IntroAnalysis

2025-01-11

Contents

1 Dependencies

The R code in this repository is provided in R Markdown files. This is a notebook format that is well supported by IDEs such as R Studio.

1.1 Packages

The animation depends on ffmpeg to stitch together frames into an animation. If you are installing R packages from source, you may also need to install a few packages. On MacOS, you should install:

```
brew install ffmpeg
brew install gdal
brew install units
```

1.2 Libraries

The following dependent libraries must be installed in your version of R to run the notebooks here.

```r
install.packages(c("tidyr", "dplyr", "ggplot2", "ggpattern", "arrow", "zoo",
                   "lubridate", "cowplot"))
```

2 Reading the Data

2.1 R

This provided parquet file can be read into R or Python with the relevant parquet libraries, or queried directly with Snowflake, DuckDB, or other tools.

We start by examing the first few rows of the data. The total dataset has 604,752 rows.

```
library(arrow)
data <- read_parquet("./hourly_normalized.parquet")
head(data)
```

```
## # A tibble: 6 x 4
##   USAGE_HOUR          REGION_NUM INSTANCE_TYPE NORM_USAGE
##   <dttm>                   <dbl> <chr>              <dbl>
## 1 2021-05-01 00:00:00          4 A                     61
## 2 2021-05-01 00:00:00          3 I                      9
## 3 2021-05-01 00:00:00          3 A                    193
## 4 2021-05-01 00:00:00          1 A                    186
## 5 2021-05-01 00:00:00          4 I                      9
## 6 2021-05-01 00:00:00          2 A                    140
```

```
dim(data)
```

```
## [1] 604752      4
```

2.2 Python

See IntroAnalysis-py.ipynb for a Jupyter notebook to read in and plot this dataset with Python.

2.3 DuckDB

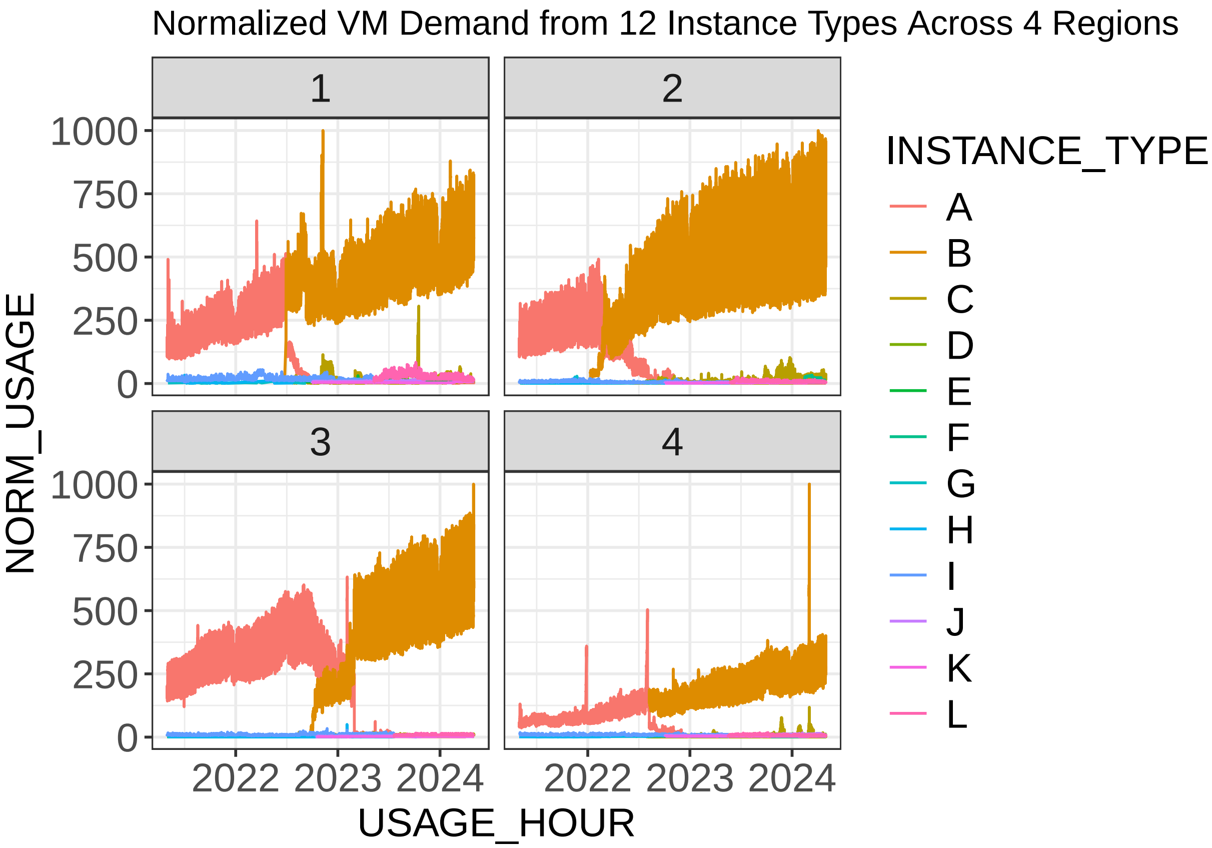
```
duckdb
D select count(*) from read_parquet('./hourly_normalized.parquet');
604752
```

3 Visualization

3.1 Complete Dataset

The following graph shows every data point in the dataset. There is a separate panel for each region, and then a separate colored line for each instance type within the region, showing the relative demand for different VM types across time.

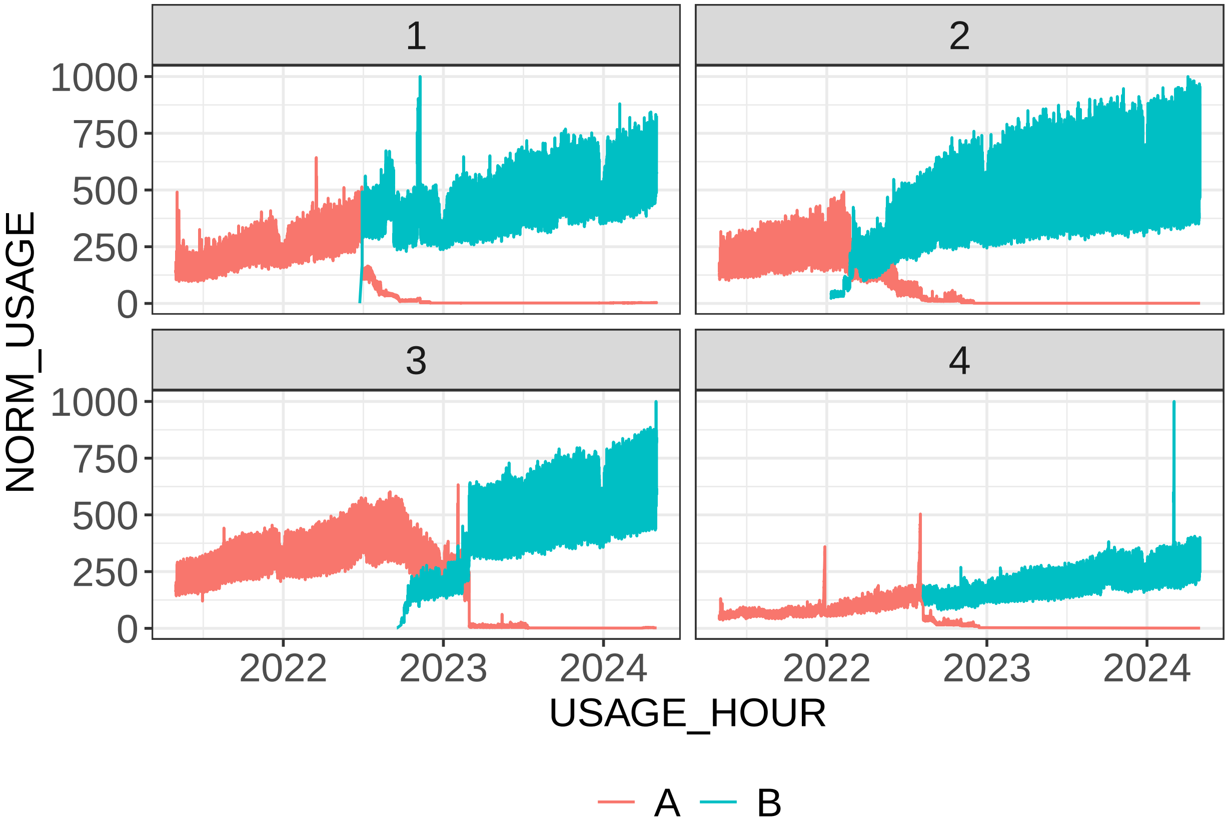
```
ggplot(data, aes(x=USAGE_HOUR, y=NORM_USAGE, color=INSTANCE_TYPE)) +
  geom_line() + theme_bw() +
  theme(axis.text=element_text(size=15), axis.title=element_text(size=15),
        legend.text = element_text(size=15),
        legend.title = element_text(size=15),
        strip.text=element_text(size=15)) +
  ggtitle("Normalized VM Demand from 12 Instance Types Across 4 Regions") +
  facet_wrap(. ~ REGION_NUM)
```



```
## pdf
##   2
```

3.2 Subset of 2 specific instance types

```
data.sub <- data %>% subset(INSTANCE_TYPE %in% c("A", "B"))
ggplot(data.sub, aes(x=USAGE_HOUR, y=NORM_USAGE, color=INSTANCE_TYPE)) +
  geom_line() + theme_bw() +
  theme(axis.text=element_text(size=15), axis.title=element_text(size=15),
        legend.text = element_text(size=15), legend.title = element_blank(),
        legend.position="bottom", strip.text=element_text(size=15)) +
  facet_wrap(. ~ REGION_NUM)
```

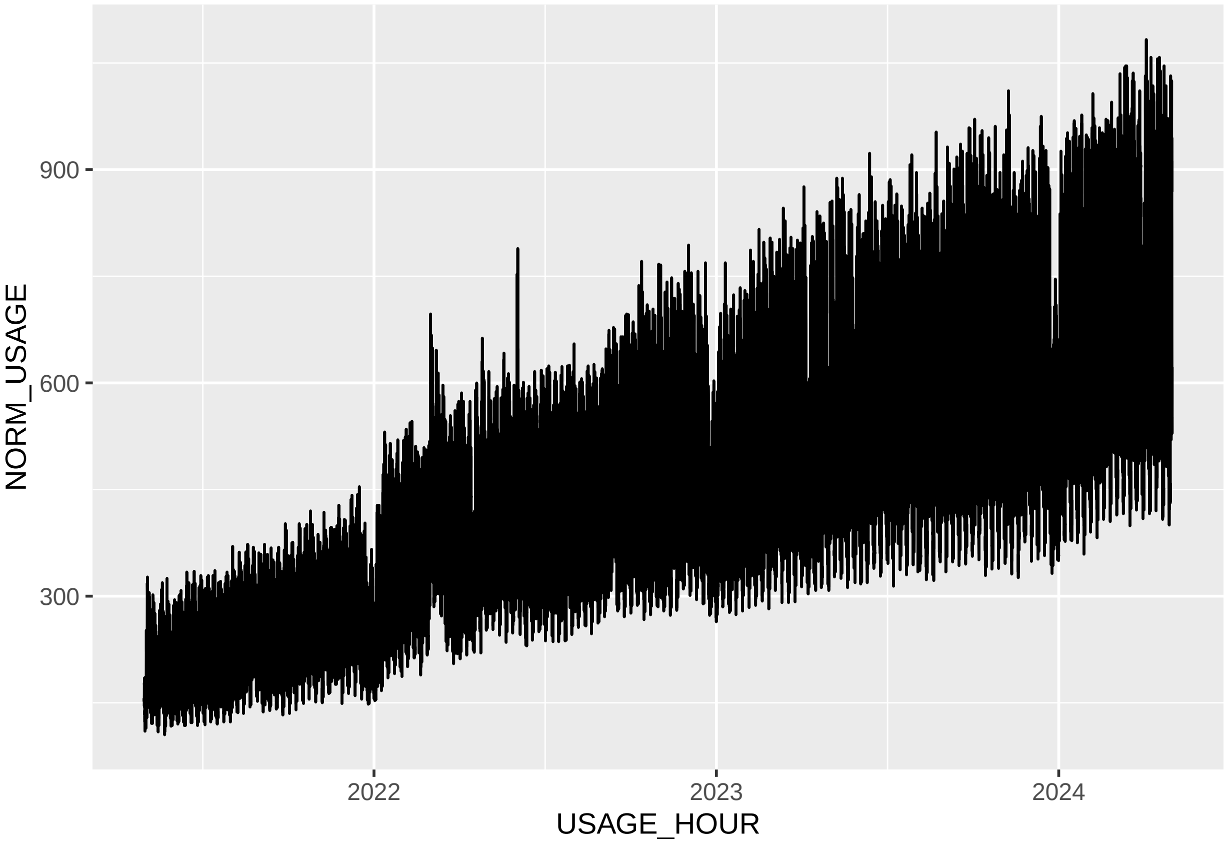


3.3 Subsetting to a specific region and then aggregating all VM types together for a regional demand timeseries

```
data.2 <- data %>% subset(REGION_NUM==2) %>% group_by(USAGE_HOUR) %>%
  summarise(across(NORM_USAGE, sum))
head(data.2)
```

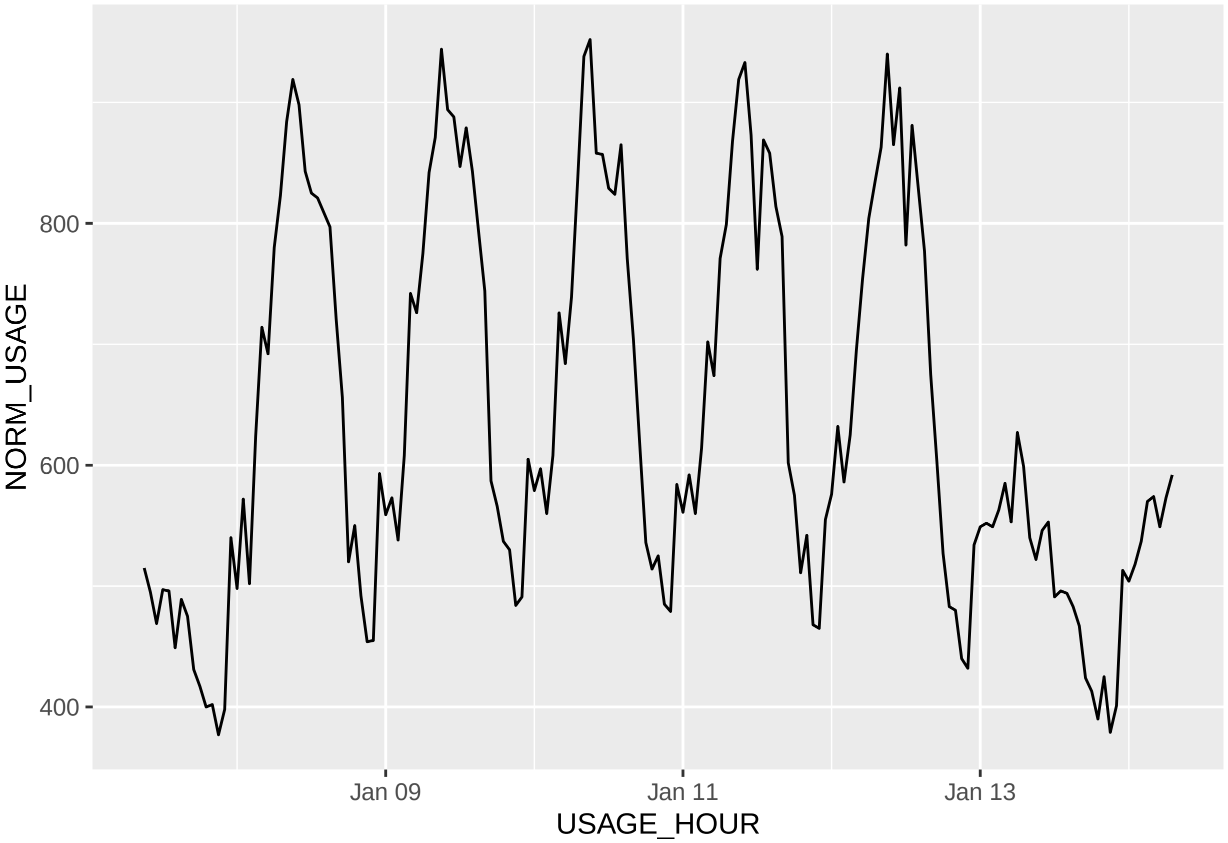
```
## # A tibble: 6 x 2
##   USAGE_HOUR          NORM_USAGE
##   <dttm>                   <dbl>
## 1 2021-05-01 00:00:00        145
## 2 2021-05-01 01:00:00        156
## 3 2021-05-01 02:00:00        152
## 4 2021-05-01 03:00:00        164
## 5 2021-05-01 04:00:00        178
## 6 2021-05-01 05:00:00        176
```

```
ggplot(data.2, aes(x=USAGE_HOUR, y=NORM_USAGE)) + geom_line()
```



3.4 Subset of a week to examine weekly pattern

```
data.sub <- data.2 %>%
  subset(USAGE_HOUR > "2024-01-07") %>%
  subset(USAGE_HOUR < "2024-01-14")
ggplot(data.sub, aes(x=USAGE_HOUR, y=NORM_USAGE)) + geom_line()
```



```
data.sub
```

```
## # A tibble: 167 x 2
##    USAGE_HOUR          NORM_USAGE
##    <dttm>                   <dbl>
##  1 2024-01-07 09:00:00        515
##  2 2024-01-07 10:00:00        495
##  3 2024-01-07 11:00:00        469
##  4 2024-01-07 12:00:00        497
##  5 2024-01-07 13:00:00        496
##  6 2024-01-07 14:00:00        449
##  7 2024-01-07 15:00:00        489
##  8 2024-01-07 16:00:00        475
##  9 2024-01-07 17:00:00        431
## 10 2024-01-07 18:00:00        417
## # i 157 more rows
```