

STATS 500 HW6

Minxuan Chen

2023-10-31

Table of contents

Problem 1	1
(a)	1
(b)	5
(c)	6
Problem 2	8
(a)	8
(b)	9
(c)	10
(d)	12
(e)	13
(f)	15
(g)	17

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

Problem 1

(a)

```
1 library(faraway)
2 library(quantreg)
```

Loading required package: SparseM

Attaching package: 'SparseM'

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

backsolve

```
1 library(MASS)
2 data("stackloss")
3
4 # least squares
5 lsq <- lm(stack.loss ~ ., data=stackloss)
6 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 an adjusted R^2 of 0.8983. And the predictors `Air.Flow`, `Water.Temp` and the intercept are significant.

```

1 #least absolute deviations
2 glad <- quantreg::rq(stack.loss ~ ., data=stackloss)
3 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. The confidence intervals show all predictors are significant, but we observe the upper bound of `Acid.Conc` is -0.02891, which is close to 0. So we think maybe `Acid.Conc` is only weakly significant. Moreover, although, the estimation of `Water.Temp` becomes 0.57, which is half of the previous estimation, we observe it has a relative wide confidence interval. The upper bound is 1.4109, which contains the estimation of least squares model.

```

1 #Huber method
2 ghuber <- MASS::rlm(stack.loss ~ ., data=stackloss)
3 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 errors are benenerally smaller (2.441).

```

1 #least trimmed squares
2 glts <- MASS::ltsreg(stack.loss ~ ., data=stackloss, nsamp="exact")

```

```
3 round(glts$coefficients, 4)
```

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

Comparing to least squares result, there are relatively obvious change in estimation of `Water.Temp`, since its value lies outside the confidence interval of least squares estimation. We use bootstrap to get standard errors for the LTS regression coefficients.

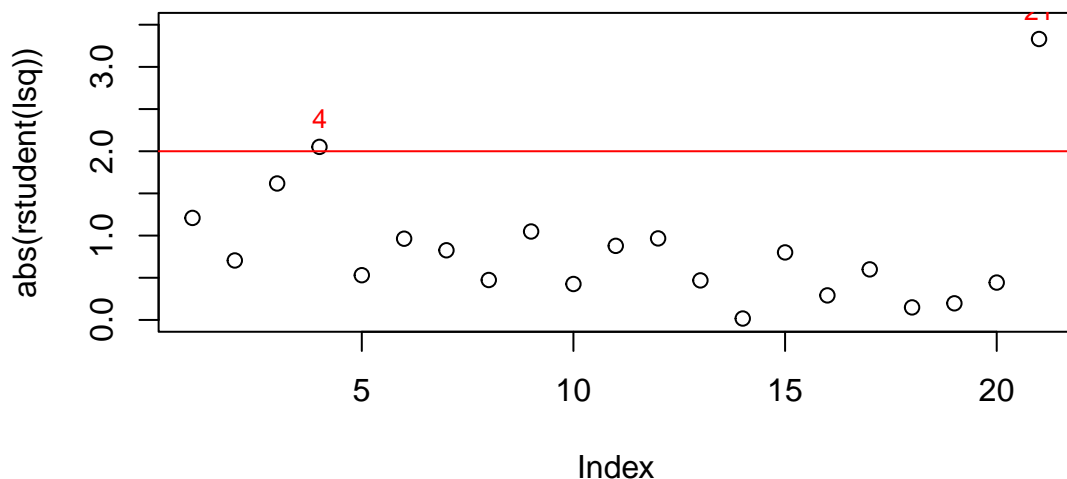
```
1 bcoef <- matrix(0,1000,4)
2 for(i in 1:1000){
3   newy <- glts$fitted.values + glts$residuals[sample(21, rep=T)]
4   bcoef[i,] <- MASS::ltsreg(stack.x, newy, nsamp="best")$coef
5 }
6 apply(bcoef,2,function(x) quantile(x,c(0.025,0.975)))
```

```
      [,1]      [,2]      [,3]      [,4]
2.5% -59.75412 0.5356633 -0.2140917 -0.1938948
97.5% -19.08888 0.9476866  0.7781915  0.2747162
```

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

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

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



```
1 car::outlierTest(lsq)
```

No Studentized residuals with Bonferroni $p < 0.05$

Largest $|rstudent|$:

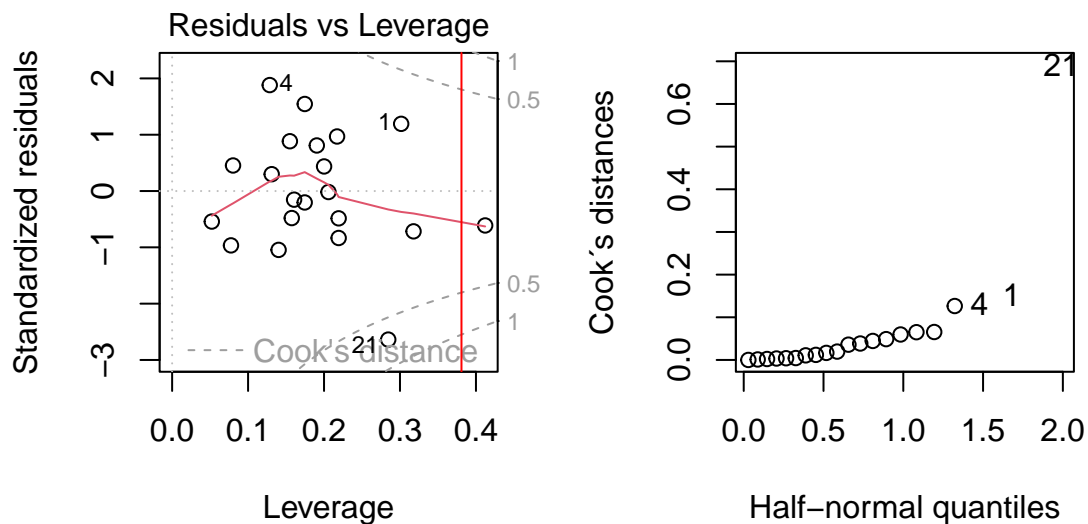
	rstudent	unadjusted p-value	Bonferroni p
21	-3.330493	0.004238	0.088999

From this plot, it's likely that all points are not outliers. From the test result, the bonferroni p-value is larger than 0.05, therefore, there are no outliers.

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

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



```
1 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.

```
1 lsq.rm <- lm(stack.loss ~ ., data=stackloss, subset=-c(4,21))
2 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

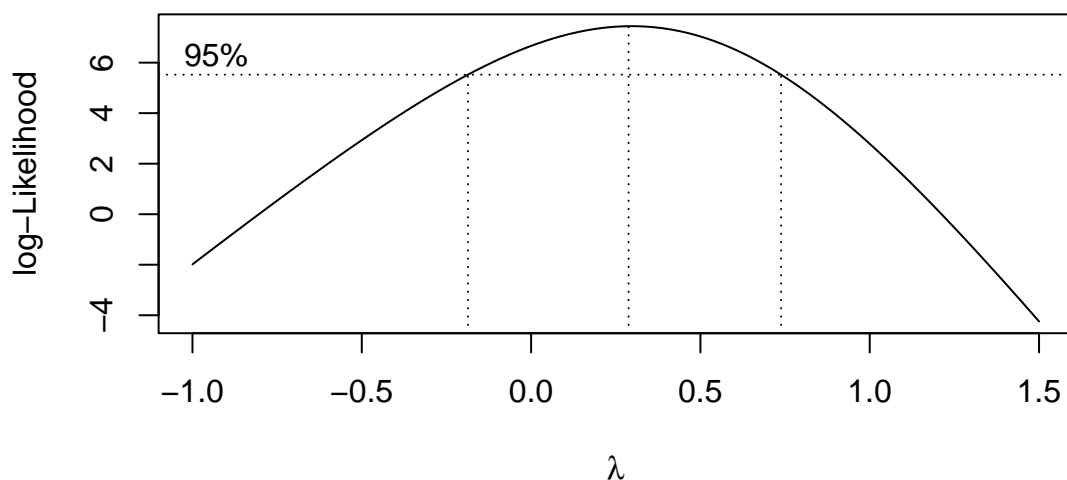
Multiple R-squared: 0.9693, Adjusted R-squared: 0.9632

F-statistic: 158.1 on 3 and 15 DF, p-value: 1.426e-11

Comparing to least squares method, there is an obvious change in coefficients. And the R^2 becomes greater. So we get a better fit.

(b)

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



The confidence interval of λ is about $(-0.2, 0.7)$. Note that $\lambda = 1$ is not in this range. So it's necessary to perform some transformation on the `stack.loss` variable. For convenience and interpretation, we can take $\lambda = 0$, i.e. $\log(\text{stack.loss})$ or $\lambda = 0.5$, i.e. $\sqrt{\text{stack.loss}}$. The likelihood values of these two choices are almost.

(c)

We use the square root transformation.

```
1 lsq.sqrt <- lm(sqrt(stack.loss) ~ ., data=stackloss)
2 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

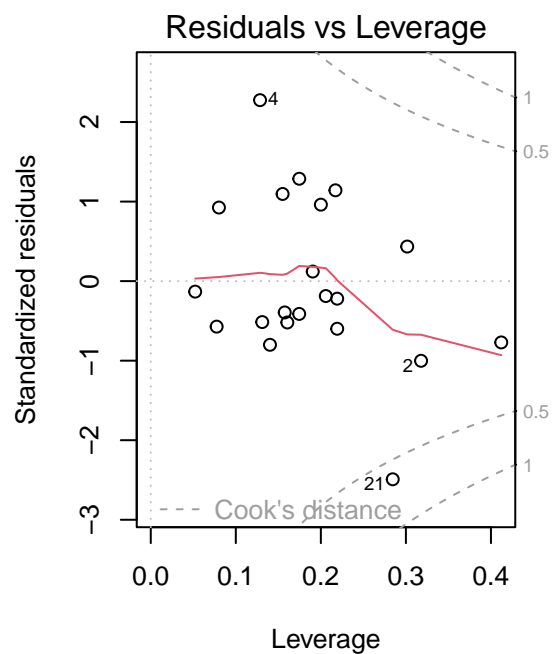
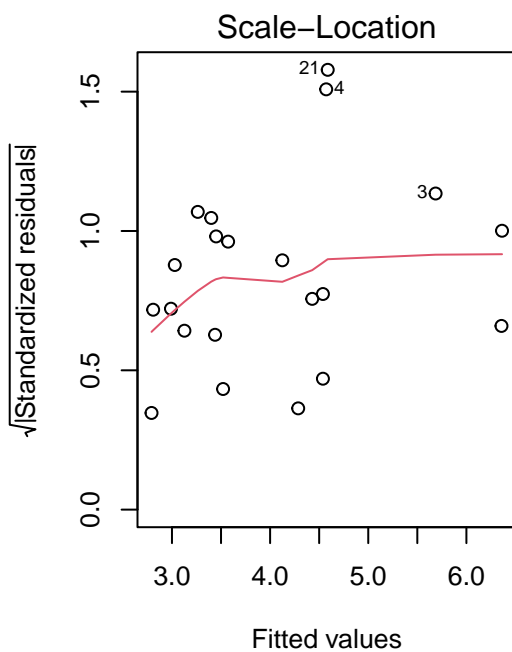
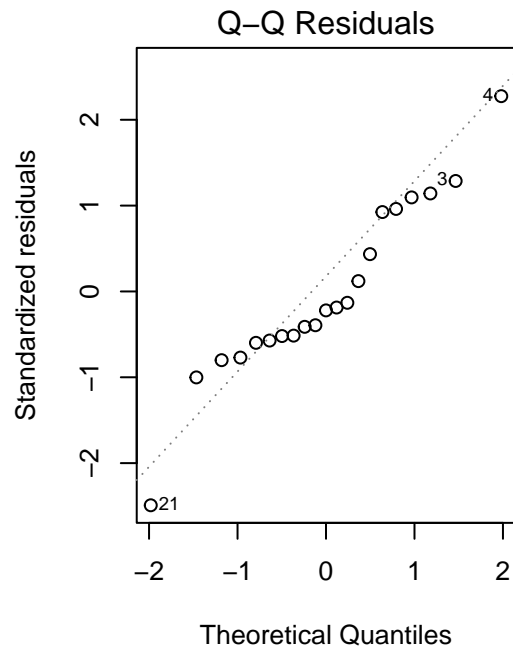
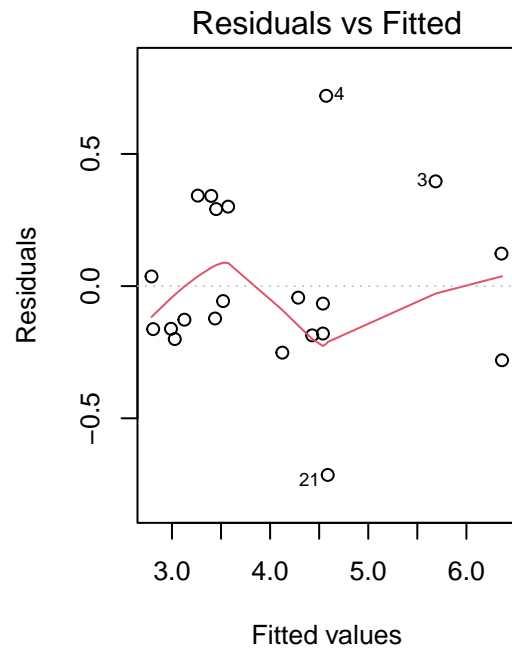
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

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



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

From the residuals vs fitted values and QQ plots, there is no obvious problems concerning linearity and normality. From scale-location plot, there is no obvious violation about homoscedasticity. Moreover, the residuals vs Leverage plot show there no obvious influential points.

Generally, after Box-Cox transformation, the least squares regression results appear to satisfy the standard assumption.

Problem 2

(a)

```
1 data("aatemp")
2 m1 <- lm(temp~year, data=aatemp)
3 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 **

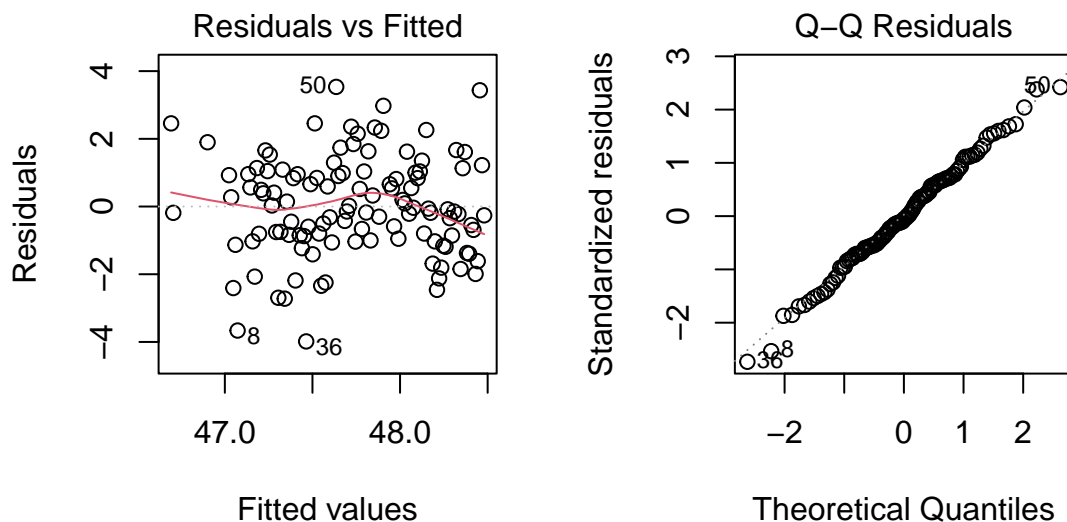
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

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



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

Note the estimation of year is significant, so the results suggest there is a non-constant linear trend.

There is no obvious problems in QQ plot. But in residuals vs fitted plot, we can observe a slight horizontal “S” curve in the residuals. The red lowess line shows this.

(b)

```
1 library(nlme)
2 m2.corr <- nlme::gls(temp~year,
3                     correlation=corAR1(form=~year), data=aatemp)
4 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

```
1 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 doesn't contain 0. So we think the correlation is significant.

For the linear trend, the estimate of year is still significant at level $\alpha = 0.05$. However, its p-value is not notably small and the estimate is close to zero, so we think the linear trend is not as significant as that in (a).

(c)

```
1 #we start with poly(year, 10) and check whether the highest polynomial term is significant
2 m.try <- NULL
3 summ <- NULL
4 for (i in 10:1){
5   summ <- summary(m.try <- lm(temp ~ poly(year, i), data=aatemp))
6   deg.max <- summ$coefficients[i+1, "Pr(>|t|)"]
7   if (deg.max<0.05) break
8 }
9 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.

For the specific final “polynomial” model, it is with degree 5. And the regression equation is (ref: [click](#))

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

in which, β_i is as follow,

```
1 beta <- coef(m.try)
2 print(sapply(1:length(beta), function(i) {
3   paste("beta_", i-1, ": ", beta[i], sep = "")
4 })))
```

```
[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

```
1 ai <- attributes(z <- poly(aatemp$year, 5))$coefs$alpha
2 ni <- attributes(z)$coefs$norm2
3 F.i <- function(x, i){
4   if(i==0){
5     return (1/sqrt(ni[2]))
6   }
7   else if(i==1){
8     return ((x-ai[1])/sqrt(ni[3]))
9   }
10  else{
11    return (((x-ai[i])*sqrt(ni[i+1]))*F.i(x, i-1) -
12            ni[i+1]/sqrt(ni[i])*F.i(x, i-2))/sqrt(ni[i+2]))
13  }
```

```

14 }
15 # ai
16 ai

```

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

```

1 # ni
2 ni

```

```

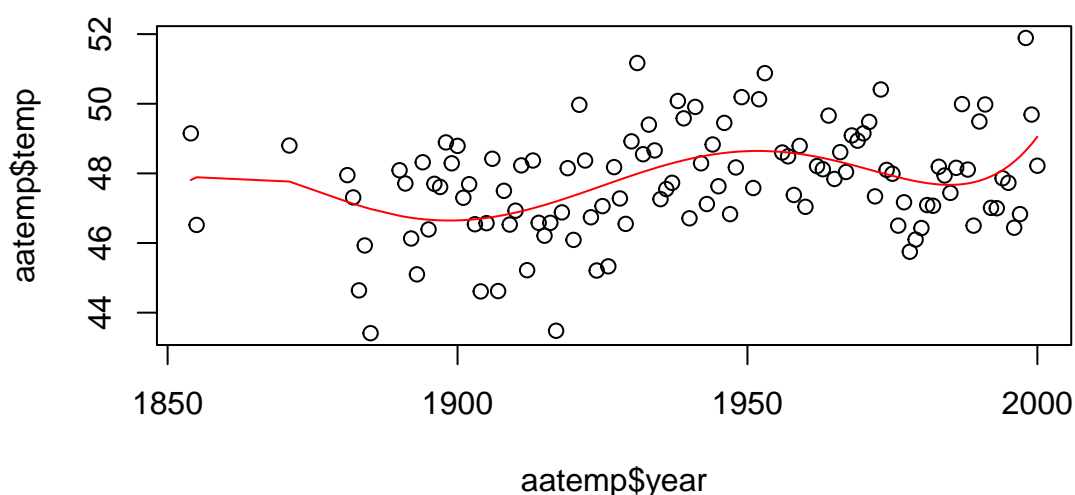
[1] 1.000000e+00 1.150000e+02 1.514022e+05 2.068262e+08 3.593395e+11
[6] 5.349960e+14 6.163661e+17

```

```

1 #plot the fitted model on top of the data
2 plot(aatemp$year, aatemp$temp)
3 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

```

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

```

```

1
60.07774

```

```

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

```

```

      fit      lwr      upr
1 60.07774 50.22436 69.93112

```

```

1 #predictive interval
2 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

```

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

```

```

      1
48.72478

```

```

1 #confidence interval
2 predict(m1, newdata = newx, interval = "confidence", level = 0.95)

```

```

      fit      lwr      upr
1 48.72478 48.0672 49.38237

```

```

1 #predictive interval
2 predict(m1, newdata = newx, interval = "prediction", level = 0.95)

```

```

      fit      lwr      upr
1 48.72478 45.74636 51.7032

```

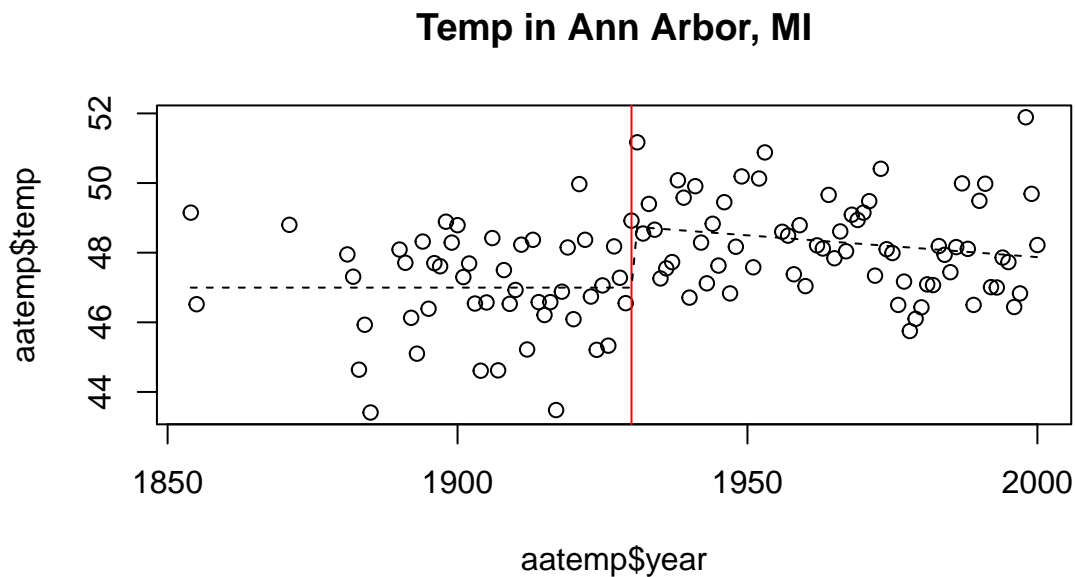
For this extrapolation, the polynomial model gives a greater prediction on the temperature in 2020 and a much wider confidence/prediction interval. This is due to the unstablility of polynomial methods for extrapolation (Runge's phenomenon). If we check the real data, NOAA records that the annual mean temperature is about 50 F. So the prediction given by polynomial model is unreliable at this point.

(e)

```

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

```



```

1 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

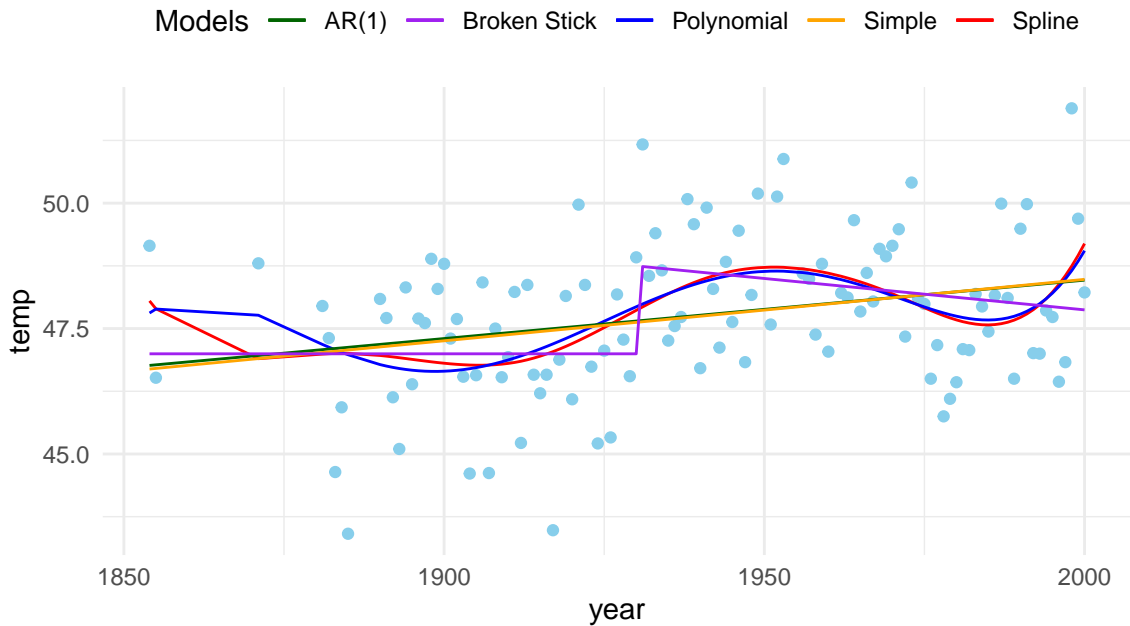
Multiple R-squared: 0.1954, Adjusted R-squared: 0.181

F-statistic: 13.6 on 2 and 112 DF, p-value: 5.167e-06

The R^2 of this model is 0.1954, much higher than the simple linear model. And the plot show different pattern before and after year 1930. So it seems that the claim is reasonable.

(f)

```
1 library(splines)
2 attach(aatemp)
3 knots <- c(1854, 1854, 1854, seq(1854, 2000, length.out=6),
4           2000, 2000, 2000)
5 byear <- splineDesign(knots, aatemp$year)
6 gs <- lm(aatemp$temp ~ byear-1)
7 df.plot <- data.frame(cbind(aatemp$year, aatemp$temp,
8                             gs$fitted.values,
9                             m.try$fitted.values,
10                            m2.corr$fitted,
11                            m1$fitted.values,
12                            py))
13 colnames(df.plot) <- c("year", "temp", "Spline",
14                        "Polynomial", "AR(1)",
15                        "Simple", "Broken Stick")
16 library(ggplot2)
17 # Create a ggplot object
18 gg <- ggplot(data = df.plot, aes(x = year)) +
19   geom_point(aes(y = temp), color = "skyblue") +
20   geom_line(aes(y = Spline, color = "Spline")) +
21   geom_line(aes(y = Polynomial, color = "Polynomial")) +
22   geom_line(aes(y = `AR(1)`, color = "AR(1)")) +
23   geom_line(aes(y = Simple, color = "Simple")) +
24   geom_line(aes(y = `Broken Stick`, color = "Broken Stick")) +
25   scale_color_manual(values = c("Spline" = "red", "Polynomial" = "blue", "AR(1)" = "darkblue", "Simple" = "green", "Broken Stick" = "black")) +
26   labs(color = "Line") +
27   theme_minimal() +
28   theme(legend.position = "top") +
29   guides(color = guide_legend(title = "Models"))
30 gg
```

Generally, the cubic spline fit looks similar to polynomial fit of degree 5. And after year 1900, these two fitted lines are close.

```
1 summary(gs)
```

Call:

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

Residuals:

Min	1Q	Median	3Q	Max
-3.5957	-0.9526	-0.1826	0.9826	3.2389

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
byear1	48.0512	1.0547	45.56	<2e-16 ***
byear2	46.4530	2.4594	18.89	<2e-16 ***
byear3	47.4927	1.1978	39.65	<2e-16 ***
byear4	46.1263	0.7633	60.43	<2e-16 ***
byear5	49.1577	0.7096	69.27	<2e-16 ***
byear6	48.4443	0.8667	55.89	<2e-16 ***
byear7	46.7959	0.8893	52.62	<2e-16 ***
byear8	49.1936	0.7729	63.65	<2e-16 ***

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

Residual standard error: 1.404 on 107 degrees of freedom

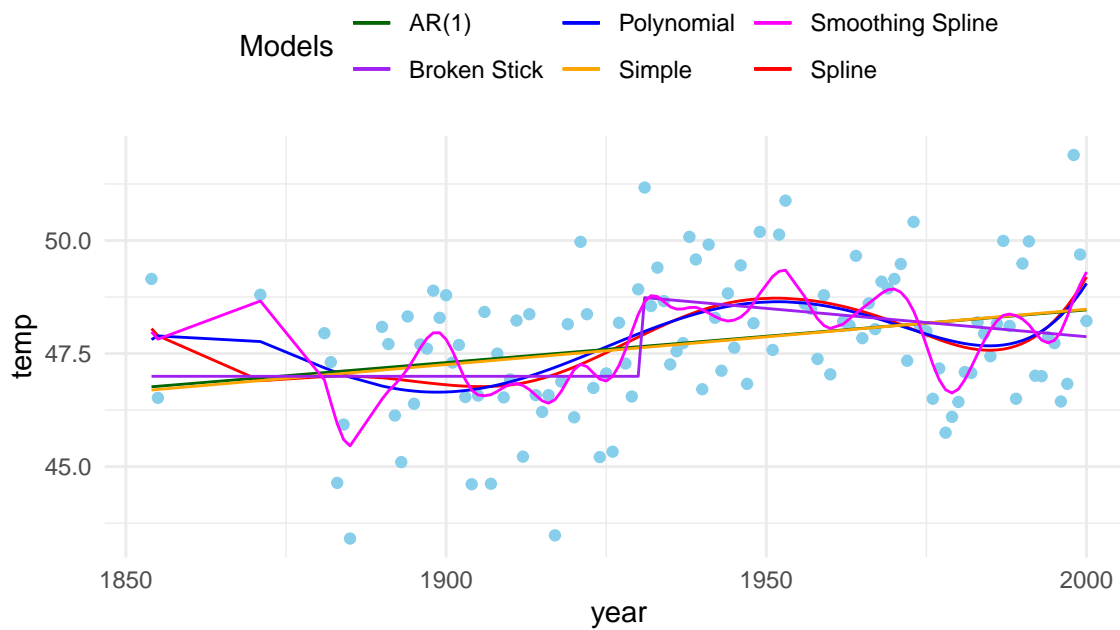
Multiple R-squared: 0.9992, Adjusted R-squared: 0.9991

F-statistic: 1.662e+04 on 8 and 107 DF, p-value: < 2.2e-16

From the result, this model fit much better than the straight-line model. Even it is like a perfect fit since R^2 is very close to 1.

(g)

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



From the