Advanced Statistics

Regression Analysis

Leslie Salt Data Set (Regression Analysis)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | variables |  |  | Nature of variables | | |  |  |
|  |  |  |  |  |
|  | Price | | continuous variable |  | Interval | Dependent variable | |  |
|  | County | | categorical variable |  | Nominal | Independent variable | |  |
|  | Size | | continuous variable |  | Interval | Independent variable | |  |
| Elevation | | | continuous variable |  | Interval | Independent variable | |  |
|  | Sewer | | continuous variable |  | Interval | Independent variable | |  |
|  | Date | | continuous variable |  | Interval | Independent variable | |  |
|  | Flood | | categorical variable |  | Nominal | Independent variable | |  |
|  | Distance | | continuous variable |  | Interval | Independent variable | |  |

R Code for EDA and Model creation

#import:

saltdata=read.table("LESLIE\_SALT.csv",header=TRUE,sep=";",dec=".")

head(saltdata)

summary(saltdata)

dim(saltdata)

pairs(saltdata)

* Scatterplot Matrices from the glus Package library(gclus)

dta <- saltdata # get data - just put your own data here! dta.r <- abs(cor(dta)) # get correlations

dta.col <- dmat.color(dta.r) # get colors

* reorder variables so those with highest correlation
* are closest to the diagonal

dta.o <- order.single(dta.r)

cpairs(dta, dta.o, panel.colors=dta.col, gap=.5, main="Variables Ordered and Colored by Correlation" )

* 8 Normal probility plots on the same plot page: par(mfrow=c(2,4))

for (i in 1:8) qqnorm(saltdata[,i])

* Adding the log-price:

saltdata$logPrice=log(saltdata$Price)

# 9 histograms with color:

par(mfrow=c(3,3)) for (i in 1:9) hist(saltdata[,i],col=i)

attach(saltdata)

* scatterplot for logprice: with Added fit lines par(mfrow=c(1,1))

plot(logPrice˜Size) abline(lm(logPrice˜Size), col="red")

* scatterplot for logprice: identifying the observation numbers par(mfrow=c(1,1))

plot(logPrice˜Size,type="n")

text(Size,logPrice,labels=row.names(saltdata)) abline(lm(logPrice˜Size), col="red")

* For all of them:

par(mfrow=c(2,4)) for (i in 2:8)

{

plot(logPrice˜saltdata[,i],type="n",xlab=names(saltdata)[i])

text(saltdata[,i],logPrice,labels=row.names(saltdata))

abline(lm(logPrice˜saltdata[,i]), col="red")

}

# Saving directly to a pgn-file:

png("logprice\_relations.png",width=800,height=600)

par(mfrow=c(2,4)) for (i in 2:8)

{

plot(logPrice˜saltdata[,i],type="n",xlab=names(saltdata)[i])

text(saltdata[,i],logPrice,labels=row.names(saltdata))

abline(lm(logPrice˜saltdata[,i]), col="red")

}

dev.off()

* Correlation (with 2 decimals) round(cor(saltdata),2)
* Making a table for viewing in a browser - to be included in e.g. Word/Powerpoint
* Go find/use the cortable.html file afterwards

library(xtable)

capture.output(print(xtable(cor(saltdata)),type="html"),file="cortable.html")

# Regression part

#import: saltdata=read.table("LESLIE\_SALT.csv",header=TRUE,sep=";",dec=".")

* Simple regression lm1=lm(logPrice˜Distance,data=saltdata)

summary(lm1)

# Full MLR model:

lm2=lm(logPrice˜County+Size+Elevation+Sewer+Date+Flood+Distance,data=saltdata)

summary(lm2)

step(lm2,direction="backward")

lm3=lm(formula = logPrice ˜ Elevation + Sewer + Date + Flood + Distance, data = saltdata)

summary(lm3)

* plot fitted vs. observed - identifying observation nr: par(mfrow=c(1,1)) plot(saltdata$logPrice,lm3$fitted,type="n") text(saltdata$logPrice,lm3$fitted,labels=row.names(saltdata))
* Regression Diagnostics plots given automatically, either 4: par(mfrow=c(2,2))

plot(lm3)

* Or 6:

par(mfrow=c(3,2))

plot(lm3,1:6)

* Plot residuals versus individual xs: par(mfrow=c(2,3))

for (i in 4:8)

{

plot(lm3$residuals˜saltdata[,i],type="n",xlab=names(saltdata)[i]) text(saltdata[,i],lm3$residuals,labels=row.names(saltdata)) lines(lowess(saltdata[,i],lm3$residuals),col="blue")

}

* Check for potential quadratic effects and/or interactions:
* Here just two examplesof such:

saltdata$Distance\_square=saltdata$Distanceˆ2

saltdata$County\_Elevation=saltdata$County\*saltdata$Elevation

lm4=lm(formula = logPrice ˜ Elevation + Sewer + Date + Flood + Distance + Distance\_square + County\_Elevation, data = saltdata) summary(lm4)

* How to remove a potential outlier and re-analyze:
  + E.g. Let’s try without observation no 2 and 10:
* Remove and copy-paste:

saltdata\_red=saltdata[-c(2,10),]

* Check:

dim(saltdata) dim(saltdata\_red) row.names(saltdata\_red)

lm3\_red=lm(formula = logPrice ˜ Elevation + Sewer + Date + Flood + Distance, data =

saltdata\_red)

summary(lm3\_red)

par(mfrow=c(3,2))

plot(lm3\_red,1:6)

1. **All Greens Franchise**

The main goal of this research is to analyse how different factors affect the sales at All Greens. Some inter-factor relations will be analysed as well on more detail.

To import data in SAS

**proc import**

datafile="\\ncfile05\care\_analytics\_dev\Users\Vignesh\learning\AS\Gr een\_fra.csv.csv"

out=green Dbms=csv replace;

getnames=yes;

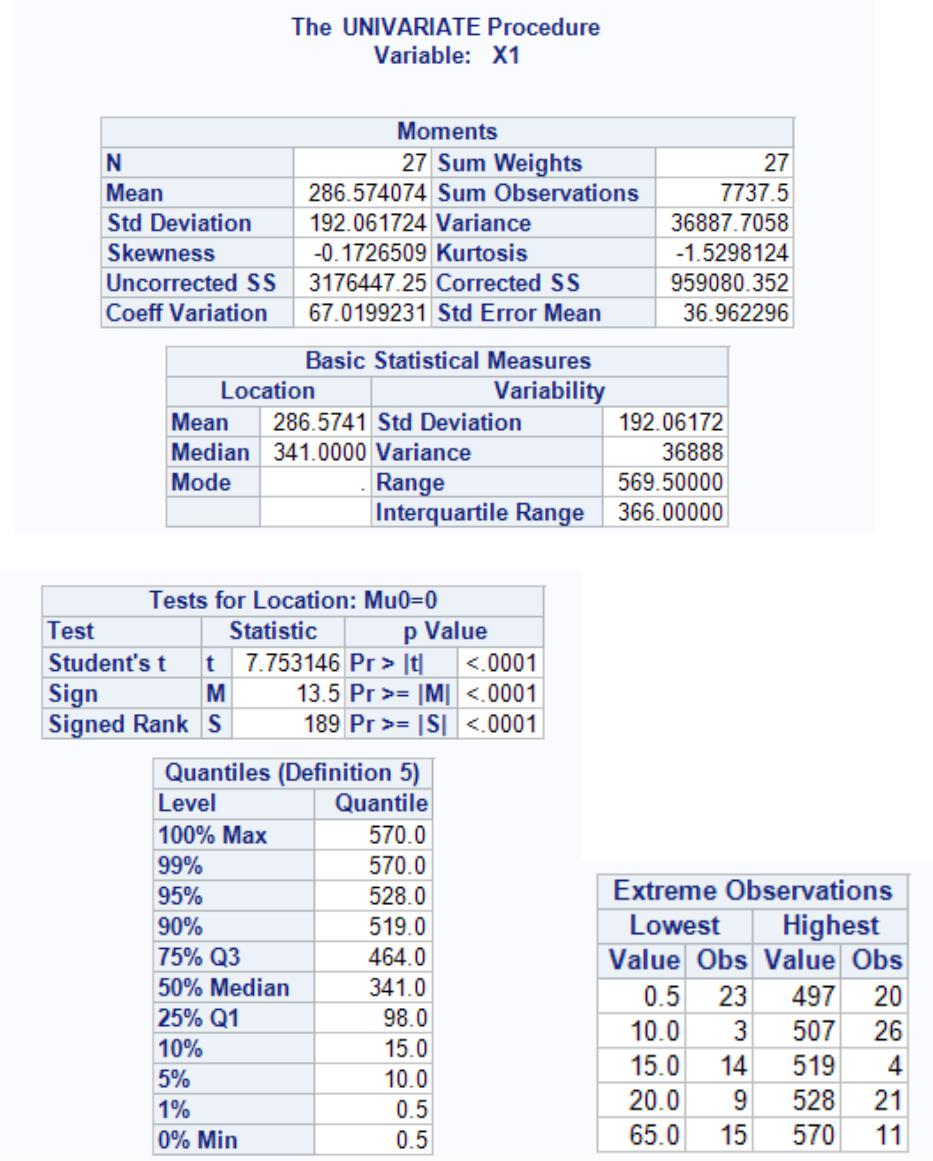
**run**;

First, we need to delete outliers (extreme data points) since they might skew the analysis. To do this, we use summary statistics for the sales variable. It’s obvious that at least two data points are too far from other values: sales = 0.5 and sales = 570. Let’s delete them.

/\* Calculate summary statistics. \*/ **proc univariate** data= green;

var x1;

**run**;



/\* Delete outliers. \*/

**Data** green1;

Set green;

if sales = **0.5** or sales = **570** then delete; **run**;

/\* Now let’s look at some of the scatter plots which show how sales are influenced by other factors\*/

**proc sgplot** data= green1;

scatter x = X2 y = X1;

**run**;

**proc sgplot** data= A1;

scatter x = X3 y = X1;

**run**;

**proc sgplot** data= A1;

scatter x = X4 y = X1;

**run**;

**proc sgplot** data= A1;

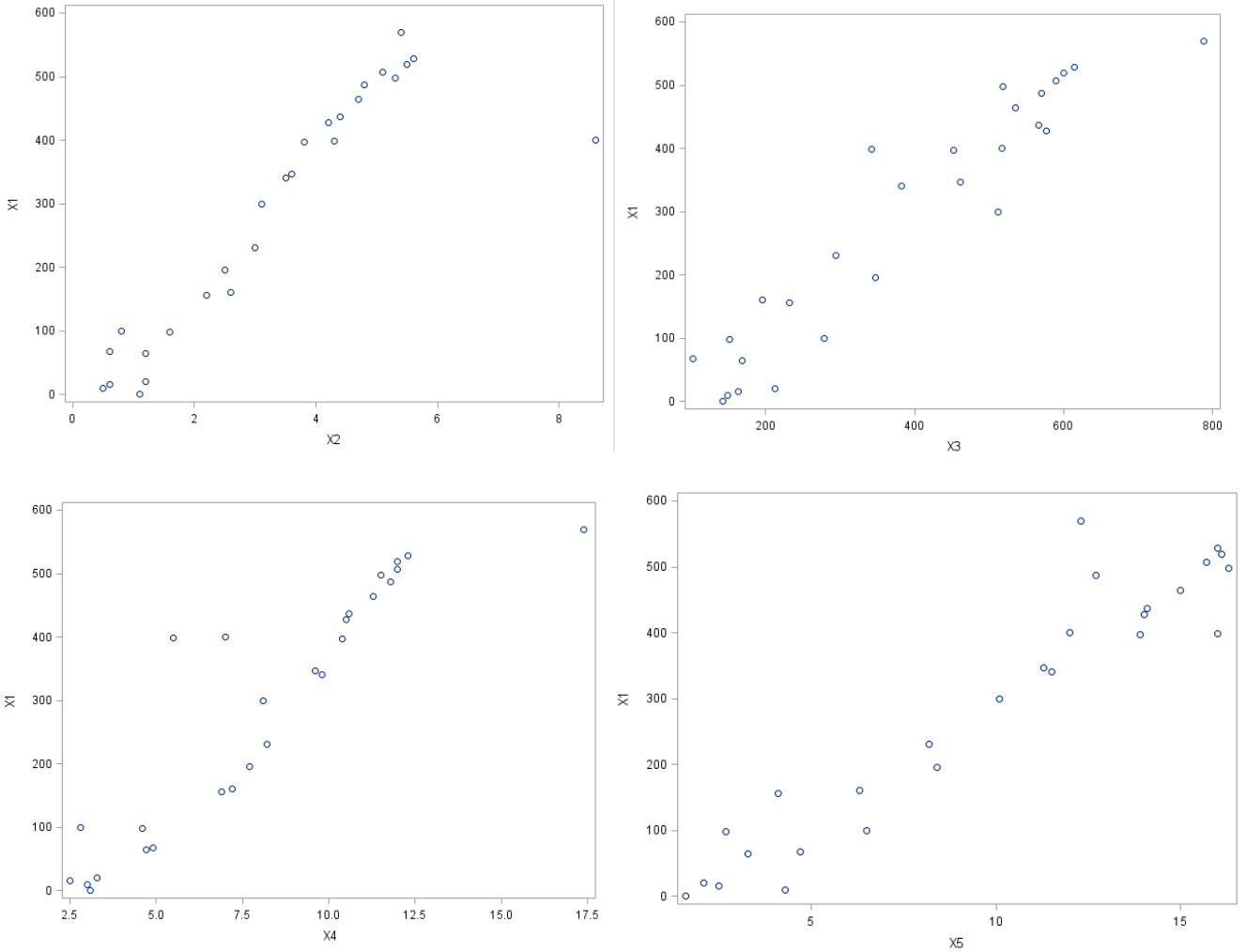
scatter x = X5 y = X1;

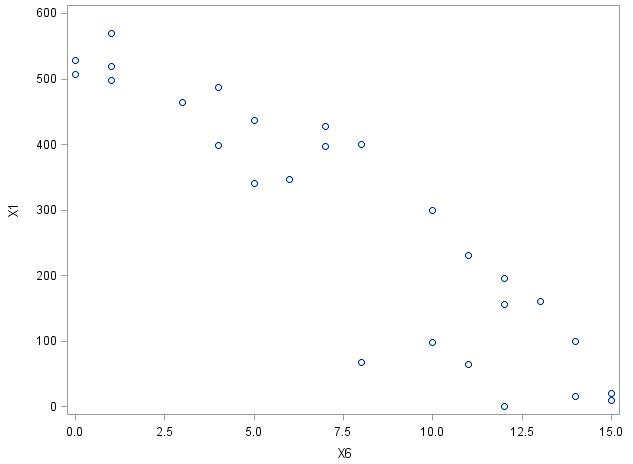
**run**;

**proc sgplot** data= A1;

scatter x = X6 y = X1;

**run**;





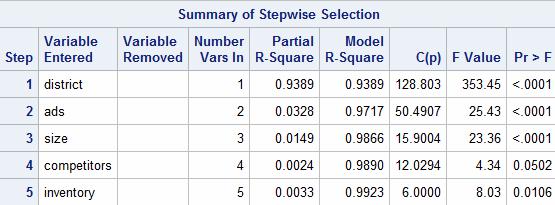
All these plots clearly indicate a linear relation, either direct (for instance, an increase in the money spent on advertising leads to an increase in sales) or opposite (an increase in competitors leads to a decrease in sales).

Since all relations seem to be linear, the obvious choice here is to use a linear multiple regression. To do that, we need an equation that will best describe how sales are affected by different factors. Since we want a linear relation, we’ll use y = a \* x1 + b \* x2 + c \* x3… where a, b, c are coefficients describing each factor’s contribution (how much sales change with every unit increase in the factor); x1, x2, x3 are predicted values of the factors itself; and y is a predicted value of sales.

How do we find the best fitted model (i.e. the model that would have the best prediction results)? It might be that we shouldn’t use all the factors and only need to use two or three of them. To find out, we’ll do the stepwise regression in SAS. What it does is it adds the factors into the model one by one and checks if the new model can predict significantly more results than the old one. The coefficient that describes how well we can predict our data based on the model is called R-squared (sometimes adjusted R-squared is used which is always less or equal than R-squared). R-squared values range from 0 to 1 where 0 means the model is a bad fit and 1 means the model is a perfect fit. So if R-squared increases significantly with adding an additional factor, it means that the factor should stay in the equation.

We have to be careful interpreting the results. R-squared cannot be used as a bullet-proof indicator, especially if the model turns out to be nonlinear. That’s why we analyzed the scatter plots first, to see what we’re dealing with.

Now let’s finally do the regression. Here are the results



As we can see, SAS decided to leave all five factors since they’re all significant at the 0.05 level (as indicated by the last column which shows p-values). Partial R-squared represents

how much information is contributed to the model by the factor if all of them are considered. It’s clear that the most contribution (93.89%) is from the district size, i.e. the more people live in the area, the bigger the sales. Next go the money spent on advertising (3.28%). Three other factors don’t contribute a lot (1.49% for the size of the store, 0.24% for competitors, and 0.33% for inventory). Notice, however, that model R-squared values (that we’ll get if we only use one factor at a time) are all extremely high meaning that all factors affect sales a lot on their own.

Now let’s look at the residuals (residuals represent the difference between the observed values and the predicted values):

/\* Do a stepwise multiple linear regression. \*/

**proc reg** data= green1;

model X1 = X2 X3 X4 X5 X6 / selection = stepwise; output residual = res predicted = pre out = B; plot res. \* pre.;

**run**;

/\* Draw a residuals plot. \*/

**proc rank** normal= blomdata= B;

var res;

ranks nr;

**run**;

/\* Draw a normal residuals plot. \*/ **proc gplot**;

plot res \* nr;

**quit**;

TO interception giving name to the factors

**X1-Annual net sales**

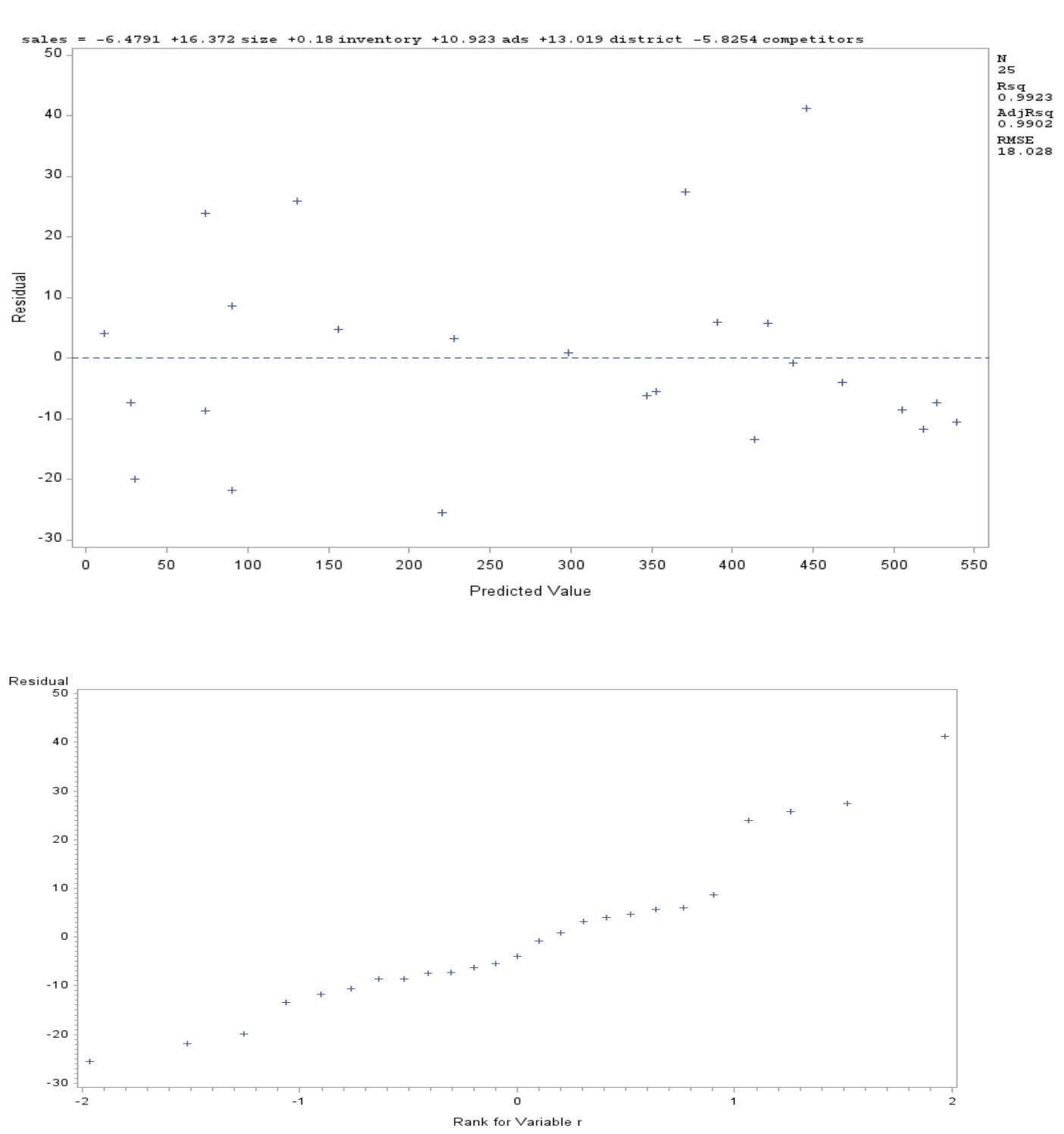
**X2-Size of the store**

**X3-Inventory**

**X4-Money spent on advertising**

**X5-Number of competitors in the area**

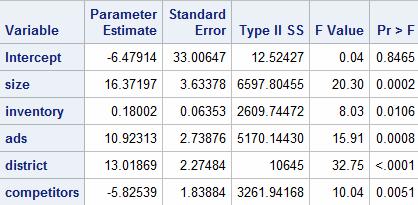
**X6-Competitors**



The first plot shows that the assumption of homoscedasticity (constant variance) is not violated -- the residuals exhibit random scatter behavior and don’t seem to either increase or decrease in magnitude over time.

The second plot shows that the residuals distribution follows a normal pattern (close to a straight line). Since we removed a couple outliers, the plot isn’t skewed too much.

All of that shows that the model is a good fit for our data. Let’s look at the resulting coefficients on explanatory variables:



So the equation is y = -6.47914 + 16.37197 \* size + 0.18002 \* inventory + 10.92313 \* ads + 13.01869 \* district - 5.82539 \* competitors.

We can now use it to predict y based on the values of size, inventory, ads, district, and competitors.

**Findings:**

Although this particular data set is perfect for using linear regression, often real-life data is nonlinear. There are several workarounds for this problem one of them is to transform variables in a way that, upon transformation, will yield a linear relation. For example, a lot of financial data follows an exponential shape, and we can easily find a fitting transformation.