A sudy of toxic behavior of Reddit users with toxic usernames

Rafal Urbaniak, Patrycja Tempska, Michał Ptaszynski, …

1 Data analysis

Study design and data collection. We used http://stream.pushshift.io (no longer available) and our scripts for processing it to collect two data sets from Reddit, removing users with "bot" in their usernames at this preliminary cleaning stage.

- **Dataset A**: 122k Reddit users with toxic usernames and 122k random Reddit users with neutral usernames coming from one week of monitoring the entire platform between GMT: Wednesday, February 12, 2020 1:27:03 PM and GMT: Saturday, February 22, 2020 2:56:02 PM.
- **Dataset B**: with 207k Reddit users with toxic usernames and 207k random Reddit users with neutral usernames coming from 20 days of monitoring the entire platform between GMT: Saturday, June 20, 2020 00:00:00 AM and GMT: Thursday, July 9, 2020 11:59:59 PM.

We used Samurai Labs tools to evaluate username toxicity with 4 distinct categories (sexual/inappropriate/offensive/profanity) and Samurai Labs models to detect toxic comments in all the content generated by users during the monitoring period, with 6 different categories of toxic content distinguished (profanity, sexual remarks, sexual harassment, personal attacks, bad wish, rejection)—note that these types differ in severity, profanities and sexual remarks being the lightest (see definitions and examples in). It is also noteworthy that both toxic behavior and toxic usernames can overlap—we will have more to say about this later on, but to avoid over-complication, most of the time we will present results for users whose usernames fell into single categories (97% users in both data sets). We do, however, bring up users in overlapping categories near the end of the analysis to illustrate an interesting interaction when we build ensemble yearly predictions.

add ref

Data cleaning. After the usual initial clean-up (loading, renaming columns, converting categorical variables to factors and character variables to strings, adding dummy variables), we inspected the histograms of activity levels in the two datasets (Figure 1).

```
datasetA <- as.data.frame(fread("datasets/usernames_vs_content_nobots.csv"))
types <- as.data.frame(fread(file = 'datasets/usernamesToxicOrNot.tsv',
                             header = FALSE))
colnames(types) <- c("user", "toxic", "V3", "type")
types$toxic <- as.factor(types$toxic)
types$type <- as.factor(types$type)
types$toxicInt <- as.integer(types$toxic)
types$nickInappropriate <- as.integer(grepl("inappropriate", types$type,
                                            fixed = TRUE))
types$nickOffensive <-  as.integer(grepl("offensive", types$type,
                                         fixed = TRUE))
types$nickProfanity <-  as.integer(grepl("profanity", types$type,
                                         fixed = TRUE))
types$nickSexual <-  as.integer(grepl("sexual", types$type,
                                      fixed = TRUE))
colnames(datasetA) <- c("user","comments","pure","legit","profanity",
                        "sexualRemark","personalAttack","sexualHarassment","badWish",
                        "rejection","revealing","extortionInf","blackmail",
                        "antisemitism","intent","prejudice","extortionNude","X")
datasetA$user <- as.character(datasetA$user)
datasetACut <- datasetA %>% select(user, comments, pure, legit, profanity,
                                   personalAttack, sexualRemark,sexualHarassment,
                                   badWish, rejection)

datasetB <- as.data.frame(fread("datasets/20days_nobots.csv"))
typesB <- as.data.frame(fread(file = 'datasets/20days_users_txnd.tsv',
                              header = FALSE))
colnames(typesB) <- c("user", "toxic", "type")
typesB$toxic <- as.factor(typesB$toxic)
typesB$type <- as.factor(typesB$type)
typesB$toxicInt <- as.integer(typesB$toxic)
typesB$nickInappropriate <- as.integer(grepl("inappropriate", typesB$type,
                                             fixed = TRUE))
typesB$nickOffensive <-  as.integer(grepl("offensive", typesB$type,
                                          fixed = TRUE))
typesB$nickProfanity <-  as.integer(grepl("profanity", typesB$type,
                                          fixed = TRUE))
typesB$nickSexual <-  as.integer(grepl("sexual", typesB$type,
                                       fixed = TRUE))

colnames(datasetB) <- c("user","comments","pure","legit","profanity","sexualRemark",
                        "personalAttack","sexualHarassment","badWish","rejection",
                        "revealing","extortionInf","blackmail","antisemitism",
                        "intent","prejudice","extortionNude","X")

datasetB$user <- as.character(datasetB$user)
datasetBCut <- datasetB %>% select(user, comments, pure, legit, profanity,
                            personalAttack, sexualRemark, sexualHarassment,
                            badWish, rejection)

activityA <- ggplot(datasetACut)+geom_histogram(aes(x= comments), bins = 50)+
  xlim(c(0,150))+theme_tufte()+
  theme(plot.title.position = "plot",plot.title = element_text(face = "bold"))+
  ggtitle("User activity (dataset A, 7 days)")+
  labs(subtitle = "x axis restricted to (0,150)")

activityB <- ggplot(datasetBCut)+geom_histogram(aes(x= comments), bins = 50)+
  xlim(c(0,250))+theme_tufte()+
  theme(plot.title.position = "plot",plot.title = element_text(face = "bold"))+
  ggtitle("User activity (dataset B, 20 days)")+
  labs(subtitle = "x axis restricted to (0,250)")
```

In fact, around 0.008% users in dataset A and 0.07% did have higher activity than 700; we suspected them to be mostly bots that the straightforward nickname elimination failed to filter out. This threshold was far off from where the bulk of the distribution was (7.332 standard deviations from the mean in dataset A and 6.246 standard deviations from the mean in dataset B). To get a clearer picture of these outliers we also randomly sampled 100 users from the “above 700” group and manually checked their status; the proportion of accounts that either went missing (suspended or deleted by a user) or turned out bots was very high (45%, see Figure 2).

```
botORnotOutliers <- read.csv("datasets/botORnotOutliers.csv")
#human = 1, bot = 0, missing  = 2
colnames(botORnotOutliers) <- c("user","status")
botORnotOutliers$status <- as.factor(botORnotOutliers$status)
levels(botORnotOutliers$status) <- c("bot","human", "missing")
botORnotOutliers <- botORnotOutliers[,-3]
botORnotTable <- as.data.frame(table(botORnotOutliers$status))
colnames(botORnotTable) = c("status","freq")

outlierPlot <- ggplot(botORnotTable, aes(x =reorder(status,-freq), y = freq))+
  geom_col(alpha = .5)+theme_tufte()+
  ggtitle("Bots and missing accounts among the >700 outliers")+
  theme(plot.title.position = "plot",plot.title = element_text(face = "bold"))+coord_flip()+
  removeX + xlab("")+geom_text(aes(label = paste(freq, "%", sep = "")),
  hjust = 1, nudge_y = -.03, size = 6)
```

```
highA <- datasetACut %>% filter(comments > 700)
highB <- datasetBCut %>% filter(comments > 700)
sampleA <- sample(highA$user, 50, replace = FALSE)
```

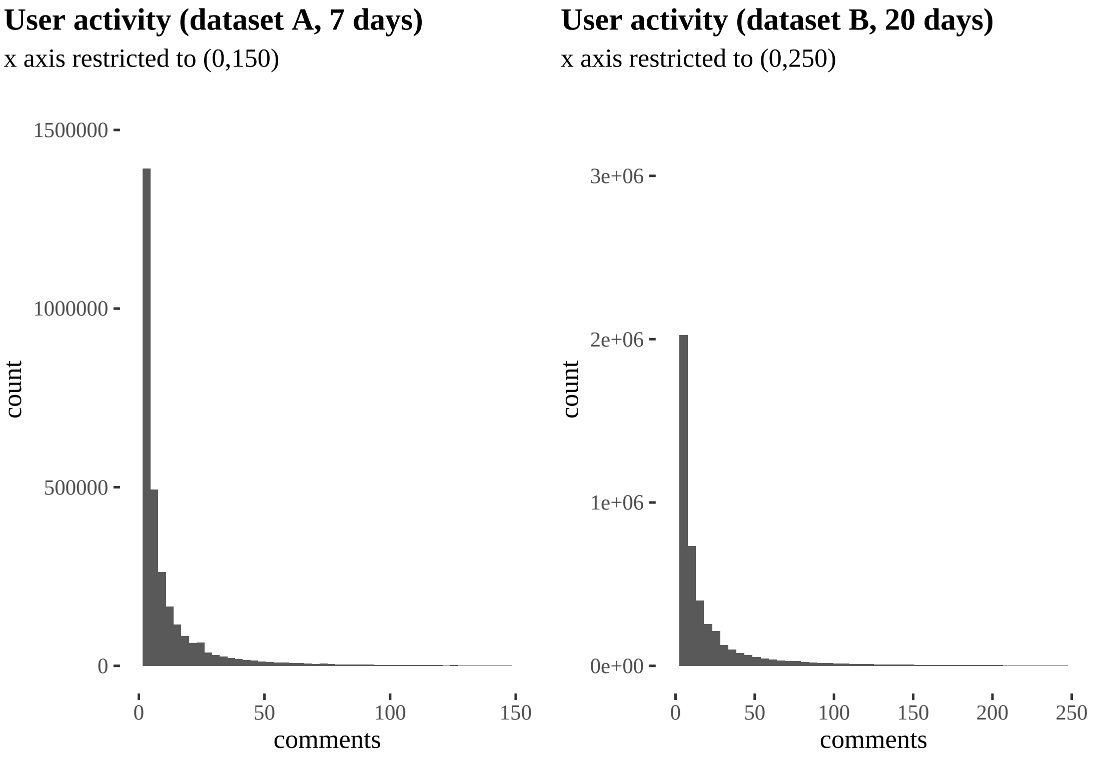


Figure 1: User activity in the two datasets, *x* axes restricted for visibility (the histograms are visually flat above the limits.)

“‘r outlierPlot “‘

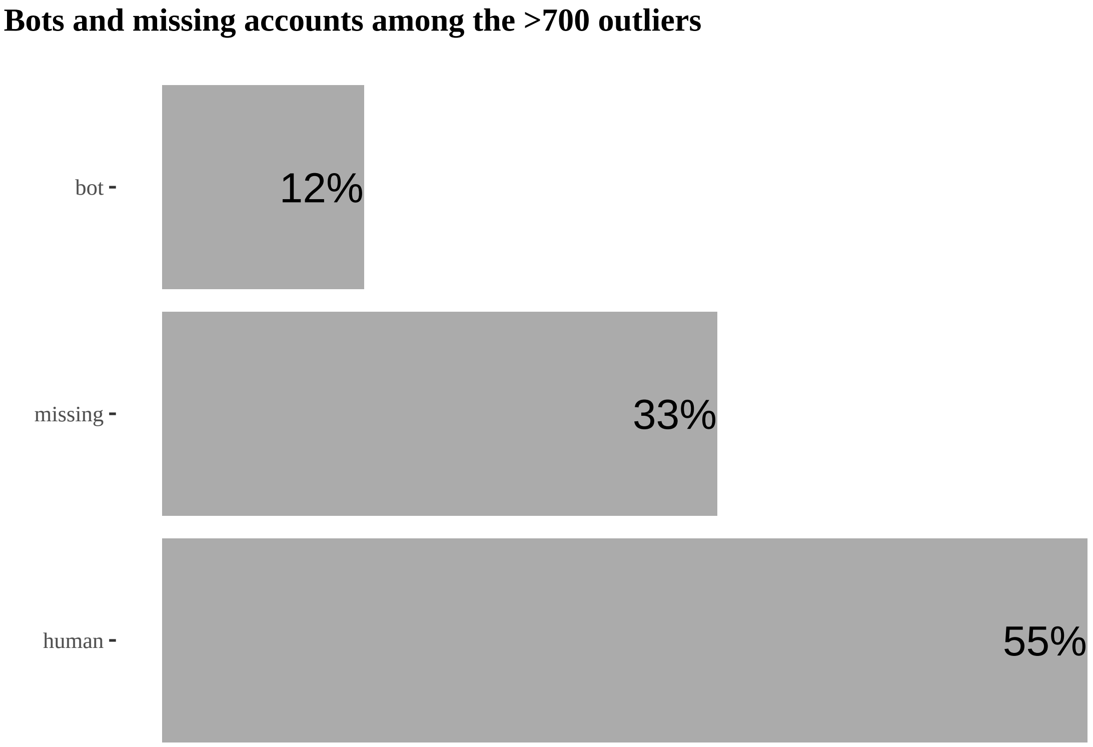


Figure 2: Bots and missing accounts in a random sample of 100 users belonging to the >700 outliers.

```
sampleB <- sample(highB$user, 50, replace = FALSE)
sampleHigh <- data.frame(user = c(sampleA, sampleB))
write_csv(sampleHigh, file = "datasets/sampleHigh.csv")
datasetACut <- datasetACut %>% filter(comments < 700)
datasetBCut <- datasetBCut %>% filter(comments < 700)

jointA <- merge(x = datasetACut, y = types, by = "user",
                all.x = TRUE)
jointB <- merge(x = datasetBCut, y = typesB, by = "user",
                all.x = TRUE)

jointA$nonpure <- jointA$comments - jointA$pure
jointB$nonpure <- jointB$comments - jointB$pure

#saveRDS(jointA, file = "datasets/jointA.rds")
#saveRDS(jointB, file = "datasets/jointB.rds")
```

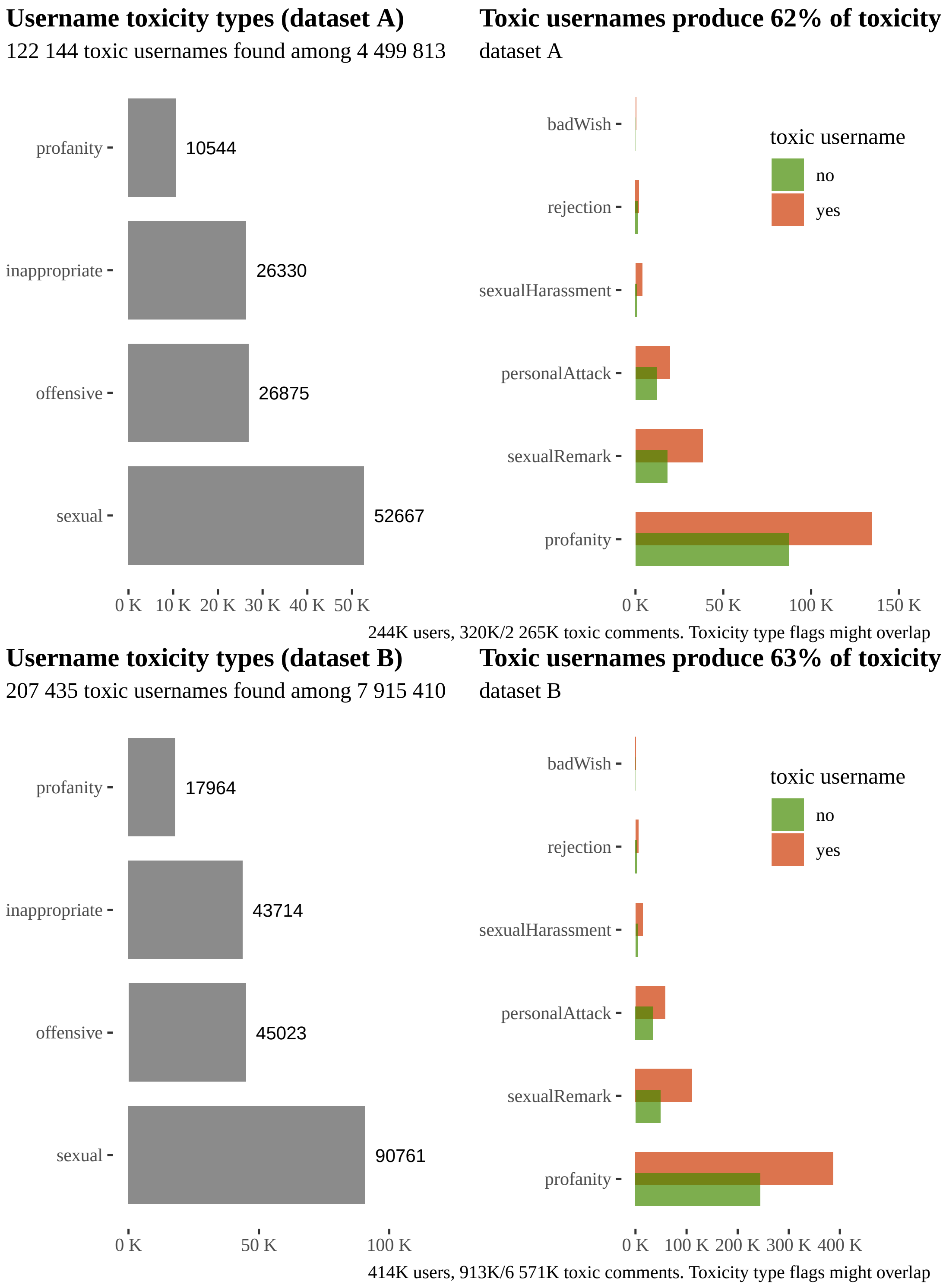
To prevent such unusual users from having impact on the analysis, we decided to exclude them, in effect dropping 5930 out of 12415223 data points. Then we joined the data with nickname toxicity data, obtained by a separate API. For the study of correlation between user nickname toxicity and toxic behavior for each set we filtered all users with toxic usernames and added an equal number of randomly drawn users with non-toxic usernames (otherwise, since only 2.7% (dataset A) and 2.6% (dataset B) users had toxic usernames, we would be vastly expanding the computational cost of Bayesian modelling without any major improvement to uncertainty gauging for the classes of interest). For the study of the correlation between toxic username and suspension we randomly drawn 50k users with toxic usernames and 50k users with non-toxic usernames from each dataset.

```
jointAToxic <- jointA %>% filter(toxicInt == 2)
jointANonToxic <- jointA %>% filter(toxicInt == 1) %>%
  sample_n(size = nrow(jointAToxic))
jointASample <- rbind(jointAToxic,jointANonToxic)

#saveRDS(jointSample, file = "datasets/jointASample.rds")

suspensionsAToxic <- sample_n(jointAToxic, 50000)
suspensionsSNonToxic <- sample_n(jointANonToxic, 50000)

#write.csv(suspensions,"datasets/suspensions.csv", row.names = TRUE)

jointBToxic <- jointB %>% filter(toxicInt == 2)
jointBNonToxic <- jointB %>% filter(toxicInt == 1) %>%
  sample_n(size = nrow(jointBToxic))
jointBSample <- rbind(jointBToxic,jointBNonToxic)

#saveRDS(jointBSample, file = "datasets/jointBSample.rds")

suspensionsBToxic <- sample_n(jointBToxic, 50000)
suspensionsBNonToxic <- sample_n(jointBNonToxic, 50000)
suspensionsB <- rbind(suspensionsBToxic,suspensionsBNonToxic)
#write.csv(suspensionsB,"datasets/suspensionsB.csv", row.names = TRUE)
```

Exploratory analysis. We first inspect the distribution of the username toxicity types in both data sets and the distribution of toxic content divided by whether the users producing it had toxic usernames (Figure 3). Note that while 50% users in each dataset did not have toxic usernames, users with toxic names in fact generated 62% (dataset A) and 64% (dataset B) of toxic content.

```
#code generating the exploratory visualizations is in a separate file
source("scripts/group4.R")
```

Bayesian model building for toxic content production. We used the data sets and the rethinking package (McElreath, 2018) to build Bayesian models studying the correlations between username toxicity and the amount of toxic content produced by the users. Prior to the analysis all the numerical variables used have been standardized:



Username toxicity types (dataset A)

122 144 toxic usernames found among 4 499 813

Toxic usernames produce 62% of toxicity

dataset A

244K users, 320K/2 265K toxic comments. Toxicity type flags might overlap

Username toxicity types (dataset B)

207 435 toxic usernames found among 7 915 410

Toxic usernames produce 63% of toxicity

dataset B

414K users, 913K/6 571K toxic comments. Toxicity type flags might overlap

Figure 3: Distributions of toxic username types and toxic content generated.

```
jointASample <- readRDS( file = "datasets/jointSample.rds")
toxic <- jointSample$toxicInt
commentsS <- standardize(jointSample$comments)
nonpureS <- standardize(jointSample$nonpure)
tox <- data.frame(toxic, commentsS, nonpureS)
```

For the model structure selection, let's walk through the method we used for the prediction of nonpure based on toxicInt. First, we selected plausible model structure candidates.

- The simplest model (toxicVSnonpure0) assumed the output is normally distributed around the overall mean m shifted by a toxicInt-specific coefficient, with the prior for σ being $\mathsf{Exp}(1)$, and the priors for the mean and the coefficients being $\mathsf{Norm}(0,.9)$ (we'll say a bit more about the choice a priors soon). In fact, the distribution of residuals is approximately normal.

$$\begin{aligned}\mu_i &= m + a[\mathsf{toxic}_i]\\ \sigma &\sim \mathsf{dexp}(1)\\ a[\mathsf{toxic}] &\sim \mathsf{dnorm}(0,.9)\\ m &\sim \mathsf{dnorm}(0,.9)\end{aligned}$$

- A somewhat more complex model (toxicVSnonpure1) assumed linear impact of user activity, with an interaction with the toxicity type (that is, the comments coefficients were toxicInt-group specific):

$$\begin{aligned}\mu_i &= m + a[\mathsf{toxic}_i] + b[\mathsf{toxic}_i] * \mathsf{commentsS}_i\\ \sigma &\sim \mathsf{dexp}(1)\\ a[\mathsf{toxic}] &\sim \mathsf{dnorm}(0,.9)\\ b[\mathsf{toxic}] &\sim \mathsf{dnorm}(0,.9)\\ m &\sim \mathsf{dnorm}(0,.9)\end{aligned}$$

(in general, unless we specify otherwise, the priors for coefficients and for σ will be the same throughout, so we will not list them further on).

- A slightly less complex model (toxicVSnonpure2) tried to dispose of group-specific a, leaving just group-specific commentsS coefficient b:

$$\mu_i = m + b[\mathsf{toxic}_i] * \mathsf{commentsS}_i$$

Since for data sets of this size Markov-Chain Monte Carlo methods were computationally unfeasible (especially given that a number of models were to be built), we used quadratic approximation of the posterior. Once the models were built, Widely Acceptable Information Criterion (WAIC) was used to select the most predictive Bayesian model architecture for these data sets (Table 1).

```
toxicVSnonpure0 <- quap(
  alist(
    nonpureS ~ dnorm(mu, sigma),
    mu <-  m + a[toxic],
    sigma ~ dexp(1),
    a[toxic] ~ dnorm(0,.9),
    m ~ dnorm(0,.9)
  ), data = tox
)

toxicVSnonpure1 <- quap(
  alist(
    nonpureS ~ dnorm(mu, sigma),
    mu <-  m + a[toxic] + b[toxic] * commentsS,
    sigma ~ dexp(1),
    a[toxic] ~ dnorm(0,.9),
    m ~ dnorm(0,.9),
    b[toxic] ~ dnorm(0,.9)
  ), data = tox
)


toxicVSnonpure2 <- quap(
  alist(
    nonpureS ~ dnorm(mu, sigma),
    mu <-  m  + b[toxic] * commentsS,
    sigma ~ dexp(1),
    m ~ dnorm(0,.9),
    b[toxic] ~ dnorm(0,.9)
  ), data = tox
)

comparisonA <- compare(toxicVSnonpure0,toxicVSnonpure1,toxicVSnonpure2)
#saveRDS(comparisonA, file = "datasets/comparisonA.rds")
```

Here's a more detailed explanation of the model comparison method we used, uninterested reader is invited to skip forward. Let y be the observations and Θ a posterior distribution. First, log-pointwise-predictive-density is defined by:

$$\mathsf{lppd}(y,\Theta) = \sum_i log \frac{1}{S} \sum_s p(y_i|\Theta_s)$$

where S is the number of samples in the posterior, and Θ_s is the s-th combination of sampled parameter values in the posterior distribution. That is, for each observation and each combination of parameters in the posterior we first compute its density, then we take the average density of that observation over all combinations of parameters in the posterior, and then take the logarithm. Finally, we sum these values up for all the observations. Crucially, when comparing posterior distributions with respect to the same dataset, lppds are proportional to unbiased estimates of their divergence from the real distribution (note that it is *only* proportional, and for this reason can be used for comparison of distributions only and makes no intuitive sense on its own). However, lppd always improves as the model gets more complex, so for model comparison it makes more sense to use the Widely Applicable Information Criterion (WAIC), which is an approximation of the out-of-sample deviance that converges to the cross-validation approximation in a large sample. It is defined as the log-posterior-predictive-density with an additional penalty proportional to the variance in the posterior predictions:

$$\mathsf{WAIC}(\mathsf{y},\Theta) = -2(\mathsf{lppd} - \overbrace{\sum_i var_\theta \mathsf{log} p(y_i|\theta)}^{\mathsf{penalty}})$$

Thus to construct the penalty, we calculate the variance in log-probabilities for each observation and sum them up. Because of the analogy to Akaike's criterion, the penalty is sometimes called the effective number of parameters, p_{WAIC}. How does WAIC compare to other information criteria? AIC uses MAP estimates instead of the posterior and requires that priors be flat or overwhelmed by the likelihood, and assumes that the posterior distribution is approximately multivariate Gaussian and the sample size is much greater than the number of parameters used in the model. Bayesian Information Criterion (BIC) also requires flat priors and uses MAP estimates. WAIC does not make these assumptions, and provides almost exactly the same results as AIC, when AIC's assumptions are met.

	WAIC	SE	dWAIC	dSE	pWAIC	weight
toxicVSnonpure1	510123.4	30995.76	0	NA	2119.7539	1
toxicVSnonpure2	511317	30989.46	1193.687	88.77891	2183.9273	0
toxicVSnonpure0	693921.3	21168.82	183797.896	11841.10427	906.4418	0

Table 1: Model selection results for the three models of toxic content vs. nickname toxicity and user activity.

Now we are ready to interpret the model comparison table. The first column compares the WAIC values (the smaller the better), the pWAIC column is the penalty term in the WAIC calculation. dWAIC is the difference between a model's WAIC and the best WAIC. The SE and dSE columns contain estimated standard errors for WAIC and dWAIC. Finally, the weight of a model i is:

$$w_i = \frac{exp(-.5\mathsf{dWAIC}_i)}{\sum_j exp(-.5\mathsf{dWAIC}_j)}$$

Weights are a traditional way to summarize relative support for each model and sum up to 1.

Now, back to the model building.

A similar model structure selection strategy has been used for the more complex models that evaluate the correlation between particular nick name toxicity type and particular toxic content type production, in such cases leading to models of the form:

$$\begin{aligned}
v_i &\sim \mathsf{dnorm}(\mu_i, \sigma)\\
\mu_i &= m + \mathsf{inapp}[\mathsf{nickInappropriate}_i] + \mathsf{off}[\mathsf{nickOffensive}_i] +\\
&\quad + \mathsf{prof}[\mathsf{nickProfanity}_i] + \mathsf{sex}[\mathsf{nickSexual}_i] + b[\mathsf{type}] \times \mathsf{comments}_i
\end{aligned}$$

Such a model estimates the dataset mean m, type-specific subgroup shifts (e.g. inapp, if a given username was deemed inappropriate), and user's activity multiplied by nickname-type specific coefficient b[type].

For models that we built, which incorporate interactions, it is hard to interpret estimated the obtained coefficients directly, and it is much better to inspect posterior predictions for a range of predictor settings, so this is what we do here. A general picture of toxic content production depending on whether the username was toxic can be inspected in Figure 5.

Let's turn to the choice of the priors now. First, the choice of distributions. The guiding principle is that of maximum entropy: we want to start with the least informative distributions consistent with what we know.[1]

Accordingly, since the exponential distribution has maximum entropy among all non-negative continuous distributions with the same average displacement, we use this distribution (with a fairly slow slope, setting $\lambda = 1$ for the prior for σ, and since the Gaussian has the largest entropy of any distribution with a finite variance, we use the Gaussian distribution for the prior of continuous variables that are not bound to be positive (and our count variables after standardization can be taken to be continuous variables of this sort). The Gaussian priors have mean set on 0, because after standardization this means they are mildly skeptical: the departure point for the most likely impact of various predictors is null. For the role played by the standard deviations, prior predictive check reveals that setting the value to 1 results in unrealistically wide priors (still a lot of weight is outside of $\pm 2.5sd$ range), whereas setting them to .9 results in a realistic, but not too narrow priors (Figure 4). These considerations apply to all coefficient priors in the linear models of toxic content production we used (note: not in the logistic model we used

[1] Why do we use entropy? A measure of uncertainty should (1) be continuous, (2) increase as the number of possible events increases, and (3) be additive. Only one function satisfies these desiderata (up to linear transformation): If there are n different possible events and each event i has probability p_i , and we call the list of probabilities p, then the measure is:

$$H(p) = -E\log(p_i) = -\sum_{i=1}^{n} p_o \log(p_i)$$

Now, to measure distances between measures given various assumptions, we measure divergence understood as the additional uncertainty induced by using probabilities from one distribution to describe another distribution, and its standard measure is the average difference in log probability between the target (p) and model (q), the Kullback-Leibler divergence:

$$D_{KL}(p,q) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right)$$

How is it arrived at? First, cross-entropy is defined as $H(p,q) = \sum_i p_i log(q_i)$, and then divergence turns out to be the additional entropy introduced by using q:

$$D_{KL}(p,q) = H(p,q) - H(p).$$

for the probability of suspension, which will be described later on), which are, accordingly, all set to $\mathsf{Norm}(0, .9)$.

```
toxicG <- 1:2
commentsG <- mean(tox$commentsS)
toxGrid <- expand.grid(toxic = toxicG, commentsS = commentsG)

winner <-   readRDS(file = "models/toxicVSnonpure.rds")

prior <- extract.prior( winner )
muPrior <- link(winner, post = prior, data = toxGrid)
muPriorMean <- apply( muPrior , 2, mean )
muPriorHPDI <- data.frame(t(apply( muPrior , 2 , HPDI )))
muPriorDF <- cbind(toxGrid, muPriorMean, muPriorHPDI)
colnames(muPrior) <- 1:2
muPriorLong <- melt(as.data.frame(muPrior))
colnames(muPriorLong) <- c("type", "nonpureS")

priorCheckPlotProper <- ggplot(muPriorLong, aes(x = type, y = nonpureS))+
  geom_violin()+theme_tufte()+
theme(plot.title.position = "plot",plot.title = element_text(face = "bold"))+
  ggtitle("Prior coefficient sd set to  .9")+
  scale_x_discrete(labels = c("no", "yes"))+
  ylab("toxic comments (standardized)")+xlab("toxic username")

toxicVSnonpure1wide <- quap(
  alist(
    nonpureS ~ dnorm(mu, sigma),
    mu <-  m + a[toxic] + b[toxic] * commentsS,
    sigma ~ dexp(1),
    a[toxic] ~ dnorm(0,1),
    m ~ dnorm(0,1),
    b[toxic] ~ dnorm(0,1)
  ), data = tox
)


priorWide <- extract.prior( toxicVSnonpure1wide )
muPriorWide <- link(toxicVSnonpure1wide, post = priorWide, data = toxGrid)
muPriorMeanWide <- apply( muPriorWide , 2, mean )
muPriorHPDIWide <- data.frame(t(apply( muPriorWide , 2 , HPDI )))
muPriorDFWide <- cbind(toxGrid, muPriorMeanWide, muPriorHPDIWide)
colnames(muPriorWide) <- 1:2
muPriorLongWide <- melt(as.data.frame(muPriorWide))
colnames(muPriorLongWide) <- c("type", "nonpureS")


priorCheckPlotWide <- ggplot(muPriorLongWide, aes(x = type, y = nonpureS))+
  geom_violin()+theme_tufte()+
  theme(plot.title.position = "plot",plot.title = element_text(face = "bold"))+
  ggtitle("Prior coefficient sd set to 1")+
  scale_x_discrete(labels = c("no", "yes"))+
  ylab("toxic comments (standardized)")+xlab("toxic username")

priorPredictiveCheck <- ggarrange(priorCheckPlotWide,priorCheckPlotProper )
```

Results for toxic content production. First, effect plots for models predicting toxic content in general using activity and username toxicity as predictors (Figure **??**).

```
source("scripts/toxicVSnonpureA1.R")
source("scripts/toxicVSnonpureB1.R")
source("scripts/toxicVSnonpureA2.R")
source("scripts/toxicVSnonpureB2.R")

plotAverageB <- ggplot(muPosteriorLong, aes(x = type, y = nonpureS))+
  theme_tufte(base_size = 10)+ylab("toxic comments")+
  ggtitle("\n")+xlab("toxicity")+
  scale_y_continuous(breaks = at, labels = labels)+
  labs(subtitle = "20 days, X comments per week, dataset B")+
  scale_x_discrete(labels = c("no", "yes"))+
```

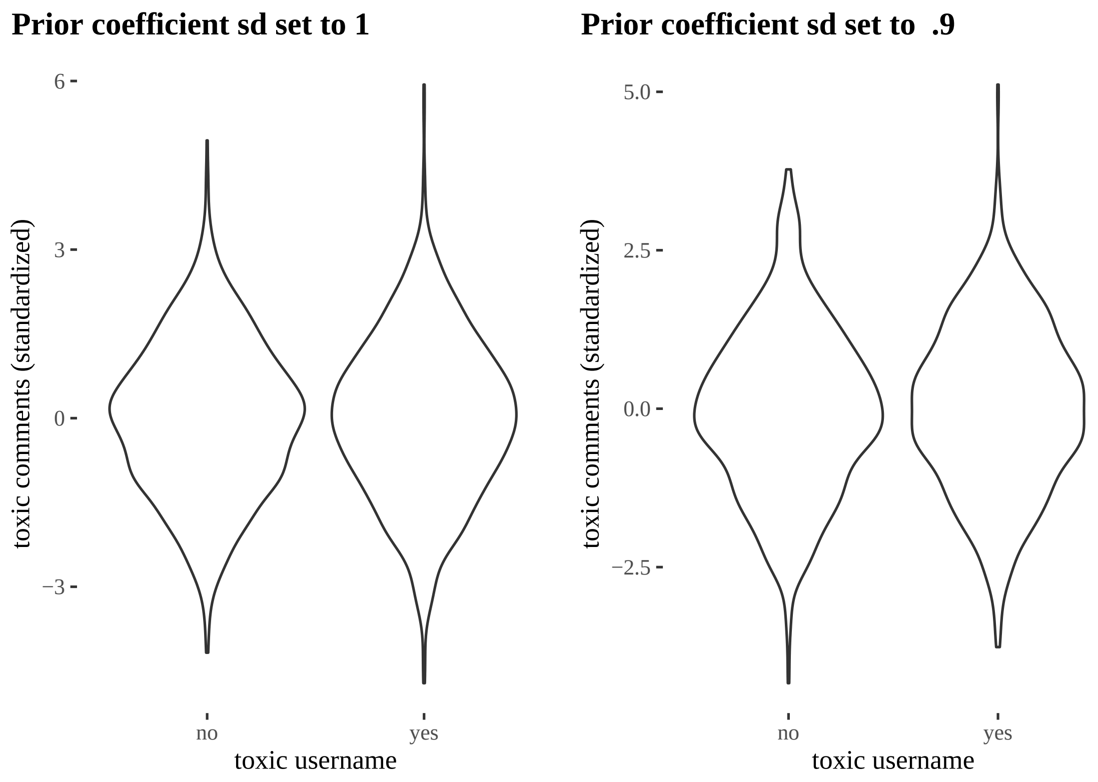


Figure 4: Setting standard deviation for the coefficient priors result in too wide prior predictions, setting them to .9 yields a more realistic but still fairly wide regularizing prior.

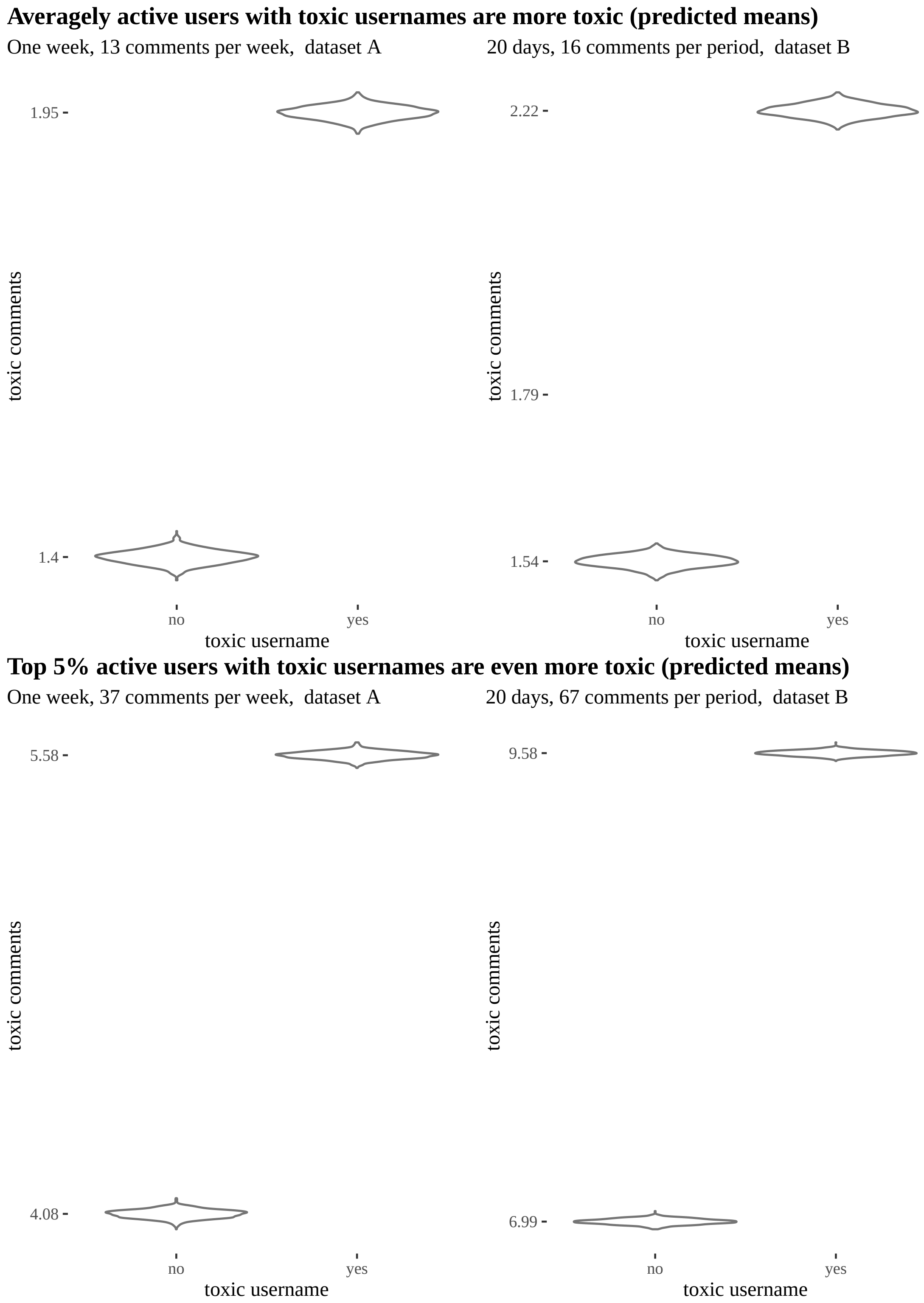
```
  theme(plot.title.position = "plot",plot.title = element_text(face = "bold"))


plotActiveB <- ggplot(muPosteriorLong, aes(x = type, y = nonpureS))+
  theme_tufte(base_size = 10)+ylab("toxic comments")+
  ggtitle("")+xlab("toxicity")+
  scale_y_continuous(breaks = at, labels = labels)+
  labs(subtitle = "20 days, X comments per week, dataset B",
       caption ="predicted means with 89% Highest Posterior Density Intervals")+
  scale_x_discrete(labels = c("no", "yes"))+
  theme(plot.title.position = "plot",plot.title = element_text(face = "bold"))
```

Next, we take a more fine-grained perspective and look at various types of toxic behavior depending on username toxicity (Figure 6), and at various types of toxic content versus activity level and toxic username type: profanities (Figure 7), personal attacks (Figure 8), sexual harassment (Figure 9), sexual remarks (Figure 10), bad wishes (Figure 11), rejections (Figure 12)

Bayesian models for suspensions. Another question pertained to the relation between toxic username type and the probability of suspension later on. For the latter we sued information about suspensions as of 21 Oct, 2021. To answer this question we built a model with the following specification:

$$\begin{aligned}
\mathsf{suspended} &\sim \mathsf{Binom}(1, p)\\
\mathsf{logit}(p) &= a + b[\mathsf{toxicInt}] + c[\mathsf{toxicInt}] * commentsS\\
a &\sim \mathsf{Norm}(0, .5)\\
b[\mathsf{toxicInt}] &\sim \mathsf{Norm}(0, .9)\\
c[\mathsf{toxicInt}] &\sim \mathsf{Norm}(0, .9)
\end{aligned}$$

This model WAIC-dominated the one obtained by deleting $c[\mathsf{toxicInt}] * commentsS$ and the null model obtained by taking $\mathsf{logit}(p) = a$. The justification for the priors is that they result in a slightly skeptical prior with the most likely impact of the toxicity types being null (Figure **??**).



Averagely active users with toxic usernames are more toxic (predicted means)

One week, 13 comments per week, dataset A

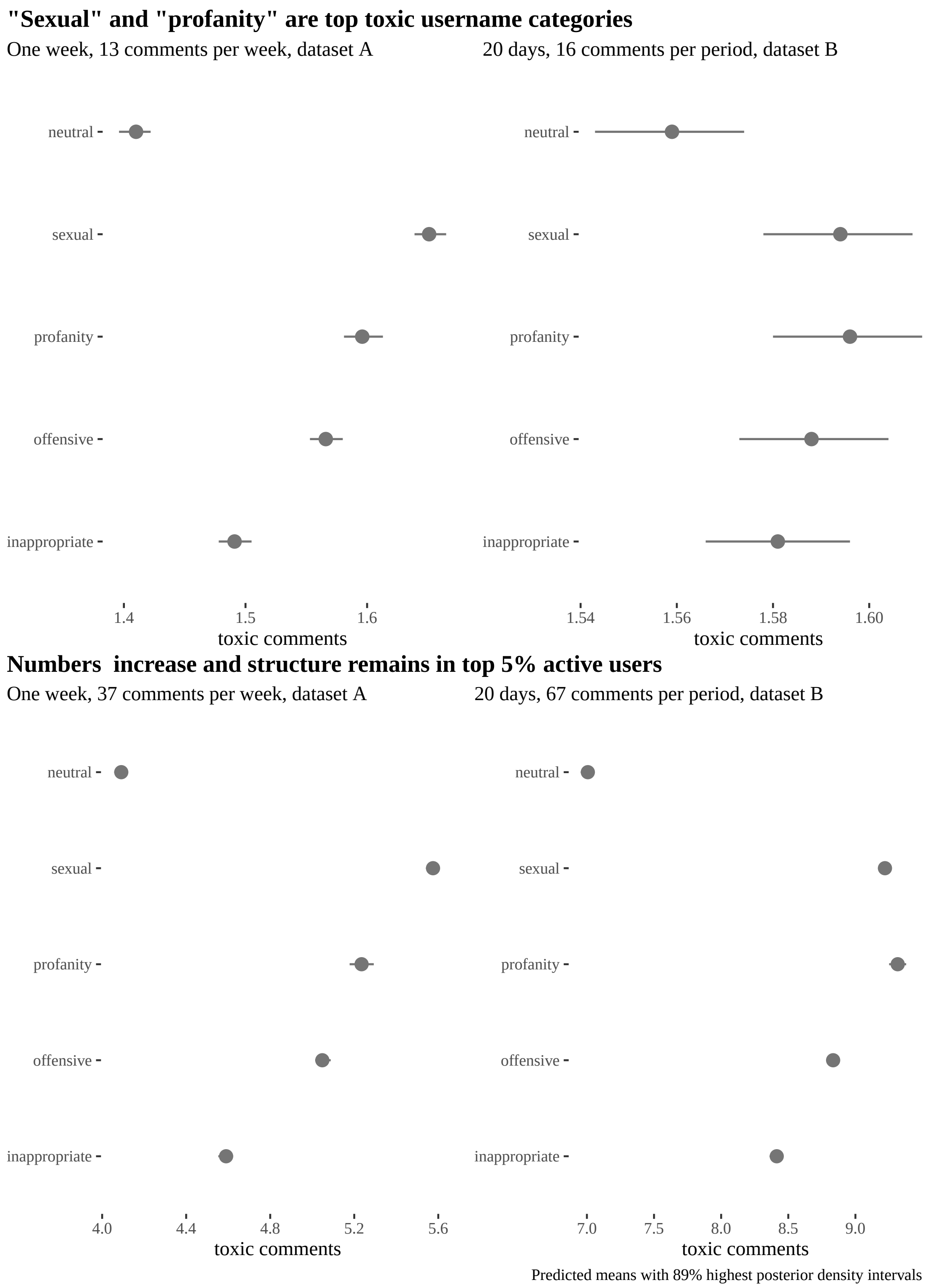
20 days, 16 comments per period, dataset B

Top 5% active users with toxic usernames are even more toxic (predicted means)

One week, 37 comments per week, dataset A

20 days, 67 comments per period, dataset B

Figure 5: Estimated means of toxic content production using the nickname toxicity as a predictor. Since activity is a co-variate, we plot the effects for modestly active users and for a top 5% active user.



"Sexual" and "profanity" are top toxic username categories

One week, 13 comments per week, dataset A

20 days, 16 comments per period, dataset B

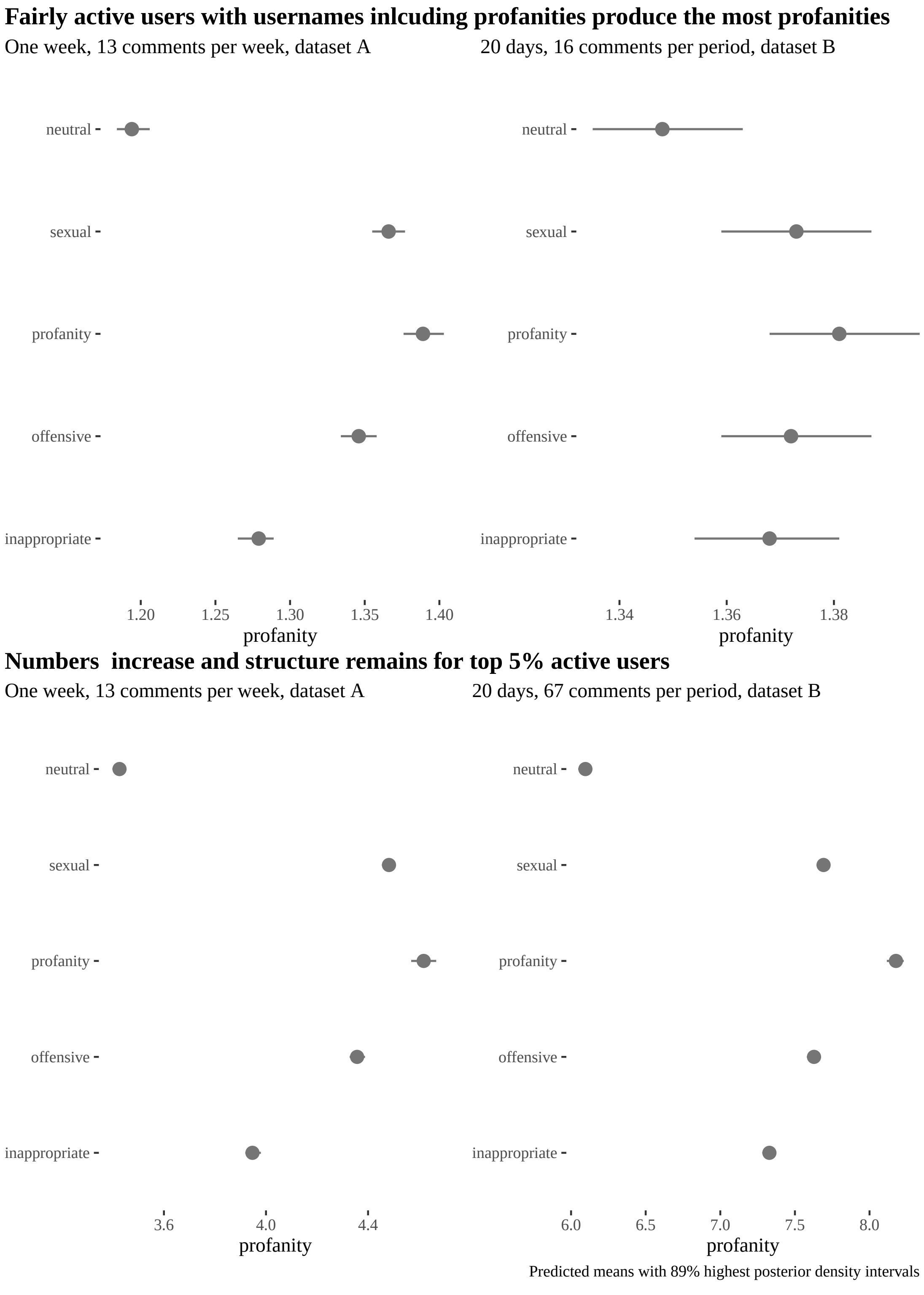
Numbers increase and structure remains in top 5% active users

One week, 37 comments per week, dataset A

20 days, 67 comments per period, dataset B

Predicted means with 89% highest posterior density intervals

Figure 6: Toxic content production by toxic username type.



Fairly active users with usernames inlcuding profanities produce the most profanities

One week, 13 comments per week, dataset A

20 days, 16 comments per period, dataset B

Numbers increase and structure remains for top 5% active users

One week, 13 comments per week, dataset A

20 days, 67 comments per period, dataset B

Predicted means with 89% highest posterior density intervals

Figure 7: Toxic username type vs. profanities produced.

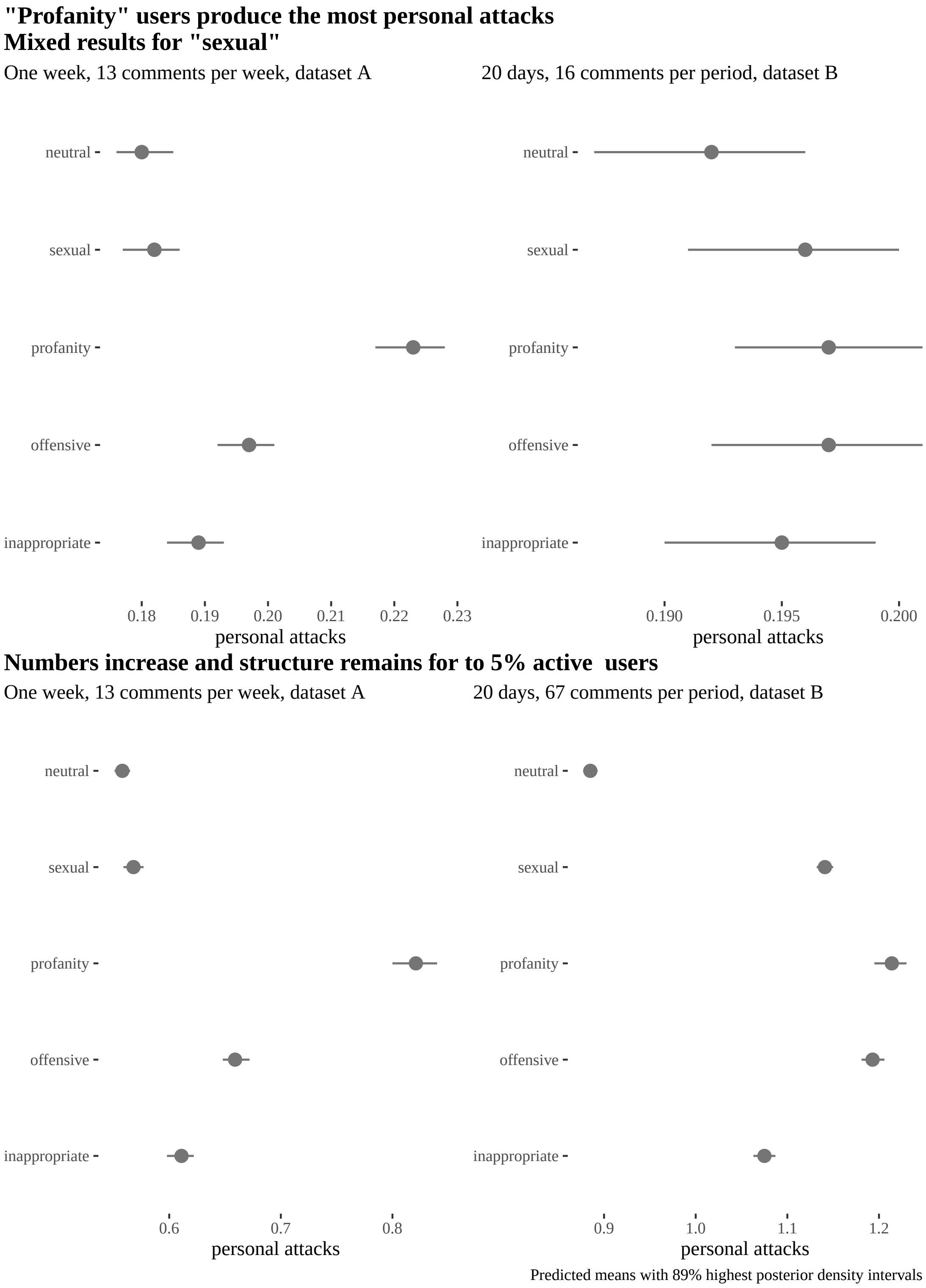


Figure 8: Toxic username type vs. personal attacks produced.

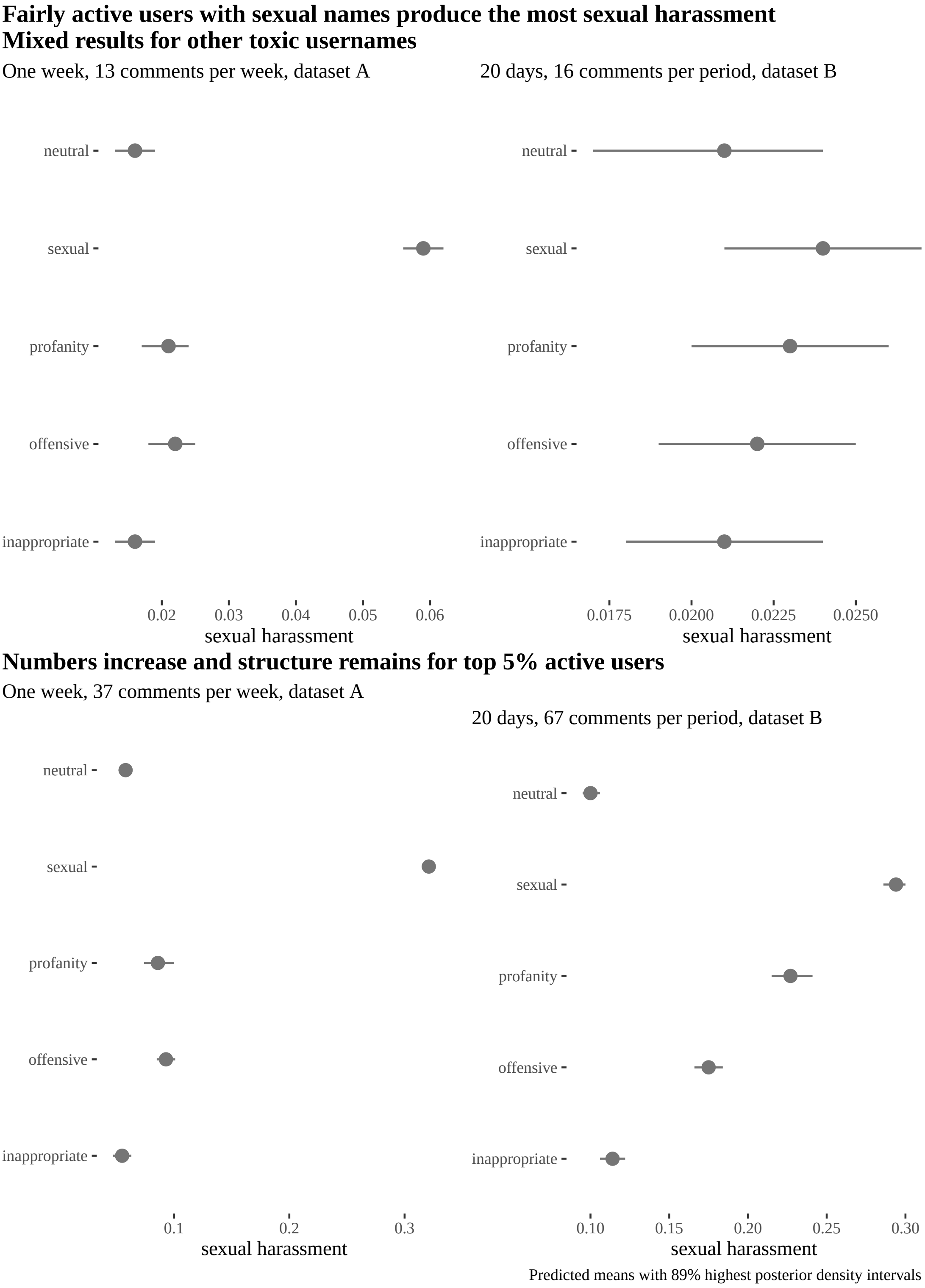


Figure 9: Toxic username type vs. sexual harrassment produced.

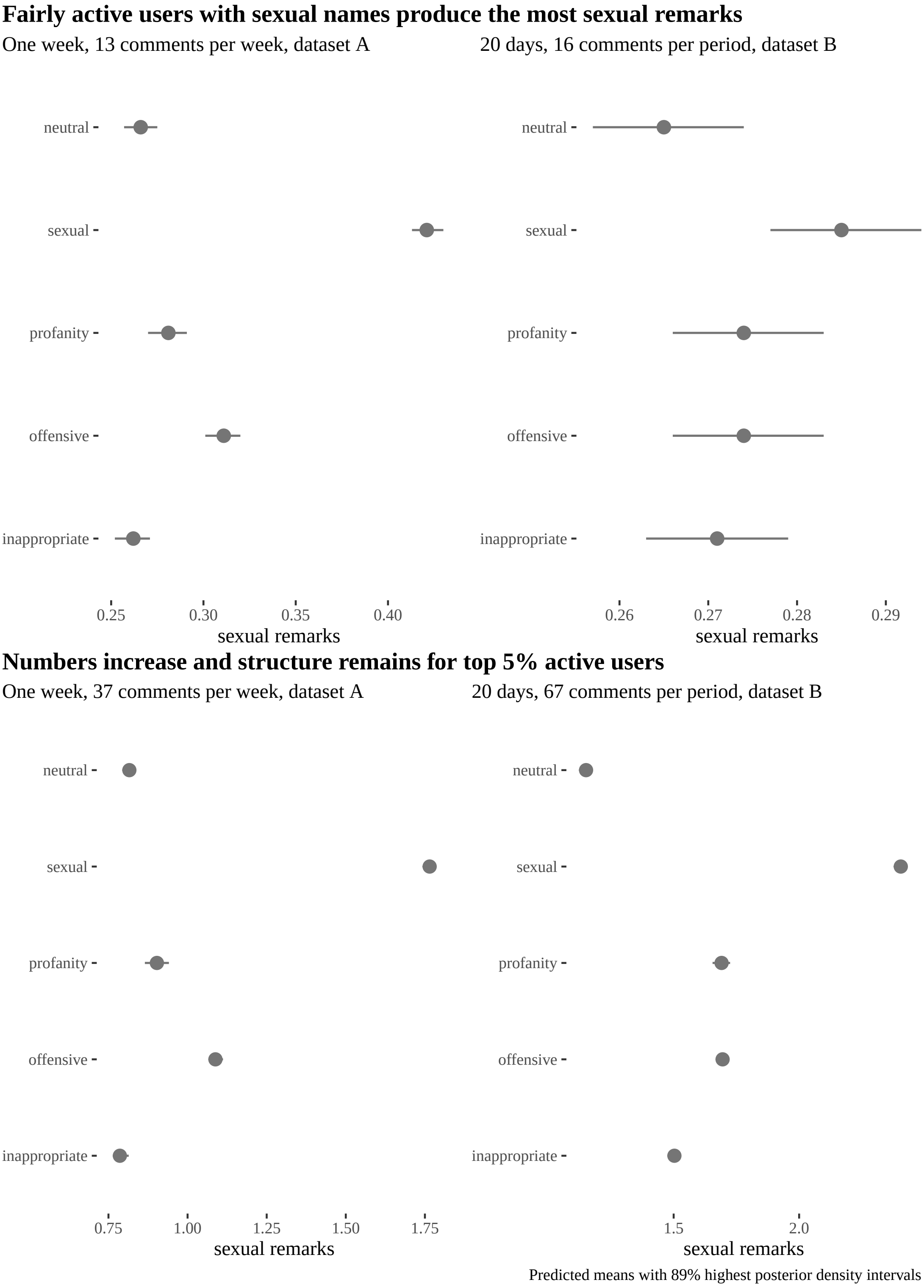


Figure 10: Toxic username type vs. sexual remarks produced.

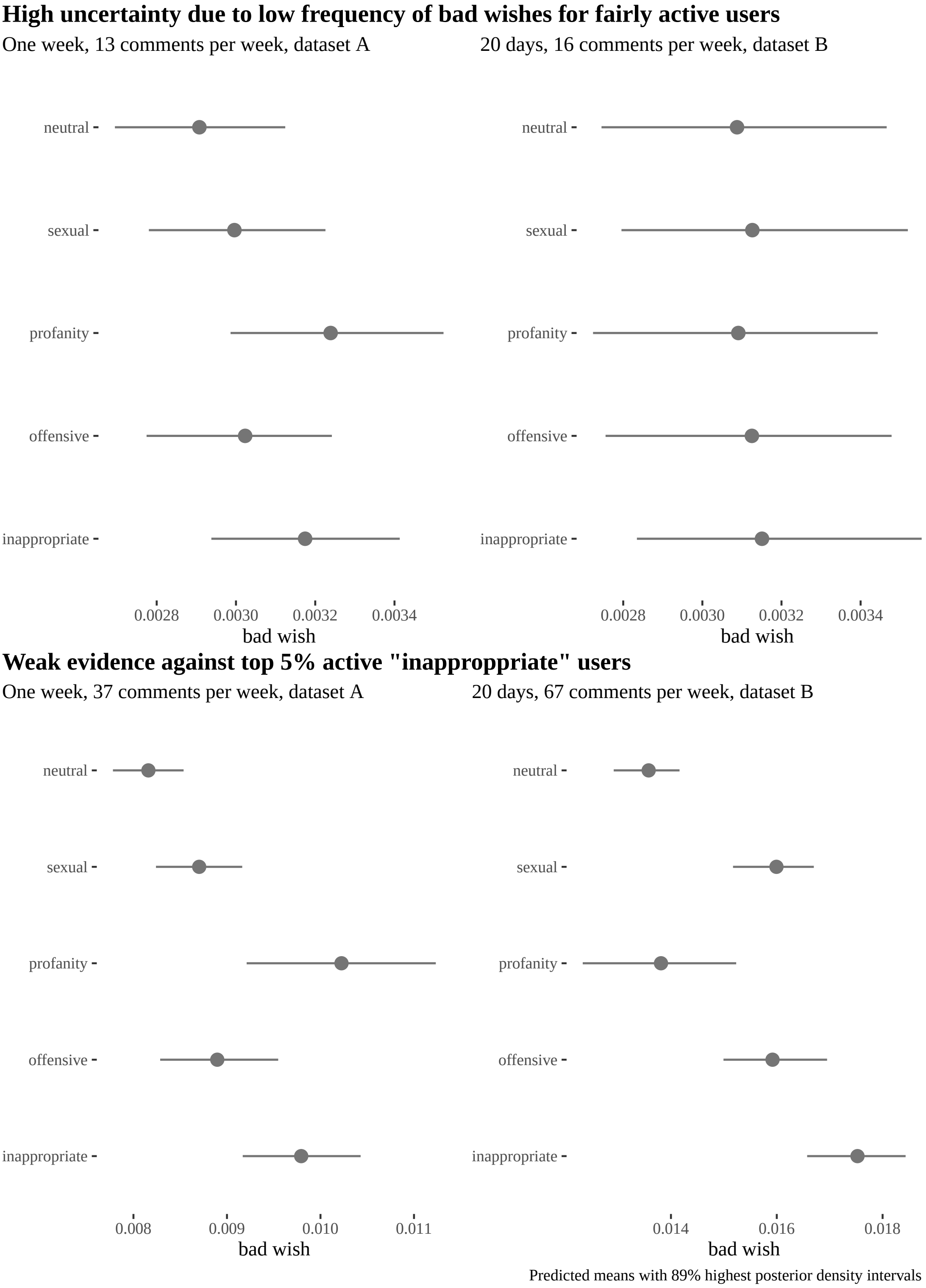


Figure 11: Toxic username type vs. bad wishes produced.

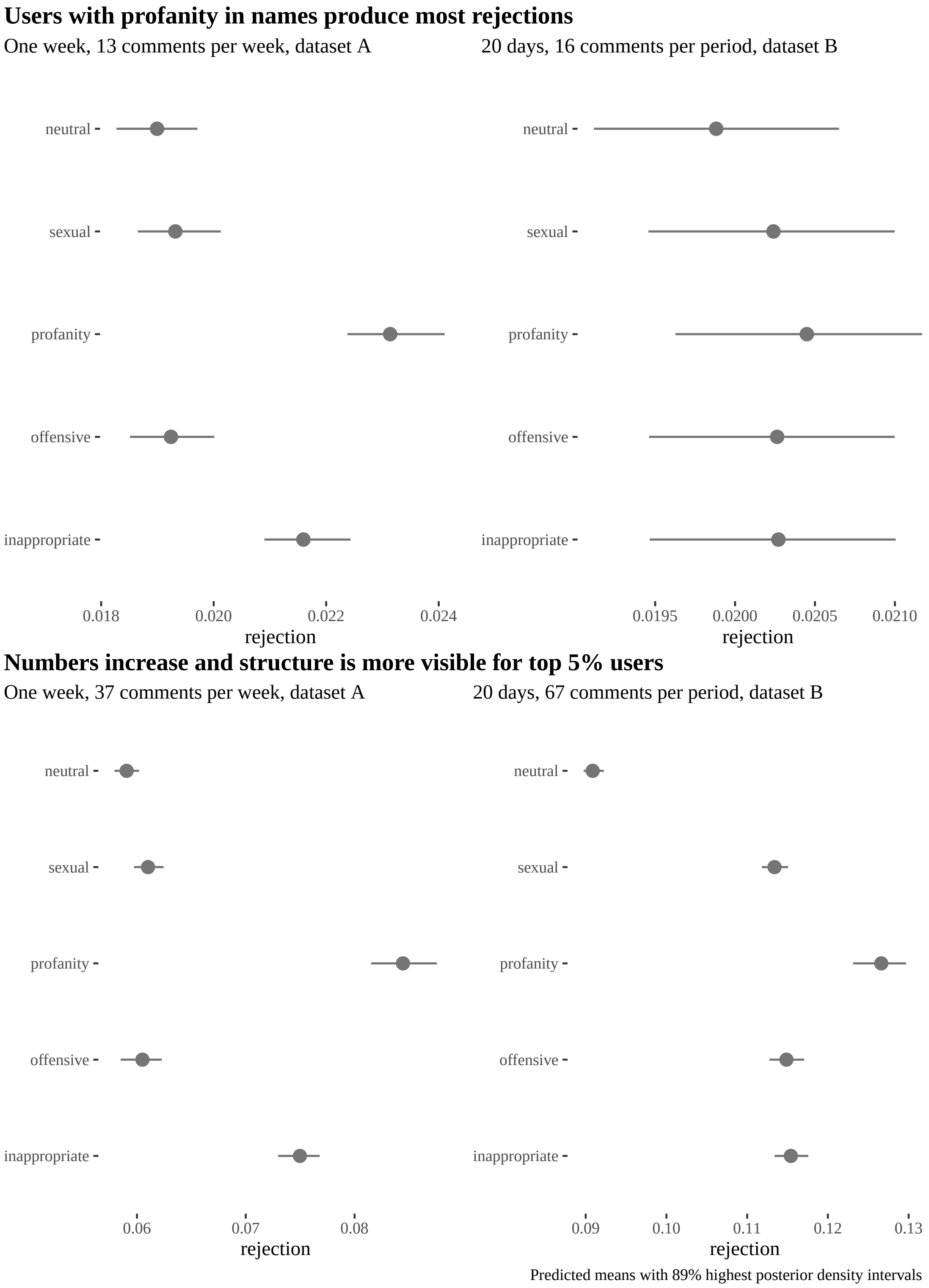


Figure 12: Toxic username type vs. rejections produced.

```
suspendedToxicComment <- readRDS(file = "models/suspendedToxicComment.rds")
suspensions <- readRDS(file = "datasets/suspensionsCut.rds")
suspensions <- suspensions[suspensions$status != "deleted",]
suspensions$suspended <- as.integer(suspensions$status == "suspended")
suspensions$commentsS <- standardize(suspensions$comments)
prior <- extract.prior( suspendedToxicComment , n=1e5 )
p <- sapply( 1:2 , function(k) inv_logit( prior$a + prior$b[,k] +
                        prior$b[,k] * mean(suspensions$commentsS)) )
suspensionsPrior <- dens( ( p[,1] - p[,2] ) , adj=0.1 )
```

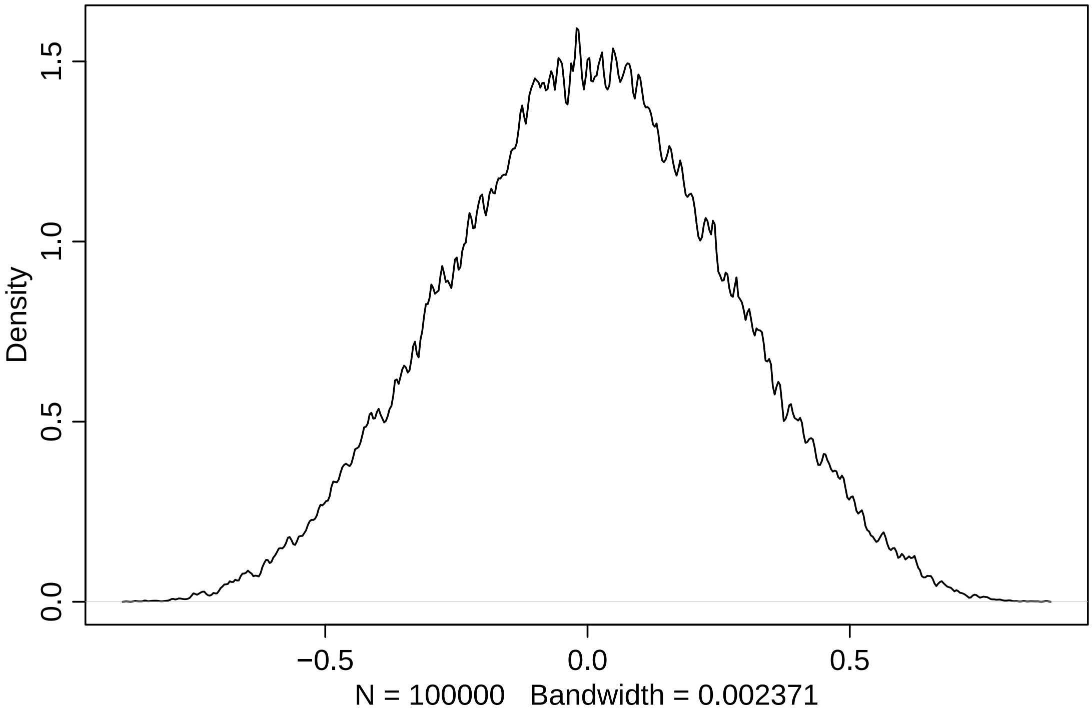


Figure 13: Prior predictive check (coefficients) for the logistic model.

In fact, a look at the empirical distribution suggests that having a toxic username correlates to some extent with the account being suspended later on (Figure 14)–notice the higher density of points in the upper row for the toxic user names as compared to the upper row for the non-toxic user names. The model predictions are in Figure 15.

```
jointSample <- readRDS( file = "datasets/jointSample.rds")
jointBSample <- readRDS( file = "datasets/jointBSample.rds")
type <- c(jointSample$type, jointBSample$type)
typeTox <- type[type != "[]"]
#round(sum(table(type)[c(1,2,3,4,5,6,7,9,10,11,13)])/length(type),3)
#round(sum(table(type)[c(1,2,3,4,5,6,7,9,10,11,13)])/length(typeTox),3)
```

Ensemble yearly predictions. Finally, we averaged the weekly predictions obtained by the two top models built on the two data sets to predict yearly toxic content production. This time we also looked at users whose usernames qualified into more than one category (such users constituted 2.4% of the two data sets, and 4.8% of users with toxic usernames.

Interestingly, the combination of inappropriate and sexual in the user name classification is an example of an interesting interaction of nickname type labels. All the ensemble predictions are illustrated in Figures 16, 17, and 18.

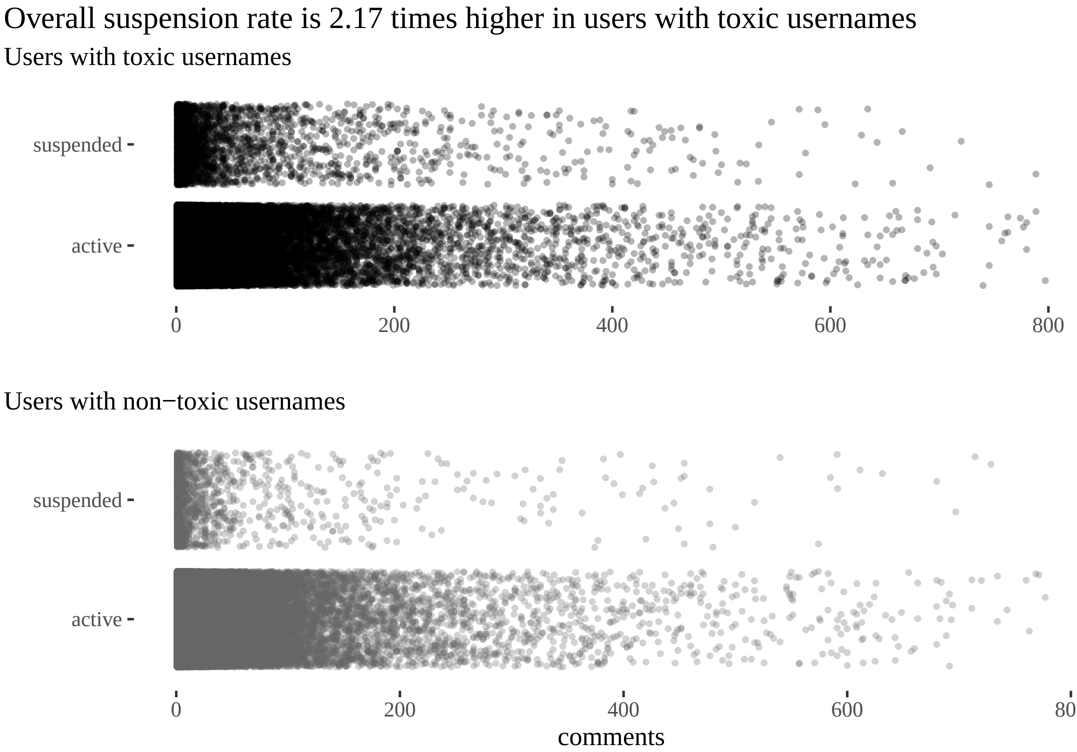


Figure 14: Empirical distribution of suspensions vs. username toxicity.

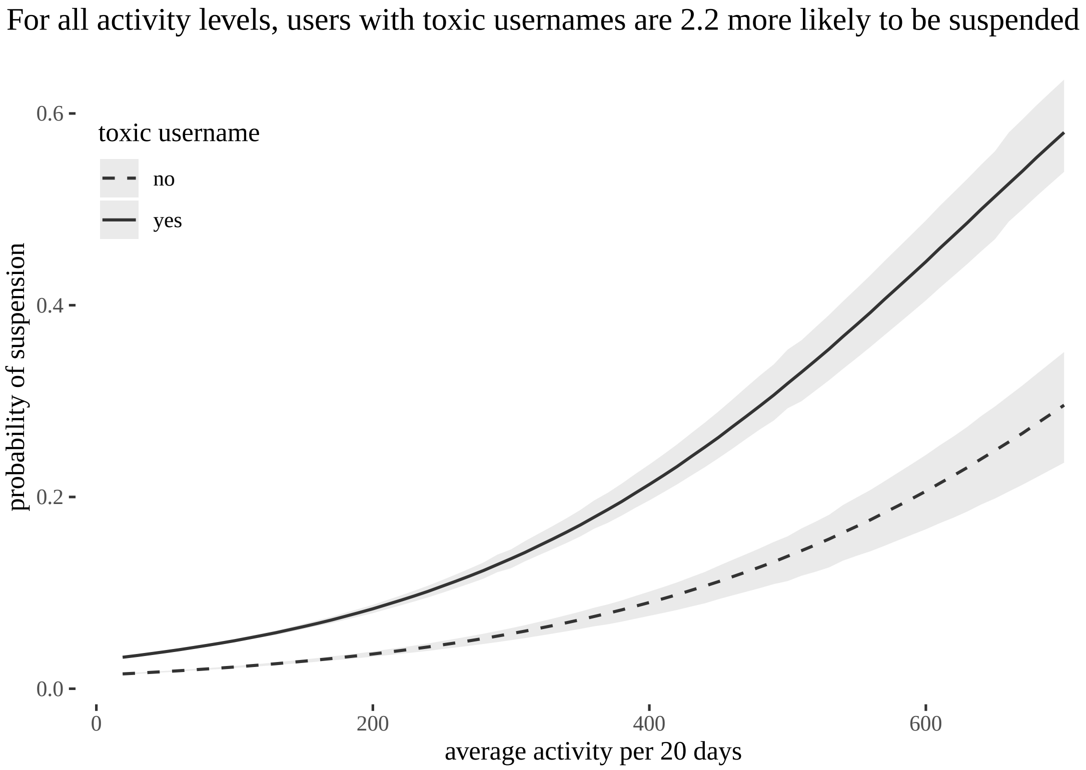


Figure 15: Logistic model predictions of suspensions depending on username toxicity.

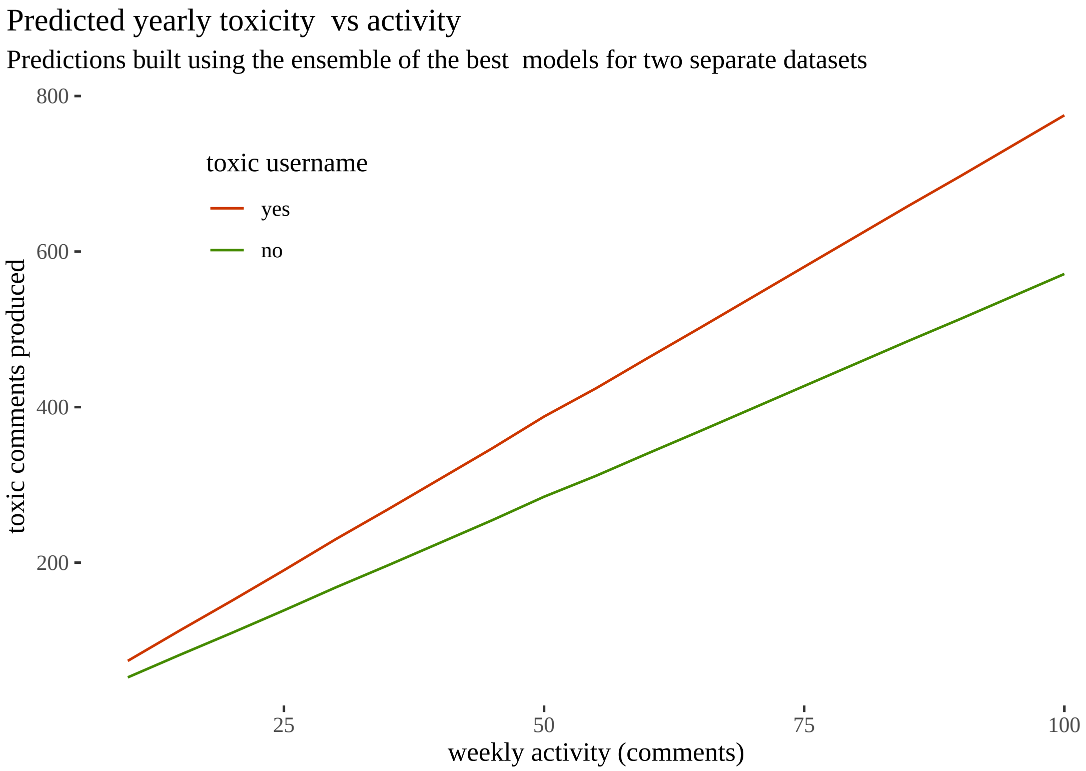


Figure 16: Yearly ensemble predictions of toxic content production vs. average weekly activity, split by toxic username, constructed using two WAIC-dominant bayesian models built on two separate datasets.

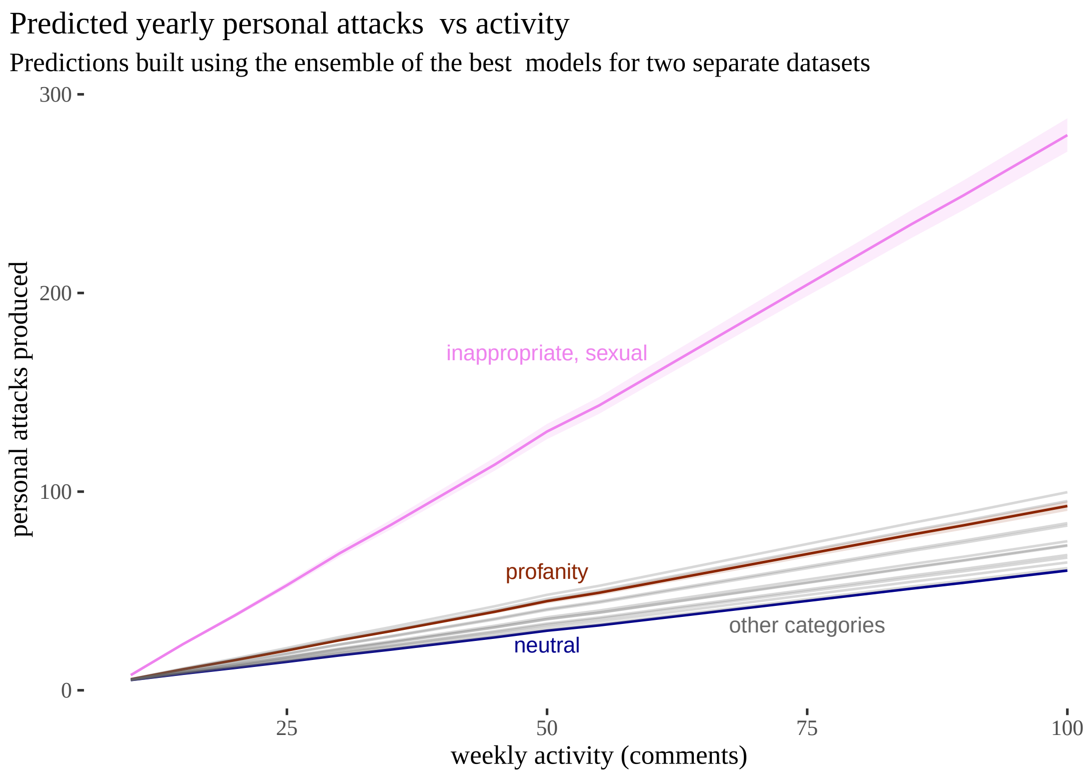


Figure 17: Yearly ensemble predictions of personal attacks vs. average weekly activity, split by multiple toxicity types, constructed using two WAIC-dominant bayesian models built on two separate datasets.

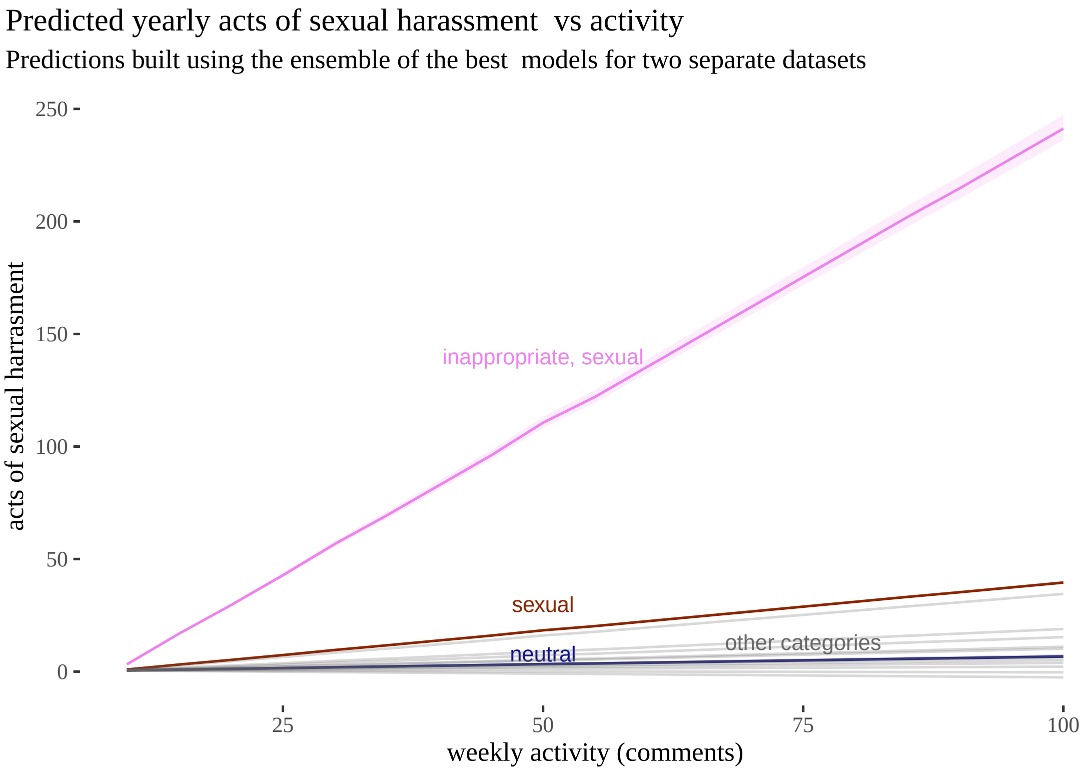


Figure 18: Yearly ensemble predictions of sexual harassment produced vs. average weekly activity, split by multiple toxicity types, constructed using two WAIC-dominant bayesian models built on two separate datasets.

References

McElreath, R. (2018). *Statistical rethinking: A bayesian course with examples in r and stan*. Chapman; Hall/CRC.