

Introducción

Mtro. José Gustavo Fuentes Cabrera



Facultad de Estudios Superiores

Acatlán

Apuntes de Análisis Multivariado

Licenciatura en Actuaría

Índice

1. Introducción	3
2. Plan de estudios	3
3. Contextualización	4
4. Software	7
5. Introducción a la tecnología necesaria para el aprovechamiento del curso	9
5.1. Sistema operativo basado en Unix	10
5.1.1. Archivos y directorios	10
5.1.2. Estructura de directorios	10
5.1.3. La terminal	11
5.1.4. Lista de comandos básicos	11
5.2. Lenguaje de programación python	12
5.2.1. Breve historia	12
5.2.2. Sintaxis	12
5.2.3. Cadenas de caracteres	12
5.2.4. Estructuras de datos básicas	13
5.2.5. Sentencias de control	13
5.2.6. Funciones	13
5.2.7. Importar módulos	13
5.2.8. Manejo de archivos	14
5.3. Git	14

1. Introducción

Los contenidos del presente curso son de particular relevancia para la vida profesional del actuario. El análisis multivariante se ha convertido en una de las herramientas más potentes de análisis en cualquier rama del conocimiento e industria. La presencia en el mercado de software especializado y técnicas de aprendizaje automático permite tener acceso inmediato a potentes herramientas analíticas a cero costo. El enfoque que tendremos en este curso será completamente aplicado, si bien revisaremos la base teórica suficiente que sustenta cada técnica, el objetivo es que el estudiante desarrolle sus capacidades analíticas y de aplicación cualesquiera que sea su interés de desempeño profesional. En la materia de Análisis Multivariado, revisaremos desde el punto de vista estadístico, los problemas generales de reducción de dimensiones, predicción, clasificación y agrupamiento. En este contexto, la materia prima con la que contaremos serán datos en cantidad suficiente para poder determinar las relaciones entre una variable dependiente con un conjunto de predictoras, o en su defecto, los patrones y relaciones subyacentes dentro de un conjunto de variables.

2. Plan de estudios

El plan de estudios en su versión 2014 tuvo una revisión y se añadieron algunos temas como unidades separadas, sin embargo, muchos de ellos no tienen el enfoque demandado por el mercado laboral, el temario oficial contempla:

1. Análisis descriptivo multivariado
2. Componentes principales
3. Escalamiento multidimensional
4. Análisis de conglomerados
5. Análisis factorial
6. Análisis discriminante

Lo anterior, queda muy corto ante las necesidades actuales del profesional de la Actuaría, por ejemplo, no existe enlace tecnológico explícito, los temas 2, 3 y 5 son subtemas de reducción de dimensionalidad, el tema 1 es muy pequeño comparado contra todas las capacidades de visualización de datos que existen hoy y se basan fundamentalmente en resumir la información matemáticamente. Conglomerados solo contempla el método jerárquico (de presencia casi nula en la práctica) y por último análisis discriminante es una técnica vieja y solo se utiliza como una opción más dentro de la modelación supervisada. Por ello, se propone al grupo un temario mucho más útil y actualizado incorporando técnicas en el estado del arte y tecnología de punta sin dejar de revisar los temas que la cátedra exige.

1. Introducción: Breve contextualización y uso de software de última generación.
2. Análisis Factorial y Componentes principales: Técnicas de reducción de dimensiones
3. Análisis Clúster: Modelación no supervisada por medio de técnicas jerárquicas, de optimización y difusas.
4. Modelación Supervisada: Técnicas de machine learning aplicadas desde el punto de vista estadístico: análisis discriminante, árboles de decisión, regresión logística, redes neuronales, máquinas vector soporte, k vecinos más cercanos, gradiente estocástico, Bayes Ingenuo, ensambles.

3. Contextualización

En el mundo actual (2019), el análisis multivariante ha evolucionado al convertirse en el soporte matemático (estadístico) de las técnicas de aprendizaje máquina (machine learning) en contraste con su otrora función de componente aislado y extensión a varias dimensiones de la estadística descriptiva e inferencial. En este entendido, se ha conformado una nueva ola de disciplinas de la ciencia que ha pasado por una vertiginosa transformación. Lo anterior fue posible gracias a la amplia disponibilidad de datos digitales como consecuencia inmediata del proceso de digitalización

de las compañías. Una vez que los datos estuvieron disponibles, surgió la disciplina denominada inteligencia de negocios (Business Intelligence: BI) cuyo propósito fundamental es convertir el dato en información relevante mediante herramientas de visualización, es análogo a la estadística descriptiva donde resumimos información y la presentamos por medio de gráficos o estadígrafos (media, moda, varianza, cuantiles, etc) para tener un mejor entendimiento de los datos que nos han sido proporcionados. En el siguiente nivel se encuentra la disciplina conocida como minería de datos (surgido a finales de los 90 y principios del milenio), cuyo propósito es la obtención de patrones e información no trivial dentro de grandes volúmenes de datos, aquí surge una duda común, ¿en qué se diferencia con respecto al BI?, de igual forma que la estadística descriptiva con la estadística inferencial, el propósito está enfocado en la predicción (Donde se necesitará matemática más sofisticada). Durante los últimos años se han acuñado nuevas definiciones y se han extendido las disciplinas. Una de estas es el término Analytics (analítica en español) que consiste en el descubrimiento, interpretación y comunicación de patrones dentro de los datos, como vemos, es parecida a minería de datos, con el añadido de la comunicación e interpretación y no solo la extracción. El término sigue siendo ampliamente utilizado, sin embargo una nueva confusión surgió con la popularización del término Ciencia de datos. Ciencia de datos se considera una evolución multidisciplinaria en los campos de análisis de negocio, ciencias de la computación, modelación matemática, estadística, analítica y minería de datos. El siguiente diagrama presenta un buen resumen de los tópicos asociados a la ciencia de datos:

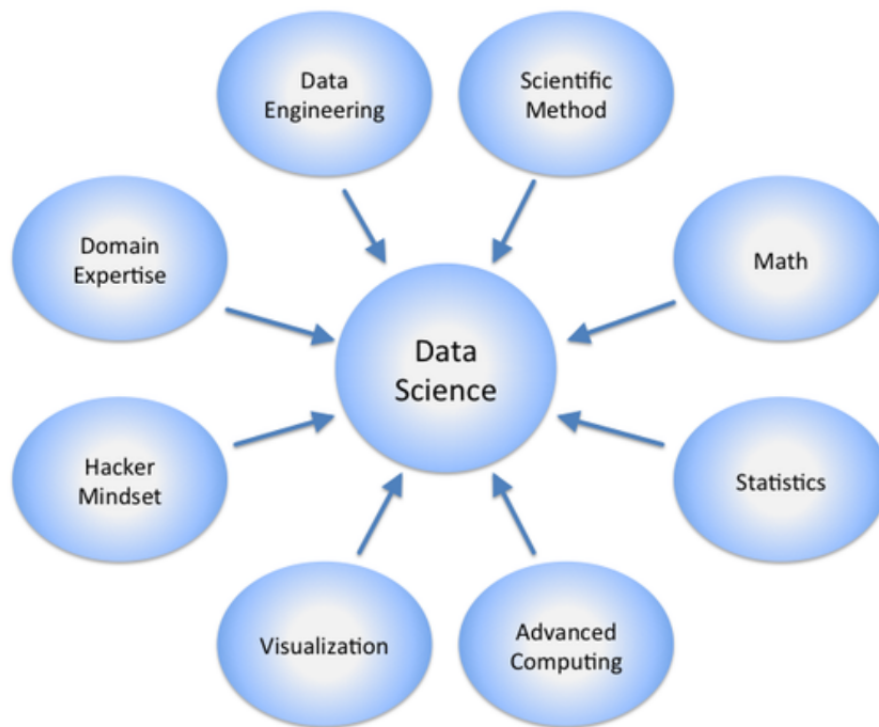


Figura 3.1: Componentes de la ciencia de datos

Es decir, un científico de datos debe poseer además de las habilidades de analítica, un fuerte componente de habilidades computacionales, inventivas y científicas.

Con respecto al término Inteligencia Artificial (malamente usado de forma intercambiable con aprendizaje automático o ciencia de datos inclusive) se refiere a la rama de las ciencias de la computación encargada del estudio y creación de máquinas que asemejen comportamientos humanos inteligentes, a saber:

- Visión
- Reconocimiento de lenguaje
- Planeación
- Resolución de problemas

La confusión se da debido a que el estado actual de la inteligencia artificial depende muchísimo de la abundancia de datos y, por tanto, del correcto entrenamiento de algoritmos (machine learning) que aprendan de los mismos para realizar tareas específicas, por ello a menudo se confunde a la ciencia de datos con la inteligencia artificial, aunque convergan en estos particulares tópicos, son más bien conjuntos intersectados.

Por último, el término Big Data se añade a la discusión, BigData se refiere a toda información (estructurada, no estructurada, semiestructurada) que no puede ser procesada o analizada usando procedimientos y herramientas tradicionales. La situación actual está enfocada en la recolección y procesamiento de esas enormes cantidades de información (decenas de miles de terabytes en adelante) mediante la infraestructura adecuada de hardware y software, por tanto, es independiente de los términos revisados con anterioridad ya que se añaden únicamente los componentes de volumen, velocidad y variedad de datos que fungen como materia prima del científico de datos, el analista de información, el diseñador de tableros de BI, etc.

4. Software

Anteriormente, el software estadístico estaba encasillado dentro de un nicho muy específico cuyo propósito principal era la automatización de los cálculos propios de la disciplina. Actualmente, los simples cálculos se obvian y se opta por empoderar las herramientas de software expandiendo su funcionalidad. En el mercado existen tres jugadores principales, sin embargo, existe una gran cantidad de alternativas (libres y comerciales) la mejor herramienta no existe, cada quien resolverá los problemas que se le planteen de acuerdo a su necesidad, comodidad y recursos disponibles. En primera instancia, las herramientas deberán ser separadas por funcionalidad:

- Herramienta de manipulación de datos: Necesaria para construir tablas analítica de datos o alguna estructura de datos (resumen, consulta) que fungirá como entrada, adicionalmente, para realizar la limpieza y pretratamiento de los datos.

- Herramienta de visualización de datos: Aunque no indispensable, nos permite tener “primeras impresiones” de lo que muestran los datos.
- Software estadístico: Necesario para realizar análisis estadístico básico, descriptivo e inferencial.
- Software para machine learning: Necesario para generar modelos de soporte no supervisados y supervisados.
- Software para presentaciones: Necesario para comunicar resultados, proyectos, hallazgos, estrategias, etc.
- Hoja de cálculo: Herramienta hiperbásica de análisis, permite crear reportes, tablas pivote y gráficos de forma muy simple.
- Lenguaje de programación: Una herramienta que se vuelve cada vez más imprescindible. Brinda la capacidad de automatizar tareas y expandir nuestras posibilidades mediante la creación de herramientas propias, enlace a API's, encapsulamiento del conocimiento, integración tecnológica, etc.

En el mercado hay tres suites que cubren las primeras 4 necesidades a cabalidad; dos son alternativas de software libre y una de ellas de software propietario: Python, R-Project y SAS respectivamente. En cuanto al lenguaje de programación, las alternativas libres lo poseen, mientras que SAS corresponde más a un lenguaje de manipulación de datos que a uno de programación. Se obviará la parte de ofimática.

No existe una “mejor” herramienta, todas poseen ventajas y desventajas, la elección dependerá de las circunstancias de cada empresa/persona.

Es común que en organizaciones grandes se privilegie la “facilidad de uso”(herramientas GUI, plataforma universal, mínimo conocimiento tecnológico) versus el potencial (es a menudo más costoso capacitar al personal y/o atraer talento específico), por ello SAS es un software muy popular en estas circunstancias, sin embargo, en la práctica es utilizado como manejador de base de datos y muy pocos profesionales explotan al máximo sus capacidades estadísticas. Es importante también mencionar que la creación de modelos predictivos es relativamente sencilla aunque su implementación práctica depende muchísimo de que tan integrada esté la tecnología SAS a los sistemas de la organización. En el caso de R-Project,

existe una gran comunidad que le da soporte y es muy completo en cuanto a herramientas, en su contra juega que su integración con otros sistemas no es a menudo transparente y tiene que adaptarse a él. R-Project es un software de nicho que ha ido escalando en funcionalidad, es ampliamente utilizado en la academia donde un ambiente standalone es suficiente, sin embargo, su escalabilidad no es óptima para proyectos de gran envergadura.

En cuanto al lenguaje de programación Python, es un poderoso y muy simple lenguaje de programación que nos brinda posibilidades ilimitadas. Su potencia está dada por la gran cantidad de librerías disponibles, la gran comunidad que le da soporte, la integración transparente con múltiples tecnologías y su curva de aprendizaje acelerada. Recientemente Python desplazó a Java y C++ como el lenguaje de programación más utilizado del mundo, adicional a esto, es una habilidad cada vez más requerida por el mercado laboral (incluso en grandes organizaciones) y permite formar una disciplina de desarrollo de software más avanzada. Por ello, nuestra elección para este curso será Python.

Por último, es muy importante señalar que el valor de un profesional no se relaciona con el software que utiliza, sino con su capacidad para GENERAR VALOR.

5. Introducción a la tecnología necesaria para el aprovechamiento del curso

En esta sección presentaremos los pormenores para poder desarrollar el curso de manera óptima, si existe fallo en alguno de los conceptos que expondremos consistirá en un bloqueador, por tanto, el estudiante debe dominar funcionalmente cada tema (no necesariamente a profundidad) antes de proceder con los contenidos matemáticos de las unidades subsecuentes.

5.1. Sistema operativo basado en Unix

Para aspirar a ser un científico de datos, se necesita poder manipular y aprovechar las funcionalidades de un sistema operativo serio, aunque windows sea el líder en cómputo personal, sus importantes brechas de seguridad e ineficiencias en el uso del hardware de nuestra computadora, hacen que windows sea una opción que debemos descartar. La potencia de un sistema operativo basada en Unix justifica su curva de aprendizaje más pronunciada, el sacrificio vale la pena. El estudiante es libre de elegir entre el sistema operativo OSX de Apple o diversas distribuciones de linux (Ubuntu, CentOS, Debian, Mint, OpenSUSE, ArchLinux, etc) que además poseen la ventaja de ser de código abierto y completamente personalizables. En la presente introducción, eligiremos linux como sistema operativo, la gran mayoría de los conceptos serán aplicables con pequeñas o nulas modificaciones a OSX.

5.1.1. Archivos y directorios

Se inicia con el concepto de archivo, un archivo es un conjunto de datos con un nombre asignado. En sistemas basados en UNIX los nombres de los archivos pueden contener cualquier caracter excepto "/" y tener una longitud máxima de 256 caracteres. Posteriormente definimos directorio, un directorio es un contenedor de archivos que presentará una estructura de árbol. En virtud de lo anterior, todo archivo podrá ser referenciado por su nombre o su ruta completa(path), por ejemplo:

```
/home/jose/train.csv
```

Se refiere a un archivo llamado `train` de tipo `csv` que se encuentra en el subdirectorio `jose` del directorio `home`.

5.1.2. Estructura de directorios

En Linux, el directorio principal llamado raíz(root), es el directorio donde inicia la jerarquía del árbol de directorios y se denota por el caracter `/`. Directamente debajo de éste, se encuentran directorios importantes que

contienen programas esenciales y archivos de configuración, por ejemplo : `/etc`, `/bin`, `/dev`, `/usr`. Con respecto a los usuarios, por cada usuario creado en el sistema le corresponderá un directorio creado en `/home`, así, por ejemplo `/home/jose` corresponderá al directorio `home` del usuario `jose`, la notación `~` es un sinónimo de la carpeta `home` del usuario actual.

5.1.3. La terminal

Un emulador de terminal es un programa que permite interactuar con la computadora en un ambiente de línea de comandos (normalmente se trabaja en una interfaz gráfica). A través de la terminal ejecutamos variedad de comandos que nos permiten realizar tareas de diversos grados de complejidad. A través de los comandos tenemos acceso a todas las funcionalidades del sistema operativo por lo que a menudo se prefiere su utilización en lugar de una interfaz gráfica.

5.1.4. Lista de comandos básicos

A continuación presentamos una lista de comandos junto con su funcionalidad:

- `cd`: Cambiar de directorio
- `ls`: listar archivos/directorios
- `cp`: copiar archivos
- `mkdir`: crear directorio
- `rmdir`: borrar directorio
- `mv`: mover archivos
- `rm`: borrar archivo
- `cat`: imprimir contenidos de un archivo en pantalla

5.2. Lenguaje de programación python

En la presente sección, daremos una breve introducción al lenguaje de programación Python en el cual desarrollaremos todo el curso.

5.2.1. Breve historia

Python es un poderoso lenguaje de programación multipropósito que se ha abierto paso en el mundo hasta convertirse en el más utilizado de acuerdo al IEEE. Fue creado en 1991 por Guido Van Rossum. El lenguaje es multiplataforma y está disponible para sistemas MacOS, Linux y Windows, pertenece a la comunidad open source y no tiene costo. Es un lenguaje fuertemente tipado, dinámicamente tipado e implícitamente tipado, sensible a mayúsculas y orientado a objetos.

5.2.2. Sintaxis

Python no tiene caracteres de terminación de bloques (llaves, punto y coma), los bloques son especificados mediante indentación. Aquellas sentencias que requieran un nivel de indentación forzoso se terminan con (:). Las asignaciones se realizan mediante el signo de =, las comparaciones con ==, se permiten los operadores de incremento y decremento += y -= respectivamente. Los comentarios de una línea se definen mediante el símbolo #, mientras que los comentarios multilinea se realizan a través de cadenas de caracteres multilinea. Los operadores relacionales básicos complementarios son <=, >=, <, >, !=.

5.2.3. Cadenas de caracteres

Las cadenas de caracteres pueden declararse tanto con comillas simples como con comillas dobles. Anteponiendo u a una cadena la convierte en texto unicode.

5.2.4. Estructuras de datos básicas

- Listas: Arreglos unidimensionales.
- Tuplas: Arreglos unidimensionales inmutables.
- Diccionarios: Arreglos asociativos.
- Conjuntos: Arreglos de elementos únicos.

Todos los índices iniciarán en 0, se pueden seleccionar rangos de los arreglos mediante el caracter ":".

5.2.5. Sentencias de control

- if: Para evaluar condicionales, se combina con elif(else-if) y else.
- for: Enumera a través de los elementos de un iterable.
- while: Repite un bloque de código mientras se cumpla una condición.

5.2.6. Funciones

Las funciones se declaran mediante la palabra reservada `def`, los argumentos opcionales se declaran posteriores a los argumentos obligatorios (asignando a priori un valor por defecto). Las funciones lambda son funciones específicas que se componen de un solo enunciado.

5.2.7. Importar módulos

Las librerías externas usando la palabra reservada `import`, por ejemplo: `import [nombre de la librería]`, si se requiere importar una sola función de una librería, se puede utilizar la siguiente forma: `from [nombre de librería] import [nombre de función]`

5.2.8. Manejo de archivos

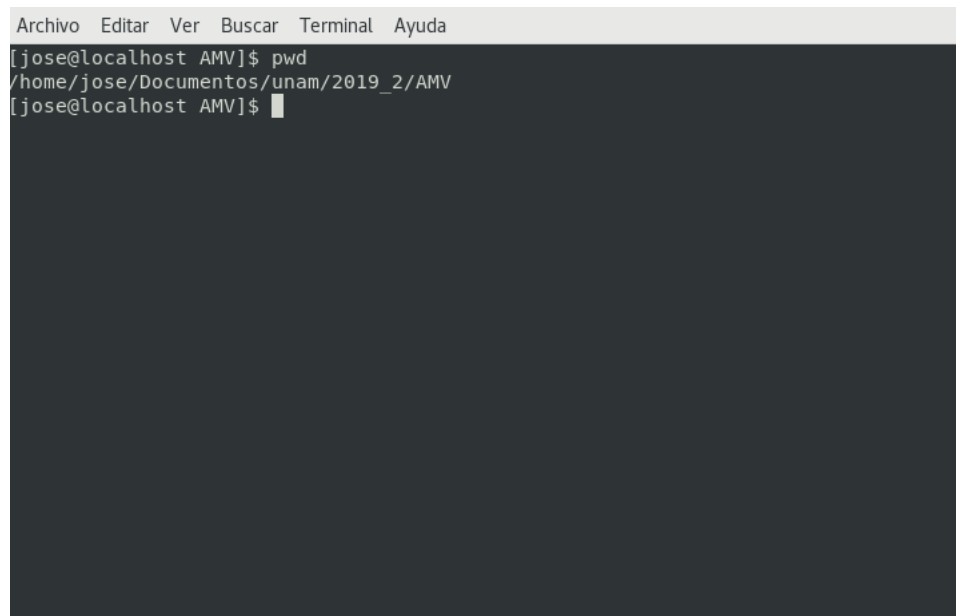
Para el manejo de archivos utilizamos la función `open`, si queremos manipular archivos dentro del sistema operativo podemos usar la librería `os`.

5.3. Git

Una herramienta adicional que hará nuestro trabajo más fácil además de ser una tendencia a nivel mundial será el uso de Git, el cual es muy popular en el mundo del desarrollo de software. Como actuarios, git no será utilizado a profundidad, sin embargo, será una herramienta imprescindible para compartir código, prácticas, tener todos las mismas versiones y evitar compartir archivos por correo electrónico u otros medios de manera innecesaria. Git es un software de control de versionado escrito por Linus Torvalds (el creador del kernel de linux), para poder instalarlo, podemos seguir las instrucciones del sitio <https://git-scm.com/download/>. Una manera más cómoda de gestión de repositorios git es BitBucket(<https://bitbucket.org/>). Consiste en una plataforma en la nube para gestionar repositorios de código. Existen alternativas como Github <https://github.com/> y Gitlab <https://about.gitlab.com/>. El repositorio que tendrá todo el contenido de nuestro curso está en la dirección https://bitbucket.org/jgfcunam/amv_2019_2/src.

Ahora mostraremos el proceso para clonar el repositorio, conforme avancemos en el curso, dicho repositorio será actualizado periódicamente y será cuestión únicamente de decirle a nuestro repositorio local que se sincronice con el repositorio remoto en la nube.

Abriendo una terminal, nos posicionamos en el directorio donde queramos clonar el repositorio, en este ejemplo es `~/Documentos/unam/2019_2/AMV`

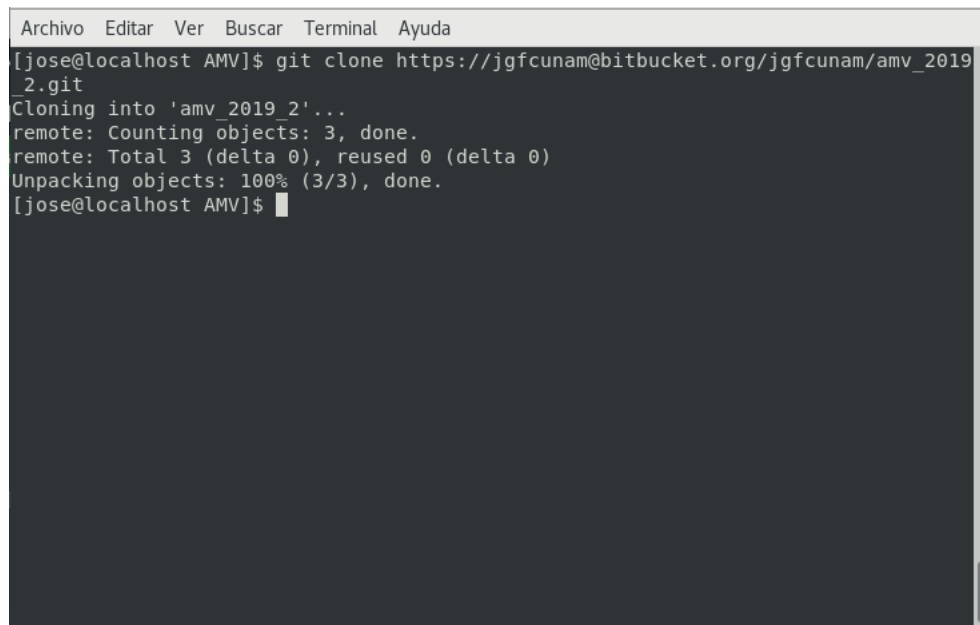
A screenshot of a terminal window with a menu bar at the top containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal text shows a user named 'jose' at 'localhost' in a directory named 'AMV'. They have entered the 'pwd' command, and the output is '/home/jose/Documentos/unam/2019_2/AMV'. The prompt is now ready for the next command.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[jose@localhost AMV]$ pwd
/home/jose/Documentos/unam/2019_2/AMV
[jose@localhost AMV]$
```

Figura 5.2: Directorio de trabajo

Una vez ahí, escribimos en la terminal:

```
git clone https://jgfcunam@bitbucket.org/jgfcunam/amv_2019_2.git
```

A screenshot of a terminal window with a menu bar at the top containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal text shows a user named 'jose' at 'localhost' in the 'AMV' directory running the command 'git clone https://jgfcunam@bitbucket.org/jgfcunam/amv_2019_2.git'. The output indicates the repository is being cloned into 'amv_2019_2', with 3 objects counted and unpacked successfully. The prompt returns to '[jose@localhost AMV]\$' with a cursor.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[jose@localhost AMV]$ git clone https://jgfcunam@bitbucket.org/jgfcunam/amv_2019_2.git
Cloning into 'amv_2019_2'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
[jose@localhost AMV]$
```

Figura 5.3: Repositorio Clonado

y listo, tendremos clonado el repositorio de la nube en nuestro equipo local.