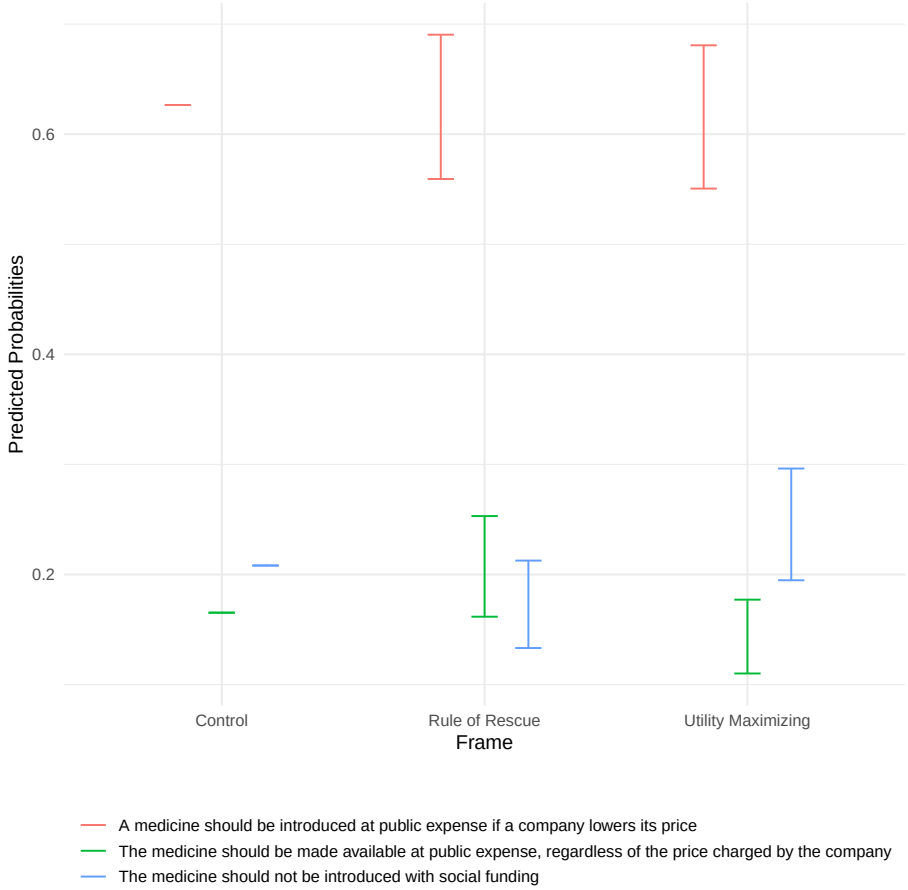
```r
#########
# Framing Experiment
#########
cat("\014")
```

```r
rm(list=ls())
setwd("/Users/hectorbahamonde/research/Fimea/")

# Pacman
if (!require("pacman")) install.packages("pacman"); library(pacman)
```

```
## Loading required package:  pacman
```

```r
# loadings
load("/Users/hectorbahamonde/research/Fimea/dat.RData")


# recoding
p_load(dplyr)
dat <- dat %>%
  mutate(outcome = factor(outcome,
                          levels = c(
                            "The medicine should not be introduced with social funding",  #
                            "A medicine should be introduced at public expense if a company
                            "The medicine should be made available at public expense, regard
                          ), ordered = TRUE))

# Remove rows
p_load(tidyverse)
dat <- dat %>% filter(!is.na(outcome))
dat <- dat %>% filter(Frame != "Frame A") %>% mutate(Frame = fct_drop(Frame))  # Drop unused

# Recode the Frame variable
dat$Frame <- factor(dat$Frame,
                    levels = c("Frame B", "Frame C", "Frame D"),  # Questionnaires
                    labels = c("Control", "Rule of Rescue", "Utility Maximizing"))

# Set "Control" as the reference category
dat$Frame <- relevel(dat$Frame, ref = "Control")


# models

# Fit the ordinal logistic regression model
p_load(MASS)
model <- polr(outcome ~ Frame + M1_1 + M1_2_1 + M1_3 + M1_5, data = dat, Hess = TRUE)
```

```r
#model <- polr(outcome ~ Frame + M2_1 + M2_2 + M1_1 + M1_2_1 + M1_3 + M1_5, data = dat, Hess
#summary(model)

# Generate predicted probabilities for a predictor
p_load(ggplot2, ggeffects)
predicted_probs <- ggpredict(model, terms = "Frame")

# plot
p_load(ggplot2)
ggplot(predicted_probs,
       aes(x = x, y = predicted, color = response.level)) +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high),
                position = position_dodge(width = 0.5), width = 0.3) +
  theme_minimal() +
  theme(
    legend.position = "bottom",  # Keep legend at the bottom
    legend.direction = "vertical",  # Stack legend levels vertically
    legend.key.height = unit(0.5, "cm")  # Increase spacing between legend items
  ) +
  guides(colour = guide_legend(title = "", ncol = 1)) +
  labs(x = "Frame", y = "Predicted Probabilities")
```

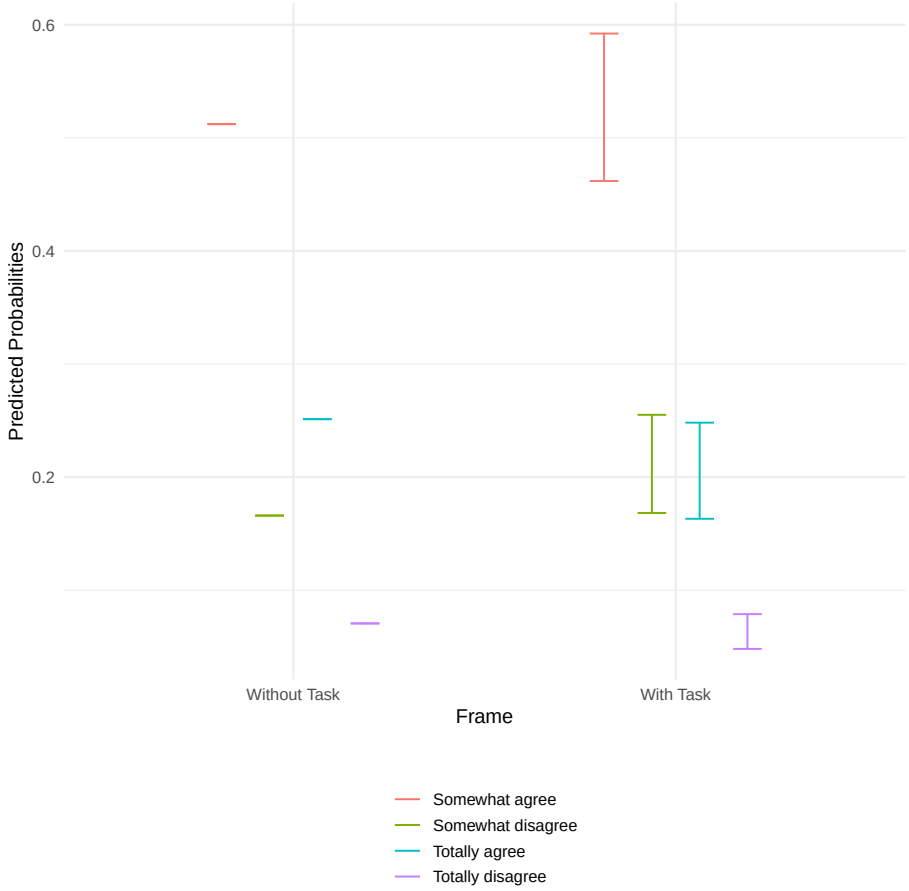



```
#########
# Decision Making Experiment
#########
cat("\014")
```

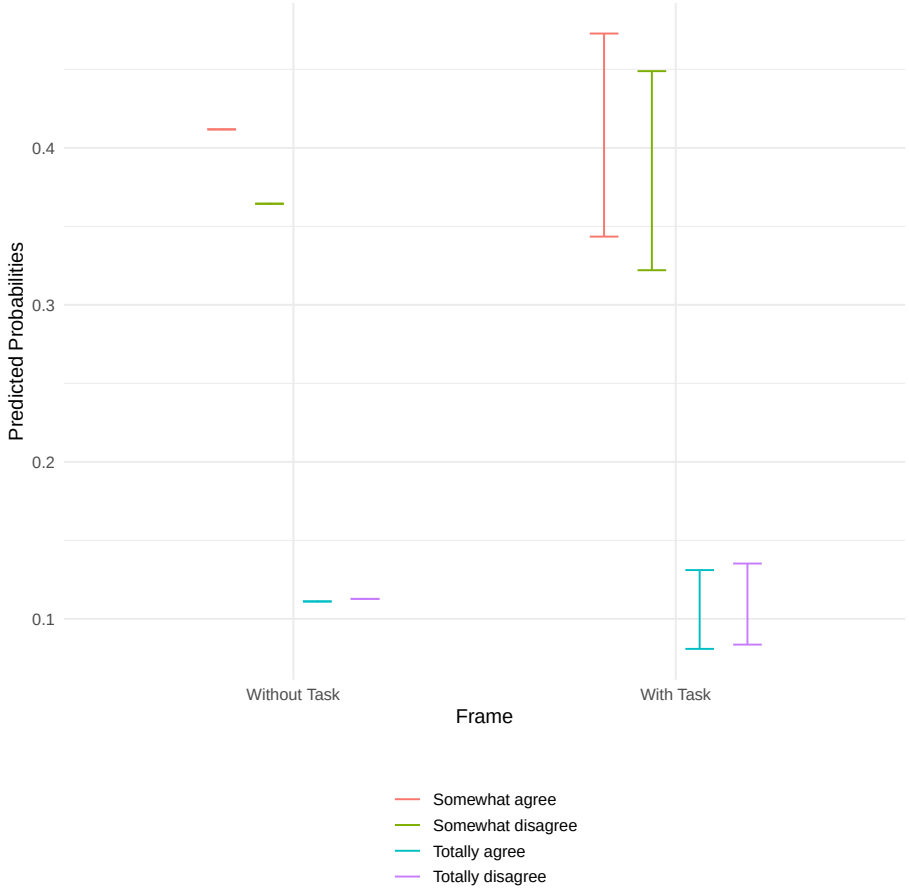
```
rm(list=ls())
setwd("/Users/hectorbahamonde/research/Fimea/")

# Pacman
if (!require("pacman")) install.packages("pacman"); library(pacman)

# loadings
load("/Users/hectorbahamonde/research/Fimea/dat.RData")

# From Katri
```

```r
# The hypotheses are broadly that
## 1. Framing Exp: framing affects how individuals answer the decision making task.
## 2. Dec Mak Exp: having to do the decision making task (vs. not having to do it) affects

# "Frame A" (no dec making task but yes questions) and "Frame B" (yes dec mak task and yes
## Right comparison is A (dec mak) vs B (no dec mak)
## Can't compare A v B+C+D because C and D do have other framings (rule of resq. and ut. ma

# keep Frame A and B only
p_load(tidyverse)
dat <- dat %>% filter(Frame %in% c("Frame A", "Frame B")) %>% mutate(Frame = fct_drop(Frame)

# re-factoring
p_load(tidyverse)
dat <- dat %>%
  filter(Frame %in% c("Frame A", "Frame B")) %>%
  mutate(Frame = recode(Frame,
                        "Frame A" = "Without Task",
                        "Frame B" = "With Task"),
         Frame = factor(Frame, levels = c("Without Task", "With Task"))) # Convert to facto


# recoding
dat$M6_7_1 <- na_if(dat$M6_7_1, "I don't know")
dat$M6_7_2 <- na_if(dat$M6_7_2, "I don't know")

p_load(dplyr, stringr)

# M6_7_1
dat <- dat %>%
  mutate(M6_7_1 = str_replace(M6_7_1, "Jokseenkin eri mieltä", "Somewhat disagree")
         )

# M6_7_2
dat <- dat %>% mutate(M6_7_2 = str_replace(M6_7_2, "Jokseenkin eri mieltä", "Somewhat disagr
dat <- dat %>% mutate(M6_7_2 = str_replace(M6_7_2, "Jokseenkin samaa mieltä", "Somewhat agre

# re-factor ordered
dat$M6_7_1 <- factor(dat$M6_7_1,
                     levels = c("Somewhat disagree", "Somewhat agree", "Totally disagree", "
                     ordered = TRUE)

dat$M6_7_2 <- factor(dat$M6_7_2,
                     levels = c("Somewhat disagree", "Somewhat agree", "Totally disagree", "
                     ordered = TRUE
```

```
                    )

# to numeric
dat$M6_8 <- as.numeric(sub("^([0-9]+).*", "\\1", dat$M6_8))


# M6_7_1 M6_7_2 M6_8

# models
m1.a <- lm(M6_8 ~ Frame, data = dat)
m1.b <- lm(M6_8 ~ Frame + M2_1 + M2_2 + M1_1 + M1_2_1 + M1_3 + M1_5, data = dat)
#summary(m1)

p_load(MASS)
model.1 <- polr(M6_7_1 ~ Frame + M1_1 + M1_2_1 + M1_3 + M1_5, data = dat, Hess = TRUE)
model.2 <- polr(M6_7_2 ~ Frame + M1_1 + M1_2_1 + M1_3 + M1_5, data = dat, Hess = TRUE)


# Generate predicted probabilities for a predictor
p_load(ggplot2, ggeffects)
predicted_probs.1 <- ggpredict(model.1, terms = "Frame")
predicted_probs.2 <- ggpredict(model.2, terms = "Frame")

# plot
p_load(ggplot2)
ggplot(predicted_probs.1,
       aes(x = x, y = predicted, color = response.level)) +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high),
                position = position_dodge(width = 0.5), width = 0.3) +
  theme_minimal() +
  theme(
    legend.position = "bottom",  # Keep legend at the bottom
    legend.direction = "vertical",  # Stack legend levels vertically
    legend.key.height = unit(0.5, "cm")  # Increase spacing between legend items
  ) +
  guides(colour = guide_legend(title = "", ncol = 1)) +
  labs(x = "Frame", y = "Predicted Probabilities")
```




```r
# plot
p_load(ggplot2)
ggplot(predicted_probs.2,
       aes(x = x, y = predicted, color = response.level)) +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high),
                position = position_dodge(width = 0.5), width = 0.3) +
  theme_minimal() +
  theme(
    legend.position = "bottom",  # Keep legend at the bottom
    legend.direction = "vertical",  # Stack legend levels vertically
    legend.key.height = unit(0.5, "cm")  # Increase spacing between legend items
  ) +
  guides(colour = guide_legend(title = "", ncol = 1)) +
  labs(x = "Frame", y = "Predicted Probabilities")
```




```
# Load necessary package
p_load(texreg)

# Use cat() to print LaTeX output correctly
texreg(list(m1.a, m1.b), booktabs = F,
       omit.coef = "_2_",
       scalebox = 0.6)
```

graphicx

	Model 1	Model 2
(Intercept)	6.28***	7.84***
	(0.09)	(1.89)
FrameWith Task	−0.09	−0.07
	(0.14)	(0.15)
M2_1Good for		1.73*
		(0.85)
M2_1Moderate		1.10
		(0.83)
M2_1Pretty good		1.33
		(0.84)
M2_1Quite bad		0.26
		(0.88)
M2_2Yes		−0.03
		(0.17)
M1_1Male		−0.29*
		(0.15)
M1_1Other		0.30
		(0.63)
M1_3Separated or divorced		−0.66**
		(0.24)
M1_3Unmarried		−0.02
		(0.18)
M1_3Widow		−0.67
		(0.38)
M1_5I am on maternity or paternity leave, parental leave or childcare leave		0.26
		(1.07)
M1_5I am partly working, partly retired		−0.62
		(0.93)
M1_5I am retired		−1.14
		(0.79)
M1_5I am unemployed or laid off		−1.59*
		(0.80)
M1_5I study		−0.94
		(0.80)
M1_5I work full-time		−1.08
		(0.76)
M1_5I work part-time		−1.05
		(0.83)
M1_5Something else		−1.64
		(0.90)
R^2	0.00	0.12
Adj. R^2	−0.00	0.04
Num. obs.	1040	1040

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 1: Statistical models