

Modelo de Predicción de Demanda y Pricing Dinámico

Oscar Daniel Camarena Gómez

Contents

1	Objetivo	1
2	Contexto de Negocio	1
3	Descripción del Problema	3
3.1	Gestión de las propiedades	3
3.2	Situación Actual	6
3.3	Propuesta de mejora al proceso	6
4	Metodología Propuesta	7
4.1	Plan de implementación	7
4.2	Fuentes de datos	7
4.3	Análisis Exploratorio de Datos	8
4.4	Propuesta de Modelo Predictivo	25
4.5	Interpretación del modelo	28
5	Anexo A: Implementación del modelo pronóstico de demanda	32
6	Anexo B: Implementación del Modelo de Pricing	36
7	Referencias	40

1 Objetivo

El objetivo del presente trabajo es generar un producto de datos que permita a la cadena de hoteles CityExpress pronosticar la demanda de ocupación de sus propiedades. Esto permitirá que el equipo comercial que opera las propiedades manipulen los precios de cada una de las tarifas disponibles de tal forma que puedan maximizar el ingreso de cada una de ellas.

Para poder generar el modelo deseado CityExpress puso a nuestra disposición data sets que contienen información de las reservaciones generadas en cada una de las propiedades desde el año 2013, información de las reservaciones pendientes por llegar para el año 2018 - 2019, información de las principales variables hoteleras para la competencia e información de los precios de su competencia publicados en las agencias de viajes en línea (Expedia, Booking.com, etc).

Antes de hacer un análisis exhaustivo a los datos de la cadena, se hizo un análisis de negocio lo cual ayudó a que pudieramos dar un contexto social a la información que nos proporcionaron y a su vez ayudó a la interpretación de los resultados arrojados por los distintos análisis ejecutados.

2 Contexto de Negocio

CityExpress es una cadena de hoteles 100% mexicana enfocada al turismo de negocios. Actualmente cuenta con 140 hoteles en México y Latinoamérica (Colombia, Costa Rica y Chile). CityExpress cuenta con hoteles pensados para el viajero de negocios, sus instalaciones son prácticas, la tarifa es baja y ofrece servicios limitados pero valiosos durante un viaje de negocios, por ejemplo:

- Internet gratuito en las habitaciones
- Desayuno de cortesía
- Transporte gratuito 10 km a la redonda
- Servicio estandarizado con bajo costo

CityExpress cuenta con 4 marcas, cada una de ellas enfocada a distintos segmentos de mercado:

- **CityExpress:** Marca emblema, enfocada al viajero de negocios. Los hoteles están ubicados cerca de los centros de negocios de las distintas ciudades en la República Mexicana.
- **CityExpress Plus:** Hoteles ubicados en las principales ciudades de la República Mexicana (CDMX, Guadalajara, Monterrey) y en Latinoamérica (Colombia). Ofrecen diseños vanguardistas y mayor espacio en las habitaciones.
- **CityExpress Suites:** Marca enfocada en los viajeros de larga estancia. Sus instalaciones cuentan con cocineta, estancia y habitaciones de mayor tamaño.
- **CityExpress Junior:** Hoteles de menor precio enfocados al turismo con presupuesto limitado ofreciendo un producto de calidad y con servicio estandarizado.

Ubicación

Típicamente un hotel de CityExpress está ubicado cerca de zonas industriales o zonas con fuerte actividad económica, es decir, en el caso de las grandes ciudades encontraremos hoteles CityExpress cerca de una zona donde hay alta densidad de locales comerciales, oficinas corporativas o centros de negocios. En el caso de ciudades mas pequeñas o con actividades económicas específicas encontraremos hoteles cerca de los centros económicos de cada una de las ciudades, este hecho permite que la cadena agrupe a sus hoteles en distintos **corredores** dependiendo de la actividad de las zonas económicas donde se encuentran. A continuación se presentan los distintos corredores:

- Energético
- Exportacion-Agricultura
- Franquicia
- Internacional
- Manufactura
- Maquila
- Minería
- Servicios

Hoteles

Las propiedades de Hoteles CityExpress siguen un estándar desde su construcción hasta en su operación diaria. En cuanto al estándar de construcción las propiedades generalmente cuentan con 6 plantas, 120 habitaciones, un lobby para la recepción de huéspedes y áreas comunes; por ejemplo, área designada para fumar, desayunador, salas de juntas, salones de eventos y alberca (en algunas propiedades). En cuanto a las habitaciones el tamaño puede variar por marca, sin embargo ofrecen los mismos servicios independientemente de ella:

- Internet Gratuito
- Área de Trabajo
- Lavandería
- Aire Acondicionado
- Pantalla de televisión
- Desayuno de cortesía

Crecimiento

Hoteles CityExpress fue fundada en 2002 con su primer hotel construido en Saltillo. A partir de ese entonces se ejecutó un ambicioso plan de crecimiento en el cuál se impuso la meta de abrir un nuevo hotel cada 6.8 semanas. Esto llevo a que la cadena de hoteles contara con más de 140 propiedades en un lapso de 16 años. Hoy en día Hoteles CityExpress cuenta con propiedades en 34 de los 36 estados de la República Mexicana y 5 mas en Latinoamérica: 1 en Costa Rica, 1 en Chile, 3 en Colombia. El plan de crecimiento sigue siendo ambicioso y la cadena espera contar con 180 propiedades para el 2020.

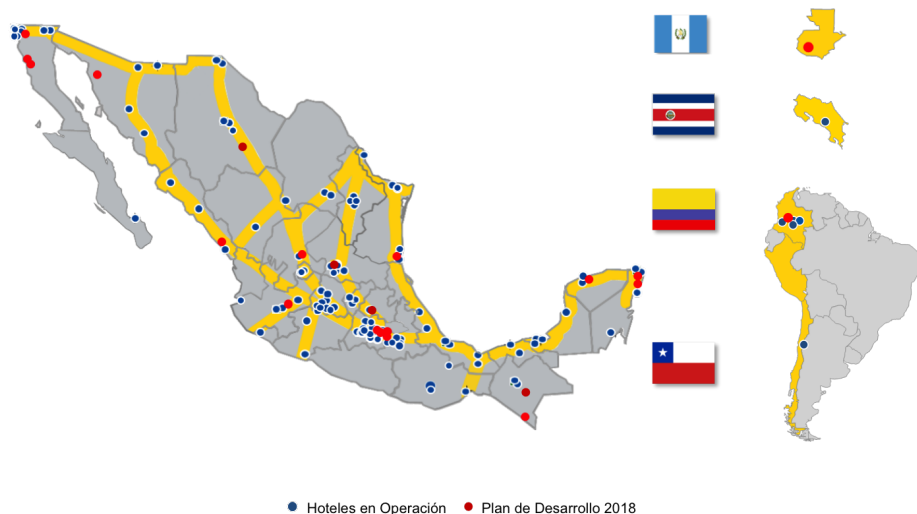


Figure 1: Plan de crecimiento

3 Descripción del Problema

3.1 Gestión de las propiedades

Hoteles CityExpress cuenta con un modelo de gestión comercial de sus propiedades ejecutado desde sus oficinas centrales ubicadas en la Ciudad de México. Este modelo de gestión se enfoca en optimizar la venta de los cuartos de las distintas propiedades, buscando siempre maximizar el ingreso obtenido por estas ventas. Para poder entender mejor el modelo de gestión comercial debemos estudiar antes los canales distribución que utiliza para la venta de sus productos así como sus distintos segmentos de mercado. Posteriormente presentaremos las principales variables utilizadas para calificar el desempeño de las propiedades así como las acciones que puede tomar cada uno de los gerentes de las propiedades para poder cambiar el desempeño del hotel.

Canales de Venta

CityExpress cuenta con un modelo de venta estandarizado para todas sus propiedades. Este modelo de venta contempla distintos canales de distribución que ayudan al grupo a poder incrementar la oferta de su producto llegando a distintos segmentos de mercado. Es importante mencionar que cada uno de los canales de venta tienen un costo asociado, es decir, del ingreso que se perciba por cada cuarto vendido se debe tomar en cuenta el costo de cada canal para poder calcular la utilidad bruta de la venta, por eso es importante conocer cómo opera cada uno de ellos. Dentro de CityExpress existen dos tipos de canales de venta: **Canales de venta tradicionales** y **Canales de venta electrónicos**. Los canales de venta tradicionales, son aquellos en los cuales se requiere la inter-

vención de un agente de ventas, los electrónicos permiten que el cliente final realice su reservación sin requerir la intervención del agente:

Canales de venta tradicionales

- Centro de Contacto (Call Center)
- Hotel

Canales de venta electrónicos

- App (IOS y Android)
- Portal Cliente Frecuente (City Premios)
- Motor de reservaciones corporativas (CityAccess)
- Motor de reservaciones (www.cityexpress.com.mx)
- Agencias de viajes en línea (OTAs)

Segmentos de Mercado

Los segmentos de mercado son un punto medular para poder entender el modelo de gestión en CityExpress ya que estos permiten al área comercial poder establecer una estrategia de venta en cada uno de los canales fijando precios en las distintas tarifas para poder maximizar la ocupación del hotel y el ingreso por cuartos vendidos.

Típicamente un hotel cuenta con un conjunto de tarifas las cuales son ofrecidas a diferentes segmentos de mercado en diferentes canales. Por ejemplo, un huésped que viaja por placer y reserva por la página web de CityExpress muy probablemente recibirá una tarifa distinta al huésped que viaja por negocios y reservó por el centro de contacto.

A continuación se presentan los distintos segmentos de mercado manejados por CityExpress:

Negocios

- Directos
- Convenios
- Viajero Frecuente
- Corporativo
- Consorcios
- Promociones
- Otros

Placer

- Directos
- Fin de Semana
- Otros
- Promociones

Mayoreo

- Agencias Minoristas
- OTAs
- Lineas Aéreas / Tripulaciones
- Agencias Mayoristas
- Otros

Grupos

- Grupos, Ferias, Congresos y Convenciones
- Sociales
- Juntas
- Deportes

Otros

- Uso Casa
- Cortesías
- Intercambio
- Empleados
- Industria Turística
- Otros

Tarifas

Como se mencionó anteriormente, un hotel tiene un catálogo de 36 tarifas que puede ofrecer a los huéspedes. Cada tarifa va dirigida a un segmento de mercado y un canal en específico. Hay tarifas que pueden cambiar de precio hasta 3 veces al día y hay otras que al ser tarifas convenidas con otras empresas, permanecen estáticas la mayor parte del año. Es responsabilidad de cada uno de los hoteles el revisar los distintos reportes expuestos en la organización en donde pueden encontrar: sus niveles de ocupación, ocupación de la competencia, tarifas promedio propias, tarifas promedio de la competencia, penetración de ocupación: $\frac{\%ocupacion\ propia}{\%ocupacion\ de\ la\ competencia}$ y penetración de tarifa: $\frac{tarifa\ promedio\ propia}{tarifa\ promedio\ de\ la\ competencia}$ para decidir si deben hacer un incremento o decremento en los precios ofertados.

Variables estudiadas

Dentro de CityExpress las decisiones de cambios de precios en las tarifas se toma analizando distintas variables. La mayoría de estas variables se alimentan de información propia del grupo hotelero aunque hay algunas que se obtienen de información de la competencia. A continuación se detallará cada una de las variables utilizadas durante la toma de estas decisiones:

- %Ocupacion

- Tarifa Promedio
- Tarifa Efectiva
- Penetración de Ocupación
- Penetración de Tarifa
- Eventos por plaza
- %Ocupación de la competencia
- Tarifa Promedio de la competencia
- Tarifa Efectiva de la competencia

Derivado de las conclusiones del análisis de estas variables, los hoteles y el equipo de la oficina central pueden tomar alguna de las siguientes decisiones para alcanzar el objetivo deseado:

- Subir precio de tarifa
- Bajar precio de tarifa
- Abrir un canal de venta
- Cerrar un canal de venta
- Promover la venta de un segmento específico mediante una promoción
- Inhibir la venta de un segmento en específico cerrando disponibilidad de alguna tarifa

3.2 Situación Actual

Como se mencionó en el apartado anterior, hoy en día CityExpress gestiona su estrategia comercial desde las oficinas centrales delegando tareas clave a cada uno de los gerentes de las distintas propiedades, esto aunado al ambicioso plan de crecimiento de la cadena ha hecho que este modelo de gestión comercial se vuelva un proceso complicado de administrar y ejecutar de manera efectiva ya que se tienen las siguientes dependencias en el proceso:

- Publicación de reportes diarios
- Análisis de información publicada
- Presentación de plan de acción de acuerdo al modelo de gestión
- Aprobación del plan de acción por parte de la oficina central
- Ejecución del plan de acción en los sistemas de control
- Validación de resultados del plan de acción

3.3 Propuesta de mejora al proceso

Este proyecto tiene como objetivo mejorar el proceso de gestión de hoteles en CityExpress mediante la construcción de un sitio en donde se integre toda la información que hoy en día se expone en distintos reportes dentro de la empresa y además cuente con un modelo que permita pronosticar el *% de ocupacion* de cada una de las propiedades y a su vez haga una recomendación del precio para

la tarifa que se ofrece al público en general que ayude a maximizar el ingreso por venta de cuartos dentro de la propiedad. Con esta herramienta se espera que el modelo de gestión comercial se automatice y la ejecución del plan de acción se realice de manera más ágil.

4 Metodología Propuesta

4.1 Plan de implementación

A continuación se presentan los pasos que se siguieron para la implementación de este nuevo sistema de gestión:

- **Entendimiento del negocio:** Antes de construir una herramienta / producto de datos se dedicó un tiempo a conocer los procesos operativos de la cadena hotelera.
- **Entendimiento del sistema de gestión comercial:** Para que el producto final agregue verdadero valor a la empresa se debió comprender el tramo de control de cada puesto operativo destinado a hacer uso de esta herramienta, es decir, se identificaron las acciones que cada usuario puede ejecutar en los distintos sistemas para alcanzar sus objetivos comerciales.
- **Identificación de posibles fuentes de datos:** Una vez comprendidos los procesos operativos y de toma de decisiones, se identificaron los sistemas que proveen la información que ayuda al personal de CityExpress durante la toma de decisiones.
- **Análisis exploratorio de datos:** Una vez identificadas las fuentes de datos, se ejecutó un análisis exploratorio de datos con el fin de poder conocer los datos con que se trabajaron para posteriormente fijar un alcance para el producto final.
- **Modelo de predicción de ocupación y sugerencia de precios:** Una vez perfilados los datos y con un EDA concluido se realizó un esfuerzo para construir un modelo que fuera capaz de pronosticar el % de ocupación y que sugiriera el *precio público* óptimo de cada una de las propiedades.
- **Construcción de prototipo de producto de datos:** Se definió la arquitectura final del producto de datos y se construyó un prototipo totalmente funcional que apoya la toma de decisiones de los participantes del proceso de gestión comercial.

4.2 Fuentes de datos

Una vez hecho el análisis de negocio y conociendo la operación de CityExpress,

detectamos que las fuentes principales para la construcción del producto de datos son las siguientes:

- **Central de Reservaciones:** Módulo encargado de realizar las reservaciones de cuartos en las distintas propiedades. De este sistema obtuvimos el comportamiento histórico de cada una de las propiedades, es decir, cuantas habitaciones fueron reservadas, en qué momento y con qué tarifa.
- **Property Management System (PMS):** Módulo encargado de gestionar las propiedades. Aquí se pueden consultar las habitaciones ocupadas, disponibles y fuera de servicio, así como las llegadas y salidas del día. De este sistema obtuvimos las reservaciones en libros (bookings) o llegadas pendientes, esta información es crucial para poder conocer en la demanda real de cada una de las propiedades en el tiempo.
- **Market Share:** Sistema encargado de recopilar la información de la competencia. Las variables recopiladas son Tarifa Promedio, Habitaciones Disponibles, Habitaciones ocupadas, entre otras.
- **Tarifas de la competencia:** Consulta de las tarifas disponibles en las agencias de viajes en línea para la competencia.

4.3 Análisis Exploratorio de Datos

A continuación se mostrará a detalle análisis exploratorio realizado al dataset inicial. Este data set comprendía información de 5 hoteles y el objetivo de este era brindar un apoyo a la comprensión de los datos para poder establecer una metodología de trabajo al momento de crear un modelo de predicción de demanda para cada una de las propiedades. Para fines de este trabajo detallaremos el "EDA" utilizando datos de una sola propiedad: "CEXXX".

Descripción del DataSet

Durante la construcción del modelo se hizo uso de tres datasets principalmente. El primero de ellos llamado "histórico_ocupación" fue obtenido del sistema de gestión de propiedades (**PMS**) y contiene la información del desempeño histórico de las propiedades. A continuación se presenta un resumen del data set en cuestión.

```
##      rsrv_code      date_create
##  Min.      :6265739  Min.      :2016-07-20 18:31:59
##  1st Qu.:7499788    1st Qu.:2017-03-27 10:10:09
##  Median :7999769    Median :2017-06-30 12:12:57
##  Mean   :8007253    Mean   :2017-06-27 13:38:02
##  3rd Qu.:8505582    3rd Qu.:2017-09-27 13:22:36
##  Max.   :9053354    Max.   :2018-01-01 03:08:21
##      date_in
##  Min.      :2017-01-01 00:00:00
```

```

## 1st Qu.:2017-04-04 00:00:00
## Median :2017-07-06 00:00:00
## Mean   :2017-07-06 02:18:17
## 3rd Qu.:2017-10-08 00:00:00
## Max.    :2017-12-31 00:00:00
##      date_out                      nights
## Min.    :2017-01-02 00:00:00   Min.    :1
## 1st Qu.:2017-04-05 00:00:00   1st Qu.:1
## Median :2017-07-07 00:00:00   Median :1
## Mean    :2017-07-07 02:18:17   Mean    :1
## 3rd Qu.:2017-10-09 00:00:00   3rd Qu.:1
## Max.    :2018-01-01 00:00:00   Max.    :1
##      prop_code      mkt_sgm      Dia_Sem
## Length:44391      Length:44391      Length:44391
## Class :character   Class :character   Class :character
## Mode  :character   Mode  :character   Mode  :character
##
##
##      rate_code      bucket      Ingresos
## Length:44391      Length:44391      Min.    : 0
## Class :character   Class :character   1st Qu.:1200
## Mode  :character   Mode  :character   Median :1462
##                                     Mean    :1449
##                                     3rd Qu.:1658
##                                     Max.    :3900
##      rsrv_src      rsrv_type      room_type
## Length:44391      Length:44391      Length:44391
## Class :character   Class :character   Class :character
## Mode  :character   Mode  :character   Mode  :character
##
##
##      pax
## Min.    :0.000
## 1st Qu.:1.000
## Median :1.000
## Mean    :1.271
## 3rd Qu.:1.000
## Max.    :5.000

```

A continuación se presenta el diccionario de datos para el data set presentado:

- rsrv_code: Código de confirmación de reserva
- date_create: Fecha de creación de la reserva

- date_in: Fecha de entrada al hotel
- date_out: Fecha de salida del hotel
- nights: Número de noches en el hotel
- prop_code: Código de Hotel
- mkt_sgm: Segmento de Mercado
- Dia_Sem: Día de la semana para la fecha de entrada al hotel
- rate_code: Código de tarifa
- bucket: Nivel de tarifa
- Ingresos: Ingresos obtenidos por la reservación
- rsrv_src: Canal de la reservación
- rsrv_type: Tipo de reservación
- room_type: Tipo de habitación
- PAX: Personas amparadas por la reservación

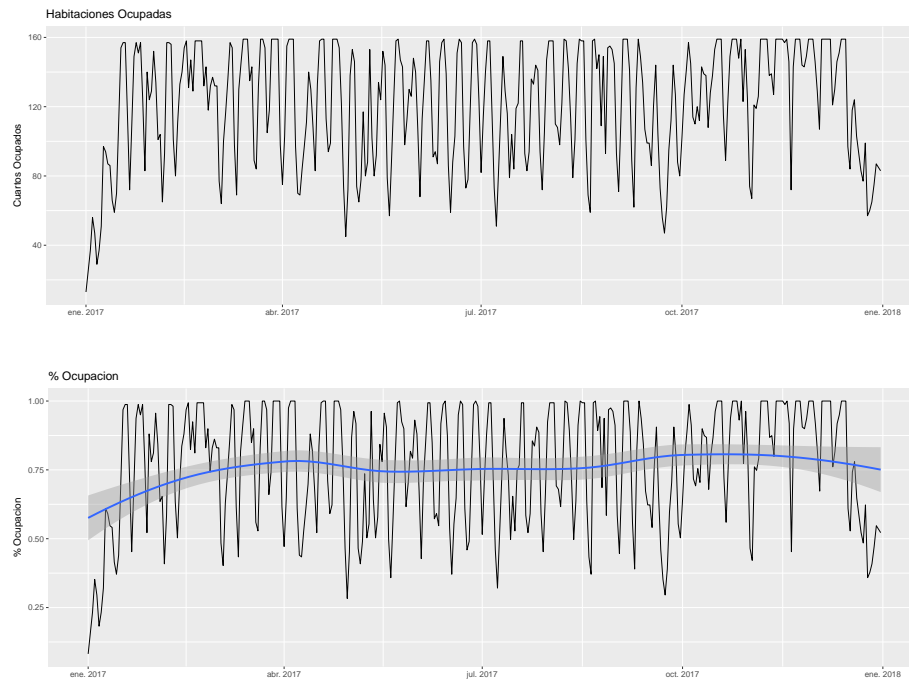
Revisando el resumen de este data set es importante resaltar que los datos fueron trabajados previamente para presentarse de manera desagregada, es decir, si una reserva ampara una estancia de 5 noches tendremos un registro por cada noche de la estancia, por ejemplo:

rsrv_code	date_create	date_in	date_out	nights	room_type
6265739	2016-07-20 18:31:59	2017-04-07	2017-04-08	1	NSK
6265739	2016-07-20 18:31:59	2017-04-08	2017-04-09	1	NSK
6265739	2016-07-20 18:31:59	2017-04-09	2017-04-10	1	NSK
6265739	2016-07-20 18:31:59	2017-04-10	2017-04-11	1	NSK
6265739	2016-07-20 18:31:59	2017-04-11	2017-04-12	1	NSK
6265739	2016-07-20 18:31:59	2017-04-12	2017-04-13	1	NSK

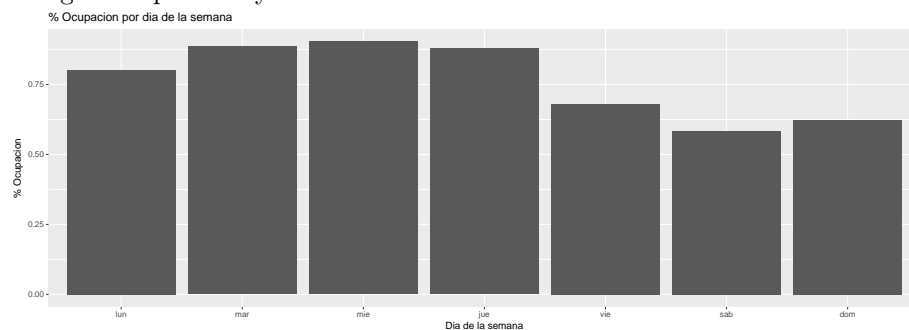
Como podemos observar el numero de reservación es el mismo en los 6 registros, lo mismo ocurre con la fecha de creación, las noches, y el tipo de habitación, sin embargo, la fecha de entrada y la fecha de salida va cambiando en cada registro. En algunos casos es más fácil trabajar con los registros desagregados, sin embargo hay casos en dónde necesitaremos agrupar la información.

Ocupación

De las primeras preguntas que se vienen a la mente al momento de analizar la información de esta propiedad es ¿Qué nivel de ocupación tiene este hotel a lo largo del tiempo? Para contestar esta pregunta, se generaron las siguientes gráficas:



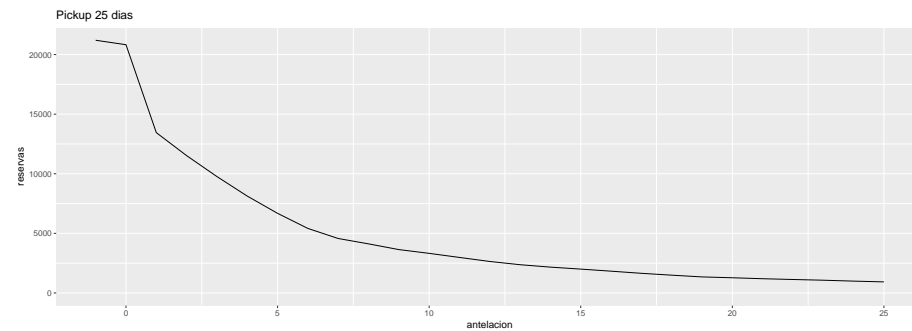
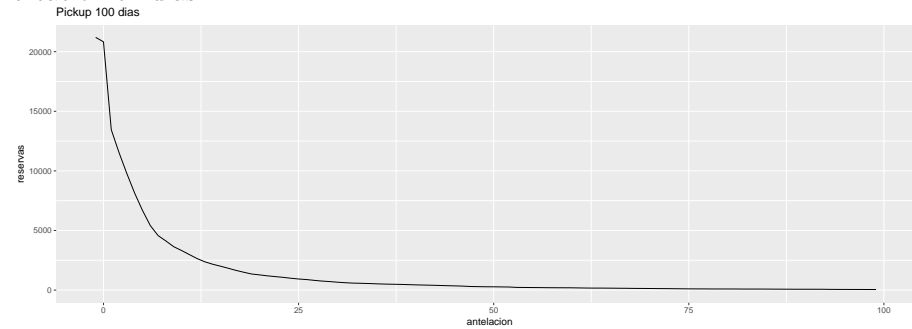
Como podemos observar esta propiedad tiene niveles de ocupación alrededor del 75% llegando en repetidas ocasiones al 100% de ocupación. Las caídas en los niveles de ocupación se deben a los fines de semana. A continuación presentamos una gráfica que nos ayudará a sustentar esta conclusión.



Pickup

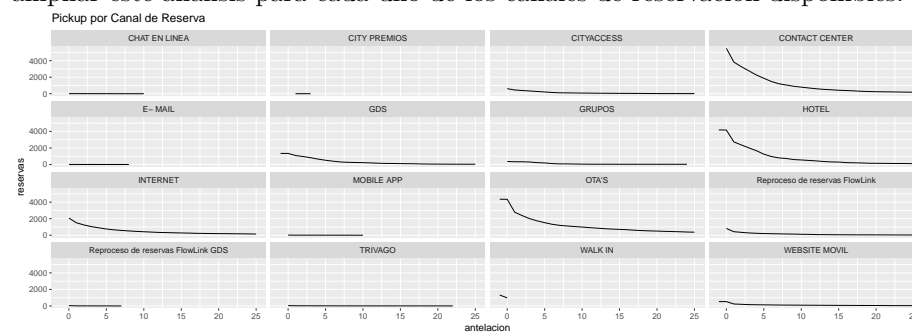
Al realizar este tipo de análisis resulta interesante estudiar el tiempo de antelación con el cual el cliente reserva un cuarto, a esto se le llama curva de pickup. Para poder graficar esta curva necesitaremos agrupar los datos por número de reservación obteniendo la mínima fecha de entrada y la máxima fecha de salida para cada una de ellas. Una vez trabajados los datos podremos calcular la antelación obteniendo la diferencia entre la fecha de entrada y la fecha de

creación en días.



Pickup por Canal de Venta

Como podemos observar en la gráfica "Pickup 25 días" el mayor número de reservaciones en esta propiedad ocurre 5 días antes de que el huésped llegue al hotel. Con esto podemos concluir que los clientes que visitan esta propiedad planean su estancia con a lo más 5 días de antelación. Resulta interesante ampliar este análisis para cada uno de los canales de reservación disponibles.

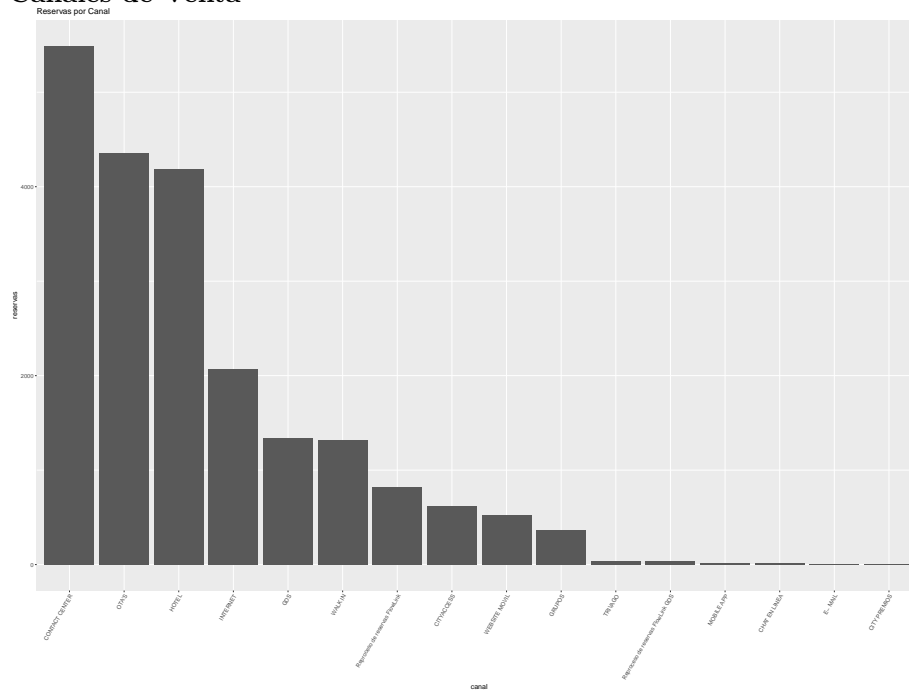


Podemos notar que para este hotel en particular el pickup es similar (alrededor de 5 días de antelación) en los canales que producen ventas de cuartos, sin embargo resulta interesante observar que no todos los canales tienen una producción significativa en esta propiedad ya que el grueso de las ventas se

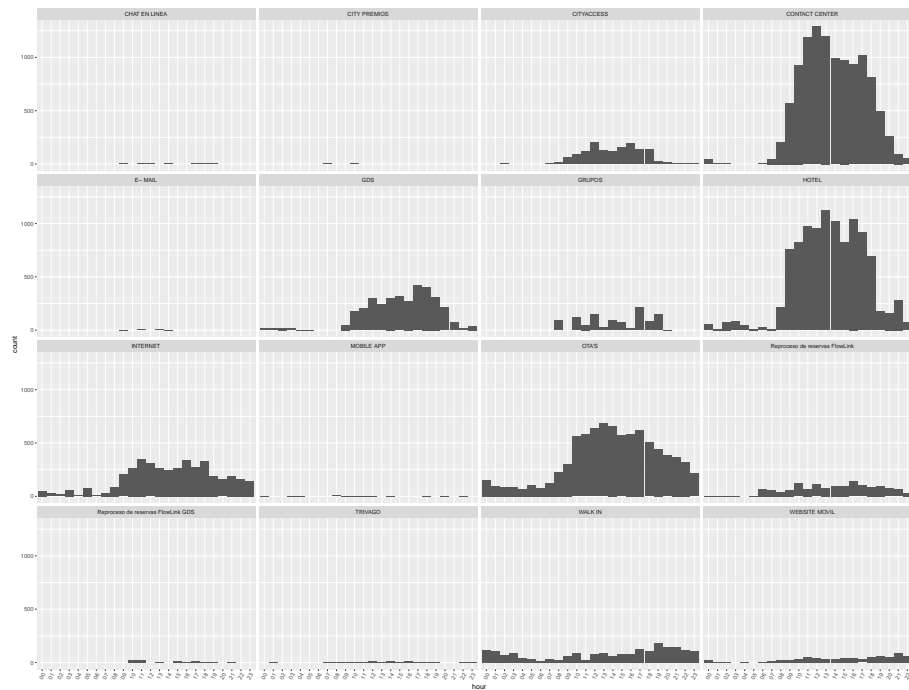
concentra en los siguientes canales:

- Contact Center
- Hotel
- Ota's
- Internet (Website)
- GDS's

Canales de Venta



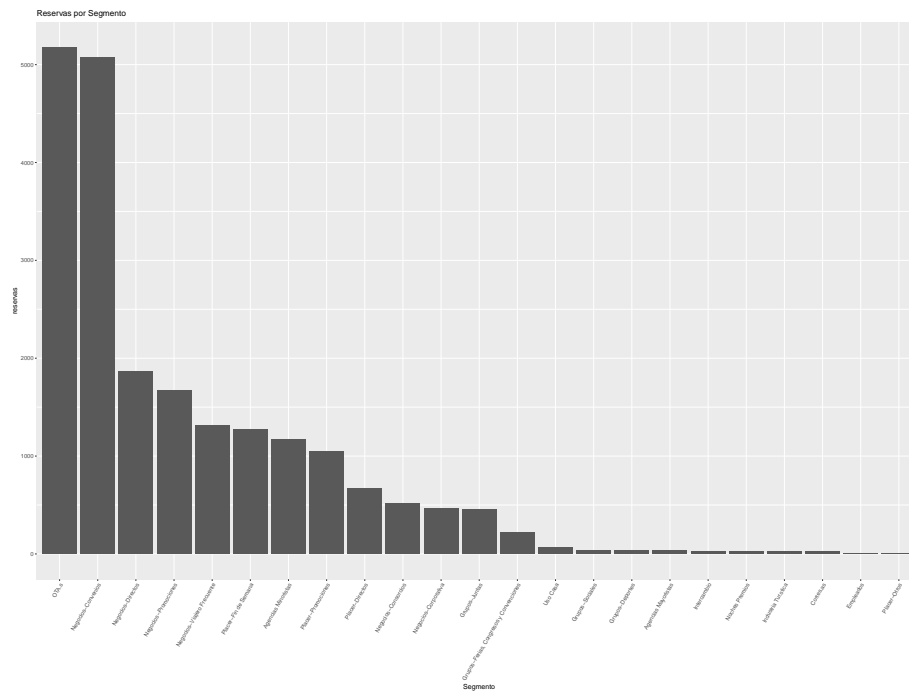
Por último estudiaremos cómo se comporta cada uno de los canales a lo largo del día, esto para poder encontrar un patrón en el uso de cada uno de los canales.



Con la gráfica anterior podemos corroborar los canales de venta más activos para este hotel y podemos concluir también que las horario en el que esta propiedad genera más reservas es de 9:00 am a 9:00 pm, con excepción del canal "walk in" el cuál tiene una distribución uniforme en las reservas generadas a lo largo del día.

Segmentos de Mercado

Como se mencionó anteriormente, City Express implementa una segmentación de mercado en cada una de las reservas que genera, esto nos permitirá analizar el perfil del huésped que visita cada una de las propiedades. A continuación presentaremos un análisis particular para una de las propiedades. A continuación se presentan los segmentos de mercado y las reservas que produjeron durante un periodo de tiempo:



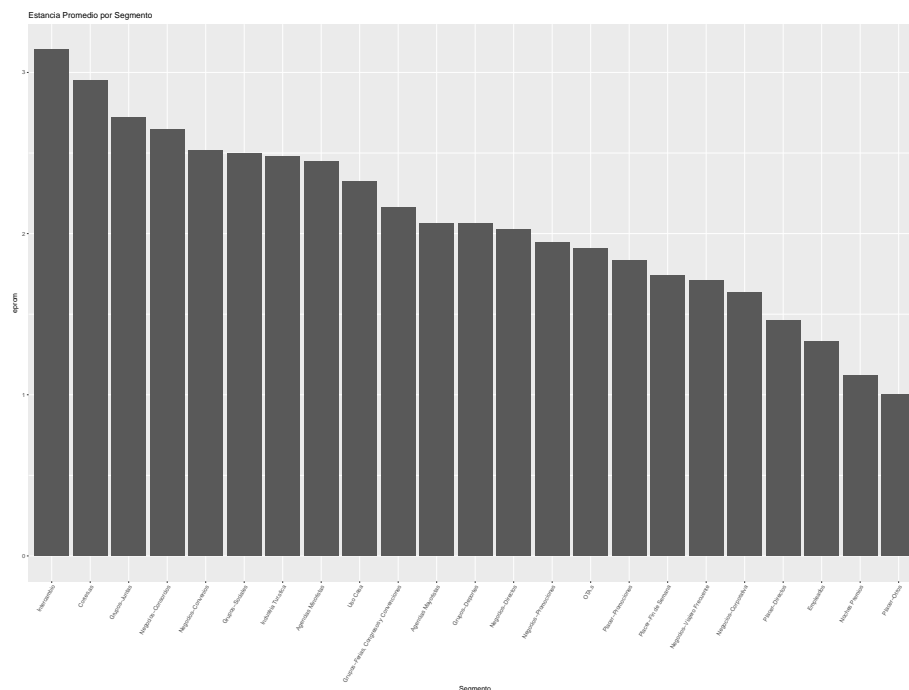
Derivado de este analisis podemos observar lo siguiente: Los segmentos que mas reservan son:

- OTA's (Agencias de Viaje en línea)
- Negocios-Convenios (Empresas con tarifa convenio)
- Negocios-Directos (Viajeros de negocios sin tarifas convenio)
- Negocios-Promociones (Viajeros de negocios con cupones promocionales)

Podemos decir que el perfil del huésped que visita esta propiedad es el viajero de negocios que visita oficinas corporativas cercanas a la ubicación del hotel durante la semana laboral (Lunes - Viernes).

Producción de Cuartos por Segmentos

Contrastaremos la gráfica anterior contra la producción de cuartos vendidos por cada uno de los segmentos para poder tener una idea de cuantos cuartos vende cada segmento en cada una de las reservaciones generadas.



Como podemos ver, la estancia promedio más alta la tiene el segmento de Negocios-Directos, con una estancia promedio de aproximadamente tres noches, contrastada con el segmento de Placer-Otros con una estancia promedio de aproximadamente noche. Esta información nos ayuda a complementar el perfil del huésped para esta propiedad ya que nos refuerza la idea de que el viajero de negocios utiliza las propiedades durante la semana laboral, y un viajero de placer puede que se hospede un día en fin de semana.

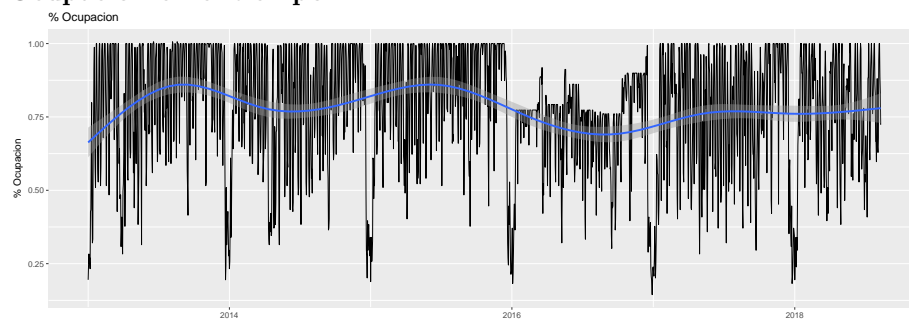
El segundo data set con el que contamos, trae información diaria sobre los niveles de ocupación, tarifa promedio, tarifa efectiva, habitaciones disponibles, penetración de ocupación y penetración de tarifa para los hoteles sujetos al estudio. Para efectos de este trabajo tomaremos información sobre el hotel con el que hemos venido trabajando.

```
##      Hotel      fecha      occ
## Length:2049   Min.    :2013-01-01   Min.    :0.1447
## Class :character 1st Qu.:2014-05-28   1st Qu.:0.6478
## Mode  :character Median :2015-10-22   Median :0.7987
##                      Mean  :2015-10-22   Mean  :0.7786
##                      3rd Qu.:2017-03-17   3rd Qu.:0.9811
##                      Max.   :2018-08-11   Max.   :1.0063
##      habs_disp      tp      po
## Min.    :159   Min.    : -267.6   Min.    :0.3158
## 1st Qu.:159   1st Qu.:1046.1   1st Qu.:0.9592
```

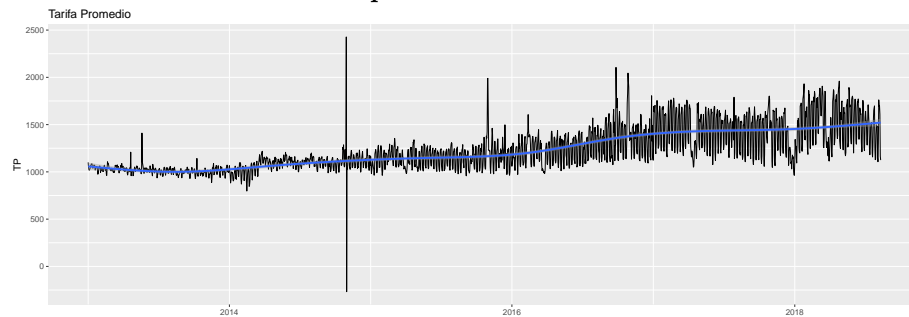
```
## Median :159      Median :1154.8    Median :1.0583
## Mean   :159      Mean   :1227.6    Mean   :1.0709
## 3rd Qu.:159      3rd Qu.:1344.8    3rd Qu.:1.1645
## Max.   :159      Max.   :2427.2    Max.   :2.3956
##      pt          te
## Min.   :0.0000    Min.   : -228.9
## 1st Qu.:0.9527    1st Qu.:  718.0
## Median :0.9941    Median :  979.4
## Mean   :1.0024    Mean   :  963.9
## 3rd Qu.:1.0333    3rd Qu.:1159.7
## Max.   :8.0211    Max.   :2320.3
```

Lo primero que podemos observar es que contamos con datos para el hotel CEXXX a partir del "2013-01-01" y hasta el "2018-11-11". Esta propiedad tiene 159 habitaciones disponibles y maneja niveles de ocupación alrededor del 81.7%; su tarifa promedio se encuentra alrededor de \$1486.60 y con PO y PT muy cercana a 1 indicando que es un hotel fuerte dentro del set competitivo de la plaza. A continuación presentamos los datos de manera gráfica.

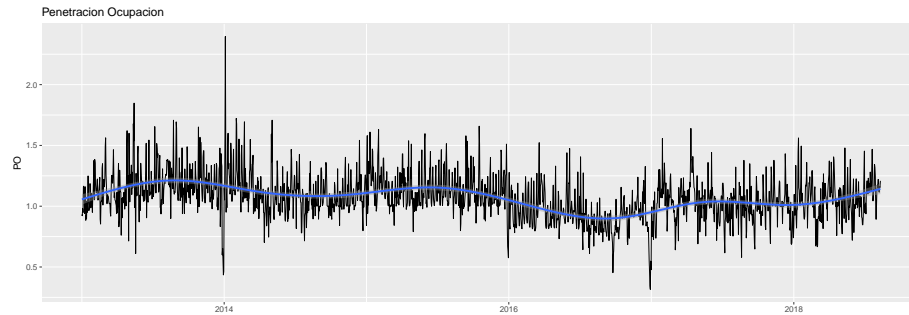
Ocupación en el tiempo



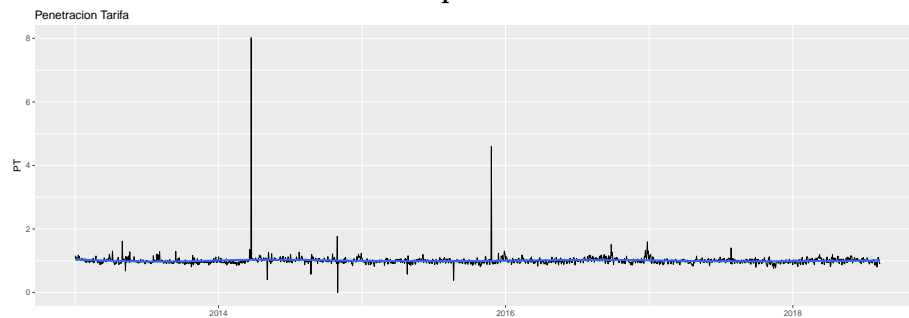
Tarifa Promedio en el tiempo



Penetración de Ocupación en el tiempo

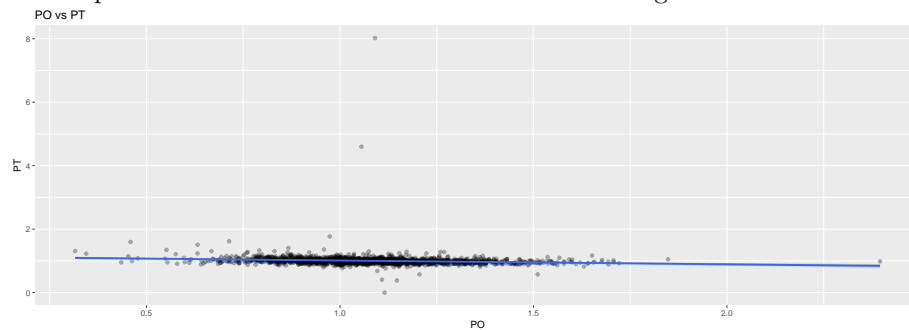


Penetración de Tarifa en el tiempo



Penetración de Ocupación vs Penetración de Tarifa

Al estudiar los datos notamos que hay una correlación negativa entre la penetración de ocupación y la penetración de la tarifa, recordemos que la primera indica cuanta ocupación tiene una de las propiedades de CityExpress con respecto a la ocupación de su plaza, y la segunda indica cuánto del ingreso disponible se está quedando dentro de la propiedad de interés. En otras palabras podemos pensar en una relación similar a la del precio de un bien vs la demanda de un bien. A mayor precio menor demanda y a menor precio mayor demanda, lo cual indica que la relación entre ambas variables debe ser negativa.

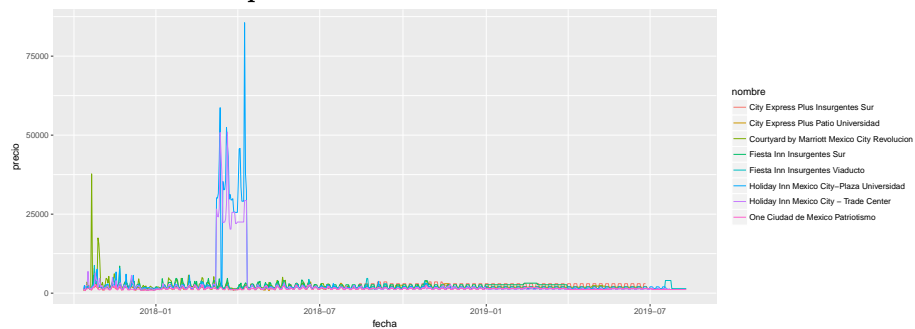


Precio de la competencia

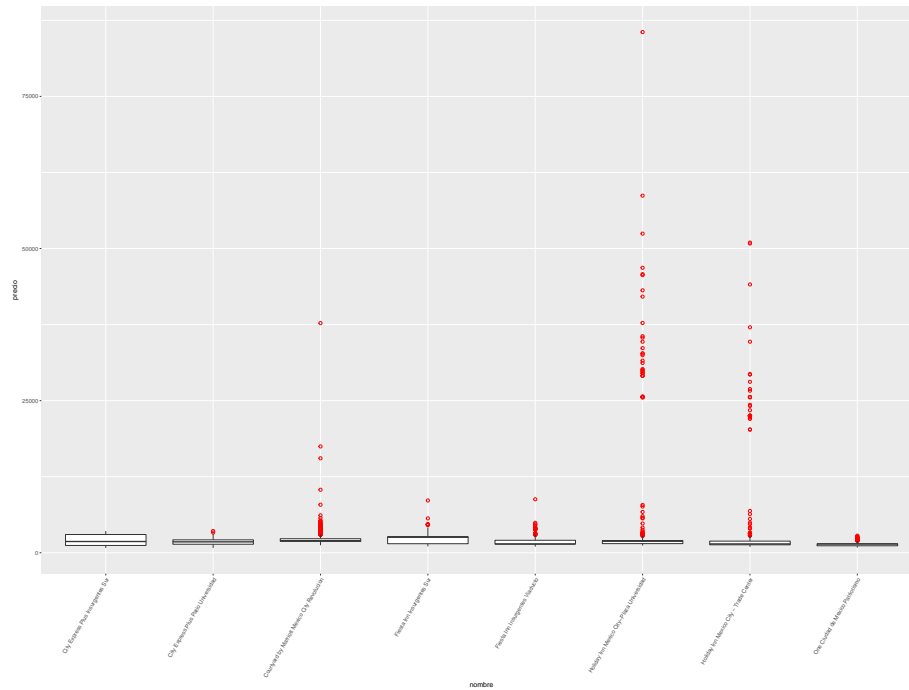
Para este proyecto contamos con un dataset que contiene información sobre los precios de la competencia definida para CEXXX publicados en las agencias de viajes en línea.

##	nombre	fecha	precio
##	Length:5109	Min. :2017-10-13	Min. : 794
##	Class :character	1st Qu.:2018-03-22	1st Qu.: 1399
##	Mode :character	Median :2018-08-29	Median : 1650
##		Mean :2018-08-29	Mean : 2244
##		3rd Qu.:2019-02-05	3rd Qu.: 2122
##		Max. :2019-08-10	Max. :85572

Precios en el tiempo



Nos llaman la atención algunos datos que salen del patrón del comportamiento, para estudiarlos con mas detalle veamos la siguiente gráfica:



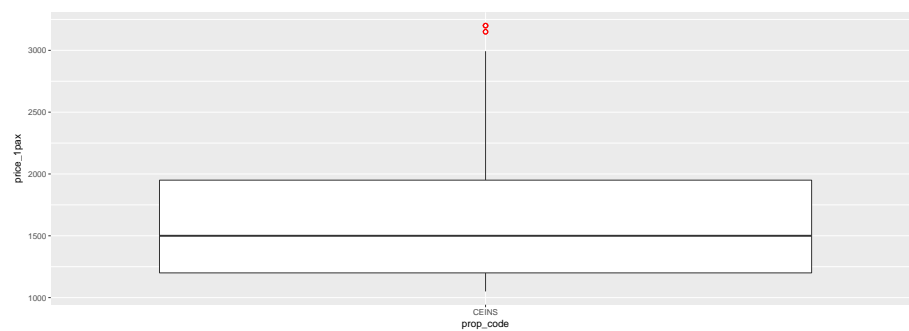
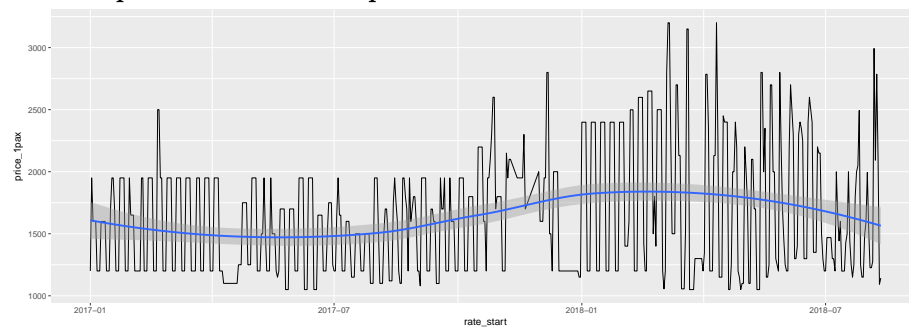
Podemos observar que este data set contiene datos atípicos, por ejemplo los hoteles "Courtyard by Marriot Mexico City Revolución", "Holiday Inn Mexico City-Plaza Universidad" y "Holiday Inn Mexico City - Trade Center" tienen tarifas arriba de los \$10,000 MXN, lo cual es poco probable ya que son hoteles que compiten directamente con CityExpress y su mercado es el mismo (Viajero de Negocios). Estos casos se deberán tomar en cuenta al momento de construir el modelo de pronóstico y pricing dinámico para evitar resultados erróneos.

Precios publicos Estudiaremos también los precios públicos para la propiedad que esta siendo sujeta al análisis

```
##   prop_code      rate_start      rate_key
## Length:575      Min.   :2017-01-01      Length:575
## Class :character 1st Qu.:2017-05-24      Class :character
## Mode  :character Median :2017-10-15      Mode  :character
##                  Mean  :2017-10-20
##                  3rd Qu.:2018-03-20
##                  Max.   :2018-08-11
##   curr_code      price_1pax
## Length:575      Min.   :1050
## Class :character 1st Qu.:1200
## Mode  :character Median :1500
##                  Mean   :1641
```

```
##          3rd Qu.:1950
##          Max.    :3200
```

Precios publicos en el tiempo



Calidad

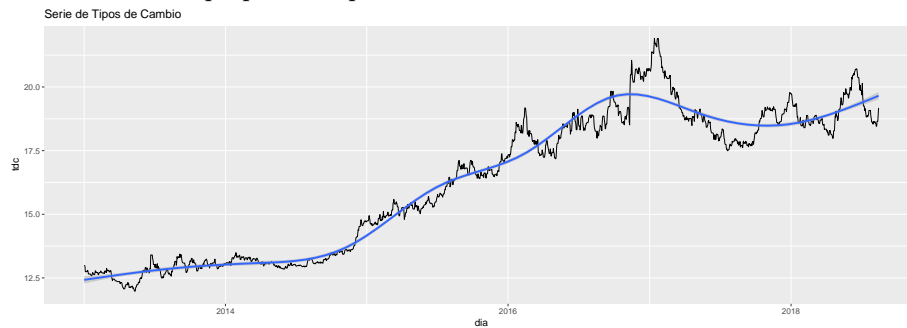
```
##      plaza      hotel      calidad
## Length:779    Length:779    Min.   :0.000
## Class :character Class :character 1st Qu.:3.400
## Mode  :character Mode  :character Median :4.000
##                                     Mean  :3.596
##                                     3rd Qu.:4.400
##                                     Max.   :5.000
```

Calidad vs Precio


```
## Error in convertUnit(x, unitTo, "x", "dimension", "x", "dimension",  
valueOnly = valueOnly): 'x' argument must be a unit object
```

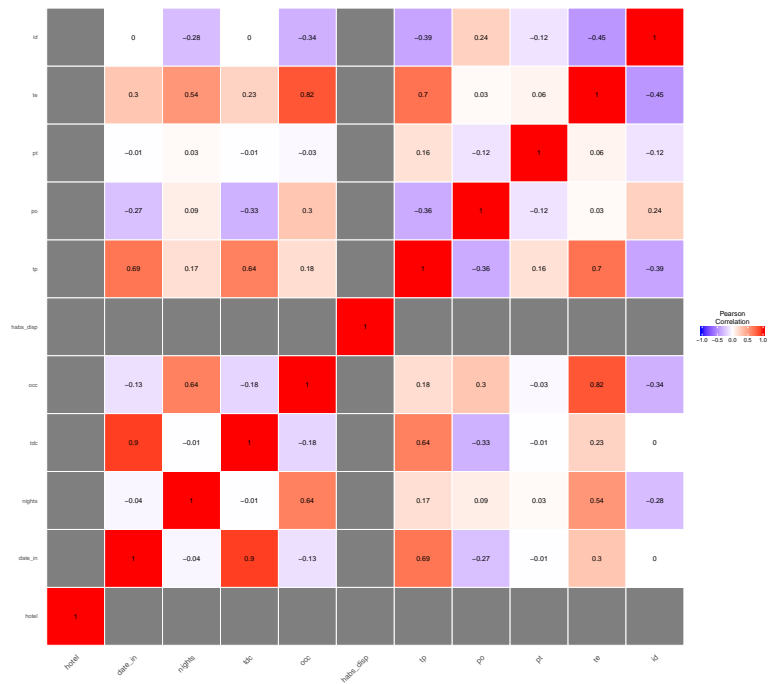
Tipo de Cambio

Otro data set con el que contamos es el tipo de cambio, a continuación mostramos las series de tiempo para el tipo de cambio:

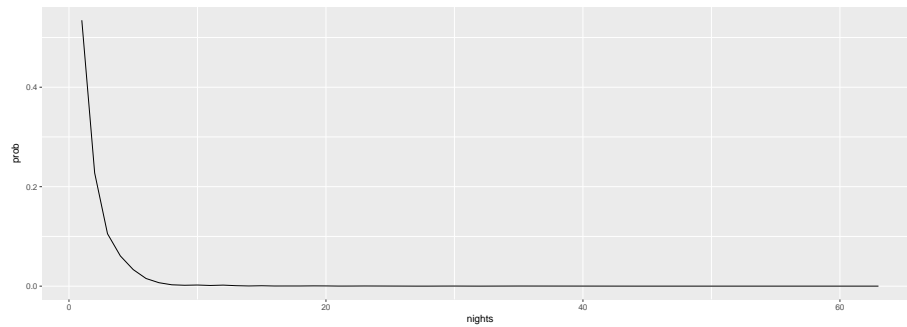


Correlación de variables

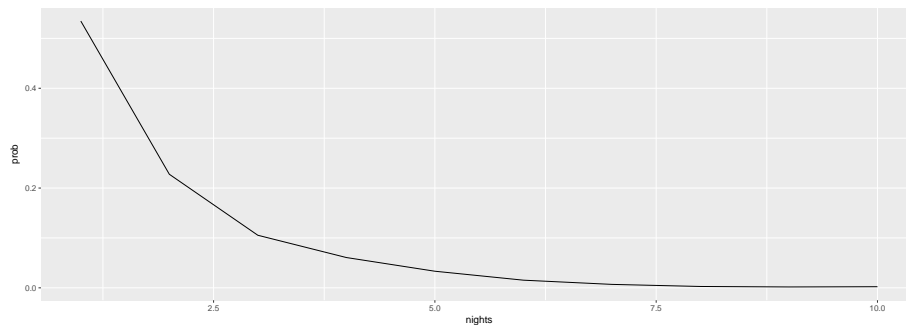
Para concluir el análisis exploratorio de datos estudiaremos la correlación de cada una de las variables contra el nivel de ocupación, esto nos ayudará a decidir qué variables debemos utilizar dentro del modelo construido.



Duración de Estancia



Podemos ver que la longitud de estancia se concentra entre 0 y 10 noches, por lo tanto estudiaremos este intervalo de cerca



Derivado de este análisis podemos concluir lo siguiente:

Conclusiones

4.4 Propuesta de Modelo Predictivo

Antecedentes del modelo

Se le ha llamado "yield management" a la estrategia de variar precios basándose en el entendimiento, anticipación e influenciando el comportamiento de los clientes con el fin de poder maximizar el ingreso o utilidad generada por la venta de un inventario fijo de productos perecederos o que tienen un tiempo de venta limitado (como asientos en un vuelo o cuartos de hotel). La estrategia de "yield management" involucra una estrategia de control de inventario para vender el producto correcto al cliente correcto en el tiempo correcto por un precio correcto. Dicho proceso puede resultar en una discriminación de precio, en la cual clientes consumiendo el mismo producto o servicio pagan precios totalmente diferentes.

Para poder aplicar el concepto de "yield management" necesitamos que se den tres condiciones:

- Debe existir un inventario fijo disponible para la venta.
- Los productos deben de ser perecederos (o deben contar con un tiempo limite de venta. Una vez transcurrido este tiempo el producto pierde su valor).
- Diferentes clientes están dispuestos a pagar diferentes precios por el mismo producto.

Las estrategias de "yield management" han alterado la industria del transporte y hospitalidad desde los años 80's. Desde el momento en el que estas estrategias fueron concebidas, estas empresas han requerido contar con analistas con un conocimiento detallado del mercado así como con sistemas computacionales avanzados en los cuales se implementan sofisticadas técnicas matemáticas que analizan el comportamiento del mercado y capturan oportunidades de ingresos.

Yield Management en la hotelería

En el caso de uso específico para la hotelería podemos estudiar la venta de habitaciones como la venta de un inventario fijo (difícilmente se puede incrementar o cambiar), que además las habitaciones que no se venden cada noche dejarán de producir ingresos a la propiedad por lo que se puede decir que caducaron. Es por ello que algunos hoteles utilizan software especializado para monitorear como es que los cuartos están siendo vendidos y reaccionar con base en sus observaciones. Hay varios métodos de control del inventario por ejemplo, ofrecer descuentos en propiedades con baja demanda o caso contrario, aumentar precios en propiedades con alta demanda. Otra forma de controlar el inventario es mediante la venta anticipada, es decir, se reserva parte del inventario para la venta anticipada a 30 días, 15 días, 7 días y un día y conforme se van agotando los cuartos los precios van cambiando, de tal forma que si pocos cuartos de una categoría han sido reservados, los precios para esa categoría bajan.

Trabajo previo

Existe una gran cantidad de literatura que describe trabajos realizados previamente con el objetivo de construir un modelo que pronostique la demanda de un producto y a partir de este pronóstico se fije un precio que maximice el ingreso del inventario.

Para la construcción de este modelo se revisaron artículos disponibles así como trabajos previos en esta área y se obtuvieron las siguientes conclusiones:

- La mayoría de los modelos construídos anteriormente parten el problema de pricing en dos:
 - Modelo de pronóstico de demanda
 - Modelo de pricing dinámico
- La mayoría de los artículos utilizan una simulación de montecarlo para simular la ocupación a futuro de las propiedades
- Se debe estudiar cuidadosamente la temporalidad de la ocupación en el hotel
- Se pueden utilizar factores multiplicadores que dependen de una serie de variables para poder asignar el nuevo precio del producto.
- La demanda de la propiedad (en el caso de hoteles) puede ser modelada como un proceso poisson.
- Se pueden incluir heurísticas de control de inventario, aunque esto tiende a complicar el modelado.
- El precio dependerá también del inventario que aún está disponible en cualquier momento en el que se dé una reservación.

Tomamos estas conclusiones como punto de partida para el modelo realizado en CityExpress.

Descripción del modelo

Para el desarrollo del modelo utilizaremos los data sets que contienen la infor-

mación de las reservaciones modelados como curvas de "pick up". Estas curvas nos describen la velocidad en que un hotel vende su inventario para un día en específico. Lo que el modelo hará es ajustar una curva que resuma el comportamiento del hotel para un día similar al que se está estudiando, de esa manera podremos predecir cuantos cuartos venderá el hotel y con cuantos días de antelación. Para lograr esto se construirá un modelo de regresión Poisson fundamentado en la distribución de la llegada de las reservaciones como un proceso discreto que depende del tiempo y para darle un enfoque de uso predictivo al modelo, se utiliza un modelo de regresión lineal con predictores que dependen del tiempo para dar un valor estimado por día y por hotel de las curvas de "pick Up".

Modelo de regresión Poisson

Por la naturaleza del estudio que se está realizando se optó por implementar una regresión Poisson, la cuál es un modelo lineal generalizado usado comúnmente en problemas de conteos de datos y tablas de contingencia. La regresión de Poisson asume que la variable de respuesta Y tiene una distribución de Poisson y también asume que el logaritmo de su valor esperado puede ser modelado por una combinación lineal de los parámetros desconocidos. El modelo de regresión Poisson puede ser expresado como sigue:

$$\log(E(Y|X)) = \beta_0 + \beta_1 X$$

Para poder aplicar el modelo a la información proporcionada, se contruyeron series por día y hotel en función del número de días de antelación para la reservación.

La selección del modelo de regresión Poisson se fundamenta en la distribución de la llegada de las reservaciones como un proceso discreto que depende del tiempo. El conteo de las reservaciones es inversamente proporcional a la cercanía de la fecha de entrada al hotel y por tanto la probabilidad de ocurrencia de llenado del hotel incrementa conforme se acerca la fecha para la que se hace la reservación.

Modelo de regresión lineal

Considerando que las principales variables que describen la predictividad de la demanda son temporales, se propone un modelo que explique los parámetros de las curvas de "pick up" mediante un modelo lineal. Se generan series temporales para hacer predictivo el modelo de pronóstico de demanda y proyectamos en función de la historia en el mismo día calendario del año anterior, el día de la semana para compensar los efectos diferenciados por hotel, así como los eventos que afectan la demanda en un calendario anticipado por hotel, y finalmente el comportamiento de las tarifas que ofrece la competencia.

$$\log(E(Y|X)) = \beta_0 + \beta_1 X$$

4.5 Interpretación del modelo

El modelo genera como resultado un data set similar al que se muestra a continuación:

##	hotel	dia	AAbeta0	AAbeta1	AAvariacion		
## 1	CEINS	2018-01-01	2.335294	-2.489863e-06	0.1884625		
## 2	CEINS	2018-01-02	2.984073	-2.818019e-06	0.1321997		
## 3	CEINS	2018-01-03	3.301841	-3.535574e-06	0.3262416		
## 4	CEINS	2018-01-04	3.782523	-2.295688e-06	0.1744051		
## 5	CEINS	2018-01-05	3.761697	-1.089675e-06	0.0925871		
## 6	CEINS	2018-01-06	3.140137	-2.292639e-06	0.1887511		
##	tdc	PO	PT	diasem	mes	eventos1	eventos
## 1	19.7354	0.840382	0.872720	lunes	ene.	0	0
## 2	19.7354	1.096970	1.025310	martes	ene.	0	0
## 3	19.6629	0.927171	0.992081	miércoles	ene.	0	0
## 4	19.4899	1.045640	1.024860	jueves	ene.	0	0
## 5	19.3717	0.927771	0.953648	viernes	ene.	0	0
## 6	19.2427	1.334990	0.908828	sábado	ene.	0	0
##	pred.beta0	pred.beta1					
## 1	3.159533	-0.12278602					
## 2	3.916035	-0.11603131					
## 3	3.874017	-0.10891126					
## 4	4.159721	-0.10283409					
## 5	3.698266	-0.08467446					
## 6	3.771660	-0.09371103					

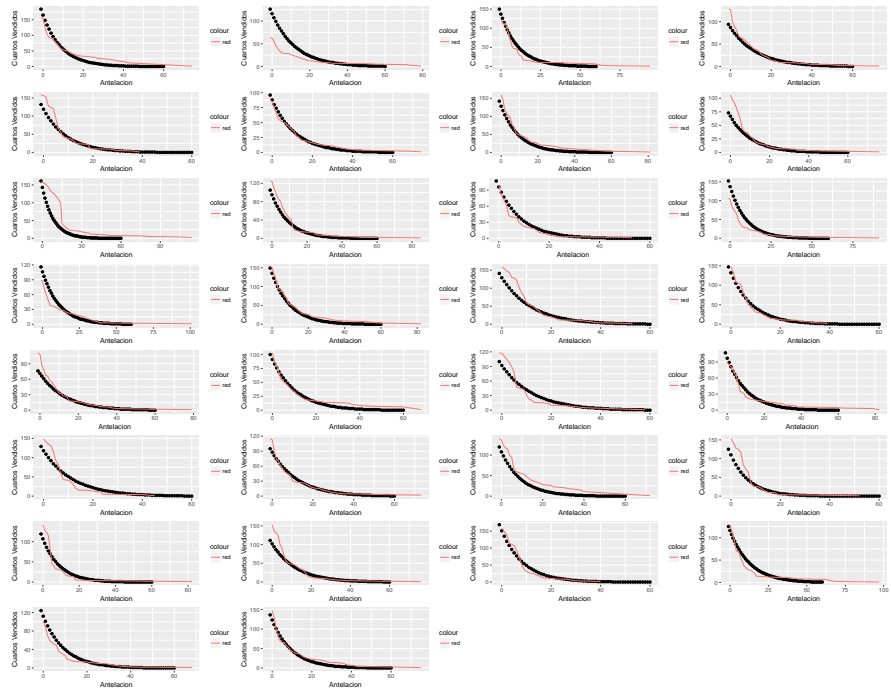
Las variables de interés son: "pred.beta0" y "pred.beta1" ya que son los parámetros generados por los modelos y que nos ayudarán a reconstruir las curvas de pickup. Para reconstruir la curva de pickup de un día en específico, debemos obtener los parámetros "pred.beta0" y "pred.beta1" para el día elegido y posteriormente debemos calcular hacer el siguiente cálculo:

$$E[y|x] = e^{\beta_0 + \beta_1 x}$$

Dónde:

- $E[y|x]$ = El valor esperado de cuartos noches para un día en específico
- β_0 = pred.beta0
- β_1 = pred.beta1
- x = días de antelación al día analizado

Validamos el modelo con fechas aleatorias y obtenemos las siguientes gráficas:



Modelo simple de pricing dinámico

Una vez pronosticado el nivel de ocupación para cada una de las propiedades de la cadena, se alimentó un modelo de pricing dinámico que entrega recomendaciones de precios para la tarifa pública tomando en cuenta la demanda pronosticada para cierto día del año.

Problema de asignación de precios

Típicamente el gerente de la propiedad controla la cantidad de cuartos ofrecidos a diferentes precios, es decir, se asigna cierto número de cuartos a cada nivel de tarifa de tal forma que cuando el inventario asignado al precio mas bajo se agota, se consume el inventario asignado al siguiente nivel de tarifa que será mayor al primer precio ofertado.

A continuación se muestra un esquema donde se ejemplifica la idea mencionada anteriormente:

Como se puede observar, tenemos un inventario de 32 cuartos dividido en 4 niveles de precios:

- 12 cuartos son ofrecidos a un valor de 110 USD

Deluxe King, 32 rooms city view

\$110	\$110	\$110	\$110	\$110	\$110	\$110	\$110
\$110	\$110	\$110	\$110	\$120	\$120	\$120	\$120
\$120	\$120	\$120	\$120	\$130	\$130	\$130	\$130
\$130	\$130	\$130	\$155	\$155	\$155	\$155	\$155

Figure 2: Asignacion de inventario

- 8 cuartos son ofrecidos a un valor de 120 USD
- 7 cuartos son ofrecidos a un valor de 130 USD
- 5 cuartos son ofrecidos a un valor de 155 USD

Hay que considerar que el hecho de poder encontrar el precio que maximiza el ingreso de una propiedad constituye un problema complejo de optimización, en donde el hotel debería explorar un gran número de escenarios, sin embargo, para superar la complejidad computacional que este calculo requiere muchos hoteles han optado por usar varias simplificaciones por ejemplo, utilizar un enfoque basado en la asignación de precios, en la cual se pre asigna un número fijo de inventario disponible a cada nivel de precios y los precios pueden variar a lo largo del tiempo.

Formulación Matemática

Un problema de maximización de ingresos es básicamente un problema de optimización sujeto a restricciones, en este caso, se quiere maximizar la siguiente función objetivo:

$$\sum_{n=0}^n p_i * o_i$$

En donde i es la índice de la noche, p_i es el precio del cuarto para la i -ésima noche y o_i es la ocupación pronosticada (demanda) para la i -ésima noche al precio p_i . Asumimos también que la función de la ocupación (demanda) está

sujeta al precio bajo la siguiente relación:

$$o = o_{nominal} * \left(\frac{p}{p_{nominal}} \right)^e$$

Donde $o_{nominal}$ corresponde a la ocupación pronosticada para una noche dada tomando como base un precio nominal ($p_{nominal}$). Para este caso el $p_{nominal}$ es igual a la tarifa pública promediada a lo largo del año. El valor de e (elasticidad) toma un valor = -2. En otras palabras, p incrementa en un 10% y la demanda decrece cerca de un 20%.

Para poder hacer una asignación dinámica de los precios, evitando tener que correr el modelo conforme la información del hotel es actualizada, se partió el inventario en 4 niveles de disponibilidad (159,120,80 y 40 cuartos disponibles). Para cada nivel de capacidad se resolvió la función objetivo para obtener 4 conjuntos de precios, de esta forma, el gerente de la propiedad puede saber a qué nivel de precio debe subir la tarifa pública de la propiedad conforme las habitaciones disponibles decrecen.

Limitantes del modelo

Este modelo se formuló considerando los siguientes puntos:

- Se consideran estancias de una sola noche
- No se toman en cuenta reservaciones para grupos ni cancelaciones
- Asumimos paridad de precios entre los canales, es decir, los precios no varían entre los canales de venta
- Se considera solamente un tipo de habitación, el incremento por cambio de habitación es un monto fijo conocido como gap
- Se asume un valor para la elasticidad (-2) que es un valor razonable para la industria

```
## Error in eval(substitute(expr), envir, enclos): Failed to open  
'C:\Users\lucky\Documents\GitHub\mcdatos_estancia\result\forecasted_demand.csv'.
```

Resultados del modelo de pricing

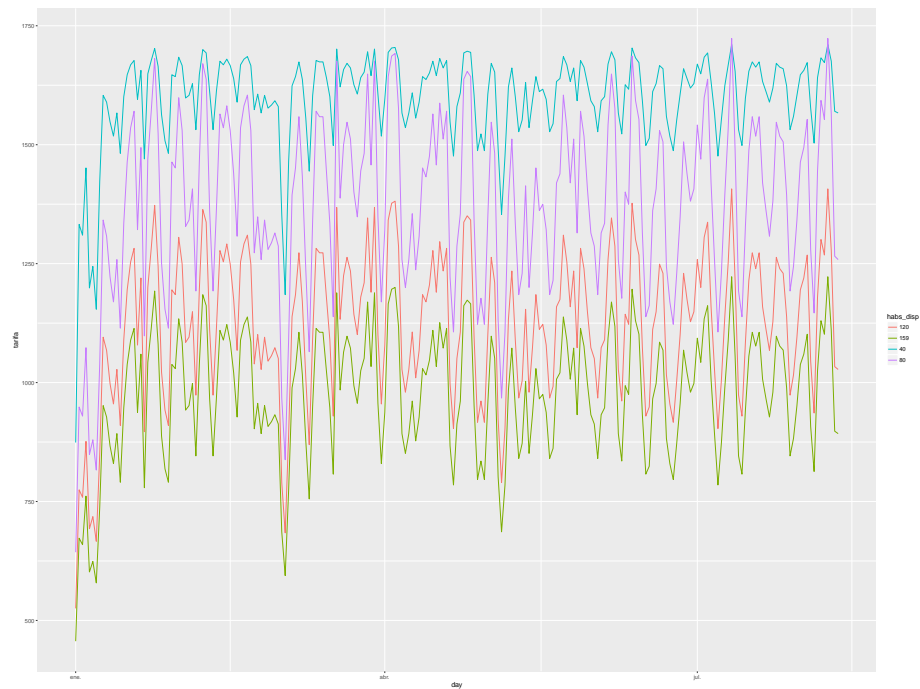
El modelo de pricing, como comentamos anteriormente, arroja como resultado una matriz de día vs precio por niveles de inventario. A continuación se muestra un ejemplo:

##	day	40	80	120	159
## 137	2018-05-17	1616.650	1374.773	1122.4972	975.1633
## 77	2018-03-18	1701.113	1675.709	1368.2105	1188.6253
## 52	2018-02-21	1666.334	1529.706	1248.9996	1085.0615
## 16	2018-01-16	1647.043	1463.557	1194.9895	1038.1406

##	24	2018-01-24	1702.217	1681.071	1372.5888	1192.4289
##	93	2018-04-03	1703.315	1686.416	1376.9532	1196.2205
##	78	2018-03-19	1621.479	1387.804	1133.1372	984.4067
##	167	2018-06-16	1498.239	1138.420	929.5160	807.5119
##	139	2018-05-19	1527.223	1184.905	967.4709	840.4850
##	174	2018-06-23	1518.070	1169.615	954.9869	829.6396

Graficando los resultados:

##	day	40	80	120	159
## 1	2018-01-01	874.0001	643.4283	525.3570	456.4010
## 2	2018-01-02	1332.4657	948.6833	774.5967	672.9266
## 3	2018-01-03	1309.8597	929.5160	758.9466	659.3307
## 4	2018-01-04	1451.1448	1073.3126	876.3561	761.3295
## 5	2018-01-05	1198.8353	848.5281	692.8203	601.8838
## 6	2018-01-06	1244.1752	879.7727	718.3314	624.0464



5 Anexo A: Implementación del modelo pronóstico de demanda

```

library(plyr)
HIS_CEINS<-read_csv("datos/EST_PAS_UNIC_CEINS.csv")
RES_CEINS <- read_csv("datos/CEINS_RESERVAS.csv")
#Juntamos todas las tablas en un solo dataset
#datos <- rbind(HIS_CEINS, RES_CEINS)
datos <- HIS_CEINS
#Modelo de regresion Poisson para obtener
#parametros de modelo predictivo
modelo.1<-function(base,h){
  hotel<-subset(base,prop_code==h)
  dia<-sort(unique(hotel$date_in))
  ndia<-length(dia)
  beta0<-rep(0,ndia)
  beta1<-rep(0,ndia)
  variacion<-rep(0,ndia)
  for(i in 1:ndia){
    eleccion <- ddply(subset(hotel,as.character(date_in)==
                           as.character(dia[i])),
                      select = c('antelacion','nights')),
                    .(antelacion),summarise,
                    nights = sum(nights))
    eleccion <- eleccion[with(eleccion,order(-antelacion)), ]
    eleccion$nights <- cumsum(eleccion$nights)
    eleccion <- eleccion[eleccion$antelacion<=60,]
    mod <- glm(nights~antelacion, family = "poisson",
               data = eleccion)
    coeficientes <- coef(mod)
    beta0[i] <- coeficientes[1]
    beta1[i]<- coeficientes[2]
    variacion[i] <- sqrt(mod$deviance)
  }
  yresp <- data.frame(hotel=rep(h,ndia),dia,beta0,beta1,variacion)
  return(yresp)
}

# Prepara datos 2017 de entrada del modelo
res <- ddply(datos,.(prop_code,date_create,date_in),
             summarise, nights=sum(nights))
res$prop_code<-toupper(res$prop_code)
res$antelacion <- as.numeric(res$date_in - res$date_create)
res <- subset(res,format(res$date_in, "%Y")=="2017")
yresp.1 <- modelo.1(res,"CEINS")
yresp<-yresp.1

```

```

### Preparacion de datos para modelo predictivo (2018)
series2018 <- subset(yresp,format(yresp$dia, "%Y")=="2017")
series2018$dia <- as.Date(series2018$dia)+365
names(series2018) <- c("hotel","dia","AAbeta0",
                      "AAbeta1","AAvariacion")

### Series a la fecha de corte en la extraccion

#Reservas
RES_CEXXX<-RES_CEINS
#
res<-RES_CEXXX
res <- ddply(res,.(prop_code,date_create,date_in),
             summarise, nights=sum(nights))
res$prop_code<-toupper(res$prop_code)
res<-res %>% filter(date_create!="0000-00-00 00:00:00") %>%
  mutate(date_create=as.Date(date_create),
         date_in=as.Date(date_in))
res$antelacion <- as.numeric(res$date_in - res$date_create)
series2018al11ago <- subset(series2018,as.numeric(dia)<
                           as.numeric(as.Date("2018-08-11")))
res2018al11ago <- subset(res,as.numeric(date_in)<
                       as.numeric(as.Date("2018-08-11")))
res2018al11ago <- subset(res2018al11ago,
                       format(res2018al11ago$date_in, "%Y")=="2018")

### Aplicacion del modelo a la fecha de corte por extraccion
param.1 <- modelo.1(res2018al11ago,"CEINS")

param2018 <- param.1
series2018al11ago <- join(series2018al11ago,param2018)

### Generacion de predictores de modelo predictivo
series2018al11ago$diasem <- as.factor(weekdays
                                     (series2018al11ago$dia))
series2018al11ago$mes <- as.factor(format(
  series2018al11ago$dia, "%b"))
series2018al11ago <- join(series2018al11ago,TDC)
series2018al11ago$eventos <- rep(0,nrow(series2018al11ago))
series2018al11ago$eventos <- as.factor(ifelse(
  as.numeric(series2018al11ago$dia)
  <=as.numeric(as.Date("2018-08-11")) &
  as.numeric(series2018al11ago$dia)>=
  as.numeric(as.Date("2018-03-27")),
  1,0))

```

```

#series2018al11ago <-join(series2018al11ago,eventos)
series2018al11ago$eventos<-as.factor(series2018al11ago$eventos)
PO_PT <- indicadores %>% select(Hotel,fecha,po,pt)

names(PO_PT) <- c("hotel","dia","PO","PT")
series2018al11ago <- join(series2018al11ago,PO_PT)

### Generacion de modelos predictivos por hotel
#CEINS
datos2018al11agoCEINS <- subset(series2018al11ago,hotel=="CEINS")
mod.CEINS.beta0 <- lm(beta0~AAbeta0+diasem+eventos+tdc+PO+PT,
                      data=datos2018al11agoCEINS)
datos2018al11agoCEINS$pred.beta0 <-predict(mod.CEINS.beta0,
                                           type='response')
datos2018al11agoCEINS$AAbeta1[is.na(
                                datos2018al11agoCEINS$AAbeta1)]<-0
mod.CEINS.beta1 <- lm(beta1~AAbeta1+diasem+eventos+tdc+PO+PT,
                      data=datos2018al11agoCEINS)
datos2018al11agoCEINS$pred.beta1<-predict(mod.CEINS.beta1,
                                           type='response')

### Aplica los modelos a la serie completa 2018
series2018 <- join(series2018,TDC)
series2018$AAbeta1[is.na(series2018$AAbeta1)]<-0
series2018 <- join(series2018,PO_PT)
series2018$diasem <- as.factor(weekdays(series2018$dia))
series2018$mes <- as.factor(format(series2018$dia, "%b"))

series2018$eventos1 <- ifelse(as.numeric(series2018$dia)<=
                             as.numeric(as.Date("2018-04-04")) &
                             as.numeric(series2018$dia)>=
                             as.numeric(as.Date("2018-03-27"))),
                             1,0)

series2018$eventos <- series2018$eventos1
series2018$eventos <- as.factor(series2018$eventos)
series2018$pred.beta0 <- ifelse(series2018$hotel=="CEINS",
                                predict(mod.CEINS.beta0,
                                         newdata=series2018,
                                         type="response"),0)
series2018$pred.beta1 <- ifelse(series2018$hotel=="CEINS",
                                predict(mod.CEINS.beta1,
                                         newdata=series2018,
                                         type="response"),0)

```

6 Anexo B: Implementación del Modelo de Pricing

```
import pandas as pd
import numpy as np
import scipy
import math as mt
import pickle as pkl
import os
import psycpg2
import matplotlib
import matplotlib.pyplot as plt
from numpy.random import normal
import calendar
from scipy.optimize import curve_fit
%matplotlib inline
plt.rcParams['figure.figsize'] = (16,8)
import warnings
warnings.filterwarnings('ignore')
import plotly.plotly as py
import plotly.graph_objs as go
import plotly.figure_factory as ff
from plotly.offline import download_plotlyjs,
init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
from datetime import datetime
from datetime import timedelta

# Cargamos los resultados de la regresion poisson para pronosticar ocupacion
data = pd.read_csv("data/forecasted_demand.csv")
date_start = min(data.dia.values)
date_end = max(data.dia.values)
date_end=datetime.strptime(date_end, '%Y-%m-%d')
date_end= date_end + timedelta(days=1)
nb_days = (pd.to_datetime(date_end) - pd.to_datetime(date_start)).days
forecasted_demand = data.cn_predic.values
forecasted_demand=pd.DataFrame(forecasted_demand,
                               index=pd.date_range(start=date_start,
                                                    end=date_end,closed='left'),
                               columns=['occupancy'])
forecasted_demand.occupancy=data.cn_predic.values

# Let's assume a demand price elasticity function:
```

```

def demand_price_elasticity(price, nominal_demand, elasticity=-2.0, nominal_price=120.0):
    """Returns demand given a value for the elasticity, nominal demand and nominal price.

    Parameters
    -----

    price (numpy.ndarray):
        one-dimensional price array. The length of that array should correspond to the
        length of the forecast period.

    nominal_demand (numpy.ndarray):
        one-dimensional forecasted occupancy array. The length of that array should
        correspond to the length of the forecast period.

    elasticity (float):
        value of the elasticity between price and demand. A value of e=-2 is reasonable.

    nominal_price (float):
        room rate for which the forecast was computed.

    Returns
    -----

    A numpy.ndarray of expected demand.
    """

    return nominal_demand * ( price / nominal_price ) ** (elasticity)

import scipy.optimize as optimize

# definition of the objective function:

def objective(p_t, nominal_demand=np.array([50,40,30,20]),
              elasticity=-2.0, nominal_price=1200.0):
    """
    Definition of the objective function. This is the function that want to minimize.
    (minus sign in front)

    Parameters
    -----

    p_t (numpy.ndarray):
        one-dimensional price array. The length of that array should correspond to the
        length of the forecast period.

```

```

    nominal_demand (numpy.ndarray):
        one-dimensional forecasted occupancy array. The length of that array should
        correspond to the length of the forecast period.

    elasticity (float):
        value of the elasticity between price and demand. A value of e=-2 is
        reasonable.

    nominal price (float):
        room rate for which the forecast was computed.

    Returns
    -----

    Value of the objective function (float).

    Note: here we're trying to minimize the objective function. That's where the
    minus sign comes_in.

    """

    return (-1.0 * np.sum( p_t * demand_price_elasticity(p_t, nominal_demand=nominal_demand,
                                                         elasticity=elasticity,
                                                         nominal_price=nominal_price) )) / 10

def constraint_1(p_t):
    """ This constraint ensures that the prices are positive.
    """
    return p_t

def constraint_2(p_t, capacity=20, forecasted_demand=35.0,
                 elasticity=-2.0, nominal_price=1200.0):
    """ This constraint ensures that the demand does not exceed
    capacity.

    Parameters
    -----

    p_t (float):
        Room price

    capacity (integer):
        Capacity of the hotel (in rooms).

```



```

forecasted_demand (float):
    Forecasted demand (in rooms) for that night

elasticity (float):
    slope of the

nominal_price (float):
    The price for which the forecasted_demand was computed.

Returns
-----
Returns an array of excess capacity.

"""
return capacity - demand_price_elasticity(p_t, nominal_demand=forecasted_demand,
                                           elasticity=elasticity,
                                           nominal_price=nominal_price)

# Let's run the optimization algorithm over four overlapping segments
# of 20, 40, 60, 80 room capacity.

# We look at four capacity segments: 20, 40, 60, and 80 (full capacity)
# rooms available.
capacities = [20.0, 40.0, 60.0, 80.0]

optimization_results = {}
for capacity in capacities:

    # Nominal price associated with forecasted demand:
    nominal_price = 1200.0
    # Forecasted demand:
    nominal_demand = forecasted_demand['occupancy'].values
    # Assumed price elasticity:
    elasticity = -2.0

    # Starting values:
    p_start = 1250.0 * np.ones(len(nominal_demand))

    # bounds on the prices. Let's stick with reasonable values.
    # One could be more sophisticated here and apply constraints
    # that limit the prices to be in range of what competitors
    # are charging, for example.
    bounds = tuple((100.0, 2000.0) for p in p_start)

    # Constraints:

```

```

constraints = ({'type': 'ineq', 'fun': lambda x: constraint_1(x)},
               {'type': 'ineq', 'fun': lambda x, capacity=capacity,
                                     forecasted_demand=nominal_demand,
                                     elasticity=elasticity,
                                     nominal_price=nominal_price: constraint_2(x, capacity,
                                     forecasted_demand, elasticity, nominal_price)

opt_results = optimize.minimize(objective, p_start, args=(nominal_demand,
                                                         elasticity,
                                                         nominal_price),
                                method='SLSQP', bounds=bounds,
                                constraints=constraints)

optimization_results[capacity] = opt_results

# Plotting the resulting rates vs dates.

time_array = np.linspace(1,len(nominal_demand),len(nominal_demand))
rate_df = pd.DataFrame(index=time_array)

for capacity in optimization_results.keys():
    rate_df = pd.concat([rate_df,
                        pd.DataFrame(optimization_results[capacity]['x'],
                                    columns=['{}'.format(capacity)],
                                    index=time_array)],
                        axis=1)

rate_df.index.name = 'Day'
datelist = pd.date_range(start=date_start, end=date_end, closed='left').tolist()
rate_df.index = [ x.date() for x in datelist]

# Save rate dataframe to local folder:

rate_df.to_csv('data/rates.csv')

```

7 Referencias