reputation_analyses

Contents

1 Study 1 - Who Lied about their Poor Performance?

```
# read in data
lap_data = read_csv("../data/study1_lying_about_poor_perf/data_study1.csv")
```

```r
# data cleaning
lap_data = lap_data %>%
  mutate(Script = as.factor(Script)) %>%
  mutate(Sex = ifelse(Sex == "0",
                      "female",
                      "male"),
         WhoLied = ifelse(WhoLied == "0",
                          "Intrinsic",
                          "Reputational")) %>% # recode data
  mutate(AgeGroup = ifelse((Age_Yrs == 4|Age_Yrs == 5), "4-5",
                           ifelse((Age_Yrs == 6|Age_Yrs == 7), "6-7",
                           ifelse((Age_Yrs == 8|Age_Yrs ==9), "8-9", NA)))) %>%
    mutate(AgeGroup = as.factor(AgeGroup)) %>%
    mutate(WhoLied = as.factor(as.character(WhoLied))) %>%
    mutate(Sex = as.factor(as.character(Sex)))
```

1.1 Summary

```r
 # generates table with percentage breakdown of children who identified either
# the reputational or intrinsic character as the one who lied about their poor performance
lap_summary = lap_data %>%
  group_by(AgeGroup) %>%
  summarise(percent_intrinsic = (sum(WhoLied == "Intrinsic")/length(Sub_ID))*100,
            number_intrinsic = (sum(WhoLied == "Intrinsic")),
            percent_reputational = (sum(WhoLied == "Reputational")/length(Sub_ID))*100,
            number_reputational = (sum(WhoLied == "Reputational")),
            total_n = (sum(WhoLied == "Intrinsic"|WhoLied == "Reputational")))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
# generate table
lap_summary %>%
  kable(caption = "Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 1: Summary

AgeGroup	percent_intrinsic	number_intrinsic	percent_reputational	number_reputational	total_n
4-5	56.250	18	43.750	14	32
6-7	28.125	9	71.875	23	32
8-9	6.250	2	93.750	30	32

1.2 Demographics

```r
# gender
# create dataframe with percentage breakdown for gender of participants
lap_gender_summary = lap_data %>%
  group_by(AgeGroup, Sex) %>%
  summarise(n = length(Sub_ID)) %>%
              mutate(percentage = (n / sum(n))*100)
```

```
## `summarise()` regrouping output by 'AgeGroup' (override with `.groups` argument)
```

```r
# generate table
lap_gender_summary %>%
  kable(caption = "Gender Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 2: Gender Summary

AgeGroup	Sex	n	percentage
4-5	female	23	71.875
4-5	male	9	28.125
6-7	female	18	56.250
6-7	male	14	43.750
8-9	female	18	56.250
8-9	male	14	43.750

```r
# age
# create dataframe with summary stats (mean and SD) for age of participants
lap_age_summary <- lap_data %>%
  group_by(AgeGroup) %>%
  summarise(mean_age = mean(Age_Ex),
            SD_age = sd(Age_Ex))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
# generate table
lap_age_summary %>%
  kable(caption = "Age Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 3: Age Summary

AgeGroup	mean_age	SD_age
4-5	4.923785	0.5643043
6-7	6.797309	0.5004998
8-9	8.789844	0.5637009

1.3 Confirmatory Analyses

```r
# generates logistic regression model examining the effect of age (entered as a continuous
# predictor) on whether children identified the character with
# reputational or intrinsic concerns as the one who would lie about their poor performance
lap_logit_model = glm(WhoLied ~ Age_Ex,
                      data = lap_data,
                      family = binomial(link = "logit"))

# generates summary of logistic regression model
summary(lap_logit_model)
```

```
##
## Call:
## glm(formula = WhoLied ~ Age_Ex, family = binomial(link = "logit"),
##     data = lap_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2412  -0.9495   0.4681   0.7751   1.5501
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.5555     1.1025  -3.225  0.00126 **
## Age_Ex        0.6770     0.1737   3.898 9.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 117.623  on 95  degrees of freedom
## Residual deviance:  97.951  on 94  degrees of freedom
## AIC: 101.95
##
## Number of Fisher Scoring iterations: 4
```

```r
# computes Wald chi-square value for logistic regression model above
lap_wald_test = wald.test(b = coef(lap_logit_model),
                          Sigma = vcov(lap_logit_model),
                          Terms = 2,
                          verbose = TRUE)

# prints results of Wald chi-square test
print(lap_wald_test, digits = 5)
```

```
## Wald test:
## ----------
##
## Coefficients:
## (Intercept)      Age_Ex
##    -3.55555     0.67704
##
```

```
## Var-cov matrix of the coefficients:
##             (Intercept) Age_Ex
## (Intercept)  1.215470   -0.186581
## Age_Ex      -0.186581    0.030166
##
## Test-design matrix:
##    (Intercept) Age_Ex
## L1           0      1
##
## Positions of tested coefficients in the vector of coefficients: 2
##
## H0:  Age_Ex = 0
##
## Chi-squared test:
## X2 = 15.196, df = 1, P(> X2) = 9.6927e-05
```

```r
# generates binomial tests for each age group that test whether children chose the
# reputational character more often than would be predicted by chance
lap_binomial_tests = lap_summary %>%
  group_by(AgeGroup) %>%
  do(test = tidy(binom.test(sum(.$number_reputational),
                     sum(.$total_n),
                     p = .5,
                     conf.level = 0.95)))

# generates dataframe with results of binomial tests described above
lap_binomial_test_summary = lap_binomial_tests$test %>%
  bind_rows %>%
  bind_cols(lap_binomial_tests[1], .)

# generates table of binomial test estimates
lap_binomial_test_summary %>%
  kable(caption = "Binomial Tests") %>%
  kable_styling(bootstrap_options = "striped",
              full_width = F,
              latex_options = c("hold_position", "scale_down"))
```

Table 4: Binomial Tests

AgeGroup	estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
4-5	0.43750	14	0.5966149	32	0.2636381	0.6233743	Exact binomial test	two.sided
6-7	0.71875	23	0.0200616	32	0.5325289	0.8625431	Exact binomial test	two.sided
8-9	0.93750	30	0.0000002	32	0.7919306	0.9923393	Exact binomial test	two.sided

2 Study 2 - Who Lied about their Successful Performance?

```r
# read in data
las_data = read_csv("../data/study2_lying_about_good_perf/data_study2.csv")
```

```r
# data cleaning
las_data = las_data %>%
  mutate(Script = as.factor(Script)) %>%
  mutate(Sex = ifelse(Sex == "0", "female", "male"),
         WhoLied = ifelse(WhoLied == "0", "Reputational", "Intrinsic")) %>% # recode data
  mutate(AgeGroup = ifelse((Age_Yrs == 4|Age_Yrs == 5), "4-5",
                           ifelse((Age_Yrs == 6|Age_Yrs == 7), "6-7",
                           ifelse((Age_Yrs == 8|Age_Yrs ==9), "8-9", NA)))) %>%
    mutate(AgeGroup = as.factor(AgeGroup)) %>%
    mutate(WhoLied = as.factor(as.character(WhoLied))) %>%
    mutate(Sex = as.factor(as.character(Sex)))
```

2.1 Summary

```r
# generates table with percentage breakdown of children who identified either the
# reputational or intrinsic character as the one who lied about their poor performance
las_summary = las_data %>%
  group_by(AgeGroup) %>%
  summarise(percent_intrinsic = (sum(WhoLied == "Intrinsic")/length(Sub_ID))*100,
            number_intrinsic = (sum(WhoLied == "Intrinsic")),
            percent_reputational = (sum(WhoLied == "Reputational")/length(Sub_ID))*100,
            number_reputational = (sum(WhoLied == "Reputational")),
            total_n = (sum(WhoLied == "Intrinsic"|WhoLied == "Reputational")))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
# generate table
las_summary %>%
  kable(caption = "Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 5: Summary

AgeGroup	percent_intrinsic	number_intrinsic	percent_reputational	number_reputational	total_n
4-5	53.125	17	46.875	15	32
6-7	56.250	18	43.750	14	32
8-9	84.375	27	15.625	5	32

2.2 Demographics

```r
# gender

# creates dataframe with percentage breakdown for gender of participants
las_gender_summary = las_data %>%
  group_by(AgeGroup, Sex) %>%
```

```r
  summarise(n = length(Sub_ID)) %>%
            mutate(percentage = (n / sum(n))*100)
```

```
## `summarise()` regrouping output by 'AgeGroup' (override with `.groups` argument)
```

```r
# generates table
las_gender_summary %>%
  kable(caption = "Gender Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 6: Gender Summary

AgeGroup	Sex	n	percentage
4-5	female	11	34.375
4-5	male	21	65.625
6-7	female	13	40.625
6-7	male	19	59.375
8-9	female	15	46.875
8-9	male	17	53.125

```r
# age

# creates dataframe with summary stats (mean and SD) for age of participants
las_age_summary = las_data %>%
  group_by(AgeGroup) %>%
  summarise(mean_age = mean(Age_Ex),
            SD_age = sd(Age_Ex))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
# generates table
las_age_summary %>%
  kable(caption = "Age Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 7: Age Summary

AgeGroup	mean_age	SD_age
4-5	5.021441	0.6068109
6-7	6.804774	0.5926409
8-9	8.940365	0.5656366

2.3 Confirmatory Analyses

```r
# generates logistic regression model examining the effect of age (entered as a continuous
# predictor) on whether children identified the character with
# reputational or intrinsic concerns as the one who would lie about their poor performance
las_logit_model = glm(WhoLied ~ Age_Ex,
                      data = las_data,
                      family = binomial(link = "logit"))

 # generates summary of logistic regression model
summary(las_logit_model)
```

```
##
## Call:
## glm(formula = WhoLied ~ Age_Ex, family = binomial(link = "logit"),
##     data = las_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4064  -0.8961  -0.6351   1.1468   1.9561
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.2042     0.9533   2.312  0.02077 *
## Age_Ex       -0.4157     0.1409  -2.951  0.00316 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 124.80  on 95  degrees of freedom
## Residual deviance: 115.03  on 94  degrees of freedom
## AIC: 119.03
##
## Number of Fisher Scoring iterations: 4
```

```r
# computes Wald chi-square value for logistic regression model described above
las_wald_test = wald.test(b = coef(las_logit_model),
                          Sigma = vcov(las_logit_model),
                          Terms = 2,
                          verbose = TRUE)

# prints results of Wald chi-square test
print(las_wald_test, digits = 5)
```

```
## Wald test:
## ----------
##
## Coefficients:
## (Intercept)      Age_Ex
##     2.20417    -0.41573
##
## Var-cov matrix of the coefficients:
##             (Intercept) Age_Ex
```

```
## (Intercept)  0.90886    -0.13050
## Age_Ex      -0.13050     0.01984
##
## Test-design matrix:
##    (Intercept) Age_Ex
## L1           0      1
##
## Positions of tested coefficients in the vector of coefficients: 2
##
## H0:  Age_Ex = 0
##
## Chi-squared test:
## X2 = 8.7111, df = 1, P(> X2) = 0.0031627
```

```
# generates binomial tests for each age group that test whether children chose the
# reputational character more often than would be predicted by chance
las_binomial_tests = las_summary %>%
  group_by(AgeGroup) %>%
  do(test = tidy(binom.test(sum(.$number_reputational),
                     sum(.$total_n),
                     p = .5,
                     conf.level = 0.95)))

# generates dataframe with results of binomial tests described above
las_binomial_test_summary = las_binomial_tests$test %>%
  bind_rows %>%
  bind_cols(las_binomial_tests[1], .)

# generates table of binomial test estimates
las_binomial_test_summary %>%
  kable(caption = "Binomial Tests") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = c("hold_position", "scale_down"))
```

Table 8: Binomial Tests

AgeGroup	estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
4-5	0.46875	15	0.8600501	32	0.2909398	0.6525632	Exact binomial test	two.sided
6-7	0.43750	14	0.5966149	32	0.2636381	0.6233743	Exact binomial test	two.sided
8-9	0.15625	5	0.0001131	32	0.0527506	0.3278788	Exact binomial test	two.sided

3 Study 3 - Who Sought Help Publicly?

```
# read in data
psh_data = read_csv("../data/study3_public_help_seeking/data_study3.csv")
```

```
# data cleaning
psh_data = psh_data %>%
  mutate(Script = as.factor(Script)) %>%
```

```
  mutate(Sex = ifelse(Sex == "0",
                      "female",
                      "male"),
         WhoAsked = ifelse(WhoAsked == "0",
                           "Reputational",
                           "Intrinsic")) %>% # recode data
  mutate(AgeGroup = ifelse((Age_Yrs == 4|Age_Yrs == 5), "4-5",
                           ifelse((Age_Yrs == 6|Age_Yrs == 7), "6-7",
                           ifelse((Age_Yrs == 8|Age_Yrs == 9), "8-9", NA)))) %>%
    mutate(AgeGroup = as.factor(AgeGroup)) %>%
    mutate(WhoAsked = as.factor(as.character(WhoAsked))) %>%
    mutate(Sex = as.factor(as.character(Sex)))
```

3.1 Summary

```
# generates table with percentage breakdown of children who identified either the
# reputational or intrinsic character as the one who publicly sought help
psh_summary = psh_data %>%
  group_by(AgeGroup) %>%
  summarise(percent_intrinsic = (sum(WhoAsked == "Intrinsic")/length(Sub_ID))*100,
            number_intrinsic = (sum(WhoAsked == "Intrinsic")),
            percent_reputational = (sum(WhoAsked == "Reputational")/length(Sub_ID))*100,
            number_reputational = (sum(WhoAsked == "Reputational")),
            total_n = (sum(WhoAsked == "Intrinsic"|WhoAsked == "Reputational")))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# generates table
psh_summary %>%
  kable(caption = "Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 9: Summary

AgeGroup	percent_intrinsic	number_intrinsic	percent_reputational	number_reputational	total_n
4-5	59.375	19	40.625	13	32
6-7	68.750	22	31.250	10	32
8-9	81.250	26	18.750	6	32

3.2 Demographics

```
# gender

# creates dataframe with percentage breakdown for gender of participants
psh_gender_summary = psh_data %>%
```

```r
  group_by(AgeGroup, Sex) %>%
  summarise(n = length(Sub_ID)) %>%
              mutate(percentage = (n / sum(n))*100)
```

```
## `summarise()` regrouping output by 'AgeGroup' (override with `.groups` argument)
```

```r
# generates table
psh_gender_summary %>%
  kable(caption = "Gender Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 10: Gender Summary

AgeGroup	Sex	n	percentage
4-5	female	21	65.625
4-5	male	11	34.375
6-7	female	18	56.250
6-7	male	14	43.750
8-9	female	15	46.875
8-9	male	17	53.125

```r
# age

# creates dataframe with summary stats (mean and SD) for age of participants
psh_age_summary = psh_data %>%
  group_by(AgeGroup) %>%
  summarise(mean_age = mean(Age_Ex),
            SD_age = sd(Age_Ex))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
# generates table
psh_age_summary %>%
  kable(caption = "Age Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 11: Age Summary

AgeGroup	mean_age	SD_age
4-5	4.951910	0.6556260
6-7	6.818924	0.6134484
8-9	8.961806	0.5731432

3.3 Confirmatory Analyses

```r
# generates logistic regression model examining the effect of age (entered as a continuous
# predictor) on whether children identified the character with
# reputational or intrinsic concerns as the one who would publicly seek help
psh_logit_model = glm(WhoAsked ~ Age_Ex,
                              data = psh_data,
                              family = binomial(link = "logit"))

# generates summary of logistic regression model
summary(psh_logit_model)
```

```
##
## Call:
## glm(formula = WhoAsked ~ Age_Ex, family = binomial(link = "logit"),
##     data = psh_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.159  -0.902  -0.656   1.250   1.945
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.2103     0.9222   1.312   0.1894
## Age_Ex       -0.3043     0.1363  -2.232   0.0256 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 117.62  on 95  degrees of freedom
## Residual deviance: 112.29  on 94  degrees of freedom
## AIC: 116.29
##
## Number of Fisher Scoring iterations: 4
```

```r
# computes Wald chi-square value for logistic regression model described above
psh_wald_test = wald.test(b = coef(psh_logit_model),
                                  Sigma = vcov(psh_logit_model), Terms = 2,
                                  verbose = TRUE)

 # prints results of Wald chi-square test
print(psh_wald_test, digits = 4)
```

```
## Wald test:
## ----------
##
## Coefficients:
## (Intercept)      Age_Ex
##      1.2103     -0.3043
##
## Var-cov matrix of the coefficients:
```

```
##              (Intercept) Age_Ex
## (Intercept)  0.85045    -0.12180
## Age_Ex      -0.12180     0.01859
##
## Test-design matrix:
##    (Intercept) Age_Ex
## L1           0      1
##
## Positions of tested coefficients in the vector of coefficients: 2
##
## H0:  Age_Ex = 0
##
## Chi-squared test:
## X2 = 4.982, df = 1, P(> X2) = 0.02562
```

```
# generates binomial tests for each age group that test whether children chose the
# reputational character more often than would be predicted by chance
psh_binomial_tests = psh_summary %>%
  group_by(AgeGroup) %>%
  do(test = tidy(binom.test(sum(.$number_reputational),
                     sum(.$total_n),
                     p = .5,
                     conf.level = 0.95)))

# generates dataframe with results of binomial tests described above
psh_binomial_test_summary = psh_binomial_tests$test %>%
  bind_rows %>%
  bind_cols(psh_binomial_tests[1], .)

# generates table of binomial test estimates
psh_binomial_test_summary %>%
  kable(caption = "Binomial Tests") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = c("hold_position", "scale_down"))
```

Table 12: Binomial Tests

AgeGroup	estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
4-5	0.40625	13	0.3770856	32	0.2369841	0.5935508	Exact binomial test	two.sided
6-7	0.31250	10	0.0501025	32	0.1611847	0.5000776	Exact binomial test	two.sided
8-9	0.18750	6	0.0005351	32	0.0720762	0.3643923	Exact binomial test	two.sided

4 Study 4 - Who Sought Help Publicly versus Privately?

```
# read in data
pvp_data = read_csv("../data/study4_public_vs_private/data_study4.csv")
```

```r
# data cleaning
pvp_data = pvp_data %>%
  mutate(Sex = as.factor(Sex)) %>%
  mutate(Condition = ifelse(Condition == "1",
                            "private",
                            "public"),
         Sex = ifelse(Sex == "0",
                      "female",
                      "male"),
         SoughtHelp = ifelse(SoughtHelp == "0",
                             "Reputational",
                             "Intrinsic")) %>% # recode data
  mutate(AgeGroup = ifelse((Age_Yrs == 4|Age_Yrs == 5), "4-5",
                          ifelse((Age_Yrs == 6|Age_Yrs == 7), "6-7",
                          ifelse((Age_Yrs == 8|Age_Yrs ==9), "8-9", NA)))) %>%
    mutate(AgeGroup = as.factor(AgeGroup)) %>%
    mutate(SoughtHelp = as.factor(as.character(SoughtHelp)),
           Condition = as.factor(as.character(Condition)))
```

```r
# create dataframe with data from private condition ONLY
study4_private_data = pvp_data %>%
  filter(Condition == "private")

# create dataframe with data from public condition ONLY
study4_public_data = pvp_data %>%
  filter(Condition == "public")
```

4.1 Summary

```r
# generates table with percentage breakdown of children who identified either the
# reputational or intrinsic character as the one who sought help
pvp_summary = pvp_data %>%
  group_by(Condition, AgeGroup) %>%
  summarise(percent_intrinsic = (sum(SoughtHelp == "Intrinsic")/length(Sub_ID))*100,
            number_intrinsic = (sum(SoughtHelp == "Intrinsic")),
            percent_reputational = (sum(SoughtHelp == "Reputational")/length(Sub_ID))*100,
            number_reputational = (sum(SoughtHelp == "Reputational")),
            total_n = (sum(SoughtHelp == "Intrinsic"|SoughtHelp == "Reputational")))
```

```
## `summarise()` regrouping output by 'Condition' (override with `.groups` argument)
```

```r
# generates table
pvp_summary %>%
  kable(caption = "Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = c("hold_position", "scale_down"))
```

Table 13: Summary

Condition	AgeGroup	percent_intrinsic	number_intrinsic	percent_reputational	number_reputational	total_n
private	4-5	65.625	21	34.375	11	32
private	6-7	28.125	9	71.875	23	32
private	8-9	40.625	13	59.375	19	32
public	4-5	34.375	11	65.625	21	32
public	6-7	46.875	15	53.125	17	32
public	8-9	71.875	23	28.125	9	32

4.2 Demographics

```
# gender

# creates dataframe with percentage breakdown for gender of participants
pvp_gender_summary = pvp_data %>%
  group_by(Condition, AgeGroup, Sex) %>%
  summarise(n = length(Sub_ID)) %>%
            mutate(percentage = (n / sum(n))*100)
```

```
## `summarise()` regrouping output by 'Condition', 'AgeGroup' (override with `.groups` argument)
```

```
 # generates table
pvp_gender_summary %>%
  kable(caption = "Gender Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 14: Gender Summary

Condition	AgeGroup	Sex	n	percentage
private	4-5	female	18	56.250
private	4-5	male	14	43.750
private	6-7	female	23	71.875
private	6-7	male	9	28.125
private	8-9	female	17	53.125
private	8-9	male	15	46.875
public	4-5	female	15	46.875
public	4-5	male	17	53.125
public	6-7	female	16	50.000
public	6-7	male	16	50.000
public	8-9	female	15	46.875
public	8-9	male	17	53.125

```
# age

# creates dataframe with summary stats (mean and SD) for age of participants
pvp_age_summary <- pvp_data %>%
  group_by(Condition, AgeGroup) %>%
```

```r
  summarise(mean_age = mean(Age_Ex),
            SD_age = sd(Age_Ex))
```

```
## `summarise()` regrouping output by 'Condition' (override with `.groups` argument)
```

```r
# generates table
pvp_age_summary %>%
  kable(caption = "Age Summary") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = "hold_position")
```

Table 15: Age Summary

Condition	AgeGroup	mean_age	SD_age
private	4-5	4.983246	0.6207233
private	6-7	6.967187	0.6035094
private	8-9	8.960851	0.5645325
public	4-5	4.803385	0.5404293
public	6-7	6.744010	0.5703226
public	8-9	8.974306	0.6322552

4.3 Confirmatory Analyses

4.3.1 Examining interaction between condition and age

```r
# recoding data such that 'intrinsic' is 0 and 'reputational' is 1 &
# private condition is 0 and public condition is 1
recoded_pvp_data = pvp_data %>%
  mutate(Condition = ifelse(Condition == "private",
                            "0",
                            "1"),
         SoughtHelp = ifelse(SoughtHelp == "Intrinsic",
                             "0",
                             "1")) %>%
  mutate(Condition = factor(Condition, levels = c(0, 1)),
         SoughtHelp = factor(SoughtHelp, levels = c(0,1)))
```

```r
# generates logistic regression model examining the effect of age (entered as a continuous
# predictor) and the effect of condition (entered as a categorical predictor) on whether
# children identified the character with reputational or intrinsic concerns as
# the one who would seek help
pvp_logit_model = glm(formula = SoughtHelp ~ Age_Ex * factor(Condition),
                      data = recoded_pvp_data,
                      family = "binomial")

# prints summary of above logit model
summary(pvp_logit_model)
```

```
##
## Call:
## glm(formula = SoughtHelp ~ Age_Ex * factor(Condition), family = "binomial",
##     data = recoded_pvp_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6204  -1.1233   0.8024   1.0568   1.6831
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -1.5221     0.8786  -1.732 0.083190 .
## Age_Ex                      0.2497     0.1237   2.018 0.043544 *
## factor(Condition)1          4.0909     1.2431   3.291 0.000998 ***
## Age_Ex:factor(Condition)1  -0.6325     0.1762  -3.589 0.000332 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 265.84  on 191  degrees of freedom
## Residual deviance: 250.57  on 188  degrees of freedom
## AIC: 258.57
##
## Number of Fisher Scoring iterations: 4
```

```r
# computes Wald chi-square value for main effect of age
pvp_wald_test_age_main_effect = wald.test(b=coef(pvp_logit_model),
                                          Sigma = vcov(pvp_logit_model),
                                          Terms = 2,
                                          verbose = TRUE)

# prints output of Wald test described above
print(pvp_wald_test_age_main_effect, digits = 5)
```

```
## Wald test:
## ----------
##
## Coefficients:
##               (Intercept)                    Age_Ex        factor(Condition)1
##                  -1.52213                   0.24972                   4.09091
## Age_Ex:factor(Condition)1
##                  -0.63248
##
## Var-cov matrix of the coefficients:
##                           (Intercept) Age_Ex    factor(Condition)1
## (Intercept)                0.771922   -0.105553 -0.771922
## Age_Ex                    -0.105553    0.015307  0.105553
## factor(Condition)1        -0.771922    0.105553  1.545229
## Age_Ex:factor(Condition)1  0.105553   -0.015307 -0.212544
##                           Age_Ex:factor(Condition)1
## (Intercept)                0.105553
## Age_Ex                    -0.015307
## factor(Condition)1        -0.212544
```

```
## Age_Ex:factor(Condition)1  0.031057
##
## Test-design matrix:
##    (Intercept) Age_Ex factor(Condition)1 Age_Ex:factor(Condition)1
## L1           0      1                  0                         0
##
## Positions of tested coefficients in the vector of coefficients: 2
##
## H0:  Age_Ex = 0
##
## Chi-squared test:
## X2 = 4.0742, df = 1, P(> X2) = 0.043544
```

```
# computes Wald chi-square value for main effect of Condition
pvp_wald_test_condition_main_effect <- wald.test(b=coef(pvp_logit_model),
                                                 Sigma = vcov(pvp_logit_model),
                                                 Terms = 3,
                                                 verbose = TRUE)

 # prints Wald test described above
print(pvp_wald_test_condition_main_effect, digits = 5)
```

```
## Wald test:
## ----------
##
## Coefficients:
##               (Intercept)                    Age_Ex        factor(Condition)1
##                  -1.52213                   0.24972                   4.09091
## Age_Ex:factor(Condition)1
##                  -0.63248
##
## Var-cov matrix of the coefficients:
##                           (Intercept) Age_Ex    factor(Condition)1
## (Intercept)                0.771922   -0.105553 -0.771922
## Age_Ex                    -0.105553    0.015307  0.105553
## factor(Condition)1        -0.771922    0.105553  1.545229
## Age_Ex:factor(Condition)1  0.105553   -0.015307 -0.212544
##                           Age_Ex:factor(Condition)1
## (Intercept)                0.105553
## Age_Ex                    -0.015307
## factor(Condition)1        -0.212544
## Age_Ex:factor(Condition)1  0.031057
##
## Test-design matrix:
##    (Intercept) Age_Ex factor(Condition)1 Age_Ex:factor(Condition)1
## L1           0      0                  1                         0
##
## Positions of tested coefficients in the vector of coefficients: 3
##
## H0:  factor(Condition)1 = 0
##
## Chi-squared test:
## X2 = 10.83, df = 1, P(> X2) = 0.00099844
```

```r
# computes Wald chi-square value for interaction between age and condition
pvp_wald_test_interaction = wald.test(b = coef(pvp_logit_model),
                                      Sigma = vcov(pvp_logit_model),
                                      Terms = 4,
                                      verbose = TRUE)

# prints Wald test described above
print(pvp_wald_test_interaction, digits = 5)
```

```
## Wald test:
## ----------
##
## Coefficients:
##              (Intercept)                  Age_Ex       factor(Condition)1
##                 -1.52213                 0.24972                  4.09091
## Age_Ex:factor(Condition)1
##                 -0.63248
##
## Var-cov matrix of the coefficients:
##                           (Intercept) Age_Ex    factor(Condition)1
## (Intercept)                0.771922   -0.105553 -0.771922
## Age_Ex                    -0.105553    0.015307  0.105553
## factor(Condition)1        -0.771922    0.105553  1.545229
## Age_Ex:factor(Condition)1  0.105553   -0.015307 -0.212544
##                           Age_Ex:factor(Condition)1
## (Intercept)                0.105553
## Age_Ex                    -0.015307
## factor(Condition)1        -0.212544
## Age_Ex:factor(Condition)1  0.031057
##
## Test-design matrix:
##    (Intercept) Age_Ex factor(Condition)1 Age_Ex:factor(Condition)1
## L1           0      0                  0                         1
##
## Positions of tested coefficients in the vector of coefficients: 4
##
## H0:  Age_Ex:factor(Condition)1 = 0
##
## Chi-squared test:
## X2 = 12.881, df = 1, P(> X2) = 0.00033199
```

4.3.2 Examining effect of age within private condition

```r
# generates logistic regression model examining the effect of age (entered as a continuous
# predictor) on whether children identified the character with
# reputational or intrinsic concerns as the one who would privately seek help
study4_private_logit_model = glm(SoughtHelp ~ Age_Ex,
                                 data = study4_private_data,
                                 family = binomial(link = "logit"))

# generates summary of logistic regression model
summary(study4_private_logit_model)
```

```
##
## Call:
## glm(formula = SoughtHelp ~ Age_Ex, family = binomial(link = "logit"),
##     data = study4_private_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6063  -1.1476   0.8633   1.0384   1.3959
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.5221     0.8786  -1.732   0.0832 .
## Age_Ex        0.2497     0.1237   2.018   0.0435 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 132.04  on 95  degrees of freedom
## Residual deviance: 127.80  on 94  degrees of freedom
## AIC: 131.8
##
## Number of Fisher Scoring iterations: 4
```

```
# computes Wald chi-square value for logistic regression model described above
study4_private_wald_test = wald.test(b = coef(study4_private_logit_model),
                                     Sigma = vcov(study4_private_logit_model),
                                     Terms = 2,
                                     verbose = TRUE)

# prints results of Wald chi-square test
print(study4_private_wald_test, digits = 5)
```

```
## Wald test:
## ----------
##
## Coefficients:
## (Intercept)      Age_Ex
##    -1.52213     0.24972
##
## Var-cov matrix of the coefficients:
##             (Intercept) Age_Ex
## (Intercept)  0.771922   -0.105553
## Age_Ex      -0.105553    0.015307
##
## Test-design matrix:
##    (Intercept) Age_Ex
## L1           0      1
##
## Positions of tested coefficients in the vector of coefficients: 2
##
## H0:  Age_Ex = 0
##
## Chi-squared test:
```

```
## X2 = 4.0742, df = 1, P(> X2) = 0.043544
```

```
# generates binomial tests for each age group that test whether children chose the
# reputational character more often than would be predicted by chance
study4_private_binomial_tests = pvp_summary %>%
  filter(Condition == "private") %>%
  group_by(AgeGroup) %>%
  do(test = tidy(binom.test(sum(.$number_reputational),
                     sum(.$total_n),
                     p = .5,
                     conf.level = 0.95)))

# generates dataframe with results of binomial tests described above
study4_private_binomial_test_summary = study4_private_binomial_tests$test %>%
  bind_rows %>%
  bind_cols(study4_private_binomial_tests[1], .)

# generates table of binomial test estimates
study4_private_binomial_test_summary %>%
  kable(caption = "Binomial Tests") %>%
  kable_styling(bootstrap_options = "striped",
                full_width = F,
                latex_options = c("hold_position", "scale_down"))
```

Table 16: Binomial Tests

AgeGroup	estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
4-5	0.34375	11	0.1101842	32	0.1857191	0.5319310	Exact binomial test	two.sided
6-7	0.71875	23	0.0200616	32	0.5325289	0.8625431	Exact binomial test	two.sided
8-9	0.59375	19	0.3770856	32	0.4064492	0.7630159	Exact binomial test	two.sided

4.3.3 Examining effect of age within public condition

```
# generates logistic regression model examining the effect of age (entered as a continuous
# predictor) on whether children identified the character with
# reputational or intrinsic concerns as the one who would publicly seek help
study4_public_logit_model = glm(SoughtHelp ~ Age_Ex,
                          data = study4_public_data,
                          family = binomial(link = "logit"))

# generates summary of logistic regression model
summary(study4_public_logit_model)
```

```
##
## Call:
## glm(formula = SoughtHelp ~ Age_Ex, family = binomial(link = "logit"),
##     data = study4_public_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
```

```
## -1.620  -1.075  -0.717   1.078   1.683
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.5688     0.8794   2.921  0.00349 **
## Age_Ex       -0.3828     0.1255  -3.050  0.00229 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 133.04  on 95  degrees of freedom
## Residual deviance: 122.77  on 94  degrees of freedom
## AIC: 126.77
##
## Number of Fisher Scoring iterations: 4
```

```
# computes Wald chi-square value for logistic regression model described above
study4_public_wald_test = wald.test(b=coef(study4_public_logit_model),
                                    Sigma = vcov(study4_public_logit_model),
                                    Terms = 2,
                                    verbose = TRUE)

 # prints results of Wald chi-square test
print(study4_public_wald_test, digits = 5)
```

```
## Wald test:
## ----------
##
## Coefficients:
## (Intercept)      Age_Ex
##     2.56878    -0.38276
##
## Var-cov matrix of the coefficients:
##             (Intercept) Age_Ex
## (Intercept)  0.77331    -0.10699
## Age_Ex      -0.10699     0.01575
##
## Test-design matrix:
##    (Intercept) Age_Ex
## L1           0      1
##
## Positions of tested coefficients in the vector of coefficients: 2
##
## H0:  Age_Ex = 0
##
## Chi-squared test:
## X2 = 9.3018, df = 1, P(> X2) = 0.0022893
```

```
# generates binomial tests for each age group that test whether children chose the
# reputational character more often than would be predicted by chance
study4_public_binomial_tests = pvp_summary %>%
  filter(Condition == "public") %>%
```

```r
  group_by(AgeGroup) %>%
  do(test = tidy(binom.test(sum(.$number_reputational),
                      sum(.$total_n),
                      p = .5,
                      conf.level = 0.95)))

# generates dataframe with results of binomial tests described above
study4_public_binomial_test_summary = study4_public_binomial_tests$test %>%
  bind_rows %>%
  bind_cols(study4_public_binomial_tests[1], .)

# generates table of binomial test estimates
study4_public_binomial_test_summary %>%
  kable(caption = "Binomial Tests") %>%
  kable_styling(bootstrap_options = "striped",
              full_width = F,
              latex_options = c("hold_position", "scale_down"))
```

Table 17: Binomial Tests

AgeGroup	estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
4-5	0.65625	21	0.1101842	32	0.4680690	0.8142809	Exact binomial test	two.sided
6-7	0.53125	17	0.8600501	32	0.3474368	0.7090602	Exact binomial test	two.sided
8-9	0.28125	9	0.0200616	32	0.1374569	0.4674711	Exact binomial test	two.sided

5 Plots

```r
# data wrangling for predicted probability plot (Studies 1 & 2)
df.lap_logit_model = ggpredict(model = lap_logit_model,
                              terms = "Age_Ex [all]") %>%
  rename_all(function(x) paste0("lap_", x)) %>%
  mutate(id = 1:n()) %>%
  select(-lap_group)

df.las_logit_model = ggpredict(model = las_logit_model,
                              terms = "Age_Ex [all]") %>%
  rename_all(function(x) paste0("las_", x)) %>%
  mutate(id = 1:n()) %>%
  select(-las_group)

df.combined_model = df.lap_logit_model %>%
  left_join(df.las_logit_model, by = "id") %>%
  pivot_longer(cols = -id,
             names_to = c("study", "value_type"),
             names_sep = "_",
             values_to = "value") %>%
  pivot_wider(names_from = "value_type",
            values_from = "value") %>%
  mutate(study = ifelse(study == "lap",
                       "Lying about Performance",
```

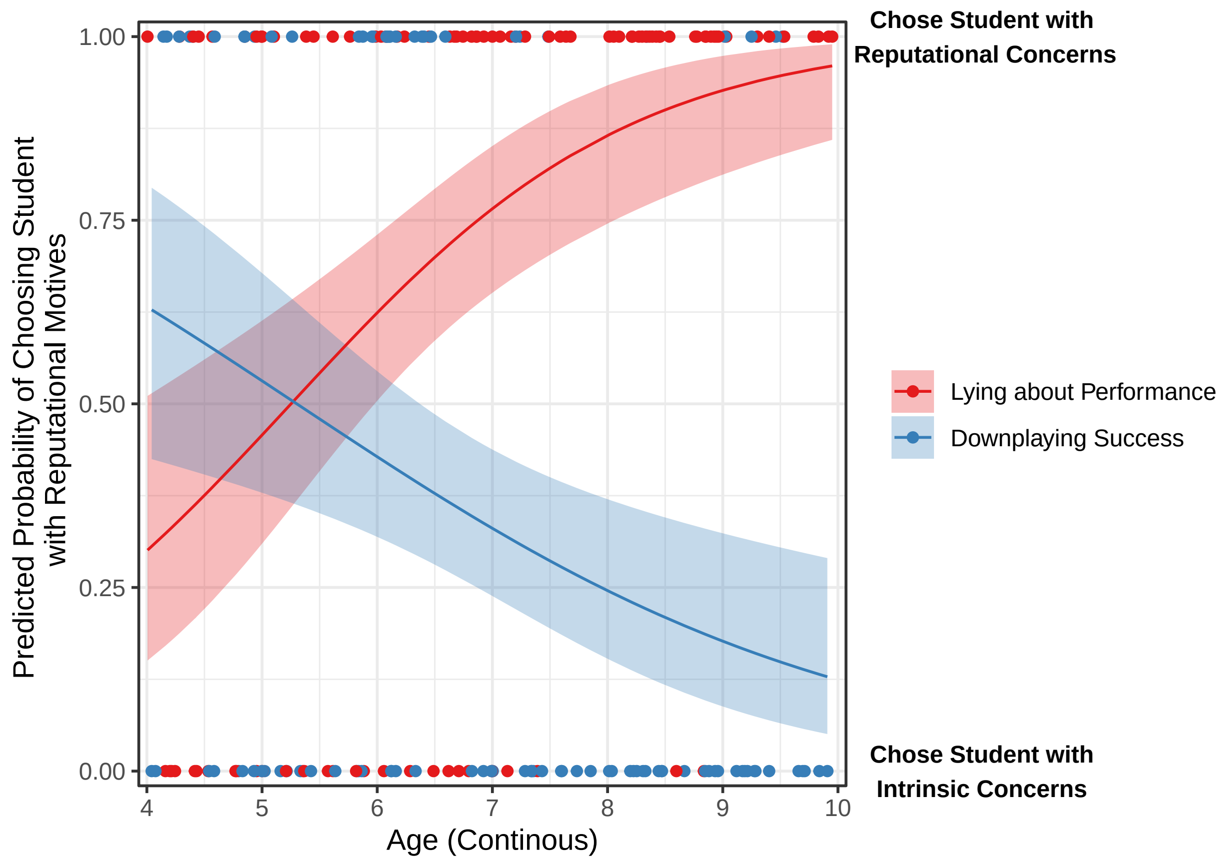
```
                        "Downplaying Success")) %>%
  mutate(study = factor(study,
                        levels = c("Lying about Performance",
                                   "Downplaying Success")))
```

```
# for individual data points (lying about poor performance)
df.lap_data = lap_data %>%
  rename_all(function(x) paste0("lap_", x)) %>%
  rename(Sub_ID = lap_Sub_ID,
         lap_AgeEx = lap_Age_Ex) %>%
  select(c(Sub_ID, lap_AgeEx, lap_WhoLied)) %>%
  mutate(lap_WhoLied = ifelse(lap_WhoLied == "Intrinsic", 0, 1))

# for individual data points (lying about good performance)
df.las_data = las_data %>%
  rename_all(function(x) paste0("las_", x)) %>%
  rename(Sub_ID = las_Sub_ID,
         las_AgeEx = las_Age_Ex) %>%
  select(c(Sub_ID, las_AgeEx, las_WhoLied)) %>%
  mutate(las_WhoLied = ifelse(las_WhoLied == "Intrinsic", 0, 1))

# combined df with individual data (studies 1 & 2)
df.combined_lap_las = df.lap_data %>%
  left_join(df.las_data, by = "Sub_ID") %>%
  pivot_longer(cols = -Sub_ID,
               names_to = c("study", "value_type"),
               names_sep = "_",
               values_to = "value") %>%
  pivot_wider(names_from = "value_type",
              values_from = "value") %>%
  mutate(study = ifelse(study == "lap",
                        "Lying about Performance",
                        "Downplaying Success")) %>%
  mutate(study = factor(study,
                        levels = c("Lying about Performance",
                                   "Downplaying Success")))
```

```
# text labels for plot
text_reputational = textGrob("Chose Student with\n Reputational Concerns",
                             gp=gpar(fontsize=9,
                                     fontface="bold"))
text_intrinsic = textGrob("Chose Student with\nIntrinsic Concerns",
                          gp=gpar(fontsize=9,
                                  fontface="bold"))
```

```
# generate plot
df.combined_model %>%
  ggplot(mapping = aes(x = x,
                       y = predicted,
                       group = study,
                       color = study,
                       fill = study)) +
  geom_ribbon(data = df.combined_model,
```

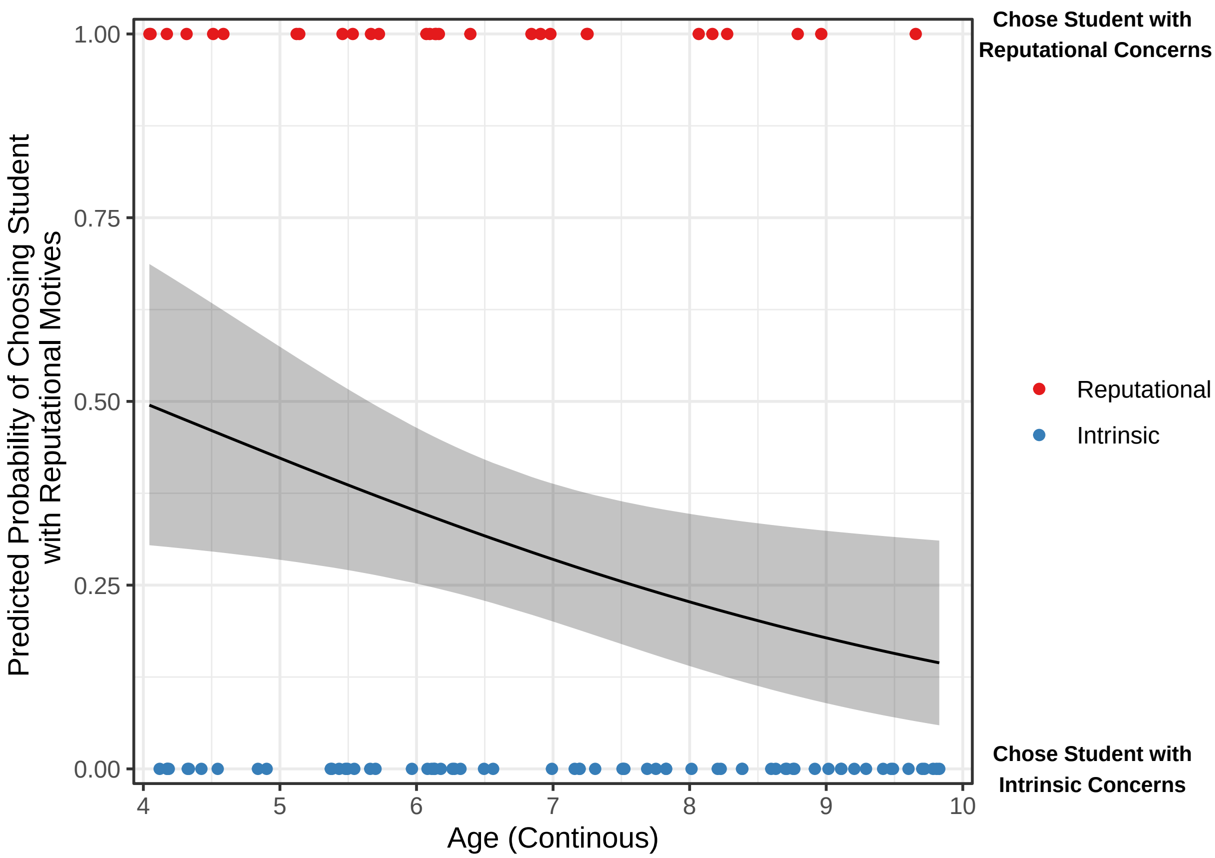
```r
              mapping = aes(ymin = conf.low,
                            ymax = conf.high),
              linetype = 0,
              alpha = 0.3) +
  geom_line(na.rm = TRUE) +
  geom_point(data = df.combined_lap_las,
             mapping = aes(x = AgeEx,
                           y = WhoLied,
                           color = study)) +
  scale_x_continuous(breaks = seq(4, 10, by = 1),
                     expand = c(.01, .01)) +
  coord_cartesian(xlim = c(4, 10),
                  ylim = c(0, 1),
                  clip = "off") +
  annotation_custom(text_reputational,
                    xmin=11.25,
                    xmax=11.25,
                    ymin=1,
                    ymax=1) +
  annotation_custom(text_intrinsic,
                    xmin=11.25,
                    xmax=11.25,
                    ymin=0,
                    ymax=0) +
  scale_y_continuous(breaks = seq(0, 1.00, by = .25),
                     expand = c(.01, .01)) +
  scale_color_brewer(palette = "Set1") +
  scale_fill_brewer(palette = "Set1") +
  theme(text = element_text(size = 15,
                            family = "Times New Roman",
                            color = "black")) +
  labs(x = "Age (Continous)",
       y = "Predicted Probability of Choosing Student \n with Reputational Motives",
       fill = element_blank(),
       color = element_blank()) +
  theme_bw()
```




```r
# data wrangling for predicted probability plot (Study 3)
df.psh_logit_model = ggpredict(model = psh_logit_model,
                               terms = "Age_Ex [all]")

# for individual data points
df.psh_data = psh_data %>%
  select(c(Sub_ID, Age_Ex, WhoAsked)) %>%
  mutate(WhoAsked = ifelse(WhoAsked == "Intrinsic", 0, 1),
         WhoAsked_label = ifelse(WhoAsked == 0,
                                 "Intrinsic",
                                 "Reputational"))

df.psh_data = df.psh_data %>%
  mutate(WhoAsked_label = factor(WhoAsked_label,
                                 levels = c("Reputational", "Intrinsic"),
                                 labels = c("Reputational", "Intrinsic")))
```

```r
# text labels for plot
text_reputational = textGrob("Chose Student with\n Reputational Concerns",
                             gp=gpar(fontsize=8,
                                     fontface="bold"))
text_intrinsic = textGrob("Chose Student with\nIntrinsic Concerns",
                          gp=gpar(fontsize=8,
                                  fontface="bold"))
```

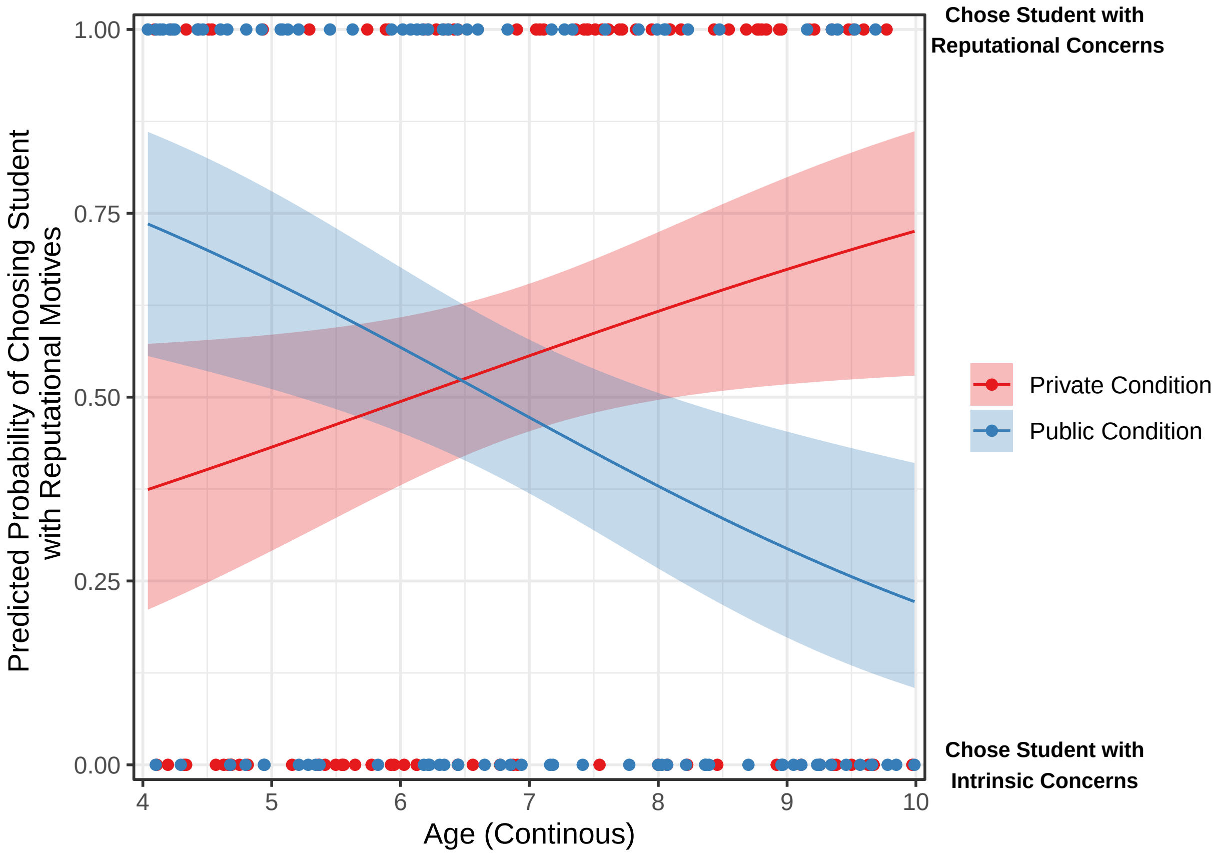
```
# generate plot
df.psh_logit_model %>%
  ggplot(mapping = aes(x = x,
                       y = predicted)) +
  geom_ribbon(data = df.psh_logit_model,
              mapping = aes(ymin = conf.low,
                            ymax = conf.high),
              linetype = 0,
              alpha = 0.3) +
  geom_line(na.rm = TRUE) +
  geom_point(data = df.psh_data,
             mapping = aes(x = Age_Ex,
                           y = WhoAsked,
                           color = as.factor(WhoAsked_label))) +
  annotation_custom(text_reputational,
                    xmin=10.95,
                    xmax=10.95,
                    ymin=1,
                    ymax=1) +
  annotation_custom(text_intrinsic,
                    xmin=10.95,
                    xmax=10.95,
                    ymin=0,
                    ymax=0) +
  scale_x_continuous(breaks = seq(4, 10, by = 1),
                     expand = c(.01, .01)) +
  scale_y_continuous(breaks = seq(0, 1.00, by = .25),
                     expand = c(.01, .01)) +
  scale_color_brewer(palette = "Set1") +
  labs(x = "Age (Continous)",
       y = "Predicted Probability of Choosing Student \n with Reputational Motives",
       fill = element_blank(),
       color = element_blank()) +
  theme(text = element_text(size = 15,
                            family = "Times New Roman",
                            color = "black")) +
  coord_cartesian(xlim = c(4, 10),
                  ylim = c(0, 1),
                  clip = "off") +
  theme_bw()
```




```
# data wrangling for predicted probability plot (Study 4)
df.pvp_logit_model = ggpredict(model = pvp_logit_model,
                               terms = c("Age_Ex [all]", "Condition"))

# creating data frame for plot
df.pvp_logit_plot = df.pvp_logit_model %>%
  rename(condition = group) %>%
  mutate(condition = ifelse(condition == "0",
                            "Private Condition",
                            "Public Condition")) %>%
  mutate(condition = factor(condition, levels = c("Private Condition",
                                                  "Public Condition")))

# formatting data for individual data points
df.pvp_data = pvp_data %>%
  select(c(Sub_ID, Age_Ex, SoughtHelp, Condition)) %>%
  rename(condition = Condition) %>%
  mutate(condition = ifelse(condition == "private",
                            "Private Condition",
                            "Public Condition")) %>%
  mutate(SoughtHelp = ifelse(SoughtHelp == "Intrinsic", 0, 1))
```

```
# create predicted probability plot
df.pvp_logit_plot %>%
  ggplot(mapping = aes(x = x,
                       y = predicted,
```

```r
                        group = condition,
                        color = condition,
                        fill = condition)) +
  geom_ribbon(data = df.pvp_logit_plot,
              mapping = aes(ymin = conf.low,
                            ymax = conf.high),
              linetype = 0,
              alpha = 0.3) +
  geom_line(na.rm = TRUE) +
  scale_color_brewer(palette = "Set1") +
  scale_fill_brewer(palette = "Set1") +
  geom_point(data = df.pvp_data,
             mapping = aes(x = Age_Ex,
                           y = SoughtHelp,
                           color = condition)) +
  scale_x_continuous(breaks = seq(4, 10, by = 1),
                     expand = c(.01, .01)) +
  coord_cartesian(xlim = c(4, 10),
                  ylim = c(0, 1),
                  clip = "off") +
  annotation_custom(text_reputational,
                    xmin=11,
                    xmax=11,
                    ymin=1,
                    ymax=1) +
  annotation_custom(text_intrinsic,
                    xmin=11,
                    xmax=11,
                    ymin=0,
                    ymax=0) +
  scale_y_continuous(breaks = seq(0, 1.00, by = .25),
                     expand = c(.01, .01)) +
  theme(text = element_text(size = 15,
                            family = "Times New Roman",
                            color = "black")) +
  labs(x = "Age (Continous)",
       y = "Predicted Probability of Choosing Student \n with Reputational Motives",
       fill = element_blank(),
       color = element_blank()) +
  theme_bw()
```



References

Devleesschauwer, Brecht, Paul Torgerson, Johannes Charlier, Bruno Levecke, Nicolas Praet, Sophie Roelandt, Suzanne Smit, Pierre Dorny, Dirk Berkvens, and Niko Speybroeck. 2015. *Prevalence: Tools for Prevalence Assessment Studies*. https://CRAN.R-project.org/package=prevalence.

Henry, Lionel, and Hadley Wickham. 2020. *Purrr: Functional Programming Tools*. https://CRAN.R-project.org/package=purrr.

J, Lemon. 2006. "Plotrix: A Package in the Red Light District of R." *R-News* 6 (4): 8–12.

Lemon, Jim, Ben Bolker, Sander Oom, Eduardo Klein, Barry Rowlingson, Hadley Wickham, Anupam Tyagi, et al. 2020. *Plotrix: Various Plotting Functions*. https://CRAN.R-project.org/package=plotrix.

Lesnoff, Matthieu, and Renaud Lancelot. 2019. *Aod: Analysis of Overdispersed Data*. https://CRAN.R-project.org/package=aod.

Lüdecke, Daniel. 2018. "Ggeffects: Tidy Data Frames of Marginal Effects from Regression Models." *Journal of Open Source Software* 3 (26): 772. https://doi.org/10.21105/joss.00772.

———. 2020. *Ggeffects: Create Tidy Data Frames of Marginal Effects for 'Ggplot' from Model Outputs*. https://CRAN.R-project.org/package=ggeffects.

Müller, Kirill, and Hadley Wickham. 2020. *Tibble: Simple Data Frames*. https://CRAN.R-project.org/package=tibble.

Plummer, Martyn. 2019. *Rjags: Bayesian Graphical Models Using Mcmc*. https://CRAN.R-project.org/package=rjags.

Plummer, Martyn, Nicky Best, Kate Cowles, and Karen Vines. 2006. "CODA: Convergence Diagnosis and Output Analysis for Mcmc." *R News* 6 (1): 7–11. https://journal.r-project.org/archive/.

Plummer, Martyn, Nicky Best, Kate Cowles, Karen Vines, Deepayan Sarkar, Douglas Bates, Russell Almond, and Arni Magnusson. 2019. *Coda: Output Analysis and Diagnostics for Mcmc*. https://CRAN.R-project.org/package=coda.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Robinson, David, and Alex Hayes. 2020. *Broom: Convert Statistical Analysis Objects into Tidy Tibbles*. https://CRAN.R-project.org/package=broom.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

———. 2019a. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

———. 2019b. *Tidyverse: Easily Install and Load the 'Tidyverse'*. https://CRAN.R-project.org/package=tidyverse.

———. 2020. *Forcats: Tools for Working with Categorical Variables (Factors)*. https://CRAN.R-project.org/package=forcats.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2020. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Lionel Henry. 2020. *Tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, Jim Hester, and Romain Francois. 2018. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Winston Chang. 2014. *Extrafont: Tools for Using Fonts*. https://CRAN.R-project.org/package=extrafont.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

———. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.org/knitr/.

———. 2020. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://CRAN.R-project.org/package=knitr.

Zhu, Hao. 2019. *KableExtra: Construct Complex Table with 'Kable' and Pipe Syntax*. https://CRAN.R-project.org/package=kableExtra.