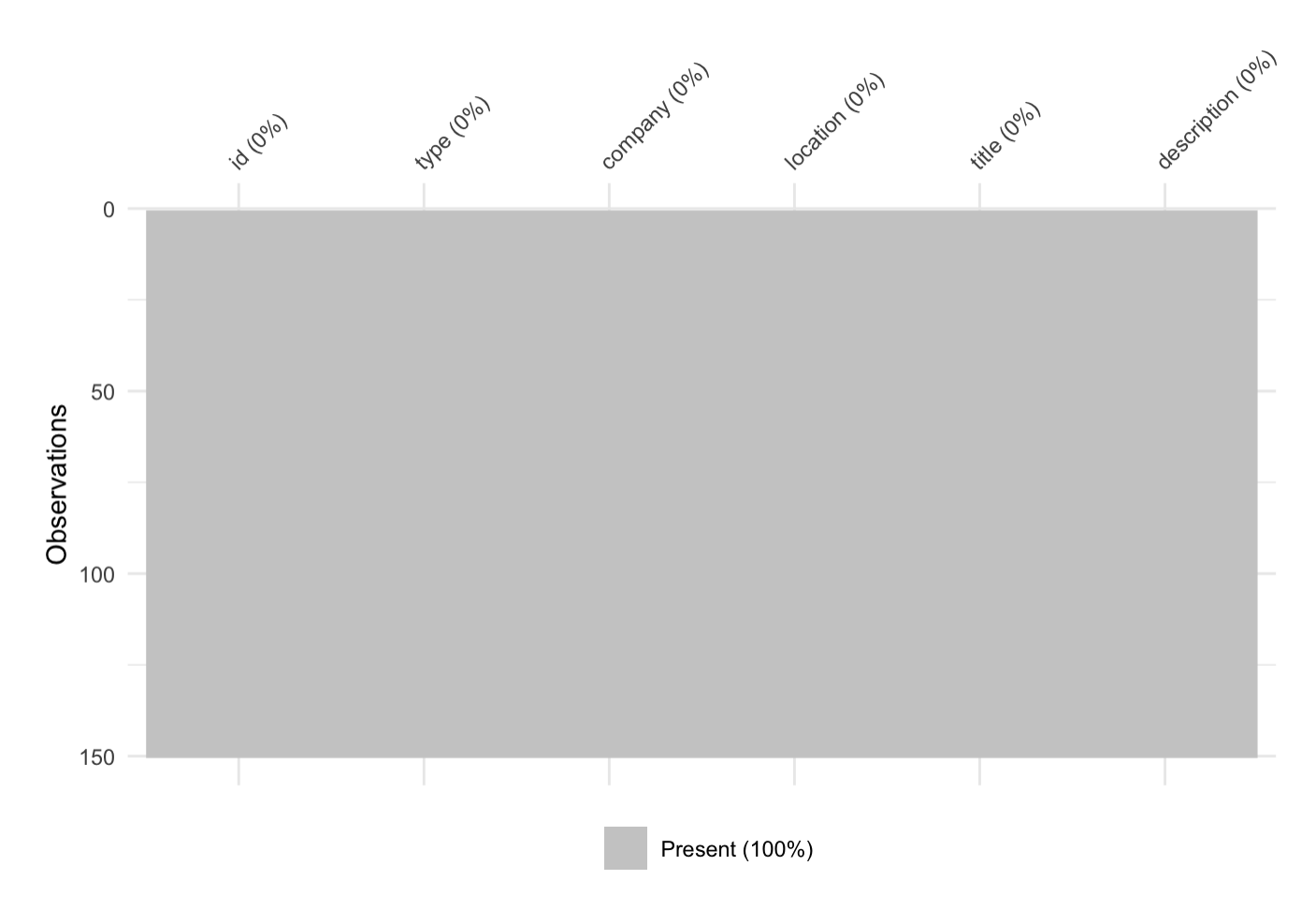
2.1 Missing Value

There is no missing existing in the data set.

```
missing_col<-df %>% is.na() %>% colSums()
  missing_col <- data.frame(missing_col) %>% rownames_to_column()
  missing_df <- missing_col %>%
    mutate(Percentage_of_Missing=missing_col/nrow(df)*100) %>% arrange(desc(Percentage_of_Missing))
print(missing_df)
```

```
##       rowname missing_col Percentage_of_Missing
## 1          id           0                     0
## 2        type           0                     0
## 3     company           0                     0
## 4    location           0                     0
## 5       title           0                     0
## 6 description           0                     0
```

```
vis_miss(df)
```



2.2 Summary for Type, Company, Title and Location

There are two types of job, i.e. Contract and Full Time. There are nearly 150 fulltime jobs compared with Contract's. Top 5 companies listed IT positions are Genpact, Gemini, Raycast, InnoGames Gmbh, and Axios. Top 5 titles listed on Github are Senior Software Engineer, DevOps Engineer, Lead Consultant-UI with React, Node JS, Full Stack Engineer and Developer Advocate. Top 5 work locations are Remote, Berlin, Bangalore and Hamberg.

```
df %>% summarise(distinct_type=n_distinct(type), distinct_company=n_distinct(company),distinct_title=n_dist
```

```
##   distinct_type distinct_company distinct_title distinct_lication
## 1             2              113            141                91
```

```
type_bar <- ggplot(df, aes(x=type)) +
          geom_bar(fill="green") +
          theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
                legend.position = "none", plot.title = element_text(hjust = 0.5, size=25), text = element
company_bar <- ggplot(df %>% group_by(company) %>% summarise(count_company=n()) %>% arrange(-count_company)
          geom_bar(stat="identity", fill="darkred") +coord_flip()+
          theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
                legend.position = "none", plot.title = element_text(hjust = 0.5, size=25), text = element
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

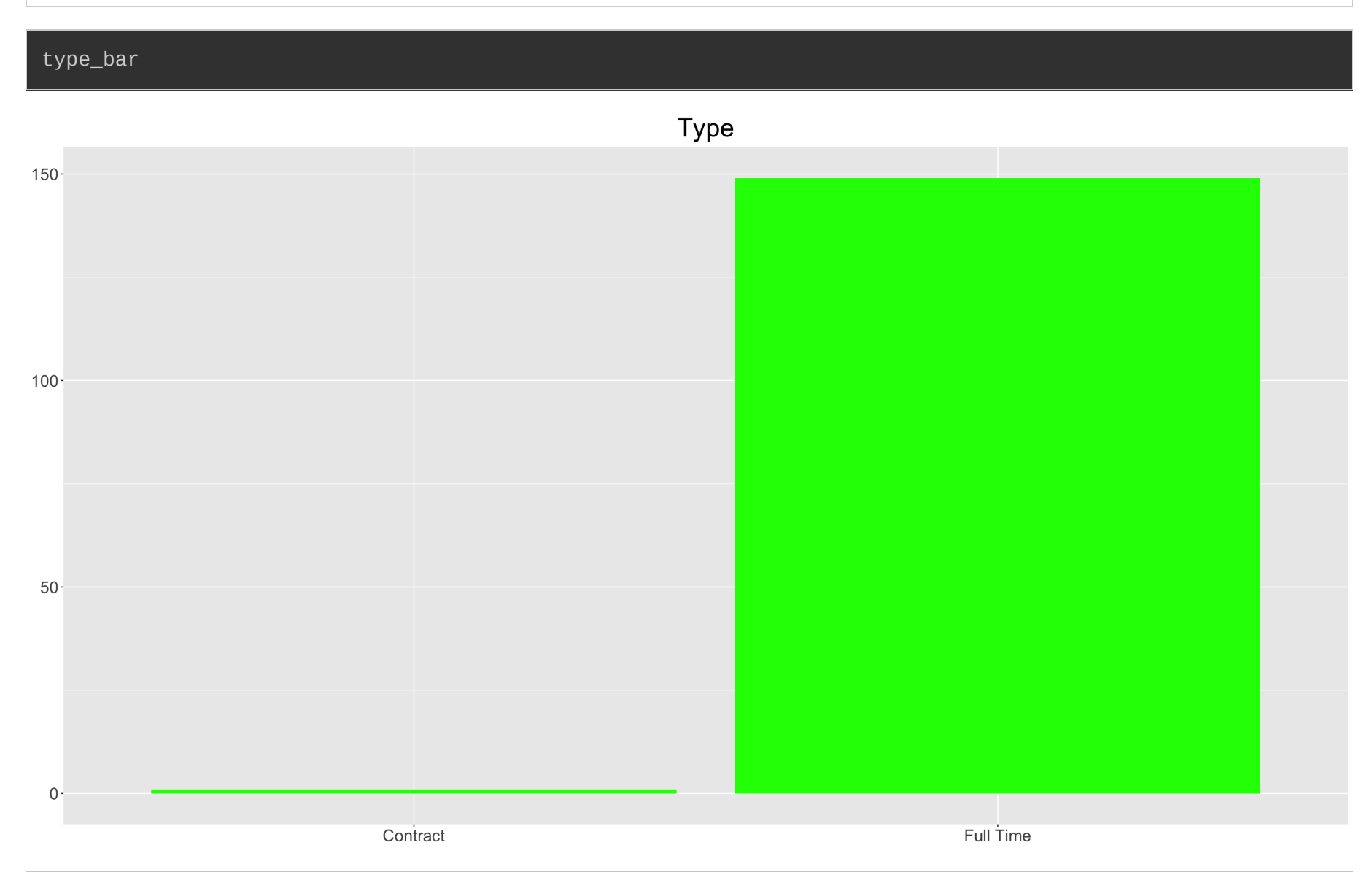
```
title_bar <- ggplot(df %>% group_by(title) %>% summarise(count_title=n()) %>% arrange(-count_title) %>% hea
          geom_bar(stat="identity", fill="purple") +coord_flip()+
          theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
                legend.position = "none", plot.title = element_text(hjust = 0.5, size=25), text = element
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

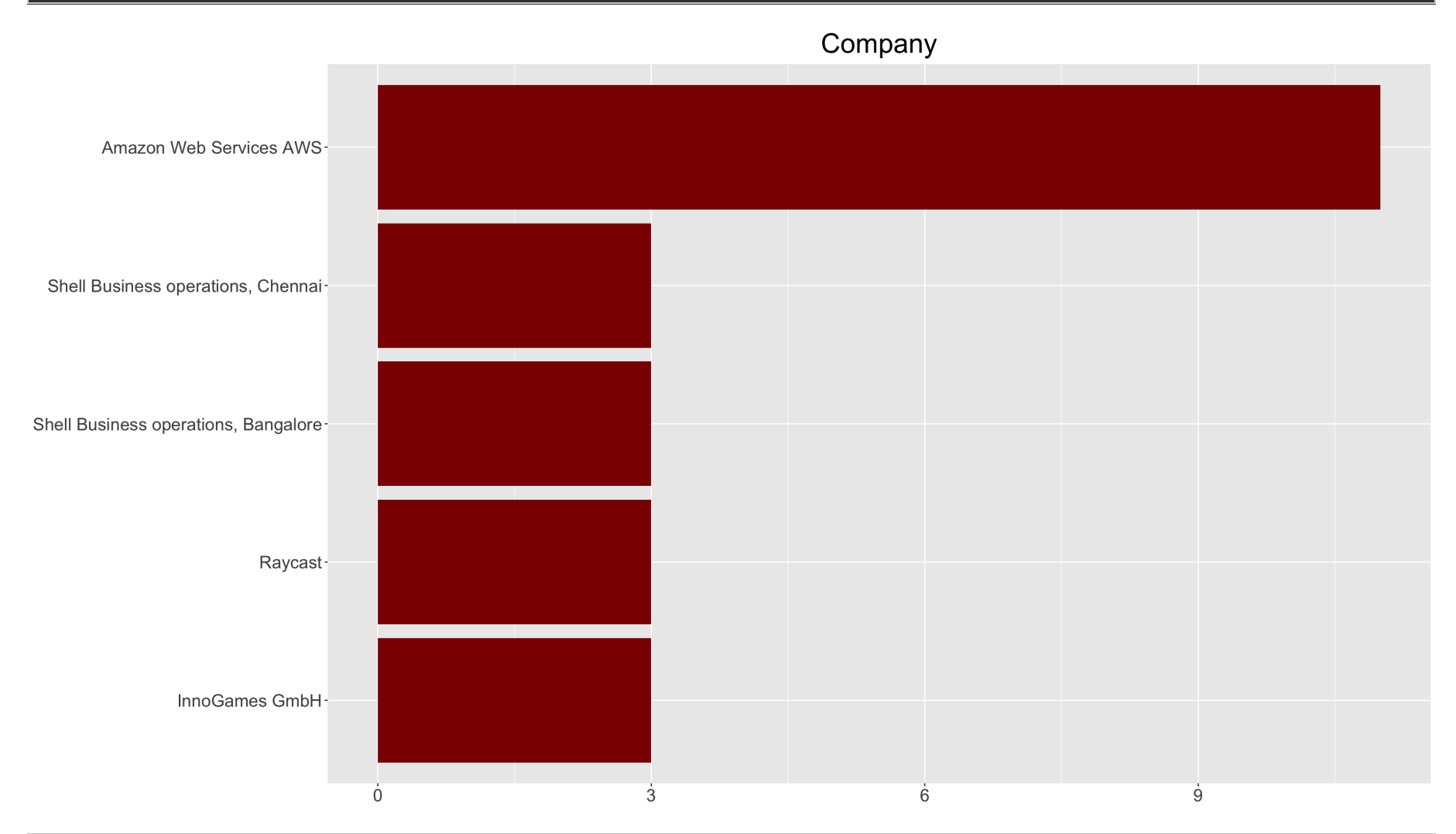
```
location_bar <- ggplot(df %>% group_by(location) %>% summarise(count_location=n()) %>% arrange(-count_locat
          geom_bar(stat="identity", fill="steelblue") + coord_flip()+
          theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
                legend.position = "none", plot.title = element_text(hjust = 0.5, size=25), text = element
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

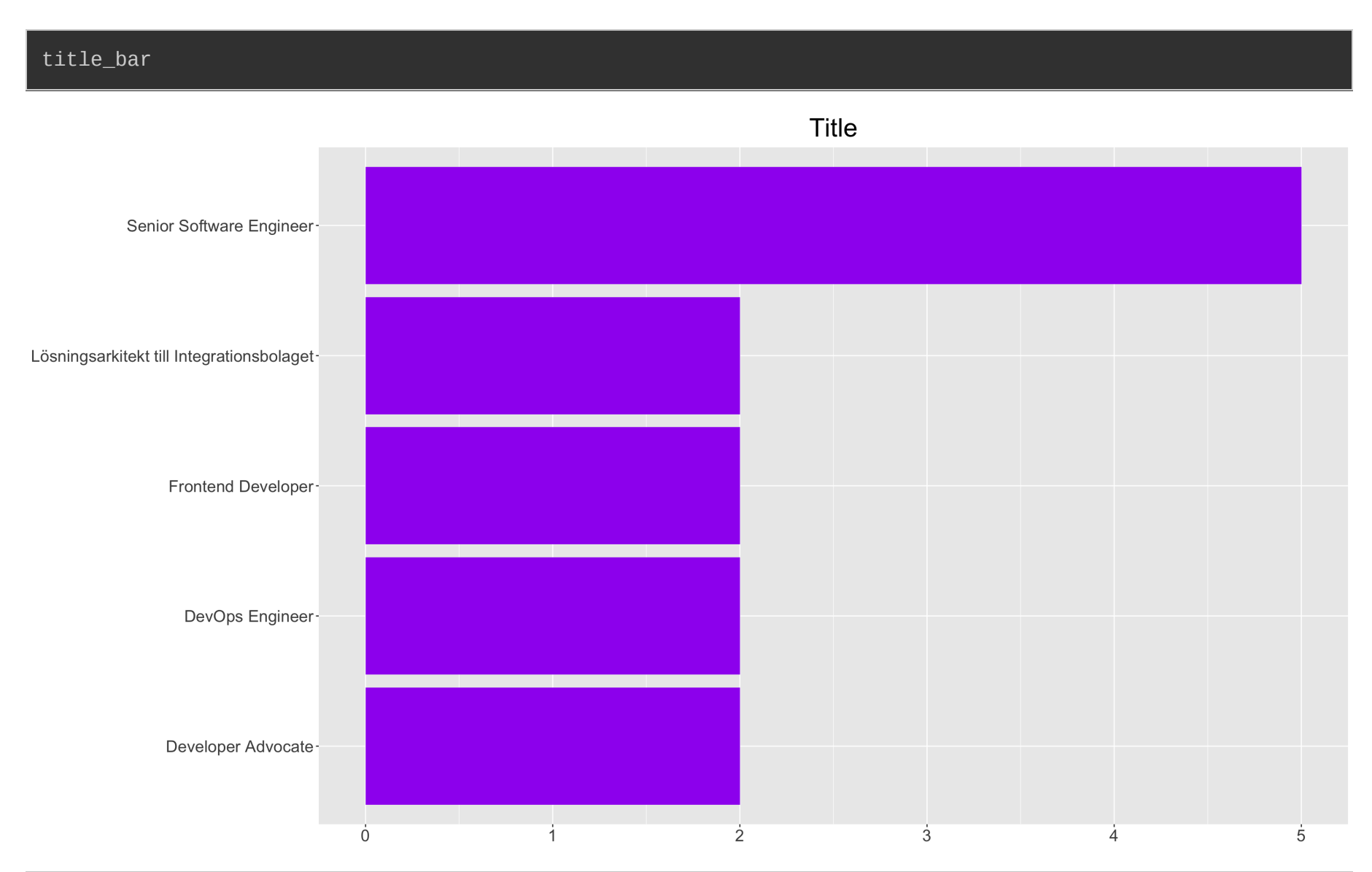
```
type_bar
```



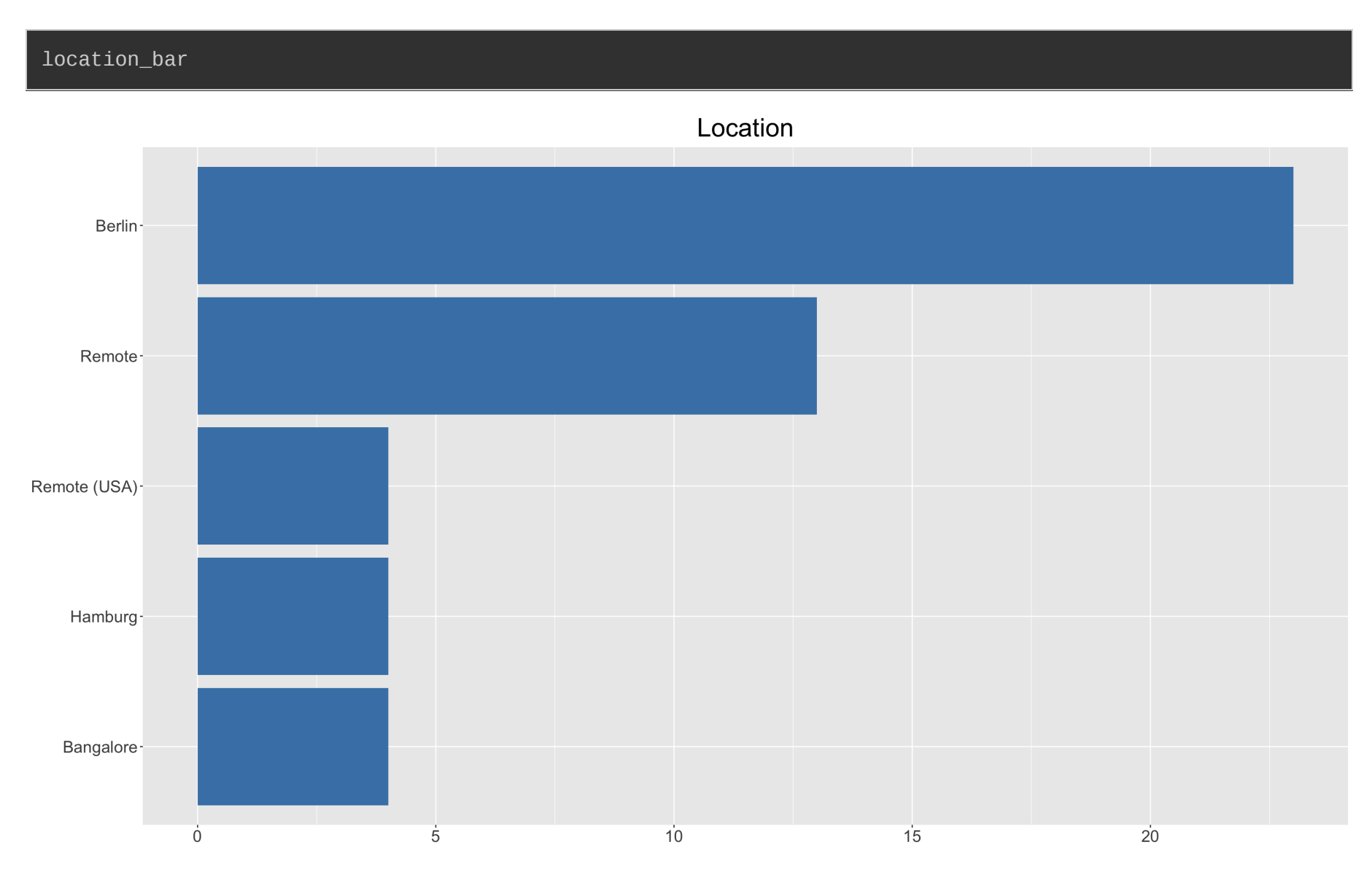

```
company_bar
```




```
title_bar
```




```
location_bar
```

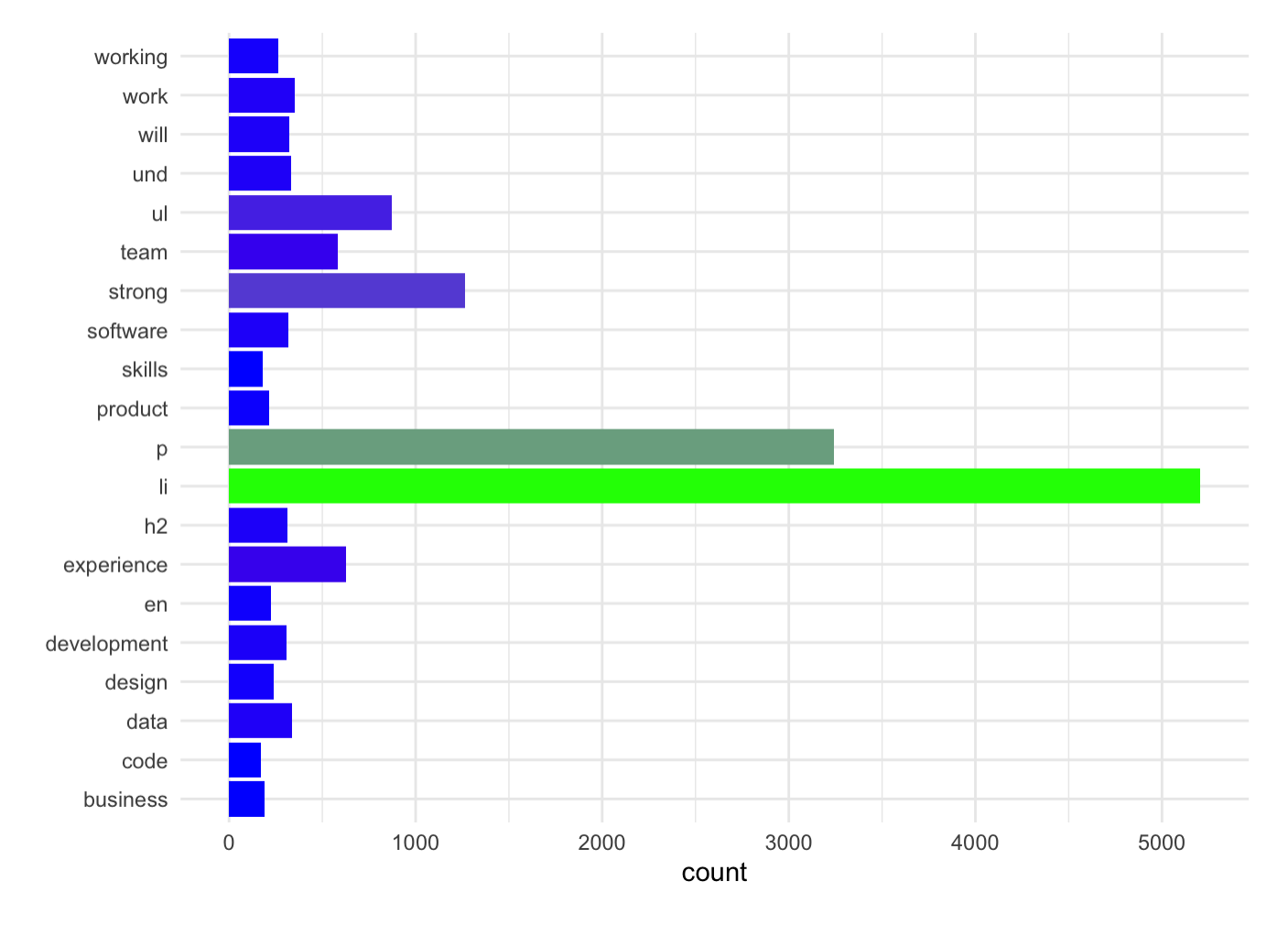


2.3 Description

2.3.1 Common words

First, check the top 20 common words in job description. From plot below, there are some html decoration words which are from websites existing. Besides those html format words, strong, work, experience, data and etc. are the most common words in the descriptions.

```
STOPWORDS <- stopwords(kind = "en")
generate_cm_wd <- function(ccc){
  words <- paste(ccc, collapse = " ")
  words <- tokenize_words(words)
  tab <- table(words[[1]])
  tab <- data_frame(word = names(tab), count = as.numeric(tab))
  tab <- tab %>% filter(!word %in% STOPWORDS) %>% arrange(desc(count)) %>% head(20)
  return(tab)
}

df_cm_wd <- generate_cm_wd(df$description)
df_cmwd_bar<-ggplot(data=df_cm_wd, aes(x=word, y=count, fill=count)) +
  geom_bar(stat="identity")+coord_flip()+
  scale_fill_continuous(low="blue", high="green")+
  theme_minimal() +theme(axis.title.y = element_blank(),
                       legend.position = "none")
df_cmwd_bar
```



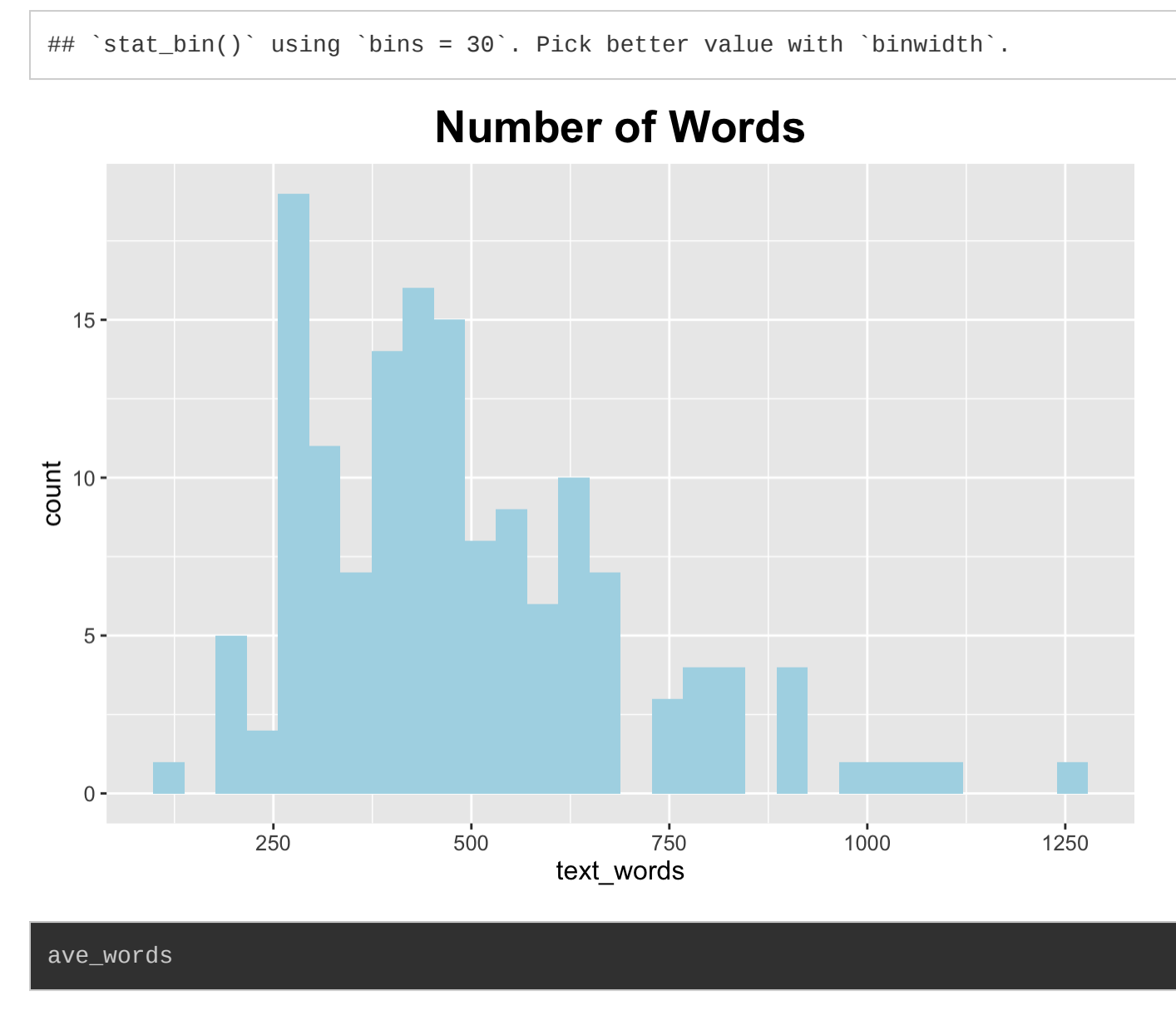
2.3.2 Number of words in each job description and Average length for each job description

Most job descriptions contain between 250 and 800 words. The distribution is little bit right skewed. Job descriptions with more than 1000 words are less than 5.

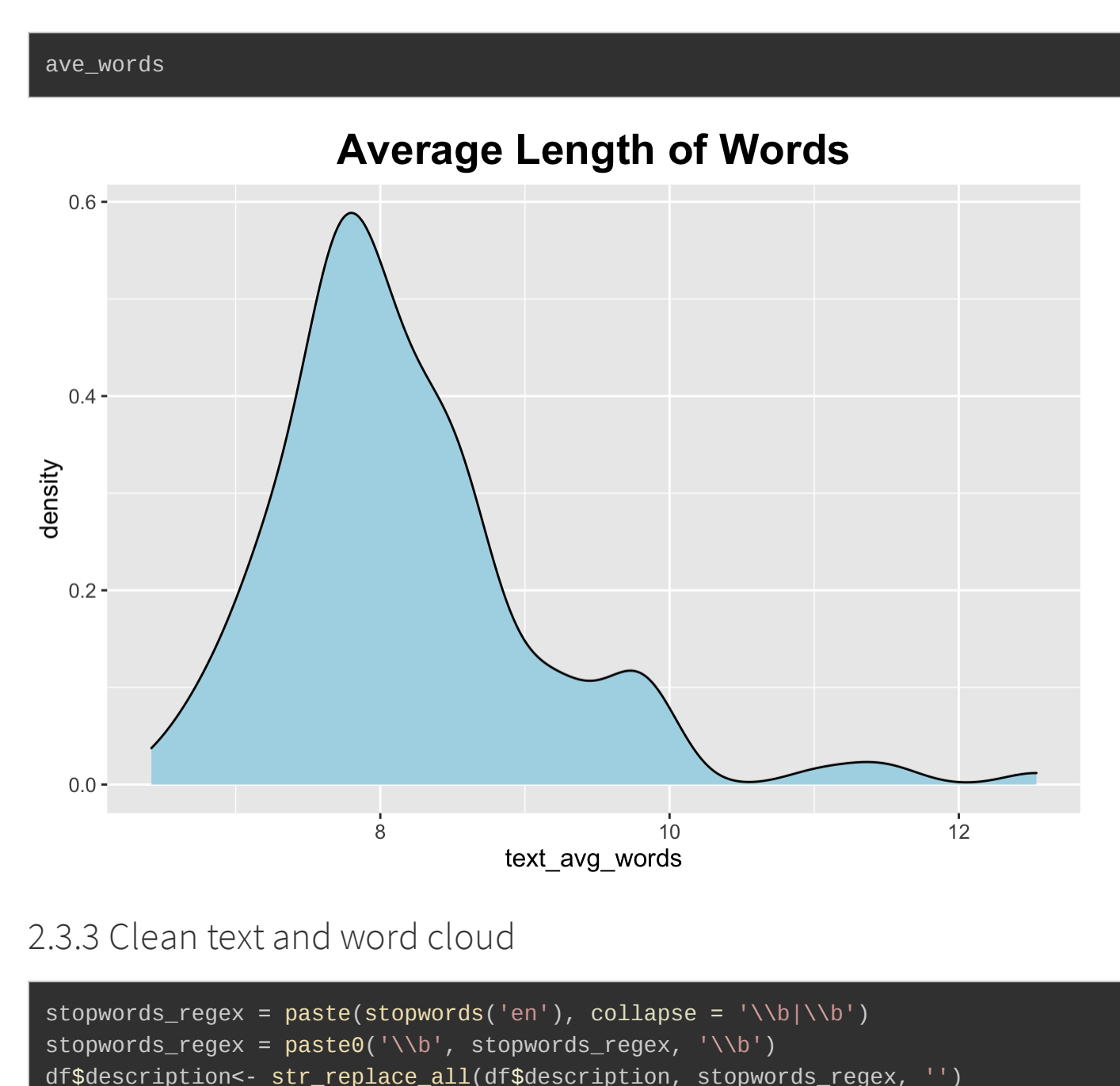
Average words in the description is right skewed

```
df <- df %>% mutate(text_words = sapply(strsplit(df$description, " "), length))
df <- df %>% mutate(text_len = str_count(description))
df <- df %>% mutate(text_avg_words = text_len/text_words)

# Number of words
num_words <- ggplot(df, aes(x=text_words)) +
          geom_histogram(fill="lightblue") +
          theme(legend.position = "none", plot.title = element_text(size = 20, face = "bold", hjust = 0.5
# Average words
ave_words <- ggplot(df, aes(x=text_avg_words)) +
          geom_density(fill="lightblue") +
          theme(legend.position = "none", plot.title = element_text(size = 20, face = "bold", hjust = 0.5


num_words
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ave_words
```



2.3.3 Clean text and word cloud

```
stopwords_regex = paste(stopwords('en'), collapse = '\\b|\\b')
stopwords_regex = paste0('\\b', stopwords_regex, '\\b')
df$description<- str_replace_all(df$description, stopwords_regex, '')
df$description <- str_replace_all(df$description, regex('[:punct:]'), "")
df$description <- str_replace_all(df$description, regex('[:digit:]'), "")
df$description <- str_replace_all(df$description, regex('<p>'), " ")
df$description <- str_replace_all(df$description, regex('und'), " ")
df$description <- str_replace_all(df$description, regex('</p>'), " ")
df$description <- str_replace_all(df$description, regex('will'), " ")
df$description <- str_replace_all(df$description, regex('you'), "")
df$description <- str_replace_all(df$description, regex('the'), "")
df$description <- str_to_lower(df$description, locale = 'en')
description <- Corpus(VectorSource(df %>% select(description)))
desp <- description %>%
    tm_map(removeNumbers) %>%
    tm_map(removePunctuation) %>%
    tm_map(removeWords, stopwords("english")) %>%
    tm_map(stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(., removeNumbers): transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., removePunctuation): transformation drops
## documents
```

```
## Warning in tm_map.SimpleCorpus(., removeWords, stopwords("english")):
## transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., stripWhitespace): transformation drops
## documents
```

```
desp <- tm_map(desp, content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(desp, content_transformer(tolower)):
## transformation drops documents
```

```
desp <- TermDocumentMatrix(desp)
matrix <- as.matrix(desp)
words <- sort(rowSums(matrix),decreasing=TRUE)
pre_words <- data.frame(word = names(words),freq=words)


clean_desp <- wordcloud2(pre_words, size=1.6, color='random-dark')+ WCtheme(1)
clean_desp
```

```
ggplot(pre_words %>% head(10), aes(x=reorder(word, freq), y=freq) +
          geom_bar(stat="identity", fill="steelblue")+coord_flip()+
          theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
                legend.position = "none", plot.title = element_text(hjust = 0.5, size=25), text =
```