STAT452/652 Exercise Solution to Lecture 2c

1 Lecture 2C: Multiple Linear Regression

1.1 Question 1

```
### Activate data and extract the columns we want
data("airquality")
data = airquality[,1:4]

### Make a scatterplot matrix
pairs(data)
```

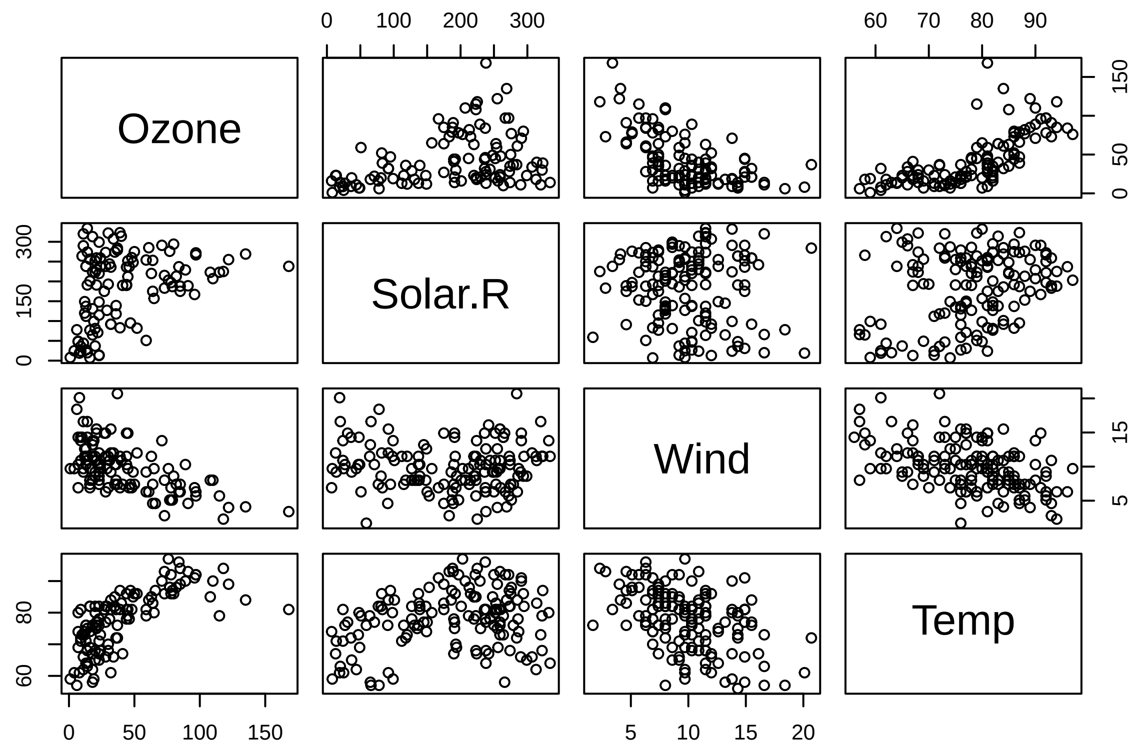

Figure 1: Scatterplot matrix for air quality data

We see in Figure 1 that temperature has a strong increasing relationsip with ozone with a few strong outliers, and is somewhat non-linear. Wind has a weaker decreasing relationship with ozone, and also looks somewhat non-linear. Solar radiation has an increasing relationship with ozone which appears linear, but is strongly heteroscedastic.

Again from Figure 1, we see that wind and temperature have a weak-moderate decreasing linear relationship. The other pairs of explanatory variables have very weak relationships. The only other notable feature is that there appears to be some heteroscedasticity in the relationship between temperature and solar radiation.

1.2 Question 2

```r
### Fit regression models
fit.solar = lm(Ozone ~ Solar.R, data=data)
fit.wind = lm(Ozone ~ Wind, data=data)
fit.temp = lm(Ozone ~ Temp, data=data)

### Create a function to extract slopes and t-values
get.reg.summary = function(fit){
  fit.detail = summary(fit)
  info.table = fit.detail$coef
  output = info.table[2,c(1,3)]
  return(output)
}

### Create container for slopes and t-values
model.output = array(0, dim = c(3,2))
colnames(model.output) = c("Slope", "T-Value")
rownames(model.output) = c("Solar.R", "Wind", "Temp")

### Extract model output
model.output[1,] = round(get.reg.summary(fit.solar), 2)
model.output[2,] = round(get.reg.summary(fit.wind), 2)
model.output[3,] = round(get.reg.summary(fit.temp), 2)

print(model.output)
```

```
##         Slope T-Value
## Solar.R  0.13    3.88
## Wind    -5.55   -8.04
## Temp     2.43   10.42
```

```r
### Set all three plots side-by-side
### I.e. Create a plot grid with 1 row and 3 columns
par(mfrow = c(1,3))

### Plot bi-variate scatterplots and add regression lines
with(data, plot(Solar.R, Ozone, main = "Solar Radiation"))
abline(fit.solar)
with(data, plot(Wind, Ozone, main = "Wind Speed"))
abline(fit.wind)
with(data, plot(Temp, Ozone, main = "Temperature"))
abline(fit.temp)
```

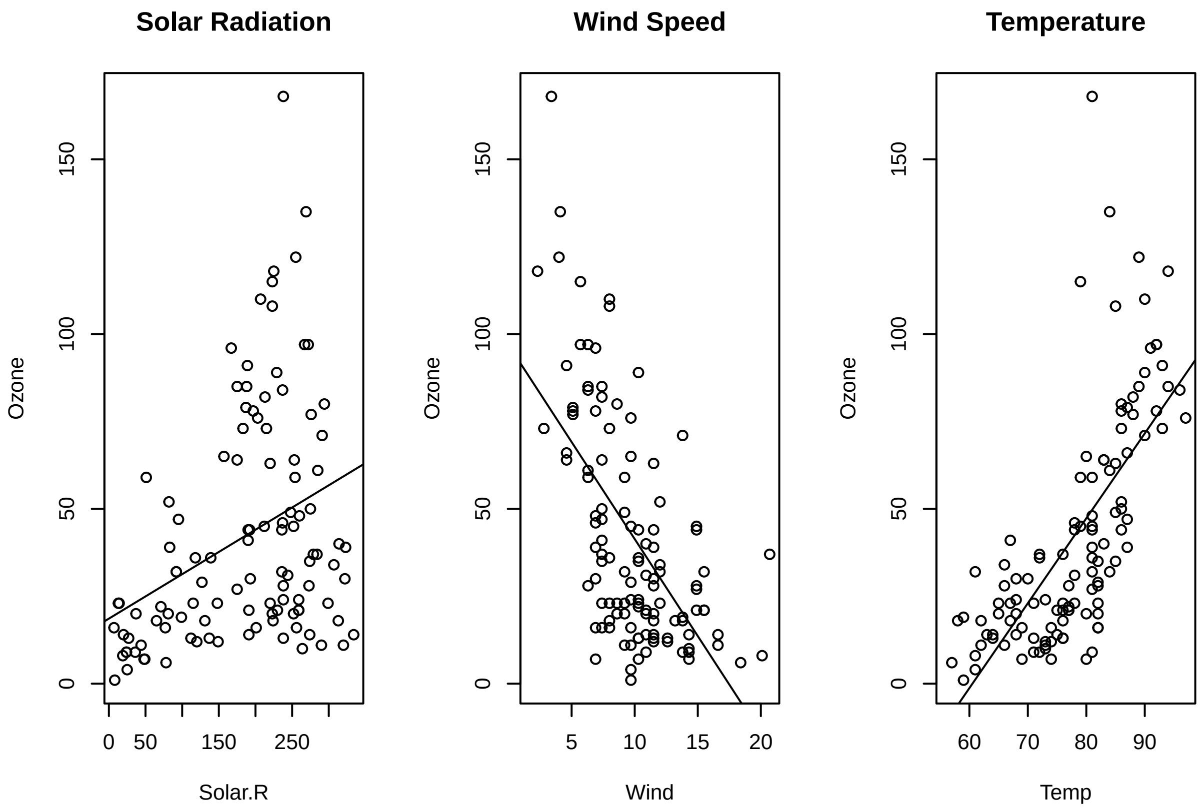

Figure 2: Univariate regression models

```r
### Reset normal plotting
### I.e. Return to a 1x1 plot grid
par(mfrow = c(1,1))
```

See Figure 2 for simple linear regression fits. The line for solar radiation provides a weak-moderate fit to the data, and I would rather that it did a better job for smaller Solar.R values where there is less variability. The line for wind speed provides a moderate-strong fit, although I am a little surprised that it did not fit the large Ozone values better. This is likely because there are so few points with Ozone values above, say, 100. The line for temperature provides a strong fit to the data, other than a few outliers with very large Ozone values.

1.3 Question 3

```r
### Suppress output from this code chunk because we took a screenshot
library(rgl)
open3d()
with(data, plot3d(Ozone ~ Temp + Wind))
```

1.4 Question 4

```r
### Fit multiple linear regression
fit2 = lm(Ozone ~ Temp + Wind, data=data)

### Extract and print slope/t-values
info2 = summary(fit2)
coef2 = info2$coef
table2 = coef2[-1,c(1,3)]
print(table2)
```

```
##       Estimate   t value
## Temp  1.840179  7.361793
## Wind -3.055491 -4.606844
```

Comparing the above table to the one from Question 2, we see that the two variables' effects have the same direction, and both slopes are still significantly different from zero, so little has changed.

```r
### Suppress output from this code chunk because we took a screenshot

### Construct a grid in Temp and Wind
vals.wind = seq(from = 0, to = 22, by = 1)
vals.temp = seq(from = 54, to = 98, by = 2)
pred.grid = expand.grid(Wind = vals.wind, Temp = vals.temp)

### Get fitted values on our grid
pred.ozone = predict(fit2, newdata = pred.grid)

### Plot fitted surface with scatterplot
open3d()
```

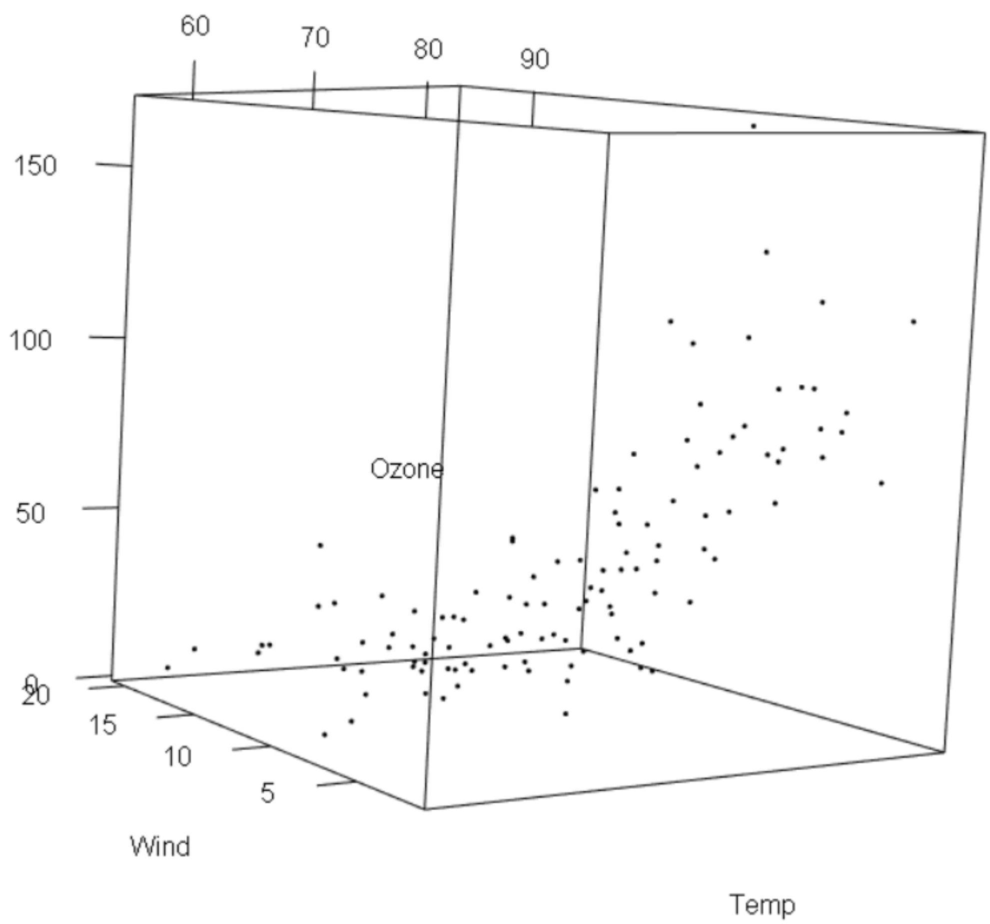

Figure 3: Scatterplot of ozone against temperature and wind

```
persp3d(x = vals.wind, y = vals.temp, z = pred.ozone, col = "orange")
with(data, points3d(Ozone ~ Wind + Temp))
```

From Figure 4, we see that, while our regression model if fairly close to most points, there does appear to be some curvature in the data that is not captured by the linear model.

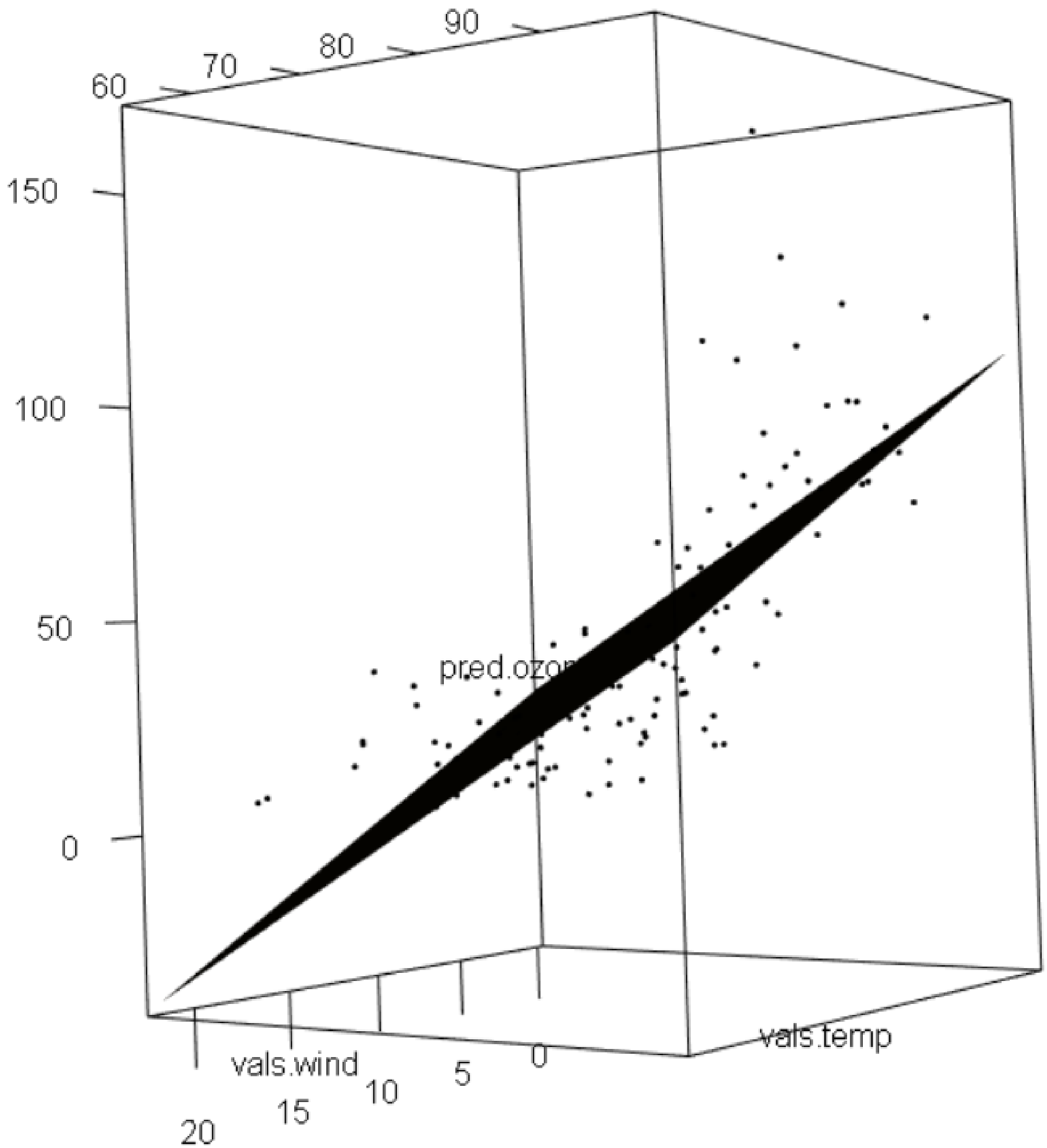

Figure 4: Regression surface of ozone on temperature and wind