Data Science Challenge-Cabify

Victor Vaquero

October 13, 2019

Part 1: Experiment design

Synopsis

In 2011 AirBnB ran some experiments which showed that when a property featured professional photography, users were much more likely to trust the property and consequently make a booking. So, AirBnB a launched free professional photography service for all hosts. From inside the listing page, hosts were able to click a link to view more about the service, request a professional photographer, and subsequently (after the photo shoot) have their property profile updated with professional photos.

The project initially proved to be a success:

1. Guests were more likely to book a property that had professional photography
2. Hosts were able to charge more for listings with professional photos

However, over time this also became a multimillion dollar operation and a challenge to manage across over 200 countries.

An additional interesting development has also been the proliferation of smartphones with powerful and high-quality cameras (+apps) over the last few years, which has made it more possible for hosts to take good quality photos of their property. There is also the opinion that perhaps millennials have come to expect smartphone photos as the norm and are less likely to expect professional photography.

Challenge

Since the professional photography service consumes so many operational and financial resources, AirBnB management are unsure if they should continue. AirBnB management have asked the Data Science team to analyse the impact of the professional photography service in order to determine whether or not they should continue funding the service.

1. Provide full details about how you will run experiments to assess the impact of this service on both hosts and guests. How will you ensure that the experiments are valid and not biased?

The idea to carry out an experiment is to get a clean and simple way to make causal interference. Prior studies have shown that the images have a strong impact on the demand of the property. Professional images can increase the demand on bookings. However, it has been shown that only one third of the host used the professional images and the guests booking accommodations with professional images paid more attention to

images tend to care more about reviews. Besides, professional pictures can create unrealistically high expectations for guests and hosts are afraid of creating false expectations (especially if the property is not as good as it appears in the images).

With this in my mind, I would run the experiment in different cities of the world, during different periods of the year and with sufficient duration to ensure that the p-value is stable. For the hosts I would ask them if they want to use professional or smartphone images. For the guests I would ask them about the quality of images and I would collect information about the property characteristics (type of accommodation, number of rooms, …), the reservation date, the number of reserve days, the rate per night, the property review, the guest review, … Then I would build a algorithm to classify the images of the properties and a regression model to observe the impact of images on reviews and number of bookings.

The experiment will provide result about if smartphones images generate lower risk in creating a dissatisfactory gap and encouraging the guest to use medium-level images and if smartphones images improve average property demand.

Part 2: Result analysis

Synopsis

In order to optimise operations, engineering team of Cabify has suggested they could query an external real time maps API that not only has roads, but also knows realtime traffic information. We refer to this distance as road distance.

In principle this assigment is more efficient and should outperform linear. However, the queries to the maps API have a certain cost (per query) and increase the complexity and reliability of a critical system within the company. So Data Science team has designed an experiment to help engineering to decide.

Experimental design

The designed expirement is very simple. For a period of 5 days, all trips in 3 cities (Bravos, Pentos and Volantis) have been randomly assigned using linear or road distance:

Trips whose trip_id starts with digits 0-8 were assigned using road distance Trips whose trip_id starts with digits 9-f were assigned using linear distance

Data description

The collected data is available available in this link. Each object represent a vehicle_interval that contains the following attributes:

- type: can be going_to_pickup, waiting_for_rider or driving_to_destination
- trip_id: uniquely identifies the trip
- duration: how long the interval last, in seconds
- distance: how far the vehicle moved in this interval, in meters

- city_id: either bravos, pentos and volantis
- started_at: when the interval started, UTC Time
- vehicle_id: uniquely identifies the vehicle
- rider_id: uniquely identifies the rider

Data Processing

First, we need to download the dataset using the link provided in the Data Science Challenge. The data frame is a json file. In order to read the file, I use the “jsonlite” library and its function fromJSON(). When we try to read the data.frame we observe that it does not have the correct structure of a json file. The JSON data describes an array, and each element of that array is an object. Different objects are separated with commas. However, in the downloaded file there are no commas between the different objects and there are no brackets ([]) at the beginning and at the end. For the first problem, I used a text editor and I replaced } –> by },. For the second issue I introduced by hand the brackets. Now We can read the json file with the function fromJSON:

```
#Loading the libraries:
library(jsonlite)
library(ggplot2)
library(lubridate)
library(dplyr)
library(tidyr)
#Reading the data.frame
table <- fromJSON("intervals_challenge.json")
#Cheching the dimension
dim1<-dim(table)
#Removing the rows with NAs:
data<-drop_na(table)
#test.rows <- sample(1:nrow(data1), 50000);
#data <- data1[test.rows, ];
#Cheching the new dimension once the NAs have been removed:
dim2<-dim(data)
```

The number of rows in the raw data.frame is 165170. The number of rows in the raw data.frame without NAs is 164013. The percentage of lost data is 0.7004904%. As we can see the percentage of lost data is very low and we do not need to think in a strategy for imputing missing data. The next step is to get a tidy data set:

```
library(lubridate)
# I define a new column in data called class which has two values (road and linear) according to:
# Trips whose trip_id starts with digits 0-8 were assigned using road distance
# Trips whose trip_id starts with digits 9-f were assigned using linear distance
data$class<-ifelse(grepl("^0|^1|^2|^3|^4|^5|^6|^7|^8",data$trip_id), "road", "linear")
# I convert the interval started, UTC Time, to POSIXct format:
data$time<-as.POSIXct(data$started_at, origin='1970-01-01', tz="UTC")
# Now I think it is a good idea to classify the trips by a discrete variable in time:
# Defining the labels of the new variable:
labels <- c("Night", "Morning", "Afternoon", "Evening")
# Defining the breaks in time:
breaks <- hour(hm("00:00", "6:00", "12:00", "18:00", "23:59"))
#The new variable is called Time_of_day
data$Time_of_day <- cut(x=hour(data$time), breaks = breaks, labels = labels, include.lowest=TRUE)
# We also get the day of the week:
data$day <- weekdays(data$time)
# We take a look to the data.frame:
head(data)
```

```
##   duration distance started_at                          trip_id
## 1      857     5384 1475499600 c00cee6963e0dc66e50e271239426914
## 2      245     1248 1475499601 427425e1f4318ca2461168bdd6e4fcbd
## 3     1249     5847 1475499602 757867f6d7c00ef92a65bfaa3895943f
## 4      471     2585 1475499602 d09d1301d361f7359d0d936557d10f89
## 5      182      743 1475499602 00f20a701f0ec2519353ef3ffaf75068
## 6      599     1351 1475499602 158e7bc8d42e1d8c94767b00c8f89568
##                          vehicle_id  city_id                  type  class
## 1 52d38cf1a3240d5cbdcf730f2d9a47d6   pentos driving_to_destination linear
## 2 8336b28f24c3e7a1e3d582073b164895 volantis        going_to_pickup   road
## 3 8885c59374cc539163e83f01ed59fd16   pentos driving_to_destination   road
## 4 81b63920454f70b6755a494e3b28b3a7   bravos        going_to_pickup linear
## 5 b73030977cbad61c9db55418909864fa   pentos        going_to_pickup   road
## 6 126e868fb282852c2fa95d88878686bf volantis        going_to_pickup   road
##                   time Time_of_day   day
## 1 2016-10-03 13:00:00   Afternoon lunes
## 2 2016-10-03 13:00:00   Afternoon lunes
## 3 2016-10-03 13:00:01   Afternoon lunes
## 4 2016-10-03 13:00:01   Afternoon lunes
## 5 2016-10-03 13:00:01   Afternoon lunes
## 6 2016-10-03 13:00:02   Afternoon lunes
```

For each trip there must be these three status in the column type: going_to_pickup, waiting_for_rider or driving_to_destination. For each trip I check the frequency with it appears:

```
# I create a table of frequency grouping by trip_id:
n_occur <- data.frame(table(data$trip_id))
numberfreq3<-sum(n_occur$Freq==3)
head(n_occur)
```

```
##                               Var1 Freq
## 1 0000c148c2a6f3c769a83913f0494bda    3
## 2 0002fa307ed23e99992ab87249e924ad    2
## 3 0005f26321516aefee1006dd74e369d1    3
## 4 0007deab499739d6601c31697de1c818    3
## 5 000803729a930d31f0fca656dd35d641    3
## 6 000a91254de20ce2e13c6aaac376f053    2
```

I restrict the analysis to the trips with the full tracking (going_to_pickup, waiting_for_rider or driving_to_destination). This corresponds to trip_ids with freq==3 . The total number of events which satisfy this conditions are 94.8656509%

```
# Getting the trip_ids with frequency equal to 3:
freq3<-n_occur$Var1[n_occur$Freq==3]
print(length(freq3))
```

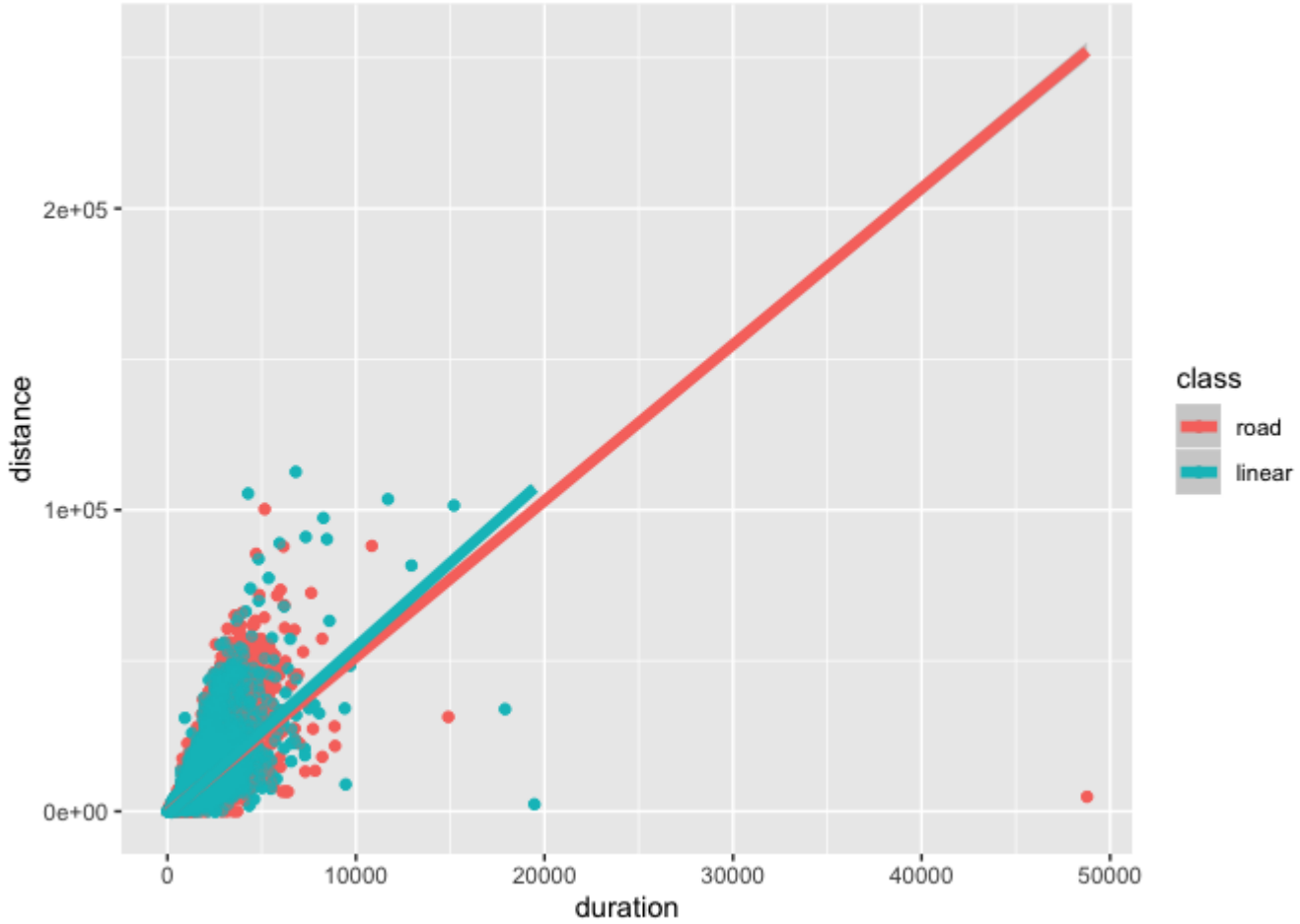
```
## [1] 51864
```

```
# I define a new data.frame to be filled with the new varibles:
newdata<-data.frame()
for(i in freq3){
  # For each trip I verify that it has information about the three types going_to_pickup, waiting_for_rider or dr
iving_to_destination because it is possible that some of them are repeated:
  if(data[data$trip_id==as.character(i),]$type[1]!=data[data$trip_id==as.character(i),]$type[2]){
    if(data[data$trip_id==as.character(i),]$type[1]!=data[data$trip_id==as.character(i),]$type[3]){
      if(data[data$trip_id==as.character(i),]$type[3]!=data[data$trip_id==as.character(i),]$type[2]){
        # Trip id:
        id<-as.character(i)
        # The duration in time of the total trip is the sum between the durations for going_to_pickup
        # and driving_to_destination. The values in waiting_for_rider I think they do not have to be
        # considered since for the most of the cases the driver is waiting without moving and this
        # is independent on the linear and road distances:
        duration<-(data[data$trip_id==as.character(i)&data$type=="going_to_pickup",]$duration
                   +data[data$trip_id==as.character(i)&data$type=="driving_to_destination",]$duration)
        # The distance of the total trip (again considering only the going_to_pickup and driving_to_destination c
ases)
        distance<-(data[data$trip_id==as.character(i)&data$type=="going_to_pickup",]$distance+
                     data[data$trip_id==as.character(i)&data$type=="driving_to_destination",]$distance)
        # I define a new variable called velocity as the ratio between distance and duration.
        velocity<-distance/duration
        # Taking the city, class, day and time of the trip. For that we can take them of any type since
        # they must be the same:
        city<-data[data$trip_id==as.character(i)&data$type=="going_to_pickup",]$city_id
        class<-data[data$trip_id==as.character(i)&data$type=="going_to_pickup",]$class
        time<-data[data$trip_id==as.character(i)&data$type=="going_to_pickup",]$Time_of_day
        day<-data[data$trip_id==as.character(i)&data$type=="going_to_pickup",]$day

        # Filling a new data.drame with the previous variables:
        newdataloop<-data.frame(id,duration,distance,city,class,time,day,velocity)
        # We join the data.frame in each step of the loop with the data.frame generated in the previous
        # iterations:
        newdata<-rbind(newdata,newdataloop)
      }
    }
  }
}
```

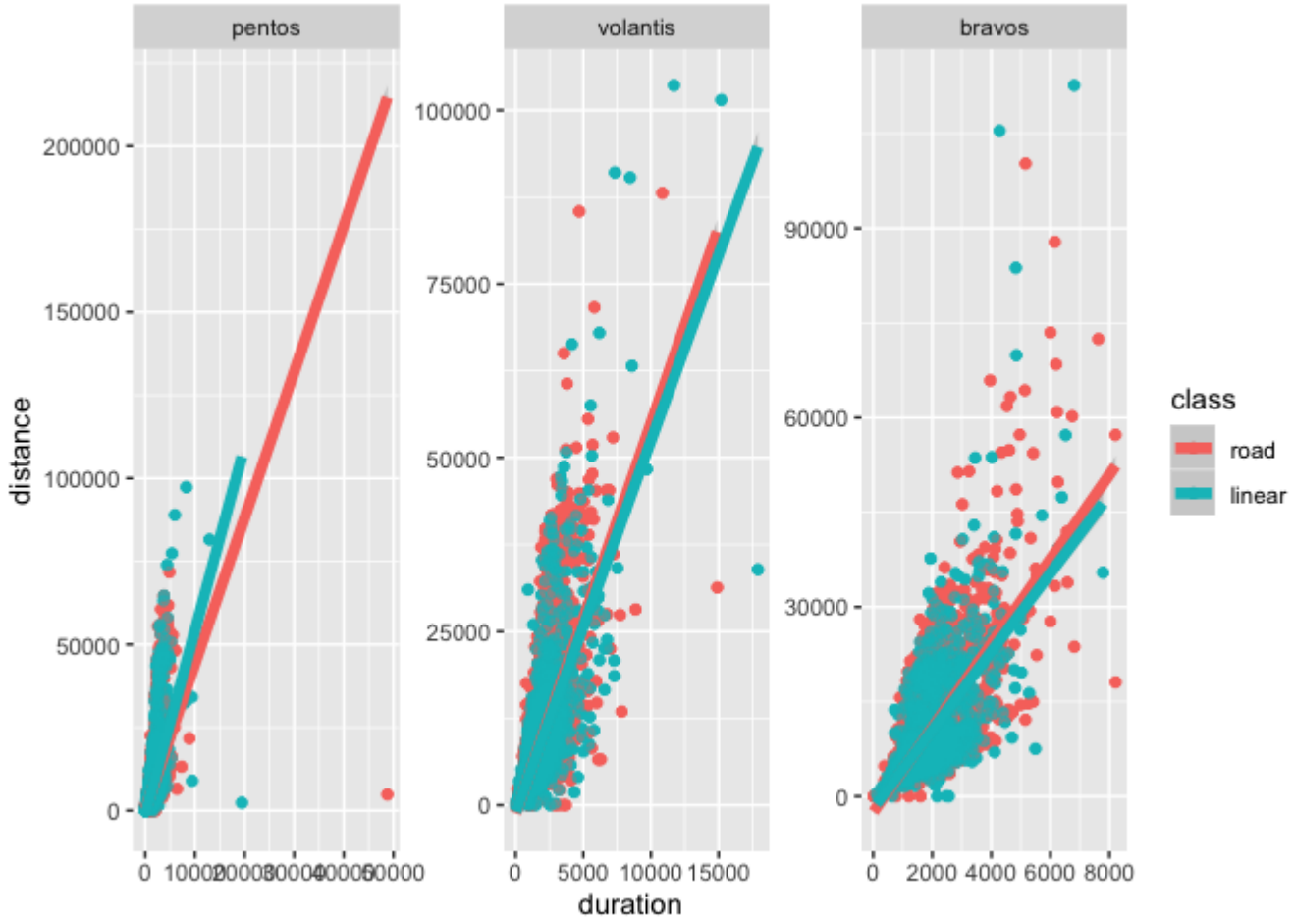
First we can start plotting the distance versus the duration for each trip classified in road or linear distance. A linear fit is included in the plot:

```
library(ggplot2)
myplot<-qplot(duration, distance, data = newdata, colour = class)+ geom_point(alpha = 0.5)+
geom_smooth(method = "lm", , size=2.,se = TRUE)
print(myplot)
```



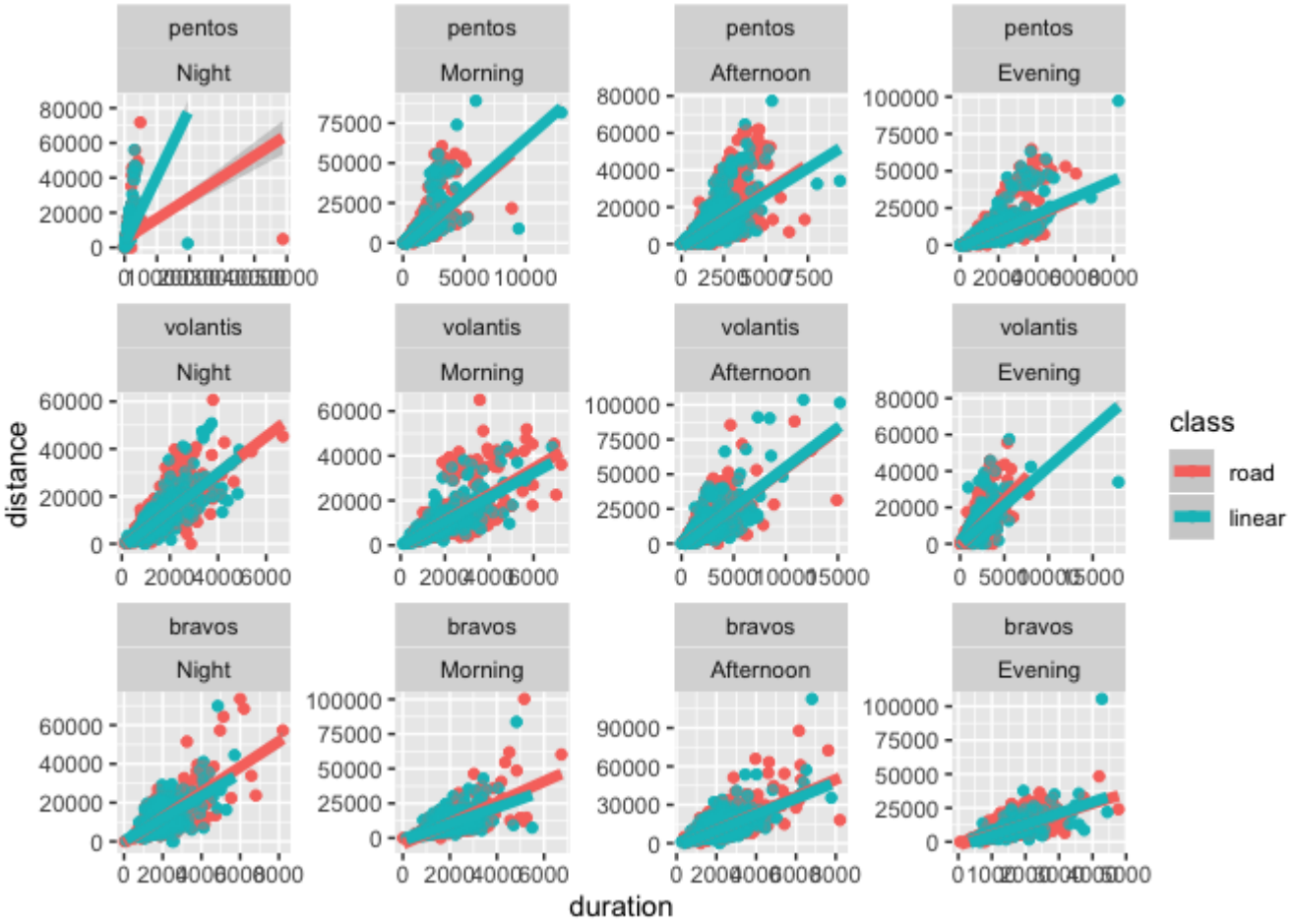
From the last plot and fits we observe that for small distances and small durations the behaviour of both types of distandes is very similar and for long distances it seems that the linear map is better than the road one. However, from this plot we can not extract strong conclusions since we have put together different variables (different cities, time, …). We can study the same kind of correlation but now filtering by the different cities:

```
myplot<-qplot(duration, distance, data = newdata, colour = class)+ geom_point(alpha = 0.5)+
  facet_wrap(city ~., scales="free") +
  geom_smooth(method = "lm", , size=2.,se = TRUE)
print(myplot)
```



We observe that for Pentos the slope of the regression line for linear distance is bigger than the one for road distance, meaning that the company should not move towards road distance in this city. For the other two cities (Volantis and Bravos) the slopes of the regression lines for road distance are bigger than the ones for linear distance and therefore it is interesting for the company to apply the new distance system for these two cities. But I think it is very interesting to study the last behaviors as a funtion of time:

```
myplot<-qplot(duration, distance, data = newdata, colour = class)+ geom_point(alpha = 0.5)+
  facet_wrap(city ~time, scales="free") +
  geom_smooth(method = "lm", , size=2.,se = TRUE)
print(myplot)
```

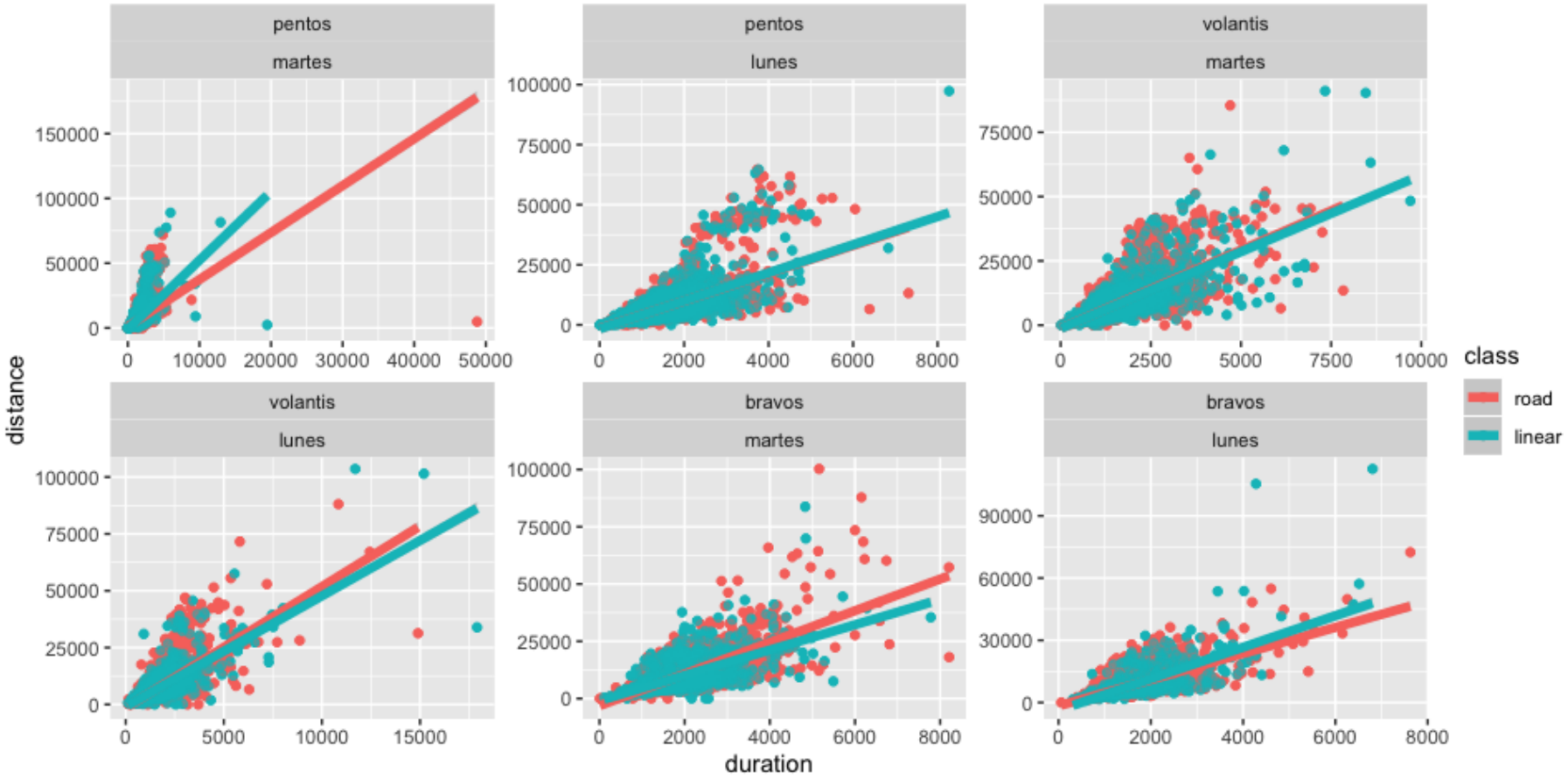


The combinations of city and time for which I consider that it is very interesting for the company to work with the road distance are:

- Pentos during the afternoon
- Volantis during the morning and evening
- Bravos during night and morning

We can also check if it exists differences between the two days of the experiment:

```
myplot<-qplot(duration, distance, data = newdata, colour = class)+ geom_point(alpha = 0.5)+
  facet_wrap(city ~day, scales="free") +
  geom_smooth(method = "lm", , size=2.,se = TRUE)
print(myplot)
```



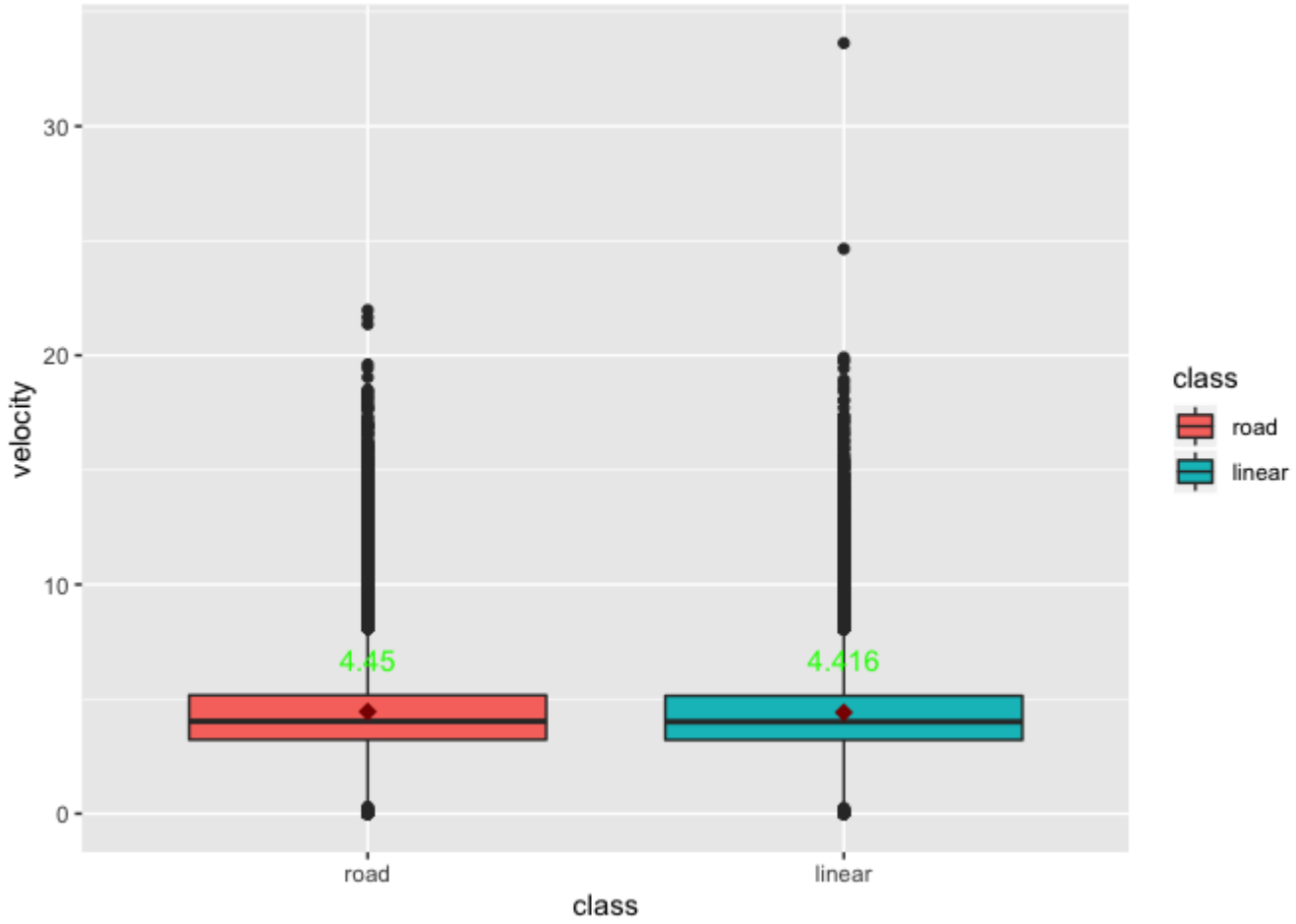
Now I am going to generate boxplots for the variable “velocity” which I defined above. Higher velocities is what interests to the company. First I start with a boxplot for the two types of distances:

```
# We calculate the means to plot them in the boxplot:
means <- aggregate(velocity ~  class, newdata, mean)
is.num <- sapply(means, is.numeric)
means[is.num] <- lapply(means[is.num], round, 3)
means
```

```
##    class velocity
## 1   road    4.450
## 2 linear    4.416
```

```
myplot2<-ggplot(newdata, aes(x = class, y = velocity, fill = class)) + geom_boxplot() +
  stat_summary(fun.y=mean, colour="darkred", geom="point",
               shape=18, size=3,show.legend = FALSE) +
  geom_text(data = means, aes(label = velocity, y = velocity + 2.28), colour="green")

print(myplot2)
```

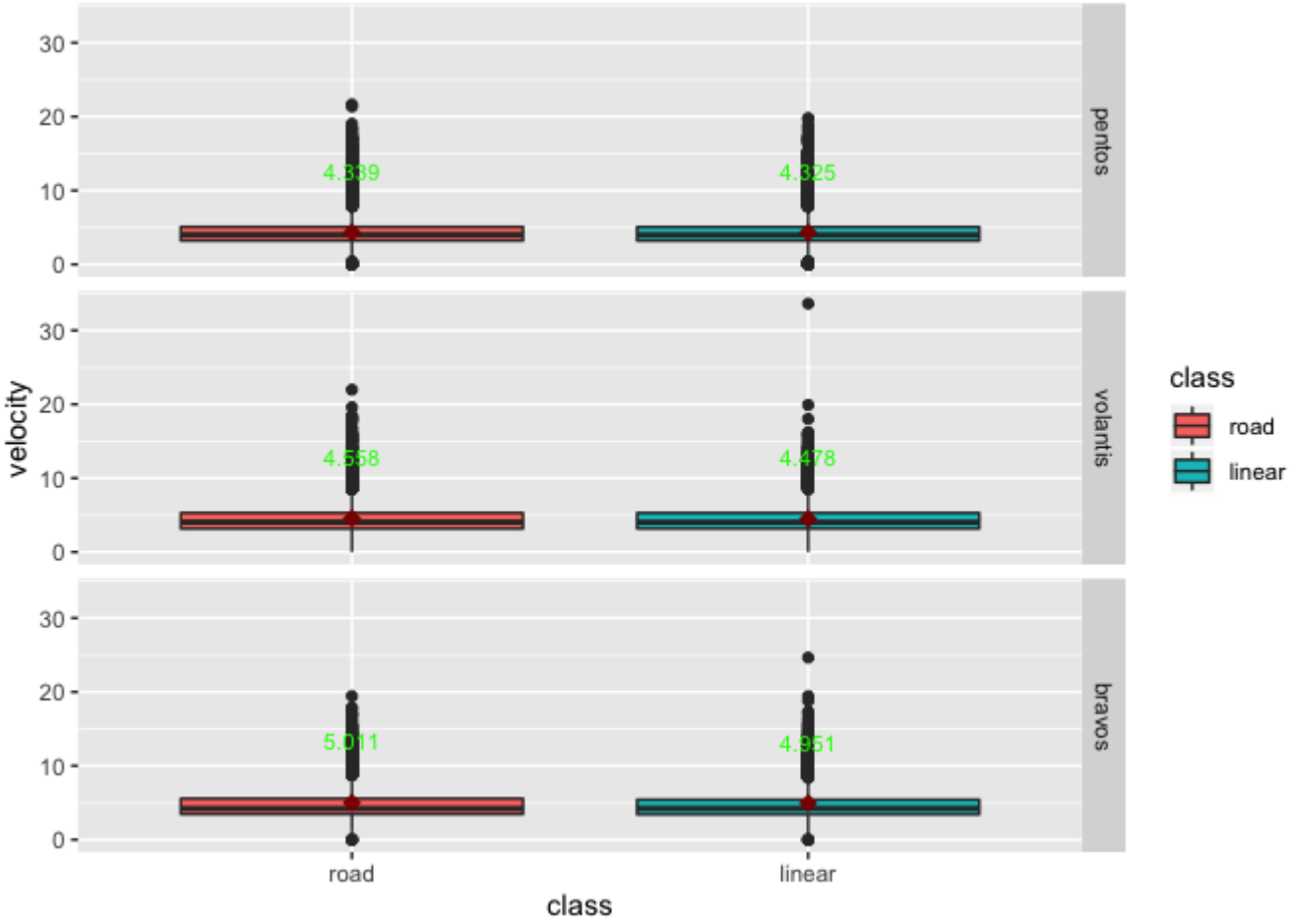


We confirm our statement above: without filter by the time and city the behaviour is very similar and the mean values for the velocity variable are almost equal (indeed lower for the linear distance). Now I filter also by the city:

```
nmeans <- aggregate(velocity ~  (class+city), newdata, mean)
is.num <- sapply(nmeans, is.numeric)
nmeans[is.num] <- lapply(nmeans[is.num], round, 3)
nmeans
```

```
##    class     city velocity
## 1   road   pentos    4.339
## 2 linear   pentos    4.325
## 3   road volantis    4.558
## 4 linear volantis    4.478
## 5   road   bravos    5.011
## 6 linear   bravos    4.951
```

```
myplot3<-ggplot(newdata, aes(x = class, y = velocity, fill = class)) + geom_boxplot() +
  facet_grid(city ~.)+
  stat_summary(fun.y=mean, colour="darkred", geom="point",
               shape=18, size=3,show.legend = FALSE) +
  geom_text(data = nmeans, aes(label = velocity, y = velocity + 8.28),  size=3, colour="green")
print(myplot3)
```



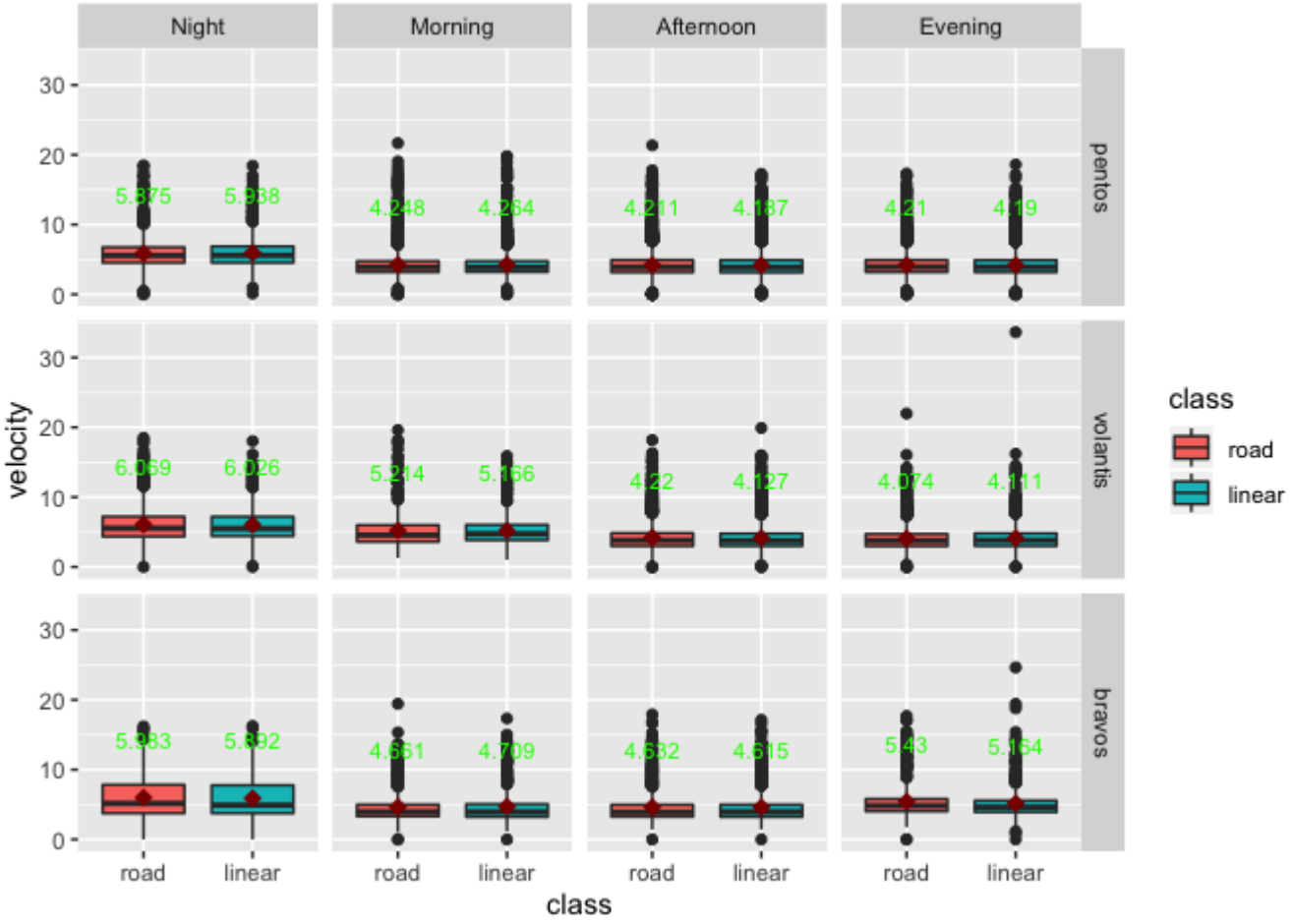
Lastly, a velocity boxplot filtering by class, city and time:

```
nmeans <- aggregate(velocity ~  (class+city+time), newdata, mean)
is.num <- sapply(nmeans, is.numeric)
nmeans[is.num] <- lapply(nmeans[is.num], round, 3)
nmeans
```

```
##     class     city      time velocity
## 1    road   pentos     Night    5.875
## 2  linear   pentos     Night    5.938
## 3    road volantis     Night    6.069
## 4  linear volantis     Night    6.026
## 5    road   bravos     Night    5.983
## 6  linear   bravos     Night    5.892
## 7    road   pentos   Morning    4.248
## 8  linear   pentos   Morning    4.264
## 9    road volantis   Morning    5.214
## 10 linear volantis   Morning    5.166
## 11   road   bravos   Morning    4.661
## 12 linear   bravos   Morning    4.709
## 13   road   pentos Afternoon    4.211
## 14 linear   pentos Afternoon    4.187
## 15   road volantis Afternoon    4.220
## 16 linear volantis Afternoon    4.127
## 17   road   bravos Afternoon    4.632
## 18 linear   bravos Afternoon    4.615
## 19   road   pentos   Evening    4.210
## 20 linear   pentos   Evening    4.190
## 21   road volantis   Evening    4.074
## 22 linear volantis   Evening    4.111
## 23   road   bravos   Evening    5.430
## 24 linear   bravos   Evening    5.164
```

```
myplot4<-ggplot(newdata, aes(x = class, y = velocity, fill = class)) + geom_boxplot() +
  facet_grid(city ~time)+
  stat_summary(fun.y=mean, colour="darkred", geom="point",
               shape=18, size=3,show.legend = FALSE) +
  geom_text(data = nmeans, aes(label = velocity, y = velocity + 8.28),  size=3, colour="green")

print(myplot4)
```



Questions

1. Should the company move towards road distance? What's the max price it would make sense to pay per query? (make all the assumptions you need, and make them explicit)

The conclussion from the analysis explained above is that the company should move towards road distance in the Volantis and Bravos cities. However I would like to stress that the experiment has been conducted under very particular conditions and should be extended. Finally, for the maximum price I suggest to pay for the trip the same price that it would correspond to the linear distance. The gains increase since as the speed is greater the company can make more trips. The customer is happy because he/she has needed less time on the trip and the company is happy because has increased profits.

2. How would you improve the experimental design? Would you collect any additional data?

I would increase the period of days of the experiment. In the studied case only two days were used. It is known that travel times can be drastically affected by traffic, accidents, road work ,… and all of these factors can strongly change depending on the day of the week (in particular during the weekend.) I would also run the experiment in different periods of the year. In addition, the experiment should be carry out considering more cities since only three is a low number. I would also ask the rider to valorate the trip in order to observe if people using road distance are more satisfied than people using linear one.