Accessing CORD19-NEKG dataset in R

This notebook demonstartes how to access and query *CORD-19 Named Entities Knowledge Graph (CORD19-NEKG)* RDF dataset [1]. The dataset describes named entities identified in the scholarly articles of the COVID-19 Open Research Dataset (CORD-19) [2], a resource of over 47,000 articles about COVID-19 and the coronavirus family of viruses.

References

[1] COVID-19 Open Research Dataset (CORD-19). 2020. Version 2020-04-03. Retrieved from https://pages.semanticscholar.org/coronavirus-research. Accessed 2020-04-06. doi:10.5281/zenodo.3715505

[2] F. Michel, L. Djimenou, C. Faron-Zucker, and J. Montagnat. Translation of Relational and Non-Relational Databases into RDF with xR2RML. In Proceedings of the 11th International Confenrence on Web Information Systems and Technologies (WEBIST 2015), Lisbon, Portugal, 2015.

Cite this work

When including CORD19-NEKG data in a publication or redistribution, please cite the dataset as follows:

R. Gazzotti, F. Michel, F. Gandon. CORD-19 Named Entities Knowledge Graph (CORD19-NEKG). University Côte d'Azur, Inria, CNRS. 2020. Retrieved from https://github.com/Wimmics/cord19-nekg.

```r
library(SPARQL)
```

```r
endpoint <- "https://covid19.i3s.unice.fr/sparql"
options <- NULL
```

```r
prefix <- c('covid','<http://ns.inria.fr/covid19/>',
            'wd',   '<http://www.wikidata.org/entity/>',
            'wdt',  '<http://www.wikidata.org/prop/direct/>')

sparql_prefix <- "
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl:    <http://www.w3.org/2002/07/owl#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>

PREFIX bibo:   <http://purl.org/ontology/bibo/>
PREFIX dce:    <http://purl.org/dc/elements/1.1/>
PREFIX dct:    <http://purl.org/dc/terms/>
PREFIX fabio:  <http://purl.org/spar/fabio/>
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
PREFIX frbr:   <http://purl.org/vocab/frbr/core#>
PREFIX oa:     <http://www.w3.org/ns/oa#>
PREFIX prov:   <http://www.w3.org/ns/prov#>
PREFIX schema: <http://schema.org/>

```

```
prefix wd:     <http://www.wikidata.org/entity/>
prefix wdt:    <http://www.wikidata.org/prop/direct/>

PREFIX covid:   <http://ns.inria.fr/covid19/>
PREFIX covidpr: <http://ns.inria.fr/covid19/property/>
"
```

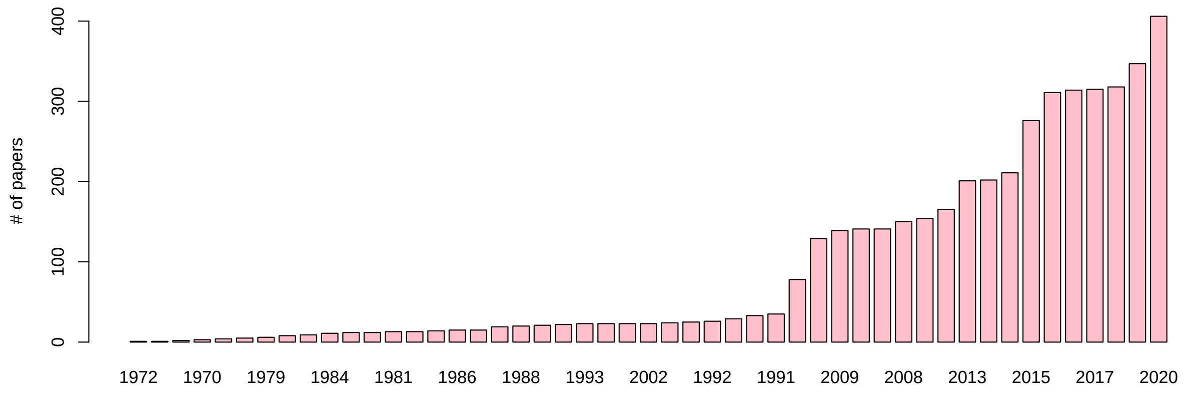
Working with article metadata

Query dataset for the articles that mention the term *coronavirus* in their abstracts.

```
query <- '
SELECT (group_concat(distinct ?name,"; ") AS ?authors)
       ?title
       (year(?date) as ?year)
       ?pub
       ?url

WHERE {
    graph <http://ns.inria.fr/covid19/graph/articles>
    {
        ?doc a ?t;
            dce:creator ?name;
            dct:title ?title;
            schema:publication ?pub;
            schema:url ?url;
            dct:abstract [ rdf:value ?abs ].

        optional { ?doc dct:issued ?date }
        filter contains(?abs, "coronavirus")
    }
}
group by ?doc ?title ?date ?pub ?url
order by desc(?date)

'

query <- paste(sparql_prefix, query)
```

```
res <- SPARQL(url= endpoint,
              query = query,
              ns=prefix,
              extra=NULL)$results
```

```
barplot(sort(table(res$year)),
        col="pink",
        space=0.3,
        ylab="# of papers",
        sub="Number of articles that mention 'coronavirus' in their abstracts per year",
        font.sub=4)
```



Number of articles that mention 'coronavirus' in their abstracts per year

Working with article annotations

Query dataset for the articles refrencing *coronavirus* and forms of *cancer* at the same time

```
query_corona_vs_cancer = '
# wdt:P279 = subclass of
# wdt:P31 = instance of
# wd:Q12078 = cancer
# wd:Q1134583 = coronavirus family

select distinct ?article ?dis1 ?dis1Label ?dis2 ?dis2Label #?dis2Subject

from <http://ns.inria.fr/covid19/graph/entityfishing>
from named <http://ns.inria.fr/covid19/graph/wikidata-named-entities>

where {
    # Look for 2 annotations of the same article with Wikidata URIs ?dis1 and ?dis2

    ?annot1 schema:about ?article; oa:hasBody ?dis1.
    ?annot2 schema:about ?article; oa:hasBody ?dis2.

    graph <http://ns.inria.fr/covid19/graph/wikidata-named-entities>
    {
      ?entity1 rdfs:label "cancer"@en. # ?entity1 is wd:Q12078

      { ?dis1 rdfs:label ?dis1Label.
        filter (?dis1 = ?entity1) } # ?dis1 is "cancer"

      UNION

      { ?dis1 wdt:P279 ?entity1;
              rdfs:label ?dis1Label. }  # ?dis1 is a subclass of "cancer" (at any depth)

      UNION

      { ?dis1 wdt:P31 ?entity1;
```

```
              rdfs:label ?dis1Label. }  # ?dis1 is an instance of "cancer" or a subclass thereof

      ?entity2 rdfs:label "Coronaviridae"@en. # ?entity2 is wd:Q1134583

      { ?dis2 rdfs:label ?dis2Label.
      filter (?dis2 = ?entity2) }

      UNION

      { ?dis2 wdt:P279 ?entity2;
              rdfs:label ?dis2Label. } # ?dis2 is a subclass of "Coronaviridae" (at any depth)

      UNION

      { ?dis2 wdt:P31 ?entity2;
              rdfs:label ?dis2Label. }  # ?dis2 is an instance of "Coronaviridae" or a subclass thereof

    }


}
order by ?dis1 ?dis2
limit 1000
'

query_corona_vs_cancer <- paste(sparql_prefix, query_corona_vs_cancer)
```

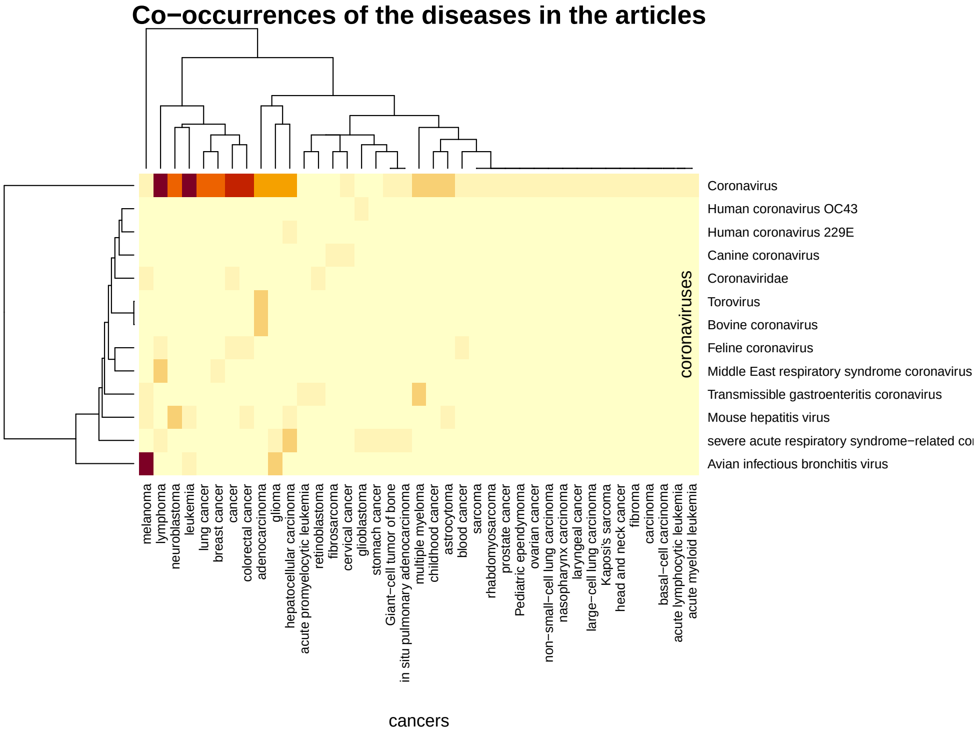
```
res <- SPARQL(url= endpoint,
              query = query_corona_vs_cancer,
              ns=prefix,
              extra=NULL)$results
```

```
# remove label decorations
res <- data.frame(lapply(res, function(x) {
                  gsub("@en", "", x)
              }))
res <- data.frame(lapply(res, function(x) {
                  gsub("\"", "", x)
              }))
```

Visualize query results in different ways

Plot hierarchically-clustered heatmap

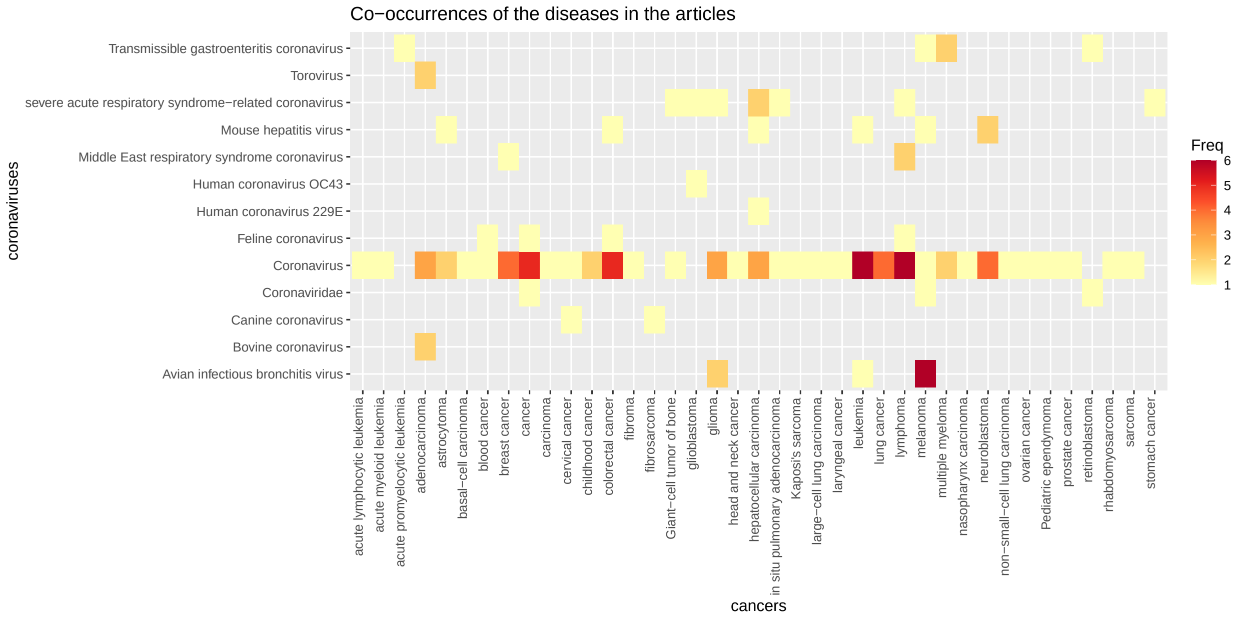
```
hm <- table(res[, c('dis2Label','dis1Label')])
hm <- as.matrix(hm)

heatmap(hm,  margins = c(15,0), #Rowv = NA, Colv = NA,
        col = hcl.colors(12, "YlOrRd", rev = TRUE),
        scale="none",
        cexCol=0.9,
        cexRow = 0.9,
        main="Co-occurrences of the diseases in the articles",
        xlab="cancers",
        ylab="coronaviruses")
```



Plot heatmap with ggplot

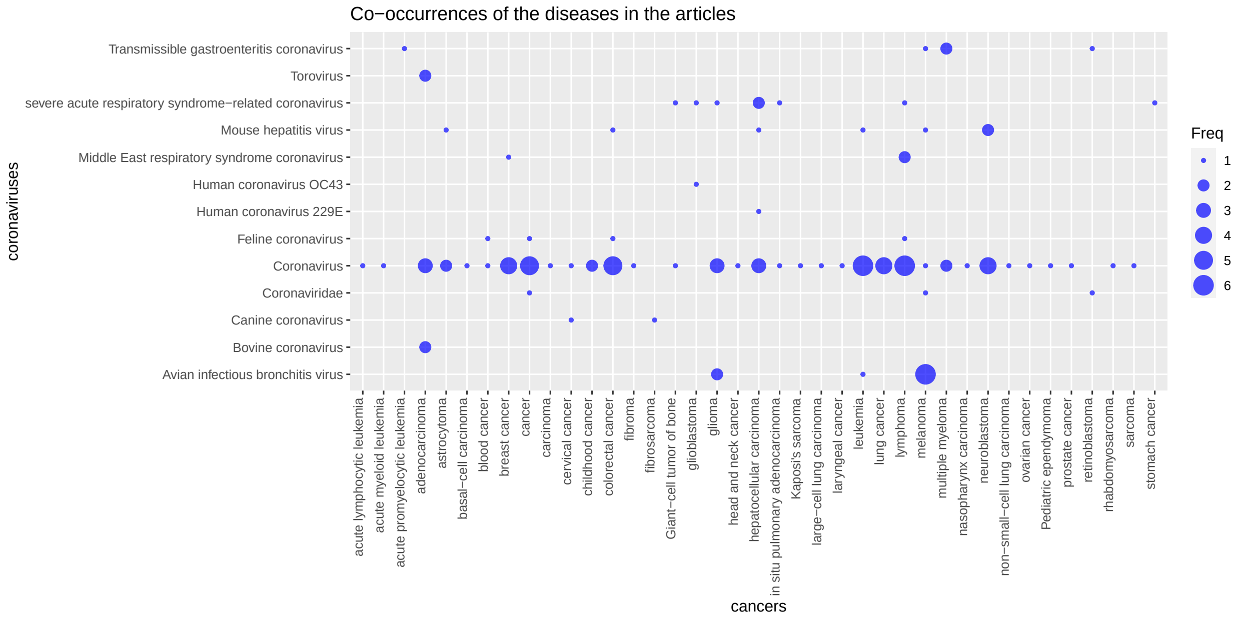
```
library(ggplot2)
```

```
counts <- table(res[, c('dis2Label','dis1Label')])
counts <- as.data.frame(counts)
counts <- counts[counts$Freq > 0, ]

ggplot(counts, aes(x=dis1Label, y=dis2Label, fill=Freq)) +
       geom_tile() +
       theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust=0)) +
       scale_fill_distiller(palette = "YlOrRd", direction = 1) +
       ggtitle("Co-occurrences of the diseases in the articles") +
       xlab("cancers") +
       ylab("coronaviruses")
```



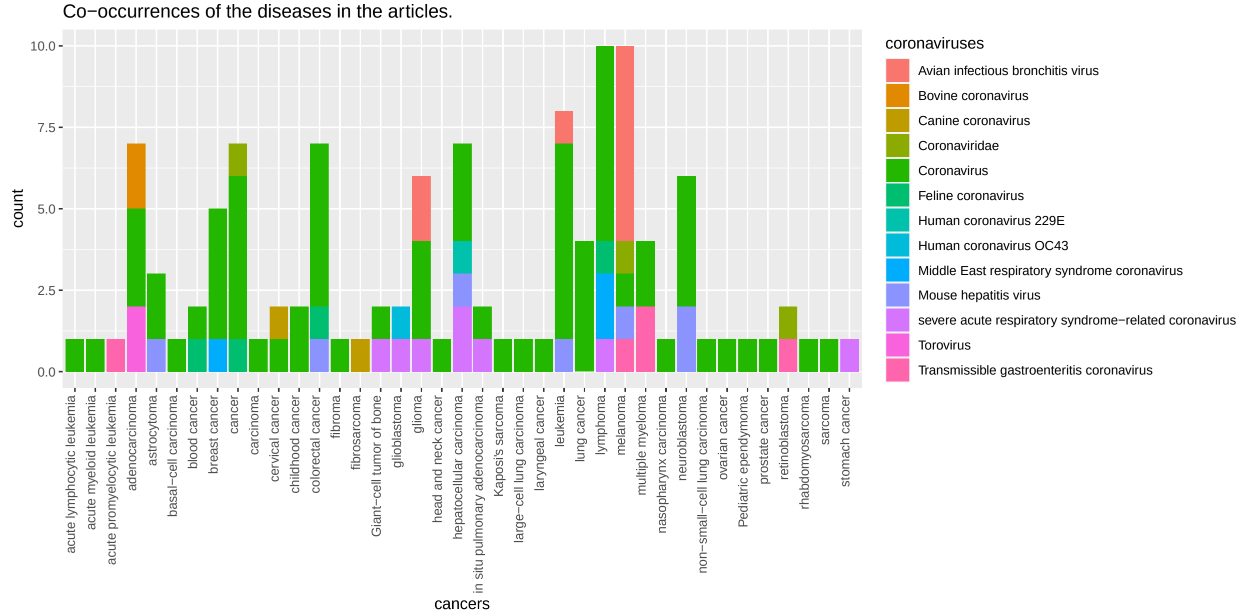
Plot bubble chart

```
# Most basic bubble plot
ggplot(counts, aes(x=dis1Label, y=dis2Label, size = Freq)) +
       geom_point(alpha=0.7, color='blue') +
       theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust=0)) +
       ggtitle("Co-occurrences of the diseases in the articles") +
       xlab("cancers") +
       ylab("coronaviruses")
```



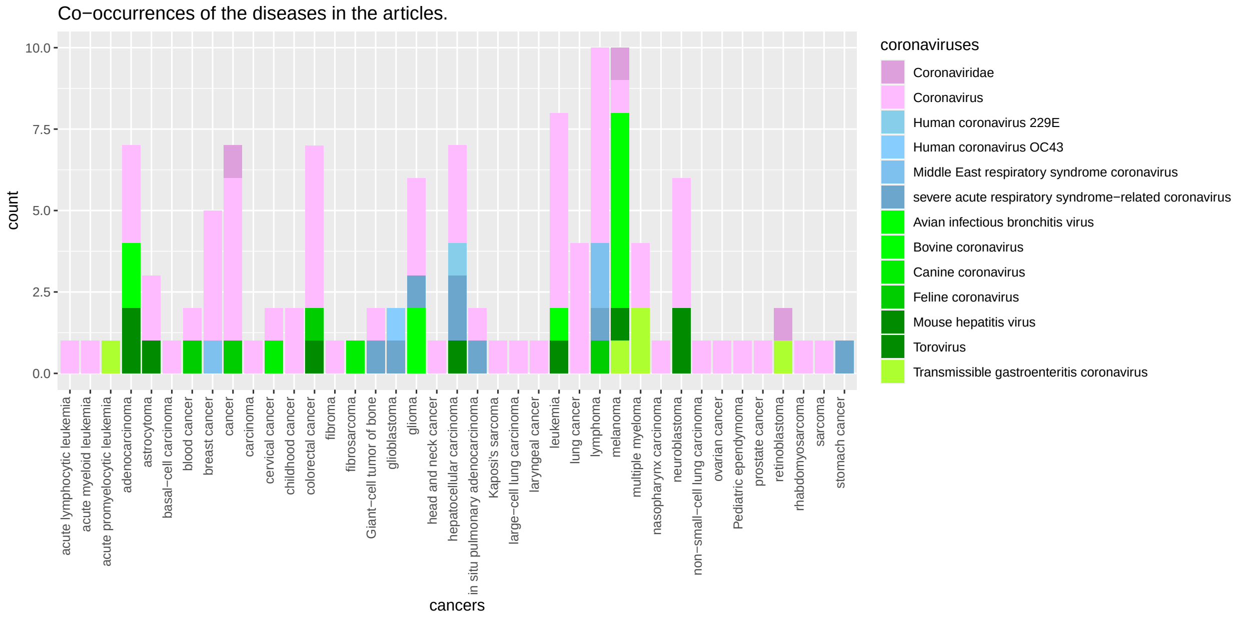
Plot stacked bar chart with default colors

```
# stacked bar chart
ggplot(res,
       aes(x = dis1Label,
           fill = dis2Label)) +
  geom_bar(position = "stack") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust=0)) +
  ggtitle("Co-occurrences of the diseases in the articles.") +
  xlab("cancers") +
  ylab("count") +
  labs(fill = "coronaviruses")
```



Plot barchart with manual coloring

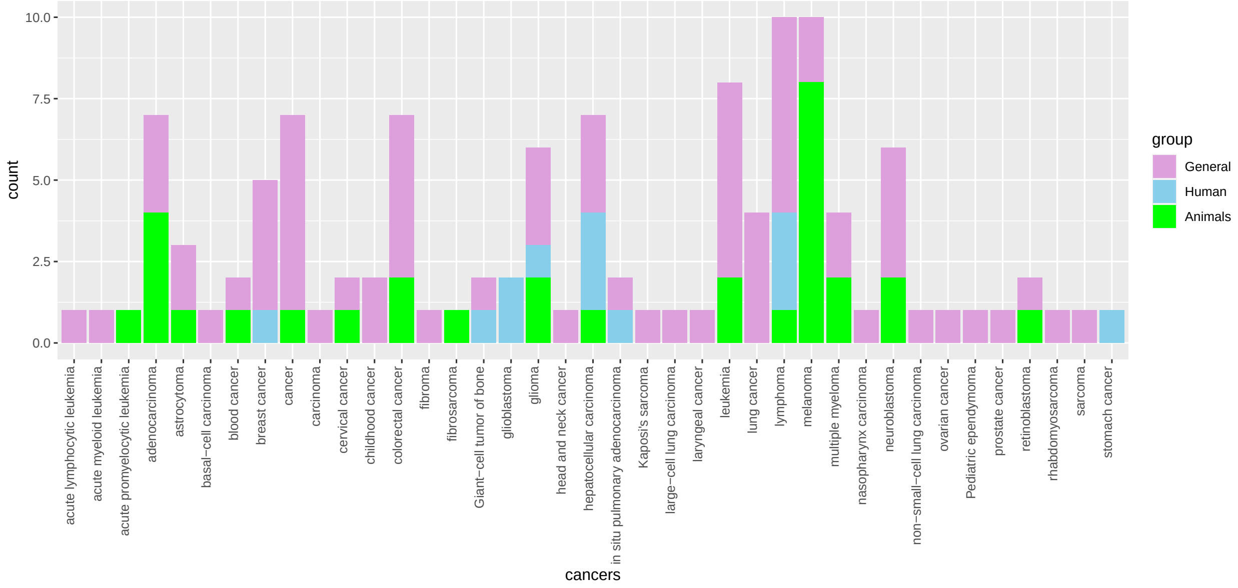
```
# Manually group coronaviruses in 3 groups (general(1), human(2), animal(3)) for reordering
res$group <- rep(3, 117)
res$group[grep('Human', res$dis2Label )] <- 2
res$group[grep('Corona', res$dis2Label )] <- 1
res$group[grep('severe acute', res$dis2Label )] <- 2
res$group[grep('Middle', res$dis2Label )] <- 2

# Manually choose colors for coronaviruses (shades of purple for general (1), blues for humans(2), and
my_colors <- c(colors()[254:256], #greens
               colors()[541:542], #purples
               colors()[257],
               colors()[589:591], #blues
               colors()[258], colors()[592], colors()[258:259] )

names(my_colors) <- levels(res$dis2Label)

ggplot(res,
       aes(x = dis1Label,
           fill = reorder(dis2Label , group))) +
  geom_bar(position = "stack") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust=0)) +
  scale_fill_manual(name = "coronaviruses", values = my_colors) +
  ggtitle("Co-occurrences of the diseases in the articles.") +
  xlab("cancers") +
  ylab("count") +
  labs(fill = "coronaviruses")
```



Plot barplot for grouped diseases

```
group_colors = c(colors()[541], #purple
                 colors()[589], #blue
                 colors()[254]) #green
ggplot(res,
       aes(x = dis1Label,
           fill =  factor(group, levels = c(1, 2, 3), labels = c("General", "Human", "Animals")))) +
  geom_bar(position = "stack") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust=0)) +
  scale_fill_manual(name = "group", values = group_colors) +
  ggtitle("Co-occurrences of the groups of coronavirus diseases in the articles.") +
  xlab("cancers") +
  ylab("count") +
  labs(fill = "coronaviruses")
```

Co−occurrences of the groups of coronavirus diseases in the articles.


count
group
General
Human
Animals
acute lymphocytic leukemia
acute myeloid leukemia
acute promyelocytic leukemia
adenocarcinoma
astrocytoma
basal−cell carcinoma
blood cancer
breast cancer
cancer
carcinoma
cervical cancer
childhood cancer
colorectal cancer
fibroma
fibrosarcoma
Giant−cell tumor of bone
glioblastoma
glioma
head and neck cancer
hepatocellular carcinoma
in situ pulmonary adenocarcinoma
Kaposi's sarcoma
large−cell lung carcinoma
laryngeal cancer
leukemia
lung cancer
lymphoma
melanoma
multiple myeloma
nasopharynx carcinoma
neuroblastoma
non−small−cell lung carcinoma
ovarian cancer
Pediatric ependymoma
prostate cancer
retinoblastoma
rhabdomyosarcoma
sarcoma
stomach cancer
cancers