

Newfood

Question 1

The screenshot shows the R Studio environment with the following components:

- Source Editor:** Contains R code for loading data, calculating a correlation matrix, and fitting three linear models.
- Environment:** Lists the objects in the global environment: `correlation_mat`, `newfood`, `regression_price`, and `regression_price_ad_loc`.
- Console:** Displays the output of the R code, including the correlation matrix and the results of the linear models.

```
# Newfood
getwd()
setwd("/Users/veerabhmahadik/Desktop")
newfood<-read.csv("newfood.csv")

# Question1
# Correlation matrix
correlation_matrix <- cor(newfood)
print(correlation_matrix)

# Question 2
#Regression of sales on price alone, price and ad, price, ad, and loc
regression_price <- lm(sales ~ price, data = newfood)
regression_price_ad <- lm(sales ~ price + ad, data = newfood)
regression_price_ad_loc <- lm(sales ~ price + ad + loc, data = newfood)

# Coefficients of price
print(summary(regression_price)$coefficients[, "Estimate"])
print(summary(regression_price_ad)$coefficients[, "Estimate"])
print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])

# Coefficients of ad
print(summary(regression_price_ad)$coefficients[, "Estimate"])
print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])
```

Console Output:

```
R 4.2.1 ~ ./Desktop />
> setwd("/Users/veerabhmahadik/Desktop")
> newfood<-read.csv("newfood.csv")
> # Question1
> # Correlation matrix
> correlation_matrix <- cor(newfood)
> print(correlation_matrix)
      sales      price      ad      loc      income      volume      city
sales 1.000000000 -0.7039149  0.1170423  0.01046719  0.1857936  0.39299715  0.000425552
price -0.703914896  1.0000000  0.0000000  0.00000000 -0.1305879 -0.17850621  0.000000000
ad     0.117042260  0.0000000  1.0000000  0.00000000 -0.7463722 -0.74199852 -0.894427191
loc    0.010467194  0.0000000  0.0000000  1.00000000  0.00000000 -0.03974992  0.000000000
income 0.185793603 -0.1305879 -0.7463722  0.00000000  1.00000000  0.80867129  0.791553868
volume 0.392997149 -0.1785062 -0.7419985 -0.03974992  0.8086713  1.00000000  0.740696036
city   0.000425552  0.0000000 -0.8944272  0.00000000  0.7915539  0.74069604  1.000000000

> # Question 2
> #Regression of sales on price alone, price and ad, price, ad, and loc
> regression_price <- lm(sales ~ price, data = newfood)
> regression_price_ad <- lm(sales ~ price + ad, data = newfood)
>
```

The correlation matrix will show correlations between variables. Zero correlations between location and advertising indicate no linear relationship between those variables.

Question 2

The screenshot displays the R Studio environment. The script editor on the left contains R code for three regression models. The console on the bottom left shows the output of these models. The environment pane on the right lists the objects in the global environment.

```
10  
11 # Question 2  
12 #Regression of sales on price alone, price and ad, price, ad, and loc  
13 regression_price <- lm(sales ~ price, data = newfood)  
14 regression_price_ad <- lm(sales ~ price + ad, data = newfood)  
15 regression_price_ad_loc <- lm(sales ~ price + ad + loc, data = newfood)  
16  
17 # Coefficients of price  
18 print(summary(regression_price)$coefficients[, "Estimate"])  
19 print(summary(regression_price_ad)$coefficients[, "Estimate"])  
20 print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])  
21  
22 # Coefficients of ad  
23 print(summary(regression_price_ad)$coefficients[, "Estimate"])  
24 print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])  
25  
26 # Question 3  
27 # Regression of sales against price, ad, loc, and volume  
28 regression_price_ad_loc_volume <- lm(sales ~ price + ad + loc + volume, data = newfood)  
29 print(summary(regression_price_ad_loc_volume)$coefficients[, "Estimate"])  
30  
31 # Question 4  
32 # Regression with all variables  
33 regression_full <- lm(sales ~ price + ad + loc + volume + income + city, data = newfood)  
34  
26.1 (Top Level) >
```

Console Output:

```
R 4.2.1 ~./Desktop/ >  
> regression_price_ad_loc <- lm(sales ~ price + ad + loc, data = newfood)  
> # Coefficients of price  
> print(summary(regression_price)$coefficients[, "Estimate"])  
(Intercept) price  
673.9 -15.1  
> print(summary(regression_price_ad)$coefficients[, "Estimate"])  
(Intercept) price ad  
663.65 -15.10 20.50  
> print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])  
(Intercept) price ad loc  
662.733333 -15.100000 20.500000 1.833333  
> # Coefficients of ad  
> print(summary(regression_price_ad)$coefficients[, "Estimate"])  
(Intercept) price ad  
663.65 -15.10 20.50  
> print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])  
(Intercept) price ad loc  
662.733333 -15.100000 20.500000 1.833333  
>
```

Environment Pane:

Object	Class	Attributes
correlation_mat	num	[1:7, 1:7] 1 -0.7039 0.117 0.0105 0...
newfood	data.frame	24 obs. of 7 variables
regression_price	lm	List of 12
regression_price_ad	lm	List of 12
regression_price_ad_loc	lm	List of 12

1. Regression of sales on price alone:

- The coefficient of price (-15.1) suggests that for every unit increase in price, sales decrease by 15.1 units, holding other variables constant.

2. Regression of sales on price and ad:

- The coefficient of price remains consistent (-15.1), indicating that its effect on sales remains the same when advertising expenditure is included in the model.

- The coefficient of ad (20.5) indicates that for every unit increase in advertising expenditure, sales increase by 20.5 units, holding other variables constant.

3. Regression of sales on price, ad, and loc:

- The coefficient of price remains consistent (-15.1), suggesting its effect on sales remains unchanged even with the inclusion of location and advertising.

- The coefficient of ad remains consistent (20.5), indicating its consistent effect on sales even when accounting for location.

- The coefficient of loc (1.833) indicates that placing the product in the instant breakfast section increases sales by 1.833 units compared to placing it in the bread section, holding other variables constant.

Question 3:

The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains R code for regression analysis. Lines 16-20 show coefficients for price. Lines 22-24 show coefficients for ad. Lines 27-30 show the regression for sales against price, ad, loc, and volume. Lines 31-34 show the regression with all variables.
- Environment:** Lists objects in the Global Environment, including 'correlation_mat', 'newfood', and several 'regression_price' objects.
- Console:** Shows the output of the R commands. It includes the coefficients for price, ad, and loc, and the results of the regression models.

Console Output:

```
(Intercept) price
673.9 -15.1
> print(summary(regression_price_ad)$coefficients[, "Estimate"])
(Intercept) price ad
663.65 -15.10 20.50
> print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])
(Intercept) price ad loc
662.733333 -15.100000 20.500000 1.833333
> # Coefficients of ad
> print(summary(regression_price_ad)$coefficients[, "Estimate"])
(Intercept) price ad
663.65 -15.10 20.50
> print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])
(Intercept) price ad loc
662.733333 -15.100000 20.500000 1.833333
> # Question 3
> # Regression of sales against price, ad, loc, and volume
> regression_price_ad_loc_volume <- lm(sales ~ price + ad + loc + volume, data = newfood)
> print(summary(regression_price_ad_loc_volume)$coefficients[, "Estimate"])
(Intercept) price ad loc volume
125.93100 -11.83586 131.28287 7.76813 11.86959
>
```

Regression Coefficients:

- For regression with price, ad, loc, and volume (regression_price_ad_loc_volume):
 - Intercept: 125.931
 - Price: -11.836
 - Ad: 131.283
 - Loc: 7.768
 - Volume: 11.870
- For regression with price, ad, and loc (regression_price_ad_loc):
 - Intercept: 662.733

- Price: -15.100

- Ad: 20.500

- Loc: 1.833

- The coefficients of price and ad change slightly while intercept changes from 662.733 to 125.931

- The coefficient of loc remains approximately the same in both regressions.

- In regression_price_ad_loc, there is no volume variable, so its introduction in regression_price_ad_loc_volume leads to a significant change in its coefficient. This happens because the volume variable captures additional variation in sales that was not explained by the previous variables (price, ad, and loc).

Question 4

The screenshot displays the RStudio environment with a script editor on the left and a console on the bottom. The script contains R code for fitting three linear regression models: a base model with price, ad, and loc; a model adding volume; and a full model with all variables including income and city. The console shows the output of these models, with coefficients for price, ad, loc, and volume. The volume coefficient is notably different in the full model compared to the model without it.

```
# Lab3_StatsMethods_VeerabhMahad...  
20 print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])  
21  
22 # Coefficients of ad  
23 print(summary(regression_price_ad)$coefficients[, "Estimate"])  
24 print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])  
25  
26 # Question 3  
27 # Regression of sales against price, ad, loc, and volume  
28 regression_price_ad_loc_volume <- lm(sales ~ price + ad + loc + volume, data = newfood)  
29 print(summary(regression_price_ad_loc_volume)$coefficients[, "Estimate"])  
30  
31 # Question 4  
32 # Regression with all variables  
33 regression_full <- lm(sales ~ price + ad + loc + volume + income + city, data = newfood)  
34 print(summary(regression_full)$coefficients)  
35  
36 # Question 5  
37 # Additional regression runs  
38 # Regression with interaction between price and advertising  
39 regression_interaction <- lm(sales ~ price * ad + loc + volume, data = newfood)  
40  
36:1 (Top Level) z
```

Console Output:

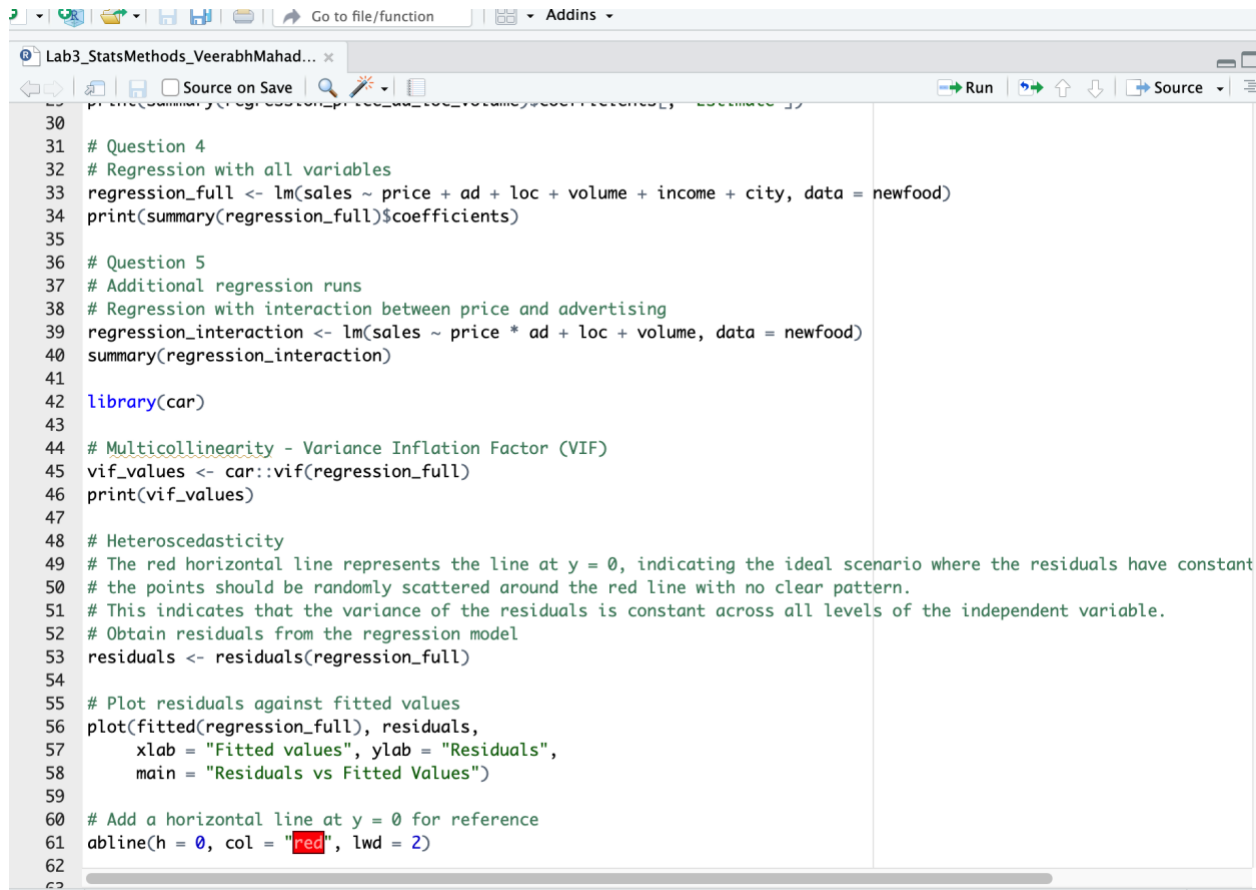
```
> print(summary(regression_price_ad_loc)$coefficients[, "Estimate"])  
(Intercept)      price          ad          loc  
662.7333333 -15.1000000  20.5000000  1.8333333  
> # Question 3  
> # Regression of sales against price, ad, loc, and volume  
> regression_price_ad_loc_volume <- lm(sales ~ price + ad + loc + volume, data = newfood)  
> print(summary(regression_price_ad_loc_volume)$coefficients[, "Estimate"])  
(Intercept)      price          ad          loc      volume  
125.9310000 -11.83586  131.28287  7.76813  11.86959  
> # Question 4  
> # Regression with all variables  
> regression_full <- lm(sales ~ price + ad + loc + volume + income + city, data = newfood)  
> print(summary(regression_full)$coefficients)  
              Estimate Std. Error t value Pr(>|t|)  
(Intercept) 158.477999 161.414867  0.9818055 3.399630e-01  
price       -12.111446  2.277839 -5.3170772 5.675513e-05  
ad          159.380489  41.422266  3.8477009 1.290101e-03  
loc         7.917013  17.858955  0.4433077 6.631331e-01  
volume      12.167360  2.631440  4.6238401 2.424338e-04  
income     -14.298812  22.278454 -0.6418225 5.295485e-01  
city        21.000205  19.693020  1.0663781 3.011639e-01  
>
```

- The coefficients for price, ad, loc, and volume are relatively consistent between the two models.

There are slight variations in the estimates and standard errors, but the overall trends remain similar.

- The regression_full model includes additional variables for income and city, which were not present in the regression_price_ad_loc_volume model.
- The coefficient for income is estimated to be -14.30, indicating a negative relationship with sales, although it is not statistically significant (p-value > 0.05).
- The coefficient for city is estimated to be 21.00, indicating a positive relationship with sales, but again, it is not statistically significant (p-value > 0.05).
- The intercept in the regression_full model is estimated to be 158.48, which is different from the intercept in the regression_price_ad_loc_volume model (125.93). This difference is a result of the inclusion of additional variables.

Question 5



```
30  
31 # Question 4  
32 # Regression with all variables  
33 regression_full <- lm(sales ~ price + ad + loc + volume + income + city, data = newfood)  
34 print(summary(regression_full)$coefficients)  
35  
36 # Question 5  
37 # Additional regression runs  
38 # Regression with interaction between price and advertising  
39 regression_interaction <- lm(sales ~ price * ad + loc + volume, data = newfood)  
40 summary(regression_interaction)  
41  
42 library(car)  
43  
44 # Multicollinearity - Variance Inflation Factor (VIF)  
45 vif_values <- car::vif(regression_full)  
46 print(vif_values)  
47  
48 # Heteroscedasticity  
49 # The red horizontal line represents the line at y = 0, indicating the ideal scenario where the residuals have constant  
50 # the points should be randomly scattered around the red line with no clear pattern.  
51 # This indicates that the variance of the residuals is constant across all levels of the independent variable.  
52 # Obtain residuals from the regression model  
53 residuals <- residuals(regression_full)  
54  
55 # Plot residuals against fitted values  
56 plot(fitted(regression_full), residuals,  
57      xlab = "Fitted values", ylab = "Residuals",  
58      main = "Residuals vs Fitted Values")  
59  
60 # Add a horizontal line at y = 0 for reference  
61 abline(h = 0, col = "red", lwd = 2)  
62  
63
```

```

Lab3_StatsMethods_VeerabhMahad... x
47
63:1 (Top Level) R Script
Console Terminal Background Jobs
R 4.2.1 ~ /Desktop/
income -14.298814 22.278454 -0.6418225 5.295485e-01
city 21.000205 19.693020 1.0663781 3.011639e-01
> # Question 5
> # Additional regression runs
> # Regression with interaction between price and advertising
> regression_interaction <- lm(sales ~ price * ad + loc + volume, data = newfood)
> summary(regression_interaction)

Call:
lm(formula = sales ~ price * ad + loc + volume, data = newfood)

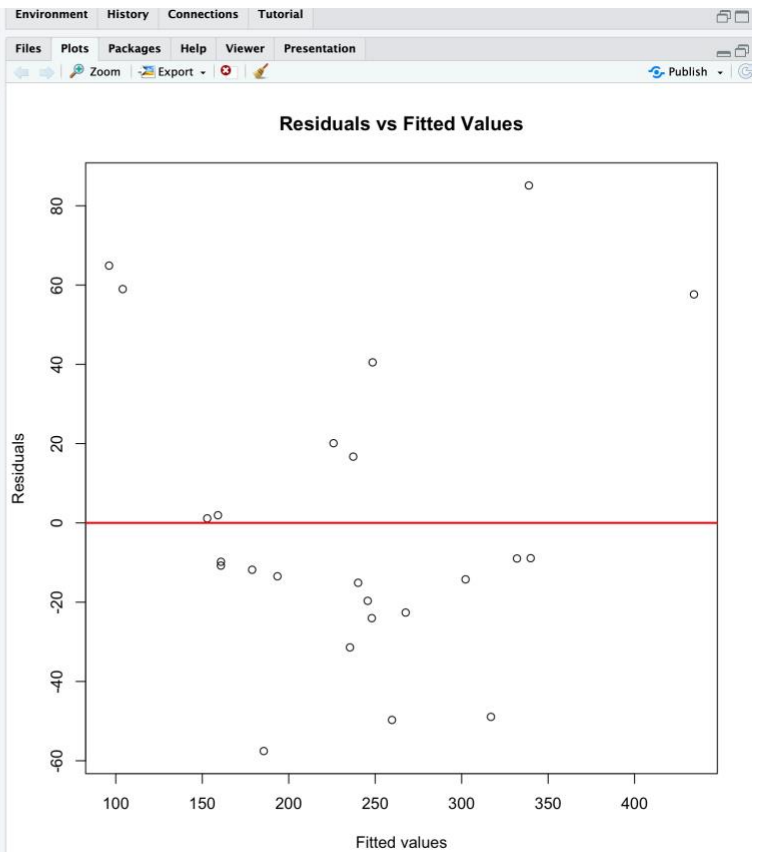
Residuals:
    Min       1Q   Median       3Q      Max
-61.394 -25.448  -0.473  25.769  54.221

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.124    114.215   0.054  0.95783
price        -7.247     2.712  -2.672  0.01553 *
ad          399.776    111.598   3.582  0.00213 **
loc           7.590     15.529   0.489  0.63091
volume       11.514     1.916   6.009 1.11e-05 ***
price:ad      -9.373     3.808  -2.462  0.02415 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 37.97 on 18 degrees of freedom
Multiple R-squared:  0.859, Adjusted R-squared:  0.8199
F-statistic: 21.94 on 5 and 18 DF, p-value: 4.423e-07

> library(car)
> # Multicollinearity - Variance Inflation Factor (VIF)
> vif_values <- car::vif(regression_full)
> print(vif_values)
      price      ad      loc      volume      income      city
1.090452 5.409028 1.005457 3.453871 3.823000 6.112884
> # Heteroscedasticity
> # The red horizontal line represents the line at y = 0, indicating the ideal scenario w
here the residuals have constant variance around zero.
> # the points should be randomly scattered around the red line with no clear pattern.
> # This indicates that the variance of the residuals is constant across all levels of th
e independent variable.

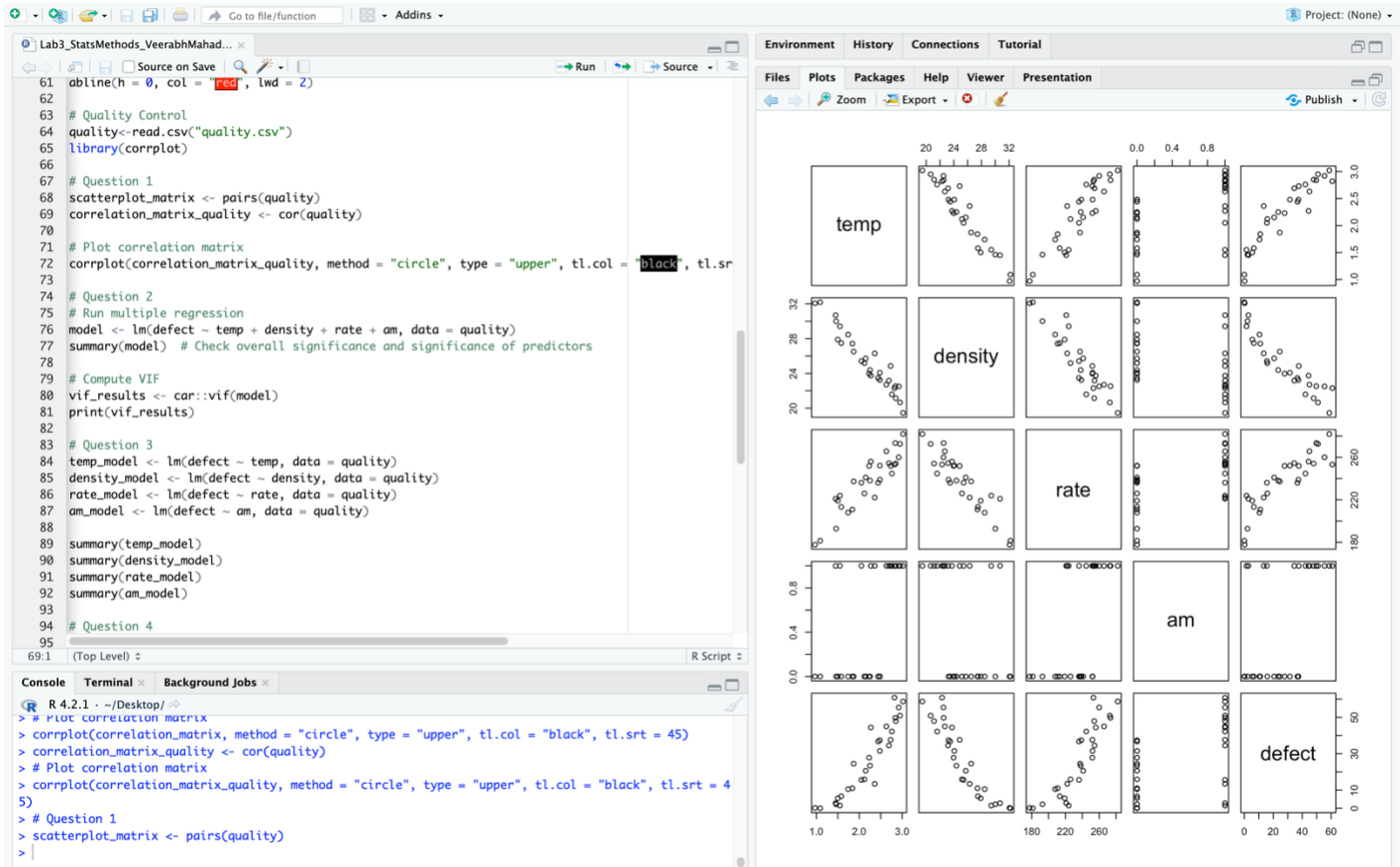
```

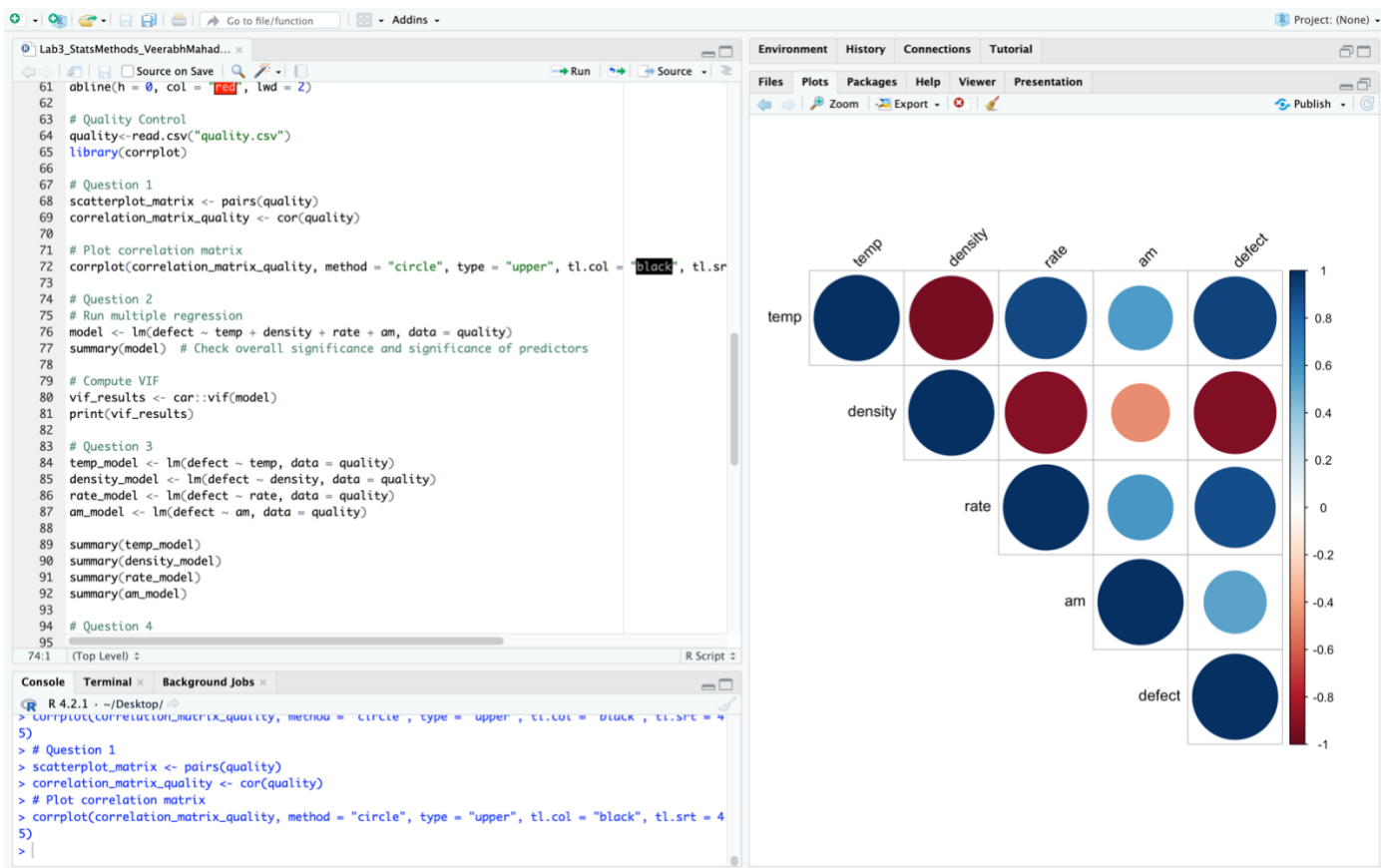


- The VIF values reveal the extent of multicollinearity for each predictor variable in the regression model..
- A VIF value close to 1 implies minimal multicollinearity between the predictor variable and other variables in the model. Variables with VIF values near 1, such as "price," "loc," and "volume," indicate minimal multicollinearity with other variables.
- Typically, a VIF exceeding 10 indicates substantial multicollinearity, signifying a significant increase in the variance of the coefficients. Predictor variables like "ad," "income," and "city" have VIF values ranging from 3.823 to 6.113, suggesting moderate multicollinearity but not to a concerning extent.
- Heteroscedasticity: The red horizontal line represents the line at $y = 0$, indicating the ideal scenario where the residuals have constant variance around zero. The points are randomly scattered around the red line with no clear pattern. This indicates that the variance of the residuals is constant across all levels of the independent variable.

Quality Control

Question 1





- Strong correlations exist between temp and density, temp and rate, and density and rate. This indicates a tight interrelation among these variables.
- temp, rate, and am show moderate positive correlations with defect, suggesting that increases in these variables are associated with higher defect rates.
- In contrast, density exhibits a strong negative correlation with defect, implying that higher density correlates with lower defect rates.
- The strong correlations between temp, density, and rate hint at potential multicollinearity issues in regression analysis.

Question 2

The screenshot displays the R Studio environment with a script editor on the left and the console on the right. The script editor contains the following code:

```

74 # Question 2
75 # Run multiple regression
76 model <- lm(defect ~ temp + density + rate + am, data = quality)
77 summary(model) # Check overall significance and significance of predictors
78
79 # Compute VIF
80 vif_results <- car::vif(model)
81 print(vif_results)
82
83 # Question 3
84 temp_model <- lm(defect ~ temp, data = quality)

```

The console output shows the results of the linear regression model:

```

> corplot(correlation_matrix_quality, method = "circle", type = "upper", tl.col = "black", tl.srt = 45)
> # Question 2
> # Run multiple regression
> model <- lm(defect ~ temp + density + rate + am, data = quality)
> summary(model) # Check overall significance and significance of predictors

Call:
lm(formula = defect ~ temp + density + rate + am, data = quality)

Residuals:
    Min       1Q   Median       3Q      Max
-13.3478  -3.7614  -0.7298   3.0401  15.8935

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  31.14457    77.09396   0.404   0.690
temp         13.95814     9.26003   1.507   0.144
density      -2.24388     1.69777  -1.322   0.198
rate          0.08883     0.14253   0.623   0.539
am            1.94741     3.65017   0.534   0.598

Residual standard error: 7.203 on 25 degrees of freedom
Multiple R-squared:  0.8813,    Adjusted R-squared:  0.8623 
F-statistic: 46.42 on 4 and 25 DF,  p-value: 3.23e-11

> # Compute VIF
> vif_results <- car::vif(model)
> print(vif_results)
      temp  density    rate      am
16.314613 18.205395  7.706898  1.926129

```

The Environment pane on the right shows the following objects:

- regression_full: List of 12
- regression_interac...: List of 12
- regression_price: List of 12
- regression_price_ad: List of 12
- regression_price_a...: List of 12
- regression_price_a...: List of 12

The Values pane shows the following data:

- residuals: Named num [1:24] -15.11 -8.99 85.12 -48.93 -24.03 ...
- scatterplot_matrix: NULL
- vif_results: Named num [1:4] 16.31 18.21 7.71 1.93

- The overall model is statistically significant (F-statistic = 46.42, p-value = 3.23e-11), indicating that at least one predictor has a significant relationship with the defect variable.
- None of the individual predictors (temp, density, rate, am) are statistically significant at the 0.05 level, as indicated by their p-values.
- VIF values for temp, density, and rate are considerably high, suggesting multicollinearity issues among these predictors.
- While the overall model is significant, the lack of significance among individual predictors and high VIF values indicate potential issues with model specification.

Question 3

Lab3_StatsMethods_VeerabhMahad... x

Source on Save

Run

Source

```
78
79 # Compute VIF
80 vif_results <- car::vif(model)
81 print(vif_results)
82
83 # Question 3
84 temp_model <- lm(defect ~ temp, data = quality)
85 density_model <- lm(defect ~ density, data = quality)
86 rate_model <- lm(defect ~ rate, data = quality)
87 am_model <- lm(defect ~ am, data = quality)
88
89 summary(temp_model)
90 summary(density_model)
91 summary(rate_model)
92 summary(am_model)
93
94 # Question 4
95 # Check appropriateness of regressing defect on the first component
96 pca <- prcomp(~ temp + density + rate + am, data = quality, scale. = TRUE)
97 summary(pca)
98
99 # Question 5
100
```

(Top Level) ± R Script

Console Terminal Background Jobs x

R 4.2.1 ~ /Desktop/

Call:
lm(formula = defect ~ rate, data = quality)

Residuals:

	Min	1Q	Median	3Q	Max
	-17.3159	-5.1129	-0.7204	7.6170	22.6529

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-128.90616	15.57665	-8.276	5.26e-09 ***
rate	0.65977	0.06548	10.077	8.13e-11 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.185 on 28 degrees of freedom
Multiple R-squared: 0.7838, Adjusted R-squared: 0.7761
F-statistic: 101.5 on 1 and 28 DF, p-value: 8.132e-11

> |

Environment History Connections Tutorial

R - Global Environment

426 MiB

Data

- am_model List of 12
- correlation_matrix num [1:5, 1:5] 1 -0.959 0.908 0.563 0.929 ...
- correlation_matrix num [1:5, 1:5] 1 -0.959 0.908 0.563 0.929 ...
- density_model List of 12
- model List of 12
- newfood 24 obs. of 7 variables
- quality 30 obs. of 5 variables
- rate_model List of 12
- regression_full List of 12

Files Plots Packages Help Viewer Presentation

Zoom Export

R 4.2.1 · ~/Desktop/

```
> # Question 3
> temp_model <- lm(defect ~ temp, data = quality)
> density_model <- lm(defect ~ density, data = quality)
> rate_model <- lm(defect ~ rate, data = quality)
> am_model <- lm(defect ~ am, data = quality)
> summary(temp_model)
```

Call:
lm(formula = defect ~ temp, data = quality)

Residuals:

	Min	1Q	Median	3Q	Max
	-18.5937	-4.9138	-0.6179	4.2113	15.1887

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-40.966	5.295	-7.736	1.99e-08 ***
temp	30.915	2.326	13.291	1.29e-13 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.308 on 28 degrees of freedom
Multiple R-squared: 0.8632, Adjusted R-squared: 0.8583
F-statistic: 176.6 on 1 and 28 DF, p-value: 1.29e-13

```
> summary(density_model)
```

Call:
lm(formula = defect ~ density, data = quality)

Residuals:

	Min	1Q	Median	3Q	Max
	-11.802	-4.702	-0.803	4.419	17.740

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	161.979	10.685	15.16	5.01e-15 ***
density	-5.333	0.419	-12.73	3.68e-13 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.585 on 28 degrees of freedom
Multiple R-squared: 0.8526, Adjusted R-squared: 0.8473
F-statistic: 162 on 1 and 28 DF, p-value: 3.677e-13

Environment History Connections Tutorial

R · Global Environment

Data

- am_model List of 12
- correlation_matrix num [1:5, 1:5] 1 -0.959 0.908 0.563 0.929 ...
- correlation_matrix num [1:5, 1:5] 1 -0.959 0.908 0.563 0.929 ...
- density_model List of 12
- model List of 12
- newfood 24 obs. of 7 variables
- quality 30 obs. of 5 variables
- rate_model List of 12
- regression_full List of 12

Files Plots Packages Help Viewer Presentation

Zoom Export

The screenshot displays the RStudio interface with the following components:

- Console:** Shows the execution of three linear models and their summaries.
 - Model 1 (density_model):** `lm(formula = defect ~ density, data = quality)`. Summary statistics: Residual standard error: 7.585, Multiple R-squared: 0.8632, Adjusted R-squared: 0.8583, F-statistic: 176.6 on 1 and 28 DF, p-value: 1.29e-13. Coefficients: (Intercept) 161.979, density -5.333. Significance: density is highly significant (p < 0.001).
 - Model 2 (rate_model):** `lm(formula = defect ~ rate, data = quality)`. Summary statistics: Residual standard error: 9.185, Multiple R-squared: 0.7838, Adjusted R-squared: 0.7761, F-statistic: 101.5 on 1 and 28 DF, p-value: 8.132e-11. Coefficients: (Intercept) -128.90616, rate 0.65977. Significance: rate is highly significant (p < 0.001).
 - Model 3 (am_model):** `lm(formula = defect ~ am, data = quality)`. Summary statistics: Residual standard error: 7.585, Multiple R-squared: 0.8632, Adjusted R-squared: 0.8583, F-statistic: 176.6 on 1 and 28 DF, p-value: 1.29e-13. Coefficients: (Intercept) 161.979, am -0.001. Significance: am is not significant (p > 0.05).
- Environment:** Lists the objects created in the Global Environment: `am_model`, `correlation_matrix`, `density_model`, `model`, `newfood`, `quality`, `rate_model`, and `regression_full`.

- The coefficient estimate for temp is highly significant (p-value < 0.001), indicating a strong relationship between temperature and defect rate.

- Similarly, the coefficient estimate for density is highly significant (p-value < 0.001), suggesting a significant association between density and defect rate.

- The coefficient estimate for rate is also highly significant (p-value < 0.001), indicating a significant relationship between rate and defect rate.

- The coefficient estimate for am is not significant (p-value > 0.05), suggesting that there is insufficient evidence to conclude that am has a significant impact on defect rate in this model.

In summary, temp, density, and rate are significant predictors of defect rate, while `am` is not significant in this analysis.

Question 4

The screenshot shows an RStudio interface with a script editor on the left and a console at the bottom. The script editor contains R code for computing VIF, fitting linear models, and performing PCA. The console shows the output of these operations, including model summaries and PCA results.

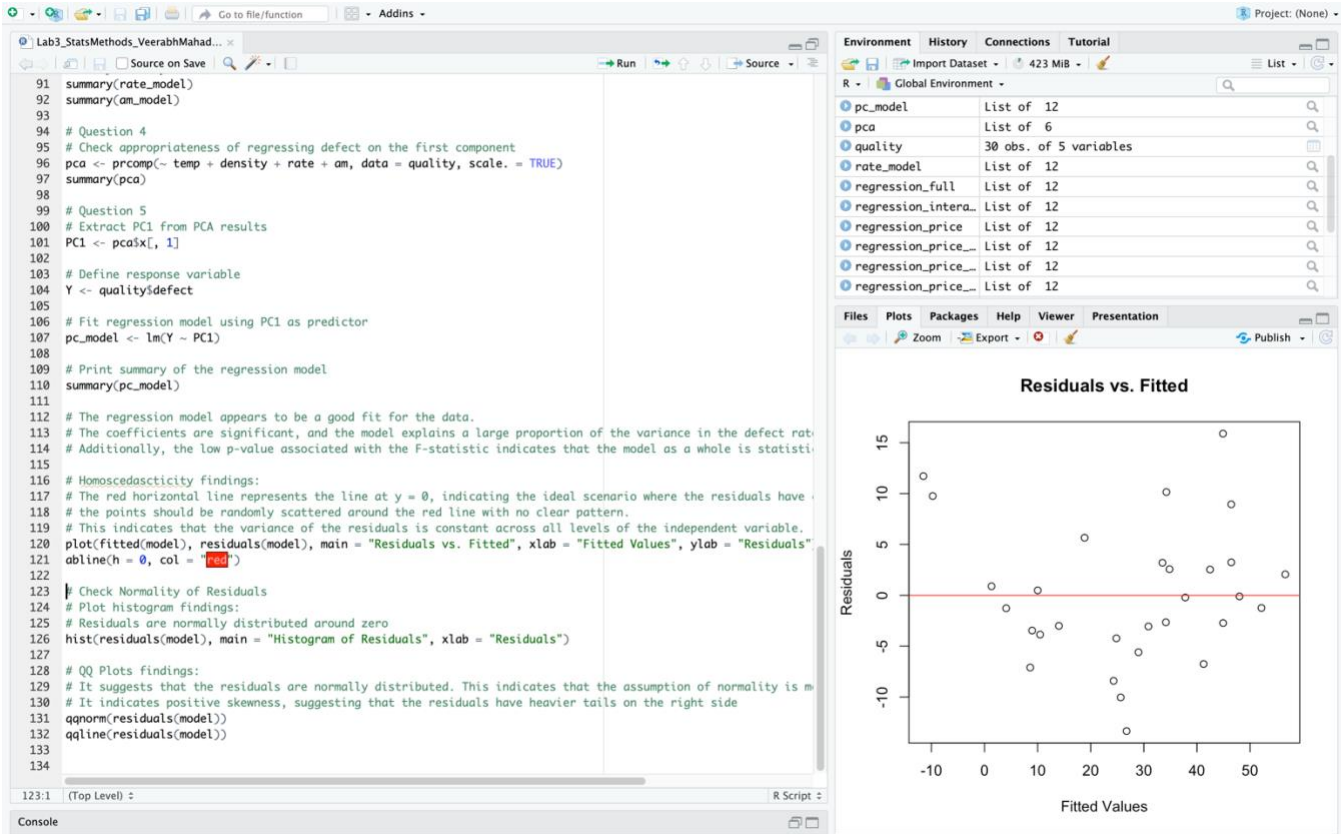
```
78  
79 # Compute VIF  
80 vif_results <- car::vif(model)  
81 print(vif_results)  
82  
83 # Question 3  
84 temp_model <- lm(defect ~ temp, data = quality)  
85 density_model <- lm(defect ~ density, data = quality)  
86 rate_model <- lm(defect ~ rate, data = quality)  
87 am_model <- lm(defect ~ am, data = quality)  
88  
89 summary(temp_model)  
90 summary(density_model)  
91 summary(rate_model)  
92 summary(am_model)  
93  
94 # Question 4  
95 # Check appropriateness of regressing defect on the first component  
96 pca <- prcomp(~ temp + density + rate + am, data = quality, scale. = TRUE)  
97 summary(pca)  
98  
99 # Question 5  
100 # Extract PC1 from PCA results  
101 PC1 <- pca$x[, 1]  
102  
103  
99:1 (Top Level) ↕
```

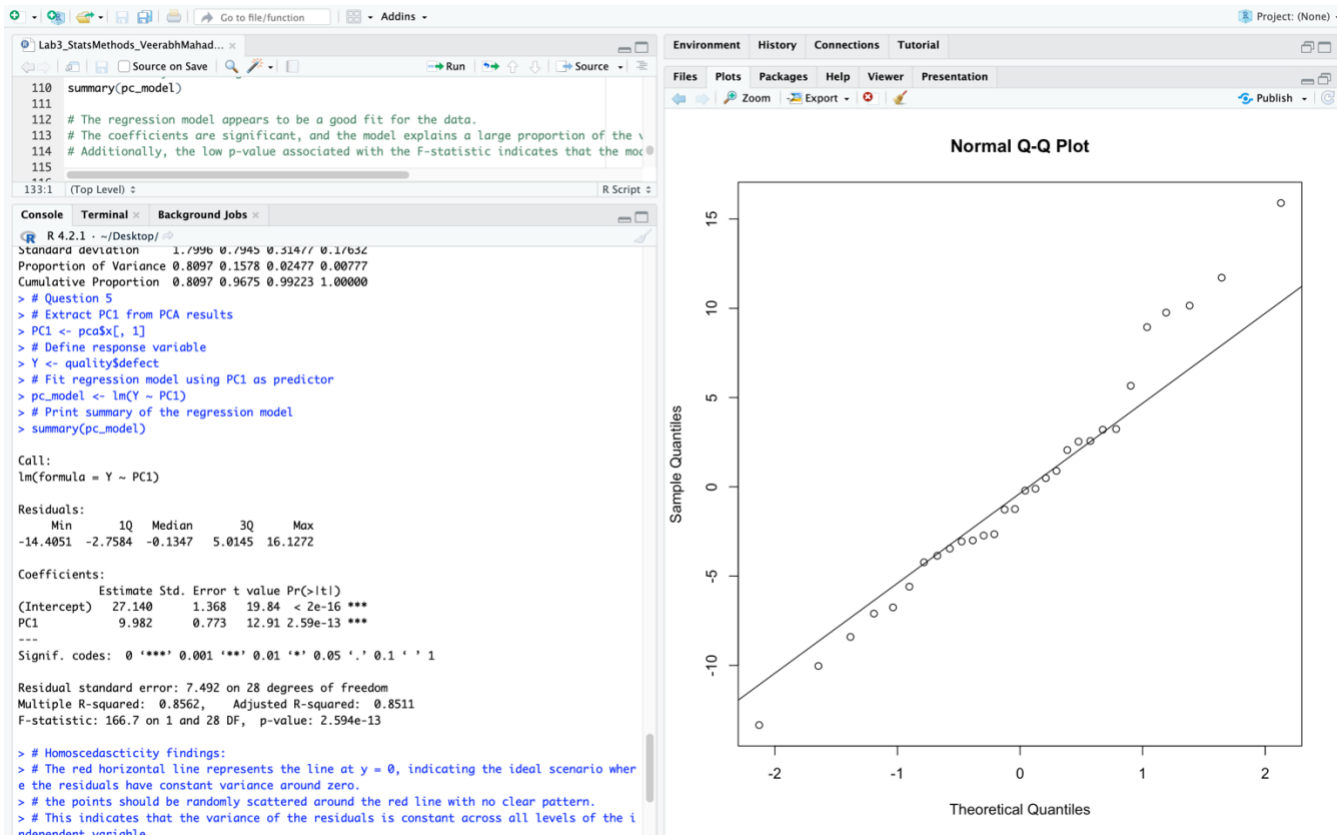
Console Output:

```
R 4.2.1 ~ ./Desktop/ ↕  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 9.185 on 28 degrees of freedom  
Multiple R-squared:  0.7838,    Adjusted R-squared:  0.7761  
F-statistic: 101.5 on 1 and 28 DF,  p-value: 8.132e-11  
  
> # Question 4  
> # Check appropriateness of regressing defect on the first component  
> pca <- prcomp(~ temp + density + rate + am, data = quality, scale. = TRUE)  
> summary(pca)  
Importance of components:  
          PC1      PC2      PC3      PC4  
Standard deviation  1.7996  0.7945  0.31477  0.17632  
Proportion of Variance 0.8097  0.1578  0.02477  0.00777  
Cumulative Proportion 0.8097  0.9675  0.99223  1.00000  
>
```

PC1 accounts for approximately 80.97% of the total variance in the data. Regressing defect on the first principal component (PC1) may be appropriate given its high proportion of variance.

Question 5

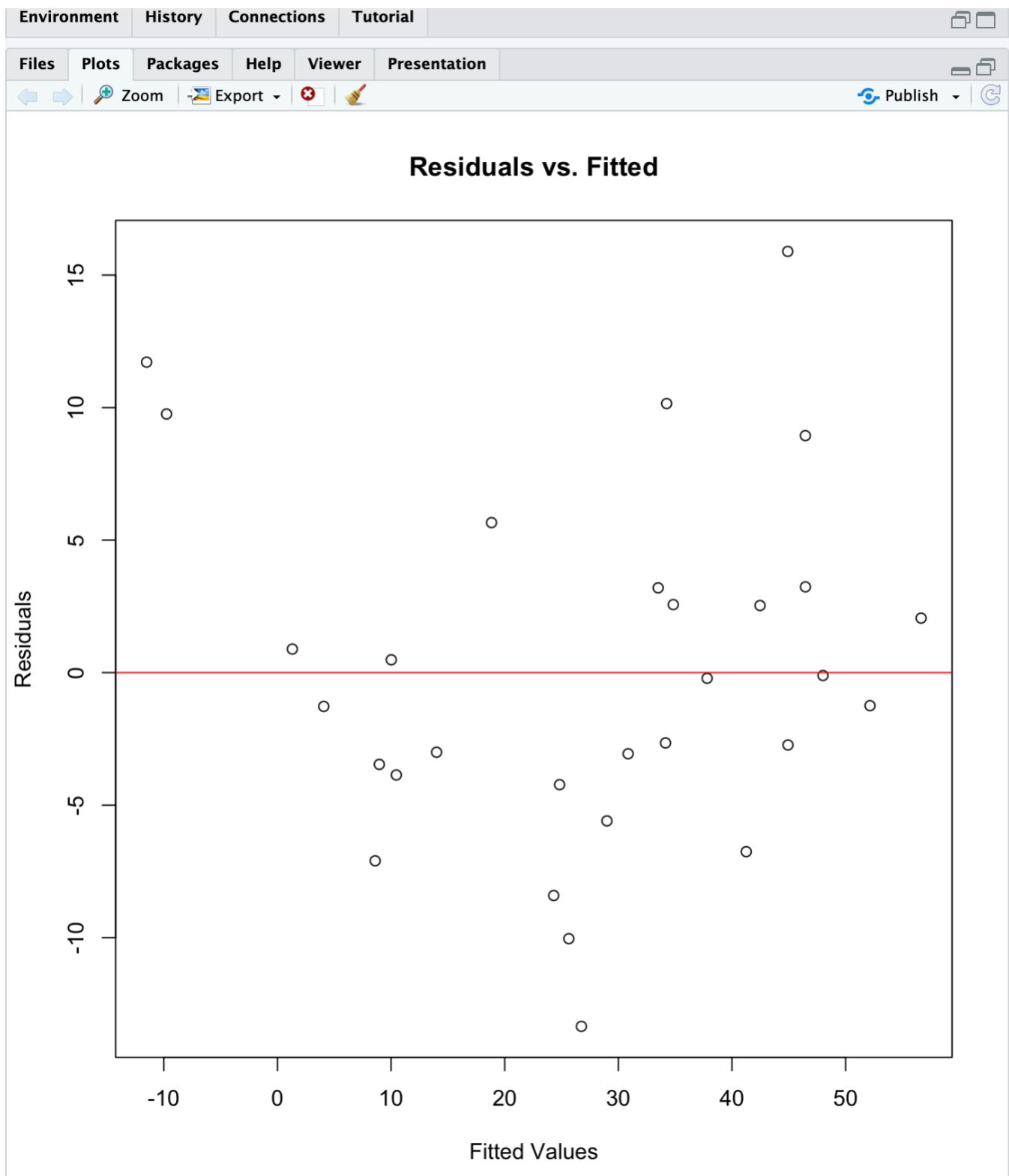




The regression model appears to be a good fit for the data.

The coefficients are significant, and the model explains a large proportion of the variance in the defect rate.

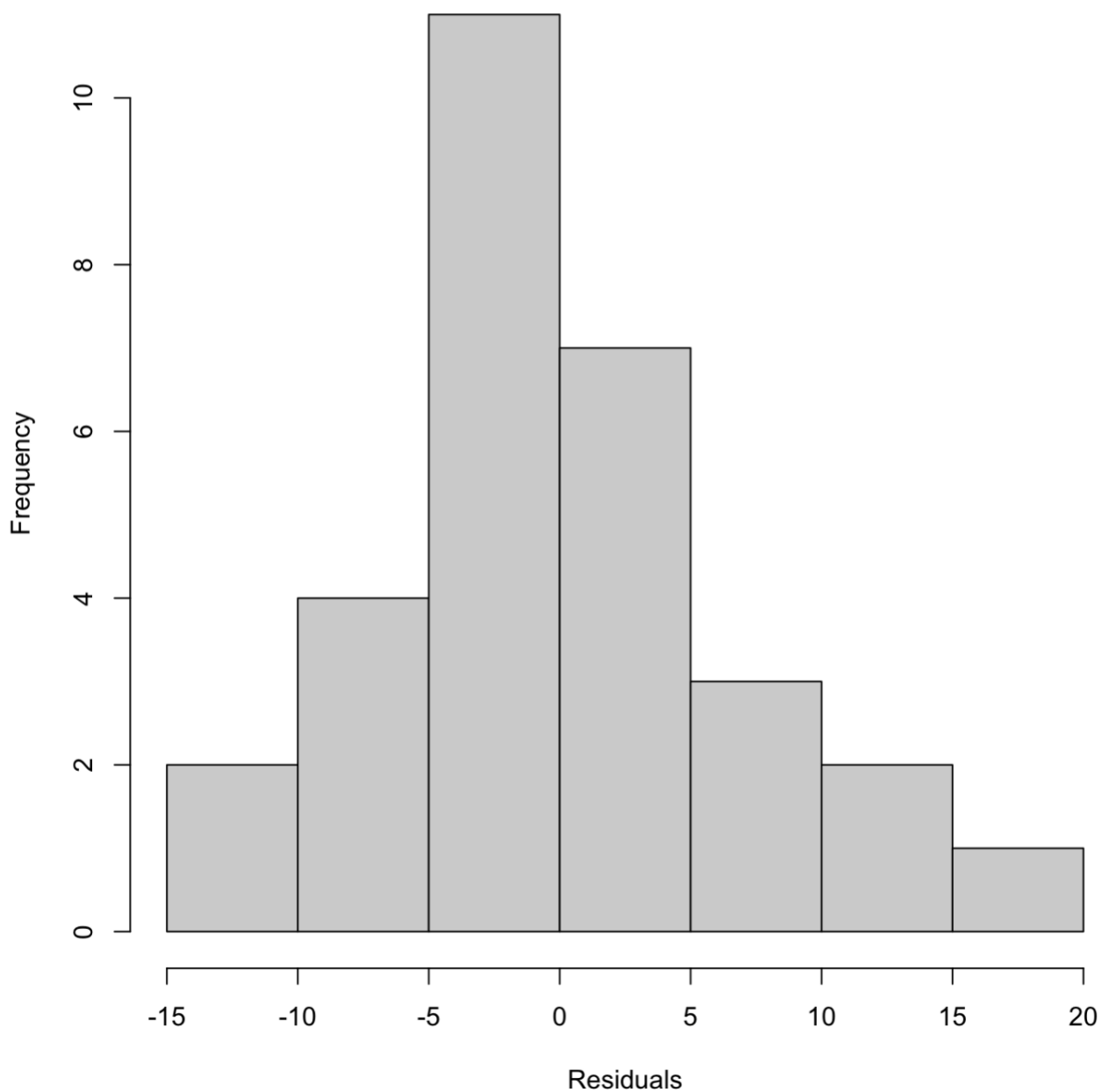
Additionally, the low p-value associated with the F-statistic indicates that the model as a whole is statistically significant.



Homoscedasticity:

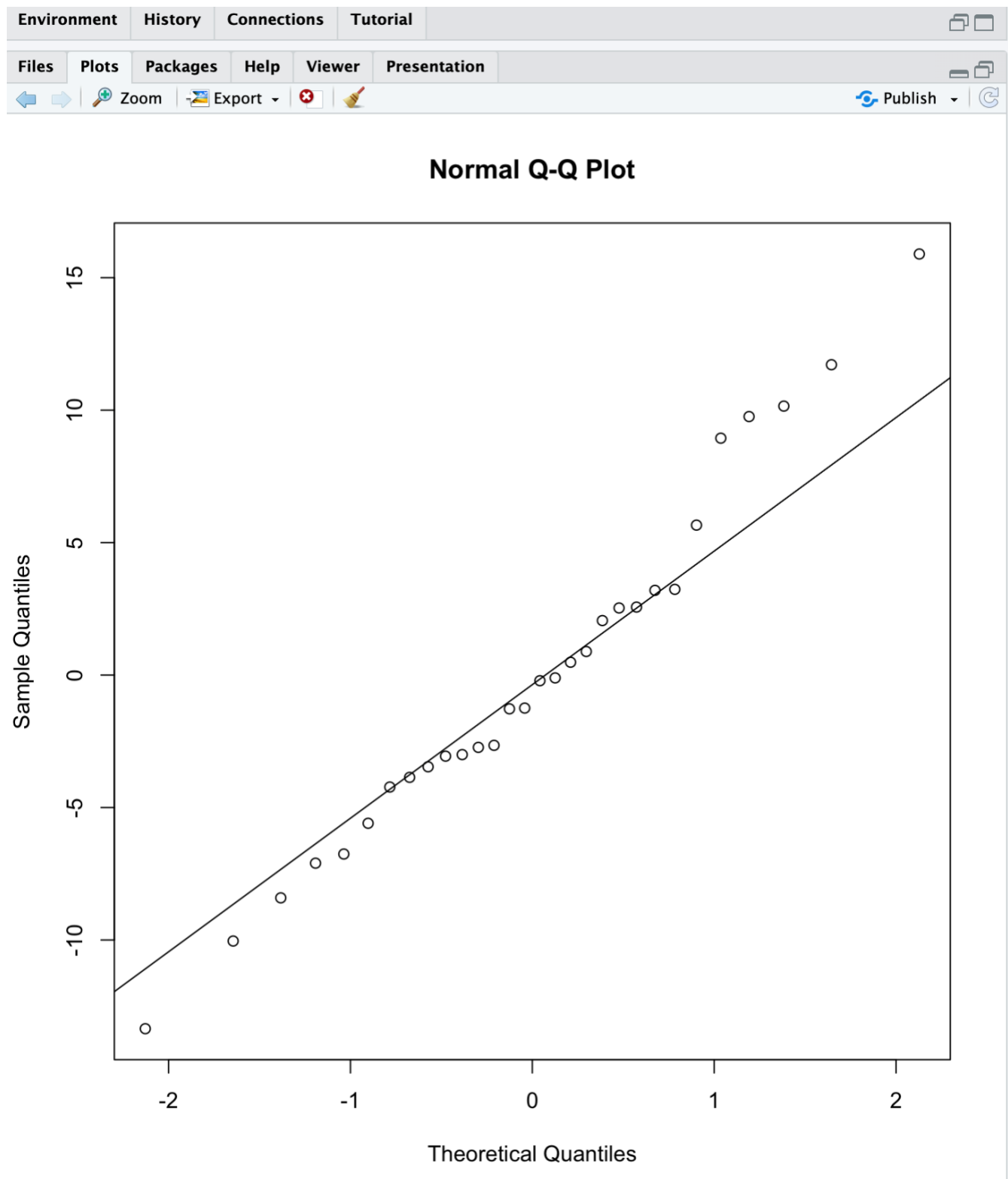
- The red horizontal line represents the line at $y = 0$, indicating the ideal scenario where the residuals have constant variance around zero.
- The points are randomly scattered around the red line with no clear pattern. This indicates that the variance of the residuals is constant across all levels of the independent variable.

Histogram of Residuals



Check Normality of Residuals

-Residuals are normally distributed around zero



QQ Plots findings:

- It suggests that the residuals are normally distributed. This indicates that the assumption of normality is met.
- It indicates positive skewness, suggesting that the residuals have heavier tails on the right side

Question 6

- The simple linear regression models also reveal that both temperature (temp) and density individually have a significant impact on the defect, as indicated by their low p-values and high t-values.
 - The multiple regression model has an Adjusted R-squared value of 0.8623, indicating that approximately 86.23% of the variance in the defect can be explained by the predictors included in the model.
- The F-statistic's low p-value (3.23×10^{-11}) suggests that the model is statistically significant, meaning that at least one of the predictors has a non-zero effect on the defect.
 - The VIF values for each predictor indicate that multicollinearity might be an issue, especially for temperature (temp) and density, which have high VIF values above 10.
- The reliability of the model can be ensured by QQ plot, normality of residuals, PCA, heteroscedasticity.