residual\_emissions\_scenarios

William F. Lamb

5 12 2022

median\_and\_range <- function(data,grouping\_vars) {  
   
 data <- data %>%   
 group\_by\_at(grouping\_vars) %>%   
 summarise(median = signif(median(value),2),  
 percentile\_5th = signif(quantile(value, probs = c(0.05)),2),  
 percentile\_95th = signif(quantile(value, probs = c(0.95)),2)) %>%   
 mutate(value=paste0(median," [",percentile\_5th,"-",percentile\_95th,"]")) %>%   
 select(-median,-percentile\_5th,-percentile\_95th)  
   
}

## identify gases in the scenario data  
wd\_scenarios\_emissions <- data\_r1 %>%   
 ungroup() %>%   
 mutate(gas=ifelse(grepl("CO2",var),"CO2",NA)) %>%   
 mutate(gas=ifelse(grepl("CH4",var),"CH4",gas)) %>%   
 mutate(gas=ifelse(grepl("N2O",var),"N2O",gas)) %>%   
 mutate(gas=ifelse(grepl("F-Gases",var),"F-Gases",gas)) %>%   
 mutate(gas=ifelse(grepl("Kyoto Gases",var),"Kyoto Gases",gas)) %>%   
 select(-region)  
  
  
## join GWPs and calculate CO2e by sector  
wd\_scenarios\_emissions <- left\_join(wd\_scenarios\_emissions,cc\_gwps %>%   
 select(gas,gwp100\_ar6) %>%   
 mutate(gwp100\_ar6=ifelse(gas=="CH4",27,gwp100\_ar6)),by="gas")  
  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 mutate(gwp100\_ar6=ifelse(unit=="Mt CO2-equiv/yr",1,gwp100\_ar6)) %>%   
 mutate(value=ifelse(unit=="kt N2O/yr",value/1000,value)) %>% # N2O is in kt, not Mt   
 mutate(value\_gwp=value\*gwp100\_ar6) %>%   
 select(-value,-gwp100\_ar6)  
  
  
## create unique scenario IDs  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 mutate(id=paste0(model,"|",scenario)) %>%   
 select(id,everything()) %>%   
 arrange(id)  
  
  
## exclude scenarios based on whether they have semi-detailed sector data  
cc\_sectors <- read.xlsx("../Data/scenario\_sector\_mapping.xlsx",sheet=2)  
cc\_scenarios <- left\_join(wd\_scenarios\_emissions,cc\_sectors %>% select(var) %>% mutate(include=1),by="var")  
cc\_scenarios <- cc\_scenarios %>%   
 ungroup() %>%   
 select(-unit,-gas) %>%   
 filter(year==2050) %>%   
 filter(include==1)  
cc\_scenarios <- spread(cc\_scenarios,var,value\_gwp)  
# Manually fill Emissions|CO2|Other as this is important for resolving totals in some scenarios, but isn't universally available  
cc\_scenarios <- cc\_scenarios %>%   
 mutate(`Emissions|CO2|Other`=ifelse(is.na(`Emissions|CO2|Other`),0,`Emissions|CO2|Other`))  
cc\_scenarios <- cc\_scenarios %>%   
 na.omit() %>%   
 select(id)  
  
wd\_scenarios\_emissions <- left\_join(wd\_scenarios\_emissions,cc\_scenarios %>% mutate(include=1),by="id") %>%   
 filter(include==1) %>%   
 select(-include)  
  
  
## rejoin the list of emissions sectors  
wd\_scenarios\_emissions <- left\_join(wd\_scenarios\_emissions,cc\_sectors,by=join\_by(var,gas))  
  
  
# infill the Emissions|CO2|Other sector  
wd\_co2\_other\_fix <- wd\_scenarios\_emissions %>%   
 ungroup() %>%   
 filter(var=="Emissions|CO2") %>%   
 select(id,model,scenario,category,unit,year)  
wd\_co2\_other\_fix <- left\_join(wd\_co2\_other\_fix,wd\_scenarios\_emissions %>%   
 ungroup() %>%   
 filter(var=="Emissions|CO2|Other") %>%  
 select(id,model,scenario,category,unit,year,value\_gwp),  
 by = join\_by(id, model, scenario, category, unit, year))  
wd\_co2\_other\_fix <- wd\_co2\_other\_fix %>%   
 mutate(var="Emissions|CO2|Other") %>%   
 mutate(value\_gwp=ifelse(is.na(value\_gwp),0,value\_gwp)) %>%   
 mutate(gas="CO2") %>%   
 mutate(sector="Other") %>%   
 mutate(subsector="Total")  
  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 filter(var!="Emissions|CO2|Other")  
wd\_scenarios\_emissions <- rbind(wd\_scenarios\_emissions,wd\_co2\_other\_fix)  
  
blarg <- wd\_scenarios\_emissions %>% filter(id=="IMAGE 3.0|EN\_NPi2020\_1000") %>% filter(year==2050)  
  
## resolve sector and gas inconsistencies  
wd\_sector\_totals <- wd\_scenarios\_emissions %>%   
 ungroup() %>%   
 filter(subsector=="Total") %>%   
 group\_by(id,model,scenario,year,gas,sector) %>%   
 summarise(sector\_total=sum(value\_gwp))

## `summarise()` has grouped output by 'id', 'model', 'scenario', 'year', 'gas'.  
## You can override using the `.groups` argument.

wd\_gas\_totals <- wd\_scenarios\_emissions %>%   
 ungroup() %>%   
 filter(sector=="Total") %>%   
 group\_by(id,model,scenario,year,gas) %>%   
 summarise(gas\_total=sum(value\_gwp))

## `summarise()` has grouped output by 'id', 'model', 'scenario', 'year'. You can  
## override using the `.groups` argument.

wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 filter(!is.na(sector)) %>%   
 filter(!is.na(subsector)) %>%   
 filter(subsector!="Total") %>%   
 group\_by(id,model,scenario,category,gas,sector,subsector,year) %>%  
 summarise(value\_gwp=sum(value\_gwp,na.rm=TRUE))

## `summarise()` has grouped output by 'id', 'model', 'scenario', 'category',  
## 'gas', 'sector', 'subsector'. You can override using the `.groups` argument.

# sector inconsistencies  
wd\_scenarios\_emissions <- full\_join(wd\_scenarios\_emissions,wd\_sector\_totals,by = join\_by(id, model, scenario, year, gas,sector))  
  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 mutate(value\_gwp=ifelse(is.na(subsector),sector\_total,value\_gwp)) %>%   
 mutate(subsector=ifelse(is.na(subsector),"Other",subsector))  
  
residuals <- wd\_scenarios\_emissions %>%   
 group\_by(id,model,scenario,year,gas,sector) %>%  
 summarise(value\_gwp=sum(value\_gwp),sector\_total=first(sector\_total)) %>%   
 mutate(value\_gwp=sector\_total-value\_gwp) %>%   
 mutate(subsector="Other") %>%   
 select(-sector\_total) %>%   
 filter(value\_gwp>0)

## `summarise()` has grouped output by 'id', 'model', 'scenario', 'year', 'gas'.  
## You can override using the `.groups` argument.

wd\_scenarios\_emissions <- rbind(wd\_scenarios\_emissions,residuals)  
  
  
## gas inconsistencies  
wd\_scenarios\_emissions <- left\_join(wd\_scenarios\_emissions %>% select(-sector\_total),wd\_gas\_totals,by = join\_by(id, model, scenario, year, gas))  
  
residuals <- wd\_scenarios\_emissions %>%   
 group\_by(id,model,scenario,year,gas) %>%  
 summarise(value\_gwp=sum(value\_gwp),gas\_total=first(gas\_total))%>%   
 mutate(value\_gwp=gas\_total-value\_gwp) %>%   
 mutate(sector="Other") %>%   
 select(-gas\_total) %>%   
 filter(value\_gwp>0)

## `summarise()` has grouped output by 'id', 'model', 'scenario', 'year'. You can  
## override using the `.groups` argument.

wd\_scenarios\_emissions <- rbind(wd\_scenarios\_emissions,residuals)  
  
  
## tidy up naming conventions  
  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 mutate(var=ifelse(sector=="Other",paste0(gas,"|",sector),paste0(gas,"|",sector,"|",subsector))) %>%   
 mutate(var=ifelse(var=="F-Gases|F-Gases|Other","F-Gases",var)) %>%   
 mutate(var=ifelse(var=="CO2|AFOLU|Other","CO2|AFOLU",var)) %>%   
 mutate(var=ifelse(var=="N2O|AFOLU|Other","N2O|AFOLU",var)) %>%   
 mutate(var=ifelse(var=="CH4|Energy|Other","CH4|Energy",var)) %>%   
 mutate(var=ifelse(var=="CO2|Industrial processes|Other","CO2|Industrial processes",var)) %>%   
 mutate(var=ifelse(gas=="Kyoto Gases","Kyoto Gases",var)) %>%   
 mutate(subsector=ifelse(is.na(subsector),"Other",subsector))  
  
## save the updated list of sectors  
cc\_sectors <- wd\_scenarios\_emissions %>%   
 ungroup() %>%   
 select(gas,sector,subsector,var) %>%   
 distinct()  
  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 group\_by(id,model,scenario) %>%   
 mutate(category=first(category))  
  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 group\_by(id,model,scenario,category,gas,sector,subsector,var,year) %>%   
 summarise(value\_gwp=sum(value\_gwp,na.rm=TRUE))

## `summarise()` has grouped output by 'id', 'model', 'scenario', 'category',  
## 'gas', 'sector', 'subsector', 'var'. You can override using the `.groups`  
## argument.

## Check for major discrepancies between the reconstructed sectors and the Kyoto gas total  
  
# test\_residual <- wd\_scenarios\_emissions %>%   
# filter(year==2050) %>%   
# filter(gas!="Kyoto Gases") %>%   
# group\_by(id,year) %>%   
# summarise(value=sum(value,na.rm=TRUE))  
#   
# test\_residual <- left\_join(test\_residual,wd\_scenarios\_emissions %>% ungroup %>% filter(year==2050) %>% filter(gas=="Kyoto Gases") %>% select(id,kyoto\_gas\_total=value))  
#   
# test\_residual <- test\_residual %>%   
# mutate(difference=value-kyoto\_gas\_total) %>%   
# mutate(difference\_abs=abs(difference)) %>%   
# group\_by(year) %>%   
# summarise(median=signif(median(difference\_abs),2),  
# percentile\_5th = signif(quantile(difference\_abs, probs = c(0.05)),2),  
# percentile\_95th = signif(quantile(difference\_abs, probs = c(0.95)),2))  
#

## filter scenarios  
wd\_scenarios\_removals <- data\_r1 %>%   
 ungroup() %>%   
 mutate(id=paste0(model,"|",scenario)) %>%   
 select(id,everything(),-region,-unit) %>%   
 rename(value\_gwp=value) %>%   
 arrange(id)  
  
wd\_scenarios\_removals <- left\_join(wd\_scenarios\_removals,cc\_scenarios %>% mutate(include=1),by="id") %>%   
 filter(include==1) %>%   
 select(-include)  
  
  
## identify relevant sectors  
cc\_removals <- read.xlsx("../Data/scenario\_sector\_mapping.xlsx",sheet=3)  
wd\_scenarios\_removals <- left\_join(wd\_scenarios\_removals,cc\_removals %>% mutate(include=1),by="var")  
wd\_scenarios\_removals <- wd\_scenarios\_removals %>%   
 filter(include==1) %>%   
 select(-include)  
  
  
## subtract CCS|Biomass|industrial processes from CCS|Biomass, so that the former aren't double counted when allocated to the process sector  
wd\_scenarios\_removals <- spread(wd\_scenarios\_removals,var,value\_gwp)  
wd\_scenarios\_removals <- wd\_scenarios\_removals %>%  
 mutate(`Carbon Sequestration|CCS|Biomass|Industrial Processes`=ifelse(is.na(`Carbon Sequestration|CCS|Biomass|Industrial Processes`),0,`Carbon Sequestration|CCS|Biomass|Industrial Processes`)) %>%   
 mutate(`Carbon Sequestration|CCS|Biomass`=`Carbon Sequestration|CCS|Biomass`-`Carbon Sequestration|CCS|Biomass|Industrial Processes`)  
  
  
## merge the other removal categories  
wd\_scenarios\_removals <- wd\_scenarios\_removals %>%   
 rowwise() %>%   
 mutate(`Carbon Sequestration|Other`= sum(`Carbon Sequestration|Direct Air Capture`,  
 `Carbon Sequestration|Enhanced Weathering`,  
 `Carbon Sequestration|Feedstocks`,  
 `Carbon Sequestration|Other`,na.rm = TRUE))  
  
  
## update the scenario inclusion list to exclude scenarios without land use removals reporting   
cc\_scenarios <- anti\_join(cc\_scenarios,wd\_scenarios\_removals %>% filter(is.na(`Carbon Sequestration|Land Use`)) %>% select(id))

## Joining with `by = join\_by(id)`

## tidy up   
wd\_scenarios\_removals <- gather(wd\_scenarios\_removals,var,value\_gwp,`Carbon Sequestration|CCS|Biomass`:`Carbon Sequestration|Other`) %>%   
 filter(var %in% c("Carbon Sequestration|CCS|Biomass",  
 "Carbon Sequestration|CCS|Biomass|Industrial Processes",  
 "Carbon Sequestration|Land Use",  
 "Carbon Sequestration|Other"))  
  
  
## join data to main dataset  
wd\_scenarios\_emissions <- rbind(wd\_scenarios\_emissions,wd\_scenarios\_removals) %>%   
 arrange(id,year)  
  
  
## calculate gross emissions by sector  
wd\_scenarios\_emissions <- spread(wd\_scenarios\_emissions %>%   
 ungroup() %>%   
 select(id,model,scenario,category,year,var,value\_gwp),var,value\_gwp)  
  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 mutate(`CO2|Energy|Supply`=`CO2|Energy|Supply`+`Carbon Sequestration|CCS|Biomass`) %>%   
 mutate(`CO2|AFOLU`=`CO2|AFOLU`+`Carbon Sequestration|Land Use`) %>%   
 mutate(`CO2|Other`=`CO2|Other`+`Carbon Sequestration|Other`)  
  
wd\_scenarios\_emissions <- gather(wd\_scenarios\_emissions,var,value,-id,-model,-scenario,-category,-year)  
wd\_scenarios\_emissions <- left\_join(wd\_scenarios\_emissions,cc\_sectors,by="var")  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 select(id,model,scenario,category,gas,sector,subsector,var,year,value)  
  
  
  
## source the net zero years from scenario metadata  
cc\_netzero <- read.xlsx("../Data/AR6\_Scenarios\_Database\_metadata\_indicators\_v1.1\_1.xlsx",sheet=2) %>%   
 mutate(id=paste0(Model,"|",Scenario)) %>%   
 select(id,net\_zero\_CO2=`Year.of.netzero.CO2.emissions.(Harm-Infilled).Table.SPM2`,  
 net\_zero\_GHG=`Year.of.netzero.GHG.emissions.(Harm-Infilled).Table.SPM2`)  
cc\_netzero <- left\_join(cc\_netzero,cc\_scenarios %>% mutate(include=1),by="id") %>%   
 filter(include==1) %>%   
 select(-include)  
  
wd\_scenarios\_emissions <- left\_join(wd\_scenarios\_emissions,cc\_netzero,by="id")  
  
  
## exclude the scenarios that don't reach net zero by 2100 and tidy up  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 filter(!is.na(net\_zero\_CO2)) %>%   
 arrange(id,year) %>%   
 mutate(value=value/1000)  
  
no\_scenarios <- wd\_scenarios\_emissions %>%   
 select(id) %>%  
 distinct()  
  
  
## simplify the model names  
  
wd\_scenarios\_emissions <- wd\_scenarios\_emissions %>%   
 mutate(model=ifelse(grepl("IMAGE",model),"IMAGE",model)) %>%   
 mutate(model=ifelse(grepl("MESSAGE",model),"MESSAGE",model)) %>%   
 mutate(model=ifelse(grepl("REMIND",model),"REMIND",model))  
  
  
wd\_scenarios\_emissions$category <- as.factor(wd\_scenarios\_emissions$category)  
wd\_scenarios\_emissions$category <- fct\_relevel(wd\_scenarios\_emissions$category,"C3","C2","C1")  
  
  
wd\_scenarios\_emissions$gas <- as.factor(wd\_scenarios\_emissions$gas)  
wd\_scenarios\_emissions$gas <- fct\_relevel(wd\_scenarios\_emissions$gas,"CO2","CH4","N2O","F-Gases")  
  
  
wd\_scenarios\_emissions$sector <- as.factor(wd\_scenarios\_emissions$sector)  
wd\_scenarios\_emissions$sector <- fct\_relevel(wd\_scenarios\_emissions$sector,"Energy","AFOLU","Industrial processes","F-Gases","Other")

## any scenarios where final gross emissions by sector are still negative?  
  
check <- wd\_scenarios\_emissions %>%  
 # mutate(include=ifelse(year==net\_zero\_CO2,1,0)) %>%   
 # filter(include==1) %>%   
 filter(year==2100) %>%   
 filter(var=="CO2|AFOLU") %>%   
 filter(value<0) %>%   
 select(id,model,var,value) %>%   
 mutate(group="blarg")  
  
check <- median\_and\_range(check,"group")

colors=c("#e6550d","#3182bd","#31a354")  
  
## gross CO2 emissions at net zero CO2  
wd\_totals\_co2 <- wd\_scenarios\_emissions %>%   
 filter(var!="Kyoto Gases") %>%   
 filter(!grepl("Carbon Sequestration",var)) %>%   
 filter(gas=="CO2") %>%   
 mutate(value=ifelse(is.na(value),0,value)) %>%   
 mutate(include=ifelse(net\_zero\_CO2==year,1,0)) %>%   
 filter(include==1) %>%   
 group\_by(id,category,year) %>%   
 summarise(value=sum(value)) %>%   
 mutate(var="CO2")

## `summarise()` has grouped output by 'id', 'category'. You can override using  
## the `.groups` argument.

## gross GHG emissions at net zero CO2  
wd\_totals\_ghg <- wd\_scenarios\_emissions %>%   
 filter(var!="Kyoto Gases") %>%   
 filter(!grepl("Carbon Sequestration",var)) %>%   
 mutate(value=ifelse(is.na(value),0,value)) %>%   
 mutate(include=ifelse(net\_zero\_CO2==year,1,0)) %>%   
 filter(include==1) %>%   
 group\_by(id,category,year) %>%   
 summarise(value=sum(value)) %>%   
 mutate(var="GHG")

## `summarise()` has grouped output by 'id', 'category'. You can override using  
## the `.groups` argument.

## join  
wd\_totals <- rbind(wd\_totals\_co2,wd\_totals\_ghg) %>%   
 select(id,category,year,var,value) %>%   
 arrange(var,desc(category))  
  
  
# labels  
wd\_totals <- wd\_totals %>%   
 mutate(category=as.character(category)) %>%   
 mutate(category=ifelse(grepl("C1",category),"C1: Below 1.5°C",category)) %>%   
 mutate(category=ifelse(grepl("C2",category),"C2: Below 1.5°C with overshoot",category)) %>%   
 mutate(category=ifelse(grepl("C3",category),"C3: Below 2°C",category)) %>%   
 mutate(category=as.factor(category))  
  
wd\_totals$category <- fct\_relevel(wd\_totals$category,  
 "C3: Below 2°C",  
 "C2: Below 1.5°C with overshoot",  
 "C1: Below 1.5°C")   
  
  
## plot CO2  
p1 <- wd\_totals %>%   
 filter(var=="CO2") %>%   
 ggplot(.,aes(x=value,fill=category)) +  
 geom\_density(alpha=0.3,color='#636363') +  
 scale\_x\_continuous(limits=c(0,30),breaks=c(0,5,10,15,20,25,30)) +  
 scale\_fill\_manual(values=colors) +  
 theme(axis.title = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text = element\_blank(),  
 panel.border = element\_rect(color='#636363',fill=NA),  
 panel.background = element\_blank(),  
 panel.grid.major.y = element\_blank(),  
 panel.grid.major.x = element\_line(color='#f0f0f0'),  
 legend.position = "none",  
 plot.background = element\_blank(),  
 plot.margin = margin(0,0,0,0,"cm"),  
 plot.title = element\_text(color='#636363',size = 14,face = 'bold'),  
 plot.subtitle = element\_text(color='#636363')) +  
 labs(title=bquote(bold("a. Residual"~CO[2]\*" emissions at net zero")),  
 subtitle=bquote("Gt"~CO[2]\*"e/yr"))

## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.

p2 <- wd\_totals %>%   
 filter(var=="CO2") %>%   
 ggplot(.,aes(x=value,y=category,fill=category)) +  
 geom\_point(shape=21,size=3,alpha=0.4,color="#636363") +  
 scale\_y\_discrete(labels = function(x) str\_wrap(x, width = 17)) +  
 scale\_fill\_manual(values=colors) +  
 theme\_wl() +  
 theme(legend.position="none",  
 axis.title = element\_blank(),  
 plot.margin = margin(0,0,0,0,"cm")) +  
 scale\_x\_continuous(limits=c(0,30),breaks=c(0,5,10,15,20,25,30))

## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.

## Warning: The `size` argument of `element\_line()` is deprecated as of ggplot2 3.4.0.  
## ℹ Please use the `linewidth` argument instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: The `size` argument of `element\_rect()` is deprecated as of ggplot2 3.4.0.  
## ℹ Please use the `linewidth` argument instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

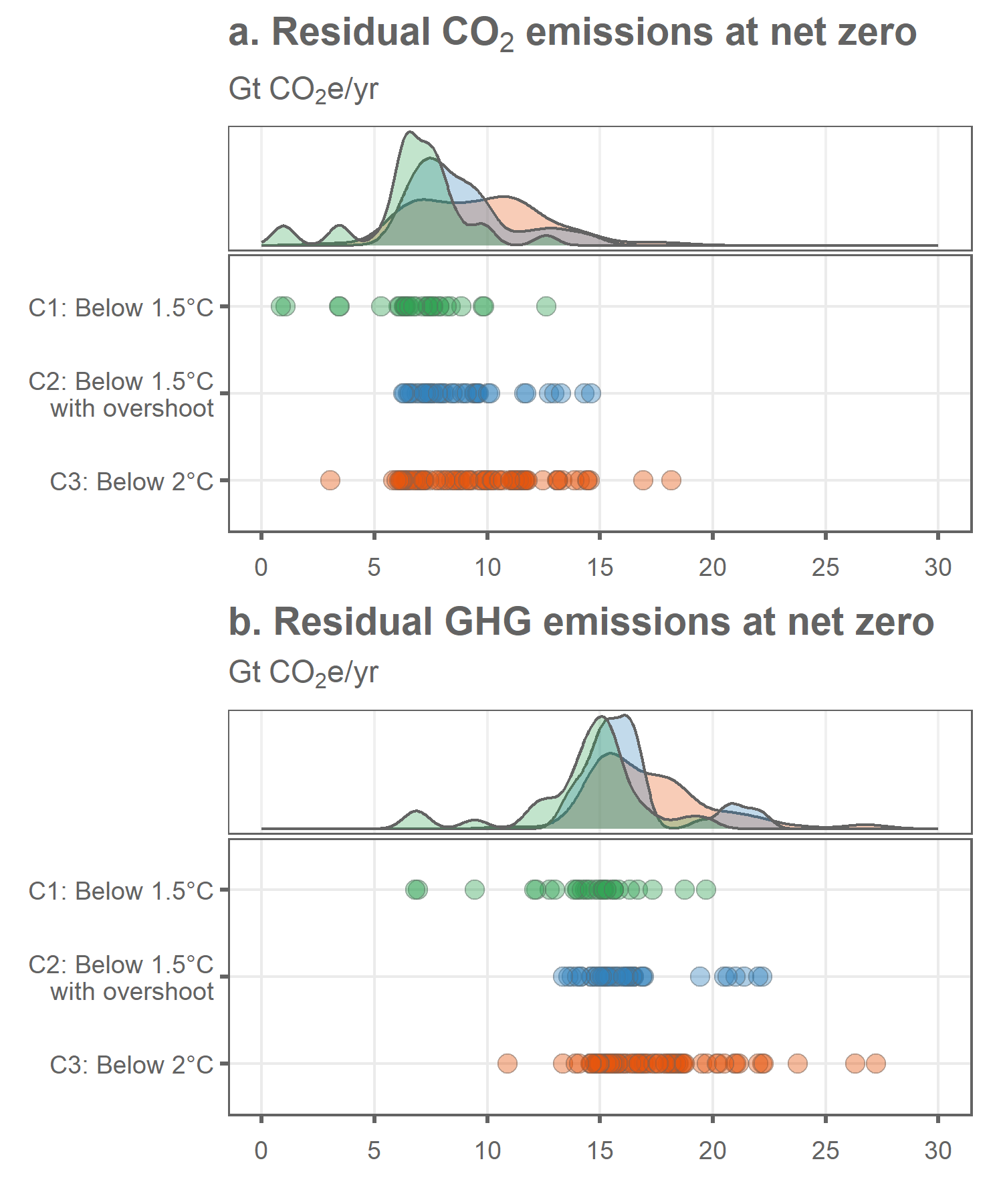
## plot GHG  
p3 <- wd\_totals %>%   
 filter(var=="GHG") %>%   
 ggplot(.,aes(x=value,fill=category)) +  
 geom\_density(alpha=0.3,color='#636363') +  
 scale\_x\_continuous(limits=c(0,30),breaks=c(0,5,10,15,20,25,30)) +  
 scale\_fill\_manual(values=colors) +  
 theme(axis.title = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text = element\_blank(),  
 panel.border = element\_rect(color='#636363',fill=NA),  
 panel.background = element\_blank(),  
 panel.grid.major.y = element\_blank(),  
 panel.grid.major.x = element\_line(color='#f0f0f0'),  
 legend.position = "none",  
 plot.background = element\_blank(),  
 plot.margin = margin(0,0,0,0,"cm"),  
 plot.title = element\_text(color='#636363',size = 14,face = 'bold'),  
 plot.subtitle = element\_text(color='#636363')) +  
 labs(title=bquote(bold("b. Residual GHG emissions at net zero")),  
 subtitle=bquote("Gt"~CO[2]\*"e/yr"))

## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.

p4 <- wd\_totals %>%   
 filter(var=="GHG") %>%   
 ggplot(.,aes(x=value,y=category,fill=category)) +  
 geom\_point(shape=21,size=3,alpha=0.4,color="#636363") +  
 scale\_y\_discrete(labels = function(x) str\_wrap(x, width = 17)) +  
 scale\_fill\_manual(values=colors) +  
 theme\_wl() +  
 theme(legend.position="none",  
 axis.title = element\_blank(),  
 plot.margin = margin(0,0,0,0,"cm")) +  
 scale\_x\_continuous(limits=c(0,30),breaks=c(0,5,10,15,20,25,30))

## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.

p1/plot\_spacer()/p2/p3/plot\_spacer()/p4 + plot\_layout(heights=c(1.8,-0.5,4,1.8,-0.5,4))



## save plot data  
addWorksheet(wb\_figures,"fig\_1\_total\_residual")  
writeData(wb\_figures, sheet = "fig\_1\_total\_residual",  
 wd\_totals, colNames = T, rowNames = F)

## summary of scenario totals  
wd\_totals\_summary <- median\_and\_range(wd\_totals,c("var","category"))

## `summarise()` has grouped output by 'var'. You can override using the `.groups`  
## argument.

wd\_totals\_summary <- left\_join(wd\_totals\_summary,wd\_totals %>%   
 group\_by(var,category) %>%   
 summarise(n=n(),  
 year\_median = signif(median(year),4),  
 year\_5th = signif(quantile(year, probs = c(0.05)),4),  
 year\_95th = signif(quantile(year, probs = c(0.95)),4)) %>%   
 mutate(year=paste0(year\_median," [",year\_5th,"-",year\_95th,"]")) %>%   
 select(var,category,year,n),by = join\_by(var, category))

## `summarise()` has grouped output by 'var'. You can override using the `.groups`  
## argument.

wd\_totals\_summary <- rbind(wd\_totals\_summary,  
 left\_join(median\_and\_range(wd\_totals,"var"),wd\_totals %>%   
 group\_by(var) %>%   
 summarise(n=n(),year\_median = signif(median(year),4),  
 year\_5th = signif(quantile(year, probs = c(0.05)),4),  
 year\_95th = signif(quantile(year, probs = c(0.95)),4)) %>%   
 mutate(year=paste0(year\_median," [",year\_5th,"-",year\_95th,"]")) %>%   
 select(var,year,n),by = join\_by(var)) %>%   
 mutate(category="all"))  
  
  
## calculate reductions and join  
wd\_total\_reductions <- wd\_scenarios\_emissions %>%   
 mutate(include=ifelse(net\_zero\_CO2==year,1,0)) %>%   
 mutate(include=ifelse(year==2020,1,include)) %>%   
 filter(include==1) %>%   
 filter(gas=="CO2") %>%   
 group\_by(id,gas,category,year) %>%   
 summarise(value=sum(value,na.rm=TRUE))

## `summarise()` has grouped output by 'id', 'gas', 'category'. You can override  
## using the `.groups` argument.

wd\_total\_reductions <- rbind(wd\_total\_reductions,wd\_scenarios\_emissions %>%   
 mutate(include=ifelse(net\_zero\_CO2==year,1,0)) %>%   
 mutate(include=ifelse(year==2020,1,include)) %>%   
 filter(include==1) %>%   
 filter(var!="Kyoto Gases") %>%   
 filter(!grepl("Carbon Sequestration",var)) %>%   
 group\_by(id,category,year) %>%   
 summarise(value=sum(value,na.rm=TRUE)) %>%   
 mutate(gas="GHG"))

## `summarise()` has grouped output by 'id', 'category'. You can override using  
## the `.groups` argument.

wd\_total\_reductions <- wd\_total\_reductions %>%   
 group\_by(id,gas,category) %>%   
 summarise(rel\_change=(first(value)-last(value))/first(value)) %>%   
 mutate(rel\_change=rel\_change\*100)

## `summarise()` has grouped output by 'id', 'gas'. You can override using the  
## `.groups` argument.

wd\_total\_reductions\_summary <- median\_and\_range(wd\_total\_reductions %>% mutate(value=rel\_change),c("gas","category"))

## `summarise()` has grouped output by 'gas'. You can override using the `.groups`  
## argument.

wd\_total\_reductions\_summary <- rbind(wd\_total\_reductions\_summary,  
 median\_and\_range(wd\_total\_reductions %>% mutate(value=rel\_change),c("gas")) %>%   
 mutate(category="all"))  
  
wd\_total\_reductions\_summary <- wd\_total\_reductions\_summary %>%   
 mutate(category=as.character(category)) %>%   
 mutate(category=ifelse(grepl("C1",category),"C1: Below 1.5°C",category)) %>%   
 mutate(category=ifelse(grepl("C2",category),"C2: Below 1.5°C with overshoot",category)) %>%   
 mutate(category=ifelse(grepl("C3",category),"C3: Below 2°C",category)) %>%   
 mutate(category=as.factor(category))  
  
wd\_totals\_summary <- left\_join(wd\_totals\_summary %>% rename(residual=value,gas=var),  
 wd\_total\_reductions\_summary %>% rename(reductions=value),  
 by = join\_by(gas,category))  
  
  
wd\_totals\_summary <- gather(wd\_totals\_summary,var,value,residual,reductions)  
wd\_totals\_summary <- wd\_totals\_summary %>%   
 ungroup() %>%   
 mutate(var=paste0(gas,"\_",var)) %>%   
 select(-gas)  
wd\_totals\_summary <- spread(wd\_totals\_summary,var,value)  
wd\_totals\_summary <- wd\_totals\_summary %>%   
 select(category,n,year,CO2\_residual,CO2\_reductions,GHG\_residual,GHG\_reductions)  
  
addWorksheet(wb\_analysis,"total\_residual\_at\_nz")  
writeData(wb\_analysis, sheet = "total\_residual\_at\_nz",  
 wd\_totals\_summary, colNames = T, rowNames = F)  
  
  
  
## how many scenarios were excluded??  
total\_scenarios <- data\_r1 %>%   
 ungroup() %>%   
 select(model,scenario,category) %>%   
 distinct() %>%   
 group\_by(category) %>%   
 summarise(n=n())

blarg <- wd\_scenarios\_emissions %>%   
 filter(year %in% c(2020,2100)) %>%   
 filter(var!="Kyoto Gases") %>%   
 filter(!grepl("Carbon Sequestration",var)) %>%   
 group\_by(id,category,year) %>%   
 summarise(value=sum(value,na.rm=TRUE)) %>%   
 mutate(gas="GHG")

## `summarise()` has grouped output by 'id', 'category'. You can override using  
## the `.groups` argument.

blarg <- blarg %>%   
 group\_by(id,gas,category) %>%   
 summarise(level=last(value),  
 rel\_change=(first(value)-last(value))/first(value)) %>%   
 mutate(rel\_change=rel\_change\*100)

## `summarise()` has grouped output by 'id', 'gas'. You can override using the  
## `.groups` argument.

blarg <- median\_and\_range(blarg %>% mutate(value=rel\_change),c("gas","category"))

## `summarise()` has grouped output by 'gas'. You can override using the `.groups`  
## argument.

## emissions by gas at NZ  
wd\_gases <- wd\_scenarios\_emissions %>%   
 filter(var!="Kyoto Gases") %>%   
 filter(!grepl("Carbon Sequestration",var)) %>%   
 mutate(value=ifelse(is.na(value),0,value)) %>%   
 mutate(include=ifelse(net\_zero\_CO2==year,1,0)) %>%   
 filter(include==1) %>%   
 group\_by(id,model,scenario,category,year,gas) %>%   
 summarise(value=sum(value,na.rm=TRUE))

## `summarise()` has grouped output by 'id', 'model', 'scenario', 'category',  
## 'year'. You can override using the `.groups` argument.

## emissions by sector at NZ  
wd\_sectors <- wd\_scenarios\_emissions %>%   
 filter(var!="Kyoto Gases") %>%   
 filter(!grepl("Carbon Sequestration",var)) %>%   
 mutate(value=ifelse(is.na(value),0,value)) %>%   
 mutate(include=ifelse(net\_zero\_CO2==year,1,0)) %>%   
 filter(include==1) %>%   
 mutate(sector=as.character(sector)) %>%   
 mutate(sector=ifelse(sector %in% c("Industrial processes","F-Gases","Other"),"Other (Industrial processes, F-gases, other)",sector)) %>%   
 group\_by(id,model,scenario,category,year,sector) %>%   
 summarise(value=sum(value,na.rm=TRUE))

## `summarise()` has grouped output by 'id', 'model', 'scenario', 'category',  
## 'year'. You can override using the `.groups` argument.

## energy by fuel type at NZ  
wd\_fuels <- data\_r1 %>%   
 ungroup() %>%   
 mutate(id=paste0(model,"|",scenario)) %>%   
 select(-region,-unit) %>%   
 filter(var %in% c("Primary Energy",  
 "Primary Energy|Fossil|w/ CCS",  
 "Primary Energy|Coal|w/o CCS",  
 "Primary Energy|Gas|w/o CCS",  
 "Primary Energy|Oil|w/o CCS"))  
  
wd\_fuels <- left\_join(wd\_fuels,cc\_netzero,by="id") %>%   
 mutate(include=ifelse(net\_zero\_CO2==year,1,0)) %>%   
 filter(include==1) %>%   
 select(-include)  
  
wd\_fuels <- spread(wd\_fuels,var,value)  
wd\_fuels <- wd\_fuels %>%   
 mutate(`Primary Energy|Renewable & nuclear`=`Primary Energy`-  
 `Primary Energy|Fossil|w/ CCS`-  
 `Primary Energy|Coal|w/o CCS` -  
 `Primary Energy|Gas|w/o CCS` -  
 `Primary Energy|Oil|w/o CCS`)  
  
wd\_fuels <- gather(wd\_fuels,var,value,`Primary Energy`:`Primary Energy|Renewable & nuclear`) %>%   
 filter(var!="Primary Energy")  
  
wd\_fuels <- left\_join(wd\_fuels,data.frame(var=c("Primary Energy|Renewable & nuclear",  
 "Primary Energy|Fossil|w/ CCS",  
 "Primary Energy|Coal|w/o CCS",  
 "Primary Energy|Gas|w/o CCS",  
 "Primary Energy|Oil|w/o CCS"),  
 fuel=c("Renewables, biomass & nuclear",  
 "Fossil with CCS",  
 "Coal",  
 "Gas",  
 "Oil")),by="var")  
  
wd\_fuels$fuel <- as.factor(wd\_fuels$fuel)  
wd\_fuels$fuel <- fct\_relevel(wd\_fuels$fuel,"Renewables, biomass & nuclear","Fossil with CCS","Oil","Gas","Coal")  
  
wd\_fuels$category <- as.factor(wd\_fuels$category)  
wd\_fuels$category <- fct\_relevel(wd\_fuels$category,"C3","C2","C1")  
  
wd\_fuels <- wd\_fuels %>%   
 mutate(model=ifelse(grepl("IMAGE",model),"IMAGE",model)) %>%   
 mutate(model=ifelse(grepl("REMIND",model),"REMIND",model)) %>%   
 mutate(model=ifelse(grepl("MESSAGE",model),"MESSAGE",model))  
  
  
  
## data summaries  
## at this point we ignore C1-C3 really  
wd\_gases\_summary <- median\_and\_range(wd\_gases,c("model","gas"))

## `summarise()` has grouped output by 'model'. You can override using the  
## `.groups` argument.

wd\_gases\_summary <- rbind(wd\_gases\_summary,  
 median\_and\_range(wd\_gases,"gas") %>% mutate(model="all"))  
wd\_gases\_summary <- spread(wd\_gases\_summary,gas,value) %>%   
 arrange(model)  
  
wd\_sectors\_summary <- median\_and\_range(wd\_sectors,c("model","sector"))

## `summarise()` has grouped output by 'model'. You can override using the  
## `.groups` argument.

wd\_sectors\_summary <- rbind(wd\_sectors\_summary,  
 median\_and\_range(wd\_sectors,"sector") %>% mutate(model="all"))  
wd\_sectors\_summary <- spread(wd\_sectors\_summary,sector,value) %>%   
 arrange(model)  
  
wd\_fuels\_extra\_vars <- spread(wd\_fuels %>% select(-var),fuel,value) %>%   
 mutate(`Fossil total`=Oil+Coal+Gas) %>%   
 mutate(Total=`Fossil total`+`Renewables, biomass & nuclear`+`Fossil with CCS`)  
wd\_fuels\_extra\_vars <- gather(wd\_fuels\_extra\_vars,fuel,value,`Renewables, biomass & nuclear`:Total)  
  
wd\_fuels\_summary <- median\_and\_range(wd\_fuels\_extra\_vars,c("model","fuel"))

## `summarise()` has grouped output by 'model'. You can override using the  
## `.groups` argument.

wd\_fuels\_summary <- rbind(wd\_fuels\_summary,  
 median\_and\_range(wd\_fuels\_extra\_vars,"fuel") %>% mutate(model="all"))  
wd\_fuels\_summary$fuel <- as.factor(wd\_fuels\_summary$fuel)  
wd\_fuels\_summary$fuel <- fct\_relevel(wd\_fuels\_summary$fuel,"Coal","Oil","Gas","Fossil total","Fossil with CCS","Renewables, biomass & nuclear","Total")  
wd\_fuels\_summary <- spread(wd\_fuels\_summary,fuel,value) %>%   
 arrange(model)  
  
addWorksheet(wb\_analysis,"gas\_residual\_at\_nz")  
addWorksheet(wb\_analysis,"sector\_residual\_at\_nz")  
addWorksheet(wb\_analysis,"fuel\_at\_nz")  
writeData(wb\_analysis, sheet = "gas\_residual\_at\_nz",  
 wd\_gases\_summary, colNames = T, rowNames = F)  
writeData(wb\_analysis, sheet = "sector\_residual\_at\_nz",  
 wd\_sectors\_summary, colNames = T, rowNames = F)  
writeData(wb\_analysis, sheet = "fuel\_at\_nz",  
 wd\_fuels\_summary, colNames = T, rowNames = F)

## what about primary energy supply if we compare across the same year??  
  
wd\_fuels\_check <- data\_r1 %>%   
 ungroup() %>%   
 mutate(id=paste0(model,"|",scenario)) %>%   
 select(-region,-unit) %>%   
 filter(var %in% c("Primary Energy",  
 "Primary Energy|Fossil|w/ CCS",  
 "Primary Energy|Coal|w/o CCS",  
 "Primary Energy|Gas|w/o CCS",  
 "Primary Energy|Oil|w/o CCS"))  
  
wd\_fuels\_check <- left\_join(wd\_fuels\_check,cc\_scenarios %>% mutate(include=1)) %>%   
 filter(year %in% c(2050,2075,2100)) %>%   
 filter(include==1) %>%   
 select(-include)

## Joining with `by = join\_by(id)`

wd\_fuels\_check <- spread(wd\_fuels\_check,var,value)  
wd\_fuels\_check <- wd\_fuels\_check %>%   
 mutate(`Primary Energy|Renewable & nuclear`=`Primary Energy`-  
 `Primary Energy|Fossil|w/ CCS`-  
 `Primary Energy|Coal|w/o CCS` -  
 `Primary Energy|Gas|w/o CCS` -  
 `Primary Energy|Oil|w/o CCS`)  
  
wd\_fuels\_check <- gather(wd\_fuels\_check,var,value,`Primary Energy`:`Primary Energy|Renewable & nuclear`) %>%   
 filter(var!="Primary Energy")  
  
wd\_fuels\_check <- left\_join(wd\_fuels\_check,data.frame(var=c("Primary Energy|Renewable & nuclear",  
 "Primary Energy|Fossil|w/ CCS",  
 "Primary Energy|Coal|w/o CCS",  
 "Primary Energy|Gas|w/o CCS",  
 "Primary Energy|Oil|w/o CCS"),  
 fuel=c("Renewables, biomass & nuclear",  
 "Fossil with CCS",  
 "Coal",  
 "Gas",  
 "Oil")),by="var")  
  
wd\_fuels\_check <- wd\_fuels\_check %>%   
 group\_by(id,year) %>%   
 summarise(value=sum(value))

## `summarise()` has grouped output by 'id'. You can override using the `.groups`  
## argument.

wd\_fuels\_check <- median\_and\_range(wd\_fuels\_check,c("year"))

p1 <- wd\_gases %>% ggplot(.,aes(x=id,y=value,fill=gas)) +  
 geom\_bar(stat="identity",color="#969696") +  
 facet\_nested(.~model + fct\_rev(category),scales="free\_x",space="free",switch = "x",  
 nest\_line = element\_line(linetype = 1,color='#636363')) +  
 theme\_wl() +  
 scale\_y\_continuous(limits=c(0,25),breaks=c(0,5,10,15,20,25)) +  
 #scale\_fill\_manual(values=colors[1:4]) +  
 scale\_fill\_brewer(palette="Greens") +  
 theme(axis.title = element\_blank(),  
 axis.text.x = element\_blank(),  
 axis.ticks.x = element\_blank(),  
 panel.grid.major.x = element\_blank(),  
 legend.position="bottom",  
 legend.title=element\_blank(),  
 legend.background = element\_blank(),  
 legend.margin = margin(t=-5),  
 strip.background = element\_blank(),  
 panel.spacing.x=unit(0.4, "lines"),  
 panel.border = element\_blank(),  
 strip.text = element\_text(color='#636363'),) +  
 labs(title="a. Residual GHG emissions at net zero by gas",  
 subtitle=bquote("Gt"~CO[2]\*"e/yr"))

## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.

wd\_sectors$sector <- as.factor(wd\_sectors$sector)  
wd\_sectors$sector <- fct\_relevel(wd\_sectors$sector,"Energy","AFOLU","Other (Industrial processes, F-gases, other)")  
  
p2 <- wd\_sectors %>% ggplot(.,aes(x=id,y=value,fill=sector)) +  
 geom\_bar(stat="identity",color="#969696") +  
 facet\_nested(.~model + fct\_rev(category),scales="free\_x",space="free",switch = "x",  
 nest\_line = element\_line(linetype = 1,color='#636363')) +  
 theme\_wl() +  
 scale\_y\_continuous(limits=c(0,25),breaks=c(0,5,10,15,20,25)) +  
 scale\_fill\_brewer(palette="Blues") +  
 theme(axis.title = element\_blank(),  
 axis.text.x = element\_blank(),  
 axis.ticks.x = element\_blank(),  
 panel.grid.major.x = element\_blank(),  
 legend.position="bottom",  
 legend.title=element\_blank(),  
 legend.background = element\_blank(),  
 legend.margin = margin(t=-5),  
 strip.background = element\_blank(),  
 panel.spacing.x=unit(0.4, "lines"),  
 panel.border = element\_blank(),  
 strip.text = element\_text(color='#636363'),  
 strip.clip = element\_blank()  
 ) +  
 labs(title="b. Residual GHG emissions at net zero by sector",  
 subtitle=bquote("Gt"~CO[2]\*"e/yr"))

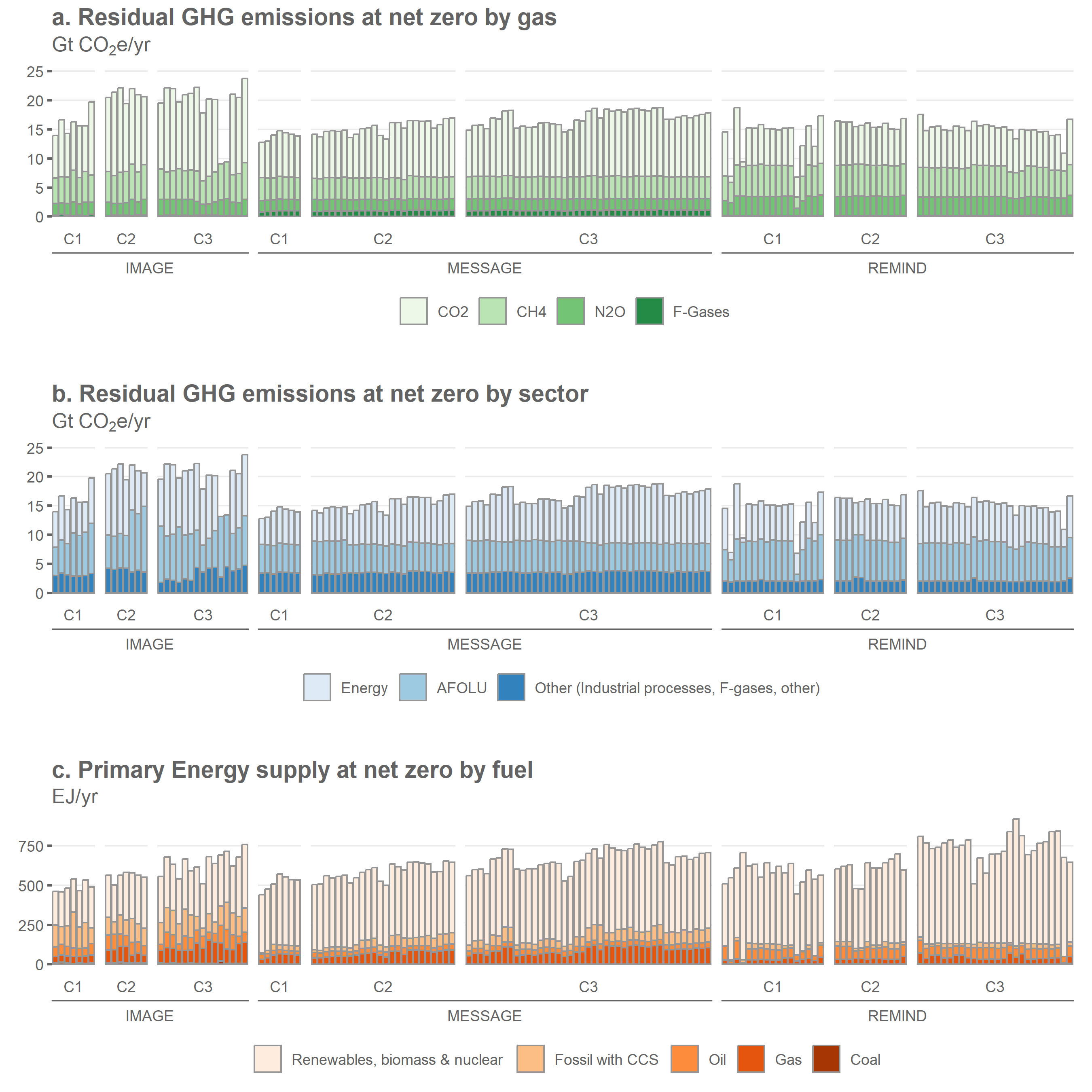
## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.

p3 <- wd\_fuels %>% ggplot(.,aes(x=id,y=value,fill=fuel)) +  
 geom\_bar(stat="identity",color="#969696") +  
 facet\_nested(.~model + fct\_rev(category),scales="free\_x",space="free",switch = "x",  
 nest\_line = element\_line(linetype = 1,color='#636363')) +  
 theme\_wl() +  
 scale\_fill\_brewer(palette="Oranges") +  
 theme(axis.title = element\_blank(),  
 axis.text.x = element\_blank(),  
 axis.ticks.x = element\_blank(),  
 panel.grid.major.x = element\_blank(),  
 legend.position="bottom",  
 legend.title=element\_blank(),  
 legend.background = element\_blank(),  
 legend.margin = margin(t=-5),  
 strip.background = element\_blank(),  
 panel.spacing.x=unit(0.4, "lines"),  
 panel.border = element\_blank(),  
 strip.text = element\_text(color='#636363'),  
 strip.clip = element\_blank()  
 ) +  
 labs(title="c. Primary Energy supply at net zero by fuel",  
 subtitle=bquote("EJ/yr"))

## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.

p1/plot\_spacer()/p2/plot\_spacer()/p3 + plot\_layout(heights=c(5,1,5,1,5))

## Warning: Removed 2 rows containing missing values (`geom\_bar()`).  
## Removed 2 rows containing missing values (`geom\_bar()`).



wd\_pca <- wd\_scenarios\_emissions %>%  
 filter(var!="Kyoto Gases") %>%  
 filter(!grepl("Carbon Sequestration",var)) %>%  
 mutate(value=ifelse(is.na(value),0,value)) %>%  
 mutate(include=ifelse(net\_zero\_CO2==year,1,0)) %>%   
 filter(include==1) %>%  
 group\_by(id,model,var) %>%  
 summarise(value=sum(value,na.rm=TRUE))

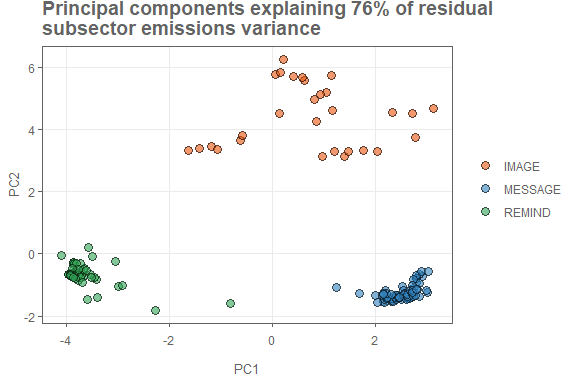
## `summarise()` has grouped output by 'id', 'model'. You can override using the  
## `.groups` argument.

wd\_pca <- spread(wd\_pca,var,value)  
pca <- prcomp(wd\_pca %>% ungroup() %>% select(-id,-model),scale=TRUE)  
wd\_pca <- cbind(wd\_pca %>% select(id,model),pca$x)  
pca\_variance <-get\_eigenvalue(pca)$variance.percent  
sum(pca\_variance[1:2])

## [1] 75.74677

colors=c("#e6550d","#3182bd","#31a354")  
wd\_pca %>% ggplot(.,aes(x=PC1,y=PC2,fill=model)) +  
 geom\_point(shape=21,size=3,alpha=0.6) +  
 theme\_wl() +  
 scale\_fill\_manual(values=colors) +  
 theme(legend.title=element\_blank()) +  
 labs(title=paste0("Principal components explaining ",signif(sum(pca\_variance[1:2]),2), "% of residual\nsubsector emissions variance"))

## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.



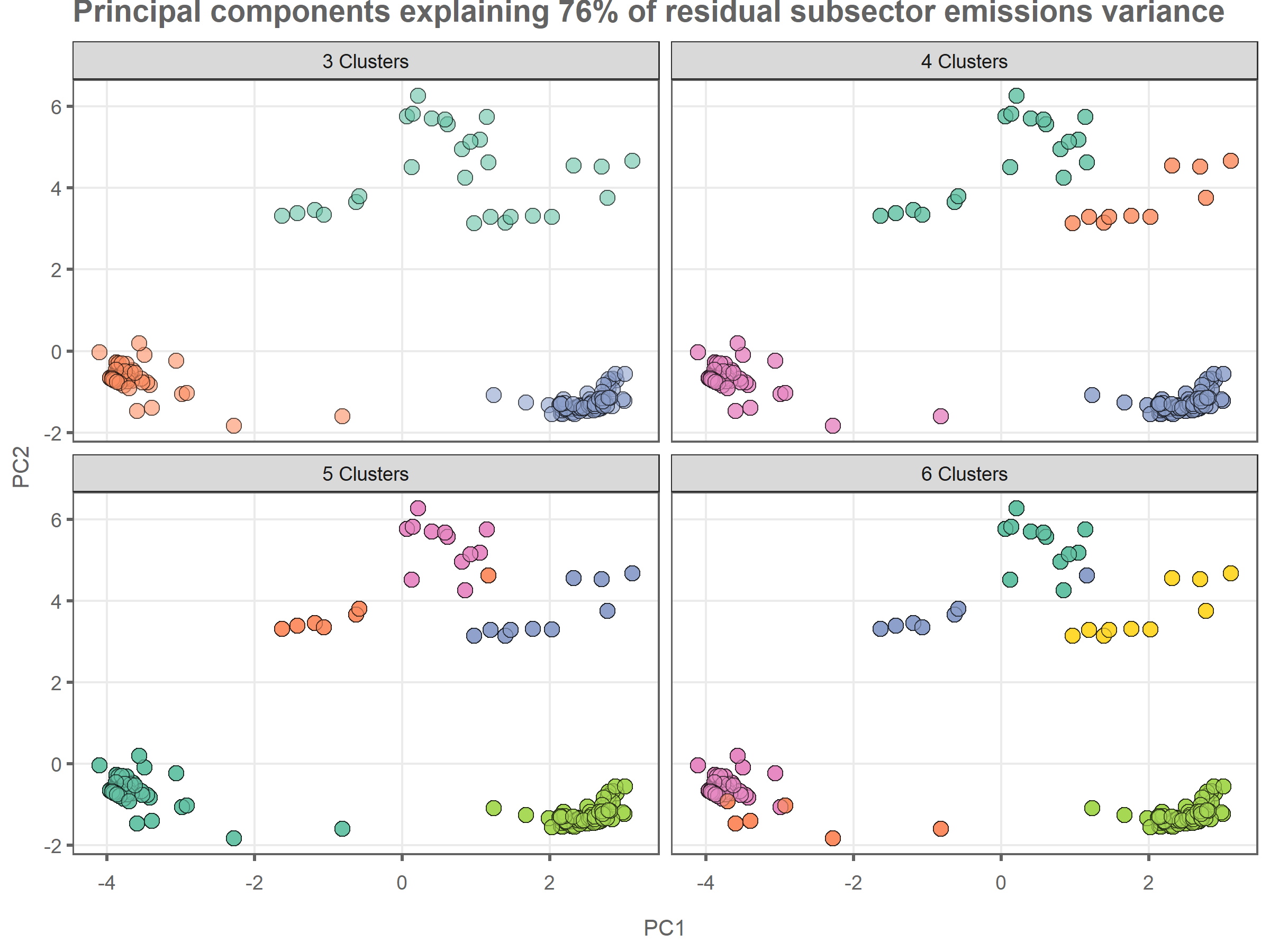
wd\_clusters <- wd\_scenarios\_emissions %>%  
 filter(var!="Kyoto Gases") %>%  
 filter(!grepl("Carbon Sequestration",var)) %>%  
 mutate(value=ifelse(is.na(value),0,value)) %>%  
 mutate(include=ifelse(net\_zero\_CO2==year,1,0)) %>%  
 filter(include==1) %>%  
 group\_by(id,category,var) %>%  
 summarise(value=sum(value,na.rm=TRUE))

## `summarise()` has grouped output by 'id', 'category'. You can override using  
## the `.groups` argument.

wd\_clusters <- spread(wd\_clusters,var,value) %>%  
 ungroup()  
  
cc\_clusters = wd\_clusters %>%  
 select(id) %>%   
 mutate(no\_clusters=NA) %>%   
 mutate(value=NA)  
  
for(x in 3:6) {  
   
 wd\_cluster\_summary <- kmeans(scale(wd\_clusters %>% select(-id,-category)),x,nstart=25)  
   
 blarg <- cbind(cc\_clusters %>% select(id),value=wd\_cluster\_summary$cluster) %>%   
 mutate(no\_clusters=x)  
 cc\_clusters <- rbind(cc\_clusters,blarg)  
   
}  
  
cc\_clusters <- left\_join(cc\_clusters %>% filter(!is.na(no\_clusters)),wd\_pca %>% select(id,model,PC1,PC2))

## Joining with `by = join\_by(id)`

cc\_clusters <- cc\_clusters %>%   
 mutate(label=paste0(no\_clusters," Clusters"))  
  
cc\_clusters$value <- as.factor(cc\_clusters$value)  
  
  
cc\_clusters %>% ggplot(.,aes(x=PC1,y=PC2,fill=value)) +  
 geom\_point(shape=21,size=3,alpha=0.6) +  
 theme\_wl() +  
 facet\_wrap(.~label) +  
 theme(legend.position="none") +  
 labs(title=paste0("Principal components explaining ",signif(sum(pca\_variance[1:2]),2), "% of residual subsector emissions variance"))



## get the scenario data  
wd\_subsectors <- wd\_scenarios\_emissions %>%  
 filter(var!="Kyoto Gases") %>%  
 filter(!grepl("Carbon Sequestration",var)) %>%  
 mutate(value=ifelse(is.na(value),0,value)) %>%  
 mutate(include=ifelse(net\_zero\_CO2==year,1,0)) %>%  
 mutate(include=ifelse(year==2020,1,include)) %>%   
 filter(include==1) %>%  
 group\_by(id,model,category,year,sector,var) %>%  
 summarise(value=sum(value,na.rm=TRUE))

## `summarise()` has grouped output by 'id', 'model', 'category', 'year',  
## 'sector'. You can override using the `.groups` argument.

## generate median scenarios for each  
  
wd\_subsectors\_total <- wd\_subsectors %>%   
 mutate(model="All") %>%   
 mutate(year=ifelse(year!=2020,2050,year)) %>%  
 group\_by(model,year,sector,var) %>%  
 summarise(percentile\_5th = quantile(value, probs = c(0.05)),  
 percentile\_95th = quantile(value, probs = c(0.95)),  
 value=median(value,na.rm=TRUE)) %>%   
 mutate(model="All")

## `summarise()` has grouped output by 'model', 'year', 'sector'. You can override  
## using the `.groups` argument.

wd\_subsectors <- wd\_subsectors %>%   
 mutate(year=ifelse(year!=2020,2050,year)) %>%  
 group\_by(model,year,sector,var) %>%  
 summarise(percentile\_5th = quantile(value, probs = c(0.05)),  
 percentile\_95th = quantile(value, probs = c(0.95)),  
 value=median(value,na.rm=TRUE))

## `summarise()` has grouped output by 'model', 'year', 'sector'. You can override  
## using the `.groups` argument.

## calculate relative and absolute changes for each model and variable  
wd\_subsectors <- rbind(wd\_subsectors,wd\_subsectors\_total)  
wd\_subsectors <- wd\_subsectors %>%   
 group\_by(model,sector,var) %>%   
 mutate(change\_rel = (first(value)-last(value))/first(value)) %>%   
 mutate(change\_rel\_5th = (first(percentile\_5th)-last(percentile\_5th))/first(percentile\_5th)) %>%   
 mutate(change\_rel\_95th = (first(percentile\_95th)-last(percentile\_95th))/first(percentile\_95th)) %>%   
 mutate(change\_rel=-round(change\_rel\*100,0)) %>%   
 mutate(change\_rel\_5th=-round(change\_rel\_5th\*100,0)) %>%   
 mutate(change\_rel\_95th=-round(change\_rel\_95th\*100,0))  
  
  
## rebase the data so that totals in 2020 vs 2050 are shown  
wd\_subsectors <- wd\_subsectors %>%   
 group\_by(model,sector,var) %>%   
 mutate(value\_rebased=first(value)-last(value)) %>%   
 mutate(value\_rebased=ifelse(year==2050,value,value\_rebased))  
  
wd\_subsectors <- wd\_subsectors %>%  
 mutate(value\_rebased=ifelse(value\_rebased<0,0,value\_rebased))  
  
wd\_subsectors$year = as.factor(wd\_subsectors$year)  
wd\_subsectors$year <- fct\_relevel(wd\_subsectors$year,"2020","2050")  
  
p1 <- wd\_subsectors %>%   
 filter(model!="All") %>%   
 filter(sector=="Energy") %>%   
 mutate(var=gsub("\\|Energy","",var)) %>%   
 ggplot(.,aes(x=value\_rebased,y=reorder(var,value\_rebased),alpha=year)) +  
 geom\_col(color='#636363',fill="#8da0cb") +  
 geom\_errorbar(data=wd\_subsectors %>%  
 filter(model!="All") %>%   
 filter(year==2050) %>%  
 filter(sector=="Energy") %>%  
 mutate(var=gsub("\\|Energy","",var)),inherit.aes = FALSE,  
 aes(xmin=percentile\_5th,xmax=percentile\_95th,y=reorder(var,value\_rebased)),  
 width=0.3) +  
 geom\_text(data=wd\_subsectors %>%  
 filter(model!="All") %>%   
 filter(sector=="Energy") %>%  
 mutate(var=gsub("\\|Energy","",var)) %>%   
 group\_by(model,var,change\_rel) %>%  
 summarise(value=sum(value\_rebased)),  
 inherit.aes = FALSE,aes(x=value+1,y=var,  
 label=ifelse(change\_rel<0,  
 paste0(change\_rel,"%"),  
 paste0("+",change\_rel,"%"))),  
 hjust=0,size=3) +  
 scale\_x\_continuous(expand = expansion(mult = c(0.05, 0.25))) +  
 scale\_alpha\_discrete(range=c(0.3,1)) +  
 scale\_fill\_brewer(palette="Blues") +  
 facet\_grid(.~model) +  
 theme\_wl() +  
 theme(legend.position="none",  
 axis.title=element\_blank(),  
 panel.grid.major.y = element\_blank()) +  
 labs(title="a. Energy sector emissions from 2020 to net zero",  
 subtitle=bquote("Gt"~CO[2]\*"e/yr"))

## `summarise()` has grouped output by 'model', 'var'. You can override using the  
## `.groups` argument.

## Warning: Using alpha for a discrete variable is not advised.

## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.

p2 <- wd\_subsectors %>%  
 filter(model!="All") %>%   
 filter(sector=="AFOLU") %>%   
 mutate(var=gsub("\\|AFOLU","",var)) %>%   
 ggplot(.,aes(x=value\_rebased,y=reorder(var,value\_rebased),alpha=year)) +  
 geom\_col(color='#636363',fill="#66c2a5") +  
 geom\_errorbar(data=wd\_subsectors %>%  
 filter(model!="All") %>%   
 filter(year==2050) %>%  
 filter(sector=="AFOLU") %>%   
 mutate(var=gsub("\\|AFOLU","",var)),inherit.aes = FALSE,  
 aes(xmin=percentile\_5th,xmax=percentile\_95th,y=reorder(var,value\_rebased)),  
 width=0.3) +  
 geom\_text(data=wd\_subsectors %>%  
 filter(model!="All") %>%   
 filter(sector=="AFOLU") %>%  
 mutate(var=gsub("\\|AFOLU","",var)) %>%   
 group\_by(model,var,change\_rel) %>%  
 summarise(value=sum(value\_rebased)),  
 inherit.aes = FALSE,aes(x=value+1.2,y=var,  
 label=ifelse(change\_rel<0,  
 paste0(change\_rel,"%"),  
 paste0("+",change\_rel,"%"))),  
 hjust=0,size=3) +  
 scale\_x\_continuous(expand = expansion(mult = c(0.05, 0.25))) +  
 scale\_alpha\_discrete(range=c(0.3,1)) +  
 #scale\_fill\_brewer(palette="Greens") +  
 facet\_grid(.~model) +  
 theme\_wl() +  
 theme(legend.position="none",  
 axis.title=element\_blank(),  
 panel.grid.major.y = element\_blank()) +  
 labs(title="b. AFOLU sector emissions from 2020 to net zero",  
 subtitle=bquote("Gt"~CO[2]\*"e/yr"))

## `summarise()` has grouped output by 'model', 'var'. You can override using the  
## `.groups` argument.

## Warning: Using alpha for a discrete variable is not advised.

p3 <- wd\_subsectors %>%   
 filter(model!="All") %>%   
 filter(sector!="Energy") %>%   
 filter(sector!="AFOLU") %>%   
 mutate(var=gsub("\\|Other","",var)) %>%   
 mutate(var=ifelse(var=="CO2","CO2|Other",var)) %>%   
 ggplot(.,aes(x=value\_rebased,y=reorder(var,value\_rebased),alpha=year)) +  
 geom\_col(color='#636363',fill="#fc8d62") +  
 geom\_errorbar(data=wd\_subsectors %>%  
 filter(model!="All") %>%   
 filter(year==2050) %>%   
 filter(sector!="Energy") %>%   
 filter(sector!="AFOLU") %>%   
 mutate(var=gsub("\\|Other","",var)) %>%   
 mutate(var=ifelse(var=="CO2","CO2|Other",var)),inherit.aes = FALSE,  
 aes(xmin=percentile\_5th,xmax=percentile\_95th,y=reorder(var,value\_rebased)),  
 width=0.3) +  
 geom\_text(data=wd\_subsectors %>%  
 filter(model!="All") %>%   
 filter(sector!="Energy") %>%  
 filter(sector!="AFOLU") %>%  
 mutate(var=gsub("\\|Other","",var)) %>%  
 mutate(var=ifelse(var=="CO2","CO2|Other",var)) %>%  
 group\_by(model,var,change\_rel) %>%  
 summarise(value=sum(value\_rebased)),  
 inherit.aes = FALSE,aes(x=value+0.4,y=var,  
 label=ifelse(change\_rel<0,  
 paste0(change\_rel,"%"),  
 paste0("+",change\_rel,"%"))),  
 hjust=0,size=3) +  
 scale\_x\_continuous(expand = expansion(mult = c(0.05, 0.25))) +  
 scale\_alpha\_discrete(range=c(0.3,1)) +  
 scale\_fill\_brewer(palette="Oranges") +  
 facet\_grid(.~model) +  
 theme\_wl() +  
 theme(legend.position="none",  
 axis.title=element\_blank(),  
 panel.grid.major.y = element\_blank()) +  
 labs(title="c. Other sector emissions from 2020 to net zero",  
 subtitle=bquote("Gt"~CO[2]\*"e/yr"))

## `summarise()` has grouped output by 'model', 'var'. You can override using the  
## `.groups` argument.

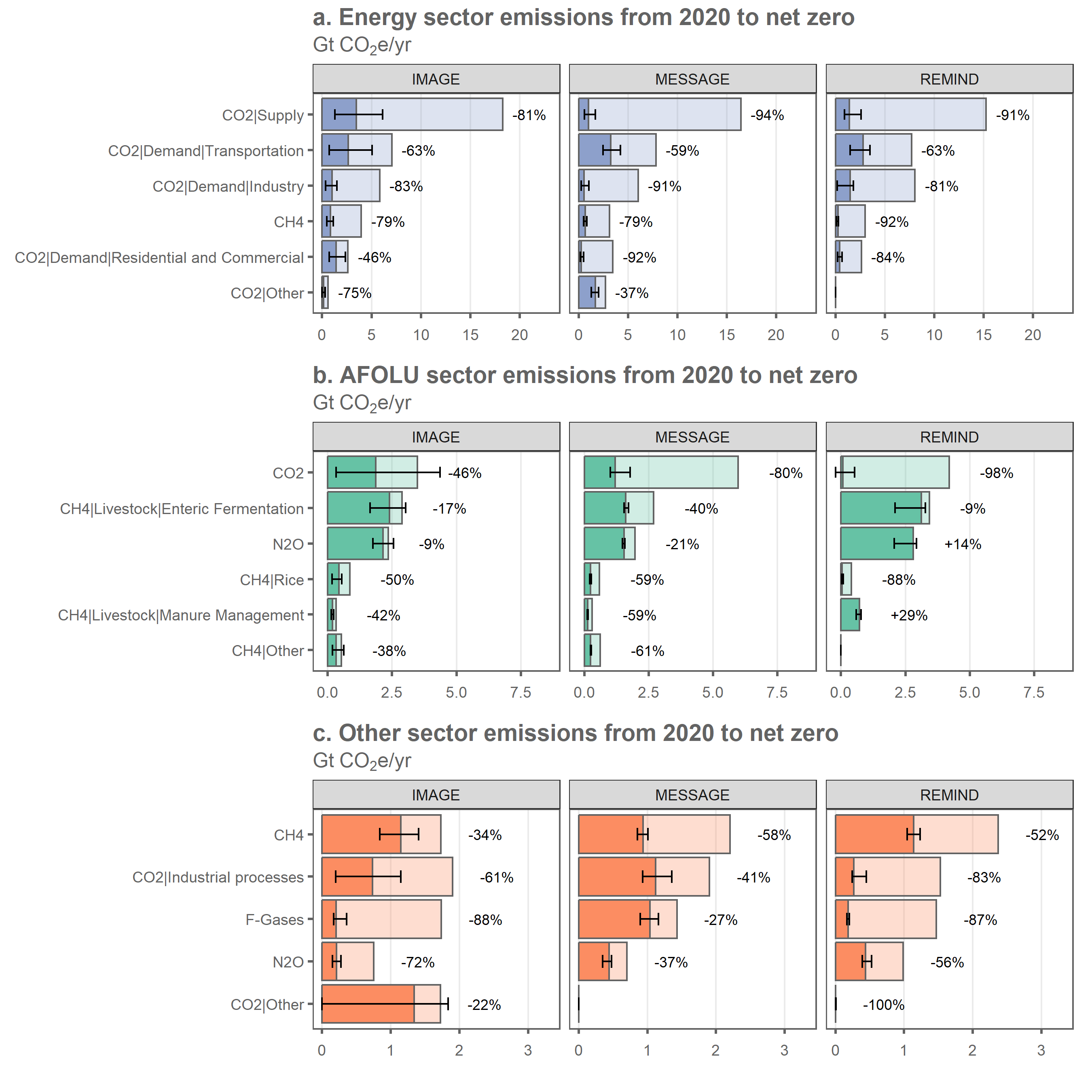
## Warning: Using alpha for a discrete variable is not advised.

## Scale for fill is already present.  
## Adding another scale for fill, which will replace the existing scale.

p1 / p2 / p3

## Warning: Removed 1 rows containing missing values (`geom\_text()`).

## Warning: Removed 1 rows containing missing values (`geom\_text()`).  
## Removed 1 rows containing missing values (`geom\_text()`).



wd\_subsectors\_summary <- wd\_subsectors %>%   
 ungroup() %>%   
 filter(year==2050) %>%   
 mutate(sector=as.character(sector)) %>%   
 mutate(sector=ifelse(sector=="F-Gases","Other",sector)) %>%   
 mutate(sector=ifelse(sector=="Industrial processes","Other",sector)) %>%   
 mutate(change\_rel\_5th=ifelse(percentile\_5th==0,-100,change\_rel\_5th)) %>%   
 mutate(value=paste0(signif(value,2)," [",signif(percentile\_5th,2),"-",signif(percentile\_95th,2),"]")) %>%   
 mutate(change\_rel=paste0(change\_rel," [",change\_rel\_5th," to ",change\_rel\_95th,"]")) %>%   
 select(model,sector,var,value,change\_rel)  
  
wd\_subsectors\_summary$sector <- as.factor(wd\_subsectors\_summary$sector)  
wd\_subsectors\_summary$sector <- fct\_relevel(wd\_subsectors\_summary$sector,  
 "AFOLU","Energy","Other")  
  
wd\_subsectors\_summary <- cbind(spread(wd\_subsectors\_summary %>%   
 mutate(model=paste0(model,"\_level")) %>%   
 select(-change\_rel),model,value),  
 spread(wd\_subsectors\_summary %>%   
 mutate(model=paste0(model,"\_reduction")) %>%   
 select(-value),model,change\_rel) %>%   
 select(-sector,-var)) %>%   
 arrange(sector,var)  
  
  
addWorksheet(wb\_analysis,"subsector\_residual\_at\_nz")  
writeData(wb\_analysis, sheet = "subsector\_residual\_at\_nz",  
 wd\_subsectors\_summary, colNames = T, rowNames = F)

cc\_expl <- read.xlsx('../Data/AR6\_Scenarios\_Database\_metadata\_indicators\_v1.1\_1.xlsx',sheet=2)  
cc\_expl <- cc\_expl %>%  
 mutate(id=paste0(Model,"|",Scenario)) %>%   
 select(id,Policy\_category,Policy\_category\_name,Project\_study,reference=`Literature.Reference.(if.applicable)`,  
 Ssp\_family,Technology\_category\_name)  
  
## explanatory factors vs. total residual?  
wd\_expl <- left\_join(wd\_scenarios\_emissions,cc\_expl,by="id") %>%   
 mutate(include=ifelse(year==net\_zero\_CO2,1,0)) %>%   
 filter(include==1) %>%   
 select(-include) %>%   
 filter(var!="Kyoto Gases") %>%   
 filter(!is.na(subsector))  
  
wd\_expl <- wd\_expl %>%   
 group\_by(id,model,category,Policy\_category,Policy\_category\_name,Project\_study,reference,  
 Ssp\_family,Technology\_category\_name) %>%   
 summarise(value=sum(value,na.rm=TRUE)) %>%  
 mutate(Ssp\_family=as.character(Ssp\_family)) %>%   
 mutate(Ssp\_family=ifelse(Ssp\_family=="1","SSP1",Ssp\_family)) %>%  
 mutate(Ssp\_family=ifelse(Ssp\_family=="2","SSP2",Ssp\_family)) %>%   
 mutate(Ssp\_family=as.factor(Ssp\_family))

## `summarise()` has grouped output by 'id', 'model', 'category',  
## 'Policy\_category', 'Policy\_category\_name', 'Project\_study', 'reference',  
## 'Ssp\_family'. You can override using the `.groups` argument.

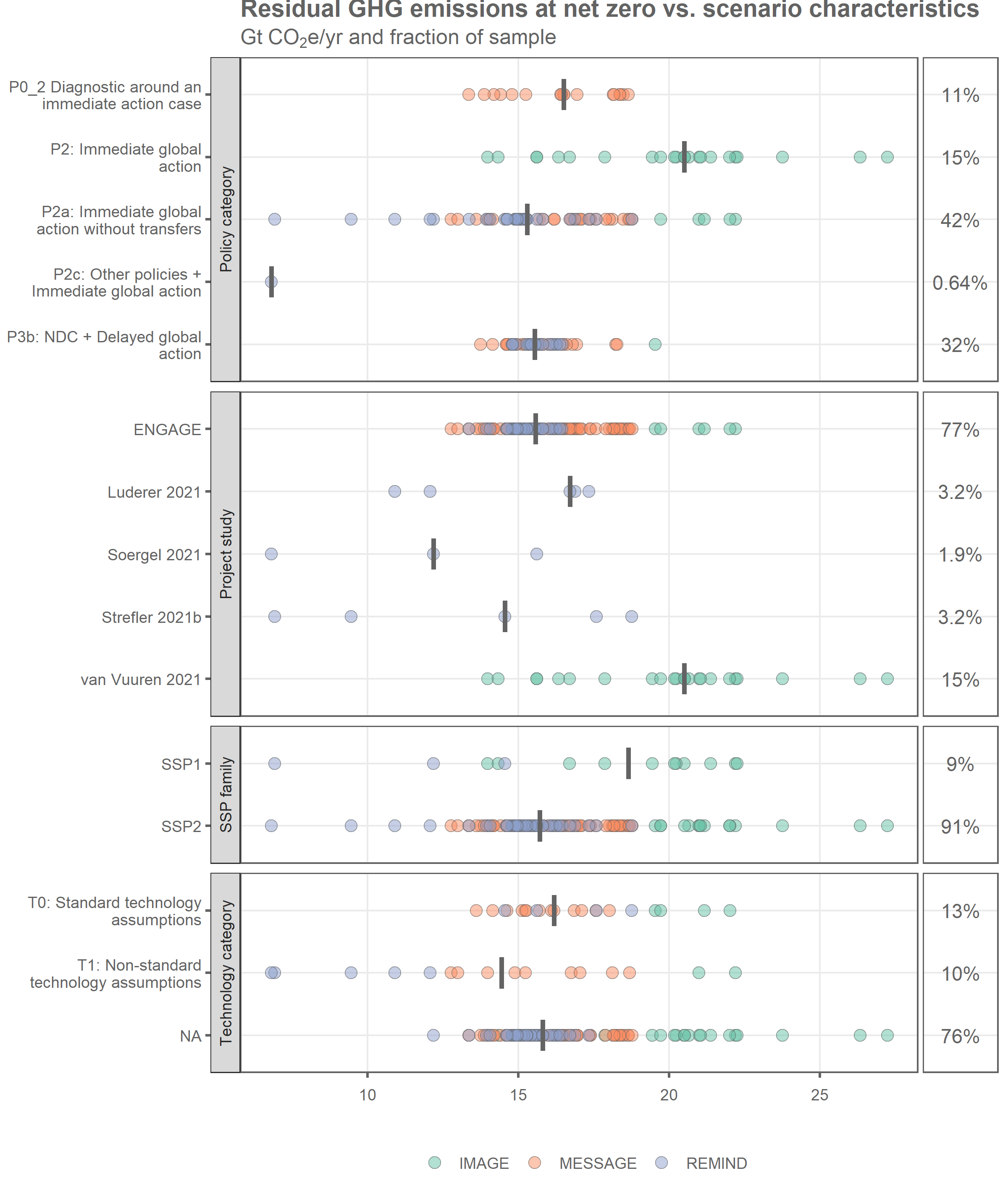
wd\_expl <- wd\_expl %>%   
 ungroup() %>%   
 select(-category,-Policy\_category,-reference)  
wd\_expl <- gather(wd\_expl,key,var,-id,-value,-model)

## Warning: attributes are not identical across measure variables; they will be  
## dropped

p1 <- wd\_expl %>%   
 ggplot(.,aes(x=value,y=var)) +  
 geom\_point(shape=21,size=3,alpha=0.5,color="#636363",aes(fill=model)) +  
 stat\_summary(fun=median,geom="crossbar", color="#636363",width=0.5,show.legend = FALSE) +  
 facet\_grid(key~.,scales="free\_y",space="free",switch="y",  
 labeller=as\_labeller(c(Policy\_category\_name="Policy category",  
 Project\_study="Project study",  
 Ssp\_family="SSP family",  
 Technology\_category\_name="Technology category"))) +  
 scale\_y\_discrete(limits=rev,labels = function(x) str\_wrap(x, width = 25)) +  
 theme\_wl() +  
 theme(legend.position="bottom",  
 legend.title=element\_blank(),  
 axis.title = element\_blank(),  
 strip.text = element\_text(color="#252525"),  
 plot.margin = margin(0,0,0,0,"cm"),) +  
 labs(title="Residual GHG emissions at net zero vs. scenario characteristics",  
 subtitle=bquote("Gt"~CO[2]\*"e/yr and fraction of sample"))  
  
  
## how many scenarios are from the ENGAGE project?  
wd\_expl\_totals <- wd\_expl %>%   
 group\_by(key,var) %>%   
 summarise(n=n(),  
 fraction=n/length(no\_scenarios$id)) %>%   
 mutate(fraction=signif(fraction\*100,2))

## `summarise()` has grouped output by 'key'. You can override using the `.groups`  
## argument.

p2 <- wd\_expl\_totals %>% ggplot(.,aes(x="% of sample",y=var,label=paste0(fraction,"%"))) +  
 geom\_text(size=4,color="#636363") +  
 facet\_grid(key~.,scales="free\_y",space="free") +  
 scale\_y\_discrete(limits=rev) +  
 theme\_wl() +  
 theme(legend.position="none",  
 axis.title = element\_blank(),  
 axis.text = element\_blank(),  
 plot.background = element\_blank(),  
 strip.background = element\_blank(),  
 strip.text = element\_blank(),  
 axis.ticks = element\_blank(),  
 panel.grid.major.x = element\_blank(),  
 plot.margin = margin(0,0,0,0,"cm"),  
 axis.text.x = element\_blank())  
  
  
p1 + p2 + plot\_layout(widths=c(9,1))



# pd <- pd %>%   
# mutate(include=ifelse(year==2020,1,0)) %>%   
# mutate(include=ifelse(year==net\_zero\_CO2,1,include)) %>%   
# mutate(include=ifelse(year==2100,1,include)) %>%   
# filter(include==1)  
  
  
  
# pd <- wd\_scenarios\_emissions %>%   
# filter(var!="Kyoto Gases") %>%   
# filter(!grepl("Carbon Sequestration",var)) %>%   
# mutate(include=ifelse(year>=net\_zero\_CO2,1,0)) %>%   
# filter(include==1) %>%   
# group\_by(id,model,scenario,category,year,net\_zero\_CO2) %>%   
# summarise(value=sum(value,na.rm=TRUE)) #%>%   
# group\_by(id,model,scenario,category) %>%   
# summarise(emissions\_nz=first(value),  
# emissions\_2100=last(value))  
  
  
  
# pd <- gather(pd,var,value,-id,-category)  
#   
# pd %>% filter(var=="change\_rel") %>% ggplot(.,aes(x=value,y=category,fill=category)) +  
# geom\_point(shape=21,size=3,alpha=0.6,color="#636363") +  
# theme\_wl() +  
# theme(legend.position="none",  
# axis.title = element\_blank(),  
# plot.margin = margin(0,0,0,0,"cm"))  
  
#   
#   
# pd %>% ggplot(.,aes(x=emissions\_2100,y=emissions\_nz))+  
# geom\_abline(intercept=0,color="#636363") +  
# geom\_point(shape=21,size=3,alpha=0.6,color="#636363",aes(fill=category)) +  
# geom\_text(data=data.frame(emissions\_nz=c(6,5),  
# emissions\_2100=c(5,6),  
# label=c("Decrease","Increase")),  
# aes(label=label),  
# color="#636363",  
# angle=35,  
# hjust=0) +  
# facet\_wrap(.~category) +  
# theme\_wl() +  
# theme(legend.title=element\_blank()) +  
# xlim(5,25) +  
# ylim(5,25) +  
# ylab("GHG emissions at net-zero") +  
# xlab("GHG emissions at 2100") +  
# labs(title="Residual GHG emissions after net-zero",  
# subtitle="GtCO2e/yr")  
#   
#   
# pd <- wd\_scenarios\_emissions %>%   
# filter(var!="Kyoto Gases") %>%   
# filter(!grepl("Carbon Sequestration",var)) %>%   
# group\_by(id,model,scenario,category,year,net\_zero\_CO2) %>%   
# summarise(value=sum(value,na.rm=TRUE))  
#   
# pd %>% ggplot(.,aes(x=year,y=value,group=id,color=model)) +  
# geom\_line() +  
# facet\_grid(category~model) +  
# theme\_wl() +  
# theme(legend.position="none") +  
# scale\_y\_continuous(limits=c(0,60),breaks=c(0,10,20,30,40,50,60))

wd\_standards <- wd\_scenarios\_emissions %>%   
 filter(var!="Kyoto Gases") %>%   
 filter(!grepl("Carbon Sequestration",var)) %>%   
 mutate(include=ifelse(year==net\_zero\_CO2,1,0)) %>%  
 mutate(include=ifelse(year==2020,1,include)) %>%   
 filter(include==1) %>%   
 select(-include)  
  
wd\_standards <- wd\_standards %>%   
 mutate(mapping=ifelse(sector=="AFOLU","AFOLU",NA)) %>%   
 mutate(mapping=ifelse(var=="CO2|Energy|Supply","Power",mapping)) %>%   
 mutate(mapping=ifelse(var=="CO2|Energy|Demand|Industry","Cement/Iron and Steel",mapping)) %>%   
 mutate(mapping=ifelse(var=="CO2|Industrial processes","Cement/Iron and Steel",mapping)) %>%   
 mutate(mapping=ifelse(var=="CO2|Energy|Demand|Residential and Commercial","Buildings",mapping))  
  
wd\_standards <- wd\_standards %>%   
 filter(!is.na(mapping)) %>%   
 group\_by(id,category,year,mapping) %>%   
 summarise(value=sum(value,na.rm=TRUE))

## `summarise()` has grouped output by 'id', 'category', 'year'. You can override  
## using the `.groups` argument.

wd\_standards <- wd\_standards %>%   
 group\_by(id,category,mapping) %>%   
 mutate(rel\_change=(first(value)-last(value))/first(value)) %>%   
 mutate(rel\_change=rel\_change\*100)  
  
wd\_standards <- median\_and\_range(wd\_standards %>% select(id,category,sector=mapping,value=rel\_change),c("sector","category"))

## `summarise()` has grouped output by 'sector'. You can override using the  
## `.groups` argument.

wd\_standards$category <- fct\_relevel(wd\_standards$category,"C1","C2","C3")  
wd\_standards <- spread(wd\_standards,category,value)  
  
  
addWorksheet(wb\_analysis,"scenarios\_vs\_standards")  
writeData(wb\_analysis, sheet = "scenarios\_vs\_standards",  
 wd\_standards, colNames = T, rowNames = F)

saveWorkbook(wb\_figures,"../Results/figure\_data.xlsx",overwrite=T)

## Warning in file.create(to[okay]): kann Datei '../Results/figure\_data.xlsx'  
## nicht erzeugen. Grund 'Permission denied'

saveWorkbook(wb\_analysis,"../Results/analysis.xlsx",overwrite=T)

## Warning in file.create(to[okay]): kann Datei '../Results/analysis.xlsx' nicht  
## erzeugen. Grund 'Permission denied'

#saveWorkbook(wb\_data,"../Results/processed\_data.xlsx",overwrite=T)