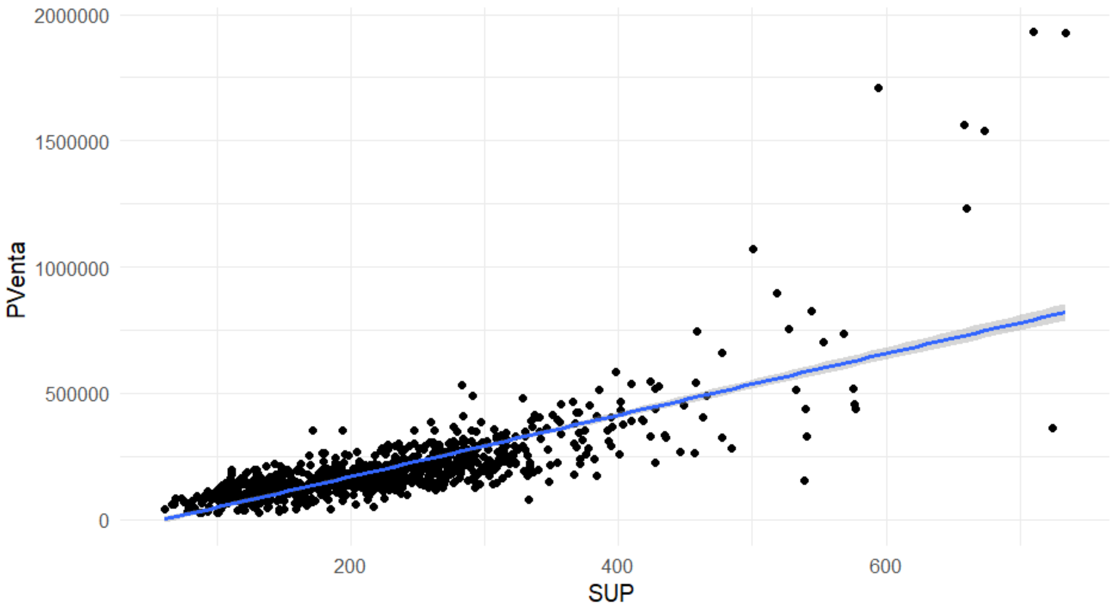
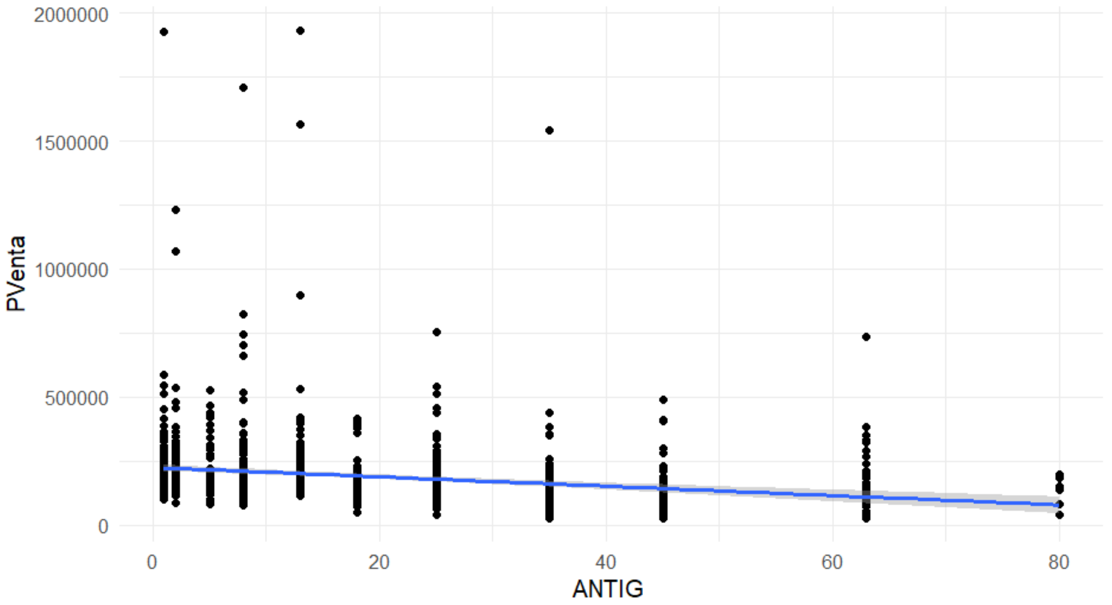
a)

Tras ajustar un modelo de regresión a ls datos obtenemos:

$$Y = -51280.80 + -923.03X_1 + 1197.06X_2 \ + \ e$$

Donde X_1 representa la antigüedad de la casa y X_2 representa la superficie de la casa en metros cuadrados.





Código R:

```r
library(tidyverse)
library(GGally)
library(modelsummary)
library(skimr)
library(qtl2)

Inmov <- read.table("C:/Users/November/Downloads/INMOB.csv", header =
FALSE, sep = "", stringsAsFactors = FALSE)

Inmov <- na.omit(Inmov)

Inmov <- Inmov %>%
  mutate(across(everything(), as.numeric))

Inmov
str(Inmov) #V1 Antig, V2 Pventa, V3 SUP
dim(Inmov)
str(Inmov)
head(Inmov)
tail(Inmov)
summary(Inmov)
skim(Inmov)
ggplot(Inmov, aes(x = V1, y = V2 )) +
  geom_point() +
  geom_smooth(method='lm', formula = y~x) +
  xlab("ANTIG ()") +
  ylab("PVenta ()") +
  theme_minimal()

ggplot(Inmov, aes(x = V3, y = V2 )) +
  geom_point() +
  geom_smooth(method='lm', formula = y~x) +
  xlab("SUP ()") +
  ylab("PVenta ()") +
  theme_minimal()
```

Código Python:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from scipy.stats import t

# Read the CSV file
```

```python
Inmov = pd.read_csv("C:/Users/November/Downloads/INMOB.csv", header=0,
delim_whitespace=True)

# Remove NA values
Inmov = Inmov.dropna()

# Convert all columns to numeric
Inmov = Inmov.apply(pd.to_numeric)

# Display the DataFrame
print(Inmov)

# Structure of the DataFrame
print(Inmov.info())  # V1 Antig, V2 Pventa, V3 SUP
print(Inmov.shape)
print(Inmov.info())
print(Inmov.head())
print(Inmov.tail())
print(Inmov.describe())

# Plotting
plt.figure(figsize=(10, 5))
sns.scatterplot(data=Inmov, x=Inmov['ANTIG'], y=Inmov['PVENTA'])
sns.regplot(data=Inmov, x=Inmov['ANTIG'], y=Inmov['PVENTA'],
scatter=False, color='red')
plt.xlabel("ANTIG ()")
plt.ylabel("PVENTA ()")
plt.title("Scatter plot of ANTIG vs PVenta")
plt.show()

plt.figure(figsize=(10, 5))
sns.scatterplot(data=Inmov, x=Inmov['SUP'], y=Inmov['PVENTA'])
sns.regplot(data=Inmov, x=Inmov['SUP'], y=Inmov['PVENTA'], scatter=False,
color='red')
plt.xlabel("SUP ()")
plt.ylabel("PVenta ()")
plt.title("Scatter plot of SUP vs PVenta")
plt.show()
```

b)

Se obtiene un R cuadrado de 0.5896, es decir el modelo se ajusta de manera moderada a los datos. No tiene mucho sentido estudiar el R cuadrado ajustado ya que solo hay dos variables. Al realizar un contraste de hipótesis $H_0\colon b_1 = b_2 = 0$, H_1 LANC. Obtenemos un resultado significativo, es decir, podemos suponer que los datos siguen un modelo lineal. El p-valor obtenido al contrastar el modelo es menor que 2.2e-16, que a su vez es menor que 0.05.

Código R:

```r
lin_model <- lm(formula = V2 ~ V1 + V3, data = Inmov)

summary(lin_model)
```

Código Python:

```python
# Linear model
X = Inmov[['SUP', 'ANTIG']]  # V1 and V3
y = Inmov['PVENTA']        # V2
X = sm.add_constant(X)  # Adds a constant term to the predictor

lin_model = sm.OLS(y, X).fit()

# Summary of the linear model
print(lin_model.summary())
```

c)

Si, los parámetros son estadísticamente significativos, para comprobarlo planteamos los contrastes $H_0: b_1 = 0$, H_1 LANC. Al realizar el contraste obtenemos un p-valor de 1.02e-07 < 0.05, es decir, rechazamos la hipótesis nula y por tanto podemos suponer que el coeficiente asociado a los años del piso es significativo, para el segundo parámetro planteamos el contraste $H_0: b_2 = 0$, H_1 LANC, obtenemos un p-valor menor que 2e-16, al igual que antes esto significa que podemos rechazar la hipótesis nula y por tanto el parámetro asociado con la superficie del piso es significativo.

Código R:

```r
lin_model <- lm(formula = V2 ~ V1 + V3, data = Inmov)

summary(lin_model)
```

Código Python:

```python
# Linear model
X = Inmov[['SUP', 'ANTIG']]  # V1 and V3
y = Inmov['PVENTA']        # V2
X = sm.add_constant(X)  # Adds a constant term to the predictor

lin_model = sm.OLS(y, X).fit()

# Summary of the linear model
print(lin_model.summary())
```

d)

Sabemos por la teoría que $\Delta E[\,Y \mid X_1, X_2, \dots, X_k\,] = \beta_1(\Delta X_1)$, Por tanto para estudiar la variación del precio de un piso cuando la antigüedad del piso aumenta en un año suponiendo que el resto de variables se mantienen fijas (la superficie del piso) tenemos que $\Delta E[\,Y \mid X_1, X_2, \dots, X_k\,] = \beta_1 \cdot 1 =$ -923.03. En resumen, el precio disminuye 923.03 euros por cada año que pasa.

Para estudiar el impacto en el precio que tendrá el aumento de la superficie de 15 metros cuadrados volvemos a aplicar la fórmula $\Delta E[\,Y \mid X_1, X_2, \dots, X_k\,] = \beta_2(\Delta X_2)$ En este caso $\Delta X_2 = 15$, por tanto tenemos que $\Delta E[\,Y \mid X_1, X_2, \dots, X_k\,] = 15 \cdot 1197.06 = 17955.9$ Es decir, una aumento de 15 metros cuadrados en el piso se traduce en un aumento de 17955.9 euros en el precio de la vivienda.

Codigo R:

```
cat("cada año que pasa el precio de la vivienda decrece ",-923.03)
cat("Si se aumentase el tamaño de l vivienda 15 m^2 el precio aumentaría en", 15*1197.06)
```

Codigo Python:

```
print("cada año que pasa el precio de la vivienda decrece", -923.03)

print("Si se aumentase el tamaño de la vivienda en 15 m^2, el precio aumentaría en", 15 * 1197.06)
```

e)

basta sustituir la fórmula:

$$\widehat{\beta_2} \pm Z_{0.05/2} \cdot \sigma_{\widehat{\beta_2}},$$

Donde $Z_{0.05/2}$ representa el cuantil de una distribución normal al nivel 0.05/2, $\widehat{\beta_2}$ la estimación del parámetro β_2 y $\sigma_{\widehat{\beta_2}}$ la estimación de la varianza de dicho parámetro, obtenemos un intervalo:

[1145.006 , 1249.1234]

Como $1500 \notin [1145.006, 1249.1234]$ Rechazamos la hipótesis $H_0\!: b_2 = 1500$

Código R:

```
confint(lin_model,level = 0.90)
```

Código Python:

```
conf_int = lin_model.conf_int(alpha=0.10)
print("Intervalo de confianza al 90% para los coeficientes:\n", conf_int)
```

f)

Planteamos el contraste:

$H_0: b_1 \leq -1000$, H_1 LANC

Obtenemos un p-valor de 0.327 > 0.05, es decir, no podemos rechazar la hipótesis de que un año más de antigüedad disminuye el precio de venta en 1000 euros o más

Codigo R:

```r
coef_antiguedad <- summary(lin_model)$coefficients["V1", "Estimate"]
se_antiguedad <- summary(lin_model)$coefficients["V1", "Std. Error"]

# Calcular el estadístico t
t_stat <- (coef_antiguedad + 1000) / se_antiguedad  # +1000 porque queremos ver si es mayor que -1000

# Calcular el valor p (unilateral)
p_value <- pt(t_stat, df = df.residual(lin_model), lower.tail = FALSE)
p_value
# Resultados
cat("Estadístico t:", t_stat, "\n")
cat("Valor p:", p_value, "\n")

# Definir nivel de significancia
alpha <- 0.05

# Contraste de hipótesis
if (p_value < alpha) {
  cat("Se rechaza la hipótesis nula. Hay suficiente evidencia para afirmar que la antigüedad disminuye el precio de venta en menos de 1000 euros.\n")
} else {
  cat("No se rechaza la hipótesis nula. No hay suficiente evidencia para afirmar que la antigüedad disminuye el precio de venta en menos de 1000 euros.\n")
}
```

Código Python:

```python
coef_antig = lin_model.params['ANTIG']
se_antig = lin_model.bse['ANTIG']
t_stat = (coef_antig + 1000) / se_antig

# Calcular el valor p (unilateral)
p_value = t.sf(t_stat, df=lin_model.df_resid)

# Resultados
print("Estadístico t:", t_stat)
```

```python
print("Valor p:", p_value)

# Definir nivel de significancia
alpha = 0.05

# Contraste de hipótesis
if p_value < alpha:
    print("Se rechaza la hipótesis nula. Hay suficiente evidencia para afirmar que la antigüedad disminuye el precio de venta en menos de 1000 euros.")
else:
    print("No se rechaza la hipótesis nula. No hay suficiente evidencia para afirmar que la antigüedad disminuye el precio de venta en menos de 1000 euros.")
```

g)

Basta sustituir dichos datos en el modelo calculado en a), obteniendo un valor de 174286.6 euros

Código R:

```r
nuevos_valores <- data.frame(V1 = 15, V3 = 200)

# Hacer la predicción
predict(lin_model, newdata = nuevos_valores)
```

Código Python:

```python
nuevos_valores = pd.DataFrame({'const': [1], 'ANTIG': [15], 'SUP': [200]})  # 'const' para intercepto
prediccion = lin_model.predict(nuevos_valores)
print("Predicción del precio de la vivienda:", prediccion[0])
```