Data Analytics & Engineering Team Manual

Nikita Voevodin: Senior Data Engineer

2021-11-29

Contents

Chapter 1

Scope

The New York City Taxi and Limousine Commission (TLC) is the City agency responsible for regulating for-hire transportation in New York City, including taxis, street hail liveries, high-volume for-hire services such as Uber and Lyft, black cars, luxury limousines, livery vehicles, commuter vans, and paratransit vehicles. The TLC licenses about 175,000 drivers, 115,000 vehicles, and 1,000 businesses, which together transport more than a million passengers a day, making TLC the most active for-hire transportation regulatory agency in the world with oversight of a key component of the City's transportation network. With the introduction of new apps and technologies, TLC is on the front lines of a rapidly changing mobility landscape and our innovative efforts–whether regulating driver pay, ensuring wheelchair accessibility, working to eliminate traffic fatalities, or preventing discriminatory service–often serve as a model for other cities.

The purpose of this document is to document practices, procedures and processes of analysis and work at the Taxi and Limousine Commission (TLC) with a focus on the Analytics Unit. This will be an evolving document meant to be shared with current and incoming staff, in order to maintain and capture all knowledge relevant to completing work at the TLC. In addition to laying the groundwork for standards, it will serve as a living document of the vision and strategy employed by the analytics team to serve the constituency with data driven decisions and support for policy research. The vision for the analytics team is to create a highly effective rapid-prototyping research element within TLC that will serve to do the following (examples provided below each point):

- **Provide policy research support**
  - Medallion Relief Program
  - Black Car and Livery Task Force and Report

  - Regulatory Review
  - Battery Electric Vehicle Pilot Program
  - Driver income study analytics support
  - Vehicle retirement adjustment

- **Maintain automated metrics and KPI's for rapid access internally and externally**
  - Respond to internal and external data requests
  - Open data support

- **Rapid prototype applications and analytical processes**
  - Testing new technologies for integration with IT & the taxi industry
  - Maintaing and imporving existing solutions like TLC Data Hub
  - Creating new tools for internal and external users

- **Modernize data infrastructure**
  - Working towards speeding up data processes with technologies like SQL Server Datawarehousing/Apache Spark

- **Engage with the public**
  - Publish innovative research on For-Hire industry
  - Partnering with academia and think thanks

Chapter 2

The Team

The analytics team in its current form consists of the Senior Data Engineer (Unit Head), a Data Engineer, a Data Analyst, and a College Aide. The team's primary focus is:

1. **Production of KPI's & metrics relevant to the industry**
2. **Rapid prototyping of algorithms and analytical tools**
3. **Data exploration and analysis**

The chain of command is as follows:

AC Data & Tech -> Sr.Data Engineer -> Data Engineer, Data Analyst, College Aide

TBD -> Nikita Voevodin -> TBD, TBD, Phillip Wong

When the Sr. Data Engineer is not present, the TBD will be in charge. All employees ultimately answer to the Assistant Commissioner of Data and technology. The current team members are:

Table 2.1: Team

Name	Title
TBD	Assistant Commissioner of Data & Technology
Nikita Voevodin	Sr.Data Engineer
TBD	Data Engineer
TBD	Data Analyst
TBD	Data Analyst
Phillip Wong	College Aide

Chapter 3

Workflow

3.1 Establishing a Data Science Environment

Each analyst operates in their own way however the following setups should be followed in order to leverage collaboration across the team:

- **Install R, Rstudio & adjacent tools**
  - R Base
  - Rstudio
- **Install Python 3.xxx**
  - Anaconda
  - Set proxies
- **SQL Server Management Studio**
- **ArcGIS or QGIS (if needed)**
- **Git**
- **NPM**

3.2 Professional Development

Each analyst has tools available to them for improving their data skillset:

- **DataCamp**

  - TLC maintains a subscription to datacamp for R, Python and more; for membership access speak to your supervisor

- **FreeCodeCamp**
  - An excellent free course for learning html, css and javascript
- **StackOverflow**
  - Is your friend if you are stuck
- **YouTube**
  - Most things you can learn there
- **R, Not the best practices**
  - 90% of the R code related stuff that you will be doing at TLC is covered there

3.3 Metrics, Analytics & Automation

Important: Moving forward we will focus on automating our work as best as possible. We already do for the most part. The general process:

Create a script -> Output a result -> Automate the script to run on schedule -> document (the what, the how, the when, and the why)

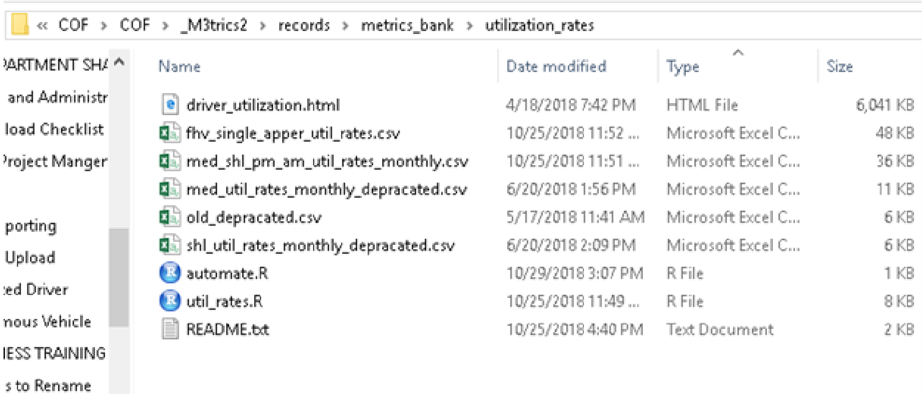
Note: Automation is a judgement call and should be considered the default. If things are ad hoc or to custom based to automate then note that in the documentation. The rule of thumb is: automate when there is a temporal element since this generally means someone will ask for this again.

All final reports and analyses should go under the directory below unless otherwise directed:

I:/COF/COF/*DA&E*/your_name…

This folder holds in it other subdirectories which various output files like images, excel documents and other relevant data. The general structure analysts on the analytics team should follow are:

- Unless otherwise directed, create a folder in your folder and label it with topic words of the project you are working on
- In that root folder insert your main script, cache any data files in subdirectories, and provide a documentation file. Below is an example:



Name	Date modified	Type	Size
driver_utilization.html	4/18/2018 7:42 PM	HTML File	6,041 KB
fhv_single_apper_util_rates.csv	10/25/2018 11:52 ...	Microsoft Excel C...	48 KB
med_shl_pm_am_util_rates_monthly.csv	10/25/2018 11:51 ...	Microsoft Excel C...	36 KB
med_util_rates_monthly_depracated.csv	6/20/2018 1:56 PM	Microsoft Excel C...	11 KB
old_depracated.csv	5/17/2018 11:41 AM	Microsoft Excel C...	6 KB
shl_util_rates_monthly_depracated.csv	6/20/2018 2:09 PM	Microsoft Excel C...	6 KB
automate.R	10/29/2018 3:07 PM	R File	1 KB
util_rates.R	10/25/2018 11:49 ...	R File	8 KB
README.txt	10/25/2018 4:40 PM	Text Document	2 KB

Figure 3.1: Folder Example

Note that in the folder we have output files, the main script, and an html report which is meant to help aid in presenting the work. The automate script leverages the taskcheduleR package in R to make the script run every month, this way utilization rates for medallions and shls are updated automatically.

3.4 Dashboards & Apps

Currently we are working on standardizing our analyses as best as possible in order to allow for quick servicing of both routine and some ad-hoc requests. As tools become more streamlined we will be able to expand with internal dashboarding tools; at the moment limitation in licensing tools and opensource software acceptance has hindered our ability to use these. Tools which we currently use for dashboarding are:

- **Dashboards**
  - Shiny R
  - PowerBI
- **Apps**
  - Shiny R
  - ReactJS
  - React Native

3.5 Passwords, Usernames, Accounts

TBD

Chapter 4

Data Connections & Access

4.1 SQL Server

Our primary data source is the sql server which holds a myriad of data bases each with their own sets of tables accessible through different tools. Currently the teams lean heavily on 3 of the following tools for accessing data from the server:

- **Sql server management studio**
- **R**
- **Python**

4.2 Pertinent IT Databases

There area series of databases which we use to collect data. Access to these servers is restricted by your windows profile and a ticket with IT is required to access the database, but talk to your supervisor first to see if there are accessible points already in place before approaching IT. below are some credentials provided for a few databases:

Table 4.1: ODBC Connections

Database	Driver	Server	Cr
DataWarehouse	odbc driver 17 for sql server	TLCBDWH	rec
Azure_Trip_Data	odbc driver 17 for sql server	tlcsqlmi01.fa986d691ca7.database.windows.net	rec
APPLUS	sql server	10.224.244.114	rec

TLC_Policy_Programs_Dev	odbc driver 17 for sql server	msdwvd-tlctxy01.csc.nycnet

4.3 Open Data Database

Before we submit data to the city Open Data portal, we store it in the Open Data sql server. If you ever need access to it, contact Nikita Voevodin or Maxim Smolyaninov from IT.

4.4 Open Data

City agencies that work with data are often required to post their data to the city open data portal. It can be very useful to know how to pull data from there. Even for our own data, because different departments publish different datasets, which might not be shared between departments. You can either download the whole datasets as csv, json, or whatever, or you can use their API. They have useful code snippets of how to work with their data in different programming languages. The first thing that you should do though is set up an account and get an api key. Then, you can do somethig like:

- **R:**

```
library(RSocrata)

date <- Sys.Date()

test <- read.socrata(
  paste0("https://data.cityofnewyork.us/resource/rhe8-mgbb.json?last_updated_date=",da
  app_token = "yourtokenhere",
  email     = "yourcreds",
  password  = "yourcreds"
)
```

- **Python:**

```
from sodapy import Socrata;
import pandas as pd;

client = Socrata("data.cityofnewyork.us",
                 "token",
                 username="yourcreds",
                 password="yourcreds")
```

```
today = date.today()

results = client.get("rhe8-mgbb", limit = 20000, last_updated_date=today)

# Convert to pandas DataFrame
results_df = pd.DataFrame.from_records(results)
```

4.5 Proxy Settings

At some point you might encounter a proxy problem. That is basically a firewall blocking certain connections from outside. Unfortunately, everything that has to do with installing python or javascript packages is considered undesirable by our firewall and ultimately will be blocked. That is a very annoying issue to deal with for anybody. Fortunately, there is a solution. For most tasks, adding the following proxy settings will fix the problem.

Open Anaconda -> file -> preferences -> configure Conda. Paste the following in there (use your username):

```
channels:
  - defaults
proxy_servers:
  http: http://csc\yourusername@10.155.126.15:8080
  https: http://csc\yourusername@10.155.126.15:8080

ssl_verify: false
```

Here are the settings for other languages and systems:

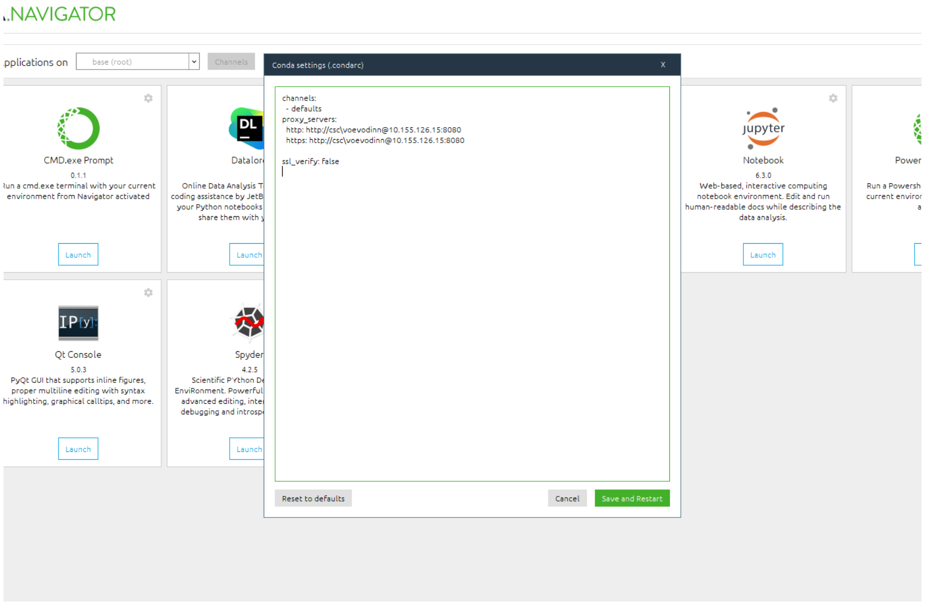
- **Linux**

```
#Use the following syntax to configure the proxy for http, https and ftp traffic on the Linux
#command line :

# export http_proxy="http://bcpxy.nycnet:8080"
# export https_proxy="https://bcpxy.nycnet:8080"
# export ftp_proxy="http://bcpxy.nycnet:8080"

--------------------------------------------------------------------------------
#Use the following syntax if the proxy server requires authentication :

# export http_proxy="http://user:password@bcpxy.nycnet:8080"
```



```
channels:
  - defaults
proxy_servers:
  http: http://csc\voevodinn@10.155.126.15:8080
  https: http://csc\voevodinn@10.155.126.15:8080

ssl_verify: false
```

Figure 4.1: Proxy Settings

```
# export https_proxy="https://user:password@bcpxy.nycnet:8080"
# export ftp_proxy="http://user:password@bcpxy.nycnet:8080"

----------------------------------------------------------------------------------------
#Using your email address username%40agency.nyc.gov

# export http_proxy="http://username%40agency.nyc.gov:<password>@bcpxy.nycnet:8080"
# export https_proxy=https://username%40agency.nyc.gov:<password>@bcpxy.nycnet:8080
# export ftp_proxy=http://username%40agency.nyc.gov:<password>@bcpxy.nycnet:8080
```

- **NPM**

```
# npm config set proxy http://bcpxy.nycnet:8080
# npm config set https-proxy http://bcpxy.nycnet:8080
```

If none of this helps, contact either Nikita Voevodin or Jordan Mamet from IT.

Chapter 5

Tasks & CheckUps

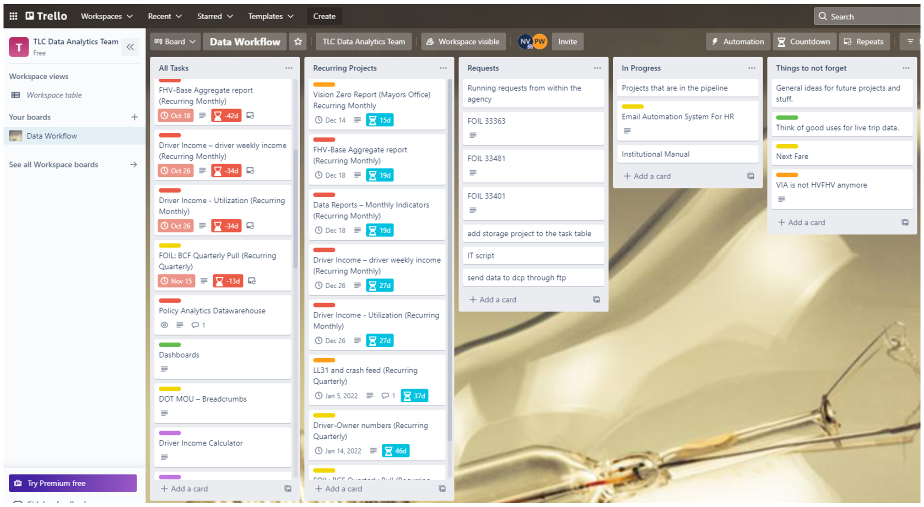
There are 4 types of tasks:

- **Major project where the whole team works together**
  - These tasks usually have scopes, timelines, and are well documented and articulated.
- **Pull requests and adhoc things**
  - You supervisor would usually assign and oversee those. They are usually data requests from the leadership, other departments, or outside of the agency. They are usually less documented and deadlines depend on where it came from.
- **Requests that went around your supervisor and straight to you**
  - If it is from a supervisor of your supervisor, just do it. If it is from other department, prioritize it based on your availability and the nature of the request. If it is from outside of the agency, definitely notify your supervisor. In most cases though, it is up to you as long as the priority tasks are getting done.
- **Your own initiatives**
  - You absolutely encouraged to have initiatives of your own. Especially, if they can benefit the agency and your growth. Most of the valuable agency projects start as small initiatives. Your supervisor will support your initiative in 99% cases.

We have a Trackit system in place. Some other departments will require you to officially submit your requests through that. You can do that too, but I would advice against that as it creates an impressions that you are unwilling to help or cooperate. It is your call however.

5.1 Trello

There is a bunch of ways to track your own projects and you are free to use what works for you. I will try to stick with Trello task management software to assign and track tasks. It is free and convenient. Give it a shot.



All Tasks

- FHV-Base Aggregate report (Recurring Monthly) — Oct 18, -42d
- Driver Income – driver weekly income (Recurring Monthly) — Oct 26, -34d
- Driver Income - Utilization (Recurring Monthly) — Oct 26, -34d
- FOIL: BCF Quarterly Pull (Recurring Quarterly) — Nov 15, -13d
- Policy Analytics Datawarehouse — 1
- Dashboards
- DOT MOU – Breadcrumbs
- Driver Income Calculator

Recurring Projects

- Vision Zero Report (Mayors Office) Recurring Monthly — Dec 14, 15d
- FHV-Base Aggregate report (Recurring Monthly) — Dec 18, 19d
- Data Reports – Monthly Indicators (Recurring Monthly) — Dec 18, 19d
- Driver Income – driver weekly income (Recurring Monthly) — Dec 26, 27d
- Driver Income - Utilization (Recurring Monthly) — Dec 26, 27d
- LL31 and crash feed (Recurring Quarterly) — Jan 5, 2022, 1, 37d
- Driver-Owner numbers (Recurring Quarterly) — Jan 14, 2022, 46d

Requests

- Running requests from within the agency
- FOIL 33363
- FOIL 33481
- FOIL 33401
- add storage project to the task table
- IT script
- send data to dcp through ftp

In Progress

- Projects that are in the pipeline
- Email Automation System For HR
- Institutional Manual

Things to not forget

- General ideas for future projects and stuff.
- Think of good uses for live trip data.
- Next Fare
- VIA is not HVFHV anymore

Figure 5.1: Trello ex

5.2 Weekly Meetings

Your supervisor will place a weekly meeting on your calendar. During that (informal) meeting you can update them on how things are going, your ideas, concerns, etc. You can obviously reach out to your supervisor for guidance at any point on any other day.

5.3 Evaluations

Performance evaluations will be given once every six months, once in January and again in June. Whichever is closest to the start time of the employee, so long as the time is not three months or less. This will give an employee a minimum of three months to get settled in so a baseline can be determined.

Chapter 6

Other Teams & Contacts

The TLC licenses about 175,000 drivers, 115,000 vehicles, and 1,000 businesses, which together transport more than a million passengers a day, making TLC the most active for-hire transportation regulatory agency in the world with oversight of a key component of the City's transportation network. To do all that we obviously need people and space. We have 3 main offices in NYC. Below, I would like to list the teams and contacts that might be relevant to our work here. It might not be complete as i do not know everybody at TLC, so if you are reading this and thinking that you should be here, email me at voevodinn with relevant info.

- **Beaver Street (Head Office)**
- **LIC (Licensing and Court)**
- **Queens (Inspection and Enforcement)**

6.1 Beaver

TLC head office. Most of the decision-making happens here. TLC Commissioner's office, IT, HR, PR, Legal, Policy, Education, External Affairs, Programs and other admin staff are all located at Beaver. Here are some useful contacts for you (always in works):

6.1.1 Data Analytics & Engineering

- Nikita Voevodin
  - Senior Data Engineer (Unit Head)

  - email voevodinn@
  - ext 1195
  - ask me about: trip data, our tech, programming, data projects

6.1.2 Policy

- James DiGiovanni
  - Executive Director
  - email digiovannij@
  - ext ...
  - ask me about: ...
- Ted Metz
  - Policy Analyst
  - email metzt@
  - ext ...
  - ask me about: ...

6.1.3 Programs

6.1.4 IT (DATA)

6.1.5 Legal

6.1.6 PR

6.1.7 External Affairs

6.1.8 HR

6.2 LIC (Licensing and Courts)

This office processes drivers. Our prosecution and another analytics teams are located there. They deal with with a lot of interesting questions and data and hold a lot of institutional knowledge.

6.2.1 Data Analytics

- Adrian Chamorro
  - Data Engineer

  - email chamorroa@
  - ext ...
  - ask me about: ...

6.2.2 Prosecution

- Serge Router
  - Prosecution Data Support Unit
  - email Routers@
  - ext ...
  - ask me about: ...

6.3 Queens (Inspection and Enforcement)

This office inspects and processes vehicles. Our police and another analytics teams are located there. Inspection data is super useful and we at Beave have little knowledge of what goes into it.

6.3.1 Data Analytics

Chapter 7

Most Used Tables

7.1 Traditional Data

Within the databases that we covered in one of the previous sections are a litany of tables which we access for various purposes. Below is a list of the popular tables we reference with a short description of what they are for. If you have any questions talk to your supervisor.

- **FHVHV_TripRecord**
  - Trip record table for High Volume (UBER, Lyft, VIA prior to Sep 2021) trips after 2019-01, each row represents a trip.
  - Database: Azure_Trip_Data
  - Note: Dont ever pull the whole table, it will crash your PC. Use 'datetimeid' for dates - it is indexed.
  - Sample pull:

```
SELECT top 100 *
  FROM [TPEP_AZURE].[TPEPDW].[dbo].[FHVHV_TripRecord]
  where datetimeid >= 2021080100 and datetimeid < 2021110100
```

- **FHV_Prd_TripRecord**
  - Trip record table for all fhvs (high volume and non high volume before 2019-02) and just traditional fhvs (after 2019-01). Each row represents a trip.
  - Database: Azure_Trip_Data
  - Note: Dont ever pull the whole table, it will crash your PC. Use 'datetimeid' for dates - it is indexed.

  - Sample pull:

```
SELECT top 100 *
  FROM [TPEP_AZURE].[TPEPDW].[dbo].[FHV_Prd_TripRecord]
  where datetimeid >= 2021080100 and datetimeid < 2021110100
```

- **vw_FHVALL_Triprecord**
  - Combined view of the 2 tables above combined. Each row represents a trip.
  - Database: Azure_Trip_Data
  - Note: Not every column that is present in the FHVHV_TripRecord present in the FHV_Prd_TripRecord. For example, anything that has to do with financial information.
  - Sample pull:

```
SELECT top 100 *
  FROM [TPEP_AZURE].[TPEPDW].[dbo].[vw_FHVALL_Triprecord]
  where datetimeid >= 2021080100 and datetimeid < 2021110100
```

- **Tpep2_triprecord**
  - Trip record table for medallion (yellow) trips after 2010, each row represents a yellow cab trip .
  - Database: Azure_Trip_Data
  - Note: Dont ever pull the whole table, it will crash your PC. Use ‘datetimeid’ for dates - it is indexed.
  - Sample pull:

```
SELECT top 100 *
  FROM [TPEP_AZURE].[TPEPDW].[dbo].[Tpep2_triprecord]
  where datetimeid >= 2021080100 and datetimeid < 2021110100
```

- **Lpep2_triprecord**
  - Trip record table for SHL (green) trips after 2010, each row represents a yellow cab trip .
  - Database: Azure_Trip_Data
  - Note: This table’s tructure is very similar to Tpep2_triprecord, but it has much fewer records. Use ‘datetimeid’ for dates - it is indexed.
  - Sample pull:

```
SELECT top 100 *
  FROM [TPEP_AZURE].[TPEPDW].[dbo].[Lpep2_triprecord]
  where datetimeid >= 2021080100 and datetimeid < 2021110100
```

- **DimLocation**
  - Super important table if you are doing spacial analysis.
  - Database: Azure_Trip_Data
  - Note: Useful when you are joining it to the trip records by the locationid column
  - Sample pull:

```
SELECT *
  FROM [TPEP_AZURE].[TPEPDW].[dbo].[DimLocation]
```

- **Fhv_base_list**
  - The list of bases. Bases are companies that dipatch trips. The table might not be super useful on its own, but it is super useful when you join it to trip tables to figure out an industry or a company name of a base that dispathced a trip.
  - Database: Azure_Trip_Data
  - Note: Not a big table. In the example below, look at the last 5 columns.
  - Sample pull:

```
select top 100 *
    FROM
     [TPEP_AZURE].[TPEPDW].[dbo].[FHV_Prd_TripRecord]
    AS TRIPS
    INNER JOIN [TPEP_AZURE].[TPEPDW].[dbo].[fhv_base_list] bases on
    TRIPS.[Dispatching_base_num] = bases.[LIC_NO]
```

- **Tlc_camis_entity**
  - A snapshot of all entities (all licensees including but not limited to drivers, vehicles, bases) and pertinent information like license application date, addresses etc.
  - Database: DataWarehouse
  - Note: Very important table. We have a very extensive printed documentation for it. Ask your supervisor for it.
  - Sample pull: All active drivers

```
SELECT entity_nam, rtrim(ltrim(fed_id)) as fed_id, rtrim(ltrim(lic_no)) as lic_no, lic_code, lic_
     from tlc_camis_entity
     where lic_code in ('HDR','CDR') and STAT_ENTITY_LIC IN ('002','009','010','RNA','ANL')
```

- **Tlc_plate**

  - A table holding all current and historical plate information for vehicles.
  - Database: DataWarehouse
  - Note: .
  - Sample pull: pull top 100

```
SELECT top 100 *
     from tlc_plate
```

7.2 TLC Datawarehouse

The Data Team, collaborating with IT, built out a data warehouse that automatically aggregates on a set schedule the most often requested data points, drastically increasing the speed with which data can be pulled and analyzed. Most tables in the warehouse update automatically and run on a set schedule. Standard workflow of creating a new table is as follows:

- Create a new table with some initial data straight from the SSMS, Python, or R.
- Create a Stored Procedure script. The goal of that script is to update the table that you created.
- Create a Job in the SSMS job scheduler. That job will run the Stored Procedure that you created in the step 2 on schedule that you specify.

Datawarehouse SSMS view:

Here is a connection example using R and Python. Note: you must have the ODBC connection set up the way shown in the section 4.2 of this manual.

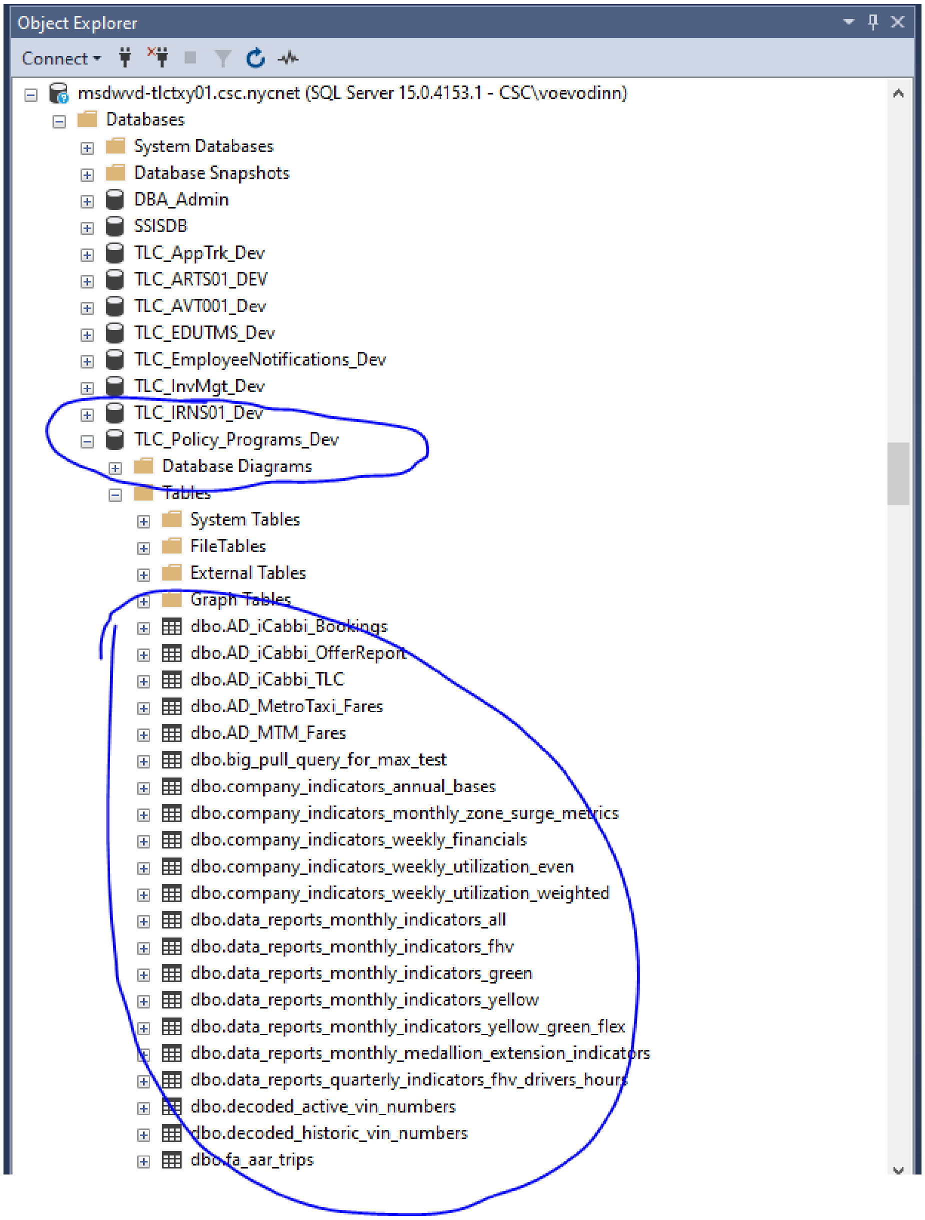
- **R:**

```
library(RODBC)

tp2 = odbcConnect("TLC_Policy_Programs_Dev", uid = "...")

test <- sqlQuery(tp2,
                 "SELECT *
                 FROM [TLC_Policy_Programs_Dev].[dbo].[high_volume_indicators_weekly_f
```

- **Python:**



```
msdwvd-tlctxy01.csc.nycnet (SQL Server 15.0.4153.1 - CSC\voevodinn)
  Databases
    System Databases
    Database Snapshots
    DBA_Admin
    SSISDB
    TLC_AppTrk_Dev
    TLC_ARTS01_DEV
    TLC_AVT001_Dev
    TLC_EDUTMS_Dev
    TLC_EmployeeNotifications_Dev
    TLC_InvMgt_Dev
    TLC_IRNS01_Dev
    TLC_Policy_Programs_Dev
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        dbo.AD_iCabbi_Bookings
        dbo.AD_iCabbi_OfferReport
        dbo.AD_iCabbi_TLC
        dbo.AD_MetroTaxi_Fares
        dbo.AD_MTM_Fares
        dbo.big_pull_query_for_max_test
        dbo.company_indicators_annual_bases
        dbo.company_indicators_monthly_zone_surge_metrics
        dbo.company_indicators_weekly_financials
        dbo.company_indicators_weekly_utilization_even
        dbo.company_indicators_weekly_utilization_weighted
        dbo.data_reports_monthly_indicators_all
        dbo.data_reports_monthly_indicators_fhv
        dbo.data_reports_monthly_indicators_green
        dbo.data_reports_monthly_indicators_yellow
        dbo.data_reports_monthly_indicators_yellow_green_flex
        dbo.data_reports_monthly_medallion_extension_indicators
        dbo.data_reports_quarterly_indicators_fhv_drivers_hours
        dbo.decoded_active_vin_numbers
        dbo.decoded_historic_vin_numbers
        dbo.fa_aar_trips
```

Figure 7.1: Datawarehouse ex

```
import pyodbc

params= urllib.parse.quote_plus("DRIVER={SQL Server};SERVER=msdwvd-tlctxy01.csc.nycnet
engine = create_engine("mssql+pyodbc:///?odbc_connect=%s" % params)


sql = '''
SELECT *
FROM [TLC_Policy_Programs_Dev].[dbo].[high_volume_indicators_weekly_financials]
    '''

test = pd.read_sql_query(sql, engine)
```

The data dictionaries for the majority of tables in the Datawarehouse are in here:

```
I:\COF\COF\_M3trics2\automation\data_dictionaries
```

There is also a standardization guide for creating tables and views in the warehouse. You can access it here:

```
I:\COF\COF\_DA&E_\Nikita\Supporting_docs
```

7.3 Tables

There are many useful tables in the Datawarehouse. I recommend going through the documentation folder to get accustomed with some of them. I would like to list top 5 most used tables in this document though:

- **industry__indicators__daily__trips**
    - This table goes back to 2014 (inclusive) for yellow and green, and to 2015 for fhvs. It contains trip counts aggregated by day, split by every industry.
    - Database: TLC__Policy__Programs__Dev
    - Note: This table will save you a ton of time.
    - Sample pull:

```
SELECT TOP (1000) [period_start]
      ,[period_end]
      ,[metric_day]
      ,[industry]
      ,[count_trips]
  FROM [TLC_Policy_Programs_Dev].[dbo].[industry_indicators_daily_trips]
```

- **data_reports_monthly_indicators_all**
  - These are a set of published metrics here that are updated every month and reviewed with the commissioner before updating. They cover a myriad of relevant metrics for certain industries we regulate.
  - Database: TLC_Policy_Programs_Dev
  - Note: Serves as a base for the monthly indicators that we publish to our website.
  - Sample pull:

```
SELECT TOP (1000) [Month_Year]
      ,[License_Class]
      ,[Trips_Per_Day]
      ,[Farebox_Per_Day]
      ,[Unique_Drivers]
      ,[Unique_Vehicles]
      ,[Vehicles_Per_Day]
      ,[Avg_Days_Vehicles_on_Road]
      ,[Avg_Hours_Per_Day_Per_Vehicle]
      ,[Avg_Days_Drivers_on_Road]
      ,[Avg_Hours_Per_Day_Per_Driver]
      ,[Avg_Minutes_Per_Trip]
      ,[Percent_of_Trips_Paid_with_Credit_Card]
      ,[Trips_Per_Day_Shared]
  FROM [TLC_Policy_Programs_Dev].[dbo].[data_reports_monthly_indicators_all]
```

- **high_volume_indicators_weekly_financials**
  - These are a set of metrics we created to track driver income on a Monday to Sunday weekly schedule.
  - Database: TLC_Policy_Programs_Dev
  - Note: Created in a python script they piggyback off utilization to come up with our best estimate on high volume driver income.
  - Sample pull:

```
SELECT TOP (1000) [date]
      ,[metric_week]
      ,[aggregate_pay]
      ,[aggregate_hours]
      ,[aggregate_hourly_pay]
      ,[median_total_pay]
      ,[median_logon_hours]
      ,[median_hourly_pay]
      ,[driver_count]
      ,[pay_per_driver]
```

```
  FROM [TLC_Policy_Programs_Dev].[dbo].[high_volume_indicators_weekly_financials]
  order by metric_week desc
```

- **industry_zone_indicators_monthly_pickups**
  - Count of pickups split by month, industry, and taxi zone (265).
  - Database: TLC_Policy_Programs_Dev
  - Note: This table will save you a ton of time.
  - Sample pull:

```
SELECT TOP (1000) [period_start]
      ,[period_end]
      ,[metric_month]
      ,[industry]
      ,[zone]
      ,[count_pickups]
      ,[count_pickups_shared]
      ,[count_pickups_ehail]
  FROM [TLC_Policy_Programs_Dev].[dbo].[industry_zone_indicators_monthly_pickups]
  order by [metric_month] desc
```

- **company_indicators_weekly_utilization_even**
  - Driver utilization is calculated and loaded into our policy dev server. It is currently run unweighted, meaning that app logon time which is the denominator in this calculation is evenly split for apps a driver is logged into simultaneously.
  - Database: TLC_Policy_Programs_Dev
  - Note: Note that every nth time a year we re-evaluate utilization publicly as per the law – legal can provide more assistance on the timeline as Ryan wrote the rules.
  - Sample pull:

```
SELECT TOP (1000) [period_start]
      ,[period_end]
      ,[metric_week]
      ,[company]
      ,[sum_cruising_seconds]
      ,[sum_passenger_seconds]
      ,[pct_utilization]
  FROM [TLC_Policy_Programs_Dev].[dbo].[company_indicators_weekly_utilization_even]
```

This section will be developed more in the future.

Chapter 8

Major Projects

Data team is responsible for a lot of interesing and essential projects. Some of these projects are very important to the day-to-day operations of the agency. Here is an overview of the top 5 most important data projects. This section is flexible and will be either expanded or shrunk as needed.

8.1 TLC Datawarehouse

- **What**

known as our policy dev server, this is a database provided by IT where we store aggregated tables and important metrics. Every 1-2 weeks numbers are updated to support policy.

- **Who**

built by the policy analytics team, it is maintained now by Nikita Voevodin with IT support from Maxim Smolyaninov.

- **Next steps**

new tables should be created for requests that are deemed repetitive and automatable.

- **Directories and sources**

```
I:\COF\COF\_M3trics2\automation
```

- **Catalog and data dictionaries here**

```
Data science reference: I:\COF\COF\_M3trics2\automation\data_dictionaries
```

- **Point**

Nikita Voevodin should oversee this. Note that this is a very large piece of the work we do – without this database we would have to redo requests constantly, slowing down work greatly. Almost every basic number fielded by public affairs, PR, senior management comes from some table that lives

- **IT Support**

Maxim Smolyaninov

8.2 Data Reports - Monthly Indicators

- **What** This report is published on open data and on our website and is reviewed with the commissioner every month. It includes a lot of relevant data from the traditional fhv bases (trip patterns, vehicle and driver counts etc)

Table published on open data with the following columns: Base License Number, Base Name, DBA, Year, Month, Month Name, Total Dispatched Trips, Total Dispatched Shared Trips, Unique Dispatched Vehicles.

- **Location:**

https://data.cityofnewyork.us/Transportation/FHV-Base-Aggregate-Report/2v9c-2k7f

- **Point** Nikita
- **Support**

IT/ Web: Konstantin Onishchenko, PR: Alan Fromberg, Rebecca Harshbarger

8.3 Driver Utilization data

- **What**

Driver utilization is calculated and loaded into our policy datawarehouse. It is currently run unweighted, meaning that app logon time which is the denominator in this calculation is evenly split for apps a driver is logged into simultaneously. Note that every nth time a year we re-evaluate utilization publicly as per the law – legal can provide more assistance on the timeline as Ryan wrote the rules.

- **Metrics:**

```
I:\COF\COF\_M3trics2\automation\data_dictionaries. File: "company_indicators_weekly_utilization_e
```

- **Point:**

Nikita

8.4 TLC Data Hub

- **What** TLC Data Hub offers users a new and convenient location to access and visualize taxi and for hire industry data. TLC Data Hub uses public data available on Open Data and the TLC website and does not use, track or display any private information of the drivers or companies. The Hub currently consists of two dashboards. The 'Trip Viz' dashboard allows the public to run queries on TLC-collected trip data while the 'Industry metrics' dashboard provides standard visualizations of monthly industry trends.

- **Address**

https://tlcanalytics.shinyapps.io/dash_test/

- **Point**

Nikita

8.5 Raw Trip Records publishing

- **What** Every 6 months TLC aims to release the previous 6 months of raw trip record data publicly both on our website and Open Data.
- **Process:** Ticket to Lana to create monthly files

Review with Chair

Send Konstantin links for him to stage (they will have predictable names based on month and industry)

Ticket to Lana to load files to AWS

Send links to Alex Finkel at DoITT to post to Open Data

- **User Guide:** https://www1.nyc.gov/assets/tlc/downloads/pdf/trip_record_user_guide.pdf
- **Yellow Dictionary:** https://www1.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf
- **Green Dictionary:** https://www1.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_green.pdf
- **FHV Dictionary:** https://www1.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_fhv.pdf
- **High Volume Dictionary:** https://www1.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_hvfhs.pdf
- **Point**

Nikita

- **Support**

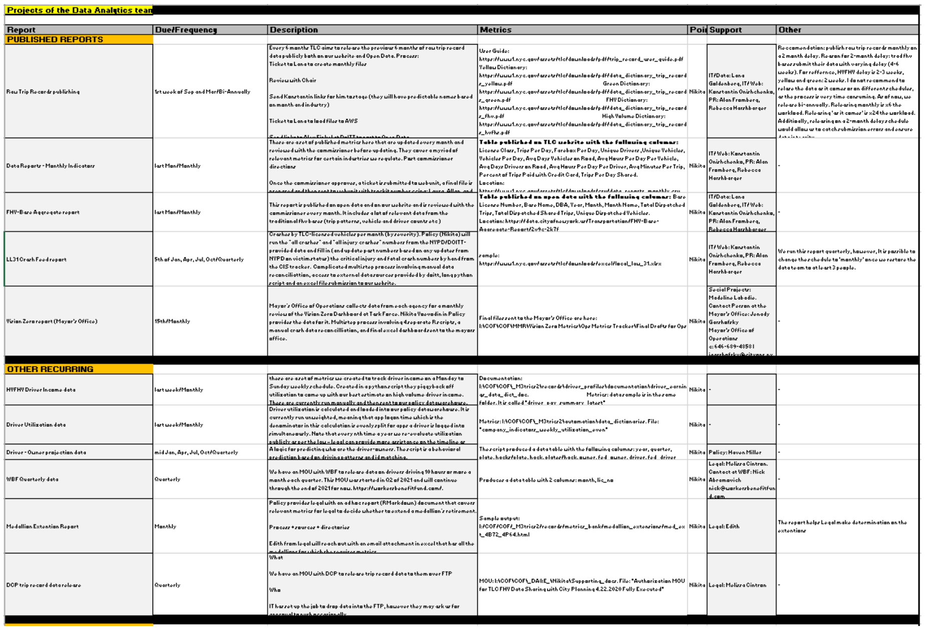
IT/Data: Lana Goldenberg, IT/ Web: Konstantin Onishchenko, PR: Alan Fromberg, Rebecca Harshbarger

- **Reccomendation**

Publish raw trip records monthly on a 2 month delay. Reason for 2-month delay: trad fhv bases submit their data with varying delay (4-6 weeks). For reffernce, HVFHV delay is 2-3 weeks, yellow and green: 2 weeks. I do not recommend to relase the data as it comes or on different schedules, as the process is very time consuming. As of now, we release bi-annually. Releasing monthly is x6 the workload. Releasing 'as it comes' is x24 the workload. Additioally, releasing on a 2-month delay schedule would allow us to catch submission errors and ensure data integrity.

There are many more tasks that we handle. Some of them are listed in the "Recurring_Tasks" Document located at:

```
I:\COF\COF\_DA&E_\Nikita\Reports\Task_spreadsheet
```



Projects of the Data Analytics team

Report	Due/Frequency	Description	Metrics	Poi	Support	Other
PUBLISHED REPORTS						
Raw Trip Records publishing	1st week of Sep and Mar/Bi-Annually	Every 6 months TLC aims to release the previous 6 months of raw trip record data publicly both on our website and Open Data. Process: Ticket to Lana to create monthly files Review with Chair Send Konstantin links for him to stage (they will have predictable names based on month and industry) Ticket to Lana to load files to AWS Send links to Alex Fishel at DoITT to post to Open Data	User Guide: https://www1.nyc.gov/assets/tlc/downloads/pdf/trip_record_user_guide.pdf Yellow Dictionary: https://www1.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf Green Dictionary: https://www1.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_green.pdf FHV Dictionary: https://www1.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_fhv.pdf High Volume Dictionary: https://www1.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_hvfhs.pdf	Nikita	IT/Data: Lana Goldenberg, IT/ Web: Konstantin Onishchenko, PR: Alan Fromberg, Rebecca Harshbarger	Reccomendation: publish raw trip records monthly on a 2 month delay. Reason for 2-month delay: trad fhv bases submit their data with varying delay (4-6 weeks). For refference, HVFHV delay is 2-3 weeks, yellow and green: 2 weeks. I do not recommend to relase the data as it comes or on different schedules, as the process is very time consuming. As of now, we release bi-annually. Releasing monthly is x6 the workload. Releasing 'as it comes' is x24 the workload. Additioally, releasing on a 2-month delay schedule would allow us to catch submission errors and ensure data integrity.
Data Reports - Monthly Indicators	last Mon/Monthly	There are a set of published metrics here that are updated every month and reviewed with the commissioner before updating. They cover a myriad of relevant metrics for certain industries we regulate. Past commissioner directions Once the commissioner approves, a ticket is submitted to webunit, a final file is prepared and then sent to webunit with trackit number cc-ing: Laura, Allan, and	**Table published on TLC website with the following columns:** License Class, Trips Per Day, Farebox Per Day, Unique Drivers ,Unique Vehicles, Vehicles Per Day, Avg Days Vehicles on Road, Avg Hours Per Day Per Vehicle, Avg Days Drivers on Road , Avg Hours Per Day Per Driver, Avg Minutes Per Trip, Percent of Trips Paid with Credit Card, Trips Per Day Shared. Location: https://www1.nyc.gov/assets/tlc/downloads/csv/data_reports_monthly.csv	Nikita	IT/ Web: Konstantin Onishchenko, PR: Alan Fromberg, Rebecca Harshbarger	-
FHV-Base Aggregate report	last Mon/Monthly	This report is published on open data and on our website and is reviewed with the commissioner every month. It includes a lot of relevant data from the traditional fhv bases (trip patterns, vehicle and driver counts etc)	**Table published on open data with the following columns:** Base License Number, Base Name, DBA, Year, Month, Month Name, Total Dispatched Trips, Total Dispatched Shared Trips, Unique Dispatched Vehicles. Location: https://data.cityofnewyork.us/Transportation/FHV-Base-Aggregate-Report/2v9c-2k7f	Nikita	IT/Data: Lana Goldenberg, IT/ Web: Konstantin Onishchenko, PR: Alan Fromberg, Rebecca Harshbarger	-
LL31 Crash Feed report	5th of Jan, Apr, Jul, Oct/Quarterly	Crashes by TLC-licensed vehicles per month (by severity). Policy (Nikita) will run the "all crashes" and "all injury crashes" numbers from the NYPD/DOITT-provided data and fill in (and update past numbers based on any updates from NYPD on victim status) the critical injury and fatal crash numbers by hand from the CIS tracker. Complicated multistep process involving manual data reconciliattion, access to external datasources provided by doitt, long python script and an excel file submission to our website.	sample: https://www1.nyc.gov/assets/tlc/downloads/excel/local_law_31.xlsx	Nikita	IT/ Web: Konstantin Onishchenko, PR: Alan Fromberg, Rebecca Harshbarger	We run this report quarterly, however, It is possible to change the schedule to 'monthly' once we restore the data team to at least 3 people.
Vision Zero report (Mayor's Office)	15th/Monthly	Mayor's Office of Operations collects data from each agency for a monthly review of the Vision Zero Dashboard at Task Force. Nikita Voevodin in Policy provides the data for it. Multistep process involving 4 separate R scripts, a manual crash data reconcilliation, and final excel dashboard sent to the mayors office.	Final files sent to the Mayor's Office are here: I:\COF\COF\MMR\Vision Zero Metrics\Ops Metrics Tracker\Final Drafts for Ops	Nikita	Social Projects: Madeline Labadie. Contact Person at the Mayor's Office: Jenady Gershofsky Mayor's Office of Operations c: 646-689-48581 jgershofsky@cityops.nyc	-
OTHER RECURRING						
HVFHV Driver Income data	last week/Monthly	there are a set of metrics we created to track driver income on a Monday to Sunday weekly schedule. Created in a python script they piggyback off utilization to come up with our best estimate on high volume driver income. These are currently run manually and then sent to our policy datawarehouse.	Documentation: I:\COF\COF_M3trics2\records\driver_profiles\documentation\driver_earnings_data_dict_doc. Metrics: data sample is in the same folder. It is called "driver_pay_summary_latest"	Nikita	-	-
Driver Utilization data	last week/Monthly	Driver utilization is calculated and loaded into our policy datawarehouse. It is currently run unweighted, meaning that app logon time which is the denominator in this calculation is evenly split for apps a driver is logged into simultaneously. Note that every nth time a year we re-evaluate utilization publicly as per the law – legal can provide more assistance on the timeline as	Metrics: I:\COF\COF_M3trics2\automation\data_dictionaries. File: "company_indicators_weekly_utilization_even"	Nikita	-	-
Driver - Owner projection data	mid Jan, Apr, Jul, Oct/Quarterly	A logic for predicting who are the driver-owners. The script is a behavioral prediction based on driving patterns and id matching.	The script produced a data table with the following columns: year, quarter, plate, hacks/plate, hack, plates/hack, owner, fed_owner, driver, fed_driver	Nikita	Policy: Haven Miller	-
WBF Quarterly data	Quarterly	We have an MOU with WBF to release data on drivers driving 10 hours or more a month each quarter. This MOU was started in Q2 of 2021 and will continue through the end of 2021 for now. https://workersbenefitfund.com/.	Produces a data table with 2 columns: month, lic_no	Nikita	Legal: Melissa Cintron. Contact at WBF: Nick Abramovich nick@workersbenefitfund.com	-
Medallion Extention Report	Monthly	Policy provides legal with an ad hoc report (RMarkdown) document that covers relevant metrics for legal to decide whether to extend a medallion's retirement. Process + sources + directories Edith from legal will reach out with an email attachment in excel that has all the medallions for which she requires metrics	Sample output: I:/COF/COF/_M3trics2/records/metrics_bank/medallion_extensions/med_ext_4B72_4P64.html	Nikita	Legal: Edith	The report helps Legal make determination on the extentions
DCP trip record data release	Quarterly	What We have an MOU with DCP to release trip record data to them over FTP Who IT has set up the job to drop data into the FTP, however they may ask us for approval to push occasionally	MOU: I:\COF\COF_DA&E_\Nikita\Supporting_docs. File: "Authorization MOU for TLC FHV Data Sharing with City Planning 4.22.2020 Fully Executed"	Nikita	Legal: Melissa Cintron	-

Figure 8.1: Recurring tasks ex

Chapter 9

Useful Code Snippets & Tricks

TBD