STATS 500 HW6

Minxuan Chen

2023-11-01

Table of contents

Github repo: https://github.com/PKUniiiiice/STATS_500

Problem 1

(a)

```r
library(faraway)
library(quantreg)
```

```
Loading required package: SparseM

Attaching package: 'SparseM'

The following object is masked from 'package:base':

    backsolve
```

```r
library(MASS)
data("stackloss")

# least squares
lsq <- lm(stack.loss ~ ., data=stackloss)
summary(lsq)
```

```
Call:
lm(formula = stack.loss ~ ., data = stackloss)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.2377 -1.7117 -0.4551  2.3614  5.6978 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -39.9197    11.8960  -3.356  0.00375 ** 
Air.Flow      0.7156     0.1349   5.307  5.8e-05 ***
Water.Temp    1.2953     0.3680   3.520  0.00263 ** 
Acid.Conc.   -0.1521     0.1563  -0.973  0.34405    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.243 on 17 degrees of freedom
Multiple R-squared:  0.9136,    Adjusted R-squared:  0.8983 
F-statistic:  59.9 on 3 and 17 DF,  p-value: 3.016e-09
```

Least squares works well and we observe a R^2 of 0.9136 and a adjusted R^2 of 0.8983. And the predictors `Air.Flow`, `Water.Temp` and the intercept are significant.

```r
#least absolute deviations
glad <- quantreg::rq(stack.loss ~ ., data=stackloss)
summary(glad)
```

```
Call: quantreg::rq(formula = stack.loss ~ ., data = stackloss)

tau: [1] 0.5

Coefficients:
            coefficients lower bd  upper bd
(Intercept) -39.68986    -41.61973 -29.67754
Air.Flow      0.83188      0.51278   1.14117
Water.Temp    0.57391      0.32182   1.41090
Acid.Conc.   -0.06087     -0.21348  -0.02891
```

There is some change in the coefficients. While the confidence intervals indicate significance for all predictors, we note that the upper bound of `Acid.Conc` is -0.02891, a value close to zero. This suggests that `Acid.Conc` may only be weakly significant. Additionally, although the estimate for `Water.Temp` has reduced to 0.57, which is half of the previous estimation, its confidence interval is relatively wide. The upper bound is 1.4109, encompassing the estimate from the least squares model.

```r
#Huber method
ghuber <- MASS::rlm(stack.loss ~ ., data=stackloss)
summary(ghuber)
```

```
Call: rlm(formula = stack.loss ~ ., data = stackloss)
Residuals:
     Min       1Q   Median       3Q      Max
-8.91753 -1.73127  0.06187  1.54306  6.50163

Coefficients:
            Value    Std. Error t value
(Intercept) -41.0265   9.8073    -4.1832
Air.Flow      0.8294   0.1112     7.4597
Water.Temp    0.9261   0.3034     3.0524
Acid.Conc.   -0.1278   0.1289    -0.9922

Residual standard error: 2.441 on 17 degrees of freedom
```

Again, there is some change in the coefficients. The confidence intervals suggest that `Acid.Conc` is not significant. And the standard error becomes smaller (2.441).

```
#least trimmed squares
glts <- MASS::ltsreg(stack.loss ~ ., data=stackloss, nsamp="exact")
round(glts$coefficients, 4)
```

```
(Intercept)    Air.Flow  Water.Temp  Acid.Conc.
   -35.8056      0.7500      0.3333      0.0000
```

Comparing the results to the least squares model, we observe significant changes in the estimation of `Water.Temp`. In fact, its value falls outside the confidence interval of the least squares estimation. To obtain standard errors for the LTS regression coefficients, we employed a bootstrap method.

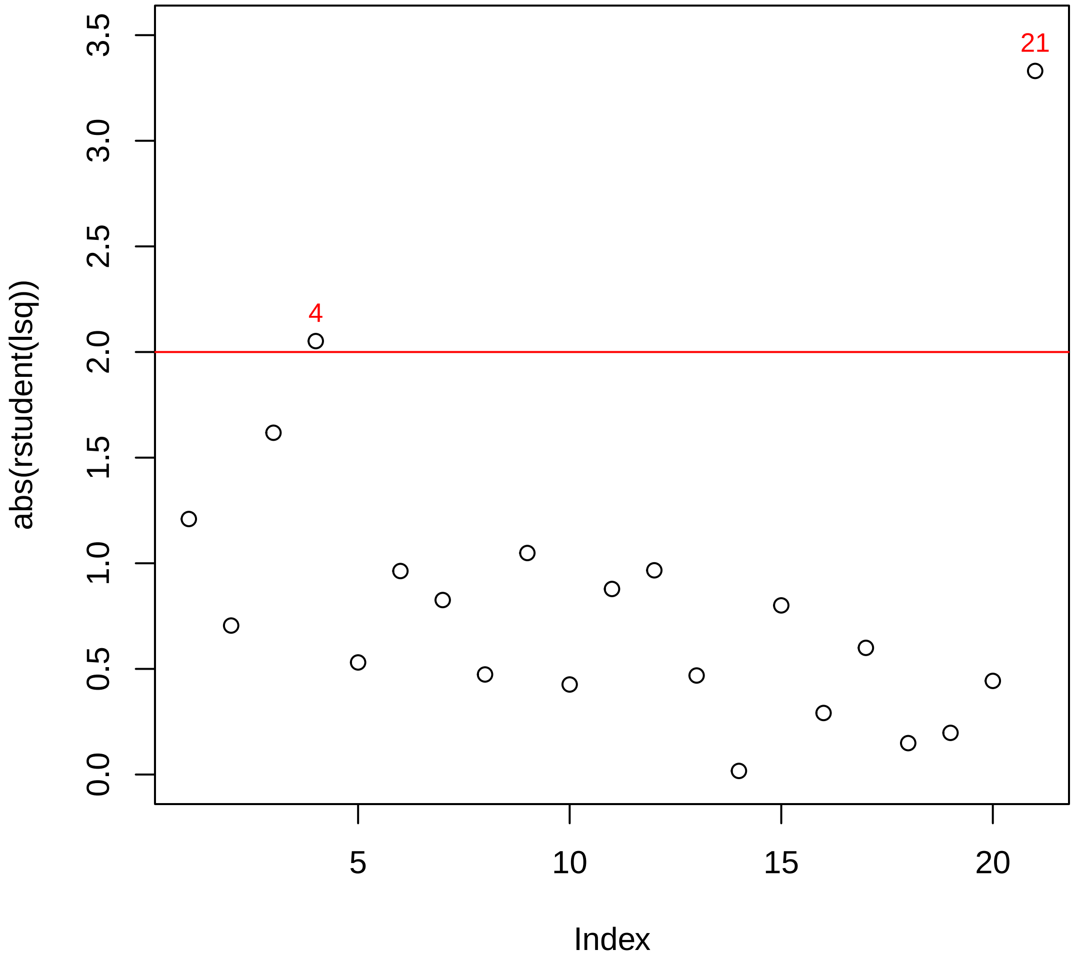
```
bcoef <- matrix(0,1000,4)
for(i in 1:1000){
  newy <- glts$fitted.values + glts$residuals[sample(21, rep=T)]
  bcoef[i,] <- MASS::ltsreg(stack.x, newy, nsamp="best")$coef
}
apply(bcoef,2,function(x) quantile(x,c(0.025,0.975)))
```

```
           [,1]      [,2]       [,3]       [,4]
2.5%  -51.11928 0.5849458 -0.2962173 -0.2222222
97.5% -18.71718 0.9654429  0.7433241  0.2211682
```

From the bootstrap, both `Water.Temp` and `Acid.Conc` are not significant.

Now, we use diagnostic methods to detect outliers or influential points. We consider the least squares model.

```
#outlier
plot(abs(rstudent(lsq)),ylim=c(0, 3.5))
abline(h=2, col='red')
text(x=c(4,21), y=abs(rstudent(lsq))[c(4,21)],
     labels=c(4,21), pos=3, col="red", cex=0.8)
```




```r
car::outlierTest(lsq)
```

```
No Studentized residuals with Bonferroni p < 0.05
Largest |rstudent|:
    rstudent unadjusted p-value Bonferroni p
21 -3.330493           0.004238     0.088999
```

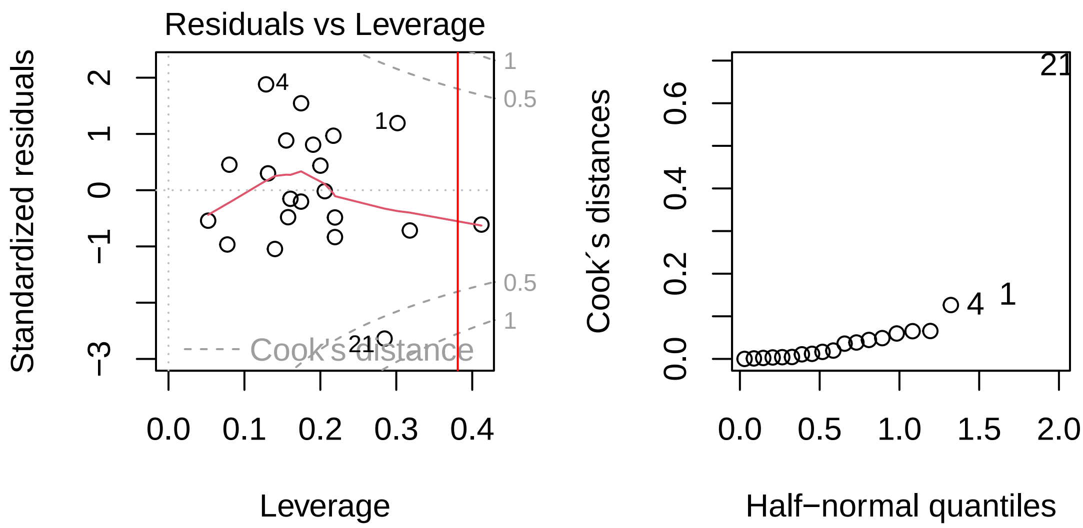
Based on this plot, it appears that there are no outliers among the data points. Furthermore, the test results confirm this observation, as the Bonferroni p-value exceeds 0.05, indicating the absence of outliers.

```r
#leverage and influential
par(mfrow=c(1,2))
plot(lsq, which=5)
abline(v=2*length(lsq$coefficients)/nrow(stackloss), col='red')
hatvalues(lsq)>2*length(lsq$coefficients)/nrow(stackloss)
```

```
    1     2     3     4     5     6     7     8     9    10    11    12    13
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
   14    15    16    17    18    19    20    21 
FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE 
```

```
halfnorm(cooks.distance(lsq),3,ylab="Cook's distances")
```




```
par(mfrow=c(1,1))
```

There is only one point with high leverage, however, from the Cook's distance line, this point is not influential.

From the half-norm plot, case No.21 is likely to be influential.

From above, we remove case No.4 and No.21 and then use least squares.

```
lsq.rm <- lm(stack.loss ~ ., data=stackloss, subset=-c(4,21))
summary(lsq.rm)
```

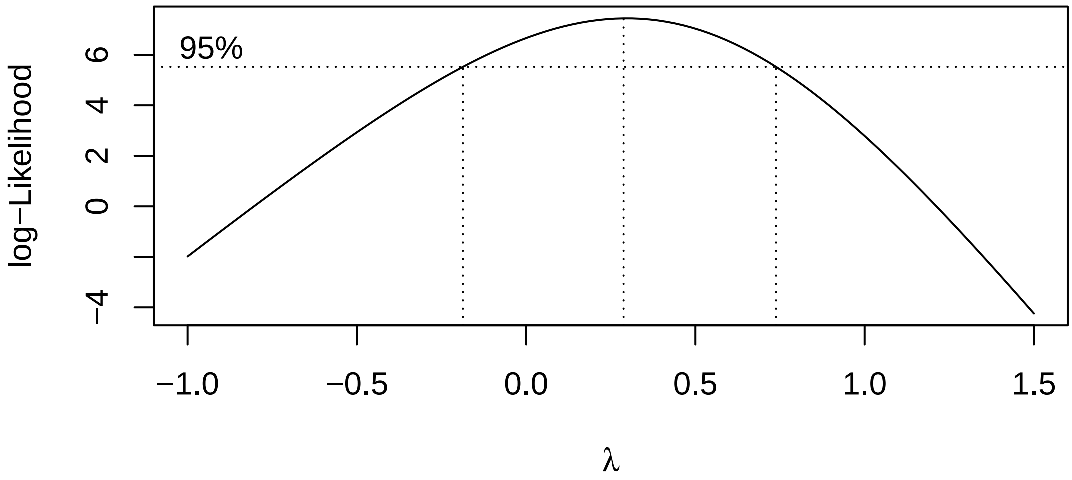
```
Call:
lm(formula = stack.loss ~ ., data = stackloss, subset = -c(4, 
    21))

Residuals:
    Min      1Q  Median      3Q     Max 
-3.1114 -1.4080 -0.0749  1.0946  3.6074 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -42.45308    7.38458  -5.749 3.85e-05 ***
Air.Flow      0.95660    0.09447  10.126 4.24e-08 ***
Water.Temp    0.55557    0.26403   2.104   0.0526 .  
Acid.Conc.   -0.10877    0.09678  -1.124   0.2787    
```

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

Residual standard error: 1.996 on 15 degrees of freedom
Multiple R-squared:  0.9693,    Adjusted R-squared:  0.9632
F-statistic: 158.1 on 3 and 15 DF,  p-value: 1.426e-11
```

Comparing the results to the original least squares method, we observe obvious changes in the coefficients. Additionally, the R^2 value has increased, indicating a better fit.

(b)

```r
MASS::boxcox(lsq, plotit=T, lambda=seq(-1, 1.5, by=0.1))
```



The confidence interval of λ is approximately $(-0.2, 0.7)$. Notably, $\lambda = 1$ falls outside of this range. Therefore, it becomes necessary to consider some transformation of the `stack.loss` variable. For the sake of convenience and interpretation, we can explore two options: setting $\lambda = 0$, which results in log(`stack.loss`), or choosing $\lambda = 0.5$, which leads to $\sqrt{\texttt{stack.loss}}$. The likelihood values for these two choices are close"

(c)

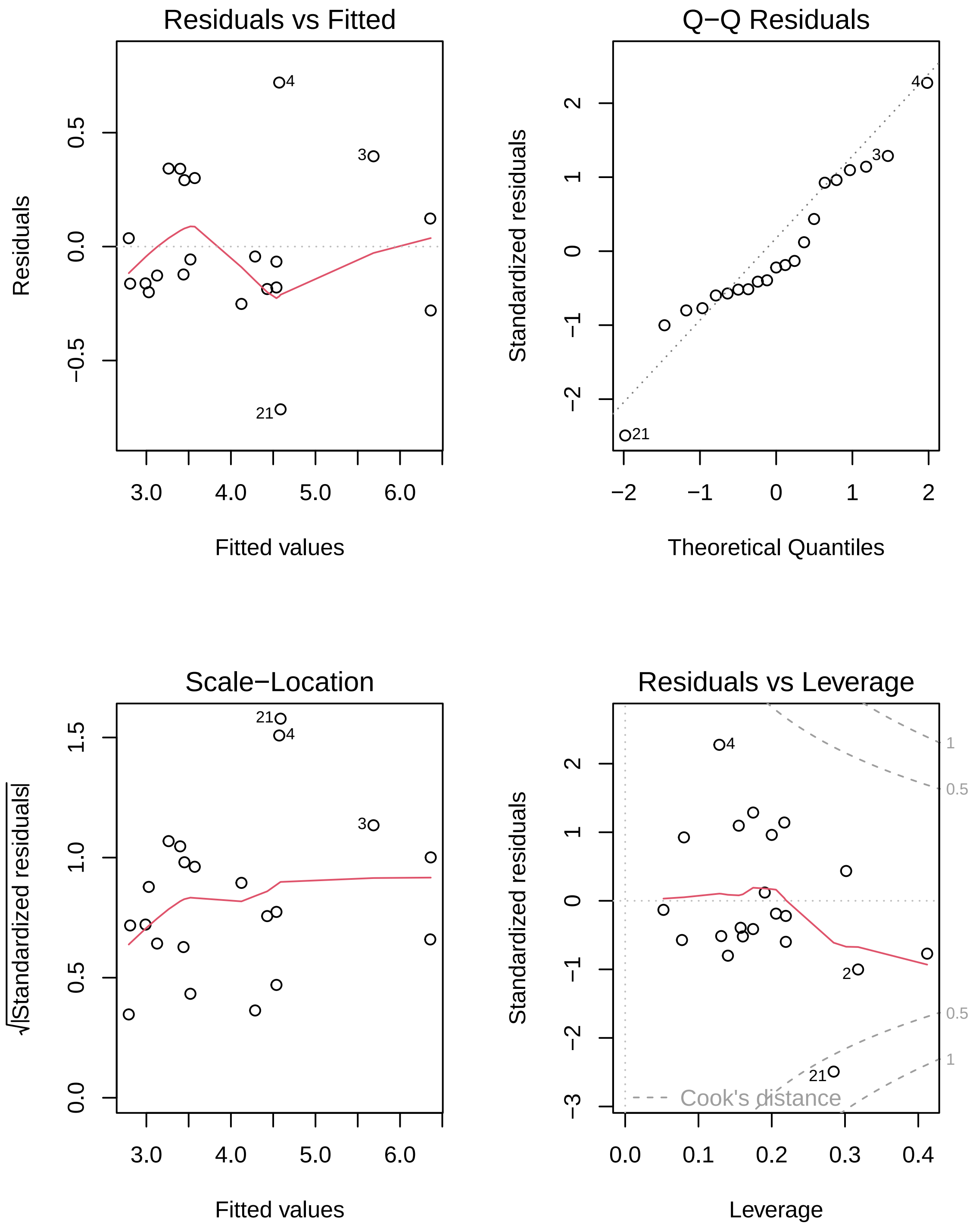
We use the square root transformation.

```r
lsq.sqrt <- lm(sqrt(stack.loss) ~ ., data=stackloss)
summary(lsq.sqrt)
```

```
Call:
lm(formula = sqrt(stack.loss) ~ ., data = stackloss)
```

```
Residuals:
     Min       1Q   Median       3Q      Max 
-0.71429 -0.17935 -0.06611  0.29145  0.71996 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -3.08547    1.24328  -2.482  0.02382 *  
Air.Flow     0.07607    0.01409   5.397 4.82e-05 ***
Water.Temp   0.14265    0.03846   3.709  0.00174 ** 
Acid.Conc.  -0.00555    0.01633  -0.340  0.73820    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.339 on 17 degrees of freedom
Multiple R-squared:  0.9218,    Adjusted R-squared:  0.908 
F-statistic: 66.78 on 3 and 17 DF,  p-value: 1.296e-09
```

```r
par(mfrow=c(2,2))
plot(lsq.sqrt)
```




```
par(mfrow=c(1,1))
```

From the residuals vs fitted values and QQ plots, there is no obvious issues regarding linearity and normality. The scale-location plot also does not reveal any significant violations of homoscedasticity. Additionally, the residuals vs. Leverage plot indicates no clear influential points.

In general, after applying the Box-Cox transformation, the results of the least squares regression seem to align with the standard assumptions.

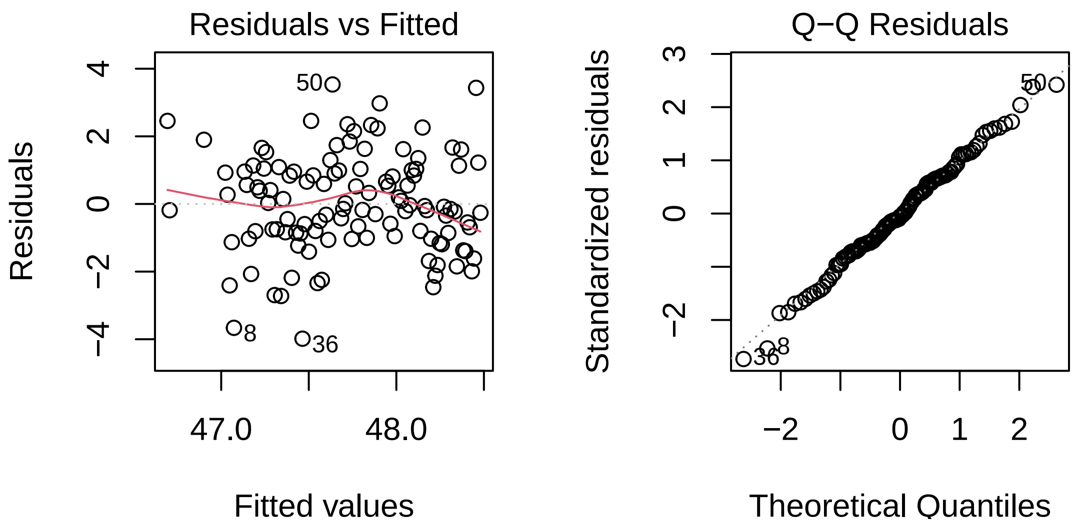
Problem 2

(a)

```r
data("aatemp")
m1 <- lm(temp~year, data=aatemp)
summary(m1)
```

```
Call:
lm(formula = temp ~ year, data = aatemp)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.9843 -0.9113 -0.0820  0.9946  3.5343 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) 24.005510   7.310781   3.284  0.00136 **
year         0.012237   0.003768   3.247  0.00153 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.466 on 113 degrees of freedom
Multiple R-squared:  0.08536,	Adjusted R-squared:  0.07727 
F-statistic: 10.55 on 1 and 113 DF,  p-value: 0.001533
```

```r
par(mfrow=c(1,2))
plot(m1, which=c(1,2))
```



```r
par(mfrow=c(1,1))
```

Note that the estimation of the year variable is statistically significant, suggesting the presence of a non-constant linear trend.

In the QQ plot, there are no apparent issues. However, in the residuals vs. fitted plot, a slight horizontal "S" curve is discernible in the residuals, as indicated by the red lowess line. This observation implies structural problems within the linear model concerning the year variable. Therefore, it may be necessary to move to a nonlinear model.

(b)

```r
library(nlme)
m2.corr <- nlme::gls(temp~year,
                     correlation=corAR1(form=~year), data=aatemp)
summary(m2.corr)
```

```
Generalized least squares fit by REML
  Model: temp ~ year
  Data: aatemp
       AIC     BIC    logLik
  426.5694 437.479 -209.2847

Correlation Structure: ARMA(1,0)
 Formula: ~year
 Parameter estimate(s):
     Phi1
0.2303887

Coefficients:
               Value Std.Error  t-value p-value
(Intercept) 25.18407  8.971864 2.807006  0.0059
year         0.01164  0.004626 2.516015  0.0133

 Correlation:
     (Intr)
year -1

Standardized residuals:
       Min         Q1        Med         Q3        Max
-2.7230803 -0.6321970 -0.0520135  0.6645795  2.3775123

Residual standard error: 1.475718
Degrees of freedom: 115 total; 113 residual
```

```r
intervals(m2.corr, which="var-cov")
```

```
Approximate 95% confidence intervals

 Correlation structure:
          lower      est.     upper
Phi1 0.02937005 0.2303887 0.4134963

 Residual standard error:
   lower     est.    upper
1.284098 1.475718 1.695932
```

The correlation is 0.2304, and the confidence interval does not contain 0. Therefore, we conclude that the correlation is statistically significant.

Regarding the linear trend, the estimated coefficient for the 'year' variable remains statistically significant at level of $\alpha = 0.05$.' However, it's worth noting that its p-value is not particularly small, and the estimate is close to zero. As a result, we consider the linear trend in this case to be less significant compared to the one in (a).

(c)

```r
#we start with poly(year, 10) and check whether the highest polynomial term is significa
m.try <- NULL
summ <- NULL
for (i in 10:1){
  summ <- summary(m.try <- lm(temp ~ poly(year, i), data=aatemp))
  deg.max <- summ$coefficients[i+1, "Pr(>|t|)"]
  if (deg.max<0.05) break
}
summ
```

```
Call:
lm(formula = temp ~ poly(year, i), data = aatemp)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7142 -0.9198 -0.1420  0.9903  3.2364

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     47.7426     0.1306 365.604  < 2e-16 ***
poly(year, i)1   4.7616     1.4004   3.400 0.000942 ***
poly(year, i)2  -0.9071     1.4004  -0.648 0.518500
```

```
poly(year, i)3  -3.3132     1.4004  -2.366 0.019749 *
poly(year, i)4   2.4383     1.4004   1.741 0.084470 .
poly(year, i)5   3.3824     1.4004   2.415 0.017384 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.4 on 109 degrees of freedom
Multiple R-squared:  0.1952,    Adjusted R-squared:  0.1583
F-statistic: 5.289 on 5 and 109 DF,  p-value: 0.0002176
```

Therefore, the highest degree polynomial to fit the model is 5.

The specific final model is a polynomial model of degree 5. And the regression equation is (ref: click)

$$\begin{aligned} \text{temp} =& \beta_0 F_0(\text{year}) + \beta_1 F_1(\text{year}) + \beta_2 F_2(\text{year}) + \\ & \beta_3 F_3(\text{year}) + \beta_4 F_4(\text{year}) + \beta_5 F_5(\text{year}) \end{aligned}$$

in which, β_i is as follows,

```r
beta <- coef(m.try)
print(sapply(1:length(beta), function(i) {
  paste("beta_", i-1, ": ", beta[i], sep = "")
}))
```

```
[1] "beta_0: 47.7426086956522"   "beta_1: 4.76157203343734"
[3] "beta_2: -0.907109792628081" "beta_3: -3.3132428905303"
[5] "beta_4: 2.43833195639968"   "beta_5: 3.3823682056615"
```

And $F_i(\text{year})$ is defined recursively by

$$\begin{aligned} F_0(x) &= 1/\sqrt{n_2} \\ F_1(x) &= (x - a_1)/\sqrt{n_3} \\ F_i(x) &= \frac{(x - a_i) \cdot \sqrt{n_{i+1}} \cdot F_{i-1}(x) - \frac{n_{i+1}}{\sqrt{n_i}} \cdot F_{i-2}(x)}{\sqrt{n_{i+2}}}, \quad i \geq 2 \end{aligned}$$

in which a_i and n_i are No.i element of the following vector

```r
ai <- attributes(z <- poly(aatemp$year, 5))$coefs$alpha
ni <- attributes(z)$coefs$norm2
F.i <- function(x, i){
  if(i==0){
    return (1/sqrt(ni[2]))
  }
  else if(i==1){
    return ((x-ai[1])/sqrt(ni[3]))
  }
```

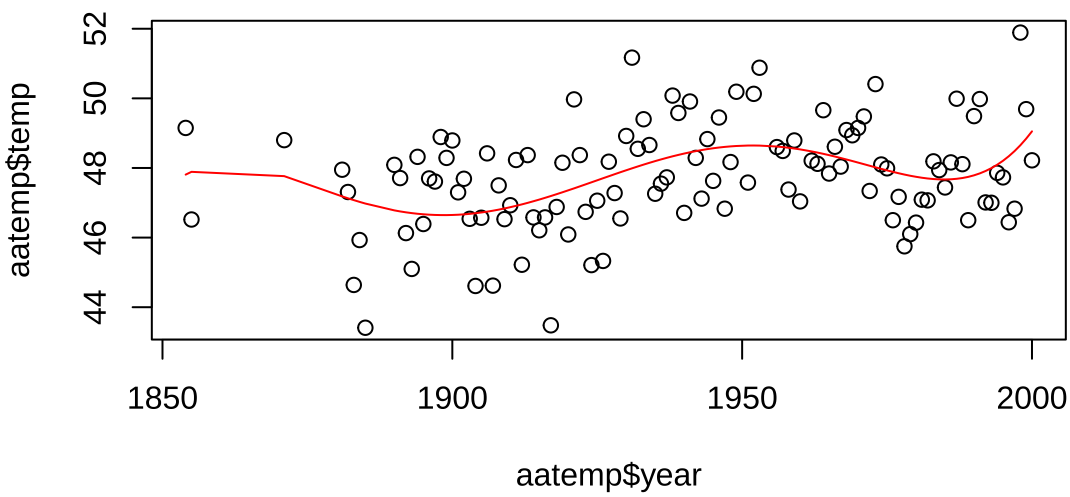
```
  else{
    return ((((x-ai[i])*sqrt(ni[i+1]))*F.i(x, i-1) -
    ni[i+1]/sqrt(ni[i])*F.i(x, i-2))/sqrt(ni[i+2]))
  }
}
# ai
ai
```

```
[1] 1939.739 1935.241 1919.775 1920.484 1929.649
```

```
# ni
ni
```

```
[1] 1.000000e+00 1.150000e+02 1.514022e+05 2.068262e+08 3.593395e+11
[6] 5.349960e+14 6.163661e+17
```

```
#plot the fitted model on top of the data
plot(aatemp$year, aatemp$temp)
lines(aatemp$year, m.try$fitted.values, col="red")
```



For the polynomial model, R^2 is 0.1952 and adjusted R^2 is 0.1583. This result is better than the simple linear model. The plot also shows the model fits the data well.

(d)

Results of polynomial model

```r
newx <- data.frame(year=2020)
new.temp <- predict(m.try, newdata=newx); new.temp
```

```
       1 
60.07774 
```

```r
#confidence interval
predict(m.try, newdata = newx, interval = "confidence", level = 0.95)
```

```
       fit      lwr      upr
1 60.07774 50.22436 69.93112
```

```r
#predictive interval
predict(m.try, newdata = newx, interval = "prediction", level = 0.95)
```

```
       fit      lwr      upr
1 60.07774 49.84092 70.31456
```

Results of simple linear model

```r
new.temp <- predict(m1, newdata=newx); new.temp
```

```
       1 
48.72478 
```

```r
#confidence interval
predict(m1, newdata = newx, interval = "confidence", level = 0.95)
```

```
       fit     lwr      upr
1 48.72478 48.0672 49.38237
```

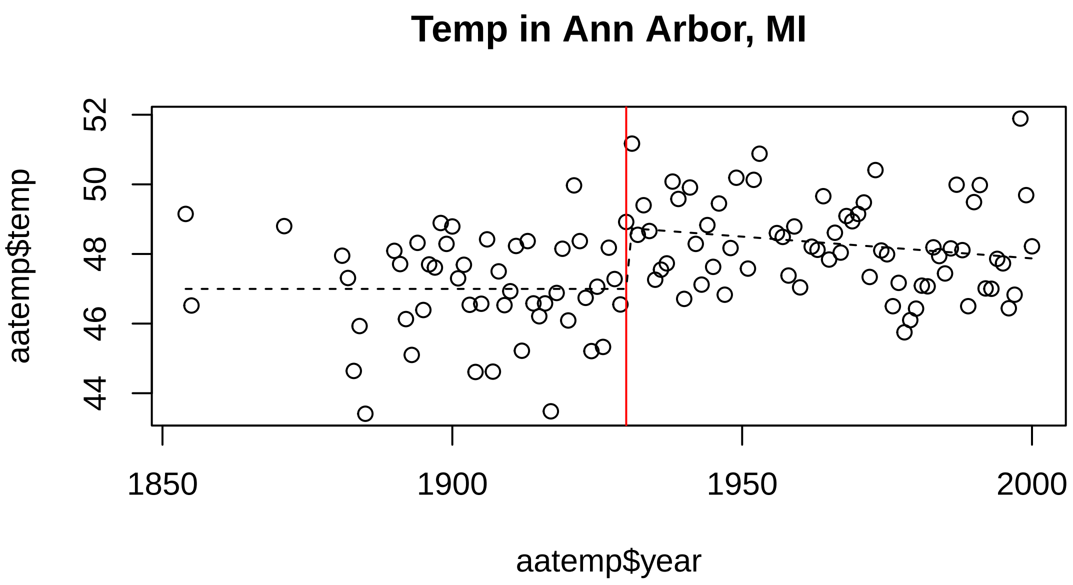
```r
#predictive interval
predict(m1, newdata = newx, interval = "prediction", level = 0.95)
```

```
       fit      lwr     upr
1 48.72478 45.74636 51.7032
```

For this extrapolation, the polynomial model provides a higher temperature prediction for the year 2020 and a significantly wider confidence/prediction interval. This is primarily attributed to the inherent instability of polynomial methods for extrapolation (Runge's phenomenon). In reality, according to NOAA records, the annual mean temperature is approx 50°F. Therefore, the temperature prediction offered by the polynomial model is unreliable in this context. In general, both models raise doubts when extrapolating to such a distant future. However, the linear model outperforms the polynomial model significantly.

(e)

```r
plot(aatemp$year, aatemp$temp, main="Temp in Ann Arbor, MI")
abline(v=1930, col="red")
lhs <- function(x) ifelse(x<=1930, mean(aatemp$temp[1:49]), 0)
rhs <- function(x) ifelse(x>1930, x-1930, 0)
m2 <- lm(temp~lhs(year)+rhs(year), data=aatemp)
x <- aatemp$year
py <- m2$coef[1] + m2$coef[2]*lhs(x) + m2$coef[3]*rhs(x)
lines(x, py, lty=2)
```




```r
summary(m2)
```

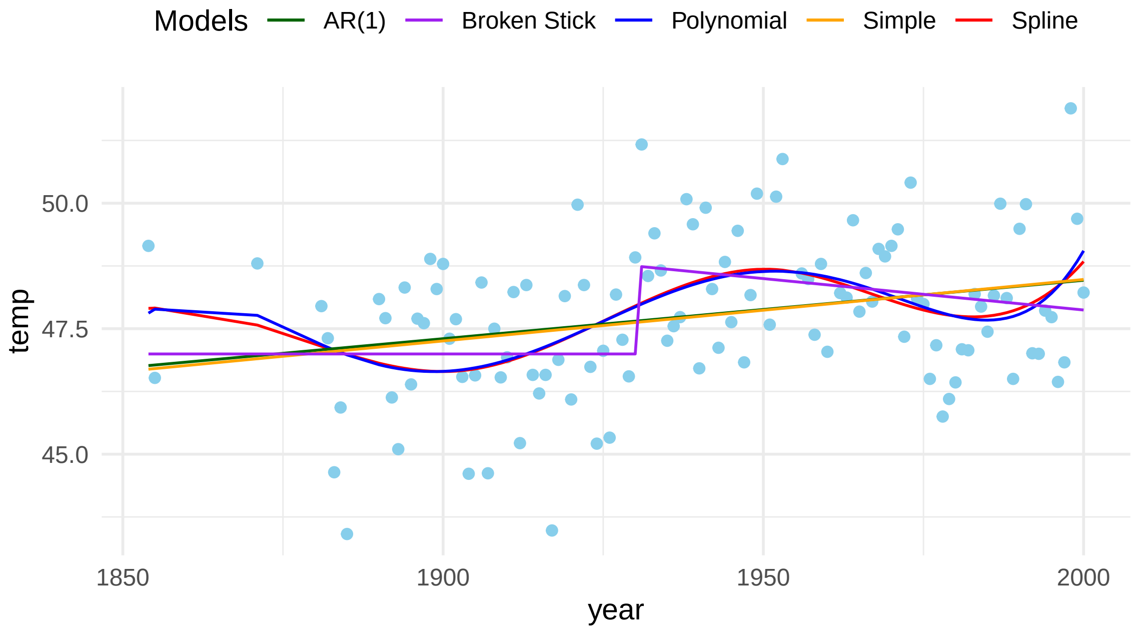
```
Call:
lm(formula = temp ~ lhs(year) + rhs(year), data = aatemp)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5867 -0.9456 -0.0979  1.0233  3.9925

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 48.748738   0.343152 142.062  < 2e-16 ***
lhs(year)   -0.037279   0.008423  -4.426 2.24e-05 ***
rhs(year)   -0.012519   0.008248  -1.518    0.132
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.381 on 112 degrees of freedom
Multiple R-squared:  0.1954,	Adjusted R-squared:  0.181 
F-statistic:  13.6 on 2 and 112 DF,  p-value: 5.167e-06
```

While the R^2 of this model is 0.1954, significantly higher than that of the simple linear model, we observe that the coefficient of `rhs(year)` is not statistically significant. We cannot assert the presence of a linear trend after 1930. Therefore, the assertion doesn't appears to be reasonable.

(f)

```r
library(splines)
knots <- c(1854, 1854, 1854, seq(1854, 2000, length.out=4),
           2000, 2000, 2000)
byear <-  splineDesign(knots, aatemp$year)
gs <-  lm(aatemp$temp ~ byear-1)
df.plot <- data.frame(cbind(aatemp$year, aatemp$temp,
                            gs$fitted.values,
                            m.try$fitted.values,
                            m2.corr$fitted,
                            m1$fitted.values,
                            py))
colnames(df.plot) <- c("year", "temp", "Spline",
                       "Polynomial","AR(1)",
                       "Simple", "Broken Stick")
library(ggplot2)
# Create a ggplot object
gg <- ggplot(data = df.plot, aes(x = year)) +
  geom_point(aes(y = temp), color = "skyblue") +
  geom_line(aes(y = Spline, color = "Spline")) +
  geom_line(aes(y = Polynomial, color = "Polynomial")) +
  geom_line(aes(y = `AR(1)`, color = "AR(1)")) +
  geom_line(aes(y = Simple, color = "Simple")) +
  geom_line(aes(y = `Broken Stick`, color = "Broken Stick")) +
  scale_color_manual(values = c("Spline" = "red", "Polynomial" = "blue",
                                "AR(1)" = "darkgreen","Simple" = "orange",
                                "Broken Stick" = "purple")) +
  labs(color = "Line") +
  theme_minimal() +
  theme(legend.position = "top") +
  guides(color = guide_legend(title = "Models"))
gg
```



Generally, the cubic spline fit looks similar to polynomial fit of degree 5.

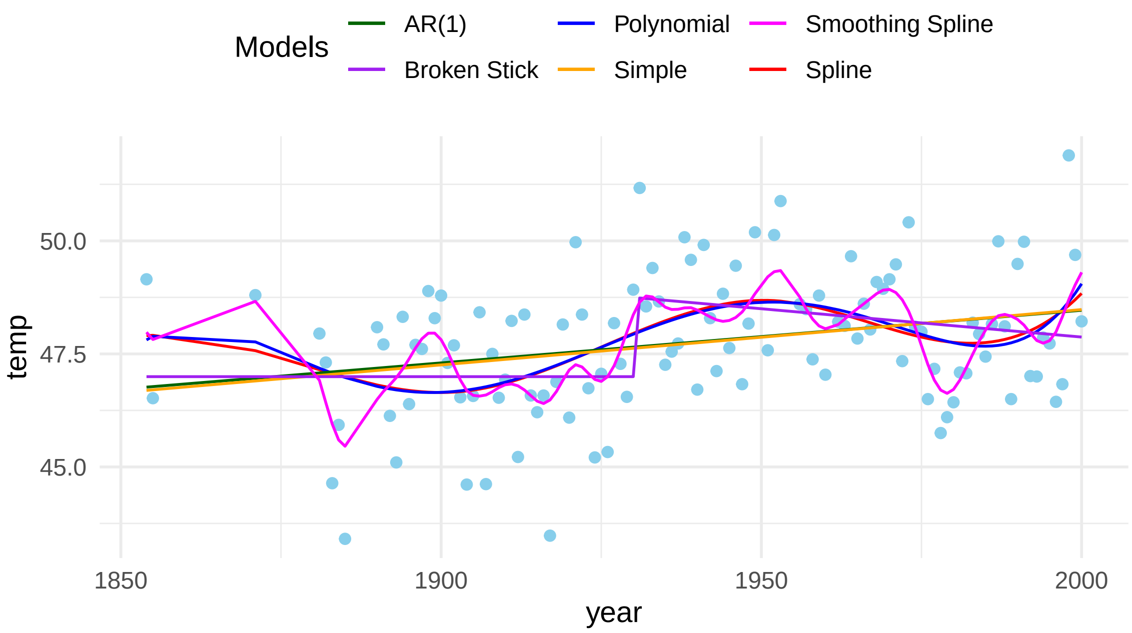
```
summary(gs)
```

```
Call:
lm(formula = aatemp$temp ~ byear - 1)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.7002 -0.9822 -0.1254  1.0076  3.3186 

Coefficients:
       Estimate Std. Error t value Pr(>|t|)    
byear1  47.9049     1.0147   47.21   <2e-16 ***
byear2  48.0395     1.1693   41.08   <2e-16 ***
byear3  44.9185     0.9138   49.15   <2e-16 ***
byear4  50.6944     0.8509   59.58   <2e-16 ***
byear5  46.4986     0.7448   62.43   <2e-16 ***
byear6  48.8377     0.6262   77.99   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.402 on 109 degrees of freedom
Multiple R-squared:  0.9992,    Adjusted R-squared:  0.9991 
F-statistic: 2.225e+04 on 6 and 109 DF,  p-value: < 2.2e-16
```

Based on the results, this model fits the data significantly better than the simple straight-line model. It can be described as almost a perfect fit, given that the R^2 value is very close to 1.

(g)

```
gsmooth <- smooth.spline(aatemp$year, aatemp$temp)
```



From the plot, it's hard to conclude that this model fits better than the straight-line and the spline model in (f). The fitted line appears overly wavy, suggesting that while it might be better for fitting the data, it is clearly overfitting in practice.