TRABAJO PARA MACRO-ECONOMÍA AVANZADA

Álvaro Revuelta Martínez

Curso 2018/2019

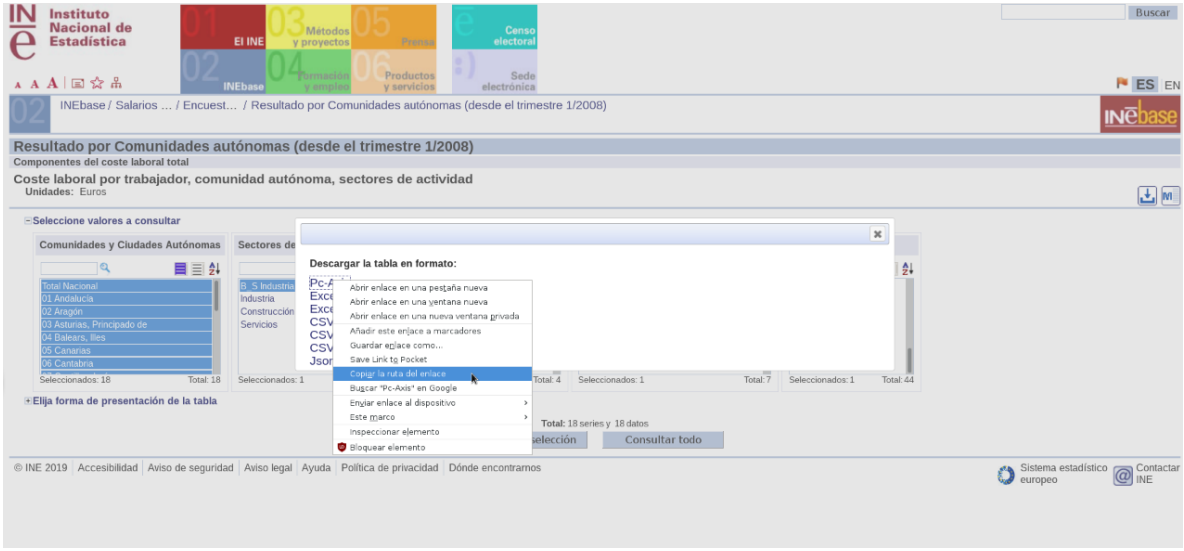
ÍNDICE:

A) Protocolo de obtención de los datos.

- Texto o código R que replique el proceso de principio a fin.

Accedemos la página web de cada uno de los datos que se desean obtener (por ejemplo www.ine.es/jaxiT3/Tabla.htm?t=6061)

Una vez dentro, se pulsa sobre el segundo botón de arriba a la derecha para acceder a la descarga de los datos, elegimos el formato PC-Axis y se copia la ruta del enlace(click derecho, copiar ruta del enlace).



Para la obtención de los datos dentro de R hago uso de la función *read.px*, que forma parte de la librería *pxR*, de la siguiente manera:

```
#Coste laboral por trabajador, comunidad aut?noma, sectores de actividad
data_1 <- read.px("http://www.ine.es/jaxiT3/files/t/es/px/6061.px?nocab=1", encoding = "latin1") %>% as.data.frame()


#Coste laboral por hora efectiva, comunidad aut?noma, sectores de actividad
data_2 <- read.px("http://www.ine.es/jaxiT3/files/t/es/px/6062.px?nocab=1", encoding = "latin1") %>% as.data.frame()

#Número de vacantes comunidad aut?noma
data_3 <- read.px("http://www.ine.es/jaxiT3/files/t/es/px/6064.px?nocab=1", encoding = "latin1") %>% as.data.frame()

#Motivos por los que no existen vacantes por comunidad aut?noma
data_4 <- read.px("http://www.ine.es/jaxiT3/files/t/es/px/6066.px?nocab=1", encoding = "latin1") %>% as.data.frame()
```

Lo que guarda desde *data_1* hasta *data_4* y como objetos *data.frame* los datos del Coste Laboral que se desean analizar.

- Extracción de datos obtenida, en formato .csv y .RData

Para descargar los datos tanto en csv como RData, transformamos los *data.frame* en dichos formatos haciendo uso de las funciones *write.csv*(incluida por defecto en R) y *write_rds*(libreria *readr*) .

Para crear homogeneidad en la presentación de todo el trabajo, he incluido un botón de descarga en la aplicación Shiny que se detallará más adelante, el código de descarga quedaría de la siguiente manera:

```
output$downloadCSV_1 <- downloadHandler(
  filename = "coste_laboral_por_trabajador_comunidad_autonoma_sectores_de_actividad.csv",
  content = function(file) {
    write.csv(data_1, file, row.names = FALSE)
  }
)
output$downloadRdata_1 <- downloadHandler(
  filename = "coste_laboral_por_trabajador_comunidad_autonoma_sectores_de_actividad.rds",
  content = function(file) {
    write_rds(data_1,file)
  }
)
```

Replicado para las 4 variables analizadas.

Dicho botón se crearía en la parte de la UI(interfaz que el usuario ve) de la siguiente manera:

```
downloadButton('downloadCSV_1', 'Descarga en CSV'),
downloadButton("downloadRdata_1", 'Descarga en Rdata')
```

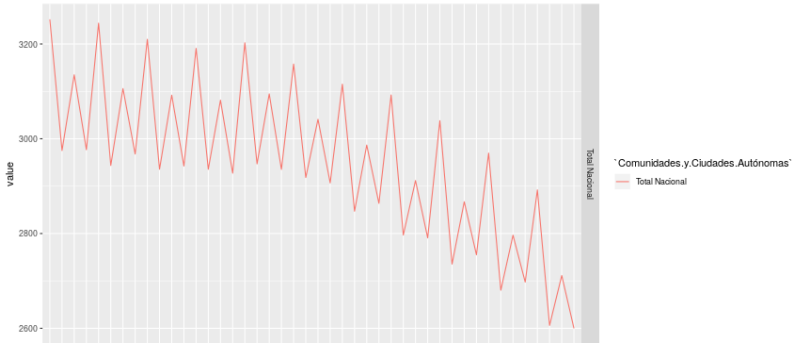
Y sería accesible desde la aplicación:

B) Informe Final.

- <u>Breve estudio de los datos, valiéndose de al menos un gráfico por cada variable.</u>

A) <u>Coste laboral por trabajador, comunidad autónoma, sectores de actividad</u>

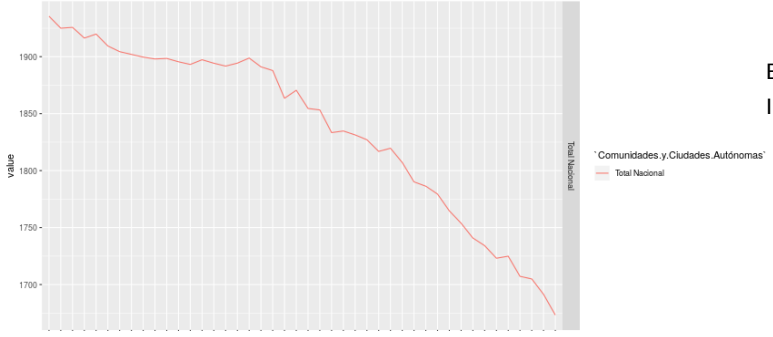
En esta variable (basándonos en la clasificación CNAE de sectores de actividad), se puede observar como el *coste laboral total* para el sector *industria* ha crecido desde el 2008, aunque de manera muy irregular ya que se observan subidas y bajadas del coste muy grandes de un año hacia otro.



Evolución del coste laboral total en Industria desde el 2018 hasta el 2008

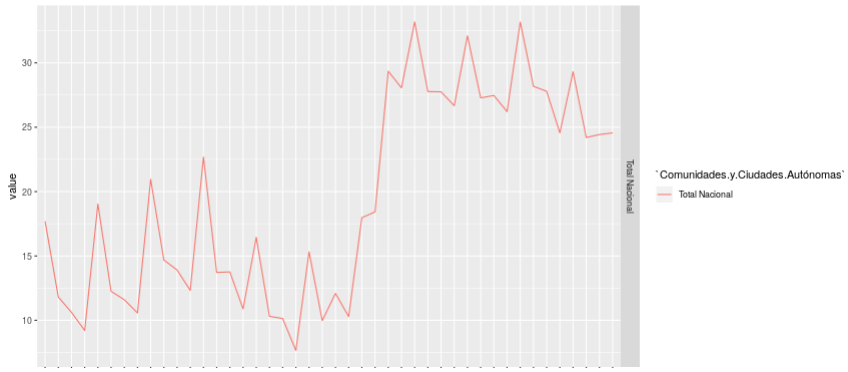
Grosso modo, se puede afirmar que un trabajador de dicho sector hoy cuesta más que hace una década, pero esta información por si sola no nos vale nada ya que habría que examinar también la productividad por trabajador.

Si desglosamos dicho coste en sus componentes podemos observar como el coste salarial ordinario ha aumentado de manera ininterrumpida desde el mismo periodo, lo que podemos asociar a un incremento de los sueldos.



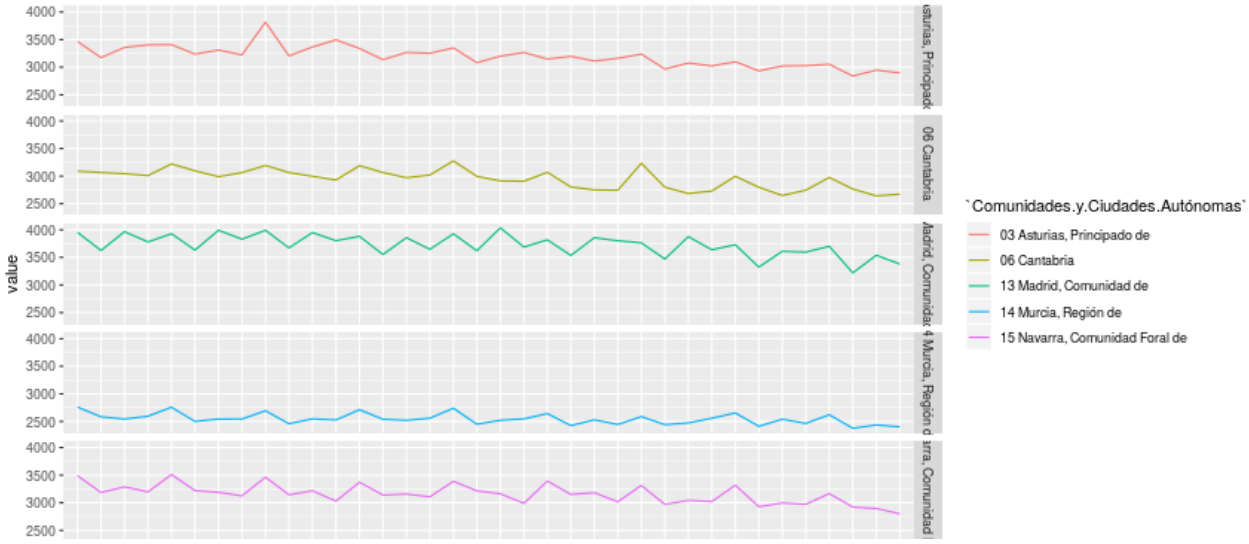
Evolución del coste salarial ordinario en Industria desde el 2018 hasta el 2008

Sin embargo, si observamos las Subvenciones y bonificaciones de la Seguridad social, dicho coste se ha reducido considerablemente desde el 2018, con una bajada muy elevada en el año 2012/2013 coincidiendo con la entrada de un nuevo gobierno en España y los años más duros de la crisis.



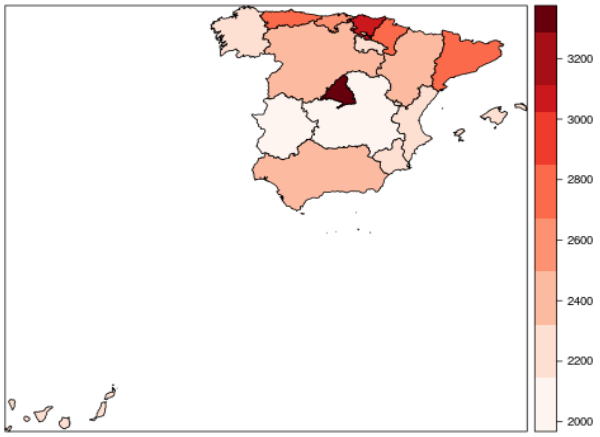
Evolución de las subvenciones y bonificaciones de la S.Social en Industria desde el 2018 hasta el 2008

Si observamos la evolución de esta variable en diferentes comunidades autónomas sigue el mismo patrón(o uno parecido) que el de Total Nacional, un crecimiento en los últimos 10 años, en algunos lugares de manera más pronunciada que en otros.

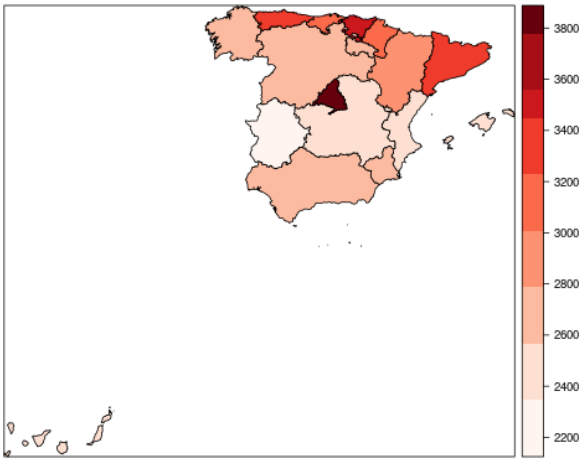


Evolución del Coste laboral total en Industria desde el 2018 hasta el 2008

Para poder analizar los datos de varias comunidades con más comodidad pasaremos a usar un mapa de calor, los datos siguen siendo los mismos pero con otra forma de representarlos.



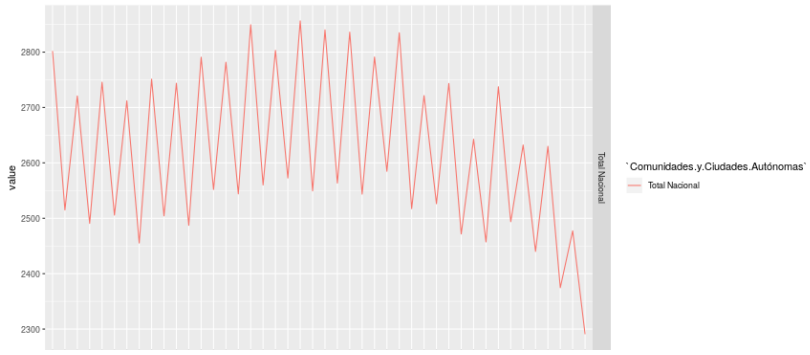
Coste laboral total en Industria en el 2008



Coste laboral total en Industria en el 2018

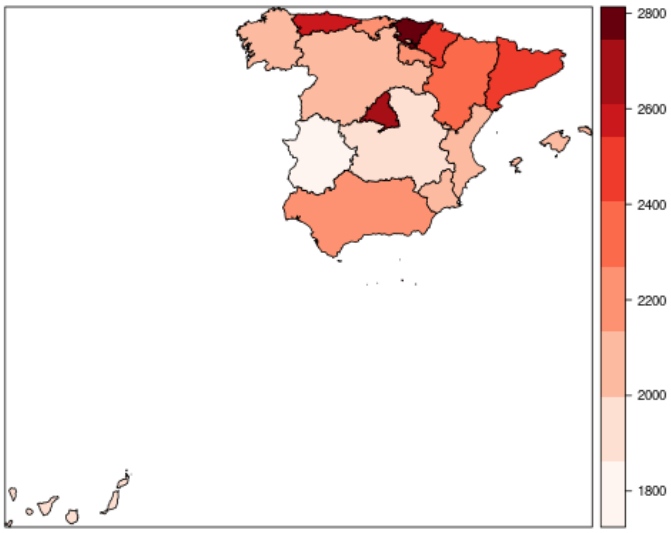
Esta representación confirma la afirmación anterior, y es el crecimiento del coste laboral por trabajador desde el 2008 en todas las comunidades, además se puede apreciar como aquellas comunidades más industrializadas tienen los mayores costes (Madrid, Barcelona, País vasco).

Pasando ahora al sector de la construcción, podemos ver una evolución más irregular, teniendo un ritmo creciente y luego decreciente.

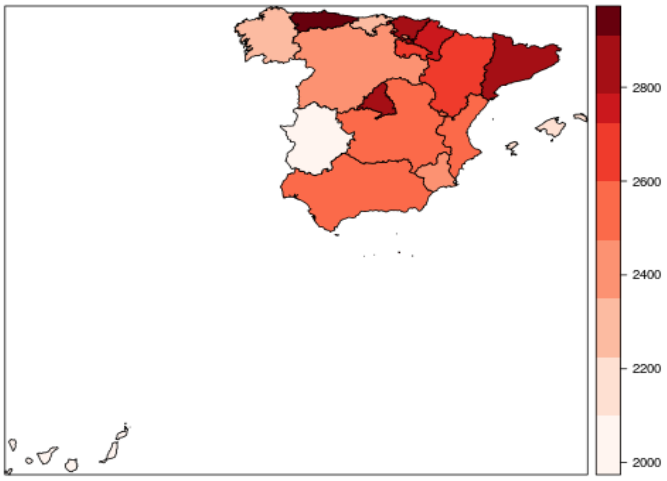


Evolución del coste laboral total en el sector Construcción desde el 2018 hasta el 2008

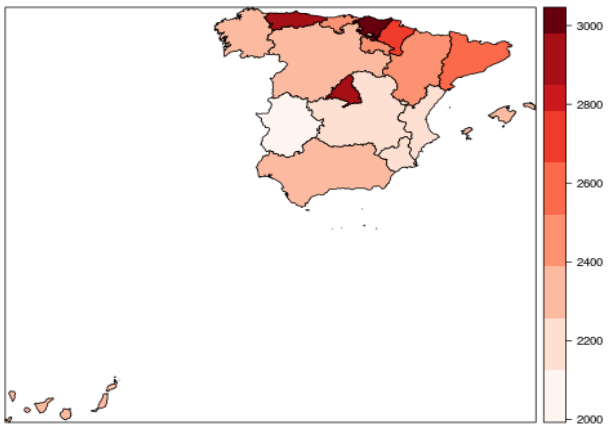
Dicha situación puede deberse a el comienzo de la crisis financiera del 2008, agudizada en España debida a la especulación inmobiliaria.



Coste laboral total en el sector construcción en el año 2008



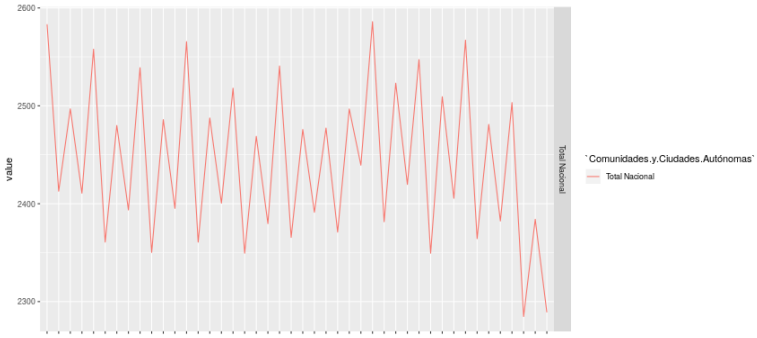
Coste laboral total en el sector construcción en el año 2012



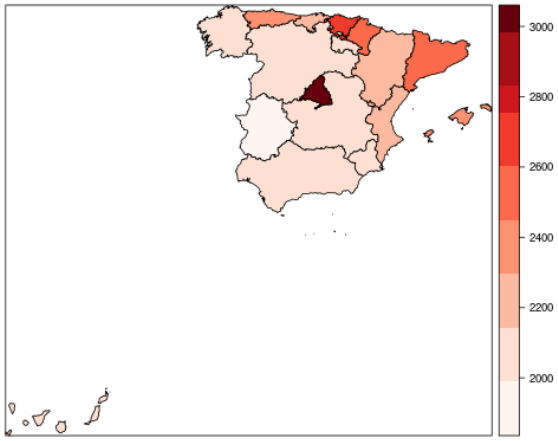
Coste laboral total en el sector construcción en el año 2018

Con los diferentes mapas de calor se puede ver como del 2008 al 2012 el coste crece de manera significativa, y del 2012 al 2018, no solo no crece, sino que además decrece en ciertos lugares mientras se mantiene más o menos constante en otros.

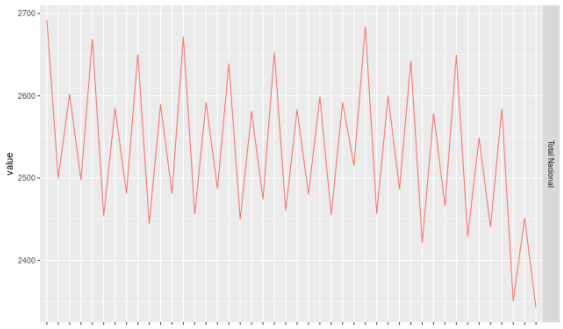
En cuanto a los dos sectores de esta variable que quedan por observar, el análisis sería muy parecido que el del sector industria.



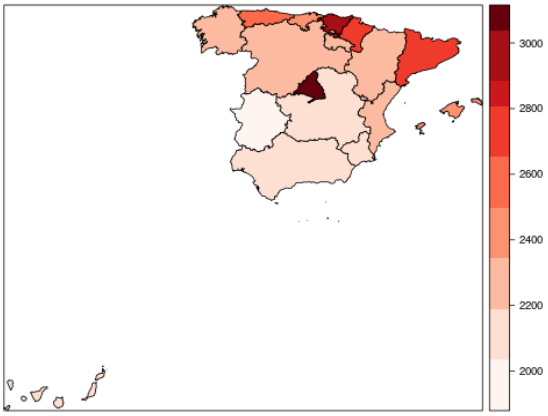
Evolución del coste laboral total en el sector Servicios desde el 2018 hasta el 2008



Coste laboral total en el sector servicios en el año 2018



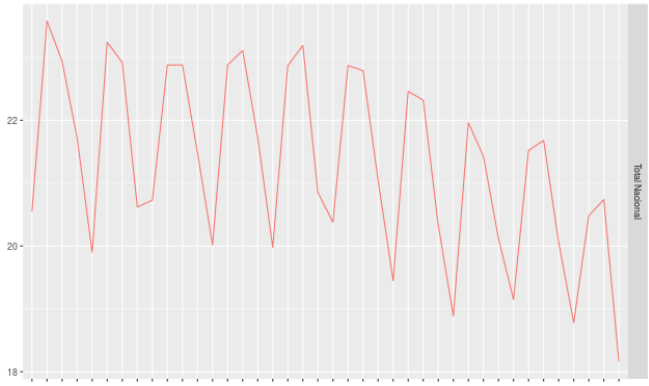
Evolución del coste laboral total en el sector B_S Industria, construcción y servicios desde el 2018 hasta el 2008



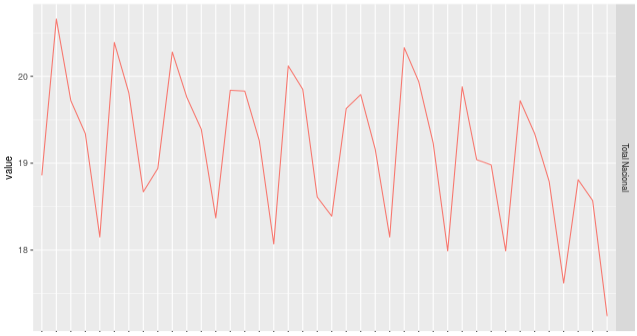
Coste laboral total en el sector _S Industria, construcción y servicios en el año 2018

B) <u>Coste laboral por hora efectiva, comunidad autónoma, sectores de actividad</u>

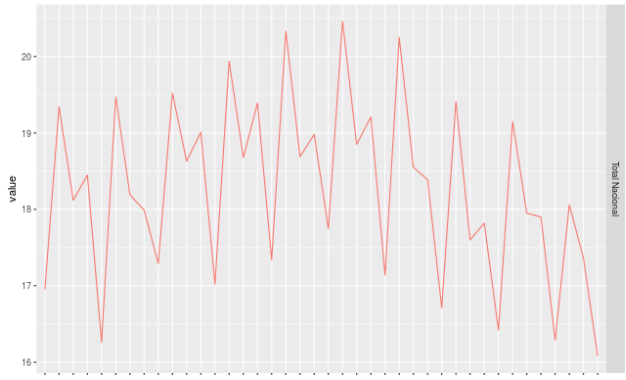
Ahora estamos analizando los mismos componentes y sectores, pero por hora efectiva en lugar de trabajador. En esta variable los resultados son prácticamente idénticos que el anterior, pudiéndose observar un crecimiento del coste desde el 2008 hasta el 2018 en t todos los sectores, además en el de la construcción la gráfica sigue el mismo patrón, dibujando una parábola, crece hasta los años 2012/2013 y luego decrece, aún así en el 2018 sigue siendo superior al año 2008.



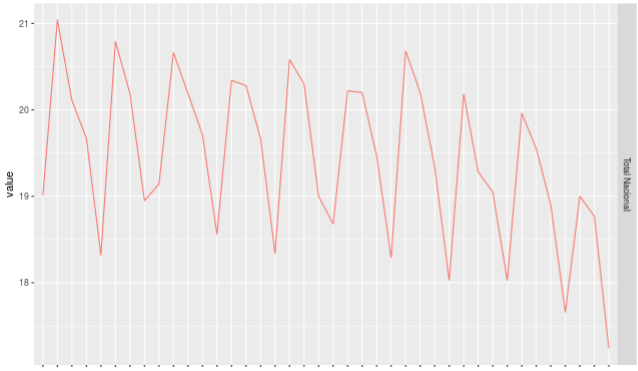
Evolución del coste laboral total en el sector Industria desde el 2018 hasta el 2008



Evolución del coste laboral total en el sector Servicios desde el 2018 hasta el 2008



Evolución del coste laboral total en el sector Construcción desde el 2018 hasta el 2008



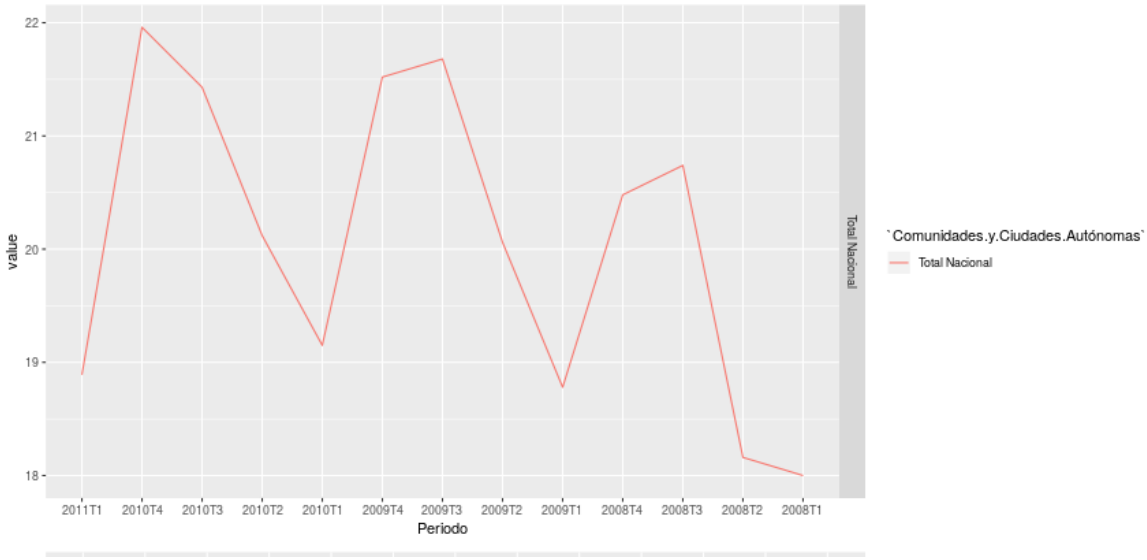
Evolución del coste laboral total en el sector B_S Industria, construcción y servicios desde el 2018 hasta el 2008

Además es importante destacar, que gracias a tener el periodo desglosado por Trimestres podemos observar que la evolución dentro de cada año es siempre igual.

Entre los Trimestres 1 al 3(es decir, los meses de Enero-Julio) el coste crece, con una crecida mucha más pronunciada entre los trimestres 2 y 3(Abril-Julio) .

Sin embargo, entre los Trimestres 3,4 y 1(Julio-Enero) el coste decrece, con una bajada más pronunciada entre los trimestres 4 y 1 (Octubre-Enero).

Lo que significa que en los meses de verano hasta navidad las horas de trabajo son más baratas.



Evolución del coste laboral total en el sector Industria desde el 2011 hasta el 2008

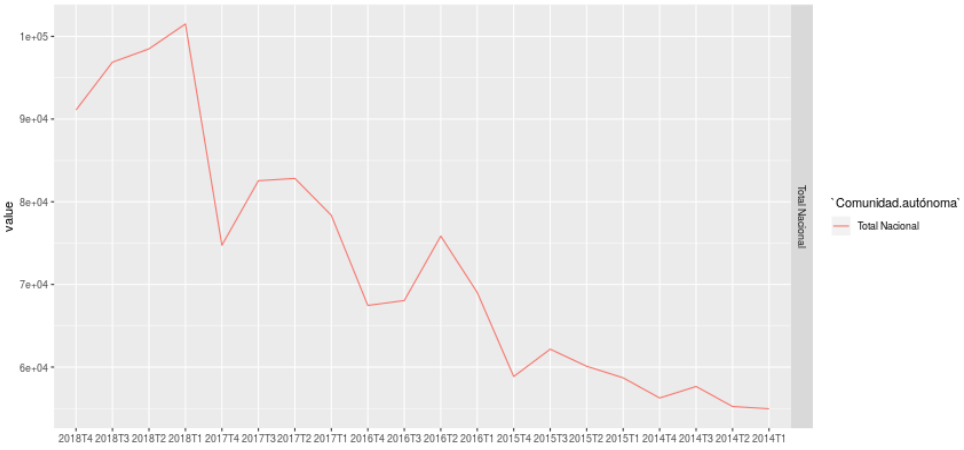
Usando este gráfico de ejemplo, podemos ver como del 2008T1 hasta el 2008T3 el coste aumenta, luego entre 2008T3 y 2008T4 se mantiene a un nivel muy constante pero aún así decrece un poco y luego entre 2008T4 y 2009T1 disminuye bastante más rápido. Una secuencia parecida se repite entre 2009T1,2010T1 y 2011T1.

C) <u>Número de vacantes comunidad autónoma</u>

Esta variable recoge la cantidad numérica de puestos de trabajo libres(vacantes) en cada comunidad autónoma.

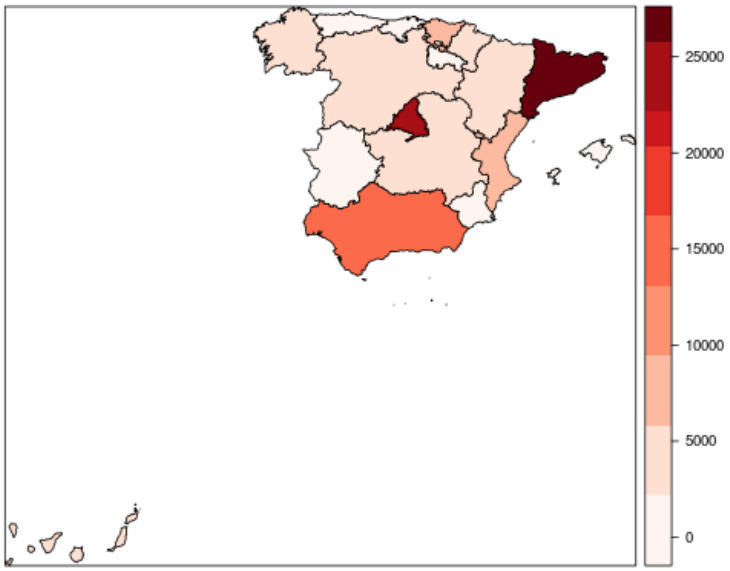
Estos datos, a diferencia de los dos anteriores relacionados con el coste laboral, comienzan en el año 2014 lo que imposibilita poder realizar un análisis sobre una evolución antes y después del comienzo de la crisis.

Podemos ver como desde el 2014 el número de vacantes ha aumentado, además se produce un aumento importante de las vacantes al cambiar de año. En contraposición a las gráficas de las anteriores variables que indicaban que el coste por trabajador y por hora era menor entre los trimestres 4 y 1(Octubre-Enero), pese a ello el número de vacantes aumenta. Es probable que este relacionado con el comienzo de un nuevo ciclo económico. Durante los trimestres 2 al 4(Abril-Octubre) el número de vacantes se reduce, probablemente por ser meses de verano en los cuales suele haber una mayor disponibilidad laboral.



Evolución del número de vacantes desde el 2018 hasta el 2014

Observado todas las comunidades nos damos que aquellas con mayor número de habitantes también tienen el mayor número de vacantes(lo cual resulta obvio dado que no estamos comparando los datos sobre el total de la población de cada zona)



Número de vacantes en el 2018

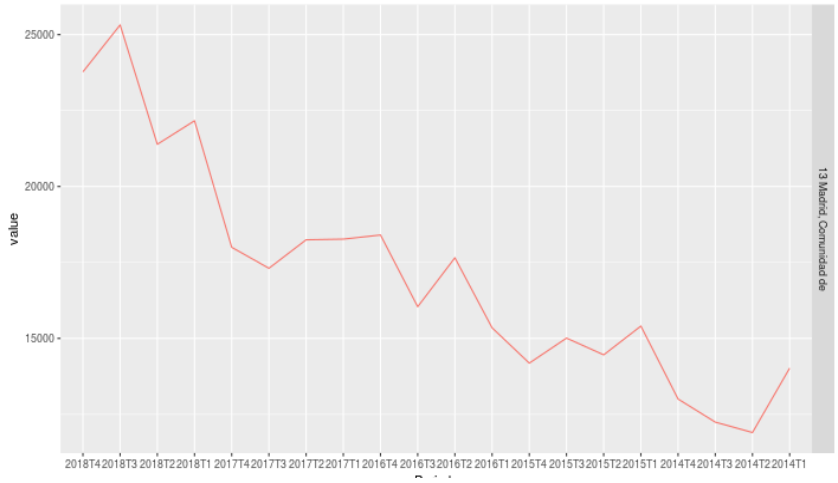
Resultaría interesante, entonces, coger dos Comunidades una con un número elevado de vacantes(y población), por ejemplo, Madrid, y otra con un número reducido de vacantes(y población), por ejemplo, Murcia, y ver si siguen la misma evolución que el Total Nacional.

En el caso de Madrid la evolución es muy similar a la ya vista, con crecimientos entre los trimestres 4 y 1(sobretodo a partir de 2016) y decrecimientos entre los 2 y 4.

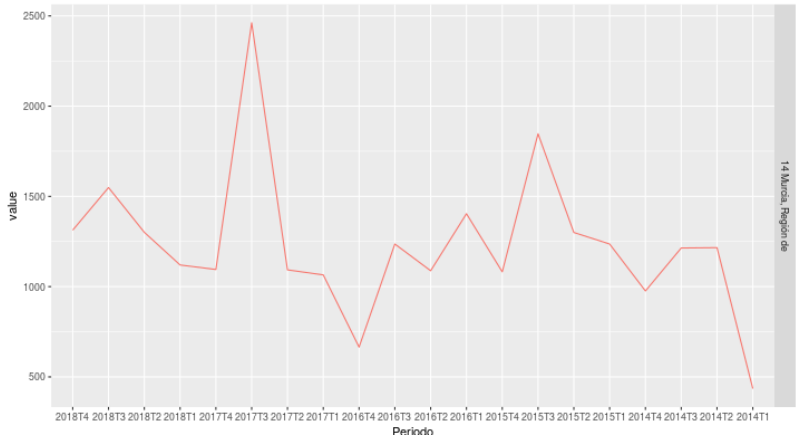
En Murcia, sin embargo, sigue una evolución muy irregular con picos en los terceros trimestres de cada año y volviendo a niveles muy parecidos a los de antes después.

Esta circunstancia puede ser debido a que se coincida con la temporada de algún producto típico de la zona.

Extrapolando, podemos decir que en las grandes ciudades sigue un comportamiento parecido y en las zonas con menos habitantes es mucho más dependiente de otros factores externos.



Evolución del número de vacantes desde el 2018 hasta el 2014 en Madrid

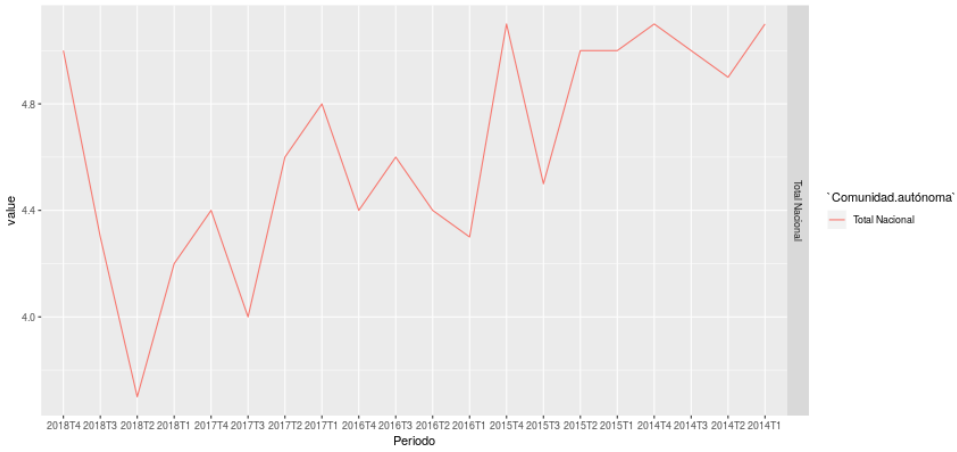


Evolución del número de vacantes desde el 2018 hasta el 2014 en Murcia

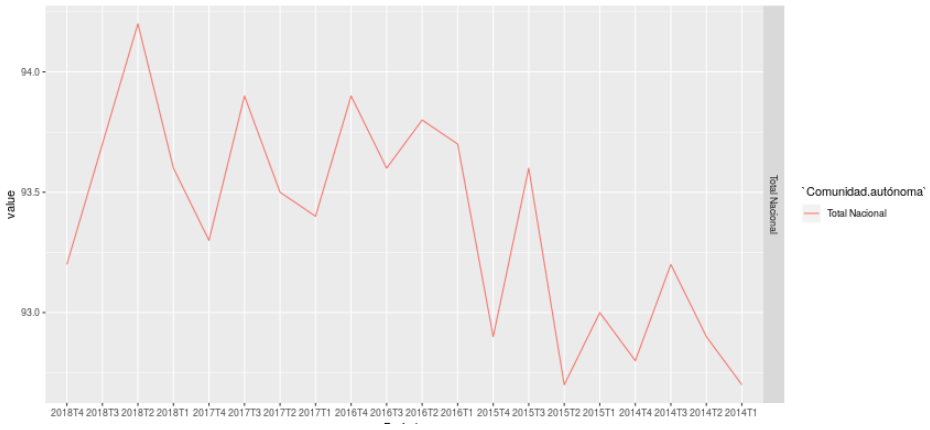
D) <u>Motivos por los que no existen vacantes por comunidad autónoma</u>

Esta variable complementa a la anterior dado que podemos ver los motivos por los que no existen vacantes en las diferentes comunidades autónomas. Estos motivos pueden ser, fundamentalmente, “No se necesita ningún trabajador” o “elevado coste de contratación”.

Los valores están en porcentaje y podemos ver como la gran mayoría se concentran en “No se necesita ningún trabajador”, entre el 93% y el 95%, dejando a “Elevado coste de contratación” alrededor de un 5% y el resto en “otros motivos”.



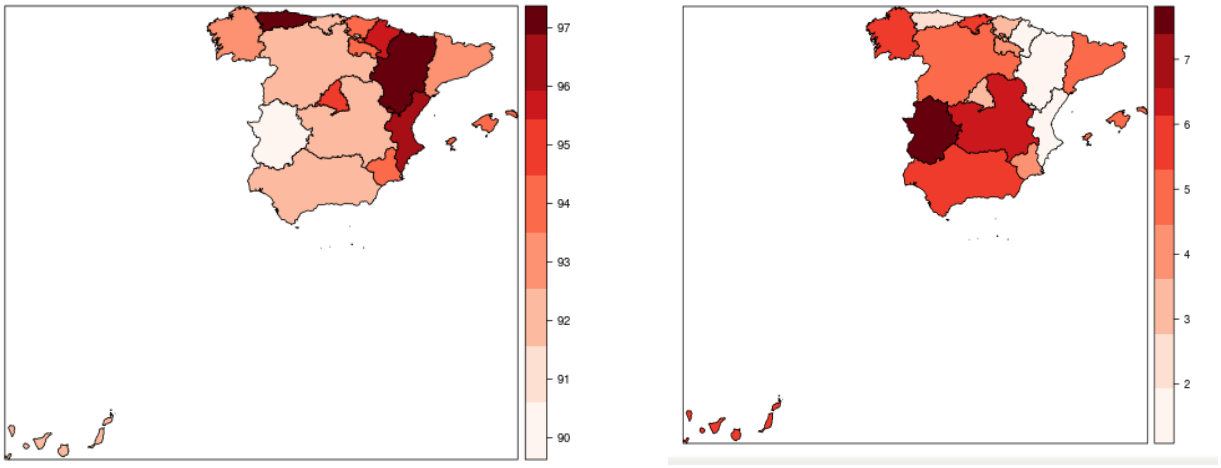
Evolución del motivo “Elevado coste de contratación” como motivo de no vacantes, desde el 2018 hasta el 2014



Evolución del motivo “No se necesita ningún trabajador” como motivo de no vacantes, desde el 2018 hasta el 2014

Comparando las diferentes comunidades autónomas, podemos darnos cuenta que en comunidades como Extremadura, Andalucía y Castilla la Mancha son las primeras en “elevado coste de contratación” y Aragón, Valencia y Asturias en “No se necesita ningún trabajador”.

Sin embargo, como hemos visto más arriba, Extremadura, Andalucía o Castilla La Mancha no son zonas que tengan los mayores costes laborales por trabajador u hora, así que ese motivo de falta de vacantes tiene que estar ocasionado por otro factor que no se ha analizado.



Motivo “no se necesita ningún trabajador”(izda) comparado con “elevado coste de contratación”(dcha) en 2018

- Diseño de un tipo de gráfico original para esa variable/grupo de variables en Shiny -o similar- publicable online.

Para la creación de los gráficos primero debemos filtrar los datos obtenidos con base en lo que el usuario quiera visualizar.

Por tanto dividimos la tarea en la parte que realiza la función *UI* y la función *Server* de Shiny.

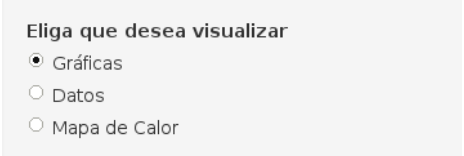
- **UI**

Dicha parte se encarga de mostrar los elementos con los que el usuario interacciona(Una aplicación Shiny está construida de manera reactiva, lo que quiere decir que se actualiza de manera dinámica cuando seleccionados unas opciones distintas).

Shiny permite, además, crear Widgets para realizar esa interacción

Antes de nada, crearemos un Widget para elegir si queremos visualizar la gráfica, los datos en bruto o un mapa de calor.

```
radioButtons(
  'elecc_1', label='Eliga que desea visualizar', choices = c("Gráficas"= "gr", "Datos"= "dt", "Mapa de Calor" = "mc"),
  selected="gr"
```

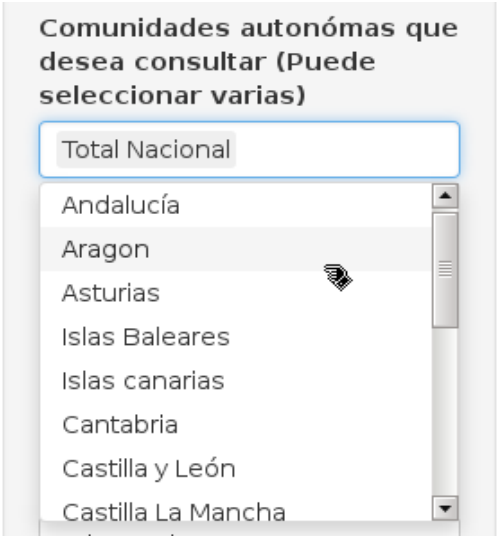


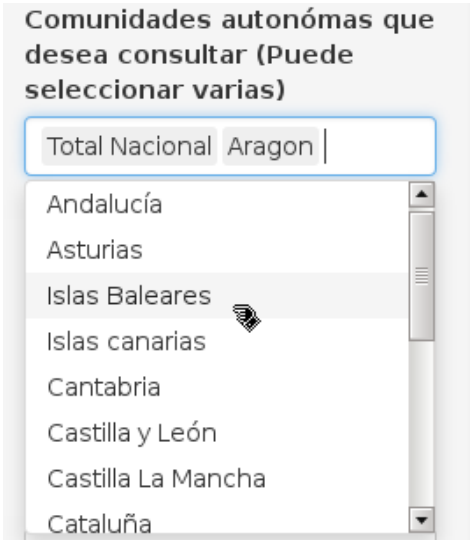
1. Comunidades Autónomas

Para la elección de las comunidades autónomas he usado la función *selectInput*, que permite elegir según una lista de opciones, activando la posibilidad de seleccionar más de una.

```
selectInput('Comunidades_1',
            label = 'Comunidades autonómas que desea consultar (Puede seleccionar varias)',
            choices = c("Total Nacional"="Total Nacional",
                        "Andalucía"="01 Andalucía",
                        "Aragon" = "02 Aragón",
                        "Asturias" = "03 Asturias, Principado de",
                        "Islas Baleares" = "04 Balears, Illes",
                        "Islas canarias" = "05 Canarias",
                        "Cantabria" = "06 Cantabria",
                        "Castilla y León" = "07 Castilla y León",
                        "Castilla La Mancha" = "08 Castilla - La Mancha",
                        "Cataluña" = "09 Cataluña",
                        "Valencia" = "10 Comunitat Valenciana",
                        "Extremadura" = "11 Extremadura",
                        "Galicia" = "12 Galicia",
                        "Madrid" = "13 Madrid, Comunidad de",
                        "Murcia" = "14 Murcia, Región de",
                        "Navarra" = "15 Navarra, Comunidad Foral de",
                        "País Vasco" = "16 País Vasco",
                        "La Rioja" = "17 Rioja, La"
            ),#choices
            selected = "Total Nacional",
            multiple = TRUE
),#COMUNIDADES_SelectInput
```

La visualización en la interfaz quedaría de la siguiente manera:





2. Periodo

Dado que mi muestra de datos estaba dividida en años y trimestres, el filtrado del periodo a visualizar se hará eligiendo primero el rango de años y luego los trimestres.

➔ Para elegir el rango de años:

La función *dataRangeInput* permite realizarlo ya que despliega un calendario, pero dicho calendario muestra también meses, semanas y días, los cuales me resultaban irrelevantes, así que realizo una función auxiliar que solamente muestre los años.

```
dateRangeInput2 <- function(inputId, label, minview = "years", maxview = "decades", ...) {
  d <- shiny::dateRangeInput(inputId, label, ...)
  d$children[[2L]]$children[[1]]$attribs[["data-date-min-view-mode"]] <- minview
  d$children[[2L]]$children[[3]]$attribs[["data-date-min-view-mode"]] <- minview
  d$children[[2L]]$children[[1]]$attribs[["data-date-max-view-mode"]] <- maxview
  d$children[[2L]]$children[[3]]$attribs[["data-date-max-view-mode"]] <- maxview
  d
}
```

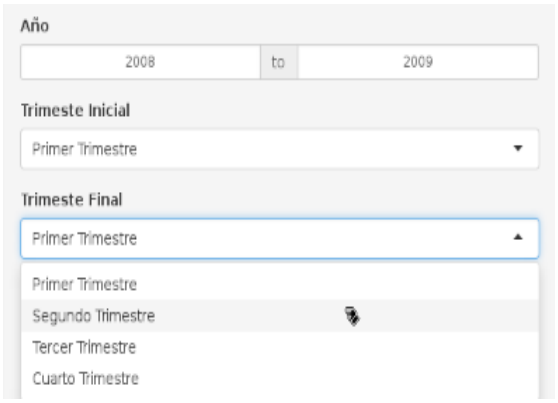
```
dateRangeInput2("año_1", "Año",
                startview = "year",
                min = "2008-01-01", max = "2018-01-01",
                minview = "years", maxview = "decades",
                format ="yyyy" , start = "2008-01-01",

                end ="2009-01-01"
),#AÑO_RangeInput
```

➔ Para elegir los trimestres:

La selección se realiza mediante la misma función que la de las Comunidades Autónomas, *selectInput*.

```
  selectInput('TC_1',
              label = 'Trimeste Inicial',
              choices = c("Primer Trimestre" ="T1",
                          "Segundo Trimestre" = "T2",
                          "Tercer Trimestre" ="T3",
                          "Cuarto Trimestre" = "T4"
              )#choices
  ),#TC_SelectInput
  selectInput('TF_1',
              label = 'Trimeste Final',
              choices = c("Primer Trimestre" ="T1",
                          "Segundo Trimestre" = "T2",
                          "Tercer Trimestre" ="T3",
                          "Cuarto Trimestre" = "T4"
              )#choices
  )#TF_SelectInput
```

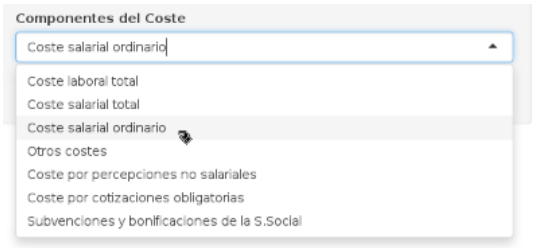
Mostrándose todo en la aplicación de esta manera:



3. <u>Sector, Componente del coste, Motivos de las vacantes</u>

Dependiendo de la variable se puede filtrar por otros criterios, como el sector, componente del coste…
Dicho filtrado se realiza con el *selectInput*.

```
selectInput('Componentes_1',
            label = 'Componentes del Coste',
            choices = c("Coste laboral total" ="Coste laboral total",
                        "Coste salarial total" = "Coste salarial total",
                        "Coste salarial ordinario" ="Coste salarial ordinario",
                        "Otros costes" = "Otros costes",
                        "Coste por percepciones no salariales" ="Coste por percepciones no salariales",
                        "Coste por cotizaciones obligatorias" = "Coste por cotizaciones obligatorias",
                        "Subvenciones y bonificaciones de la S.Social" = "Subvenciones y bonificaciones de la S.Social"
            )#choices
),#COMPONENTES_SelectInput
```



(Ejemplo para los componentes del coste, sería igual para los otros criterios).

Finalmente se mostrarían las gráficas y el mapa con la función *plotOutput*

```
plotOutput('grafico1_2'),
plotOutput('grafico2_2') ),
```

■ Server

En la parte de Server(Backend) se realizan todas las operaciones que el usuario no ve.

1. Creación del gráfico

   Para la creación del gráfico primero debemos filtrar basándose en los parámetros que el usuario ha seleccionado.

   Para ello he creado una función llamada *filtrar* que, valga la redundancia, filtra, pero de manera reactiva(se actualiza al actualizar los inputs).

```
Filtrar_1 <- reactive ({

  #busqueda contiene un dataFrame con todos los datos filtrados excepto el Periodo
  busqueda = filter(data_1, data_1$Comunidades.y.Ciudades.Autónomas %in% input$Comunidades_1 &
                    data_1$Sectores.de.actividad.CNAE.2009 == input$Sectores_1 &
                    data_1$Componentes.del.coste == input$Componentes_1 )

  #Vector con todas las posiciones que contengan el año y trimestre desde el que comienzo a visualizar
  vectorPosInicio =  grep(Inicio_1(), busqueda$Periodo)

  #Vector con todas las posiciones que contengan el año y trimestre desde el que Termino la visualización
  vectorPosFin = grep(Fin_1(), busqueda$Periodo)

  #Filtra en base al rango en Años y Trimestres desde el que se pretende visualizar
  aux = slice(busqueda, vectorPosFin[1]:vectorPosInicio[1] )
  for(i in 1:length(vectorPosInicio)){
    aux2 = slice(busqueda, vectorPosFin[i]:vectorPosInicio[i] )
    aux = full_join(aux, aux2)
  }#for
  aux = select(aux, Periodo, Comunidades.y.Ciudades.Autónomas, value) %>% arrange(Periodo)
  aux
})#Filtrar1
```

   Esta función hace uso de otras dos auxiliares llamadas *Inicio* y *Fin*, cuya función es juntar el año y trimestre(dado que se han obtenido de 2 inputs diferentes).

```
  #Une el Año y trimestre inicial seleccionado
  Inicio_1 <- reactive({

    paste(as.character(format(input$año_1[1],format="%Y")),input$TC_1, sep="")
  })
  #Une el año y trimestre final seleccionado
  Fin_1 <- reactive({
    paste(as.character(format(input$año_1[2],format="%Y")),input$TF_1, sep="")
  })
```

   Una vez realizado esto, se puede usar *ggplot* (librería *ggplot2)* para construir los gráficos, en mi caso genero dos, uno de líneas y otro de barras.

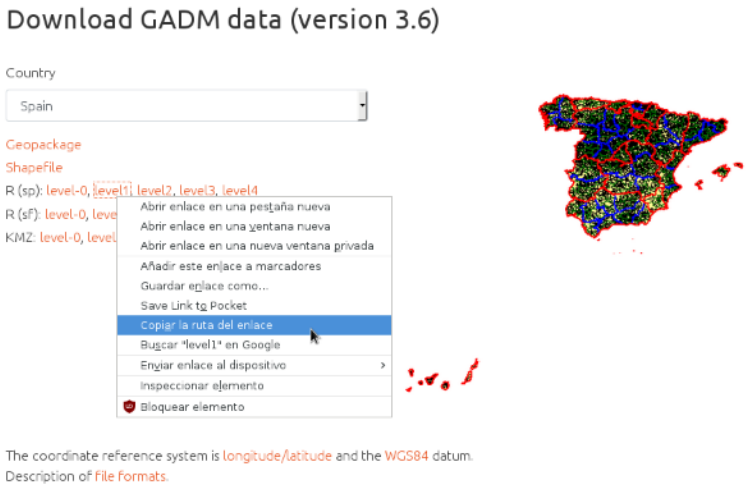
```
output$grafico1_1 <- renderPlot({
  query = Filtrar_1()
  ggplot(query, aes(x =Periodo , y = value, group=1, colour=Comunidades.y.Ciudades.Autónomas )) + geom_line() + facet_grid(Comunidades.y.Ciudades.Autónomas ~ .)
})#Grafica1_1
output$grafico2_1 <- renderPlot ({
  query = Filtrar_1()
  ggplot(query, aes(x=Periodo, y=value, fill=Comunidades.y.Ciudades.Autónomas)) + geom_bar(stat="identity", position=position_dodge())
})#Grafico2_1
```

Como se puede leer, en el eje X sitúo el periodo que el usuario ha decidido que quiere visualizar, en el eje Y los valores numéricos relacionados con esa variable y cada comunidad autónoma de diferente color.

2. Creación del mapa de calor

Para generar el mapa, lo obtengo a partir de la siguiente página web *https://gadm.org/download_country_v3.html*

Ahí seleccionamos Spain y buscamos el link del objeto sp y nivel de administración 1. Después se copia la ruta del enlace.



A continuación uso *readRDS* para cargarlo en R (generará un objeto de tipo *SpatialPolygonsDataFrame)*

```
spain <- readRDS(gzcon(url("https://biogeo.ucdavis.edu/data/gadm3.6/Rsp/gadm36_ESP_1_sp.rds")))
```

Además creamos la paleta de colores para colorear el mapa

```
coloresRojos <- c('#FFF5F0','#FFF5F0','#FEE0D2','#FEE0D2',
                  '#FCBBA1','#FCBBA1','#FC9272','#FC9272',
                  '#FB6A4A','#FB6A4A','#EF3B2C','#EF3B2C',
                  '#CB181D','#A50F15','#A50F15','#67000D')
```

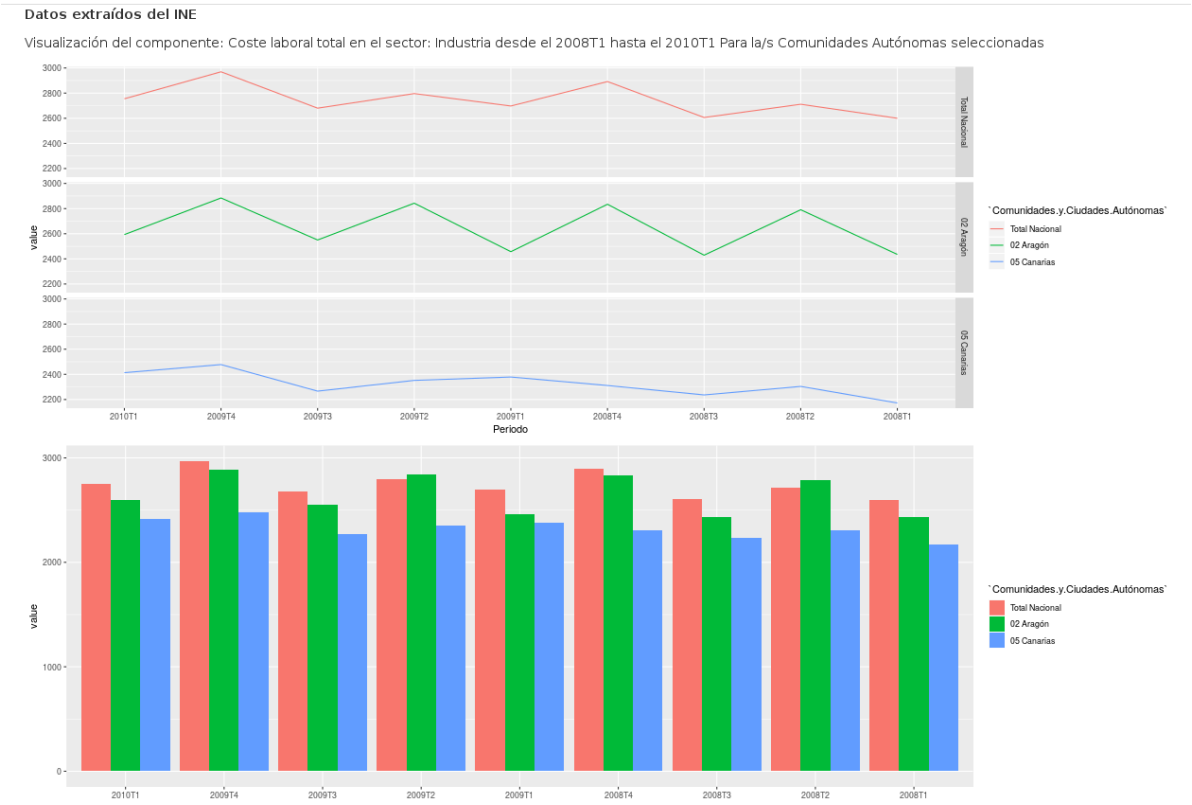
Para dibujar el mapa uso el siguiente algoritmo:

```
output$mapita_1 <- renderPlot({
  query = filter(data_1, Componentes.del.coste==input$Componentes_1, Sectores.de.actividad.CNAE.2009==input$Sectores_1,
                 Periodo==paste(as.character(format(input$añoMc_1,format="%Y")),input$Tmc_1, sep=""),Comunidades.y.Ciudades.Autónomas!="Total Nacional" )
  aux = select(query,  value, Comunidades.y.Ciudades.Autónomas)

  colLevel <- vector(length = length(par))
  for(i in 1:length(colLevel)){
    if(i!=7){
      q = filter(aux, Comunidades.y.Ciudades.Autónomas==par[[i]][1])
      q = select(q, value)
      colLevel[i] = q
    }#if
  }
  valores <- unlist(colLevel)
  valores[7] = valores[6]
  spain@data$mc=valores
  spplot(spain, 'mc', col.regions=coloresRojos )

})#representacion del mapa_1
```

El cual crea un vector llamado colLevel del tamaño el número de comunidades autónomas, lo recorre asignando el valor que se ha obtenido al filtrar y, finalmente, lo guarda en el mapa de España y lo muestra usando *ssplot*

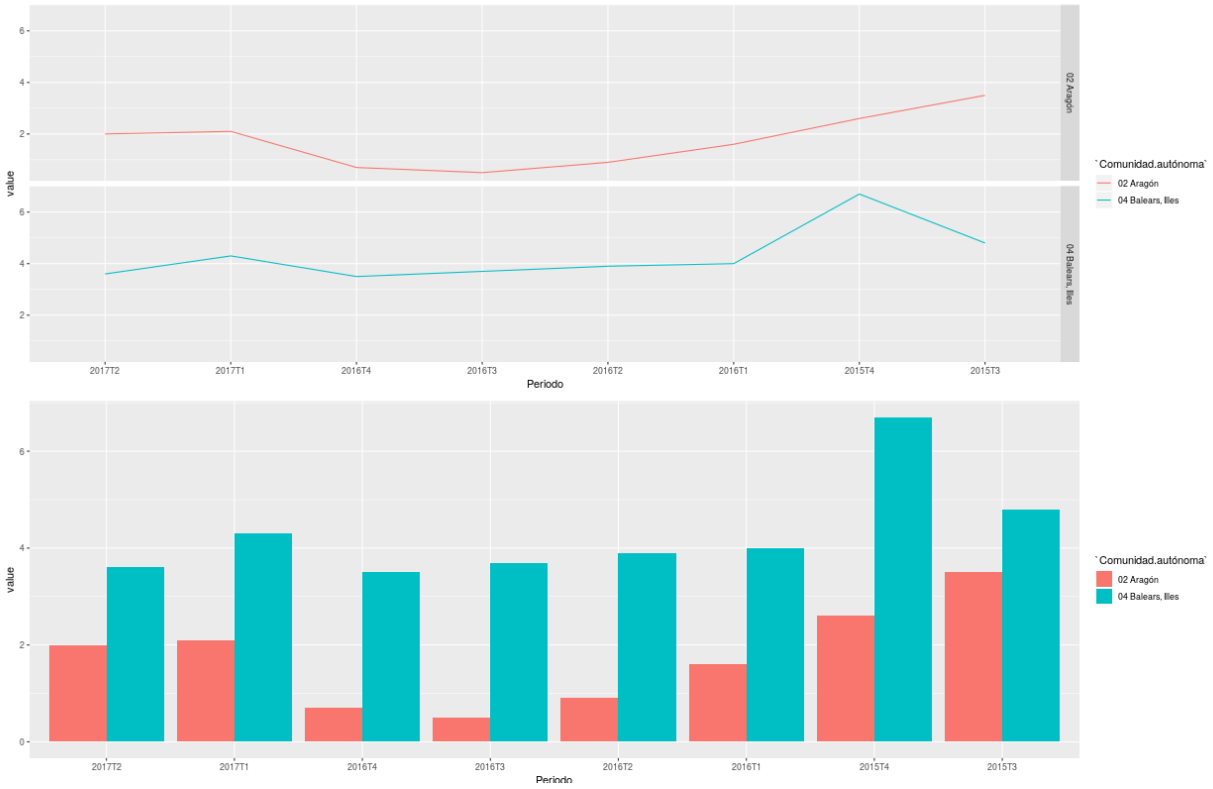
- **Resultado:**

El resultado final que se muestra en la aplicación Shiny es el siguiente:



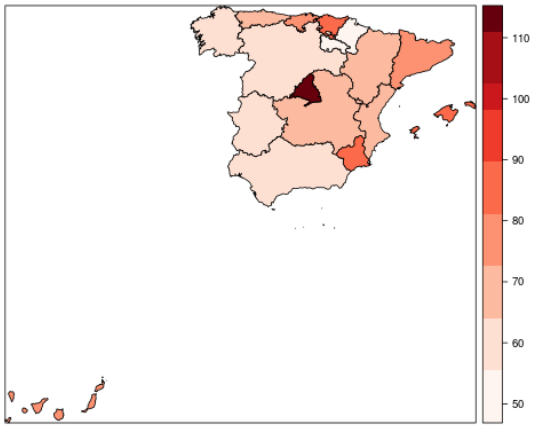
Datos extraídos del INE

Visualización de los datos del Motivo: Elevado coste de contratación desde el 2015T3 hasta el 2017T2 Para la/s Comunidades Autónomas seleccionadas



Datos extraídos del INE

Visualización del mapa de calor del componente: Coste por percepciones no salariales en el sector: Servicios y durante el año 2008 T1



C) *Presentación y visualización de todas las partes anteriores, incluidos los datos, gráficos, tablas, informe, código, etc.*

Todo el trabajo, incluyendo el código desarrollado puede ser accedido en GitHub en un repositorio público que he creado.

https://github.com/rv0lt/TrabajoMacro

Dentro del mismo se puede encontrar:

- El código dividido en server.R y ui.R
- El código adaptado a una plantilla Rmd
- Este documento

Además como el código está publico cualquier persona con conexión a internet y R instalado puede ejecutar la aplicación escribiendo estos dos comandos desde una consola de R.

```
> library(shiny)
> runGitHub('TrabajoMacro', 'rv0lt')
```