Journal Data Policy

Sebastian Karcher

June 29, 2019

Reading in and preparing the data

```
jpolicy <- read.csv(file = here("journal_data_policy_20190629.tsv"), sep = "\t", encoding = "UTF-8")


# Converting strings to character variables
jpolicy$title <- as.character(jpolicy$title)
jpolicy$ISSN <- as.character(jpolicy$ISSN)
jpolicy$policy_text <- as.character(jpolicy$policy_text)
jpolicy$policy_link <- as.character(jpolicy$policy_link)
jpolicy$nomenclature <- as.character(jpolicy$nomenclature)
```

Generate Codebook

```
library(memisc)
codebook_data <- as.data.set(jpolicy)
codebook_data <- within(codebook_data, {
  description(discipline) <- "Discipline"
  description(ranking) <- "Rank within Discipline"
  description(title) <- "Journal title"
  description(publisher) <- "Journal publisher"
  description(IF) <- "Journal impact factor"
  description(ISSN) <- "Journal ISSN"
  description(language) <- "Journal publication language"
  description(frequency) <- "Publication frequency"
  description(year) <- "Year of first publication"
  description(policy_text) <- "Excerpt from data policy"
  description(policy_link) <- "Link to data policy"
  description(DART) <- "Journal has signed JETS"
  description(TOPS) <- "Journal has signed TOP"
  description(has_policy) <- "Journal has data policy"
  description(source) <- "Source of data policy"
  description(strictness) <- "Strictness of data policy"
  description(qual_data) <- "Mention of qualitative data in policy"
  description(data_location) <- "Location for data sharing"
  description(submission_time) <- "Timing of data submission"
  description(publication_time) <- "Timing of data publication"
  description(nomenclature) <- "Nomenclature used for data"
  description(codebooks) <- "Mention of codebooks"
  description(scripts) <- "Mention of analysis scripts"
  description(benefits) <- "Mention of benefits of data sharing"
  description(citation) <- "Data citation instructions"
  description(placement) <- "Data availability in journal"
  description(legality) <- "Ethical/legal exemptions"
  description(consequences) <- "Consequence for non-compliance"
```

```
 foreach(x=c(DART, TOPS, has_policy, source, strictness, qual_data, data_location, submission_time, publ
    measurement(x) <- "nominal"
    })
  labels(DART) <-c("No" =0, "Yes" =1)
  labels(TOPS) <- c("No" =0, "Yes" =1)
  labels(has_policy) <- c("No" =0, "Yes" =1)
  labels(source) <- c("No policy" =0, "Journal" =1, "Publisher" =2, "Association"=3)
  labels(strictness) <- c("No policy"=0, "Encourage"=1, "Require" =2)
  labels(qual_data) <- c("No mention" =0, "Implicit" =1, "Explicit" =2)
  labels(data_location) <-  c('No policy' =0, 'On request'=1,'No location specified'=2,'Author website'


  labels(submission_time) <- c("No data policy" =0, "No timing specified" =1, "After publication" =2, "A
  labels(publication_time) <- c("No data policy" =0, "No timing specified" =1, "After embargo"=2, "Afte
  labels(codebooks) <- c("No mention" =0, "Mentioned" =1, "Required" =2)
  labels(scripts) <- c("No mention" =0, "Mentioned" =1, "Required" =2)
  labels(benefits) <- c("No" =0, "Yes" =1)
  labels(citation) <- c("No mention" =0, "Mentions citing" =1, "Specific Template" =2)
  labels(placement) <- c("No" =0, "Yes" =1)
  labels(legality) <- c("No" =0, "Yes" =1)
  labels(consequences) <- c("No" =0, "Yes" =1)


  wording(discipline) <- "Discipline the journal is listed under in the Journal Citation Report"
  wording(ranking) <- "Ranking of the journal by impact factor within its discipline in the Journal Cita
  wording(publisher) <- "The journal's publisher. Where a journal is published by a publisher 'on behali
  wording(IF) <- "The journal's 2016 Thomson Reuters (now Claryvate) Impact Factor"
  wording(ISSN) <- "The journal's International Standard Serial Number (ISSN)"
  wording(language) <- "In which language(s) does the journal currently publish articles?"
  wording(frequency) <- "How many issues does the journal publish per year?"
  wording(year) <- "In which year was the journal first published?"
  wording(policy_text) <- "Relevant passages from the data policy unless too long"
  wording(DART) <- "Has the journal signed onto the DA-RT Journal Editors' Transparency Statement? (poli
  wording(TOPS) <- "Has the journal signed onto the TOP guidelines? (by time of data collection fall 201
  wording(has_policy) <- "Does the journal have a data policy? Any mention of depositing data in the aut
  wording(source) <- "What is the source of the data policy? Defaults to journal. Associations are coded
  wording(strictness) <- "Does the journal require data sharing or just mention/encourage it? Any mentio
  wording(qual_data) <- "Does the data policy mention qualitative data? Implicit mention includes inclus
  wording(data_location) <- "Where should shared data be deposited? Codes preferred option where multipl
  wording(submission_time) <- "When should authors submit the data in relation to the article they accor
  wording(publication_time) <- "When are data published in relation to the publication time of the artic
  wording(nomenclature) <- "What language is used to describe data and other materials?"
  wording(codebooks) <- "Does the data policy mention or require the deposit of codebooks?"
  wording(scripts) <- "Does the data policy mention or require the deposit of analysis scripts? Any ment
  wording(benefits) <- "Does the data policy mention any benefits of sharing data?"
  wording(citation) <- "Does the journal specify how data should be cited? (Can refer to either authors
  wording(placement) <- "Does the journal specify that replication data should be mentioned in the manus
  wording(legality) <- "Does the policy mention any exception or allowance for legal concerns related to
  wording(consequences) <- "Does the journal specify consequences for not sharing data? (e.g., article w
})

filename_codebook = paste("codebook_", Sys.Date(), sep = "", ".txt")
policy_codebook <- codebook(codebook_data)
```

```
Write(policy_codebook, file = filename_codebook)
```

Recode data and generate new variables

```
# Turn categorical variables into factors
jpolicy$DART <- factor(jpolicy$DART, labels=c("No", "Yes"))
jpolicy$TOPS <- factor(jpolicy$TOPS, labels=c("No", "Yes"))
jpolicy$has_policy <- factor(jpolicy$has_policy, levels = c(0, 1), labels=c("No", "Yes"))
jpolicy$source <-  factor(jpolicy$source, labels=c("No policy", "Journal", "Publisher", "Association"))
jpolicy$strictness <- factor(jpolicy$strictness, labels=c("No policy", "Encourage", "Require"))
jpolicy$qual_data <- factor(jpolicy$qual_data, labels=c("No mention", "Implicit", "Explicit"))
jpolicy$codebooks <- factor(jpolicy$codebooks, labels=c("No mention", "Mentioned", "Required"))
jpolicy$benefits <- factor(jpolicy$benefits, labels=c("No", "Yes"))
jpolicy$citation <- factor(jpolicy$citation, labels=c("No mention", "Mentions citing", "Specific Templa
jpolicy$legality <- factor(jpolicy$legality, labels=c("No", "Yes"))
jpolicy$consequences <- factor(jpolicy$consequences, labels=c("No", "Yes"))
jpolicy$scripts <- factor(jpolicy$scripts, labels=c("No mention", "Mentioned", "Required"))

jpolicy$publication_time <- factor(jpolicy$publication_time, levels=c("0", "1", "2", "3", "4"), labels=

jpolicy$submission_time <- factor(jpolicy$submission_time, levels = c("0", "1", "2", "3", "4", "5", "6"

jpolicy$placement <- factor(jpolicy$placement, labels=c("No", "Yes"))




# Generate new variables
jpolicy$age <- 2017 - jpolicy$year

## Consolidating the Location variable
##The rational for this is explained below, but let's do all recoding here
jpolicy$data_location_simple <-dplyr::recode_factor(jpolicy$data_location, `0` = "No data policy", `1`=


jpolicy$data_location <- factor(jpolicy$data_location, levels = c("0", "1", "2", "3", "4", "5", "6"), la

## Create a binary strictness variable for the with policy data
jpolicy$bin_strictness <- factor(ifelse(jpolicy$strictness == "Require", "Require", "Don't require"))


# List and remove duplicates
jpolicy$title[duplicated(jpolicy$title)]
```

```
## [1] "Comparative Studies in Society and History"
## [2] "Global Networks"
## [3] "Human Ecology"
## [4] "International Political Sociology"
## [5] "Review of International Organizations"
## [6] "Review of International Political Economy"
## [7] "Social Networks"
## [8] "Socio-Economic Review"
```

```
## [9] "Socio-Economic Review"
```

```
jpolicy_dedup <-  distinct(jpolicy, title, .keep_all = TRUE)

# Create a subset with only journals that have a policy
jpolicy_with_policy <- subset(jpolicy, has_policy =="Yes")
# And one with without the duplicates
jpolicy_with_policy_dedup <- subset(jpolicy_dedup,has_policy == "Yes")
```

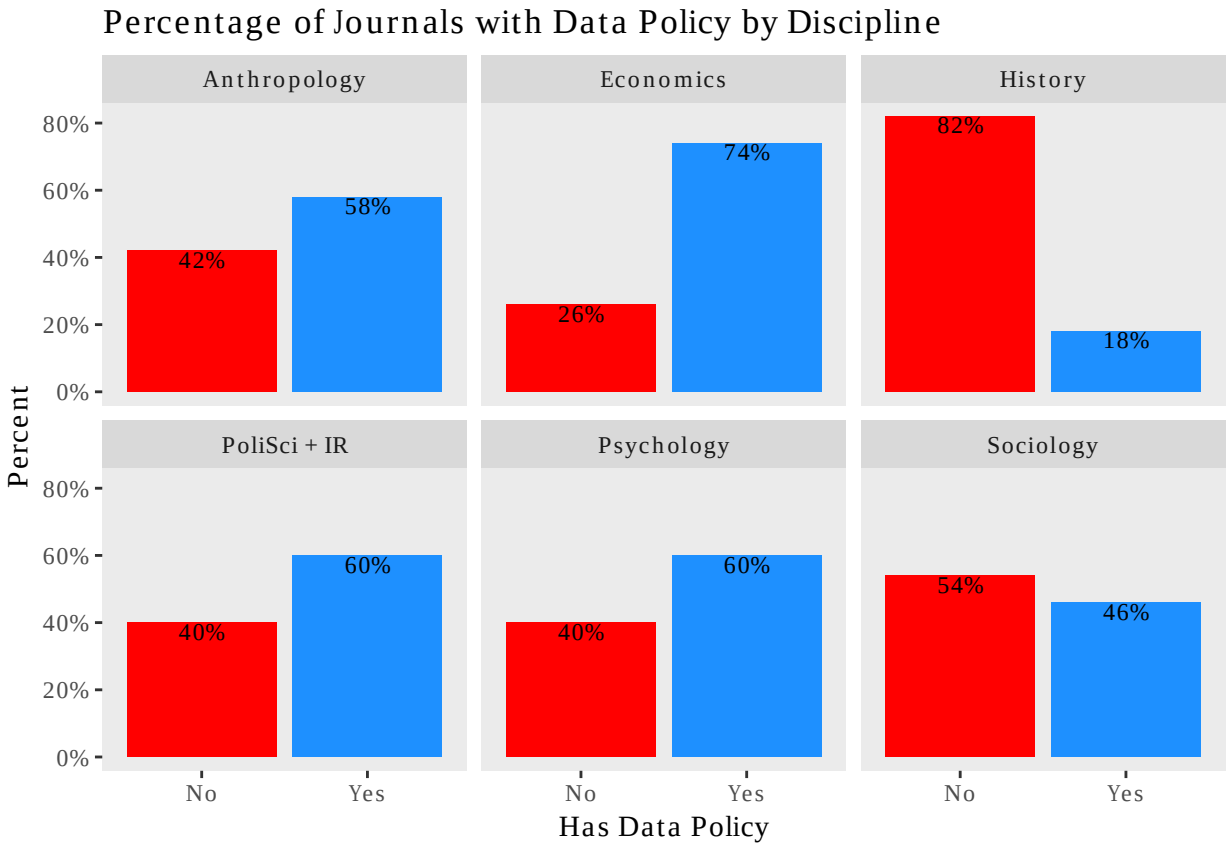
Policy by discipline

This generates *figure 1* and the total number of journals with a policy (the latter taken without duplicates)

```
jpolicy_dedup %>% dplyr::select(has_policy) %>% table()
```

```
## .
##  No Yes
## 136 155
```

```
policyplot <- ggplot(data=jpolicy, aes(has_policy, group="discipline")) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
  geom_text(aes(label = scales::percent_format(accuracy=1)(..prop..),
                  y= ..prop.. ), stat= "count", family= "Open Sans", size = 3, vjust = 1) +
  guides(fill=FALSE) +
  scale_fill_manual(values = c("red", "dodgerblue")) +
  labs(title ="Percentage of Journals with Data Policy by Discipline", y = "Percent", x = "Has Data Poli
  scale_y_continuous(labels=scales::percent_format(accuracy=1)) +
  facet_wrap(~ discipline, nrow = 2) +
  theme(text = element_text(family = "Open Sans"), strip.text = element_text(face = "bold"), panel.grid

print(policyplot)
```




```
ggsave("Has Policy By Discipline.png", plot = policyplot)
```

```
## Saving 6.5 x 4.5 in image
```

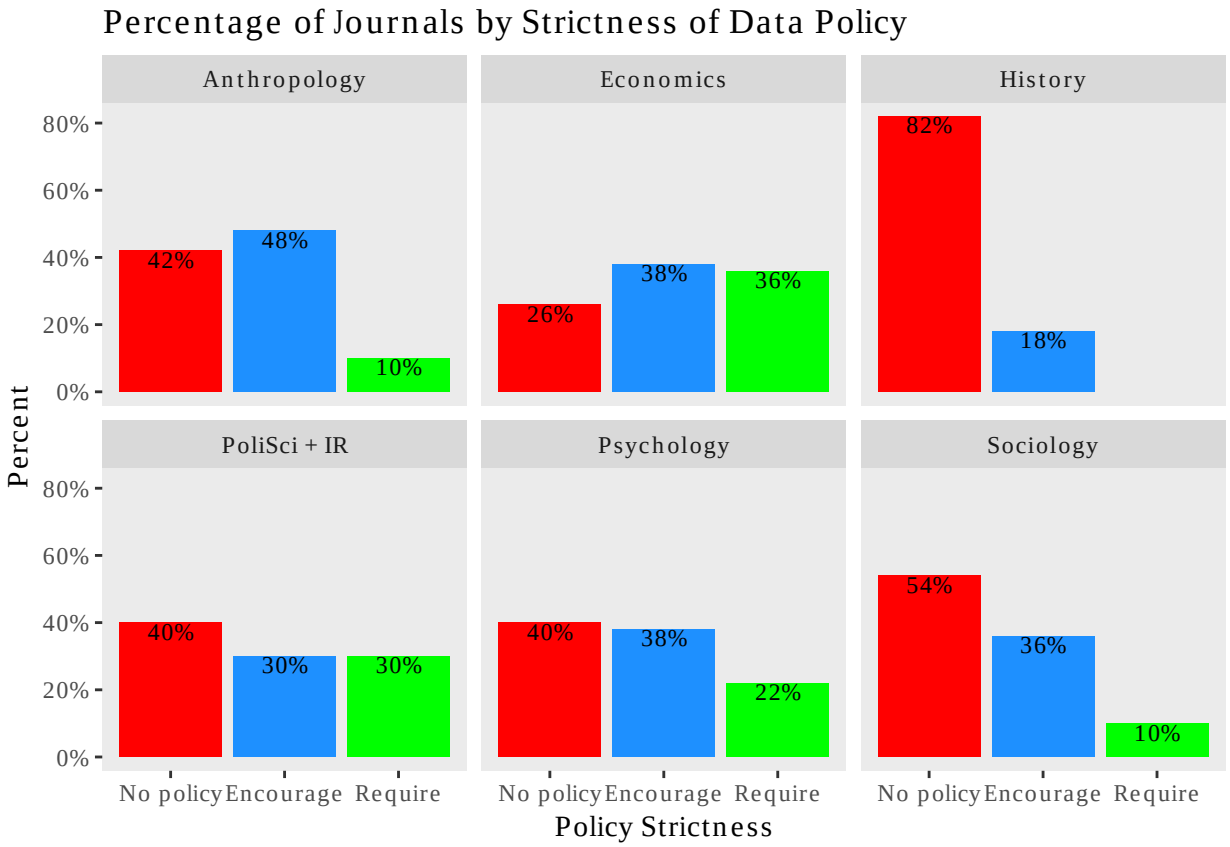
Strictness by discipline

This generates *figure 2* and the number of journals with a strict policy. Using the whole dataset for the graph but only interested in the deduped journals for the number

```
jpolicy_dedup %>% filter(has_policy=="Yes") %>% dplyr::select(strictness) %>% table()
```

```
## .
## No policy Encourage   Require
##         0       102        53
```

```
strictnessplot <- ggplot(data=jpolicy, aes(factor(strictness, labels=c("No policy", "Encourage", "Requi
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
  geom_text(aes( label = scales::percent_format(accuracy = 1)(..prop..),
                 y= ..prop.. ), stat= "count",family= "Open Sans", size = 3, vjust = 1) +
  guides(fill=FALSE) +
  scale_fill_manual(values = c("red", "dodgerblue", "green")) +
  labs(title ="Percentage of Journals by Strictness of Data Policy", y = "Percent", x = "Policy Strictn
  scale_y_continuous(labels=scales::percent_format(accuracy=1)) +
  facet_wrap(~ discipline, nrow = 2) +
  theme(text = element_text(family = "Open Sans"), strip.text = element_text(face = "bold"), panel.grid

print(strictnessplot)
```



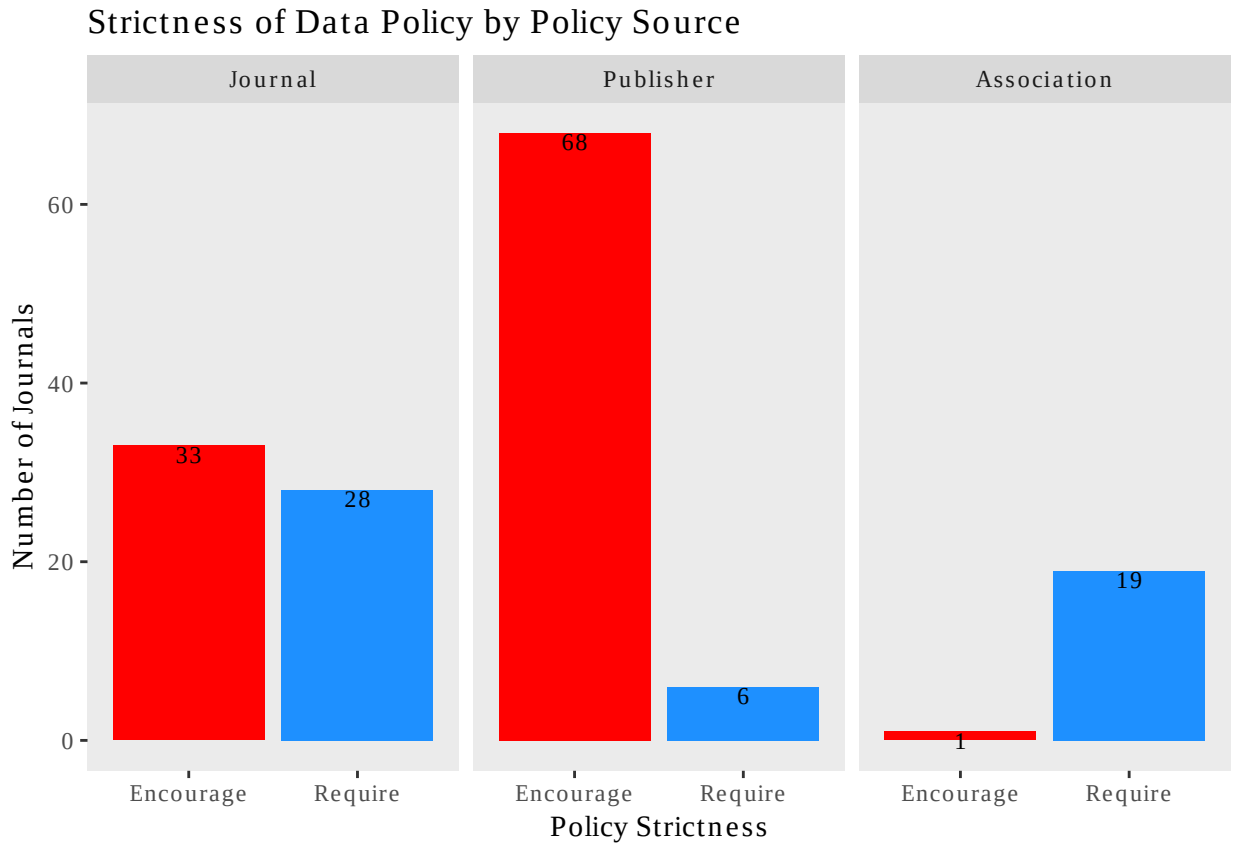

```
ggsave("Policy Strictness by Discipline.png", plot = strictnessplot)
```

```
## Saving 6.5 x 4.5 in image
```

Policy by policy source

We're running this only on deduped journals with policy. This produces *figure 3*.

```
sourcestrictplot <- jpolicy_with_policy_dedup %>% ggplot(aes(x = as.factor(strictness), group="source")
  geom_bar(aes(x = strictness, fill = factor (..x..)), stat="count") +
  geom_text(aes(label = ..count..), stat= "count", family= "Open Sans", size = 3, vjust = 1) +
  guides(fill=FALSE) +
    scale_fill_manual(values = c("red", "dodgerblue")) +
  labs(title ="Strictness of Data Policy by Policy Source", y = "Number of Journals", x = "Policy Strict
  scale_x_discrete(labels = c("Encourage", "Require"))+
  facet_wrap(~ source) +
  theme(text = element_text(family = "Open Sans"), strip.text = element_text(face = "bold"), panel.grid
print(sourcestrictplot)
```




```
ggsave("Strictness by Policysource.png", plot = sourcestrictplot)
```

```
## Saving 6.5 x 4.5 in image
```

Data Policy and Journal Ranking

The regressions underly the "Data Policy by Journal Ranking and Age" section. We only report estimated effect sizes for statisticially significant effects. The predicted effects and CIs are taken from the `head()` and `tail()` of the `cplot` for each regression.

```
policymodel <- glm(has_policy ~ discipline + IF, data = jpolicy_dedup, family = binomial)
summary(policymodel)
```

```
##
## Call:
## glm(formula = has_policy ~ discipline + IF, family = binomial,
##     data = jpolicy_dedup)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0312  -1.1417   0.7003   0.9905   1.8765
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.09762    0.37412  -0.261  0.79414
## disciplineEconomics    0.29563    0.49298   0.600  0.54872
```

```
## disciplineHistory      -1.54824    0.49565  -3.124  0.00179 **
## disciplinePoliSci + IR -0.01429    0.43570  -0.033  0.97383
## disciplinePsychology   -0.37332    0.44679  -0.836  0.40340
## disciplineSociology    -0.73036    0.41782  -1.748  0.08046 .
## IF                      0.26750    0.13503   1.981  0.04758 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 402.17  on 290  degrees of freedom
## Residual deviance: 355.55  on 284  degrees of freedom
## AIC: 369.55
##
## Number of Fisher Scoring iterations: 5
```

```
pdf("Effects of Journal Impact Factor on Data Policy.pdf", width = 11, height = 8)
margpolicyIFplot <- cplot(policymodel, x="IF", xlab ="Journal Impact Factor", main = "Predicted probabil
```

```
##        xvals     yvals     upper     lower
## 1   0.250000 0.4003357 0.6195170 0.1811543
## 2   1.070833 0.4540074 0.6421653 0.2658495
## 3   1.891667 0.5087698 0.6694098 0.3481297
## 4   2.712500 0.5633225 0.7077032 0.4189418
## 5   3.533333 0.6163849 0.7602600 0.4725099
## 6   4.354167 0.6668132 0.8223908 0.5112356
## 7   5.175000 0.7136916 0.8852434 0.5421398
## 8   5.995833 0.7563840 0.9419130 0.5708550
## 9   6.816667 0.7945430 0.9888662 0.6002199
## 10  7.637500 0.8280835 1.0250703 0.6310967
## 11  8.458333 0.8571331 1.0510118 0.6632545
## 12  9.279167 0.8819737 1.0679856 0.6959618
## 13 10.100000 0.9029840 1.0776191 0.7283489
## 14 10.920833 0.9205910 1.0815687 0.7596132
## 15 11.741667 0.9352317 1.0813458 0.7891177
## 16 12.562500 0.9473276 1.0782333 0.8164218
## 17 13.383333 0.9572677 1.0732634 0.8412720
## 18 14.204167 0.9654005 1.0672291 0.8635718
## 19 15.025000 0.9720306 1.0607151 0.8833462
## 20 15.845833 0.9774200 1.0541345 0.9007055
```

```
dev.off()
```

```
## pdf
##   2
```

```
head(margpolicyIFplot)
```

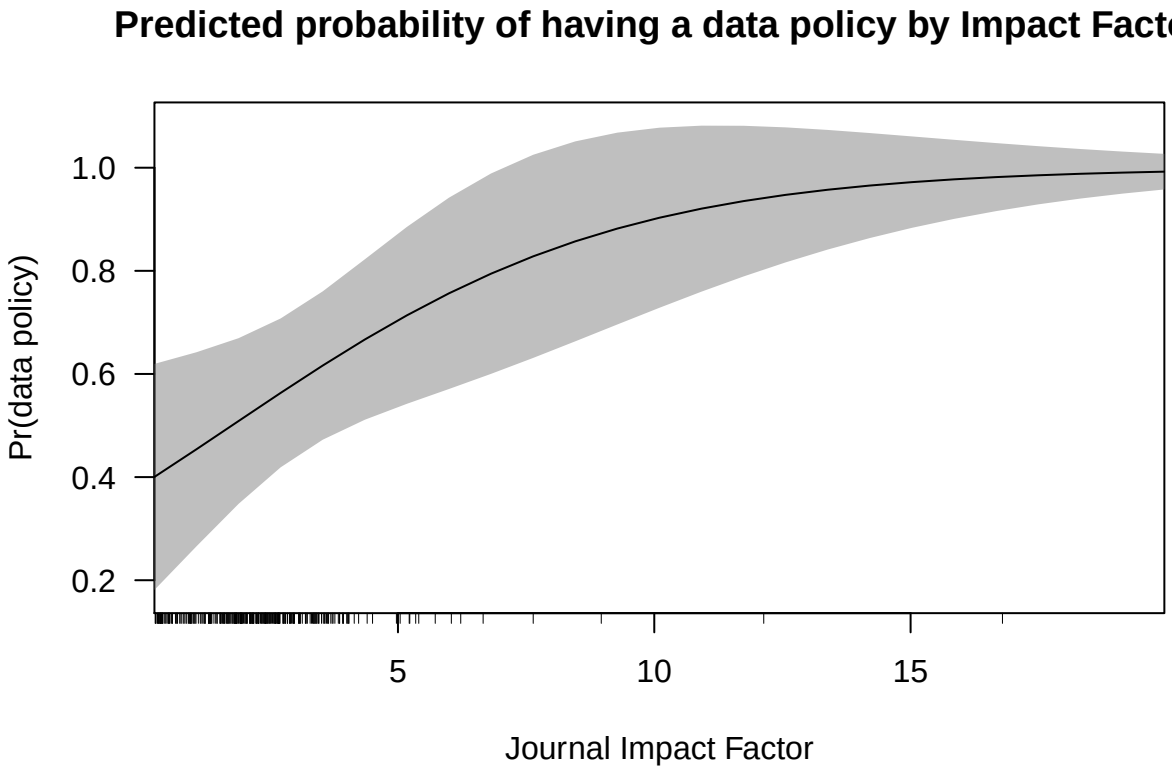
```
##      xvals     yvals     upper     lower
## 1 0.250000 0.4003357 0.6195170 0.1811543
## 2 1.070833 0.4540074 0.6421653 0.2658495
## 3 1.891667 0.5087698 0.6694098 0.3481297
## 4 2.712500 0.5633225 0.7077032 0.4189418
## 5 3.533333 0.6163849 0.7602600 0.4725099
## 6 4.354167 0.6668132 0.8223908 0.5112356
```

```
tail(margpolicyIFplot)
```

```
##       xvals     yvals    upper     lower
## 20 15.84583 0.9774200 1.054134 0.9007055
## 21 16.66667 0.9817904 1.047765 0.9158157
## 22 17.48750 0.9853275 1.041781 0.9288737
## 23 18.30833 0.9881859 1.036283 0.9400890
## 24 19.12917 0.9904928 1.031314 0.9496712
## 25 19.95000 0.9923527 1.026884 0.9578213
```

```
cplot(policymodel, x="IF", xlab ="Journal Impact Factor", main = "Predicted probability of having a data
```

```
##        xvals     yvals     upper     lower
## 1   0.250000 0.4003357 0.6195170 0.1811543
## 2   1.070833 0.4540074 0.6421653 0.2658495
## 3   1.891667 0.5087698 0.6694098 0.3481297
## 4   2.712500 0.5633225 0.7077032 0.4189418
## 5   3.533333 0.6163849 0.7602600 0.4725099
## 6   4.354167 0.6668132 0.8223908 0.5112356
## 7   5.175000 0.7136916 0.8852434 0.5421398
## 8   5.995833 0.7563840 0.9419130 0.5708550
## 9   6.816667 0.7945430 0.9888662 0.6002199
## 10  7.637500 0.8280835 1.0250703 0.6310967
## 11  8.458333 0.8571331 1.0510118 0.6632545
## 12  9.279167 0.8819737 1.0679856 0.6959618
## 13 10.100000 0.9029840 1.0776191 0.7283489
## 14 10.920833 0.9205910 1.0815687 0.7596132
## 15 11.741667 0.9352317 1.0813458 0.7891177
## 16 12.562500 0.9473276 1.0782333 0.8164218
## 17 13.383333 0.9572677 1.0732634 0.8412720
## 18 14.204167 0.9654005 1.0672291 0.8635718
## 19 15.025000 0.9720306 1.0607151 0.8833462
## 20 15.845833 0.9774200 1.0541345 0.9007055
```



Let's do the same thing but with strictness. We're limitting this, again, to journals with a policy

```
## we now run this as a logit model with just the journals that have a policy

jpolicy_with_policy_dedup %>% dplyr::select(bin_strictness) %>%summary()
```

```
##        bin_strictness
##  Don't require:102
##  Require      : 53
```

```
strictnessmodel <- glm(bin_strictness  ~ discipline + IF, data = jpolicy_with_policy_dedup, family = bi
summary(strictnessmodel)
```

```
##
## Call:
## glm(formula = bin_strictness ~ discipline + IF, family = binomial,
##     data = jpolicy_with_policy_dedup)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0543  -0.9051  -0.6050   1.1493   1.9001
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -1.89611    0.53389  -3.551 0.000383 ***
## disciplineEconomics        1.34057    0.61027   2.197 0.028042 *
## disciplineHistory        -15.75684 1318.23513  -0.012 0.990463
## disciplinePoliSci + IR     1.46383    0.61630   2.375 0.017540 *
```

```
## disciplinePsychology      0.63818    0.66425   0.961 0.336675
## disciplineSociology       0.07181    0.74426   0.096 0.923139
## IF                        0.16234    0.09970   1.628 0.103474
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 199.12  on 154  degrees of freedom
## Residual deviance: 174.65  on 148  degrees of freedom
## AIC: 188.65
##
## Number of Fisher Scoring iterations: 16
```

The effect of IF on the strictness of data is quite weak if at all existent. Let's try the same with ranking, which measures the same thing but gives much less impact to outliers at the top.

We create the figure as both Cplot and ggplot, but will used the ggplot as **Figure 4a** in the table.

```
policyRankingmodel <- glm(has_policy  ~ discipline + ranking, data = jpolicy_dedup, family = binomial)
summary(policyRankingmodel)
```

```
##
## Call:
## glm(formula = has_policy ~ discipline + ranking, family = binomial,
##     data = jpolicy_dedup)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8834  -1.1534   0.6815   0.9546   2.0421
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              0.900339   0.375178   2.400   0.0164 *
## disciplineEconomics      0.728109   0.446363   1.631   0.1028
## disciplineHistory       -1.890249   0.474232  -3.986 6.72e-05 ***
## disciplinePoliSci + IR   0.206944   0.424601   0.487   0.6260
## disciplinePsychology     0.035946   0.413962   0.087   0.9308
## disciplineSociology     -0.632503   0.415821  -1.521   0.1282
## ranking                 -0.020476   0.008902  -2.300   0.0214 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 402.17  on 290  degrees of freedom
## Residual deviance: 356.20  on 284  degrees of freedom
## AIC: 370.2
##
## Number of Fisher Scoring iterations: 4
```

```
pdf("Effects of Journal Rank on Data Policy.pdf", width = 11, height = 8)
margpolicyplot <- cplot(policyRankingmodel, x="ranking", se.type="shade")
```

```
##        xvals     yvals     upper     lower
## 1   1.000000 0.7141874 0.8617163 0.5666585
```

```
## 2   3.041667 0.7055783 0.8513087 0.5598479
## 3   5.083333 0.6968200 0.8407914 0.5528485
## 4   7.125000 0.6879163 0.8302038 0.5456289
## 5   9.166667 0.6788717 0.8195879 0.5381556
## 6  11.208333 0.6696909 0.8089884 0.5303933
## 7  13.250000 0.6603788 0.7984517 0.5223059
## 8  15.291667 0.6509411 0.7880250 0.5138572
## 9  17.333333 0.6413835 0.7777550 0.5050121
## 10 19.375000 0.6317123 0.7676868 0.4957379
## 11 21.416667 0.6219341 0.7578623 0.4860058
## 12 23.458333 0.6120556 0.7483188 0.4757924
## 13 25.500000 0.6020842 0.7390877 0.4650806
## 14 27.541667 0.5920273 0.7301933 0.4538613
## 15 29.583333 0.5818927 0.7216519 0.4421335
## 16 31.625000 0.5716885 0.7134719 0.4299052
## 17 33.666667 0.5614230 0.7056533 0.4171928
## 18 35.708333 0.5511047 0.6981886 0.4040207
## 19 37.750000 0.5407421 0.6910637 0.3904206
## 20 39.791667 0.5303442 0.6842584 0.3764301
```

```
dev.off()
```

```
## pdf
##   2
```

```
margpolicyggplot <- ggplot(margpolicyplot, aes(x = xvals)) +
  geom_line(aes(y = yvals)) +
  geom_line(aes(y = upper), linetype = 2) +
  geom_line(aes(y = lower), linetype = 2) +
  geom_hline(yintercept = 0) +
  ggtitle("Predicted probability of having a journal policy by journal rank") +
  xlab("Journal rank") + ylab("Predicted probability") +
  theme(text = element_text(family = "Open Sans"), panel.grid = element_blank(), panel.background = ele
ggsave("Pr-policy by rank.png", plot = margpolicyggplot)
```

```
## Saving 6.5 x 4.5 in image
```

```
head(margpolicyplot)
```

```
##       xvals     yvals     upper     lower
## 1  1.000000 0.7141874 0.8617163 0.5666585
## 2  3.041667 0.7055783 0.8513087 0.5598479
## 3  5.083333 0.6968200 0.8407914 0.5528485
## 4  7.125000 0.6879163 0.8302038 0.5456289
## 5  9.166667 0.6788717 0.8195879 0.5381556
## 6 11.208333 0.6696909 0.8089884 0.5303933
```

```
tail(margpolicyplot)
```

```
##       xvals     yvals     upper     lower
## 20 39.79167 0.5303442 0.6842584 0.3764301
## 21 41.83333 0.5199199 0.6777483 0.3620916
## 22 43.87500 0.5094783 0.6715055 0.3474511
## 23 45.91667 0.4990283 0.6654998 0.3325569
## 24 47.95833 0.4885792 0.6596998 0.3174586
## 25 50.00000 0.4781401 0.6540738 0.3022063
```

We do the same for strictness. The ggplot version of the plot if **Figure 4b** in the paper.

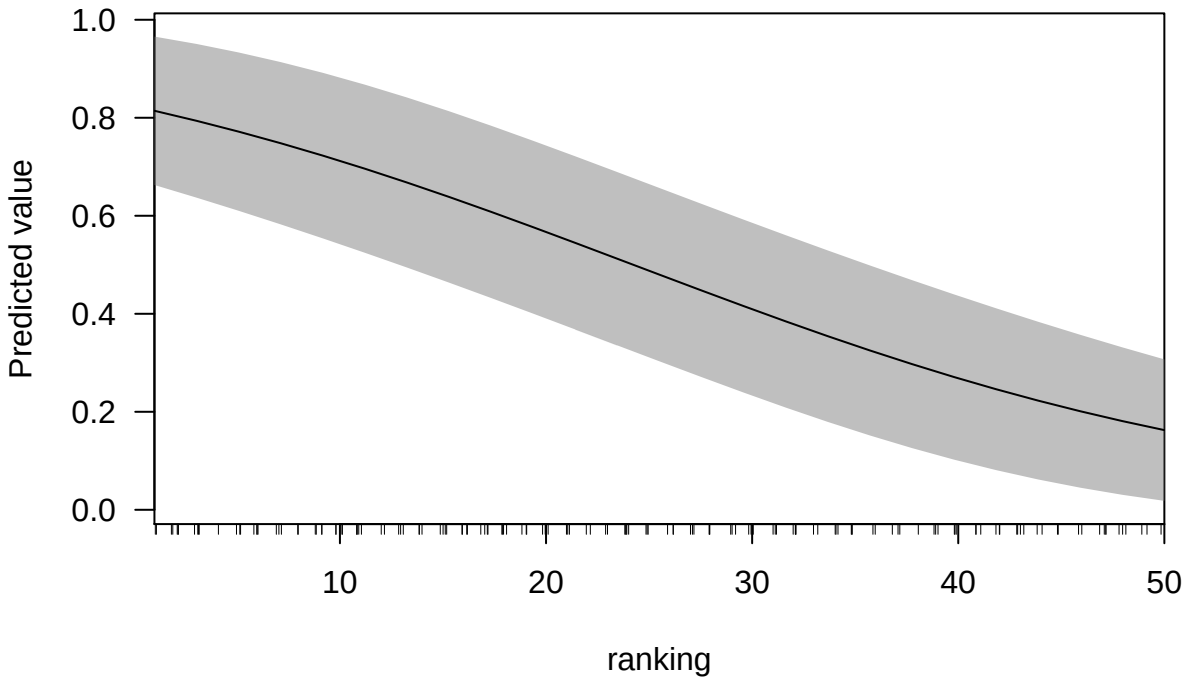
```
strictnessRankingmodel <- glm(bin_strictness  ~ discipline + ranking, data = jpolicy_with_policy_dedup,
summary(strictnessRankingmodel)
```

```
##
## Call:
## glm(formula = bin_strictness ~ discipline + ranking, family = binomial,
##     data = jpolicy_with_policy_dedup)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6926  -0.8591  -0.4058   0.8694   1.9570
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -0.41980    0.56126  -0.748  0.45449
## disciplineEconomics        1.96121    0.64723   3.030  0.00244 **
## disciplineHistory        -15.57536 1193.15348  -0.013  0.98958
## disciplinePoliSci + IR     1.89888    0.66514   2.855  0.00431 **
## disciplinePsychology       1.18059    0.66035   1.788  0.07380 .
## disciplineSociology        0.27687    0.78459   0.353  0.72418
## ranking                   -0.06361    0.01560  -4.077 4.56e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 199.12  on 154  degrees of freedom
## Residual deviance: 157.94  on 148  degrees of freedom
## AIC: 171.94
##
## Number of Fisher Scoring iterations: 16
```

```
margstrict <- margins(strictnessRankingmodel)
margstrictplot <- cplot(strictnessRankingmodel, x="ranking", se.type="shade")
```

```
##        xvals     yvals     upper     lower
## 1   1.000000 0.8142412 0.9656633 0.6628191
## 2   3.041667 0.7937933 0.9505978 0.6369888
## 3   5.083333 0.7717255 0.9332731 0.6101779
## 4   7.125000 0.7480457 0.9136415 0.5824499
## 5   9.166667 0.7227918 0.8917160 0.5538676
## 6  11.208333 0.6960353 0.8675795 0.5244910
## 7  13.250000 0.6678828 0.8413872 0.4943784
## 8  15.291667 0.6384772 0.8133640 0.4635904
## 9  17.333333 0.6079962 0.7837945 0.4321978
## 10 19.375000 0.5766492 0.7530052 0.4002932
## 11 21.416667 0.5446729 0.7213417 0.3680040
## 12 23.458333 0.5123239 0.6891434 0.3355043
## 13 25.500000 0.4798713 0.6567206 0.3030220
## 14 27.541667 0.4475877 0.6243375 0.2708378
## 15 29.583333 0.4157400 0.5922052 0.2392747
## 16 31.625000 0.3845807 0.5604825 0.2086790
## 17 33.666667 0.3543405 0.5292849 0.1793961
```

```
## 18 35.708333 0.3252216 0.4986978 0.1517453
## 19 37.750000 0.2973933 0.4687883 0.1259984
## 20 39.791667 0.2709900 0.4396167 0.1023633
```




```
head(margstrictplot)
```

```
##       xvals     yvals     upper     lower
## 1  1.000000 0.8142412 0.9656633 0.6628191
## 2  3.041667 0.7937933 0.9505978 0.6369888
## 3  5.083333 0.7717255 0.9332731 0.6101779
## 4  7.125000 0.7480457 0.9136415 0.5824499
## 5  9.166667 0.7227918 0.8917160 0.5538676
## 6 11.208333 0.6960353 0.8675795 0.5244910
```

```
tail(margstrictplot)
```

```
##       xvals     yvals     upper      lower
## 20 39.79167 0.2709900 0.4396167 0.10236330
## 21 41.83333 0.2461097 0.4112427 0.08097675
## 22 43.87500 0.2228156 0.3837286 0.06190259
## 23 45.91667 0.2011380 0.3571391 0.04513684
## 24 47.95833 0.1810780 0.3315394 0.03061660
## 25 50.00000 0.1626114 0.3069914 0.01823140
```

```
margstrictplotgg <- ggplot(margstrictplot, aes(x = xvals)) +
  geom_line(aes(y = yvals)) +
  geom_line(aes(y = upper), linetype = 2) +
  geom_line(aes(y = lower), linetype = 2) +
```

```
  geom_hline(yintercept = 0) +
  ggtitle("Predicted probability of required data sharing by journal rank") +
  xlab("Journal ranking") + ylab("Predicted Probability") +
  theme(text = element_text(family = "Open Sans"), panel.grid = element_blank(), panel.background = ele

ggsave("Probability Strict by ranking.png", plot = margstrictplotgg)
```

```
## Saving 6.5 x 4.5 in image
```

```
png("Effects of Journal Rank on Strictness of  Data Policy.png")
margstrictplot <- cplot(strictnessRankingmodel, x="ranking", se.type="shade")
```

```
##        xvals     yvals     upper     lower
## 1   1.000000 0.8142412 0.9656633 0.6628191
## 2   3.041667 0.7937933 0.9505978 0.6369888
## 3   5.083333 0.7717255 0.9332731 0.6101779
## 4   7.125000 0.7480457 0.9136415 0.5824499
## 5   9.166667 0.7227918 0.8917160 0.5538676
## 6  11.208333 0.6960353 0.8675795 0.5244910
## 7  13.250000 0.6678828 0.8413872 0.4943784
## 8  15.291667 0.6384772 0.8133640 0.4635904
## 9  17.333333 0.6079962 0.7837945 0.4321978
## 10 19.375000 0.5766492 0.7530052 0.4002932
## 11 21.416667 0.5446729 0.7213417 0.3680040
## 12 23.458333 0.5123239 0.6891434 0.3355043
## 13 25.500000 0.4798713 0.6567206 0.3030220
## 14 27.541667 0.4475877 0.6243375 0.2708378
## 15 29.583333 0.4157400 0.5922052 0.2392747
## 16 31.625000 0.3845807 0.5604825 0.2086790
## 17 33.666667 0.3543405 0.5292849 0.1793961
## 18 35.708333 0.3252216 0.4986978 0.1517453
## 19 37.750000 0.2973933 0.4687883 0.1259984
## 20 39.791667 0.2709900 0.4396167 0.1023633
```

```
dev.off()
```

```
## pdf
##   2
```

Journal Age

Let's see if journal age has an impact, controlling for discipline

```
policyRankingmodel <- glm(has_policy  ~ discipline + age, data = jpolicy_dedup, family = binomial)
summary(policyRankingmodel)
```

```
##
## Call:
## glm(formula = has_policy ~ discipline + age, family = binomial,
##     data = jpolicy_dedup)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7585  -1.1457   0.7139   0.9784   1.8749
##
```

```
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              0.679533   0.361986   1.877   0.0605 .
## disciplineEconomics      0.690108   0.444615   1.552   0.1206
## disciplineHistory       -1.894557   0.472187  -4.012 6.01e-05 ***
## disciplinePoliSci + IR   0.191111   0.422069   0.453   0.6507
## disciplinePsychology    -0.024216   0.413153  -0.059   0.9533
## disciplineSociology     -0.653626   0.413437  -1.581   0.1139
## age                     -0.006308   0.004359  -1.447   0.1479
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 402.17  on 290  degrees of freedom
## Residual deviance: 359.49  on 284  degrees of freedom
## AIC: 373.49
##
## Number of Fisher Scoring iterations: 4
```

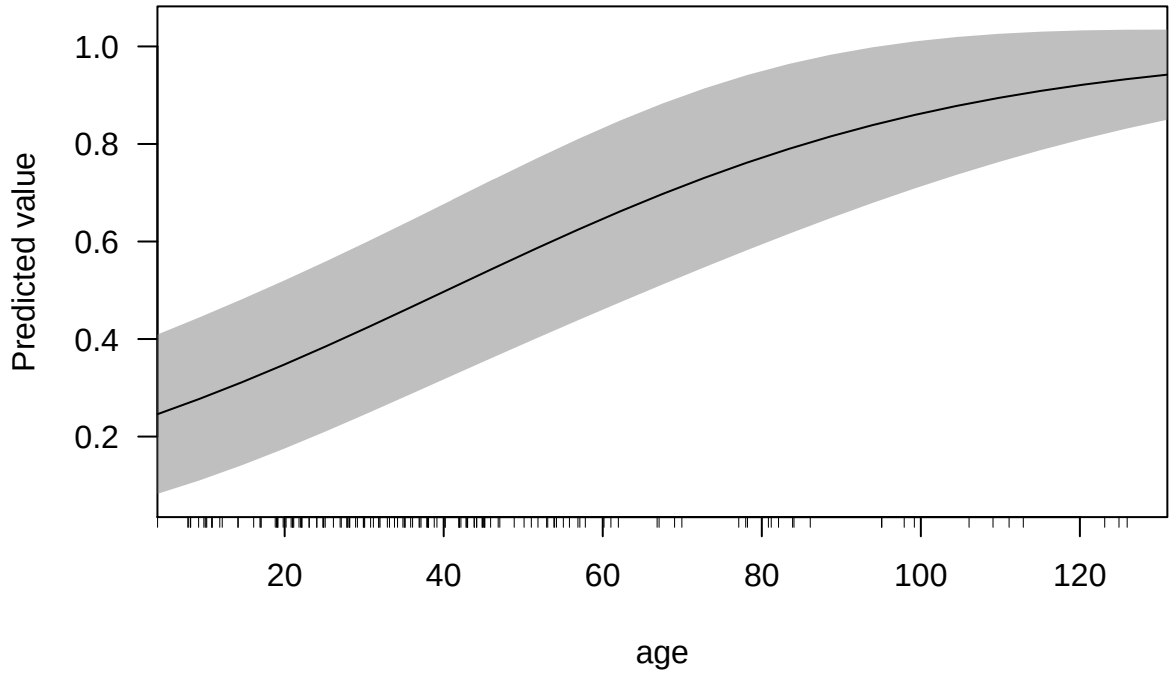
That's a no. However, there's a sizable effect of age on strictness. The ggplot version of the below is **Figure 5**

```
strictnessAgemodel <- glm(bin_strictness  ~ discipline + age, data = jpolicy_with_policy_dedup, family =
summary(strictnessAgemodel)
```

```
##
## Call:
## glm(formula = bin_strictness ~ discipline + age, family = binomial,
##     data = jpolicy_with_policy_dedup)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4791  -0.9167  -0.4857   0.9355   2.2622
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -3.13502    0.68946  -4.547 5.44e-06 ***
## disciplineEconomics       1.89105    0.65215   2.900 0.003735 **
## disciplineHistory       -15.26386 1285.66810  -0.012 0.990527
## disciplinePoliSci + IR    1.87184    0.65440   2.860 0.004231 **
## disciplinePsychology      1.41898    0.67432   2.104 0.035353 *
## disciplineSociology       0.04086    0.78267   0.052 0.958370
## age                       0.03079    0.00824   3.737 0.000186 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 199.12  on 154  degrees of freedom
## Residual deviance: 160.14  on 148  degrees of freedom
## AIC: 174.14
##
## Number of Fisher Scoring iterations: 16
```

```
strictnessAgePlot <- cplot(strictnessAgemodel, data = jpolicy_with_policy_dedup, x="age", se.type="shad
```

```
##         xvals     yvals     upper      lower
## 1    4.000000 0.2458635 0.4092876 0.08243939
## 2    9.291667 0.2773105 0.4445366 0.11008435
## 3   14.583333 0.3111203 0.4812709 0.14096958
## 4   19.875000 0.3470725 0.5194889 0.17465614
## 5   25.166667 0.3848583 0.5591649 0.21055168
## 6   30.458333 0.4240859 0.6001941 0.24797765
## 7   35.750000 0.4642939 0.6423280 0.28625991
## 8   41.041667 0.5049716 0.6851251 0.32481810
## 9   46.333333 0.5455836 0.7279422 0.36322495
## 10  51.625000 0.5855979 0.7699784 0.40121747
## 11  56.916667 0.6245138 0.8103640 0.43866352
## 12  62.208333 0.6618852 0.8482660 0.47550431
## 13  67.500000 0.6973396 0.8829821 0.51169713
## 14  72.791667 0.7305896 0.9140042 0.54717508
## 15  78.083333 0.7614364 0.9410438 0.58182908
## 16  83.375000 0.7897677 0.9640262 0.61550924
## 17  88.666667 0.8155495 0.9830602 0.64803869
## 18  93.958333 0.8388147 0.9983968 0.67923265
## 19  99.250000 0.8596505 1.0103837 0.70891734
## 20 104.541667 0.8781841 1.0194231 0.73694504
```




```
head(strictnessAgePlot)
```

```
##       xvals     yvals     upper      lower
```

```
## 1  4.000000 0.2458635 0.4092876 0.08243939
## 2  9.291667 0.2773105 0.4445366 0.11008435
## 3 14.583333 0.3111203 0.4812709 0.14096958
## 4 19.875000 0.3470725 0.5194889 0.17465614
## 5 25.166667 0.3848583 0.5591649 0.21055168
## 6 30.458333 0.4240859 0.6001941 0.24797765
```

```
tail(strictnessAgePlot)
```

```
##       xvals     yvals    upper     lower
## 20 104.5417 0.8781841 1.019423 0.7369450
## 21 109.8333 0.8945704 1.025937 0.7632041
## 22 115.1250 0.9089809 1.030338 0.7876236
## 23 120.4167 0.9215944 1.033015 0.8101735
## 24 125.7083 0.9325895 1.034317 0.8308616
## 25 131.0000 0.9421396 1.034551 0.8497282
```

```
png("Effects of Journal Rank on Strictness of  Data Policy.png")
cplot(strictnessAgemodel, x="age", se.type="shade")
```

```
##         xvals     yvals     upper      lower
## 1    4.000000 0.2458635 0.4092876 0.08243939
## 2    9.291667 0.2773105 0.4445366 0.11008435
## 3   14.583333 0.3111203 0.4812709 0.14096958
## 4   19.875000 0.3470725 0.5194889 0.17465614
## 5   25.166667 0.3848583 0.5591649 0.21055168
## 6   30.458333 0.4240859 0.6001941 0.24797765
## 7   35.750000 0.4642939 0.6423280 0.28625991
## 8   41.041667 0.5049716 0.6851251 0.32481810
## 9   46.333333 0.5455836 0.7279422 0.36322495
## 10  51.625000 0.5855979 0.7699784 0.40121747
## 11  56.916667 0.6245138 0.8103640 0.43866352
## 12  62.208333 0.6618852 0.8482660 0.47550431
## 13  67.500000 0.6973396 0.8829821 0.51169713
## 14  72.791667 0.7305896 0.9140042 0.54717508
## 15  78.083333 0.7614364 0.9410438 0.58182908
## 16  83.375000 0.7897677 0.9640262 0.61550924
## 17  88.666667 0.8155495 0.9830602 0.64803869
## 18  93.958333 0.8388147 0.9983968 0.67923265
## 19  99.250000 0.8596505 1.0103837 0.70891734
## 20 104.541667 0.8781841 1.0194231 0.73694504
```

```
dev.off()
```

```
## pdf
##   2
```

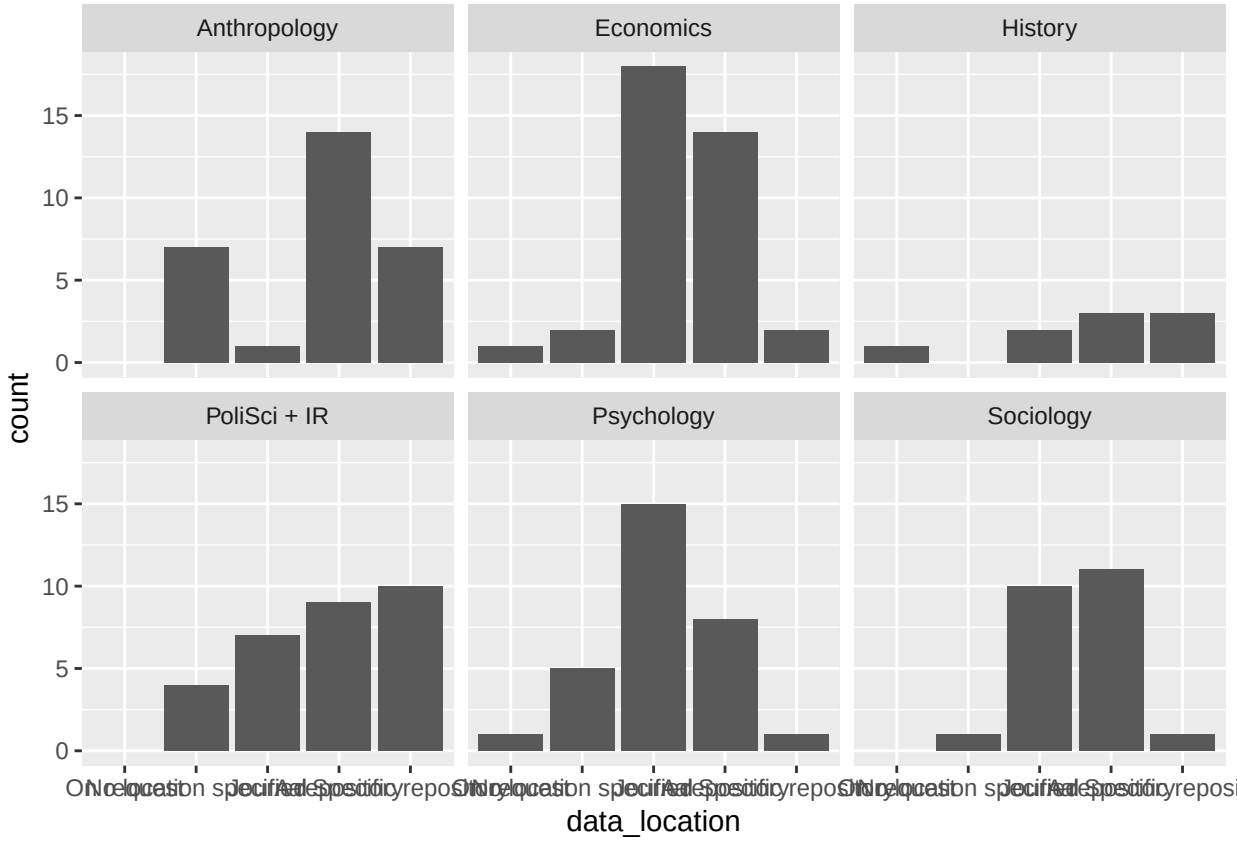
```
strictnessAgePlotgg <- ggplot(strictnessAgePlot, aes(x = xvals)) +
  geom_line(aes(y = yvals)) +
  geom_line(aes(y = upper), linetype = 2) +
  geom_line(aes(y = lower), linetype = 2) +
  geom_hline(yintercept = 0) +
  ggtitle("Predicted probability of required data sharing by journal age") +
  xlab("Journal age (in years)") + ylab("Predicted probability") +
  theme(text = element_text(family = "Open Sans"), panel.grid = element_blank(), panel.background = ele

ggsave("Strictness by Age.png", plot = strictnessAgePlotgg)
```

```
## Saving 6.5 x 4.5 in image
```

Data Location by Discipline

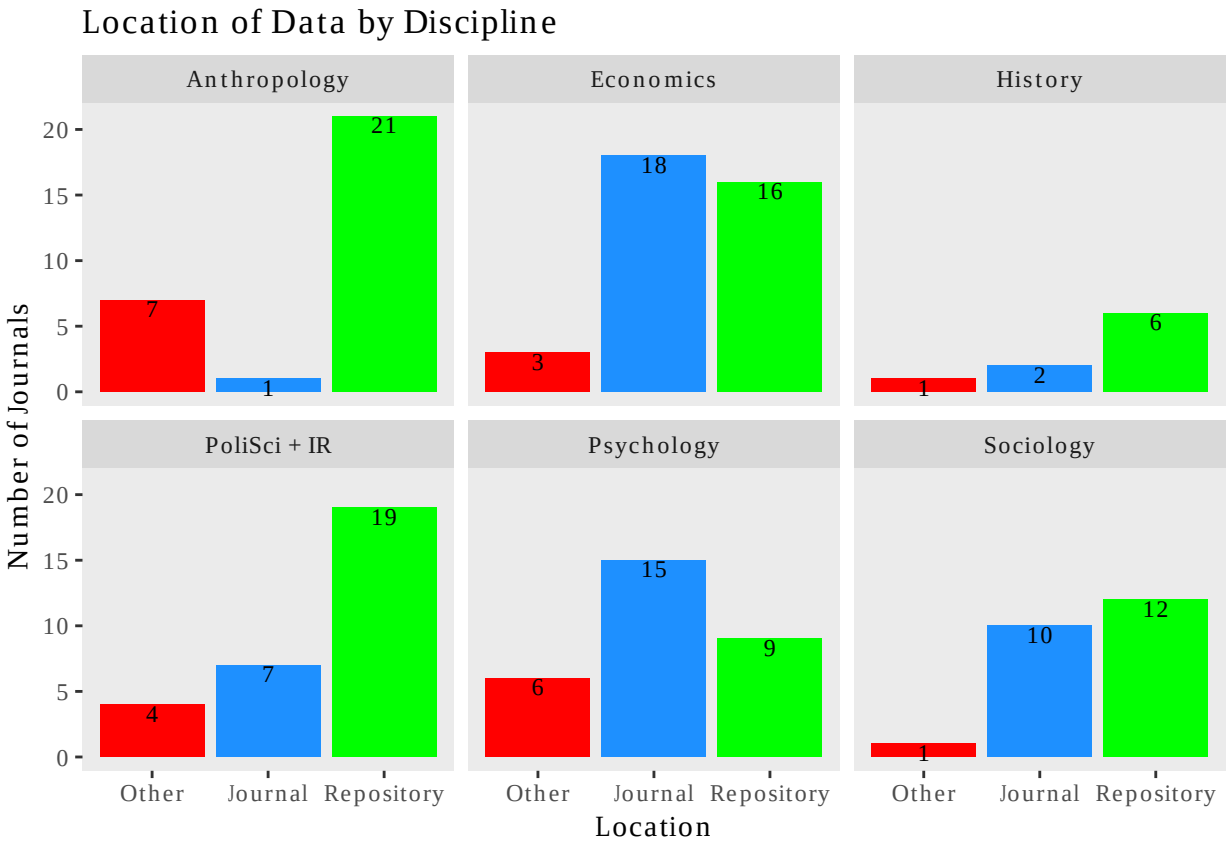
This is the graph using the full location categorization. We're collapsing categories 1-3 as there are hardly any journals in 1 and 3, making this graph unnecessarily cluttered.

```
ggplot(data=jpolicy_with_policy) + geom_bar(mapping = aes(x = data_location)) + facet_wrap(~ discipline
```



We have above recoded the location variable and this code produces *figure 6* as well as the numbers given in the "Where do journals tell authors to share data?" section of the analysis.

```
datalocDisciplinePlot <- ggplot(data=jpolicy_with_policy, aes(data_location_simple, group="discipline")
  geom_bar(aes(x = data_location_simple, fill = factor (..x..)), stat="count") +
  geom_text(aes(label = ..count..), stat= "count", family= "Open Sans", size = 3, vjust = 1) +
  guides(fill=FALSE) +
  scale_fill_manual(values = c("red", "dodgerblue", "green")) +
  labs(title ="Location of Data by Discipline", y = "Number of Journals", x = "Location") +
  facet_wrap(~ discipline, nrow = 2)+
  theme(text = element_text(family = "Open Sans"), strip.text = element_text(face = "bold"), panel.grid
print(datalocDisciplinePlot)
```



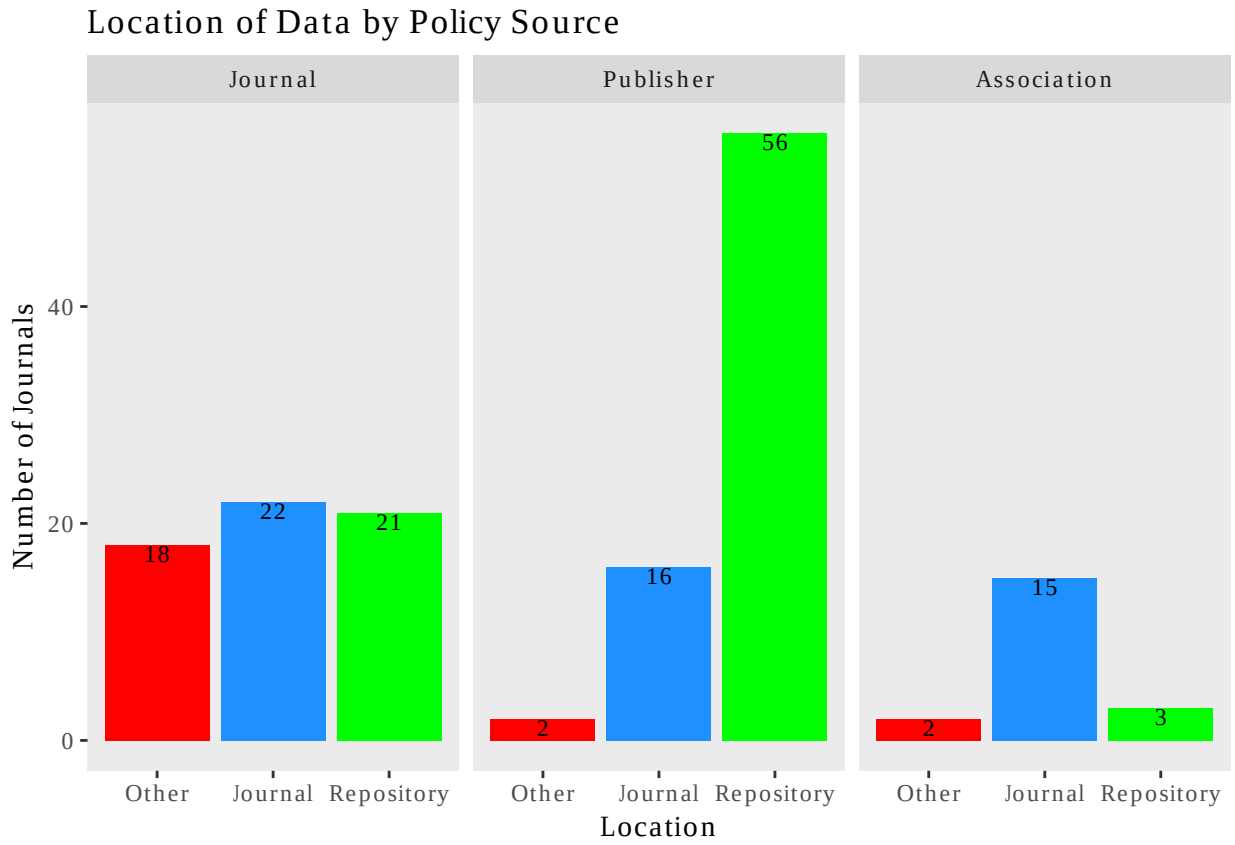

```
ggsave("Location of Data by Discipline.png", plot = datalocDisciplinePlot)
```

```
## Saving 6.5 x 4.5 in image
```

Data Location by policy source

This produces **figure 7**

```
datalocSourcePlot <- ggplot(data=jpolicy_with_policy_dedup, aes(data_location_simple, group="source")) +
  geom_bar(aes(x = data_location_simple, fill = factor (..x..)), stat="count") +
  geom_text(aes(label = ..count..), stat= "count", family= "Open Sans", size = 3, vjust = 1) +
  guides(fill=FALSE) +
  scale_fill_manual(values = c("red", "dodgerblue", "green")) +
  labs(title ="Location of Data by Policy Source", y = "Number of Journals", x = "Location") +
  facet_wrap(~ source) +
  theme(text = element_text(family = "Open Sans"), strip.text = element_text(face = "bold"), panel.grid

print(datalocSourcePlot)
```




```
ggsave("Location of Data by Policy Source.png", plot = datalocSourcePlot)
```

```
## Saving 6.5 x 4.5 in image
```

Data Strictness by TOPS

Let's first look at whether TOP signers are more likely to have a policy. This and the following tables produce the numbers reported in the section "Effect of TOP and DA-RT on data policies"

```
table(jpolicy_dedup$TOPS, distinct(jpolicy, title, .keep_all = TRUE)$has_policy)
```

```
##
##        No Yes
##   No  133 133
##   Yes   3  22
```

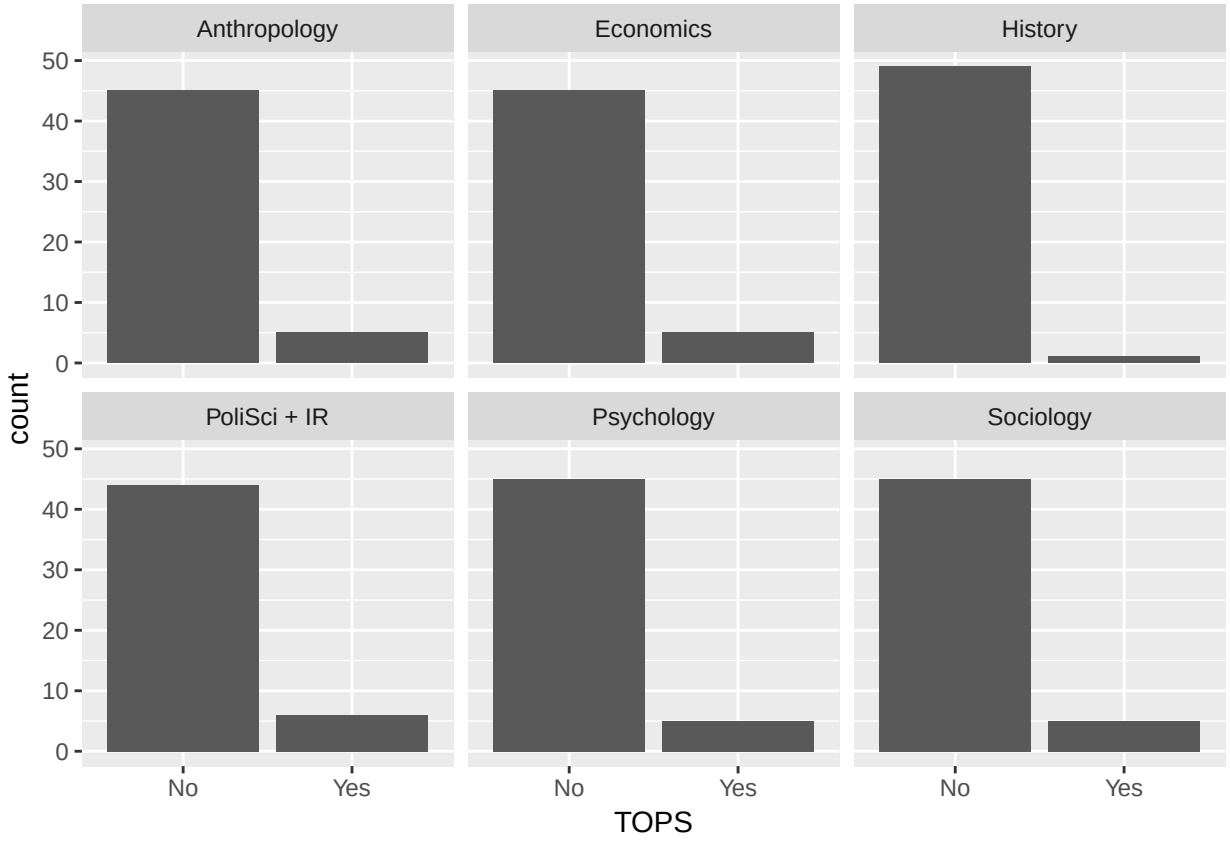
Among those journals with a data policy, does TOPS correlate with a stronger policy? We're using the binary variable we generated above

```
table(jpolicy_with_policy_dedup$TOPS, jpolicy_with_policy_dedup$bin_strictness, useNA = "no")
```

```
##
##       Don't require Require
##   No             86      47
##   Yes            16       6
```

We did check TOP journals by discipline to see if this was driving the results. It is not and we don't report this in the paper

```
ggplot(data=jpolicy) + geom_bar(mapping = aes(x = TOPS))+ facet_wrap(~ discipline, nrow = 2)
```



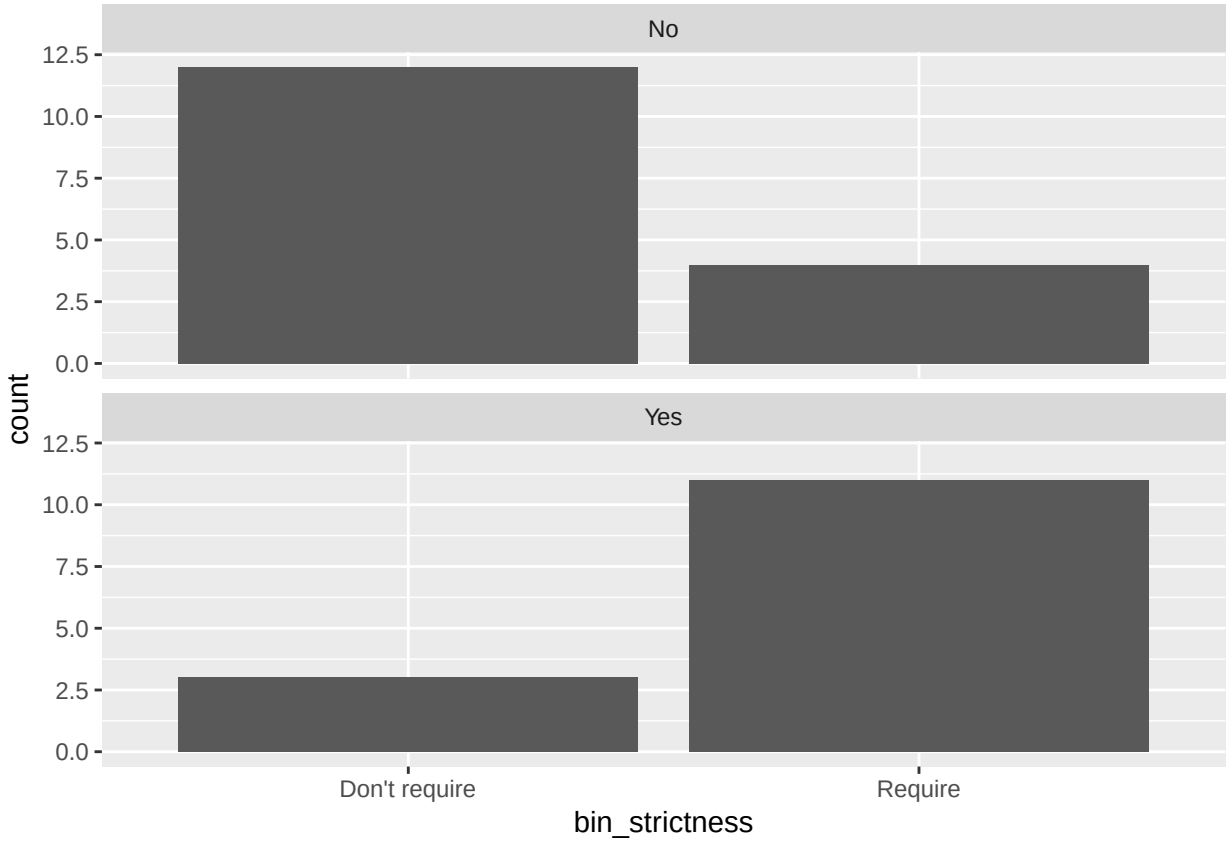
DA-RT

Let's see if DA-RT also affects the strictness of policy among journals who have one at all:

```
table(jpolicy_with_policy$DART, jpolicy_with_policy$bin_strictness, useNA = "no")
```

```
##
##       Don't require Require
##   No             12       4
##   Yes             3      11
```

```
jpolicy_with_policy %>%
  drop_na(DART) %>%
  ggplot() + geom_bar(mapping = aes(x = bin_strictness)) + facet_wrap(~ DART, nrow = 2)
```



Here are the tables for JETS/DA-RT. We're running this on the whole dataset since DART is only coded for polisci anyway, so there are no duplicates..

```
table(jpolicy$DART, jpolicy$has_policy, useNA = "no")
```

```
##
##       No Yes
##   No  19  16
##   Yes  1  14
```

Qualitative Data by Discipline

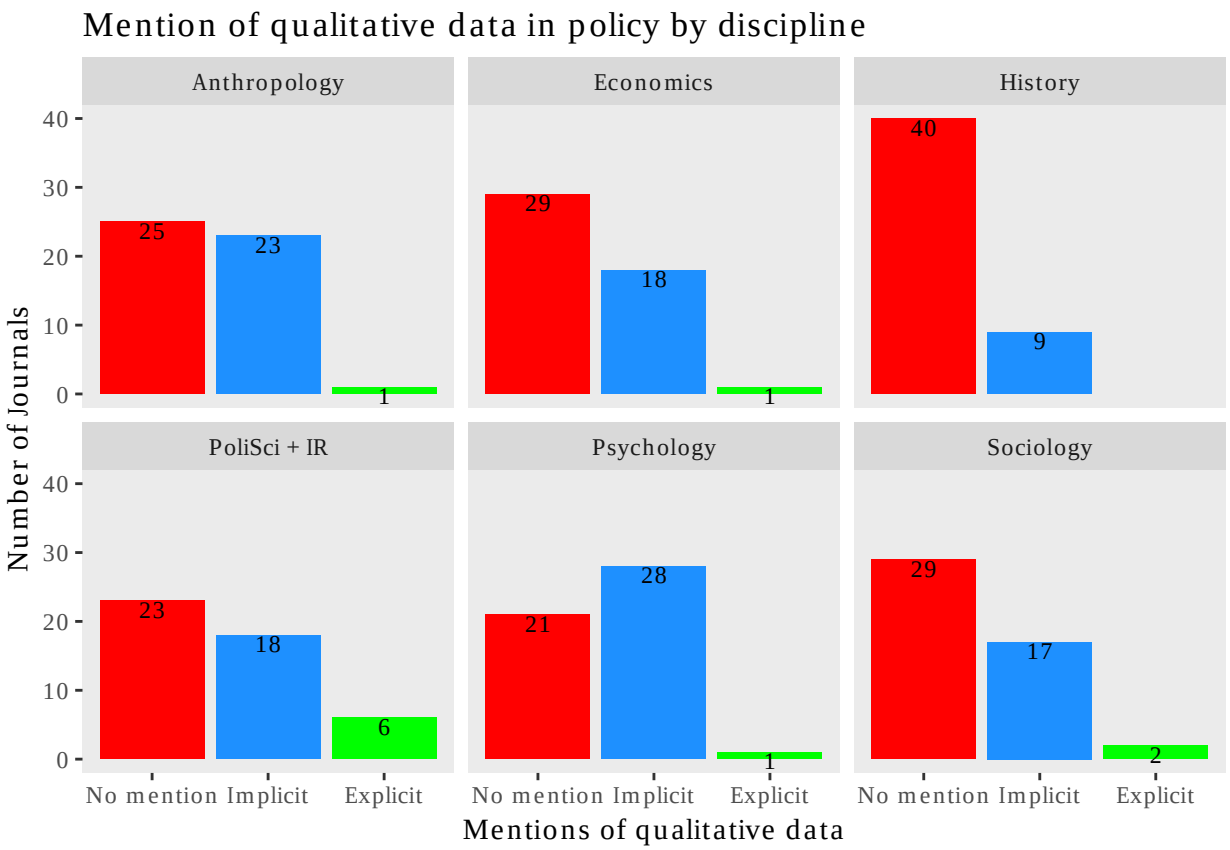
Finally, we're interested to see how many journals talk about qualitative data. We're only looking at the journals that have any policy. The below table and graph are the source for the numbers in the section "Data policies and qualitative data"

```
table(jpolicy_dedup$qual_data)
```

```
##
## No mention   Implicit   Explicit
##        167        113         11
```

```
qualdataPlot <- ggplot(data=jpolicy_dedup, aes(x = qual_data, group="discipline")) +
  geom_bar(aes(x = qual_data, fill = factor (..x..)), stat="count") +
  geom_text(aes(label = ..count..), family= "Open Sans", size = 3, stat= "count", vjust = 1) +
  guides(fill=FALSE) +
  scale_fill_manual(values = c("red", "dodgerblue", "green")) +
```

```
  labs(title ="Mention of qualitative data in policy by discipline", y = "Number of Journals", x = "Men
  facet_wrap(~ discipline, nrow=2)+
  theme(text = element_text(family = "Open Sans"), panel.grid = element_blank())
print(qualdataPlot)
```




```
ggsave("Qualitative Data by Discipline.png", plot = qualdataPlot)
```

```
## Saving 6.5 x 4.5 in image
```

Benefits

This calculates the figures in the "Recommendation" section under "Include the benefits of data sharing"

```
table(jpolicy_with_policy_dedup$benefits, jpolicy_with_policy_dedup$source)
```

```
##
##       No policy Journal Publisher Association
##   No          0      55        34          20
##   Yes         0       6        40           0
```