Trabajo Practico C.C.A.

GRUPO 16

Gustavo Chac
Francisco Piccione
Luciano Bustamante

21/05/2022

Contents

Introducción **2**

Ejercicio 1 **3**
Solución Propuesta: . . . 4
Supuestos . . . 4
Datasets . . . 4
Solución 1.A: . . . 5
Solución 1.B: . . . 6
Supuestos: . . . 6

Ejercicio 2 **8**
Solución Propuesta: . . . 9
Dataset: . . . 9
Depuración de Dataset: . . . 10
Graficos de Exploratorio . . . 10
Summary de variables de interés . . . 13
Solución propuesta 2.A: . . . 14
Metricas distancia y tiempo . . . 14
Probabilidad de reasignación . . . 15
Solución propuesta 2.B: . . . 16
Solución propuesta 2.C: . . . 17
Supuestos: . . . 17
Generacion de variable . . . 17
Segmentación y comparación de población . . . 17

Introducción

En el presente trabajo practico simularemos una situación de la vida real de la empresa de envíos ("ElMorfi"). El trabajo consiste en abordar dos ejercicios cada uno con caracteristicas y objetivos distintos.

Elegimos como plataforma de desarrollo RMarkdown donde desarrollaremos el codigo necesario y además lo dejaremos preparado para emitir un reporte amigable para el consumidor final (con el codigo oculto).

Para el presente trabajo utilizamos las siguientes librerias/packages:

- dplyr
- lubridate
- ggplot2
- data.table
- knitr
- stringr
- ggcorrplot

Además utilizaremos los siguientes datasets:

- EJERCICIO 1
  - **ordenes**
  - **ordenes_agrupadas**
- EJERCICIO 2
  - **dataset**

Ejercicio 1

En ElMorfi existe la posibilidad de agrupar en un paquete dos órdenes de distintos clientes si ambos realizaron su pedido al mismo local aproximadamente al mismo tiempo. En el caso de un pedido agrupado, un único repartidor recoje ambas órdenes aproximadamente al mismo momento de la tienda y procede a la dirección del primer cliente a entregar la primer órden y posteriormente a la dirección del segundo cliente. En particular durante horas pico, agrupar pedidos puede incrementar la capacidad de la flota llevando a menores tiempos de entrega (en promedio).

Suponga que cuenta con dos data.frame (adjunto se enviaron dos CSV con ejemplos de la estructura de dichas tablas), uno llamado **ordenes** el cual contiene información de las órdenes de todos los clientes y otro llamado **ordenes_agrupadas** el cual contiene información solamente de las órdenes que fueron agrupadas.

El objetivo es, utilizando sintaxis de R base, dplyr/tidyverse, data.table o cualquier libreria que consideren relevante definir indicar los comandos que respondan de forma mas precisa las siguientes preguntas:

- A. Dadas las dos tablas presentadas anteriormente, se busca comparar el porcentaje de ordenes que fueron agrupadas en la ciudad con el city code GLV contra las de la ciudad con el city code PLY en el 1ro de noviembre de 2021. Aclaración: las órdenes no agrupadas no deben ser consideradas como agrupadas.
- B. En segundo lugar se busca calcular la velocidad promedio de cada repartidor desde que recojen los pedidos hasta que entregan los mismos, para cada ciudad en los últimos 30 días. En caso de órdenes agrupadas- considerar solo la trayectoria al primer punto de entrega, siendo el mismo aquel con menor distancia entre la dirección donde se retira el pedido y la dirección donde se entrega. Las columnas pd_dist, pickup_time y enters_delivery representan la distancia entre direcciones retiro y entrega, la fecha en que el repartidor retira el pedido y la fecha en que el mismo entra en la cercanía de la dirección de entrega, respectivamente.

Solución Propuesta:

Supuestos

Para el desarrollo del presente ejercicio se establecen los siguientes supuestos:

1. Los datasets **'ordenes'** y **'ordenes_agrupadas'** se vinculan a través de order_id.
2. El dataset **'ordenes'** contiene todas las ordenes, es decir, ordenes agrupadas y no agrupadas.
3. El dataset **'ordenes_agrupadas'** sólo contiene las ordenes agrupadas.
4. Una orden del dataset **'ordenes'** es agrupada si el order_id se encuentra en el dataset de **'ordenes_agrupadas'**.

Datasets

Mostramos la forma de cada dataset:

- **Ordenes**

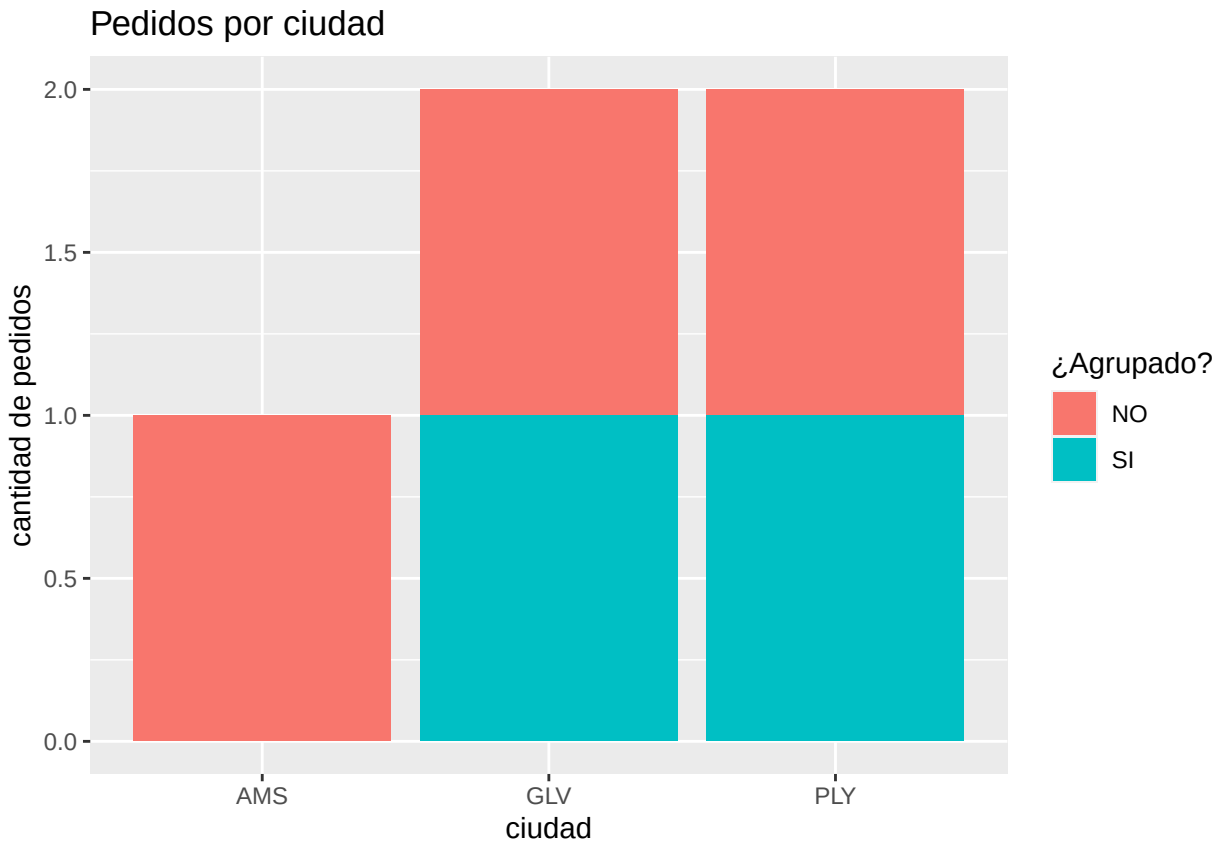
order_id	city_code	store_id	creation_time	pickup_time	enters_delivery	pd_dist	final_status
4596184593	AMS	3372	2021-11-01 23:23:04	2021-11-01 23:33:52	2021-11-01 23:43:17	1503	DeliveredStatus
4569203459	GLV	8844	2021-11-02 11:13:23	NULL	NULL	2004	CanceledStatus
4596020394	GLV	99103	2021-11-01 20:56:01	2021-11-01 21:03:22	2021-11-01 21:11:20	1842	DeliveredStatus
4592303948	PLY	12287	2021-11-01 16:49:18	2021-11-01 16:55:05	2021-11-01 16:55:35	5	DeliveredStatus
4592303949	PLY	12287	2021-11-01 16:50:30	2021-11-01 16:59:45	2021-11-01 17:12:48	1562	DeliveredStatus

- **Ordenes Agrupadas**

order_id	bundle_id	is_bundled	is_unbundled
4395449294	87632847	TRUE	FALSE
4596020394	87632847	TRUE	FALSE
4339452836	87632239	TRUE	TRUE
4592303948	87632239	TRUE	TRUE
4395529454	87633554	TRUE	FALSE

Solución 1.A:

A continuación veamos la comparación de pedidos totales en ciudades, segmentado por envios agrupados:

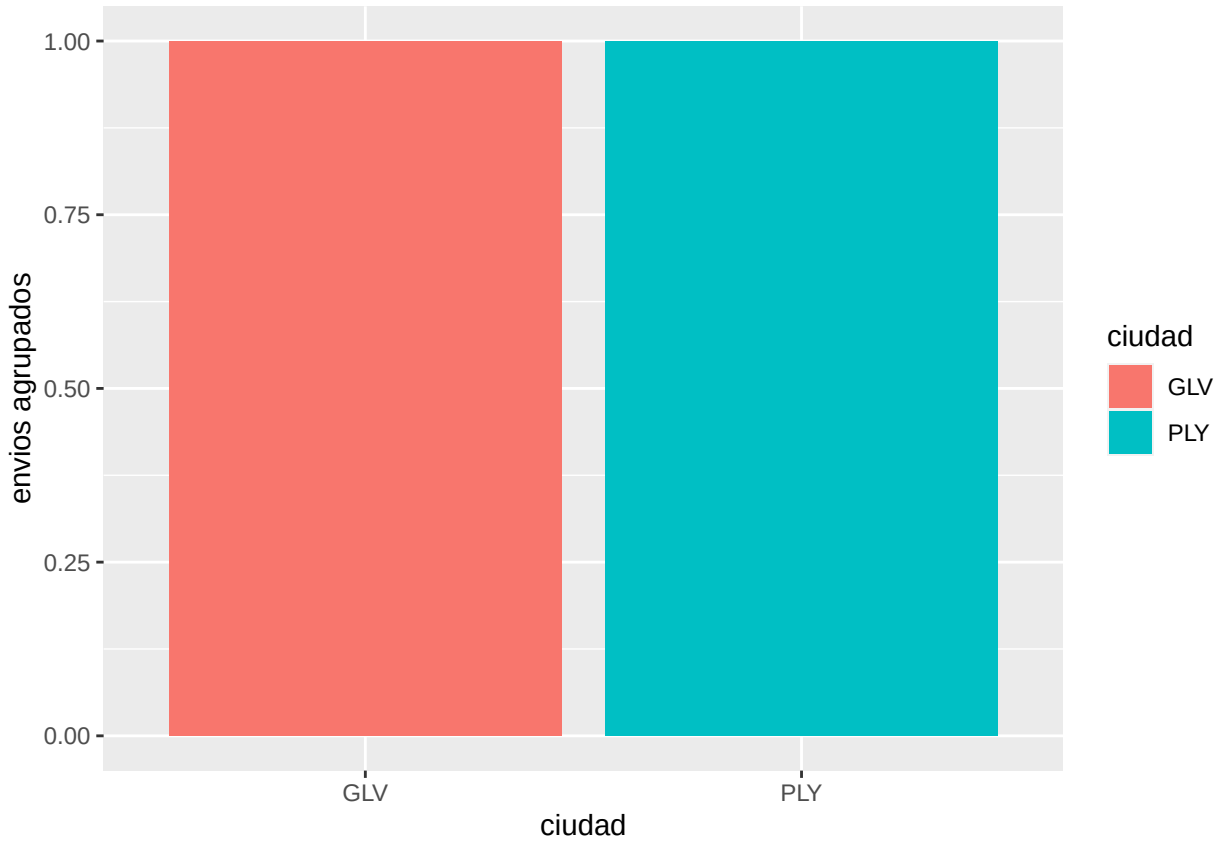


Ahora veamos la comparación de pedidos entre ciudades:

ciudad	envios totales	agrupados	(%) agrupados
GLV	2	1	50%
PLY	2	1	50%
AMS	1	0	0%

Por ultimo comparemos las pedidos agrupados entre las ciudades GLV y PLY:

ciudad	envios agrupados	(%) envios agrupados
GLV	1	50%
PLY	1	50%

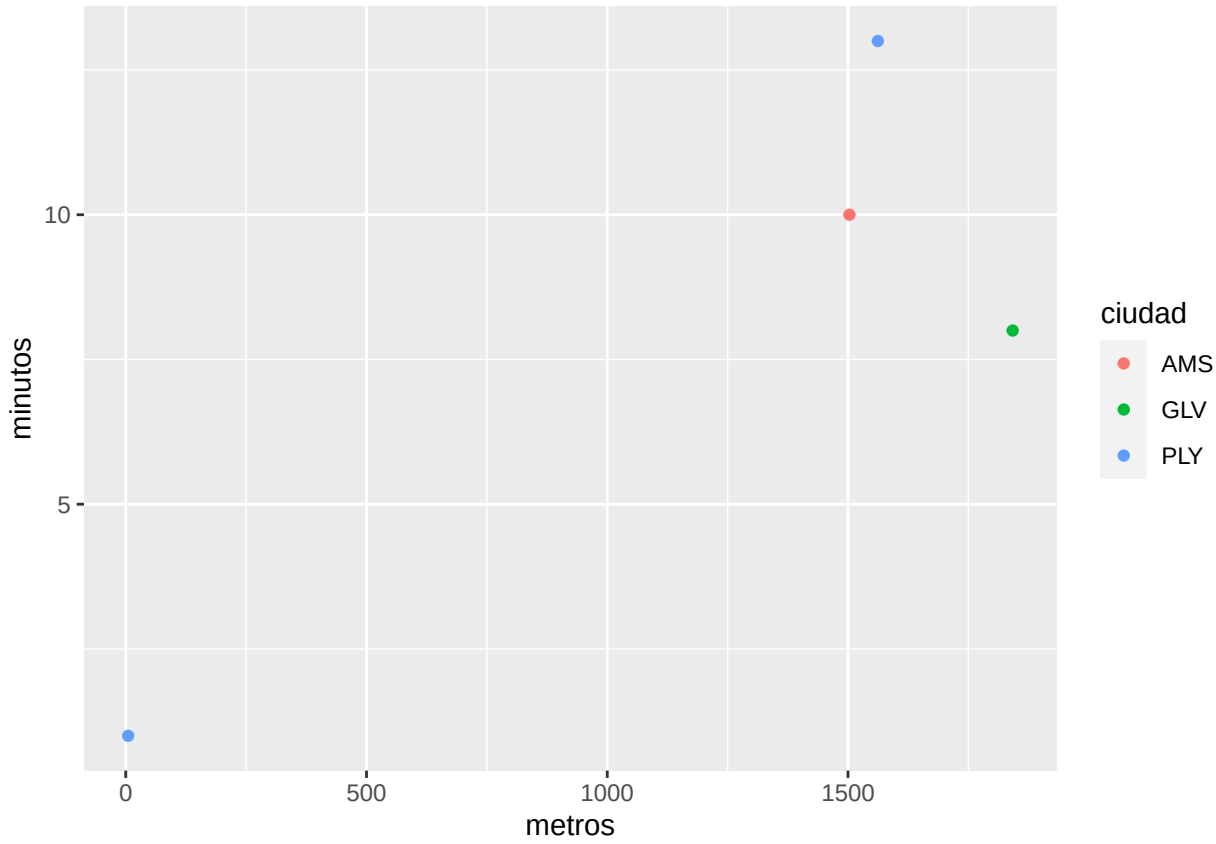


Solución 1.B:

Supuestos:

- Se toma como ultimos 30 dias desde el 01/11/2021 hasta 30/11/2021.
- Asumimos que pd_dist expresa metros y lo trataremos como tal.
- Generamos la siguiente variable para medir la velocidad $\frac{kilometros}{horas}$
- Dado que no existe una variable que nos permita reconocer al repartidor, asumimos que el pedido se hace sobre velocidad de envio por ciudad.

A continuación mostramos un grafico de disperción de la velocidad de los envíos (metros y minutos).



Por último evaluamos algunas metricas interesantes.

tiempo (minutos)	distancia (metros)	velocidad (kmh)
Min. : 1.00	Min. : 5	Min. : 0.500
1st Qu.: 6.25	1st Qu.:1128	1st Qu.: 5.450
Median : 9.00	Median :1532	Median : 8.247
Mean : 8.00	Mean :1228	Mean : 7.791
3rd Qu.:10.75	3rd Qu.:1632	3rd Qu.:10.588
Max. :13.00	Max. :1842	Max. :14.169

Ejercicio 2

Adjunto van a encontrar un archivo llamado dataset_ejercicio2 el cual contiene información de órdenes ocurridas durante una semana.

A. En base a los datos provistos, provean visibilidad sobre las siguientes métricas:

1. "Tiempo de entrega al cliente". Cuánto esperó el cliente su pedido.
2. "Distancia de entrega". Distancia total recorrida por los repartidores.
3. "Probabilidad de reasignación"

B. Con respecto a las reasignaciones, obervan alguna tendencia/correlación entre esta métrica y alguno de los campos del dataset?

C. Una vez que el repartidor llega a la tienda, el tiempo hasta que la orden es retirada (waiting_time_at_pickup), puede presentar alta variabilidad. Con los datos provistos, presentar un análisis mostrando esto mismo. Pueden identificar alguna tendencia clara con respecto a los casos con mayor tiempo de demora? Cuales serían las posibles causas de que un repartidor esté esperando mucho tiempo para retirar un pedido?

Solución Propuesta:

Dataset:

```
##         id    final_status store_address_id customer_id courier_id
## 1 53223617 DeliveredStatus            19434     3291674   13762181
## 2 52701851 DeliveredStatus            18300     6558338   18225430
## 3 52496295 DeliveredStatus            23518     6128839    6212245
## 4 53145468  CanceledStatus            16593     6156985    8046350
## 5 52685424 DeliveredStatus            18300     8403200    7290210
## 6 52719455  CanceledStatus            30640    10659079   15239373
##         vertical is_food transport number_of_assignments total_real_distance
## 1 WALL - Partner    TRUE       CAR                     1               7.789
## 2 WALL - Partner    TRUE   BICYCLE                     1               4.751
## 3 WALL - Partner    TRUE MOTORBIKE                     1               4.935
## 4 WALL - Partner    TRUE MOTORBIKE                     4               7.478
## 5 WALL - Partner    TRUE   BICYCLE                     2               4.266
## 6 WALL - Partner    TRUE       CAR                     1               7.700
##   activation_time_local last_courier_assigned_time courier_started_order_local
## 1  2019-07-13T21:17:06Z       2019-07-13T21:25:56Z        2019-07-13T21:26:49Z
## 2  2019-07-10T21:40:40Z       2019-07-10T22:11:05Z        2019-07-10T22:11:37Z
## 3  2019-07-09T17:33:49Z       2019-07-09T17:34:25Z        2019-07-09T17:34:45Z
## 4  2019-07-13T14:16:25Z       2019-07-13T14:52:19Z        2019-07-13T14:52:51Z
## 5  2019-07-10T20:35:40Z       2019-07-10T21:18:15Z        2019-07-10T21:24:35Z
## 6  2019-07-10T23:02:15Z       2019-07-10T23:33:28Z        2019-07-10T23:33:44Z
##   courier_enters_pickup_time_local    pickup_time_local
## 1             2019-07-13T21:35:32Z 2019-07-13T21:45:32Z
## 2             2019-07-10T21:40:42Z 2019-07-10T22:30:10Z
## 3             2019-07-09T17:35:56Z 2019-07-09T17:41:48Z
## 4                             <NA>                 <NA>
## 5             2019-07-10T20:35:54Z 2019-07-10T21:53:25Z
## 6             2019-07-10T23:38:08Z 2019-07-10T23:59:49Z
##   courier_enters_delivery_point_time_local     termination_time Count
## 1                    2019-07-13T21:59:36Z 2019-07-13T22:02:59Z    NA
## 2                    2019-07-10T22:37:52Z 2019-07-10T22:41:50Z    NA
## 3                    2019-07-09T17:49:10Z 2019-07-09T17:54:08Z    NA
## 4                                   <NA> 2019-07-13T15:07:23Z    NA
## 5                    2019-07-10T21:59:11Z 2019-07-10T22:01:56Z    NA
## 6                    2019-07-11T00:05:20Z 2019-07-11T00:27:19Z    NA
##   Total.Time ...20 ...21 ...22 ...23 ...24 ...25
## 1         45    NA    NA    NA    NA    NA    NA
## 2          1    NA    NA    NA    NA    NA    NA
## 3         20    NA    NA    NA    NA    NA    NA
## 4         50    NA    NA    NA    NA    NA    NA
## 5         26    NA    NA    NA    NA    NA    NA
## 6         25    NA    NA    NA    NA    NA    NA
```

Depuración de Dataset:

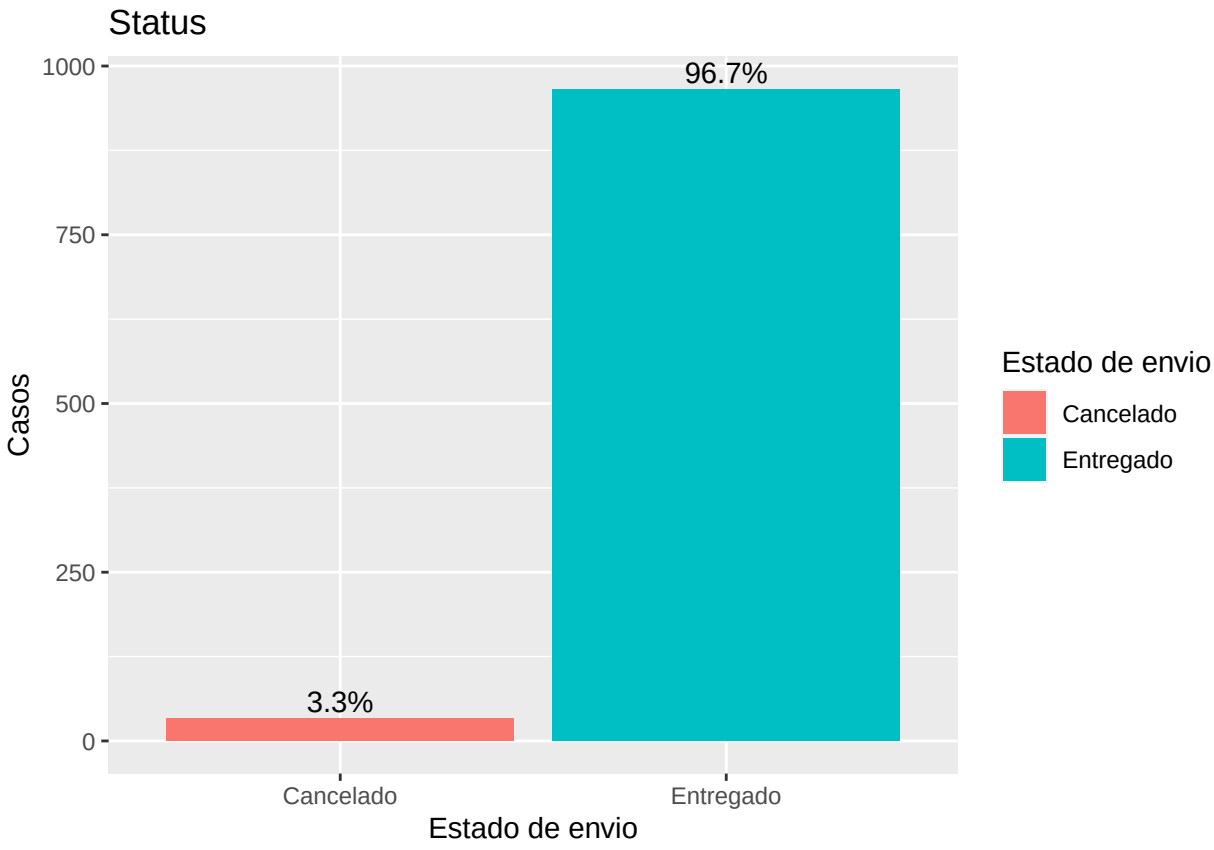
Antes de iniciar abordando cada apartado del ejercicio realizamos una depuración del dataset donde encontramos lo siguiente:

- Eliminamos siete columnas que contenian solo nulos "Count","...20","...21","...22","...23","...24","...25".
- Eliminamos una fila que continua todos los campos nulos.
- Notamos una relación entre la vertical Quiero y el store_id, todos los envios de tal vertical poseen store_id nulo.
- Notamos que los envios cancelados no completan el ciclo de reporte de status, cosa que tiene sentido.
- Notamos que existia una columna erronea Total.Time y la recalculamos. $Total.Time = terminationtime - activationtimelocal$. Esta columna tenia mal calculado el tiempo.

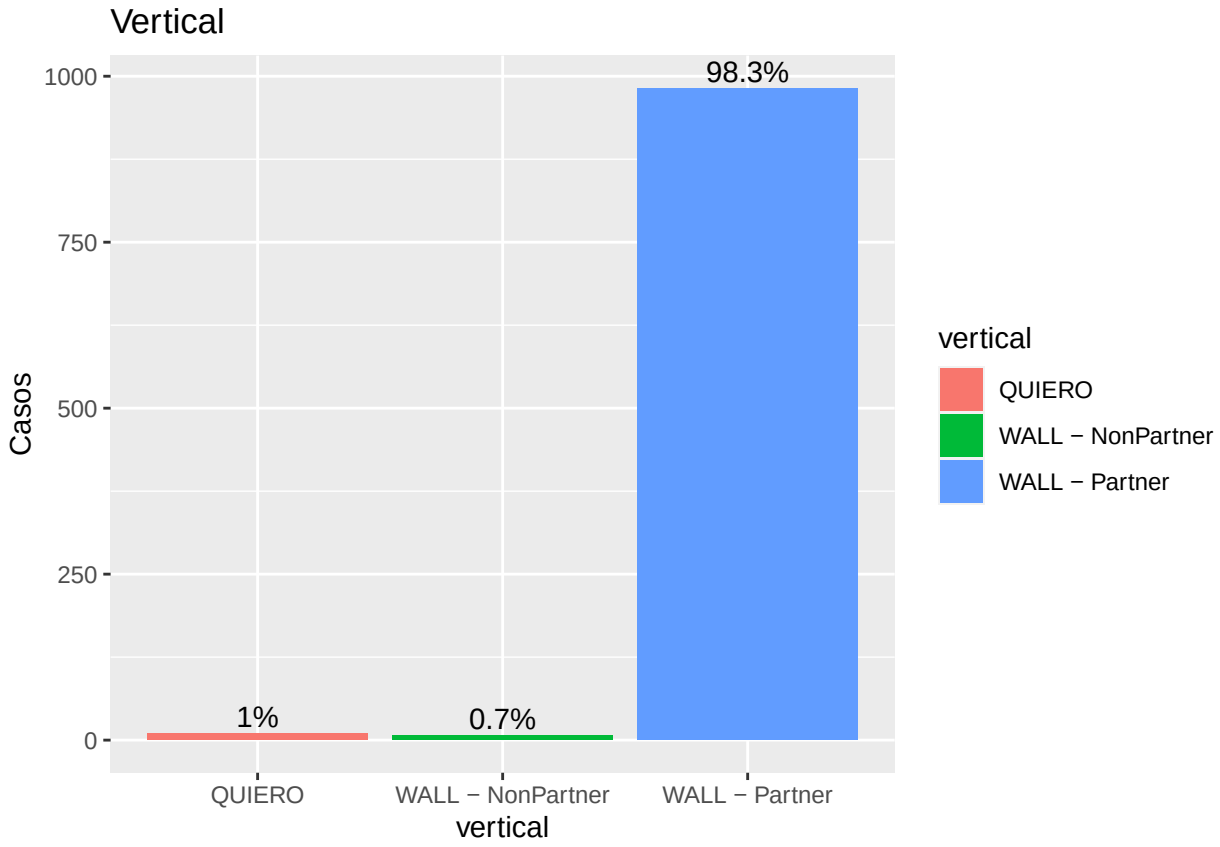
Graficos de Exploratorio

A continuación ilustramos en algunos gráficos comportamientos que encontramos en el exploratorio del dataset.

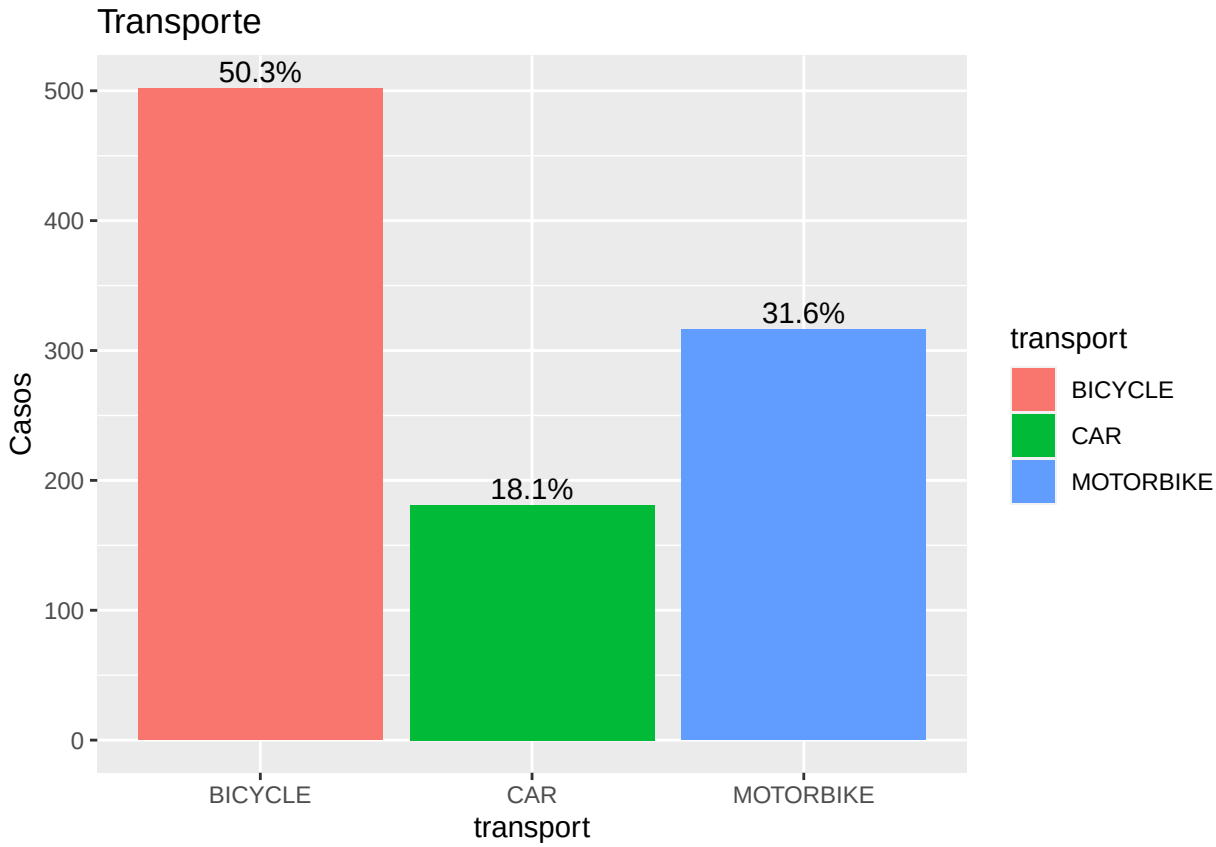
- ENVIOS COMPLETADOS



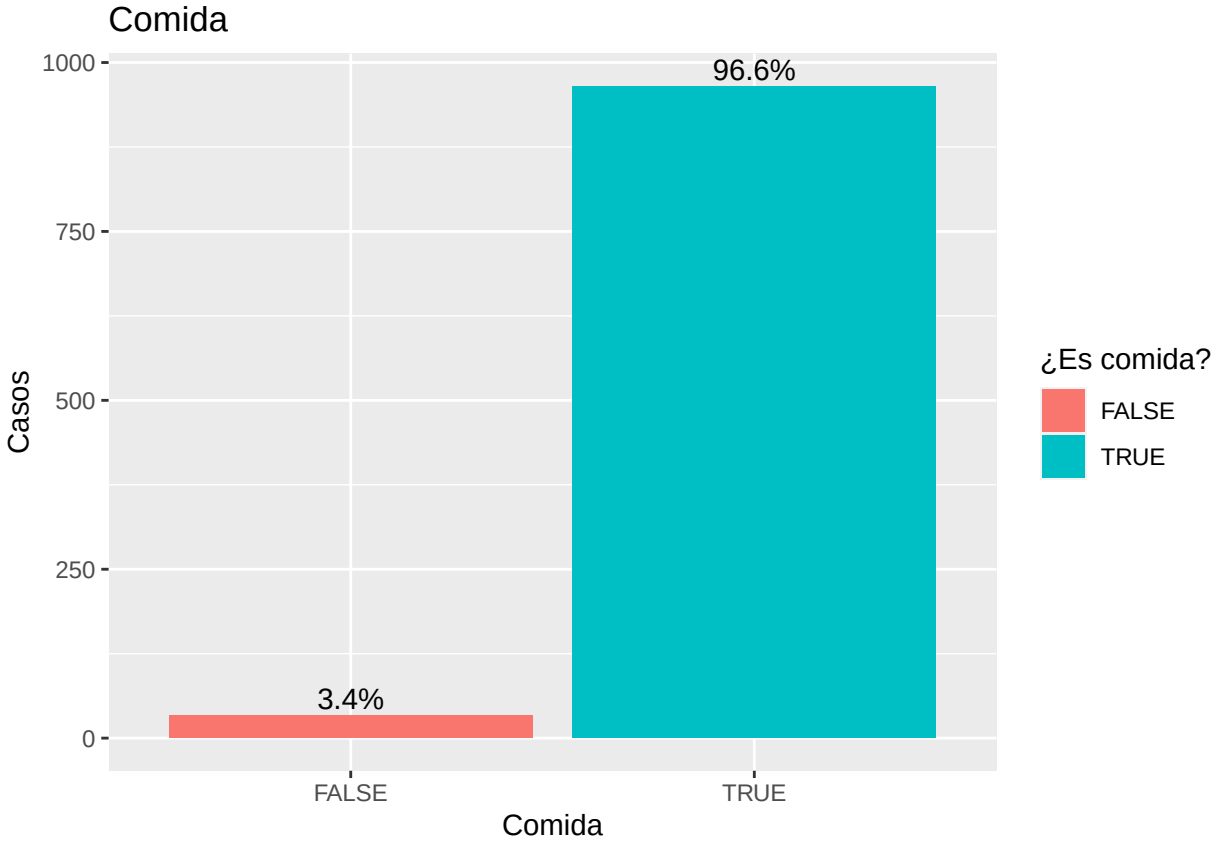
- ENVIOS POR VERTICAL



- ENVIOS POR TIPO DE TRANSPORTE



- ENVIOS SEGUN TIPO



Summary de variables de interés

- RESUMEN DE VARIABLES REFERENTES A DISTANCIA

total_real_distance	Total.Time
Min. : 0.708	Min. : 0.00
1st Qu.: 3.349	1st Qu.: 25.00
Median : 4.513	Median : 35.00
Mean : 5.099	Mean : 38.03
3rd Qu.: 6.218	3rd Qu.: 47.00
Max. :16.527	Max. :123.00

Solución propuesta 2.A:

Supuestos:

- Asumimos como probabilidad de reasignación que el envío posea 2 o más asignaciones.
- Asumimos que la distancia esta dada en kilometros (km)

Metricas distancia y tiempo

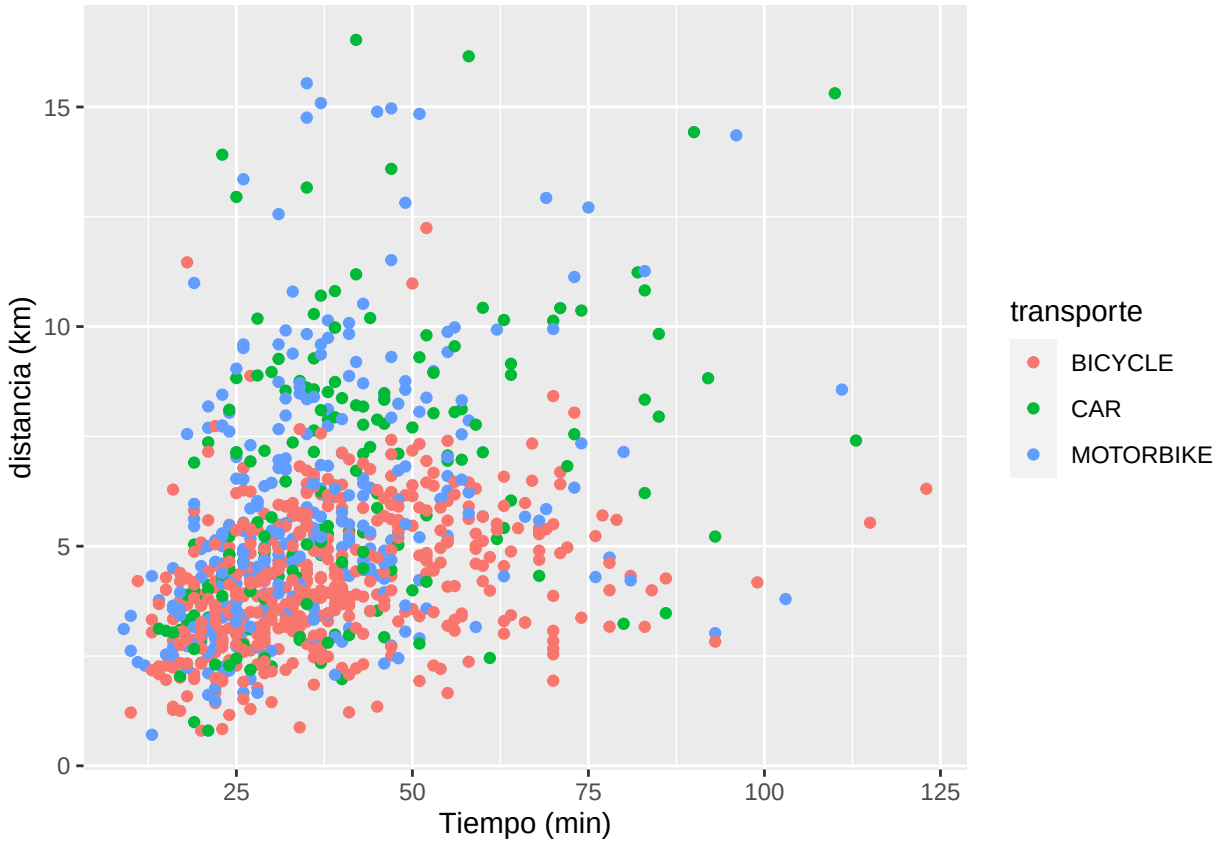
En vista de que ambas variables se pueden vincular para obtener información de que tan rápido se entregan los envíos generamos la variable velocidad que será:

$velocidad = \frac{distancia(km)}{tiempo(hora)}$

Además consideramos que una variable fundamental para determinar la velocidad de un envío es el transporte por el que se envía, es por ello que para visibilizar las metricas de distancia y tiempo de entrega lo segmentamos por tipo de transporte.

transporte	tiempo medio	tiempo desviacion std	distancia media	distancia desviacion std	velocidad (kmh)
BICYCLE	37.67617	16.83361	4.203872	1.562691	6.694744
CAR	41.32000	19.54630	6.344143	3.089012	9.212211
MOTORBIKE	36.06667	16.14821	5.786227	2.816905	9.625885

En el siguiente gráfico de dispersión vemos los distintos envios entregados por tipo de transporte y distancia:



Probabilidad de reasignación

Definiendo x como la cantidad de asignaciones y la reasignación como una situacion en la que $x > 1$ entonces:

$P(x > 1) = \frac{asignaciones(x>1)}{totalenvios}$

Hallemos tal valor:

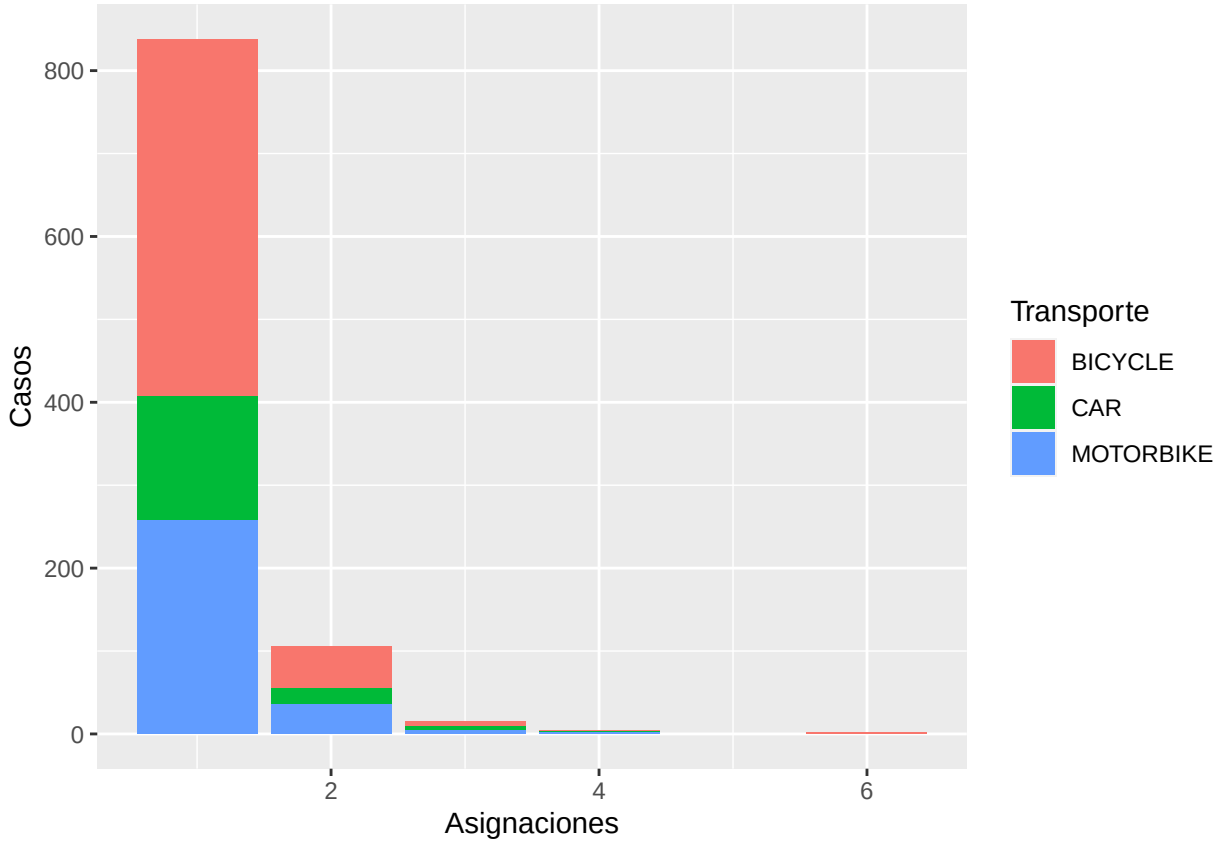
```
probabilidad_reasignacion1 = length(df[df$number_of_assignments>1,"number_of_assignments"])/nrow(df)

probabilidad_reasignacion2 = length(df[(df$number_of_assignments>1) & (df$final_status=='DeliveredStatus

print(paste('La probabilidad de reasignacion tomando en cuenta todos los envios es:',probabilidad_reasig
```

```
## [1] "La probabilidad de reasignacion tomando en cuenta todos los envios es: 0.137137137137137"
```

```
print(paste('La probabilidad de reasignacion tomando en cuenta solo envios entregados:',probabilidad_rea
```

```
## [1] "La probabilidad de reasignacion tomando en cuenta solo envios entregados: 0.132505175983437"
```

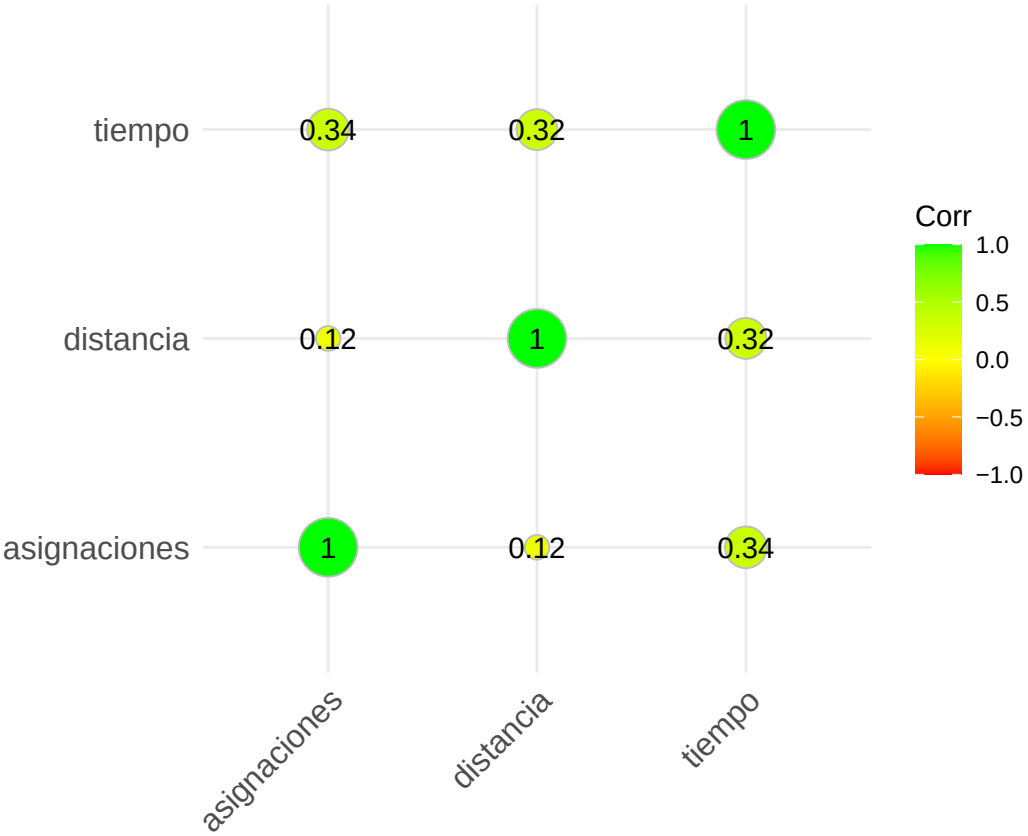
Además veamos en el siguiente gráfico como es el comportamiento de las asignaciones para envios entregados, de acuerdo al tipo de transporte:



Solución propuesta 2.B:

En el apartado anterior pudimos notar que aparentemente la cantidad de asignaciones no se relaciona con el tipo de transporte, ahora veamos la relación que tiene con las columnas numericas.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



Tiene una leve relación positiva con el tiempo.

Solución propuesta 2.C:

Supuestos:

- Para generar una variable que contenga información acerca del tiempo de espera en el local asumimos que el mismo viene dado por: $Tiempodeespera = pickuptimelocal - courierenterspickuptimelocal$

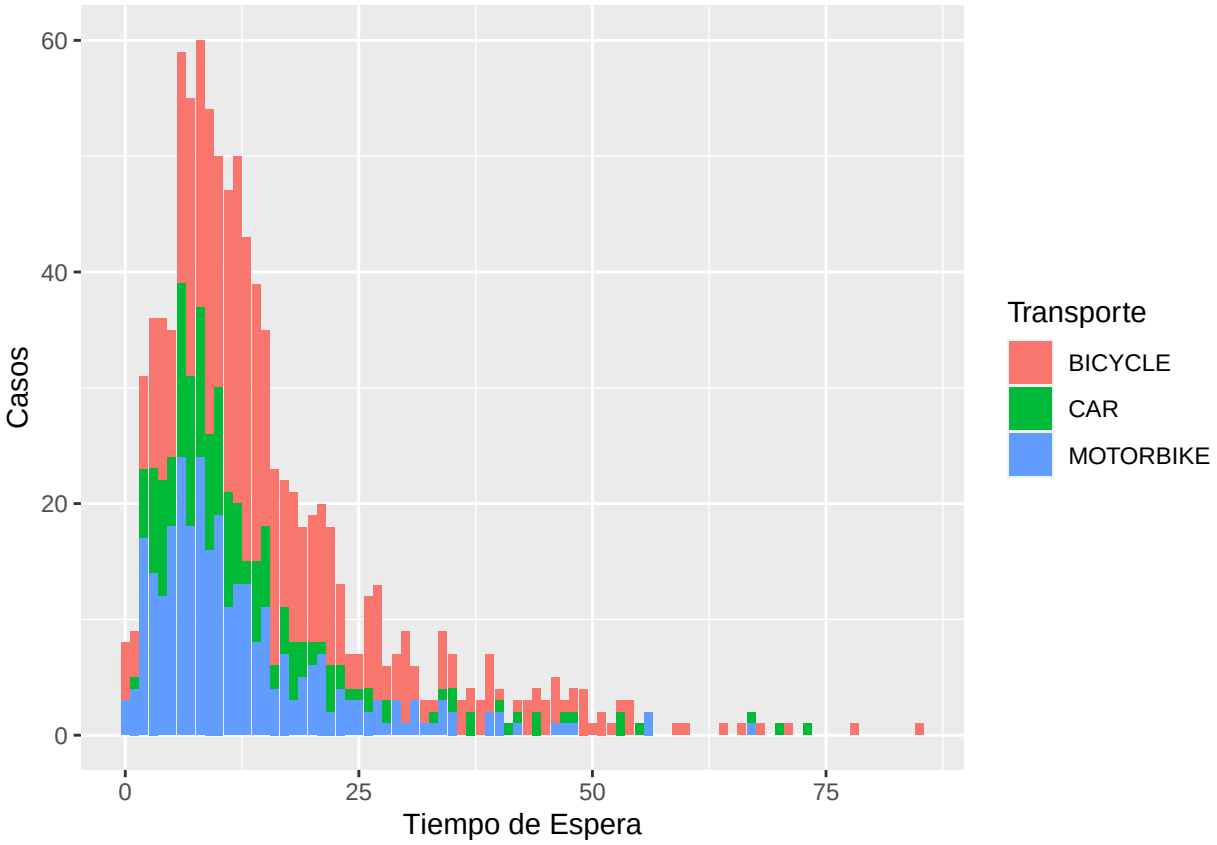
Generacion de variable

Generamos la variable tiempo de espera para evaluar su comportamiento:

- Tabla tiempo de espera

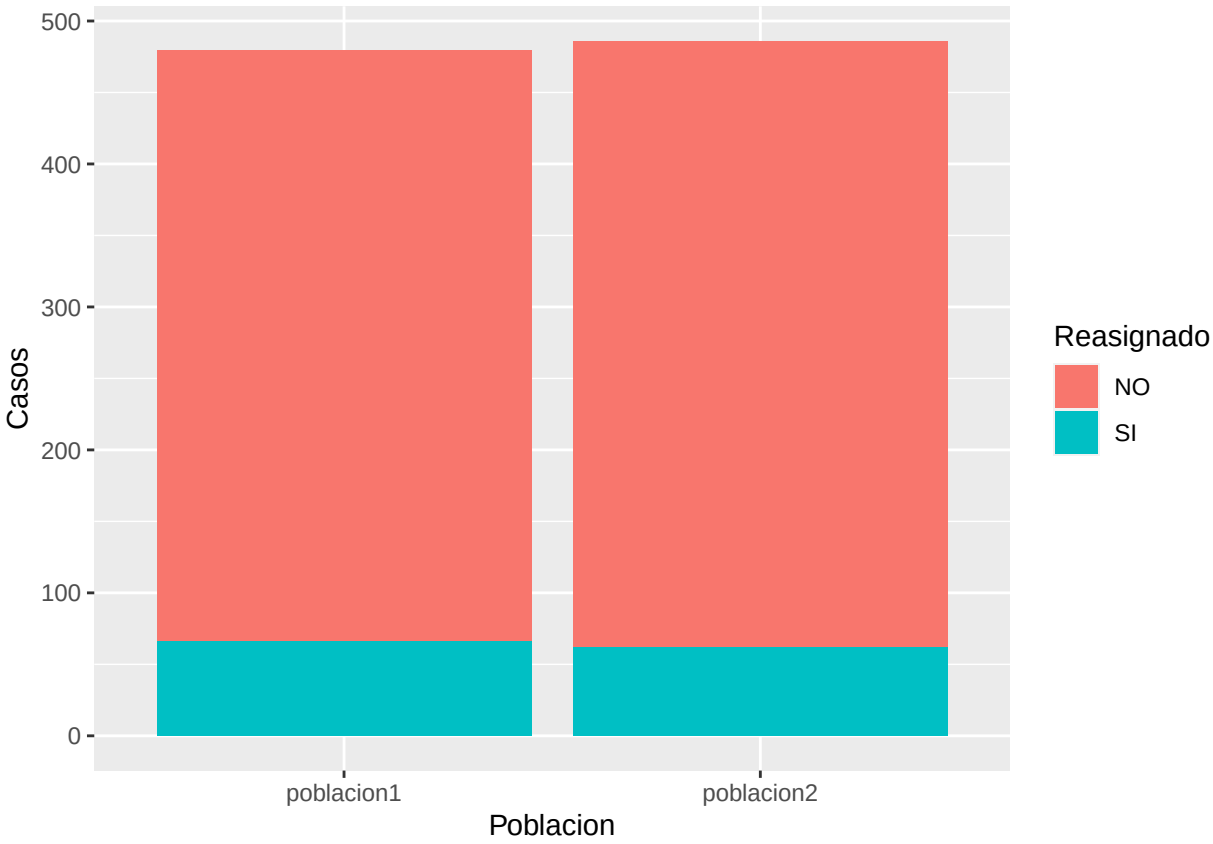
media	desviacion std	mediana
15.19151	12.62072	12

- Distribucion tiempo de espera según transporte



Segmentación y comparación de población

A continuación comparamos la relación que existe entre la cantidad de asignaciones y el tiempo de espera, para eso generamos dos poblaciones la poblacion1 esta compuesta por casos cuyo tiempo de espera es menor a la media y la poblacion2 por los demas casos.

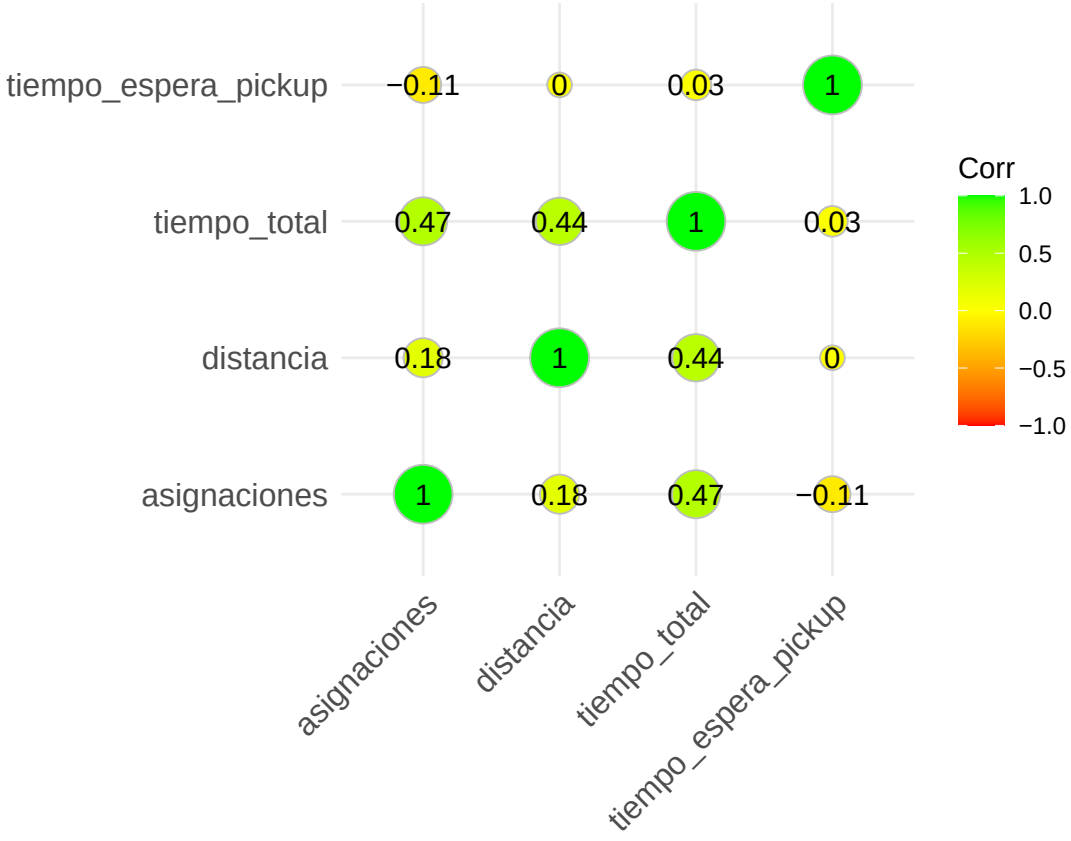


Aparentemente no existe relación lineal entre la reasignación y el tiempo de espera.

Ahora verificamos la relación que puede tener el tiempo de espera con otras variables númericas, para cada población previamente segmentada:

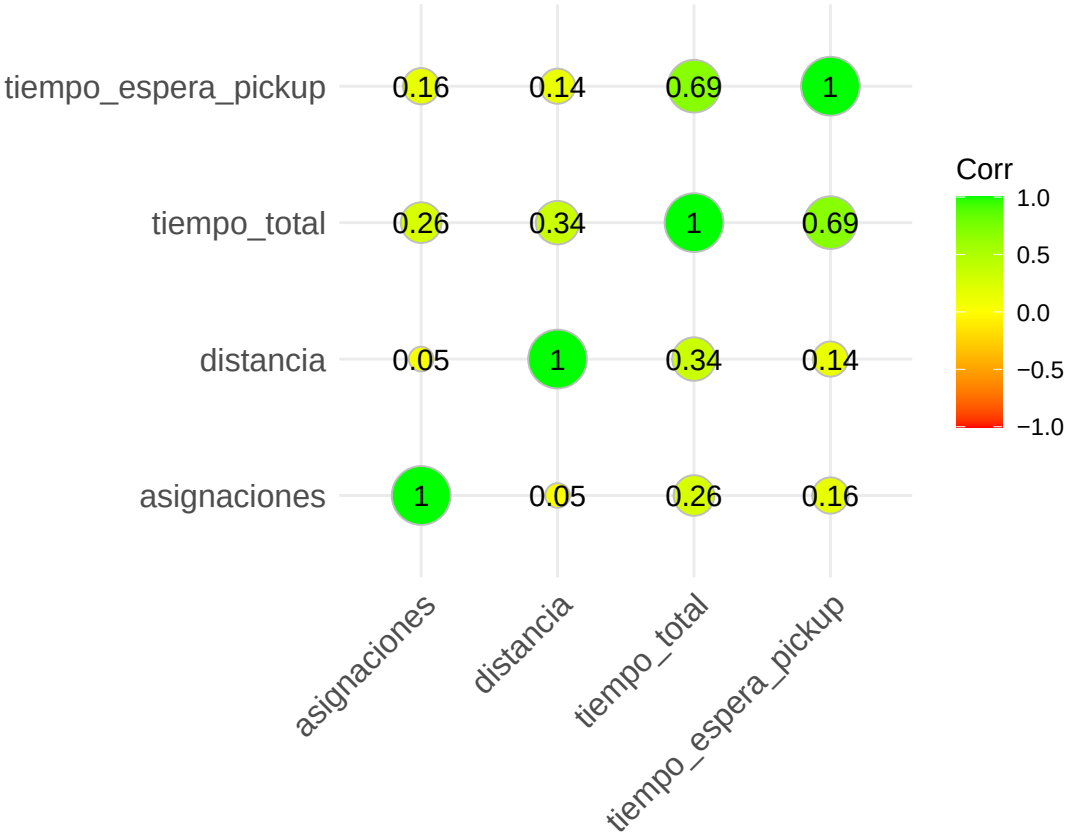
- Población1 (tiempo_espera < mediana):

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



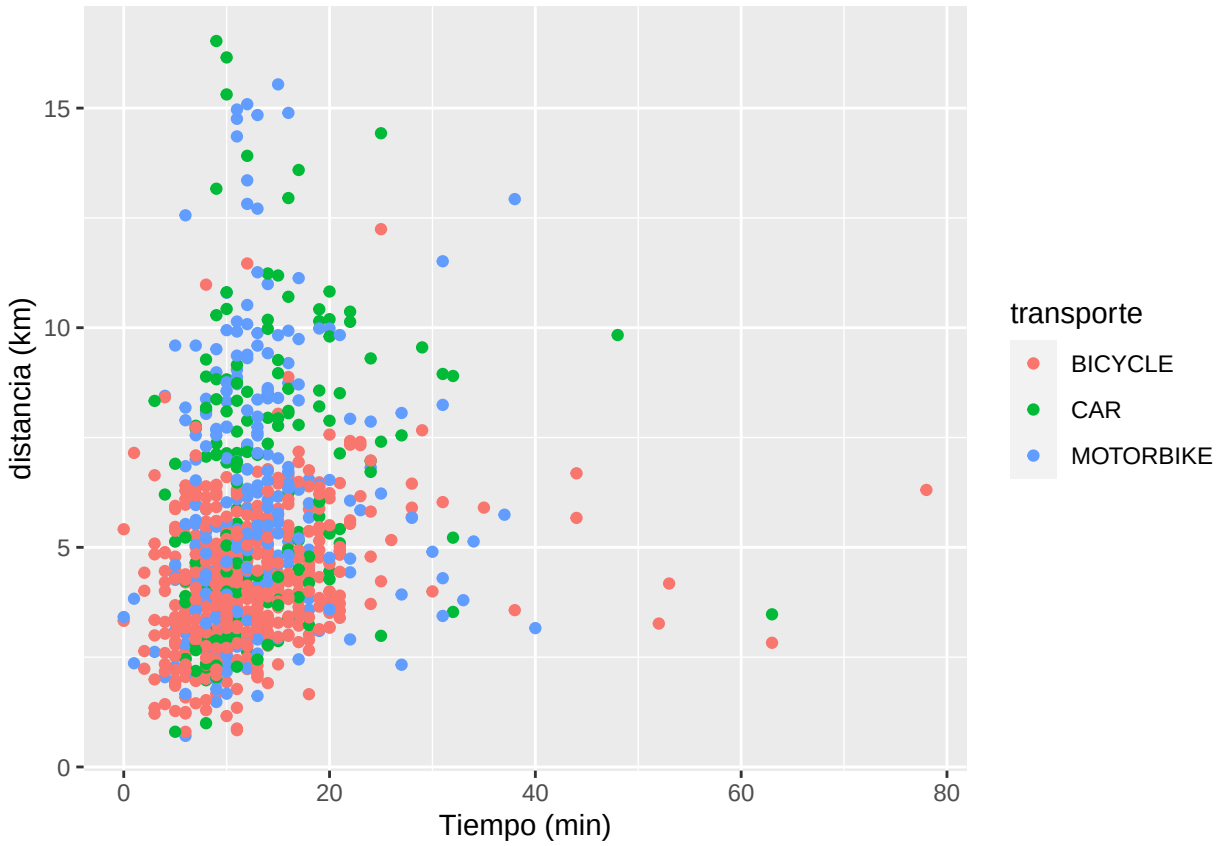
- Población2 (tiempo_espera >= mediana):

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



Podemos destacar como dato relevante de la comparación que la alta volatilidad que posee el tiempo de espera genera un impacto muy fuerte en el tiempo total de entrega, para la población2 se observa que puede llegar a representar incluso el mayor impacto en el tiempo total.

A continuación presetamos un gráfico donde se puede apreciar mejor generando una nueva variable del tiempo transcurrido desde el retiro en el local hasta la entrega, comparar el siguiente gráfico de dispersión con el expuesto en la solucion 2.A



Se puede notar que comparando con el grafico anterior este agrupa muchos mas puntos de tipo Car y Motorbike a la izquierda, lo que representa mucho mejor la velocidad de entrega de este tipo de vehiculos.