

# Práctica 2 - Tipología y ciclo de vida de los datos

Autores: Luis Arnaldo Torres González, Gabriel Patricio Bonilla Sanchez

Junio - 2021

## Índice general

<b>Introducción</b>	<b>1</b>
Competencias . . . . .	2
Objetivos . . . . .	2
<b>1 Descripción del dataset</b>	<b>2</b>
<b>2 Integración y selección de los datos de interés a analizar</b>	<b>9</b>
<b>3 Limpieza de los datos</b>	<b>10</b>
3.1 Tratamiento de valores nulos o ceros . . . . .	11
3.2 Identificación y tratamiento de valores extremos. . . . .	23
<b>4 Análisis de los datos</b>	<b>31</b>
4.1 Selección de los grupos de datos a analizar . . . . .	31
4.2 Comprobación de la normalidad y homogeneidad de la varianza . . . . .	33
4.3 Pruebas estadísticas . . . . .	46
<b>5 Representación de los resultados a partir de tablas y gráficas</b>	<b>53</b>
<b>6 Conclusiones</b>	<b>55</b>
<b>7 Código</b>	<b>55</b>

## Introducción

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas. Para hacer esta práctica tendréis que trabajar en grupos de 2 personas. Tendréis que entregar un solo archivo con el enlace Github (<https://github.com>) donde se encuentren las soluciones incluyendo los nombres de los componentes del equipo. Podéis utilizar la Wiki de Github para describir vuestro equipo y los diferentes archivos que corresponden a vuestra entrega. Cada miembro del equipo tendrá que contribuir con su usuario Github. Aunque no se trata del mismo enunciado, los siguientes ejemplos de ediciones anteriores os pueden servir como guía:

- Ejemplo: <https://github.com/Bengis/nba-gap-cleaning>
- Ejemplo complejo (archivo adjunto).

## Competencias

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

## Objetivos

Los objetivos concretos de esta práctica son:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

## 1 Descripción del dataset

En este apartado resumiremos 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 países. 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:

- 1) *Respondent*: número de identificación del encuestado aleatorizado (no en orden de tiempo de respuesta de la encuesta)
- 2) *MainBranch*: ¿Cuál de las siguientes opciones te describe mejor hoy?
- 3) *Hobbyist*: ¿Desarrollas como pasatiempo?
- 4) *Age*: ¿Cuál es su edad (en años)?
- 5) *Age1stCode*: ¿A qué edad escribiste tu primera línea de código o programa?
- 6) *CompFreq*: ¿Esa compensación es semanal, mensual o anual?
- 7) *CompTotal*: ¿Cuál es su compensación total actual (salario, bonificaciones y beneficios, antes de impuestos y deducciones), en “CurrencySymbol”? Número entero.

- 8) *ConvertedComp*: Salario anual en USD, utilizando el tipo de cambio del 19 de febrero de 2020, asumiendo 12 meses laborales y 50 semanas laborales”.
- 9) *Country*: País dónde vive.
- 10) *CurrencyDesc*: ¿Qué moneda utiliza a diario? Descripción.
- 11) *CurrencySymbol*: ¿Qué moneda usa a diario? Forma abreviada.
- 12) *DatabaseDesireNextYear*: ¿En qué entornos de base de datos desea trabajar durante el próximo año?
- 13) *DatabaseWorkedWith*: ¿En qué entornos de base de datos ha realizado un trabajo de desarrollo extenso durante el año pasado?
- 14) *DevType*: ¿Cuál de los siguientes lo describe?
- 15) *EdLevel*: ¿Cuál de las siguientes opciones describe mejor el nivel más alto de educación formal que ha completado?
- 16) *Employment*: ¿cuál de las siguientes opciones describe mejor su situación laboral actual?
- 17) *Ethnicity*: ¿Cuál de los siguientes grupos étnicos lo describe?
- 18) *Gender*: ¿Cuál de las siguientes opciones de sexo lo describe?
- 19) *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?
- 20) *JobSat*: ¿Qué tan satisfecho está con su trabajo actual?
- 21) *JobSeek*: ¿Cuál de las siguientes opciones describe mejor su estado actual de búsqueda de empleo?
- 22) *LanguageDesireNextYear*: ”¿En qué lenguajes de programación, scripting y marcado desea trabajar durante el próximo año?.
- 23) *LanguageWorkedWith*: ¿En qué lenguajes de programación, scripting y marcado ha realizado un trabajo de desarrollo extenso durante el año pasado?.
- 24) *MiscTechDesireNextYear*: ¿En qué otros frameworks, bibliotecas y herramientas desea trabajar durante el próximo año?.
- 25) *MiscTechWorkedWith*: ¿En qué otros frameworks, bibliotecas y herramientas ha realizado un trabajo de desarrollo extenso durante el año pasado?.
- 26) *NEWCollabToolsDesireNextYear*: ¿En qué herramientas de colaboración desea trabajar durante el próximo año?
- 27) *NEWCollabToolsWorkedWith*: ¿En qué herramientas de colaboración ha realizado un trabajo de desarrollo extenso durante el año pasado?
- 28) *NEWDevOps*: ¿Su empresa tiene una persona dedicada a DevOps?
- 29) *NEWDevOpsImpt*: ¿Qué importancia tiene la práctica de DevOps para escalar el desarrollo de software?
- 30) *NEWEdImpt*: ¿Qué importancia tiene una educación formal, como un título universitario en ciencias de la computación, para su carrera?
- 31) *NEWJobHunt*: En general, ¿Cuáles son las motivaciones que lo impulsan a buscar un nuevo trabajo?.
- 32) *NEWJobHuntResearch*: Cuando busca trabajo, ¿cómo puede obtener más información sobre una empresa?
- 33) *NEWLearn*: ¿Con qué frecuencia aprende un nuevo lenguaje o marco?
- 34) *NEWOffTopic*: ¿Crees que Stack Overflow debería relajar las restricciones sobre lo que se considera “fuera de tema”?
- 35) *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)
- 36) *NEWOtherComms*: ¿Es miembro de alguna otra comunidad de desarrolladores en línea?
- 37) *NEWOvertime*: ¿Con qué frecuencia trabaja horas extraordinarias o más allá de las expectativas formales de su trabajo?
- 38) *NEWPurchaseResearch*: Al comprar una nueva herramienta o software, ¿cómo descubre e investiga las soluciones disponibles?
- 39) *NEWPurpleLink*: Busca una solución de codificación en línea y el primer enlace de resultado es violeta porque ya lo visitó. ¿Cómo se siente?
- 40) *NEWSOSites*: ¿Cuál de los siguientes sitios de Stack Overflow ha visitado?
- 41) *NEWStuck*: ¿Qué hace cuando se queda atascado en un problema?
- 42) *OpSys*: ¿Cuál es el sistema operativo principal en el que trabaja?
- 43) *OrgSize*: Aproximadamente, ¿cuántas personas emplea la empresa u organización para la que trabaja actualmente?
- 44) *PlatformDesireNextYear*: ¿En qué plataformas desea trabajar durante el próximo año?

- 45) *PlatformWorkedWith*: ¿En qué plataformas ha realizado un trabajo de desarrollo extenso durante el año pasado?
- 46) *PurchaseWhat*: ¿Qué nivel de influencia tiene usted, personalmente, sobre las compras de nueva tecnología en su organización?
- 47) *Sexuality*: ¿Cuál de los siguientes lo describe a usted sobre su sexualidad?.
- 48) *SOAccount*: ¿Tiene una cuenta de Stack Overflow?
- 49) *SOCComm*: ¿Te consideras miembro de la comunidad de Stack Overflow?
- 50) *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.
- 51) *SOVisitFreq*: ¿Con qué frecuencia visita Stack Overflow?
- 52) *SurveyEase*: ¿Qué tan fácil o difícil fue completar esta encuesta?
- 53) *SurveyLength*: ¿Qué opina de la duración de la encuesta este año?
- 54) *Trans*: ¿Eres transgénero?
- 55) *UndergradMajor*: ¿Cuál fue su campo de estudio principal?
- 56) *WebframeDesireNextYear*: ¿En qué frameworks web desea trabajar durante el próximo año?
- 57) *WebframeWorkedWith*: ¿En qué frameworks web ha realizado un extenso trabajo de desarrollo durante el año pasado?
- 58) *WelcomeChange*: En comparación con el año pasado, ¿qué tan bienvenido se siente en Stack Overflow?
- 59) *WorkWeekHrs*: En promedio, ¿cuántas horas por semana trabaja?
- 60) *YearsCode*: Incluyendo cualquier educación, ¿cuántos años ha estado programando en total?
- 61) *YearsCodePro*: NO incluye educación, ¿cuántos años ha programado profesionalmente (como parte de su trabajo)?

Las capacidades analíticas 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 numéricas que representen el número de tecnologías que domina cada encuestado.
- Al incluir las tecnologías usadas por desarrolladores en: base de datos, lenguajes 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. Age
2. ConvertedComp
3. Country
4. DatabaseWorkedWith
5. EdLevel
6. Employment
7. LanguageWorkedWith
8. MiscTechWorkedWith
9. NEWCollabToolsWorkedWith
10. NEWLearn
11. NEWOvertime
12. OpSys
13. PlatformWorkedWith
14. SOPartFreq

15. UndergradMajor
16. WebframeWorkedWith
17. WorkWeekHrs
18. 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.

Vamos a trabajar preliminarmente con 19 variables propias del dataset original, de las cuales 4 son numéricas (Age, ConvertedComp, WorkWeekHrs y YearsCodePro). También tenemos variables no numéricas, las cuales vamos a realizar un análisis más detallado posteriormente, generando variables numéricas a partir de ellas, las cuales son:

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

Estas nuevas variables numéricas a generarse posteriormente servirán principalmente cuando se defina el número total de tecnologías conocidas y usadas por los encuestados.

Este dataset ayudará a dar respuesta a la siguientes interrogantes:

¿Existe una fuerte correlación entre las variables seleccionadas en nuestro dataset? ¿Cuáles son las variables que influyen en el salario de los entrevistados de la comunidad Stack Overflow?

```
# Cargamos el dataset completo
data_devs <- read.csv('../data/survey_results_public.csv', sep=",", encoding = "UTF-8")

# Resumen del dataset original
head(data_devs)
```

```
##   Respondent                               MainBranch Hobbyist
## 1           1           I am a developer by profession Yes
## 2           2           I am a developer by profession No
## 3           3           I code primarily as a hobby Yes
## 4           4           I am a developer by profession Yes
## 5           5 I used to be a developer by profession, but no longer am Yes
## 6           6           I am a developer by profession No
##   Age Age1stCode CompFreq CompTotal ConvertedComp          Country
## 1  NA      13 Monthly       NA       NA        Germany
## 2  NA      19    <NA>       NA       NA United Kingdom
## 3  NA      15    <NA>       NA       NA Russian Federation
## 4  25      18    <NA>       NA       NA        Albania
## 5  31      16    <NA>       NA       NA United States
## 6  NA      14    <NA>       NA       NA        Germany
##   CurrencyDesc CurrencySymbol DatabaseDesireNextYear
## 1 European Euro             EUR Microsoft SQL Server
## 2 Pound sterling            GBP             <NA>
## 3 <NA>                   <NA>             <NA>
## 4 Albanian lek              ALL             <NA>
## 5 <NA>                   <NA> MySQL;PostgreSQL
## 6 European Euro              EUR             <NA>
##                               DatabaseWorkedWith
```

```

## 1 Elasticsearch;Microsoft SQL Server;Oracle
## 2 <NA>
## 3 <NA>
## 4 <NA>
## 5 MySQL;PostgreSQL;Redis;SQLite
## 6 <NA>

## DevType
## 1 Developer, desktop or enterprise applications;Developer, full-stack
## 2 Developer, full-stack;Developer, mobile
## 3 <NA>
## 4 <NA>
## 5 <NA>
## 6 Designer;Developer, front-end;Developer, mobile

## EdLevel
## 1 Master's degree (M.A., M.S., M.Eng., MBA, etc.)
## 2 Bachelor's degree (B.A., B.S., B.Eng., etc.)
## 3 <NA>
## 4 Master's degree (M.A., M.S., M.Eng., MBA, etc.)
## 5 Bachelor's degree (B.A., B.S., B.Eng., etc.)
## 6 Secondary school (e.g. American high school, German Realschule or Gymnasium, etc.)

## Employment
## 1 Independent contractor, freelancer, or self-employed
## 2 Employed full-time
## 3 <NA>
## 4 <NA>
## 5 Employed full-time
## 6 Employed full-time

## Ethnicity Gender
## 1 White or of European descent Man
## 2 <NA> <NA>
## 3 <NA> <NA>
## 4 White or of European descent Man
## 5 White or of European descent Man
## 6 White or of European descent Man

## Languages, frameworks, and other technologies I'd be working with;Remote work option
## 1
## 2
## 3
## 4 Flex time or a flexible schedule;Office environment or company culture
## 5
## 6 Diversity of the company or organization;Languages, frameworks, and other technologies I'd be working with

## JobSat
## 1 Slightly satisfied
## 2 Very dissatisfied
## 3 <NA>
## 4 Slightly dissatisfied
## 5 <NA>
## 6 Slightly satisfied

## JobSeek
## 1 I am not interested in new job opportunities
## 2 I am not interested in new job opportunities
## 3 <NA>
## 4 I'm not actively looking, but I am open to new opportunities
## 5 <NA>
## 6 I am not interested in new job opportunities

## LanguageDesireNextYear LanguageWorkedWith
## 1 C#;HTML/CSS;JavaScript C#;HTML/CSS;JavaScript
## 2 Python;Swift JavaScript;Swift

```

```

## 3 Objective-C;Python;Swift Objective-C;Python;Swift
## 4 <NA> <NA>
## 5 Java;Ruby;Scala HTML/CSS;Ruby;SQL
## 6 HTML/CSS;Java;JavaScript HTML/CSS;Java;JavaScript
## MiscTechDesireNextYear MiscTechWorkedWith
## 1 .NET Core;Xamarin .NET;.NET Core
## 2 React Native;TensorFlow;Unity 3D React Native
## 3 <NA> <NA>
## 4 <NA> <NA>
## 5 Ansible;Chef Ansible
## 6 <NA> <NA>
## NEWCollabToolsDesireNextYear
## 1 Microsoft Teams;Microsoft Azure;Trello
## 2 Github;Slack
## 3 <NA>
## 4 <NA>
## 5 Github;Google Suite (Docs, Meet, etc)
## 6 Github;Slack
## NEWCollabToolsWorkedWith NEWDevOps
## 1 Confluence;Jira;Slack;Microsoft Azure;Trello No
## 2 Confluence;Jira;Github;Gitlab;Slack <NA>
## 3 <NA> <NA>
## 4 <NA> No
## 5 Confluence;Jira;Github;Slack;Google Suite (Docs, Meet, etc) <NA>
## 6 Confluence;Github;Slack;Trello Not sure
## NEWDevOpsImpt NEWEdImpt
## 1 Somewhat important Fairly important
## 2 <NA> Fairly important
## 3 <NA> <NA>
## 4 <NA> Not at all important/not necessary
## 5 <NA> Very important
## 6 <NA> Fairly important
## NEWJobHunt
## 1 <NA>
## 2 <NA>
## 3 <NA>
## 4 Curious about other opportunities;Wanting to work with new technologies
## 5 <NA>
## 6 <NA>
## NEWJobHuntResearch NEWLearn NEWOffTopic NEWOnboardGood NEWOtherComms
## 1 <NA> Once a year Not sure <NA> No
## 2 <NA> Once a year Not sure <NA> No
## 3 <NA> Once a decade <NA> <NA> No
## 4 <NA> Once a year Not sure Yes Yes
## 5 <NA> Once a year No <NA> Yes
## 6 <NA> Once a year No No No
## NEWOvertime
## 1 Often: 1-2 days per week or more
## 2 <NA>
## 3 <NA>
## 4 Occasionally: 1-2 days per quarter but less than monthly
## 5 <NA>
## 6 Never
## 
## 1
## 2
## 3
## 4

```

```

## 5 Start a free trial;Ask developers I know/work with;Visit developer communities like Stack Overflow
## 6
##      NEWPurpleLink
## 1          Amused
## 2          Amused
## 3          <NA>
## 4          <NA>
## 5 Hello, old friend
## 6          Amused
##
## 1
## 2
## 3
## 4
## 5 Stack Overflow (public Q&A for anyone who codes);Stack Exchange (public Q&A for a variety of topics)
## 6
##
## 1                               Visit Stack Overflow;Go for a walk or do other work and come back
## 2
## 3
## 4
## 5 Call a coworker or friend;Visit Stack Overflow;Watch help / tutorial videos;Do other work and come back
## 6                               Play games;Visit Stack Overflow;Watch help / tutorial videos
##
##      OpSys          OrgSize
## 1    Windows        2 to 9 employees
## 2    MacOS         1,000 to 4,999 employees
## 3 Linux-based           <NA>
## 4 Linux-based        20 to 99 employees
## 5    Windows          <NA>
## 6    Windows          <NA>
##
##      PlatformDesireNextYear
## 1    Android;iOS;Kubernetes;Microsoft Azure;Windows
## 2                      iOS;Kubernetes;Linux;MacOS
## 3
## 4
## 5 Docker;Google Cloud Platform;Heroku;Linux;Windows
## 6                      Android
##
##      PlatformWorkedWith          PurchaseWhat
## 1                      Windows          <NA>
## 2                      iOS      I have little or no influence
## 3
## 4
## 5 AWS;Docker;Linux;MacOS;Windows          <NA>
## 6    Android;Docker;WordPress          I have some influence
##
##      Sexuality SOAccount          SOComm
## 1 Straight / Heterosexual        No  No, not at all
## 2
## 3
## 4 Straight / Heterosexual        Yes Yes, definitely
## 5 Straight / Heterosexual        Yes  Yes, somewhat
## 6 Straight / Heterosexual        Yes  Yes, somewhat
##
##      SOPartFreq          SOVisitFreq
## 1
## 2 Less than once per month or monthly        Multiple times per day
## 3 A few times per month or weekly          Daily or almost daily
## 4 A few times per month or weekly          Multiple times per day
## 5 Less than once per month or monthly        A few times per month or weekly
## 6 A few times per month or weekly          A few times per week

```

```

##           SurveyEase          SurveyLength Trans
## 1 Neither easy nor difficult Appropriate in length   No
## 2                               <NA>      <NA>  <NA>
## 3 Neither easy nor difficult Appropriate in length  <NA>
## 4                               <NA>      <NA>   No
## 5           Easy            Too short   No
## 6 Neither easy nor difficult Appropriate in length  <NA>
##                                         UndergradMajor
## 1 Computer science, computer engineering, or software engineering
## 2 Computer science, computer engineering, or software engineering
## 3                                         <NA>
## 4 Computer science, computer engineering, or software engineering
## 5 Computer science, computer engineering, or software engineering
## 6                                         <NA>
##   WebframeDesireNextYear  WebframeWorkedWith
## 1           ASP.NET Core  ASP.NET;ASP.NET Core
## 2             <NA>          <NA>
## 3             <NA>          <NA>
## 4             <NA>          <NA>
## 5 Django;Ruby on Rails       Ruby on Rails
## 6           React.js        <NA>
##           WelcomeChange WorkWeekHrs YearsCode YearsCodePro
## 1 Just as welcome now as I felt last year        50     36     27
## 2 Somewhat more welcome now than last year      NA      7      4
## 3 Somewhat more welcome now than last year      NA      4    <NA>
## 4 Somewhat less welcome now than last year      40     7      4
## 5 Just as welcome now as I felt last year      NA     15      8
## 6                                         <NA>     NA      6      4

```

## 2 Integración y selección de los datos de interés a analizar

```

# Creamos un juego de datos resumido
data_so <- data_devs[, c(4, 8, 9, 13, 15, 16, 23, 25, 27, 33, 37, 42, 45, 50, 55, 57, 59, 61)]

head(data_so)

```

```

##   Age ConvertedComp          Country
## 1  NA         NA        Germany
## 2  NA         NA  United Kingdom
## 3  NA         NA Russian Federation
## 4  25         NA       Albania
## 5  31         NA  United States
## 6  NA         NA        Germany
##                                         DatabaseWorkedWith
## 1 Elasticsearch;Microsoft SQL Server;Oracle
## 2                               <NA>
## 3                               <NA>
## 4                               <NA>
## 5 MySQL;PostgreSQL;Redis;SQLite
## 6                               <NA>
##                                         EdLevel
## 1 Master's degree (M.A., M.S., M.Eng., MBA, etc.)
## 2 Bachelor's degree (B.A., B.S., B.Eng., etc.)      <NA>
## 3                                         <NA>
## 4 Master's degree (M.A., M.S., M.Eng., MBA, etc.)

```

```

## 5 Bachelor's degree (B.A., B.S., B.Eng., etc.)
## 6 Secondary school (e.g. American high school, German Realschule or Gymnasium, etc.)
##
## 1 Independent contractor, freelancer, or self-employed Employment LanguageWorkedWith
## 2 Employed full-time JavaScript;Swift
## 3 <NA> Objective-C;Python;Swift
## 4 <NA> <NA>
## 5 Employed full-time HTML/CSS;Ruby;SQL
## 6 Employed full-time HTML/CSS;Java;JavaScript

## MiscTechWorkedWith
## 1 .NET;.NET Core
## 2 React Native
## 3 <NA>
## 4 <NA>
## 5 Ansible
## 6 <NA>

## NEWCollabToolsWorkedWith NEWLearn
## 1 Confluence;Jira;Slack;Microsoft Azure;Trello Once a year
## 2 Confluence;Jira;Github;Gitlab;Slack Once a year
## 3 <NA> Once a decade
## 4 <NA> Once a year
## 5 Confluence;Jira;Github;Slack;Google Suite (Docs, Meet, etc) Once a year
## 6 Confluence;Github;Slack;Trello Once a year

## NEWOvertime OpSys
## 1 Often: 1-2 days per week or more Windows
## 2 <NA> MacOS
## 3 <NA> Linux-based
## 4 Occasionally: 1-2 days per quarter but less than monthly Linux-based
## 5 <NA> Windows
## 6 Never Windows

## PlatformWorkedWith SOPartFreq
## 1 Windows <NA>
## 2 iOS Less than once per month or monthly
## 3 <NA> A few times per month or weekly
## 4 <NA> A few times per month or weekly
## 5 AWS;Docker;Linux;MacOS;Windows Less than once per month or monthly
## 6 Android;Docker;WordPress A few times per month or weekly

## UndergradMajor
## 1 Computer science, computer engineering, or software engineering
## 2 Computer science, computer engineering, or software engineering
## 3 <NA>
## 4 Computer science, computer engineering, or software engineering
## 5 Computer science, computer engineering, or software engineering
## 6 <NA>

## WebframeWorkedWith WorkWeekHrs YearsCodePro
## 1 ASP.NET;ASP.NET Core 50 27
## 2 <NA> NA 4
## 3 <NA> NA <NA>
## 4 <NA> 40 4
## 5 Ruby on Rails NA 8
## 6 <NA> NA 4

```

### 3 Limpieza de los datos

### 3.1 Tratamiento de valores nulos o ceros

Se ha revisado que existen valores nulos en todas las variables seleccionadas, por lo que se procederá a filtrar los registros que no contengan valores *NA* en ninguna de las variables.

Además, en este apartado se ha decidido filtrar los valores que no representan una realidad en el estudio de la variable en cuestión. Por ejemplo un valor de 0 en el sueldo anual (*variable ConvertedComp*) de un desarrollador no representa la realidad. A pesar de ser un valor que puede ser tratado como un valor extremo, para esta ocasión se lo tratará como no válido.

```
# Eliminar filas con valores nulos
data_wo_na <- data_so[!is.na(data_so$Age), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$ConvertedComp), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$Country), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$DatabaseWorkedWith), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$EdLevel), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$Employment), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$LanguageWorkedWith), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$MiscTechWorkedWith), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$NEWCollabToolsWorkedWith), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$NEWLearn), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$NEWOvertime), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$OpSys), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$PlatformWorkedWith), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$SOPartFreq), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$UndergradMajor), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$WebframeWorkedWith), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$WorkWeekHrs), ]
data_wo_na <- data_wo_na[!is.na(data_wo_na$YearsCodePro), ]

# Campos con valores cero
data_wo_na <- data_wo_na[data_wo_na$ConvertedComp != 0, ]

head(data_wo_na)
```

```
##   Age ConvertedComp      Country          DatabaseWorkedWith
## 8    36     116000 United States      MariaDB;MySQL;Redis
## 10   22      32315 United Kingdom Microsoft SQL Server
## 11   23      40070 United Kingdom Firebase;MongoDB;PostgreSQL;SQLite
## 14   27      66000 United States      Firebase;Microsoft SQL Server
## 19   24      83400 United States      MariaDB;Microsoft SQL Server
## 33   39      63564    Belgium    MongoDB;MySQL;PostgreSQL;SQLite
##                               EdLevel      Employment
## 8 Bachelor's degree (B.A., B.S., B.Eng., etc.) Employed full-time
## 10 Master's degree (M.A., M.S., M.Eng., MBA, etc.) Employed full-time
## 11 Bachelor's degree (B.A., B.S., B.Eng., etc.) Employed full-time
## 14 Associate degree (A.A., A.S., etc.) Employed full-time
## 19 Bachelor's degree (B.A., B.S., B.Eng., etc.) Employed full-time
## 33 Bachelor's degree (B.A., B.S., B.Eng., etc.) Employed full-time
##                               LanguageWorkedWith
## 8                           Python;SQL
## 10                          HTML/CSS;Java;JavaScript;Python;SQL
## 11                           C#;JavaScript;Swift
## 14                          HTML/CSS;JavaScript;SQL;TypeScript
## 19 Bash/Shell/PowerShell;C#;HTML/CSS;JavaScript;SQL;TypeScript
## 33                          HTML/CSS;JavaScript;TypeScript
##                               MiscTechWorkedWith
## 8                  Ansible
```

```

## 10          Pandas
## 11          Node.js
## 14          Node.js
## 19 .NET;.NET Core;Node.js
## 33  Node.js;React Native
##
## 8          Confluence;Jira;Github;Slack;Microsoft Teams;Google Suite (Docs, Meet, etc)
## 10
## 11 Confluence;Jira;Github;Slack;Microsoft Teams;Microsoft Azure;Trello;Google Suite (Docs, Meet, etc)
## 14          Confluence;Jira;Github;Slack;Google Suite (Docs, Meet, etc)
## 19
## 33          Confluence;Jira;Github;Gitlab;Slack;Google Suite (Docs, Meet, etc)
##
##          NEWLearn          NEWOvertime
## 8      Once a year Occasionally: 1-2 days per quarter but less than monthly
## 10     Every few months          Often: 1-2 days per week or more
## 11     Every few months          Rarely: 1-2 days per year or less
## 14     Every few months Occasionally: 1-2 days per quarter but less than monthly
## 19      Once a year Occasionally: 1-2 days per quarter but less than monthly
## 33      Once a year      Sometimes: 1-2 days per month but less than weekly
##
##          OpSys
## 8  Linux-based
## 10    Windows
## 11    Windows
## 14    Windows
## 19    Windows
## 33    MacOS
##
##          PlatformWorkedWith
## 8
## 10          Docker
## 11          Android;Linux;Raspberry Pi;Windows
## 12          AWS;Heroku;iOS
## 14          Google Cloud Platform;Windows
## 19          Windows
## 33 AWS;Docker;Google Cloud Platform;Heroku;iOS;Kubernetes;MacOS;Raspberry Pi
##
##          SOPartFreq
## 8      Less than once per month or monthly
## 10          Multiple times per day
## 11 I have never participated in Q&A on Stack Overflow
## 12          A few times per week
## 19      Less than once per month or monthly
## 33      Less than once per month or monthly
##
##          UndergradMajor
## 8  Computer science, computer engineering, or software engineering
## 10          Mathematics or statistics
## 11 Computer science, computer engineering, or software engineering
## 14 Computer science, computer engineering, or software engineering
## 19 Computer science, computer engineering, or software engineering
## 33 Computer science, computer engineering, or software engineering
##
##          WebframeWorkedWith WorkWeekHrs YearsCodePro
## 8          Flask 39 13
## 10         Flask;jQuery 36 4
## 11 Angular;Angular.js;Django;React.js 40 2
## 14          Angular;Vue.js 40 1
## 19 Angular;Angular.js;ASP.NET Core 35 3
## 33 Angular;Angular.js;Django;Express;React.js 40 14

```

Analizamos el tipo de dato de las columnas antes mencionadas

```

library(dplyr)
#Vemos el tipo de dato de las variables
glimpse(data_wo_na)

## Rows: 12,973
## Columns: 18
## $ Age <dbl> 36, 22, 23, 27, 24, 39, 34, 35, 32, 22, 53...
## $ ConvertedComp <dbl> 116000, 32315, 40070, 66000, 83400, 63564, ...
## $ Country <chr> "United States", "United Kingdom", "United...
## $ DatabaseWorkedWith <chr> "MariaDB;MySQL;Redis", "Microsoft SQL Serv...
## $ EdLevel <chr> "Bachelor's degree (B.A., B.S., B.Eng., et...
## $ Employment <chr> "Employed full-time", "Employed full-time"...
## $ LanguageWorkedWith <chr> "Python;SQL", "HTML/CSS;Java;JavaScript;Py...
## $ MiscTechWorkedWith <chr> "Ansible", "Pandas", "Node.js", "Node.js", ...
## $ NEWCollabToolsWorkedWith <chr> "Confluence;Jira;Github;Slack;Microsoft Te...
## $ NEWLearn <chr> "Once a year", "Every few months", "Every ...
## $ NEWOvertime <chr> "Occasionally: 1-2 days per quarter but le...
## $ OpSys <chr> "Linux-based", "Windows", "Windows", "Wind...
## $ PlatformWorkedWith <chr> "Docker", "Android;Linux;Raspberry Pi;Wind...
## $ SOPartFreq <chr> "Less than once per month or monthly", "Mu...
## $ UndergradMajor <chr> "Computer science, computer engineering, o...
## $ WebframeWorkedWith <chr> "Flask", "Flask;jQuery", "Angular;Angular....
## $ WorkWeekHrs <dbl> 39, 36, 40, 40, 35, 40, 40, 40, 37, 35, 40...
## $ YearsCodePro <chr> "13", "4", "2", "1", "3", "14", "3", "12", ...

```

```

# Otra forma de ver el tipo de dato de cada columna
sapply(data_wo_na, class)

```

##	Age	ConvertedComp	Country
##	"numeric"	"numeric"	"character"
##	DatabaseWorkedWith	EdLevel	Employment
##	"character"	"character"	"character"
##	LanguageWorkedWith	MiscTechWorkedWith	NEWCollabToolsWorkedWith
##	"character"	"character"	"character"
##	NEWLearn	NEWOvertime	OpSys
##	"character"	"character"	"character"
##	PlatformWorkedWith	SOPartFreq	UndergradMajor
##	"character"	"character"	"character"
##	WebframeWorkedWith	WorkWeekHrs	YearsCodePro
##	"character"	"numeric"	"character"

Para la variable YearsCodePro observamos que el tipo de datos es “character”, por lo que contiene valores cualitativos: *More than 50 years* y *Less than 1 year*, por lo que vamos a transformarlos en valores numéricos o cuantitativos

```

# Convertimos los valores categoricos en numéricos para la variable YearsCodePro
data_temp <- data_wo_na[data_wo_na$YearsCodePro %in% c("More than 50 years", "Less than 1 year"), ]

dim(data_temp)

```

```

## [1] 375 18

```

```

# datosEcuador$YearsCode[datosEcuador$YearsCode=="Less than 1 year"] <- 1

```

Existen 375 valores con valores cualitativos que deben ser transformados a valores cuantitativos

```

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

# Finalmente convertimos dicha columna en númerica
data_wo_na$YearsCodePro <- as.numeric(data_wo_na$YearsCodePro)

filas_df=dim(data_wo_na) [1]

```

Verificamos nuevamente el tipo de dato de la columna YearsCodePro para validar que se convirtió correctamente de *character* a *numeric*

```

# Otra forma de ver el tipo de dato de cada columna
sapply(data_wo_na, class)

```

	Age	ConvertedComp	Country
##	"numeric"	"numeric"	"character"
##	DatabaseWorkedWith	EdLevel	Employment
##	"character"	"character"	"character"
##	LanguageWorkedWith	MiscTechWorkedWith	NEWCollabToolsWorkedWith
##	"character"	"character"	"character"
##	NEWLearn	NEWOvertime	OpSys
##	"character"	"character"	"character"
##	PlatformWorkedWith	SOPartFreq	UndergradMajor
##	"character"	"character"	"character"
##	WebframeWorkedWith	WorkWeekHrs	YearsCodePro
##	"character"	"numeric"	"numeric"

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**:

```

data_wo_na$db_techs <- 0

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

```

data_wo_na$prog_langs <- 0

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

```

```
    }
}
```

Ahora vamos a agregar una nueva variable para el número de frameworks, librerías y demás herramientas 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**:

```
data_wo_na$misc_techs <- 0

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

Haremos lo mismo para el número de herramientas 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**:

```
data_wo_na$collab_techs <- 0

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

```
data_wo_na$plat_techs <- 0

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

```
data_wo_na$web_techs <- 0

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

```

if (is.na(data_wo_na$WebframeWorkedWith[i])) {
  data_wo_na$web_techs[i] <- 0
} else {
  longitud <- sapply(strsplit(data_wo_na$WebframeWorkedWith[i], ";"), length)
  data_wo_na$web_techs[i] <- longitud
}
}

```

Eliminamos las variables cualitativas previamente tratadas

```

data_wo_na$DatabaseWorkedWith = NULL
data_wo_na$LanguageWorkedWith = NULL
data_wo_na$MiscTechWorkedWith = NULL
data_wo_na$NEWCollabToolsWorkedWith = NULL
data_wo_na$PlatformWorkedWith = NULL
data_wo_na$WebframeWorkedWith = NULL

```

La variable *EdLevel* tiene valores como cadenas de texto muy extensas que dificultan su legibilidad, 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
data_wo_na$EdLevel[data_wo_na$EdLevel=="I never completed any formal education"] <- 'NEVER'

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

# Reasigamos el valor
data_wo_na$EdLevel[data_wo_na$EdLevel=="Secondary school (e.g. American high school, German Realschule o"]

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

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

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

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

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

```

```

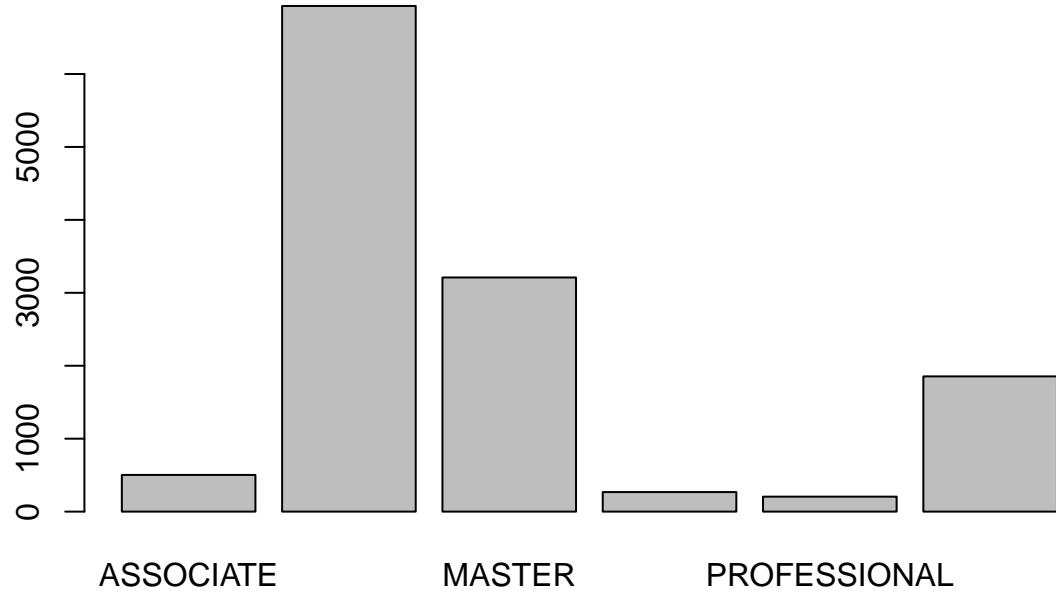
# Reasigamos el valor
data_wo_na$EdLevel[data_wo_na$EdLevel=="Other doctoral degree (Ph.D., Ed.D., etc.)"] <- 'OTHER_PHD'

table(as.factor(data_wo_na$EdLevel))

##          ASSOCIATE      BACHELOR        MASTER
##            503           6933         3210
## OTHER_PHD          268          205        1854

plot(as.factor(data_wo_na$EdLevel))

```



De la misma manera procedemos con la variable *Employment*, 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
data_wo_na$Employment[data_wo_na$Employment=="Employed full-time"] <- 'FULL_TIME'

```

```

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

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

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

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

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

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

table(as.factor(data_wo_na$Employment))

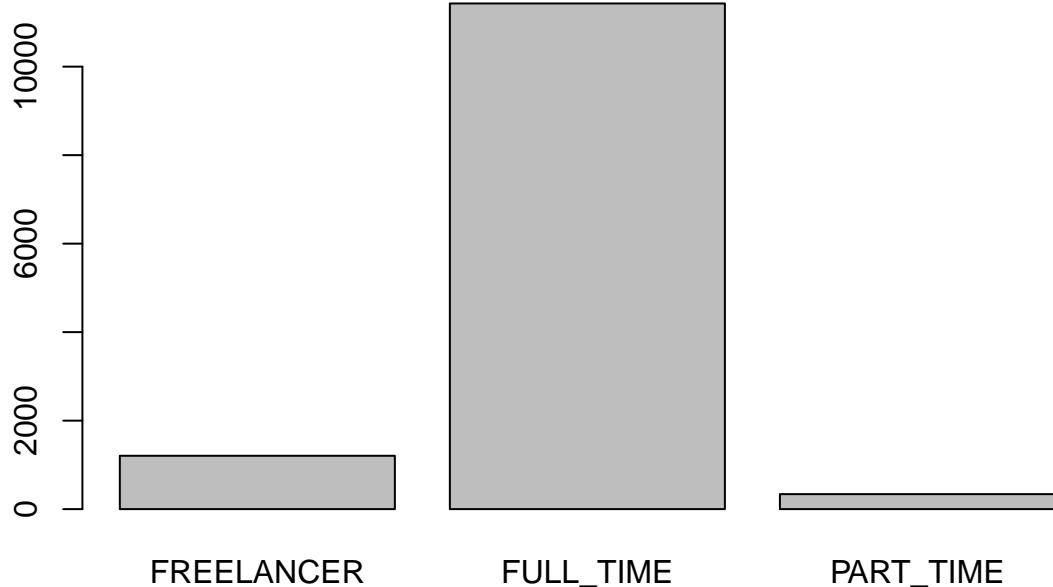
```

```

## 
## FREELANCER  FULL_TIME  PART_TIME
##      1206       11428        339

```

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



Para la variable *NEWOvertime* vamos a realizar una reasignación de valores, con valores más cortos, para lo cual procederemos de la siguiente manera:

- “Never” -> “NEVER”
- “Occasionally: 1-2 days per quarter but less than monthly” -> “OCCASIONALLY”
- “Often: 1-2 days per week or more” -> “OFTEN”
- “Rarely: 1-2 days per year or less” -> “RARELY”
- “Sometimes: 1-2 days per month but less than weekly” -> “SOMETIMES”

```
# Reasigamos el valor
data_wo_na$NEWOvertime[data_wo_na$NEWOvertime=="Never"] <- 'NEVER'

# Reasigamos el valor
data_wo_na$NEWOvertime[data_wo_na$NEWOvertime=="Occasionally: 1-2 days per quarter but less than monthly"] <- 'OCCASIONALLY'

# Reasigamos el valor
data_wo_na$NEWOvertime[data_wo_na$NEWOvertime=="Often: 1-2 days per week or more"] <- 'OFTEN'

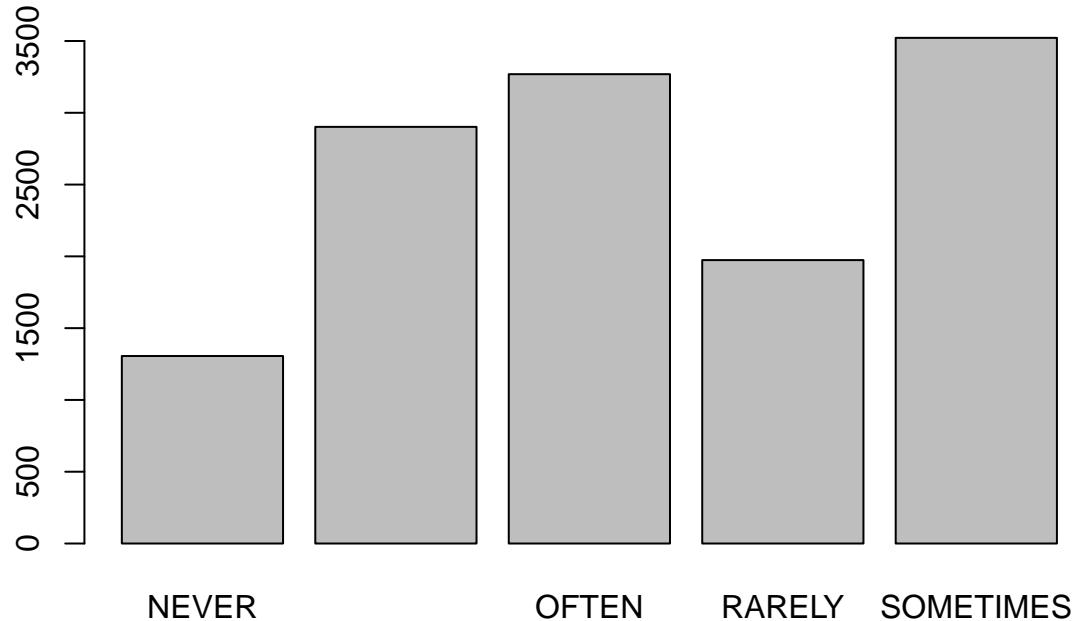
# Reasigamos el valor
data_wo_na$NEWOvertime[data_wo_na$NEWOvertime=="Rarely: 1-2 days per year or less"] <- 'RARELY'

# Reasigamos el valor
data_wo_na$NEWOvertime[data_wo_na$NEWOvertime=="Sometimes: 1-2 days per month but less than weekly"] <- 'SOMETIMES'

table(as.factor(data_wo_na$NEWOvertime))

##          NEVER OCCASIONALLY      OFTEN      RARELY      SOMETIMES
##            1306           2902       3269        1974       3522
```

plot(as.factor(data\_wo\_na\$NEWOvertime))



Para la variable *SOPartFreq* vamos a realizar una reasignación de valores, con valores más cortos, para lo cual procederemos de la siguiente manera:

- “Less than once per month or monthly” -> “LESS\_ONCE\_MONTH”
- “A few times per month or weekly” -> “FEW\_TIMES\_MONTH”
- “Multiple times per day” -> “MULTIPLE\_TIMES\_DAY”
- “I have never participated in Q&A on Stack Overflow” -> “NEVER”
- “A few times per week” -> “FEW\_TIMES\_WEEK”
- “Daily or almost daily” -> “DAILY”

```
# Reasigamos el valor
data_wo_na$SOPartFreq[data_wo_na$SOPartFreq=="Less than once per month or monthly"] <- 'LESS_ONCE_MONTH'

# Reasigamos el valor
data_wo_na$SOPartFreq[data_wo_na$SOPartFreq=="A few times per month or weekly"] <- 'FEW_TIMES_MONTH'

# Reasigamos el valor
data_wo_na$SOPartFreq[data_wo_na$SOPartFreq=="Multiple times per day"] <- 'MULTIPLE_TIMES_DAY'

# Reasigamos el valor
data_wo_na$SOPartFreq[data_wo_na$SOPartFreq=="I have never participated in Q&A on Stack Overflow"] <- 'NEVER'

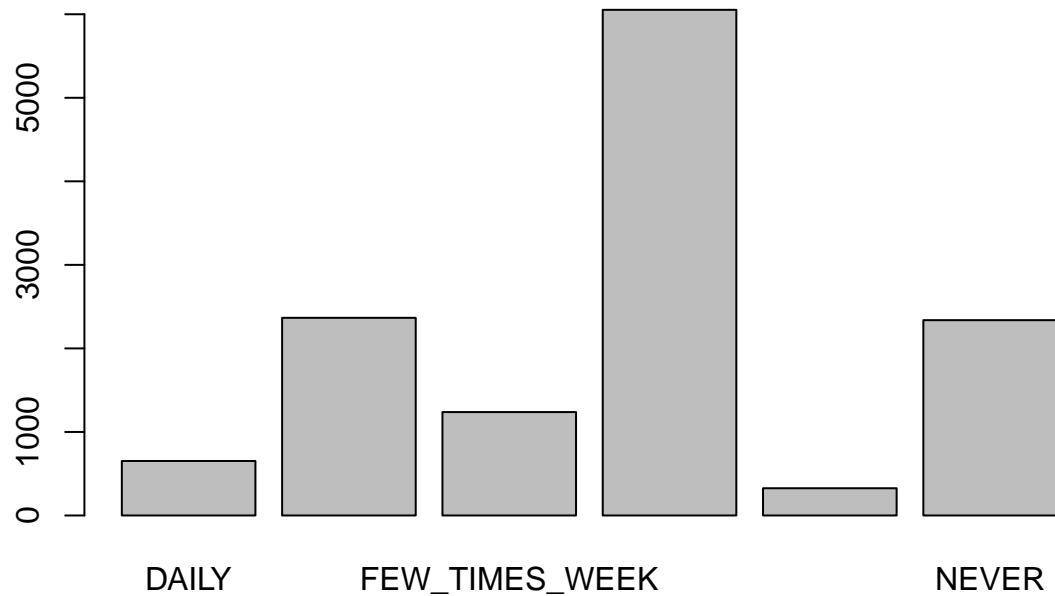
# Reasigamos el valor
data_wo_na$SOPartFreq[data_wo_na$SOPartFreq=="A few times per week"] <- 'FEW_TIMES_WEEK'

# Reasigamos el valor
data_wo_na$SOPartFreq[data_wo_na$SOPartFreq=="Daily or almost daily"] <- 'DAILY'

table(as.factor(data_wo_na$SOPartFreq))

##          DAILY   FEW_TIMES_MONTH   FEW_TIMES_WEEK LESS_ONCE_MONTH
##             652           2366          1238            6053
## MULTIPLE_TIMES_DAY               NEVER
##             326           2338
```

```
plot(as.factor(data_wo_na$SOPartFreq))
```



Para la variable *UndergradMajor* vamos a realizar una reasignación de valores, con valores más cortos, 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\_DISCIPLINE”
- “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
data_wo_na$UndergradMajor[data_wo_na$UndergradMajor=="Computer science, computer engineering, or software engineering"] <- 'COMPUTER_SCIENCE'

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

# Reasigamos el valor
data_wo_na$UndergradMajor[data_wo_na$UndergradMajor=="Information systems, information technology, or system administration"] <- 'INFORMATION_SYSTEMS'

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

# Reasigamos el valor
data_wo_na$UndergradMajor[data_wo_na$UndergradMajor=="Another engineering discipline (such as civil, electrical, mechanical, etc.)"] <- 'ANOTHER_ENGINEERING_DISCIPLINE'
```

```

# Reasigamos el valor
data_wo_na$UndergradMajor[data_wo_na$UndergradMajor=="A business discipline (such as accounting, finance, marketing, etc.)"] <- 'BUSINESS'

# Reasigamos el valor
data_wo_na$UndergradMajor[data_wo_na$UndergradMajor=="A health science (such as nursing, pharmacy, radiology, etc.)"] <- 'FINE_ARTS'

# Reasigamos el valor
data_wo_na$UndergradMajor[data_wo_na$UndergradMajor=="A humanities discipline (such as literature, history, philosophy, etc.)"] <- 'HUMANITIES'

# Reasigamos el valor
data_wo_na$UndergradMajor[data_wo_na$UndergradMajor=="A natural science (such as biology, chemistry, physics, etc.)"] <- 'MATHS_STATS'

# Reasigamos el valor
data_wo_na$UndergradMajor[data_wo_na$UndergradMajor=="A social science (such as anthropology, psychology, sociology, etc.)"] <- 'SOCIAL_SCIENCE'

# Reasigamos el valor
data_wo_na$UndergradMajor[data_wo_na$UndergradMajor=="Fine arts or performing arts (such as graphic design, music, theater, etc.)"] <- 'ARTS'

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

table(as.factor(data_wo_na$UndergradMajor))

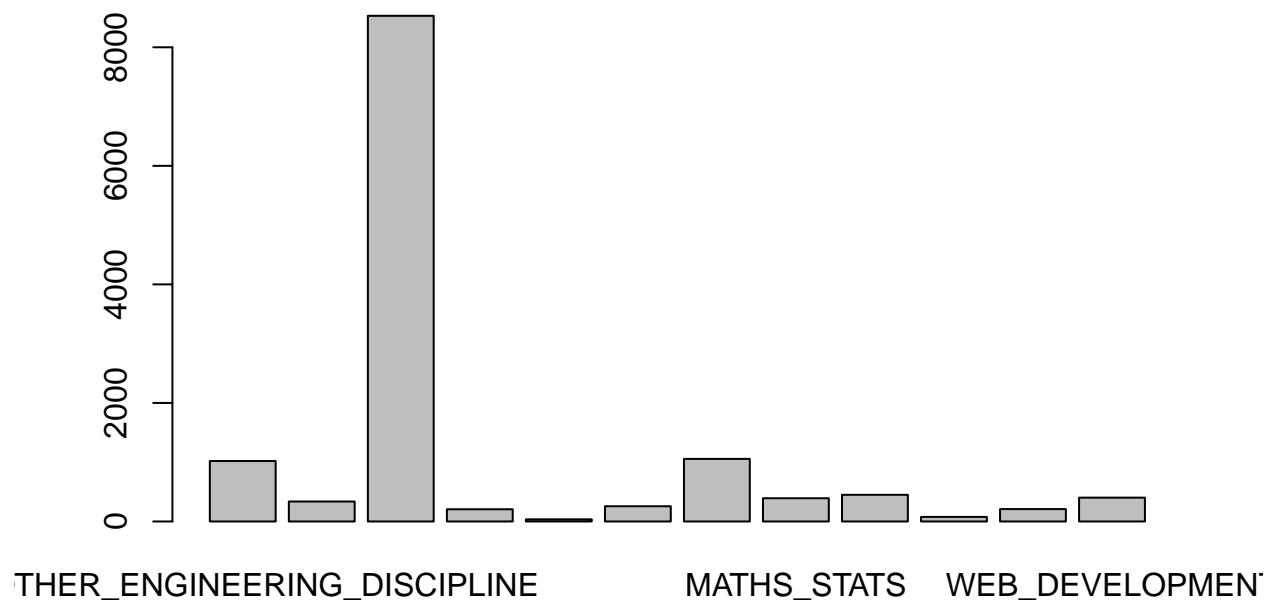
```

```

##
## ANOTHER_ENGINEERING_DISCIPLINE          BUSINESS
##                               1020           337
## COMPUTER_SCIENCE                      FINE_ARTS
##                               8531           206
## HEALTH                                HUMANITIES
##                               36            257
## INFORMATION_SYSTEMS                    MATHS_STATS
##                               1056           392
## NATURAL_SCIENCE                       NEVER_MAJOR
##                               450            77
## SOCIAL_SCIENCE                        WEB_DEVELOPMENT
##                               209           402

```

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

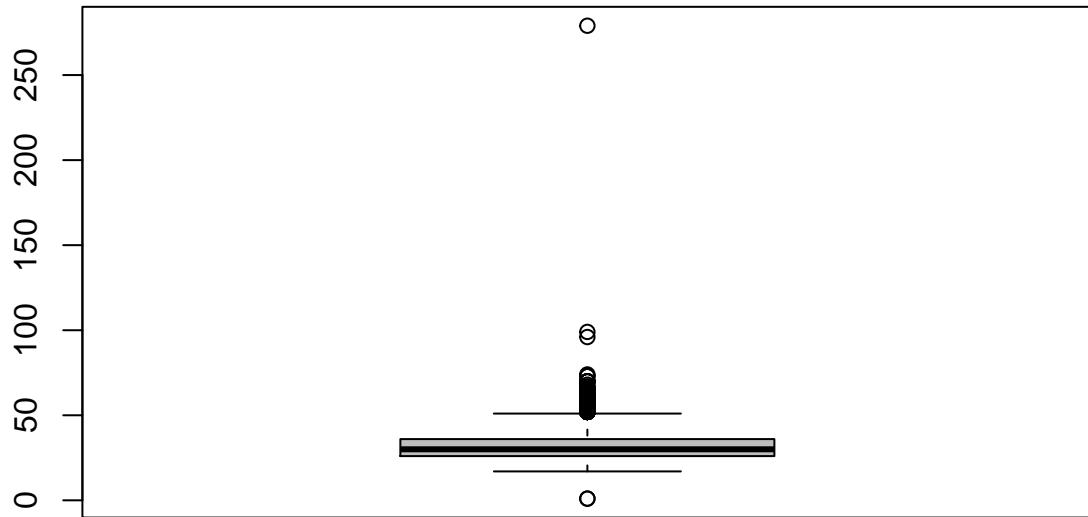


### 3.2 Identificación y tratamiento de valores extremos.

- Para la variable `Age`

```
boxplot(as.numeric(data_wo_na$Age), main="Valores para Age", col="gray")
```

## Valores para Age



Para la variable Age los valores extremos los obtenemos de la siguiente manera:

```
atipicos_age <- boxplot.stats(as.numeric(data_wo_na$Age))

head(sort(atipicos_age$out, decreasing = TRUE), n=10)

## [1] 279 99 96 74 73 73 70 70 70 70

tail(sort(atipicos_age$out, decreasing = TRUE), n=10)

## [1] 52 52 52 52 52 52 52 52 1 1
```

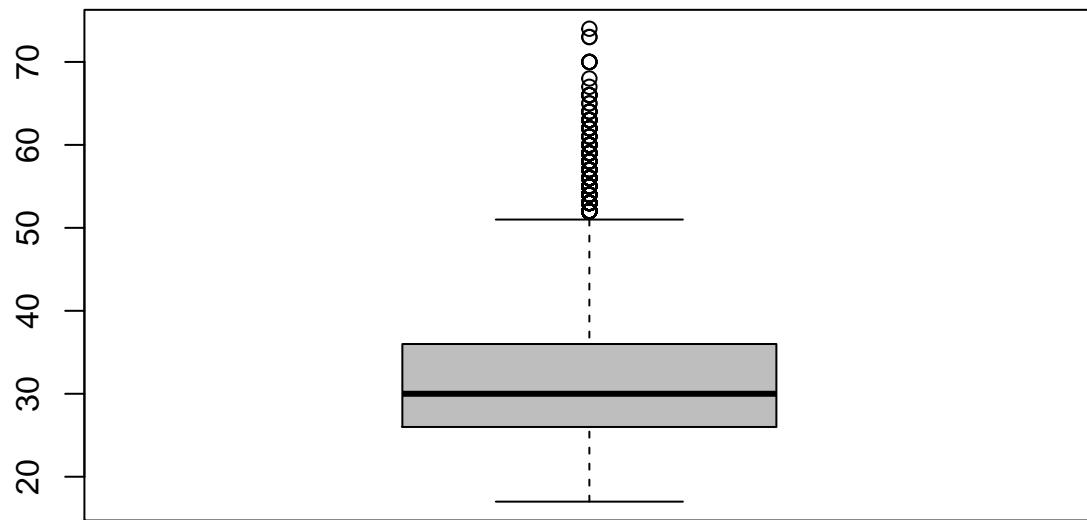
Al ordenar los valores y analizarlos encontramos que los valores máximos que podrían considerarse outliers son los mayores o iguales a 96 y los menores o iguales a 1.

Por lo tanto, procedemos a borrar esas observaciones que los contienen

```
data_wo_na <- data_wo_na[data_wo_na$Age < 96, ]
data_wo_na <- data_wo_na[data_wo_na$Age > 1, ]

# volvemos a revisar los valores outliers
boxplot(as.numeric(data_wo_na$Age), main="Valores para Age", col="gray")
```

## Valores para Age

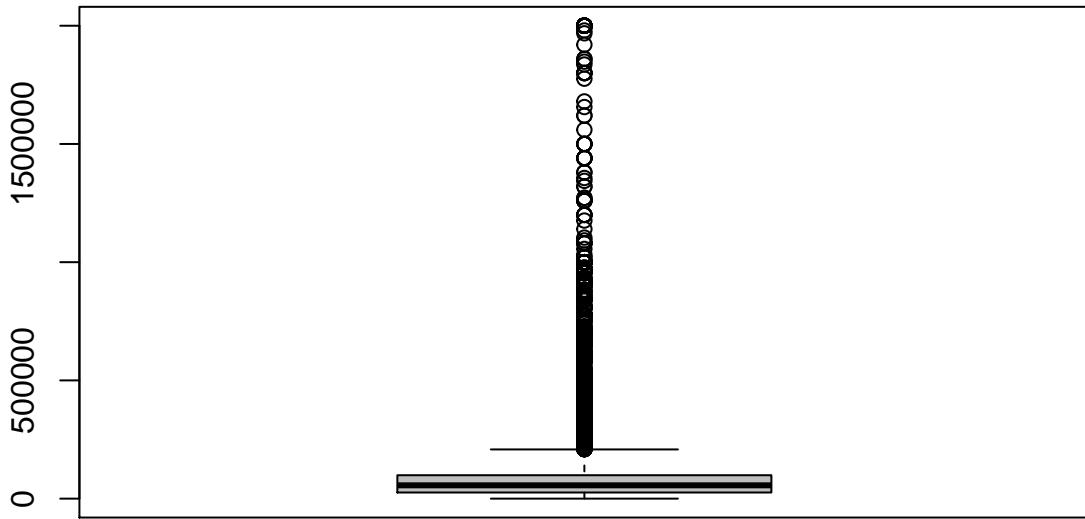


A pesar de encontrarse unos valores superiores, por fuera del tercer cuartil (75%) aún se consideran válidos.

- Para la variable **ConvertedComp**

```
boxplot(as.numeric(data_wo_na$ConvertedComp), main="Valores para Sueldo Anual (ConvertedComp)", col="gray")
```

## Valores para Sueldo Anual (ConvertedComp)



Para la variable Age los valores extremos los obtenemos de la siguiente manera:

```
atipicos_sueldo_anual <- boxplot.stats(as.numeric(data_wo_na$ConvertedComp))
```

```
head(sort(atipicos_sueldo_anual$out, decreasing = TRUE), n=10)
```

```
## [1] 2e+06 2e+06 2e+06 2e+06 2e+06 2e+06 2e+06 2e+06 2e+06 2e+06
```

```
tail(sort(atipicos_sueldo_anual$out, decreasing = TRUE), n=10)
```

```
## [1] 210000 210000 210000 210000 210000 210000 210000 210000 210000 210000
```

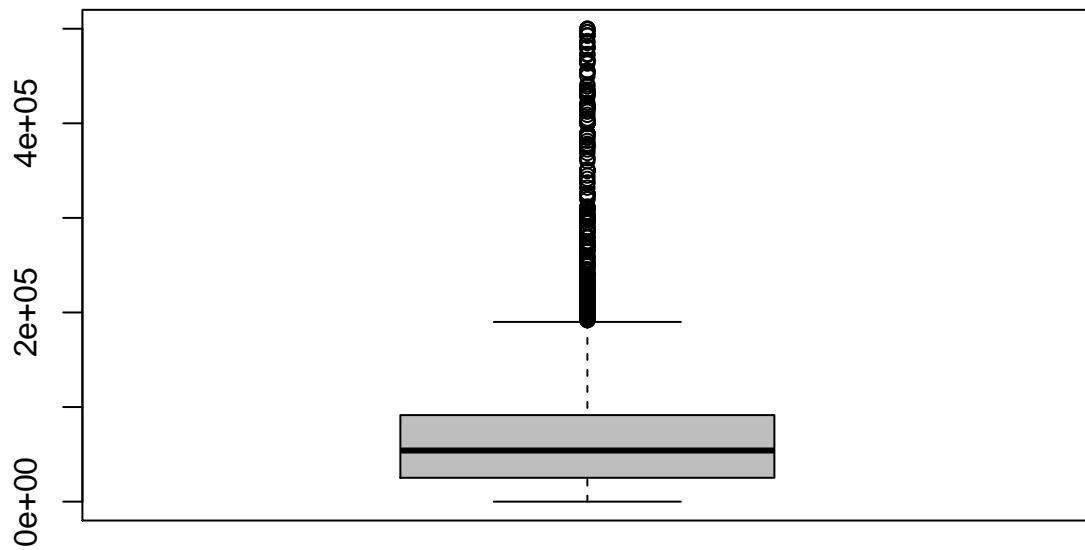
Al ordenar los valores y analizarlos encontramos que los valores máximos que podrían considerarse outliers son los mayores o iguales a 208000. Sin embargo, para efectos del análisis vamos a considerar como valores a los no sobrepasan el límite de 500K. Por lo tanto, procedemos a borrar esas observaciones que contienen los valores superiores a el límite.

```
data_wo_na <- data_wo_na[data_wo_na$ConvertedComp <= 500000, ]
```

# volvemos a revisar los valores outliers

```
boxplot(as.numeric(data_wo_na$ConvertedComp), main="Valores para Sueldo Anual (ConvertedComp)", col="gray")
```

## Valores para Sueldo Anual (ConvertedComp)

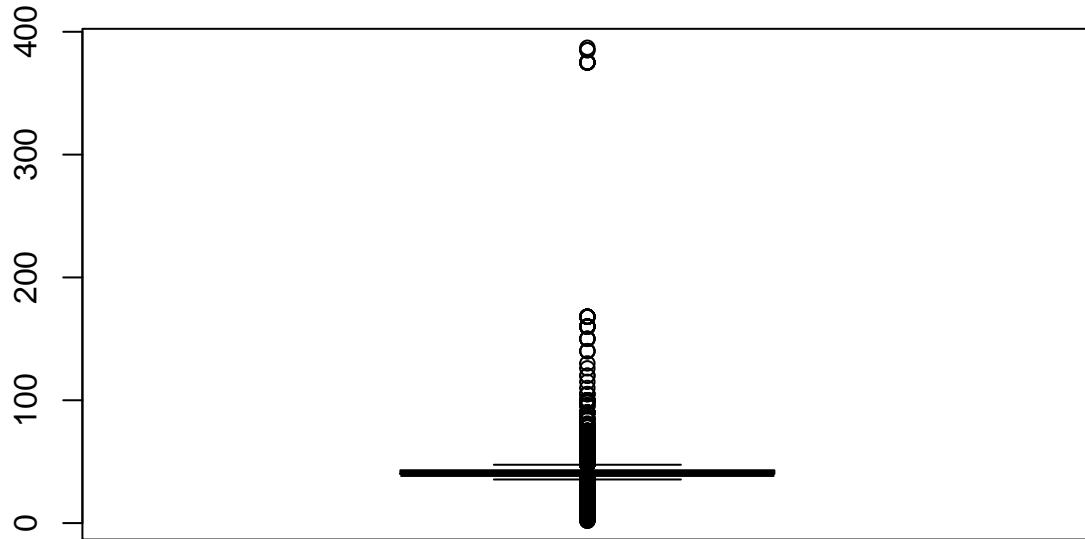


A pesar de encontrarse unos valores superiores, por fuera del tercer cuartil (75%) aún se consideran válidos.

- Para la variable **WorkWeekHrs**

```
boxplot(as.numeric(data_wo_na$WorkWeekHrs), main="Valores para Horas trabajadas en la semana (WorkWeekHrs)
```

## Valores para Horas trabajadas en la semana (WorkWeekHrs)



Para la variable Age los valores extremos los obtenemos de la siguiente manera:

```
atipicos_horas_semanales <- boxplot.stats(as.numeric(data_wo_na$WorkWeekHrs))

head(sort(atipicos_horas_semanales$out, decreasing = FALSE), n=10)

## [1] 2 2 2 2 3 3 3 3 4 4

tail(sort(atipicos_horas_semanales$out, decreasing = FALSE), n=10)

## [1] 375 375 375 375 375 375 375 385 385 387
```

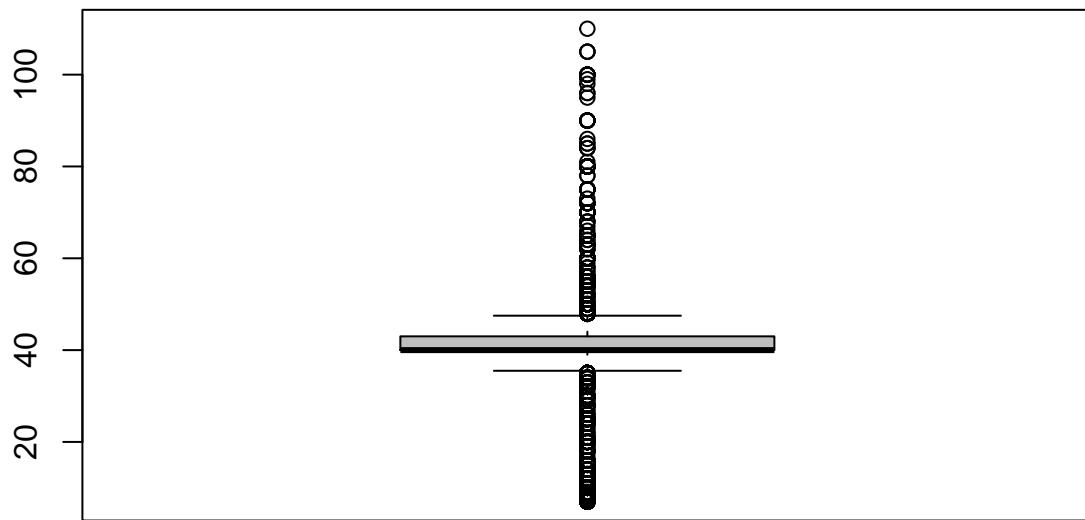
Al ordenar los valores y analizarlos encontramos que los valores máximos que podrían considerarse outliers son los mayores a 112 horas, ya que trabajar más de 16 horas diarias se encuentra poco probable. De la misma manera, se considera poco probable trabajar menos de una hora diaria (7 horas semanales, incluso en un trabajo a tiempo parcial). Por lo tanto, procedemos a borrar esas observaciones que contienen los valores superiores e inferiores a los límites mencionados.

Cabe recalcar que el valor 112 se obtiene en el supuesto caso que se trabaje 16 horas diarias los 7 días de la semana, en el caso extremo.

```
data_wo_na <- data_wo_na[data_wo_na$WorkWeekHrs <= 112, ]
data_wo_na <- data_wo_na[data_wo_na$WorkWeekHrs >= 7, ]

# volvemos a revisar los valores outliers
boxplot(as.numeric(data_wo_na$WorkWeekHrs), main="Valores para Horas trabajadas en la semana (WorkWeekHrs")
```

## Valores para Horas trabajadas en la semana (WorkWeekHrs)

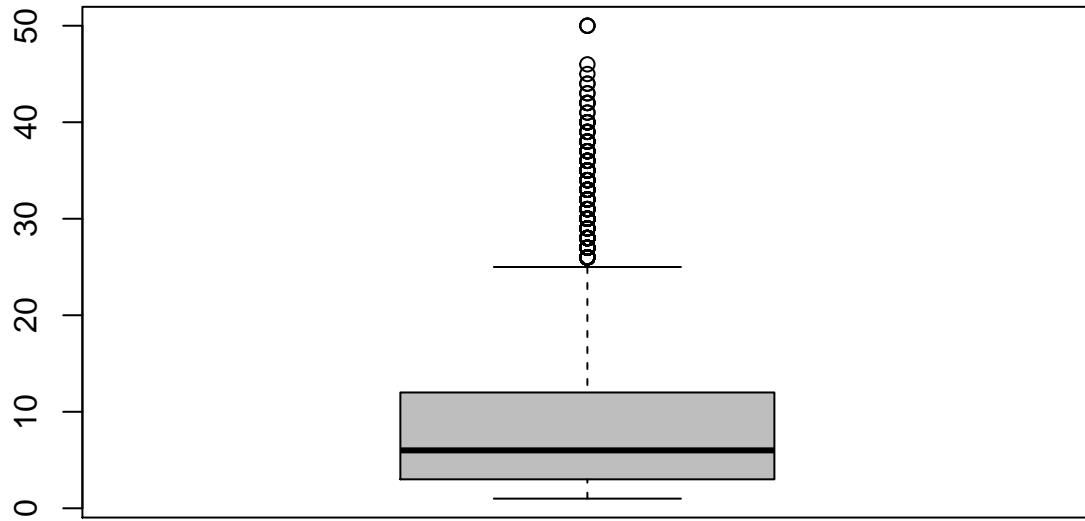


A pesar de encontrarse unos valores superiores e inferiores, por fuera del tercer(75%) y primer (25%) cuartil, respectivamente; aún se consideran válidos para efectos del análisis que realizaremos.

- Para la variable **YearsCodePro**

```
boxplot(as.numeric(data_wo_na$YearsCodePro), main="Valores para Años Profesionales (YearsCodePro)", col=
```

## Valores para Años Profesionales (YearsCodePro)



Para la variable Age los valores extremos los obtenemos de la siguiente manera:

```
atipicos_years_pro <- boxplot.stats(as.numeric(data_wo_na$YearsCodePro))

head(sort(atipicos_years_pro$out, decreasing = TRUE), n=10)

## [1] 50 50 50 46 45 44 44 43 43 42

tail(sort(atipicos_years_pro$out, decreasing = TRUE), n=10)

## [1] 26 26 26 26 26 26 26 26 26 26
```

Al ordenar los valores outliers de manera descendente, encontramos los valores en el rango entre 26 y 50 años. Sin embargo, yendo a la lógica actual en los años que podría tener un desarrollador no encontramos como valor atípico una experiencia comprendida entre los mencionados límites. Para verificar de manera más detallada esta variable, vamos a comparar la variable Age vs. la variable YearsCodePro para descartar alguna anomalía, como que los años de profesional sean mayor a la edad actual o no haya lógica entre la una con la otra.

Para esto vamos a proceder a generar un data frame entre las 2 variables considerando los valores comprendidos en el rango antes mencionado.

```
data_age_years_pro <- data_wo_na[data_wo_na$YearsCodePro >= 26, c(1, 12)]

# Obtenemos una variable que almacene la diferencia en la edad como profesional y la edad actual
filas_ndf <- dim(data_age_years_pro)[1]

data_age_years_pro$diferencia <- 0
```

```

for(i in 1:filas_ndf) {
  data_age_years_pro$diferencia[i] <- data_age_years_pro$Age[i] - data_age_years_pro$YearsCodePro[i]
}

head(sort(data_age_years_pro$diferencia, decreasing = TRUE), n=10)

## [1] 42 40 37 32 32 32 32 32 31 31

tail(sort(data_age_years_pro$diferencia, decreasing = TRUE), n=10)

## [1] 14 14 14 14 12 12 12 11 10 10 8

```

Todos los valores en las diferencias obtenidas son lógicos. Sin embargo, a pesar de encontrarse una diferencia mínima de 8 años entre la experiencia como profesional y la edad actual, sabemos de casos extremos de niños que pueden aprender a programar o cualquier otra habilidad a corta edad, por ende no eliminaremos ningún registro.

## 4 Análisis de los datos

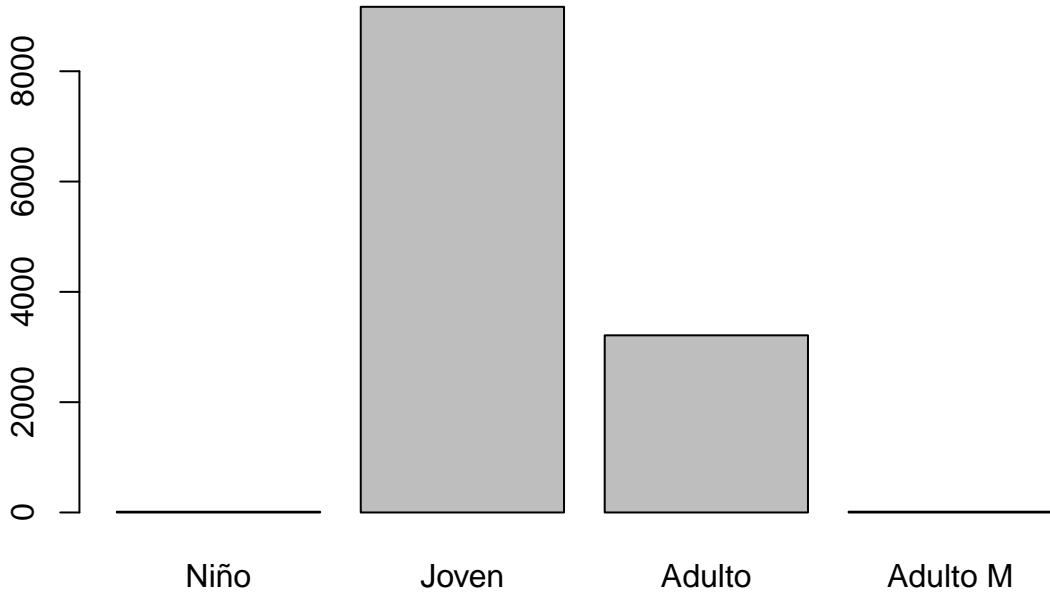
### 4.1 Selección de los grupos de datos a analizar

Se seleccionan las variables que se usarán de nuestro dataset las cuales serán las que darán respuestas a nuestros cuestionamientos iniciales. Alguna de estas puede no ser relevantes en los siguientes análisis ya que pueden no tener afectación en las respuestas que se buscan.

```

#Discretizamos por Age
data_wo_na$nivel_edad <- cut(data_wo_na$Age, breaks = c(0,18,35,65,100), labels = c("Niño", "Joven", "Adulto"))
plot(as.factor(data_wo_na$nivel_edad))

```



```
# Tamaño de nuestro dataset y nombre de las columnas
dim(data_wo_na)

## [1] 12407    19

names(data_wo_na)

##  [1] "Age"           "ConvertedComp"   "Country"        "EdLevel"
##  [5] "Employment"    "NEWLearn"       "NEWOvertime"    "OpSys"
##  [9] "SOPartFreq"    "UndergradMajor" "WorkWeekHrs"   "YearsCodePro"
## [13] "db_techs"      "prog_langs"     "misc_techs"    "collab_techs"
## [17] "plat_techs"    "web_techs"     "nivel_edad"

#Resumen de los valores
summary(data_wo_na)

##          Age      ConvertedComp      Country      EdLevel
##  Min.   :17.00  Min.   : 1   Length:12407  Length:12407
##  1st Qu.:26.00  1st Qu.: 25610  Class :character  Class :character
##  Median :30.00  Median : 54072  Mode  :character  Mode  :character
##  Mean   :31.76  Mean   : 67633
##  3rd Qu.:36.00  3rd Qu.: 91883
##  Max.   :74.00  Max.   :500000
## 
##          Employment      NEWLearn      NEWOvertime      OpSys
##  Length:12407  Length:12407  Length:12407  Length:12407
##  Class :character  Class :character  Class :character  Class :character
##  Mode  :character  Mode  :character  Mode  :character  Mode  :character
## 
```

```

## 
## 
##   S0PartFreq      UndergradMajor      WorkWeekHrs      YearsCodePro
##   Length:12407    Length:12407       Min.   : 7.00      Min.   : 1.000
##   Class :character  Class :character  1st Qu.: 40.00     1st Qu.: 3.000
##   Mode  :character  Mode  :character  Median  : 40.00     Median  : 6.000
##                               Mean   : 40.37     Mean   : 8.602
##                               3rd Qu.: 43.00     3rd Qu.:12.000
##                               Max.  :110.00    Max.  :50.000
## 
##   db_techs      prog_langs      misc_techs      collab_techs
##   Min.   : 1.000  Min.   : 1.00  Min.   : 1.000  Min.   : 1.000
##   1st Qu.: 2.000  1st Qu.: 4.00  1st Qu.: 1.000  1st Qu.: 3.000
##   Median  : 3.000  Median  : 5.00  Median  : 2.000  Median  : 4.000
##   Mean    : 3.168  Mean    : 5.71  Mean    : 2.325  Mean    : 4.389
##   3rd Qu.: 4.000  3rd Qu.: 7.00  3rd Qu.: 3.000  3rd Qu.: 6.000
##   Max.   :13.000  Max.   :22.00  Max.   :15.000  Max.   :11.000
## 
##   plat_techs      web_techs      nivel_edad
##   Min.   : 1.00  Min.   : 1.000 Niño   : 13
##   1st Qu.: 2.00  1st Qu.: 2.000 Joven  :9168
##   Median  : 4.00  Median  : 3.000 Adulto :3212
##   Mean    : 3.96  Mean    : 3.076 Adulto M: 14
##   3rd Qu.: 5.00  3rd Qu.: 4.000
##   Max.   :16.00  Max.   :15.000

```

## 4.2 Comprobación de la normalidad y homogeneidad de la varianza

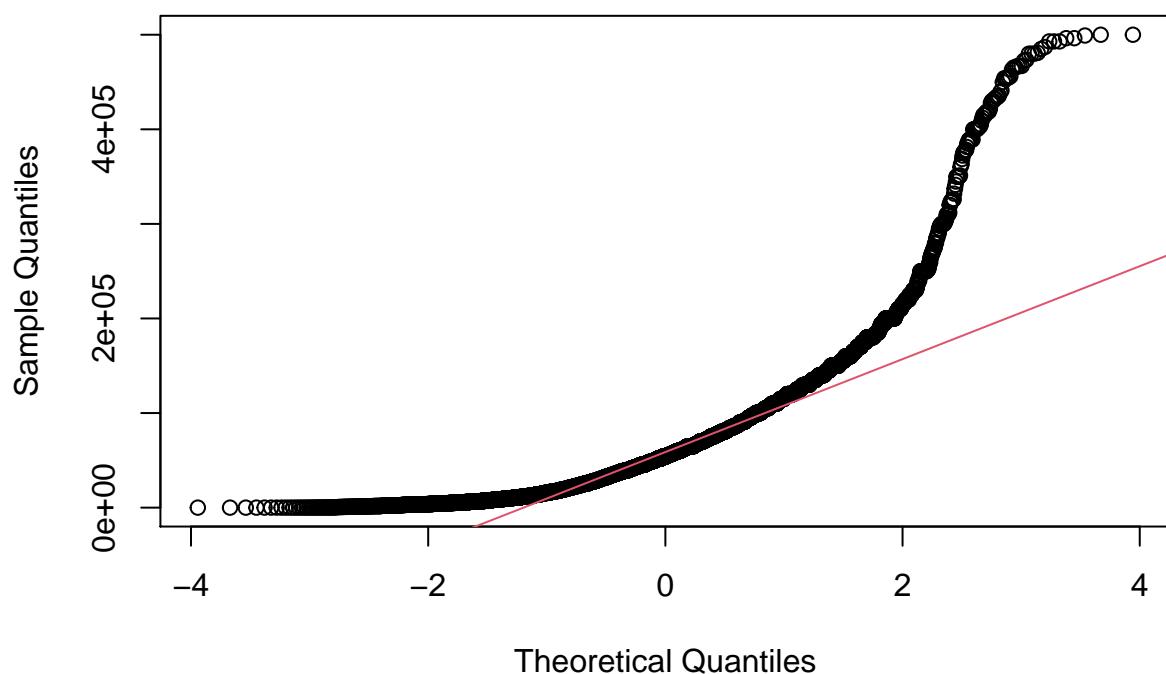
Para este apartado trabajaremos con un análisis visual mediante graficas que permiten observar la similitud entre las distribuciones de los datos y una distribución normal ideal, además realizaremos tests de Shapiro-Wilk para revalidar la normalidad de las variables, asumiremos como hipótesis nula que la población está distribuida normalmente, si el *p-valor* es menor al *nivel de significancia = 0.05*.

```

#Grafica ConvertedComp
qqnorm(data_wo_na$ConvertedComp)
qqline(data_wo_na$ConvertedComp,col=2)

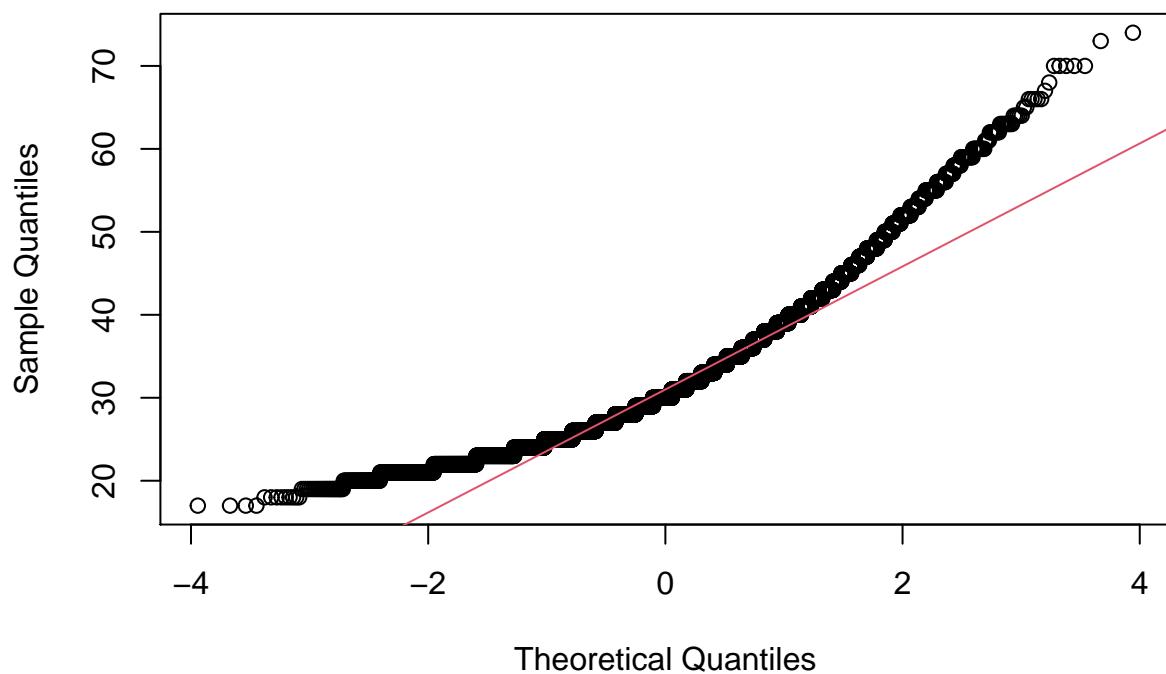
```

## Normal Q-Q Plot



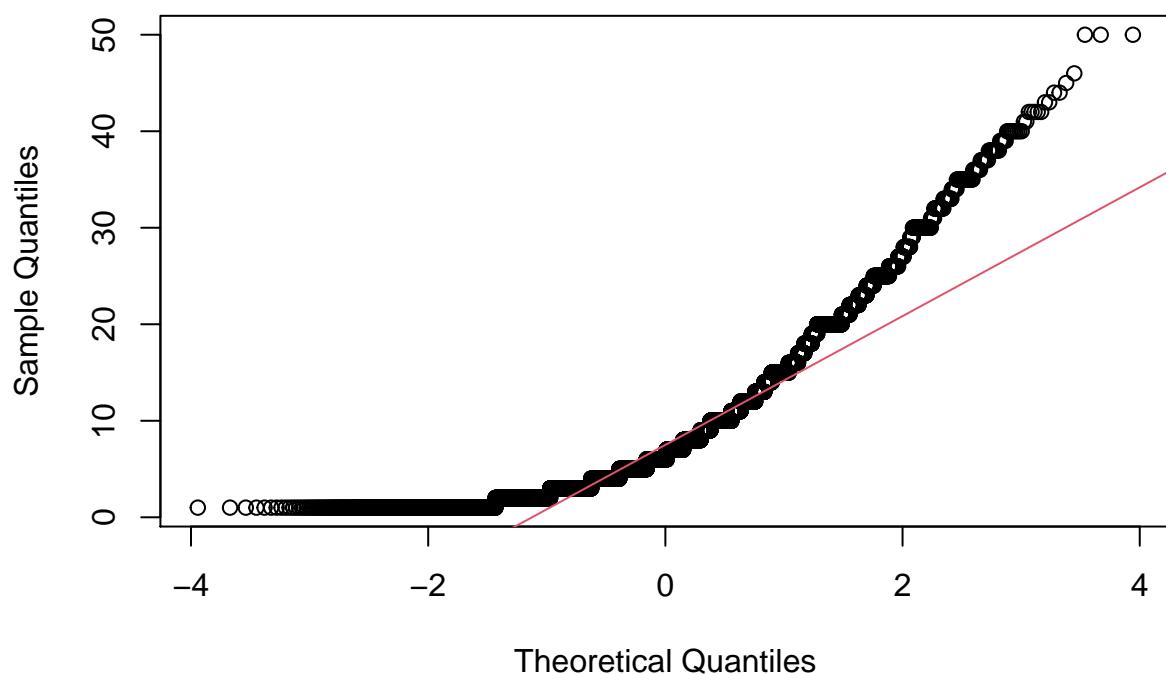
```
#Grafica Age
qqnorm(data_wo_na$Age)
qqline(data_wo_na$Age,col=2)
```

## Normal Q-Q Plot



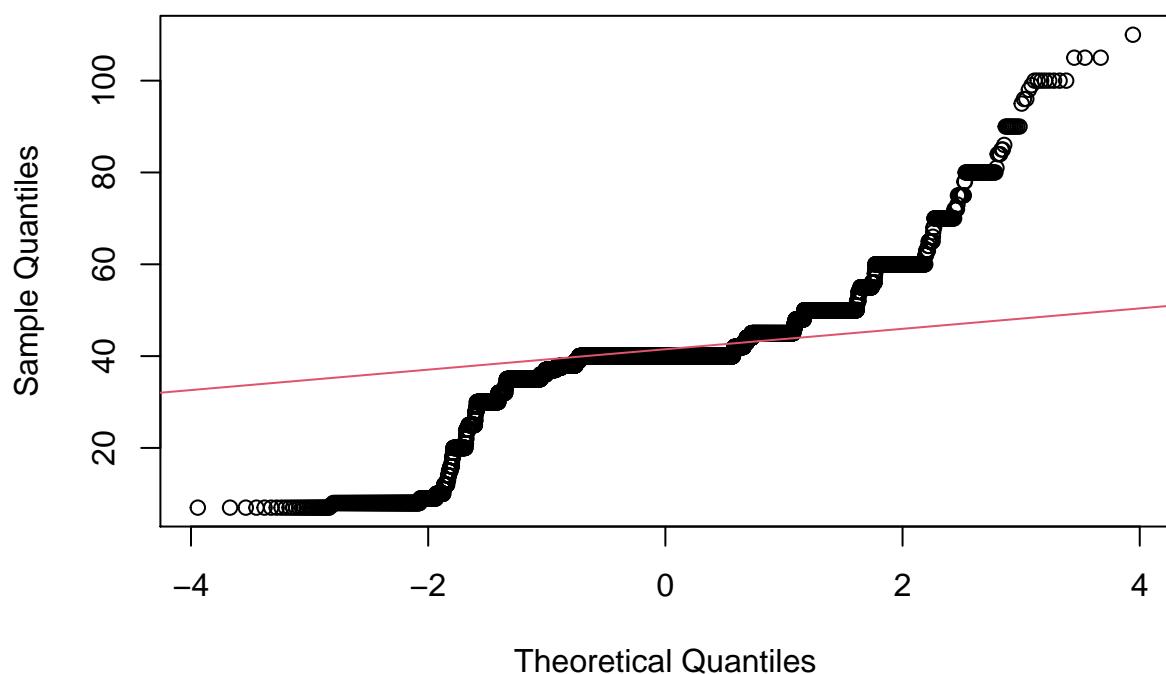
```
#Grafica YearsCodePro
qqnorm(data_wo_na$YearsCodePro)
qqline(data_wo_na$YearsCodePro,col=2)
```

## Normal Q-Q Plot



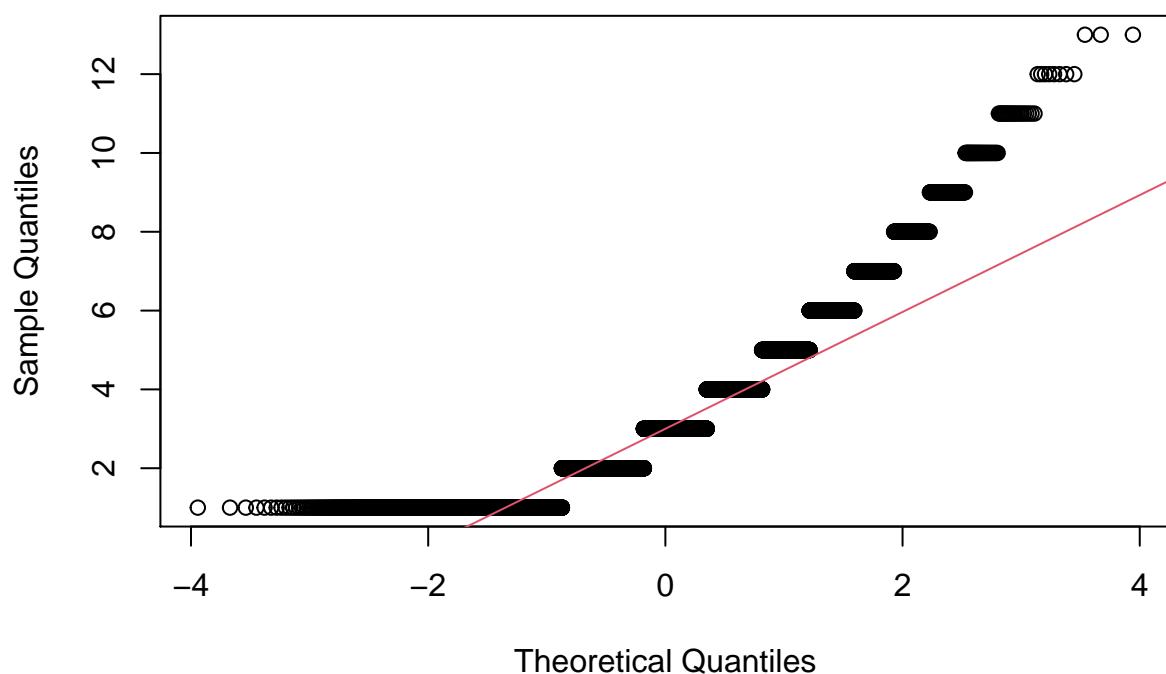
```
#Grafica WorkWeekHrs  
qqnorm(data_wo_na$WorkWeekHrs)  
qqline(data_wo_na$WorkWeekHrs,col=2)
```

## Normal Q-Q Plot



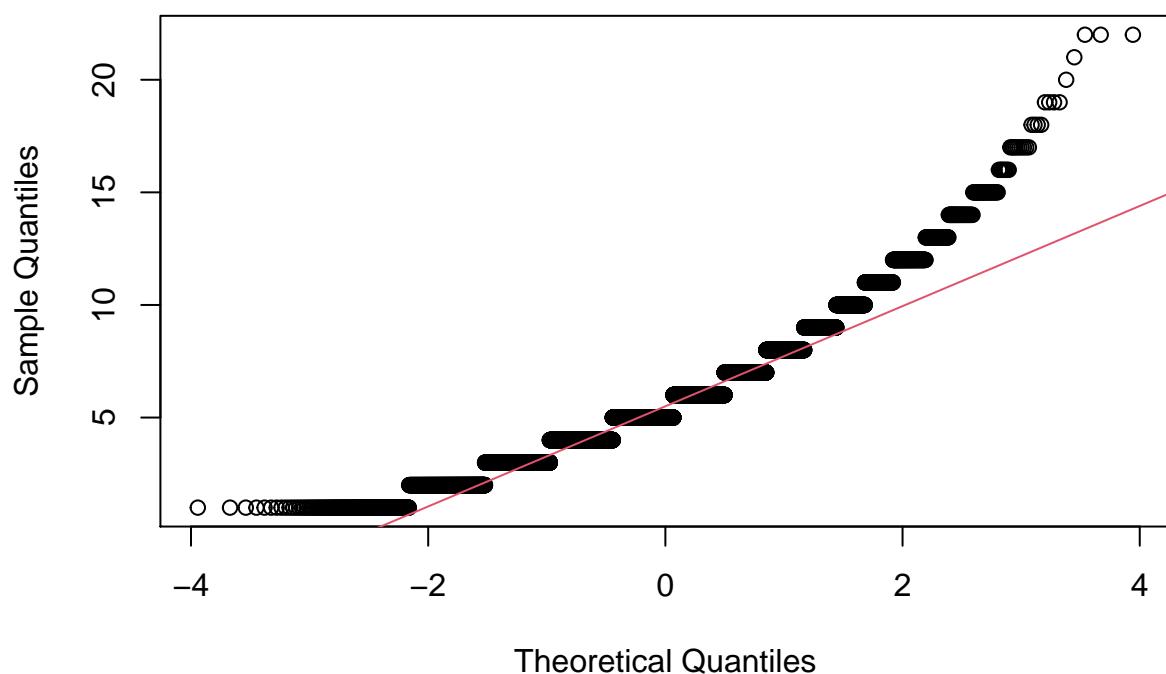
```
#Grafica db_techs  
qqnorm(data_wo_na$db_techs)  
qqline(data_wo_na$db_techs,col=2)
```

## Normal Q-Q Plot



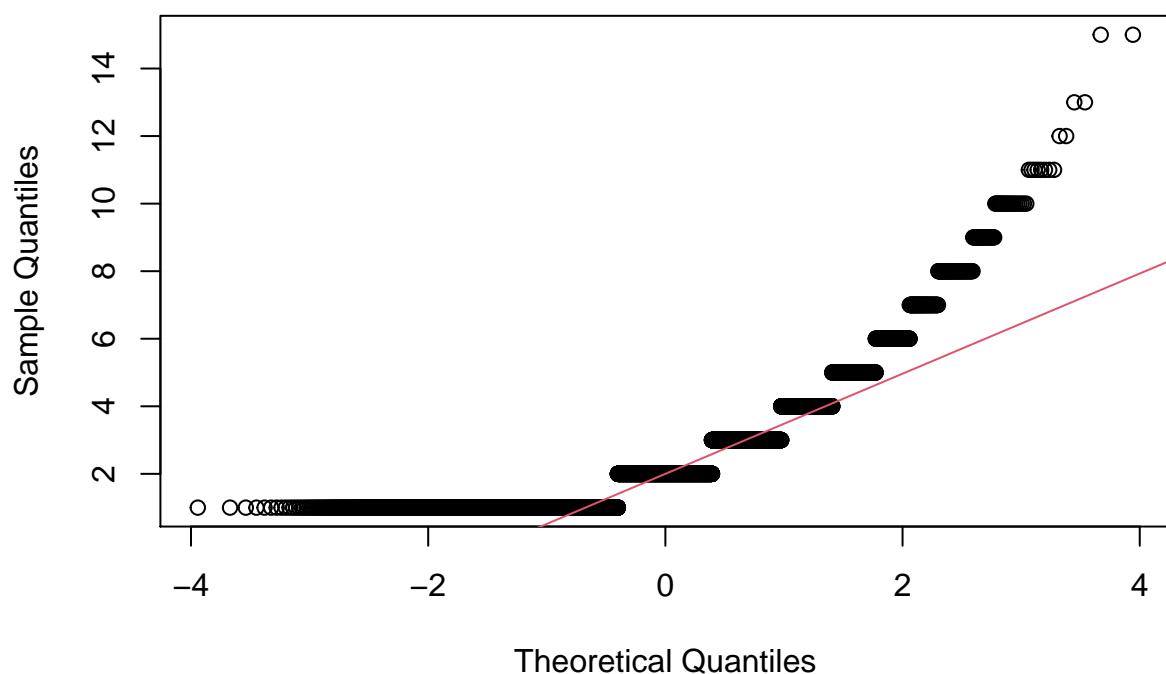
```
#Grafica prog_langs  
qqnorm(data_wo_na$prog_langs)  
qqline(data_wo_na$prog_langs,col=2)
```

## Normal Q-Q Plot



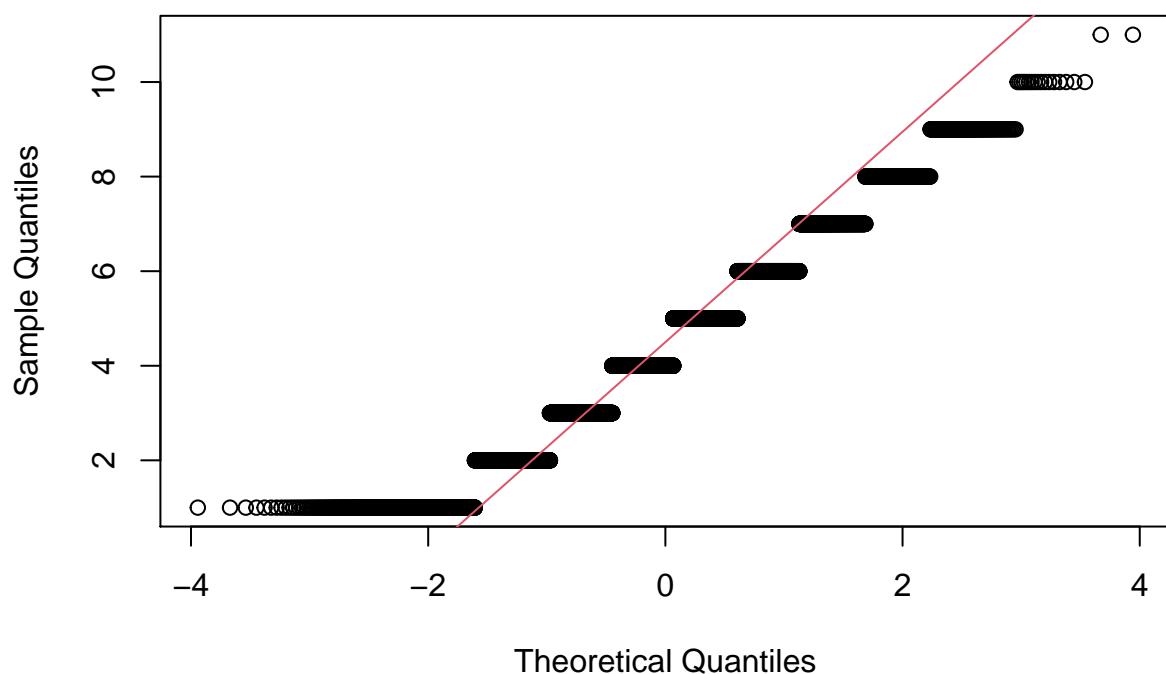
```
#Grafica misc_techs
qqnorm(data_wo_na$misc_techs)
qqline(data_wo_na$misc_techs,col=2)
```

## Normal Q-Q Plot



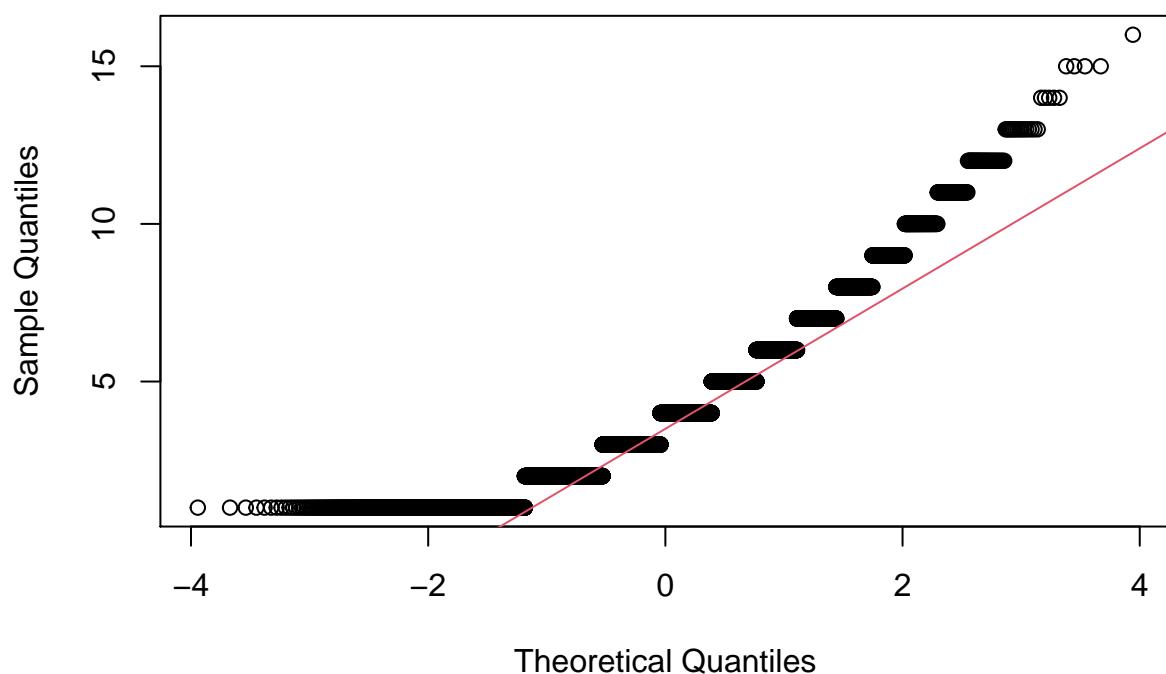
```
#Grafica collab_techs
qqnorm(data_wo_na$collab_techs)
qqline(data_wo_na$collab_techs,col=2)
```

## Normal Q-Q Plot



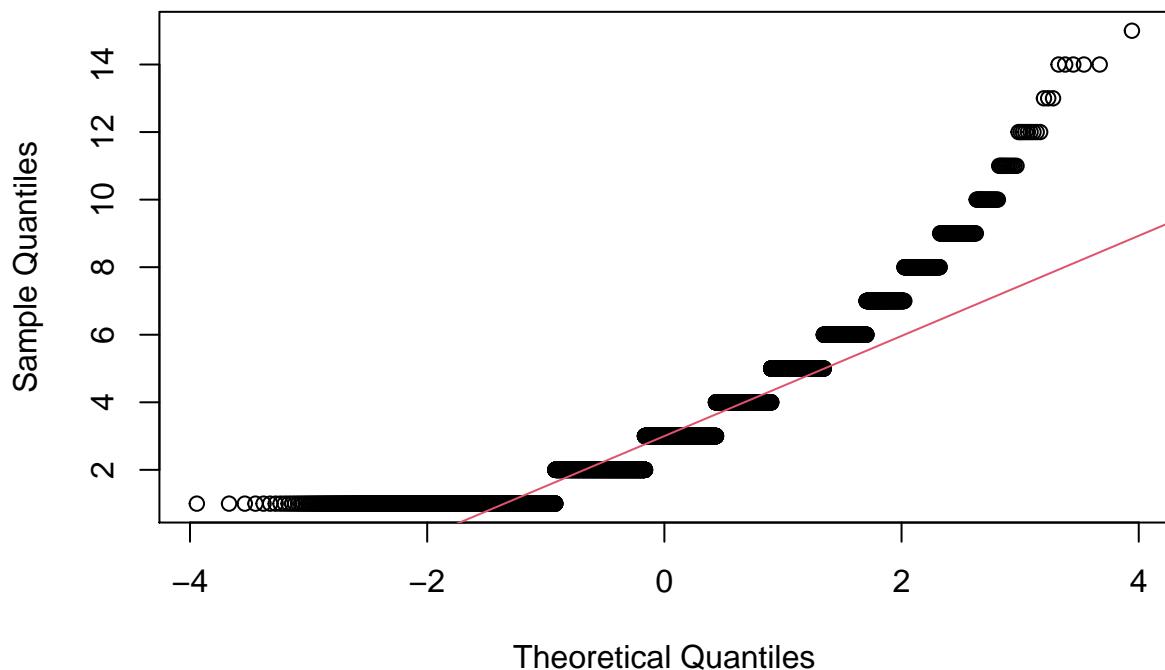
```
#Grafica plat_techs
qqnorm(data_wo_na$plat_techs)
qqline(data_wo_na$plat_techs,col=2)
```

## Normal Q-Q Plot



```
#Grafica web_techs  
qqnorm(data_wo_na$web_techs)  
qqline(data_wo_na$web_techs,col=2)
```

## Normal Q-Q Plot



Podemos observar en los histogramas que ninguno de los atributos se asemeja a una normal, el más cercano es el campo **collab\_techs**.

Ahora comprobaremos mediante el test de shapiro si nuestra observación es correcta.

```
#Realizamos una muestra de 5.000 registros, esto para aplicar el test de shapiro
n<-5000
muestra<- sample(1:nrow(data_wo_na),size=n,replace=FALSE)
data_wo_na_muestra<- data_wo_na[muestra, ]

#Test ConvertedComp
shapiro.test(data_wo_na_muestra$ConvertedComp)

##
##  Shapiro-Wilk normality test
##
## data: data_wo_na_muestra$ConvertedComp
## W = 0.80612, p-value < 2.2e-16

#Test Age
shapiro.test(data_wo_na_muestra$Age)

##
##  Shapiro-Wilk normality test
##
## data: data_wo_na_muestra$Age
## W = 0.92968, p-value < 2.2e-16
```

```
#Test YearsCodePro  
shapiro.test(data_wo_na_muestra$YearsCodePro)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: data_wo_na_muestra$YearsCodePro  
## W = 0.86493, p-value < 2.2e-16
```

```
#Test WorkWeekHrs  
shapiro.test(data_wo_na_muestra$WorkWeekHrs)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: data_wo_na_muestra$WorkWeekHrs  
## W = 0.77161, p-value < 2.2e-16
```

```
#Test db_techs  
shapiro.test(data_wo_na_muestra$db_techs)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: data_wo_na_muestra$db_techs  
## W = 0.90108, p-value < 2.2e-16
```

```
#Test prog_langs  
shapiro.test(data_wo_na_muestra$prog_langs)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: data_wo_na_muestra$prog_langs  
## W = 0.94113, p-value < 2.2e-16
```

```
#Test misc_techs  
shapiro.test(data_wo_na_muestra$misc_techs)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: data_wo_na_muestra$misc_techs  
## W = 0.80565, p-value < 2.2e-16
```

```
#Test collab_techs  
shapiro.test(data_wo_na_muestra$collab_techs)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: data_wo_na_muestra$collab_techs  
## W = 0.96644, p-value < 2.2e-16
```

```
#Test plat_techs  
shapiro.test(data_wo_na_muestra$plat_techs)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: data_wo_na_muestra$plat_techs  
## W = 0.9216, p-value < 2.2e-16
```

```
#Test web_techs  
shapiro.test(data_wo_na_muestra$web_techs)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: data_wo_na_muestra$web_techs  
## W = 0.89035, p-value < 2.2e-16
```

## Conclusión:

Mediante este test podemos concluir que ningun atributo sigue una distribucion normal.

Para comprobar la homocedasticidad (igualdad de varianzas entre los grupos que se van a comparar) se pueden realizar dos tipos de test *Levene* y *Fligner-Killeen*, en nuestro caso como los datos no siguen una distribución normal usaremos *Fligner-Killeen*.

Compararemos la varianza de **ConvertedComp** entre los grupos de *nivel\_edad*, *EdLevel*, *Employment* y *NEWOverTime*.

```
#Homocedasticidad  
fligner.test(ConvertedComp ~ nivel_edad, data = data_wo_na)
```

```
##  
## Fligner-Killeen test of homogeneity of variances  
##  
## data: ConvertedComp by nivel_edad  
## Fligner-Killeen:med chi-squared = 205.61, df = 3, p-value < 2.2e-16
```

```
fligner.test(ConvertedComp ~ EdLevel, data = data_wo_na)
```

```
##  
## Fligner-Killeen test of homogeneity of variances  
##  
## data: ConvertedComp by EdLevel  
## Fligner-Killeen:med chi-squared = 80.284, df = 5, p-value = 7.32e-16
```

```
fligner.test(ConvertedComp ~ Employment, data = data_wo_na)
```

```
##  
## Fligner-Killeen test of homogeneity of variances  
##  
## data: ConvertedComp by Employment  
## Fligner-Killeen:med chi-squared = 164.21, df = 2, p-value < 2.2e-16
```

```

fligner.test(ConvertedComp ~ NEWOvertime, data = data_wo_na)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: ConvertedComp by NEWOvertime
## Fligner-Killeen:med chi-squared = 58.465, df = 4, p-value = 6.095e-12

```

**Conclusión:** Dado que en todos los casos el valor de p-value es inferior a 0.05 se no se acepta la hipótesis de que las varianzas son homogéneas.

## 4.3 Pruebas estadísticas

### 4.3.1 Comparación entre más de dos grupos

En este caso queremos conocer si el número de lenguajes de programación que maneja un programador es similar en las situaciones laborales que se puedan tener (FULL\\_TIME, PART\\_TIME, FREELANCER, etc.).

Para esto usaremos la prueba de *Kruskal-Wallis*, que es un método no paramétrico que busca probar si un grupo de datos proviene de una misma población.

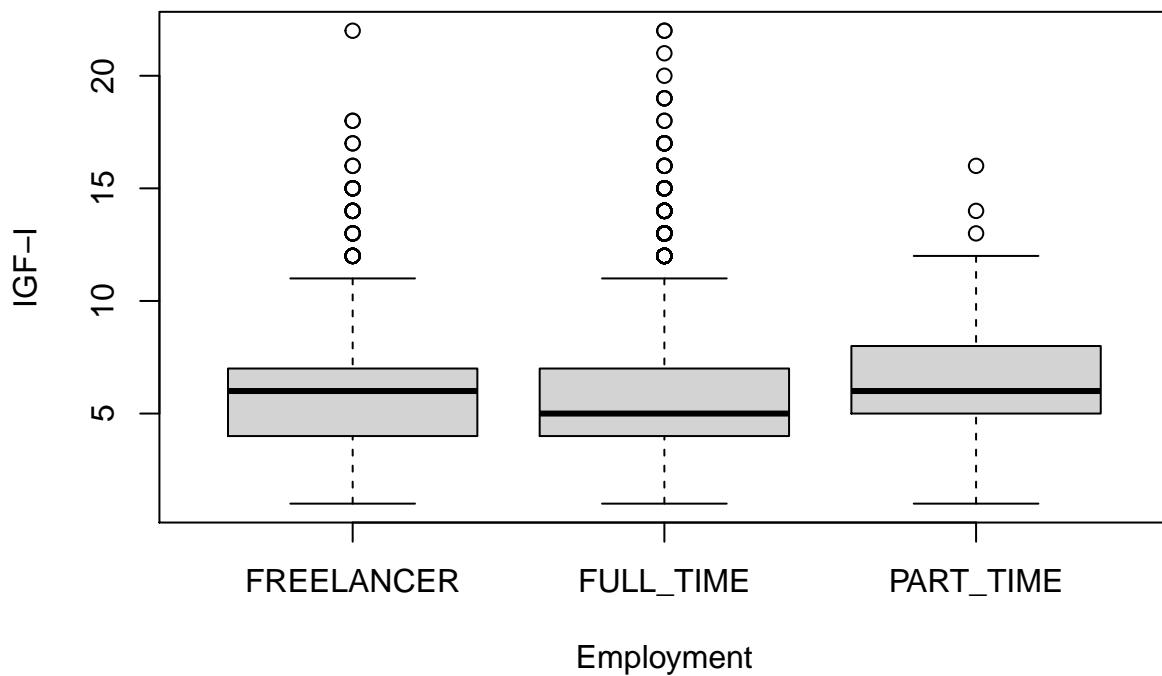
```

#Prueba de Kruskal-Wallis entre prog_langs y Employment
kruskal.test(prog_langs ~ Employment, data = data_wo_na)

##
## Kruskal-Wallis rank sum test
##
## data: prog_langs by Employment
## Kruskal-Wallis chi-squared = 35.952, df = 2, p-value = 1.56e-08

#Representación visual
boxplot(data_wo_na$prog_langs ~ data_wo_na$Employment, ylab="IGF-I", xlab="Employment")

```



El valor de  $p-value$  obtenido es menor al nivel de significancia se puede concluir que el número de lenguajes que maneja un programador dado su tipo de trabajo es diferente dependiendo de los tipos de situaciones laborales que se estén.

#### 4.3.2 Correlación con el salario

En nuestra búsqueda de conocer cómo se relacionan los distintos atributos del dataset buscaremos la correlación que existe entre estos campos, dado que nuestros valores son no paramétricos, utilizaremos la *correlación de Spearman*.

```
#Correlación ConvertedComp-Age
cor.test(data_wo_na$ConvertedComp,data_wo_na$Age, method="spearman")
```

```
##
##  Spearman's rank correlation rho
##
## data: data_wo_na$ConvertedComp and data_wo_na$Age
## S = 1.7418e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## 0.4528101
```

```
#Correlación ConvertedComp-YearsCodePro
cor.test(data_wo_na$ConvertedComp,data_wo_na$YearsCodePro, method="spearman")
```

```
##
##  Spearman's rank correlation rho
```

```

## 
## data: data_wo_na$ConvertedComp and data_wo_na$YearsCodePro
## S = 1.7118e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## 0.4622321

#Correlación ConvertedComp-WorkWeekHrs
cor.test(data_wo_na$ConvertedComp,data_wo_na$WorkWeekHrs, method="spearman")

## 
## Spearman's rank correlation rho
## 
## data: data_wo_na$ConvertedComp and data_wo_na$WorkWeekHrs
## S = 3.0514e+11, p-value = 4.053e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## 0.04136183

#Correlación ConvertedComp-db_techs
cor.test(data_wo_na$ConvertedComp,data_wo_na$db_techs, method="spearman")

## 
## Spearman's rank correlation rho
## 
## data: data_wo_na$ConvertedComp and data_wo_na$db_techs
## S = 3.2943e+11, p-value = 9.978e-05
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## -0.03492643

#Correlación ConvertedComp-prog_langs
cor.test(data_wo_na$ConvertedComp,data_wo_na$prog_langs, method="spearman")

## 
## Spearman's rank correlation rho
## 
## data: data_wo_na$ConvertedComp and data_wo_na$prog_langs
## S = 2.9869e+11, p-value = 6.31e-12
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## 0.06164661

#Correlación ConvertedComp-misc_techs
cor.test(data_wo_na$ConvertedComp,data_wo_na$misc_techs, method="spearman")

## 
## Spearman's rank correlation rho
## 
## data: data_wo_na$ConvertedComp and data_wo_na$misc_techs
## S = 3.0348e+11, p-value = 2.089e-07

```

```

## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.0465846

#Correlación ConvertedComp-collab_techs
cor.test(data_wo_na$ConvertedComp,data_wo_na$collab_techs, method="spearman")

## 
## Spearman's rank correlation rho
##
## data: data_wo_na$ConvertedComp and data_wo_na$collab_techs
## S = 2.7064e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.1497572

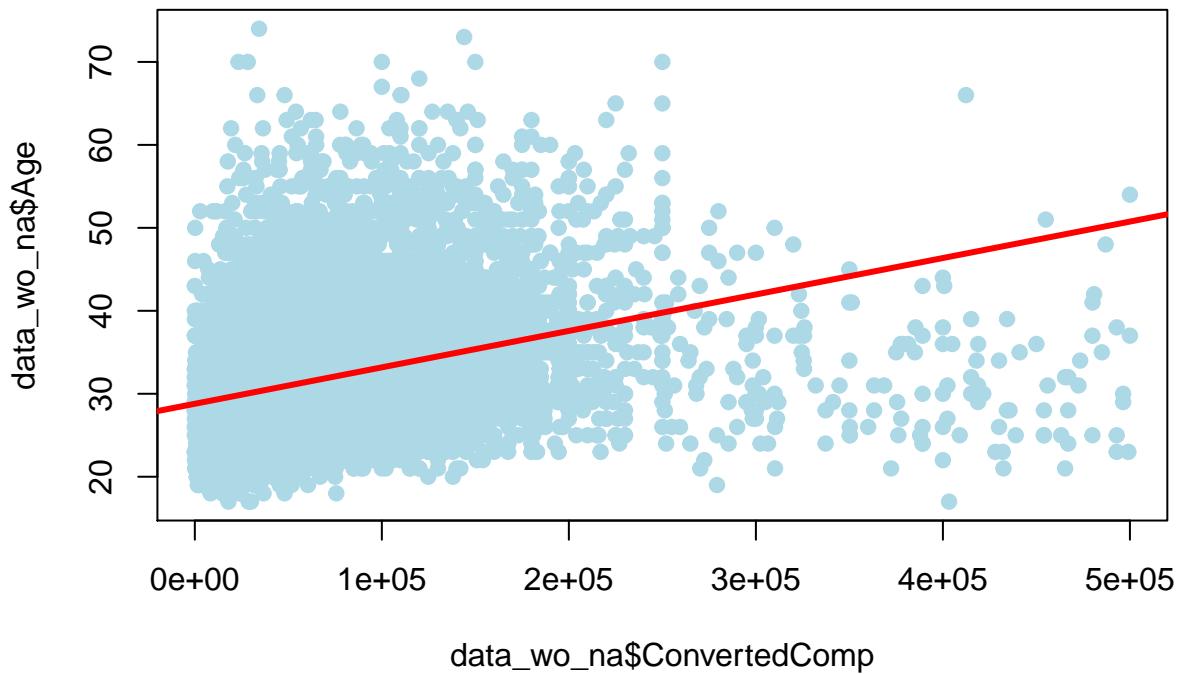
#Correlación ConvertedComp-plat_techs
cor.test(data_wo_na$ConvertedComp,data_wo_na$plat_techs, method="spearman")

## 
## Spearman's rank correlation rho
##
## data: data_wo_na$ConvertedComp and data_wo_na$plat_techs
## S = 2.8421e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.1071132

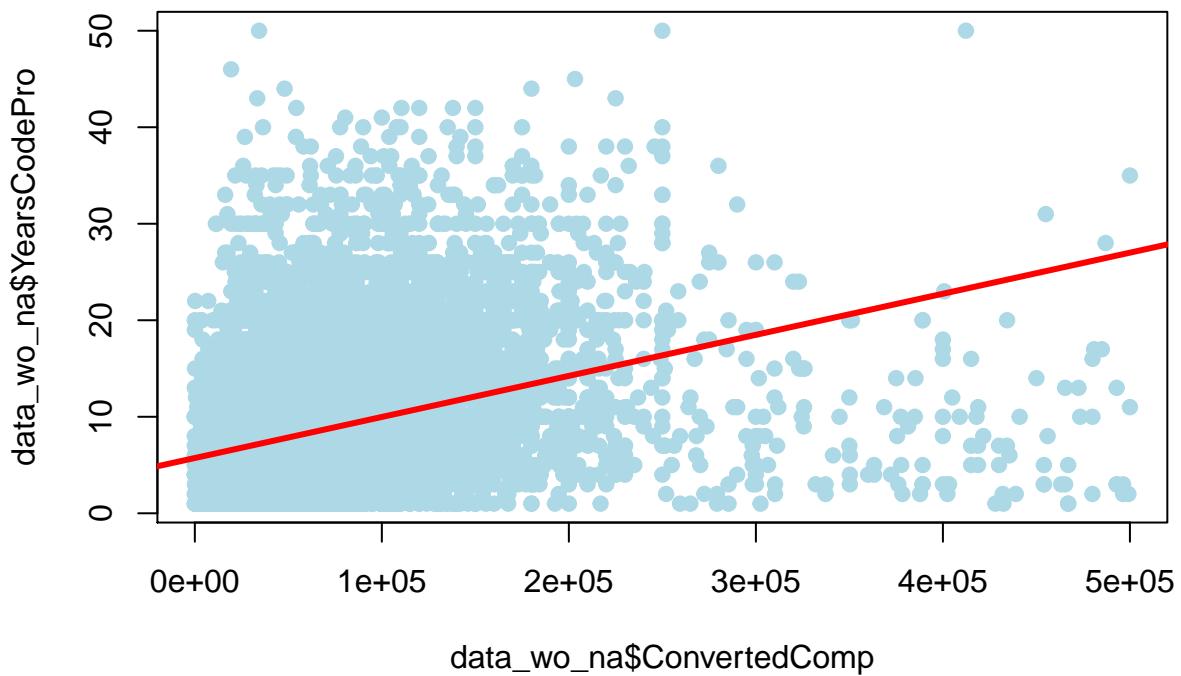
#Correlación ConvertedComp-web_techs
cor.test(data_wo_na$ConvertedComp,data_wo_na$web_techs, method="spearman")

## 
## Spearman's rank correlation rho
##
## data: data_wo_na$ConvertedComp and data_wo_na$web_techs
## S = 3.3637e+11, p-value = 2.557e-10
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.05673045

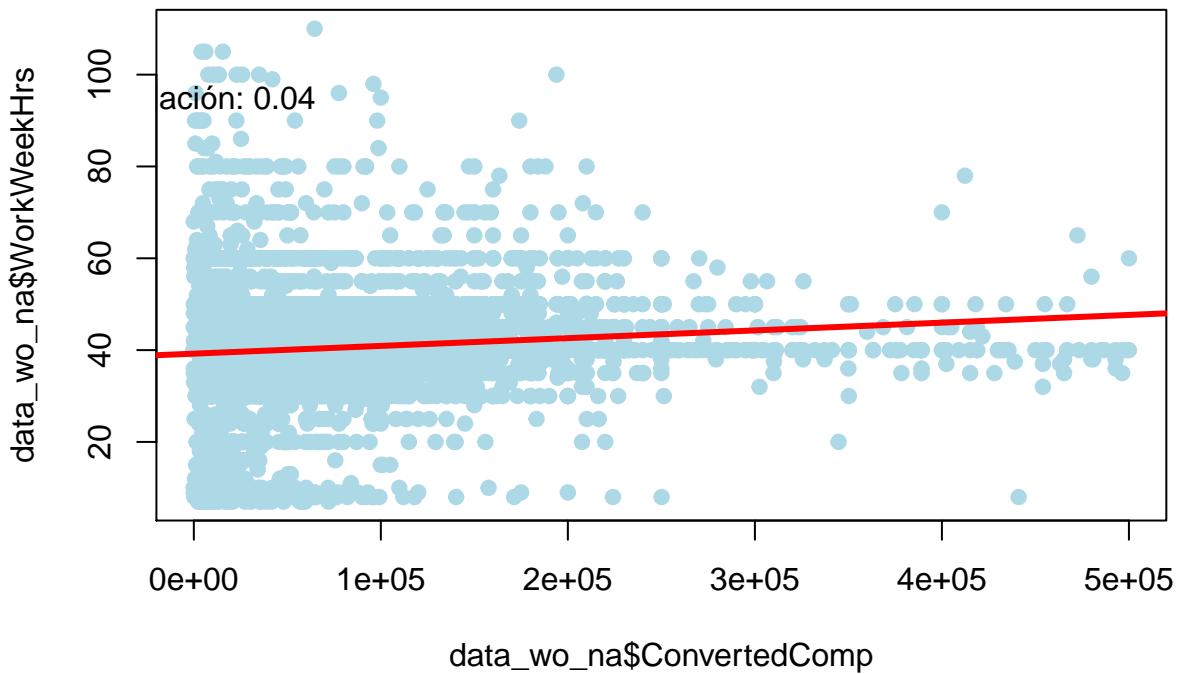
#Grafica ConvertedComp-Age
plot(data_wo_na$ConvertedComp, data_wo_na$Age, pch = 19, col = "lightblue")
abline(lm(data_wo_na$Age ~ data_wo_na$ConvertedComp), col = "red", lwd = 3)
text(paste("Correlación:", round(cor(data_wo_na$ConvertedComp, data_wo_na$Age, method="spearman"), 2)),
```



```
#Grafica ConvertedComp-YearsCodePro
plot(data_wo_na$ConvertedComp, data_wo_na$YearsCodePro, pch = 19, col = "lightblue")
abline(lm(data_wo_na$YearsCodePro ~ data_wo_na$ConvertedComp), col = "red", lwd = 3)
text(paste("Correlación:", round(cor(data_wo_na$ConvertedComp, data_wo_na$YearsCodePro, method="spearman"), 2)))
```



```
#Grafica ConvertedComp-WorkWeekHrs
plot(data_wo_na$ConvertedComp, data_wo_na$WorkWeekHrs, pch = 19, col = "lightblue")
abline(lm(data_wo_na$WorkWeekHrs ~ data_wo_na$ConvertedComp), col = "red", lwd = 3)
text(paste("Correlación:", round(cor(data_wo_na$ConvertedComp, data_wo_na$WorkWeekHrs, method="spearman"), 2)))
```



### Conclusión:

Vemos que los atributos de **Age** y **YearsCodePro** obtuvieron las correlaciones más grandes respecto al salario (0.45-0.46) y los atributos **WorkWeekHrs-db\_techs** fueron los que menos correlación tienen, esto nos da a entender el número de horas trabajadas no explica tanto el salario de los programadores, al igual que el número de tecnologías que manejan.

#### 4.3.3 Regresión lineal y multiple

En el paso anterior hemos conocido los campos que están más relacionados, ahora usaremos un método de regresión para poder conocer el salario (**ConvertedComp**) de los programadores a través de otras variables.

Atributos cuantitativos

- *YearsCodePro*
- *Age*
- *WorkWeekHrs*

Atributos cualitativos

- *Country*
- *EdLevel*
- *Employment*

Atributo a predecir

- *ConvertedComp*

```

#Seleccionamos los atributos a usar
data_wo_na_select<-data_wo_na[,c("ConvertedComp","YearsCodePro", "Age", "WorkWeekHrs","Country", "EdLevel")]

#Regresión lineal
m1 = lm(ConvertedComp ~ YearsCodePro,data=data_wo_na_select)

m2 = lm(ConvertedComp ~ Age,data=data_wo_na_select)

m3 = lm(ConvertedComp ~ WorkWeekHrs,data=data_wo_na_select)

#Regresión lineal multiple

m4 = lm(ConvertedComp ~ YearsCodePro + Country,data=data_wo_na_select)

m5 = lm(ConvertedComp ~ Age + EdLevel,data=data_wo_na_select)

m6 = lm(ConvertedComp ~ WorkWeekHrs + Employment,data=data_wo_na_select)

m7 <- lm(ConvertedComp ~ ., data = data_wo_na_select)

```

## 5 Representación de los resultados a partir de tablas y gráficas

```

#Coeficientes de determinación
tabla.coeficientes <- matrix(c(1, summary(m1)$r.squared,
                                2, summary(m2)$r.squared,
                                3, summary(m3)$r.squared,
                                4, summary(m4)$r.squared,
                                5, summary(m5)$r.squared,
                                6, summary(m6)$r.squared,
                                7,summary(m7)$r.squared),
                                ncol = 2, byrow = TRUE)

colnames(tabla.coeficientes) <- c("Modelo", "R^2")
tabla.coeficientes

```

```

##      Modelo      R^2
## [1,] 1 0.12967003
## [2,] 2 0.11660238
## [3,] 3 0.01140068
## [4,] 4 0.43183190
## [5,] 5 0.11986977
## [6,] 6 0.01849463
## [7,] 7 0.44407152

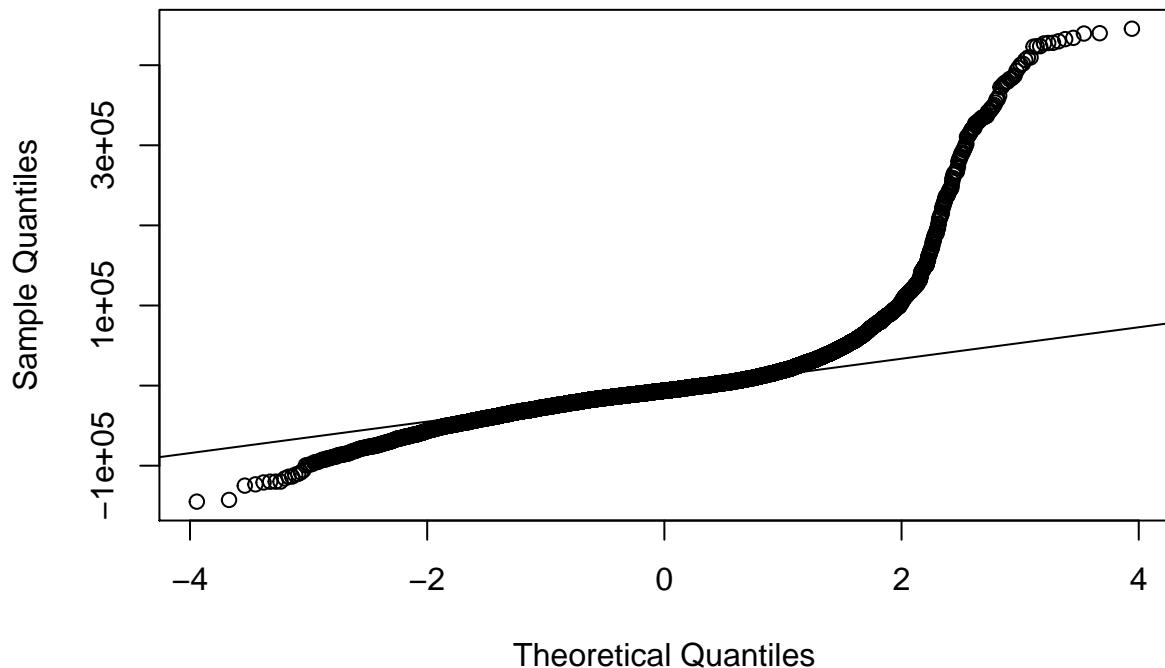
```

```

#Grafica modelo
qqnorm(m4$residuals)
qqline(m4$residuals)

```

## Normal Q-Q Plot



### Conclusión:

Vemos que los coeficientes no han salido tan altos, en este caso los modelos 4 y 7 son los que han dado mejores resultados con 44% de capacidad de predicción, en este caso nos quedamos con el modelo 4 ya que el 7 contiene todos los atributos y no estaríamos reduciendo entradas.

Se aplican 2 test con el modelo 4 para ver su funcionalidad.

```
#Test usando el modelo 4
test_1 <- data.frame(
  YearsCodePro = 10,
  Country = "United States"
)
# Predecir salario
predict(m4, test_1)
```

```
##           1
## 120118
```

```
test_2 <- data.frame(
  YearsCodePro = 10,
  Country = "France"
)
# Predecir salario
predict(m4, test_2)
```

```
##           1
## 61945.64
```

## 6 Conclusiones

Dados los resultados obtenidos en las pruebas que hemos realizado los resultados no han sido del todo satisfactorio ya que hemos encontrado una baja correlación entre los atributos de nuestro dataset lo que ha limitado la capacidad de predicción de los modelos de regresión lineal simple y múltiple, obteniendo solo un 44% de capacidad de acierto en la predicción de los salarios bajo los atributos de *YearsCodePro* y *Country*, por otra parte, el ejercicio de comparación entre grupos nos dice que el numero de lenguajes de programación aprendidos va a depender de la situación laboral, lo que tiene sentido.

Siguiendo esta misma línea, los resultados no son extremos, sino que igual se ve una coloración en los trabajos full time y freelancer, lo que da a entender que estas personas tienen mayor experiencia que un part time.

El salario depende en gran medida de muchas variables técnicas y socio demográficas. Al tener encuestados de varios países, donde las condiciones socio demográficas son muy distintas entre ellos, sin duda influye en el desarrollo de habilidades técnicas que garanticen un salario adecuado. Quizás al realizar un análisis más específico a los encuestados de un país determinado se mejore la capacidad predictiva en los salarios.

Quizás al tener una variable donde se detalle el nivel del idioma inglés que domine el encuestado se hubiera podido realizar una análisis más específico. También es importante reconocer que la comunidad de StackOverflow no solo se compone de desarrolladores, sino de un conjunto variado de perfiles técnicos que a diario usan el sistema de preguntas/respuestas. Algunos perfiles enfocados más a la investigación o ciencia pueden verse beneficiados al tener salarios más altos que un simple programador.

Los resultados de nuestros modelos quizás no han sido tan buenos, pero nos ha permitido visualizar la importancia de que la información que estemos usando como fuente tenga la menor cantidad de valores ausentes posibles, ya que con los tratamientos de integración y limpieza se puede mejorar estos resultados de forma estratégica, pero siempre nos estarán alejando de resultados fundamentados en los datos de origen.

## 7 Código

El código fuente y el dataset usado para el análisis presentado en este documento se encuentra disponible en el siguiente repositorio de GitHub: <https://github.com/gpbonillas/stackoverflow2020>