

MOF-801 Adsorption Isotherm of Water

T.L. Hsieh

2023-01-18

view the dataset with table and plot

```
dataset <- read_excel("dataset.xlsx")
colnames(dataset)
```

```
## [1] "time_s" "hvp"      "uptake"
```

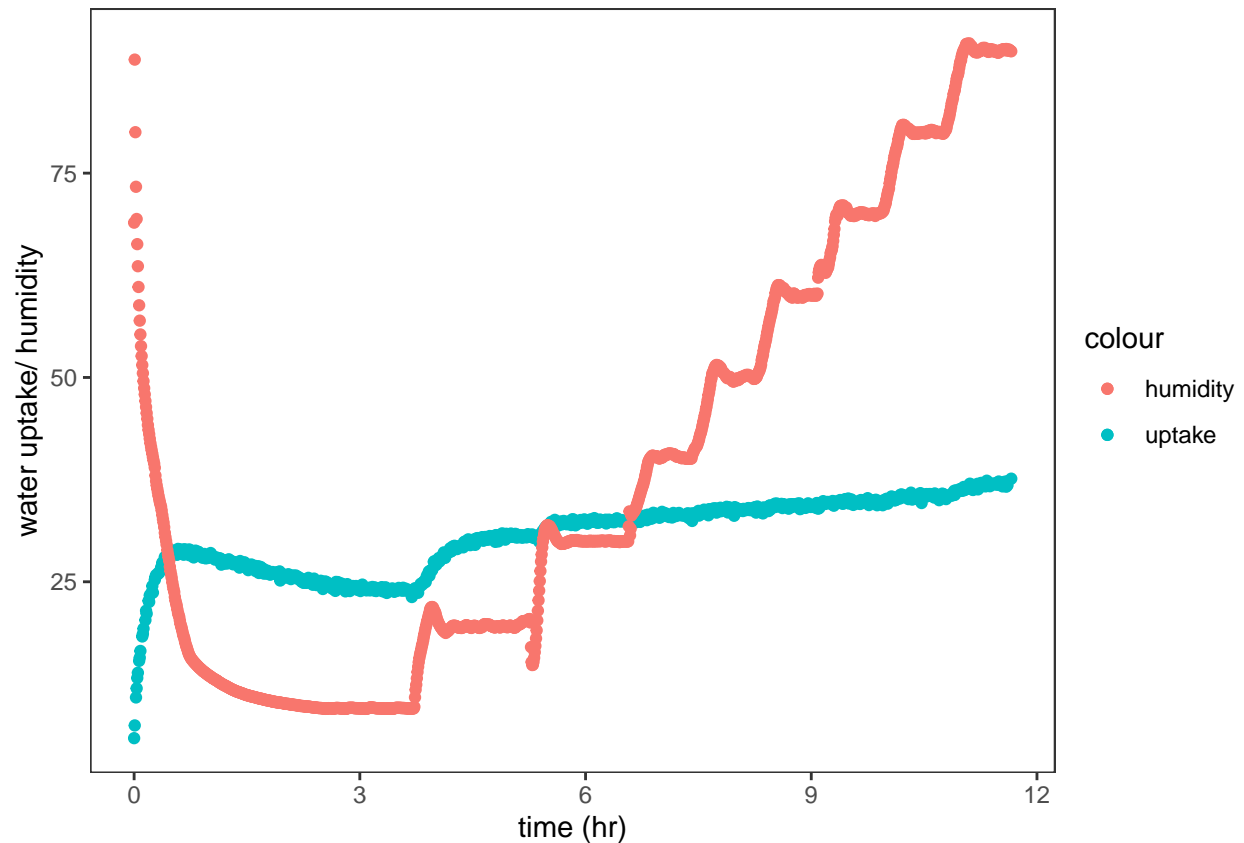
```
head(dataset)
```

```
## # A tibble: 6 x 3
##   time_s  hvp uptake
##   <dbl> <dbl> <dbl>
## 1     0  68.9  0.0587
## 2    30  88.9  0.0742
## 3    60  80.0    NA
## 4    90  73.3  0.108
## 5   120  69.4  0.120
## 6   150  66.3  0.132
```

```
p <- ggplot(data=dataset)+
  geom_point(mapping=aes(x=time_s/3600, y=uptake*100, color='uptake'))+
  geom_point(mapping=aes(x=time_s/3600, y=hvp, color='humidity'))+
  ylab("water uptake/ humidity")+
  xlab('time (hr)')+
  theme_test()
```

p

```
## Warning: Removed 448 rows containing missing values (‘geom_point()’).
```



```
ggplotly(p)
```

PhantomJS not found. You can install it with `webshot::install_phantomjs()`. If it is installed, please

Convert excel file to csv

```
df_excel <- read_excel("dataset.xlsx")
write.csv(df_excel, "dataset.csv")
```

Sometimes, R is having problem with .xlsx file when plotting, so it is better to convert xlsx to csv file, which will save you among of time in a long run.

Clean Data: reomve unwanted data points

```
df_csv <- read.csv("dataset.csv")
colnames(df_csv)
```

```
## [1] "X"      "time_s" "hpv"    "uptake"
```

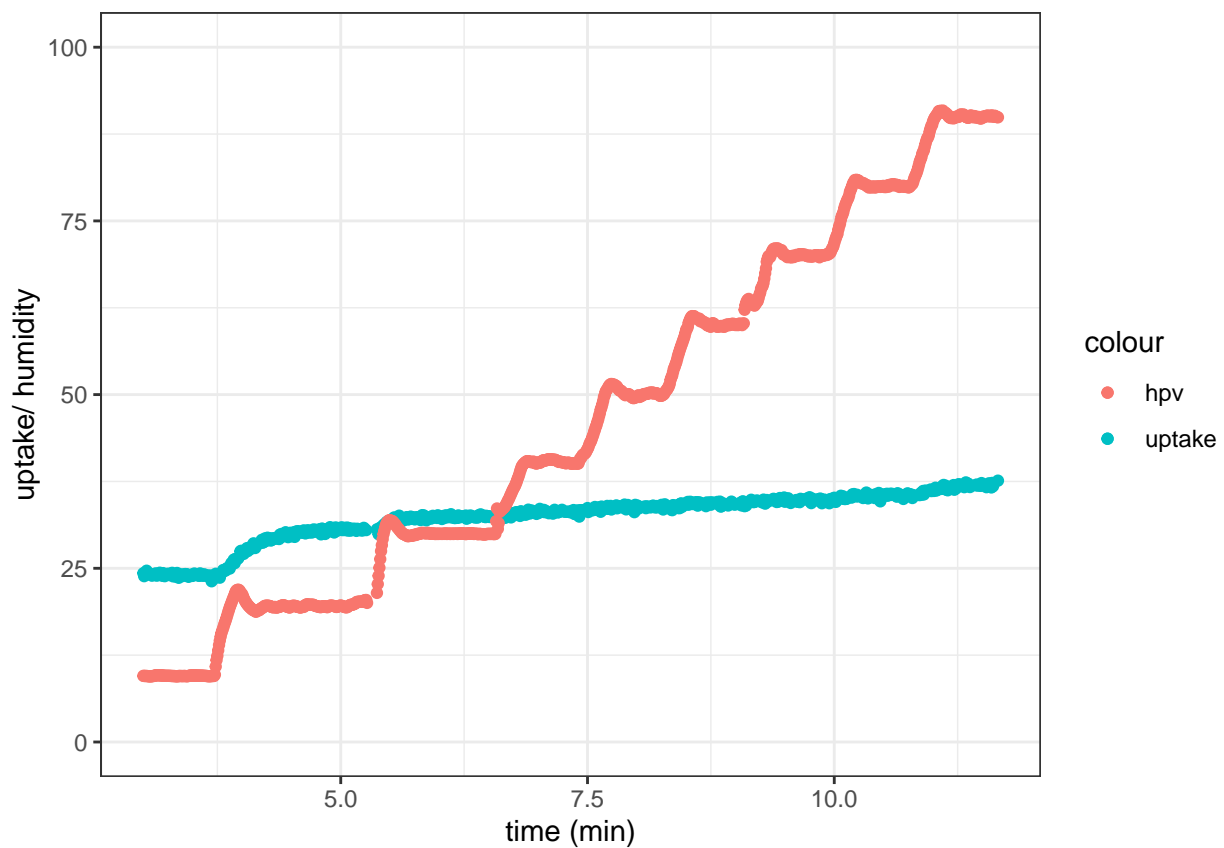
```
# data clean
df_clean <- df_csv %>% filter(time_s/3600>=3) %>%
  filter(!((time_s/3600 >= 5.267 & time_s/3600 <= 5.359) & (hvp >= 14.80 & hvp <= 20.28)))
```

Here we use pipe operator (`%>%`) to clean the data. First, we use filter function to extract the data with 'time_s/3600' greater than 3, which means only the data collected after 3 hours of measurement will be included. Second, we remove the points with the time within 5.267 to 5.359 hours and the hvp (humidity) between 14.80 to 20.28 because it will affect the accuracy of the finally result. Finally, we checked the cleaned data by visualizing it.

```
p1 <- ggplot(data=df_clean)+
  geom_point(mapping=aes(x=time_s/3600, y=uptake*100, color='uptake'))+
  geom_point(mapping=aes(x=time_s/3600, y=hvp, color='hvp'))+
  ylim(0, 100)+
  labs(x="time (min)", y="uptake/ humidity")+
  theme_bw()
```

```
p1
```

```
## Warning: Removed 347 rows containing missing values ('geom_point()').
```



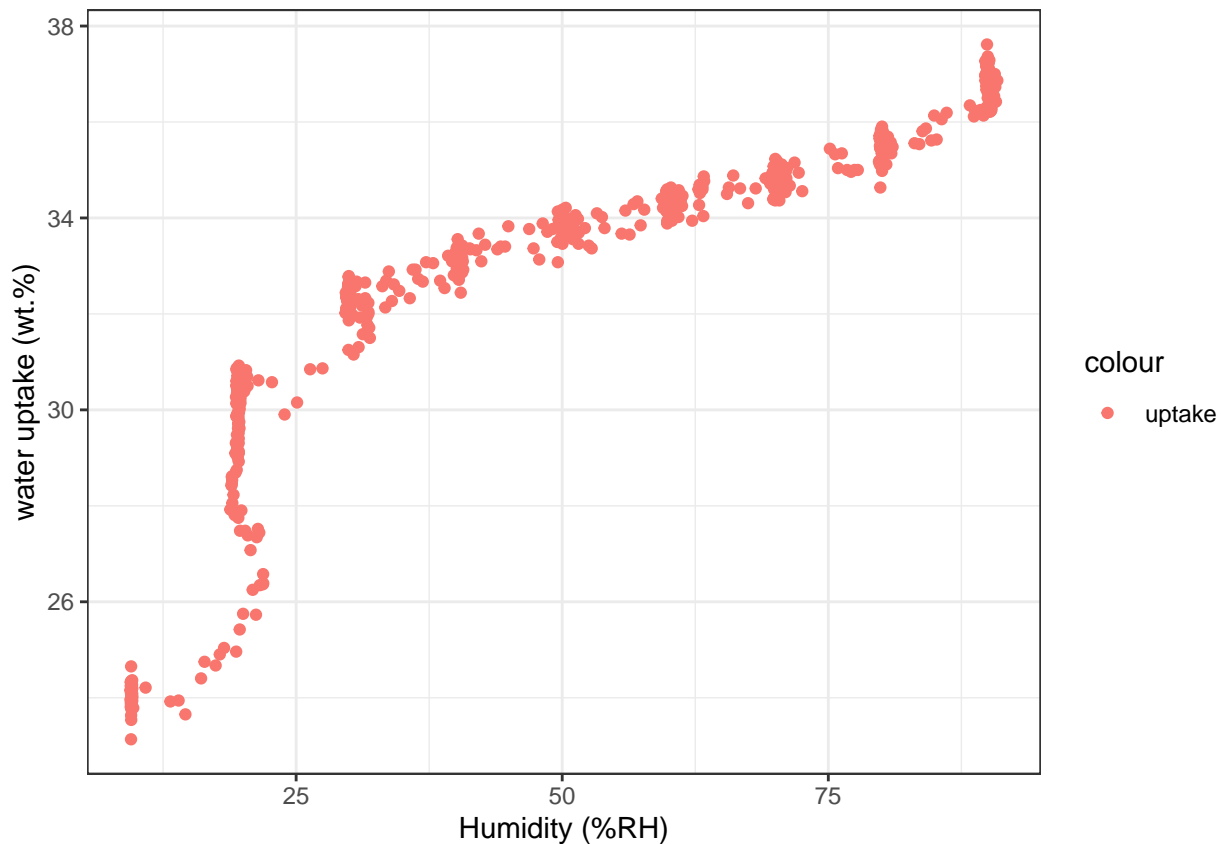
```
ggplotly(p1)
```

plot: water uptake vs humidity

```
p2 <- ggplot(data=df_clean)+  
  geom_point(mapping=aes(x=hpv, y=uptake*100, color='uptake'))+  
  labs(x="Humidity (%RH)", y="water uptake (wt.%)")+  
  theme_bw()
```

p2

```
## Warning: Removed 347 rows containing missing values ('geom_point()').
```



```
ggplotly(p2)
```

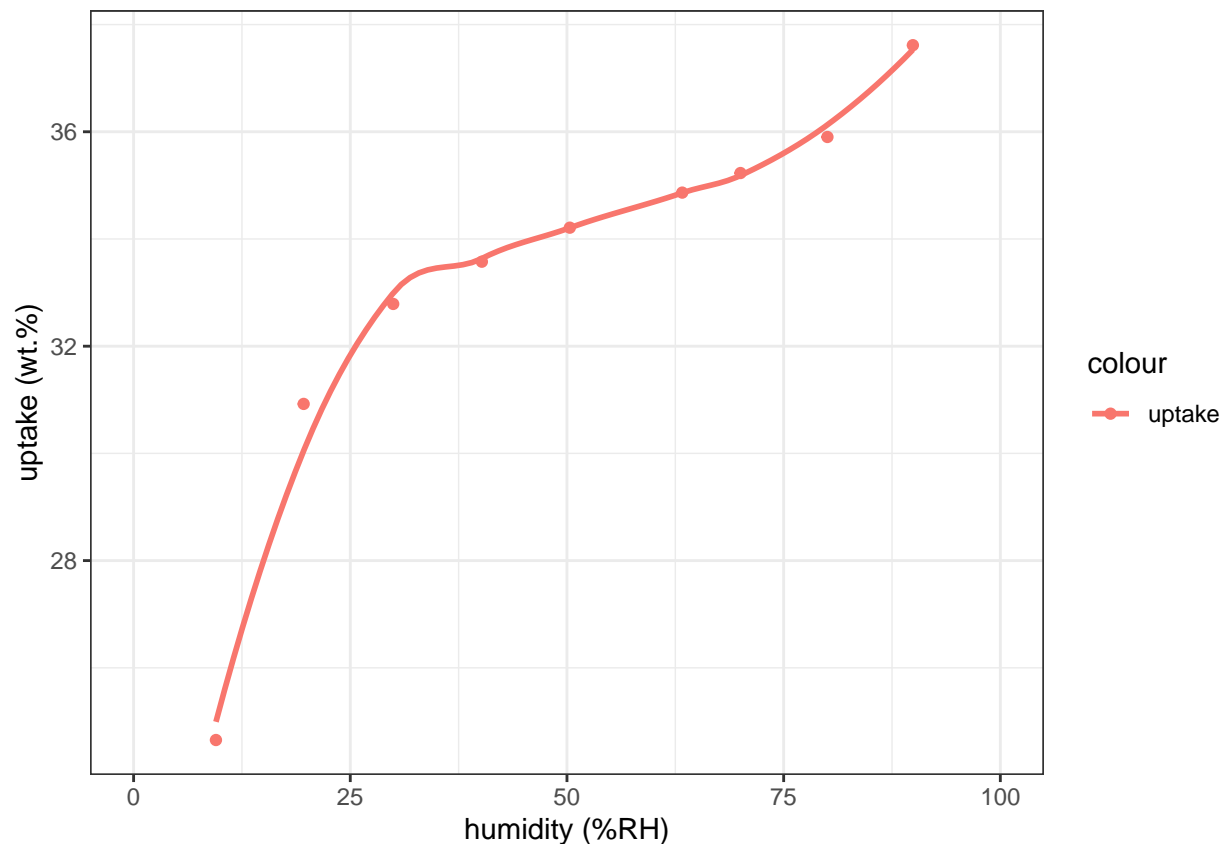
With the plot above we can easily type down the maximum water uptake of the test material at certain humidity. For example, the maximum water uptake at humidity around 10 %RH will be 24.653 wt.%. After collecting all the maximum uptake at the humidity we want to include, we can save the data set as .csv file and make the adsorption isotherm plot (shown below).

```
humidity = c(9.50, 19.62, 29.95, 40.18, 50.33,63.30,70.02,80.05,89.91)  
water_uptake = c(24.653,30.923, 32.788, 33.576, 34.211,34.865,35.230,35.903,37.615)  
  
point_pick = data.frame(humidity, water_uptake)  
write.csv(point_pick, 'point_pick.csv', row.names = FALSE)
```

```
p3 <- ggplot(data=point_pick, mapping=aes(x=humidity, y=water_uptake, color='uptake'))+
  geom_point()+
  geom_smooth(se=FALSE)+
  labs(x="humidity (%RH)", y="uptake (wt.%)")+
  xlim(0,100)+
  theme_bw()
```

p3

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```



```
ggplotly(p3)
```

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```

Group the data by humidity

Another way to process the data is to group the data by humidity. First we use mutate function to add extra column named "hvpv_1" with data that round of hvpv to first decimal. Second we group the whole data by hvpv_1 and use summarise function to calculate the maximum and mean value of the water uptake.

```
df_group <- df_clean %>%
  mutate(hpv_1 = round(hpv,1)) %>%
  group_by(hpv_1) %>%
  summarise(uptake_max = max(uptake), uptake_mean = mean(uptake))
```

Plot with smooth line

Finally, we plot the df_group data by hpv_1 and uptake, and add smooth line to the plot (as shown below).

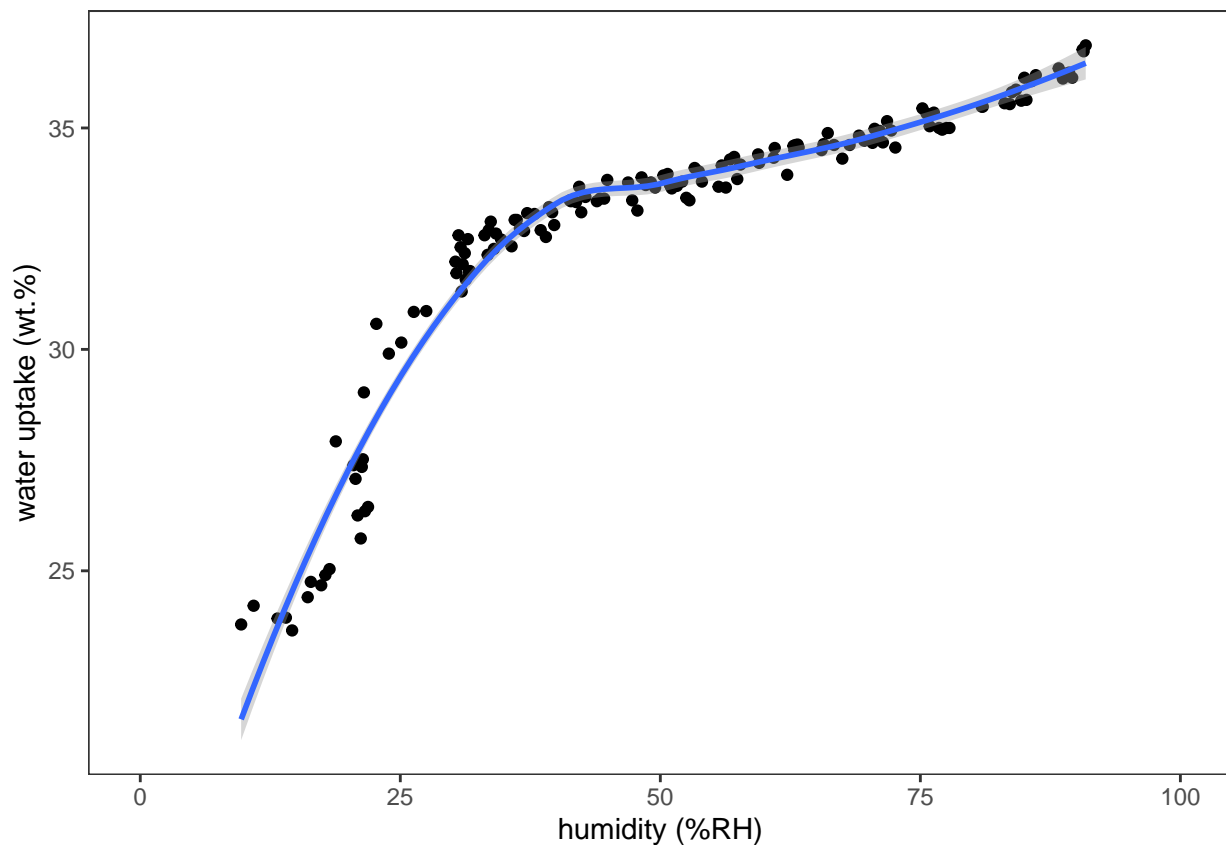
```
p4 <- ggplot(data=df_group)+
  geom_point(mapping=aes(x=hpv_1, y=uptake_mean*100))+
  geom_smooth(mapping=aes(x=hpv_1, y=uptake_mean*100))+
  ylab('water uptake (wt.%)')+
  xlab('humidity (%RH)')+
  xlim(0,100)+
  theme_test()
```

p4

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 155 rows containing non-finite values ('stat_smooth()').
```

```
## Warning: Removed 155 rows containing missing values ('geom_point()').
```



```
ggplotly(p4)
```

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 155 rows containing non-finite values ('stat_smooth()').
```

The figure above shows that the material adsorbs water as the increase of humidity.

save final result

```
df_dropna <- drop_na(df_group)
write.csv(df_group, "result_r.csv", row.names = FALSE)
```