PumpItUp_ExploratoryAnalysis.R

Jeremy

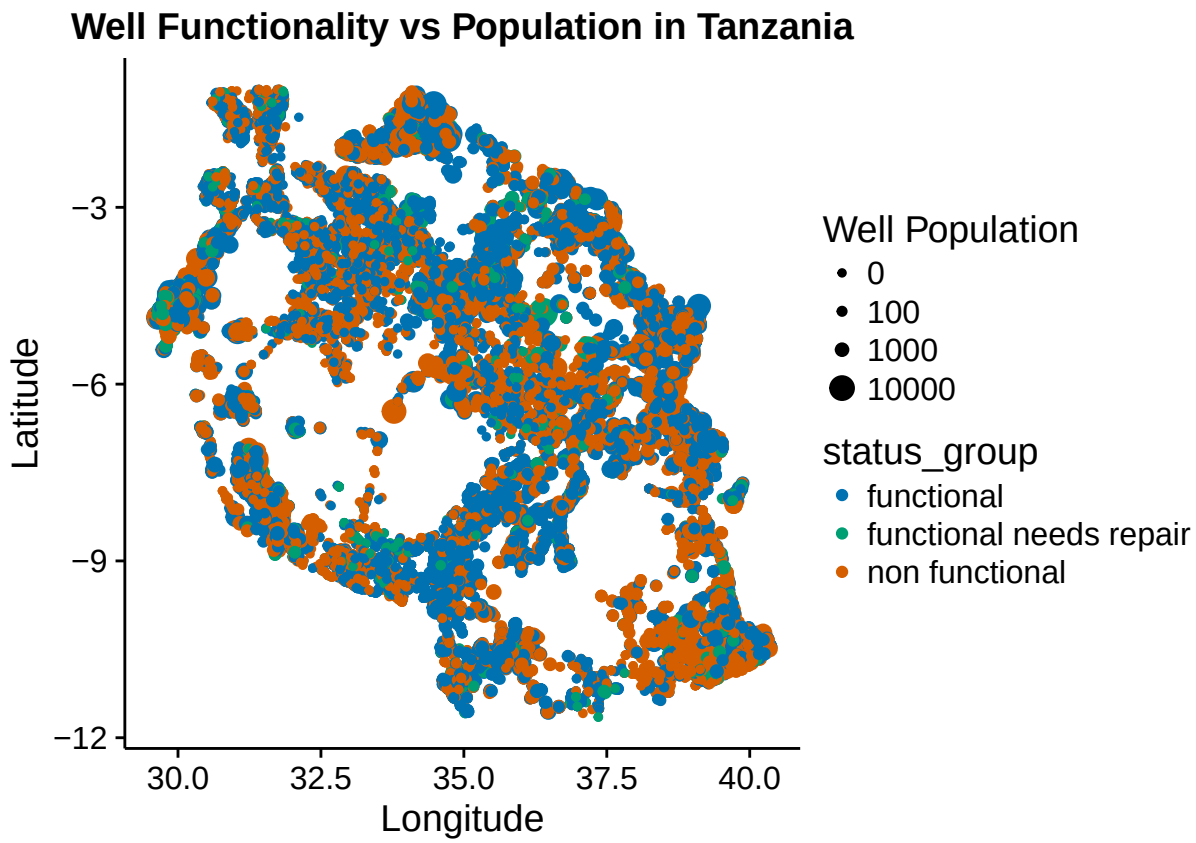
Sat Mar 18 18:27:29 2017

```
#Load packages
library(ggplot2)
library(reshape2)
library(readr)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

```
# Load datasets
setwd("C:/Users/Jeremy/OneDrive/Data Science/Data Driven")
train <- read.csv("train.csv")
# test <- read.csv("test.csv")

######################## Well Location ################################
# Plots of well functionality based on population in Tanzania show where some clusters of functional,
# non functional, and needing repair wells exist. Overall it appears that functional and non functional
# are fairly evenly dispersed throughout the country.

ggplot(subset(train, longitude > 0), aes(x = longitude, y = latitude)) +
        geom_point(aes(color = status_group, size = population)) +
        scale_colour_manual(values= c("#0072B2", "#009E73", "#D55E00")) +
        scale_size("Well Population",breaks=c(0, 100, 1000, 10000),labels=c(0,100,1000,10000)) +
        labs(title = 'Well Functionality vs Population in Tanzania', x = 'Longitude', y = 'Latitude')
```



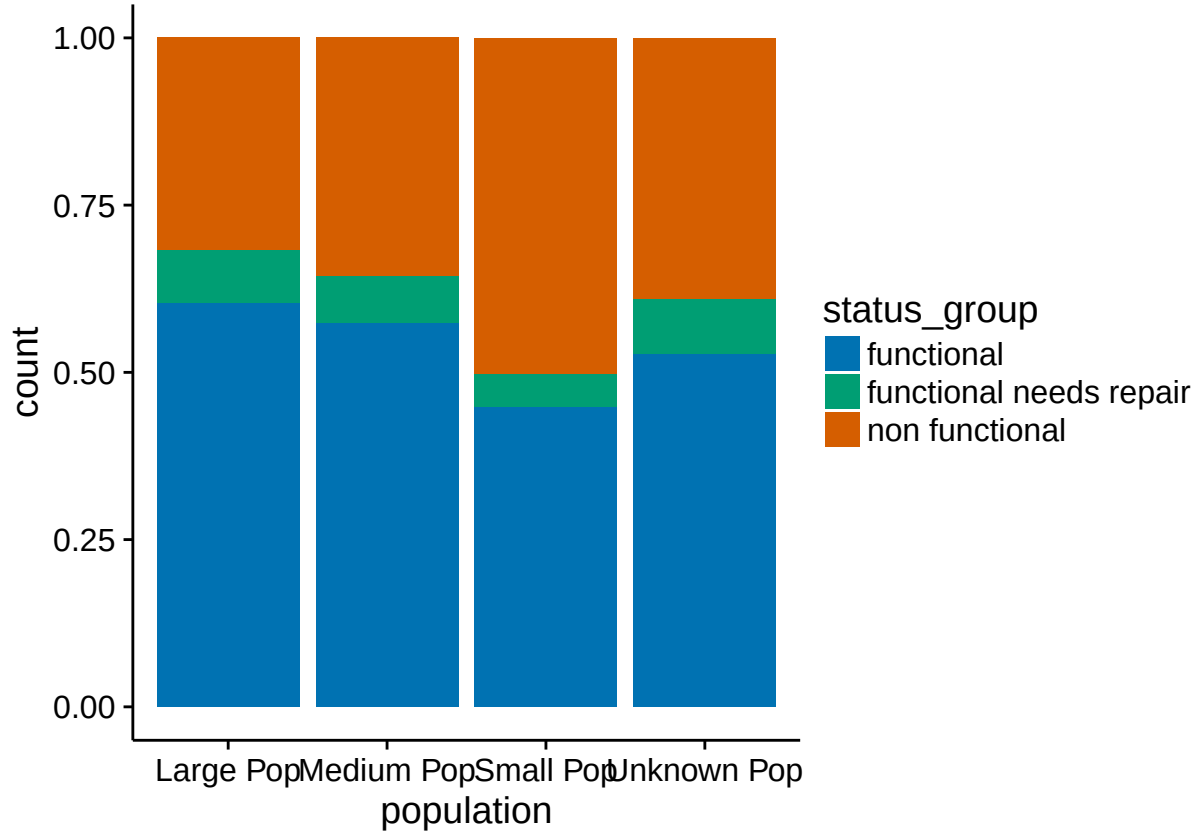

```
# Create factor levels of small, medium, large, and unknown population.
for (i in 1:nrow(train)) {
        if (train$population[i] > 500) {
                train$population[i] <- 'Large Pop'
        }
        else if (train$population[i] > 100) {
                train$population[i] <- 'Medium Pop'
        }
        else if (train$population[i] > 0) {
                train$population[i] <- 'Small Pop'
        }
        else {
                train$population[i] <- 'Unknown Pop'
        }
}
train$population <- as.factor(train$population)
summary(train$population)
```

```
##   Large Pop  Medium Pop   Small Pop Unknown Pop
##        6493       23192        8334       21381
```

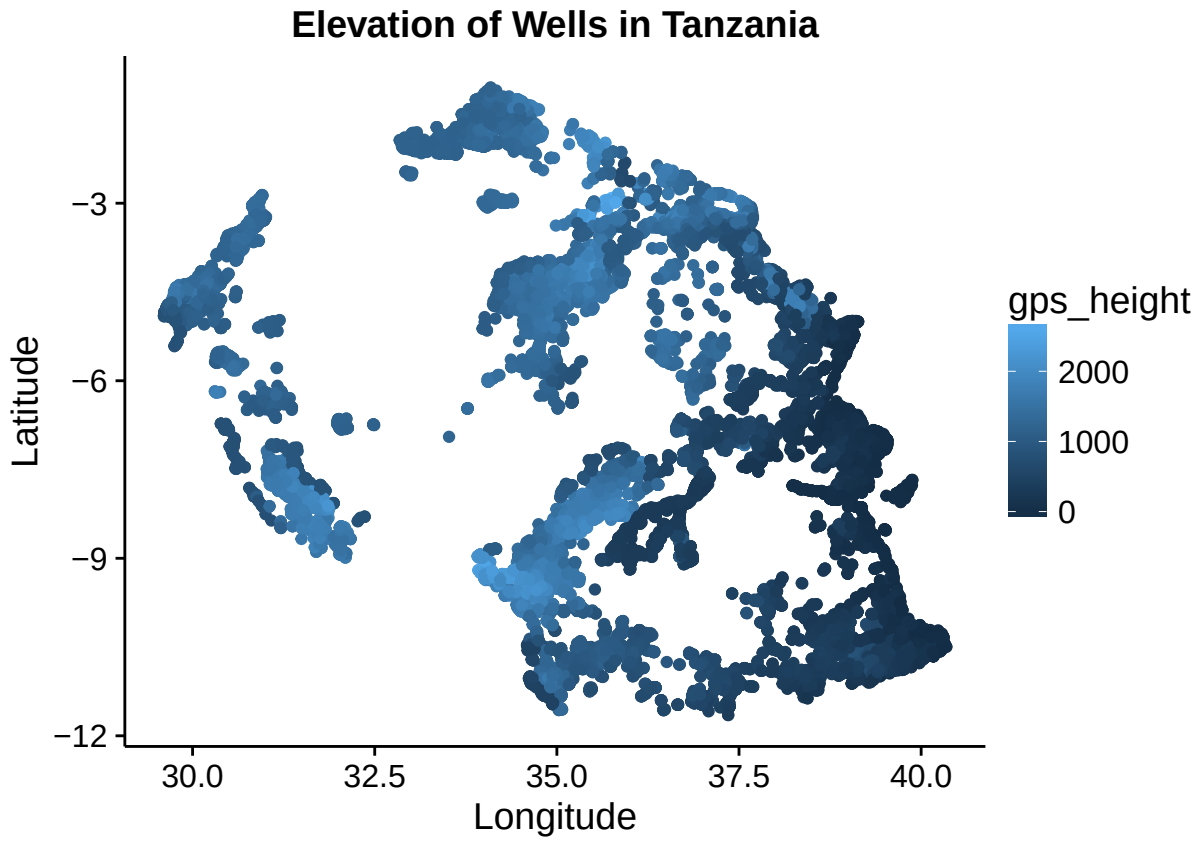
```
##   Large Pop  Medium Pop   Small Pop Unknown Pop
##        6493       23192        8334       21381
```

```
# Plot of population factors shows larger populations tend to have more functional wells, while the unk
ggplot(train, aes(population, fill = status_group)) +
        geom_bar(position = 'fill') +
```

```
        scale_fill_manual(values= c("#0072B2", "#009E73", "#D55E00"))
```




```
# Remove rows with gps elevation of 0 recorded, as most of them are clearly missing values and obscure
well_heights <- subset(train, gps_height != 0)
# Plot of elevation shows fairly linear trend accross the country
ggplot(well_heights, aes(x = longitude, y = latitude)) +
        geom_point(aes(color = gps_height)) +
        labs(title = 'Elevation of Wells in Tanzania', x = 'Longitude', y = 'Latitude')
```




```
# Regression model which can be used to impute elevation of wells with missing data. This methodology m
elev.fit <- lm(gps_height ~ longitude + latitude, data = well_heights)
summary(elev.fit)
```

```
##
## Call:
## lm(formula = gps_height ~ longitude + latitude, data = well_heights)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1119.99  -433.93   -54.48   381.20  1588.49
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5892.4697    35.0168  168.28   <2e-16 ***
## longitude   -128.5318     0.9882 -130.06   <2e-16 ***
## latitude      39.7443     0.9098   43.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 490.8 on 38959 degrees of freedom
## Multiple R-squared:  0.3581, Adjusted R-squared:  0.358
## F-statistic: 1.087e+04 on 2 and 38959 DF,  p-value: < 2.2e-16
```

```
####################### Funder Analysis ######################
# Combine typo factor levels for predictive funders
for (i in 1:nrow(train)) {
```

```
        if (train$funder[i] == 'Ces(gmbh)') {
                train$funder[i] <- 'Ces (gmbh)'
        }
        else if (train$funder[i] == 'Dwssp') {
                train$funder[i] <- 'Dwsp'
        }
        else if (train$funder[i] == 'Fin Water' | train$funder[i] == 'Finw' | train$funder[i] == 'Finwa
                train$funder[i] <- 'Fini Water'
        }
        else if (train$funder[i] == 'Rc' | train$funder[i] == 'Rc Ch' | train$funder[i] == 'Rc Churc' |
                train$funder[i] <- 'Rc Church'
        }
        else if (train$funder[i] == 'Tassaf') {
                train$funder[i] <- 'Tasaf'
        }
}

# Pivot table of funders by well functionality
funder.pt <- as.data.frame(dcast(train, funder ~ status_group, length))
```

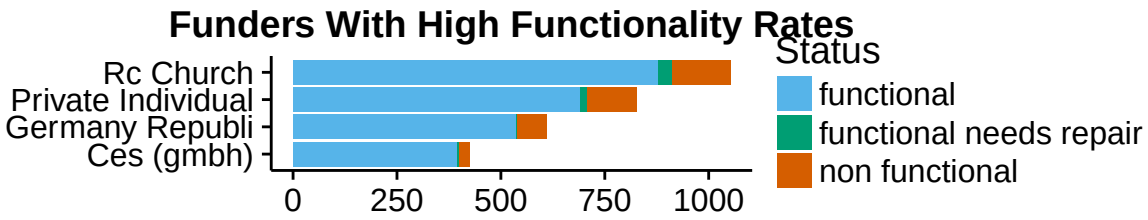
```
## Using status_group as value column: use value.var to override.
```

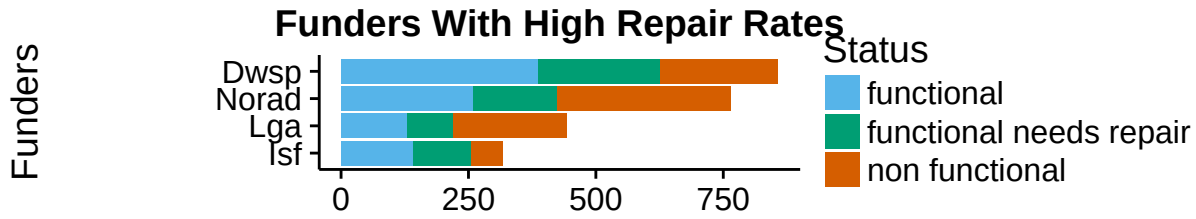
```
## Using status_group as value column: use value.var to override.
# store funder names to recombine with pivot table after calculations
funders <- funder.pt$funder
# remove funder column for processing
funder.pt <- funder.pt[, -which(names(funder.pt) == "funder")]
# Divide each well type by the row total to get percentage
num_wells <- character()
for (i in 1:nrow(funder.pt)) {
        temp <- sum(funder.pt[i,])
        num_wells <- c(num_wells, temp)
        if (temp != 0) {
                funder.pt[i,1] <- funder.pt[i,1] / temp
                funder.pt[i,2] <- funder.pt[i,2] / temp
                funder.pt[i,3] <- funder.pt[i,3] / temp
        }
}
# Add back funder names and the number of wells for each funder
funder.pt$num_wells <- as.numeric(num_wells)
funder.pt$funder <- funders

# Determine baseline level for function, repair, and non-function
summary(train$status_group) / 59400
```

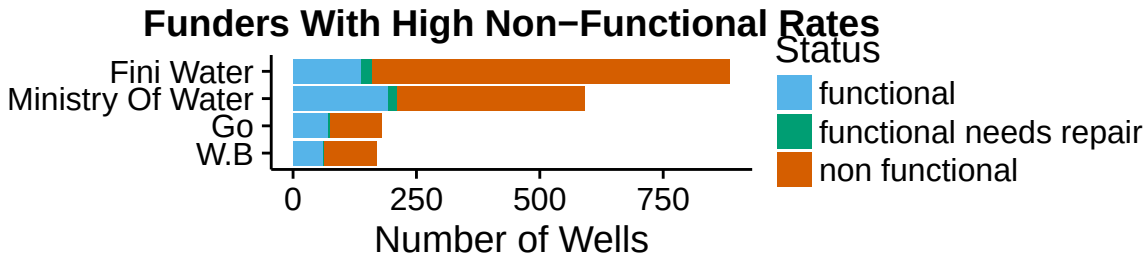
```
##              functional functional needs repair          non functional
##              0.54308081              0.07267677              0.38424242
```

```
##              functional functional needs repair          non functional
##              0.54308081              0.07267677              0.38424242

# Extract most predictive funder names based on functionality proportion and sample size
top_functional <- character()
top_repair <- character()
top_nonfunctional <- character()
```

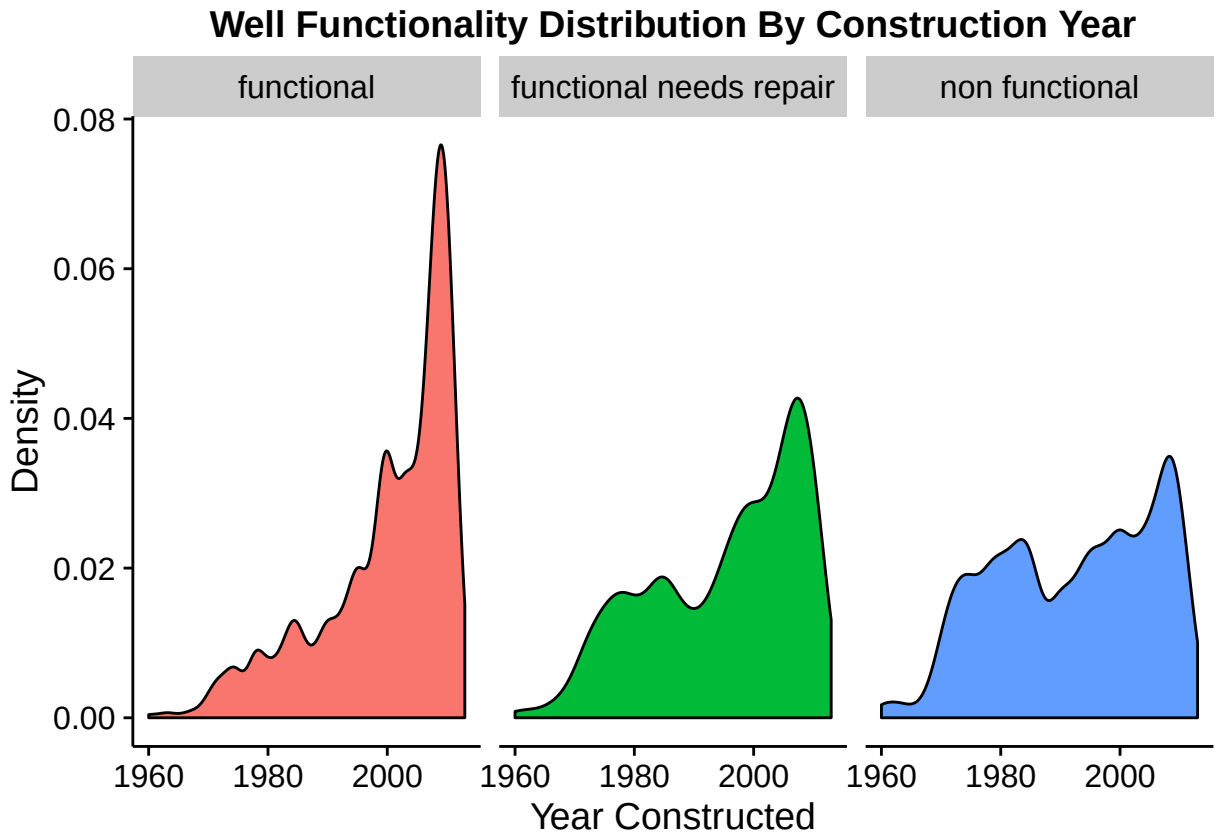
```
for (i in 1:nrow(funder.pt)) {
        if (funder.pt[i,1] > 0.82 && funder.pt[i,4] > 200 && !(as.character(funder.pt[i,5]) %in% top_fu
                top_functional <- c(top_functional, as.character(funder.pt[i,5]))
        }
        else if (funder.pt[i,2] > 0.2 && funder.pt[i,4] > 250 && !(as.character(funder.pt[i,5]) %in% to
                top_repair <- c(top_repair, as.character(funder.pt[i,5]))
        }
        else if (funder.pt[i,3] > 0.58 && funder.pt[i, 4] > 140 && !(as.character(funder.pt[i,5]) %in%
                top_nonfunctional <- c(top_nonfunctional, as.character(funder.pt[i,5]))
        }
}

func <- subset(train, funder %in% top_functional)
p1 <- qplot(reorder(factor(funder),factor(funder),length),
            data = func,geom = "bar", fill = status_group,
            xlab = '',
            ylab = '',
            main = 'Funders With High Functionality Rates') +
        scale_fill_manual(values = c('#56B4E9', '#009E73', '#D55E00'), guide = guide_legend(title = 'Sta
        coord_flip()
repair <- subset(train, funder %in% top_repair)
p2 <- qplot(reorder(factor(funder),factor(funder),length),
            data = repair,geom = "bar", fill = status_group,
            xlab = 'Funders',
            ylab = '',
            main = 'Funders With High Repair Rates') +
        scale_fill_manual(values = c('#56B4E9', '#009E73', '#D55E00'), guide = guide_legend(title = 'Sta
        coord_flip()
nonfunc <- subset(train, funder %in% top_nonfunctional)
p3 <- qplot(reorder(factor(funder),factor(funder),length),
            data = nonfunc,geom = "bar", fill = status_group,
            xlab = '',
            ylab = 'Number of Wells',
            main = 'Funders With High Non-Functional Rates') +
        scale_fill_manual(values = c('#56B4E9', '#009E73', '#D55E00'), guide = guide_legend(title = 'Sta
        coord_flip()

# Plot of most funders with the most significant proportions of function, needing repair, and non-funct
plot_grid(p1, p2, p3, ncol = 1, align = 'v')
```








```
###################### Construction Year Analysis ######################
# Subset dataset to remove missing construction year values
with_constr_date <- subset(train, construction_year != 0)

# Plot showing how well functionality is distributed over time. Among functional wells, a much higher p
ggplot(with_constr_date, aes(construction_year, fill = status_group), fill = status_group) +
        geom_density(position = 'identity', show.legend = FALSE) +
        facet_wrap(~ status_group) +
        scale_color_identity() +
        labs(title = 'Well Functionality Distribution By Construction Year', x = 'Year Constructed', y =
```



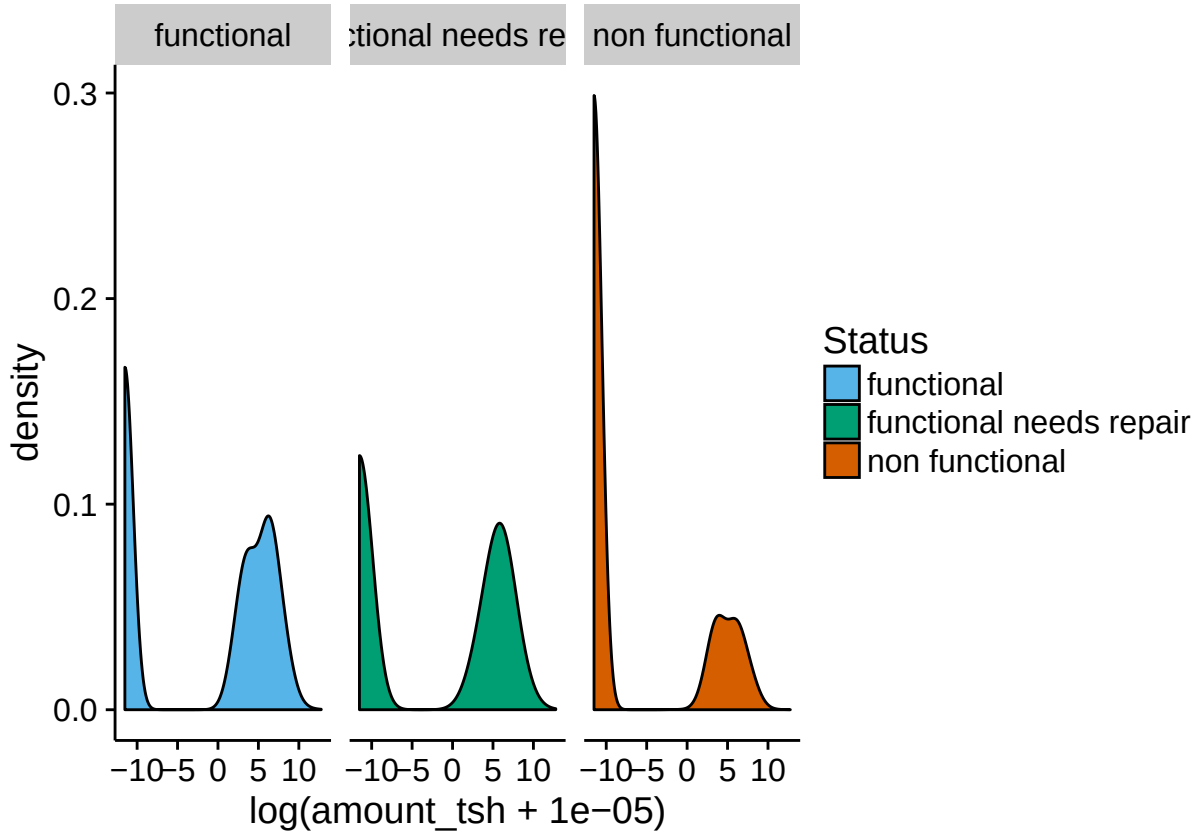

```
###################### Amount of water at wells ########################
# Analysis of wells with no water
noWaterData <- subset(train, amount_tsh == 0)
table(noWaterData$status_group)
```

```
##
##              functional functional needs repair          non functional
##                   19706                    3048                   18885
```

```
# Of the 20438 recorded data points with no gps height, 20073 have a value of zero recorded for the amo
# While Tanzania does have a stretch of coastline, it is seems likely that most of the records without
no_gps_record <- subset(train, gps_height == 0)
table(no_gps_record$amount_tsh)
```

```
##
##     0    10    20    25    30    40    50   100   150   200   300   500
## 20073    12    83     2    30     1    44     1     1    20     7    12
##  1000  1200  1500  2000  3000
##   106    20     8    16     2
```

```
# Analysis of data with non-zero recorded amount of water
waterAvail <- subset(train, amount_tsh != 0 | gps_height != 0)
# Add small number to the amount of water so that log can be taken of values of zero
ggplot(waterAvail, aes(log(amount_tsh + .00001), fill = status_group)) +
        geom_density() +
        facet_wrap(~ status_group) +
        scale_fill_manual(values = c('#56B4E9', '#009E73', '#D55E00'), guide = guide_legend(title = 'Sta
```


functional
ctional needs re
non functional
0.3
0.2
0.1
0.0
density
−10
−5
0
5
10
log(amount_tsh + 1e−05)
Status
functional
functional needs repair
non functional