

Nicole's Violence Grant

Randi Garcia

June 20, 2016

This project uses data from ICPSR 36140 National Intimate Partner Sexual Violence Survey (NISVS).

From documentation (36140-descriptioncitation.pdf): 'Users of NISVS data will need to provide the "design" and "stratum" specifications to whichever software (such as SUDAAN or a SAS complex survey procedure) is being used for analysis and apply the correct weights when producing estimates. This is necessary because the NISVS data were collected through a complex sampling design. Information about the Design and Weight Variables can be found in the Orientation to Analyzing the NISVS Raw Data data documentation. Additional information can be found in the Technical Report.' More information on weighting can be found at 36140-Documentation-Weighting_REST.pdf.

Note that this documentation sheet indicates that the weird for the AIAN sample were incorrect in the original data file were incorrect, so the correct ones are in dataset 5.

Loading data into R. Note: If I add options(stringsAsFactors = FALSE) when loading the data it won't make EVERY variable a factor. Then I won't have the issue when subsetting below.

```
setwd(dir="C:/Users/Randi/Dropbox/Clark Consulting/Nicole/Violence Grant")

#add options(stringsAsFactors = FALSE) ??
resp <- read.csv(file="Da36140p3.csv", header=T)
perp <- read.csv(file="Da36140p4.csv", header=T)

dim(perp)
```

```
## [1] 51535 446
```

```
dim(resp)
```

```
## [1] 21378 493
```

I wanted my final bar graph to not have a bunch of unnecessary factors. I wanted only those instances with perps where the respondent said there was a need.

```
perp$FU6P1[1:20]
```

```
## [1] (0) No (0) No (0) No (0) No (0) No
## [6] (0) No (0) No (0) No (0) No (0) No
## [11] (0) No (0) No (1) Yes (0) No (-9) Unknown
## [16] (-9) Unknown (-9) Unknown (-9) Unknown (-9) Unknown (-9) Unknown
## Levels: (-9) Unknown (-99) N/A: INGFU=0 (0) No (1) Yes
```

```
int <- as.integer(perp$FU6P1)
wasmedneed <- as.numeric(int==4)
int2 <- as.integer(perp$FU6P2)
washousneed <- as.numeric(int2==4)
int3 <- as.integer(perp$FU6P3)
```

```

wascommneed <- as.numeric(int3==4)
int4 <- as.integer(perp$FU6P4)
wasadvoneed <- as.numeric(int4==4)
int5 <- as.integer(perp$FU6P5)
waslegalneed <- as.numeric(int5==4)

perp <- cbind(perp, wasmedneed, washousneed, wascommneed, wasadvoneed, waslegalneed)
datamedneed <- perp[perp$wasmedneed==1,]
datahousneed <- perp[perp$washousneed==1,]
datacommneed <- perp[perp$wascommneed==1,]
dataadvoneed <- perp[perp$wasadvoneed==1,]
datalegalneed <- perp[perp$waslegalneed==1,]

```

Bar charts for met versus unmet medical need.

```

datamedneed$FU6B1 <- droplevels(datamedneed$FU6B1)
summary(datamedneed$FU6B1)

```

```

## (-9) Unknown      (0) No      (1) Yes
##                9        176      1706

```

```

datahousneed$FU6B2 <- droplevels(datahousneed$FU6B2)
summary(datahousneed$FU6B2)

```

```

## (-9) Unknown      (0) No      (1) Yes
##                4        119      392

```

```

datacommneed$FU6B3 <- droplevels(datacommneed$FU6B3)
summary(datacommneed$FU6B3)

```

```

## (-9) Unknown      (0) No      (1) Yes
##                4        103      445

```

```

dataadvoneed$FU6B4 <- droplevels(dataadvoneed$FU6B4)
summary(dataadvoneed$FU6B4)

```

```

## (-9) Unknown      (0) No      (1) Yes
##                1        185      488

```

```

datalegalneed$FU6B5 <- droplevels(datalegalneed$FU6B5)
summary(datalegalneed$FU6B5)

```

```

## (-9) Unknown      (0) No      (1) Yes
##                6        237      1931

```

```

medneed <- table(datamedneed$FU6B1)
housneed <- table(datahousneed$FU6B2)
commneed <- table(datacommneed$FU6B3)
advoneed <- table(dataadvoneed$FU6B4)

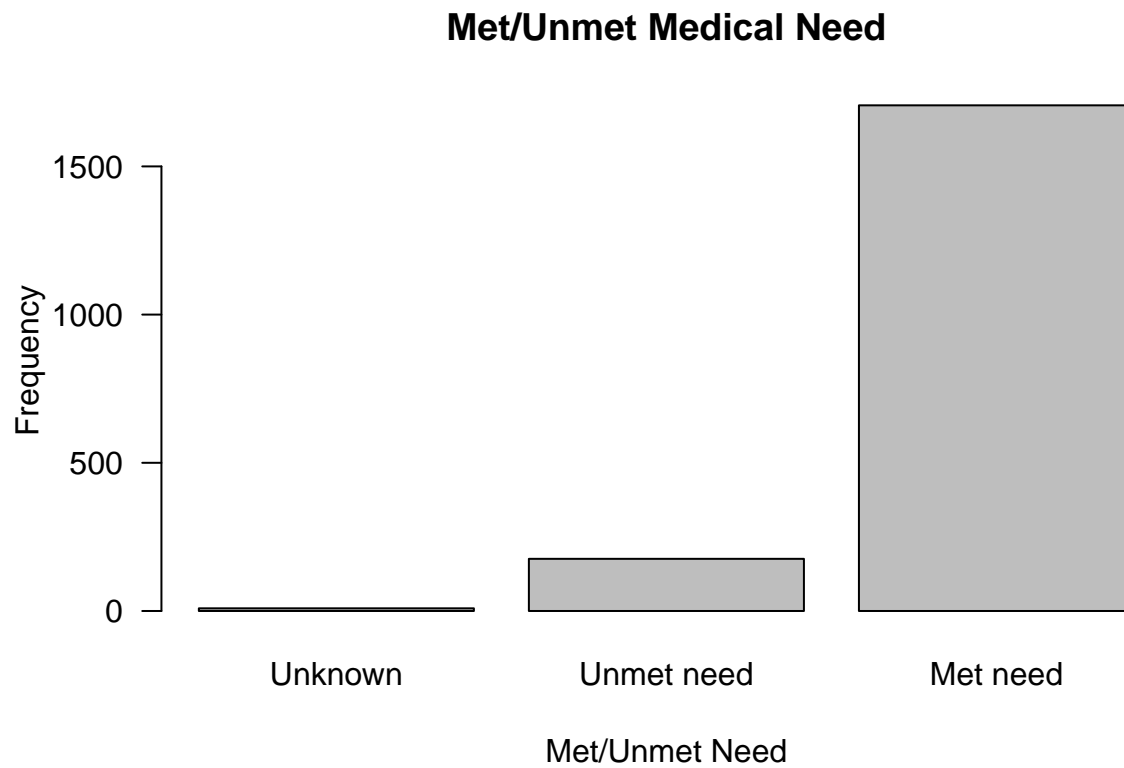
```

```
legalneed <- table(datalegalneed$FU6B5)
```

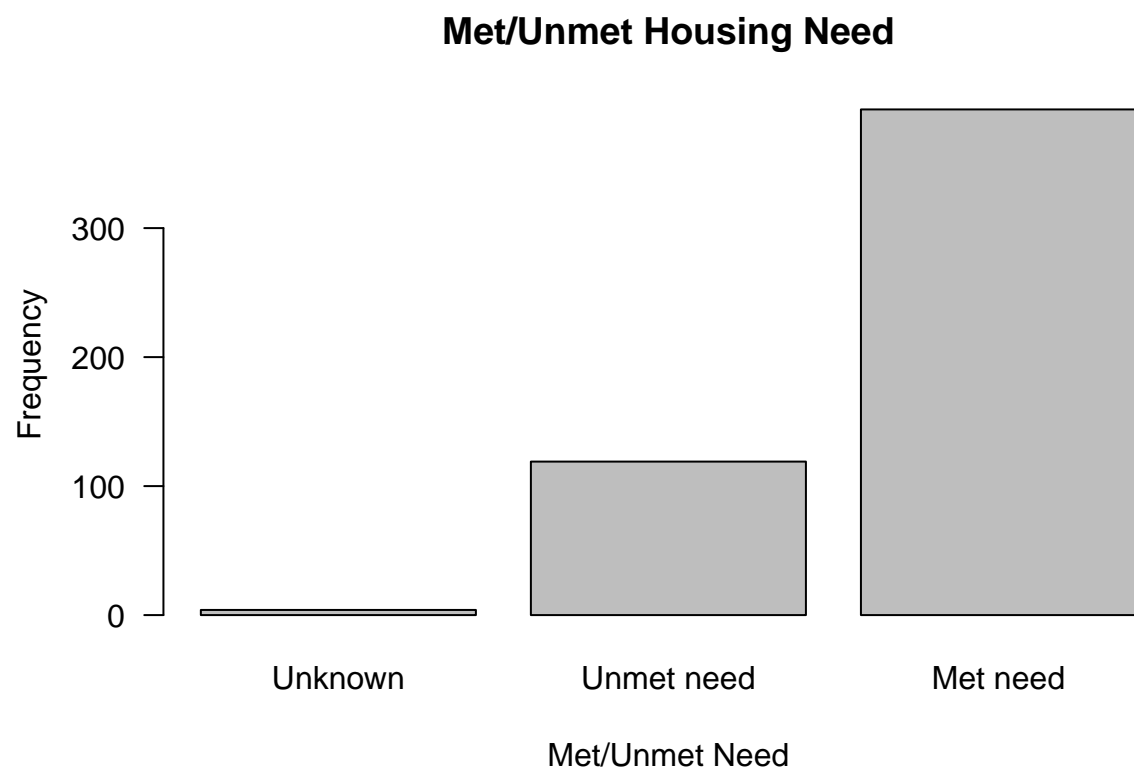
```
#summary(datamedneed$FU6P1)
```

```
#medneedperc <- table(datamedneed$FU6B1)/1891 #how to make this not hard coded in?
```

```
barplot(medneed, main="Met/Unmet Medical Need", xlab="Met/Unmet Need", ylab="Frequency", las=1, names.a
```

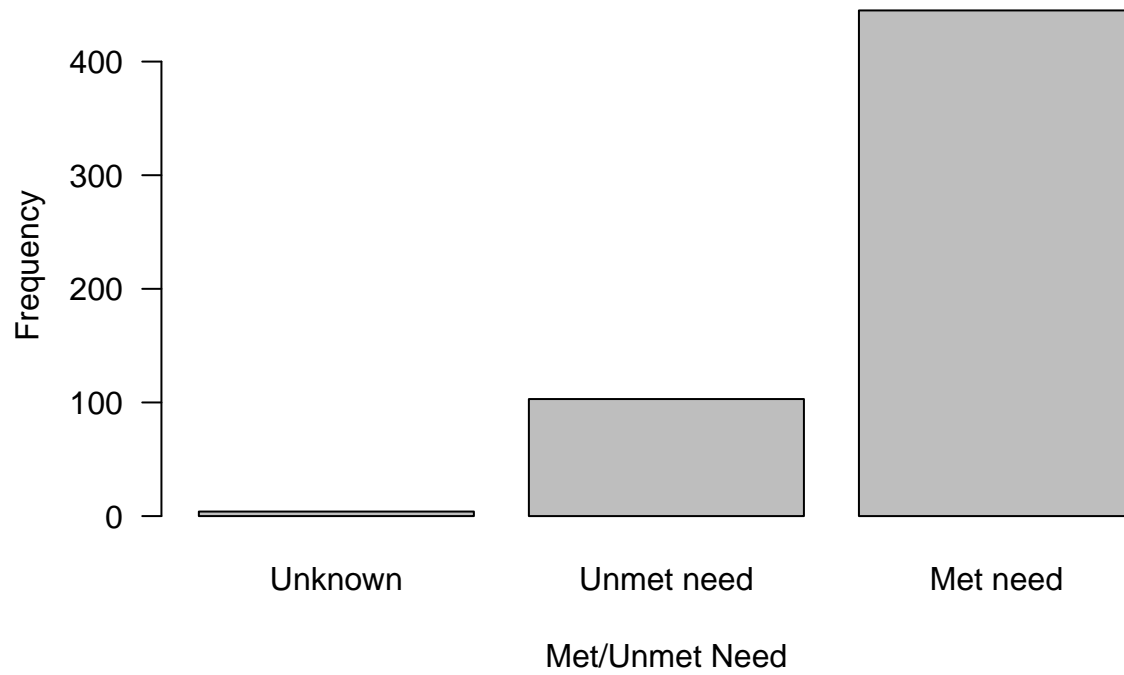


```
barplot(housneed, main="Met/Unmet Housing Need", xlab="Met/Unmet Need", ylab="Frequency", las=1, names.a
```

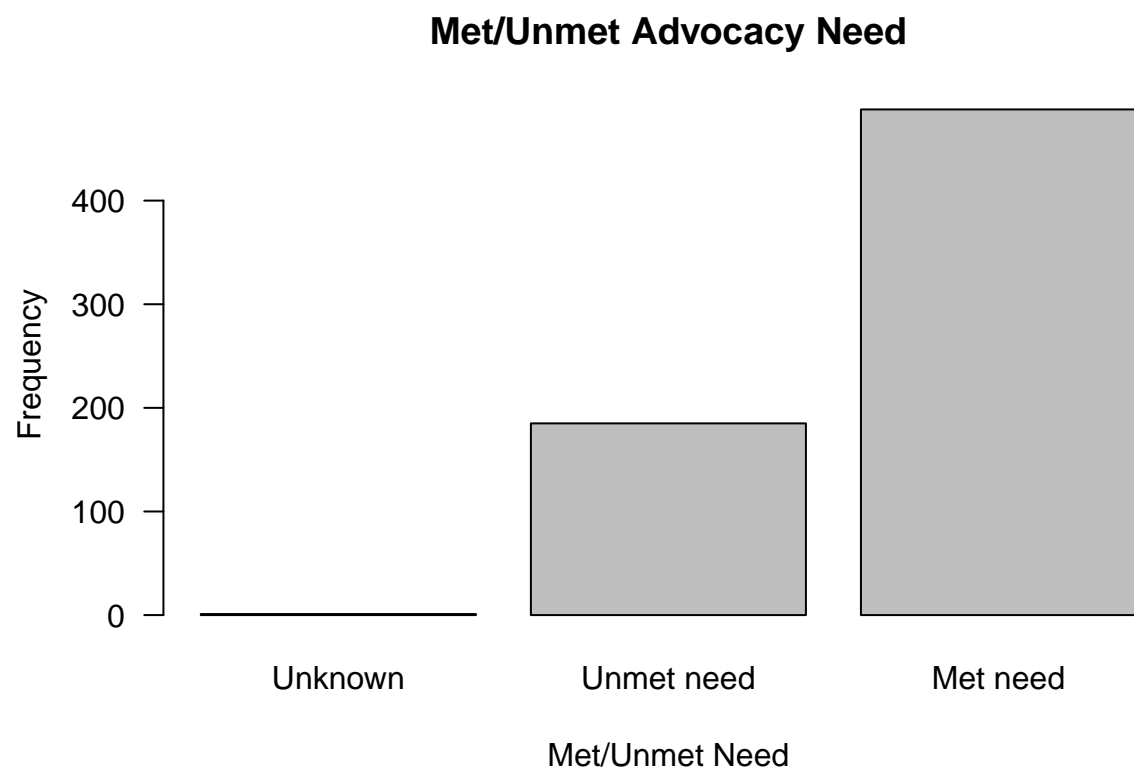


```
barplot(commneed, main="Met/Unmet Community Service Need", xlab="Met/Unmet Need", ylab="Frequency", las=1)
```

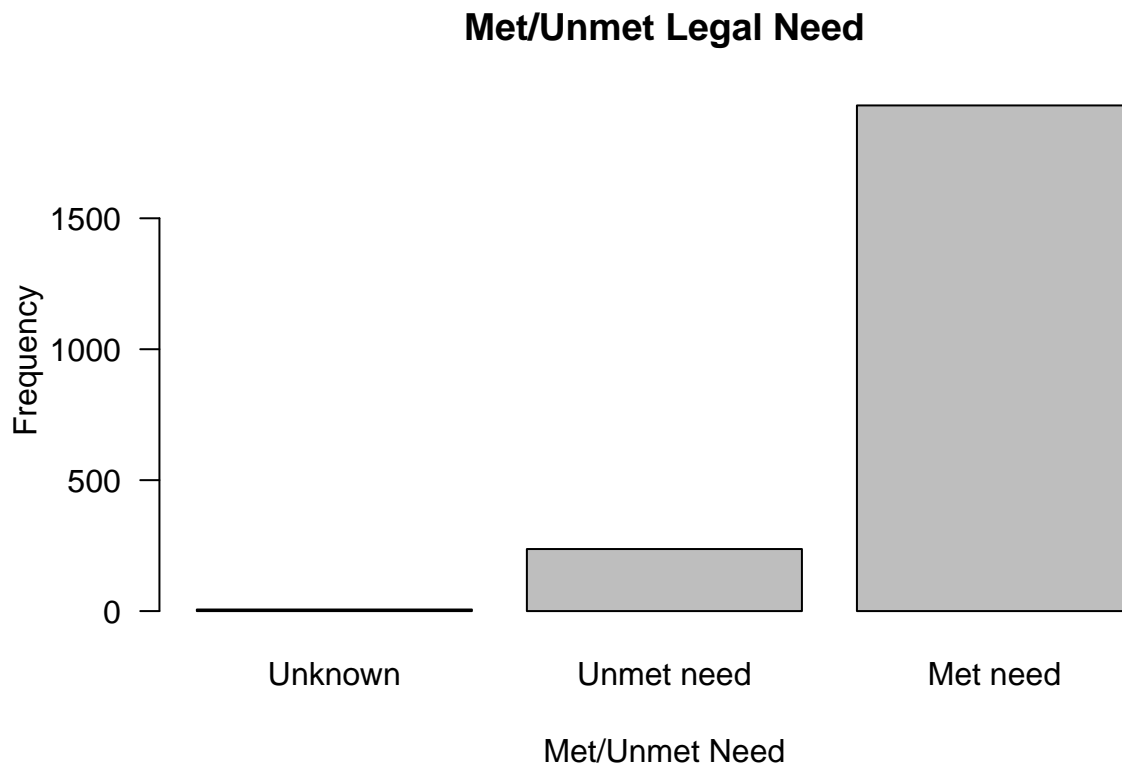
Met/Unmet Community Service Need



```
barplot(advoneed, main="Met/Unmet Advocacy Need", xlab="Met/Unmet Need", ylab="Frequency", las=1, names
```



```
barplot(legalneed, main="Met/Unmet Legal Need", xlab="Met/Unmet Need", ylab="Frequency", las=1, names.a
```



Exploring state data. Recoding so that state data is not separated by cell versus landline.

```
#missings
state <- rep("(-99) N/A: Sample=2", length(resp$GENSTRAT))
stateName <- rep("(-99) N/A: Sample=2", length(resp$GENSTRAT))
resp <- cbind(resp, state, stateName)

#Problem here: Creates the state abbreviations, but "N/" remains from the missing
resp$state <- substring(resp$GENSTRAT, 7, 8)
```

Data wrangling: making small datasets and attaching the state variable.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.2.5
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```

perpsmall <- select(perp, ZRID, FU6P1, FU6B1, FU6P2, FU6B2, FU6P3, FU6B3, FU6P4, FU6B4, FU6P5, FU6B5)
respsmall <- select(resp, ZRID, state)

perpnew <- left_join(perpsmall, respsmall, by = "ZRID")

propunmetmedneed <- perpnew %>%
  filter(FU6P1=="(1) Yes" & state != "N/") %>%
  group_by(state) %>%
  summarise(total = n(), unmet = sum(FU6B1 == "(0) No")) %>%
  mutate(prop = unmet/total)

propunmethousneed <- perpnew %>%
  filter(FU6P2=="(1) Yes" & state != "N/") %>%
  group_by(state) %>%
  summarise(total = n(), unmet = sum(FU6B2 == "(0) No")) %>%
  mutate(prop = unmet/total)

propunmetcommneed <- perpnew %>%
  filter(FU6P3=="(1) Yes" & state != "N/") %>%
  group_by(state) %>%
  summarise(total = n(), unmet = sum(FU6B3 == "(0) No")) %>%
  mutate(prop = unmet/total)

propunmetadvoneed <- perpnew %>%
  filter(FU6P4=="(1) Yes" & state != "N/") %>%
  group_by(state) %>%
  summarise(total = n(), unmet = sum(FU6B4 == "(0) No")) %>%
  mutate(prop = unmet/total)

propunmetlegalneed <- perpnew %>%
  filter(FU6P5=="(1) Yes" & state != "N/") %>%
  group_by(state) %>%
  summarise(total = n(), unmet = sum(FU6B5 == "(0) No")) %>%
  mutate(prop = unmet/total)

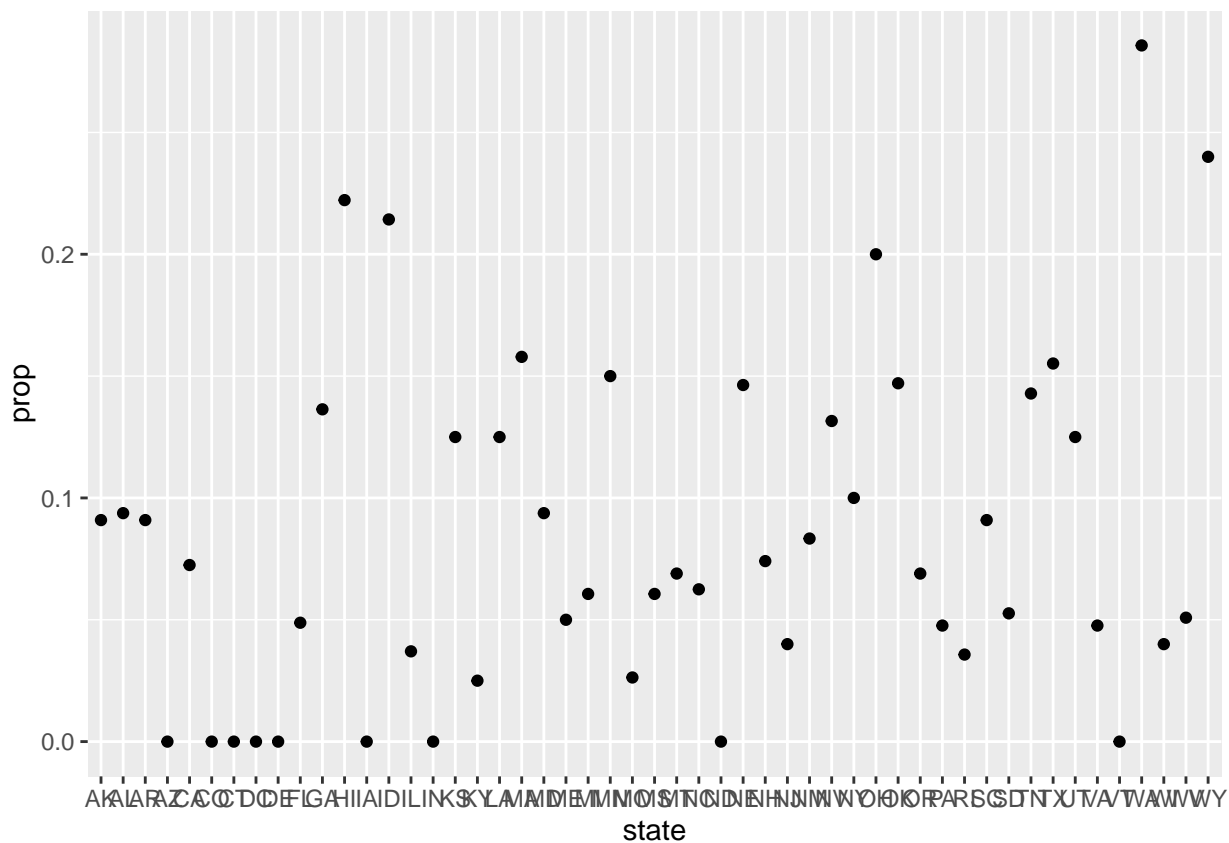
```

US maps of proportion (of the Yes-had the need) unmet need. Issues to resolve: Proportion unmet need is based on very small counts per state (e.g., 0, 3), also these numbers are based on perpetrators NOT respondent thus it is entirely possible (and maybe likely) that ALL of the unmet need in a state is from a single person.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```
ggplot(data = propunmetmedneed, aes(x=state, y=prop))+
  geom_point()
```

```
library(mosaic)
```

```
## Warning: package 'mosaic' was built under R version 3.2.5
```

```
## Loading required package: lattice
```

```
## Loading required package: mosaicData
```

```
## Warning: package 'mosaicData' was built under R version 3.2.5
```

```
## Loading required package: Matrix
```

```
##
```

```
## The 'mosaic' package masks several functions from core packages in order to add additional features.
## The original behavior of these functions should not be affected by this.
```

```
##
```

```
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
```

```
##
```

```
## mean
```

```
## The following objects are masked from 'package:dplyr':
##
##   count, do, tally

## The following objects are masked from 'package:stats':
##
##   binom.test, cor, cov, D, fivenum, IQR, median, prop.test,
##   quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##   max, mean, min, prod, range, sample, sum
```

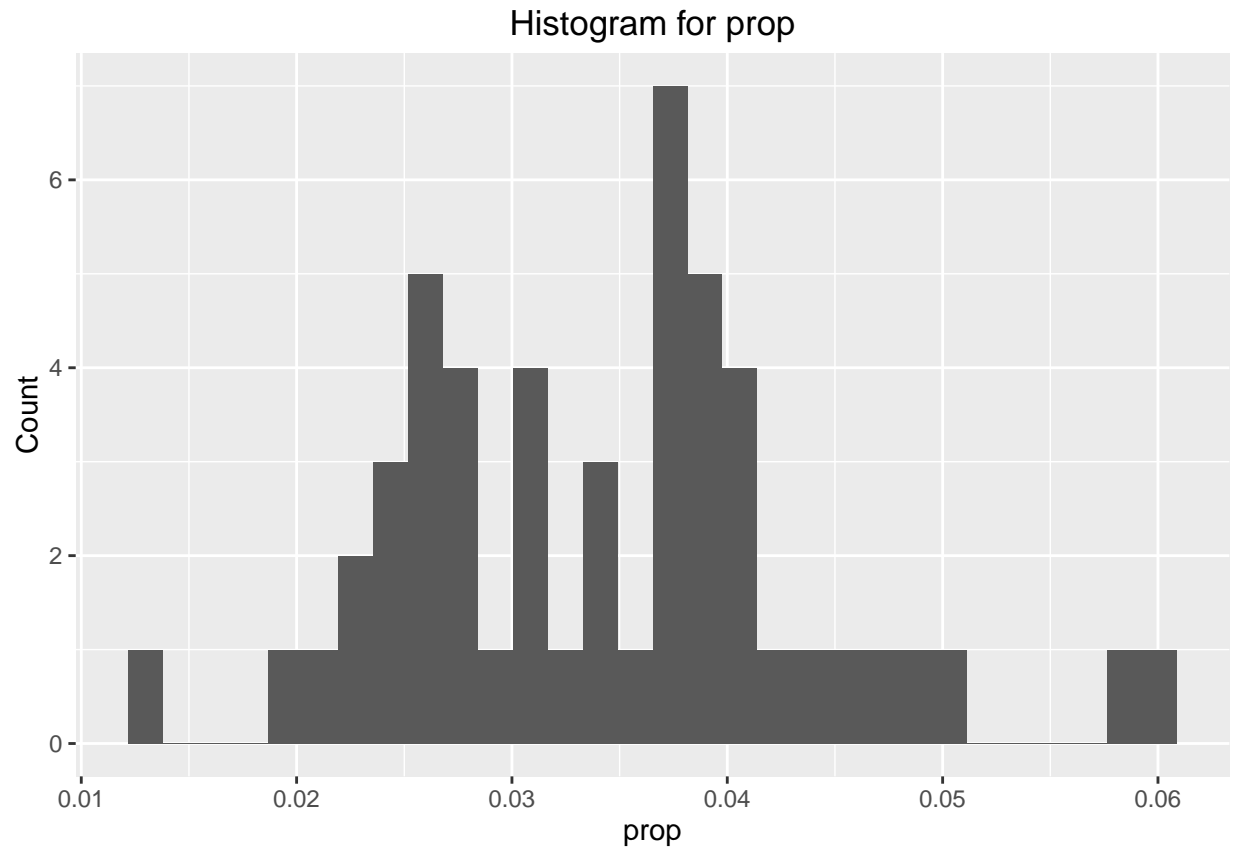
```
# mUSMap(propunmetmedneed, key="state", fill = "prop")+
#   scale_fill_continuous(low="white", high="red")+
#   ggtitle("Proportion Unmet Medical Need")
#
# mUSMap(propunmethousneed, key="state", fill = "prop")+
#   scale_fill_continuous(low="white", high="green")+
#   ggtitle("Proportion Unmet Housing Need")
#
# mUSMap(propunmetcommneed, key="state", fill = "prop")+
#   scale_fill_continuous(low="white", high="yellow")+
#   ggtitle("Proportion Unmet Community Service Need")
#
# mUSMap(propunmetadvoneed, key="state", fill = "prop")+
#   scale_fill_continuous(low="white", high="pink")+
#   ggtitle("Proportion Unmet Advocacy Service Need")
#
# mUSMap(propunmetlegalneed, key="state", fill = "prop")+
#   scale_fill_continuous(low="white", high="orange")+
#   ggtitle("Proportion Unmet Legal Need")
```

Medical Service Priority

```
propmedneed <- perpnew %>%
  filter(state != "N/") %>%
  group_by(state) %>%
  summarise(total = n(), need = sum(FU6P1 == "(1) Yes")) %>%
  mutate(prop = need/total)

ggplot(data=propmedneed, aes(prop)) +
  geom_histogram() +
  labs(title="Histogram for prop") +
  labs(x="prop", y="Count")
```

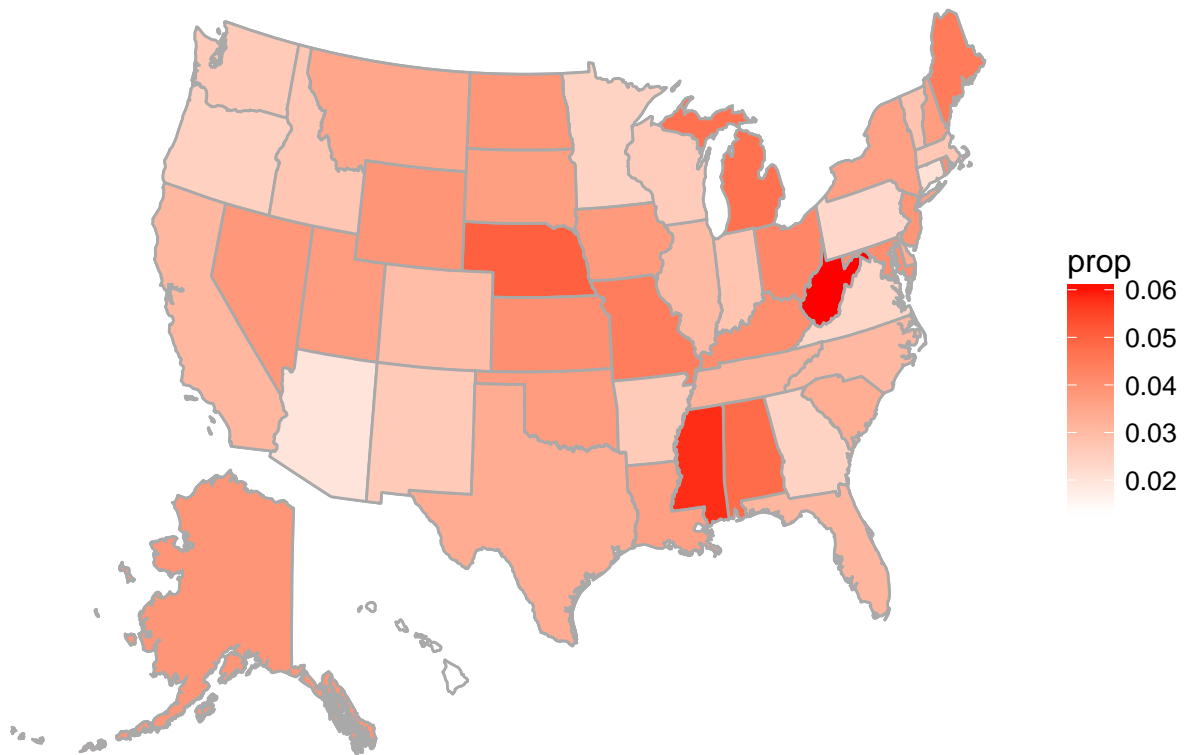
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mUSMap(propmedneed, key="state", fill = "prop")+  
  scale_fill_continuous(low="white", high="red")+  
  ggtitle("Proportion of State Medical Need")
```

Mapping API still under development and may change in future releases.

Proportion of State Medical Need

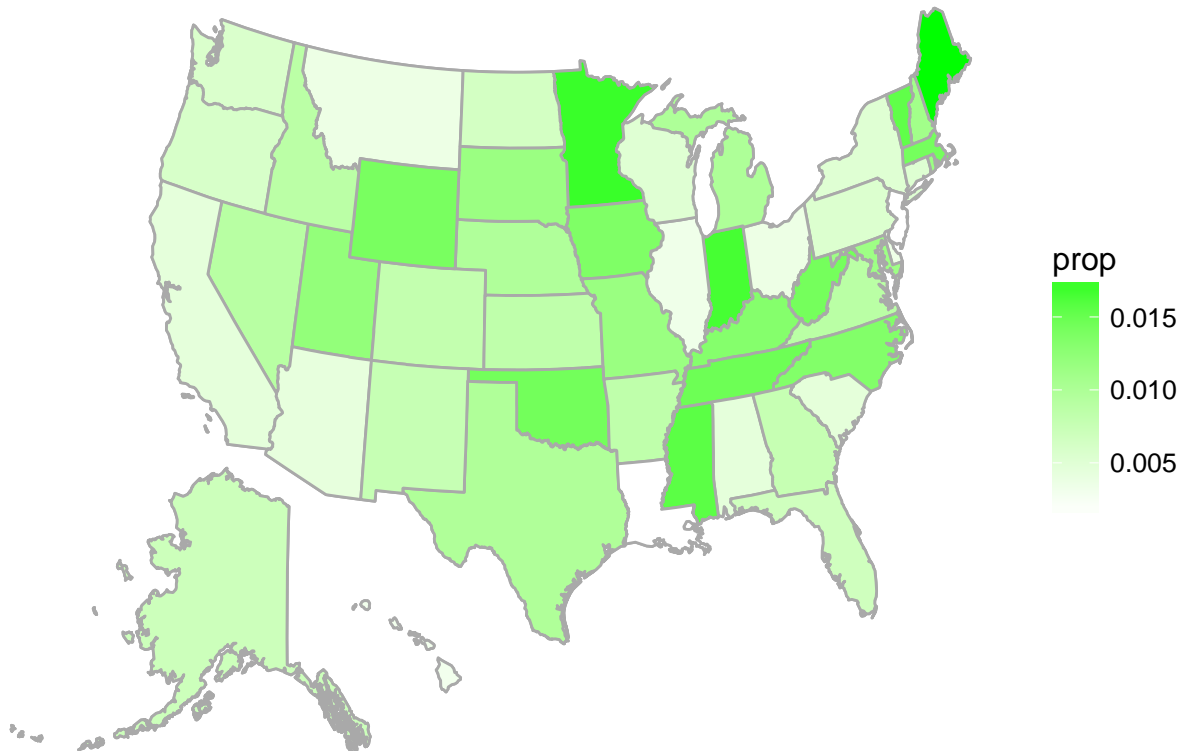


Housing Service Priority

```
prophousneed <- perpnew %>%  
  filter(state != "N/") %>%  
  group_by(state) %>%  
  summarise(total = n(), need = sum(FU6P2 == "(1) Yes")) %>%  
  mutate(prop = need/total)  
  
ggplot(data=prophousneed, aes(prop)) +  
  geom_histogram() +  
  labs(title="Histogram for prop") +  
  labs(x="prop", y="Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

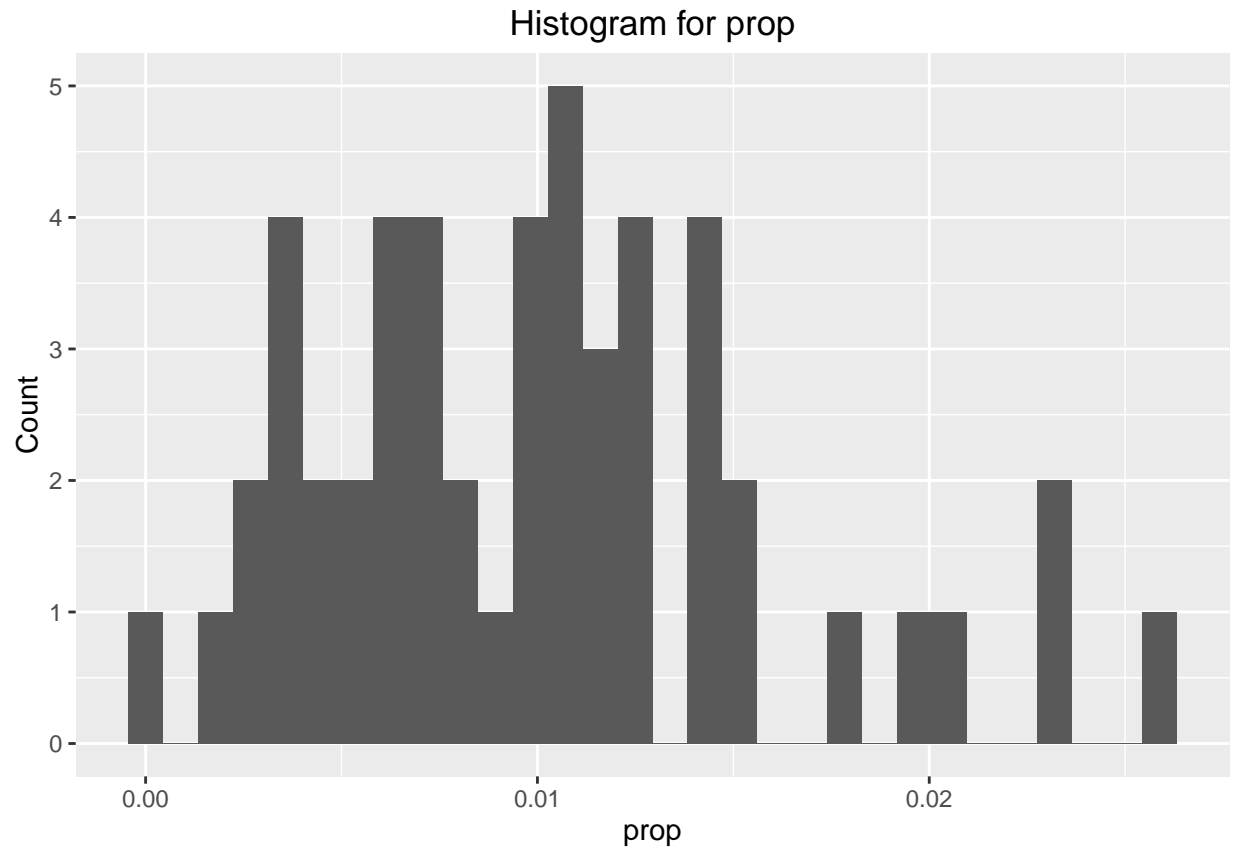

Proportion of State Housing Need



Community Service Priority

```
propcommneed <- perpnew %>%  
  filter(state != "N/") %>%  
  group_by(state) %>%  
  summarise(total = n(), need = sum(FU6P3 == "(1) Yes")) %>%  
  mutate(prop = need/total)  
  
ggplot(data=propcommneed, aes(prop)) +  
  geom_histogram() +  
  labs(title="Histogram for prop") +  
  labs(x="prop", y="Count")
```

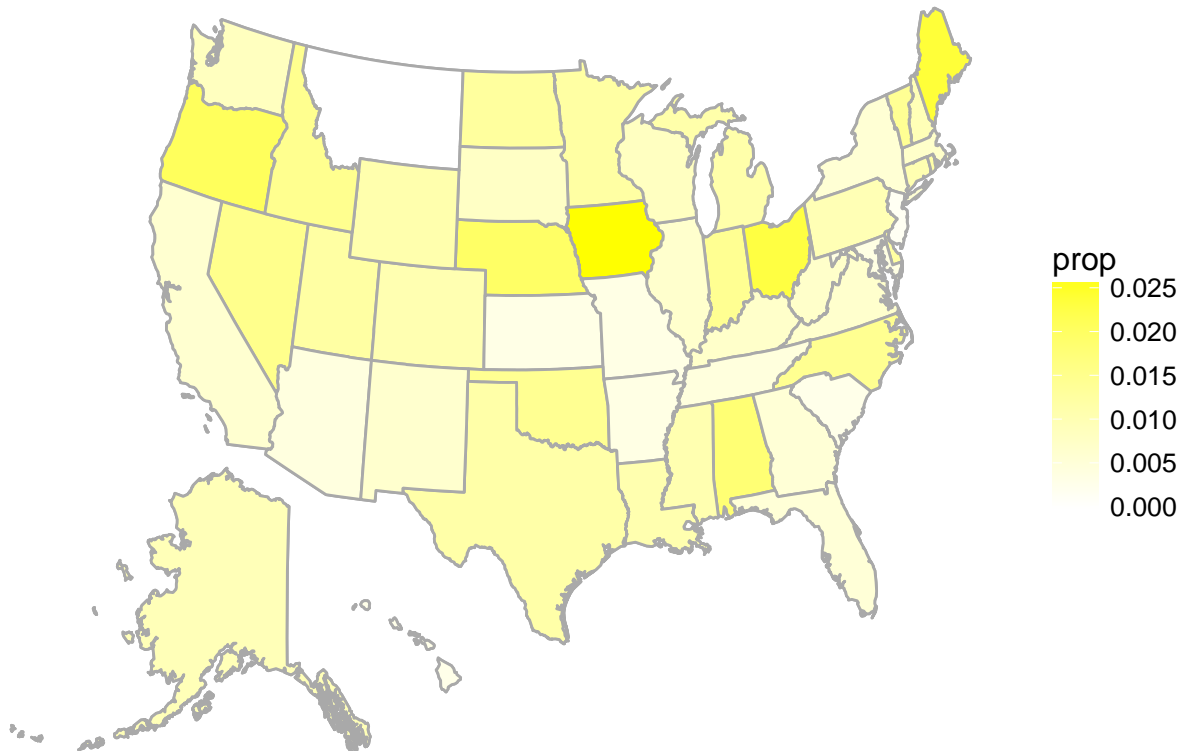
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mUSMap(propcommneed, key="state", fill = "prop")+  
  scale_fill_continuous(low="white", high="yellow")+  
  ggtitle("Proportion of State Community Need")
```

Mapping API still under development and may change in future releases.

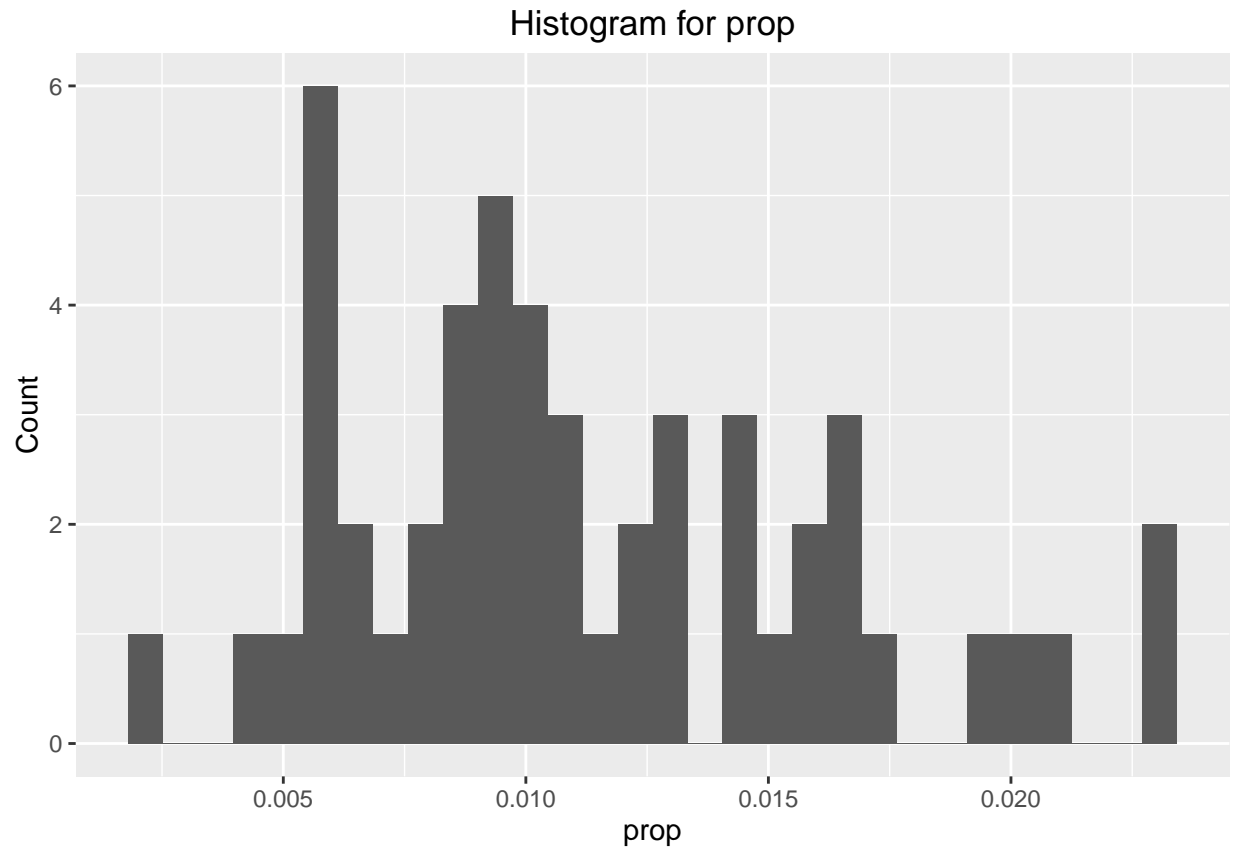
Proportion of State Community Need



Advocacy Service Priority

```
propadvoneed <- perpnew %>%  
  filter(state != "N/") %>%  
  group_by(state) %>%  
  summarise(total = n(), need = sum(FU6P4 == "(1) Yes")) %>%  
  mutate(prop = need/total)  
  
ggplot(data=propadvoneed, aes(prop)) +  
  geom_histogram() +  
  labs(title="Histogram for prop") +  
  labs(x="prop", y="Count")
```

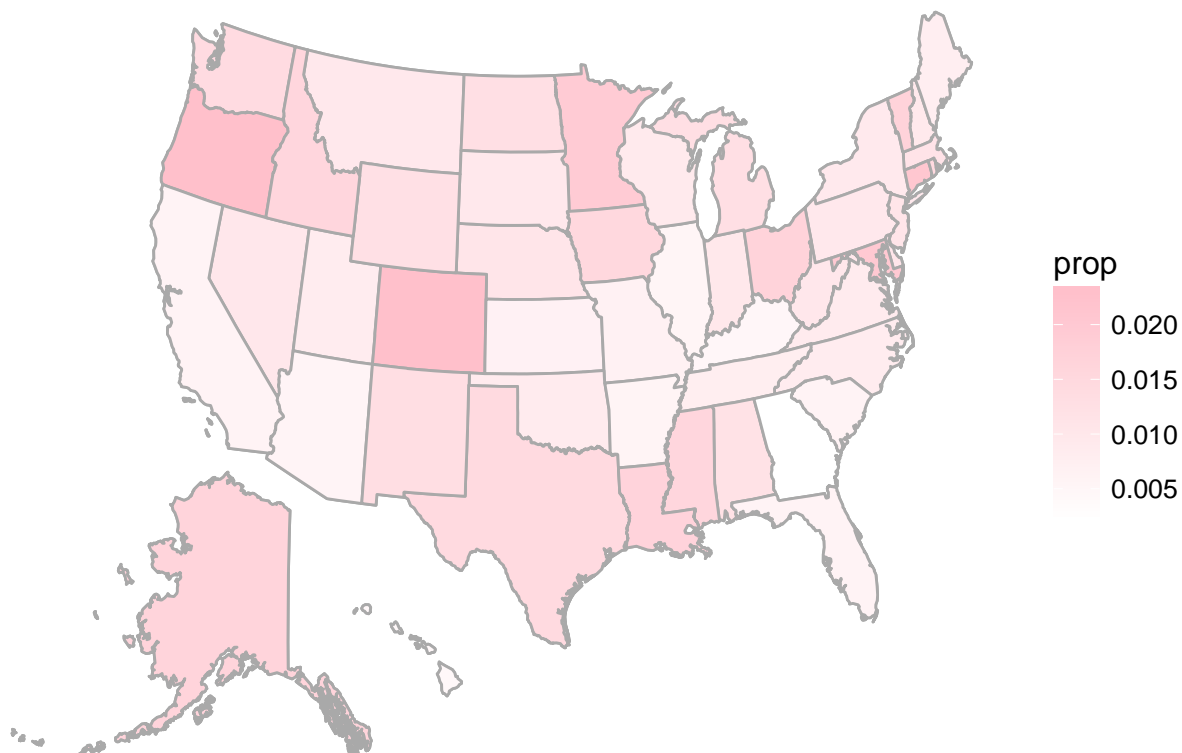
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
mUSMap(propadvoneed, key="state", fill = "prop")+  
  scale_fill_continuous(low="white", high="pink")+  
  ggtitle("Proportion of Advocacy Need")
```

Mapping API still under development and may change in future releases.

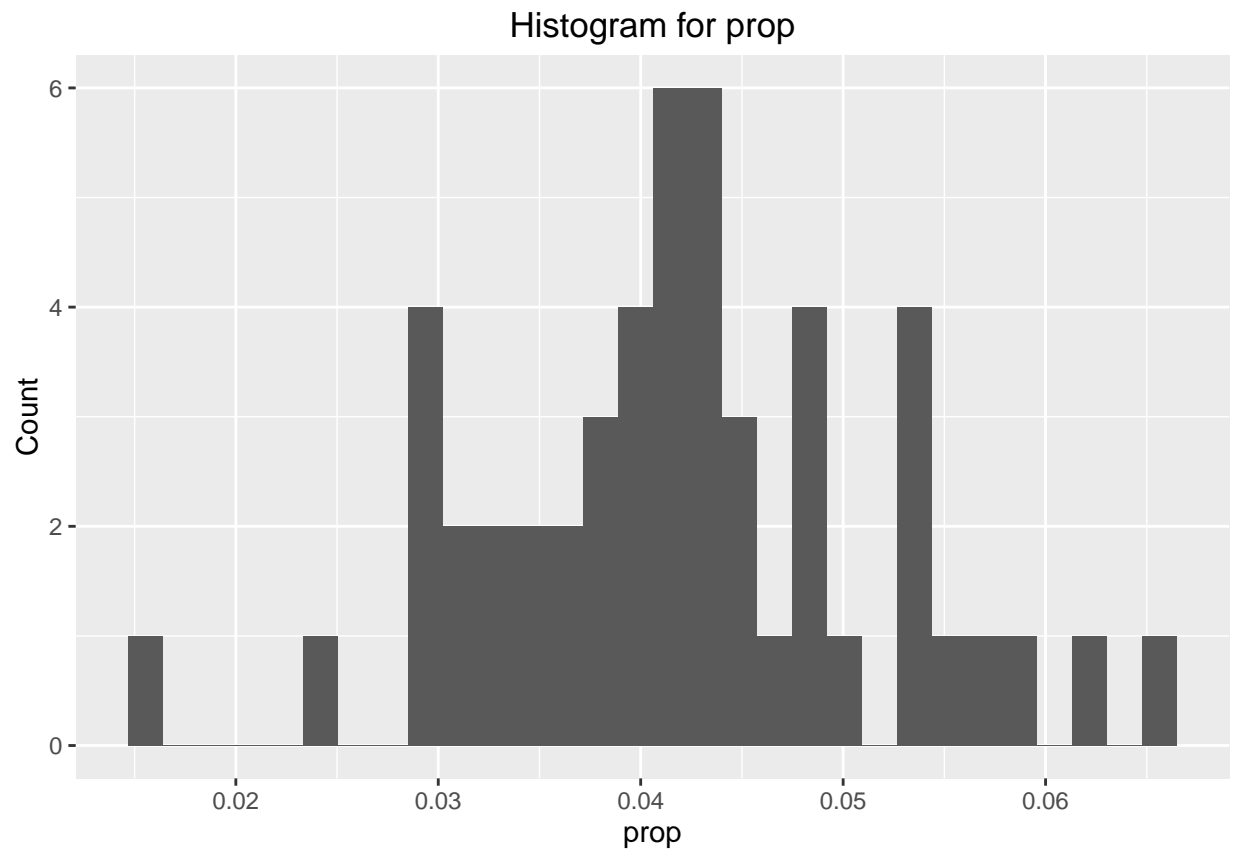
Proportion of Advocacy Need



Legal Service Priority

```
proplegalneed <- perpnew %>%  
  filter(state != "N/") %>%  
  group_by(state) %>%  
  summarise(total = n(), need = sum(FU6P5 == "(1) Yes")) %>%  
  mutate(prop = need/total)  
  
ggplot(data=proplegalneed, aes(prop)) +  
  geom_histogram() +  
  labs(title="Histogram for prop") +  
  labs(x="prop", y="Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mUSMap(proplegalneed, key="state", fill = "prop")+  
  scale_fill_continuous(low="white", high="orange")+  
  ggtitle("Proportion of Legal Need")
```

Mapping API still under development and may change in future releases.

Proportion of Legal Need

