Start coding or generate with AI.

```
install.packages("wordcloud")
install.packages("plot3D")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘misc3d’
```

```
library(dplyr)
library(wordcloud)
library(RColorBrewer)
library(gridExtra)
library(plot3D)
```

```
Warning message:
“no DISPLAY variable so Tk is not available”
```

```
df <- read.csv("/content/Housing.csv")
```

```
colnames(df)
```

```

```

```
furnishing_freq <- table(df$furnishingstatus)
wordcloud(names(furnishing_freq), furnishing_freq, scale=c(3,0.5), colors=brewe
```

```

```

```
# A word cloud visualizes how frequently different furnishing statuses (like fu
```

```
ggplot(df, aes(x=factor(bedrooms), y=price)) +
  geom_boxplot(aes(fill=factor(bedrooms))) +
  labs(title="Boxplot of Price by Number of Bedrooms", x="Number of Bedrooms",
```

```
# The boxplot shows the price distribution for different numbers of bedrooms, i
```

```
ggplot(df, aes(x=factor(stories), y=price, fill=factor(stories))) +
  geom_violin(trim=FALSE) +
  labs(title="Violin Plot of Price by Stories", x="Number of Stories", y="Price
```

```
# The violin plot visualizes the distribution of house prices for various stori
```

```
ggplot(df, aes(x=area, y=price)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE, color="red") +
  labs(title="Linear Regression of Price vs Area", x="Area", y="Price")
```

```
ggplot(df, aes(x=area, y=price)) +
  geom_point() +
  geom_smooth(method="loess", se=FALSE, color="blue") +
  labs(title="Nonlinear Regression of Price vs Area", x="Area", y="Price")
```

```
scatter3D(x = df$area, y = df$bedrooms, z = df$price,
          xlab = "Area", ylab = "Bedrooms", zlab = "Price",
          col = rainbow(100)[as.numeric(cut(df$price, breaks = 100))],
          pch = 19, cex = 1.5,
          main = "3D Plot of Price vs Area vs Bedrooms")
```

```
# The 3D plot allows us to examine how price varies with both area and the
number of bedrooms simultaneously, revealing clusters or patterns in the data.
```

```
# Jitter plot
ggplot(df, aes(x=factor(parking), y=price, color=factor(parking))) +
```

```
  geom_jitter(width=0.2) +
  labs(title="Jitter Plot of Price by Parking", x="Number of Parking Spaces",
  y="Price")
```

```
# The jitter plot reveals the distribution of prices for different numbers of
parking spaces and helps identify overlapping points.
```