Minería de datos: PRA2 - Modelado de un juego de datos

Autor: Gabriel Patricio Bonilla Sanchez

Enero 2021

Contents

Introducción

Presentación

Esta práctica cubre de forma transversal la asignatura.

Las Prácticas 1 y 2 de la asignatura se plantean de una forma conjunta de modo que la Práctica 2 será continuación de la 1.

El objetivo global de las dos prácticas consiste en seleccionar uno o varios juegos de datos, realizar las tareas de preparación y análisis exploratorio con el objetivo de disponer de datos listos para aplicar algoritmos de clustering, asociación y clasificación.

Competencias

Las competencias que se trabajan en esta prueba son:

- Uso y aplicación de las TIC en el ámbito académico y profesional.
- Capacidad para innovar y generar nuevas ideas.
- Capacidad para evaluar soluciones tecnológicas y elaborar propuestas de proyectos teniendo en cuenta los recursos, las alternativas disponibles y las condiciones de mercado.
- Conocer las tecnologías de comunicaciones actuales y emergentes así como saberlas aplicar convenientemente para diseñar y desarrollar soluciones basadas en sistemas y tecnologías de la información.
- Aplicación de las técnicas específicas de ingeniería del software en las diferentes etapas del ciclo de vida de un proyecto.
- Capacidad para aplicar las técnicas específicas de tratamiento, almacenamiento y administración de datos.
- Capacidad para proponer y evaluar diferentes alternativas tecnológicas para resolver un problema concreto.

Objetivos

La correcta asimilación de todos los aspectos trabajados durante el semestre.
En esta práctica abordamos un caso real de minería de datos donde tenemos que poner en juego todos los conceptos trabajados. Hay que trabajar todo el ciclo de vida del proyecto. Desde el objetivo del proyecto hasta la implementación del conocimiento encontrado pasando por la preparación, limpieza de los datos, conocimiento de los datos, generación del modelo, interpretación y evaluación.

Descripción de la PEC a realizar

Recursos Básicos

Material docente proporcionado por la UOC.

Criterios de valoración

Ejercicios prácticos

Para todas las PEC es necesario documentar en cada apartado del ejercicio práctico que se ha hecho y como se ha hecho.

Formato y fecha de entrega

El formato de entrega es: usernameestudiante-PECn.html/doc/docx/odt/pdf/rmd
Fecha de entrega: 15/01/2020
Se debe entregar la PEC en el buzón de entregas del aula

Nota: Propiedad intelectual

> A menudo es inevitable, al producir una obra multimedia, hacer uso de recursos creados por terceras personas. Es por lo tanto comprensible hacerlo en el marco de una práctica de los estudios de Informática, Multimedia y Telecomunicación de la UOC, siempre y cuando esto se documente claramente y no suponga plagio en la práctica.
>
> Por lo tanto, al presentar una práctica que haga uso de recursos ajenos, se debe presentar junto con ella un documento en que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y su estatus legal: si la obra esta protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL ...). El estudiante deberá asegurarse de que la licencia no impide específicamente su uso en el marco de la práctica. En caso de no encontrar la información correspondiente tendrá que asumir que la obra esta protegida por copyright.
>
> Deberéis, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente si corresponde.

Resumen Práctica 1 (PRA 1)

En este apartado resummiremos los pasos necesarios para la preparación del dataset final realizado en la PRA 1:

El dataset seleccionado ha sido obtenido desde el siguiente enlace: https://www.kaggle.com/aitzaz/stack-overflow-developer-survey-2020. Este juego de datos contiene los resultados de la Encuesta Anual a Desarrolladores StackOverflow 2020. Se obtuvo alrededor de 65000 participaciones de programadores y desarrolladores de 180 paises. La encuesta aborda varios ámbitos, tanto a nivel de experiencia, formación académica y skills (habilidades técnicas) en diferentes tecnologías que el encuestado ha ido adquiriendo a lo largo del tiempo.

Esta encuesta anual ha recolectado datos sobre 61 variables que se pasan a detallar a continuación:

- *Respondent:* número de identificación del encuestado aleatorizado (no en orden de tiempo de respuesta de la encuesta)

- *MainBranch:* ¿Cuál de las siguientes opciones te describe mejor hoy?
- *Hobbyist:* ¿Desarrollas como pasatiempo?
- *Age:* ¿Cuál es su edad (en años)?
- *Age1stCode:* ¿A qué edad escribiste tu primera línea de código o programa?
- *CompFreq:* ¿Esa compensación es semanal, mensual o anual?
- *CompTotal:* ¿Cuál es su compensación total actual (salario, bonificaciones y beneficios, antes de impuestos y deducciones), en "CurrencySymbol"?. Número entero.
- *ConvertedComp:* Salario anual en USD, utilizando el tipo de cambio del 19 de febrero de 2020, asumiendo 12 meses laborales y 50 semanas laborales".
- *Country:* País dónde vive.
- *CurrencyDesc:* ¿Qué moneda utiliza a diario? Descripción.
- *CurrencySymbol:* ¿Qué moneda usa a diario? Forma abreviada.
- *DatabaseDesireNextYear:* ¿En qué entornos de base de datos desea trabajar durante el próximo año?
- *DatabaseWorkedWith:* ¿En qué entornos de base de datos ha realizado un trabajo de desarrollo extenso durante el año pasado?
- *DevType:* ¿Cuál de los siguientes lo describe?
- *EdLevel:* ¿Cuál de las siguientes opciones describe mejor el nivel más alto de educación formal que ha completado?
- *Employment:* ¿cuál de las siguientes opciones describe mejor su situación laboral actual?
- *Ethnicity:* ¿Cuál de los siguientes grupos étnicos lo describe?
- *Gender:* ¿Cuál de las siguientes opciones de sexo lo describe?
- *JobFactors:* Para el caso de decidiendo entre dos ofertas de trabajo con la misma compensación, beneficios y ubicación. ¿Qué factores son los más importantes para usted?
- *JobSat:* ¿Qué tan satisfecho está con su trabajo actual?
- *JobSeek:* ¿Cuál de las siguientes opciones describe mejor su estado actual de búsqueda de empleo?
- *LanguageDesireNextYear:* "¿En qué lenguajes de programación, scripting y marcado desea trabajar durante el próximo año?.
- *LanguageWorkedWith:* ¿En qué lenguajes de programación, scripting y marcado ha realizado un trabajo de desarrollo extenso durante el año pasado?.
- *MiscTechDesireNextYear:* ¿En qué otros frameworks, bibliotecas y herramientas desea trabajar durante el próximo año?.
- *MiscTechWorkedWith:* ¿En qué otros frameworks, bibliotecas y herramientas ha realizado un trabajo de desarrollo extenso durante el año pasado?.
- *NEWCollabToolsDesireNextYear:* ¿En qué herramientas de colaboración desea trabajar durante el próximo año?
- *NEWCollabToolsWorkedWith:* ¿En qué herramientas de colaboración ha realizado un trabajo de desarrollo extenso durante el año pasado?
- *NEWDevOps:* ¿Su empresa tiene una persona dedicada a DevOps?
- *NEWDevOpsImpt:* ¿Qué importancia tiene la práctica de DevOps para escalar el desarrollo de software?
- *NEWEdImpt:* ¿Qué importancia tiene una educación formal, como un título universitario en ciencias de la computación, para su carrera?
- *NEWJobHunt:* En general, ¿Cuáles son las motivaciones que lo impulsan a buscar un nuevo trabajo?.
- *NEWJobHuntResearch:* Cuando busca trabajo, ¿cómo puede obtener más información sobre una empresa?
- *NEWLearn:* ¿Con qué frecuencia aprende un nuevo lenguaje o marco?
- *NEWOffTopic:* ¿Crees que Stack Overflow debería relajar las restricciones sobre lo que se considera "fuera de tema"?
- *NEWOnboardGood:* ¿Cree que su empresa tiene un buen proceso de incorporación? (Por incorporación, nos referimos al proceso estructurado para que se adapte a su nuevo puesto en una empresa)
- *NEWOtherComms:* ¿Es miembro de alguna otra comunidad de desarrolladores en línea?
- *NEWOvertime:* ¿Con qué frecuencia trabaja horas extraordinarias o más allá de las expectativas formales de su trabajo?
- *NEWPurchaseResearch:* Al comprar una nueva herramienta o software, ¿cómo descubre e investiga las

soluciones disponibles?
- *NEWPurpleLink:* Busca una solución de codificación en línea y el primer enlace de resultado es violeta porque ya lo visitó. ¿Cómo se siente?
- *NEWSOSites:* ¿Cuál de los siguientes sitios de Stack Overflow ha visitado?
- *NEWStuck:* ¿Qué hace cuando se queda atascado en un problema?
- *OpSys:* ¿Cuál es el sistema operativo principal en el que trabaja?
- *OrgSize:* Aproximadamente, ¿cuántas personas emplea la empresa u organización para la que trabaja actualmente?
- *PlatformDesireNextYear:* ¿En qué plataformas desea trabajar durante el próximo año?
- *PlatformWorkedWith:* ¿En qué plataformas ha realizado un trabajo de desarrollo extenso durante el año pasado?
- *PurchaseWhat:* ¿Qué nivel de influencia tiene usted, personalmente, sobre las compras de nueva tecnología en su organización?
- *Sexuality:* ¿Cuál de los siguientes lo describe a usted sobre su sexualidad?.
- *SOAccount:* ¿Tiene una cuenta de Stack Overflow?
- *SOComm:* ¿Te consideras miembro de la comunidad de Stack Overflow?
- *SOPartFreq:* ¿Con qué frecuencia diría que participa en preguntas y respuestas en Stack Overflow? Por participar nos referimos a preguntar, responder, votar o comentar preguntas.
- *SOVisitFreq:* ¿Con qué frecuencia visita Stack Overflow?
- *SurveyEase:* ¿Qué tan fácil o difícil fue completar esta encuesta?
- *SurveyLength:* ¿Qué opina de la duración de la encuesta este año?
- *Trans:* ¿Eres transgénero?
- *UndergradMajor:* ¿Cuál fue su campo de estudio principal?
- *WebframeDesireNextYear:* ¿En qué frameworks web desea trabajar durante el próximo año?
- *WebframeWorkedWith:* ¿En qué frameworks web ha realizado un extenso trabajo de desarrollo durante el año pasado?
- *WelcomeChange:* En comparación con el año pasado, ¿qué tan bienvenido se siente en Stack Overflow?
- *WorkWeekHrs:* En promedio, ¿cuántas horas por semana trabaja?
- *YearsCode:* Incluyendo cualquier educación, ¿cuántos años ha estado programando en total?
- *YearsCodePro:* NO incluye educación, ¿cuántos años ha programado profesionalmente (como parte de su trabajo)?

Las capacidades análiticas del dataset, que se tomaron en cuenta para elegirlo son:

- Cuenta con una cantidad suficientes variables, tanto numéricas, categóricas. Las variables categóricas también pueden volverse a convertir a variables numéricas. Esto permitiría aplicar algoritmos supervisados y no supervisados, donde se puede clasificar a los programadores o desarrolladores según la experticia actual.

- También permite agregar nuevas variables númericas que representen el número de tecnologías que domina cada encuestado.

- Al incluir las tecnologías usadas por desarrolladores en: base de datos, lenguages de programación, frameworks y demás herramientas, permite tener una gran cantidad de preferencias de las que se puede extraer reglas de asociación interesantes sobre las tecnologías más usadas entre los distintos tipos de desarrolladores.

- Cuenta con variables que pueden discretizarse y otras donde se puede aplicar tareas de limpieza y preparación previa antes de aplicar los distintos métodos.

Sin embargo, para efectos del análisis, del dataset original, se excluirán las siguientes variables:

1. Respondent
2. MainBranch

3. Hobbyist
4. Age1stCode
5. CompFreq
6. CompTotal (+)
7. CurrencyDesc
8. CurrencySymbol
9. DatabaseDesireNextYear
10. Ethnicity
11. JobFactors
12. JobSat
13. JobSeek
14. LanguageDesireNextYear
15. MiscTechDesireNextYear
16. NEWCollabToolsDesireNextYear
17. NEWDevOps
18. NEWDevOpsImpt
19. NEWEdImpt
20. NEWJobHunt
21. NEWJobHuntResearch
22. NEWLearn
23. NEWOffTopic
24. NEWOnboardGood
25. NEWOtherComms
26. NEWOvertime
27. NEWPurchaseResearch
28. NEWPurpleLink
29. NEWSOSites
30. NEWStuck
31. PlatformDesireNextYear
32. PurchaseWhat
33. Sexuality
34. SOComm
35. SOVisitFreq (+)
36. SurveyEase
37. SurveyLength
38. Trans
39. UndergradMajor (+)
40. WebframeDesireNextYear
41. WelcomeChange
42. YearsCodePro (+)

Muchos de estos campos no son relevantes para el alcance de la Práctica #1 y #2; otros reflejan deseos de los programadores respecto a tecnologías, para lo cual solo tomaremos los datos que reflejan la experiencia actual del programador.

Los campos marcados con (+) se los ha excluído, ya que se existe otra variable similar, que en caso de mantenerse significaría agregar información redundante al dataset.

Con la finalidad de disminuir el número de observaciones o individuos, vamos a limitar el estudio de este dataset a mi país natal, *Ecuador*. Con esto no incluiremos la variable **Country**.

En conclusión, vamos a trabajar a con 18 variables propias del dataset original, de las cuales 4 son númericas (Age, ConvertedComp, WorkWeekHrs y YearsCode). También tenemos variables no numéricas, las cuales vamos a realizar un análisis más detallado posteriormente, generando variables númericas a partir de ellas, las cuales son:

- DatabaseWorkedWith
- LanguageWorkedWith
- MiscTechWorkedWith
- NEWCollabToolsWorkedWith
- PlatformWorkedWith
- WebframeWorkedWith

Estas nuevas variables numéricas a generarse posteriormente servirán principalmente cuando se intente aplicar algoritmos no supervisados, como K-Means, y también serán usadas para crear un nuevo dataset sobre el que se aplicará el algoritmo SVD o PCA.

Este nuevo dataset de variables numéricas, ayudará a dar respuesta a las siguientes preguntas:

> ¿Hay relación directa entre el número de tecnologías que domina el programador y su sueldo anual, en Ecuador? ¿Hay relación directa entre el número de años de experiencia que tiene el programador y el número de tecnologías que domina, en Ecuador? ¿En Ecuador, influye el número de años de experiencia del programador con el número de tecnologías que domina o conoce? ¿Qué relación hay entre el número de años de experiencia del programador y el sueldo que percibe anualmente en el mercado ecuatoriano? ¿Como afecta el número de horas trabajadas a la semana sobre el sueldo que percibe anualmente el programador ecuatoriano? ¿Hay relación directa entre el número de años de experiencia que tiene el programador y la edad del mismo en Ecuador?

Análisis exploratorio del juego de datos seleccionado.

- Cargar el dataset

```
# Cargamos los paquetes R que vamos a usar
library(ggplot2)
library(dplyr)
library(car)

options('max.print' = 100000)   # or whatever value you want

# Cargamos el fichero de datos_original
datos_original <- read.csv('survey_results_public.csv', sep=",", encoding = "UTF-8")
filas_original=dim(datos_original)[1]

# Creamos un juego de datos resumido
datos <- datos_original[, c(4, 8:9, 13:16, 18, 23, 25, 27, 42:43, 45, 48, 50, 57, 59:60)]
filas=dim(datos)[1]

# Filtramos solo los registros que corresponden con Ecuador
datosEcuador <- datos[datos$Country %in% c("Ecuador"), ]
filasEcuador=dim(datosEcuador)[1]

# Anulamos la variable Country del dataset datosEcuador
datosEcuador$Country = NULL

# Rellanamos los campos con NA con valores medios de cada variable
datosEcuador$Age[is.na(datosEcuador$Age)] <- mean(datosEcuador$Age,na.rm=T)

datosEcuador$ConvertedComp[is.na(datosEcuador$ConvertedComp)] <- mean(datosEcuador$ConvertedComp,na.rm=
```

```
datosEcuador$WorkWeekHrs[is.na(datosEcuador$WorkWeekHrs)] <- mean(datosEcuador$WorkWeekHrs,na.rm=T)

# Convertimos los valores categoricos en numéricos para la variable YearsCode
datosEcuador$YearsCode[datosEcuador$YearsCode=="More than 50 years"] <- 50
datosEcuador$YearsCode[datosEcuador$YearsCode=="Less than 1 year"] <- 1

# Finalmente convertimos dicha columna en número
datosEcuador$YearsCode <- as.numeric(datosEcuador$YearsCode)

# Llenamos con la media los valores faltantes
datosEcuador$YearsCode[is.na(datosEcuador$YearsCode)] <- mean(datosEcuador$YearsCode, na.rm=T)

# Definimos la variable **techs**. Las columnas a concatenar son: *DatabaseWorkedWith*, *LanguageWorked
datosEcuador$techs <- paste(datosEcuador$DatabaseWorkedWith, ";", datosEcuador$LanguageWorkedWith, ";",
```

Ahora vamos a agregar nuevas variables que contabilizan el número de tecnologías o herramientas de: bases de datos, lenguajes de programación, de colaboración, entre otros. Primero para la base de datos, vamos a usar la columna *DatabaseWorkedWith*. La variable a crearse será **db_techs**:

```
datosEcuador$db_techs <- 0

for(i in 1:filasEcuador) {
  if (is.na(datosEcuador$DatabaseWorkedWith[i])) {
    datosEcuador$db_techs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(datosEcuador$DatabaseWorkedWith[i], ";"), length)
    datosEcuador$db_techs[i] <- longitud
  }
}
```

Ahora vamos a agregar una nueva variable para el número de carreras u oficios que tiene el encuestado. Para esto vamos a usar la columna *DevType*, ya que la misma es una concatenación de todas las opciones que seleccionó el participante durante la encuesta. La variable a crearse será **num_types**:

```
datosEcuador$num_types <- 0

for(i in 1:filasEcuador) {
  if (is.na(datosEcuador$DevType[i])) {
    datosEcuador$num_types[i] <- 0
  } else {
    longitud  <- sapply(strsplit(datosEcuador$DevType[i], ";"), length)
    datosEcuador$num_types[i] <- longitud
  }
}
```

Para agregar una nueva variable que represente el número de lenguajes de programación que usa. Este dato se basa en la experiencia ya adquirida y no en los deseos para usar o aprender el siguiente año. Para esto usaremos la columna *LanguageWorkedWith*. La variable a crearse será **prog_langs**:

```
datosEcuador$prog_langs <- 0

for(i in 1:filasEcuador) {
```

```
  if (is.na(datosEcuador$LanguageWorkedWith[i])) {
    datosEcuador$prog_langs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(datosEcuador$LanguageWorkedWith[i], ";"), length)
    datosEcuador$prog_langs[i] <- longitud
  }
}
```

Ahora vamos a agregar una nueva variable para el número de frameworks, librerías y demás herramientras que usa el desarrollador. Este dato se basa en la experiencia ya adquirida y no en los deseos para usar o aprender el siguiente año. Para esto usaremos la columna *MiscTechWorkedWith*. La variable a crearse será **misc_techs**:

```
datosEcuador$misc_techs <- 0

for(i in 1:filasEcuador) {
  if (is.na(datosEcuador$MiscTechWorkedWith[i])) {
    datosEcuador$misc_techs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(datosEcuador$MiscTechWorkedWith[i], ";"), length)
    datosEcuador$misc_techs[i] <- longitud
  }
}
```

Haremos lo mismo para el número de herramientras colaborativas que usa el desarrollador, según el contenido de la columna *NEWCollabToolsWorkedWith*. Este dato se basa en la experiencia ya adquirida y no en los deseos para usar o aprender el siguiente año. La variable a crearse será **collab_techs**:

```
datosEcuador$collab_techs <- 0

for(i in 1:filasEcuador) {
  if (is.na(datosEcuador$NEWCollabToolsWorkedWith[i])) {
    datosEcuador$collab_techs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(datosEcuador$NEWCollabToolsWorkedWith[i], ";"), length)
    datosEcuador$collab_techs[i] <- longitud
  }
}
```

También vamos a agregar una variable para el número de plataformas que usa el desarrollador. Este dato se basa en la experiencia ya adquirida y no en los deseos para usar o aprender el siguiente año. Usaremos el contenido de la columna *PlatformWorkedWith*. La variable a crearse será **plat_techs**:

```
datosEcuador$plat_techs <- 0

for(i in 1:filasEcuador) {
  if (is.na(datosEcuador$PlatformWorkedWith[i])) {
    datosEcuador$plat_techs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(datosEcuador$PlatformWorkedWith[i], ";"), length)
    datosEcuador$plat_techs[i] <- longitud
  }
}
```

Finalmente, agregaremos una variable para el número de *frameworks web* que usa el desarrollador. Este dato se basa en la experiencia ya adquirida y no en los deseos para usar o aprender el siguiente año. Para esto usaremos la columna *WebframeWorkedWith*. La variable a crearse será **web_techs**:

```
datosEcuador$web_techs <- 0

for(i in 1:filasEcuador) {
  if (is.na(datosEcuador$WebframeWorkedWith[i])) {
    datosEcuador$web_techs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(datosEcuador$WebframeWorkedWith[i], ";"), length)
    datosEcuador$web_techs[i] <- longitud
  }
}
```

Por último se aplicó discretización sobre los campos: Age, ConvertedComp, WorkWeekHrs y YearsCode

```
# Discretizamos para la variable Age
datosEcuador$segmento_edad <- cut(datosEcuador$Age, breaks = c(0,10,20,30,40,50,60,70,100), labels = c(

# Discretizamos para la variable ConvertedComp
datosEcuador$segmento_salario <- cut(datosEcuador$ConvertedComp, breaks = c(0,25000,50000,75000,100000,

# Discretizamos para la variable WorkWeekHrs
datosEcuador$segmento_horas_trab <- cut(datosEcuador$WorkWeekHrs, breaks = c(0,20,40,60,80,168), labels

# Discretizamos para la variable YearsCode
datosEcuador$segmento_years_code <- cut(datosEcuador$YearsCode, breaks = c(0,5,10,15,20,30,40,50), label
```

> **NOTA**: El dataset preparado previamente estaba pensado para aplicar reglas de asociación, mediante el algoritmo "a priori" y PCA/SVD, por lo que no ayudará mucho en la aplicación de todos los algoritmos de módelos supervisados o no supervisados. Se realizará la preparación de datos correspondientes dependiendo del modelo en los enunciados siguientes.

Enunciado

Como continuación del estudio iniciado en la práctica 1, procedemos en esta práctica 2 a aplicar modelos analíticos sobe el juego de datos seleccionado y preparado en la práctica anterior.

Antes de proceder a resolver los enunciados vamos a limitar el alcance de los encuestados. En la PRA 1 se limitó a través de la variable *Country* filtrando los registros del país Ecuador. Para efectos de la continuacón con la PRA 2 vamos a ampliar este dataset para que contenga los registros de todos los encuestados en los países de Sudamérica: Colombia, Venezuela, Ecuador, Perú, Chile, Uruguay, Brazil, Argentina, Bolivia y Paraguay

```
dataSudamerica <- datos_original[datos_original$Country %in% c("Argentina", "Bolivia", "Chile", "Colombi

head(dataSudamerica)
```

```
##     Respondent                    MainBranch Hobbyist Age Age1stCode CompFreq
## 44          44 I am a developer by profession       No  32         21   Yearly
## 59          59 I am a developer by profession       No  38         15  Monthly
## 189        190 I am a developer by profession      Yes  NA         20  Monthly
## 240        241 I am a developer by profession      Yes  28         23  Monthly
## 290        291    I code primarily as a hobby      Yes  18         13     <NA>
## 309        310 I am a developer by profession      Yes  27         18  Monthly
##     CompTotal ConvertedComp Country    CurrencyDesc CurrencySymbol
## 44     244000         55893  Brazil  Brazilian real            BRL
## 59       6000         16488  Brazil  Brazilian real            BRL
## 189        NA            NA Uruguay  Uruguayan peso            UYU
## 240      3000          8244  Brazil  Brazilian real            BRL
## 290        NA            NA   Chile            <NA>           <NA>
## 309       960          8712  Brazil Canadian dollar            CAD
##
## 44
## 59
## 189                                                                    Elasticsearch;Firebase;M
## 240
## 290
## 309 Cassandra;Couchbase;DynamoDB;Elasticsearch;Firebase;IBM DB2;MariaDB;Microsoft SQL Server;MongoDB
##                                          DatabaseWorkedWith
## 44                                     Microsoft SQL Server
## 59                 Microsoft SQL Server;MySQL;PostgreSQL;SQLite
## 189                                     MySQL;Oracle;SQLite
## 240                                                    <NA>
## 290                                    MongoDB;MySQL;SQLite
## 309 DynamoDB;Firebase;MariaDB;Microsoft SQL Server;MySQL;SQLite
##
## 44                                                                  Data or business analyst;Develop
## 59                 Data or business analyst;Developer, back-end;Developer, front-end;Developer, full
## 189 Developer, back-end;Developer, embedded applications or devices;Developer, front-end;Developer,
## 240
## 290
## 309                 Developer, back-end;Developer, desktop or enterprise applications;Developer, fro
##                                              EdLevel         Employment
## 44         Master's degree (M.A., M.S., M.Eng., MBA, etc.) Employed full-time
## 59            Bachelor's degree (B.A., B.S., B.Eng., etc.) Employed full-time
## 189                    Associate degree (A.A., A.S., etc.) Employed full-time
## 240 Some college/university study without earning a degree Employed full-time
## 290                             Primary/elementary school            Student
## 309 Some college/university study without earning a degree Employed full-time
##                                         Ethnicity Gender
## 44                        White or of European descent    Man
## 59                              Hispanic or Latino/a/x    Man
## 189                       White or of European descent    Man
## 240                       White or of European descent    Man
## 290                             Hispanic or Latino/a/x    Man
## 309 Hispanic or Latino/a/x;White or of European descent    Man
##
## 44                      Industry that I'd be working in;Specific department or team I'd be working
## 59                                                 Industry that I'd be working in;Diversity of th
## 189      Flex time or a flexible schedule;Languages, frameworks, and other technologies I'd be worki
## 240 Flex time or a flexible schedule;Languages, frameworks, and other technologies I'd be working wi
```

```
## 290
## 309                              Flex time or a flexible schedule;Office environment or company cultu
##                                  JobSat
## 44  Neither satisfied nor dissatisfied
## 59                   Slightly satisfied
## 189               Slightly dissatisfied
## 240                  Slightly satisfied
## 290                                <NA>
## 309                  Slightly satisfied
##                                                                JobSeek
## 44  I'm not actively looking, but I am open to new opportunities
## 59                   I am not interested in new job opportunities
## 189 I'm not actively looking, but I am open to new opportunities
## 240 I'm not actively looking, but I am open to new opportunities
## 290                                                             <NA>
## 309                               I am actively looking for a job
##                                                  LanguageDesireNextYear
## 44                            HTML/CSS;Java;JavaScript;Python;R;SQL
## 59                            C#;HTML/CSS;JavaScript;SQL;TypeScript
## 189 Bash/Shell/PowerShell;HTML/CSS;Java;Kotlin;Python;TypeScript
## 240                                      C#;HTML/CSS;JavaScript
## 290                    C;C#;C++;Go;HTML/CSS;Java;JavaScript;Python
## 309           Bash/Shell/PowerShell;C;C#;HTML/CSS;Python;SQL;Swift
##                                                                    LanguageWorkedWith
## 44                                                        HTML/CSS;Python;R;SQL;VBA
## 59                                                  C#;HTML/CSS;JavaScript;PHP;Ruby
## 189                                  Bash/Shell/PowerShell;HTML/CSS;Java;JavaScript
## 240                                                         HTML/CSS;JavaScript
## 290                                                                     C#;Python
## 309 Bash/Shell/PowerShell;C#;Dart;HTML/CSS;JavaScript;Kotlin;Python;Swift;TypeScript
##
## 44
## 59
## 189
## 240
## 290
## 309 .NET Core;Ansible;Apache Spark;Chef;Hadoop;Node.js;Pandas;Puppet;TensorFlow;Teraform;Torch/PyTor
##                                                 MiscTechWorkedWith
## 44                                                             <NA>
## 59                                          .NET;Node.js;Unity 3D
## 189                                              Cordova;Node.js
## 240                                          React Native;Xamarin
## 290                                                .NET;.NET Core
## 309 .NET Core;Cordova;Flutter;Node.js;Pandas;Unity 3D;Unreal Engine
##                                          NEWCollabToolsDesireNextYear
## 44                                                 Github;Slack;Trello
## 59                         Github;Slack;Microsoft Teams;Microsoft Azure
## 189 Confluence;Jira;Github;Gitlab;Slack;Google Suite (Docs, Meet, etc)
## 240                                                              Github
## 290                                                              Github
## 309                        Confluence;Jira;Github;Slack;Microsoft Azure
##                                           NEWCollabToolsWorkedWith
## 44                                             Github;Slack;Trello
## 59                                             Github;Slack;Trello
```

```
## 189 Confluence;Jira;Github;Gitlab;Slack;Google Suite (Docs, Meet, etc)
## 240                                                                 Github
## 290                                                         Github;Gitlab
## 309                                          Github;Slack;Microsoft Azure
##     NEWDevOps       NEWDevOpsImpt            NEWEdImpt
## 44         No  Somewhat important       Very important
## 59         No  Somewhat important     Fairly important
## 189        No Extremely important Critically important
## 240        No  Somewhat important     Fairly important
## 290      <NA>                <NA>                 <NA>
## 309        No Extremely important   Somewhat important
##
## 44  Having a bad day (or week or month) at work;Curious about other opportunities;Trouble with my tea
## 59                                Curious about other opportunities;Better compensation;Trouble with m
## 189
## 240
## 290
## 309                                                Wanting to share accomplishments with a wider network
##
## 44                                                                                 Company reviews from third p
## 59
## 189 Read company media, such as employee blogs or company culture videos;Company reviews from third p
## 240
## 290
## 309                                                                                                    Company
##                 NEWLearn NEWOffTopic               NEWOnboardGood NEWOtherComms
## 44  Once every few years          No                           No            No
## 59      Every few months          No                           No            No
## 189 Once every few years          No                          Yes            No
## 240          Once a year          No Onboarding? What onboarding?            No
## 290 Once every few years    Not sure                         <NA>           Yes
## 309     Every few months    Not sure                           No           Yes
##                           NEWOvertime
## 44   Often: 1-2 days per week or more
## 59                              Never
## 189                             Never
## 240 Rarely: 1-2 days per year or less
## 290                              <NA>
## 309                             Never
##
## 44  Start a free trial;Ask developers I know/work with;Visit developer communities like Stack Overflo
## 59
## 189
## 240
## 290                                                                 Start a free trial;Ask developers I kno
## 309          Start a free trial;Ask developers I know/work with;Research companies that have advertis
##         NEWPurpleLink
## 44  Hello, old friend
## 59        Indifferent
## 189 Hello, old friend
## 240           Annoyed
## 290 Hello, old friend
## 309 Hello, old friend
##
```

```
## 44                                      Stack Overflow (public Q&A for anyone who codes);Stack Excl
## 59                                                                                          Stack
## 189 Stack Overflow (public Q&A for anyone who codes);Stack Exchange (public Q&A for a variety of topi
## 240                                     Stack Overflow (public Q&A for anyone who codes);Stack Excl
## 290                                     Stack Overflow (public Q&A for anyone who codes);Stack Excl
## 309 Stack Overflow (public Q&A for anyone who codes);Stack Exchange (public Q&A for a variety of topi
##
## 44
## 59
## 189
## 240
## 290                                                                                  Play games;Call a cou
## 309 Meditate;Call a coworker or friend;Visit Stack Overflow;Go for a walk or other physical activity
##          OpSys                                          OrgSize
## 44     Windows                                  10 to 19 employees
## 59     Windows                                 500 to 999 employees
## 189 Linux-based                                100 to 499 employees
## 240    Windows                                  10 to 19 employees
## 290    Windows                                              <NA>
## 309    Windows Just me - I am a freelancer, sole proprietor, etc.
##                                                                                      PlatformDesire
## 44                                                                                           Linux
## 59                                                                                  Microsoft Azure
## 189                                          Android;AWS;Google Cloud Platform;Kubernetes;Linux;Rasp
## 240                                                                                        Androi
## 290                                        Android;Arduino;Docker;Heroku;iOS;Linux;MacOS;Raspberry Pi
## 309 Android;Docker;Heroku;IBM Cloud or Watson;iOS;Kubernetes;Linux;MacOS;Microsoft Azure;Raspberry Pi
##                                  PlatformWorkedWith
## 44                                          Windows
## 59                                    Docker;MacOS
## 189                        Android;AWS;Docker;Linux
## 240                              Android;Windows
## 290       Android;Arduino;Docker;Heroku;Linux;Windows
## 309 Android;Linux;Microsoft Azure;Raspberry Pi;Windows
##                        PurchaseWhat              Sexuality SOAccount
## 44            I have some influence Straight / Heterosexual       Yes
## 59    I have little or no influence Straight / Heterosexual       Yes
## 189   I have little or no influence Straight / Heterosexual       Yes
## 240   I have little or no influence Straight / Heterosexual       Yes
## 290                          <NA> Straight / Heterosexual       Yes
## 309 I have a great deal of influence Straight / Heterosexual       Yes
##             SOComm                        SOPartFreq
## 44   Yes, somewhat     A few times per month or weekly
## 59   Yes, somewhat                A few times per week
## 189        Neutral     A few times per month or weekly
## 240 Yes, definitely              Multiple times per day
## 290 Yes, definitely     A few times per month or weekly
## 309        Neutral Less than once per month or monthly
##                       SOVisitFreq                SurveyEase
## 44          Multiple times per day Neither easy nor difficult
## 59            A few times per week                      Easy
## 189          Daily or almost daily Neither easy nor difficult
## 240          Daily or almost daily                      Easy
## 290          Multiple times per day                      Easy
```

```
## 309 A few times per month or weekly                      Easy
##              SurveyLength Trans
## 44  Appropriate in length    No
## 59  Appropriate in length    No
## 189 Appropriate in length    No
## 240             Too short    No
## 290 Appropriate in length    No
## 309 Appropriate in length    No
##                                                                 UndergradMajor
## 44  Another engineering discipline (such as civil, electrical, mechanical, etc.)
## 59               Computer science, computer engineering, or software engineering
## 189              Computer science, computer engineering, or software engineering
## 240        Information systems, information technology, or system administration
## 290                                                                       <NA>
## 309              Computer science, computer engineering, or software engineering
##                        WebframeDesireNextYear
## 44                                       <NA>
## 59                       Angular;ASP.NET;jQuery
## 189                             Angular;Django
## 240                                     jQuery
## 290                                      Flask
## 309 Angular;ASP.NET Core;Django;Express;Vue.js
##                         WebframeWorkedWith
## 44                                    <NA>
## 59  ASP.NET;jQuery;Laravel;Ruby on Rails
## 189                               React.js
## 240                                 jQuery
## 290                                  Flask
## 309                 ASP.NET Core;Express
##                                 WelcomeChange WorkWeekHrs YearsCode YearsCodePro
## 44  Just as welcome now as I felt last year          45        10            6
## 59  Just as welcome now as I felt last year          40        24           15
## 189 Just as welcome now as I felt last year          40        20           12
## 240 Just as welcome now as I felt last year          44         5            3
## 290 Just as welcome now as I felt last year          NA         5         <NA>
## 309 Just as welcome now as I felt last year          40        10           10
```

De este modo se pide al estudiante que complete los siguientes pasos:

15%. Se generan reglas y se comentan e interpretan las más significativas. Adicionalmente se genera matriz de confusión pa

1. Aplicar un modelo de generación de reglas a partir de **árboles de decisión**.

Ahora procedemos a elegir las variables categóricas del dataset *dataSudamerica*. También existen variables extras que se pueden agregar o discretizar previo a obtener las reglas del árbol de decisión.

Seleccionaremos las variables que formarán parte del dataset que se evaluará con el árbol de decisión

```
# Creamos un juego de datos resumido
dataAD <- dataSudamerica[, c(2, 4, 8, 15:16, 18, 27, 48, 55, 59:60)]

head(dataAD)
```

```
##                         MainBranch Age ConvertedComp
```

```
## 44  I am a developer by profession  32         55893
## 59  I am a developer by profession  38         16488
## 189 I am a developer by profession  NA           NA
## 240 I am a developer by profession  28         8244
## 290    I code primarily as a hobby  18           NA
## 309 I am a developer by profession  27         8712
##                                                    EdLevel         Employment
## 44         Master's degree (M.A., M.S., M.Eng., MBA, etc.) Employed full-time
## 59            Bachelor's degree (B.A., B.S., B.Eng., etc.) Employed full-time
## 189                    Associate degree (A.A., A.S., etc.) Employed full-time
## 240 Some college/university study without earning a degree Employed full-time
## 290                              Primary/elementary school            Student
## 309 Some college/university study without earning a degree Employed full-time
##     Gender                                              NEWCollabToolsWorkedWith
## 44     Man                                                   Github;Slack;Trello
## 59     Man                                                   Github;Slack;Trello
## 189    Man Confluence;Jira;Github;Gitlab;Slack;Google Suite (Docs, Meet, etc)
## 240    Man                                                                Github
## 290    Man                                                        Github;Gitlab
## 309    Man                                          Github;Slack;Microsoft Azure
##     SOAccount
## 44        Yes
## 59        Yes
## 189       Yes
## 240       Yes
## 290       Yes
## 309       Yes
##                                                                    UndergradMajor
## 44  Another engineering discipline (such as civil, electrical, mechanical, etc.)
## 59               Computer science, computer engineering, or software engineering
## 189              Computer science, computer engineering, or software engineering
## 240        Information systems, information technology, or system administration
## 290                                                                          <NA>
## 309              Computer science, computer engineering, or software engineering
##     WorkWeekHrs YearsCode
## 44           45        10
## 59           40        24
## 189          40        20
## 240          44         5
## 290          NA         5
## 309          40        10
```

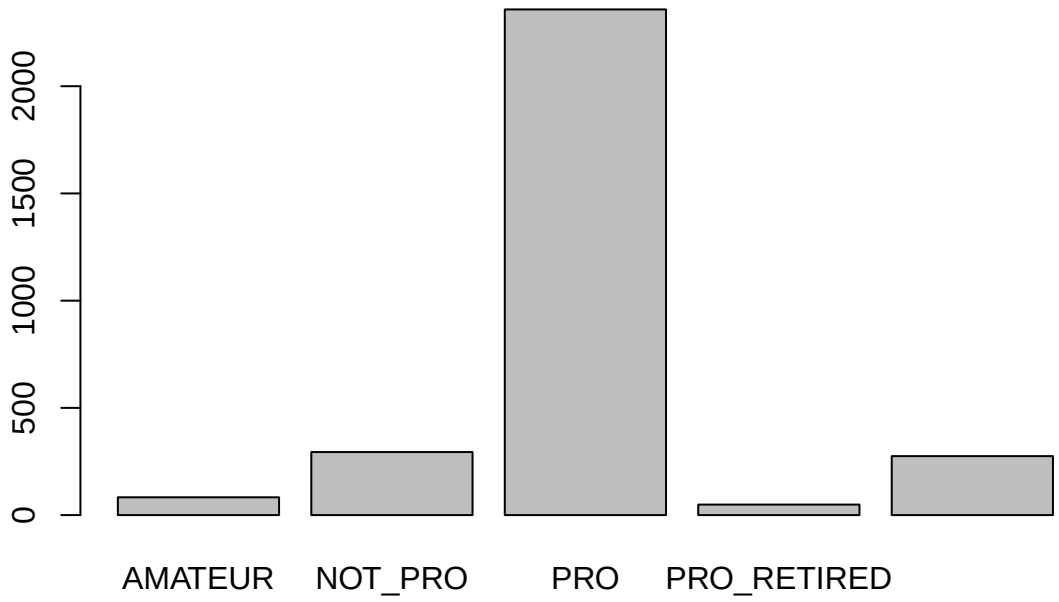
La clase será la variable *MainBranch*, para lo cual vamos a realizar una reasignación de valores, con nomenclaturas más cortas, para lo cual procederemos de la siguiente manera:

- "I am a developer by profession" -> "PRO"
- "I am not primarily a developer, but I write code sometimes as part of my work" -> "NOT_PRO"
- "I used to be a developer by profession, but no longer am" -> "PRO_RETIRED"
- "I am a student who is learning to code" -> "STUDENT"
- "I code primarily as a hobby" -> "AMATEUR"
- "NA" -> "NOT_DEFINED"

```r
# Reasigamos el valor
dataAD$MainBranch[dataAD$MainBranch=="I am a developer by profession"] <- 'PRO'

# Reasigamos el valor
dataAD$MainBranch[dataAD$MainBranch=="I am not primarily a developer, but I write code sometimes as par

# Reasigamos el valor
dataAD$MainBranch[dataAD$MainBranch=="I used to be a developer by profession, but no longer am"] <- 'PR

# Reasigamos el valor
dataAD$MainBranch[dataAD$MainBranch=="I am a student who is learning to code"] <- 'STUDENT'

# Reasigamos el valor
dataAD$MainBranch[dataAD$MainBranch=="I code primarily as a hobby"] <- 'AMATEUR'

# Eliminamos los valores que no tienen clase (NA)
dataAD <- dataAD[!is.na(dataAD$MainBranch), ]

#Definimos el total de filas del dataset
filasAD=dim(dataAD)[1]

table(as.factor(dataAD$MainBranch))
```

```
##
##      AMATEUR      NOT_PRO          PRO PRO_RETIRED      STUDENT
##           83          294         2358           49          275
```

```r
plot(as.factor(dataAD$MainBranch))
```



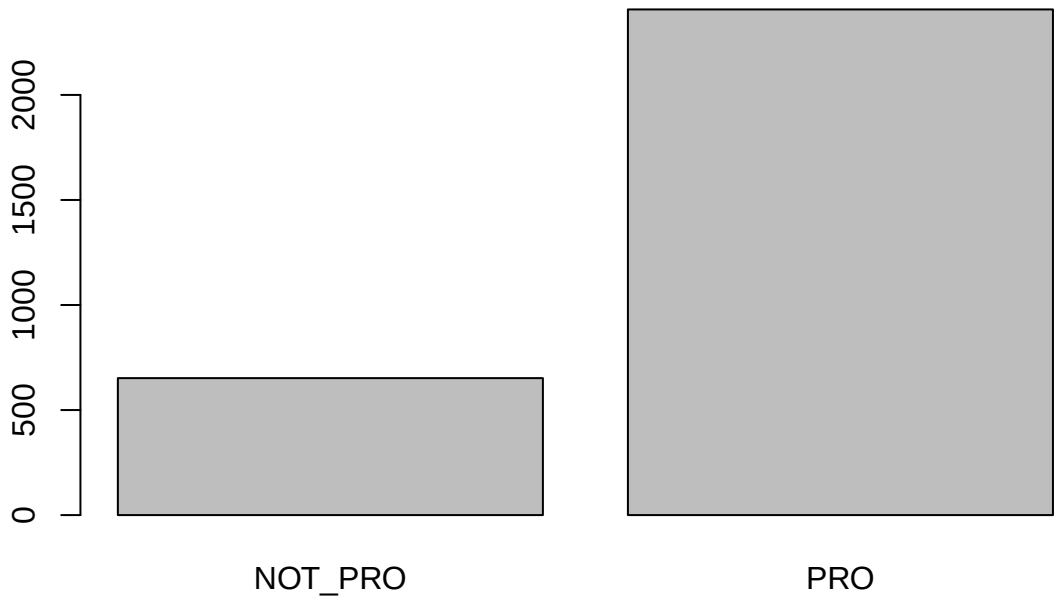
Vamos a tratar de redefinir los valores de la variable clase *MainBranch* para agrupar y dejar solo 2 clases:

- PRO (Desarrollador profesional)
- NOT_PRO (Desarrollador no profesional)

Procedemos a generar una nueva variable que contendrá las 2 clases antes mencionadas, para lo cual vamos a proceder a agruparlas de acuerdo al siguiente detalle:

- PRO (Desarrollador profesional)
  - PRO
  - PRO_RETIRED
- NOT_PRO (Desarrollador no profesional)
  - NOT_PRO
  - STUDENT
  - AMATEUR

```
# Agrupamos las clases
dataAD$clase[dataAD$MainBranch %in% c("PRO", "PRO_RETIRED")] = "PRO"

dataAD$clase[dataAD$MainBranch %in% c("NOT_PRO", "STUDENT", "AMATEUR")] = "NOT_PRO"

dataAD$MainBranch = NULL

plot(as.factor(dataAD$clase))
```



Rellenamos los valores NA en los campos Age, ConvertedComp y WorkWeekHrs

```
# Rellanamos los campos con NA con valores medios de cada variable
dataAD$Age[is.na(dataAD$Age)] <- mean(dataAD$Age,na.rm=T)

dataAD$ConvertedComp[is.na(dataAD$ConvertedComp)] <- mean(dataAD$ConvertedComp,na.rm=T)

dataAD$WorkWeekHrs[is.na(dataAD$WorkWeekHrs)] <- mean(dataAD$WorkWeekHrs,na.rm=T)
```

En el análisis previo de los tipos de datos vimos que la variable YearsCode es de tipo caracter y no numérico como se esperaba, por lo que debemos realizar una conversión. **Aunque en el caso de Ecuador no se dan casos de respuestas no numéricas (más de 50 años y menos de un año), indicaremos el tratamiento que se debe dar a los casos donde los programadores, hayan optado por estas respuestas categóricas**. Para los registros que tienen *"More than 50 years"* definiremos el valor a 50 y para el caso de *"Less than 1 year"* el valor será 1:

```
dataAD$YearsCode[dataAD$YearsCode=="More than 50 years"] <- 50
dataAD$YearsCode[dataAD$YearsCode=="Less than 1 year"] <- 1

# Finalmente convertimos dicha columna en númerica
dataAD$YearsCode <- as.numeric(dataAD$YearsCode)

# Llenamos con la media los valores faltantes
dataAD$YearsCode[is.na(dataAD$YearsCode)] <- mean(dataAD$YearsCode, na.rm=T)
```

Verificamos los valores NA en las demás variables. Para los valores NA utilizaremos la nomenclatura *NOT_DEFINED*.

```
colSums(is.na(dataAD))
```

```
##                      Age            ConvertedComp                  EdLevel
##                        0                        0                      336
##               Employment                   Gender NEWCollabToolsWorkedWith
##                       12                      639                      525
##                SOAccount           UndergradMajor              WorkWeekHrs
##                      344                      600                        0
##                YearsCode                   clase
##                        0                        0
```

```
# Rellenamos los valores NA para la variable EdLevel
dataAD$EdLevel[is.na(dataAD$EdLevel)] <- 'NOT_DEFINED'

# Rellenamos los valores NA para la variable Employment
dataAD$Employment[is.na(dataAD$Employment)] <- 'NOT_DEFINED'

# Rellenamos los valores NA para la variable Gender
dataAD$Gender[is.na(dataAD$Gender)] <- 'NOT_DEFINED'

# Rellenamos los valores NA para la variable NEWCollabToolsWorkedWith
dataAD$NEWCollabToolsWorkedWith[is.na(dataAD$NEWCollabToolsWorkedWith)] <- 'NOT_DEFINED'

# Rellenamos los valores NA para la variable SOAccount
dataAD$SOAccount[is.na(dataAD$SOAccount)] <- 'NOT_DEFINED'

# Rellenamos los valores NA para la variable UndergradMajor
dataAD$UndergradMajor[is.na(dataAD$UndergradMajor)] <- 'NOT_DEFINED'
```

La variable *EdLevel* tiene valores muy largos que no ayudarán a obtener las reglas, para lo cual vamos a realizar una reasignación de valores, con nomenclaturas más cortas, para lo cual procederemos de la siguiente manera:

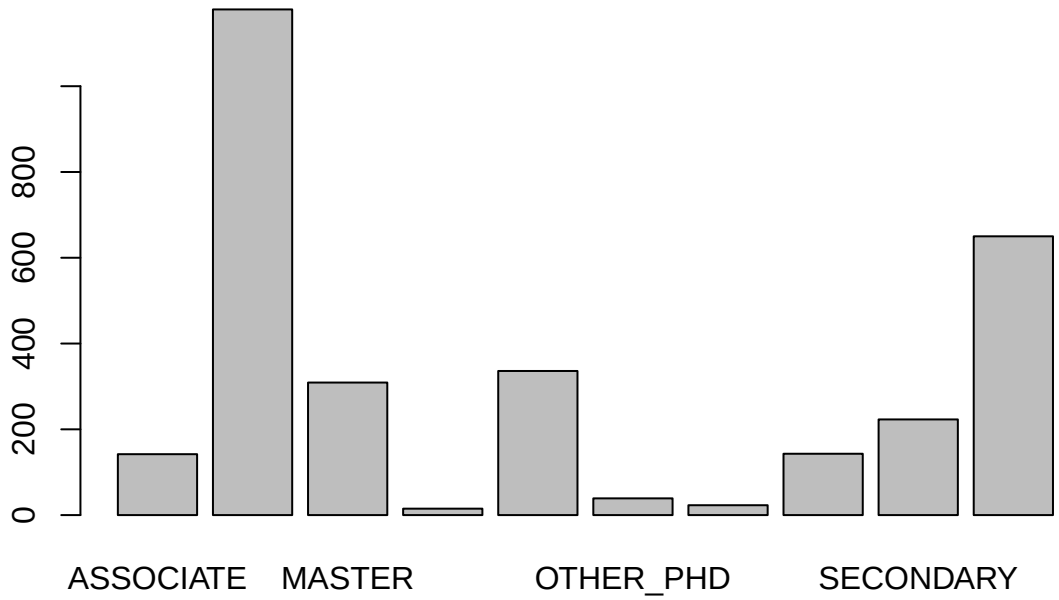
- "I never completed any formal education" -> "NEVER"
- "Primary/elementary school" -> "PRIMARY"
- "Secondary school (e.g. American high school, German Realschule or Gymnasium, etc.)" -> "SECONDARY"
- "Some college/university study without earning a degree" -> "SOME_STUDY_WITHOUT_DEGREE"
- "Associate degree (A.A., A.S., etc.)" -> "ASSOCIATE"
- "Bachelor's degree (B.A., B.S., B.Eng., etc.)" -> "BACHELOR"
- "Master's degree (M.A., M.S., M.Eng., MBA, etc.)" -> "MASTER"
- "Professional degree (JD, MD, etc.)" -> "PROFESSIONAL"
- "Other doctoral degree (Ph.D., Ed.D., etc.)" -> "OTHER_PHD"

```
# Reasigamos el valor
dataAD$EdLevel[dataAD$EdLevel=="I never completed any formal education"] <- 'NEVER'

# Reasigamos el valor
dataAD$EdLevel[dataAD$EdLevel=="Primary/elementary school"] <- 'PRIMARY'

# Reasigamos el valor
dataAD$EdLevel[dataAD$EdLevel=="Secondary school (e.g. American high school, German Realschule or Gymnas
```

```
# Reasigamos el valor
dataAD$EdLevel[dataAD$EdLevel=="Some college/university study without earning a degree"] <- 'SOME_STUDY_

# Reasigamos el valor
dataAD$EdLevel[dataAD$EdLevel=="Associate degree (A.A., A.S., etc.)"] <- 'ASSOCIATE'

# Reasigamos el valor
dataAD$EdLevel[dataAD$EdLevel=="Bachelor's degree (B.A., B.S., B.Eng., etc.)"] <- 'BACHELOR'

# Reasigamos el valor
dataAD$EdLevel[dataAD$EdLevel=="Master's degree (M.A., M.S., M.Eng., MBA, etc.)"] <- 'MASTER'

# Reasigamos el valor
dataAD$EdLevel[dataAD$EdLevel=="Professional degree (JD, MD, etc.)"] <- 'PROFESSIONAL'

# Reasigamos el valor
dataAD$EdLevel[dataAD$EdLevel=="Other doctoral degree (Ph.D., Ed.D., etc.)"] <- 'OTHER_PHD'


table(as.factor(dataAD$EdLevel))
```

```
##
##                 ASSOCIATE                  BACHELOR                    MASTER
##                       142                      1179                       309
##                     NEVER                NOT_DEFINED                 OTHER_PHD
##                        15                       336                        39
##                   PRIMARY                PROFESSIONAL                 SECONDARY
##                        23                       143                       223
## SOME_STUDY_WITHOUT_DEGREE
##                       650
```

```
plot(as.factor(dataAD$EdLevel))
```



La variable *Employment* tiene valores muy largos que no ayudarán a obtener las reglas, para lo cual vamos a realizar una reasignación de valores, con nomenclaturas más cortas, para lo cual procederemos de la siguiente manera:

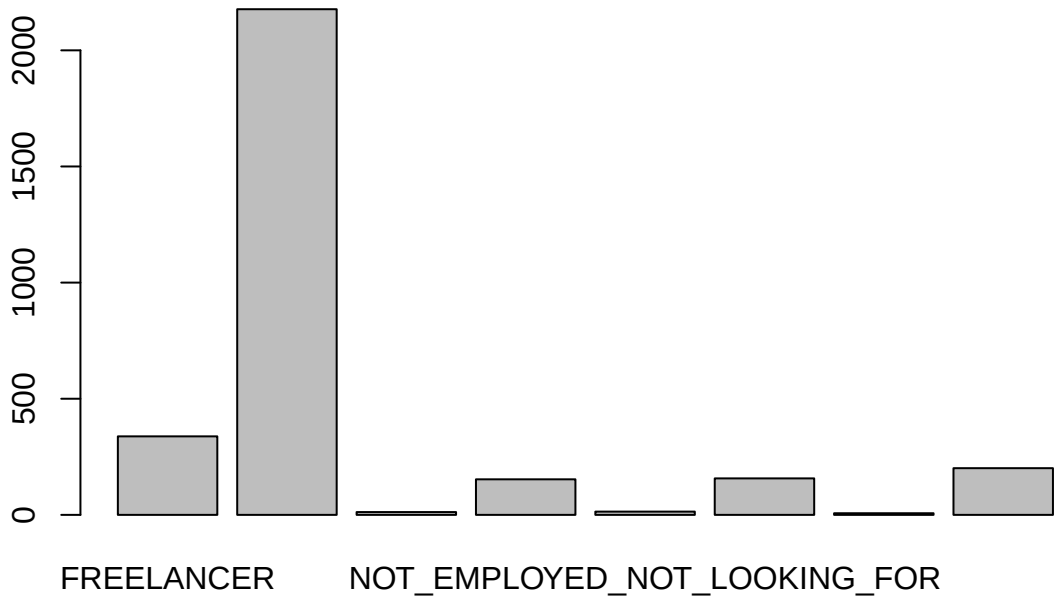
- "Employed full-time" -> "FULL_TIME"
- "Employed part-time" -> "PART_TIME"
- "Independent contractor, freelancer, or self-employed" -> "FREELANCER"
- "Not employed, but looking for work" -> "NOT_EMPLOYED_LOOKING_FOR"
- "Not employed, and not looking for work" -> "NOT_EMPLOYED_NOT_LOOKING_FOR"
- "Student" -> "STUDENT"
- "Retired" -> "RETIRED"

```
# Reasigamos el valor
dataAD$Employment[dataAD$Employment=="Employed full-time"] <- 'FULL_TIME'

# Reasigamos el valor
dataAD$Employment[dataAD$Employment=="Employed part-time"] <- 'PART_TIME'

# Reasigamos el valor
dataAD$Employment[dataAD$Employment=="Independent contractor, freelancer, or self-employed"] <- 'FREELA

# Reasigamos el valor
dataAD$Employment[dataAD$Employment=="Not employed, but looking for work"] <- 'NOT_EMPLOYED_LOOKING_FOR

# Reasigamos el valor
dataAD$Employment[dataAD$Employment=="Not employed, and not looking for work"] <- 'NOT_EMPLOYED_NOT_LOO
```

```
# Reasigamos el valor
dataAD$Employment[dataAD$Employment=="Student"] <- 'STUDENT'

# Reasigamos el valor
dataAD$Employment[dataAD$Employment=="Retired"] <- 'RETIRED'

table(as.factor(dataAD$Employment))
```

```
##
##                     FREELANCER                      FULL_TIME
##                            338                           2177
##                    NOT_DEFINED     NOT_EMPLOYED_LOOKING_FOR
##                             12                            153
## NOT_EMPLOYED_NOT_LOOKING_FOR                     PART_TIME
##                             14                            157
##                        RETIRED                        STUDENT
##                              7                            201
```

```
plot(as.factor(dataAD$Employment))
```



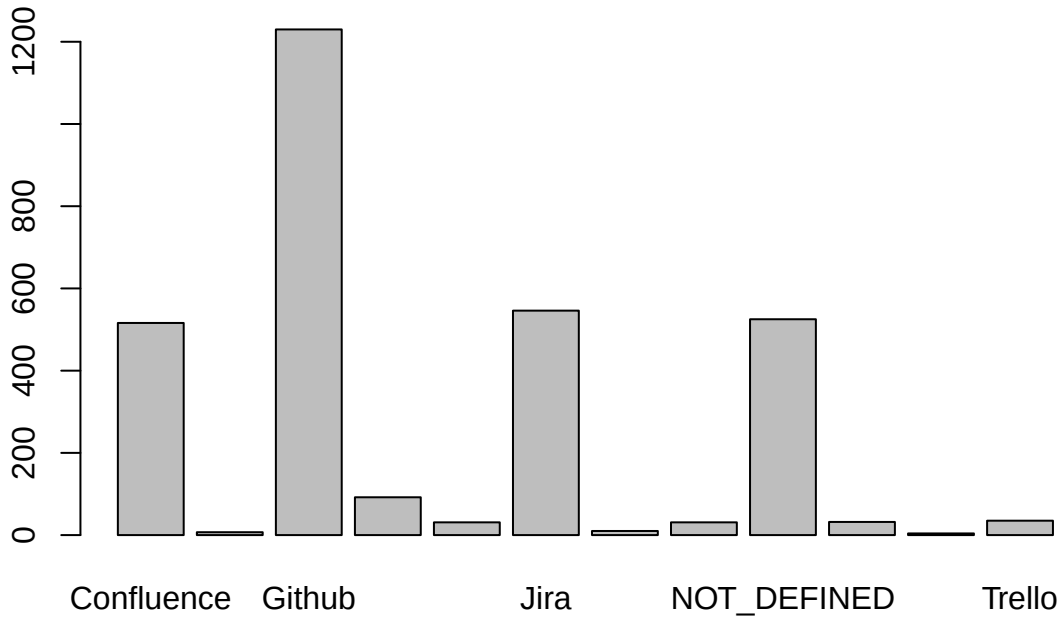
Para la variable *NEWCollabToolsWorkedWith* vamos a realizar una transformación del valor de cada observación. Independientemente del lenguaje de programación, base de datos o demás herramientas donde se pueda volver profesional un desarrollador es un buen indicador de todo programador profesional el usar herramientas colaborativas, como GitHub, Jira, Slack, Trello o Stack Overflow for Teams. Estas herramientas permiten categorizar indiscutiblemente a los programadores profesionales. Así que vamos a truncar

los valores seleccionados por cada encuestado y quedarnos con la primera herramienta colaborativa que se menciona.

```
dataAD$main_colab <- ''

for(i in 1:filasAD) {
  dataAD$main_colab[i]  <- strsplit(dataAD$NEWCollabToolsWorkedWith[i], ";")[[1]][1]
}

# Reasigamos el valor
dataAD$main_colab[dataAD$main_colab=="Google Suite (Docs, Meet, etc)"] <- 'Google Suite'

table(dataAD$main_colab)
```

```
##
##              Confluence       Facebook Workplace                  Github
##                     516                        7                    1230
##                  Gitlab            Google Suite                    Jira
##                      92                       31                     546
##         Microsoft Azure          Microsoft Teams                NOT_DEFINED
##                      10                       31                     525
##                   Slack Stack Overflow for Teams                  Trello
##                      32                        4                      35
```

```
plot(as.factor(dataAD$main_colab))
```



Eliminamos la variable *NEWCollabToolsWorkedWith*

```
dataAD$NEWCollabToolsWorkedWith = NULL
```

La variable *UndergradMajor* tiene valores muy largos que no ayudarán a obtener las reglas, para lo cual vamos a realizar una reasignación de valores, con nomenclaturas más cortas, para lo cual procederemos de la siguiente manera:

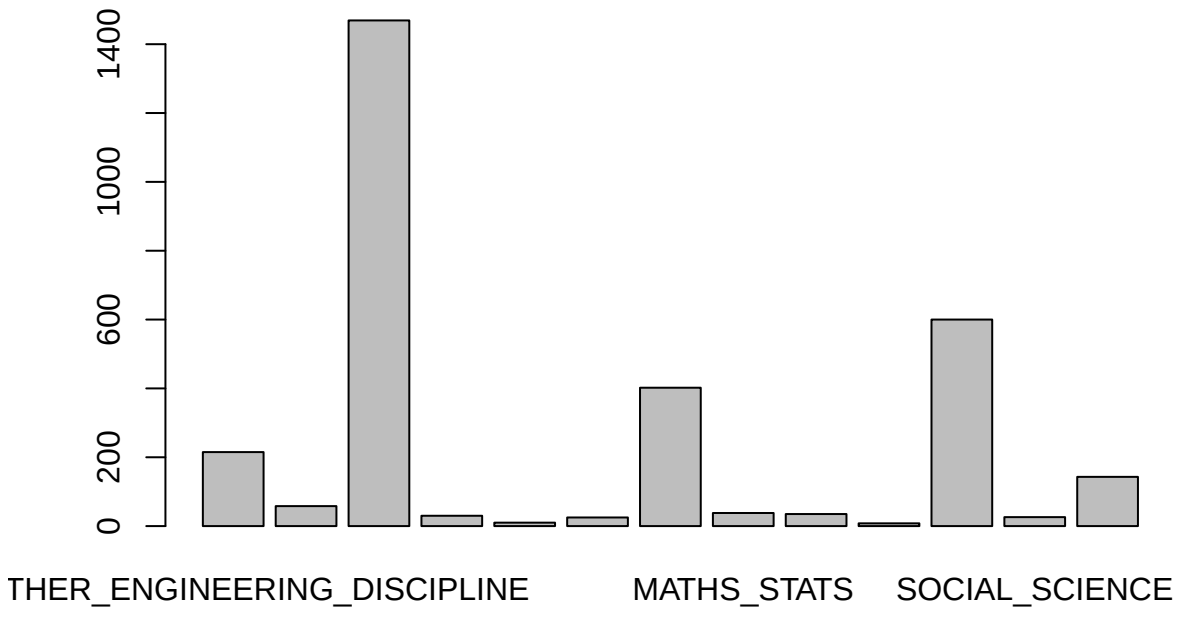
- "Computer science, computer engineering, or software engineering" -> "COMPUTER_SCIENCE"
- "Web development or web design" -> "WEB_DEVELOPMENT"
- "Information systems, information technology, or system administration" -> "INFORMATION_SYSTEMS"
- "Mathematics or statistics" -> "MATHS_STATS"
- "Another engineering discipline (such as civil, electrical, mechanical, etc.)" -> "ANOTHER_ENGINEERING_DISCIPI
- "A business discipline (such as accounting, finance, marketing, etc.)" -> "BUSINESS"
- "A health science (such as nursing, pharmacy, radiology, etc.)" -> "HEALTH"
- "A humanities discipline (such as literature, history, philosophy, etc.)" -> "HUMANITIES"
- "A natural science (such as biology, chemistry, physics, etc.)" -> "NATURAL_SCIENCE"
- "A social science (such as anthropology, psychology, political science, etc.)" -> "SOCIAL_SCIENCE"
- "Fine arts or performing arts (such as graphic design, music, studio art, etc.)" -> "FINE_ARTS"
- "I never declared a major" -> "NEVER_MAJOR"

```
# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="Computer science, computer engineering, or software engine

# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="Web development or web design"] <- 'WEB_DEVELOPMENT'

# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="Information systems, information technology, or system adm

# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="Mathematics or statistics"] <- 'MATHS_STATS'

# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="Another engineering discipline (such as civil, electrical

# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="A business discipline (such as accounting, finance, marke

# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="A health science (such as nursing, pharmacy, radiology, e

# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="A humanities discipline (such as literature, history, phi

# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="A natural science (such as biology, chemistry, physics, e

# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="A social science (such as anthropology, psychology, polit

# Reasigamos el valor
```

```
dataAD$UndergradMajor[dataAD$UndergradMajor=="Fine arts or performing arts (such as graphic design, musi

# Reasigamos el valor
dataAD$UndergradMajor[dataAD$UndergradMajor=="I never declared a major"] <- 'NEVER_MAJOR'


table(as.factor(dataAD$UndergradMajor))
```

```
##
## ANOTHER_ENGINEERING_DISCIPLINE                    BUSINESS
##                            215                          58
##                COMPUTER_SCIENCE                FINE_ARTS
##                           1469                          30
##                         HEALTH                HUMANITIES
##                             10                          25
##             INFORMATION_SYSTEMS                MATHS_STATS
##                            402                          38
##                NATURAL_SCIENCE                NEVER_MAJOR
##                             35                           8
##                    NOT_DEFINED             SOCIAL_SCIENCE
##                            600                          26
##                WEB_DEVELOPMENT
##                            143
```

```
plot(as.factor(dataAD$UndergradMajor))
```



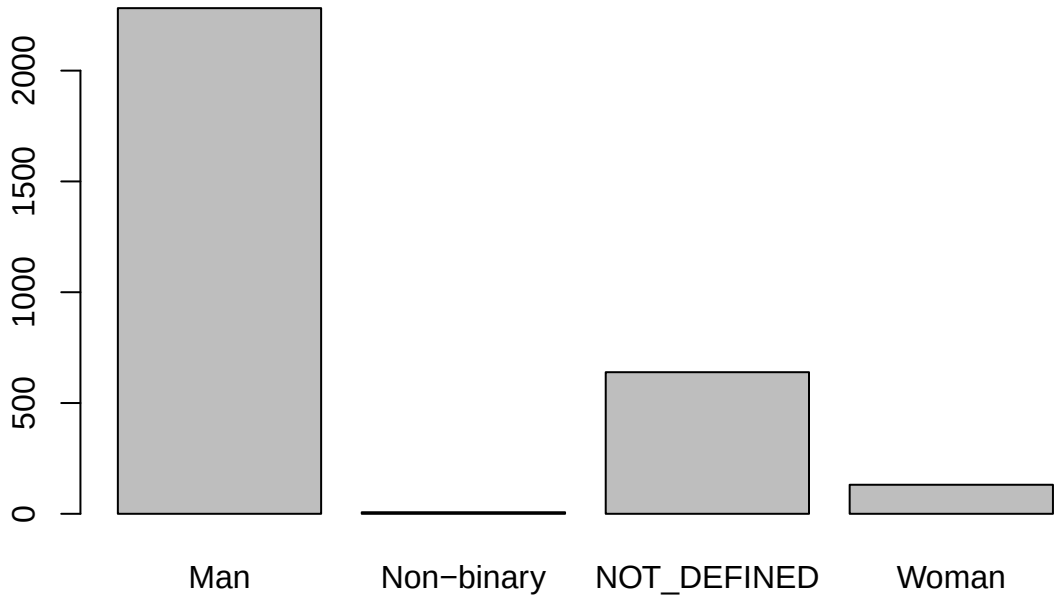
La variable *Gender* también debe ser tratada para corregir valores nulos y/o atípicos, para lo cual procederemos de la siguiente manera:

- "NA" -> "NOT_DEFINED"
- "Non-binary, genderqueer, or gender non-conforming" -> "Non-binary"
- Para los valores Woman y Man los valores no se cambian

```
dataAD$genero <- ''

for(i in 1:filasAD) {
  if (dataAD$Gender[i]=="Non-binary, genderqueer, or gender non-conforming") {
    dataAD$genero[i] <- 'Non-binary'
  } else {
    dataAD$genero[i] <- strsplit(dataAD$Gender[i], ";")[[1]][1]
  }
}

table(as.factor(dataAD$genero))
```

```
##
##         Man  Non-binary NOT_DEFINED       Woman
##        2282           7         639         131
```

```
plot(as.factor(dataAD$genero))
```



Eliminamos la variable *Gender*

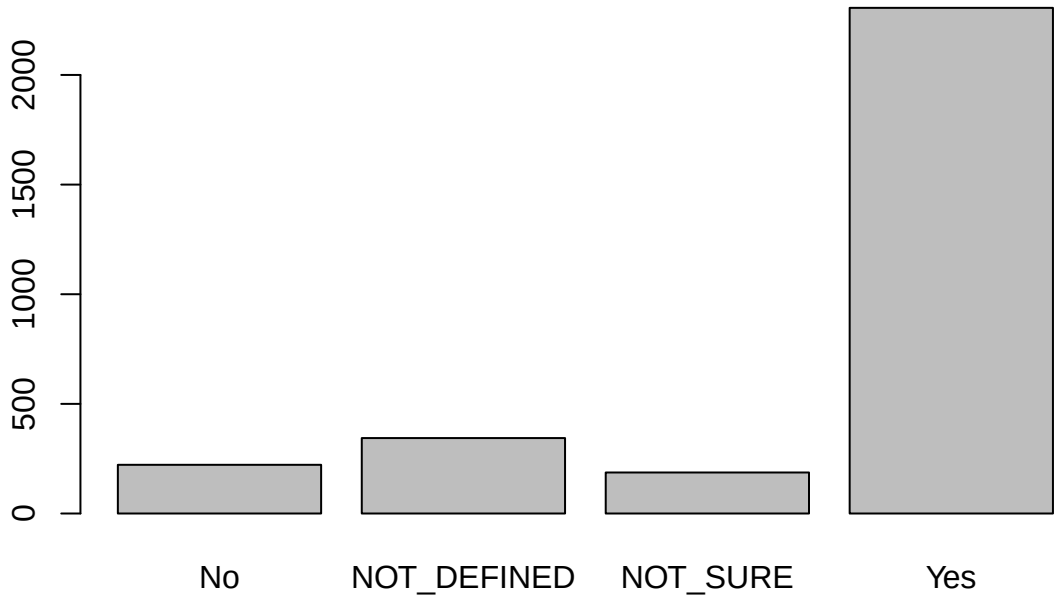
```
dataAD$Gender = NULL
dataAD$sexo = NULL
```

Para la variable *SOAccount* también vamos a realizar una limpieza en los valores nulos o que tienen valores muy extensos para hacerlos más representativos y fáciles de tratar al momento de aplicar el modelo.

```
dataAD$SOAccount[dataAD$SOAccount=="Not sure/can't remember"] <- 'NOT_SURE'

table(dataAD$SOAccount)
```

```
##
##          No NOT_DEFINED    NOT_SURE         Yes
##         222         344         187        2306
```

```
plot(as.factor(dataAD$SOAccount))
```



Por último aplicamos discretización sobre los campos: Age, ConvertedComp, WorkWeekHrs y YearsCode

```
# Discretizamos para la variable Age
dataAD$segm_edad <- cut(dataAD$Age, breaks = c(0,10,20,30,40,50,60,70,100), labels = c("0-9", "10-19",

# Discretizamos para la variable ConvertedComp
dataAD$segm_salario_anual <- cut(dataAD$ConvertedComp, breaks = c(0,25000,50000,75000,100000,125000,200
```

```
# Discretizamos para la variable WorkWeekHrs
dataAD$segm_horas_trab <- cut(dataAD$WorkWeekHrs, breaks = c(0,20,40,60,80,168), labels = c("0-19", "20

# Discretizamos para la variable YearsCode
dataAD$segm_years_code <- cut(dataAD$YearsCode, breaks = c(0,5,10,15,20,30,40,50), labels = c("0-4", "5

#Eliminamos las variables numéricas ya discretizadas
dataAD$Age = NULL
dataAD$ConvertedComp = NULL
dataAD$WorkWeekHrs = NULL
dataAD$YearsCode = NULL
```

Procedemos a "desordenar" el dataset. Guardaremos los datos en un nuevo dataset: "data_random".

```
set.seed(1)
datos_random <- dataAD[sample(nrow(dataAD)),]
```

La variable que usaremos para la clasificación es el campo *clase*, que está en la quinta columna del conjunto de datos.

```
set.seed(1234)
y <- datos_random[,5]
X <- datos_random[,c(1:4, 6:11)]
```

Podemos elegir el subconjunto de entrenamiento y de prueba de diversas maneras. La primer opción consiste en calcular a cuántas filas corresponde dos tercios de los datos (2*3059/3= ~2040) y dividir "manualmente" el conjunto.

```
trainX <- X[1:2040,]
trainy <- y[1:2040]
testX <- X[2041:3059,]
testy <- y[2041:3059]
```

Creación del modelo, calidad del modelo y extracción de reglas

Se crea el árbol de decisión usando los datos de entrenamiento:

```
trainy = as.factor(trainy)

model <- C50::C5.0(trainX, trainy, rules=TRUE )

summary(model)
```

```
##
## Call:
## C5.0.default(x = trainX, y = trainy, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Thu Jan 21 08:18:37 2021
## -----------------------------
```

```
##
## Class specified by attribute 'outcome'
##
## Read 2040 cases (11 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (21/1, lift 4.1)
##  UndergradMajor in {SOCIAL_SCIENCE, NOT_DEFINED,
##                             ANOTHER_ENGINEERING_DISCIPLINE, INFORMATION_SYSTEMS,
##                             NATURAL_SCIENCE, MATHS_STATS}
##  main_colab in {Google Suite, Trello, Microsoft Azure,
##                         Stack Overflow for Teams}
##  ->  class NOT_PRO  [0.913]
##
## Rule 2: (158/26, lift 3.7)
##  Employment in {STUDENT, NOT_DEFINED}
##  segm_salario_anual = 25000-49999
##  ->  class NOT_PRO  [0.831]
##
## Rule 3: (144/27, lift 3.6)
##  UndergradMajor in {SOCIAL_SCIENCE, NOT_DEFINED,
##                             ANOTHER_ENGINEERING_DISCIPLINE, HEALTH,
##                             INFORMATION_SYSTEMS, HUMANITIES, NATURAL_SCIENCE,
##                             FINE_ARTS, BUSINESS, MATHS_STATS}
##  main_colab in {Github, NOT_DEFINED, Microsoft Teams, Facebook Workplace}
##  segm_salario_anual = 25000-49999
##  segm_years_code = 0-4
##  ->  class NOT_PRO  [0.808]
##
## Rule 4: (34/11, lift 3.0)
##  EdLevel in {PROFESSIONAL, OTHER_PHD}
##  UndergradMajor in {SOCIAL_SCIENCE, NOT_DEFINED,
##                             ANOTHER_ENGINEERING_DISCIPLINE, HEALTH,
##                             INFORMATION_SYSTEMS, HUMANITIES, NATURAL_SCIENCE,
##                             FINE_ARTS, BUSINESS, MATHS_STATS}
##  main_colab in {Github, Slack, NOT_DEFINED, Facebook Workplace}
##  ->  class NOT_PRO  [0.667]
##
## Rule 5: (378/145, lift 2.7)
##  Employment in {PART_TIME, STUDENT, NOT_EMPLOYED_NOT_LOOKING_FOR,
##                         NOT_EMPLOYED_LOOKING_FOR, RETIRED, NOT_DEFINED}
##  ->  class NOT_PRO  [0.616]
##
## Rule 6: (943/61, lift 1.2)
##  Employment in {FULL_TIME, FREELANCER}
##  UndergradMajor in {COMPUTER_SCIENCE, WEB_DEVELOPMENT, NEVER_MAJOR}
##  ->  class PRO  [0.934]
##
## Rule 7: (726/52, lift 1.2)
##  Employment in {FULL_TIME, FREELANCER}
##  main_colab in {Confluence, Jira, Gitlab}
##  ->  class PRO  [0.927]
##
```

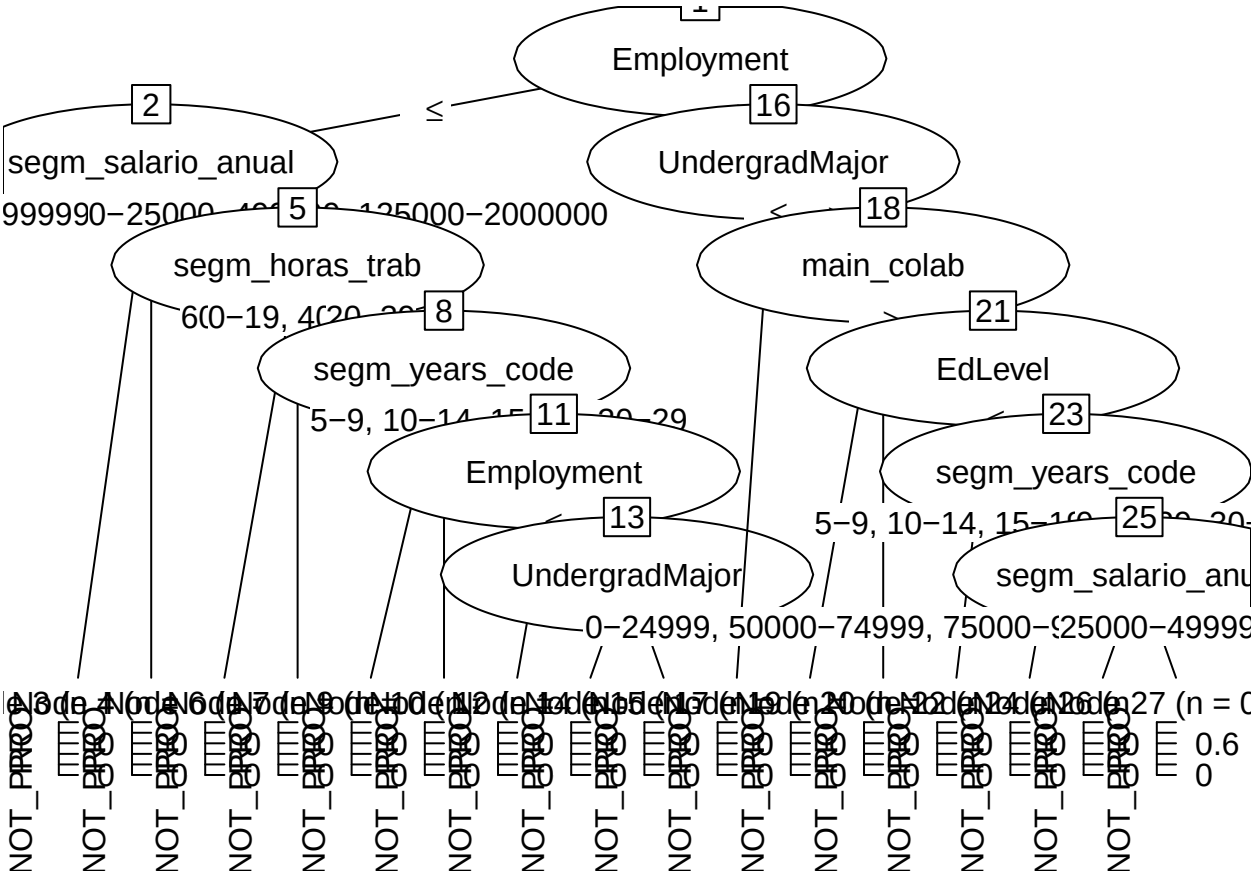
```
## Rule 8: (837/72, lift 1.2)
##  segm_salario_anual = 0-24999
##  ->  class PRO  [0.913]
##
## Rule 9: (1304/132, lift 1.2)
##  EdLevel in {MASTER, BACHELOR, SOME_STUDY_WITHOUT_DEGREE, SECONDARY,
##                     NOT_DEFINED, ASSOCIATE, NEVER, PRIMARY}
##  Employment in {FULL_TIME, FREELANCER}
##  segm_years_code in {5-9, 10-14, 15-19, 20-29, 30-39, > 40}
##  ->  class PRO  [0.898]
##
## Rule 10: (1252/180, lift 1.1)
##  UndergradMajor in {COMPUTER_SCIENCE, NOT_DEFINED, HEALTH,
##                             WEB_DEVELOPMENT, NATURAL_SCIENCE, FINE_ARTS,
##                             BUSINESS, MATHS_STATS}
##  segm_years_code in {5-9, 10-14, 15-19, 20-29}
##  ->  class PRO  [0.856]
##
## Default class: PRO
##
##
## Evaluation on training data (2040 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##      10  284(13.9%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     244   213    (a): class NOT_PRO
##      71  1512    (b): class PRO
##
##
##  Attribute usage:
##
##   93.58% Employment
##   82.65% segm_years_code
##   77.70% UndergradMajor
##   65.59% EdLevel
##   52.94% segm_salario_anual
##   45.05% main_colab
##
##
## Time: 0.0 secs
```

En el resumen, en la sección de evaluación en los datos de entrenamiento, el valor *Errors* muestra una tasa de error de 13.9% para las reglas obtenidas a partir del dataset de entrenamiento (trainx), la cual es muy aceptable. El árbol obtenido clasifica erróneamente 284 de los 2040 casos presentes.

A continuación mostramos el árbol obtenido.

```
model <- C50::C5.0(trainX, trainy)
plot(model)
```

```
## Warning in partysplit(varid = as.integer(i), breaks = as.numeric(j[1]), : NAs
## introduced by coercion

## Warning in partysplit(varid = as.integer(i), breaks = as.numeric(j[1]), : NAs
## introduced by coercion

## Warning in partysplit(varid = as.integer(i), breaks = as.numeric(j[1]), : NAs
## introduced by coercion

## Warning in partysplit(varid = as.integer(i), breaks = as.numeric(j[1]), : NAs
## introduced by coercion

## Warning in partysplit(varid = as.integer(i), breaks = as.numeric(j[1]), : NAs
## introduced by coercion

## Warning in partysplit(varid = as.integer(i), breaks = as.numeric(j[1]), : NAs
## introduced by coercion


## Warning in .bincode(as.numeric(x), breaks = unique(c(-Inf,
## breaks_split(split), : NAs introduced by coercion

## Warning in .bincode(as.numeric(x), breaks = unique(c(-Inf,
## breaks_split(split), : NAs introduced by coercion

## Warning in .bincode(as.numeric(x), breaks = unique(c(-Inf,
## breaks_split(split), : NAs introduced by coercion

## Warning in .bincode(as.numeric(x), breaks = unique(c(-Inf,
## breaks_split(split), : NAs introduced by coercion

## Warning in .bincode(as.numeric(x), breaks = unique(c(-Inf,
## breaks_split(split), : NAs introduced by coercion
```



A partir del árbol de decisión de dos hojas que hemos modelado, se pueden extraer las siguientes reglas de decisión:

- Regla 1: Si el campo de estudio principal está entre: Ciencias Sociales, Otras disciplinas de ingeniería, sistemas de información, ciencias naturales o matemáticas y; las herramientas colaborativas que usa son: Google Suite, Trello, Microsoft Azure o Stack Overflow for Teams -> NO es un desarrollador profesional. **Validez = 91.3%**
- Regla 2: Si es un estudiante o no tiene definido su estado de empleo y además de eso su salario anual está entre 25K y 50K -> NO es un desarrollador profesional. **Validez = 83.1%**
- Regla 3: Si el campo de estudio principal es cualquier otro que no sea Ciencias de la Computación y sus herramientas colaborativas más usadas son: GitHub, Microsoft Teams, Facebook Workplace o no lo ha definido; además su salario anual flutúa entre los 25K a 50K y los años que tiene programando son entre 0 a 4 -> NO es un desarrollador profesional. **Validez = 80.8%**
- Regla 4: Si el nivel más alto de educación formal completado por el encuestado es un grado profesional o Ph.D y; el campo de estudio principal es diferente al de ciencias de la coomputación o desarrollo web y; además usa las herramientas colaborativas GitHub, Slack, Facebook Workplace -> No es un desarrollador profesional. **Validez = 66.7%**
- Regla 5: Si el encuestado NO trabaja a tiempo completo o NO es un freelancer -> No es un desarrollador profesional. **Validez = 61.6%**
- Regla 6: Si trabaja a tiempo completo o es un freelancer y; además tiene un grado en ciencias de la computación, desarrollo web -> SI es un desarrollador profesional. **Validez = 93.4%**
- Regla 7: Si trabaja a tiempo completo o es un freelancer y; además usa como herramientas colaborativas Confluence, Jira o Gitlab -> SI es un desarrollador profesional. **Validez = 92.7%**

- Regla 8: Si el salario anual es inferior a $25K en Sudamérica -> SI es un desarrollador profesional. **Validez = 91.3%**
- Regla 9: Si el nivel de estudio más alto completado es un master, bachelor, secundaria, associate o primaria; trabaja a tiempo completo o es un freelancer y la experiencia en años es superior a 5 años -> SI es un desarrollador profesional. **Validez = 89.8%**
- Regla 10: Si el campo de estudio del encuestado está entre ciencias de la computación, salud, desarrollo web, ciencias naturales, negocios, matemáticas o estadísticas y artes y tiene más de 5 años de experiencia escribiendo código -> SI es un desarrollador profesional. **Validez = 85.6%**

Una vez tenemos el modelo, podemos comprobar su calidad prediciendo la clase para los datos de prueba que nos hemos reservado al principio.

```
predicted_model <- predict(model, testX, type="class")
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(predicted_model == testy) / length(predicted_
```

```
## [1] "La precisión del árbol es: 84.2002 %"
```

15%. Se genera modelo no supervisado, se muestran y comentan medidas de calidad del modelo generado y se comentan las

2. Aplicar un modelo **no supervisado** y basado en el concepto de **distancia**, sobre el juego de datos.

Ahora procedemos a elegir las variables categóricas del dataset *dataSudamerica*. Seleccionaremos las variables que formarán parte del dataset que se evaluará con el **método no supervisado K-Means**

```
# Creamos un juego de datos resumido
dataWithClass <- dataSudamerica[, c(2, 4, 8, 13, 23, 25, 27, 45, 57, 59:60)]

head(dataWithClass)
```

```
##                         MainBranch Age ConvertedComp
## 44  I am a developer by profession  32         55893
## 59  I am a developer by profession  38         16488
## 189 I am a developer by profession  NA            NA
## 240 I am a developer by profession  28          8244
## 290    I code primarily as a hobby  18            NA
## 309 I am a developer by profession  27          8712
##                                          DatabaseWorkedWith
## 44                                        Microsoft SQL Server
## 59                 Microsoft SQL Server;MySQL;PostgreSQL;SQLite
## 189                                       MySQL;Oracle;SQLite
## 240                                                      <NA>
## 290                                       MongoDB;MySQL;SQLite
## 309 DynamoDB;Firebase;MariaDB;Microsoft SQL Server;MySQL;SQLite
##                                                                LanguageWorkedWith
## 44                                                        HTML/CSS;Python;R;SQL;VBA
## 59                                                C#;HTML/CSS;JavaScript;PHP;Ruby
## 189                                 Bash/Shell/PowerShell;HTML/CSS;Java;JavaScript
## 240                                                        HTML/CSS;JavaScript
## 290                                                                    C#;Python
## 309 Bash/Shell/PowerShell;C#;Dart;HTML/CSS;JavaScript;Kotlin;Python;Swift;TypeScript
##                                          MiscTechWorkedWith
```

```
## 44                                                                    <NA>
## 59                                                  .NET;Node.js;Unity 3D
## 189                                                       Cordova;Node.js
## 240                                                React Native;Xamarin
## 290                                                        .NET;.NET Core
## 309 .NET Core;Cordova;Flutter;Node.js;Pandas;Unity 3D;Unreal Engine
##                                                 NEWCollabToolsWorkedWith
## 44                                                   Github;Slack;Trello
## 59                                                   Github;Slack;Trello
## 189 Confluence;Jira;Github;Gitlab;Slack;Google Suite (Docs, Meet, etc)
## 240                                                                Github
## 290                                                       Github;Gitlab
## 309                                      Github;Slack;Microsoft Azure
##                                    PlatformWorkedWith
## 44                                            Windows
## 59                                        Docker;MacOS
## 189                           Android;AWS;Docker;Linux
## 240                                   Android;Windows
## 290        Android;Arduino;Docker;Heroku;Linux;Windows
## 309 Android;Linux;Microsoft Azure;Raspberry Pi;Windows
##                       WebframeWorkedWith WorkWeekHrs YearsCode
## 44                                  <NA>          45        10
## 59  ASP.NET;jQuery;Laravel;Ruby on Rails          40        24
## 189                              React.js          40        20
## 240                                jQuery          44         5
## 290                                 Flask          NA         5
## 309                 ASP.NET Core;Express          40        10
```

La clase será la variable *MainBranch*, para lo cual vamos a realizar una reasignación de valores, con nomenclaturas más cortas, para lo cual procederemos de la siguiente manera:

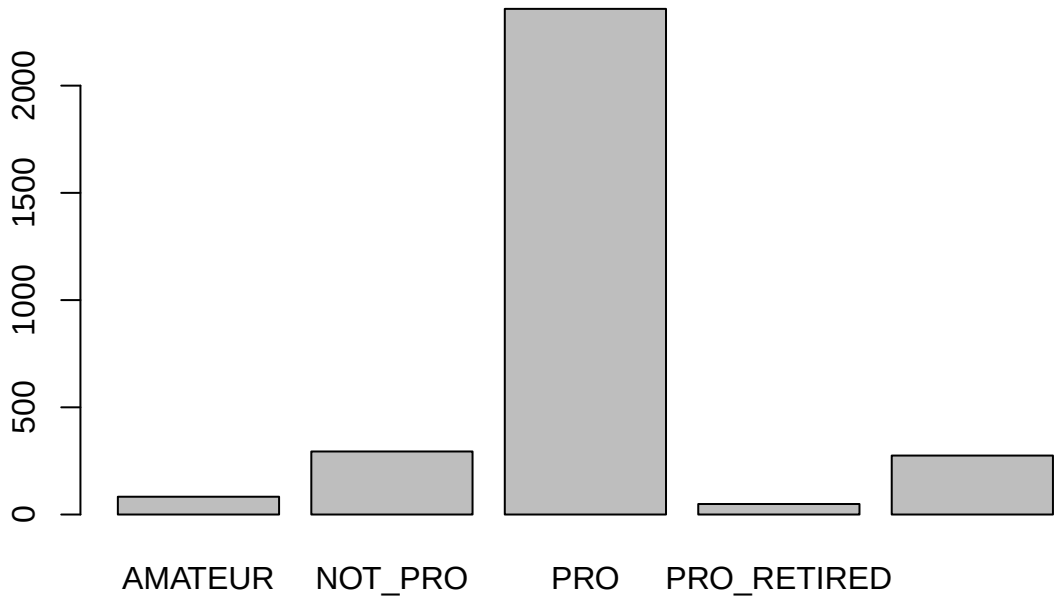
- "I am a developer by profession" -> "PRO"
- "I am not primarily a developer, but I write code sometimes as part of my work" -> "NOT_PRO"
- "I used to be a developer by profession, but no longer am" -> "PRO_RETIRED"
- "I am a student who is learning to code" -> "STUDENT"
- "I code primarily as a hobby" -> "AMATEUR"

```
# Reasigamos el valor
dataWithClass$MainBranch[dataWithClass$MainBranch=="I am a developer by profession"] <- 'PRO'

# Reasigamos el valor
dataWithClass$MainBranch[dataWithClass$MainBranch=="I am not primarily a developer, but I write code so

# Reasigamos el valor
dataWithClass$MainBranch[dataWithClass$MainBranch=="I used to be a developer by profession, but no long

# Reasigamos el valor
dataWithClass$MainBranch[dataWithClass$MainBranch=="I am a student who is learning to code"] <- 'STUDEN

# Reasigamos el valor
dataWithClass$MainBranch[dataWithClass$MainBranch=="I code primarily as a hobby"] <- 'AMATEUR'

# Eliminamos los valores que no tienen clase (NA)
```

```r
dataWithClass <- dataWithClass[!is.na(dataWithClass$MainBranch), ]

#Definimos el total de filas del dataset
filasNS=dim(dataWithClass)[1]

table(as.factor(dataWithClass$MainBranch))
```

```
##
##      AMATEUR      NOT_PRO          PRO PRO_RETIRED      STUDENT
##           83          294         2358           49          275
```

```r
plot(as.factor(dataWithClass$MainBranch))
```



Tratamiento de valores nulos y categóricos

```r
dataNS <- dataWithClass[, 2:11]

# Validamos que hay valores NA
colSums(is.na(dataNS))
```

```
##                      Age            ConvertedComp       DatabaseWorkedWith
```

```
##                     764                    1170                     605
##       LanguageWorkedWith       MiscTechWorkedWith NEWCollabToolsWorkedWith
##                     351                    1085                     525
##       PlatformWorkedWith       WebframeWorkedWith             WorkWeekHrs
##                     489                     865                     960
##               YearsCode
##                     332
```

```
# Rellanamos los campos con NA con valores medios de cada variable
dataNS$Age[is.na(dataNS$Age)] <- mean(dataNS$Age,na.rm=T)

dataNS$ConvertedComp[is.na(dataNS$ConvertedComp)] <- mean(dataNS$ConvertedComp,na.rm=T)

dataNS$WorkWeekHrs[is.na(dataNS$WorkWeekHrs)] <- mean(dataNS$WorkWeekHrs,na.rm=T)

# Convertimos los valores categoricos en numéricos para la variable YearsCode
dataNS$YearsCode[dataNS$YearsCode=="More than 50 years"] <- 50
dataNS$YearsCode[dataNS$YearsCode=="Less than 1 year"] <- 1

# Finalmente convertimos dicha columna en númerica
dataNS$YearsCode <- as.numeric(dataNS$YearsCode)

# Llenamos con la media los valores faltantes
dataNS$YearsCode[is.na(dataNS$YearsCode)] <- mean(dataNS$YearsCode, na.rm=T)

head(dataNS)
```

```
##          Age ConvertedComp
## 44  32.00000      55893.00
## 59  38.00000      16488.00
## 189 29.97712      27794.04
## 240 28.00000       8244.00
## 290 18.00000      27794.04
## 309 27.00000       8712.00
##                                          DatabaseWorkedWith
## 44                                          Microsoft SQL Server
## 59                Microsoft SQL Server;MySQL;PostgreSQL;SQLite
## 189                                        MySQL;Oracle;SQLite
## 240                                                       <NA>
## 290                                        MongoDB;MySQL;SQLite
## 309 DynamoDB;Firebase;MariaDB;Microsoft SQL Server;MySQL;SQLite
##                                                                LanguageWorkedWith
## 44                                                         HTML/CSS;Python;R;SQL;VBA
## 59                                                 C#;HTML/CSS;JavaScript;PHP;Ruby
## 189                                 Bash/Shell/PowerShell;HTML/CSS;Java;JavaScript
## 240                                                        HTML/CSS;JavaScript
## 290                                                                    C#;Python
## 309 Bash/Shell/PowerShell;C#;Dart;HTML/CSS;JavaScript;Kotlin;Python;Swift;TypeScript
##                                          MiscTechWorkedWith
## 44                                                      <NA>
## 59                                        .NET;Node.js;Unity 3D
## 189                                            Cordova;Node.js
## 240                                         React Native;Xamarin
## 290                                               .NET;.NET Core
```

```
## 309 .NET Core;Cordova;Flutter;Node.js;Pandas;Unity 3D;Unreal Engine
##                                                 NEWCollabToolsWorkedWith
## 44                                                   Github;Slack;Trello
## 59                                                   Github;Slack;Trello
## 189 Confluence;Jira;Github;Gitlab;Slack;Google Suite (Docs, Meet, etc)
## 240                                                             Github
## 290                                                    Github;Gitlab
## 309                                       Github;Slack;Microsoft Azure
##                                     PlatformWorkedWith
## 44                                             Windows
## 59                                          Docker;MacOS
## 189                              Android;AWS;Docker;Linux
## 240                                 Android;Windows
## 290        Android;Arduino;Docker;Heroku;Linux;Windows
## 309 Android;Linux;Microsoft Azure;Raspberry Pi;Windows
##                         WebframeWorkedWith WorkWeekHrs YearsCode
## 44                                    <NA>    45.00000        10
## 59  ASP.NET;jQuery;Laravel;Ruby on Rails    40.00000        24
## 189                               React.js    40.00000        20
## 240                                 jQuery    44.00000         5
## 290                                  Flask    39.01358         5
## 309                  ASP.NET Core;Express    40.00000        10
```

Ahora vamos a agregar nuevas variables que contabilizan el número de tecnologías o herramientas de: bases de datos, lenguajes de programación, de colaboración, entre otros. Primero para la base de datos, vamos a usar la columna *DatabaseWorkedWith*. La variable a crearse será **db_techs**:

```
dataNS$db_techs <- 0

for(i in 1:filasNS) {
  if (is.na(dataNS$DatabaseWorkedWith[i])) {
    dataNS$db_techs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(dataNS$DatabaseWorkedWith[i], ";"), length)
    dataNS$db_techs[i] <- longitud
  }
}
```

Para agregar una nueva variable que represente el número de lenguajes de programación que usa. Este dato se basa en la experiencia ya adquirida y no en los deseos para usar o aprender el siguiente año. Para esto usaremos la columna *LanguageWorkedWith*. La variable a crearse será **prog_langs**:

```
dataNS$prog_langs <- 0

for(i in 1:filasNS) {
  if (is.na(dataNS$LanguageWorkedWith[i])) {
    dataNS$prog_langs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(dataNS$LanguageWorkedWith[i], ";"), length)
    dataNS$prog_langs[i] <- longitud
  }
}
```

Ahora vamos a agregar una nueva variable para el número de frameworks, librerías y demás herramientras que usa el desarrollador. Este dato se basa en la experiencia ya adquirida y no en los deseos para usar o

aprender el siguiente año. Para esto usaremos la columna *MiscTechWorkedWith*. La variable a crearse será **misc_techs**:

```
dataNS$misc_techs <- 0

for(i in 1:filasNS) {
  if (is.na(dataNS$MiscTechWorkedWith[i])) {
    dataNS$misc_techs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(dataNS$MiscTechWorkedWith[i], ";"), length)
    dataNS$misc_techs[i] <- longitud
  }
}
```

Haremos lo mismo para el número de herramientras colaborativas que usa el desarrollador, según el contenido de la columna *NEWCollabToolsWorkedWith*. Este dato se basa en la experiencia ya adquirida y no en los deseos para usar o aprender el siguiente año. La variable a crearse será **collab_techs**:

```
dataNS$collab_techs <- 0

for(i in 1:filasNS) {
  if (is.na(dataNS$NEWCollabToolsWorkedWith[i])) {
    dataNS$collab_techs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(dataNS$NEWCollabToolsWorkedWith[i], ";"), length)
    dataNS$collab_techs[i] <- longitud
  }
}
```

También vamos a agregar una variable para el número de plataformas que usa el desarrollador. Este dato se basa en la experiencia ya adquirida y no en los deseos para usar o aprender el siguiente año. Usaremos el contenido de la columna *PlatformWorkedWith*. La variable a crearse será **plat_techs**:

```
dataNS$plat_techs <- 0

for(i in 1:filasNS) {
  if (is.na(dataNS$PlatformWorkedWith[i])) {
    dataNS$plat_techs[i] <- 0
  } else {
    longitud  <- sapply(strsplit(dataNS$PlatformWorkedWith[i], ";"), length)
    dataNS$plat_techs[i] <- longitud
  }
}
```

Finalmente, agregaremos una variable para el número de *frameworks web* que usa el desarrollador. Este dato se basa en la experiencia ya adquirida y no en los deseos para usar o aprender el siguiente año. Para esto usaremos la columna *WebframeWorkedWith*. La variable a crearse será **web_techs**:

```
dataNS$web_techs <- 0

for(i in 1:filasNS) {
  if (is.na(dataNS$WebframeWorkedWith[i])) {
    dataNS$web_techs[i] <- 0
```

```
  } else {
    longitud  <- sapply(strsplit(dataNS$WebframeWorkedWith[i], ";"), length)
    dataNS$web_techs[i] <- longitud
  }
}
```

Por último eliminamos las variables categóricas:

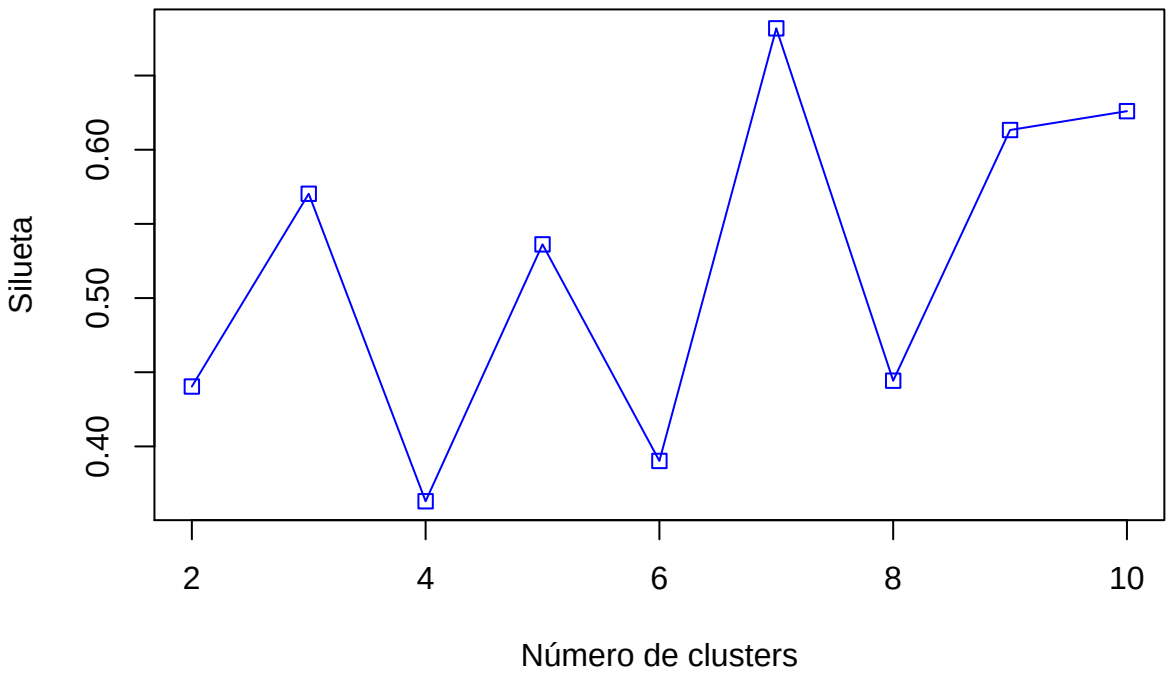
```
dataNS$DatabaseWorkedWith = NULL
dataNS$LanguageWorkedWith = NULL
dataNS$MiscTechWorkedWith = NULL
dataNS$NEWCollabToolsWorkedWith = NULL
dataNS$PlatformWorkedWith = NULL
dataNS$WebframeWorkedWith = NULL
```

Ahora probamos a definir el número óptimo de clusters con la métrica **euclidean**

```
library(cluster) # cargamos la librería que permite trabajar con las funciones de clustering

d <- daisy(dataNS, metric="euclidean")
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(dataNS, i)
  y_cluster     <- fit$cluster
  sk            <- silhouette(y_cluster, d)
  resultados[i] <- mean(sk[,3])
}
```

Mostramos en un gráfica los valores de las siluetas media de cada prueba para comprobar que número de clústers es el mejor

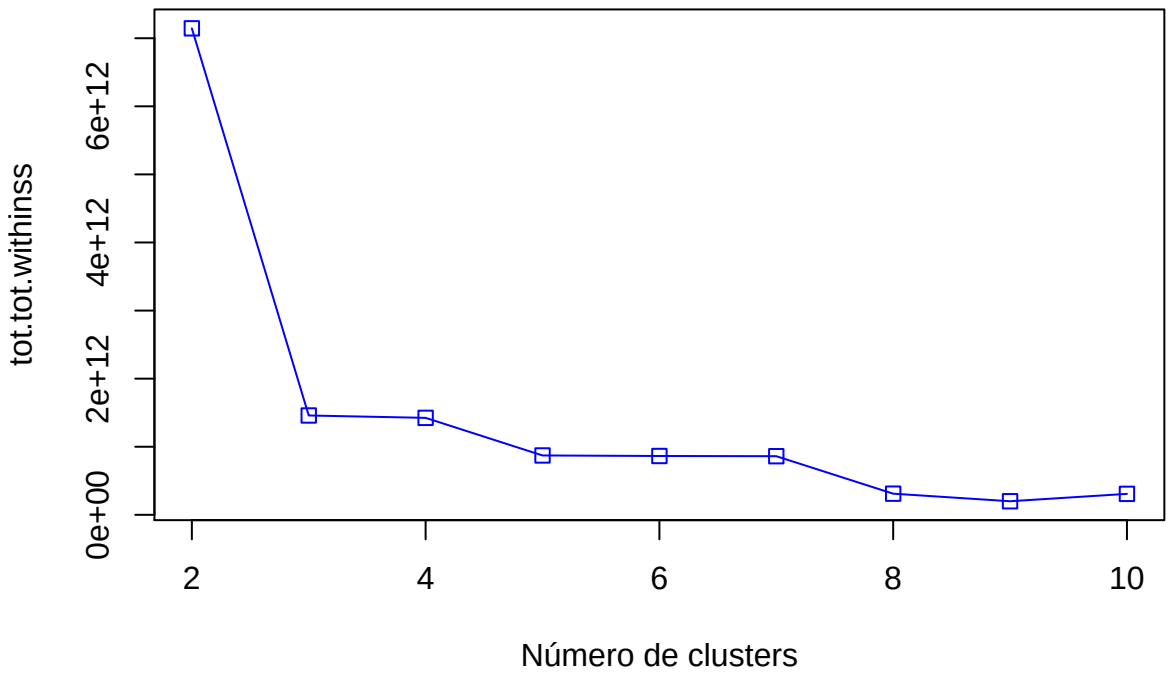
```
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Número de clusters",ylab="Silueta")
```



Según la gráfica anterior, vemos que el número K con mejor puntuación **k=7**, luego le sigue **k=10** y **k=9**, sin embargo al ser un método no supervisado debemos seguir intentando determinar el valor óptimo de K usando otras técnicas.

Otro forma de evaluar cual es el mejor número de clústers es considerar el mejor modelo, aquel que ofrece la menor suma de los cuadrados de las distancias de los puntos de cada grupo con respecto a su centro (withinss), con la mayor separación entre centros de grupos (betweenss). Como se puede comprobar es una idea conceptualmente similar a la silueta. Una manera común de hacer la selección del número de clústers consiste en aplicar el método elbow (codo), que no es más que la selección del número de clústers en base a la inspección de la gráfica que se obtiene al iterar con el mismo conjunto de datos para distintos valores del número de clústers. Se seleccionará el valor que se encuentra en el "codo" de la curva

```
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(dataNS, i)
  resultados[i] <- fit$tot.withinss
}
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Número de clusters",ylab="tot.tot.withinss")
```



Para este caso hemos obtenido un valor promedio de **k=7**, clusters, que es donde la curva ya ha descendido y se ha estabilizado.

También se puede usar la función kmeansruns del paquete fpc que ejecuta el algoritmo kmeans con un conjunto de valores, para después seleccionar el valor del número de clústers que mejor funcione de acuerdo a dos criterios: la silueta media ("asw") y Calinski-Harabasz ("ch").

```
library(fpc)
fit_ch  <- kmeansruns(dataNS, krange = 1:10, criterion = "ch")
fit_asw <- kmeansruns(dataNS, krange = 1:10, criterion = "asw")
```

Podemos comprobar el valor con el que se ha obtenido el mejor resultado y también mostrar el resultado obtenido para todos los valores de k usando ambos criterios

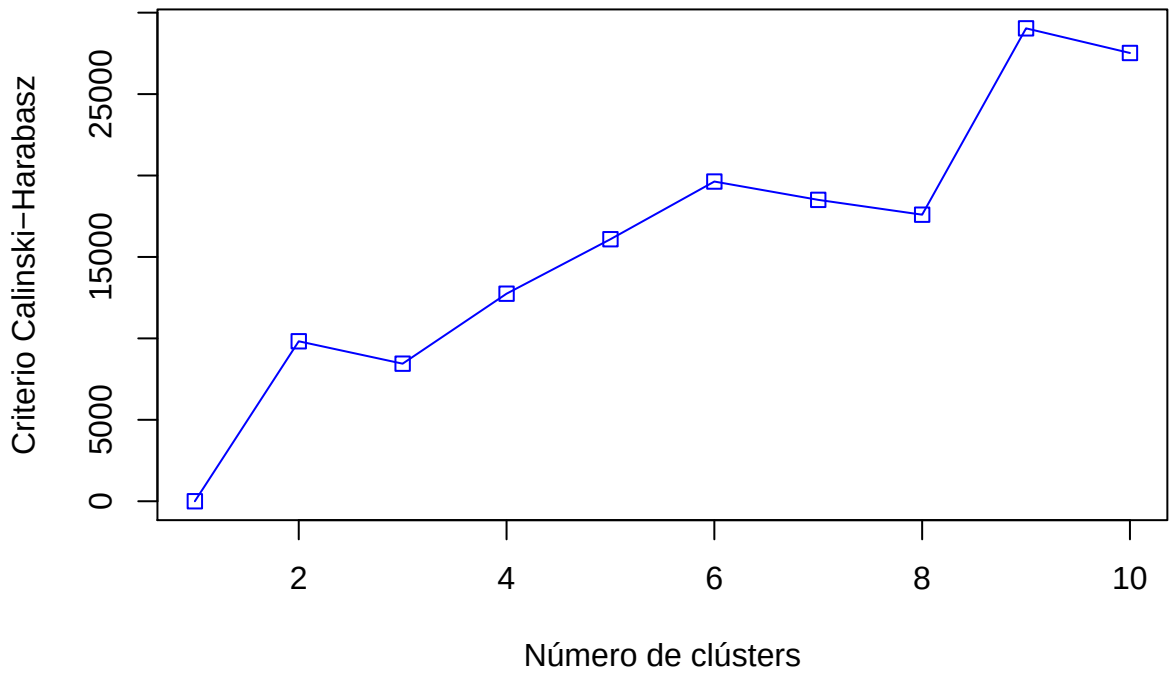
```
fit_ch$bestk
```

```
## [1] 9
```

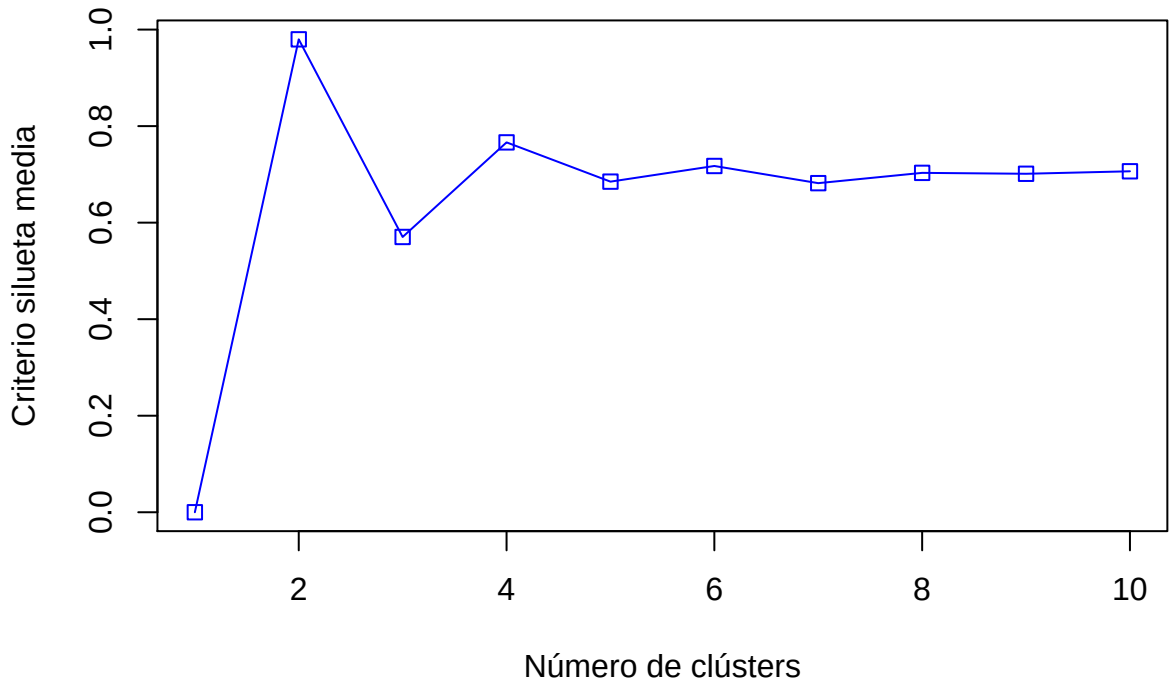
```
fit_asw$bestk
```

```
## [1] 2
```

```
plot(1:10,fit_ch$crit,type="o",col="blue",pch=0,xlab="Número de clústers",ylab="Criterio Calinski-Haraba
```




```
plot(1:10,fit_asw$crit,type="o",col="blue",pch=0,xlab="Número de clústers",ylab="Criterio silueta media"
```

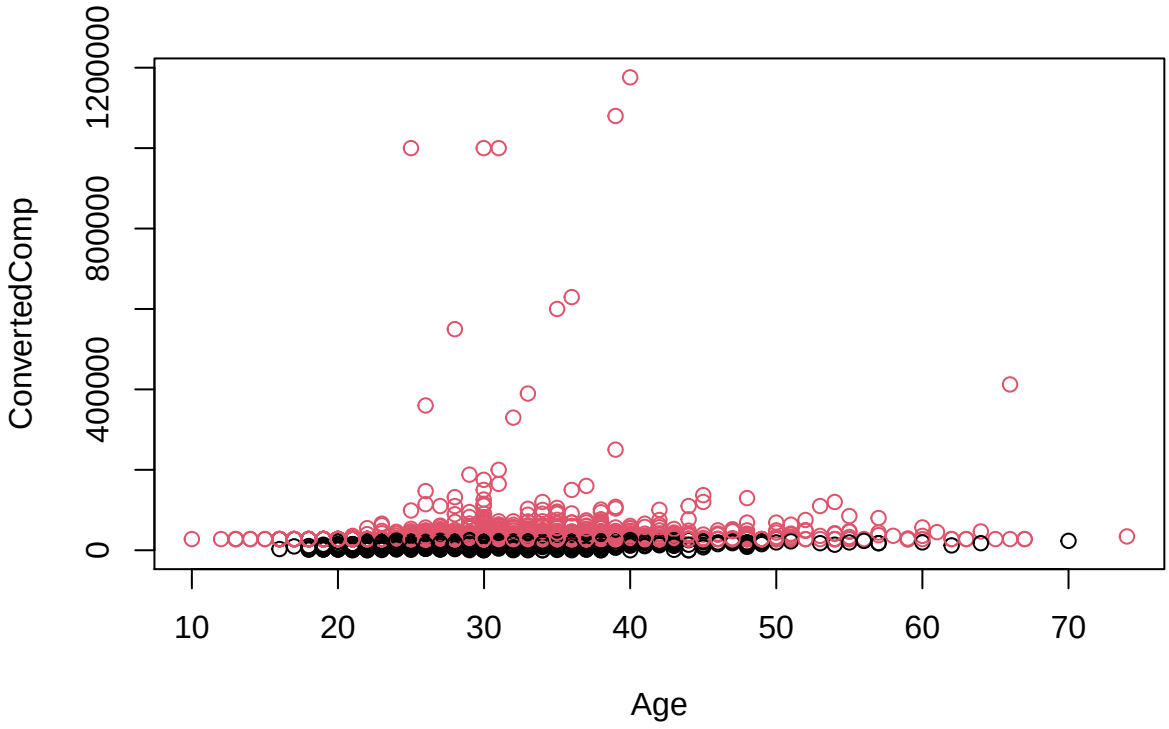


Los resultados obtenidos son muy diferentes. Para el caso de las siluetas medias obtuvimos *2* y luego con el método elbow (codo) obtuvimos *6*. Finalmente usando la función kmeansruns y aplicando los criterios Calinski-Harabasz y silueta media, se obtiene *9* y *2*, respectivamente. Notamos que el valor k=2 coincide con el obtenido inicialmente.

Ahora vamos a analizar los datos de manera visual, comparandolos de 2 en 2, con el valor real que sabemos está almacenado en el campo "MainBranch" del dataset original.

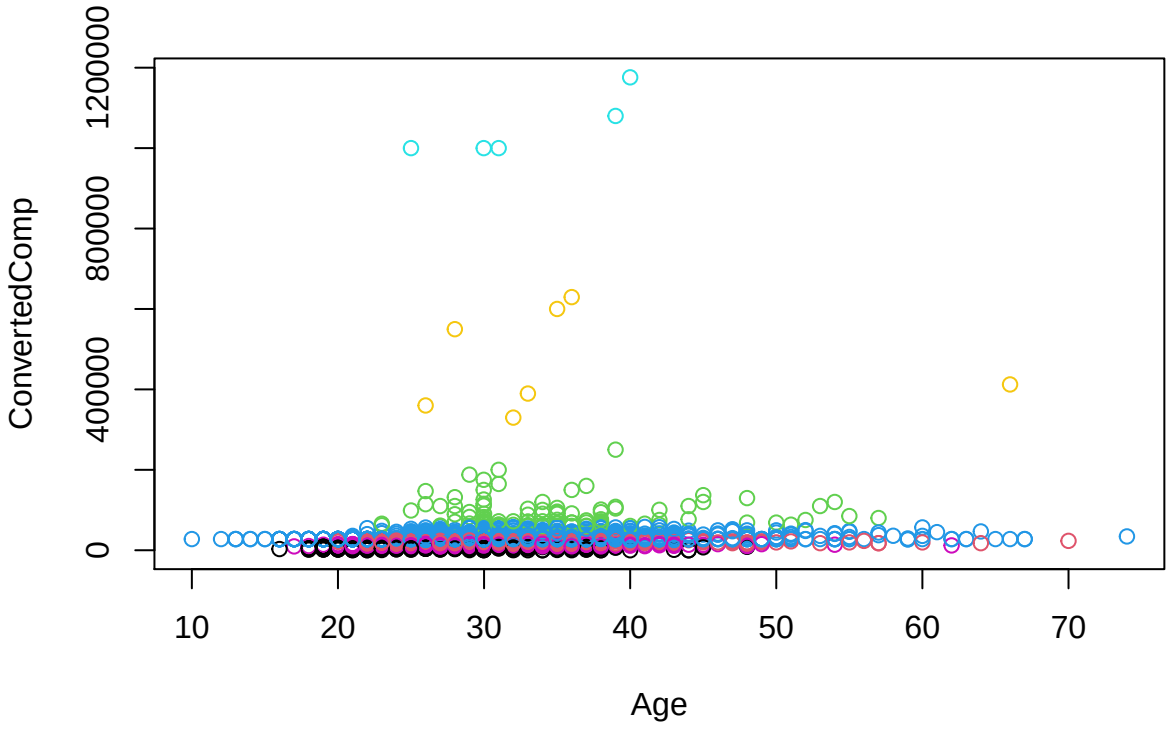
Vamos a analizar primero con el valor de k=2, para el par de atributos *Age* (Edad) y *ConvertedComp* (Salario Anual en dolares):

```
dev2clusters <- kmeans(dataNS, 2)

# Edad y Salario Anual
plot(dataNS[c(1,2)], col=dev2clusters$cluster)
```

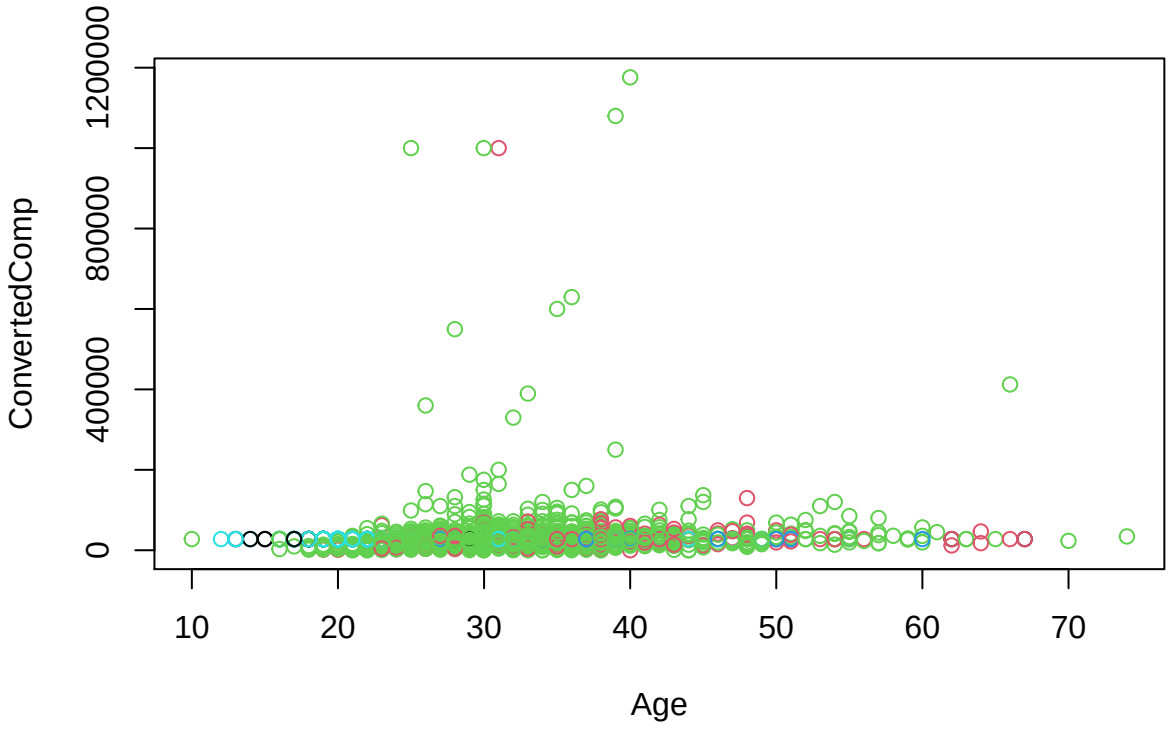


Vamos a analizar primero con el valor de k=7, para el par de atributos *Age* (Edad) y *ConvertedComp* (Salario Anual en dolares):

```
dev7clusters <- kmeans(dataNS, 7)

# Edad y Salario Anual
plot(dataNS[c(1,2)], col=dev7clusters$cluster)
```

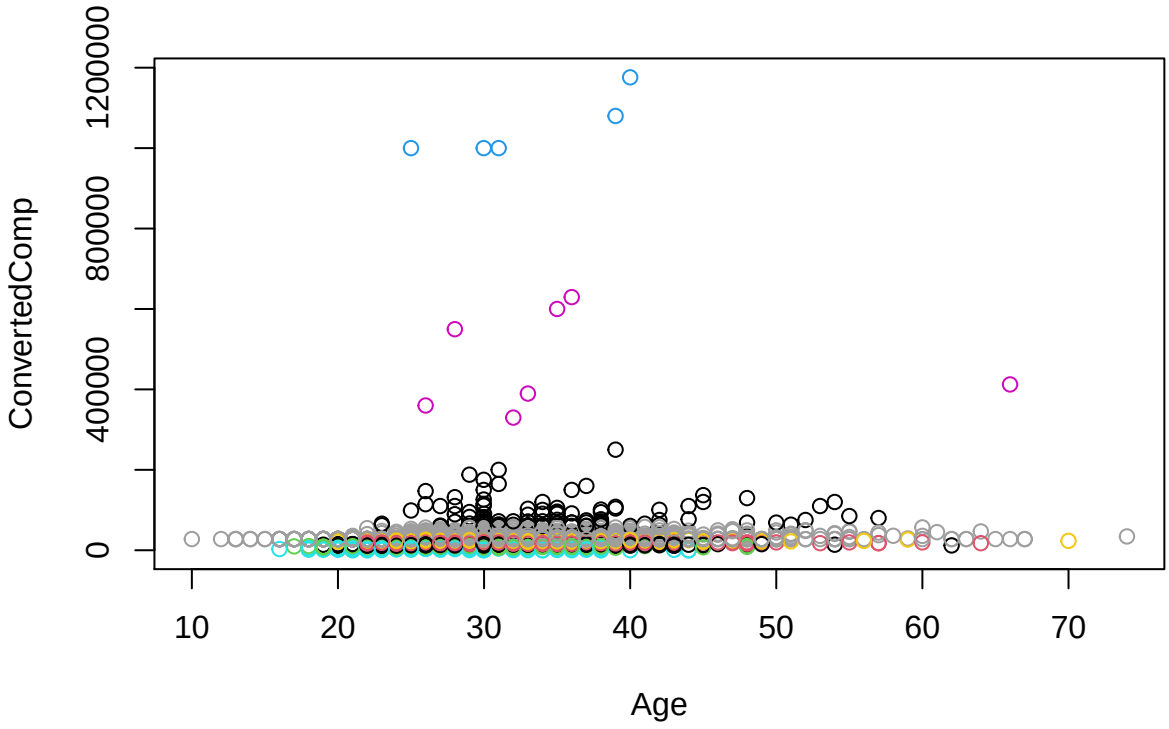



```
plot(dataNS[c(1,2)], col=as.factor(dataWithClass$MainBranch))
```

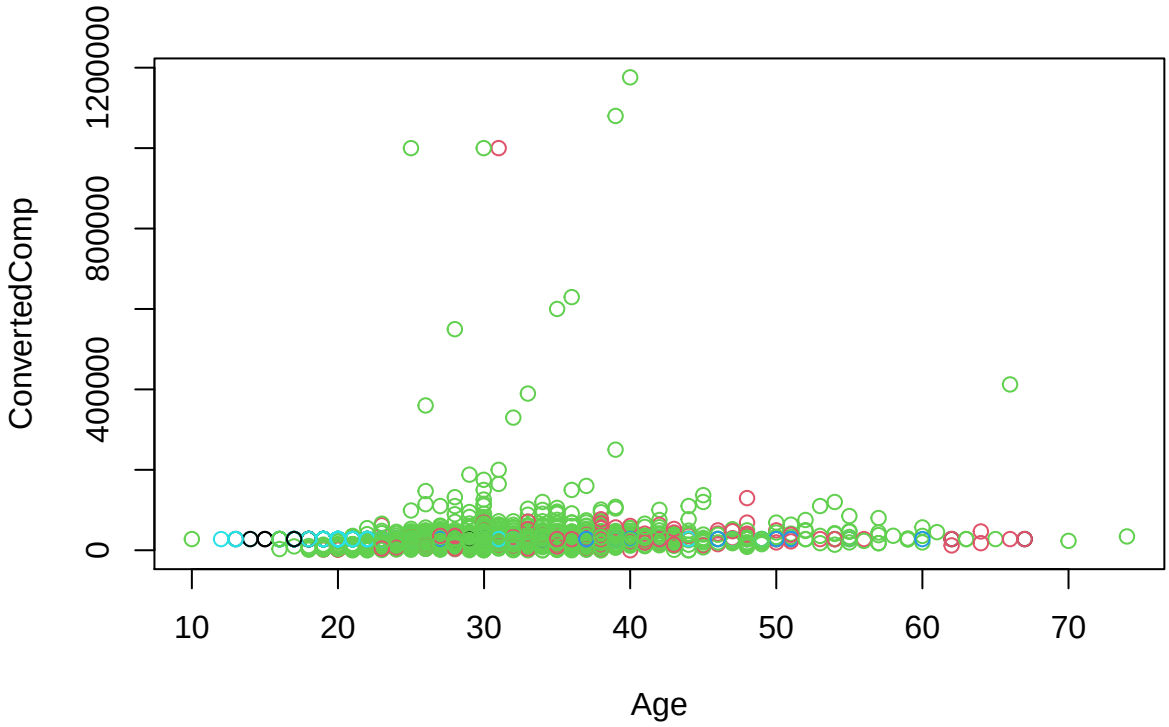


Vamos a analizar primero con el valor de k=9, para el par de atributos *Age* (Edad) y *ConvertedComp* (Salario Anual en dolares):

```
dev9clusters <- kmeans(dataNS, 9)

# Edad y Salario Anual
plot(dataNS[c(1,2)], col=dev9clusters$cluster)
```




```
plot(dataNS[c(1,2)], col=as.factor(dataWithClass$MainBranch))
```

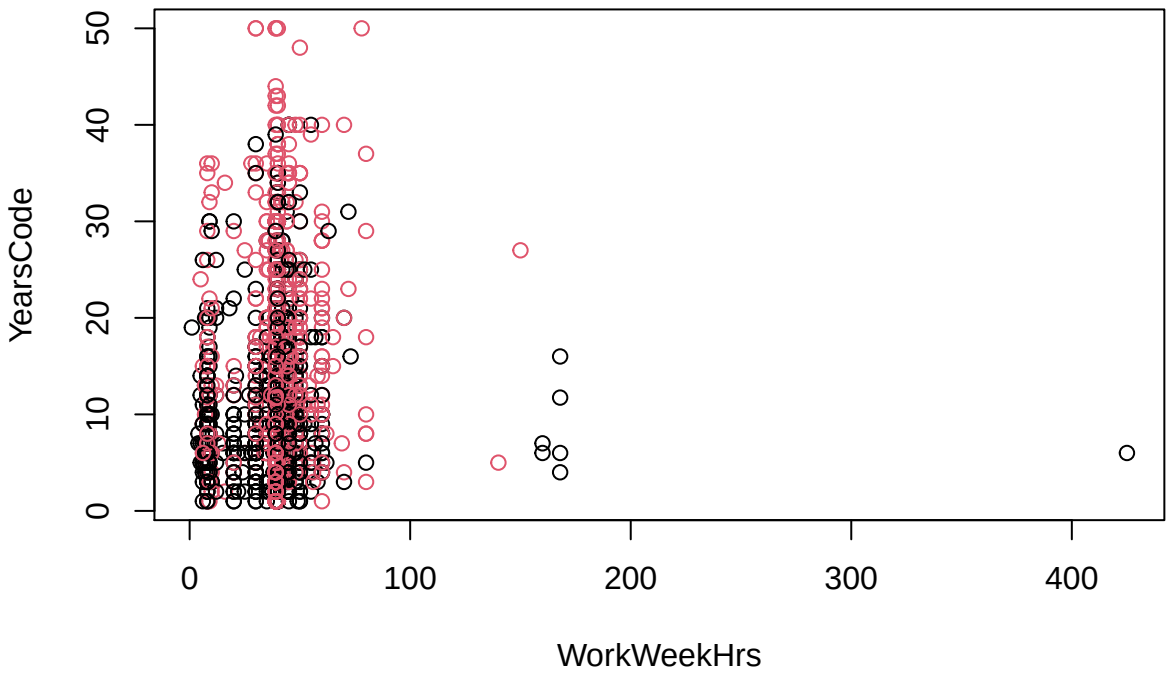


Vemos que a medida que se aumenta el valor de k se entremezclan más los grupos, por tal motivo no es una buena elección haber elegido la edad y el salario anual como variables para agrupar. Es razonable, ya que el salario anual depende de mucho factores, no necesariamente con la edad, pero será necesario analizarlo con otros atributos de los individuos.

Ahora procedamos a analizar los clusters WorkWeekHrs (Horas de trabajo) vs. YearsCode (años de experiencia) y los mismos valores de k.

Para k=2:

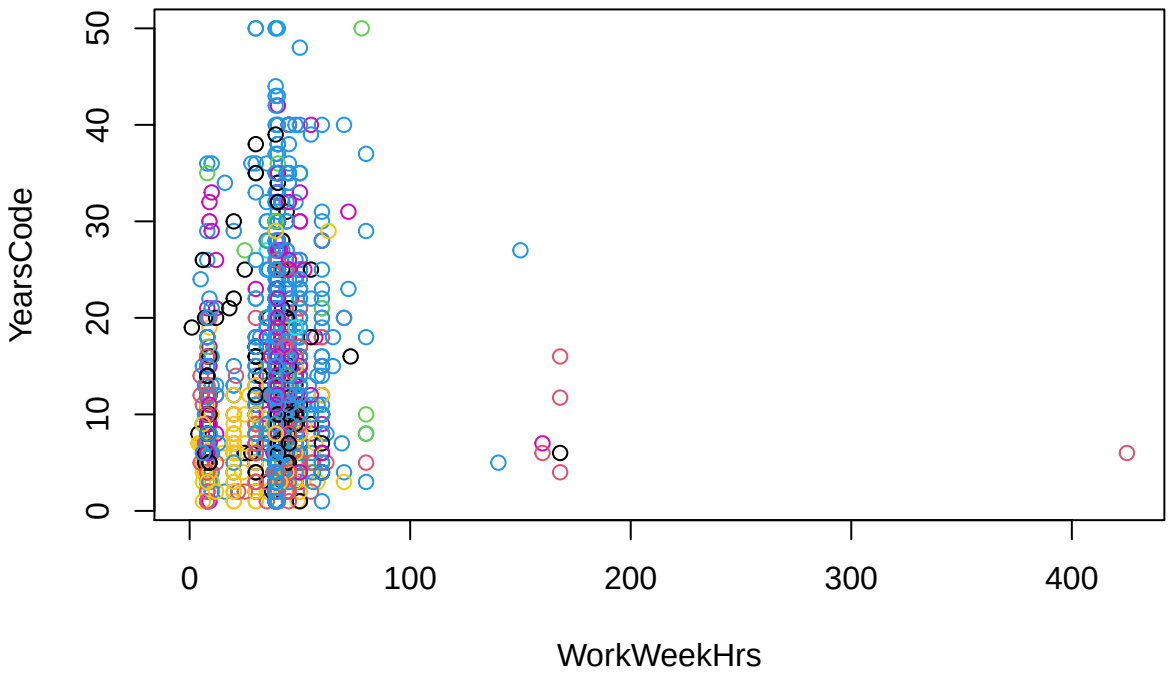
```
# WorkWeekHrs y YearsCode
dev2HYclust <- kmeans(dataNS, 2)

plot(dataNS[c(3,4)], col=dev2HYclust$cluster)
```



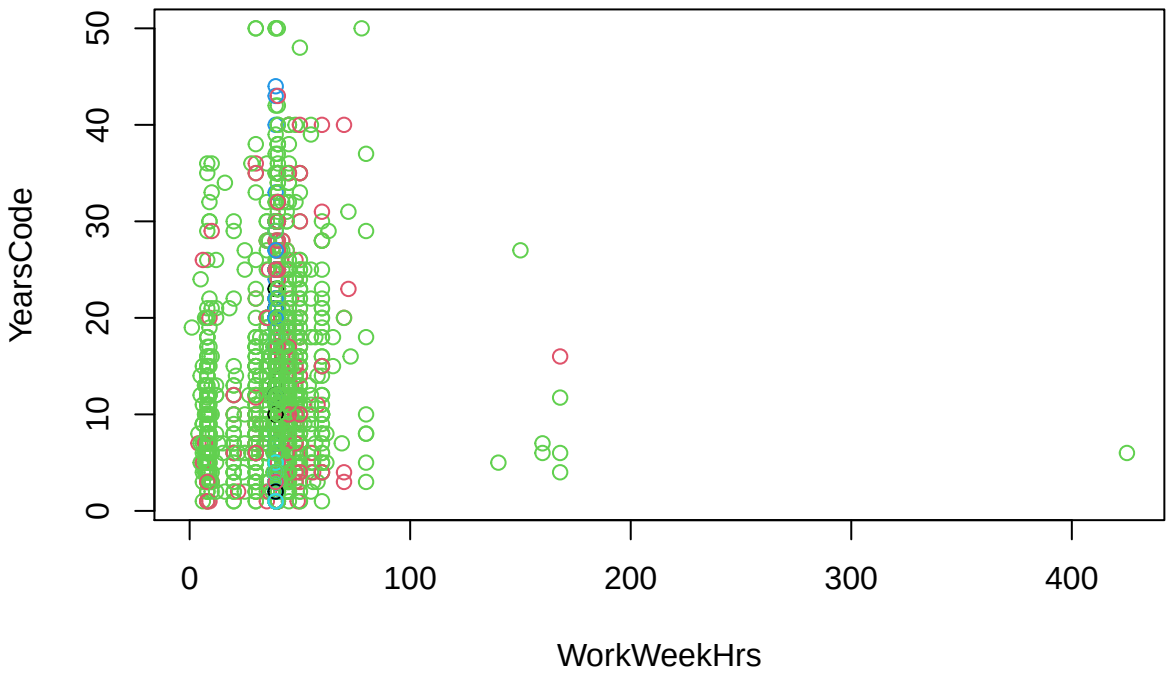
Vemos que para el análisis entre las Horas de trabajo y los años de experiencia, para el valor de k=2, no se obtiene grupos claramente definidos.

Ahora para k=7:

```
# WorkWeekHrs y YearsCode
dev7HYclust <- kmeans(dataNS, 7)

plot(dataNS[c(3,4)], col=dev7HYclust$cluster)
```



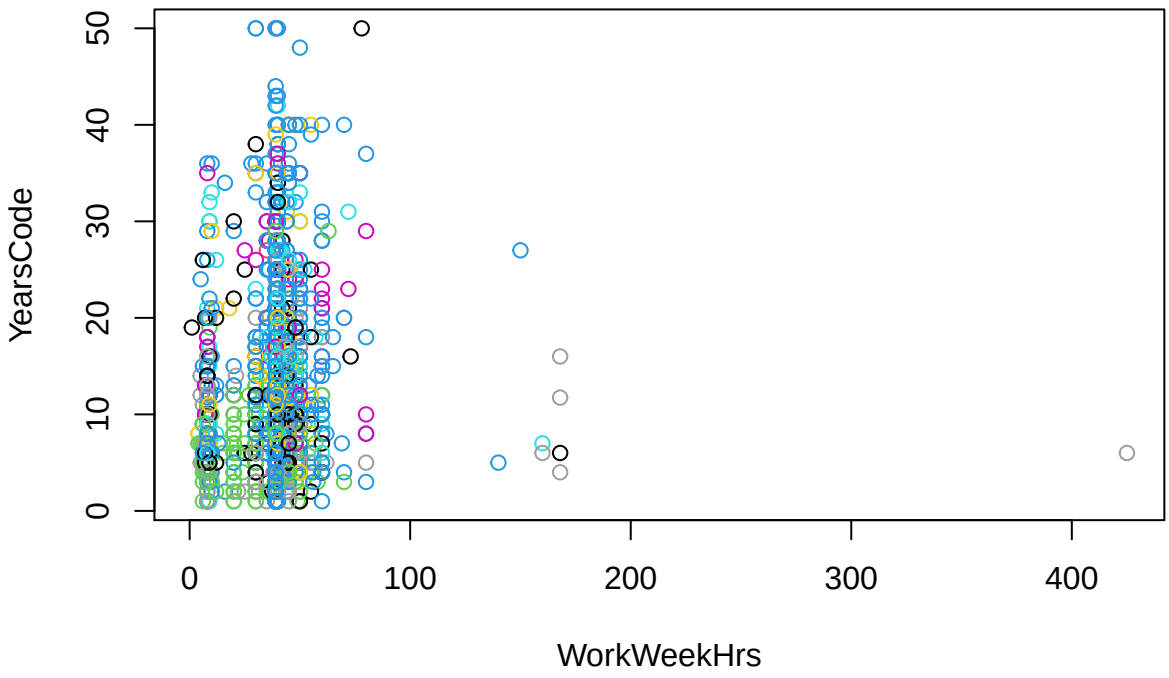

```
plot(dataNS[c(3,4)], col=as.factor(dataWithClass$MainBranch))
```



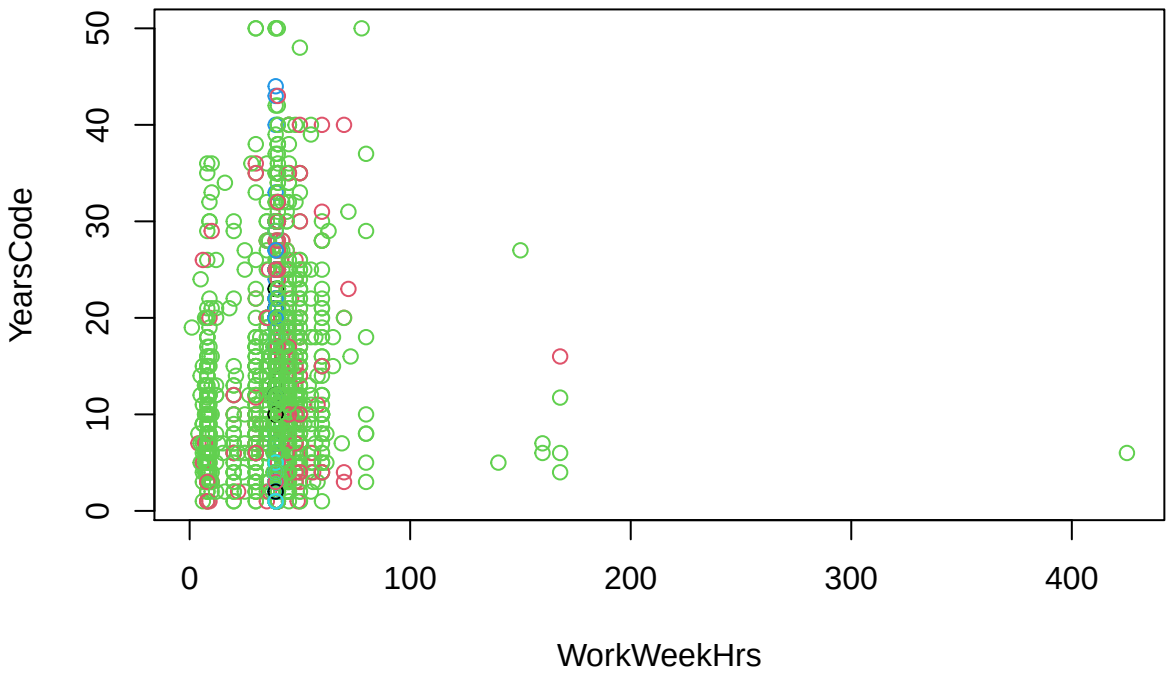
Vemos que para las 2 variables analizadas, el valor de k=7, no define los clusters claramente.

Ahora para k=9:

```
# WorkWeekHrs y YearsCode
dev9HYclust <- kmeans(dataNS, 9)

plot(dataNS[c(3,4)], col=dev9HYclust$cluster)
```




```
plot(dataNS[c(3,4)], col=as.factor(dataWithClass$MainBranch))
```

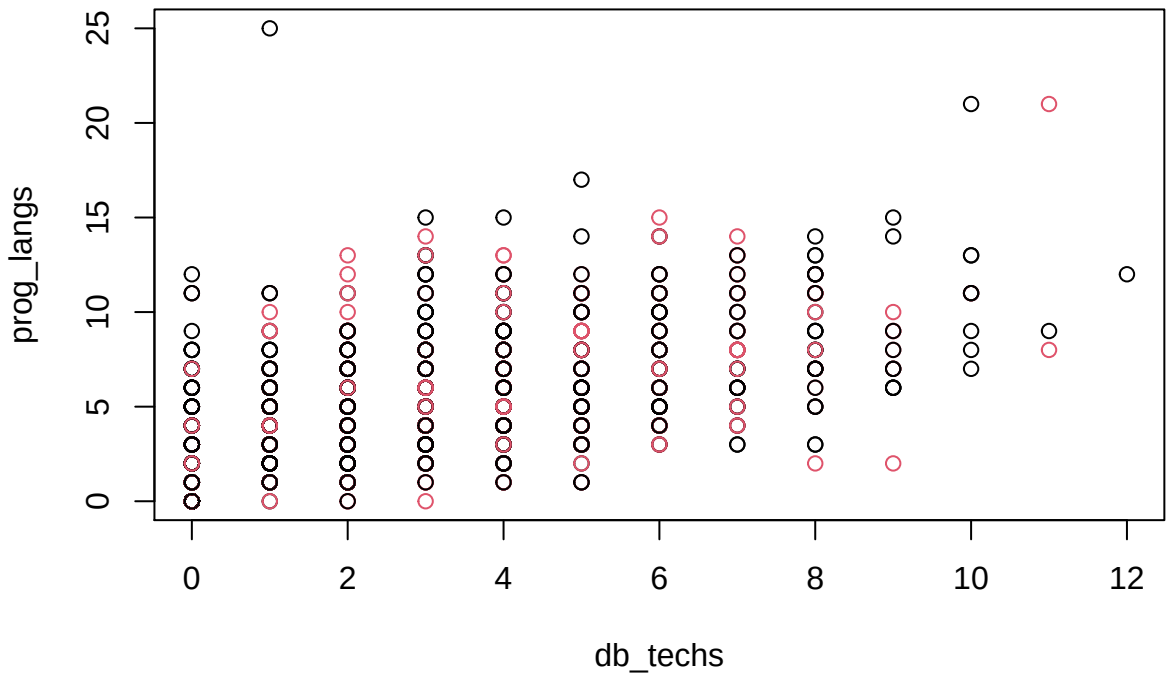


Para estas 2 variables seleccionadas tampoco se aplica de manera optima la agrupación por clusters, no coinciden con los valores los grupos de la variable clase. Es lógico que ambas variables no vayan a definir claramente los clusters ya que no guardan relación.

Ahora procedamos a analizar los clusters db_techs (Tecnologías de BD usadas) vs. prog_langs (lenguajes de programación) y los mismos valores de k.

Para k=2:

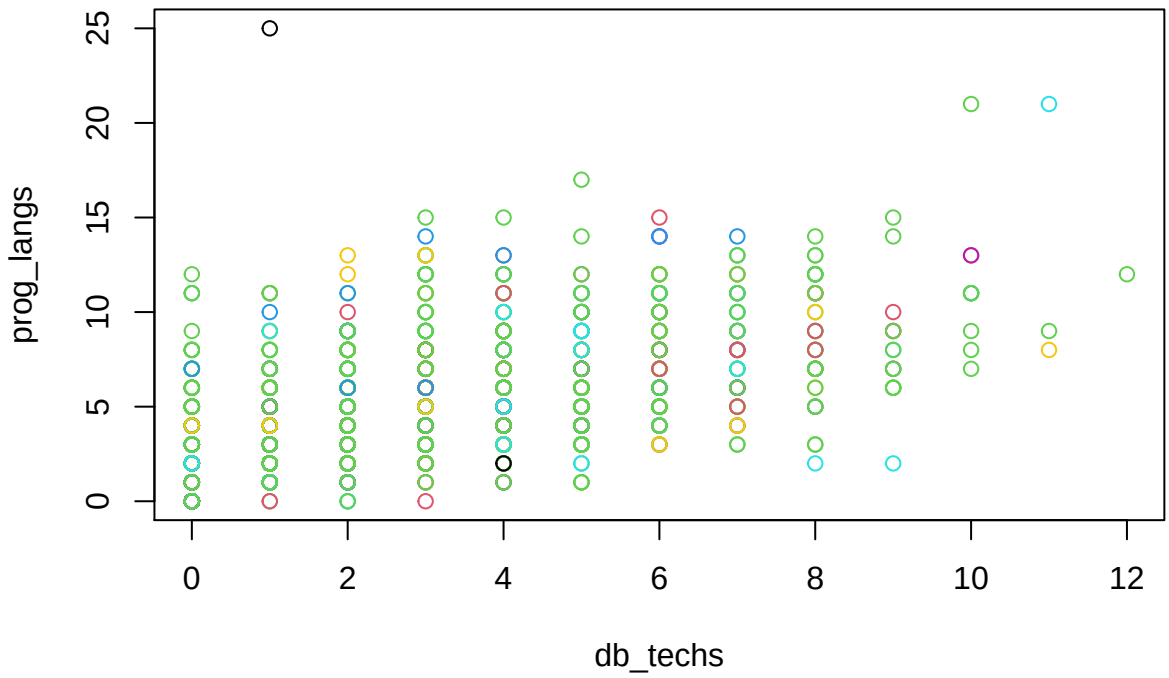
```
# db_techs y prog_langs
dev2DPclust <- kmeans(dataNS, 2)

plot(dataNS[c(5,6)], col=dev2DPclust$cluster)
```



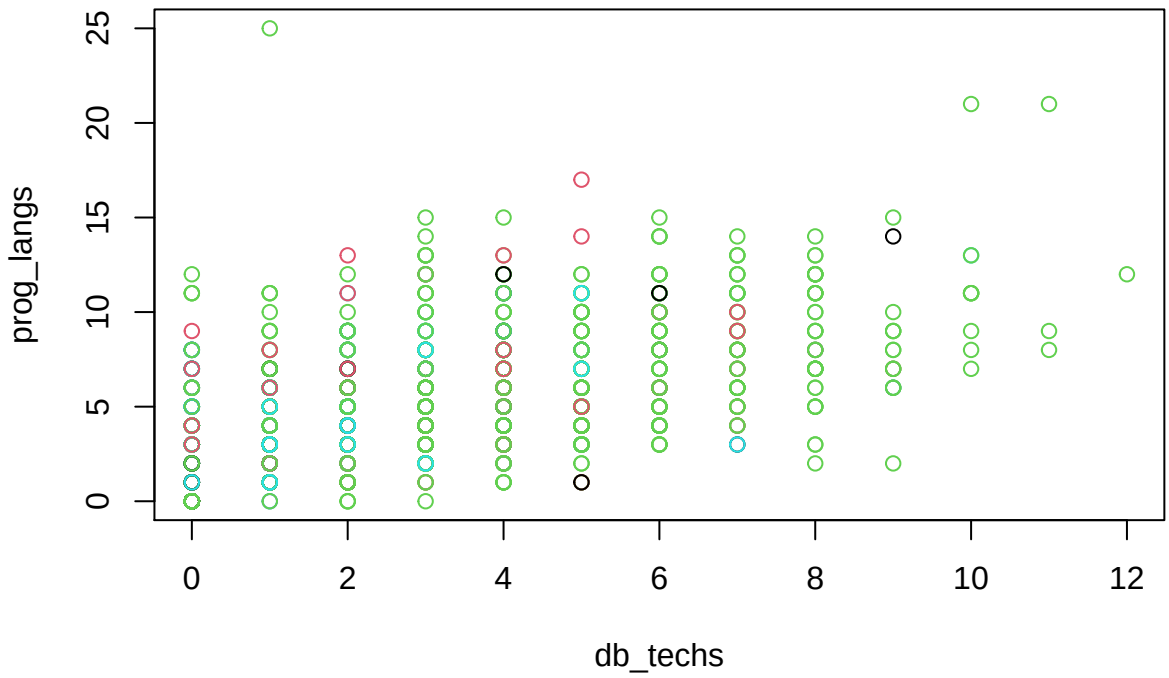
Vemos que los clusters no están definidos. Los puntos están muy dispersos en relación con las comparaciones de 2 en 2 con las primeras 4 variables del dataset.

Ahora para k=7:

```
# db_techs y prog_langs
dev7DPclust <- kmeans(dataNS, 7)

plot(dataNS[c(5,6)], col=dev7DPclust$cluster)
```



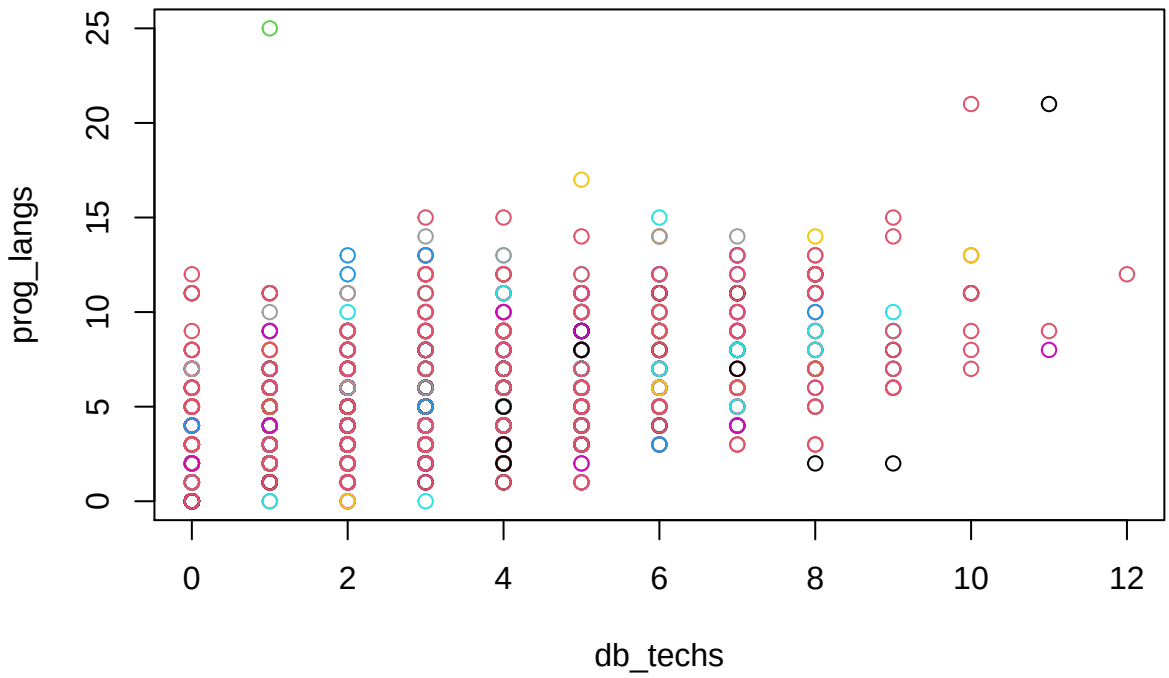

```
plot(dataNS[c(5,6)], col=as.factor(dataWithClass$MainBranch))
```



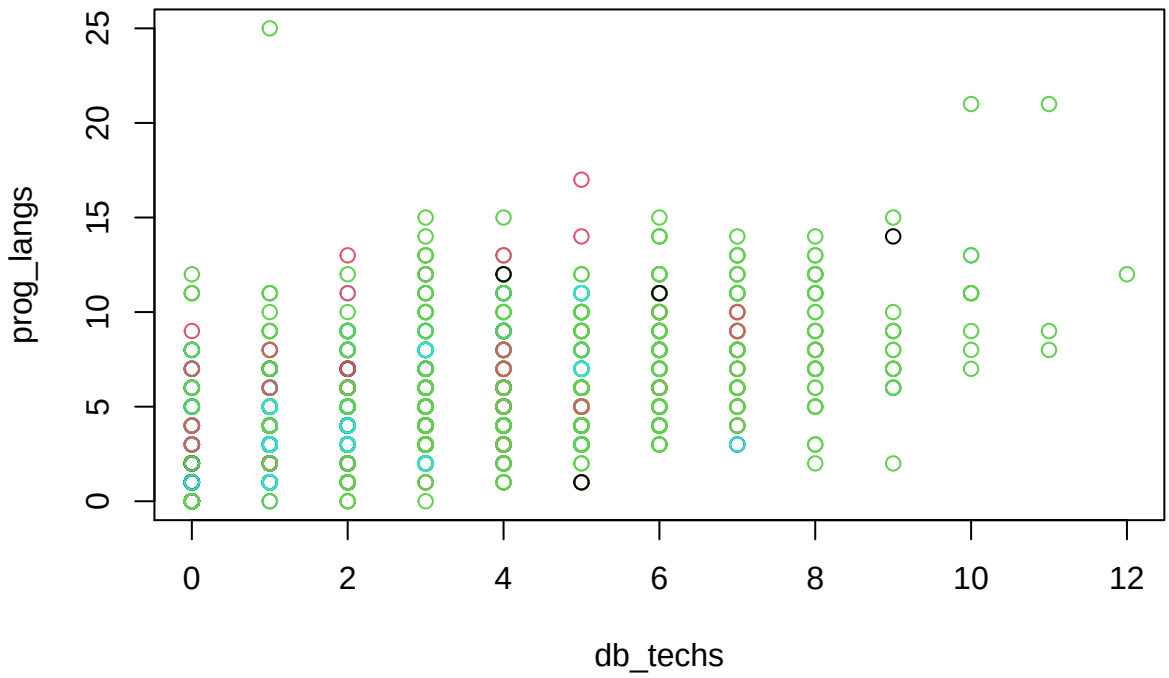
De igual manera que para k=2, vemos que los clusters no están definidos. Los puntos están muy dispersos.

Ahora para k=9:

```
# db_techs y prog_langs
dev9DPclust <- kmeans(dataNS, 9)

plot(dataNS[c(5,6)], col=dev9DPclust$cluster)
```




```
plot(dataNS[c(5,6)], col=as.factor(dataWithClass$MainBranch))
```

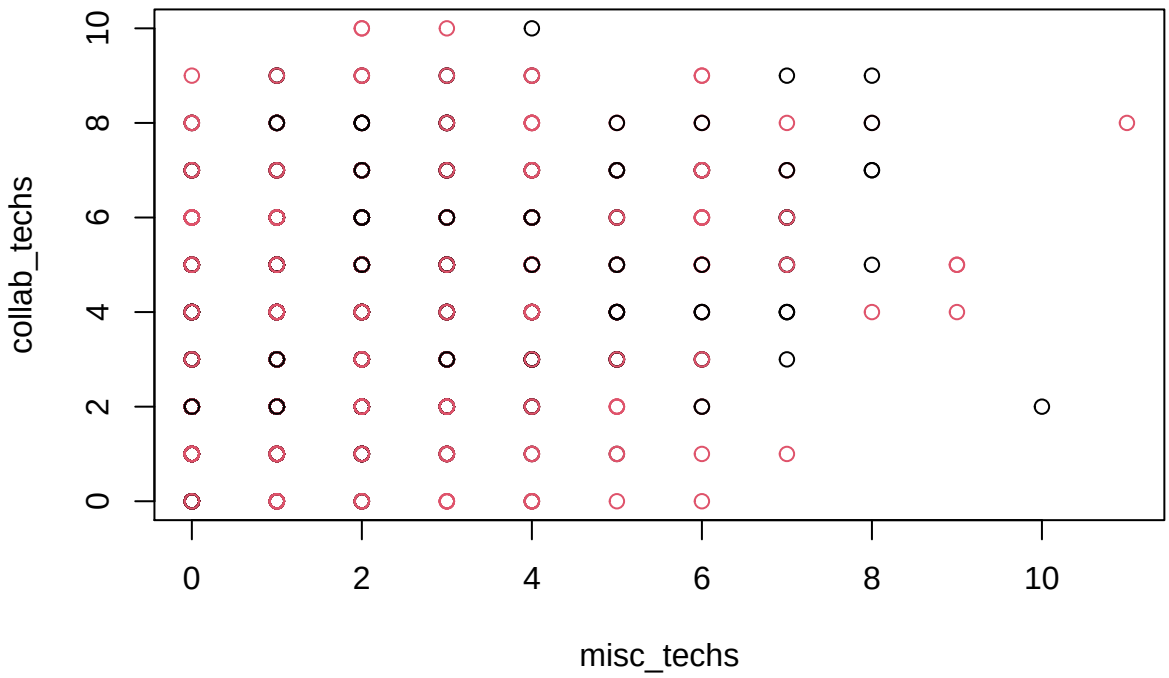


La elección de estas 2 variables no aportan luz para la agrupación de los datos en los clusters de 2, 7 y 9.

Ahora procedamos a analizar los clusters misc_techs (Tecnologías varias) vs. collab_techs (herramientas de colaboración) y los mismos valores de k.

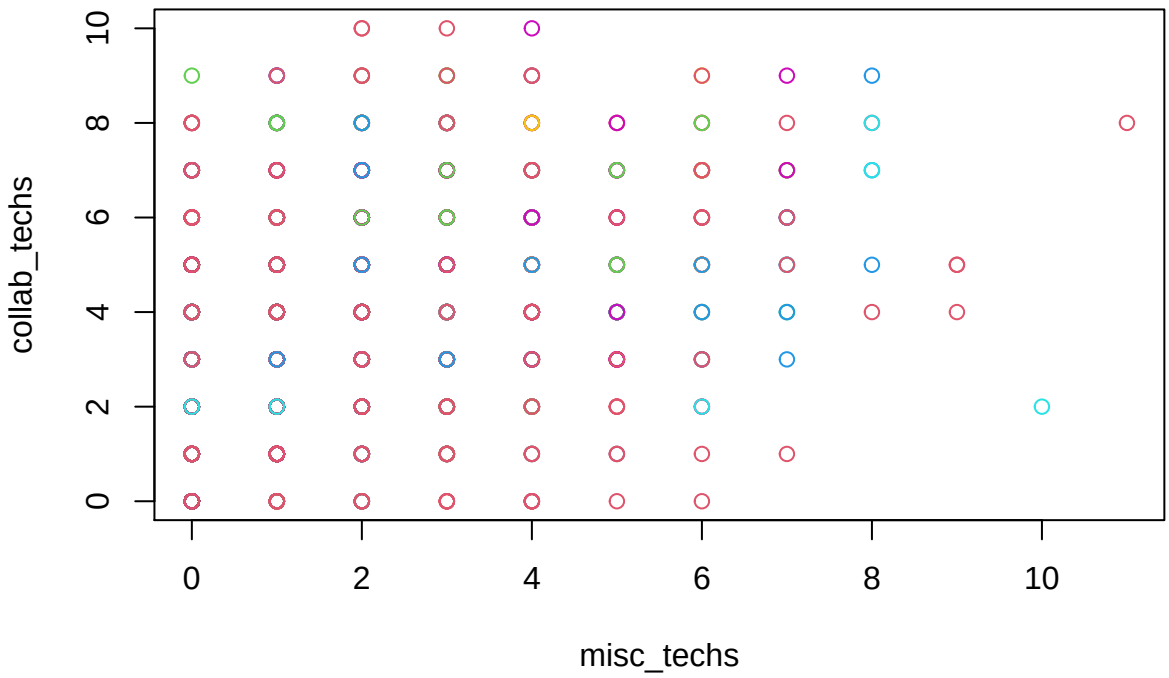
Para k=2:

```
# misc_techs y collab_techs
dev2MCclust <- kmeans(dataNS, 2)

plot(dataNS[c(7,8)], col=dev2MCclust$cluster)
```

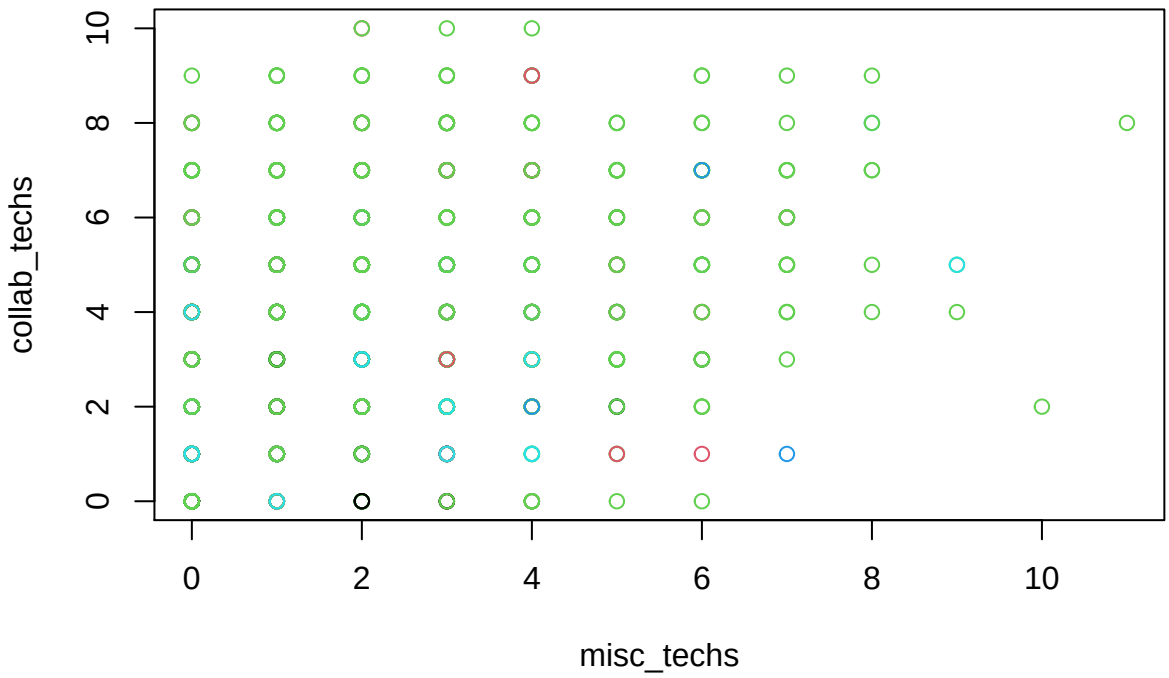


Ahora para k=7:

```
# misc_techs y collab_techs
dev7MCclust <- kmeans(dataNS, 7)

plot(dataNS[c(7,8)], col=dev7MCclust$cluster)
```

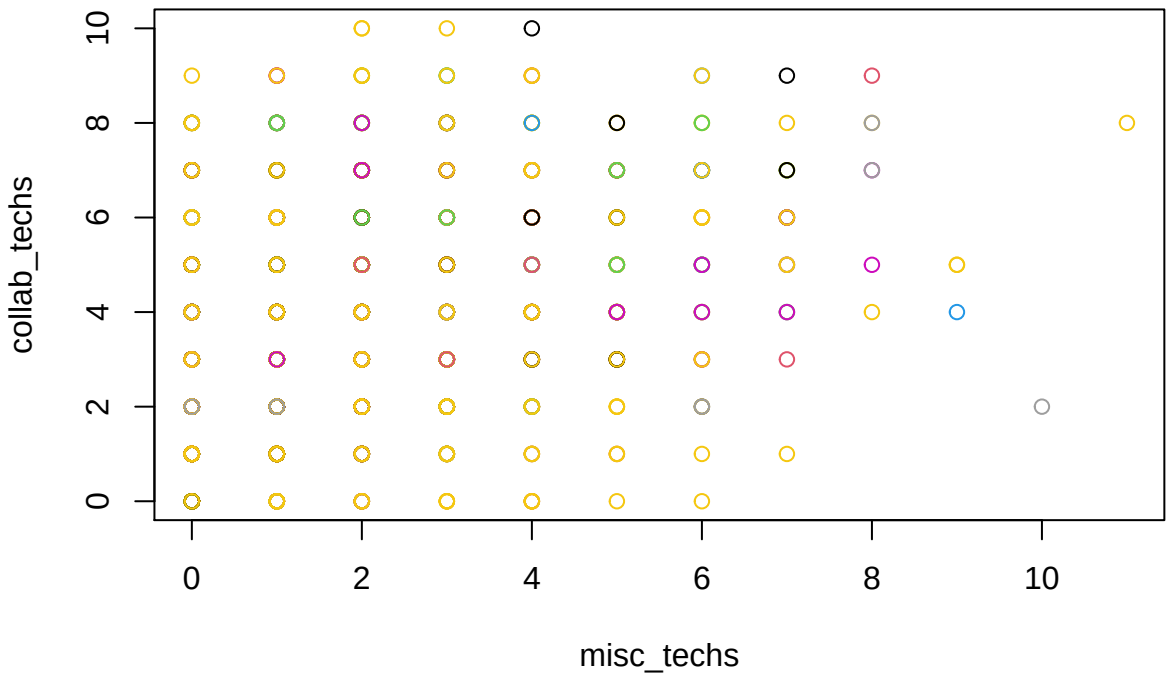



```
plot(dataNS[c(7,8)], col=as.factor(dataWithClass$MainBranch))
```

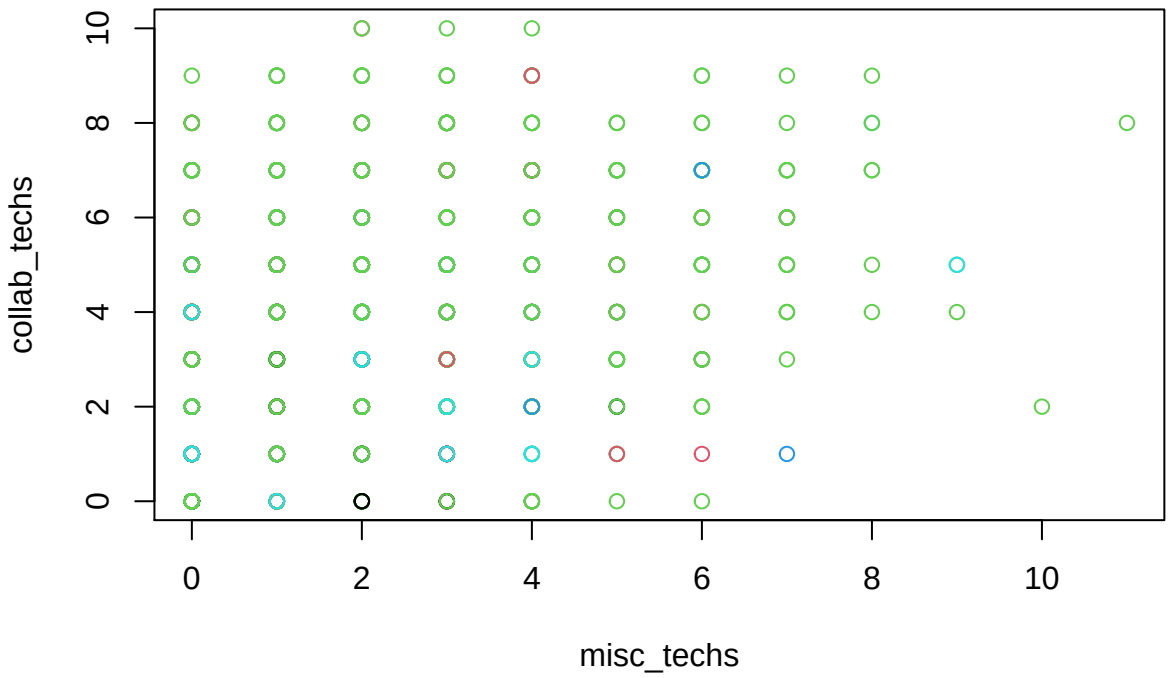


Ahora para k=9:

```
# misc_techs y collab_techs
dev9MCclust <- kmeans(dataNS, 9)

plot(dataNS[c(7,8)], col=dev9MCclust$cluster)
```




```
plot(dataNS[c(7,8)], col=as.factor(dataWithClass$MainBranch))
```

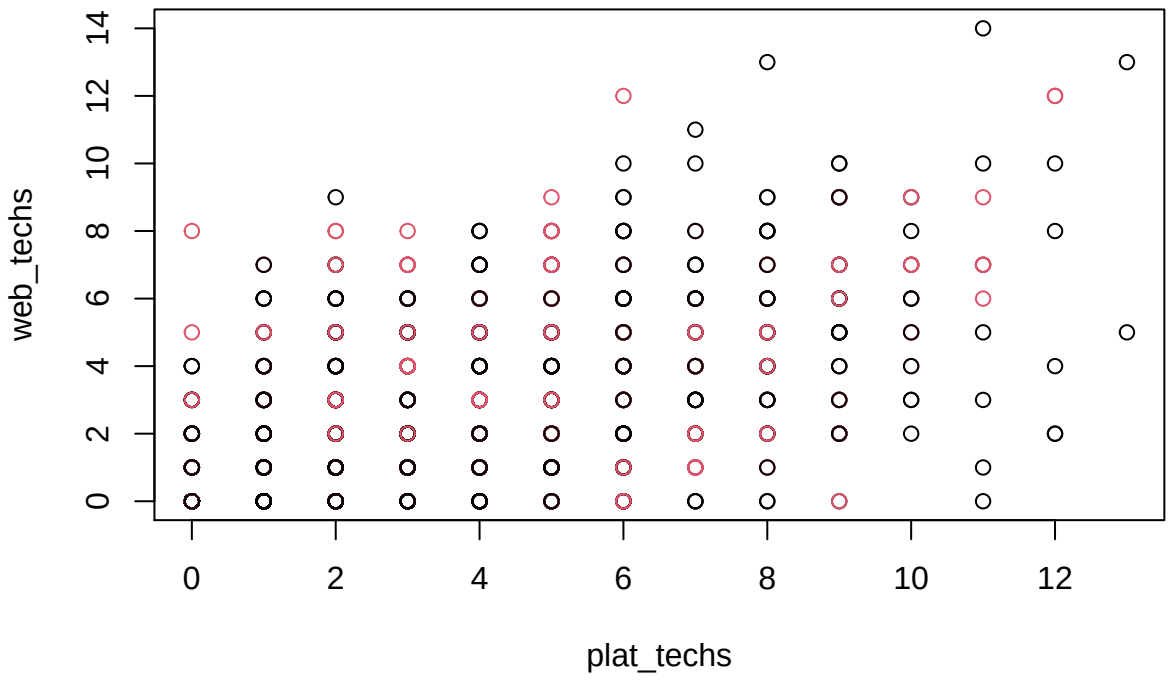


Al trabajar con estas 2 variables se obtiene un resultado similar al momento de agrupar visualmente los datos analizados. Vemos que los puntos están demasiado dispersos.

Probamos ahora con las 2 ultimas variables: **plat_techs** y **web_techs** del dataset para verificar si se puede encontrar los clusters con los valores de k, obtenidos previamente.

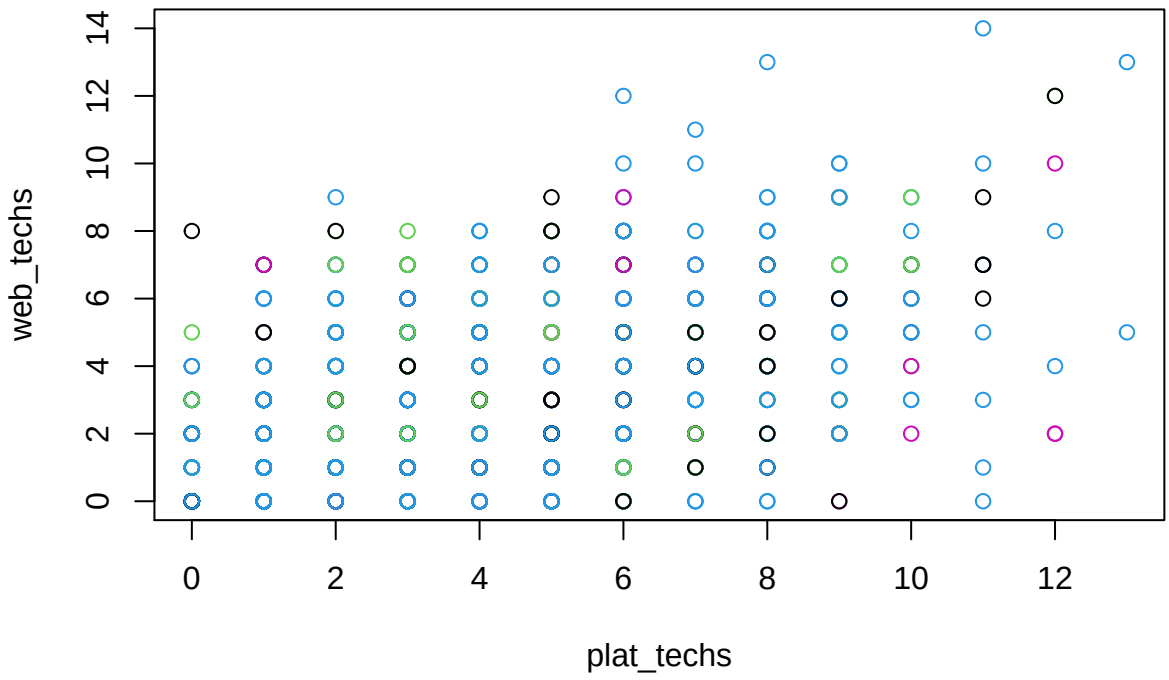
Para k=2

```
# plat_techs y web_techs
dev2PWclust <- kmeans(dataNS, 2)

plot(dataNS[c(9,10)], col=dev2PWclust$cluster)
```

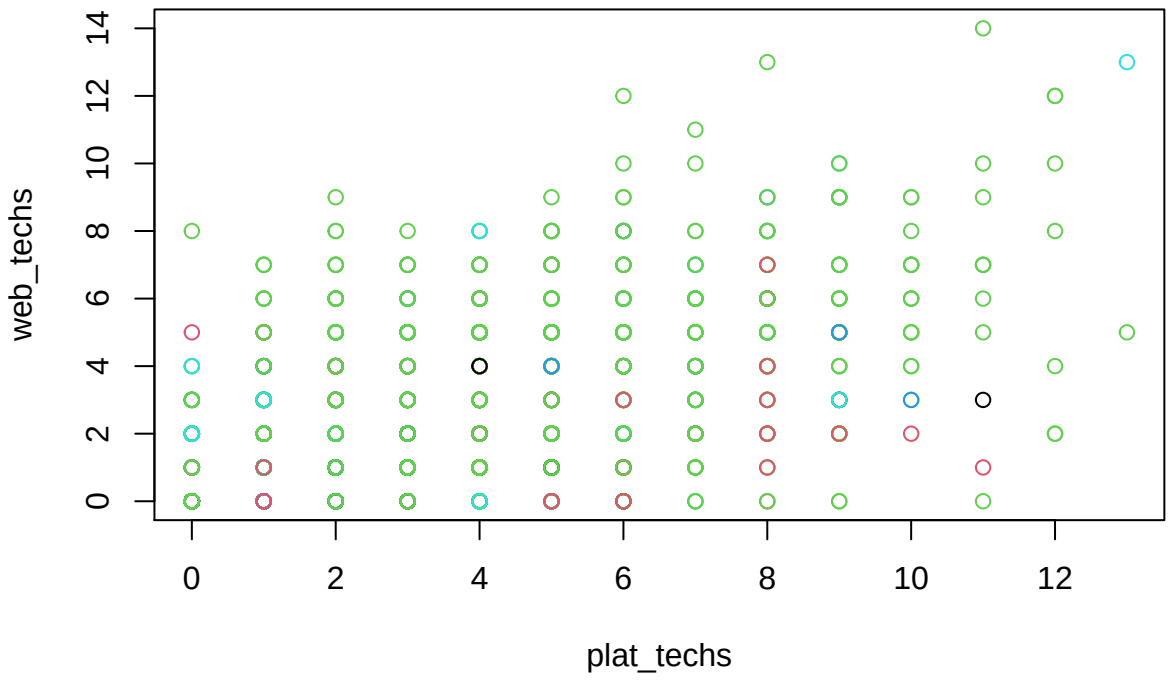


Para k=7

```
# plat_techs y web_techs
dev7PWclust <- kmeans(dataNS, 7)

plot(dataNS[c(9,10)], col=dev7PWclust$cluster)
```

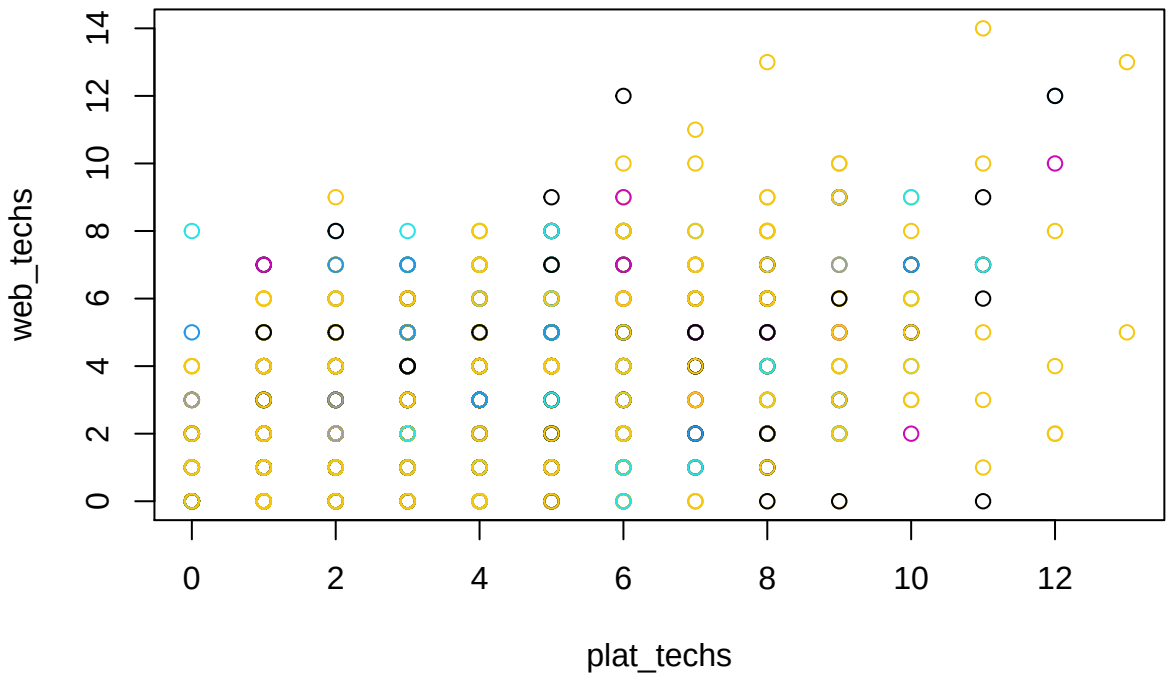



```
plot(dataNS[c(9,10)], col=as.factor(dataWithClass$MainBranch))
```

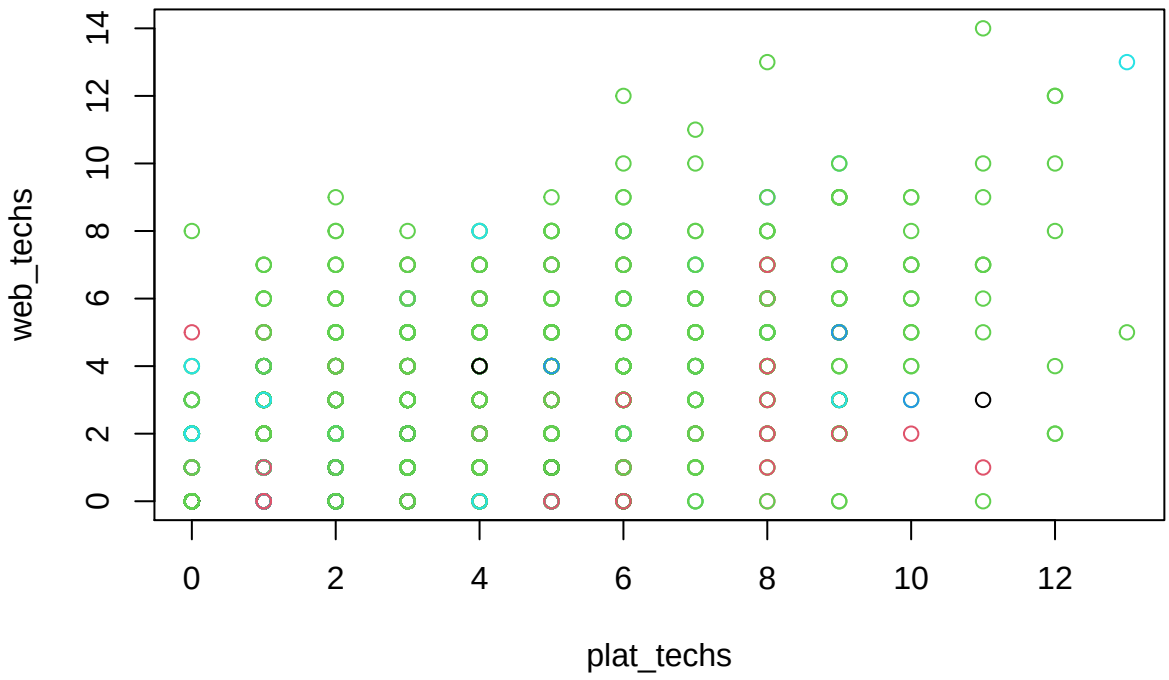


Para k=9

```
# plat_techs y web_techs
dev9PWclust <- kmeans(dataNS, 9)

plot(dataNS[c(9,10)], col=dev9PWclust$cluster)
```



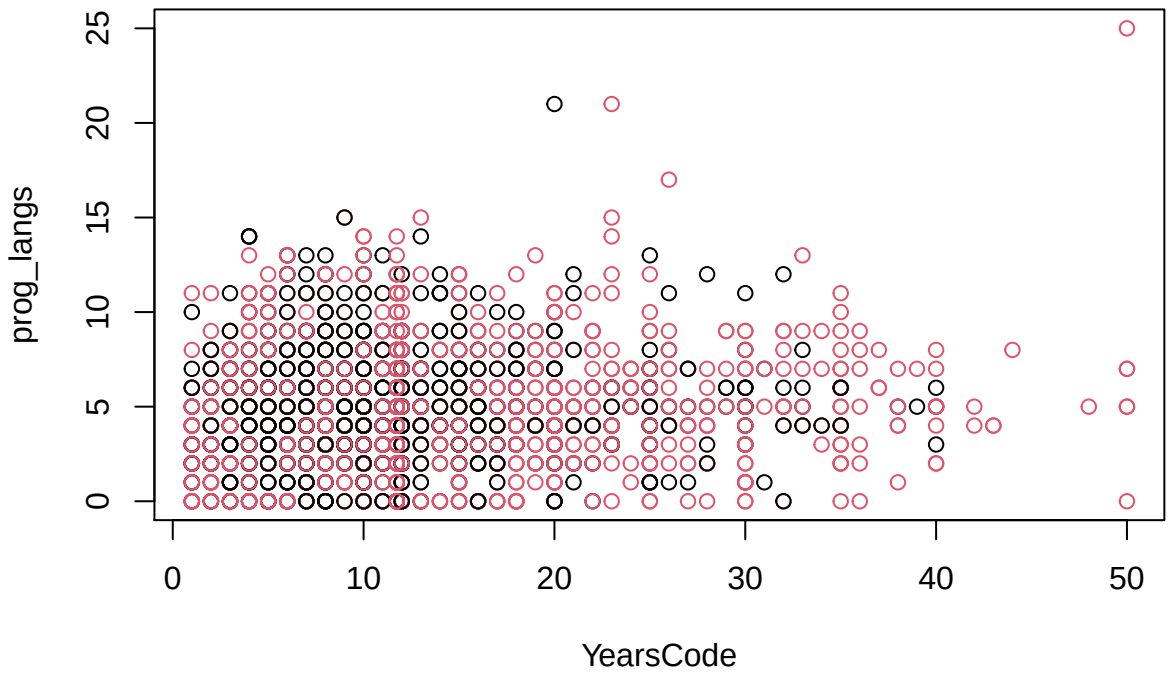

```
plot(dataNS[c(9,10)], col=as.factor(dataWithClass$MainBranch))
```



Vemos que de igual manera no se obtuvo una clara definición de los clusters. Ahora vamos a analizar 2 variables que guardan más relación entre las 10 variables que existen. Vamos a analizar **YearsCode** vs **prog_langs**

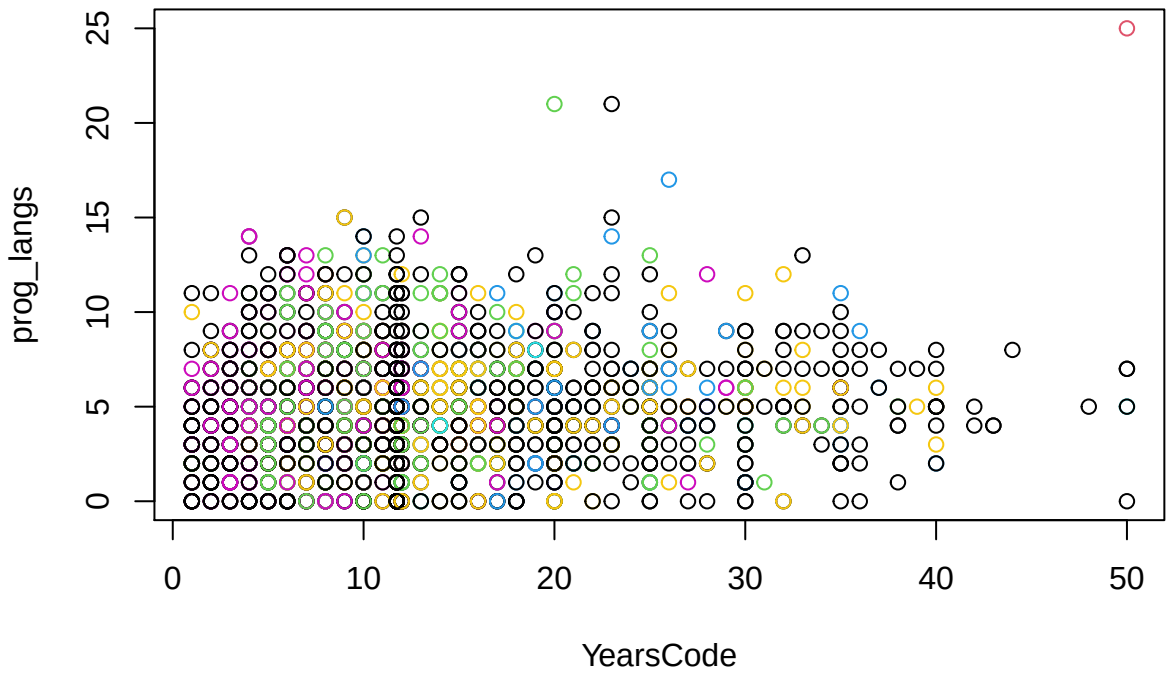
Comenzamos analizando para k=2

```
# YearsCode y prog_langs
dev2YPclust <- kmeans(dataNS, 2)

plot(dataNS[c(4,6)], col=dev2YPclust$cluster)
```

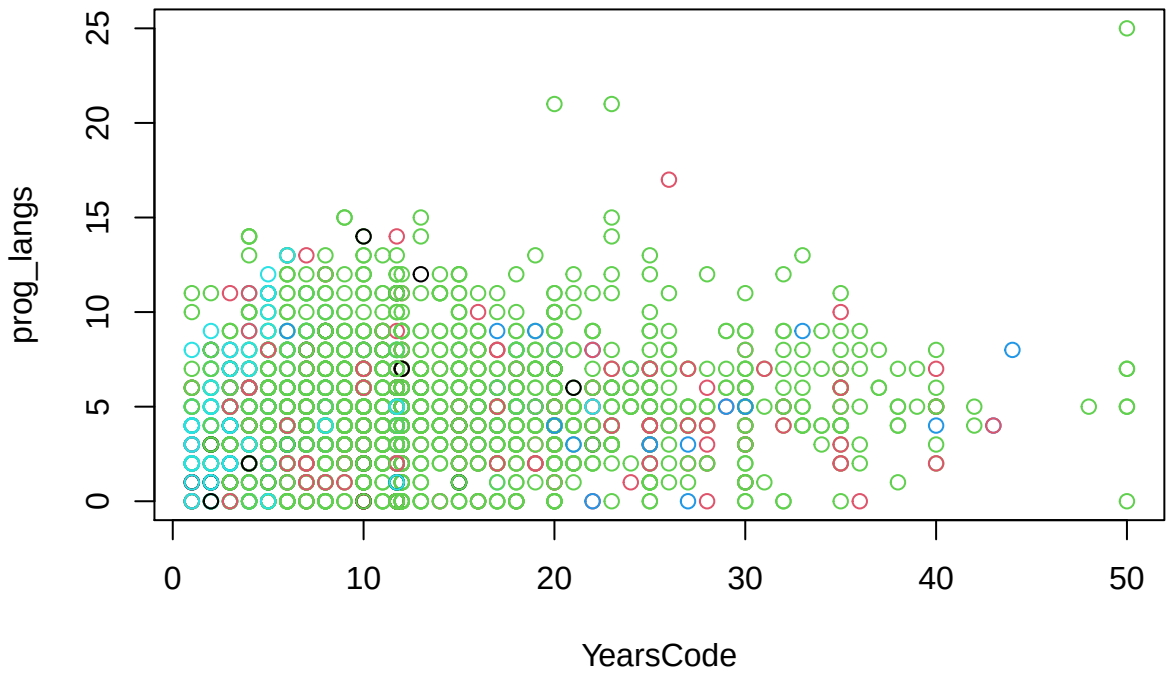


Probamos para k=7

```
# YearsCode y prog_langs
dev7YPclust <- kmeans(dataNS, 7)

plot(dataNS[c(4,6)], col=dev7YPclust$cluster)
```

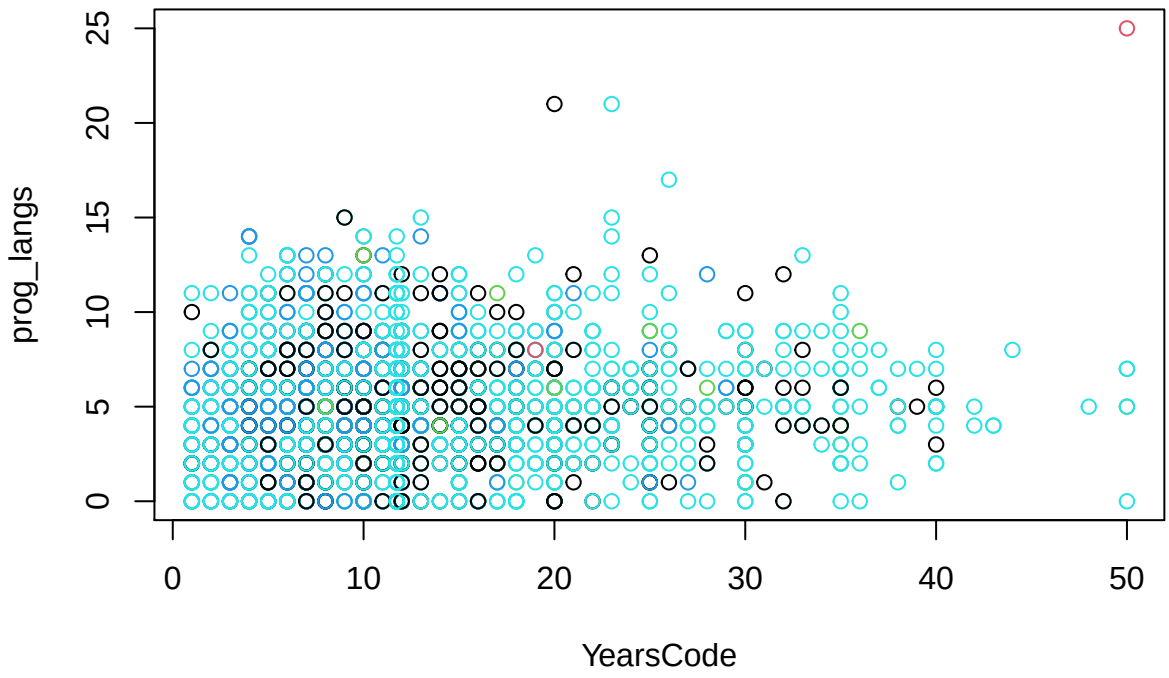



```
plot(dataNS[c(4,6)], col=as.factor(dataWithClass$MainBranch))
```

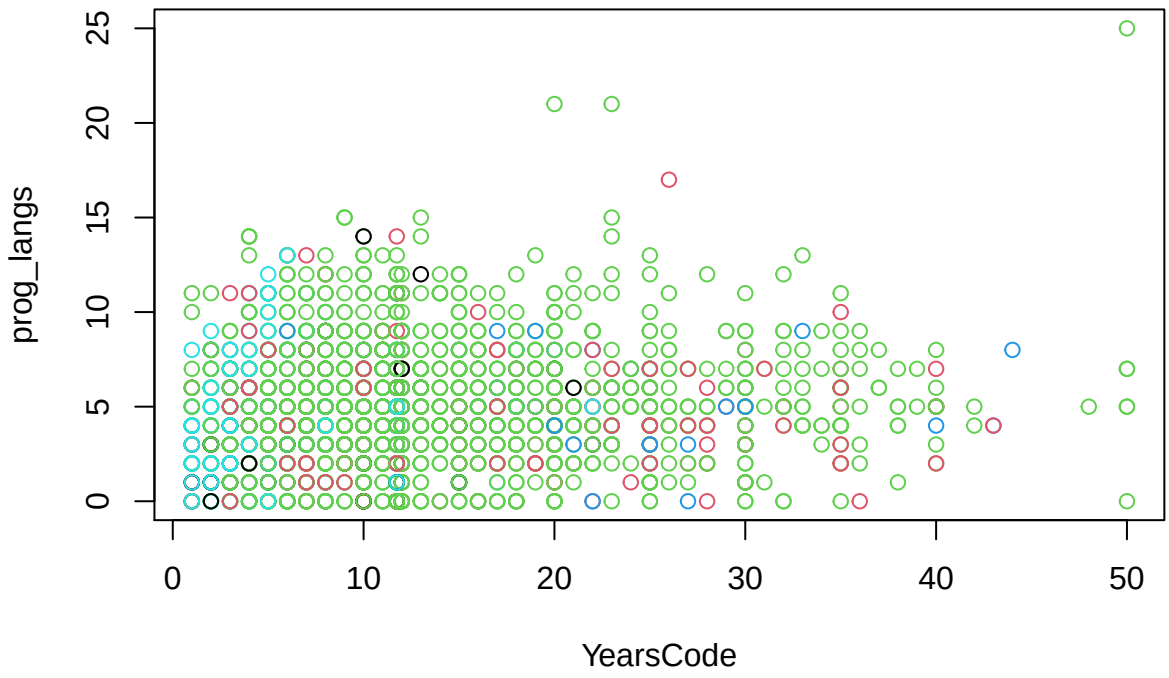


Probamos para k=9

```
# YearsCode y prog_langs
dev9YPclust <- kmeans(dataNS, 5)

plot(dataNS[c(4,6)], col=dev9YPclust$cluster)
```




```
plot(dataNS[c(4,6)], col=as.factor(dataWithClass$MainBranch))
```



CONCLUSIÓN:

Al no tener claramente definidos los clusters, vamos a proceder a cambiar la métrica de distancia en la aplicación del método K-Means para revisar si varian los valores de k y podemos obtener los grupos.

20%. Se genera modelo no supervisado con métrica de distancia distinta al anterior. Se muestran y comentan medidas de ca

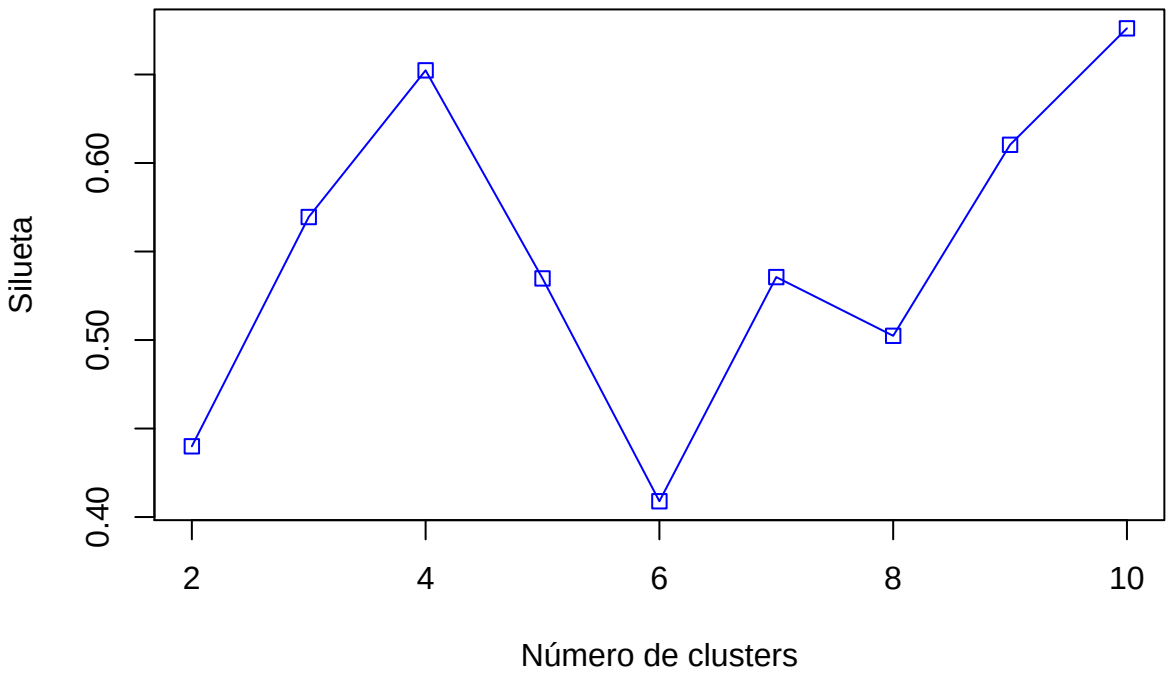
3. Aplica de nuevo el modelo anterior, pero usando una **métrica distinta** y compara los resultados.

Ahora probamos a definir el número óptimo de clusters con la métrica **manhattan**

```
d2 <- daisy(dataNS, metric="manhattan")
resultados2 <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit2           <- kmeans(dataNS, i)
  y_cluster2     <- fit2$cluster
  sk2            <- silhouette(y_cluster2, d2)
  resultados2[i] <- mean(sk2[,3])
}
```

Mostramos en un gráfica los valores de las siluetas media de cada prueba para comprobar que número de clústers es el mejor

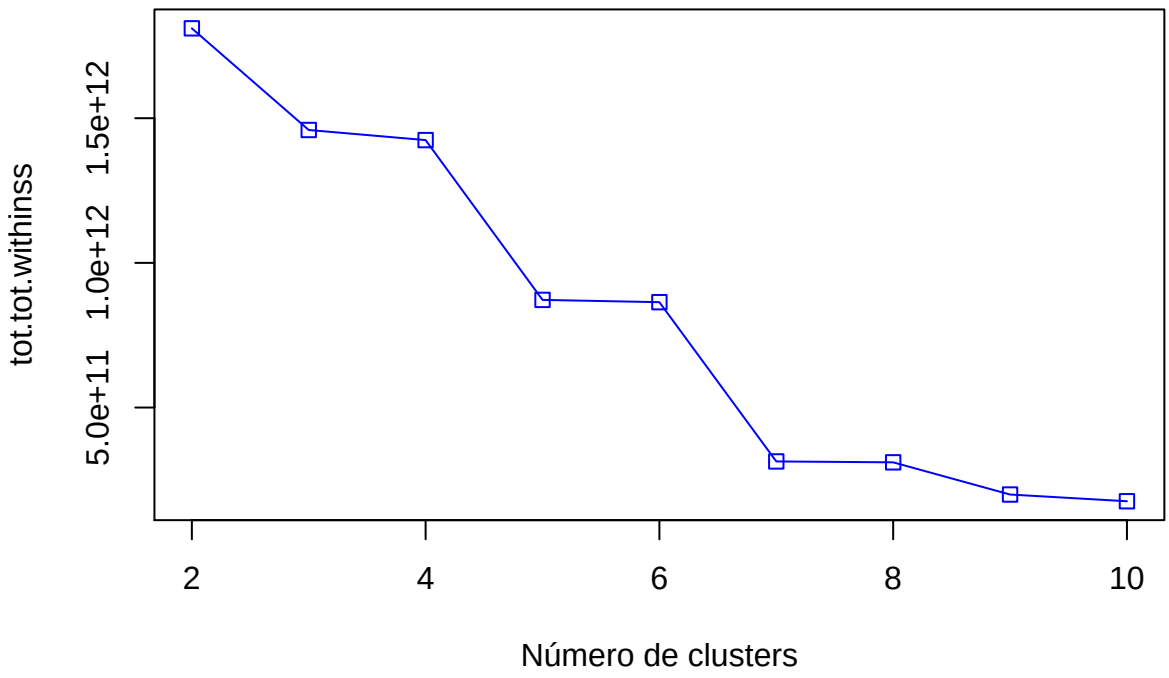
```
plot(2:10,resultados2[2:10],type="o",col="blue",pch=0,xlab="Número de clusters",ylab="Silueta")
```



Según la gráfica anterior, vemos que el número K con mejor puntuación **k=9**, luego le sigue **k=3** y **k=5**, sin embargo al ser un método no supervisado debemos seguir intentando determinar el valor óptimo de K usando otras técnicas.

Otro forma de evaluar cual es el mejor número de clústers es considerar el mejor modelo, aquel que ofrece la menor suma de los cuadrados de las distancias de los puntos de cada grupo con respecto a su centro (withinss), con la mayor separación entre centros de grupos (betweenss). Como se puede comprobar es una idea conceptualmente similar a la silueta. Una manera común de hacer la selección del número de clústers consiste en aplicar el método elbow (codo), que no es más que la selección del número de clústers en base a la inspección de la gráfica que se obtiene al iterar con el mismo conjunto de datos para distintos valores del número de clústers. Se seleccionará el valor que se encuentra en el "codo" de la curva

```
resultados2 <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit2           <- kmeans(dataNS, i)
  resultados2[i] <- fit2$tot.withinss
}
plot(2:10,resultados2[2:10],type="o",col="blue",pch=0,xlab="Número de clusters",ylab="tot.tot.withinss"
```



Para este caso hemos obtenido un valor promedio de **k=3**, clusters, que es donde la curva ya ha descendido y se ha estabilizado.

También se puede usar la función kmeansruns del paquete fpc que ejecuta el algoritmo kmeans con un conjunto de valores, para después seleccionar el valor del número de clústers que mejor funcione de acuerdo a dos criterios: la silueta media ("asw") y Calinski-Harabasz ("ch").

```
library(fpc)
set.seed(20000)
fit_ch2  <- kmeansruns(dataNS, krange = 1:10, criterion = "ch")
fit_asw2 <- kmeansruns(dataNS, krange = 1:10, criterion = "asw")
```

Podemos comprobar el valor con el que se ha obtenido el mejor resultado y también mostrar el resultado obtenido para todos los valores de k usando ambos criterios

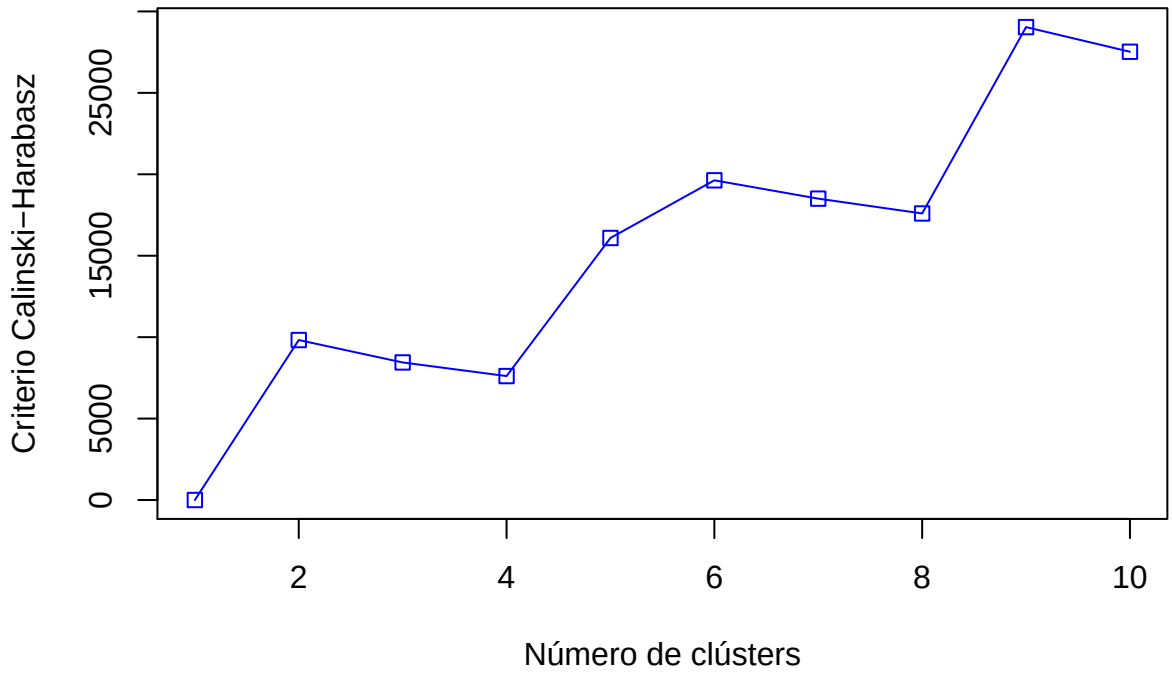
```
fit_ch2$bestk
```

```
## [1] 9
```

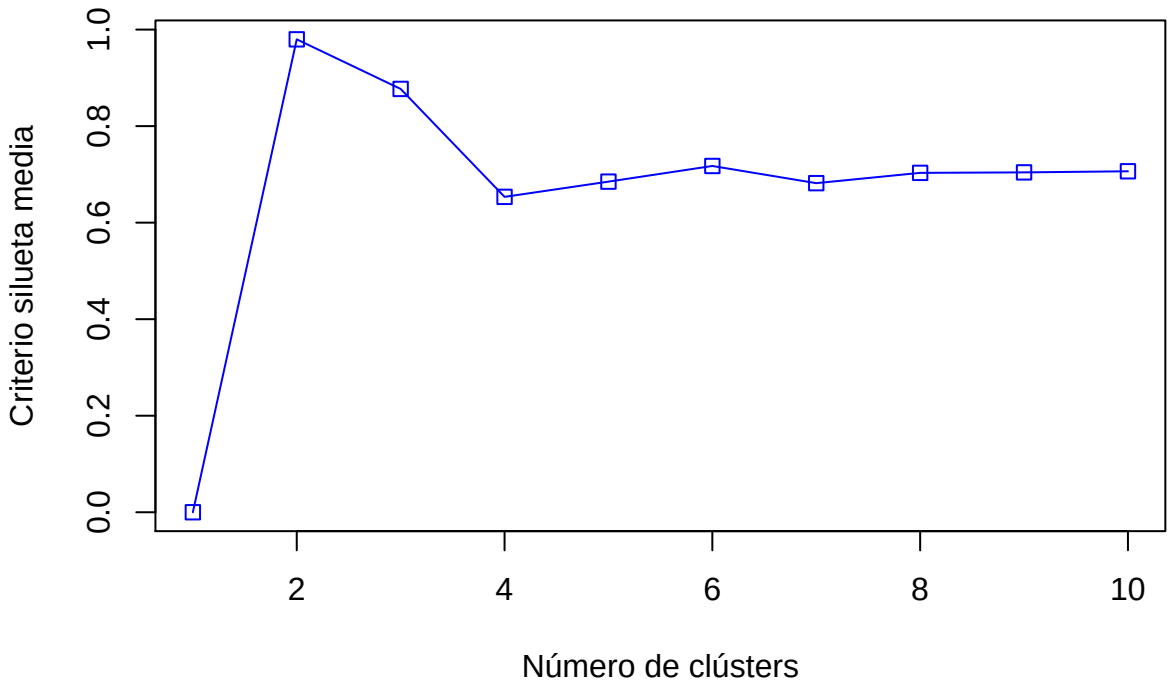
```
fit_asw2$bestk
```

```
## [1] 2
```

```
plot(1:10,fit_ch2$crit, type="o", col="blue",pch=0,xlab="Número de clústers",ylab="Criterio Calinski-Ha
```




```
plot(1:10,fit_asw2$crit, type="o", col="blue",pch=0,xlab="Número de clústers",ylab="Criterio silueta me
```

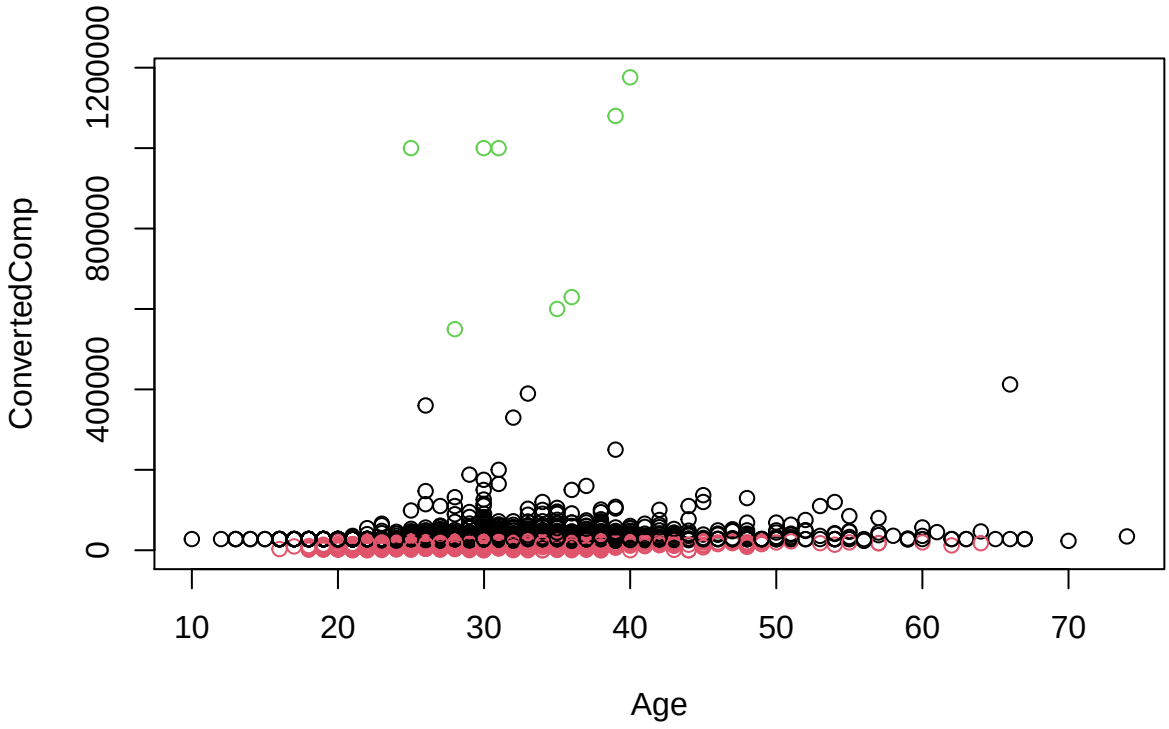


Los resultados obtenidos son muy diferentes. Para el caso de las siluetas medias obtuvimos *2* y luego con el método elbow (codo) obtuvimos *6*. Finalmente usando la función kmeansruns y aplicando los criterios Calinski-Harabasz y silueta media, se obtiene *9* y *2*, respectivamente. Notamos que el valor k=2 coincide con el obtenido inicialmente.

Ahora vamos a analizar los datos de manera visual, comparandolos de 2 en 2, con el valor real que sabemos está almacenado en el campo "MainBranch" del dataset original.

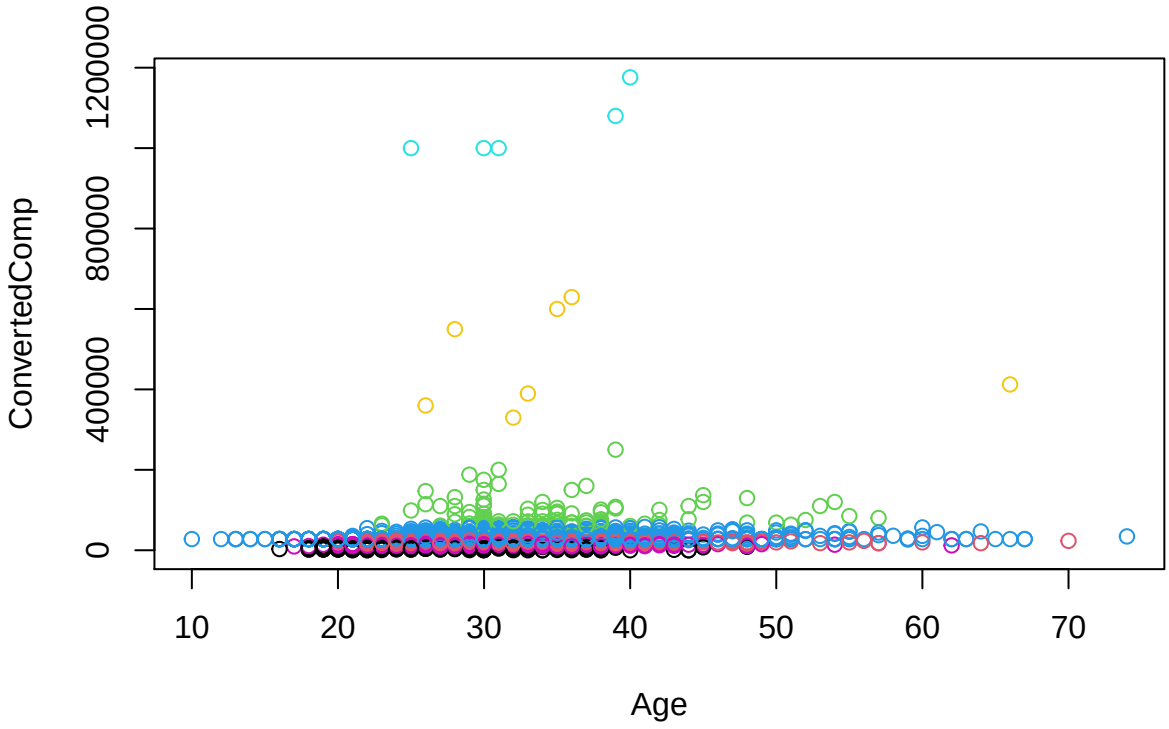
Vamos a analizar visualmente para los valores de **k** que aún no han sido testeados. En este caso para el valor de k=3, con los atributos *Age* (Edad) y *ConvertedComp* (Salario Anual en dolares):

```
dev3ACclusters <- kmeans(dataNS, 3)

# Edad y Salario Anual
plot(dataNS[c(1,2)], col=dev3ACclusters$cluster)
```

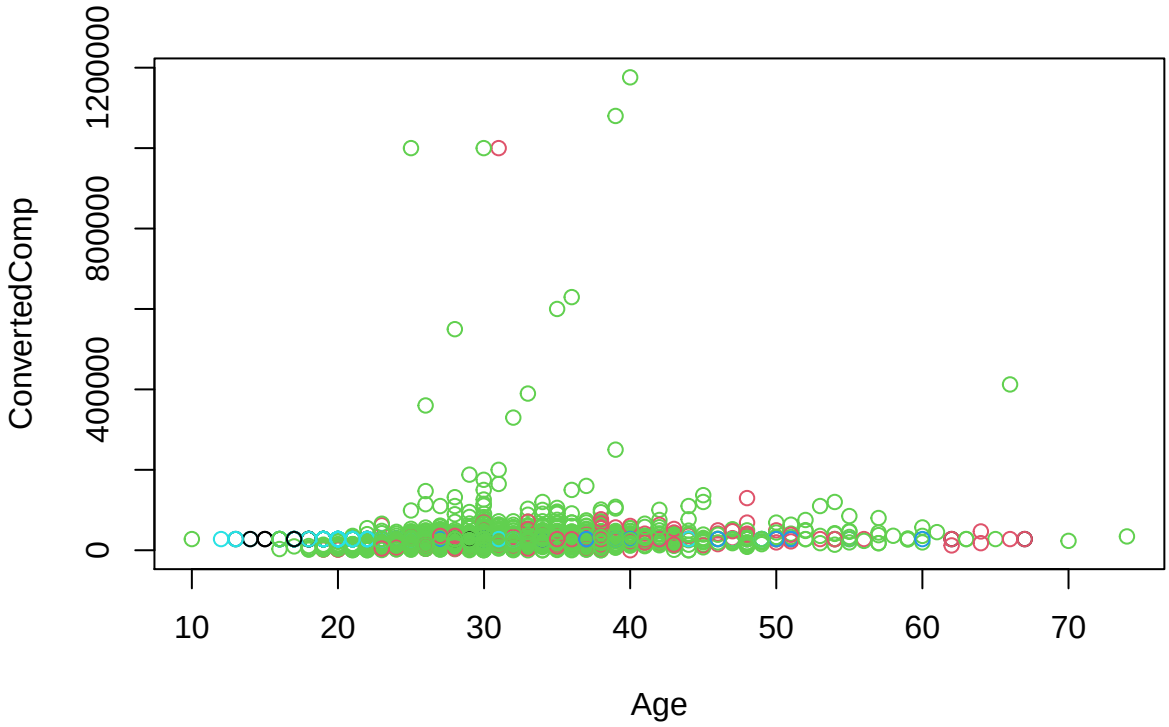


Vamos a analizar primero con el valor de k=5, para el par de atributos *Age* (Edad) y *ConvertedComp* (Salario Anual en dolares):

```
dev5ACclusters <- kmeans(dataNS, 5)

# Edad y Salario Anual
plot(dataNS[c(1,2)], col=dev7clusters$cluster)
```




```
plot(dataNS[c(1,2)], col=as.factor(dataWithClass$MainBranch))
```

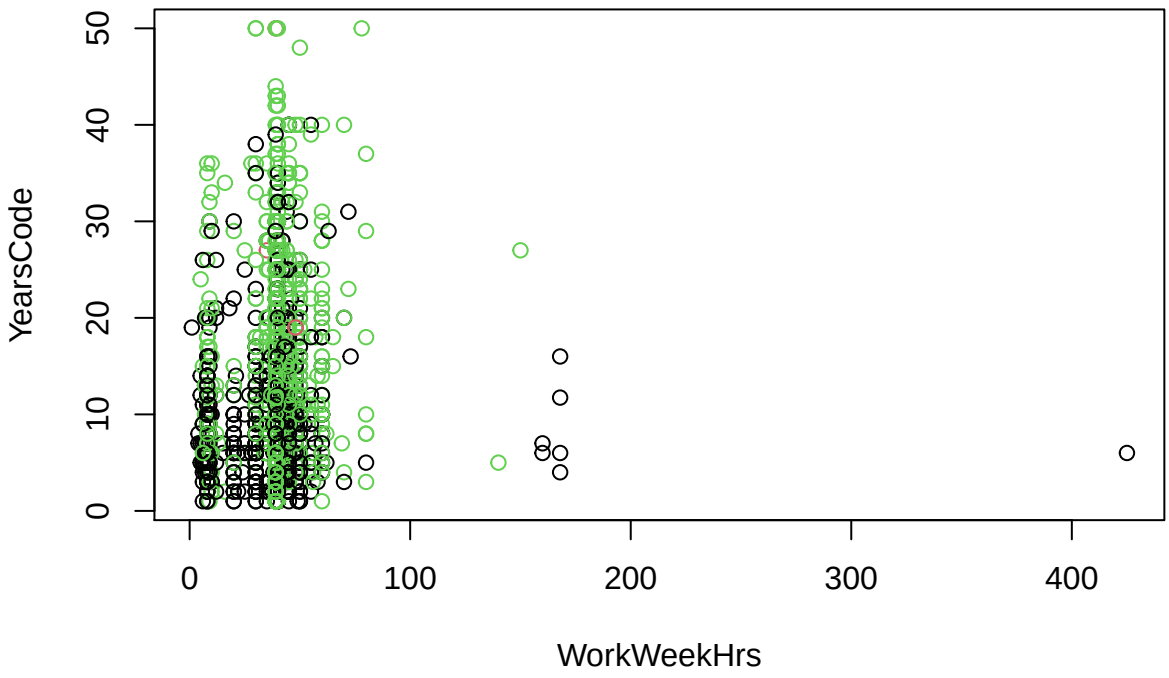


Vemos que se forman franjas de colores formadas por los puntos, lo que podría dar indicios de grupos. Entre la edad de 20 a 50 años se visualizan la mayor parte de estas agrupaciones. Hay 2 grupos que se alejan mucho del resto, entre los 25 años aproximadamente y los 40 años, estos tienen sueldos por encima de los 150.000 anual.

Ahora procedamos a analizar los clusters WorkWeekHrs (Horas de trabajo) vs. YearsCode (años de experiencia) y los valores de k=3 y k=5.

Para k=3:

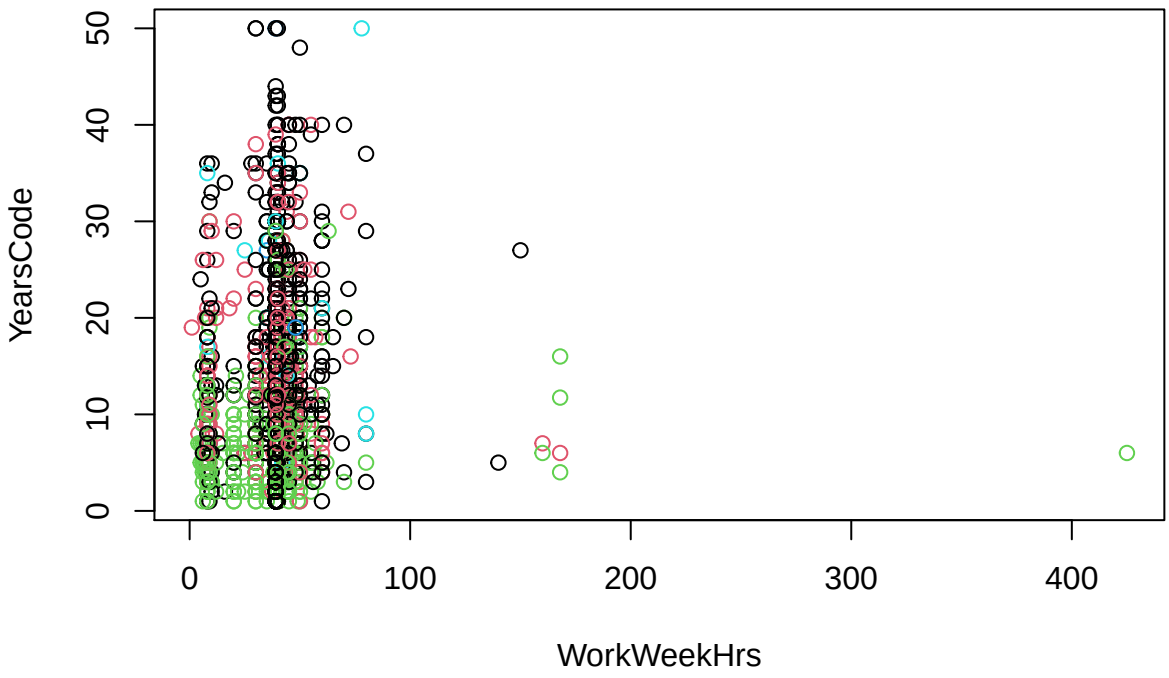
```
# WorkWeekHrs y YearsCode
dev3HYclust <- kmeans(dataNS, 3)

plot(dataNS[c(3,4)], col=dev3HYclust$cluster)
```



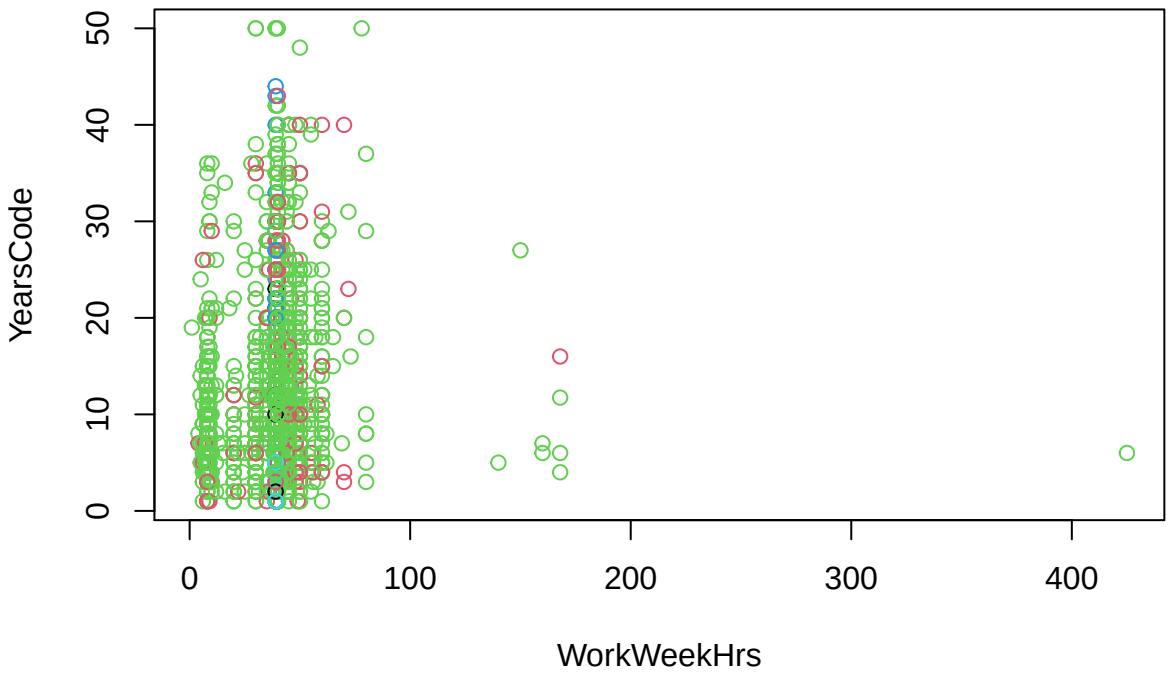
Vemos que para el análisis entre las Horas de trabajo y los años de experiencia, para el valor de k=3, no se obtiene grupos claramente definidos.

Ahora para k=5:

```
# WorkWeekHrs y YearsCode
dev5HYclust <- kmeans(dataNS, 5)

plot(dataNS[c(3,4)], col=dev5HYclust$cluster)
```




```
plot(dataNS[c(3,4)], col=as.factor(dataWithClass$MainBranch))
```



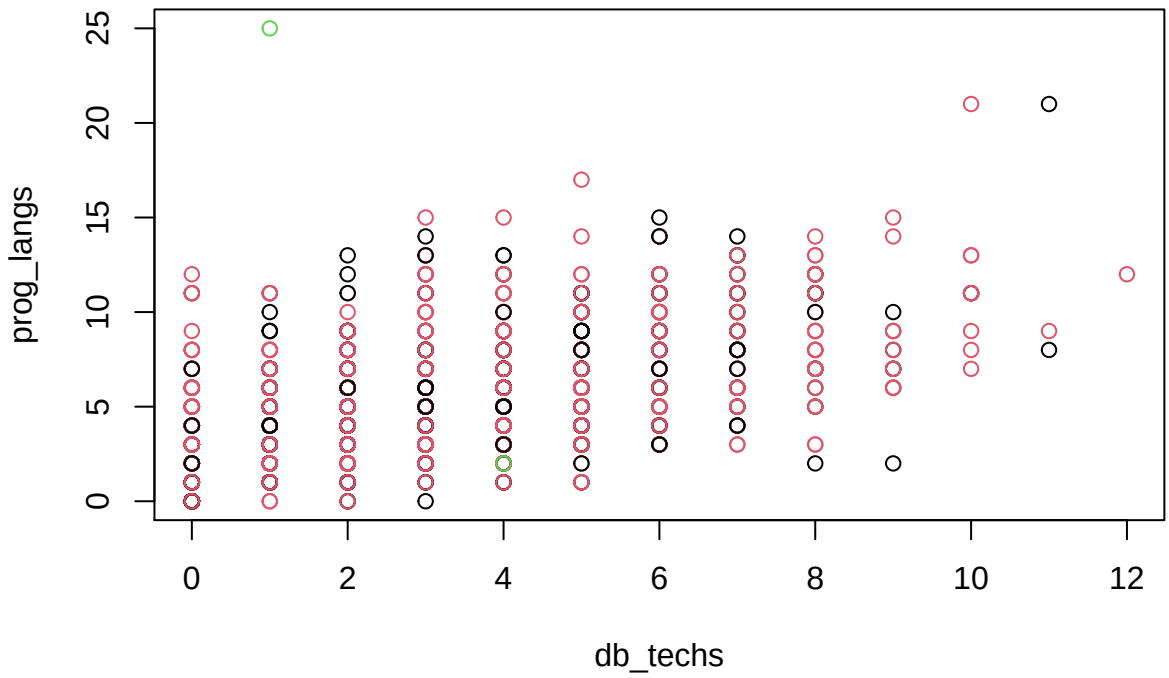
Vemos que para las 2 variables analizadas, el valor de k=5, tampoco define claramente los clusters

Para estas 2 variables seleccionadas tampoco se aplica de manera optima la agrupación por clusters, no coinciden con los valores los grupos de la variable clase. Es lógico que ambas variables no vayan a definir claramente los clusters ya que no guardan relación.

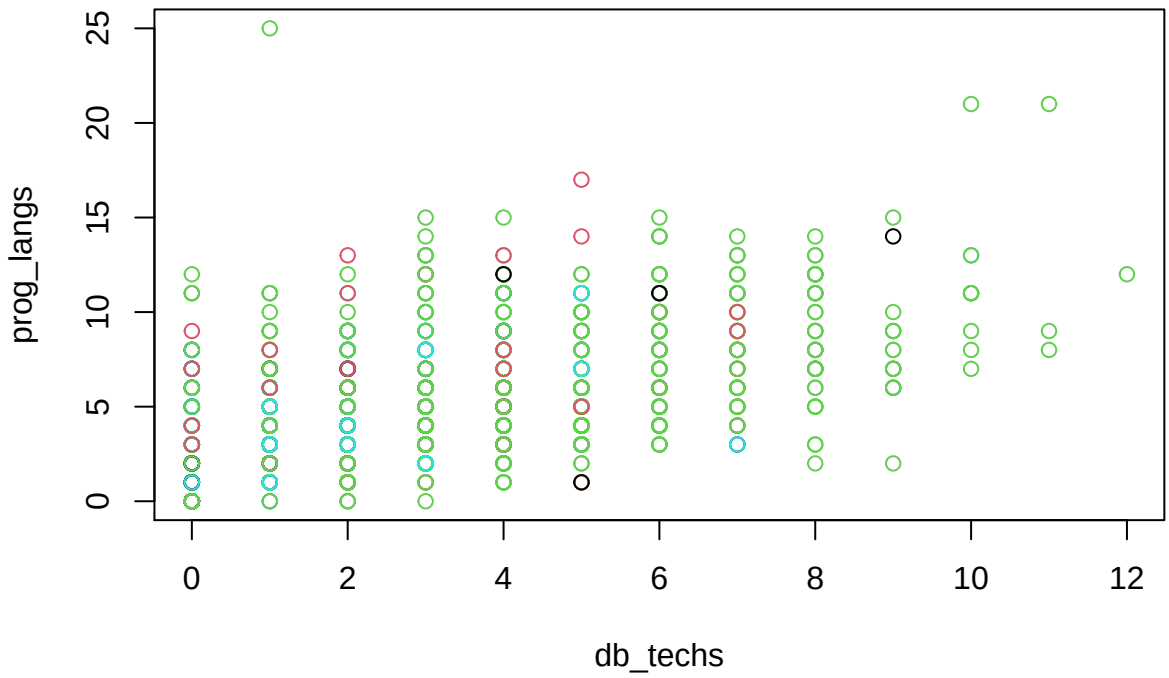
Ahora procedamos a analizar los clusters db_techs (Tecnologías de BD usadas) vs. prog_langs (lenguajes de programación) y los mismos valores de k=3 y k=5.

Para k=3:

```
# db_techs y prog_langs
dev3DPclust <- kmeans(dataNS, 3)

plot(dataNS[c(5,6)], col=dev3DPclust$cluster)
```



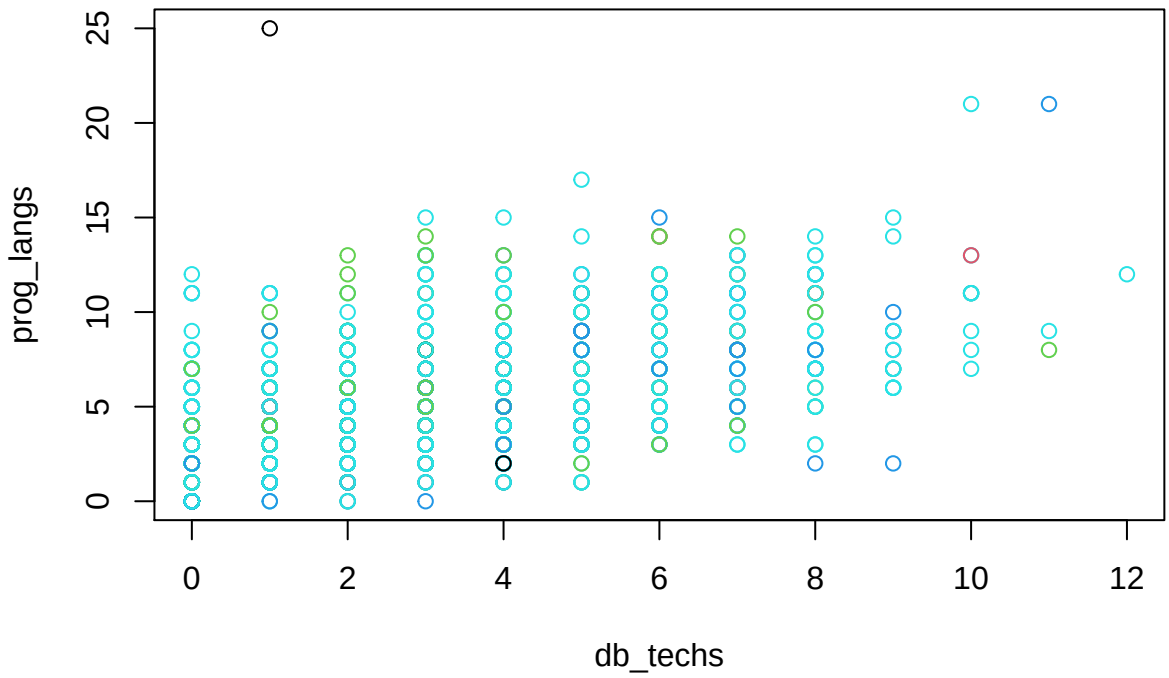

```
plot(dataNS[c(5,6)], col=as.factor(dataWithClass$MainBranch))
```



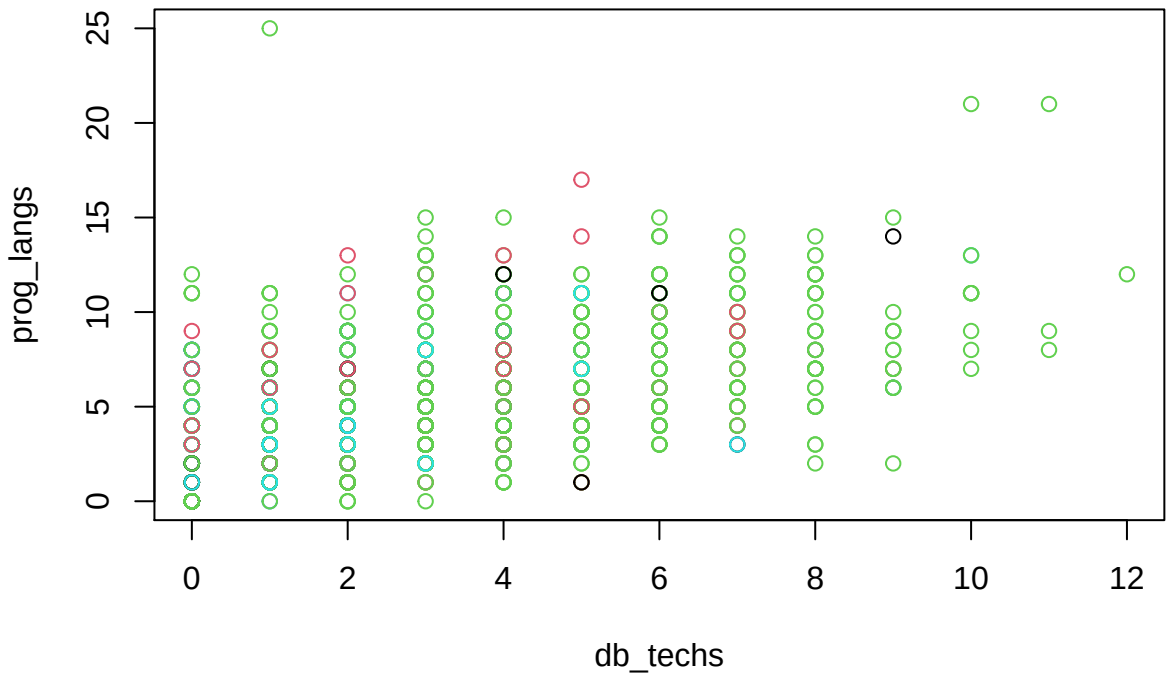
Vemos que los clusters no están definidos. Los puntos están muy dispersos en relación con las comparaciones de 2 en 2 con las primeras 4 variables del dataset.

Ahora para k=5:

```
# db_techs y prog_langs
dev5DPclust <- kmeans(dataNS, 5)

plot(dataNS[c(5,6)], col=dev5DPclust$cluster)
```




```
plot(dataNS[c(5,6)], col=as.factor(dataWithClass$MainBranch))
```

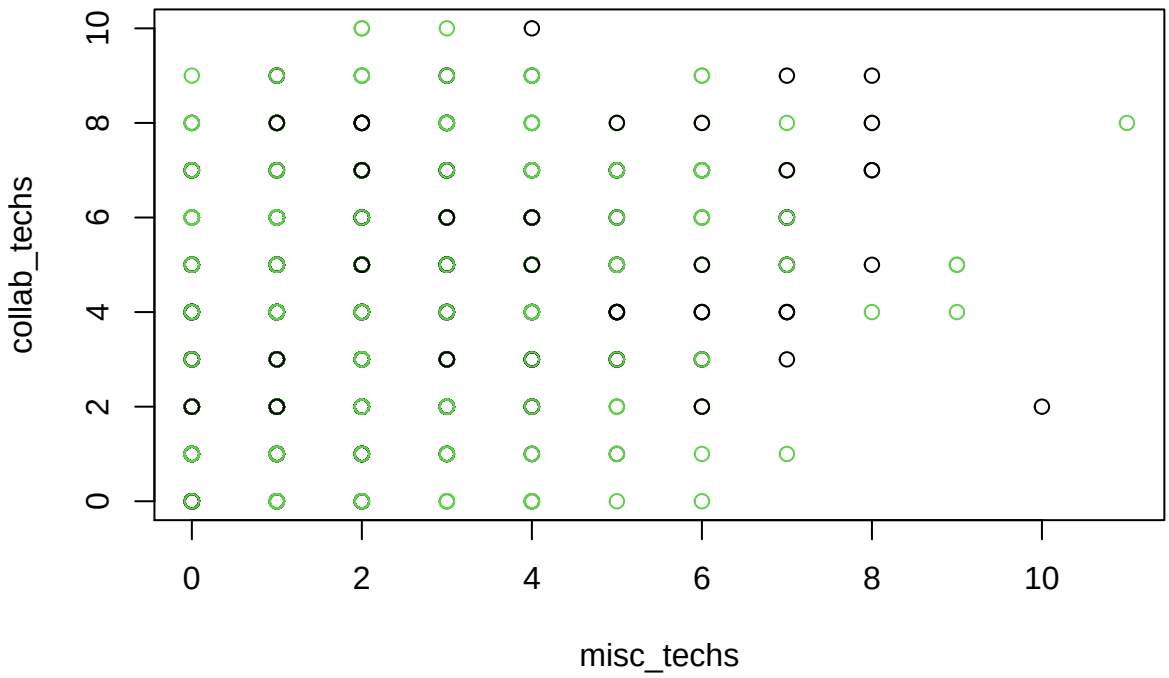


De igual manera que para k=3, vemos que los clusters no están definidos. Los puntos están muy dispersos.

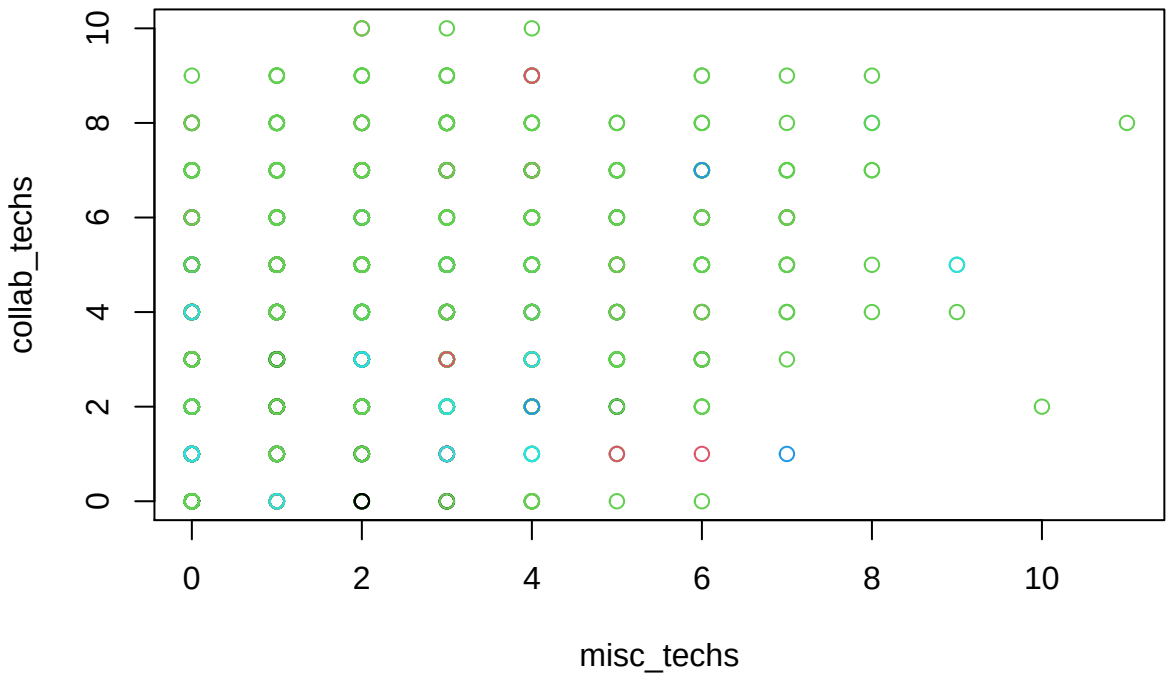
Ahora procedamos a analizar los clusters misc_techs (Tecnologías varias) vs. collab_techs (herramientas de colaboración) y los mismos valores de k.

Para k=3:

```
# misc_techs y collab_techs
dev3MCclust <- kmeans(dataNS, 3)

plot(dataNS[c(7,8)], col=dev3MCclust$cluster)
```

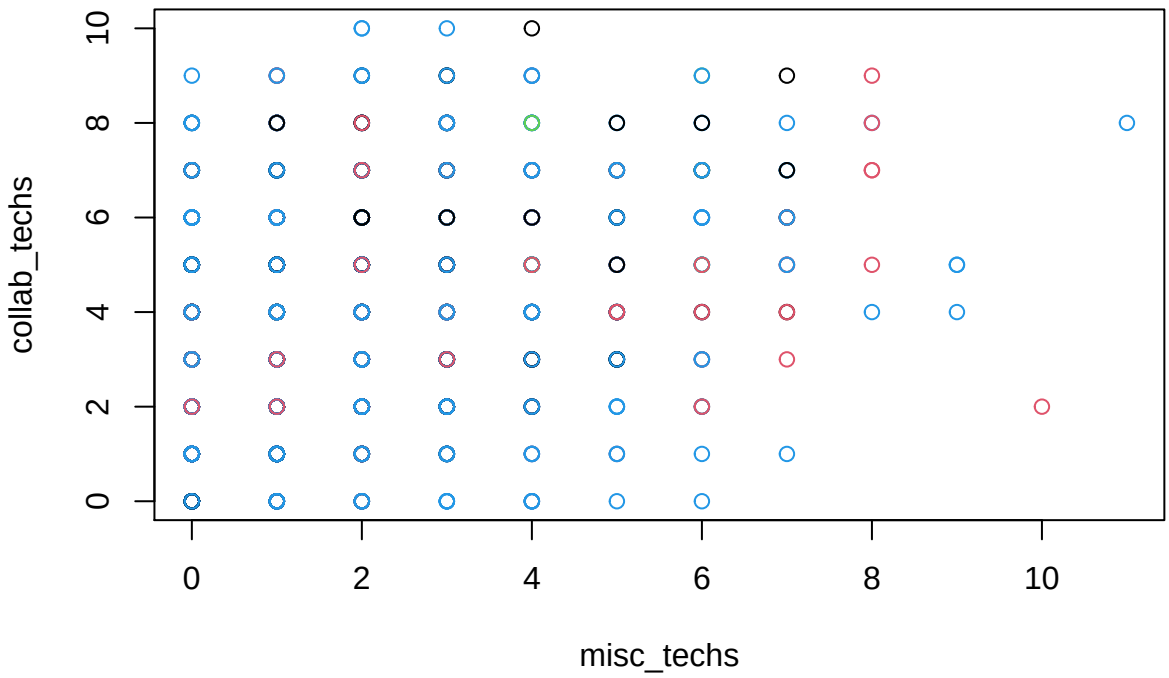



```
plot(dataNS[c(7,8)], col=as.factor(dataWithClass$MainBranch))
```

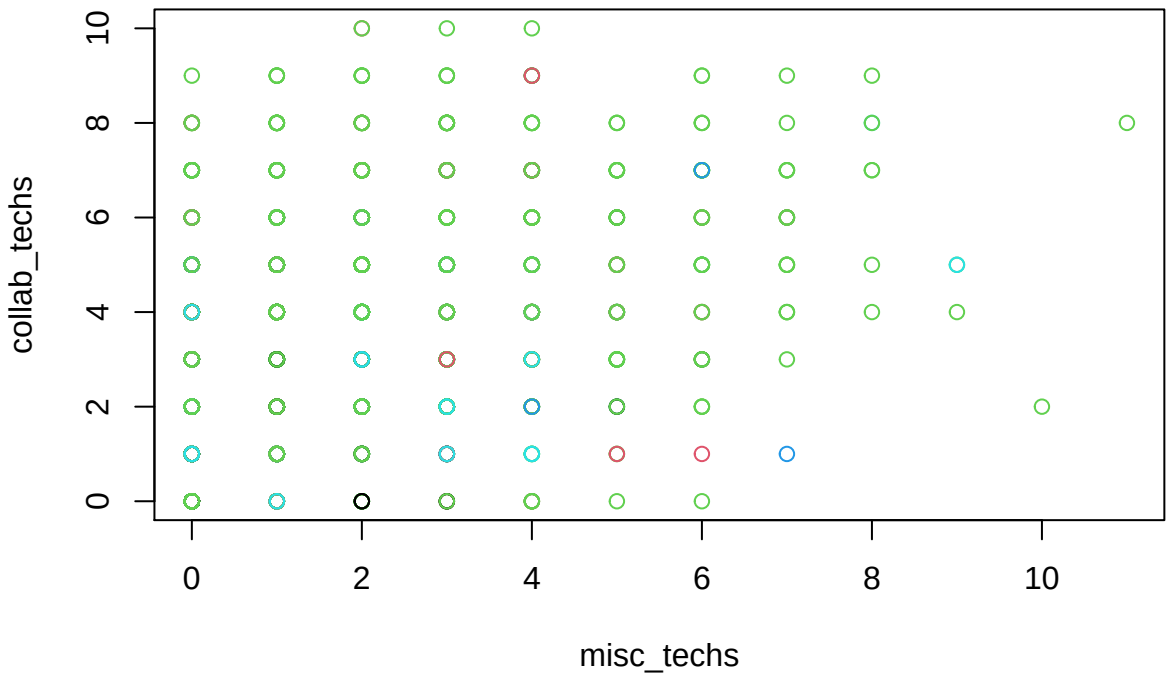


Ahora para k=5:

```
# misc_techs y collab_techs
dev5MCclust <- kmeans(dataNS, 5)

plot(dataNS[c(7,8)], col=dev5MCclust$cluster)
```




```
plot(dataNS[c(7,8)], col=as.factor(dataWithClass$MainBranch))
```

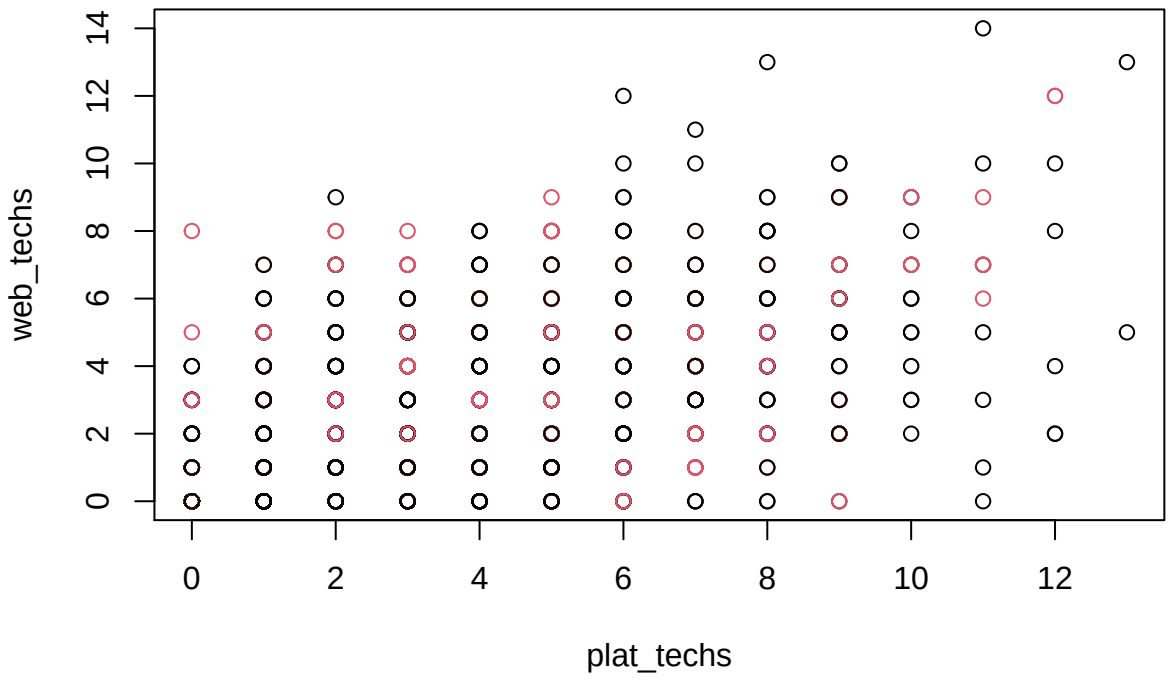


Al trabajar con estas 2 variables se obtiene un resultado similar al momento de agrupar visualmente los datos analizados. Vemos que los puntos están demasiado dispersos.

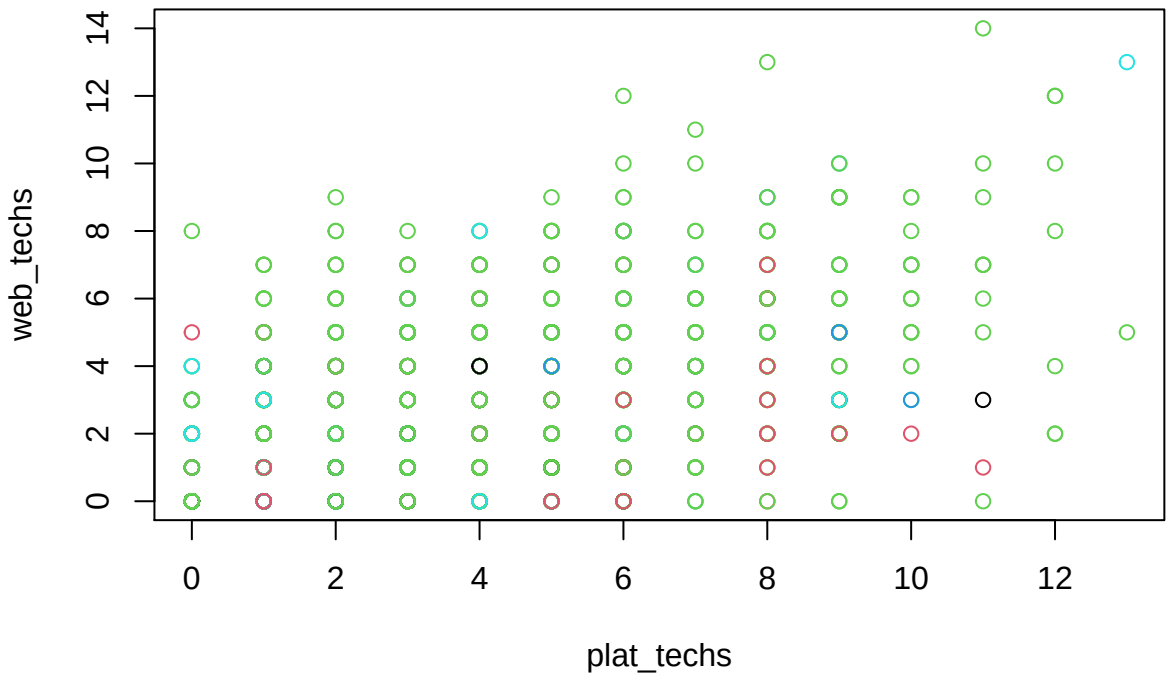
Probamos ahora con las 2 ultimas variables: **plat_techs** y **web_techs** del dataset para verificar si se puede encontrar los clusters con los valores de k, obtenidos previamente.

Para k=3

```
# plat_techs y web_techs
dev3PWclust <- kmeans(dataNS, 3)

plot(dataNS[c(9,10)], col=dev3PWclust$cluster)
```

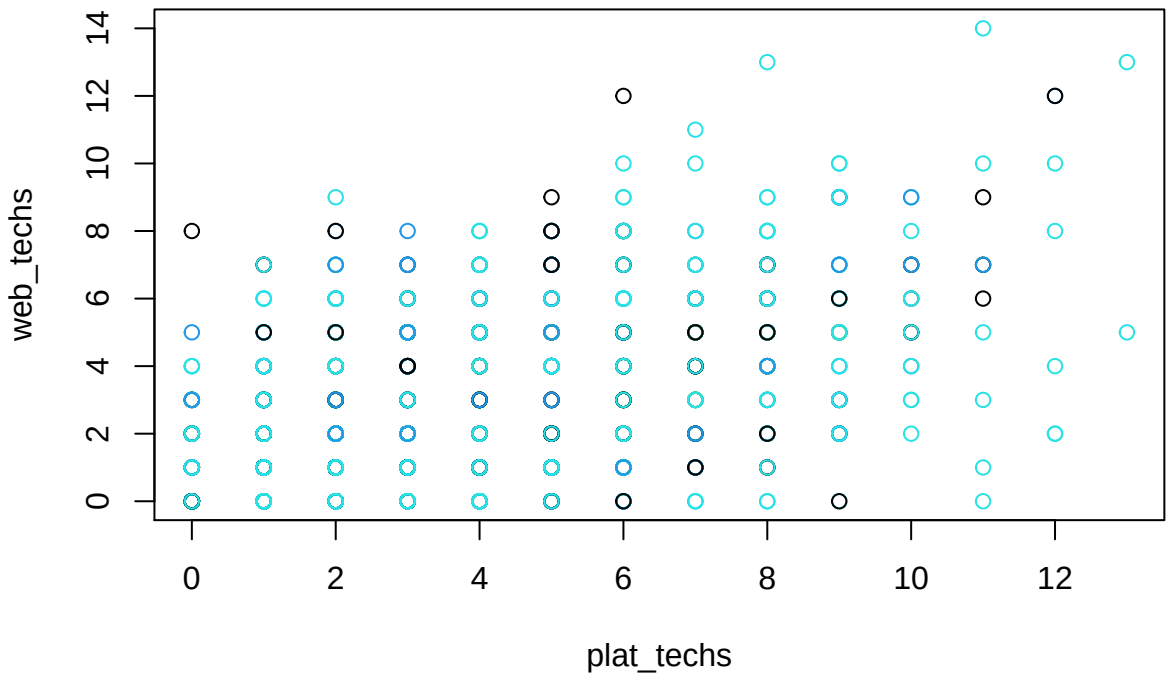



```
plot(dataNS[c(9,10)], col=as.factor(dataWithClass$MainBranch))
```

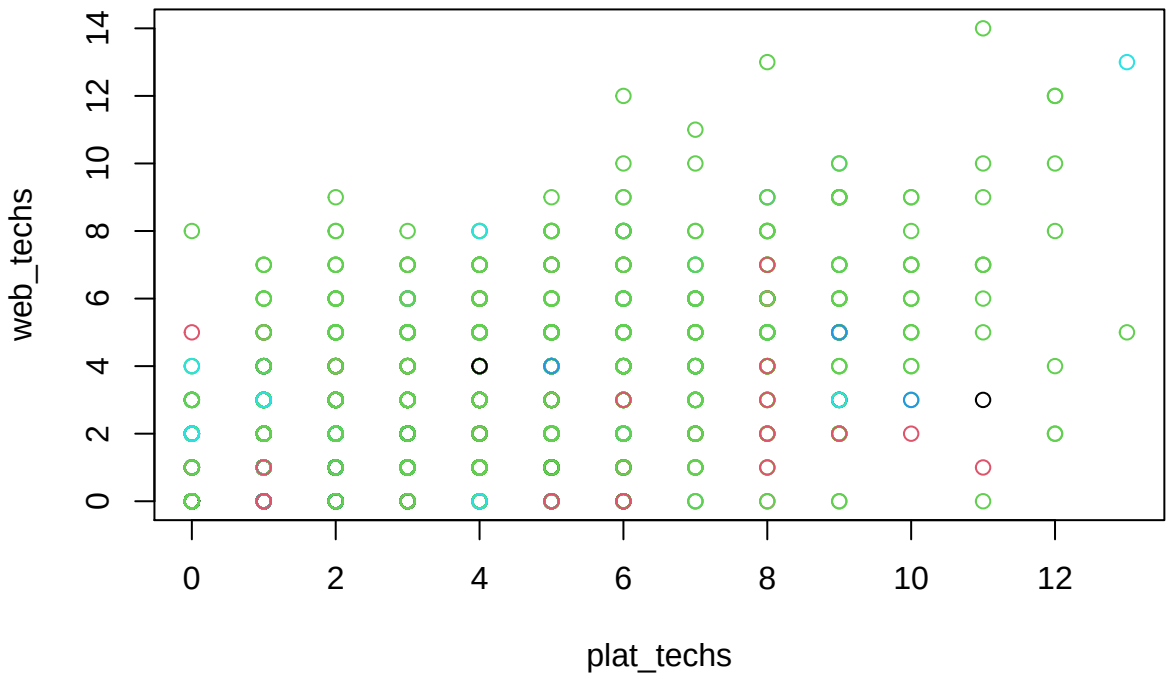


Para k=5

```
# plat_techs y web_techs
dev5PWclust <- kmeans(dataNS, 5)

plot(dataNS[c(9,10)], col=dev5PWclust$cluster)
```



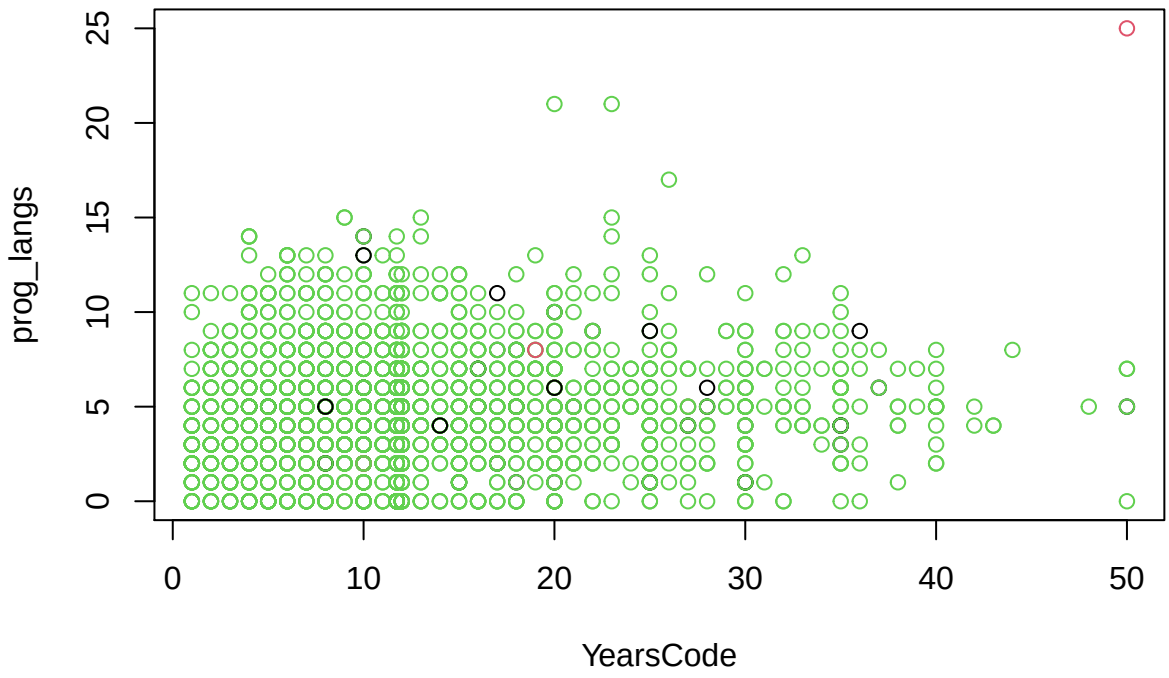

```
plot(dataNS[c(9,10)], col=as.factor(dataWithClass$MainBranch))
```



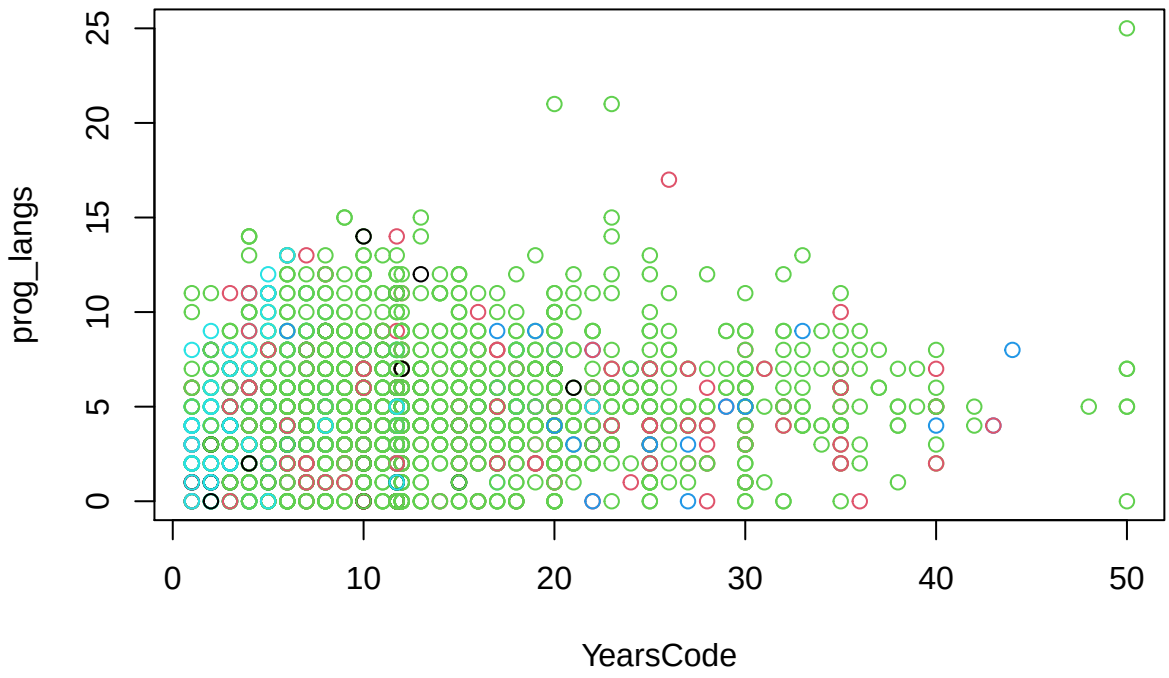
Vemos que de igual manera no se obtuvo una clara definición de los clusters. Ahora vamos a analizar 2 variables que guardan más relación entre las 10 variables que existen. Vamos a analizar **YearsCode** vs **prog_langs**

Comenzamos analizando para k=3

```
# YearsCode y prog_langs
dev3YPclust <- kmeans(dataNS, 3)

plot(dataNS[c(4,6)], col=dev3YPclust$cluster)
```

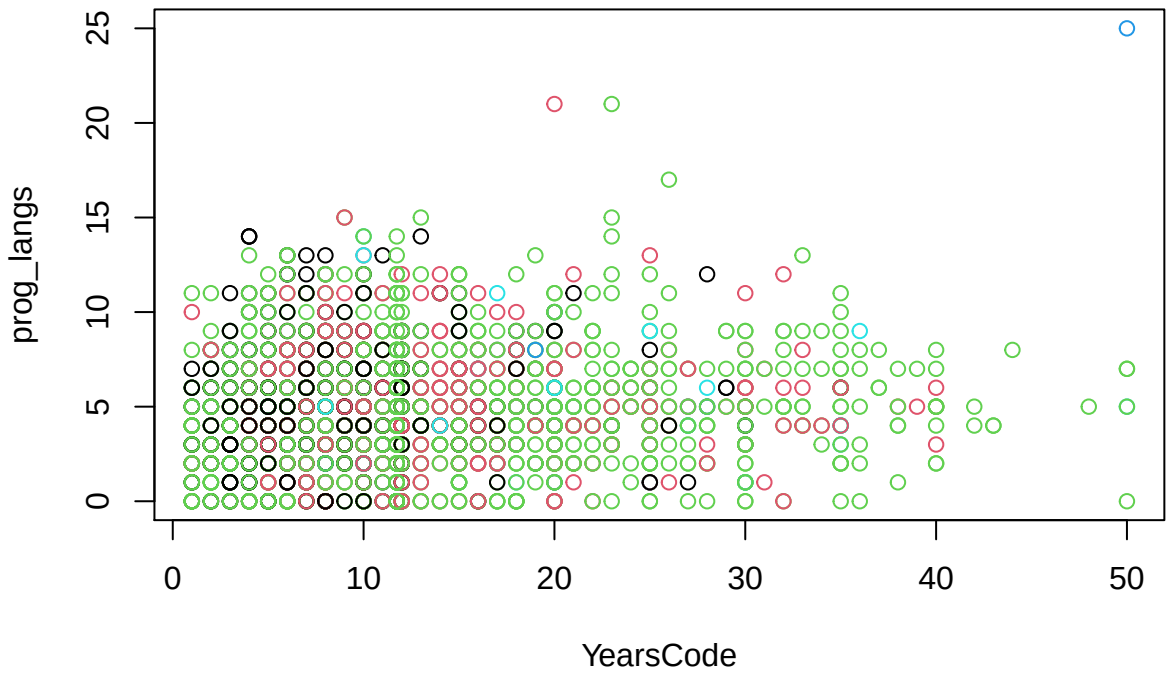



```
plot(dataNS[c(4,6)], col=as.factor(dataWithClass$MainBranch))
```

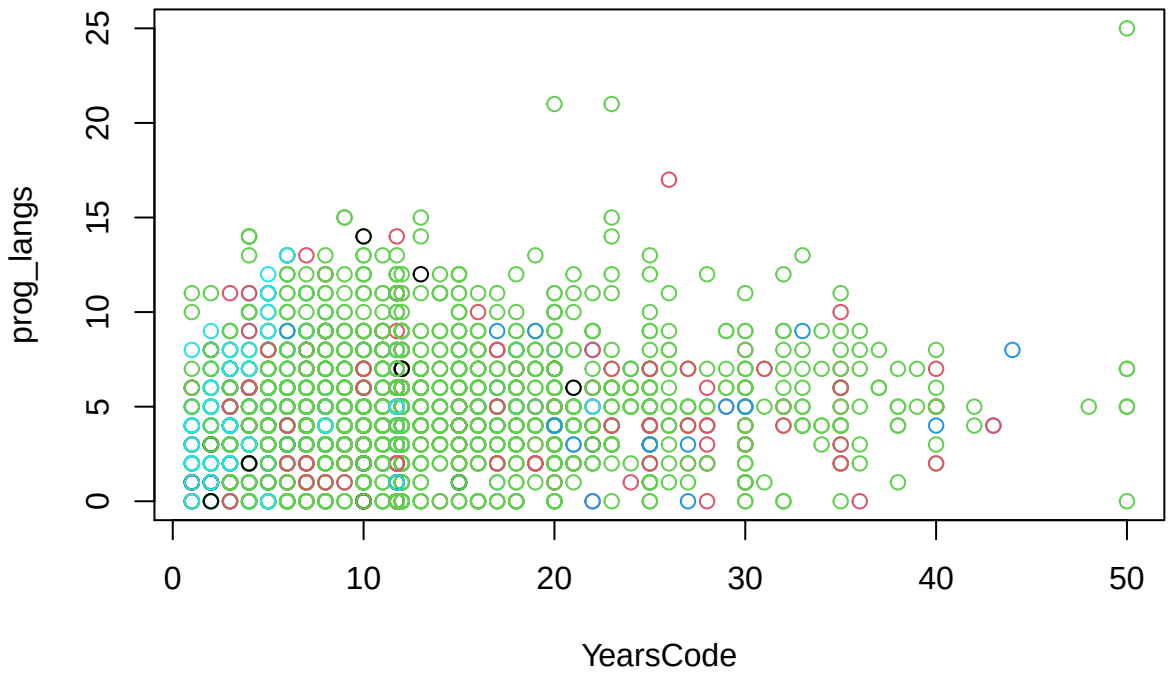


Probamos para k=5

```
# YearsCode y prog_langs
dev5YPclust <- kmeans(dataNS, 5)

plot(dataNS[c(4,6)], col=dev5YPclust$cluster)
```



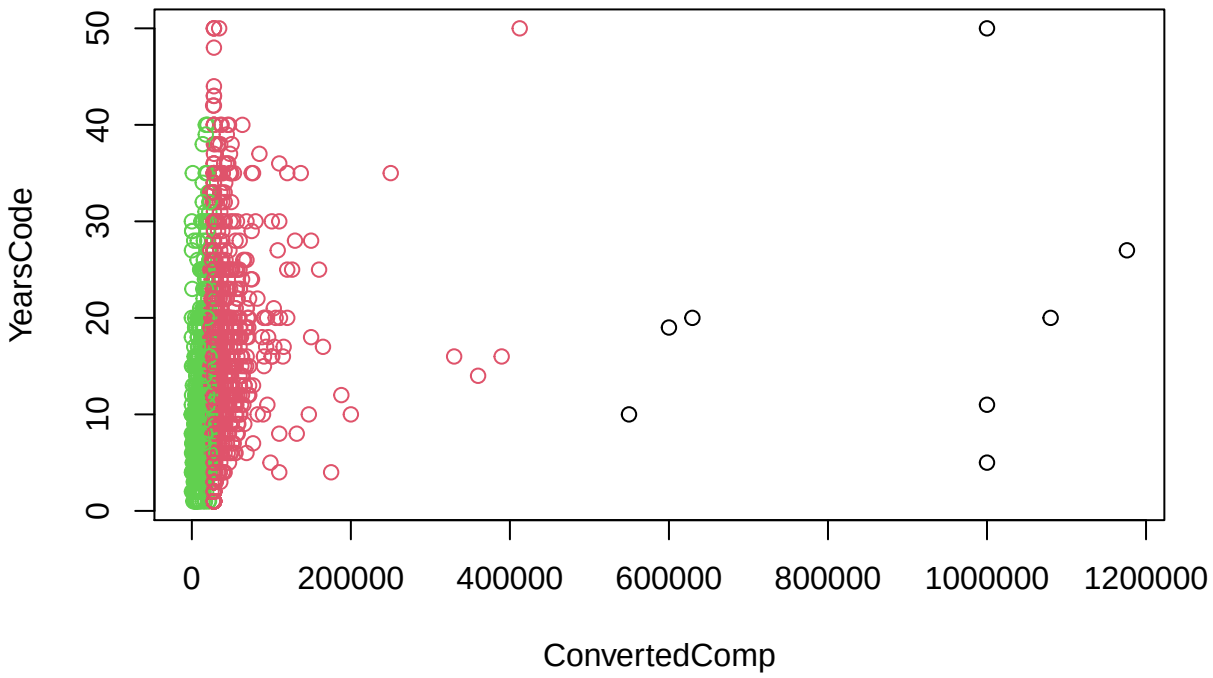

```
plot(dataNS[c(4,6)], col=as.factor(dataWithClass$MainBranch))
```



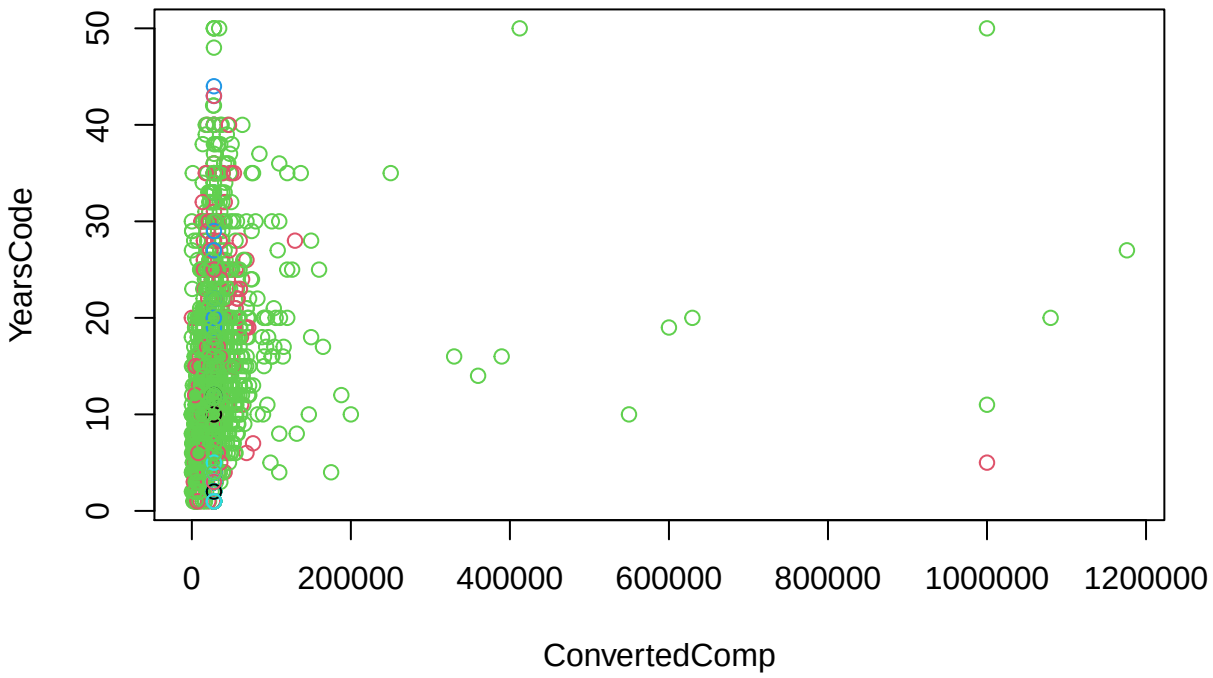
Vemos que de igual manera no se obtuvo una clara definición de los clusters. Ahora vamos a analizar otras 2 variables relación referente al desarrollador. Vamos a analizar **YearsCode** vs **ConvertedComp**

Comenzamos analizando para k=3

```
# YearsCode y ConvertedComp
dev3YCclust <- kmeans(dataNS, 3)

plot(dataNS[c(2,4)], col=dev3YCclust$cluster)
```




```
plot(dataNS[c(2,4)], col=as.factor(dataWithClass$MainBranch))
```

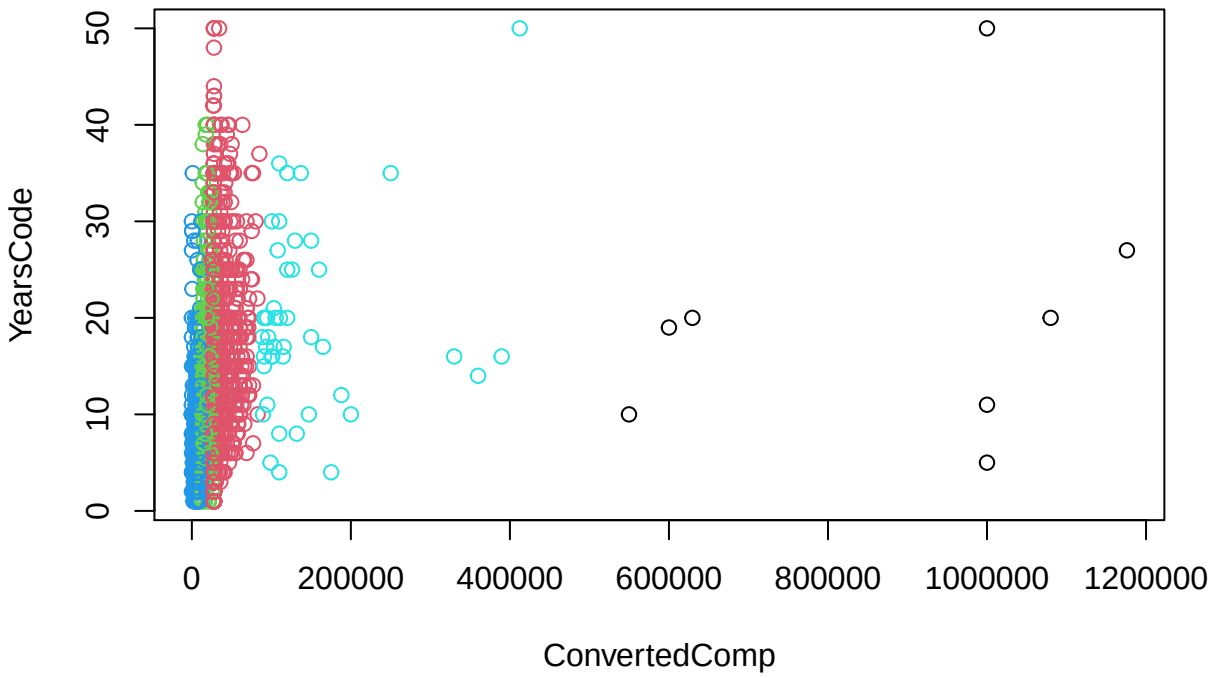


Vemos que se marcan o se diferencian ligeramente 3 grupos:

- El primero con un sueldo bajo pero con experiencia en programar inferior a 20 años, en su mayoría.
- El segundo grupo tiene un sueldo un poco más elevado, sobrepasando los 25K y llegando en ciertos casos a 400K. La experiencia es variada, alcanzando en algunos casos los 50 años.
- El tercer grupo son casos especiales y aislados de desarrolladores que tienen años de experiencias variados pero sus sueldos sobrepasan los 400K.

Ahora probamos para k=5

```
# YearsCode y ConvertedComp
dev5YCclust <- kmeans(dataNS, 5)

plot(dataNS[c(2,4)], col=dev5YCclust$cluster)
```



CONCLUSIONES:

Vemos que se marcan o se diferencian ligeramente 5 grupos:

- Los 2 primeros grupos que se diferencian tienen similitudes en cuanto al tiempo de experiencia (YearsCode) y con un sueldo bajo no superior a 25K.
- El tercer grupo tiene un sueldo un poco más elevado, sobrepasando los 25K y por debajo de los 100K. En algunos casos la experiencia sobrepasa los 40 años como desarrollador
- El cuarto grupo es más pequeño pero tiene sueldos superiores a 100K y llega hasta los 400K anuales. La experiencia en programación en este grupo está entre los 5 a 40 años aproximadamente.
- El último grupo son casos especiales y aislados de desarrolladores que tienen años de experiencias variados pero sus sueldos sobrepasan los 400K.

Ahora vamos a evaluar la calidad del proceso de agregación. Para ello usaremos la función silhouette que calcula la silueta de cada muestra. Para los valores de k=3, k=5 obtenemos que:

```
d  <- daisy(dataNS, metric="manhattan")
skc3 <- silhouette(dev3YCclust$cluster, d)
skc5 <- silhouette(dev5YCclust$cluster, d)
```

La función silhouette devuelve para cada muestra, el clúster dónde ha sido asignado, el clúster vecino y el valor de la silueta. Por lo tanto, calculando la media de la tercera columna podemos obtener una estimación de la calidad del agrupamiento

```
mean(skc3[,3])
```

```
## [1] 0.569479
```

```
mean(skc5[,3])
```

```
## [1] 0.5348022
```

Según los valores obtenidos previamente, sobre la calidad del clustering para los valores de k=3 y k=5, tenemos que agrupar los valores en 3 (56.9%) clusters es mejor que en 5 (53.5%).

Para finalizar con lo solicitado en este enunciado vamos a nombrar los grupos o clusters, según las conclusiones obtenidas previamente:

Si tomamos como referencia que las variables que permitieron agruparlos son: **YearsCode** y **ConvertedComp**

- Grupo 1: Desarrolladores con ingresos bajos y experiencia menor a 40 años.
- Grupo 2: Desarrolladores con ingresos medios y experiencia menor a 50 años.
- Grupo 3: Desarrolladores con ingresos medios y experiencia menor a 50 años.

Esto nos ayuda a entender cómo están formados los grupos y a referirnos a ellos en análisis posteriores.

15%. Se genera un modelo supervisado sin PCA/SVD previo, se muestran y comentan medidas de calidad del modelo gener

4. Aplicar un **modelo supervisado** sobre el juego de datos **sin** haber aplicado previamente **PCA/SVD**.

Ahora procedemos a elegir las variables categóricas del dataset *dataSudamerica*. Seleccionaremos las variables que formarán parte del dataset que se evaluará con el **método supervisado Redes Neuronales**

```
# Creamos un juego de datos resumido
dataNN <- cbind(dataNS, dataWithClass$MainBranch)


dataNN$clase[dataNN$`dataWithClass$MainBranch` == "PRO"] <- 1
dataNN$clase[dataNN$`dataWithClass$MainBranch` == "PRO_RETIRED"] <- 1
dataNN$clase[dataNN$`dataWithClass$MainBranch` == "NOT_PRO"] <- 0
dataNN$clase[dataNN$`dataWithClass$MainBranch` == "AMATEUR"] <- 0
dataNN$clase[dataNN$`dataWithClass$MainBranch` == "STUDENT"] <- 0

dataNN$`dataWithClass$MainBranch` = NULL

dataNN$clase <- as.numeric(dataNN$clase)

#Visualizar un resumen de los datos
summary(dataNN)
```

```
##       Age         ConvertedComp      WorkWeekHrs       YearsCode
##  Min.   :10.00   Min.   :      0   Min.   :  1.00   Min.   : 1.00
##  1st Qu.:25.00   1st Qu.:  14400   1st Qu.: 39.01   1st Qu.: 6.00
##  Median :29.98   Median :  27794   Median : 40.00   Median :11.00
##  Mean   :29.98   Mean   :  27794   Mean   : 39.01   Mean   :11.74
##  3rd Qu.:32.00   3rd Qu.:  27794   3rd Qu.: 40.00   3rd Qu.:15.00
##  Max.   :74.00   Max.   :1176000   Max.   :425.00   Max.   :50.00
##     db_techs        prog_langs       misc_techs      collab_techs
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 1.000   1st Qu.: 3.000   1st Qu.: 0.000   1st Qu.: 1.000
```

```
##  Median : 2.000   Median : 4.000   Median : 1.000   Median : 3.000
##  Mean   : 2.453   Mean   : 4.433   Mean   : 1.423   Mean   : 3.153
##  3rd Qu.: 4.000   3rd Qu.: 6.000   3rd Qu.: 2.000   3rd Qu.: 5.000
##  Max.   :12.000   Max.   :25.000   Max.   :11.000   Max.   :10.000
##    plat_techs       web_techs          clase
##  Min.   : 0.000   Min.   : 0.000   Min.   :0.0000
##  1st Qu.: 1.000   1st Qu.: 0.000   1st Qu.:1.0000
##  Median : 2.000   Median : 2.000   Median :1.0000
##  Mean   : 2.917   Mean   : 2.099   Mean   :0.7869
##  3rd Qu.: 4.000   3rd Qu.: 3.000   3rd Qu.:1.0000
##  Max.   :13.000   Max.   :14.000   Max.   :1.0000
```

```
head(dataNN)
```

```
##          Age ConvertedComp WorkWeekHrs YearsCode db_techs prog_langs misc_techs
## 44  32.00000      55893.00    45.00000        10        1          5          0
## 59  38.00000      16488.00    40.00000        24        4          5          3
## 189 29.97712      27794.04    40.00000        20        3          4          2
## 240 28.00000       8244.00    44.00000         5        0          2          2
## 290 18.00000      27794.04    39.01358         5        3          2          2
## 309 27.00000       8712.00    40.00000        10        6          9          7
##     collab_techs plat_techs web_techs clase
## 44             3          1         0     1
## 59             3          2         4     1
## 189            6          4         1     1
## 240            1          2         1     1
## 290            2          6         1     0
## 309            3          5         2     1
```

Datos de entrenamiento y prueba

```
# Transferimos la variable objetivo a la primera posición de la tabla
#dataNN[,1] <- as.numeric(dataNN[,11])

#Separamos los datos de las etiquetas
labels <- dataNN[,11]
data <- dataNN[,1:11]

#Seleccionamos el 80% para entrenamiento y el 20% para test
NObs <-  nrow(data)
NTrain <- round(NObs*0.8)
NTest <- NObs - NTrain

#Definir el conjunto de entrenamiento:
train <- data[1:NTrain,]
labelsTrain <- labels[1:NTrain]

#Definir el conjunto de test:
test <- data[(NTrain+1):NObs,]
labelsTest <- labels[(NTrain+1):NObs]
```

```
head(train)
```

```
##          Age ConvertedComp WorkWeekHrs YearsCode db_techs prog_langs misc_techs
## 44  32.00000      55893.00    45.00000        10        1          5          0
## 59  38.00000      16488.00    40.00000        24        4          5          3
## 189 29.97712      27794.04    40.00000        20        3          4          2
## 240 28.00000       8244.00    44.00000         5        0          2          2
## 290 18.00000      27794.04    39.01358         5        3          2          2
## 309 27.00000       8712.00    40.00000        10        6          9          7
##     collab_techs plat_techs web_techs clase
## 44             3          1         0     1
## 59             3          2         4     1
## 189            6          4         1     1
## 240            1          2         1     1
## 290            2          6         1     0
## 309            3          5         2     1
```

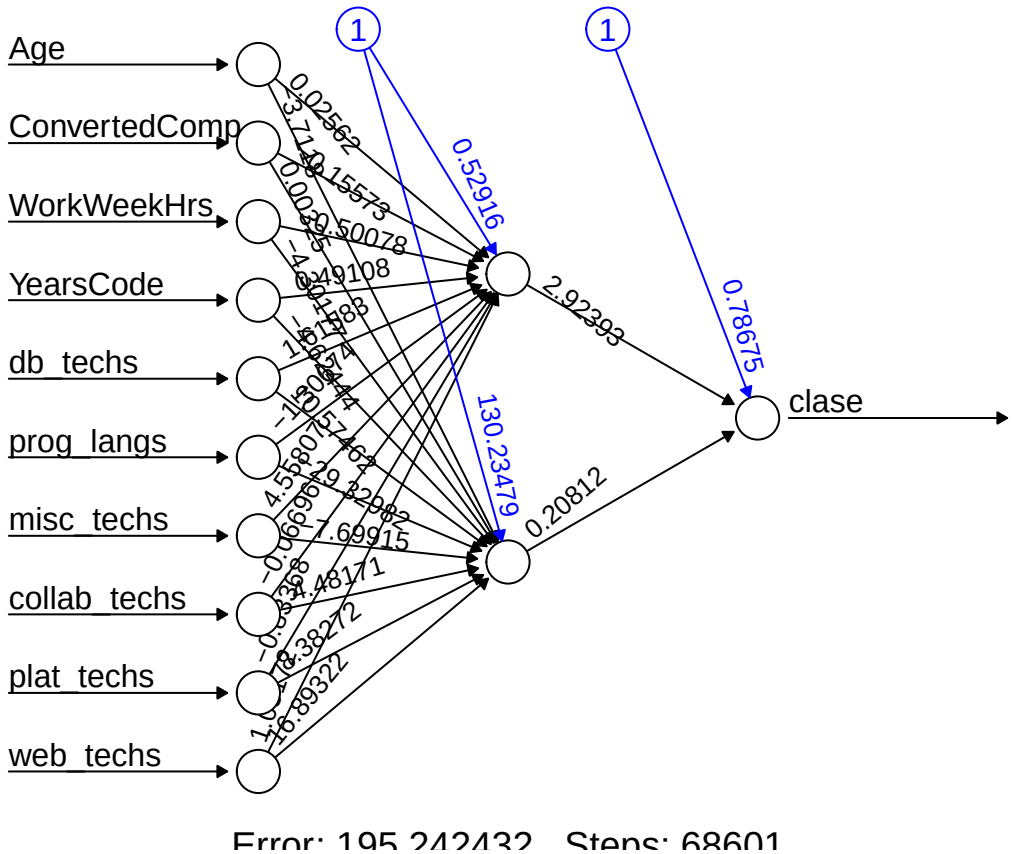
Aplicación del modelo supervisado

Modelo con una capa de 2 neuronas y con un criterio de parada de 0.01

```
require(neuralnet)

set.seed(10)
neural_net <- neuralnet(clase ~ ., data=train, hidden=2, threshold=0.01)

# Obtenemos un resumen del nivel de precisión del modelo
neural_net$result.matrix[1:3,]
```

```
##             error reached.threshold             steps
##      1.952424e+02      8.318361e-03      6.860100e+04
```

```
#graficamos la red
plot(neural_net, rep="best")
```



Vamos ahora a calcular la predicción y el grado de acierto

```
# Predicción
nn_predict1 <- compute(neural_net, test[,1:10])
pred1 <- round(nn_predict1$net.result)

# Generamos una matriz de confusión para medir el
# grado de acierto de la predicción
t1 <- table(pred = pred1, real = test[,11])
t1
```

```
##     real
## pred   0   1
##    1 156 456
```

```
round(100 * sum(diag(t1)) / sum(t1), digits = 2)
```

```
## [1] 25.49
```

Conclusión:

El grado de acierto de la predicción según la matriz de confusión es del 100%

15%. Se genera un modelo supervisado con PCA/SVD previo, se muestran y comentan medidas de calidad del modelo gene

5. Aplicar un **modelo supervisado** sobre el juego de datos habiendo aplicado previamente **PCA/SVD**.

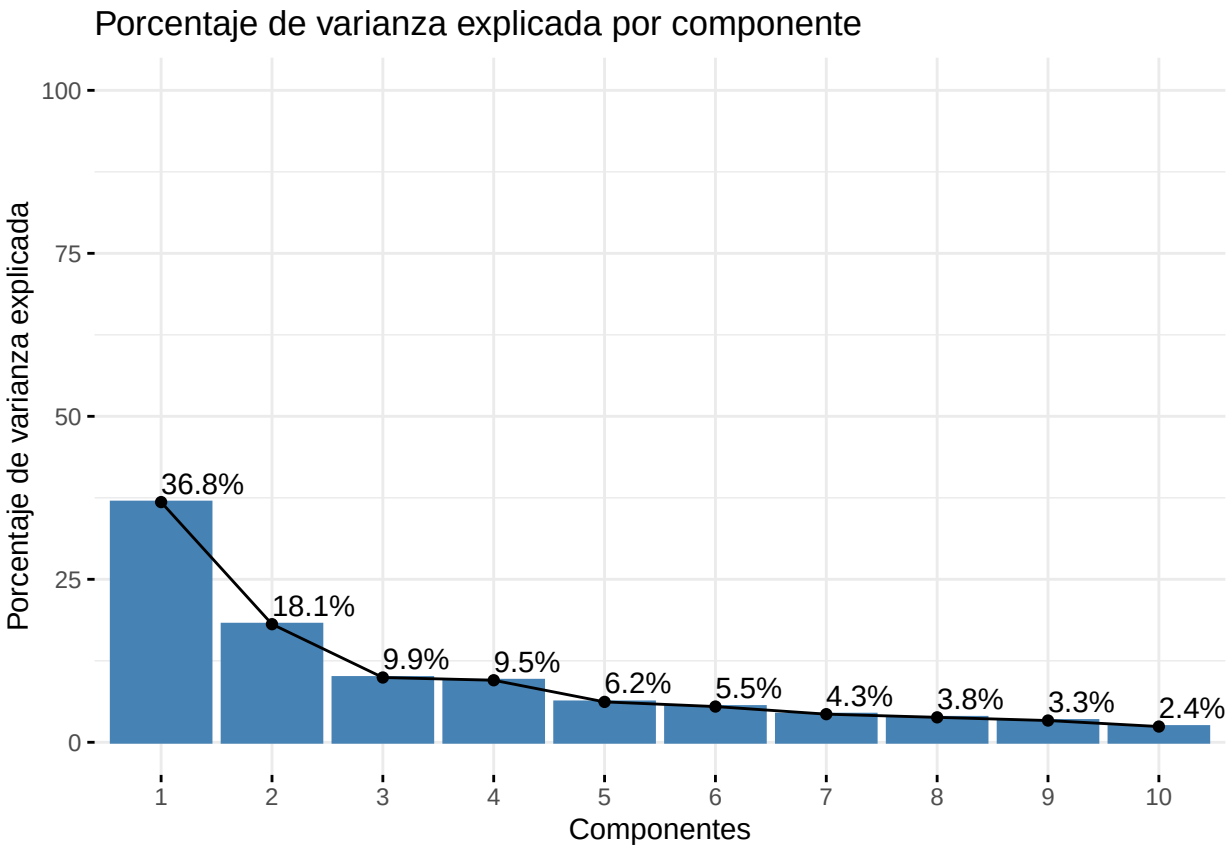
```
dataPCA <- dataNN[, 1:10]

# Aplicamos PCA a la matriz A
pca <- prcomp(dataPCA, center = TRUE, scale = TRUE)

summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.9194 1.3460 0.99687 0.9757 0.78721 0.74056 0.65699
## Proportion of Variance 0.3684 0.1812 0.09938 0.0952 0.06197 0.05484 0.04316
## Cumulative Proportion  0.3684 0.5496 0.64897 0.7442 0.80614 0.86098 0.90414
##                           PC8     PC9    PC10
## Standard deviation     0.61858 0.57803 0.49172
## Proportion of Variance 0.03826 0.03341 0.02418
## Cumulative Proportion  0.94241 0.97582 1.00000
```

La reducción de la dimensionalidad a n componentes que cubran una proporción de varianza, superior al 90%, en base al resumen del objeto *pca*, nos deja que entre el componente 7 (PC7) y el componente 8 (PC8) se alcanza este umbral.

También podemos ver los componentes de manera visual:

```
library(factoextra)

fviz_eig(
    X         = pca,
    choice    = "variance",
    addlabels = TRUE,
    ncp       = 11,
    ylim      = c(0, 100),
    main      = "Porcentaje de varianza explicada por componente",
    xlab      = "Componentes",
    ylab      = "Porcentaje de varianza explicada"
)
```



Para efectos de la continuación de la práctica tomaremos la matriz transformada **x** del objeto **pca**. Para que pueda ser usada en la aplicación del metodo supervisado tomando como referencia la matriz reducida se procederá a tomar los primeros 8 componentes y agregarle la variable clase al final para proceder a ejecutar nuevamente el modelo.

```
# Seleccionamos los 8 PC
X = pca$x[, 1:8]

# Unimos los componentes o matriz reducida con la clase
X <- cbind(X, dataNN[, 11])

# renombramos la columna sin nombre
colnames(X)[9] <- "clase"

tail(X)
```

```
##              PC1         PC2         PC3         PC4         PC5         PC6
## 64122 -0.2724635  1.54378530  1.60429082  0.72584015 -0.2690857 -1.01891198
## 64140 -0.9709292  0.12683996 -0.06160882 -0.03099991 -0.6013023 -1.58096101
## 64145 -3.0227663 -0.57848024  0.03269767 -0.13456674 -0.2610566  0.07520341
## 64149 -2.9660296  0.08224622  0.05992805  0.17423484 -0.3011031  0.07142875
## 64154 -3.0074719 -0.33875809  0.46728184  0.14376998 -0.2705362  0.05722351
## 64160 -2.9643885  0.09429856 -0.02263957  0.01255964 -0.3077681  0.07679603
##               PC7        PC8 clase
## 64122 -0.08926084 -0.6212892     1
## 64140  0.86222777 -0.3401554     1
```

```
## 64145 -0.33229444  0.3690595     1
## 64149 -0.21289099  0.2984875     1
## 64154 -0.27798553  0.3318356     1
## 64160 -0.22724154  0.3080203     1
```

Datos de entrenamiento y prueba luego de aplicar PCA

```
#Separamos los datos de las etiquetas
labelsPCA <- X[,9]
dataACP <- X[,1:9]

#Seleccionamos el 80% para entrenamiento y el 20% para test
NObsPCA <-  nrow(dataACP)
NTrainPCA <- round(NObsPCA*0.8)
NTestPCA <- NObsPCA - NTrainPCA

#Definir el conjunto de entrenamiento:
trainPCA <- dataACP[1:NTrainPCA,]
labelsTrainPCA <- labelsPCA[1:NTrainPCA]

#Definir el conjunto de test:
testPCA <- dataACP[(NTrainPCA+1):NObsPCA,]
labelsTestPCA <- labelsPCA[(NTrainPCA+1):NObsPCA]


head(trainPCA)
```

```
##             PC1        PC2         PC3        PC4         PC5        PC6
## 44  -1.37334496  0.3234812  0.35865995 -0.6373915  0.38964498 -0.2926020
## 59   1.06291071  1.6542810 -0.03372932  0.7935061 -0.90578880  0.2979789
## 189  0.68781108  0.7575974  0.06477985  0.1178637  0.93417150  0.7187198
## 240 -1.52078358 -0.8524096  0.46272915  0.1806205 -0.72594274  0.7348190
## 290  0.00886149 -1.6646823  0.17657799 -0.4268834 -0.00971123  0.1000497
## 309  3.05729932 -0.6802799  0.24909965  0.3914412 -2.03970249  0.9487554
##            PC7        PC8 clase
## 44   0.4160539 -0.7502356     1
## 59  -0.7314924 -0.3075854     1
## 189  0.6053634 -0.0439300     1
## 240  0.1668701  0.3922962     1
## 290  0.7571441  1.6095494     0
## 309  1.3291725 -0.0746684     1
```

Aplicación del modelo supervisado luego de aplicado PCA

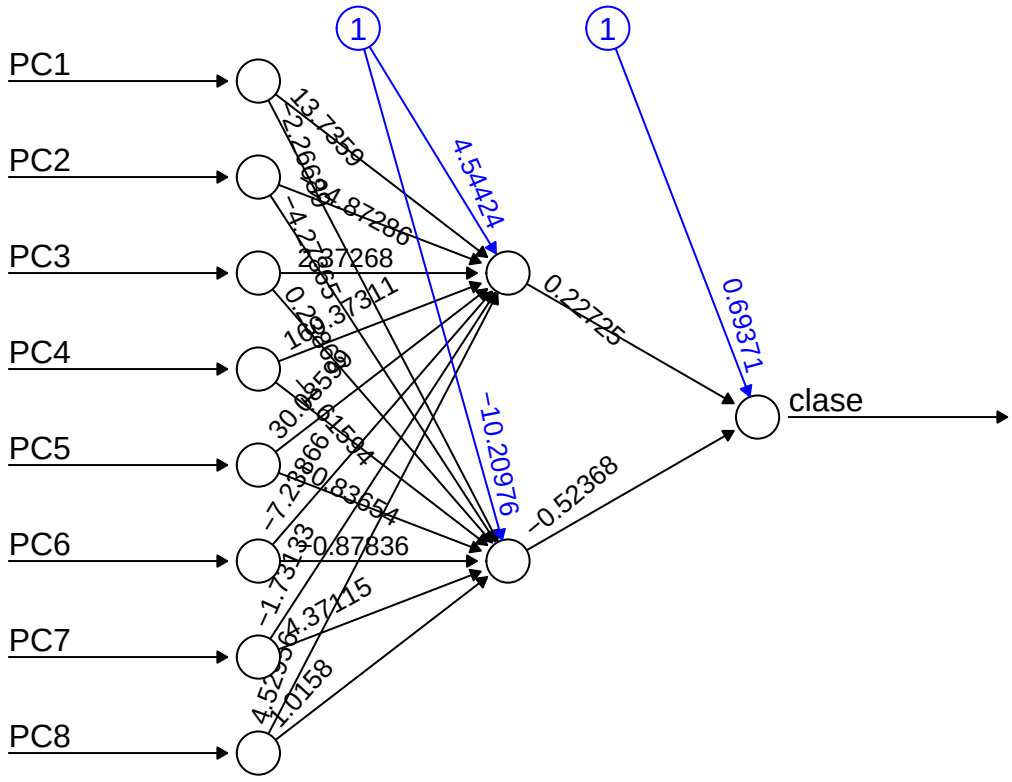
Modelo con una capa de 2 neuronas y con un criterio de parada de 0.01

```
set.seed(10)
neural_netPCA <- neuralnet(clase ~ ., data=trainPCA, hidden=2, threshold=0.01)
```

```
# Obtenemos un resumen del nivel de precisión del modelo
neural_netPCA$result.matrix[1:3,]
```

```
##                error reached.threshold              steps
##       1.607536e+02      9.828253e-03      2.841000e+03
```

```
#graficamos la red
plot(neural_netPCA, rep="best")
```



Vamos ahora a calcular la predicción y el grado de acierto

```
# Predicción
nn_predictPCA <- compute(neural_netPCA, testPCA[,1:8])
predPCA <- round(nn_predictPCA$net.result)

# Generamos una matriz de confusión para medir el
# grado de acierto de la predicción
t1PCA <- table(pred = predPCA, real = testPCA[,9])
t1PCA
```

```
##     real
## pred   0   1
##    0  28   7
##    1 128 449
```

```
round(100 * sum(diag(t1PCA)) / sum(t1PCA), digits = 2)
```

```
## [1] 77.94
```

20%. Se compara la capacidad predictiva de los dos modelos supervisados y se comenta la diferencia de rendimiento en base

6. ¿Ha habido mejora en capacidad predictiva, tras aplicar PCA/SVD? ¿A qué crees que es debido?.

Conclusión:

- El grado de acierto de la predicción según la matriz de confusión es del 100%. No hubo mejora entre la aplicación del modelo de **Red Neuronal** sin PCA y con PCA (luego de reducida la matriz). Esto se debe a que PCA o SVD reducen la matriz inicial u original tratando de conservar en cada uno de sus componentes una representación distinta del dataset original. Al haber seleccionado 8 componentes garantizamos que luego de aplicar PCA se mantendría más del 90% de esa matriz inicial para que no haya mayor variación
- La matriz reducida luego de aplicar PCA tiene las dimensiones 3059x8, es decir se conservan las filas y **las variables iniciales que eran 10 se han transformado en 8 componentes**, que tienen un 94.24% de representatividad de la matriz inicial (antes de aplicar PCA).
- Vemos que la red neuronal realizó la predicción perfecta al 100%, *es decir para la clase 0, ubicó 156 en la clase 0* (Desarrollador no profesional) y 0 observaciones en la clase 1; y *para la clase #1, ubicó 456 en la clase 1 (desarrollador profesional) sin calcular ninguna predicción en la clase 0*

Rúbrica

- 15%. Se generan reglas y se comentan e interpretan las más significativas. Adicionalmente se genera matriz de confusión para medir la capacidad predictiva del algoritmo.
- 15%. Se genera modelo no supervisado, se muestran y comentan medidas de calidad del modelo generado y se comentan las conclusiones.
- 20%. Se genera modelo no supervisado con métrica de distancia distinta al anterior. Se muestran y comentan medidas de calidad del modelo generado y se comentan las conclusiones. Adicionalmente se comparan los dos modelos no supervisados con métricas de distancia distinta.
- 15%. Se genera un modelo supervisado sin PCA/SVD previo, se muestran y comentan medidas de calidad del modelo generado y se comenta extensamente el conocimiento extraído del modelo.
- 15%. Se genera un modelo supervisado con PCA/SVD previo, se muestran y comentan medidas de calidad del modelo generado y se comenta extensamente el conocimiento extraído del modelo.
- 20%. Se compara la capacidad predictiva de los dos modelos supervisados y se comenta la diferencia de rendimiento en base al efecto PCA/SVD.

Recursos de programación

- Incluimos en este apartado una lista de recursos de programación para minería de datos donde podréis encontrar ejemplos, ideas e inspiración:
  - Material adicional del libro: Minería de datos Modelos y Algoritmos
  - Espacio de recursos UOC para ciencia de datos
  - Buscador de código R
  - Colección de cheatsheets en R
