Aufgabe3

Max Melchior Lang

8/4/2021

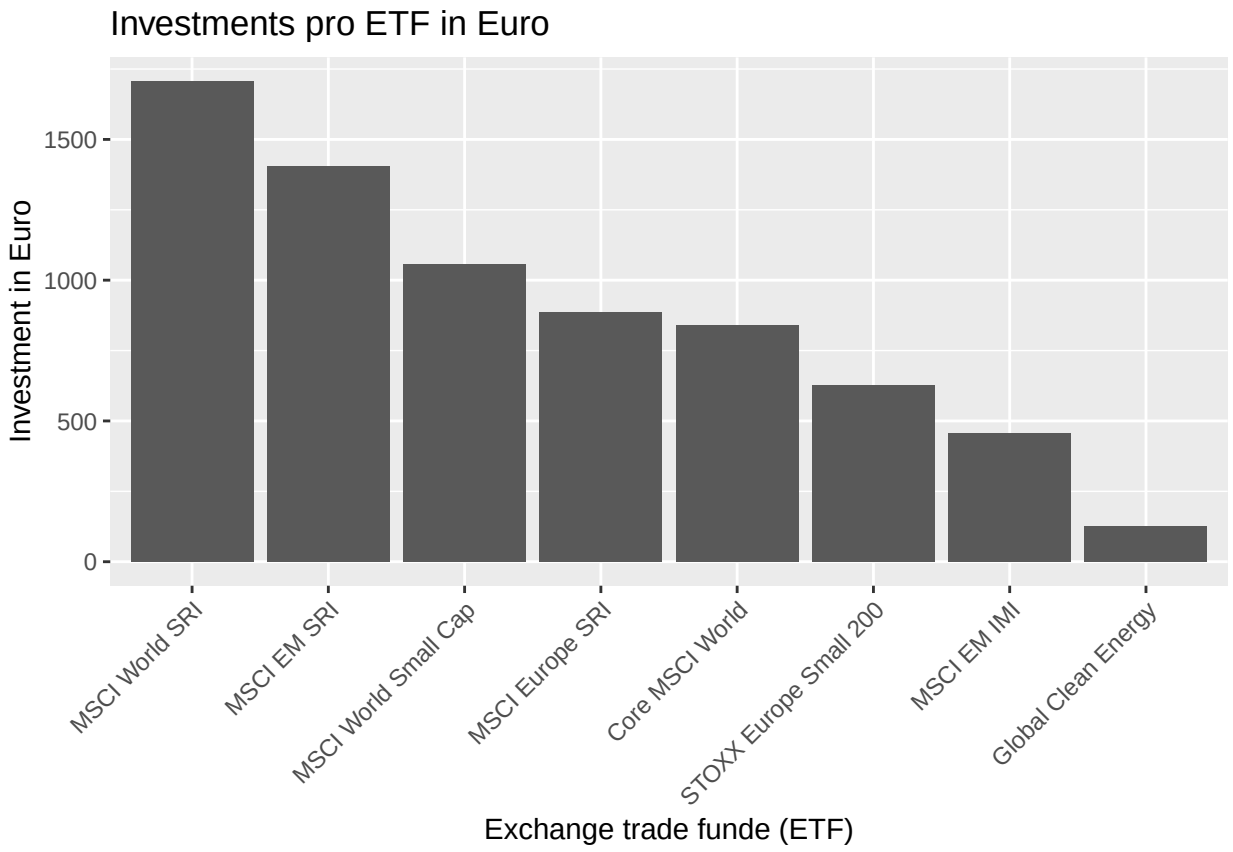
Aufgabe 3

```
etf_overview <- readRDS("etf-overview.Rds")
etf_overview
```

```
##   ETF_ID                    ETF Anteile  Kurs
## 1      1          MSCI World SRI     200  8.53
## 2      2             MSCI EM SRI     180  7.81
## 3      3   MSCI World Small Cap     170  6.21
## 4      4         MSCI Europe SRI      15 59.12
## 5      5         Core MSCI World      12 70.08
## 6      6 STOXX Europe Small 200      17 36.87
## 7      7             MSCI EM IMI      14 32.48
## 8      8    Global Clean Energy      11 11.33
```

a)

```
etf_overview <- etf_overview %>%
  mutate("Investment"= Anteile*Kurs)

ggplot(etf_overview, aes(x= reorder(ETF, -Investment, sum), y= Investment))+
  geom_col()+
  guides(x =  guide_axis(angle = 45))+
  ggtitle("Investments pro ETF in Euro")+
  xlab("Exchange trade funde (ETF)")+
  ylab("Investment in Euro")
```



b)

```
### import_etf function
###
### Imports single .csv files containing ETF Data. Refers to the previously read in
### etf_overview data frame to get specified ETF ID.
###
### Arguments:
### path: A character vector of length 1: The path to the respected .csv file
### id: Specified ETF_ID from etf_overview
### Returns: A data frame. The read in .csv file.

import_etf <- function(path, id){
  #Input checks
  if(!(is.character(path) && (length(path)==1))){
    stop("The path argument has to be a character vector of length 1")
  }
  if(!(is.numeric(id) && (length(id)==1))){
    stop("The id argument has to be a numeric vector of length 1")
  }

  data <- read_delim(path, ":",col_names = TRUE, comment= "Stand:")
  data <- add_column(data, ETF_ID= etf_overview[["ETF_ID"]][id], .before = 1)
  return(data)
}
```

```
core_msci_world <-  import_etf(
  path = "data/CoreMSCIWorld.csv",
  id = 5 )

core_msci_world %>%
  select(ETF_ID, ISIN, Name, Kurs)
```

```
## # A tibble: 1,616 x 4
##    ETF_ID ISIN         Name                    Kurs
##     <int> <chr>        <chr>                  <dbl>
##  1      5 US0378331005 APPLE INC              136.33
##  2      5 US5949181045 MICROSOFT CORP         271.4
##  3      5 US0231351067 AMAZON COM INC        3448.14
##  4      5 US30303M1027 FACEBOOK CLASS A INC   351.89
##  5      5 US02079K1079 ALPHABET INC CLASS C 2520.37
##  6      5 US02079K3059 ALPHABET INC CLASS A 2445.45
##  7      5 US88160R1014 TESLA INC              680.76
##  8      5 US67066G1040 NVIDIA CORP            801.07
##  9      5 US46625H1005 JPMORGAN CHASE & CO    154.14
## 10      5 US4781601046 JOHNSON & JOHNSON      164.03
## # ... with 1,606 more rows
```

c)

```
files <- list.files(path= "data", pattern = ".csv", full.names = TRUE)

# Cleaning files for agrep
files_name <- str_remove(files, ".csv") %>%
  str_remove(., "data/") %>%
  str_replace(., "MSCI\\s*(?!$)", "MSCI ") %>%
  str_replace(., "SRI$", " SRI") %>%
  str_replace(., "Stoxx200Small","STOXX Europe Small 200" )

# Matchin IDs
id_vector <- vector()
for(i in seq_along(files_name)){
  id_vector[i] <- agrep(files_name[i], etf_overview[["ETF"]])[1]
}

etf_data <- map2_df(.x= files, .y= id_vector, .f= ~import_etf(path= .x, id= .y)) %>%
  arrange(.$ETF_ID)
```

```
etf_data %>%
  head()
```

```
## # A tibble: 6 x 15
##   ETF_ID Emittententicker Name     Anlageklasse `Gewichtung (%)`   Kurs Nominale
##    <int> <chr>            <chr>    <chr>                   <dbl>  <dbl>    <dbl>
## 1      1 MSFT             MICROSO~ Aktien                   4.78 271.4    823502
## 2      1 TSLA             TESLA I~ Aktien                   3.92 680.76   269670
## 3      1 NVDA             NVIDIA ~ Aktien                   3.72 801.07   217385
## 4      1 HD               HOME DE~ Aktien                   2.58 318.24   378591
## 5      1 DIS              WALT DI~ Aktien                   2.37 173.93   636707
## 6      1 ASML             ASML HO~ Aktien                   2.22 702.28   147795
```

```
## # ... with 8 more variables: Marktwert <dbl>, Nominalwert <dbl>, Sektor <chr>,
## #   ISIN <chr>, Börse <chr>, Standort <chr>, Marktwährung <chr>, Art <chr>
```

d)

```
colnames(etf_data) <- str_replace_all(colnames(etf_data),
                                      pattern= c("ö"= "oe", "ü"= "ue",
                                                 "ä"="ae","ß"= "ss", " "= "",
                                                 "\\("= "", "%"="","\\)"= ""))

colnames(etf_data)
```

```
##  [1] "ETF_ID"           "Emittententicker" "Name"             "Anlageklasse"
##  [5] "Gewichtung"       "Kurs"             "Nominale"         "Marktwert"
##  [9] "Nominalwert"      "Sektor"           "ISIN"             "Boerse"
## [13] "Standort"         "Marktwaehrung"    "Art"
```

e)

```
ETF_ID_Investment <- etf_overview %>%
  select(ETF_ID, Investment) %>%
  rename(., InvestmentPerETF = Investment ) #new_name = old_name

# Invested amount per Company for each ETF
etf_data <- left_join(etf_data, ETF_ID_Investment, by= "ETF_ID") %>%
  mutate(InvestmentPerCompany= (Gewichtung/100)*InvestmentPerETF)


etf_data <- etf_data %>%
  mutate(Investment= (ave(InvestmentPerCompany, ISIN, FUN=sum)))

etf_data %>%
  select(ETF_ID, Name, Investment)
```

```
## # A tibble: 9,099 x 3
##    ETF_ID Name                 Investment
##     <int> <chr>                     <dbl>
##  1      1 MICROSOFT CORP       109.634864
##  2      1 TESLA INC             74.44384
##  3      1 NVIDIA CORP           70.61136
##  4      1 HOME DEPOT INC        48.976464
##  5      1 WALT DISNEY           44.973384
##  6      1 ASML HOLDING NV       85.526616
##  7      1 ROCHE HOLDING PAR AG  81.618976
##  8      1 CISCO SYSTEMS INC     31.685848
##  9      1 COCA-COLA             31.344648
## 10      1 PEPSICO INC           29.04516
## # ... with 9,089 more rows
```

f)

```
etf_data <- etf_data %>%
  replace_na(list(Investment = 0))
```

g)

```
distinct_list <- list("ETF_ID"= vector(), "n_ISIN"= vector())
for(i in 1:length(unique(etf_data$ETF_ID))){
   distinct_list[[2]][i] <- etf_data %>%
     dplyr::filter(ETF_ID == i) %>%
     select(ISIN)%>%
     n_distinct()

  distinct_list[[1]][i] <- i
}

data.frame(distinct_list) %>%
  left_join(etf_overview[c("ETF", "ETF_ID")], by= "ETF_ID") %>%
  arrange(desc(.$n_ISIN))
```

```
##   ETF_ID n_ISIN                    ETF
## 1      3   3457   MSCI World Small Cap
## 2      7   2996            MSCI EM IMI
## 3      5   1569        Core MSCI World
## 4      1    382         MSCI World SRI
## 5      2    182            MSCI EM SRI
## 6      4    121        MSCI Europe SRI
## 7      6     92 STOXX Europe Small 200
## 8      8     83    Global Clean Energy
```

h)

```
  etf_data %>%
  dplyr::filter(nchar(ISIN)==12) %>%
  select(ISIN) %>%
  duplicated() %>%
  any()
```

```
## [1] TRUE
```

```
# Duplicated Values are in the dataset

full_isin_etf_data <- etf_data %>%
  dplyr::filter(nchar(ISIN)==12)

TotalInvestmentPerStock <- full_isin_etf_data %>%
  group_by(ISIN) %>%
  summarise("TotalInvestment"= sum(InvestmentPerCompany))

TotalInvestmentPerStock <- aggregate(
  list("TotalInvestment"=full_isin_etf_data$InvestmentPerCompany),
  by= list(ISIN= full_isin_etf_data$ISIN),
  sum)

TotalInvestmentPerStock %>%
    arrange(desc(TotalInvestment)) %>%
  head()
```

```
##           ISIN TotalInvestment
```

```
## 1 US5949181045       109.63486
## 2 TW0002330008        86.37294
## 3 NL0010273215        85.52662
## 4 CH0012032048        81.61898
## 5 KYG596691041        79.77766
## 6 US88160R1014        74.44384
```

```
left_join(TotalInvestmentPerStock, etf_data[c("Name","ISIN")], by= "ISIN") %>%
  arrange(desc(TotalInvestment)) %>%
  unique() %>% # Unique call because after join duplicated values
  head()
```

```
##            ISIN TotalInvestment                              Name
## 1  US5949181045       109.63486                 MICROSOFT CORP
## 3  TW0002330008        86.37294 TAIWAN SEMICONDUCTOR MANUFACTURING
## 5  NL0010273215        85.52662                 ASML HOLDING NV
## 8  CH0012032048        81.61898           ROCHE HOLDING PAR AG
## 11 KYG596691041        79.77766                        MEITUAN
## 13 US88160R1014        74.44384                    TESLA INC
```

i)

```
top_investments_ETF_ID <- etf_data %>%
  mutate("Proportion"= InvestmentPerCompany/InvestmentPerETF) %>%
  arrange(desc(.$Proportion)) %>%
  group_by(ETF_ID) %>%
  slice(1:3)

top_investments_ETF_ID %>%
  group_by(Standort) %>%
  count(Standort) %>%
  arrange(desc(.$n))
```

```
## # A tibble: 10 x 2
## # Groups:   Standort [10]
##    Standort               n
##    <chr>              <int>
##  1 Vereinigte Staaten    10
##  2 China                  4
##  3 Dänemark               2
##  4 Taiwan                 2
##  5 Belgien                1
##  6 Deutschland            1
##  7 Frankreich             1
##  8 Italien                1
##  9 Niederlande            1
## 10 Schweiz                1
```

```
top_investments_ETF_ID %>%
  group_by(Sektor) %>%
  count(Sektor) %>%
  arrange(desc(.$n))
```

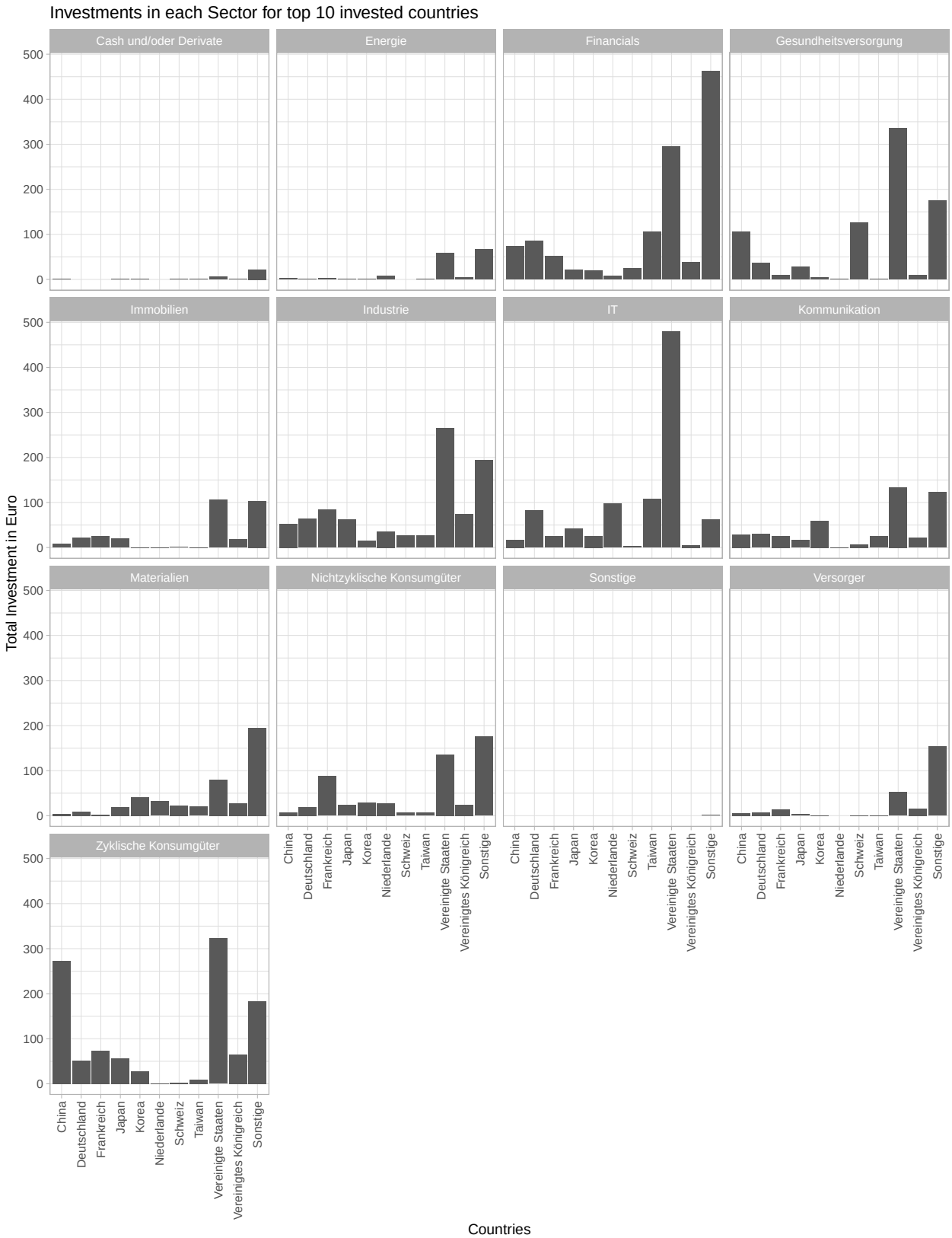
```
## # A tibble: 7 x 2
## # Groups:   Sektor [7]
```

```
##   Sektor                     n
##   <chr>                  <int>
## 1 IT                        11
## 2 Zyklische Konsumgüter      6
## 3 Gesundheitsversorgung      2
## 4 Kommunikation              2
## 5 Financials                 1
## 6 Industrie                  1
## 7 Versorger                  1
```

Die USA sind damit in diesen Daten (`top_investments_ETF_ID`) am häufigsten vertreten (`n=10`). Der Sektor IT ist mit `n= 11` Beobachtungen am häufigsten in diesen Daten vertreten.

j)

```
aggregated_etf_data <- etf_data %>%
  group_by(Standort, Sektor) %>%
  summarise(TotalInvestment= sum(InvestmentPerCompany)) %>%
  arrange(desc(TotalInvestment))


aggregated_etf_data[["Standort"]] <- as.factor(aggregated_etf_data[["Standort"]])


aggregated_etf_data <-  aggregated_etf_data %>%
  group_by(Standort) %>%
  mutate(StandortInvest= sum(TotalInvestment, na.rm = TRUE))


aggregated_etf_data[["Standort_lumped"]] <- fct_lump(aggregated_etf_data$Standort,
                                            n = 10,
                                            w = aggregated_etf_data$StandortInvest,
                                            other_level = "Sonstige")




ggplot(aggregated_etf_data, aes(x= Standort_lumped, y= TotalInvestment))+
  geom_col()+
  facet_wrap(~Sektor)+
  guides(x =  guide_axis(angle = 90))+
  theme_light()+
  ggtitle("Investments in each Sector for top 10 invested countries")+
  xlab("Countries")+
  ylab("Total Investment in Euro")
```


Investments in each Sector for top 10 invested countries
Cash und/oder Derivate
Energie
Financials
Gesundheitsversorgung
Immobilien
Industrie
IT
Kommunikation
Materialien
Nichtzyklische Konsumgüter
Sonstige
Versorger
Zyklische Konsumgüter
Total Investment in Euro
0
100
200
300
400
500
China
Deutschland
Frankreich
Japan
Korea
Niederlande
Schweiz
Taiwan
Vereinigte Staaten
Vereinigtes Königreich
Sonstige
Countries

Session Info

```
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS  10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] de_DE.UTF-8/de_DE.UTF-8/de_DE.UTF-8/C/de_DE.UTF-8/de_DE.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7     purrr_0.3.4
##  [5] readr_1.4.0     tidyr_1.1.3     tibble_3.1.1    ggplot2_3.3.3
##  [9] tidyverse_1.3.1 pammtools_0.5.7
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.6          lubridate_1.7.10    mvtnorm_1.1-1
##  [4] lattice_0.20-44     assertthat_0.2.1    digest_0.6.27
##  [7] foreach_1.5.1       utf8_1.2.1          R6_2.5.0
## [10] cellranger_1.1.0    backports_1.2.1     reprex_2.0.0
## [13] evaluate_0.14       highr_0.9           httr_1.4.2
## [16] pillar_1.6.0        rlang_0.4.11        lazyeval_0.2.2
## [19] readxl_1.3.1        rstudioapi_0.13     Matrix_1.3-3
## [22] checkmate_2.0.0     rmarkdown_2.8       labeling_0.4.2
## [25] splines_4.0.2       munsell_0.5.0       broom_0.7.6
## [28] compiler_4.0.2      numDeriv_2016.8-1.1 modelr_0.1.8
## [31] xfun_0.22           pkgconfig_2.0.3     mgcv_1.8-35
## [34] htmltools_0.5.1.1   tidyselect_1.1.1    prodlim_2019.11.13
## [37] codetools_0.2-18    fansi_0.4.2         withr_2.4.2
## [40] crayon_1.4.1        dbplyr_2.1.1        timereg_2.0.0
## [43] grid_4.0.2          nlme_3.1-152        jsonlite_1.7.2
## [46] gtable_0.3.0        lifecycle_1.0.0     DBI_1.1.1
## [49] magrittr_2.0.1      scales_1.1.1        cli_2.5.0
## [52] stringi_1.6.1       farver_2.1.0        fs_1.5.0
## [55] xml2_1.3.2          ellipsis_0.3.2      generics_0.1.0
## [58] vctrs_0.3.8         Formula_1.2-4       lava_1.6.9
## [61] iterators_1.0.13    tools_4.0.2         glue_1.4.2
## [64] hms_1.0.0           pec_2020.11.17      survival_3.2-11
## [67] yaml_2.2.1          colorspace_2.0-1    rvest_1.0.0
## [70] knitr_1.33          haven_2.4.1
```