

**Apartado a (1 punto)**

```
def lee_datos_ventas(fichero):
    registros = []
    with open(fichero, encoding='utf-8') as f:
        lector = csv.reader(f)
        next(lector)
        for año, comunidad, contratos, llamadas, redes in lector:
            año = int(año)
            contratos = int(contratos)
            llamadas = int(llamadas)
            redes = int(redes)
            registros.append(DatosVentas(
                año, comunidad, contratos, llamadas, redes))
    return registros
```

Apartado b (1 punto)

```
def total_contratos(registros, comunidad=None):
    return sum(r.contratos for r in registros
               if comunidad is None or comunidad == r.comunidad)
```

Apartado c (1 punto)

```
def año_mas_contratos_por_llamadas_comunidad(registros, comunidad):
    return max(((r.año, r.contratos * 100 / r.llamadas)
               for r in registros
               if r.comunidad == comunidad), key = lambda t:t[1])
```

Apartado d (1,5 puntos)

```
def variaciones_anuales_contratos(registros, comunidad):
    registros_comunidad = [r for r in registros
                           if r.comunidad == comunidad]

    return [(r1.año, r2.año, r2.contratos - r1.contratos)
            for r1, r2 in zip(registros_comunidad, registros_comunidad[1:])]
```



Apartado e (1,5 puntos)

```
def contratos_por_año(registros):
    res = {}
    for r in registros:
        if r.año not in res:
            res[r.año] = 0
        res[r.año] += r.contratos
    return res

def año_mas_contratos(registros):
    return max(((año, contratos)
                for año, contratos in contratos_por_año(registros).items()),
               key = lambda t:t[1])
```

Apartado f (2 puntos)

```
def agrupa_datos_por_años(registros):
    aux={}
    for r in registros:
        if r.año not in aux:
            aux[r.año] = [0, 0, 0]
        aux[r.año][0]+=r.contratos
        aux[r.año][1]+=r.llamadas
        aux[r.año][2]+=r.redes
    return[DatosVentas(año, "Todas", n1, n2, n3)
           for año, (n1, n2, n3) in aux.items()]

def agrupa_datos_por_años(registros):
    aux={}
    for r in registros:
        if r.año not in aux:
            aux[r.año] = [0, 0, 0]
        for i in range(0,3):
            aux[r.año][i] += r[2+i]
    return[DatosVentas(año, "Todas", n1, n2, n3)
           for año, (n1, n2, n3) in aux.items()]

def agrupa_datos_por_años(registros):
    aux = {}
    for r in registros:
        if r.año not in aux:
            aux[r.año] = [0, 0, 0]
        aux[r.año] = [x+y for x, y in zip(aux[r.año], r[2:])]
    return [DatosVentas(año, "Todas", n1, n2, n3)
           for año, (n1, n2, n3) in aux.items()]
```

**Apartado g (2 puntos)**

```
def correlaciones_datos_anuales(registros):
    aux = agrupa_datos_por_años(registros)
    serie_contratos = [r[2] for r in aux]

    return pearson(serie_contratos, [r[3] for r in aux]),
            pearson(serie_contratos, [r[4] for r in aux])
```