2010 California Household Travel Survey

Linked Trip Summaries

2014-10-09 09:12:40

Introduction

These slides present a summary of the method being used to generate linked trip tables on the 2010 CHTS data.

- A pdf version of this document
- The linked trip table is available as a compressed SQL file upon request.
- The code for this project is maintained as a git repository. Access is available upon request.
- PLEASE NOTE: This is a working document very much still a draft. ***It was last updated on 2014-10-09 09:12:40***

Change Log

```
commit cf2438e7a7217d7588cc583f9395074bd1eb45dd
Author: Craig Rindt <crindt@gmail.com>
Date:   Wed Sep 10 00:17:36 2014 -0700

    * Added triptypes by region tables (still some work to do)
    * Tweaked css to reduce table font size for region tables
    * Streamlined group specifications for weekdays/weekends, and driver-trips/person-trips

commit 7bc125b889774c4652c55be4197a5d18ec5999ec
Author: Craig Rindt <crindt@gmail.com>
Date:   Tue Sep 9 17:41:52 2014 -0700

    * Merged code to set factors on raw data into load-trip-tables.R
    * Added some previously missing source files to repo.

commit b8bcbc7b03998096a15331c62adfe9f5e91fb2c9
Author: Craig Rindt <crindt@gmail.com>
Date:   Tue Sep 9 17:36:56 2014 -0700

    * Added note regarding git repo
    * Added script to sychronize slides

commit d1c2e668f0c1a61820edcb611321084de314ec0c
```

```
Author: Craig Rindt <crindt@gmail.com>
Date:   Tue Sep 9 17:31:24 2014 -0700

    Elaborated on the trip type definitions in the SQL functions

commit 18af2b6f3be1aadf06b63578f858f773e427a911
Author: Craig Rindt <crindt@gmail.com>
Date:   Tue Sep 9 17:06:56 2014 -0700

    Change margins to 1in on pdf output

commit be77bf463ceaef549f305540cd5e7731ed75986f
Author: Craig Rindt <crindt@gmail.com>
Date:   Tue Sep 9 17:02:32 2014 -0700

    * Removed linked trip table sql (shouldn't be public)
    * Add robots.txt to discourage crawlers

commit 98b147abdbc227ac99596012a01a07c1f70bd364
Author: Craig Rindt <crindt@gmail.com>
Date:   Tue Sep 9 16:53:30 2014 -0700

    * Switch to use of loaded code chunks (for reusability).
    * Included slide on how trip types are defined

commit 339a7fd04410751a17a45ec78c70668456cd1f73
Author: Craig Rindt <crindt@gmail.com>
Date:   Tue Sep 9 15:39:50 2014 -0700

    More cleanup

commit d8bcdda9657301735077af9ce2c2da2c3c745c5e
Author: Craig Rindt <crindt@gmail.com>
Date:   Tue Sep 9 15:39:38 2014 -0700

    Fix runit export/import command to use database directly

commit b53ee1578eff789ec5d8460e44fc04d1373bfed7
Author: Craig Rindt <crindt@gmail.com>
Date:   Tue Sep 9 15:38:28 2014 -0700

    Cleaning up unneeded files

commit dbab482bc80255af308bc6d76e7cc18cdc303ca5
Author: Craig Rindt <crindt@gmail.com>
Date:   Tue Sep 9 15:37:40 2014 -0700

    Initial commit
```

Target Tools

The following tools have been chosen as the target tools because they are mature and actively maintained free and open source software.

- Relational datastore: PostgresSQL
- Statistical package: R

Conversion steps

Platform

The following conversion was performed on a mid-level workstation running the 13.04 Ubuntu Linux distribution. Only open source tools were required. (Use down arrow to see details)

Convert raw MS-Access (.mdb) databases for use PostgreSQL

The following shell script uses the `mdb-tools` package and standard unix tools to convert complete set of tables in the Access database to tab separated value files, which are then loaded into `PostgreSQL`.

```
#!/bin/bash
# modified from: http://barbedwirebytecodebaconburger.wordpress.com/2009/08/03/migrating-an-old-ms-acce
DBFILE=$1
DBNAME=$2
BASENAME=`basename $1 .mdb`
OUTFILE=${BASENAME}-schema.sql
DIALECT=postgres

runtraced() {
    echo "$@"
    "$@"
}
tolower() {
    echo "$@" | tr '[:upper:]' '[:lower:]'
}


#Check for correct number of arguments
if [ $# -lt 2 ]; then
echo "Usage: access2mysql.sh DBFILE DBNAME [DIALECT]"
echo "Example: access2mysql.sh msaccess.mdb mysql.sql"
exit 1
fi

if [ $# -eq 3 ]; then
    DIALECT=$3
fi
echo "Selected Dialect: $DIALECT"

#Check that DBFILE really exists
if [ ! -f $DBFILE ]; then
echo "$DBFILE does not exist."
exit 1
fi

#All is good, here we go!
```

```
#Create schema

mdb-schema --no-indexes $DBFILE $DIALECT > $OUTFILE
runtraced dropdb -U postgres ${DBNAME}
runtraced createdb -U postgres ${DBNAME}
psql -U postgres ${DBNAME} < $OUTFILE

#Export table data
TABLES=`mdb-tables $DBFILE`


for TT in $TABLES; do
    FTT=${BASENAME}-${TT}

    # Do the export on this table, escaping end of line records so we can handle
    # multiple line text fields
    mdb-export -Q -d '\t' -D '%Y-%m-%d %H:%M:%S' -R "XXTIANRXX" t.mdb "$TT" \

        # pipe the export through a perl filter to escape newlines in records
        | perl -p -e 's/^M\n/\\n/' \  #

        # pipe again through a perl filter to escape quote characters
        | perl -p -e 's/"/""/g' | perl -p -e 's/XXTIANRXX/\n/g' \

        # send the result into a tsv file
        > ${TT}.tsv

    # Clean up some Windows-character stuff
    dos2unix ${TT}.tsv

    # Copy the tsv file into the appropriate database table
    runtraced psql -U postgres -c "\copy \"${TT}\" from '${TT}.tsv' using delimiters E'\t' with CSV hea
done

# DOWNCASE TABLE NAMES
for TT in $TABLES; do
    LTT=`tolower "${TT}"`
    runtraced psql -U postgres -c "ALTER TABLE \"${TT}\" RENAME TO \"${LTT}\"" ${DBNAME}
done


#dos2unix $OUTFILE
exit 0
```

Generate static linked trip table consistent with 2001 CHTS datasets

The following SQL command is used to generate `tbllinkedtrip`, containing the linked trip tables for use in R

```
--- Here we create a view that generates linked trips and gives each a distinct,
```

```
--- ordered, ID
drop view if exists linked_trip_order cascade;
create or replace view linked_trip_order AS
-- This query generates the ordered linked trips for each SAMPN, PERNO with
-- merged aggregates based upon the unique linked trip numbering generated in
-- the embedded subquery `q`
select
       -- Rows for this view are ordered linked trips for each SAMPN, PERNO
       qq."SAMPN",qq."PERNO",qq.linked_tripno,

       --- We tack on some aggregated values for later use
       --- NOTE: array_agg is a postgresql function
       array_agg(qq."PLANO") jplano,  -- array of joined place numbers for this linked trip (1 or more)
       array_agg(qq."ACTNO") jactno,  -- array of joined activity numbers (1 or more)
       array_agg(qq."APURP") jpurp,   -- array of joined purposes (1 or more)
       array_agg(qq.adjtripno) jtrip, -- array of adjusted trip numbers [nulls -> 0] (1 or more)

       array_agg(dp."PERWGT") perwgt,
       array_agg(dp."EXPPERWGT") expperwgt,
       array_agg(dp."TCF") tcf,
       array_agg(dp."TCFPERWGT") tcfperwgt,
       array_agg(dp."EXPTCFPERWGT") exptcfperwgt,

       SUM(dp."TRIPDUR") trpdur,         -- sum of durations of joined trips
       SUM(dp."ACTDUR") jactdur           -- sum of activity durations of joined trips/activities
from (
        select *,
               -- this record is the point of this subquery.  We sum the linked_tripcnt of the ordered
               -- to generate a unique linked trip number for every activity
               sum(linked_tripcnt) OVER (ORDER BY "SAMPN","PERNO","PLANO","ACTNO",adjtripno) AS linked_
        from (
                --- this subquery generates an ordered listing of trips where
                --- adjacent pick-up/delivery or mode switches are collapsed
                --- into a single trip
                SELECT da."SAMPN",da."PERNO",da."PLANO",da."ACTNO",
                       null_to_zero("TRIPNO") adjtripno,  -- convert NULL tripno to 0 for later computa
                       da."APURP",

                       --- compute the linked trip counter...
                       --- linked_tripcnt is zero if...
                       case when
                            -- the place number is the same
                                 -- this makes sure we capture all distinct
                                 -- *activities*, which means they might be
                                 -- occuring at the same location.
                                 -- "PLANO" = lag("PLANO") over (PARTITION BY "SAMPN","PERNO" ORDER BY
                                 false  -- not sure we want the above so omitting

                            -- OR the trip number differs from the last AND the trip purpse is
                            --    APURP=21:Mode change/Transfer; 22:Pick-up/Drop off
                                 OR (
                                        null_to_zero("TRIPNO") <> lag(null_to_zero("TRIPNO"))
                                              OVER (PARTITION BY "SAMPN","PERNO" ORDER BY null_to_zero(
                                        AND lag("APURP")
```

```
                                                OVER (PARTITION BY "SAMPN","PERNO" ORDER BY null_to_zero(
                                                NOT IN (21,22)
                                        )
                                THEN 1    -- increment the linked trip count
                                ELSE 0    -- DON'T increment the linked trip count
                                END AS linked_tripcnt

                    FROM deliv_activity da
                    ORDER BY null_to_zero("TRIPNO"),"PLANO","ACTNO"
                ) q
) qq
     LEFT JOIN deliv_place dp USING ( "SAMPN","PERNO","PLANO")
     GROUP BY "SAMPN","PERNO",linked_tripno order by "SAMPN","PERNO",linked_tripno;


-- This is an intermediate view that we use to add the origin and destination
-- place/activity pairs to the linked_trip_order view
DROP VIEW IF EXISTS ltf CASCADE;
CREATE OR REPLACE VIEW ltf AS
SELECT linked_tripno -- unique linked trip number (to index)
       ,lto."SAMPN"  -- hh number (key)
       , lto."PERNO" -- person number in hh (key)

       -- Here, use window functions to get the last place/activity from the
       -- prior record for this person.  Since we've ordered the trips, this
       -- will be the source place/activity for this trip
       , last_elem(lag(jplano) OVER (PARTITION BY "SAMPN","PERNO" ORDER BY "SAMPN","PERNO",linked_tripn
       , last_elem(lag(jactno) OVER (PARTITION BY "SAMPN","PERNO" ORDER BY "SAMPN","PERNO",linked_tripn

       -- the dest place/activity is simply the last of the joined
       -- place/activities in this linked trip
       , last_elem(jplano) dest_plano   -- destination "place" of this trip
       , last_elem(jactno) as dest_act  -- activity to which this trip arrives

       -- pass along the place/activity arrays for convenience
       , jplano
       , jactno

       -- pass along the trip and activity durations for this linked trip
       , trpdur
       , jactdur

       -- pass along weights
       , perwgt
       , expperwgt
       , tcf
       , tcfperwgt
       , exptcfperwgt

FROM linked_trip_order lto;

-- this is the final view, that adds the details for the source and destination
-- ends of each linked trip
DROP VIEW IF EXISTS theone CASCADE;
```

```
CREATE OR REPLACE VIEW theone AS
SELECT ltf.*
       , NULL dayno  -- legacy
       , NULL source_locno   -- legacy specification of geocoded source location
       , NULL dest_locno     -- legacy specification of geocoded destination location
       , NULL dtype          -- ?? Destination type?
       , dpdst."MODE" tmode  -- primary mode of travel---*last* mode of joined travel
       , NULL mapped_mode    -- primary mode mapped to reduced mode class

       -- HW, HO, HS, WO, OO
       , canonical_trip_type(canonical_place_type(dpsrc."PNAME",dasrc."APURP")
                           || canonical_place_type(dpdst."PNAME",dadst."APURP"))
         AS triptype

       , dpsrc."DEP_HR" dep_hr         -- depature hour
       , dpsrc."DEP_MIN" dep_min       -- depature min
       , dpdst."ARR_HR" arr_hr         -- arrival hour
       , dpdst."ARR_MIN" arr_min       -- arrival min
       , dpdst."ACTDUR" AS actdur      -- duration of activity at destination
       , jactdur - dpdst."ACTDUR" AS ignoredactdur  -- duration of transit/serve passenger type activit
       , null vehavail       -- ?? whether a vehicle was available to the traveler?
       , null vehno          -- ?? HH vehicle used?
       , null party          -- ?? number of travelers in party?, null or 1 if 1
       -- , null DOM_WDWGT
       -- , null DOM_WEWGT
       -- , null DOM_SDWGT
       -- , null DOM_AWDWGT
       -- , null DOM_ASDWGT
       -- , null Orig_DOM_AWDWGT
       -- , null Orig_DOM_WEWGT
       -- , null PHASE
       from ltf

       -- join the activity and place tables for the source and destination ends
       -- of the trip
       left join deliv_activity dasrc
            ON (ltf."SAMPN" = dasrc."SAMPN" AND ltf."PERNO" = dasrc."PERNO"
                AND ltf.source_plano = dasrc."PLANO" AND ltf.source_act = dasrc."ACTNO" )
       left join deliv_activity dadst
            ON (ltf."SAMPN" = dadst."SAMPN" AND ltf."PERNO" = dadst."PERNO"
                AND ltf.dest_plano = dadst."PLANO" AND ltf.dest_act = dadst."ACTNO" )
       left join deliv_place dpsrc
            ON (ltf."SAMPN" = dpsrc."SAMPN" AND ltf."PERNO" = dpsrc."PERNO"
                AND ltf.source_plano = dpsrc."PLANO" )
       left join deliv_place dpdst
            ON (ltf."SAMPN" = dpdst."SAMPN" AND ltf."PERNO" = dpdst."PERNO"
                AND ltf.dest_plano = dpdst."PLANO" )
       left join deliv_per p
            ON (ltf."SAMPN" = p."SAMPN" AND ltf."PERNO" = p."PERNO")

       -- Omit records where the source place and destination place are
       -- identical or it's the first place/activity for the person (those
       -- aren't trips!)
       where ltf.source_plano != ltf.dest_plano AND ltf.source_plano IS NOT NULL AND ltf.source_act IS
```

```
      order by ltf."SAMPN", ltf."PERNO", source_plano, source_act;

--- Finally, create the linked trip table
drop table tbllinkedtrip;
select * into tbllinkedtrip from theone;
```

Function definitions...

```
--- Quick function to convert nulls to zero
CREATE FUNCTION null_to_zero (integer) RETURNS integer AS $$
 SELECT CASE WHEN $1 IS NULL THEN 0 ELSE $1 END
$$ LANGUAGE SQL;

--- function to get last element of arbitrarily sized array
--- from: http://stackoverflow.com/questions/2949881/getting-the-last-element-of-a-postgres-array-decla
CREATE FUNCTION last_elem (integer[]) RETURNS integer AS $$
 SELECT $1[array_upper($1,1)];
$$ LANGUAGE SQL;

--- function to get first element of array
CREATE FUNCTION first_elem (integer[]) RETURNS integer AS $$
 SELECT $1[1];
$$ LANGUAGE SQL;

--- convert 3am->3am hours to 3:00->27:00 for sorting purposes
CREATE FUNCTION thr (integer) RETURNS integer AS $$
 SELECT CASE WHEN $1 < 3 THEN $1+24 ELSE $1 END
$$ LANGUAGE SQL;

-- time difference in minutes
CREATE FUNCTION tdiff (integer,integer,integer,integer) RETURNS integer AS $$
 SELECT 60*thr($3)+$4 - (60*thr($1)+$2)
$$ LANGUAGE SQL;


CREATE OR REPLACE FUNCTION array_avg(double precision[])
RETURNS double precision AS $$
SELECT avg(v) FROM unnest($1) g(v)
$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION array_max(double precision[])
RETURNS double precision AS $$
SELECT max(v) FROM unnest($1) g(v)
$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION array_min(double precision[])
RETURNS double precision AS $$
SELECT min(v) FROM unnest($1) g(v)
$$ LANGUAGE sql;
```

Importing data from PostgreSQL to R

```
# Load one of the many Postgresql interface library
library(RPostgreSQL)

# create an PostgreSQL instance and create one connection.
drv <- dbDriver("PostgreSQL")
con <- dbConnect(drv, dbname = "chts-2010", user = "postgres")

# query the linked trip table and put into a data frame
rs <- dbSendQuery(con, paste("select tmode as \"MODE\", ", "triptype,\"DOW\",",
    "array_avg(exptcfperwgt) as weight, perwgt[1], tcf, ", "tcfperwgt[1] tcfperwgt, hh.\"AREA\" ",
    "from tbllinkedtrip", " left join deliv_hh hh using (\"SAMPN\")"))
df <- fetch(rs, n = -1)

# assign factor labels
df$triptype <- factor(df$triptype, levels = c("HW", "HO", "HS", "WO", "OO"),
    labels = c("Home-Work", "Home-Other", "Home-Shop", "Work-Other", "Other-Other"))

df$DOW <- factor(df$DOW, levels = c(1, 2, 3, 4, 5, 6, 7), labels = c("Mo", "Tu",
    "We", "Th", "Fr", "Sa", "Su"))

df$MODE <- factor(df$MODE, levels = seq(1, 29), labels = c("Walk", "Bike", "Wheelchair / Mobility Scoot
    "Other Non-Motorized", "Auto / Van / Truck Driver", "Auto / Van / Truck Passenger",
    "Carpool / Vanpool", "Motorcycle / Scooter / Moped", "Taxi / Hired Car / Limo",
    "Rental Car/Vehicle", "Private shuttle (SuperShuttle, employer, hotel, etc.)",
    "Greyhound Bus", "Plane", "Other Private Transit", "Local Bus, Rapid Bus",
    "Express Bus / Commuter Bus (AC Transbay, Golden Gate Transit, etc)", "Premium Bus ( Metro Orange /
    "School Bus", "Public Transit Shuttle (DASH, Emery Go Round, etc.)", "AirBART / LAX FlyAway",
    "Dial-a-Ride / Paratransit (Access Services, etc.)", "Amtrak Bus", "Other Bus",
    "BART, Metro Red / Purple Line", "ACE, Amtrak, Caltrain, Coaster, Metrolink",
    "Metro Blue / Green / Gold Line, Muni Metro, Sacramento Light Rail, San Diego Sprinter / Trolley / (
    "Street Car / Cable Car", "Other Rail", "Ferry / Boat"))

df$AREA <- factor(df$AREA, levels = seq(1, 39), labels = c("Alpine", "Amador",
    "AMBAG", "Butte", "Calaveras", "Colusa", "Del Norte", "Fresno", "Glenn",
    "Humboldt", "Inyo", "Kern", "Kings", "Lake", "Lassen", "Madera", "Mariposa",
    "Mendocino", "Merced", "Modoc", "Mono", "MTC", "Nevada", "Plumas", "SACOG",
    "San Joaquin", "San Luis Obispo", "SANDAG", "Santa Barbara", "SCAG", "Shasta",
    "Sierra", "Siskiyou", "Stanislaus", "Tehama", "TMPO", "Trinity", "Tulare",
    "Tuolumne"))
```

Trips by Type

A note on trip types

For the purposes of this analysis, each linked trip origin and destination was classified using the following `SQL` functions. There is room for discussion here regarding these definitions. Arrow down to see trip type distributions.

```
-- Function to determine the "TYPE" of a "place" based upon
--   $1: the place name
```

```
--   $2: the activity type performed at that place
CREATE OR REPLACE FUNCTION canonical_place_type (VARCHAR(510),INTEGER)
RETURNS character(1) AS $$
       SELECT (CASE WHEN $1 = 'HOME' THEN 'H'
                    WHEN $1 = 'WORK' OR $2 IN (9
                                               --,11,12,16,25
                                               ) THEN 'W'
                    -- NOTE:
                    -- 9=WORK/JOB DUTIES
                    -- 11=MEALS AT WORK
                    -- 12=WORK-SPONSORED SOCIAL ACTIVITIES
                    -- 16=ALL OTHER WORK-RELATED ACTIVITIES AT MY WORK
                    -- 25=WORK-RELATED (MEETING, SALES CALL, DELIVERY)
                    WHEN $2 IN (26,27,28,29) THEN 'S'
                    -- 26=SERVICE PRIVATE VEHICLE (GAS, OIL, LUBE, REPAIRS)
                    -- 27=ROUTINE SHOPPING (GROCERIES, CLOTHING...)
                    -- 28=SHOPPING FOR MAJOR PURCHASES OR SPECIALTY ITEMS...
                    -- 29=HOUSEHOLD ERRANDS (BANK, DRY CLEANING, ETC.)
                    ELSE 'O'
                    END)
$$ LANGUAGE SQL;

-- Convert directional trip types to canonical trip types
--   $1: A two character string representing the "place" "types" of a trip's
--       origin and destination
--
-- We follow standard conventions here:
--
--   * Any trips with an origin or destination at home is a home-based trip so
--     WH->HW, SH->HS, OH->HO.  Similarly, any trip with and origin or
--     destination at work but *without* an end at home is a work based trip so
--     OW->WO, SW->WS.  Everything else is an OO trip.
CREATE OR REPLACE FUNCTION canonical_trip_type (CHARACTER(2))
RETURNS character(2) AS $$
 SELECT (CASE WHEN $1 = 'WH' THEN 'HW'
              WHEN $1 = 'OH' THEN 'HO'
              WHEN $1 = 'SH' THEN 'HS'
              WHEN $1 = 'OW' THEN 'WO'
              WHEN $1 = 'SS' THEN 'OO'
              WHEN $1 = 'OS' THEN 'OO'
              WHEN $1 = 'SO' THEN 'OO'
              WHEN $1 = 'WS' THEN 'OO'
              WHEN $1 = 'SW' THEN 'OO'
              -- treat HH trips as HO, we see these in linked trips when someone
              -- drops people off and returns home
              WHEN $1 = 'HH' THEN 'HO'
              -- treat WW trips as WO, we see these in linked trips when someone
              -- drops people off and returns to work, and when someone travels to
              -- perform work-related business from their workplace
              WHEN $1 = 'WW' THEN 'WO'
              ELSE $1
              END);
$$ LANGUAGE SQL;
```

Unweighted weekday trips by type

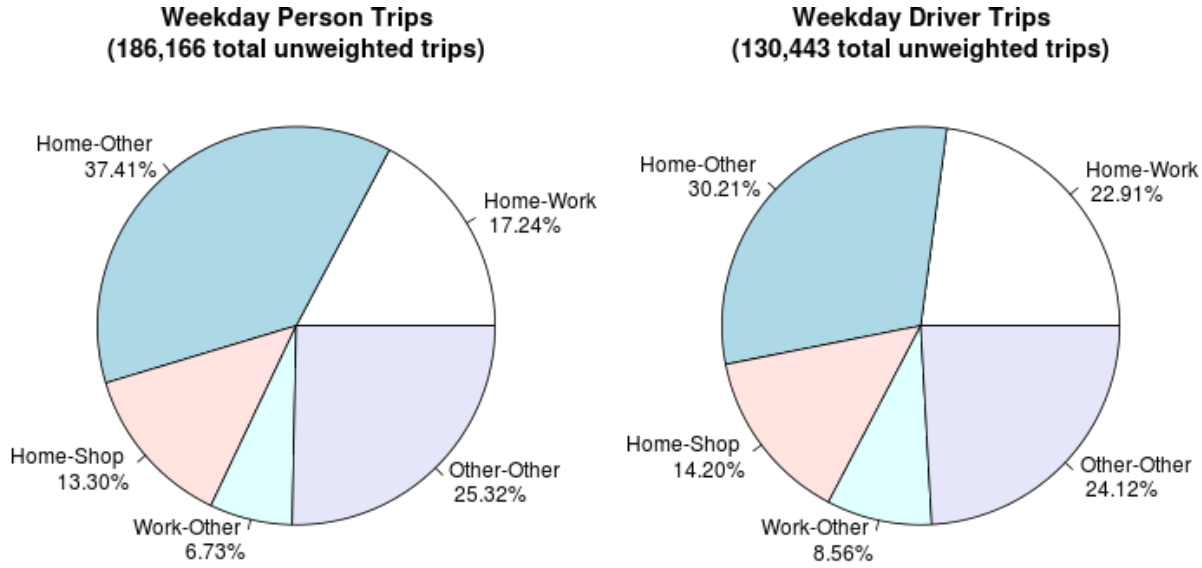

Figure 1:

```
# Load Hmisc for %nin% operator
library(Hmisc)


weekend <- df$DOW %in% c("Sa", "Su")
weekday <- df$DOW %nin% c("Sa", "Su")

# Per CHTS 2001: * Person trips include all trips except walk, bicycle,
# airplane-commercial, Airplane private, and 'other' mode trips.  * Driver
# trips include automobile, pickup truck, RV, Sport Utility Vehicle, van,
# truck, and motorcycle/Moped driver trips
persontrip <- df$MODE %nin% c("Walk", "Bike", "Wheelchair / Mobility Scooter",
    "Other Non-Motorized", "Other Private Transit", "Other Bus", "Other Rail")
drivertrip <- df$MODE %in% c("Auto / Van / Truck Driver", "Motorcycle / Scooter / Moped")


# side by side plots
par(mfrow = c(1, 2), mar = c(1.5, 1.5, 2, 1.5))

# Trip type counts
xt <- xtabs(~triptype, data = df[weekday & persontrip, ])
pie(xt/sum(xt), main = paste("Weekday Person Trips", "\n", paste("(", prettyNum(sum(xt),
    big.mark = ",", scientific = F), " total unweighted trips", ")", sep = ""),
    sep = ""), labels = paste(names(xt), "\n", sprintf("%1.2f%%", xt/sum(xt) *
    100, sep = "")))

xt <- xtabs(~triptype, data = df[weekday & drivertrip, ])
pie(xt/sum(xt), main = paste("Weekday Driver Trips", "\n", paste("(", prettyNum(sum(xt),
```

```
    big.mark = ",", scientific = F), " total unweighted trips", ")", sep = ""),
    sep = ""), labels = paste(names(xt), "\n", sprintf("%1.2f%%", xt/sum(xt) *
    100, sep = "")))
```

Weighted weekday trips by type

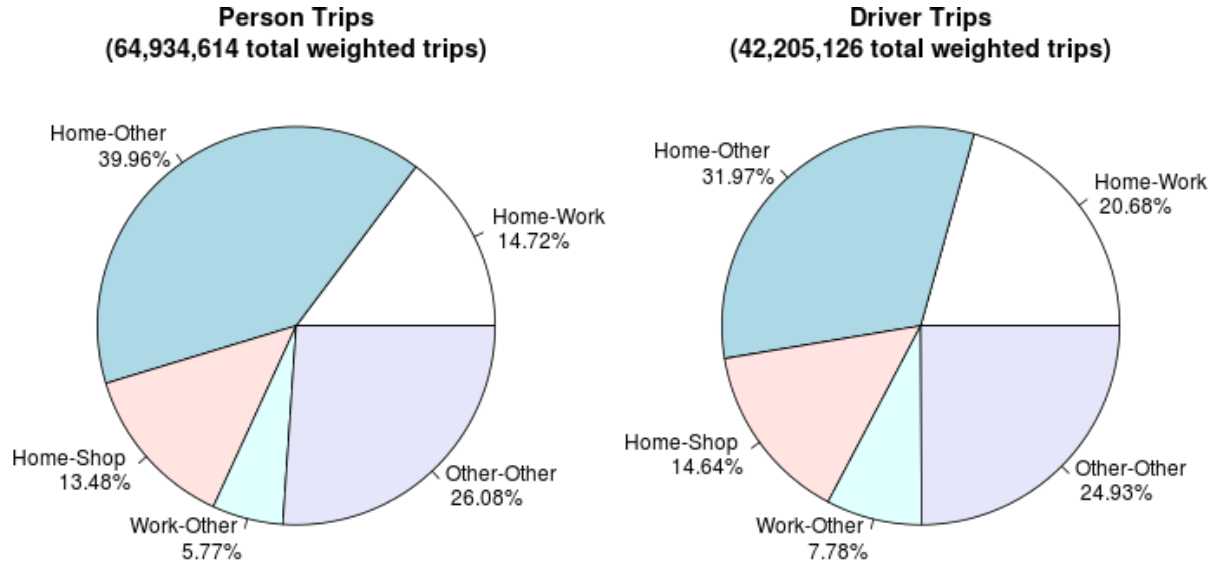

Figure 2:

```
# Load Hmisc for %nin% operator
library(Hmisc)

weekend <- df$DOW %in% c("Sa", "Su")
weekday <- df$DOW %nin% c("Sa", "Su")

# Per CHTS 2001: * Person trips include all trips except walk, bicycle,
# airplane-commercial, Airplane private, and 'other' mode trips.  * Driver
# trips include automobile, pickup truck, RV, Sport Utility Vehicle, van,
# truck, and motorcycle/Moped driver trips
persontrip <- df$MODE %nin% c("Walk", "Bike", "Wheelchair / Mobility Scooter",
    "Other Non-Motorized", "Other Private Transit", "Other Bus", "Other Rail")

drivertrip <- df$MODE %in% c("Auto / Van / Truck Driver", "Motorcycle / Scooter / Moped")

# side by side plots
par(mfrow = c(1, 2), mar = c(1.5, 1.5, 2, 1.5))

# create cross tabulation of summed weights grouped by triptype ...and
# filtered for non weekend days
xt <- xtabs(weight ~ triptype, data = df[weekday & persontrip, ])
pie(xt/sum(xt), main = paste("Person Trips", "\n", paste("(", prettyNum(sum(xt),
    big.mark = ",", scientific = F), " total weighted trips", ")", sep = ""),
    sep = ""), labels = paste(names(xt), "\n", sprintf("%1.2f%%", xt/sum(xt) *
```

```
    100, sep = "")))

xt <- xtabs(weight ~ triptype, data = df[weekday & drivertrip, ])
pie(xt/sum(xt), main = paste("Driver Trips", "\n", paste("(", prettyNum(sum(xt),
    big.mark = ",", scientific = F), " total weighted trips", ")", sep = ""),
    sep = ""), labels = paste(names(xt), "\n", sprintf("%1.2f%%", xt/sum(xt) *
    100, sep = "")))
```

Weighted weekend person trips by type

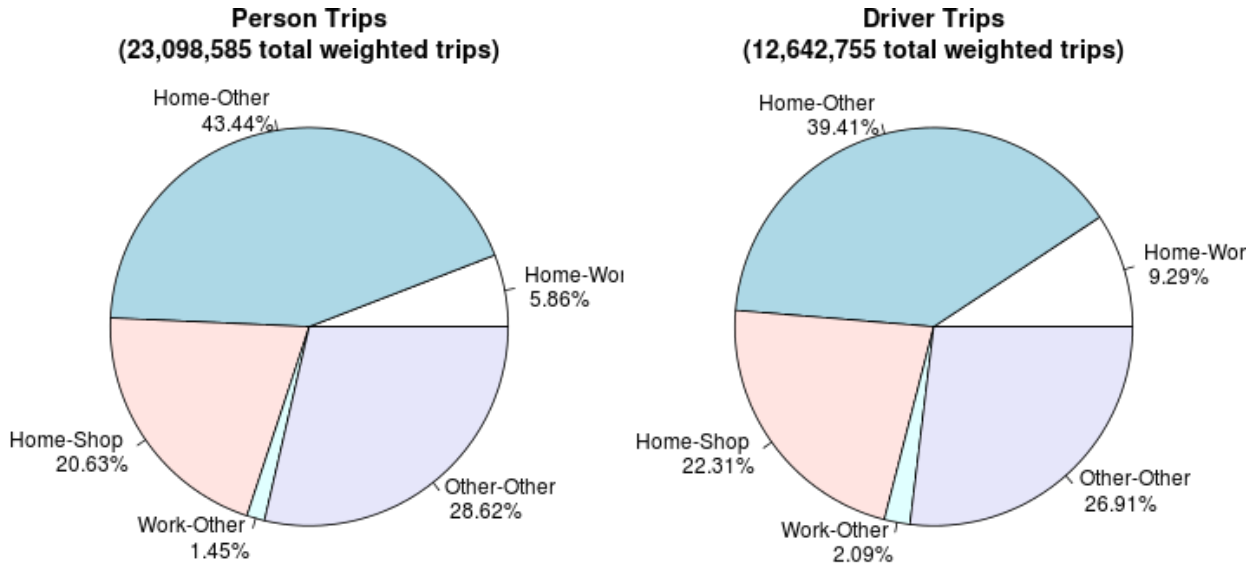

Figure 3:

```
# Load Hmisc for %nin% operator
library(Hmisc)

weekend <- df$DOW %in% c("Sa", "Su")
weekday <- df$DOW %nin% c("Sa", "Su")

# Per CHTS 2001: * Person trips include all trips except walk, bicycle,
# airplane-commercial, Airplane private, and 'other' mode trips.  * Driver
# trips include automobile, pickup truck, RV, Sport Utility Vehicle, van,
# truck, and motorcycle/Moped driver trips
persontrip <- df$MODE %nin% c("Walk", "Bike", "Wheelchair / Mobility Scooter",
    "Other Non-Motorized", "Other Private Transit", "Other Bus", "Other Rail")

drivertrip <- df$MODE %in% c("Auto / Van / Truck Driver", "Motorcycle / Scooter / Moped")

# side by side plots
par(mfrow = c(1, 2), mar = c(1.5, 1.5, 2, 1.5))

# create cross tabulation of summed weights grouped by triptype ...and
# filtered for non weekend days
```

```
xt <- xtabs(weight ~ triptype, data = df[weekday & persontrip, ])
pie(xt/sum(xt), main = paste("Person Trips", "\n", paste("(", prettyNum(sum(xt),
    big.mark = ",", scientific = F), " total weighted trips", ")", sep = ""),
    sep = ""), labels = paste(names(xt), "\n", sprintf("%1.2f%%", xt/sum(xt) *
    100, sep = "")))

xt <- xtabs(weight ~ triptype, data = df[weekday & drivertrip, ])
pie(xt/sum(xt), main = paste("Driver Trips", "\n", paste("(", prettyNum(sum(xt),
    big.mark = ",", scientific = F), " total weighted trips", ")", sep = ""),
    sep = ""), labels = paste(names(xt), "\n", sprintf("%1.2f%%", xt/sum(xt) *
    100, sep = "")))
```

Weighted weekday trips by region

All Areas

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	10343671	13.351	9561041	14.724	8727700	20.679
Home-Other	32459318	41.897	25947302	39.959	13491041	31.965
Home-Shop	10073713	13.003	8750149	13.475	6179157	14.641
Work-Other	4435812	5.726	3743599	5.765	3284343	7.782
Other-Other	20161622	26.024	16932523	26.076	10522884	24.933

Alpine

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	243.73	24.826	177.25	29.108	177.25	40.688
Home-Other	296.88	30.239	268.16	44.036	104.84	24.066
Home-Shop	214.44	21.842	83.18	13.660	73.19	16.800
Work-Other	36.06	3.672	36.06	5.921	36.06	8.276
Other-Other	190.67	19.421	44.31	7.276	44.31	10.171

Amador

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	8802	12.576	8571	13.649	7311	15.821
Home-Other	24741	35.349	19754	31.460	12899	27.914
Home-Shop	10644	15.209	10159	16.179	7549	16.336
Work-Other	3862	5.519	3862	6.151	3533	7.645
Other-Other	21940	31.348	20446	32.561	14918	32.283

AMBAG

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	239486	14.865	222731	17.160	192591	22.424

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Other	670868	41.641	494851	38.124	270081	31.446
Home-Shop	203669	12.642	178766	13.772	130553	15.201
Work-Other	75851	4.708	66621	5.133	60320	7.023
Other-Other	421210	26.145	335034	25.812	205328	23.907

Butte

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	47501	11.23	44531	11.898	41319	16.938
Home-Other	166732	39.40	144068	38.494	74838	30.679
Home-Shop	68499	16.19	57092	15.255	35461	14.537
Work-Other	26953	6.37	24660	6.589	22679	9.297
Other-Other	113456	26.81	103909	27.764	69641	28.549

Calaveras

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	12001	14.339	12001	16.067	11774	21.528
Home-Other	30066	35.923	24685	33.047	14771	27.008
Home-Shop	12599	15.054	12599	16.868	11430	20.899
Work-Other	4477	5.349	3982	5.332	3132	5.727
Other-Other	24553	29.335	21428	28.687	13584	24.837

Colusa

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	6907	14.635	6787	15.80	4840	19.216
Home-Other	23503	49.802	19742	45.95	10278	40.806
Home-Shop	2749	5.825	2715	6.32	2213	8.787
Work-Other	4890	10.362	4890	11.38	2958	11.743
Other-Other	9144	19.376	8828	20.55	4898	19.447

Del Norte

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	6401	13.243	6372	14.551	6131	18.598
Home-Other	18854	39.006	15405	35.177	9356	28.383
Home-Shop	6773	14.013	6773	15.467	5267	15.979
Work-Other	3412	7.059	3054	6.974	2722	8.257
Other-Other	12895	26.679	12188	27.831	9489	28.784

Fresno

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	261903	14.104	247455	15.365	227695	22.232
Home-Other	819412	44.127	667512	41.447	340583	33.255
Home-Shop	206587	11.125	171727	10.663	108023	10.548
Work-Other	105171	5.664	100631	6.248	98598	9.627
Other-Other	463882	24.981	423185	26.276	249258	24.338

Glenn

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	9421	19.340	9065	20.886	8662	30.062
Home-Other	21230	43.581	17478	40.271	7909	27.449
Home-Shop	4482	9.201	4161	9.588	2652	9.205
Work-Other	2586	5.308	2474	5.701	2405	8.347
Other-Other	10995	22.570	10223	23.554	7185	24.937

Humboldt

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	45108	15.494	39620	16.406	37393	20.907
Home-Other	100407	34.489	80121	33.178	48687	27.221
Home-Shop	52656	18.087	42036	17.407	35128	19.641
Work-Other	13127	4.509	12265	5.079	11093	6.202
Other-Other	79828	27.420	67447	27.930	46555	26.029

Inyo

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	5887	15.11	4703	14.67	4663	18.53
Home-Other	14132	36.27	11270	35.14	6850	27.22
Home-Shop	4338	11.13	3319	10.35	2825	11.23
Work-Other	4767	12.23	4529	14.12	4493	17.85
Other-Other	9844	25.26	8248	25.72	6335	25.17

Kern

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	213973	13.613	200931	15.111	179297	21.204
Home-Other	600853	38.228	471838	35.485	243650	28.815
Home-Shop	264682	16.840	223468	16.806	136608	16.156
Work-Other	79042	5.029	78013	5.867	69156	8.179
Other-Other	413222	26.290	355419	26.730	216853	25.646

Kings

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	34362	11.656	32043	13.123	24154	18.563
Home-Other	136136	46.179	105745	43.308	50931	39.142
Home-Shop	61683	20.924	49168	20.137	19946	15.329
Work-Other	6298	2.136	6298	2.579	4911	3.774
Other-Other	56320	19.104	50916	20.853	30176	23.192

Lake

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	16461	12.395	16255	13.880	15762	19.553
Home-Other	46323	34.879	41841	35.727	21368	26.506
Home-Shop	22348	16.827	16264	13.887	14689	18.221
Work-Other	8674	6.531	8674	7.407	5532	6.863
Other-Other	39004	29.368	34080	29.100	23263	28.857

Lassen

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	8008	14.909	7193	14.298	6240	18.185
Home-Other	14331	26.681	13136	26.113	7577	22.081
Home-Shop	6430	11.972	6430	12.782	4687	13.659
Work-Other	2775	5.166	1764	3.507	1650	4.808
Other-Other	22168	41.272	21783	43.300	14160	41.267

Madera

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	40313	14.101	33984	14.377	28892	20.378
Home-Other	140212	49.045	106745	45.158	48556	34.247
Home-Shop	42572	14.891	40355	17.072	26867	18.949
Work-Other	9815	3.433	9714	4.109	9403	6.632
Other-Other	52971	18.529	45585	19.284	28065	19.795

Mariposa

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	3566	13.15	3372	13.524	3022.3	19.802
Home-Other	9657	35.60	8359	33.523	4417.3	28.941
Home-Shop	3985	14.69	3769	15.114	2593.1	16.990
Work-Other	1044	3.85	1027	4.117	545.2	3.572
Other-Other	8873	32.71	8408	33.721	4685.0	30.695

Mendocino

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	30944	13.802	30222	16.265	27234	21.64
Home-Other	72513	32.344	58994	31.749	28223	22.43
Home-Shop	38065	16.979	35359	19.029	24509	19.48
Work-Other	19177	8.554	17200	9.257	16712	13.28
Other-Other	63493	28.321	44039	23.701	29145	23.16

Merced

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	51795	9.676	49238	10.163	46590	16.385
Home-Other	213265	39.842	179156	36.979	82304	28.946
Home-Shop	84683	15.820	79840	16.479	52852	18.588
Work-Other	17459	3.262	16882	3.485	16058	5.647
Other-Other	168081	31.400	159365	32.894	86537	30.434

Modoc

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	2206.9	11.829	2094.8	12.065	1953.9	18.148
Home-Other	9284.3	49.766	8505.6	48.988	4558.0	42.335
Home-Shop	2146.8	11.507	1807.3	10.409	1438.6	13.362
Work-Other	490.3	2.628	490.3	2.824	485.9	4.514
Other-Other	4527.7	24.270	4464.5	25.714	2329.9	21.641

Mono

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	3584.7	14.812	3055.2	16.564	3055.2	20.203
Home-Other	10546.5	43.578	6156.3	33.376	3827.3	25.309
Home-Shop	4070.2	16.818	4070.2	22.067	3611.8	23.884
Work-Other	421.2	1.741	421.2	2.284	421.2	2.786
Other-Other	5579.0	23.052	4742.2	25.710	4206.8	27.818

MTC

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	2445098	12.084	2131143	13.192	1947584	18.401
Home-Other	8112451	40.092	6311371	39.068	3340335	31.560
Home-Shop	2495060	12.331	2109249	13.057	1542353	14.572
Work-Other	1435326	7.093	1071593	6.633	901337	8.516
Other-Other	5746521	28.400	4531289	28.049	2852555	26.951

Nevada

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	29072	14.426	27184	14.816	24030	17.59
Home-Other	68948	34.214	61174	33.343	36370	26.63
Home-Shop	29793	14.784	28870	15.735	25721	18.83
Work-Other	12974	6.438	9998	5.449	9520	6.97
Other-Other	60730	30.136	56246	30.656	40934	29.97

Plumas

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	5796	13.89	5529	14.535	5051	20.083
Home-Other	14654	35.12	12744	33.503	6479	25.758
Home-Shop	7048	16.89	6950	18.269	5903	23.469
Work-Other	2036	4.88	1880	4.942	1743	6.928
Other-Other	12187	29.21	10937	28.751	5977	23.763

SACOG

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	601811	13.249	551769	13.764	511630	19.138
Home-Other	1994178	43.902	1709037	42.631	918026	34.340
Home-Shop	568070	12.506	502281	12.529	378095	14.143
Work-Other	254181	5.596	222711	5.555	194166	7.263
Other-Other	1124119	24.747	1023082	25.520	671404	25.115

San Joaquin

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	196240	14.936	189228	16.598	169462	22.982
Home-Other	547993	41.709	442664	38.829	234980	31.867
Home-Shop	197366	15.022	171919	15.080	117582	15.946
Work-Other	63048	4.799	61057	5.356	45963	6.233
Other-Other	309193	23.534	275163	24.136	169384	22.971

San Luis Obispo

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	93175	14.571	88034	15.379	84578	20.488
Home-Other	268405	41.973	234553	40.975	135425	32.805
Home-Shop	82470	12.897	75382	13.169	59683	14.458
Work-Other	41687	6.519	38654	6.753	34647	8.393
Other-Other	153735	24.041	135812	23.725	98479	23.856

SANDAG

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	743343	12.737	710778	14.09	647700	19.967
Home-Other	2401059	41.142	1966795	38.99	1024863	31.595
Home-Shop	787850	13.500	697330	13.82	482436	14.873
Work-Other	319889	5.481	283484	5.62	258463	7.968
Other-Other	1583951	27.141	1386081	27.48	830327	25.597

Santa Barbara

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	160252	15.800	141037	17.137	128695	21.859
Home-Other	421875	41.595	324621	39.445	197364	33.523
Home-Shop	114484	11.288	89887	10.922	70831	12.031
Work-Other	74786	7.374	60961	7.407	56796	9.647
Other-Other	242847	23.944	206467	25.088	135053	22.939

SCAG

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	4641225	13.924	4370424	15.544	4000268	22.128
Home-Other	14472695	43.419	11545708	41.065	5843610	32.325
Home-Shop	4314492	12.944	3771938	13.416	2646180	14.638
Work-Other	1714291	5.143	1507475	5.362	1333525	7.377
Other-Other	8190175	24.571	6920293	24.614	4253964	23.532

Shasta

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	51550	11.511	49668	12.080	45586	17.745
Home-Other	159048	35.517	148359	36.082	85748	33.378
Home-Shop	57675	12.879	52750	12.829	30693	11.948
Work-Other	18830	4.205	17945	4.365	17945	6.985
Other-Other	160712	35.888	142445	34.644	76926	29.944

Sierra

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	588.6	11.722	218.2	5.214	195.3	8.314
Home-Other	1305.8	26.004	1043.9	24.945	831.7	35.407
Home-Shop	937.7	18.674	737.5	17.623	472.1	20.096
Work-Other	198.9	3.961	198.9	4.753	186.4	7.937
Other-Other	1990.5	39.639	1986.3	47.465	663.5	28.246

Siskiyou

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	17493	18.076	15515	17.964	13186	23.248
Home-Other	33481	34.598	26542	30.731	15681	27.646
Home-Shop	10244	10.585	9911	11.475	6138	10.821
Work-Other	4691	4.847	4691	5.431	3905	6.885
Other-Other	30866	31.895	29709	34.398	17811	31.400

Stanislaus

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	122271	14.17	120599	15.795	109039	21.940
Home-Other	363370	42.10	304164	39.836	159533	32.099
Home-Shop	135684	15.72	123225	16.139	84097	16.921
Work-Other	34530	4.00	33795	4.426	31440	6.326
Other-Other	207335	24.02	181761	23.805	112890	22.714

Tehama

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	19155	15.199	18622	16.321	17197	21.234
Home-Other	51545	40.899	42395	37.155	24166	29.839
Home-Shop	18630	14.782	17616	15.439	12746	15.738
Work-Other	7336	5.821	7293	6.391	6788	8.382
Other-Other	29362	23.298	28175	24.693	20090	24.807

TMPO

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	9474	13.677	9174	14.941	9042	21.295
Home-Other	20789	30.010	16980	27.654	8726	20.551
Home-Shop	10812	15.608	9903	16.129	7105	16.734
Work-Other	4505	6.503	4260	6.937	3988	9.392
Other-Other	23693	34.202	21084	34.339	13599	32.028

Trinity

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	2734.8	11.043	2641.1	11.199	2489.3	15.370
Home-Other	7802.7	31.508	6724.8	28.516	3288.1	20.302
Home-Shop	4340.4	17.527	4330.0	18.361	3168.4	19.563
Work-Other	645.6	2.607	645.6	2.738	612.4	3.781
Other-Other	9240.9	37.315	9240.9	39.185	6637.5	40.983

Tulare

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	140010	16.047	135059	18.229	120122	24.603
Home-Other	336798	38.601	260142	35.111	142373	29.160
Home-Shop	116372	13.338	110599	14.928	64525	13.216
Work-Other	49091	5.626	42450	5.729	39668	8.125
Other-Other	230232	26.388	192653	26.002	121556	24.897

Tuolumne

Trip Type	Total Trips	% Total	Person Trips	% Person Trips	Driver Trips	% Driver Trips
Home-Work	15511	12.55	14018	12.156	13092	16.003
Home-Other	39556	32.01	36652	31.783	21477	26.253
Home-Shop	18508	14.98	17312	15.013	12452	15.221
Work-Other	7439	6.02	7020	6.087	6808	8.322
Other-Other	42556	34.44	40317	34.961	27981	34.202

```
# Load Hmisc for %nin% operator
library(Hmisc)
library(data.table)

weekend <- df$DOW %in% c("Sa", "Su")
weekday <- df$DOW %nin% c("Sa", "Su")

# Per CHTS 2001: * Person trips include all trips except walk, bicycle,
# airplane-commercial, Airplane private, and 'other' mode trips.  * Driver
# trips include automobile, pickup truck, RV, Sport Utility Vehicle, van,
# truck, and motorcycle/Moped driver trips
persontrip <- df$MODE %nin% c("Walk", "Bike", "Wheelchair / Mobility Scooter",
    "Other Non-Motorized", "Other Private Transit", "Other Bus", "Other Rail")
drivertrip <- df$MODE %in% c("Auto / Van / Truck Driver", "Motorcycle / Scooter / Moped")


cat(paste("\n\n## All Areas\n\n"))

# compute totals by type for each class of trip
xt.total <- xtabs(weight ~ triptype, data = df[weekday, ])
xt.person <- xtabs(weight ~ triptype, data = df[weekday & persontrip, ])
xt.driver <- xtabs(weight ~ triptype, data = df[weekday & drivertrip, ])

kable(data.table(`Trip Type` = names(xt.total), `Total Trips` = xt.total, `% Total` = xt.total/sum(xt.t
    100, `Person Trips` = xt.person, `% Person Trips` = xt.person/sum(xt.person) *
    100, `Driver Trips` = xt.driver, `% Driver Trips` = xt.driver/sum(xt.driver) *
    100))

cat("\n")


for (area in levels(df$AREA)) {
```

```
    inarea <- df$AREA == area

    # compute totals by type for each class of trip
    xt.total <- xtabs(weight ~ triptype, data = df[weekday & inarea, ])
    xt.person <- xtabs(weight ~ triptype, data = df[weekday & persontrip & inarea,
        ])
    xt.driver <- xtabs(weight ~ triptype, data = df[weekday & drivertrip & inarea,
        ])

    cat(paste("\n\n##", area, "\n\n"))

    kable(data.table(`Trip Type` = names(xt.total), `Total Trips` = xt.total,
        `% Total` = xt.total/sum(xt.total) * 100, `Person Trips` = xt.person,
        `% Person Trips` = xt.person/sum(xt.person) * 100, `Driver Trips` = xt.driver,
        `% Driver Trips` = xt.driver/sum(xt.driver) * 100))

    cat("\n")
}
```

Data cleaning

More to come. . .

This is a living document that we are actively improving. Check back often. . .