Research Appendix

Nicholas K. Neuteufel

November 9, 2015

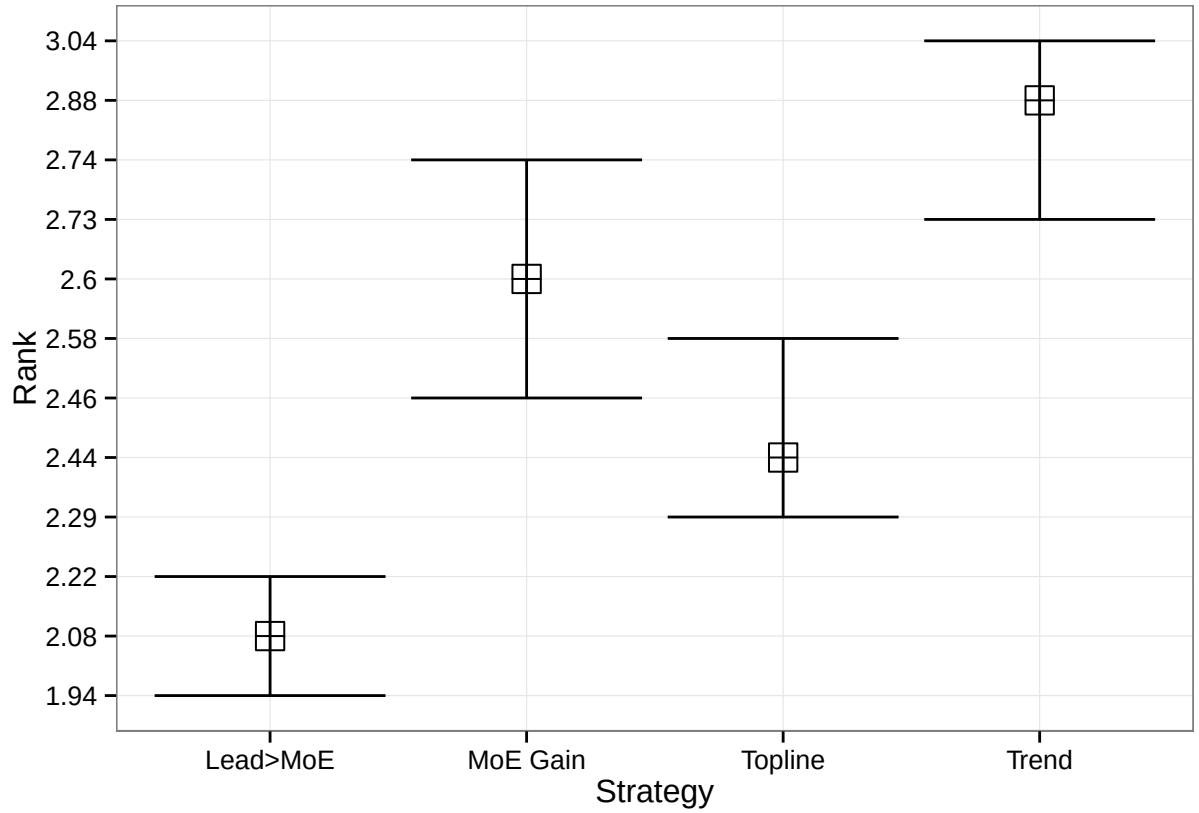
R code for the data collection, management, and analysis (including the traditional ordered probit):

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

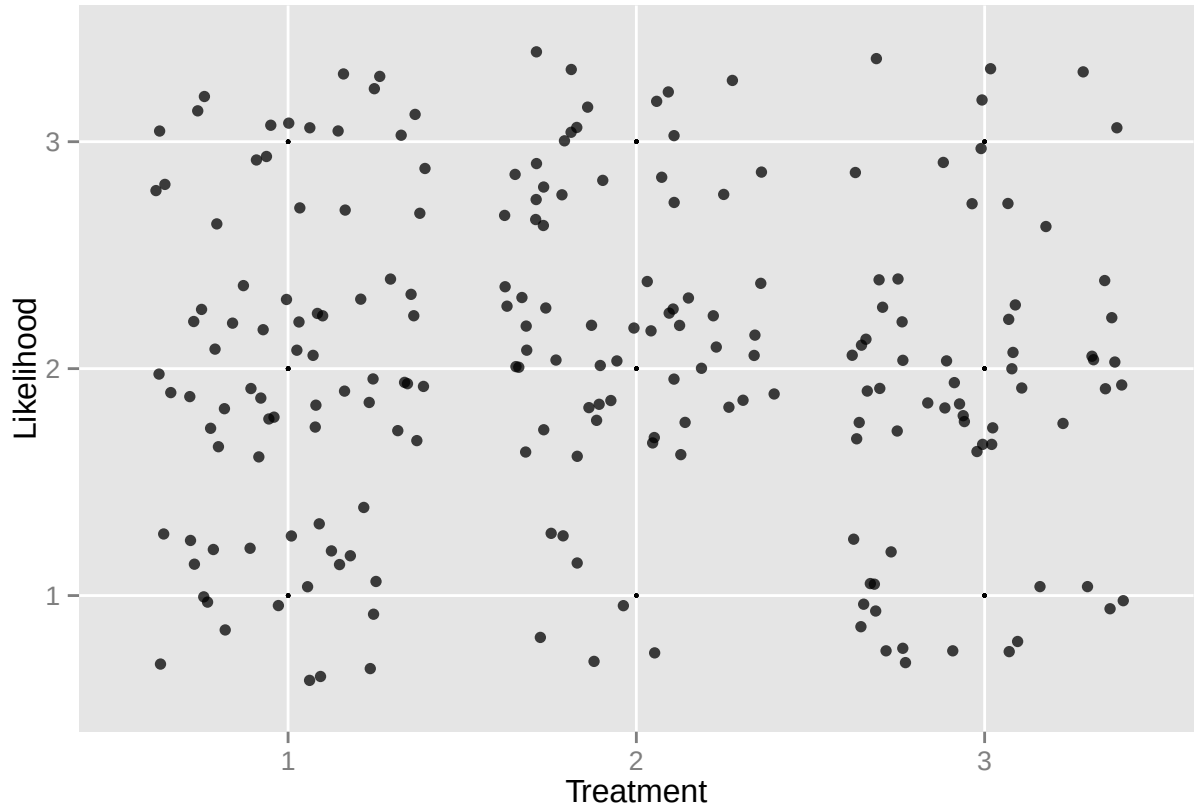
```
#Import csv file from Qualtrics here
data <- read.csv(file="10-26data.csv", h=T, na.strings = "")

#Summary statistics

#Ranking the comms strategies
#n of 212 is different than typical n because drop-out/non-response/not changing the order
PollResult <- data$PollResult[!is.na(data$PollResult)]
LeadOverMoE <- data$LeadOverMoE[!is.na(data$LeadOverMoE)]
BattleForMoE <- data$BattleForMoE[!is.na(data$BattleForMoE)]
Trend <- data$Trend[!is.na(data$Trend)]

resultUpper <- round(mean(PollResult) + 1.96*(sd(PollResult)/sqrt(212)), 2)
resultLower <- round(mean(PollResult) - 1.96*(sd(PollResult)/sqrt(212)), 2)
resultMean <- round(mean(PollResult), 2)
result <- c(resultMean, resultLower, resultUpper)

leadUpper <- round(mean(LeadOverMoE) + 1.96*(sd(LeadOverMoE)/sqrt(212)), 2)
leadLower <- round(mean(LeadOverMoE) - 1.96*(sd(LeadOverMoE)/sqrt(212)), 2)
leadMean <- round(mean(LeadOverMoE), 2)
lead <- c(leadMean, leadLower, leadUpper)

battleUpper <- round(mean(BattleForMoE) + 1.96*(sd(BattleForMoE)/sqrt(212)), 2)
battleLower <- round(mean(BattleForMoE) - 1.96*(sd(BattleForMoE)/sqrt(212)), 2)
battleMean <- round(mean(BattleForMoE), 2)
battle <- c(battleMean, battleLower, battleUpper)

trendUpper <- round(mean(Trend) + 1.96*(sd(Trend)/sqrt(212)), 2)
trendLower <- round(mean(Trend) - 1.96*(sd(Trend)/sqrt(212)), 2)
trendMean <- round(mean(Trend), 2)
```

```
trend <- c(trendMean, trendLower, trendUpper)

comms <- rbind(result, lead, battle, trend)
strat <- c("Topline", "Lead>MoE", "MoE Gain", "Trend")
comms <- cbind(comms, strat)
colnames(comms) <- c("Mean", "Upper", "Lower", "Strategy")
comms <- data.frame(comms)

strategies <- ggplot(comms, aes(y=Mean, x=Strategy)) + guides(colour=F)
strategies <- strategies +
  geom_point(shape=12, size=5) + geom_errorbar(wigth=0.1, aes(ymax=comms$Upper, ymin=comms$Lower))
strategies + ylab("Rank") + theme_bw()
```




```
#Subset by assignment group
controlGroup <- filter(data, !is.na(Control))
barGroup <- filter(data, !is.na(Bar))
dotGroup <- filter(data, !is.na(Dot))

#Make a result vector/column (called Likelihood)for the probit model
#and tell R the treatment group
controlGroup$Treatment <- 1
controlGroup$Likelihood <- controlGroup$Control
barGroup$Treatment <- 2
barGroup$Likelihood <- barGroup$Bar
dotGroup$Treatment <- 3
```

```
dotGroup$Likelihood <- dotGroup$Dot

final <- rbind(controlGroup, barGroup, dotGroup)
final <- filter(final, Likelihood != "I Do Not Know",
                Likelihood != "Very Unlikely", Likelihood != "Unlikely",
                Likelihood != "Somewhat Unlikely")
final <- select(final, -c(Control, Dot, Bar))
final$Likelihood <- as.character(final$Likelihood)
final$Likelihood[final$Likelihood=="Neither Likely nor Unlikely"] <- "Neither Likely Nor Unlikely"
#Stupid me
final$Likelihood[final$Likelihood=="Very Likely"] <- "Likely" #Group these
#Ordering the Likelihoods
final$Likelihood[final$Likelihood=="Neither Likely Nor Unlikely"] <- 1
final$Likelihood[final$Likelihood=="Somewhat Likely"] <- 2
final$Likelihood[final$Likelihood=="Likely"] <- 3
#Factor the important things
final$Likelihood <- as.factor(final$Likelihood)
final$Treatment <- as.factor(final$Treatment)

#Treatment summary statistics
library(ggplot2)
ggplot(final, aes(x=Treatment, y=Likelihood)) + geom_point(size=0.75) +
  geom_jitter(alpha = 0.75)
```



```r
#Marginal effects: Simple
library(mfx)
```

```
## Loading required package: sandwich
## Loading required package: lmtest
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: MASS
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
##
## Loading required package: betareg
```

```r
probeME <- probitmfx(as.ordered(Likelihood) ~
                       PresentationPreference+StatsClass+Treatment,
                     data=final, atmean=F, robust=T)
probeME
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ PresentationPreference +
##     StatsClass + Treatment, data = final, atmean = F, robust = T)
##
## Marginal Effects:
##                                dF/dx Std. Err.       z   P>|z|
## PresentationPreferenceText  0.037484  0.060262  0.6220 0.53393
## StatsClassYes              -0.026538  0.059009 -0.4497 0.65290
## Treatment2                  0.172546  0.054643  3.1577 0.00159 **
## Treatment3                  0.012167  0.061949  0.1964 0.84429
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
## [1] "PresentationPreferenceText" "StatsClassYes"
## [3] "Treatment2"                 "Treatment3"
```

```r
#Marginal effects: Pretreatment too
#Make poll trust into "Agree", "Disagree", and "Neither"
final$PollTrust <- as.character(final$PollTrust)
final$PollTrust[final$PollTrust=="Strongly Agree"] <- "Agree"
final$PollTrust[final$PollTrust=="Somewhat Agree"] <- "Agree"
final$PollTrust[final$PollTrust=="Strongly Disagree"] <- "Disagree"
```

```r
final$PollTrust[final$PollTrust=="Somewhat Disagree"] <- "Disagree"
final$PollTrust <- as.factor(final$PollTrust)
#Group "Less than once a month" for horse race exposure into Never and
#"Once a month/weeek" into "Occasionally" and "2-3x a week" and "Daily" into "Often"
final$HorseRaceExposure <- as.character(final$HorseRaceExposure)
final$HorseRaceExposure[final$HorseRaceExposure=="Less than Once a Month"] <- "Never"
final$HorseRaceExposure[final$HorseRaceExposure=="Once a Month"] <- "Occasionally"
final$HorseRaceExposure[final$HorseRaceExposure=="Once a Week"] <- "Occasionally"
final$HorseRaceExposure[final$HorseRaceExposure=="2-3 Times a Week"] <- "Often"
final$HorseRaceExposure[final$HorseRaceExposure=="Daily"] <- "Often"
final$HorseRaceExposure <- as.factor(final$HorseRaceExposure)
#Find marginal effects
probeMEpretreatment <- probitmfx(as.ordered(Likelihood) ~
                                   PollTrust+HorseRaceExposure+
                                   PresentationPreference+StatsClass+Treatment,
                                 data=final, atmean=F, robust=T)
probeMEpretreatment
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ PollTrust + HorseRaceExposure +
##     PresentationPreference + StatsClass + Treatment, data = final,
##     atmean = F, robust = T)
##
## Marginal Effects:
##                                       dF/dx Std. Err.       z     P>|z|
## PollTrustDisagree                  -0.202659  0.065300 -3.1035 0.0019123
## PollTrustNeither Agree nor Disagree -0.029418  0.083649 -0.3517 0.7250788
## HorseRaceExposureOccasionally        0.126011  0.062198  2.0260 0.0427696
## HorseRaceExposureOften               0.044092  0.066912  0.6589 0.5099283
## PresentationPreferenceText           0.032503  0.058777  0.5530 0.5802675
## StatsClassYes                       -0.043860  0.057621 -0.7612 0.4465455
## Treatment2                           0.193233  0.051269  3.7690 0.0001639
## Treatment3                           0.025062  0.058807  0.4262 0.6699775
##
## PollTrustDisagree                  **
## PollTrustNeither Agree nor Disagree
## HorseRaceExposureOccasionally      *
## HorseRaceExposureOften
## PresentationPreferenceText
## StatsClassYes
## Treatment2                         ***
## Treatment3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
## [1] "PollTrustDisagree"
## [2] "PollTrustNeither Agree nor Disagree"
## [3] "HorseRaceExposureOccasionally"
## [4] "HorseRaceExposureOften"
## [5] "PresentationPreferenceText"
## [6] "StatsClassYes"
```

```
## [7] "Treatment2"
## [8] "Treatment3"
```

```
#Test the standard errors for heterosckedasicity
probeMEpretreatmentNotRobust <- probitmfx(as.ordered(Likelihood) ~
                                          PollTrust+HorseRaceExposure+PresentationPreference+
                                          StatsClass+Treatment,
                                        data=final, atmean=F, robust=F)
probeMEpretreatmentNotRobust
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ PollTrust + HorseRaceExposure +
##     PresentationPreference + StatsClass + Treatment, data = final,
##     atmean = F, robust = F)
##
## Marginal Effects:
##                                      dF/dx Std. Err.       z     P>|z|
## PollTrustDisagree                  -0.202659  0.066874 -3.0305 0.0024417
## PollTrustNeither Agree nor Disagree -0.029418  0.088468 -0.3325 0.7394935
## HorseRaceExposureOccasionally        0.126011  0.062599  2.0130 0.0441171
## HorseRaceExposureOften               0.044092  0.067699  0.6513 0.5148609
## PresentationPreferenceText           0.032503  0.058572  0.5549 0.5789457
## StatsClassYes                       -0.043860  0.056796 -0.7722 0.4399728
## Treatment2                           0.193233  0.052637  3.6710 0.0002416
## Treatment3                           0.025062  0.059216  0.4232 0.6721260
##
## PollTrustDisagree                  **
## PollTrustNeither Agree nor Disagree
## HorseRaceExposureOccasionally      *
## HorseRaceExposureOften
## PresentationPreferenceText
## StatsClassYes
## Treatment2                         ***
## Treatment3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
## [1] "PollTrustDisagree"
## [2] "PollTrustNeither Agree nor Disagree"
## [3] "HorseRaceExposureOccasionally"
## [4] "HorseRaceExposureOften"
## [5] "PresentationPreferenceText"
## [6] "StatsClassYes"
## [7] "Treatment2"
## [8] "Treatment3"
```

```
#Compare them by converting them to dataframes and then subtracting the Std..Err. vectors
robust <- data.frame(probeMEpretreatment$mfxest)
notRobust <- data.frame(probeMEpretreatmentNotRobust$mfxest)
difference <- robust$Std..Err.-notRobust$Std..Err.
#Average differences
mean(difference)
```

```
## [1] -0.001041118
```

```
median(difference)
```

```
## [1] -0.0005983148
```

```
#Add Education analysis
finalEdu <- final
finalEdu$EduLevel <- as.character(finalEdu$EduLevel)
finalEdu$EduLevel[finalEdu$EduLevel=="Completed trade/vocational school"] <- "Before college"
finalEdu$EduLevel[finalEdu$EduLevel=="Some HS or less"] <- "Before college"
finalEdu$EduLevel[finalEdu$EduLevel=="HS Graduate/GED completed"] <- "Before college"
finalEdu$EduLevel[finalEdu$EduLevel=="Graduated college"] <- "Post-college"
finalEdu$EduLevel[finalEdu$EduLevel=="Post-graduate or more"] <- "Post-college"
finalEdu$EduLevel <- as.factor(finalEdu$EduLevel)
summary(finalEdu$EduLevel)
```

```
## Before college   Post-college   Some college
##             31             65            119
```

```
#Marginal effects
probeMEpretreatmentEdu <- probitmfx(as.ordered(Likelihood) ~ EduLevel+PollTrust+HorseRaceExposure+
                                      PresentationPreference+StatsClass+Treatment,
                                    data=finalEdu, atmean=F, robust=T)
probeMEpretreatmentEdu
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ EduLevel + PollTrust +
##     HorseRaceExposure + PresentationPreference + StatsClass +
##     Treatment, data = finalEdu, atmean = F, robust = T)
##
## Marginal Effects:
##                                        dF/dx  Std. Err.       z
## EduLevelPost-college                0.0029566  0.0957135  0.0309
## EduLevelSome college               -0.0335724  0.0875752 -0.3834
## PollTrustDisagree                  -0.1987519  0.0660684 -3.0083
## PollTrustNeither Agree nor Disagree -0.0277481  0.0822093 -0.3375
## HorseRaceExposureOccasionally       0.1284187  0.0627319  2.0471
## HorseRaceExposureOften              0.0478337  0.0682219  0.7011
## PresentationPreferenceText          0.0283822  0.0600356  0.4728
## StatsClassYes                      -0.0369665  0.0622689 -0.5937
## Treatment2                          0.1932427  0.0517961  3.7308
## Treatment3                          0.0249352  0.0589761  0.4228
##                                       P>|z|
## EduLevelPost-college               0.9753575
## EduLevelSome college               0.7014564
## PollTrustDisagree                  0.0026273 **
## PollTrustNeither Agree nor Disagree 0.7357175
## HorseRaceExposureOccasionally      0.0406481 *
## HorseRaceExposureOften             0.4832104
## PresentationPreferenceText         0.6363875
## StatsClassYes                      0.5527403
```

```
## Treatment2                           0.0001908 ***
## Treatment3                           0.6724405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
##  [1] "EduLevelPost-college"
##  [2] "EduLevelSome college"
##  [3] "PollTrustDisagree"
##  [4] "PollTrustNeither Agree nor Disagree"
##  [5] "HorseRaceExposureOccasionally"
##  [6] "HorseRaceExposureOften"
##  [7] "PresentationPreferenceText"
##  [8] "StatsClassYes"
##  [9] "Treatment2"
## [10] "Treatment3"
```

```
#Add Gender
final$Gender <- as.character(final$Gender)
summary(final$Gender)
```

```
##    Length     Class      Mode
##       215 character character
```

```
final$Gender[final$Gender=="Other/Prefer not to respond"] <- NA
probeMEpretreatmentEduG <- probitmfx(as.ordered(Likelihood) ~
                                       Gender+EduLevel+PollTrust+HorseRaceExposure+
                                       PresentationPreference+StatsClass+Treatment,
                                     data=finalEdu, atmean=F, robust=T)
probeMEpretreatmentEduG
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ Gender + EduLevel +
##     PollTrust + HorseRaceExposure + PresentationPreference +
##     StatsClass + Treatment, data = finalEdu, atmean = F, robust = T)
##
## Marginal Effects:
##                                       dF/dx  Std. Err.        z
## GenderMale                        -0.0500398  0.0566988  -0.8826
## GenderOther/Prefer not to respond -0.7916556  0.0260568 -30.3819
## EduLevelPost-college              -0.0090686  0.0970645  -0.0934
## EduLevelSome college              -0.0303004  0.0867544  -0.3493
## PollTrustDisagree                 -0.2082098  0.0670123  -3.1070
## PollTrustNeither Agree nor Disagree  0.0081446  0.0791685   0.1029
## HorseRaceExposureOccasionally        0.1043805  0.0639675   1.6318
## HorseRaceExposureOften               0.0379407  0.0675837   0.5614
## PresentationPreferenceText           0.0387818  0.0590933   0.6563
## StatsClassYes                     -0.0229612  0.0623994  -0.3680
## Treatment2                           0.2051667  0.0500800   4.0968
## Treatment3                           0.0264982  0.0572553   0.4628
##                                      P>|z|
## GenderMale                         0.37748
```

```
## GenderOther/Prefer not to respond   < 2.2e-16 ***
## EduLevelPost-college                  0.92556
## EduLevelSome college                  0.72689
## PollTrustDisagree                     0.00189 **
## PollTrustNeither Agree nor Disagree   0.91806
## HorseRaceExposureOccasionally         0.10273
## HorseRaceExposureOften                0.57453
## PresentationPreferenceText            0.51164
## StatsClassYes                         0.71289
## Treatment2                          4.189e-05 ***
## Treatment3                            0.64350
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
##  [1] "GenderMale"
##  [2] "GenderOther/Prefer not to respond"
##  [3] "EduLevelPost-college"
##  [4] "EduLevelSome college"
##  [5] "PollTrustDisagree"
##  [6] "PollTrustNeither Agree nor Disagree"
##  [7] "HorseRaceExposureOccasionally"
##  [8] "HorseRaceExposureOften"
##  [9] "PresentationPreferenceText"
## [10] "StatsClassYes"
## [11] "Treatment2"
## [12] "Treatment3"
```

```
#From general
#Drop trust
probeMEpretreatmentMinusTrust <- probitmfx(as.ordered(Likelihood) ~ HorseRaceExposure+
                                             PresentationPreference+StatsClass+Treatment,
                                           data=final, atmean=F, robust=T)
probeMEpretreatmentMinusTrust
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ HorseRaceExposure +
##     PresentationPreference + StatsClass + Treatment, data = final,
##     atmean = F, robust = T)
##
## Marginal Effects:
##                                  dF/dx Std. Err.       z    P>|z|
## HorseRaceExposureOccasionally  0.144389  0.063788  2.2636 0.023600 *
## HorseRaceExposureOften         0.090247  0.064782  1.3931 0.163593
## PresentationPreferenceText     0.037646  0.060944  0.6177 0.536764
## StatsClassYes                 -0.038956  0.058396 -0.6671 0.504701
## Treatment2                     0.175866  0.053975  3.2583 0.001121 **
## Treatment3                     0.010054  0.060828  0.1653 0.868718
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
```

```
## [1] "HorseRaceExposureOccasionally" "HorseRaceExposureOften"
## [3] "PresentationPreferenceText"    "StatsClassYes"
## [5] "Treatment2"                    "Treatment3"
```

```
#Trust instead of exposure
probeMEminusExposure <- probitmfx(as.ordered(Likelihood) ~
                                  PollTrust+PresentationPreference+
                                  StatsClass+Treatment, data=final, atmean=F, robust=T)
probeMEminusExposure
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ PollTrust + PresentationPreference +
##     StatsClass + Treatment, data = final, atmean = F, robust = T)
##
## Marginal Effects:
##                                       dF/dx Std. Err.       z    P>|z|
## PollTrustDisagree                   -0.207346  0.066041 -3.1397 0.001691
## PollTrustNeither Agree nor Disagree -0.036609  0.083337 -0.4393 0.660448
## PresentationPreferenceText           0.024210  0.058774  0.4119 0.680404
## StatsClassYes                       -0.034187  0.058404 -0.5854 0.558305
## Treatment2                           0.196461  0.050997  3.8524 0.000117
## Treatment3                           0.028794  0.059388  0.4849 0.627781
##
## PollTrustDisagree                   **
## PollTrustNeither Agree nor Disagree
## PresentationPreferenceText
## StatsClassYes
## Treatment2                          ***
## Treatment3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
## [1] "PollTrustDisagree"
## [2] "PollTrustNeither Agree nor Disagree"
## [3] "PresentationPreferenceText"
## [4] "StatsClassYes"
## [5] "Treatment2"
## [6] "Treatment3"
```

```
#Neither
probeMEneither <- probitmfx(as.ordered(Likelihood) ~
                              PresentationPreference+StatsClass+Treatment,
                            data=final, atmean=F, robust=T)
probeMEneither
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ PresentationPreference +
##     StatsClass + Treatment, data = final, atmean = F, robust = T)
##
## Marginal Effects:
##                                 dF/dx Std. Err.       z   P>|z|
```

```
## PresentationPreferenceText  0.037484  0.060262  0.6220 0.53393
## StatsClassYes              -0.026538  0.059009 -0.4497 0.65290
## Treatment2                  0.172546  0.054643  3.1577 0.00159 **
## Treatment3                  0.012167  0.061949  0.1964 0.84429
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
## [1] "PresentationPreferenceText" "StatsClassYes"
## [3] "Treatment2"                 "Treatment3"
```

```
#Analyze surprising result of stats class
finalStats <- filter(final, !is.na(final$StatsClassAgo))
nrow(finalStats)
```

```
## [1] 153
```

```
finalStats$StatsClassAgo <- as.character(finalStats$StatsClassAgo)
finalStats$StatsClassAgo[finalStats$StatsClassAgo=="6+ years ago"] <- "3-5 years ago"
#Group not-recent
finalStats$StatsClassAgo[finalStats$StatsClassAgo=="Less than a year ago"] <- 1
finalStats$StatsClassAgo[finalStats$StatsClassAgo=="1-3 years ago"] <- 2
finalStats$StatsClassAgo[finalStats$StatsClassAgo=="3-5 years ago"] <- 3
finalStats$StatsClassAgo <- as.factor(finalStats$StatsClassAgo)
summary(finalStats$StatsClassAgo)
```

```
##  1  2  3
## 39 55 59
```

```
#New model replacing stats class with this new factor
probeMEduration <- probitmfx(as.ordered(Likelihood) ~
                             HorseRaceExposure+PollTrust+PresentationPreference
                            +StatsClassAgo+Treatment, data=finalStats, atmean=F, robust=T)
probeMEduration
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ HorseRaceExposure +
##     PollTrust + PresentationPreference + StatsClassAgo + Treatment,
##     data = finalStats, atmean = F, robust = T)
##
## Marginal Effects:
##                                       dF/dx  Std. Err.       z    P>|z|
## HorseRaceExposureOccasionally        0.2096432  0.0702413  2.9846 0.002839
## HorseRaceExposureOften               0.1523638  0.0746401  2.0413 0.041220
## PollTrustDisagree                   -0.2126723  0.0741197 -2.8693 0.004114
## PollTrustNeither Agree nor Disagree  0.1547511  0.0728513  2.1242 0.033653
## PresentationPreferenceText           0.0808965  0.0658450  1.2286 0.219226
## StatsClassAgo2                      -0.0408687  0.0858499 -0.4760 0.634040
## StatsClassAgo3                      -0.0067257  0.0886815 -0.0758 0.939545
## Treatment2                           0.0956562  0.0685001  1.3964 0.162583
## Treatment3                          -0.0399409  0.0740733 -0.5392 0.589744
```

```
##
## HorseRaceExposureOccasionally       **
## HorseRaceExposureOften              *
## PollTrustDisagree                   **
## PollTrustNeither Agree nor Disagree *
## PresentationPreferenceText
## StatsClassAgo2
## StatsClassAgo3
## Treatment2
## Treatment3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
## [1] "HorseRaceExposureOccasionally"
## [2] "HorseRaceExposureOften"
## [3] "PollTrustDisagree"
## [4] "PollTrustNeither Agree nor Disagree"
## [5] "PresentationPreferenceText"
## [6] "StatsClassAgo2"
## [7] "StatsClassAgo3"
## [8] "Treatment2"
## [9] "Treatment3"
```

```
#Take out both other factors
probeMEdurationNo <- probitmfx(as.ordered(Likelihood) ~
                                 PresentationPreference+StatsClassAgo+Treatment,
                               data=finalStats, atmean=F, robust=T)
probeMEdurationNo
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ PresentationPreference +
##     StatsClassAgo + Treatment, data = finalStats, atmean = F,
##     robust = T)
##
## Marginal Effects:
##                                 dF/dx  Std. Err.       z  P>|z|
## PresentationPreferenceText  0.0862138  0.0694233  1.2419 0.2143
## StatsClassAgo2             -0.0540920  0.0906335 -0.5968 0.5506
## StatsClassAgo3              0.0061172  0.0909569  0.0673 0.9464
## Treatment2                  0.1059824  0.0739385  1.4334 0.1517
## Treatment3                 -0.0176938  0.0780701 -0.2266 0.8207
##
## dF/dx is for discrete change for the following variables:
##
## [1] "PresentationPreferenceText" "StatsClassAgo2"
## [3] "StatsClassAgo3"             "Treatment2"
## [5] "Treatment3"
```

```
#Just treatment
probeMEdurationSimple <- probitmfx(as.ordered(Likelihood) ~
                                     Treatment, data=finalStats, atmean=F, robust=T)
probeMEdurationSimple
```

```
## Call:
## probitmfx(formula = as.ordered(Likelihood) ~ Treatment, data = finalStats,
##     atmean = F, robust = T)
##
## Marginal Effects:
##                 dF/dx  Std. Err.       z  P>|z|
## Treatment2  0.1188321  0.0751319  1.5816 0.1137
## Treatment3 -0.0046049  0.0784025 -0.0587 0.9532
##
## dF/dx is for discrete change for the following variables:
##
## [1] "Treatment2" "Treatment3"
```

```
#Analyzing the trend question via ordered probit
finalTrend <- subset(final, !is.na(final$Trend))
finalTrend$Trend <- as.factor(finalTrend$Trend)
probeTrend <- probitmfx(Trend~
                          PollTrust+HorseRaceExposure,
                        data=finalTrend, atmean=F, robust=T)
probeTrend
```

```
## Call:
## probitmfx(formula = Trend ~ PollTrust + HorseRaceExposure, data = finalTrend,
##     atmean = F, robust = T)
##
## Marginal Effects:
##                                         dF/dx  Std. Err.       z  P>|z|
## PollTrustDisagree                  -0.0024159  0.0613097 -0.0394 0.9686
## PollTrustNeither Agree nor Disagree  0.1040355  0.0640814  1.6235 0.1045
## HorseRaceExposureOccasionally      -0.0228489  0.0777822 -0.2938 0.7689
## HorseRaceExposureOften              0.0182399  0.0763585  0.2389 0.8112
##
## dF/dx is for discrete change for the following variables:
##
## [1] "PollTrustDisagree"
## [2] "PollTrustNeither Agree nor Disagree"
## [3] "HorseRaceExposureOccasionally"
## [4] "HorseRaceExposureOften"
```

```
#Export to CSV
finalStata <- final
#Convert poll trust to numbers
finalStata$PollTrust <- as.character(finalStata$PollTrust)
finalStata$PollTrust[finalStata$PollTrust=="Agree"] <- 1
finalStata$PollTrust[finalStata$PollTrust=="Disagree"] <- 2
finalStata$PollTrust[finalStata$PollTrust=="Neither Agree nor Disagree"] <- 3
finalStata$PollTrust <- as.numeric(finalStata$PollTrust)
#Presentation preference
finalStata$PresentationPreference <- as.character(finalStata$PresentationPreference)
finalStata$PresentationPreference[finalStata$PresentationPreference=="Graphics"] <- 1
finalStata$PresentationPreference[finalStata$PresentationPreference=="Text"] <- 2
finalStata$PresentationPreference <- as.numeric(finalStata$PresentationPreference)
#Horse race exposure
```

```
finalStata$HorseRaceExposure <- as.character(finalStata$HorseRaceExposure)
finalStata$HorseRaceExposure[finalStata$HorseRaceExposure=="Never"] <- 1
finalStata$HorseRaceExposure[finalStata$HorseRaceExposure=="Occasionally"] <- 2
finalStata$HorseRaceExposure[finalStata$HorseRaceExposure=="Often"] <- 3
finalStata$HorseRaceExposure <- as.numeric(finalStata$HorseRaceExposure)
#Stats class
finalStata$StatsClass <- as.character(finalStata$StatsClass)
finalStata$StatsClass[finalStata$StatsClass=="No"] <- 1
finalStata$StatsClass[finalStata$StatsClass=="Yes"] <- 2
finalStata$StatsClass <- as.numeric(finalStata$StatsClass)
#Write the csv--saved time bc Treatment and Likelihood already numeric
write.csv(file="stataExport.csv", x=finalStata)
```

Bayesian ordered probit:

```
bayesData <- read.csv(file="gistStuff.csv", h=T) #Formerly known as `final'
set.seed(12) #reproducibility
bayesData$Treatment <- as.factor(bayesData$Treatment)

library(Zelig)
```

```
## Loading required package: boot
## ZELIG (Versions 4.2-1, built: 2013-09-12)
##
## +-----------------------------------------------------------------+
## |  Please refer to http://gking.harvard.edu/zelig for full        |
## |  documentation or help.zelig() for help with commands and       |
## |  models support by Zelig.                                       |
## |                                                                 |
## |  Zelig project citations:                                       |
## |    Kosuke Imai, Gary King, and Olivia Lau.  (2009).             |
## |    ``Zelig: Everyone's Statistical Software,''                  |
## |    http://gking.harvard.edu/zelig                               |
## |   and                                                           |
## |    Kosuke Imai, Gary King, and Olivia Lau. (2008).              |
## |    ``Toward A Common Framework for Statistical Analysis         |
## |    and Development,'' Journal of Computational and              |
## |    Graphical Statistics, Vol. 17, No. 4 (December)              |
## |    pp. 892-913.                                                 |
## |                                                                 |
## |   To cite individual Zelig models, please use the citation      |
## |   format printed with each model run and in the documentation.  |
## +-----------------------------------------------------------------+
##
##
##
## Attaching package: 'Zelig'
##
## The following objects are masked from 'package:dplyr':
##
##     combine, summarize
##
## The following object is masked from 'package:utils':
```

```
##
##     cite
```

```
bayesianOProbit <- zelig(Likelihood~
                           Treatment+HorseRaceExposure+PollTrust+
                           PresentationPreference+StatsClass,
                         model="oprobit.bayes", data=bayesData)
```

```
## Loading required package: MCMCpack
## Loading required package: coda
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
## ## Copyright (C) 2003-2015 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
## ##
## ## Support provided by the U.S. National Science Foundation
## ## (Grants SES-0350646 and SES-0350613)
## ##

## Warning in model.response(mf, "numeric"): using type = "numeric" with a
## factor response will be ignored

##
##
## MCMCoprobit iteration 1 of 11000
## beta =
##    1.06671
##    0.46111
##   -0.26236
##    0.42473
##    0.11529
##   -0.71386
##   -0.07747
##    0.24379
##   -0.53734
## Metropolis acceptance rate for gamma = 1.00000
##
##
##
## MCMCoprobit iteration 1101 of 11000
## beta =
##    0.90430
##    0.46835
##    0.35711
##    0.12385
##    0.14530
##   -0.44675
##   -0.50620
##    0.00869
##   -0.46750
## Metropolis acceptance rate for gamma = 0.94460
##
##
##
```

```
## MCMCoprobit iteration 2201 of 11000
## beta =
##    1.13981
##    0.78015
##   -0.04366
##    0.14122
##   -0.14166
##   -0.67573
##   -0.63748
##    0.32322
##   -0.44804
## Metropolis acceptance rate for gamma = 0.94684
##
##
##
## MCMCoprobit iteration 3301 of 11000
## beta =
##    1.23530
##    0.52936
##    0.08648
##    0.48580
##    0.13062
##   -0.59380
##   -0.31150
##    0.13866
##   -0.54487
## Metropolis acceptance rate for gamma = 0.95062
##
##
##
## MCMCoprobit iteration 4401 of 11000
## beta =
##    1.16017
##    0.53098
##   -0.15350
##    0.34099
##    0.09255
##   -0.89772
##   -0.31492
##   -0.14480
##   -0.23379
## Metropolis acceptance rate for gamma = 0.95001
##
##
##
## MCMCoprobit iteration 5501 of 11000
## beta =
##    1.08573
##    0.55981
##    0.02824
##    0.13557
##    0.12171
##   -0.42040
##   -0.34573
```

```
##    0.05773
##   -0.34496
## Metropolis acceptance rate for gamma = 0.94965
##
##
##
## MCMCoprobit iteration 6601 of 11000
## beta =
##    0.79545
##    0.66373
##    0.04377
##    0.53352
##    0.33265
##   -0.52115
##    0.18370
##    0.02738
##   -0.24921
## Metropolis acceptance rate for gamma = 0.94698
##
##
##
## MCMCoprobit iteration 7701 of 11000
## beta =
##    0.79936
##    0.49139
##   -0.20176
##    0.46167
##    0.39657
##   -0.35396
##   -0.26755
##    0.06211
##   -0.21436
## Metropolis acceptance rate for gamma = 0.94585
##
##
##
## MCMCoprobit iteration 8801 of 11000
## beta =
##    0.66263
##    0.47887
##   -0.05481
##    0.19915
##    0.05342
##   -0.46416
##   -0.24920
##    0.23965
##   -0.07667
## Metropolis acceptance rate for gamma = 0.94444
##
##
##
## MCMCoprobit iteration 9901 of 11000
## beta =
##    0.91496
```

```
##    0.59127
##   -0.18838
##    0.31904
##    0.23439
##   -0.59249
##    0.09782
##    0.10146
##   -0.18158
## Metropolis acceptance rate for gamma = 0.94364
##
##
##
## @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
## The Metropolis acceptance rate for beta was 0.94455
## @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
##
##
##  How to cite this model in Zelig:
##   Ben Goodrich, and Ying Lu. 2015.
##   "oprobit.bayes: Bayesian Probit Regression for Dichotomous Dependent Variables"
##   in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
##   http://gking.harvard.edu/zelig
##
```

```
#Check for convergence
heidel.diag(bayesianOProbit$result$coefficients) #Yay
```

```
##
##                                      Stationarity start     p-value
##                                      test         iteration
## (Intercept)                          passed       1         0.6555
## Treatment2                           passed       1         0.0702
## Treatment3                           passed       1         0.7917
## HorseRaceExposureOccasionally        passed       1         0.9792
## HorseRaceExposureOften               passed       1         0.5017
## PollTrustDisagree                    passed       1         0.1259
## PollTrustNeither Agree nor Disagree passed       1         0.7228
## PresentationPreferenceText           passed       1         0.7162
## StatsClassYes                        passed       1         0.5631
## gamma2                               passed       1         0.8924
##
##                                      Halfwidth Mean    Halfwidth
##                                      test
## (Intercept)                          passed     0.9822 0.01561
## Treatment2                           passed     0.5134 0.00489
## Treatment3                           passed    -0.0962 0.00474
## HorseRaceExposureOccasionally        passed     0.2496 0.00522
## HorseRaceExposureOften               passed     0.1066 0.00536
## PollTrustDisagree                    passed    -0.5583 0.00525
## PollTrustNeither Agree nor Disagree passed    -0.3652 0.00566
## PresentationPreferenceText           passed     0.1567 0.00422
## StatsClassYes                        passed    -0.2772 0.00440
## gamma2                               passed     1.5616 0.03482
```

```r
raftery.diag(bayesianOProbit$result$coefficients) #Yay
```

```
##
## Quantile (q) = 0.025
## Accuracy (r) = +/- 0.005
## Probability (s) = 0.95
##
##                                     Burn-in  Total Lower bound
##                                     (M)      (N)   (Nmin)
##  (Intercept)                        3        4197  3746
##  Treatment2                         3        4061  3746
##  Treatment3                         3        4061  3746
##  HorseRaceExposureOccasionally      3        4163  3746
##  HorseRaceExposureOften             2        3994  3746
##  PollTrustDisagree                  3        4028  3746
##  PollTrustNeither Agree nor Disagree 2       3962  3746
##  PresentationPreferenceText         2        3994  3746
##  StatsClassYes                      3        4410  3746
##  gamma2                             60       63477 3746
##
##  Dependence
##  factor (I)
##   1.12
##   1.08
##   1.08
##   1.11
##   1.07
##   1.08
##   1.06
##   1.07
##   1.18
##  16.90
```

```r
#Summarize results
summary(bayesianOProbit)
```

```
##
## Call: zelig(formula = Likelihood ~ Treatment + HorseRaceExposure +
##     PollTrust + PresentationPreference + StatsClass, model = "oprobit.bayes",
##     data = bayesData)
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
##  Mean, standard deviation, and quantiles for marginal posterior distributions.
##                                       Mean     SD    2.5%     50%   97.5%
## (Intercept)                         0.9822 0.2568  0.4818  0.9818  1.4766
## Treatment2                          0.5134 0.1906  0.1381  0.5149  0.8878
## Treatment3                         -0.0962 0.1946 -0.4770 -0.0974  0.2888
## HorseRaceExposureOccasionally       0.2496 0.2091 -0.1622  0.2485  0.6566
```

```
## HorseRaceExposureOften               0.1066 0.2181 -0.3258  0.1062  0.5310
## PollTrustDisagree                   -0.5583 0.1788 -0.9134 -0.5581 -0.2088
## PollTrustNeither Agree nor Disagree -0.3652 0.2286 -0.8071 -0.3657  0.0829
## PresentationPreferenceText           0.1567 0.1760 -0.1904  0.1566  0.5057
## StatsClassYes                       -0.2772 0.1751 -0.6220 -0.2769  0.0700
## gamma2                               1.5616 0.1186  1.3371  1.5647  1.8052
```