Analysis of Simulated Vance County EMS Response Data

September 18, 2023

In what follows, we carry out a basic analysis of the simulated data to test the R library mapsapi's Google API interface.

1 Input Data; Basic Data Processing

Here we import the data elements, carry out some basic data processing and display a few summaries of the data set:

```
rm(list=ls()) ## Completely clear the workspace.
opts_chunk$set(fig.path='./figs/',cache.path='./cache/')
library(mapsapi)
library(mgcv)
library(xtable)
library(lme4)
api.key<-scan("../api.key",what=" ")
```

```
##x<-read.csv("VanceModelDataset.csv")
x<-read.csv("VanceMockData1.csv")
head(x)

##    REF.GRID DISPATCH.PRIORITY.NAME REF.GPS.LAT REF.GPS.LON BASE.NAME VEH.GRID
## 1   3 South              Emergency     36.3085    -78.4563 Company 9  Medic 5
## 2 2 Central              Emergency     36.3306    -78.4040 Company 9  Medic 6
## 3 2 Central              Emergency     36.3335    -78.4399 Company 9  Medic 1
## 4 2 Central              Emergency     36.3351    -78.4410 Company 9  Medic 5
## 5 2 Central          Non Emergency     36.3401    -78.4017 Company 9  Medic 6
## 6 2 Central              Emergency     36.3315    -78.3929 Company 9  Medic 1
##            VEHCGPS             DT.DISP          DT.ENROUTE           DT.ARRIVE
## 1 36.345, -78.3905 01/01/1789 06:46:00 01/01/1789 06:46:00 01/01/1789 06:52:00
## 2 36.345, -78.3905 01/01/1789 08:30:00 01/01/1789 08:30:00 01/01/1789 08:34:00
## 3 36.345, -78.3905 01/01/1789 10:22:00 01/01/1789 10:22:00 01/01/1789 10:27:00
## 4 36.345, -78.3905 01/01/1789 11:38:00 01/01/1789 11:38:00 01/01/1789 11:44:00
## 5 36.345, -78.3905 01/01/1789 12:33:00 01/01/1789 12:33:00 01/01/1789 12:37:00
## 6 36.345, -78.3905 01/01/1789 14:18:00 01/01/1789 14:18:00 01/01/1789 14:22:00
##              DT.LVREF           DT.ARVREC        DT.AVAILABLE
## 1 01/01/1789 07:07:00 01/01/1789 07:13:00 01/01/1789 07:32:00
## 2 01/01/1789 08:39:00 01/01/1789 08:46:00 01/01/1789 09:00:00
## 3 01/01/1789 10:36:00 01/01/1789 10:39:00 01/01/1789 10:54:00
## 4                                         01/01/1789 12:08:00
```

```
## 5 01/01/1789 12:38:00 01/01/1789 12:45:00 01/01/1789 12:52:00
## 6 01/01/1789 14:38:00 01/01/1789 14:47:00 01/01/1789 15:11:00
##               REC.NAME
## 1 Maria Parham Hospital
## 2 Maria Parham Hospital
## 3 Maria Parham Hospital
## 4
## 5 Maria Parham Hospital
## 6 Maria Parham Hospital
```

```
summary(x)
```

```
##    REF.GRID         DISPATCH.PRIORITY.NAME  REF.GPS.LAT     REF.GPS.LON
##  Length:499         Length:499             Min.   :35.96   Min.   :-78.81
##  Class :character   Class :character       1st Qu.:36.32   1st Qu.:-78.43
##  Mode  :character   Mode  :character       Median :36.33   Median :-78.40
##                                            Mean   :36.33   Mean   :-78.41
##                                            3rd Qu.:36.34   3rd Qu.:-78.39
##                                            Max.   :36.52   Max.   :-78.20
##                                            NA's   :2       NA's   :2
##   BASE.NAME           VEH.GRID           VEHCGPS            DT.DISP
##  Length:499         Length:499         Length:499         Length:499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   DT.ENROUTE         DT.ARRIVE           DT.LVREF           DT.ARVREC
##  Length:499         Length:499         Length:499         Length:499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  DT.AVAILABLE         REC.NAME
##  Length:499         Length:499
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

1.1 Get Station Coordinates

```
gps.cent<-unique(x$VEHCGPS[x$BASE.NAME=="Company 9"])
gps.south<-unique(x$VEHCGPS[x$BASE.NAME=="Company 1"])
gps.centN<-as.numeric(strsplit(gps.cent,",")[[1]])
```

```r
gps.southN<-as.numeric(strsplit(gps.south,",")[[1]])
## First option for a north station
gps.northNN<-c(36.430596, -78.431689) ##NN=Near North
##Second option for north station
gps.northFN<-c(36.495537112943886,-78.42090194629898) ##FN=Far North
## GPS coordinates of Maria Parham Hospital
gps.hospital<-c(36.33089064918619, -78.44930886477614)
```

Destination hospital coordinates:

```r
x$REC.LON<-rep(NA,nrow(x))
x$REC.LAT<-rep(NA,nrow(x))
x$REC.LON[x$REC.NAME=="Maria Parham Hospital"]<-(-78.44930886477614)
x$REC.LAT[x$REC.NAME=="Maria Parham Hospital"]<-(36.33089064918619)
x$REC.LON[x$REC.NAME=="Granville Medical Center"]<-(-78.59367173834997)
x$REC.LAT[x$REC.NAME=="Granville Medical Center"]<-(36.33043072571129)
x$REC.LON[x$REC.NAME=="Duke Health Duke University Medical Center"]<-(-78.93687608445487)
x$REC.LAT[x$REC.NAME=="Duke Health Duke University Medical Center"]<-(36.00643609468812)
```

Drop cases with missing call GPS coordinates:

```r
table(is.na(x$REF.GPS.LON))

##
## FALSE  TRUE
##   497     2

x<-x[!is.na(x$REF.GPS.LON),]
```

1.2 Format Time Character Strings as Times

1.2.1 Google Needs Current/Future Times

Change the year from 1789 to 2024.

```r
x$DT.DISP<-sub("1789","2024",x$DT.DISP)
x$DT.ENROUTE<-sub("1789","2024",x$DT.ENROUTE)
x$DT.ARRIVE<-sub("1789","2024",x$DT.ARRIVE)
x$DT.LVREF<-sub("1789","2024",x$DT.LVREF)
x$DT.ARVREC<-sub("1789","2024",x$DT.ARVREC)
x$DT.AVAILABLE<-sub("1789","2024",x$DT.AVAILABLE)
```

1.2.2 Google Needs Times in POSIXct Format

Convert times importated as character strings to times formated as R POSIX values.

```r
x$DT.DISP<-strptime(x$DT.DISP,format="%m/%d/%Y %H:%M:%S",tz="EST")
x$DT.ENROUTE<-strptime(x$DT.ENROUTE,format="%m/%d/%Y %H:%M:%S",tz="EST")
x$DT.ARRIVE<-strptime(x$DT.ARRIVE,format="%m/%d/%Y %H:%M:%S",tz="EST")
x$DT.LVREF<-strptime(x$DT.LVREF,format="%m/%d/%Y %H:%M:%S",tz="EST")
x$DT.ARVREC<-strptime(x$DT.ARVREC,format="%m/%d/%Y %H:%M:%S",tz="EST")
x$DT.AVAILABLE<-strptime(x$DT.AVAILABLE,format="%m/%d/%Y %H:%M:%S",tz="EST")
```

2 Estimate Response Travel Times from Each Station

Compute travel times between the two existing and two proposed station locations and each destination. Do so assuming the pessimistic, best guess and optimistic traffic assumptions, in turn. In addition, save the 'green light' duration and distance.

2.1 Best Guess Scenario

```
travelTimeBG<-NULL
distance<-NULL
durationGL<-NULL
for (i in 1:nrow(x)){
    api.out<-mp_matrix(
        origins = rbind(gps.southN[c(2,1)],gps.centN[c(2,1)],
                        gps.northNN[c(2,1)],gps.northFN[c(2,1)]),
        destinations = cbind(x$REF.GPS.LON,x$REF.GPS.LAT)[i,],
        mode="driving",
        traffic_model="best_guess",
        departure_time=as.POSIXct(x$DT.ENROUTE[i]),  ##as POSIXct
        ##departure_time=Sys.time() + as.difftime(4, units = "hours"),
        key = api.key,
        quiet = TRUE)
    times.out<-mp_get_matrix(api.out,
                             value = "duration_in_traffic_s")
    gl.out<-mp_get_matrix(api.out,
                             value = "duration_s")
    dist.out<-mp_get_matrix(api.out,
                             value = "distance_m")
    travelTimeBG<-rbind(travelTimeBG,matrix(times.out,nrow=1))
    distance<-rbind(distance,matrix(dist.out,nrow=1))
    durationGL<-rbind(durationGL,matrix(gl.out,nrow=1))
    Sys.sleep(0.5)
}
```

Google API computed response travel times from each station, measured in seconds, where 'So' refers to the south station, 'Ce' refers to the central, 'NN' to the proposed near north station and 'FN' to the proposed far north station. 'BG' refers to the 'best guess' scenario.

```
colnames(travelTimeBG)<-c("eTT.BG.So","eTT.BG.Ce","eTT.BG.NN","eTT.BG.FN")
colnames(distance)<-c("Dist.So","Dist.Ce","Dist.NN","Dist.FN")
colnames(durationGL)<-c("eTT.GL.So","eTT.GL.Ce","eTT.GL.NN","eTT.GL.FN")
head(travelTimeBG)

##      eTT.BG.So eTT.BG.Ce eTT.BG.NN eTT.BG.FN
## [1,]       539       406       805      1173
## [2,]       549       220       633       993
## [3,]       752       346       743      1127
## [4,]       770       306       712      1085
## [5,]       766       181       670      1036
## [6,]       722       235       815      1186

summary(travelTimeBG)
```

```
##    eTT.BG.So        eTT.BG.Ce        eTT.BG.NN        eTT.BG.FN
##  Min.   :  38.0   Min.   :   9.0   Min.   :  13.0   Min.   :  11
##  1st Qu.: 491.0   1st Qu.: 280.0   1st Qu.: 679.0   1st Qu.:1054
##  Median : 604.0   Median : 385.0   Median : 779.0   Median :1150
##  Mean   : 661.3   Mean   : 440.2   Mean   : 816.7   Mean   :1177
##  3rd Qu.: 737.0   3rd Qu.: 544.0   3rd Qu.: 937.0   3rd Qu.:1318
##  Max.   :2890.0   Max.   :2412.0   Max.   :2900.0   Max.   :3284
```

2.2 Pessimistic Scenario

```r
travelTimePe<-NULL
for (i in 1:nrow(x)){
    api.out<-mp_matrix(
        origins = rbind(gps.southN[c(2,1)],gps.centN[c(2,1)],
                        gps.northNN[c(2,1)],gps.northFN[c(2,1)]),
        destinations = cbind(x$REF.GPS.LON,x$REF.GPS.LAT)[i,],
        mode="driving",
        traffic_model="pessimistic",
        departure_time=as.POSIXct(x$DT.ENROUTE[i]),  ##as POSIXct
        ##departure_time=Sys.time() + as.difftime(4, units = "hours"),
        key = api.key,
        quiet = TRUE)
    times.out<-mp_get_matrix(api.out,
                              value = "duration_in_traffic_s")
    travelTimePe<-rbind(travelTimePe,matrix(times.out,nrow=1))
    Sys.sleep(0.5)
}
```

Google API computed response travel times from each station, measured in seconds; 'Pe' refers to the 'pessimistic' scenario.

```r
colnames(travelTimePe)<-c("eTT.Pe.So","eTT.Pe.Ce","eTT.Pe.NN","eTT.Pe.FN")
head(travelTimePe)

##      eTT.Pe.So eTT.Pe.Ce eTT.Pe.NN eTT.Pe.FN
## [1,]       616       440       859      1267
## [2,]       650       251       689      1076
## [3,]       918       384       796      1217
## [4,]      1026       363       784      1193
## [5,]       965       220       725      1123
## [6,]       890       250       963      1358

summary(travelTimePe)

##    eTT.Pe.So        eTT.Pe.Ce      eTT.Pe.NN        eTT.Pe.FN
##  Min.   :  39.0   Min.   :   9   Min.   :  15.0   Min.   :  12
##  1st Qu.: 556.0   1st Qu.: 313   1st Qu.: 730.0   1st Qu.:1128
##  Median : 695.0   Median : 434   Median : 866.0   Median :1261
##  Mean   : 756.8   Mean   : 488   Mean   : 896.8   Mean   :1279
##  3rd Qu.: 891.0   3rd Qu.: 595   3rd Qu.:1042.0   3rd Qu.:1429
##  Max.   :3174.0   Max.   :2680   Max.   :3204.0   Max.   :3655
```

2.3 Optimistic Scenario

```
travelTimeOp<-NULL
for (i in 1:nrow(x)){
    api.out<-mp_matrix(
        origins = rbind(gps.southN[c(2,1)],gps.centN[c(2,1)],
                        gps.northNN[c(2,1)],gps.northFN[c(2,1)]),
        destinations = cbind(x$REF.GPS.LON,x$REF.GPS.LAT)[i,],
        mode="driving",
        traffic_model="optimistic",
        departure_time=as.POSIXct(x$DT.ENROUTE[i]),  ##as POSIXct
        ##departure_time=Sys.time() + as.difftime(4, units = "hours"),
        key = api.key,
        quiet = TRUE)
    times.out<-mp_get_matrix(api.out,
                             value = "duration_in_traffic_s")
    travelTimeOp<-rbind(travelTimeOp,matrix(times.out,nrow=1))
    Sys.sleep(0.5)
}
```

Google API computed response travel times from each station, measured in seconds; 'Op' refers to the 'optimal' scenario.

```
colnames(travelTimeOp)<-c("eTT.Op.So","eTT.Op.Ce","eTT.Op.NN","eTT.Op.FN")
head(travelTimeOp)

##      eTT.Op.So eTT.Op.Ce eTT.Op.NN eTT.Op.FN
## [1,]       507       372       758      1097
## [2,]       508       210       587       933
## [3,]       699       343       728      1090
## [4,]       679       283       671      1032
## [5,]       728       187       642       994
## [6,]       668       243       753      1124

summary(travelTimeOp)

##    eTT.Op.So        eTT.Op.Ce      eTT.Op.NN        eTT.Op.FN
##  Min.   :  38.0   Min.   :   9   Min.   :  13.0   Min.   :  10
##  1st Qu.: 483.0   1st Qu.: 277   1st Qu.: 653.0   1st Qu.:1014
##  Median : 586.0   Median : 370   Median : 749.0   Median :1113
##  Mean   : 637.1   Mean   : 428   Mean   : 780.3   Mean   :1132
##  3rd Qu.: 697.0   3rd Qu.: 529   3rd Qu.: 899.0   3rd Qu.:1258
##  Max.   :2815.0   Max.   :2340   Max.   :2824.0   Max.   :3150
```

2.4 Observed Times

Observed times, measured in seconds:

```
## Duration from dispatch to clear
x$dispToClearTime<-difftime(x$DT.AVAILABLE,x$DT.DISP,units="secs")
## Duration from dispatch to enroute
```

```r
x$timeToEnroute<-difftime(x$DT.ENROUTE,x$DT.DISP,units="secs")
## Response Time, Station to Scene
x$observedTT<-difftime(x$DT.ARRIVE,x$DT.ENROUTE,units="secs")
## Duration on Scene
x$onSceneDur<-difftime(x$DT.LVREF,x$DT.ARRIVE,units="secs")
## Scene to Hospital Travel Time
x$toHospitalTT<-difftime(x$DT.ARVREC,x$DT.LVREF,units="secs")
## Duration at Hospital
x$atHospitalDur<-difftime(x$DT.AVAILABLE,x$DT.ARVREC,units="secs")
## Time from arriving at scene to clear
x$arriveToClearTime<-difftime(x$DT.AVAILABLE,x$DT.ARRIVE,units="secs")
```

3 Estimate Travel Times to Hospital

Compute travel times between the call locations and destination hospital under each of the traffic scenarios. Save green light times and distances.

3.1 Best Guess Traffic Model

```
travelTime2bg<-rep(NA,nrow(x))
hosp.GL<-rep(NA,nrow(x))
hosp.Dist<-rep(NA,nrow(x))
for (i in 1:nrow(x)){
  if (!is.na(x$REC.LON[i])){
    api.out2<-mp_matrix(
      origins = cbind(x$REF.GPS.LON,x$REF.GPS.LAT)[i,],
      destinations = cbind(x$REC.LON,x$REC.LAT)[i,],
      mode="driving",
      traffic_model="best_guess",
      departure_time=as.POSIXct(x$DT.LVREF[i]),  ##as POSIXct
      ##departure_time=Sys.time() + as.difftime(4, units = "hours"),
      key = api.key,
      quiet = TRUE
    )
    travelTime2bg[i]<-mp_get_matrix(api.out2,
                                  value = "duration_in_traffic_s")
    hosp.GL[i]<-mp_get_matrix(api.out2,
                              value = "duration_s")
    hosp.Dist[i]<-mp_get_matrix(api.out2,
                              value = "distance_m")
    Sys.sleep(0.5)
  }
}
```

3.2 Pessimistic Traffic Model

```
travelTime2pe<-rep(NA,nrow(x))
for (i in 1:nrow(x)){
  if (!is.na(x$REC.LON[i])){
    api.out2<-mp_matrix(
      origins = cbind(x$REF.GPS.LON,x$REF.GPS.LAT)[i,],
      destinations = cbind(x$REC.LON,x$REC.LAT)[i,],
      mode="driving",
      traffic_model="pessimistic",
      departure_time=as.POSIXct(x$DT.LVREF[i]),  ##as POSIXct
      ##departure_time=Sys.time() + as.difftime(4, units = "hours"),
      key = api.key,
      quiet = TRUE
    )
    travelTime2pe[i]<-mp_get_matrix(api.out2,
                                  value = "duration_in_traffic_s")
```

```
    Sys.sleep(0.5)
  }
}
```

3.3 Optimistic Traffic Model

```
travelTime2op<-rep(NA,nrow(x))
for (i in 1:nrow(x)){
  if (!is.na(x$REC.LON[i])){
    api.out2<-mp_matrix(
      origins = cbind(x$REF.GPS.LON,x$REF.GPS.LAT)[i,],
      destinations = cbind(x$REC.LON,x$REC.LAT)[i,],
      mode="driving",
      traffic_model="optimistic",
      departure_time=as.POSIXct(x$DT.LVREF[i]),  ##as POSIXct
      ##departure_time=Sys.time() + as.difftime(4, units = "hours"),
      key = api.key,
      quiet = TRUE
    )
    travelTime2op[i]<-mp_get_matrix(api.out2,
                                    value = "duration_in_traffic_s")
    Sys.sleep(0.5)
  }
}
```

4 Assemble Travel Time Estimates

```
apiEstimates<-cbind(distance,durationGL,
                    travelTimePe,travelTimeBG,travelTimeOp,
                    hosp.Dist,hosp.GL,
                    eTT.Pe.Hosp=travelTime2pe,
                    eTT.BG.Hosp=travelTime2bg,
                    eTT.Op.Hosp=travelTime2op)
head(apiEstimates)

##      Dist.So Dist.Ce Dist.NN Dist.FN eTT.GL.So eTT.GL.Ce eTT.GL.NN eTT.GL.FN
## [1,]    9258    8434   17426   25709       561       411       827      1198
## [2,]    7048    2422   12212   20495       578       234       635      1007
## [3,]   10969    5301   12540   20823       759       366       752      1124
## [4,]    8781    5068   12307   20590       734       298       685      1056
## [5,]    8967    1516   12228   20511       770       191       656      1027
## [6,]    7800    2298   13267   21550       696       245       795      1166
##      eTT.Pe.So eTT.Pe.Ce eTT.Pe.NN eTT.Pe.FN eTT.BG.So eTT.BG.Ce eTT.BG.NN
## [1,]       616       440       859      1267       539       406       805
## [2,]       650       251       689      1076       549       220       633
## [3,]       918       384       796      1217       752       346       743
## [4,]      1026       363       784      1193       770       306       712
```

```
## [5,]       965       220       725      1123       766       181       670
## [6,]       890       250       963      1358       722       235       815
##      eTT.BG.FN eTT.Op.So eTT.Op.Ce eTT.Op.NN eTT.Op.FN hosp.Dist hosp.GL
## [1,]      1173       507       372       758      1097      3151     309
## [2,]       993       508       210       587       933      6060     443
## [3,]      1127       699       343       728      1090      1372     219
## [4,]      1085       679       283       671      1032        NA      NA
## [5,]      1036       728       187       642       994      6076     447
## [6,]      1186       668       243       753      1124      8416     557
##      eTT.Pe.Hosp eTT.BG.Hosp eTT.Op.Hosp
## [1,]         300         271         267
## [2,]         489         404         393
## [3,]         222         178         208
## [4,]          NA          NA          NA
## [5,]         557         438         414
## [6,]         661         553         531
```

4.1 Parsing the API–Computed Column Names

- **So** indicates the existing south EMS station
- **Ce** indicates the existing central EMS station
- **NN** indicates the proposed near–north EMS station
- **FN** indicates the proposed far–north EMS station
- **Dist** indicates distance travelled in meters.
- **eTT** indicates an estimated travel time.
- **GL** is the "green light" distance.
- **Pe** is the pessimistic travel time in traffic.
- **BG** is the best–guess travel time in traffic.
- **Op** is the optimistic travel time in traffic.
- **Hosp** is the hospital used, if such a trip is made.

4.2 Export Data Set for EDA

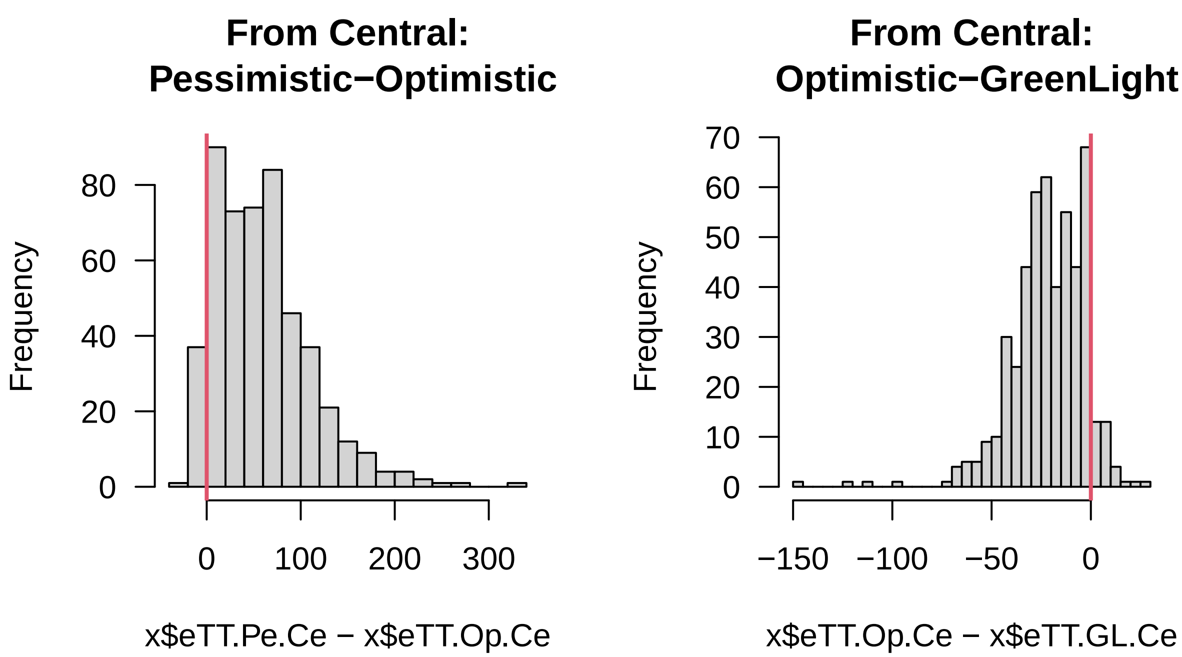
```
x<-cbind(x,apiEstimates)
save(x,file="emsData.RData")
gc(); save.image()
```

```
##           used  (Mb) gc trigger  (Mb) limit (Mb) max used  (Mb)
## Ncells 2156672 115.2    3797542 202.9         NA  3797542 202.9
## Vcells 4065674  31.1   10350180  79.0      16384 10348928  79.0
```

5 A Quick Look at Travel Times

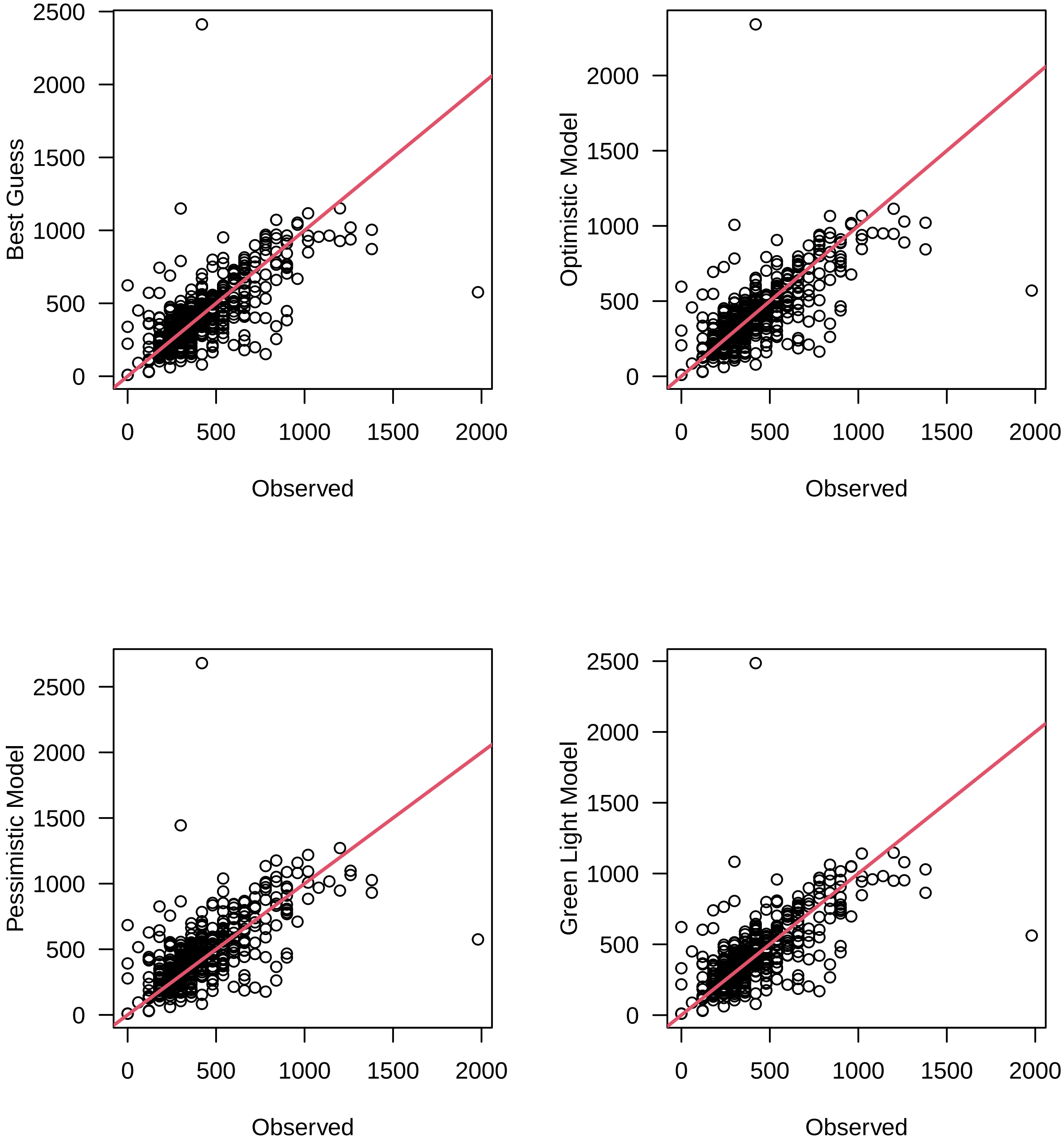
5.1 Ordered (?) Differences

```
par(mfrow=c(1,2))
hist(x$eTT.Pe.Ce - x$eTT.Op.Ce,nclass=25,las=1,cex=0.55,
     main="From Central:\n Pessimistic-Optimistic")
abline(v=0,col=2,lwd=2)
hist(x$eTT.Op.Ce - x$eTT.GL.Ce,nclass=25,las=1,cex=0.55,
     main="From Central:\n Optimistic-GreenLight")
abline(v=0,col=2,lwd=2)
```



5.2 Current Scenario (Scenario 0) Observed and Estimated Travel Times

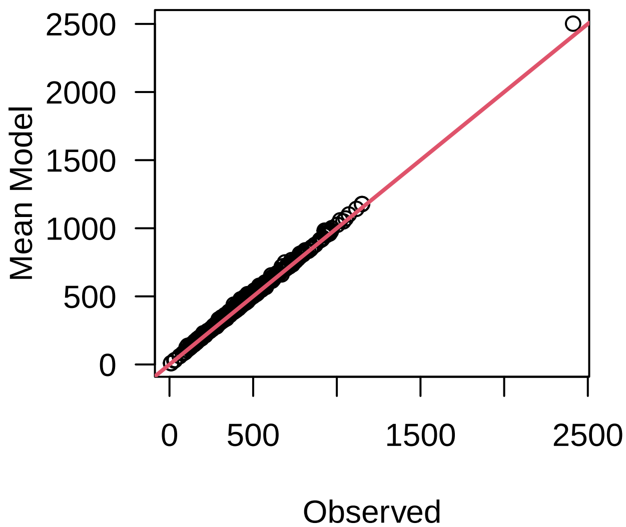
```
par(mfrow=c(2,2))
## Best guess travel time under current scenario
##   (dispatch same as observed):
x$eTT.BG.Scen0<-x$eTT.BG.Ce
x$eTT.BG.Scen0[x$BASE.NAME=="Company 1"]<-x$eTT.BG.So[x$BASE.NAME=="Company 1"]
x$eTT.BG.Scen0[is.na(x$observedTT)]<-NA
plot(x$observedTT,x$eTT.BG.Scen0,las=1,xlab="Observed",ylab="Best Guess")
abline(a=0,b=1,lwd=2,col=2)
## Optimistic travel time under current scenario
##   (dispatch same as observed):
x$eTT.Op.Scen0<-x$eTT.Op.Ce
x$eTT.Op.Scen0[x$BASE.NAME=="Company 1"]<-x$eTT.Op.So[x$BASE.NAME=="Company 1"]
x$eTT.Op.Scen0[is.na(x$observedTT)]<-NA
plot(x$observedTT,x$eTT.Op.Scen0,las=1,xlab="Observed",ylab="Optimistic Model")
abline(a=0,b=1,lwd=2,col=2)
## Pessimistic travel time under current scenario
##   (dispatch same as observed):
x$eTT.Pe.Scen0<-x$eTT.Pe.Ce
x$eTT.Pe.Scen0[x$BASE.NAME=="Company 1"]<-x$eTT.Pe.So[x$BASE.NAME=="Company 1"]
x$eTT.Pe.Scen0[is.na(x$observedTT)]<-NA
plot(x$observedTT,x$eTT.Pe.Scen0,las=1,xlab="Observed",ylab="Pessimistic Model")
abline(a=0,b=1,lwd=2,col=2)
## Green Light travel time under current scenario
##   (dispatch same as observed):
x$eTT.GL.Scen0<-x$eTT.GL.Ce
x$eTT.GL.Scen0[x$BASE.NAME=="Company 1"]<-x$eTT.GL.So[x$BASE.NAME=="Company 1"]
x$eTT.GL.Scen0[is.na(x$observedTT)]<-NA
plot(x$observedTT,x$eTT.GL.Scen0,las=1,xlab="Observed",ylab="Green Light Model")
abline(a=0,b=1,lwd=2,col=2)
```



Best Guess
Observed
Optimistic Model
Observed
Pessimistic Model
Observed
Green Light Model
Observed

5.3 'Best Guess' Times are Near The Mean of the Pessimistic and Optimistic (and Green Light) Values:

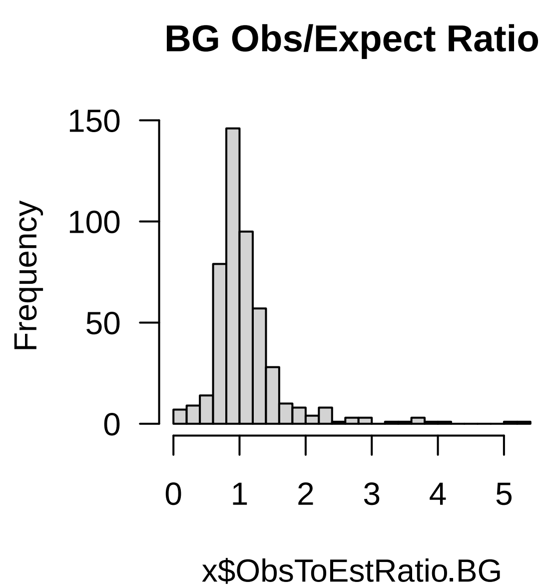
```
par(mfrow=c(1,1))
## Best Guess times are close to mean of Pe, Op and GL:
mu<-apply(x[,c("eTT.Op.Scen0","eTT.Pe.Scen0","eTT.GL.Scen0")],1,mean)
plot(x$eTT.BG.Scen0,mu,las=1,xlab="Observed",ylab="Mean Model")
abline(a=0,b=1,lwd=2,col=2)
```

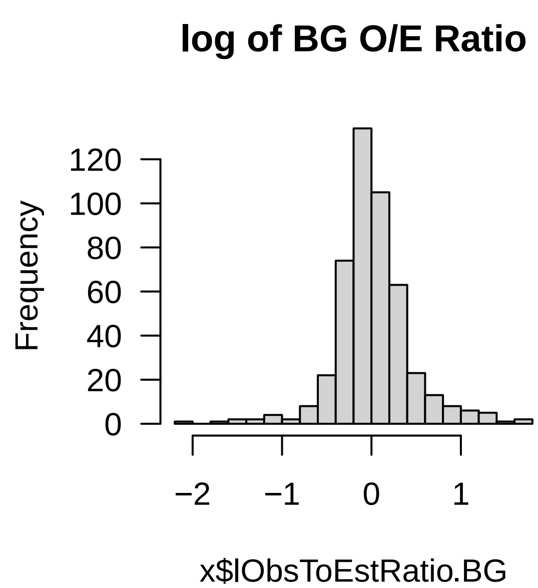


6 Do Observed and Estimated Travel Times Deviate Systematically?

6.1 Hour of Day, Day of Week and Observed–to–Expected Ratio

```
par(mfrow=c(1,2))
## extract day of week
table(x$DayOfWeek<-as.factor(weekdays(x$DT.DISP)))

##
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
##        62        76        63        60        74        66        96

## extract hour of day
table(x$HourOfDay<-as.factor(format(x$DT.DISP,format="%H")))

##
## 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23
##  9 16  8 10 17  6 12 12 23 25 22 32 29 30 24 31 25 29 27 31 33 16 19 11

x$nHourOfDay<-as.numeric(format(x$DT.DISP,format="%H"))
## predictions:
x$ObsToEstRatio.BG<-as.numeric(x$observedTT)/x$eTT.BG.Scen0
hist(x$ObsToEstRatio.BG,nclass=25, main="BG Obs/Expect Ratio",las=1)
x$lObsToEstRatio.BG<-log(x$ObsToEstRatio.BG)
## Protect against zero ratios:
x$lObsToEstRatio.BG[x$ObsToEstRatio.BG==0]<-NA
hist(x$lObsToEstRatio.BG,nclass=25, main="log of BG O/E Ratio",las=1)
```





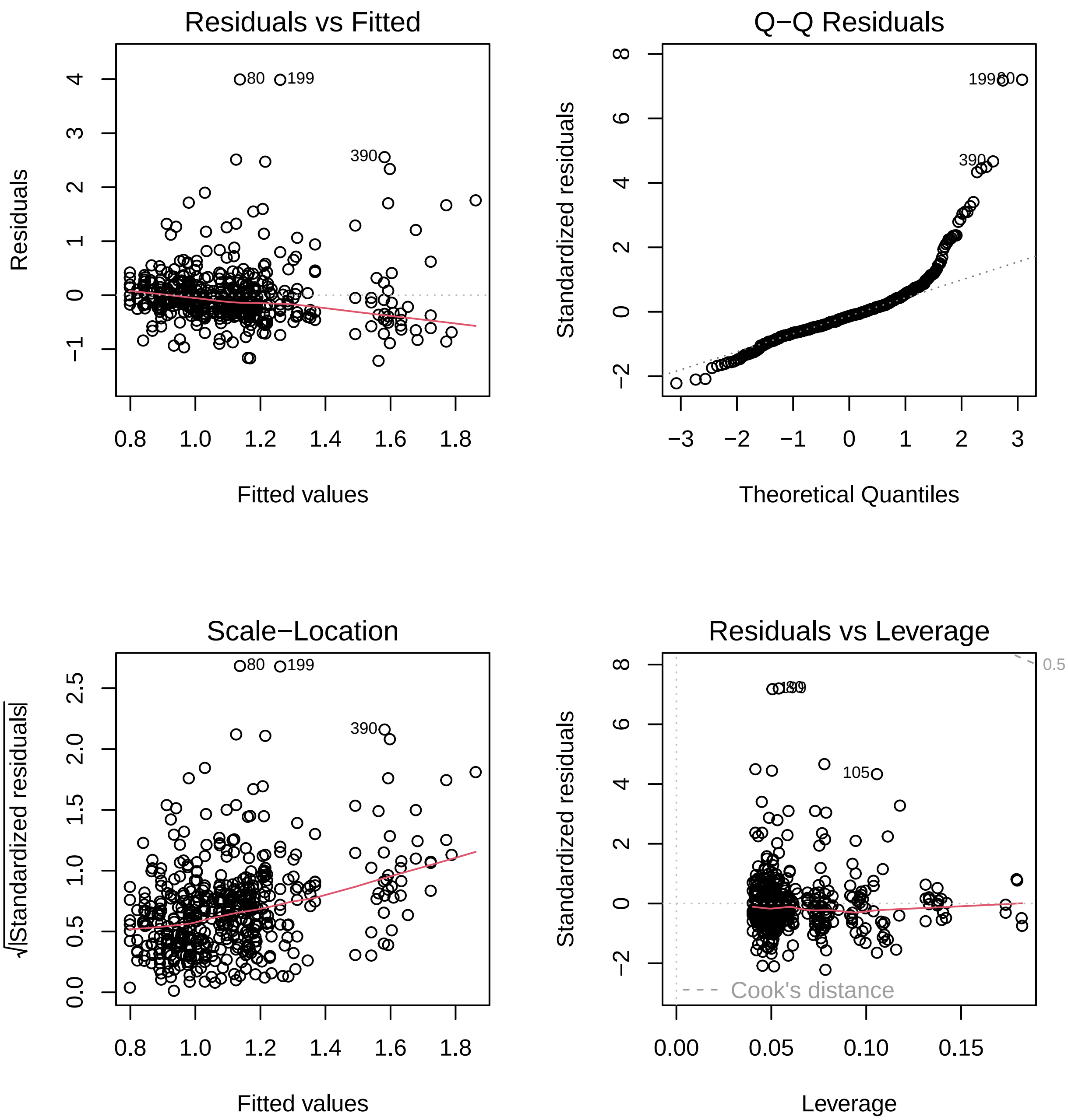
6.2 Is the O/E Ratio Predictable?

6.2.1 Additive Model

```
par(mfrow=c(2,2))
lm.add<-lm(ObsToEstRatio.BG~DayOfWeek+HourOfDay,data=x)  ## Additive Model
summary(lm.add)
```

```
##
## Call:
## lm(formula = ObsToEstRatio.BG ~ DayOfWeek + HourOfDay, data = x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2153 -0.2779 -0.0755  0.1367  3.9946
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.06060    0.21156   5.013  7.7e-07 ***
## DayOfWeekMonday       0.02434    0.10273   0.237   0.8128
## DayOfWeekSaturday     0.01854    0.10718   0.173   0.8627
## DayOfWeekSunday       0.20835    0.10920   1.908   0.0570 .
## DayOfWeekThursday     0.07019    0.10363   0.677   0.4985
## DayOfWeekTuesday      0.02970    0.10492   0.283   0.7772
## DayOfWeekWednesday -0.02216    0.09757  -0.227   0.8205
## HourOfDay01           0.50249    0.24788   2.027   0.0432 *
## HourOfDay02           0.09452    0.28682   0.330   0.7419
## HourOfDay03           0.59276    0.27172   2.182   0.0297 *
## HourOfDay04           0.22271    0.24853   0.896   0.3707
## HourOfDay05           0.04617    0.30859   0.150   0.8811
## HourOfDay06           0.14384    0.26137   0.550   0.5824
## HourOfDay07           0.07994    0.26224   0.305   0.7606
## HourOfDay08          -0.08090    0.23494  -0.344   0.7307
## HourOfDay09           0.09898    0.23319   0.424   0.6714
## HourOfDay10          -0.14409    0.23667  -0.609   0.5429
## HourOfDay11          -0.23974    0.22868  -1.048   0.2950
## HourOfDay12           0.01704    0.23080   0.074   0.9412
## HourOfDay13          -0.12666    0.22895  -0.553   0.5804
## HourOfDay14          -0.11383    0.23638  -0.482   0.6303
## HourOfDay15          -0.05597    0.22906  -0.244   0.8071
## HourOfDay16           0.13004    0.23337   0.557   0.5776
## HourOfDay17           0.07635    0.22954   0.333   0.7396
## HourOfDay18          -0.21696    0.23305  -0.931   0.3524
## HourOfDay19           0.04044    0.22766   0.178   0.8591
## HourOfDay20          -0.19529    0.22705  -0.860   0.3902
## HourOfDay21          -0.10476    0.25103  -0.417   0.6766
## HourOfDay22          -0.05763    0.24723  -0.233   0.8158
## HourOfDay23           0.51880    0.26661   1.946   0.0523 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5706 on 451 degrees of freedom
```

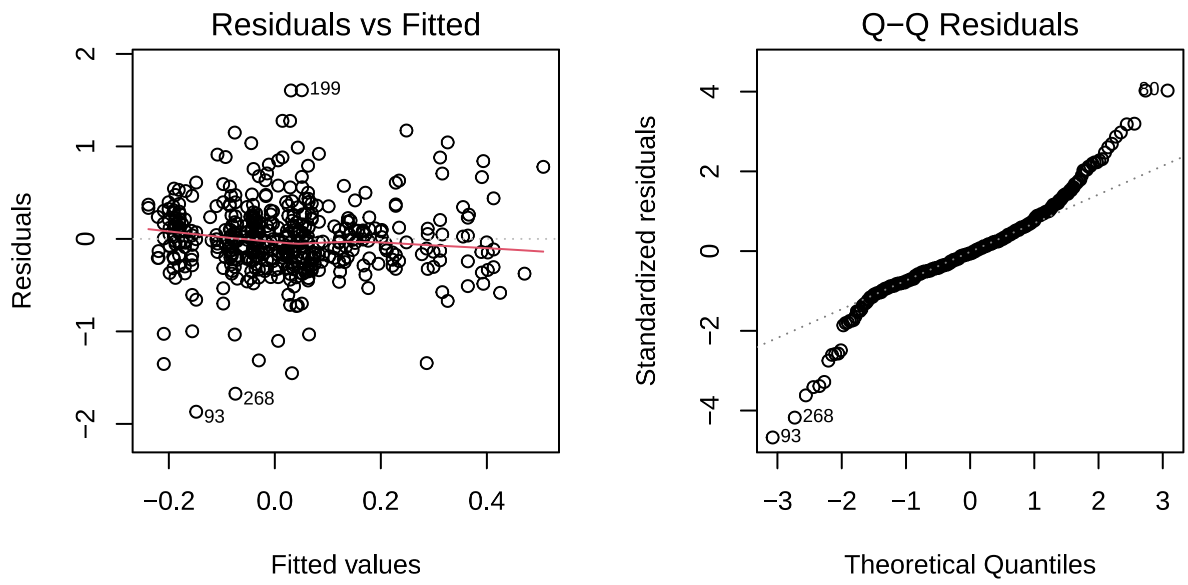
```
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.1181,Adjusted R-squared:  0.06134
## F-statistic: 2.082 on 29 and 451 DF,  p-value: 0.0009935

plot(lm.add)
```

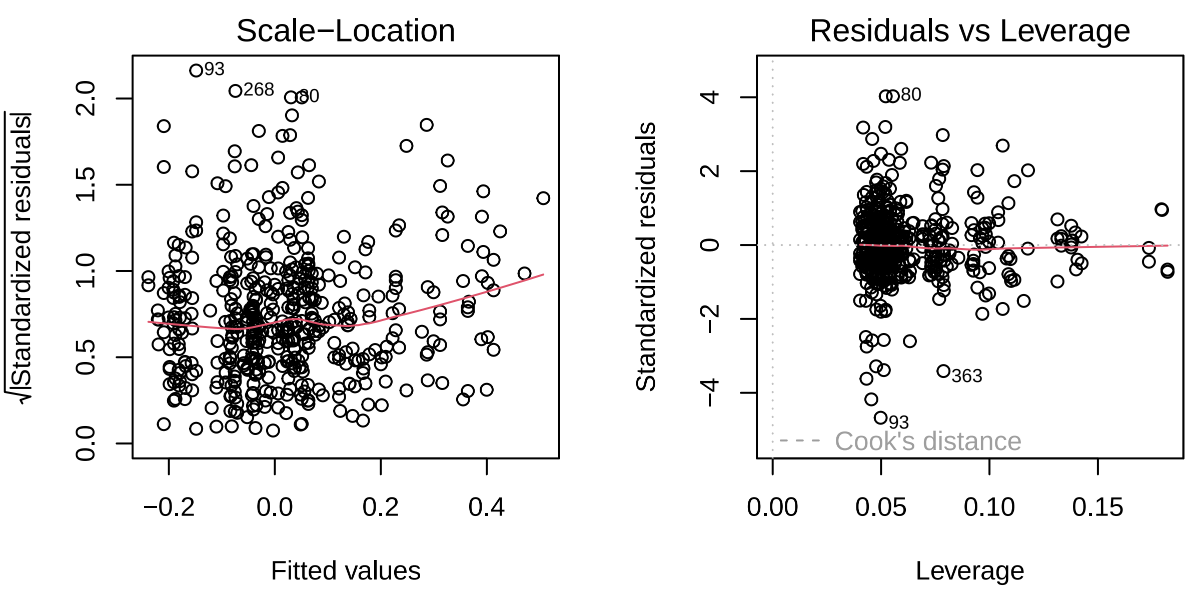


6.2.2 Multiplicative Model

```
par(mfrow=c(2,2))
lm.mult<-lm(lObsToEstRatio.BG~DayOfWeek+HourOfDay,data=x) ## Multiplicative Model
summary(lm.mult)

##
## Call:
## lm(formula = lObsToEstRatio.BG ~ DayOfWeek + HourOfDay, data = x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86873 -0.20082 -0.02294  0.18374  1.60724
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.0508137  0.1521627   0.334   0.7386
## DayOfWeekMonday    -0.0090055  0.0743145  -0.121   0.9036
## DayOfWeekSaturday  -0.0381276  0.0780076  -0.489   0.6252
## DayOfWeekSunday     0.1069743  0.0788851   1.356   0.1758
## DayOfWeekThursday   0.0129813  0.0748774   0.173   0.8624
## DayOfWeekTuesday    0.0254561  0.0758157   0.336   0.7372
## DayOfWeekWednesday  0.0020112  0.0707099   0.028   0.9773
## HourOfDay01         0.2358584  0.1781567   1.324   0.1862
## HourOfDay02         0.0901809  0.2061474   0.437   0.6620
## HourOfDay03         0.3491825  0.1952881   1.788   0.0744 .
## HourOfDay04         0.1584568  0.1786324   0.887   0.3755
## HourOfDay05         0.0378071  0.2217762   0.170   0.8647
## HourOfDay06         0.1256303  0.1878514   0.669   0.5040
## HourOfDay07        -0.0009544  0.1884785  -0.005   0.9960
## HourOfDay08        -0.0951928  0.1688550  -0.564   0.5732
## HourOfDay09         0.0705952  0.1676122   0.421   0.6738
## HourOfDay10        -0.1346414  0.1701022  -0.792   0.4291
## HourOfDay11        -0.2334030  0.1649266  -1.415   0.1577
## HourOfDay12        -0.0063150  0.1659223  -0.038   0.9697
## HourOfDay13        -0.1610767  0.1652794  -0.975   0.3303
## HourOfDay14        -0.0930202  0.1708288  -0.545   0.5864
## HourOfDay15        -0.0939013  0.1646360  -0.570   0.5687
## HourOfDay16        -0.0128070  0.1687353  -0.076   0.9395
## HourOfDay17        -0.0203198  0.1655461  -0.123   0.9024
## HourOfDay18        -0.2513236  0.1675021  -1.500   0.1342
## HourOfDay19        -0.0272962  0.1636264  -0.167   0.8676
## HourOfDay20        -0.2321251  0.1631886  -1.422   0.1556
## HourOfDay21        -0.0880762  0.1804174  -0.488   0.6257
## HourOfDay22        -0.0873313  0.1776765  -0.492   0.6233
## HourOfDay23         0.3136439  0.1916311   1.637   0.1024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.41 on 446 degrees of freedom
##   (21 observations deleted due to missingness)
```

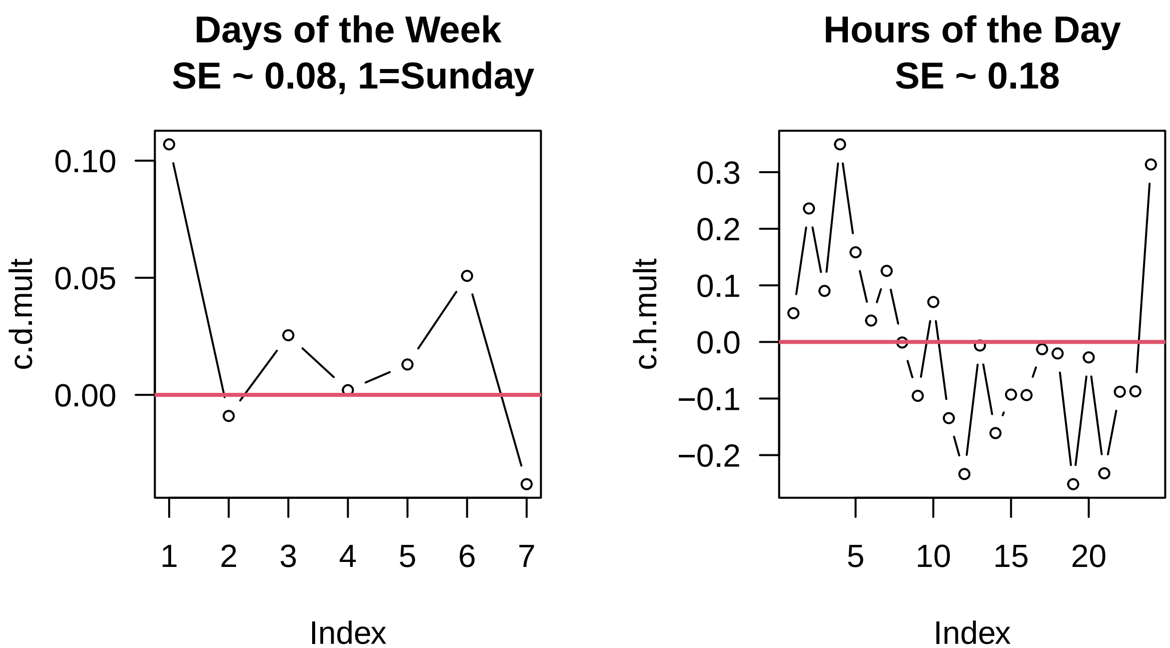
```
## Multiple R-squared:  0.1199,Adjusted R-squared:  0.06272
## F-statistic: 2.096 on 29 and 446 DF,  p-value: 0.000898

plot(lm.mult)
```





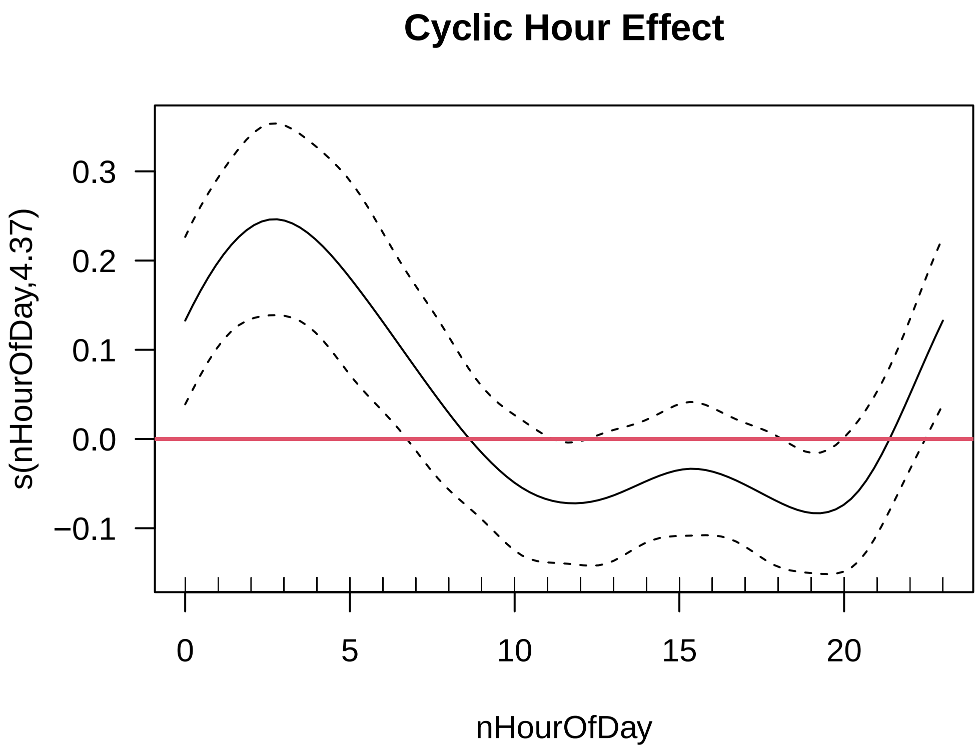
6.2.3 Temporal Patterning to Coefficients?

```
par(mfrow=c(1,2))
c.mult<-coefficients(lm.mult)
c.h.mult<-c.mult[substr(names(c.mult),1,4)=="Hour"]
c.d.mult<-c.mult[substr(names(c.mult),1,3)=="Day"]
c.h.mult<-c(c.mult[1],c.h.mult)
c.d.mult<-c(c.mult[1],c.d.mult)
c.d.mult<-c.d.mult[c(4,2,6,7,5,1,3)]
plot(c.d.mult,type="b",main="Days of the Week\n SE ~ 0.08, 1=Sunday",las=1,cex=0.7)
abline(h=0,lwd=2,col=2)
plot(c.h.mult,type="b",main="Hours of the Day\n SE ~ 0.18",las=1,cex=0.7)
abline(h=0,lwd=2,col=2)
```

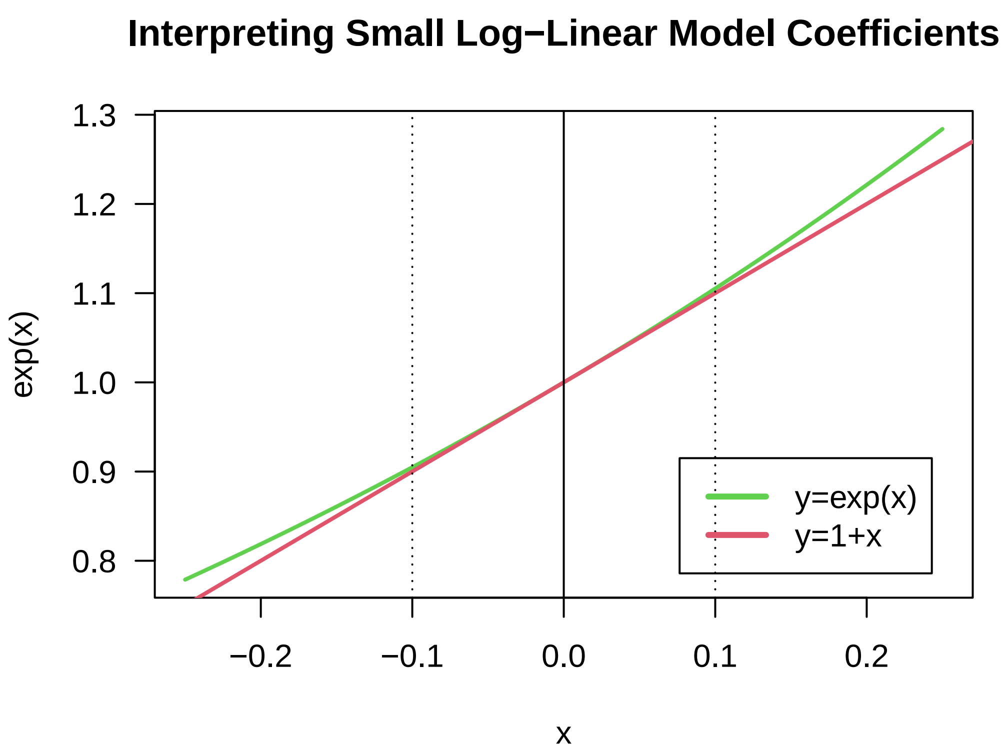


6.2.4 Smooth Hour–of–Day Effect

```
par(mfrow=c(2,1))
## cyclic cubic regression spline for hour of day
gam.mult<-mgcv::gam(lObsToEstRatio.BG~DayOfWeek+s(nHourOfDay,bs="cc"),
                    data=x)  ## Multiplicative GAM
summary(gam.mult)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## lObsToEstRatio.BG ~ DayOfWeek + s(nHourOfDay, bs = "cc")
##
## Parametric coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.016587   0.054590   0.304    0.761
## DayOfWeekMonday    -0.011665   0.073522  -0.159    0.874
## DayOfWeekSaturday  -0.018070   0.076771  -0.235    0.814
## DayOfWeekSunday     0.077975   0.077471   1.007    0.315
## DayOfWeekThursday  -0.009932   0.073840  -0.135    0.893
## DayOfWeekTuesday    0.006388   0.074708   0.086    0.932
## DayOfWeekWednesday -0.021460   0.069379  -0.309    0.757
##
## Approximate significance of smooth terms:
##                 edf Ref.df     F  p-value
## s(nHourOfDay) 4.367      8 3.686 3.57e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0512   Deviance explained = 7.19%
## GCV = 0.17436  Scale est. = 0.1702    n = 476

plot(gam.mult,las=1,main="Cyclic Hour Effect")
abline(h=0,lwd=2,col=2)
## Interpreting Small Coefficients in a Log-Linear Model:
grid<-seq(-0.25,0.25,by=0.0001)
e<-exp(grid)
plot(grid,e,xlab="x",ylab="exp(x)",col=3,type="l",las=1,lwd=2,
main="Interpreting Small Log-Linear Model Coefficients")
abline(a=1,b=1,col=2,lwd=2)
abline(v=0)
abline(v=c(-0.1,0.1),lty=3)
legend("bottomright",inset=0.05,col=c(3,2),lwd=c(3,3),
       lty=1,legend=c("y=exp(x)","y=1+x"))
```



Cyclic Hour Effect
s(nHourOfDay,4.37)
0.3
0.2
0.1
0.0
−0.1
0
5
10
15
20
nHourOfDay



Interpreting Small Log−Linear Model Coefficients
exp(x)
1.3
1.2
1.1
1.0
0.9
0.8
−0.2
−0.1
0.0
0.1
0.2
x
y=exp(x)
y=1+x

7 Estimated Scenario–Specific Travel Times

7.1 Scenario Definitions

	South	Central	NearNorth	FarNorth
Current(0)	1.00	3.00	0.00	0.00
Scenario1	0.00	3.00	1.00	0.00
Scenario2	0.00	3.00	0.00	1.00
Scenario3	1.00	2.00	1.00	0.00
Scenario4	1.00	2.00	0.00	1.00

Table 1: Vehicle allocation matrix. Rows reference placement scenarios and columns station locations. Cells are vehicle counts.

7.2 Compute Scenario–Specific Travel Times

Note that dispatch base and destination regions don't always agree:

```
table(x$REF.GRID,x$BASE.NAME,useNA="always")

##
##               Company 1 Company 9 <NA>
##   1 North             0        52    0
##   2 Central           4       342    0
##   3 South            26        73    0
##   <NA>                0         0    0
```

Compute all scenario projections, include those for the existing scenario, assuming that there is an available vehicle at each of the bases present in the scenario. For each call assume the vehicle in the grid region is used and when there is not a base in the grid region, use the base that is nearest in distance. Occasionally, we compare distances and assign based on minimum distances. **Our choices are somewhat idiosyncratic; please evaluate these definitions and correct them as desired:**

7.2.1 Scenario 0: The Current Configuration

```
x$eTT.BG.Scen0<-rep(NA,nrow(x))
x$eTT.BG.Scen0[x$REF.GRID=="3 South"]<-x$eTT.BG.So[x$REF.GRID=="3 South"]
x$eTT.BG.Scen0[x$REF.GRID=="2 Central"]<-x$eTT.BG.Ce[x$REF.GRID=="2 Central"]
condition<-((x$REF.GRID=="1 North")&(x$Dist.Ce < x$Dist.So)) ## Ce closer
x$eTT.BG.Scen0[condition]<-x$eTT.BG.Ce[condition]
condition<-((x$REF.GRID=="1 North")&(x$Dist.Ce >= x$Dist.So)) ## So closer
x$eTT.BG.Scen0[condition]<-x$eTT.BG.So[condition]
summary(x$eTT.BG.Scen0)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     9.0   265.0   358.0   401.3   481.0  2412.0
```

7.2.2 Scenario 1: Drop South, Add Near North

```
x$eTT.BG.Scen1<-rep(NA,nrow(x))
x$eTT.BG.Scen1[x$REF.GRID=="1 North"]<-x$eTT.BG.NN[x$REF.GRID=="1 North"]
x$eTT.BG.Scen1[x$REF.GRID=="2 Central"]<-x$eTT.BG.Ce[x$REF.GRID=="2 Central"]
condition<-((x$REF.GRID=="3 South")&(x$Dist.NN < x$Dist.Ce)) ## NN closer
x$eTT.BG.Scen1[condition]<-x$eTT.BG.NN[condition]
condition<-((x$REF.GRID=="3 South")&(x$Dist.NN >= x$Dist.Ce)) ## Ce closer
x$eTT.BG.Scen1[condition]<-x$eTT.BG.Ce[condition]
summary(x$eTT.BG.Scen1)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     9.0   268.0   374.0   430.3   526.0  2412.0
```

7.2.3 Scenario 2: Drop South, Add Far North

```
x$eTT.BG.Scen2<-rep(NA,nrow(x))
x$eTT.BG.Scen2[x$REF.GRID=="2 Central"]<-x$eTT.BG.Ce[x$REF.GRID=="2 Central"]
condition<-((x$REF.GRID=="1 North")&(x$Dist.FN < x$Dist.Ce)) ## FN closer
x$eTT.BG.Scen2[condition]<-x$eTT.BG.FN[condition]
condition<-((x$REF.GRID=="1 North")&(x$Dist.FN >= x$Dist.Ce)) ## Ce closer
x$eTT.BG.Scen2[condition]<-x$eTT.BG.Ce[condition]
condition<-((x$REF.GRID=="3 South")&(x$Dist.FN < x$Dist.Ce)) ## FN closer
x$eTT.BG.Scen2[condition]<-x$eTT.BG.FN[condition]
condition<-((x$REF.GRID=="3 South")&(x$Dist.FN >= x$Dist.Ce)) ## Ce closer
x$eTT.BG.Scen2[condition]<-x$eTT.BG.Ce[condition]
summary(x$eTT.BG.Scen2)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     9.0   275.0   383.0   422.4   522.0  2412.0
```

7.2.4 Scenario 3: Move One From Central to Near North

```
x$eTT.BG.Scen3<-rep(NA,nrow(x))
x$eTT.BG.Scen3[x$REF.GRID=="2 Central"]<-x$eTT.BG.Ce[x$REF.GRID=="2 Central"]
x$eTT.BG.Scen3[x$REF.GRID=="3 South"]<-x$eTT.BG.So[x$REF.GRID=="3 South"]
condition<-((x$REF.GRID=="1 North")&(x$Dist.NN < x$Dist.Ce)) ## NN closer
x$eTT.BG.Scen3[condition]<-x$eTT.BG.NN[condition]
condition<-((x$REF.GRID=="1 North")&(x$Dist.NN >= x$Dist.Ce)) ## Ce closer
x$eTT.BG.Scen3[condition]<-x$eTT.BG.Ce[condition]
summary(x$eTT.BG.Scen3)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     9.0   257.0   349.0   375.5   457.0  2412.0
```

7.2.5 Scenario 4: Move One From Central to Far North

```
x$eTT.BG.Scen4<-rep(NA,nrow(x))
x$eTT.BG.Scen4[x$REF.GRID=="3 South"]<-x$eTT.BG.So[x$REF.GRID=="3 South"]
x$eTT.BG.Scen4[x$REF.GRID=="2 Central"]<-x$eTT.BG.Ce[x$REF.GRID=="2 Central"]
condition<-((x$REF.GRID=="1 North")&(x$Dist.FN < x$Dist.Ce)) ## FN closer
x$eTT.BG.Scen4[condition]<-x$eTT.BG.FN[condition]
condition<-((x$REF.GRID=="1 North")&(x$Dist.FN >= x$Dist.Ce)) ## Ce closer
x$eTT.BG.Scen4[condition]<-x$eTT.BG.Ce[condition]
summary(x$eTT.BG.Scen4)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     9.0   264.0   354.0   383.4   469.0  2412.0
```

8 Use Mixed Model to Evaluate the Scenarios

Note: We will want to drop out–of–county calls. At least the durham call.

8.1 Create Model Matrix

```
ID<-paste0("Call",1:nrow(x))
x0<-cbind(x[,c("eTT.BG.Scen0","REF.GRID","DISPATCH.PRIORITY.NAME","HourOfDay",
         "nHourOfDay")],ID,Scenario=rep("S0",nrow(x)))
x1<-cbind(x[,c("eTT.BG.Scen1","REF.GRID","DISPATCH.PRIORITY.NAME","HourOfDay",
         "nHourOfDay")],ID,Scenario=rep("S1",nrow(x)))
x2<-cbind(x[,c("eTT.BG.Scen2","REF.GRID","DISPATCH.PRIORITY.NAME","HourOfDay",
         "nHourOfDay")],ID,Scenario=rep("S2",nrow(x)))
x3<-cbind(x[,c("eTT.BG.Scen3","REF.GRID","DISPATCH.PRIORITY.NAME","HourOfDay",
         "nHourOfDay")],ID,Scenario=rep("S3",nrow(x)))
x4<-cbind(x[,c("eTT.BG.Scen4","REF.GRID","DISPATCH.PRIORITY.NAME","HourOfDay",
         "nHourOfDay")],ID,Scenario=rep("S4",nrow(x)))
colnames(x0)[1]<-"eTT"
colnames(x1)[1]<-"eTT"
colnames(x2)[1]<-"eTT"
colnames(x3)[1]<-"eTT"
colnames(x4)[1]<-"eTT"
x.mm<-rbind(x0,x1,x2,x3,x4)
colnames(x.mm)[3]<-"Priority"
x.mm$Scenario<-factor(x.mm$Scenario)
x.mm$Priority<-factor(x.mm$Priority)
x.mm$ID<-factor(x.mm$ID)
x.mm$l.eTT<-log(x.mm$eTT)
dim(x.mm)

## [1] 2485    8

head(x.mm)

##   eTT  REF.GRID      Priority HourOfDay nHourOfDay    ID Scenario    l.eTT
## 1 539   3 South     Emergency        06          6 Call1       S0 6.289716
## 2 220 2 Central     Emergency        08          8 Call2       S0 5.393628
## 3 346 2 Central     Emergency        10         10 Call3       S0 5.846439
```

```
## 4 306 2 Central     Emergency        11          11 Call4       S0 5.723585
## 5 181 2 Central Non Emergency        12          12 Call5       S0 5.198497
## 6 235 2 Central     Emergency        14          14 Call6       S0 5.459586

table(x.mm$Scenario)

##
##  S0  S1  S2  S3  S4
## 497 497 497 497 497

table(table(x.mm$ID))

##
##   5
## 497
```

8.2 Fit Model – Take 1

8.2.1 Multiple Regression Model Using lm()

```
lm.out<-lm(eTT ~ Scenario + Priority,data=x.mm)
summary(lm.out)

##
## Call:
## lm(formula = eTT ~ Scenario + Priority, data = x.mm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -420.97 -137.93  -38.93   87.50 2035.29
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            400.926     10.370  38.664   <2e-16 ***
## ScenarioS1              29.040     14.274   2.035   0.0420 *
## ScenarioS2              21.161     14.274   1.483   0.1383
## ScenarioS3             -25.755     14.274  -1.804   0.0713 .
## ScenarioS4             -17.907     14.274  -1.255   0.2098
## PriorityNon Emergency    1.535     10.945   0.140   0.8885
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 225 on 2479 degrees of freedom
## Multiple R-squared:  0.008903,Adjusted R-squared:  0.006904
## F-statistic: 4.454 on 5 and 2479 DF,  p-value: 0.0004833
```

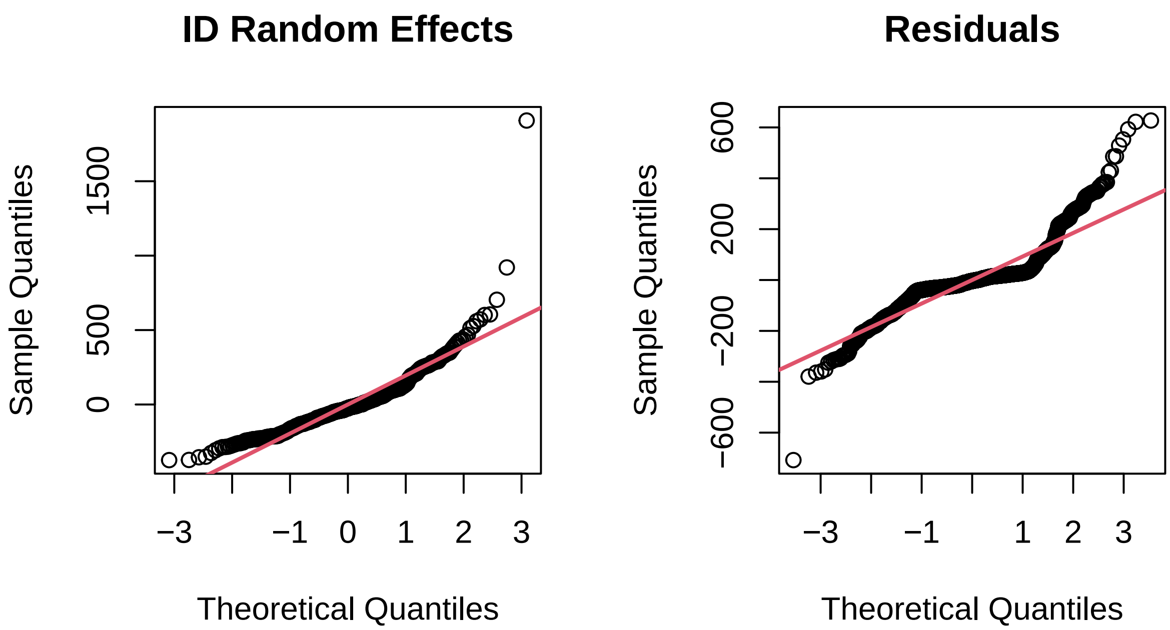
8.2.2 Linear Mixed Model (LMM) using lme4::lmer()

```r
lmm.out<-lmer(eTT ~ Scenario + Priority + (1|ID),data=x.mm)
summary(lmm.out)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: eTT ~ Scenario + Priority + (1 | ID)
##    Data: x.mm
##
## REML criterion at convergence: 31525.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.8834 -0.2954 -0.0342  0.1965  6.0965
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ID       (Intercept) 40116    200.3
##  Residual             10578    102.9
## Number of obs: 2485, groups:  ID, 497
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)            400.926     11.207  35.775
## ScenarioS1              29.040      6.524   4.451
## ScenarioS2              21.161      6.524   3.243
## ScenarioS3             -25.755      6.524  -3.947
## ScenarioS4             -17.907      6.524  -2.745
## PriorityNon Emergency    1.535     22.352   0.069
##
## Correlation of Fixed Effects:
##             (Intr) ScnrS1 ScnrS2 ScnrS3 ScnrS4
## ScenarioS1  -0.291
## ScenarioS2  -0.291  0.500
## ScenarioS3  -0.291  0.500  0.500
## ScenarioS4  -0.291  0.500  0.500  0.500
## PrrtyNnEmrg -0.433  0.000  0.000  0.000  0.000
```

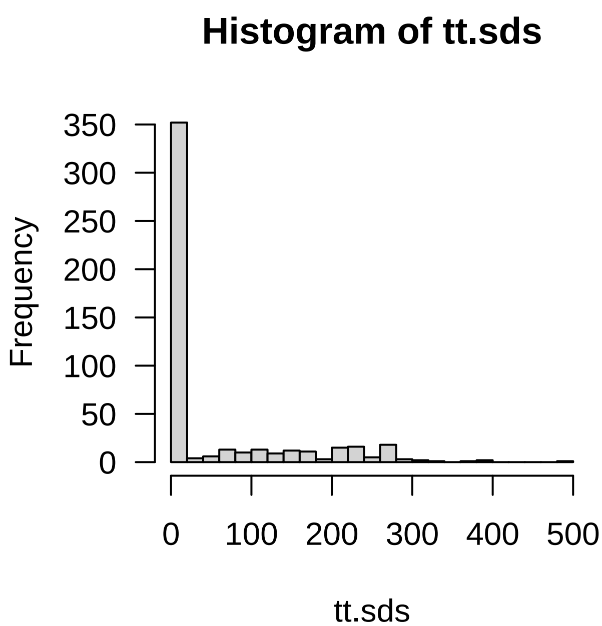
```r
plot(lmm.out)
```

```
par(mfrow=c(1,2))
re<-ranef(lmm.out)$ID
qqnorm(re[,1],main="ID Random Effects")
abline(a=0,b=sd(re[,1]),lwd=2,col=2)
qqnorm(residuals(lmm.out),main="Residuals")
abline(a=0,b=sd(residuals(lmm.out)),lwd=2,col=2)
```



8.2.3 What's Wrong?

```
scenarios<-x[,c("eTT.BG.Scen0","eTT.BG.Scen1","eTT.BG.Scen2",
                "eTT.BG.Scen3","eTT.BG.Scen4")]
tt.sds<-apply(scenarios,1,sd)
hist(tt.sds,las=1,nclass=20)
```




```
table(tt.sds==0,useNA="always")

##
## FALSE  TRUE  <NA>
##   151   346     0
```

8.3 Fit Model – Take 2

8.3.1 Create V2 Model Matrix

```
x.temp<-x[tt.sds>0,]
ID<-paste0("Call",1:nrow(x.temp))
x0<-cbind(x.temp[,c("eTT.BG.Scen0","REF.GRID","DISPATCH.PRIORITY.NAME","HourOfDay",
         "nHourOfDay")],ID,Scenario=rep("S0",nrow(x.temp)))
x1<-cbind(x.temp[,c("eTT.BG.Scen1","REF.GRID","DISPATCH.PRIORITY.NAME","HourOfDay",
         "nHourOfDay")],ID,Scenario=rep("S1",nrow(x.temp)))
x2<-cbind(x.temp[,c("eTT.BG.Scen2","REF.GRID","DISPATCH.PRIORITY.NAME","HourOfDay",
         "nHourOfDay")],ID,Scenario=rep("S2",nrow(x.temp)))
x3<-cbind(x.temp[,c("eTT.BG.Scen3","REF.GRID","DISPATCH.PRIORITY.NAME","HourOfDay",
         "nHourOfDay")],ID,Scenario=rep("S3",nrow(x.temp)))
x4<-cbind(x.temp[,c("eTT.BG.Scen4","REF.GRID","DISPATCH.PRIORITY.NAME","HourOfDay",
         "nHourOfDay")],ID,Scenario=rep("S4",nrow(x.temp)))
colnames(x0)[1]<-"eTT"
colnames(x1)[1]<-"eTT"
colnames(x2)[1]<-"eTT"
colnames(x3)[1]<-"eTT"
colnames(x4)[1]<-"eTT"
x.mm<-rbind(x0,x1,x2,x3,x4)
rm(x0,x1,x2,x3,x4,x.temp)
colnames(x.mm)[3]<-"Priority"
x.mm$Scenario<-factor(x.mm$Scenario)
x.mm$Priority<-factor(x.mm$Priority)
x.mm$ID<-factor(x.mm$ID)
x.mm$l.eTT<-log(x.mm$eTT)
dim(x.mm)

## [1] 755   8

head(x.mm)

##    eTT REF.GRID      Priority HourOfDay nHourOfDay    ID Scenario    l.eTT
## 1  539   3 South     Emergency        06          6 Call1       S0 6.289716
## 8  289   3 South     Emergency        15         15 Call2       S0 5.666427
## 9  284   3 South     Emergency        15         15 Call3       S0 5.648974
## 13 345   3 South Non Emergency        17         17 Call4       S0 5.843544
## 14 623   1 North     Emergency        17         17 Call5       S0 6.434547
## 15 290   3 South     Emergency        19         19 Call6       S0 5.669881

table(x.mm$Scenario)

##
##  S0  S1  S2  S3  S4
## 151 151 151 151 151

table(table(x.mm$ID))

##
##   5
## 151
```

8.3.2 V2 Multiple Regression Model Using lm()

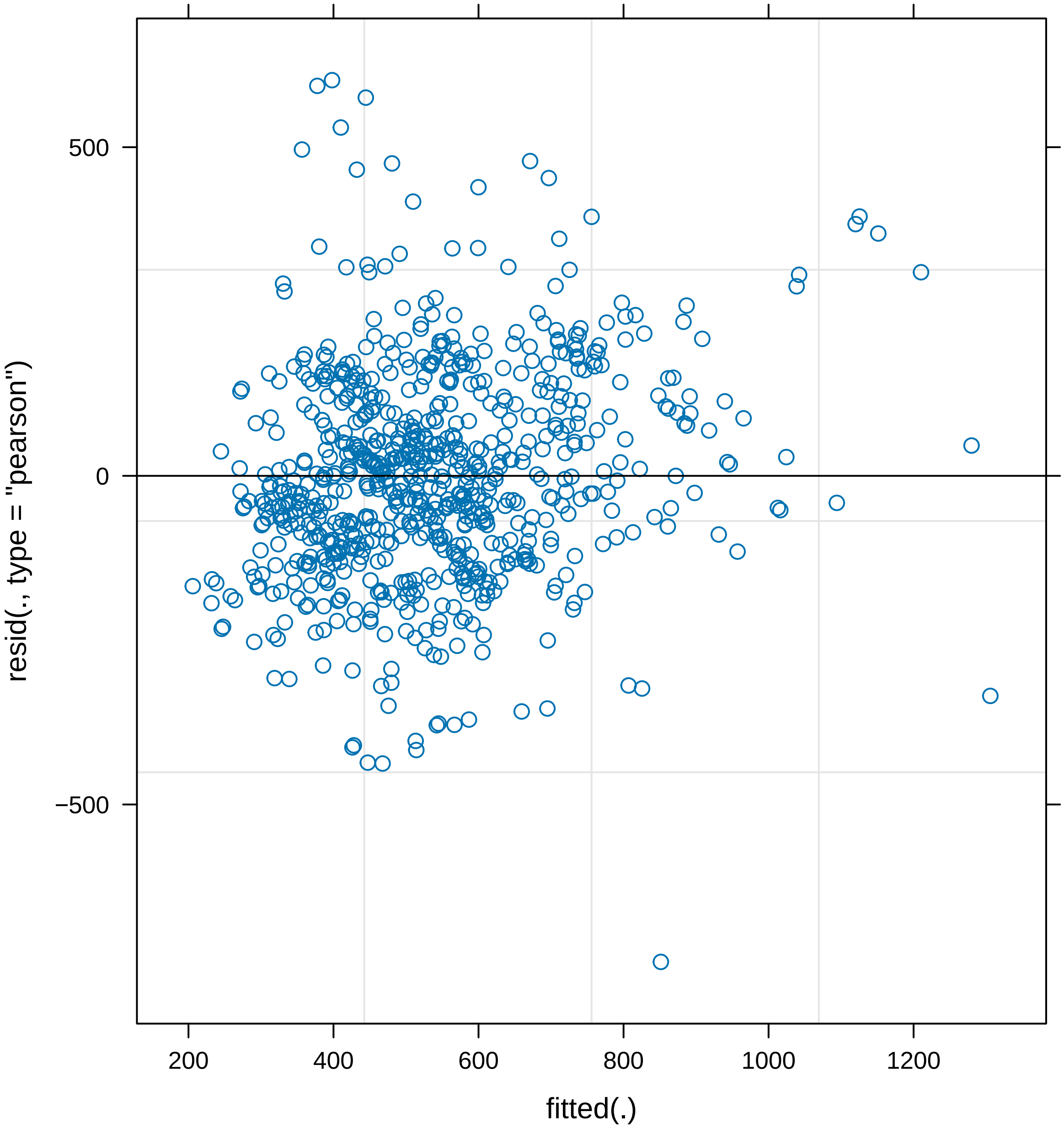
```
lm.out<-lm(eTT ~ Scenario + Priority,data=x.mm)
summary(lm.out)

##
## Call:
## lm(formula = eTT ~ Scenario + Priority, data = x.mm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -609.29 -164.26  -12.94  127.14 1078.06
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             526.71      19.91  26.450  < 2e-16 ***
## ScenarioS1               95.58      27.54   3.471 0.000549 ***
## ScenarioS2               69.65      27.54   2.529 0.011647 *
## ScenarioS3              -84.77      27.54  -3.078 0.002161 **
## ScenarioS4              -58.94      27.54  -2.140 0.032672 *
## PriorityNon Emergency    27.81      22.41   1.241 0.215078
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 239.3 on 749 degrees of freedom
## Multiple R-squared:  0.0813,Adjusted R-squared:  0.07516
## F-statistic: 13.26 on 5 and 749 DF,  p-value: 2.149e-12
```

8.3.3 V2 Linear Mixed Model (LMM) using lme4::lmer()

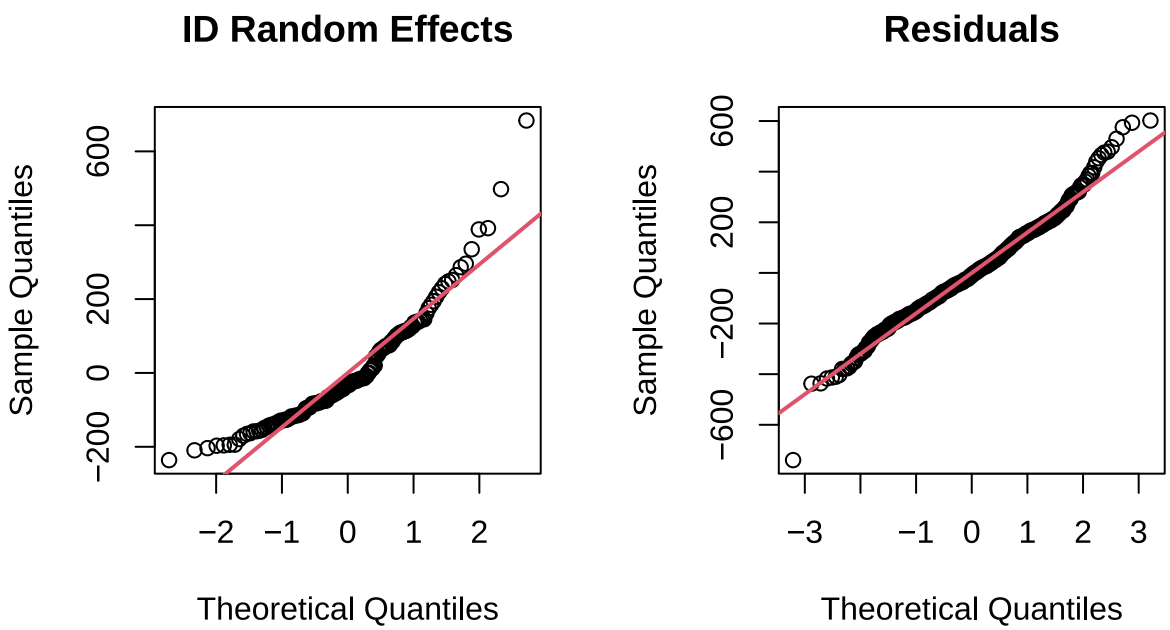
```
lmm.out<-lmer(eTT ~ Scenario + Priority + (1|ID),data=x.mm)
summary(lmm.out)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: eTT ~ Scenario + Priority + (1 | ID)
##    Data: x.mm
##
## REML criterion at convergence: 10143.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.2220 -0.5845 -0.0468  0.5653  3.4369
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ID       (Intercept) 26735    163.5
##  Residual             30677    175.1
## Number of obs: 755, groups:  ID, 151
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)             526.71      20.73  25.407
## ScenarioS1               95.58      20.16   4.742
## ScenarioS2               69.65      20.16   3.455
## ScenarioS3              -84.77      20.16  -4.205
## ScenarioS4              -58.94      20.16  -2.924
## PriorityNon Emergency    27.81      37.96   0.732
##
## Correlation of Fixed Effects:
##             (Intr) ScnrS1 ScnrS2 ScnrS3 ScnrS4
## ScenarioS1  -0.486
## ScenarioS2  -0.486  0.500
## ScenarioS3  -0.486  0.500  0.500
## ScenarioS4  -0.486  0.500  0.500  0.500
## PrrtyNnEmrg -0.340  0.000  0.000  0.000  0.000
```

```
plot(lmm.out)
```



```r
par(mfrow=c(1,2))
re<-ranef(lmm.out)$ID
qqnorm(re[,1],main="ID Random Effects")
abline(a=0,b=sd(re[,1]),lwd=2,col=2)
qqnorm(residuals(lmm.out),main="Residuals")
abline(a=0,b=sd(residuals(lmm.out)),lwd=2,col=2)
```



8.3.4 Scenario Definitions:

	South	Central	NearNorth	FarNorth
Current(0)	1.00	3.00	0.00	0.00
Scenario1	0.00	3.00	1.00	0.00
Scenario2	0.00	3.00	0.00	1.00
Scenario3	1.00	2.00	1.00	0.00
Scenario4	1.00	2.00	0.00	1.00

Table 2: Vehicle allocation matrix. Rows reference placement scenarios and columns station locations. Cells are vehicle counts.

8.3.5 Mixed Model Coefficient Matrix, Again:

```
summary(lmm.out)$coefficients

##                        Estimate Std. Error    t value
## (Intercept)           526.71162   20.73081 25.4071898
## ScenarioS1             95.58278   20.15740  4.7418201
## ScenarioS2             69.64901   20.15740  3.4552569
## ScenarioS3            -84.76821   20.15740 -4.2053140
## ScenarioS4            -58.94040   20.15740 -2.9240074
## PriorityNon Emergency  27.80517   37.96284  0.7324313
```

9 Load on System Analysis

Since these comparisons are based on absolute times of day that are estimated using google travel time data, is there value to adding an estimated, model–based correction factor? Treating these values as missing data and using multiple imputations?

9.1 Compute Scenario–Specific Time–of–Day When the Ambulance is Again Available

```
## compute scenario-specific total times unavailable & time when available:
##   1) Observed time from dispatch to enroute:
summary(as.numeric(x$timeToEnroute))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   3.984   0.000 660.000

##      None are missing; all are positive.
##   2) Observed time from arrival at site to clearance:
##      check all are positive, look for NAs
summary(as.numeric(x$arriveToClearTime))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0    1140    1860    1854    2460    9180      16

##      Some are missing (turn-backs?)
##      Is time from dispatch to clearance available for these?
summary(as.numeric(x$dispToClearTime)) ## none missing

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      60    1440    2220    2224    2880    9660

summary(as.numeric(x$dispToClearTime[is.na(x$arriveToClearTime)]))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    60.0   165.0   270.0   348.8   435.0  1020.0

##   3) Scenario-specific estimated clearance/available time:
##      -- Scenario 0 --
x$eTAvail.BG.Scen0<-(x$DT.DISP + as.difftime(x$eTT.BG.Scen0,units="secs")
                           + x$arriveToClearTime)
##      Fill in missing values (turnbacks) with dispatch to clearance times:
x$eTAvail.BG.Scen0[is.na(x$eTAvail.BG.Scen0)]<-x$DT.DISP[is.na(x$eTAvail.BG.Scen0)]
summary(x$eTAvail.BG.Scen0)

##                       Min.                    1st Qu.
## "2024-01-01 07:34:59.0000" "2024-01-08 14:12:00.0000"
##                     Median                       Mean
## "2024-01-14 10:06:42.0000" "2024-01-13 21:21:35.6237"
##                    3rd Qu.                       Max.
## "2024-01-19 11:10:57.0000" "2024-01-25 19:49:43.0000"

x$dispToCLear.Scen0<-difftime(x$eTAvail.BG.Scen0,x$DT.DISP,units="secs")
##      Check that estimated times are positive:
summary(x$dispToCLear.Scen0)
```

```
##   Length    Class     Mode
##      497 difftime  numeric

##      -- Scenario 1 --
x$eTAvail.BG.Scen1<-(x$DT.DISP + as.difftime(x$eTT.BG.Scen1,units="secs")
                            + x$arriveToClearTime)
##      Fill in missing values (turnbacks) with dispatch to clearance times:
x$eTAvail.BG.Scen1[is.na(x$eTAvail.BG.Scen1)]<-x$DT.DISP[is.na(x$eTAvail.BG.Scen1)]
summary(x$eTAvail.BG.Scen1)

##                        Min.                     1st Qu.
## "2024-01-01 07:32:46.0000" "2024-01-08 14:12:00.0000"
##                      Median                        Mean
## "2024-01-14 10:06:42.0000" "2024-01-13 21:22:03.6136"
##                     3rd Qu.                        Max.
## "2024-01-19 11:18:05.0000" "2024-01-25 19:49:43.0000"

x$dispToCLear.Scen1<-difftime(x$eTAvail.BG.Scen1,x$DT.DISP,units="secs")
##      Check that estimated times are positive:
summary(x$dispToCLear.Scen1)

##   Length    Class     Mode
##      497 difftime  numeric

##      -- Scenario 2 --
x$eTAvail.BG.Scen2<-(x$DT.DISP + as.difftime(x$eTT.BG.Scen2,units="secs")
                            + x$arriveToClearTime)
##      Fill in missing values (turnbacks) with dispatch to clearance times:
x$eTAvail.BG.Scen2[is.na(x$eTAvail.BG.Scen2)]<-x$DT.DISP[is.na(x$eTAvail.BG.Scen2)]
summary(x$eTAvail.BG.Scen2)

##                        Min.                     1st Qu.
## "2024-01-01 07:32:46.0000" "2024-01-08 14:12:00.0000"
##                      Median                        Mean
## "2024-01-14 10:06:42.0000" "2024-01-13 21:21:56.6599"
##                     3rd Qu.                        Max.
## "2024-01-19 11:18:05.0000" "2024-01-25 19:49:43.0000"

x$dispToCLear.Scen2<-difftime(x$eTAvail.BG.Scen2,x$DT.DISP,units="secs")
##      Check that estimated times are positive:
summary(x$dispToCLear.Scen2)

##   Length    Class     Mode
##      497 difftime  numeric

##      -- Scenario 3 --
x$eTAvail.BG.Scen3<-(x$DT.DISP + as.difftime(x$eTT.BG.Scen3,units="secs")
                            + x$arriveToClearTime)
##      Fill in missing values (turnbacks) with dispatch to clearance times:
x$eTAvail.BG.Scen3[is.na(x$eTAvail.BG.Scen3)]<-x$DT.DISP[is.na(x$eTAvail.BG.Scen3)]
summary(x$eTAvail.BG.Scen3)
```

```
##                         Min.                     1st Qu.
## "2024-01-01 07:34:59.0000" "2024-01-08 14:12:00.0000"
##                       Median                        Mean
## "2024-01-14 10:06:42.0000" "2024-01-13 21:21:11.0724"
##                      3rd Qu.                        Max.
## "2024-01-19 11:10:57.0000" "2024-01-25 19:49:43.0000"
```

```
x$dispToCLear.Scen3<-difftime(x$eTAvail.BG.Scen3,x$DT.DISP,units="secs")
##      Check that estimated times are positive:
summary(x$dispToCLear.Scen3)
```

```
##   Length    Class     Mode
##      497 difftime  numeric
```

```
##      -- Scenario 4 --
x$eTAvail.BG.Scen4<-(x$DT.DISP + as.difftime(x$eTT.BG.Scen4,units="secs")
                            + x$arriveToClearTime)
##      Fill in missing values (turnbacks) with dispatch to clearance times:
x$eTAvail.BG.Scen4[is.na(x$eTAvail.BG.Scen4)]<-x$DT.DISP[is.na(x$eTAvail.BG.Scen4)]
summary(x$eTAvail.BG.Scen4)
```

```
##                         Min.                     1st Qu.
## "2024-01-01 07:34:59.0000" "2024-01-08 14:12:00.0000"
##                       Median                        Mean
## "2024-01-14 10:06:42.0000" "2024-01-13 21:21:19.6680"
##                      3rd Qu.                        Max.
## "2024-01-19 11:10:57.0000" "2024-01-25 19:49:43.0000"
```

```
x$dispToCLear.Scen4<-difftime(x$eTAvail.BG.Scen4,x$DT.DISP,units="secs")
##      Check that estimated times are positive:
summary(x$dispToCLear.Scen4)
```

```
##   Length    Class     Mode
##      497 difftime  numeric
```

9.2 Are the Data Temporally Ordered by Dispatch Time?

Look at lagged differences in times and check that they are non–negative:

```
d.time<-diff(x$DT.DISP)
summary(as.numeric(d.time))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    1020    2820    4274    5745   35880
```

9.3 For Each Dispatch, the Number of Trucks Already Out Under Each Scenario

```r
##  Is the truck dispatched while one or more previous trucks are unavailable:
##   ---- Scenario 0 -----
temp<-outer(x$DT.DISP,x$eTAvail.BG.Scen0,FUN="<=")
temp[1:15,1:10]
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
##  [1,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [2,] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [3,] FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [4,] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [5,] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [6,] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [7,] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [8,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
##  [9,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [10,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [11,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
## [12,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [14,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [15,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```r
temp[upper.tri(temp)]<-NA
temp[1:15,1:10]
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
##  [1,]  TRUE    NA    NA    NA    NA    NA    NA    NA    NA    NA
##  [2,] FALSE  TRUE    NA    NA    NA    NA    NA    NA    NA    NA
##  [3,] FALSE FALSE  TRUE    NA    NA    NA    NA    NA    NA    NA
##  [4,] FALSE FALSE FALSE  TRUE    NA    NA    NA    NA    NA    NA
##  [5,] FALSE FALSE FALSE FALSE  TRUE    NA    NA    NA    NA    NA
##  [6,] FALSE FALSE FALSE FALSE FALSE  TRUE    NA    NA    NA    NA
##  [7,] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE    NA    NA    NA
##  [8,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE    NA    NA
##  [9,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE    NA
## [10,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [11,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
## [12,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [14,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [15,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```r
n.out<-apply(temp,1,sum,na.rm=TRUE)
x$n.out.Scen0<-(n.out-1) ## don't count this truck
tbl0<-table(x$n.out.Scen0,useNA="always")
##   ---- Scenario 1 -----
temp<-outer(x$DT.DISP,x$eTAvail.BG.Scen1,FUN="<=")
temp[upper.tri(temp)]<-NA
n.out<-apply(temp,1,sum,na.rm=TRUE)
x$n.out.Scen1<-(n.out-1) ## don't count this truck
tbl1<-table(x$n.out.Scen1,useNA="always")
##   ---- Scenario 2 -----
```

```r
temp<-outer(x$DT.DISP,x$eTAvail.BG.Scen2,FUN="<=")
temp[upper.tri(temp)]<-NA
n.out<-apply(temp,1,sum,na.rm=TRUE)
x$n.out.Scen2<-(n.out-1) ## don't count this truck
tbl2<-table(x$n.out.Scen2,useNA="always")
##   ---- Scenario 3 -----
temp<-outer(x$DT.DISP,x$eTAvail.BG.Scen3,FUN="<=")
temp[upper.tri(temp)]<-NA
n.out<-apply(temp,1,sum,na.rm=TRUE)
x$n.out.Scen3<-(n.out-1) ## don't count this truck
tbl3<-table(x$n.out.Scen3,useNA="always")
##   ---- Scenario 4 -----
temp<-outer(x$DT.DISP,x$eTAvail.BG.Scen4,FUN="<=")
temp[upper.tri(temp)]<-NA
n.out<-apply(temp,1,sum,na.rm=TRUE)
x$n.out.Scen4<-(n.out-1) ## don't count this truck
tbl4<-table(x$n.out.Scen4,useNA="always")
rm(temp,n.out)
tbl0; tbl1; tbl2; tbl3; tbl4
```

```
##
##    0    1    2    3    4    5 <NA>
##  280  164   38   12    2    1    0
##
##    0    1    2    3    4    5 <NA>
##  279  162   41   12    2    1    0
##
##    0    1    2    3    4    5 <NA>
##  280  164   37   12    3    1    0
##
##    0    1    2    3    4    5 <NA>
##  281  164   37   12    2    1    0
##
##    0    1    2    3    4    5 <NA>
##  281  164   37   12    2    1    0
```

10 BART/RF adjustment model

11 Mult Imputations of Travel Times from Tree-Based Model with MICE Adjustment?

random adjustments made prior to creating scenario–specific travel time variables (times may be repeated across scenarios if they share a closest station) in order to be self–consistent. Maybe just describe this; implement only the deterministic correction.