[1]:	<pre>import matplotlib import pandas as</pre>	s sns					
	<pre>import datetime import numpy as n import math %matplotlib inlin</pre>	pd					
[2]:	<pre>data = pd.read_cs data.head() id vendor_</pre>		ropoff_datetime pa 2016-03-14 17:32:30	nssenger_count pio	ckup_longitude p -73.982155	pickup_latitude dr 40.767937	opoff_longitude di -73.964630
	 id2377394 id3858529 id3504673 	1 2016-06-12 00:43:35 2 2016-01-19 11:35:24 2 2016-04-06 19:32:31 2 2016-03-26	2016-06-12 00:54:38 2016-01-19 12:10:48 2016-04-06 19:39:40 2016-03-26	1 1 1 1	-73.980415 -73.979027 -74.010040 -73.973053	40.738564 40.763939 40.719971 40.793209	-73.999481 -74.005333 -74.012268
[3]: [3]:		13:30:55 In'y a pas de vaisnull(), yticklai	13:38:10 leurs manquant bels= False ,cba	res dans le je r= False, cmap=	ı de données.		-73.972923
	id – vendor_id – pickup_datetime – dropoff_datetime –	pickup_longitude	store_and_fwd_flag - trip_duration -				
[4]:	# distribution designs.distplot(data	manquantes dans le je	u de données. Pou jet]/60)		utes les informati	ons ont été rense	ignées.
	0.0008 0.0007 0.0006 0.0005			041330007			
	0.0004 0.0003 0.0002 0.0000 0 10000	20000 30000 4000 trip_duration	0 50000 60000				
[5]: [5]:	On remarque que la plus sns.distplot(data <matplotlib.axes.< td=""><td>[data['trip_dura</td><td>tion']/3600<2]</td><td>['trip_duration</td><td></td><td>zoomer sur cette</td><td>fenêtre.</td></matplotlib.axes.<>	[data['trip_dura	tion']/3600<2]	['trip_duration		zoomer sur cette	fenêtre.
	0.0008 - 0.0006 - 0.0004 -						
[6]:	quantiles = np.qu tab = pd.DataFram	ne(['5%','10%','5	p_duration'], 0%','90%','95%	']).T			
	<pre>tab = tab.append(tab.columns = tab tab.drop(0, axis= print('Durée moye print('Ecart type tab Durée moyenne des Ecart type: 5237</pre>	c.loc[0,:] c0, inplace=True) cnne des trajets: c: ', data['trip_c' trajets: 959.49	', data['trip duration'].std	duration'].me			
[6]:	5% 10% 50% 9 1 180 245 662 16 Nous avons tracé la dis 5237s (environ 87mn).	Nous avons aussi con	staté la présence d	de trajets avec des	durées très élev	ées qui nous sem	nblent aberrantes. I
	effet, il y a par exemple constituent un faible no Feature engin L'objectif ici est d'explo 'pickup datetime' en sé	embre puisque 95% de eering iter les informations su	s trajets du jeu de ur les trajets pour e	données font moin n expliciter de nou	s de 2104s (envi velles. On comm	ron 35mn). ence d'abord par	la variable
[3]:	# On isole la dat data['pickup_time data['pickup_time data.head()	ce et l'heure du e'] = data['pickup'] = data['pickup']	trajet p_datetime'].a p_datetime'].a	éclenché (début du pply(lambda x pply(lambda x	: x.split()[: x.split()[0])	
.:	id vendor_i 0 id2875421 1 id2377394	2 2016-03-14 17:24:55 1 2016-06-12 00:43:35 2 2016-01-19 11:35:24	2016-03-14 17:32:30 2016-06-12 00:54:38 2016-01-19 12:10:48	1 1	-73.982155 -73.980415 -73.979027	40.767937 40.738564 40.763939	-73.964630 -73.999481 -74.005333
[4]:	# On isole le moi data['pickup_mont	2 2016-04-06 19:32:31 2 2016-03-26 13:30:55 as et on trace la h'] = data['pick'	2016-04-06 19:39:40 2016-03-26 13:38:10 distribution up_date'].appl	y(lambda x : :	k.split(<mark>'-'</mark>)[1])	-74.012268 -73.972923
	<pre>result = data.gro #ax = sns.barplot(ax = sns.barplot(ax.set(xlabel='mo plt.show()</pre>	<pre>upby('pickup_mon t(x='pickup_month' x='pickup_month'</pre>	th')['trip_dur ',y='trip_dura ,y='trip_durat rée moyenne',	ation'].aggred tion',data=resion',data=resi	gate(np.mean) sult,order=re ult)	<pre>.reset_index esult['pickup_</pre>	_month'])
	durée moyenne - 009						
	D'après ce graphique, o durée du trajet.	mois	05 06 le mois où le trajet	est effectué, sa du	urée moyenne va	rie. Le mois a doi	nc une influence sı
[5]:	<pre>def get_day(x): correspondanc int_day = dat ay()</pre>	re = ['Mon','Tus' etime.datetime(int_day	,'Wed','Thu',' nt(x.split('-'	Fri','Sat','S), int(x.spli	it('-')[2])).w
[8]:	<pre>data['pickup_day' result = data.gro ax = sns.barplot(t','Sun']) ax.set(xlabel='joine') plt.show()</pre>	oupby(["pickup_da x='pickup_day',y	y"])['trip_dur ='trip_duratio rée moyenne',	ation'].aggredn',data=result	c, order=['Mo	on','Tus','Wed	d','Thu','Fri'
	1000 - 800 - 600 -	ne du trajet selon le jo	ur de la semanie				
	200 - 0 Mon Tus	Wed Thu Fri jour	Sat Sun				
[9]:	D'après ce graphique, de le samedi sont les jours où les trajets en taxi so d'ata['pickup_hour result = data.groax = sns.barplot(s où les trajets en taxis ent susceptibles de dur ment de la journé et'] = data['pickup pupby(["pickup_ho	sont susceptibles of er moins longtemp e (heure) où l p_time'].apply ur"])['trip_du	de durer plus longtes. e trajet déma. (lambda x : x ration'].aggre	emps. En revanc	he, le dimanche e	et le lundi sont les j
	<pre>ax.set(xlabel=r'h l'heure de la jo plt.show()</pre>	eure de la journ	ée', ylabel=r'			rée moyenne d	du trajet selo
	- 008						
	durée moyenne - 009 - 200 -						
	200 -	journée où on prend le		venne du trajet diffe	ère. Elle est plus	élevée par exem _l	ole les après-midis
11]:	Selon le moment de la 12h et 18h. # On calcule la cafrom math import def compute_distatry: R = 6373. lat1 = ralon1 = ra	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, ance (row):	etaxi, la durée moy oiseau entre latan2, radians ff_latitude']) ff_longitude']	e point de dép			ple les après-midis
11]:	Selon le moment de la 12h et 18h. # On calcule la cafrom math import def compute_distatry: R = 6373. lat1 = rallon1 = rallon2 = rallon2 = rallon2 = rallon2 = rallon3 = sin (distance)	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, nce(row): 0 dians(row['dropo dians(row['dropo dians(row['pickup dians(row['p	etaxi, la durée moy oiseau entre latan2, radians ff_latitude']) ff_longitude'] p_latitude']) p_longitude']) s(lat1) * cos(e point de dép	oart et la de		ole les après-midis
12]: 13]:	Selon le moment de la 12h et 18h. # On calcule la cafrom math import def compute_distatry: R = 6373. lat1 = ralon1 = ralon2 = radlat = ladlon = loa = sin(datatry) = raturn Recept KeyErr	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, nce(row): 0 dians(row['dropo dians(row['dropo dians(row['picku] dians(row['p	etaxi, la durée moy oiseau entre latan2, radians ff_latitude']) ff_longitude']) p_latitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d	e point de dép) lat2) * sin(di	part et la de		ple les après-midis
12]: 13]:	Selon le moment de la 12h et 18h. # On calcule la cafrom math import def compute_distatry: R = 6373. lat1 = ralon1 = ralon2 = ralon2 = radlat = ladlon = loa = sin(datatry) = laturn Revent KeyErreprint(row) data['trip_dist'] sns.jointplot(x=')	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, nce(row): 0 dians(row['dropo dians(row['dropo dians(row['picku] dians(row['p	etaxi, la durée moy oiseau entre latan2, radians ff_latitude']) ff_longitude']) p_latitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d	e point de dép) lat2) * sin(di	part et la de		ple les après-midis
12]: 13]:	Selon le moment de la 12h et 18h. # On calcule la confrom math import def compute_distatry: R = 6373. lat1 = rallon1 = rallon2 = rallon2 = rallon2 = rallon2 = rallon2 = rallon2 = rallon3 = sin(done = loa = lo	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, nce(row): 0 dians(row['dropo dians(row['dropo dians(row['picku] dians(row['p	etaxi, la durée moy oiseau entre latan2, radians ff_latitude']) ff_longitude']) p_latitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d	e point de dép) lat2) * sin(di	oart et la de		ple les après-midis
12]: 13]:	Selon le moment de la 12h et 18h. # On calcule la confrom math import def compute distative: R = 6373. lat1 = ralon1 = ralon2 = radlat = ladlon = lodlat = ladlon = lodlate = ladlon =	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, nce (row): 0 dians (row ['dropo dians (row ['picku] dians (row ['p	etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) ff_longitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348>	e point de départe de la lata de lata de la lata de lata	aremarqué plus l	estination.	bution de la durée
12]: 13]:	Selon le moment de la 12h et 18h. # On calcule la ca from math import def compute_distatry: R = 6373. lat1 = ralon1 = ralon2 = radlat = ladlon = lodlat = ladlon = lodlate = ladlon = lodlon = lodlate = ladlon = lodlate =	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, ance (row): 0 dians (row ['dropo dians (row ['picku] dians (row ['	e taxi, la durée moy oiseau entre la atan2, radians ff_latitude']) ff_longitude']) p_latitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> seau calculée, seau calculée,	e point de départe de la contra del contra de la contra del l	a remarqué plus le voit sur ce graphes trajets de ces trajets de c	naut avec la distri nique que des traju ce qui est imposs ajets et on supprir au fait que c'est la	bution de la durée ets de moins de 10 sible. On va donc mera les trajets ave a vitesse maximale
12]: 13]: 14]:	Selon le moment de la 12h et 18h. # On calcule la confrom math import def compute_distattry: R = 6373. lat1 = ratlon1 = ratlon2 = ratlon2 = ratlon2 = ratlon2 = ratlon2 = ratlon2 = ratlon3 = sin(dougle conformation = sin(doug	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, ince (row): 0 dians (row['dropo: dians (row['dropo: dians (row['picku; dians (row['pi	e taxi, la durée moy oiseau entre latan2, radians ff_latitude']) ff_longitude']) p_longitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> seau calculée, /'data['trip_d' 'spead']!=0) & ip_duration', d 'scatter')	nce. Comme on l'a rants. En effet, on source des la vitesse mo le choix du seuil à on calcule la uration']/3600 (data['trip_comme	a remarqué plus le voit sur ce graphes trajets de Oken, pyenne de ces trajones de vitesse moy duration']/36	naut avec la distri nique que des traje ce qui est imposs ajets et on supprir au fait que c'est la venne des traje	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avitesse maximale jets [0]/data.shape
12]: 13]: 14]: 15]:	Selon le moment de la 12h et 18h. # On calcule la confirmat def compute_distatury: R = 6373. lat1 = rallon1 = rallon2 = rallon2 = rallon2 = rallon2 = rallon2 = rallon3 = sin(done to a = si	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, d' sidans (row['dropo dians (row['dropo dians (row['pickur dians (r	etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) ff_longitude']) p_latitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 'spead']!=0) & ip_duration', d 'spead']!=0) & ip_duration', d 'scalculée, d 'spead']!=0) &	nce. Comme on l'a rants. En effet, on source des la vitesse mo le choix du seuil à on calcule la uration']/3600 (data['trip_comme	a remarqué plus le voit sur ce graphes trajets de Oken, pyenne de ces trajones de vitesse moy duration']/36	naut avec la distri nique que des traje ce qui est imposs ajets et on supprir au fait que c'est la venne des traje	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avitesse maximale jets [0]/data.shape
12]: 13]: 14]: 15]:	Selon le moment de la 12h et 18h. # On calcule la cafrom math import def compute_distatry: R = 6373. lat1 = ralon1 = ralon2 = radlat = ladlon = lodlat = ladlon = lodlate = ladlon = l	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, d' sidans (row['dropo dians (row['dropo dians (row['pickur dians (r	etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) ff_longitude']) p_latitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 'spead']!=0) & ip_duration', d 'spead']!=0) & ip_duration', d 'scalculée, d 'spead']!=0) &	nce. Comme on l'a rants. En effet, on source des la vitesse mo le choix du seuil à on calcule la uration']/3600 (data['trip_comme	a remarqué plus le voit sur ce graphes trajets de Oken, pyenne de ces trajones de vitesse moy duration']/36	naut avec la distri nique que des traje ce qui est imposs ajets et on supprir au fait que c'est la venne des traje	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avitesse maximale jets [0]/data.shape
12]: 13]: 14]: 15]:	Selon le moment de la 12h et 18h. # On calcule la de from math import def compute distaty: R = 6373. lat1 = rallon1 = rallon2 = rallon2 = rallon2 = rallon4 = ladlon = lodlon = lodl	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, d' sidans (row['dropo dians (row['dropo dians (row['pickur dians (r	etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) ff_longitude']) p_latitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 'spead']!=0) & ip_duration', d 'spead']!=0) & ip_duration', d 'scalculée, d 'spead']!=0) &	nce. Comme on l'a rants. En effet, on source des la vitesse mo le choix du seuil à on calcule la uration']/3600 (data['trip_comme	a remarqué plus le voit sur ce graphes trajets de Oken, pyenne de ces trajones de vitesse moy duration']/36	naut avec la distri nique que des traje ce qui est imposs ajets et on supprir au fait que c'est la venne des traje	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avitesse maximale jets [0]/data.shape
12]: 13]: 14]: 15]:	Selon le moment de la 12h et 18h. # On calcule la difrom math import def compute dista try: R = 6373. lat1 = ra lon2 = ra dlat = la dlon = lo a = sin (a c = 2 * xa return R except KeyErr print (row data['trip_dist'] sns.jointplot (x=' <seaborn.axisgrid =="" data['spead']="data['spead']</td"><td>heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, ince (row): 0 dians (row ['dropo dians (row ['dropo dians (row ['pickun dians (row ['dropo dians (row ['dro</td><td>etaxi, la durée moy oiseau entre la atàn2, radians ff_latitude']) ff_longitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 2813558f348> ip_duration', d 'spead']!=0) & ip_duration', d 'scatter') =r'durée du tr 2812e61e2c8></td><td>e point de déplante de point de déplante de la contraction de la c</td><td>a remarqué plus la voit sur ce graph strajets de Okm, byenne de ces tra 130km/h est dû a vitesse moy duration']/36 a ['spead']<13 a ['spead']<13 a ['spead']<13</td><td>naut avec la distri ique que des traj ce qui est imposs ajets et on supprir au fait que c'est la venne des traj (00<3)].shape</td><td>bution de la durée ets de moins de 10 sible. On va donc mera les trajets ave a vitesse maximale jets [0]/data.shape spead']!=0) & tion de la dis</td></seaborn.axisgrid>	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, ince (row): 0 dians (row ['dropo dians (row ['dropo dians (row ['pickun dians (row ['dropo dians (row ['dro	etaxi, la durée moy oiseau entre la atàn2, radians ff_latitude']) ff_longitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 2813558f348> ip_duration', d 'spead']!=0) & ip_duration', d 'scatter') =r'durée du tr 2812e61e2c8>	e point de déplante de point de déplante de la contraction de la c	a remarqué plus la voit sur ce graph strajets de Okm, byenne de ces tra 130km/h est dû a vitesse moy duration']/36 a ['spead']<13 a ['spead']<13 a ['spead']<13	naut avec la distri ique que des traj ce qui est imposs ajets et on supprir au fait que c'est la venne des traj (00<3)].shape	bution de la durée ets de moins de 10 sible. On va donc mera les trajets ave a vitesse maximale jets [0]/data.shape spead']!=0) & tion de la dis
12]: 13]: 14]: 15]:	Selon le moment de la 12h et 18h. # On calcule la cafrom math import def compute distatry: R = 6373. lat1 = ralon1 = ralon2 = radlat = ladlon = loa = sin(data = ladlon = loa = return Rever reprint (row data ['trip_dist'] sns.jointplot(x=' <seaborn.axisgrid -="" 300000="" 3000000="" 300<="" 3500000="" td=""><td>heure de la journée journée où on prend le listance à vol d' sin, cos, sqrt, since (row): 0 dians (row ['dropo dians (row ['pickuj dians (row [</td><td>etaxi, la durée moy diseau entre la atan2, radians ff_latitude']) ff_longitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 2813558f348> ip_duration', d 2812e6le2c8> ip_duration', d 2812e6le2c8></td><td>nnce. Comme on l'a (axis=1) (ata=data, kinda (ata=data, kinda</td><td>a remarqué plus l'voit sur ce graph strajets de 0km, oyenne de ces tra 130km/h est dû a vitesse moy duration']/36 a ['spead']<13 a 'Durée du tr</td><td>naut avec la distri lique que des traj ce qui est imposs ajets et on supprir au fait que c'est la renne des traj (00<3)].shape</td><td>bution de la durée ets de moins de 10 de la dise maximale de vitesse maximale jets (0) / data.shape spead']!=0) & tion de la dise trajennée (99.4 %).</td></seaborn.axisgrid>	heure de la journée journée où on prend le listance à vol d' sin, cos, sqrt, since (row): 0 dians (row ['dropo dians (row ['pickuj dians (row [etaxi, la durée moy diseau entre la atan2, radians ff_latitude']) ff_longitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 2813558f348> ip_duration', d 2812e6le2c8> ip_duration', d 2812e6le2c8>	nnce. Comme on l'a (axis=1) (ata=data, kinda	a remarqué plus l'voit sur ce graph strajets de 0km, oyenne de ces tra 130km/h est dû a vitesse moy duration']/36 a ['spead']<13 a 'Durée du tr	naut avec la distri lique que des traj ce qui est imposs ajets et on supprir au fait que c'est la renne des traj (00<3)].shape	bution de la durée ets de moins de 10 de la dise maximale de vitesse maximale jets (0) / data.shape spead']!=0) & tion de la dise trajennée (99.4 %).
12]: 13]: 14]: 15]:	Selon le moment de la 12h et 18h. # On calcule la compute dista try: R = 6373. lat1 = ra lon1 = ra lon2 = ra data = la dista = la dis	heure de la journée journée où on prend le distance à vol d' sin, cos, sqrt, ince (row): 0 dians (row ['dropo dians (row ['dropo dians (row ['picku] dians (row ['dropo dians (row ['picku] dians (row ['dropo	etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) ff_longitude']) p_latitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 2813558f348> ip_duration', d 2812e6le2c8> ip_duration', d 2812e6le2c8> ip_duration', d 2812e6le2c8> ip_duration', d 2812e6le2c8>	e point de déplant de déplant de déplant de déplant de déplant de la contract. En effet, on son remarque des louler la vitesse mouve choix du seuil à contract de la distance de la contract de la distance de la distan	a remarqué plus l'voit sur ce graph strajets de Okm, pyenne de ces tra 130km/h est dû a vitesse moy duration']/36 a ('spead']<13 ar'Durée du tra conservant pre du tra conservant pre du tra de conservant pre d	naut avec la distribique que des trajice qui est impossajets et on supprirau fait que c'est la renne des trajico (data['srajet en fonction au faita_bis['trigickup_date', 'pickup_date', '	bution de la durée ets de moins de 10 dible. On va donc mera les trajets aver a vitesse maximale jets [0]/data.shape spead']!=0) & tion de la dis tion de la dis capadiantion']/3 ickup_time','s
12]: 13]: 14]: 15]: 17]: 18]:	Selon le moment de la 12h et 18h. # On calcule la compute dista try: R = 6373. lat1 = ra lon1 = ra lon2 = ra data = la dista = la dis	heure de la journée journée où on prend le journée o	etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) ff_longitude']) platitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> are eux étaient aber viron 416h). De plu iire, nous allons cal rieure à 130km/h. L seau calculée, / (data['trip_d' 'spead']!=0) & ip_duration' allongitude' ip_duration', d 'scatter') =r'durée du tr 2812e61e2c8> are durée du tr	late point de départe de la late la la	a remarqué plus l'voit sur ce graph strajets de Okm, pyenne de ces tra 130km/h est dû a vitesse moy duration']/36 a ('spead']<13 ar'Durée du tra conservant pre du tra conservant pre du tra de conservant pre d	naut avec la distribique que des trajice qui est impossajets et on supprirau fait que c'est la renne des trajico (data['srajet en fonction au faita_bis['trigickup_date', 'pickup_date', '	bution de la durée ets de moins de 10 dible. On va donc mera les trajets aver a vitesse maximale jets [0]/data.shape spead']!=0) & tion de la dis tion de la dis capadiantion']/3 ickup_time','s
12]: 13]: 14]: 15]: 17]: 18]:	Selon le moment de la 12h et 18h. # On calcule la control de compute dista try: R = 6373. lat1 = ra lon1 = ra lon2 = ra dolat = la	heure de la journée journée où on prend le journée où or syrt, on dians (row ['dropo dians (row ['	etaxi, la durée moy ciseau entre la atan2, radians ff latitude']) ff longitude']) platitude']) platitude']) platitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 2813558f348> ip_duration', d 2813558f348> ip_duration', d 2813564 40.763937 40.7937 40.79364 40.79370 40.7937	e point de déplant de déplant de déplant de déplant de déplant de la contract de	a remarqué plus la voit sur ce graph strajets de 0km, byenne de ces tra 130km/h est dû a vitesse moy duration']/36 a vitesse du tra durée du tra dur	naut avec la distri lique que des traj ce qui est imposs ajets et on supprir au fait que c'est la renne des traj (00<3)].shape (0) & (data['s rajet en fonce esque toute la dor (c) l'esque toute la	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429
12]: 13]: 14]: 15]: 17]: 18]:	Selon le moment de la 12h et 18h. # On calcule 1a of from math import des compute dista try: R = 6373. lat1 = ra lon1 = ra lat2 = ra lon2 = ra dlat = la dlon = lo a = sin (d c = 2 * a return R except KeyErr print (row data['trip_dist'] sns.jointplot(x=' <seaborn.axisgrid -="" -<="" 300000="" 3000000="" 3500000="" td=""><td>heure de la journée journée où on prend le journée où or syrt, ince (row): outins (row, syrt, ince (row): outins (row, d'dropo dians (row, d'dropo dians (row, pricku) dians (row, p</td><td>etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) ff_longitude']) p_latitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 2813558f348> ip_duration', d 'spead']!=0) & ip_duration', d 'spead']!=0) & ip_duration', d 2812e6le2c8> ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants m</td><td>e point de départe de point de départe de la contre la distance e la contre la contre la distance e la contre la distance e la contre la contre la distance e la contre la con</td><td>a remarqué plus la voit sur ce graph strajets de 0km, byenne de ces tra 130km/h est dû a vitesse moy duration']/36 a vitesse du tra durée du tra dur</td><td>naut avec la distri lique que des traj ce qui est imposs ajets et on supprir au fait que c'est la renne des traj (00<3)].shape (0) & (data['s rajet en fonce esque toute la dor (c) l'esque toute la</td><td>bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429</td></seaborn.axisgrid>	heure de la journée journée où on prend le journée où or syrt, ince (row): outins (row, syrt, ince (row): outins (row, d'dropo dians (row, d'dropo dians (row, pricku) dians (row, p	etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) ff_longitude']) p_latitude']) p_longitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> ip_duration', d 2813558f348> ip_duration', d 'spead']!=0) & ip_duration', d 'spead']!=0) & ip_duration', d 2812e6le2c8> ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants mentionr allono 1200 ort à la distance poude linéarité stricte daberrants m	e point de départe de point de départe de la contre la distance e la contre la contre la distance e la contre la distance e la contre la contre la distance e la contre la con	a remarqué plus la voit sur ce graph strajets de 0km, byenne de ces tra 130km/h est dû a vitesse moy duration']/36 a vitesse du tra durée du tra dur	naut avec la distri lique que des traj ce qui est imposs ajets et on supprir au fait que c'est la renne des traj (00<3)].shape (0) & (data['s rajet en fonce esque toute la dor (c) l'esque toute la	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429
12]: 13]: 14]: 15]: 17]: 18]:	Selon le moment de la 12h et 18h. # On calcule la de from math import def compute dista try: R = 6373. lat1 = ra lon1 = ra lon2 = ra dlat2 = ra lon2 = ra dlat3 = ra return R except KeyErr print (row data['trip_dist'] sns.jointplot(x=' <seaborn.axisgrid #="" =="" data['spead']="data[(data['spead']" data[(data['spead']="ra" dis="" fois="" la="" lon2="ra</td" une=""><td>heure de la journée journée où on prend le listance à vol d' sin, co, sagrt, since (row): dians (row] 'dropo dians (row] 'dropo dians (row] 'dropo dians (row] 'dropo dians (row] 'pickuj en la suite. Pour se fa gale à Okm/h ou supé par la</td><td>platitude dropoff at a distance por de linéarité stricte du trance du trance</td><td>e point de départe de point de départe la distance e la distance e la plus haut tout e la curation 1/3 e</td><td>a remarqué plus la voit sur ce graph strajets de 0km, byenne de ces tra 130km/h est dû a vitesse moy duration']/36 a vitesse du tra durée du tra dur</td><td>naut avec la distri lique que des traj ce qui est imposs ajets et on supprir au fait que c'est la renne des traj (00<3)].shape (0) & (data['s rajet en fonce esque toute la dor (c) l'esque toute la</td><td>bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429</td></seaborn.axisgrid>	heure de la journée journée où on prend le listance à vol d' sin, co, sagrt, since (row): dians (row] 'dropo dians (row] 'dropo dians (row] 'dropo dians (row] 'dropo dians (row] 'pickuj en la suite. Pour se fa gale à Okm/h ou supé par la	platitude dropoff at a distance por de linéarité stricte du trance	e point de départe de point de départe la distance e la distance e la plus haut tout e la curation 1/3 e	a remarqué plus la voit sur ce graph strajets de 0km, byenne de ces tra 130km/h est dû a vitesse moy duration']/36 a vitesse du tra durée du tra dur	naut avec la distri lique que des traj ce qui est imposs ajets et on supprir au fait que c'est la renne des traj (00<3)].shape (0) & (data['s rajet en fonce esque toute la dor (c) l'esque toute la	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429
12]: 13]: 14]: 15]: 17]: 18]:	Selon le moment de la 12h et 18h. # On calcule la ofrom math import des compute dista try: R = 6373. latl = ra lon2 = return R except KayEnr print (row data['trip_dist'] sns.jointplot(x=' <seaborn.axisgrid #="" =="" data['data['spead']="data['data['spead']" data['spead']="data['</td" dis="" fois="" la="" une=""><td>heure de la journée journée où on prand le journée où on prand le journée où on prand le listance à vol d' sin, cos, sqrt, ince (row): 0 dians (row['dropondians (row['picku, dians (row['picku, double data (row]') double data (row] double data (row) double da</td><td>etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) flongitude']) platitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> are eux étaient aber viron 416h). De plu iire, nous allons cal rieure à 130km/h. L seau calculée, /(data['trip_d 'spead']!=0) & ip_duration', d 'scatter') =r'durée du tr 2812e6le2c8> dip_duration' all distance pou de linéarité stricte aberrants mentionr 3) & (data['sp data_bis['stord all intoatoa and in</td><td>rice. Comme on l'a atalia data, kinda atalia datalia datalia datalia datalia atalia datalia datalia datalia atalia datalia</td><td>a remarqué plus la vitage de construir de la durée d'un tra la dur</td><td>naut avec la distribique que des trajice qui est impossajets et on supprir au fait que c'est la renne des trajicolos (data['stajet en fonction of the seque toute la doron of the seque toute la doron</td><td>bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429</td></seaborn.axisgrid>	heure de la journée journée où on prand le journée où on prand le journée où on prand le listance à vol d' sin, cos, sqrt, ince (row): 0 dians (row['dropondians (row['picku, dians (row['picku, double data (row]') double data (row] double data (row) double da	etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) flongitude']) platitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> are eux étaient aber viron 416h). De plu iire, nous allons cal rieure à 130km/h. L seau calculée, /(data['trip_d 'spead']!=0) & ip_duration', d 'scatter') =r'durée du tr 2812e6le2c8> dip_duration' all distance pou de linéarité stricte aberrants mentionr 3) & (data['sp data_bis['stord all intoatoa and in	rice. Comme on l'a atalia data, kinda atalia datalia datalia datalia datalia atalia datalia datalia datalia atalia datalia	a remarqué plus la vitage de construir de la durée d'un tra la dur	naut avec la distribique que des trajice qui est impossajets et on supprir au fait que c'est la renne des trajicolos (data['stajet en fonction of the seque toute la doron	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429
12]: 13]: 14]: 15]: 17]: 18]:	Selon le moment de la 12h et 18h. # On calcule la afrom math import def compute dista try: R = 6373. lat1 = ra lond = la dion = lo a = sin(d c = 2 * a return R except KeyErr print(row data['trip_dist'] sns.jointplot(x=') Seaborn.axisgrid **Seaborn.axisgrid **	heure de la journée journée où on prend le journée où or on journée où on prend le journée où on journée journée où or journée journée ou or journée journée	ataxi, la durée moy ataxi, la durée moy ataxi, la durée moy actau entre la atan2, radians atan2, radians atain2, radian	rice. Comme on l'a ata=data, kind ata=data,	art et la de coart et la de	naut avec la distri idque que des traj cique que des traj ce qui est imposs ajets et on supprir au fait que c'est la venne des traj (00<3)].shape (0) & (data['s rajet en fonct resque toute la dor lata_bis['trip N N N N N N N N N N N N N N N N N N N	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429
12]: 13]: 14]: 15]: 17]: 18]:	Selon le moment de la 12h et 18h. # On calcule la c from math import def compute dista lon1 = ra lon2 = ra dlat = la dolon = lo a = sujn(d c = 2 * a return R except KeyErr print(row data['trip_dist'] sns.jointplot(x=' <seaborn.axisgrid #="" and="" autorisée="" book="" dans="" data['data['spead']="" data['spead']="d" de="" dis="" fois="" l'e<="" l'en="" l'état="" la="" moyenne="" seaborn="" soucous="" sune="" td="" the="" une="" vitese="" é=""><td>heure de la journée heure de la journée journée où on prend le listance à vol d' sin, cos, sqrt, noe (row): 0 dians (row ('dropo dians (row ('dropo dians (row ('picku) dians (</td><td>e taxi, la durée moy e taxi, la durée moy ff latitude'] ff latitude'] p latitude'] p latitude'] p latitude'] p longitude'] scripturation', d carieure à 130km/h. I seau calculée, / (data['trip, d 'scripturée du tr latitude dropoff 40.767937 40.783564 40.76939 40.793209 ead']<130) & (pickup_datetim p latitude dropoff 40.768939 40.793209 is ['pickup_mon g'pickup_day'] data_bis ['stor me'></td><td>rote. Commerce on l'a rants. En effet, on son remarque des culer la vitesse ma e choix du seuil à on calcule 1. uration']/360 (data['trip_c ata=data[(data ajet', title=. longitude dropoff_da ata=data['dropoff_da rants. En effet, on son remarque des culer la vitesse ma e choix du seuil à on calcule 1. uration']/360 (data['trip_c ata=data[(data ajet', title=. longitude dropoff ranse plus haut tout e e', 'dropoff_da ranget', title=. longitude dropoff ranget', title=.</td><td>cart et la de cart et la de ca</td><td>naut avec la distri idique que des traj ce qui est imposs ajets et on supprir au fait que c'est la renne des traj colo (3)].shape lata_bis['trip N N N N N N N N N N N N N N N N N N N</td><td>bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429</td></seaborn.axisgrid>	heure de la journée heure de la journée journée où on prend le listance à vol d' sin, cos, sqrt, noe (row): 0 dians (row ('dropo dians (row ('dropo dians (row ('picku) dians (e taxi, la durée moy ff latitude'] ff latitude'] p latitude'] p latitude'] p latitude'] p longitude'] scripturation', d carieure à 130km/h. I seau calculée, / (data['trip, d 'scripturée du tr latitude dropoff 40.767937 40.783564 40.76939 40.793209 ead']<130) & (pickup_datetim p latitude dropoff 40.768939 40.793209 is ['pickup_mon g'pickup_day'] data_bis ['stor me'>	rote. Commerce on l'a rants. En effet, on son remarque des culer la vitesse ma e choix du seuil à on calcule 1. uration']/360 (data['trip_c ata=data[(data ajet', title=. longitude dropoff_da ata=data['dropoff_da rants. En effet, on son remarque des culer la vitesse ma e choix du seuil à on calcule 1. uration']/360 (data['trip_c ata=data[(data ajet', title=. longitude dropoff ranse plus haut tout e e', 'dropoff_da ranget', title=. longitude dropoff ranget', title=.	cart et la de ca	naut avec la distri idique que des traj ce qui est imposs ajets et on supprir au fait que c'est la renne des traj colo (3)].shape lata_bis['trip N N N N N N N N N N N N N N N N N N N	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429
12]: 13]: 14]: 15]: 20]: 20]:	Selon le moment de la 12h et 18h. # On calcule la of from math import def compute_dist try: R = 6373. lat1 = ra lond = ra lon	heure de la journée journée où on prend le journée où on prend le listance à vol d' sin, cos, sqrt, noce (row): 0 dians (row['dropo dians (row['pickundians (row['p	etaxi, la durée moy oiseau entre la atan2, radians ff_latitude']) ff_longitude']) plongitude']) plongitude']) plongitude']) plongitude']) plongitude']) mpute_distance ip_duration', d 2813558f348> are eux étaient aber wiron 416h). De plu ire, nou 3 allons cal rieure à 130km/h. L seau calculée, /(data['trip_d' 'spead']!=0) & seau calculée, /(data['trip_d' 'spead']!=0) & seau calculée, //(data['trip_d' 'spead']!=0) & seau calculée, // (data['trip_d' 'spead']!=0) & ip_duration', d ip	e point de dé e point de dé , axis=1) lata=1) ata=data, kinde ince. Comme on l'a rants. En effet, on son remarque de (culer la vitesse ma. e choix du seuil à con calcule 1. uration' /360 (data['trip_c' ata=data[(data ajet', title=. longitude dropoff /73,964630	aremarqué plus l'on / 2)**2 aremarqué plus l'on / 2)**2 aremarqué plus l'on / 2)**2 aremarqué plus l'on / 2)**3 aremarqué plus l'on / 2)**3 aremarqué plus l'on / 2) are	naut avec la distri idue que des traj ce qui est imposs ajets et on supprir au fait que c'est la renne des traj renne des traj (0) & (data['s rajet en fonct lata_bis['trip nd_fwd_flag trip_ N N N N N N N N N N N N N N N N N N N	bution de la durée ets de moins de 10 sible. On va donc mera les trajets avera vitesse maximale dets [0]/data.shape spead']!=0) & tion de la dis tion de la dis curation de la dis curation pickup_n 455 663 2124 429
12]: 13]: 14]: 17]: 18]: 20]:	Selon le moment de la 12h et 18h. # On calcule la c' from math import def compute_dista try:	heure de la journée journée où on prend le listance à vol d' sin, cos, sigrt, nice (row): Odians (row ('dropo dians (row ('dropo dians (row ('dropo dians (row ('dropo dians (row ('picku) dians (row ('dropo dians (row ('picku) dians (row ('picku	e taxi, la durée moy ciseau entre l' atan2, radians ff_latitude']) ff_longitude']) platitude']) platitude']) plongitude') s(latl) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> areux étain, be plu ire, nous allons cal rieure à 130km/h. L seau calculée, / (data['trip_d' 'spead']!=0) & 1000 1200 arrapport à la distance reviron étain, be plu ire, nous allons cal rieure à 130km/h. L seau calculée, / (data['trip_d' 'spead']!=0) & 150 175 200 arrapport à la distance reviron étain taber reviron étain de dropoff 40.767937 40.767937 40.767937 40.738564 40.763939 40.719971 40.738564 40.763939 40.719971 40.738564 40.763939 40.719971 40.738564 40.763939 40.719971 40.738564 40.763939 40.719971 40.738564 40.76937 40.738564 40.763939 40.719971 40.7191 chauli inte4 reviron de	e point de dé e point de dé i lat2) * sin(di , axis=1) ata=data, kindi	ins de 3h avec ut la durée d'un tra et la durée d'u	naut avec la distri iduation. naut avec la district iduation. naut avec la	bution de la durée ets de moins de 16 dible. On va donc mera les trajets ave a vitesse maximale de 18 dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc di
12]: 13]: 14]: 15]: 21]: 22]: 23]:	Selon le moment de la 12h et 18h. # On calcule la candidate de from math import def compute_dista try: local cule de la de from math import def compute_dista try: local cule la candidate la dolon = local cule la candidate la dolon = local candidate la local candidate la dolon = local candidate la local candidate l	heure de la journée journée où on prend le journée où or journée dians (row ['dropo dians (row ['d	etaxi, la durée moy etaxi, la durée moy olseau entre la atan2, radians ff latitude']) platitude']) platitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> are usé taient aberu irien 4160h. De pu irieure à 130km/h. L seau calculée, cladata['spead']!=0) & platitude dropoff 40.767937 40.738564 40.763939 40.763939 40.763939 40.763939 40.76364 40.76364 40.76364 40.76364 40.76364 40.763639	re point de dér e point de dér i, axis=1) ata=1) ata=data, kindi kindi ata=data, kindi con calcule la viresse mone de louler la viresse mone de louler la viresse mone de louler la vires de la vires	ins de 3h avec ut la durée d'un tra et la durée d'u	naut avec la distri iduation. naut avec la district iduation. naut avec la	bution de la durée ets de moins de 16 dible. On va donc mera les trajets ave a vitesse maximale de 18 dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc di
12]: 13]: 14]: 15]: 21]: 22]: 23]:	Selon le moment de la 12h et 18h. # On calcule la of from math import def compute dista try: loca e la dista de compute dista try: loca e sin didon = lo a = sin did	heure de la journée journée où on prend le journée où or journée dians (row ['dropo dians (row ['d	etaxi, la durée moy etaxi, la durée moy olseau entre la atan2, radians ff latitude']) platitude']) platitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> are usé taient aberu irien 4160h. De pu irieure à 130km/h. L seau calculée, cladata['spead']!=0) & platitude dropoff 40.767937 40.738564 40.763939 40.763939 40.763939 40.763939 40.76364 40.76364 40.76364 40.76364 40.76364 40.763639	re point de dér e point de dér i, axis=1) ata=1) ata=data, kindi kindi ata=data, kindi con calcule la viresse mone de louler la viresse mone de louler la viresse mone de louler la vires de la vires	ins de 3h avec ut la durée d'un tra et la durée d'u	naut avec la distri iduation. naut avec la district iduation. naut avec la	bution de la durée ets de moins de 16 dible. On va donc mera les trajets ave a vitesse maximale de 18 dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc di
12]: 13]: 14]: 15]: 21]: 22]: 23]:	Selon le moment de la 12h et 18h. # On calcule la ce from math import def compute, ista try:	heure de la journée journée où on prend le journée où or journée dians (row ['dropo dians (row ['d	etaxi, la durée moy etaxi, la durée moy olseau entre la atan2, radians ff latitude']) platitude']) platitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> are usé taient aberu irien 4160h. De pu irieure à 130km/h. L seau calculée, cladata['spead']!=0) & platitude dropoff 40.767937 40.738564 40.763939 40.763939 40.763939 40.763939 40.76364 40.76364 40.76364 40.76364 40.76364 40.763639	re point de dér e point de dér i, axis=1) ata=1) ata=data, kindi kindi ata=data, kindi con calcule la viresse mone de louler la viresse mone de louler la viresse mone de louler la vires de la vires	ins de 3h avec ut la durée d'un tra et la durée d'u	naut avec la distri iduation. naut avec la district iduation. naut avec la	bution de la durée ets de moins de 16 dible. On va donc mera les trajets ave a vitesse maximale de 18 dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc di
12]: 13]: 14]: 15]: 21]: 22]: 23]:	Selon le moment de la 12h et 18h. # On calcule la a from math import def compute dista try: R = 6373. lat1 = ra lon1 = ra lon2 = ra don1 = ra lon2 = ra don2 = ra don3 = sin.(d c = 2 * ra don = retrum R except KeyErr print (row data['trip_dist'] sns.jointplot(row caeaborn.axisgrid # Une fois la di data ('trip_dist') sns.jointplot(row source) Source	heure de la journée journée où on prend le journée où or journée dians (row ['dropo dians (row ['d	etaxi, la durée moy etaxi, la durée moy olseau entre la atan2, radians ff latitude']) platitude']) platitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> are usé taient aberu irien 4160h. De pu irieure à 130km/h. L seau calculée, cladata['spead']!=0) & platitude dropoff 40.767937 40.738564 40.763939 40.763939 40.763939 40.763939 40.76364 40.76364 40.76364 40.76364 40.76364 40.763639	re point de dér e point de dér i, axis=1) ata=1) ata=data, kindi kindi ata=data, kindi con calcule la viresse mone de louler la viresse mone de louler la viresse mone de louler la vires de la vires	ins de 3h avec ut la durée d'un tra et la durée d'u	naut avec la distri iduation. naut avec la district iduation. naut avec la	bution de la durée ets de moins de 16 dible. On va donc mera les trajets ave a vitesse maximale de 18 dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc di
12]: 13]: 14]: 15]: 21]: 22]: 23]:	Selon le moment de la 12h et 18h. # On calcule la d from math import def compute dista try: R = 6373. lat1 = ra lond = ra lon	beure'de la journée pournée où on prend le pournée où ou ser pournée où d'in prend pournée ou d'ar le pournée de trajet par rappe pournée ou le prend le pournée ou d'ori par la data, apply(pournée) prend le trajet par rappe pournée ou le prend le pournée ou le pournée o	etaxi, la durée moy etaxi, la durée moy olseau entre la atan2, radians ff latitude']) platitude']) platitude']) plongitude']) s(lat1) * cos(rt(1 - a)) mpute_distance ip_duration', d 2813558f348> are usé taient aberu irien 4160h. De pu irieure à 130km/h. L seau calculée, cladata['spead']!=0) & platitude dropoff 40.767937 40.738564 40.763939 40.763939 40.763939 40.763939 40.76364 40.76364 40.76364 40.76364 40.76364 40.763639	re point de dér e point de dér i, axis=1) ata=1) ata=data, kindi kindi ata=data, kindi con calcule la viresse mone de louler la viresse mone de louler la viresse mone de louler la vires de la vires	ins de 3h avec ut la durée d'un tra et la durée d'u	naut avec la distri iduation. naut avec la district iduation. naut avec la	bution de la durée ets de moins de 16 dible. On va donc mera les trajets ave a vitesse maximale de 18 dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc mera les trajets ave a vitesse maximale dible. On va donc di
12]: 13]: 14]: 15]: 21]: 22]: 23]:	Seion le moment de la 12t et 18t. # On calcule la crommath import In en crommath	meure de la journée journée où on prend le sistance à vol d' dians (row ('dropo dians ('dropo dian	platitude dispersation of de linéarité stricte du train des raineure à 130km/h. L'escau calculée, / (data['trip de linearité stricte dipendration', de l'escau calculée, / (data['trip de linearité stricte dipendration', de l'escau calculée, / (data['trip de l'es	pickup_latitue "rants_En effet, on son remarque de culer la vitesse ma e choix du seuil à on calcule la uration']/3600 (data['trip_o' data=data[(data ajet', title= longitude dropoff_da ajet', title= longitude dropoff_da ajet', title= longitude dropoff_da ajet', title= longitude dropoff_da ajet', data=data[(data ajet', title= longitude dropoff_da ajet', data=data[(data ajet', title= longitude dropoff_da ajet', data=data[(data ajet', data=data] ajet', data=data[(data ajet', data=data] ajet', data=data] bis['spec' e','dropoff_da ajet', data=data] ajet', data=data] bis['spec' e','dropoff_da ajet', data=data ajet', d	art et la de lon / 2)**2 lon / 2)**3 lon	naut avec la distri iduation. naut avec la district iduation. naut avec la	bution de la durée ets de moins de 10 ible. On va donc mera les trajets ave a vitesse maximale fets [0]/data.shape spead']!=0) & fion de la dis ickup_time','s duration pickup_n 455 663 2124 429 435

In []: **from sklearn import** metrics

Random Forest

In [17]: data_bis = data.copy()

def get_day(x):

data_bis.head()

data_bis.info()

pickup_month

pickup hour

Out[20]: RandomForestRegressor()

7000

6000

5000 4000

3000

2000

1000

démarche.

In [23]: y_pred = rfc.predict(X_test)

In [24]: **from sklearn import** metrics

MSE: 90637.30564498388 RMSE: 5.0176717056647755

memory usage: 94.1 MB

rfc.fit(X_train, y_train)

plt.scatter(y_test,y_pred)

pickup_day

trip_dist

d'], axis=1, inplace=True)

1

1

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1450565 entries, 0 to 1458643

passenger_count 1450565 non-null int64
pickup_longitude 1450565 non-null float64
pickup_latitude 1450565 non-null float64
dropoff_longitude 1450565 non-null float64
dropoff_latitude 1450565 non-null float64
store_and_fwd_flag 1450565 non-null uint8
trip_duration 1450565 non-null int64

trip_duration 1450565 non-null int64

In [19]: from sklearn.model_selection import train_test_split

Out[23]: <matplotlib.collections.PathCollection at 0x23323a34688>

2500 5000 7500 10000 12500 15000 17500 20000

In []: # Les variables importantes dans le modèle random forest

sorted_idx = result.importances_mean.argsort()

ax.boxplot(result.importances[sorted_idx].T,

ax.set_title("Permutation Importances (test set)")

"timeout or by a memory leak.", UserWarning

import matplotlib.pyplot as plt

ker timeout or by a memory leak.

fig, ax = plt.subplots()

fig.tight_layout()

plt.show()

from sklearn.inspection import permutation_importance

result = permutation_importance(rfc, X_test, y_test, n_repeats=10,

vert=False, labels=X test.columns[sorted idx])

approche qui est gourmande en terme de calcul. Or, mon ordinateur n'est pas assez puissant.

print('MSE:', metrics.mean_squared_error(y_test, y_pred))

print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred))/60)

dtypes: category(3), float64(5), int64(2), uint8(1)

X_train, X_test, y_train, y_test = train_test_split(

Data columns (total 11 columns):

6)]

()

0

1

2

3

Out[17]:

In [18]:

print('MSE:', metrics.mean squared error(y test, predictions))

data_bis['pickup_day'] = data_bis['pickup_date'].apply(get_day)

40.767937

40.738564

40.763939

40.719971

40.793209

data_bis['store_and_fwd_flag'] = data_bis['store_and_fwd_flag'].astype("category")

data_bis['pickup_month'] = data_bis['pickup_month'].astype("category")
data_bis['pickup_hour'] = data_bis['pickup_hour'].astype("category")
data_bis['pickup_day'] = data_bis['pickup_day'].astype("category")

1450565 non-null category

1450565 non-null category

1450565 non-null category 1450565 non-null float64

L'écart moyen de prédiction est égal à 6.71mn pour ce modèle.

-73.982155

-73.980415

-73.979027

-74.010040

-73.973053

print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions))/60)

data bis = data bis[(data bis['spead']<130) & (data bis['spead']!=0) & (data bis['trip duration']/3600<

return datetime.datetime(int(x.split('-')[0]), int(x.split('-')[1]), int(x.split('-')[2])).weekday

data_bis.drop(['id','vendor_id','pickup_datetime','dropoff_datetime','pickup_date','pickup_time','spea

passenger_count pickup_longitude pickup_latitude dropoff_longitude dropoff_latitude store_and_fwd_flag trip_duration pickup_montl

40.765602

40.731152

40.710087

40.706718

40.782520

0;

0(

0

۵,

0;

455

663

2124

429

435

-73.964630

-73.999481

-74.005333

-74.012268

-73.972923

data_bis.drop('trip_duration',axis=1), data_bis['trip_duration'], test_size=0.2, random_state=101)

Par rapport à la régression linéaire, on obtient un meilleur résultat avec un écart moyen de prédiction égal à 5.01mn. Cependant dans les deux cas, on voit que la prédiction en fonction de la valeur réelle ne suit pas une distribution parfaitement linéaire. Cela nous suggére qu'on peut encore gagner en précision avec d'autres modèles. On ne le fera pas dans ce notebook car l'objectif étant seulement d'illustrer une

random_state=42, n_jobs=-1)

C:\Users\mouha\Anaconda3\lib\site-packages\joblib\externals\loky\process_executor.py:706: UserWarnin g: A worker stopped while some jobs were given to the executor. This can be caused by a too short wor

Ici, l'objectif était de déterminer les variables influant grandement sur la prédiction de la durée du trajet. Pour cela, on utilise l'approche appelée Permutation feature importance qui consiste à mesurer directement l'importance des variables en observant comment le

remaniement aléatoire (préservant ainsi la distribution de la variable) de chaque variable influence la performance du modèle. C'est une

data_bis['store_and_fwd_flag'] = pd.get_dummies(data_bis['store_and_fwd_flag'],drop_first=True)