

Data Visualizations and Analysis

Grayson White

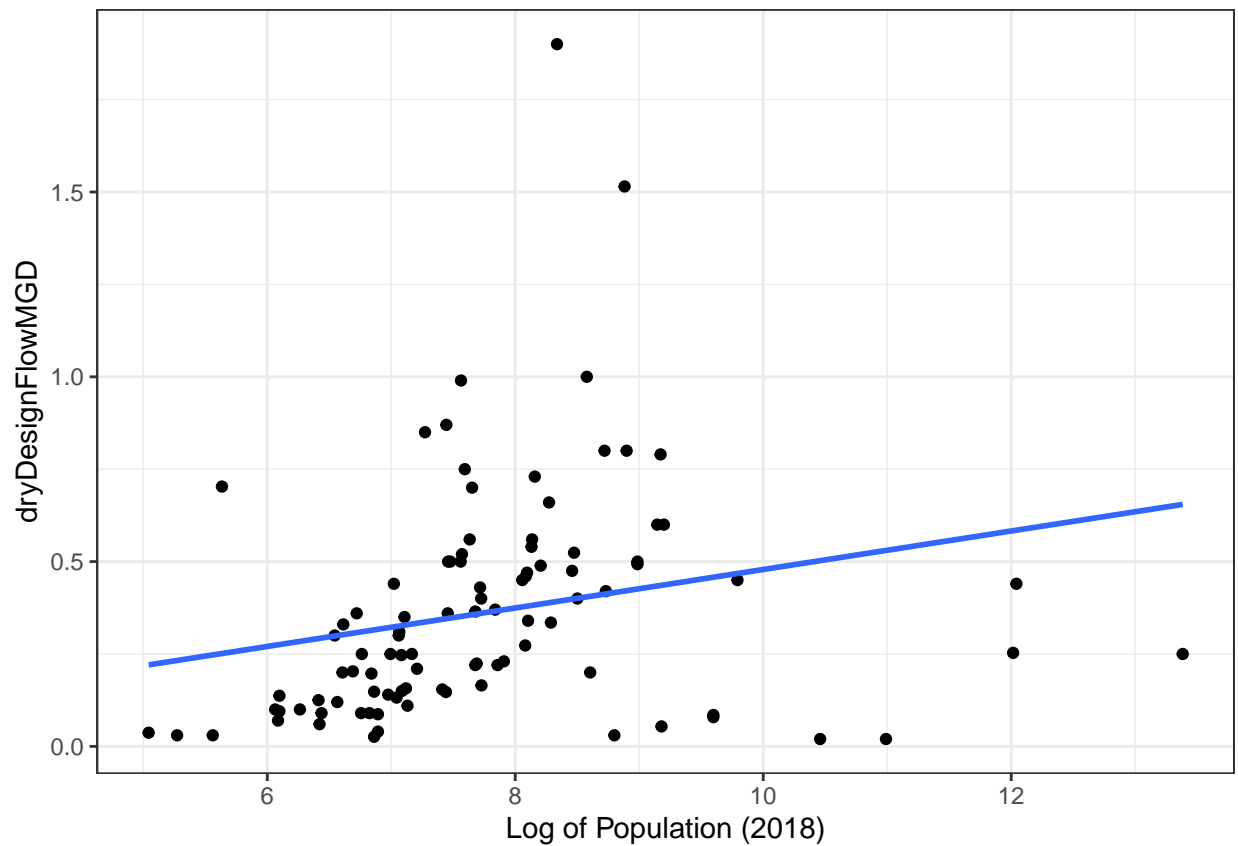
7/13/2020

```
ggplot(working_df, aes(x = log(pop_2018), y = dryDesignFlowMGD)) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE) +  
  theme_bw() +  
  labs(x = "Log of Population (2018)")
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

```
## Warning: Removed 17 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 17 rows containing missing values (geom_point).
```

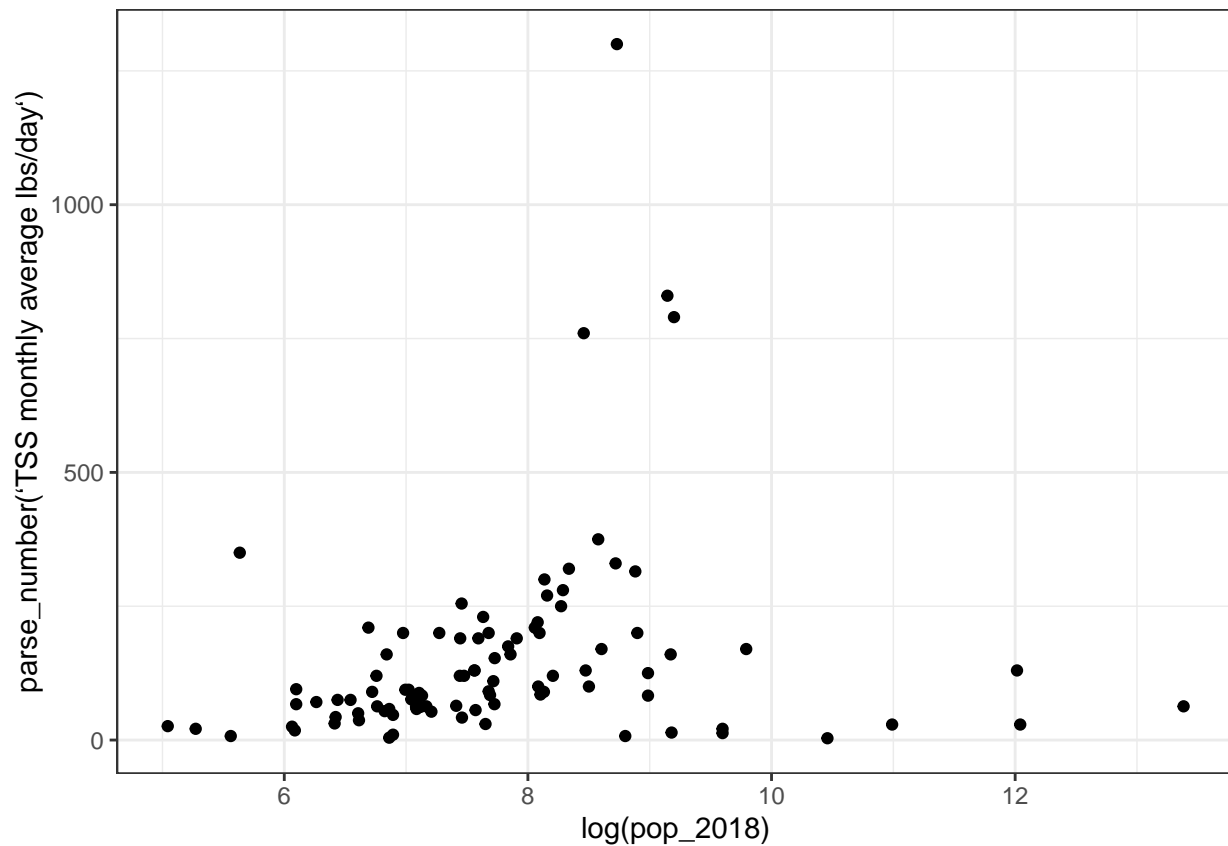


```
ggplot(working_df, aes(x = log(pop_2018),
                      y = parse_number('TSS monthly average lbs/day'))) +
  geom_point() +
  theme_bw()
```

```
## Warning: 2 parsing failures.
## row col expected actual
## 13  -- a number      na
## 45  -- a number      na
```

```
## Warning: 2 parsing failures.
## row col expected actual
## 13  -- a number      na
## 45  -- a number      na
```

```
## Warning: Removed 19 rows containing missing values (geom_point).
```



```
working_df %>%
  filter(type1 %in% c("lagoons", "activated sludge")) %>%
  group_by(type1) %>%
  summarize(median = median(pop_2018, na.rm = TRUE))
```

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

```
## # A tibble: 2 x 2
##   type1      median
##   <chr>      <dbl>
## 1 activated sludge 1962.
## 2 lagoons        1718.
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(plotly)
```

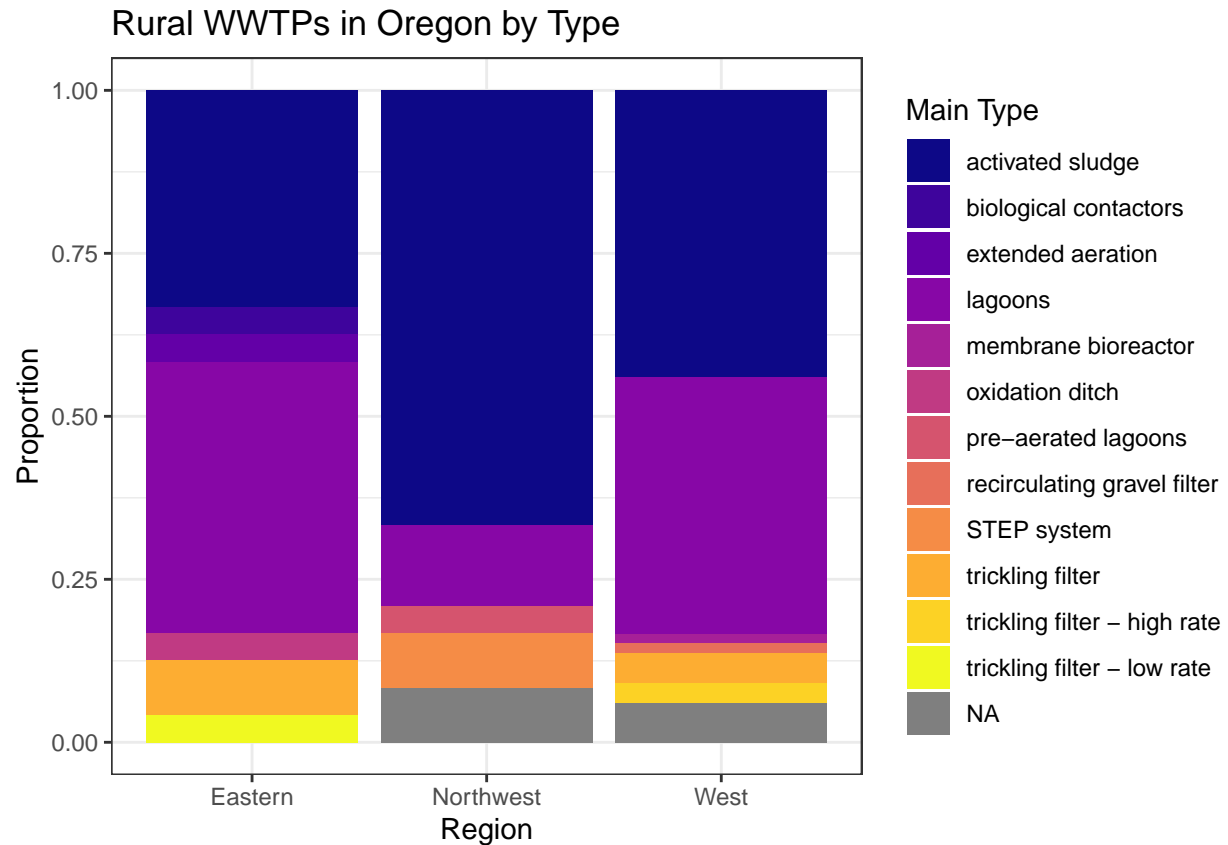
```
##
## Attaching package: 'plotly'
```

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

```
## The following object is masked from 'package:stats':
##
##   filter
```

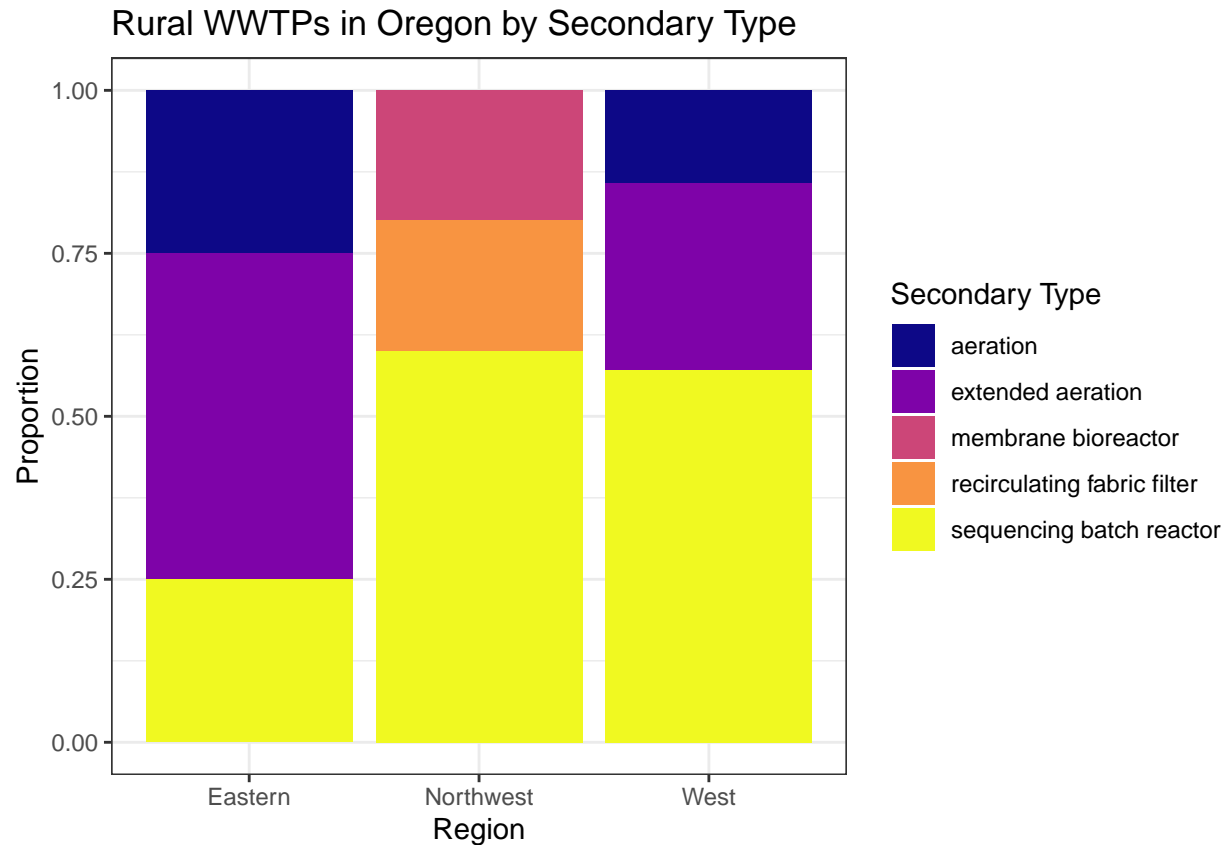
```
## The following object is masked from 'package:graphics':
##
##   layout
```

```
p <- working_df %>%
  ggplot(aes(x = Region.x,
             fill = type1)) +
  geom_bar(position = "fill") +
  scale_fill_viridis_d(option = "C", na.value = "grey50") +
  scale_x_discrete(labels=c("Eastern", "Northwest", "West")) +
  theme_bw() +
  labs(x = "Region",
       fill = "Main Type",
       y = "Proportion",
       title = "Rural WWTPs in Oregon by Type")
p
```



```
# ggplotly(p, tooltip = c("type1", "count"))

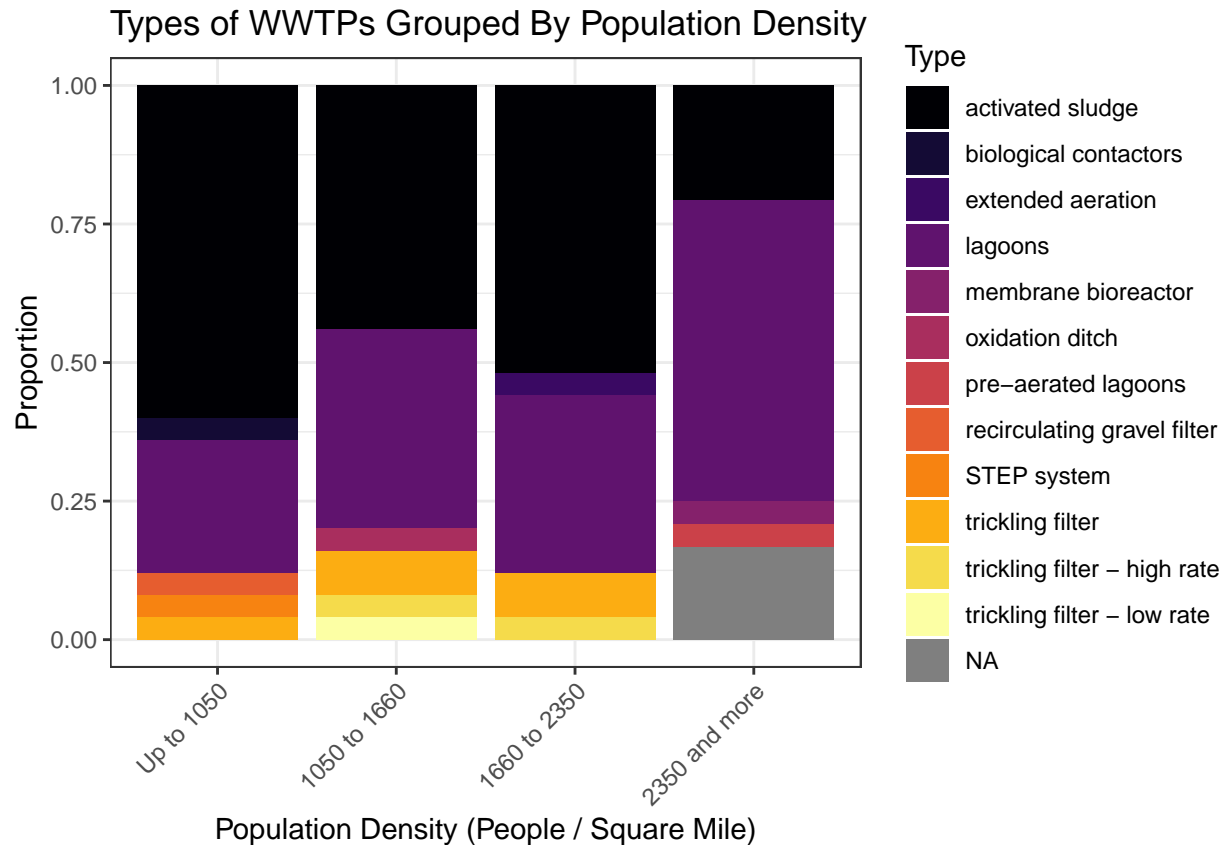
working_df %>%
  filter(type2 != c("na", NA)) %>%
  ggplot(aes(x = Region.x,
             fill = type2)) +
  geom_bar(position = "fill") +
  scale_fill_viridis_d(option = "C", na.value = "grey50") +
  scale_x_discrete(labels=c("Eastern", "Northwest", "West")) +
  theme_bw() +
  labs(x = "Region",
       fill = "Secondary Type",
       y = "Proportion",
       title = "Rural WWTPs in Oregon by Secondary Type")
```



```
library(gtools)
```

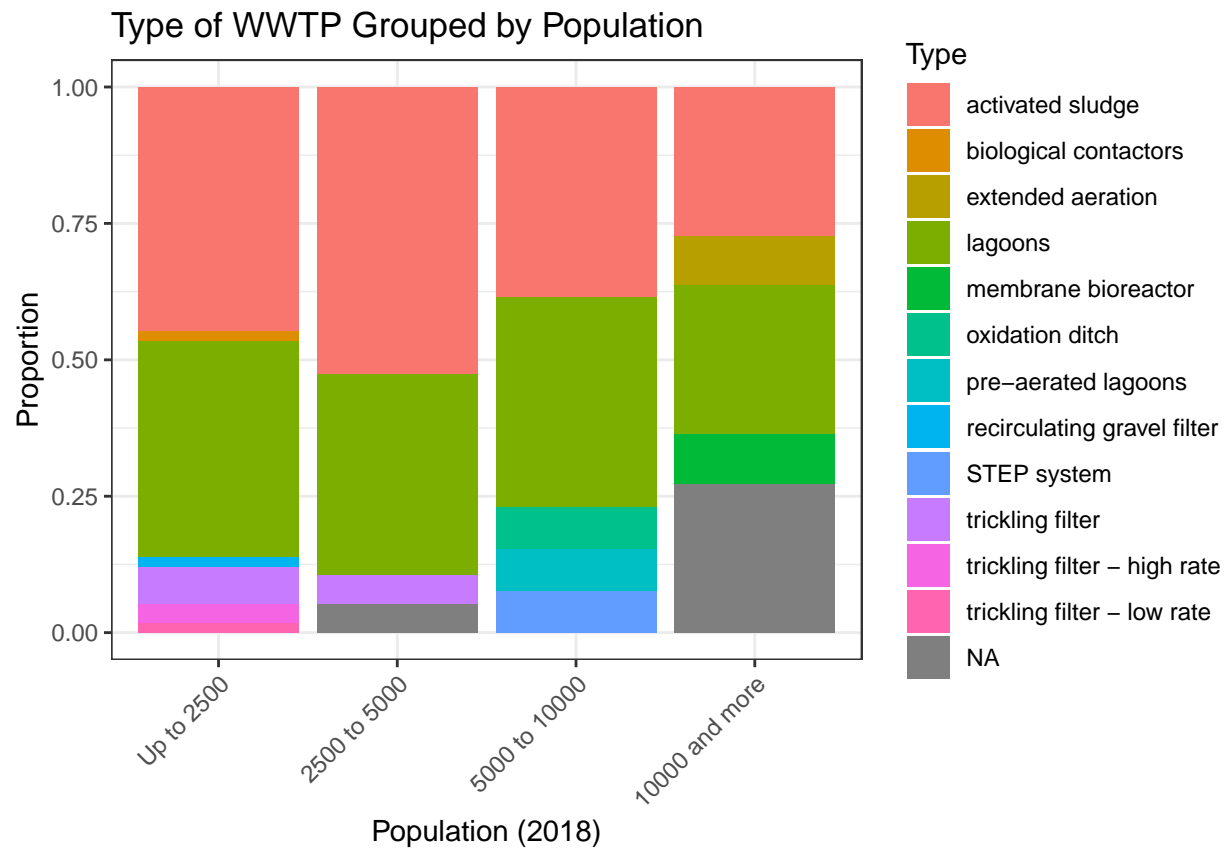
```
## Warning: package 'gtools' was built under R version 4.0.1
```

```
working_df$quantile_density <- quantcut(working_df$pop_density_2018, q = 4)
ggplot(working_df[!is.na(working_df$quantile_density), ], aes(x = quantile_density,
  fill = type1)) +
  geom_bar(position = "fill") +
  scale_x_discrete(labels=c("Up to 1050", "1050 to 1660", "1660 to 2350", "2350 and more")) +
  scale_fill_viridis_d(option = "B", na.value = "grey50") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  labs(x = "Population Density (People / Square Mile)",
    y = "Proportion",
    fill = "Type",
    title = "Types of WWTPs Grouped By Population Density")
```



```
working_df$quantile_pop_2018 <- quantcut(working_df$pop_2018, q = 4)
working_df$cut_pop_2018 <- cut(working_df$pop_2018, breaks = c(0, 2500, 5000, 10000, 1e7))
p1 <- ggplot(working_df[!is.na(working_df$cut_pop_2018), ], aes(x = cut_pop_2018,
  fill = type1)) +
  geom_bar(position = "fill") +
  scale_x_discrete(labels=c("Up to 2500", "2500 to 5000", "5000 to 10000", "10000 and more")) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  labs(x = "Population (2018)",
    y = "Proportion",
    title = "Type of WWTP Grouped by Population",
    fill = "Type")
```

p1



```
# ggplotly(p1)
```