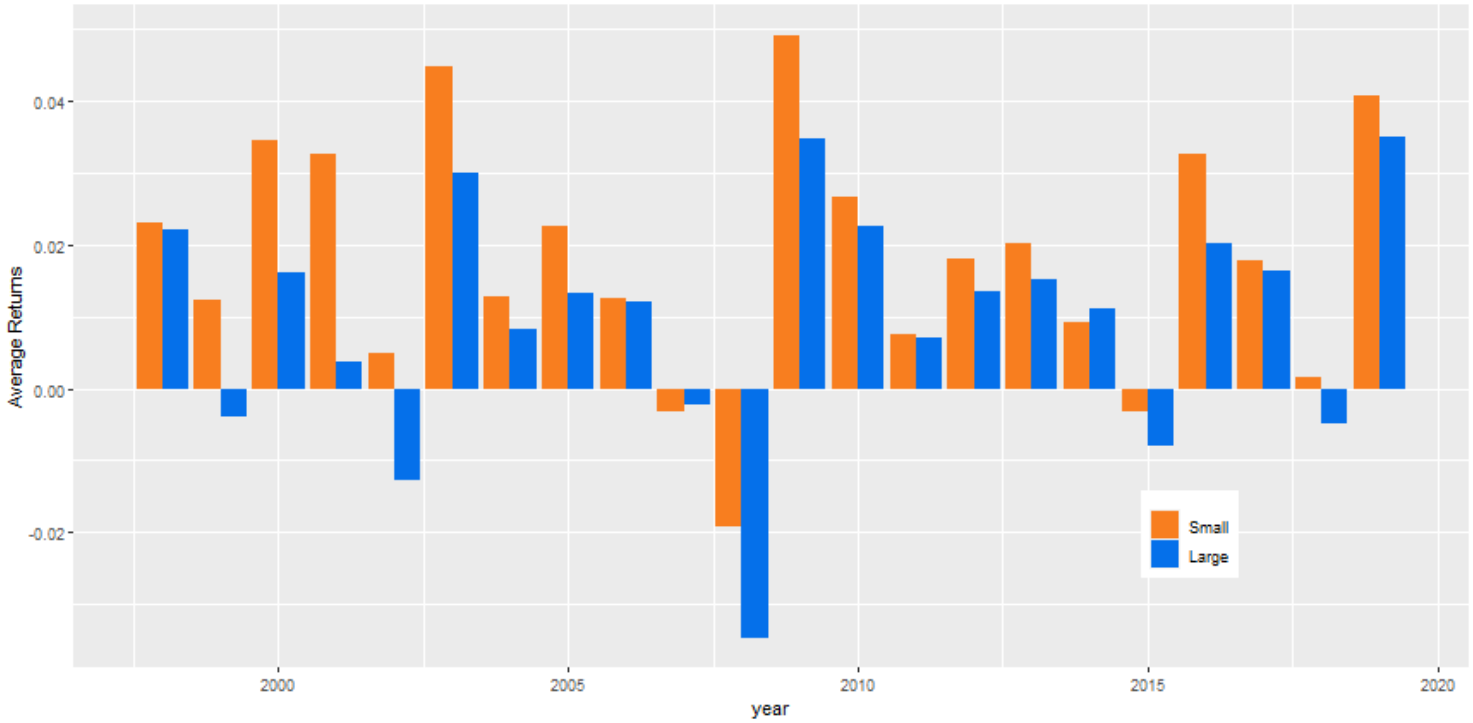
Datasets

```
load(here("Machine Learning for Factor Investing", "data_ml.RData"))
```

Example Factor, Size

```
data_ml %>%
   group_by(date) %>% # group by date
   mutate(large = Mkt_Cap_12M_Usd > median(Mkt_Cap_12M_Usd)) %>% # Creates the cap sort
   ungroup() %>% # ungroup
   mutate(year = lubridate::year(date)) %>% # Creates a gear variable
   group_by(year, large) %>% # Analyze by year & cap
   summarize(avg_return = mean(R1M_Usd)) %>% # avg return by year & cap
   ggplot(aes(x = year, y = avg_return, fill = large)) + # plot!
   geom_col(position = "dodge") + # bars side-to-side
   theme(legend.position = c(0.8, 0.2)) + # legend location
   coord_fixed(124) + # x/y aspect ration
   theme(legend.title = element_blank()) +
   scale_fill_manual(values = c("#F87E1F", "#0570EA"), name = "", # colors
                     labels = c("Small", "Large")) +
   ylab("Average Returns") +
   theme(legend.text = element_text(size=9))
```

```
`summarise()` regrouping output by 'year' (override with `.groups` argument)
```



Factors

Size

SMB = small firms minus large firms

Value

HM = high minus low: undervalued minus 'growth' firms

Momentum

WML winners minus losers

Profitability

RMW = robust minus weak profits

profitability is measured as (revenues - (cost and expenses)) / equity

Investment

CMA conservative minus aggressive

Investment is measured via the growth of total assets (divided by total assets).

Low 'risk'

BAB betting against beta

(simple vol, market beta, idiosyncratic vol, etc)

Kenneth French Factor Library

Example Factor Model

```
min_date <- "1963-07-31"; max_date <- "2020-06-30"

temp <- tempfile()

KF_website <- "http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/"
KF_file <- "ftp/F-F_Research_Data_5_Factors_2x3_CSV.zip"

link <- paste0(KF_website, KF_file)

download.file(link, temp, quiet = T)

FF_factors <- read_csv(unz(temp, "F-F_Research_Data_5_Factors_2x3.CSV"),
                       skip = 3) %>%          # Check the number of lines to skip!
    rename(date = X1, MKT_RF = `Mkt-RF`) %>%  # Change the name of first columns
    mutate_at(vars(-date), as.numeric) %>%                # Convert values to number
    mutate(date = ymd(parse_date_time(date, "%Y%m"))) %>%  # Date in right format
    mutate(date = rollback(date + months(1)))              # End of month date
```

```
Warning: Missing column names filled in: 'X1' [1]
```

```
-- Column specification --------------------------------------------------------
cols(
  X1 = col_character(),
  `Mkt-RF` = col_character(),
```

```
  SMB = col_character(),
  HML = col_character(),
  RMW = col_character(),
  CMA = col_character(),
  RF = col_character()
)

Warning: 1 parsing failure.
row col  expected    actual         file
693  -- 7 columns 1 columns <connection>

Warning: Problem with `mutate()` input `MKT_RF`.
i NAs introduced by coercion
i Input `MKT_RF` is `.Primitive("as.double")(MKT_RF)`.

Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion

Warning: Problem with `mutate()` input `SMB`.
i NAs introduced by coercion
i Input `SMB` is `.Primitive("as.double")(SMB)`.

Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion

Warning: Problem with `mutate()` input `HML`.
i NAs introduced by coercion
i Input `HML` is `.Primitive("as.double")(HML)`.

Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion

Warning: Problem with `mutate()` input `RMW`.
i NAs introduced by coercion
i Input `RMW` is `.Primitive("as.double")(RMW)`.

Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion

Warning: Problem with `mutate()` input `CMA`.
i NAs introduced by coercion
i Input `CMA` is `.Primitive("as.double")(CMA)`.

Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion

Warning: Problem with `mutate()` input `RF`.
i NAs introduced by coercion
i Input `RF` is `.Primitive("as.double")(RF)`.

Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion

Warning: Problem with `mutate()` input `date`.
i  58 failed to parse.
i Input `date` is `ymd(parse_date_time(date, "%Y%m"))`.

Warning: 58 failed to parse.
```

Table 1: Sample of Monthly Factor Returns.

date	MKT_RF	SMB	HML	RMW	CMA	RF
1963-07-31	-0.0039	-0.0045	-0.0094	0.0066	-0.0115	0.0027
1963-08-31	0.0507	-0.0082	0.0182	0.0040	-0.0040	0.0025
1963-09-30	-0.0157	-0.0048	0.0017	-0.0076	0.0024	0.0027
1963-10-31	0.0253	-0.0130	-0.0004	0.0275	-0.0224	0.0029
1963-11-30	-0.0085	-0.0085	0.0170	-0.0045	0.0222	0.0027
1963-12-31	0.0183	-0.0190	-0.0006	0.0007	-0.0030	0.0029

```
FF_factors <- FF_factors %>% mutate(MKT_RF = MKT_RF / 100,
                                    SMB = SMB / 100,
                                    HML = HML / 100,
                                    RMW = RMW / 100,
                                    CMA = CMA / 100,
                                    RF = RF / 100) %>%
   filter(date >= min_date, date <= max_date)

knitr::kable(head(FF_factors), booktabs = T,
             caption = "Sample of Monthly Factor Returns.")
```

```
FF_Avg_Returns <- FF_factors %>%
   mutate(date = year(date)) %>%
   gather(key = factor, value = value, - date) %>%
   group_by(date, factor) %>%
   summarise(value = mean(value))
```

```
`summarise()` regrouping output by 'date' (override with `.groups` argument)
```

```
FF_Avg_Returns %>%
   ggplot(aes(x = date, y = value, color = factor)) +
   geom_line() + coord_fixed(500)
```

```
FF_factors %>%
   gather(key = factor, value = return, - date) %>%
   filter(factor != 'RF') %>%
   ggplot(aes(return, group = factor)) +
      geom_density(aes(fill = factor, alpha = .25))
```

```
FF_Avg_Returns %>%
   filter(factor != 'RF') %>%
   ggplot(aes(value, group = factor)) +
      geom_density(aes(fill = factor, alpha = .25))
```

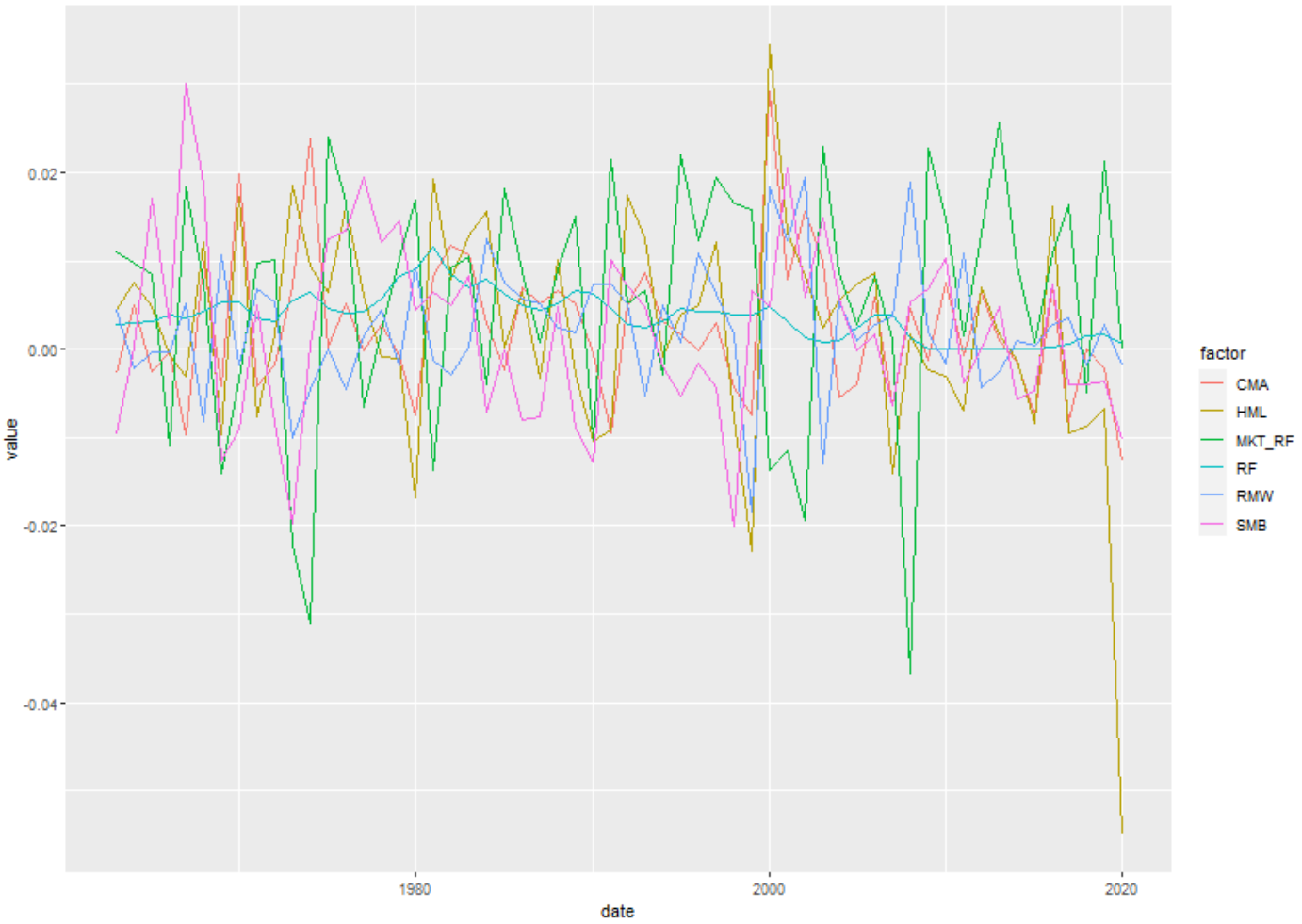


Figure 1: Factor Returns

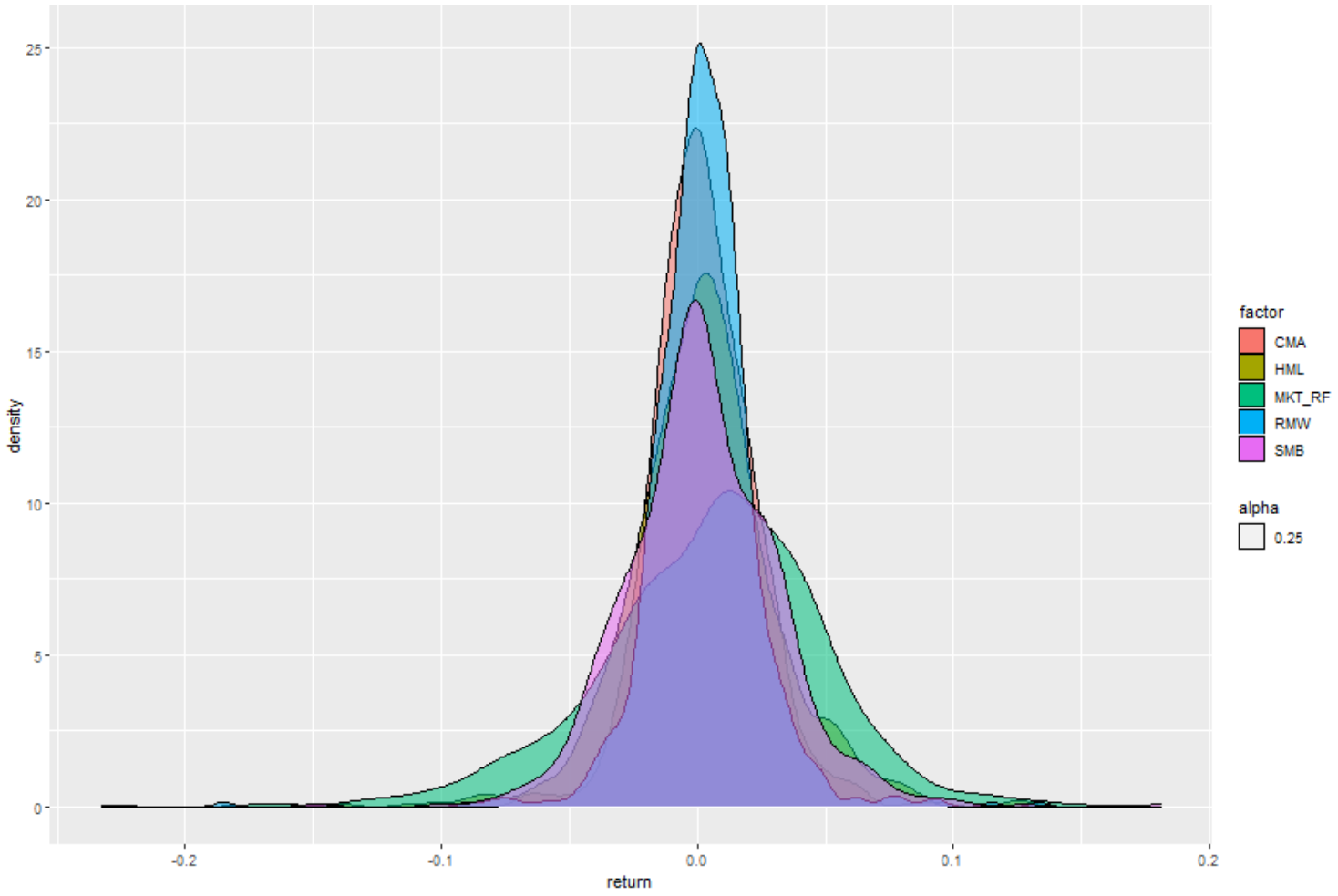


Figure 2: Return Densitites

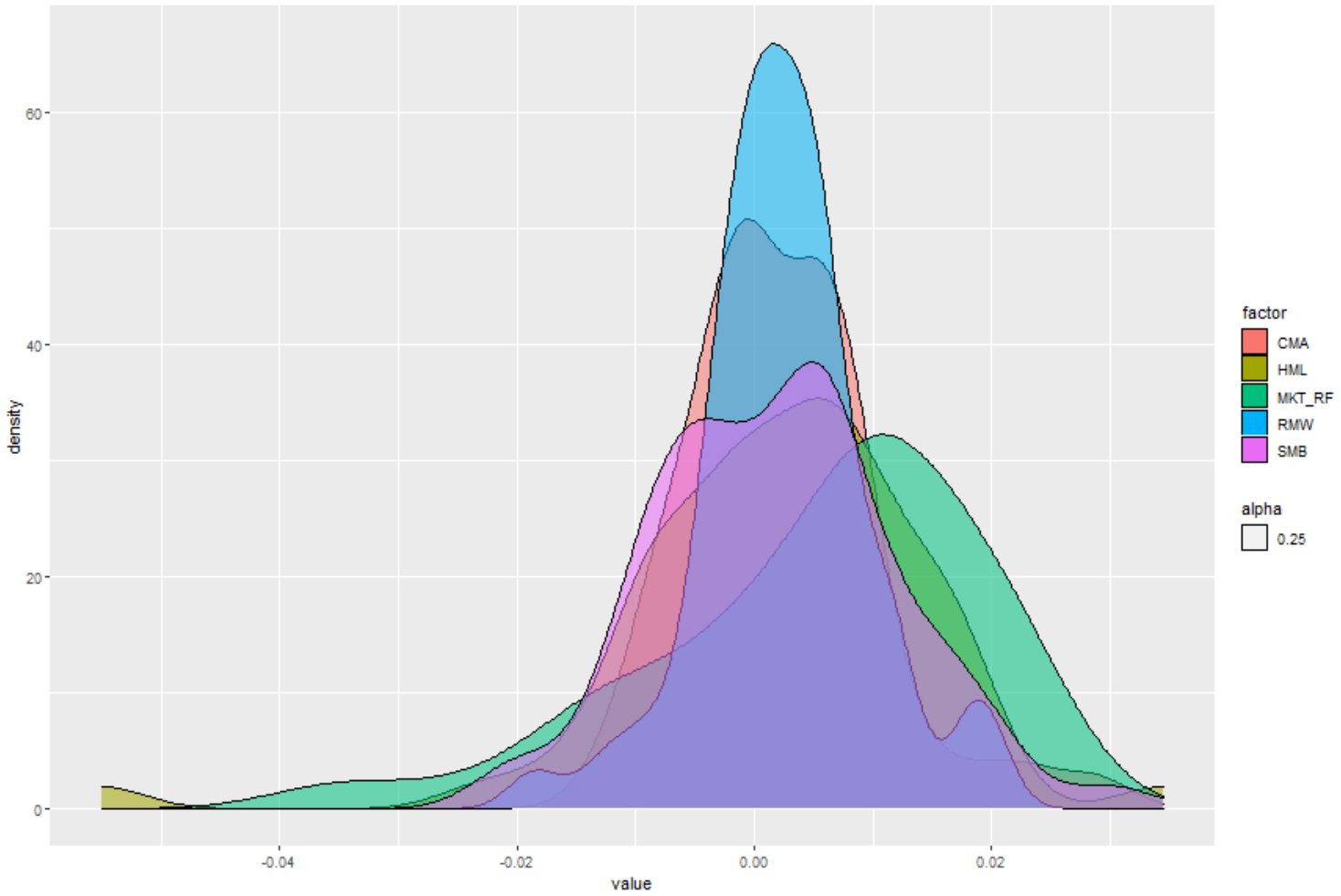


Figure 3: Yearly Avg Return Densities

```
FF_Cum_Returns <- FF_factors %>%
   gather(key = factor, value = value, -date) %>%
   group_by(factor) %>%
   mutate(lag_ret = lag(value)) %>%
   mutate(return = cumprod(1 + ifelse(is.na(lag_ret), 0, lag_ret)))

FF_Cum_Returns %>%
   ggplot(aes(date, return, group = factor)) +
   geom_line(aes(col = factor))
```

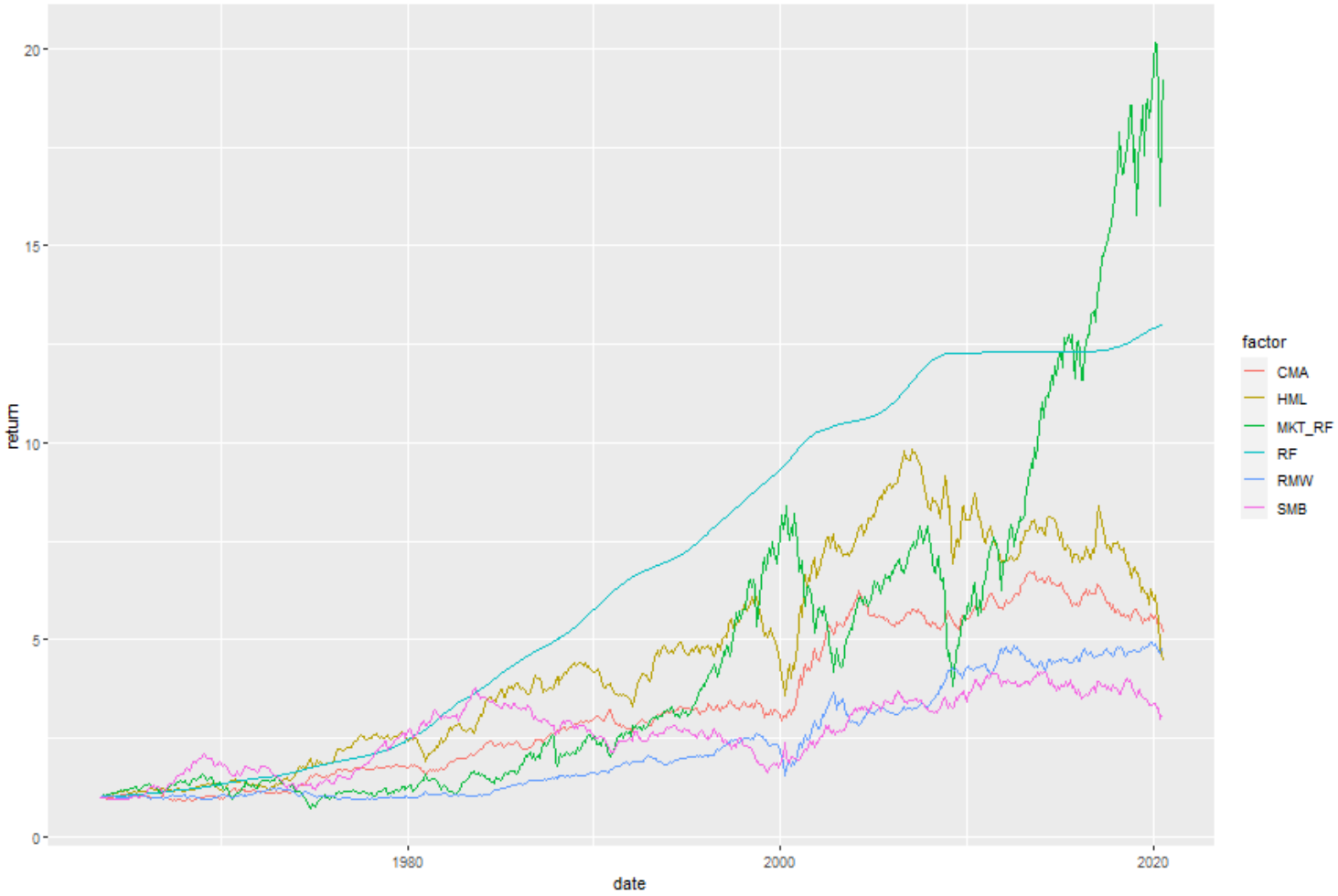


Figure 4: Growth of $1 by factor

Fama-Macbeth Regressions

```
separation_date <- as_date("2014-01-15")

traning_sample <- filter(data_ml, date < separation_date)
testing_sample <- filter(data_ml, date > separation_date)

stock_ids <- levels(as.factor(data_ml$stock_id)) # list of all stock ids

stock_days <- data_ml %>%
   group_by(stock_id) %>%
   summarise(nb = n())
```

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

```
stock_ids_short <- stock_ids[which(stock_days$nb == max(stock_days$nb))] # keep only stocks wi
```

```
# single stock example

stock_identifer <- 3

data_ml %>%
   filter(date == as_date("2006-06-30") & stock_id == 3)
```

```
# A tibble: 1 x 99
  stock_id date       Advt_12M_Usd Advt_3M_Usd Advt_6M_Usd Asset_Turnover Bb_Yld
     <int> <date>            <dbl>       <dbl>       <dbl>          <dbl>  <dbl>
1        3 2006-06-30         0.08        0.08        0.09           0.04   0.78
# ... with 92 more variables: Bv <dbl>, Capex_Ps_Cf <dbl>, Capex_Sales <dbl>,
#   Cash_Div_Cf <dbl>, Cash_Per_Share <dbl>, Cf_Sales <dbl>, Debtequity <dbl>,
#   Div_Yld <dbl>, Dps <dbl>, Ebit_Bv <dbl>, Ebit_Noa <dbl>, Ebit_Oa <dbl>,
#   Ebit_Ta <dbl>, Ebitda_Margin <dbl>, Eps <dbl>, Eps_Basic <dbl>,
#   Eps_Basic_Gr <dbl>, Eps_Contin_Oper <dbl>, Eps_Dil <dbl>, Ev <dbl>,
#   Ev_Ebitda <dbl>, Fa_Ci <dbl>, Fcf <dbl>, Fcf_Bv <dbl>, Fcf_Ce <dbl>,
#   Fcf_Margin <dbl>, Fcf_Noa <dbl>, Fcf_Oa <dbl>, Fcf_Ta <dbl>, Fcf_Tbv <dbl>,
#   Fcf_Toa <dbl>, Fcf_Yld <dbl>, Free_Ps_Cf <dbl>, Int_Rev <dbl>,
#   Interest_Expense <dbl>, Mkt_Cap_12M_Usd <dbl>, Mkt_Cap_3M_Usd <dbl>,
#   Mkt_Cap_6M_Usd <dbl>, Mom_11M_Usd <dbl>, Mom_5M_Usd <dbl>,
#   Mom_Sharp_11M_Usd <dbl>, Mom_Sharp_5M_Usd <dbl>, Nd_Ebitda <dbl>,
#   Net_Debt <dbl>, Net_Debt_Cf <dbl>, Net_Margin <dbl>, Netdebtyield <dbl>,
#   Ni <dbl>, Ni_Avail_Margin <dbl>, Ni_Oa <dbl>, Ni_Toa <dbl>, Noa <dbl>,
#   Oa <dbl>, Ocf <dbl>, Ocf_Bv <dbl>, Ocf_Ce <dbl>, Ocf_Margin <dbl>,
#   Ocf_Noa <dbl>, Ocf_Oa <dbl>, Ocf_Ta <dbl>, Ocf_Tbv <dbl>, Ocf_Toa <dbl>,
#   Op_Margin <dbl>, Op_Prt_Margin <dbl>, Oper_Ps_Net_Cf <dbl>, Pb <dbl>,
#   Pe <dbl>, Ptx_Mgn <dbl>, Recurring_Earning_Total_Assets <dbl>,
#   Return_On_Capital <dbl>, Rev <dbl>, Roa <dbl>, Roc <dbl>, Roce <dbl>,
```

```
#   Roe <dbl>, Sales_Ps <dbl>, Share_Turn_12M <dbl>, Share_Turn_3M <dbl>,
#   Share_Turn_6M <dbl>, Ta <dbl>, Tev_Less_Mktcap <dbl>, Tot_Debt_Rev <dbl>,
#   Total_Capital <dbl>, Total_Debt <dbl>, Total_Debt_Capital <dbl>,
#   Total_Liabilities_Total_Assets <dbl>, Vol1Y_Usd <dbl>, Vol3Y_Usd <dbl>,
#   R1M_Usd <dbl>, R3M_Usd <dbl>, R6M_Usd <dbl>, R12M_Usd <dbl>
```

```
stock_returns <- data_ml %>%
                  filter(stock_id == stock_identifer) %>%
                  select(date, stock_id, Return = R1M_Usd) %>%
                  group_by(stock_id) %>%
                  mutate(Return = lag(Return)) %>%
                  ungroup()

stock_returns %>%
   filter(date == as_date("2006-06-30") & stock_id == 3)
```

```
# A tibble: 1 x 3
  date       stock_id Return
  <date>        <int>  <dbl>
1 2006-06-30        3     NA
```

```
factor_data <- left_join(stock_returns, FF_factors, by = "date") %>%
               select(date, stock_id, MKT_RF, SMB, HML, RMW, CMA, RF, Return)

factor_loading <-
   coef(summary(lm(formula = "Return ~ MKT_RF + SMB + HML + RMW + CMA", data = factor_data))) %
   as.data.frame() %>%
   select(Value = Estimate) %>%
   rownames_to_column("Factor") %>%
   mutate(stock_id = stock_identifer) %>%
   spread(key = "Factor", value = "Value") %>%
   select(MKT_RF, SMB, HML, RMW, CMA)

factor_data <- cbind(factor_loading, stock_returns) %>%
   filter(!is.na(Return))
```

```
nb_factors <- 5                                                    # Number of factors

data_FM <- left_join(data_ml %>%                                   # Join the 2 datasets
                        dplyr::select(date, stock_id, R1M_Usd) %>% # (with returns...
                        filter(stock_id %in% stock_ids_short),     # ... over some stocks)
                     FF_factors,
                     by = "date") %>%
    group_by(stock_id) %>%                                         # Grouping
    mutate(R1M_Usd = lag(R1M_Usd)) %>%                             # Lag returns
    ungroup() %>%
```

Table 2: Coefficents

	Constant	MKT_RF	SMB	HML	RMW	CMA
3	-0.0017438	0.8092717	0.8280240	0.8479085	0.1198440	-0.2522913
4	0.0037247	0.3073030	0.2619254	-0.1392922	0.4354316	0.4485567
7	0.0050755	0.5203728	0.5246247	0.0376542	0.3136473	0.3198131
9	0.0044285	0.7516452	0.6174593	1.0164648	-0.0597775	-0.0562163
16	0.0010675	1.1996284	-0.1769292	1.3980331	0.1910664	-0.6164365
22	0.0019074	0.5925792	0.5670595	0.3446145	0.4738824	0.1648655

```
    na.omit() %>%                                                  # Remove missing points
    spread(key = stock_id, value = R1M_Usd)

models <- lapply(paste0("`", stock_ids_short,
                        '` ~  MKT_RF + SMB + HML + RMW + CMA'),           # Model spec
                 function(f){ lm(as.formula(f), data = data_FM,           # Call lm(.)
                                 na.action="na.exclude") %>%
                        summary() %>%                                     # Gather the output
                        "$"(coef) %>%                                     # Keep only coefs
                        data.frame() %>%                                  # Convert to datafra
                        dplyr::select(Estimate)}                          # Keep the estimates
                 )

betas <- matrix(unlist(models), ncol = nb_factors + 1, byrow = T) %>%     # Extract the betas
    data.frame(row.names = stock_ids_short)                               # Format: row names

stopifnot(nrow(betas) == length(stock_ids_short))

colnames(betas) <- c("Constant", "MKT_RF", "SMB", "HML", "RMW", "CMA")    # Format: col names
```

```
knitr::kable(head(betas), caption = "Coefficents")
```

```
factor_loadings <- betas %>%
   dplyr::select(-Constant) %>%
   data.frame()

stock_returns <- data_FM %>%
   dplyr::select(-MKT_RF, -SMB, -HML, -RMW, -CMA, -RF)

factor_returns <- stock_returns %>%
   dplyr::select(-date) %>%
   data.frame(row.names = stock_returns$date) %>%
   t()
```

```
stopifnot(nrow(factor_returns) == nrow(factor_loadings))

FM_data <- cbind(factor_loadings, factor_returns)
```

```
models <- lapply(paste("`",
                       stock_returns$date, "`",
                       ' ~  MKT_RF + SMB + HML + RMW + CMA', sep = ""),
                          function(f){ lm(as.formula(f), data = FM_data) %>%  # Call lm(.)
                             summary() %>%                                    # Gather the ou
                             "$"(coef) %>%                                    # Keep only the
                             data.frame() %>%                                 # Convert to da
                             dplyr::select(Estimate)}                         # Keep only est
                 )

gammas <- matrix(unlist(models), ncol = nb_factors + 1, byrow = T) %>%    # Switch to datafram
    data.frame(row.names = stock_returns$date)                                  # & set row na
colnames(gammas) <- c("Constant", "MKT_RF", "SMB", "HML", "RMW", "CMA")   # Set col names
```

```
gammas %>%                                                          # Take gammas:
    # The first row is omitted because the first row of returns is undefined
    dplyr::select(MKT_RF, SMB, HML) %>%                             # Select 3 factors
    bind_cols(date = data_FM$date) %>%                              # Add date
    gather(key = factor, value = gamma, -date) %>%                  # Put in tidy shape
    ggplot(aes(x = date, y = gamma, color = factor)) +              # Plot
    geom_line() + facet_grid( factor~. ) +                          # Lines & facets
    scale_color_manual(values=c("#F87E1F", "#0570EA", "#F81F40"))  # Colors
```

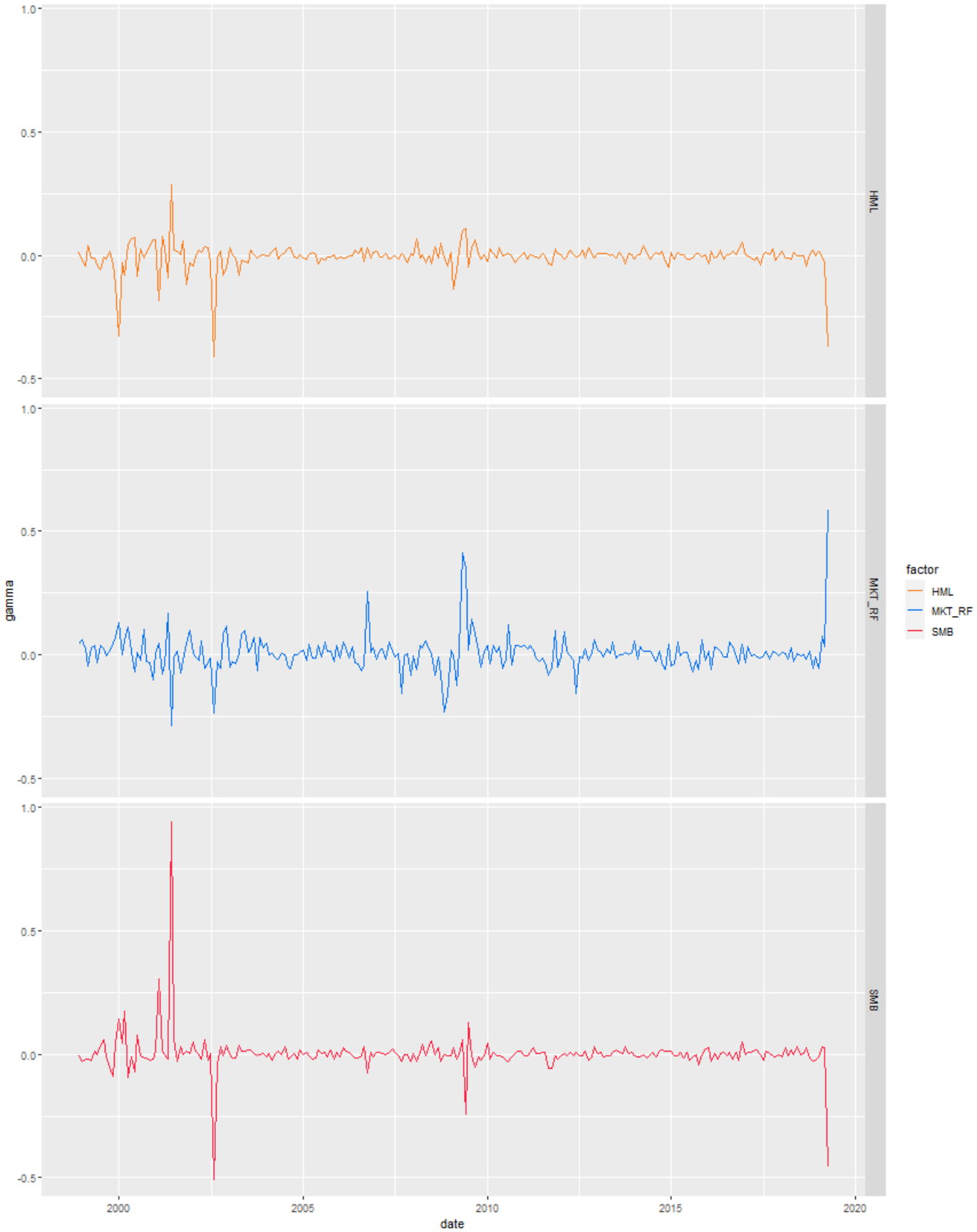


Figure 5: Gammas

Factor Competition

```
factors <- c("MKT_RF", "SMB", "HML", "RMW", "CMA")

models <- lapply(paste(factors, ' ~  MKT_RF + SMB + HML + RMW + CMA-',factors),
 function(f){ lm(as.formula(f), data = FF_factors) %>%                # Call lm(.)
                          summary() %>%                               # Gather the output
                          "$"(coef) %>%                               # Keep only the coefs
                          data.frame() %>%                            # Convert to dataframe
                          filter(rownames(.) == "(Intercept)") %>%    # Keep only the Intercept
                          dplyr::select(Estimate,`Pr...t..`)}         # Keep the coef & p-value
                  )
alphas <- matrix(unlist(models), ncol = 2, byrow = T) %>%       # Switch from list to datafram
    data.frame(row.names = factors)
# alphas # To see the alphas (optional)
```

```
results <- matrix(NA, nrow = length(factors), ncol = length(factors) + 1)    # Coefs
signif  <- matrix(NA, nrow = length(factors), ncol = length(factors) + 1)    # p-values

for(j in 1:length(factors)){
    form <- paste(factors[j],
                  ' ~  MKT_RF + SMB + HML + RMW + CMA-',factors[j])          # Build model
    fit <- lm(form, data = FF_factors) %>% summary()                         # Estimate model
    coef <- fit$coefficients[,1]                                             # Keep coefficients
    p_val <- fit$coefficients[,4]                                            # Keep p-values
    results[j,-(j+1)] <- coef                                                # Fill matrix
    signif[j,-(j+1)] <- p_val
}

signif[is.na(signif)] <- 1                                                   # Kick out NAs

results <- results %>% round(3)  %>% data.frame()                            # Basic formatting

results[signif<0.001] <- paste(results[signif<0.001]," (***)")               # 3 star signif

results[signif>0.001&signif<0.01] <-                                         # 2 star signif
    paste(results[signif>0.001&signif<0.01]," (**)")

results[signif>0.01&signif<0.05] <-                                          # 1 star signif
    paste(results[signif>0.01&signif<0.05]," (*)")

results <- cbind(factors, results)                                 # Add dep. variable

colnames(results) <- c("Dep. Variable", "Intercept", factors)                # Add column name
```

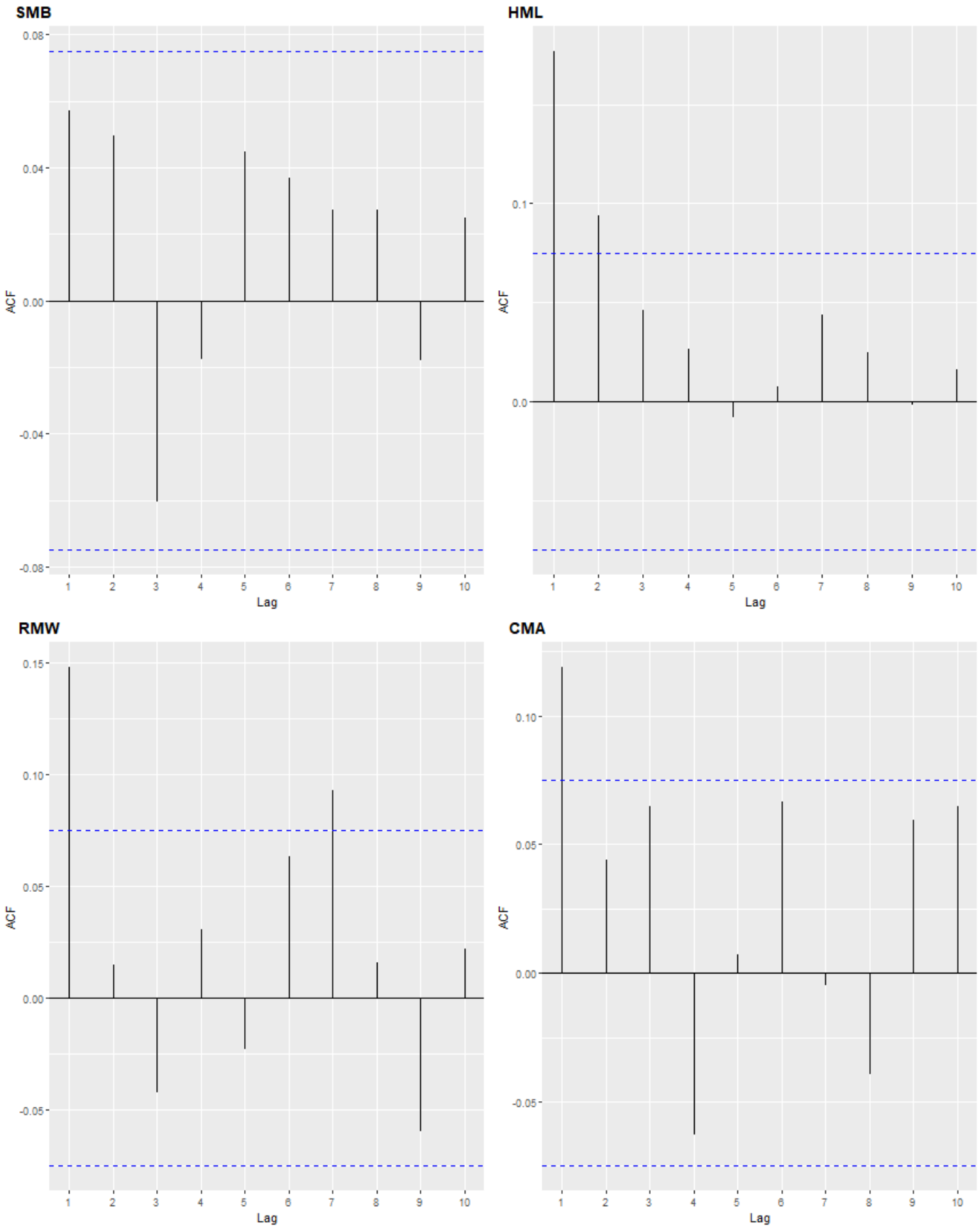
Table 3: Factor competition among the Fama and French (2015) five factors.

Dep. Variable	Intercept	MKT_RF	SMB	HML	RMW	CMA
MKT_RF	0.008 (***)	NA	0.287 (***)	0.143 (*)	-0.326 (***)	-0.951 (***)
SMB	0.003 (*)	0.143 (***)	NA	0.104 (*)	-0.423 (***)	-0.149
HML	-0.001	0.04 (*)	0.059 (*)	NA	0.172 (***)	1.027 (***)
RMW	0.004 (***)	-0.084 (***)	-0.22 (***)	0.158 (***)	NA	-0.286 (***)
CMA	0.003 (***)	-0.115 (***)	-0.036	0.441 (***)	-0.133 (***)	NA

Momentum, timing and ESG

```
acf_SMB <- ggAcf(FF_factors$SMB, lag.max = 10) + labs(title = "")
acf_HML <- ggAcf(FF_factors$HML, lag.max = 10) + labs(title = "")
acf_RMW <- ggAcf(FF_factors$RMW, lag.max = 10) + labs(title = "")
acf_CMA <- ggAcf(FF_factors$CMA, lag.max = 10) + labs(title = "")

plot_grid(acf_SMB, acf_HML, acf_RMW, acf_CMA, # plot
          labels = c("SMB", "HML", "RMW", "CMA"))
```

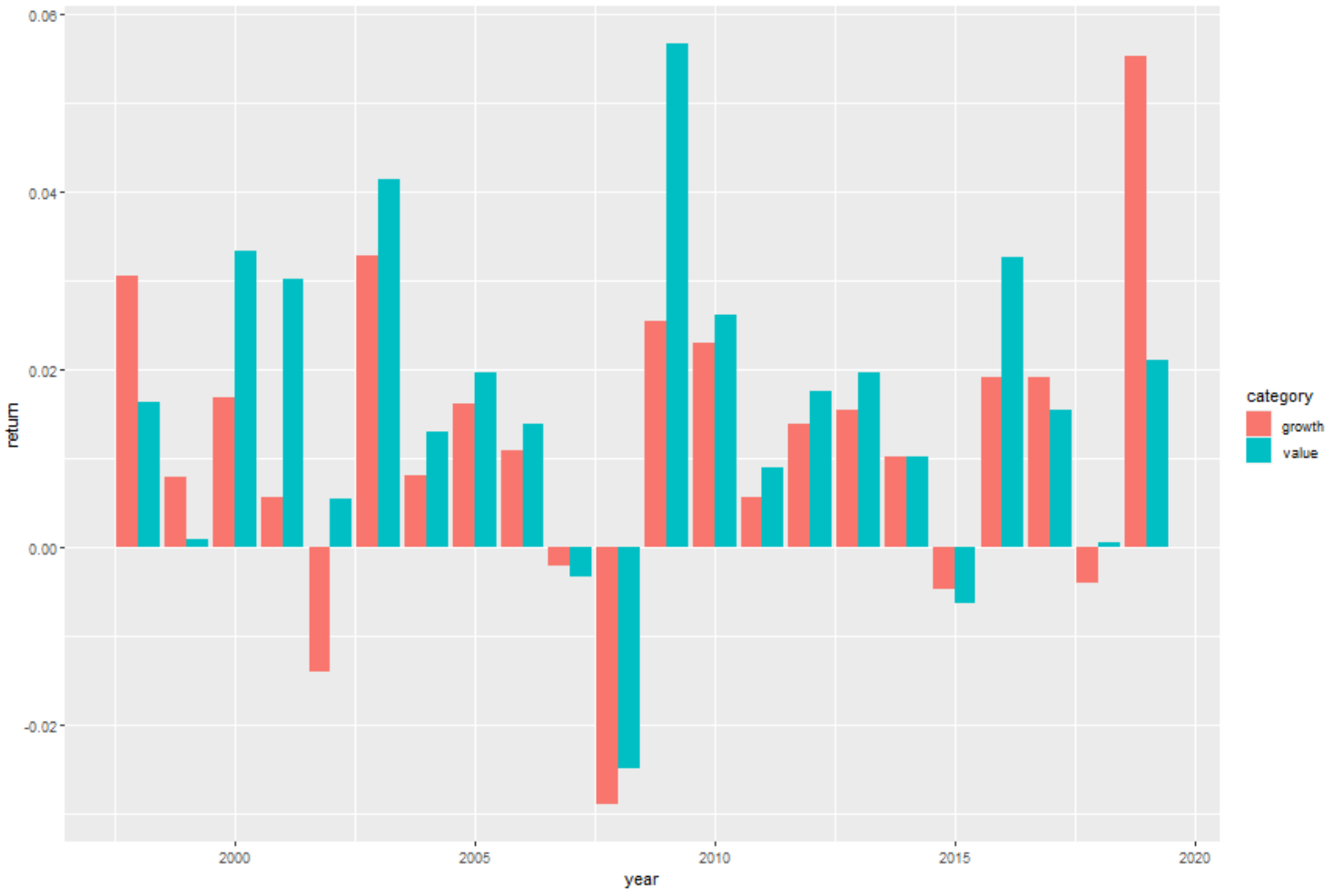

SMB
HML
RMW
CMA
ACF
Lag

Exercises

1.) Compute annual returns of the growth vs value portfolios, that is, the average return of firms with above median price-to-book ratio (*Pb*).

```
data_ml %>%
  group_by(date) %>%
  mutate(category = ifelse(Pb > median(Pb), "growth", "value")) %>%
  ungroup() %>%
  mutate(year = lubridate::year(date)) %>%
  group_by(year, category) %>%
  summarise(return = mean(R1M_Usd)) %>%
  ggplot(aes(year, return, fill = category)) +
    geom_col(position = "dodge")
```

```
`summarise()` regrouping output by 'year' (override with `.groups` argument)
```



2.) Same exercise, but compare the monthly returns and plot the value (through time) of the corresponding portfolios.

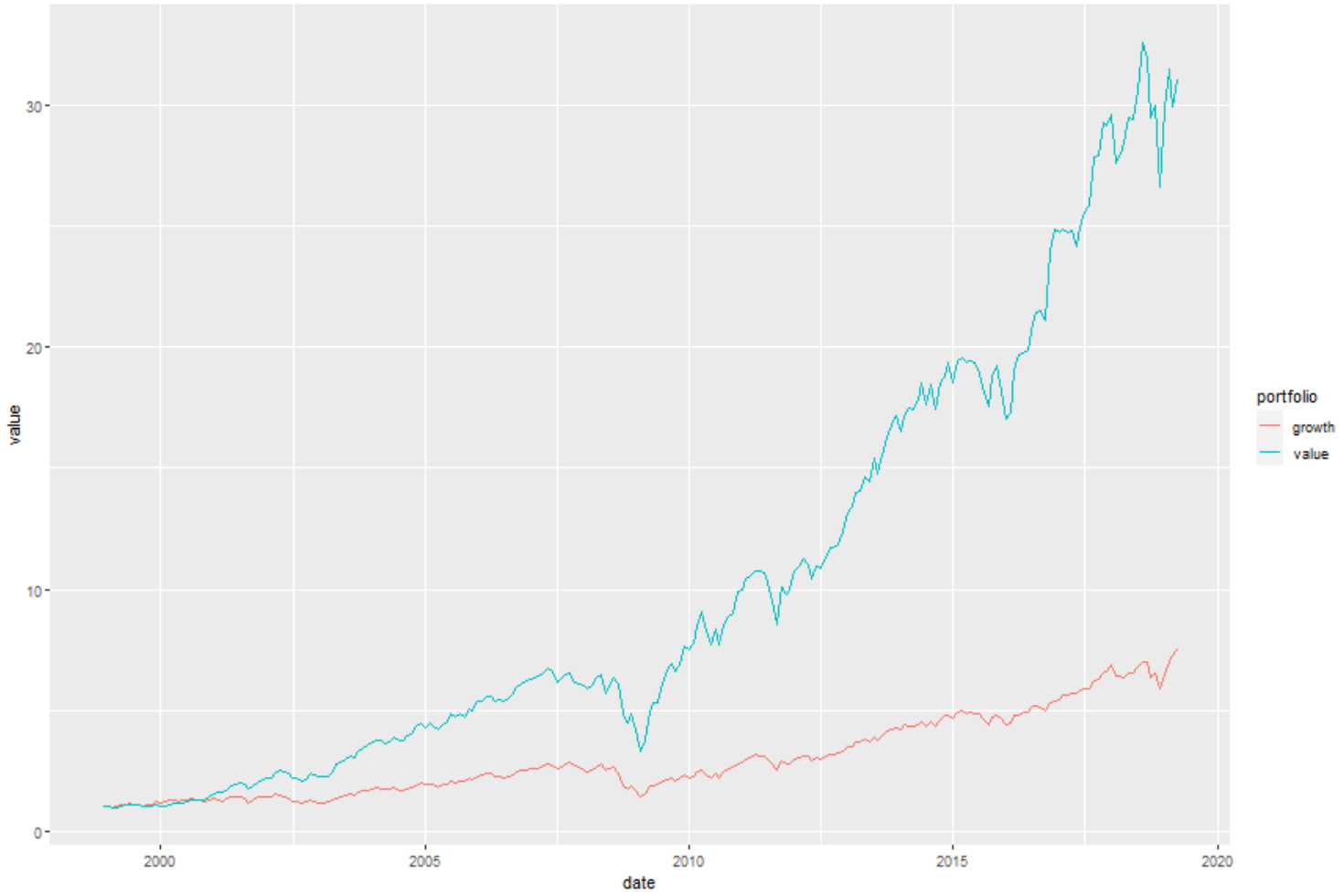
```
monthly_return <- data_ml %>%
  group_by(date) %>%
  mutate(growth = Pb > median(Pb)) %>%
  ungroup() %>%
  group_by(date, growth) %>%
  summarise(return = mean(R1M_Usd)) %>%
  spread(key = growth, value = return) %>%
  ungroup()
```

```
`summarise()` regrouping output by 'date' (override with `.groups` argument)
```

```
colnames(monthly_return)[2:3] <- c("value", "growth")

monthly_return %>%
```

```
  mutate(growth = cumprod(1 + growth), value = cumprod(1 + value)) %>%
  gather(key = portfolio, value = value, -date) %>%
  ggplot(aes(x = date, y = value, color = portfolio)) +
    geom_line()
```



3.) Instead of a unique threshold, compute simply sorted portfolios based on quartiles of market capitalization.

Compute their annual returns and plot them.

```
data_ml %>%
  group_by(date) %>%
  mutate(capitalization = ntile(Mkt_Cap_3M_Usd, 4)) %>%
  ungroup() %>%
  mutate(year = lubridate::year(date)) %>%
  group_by(year, capitalization) %>%
  summarise(return = mean(R1M_Usd)) %>%
```

```
  mutate(capitalization = factor(capitalization, levels = 1:4, labels = c("small", "medium", "l
  ggplot(aes(year, return, fill = capitalization)) +
    geom_col(position = "dodge")
```

```
`summarise()` regrouping output by 'year' (override with `.groups` argument)
```