NHTS Introduction

Darshan Pandit

NHTS: Let's write some code!

author: Darshan Pandit

date: September 17, 2019

Overview

- Installation
- NHTS: Data Organization
- Summarize NHTS Package
  - `+ Primitive Queries`
  - `+ Generating Estimates`
  - `+ Creating Custom Variables`
  - `+ Visualizing Results`
- Cases from the NHTS Summary Report

Setting up tools

- Install Rstudio: Click Here!
- Install R: Select a mirror from the list here and proceed
- If on Windows, Install Rtools found here

RStudio: Quick overview

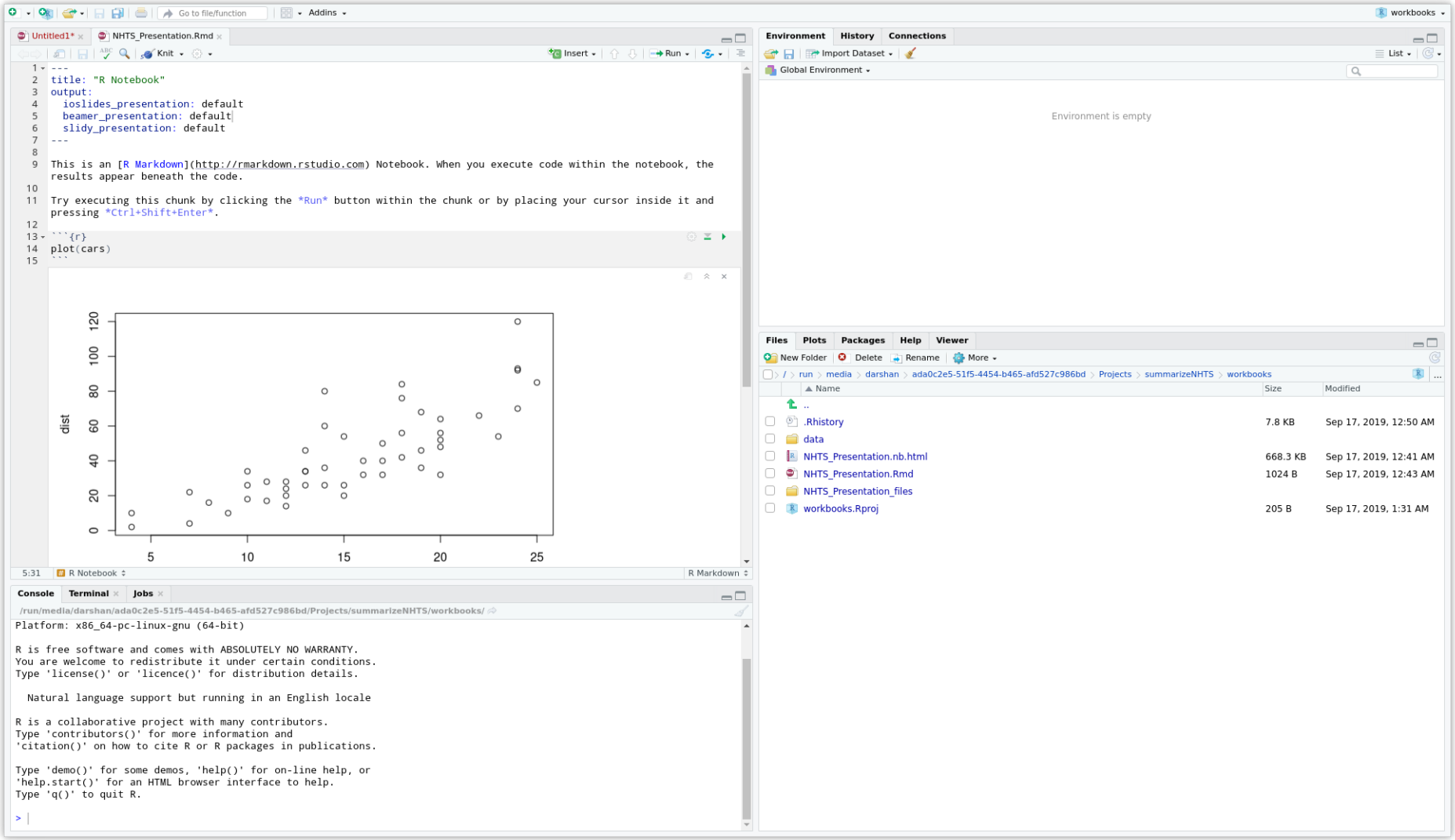

Figure 1: Layout of Rstudio

Installing summarizeNHTS package

In your Rstudio's console:

```
install.packages('devtools')
devtools::install_github('Westat-Transportation/summarizeNHTS')
```

@ Linux Users:

```
install.packages('devtools')
devtools::install_github('darshanpandit/summarizeNHTS')
```

@ MacOS Users:

```
Please contribute to my Macbook fund! :P
```

Let's verify your installation...

- Create a new R Notebook
- Insert/Modify code chunk as follows

```
library(summarizeNHTS)
download_nhts_data("2017", exdir="C:/NHTS")
```

You may change the directory of your choice or pass a relative directory. for eg: '/NHTS'

I'M READY!!! I'M READY!!! I'M READY!!!

Overview

- Installation
- NHTS: Data Organization
- Summarize NHTS Package
  - `+ Generating Estimates`
  - `+ Creating Custom Variables`
  - `+ Visualizing Results`
- Cases from the NHTS Summary Report

NHTS: Data Organization

Let's explore the variables

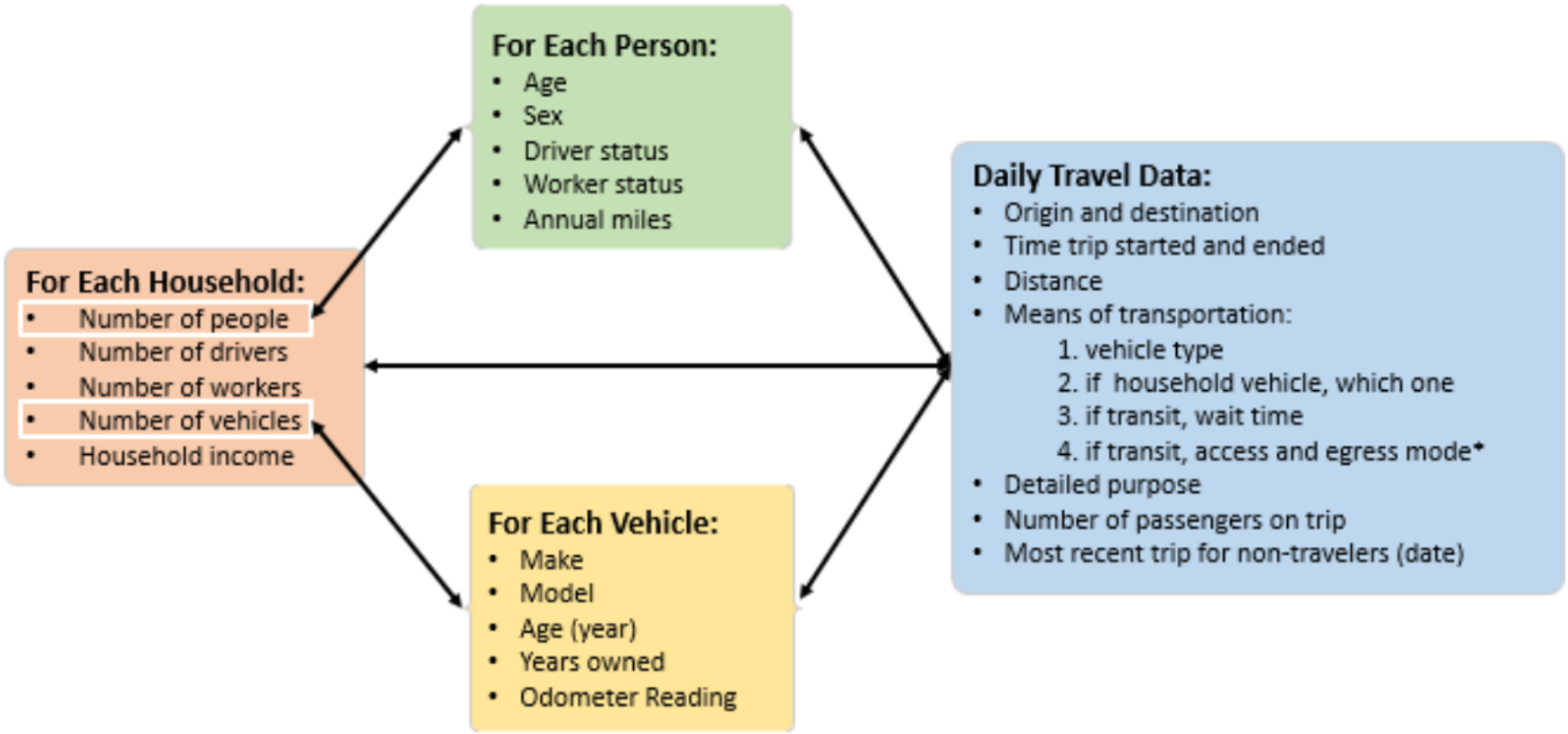

Figure 3: NHTS Schematic Diagram

NHTS: Data Organization

```
summary(dataset$data)
```

```
           Length Class      Mode
trip       61     data.table list
person     87     data.table list
household  58     data.table list
vehicle    19     data.table list
```

```
dataset$data$vehicle
```

NHTS: Data Organization

Accessing Specific Columns

```
# By position
dataset$data$vehicle[, c(1, 3)]

# By name (single variable)
dataset$data$vehicle$ANNMILES

# By name
dataset$data$vehicle[, list(HOUSEID, ANNMILES)]
```

NHTS: Data Organization

Accessing Specific Rows

```
# By row numbers (first 5 rows)
dataset$data$vehicle[1:5, ]

# By condition
dataset$data$vehicle[VEHTYPE == "01", ]

# By condition (multiple values)
dataset$data$vehicle[VEHTYPE %in% c("01","02"), ]
```

NHTS: Data Organization

Print Vehicle make, model

Use NHTS Codebook

NHTS: Data Organization

Print Vehicle make, model

```
dataset$data$vehicle[, list(MAKE, MODEL)]
```

Generating Estimates

- Introduction to the summarize_data function
- Understanding summarize_data parameters
- Statistics grouped by variables
- Exploring aggregation options
- Estimates using a subset condition
- Referencing the documentation

Generating Estimates: Introduction to the summarize_data

summarizeData provides a simple interface to perform complex queries on the NHTS datasets

```
summarize_data(
  data = dataset,
  agg = "household_count"
  )
)
```

W <dbl>	E <dbl>	S <int>	N <int>
118208251	1.186759e-07	129696	129696

1-1 of 1 rows

Figure 4: Aggregate by household_count

Generating Estimates: Introduction to the summarize_data

What do these values mean?

```
W - Weighted statistic.
    Count of households weighted to the population

E - Standard error of the weighted statistic.
    Standard error of the weighted count of households

S - Surveyed/sampled statistic (unweighted statistic).
    The count of sampled households

N - Number of observations/sample size.
    The number of observations is the same as the count
    of sampled households in this example
```

Generating Estimates: Understanding summarize_data parameters

```
summarize_data(
  data = dataset,
  agg = "household_count"
  )
)
```

Required parameters

```
data - NHTS dataset object
      Will always be the output of read_data.
      In our example, we stored the output in the dataset object.

agg - Aggregate function label. Our example used 'household_count'
      but agg could be a number of other labels.
```

Generating Estimates

- Introduction to the summarize_data function
- Understanding summarize_data parameters
- Statistics grouped by variables
- Exploring aggregation options
- Estimates using a subset condition
- Referencing the documentation

Generating Estimates: Statistics grouped by variables

```
summarize_data(
  data = dataset,
  agg = "household_count",
  by = "HOMEOWN"
)
```

HOMEOWN <fctr>	W <dbl>	E <dbl>	S <int>	N <int>
I dont know	1728.364	1060.471	3	3
I prefer not to answer	36993.405	10316.593	32	32
Own	74518546.455	6340.138	98459	98459
Rent	42463980.848	88742.230	30268	30268
Some other arrangement	1187001.928	99157.187	934	934

5 rows

Generating Estimates: Statistics grouped by variables

You can add multiple columns!

```
summarize_data(
  data = dataset,
  agg = "household_count",
  by = c("HH_RACE","HOMEOWN")
)
```

Generating Estimates: Statistics grouped by variables

Generating Frequencies/Proportions

For any of the following Aggregate Fields:

```
'household_count', 'person_count', 'trip_count', 'vehicle_
```

Parameter prop=TRUE

Generating Estimates: Statistics grouped by variables

```
# Proportion of persons by WORKER, worker status
summarize_data(
  data = dataset,
  agg = "person_count",
  by = "WORKER",
  prop = TRUE
)
```

	Person Frequency			
	Weighted	**MOE (95%)**	**Surveyed**	**N**
WORKER				
Not ascertained	0.01%	0.01%	0.01%	1700%
Appropriate skip	15.09%	0.25%	10.57%	2793800%
Yes	52.05%	0.34%	48.55%	12828800%
No	32.86%	0.49%	40.87%	10799100%

Generating Estimates

- Introduction to the summarize_data function
- Understanding summarize_data parameters
- Statistics grouped by variables
- Exploring aggregation options
- Estimates using a subset condition
- Referencing the documentation

Generating Estimates: Exploring aggregation options

Using Numeric Aggregates

Instead of Aggregate fields directly, following operands can also be passed

```
'sum', 'avg', 'median'
```

If this is done, agg_var parameter containing the field/column must be passed

agg_var must be numeric

Missing or Invalid Values (-1,-7,-8,-9) are automagically handled!

Generating Estimates: Exploring aggregation options

```
summarize_data(
  data = dataset,
  agg = "avg",
  agg_var = "TRPMILES"
)
```

Average TRPMILES			
Weighted	**MOE (95%)**	**Surveyed**	**N**
10.70	0.40	11.45	922,916

Generating Estimates: Exploring aggregation options

Generate Median Trip Miles by Homeownership ?

	Median TRPMILES			
	Weighted	**MOE (95%)**	**Surveyed**	**N**
HOMEOWN				
I dont know	1.56	69.54	1.56	29
I prefer not to answer	2.91	4.38	3.31	133
Own	3.72	0.06	3.65	730,119
Rent	2.59	0.10	2.73	186,492
Some other arrangement	3.47	0.93	2.85	6,143

Generating Estimates: Exploring aggregation options

Solution:

```
summarize_data(
  data = dataset,
  agg = "median",
  agg_var = "TRPMILES",
  by = "HOMEOWN"
)
```

Generating Estimates: Exploring aggregation options

You can use either of the following trip rate aggregates

`'household_trip_rate' - Daily Person Trips per Household`
`'person_trip_rate' - Daily Person Trips per Person`

```
summarize_data(
  data = dataset,
  agg = "person_trip_rate",
  by = "WORKER"
)
```

Generating Estimates

- Introduction to the summarize_data function
- Understanding summarize_data parameters
- Statistics grouped by variables
- Exploring aggregation options
- Estimates using a subset condition
- Referencing the documentation

Generating Estimates: Estimates using a subset condition

Pre-aggregation subset conditions can be specified using the subset parameter.

`Argument should be passed as a string.`

Subsetting character variables

```
#Distribution of Leisure Travel
summarize_data(
  data = dataset,
  agg = "trip_count",
  by = "TRAVDAY",
  prop = TRUE,
  subset = "WHYTRP90 %in% c('07','08','10')"
)
```

Generating Estimates: Estimates using a subset condition

Subsetting Numeric Variables

```
# Person trip rate by Sex (for millennials)
summarize_data(
  data = dataset,
  agg = "person_trip_rate",
  by = "R_SEX",
  subset = "R_AGE >= 18 & R_AGE <= 34"
)
```

	Person Trip Rate			
	Weighted	**MOE (95%)**	**Surveyed**	**N**
R_SEX				
I dont know	0.95	2.17	1.33	4
I prefer not to answer	3.41	1.25	3.65	124
Male	3.15	0.09	3.26	63,591
Female	3.45	0.11	3.59	77,427