ENVE 661 HW2 Problem 10-4

Tyler Bradley

April 25, 2018

This document is the work and the results for problem 10-4 of homework assignment #2 in ENVE 661.

First, the required libraries were loaded and the dataset from the data given in example 10-3 in the textbook was created.

```r
# load required libraries
library(tidyverse)
library(knitr)
library(kableExtra)
```

```r
# creating the influent dataset
influent_data <- tibble::tribble(
  ~avg_Vs, ~count_inf_part,
  0.2, 511,
  0.6, 657,
  1.0, 876,
  1.4, 1168,
  1.8, 1460,
  2.2, 1314,
  2.6, 657,
  3.0, 438,
  3.4, 292,
  3.8, 292
) %>%
  nest()

# creating vector of different settling velocities
Vc <- seq(0.5, 4.0, by = 0.5)

# combining the influent data for each of the critical velocities
influent_data <- crossing(Vc, influent_data)
```

Next, the fraction of particles removed and the count of particles removed were calculated for each avg settling velocity at each of the critical velocities.

```r
# calculating the fraction removed and the count of particles removed
# for each settling velocity at each of the eight critical velocities
removal_data <-  influent_data %>%
  mutate(
    data = map2(Vc, data, ~{
      .y %>%
        mutate(frac_removed = avg_Vs/.x,
               frac_removed = if_else(frac_removed > 1, 1, frac_removed),
               count_removed = frac_removed*count_inf_part)
    })
  ) %>%
  unnest(data)

knitr::kable(removal_data, format = "latex", booktabs = TRUE, longtable = TRUE) %>%
```

```r
collapse_rows(columns = 1, latex_hline = "major")
```

Vc	avg_Vs	count_inf_part	frac_removed	count_removed
0.5	0.2	511	0.4000000	204.40000
	0.6	657	1.0000000	657.00000
	1.0	876	1.0000000	876.00000
	1.4	1168	1.0000000	1168.00000
	1.8	1460	1.0000000	1460.00000
	2.2	1314	1.0000000	1314.00000
	2.6	657	1.0000000	657.00000
	3.0	438	1.0000000	438.00000
	3.4	292	1.0000000	292.00000
	3.8	292	1.0000000	292.00000
1.0	0.2	511	0.2000000	102.20000
	0.6	657	0.6000000	394.20000
	1.0	876	1.0000000	876.00000
	1.4	1168	1.0000000	1168.00000
	1.8	1460	1.0000000	1460.00000
	2.2	1314	1.0000000	1314.00000
	2.6	657	1.0000000	657.00000
	3.0	438	1.0000000	438.00000
	3.4	292	1.0000000	292.00000
	3.8	292	1.0000000	292.00000
1.5	0.2	511	0.1333333	68.13333
	0.6	657	0.4000000	262.80000
	1.0	876	0.6666667	584.00000
	1.4	1168	0.9333333	1090.13333
	1.8	1460	1.0000000	1460.00000
	2.2	1314	1.0000000	1314.00000
	2.6	657	1.0000000	657.00000
	3.0	438	1.0000000	438.00000
	3.4	292	1.0000000	292.00000
	3.8	292	1.0000000	292.00000
2.0	0.2	511	0.1000000	51.10000
	0.6	657	0.3000000	197.10000
	1.0	876	0.5000000	438.00000
	1.4	1168	0.7000000	817.60000
	1.8	1460	0.9000000	1314.00000
	2.2	1314	1.0000000	1314.00000
	2.6	657	1.0000000	657.00000
	3.0	438	1.0000000	438.00000
	3.4	292	1.0000000	292.00000
	3.8	292	1.0000000	292.00000
	0.2	511	0.0800000	40.88000
	0.6	657	0.2400000	157.68000
	1.0	876	0.4000000	350.40000
	1.4	1168	0.5600000	654.08000
	1.8	1460	0.7200000	1051.20000
	2.2	1314	0.8800000	1156.32000
	2.6	657	1.0000000	657.00000
	3.0	438	1.0000000	438.00000

2.5				
	3.4	292	1.0000000	292.00000
	3.8	292	1.0000000	292.00000
3.0	0.2	511	0.0666667	34.06667
	0.6	657	0.2000000	131.40000
	1.0	876	0.3333333	292.00000
	1.4	1168	0.4666667	545.06667
	1.8	1460	0.6000000	876.00000
	2.2	1314	0.7333333	963.60000
	2.6	657	0.8666667	569.40000
	3.0	438	1.0000000	438.00000
	3.4	292	1.0000000	292.00000
	3.8	292	1.0000000	292.00000
3.5	0.2	511	0.0571429	29.20000
	0.6	657	0.1714286	112.62857
	1.0	876	0.2857143	250.28571
	1.4	1168	0.4000000	467.20000
	1.8	1460	0.5142857	750.85714
	2.2	1314	0.6285714	825.94286
	2.6	657	0.7428571	488.05714
	3.0	438	0.8571429	375.42857
	3.4	292	0.9714286	283.65714
	3.8	292	1.0000000	292.00000
4.0	0.2	511	0.0500000	25.55000
	0.6	657	0.1500000	98.55000
	1.0	876	0.2500000	219.00000
	1.4	1168	0.3500000	408.80000
	1.8	1460	0.4500000	657.00000
	2.2	1314	0.5500000	722.70000
	2.6	657	0.6500000	427.05000
	3.0	438	0.7500000	328.50000
	3.4	292	0.8500000	248.20000
	3.8	292	0.9500000	277.40000

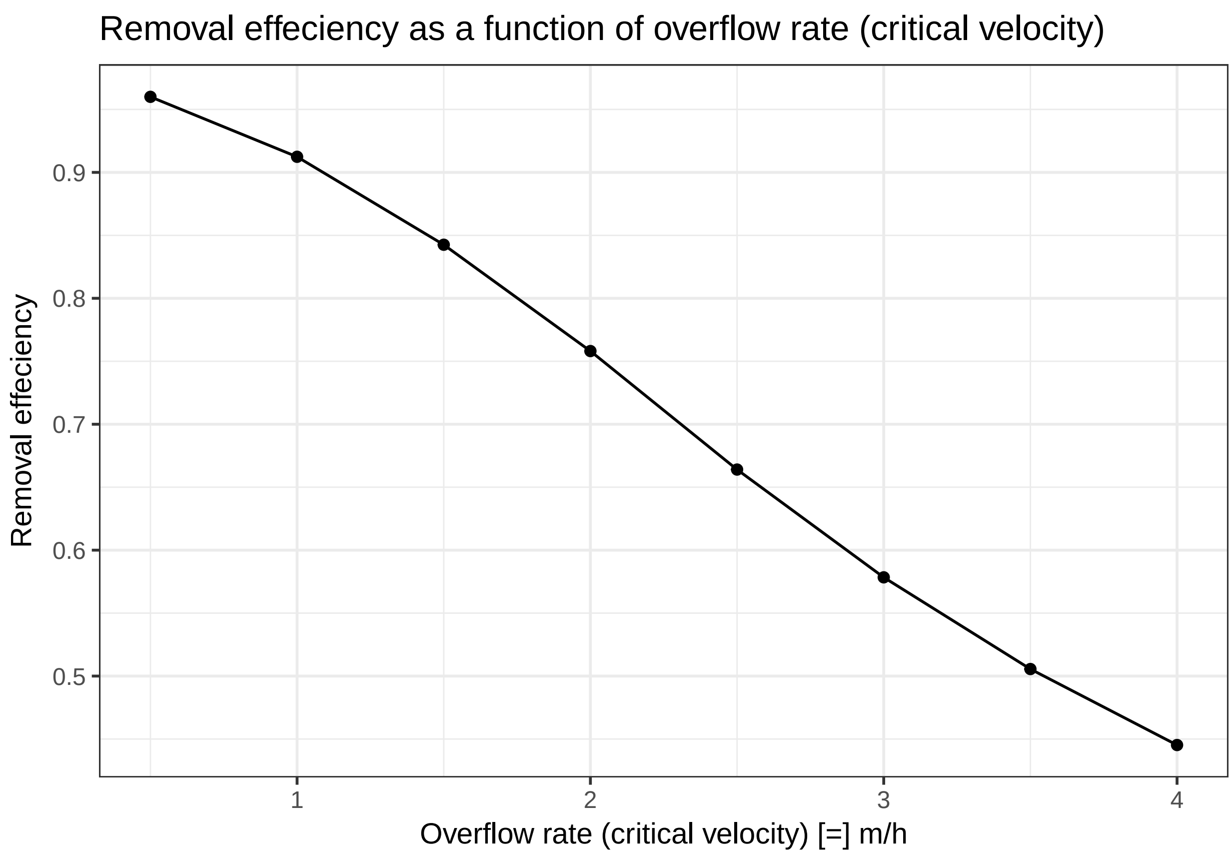
Finally the removal effeciency for each of the critical veolocities was caluclated.

```r
#' calculating the removal effeciency for each of the eight critical
#' velocities
removal_eff <- removal_data  %>%
  group_by(Vc) %>%
  summarize(removal_eff = sum(count_removed)/sum(count_inf_part))

knitr::kable(removal_eff, format = "latex", booktabs = TRUE)
```

Vc	removal_eff
0.5	0.9600000
1.0	0.9123810
1.5	0.8425397
2.0	0.7580952
2.5	0.6640000
3.0	0.5784127
3.5	0.5055782
4.0	0.4452381

```r
removal_eff %>%
  ggplot(aes(Vc, removal_eff)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Removal effeciency as a function of overflow rate (critical velocity)",
    x = "Overflow rate (critical velocity) [=] m/h",
    y = "Removal effeciency"
  ) +
  theme_bw()
```



The overflow rate required to achieve **75% removal effeciency** is **2 m/h**. The corresponding τ if the depth of the basin is 4 m is

$$\tau = \frac{depth}{v_c} = \frac{4m}{2\frac{m}{h}} = 2h$$