
proj: sage modeling for the cps repeal proposal

date: march, 2025

auth: national center for environmental economics

Instructions:

This text needs to be copied to a .R file.

This R script is then meant to be run from the root directory of the SAGE model replication files used in the 2024 Final CPS rulemaking.

In addition to running the SAGE model, this file will generate the figures in RIA Section 5.2.

The file "table_5-3_social_costs.xlsx" performs the discounting to create RIA Table 5-3.

set working directory to the root directory of the SAGE model replication files used in the 2024 Final CPS rulemaking.

```
setwd("")
```

output directory

```
output_dir = file.path("output","cps_repeal_proposal")
```

model file

```
model_file = "sage_retrofit.gms"
```

benchmark file

```
benchmark_file = "default_aggregation.gdx"
```

parameters file

```
parameter_file = "parameters.gms"
```

baseline file

```
baseline_file = "baseline.gms"
```

load libraries

load in the R utilities for sage

```
source(file.path("utilities", "R_utilities.R"))
```

define needed packages for running the model and plotting results

```
list.of.packages <-  
c("reshape2", "ggplot2", "dplyr", "tidyverse", "plotly", "knitr",  
  "kableExtra", "readxl", "tidyr", "zoo", "scales")
```

install packages if not on machine

```
new.packages <- list.of.packages[  
  !(list.of.packages %in% installed.packages()[, "Package"])]  
  
if(length(new.packages))  
  install.packages(new.packages, repos = "http://cran.rstudio.com/")
```

load package list

```
lapply(list.of.packages, library, character.only = TRUE)
```

scenario options

ria scenarios

```
ria_scen = c("12b_iter1_240324")
```

capital accounting assumption

```
cap_account = c("ipm")
```

subsidy assumption

```
subsidy = c("with_subsidy")
```

construct directory structure to house model outputs

create output directory if not exist

```
dir.create(output_dir, showWarnings=F)
```

remove any existing files in output directories

```
f <- list.files(output_dir, recursive=T)
file.remove(file.path(output_dir, f))
```

create temporary directories

```
dir.create(file.path(output_dir, "csv"), showWarnings=F)
dir.create(file.path(output_dir, "gdx"), showWarnings=F)
dir.create(file.path(output_dir, "lst"), showWarnings=F)
```

run model scenarios

solve for sage_retrofit baseline

```
run.model(model = model_file,
benchmark_file = benchmark_file,
parameter_file = parameter_file,
baseline_file = baseline_file,
output_file = "baseline.csv",
options = paste0("o=", file.path(output_dir, "lst", "sage_retrofit_baseline.lst")))
```

calibrate productivity parameters to ipm outputs

```
for (rs in ria_scen) {
  for (ca in cap_account) {
    for (sb in subsidy) {
```

```
run.model(model=model_file,
  benchmark_file = benchmark_file,
  parameter_file = parameter_file,
  baseline_file = baseline_file,
  gdx_baseline_file = "baseline.gdx",
  options = paste0("--epilogue=", file.path("model", "calibrate_sage_ipm.gms"),
    " --ria_scen=", rs,
    " --subsidy=", sb,
```

```

        " --cap_account=", ca,
        " --output_dir=", output_dir,
        "
o=", file.path(output_dir, "lst", paste0("calibrate_sage_", rs, "_", ca, "_", sb, ".lst"))
    )
}
}
}

```

use calibrated productivity parameters to solve for general equilibrium

solution

```

for (rs in ria_scen) {
for (ca in cap_account) {
for (sb in subsidy) {

```

```

    run.model(model=model_file,
              benchmark_file = benchmark_file,
              parameter_file = parameter_file,
              baseline_file = baseline_file,
              gdx_baseline_file = "baseline.gdx",
              policy_file = file.path("model", "ipm_policy_scenario.gms"),
              output_file = paste0("results_", rs, "_", ca, "_", sb, ".csv"),
              options = paste0(" --ria_scen=", rs,
                              " --subsidy=", sb,
                              " --cap_account=", ca,
                              " --output_dir=", output_dir,
                              "
o=", file.path(output_dir, "lst", paste0("policy_", rs, "_", ca, "_", sb, ".lst")))
    )
}
}
}

```

generate csv files of elements from baseline for weighting

y0

```

system(paste0("gdxdump ", file.path("output", "baseline.gdx"),
" output=", file.path(output_dir, "csv", "y0.csv"),
" symb=y0 format=csv"))

```

a0

```
system(paste0("gdxdump ",file.path("output","baseline.gdx"),
" output=",file.path(output_dir,"csv","a0.csv"),
" symb=a0 format=csv"))
```

ty0

```
system(paste0("gdxdump ",file.path("output","baseline.gdx"),
" output=",file.path(output_dir,"csv","ty0.csv"),
" symb=ty0 format=csv"))
```

id0

```
system(paste0("gdxdump ",file.path("output","baseline.gdx"),
" output=",file.path(output_dir,"csv","id0.csv"),
" symb=id0 format=csv"))
```

n0

```
system(paste0("gdxdump ",file.path("output","baseline.gdx"),
" output=",file.path(output_dir,"csv","n0.csv"),
" symb=n0 format=csv"))
```

cd0

```
system(paste0("gdxdump ",file.path("output","baseline.gdx"),
" output=",file.path(output_dir,"csv","cd0.csv"),
" symb=cd0 format=csv"))
```

consolidate results

grab calibration verification

```
shock = data.frame()
```

```
calib_report = data.frame()
```

```
for (rs in ria_scen) {
```

```
  for (ca in cap_account) {
```

```
    for (sb in subsidy) {
```

```
      # input shock report
```

```
      system(paste0("gdxdump
```

```
",file.path(output_dir,"gdx",paste0("cps_repeal_proposal_calibration_",rs,"_",ca,"_",
sb,".gdx")),
```

```
      "
```

```
      output=",file.path(output_dir,"csv",paste0("cps_repeal_proposal_report_shock_",rs,"_",
ca,"_",sb,".csv")),
```

```

        " symb=report_shock format=csv"))
temp =
read.csv(file.path(output_dir, "csv", paste0("cps_repeal_proposal_report_shock_", rs, "_"
, ca, "_", sb, ".csv")))
temp$ria_scen = rs
temp$cap_account = ca
temp$subsidy = "with_subsidy"
shock = rbind(shock, temp)

# calibration report
system(paste0("gdxdump
", file.path(output_dir, "gdx", paste0("cps_repeal_proposal_calibration_", rs, "_", ca, "_",
sb, ".gdx")),
        "
output=", file.path(output_dir, "csv", paste0("cps_repeal_proposal_calib_report_", rs, "_"
, ca, "_", sb, ".csv")),
        " symb=calib_report format=csv"))
temp =
read.csv(file.path(output_dir, "csv", paste0("cps_repeal_proposal_calib_report_", rs, "_"
, ca, "_", sb, ".csv")))
temp$ria_scen = rs
temp$cap_account = ca
temp$subsidy = "with_subsidy"
calib_report = rbind(calib_report, temp)
}

}
}

```

shock = shock %>%

rename(t=Dim1, r=Dim2, v=Dim3, ss=Dim4, s=Dim5, parm=Dim6, value=Val)

calib_report = calib_report %>%

rename(t=Dim1, r=Dim2, v=Dim3, s=Dim4, model=Dim5, calib_mech=Dim6, value=Val)

grab the baseline

baseline = read.csv(file.path("output", "baseline.csv")) %>%

rename("baseline"="value")

grab the post-policy results

results = data.frame()

for (rs in ria_scen) {

for (ca in cap_account) {

for (sb in subsidy) {

```
temp = read.csv(file.path("output", paste0("results_", rs, "_", ca, "_", sb, ".csv"))) %>%
```

```
  rename("policy"="value") %>% distinct()
```

```
temp$ria_scen = rs
```

```
temp$cap_account = ca
```

```
temp$subsidy = "with_subsidy"
```

```
results = rbind(results, temp)
```

```
}
```

```
}
```

```
}
```

merge baseline and policy files

```
results = results %>% left_join(baseline)
```

relabeling functions

```
relabel.sectors = function(x) {  
  x$label[x$sector=="agf"] = "Agriculture, forestry, fishing and hunting"  
  x$label[x$sector=="cru"] = "Crude oil"  
  x$label[x$sector=="col"] = "Coal mining"  
  x$label[x$sector=="min"] = "Metal ore and nonmetallic mineral mining"  
  x$label[x$sector=="ele"] = "Electric power"  
  x$label[x$sector=="gas"] = "Natural gas"  
  x$label[x$sector=="wsu"] = "Water, sewage, and other utilities"  
  x$label[x$sector=="con"] = "Construction"  
  x$label[x$sector=="fbm"] = "Food and beverage manufacturing"  
  x$label[x$sector=="wpm"] = "Wood product manufacturing"  
  x$label[x$sector=="ref"] = "Petroleum refineries"  
  x$label[x$sector=="chm"] = "Chemical manufacturing"  
  x$label[x$sector=="prm"] = "Plastics and rubber products manufacturing"  
  x$label[x$sector=="cem"] = "Cement manufacturing"  
  x$label[x$sector=="pmm"] = "Primary metal manufacturing"  
  x$label[x$sector=="fmm"] = "Fabricated metal product manufacturing"  
  x$label[x$sector=="cpu"] = "Electronics and technology manufacturing"  
  x$label[x$sector=="tem"] = "Transportation equipment manufacturing"  
  x$label[x$sector=="bom"] = "Other manufacturing"  
  x$label[x$sector=="trn"] = "Transportation"  
  x$label[x$sector=="ttn"] = "Truck transportation"  
  x$label[x$sector=="srv"] = "Services"  
  x$label[x$sector=="hlt"] = "Healthcare services"  
  x$label[x$sector=="all"] = "All sectors"  
  x$label = factor(x$label,c("Agriculture, forestry, fishing and hunting",  
    "Crude oil",  
    "Electric power",  
    "Coal mining",  
    "Metal ore and nonmetallic mineral mining",  
    "Natural gas",  
    "Water, sewage, and other utilities",  
    "Construction",
```

```

"Food and beverage manufacturing",
"Wood product manufacturing",
"Petroleum refineries",
"Chemical manufacturing",
"Plastics and rubber products manufacturing",
"Cement manufacturing",
"Primary metal manufacturing",
"Fabricated metal product manufacturing",
"Electronics and technology manufacturing",
"Transportation equipment manufacturing",
"Other manufacturing",
"Transportation",
"Truck transportation",
"Services",
"Healthcare services",
"All sectors"))
return(x)
}

```

plot ipm solution

overnight capital costs

```

on_capital = subset(shock, parm %in% "on_capital" & cap_account %in% "ipm" & subsidy %in%
"with_subsidy")
on_capital$ss = NULL
on_capital$s = NULL

on_capital$region[on_capital$region=="mid"] = "Midwest"
on_capital$region[on_capital$region=="nor"] = "Northeast"
on_capital$region[on_capital$region=="sou"] = "South"
on_capital$region[on_capital$region=="wes"] = "West"

on_capital$stage[on_capital$stage=="extant"] = "Extant"
on_capital$stage[on_capital$stage=="new"] = "New"

p <- ggplot(data=on_capital) +
  geom_bar(aes(x=t, y=value, fill=v),
  stat="identity", position=position_dodge(), colour="black") +
  scale_fill_manual(values=sage.palette) +
  geom_hline(yintercept=0) +
  labs(y="billions 2016$", x="Year", fill="Production\nVintage") +
  theme_bw() +
  facet_grid(ria_scen~r) +
  theme(text = element_text(color="black", family = "Inconsolata"),

```

```

panel.border = element_rect(colour = "black"),
axis.line = element_line(colour = "black"))
p
ggsave(file.path(output_dir,"overnight_capital.png"), width=8,height=4)

```

aggregated ipm_sol

```

agg_sol = subset(shock, parm %in% "agg_sol")
agg_sol$s = NULL

```

```

agg_sol$region[agg_sol$region=="mid"] = "Midwest"
agg_sol$region[agg_sol$region=="nor"] = "Northeast"
agg_sol$region[agg_sol$region=="sou"] = "South"
agg_sol$region[agg_sol$region=="wes"] = "West"

```

```

agg_sol$extant[agg_sol$extant=="extant"] = "Extant"
agg_sol$extant[agg_sol$extant=="new"] = "New"

```

order inputs

```

agg_sol$ss = factor(agg_sol$ss,
levels=c("k","l","col","gas","ref","mat"))

p <- ggplot(data=agg_sol) +
  geom_line(aes(x=t, y=-value, color=ss), lwd=2) +
  scale_color_manual(values=sage.palette) +
  geom_hline(yintercept=0) +
  labs(y="billions 2016$", x="Year",color="Inputs") +
  theme_bw() +
  facet_grid(cap_account+ria_scen+v+subsidy~r) +
  theme(text = element_text(color="black",family = "Inconsolata"),
axis.text.x = element_text(color="black", angle=45, vjust=0.5),
panel.border = element_rect(colour = "black"),
axis.line = element_line(colour = "black"))
p
ggsave(file.path(output_dir,"incremental_costs.png"), width=8,height=6)

```

verify calibration

check incremental by capital, labor, fuel, and other materials

```

chk_calib = calib_report %>%
filter(model %in% c("sage","ipm") & !(s %in% c("backed_out_cc_2016dollars","sum_ipm_costs")))
chk_calib$label[chk_calib$ss=="k"] = "Capital"
chk_calib$label[chk_calib$ss=="l"] = "FOM"
chk_calib$label[chk_calib$ss=="col"] = "Coal"

```

```

chk_calib$label[chk_calib$s=="gas"] = "Gas"
chk_calib$label[chk_calib$s=="ref"] = "Oil"
chk_calib$label[chk_calib$s=="agf"] = "VOM"
chk_calib$label[chk_calib$s=="cru"] = "VOM"
chk_calib$label[chk_calib$s=="min"] = "VOM"
chk_calib$label[chk_calib$s=="ele"] = "VOM"
chk_calib$label[chk_calib$s=="wsu"] = "VOM"
chk_calib$label[chk_calib$s=="con"] = "VOM"
chk_calib$label[chk_calib$s=="fbm"] = "VOM"
chk_calib$label[chk_calib$s=="wpm"] = "VOM"
chk_calib$label[chk_calib$s=="chm"] = "VOM"
chk_calib$label[chk_calib$s=="prm"] = "VOM"
chk_calib$label[chk_calib$s=="cem"] = "VOM"
chk_calib$label[chk_calib$s=="pmm"] = "VOM"
chk_calib$label[chk_calib$s=="fmm"] = "VOM"
chk_calib$label[chk_calib$s=="cpu"] = "VOM"
chk_calib$label[chk_calib$s=="tem"] = "VOM"
chk_calib$label[chk_calib$s=="bom"] = "VOM"
chk_calib$label[chk_calib$s=="trn"] = "VOM"
chk_calib$label[chk_calib$s=="ttn"] = "VOM"
chk_calib$label[chk_calib$s=="srv"] = "VOM"
chk_calib$label[chk_calib$s=="hlt"] = "VOM"

chk_calib = chk_calib %>%
  filter(t < 2061 & t>2021) %>%
  group_by(t,v,label,model,ria_scen,cap_account,subsidy) %>%
  summarize(value = sum(value,na.rm=TRUE))

chk_calib$v[chk_calib$v=="extant"] = "Extant"
chk_calib$v[chk_calib$v=="new"] = "New"
chk_calib$model[chk_calib$model=="sage"] = "SAGE-PE"
chk_calib$model[chk_calib$model=="ipm"] = "IPM"

plot_data = data.frame(chk_calib) %>%
  tidyr::expand(t,v,label,model,ria_scen,cap_account,subsidy)

plot_data = plot_data %>%
  left_join(chk_calib)
plot_data$value[is.na(plot_data$value)] = 0
plot_data$label = factor(plot_data$label,
  c("Coal",
    "Gas",
    "Oil",
    "FOM",
    "VOM",
    "Capital"))

p <- ggplot(data=plot_data) +
  geom_line(aes(x=t, y=-value, color=model, linetype=model), lwd=2) +
  scale_color_manual(values=sage.palette) +

```

```
geom_hline(yintercept=0) +
labs(y="billions 2016$", x="Year",color="Model",linetype="Model") +
theme_bw() +
facet_grid(ria_scen+cap_account+v+subsidy~label) +
theme(text = element_text(color="black",family = "Inconsolata"),
axis.text.x = element_text(color="black", angle=45, vjust=0.5),
panel.border = element_rect(colour = "black"),
axis.line = element_line(colour = "black"))
p
ggsave(file.path(output_dir,"calibration_check.png"), width=6,height=4)
```

social costs (aggregate and distributional)

```
stream_ev = results %>% filter(parameter %in% "exp_diff" & subsidy %in% "with_subsidy") %>%
rename("ev"="policy") %>% select(time,ria_scen,region,household,ev)

cpi_ev = results %>% filter(parameter %in% "pfx" & subsidy %in% "with_subsidy") %>%
rename("cpi"="policy") %>% select(time,cpi)

stream_ev = left_join(stream_ev, cpi_ev)
stream_ev$ev_inflate = stream_ev$ev / stream_ev$cpi
stream_ev$ev = NULL
stream_ev$cpi = NULL
```

convert to 2024 dollar years

(<https://www.govinfo.gov/content/pkg/ECONI-2025-02/pdf/ECONI-2025-02-Pg2-1.pdf>)

```
price_inflate = 125.234 / 98.208
stream_ev$ev_inflate = price_inflate * stream_ev$ev
```

report stream of undiscounted EV for social cost accounting -- calculations

done in excel

```
stream_modelyears = stream_ev %>%
group_by(time) %>%
summarize(sc = sum(ev_inflate, na.rm=TRUE))
write.csv(stream_modelyears, file.path(output_dir,"csv","stream_ev_modelyears.csv"),
row.names=FALSE)
```

distributional plot: add years between model years and interpolate using

nearest number

```
ev_all_years_hh = expand.grid(seq(2016,2051,1),
c("nor","sou","mid","wes"),
c("hh1","hh2","hh3","hh4","hh5")) %>%
rename("time"="Var1","region"="Var2","household"="Var3") %>%
mutate(time_join = time-2) %>%
left_join(stream_ev, by=join_by(region==region,household==household,closest(time_join<=time))) %>%
select("time.x","region","household","ev_inflate") %>%
rename("time"="time.x")

pv_min_year = 2024
pv_max_year = 2047
dr = 0.045
```

```
ev_all_years_hh$df = 1/((1+dr)**(ev_all_years_hh$time-pv_min_year))
ev_all_years_hh$df[ev_all_years_hh$time<pv_min_year] = 0
ev_all_years_hh$df[ev_all_years_hh$time>pv_max_year] = 0
ev_all_years_hh$sc = ev_all_years_hh$df * (ev_all_years_hh$ev_inflate)
```

read in number of households

```
n0 = read.csv(file.path(output_dir,"csv","n0.csv")) %>%
rename(n0 = Val)

region_hh_ev = ev_all_years_hh %>%
group_by(region,household) %>%
summarise(sc = sum(sc, na.rm=TRUE),
df = sum(df, na.rm=TRUE)) %>%
left_join(n0, by=c("region"="r","household"="h")) %>%
mutate(annual_ev_per_household = 1e9*(sc/df) / (n01e6),
pv_ev_per_household = 1e9(sc) / (n0*1e6))

region_hh_ev$region[region_hh_ev$region=="mid"] = "Midwest"
region_hh_ev$region[region_hh_ev$region=="nor"] = "Northeast"
region_hh_ev$region[region_hh_ev$region=="sou"] = "South"
region_hh_ev$region[region_hh_ev$region=="wes"] = "West"

region_hh_ev$region = factor(region_hh_ev$region,
levels = c("Midwest","Northeast","South","West"))

region_hh_ev$household[region_hh_ev$household=="hh1"] = "<30k"
region_hh_ev$household[region_hh_ev$household=="hh2"] = "30-50k"
region_hh_ev$household[region_hh_ev$household=="hh3"] = "50-70k"
region_hh_ev$household[region_hh_ev$household=="hh4"] = "70-150k"
region_hh_ev$household[region_hh_ev$household=="hh5"] = ">150k"
```

```

region_hh_ev$household = factor(region_hh_ev$household,
levels = c("<30k","30-50k","50-70k","70-150k",">150k"))

p = ggplot(region_hh_ev) +
geom_point(aes(x=household,y=annual_ev_per_household,color=region),fill=NA,
stroke=1,size=3)+
scale_color_manual(values=sage.palette)+
scale_y_continuous(breaks=round(seq(min(region_hh_ev$annual_ev_per_household),
max(region_hh_ev$annual_ev_per_household), by=2),0)) +
labs(x="Income Quintile",
y="Annualized Social Cost\ner Household [2024$]",
color="Regions")+
theme_bw() +
theme(text = element_text(color="black",family = "Inconsolata"),
axis.text.y = element_text(color="black", hjust=0),
axis.title.y = element_text(vjust = +3),
axis.title.x = element_text(vjust = -0.75),
panel.border = element_rect(colour = "black"),
axis.line = element_line(colour = "black"))
p
ggsave(file.path(output_dir,"region_hh_ev_distributional.png"), width=6,height=4)

p = ggplot(region_hh_ev) +
geom_point(aes(x=household,y=pv_ev_per_household,color=region),fill=NA,
stroke=1,size=3)+
scale_color_manual(values=sage.palette)+

scale_y_continuous(breaks=round(seq(min(-
region_hh_ev$annual_ev_per_household), max(-
region_hh_ev$annual_ev_per_household), by=2),0)) +
labs(x="Income Quintile",
y="Present Value of Social Cost\ner Household [2024$]",
color="Regions")+
theme_bw() +
theme(text = element_text(color="black",family = "Inconsolata"),
axis.text.y = element_text(color="black", hjust=0),
axis.title.y = element_text(vjust = +3),
axis.title.x = element_text(vjust = -0.75),
panel.border = element_rect(colour = "black"),
axis.line = element_line(colour = "black"))
p
ggsave(file.path(output_dir,"pv_region_hh_ev_distributional.png"), width=6,height=4)

```

% change in real gdp and components

```
gdp = results %>% filter(parameter %in% c("gdp_real","c_real","i_real","g_real","x_m_real") &
subsidy %in% "with_subsidy") %>%
group_by(time,parameter,ria_scen,cap_account,subsidy) %>%
summarize(policy = sum(policy,na.rm=TRUE),
baseline = sum(baseline,na.rm=TRUE)) %>%
mutate(pct_change = 100 * (policy/baseline - 1),
diff = policy - baseline) %>%
filter(time <=2056)
```

rename components

```
gdp$parameter[gdp$parameter=="c_real"] = "C"
gdp$parameter[gdp$parameter=="i_real"] = "I"
gdp$parameter[gdp$parameter=="g_real"] = "G"
gdp$parameter[gdp$parameter=="x_m_real"] = "X-M"
gdp$parameter[gdp$parameter=="gdp_real"] = "GDP"

gdp$parameter = factor(gdp$parameter,
c("C","I","G","X-M","GDP"))

gdp_plot <-
ggplot(gdp,
aes(x=time, y=-pct_change, color=parameter)) +
geom_line(lwd=2) +
scale_color_manual(values=sage.palette) +
scale_y_continuous(expand=c(0,0), limits=c(-.15,.15),
labels=scales::label_percent(accuracy=0.01,scale=1)) +
labs(y="% Change", x="Year", color="Real GDP\nComponents") +
theme_bw() +
geom_hline(yintercept=0) +
theme(text = element_text(color="black",family = "Inconsolata"),
axis.text.x = element_text(color="black"),
axis.text.y = element_text(color="black"),
axis.title.y = element_text(vjust = +3),
axis.title.x = element_text(vjust = -0.75),
panel.border = element_rect(colour = "black"),
axis.line = element_line(colour = "black"),
legend.position = "right")
gdp_plot
ggsave(file.path(output_dir,"real_gdp.png"),width=8,height=4)
```

sectoral price and quantity impacts

output quantities

```
output = results %>% filter(parameter %in% c("y", "y_ex") & subsidy %in% "with_subsidy") %>%  
group_by(time, sector, ria_scen, cap_account) %>%  
summarize(baseline = sum(baseline, na.rm=TRUE),  
policy = sum(policy, na.rm=TRUE)) %>%  
mutate(pct_change = 100 * (policy/baseline - 1)) %>%  
filter(time %in% c(2026, 2031, 2036, 2041)) %>%  
mutate(impact = "Output") %>%  
select(-c(baseline, policy))
```

reference absorption

```
y0 = read.csv(file.path(output_dir, "csv", "y0.csv")) %>%  
rename(y0=Val, region=r, sector=s)
```

report price changes relative to cpi

```
cpi = results %>% filter(parameter %in% "cpi" & subsidy %in% "with_subsidy") %>%  
rename("cpi_policy"="policy", "cpi_baseline"="baseline") %>%  
select(time, ria_scen, cap_account, cpi_policy, cpi_baseline)
```

real output prices

```
py = results %>% filter(parameter %in% "py" & subsidy %in% "with_subsidy") %>%  
rename("py_policy"="policy", "py_baseline"="baseline") %>%  
left_join(cpi) %>%  
left_join(y0) %>%  
group_by(time, sector, ria_scen, cap_account) %>%  
summarize(pct_change = 100 * (sum((py_policy/cpi_policy)*y0)/sum((py_baseline/cpi_baseline)*y0) -  
1)) %>%  
filter(time %in% c(2026, 2031, 2036, 2041)) %>%  
mutate(impact = "Real Output Price")
```

add together

```
sectoral = rbind(output, py)  
sectoral = relabel.sectors(sectoral)  
sectoral$time = as.character(sectoral$time)  
sectoral$time = factor(sectoral$time,  
c("2026", "2031", "2036", "2041"))
```

plot

```
p = ggplot(sectoral) +  
  geom_point(aes(x=label,y=-pct_change, color=time),fill=NA,  
stroke=1,size=3)+  
  scale_color_manual(values=sage.palette)+  
  scale_y_continuous(expand=c(0,0), limits=c(-1.5,1.4), labels=scales::label_percent(accuracy=0.5,  
scale=1)) +  
  labs(x="Sectors",  
y="% Change",  
color="Year")+  
  coord_flip() +  
  theme_bw() +  
  facet_wrap(~impact) +  
  theme(text = element_text(color="black",family = "Inconsolata"),  
axis.text.x = element_text(color="black"),  
axis.text.y = element_text(color="black", hjust=0),  
axis.title.y = element_text(vjust = +3),  
axis.title.x = element_text(vjust = -0.75),  
panel.border = element_rect(colour = "black"),  
axis.line = element_line(colour = "black"),  
legend.position = "bottom")  
p  
ggsave(file.path(output_dir,"sectoral_impacts.png"),width=7.5,height=5.5)
```

consumer price impact

pull cpi

```
cpi = results %>% filter(parameter %in% "cpi" & subsidy %in% "with_subsidy") %>%  
  rename("cpi_policy"="policy", "cpi_baseline"="baseline") %>%  
  select(time,ria_scen,cap_account,cpi_policy,cpi_baseline) %>%  
  mutate(pct_change = 100 * (cpi_policy/cpi_baseline - 1))
```

```
filter(time %in% c(2026,2031,2036,2041))
```

plot

```
p = ggplot(cpi) +  
  geom_bar(aes(x=time,y=-pct_change),fill=sage.palette[1], colour="black",  
stat="identity", position=position_dodge()+  
  labs(x="Time",  
y="% Change")+  
  theme_bw() +
```

```

theme(text = element_text(color="black",family = "Inconsolata"),
axis.text.x = element_text(color="black"),
axis.text.y = element_text(color="black", hjust=0),
axis.title.y = element_text(vjust = +3),
axis.title.x = element_text(vjust = -0.75),
panel.border = element_rect(colour = "black"),
axis.line = element_line(colour = "black"),
legend.position = "bottom")
p
ggsave(file.path(output_dir,"cpi.png"),width=8,height=4)

```

% change labor demand impacts (quantity)

national impacts

```

ld = results %>% filter(parameter %in% c("ld","ld_ex") & subsidy %in% "with_subsidy") %>%
group_by(time,sector,ria_scen,cap_account) %>%
summarize(baseline = sum(baseline,na.rm=TRUE),
policy = sum(policy, na.rm=TRUE)) %>%
mutate(pct_change = 100 * (policy/baseline - 1)) %>%
filter(time %in% c(2026,2031,2036,2041))

all_sectors_ld = results %>% filter(parameter %in% c("ld","ld_ex") & subsidy %in% "with_subsidy") %>%
group_by(time,ria_scen,cap_account) %>%
summarize(baseline = sum(baseline,na.rm=TRUE),
policy = sum(policy, na.rm=TRUE)) %>%
mutate(pct_change = 100 * (policy/baseline - 1)) %>%
filter(time %in% c(2026,2031,2036,2041)) %>%
mutate(sector = "all") %>%
select(time,sector,ria_scen,cap_account,baseline,policy,pct_change)

ld = rbind(ld,all_sectors_ld)

```

characterize energy and non-energy sectors

```

ipm_sectors = c("ele","gas","col")
ld$sector_subset[ld$sector %in% ipm_sectors] = "Energy Sectors"
ld$sector_subset[ld$sector %in% "all"] = "All Sectors"
ld$sector_subset[!(ld$sector %in% ipm_sectors) & !(ld$sector %in% "all")] = "Rest of Economy"

```

order sector_subset

```

ld$sector_subset = factor(ld$sector_subset,
c("Energy Sectors","Rest of Economy","All Sectors"))

```

add sectoral labels

```
ld = relabel.sectors(ld)

ld_plot_pct <-
ggplot(subset(ld,sector_subset %in% "Energy Sectors"),
aes(x=forcats::fct_rev(label), y=-pct_change)) +
geom_bar(stat="identity", position=position_dodge(), colour="black", fill=sage.palette[1]) +
labs(y="% Change", x="Sector") +
scale_fill_manual(values=sage.palette) +
scale_y_continuous(expand=c(0,0), limits=c(-3.3,2.8), labels=scales::label_percent(accuracy=1, scale=1))
+
facet_grid(sector_subset~time, scales="free_y", space="free_y") +
theme_bw() +
coord_flip() +
geom_hline(yintercept=0) +
theme(text = element_text(color="black",family = "Inconsolata"),
axis.text.x = element_text(color="black"),
axis.text.y = element_text(color="black"),
axis.title.y = element_text(vjust = +3),
axis.title.x = element_text(vjust = -0.75),
panel.border = element_rect(colour = "black"),
axis.line = element_line(colour = "black"),
legend.position = "bottom")
ld_plot_pct
ggsave(file.path(output_dir,"ld_pct_energy.png"),width=8,height=4)
```

restrict labels to be 20 characters:

```
max_width = max(nchar(as.character(ld$label)))
ld$label = sprintf(paste0("%- ", max_width, "s"), ld$label)
ld$label = paste0(str_sub(ld$label,1,20),"\n",str_sub(ld$label,21,max_width))

ld_plot_pct <-
ggplot(subset(ld,sector_subset %in% "Rest of Economy"),
aes(x=reorder(label,pct_change), y=-pct_change)) +
geom_bar(stat="identity", position=position_dodge(), colour="black", fill=sage.palette[1]) +
labs(y="% Change", x="Sector") +
scale_fill_manual(values=sage.palette) +
scale_y_continuous(expand=c(0,0), limits=c(-.2,.18), labels=scales::label_percent(accuracy=0.1, scale=1))
+
facet_grid(sector_subset~time, scales="free_y", space="free_y") +
theme_bw() +
coord_flip() +
geom_hline(yintercept=0) +
theme(text = element_text(color="black",family = "Inconsolata"),
axis.text.x = element_text(color="black"),
axis.text.y = element_text(color="black", hjust=0),
axis.title.y = element_text(vjust = +3),
```

```

axis.title.x = element_text(vjust = -0.75),
panel.border = element_rect(colour = "black"),
axis.line = element_line(colour = "black"),
legend.position = "bottom")
ld_plot_pct
ggsave(file.path(output_dir,"ld_pct_roe.png"),width=8,height=8)

ld$time = as.character(ld$time)
ld$time = factor(ld$time,
c("2026","2031","2036","2041"))
ld_plot_pct <-
ggplot(subset(ld,sector_subset %in% "All Sectors"),
aes(x=time, y=-pct_change)) +
geom_bar(stat="identity", position=position_dodge(), colour="black", fill=sage.palette[1]) +
labs(y="% Change", x="Time") +
scale_fill_manual(values=sage.palette) +
scale_y_continuous(expand=c(0,0), limits=c(-.2,.2), labels=scales::label_percent(accuracy=0.01, scale=1),
breaks = scales::pretty_breaks(n = 10)) +
theme_bw() +
geom_hline(yintercept=0) +
theme(text = element_text(color="black",family = "Inconsolata"),
axis.text.x = element_text(color="black"),
axis.text.y = element_text(color="black"),
axis.title.y = element_text(vjust = +3),
axis.title.x = element_text(vjust = -0.75),
panel.border = element_rect(colour = "black"),
axis.line = element_line(colour = "black"),
legend.position = "bottom")
ld_plot_pct
ggsave(file.path(output_dir,"ld_pct_all_sectors.png"),width=6,height=4)

```

end
