```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("broom")
install.packages("tidyr")
install.packages("glmnet")

The downloaded binary packages are in
        /var/folders/by/52c0hwv97h0v1jwqjzz1z1ldr0000gn/T/Rtmp2Q3BNb/downloaded_packages

The downloaded binary packages are in
        /var/folders/by/52c0hwv97h0v1jwqjzz1z1ldr0000gn/T/Rtmp2Q3BNb/downloaded_packages

The downloaded binary packages are in
        /var/folders/by/52c0hwv97h0v1jwqjzz1z1ldr0000gn/T/Rtmp2Q3BNb/downloaded_packages

The downloaded binary packages are in
        /var/folders/by/52c0hwv97h0v1jwqjzz1z1ldr0000gn/T/Rtmp2Q3BNb/downloaded_packages
```

```
library(ggplot2)
library(dplyr)
library(broom)
library(tidyr)
library(glmnet)
```

```
Attaching package: 'dplyr'


The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

Loading required package: Matrix

Loaded glmnet 4.1-8
```

```
housing_data = read.csv("housing.csv")
head(housing_data)[1:5,1:10]
```

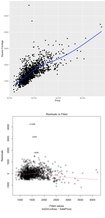
```
  longitude latitude housing_median_age total_rooms total_bedrooms population households median_income
1      -122       37                 41        880            129        322        126             8.3
2      -122       37                 21       7099           1106       2401       1138             8.3
3      -122       37                 52       1467            190        496        177             7.2
4      -122       37                 52       1274            235        558        219             5.6
5      -122       37                 52       1627            280        565        259             3.8
  median_house_value ocean_proximity
1             452600        NEAR BAY
2             358500        NEAR BAY
3             352100        NEAR BAY
4             341300        NEAR BAY
5             342200        NEAR BAY
```

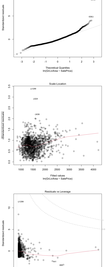
```
# Exploring Sale Price and Square Footage columns
ggplot(housing_data, aes(x = SalePrice, y = SalePrice)) +
  geom_point() +
  labs(title = "Relation Plot of Price and Square Footage", x = "Square Footage", y = "Sale Price")

# Linear relationship, as square footage goes up, so does price.
```

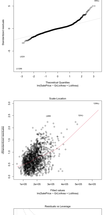
```
# Linear regression single expression
linear = lm(SalePrice ~ GrLivArea, data = housing_data)

# Regression equation
equation = summary(linear)$coefficients

# Residuals
ggplot(housing_data, aes(x = GrLivArea, y = SalePrice)) +
  geom_point() + geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Relation Plot of Price and Square Footage with Regression Line",
       x = "Price", y = "Square Footage")

summary(linear)

plot(linear)
```

```
Warning message:
In geom_smooth(method = "lm", se = FALSE, color = "blue"):
Ignoring unknown parameters:  linewidth
`geom_smooth()` using formula = 'y ~ x'
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -175.419   4.507  -38.92   <2e-16 ***
GrLivArea    107.130   1.226   87.40   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 56.4 on 1458 degrees of freedom
Multiple R-squared:  0.5252,	Adjusted R-squared:  0.5218
F-statistic:  7639 on 1 and 1458 DF,  p-value: < 2.2e-16
```

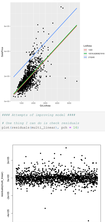







```
# Linear regression multiple expression

# Checking the correlation between the independent variables
cor(housing_data$GrLivArea, housing_data$Bedroom)

# Third linear
multi_linear = lm(SalePrice ~ GrLivArea, y = SalePrice) +
  geom_point() +
  labs(title = "Relation Plot of Sale Price and Lot Area", x = "Lot Area", y = "Sale Price")

ggplot(housing_data, aes(x = GrLivArea, y = SalePrice)) +
  geom_point() +
  labs(title = "Relation Plot of Sale Price and Square Footage", x = "Square Footage", y = "Sale Price")

# I see a linearity in both as well as in scatterplots.
```

```
0.0961001287148
```




```
# Multiple expression

multi_linear = lm(SalePrice ~ GrLivArea + OverallQual, data = housing_data)

summary(multi_linear)

plot(multi_linear)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -109475   4.47663  -24.45   <2e-16 ***
GrLivArea    [illegible]   0.[illegible]   [illegible]   <2e-16 ***
OverallQual  [illegible]   [illegible]   [illegible]   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: [illegible] on 1457 degrees of freedom
Multiple R-squared:  [illegible],	Adjusted R-squared:  [illegible]
F-statistic:  [illegible] on 2 and 1457 DF,  p-value: < 2.2e-16
```

```
# Create a new variable for the interaction between two variables
printing_data = mapped_price.
printing_data$interaction = printing_data$Bedroom; new(housing_data$Bedroom); length.out = dim.
  interaction = rbind(housing_data$Bedroom, new(housing_data$Bedroom); new(housing_data$Bedroom))

# Make predictions based on linear model
printing_data$predicted_y = predict.lm(multi_linear, newdata = printing_data)
```

```
# Make for area & price
printing_data$interaction = as.factor(printing_data$interaction)

# Plot the interaction plot
ggplot(printing_data, aes(x = GrLivArea, y = SalePrice)) +
  geom_point()

save_plot
```

```
# Add regression lines
ggplot(printing_data, aes(x = GrLivArea, y = SalePrice)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue")

save_plot
```

```
Warning messages:
Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



```
# Make attempts at improving model ####

# Use Cross-Fold and K-Fold validation
plot(cross(multi_linear), gam = 10)
```



```
# Ridge Regression
ridge_model = cv.glmnet(as.matrix(housing_data), y=[illegible], "Gaussian"[illegible]),
                        housing_data$SalePrice, alpha = 0)

# Print the optimal lambda value
cat("Optimal lambda:", ridge_model$lambda.min, "\n")

# Plot
plot(ridge_model)
```

```
Optimal lambda: 5427.562
```