Analisi Risk Score PNRR

MR

2026-06-04

- 1 Pacchetti
- 2 Preparazione dati
- 3 Variabili incluse nel modello
- 4 Missing data
  - 4.1 Modelli esplorativi della missingness
- 5 Multiple imputation
  - 5.1 Diagnostica post-imputazione
- 6 Modello completo
- 7 Selezione backward
- 8 Modello finale e hazard ratio
  - 8.1 Analisi secondaria: hazard ratio per demenza di Alzheimer
  - 8.2 Confronto AIC: modello completo vs modello finale
- 9 Costruzione del risk score
  - 9.1 Calcolo dello score nelle imputazioni
- 10 Modelli Cox basati sul risk score
- 11 Score corretto per covariate aggiuntive
- 12 Population attributable fraction
- 13 Analisi parallela: modello completo con fattori dicotomici e PAF a 144 mesi
  - 13.1 Import e pre-processing dedicati
  - 13.2 Variabili e imputazione multipla
  - 13.3 Modello Cox completo
  - 13.4 Punteggio derivato dal modello completo
  - 13.5 Population attributable fraction a 144 mesi
  - 13.6 Descrittive delle variabili incluse nel PAF
- 14 Analisi stratificata per MMSE
  - 14.1 Hazard ratio per strato MMSE
  - 14.2 Forest plot per strato MMSE
  - 14.3 Interazione tra risk score e MMSE
  - 14.4 Numerosità per categoria MMSE
- 15 Performance del risk score
  - 15.1 Coefficiente, HR e intervalli di confidenza
  - 15.2 C-index
  - 15.3 R quadrato di Royston-Sauerbrei/Nagelkerke
  - 15.4 Assunzione di proporzionalità dei rischi
  - 15.5 Distribuzione dello score
- 16 Analisi univariata
- 17 Time-dependent ROC e cut-off
  - 17.1 Andamento della AUC nel tempo
  - 17.2 Curva ROC a un tempo selezionato

- 18 Validazione esterna: coorte TRELONG
  - 18.1 Modello Cox nella coorte di validazione
  - 18.2 Time-dependent ROC nella coorte di validazione
  - 18.3 Curva ROC nella coorte di validazione a un tempo selezionato
  - 18.4 Confronto time-dependent AUC: derivazione vs validazione
- 19 Curve di sopravvivenza predette
  - 19.1 Curve per gruppo di rischio
- 20 Note operative
- 21 Appendice esplorativa: sequenze, modelli multistato e dropout
  - 21.1 Razionale dell'appendice
  - 21.2 Pacchetti aggiuntivi per le analisi esplorative
  - 21.3 Codice esplorativo originale riorganizzabile

1 Pacchetti

```
library(haven)
library(mice)
library(psfmi)      # >= 1.2.0
library(survival)
library(dplyr)
library(tibble)
library(naniar)
library(broom)
library(AF)
library(purrr)
library(ggplot2)
library(tidyr)
library(readxl)
library(Hmisc)
library(survivalROC)
```

2 Preparazione dati

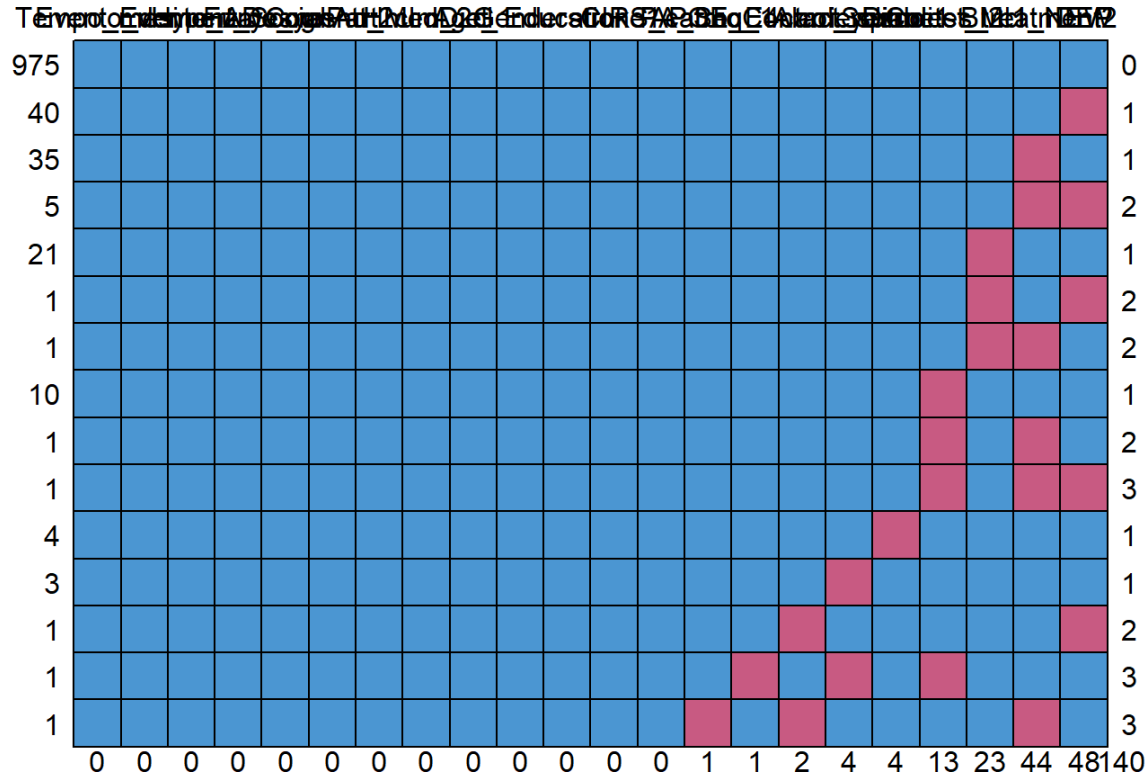
```
PNRR <- read_sav(
  "W:/LAB NEUROPSICOLOGIA/Prev-Ita-Dem_PNRR/Risk score/Analisi_MR/invece_risk_fa
ctors_pnrr_type.sav"
) |>
  zap_labels() |>
  mutate(across(where(is.labelled), as_factor)) |>
  mutate(event = as.integer(Evento_demenza == 1))

fattori <- c(
  "Alcol_yesno", "CognAtt_num_2cl", "DEP", "Diabetes_2cl", "HMedDiet",
  "Hypert_yesno", "IpColest_treatment2", "SocialPart2cl", "Solitudi1",
  "CIRS_cardio", "Gender", "APOE_E4carrier", "stroke1"
)

PNRR <- PNRR |>
  mutate(across(all_of(fattori), as_factor))
```

3 Variabili incluse nel modello

```
vars_model <- c(
  "Tempo_mesi", "Evento_demenza", "Evento_AD",
  "BMI1_NEW", "Hypert_yesno", "IpColest_treatment2",
  "Alcol_yesno", "Fumo_yesno", "PA_freq_1", "SocContact_sum",
  "SocialPart2cl", "Solitudi1", "CognAtt_num_2cl",
  "DEP", "Diabetes_2cl", "HMedDiet",
  "Age_1", "Gender", "Education",
  "APOE_E4carrier", "stroke1", "CIRS_cardio"
)
```

4 Missing data

```
md.pattern(PNRR[vars_model])
```


975 0
40 1
35 1
5 2
21 1
1 2
1 2
10 1
1 2
1 3
4 1
3 1
1 2
1 3
1 3
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 2 4 4 13 23 44 48 140

```
##     Tempo_mesi Evento_demenza Evento_AD Hypert_yesno Fumo_yesno SocialPart2cl
## 975          1              1         1            1          1             1
## 40           1              1         1            1          1             1
## 35           1              1         1            1          1             1
## 5            1              1         1            1          1             1
## 21           1              1         1            1          1             1
## 1            1              1         1            1          1             1
## 1            1              1         1            1          1             1
## 10           1              1         1            1          1             1
## 1            1              1         1            1          1             1
## 1            1              1         1            1          1             1
## 4            1              1         1            1          1             1
## 3            1              1         1            1          1             1
## 1            1              1         1            1          1             1
## 1            1              1         1            1          1             1
## 1            1              1         1            1          1             1
##              0              0         0            0          0             0
##     CognAtt_num_2cl HMedDiet Age_1 Gender Education stroke1 CIRS_cardio
## 975               1        1     1      1         1       1           1
## 40                1        1     1      1         1       1           1
## 35                1        1     1      1         1       1           1
## 5                 1        1     1      1         1       1           1
## 21                1        1     1      1         1       1           1
## 1                 1        1     1      1         1       1           1
## 1                 1        1     1      1         1       1           1
## 10                1        1     1      1         1       1           1
## 1                 1        1     1      1         1       1           1
## 1                 1        1     1      1         1       1           1
## 4                 1        1     1      1         1       1           1
## 3                 1        1     1      1         1       1           1
## 1                 1        1     1      1         1       1           1
## 1                 1        1     1      1         1       1           1
## 1                 1        1     1      1         1       1           1
##                   0        0     0      0         0       0           0
##     PA_freq_1 APOE_E4carrier SocContact_sum Alcol_yesno Solitudi1 Diabetes_2c
l
## 975         1              1              1          1         1
1
## 40          1              1              1          1         1
1
## 35          1              1              1          1         1
1
## 5           1              1              1          1         1
1
## 21          1              1              1          1         1
1
## 1           1              1              1          1         1
1
```

```
## 1           1              1              1           1         1
1
## 10          1              1              1           1         1
0
## 1           1              1              1           1         1
0
## 1           1              1              1           1         1
0
## 4           1              1              1           1         0
1
## 3           1              1              1           0         1
1
## 1           1              1              0           1         1
1
## 1           1              0              1           0         1
0
## 1           0              1              0           1         1
1
##            1              1              2           4         4           1
3
##     IpColest_treatment2 BMI1_NEW DEP
## 975                   1        1   1   0
## 40                    1        1   0   1
## 35                    1        0   1   1
## 5                     1        0   0   2
## 21                    0        1   1   1
## 1                     0        1   0   2
## 1                     0        0   1   2
## 10                    1        1   1   1
## 1                     1        0   1   2
## 1                     1        0   0   3
## 4                     1        1   1   1
## 3                     1        1   1   1
## 1                     1        1   0   2
## 1                     1        1   1   3
## 1                     1        0   1   3
##                      23       44  48 140
```

```
miss_summ <- naniar::miss_var_summary(PNRR[vars_model])
miss_summ
```

```
## # A tibble: 22 × 3
##    variable            n_miss pct_miss
##    <chr>                <int>    <num>
##  1 DEP                     48   4.36
##  2 BMI1_NEW                44   4
##  3 IpColest_treatment2     23   2.09
##  4 Diabetes_2cl            13   1.18
##  5 Alcol_yesno              4   0.364
##  6 Solitudi1                4   0.364
##  7 SocContact_sum           2   0.182
##  8 PA_freq_1                1   0.0909
##  9 APOE_E4carrier           1   0.0909
## 10 Tempo_mesi               0   0
## # ℹ 12 more rows
```

```
naniar::mcar_test(PNRR[vars_model])
```

```
## # A tibble: 1 × 4
##   statistic    df   p.value missing.patterns
##       <dbl> <dbl>     <dbl>            <int>
## 1      385.   283 0.0000514               15
```

4.1 Modelli esplorativi della missingness

```
PNRR2 <- PNRR |>
  mutate(
    log_time = log(Tempo_mesi),
    event = as.integer(Evento_demenza == 1)
  )

vars_with_na <- names(PNRR2[vars_model])[sapply(PNRR2[vars_model], \(x) any(is.n
a(x)))]

check_models <- lapply(vars_with_na, function(v) {
  PNRR2 |>
    mutate(miss = as.integer(is.na(.data[[v]]))) |>
    glm(
      miss ~ event + log_time + Age_1 + Gender + Education +
        Hypert_yesno + IpColest_treatment2 + Alcol_yesno + Fumo_yesno +
        PA_freq_1 + SocContact_sum + SocialPart2cl + Solitudi1 +
        CognAtt_num_2cl + DEP + Diabetes_2cl + HMedDiet +
        APOE_E4carrier + stroke1 + CIRS_cardio,
      data = _,
      family = binomial()
    ) |>
    tidy() |>
    mutate(target = v)
})

check_res <- bind_rows(check_models)

check_res |>
  filter(term != "(Intercept)", p.value < 0.05) |>
  arrange(target, p.value)
```

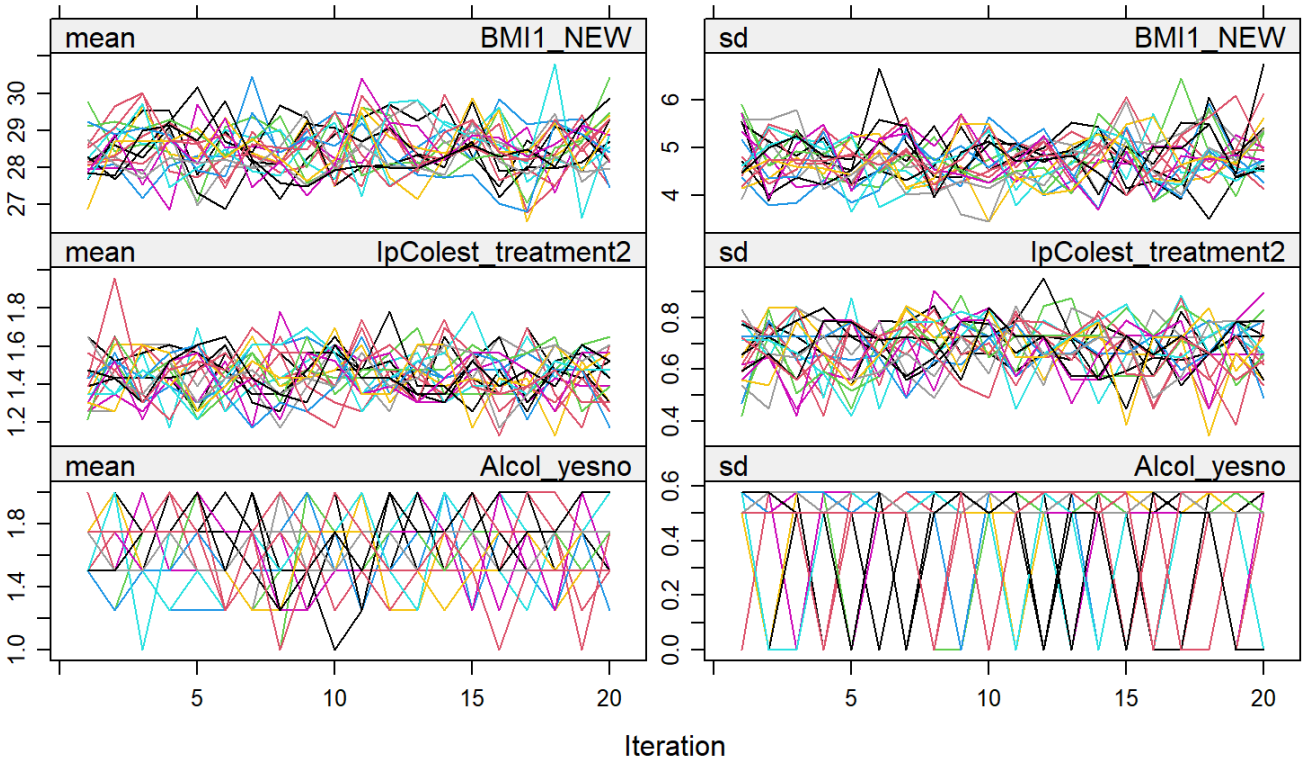
```
## # A tibble: 4 × 6
##   term           estimate std.error statistic    p.value target
##   <chr>             <dbl>     <dbl>     <dbl>      <dbl> <chr>
## 1 SocContact_sum   -0.283    0.0624     -4.54 0.00000564 BMI1_NEW
## 2 PA_freq_1        -0.249    0.113      -2.21 0.0273     BMI1_NEW
## 3 stroke11          1.07     0.524       2.05 0.0404     BMI1_NEW
## 4 CIRS_cardio4      1.76     0.864       2.03 0.0420     BMI1_NEW
```

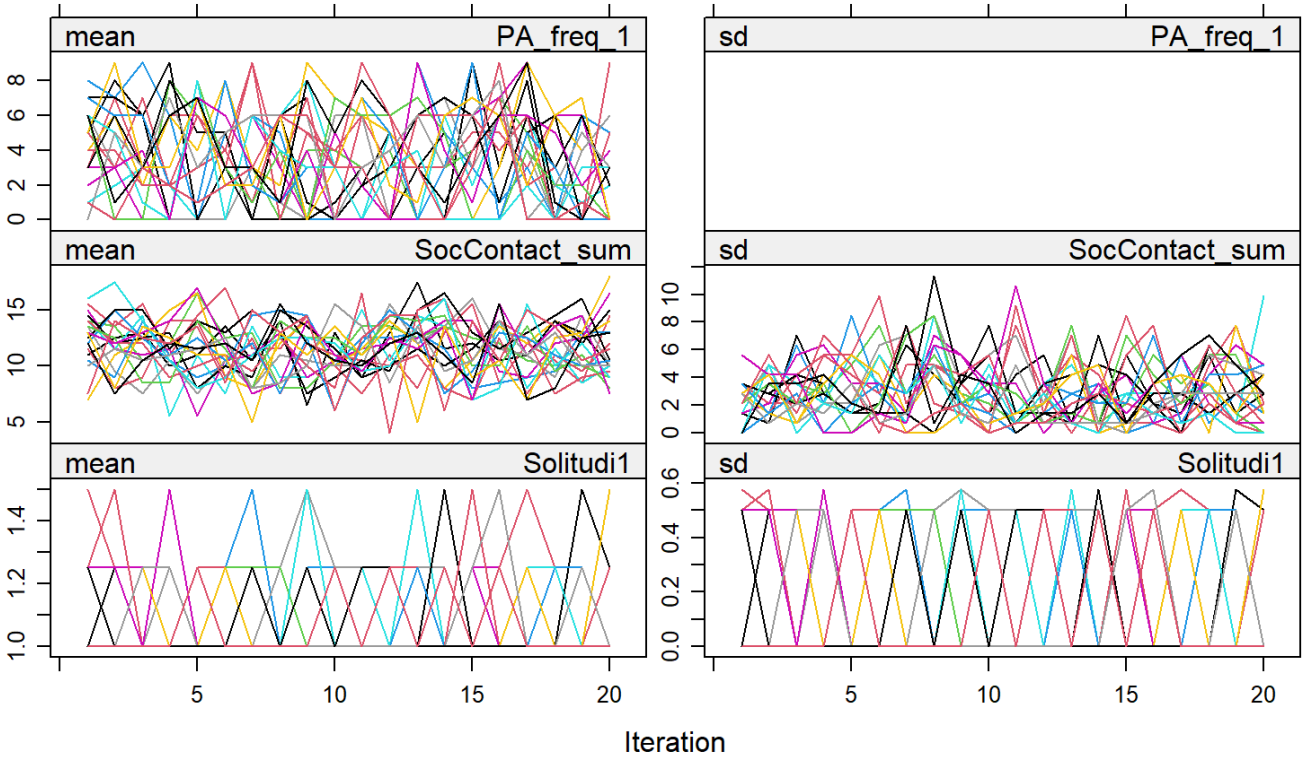
5 Multiple imputation

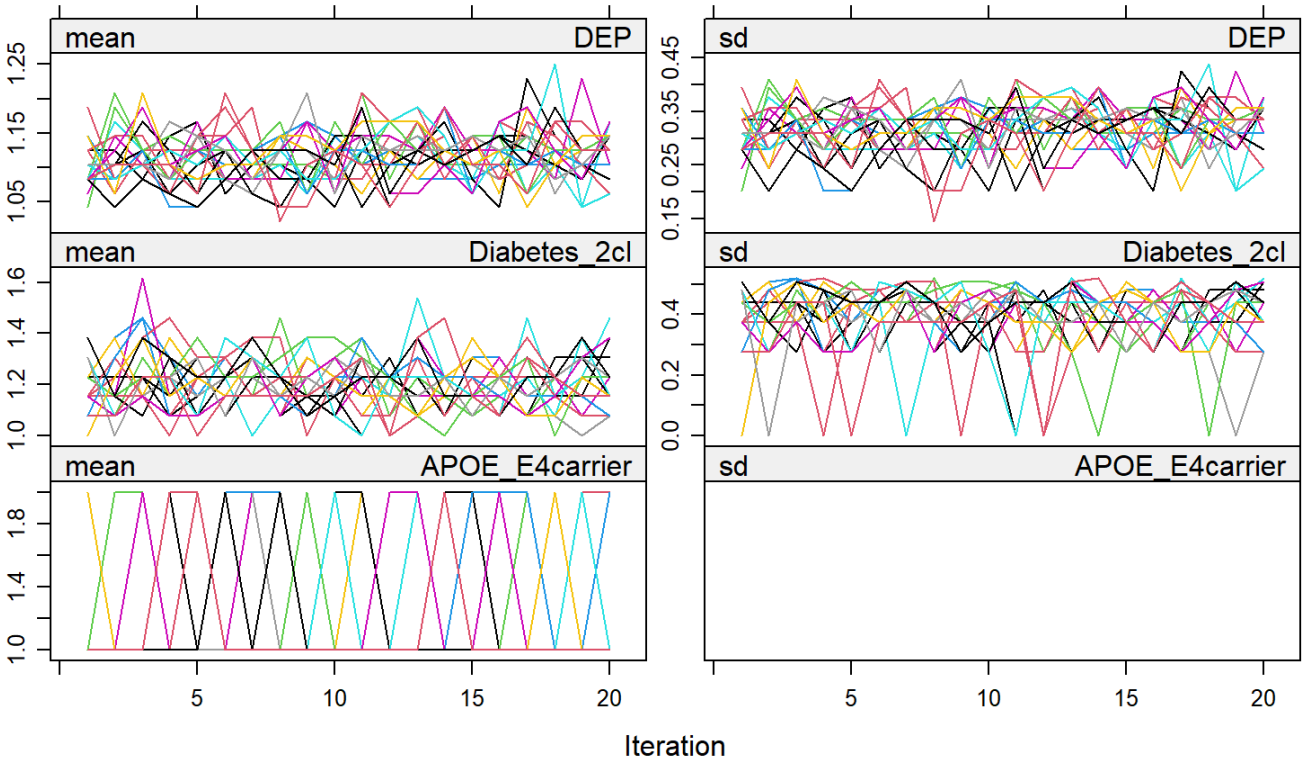
```
set.seed(2025)

ini <- mice(PNRR[vars_model], maxit = 0)
ini$method[c("Tempo_mesi", "Evento_demenza", "Evento_AD")] <- ""

imp <- mice(
  PNRR[vars_model],
  m = 20,
  maxit = 20,
  method = ini$method,
  predictorMatrix = ini$predictorMatrix,
  printFlag = FALSE
)

imp_long_full <- complete(imp, action = "long", include = TRUE)
imp_long <- imp_long_full |>
  filter(.imp != 0)
```

5.1 Diagnostica post-imputazione

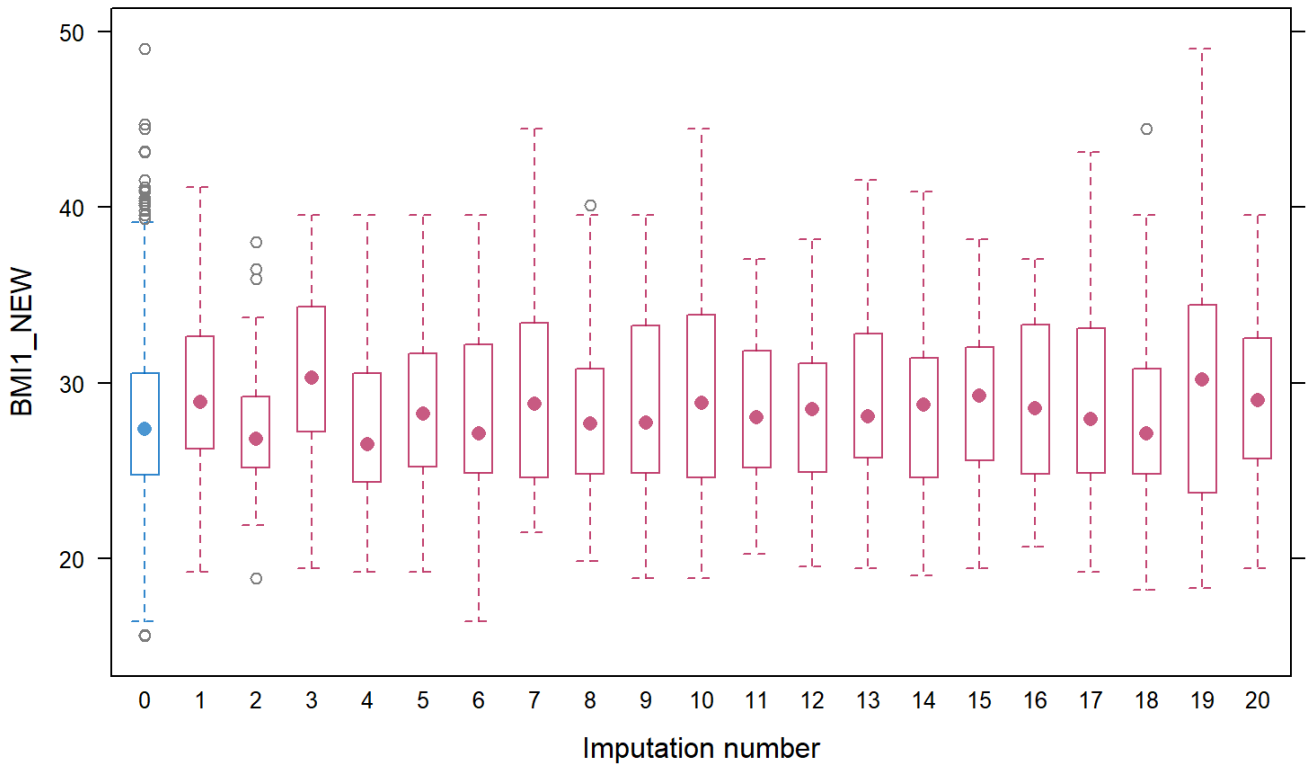
```
plot(imp)
```


mean
BMI1_NEW
sd
BMI1_NEW
mean
lpColest_treatment2
sd
lpColest_treatment2
mean
Alcol_yesno
sd
Alcol_yesno
Iteration

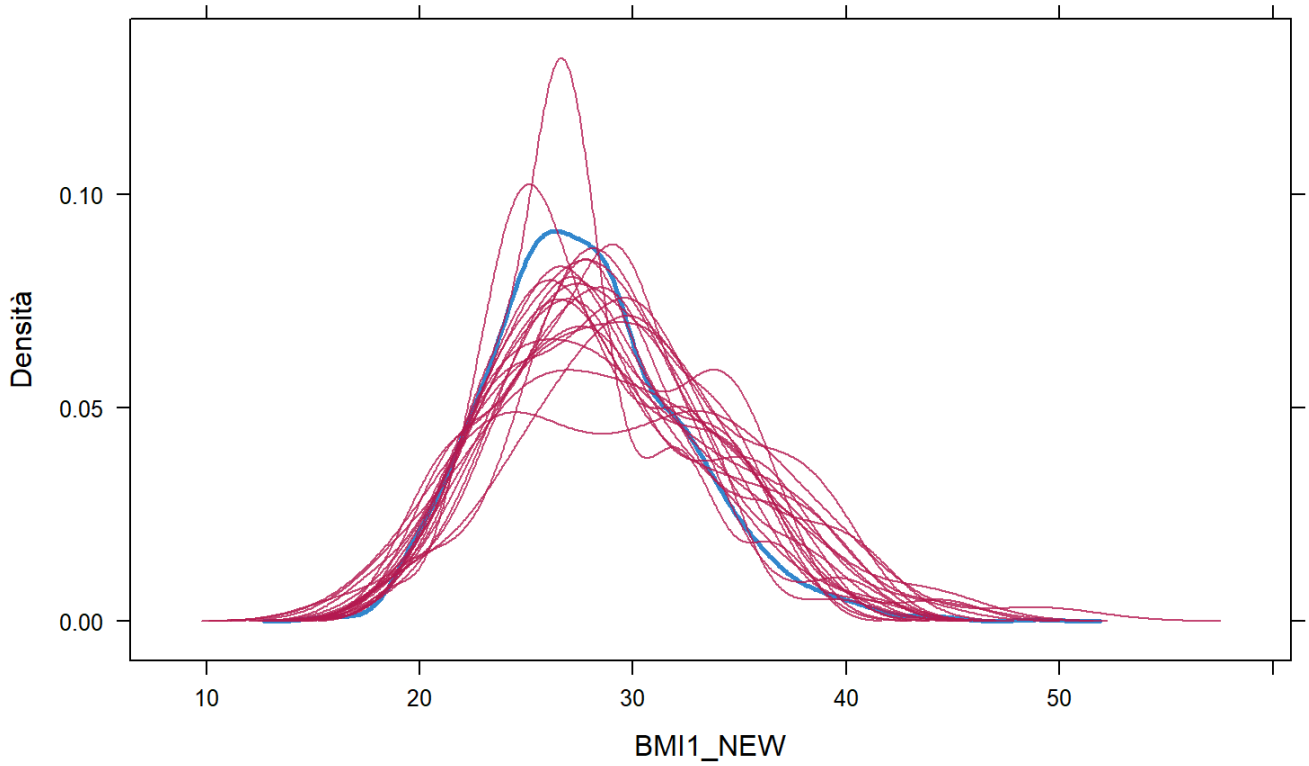

mean
PA_freq_1
sd
PA_freq_1
mean
SocContact_sum
sd
SocContact_sum
mean
Solitudi1
sd
Solitudi1
Iteration



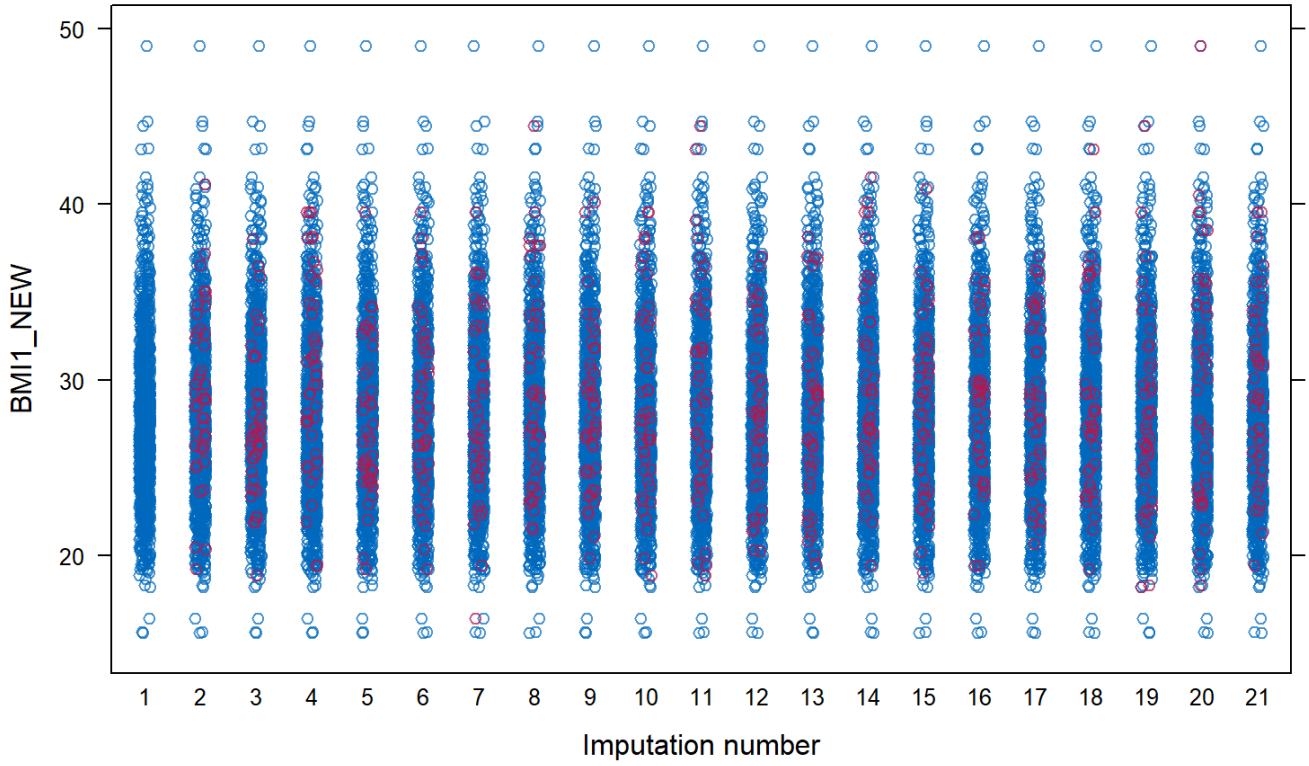

```
bwplot(imp, BMI1_NEW ~ .imp)
```




```
densityplot(imp, ~ BMI1_NEW)
```




```
stripplot(imp, BMI1_NEW ~ .imp)
```



6 Modello completo

```
form.full <- Surv(Tempo_mesi, Evento_demenza) ~
  BMI1_NEW + Hypert_yesno + IpColest_treatment2 +
  Alcol_yesno + PA_freq_1 + SocContact_sum + SocialPart2cl +
  Solitudi1 + CognAtt_num_2cl + DEP + Diabetes_2cl + HMedDiet
```

7 Selezione backward

```
multi_fattori <- c("IpColest_treatment2")

sel <- psfmi_coxr(
  data = imp_long,
  formula = form.full,
  nimp = imp$m,
  method = "D1",
  direction = "BW",
  p.crit = 0.157,
  impvar = ".imp",
  cat.predictors = multi_fattori
)

sel$predictors_final
```

```
## [1] "IpColest_treatment2" "Alcol_yesno"         "SocialPart2cl"
## [4] "CognAtt_num_2cl"     "Diabetes_2cl"
```

```
form.sel <- as.formula(
  paste(
    "Surv(Tempo_mesi, Evento_demenza) ~",
    paste(sel$predictors_final, collapse = " + ")
  )
)
```

8 Modello finale e hazard ratio

```r
fits <- lapply(1:imp$m, function(i) {
  d <- complete(imp, i)
  coxph(form.sel, data = d, method = "breslow")
})

pooled <- mice::pool(fits)
coef_all <- summary(pooled)

HR_tbl <- coef_all |>
  mutate(
    HR = exp(estimate),
    Lower95 = exp(estimate - 1.96 * std.error),
    Upper95 = exp(estimate + 1.96 * std.error)
  ) |>
  select(term, HR, Lower95, Upper95, p.value)

HR_tbl
```

```
##                   term        HR   Lower95   Upper95     p.value
## 1 IpColest_treatment21 1.2357404 0.8676119 1.7600663 0.242507839
## 2 IpColest_treatment22 1.5147978 0.9808217 2.3394797 0.062905283
## 3         Alcol_yesno1 1.5790379 1.1262879 2.2137863 0.008831324
## 4       SocialPart2cl1 0.5303827 0.3441391 0.8174189 0.004590324
## 5     CognAtt_num_2cl1 0.6767985 0.4354817 1.0518381 0.084536758
## 6        Diabetes_2cl1 1.4521498 1.0199310 2.0675311 0.040066075
```

8.1 Analisi secondaria: hazard ratio per demenza di Alzheimer

Questa analisi ripete la stima del modello finale utilizzando come outcome `Evento_AD`. I predittori sono gli stessi selezionati nella procedura backward sul modello principale per demenza all-cause; in questo modo l'analisi AD è trattata come endpoint secondario e non come nuova procedura di selezione.

```
form.sel_AD <- as.formula(
  paste(
    "Surv(Tempo_mesi, Evento_AD) ~",
    paste(sel$predictors_final, collapse = " + ")
  )
)

fits_AD <- lapply(1:imp$m, function(i) {
  d <- complete(imp, i)
  coxph(form.sel_AD, data = d, method = "breslow")
})

pooled_AD <- mice::pool(fits_AD)
coef_all_AD <- summary(pooled_AD)

HR_tbl_AD <- coef_all_AD |>
  mutate(
    HR = exp(estimate),
    Lower95 = exp(estimate - 1.96 * std.error),
    Upper95 = exp(estimate + 1.96 * std.error)
  ) |>
  select(term, HR, Lower95, Upper95, p.value)

HR_tbl_AD
```

```
##                   term        HR   Lower95  Upper95   p.value
## 1 IpColest_treatment21 0.7991063 0.4343737 1.470096 0.4736188
## 2 IpColest_treatment22 1.0812549 0.5283370 2.212815 0.8314002
## 3         Alcol_yesno1 0.9441588 0.5779644 1.542372 0.8192525
## 4       SocialPart2cl1 0.9285371 0.5164495 1.669439 0.8051638
## 5     CognAtt_num_2cl1 0.6311868 0.3178607 1.253369 0.1934317
## 6        Diabetes_2cl1 1.2253873 0.6751822 2.223954 0.5064152
```

8.2 Confronto AIC: modello completo vs

modello finale

```
one_imp_compare <- function(i, form.full, form.sel, imp) {
  d <- complete(imp, i)
  fit_full  <- coxph(form.full, data = d, method = "breslow", x = TRUE)
  fit_final <- coxph(form.sel,  data = d, method = "breslow", x = TRUE)

  aic_full  <- AIC(fit_full)
  aic_final <- AIC(fit_final)
  delta_aic <- aic_full - aic_final

  ll_full  <- as.numeric(logLik(fit_full))
  ll_final <- as.numeric(logLik(fit_final))
  df_full  <- attr(logLik(fit_full), "df")
  df_final <- attr(logLik(fit_final), "df")
  df_diff  <- df_full - df_final
  chi2     <- 2 * (ll_full - ll_final)
  p_lrt    <- if (is.finite(chi2) && df_diff > 0) {
    pchisq(chi2, df = df_diff, lower.tail = FALSE)
  } else {
    NA_real_
  }

  tibble(
    imp = i,
    AIC_full = aic_full,
    AIC_final = aic_final,
    delta_AIC = delta_aic,
    LRT_chi2 = chi2,
    df = df_diff,
    p_LRT = p_lrt
  )
}

cmp_tab <- lapply(1:imp$m, one_imp_compare, form.full = form.full, form.sel = fo
rm.sel, imp = imp) |>
  bind_rows()

summary_aic <- cmp_tab |>
  summarise(
    mean_AIC_full = mean(AIC_full, na.rm = TRUE),
    mean_AIC_final = mean(AIC_final, na.rm = TRUE),
    mean_deltaAIC = mean(delta_AIC, na.rm = TRUE),
    sd_deltaAIC = sd(delta_AIC, na.rm = TRUE),
    n_imp_used = sum(is.finite(delta_AIC))
  )

pvals <- cmp_tab$p_LRT[is.finite(cmp_tab$p_LRT) & cmp_tab$p_LRT > 0]
```

```
p_combined <- if (length(pvals) >= 1) {
  stat <- -2 * sum(log(pvals))
  pchisq(stat, df = 2 * length(pvals), lower.tail = FALSE)
} else {
  NA_real_
}

cmp_tab
```

```
## # A tibble: 20 × 7
##      imp AIC_full AIC_final delta_AIC LRT_chi2    df p_LRT
##    <int>    <dbl>     <dbl>     <dbl>    <dbl> <int> <dbl>
##  1     1    2190.     2179.      11.6     6.37     9 0.702
##  2     2    2192.     2179.      12.2     5.77     9 0.762
##  3     3    2190.     2178.      11.7     6.29     9 0.710
##  4     4    2190.     2179.      11.2     6.81     9 0.657
##  5     5    2192.     2179.      12.3     5.65     9 0.774
##  6     6    2192.     2179.      12.4     5.56     9 0.783
##  7     7    2190.     2178.      12.6     5.39     9 0.799
##  8     8    2192.     2180.      11.8     6.21     9 0.719
##  9     9    2189.     2178.      11.6     6.37     9 0.702
## 10    10    2191.     2179.      11.2     6.75     9 0.663
## 11    11    2191.     2179.      12.2     5.78     9 0.762
## 12    12    2191.     2179.      12.1     5.89     9 0.751
## 13    13    2191.     2178.      12.4     5.63     9 0.777
## 14    14    2190.     2179.      11.6     6.44     9 0.695
## 15    15    2191.     2179.      12.2     5.80     9 0.760
## 16    16    2193.     2180.      12.4     5.60     9 0.779
## 17    17    2191.     2179.      11.9     6.06     9 0.734
## 18    18    2191.     2179.      12.2     5.77     9 0.763
## 19    19    2192.     2180.      12.4     5.57     9 0.782
## 20    20    2190.     2178.      12.0     6.00     9 0.740
```

```
summary_aic
```

```
## # A tibble: 1 × 5
##   mean_AIC_full mean_AIC_final mean_deltaAIC sd_deltaAIC n_imp_used
##           <dbl>          <dbl>         <dbl>       <dbl>      <int>
## 1         2191.          2179.          12.0       0.410         20
```

```
p_combined
```

```
## [1] 0.9999944
```

9 Costruzione del risk score

```
betas <- setNames(coef_all$estimate, coef_all$term)
beta_min <- min(abs(betas))
points <- round(betas / beta_min)

score_tbl <- tibble(
  Variabile = names(points),
  BetaPool = round(betas, 3),
  Punti = points
)

score_tbl
```

```
## # A tibble: 6 × 3
##   Variabile            BetaPool Punti
##   <chr>                   <dbl> <dbl>
## 1 IpColest_treatment21    0.212     1
## 2 IpColest_treatment22    0.415     2
## 3 Alcol_yesno1            0.457     2
## 4 SocialPart2cl1         -0.634    -3
## 5 CognAtt_num_2cl1       -0.39     -2
## 6 Diabetes_2cl1           0.373     2
```

9.1 Calcolo dello score nelle imputazioni

```
rhs_terms <- delete.response(terms(form.sel))

add_score <- function(df) {
  X <- model.matrix(rhs_terms, data = df, na.action = na.pass)[, -1, drop = FALS
E]

  miss_cols <- setdiff(names(points), colnames(X))
  if (length(miss_cols)) {
    X <- cbind(
      X,
      matrix(0, nrow = nrow(X), ncol = length(miss_cols), dimnames = list(NULL,
miss_cols))
    )
  }

  X <- X[, names(points), drop = FALSE]
  score_vec <- rep(NA_real_, nrow(df))
  idx <- as.integer(rownames(X))
  score_vec[idx] <- as.numeric(X %*% points)

  df$risk_score <- score_vec
  df
}

imp_long <- complete(imp, action = "long", include = TRUE) |>
  group_split(.imp) |>
  lapply(add_score) |>
  bind_rows()

imp_long <- imp_long |>
  mutate(
    CIRS_cardio2cl = case_when(
      CIRS_cardio %in% c("1", "2") ~ "0",
      CIRS_cardio %in% c("3", "4") ~ "1",
      TRUE ~ NA_character_
    ),
    CIRS_cardio2cl = as_factor(CIRS_cardio2cl)
  )

imp_score <- as.mids(imp_long)
```

10 Modelli Cox basati sul risk score

```
fit_corr_1 <- with(
  imp_score,
  coxph(Surv(Tempo_mesi, Evento_demenza) ~ risk_score, method = "breslow")
)

pool_fit_corr_1 <- summary(pool(fit_corr_1)) |>
  mutate(
    HR = exp(estimate),
    CI_lower = exp(estimate - 1.96 * std.error),
    CI_upper = exp(estimate + 1.96 * std.error)
  )

pool_fit_corr_1
```

```
##         term  estimate  std.error statistic       df      p.value       HR
## 1 risk_score 0.2060457 0.03841664  5.363449 170.3688 2.629987e-07 1.228809
##   CI_lower CI_upper
## 1 1.139682 1.324907
```

```
fit_corr_2 <- with(
  imp_score,
  coxph(
    Surv(Tempo_mesi, Evento_demenza) ~
      risk_score + Age_1 + Gender + Education +
      APOE_E4carrier + stroke1 + CIRS_cardio2cl,
    method = "breslow"
  )
)

pool_fit_corr_2 <- summary(pool(fit_corr_2)) |>
  mutate(
    HR = exp(estimate),
    CI_lower = exp(estimate - 1.96 * std.error),
    CI_upper = exp(estimate + 1.96 * std.error)
  )

pool_fit_corr_2
```

```
##              term      estimate  std.error    statistic       df      p.value
## 1      risk_score  0.1751983894 0.04011341  4.367576336 163.3474 2.224157e-05
## 2           Age_1  0.1361307126 0.05927040  2.296773874 164.0046 2.289642e-02
## 3         Gender2  0.0002197678 0.16100504  0.001364975 163.9934 9.989126e-01
## 4       Education -0.0253093670 0.02616271 -0.967383314 163.9990 3.347769e-01
## 5 APOE_E4carrier1  0.5540630740 0.17496142  3.166772893 164.0179 1.838335e-03
## 6 APOE_E4carrier2  1.7516035364 0.71740663  2.441577023 164.0351 1.568575e-02
## 7        stroke11  0.7861248795 0.22238842  3.534918306 164.0250 5.302608e-04
## 8 CIRS_cardio2cl1  0.1189824218 0.17874750  0.665645249 163.9976 5.065730e-01
##          HR  CI_lower  CI_upper
## 1 1.1914826 1.1013934  1.288941
## 2 1.1458317 1.0201613  1.286983
## 3 1.0002198 0.7295334  1.371342
## 4 0.9750082 0.9262711  1.026310
## 5 1.7403097 1.2350836  2.452205
## 6 5.7638378 1.4126771 23.516929
## 7 2.1948745 1.4194133  3.393990
## 8 1.1263501 0.7934517  1.598919
```

11 Score corretto per covariate aggiuntive

```
corr_tbl <- summary(pool(fit_corr_2)) |>
  select(term, estimate) |>
  filter(term != "risk_score")

beta_corr <- setNames(corr_tbl$estimate, corr_tbl$term)
beta_min_corr <- 0.197
points_corr <- round(beta_corr / beta_min_corr)

score_tbl_corr <- tibble(
  Variabile = names(points_corr),
  BetaPool = round(beta_corr, 3),
  Punti = points_corr
)

score_tbl_corr
```

```
## # A tibble: 7 × 3
##   Variabile       BetaPool Punti
##   <chr>              <dbl> <dbl>
## 1 Age_1              0.136     1
## 2 Gender2            0         0
## 3 Education         -0.025     0
## 4 APOE_E4carrier1    0.554     3
## 5 APOE_E4carrier2    1.75      9
## 6 stroke11           0.786     4
## 7 CIRS_cardio2cl1    0.119     1
```

```
rhs_corr <- terms(
  ~ Age_1 + Gender + Education + APOE_E4carrier + stroke1 + CIRS_cardio2cl
)

add_corr <- function(df) {
  Z <- model.matrix(rhs_corr, data = df, na.action = na.pass)[, -1, drop = FALS
E]

  miss <- setdiff(names(beta_corr), colnames(Z))
  if (length(miss)) {
    Z <- cbind(
      Z,
      matrix(0, nrow = nrow(Z), ncol = length(miss), dimnames = list(NULL, mis
s))
    )
  }

  Z <- Z[, names(beta_corr), drop = FALSE]

  cor_vec <- rep(NA_real_, nrow(df))
  idx <- as.integer(rownames(Z))
  cor_vec[idx] <- as.numeric(Z %*% beta_corr)

  df$corr_score <- cor_vec
  df$tot_score <- df$risk_score + df$corr_score
  df
}

imp_long2 <- imp_long |>
  group_split(.imp) |>
  lapply(add_corr) |>
  bind_rows()

imp_tot <- as.mids(imp_long2)
```

12 Population attributable fraction

```
t_star <- 72

collapse_IpColest <- function(d, var = "IpColest_treatment2", newvar = "IpColest
_treat_bin") {
  x <- d[[var]]
  if (!is.factor(x)) x <- factor(x)
  lv <- levels(x)
  if (length(lv) < 3) stop("Mi aspetto 3 livelli in IpColest_treatment2.")
  d[[newvar]] <- as.integer(x %in% lv[2:3])
  d
}

form.sel.bin <- if (any(grepl("\\bIpColest_treatment2\\b", sel$predictors_fina
l))) {
  preds_bin <- sel$predictors_final
  preds_bin[preds_bin == "IpColest_treatment2"] <- "IpColest_treat_bin"
  as.formula(paste("Surv(Tempo_mesi, Evento_demenza) ~", paste(preds_bin, collap
se = " + ")))
} else {
  form.sel
}

candidati_finali <- c(
  "IpColest_treat_bin", "Alcol_yesno", "SocialPart2cl",
  "CognAtt_num_2cl", "Diabetes_2cl"
)

get_paf_one_exposure <- function(varname) {
  Q <- numeric(imp$m)
  U <- numeric(imp$m)

  for (i in 1:imp$m) {
    df <- complete(imp, i)
    df <- collapse_IpColest(df)

    xi <- df[[varname]]
    if (is.factor(xi)) {
      df[[varname]] <- as.integer(xi == levels(xi)[2])
    } else {
      u <- sort(na.omit(unique(xi)))
      if (length(u) != 2) stop(paste(varname, "non è binaria nell'imputazione",
i))
      if (!all(u %in% c(0, 1))) df[[varname]] <- as.integer(xi == max(u))
    }

    fit_i <- coxph(
      form.sel.bin,
      data = df,
```

```
      ties = "breslow",
      x = TRUE,
      y = TRUE,
      model = TRUE
    )

    AF_data <- df
    fit_i$call$data <- quote(AF_data)

    af_i <- AFcoxph(
      object = fit_i,
      data = AF_data,
      exposure = varname,
      times = t_star
    )

    Q[i] <- as.numeric(af_i$AF.est[1])
    U[i] <- as.numeric(af_i$AF.var[1])
  }

  pooled <- mice::pool.scalar(Q = Q, U = U, n = nrow(complete(imp, 1)))
  est <- pooled$qbar
  se <- sqrt(pooled$t)
  ciL <- est - qt(0.975, df = pooled$df) * se
  ciU <- est + qt(0.975, df = pooled$df) * se

  tibble(
    Variabile = varname,
    Definizione = if_else(varname == "IpColest_treat_bin", "liv.2+3 vs liv.1",
"2 livello vs 1"),
    PAF_t = t_star,
    PAF = est,
    Lower95 = ciL,
    Upper95 = ciU
  )
}

presenti <- candidati_finali[candidati_finali %in% all.vars(form.sel.bin)]

PAF_tbl <- bind_rows(lapply(presenti, get_paf_one_exposure)) |>
  arrange(desc(PAF))

PAF_tbl
```

```
## # A tibble: 5 × 6
##   Variabile          Definizione      PAF_t     PAF  Lower95   Upper95
##   <chr>              <chr>            <dbl>   <dbl>    <dbl>     <dbl>
## 1 Alcol_yesno        2 livello vs 1      72  0.258   0.0789   0.437
## 2 IpColest_treat_bin liv.2+3 vs liv.1    72  0.101  -0.0101   0.211
## 3 Diabetes_2cl       2 livello vs 1      72  0.0717 -0.00446  0.148
## 4 CognAtt_num_2cl    2 livello vs 1      72 -0.0538 -0.107   -0.000733
## 5 SocialPart2cl      2 livello vs 1      72 -0.106  -0.169   -0.0442
```

13 Analisi parallela: modello completo con fattori dicotomici e PAF a 144 mesi

Questa sezione integra l'analisi parallela basata sul dataset `invece_risk_factors_pnrr.sav` e su un set di predittori modificabile che include variabili già dicotomizzate o ricodificate, come `Obesity`, `Physical_inactivity2` e `Fumo_yesno`.

Per evitare di sovrascrivere gli oggetti della pipeline principale, tutti gli oggetti di questa analisi sono identificati con suffisso `_full`.

13.1 Import e pre-processing dedicati

```
PNRR_full <- read_sav(
  "W:/LAB NEUROPSICOLOGIA/Prev-Ita-Dem_PNRR/Risk score/Analisi_MR/invece_risk_fa
ctors_pnrr.sav"
) |>
  zap_labels() |>
  mutate(across(where(is.labelled), as_factor)) |>
  mutate(event = as.integer(Evento_demenza == 1))

fattori_full <- c(
  "Alcol_yesno", "CognAtt_num_2cl", "DEP",
  "Diabetes_2cl", "HMedDiet", "Hypert_yesno",
  "IpColest_treatment2", "SocialPart2cl", "Solitudi1",
  "CIRS_cardio", "Gender", "APOE_E4carrier", "Obesity",
  "Physical_inactivity2", "Fumo_yesno", "stroke1"
)

PNRR_full <- PNRR_full |>
  mutate(across(all_of(fattori_full), as_factor))
```

13.2 Variabili e imputazione multipla

```
vars_model_full <- c(
  "Tempo_mesi", "Evento_demenza",
  "Obesity", "Hypert_yesno", "IpColest_treatment2",
  "Alcol_yesno", "Fumo_yesno", "Physical_inactivity2",
  "SocialPart2cl", "Solitudi1", "CognAtt_num_2cl",
  "DEP", "Diabetes_2cl",
  "Age_1", "Gender", "Education",
  "APOE_E4carrier", "stroke1", "CIRS_cardio"
)

set.seed(2025)
ini_full <- mice(PNRR_full[vars_model_full], maxit = 0)
ini_full$method[c("Tempo_mesi", "Evento_demenza")] <- ""

imp_full <- mice(
  PNRR_full[vars_model_full],
  m = 20,
  maxit = 20,
  method = ini_full$method,
  predictorMatrix = ini_full$predictorMatrix,
  printFlag = FALSE
)

imp_long_full_model <- complete(imp_full, action = "long", include = TRUE)
imp_long_full_no0 <- imp_long_full_model |>
  filter(.imp != 0)
```

13.3 Modello Cox completo

```
form_full_paf <- Surv(Tempo_mesi, Evento_demenza) ~
  Obesity + Hypert_yesno + IpColest_treatment2 +
  Alcol_yesno + Physical_inactivity2 + SocialPart2cl +
  Solitudi1 + CognAtt_num_2cl + DEP + Diabetes_2cl + Fumo_yesno

predictor_full_paf <- c(
  "Obesity", "Hypert_yesno", "IpColest_treatment2",
  "Alcol_yesno", "Fumo_yesno", "Physical_inactivity2",
  "SocialPart2cl", "Solitudi1", "CognAtt_num_2cl",
  "DEP", "Diabetes_2cl"
)

fits_full_paf <- lapply(1:imp_full$m, function(i) {
  d <- complete(imp_full, i)
  coxph(form_full_paf, data = d, method = "breslow")
})

pooled_full_paf <- mice::pool(fits_full_paf)
coef_all_full_paf <- summary(pooled_full_paf)

HR_tbl_full_paf <- coef_all_full_paf |>
  mutate(
    HR = exp(estimate),
    Lower95 = exp(estimate - 1.96 * std.error),
    Upper95 = exp(estimate + 1.96 * std.error)
  ) |>
  select(term, HR, Lower95, Upper95, p.value)

HR_tbl_full_paf
```

```
##                    term        HR   Lower95   Upper95     p.value
## 1              Obesity1 1.0113883 0.7081925 1.4443901 0.950427669
## 2          Hypert_yesno1 0.9513315 0.6840679 1.3230143 0.767228168
## 3   IpColest_treatment21 1.2431592 0.8640899 1.7885233 0.242641342
## 4   IpColest_treatment22 1.5108056 0.9731687 2.3454654 0.067836392
## 5           Alcol_yesno1 1.6116268 1.1447792 2.2688576 0.006947045
## 6  Physical_inactivity21 1.1225015 0.8240163 1.5291077 0.464814377
## 7         SocialPart2cl1 0.5436036 0.3513398 0.8410797 0.006898182
## 8             Solitudi11 1.1719216 0.6948714 1.9764814 0.552762028
## 9       CognAtt_num_2cl1 0.6848702 0.4394227 1.0674168 0.096510555
## 10                 DEP1 1.2266141 0.5436573 2.7675199 0.623473392
## 11         Diabetes_2cl1 1.4606991 1.0195462 2.0927366 0.040499032
## 12           Fumo_yesno1 0.9356013 0.5481296 1.5969761 0.807532385
```

13.4 Punteggio derivato dal modello completo

```
betas_full_paf <- setNames(coef_all_full_paf$estimate, coef_all_full_paf$term)
beta_min_full_paf <- min(abs(betas_full_paf))
points_full_paf <- round(betas_full_paf / beta_min_full_paf)

score_tbl_full_paf <- tibble(
  Variabile = names(points_full_paf),
  BetaPool = round(betas_full_paf, 3),
  Punti = points_full_paf
)

score_tbl_full_paf
```

```
## # A tibble: 12 × 3
##    Variabile             BetaPool Punti
##    <chr>                    <dbl> <dbl>
##  1 Obesity1                 0.011     1
##  2 Hypert_yesno1           -0.05     -4
##  3 IpColest_treatment21     0.218    19
##  4 IpColest_treatment22     0.413    36
##  5 Alcol_yesno1             0.477    42
##  6 Physical_inactivity21    0.116    10
##  7 SocialPart2cl1          -0.61    -54
##  8 Solitudi11               0.159    14
##  9 CognAtt_num_2cl1        -0.379   -33
## 10 DEP1                     0.204    18
## 11 Diabetes_2cl1            0.379    33
## 12 Fumo_yesno1             -0.067    -6
```

13.5 Population attributable fraction a 144 mesi

```
t_star_full_paf <- 144

form_full_paf_bin <- if (any(grepl("\\bIpColest_treatment2\\b", predictor_full_p
af))) {
  preds_bin <- predictor_full_paf
  preds_bin[preds_bin == "IpColest_treatment2"] <- "IpColest_treat_bin"
  as.formula(paste("Surv(Tempo_mesi, Evento_demenza) ~", paste(preds_bin, collap
se = " + ")))
} else {
  form_full_paf
}

candidati_full_paf <- c(
  "IpColest_treat_bin", "Obesity", "Hypert_yesno",
  "Alcol_yesno", "Fumo_yesno", "Physical_inactivity2",
  "SocialPart2cl", "Solitudi1", "CognAtt_num_2cl",
  "DEP", "Diabetes_2cl"
)

get_paf_one_exposure_full <- function(varname) {
  Q <- numeric(imp_full$m)
  U <- numeric(imp_full$m)

  for (i in 1:imp_full$m) {
    df <- complete(imp_full, i)
    df <- collapse_IpColest(df)

    stopifnot(varname %in% names(df))
    xi <- df[[varname]]

    if (is.factor(xi)) {
      df[[varname]] <- as.integer(xi == levels(xi)[2])
    } else {
      u <- sort(na.omit(unique(xi)))
      if (length(u) != 2) stop(paste(varname, "non è binaria nell'imputazione",
i))
      if (!all(u %in% c(0, 1))) df[[varname]] <- as.integer(xi == max(u))
    }

    fit_i <- coxph(
      form_full_paf_bin,
      data = df,
      ties = "breslow",
      x = TRUE,
      y = TRUE,
      model = TRUE
```

```
    )

    AF_data <- df
    fit_i$call$data <- quote(AF_data)

    af_i <- AFcoxph(
      object = fit_i,
      data = AF_data,
      exposure = varname,
      times = t_star_full_paf
    )

    Q[i] <- as.numeric(af_i$AF.est[1])
    U[i] <- as.numeric(af_i$AF.var[1])
  }

  pooled <- mice::pool.scalar(Q = Q, U = U, n = nrow(complete(imp_full, 1)))
  est <- pooled$qbar
  se <- sqrt(pooled$t)
  ciL <- est - qt(0.975, df = pooled$df) * se
  ciU <- est + qt(0.975, df = pooled$df) * se

  tibble(
    Variabile = varname,
    Definizione = if_else(varname == "IpColest_treat_bin", "liv.2+3 vs liv.1",
"2 livello vs 1"),
    PAF_t = t_star_full_paf,
    PAF = est,
    Lower95 = ciL,
    Upper95 = ciU
  )
}

presenti_full_paf <- candidati_full_paf[candidati_full_paf %in% all.vars(form_fu
ll_paf_bin)]

PAF_tbl_full_paf <- bind_rows(lapply(presenti_full_paf, get_paf_one_exposure_ful
l)) |>
  arrange(desc(PAF))

PAF_tbl_full_paf
```

```
## # A tibble: 11 × 6
##    Variabile            Definizione      PAF_t      PAF  Lower95  Upper95
##    <chr>                <chr>            <dbl>    <dbl>    <dbl>    <dbl>
##  1 Alcol_yesno          2 livello vs 1     144  0.253    0.0843   0.422
##  2 IpColest_treat_bin   liv.2+3 vs liv.1   144  0.0938  -0.00926  0.197
##  3 Diabetes_2cl         2 livello vs 1     144  0.0663  -0.00328  0.136
##  4 Physical_inactivity2 2 livello vs 1     144  0.0543  -0.0784   0.187
##  5 Solitudi1            2 livello vs 1     144  0.0179  -0.0441   0.0799
##  6 DEP                  2 livello vs 1     144  0.0116  -0.0344   0.0576
##  7 Obesity              2 livello vs 1     144  0.00335 -0.0884   0.0951
##  8 Fumo_yesno           2 livello vs 1     144 -0.00563 -0.0491   0.0379
##  9 Hypert_yesno         2 livello vs 1     144 -0.0328  -0.223    0.157
## 10 CognAtt_num_2cl      2 livello vs 1     144 -0.0494  -0.101    0.00201
## 11 SocialPart2cl        2 livello vs 1     144 -0.0959  -0.158   -0.0342
```

13.6 Descrittive delle variabili incluse nel PAF

```
PNRR_desc_full <- PNRR_full |>
  collapse_IpColest() |>
  mutate(
    Evento_demenza = factor(
      Evento_demenza,
      levels = c(0, 1),
      labels = c("No demenza", "Demenza incidente")
    ),
    IpColest_treat_bin = factor(
      IpColest_treat_bin,
      levels = c(0, 1),
      labels = c("liv.1", "liv.2+3")
    )
  )

vars_paf_full <- c(
  "IpColest_treat_bin", "Obesity", "Hypert_yesno",
  "Alcol_yesno", "Fumo_yesno", "Physical_inactivity2",
  "SocialPart2cl", "Solitudi1", "CognAtt_num_2cl",
  "DEP", "Diabetes_2cl"
)

PAF_descrittive_strat_full <- PNRR_desc_full |>
  select(Evento_demenza, all_of(vars_paf_full)) |>
  mutate(across(all_of(vars_paf_full), as.character)) |>
  pivot_longer(
    cols = all_of(vars_paf_full),
    names_to = "Variabile",
    values_to = "Categoria"
  ) |>
  filter(!is.na(Categoria)) |>
  group_by(Evento_demenza, Variabile, Categoria) |>
  summarise(n = n(), .groups = "drop_last") |>
  mutate(
    N_tot = sum(n),
    perc = round(100 * n / N_tot, 1)
  ) |>
  ungroup() |>
  arrange(Variabile, Evento_demenza, desc(n))

PAF_descrittive_strat_full
```

```
## # A tibble: 44 × 6
##    Evento_demenza    Variabile       Categoria     n N_tot  perc
##    <fct>             <chr>           <chr>     <int> <int> <dbl>
##  1 No demenza        Alcol_yesno     1           596   922  64.6
##  2 No demenza        Alcol_yesno     0           326   922  35.4
##  3 Demenza incidente Alcol_yesno     1           127   174  73
##  4 Demenza incidente Alcol_yesno     0            47   174  27
##  5 No demenza        CognAtt_num_2cl 0           750   926  81
##  6 No demenza        CognAtt_num_2cl 1           176   926  19
##  7 Demenza incidente CognAtt_num_2cl 0           150   174  86.2
##  8 Demenza incidente CognAtt_num_2cl 1            24   174  13.8
##  9 No demenza        DEP             0           839   884  94.9
## 10 No demenza        DEP             1            45   884   5.1
## # ℹ 34 more rows
```

14 Analisi stratificata per MMSE

```
orig_mmse <- PNRR |>
  mutate(.id = seq_len(n())) |>
  select(.id, MMSE_1, Code)

imp_long_mmse <- complete(imp_score, action = "long", include = TRUE) |>
  left_join(orig_mmse, by = ".id") |>
  mutate(
    MMSE_raw = as.numeric(MMSE_1),
    MMSE_cat = cut(
      MMSE_raw,
      breaks = c(-Inf, 24, 27, Inf),
      labels = c("≤24", "25–27", "≥28"),
      right = TRUE
    )
  )

imp_mmse <- as.mids(imp_long_mmse)

fit_strat <- with(
  imp_mmse,
  coxph(
    Surv(Tempo_mesi, Evento_demenza) ~
      risk_score + Age_1 + Gender + Education +
      APOE_E4carrier + stroke1 + CIRS_cardio2cl + strata(MMSE_cat),
    method = "breslow"
  )
)

pool_strat <- summary(pool(fit_strat)) |>
  mutate(
    HR = exp(estimate),
    L95 = exp(estimate - 1.96 * std.error),
    U95 = exp(estimate + 1.96 * std.error)
  )

pool_strat
```

```
##              term   estimate  std.error statistic       df     p.value
HR
## 1      risk_score 0.15734662 0.04108691 3.8296049 150.4667 0.000187901 1.1704
01
## 2          Age_1 0.08682533 0.06492971 1.3372204 151.0250 0.183161278 1.0907
06
## 3         Gender2 0.11955254 0.16898381 0.7074793 151.0215 0.480359034 1.1269
92
## 4       Education 0.01994223 0.02822569 0.7065277 151.0176 0.480948656 1.0201
42
## 5 APOE_E4carrier1 0.53703875 0.18274396 2.9387497 151.0305 0.003813127 1.7109
33
## 6 APOE_E4carrier2 1.90092907 0.73535867 2.5850366 151.0361 0.010683159 6.6921
09
## 7        stroke11 0.68125949 0.23588465 2.8881043 151.0294 0.004445513 1.9763
65
## 8 CIRS_cardio2cl1 0.12004015 0.18673181 0.6428479 151.0205 0.521297623 1.1275
42
##         L95       U95
## 1 1.0798437  1.268553
## 2 0.9603698  1.238731
## 3 0.8092433  1.569506
## 4 0.9652384  1.078169
## 5 1.1958540  2.447867
## 6 1.5834818 28.282184
## 7 1.2447386  3.138024
## 8 0.7819580  1.625856
```

14.1 Hazard ratio per strato MMSE

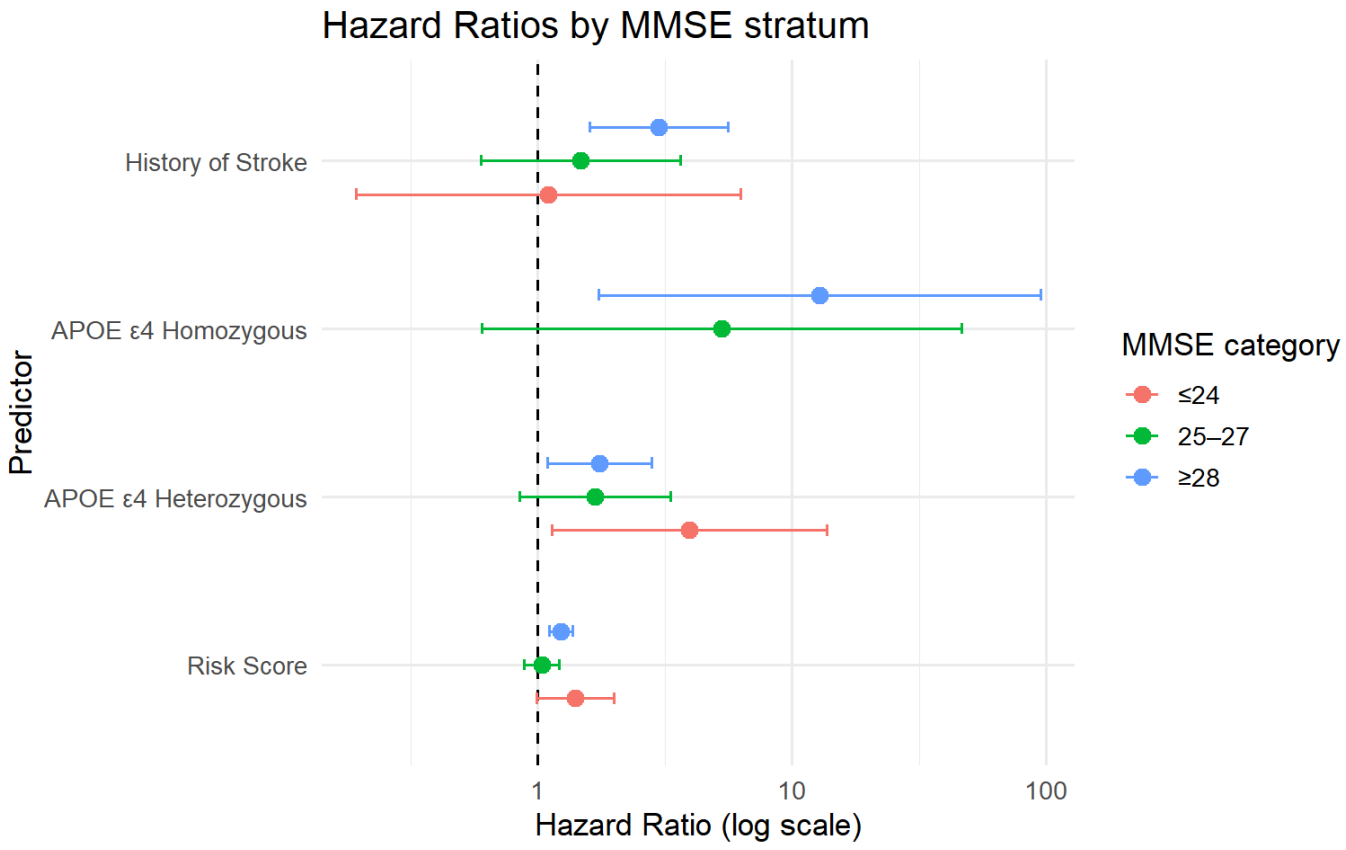
```
fit_one_level <- function(level_label) {
  fit <- with(
    imp_mmse,
    coxph(
      Surv(Tempo_mesi, Evento_demenza) ~
        risk_score + Age_1 + Gender + Education +
        APOE_E4carrier + stroke1 + CIRS_cardio2cl,
      subset = (MMSE_cat == level_label),
      method = "breslow"
    )
  )

  summary(pool(fit)) |>
    filter(term == "risk_score") |>
    mutate(
      MMSE_level = level_label,
      HR = exp(estimate),
      L95 = exp(estimate - 1.96 * std.error),
      U95 = exp(estimate + 1.96 * std.error)
    ) |>
    select(MMSE_level, HR, L95, U95, estimate, std.error, p.value, df)
}

res_mmse <- bind_rows(
  fit_one_level("≤24"),
  fit_one_level("25–27"),
  fit_one_level("≥28")
)

res_mmse
```

```
##   MMSE_level       HR       L95      U95   estimate  std.error     p.value
## 1        ≤24 1.400360 0.9871524 1.986531 0.33672955 0.17839816 0.087288437
## 2      25–27 1.035383 0.8834775 1.213408 0.03477171 0.08094955 0.669719573
## 3        ≥28 1.233654 1.1097626 1.371375 0.20998009 0.05399692 0.000209419
##         df
## 1 10.40000
## 2 42.01315
## 3 78.61397
```

14.2 Forest plot per strato MMSE

```
fit_all_terms <- function(level_label) {
  fit <- with(
    imp_mmse,
    coxph(
      Surv(Tempo_mesi, Evento_demenza) ~
        risk_score + Age_1 + Gender + Education +
        APOE_E4carrier + stroke1 + CIRS_cardio2cl,
      subset = (MMSE_cat == level_label),
      method = "breslow"
    )
  )

  summary(pool(fit)) |>
    mutate(
      MMSE_level = level_label,
      HR = exp(estimate),
      L95 = exp(estimate - 1.96 * std.error),
      U95 = exp(estimate + 1.96 * std.error)
    ) |>
    select(MMSE_level, term, estimate, std.error, HR, L95, U95, p.value)
}

res_all_mmse <- bind_rows(
  fit_all_terms("≤24"),
  fit_all_terms("25–27"),
  fit_all_terms("≥28")
) |>
  mutate(
    term = recode(
      term,
      "risk_score" = "Risk Score",
      "Age_1" = "Age",
      "Gender2" = "Female (vs Male)",
      "Education" = "Education (yrs)",
      "APOE_E4carrier1" = "APOE ε4 Heterozygous",
      "APOE_E4carrier2" = "APOE ε4 Homozygous",
      "stroke11" = "History of Stroke",
      "CIRS_cardio2cl1" = "Cardiovascular Comorbidity"
    )
  )

vars_interest <- c(
  "Risk Score", "APOE ε4 Heterozygous",
  "APOE ε4 Homozygous", "History of Stroke"
)
```

```r
forest_df <- res_all_mmse |>
  filter(term %in% vars_interest) |>
  mutate(
    term = factor(term, levels = vars_interest),
    MMSE_level = factor(MMSE_level, levels = c("≤24", "25–27", "≥28"))
  )

ggplot(forest_df, aes(x = HR, y = term, color = MMSE_level)) +
  geom_vline(xintercept = 1, linetype = 2) +
  geom_point(position = position_dodge(width = 0.6), size = 3) +
  geom_errorbarh(
    aes(xmin = L95, xmax = U95),
    position = position_dodge(width = 0.6),
    height = 0.2
  ) +
  scale_x_continuous(trans = "log10") +
  labs(
    x = "Hazard Ratio (log scale)",
    y = "Predictor",
    color = "MMSE category",
    title = "Hazard Ratios by MMSE stratum"
  ) +
  theme_minimal(base_size = 13)
```



14.3 Interazione tra risk score e MMSE

```
imp_long_inter <- imp_long_mmse |>
  mutate(MMSE_cat = factor(MMSE_cat, levels = c("≥28", "25–27", "≤24")))

imp_mmse_inter <- as.mids(imp_long_inter)

fit_inter <- with(
  imp_mmse_inter,
  coxph(
    Surv(Tempo_mesi, Evento_demenza) ~
      risk_score * MMSE_cat +
      Age_1 + Gender + Education + APOE_E4carrier + stroke1 + CIRS_cardio2cl +
      strata(MMSE_cat),
    method = "breslow"
  )
)

pool_inter <- summary(pool(fit_inter))
pool_inter
```

```
##                         term    estimate  std.error  statistic       df
## 1                 risk_score  0.20983683 0.05270780  3.9811345 146.2237
## 2              MMSE_cat25–27          NA         NA         NA       NA
## 3                MMSE_cat≤24          NA         NA         NA       NA
## 4                      Age_1  0.09331171 0.06507885  1.4338254 147.0192
## 5                    Gender2  0.12301037 0.17192219  0.7155003 147.0132
## 6                  Education  0.02058957 0.02836309  0.7259285 147.0108
## 7           APOE_E4carrier1  0.53600128 0.18292099  2.9302338 147.0294
## 8           APOE_E4carrier2  1.85320618 0.73733471  2.5133852 147.0364
## 9                  stroke11  0.68359099 0.24115011  2.8347114 147.0369
## 10          CIRS_cardio2cl1  0.12387623 0.18694541  0.6626332 147.0231
## 11 risk_score:MMSE_cat25–27 -0.17613784 0.09360032 -1.8818080 146.5781
## 12   risk_score:MMSE_cat≤24 -0.05617644 0.12497215 -0.4495116 146.9041
##         p.value
## 1  0.0001076845
## 2            NA
## 3            NA
## 4  0.1537454875
## 5  0.4754348011
## 6  0.4690367068
## 7  0.0039281899
## 8  0.0130357826
## 9  0.0052325513
## 10 0.5086029828
## 11 0.0618447280
## 12 0.6537251259
```

14.4 Numerosità per categoria MMSE

```
diag_mmse <- lapply(1:imp$m, function(i) {
  d <- complete(imp_mmse, i)
  d |>
    group_by(MMSE_cat) |>
    summarise(
      n = n(),
      events = sum(Evento_demenza == 1, na.rm = TRUE),
      .groups = "drop"
    )
}) |>
  bind_rows(.id = "imp")

summary_mmse <- diag_mmse |>
  group_by(MMSE_cat) |>
  summarise(
    n_mean = mean(n),
    n_range = paste0(min(n), "–", max(n)),
    events_mean = mean(events),
    events_range = paste0(min(events), "–", max(events)),
    .groups = "drop"
  )

summary_mmse
```

```
## # A tibble: 4 × 5
##   MMSE_cat n_mean n_range events_mean events_range
##   <fct>     <dbl> <chr>         <dbl> <chr>
## 1 ≤24          41 41–41            20 20–20
## 2 25–27       225 225–225          52 52–52
## 3 ≥28         762 762–762          89 89–89
## 4 <NA>         72 72–72            13 13–13
```

15 Performance del risk score

15.1 Coefficiente, HR e intervalli di confidenza

```
coef_tbl <- summary(pool(fit_corr_1)) |>
  mutate(
    HR = exp(estimate),
    Lower95 = exp(estimate - 1.96 * std.error),
    Upper95 = exp(estimate + 1.96 * std.error)
  ) |>
  select(term, HR, Lower95, Upper95, p.value)

coef_tbl
```

```
##         term       HR  Lower95  Upper95      p.value
## 1 risk_score 1.228809 1.139682 1.324907 2.629987e-07
```

15.2 C-index

```
pool_scalar <- function(q, se) {
  q <- as.numeric(q)
  se <- as.numeric(se)
  keep <- is.finite(q) & is.finite(se)
  q <- q[keep]
  se <- se[keep]
  m <- length(q)

  if (m < 2) {
    return(list(Qbar = NA, SE = NA, L95 = NA, U95 = NA, df = NA, m = m))
  }

  Qbar <- mean(q)
  Ubar <- mean(se^2)
  B <- stats::var(q)
  Tvar <- Ubar + (1 + 1 / m) * B
  SE <- sqrt(Tvar)
  df <- (m - 1) * (1 + Ubar / ((1 + 1 / m) * B))^2
  L95 <- Qbar - 1.96 * SE
  U95 <- Qbar + 1.96 * SE

  list(Qbar = Qbar, SE = SE, L95 = L95, U95 = U95, df = df, m = m)
}

cindex_list <- lapply(1:imp$m, function(i) {
  d <- complete(imp_score, i)

  ci <- tryCatch(
    Hmisc::rcorr.cens(d$risk_score, Surv(d$Tempo_mesi, d$Evento_demenza)),
    error = function(e) NULL
  )

  if (is.null(ci)) return(c(cindex = NA, se = NA))
  c(cindex = ci["C Index"], se = ci["S.D."])
})

cidx <- sapply(cindex_list, `[`, 1)
cse <- sapply(cindex_list, `[`, 2)

cindex_pool <- pool_scalar(cidx, cse)
cindex_pool
```

```
## $Qbar
## [1] 0.3851425
##
## $SE
## [1] 0.04525739
##
## $L95
## [1] 0.296438
##
## $U95
## [1] 0.473847
##
## $df
## [1] 12846077
##
## $m
## [1] 20
```

15.3 R quadrato di Royston-

Sauerbrei/Nagelkerke

```
nagelkerke_M1 <- function(d) {
  fit <- coxph(Surv(Tempo_mesi, Evento_demenza) ~ risk_score, data = d, method =
"breslow")
  ll0 <- logLik(update(fit, . ~ 1))
  ll1 <- logLik(fit)
  n <- fit$n
  1 - exp((2 / n) * (ll0 - ll1))
}

R2_vec <- sapply(1:imp_score$m, function(i) nagelkerke_M1(complete(imp_score,
i)))

qbar <- mean(R2_vec)
b <- var(R2_vec)
ubar <- 0
t <- ubar + (1 + 1 / imp_score$m) * b
se <- sqrt(t)

R2_pool <- tibble(
  R2 = qbar,
  SE = se,
  Lower95 = qbar - 1.96 * se,
  Upper95 = qbar + 1.96 * se
)

R2_pool
```

```
## # A tibble: 1 × 4
##       R2       SE Lower95 Upper95
##    <dbl>    <dbl>   <dbl>   <dbl>
## 1 0.0274 0.000601  0.0262  0.0286
```

15.4 Assunzione di proporzionalità dei rischi

```
with(imp_score, {
  fit <- coxph(Surv(Tempo_mesi, Evento_demenza) ~ risk_score)
  cox.zph(fit)
})
```

```
## call :
## with.mids(data = imp_score, expr = {
##     fit <- coxph(Surv(Tempo_mesi, Evento_demenza) ~ risk_score)
##     cox.zph(fit)
## })
##
## call1 :
## mice(data = data, m = m, where = where, maxit = 0, remove.collinear = FALSE,
##     allow.na = TRUE)
##
## nmis :
##  [1]  0  0  0 44  0 23  4  0  1  2  0  4  0 48 13  0  0  0  0  1  0  0 39  0
##
## analyses :
## [[1]]
##             chisq df    p
## risk_score 0.0351  1 0.85
## GLOBAL     0.0351  1 0.85
##
## [[2]]
##             chisq df    p
## risk_score 0.0336  1 0.85
## GLOBAL     0.0336  1 0.85
##
## [[3]]
##             chisq df   p
## risk_score 0.0168  1 0.9
## GLOBAL     0.0168  1 0.9
##
## [[4]]
##             chisq df    p
## risk_score 0.0762  1 0.78
## GLOBAL     0.0762  1 0.78
##
## [[5]]
##             chisq df    p
## risk_score 0.0299  1 0.86
## GLOBAL     0.0299  1 0.86
##
## [[6]]
##             chisq df    p
## risk_score 0.0585  1 0.81
## GLOBAL     0.0585  1 0.81
##
## [[7]]
##             chisq df    p
## risk_score 0.0198  1 0.89
## GLOBAL     0.0198  1 0.89
```

```
##
## [[8]]
##             chisq df    p
## risk_score 0.0443  1 0.83
## GLOBAL     0.0443  1 0.83
##
## [[9]]
##             chisq df    p
## risk_score 0.0821  1 0.77
## GLOBAL     0.0821  1 0.77
##
## [[10]]
##            chisq df    p
## risk_score 0.104  1 0.75
## GLOBAL     0.104  1 0.75
##
## [[11]]
##             chisq df    p
## risk_score 0.0239  1 0.88
## GLOBAL     0.0239  1 0.88
##
## [[12]]
##             chisq df    p
## risk_score 0.0551  1 0.81
## GLOBAL     0.0551  1 0.81
##
## [[13]]
##             chisq df    p
## risk_score 0.0431  1 0.84
## GLOBAL     0.0431  1 0.84
##
## [[14]]
##             chisq df    p
## risk_score 0.0555  1 0.81
## GLOBAL     0.0555  1 0.81
##
## [[15]]
##             chisq df    p
## risk_score 0.0363  1 0.85
## GLOBAL     0.0363  1 0.85
##
## [[16]]
##             chisq df    p
## risk_score 0.0395  1 0.84
## GLOBAL     0.0395  1 0.84
##
## [[17]]
##             chisq df    p
```

```
## risk_score 0.0426  1 0.84
## GLOBAL     0.0426  1 0.84
##
## [[18]]
##              chisq df    p
## risk_score 0.0328  1 0.86
## GLOBAL     0.0328  1 0.86
##
## [[19]]
##              chisq df    p
## risk_score 0.0598  1 0.81
## GLOBAL     0.0598  1 0.81
##
## [[20]]
##              chisq df    p
## risk_score 0.0376  1 0.85
## GLOBAL     0.0376  1 0.85
```

15.5 Distribuzione dello score

```
stats_by_imp <- lapply(1:imp_score$m, function(i) {
  v <- complete(imp_score, i)$risk_score
  tibble(
    mean = mean(v, na.rm = TRUE),
    var = var(v, na.rm = TRUE),
    median = median(v, na.rm = TRUE),
    q25 = quantile(v, .25, na.rm = TRUE),
    q75 = quantile(v, .75, na.rm = TRUE),
    min = min(v, na.rm = TRUE),
    max = max(v, na.rm = TRUE)
  )
}) |>
  bind_rows()

qbar <- mean(stats_by_imp$mean)
ubar <- mean(stats_by_imp$var)
b <- var(stats_by_imp$mean)
tvar <- ubar + (1 + 1 / imp_score$m) * b
sd_pool <- sqrt(tvar)

desc_pool <- tibble(
  Mean = round(qbar, 2),
  SD = round(sd_pool, 2),
  Median = round(mean(stats_by_imp$median), 2),
  Q1 = round(mean(stats_by_imp$q25), 2),
  Q3 = round(mean(stats_by_imp$q75), 2),
  Min = round(min(stats_by_imp$min), 1),
  Max = round(max(stats_by_imp$max), 1)
)

desc_pool
```

```
## # A tibble: 1 × 7
##    Mean    SD Median    Q1    Q3   Min   Max
##   <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  1.13   2.1      2     0     2    -5     6
```

```
freq_tbl <- imp_long |>
  filter(.imp != 0) |>
  count(risk_score) |>
  mutate(perc = round(100 * n / sum(n), 1)) |>
  arrange(risk_score)

freq_tbl
```

```
## # A tibble: 12 × 3
##    risk_score     n  perc
##         <dbl> <int> <dbl>
##  1         -5   220   1
##  2         -4    40   0.2
##  3         -3  1365   6.2
##  4         -2   885   4
##  5         -1  1686   7.7
##  6          0  4398  20
##  7          1  1679   7.6
##  8          2  6893  31.3
##  9          3  2175   9.9
## 10          4  1798   8.2
## 11          5   700   3.2
## 12          6   161   0.7
```

16 Analisi univariata

```
vars_uni <- c(
  "BMI1_NEW", "Hypert_yesno", "IpColest_treatment2",
  "Alcol_yesno", "PA_freq_1", "SocContact_sum",
  "SocialPart2cl", "Solitudi1", "CognAtt_num_2cl",
  "DEP", "Diabetes_2cl", "HMedDiet"
)

uni_tbl <- map_dfr(vars_uni, function(v) {
  uni_fit <- with(
    imp,
    {
      f_uni <- reformulate(v, response = "Surv(Tempo_mesi, Evento_demenza)")
      coxph(f_uni, method = "breslow")
    }
  )

  res <- summary(pool(uni_fit))

  tibble(
    var = v,
    HR = exp(res$estimate),
    Lower95 = exp(res$estimate - 1.96 * res$std.error),
    Upper95 = exp(res$estimate + 1.96 * res$std.error),
    p.value = res$p.value
  )
})

uni_tbl
```

```
## # A tibble: 15 × 5
##    var                    HR Lower95 Upper95 p.value
##    <chr>               <dbl>   <dbl>   <dbl>   <dbl>
##  1 BMI1_NEW            1.00    0.972   1.04  0.786
##  2 Hypert_yesno        1.02    0.742   1.39  0.924
##  3 IpColest_treatment2 1.37    0.964   1.94  0.0815
##  4 IpColest_treatment2 1.50    0.972   2.32  0.0689
##  5 Alcol_yesno         1.45    1.04    2.03  0.0311
##  6 PA_freq_1           0.959   0.905   1.02  0.162
##  7 SocContact_sum      0.993   0.951   1.04  0.764
##  8 SocialPart2cl       0.513   0.336   0.784 0.00238
##  9 Solitudi1           1.32    0.849   2.04  0.220
## 10 CognAtt_num_2cl     0.579   0.376   0.892 0.0141
## 11 DEP                 1.25    0.633   2.45  0.525
## 12 Diabetes_2cl        1.52    1.07    2.15  0.0193
## 13 HMedDiet            0.600   0.252   1.43  0.250
## 14 HMedDiet            0.547   0.239   1.25  0.154
## 15 HMedDiet            0.706   0.301   1.65  0.424
```

17 Time-dependent ROC e cut-off

```
feasible_time <- function(d, t, min_cases = 15, min_controls = 15) {
  cases <- sum(d$Evento_demenza == 1 & d$Tempo_mesi <= t)
  controls <- sum(d$Tempo_mesi > t)
  (cases >= min_cases) && (controls >= min_controls)
}

one_survroc <- function(d, t) {
  sr <- tryCatch(
    survivalROC(
      Stime = d$Tempo_mesi,
      status = d$Evento_demenza,
      marker = d$risk_score,
      predict.time = t,
      method = "KM"
    ),
    error = function(e) NULL
  )

  if (is.null(sr) || !is.finite(sr$AUC)) {
    return(c(AUC = NA, cutoff = NA, sens = NA, spec = NA, youden = NA, ppv = NA,
npv = NA))
  }

  TPR <- as.numeric(sr$TP)
  FPR <- as.numeric(sr$FP)
  CUT <- as.numeric(sr$cut.values)
  SPEC <- 1 - FPR

  J <- TPR + SPEC - 1
  j <- which.max(J)

  sens <- TPR[j]
  spec <- SPEC[j]
  cutoff <- CUT[j]

  prev <- mean(d$Evento_demenza == 1 & d$Tempo_mesi <= t)
  ppv <- if (is.finite(prev)) (sens * prev) / (sens * prev + (1 - spec) * (1 - p
rev)) else NA_real_
  npv <- if (is.finite(prev)) (spec * (1 - prev)) / ((1 - sens) * prev + spec *
(1 - prev)) else NA_real_

  c(
    AUC = as.numeric(sr$AUC),
    cutoff = cutoff,
    sens = sens,
    spec = spec,
    youden = J[j],
    ppv = ppv,
```

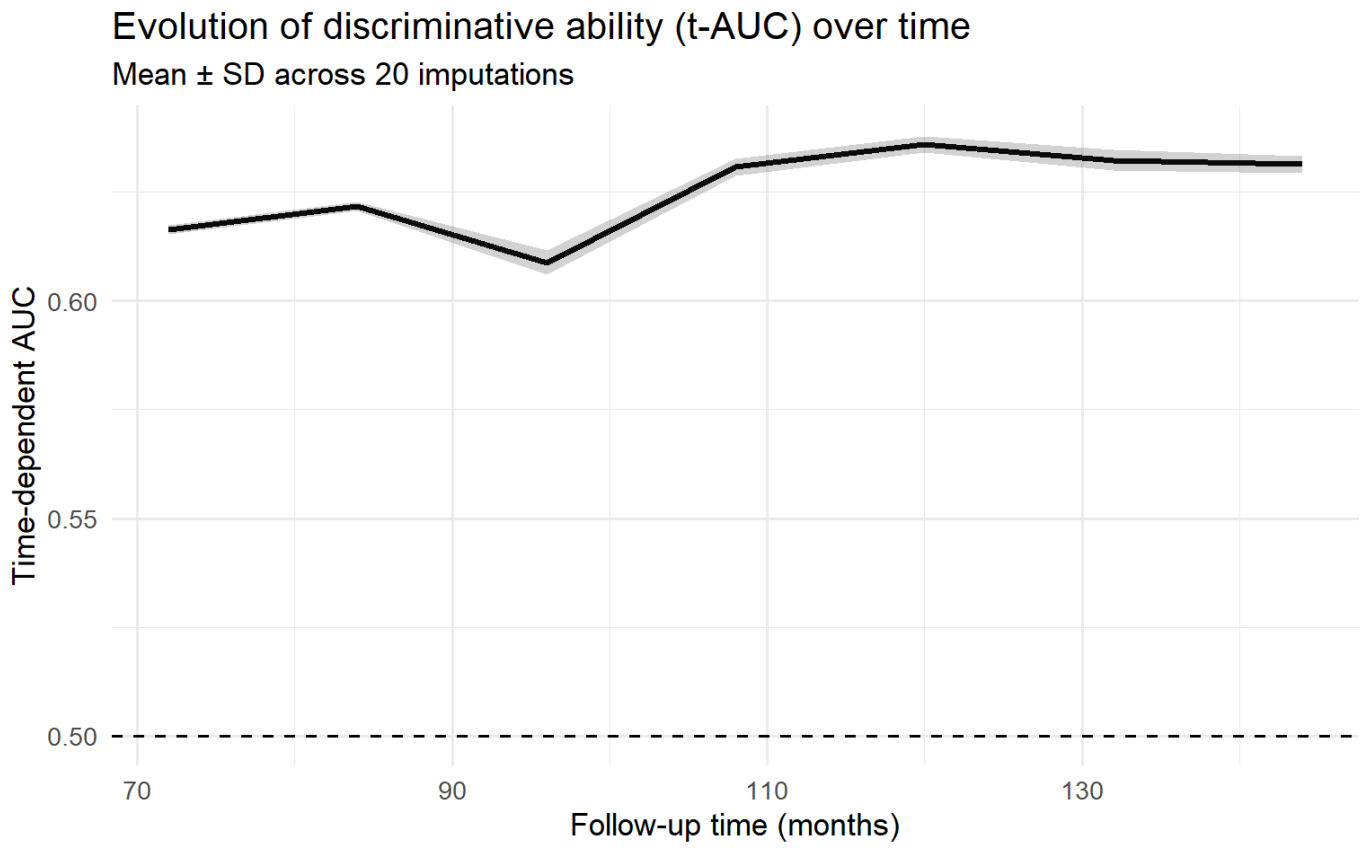
```
    npv = npv
  )
}

eval_times <- c(36, 72, 96, 120, 144, 153)

res_list <- lapply(eval_times, function(t) {
  mat <- do.call(rbind, lapply(1:imp$m, function(i) {
    d <- complete(imp_score, i)
    d <- d[is.finite(d$Tempo_mesi) & !is.na(d$Evento_demenza) & is.finite(d$risk
_score), ]
    if (!feasible_time(d, t)) {
      return(c(AUC = NA, cutoff = NA, sens = NA, spec = NA, youden = NA, ppv = N
A, npv = NA))
    }
    one_survroc(d, t)
  }))

  colnames(mat) <- c("AUC", "cutoff", "sens", "spec", "youden", "ppv", "npv")

  data.frame(
    time = t,
    AUC_mean = mean(mat[, "AUC"], na.rm = TRUE),
    AUC_sd = sd(mat[, "AUC"], na.rm = TRUE),
    cutoff_med = median(mat[, "cutoff"], na.rm = TRUE),
    Sens_mean = mean(mat[, "sens"], na.rm = TRUE),
    Spec_mean = mean(mat[, "spec"], na.rm = TRUE),
    Youden_mean = mean(mat[, "youden"], na.rm = TRUE),
    PPV_mean = mean(mat[, "ppv"], na.rm = TRUE),
    NPV_mean = mean(mat[, "npv"], na.rm = TRUE),
    n_imp_used = sum(is.finite(mat[, "AUC"]))
  )
})

auc_table <- bind_rows(res_list)
```

```
##   time  AUC_mean      AUC_sd cutoff_med Sens_mean Spec_mean Youden_mean
## 1   36 0.5787361 0.001471532          3 0.2510028 0.8815819   0.1325847
## 2   72 0.6163856 0.001053419          0 0.7877138 0.4024337   0.1901475
## 3   96 0.6087823 0.002795969          0 0.7400135 0.4359483   0.1759617
## 4  120 0.6360154 0.001914540          1 0.7270068 0.4972046   0.2242114
## 5  144 0.6314009 0.002022716          1 0.7246379 0.4820463   0.2066842
## 6  153 0.6719829 0.002225291          2 0.4053892 0.8950334   0.3004226
##     PPV_mean  NPV_mean n_imp_used
## 1 0.03777397 0.9845102         20
## 2 0.07414646 0.9689474         20
## 3 0.09683186 0.9537669         20
## 4 0.15042114 0.9370037         20
## 5 0.16499699 0.9256757         20
## 6 0.41886286 0.8896942         20
```

17.1 Andamento della AUC nel tempo

```
eval_times_cont <- seq(72, 153, by = 12)

get_auc <- function(d, t) {
  sr <- tryCatch(
    survivalROC(
      Stime = d$Tempo_mesi,
      status = d$Evento_demenza,
      marker = d$risk_score,
      predict.time = t,
      method = "KM"
    ),
    error = function(e) NULL
  )

  if (is.null(sr) || !is.finite(sr$AUC)) return(NA_real_)
  as.numeric(sr$AUC)
}

auc_list <- lapply(1:imp$m, function(i) {
  d <- complete(imp_score, i)
  sapply(eval_times_cont, function(t) get_auc(d, t))
})

auc_mat <- do.call(rbind, auc_list)

auc_df <- data.frame(
  time = eval_times_cont,
  AUC_mean = apply(auc_mat, 2, mean, na.rm = TRUE),
  AUC_sd = apply(auc_mat, 2, sd, na.rm = TRUE),
  n_imp = imp$m
) |>
  mutate(
    SE = AUC_sd / sqrt(n_imp),
    L95 = AUC_mean - 1.96 * SE,
    U95 = AUC_mean + 1.96 * SE
  )

ggplot(auc_df, aes(x = time, y = AUC_mean)) +
  geom_line(size = 1.2) +
  geom_ribbon(aes(ymin = AUC_mean - AUC_sd, ymax = AUC_mean + AUC_sd), alpha =
0.2) +
  geom_hline(yintercept = 0.5, linetype = 2) +
  labs(
    x = "Follow-up time (months)",
    y = "Time-dependent AUC",
    title = "Evolution of discriminative ability (t-AUC) over time",
```

```
    subtitle = "Mean ± SD across 20 imputations"
  ) +
  theme_minimal(base_size = 13)
```



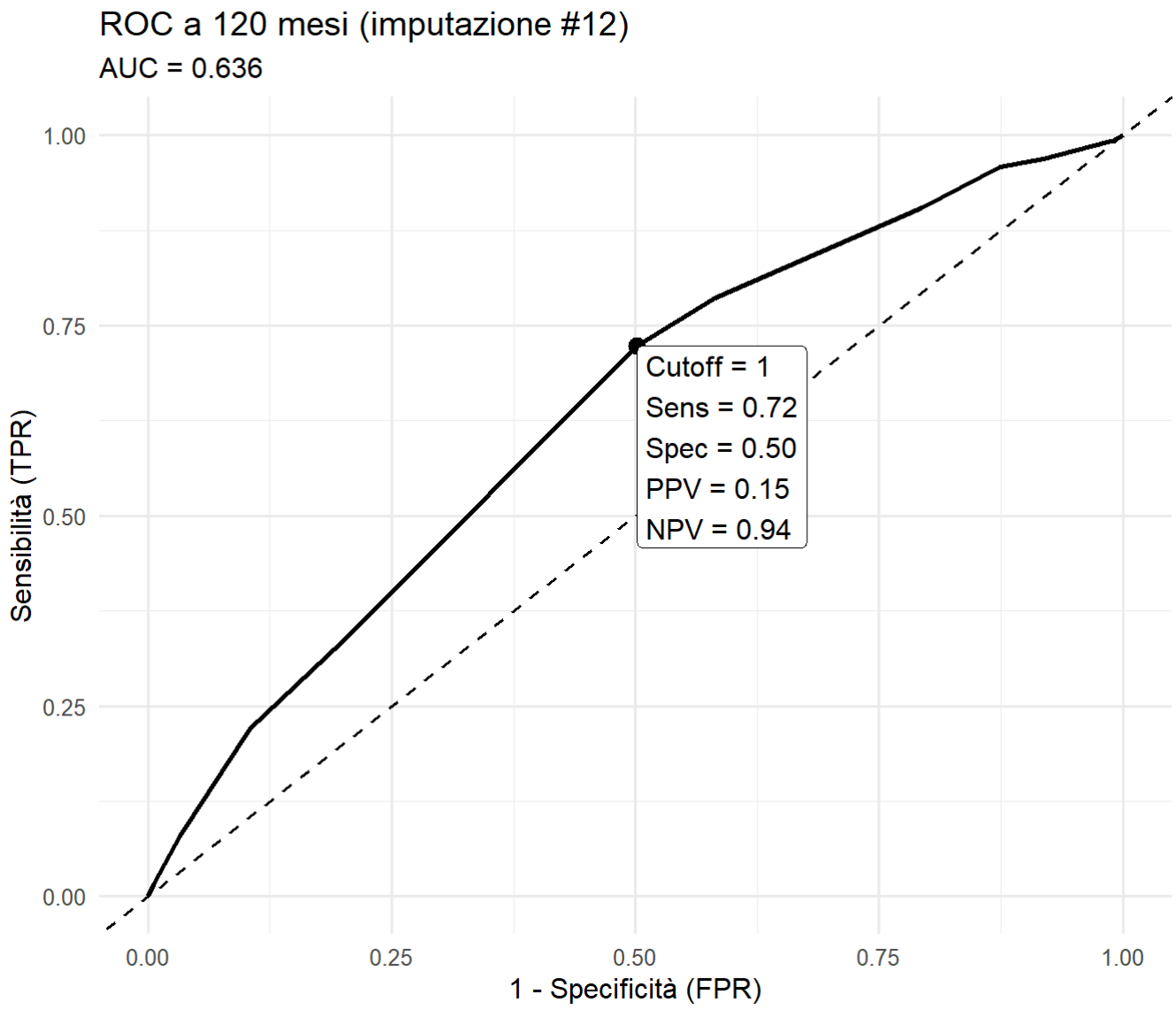

```
mean_auc_dev <- mean(auc_df$AUC_mean, na.rm = TRUE)
se_auc_dev <- mean(auc_df$SE, na.rm = TRUE)

auc_summary <- tibble(
  Cohort = "Derivation (InveCe.Ab)",
  Mean_AUC = round(mean_auc_dev, 3),
  L95 = round(mean_auc_dev - 1.96 * se_auc_dev, 3),
  U95 = round(mean_auc_dev + 1.96 * se_auc_dev, 3)
)

auc_summary
```

```
## # A tibble: 1 × 4
##   Cohort                 Mean_AUC   L95   U95
##   <chr>                     <dbl> <dbl> <dbl>
## 1 Derivation (InveCe.Ab)    0.625 0.625 0.626
```

17.2 Curva ROC a un tempo selezionato

```
t_plot <- 120

auc_each_imp <- sapply(1:imp$m, function(i) {
  d <- complete(imp_score, i)
  if (!feasible_time(d, t_plot)) return(NA_real_)

  sr <- tryCatch(
    survivalROC(
      Stime = d$Tempo_mesi,
      status = d$Evento_demenza,
      marker = d$risk_score,
      predict.time = t_plot,
      method = "KM"
    ),
    error = function(e) NULL
  )

  if (is.null(sr) || !is.finite(sr$AUC)) return(NA_real_)
  as.numeric(sr$AUC)
})

if (all(is.na(auc_each_imp))) stop("Nessuna imputazione feasible per il tempo sc
elto.")

med_auc <- median(auc_each_imp, na.rm = TRUE)
i_best <- which.min(abs(auc_each_imp - med_auc))

d_best <- complete(imp_score, i_best)
sr_best <- survivalROC(
  Stime = d_best$Tempo_mesi,
  status = d_best$Evento_demenza,
  marker = d_best$risk_score,
  predict.time = t_plot,
  method = "KM"
)

TPR <- as.numeric(sr_best$TP)
FPR <- as.numeric(sr_best$FP)
SPEC <- 1 - FPR
CUT <- as.numeric(sr_best$cut.values)
J <- TPR + SPEC - 1
j <- which.max(J)

prev <- mean(d_best$Evento_demenza == 1 & d_best$Tempo_mesi <= t_plot)
sens_j <- TPR[j]
spec_j <- SPEC[j]
```

```
cut_j <- CUT[j]
ppv_j <- (sens_j * prev) / (sens_j * prev + (1 - spec_j) * (1 - prev))
npv_j <- (spec_j * (1 - prev)) / ((1 - sens_j) * prev + spec_j * (1 - prev))

roc_df <- data.frame(FPR = FPR, TPR = TPR)

ggplot(roc_df, aes(x = FPR, y = TPR)) +
  geom_abline(slope = 1, intercept = 0, linetype = 2) +
  geom_path(size = 1) +
  geom_point(aes(x = 1 - spec_j, y = sens_j), size = 3) +
  annotate(
    "label",
    x = 1 - spec_j,
    y = sens_j,
    hjust = 0,
    vjust = 1,
    label = paste0(
      "Cutoff = ", round(cut_j, 2),
      "\nSens = ", sprintf("%.2f", sens_j),
      "\nSpec = ", sprintf("%.2f", spec_j),
      "\nPPV = ", sprintf("%.2f", ppv_j),
      "\nNPV = ", sprintf("%.2f", npv_j)
    )
  ) +
  labs(
    title = paste0("ROC a ", t_plot, " mesi (imputazione #", i_best, ")"),
    subtitle = paste0("AUC = ", sprintf("%.3f", as.numeric(sr_best$AUC))),
    x = "1 - Specificità (FPR)",
    y = "Sensibilità (TPR)"
  ) +
  theme_minimal(base_size = 12)
```



18 Validazione esterna: coorte TRELONG

Questa sezione valuta il risk score nella coorte esterna TRELONG. Il dataset di validazione contiene lo score già calcolato (`RS`), il tempo di follow-up in mesi (`Time_mesi`) e l'indicatore di demenza incidente (`Evento_Demenza`).

```
TRELOG <- read_excel(
  "W:/LAB NEUROPSICOLOGIA/Prev-Ita-Dem_PNRR/Risk score/Analisi_MR/TRELONG dati r
ichiesti/TRELOG DATI per FGC_20250630d_MR_er.xlsx",
  sheet = "senza dubbi"
)

TRELOG <- TRELOG |>
  mutate(
    evento_demenza_val = as.integer(Evento_Demenza == 1)
  )
```

18.1 Modello Cox nella coorte di validazione

```
Val1 <- coxph(
  Surv(Time_mesi, evento_demenza_val) ~ RS,
  data = TRELOG,
  method = "breslow"
)

summary(Val1)
```

```
## Call:
## coxph(formula = Surv(Time_mesi, evento_demenza_val) ~ RS, data = TRELOG,
##     method = "breslow")
##
##   n= 248, number of events= 101
##
##       coef exp(coef) se(coef)     z Pr(>|z|)
## RS 0.16965   1.18489  0.06403 2.649  0.00806 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##    exp(coef) exp(-coef) lower .95 upper .95
## RS     1.185      0.844     1.045     1.343
##
## Concordance= 0.569  (se = 0.029 )
## Likelihood ratio test= 7.38  on 1 df,   p=0.007
## Wald test            = 7.02  on 1 df,   p=0.008
## Score (logrank) test = 7.01  on 1 df,   p=0.008
```

```
Val1_tbl <- broom::tidy(Val1) |>
  mutate(
    HR = exp(estimate),
    Lower95 = exp(estimate - 1.96 * std.error),
    Upper95 = exp(estimate + 1.96 * std.error)
  ) |>
  select(term, estimate, std.error, HR, Lower95, Upper95, p.value)

Val1_tbl
```

```
## # A tibble: 1 × 7
##   term  estimate std.error    HR Lower95 Upper95 p.value
##   <chr>    <dbl>     <dbl> <dbl>   <dbl>   <dbl>   <dbl>
## 1 RS       0.170    0.0640  1.18    1.05    1.34 0.00806
```

18.2 Time-dependent ROC nella coorte di

validazione

```
eval_times_val_table <- c(36, 72, 96, 120, 144)

one_survroc_val <- function(d, t) {
  sr <- tryCatch(
    survivalROC(
      Stime = d$Time_mesi,
      status = d$evento_demenza_val,
      marker = d$RS,
      predict.time = t,
      method = "KM"
    ),
    error = function(e) NULL
  )

  if (is.null(sr) || !is.finite(sr$AUC)) return(NULL)

  TPR <- as.numeric(sr$TP)
  FPR <- as.numeric(sr$FP)
  SPEC <- 1 - FPR
  CUT <- as.numeric(sr$cut.values)

  J <- TPR + SPEC - 1
  j <- which.max(J)

  sens <- TPR[j]
  spec <- SPEC[j]
  cutoff <- CUT[j]

  prev <- mean(d$evento_demenza_val == 1 & d$Time_mesi <= t)
  ppv <- (sens * prev) / (sens * prev + (1 - spec) * (1 - prev))
  npv <- (spec * (1 - prev)) / ((1 - sens) * prev + spec * (1 - prev))

  tibble(
    time = t,
    AUC = as.numeric(sr$AUC),
    cutoff = cutoff,
    Sens = sens,
    Spec = spec,
    Youden = J[j],
    PPV = ppv,
    NPV = npv
  )
}

roc_tbl_val <- bind_rows(
  lapply(eval_times_val_table, function(t) one_survroc_val(TRELOG, t))
```

```
)

roc_tbl_val
```

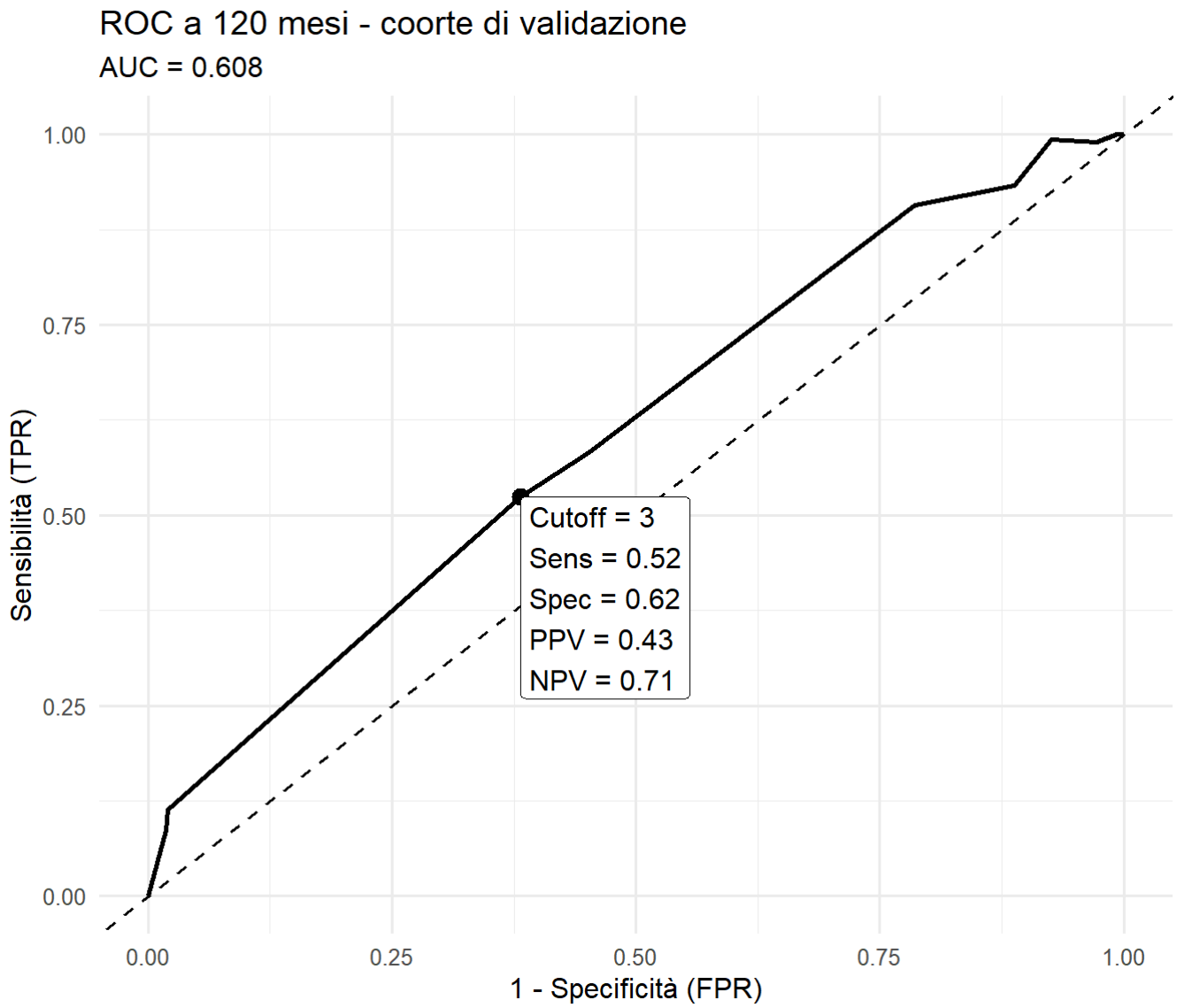
```
## # A tibble: 4 × 8
##    time   AUC cutoff  Sens  Spec Youden   PPV   NPV
##   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1    72 0.528      0 0.972 0.104 0.0757 0.156 0.956
## 2    96 0.564      3 0.515 0.589 0.105  0.335 0.752
## 3   120 0.608      3 0.524 0.618 0.142  0.426 0.706
## 4   144 0.842      2 0.715 1     0.715  1     0.836
```

18.3 Curva ROC nella coorte di validazione a un

tempo selezionato

```
t_plot_val <- 120

sr_best_val <- survivalROC(
  Stime = TRELOG$Time_mesi,
  status = TRELOG$evento_demenza_val,
  marker = TRELOG$RS,
  predict.time = t_plot_val,
  method = "KM"
)

TPR_val <- as.numeric(sr_best_val$TP)
FPR_val <- as.numeric(sr_best_val$FP)
SPEC_val <- 1 - FPR_val
CUT_val <- as.numeric(sr_best_val$cut.values)
J_val <- TPR_val + SPEC_val - 1
j_val <- which.max(J_val)

sens_j_val <- TPR_val[j_val]
spec_j_val <- SPEC_val[j_val]
cut_j_val <- CUT_val[j_val]
prev_val <- mean(TRELOG$evento_demenza_val == 1 & TRELOG$Time_mesi <= t_plot_va
l)
ppv_j_val <- (sens_j_val * prev_val) / (sens_j_val * prev_val + (1 - spec_j_val)
* (1 - prev_val))
npv_j_val <- (spec_j_val * (1 - prev_val)) / ((1 - sens_j_val) * prev_val + spec
_j_val * (1 - prev_val))

roc_df_val <- data.frame(FPR = FPR_val, TPR = TPR_val)

ggplot(roc_df_val, aes(x = FPR, y = TPR)) +
  geom_abline(slope = 1, intercept = 0, linetype = 2) +
  geom_path(size = 1) +
  geom_point(aes(x = 1 - spec_j_val, y = sens_j_val), size = 3) +
  annotate(
    "label",
    x = 1 - spec_j_val,
    y = sens_j_val,
    hjust = 0,
    vjust = 1,
    label = paste0(
      "Cutoff = ", round(cut_j_val, 2),
      "\nSens = ", sprintf("%.2f", sens_j_val),
      "\nSpec = ", sprintf("%.2f", spec_j_val),
      "\nPPV = ", sprintf("%.2f", ppv_j_val),
      "\nNPV = ", sprintf("%.2f", npv_j_val)
    )
```

```
) +
labs(
  title = paste0("ROC a ", t_plot_val, " mesi - coorte di validazione"),
  subtitle = paste0("AUC = ", sprintf("%.3f", as.numeric(sr_best_val$AUC))),
  x = "1 - Specificità (FPR)",
  y = "Sensibilità (TPR)"
) +
```

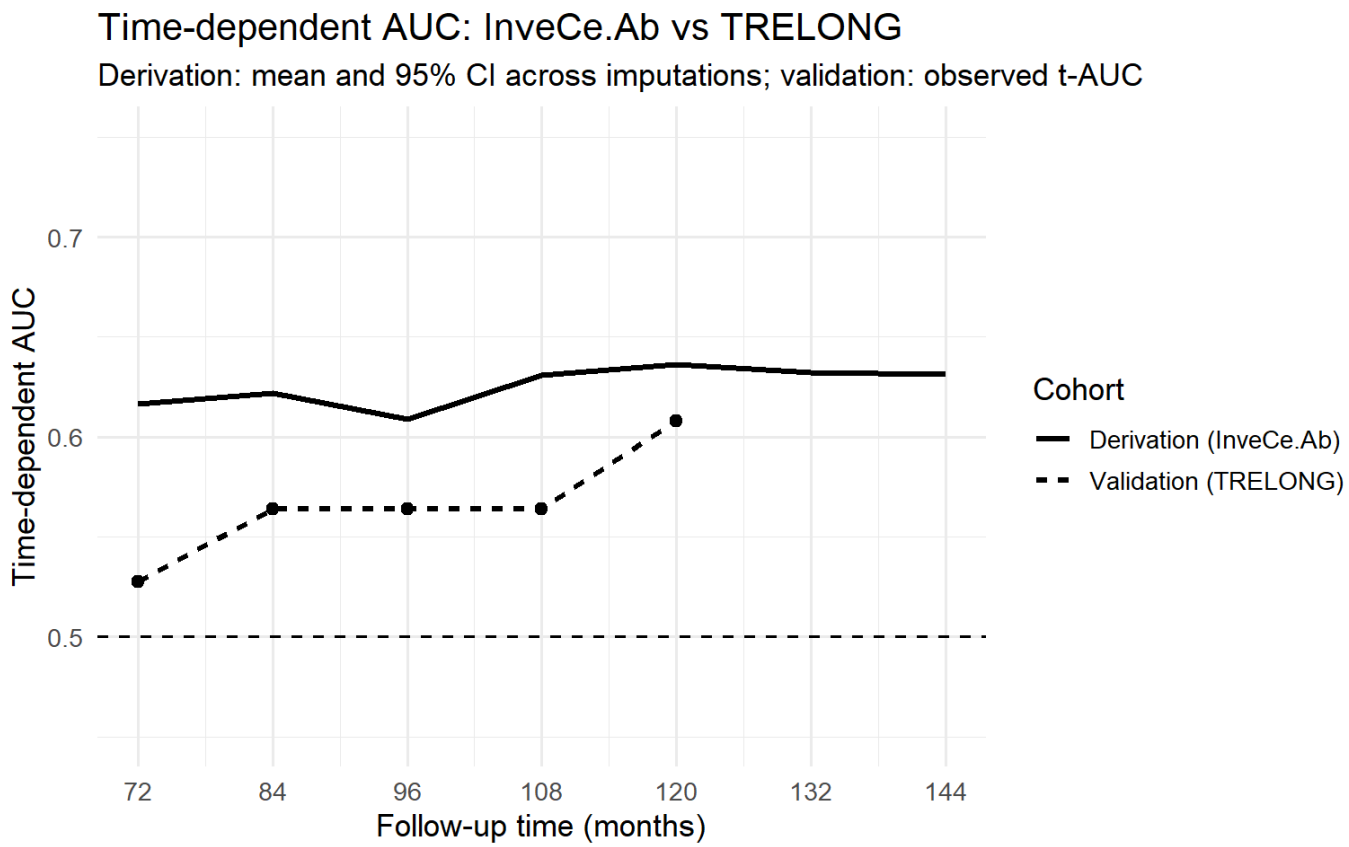


18.4 Confronto time-dependent AUC:

derivazione vs validazione

```
eval_times_val <- seq(72, 120, by = 12)

auc_val <- sapply(eval_times_val, function(t) {
  sr <- tryCatch(
    survivalROC(
      Stime = TRELOG$Time_mesi,
      status = TRELOG$evento_demenza_val,
      marker = TRELOG$RS,
      predict.time = t,
      method = "KM"
    ),
    error = function(e) NULL
  )

  if (is.null(sr) || !is.finite(sr$AUC)) return(NA_real_)
  as.numeric(sr$AUC)
})

auc_val_df <- tibble(
  time = eval_times_val,
  AUC = auc_val
)

ggplot() +
  geom_ribbon(
    data = auc_df,
    aes(x = time, ymin = L95, ymax = U95),
    alpha = 0.2
  ) +
  geom_line(
    data = auc_df,
    aes(x = time, y = AUC_mean, linetype = "Derivation (InveCe.Ab)"),
    size = 1.2
  ) +
  geom_line(
    data = auc_val_df,
    aes(x = time, y = AUC, linetype = "Validation (TRELONG)"),
    size = 1.2
  ) +
  geom_point(
    data = auc_val_df,
    aes(x = time, y = AUC),
    size = 2
  ) +
  geom_hline(yintercept = 0.5, linetype = 2) +
  labs(
```

```
    x = "Follow-up time (months)",
    y = "Time-dependent AUC",
    title = "Time-dependent AUC: InveCe.Ab vs TRELONG",
    subtitle = "Derivation: mean and 95% CI across imputations; validation: obse
rved t-AUC",
    linetype = "Cohort"
  ) +
  scale_x_continuous(breaks = seq(72, 144, by = 12), limits = c(72, 144)) +
  scale_y_continuous(limits = c(0.45, 0.75)) +
  theme_minimal(base_size = 13)
```

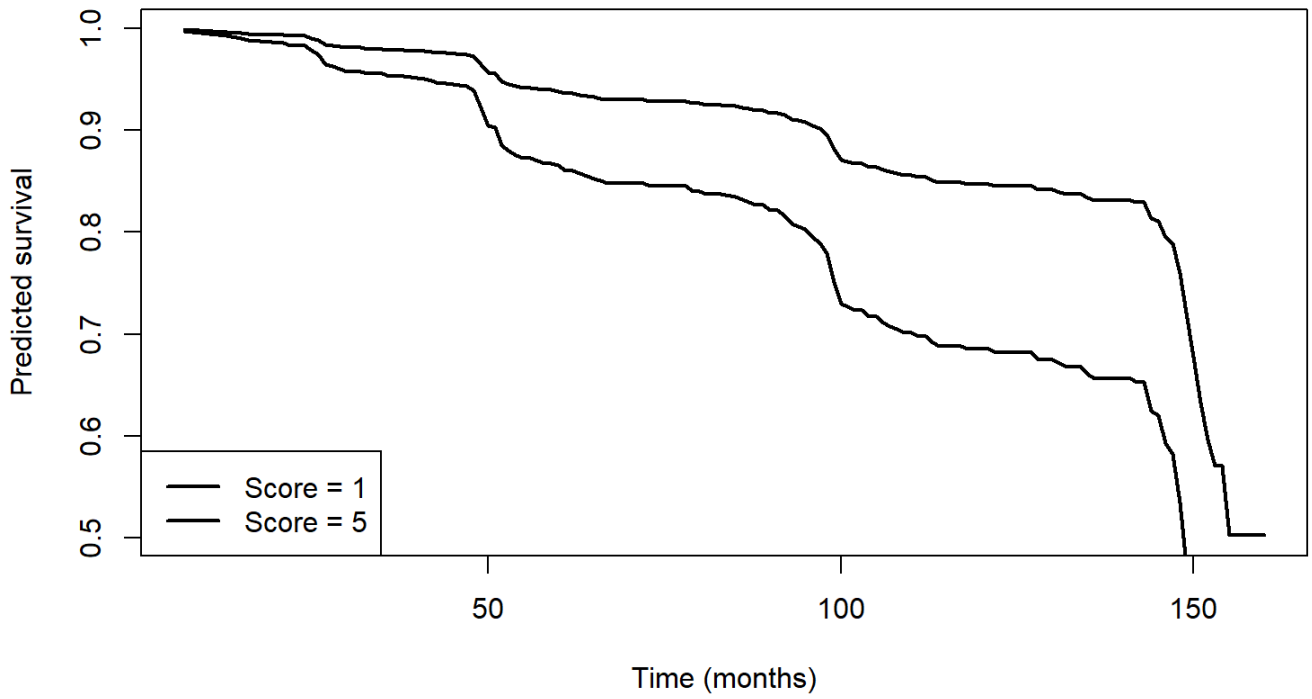



```
mean_auc_val <- mean(auc_val_df$AUC, na.rm = TRUE)
sd_auc_val <- sd(auc_val_df$AUC, na.rm = TRUE)
se_auc_val <- sd_auc_val / sqrt(sum(is.finite(auc_val_df$AUC)))

auc_summary_val <- tibble(
  Cohort = "Validation (TRELONG)",
  Mean_AUC = round(mean_auc_val, 3),
  L95 = round(mean_auc_val - 1.96 * se_auc_val, 3),
  U95 = round(mean_auc_val + 1.96 * se_auc_val, 3)
)

bind_rows(auc_summary, auc_summary_val)
```

```
## # A tibble: 2 × 4
##   Cohort                  Mean_AUC   L95   U95
##   <chr>                      <dbl> <dbl> <dbl>
## 1 Derivation (InveCe.Ab)    0.625 0.625 0.626
## 2 Validation (TRELONG)      0.565 0.541 0.59
```

19 Curve di sopravvivenza predette

```
fits_score <- lapply(1:imp$m, function(i) {
  d <- complete(imp_score, i)
  coxph(Surv(Tempo_mesi, Evento_demenza) ~ risk_score, data = d, method = "bresl
ow")
})

beta <- mean(sapply(fits_score, coef))
basefit <- survfit(fits_score[[1]])

newscore <- c(1, 5)
surv_pred <- lapply(newscore, function(s) {
  basefit$surv ^ exp(beta * s)
})

plot(
  basefit$time,
  surv_pred[[1]],
  type = "l",
  lwd = 2,
  xlab = "Time (months)",
  ylab = "Predicted survival"
)
lines(basefit$time, surv_pred[[2]], lwd = 2)
legend("bottomleft", legend = c("Score = 1", "Score = 5"), lwd = 2)
```

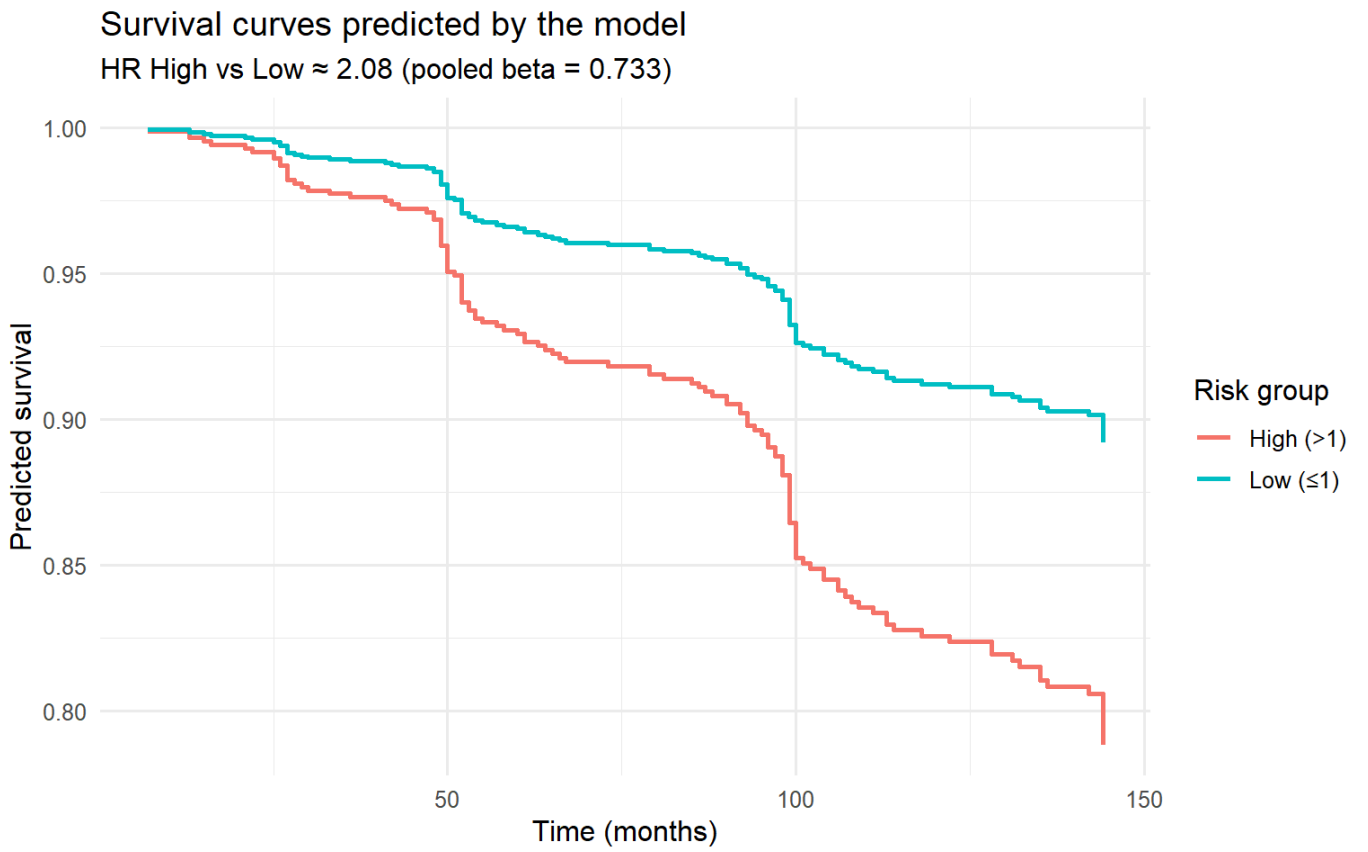

Predicted survival
1.0
0.9
0.8
0.7
0.6
0.5
Score = 1
Score = 5
50
100
150
Time (months)

19.1 Curve per gruppo di rischio

```
make_groups <- function(d) {
  d |>
    mutate(
      risk_group = factor(
        ifelse(risk_score > 1, "High", "Low"),
        levels = c("Low", "High")
      )
    )
}

fits_group <- lapply(1:imp$m, function(i) {
  d <- complete(imp_score, i) |>
    make_groups()

  coxph(
    Surv(Tempo_mesi, Evento_demenza) ~ risk_group,
    data = d,
    method = "breslow",
    x = TRUE
  )
})

pool_HR <- summary(pool(fits_group)) |>
  mutate(
    HR = exp(estimate),
    L95 = exp(estimate - 1.96 * std.error),
    U95 = exp(estimate + 1.96 * std.error)
  )

pool_HR
```

```
##             term  estimate std.error statistic       df      p.value       HR
## 1 risk_groupHigh 0.7326448 0.1627775  4.500898 169.4165 1.253722e-05 2.080576
##        L95      U95
## 1 1.512254 2.862481
```

```
beta_pool <- pool_HR$estimate[pool_HR$term == "risk_groupHigh"]
HR_pool <- exp(beta_pool)

new_low <- data.frame(risk_group = factor("Low", levels = c("Low", "High")))
new_high <- data.frame(risk_group = factor("High", levels = c("Low", "High")))

sf_low <- survfit(fits_group[[1]], newdata = new_low)
time_v <- sf_low$time
S0_t <- sf_low$surv

S_high <- S0_t ^ exp(beta_pool)
S_low <- S0_t

t_max <- 144
keep <- time_v <= t_max
time_v <- time_v[keep]
S_low <- S_low[keep]
S_high <- S_high[keep]

plot_df <- bind_rows(
  tibble(time = time_v, surv = S_low, group = "Low (≤1)"),
  tibble(time = time_v, surv = S_high, group = "High (>1)")
)

ggplot(plot_df, aes(x = time, y = surv, color = group)) +
  geom_step(size = 1) +
  labs(
    x = "Time (months)",
    y = "Predicted survival",
    color = "Risk group",
    title = "Survival curves predicted by the model",
    subtitle = paste0(
      "HR High vs Low ≈ ", sprintf("%.2f", HR_pool),
      " (pooled beta = ", sprintf("%.3f", beta_pool), ")"
    )
  ) +
  theme_minimal(base_size = 12)
```




```
interp_surv <- function(t, time, surv) {
  if (t <= min(time)) return(surv[1])
  if (t >= max(time)) return(surv[length(surv)])
  idx <- max(which(time <= t))
  surv[idx]
}

times_eval <- c(36, 60, 140)

tab_pred <- tibble(
  time = times_eval,
  S_low = sapply(times_eval, interp_surv, time = time_v, surv = S_low),
  S_high = sapply(times_eval, interp_surv, time = time_v, surv = S_high)
) |>
  mutate(
    AbsDiff = S_low - S_high,
    RelHR_approx = HR_pool
  )

tab_pred
```

```
## # A tibble: 3 × 5
##    time S_low S_high AbsDiff RelHR_approx
##   <dbl> <dbl>  <dbl>   <dbl>        <dbl>
## 1    36 0.988  0.976  0.0124         2.08
## 2    60 0.965  0.929  0.0361         2.08
## 3   140 0.903  0.808  0.0945         2.08
```

20 Note operative

Gli export Excel sono stati rimossi dal documento. Se servono, è preferibile inserirli in uno script separato oppure in chunk opzionali con `eval = FALSE`, in modo che il markdown resti centrato sulla produzione del report.

Alcuni punti dello script originale erano impliciti o duplicati. In questa versione sono stati esplicitati gli oggetti `rhs_corr` e `imp_long_mmse`, ed è stato evitato il blocco incompleto con `c_df` non definito.

21 Appendice esplorativa: sequenze, modelli multistato e dropout

Questa sezione raccoglie analisi ancora esplorative, non integrate nel flusso principale di sviluppo e validazione del risk score. L'obiettivo è documentare possibili estensioni metodologiche: analisi delle traiettorie cognitive, modelli multistato, dropout informativo e transizioni specifiche.

Per evitare che il report principale dipenda da codice ancora non stabilizzato, i chunk sono impostati con `eval = FALSE`. Quando l'analisi sarà consolidata, questa appendice potrà essere riorganizzata in una pipeline separata oppure promossa a sezione di sensitivity analysis.

21.1 Razionale dell'appendice

Le analisi qui riportate rispondono a domande diverse rispetto al modello Cox principale:

- il risk score predice il tempo alla demenza incidente?
- il risk score distingue traiettorie cognitive longitudinali diverse?
- il risk score modifica l'intensità di transizione tra stati cognitivi, morte e dropout?
- il dropout può essere trattato come stato assorbente o processo informativo?

Per questo motivo, al momento è preferibile mantenerle in appendice esplorativa e non nel corpo principale dei risultati.

21.2 Pacchetti aggiuntivi per le analisi

esplorative

```
library(tidyverse)
library(janitor)
library(msm)
library(TraMineR)
library(cluster)
library(nnet)
library(ggeffects)
library(igraph)
library(mstate)
```

21.3 Codice esplorativo originale riorganizzabile

```
# ============================================================
# BLOCCO A — Setup
# ============================================================

library(tidyverse)
library(haven)
library(janitor)
library(msm)
library(TraMineR)
library(cluster)
library(nnet)
library(broom)
library(ggplot2)
library(ggeffects)
library(purrr)

# ============================================================
# Import
# ============================================================

dat <- read_sav("W:/LAB NEUROPSICOLOGIA/Prev-Ita-Dem_PNRR/HMM/invece_risk_factor
s_pnrr_HMM.sav") %>%
  clean_names()

# ============================================================
# Costruzione dataset base
# ============================================================

# ============================================================
# BLOCCO A — Costruzione dataset long pulito
# ============================================================

build_state <- function(norm, dem, mci) {
  case_when(
    dem == 1 ~ "D",
    mci %in% c(1, 2) ~ "MCI",
    norm == 1 ~ "N",
    TRUE ~ "Missing"
  )
}

# ----------------------------
# 1. Dataset base wide
# ----------------------------
dat_base <- dat %>%
  mutate(
```

```
    id = code,
    risk_score = rs_new2
  ) %>%
  mutate(
    state_1 = build_state(cogn_norm_1, dia_demen1,  mci_1),
    state_2 = build_state(cogn_norm_2, dia_demen2,  mci_2),
    state_3 = build_state(cogn_norm_3, dia_demen3,  mci_3),
    state_4 = build_state(cogn_norm_4, dia_demen4,  mci_4),
    state_5 = build_state(cogn_norm_5, dia_demen_5, mci_5)
  )

# ----------------------------
# 2. Long per stato
# ----------------------------
long_state <- dat_base %>%
  select(id, starts_with("state_")) %>%
  pivot_longer(
    cols = starts_with("state_"),
    names_to = "wave",
    values_to = "state"
  ) %>%
  mutate(
    wave = as.integer(gsub("state_", "", wave))
  )

# ----------------------------
# 3. Long per età wave-specifica
# ATTENZIONE: prendiamo solo age1-age5, non age_1
# ----------------------------
long_age <- dat_base %>%
  select(id, age1, age2, age3, age4, age5) %>%
  pivot_longer(
    cols = age1:age5,
    names_to = "wave",
    values_to = "age_wave"
  ) %>%
  mutate(
    wave = as.integer(gsub("age", "", wave))
  )

# ----------------------------
# 4. Long per data visita wave-specifica
# ----------------------------
long_date <- dat_base %>%
  select(id, dat_socia1, dat_socia2, dat_socia3, dat_socia4, dat_socia5) %>%
  pivot_longer(
    cols = dat_socia1:dat_socia5,
    names_to = "wave",
```

```
    values_to = "date_wave"
  ) %>%
  mutate(
    wave = as.integer(gsub("dat_socia", "", wave))
  )

# ----------------------------
# 5. Variabili baseline / statiche
# ----------------------------
baseline_vars <- dat_base %>%
  select(
    id,
    age_bl = age1,
    sex, education,
    ip_colest_treatment2, alcol_yesno, social_part2cl,
    cogn_att_num_2cl, diabetes_2cl, risk_score,
    death, dateofdeath, tempo_mesi, dat_socia1
  )

# ----------------------------
# 6. Merge finale
# ----------------------------
dat_long <- long_state %>%
  left_join(long_age,  by = c("id", "wave")) %>%
  left_join(long_date, by = c("id", "wave")) %>%
  left_join(baseline_vars, by = "id")

# ----------------------------
# 7. Tempo reale e partecipazione
# ----------------------------
wave_years <- c(`1` = 0, `2` = 2, `3` = 4, `4` = 8, `5` = 12)

dat_long <- dat_long %>%
  mutate(
    time_from_baseline_years = unname(wave_years[as.character(wave)]),
    participation = if_else(state == "Missing", 0L, 1L)
  )

# controlli rapidi
cat("\n====================\nDIMENSIONI dat_long\n====================\n")
print(dim(dat_long))

cat("\n====================\nSTATI x WAVE\n====================\n")
print(table(dat_long$wave, dat_long$state, useNA = "ifany"))

cat("\n====================\nPRIME RIGHE dat_long\n====================\n")
print(
  dat_long %>%
```

```
    select(id, wave, state, age_wave, date_wave, dateofdeath, time_from_baseline
_years, participation) %>%
    head(15)
)

# ============================================================
# BLOCCO B — Inserimento Death
# ============================================================

dat_long <- dat_long %>%
  mutate(
    dat_socia1  = as.Date(dat_socia1),
    dateofdeath = as.Date(dateofdeath),
    time_death_years = as.numeric(dateofdeath - dat_socia1) / 365.25
  ) %>%
  mutate(
    state = case_when(
      !is.na(time_death_years) &
        time_from_baseline_years >= time_death_years ~ "Death",
      TRUE ~ state
    ),
    participation = if_else(state == "Missing", 0L, 1L)
  )

cat("\n====================\nSTATI x WAVE DOPO DEATH\n====================\n")
print(table(dat_long$wave, dat_long$state, useNA = "ifany"))

# ============================================================
# BLOCCO C — Sequence analysis
# ============================================================

seq_data <- dat_long %>%
  select(id, wave, state) %>%
  pivot_wider(names_from = wave, values_from = state, names_prefix = "W")

seq_obj <- seqdef(seq_data[, -1])

states_cost <- c("N", "MCI", "D", "Death", "Missing")

submat <- matrix(
  2,
  nrow = length(states_cost),
  ncol = length(states_cost),
  dimnames = list(states_cost, states_cost)
)

diag(submat) <- 0
```

```r
# Missing meno penalizzato
submat["Missing", ] <- 1
submat[, "Missing"] <- 1
submat["Missing", "Missing"] <- 0

# Death più distante
submat["Death", ] <- 3
submat[, "Death"] <- 3
submat["Death", "Death"] <- 0

cat("\n====================\nSUBSTITUTION MATRIX\n====================\n")
print(submat)

dist_om <- seqdist(
  seq_obj,
  method = "OM",
  indel = 1,
  sm = submat
)

summary(as.vector(dist_om))

clust_ward <- agnes(
  dist_om,
  diss = TRUE,
  method = "ward"
)

seq_clusters <- seq_data %>%
  mutate(
    cluster_k3 = factor(cutree(as.hclust(clust_ward), k = 3)),
    cluster_k4 = factor(cutree(as.hclust(clust_ward), k = 4))
  )

table(seq_clusters$cluster_k3)
table(seq_clusters$cluster_k4)

# ========================================================
# PLOT SEQUENZE PER CLUSTER
# ========================================================

# Distribution plot
seqdplot(
  seq_obj,
  group = seq_clusters$cluster_k3,
  with.legend = TRUE,
  main = "State distribution by cluster (k=3)"
)
```

```r
# Individual sequences
seqIplot(
  seq_obj,
  group = seq_clusters$cluster_k3,
  sortv = "from.start",
  with.legend = FALSE,
  main = "Observed sequences by cluster (k=3)"
)

# Frequent sequences
seqfplot(
  seq_obj,
  group = seq_clusters$cluster_k3,
  with.legend = TRUE,
  main = "Most frequent sequences by cluster (k=3)"
)

seq_clusters <- seq_clusters %>%
  mutate(
    cluster_label = case_when(
      cluster_k3 == "2" ~ "Stable N",
      cluster_k3 == "1" ~ "Progressive",
      cluster_k3 == "3" ~ "High mortality"
    ),
    cluster_label = relevel(factor(cluster_label), ref = "Stable N")
  )

# ============================================================
# BLOCCO D — Predizione cluster
# ============================================================

baseline <- dat_long %>%
  filter(wave == 1) %>%
  select(id, age_bl, sex, education, risk_score)

dat_cluster <- seq_clusters %>%
  left_join(baseline, by = "id")

model_cluster <- multinom(
  cluster_label ~ age_bl + sex + education + risk_score,
  data = dat_cluster
)

summary(model_cluster)

# ============================================================
# Estrazione risultati multinomiale
```

```r
# ========================================================

coef_mat <- summary(model_cluster)$coefficients
se_mat   <- summary(model_cluster)$standard.errors

# ----------------------------
# Z-score
# ----------------------------
z_mat <- coef_mat / se_mat

# ----------------------------
# P-value (two-sided)
# ----------------------------
p_mat <- 2 * (1 - pnorm(abs(z_mat)))

cat("\n====================\nP-VALUE\n====================\n")
print(round(p_mat, 4))

# ----------------------------
# Odds Ratio
# ----------------------------
OR_mat <- exp(coef_mat)

cat("\n====================\nODDS RATIO\n====================\n")
print(round(OR_mat, 3))

# ----------------------------
# Intervalli di confidenza 95%
# ----------------------------
lower <- exp(coef_mat - 1.96 * se_mat)
upper <- exp(coef_mat + 1.96 * se_mat)

cat("\n====================\nCI 95%\n====================\n")
print(round(lower, 3))
print(round(upper, 3))

results_multinom <- list(
  coef = coef_mat,
  se = se_mat,
  p = p_mat,
  OR = OR_mat,
  CI_low = lower,
  CI_high = upper
)

print(results_multinom)

pred <- ggpredict(model_cluster, terms = "risk_score")
```

```
plot(pred) +
  labs(title = "Predicted probability of cluster membership by risk score")

# ============================================================
# BLOCCO E — MSM clinico
# ============================================================

dat_clin <- dat_long %>%
  filter(state %in% c("N","MCI","D","Death")) %>%
  mutate(
    state_num = case_when(
      state == "N" ~ 1,
      state == "MCI" ~ 2,
      state == "D" ~ 3,
      state == "Death" ~ 4
      ))



Q_clin <- rbind(
  c(-0.1, 0.08, 0, 0.02),
  c(0.12, -0.30, 0.16, 0.02),
  c(0,0,-0.25,0.25),
  c(0,0,0,0)
)

rownames(Q_clin) <- colnames(Q_clin) <- c("N","MCI","D","Death")

msm_model_rs <- msm(
  state_num ~ time_from_baseline_years,
  subject = id,
  data = dat_clin,
  qmatrix = Q_clin,
  covariates = ~ risk_score,
  method = "BFGS"
)

cat("\n====================\nQ MATRIX STIMATA\n====================\n")
print(qmatrix.msm(msm_model_rs))

cat("\n====================\nP MATRIX 2 ANNI\n====================\n")
print(pmatrix.msm(msm_model_rs, t = 2))

cat("\n====================\nP MATRIX 4 ANNI\n====================\n")
print(pmatrix.msm(msm_model_rs, t = 4))

cat("\n====================\nP MATRIX 8 ANNI\n====================\n")
```

```r
print(pmatrix.msm(msm_model_rs, t = 8))

cat("\n====================\nP MATRIX 12 ANNI\n====================\n")
print(pmatrix.msm(msm_model_rs, t = 12))

cat("\n====================\nSOJOURN TIMES\n====================\n")
print(sojourn.msm(msm_model_rs))

cat("\n====================\nSTATE PREVALENCE\n====================\n")
print(pmatrix.msm(msm_model_rs, t = 12))

hazard.msm(msm_model_rs)

# ---------------------------------------------------------
# 2. Griglia temporale e matrici di transizione
# ---------------------------------------------------------
times <- seq(0, 12, by = 1)

pmats <- lapply(times, function(tt) {
  pmatrix.msm(msm_model_rs, t = tt)
})

# controllo rapido
cat("\n====================\nP MATRIX A 12 ANNI\n====================\n")
print(pmatrix.msm(msm_model_rs, t = 12))

# ---------------------------------------------------------
# 3. Grafici base di transizioni specifiche
# ---------------------------------------------------------

# N -> Death
p_N_Death <- sapply(pmats, function(P) P["N", "Death"])

plot(
  times, p_N_Death,
  type = "l", lwd = 2,
  xlab = "Years from baseline",
  ylab = "Probability",
  main = "Probability of transition N -> Death"
)

# MCI -> D
p_MCI_D <- sapply(pmats, function(P) P["MCI", "D"])

plot(
  times, p_MCI_D,
  type = "l", lwd = 2,
  xlab = "Years from baseline",
```

```r
  ylab = "Probability",
  main = "Probability of transition MCI -> D"
)

# N -> MCI
p_N_MCI <- sapply(pmats, function(P) P["N", "MCI"])

plot(
  times, p_N_MCI,
  type = "l", lwd = 2,
  xlab = "Years from baseline",
  ylab = "Probability",
  main = "Probability of transition N -> MCI"
)

# ------------------------------------------------------------
# 4. Dataset tidy per grafici completi con ggplot
# ------------------------------------------------------------

plot_df_all <- map2_dfr(pmats, times, function(P, tt) {
  expand_grid(
    from = rownames(P),
    to   = colnames(P)
  ) %>%
    mutate(
      time = tt,
      prob = map2_dbl(from, to, ~ P[.x, .y])
    )
})

# teniamo solo gli stati iniziali clinicamente rilevanti
plot_df_all <- plot_df_all %>%
  filter(from %in% c("N", "MCI", "D"))

# ------------------------------------------------------------
# 5. Grafico complessivo: tutte le probabilità di occupazione
# ------------------------------------------------------------

ggplot(plot_df_all, aes(x = time, y = prob, color = to)) +
  geom_line(linewidth = 1) +
  facet_wrap(~ from) +
  labs(
    title = "State occupation probabilities over time",
    x = "Years from baseline",
    y = "Probability",
    color = "State"
  ) +
  theme_minimal()
```

```r
# ---------------------------------------------------------
# 6. Grafico più pulito: solo esiti di interesse
# ---------------------------------------------------------

plot_df_reduced <- plot_df_all %>%
  filter(to %in% c("MCI", "D", "Death"))

ggplot(plot_df_reduced, aes(x = time, y = prob, color = to)) +
  geom_line(linewidth = 1) +
  facet_wrap(~ from, scales = "free_y") +
  labs(
    title = "Probabilities of progression and death over time",
    x = "Years from baseline",
    y = "Probability",
    color = "Destination state"
  ) +
  theme_minimal()

# ---------------------------------------------------------
# 7. Heatmap delle probabilità a 12 anni
# ---------------------------------------------------------

P12 <- pmatrix.msm(msm_model_rs, t = 12)

plot_df_heat <- as.data.frame(as.table(P12)) %>%
  rename(
    from = Var1,
    to   = Var2,
    prob = Freq
  )

ggplot(plot_df_heat, aes(x = from, y = to, fill = prob)) +
  geom_tile() +
  geom_text(aes(label = round(prob, 2))) +
  theme_minimal() +
  labs(
    title = "Transition probabilities at 12 years",
    x = "From state",
    y = "To state",
    fill = "Probability"
  )

# =========================================================
# BLOCCO PLOT RISK SCORE — effetto del risk score
# =========================================================

# ---------------------------------------------------------
```

```r
# 8. Valori del risk score da rappresentare
# continuo ma visualizzato a tre livelli descrittivi
# ----------------------------------------------------------

rs_vals <- c(
  quantile(dat_clin$risk_score, 0.25, na.rm = TRUE),
  median(dat_clin$risk_score, na.rm = TRUE),
  quantile(dat_clin$risk_score, 0.75, na.rm = TRUE)
)

rs_labels <- c("Low risk score", "Medium risk score", "High risk score")

# ----------------------------------------------------------
# 9. Probabilità nel tempo partendo da N
# ----------------------------------------------------------

plot_df_rs_N <- map2_dfr(rs_vals, rs_labels, function(rs, lab) {

  pmats_rs <- lapply(times, function(tt) {
    pmatrix.msm(
      msm_model_rs,
      t = tt,
      covariates = list(risk_score = rs)
    )
  })

  map2_dfr(pmats_rs, times, function(P, tt) {
    tibble(
      time = tt,
      risk_group = lab,
      to_state = colnames(P),
      prob = P["N", ]
    )
  })
})

ggplot(plot_df_rs_N, aes(x = time, y = prob, color = risk_group)) +
  geom_line(linewidth = 1) +
  facet_wrap(~ to_state) +
  labs(
    title = "State occupation probabilities by risk score",
    subtitle = "Initial state = N",
    x = "Years from baseline",
    y = "Probability",
    color = "Risk score"
  ) +
  theme_minimal()
```

```r
# ----------------------------------------------------------
# 10. Versione più pulita: solo MCI, D, Death partendo da N
# ----------------------------------------------------------

plot_df_rs_N_red <- plot_df_rs_N %>%
  filter(to_state %in% c("MCI", "D", "Death"))

ggplot(plot_df_rs_N_red, aes(x = time, y = prob, color = risk_group)) +
  geom_line(linewidth = 1) +
  facet_wrap(~ to_state, scales = "free_y") +
  labs(
    title = "Progression probabilities by risk score",
    subtitle = "Initial state = N",
    x = "Years from baseline",
    y = "Probability",
    color = "Risk score"
  ) +
  theme_minimal()

# ----------------------------------------------------------
# 11. Probabilità nel tempo partendo da MCI
# ----------------------------------------------------------

plot_df_rs_MCI <- map2_dfr(rs_vals, rs_labels, function(rs, lab) {

  pmats_rs <- lapply(times, function(tt) {
    pmatrix.msm(
      msm_model_rs,
      t = tt,
      covariates = list(risk_score = rs)
    )
  })

  map2_dfr(pmats_rs, times, function(P, tt) {
    tibble(
      time = tt,
      risk_group = lab,
      to_state = colnames(P),
      prob = P["MCI", ]
    )
  })
})

plot_df_rs_MCI_red <- plot_df_rs_MCI %>%
  filter(to_state %in% c("N", "D", "Death"))

ggplot(plot_df_rs_MCI_red, aes(x = time, y = prob, color = risk_group)) +
  geom_line(linewidth = 1) +
```

```
  facet_wrap(~ to_state, scales = "free_y") +
  labs(
    title = "Transition probabilities from MCI by risk score",
    subtitle = "Initial state = MCI",
    x = "Years from baseline",
    y = "Probability",
    color = "Risk score"
  ) +
  theme_minimal()

# ----------------------------------------------------------
# 12. Curva continua del risk score a tempo fisso (12 anni)
# partendo da N
# ----------------------------------------------------------

rs_seq <- seq(
  quantile(dat_clin$risk_score, 0.05, na.rm = TRUE),
  quantile(dat_clin$risk_score, 0.95, na.rm = TRUE),
  length.out = 50
)

plot_df_rs_cont <- map_dfr(rs_seq, function(rs) {

  P <- pmatrix.msm(
    msm_model_rs,
    t = 12,
    covariates = list(risk_score = rs)
  )

  tibble(
    risk_score = rs,
    to_state = colnames(P),
    prob = P["N", ]
  )
})

ggplot(plot_df_rs_cont, aes(x = risk_score, y = prob, color = to_state)) +
  geom_line(linewidth = 1) +
  labs(
    title = "12-year transition probabilities by continuous risk score",
    subtitle = "Initial state = N",
    x = "Risk score",
    y = "Probability",
    color = "State"
  ) +
  theme_minimal()

# ----------------------------------------------------------
```

```r
# 13. Versione più pulita del grafico continuo
# ----------------------------------------------------------

plot_df_rs_cont_red <- plot_df_rs_cont %>%
  filter(to_state %in% c("MCI", "D", "Death"))

ggplot(plot_df_rs_cont_red, aes(x = risk_score, y = prob, color = to_state)) +
  geom_line(linewidth = 1) +
  labs(
    title = "12-year progression probabilities by continuous risk score",
    subtitle = "Initial state = N",
    x = "Risk score",
    y = "Probability",
    color = "Destination state"
  ) +
  theme_minimal()


library(igraph)

# estrai Q numerica
Q <- qmatrix.msm(msm_model_rs, ci = "none")

# controlla
print(Q)

# archi (solo >0)
edges <- which(Q > 0, arr.ind = TRUE)

edge_list <- data.frame(
  from = rownames(Q)[edges[,1]],
  to   = colnames(Q)[edges[,2]],
  weight = Q[edges]
)

g <- graph_from_data_frame(edge_list, directed = TRUE)

plot(
  g,
  edge.arrow.size = 0.4,
  vertex.size = 30,
  vertex.label.cex = 1.2,
  main = "MSM structure (allowed transitions)"
)

E(g)$weight <- edge_list$weight

plot(
```

```
  g,
  edge.width = E(g)$weight * 10,  # scala
  edge.label = round(E(g)$weight, 3),
  vertex.size = 35,
  vertex.color = "lightblue",
  main = "Transition intensity graph (Q matrix)"
)

E(g)$weight_scaled <- log1p(E(g)$weight)

plot(
  g,
  edge.width = E(g)$weight_scaled * 5,
  edge.label = round(E(g)$weight, 3),
  vertex.size = 35,
  vertex.color = "lightblue",
  main = "MSM transition structure (log-scaled)"
)

P12 <- pmatrix.msm(msm_model_rs, t = 12)

edges_P <- which(P12 > 0.01, arr.ind = TRUE)  # soglia per pulizia

edge_list_P <- data.frame(
  from = rownames(P12)[edges_P[,1]],
  to   = colnames(P12)[edges_P[,2]],
  prob = P12[edges_P]
)

gP <- graph_from_data_frame(edge_list_P, directed = TRUE)

plot(
  gP,
  edge.width = edge_list_P$prob * 10,
  edge.label = round(edge_list_P$prob, 2),
  vertex.size = 35,
  vertex.color = "lightgreen",
  main = "12-year transition probabilities"
)

E(g)$color <- ifelse(
  edge_list$from == "N" & edge_list$to == "MCI" |
    edge_list$from == "MCI" & edge_list$to == "D",
  "red",
  "gray"
)

plot(
```

```
  g,
  layout = layout_mat,
  edge.color = E(g)$color,
  edge.width = log1p(E(g)$weight) * 5,
  vertex.size = 40,
  main = "Indirect pathway N → MCI → D"
)

# ============================================================
# BLOCCO F — Dropout informativo
# ============================================================

dat_part <- dat_long %>%
  arrange(id, time_from_baseline_years) %>%
  group_by(id) %>%
  mutate(
    state_lag = lag(state),
    participation_next = lead(participation)
  ) %>%
  ungroup() %>%
  filter(!is.na(participation_next))

model_dropout <- glm(
  participation_next ~ state_lag + time_from_baseline_years,
  data = dat_part,
  family = binomial
)

summary(model_dropout)

# ============================================================
# BLOCCO G — MSM con dropout
# ============================================================

dat_dropout <- dat_long %>%
  arrange(id, time_from_baseline_years) %>%
  group_by(id) %>%
  mutate(
    last_obs = ifelse(
      any(participation == 1),
      max(time_from_baseline_years[participation == 1]),
      NA_real_
    ),
    dropout = if_else(
      time_from_baseline_years == last_obs & state != "Death",
      1, 0
    )
  ) %>%
```

```
  ungroup()

dat_msm <- dat_msm %>%
  group_by(id) %>%
  mutate(
    exit_time = ifelse(
      any(state_ext %in% c("Death","Dropout")),
      min(time_from_baseline_years[state_ext %in% c("Death","Dropout")]),
      NA_real_
    )
  ) %>%
  filter(
    is.na(exit_time) | time_from_baseline_years <= exit_time
  ) %>%
  ungroup()

dat_msm <- dat_msm %>%
  mutate(
    state_num = case_when(
      state_ext=="N"~1,
      state_ext=="MCI"~2,
      state_ext=="D"~3,
      state_ext=="Death"~4,
      state_ext=="Dropout"~5
    )
  )

Q_ext <- rbind(
  c(-0.25,0.15,0.03,0.04,0.03),
  c(0.10,-0.30,0.12,0.05,0.03),
  c(0,0,-0.25,0.20,0.05),
  c(0,0,0,0,0),
  c(0,0,0,0,0)
)

rownames(Q_ext) <- colnames(Q_ext) <- c("N","MCI","D","Death","Dropout")

msm_dropout_model <- msm(
  state_num ~ time_from_baseline_years,
  subject = id,
  data = dat_msm,
  qmatrix = Q_ext
)

pmatrix.msm(msm_dropout_model, t=12)

pmatrix.msm(msm_model_rs, t=12)
```

```r
Qd <- qmatrix.msm(msm_dropout_model, ci = "none")
print(Qd)

edges_Qd <- which(Qd > 0, arr.ind = TRUE)

edge_list_Qd <- data.frame(
  from   = rownames(Qd)[edges_Qd[, 1]],
  to     = colnames(Qd)[edges_Qd[, 2]],
  weight = Qd[edges_Qd],
  stringsAsFactors = FALSE
)

# ordine desiderato dei nodi
vertex_order <- c("N", "MCI", "D", "Death", "Dropout")

vertices_Qd <- data.frame(
  name = vertex_order,
  stringsAsFactors = FALSE
)

g_Qd <- graph_from_data_frame(
  d = edge_list_Qd,
  vertices = vertices_Qd,
  directed = TRUE
)

# controllo
print(V(g_Qd)$name)

layout_dropout <- matrix(
  c(
    0,  0,    # N
    1,  0,    # MCI
    2,  0,    # D
    3,  0.5,  # Death
    3, -0.5   # Dropout
  ),
  ncol = 2,
  byrow = TRUE
)

plot(
  g_Qd,
  layout = layout_dropout,
  edge.arrow.size = 0.4,
  vertex.size = 35,
```

```
  vertex.label.cex = 1.1,
  main = "MSM with dropout: structural graph"
)

E(g_Qd)$weight_scaled <- log1p(E(g_Qd)$weight)

E(g_Qd)$color <- ifelse(
  edge_list_Qd$to %in% c("Death", "Dropout"),
  "firebrick",
  "steelblue"
)

plot(
  g_Qd,
  layout = layout_dropout,
  edge.width = E(g_Qd)$weight_scaled * 6,
  edge.label = round(E(g_Qd)$weight, 3),
  edge.color = E(g_Qd)$color,
  vertex.size = 38,
  vertex.label.cex = 1.1,
  main = "MSM with dropout: transition intensities"
)

P12d <- pmatrix.msm(msm_dropout_model, t = 12)

edges_Pd <- which(P12d > 0.01, arr.ind = TRUE)

edge_list_Pd <- data.frame(
  from = rownames(P12d)[edges_Pd[, 1]],
  to   = colnames(P12d)[edges_Pd[, 2]],
  prob = P12d[edges_Pd]
)

g_Pd <- graph_from_data_frame(edge_list_Pd, directed = TRUE)

E(g_Pd)$weight_scaled <- edge_list_Pd$prob * 12
E(g_Pd)$color <- ifelse(
  edge_list_Pd$to %in% c("Death", "Dropout"),
  "firebrick",
  "steelblue"
)

plot(
  g_Pd,
  layout = layout_dropout,
  edge.width = E(g_Pd)$weight_scaled,
  edge.label = round(edge_list_Pd$prob, 2),
  edge.color = E(g_Pd)$color,
```

```
  vertex.size = 38,
  vertex.label.cex = 1.1,
  main = "MSM with dropout: 12-year transition probabilities"
)

times <- seq(0, 12, by = 1)

pmats_d <- lapply(times, function(tt) {
  pmatrix.msm(msm_dropout_model, t = tt)
})

plot_df_dropout <- map2_dfr(pmats_d, times, function(P, tt) {
  expand_grid(
    from = rownames(P),
    to   = colnames(P)
  ) %>%
    mutate(
      time = tt,
      prob = map2_dbl(from, to, ~ P[.x, .y])
    )
})

plot_df_dropout <- plot_df_dropout %>%
  filter(from %in% c("N", "MCI", "D"))

ggplot(plot_df_dropout, aes(x = time, y = prob, color = to)) +
  geom_line(linewidth = 1) +
  facet_wrap(~ from) +
  labs(
    title = "State occupation probabilities over time",
    subtitle = "Model with dropout as absorbing state",
    x = "Years from baseline",
    y = "Probability",
    color = "Destination state"
  ) +
  theme_minimal()

plot_df_heat_d <- as.data.frame(as.table(P12d)) %>%
  rename(
    from = Var1,
    to   = Var2,
    prob = Freq
  )

ggplot(plot_df_heat_d, aes(x = from, y = to, fill = prob)) +
  geom_tile() +
  geom_text(aes(label = round(prob, 2))) +
  theme_minimal() +
```

```
  labs(
    title = "Transition probabilities at 12 years",
    subtitle = "Model with dropout as absorbing state",
    x = "From state",
    y = "To state",
    fill = "Probability"
  )
# ============================================================
# BLOCCO H — MSM clinico stratificato per cluster
# ============================================================

# 1. Join cluster al dataset clinico
dat_clin_cluster <- dat_clin %>%
  left_join(
    seq_clusters %>% select(id, cluster_label),
    by = "id"
  ) %>%
  mutate(
    cluster = factor(cluster_label)
  )

dat_clin_cluster <- dat_clin_cluster %>%
  mutate(
    state_lab = factor(state_num,
                       levels = 1:4,
                       labels = c("N","MCI","D","Death")),
    state_next_lab = factor(
      lead(state_num),
      levels = 1:4,
      labels = c("N","MCI","D","Death")
    )
  )

cat("\n====================\nNUMEROSITA' PER CLUSTER NEL DATASET CLINICO\n======
==============\n")
print(table(dat_clin_cluster$cluster_label, useNA = "ifany"))

# 2. Controllo rapido delle transizioni osservate per cluster
for (k in levels(dat_clin_cluster$cluster_label)) {

  cat("\n---------------------------------\n")
  cat("Cluster", k, "\n")
  cat("---------------------------------\n")

  tab_k <- dat_clin_cluster %>%
    arrange(id, time_from_baseline_years) %>%
    group_by(id) %>%
    mutate(state_next = lead(state_num)) %>%
```

```
    ungroup() %>%
    filter(cluster == k, !is.na(state_next)) %>%
    mutate(
      state_from_lab = factor(state_num,
                              levels = 1:4,
                              labels = c("N","MCI","D","Death")),
      state_to_lab = factor(state_next,
                            levels = 1:4,
                            labels = c("N","MCI","D","Death"))
    ) %>%
    count(state_from_lab, state_to_lab) %>%
    pivot_wider(names_from = state_to_lab, values_from = n, values_fill = 0) %>%
    arrange(state_from_lab)

  # ✓ stampa conteggi
  cat("\nCOUNT\n")
  print(tab_k)

  # ✓ percentuali per riga
  tab_k_pct <- tab_k %>%
    rowwise() %>%
    mutate(
      total = sum(c_across(-state_from_lab)),
      across(-state_from_lab, ~ round(. / total * 100, 1))
    ) %>%
    select(-total) %>%
    ungroup()

  cat("\nROW-WISE %\n")
  print(tab_k_pct)
}

# 3. Funzione robusta per fit MSM per cluster
fit_msm_cluster <- function(data_cluster, qmatrix_start) {

  ids_valid <- data_cluster %>%
    group_by(id) %>%
    summarise(n_obs = n(), .groups = "drop") %>%
    filter(n_obs > 1) %>%
    pull(id)

  data_cluster <- data_cluster %>%
    filter(id %in% ids_valid) %>%
    arrange(id, time_from_baseline_years)

  if (nrow(data_cluster) == 0) {
    return(list(
      ok = FALSE,
```

```
      error = "Nessuna osservazione utile dopo esclusione soggetti con 1 sola vi
sita.",
      model = NULL,
      data = data_cluster
    ))
  }

  fit <- tryCatch(
    msm(
      state_num ~ time_from_baseline_years,
      subject = id,
      data = data_cluster,
      qmatrix = qmatrix_start,
      covariates = ~ age_c + risk_score,
      method = "BFGS",
      control = list(maxit = 1000)
    ),
    error = function(e) e,
    warning = function(w) w
  )

  if (inherits(fit, "error")) {
    return(list(
      ok = FALSE,
      error = conditionMessage(fit),
      model = NULL,
      data = data_cluster
    ))
  }

  if (inherits(fit, "warning")) {
    return(list(
      ok = FALSE,
      error = conditionMessage(fit),
      model = NULL,
      data = data_cluster
    ))
  }

  list(
    ok = TRUE,
    error = NULL,
    model = fit,
    data = data_cluster
  )
}

# 4. Q matrix iniziale per il modello clinico stratificato
```

```
Q_clin_cluster <- Q_clin

# 5. Fit per ciascun cluster
cluster_fits <- list()

for (k in levels(dat_clin_cluster$cluster)) {
  cat("\n====================\nFIT MSM CLINICO - CLUSTER", k, "\n==============
=====\n")

  dat_k <- dat_clin_cluster %>%
    filter(cluster == k)

  cat("N righe:", nrow(dat_k), "\n")
  cat("N soggetti:", n_distinct(dat_k$id), "\n")

  fit_k <- fit_msm_cluster(dat_k, Q_clin_cluster)
  cluster_fits[[k]] <- fit_k

  if (fit_k$ok) {
    cat("Fit riuscito.\n")
    print(qmatrix.msm(fit_k$model))
    cat("\nHazard ratios:\n")
    print(hazard.msm(fit_k$model))
  } else {
    cat("Fit NON riuscito.\n")
    cat("Motivo:", fit_k$error, "\n")
  }
}

# 6. Riassunto finale convergenza
cat("\n====================\nRIASSUNTO FIT PER CLUSTER\n====================\n")
fit_summary <- tibble(
  cluster = names(cluster_fits),
  ok = sapply(cluster_fits, function(x) x$ok),
  message = sapply(cluster_fits, function(x) ifelse(is.null(x$error), "OK", x$er
ror))
)
print(fit_summary)

# 7. Estrazione automatica risultati dei modelli riusciti
qmat_cluster <- list()
hr_cluster <- list()
pmat12_cluster <- list()

for (k in names(cluster_fits)) {
  if (cluster_fits[[k]]$ok) {
    qmat_cluster[[k]] <- qmatrix.msm(cluster_fits[[k]]$model)
    hr_cluster[[k]] <- hazard.msm(cluster_fits[[k]]$model)
```

```
    pmat12_cluster[[k]] <- pmatrix.msm(cluster_fits[[k]]$model, t = 12)
  }
}

# 8. Stampa compatta dei modelli riusciti
for (k in names(qmat_cluster)) {
  cat("\n====================\nRISULTATI CLUSTER", k, "\n====================
\n")

  cat("\nQ MATRIX\n")
  print(qmat_cluster[[k]])

  cat("\nHAZARD RATIOS\n")
  print(hr_cluster[[k]])

  cat("\nP MATRIX A 12 ANNI\n")
  print(pmat12_cluster[[k]])
}



# ============================================================
# BLOCCO 1 — Setup
# ============================================================

library(tidyverse)
library(haven)
library(janitor)
library(survival)
library(mstate)
library(TraMineR)
library(cluster)
library(nnet)
library(broom)
library(ggplot2)
library(purrr)

# ============================================================
# BLOCCO 2 — Import
# ============================================================

dat <- read_sav("W:/LAB NEUROPSICOLOGIA/Prev-Ita-Dem_PNRR/MSM/invece_risk_factor
s_pnrr_HMM.sav") %>%
  clean_names()

# ============================================================
# BLOCCO 3 — Costruzione stato osservato
# ============================================================
```

```r
build_state <- function(norm, dem, mci) {
  case_when(
    dem == 1 ~ "D",
    mci %in% c(1, 2) ~ "MCI",   # MCI + CIND
    norm == 1 ~ "N",
    TRUE ~ "Missing"
  )
}

dat_base <- dat %>%
  mutate(
    id = code,
    risk_score = rs_new2
  ) %>%
  mutate(
    state_1 = build_state(cogn_norm_1, dia_demen1,  mci_1),
    state_2 = build_state(cogn_norm_2, dia_demen2,  mci_2),
    state_3 = build_state(cogn_norm_3, dia_demen3,  mci_3),
    state_4 = build_state(cogn_norm_4, dia_demen4,  mci_4),
    state_5 = build_state(cogn_norm_5, dia_demen_5, mci_5)
  )

long_state <- dat_base %>%
  select(id, starts_with("state_")) %>%
  pivot_longer(
    cols = starts_with("state_"),
    names_to = "wave",
    values_to = "state"
  ) %>%
  mutate(wave = as.integer(gsub("state_", "", wave)))

long_age <- dat_base %>%
  select(id, age1, age2, age3, age4, age5) %>%
  pivot_longer(
    cols = age1:age5,
    names_to = "wave",
    values_to = "age_wave"
  ) %>%
  mutate(wave = as.integer(gsub("age", "", wave)))

long_date <- dat_base %>%
  select(id, dat_socia1, dat_socia2, dat_socia3, dat_socia4, dat_socia5) %>%
  pivot_longer(
    cols = dat_socia1:dat_socia5,
    names_to = "wave",
    values_to = "date_wave"
  ) %>%
```

```
  mutate(wave = as.integer(gsub("dat_socia", "", wave)))

baseline_vars <- dat_base %>%
  select(
    id,
    age_bl = age1,
    sex, education,
    ip_colest_treatment2, alcol_yesno, social_part2cl,
    cogn_att_num_2cl, diabetes_2cl, risk_score,
    death, dateofdeath, tempo_mesi, dat_socia1
  )

dat_long <- long_state %>%
  left_join(long_age,  by = c("id", "wave")) %>%
  left_join(long_date, by = c("id", "wave")) %>%
  left_join(baseline_vars, by = "id")

wave_years <- c(`1` = 0, `2` = 2, `3` = 4, `4` = 8, `5` = 12)

dat_long <- dat_long %>%
  mutate(
    dat_socia1  = as.Date(dat_socia1),
    dateofdeath = as.Date(dateofdeath),
    time_from_baseline_years = unname(wave_years[as.character(wave)]),
    time_death_years = as.numeric(dateofdeath - dat_socia1) / 365.25
  ) %>%
  mutate(
    state = case_when(
      !is.na(time_death_years) &
        time_from_baseline_years >= time_death_years ~ "Death",
      TRUE ~ state
    ),
    participation = if_else(state == "Missing", 0L, 1L)
  )

cat("\n====================\nSTATI x WAVE\n====================\n")
print(table(dat_long$wave, dat_long$state, useNA = "ifany"))




# ============================================================
# BLOCCO 4 — Coorte analitica
# ============================================================

baseline_state <- dat_long %>%
  filter(wave == 1) %>%
  select(id, state_baseline = state)
```

```r
analytic_ids <- baseline_state %>%
  filter(state_baseline %in% c("N", "MCI")) %>%
  pull(id)

dat_analytic <- dat_long %>%
  filter(id %in% analytic_ids)

cat("\n====================\nSTATO BASELINE COORTE ANALITICA\n==================
==\n")
print(
  dat_analytic %>%
    filter(wave == 1) %>%
    count(state)
)

cat("\n====================\nALCOL BASELINE\n====================\n")
print(table(dat_analytic$alcol_yesno, useNA = "ifany"))

# opzionale: rendilo fattore leggibile
dat_analytic <- dat_analytic %>%
  mutate(
    alcol_yesno = factor(alcol_yesno)
  )

# ============================================================
# BLOCCO 5 — Intervalli osservati
# ============================================================

dat_intervals <- dat_analytic %>%
  arrange(id, time_from_baseline_years) %>%
  group_by(id) %>%
  mutate(
    next_state = lead(state),
    next_time  = lead(time_from_baseline_years),
    next_wave  = lead(wave)
  ) %>%
  ungroup() %>%
  filter(!is.na(next_state)) %>%
  transmute(
    id,
    wave_from = wave,
    wave_to   = next_wave,
    Tstart    = time_from_baseline_years,
    Tstop     = next_time,
    dt        = Tstop - Tstart,
    from      = state,
    to_obs    = next_state,
```

```
    age_bl,
    sex,
    education,
    risk_score,
    ip_colest_treatment2,
    alcol_yesno,
    social_part2cl,
    cogn_att_num_2cl,
    diabetes_2cl,
    participation,
    participation_next = lead(participation)
  )

cat("\n====================\nINTERVALLI OSSERVATI\n====================\n")
print(head(dat_intervals, 20))




# ============================================================
# BLOCCO 6 — Mappa delle transizioni cliniche consentite
# ============================================================
dat_intervals %>%
  mutate(path = case_when(
    from == "N" & to_obs == "D" ~ "Direct N→D",
    from == "N" & to_obs == "MCI" ~ "N→MCI",
    TRUE ~ "Other"
  )) %>%
  count(path)

trans_map_clin <- tribble(
  ~from,  ~to,      ~trans,
  "N",    "MCI",    "N_MCI",
  "N",    "D",      "N_D",
  "N",    "Death",  "N_Death",
  "MCI",  "N",      "MCI_N",
  "MCI",  "D",      "MCI_D",
  "MCI",  "Death",  "MCI_Death",
  "D",    "Death",  "D_Death"
)

# tieni solo intervalli clinici
dat_intervals_clin <- dat_intervals %>%
  filter(from %in% c("N", "MCI", "D"),
         to_obs %in% c("N", "MCI", "D", "Death"))

# extended data frame
dat_ext_clin <- dat_intervals_clin %>%
```

```
  left_join(trans_map_clin, by = "from") %>%
  mutate(
    status = if_else(to == to_obs, 1L, 0L),
    trans = factor(trans,
                   levels = c("N_MCI", "N_D", "N_Death", "MCI_N", "MCI_D", "MCI_
Death", "D_Death"))
  ) %>%
  arrange(id, Tstart, trans)

cat("\n====================\nEXTENDED DATA CLINICO\n====================\n")
print(head(dat_ext_clin, 20))

cat("\n====================\nEVENTI PER TRANSIZIONE\n====================\n")
print(table(dat_ext_clin$trans, dat_ext_clin$status))




# ============================================================
# BLOCCO 7 — Cox multistato clinico: effetto comune
# ============================================================

cox_clin_common <- coxph(
  Surv(Tstart, Tstop, status) ~ risk_score + strata(trans),
  data = dat_ext_clin,
  ties = "breslow"
)

summary(cox_clin_common)



# ============================================================
# BLOCCO 8 — Cox multistato clinico: effetti transition-specific
# ============================================================

dat_ext_clin <- dat_ext_clin %>%
  mutate(
    rs_N_MCI     = if_else(trans == "N_MCI",     risk_score, 0),
    rs_N_D       = if_else(trans == "N_D",       risk_score, 0),
    rs_N_Death   = if_else(trans == "N_Death",   risk_score, 0),
    rs_MCI_N     = if_else(trans == "MCI_N",     risk_score, 0),
    rs_MCI_D     = if_else(trans == "MCI_D",     risk_score, 0),
    rs_MCI_Death = if_else(trans == "MCI_Death", risk_score, 0),
    rs_D_Death   = if_else(trans == "D_Death",   risk_score, 0)
  )

cox_clin_rs <- coxph(
```

```
  Surv(Tstart, Tstop, status) ~
    rs_N_MCI + rs_N_D + rs_N_Death +
    rs_MCI_N + rs_MCI_D + rs_MCI_Death +
    rs_D_Death +
    strata(trans),
  data = dat_ext_clin,
  ties = "breslow"
)

summary(cox_clin_rs)

exp(cbind(HR = coef(cox_clin_rs), confint(cox_clin_rs)))

dat_ext_clin <- dat_ext_clin %>%
  mutate(
    alcol_bin = if_else(alcol_yesno == 1, 1, 0),
    alc_N_MCI     = if_else(trans == "N_MCI",     alcol_bin, 0),
    alc_N_D       = if_else(trans == "N_D",       alcol_bin, 0),
    alc_N_Death   = if_else(trans == "N_Death",   alcol_bin, 0),
    alc_MCI_N     = if_else(trans == "MCI_N",     alcol_bin, 0),
    alc_MCI_D     = if_else(trans == "MCI_D",     alcol_bin, 0),
    alc_MCI_Death = if_else(trans == "MCI_Death", alcol_bin, 0),
    alc_D_Death   = if_else(trans == "D_Death",   alcol_bin, 0)
  )

cox_clin_alcol <- coxph(
  Surv(Tstart, Tstop, status) ~
    alc_N_MCI + alc_N_D + alc_N_Death +
    alc_MCI_N + alc_MCI_D + alc_MCI_Death +
    alc_D_Death +
    strata(trans),
  data = dat_ext_clin,
  ties = "breslow"
)

summary(cox_clin_alcol)

exp(cbind(HR = coef(cox_clin_alcol), confint(cox_clin_alcol)))
# ============================================================
# BLOCCO 9 — Esempio con covariata tempo-dipendente
# ============================================================

# da adattare solo se hai la versione wave-specifica
# qui uso social_part2cl come placeholder: se è davvero baseline, non è TD

cox_clin_social <- coxph(
  Surv(Tstart, Tstop, status) ~ social_part2cl + strata(trans),
  data = dat_ext_clin,
```

```
  ties = "breslow"
)

summary(cox_clin_social)



# ============================================================
# BLOCCO 10 — Definizione dropout definitivo
# ============================================================

dat_dropout_flag <- dat_analytic %>%
  arrange(id, time_from_baseline_years) %>%
  group_by(id) %>%
  mutate(
    last_obs_time = if_else(
      any(participation == 1),
      max(time_from_baseline_years[participation == 1]),
      NA_real_
    ),
    dropout = if_else(
      time_from_baseline_years == last_obs_time & state != "Death",
      1L, 0L
    )
  ) %>%
  ungroup()

# stato osservato con dropout
dat_dropout_flag <- dat_dropout_flag %>%
  mutate(
    state_obs = case_when(
      state == "Death" ~ "Death",
      dropout == 1 ~ "Dropout",
      TRUE ~ state
    )
  )



# ============================================================
# BLOCCO 11 — Intervalli con dropout competitivo
# ============================================================

dat_intervals_drop <- dat_dropout_flag %>%
  arrange(id, time_from_baseline_years) %>%
  group_by(id) %>%
  mutate(
    next_state = lead(state_obs),
```

```
    next_time  = lead(time_from_baseline_years),
    next_wave  = lead(wave)
  ) %>%
  ungroup() %>%
  filter(!is.na(next_state)) %>%
  transmute(
    id,
    wave_from = wave,
    wave_to   = next_wave,
    Tstart    = time_from_baseline_years,
    Tstop     = next_time,
    dt        = Tstop - Tstart,
    from      = state_obs,
    to_obs    = next_state,
    risk_score
  )



# ============================================================
# BLOCCO 12 — Mappa transizioni con dropout
# ============================================================

trans_map_drop <- tribble(
  ~from,  ~to,        ~trans,
  "N",    "MCI",      "N_MCI",
  "N",    "D",        "N_D",
  "N",    "Death",    "N_Death",
  "N",    "Dropout",  "N_Dropout",
  "MCI",  "N",        "MCI_N",
  "MCI",  "D",        "MCI_D",
  "MCI",  "Death",    "MCI_Death",
  "MCI",  "Dropout",  "MCI_Dropout",
  "D",    "Death",    "D_Death",
  "D",    "Dropout",  "D_Dropout"
)

dat_ext_drop <- dat_intervals_drop %>%
  filter(from %in% c("N", "MCI", "D")) %>%
  left_join(trans_map_drop, by = "from") %>%
  mutate(
    status = if_else(to == to_obs, 1L, 0L),
    trans = factor(trans)
  ) %>%
  arrange(id, Tstart, trans)

cat("\n====================\nEVENTI PER TRANSIZIONE CON DROPOUT\n==============
=====\n")
```

```r
print(table(dat_ext_drop$trans, dat_ext_drop$status))



# ============================================================
# BLOCCO 13 — Cox multistato con dropout competitivo
# ============================================================

cox_drop_common <- coxph(
  Surv(Tstart, Tstop, status) ~ risk_score + strata(trans),
  data = dat_ext_drop,
  ties = "breslow"
)

summary(cox_drop_common)



# ============================================================
# BLOCCO 14 — Modello con dropout
# ============================================================


dat_ext_drop <- dat_ext_drop %>%
  mutate(
    rs_N_MCI     = if_else(trans == "N_MCI",     risk_score, 0),
    rs_N_D       = if_else(trans == "N_D",       risk_score, 0),
    rs_N_Death   = if_else(trans == "N_Death",   risk_score, 0),
    rs_N_Dropout   = if_else(trans == "N_Dropout",   risk_score, 0),
    rs_MCI_N     = if_else(trans == "MCI_N",     risk_score, 0),
    rs_MCI_D     = if_else(trans == "MCI_D",     risk_score, 0),
    rs_MCI_Death = if_else(trans == "MCI_Death", risk_score, 0),
    rs_MCI_Dropout = if_else(trans == "MCI_Dropout", risk_score, 0),
    rs_D_Death   = if_else(trans == "D_Death",   risk_score, 0),
    rs_D_Dropout   = if_else(trans == "D_Dropout",   risk_score, 0)
  )

cox_drop_pars <- coxph(
  Surv(Tstart, Tstop, status) ~ rs_N_MCI + rs_N_D + rs_N_Death + rs_N_Dropout +
    rs_MCI_N + rs_MCI_D + rs_MCI_Death + rs_MCI_Dropout +
    rs_D_Death + rs_D_Dropout + strata(trans),
  data = dat_ext_drop,
  ties = "breslow"
)

summary(cox_drop_pars)
exp(cbind(HR = coef(cox_drop_pars), confint(cox_drop_pars)))
```

```
dat_ext_drop <- dat_ext_drop %>%
  mutate(
    alcol_bin = if_else(alcol_yesno == 1, 1, 0),
    alc_N_MCI       = if_else(trans == "N_MCI",       alcol_bin, 0),
    alc_N_D         = if_else(trans == "N_D",         alcol_bin, 0),
    alc_N_Death     = if_else(trans == "N_Death",     alcol_bin, 0),
    alc_N_Dropout   = if_else(trans == "N_Dropout",   alcol_bin, 0),
    alc_MCI_N       = if_else(trans == "MCI_N",       alcol_bin, 0),
    alc_MCI_D       = if_else(trans == "MCI_D",       alcol_bin, 0),
    alc_MCI_Death   = if_else(trans == "MCI_Death",   alcol_bin, 0),
    alc_MCI_Dropout = if_else(trans == "MCI_Dropout", alcol_bin, 0),
    alc_D_Death     = if_else(trans == "D_Death",     alcol_bin, 0),
    alc_D_Dropout   = if_else(trans == "D_Dropout",   alcol_bin, 0)
  )

cox_drop_alcol <- coxph(
  Surv(Tstart, Tstop, status) ~
    alc_N_MCI + alc_N_D + alc_N_Death + alc_N_Dropout +
    alc_MCI_N + alc_MCI_D + alc_MCI_Death + alc_MCI_Dropout +
    alc_D_Death + alc_D_Dropout +
    strata(trans),
  data = dat_ext_drop,
  ties = "breslow"
)

summary(cox_drop_alcol)

exp(cbind(HR = coef(cox_drop_alcol), confint(cox_drop_alcol)))
# ============================================================
# BLOCCO 15 — Forest plot HR
# ============================================================

hr_df <- broom::tidy(cox_clin_rs, exponentiate = TRUE, conf.int = TRUE)

ggplot(hr_df, aes(x = estimate, y = term)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0.2) +
  geom_vline(xintercept = 1, linetype = 2) +
  scale_x_log10() +
  theme_minimal() +
  labs(
    title = "Transition-specific hazard ratios for risk score",
    x = "Hazard ratio (log scale)",
    y = ""
  )
```

```
hr_drop_df <- broom::tidy(cox_drop_pars, exponentiate = TRUE, conf.int = TRUE)

ggplot(hr_drop_df, aes(x = estimate, y = term)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0.2) +
  geom_vline(xintercept = 1, linetype = 2) +
  scale_x_log10() +
  theme_minimal() +
  labs(
    title = "model with dropout: hazard ratios",
    x = "Hazard ratio (log scale)",
    y = ""
  )
```