**P = tabla padron**

**L = tabla localidades\_sensales**

**DD = Tabla Diccionario deptos**

**DC = Tabla Diccionario de clases**

**S = Tabla Salarios**

Sección Decisiones tomadas:

1) Una decisión controversial tomada fue eliminar los valores NC(Nan) del atributo “establecimiento” en la tabla padrón. Tomamos esta decisión, ya que necesitábamos una PK para P, y no encontrábamos ninguna viable, que no sea incluir una gran cantidad de atributos. De esta manera los atributos “establecimiento” + “razón social” formaban PK.

Dicho esto, creemos que esta decisión no fue la mejor por algunas razones, entre ellas; Primero, a través de una consulta de SQL, nos dimos cuenta que al eliminar las filas con establecimientos NC, eliminamos TODOS los registros de CABA y Tierra del Fuego.

Además, eliminar más de 100 casos, solo para encontrar una PK viable, no nos parece razón suficiente para justificar la eliminación de tantos datos viables. Por lo tanto consideramos que esta decisión no fue la más acertada y que poseíamos alternativas mejores. Una solución distinta podría haber sido crear una columna con índices que sirva como PK.

Nos dimos cuenta que faltaban los registros de CABA y Tierra del Fuego a través de una GQM:

G: Consultar si, al eliminar las filas que poseían valores ‘NC’ en la columna ‘Establecimiento’, no se perdió cierta información.

Q: ¿Las provincias resultantes en la tabla limpia de padrón coinciden con todas las provincias posibles?

M: """

SELECT DISTINCT provincias.provincia, count(\*) as cant\_operadores

FROM padron

INNER JOIN provincias

ON provincias.provincia\_id = padron.provincia\_id

GROUP BY provincias.provincia;

"""

2) Otra complicación que tuvimos fue para vincular la tabla P con la tabla S con respecto a la actividad asociada a cada rubro, ya que cada rubro estaba asociada a más de una actividad. Para poder dar una respuesta más directa y concluyente , decidimos crear a partir de la tabla DC una tabla, la cual vincula un rubro con una única actividad. De esta manera si a un padrón le corresponde un rubro X, este rubro estará vinculado con una única actividad en esta nueva tabla. En nuestra opinión esta decisión fue acertada, ya que las conclusiones son más directas pero igualmente enriquecedoras.

Consulta GQM asociada:

G: Verificar primera forma normal dentro del dataset de padrón electoral

Q: ¿Se encuentra la columna de ‘Rubro’ en 1FN?

M: """

SELECT DISTINCT rubro

FROM padron;

"""

3) Otra decisión que tomamos tiene que ver con el atributo w\_median de la tabla CLAE. En este atributo eliminamos los sueldos que tenían valor -99. Esto lo hicimos, no solo porque un valor negativo se corresponde a un valor mal subido, sino también porque nos daría reflexiones erróneas al analizar promedios o desvíos estándar.

4) Una decisión muy importante que tomamos fue cómo limpiar el atributo “departamento” dentro de la tabla P. El principal problema radicó en que habían muchos departamentos (un 14% aproximadamente, como se evidencia en la tabla “tabla\_consulta\_2” dentro del archivo limpieza.py) que no estaban presentes en la tabla DC, por lo que no eran departamentos válidos. Esta limpieza era sumamente importante, porque podía eliminar elementos de nuestra fuente principal, lo que impactaría negativamente en nuestro análisis.

Sección Procesamiento de Datos

Respuesta b):

En este ejercicio debíamos decidir si las tablas estaban en primera forma normal (1FN) o no. Retomando la definición, un esquema se encuentra en 1FN si y sólo si se prohíbe relaciones dentro de relaciones o relaciones como valores de atributos dentro de tuplas. Básicamente pide que cada valor de la relación sea ATÓMICO. Por lo tanto, analizamos las tablas para buscar algún atributo que no sea presentado de forma atómica. Nos dimos cuenta que dos atributos no se encontraban en 1FN. En P puede verse que los atributos “PRODUCTOS” y “RUBRO” no se encontraban en 1FN, ya que tenían varios valores(tuplas) en una misma casilla. Luego para adaptarlas a la primera forma normal, creamos una nueva tabla para cada relación. Para esta tabla desanidamos las tuplas en valores atómicos dentro de su mismo atributo y luego agregamos los atributos que servían como PK(primary key) de la relación original. El proceso para encontrar la PK de este esquema no fue nada fácil y a pesar que funcione perfectamente como PK, en la sección “Decisiones tomadas” puede verse como la elección de nuestra PK fue un tanto cuestionable.

Respuesta c) insertar tablas

En este ejercicio debíamos definir las dependencias funcionales(DF) y escribirlas. Retomando la definición, las DF son una herramienta formal para el análisis de esquemas. que permite detectar y describir problemas descritos previamente

Básicamente es una restricción entre dos conjuntos de atributos X e Y de un esquema . Si tengo x—>y , esto significa que los valores que tomen los elementos de Y dependen de los valores que tomen los elementos de X. Antes de definir las DF, debíamos determinar qué atributos eran PK. Para eso realizamos numerosas consultas SQL para determinar qué combinación de atributos funcionaban como clave, como se muestra en el código de nuestras consultas.. Dicho esto, la primera DF se origina de la PK hacia todos los atributos, ya que por definición una PK es la combinación de la MÍNIMA cantidad de atributos que haga que cada tupla/fila sea ÚNICA considerando solo esos atributos. Luego otras DF eran bastante evidentes,ya que los conjuntos X e Y estaban directamente relacionados. Por ejemplo pais y pais\_id a simple vista conformaban una DF. Sin embargo en todos los casos se realizaron consultas SQL para verificar nuestras afirmaciones. Algunos ejemplos pueden verse en el archivo “consultas\_optativas”.

Las DF para cada tabla pueden verse en la presentación “Dependencias funcionales” desde la diapositiva 1 hasta la 7.

Respuesta d) En este ejercicio debíamos descomponer los esquemas con los que trabajamos en el punto anterior para que todos ellos se encuentren en 3FN y sin perder información. Retomando la definición, un esquema R está en 3FN si ningún atributo no primo de R depende transitivamente de la PK. Básicamente si algún atributo que no forma parte de la PK o de alguna SK posible se puede relacionar transitivamente, entonces NO se cumple 3 FN. Varios esquemas de los presentados anteriormente NO se encuentran en 3FN, ya que contienen varias DF transitivas. Estas DF transitivas fueron coloreadas con el color **negro** en los esquemas anteriores. Para darnos cuenta de esto, realizamos consultas en SQL mostradas en el código adjuntado.

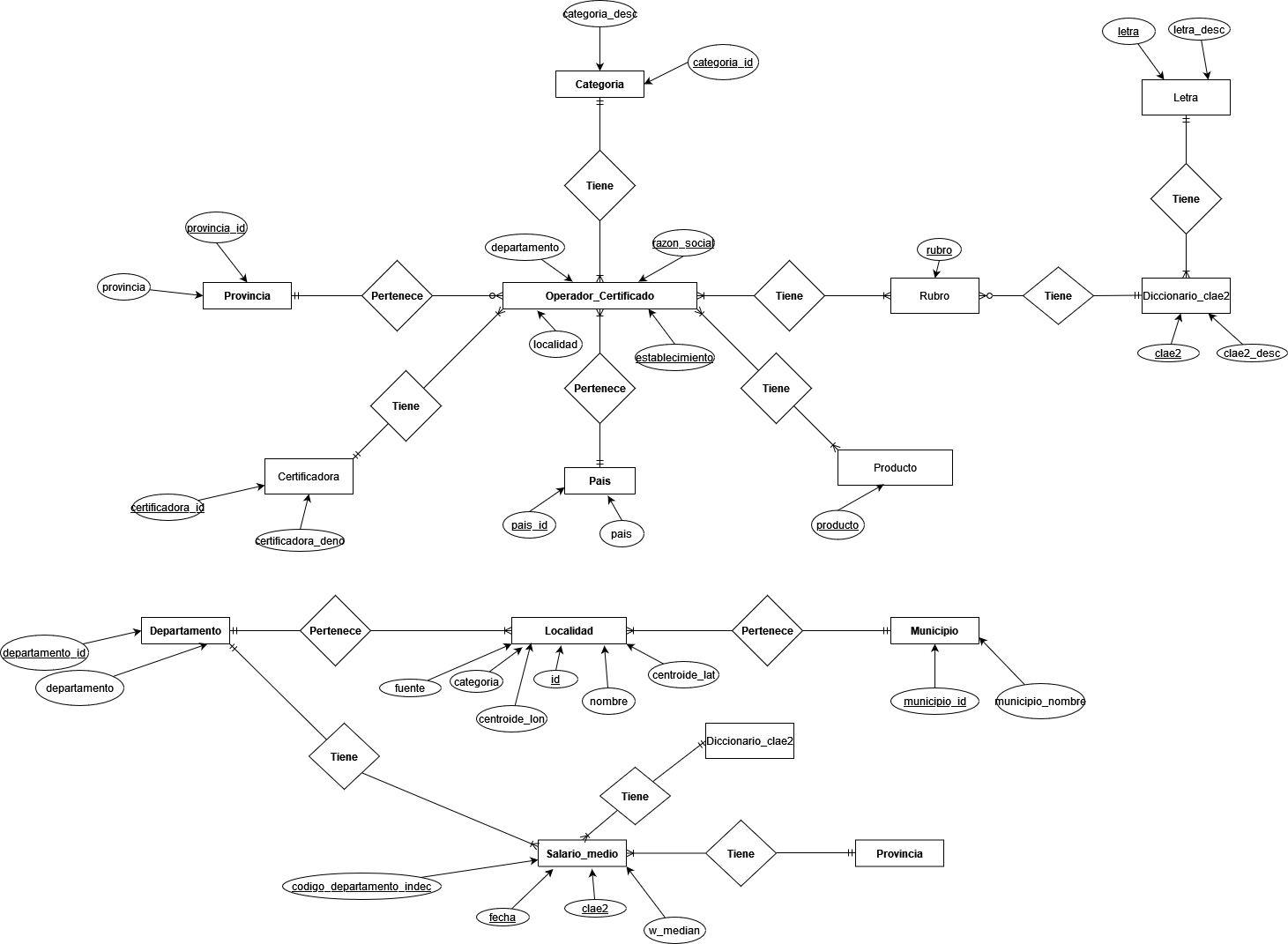
Dicho esto, para que los esquemas se encuentren en 3FN, decidimos separar las relaciones transitivas del esquema original en otro nuevo. De esta manera, teníamos a los esquemas originales en 3FN con su respectivo esquema relacionado que contenía la información de

la DF transitiva.

El ejemplo mostrado en la diapositiva 8 de la presentación “Dependencias funcionales” sirve para ilustrar la naturaleza de la descomposición en 3FN. En el apartado de las “Tablas creadas" puede verse cada tabla original en 3FN y sus tablas asociadas.

Respuesta e)

Aquí puede verse el DER, que representa un modelo conceptual de los datos



Respuesta f)

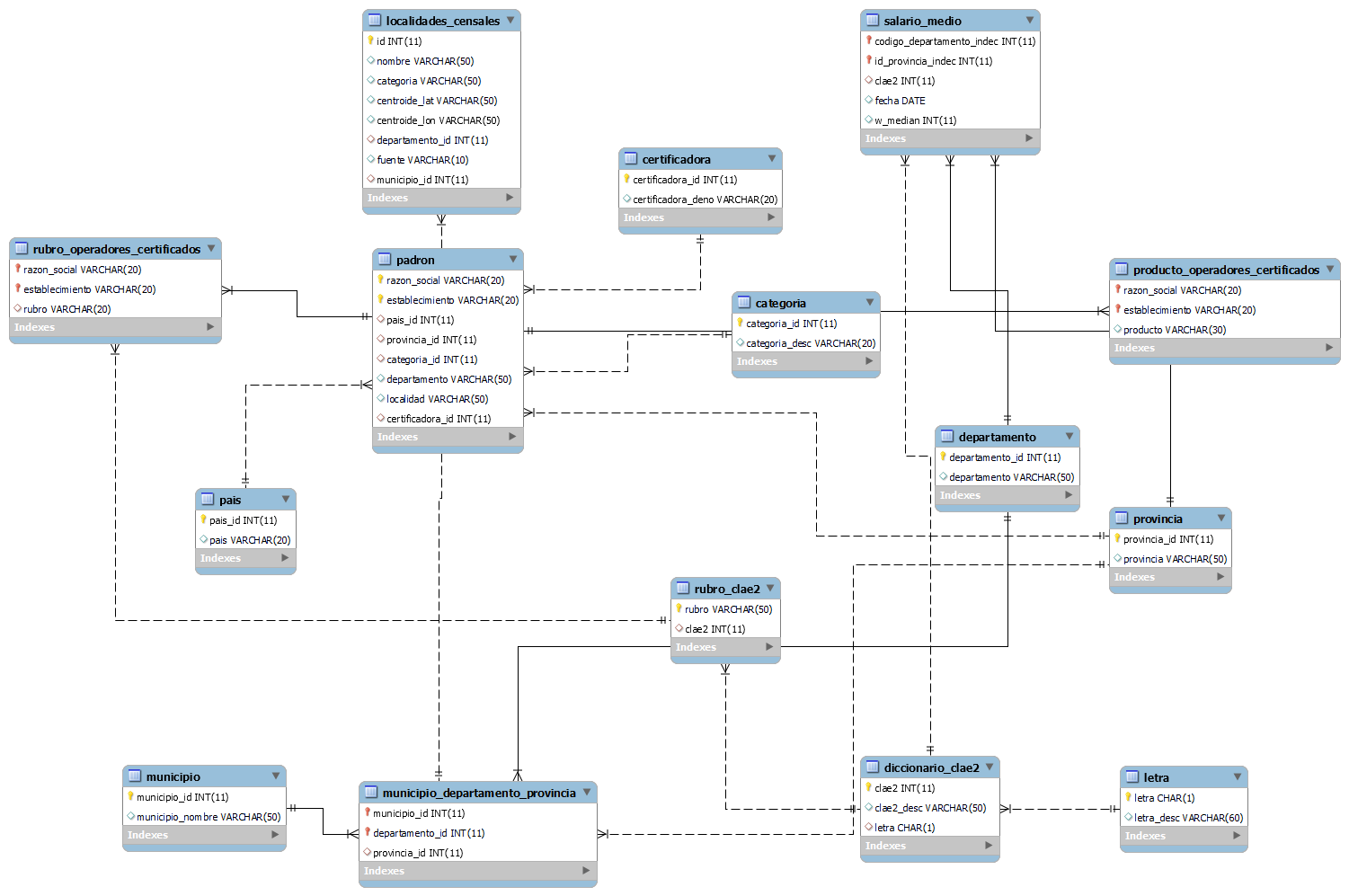
A partir de los esquemas normalizados en 3FN es posible construir un MER como se muestra en la figura (1).

Tabla padrón: Esta tabla surge de la fuente primaria “Padrón de Operadores Orgánicos Certificados”. La misma tiene como PK(primary key) la razón social y el establecimiento. A su vez tiene como FK(foreign key) pais\_id, provincia\_id, categoria\_id y certificadora\_id.

Tabla rubro\_operadores\_certificados: Esta tabla surge del proceso de normalización 1FN, ya que, en un principio el atributo rubro se encontraba en la tabla padrón con múltiples valores. Una de las formas de solucionar ese problema es crear una nueva tabla (rubro\_operadores\_certificados) la cual tenga como PK al atributo rubro ya “atomizado” y a la PK de la tabla padrón (razon\_social,establecimiento).

Tabla producto\_operadores\_cerficados: Idem que la tabla rubro\_operadores\_certificados.

Tabla localidades\_censales: Esta tabla surge de la fuente secundaria “Listado de las localidades censales según la base de datos censales del INSTITUTO NACIONAL DE ESTADÍSTICA Y CENSOS (INDEC)”. Tiene como PK al atributo id y a partir del proceso de normalización 3FN se establecen como FK al municipio\_id y departamento\_id, ya que municipio\_id + departamento\_id determinaban la provincia.

Tabla municipio\_departamento\_provincia: Esta tabla surge del proceso de normalización 3FN. La PK es municipio\_id + departamento\_id y tiene como atributo a la provincia\_id

Tabla municipio: Esta tabla surge del proceso de normalización 3FN, ya que,en un principio la tabla localidades\_censales tenía como atributos a (municipio\_id, municipio\_nombre) y municipio\_id determina municipio\_nombre, por lo cual, existe una dependencia transitiva. De esta forma queda como PK municipio\_id.

Tabla departamento: Idem que municipio.

Tabla salario\_medio: Esta tabla surge de la fuente primaria “Salarios del sector privado”. Tiene como PK a fecha+codigo\_departamento\_indec+clae2

Tabla diccionario\_clae2: Esta tabla surge de la fuente secundaria “Diccionario de clases”. Tiene como PK la clae2 y la FK es el atributo letra, la cual se estableció mediante el proceso de normalización 3FN.

Tabla rubro\_clae2: Esta tabla surge por el hecho de poder relacionar los rubros que se encuentran en la tabla padrón y los que se encuentran en la tabla diccionario\_clae2. La PK es el atributo rubro y la FK es el atributo clae2, ya que, se puede repetir por cada rubro.

g) En el archivo “normalización” estará el código para crear cada tabla y en la carpeta “ tablas\_creadas“ las respectivas tablas creadas.

Para separar las tablas en 3FN, el proceso fue bastante sencillo, ya que consistía simplemente en hacer una consulta SQL y seleccionar únicamente las columnas de interés.

Para separar los atributos en 1FN el proceso fue más complicado. La idea general fue generar una nueva tabla vacía conteniendo únicamente 3 columnas, 2 columnas correspondientes a la PK y la otra columna al atributo a normalizar. Luego la función “primera\_forma\_normal\_%% [[1]](#footnote-0)” toma esta lista vacía y separa la tupla de productos en productos atomicos .Cada producto atomico estará en una fila única acompañada de la PK.

h) En este apartado se debe describir la limpieza de datos realizada y luego identificar los problemas de calidad de las tablas originales. Vamos a comenzar explicando el proceso de limpieza de datos. Primero voy a hablar de los 3 cambios más importantes que tienen su respectiva justificación en la sección de “Decisiones tomadas”. Aquí vamos a hablar de cómo lo llevamos a cabo.

En primer lugar para eliminar las filas en las cuales contenían el valor “NC” en el atributo establecimiento en la tabla P, lo realizamos con una simple consulta SQL.[[2]](#footnote-1) Nuevamente para eliminar los salarios negativos en el atributo w\_median(salarios) de S, esto fue realizado con una simple consulta[[3]](#footnote-2). Luego el proceso para limpiar los departamentos en el atributo “departamentos” de P no fue nada sencillo. Primero se realizó una estandarización de las columnas a unir/joinear, es decir el atributo “departamento” de P y el atributo “departamento\_nombre” de LS. Esta estandarización consistía en poner todo en mayúsculas y sacar todos los tildes.[[4]](#footnote-3) Luego se realizó una consulta [[5]](#footnote-4)para ver qué departamentos de P se encontraban en LC, es decir que departamentos eran verdaderamente departamentos. Aquí podríamos haber eliminado los elementos que contenían un departamento en P que no se encontraban en LC. Sin embargo para eliminar la menor cantidad de datos posibles, decidimos hacer una consulta para ver los departamentos en P que eran municipios(osea pertenecían al atributo “municipio\_nombre”[[6]](#footnote-5) en LC.A partir de esto nos dimos cuenta que varios departamentos en P que eran en el fondo municipios, podían ser reemplazados por su departamento correspondiente. A partir de una función[[7]](#footnote-6) reemplazamos los departamentos que eran municipios, por su respectivo departamento. Resulta importante aclarar que este reemplazo se hizo contemplando que la provincia del “departamento” a cambiar sea la misma que el municipio, porque se quieren evitar reemplazos inválidos.

De esta manera los datos eliminados [[8]](#footnote-7)bajaron drásticamente( de n a m) que corresponden a departamentos en P que no eran ni departamentos ni municipios.

Luego hicimos una serie de limpiezas menores. En la tabla LC cambiamos el código de Ushuaia de 94105 a 94104 para que sea consistente con la tabla DD. Luego a partir de generar las tablas en 3FN, nos dimos cuenta que el atributo “provincia\_nombre” de la tabla LC tenía inconsistencias con el atributo “nombre\_provincia\_indec” de la tabla DD. Por esta razón reemplazamos en el atributo “provincia\_nombre” de la tabla LC “Ciudad Autónoma de Buenos Aires” por “CABA” y luego “Tierra del Fuego, Antártida e Islas del Atlántico Sur” por “Tierra del Fuego”

Las tablas contenían una gran cantidad de errores de calidad. Reiteradamente las tablas presentaban valores Nan o vacíos en sus columnas. Este error de calidad se atribuye a un error de modelo, ya que hay datos faltantes. Nosotros hemos realizado una consulta GQM para ver cuantos valores NC habían en el atributo “establecimiento” en la tabla P.

G: Averiguar cuántos valores NC hay en el atributo “establecimiento”

Q: ¿Cuántos valores NC hay en el atributo establecimiento?

M:”””

SELECT count(\*)

FROM padrón

WHERE establecimiento != ‘NC’

“””

Luego puede identificarse cierta falta de consistencia entre las distintas tablas. A pesar de que estamos describiendo un error de calidad que se atribuye al joinear varias tablas y no de una tabla en particular, algunas tablas deberían haber sido diseñadas de manera consistente para que con un simple JOIN se unieran de manera perfecta. Por ejemplo, los departamentos de la tabla P y los nombres de los departamentos de la tabla LC son poco consistentes, donde por un lado los departamentos de de la tabla P están en mayúscula y sin tildes y en la tabla LC los departamentos están en minúscula y con tildes. Esta falta de consistencia corresponde con un error de constancia.

Sección de Análisis de datos

TODOS LOS EJERCICIOS ESTAN EN EL ARCHIVO “ejercicios.py”

i) ¿Existen provincias que no presentan Operadores Orgánicos Certificados?

¿En caso de que sí, cuántas y cuáles son?

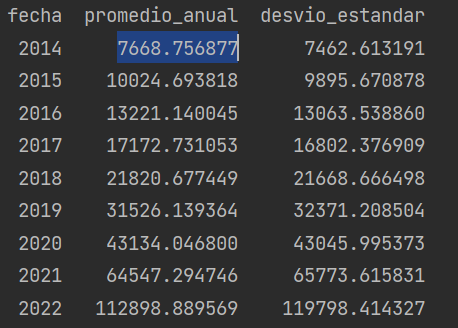
Respuesta i) A partir de una consulta SQL llegamos a que todas las provincias tienen al menos un Operador Orgánico Certificado, por lo cual no existen provincias que no presenten Operadores.

Respuesta ii) La respuesta es sí y son 430 departamentos.Para ver cuales son, correr el código “ej3” con su respectiva consulta

Respuesta iii) La actividad que más operadores tiene es Agricultura, ganadería, caza y servicios relacionados.

Respuesta iv) El salario promedio de Agricultura, ganadería, caza y servicios relacionados es de 195991 pesos argentinos

Respuesta v) Años con el salario promedio y su desvío estándar a nivel país.

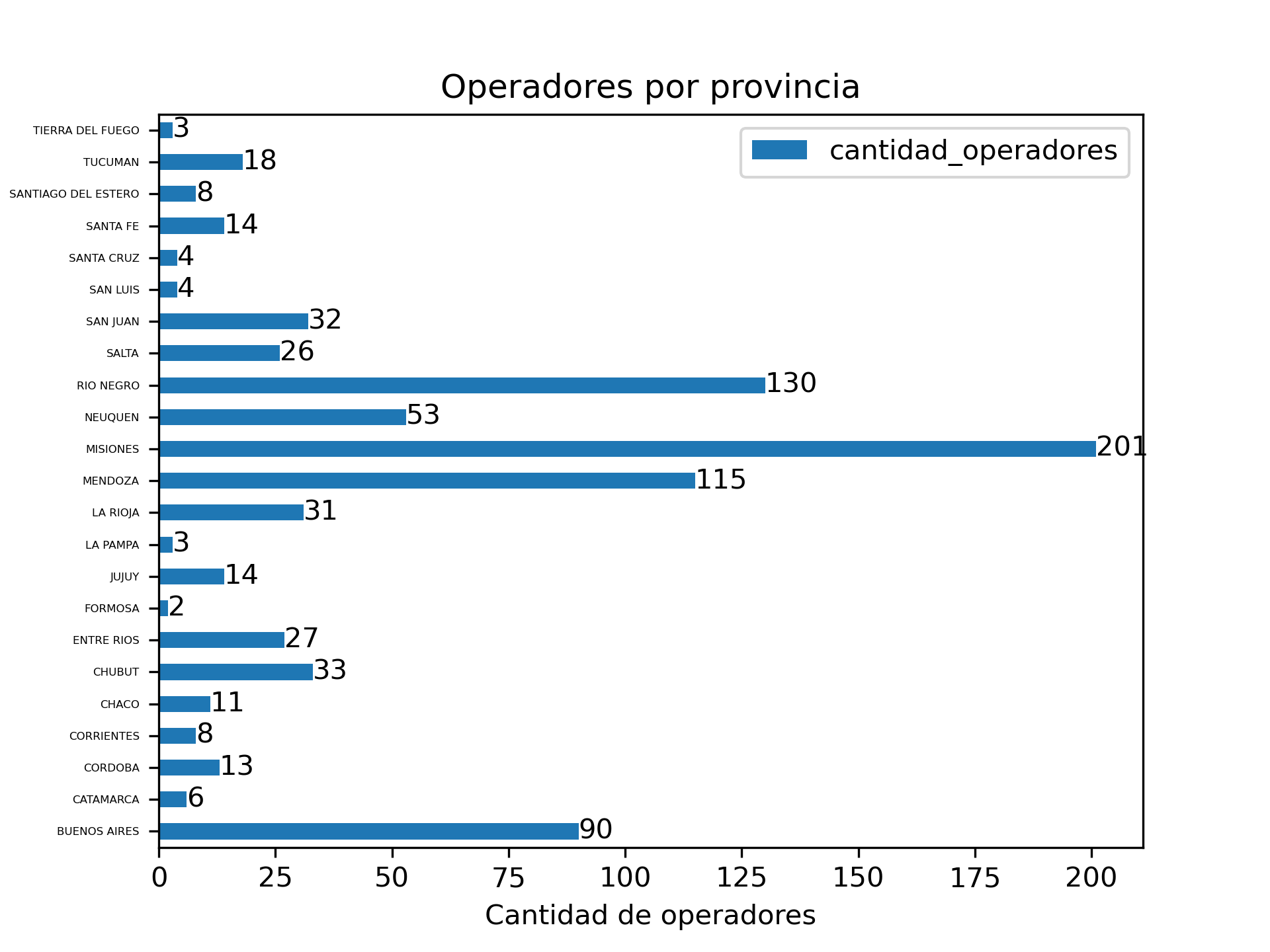


El provincial son muchas filas y no entra en una sola imagen. Puede verse al correr la función “ej4” con su respectiva consulta

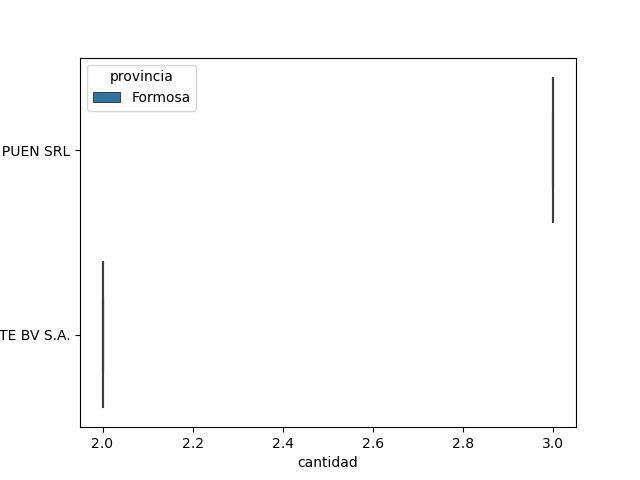
Los datos se podrían comparar a lo largo de los años a través de algún índice que sea teóricamente proporcional a los sueldos. Se nos ocurre agregar un indicador que hable sobre la canasta básica. De esta manera, entre el promedio del sueldo y el valor de la canasta básica tenemos una medida relativa para comparar entre los distintos años. Para eso necesitamos una fuente de datos secundaria que muestre el valor de la canasta básica a lo largo de los años.

j)

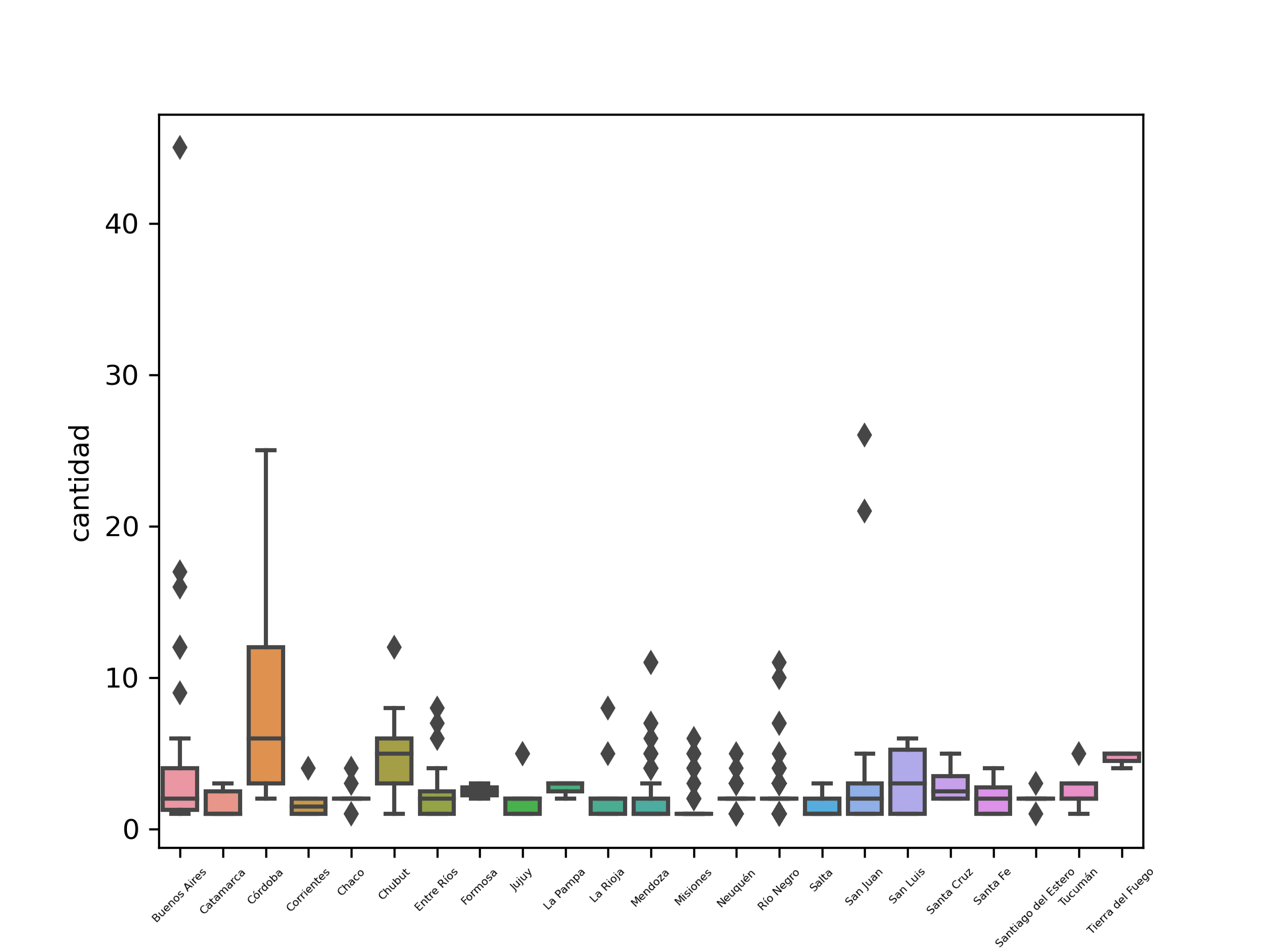
i)



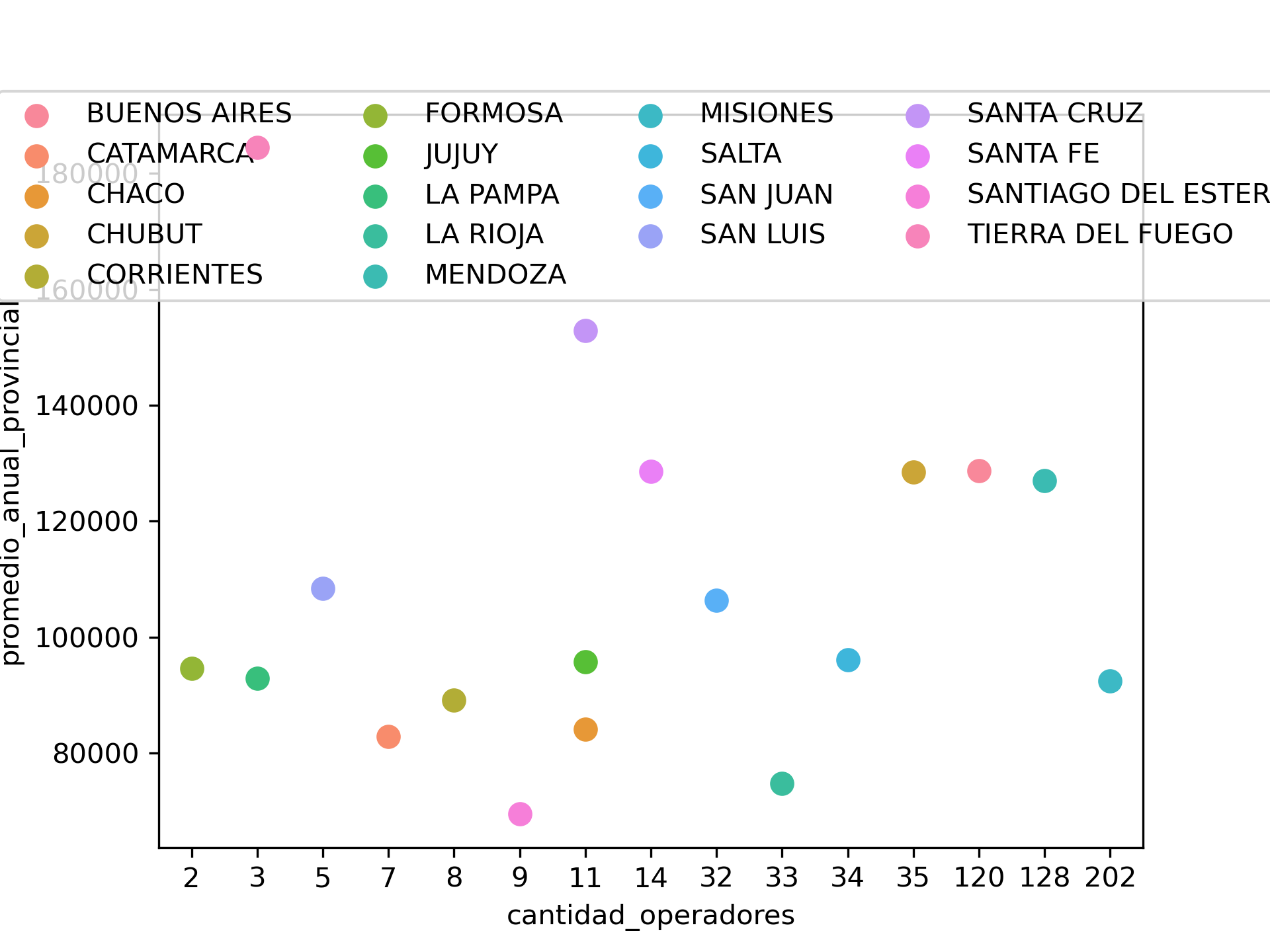
Respuesta ii) Siguiendo la consigna planteada, buscamos hacer un gráfico en base a cada una de las provincias, con cantidad de productos como variable del eje x y los operadores como variable del eje y. Sin embargo, al realizarlo, nos dimos cuenta que la información que aportaba no resultaba interesante ni era descriptiva de lo que quería mostrar, lo cual nos llevó a pensar que quizás un boxplot no era el gráfico más apropiado. Ejemplificamos con el caso particular de Formosa:



Para obtener mejor información, lo que hicimos fue graficar un boxplot (apegándonos a la consigna) conteniendo la cantidad de productos por provincia, lo cual resultó mucho más declarativo y nos aportó información más clara.



Respuesta iii)

En el eje de las x mostramos la cantidad de operadores, mientras que en el y, el salario promedio en dicha provincia (para la actividad) en el año 2022.

Respuesta iv)

En el eje de las x mostramos los salarios promedios, mientras que en el eje y mostramos las provincias. Entendemos que la poca declaratividad del gráfico viene dada por la cantidad de provincias representadas, lo cual hace que la información se pise y se vuelva muy poco declarativo.

Sección de Conclusiones

Finalmente debemos mostrar si existe alguna relación entre el desarrollo

de la actividad y el salario promedio que perciben los trabajadores del sector privado en

cada departamento de las provincias argentinas. Antes de dar una respuesta concreta, debemos considerar varios factores. Primero debemos considerar que nuestra limpieza no ha sido la mejor. Eliminar las filas de la tabla P, porque tenían valores “NC” en un atributo con poca importancia fue un grave error y un limitante en nuestro análisis, ya que no solo eliminamos los padrones de CABA y Tierra del Fuego, sino que eliminamos 400 datos.Además debido a nuestra falta de experiencia es posible que algunas tablas no “matcheen” de forma correcta al no limpiar o estandarizar esos datos. Particularmente al limpiar el atributo “departamento” en la tabla P, es muy posible que con más experiencia y herramientas muchos más datos podrían haber sido salvados.

Por todas estas razones consideramos que debido a nuestra inexperta limpieza de datos, no es posible llegar a ninguna conclusión categórica acerca de una posible relación entre el desarrollo de la actividad y el salario promedio que perciben los trabajadores del sector privado en cada departamento de las provincias argentinas.

Igualmente nos gustaría agregar que en la tabla III puede notarse que la relación entre cantidad de emprendimientos certificados y el salario promedio NO es lineal, por lo que este podría ser un argumento para negar la existencia de la relación planteada.

1. en archivo normalizacion.py, primera\_forma\_normal\_productos y primera\_forma\_normal\_rubro [↑](#footnote-ref-0)
2. en archivo limpieza.py, consulta\_sacar\_NC [↑](#footnote-ref-1)
3. en archivo limpieza.py, consukta\_sacar\_99 [↑](#footnote-ref-2)
4. en archivo limpieza.py, sección de estandarización [↑](#footnote-ref-3)
5. en archivo limpieza.py, consulta\_deptos [↑](#footnote-ref-4)
6. en archivo limpieza.py, consulta\_deptos\_que\_son\_muni [↑](#footnote-ref-5)
7. en archivo limpieza.py, cambiar\_valores\_a\_partir\_de\_tabla3 [↑](#footnote-ref-6)
8. en archivo limpieza.py, consulta eliminación [↑](#footnote-ref-7)