Índice general

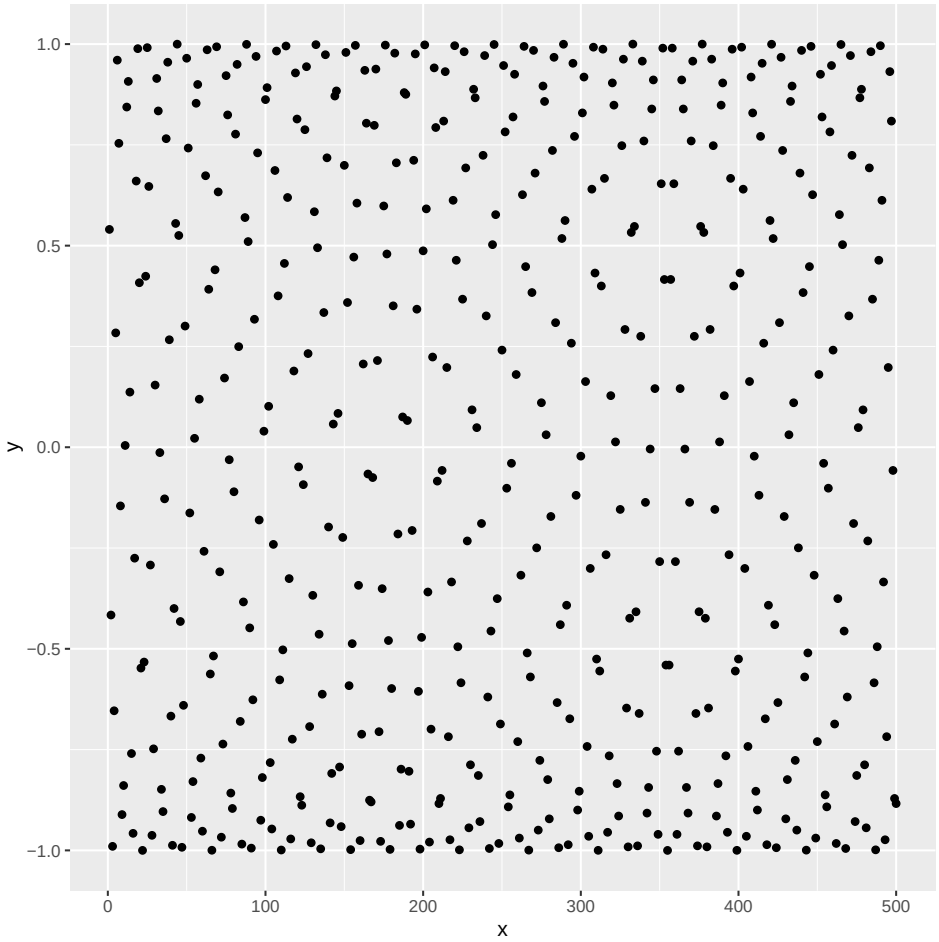


Figura 1: prueba

Capítulo 1

Introducción

Capítulo 2

Objetivos

Capítulo 3

Desarrollo

```
list.files("../scripts")

## [1] "05_Desarrollo.R" "prueba.R"

read_chunk("../scripts/05_Desarrollo.R")
```

Introducción

3.1. Matemáticas

Aquí Mates

3.2. Ingeniería Informática

Una vez se dispone de todas las herramientas matemáticas necesarias, se puede comenzar con el desarrollo de la parte de Ingeniería Informática que se aborda en el trabajo.

3.2.1. Comprensión del problema y de los datos

En primer lugar en cualquier problema de Ciencia de Datos, el primer paso es *comprender el problema y los datos que se disponen.*

El problema a abordar es el siguiente: a partir de un conjunto de partidas antiguas de StarCraft [1], se busca predecir el momento en el que la partida está decidida con una confianza determinada. Para ello, partimos de 6 bases de datos relacionales (SQL) con gran cantidad de partidas almacenadas, cada una con muchas características a observar. Cada una de ellas posee las características presentes en la figura 3.1.

Una vez se tiene conocimiento del problema y un conjunto de datos, se debe decidir qué datos y características van a ser usados y de qué forma. El

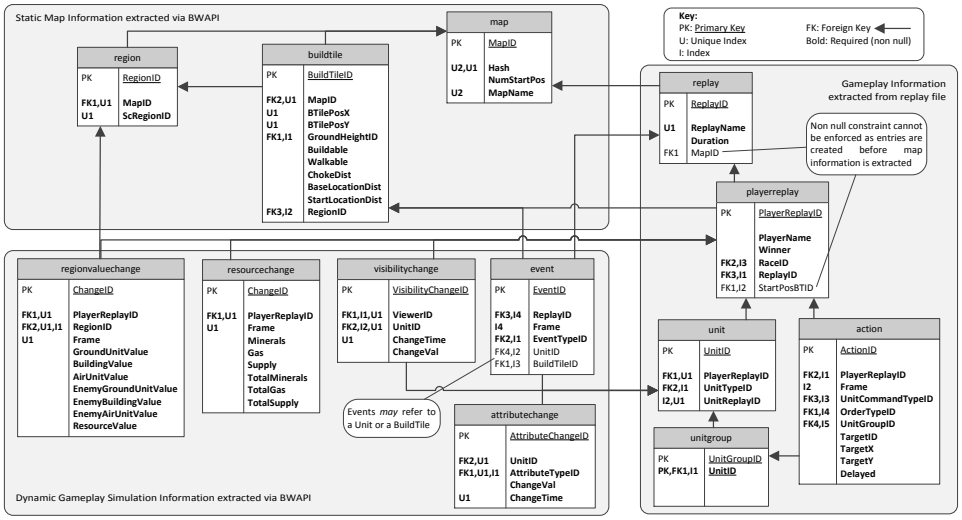


Figura 3.1: Base de datos de partidas de StarCraft

principal problema de este paso es conocer el conjunto de datos del que se dispone, ya que usualmente no es extraído por los investigadores.

Las características están sacadas casi en su totalidad directamente de valores que proporciona la API que permite interactuar con StarCraft, *BWAPI*. Otros son datos derivados, como la distancia de un jugador a la base más cercana, por ejemplo.

Las características que van a ser usados en este trabajo son, principalmente, los recursos de cada jugador, sus batallones (que son medidos de una manera determinada que se explican con más detalle más adelante), sus construcciones, y los valores estimados de batallones y contrucciones que tienen un jugador del otro. Además, también se tiene en cuenta los recursos restantes del mapa que cada jugador estima que quedan. Estas características quedan reflejadas en la figura 3.2.

Estas características son, según cada tabla:

- replay: Esta tabla contiene datos asociados a cada partida.
  - ReplayID: Identificador de cada partida.
  - Duration: Duración (en frames) de cada partida. 15 frames equivalen a 1 segundo.
- playerreplay: Esta tabla contiene datos asociados a un jugador en una partida.
  - PlayerReplayID: Identificador de un jugador en una partida.
  - ReplayID: Identificador de partida asociado.
  - Winner: Ganador de cada partida.

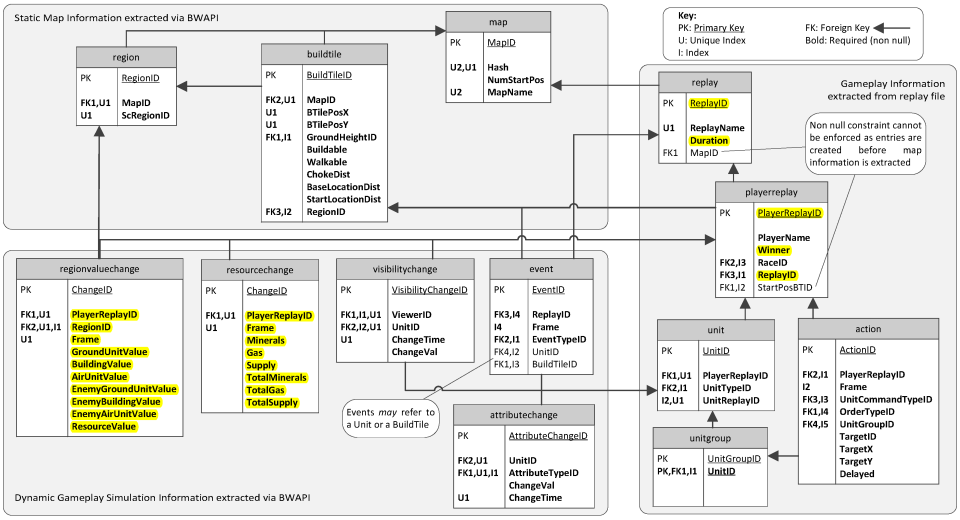


Figura 3.2: Características seleccionadas de las bases de datos

- resourcechange: Esta tabla contiene datos asociados a cambios en los recursos de un jugador.
  - PlayerReplayID: Identificador del jugador que produce un cambio.
  - Frame: Frame en el que se produce un cambio.
  - Minerals: Cantidad de minerales que tiene un jugador en ese momento.
  - Gas: Cantidad de gas que tiene un jugador en ese momento.
  - Supply: Capacidad de carga del jugador.
  - TotalMinerals: Cantidad total de minerales que ha obtenido un jugador, sin contar gastos.
  - TotalGas: Cantidad total de gas que ha obtenido un jugador, sin contar gastos.
  - TotalSupply: Capacidad que ha obtenido un jugador, sin contar gastos.
- regionvaluechange: Esta tabla contiene datos asociados a cambios de un jugador en una región del mapa determinada. Cada *value*, que llamaremos de aquí en adelante *valor*, es la suma del precio de una unidad en Minerales y Gas.

- PlayerReplayID: Identificador del jugador que produce un cambio.
- RegionID: Identificador de la región del mapa donde se produce un cambio.
- Frame: Frame en el que se produce el cambio.
- GroundUnitValue: Valor de las unidades terrestres en esta región.
- BuildingValue: Valor de las construcciones en esta región.
- AirUnitValue: Valor de las unidades aéreas en esta región.
- EnemyGroundUnitValue: Valor de las unidades terrestres del enemigo en esta región. Este valor es estimado, sólo se conoce lo que el jugador puede ver del enemigo.
- EnemyBuildingValue: Valor de las contrucciones del enemigo en esta región. Este valor es estimado, sólo se conoce lo que el jugador puede ver del enemigo.
- EnemyAirUnitValue: Valor de las unidades aéreas del enemigo en esta región. Este valor es estimado, sólo se conoce lo que el jugador puede ver del enemigo.
- ResourceValue: Valor de los recursos en esta región. Este valor es estimado, sólo se conoce lo que el jugador puede ver del mapa. Si el jugador no conoce una zona, estima que los recursos restantes es la totalidad de lo disponible en la región.

Una vez se ha decidido qué vamos a usar, hay que pasar al cómo. Esta fase es el *preprocesamiento de los datos*, que es la fase donde se organizan los datos, se corrigen si hubiera datos perdidos o ruido... para poder abordar el problema a resolver.

3.2.2. Preprocesamiento de datos

Análisis exploratorio de datos

Lectura de datos

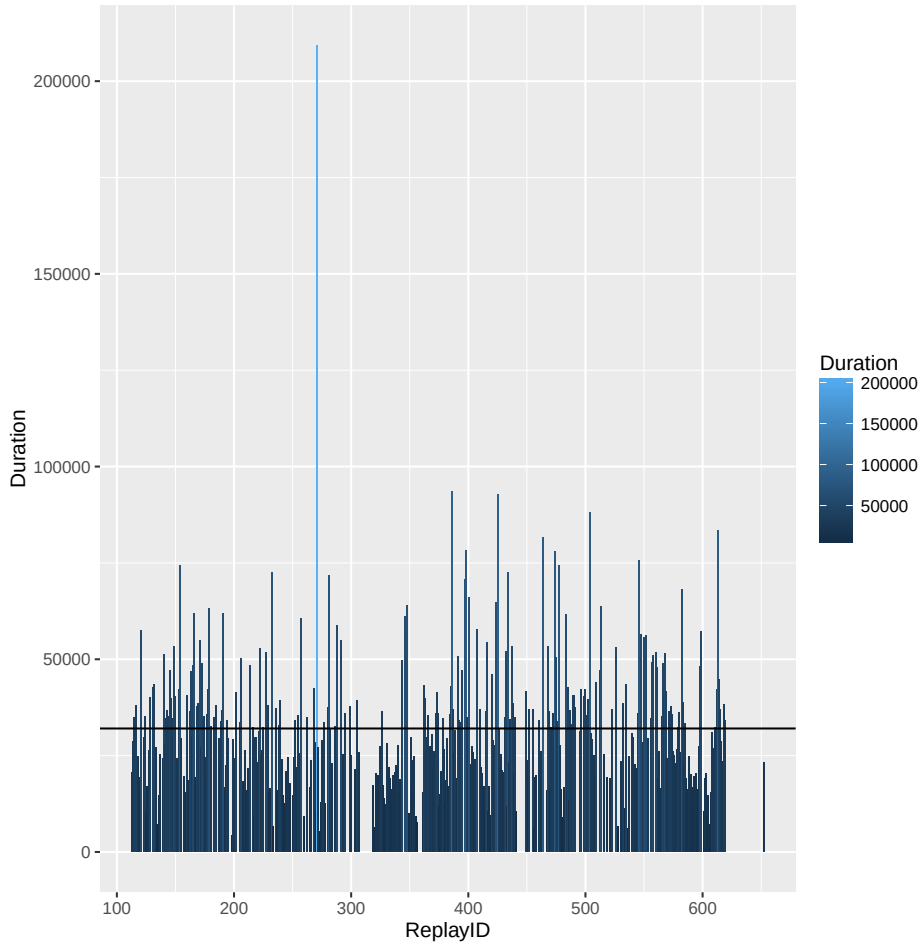
```
print(getwd())

## [1] "/home/antonio/Git/TFG/Memoria/capitulos"

data <- read.csv("../datos/data_pvp.csv")
metadata <- unique(data[, colnames(data) %in% c("ReplayID","Duration")])
# clean_data <- data[,!(colnames(data) %in% c("ReplayID","Duration"))]
# data_replay <- data[data$ReplayID==114, ]
# data_replay <- data_replay[, !(colnames(data_replay) %in% c("ReplayID","Durat
# winner <- data_replay[1,"Winner"]
# loser <- if(winner=='A') 'B' else 'A'
```

Histograma

```
ggplot(data=metadata) +
geom_bar(aes(x=ReplayID,y=Duration,fill=Duration), stat="identity") +
geom_hline(yintercept = mean(metadata$Duration))
```

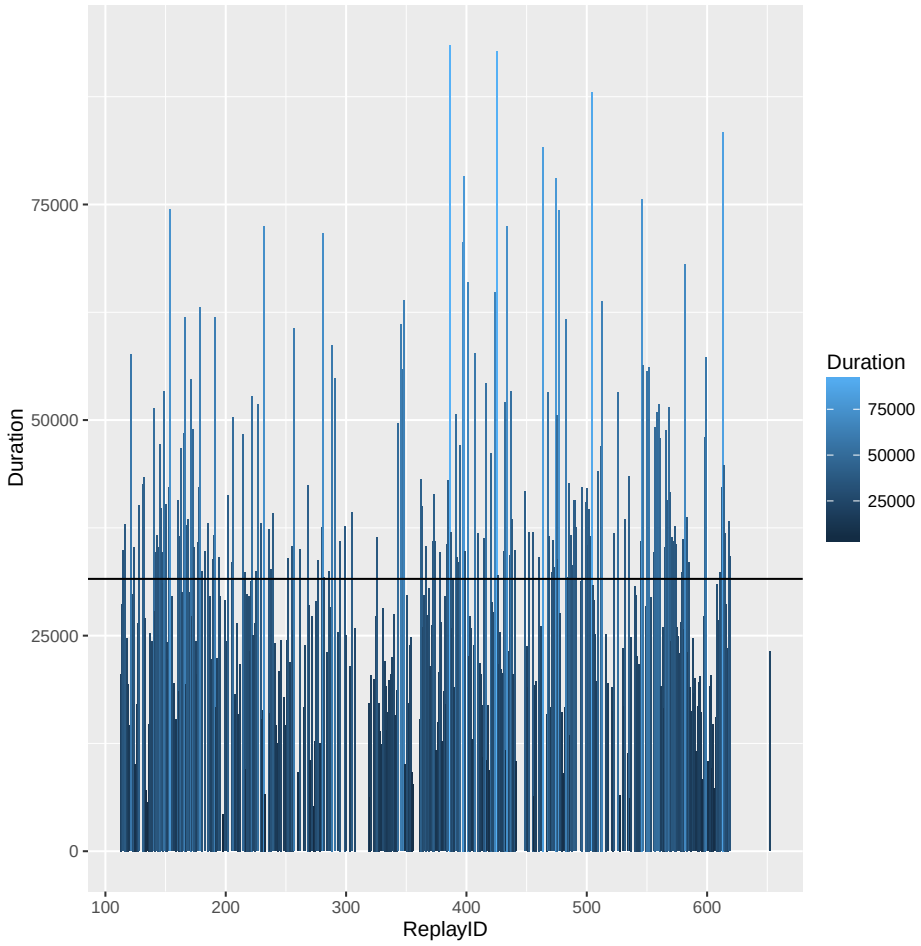



```
print(mean(metadata$Duration))

## [1] 32036.08
```

Quitar partida "ruido"

```
clean_data <- data[data$ReplayID != 271,]
clean_metadata <- unique(clean_data[, colnames(data) %in% c("ReplayID","Duration")])
ggplot(data=clean_metadata) +
geom_bar(aes(x=ReplayID,y=Duration,fill=Duration), stat="identity") +
geom_hline(yintercept = mean(clean_metadata$Duration))
```




```
print(mean(clean_metadata$Duration))
```

```
## [1] 31582.96
```

Capítulo 4

Conclusiones

Bibliografía

[1] Glen Robertson and Ian Watson. An improved dataset and extraction process for starcraft ai. 2014. Proceedings of the Florida Artificial Intelligence Research Society Conference (FLAIRS).