CAPM aplicado.

Beta, costo de capital y alpha.

Dr. Martín Lozano `https://mlozanoqf.github.io/`

04 de febrero de 2026, 11:38 p.m.

	Fundamental	Intermedio	Especializado
Finanzas	×	✓	×
Estadística	×	✓	×
R	×	✓	×

1 Introducción.

- En este video el CAPM se presenta como una herramienta operativa para decisiones financieras. El propósito no es contrastar el CAPM como teoría de equilibrio, sino obtener insumos cuantitativos con interpretación clara para análisis aplicado.
- Trabajamos con rendimientos en exceso definidos como $r_{i,t} = R_{i,t} - R_{f,t}$ para el activo o fondo y $r_{M,t} = R_{M,t} - R_{f,t}$ para el mercado. Esta notación coincide con el video académico y mantiene consistencia con la teoría financiera.
- El benchmark de mercado se aproxima con el índice S&P 500, usando su serie de niveles (`^GSPC`) para construir rendimientos mensuales. En consecuencia, el rendimiento del mercado corresponde a un índice de precio; esta elección se adopta por disponibilidad y transparencia, y se mantiene fija a lo largo del video.
- El video se organiza en tres aplicaciones basadas en la misma regresión de serie de tiempo en exceso,

$$r_{i,t} = \alpha_i + \beta_i r_{M,t} + \varepsilon_{i,t},$$

  con objetivos distintos: (1) estimar $\hat{\beta}$ para cuantificar exposición al mercado; (2) derivar un rendimiento requerido del equity $k_e \approx R_f + \hat{\beta} \cdot ERP$; y (3) evaluar desempeño de un fondo con el alpha de Jensen $\hat{\alpha}$.
- A diferencia del video académico, aquí no se realizan estimaciones en sección cruzada ni Fama–MacBeth, ya que esas técnicas se orientan a estimar el precio del riesgo y evaluar predicciones del modelo en un conjunto de activos. En este video el énfasis es el uso práctico de $\hat{\beta}$, k_e y $\hat{\alpha}$ como insumos para decisión, con supuestos explícitos.
- La muestra es mensual y utiliza una ventana de 120 observaciones hasta diciembre de 2025. Se enfatiza que β y α dependen de ventana, frecuencia y benchmark, por lo que su interpretación debe entenderse como condicional al diseño empírico adoptado.

2 Paquetes.

```r
library(tidyverse)
library(lubridate)
library(tidyquant)
library(broom)
library(lmtest)
library(sandwich)
```

3 Datos 1/2.

- Descargamos niveles de precios desde Yahoo Finance con `tidyquant` para el índice S&P 500 (`^GSPC`) y para los instrumentos de las tres aplicaciones.
- Convertimos niveles a rendimientos mensuales usando precios ajustados y rendimientos aritméticos.
- Para la tasa libre de riesgo usamos una serie mensual de `FRED` (`TB3MS`), que es una tasa anualizada; la transformamos a rendimiento mensual efectivo para construir $R_{f,t}$ en la misma frecuencia.
- La ventana se fija para obtener 120 rendimientos mensuales hasta diciembre de 2025.

```r
# Datos. Descarga y construcción (solo tidyquant, mensual, hasta 2025-12)
library(tidyverse)
library(lubridate)
library(tidyquant)
library(readr)

# ----------------------------
# Parámetros de muestra
# ----------------------------
end_date   <- as.Date("2025-12-31")
start_date <- end_date %m-% months(120) %m-% months(3)

# Instrumentos
ticker_mkt    <- "^GSPC"
tickers_asset <- c("AAPL","KO","ARKK")

# ----------------------------
# Helper robusto: asegurar clase Date
# ----------------------------
fix_date <- function(x) {
  if (inherits(x, "Date")) return(x)
  if (inherits(x, "POSIXct") || inherits(x, "POSIXt")) return(as.Date(x))
  if (is.numeric(x)) return(as.Date(x, origin = "1970-01-01"))
```

```r
  as.Date(as.character(x))
}

# ----------------------------
# Rendimientos mensuales desde Yahoo (tidyquant)
# Fecha alineada al 1er día del mes
# ----------------------------
get_monthly_returns_yahoo <- function(tickers, start, end) {

  px <- tryCatch(
    tq_get(tickers, get = "stock.prices", from = start, to = end),
    error = function(e) NULL
  )

  if (is.null(px) || is.logical(px) || !is.data.frame(px) || nrow(px) == 0) {
    stop("No se pudo descargar precios desde Yahoo Finance con tq_get().", call. = FALSE)
  }

  out <- px |>
    group_by(symbol) |>
    tq_transmute(
      select     = adjusted,
      mutate_fun = periodReturn,
      period     = "monthly",
      type       = "arithmetic",
      col_rename = "ret"
    ) |>
    ungroup() |>
    mutate(
      date = as.Date(floor_date(fix_date(date), "month")),
      ret  = 100 * ret
    ) |>
    rename(ticker = symbol) |>
    distinct(ticker, date, .keep_all = TRUE) |>
    arrange(ticker, date)

  out
}

# ----------------------------
# RF mensual (en %) con fallbacks, sin quantmod
# 1) FRED CSV (TB3MS) robusto a nombres
# 2) Yahoo ^IRX como proxy si FRED falla
# ----------------------------
get_rf_monthly <- function(start, end) {

  fred_url <- "https://fred.stlouisfed.org/graph/fredgraph.csv?id=TB3MS"

  rf1 <- tryCatch({
    raw <- readr::read_csv(fred_url, show_col_types = FALSE)
    if (!is.data.frame(raw) || nrow(raw) == 0) stop("FRED vacío")

    nms <- names(raw)

    date_col <- dplyr::case_when(
      "DATE" %in% nms ~ "DATE",
      "observation_date" %in% nms ~ "observation_date",
      "date" %in% nms ~ "date",
      TRUE ~ nms[1]
```

```r
    )

    val_candidates <- c("TB3MS", "value", "VALUE")
    val_col <- val_candidates[val_candidates %in% nms][1]
    if (is.na(val_col)) val_col <- setdiff(nms, date_col)[1]

    raw |>
      transmute(
        date_raw = .data[[date_col]],
        tb3ms    = suppressWarnings(as.numeric(.data[[val_col]]))
      ) |>
      mutate(date = fix_date(date_raw)) |>
      filter(!is.na(date), !is.na(tb3ms), date >= start, date <= end) |>
      mutate(date = as.Date(floor_date(date, "month"))) |>
      group_by(date) |>
      summarise(tb3ms = last(tb3ms), .groups = "drop") |>
      transmute(
        date = date,
        rf   = 100 * ((1 + (tb3ms/100))^(1/12) - 1)
      )
  }, error = function(e) NULL)

  if (!is.null(rf1) && nrow(rf1) > 0) return(rf1)

  rf2 <- tryCatch({
    tq_get("^IRX", get = "stock.prices", from = start, to = end) |>
      transmute(
        date = as.Date(floor_date(fix_date(date), "month")),
        irx  = adjusted
      ) |>
      group_by(date) |>
      summarise(irx = last(irx), .groups = "drop") |>
      filter(!is.na(irx)) |>
      transmute(
        date = date,
        rf   = 100 * ((1 + (irx/100))^(1/12) - 1)
      )
  }, error = function(e) NULL)

  if (!is.null(rf2) && nrow(rf2) > 0) {
    warning("RF: no se pudo obtener TB3MS desde FRED; usando ^IRX como proxy.")
    return(rf2)
  }

  stop("RF: falló FRED (TB3MS) y falló Yahoo (^IRX).", call. = FALSE)
}

# ----------------------------
# Descargas
# ----------------------------
mkt_mret <- get_monthly_returns_yahoo(ticker_mkt, start_date, end_date) |>
  transmute(date = fix_date(date), mkt_ret = ret) |>
  distinct(date, .keep_all = TRUE) |>
  arrange(date)

assets_mret <- get_monthly_returns_yahoo(tickers_asset, start_date, end_date) |>
  mutate(date = fix_date(date))

rf_m <- get_rf_monthly(start_date, end_date) |>
```

```r
  mutate(date = fix_date(date))

# ----------------------------
# Normalización final: asegurar Date y calendario común real
# ----------------------------
mkt_mret2 <- mkt_mret |>
  mutate(date = as.Date(floor_date(fix_date(date), "month"))) |>
  distinct(date, .keep_all = TRUE) |>
  arrange(date)

rf_m2 <- rf_m |>
  mutate(date = as.Date(floor_date(fix_date(date), "month"))) |>
  group_by(date) |>
  summarise(rf = last(rf), .groups = "drop") |>
  filter(!is.na(rf)) |>
  arrange(date)

assets_mret2 <- assets_mret |>
  mutate(date = as.Date(floor_date(fix_date(date), "month"))) |>
  distinct(ticker, date, .keep_all = TRUE) |>
  arrange(ticker, date)

# Mercado + RF (meses donde existen ambos)
mkt_rf <- inner_join(mkt_mret2, rf_m2, by = "date") |>
  arrange(date)

# Calendario común (forzamos Date después de intersect por seguridad)
common_dates <- intersect(unique(assets_mret2$date), unique(mkt_rf$date))
common_dates <- fix_date(common_dates)
common_dates <- sort(common_dates)

# Diagnóstico: aquí common_dates debe imprimirse como fechas, no números
range(mkt_rf$date); nrow(mkt_rf)
```

```
## [1] "2015-10-01" "2025-12-01"
```

```
## [1] 123
```

```r
range(common_dates); length(common_dates)
```

```
## [1] "2015-10-01" "2025-12-01"
```

```
## [1] 123
```

```r
tail(common_dates, 5)
```

```
## [1] "2025-08-01" "2025-09-01" "2025-10-01" "2025-11-01" "2025-12-01"
```

```r
# Últimos 120 meses
last_120 <- tail(common_dates, 120)

# Dataset maestro
capm_practical_data <- assets_mret2 |>
  filter(date %in% last_120) |>
  inner_join(mkt_rf |> filter(date %in% last_120), by = "date") |>
  mutate(
    excess_ret = ret     - rf,
    mkt_excess = mkt_ret - rf
  ) |>
  select(date, ticker, ret, rf, mkt_ret, excess_ret, mkt_excess) |>
  arrange(ticker, date)

# Checks finales: debe dar 120 meses por ticker
capm_practical_data |>
  group_by(ticker) |>
  summarise(n_months = n_distinct(date),
            start    = min(date),
            end      = max(date),
            .groups = "drop")
```

```
## # A tibble: 3 x 4
##   ticker n_months start      end
##   <chr>     <int> <date>     <date>
## 1 AAPL        120 2016-01-01 2025-12-01
## 2 ARKK        120 2016-01-01 2025-12-01
## 3 KO          120 2016-01-01 2025-12-01
```

```r
tail(capm_practical_data)
```

```
## # A tibble: 6 x 7
##   date       ticker   ret    rf mkt_ret excess_ret mkt_excess
##   <date>     <chr>  <dbl> <dbl>   <dbl>      <dbl>      <dbl>
## 1 2025-07-01 KO     -4.04 0.347   2.17       -4.39      1.82
## 2 2025-08-01 KO      1.62 0.337   1.91        1.28      1.57
## 3 2025-09-01 KO     -3.13 0.321   3.53       -3.45      3.21
## 4 2025-10-01 KO      3.89 0.313   2.27        3.58      1.96
## 5 2025-11-01 KO      6.12 0.310   0.130       5.82     -0.180
## 6 2025-12-01 KO     -3.50 0.294   0.688      -3.79      0.394
```

4 Aplicación 1. Estimar beta de AAPL (exposición al mercado)

- Objetivo: cuantificar la exposición de `AAPL` al riesgo de mercado usando el índice S&P 500 como benchmark.
- Trabajamos con rendimientos en exceso: $r_{i,t} = R_{i,t} - R_{f,t}$ y $r_{M,t} = R_{M,t} - R_{f,t}$, donde $R_{M,t}$ proviene de `^GSPC`.
- Estimamos la regresión de serie de tiempo en exceso

$$r_{i,t} = \alpha_i + \beta_i r_{M,t} + \varepsilon_{i,t},$$

  y reportamos $\hat{\beta}$ como medida de sensibilidad del activo a variaciones del mercado netas de la tasa libre de riesgo.
- Interpretación operativa: $\hat{\beta} > 1$ implica sensibilidad superior a la del mercado; $\hat{\beta} < 1$ implica sensibilidad inferior. El parámetro es condicional a ventana, frecuencia y benchmark.

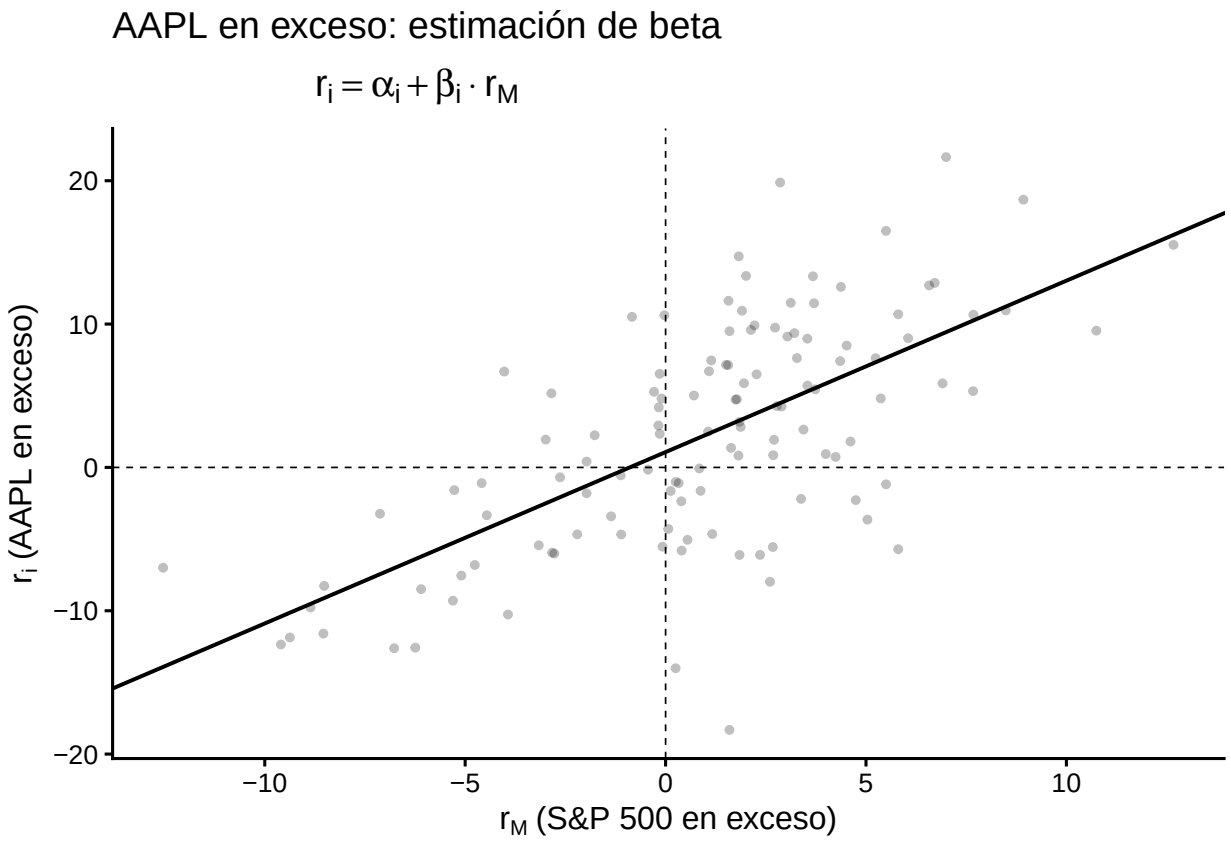
```r
# Estimación de $\beta$ para AAPL (OLS y opcional HAC).
library(broom)
library(lmtest)
library(sandwich)

# Filtrar AAPL
aapl_df <- capm_practical_data |>
  filter(ticker == "AAPL") |>
  select(date, excess_ret, mkt_excess) |>
  drop_na()

# Regresión CAPM en exceso
fit_aapl <- lm(excess_ret ~ mkt_excess, data = aapl_df)

# Resultados OLS
aapl_ols <- tidy(fit_aapl) |>
  mutate(term = recode(term, `(Intercept)` = "alpha", mkt_excess = "beta")) |>
  select(term, estimate, std.error, statistic, p.value)

aapl_ols
```

```
## # A tibble: 2 x 5
##   term  estimate std.error statistic  p.value
##   <chr>    <dbl>     <dbl>     <dbl>    <dbl>
## 1 alpha     1.07     0.564      1.89 6.05e- 2
## 2 beta      1.20     0.127      9.40 5.24e-16
```

```r
# (Opcional) Errores estándar HAC Newey--West para inferencia robusta
# Nota: el objetivo del video es operativo; HAC se reporta como referencia.
nw_L <- 3
aapl_hac <- coeftest(fit_aapl, vcov. = NeweyWest(fit_aapl, lag = nw_L, prewhite = FALSE)) |>
  tidy() |>
  mutate(term = recode(term, `(Intercept)` = "alpha", mkt_excess = "beta"),
         nw_lag = nw_L) |>
  select(term, estimate, std.error, statistic, p.value, nw_lag)

aapl_hac
```

```
## # A tibble: 2 x 6
##   term  estimate std.error statistic  p.value nw_lag
##   <chr>    <dbl>     <dbl>     <dbl>    <dbl>  <dbl>
## 1 alpha     1.07     0.523      2.04 4.33e- 2      3
## 2 beta      1.20     0.109     10.9  1.13e-19      3
```

```r
# Visualización (AAPL): dispersión $r_{i,t}$ vs $r_{M,t}$ y recta estimada.
library(ggplot2)

coef_aapl <- coef(fit_aapl)
alpha_hat <- unname(coef_aapl[1])
beta_hat  <- unname(coef_aapl[2])

ggplot(aapl_df, aes(x = mkt_excess, y = excess_ret)) +
  geom_hline(yintercept = 0, linetype = "dashed", linewidth = 0.3) +
  geom_vline(xintercept = 0, linetype = "dashed", linewidth = 0.3) +
  geom_point(alpha = 0.25, size = 1) +
  geom_abline(intercept = alpha_hat, slope = beta_hat, linewidth = 0.7) +
  labs(
    title = expression(atop("AAPL en exceso: estimación de beta",
                            r[i,t] == alpha[i] + beta[i] %.% r[M,t])),
    x = expression(r[M,t]~"(S&P 500 en exceso)"),
    y = expression(r[i,t]~"(AAPL en exceso)")
  ) +
  theme_classic(base_size = 12)
```



5 Resultados. Estimación e interpretación operativa (AAPL)

- Estimación (mensual, 120 observaciones hasta 2025-12, benchmark S&P 500):

  $\hat{\alpha} = 1.0693 \quad (\text{EE HAC} = 0.5234,\ t = 2.043,\ p = 0.043),$

  $\hat{\beta} = 1.1957 \quad (\text{EE HAC} = 0.1092,\ t = 10.946,\ p < 10^{-18}).$

- Interpretación de $\hat{\beta}$: la sensibilidad estimada es mayor que uno. En la muestra, AAPL muestra una reacción promedio superior a la del mercado ante variaciones mensuales del S&P 500 en exceso de la tasa libre de riesgo. En términos marginales, un cambio de 1% en $r_{M,t}$ se asocia con un cambio de aproximadamente 1.20% en $r_{i,t}$.

- Lectura en escenarios (en retornos en exceso). Para un escenario de mercado de $r_{M,t} = +5\%$, el componente explicado por exposición al mercado es aproximadamente

  $$\hat{\beta}\, r_{M,t} \approx 1.1957 \times 5\% \approx 5.98\%.$$

  Para $r_{M,t} = -5\%$, el componente explicado es aproximadamente -5.98%. La contribución de $\hat{\alpha}$, al ser un intercepto mensual en exceso, desplaza el nivel medio del retorno en exceso, pero su interpretación es más natural en la evaluación de desempeño (Aplicación 3) que como "predicción" puntual.

- Comentario sobre $\hat{\alpha}$: el intercepto estimado es positivo y estadísticamente distinto de cero bajo el ajuste HAC. En este video, $\hat{\alpha}$ se interpreta como un promedio residual ex post condicionado a la especificación y al benchmark; no se utiliza como garantía de rendimiento futuro ni como objetivo de optimización.

- Nota metodológica mínima: se reportan errores estándar HAC (Newey–West, rezago 3) para evitar inferencia optimista si existe autocorrelación y heterocedasticidad en rendimientos mensuales. El énfasis del video, sin embargo, es operativo: la magnitud de $\hat{\beta}$ como insumo para decisiones y comparaciones.

5.0.1 Bloque 0

En este video el CAPM se presenta como una herramienta operativa para decisiones financieras en las que se requiere cuantificar exposición al riesgo de mercado, derivar un rendimiento requerido para el capital propio y evaluar el desempeño de un fondo con un marco consistente. Trabajamos con rendimientos en exceso definidos como $r_{i,t} = R_{i,t} - R_{f,t}$ para el activo o fondo y $r_{M,t} = R_{M,t} - R_{f,t}$ para el mercado, de modo que la beta se interprete como sensibilidad a variaciones del mercado netas de la tasa libre de riesgo. Con esa beta, el CAPM se traduce en un rendimiento requerido para el equity, útil en valuación por descuento de flujos, análisis de proyectos y comparaciones de costo de capital entre empresas con distintos perfiles de riesgo. El mismo esquema de estimación permite además evaluar el desempeño de un fondo mediante el alpha de Jensen, separando el componente atribuible a exposición al mercado del componente residual, lo que ayuda a juzgar si el rendimiento observado es consistente con el riesgo tomado. El énfasis, por tanto, no está en probar el modelo como teoría de equilibrio, sino en usar sus estimaciones como insumos transparentes para decisiones de valuación y evaluación de resultados bajo supuestos explícitos.