

HRP 392_GLP-1_MASLD Progression_Model

2025-12-05

```
#####
## Markov CEA: MASLD - LSM vs Semaglutide
## Horizon: 80 Years, Monthly Cycle
## Base case: GLP-1s for Lifetime
## Alternative durations: 72 weeks (~1.38 y), 5y, 20y
#####

rm(list = ls())

## Libraries
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
## filter, lag

## The following objects are masked from 'package:base':
## intersect, setdiff, setequal, union

library(tidyr)
library(ggplot2)
library(ggrepel)
library(scales)

## Colorblind-safe palette (Okabe-Ito)
okabe_ito_strat <- c(
  LSM      = "#0072B2",
  Semaglutide = "#009E73"
)

#####
##### Helper Functions #####
#####

prob_to_rate <- function(p, t = 1) -(1/t) * log(1 - p)
rate_to_prob <- function(r, t = 1) 1 - exp(-r * t)
row_normalize <- function(m) sweep(m, 1, rowSums(m), FUN = "/")
clip01       <- function(x) pmin(pmax(x, 0), 0.999)
```

```

#####
##### Load Mortality + Background Costs (CSV) #####
#####

mort_cost_df <- read.csv("~/Downloads/age_mort_background_costs.csv")

#####
##### Model Specs #####
#####

cycle_length <- 1/12           # Monthly
age_start      <- 12
time_horizon   <- 80          # 80 years of cycles
age_end        <- age_start + time_horizon
n_cycles       <- time_horizon / cycle_length    # 960 cycles

#####
##### Health States + Initial #####
#####

v_states <- c(
  "F0", "F1", "F2", "F3", "F4_CC", "DCC",
  "HCC", "LT_Y1", "LT_Y1_P", "Post_LT", "Dead"
)
n_states <- length(v_states)

v_init <- c(
  F0=0.628, F1=0.309, F2=0.035, F3=0.016, F4_CC=0.012,
  DCC=0, HCC=0, LT_Y1_P=0, LT_Y1=0, Post_LT=0, Dead=0
)

#####
##### Costs #####
#####

# Annual state costs
costs_base <- c(
  F0=8698, F1=8698, F2=8698, F3=10372, F4_CC=42207,
  DCC=195156, HCC=141615,
  LT_Y1_P=452682, LT_Y1=452682,
  Post_LT=49851, Dead=0
)

# Low/high
costs_low <- c(
  F0=6958, F1=6958, F2=6958, F3=8297, F4_CC=33766,
  DCC=156125, HCC=113292,
  LT_Y1_P=279052, LT_Y1=279052,
  Post_LT=round(49851*0.75), Dead=0
)

```

```

costs_high <- c(
  F0=10436, F1=10436, F2=10436, F3=12447, F4_CC=50650,
  DCC=234187, HCC=169939,
  LT_Y1_P=626315, LT_Y1=626315,
  Post_LT=round(49851*1.25), Dead=0
)

### LT Complications
lt_comp_base <- c(
  acr=28950, biliary_comp=54943, HAT=112834, skin_infection=3915,
  pneumonia=80291, bloodstream_inf=102690, peritonitis=119762,
  uti=68730, cdiff=46091, other_infection=68063,
  VTE=53165, reoperation=111674, primary_nonfxn=107031,
  HVS=73838, renal_failure=82524
)
lt_additional_cost_base <- sum(lt_comp_base)

# Add LT complications to LT states
for (st in c("LT_Y1","LT_Y1_P")) {
  costs_base[st] <- costs_base[st] + lt_additional_cost_base
}

### Drug costs (annual)
cost_lsm      <- 0
cost_sema_high <- 6829
cost_sema_low  <- 350 * 12    # "TrumpRx" low-cost scenario (used only in SA)

drug_cost <- c(
  LSM        = cost_lsm,
  Semaglutide = cost_sema_high
)

#####
##### Discounting + Half-cycle #####
#####

disc_cost <- 0.03
disc_qaly <- 0.03

v_dwc <- 1/(1+disc_cost)^((0:n_cycles)*cycle_length)
v_dwu <- 1/(1+disc_qaly)^((0:n_cycles)*cycle_length)
v_wcc <- c(0.5, rep(1, n_cycles-1), 0.5)

#####
##### Age-specific Mortality + Background Costs #####
#####

ages_states <- seq(age_start, age_end, by=cycle_length)
ages_cycles <- ages_states[-length(ages_states)]

v_p_bg_annual <- approx(mort_cost_df$Age,

```

```

        mort_cost_df$overall_mortality_avg,
        ages_cycles, rule=2)$y

v_r_bg_annual <- prob_to_rate(v_p_bg_annual)
v_p_bg_month <- rate_to_prob(v_r_bg_annual, cycle_length)

v_background_cost_annual <-
  approx(mort_cost_df$Age,
         mort_cost_df$background_cost_2025,
         ages_states, rule=2)$y

v_background_cost_month <- v_background_cost_annual / 12

#####
##### Utilities #####
#####

age_vec <- c(12, 25, 35, 45, 55, 65, 75)

util_age_base <- c(0.919, 0.911, 0.841, 0.816, 0.815, 0.824, 0.811)
qaly_dec_base <- c(
  Healthy=0, F0=0.016, F1=0.016, F2=0.016, F3=0.145, F4_CC=0.145,
  HCC=0.165, DCC=0.155, LT_Y1=0.286, LT_Y1_P=0.286,
  Post_LT=0.036, Dead=1
)
v_util_age_base <- approx(age_vec, util_age_base, ages_states, rule=2)$y

state_order_for_util <- c(
  "F0", "F1", "F2", "F3", "F4_CC", "HCC",
  "DCC", "LT_Y1", "LT_Y1_P", "Post_LT", "Dead"
)

build_util_matrix <- function(v_util_age, qdec){
  m <- sapply(state_order_for_util, function(st) {
    if (st=="Dead") rep(0,length(v_util_age))
    else pmax(v_util_age - qdec[st], 0)
  })
  colnames(m) <- state_order_for_util
  m
}

m_util_base <- build_util_matrix(v_util_age_base, qaly_dec_base)

#####
##### TRANSITION PROBABILITIES #####
#####

# Observational ANNUAL fibrosis progression/regression
incidence_obs <- c(
  F0_F1 = 0.0650,

```

```

F1_F0 = 0.0250,
F1_F2 = 0.0690,
F2_F1 = 0.0460,
F2_F3 = 0.1000,
F3_F2 = 0.0880,
F3_F4 = 0.0540,
F4_F3 = 0.0680
)

# Annual transition list
p_prog <- list(
  F0_F1 = incidence_obs["F0_F1"],
  F1_F0 = incidence_obs["F1_F0"],
  F1_F2 = incidence_obs["F1_F2"],
  F2_F1 = incidence_obs["F2_F1"],
  F2_F3 = incidence_obs["F2_F3"],
  F3_F2 = incidence_obs["F3_F2"],
  F3_F4 = incidence_obs["F3_F4"],
  F4_F3 = incidence_obs["F4_F3"],

  F3_HCC = 0.0011,
  F4_HCC = 0.0022,
  F4_DCC = 0.0268,

  DCC_HCC = 0.0011,
  DCC_LT = 0.0062,
  DCC_Death = 0.0734,

  HCC_LT = 0.0062,
  HCC_Death = 0.00117,

  F4_LT = 0.0062,

  DCC_RegressF4 = 0,
  HCC_RegressF4 = 0,

  LT1_to_PostLT = 0.90,
  LT1P_to_PostLT = 0.90
)

# Convert ANNUAL + MONTHLY probabilities
p_prog_month <- lapply(p_prog, function(p_annual) {
  rate_to_prob(prob_to_rate(p_annual), cycle_length)
})

#####
##### TREATMENT EFFECTS #####
#####

treatments <- c("LSM", "Semaglutide")

## Semaglutide regression RR = 1.56

```

```

rr_sema_regress <- 1.56

rr_regress <- c(
  LSM        = 1.0,
  Semaglutide = rr_sema_regress
)

# progression RR = 1 / regression RR
rr_progress <- c(
  LSM        = 1.0,
  Semaglutide = 1/rr_sema_regress
)

#####
##### LT + MORTALITY #####
#####

hr_excess <- c(
  F0=1.3, F1=2.05, F2=2.0, F3=4.3, F4_CC=8.32,
  DCC=6.0, HCC=20, LT_Y1_P=8, LT_Y1=8,
  Post_LT=3, Dead=1
)

a_p_death <- array(0, dim = c(n_states, n_cycles),
                    dimnames = list(v_states, NULL))

for (s in v_states) {
  a_p_death[s,] <- pmin(v_p_bg_month * hr_excess[s], 0.99)
}
a_p_death["Dead",] <- 1

#####
##### CYCLE TRANSITION #####
#####

build_cycle_P <- function(rr_regress_now, rr_progress_now, age_t,
                           p_prog_month_local = p_prog_month) {

  P <- matrix(0, n_states, n_states, dimnames=list(v_states,v_states))

  ### Fibrosis progression
  P["F0","F1"]    <- clip01(p_prog_month_local$F0_F1 * rr_progress_now)
  P["F1","F2"]    <- clip01(p_prog_month_local$F1_F2 * rr_progress_now)
  P["F2","F3"]    <- clip01(p_prog_month_local$F2_F3 * rr_progress_now)
  P["F3","F4_CC"] <- clip01(p_prog_month_local$F3_F4 * rr_progress_now)

  ### Fibrosis regression
  P["F1","F0"]    <- clip01(p_prog_month_local$F1_F0 * rr_regress_now)
  P["F2","F1"]    <- clip01(p_prog_month_local$F2_F1 * rr_regress_now)
  P["F3","F2"]    <- clip01(p_prog_month_local$F3_F2 * rr_regress_now)
  P["F4_CC","F3"] <- clip01(p_prog_month_local$F4_F3 * rr_regress_now)
}

```

```

#### Advanced liver routes
P["F3","HCC"] <- clip01(p_prog_month_local$F3_HCC)
P["F4_CC","HCC"] <- clip01(p_prog_month_local$F4_HCC)
P["F4_CC","DCC"] <- clip01(p_prog_month_local$F4_DCC)

P["DCC","HCC"] <- clip01(p_prog_month_local$DCC_HCC)

#### Regression from advanced states (kept as 0 unless changed)
P["DCC","F4_CC"] <- clip01(p_prog_month_local$DCC_RegressF4 * rr_regress_now)
P["HCC","F4_CC"] <- clip01(p_prog_month_local$HCC_RegressF4 * rr_regress_now)

#### Liver transplant
if(age_t < 18){
  P["F4_CC","LT_Y1_P"] <- clip01(p_prog_month_local$F4_LT)
  P["DCC","LT_Y1_P"] <- clip01(p_prog_month_local$DCC_LT)
  P["HCC","LT_Y1_P"] <- clip01(p_prog_month_local$HCC_LT)
  P["LT_Y1_P","Post_LT"] <- clip01(p_prog_month_local$LT1P_to_PostLT)
} else {
  P["F4_CC","LT_Y1"] <- clip01(p_prog_month_local$F4_LT)
  P["DCC","LT_Y1"] <- clip01(p_prog_month_local$DCC_LT)
  P["HCC","LT_Y1"] <- clip01(p_prog_month_local$HCC_LT)
  P["LT_Y1","Post_LT"] <- clip01(p_prog_month_local$LT1_to_PostLT)
}

return(P)
}

#####
##### BUILD 4D TRANSITION ARRAY #####
#####

build_a_P <- function(rr_reg, rr_prog,
                      p_prog_month_local = p_prog_month,
                      treat_dur_cycles = NULL) {

  if(is.null(treat_dur_cycles)){
    treat_dur_cycles <- base_treat_dur_cycles
  }

  a_P <- array(0,
               dim=c(n_states,n_states,n_cycles,length(treatments)),
               dimnames=list(v_states,v_states,1:n_cycles,treatments))

  for(stg in treatments){

    dur_stg <- treat_dur_cycles[stg]

    for(t in 1:n_cycles){

      age_t <- age_start + (t-1)*cycle_length

      # Apply treatment RR while on treatment
    }
  }
}

```

```

    if(t <= dur_stg){
      rrreg <- rr_reg[stg]
      rrprog<- rr_prog[stg]
    } else {
      rrreg <- 1.0
      rrprog<- 1.0
    }

  P <- build_cycle_P(rrreg, rrprog, age_t, p_prog_month_local)

  #### Add mortality
  P[, "Dead"] <- a_p_death[,t]
  P["Dead",] <- 0
  P["Dead", "Dead"] <- 1

  #### Pediatric → adult LT at 18
  if(age_t >= 18){
    P["LT_Y1_P",] <- 0
    P["LT_Y1_P", "Dead"] <- a_p_death["LT_Y1_P",t]
    P["LT_Y1_P", "LT_Y1"] <- 1 - a_p_death["LT_Y1_P",t]
  }

  #### Stay probabilities
  for(s in v_states){
    if(s != "Dead"){
      stay <- 1 - sum(P[s,])
      P[s,s] <- max(stay, 0)
    }
  }

  P <- row_normalize(P)
  a_P[,,t,stg] <- P
}

return(a_P)
}

#####
##### Markov Model #####
#####

run_markov <- function(P4d_single, v_init){

  m_M <- matrix(NA, nrow=n_cycles+1, ncol=n_states,
                dimnames=list(0:n_cycles, v_states))

  m_M[1,] <- v_init[v_states]

  for(t in 1:n_cycles){
    m_M[t+1,] <- m_M[t,] %*% P4d_single[,,t]
  }
}

```

```

    return(m_M)
}

#####
##### SCENARIOS #####
#####

# Treatment duration vectors in years
treat_dur_scenarios_years <- list(
  `72w` = c(LSM=0,
            Semaglutide = 72/52),
  `5y` = c(LSM=0,
            Semaglutide = 5),
  `20y` = c(LSM=0,
            Semaglutide = 20),
  Lifetime = c(LSM=0,
            Semaglutide = time_horizon)
)

# Convert to cycles
treat_dur_scenarios_cycles <- lapply(treat_dur_scenarios_years, function(v){
  out <- round(v / cycle_length)
  out[out > n_cycles] <- n_cycles
  names(out) <- names(v)
  out
})

# Base Case = Lifetime
base_treat_dur_cycles <- treat_dur_scenarios_cycles[["Lifetime"]]

#####
##### OUTCOMES #####
#####

summarize_outcomes <- function(trace, drug_cost_per_year,
                                  util_matrix, bg_cost_month,
                                  cost_vector,
                                  treat_dur_cycles) {

  util_mat <- util_matrix[, v_states]

  # State costs
  v_cost_state_annual <- trace %*% matrix(cost_vector[v_states], ncol=1)
  v_cost_state_cycle <- v_cost_state_annual * cycle_length

  # Drug cost while on treatment
  on_vec <- rep(0, n_cycles+1)
  if(treat_dur_cycles > 0){
    idx_end <- min(treat_dur_cycles+1, n_cycles+1)
    on_vec[1:idx_end] <- 1
  }
}

```

```

v_cost_drug_cycle <- drug_cost_per_year * cycle_length * on_vec

# Background cost
alive <- 1 - trace[, "Dead"]
v_bg_cost_cycle <- bg_cost_month * alive

# Total cost
v_cost_total <- v_cost_state_cycle + v_cost_drug_cycle + v_bg_cost_cycle

# QALYs
v_qaly_state <- rowSums(trace * util_mat)

# Life-years (1 - Dead)
LY <- rowSums(trace[, v_states != "Dead"])

# Discount + half-cycle
tot_LY <- sum(LY * v_wcc * v_dwu) * cycle_length
tot_QALY <- sum(v_qaly_state * v_wcc * v_dwu) * cycle_length
tot_cost <- sum(v_cost_total * v_wcc * v_dwc)

c(LY=tot_LY, QALY=tot_QALY, Cost=tot_cost)
}

#####
##### SUMMARIZE STRATEGIES #####
#####

summarize_strategies <- function(traces_list, scenario_name,
                                    util_matrix, bg_cost_month,
                                    cost_vector, drug_cost_vec,
                                    treat_dur_cycles_vec) {

  res_mat <- sapply(treatments, function(stg){

    summarize_outcomes(
      trace = traces_list[[stg]],
      drug_cost_per_year = drug_cost_vec[stg],
      util_matrix = util_matrix,
      bg_cost_month = bg_cost_month,
      cost_vector = cost_vector,
      treat_dur_cycles = treat_dur_cycles_vec[stg]
    )
  })

  df <- as.data.frame(t(res_mat))
  df$Strategy <- rownames(df)
  df$Scenario <- scenario_name
  df[, c("Scenario", "Strategy", "Cost", "QALY", "LY")]
})

#####

```

```

##### RUN BASE CASE (LIFETIME) #####
#####

a_P <- build_a_P(rr_regress, rr_progress,
                    p_prog_month_local = p_prog_month,
                    treat_dur_cycles   = base_treat_dur_cycles)

traces <- lapply(treatments, function(stg){
  run_markov(a_P[,,stg], v_init)
})
names(traces) <- treatments

res_base <- summarize_strategies(
  traces, "Base (Lifetime)",
  m_util_base,
  v_background_cost_month,
  costs_base,
  drug_cost,
  treat_dur_cycles_vec = base_treat_dur_cycles
)

#####

##### CEA FUNCTIONS #####
#####

calculate_cea <- function(res_df){

  res_df %>%
    arrange(Cost) %>%
    mutate(
      dCost = Cost - lag(Cost),
      dQALY = QALY - lag(QALY),
      ICER = dCost / dQALY
    ) %>%
    filter(is.na(lag(dQALY)) | dQALY > 0)
}

cea_base <- calculate_cea(res_base)

#####

##### CEA TABLE #####
#####

calculate_cea_pub <- function(res_df){

  df <- res_df %>%
    arrange(Cost) %>%
    mutate(
      dCost = Cost - lag(Cost),
      dQALY = QALY - lag(QALY),
      ICER = dCost / dQALY
    )
}

```

```

)
df %>%
  mutate(
    QALY = round(QALY, 4),
    LY = round(LY, 4),
    Cost = round(Cost, 2),
    dCost = ifelse(is.na(dCost), NA, round(dCost,2)),
    dQALY = ifelse(is.na(dQALY), NA, round(dQALY,4)),
    ICER = ifelse(is.na(ICER), NA, round(ICER,0))
  ) %>%
  select(Strategy, QALY, LY, Cost, dCost, dQALY, ICER)
}

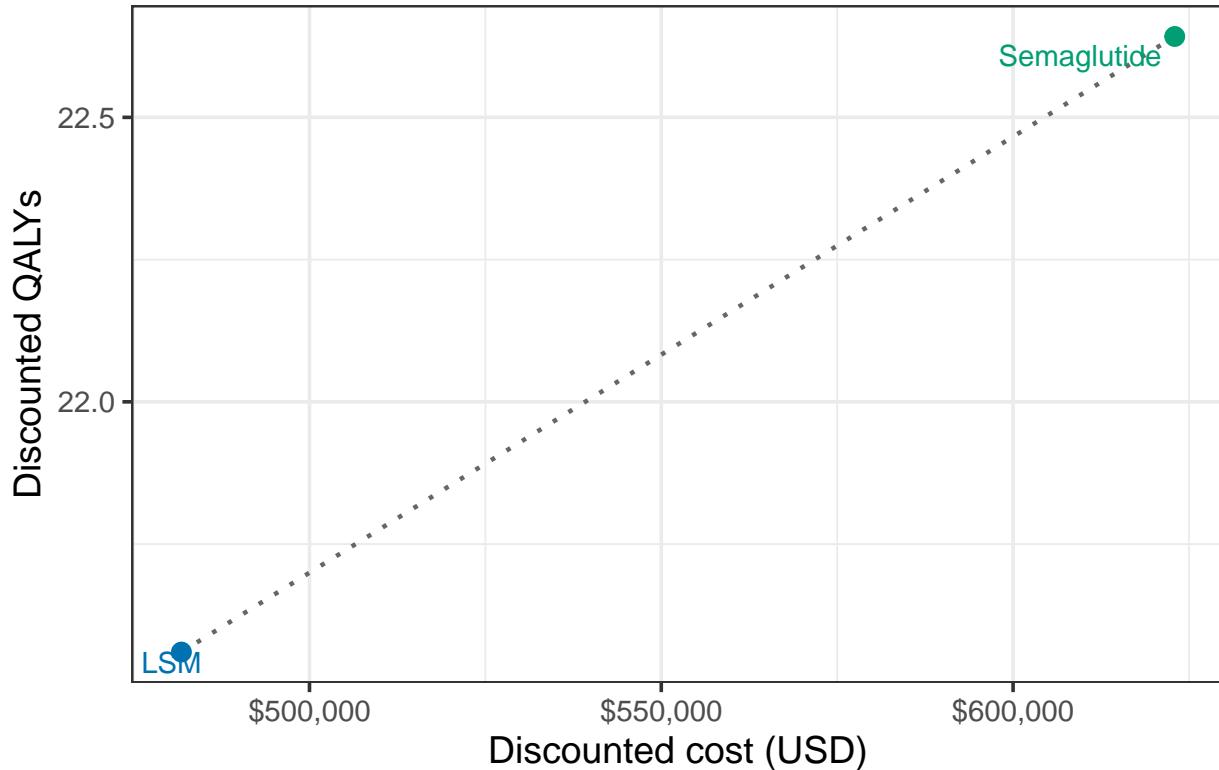
#####
##### CEA PLANE (Lifetime) #####
#####

res_base_ord <- res_base %>% arrange(Cost)

ggplot(res_base_ord,
       aes(x=Cost, y=QALY, color=Strategy)) +
  geom_line(aes(group=1), linetype="dotted",
            linewidth=0.8, color="grey40") +
  geom_point(size=3) +
  geom_text_repel(aes(label=Strategy), size=4) +
  scale_color_manual(values=okabe_ito_strat) +
  scale_x_continuous(
    labels = label_dollar(scale=1),
    breaks = pretty_breaks(n = 3)
  ) +
  scale_y_continuous(
    breaks = pretty_breaks(n = 3)
  ) +
  labs(x="Discounted cost (USD)",
       y="Discounted QALYs",
       title="Cost-Effectiveness Plane (LSM vs GLP-1s, Lifetime Base Case)") +
  theme_bw(base_size=14) +
  theme(legend.position="none")

```

Cost-Effectiveness Plane (LSM vs GLP-1s, Lifetime Baseline)



```
#####
##### GENERIC ICER FUNCTION #####
#####

get_target_icer <- function(cea_df){
  cea_df_non_dom <- cea_df %>%
    arrange(Cost) %>%
    mutate(
      dCost = Cost - lag(Cost),
      dQALY = QALY - lag(QALY),
      ICER = dCost / dQALY
    ) %>%
    filter(is.na(lag(dQALY)) | dQALY > 0)

  icer <- cea_df_non_dom$ICER[cea_df_non_dom$Strategy=="Semaglutide"]
  if(length(icer)==0) return(NA_real_)
  icer[1]
}

run_model_generic_icer <- function(rr_regress_vec=rr_regress,
                                    rr_progress_vec=rr_progress,
                                    util_matrix=m_util_base,
                                    cost_vector=costs_base,
                                    drug_cost_vec=drug_cost,
                                    bg_cost_month_vec=v_background_cost_month,
                                    p_prog_month_local=p_prog_month,
```

```

            treat_dur_cycles_vec=base_treat_dur_cycles){

a_P_loc <- build_a_P(rr_regress_vec, rr_progress_vec,
                      p_prog_month_local,
                      treat_dur_cycles=treat_dur_cycles_vec)

traces_loc <- lapply(treatments,function(stg){
  run_markov(a_P_loc[,,stg], v_init)
})
names(traces_loc) <- treatments

res_loc <- summarize_strategies(
  traces_loc, "SA",
  util_matrix,
  bg_cost_month_vec,
  cost_vector,
  drug_cost_vec,
  treat_dur_cycles_vec
)

cea_loc <- calculate_cea(res_loc)
get_target_icer(cea_loc)
}

#####
##### TORNADO #####
#####

# Scale observational fibrosis progression
scale_prog <- function(mult){
  z <- p_prog_month
  z$F0_F1 <- z$F0_F1 * mult
  z$F1_F2 <- z$F1_F2 * mult
  z$F2_F3 <- z$F2_F3 * mult
  z$F3_F4 <- z$F3_F4 * mult
  z
}

run_model_with_discount <- function(disc_cost_rate, disc_qaly_rate){

  old_v_dwc <- v_dwc
  old_v_dwu <- v_dwu

  v_dwc <- 1/(1+disc_cost_rate)^((0:n_cycles)*cycle_length)
  v_dwu <- 1/(1+disc_qaly_rate)^((0:n_cycles)*cycle_length)

  icer <- run_model_generic_icer()

  v_dwc <- old_v_dwc
  v_dwu <- old_v_dwu

  icer
}

```

```

}

run_model_with_start_age <- function(age_start_new){

  old_age_start          <- age_start
  old_age_end            <- age_end
  old_ages_states         <- ages_states
  old_ages_cycles         <- ages_cycles
  old_v_p_bg_annual      <- v_p_bg_annual
  old_v_r_bg_annual      <- v_r_bg_annual
  old_v_p_bg_month        <- v_p_bg_month
  old_v_background_cost_annual <- v_background_cost_annual
  old_v_background_cost_month <- v_background_cost_month
  old_v_util_age_base     <- v_util_age_base
  old_m_util_base         <- m_util_base
  old_a_p_death           <- a_p_death
  age_start <-> age_start_new
  age_end   <-> age_start + time_horizon

  ages_states <- seq(age_start, age_end, by = cycle_length)
  ages_cycles <- ages_states[-length(ages_states)]

  v_p_bg_annual <- approx(mort_cost_df$Age,
                           mort_cost_df$overall_mortality_avg,
                           ages_cycles, rule = 2)$y
  v_r_bg_annual <- prob_to_rate(v_p_bg_annual)
  v_p_bg_month <- rate_to_prob(v_r_bg_annual, cycle_length)

  v_background_cost_annual <-
    approx(mort_cost_df$Age,
           mort_cost_df$background_cost_2025,
           ages_states, rule = 2)$y
  v_background_cost_month <- v_background_cost_annual / 12

  v_util_age_base <- approx(age_vec, util_age_base,
                             ages_states, rule = 2)$y
  m_util_base     <- build_util_matrix(v_util_age_base, qaly_dec_base)

  a_p_death <- array(0, dim = c(n_states, n_cycles),
                     dimnames = list(v_states, NULL))
  for (s in v_states) {
    a_p_death[s, ] <- pmin(v_p_bg_month * hr_excess[s], 0.99)
  }
  a_p_death["Dead", ] <- 1

  icer <- run_model_generic_icer(util_matrix = m_util_base,
                                 bg_cost_month_vec = v_background_cost_month)

  age_start          <-> old_age_start
  age_end            <-> old_age_end
  ages_states        <-> old_ages_states
  ages_cycles        <-> old_ages_cycles
  v_p_bg_annual      <-> old_v_p_bg_annual
}

```

```

v_r_bg_annual <- old_v_r_bg_annual
v_p_bg_month <- old_v_p_bg_month
v_background_cost_annual <- old_v_background_cost_annual
v_background_cost_month <- old_v_background_cost_month
v_util_age_base <- old_v_util_age_base
m_util_base <- old_m_util_base
a_p_death <- old_a_p_death

icer
}

tornado_params <- list()

##### 1. Semaglutide annual cost
lo_drug <- drug_cost; lo_drug["Semaglutide"] <- cost_sema_low
hi_drug <- drug_cost; hi_drug["Semaglutide"] <- cost_sema_high

icer_treat_lo <- run_model_generic_icer(drug_cost_vec = lo_drug)
icer_treat_hi <- run_model_generic_icer(drug_cost_vec = hi_drug)
tornado_params[["treatment_cost"]] <- c(icer_treat_lo, icer_treat_hi)

##### 2. Observational fibrosis progression
icer_progobs_lo <- run_model_generic_icer(p_prog_month_local = scale_prog(0.75))
icer_progobs_hi <- run_model_generic_icer(p_prog_month_local = scale_prog(1.25))
tornado_params[["progression_rate_obs"]] <- c(icer_progobs_lo, icer_progobs_hi)

##### 3. 0% vs 5% for costs & QALYs
icer_disc_lo <- run_model_with_discount(0.00, 0.00)
icer_disc_hi <- run_model_with_discount(0.05, 0.05)
tornado_params[["discount_rate"]] <- c(icer_disc_lo, icer_disc_hi)

##### 4. Utility scale (QALYs)
icer_u_lo <- run_model_generic_icer(util_matrix = m_util_base * 0.9)
icer_u_hi <- run_model_generic_icer(util_matrix = pmin(m_util_base * 1.1, 1))
tornado_params[["qalys"]] <- c(icer_u_lo, icer_u_hi)

##### 5. Semaglutide trial effect on regression
rr_b <- rr_regress["Semaglutide"]

rr_lo <- rr_regress; rr_lo["Semaglutide"] <- rr_b * 0.75
rr_hi <- rr_regress; rr_hi["Semaglutide"] <- rr_b * 1.25

rp_lo <- rr_progress; rp_lo["Semaglutide"] <- 1 / rr_lo["Semaglutide"]
rp_hi <- rr_progress; rp_hi["Semaglutide"] <- 1 / rr_hi["Semaglutide"]

icer_trial_lo <- run_model_generic_icer(rr_regress_vec = rr_lo,
                                         rr_progress_vec = rp_lo)
icer_trial_hi <- run_model_generic_icer(rr_regress_vec = rr_hi,
                                         rr_progress_vec = rp_hi)
tornado_params[["progression_rate_trial"]] <- c(icer_trial_lo, icer_trial_hi)

##### 6. Five years younger / older than base
icer_age_lo <- run_model_with_start_age(age_start - 5)

```

```

icer_age_hi <- run_model_with_start_age(age_start + 5)
tornado_params[["starting_age"]] <- c(icer_age_lo, icer_age_hi)

##### 7. Scale state costs
cost_lo <- costs_base * 0.75
cost_hi <- costs_base * 1.25

icer_cost_lo <- run_model_generic_icer(cost_vector = cost_lo)
icer_cost_hi <- run_model_generic_icer(cost_vector = cost_hi)
tornado_params[["cost"]] <- c(icer_cost_lo, icer_cost_hi)

##### Tornado dataframe & plot #####
fmt_K <- scales::label_dollar(scale = 1e-3, suffix = "K")

param_labels <- c(
  treatment_cost      = "GLP-1 annual cost",
  progression_rate_obs = "Fibrosis progression (observational)",
  discount_rate       = "Discount rate",
  qalys               = "Utility scale (QALYs)",
  progression_rate_trial = "Fibrosis regression effect (trial)",
  starting_age        = "Starting age",
  cost                = "Liver disease state costs"
)

# Tornado dataframe
tornado_df <- bind_rows(lapply(names(tornado_params), function(p){
  data.frame(
    param = p,
    low   = tornado_params[[p]][1],
    high  = tornado_params[[p]][2]
  )
})) %>%
  mutate(
    width     = abs(high - low),
    param_label = param_labels[param],
    low_label  = paste0("Low: ", fmt_K(low)),
    high_label = paste0("High: ", fmt_K(high))
) %>%
  arrange(desc(width)) %>%
  mutate(
    param_label = factor(param_label, levels = rev(unique(param_label))),
    y_id        = as.numeric(param_label)
  )

# Base-case ICER
base_icer <- run_model_generic_icer()

base_label_df <- data.frame(
  x    = base_icer,
  y    = max(tornado_df$y_id) + 0.8,
  lab = paste0("Base-case ICER: ", fmt_K(base_icer))
)

```

```

# ----- TORNADO PLOT -----

ggplot(tornado_df,
       aes(y = param_label)) +

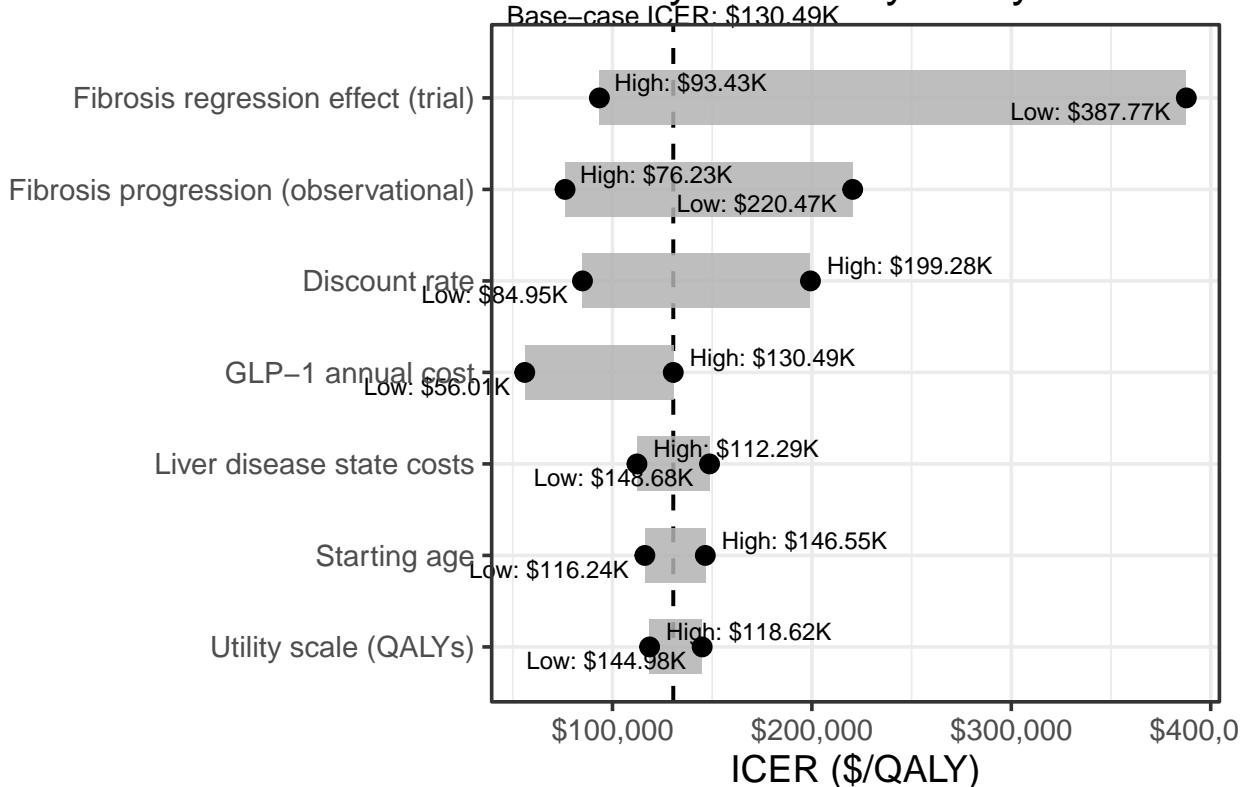
  geom_vline(xintercept = base_icer,
             linetype   = "dashed",
             color      = "black",
             linewidth  = 0.7) +

  geom_rect(aes(
    xmin = pmin(low, high),
    xmax = pmax(low, high),
    ymin = y_id - 0.3,
    ymax = y_id + 0.3
  ),
            fill = "grey70", alpha = 0.85, color = NA) +

  geom_point(aes(x = low, y = param_label), size = 3) +
  geom_point(aes(x = high, y = param_label), size = 3) +
  geom_text(aes(x = low, y = param_label, label = low_label),
            hjust = 1.1, vjust = 1.3, size = 3) +
  geom_text(aes(x = high, y = param_label, label = high_label),
            hjust = -0.1, vjust = -0.4, size = 3) +
  geom_text(data = base_label_df,
            aes(x = x, y = y, label = lab),
            inherit.aes = FALSE,
            hjust = 0.5, vjust = 0,
            size = 3.4) +
  scale_x_continuous(labels = scales::label_dollar()) +
  labs(
    title = "One-Way Sensitivity Analysis",
    x = "ICER ($/QALY)",
    y = NULL
  ) +
  theme_bw(base_size = 14) +
  coord_cartesian(clip = "off")

```

One-Way Sensitivity Analysis



```
#####
##### SCENARIO ANALYSIS #####
#####

run_duration_scenario <- function(label){

  dur_cycles <- treat_dur_scenarios[[label]]

  aP_sc <- build_a_P(
    rr_regress, rr_progress,
    p_prog_month_local=p_prog_month,
    treat_dur_cycles=dur_cycles
  )

  traces_sc <- lapply(treatments, function(stg){
    run_markov(aP_sc[,,stg], v_init)
  })
  names(traces_sc) <- treatments

  summarize_strategies(
    traces_sc, label,
    m_util_base,
    v_background_cost_month,
    costs_base,
    drug_cost,
    treat_dur_cycles_vec=dur_cycles
  )
}
```

```

    }

scenario_labels <- c("72w", "5y", "20y", "Lifetime")
res_scenarios_list <- lapply(scenario_labels, run_duration_scenario)
res_scenarios <- bind_rows(res_scenarios_list)

#####
##### TRACES #####
#####

plot_trace <- function(trace, title="Trace"){
  df <- as.data.frame(trace)
  df$age <- ages_states
  df_l <- pivot_longer(df, cols=v_states)

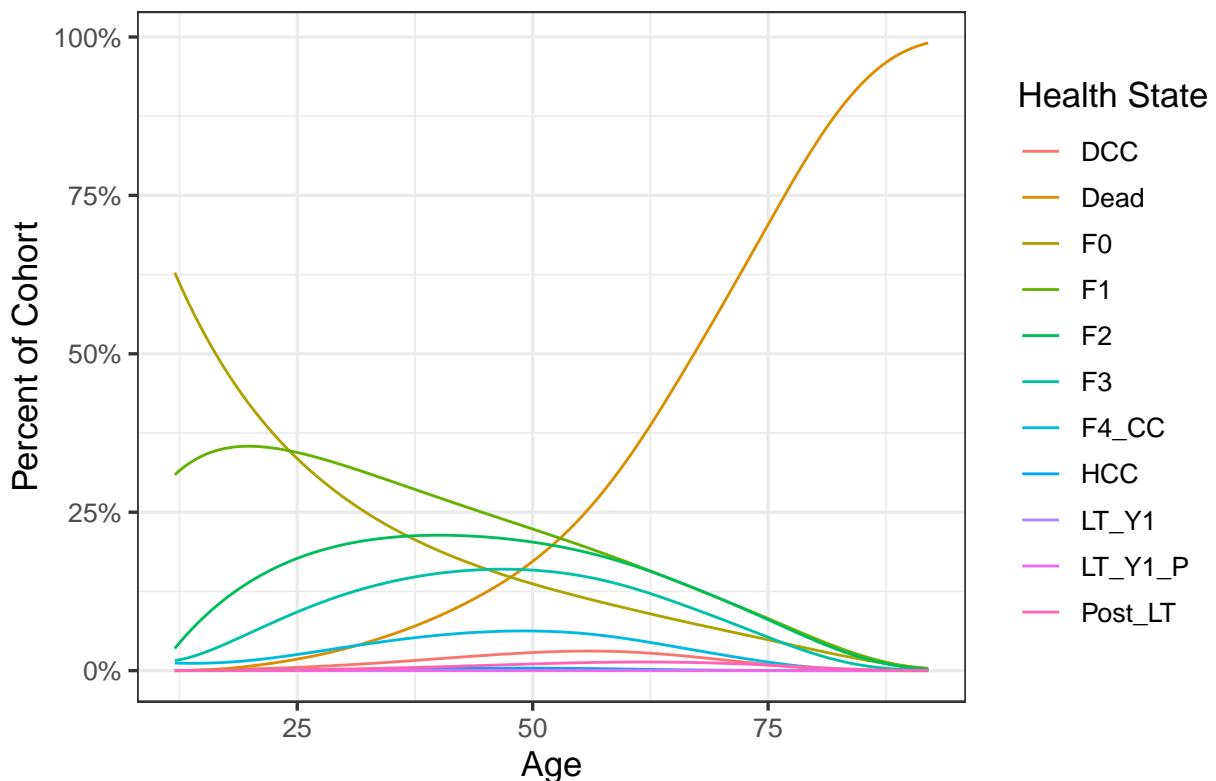
  ggplot(df_l, aes(x=age, y=value, color=name)) +
    geom_line() +
    scale_y_continuous(labels=scales::percent_format(accuracy=1)) +
    labs(x="Age", y="Percent of Cohort", color="Health State", title=title) +
    theme_bw(base_size=13)
}

plot_trace(traces[["LSM"]], "State Occupancy for LSM (Lifetime Base)")

## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use 'all_of()' or 'any_of()' instead.
##   # Was:
##   data %>% select(v_states)
##
##   # Now:
##   data %>% select(all_of(v_states))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

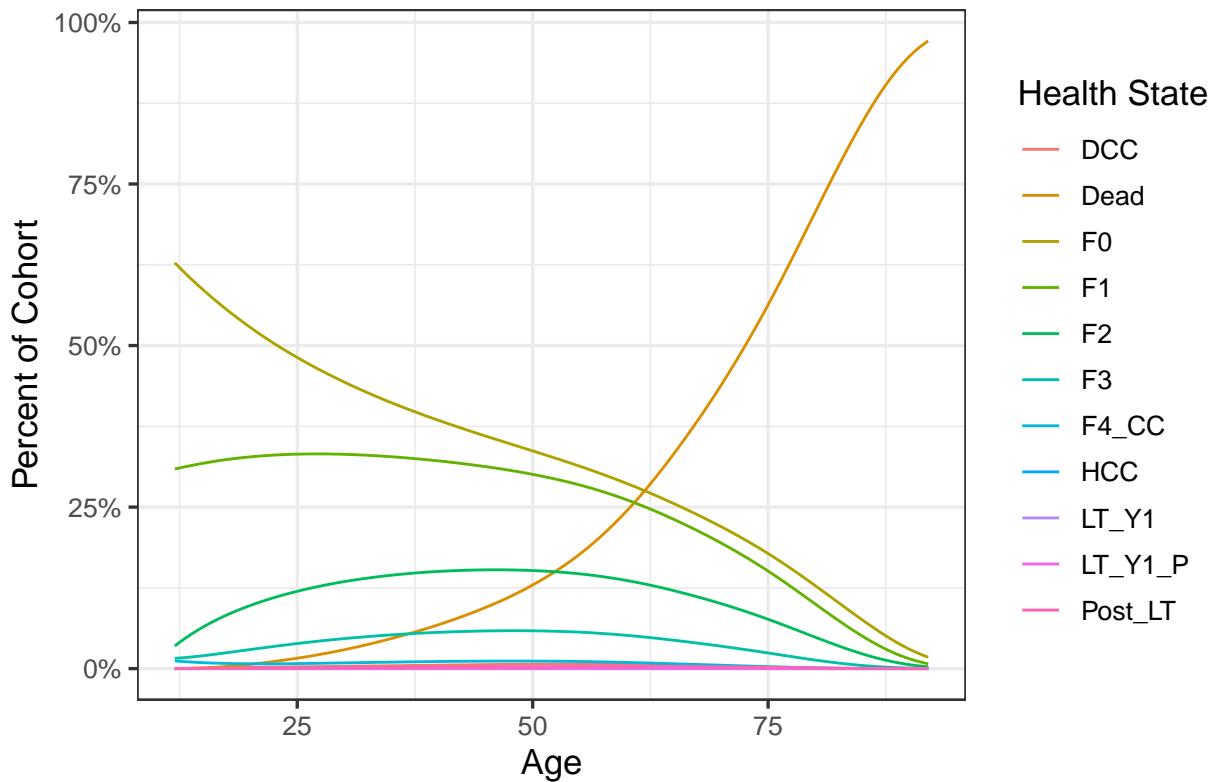
```

State Occupancy for LSM (Lifetime Base)



```
plot_trace(traces[["Semaglutide"]], "State Occupancy for GLP-1s (Lifetime Base)")
```

State Occupancy for GLP-1s (Lifetime Base)



```
#####
##### TRACE DIAGRAMS BY SCENARIO (Semaglutide) #####
#####

build_traces_for_scenario <- function(label) {

  dur_cycles <- treat_dur_scenarios_cycles[[label]]

  aP_sc <- build_a_P(
    rr_regress,
    rr_progress,
    p_prog_month_local = p_prog_month,
    treat_dur_cycles   = dur_cycles
  )

  traces_sc <- lapply(treatments, function(stg){
    run_markov(aP_sc[,,stg], v_init)
  })
  names(traces_sc) <- treatments
  traces_sc
}

# Traces for all scenarios
scenario_traces_list <- lapply(scenario_labels, build_traces_for_scenario)
names(scenario_traces_list) <- scenario_labels
```

```

trace_sem_all <- dplyr::bind_rows(lapply(names(scenario_traces_list), function(sc){

  tr <- scenario_traces_list[[sc]][["Semaglutide"]]

  df <- as.data.frame(tr)
  df$cycle <- 0:n_cycles
  df$age    <- age_start + df$cycle * cycle_length

  df_l <- tidyr::pivot_longer(
    df,
    cols      = dplyr::all_of(v_states),
    names_to  = "State",
    values_to = "Proportion"
  )

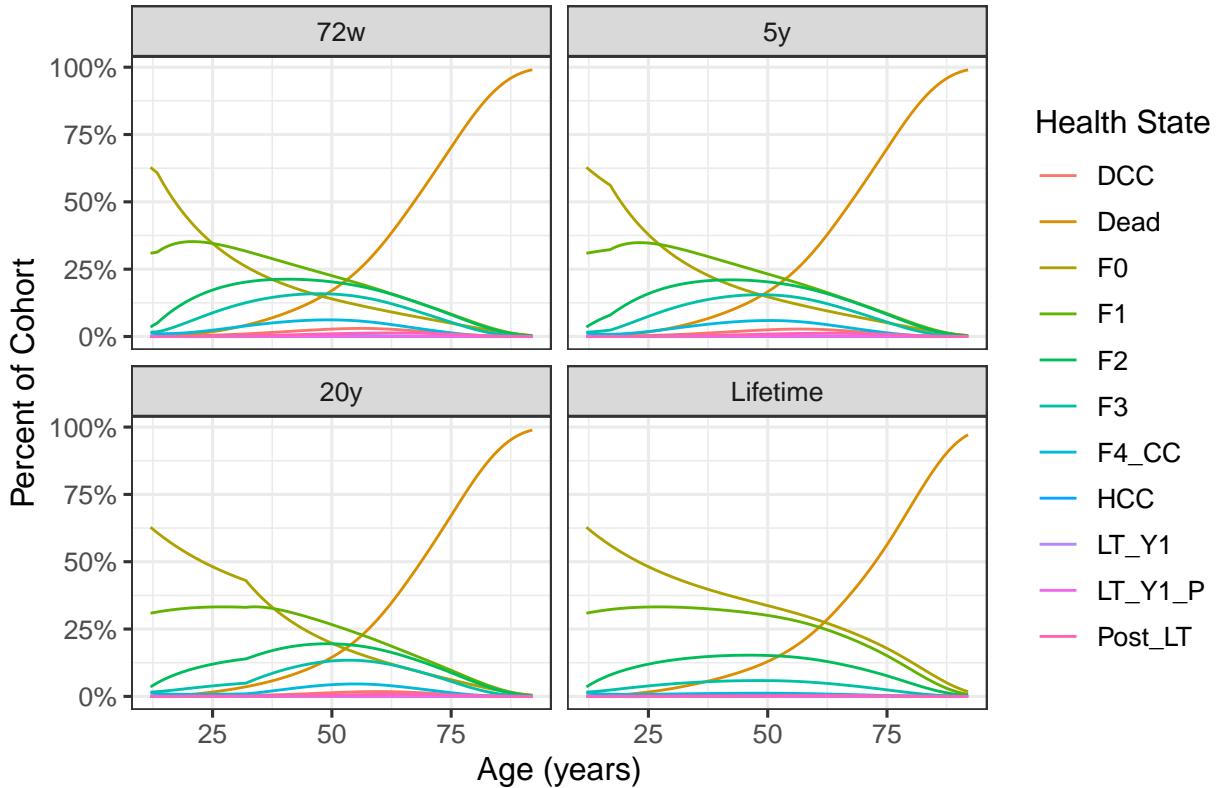
  df_l$Scenario <- sc
  df_l
}))
```

trace_sem_all\$Scenario <- factor(trace_sem_all\$Scenario,
 levels = scenario_labels)

```

ggplot(trace_sem_all,
  aes(x = age, y = Proportion, color = State)) +
  geom_line(linewidth = 0.5) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(
    x      = "Age (years)",
    y      = "Percent of Cohort",
    color = "Health State",
    title = "State Occupancy by GLP-1 Duration"
  ) +
  facet_wrap(~ Scenario, ncol = 2) +
  theme_bw(base_size = 12)
```

State Occupancy by GLP-1 Duration



```
#####
##### Cost-Effectiveness Planes for all Scenarios #####
#####

#####
#####
```

```
ce_plane_all <- res_scenarios %>%
  group_by(Scenario) %>%
  arrange(Cost, .by_group = TRUE) %>%
  ungroup()

ce_plane_all$Scenario <- factor(ce_plane_all$Scenario,
  levels = scenario_labels)

x_range <- range(ce_plane_all$Cost)
y_range <- range(ce_plane_all$QALY)

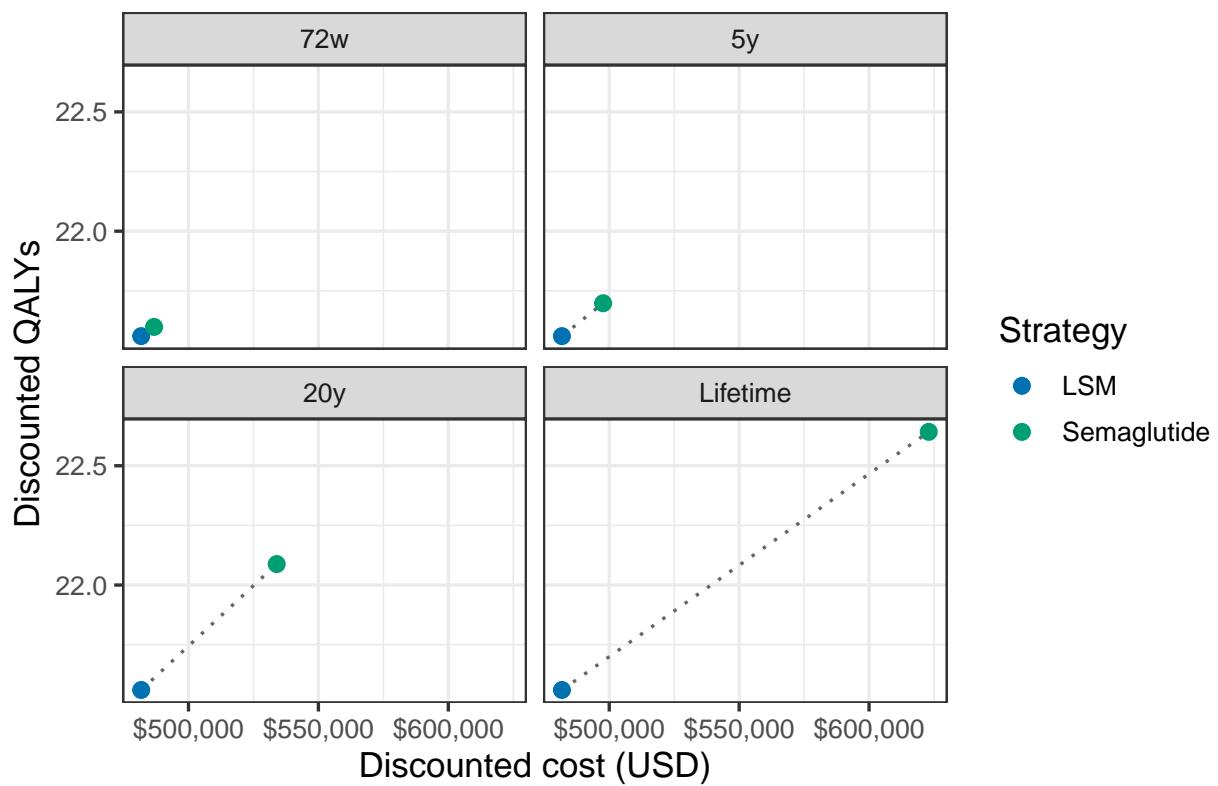
ggplot(ce_plane_all,
  aes(x = Cost, y = QALY, color = Strategy)) +
  geom_line(aes(group = Scenario),
    linetype = "dotted",
    linewidth = 0.6,
    color = "grey40") +
  geom_point(size = 2.5) +
  scale_color_manual(values = okabe_ito_strat) +
  scale_x_continuous(
    limits = x_range,
    labels = label_dollar(scale = 1),
```

```

    breaks = pretty(x_range, n = 3)
) +
scale_y_continuous(
  limits = y_range,
  breaks = pretty(y_range, n = 3)
) +
labs(
  x      = "Discounted cost (USD)",
  y      = "Discounted QALYs",
  title = "Cost-Effectiveness Planes by GLP-1 Duration"
) +
facet_wrap(~ Scenario) +
theme_bw(base_size = 13)

```

Cost-Effectiveness Planes by GLP-1 Duration



```

#####
##### FINAL OUTPUT #####
#####

cat("\n== Base Case CEA (Lifetime, incremental vs cheapest) ==\n")

##
## == Base Case CEA (Lifetime, incremental vs cheapest) ==

```

```

print(cea_base)

##                               Scenario     Strategy     Cost      QALY      LY  dCost
## LSM           Base (Lifetime)       LSM 481796.3 21.56018 25.81150    NA
## Semaglutide Base (Lifetime) Semaglutide 622976.2 22.64214 26.71296 141180
##                         dQALY      ICER
## LSM                 NA        NA
## Semaglutide 1.081959 130485.5

#####
##### CEA for Each Scenario (with ICERs) #####
#####

cea_scenarios <- res_scenarios %>%
  group_by(Scenario) %>%
  do({
    df <- select(., -Scenario)
    calculate_cea_pub(df)
  }) %>%
  ungroup()

cat("\n==== Scenario CEA Tables (with ICERs) ====\n")

##
## === Scenario CEA Tables (with ICERs) ===

print(cea_scenarios)

## # A tibble: 8 x 8
##   Scenario Strategy     QALY     LY     Cost     dCost     dQALY     ICER
##   <chr>     <chr>     <dbl>   <dbl>   <dbl>     <dbl>     <dbl>     <dbl>
## 1 20y       LSM        21.6    25.8  481796.     NA     NA        NA
## 2 20y       Semaglutide 22.1    26.2  533953.  52157.    0.528    98817
## 3 5y        LSM        21.6    25.8  481796.     NA     NA        NA
## 4 5y        Semaglutide 21.7    25.9  497651.  15855.    0.138   115097
## 5 72w       LSM        21.6    25.8  481796.     NA     NA        NA
## 6 72w       Semaglutide 21.6    25.8  486764.  4968.    0.0385  128942
## 7 Lifetime  LSM        21.6    25.8  481796.     NA     NA        NA
## 8 Lifetime  Semaglutide 22.6    26.7  622976. 141180.    1.08    130485

#####
##### ICER by Scenario (Semaglutide vs LSM) #####
#####

icer_scenarios_df <- res_scenarios %>%
  group_by(Scenario) %>%
  arrange(Cost, .by_group = TRUE) %>%
  summarise(
    dCost = Cost[Strategy == "Semaglutide"] - Cost[Strategy == "LSM"],
    dQALY = QALY[Strategy == "Semaglutide"] - QALY[Strategy == "LSM"],
    ICER = dCost / dQALY,

```

```

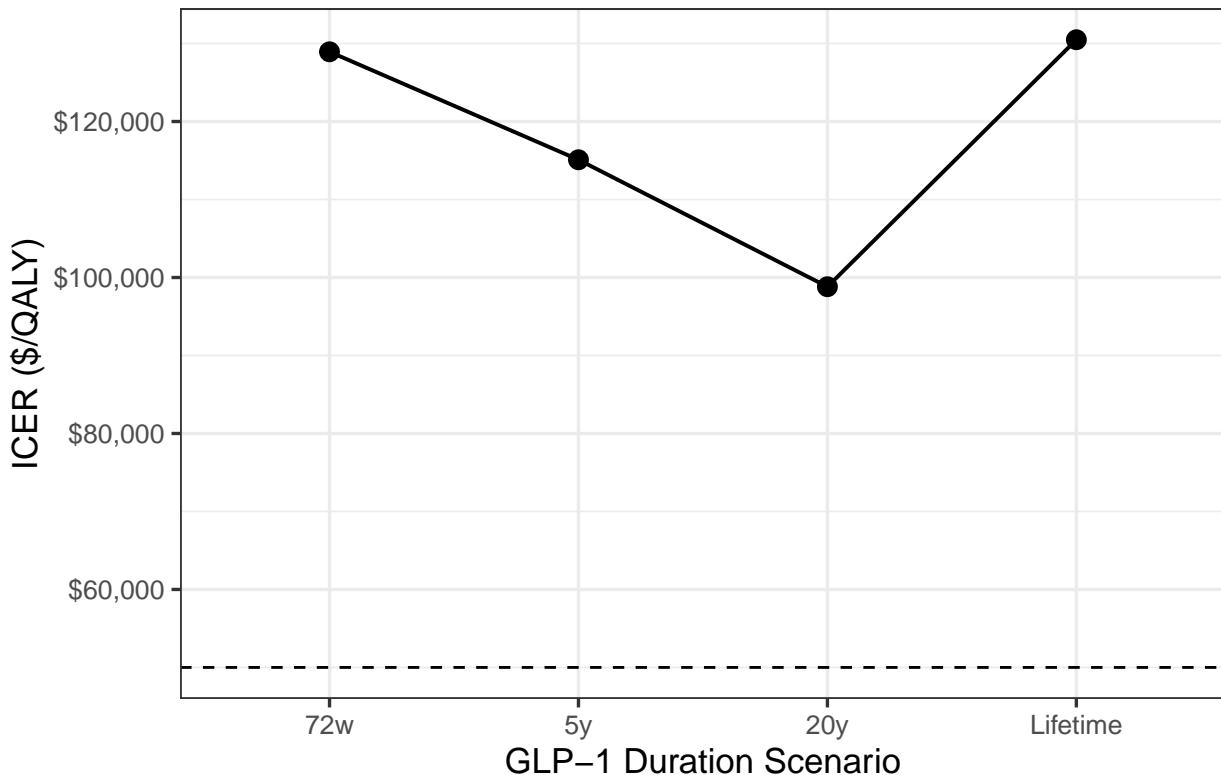
    .groups = "drop"
  )

icer_scenarios_df$Scenario <- factor(icer_scenarios_df$Scenario,
                                      levels = scenario_labels)

ggplot(icer_scenarios_df,
       aes(x = Scenario, y = ICER, group = 1)) +
  geom_hline(yintercept = 50000,
              linetype = "dashed") +
  geom_point(size = 3) +
  geom_line(linewidth = 0.7) +
  scale_y_continuous(labels = label_dollar(scale = 1)) +
  labs(
    title = "ICER by GLP-1 Duration",
    x     = "GLP-1 Duration Scenario",
    y     = "ICER ($/QALY)"
  ) +
  theme_bw(base_size = 13)

```

ICER by GLP-1 Duration



```
cat("\n==== END OF MODEL ===\n")
```

```
##  
## === END OF MODEL ===
```