# Ve406 Lecture 14

Jing Liu

UM-SJTU Joint Institute

July 4, 2018

- Notice we have discussed a few important issues:

1. Multicollinearity
2. Outliers, high leverage and influential points

- However, regarding the initial assumptions:

1. The conditional mean of the response is given by

$$\mathbb{E}\left[Y_i \mid X_{i1}, X_{i2}, \ldots, X_{ik}\right] = \mathbb{E}\left[Y_i \mid \mathbf{X}_i\right] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$$

2. The errors have zero mean and constant variance

$$\mathbb{E}\left[\varepsilon_i \mid \mathbf{X}_i\right] = 0 \quad \text{and} \quad \mathrm{Var}\left[\varepsilon_i \mid \mathbf{X}_i\right] = \sigma^2 \qquad \text{where} \quad \varepsilon_i = Y_i - \mathbb{E}\left[Y_i \mid \mathbf{X}_i\right]$$

3. The errors are independent of $\mathbf{X}_i$, and of each other.
4. The errors follow the normal distribution of $\mathrm{N}\left(0, \sigma^2\right)$.

- We have essentially only addressed the first of the above assumptions.

- Consider the following simple example to understand

heteroskedasticity

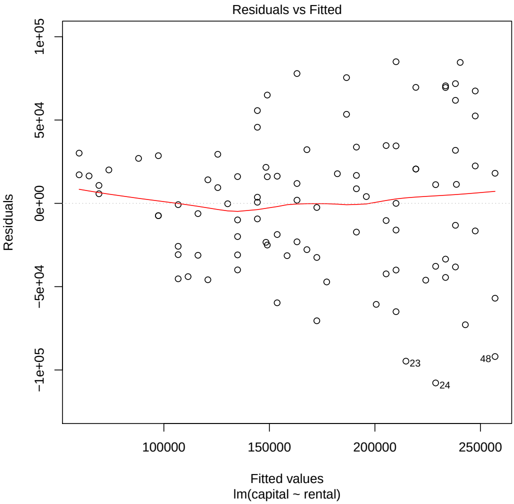
which is the technical term for having unequal/non-constant error variance.

- The data is about housing price, it was collected in hope of predicting

the `capital` value from the `rental` value

```
> cvrv.df =
+   read.table("~/Desktop/cvrv.csv",
+              sep = ",", header = TRUE)
>
> cvrv.LM = lm(capital~rental, data = cvrv.df)
>
> plot(cvrv.LM, which = 1)
```

for which the error variance is clearly unequal.




- Heteroskedasticity is often caused by the nature of the response variable.

- In the presence of heteroskedasticity, the estimator

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

  is still unbiased and consistent, but is no longer efficient.

- However, the variance of the estimator loses the consistency property as well

$$\begin{aligned}\hat{\mathrm{Var}}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right] &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\hat{\mathrm{Var}}\left[\boldsymbol{\varepsilon} \mid \mathbf{X}\right]\left(\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\right)^{\mathrm{T}} \\ &= \hat{\sigma}^2\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \qquad \text{where} \quad \hat{\sigma}^2 = \frac{1}{n-k-1}\hat{\mathbf{e}}^{\mathrm{T}}\hat{\mathbf{e}}\end{aligned}$$

- This is particularly problematic if the purpose of the model is to explain since

$$t_j = \frac{\hat{\beta}_j}{\mathrm{SE}\left(\hat{\beta}_j\right)}$$

  where $\mathrm{SE}\left(\hat{\beta}_j\right)$ is the $j$th main diagonal element of $\hat{\mathrm{Var}}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right]$.

- Recall we have the following under homoskedasticity

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim \mathrm{N}(0, \sigma^2)$$

and the estimate $\hat{\boldsymbol{\beta}} = \mathbf{b}$ is found by minimising

$$\sum_{i=1}^{n} \left( y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \cdots - \hat{\beta}_k x_{ik} \right)^2 = \hat{\mathbf{e}}^{\mathrm{T}} \hat{\mathbf{e}}$$

which is given by founding the gradient of the following and setting it to $\mathbf{0}$

$$\begin{aligned} f(\mathbf{b}) = (\mathbf{y} - \mathbf{X}\mathbf{b})^{\mathrm{T}} (\mathbf{y} - \mathbf{X}\mathbf{b}) &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + \mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{b} \\ &\Longrightarrow \hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathrm{T}}\mathbf{y} \end{aligned}$$

- We could have written the initial equation differently without affecting $\hat{\boldsymbol{\beta}}$

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \sigma\varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim \mathrm{N}(0, 1)$$

- Now with heteroskedasticity, we have

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \sigma_i \varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim \mathrm{N}(0, 1)$$

- If we scale our variables according to the true values of $\sigma_i$, we have

$$\begin{aligned} \frac{y_i}{\sigma_i} &= \beta_0 \frac{1}{\sigma_i} + \beta_1 \frac{x_{i1}}{\sigma_i} + \cdots + \beta_k \frac{x_{ik}}{\sigma_i} + \varepsilon_i \\ y_i^* &= \beta_0 + \beta_1 x_{i1}^* + \cdots + \beta_k x_{ik}^* + 1 \varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim \mathrm{N}(0, 1) \end{aligned}$$

from which we see all the nice properties will be back if we minimise

$$\begin{aligned} \hat{\mathbf{e}}^{\mathrm{T}} \hat{\mathbf{e}} &= \sum_{i=1}^{n} \left( y_i^* - \hat{\beta}_0 - \hat{\beta}_1 x_{i1}^* - \cdots - \hat{\beta}_k x_{ik}^* \right)^2 \\ &= \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \left( y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \cdots - \hat{\beta}_k x_{ik} \right)^2 \end{aligned}$$

Q: What is the rationale behind this approach?

- Let $\mathbf{W}$ denote the diagonal matrix containing the scaling factor, that is,

$$\operatorname{diag}(\mathbf{W}) = \left\{\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \cdots \frac{1}{\sigma_i^2}, \cdots, \frac{1}{\sigma_n^2}\right\}$$

- then we have the following objective function

$$\begin{aligned} f(\mathbf{b}) &= (\mathbf{y} - \mathbf{X}\mathbf{b})^{\mathrm{T}} \mathbf{W} (\mathbf{y} - \mathbf{X}\mathbf{b}) \\ &= \mathbf{y}^{\mathrm{T}}\mathbf{W}\mathbf{y} - 2\mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{y} + \mathbf{b}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}\mathbf{b} \end{aligned}$$

- Differentiating, we have the following gradient,

$$\nabla f = -2\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{y} + 2\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}\mathbf{b}$$

- Hence the following estimator will have the same nice properties as before

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{y}$$

which is known as the weighted least squares estimator.

- Of course, the true $\sigma_i^2$ are unknown in practice, thus we have estimate

$$\mathbf{W}$$

- Notice the following is true

$$\mathrm{Var}\left[\varepsilon_i \mid \mathbf{X}\right] = \mathbb{E}\left[\left(\varepsilon_i - \mathbb{E}\left[\varepsilon_i \mid \mathbf{X}\right]\right)^2 \mid \mathbf{X}\right] = \mathbb{E}\left[\varepsilon_i^2 \mid \mathbf{X}\right]$$

- Thus one of many possible approaches is to use the following estimator

$$\hat{\mathrm{Var}}\left[\varepsilon_i \mid \mathbf{X}\right] = \mathbb{E}\left[\hat{e}_i^2 \mid \mathbf{X}\right]$$

  which is a conditional mean of a random variable that we have observations for once we have fitted the original regression, thus can be estimated.

- Since $\hat{e}_i^2$ might be really small/large in practice, thus people often work with

$$z_i = 2 \ln |\hat{e}_i|$$

  the log-scale provides extra numerical stability.

- For our early example,

```
> z = 2 * log(abs(cvrv.LM$residuals))

> # Perform the auxiliary regression
> auxiliary.LM = lm(z~rental, data = cvrv.df)
```

- Transform back to obtain the estimated $\sigma_i$

```
> var.vec = exp(auxiliary.LM$fitted.values)
```

- Specify the weights according to the reciprocal of

```
> cvrv.WLS = lm(capital~rental,
+               weights = 1/var.vec, data = cvrv.df)
```

- If we compare the two models, the estimates of the slope are similar, but the standard errors are somewhat different.
- And we residual standard errors are very different as expected.

```
> summary(cvrv.LM)
```

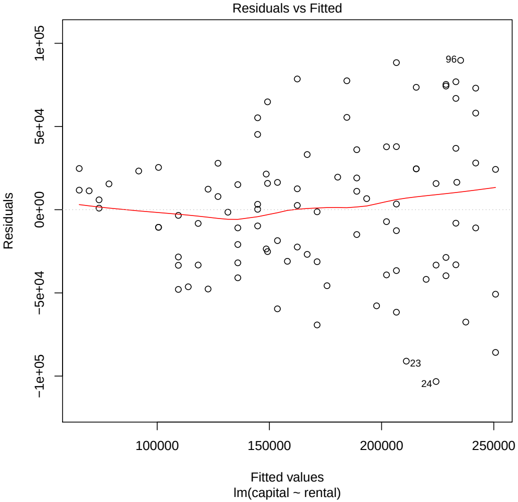
```
Call:
lm(formula = capital ~ rental, data = cvrv.df)

Coefficients:
              Estimate Std. Error t value Pr(>t)
(Intercept) -43372.326  17993.856   -2.41   0.0179 *
rental          22.559      1.822   12.38   <2e-16 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 42450 on 94 degrees of freedom
Multiple R-squared:  0.6199,    Adjusted R-squared:  0.6159
F-statistic: 153.3 on 1 and 94 DF,  p-value: < 2.2e-16
```

```
> summary(cvrv.WLS)
```

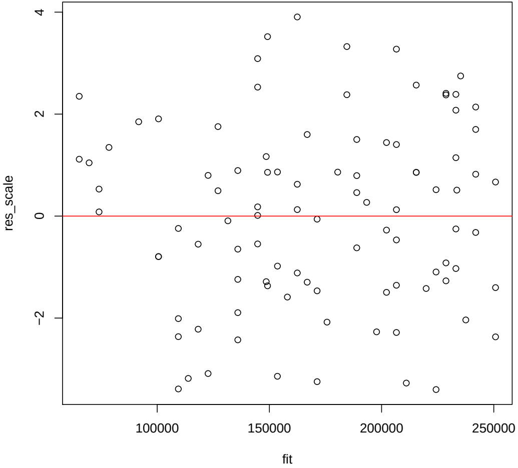
```
Call:
lm(formula = capital ~ rental, data = cvrv.df, weights = 1/var.vec)

Coefficients:
              Estimate Std. Error t value Pr(>t)
(Intercept) -31942.206  12539.229  -2.547   0.0125 *
rental          21.238      1.511  14.055   <2e-16 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 1.804 on 94 degrees of freedom
Multiple R-squared:  0.6776,    Adjusted R-squared:  0.6741
F-statistic: 197.5 on 1 and 94 DF,  p-value: < 2.2e-16
```

- Noticing we didn't remove the heteroskedasticity, we model it.

```
> plot(cvrv.WLS, which = 1)
```



```
> fit = cvrv.WLS$fitted.values
> res_scale = cvrv.WLS$residuals/sqrt(var.vec)
> plot(fit, res_scale); abline(h=0, col = "red")
```



- Consider the following simple example to understand

  lack of independence

- The data is about sales and advertising expenditure

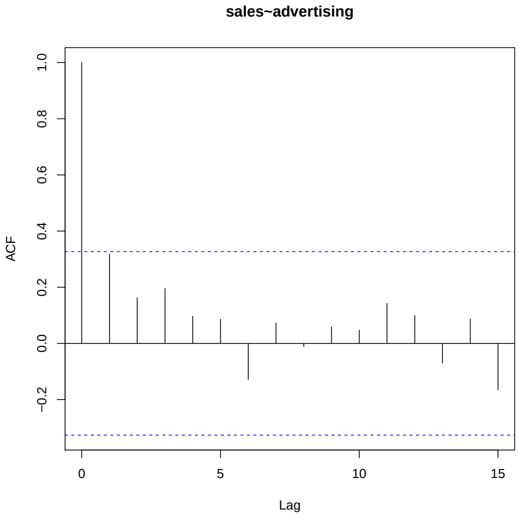
| | |
|---|---|
| sales | Monthly sales of a retailer |
| advertising | Amount spent on advertising this month |

```
> sales.df =
+   read.table("~/Desktop/sales.txt",
+                sep = "", header = TRUE)
>
> sales.LM = lm(sales~advertising, data = sales.df)
```

- Estimated Correlations between Residuals and their Lags

```
> res = sales.LM$residuals
> acf(res, main = "sales~advertising")
```

- If there is no problem, we expect the correlations to be small.




- So it indicates there might be a problem.

- Notice the estimation is done slightly differently

$$\hat{R}(k) = \frac{(n-1)\sum_{i=1}^{n-k}\left(\hat{e}_i - \frac{1}{n}\sum_{i=1}^{n}\hat{e}_i\right)\left(\hat{e}_{i+k} - \frac{1}{n}\sum_{i=1}^{n}\hat{e}_i\right)}{(n-k)\sum_{i=1}^{n}\left(\hat{e}_i - \frac{1}{n}\sum_{i=1}^{n}\hat{e}_i\right)^2}$$

instead of the typical estimation of correlation coefficient

$$r = \frac{\sum_{i=1}^{n}\left(x_i - \frac{1}{n}\sum_{i=1}^{n}x_i\right)\left(y_i - \frac{1}{n}\sum_{i=1}^{n}y_i\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \frac{1}{n}\sum_{i=1}^{n}x_i\right)^2}\sqrt{\sum_{i=1}^{n}\left(y_i - \frac{1}{n}\sum_{i=1}^{n}y_i\right)^2}}$$

- One of possible reasons that errors lack independence is having the following

$$\mathbb{E}\left[Y_i \mid X_1, X_2, \ldots X_n\right] = \beta_0 + \beta_1 x_i + \beta_2 x_{i-1} + \beta_3 x_{i-2} + \cdots$$

which is known as a distributed lag model, instead of the original assumption

$$\mathbb{E}\left[Y_i \mid X_1, X_2, \ldots X_n\right] = \beta_0 + \beta_1 x_i$$

- If the above is the cause, then taking those lags of the predictors into the model will remove correlations in errors, thus satisfy assumption 3..

```
> n = nrow(sales.df)
>
> s_pre.df =
+   data.frame(sales = sales.df$sales[-1],
+              ad = sales.df$advertising[-1],
+              ad1 = sales.df$advertising[-n])
>
> sales_pre.LM = lm(sales~ad+ad1, data = s_pre.df)
```

```
> summary(sales_pre.LM)
```

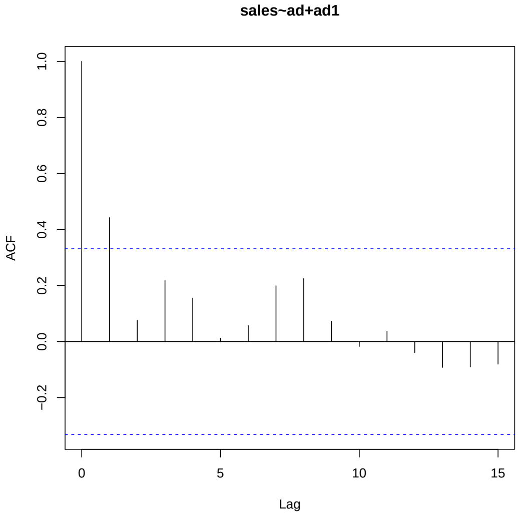
```
Call:
lm(formula = sales ~ ad + ad1, data = s_pre.df)

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) 15.60361    1.34531  11.599 5.35e-13 ***
ad           0.14242    0.03518   4.049 0.000305 ***
ad1          0.16651    0.03606   4.617 6.03e-05 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 3.652 on 32 degrees of freedom
Multiple R-squared:  0.6392,    Adjusted R-squared:  0.6167
F-statistic: 28.35 on 2 and 32 DF,  p-value: 8.233e-08
```

```
> summary(sales_lag2.LM)
```

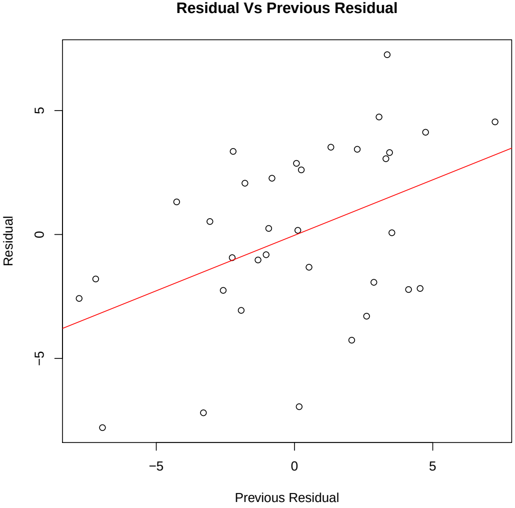
```
Call:
lm(formula = sales ~ ad + ad1 + ad2, data = s_lag2.df)

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) 14.34498    1.56379   9.173 3.28e-10 ***
ad           0.14625    0.03500   4.178 0.000233 ***
ad1          0.14671    0.03791   3.870 0.000545 ***
ad2          0.05836    0.03579   1.630 0.113474
```

- However, it does not solve the lack of independence problem for this dataset

```
> res = sales_pre.LM$residuals
> acf(res, main = "sales~ad+ad1")
```



- There still exists a trend between residual and previous residual



```
> res.now = res[-1]
>
> res.pre = res[-length(res)]
> plot(res.pre, res.now,
+      main = "Residual Vs Previous Residual",
+      ylab = "Residual", xlab = "Previous Residual")
>
> auxiliary.LM = lm(res.now~res.pre)
>
> abline(auxiliary.LM, col = "red")
>
> summary(auxiliary.LM)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept) -0.03219    0.56092  -0.057  0.95460
res.pre      0.44763    0.15921   2.812  0.00835 **
```

- Another possible reason that error lack independent is having

$$\varepsilon_i = \rho\varepsilon_{i-1} + \nu_i$$

where $\nu_i$ are independent and identically distributed

$$\nu_i \sim \mathrm{N}(0, \sigma^2)$$

- This structure in errors is called first-order autoregressive process or AR(1).
- Together with our early distributed lag model,

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_{i-1} + \varepsilon_i$$

we have

$$Y_i = \beta_0(1-\rho) + \beta_1 x_i + (\beta_2 - \rho\beta_1)x_{i-1} - \beta_2\rho x_{i-2} + \rho Y_{i-1} + \nu_i$$

- Since it is nonlinear in the coefficients, the optimisation is not trivial.

- R can construct this model, and solve the optimisation for us,

```
> attach(s_pre.df)
> sales_pre.AR1 = arima(sales, order = c(1, 0, 0),
      xreg = cbind(ad, ad1))
> detach(s_pre.df)
>
> sales_pre.AR1
```

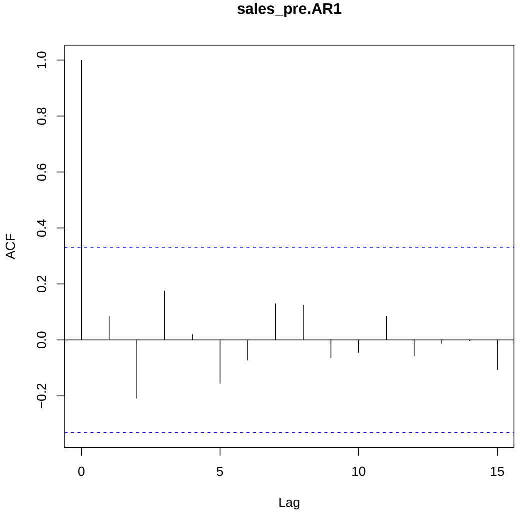
```
Coefficients:
         ar1  intercept      ad     ad1
      0.4966    16.9080  0.1218  0.1391
s.e.  0.1580     1.6716  0.0308  0.0316

sigma^2 estimated as 9.476:  log likelihood = -89.16,  aic = 188.32
```

```
> summary(sales_pre.LM)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) 15.60361    1.34531  11.599 5.35e-13 ***
ad           0.14242    0.03518   4.049 0.000305 ***
ad1          0.16651    0.03606   4.617 6.03e-05 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 3.652 on 32 degrees of freedom
```

- And it seems the new error $\nu_i$ are independent and identically distributed,

```
> acf(sales_pre.AR1$residuals, main = "sales_pre.AR1")
```



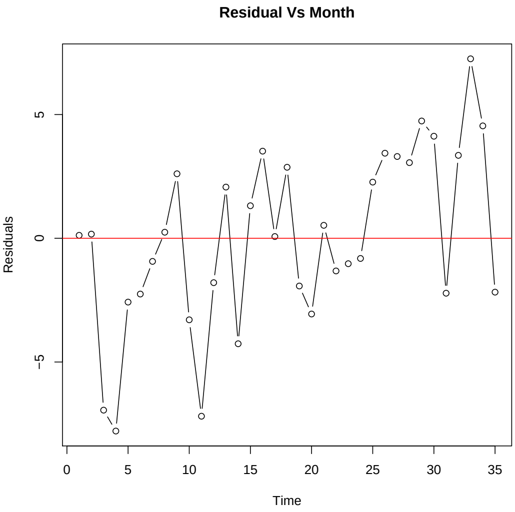
- However, more advanced models are harder to interpret,

$$Y_i = \beta_0(1-\rho) + \beta_1 x_i + (\beta_2 - \rho\beta_1)x_{i-1} - \beta_2\rho x_{i-2} + \rho Y_{i-1} + \nu_i$$

  and we would like to avoid if possible.
- In this case, there is actually a simpler model that is reasonable good.
- Consider the following model again

```
> sales_pre.LM = lm(sales~ad+ad1, data = s_pre.df)
>
> res = sales_pre.LM$residuals

> plot(res, type = "b",
+      xlab = "Time", ylab = "Residuals",
+      main = "Residual Vs Month")
>
> abline(h = 0, col = "red")
```

- Notice there seems to be an weak increasing trend



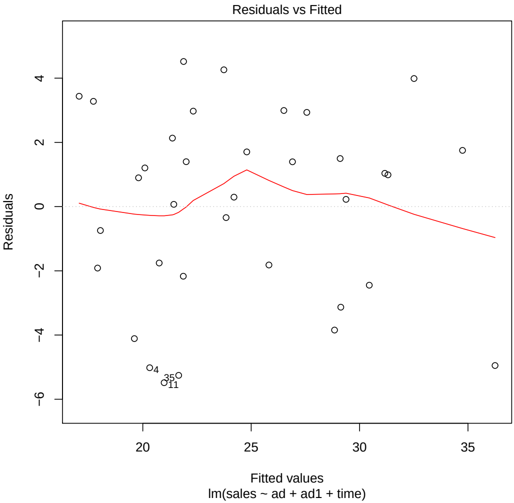
- Assuming the data is recorded/sort according to time, the plot suggests that time might help to explain the error, thus the sales number.

```
> time = 1:nrow(s_pre.df)
>
> sales_pre_time.LM =
+   lm(sales~ad+ad1+time, data = s_pre.df)
> summary(sales_pre_time.LM)
```

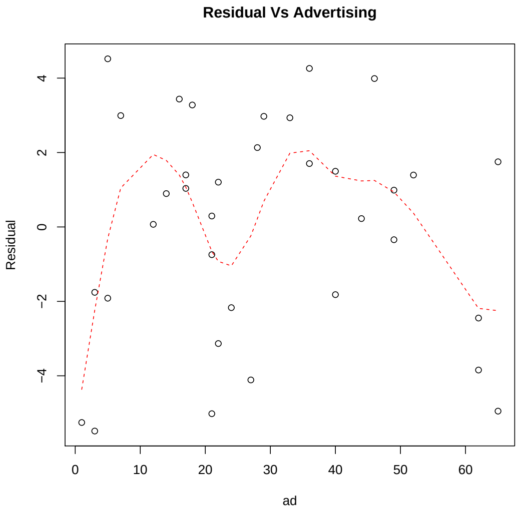
```
Call:
lm(formula = sales ~ ad + ad1 + time, data = s_pre.df)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4840 -2.0409  0.8971  1.9423  4.5192

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept) 12.03345    1.48653   8.095 3.84e-09 ***
ad           0.15308    0.02984   5.129 1.48e-05 ***
ad1          0.15914    0.03052   5.214 1.16e-05 ***
time         0.19323    0.05189   3.724 0.000781 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 3.084 on 31 degrees of freedom
Multiple R-squared:  0.7507,    Adjusted R-squared:  0.7266
F-statistic: 31.12 on 3 and 31 DF,  p-value: 1.769e-09
```

- It seems `time` is highly significant, but we have to check our assumptions,

```
> plot(sales_pre_time.LM, which = 1)
```

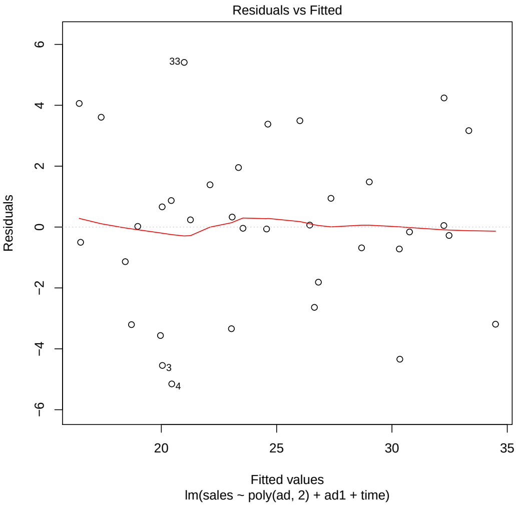


- It seems that we need a polynomial term for `ad`,



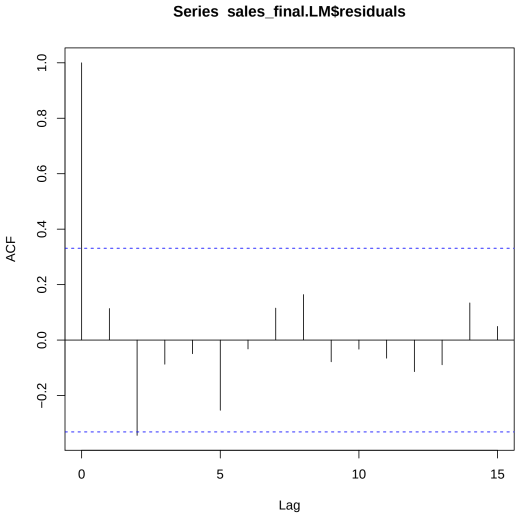
- After adding a quadratic term to the model, assumption 1. seems to be fixed

```
> plot(sales_final.LM, which = 1)
```



- And 2. seems to be OK, but the fix for 3. is not to be as good as AR(1),

```
> acf(sales_final.LM$residuals)
```



- But given we have only a relatively small number of data points, and we probably should not push for AR(2) without a significant AR(1) term.
- The normality assumption seems to be reasonable as well.

```
        Shapiro-Wilk normality test

data:   sales_final.LM$residuals
W = 0.9687, p-value = 0.4086
```

- According to AIC,

```
> AIC(sales_pre.AR1); AIC(sales_final.LM)
```

```
[1] 188.3158
[1] 179.2664
```

we prefer the model `sales_final.LM` if we are planning to use it to explain.

- However, if we want a predictive model, and we have a lot more data, then

data-splitting or cross-validation

should be used to determine which one is better.

```r
> summary(sales_final.LM)
```

```
Call:
lm(formula = sales ~ poly(ad, 2) + ad1 + time, data = s_pre.df)

Residuals:
    Min      1Q  Median      3Q     Max
-5.1523 -1.4739  0.0221  1.4357  5.4090

Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept)  15.40536    1.32144  11.658 1.15e-12 ***
poly(ad, 2)1 16.90116    3.04137   5.557 4.83e-06 ***
poly(ad, 2)2 -7.75778    3.09681  -2.505   0.0179 *
ad1           0.16483    0.02831   5.823 2.29e-06 ***
time          0.24254    0.05185   4.678 5.77e-05 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.851 on 30 degrees of freedom
Multiple R-squared:  0.7939,    Adjusted R-squared:  0.7664
F-statistic: 28.88 on 4 and 30 DF,  p-value: 6.653e-10
```

Q: If you are the manager, what conclusions can you draw from this model?