Analytical Notebook

Stefan Zimmermann

11 7 2020

Install and load R-Packages

This function installs and loads the required R-packages for the analytical report. The following packages are required for the report:

- readr (rectangular data like csv, tsv, and fwf)
- knitr (engine for dynamic report generation with R)
- dplyr (tools for data manipulation)
- tidyr (tools for data manipulation)
- sparklyr (R interface for Apache Spark)
- ggplot2 (system for declaratively creating graphics)

```
# install and load packages
loadpackage <- function(x){
  for( i in x ){
    #  require returns TRUE invisibly if it was able to load package
    if( ! require( i , character.only = TRUE ) ){
      #  If package was not able to be loaded then re-install
      install.packages( i , dependencies = TRUE )
    }
    #  Load package (after installing)
    library( i , character.only = TRUE )
  }
}

# load packages
loadpackage( c("readr", "knitr", "dplyr", "tidyr", "sparklyr", "ggplot2"))
```

Load Covid-Datasets

The datasets UID_ISO_FIPS_LookUp_Table.csv and time_series_covid19_confirmed_global.csv are loaded and then harmonised (e.g. The ID variables must be standardized) to merge both datasets. Since we are only interested in countries and not in regions we keep only a subset of the dataset without countries regions combinations. To better handle the data we reshape the data with the date variables to long format. Since data variables are easy to handle in R we tidy the date variables before setting up the spark connection. Afterwards the Spark connection is established and the data sets are uploaded to Spark.

```
# use url to current dataset
url_data1 <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/UID_I
url_data2 <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_

# load datasets
data1 <- read_csv(url_data1)
data2 <- read_csv(url_data2)

# harmonise ID-Variables in both datasets
names(data1)[names(data1) == "Long_"] <- "Long"
```

```r
names(data2)[names(data2) == "Country/Region"] <- "Country_Region"
names(data2)[names(data2) == "Province/State"] <- "Province_State"

# Since we are only interested in countries and not in regions
# we keep only a subset of the dataset without regions.
data1 <-subset(data1, is.na(data1$Province_State))
data2 <-subset(data2, is.na(data2$Province_State))

# reshape the dataset with the date variable in long format
data2 <-
  reshape(
    data = as.data.frame(data2),
    varying = list(names(data2)[5:length(data2)]),
    timevar = "day",
    v.names = "count",
    idvar = c("Country_Region"),
    direction = "long",
    times = names(data2)[5:length(data2)]
  )

data2$date <- as.Date(data2$day, format = "%m/%d/%y")
data2$datecount <- data2$date-min(data2$date)
days <- unique(data2$day)
weeks <- days[seq(1, length(days), 7)]

write_csv(data1, path = '../input/data1.csv')
write_csv(data2, path = '../input/data2.csv')

# setting up spark
sc <- spark_connect(master = "local",
                    version = "2.4.3")

data1 <- sdf_copy_to(sc, data1, overwrite = T)
data2 <- sdf_copy_to(sc, data2, overwrite = T)
src_tbls(sc)
```

Data Cleaning

Here the two data sets are merged. We select the needed variables and save the merged datasets. Now the data set is limited to the countries (Germany, Japan, United Kingdom, US, Brazil, Mexico) which are to be analyzed. Within this process new variables are created that logarithmise the number of corona cases, divide the number of corona cases by 1000 and generate the infection rate.

```r
data_merge <- merge(data2, data1, by = "Country_Region")

data_merge <- data_merge %>%
  select(Country_Region, Country_Region, day, date, count, datecount, Population)

write_csv(data_merge, path = '../input/data_merge.csv')
data_merge <- sdf_copy_to(sc, data_merge, overwrite = T)

# Germany,Japan, United Kingdom, US, Brazil, Mexico
my_data <- data_merge %>%
```

```
  filter(Country_Region=="Germany" | Country_Region=="Mexico" |
         Country_Region=="United Kingdom" | Country_Region=="US" |
         Country_Region=="Brazil" | Country_Region=="China" |
         Country_Region=="Japan") %>%
  mutate(rate = (count/Population)*100) %>%
  mutate(count2 = round((count/1000))) %>%
  mutate(logcount = log(count)) %>%
  mutate(logcount = ifelse(logcount == 0, -Inf, logcount)) %>%
  mutate(logcount = ifelse(is.na(logcount) == TRUE , 0, logcount)) %>%
  collect()

my_data <- sdf_copy_to(sc, my_data, name = "my_data", overwrite = TRUE)
```

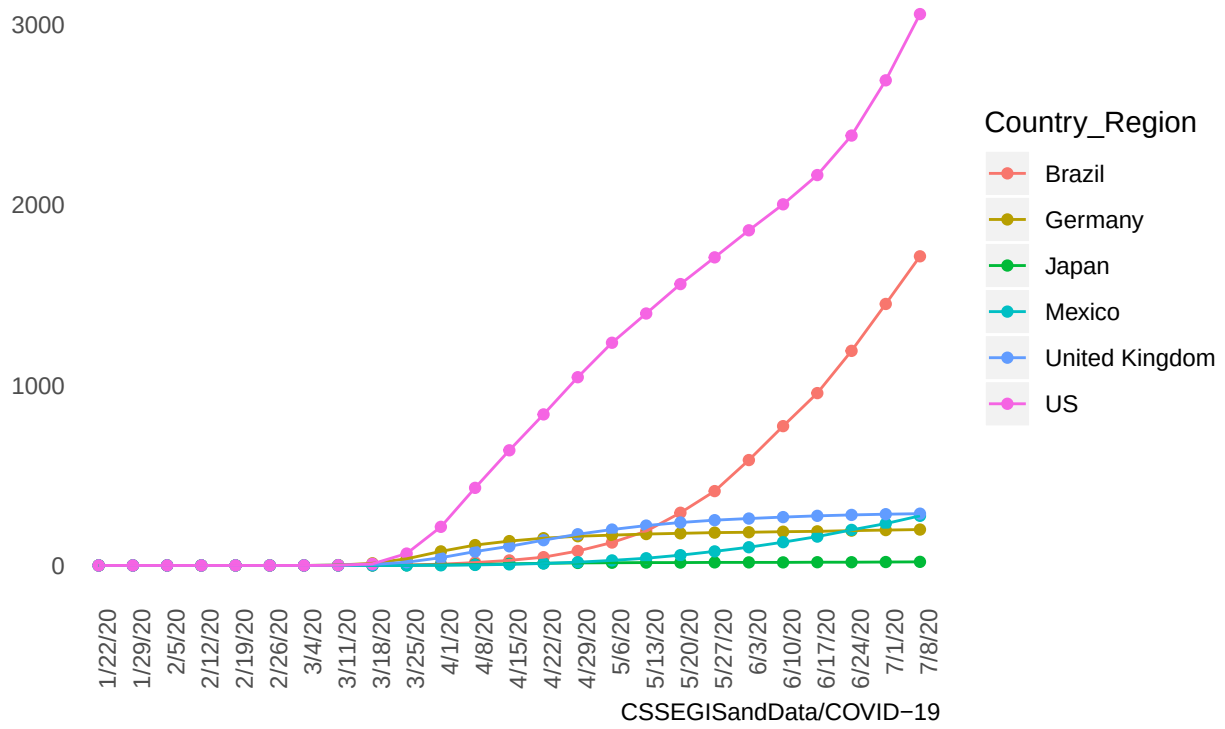
Plots

Interpretation

```
ggplot(data=my_data, aes(x=reorder(day, count2), y=count2, group=Country_Region, colour=Country_Region))
  geom_point()+
  geom_line()+
  scale_x_discrete(limit = weeks)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        axis.title=element_blank(),
        axis.ticks = element_blank(),
        strip.text = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank())+
 labs(title =  "Overall change of number of Corona Cases in TSD",
       caption = "CSSEGISandData/COVID-19")
```

```
## Warning: Removed 888 rows containing missing values (geom_point).
```

```
## Warning: Removed 888 rows containing missing values (geom_path).
```




```
ggsave("../output/total_change.png")
```

```
## Saving 6.5 x 4.5 in image
```

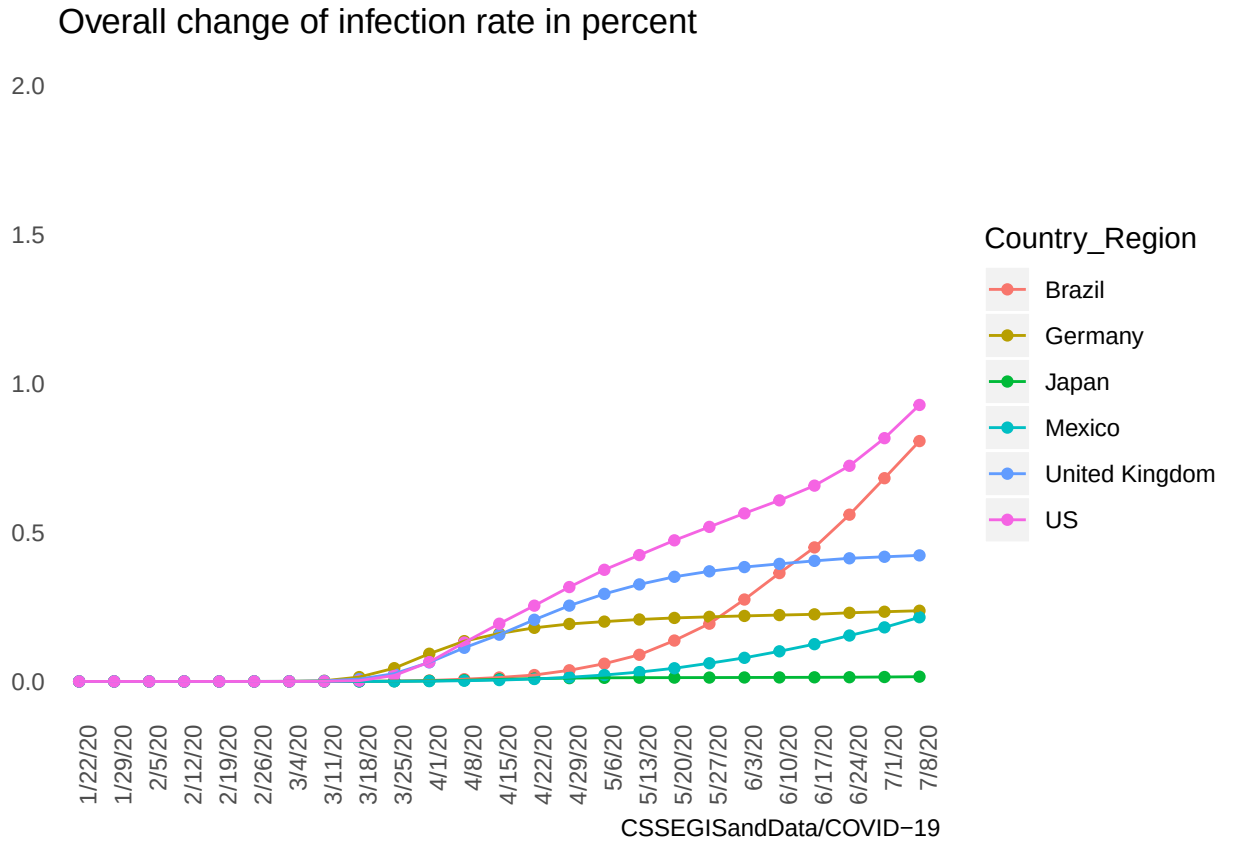
```
## Warning: Removed 888 rows containing missing values (geom_point).
```

```
## Warning: Removed 888 rows containing missing values (geom_path).
```

```
ggplot(data=my_data, aes(x=reorder(day, rate), y=rate, group=Country_Region, colour=Country_Region)) +
  geom_point()+
  geom_line()+
  scale_x_discrete(limit = weeks)+
  ylim(0,2)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        axis.title=element_blank(),
        axis.ticks = element_blank(),
        strip.text = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank())+
 labs(title =  "Overall change of infection rate in percent",
       caption = "CSSEGISandData/COVID-19")
```

```
## Warning: Removed 888 rows containing missing values (geom_point).
```

```
## Warning: Removed 888 rows containing missing values (geom_path).
```




```
ggsave("../output/rate_change.png")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 888 rows containing missing values (geom_point).
## Warning: Removed 888 rows containing missing values (geom_path).
```

Regression

Interpretation

```
partitions <- my_data %>%
  sdf_random_split(training = 0.7, test = 0.3, seed = 1111)

my_data_training <- partitions$training
my_data_test <- partitions$test

lm_model <- my_data_training %>%
  ml_linear_regression(logcount ~ Country_Region + Population + datecount) %>%
  summary()
```

```
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -335.11 -116.23  -45.36  124.13  427.88
##
## Coefficients:
```

```
##                   (Intercept)         Country_Region_Brazil
##                  9.546045e+01                -9.090564e+00
##        Country_Region_Germany         Country_Region_Mexico
##                  -3.554955e+01                -3.302603e+01
## Country_Region_United Kingdom            Country_Region_US
##                  -2.954663e+00                -9.505013e+00
##                     Population                    datecount
##                  -1.729656e-08                 1.793819e+00
##
## R-Squared: 0.2368
## Root Mean Squared Error: 162.5
```