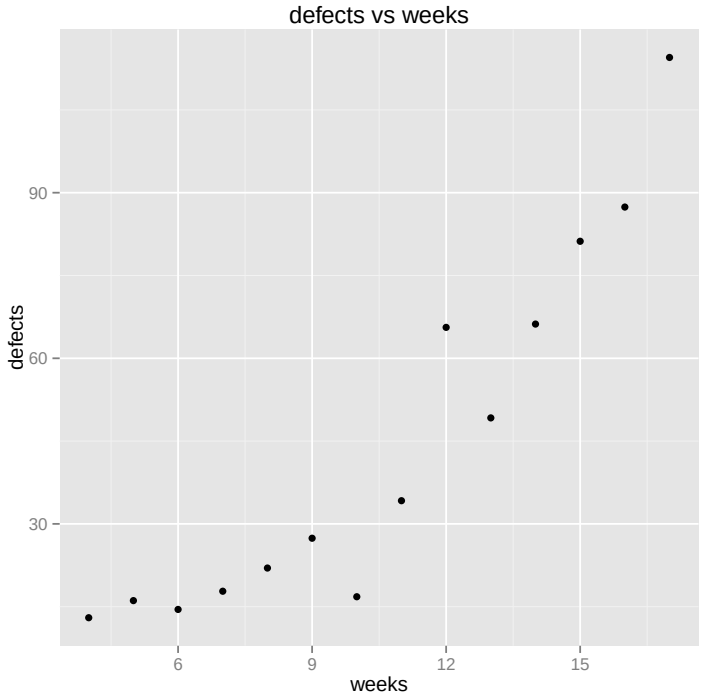
Question 1: Exercise 5.5

A:

```
> library(ggplot2)
> df<-read.csv("ex.5.5.csv")
> plot1<-qplot(weeks,defects,data=df)+ggtitle("defects vs weeks")
> print(plot1)
```



Looking at the basic scatterplot data, there is an apparent curve to the relationship. The model might not be linear.

```
> library(xtable)
> fit1<-lm(defects~weeks,data=df)
> print(xtable(summary(fit1)))
```

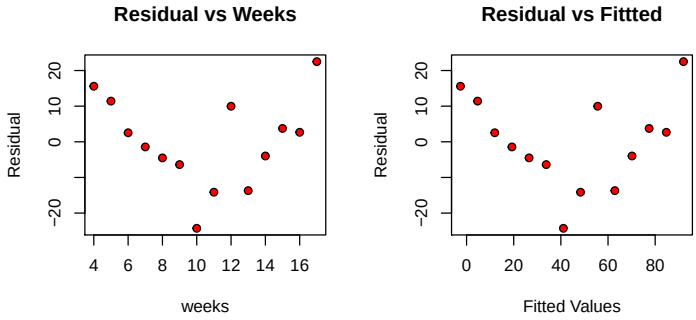
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-31.6982	9.7758	-3.24	0.0071
weeks	7.2767	0.8692	8.37	0.0000

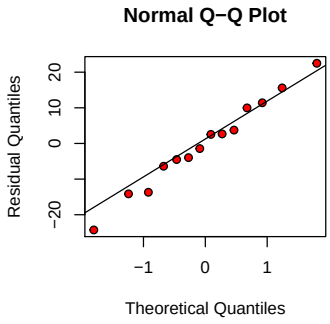
```
>
```

The model becomes defects=-31.6982+7.2767weeks. Both the intercept and the "weeks" predictor are statistically significant. The overall regression is also statistically significant with a p-value very close to zero.

Test for Model Adequacy

```
> attach(df)
> res.lm1<-residuals(fit1)
> par(mfrow = c(2,2))
> plot(weeks,res.lm1,pch = 21, bg='red',xlab='weeks',ylab ='Residual',
+      main="Residual vs Weeks")
> haty.lm1 = predict(fit1)
> plot(haty.lm1,res.lm1,pch = 21, bg='red',xlab='Fitted Values',ylab ='Residual',
+      main="Residual vs Fittted")
> qqnorm(res.lm1,pch = 21, bg='red',ylab ='Residual Quantiles')
> qqline(res.lm1)
> detach(df)
```





From the Normal Q-Q plot we can see tha the normality assumption is satisfied. However, from the Residual vs Fitted plot we can clearly see the variance is not constant. To remedy this I suggest using a natural log transformation on the response variable, defects.

```
> ## perform transformation
> df2<-df
```

```
> df2$defects<-log(df2$defects)
> ## fit model
> fit2<-lm(defects~weeks,data=df2)
> print(xtable(summary(fit2)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.7162	0.1731	9.91	0.0000
weeks	0.1735	0.0154	11.27	0.0000

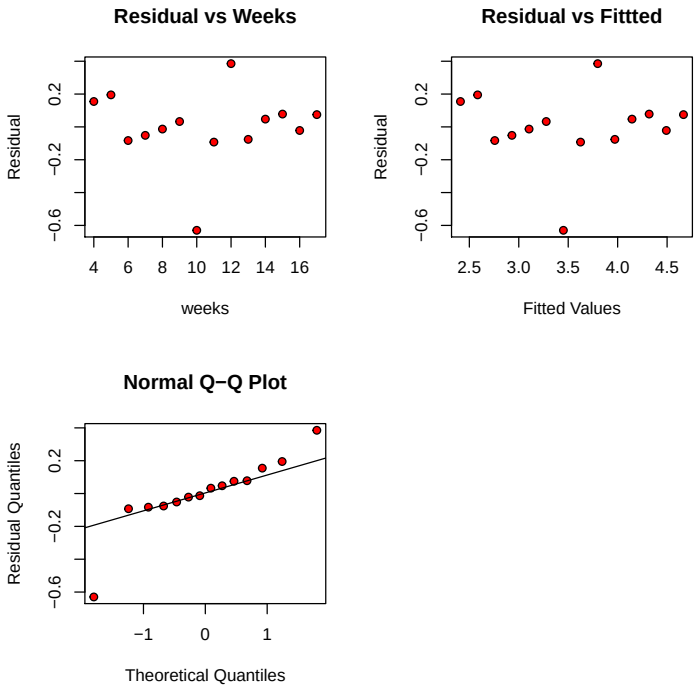
```
>
```

The new model becomes ln(defects)=1.71622+0.17351*weeks.

The model is significant in both the intercept and the weeks predictor. In addition, the model is significant overall with an r-squared if 0.9137 and a p-value of close to zero.

Test for Model Adequacy

```
> attach(df2)
> res.lm1<-residuals(fit2)
> par(mfrow = c(2,2))
> plot(weeks,res.lm1,pch = 21, bg='red',xlab='weeks',ylab ='Residual',
+      main="Residual vs Weeks")
> haty.lm1 = predict(fit2)
> plot(haty.lm1,res.lm1,pch = 21, bg='red',xlab='Fitted Values',ylab ='Residual',
+      main="Residual vs Fittted")
> qqnorm(res.lm1,pch = 21, bg='red',ylab ='Residual Quantiles')
> qqline(res.lm1)
> detach(df2)
```



From the new models, residual plot we can see that the non-constant variance assumption looks to be more appropriate than before.

Question 2: Exercise 6.3

```
> df3<-read.csv("belle.csv")
> fit3<-lm(y~x1+x2+x3+x4+x5+x6+x7,data=df3)
> inflm<-influence.measures(fit3)
> table<-summary(inflm)
```

Next page has the influential measures table.

```
> print(xtable(table))
```

	dfb.1_	dfb.x1	dfb.x2	dfb.x3	dfb.x4	dfb.x5	dfb.x6	dfb.x7	dffit	cov.r	cook.d	hat
8	0.05	0.18	-0.08	0.17	-0.53	0.04	-0.03	-0.30	-0.80	2.28	0.08	0.49
9	-0.69	1.03	0.49	-0.44	0.68	-0.07	0.44	0.18	1.66	0.95	0.31	0.49
14	-0.02	0.03	0.02	-0.02	-0.01	7406.84	-5894.03	-0.03	-7929.09	10154446286.81	8291031.75	1.00
19	-0.20	0.23	0.21	-0.20	0.21	0.01	0.18	-1.35	-1.56	1.28	0.29	0.52
23	0.10	-0.02	-0.12	0.10	-0.06	0.01	0.06	0.01	-0.14	2.33	0.00	0.35
26	-0.44	0.75	-0.05	0.47	-0.75	0.43	0.47	0.50	2.02	0.11	0.37	0.33

This table provides a summary of all the observations R considers most likely to be influential. As we can see observation 14 is most obviosuly an influential outlier. Looking at the cov.r and Cook's D values we can see that the point is extremely influential. It particularly strongly affects x5 and x6, we see this by looking at the dfb.x5 and dfb.x6 values.

Question 4

```
> library(MASS);
> df4<-read.csv("mort.csv")
> names(df4)<-c("city","mort","precip","educ","nonwhite","nox","so2")
```

A: Backward Selection

```
> fit<-lm(mort~precip+educ+nonwhite+nox+so2,data=df4)
> drop1(fit,test="F")

Single term deletions

Model:
mort ~ precip + educ + nonwhite + nox + so2
         Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>                 74289 439.28
precip    1      5737  80026 441.75  4.1704 0.0460320 *
educ      1      6103  80392 442.02  4.4362 0.0398487 *
nonwhite  1     36356 110645 461.18 26.4266 3.887e-06 ***
nox       1       880  75169 437.99  0.6394 0.4274258
so2       1     20976  95265 452.20 15.2470 0.0002642 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that the nox value is deemed unnecessary, thus we delete it.

```
> fit <- update(fit,.~.-nox)
```

Now to see if we need to drop another variable.

```
> drop1(fit, test='F')

Single term deletions

Model:
mort ~ precip + educ + nonwhite + so2
         Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>                 75169 437.99
precip    1      9339  84507 443.02  6.8330 0.0115217 *
educ      1      6882  82050 441.24  5.0352 0.0288830 *
nonwhite  1     35549 110718 459.22 26.0110 4.331e-06 ***
so2       1     21010  96179 450.78 15.3728 0.0002468 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All the other variables are clearly significant, as such we can end the algorythm here. The final model is mort precip+educ+nonwhite+so2

A: Forward Selection

```
> fit_f <- lm(mort~1,data=df4)
> add1(fit_f,mort~precip+educ+nonwhite+nox+so2,test='F')

Single term additions

Model:
mort ~ 1
         Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>                228275 496.64
precip    1     59256 169019 480.61 20.3342 3.216e-05 ***
educ      1     59604 168672 480.48 20.4955 3.022e-05 ***
nonwhite  1     95705 132571 466.03 41.8710 2.256e-08 ***
nox       1      1308 226967 498.29  0.3344 0.5653434
so2       1     41417 186858 486.63 12.8558 0.0006908 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can that the variable "nonwhite" is most significant, thus we add it to the model.

```
> fit_f <- update(fit_f,.~.+nonwhite)
```

See if we can add any more variables

```
> add1(fit_f,mort~precip+educ+nonwhite+nox+so2,test='F')

Single term additions

Model:
mort ~ nonwhite
       Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>              132571 466.03
precip  1     16826 115745 459.89  8.2861 0.0056155 **
educ    1     33677  98894 450.45 19.4105 4.707e-05 ***
nox     1      1942 130629 467.15  0.8474 0.3611623
so2     1     24165 108405 455.96 12.7062 0.0007456 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Educ is the next most significant variable, we add it as well.

```
> fit_f <- update(fit_f,.~.+educ)
```

See if more variables are needed

```
> add1(fit_f,mort~precip+educ+nonwhite+nox+so2,test='F')
```

```
Single term additions

Model:
mort ~ nonwhite + educ
       Df Sum of Sq   RSS    AIC F value   Pr(>F)
<none>              98894 450.45
precip  1    2715.1 96179 450.78  1.5809 0.213853
nox     1       0.0 98894 452.45  0.0000 0.998073
so2     1   14386.5 84507 443.02  9.5334 0.003136 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that "so2" has a p-value less than 0.1, thus we add it to our model.

```
> fit_f <- update(fit_f,.~.+so2)
```

see if more variables are needed.

```
> add1(fit_f,mort~precip+educ+nonwhite+nox+so2,test='F')

Single term additions

Model:
mort ~ nonwhite + educ + so2
       Df Sum of Sq   RSS    AIC F value  Pr(>F)
<none>              84507 443.02
precip  1    9338.7 75169 437.99  6.8330 0.01152 *
nox     1    4481.0 80026 441.75  3.0797 0.08484 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

"precip" is also significant and thus, added.

```
> fit_f <- update(fit_f,.~.+precip)
```

Now, to see wether it worth adding nox to the model.

```
> add1(fit_f,mort~precip+educ+nonwhite+nox+so2,test='F')

Single term additions

Model:
mort ~ nonwhite + educ + so2 + precip
       Df Sum of Sq   RSS    AIC F value Pr(>F)
<none>              75169 437.99
nox     1    879.66 74289 439.28  0.6394 0.4274
```

As we can see from the table the p value is larger than 0.1, thus we do not add it to our model.

A: Stepwise

```
> fit_null<-lm(mort~precip+educ+nonwhite+nox+so2,data=df4)
> step3<-stepAIC(fit_null, direction="both")

Start:  AIC=439.28
mort ~ precip + educ + nonwhite + nox + so2

           Df Sum of Sq    RSS    AIC
- nox       1       880  75169 437.99
<none>                   74289 439.28
- precip    1      5737  80026 441.75
- educ      1      6103  80392 442.02
- so2       1     20976  95265 452.20
- nonwhite  1     36356 110645 461.18

Step:  AIC=437.99
mort ~ precip + educ + nonwhite + so2

           Df Sum of Sq    RSS    AIC
<none>                   75169 437.99
+ nox       1       880  74289 439.28
- educ      1      6882  82050 441.24
- precip    1      9339  84507 443.02
- so2       1     21010  96179 450.78
- nonwhite  1     35549 110718 459.22

>
```

We can see that through stepwise selection, we still only exclude the "nox" variable.

```
> step3$anova

Stepwise Model Path
Analysis of Deviance Table

Initial Model:
mort ~ precip + educ + nonwhite + nox + so2

Final Model:
mort ~ precip + educ + nonwhite + so2


   Step Df Deviance Resid. Df Resid. Dev      AIC
1                          54   74289.05 439.2825
2 - nox  1 879.6599        55   75168.71 437.9887
```

As we can see that the nox variable is not significant, as such we end our algorythm here. The final model is mort precip+educ+nonwhite+so2. All three stepwise selection methods resulted in the same model.

B: All Subset Selection

```
> library(leaps);library(car);
> leaps<-regsubsets(mort~precip+educ+nonwhite+nox+so2,data=df4,nbest=1)
> summary(leaps)

Subset selection object
Call: regsubsets.formula(mort ~ precip + educ + nonwhite + nox + so2,
    data = df4, nbest = 1)
5 Variables  (and intercept)
         Forced in Forced out
precip       FALSE      FALSE
educ         FALSE      FALSE
nonwhite     FALSE      FALSE
nox          FALSE      FALSE
so2          FALSE      FALSE
1 subsets of each size up to 5
Selection Algorithm: exhaustive
         precip educ nonwhite nox so2
1  ( 1 ) " "    " "  "*"      " " " "
2  ( 1 ) " "    "*"  "*"      " " " "
3  ( 1 ) "*"    " "  "*"      " " "*"
4  ( 1 ) "*"    "*"  "*"      " " "*"
5  ( 1 ) "*"    "*"  "*"      "*" "*"
```

Now let us see which how good the variables are compared to one another.

```
> par(mfrow = c(2,2))
> plot(leaps,scale="r2",main="R2 vs Vars")
> plot(leaps,scale="bic",main="BIC vs Vars")
> plot(leaps,scale="adjr2",main="Adj.R2 vs Vars")
> plot(leaps,scale="Cp",main="Cp vs Vars")

> layout(matrix(1:4, ncol = 2))
> res.legend <-subsets(leaps, statistic="rsq",
+                      legend = FALSE, min.size = 1, main = "R^2")
> ## Adjusted R2
> res.legend <-subsets(leaps, statistic="adjr2",
+                      legend = FALSE, min.size = 1, main = "Adjusted R^2")
> ## Mallow Cp
> res.legend <-subsets(leaps, statistic="cp",
+                      legend = FALSE, min.size = 1, main = "Mallow Cp")
> abline(a = 1, b = 1, lty = 2)
```

Looking at the four variables we can see the ideal models for all subsections are

```
if we use r2; it is: mort~precip+educ+nonwhite+nox+so2
if we use adj.R2 it is: mort~precip+educ+nonwhite+so2
if we use mallow's Cp it is:mort~precip+educ+nonwhite+so2
```

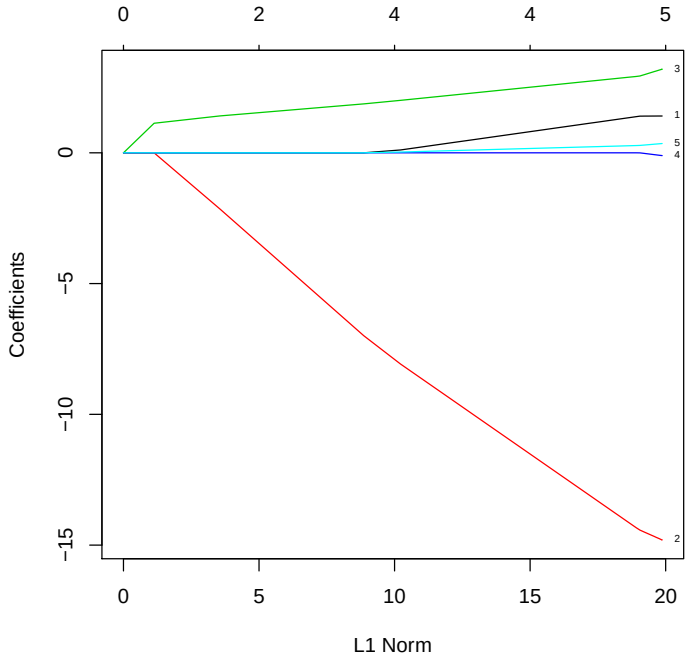
Unsuprisingly R2 is not a reliable criterion for picking models as it does not penalize the addition of more variables. The ideal adj.R2 model and mallows Cp models are identical.

C: LASSO

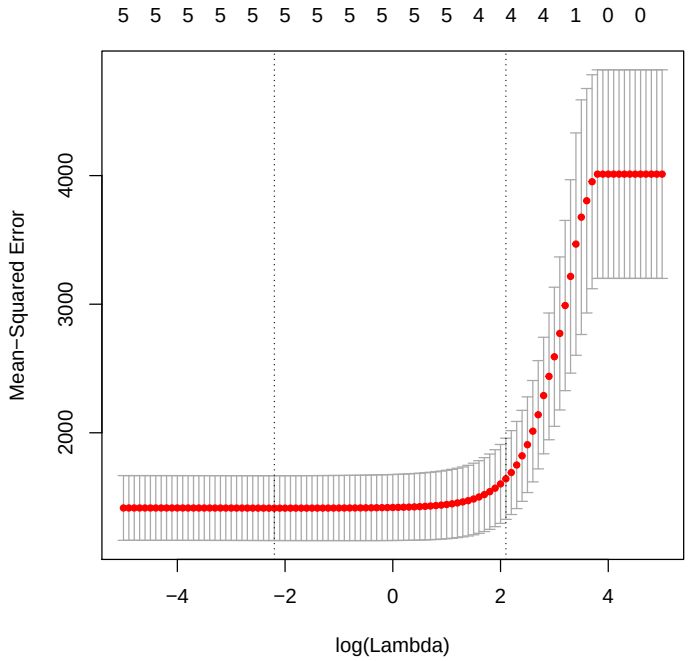
```
> library(glmnet);
> x<-X <- as.matrix(df4[,3:7])
> y<-df4$mort
> lambdas<-exp(seq(-5,5,by=0.1))
> eg.lm1 <- glmnet(x,y,lambda=lambdas)
> par(mfrow<-c(1,2))

NULL

> plot(eg.lm1,label=TRUE)
> plot(eg.lm1,xvar="lambda",label=TRUE)
```




```
> eg.cv <- cv.glmnet(x,y,lambda=lambdas, nfolds=4)
> plot(eg.cv)
```




```
> ss<-c(eg.cv$lambda.min,eg.cv$lambda.1se)
> ## find coefficients
> coef(eg.cv,s=c(eg.cv$lambda.min,eg.cv$lambda.1se))

6 x 2 sparse Matrix of class "dgCMatrix"
                      1           2
(Intercept) 995.7697906 997.3686540
precip        1.4066665   1.0142727
educ        -14.7959992 -12.5248063
nonwhite      3.1897515   2.6551196
nox          -0.1039744   .
so2           0.3524277   0.2056053
```

From the above result we see that nox is not a useful variable. Thus our final model includes all variables except nox.

This is the same result we had in each or variable selection methods, excep all subset r2, which is not a good selection method to begin with. As such our ideal model is:

```
> final_model<-lm(mort~precip+educ+nonwhite+so2,data=df4)
> print(xtable(summary(final_model)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	995.8224	91.3398	10.90	0.0000
precip	1.6350	0.6255	2.61	0.0115
educ	-15.5697	6.9386	-2.24	0.0289
nonwhite	3.0998	0.6078	5.10	0.0000
so2	0.3263	0.0832	3.92	0.0002