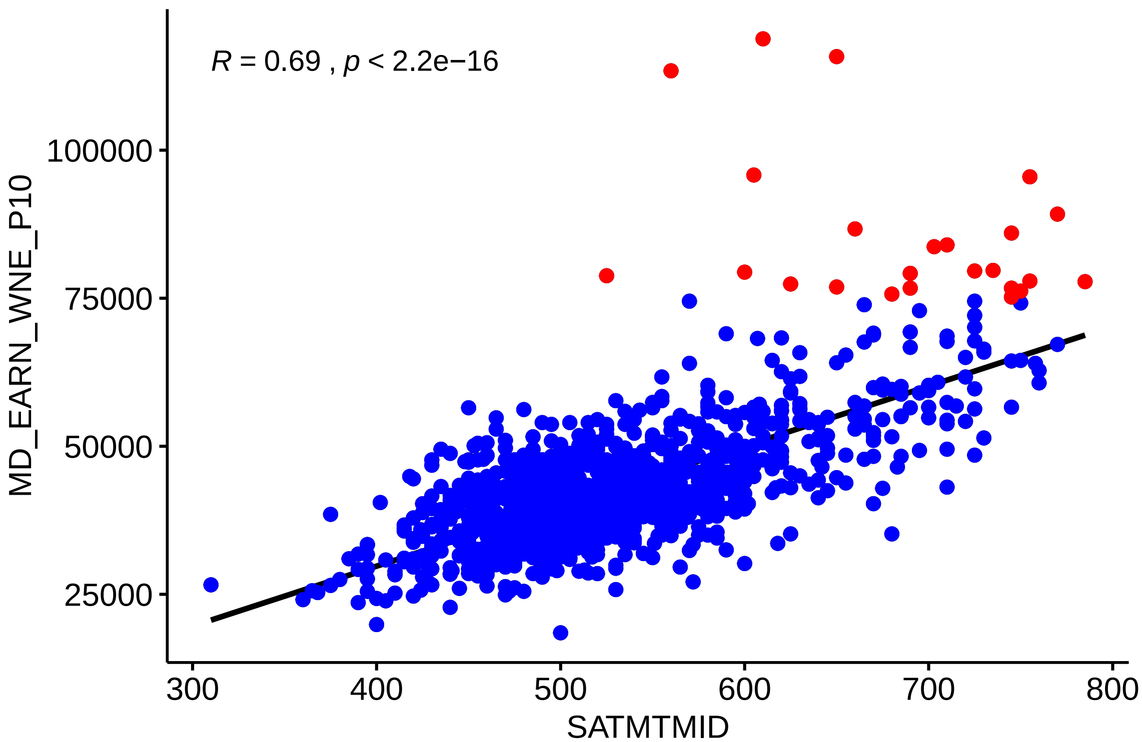
hw5_q2

10/02/2019

a) median SAT scores versus the median earnings after 10 years

Here we scatterplot the approximately linear relationship (r^2 > 0.6) between these variables. First we removed the rows with missing values. Based on the line of best fit, outliers appeared to have earnings greater than 75000 (red points). It seems the outliers belong to graduates of medical schools or prestigious schools who can earn more than average (printed below). I removed these outliers.

```
# filter out missing data
filt <- data %>%
  select(SATMTMID, MD_EARN_WNE_P10, INSTNM) %>%
  na.omit
ggscatter(filt, x = "SATMTMID", y = "MD_EARN_WNE_P10", #xlab = FALSE, ylab = FALSE,
               cor.coef = TRUE,  add = "reg.line", cor.method = "pearson")+
  geom_point(size=2,color = ifelse(filt$MD_EARN_WNE_P10 > 75000, "red", "blue"))
```




```
filt[filt$MD_EARN_WNE_P10 > 75000,"INSTNM"]
```

```
##  [1] California Institute of Technology
##  [2] Colorado School of Mines
##  [3] Georgetown University
##  [4] Rose-Hulman Institute of Technology
##  [5] Maine Maritime Academy
##  [6] Babson College
##  [7] Bentley University
##  [8] Harvard University
##  [9] MCPHS University
```

```
## [10] Massachusetts Institute of Technology
## [11] Kettering University
## [12] St Louis College of Pharmacy
## [13] Princeton University
## [14] Stevens Institute of Technology
## [15] Albany College of Pharmacy and Health Sciences
## [16] Columbia University in the City of New York
## [17] Rensselaer Polytechnic Institute
## [18] Duke University
## [19] Carnegie Mellon University
## [20] Lehigh University
## [21] University of Pennsylvania
## [22] University of the Sciences
## [23] Stanford University
## [24] DigiPen Institute of Technology
## 7535 Levels: A & W Healthcare Educators ...
```

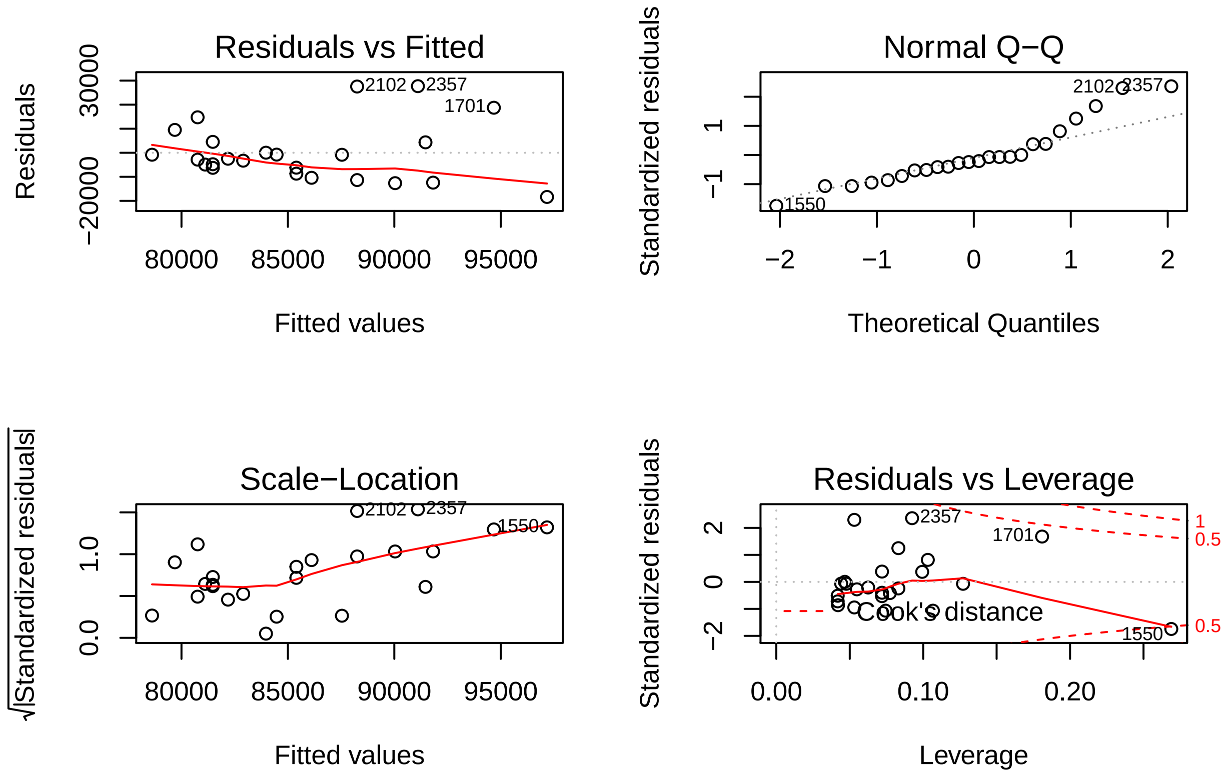
b) median earnings and SAT math scores

Here we fit an ordinary linear model of median earnings vs median SAT math scores. The diagnostic plots suggest the fit is not very linear:

- The residuals vs fitted plot shows residuals have a downward trend rather than being equally spread relative to the fitted values.
- the QQ plot has many points which are not well aligned on the y=x line indicating some residuals are not normally distributed.
- the plot of residuals vs. leverage shows a couple of potentially problematic outliers with high residuals and/or leverage, lying close to the Cook's distance curves.

Overall the model relationship is not very linear (Adjusted R-squared less than 0.2) and the diagnostic plots show some problems with the fit.

```
##
## Call:
## lm(formula = MD_EARN_WNE_P10 ~ SATMTMID, data = filt_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -18374  -6878  -2672   4384  27693
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 134643.81   25133.80   5.357 2.23e-05 ***
## SATMTMID       -71.37      36.45  -1.958    0.063 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12320 on 22 degrees of freedom
## Multiple R-squared:  0.1484, Adjusted R-squared:  0.1097
## F-statistic: 3.835 on 1 and 22 DF,  p-value: 0.06299
```



c) Nonlinear fits

We fit a second-order and third-order polynomial here and find that neither of these provide a statistically significant improvement over the original first order linear model by partial F test (no significant terms at $p < 0.05$). The diagnostic plots also do not show improvements from previous ones. For example the third order polynomial diagnostic plot of residuals versus leverage (shown below) even see the appearance of an outlier that has moved further outside of the Cook's distance lines. This suggests the nonlinear fits are comparable to the linear fit.

```r
filt_data <- data[data$MD_EARN_WNE_P10 >75000,] %>%
  select(SATMTMID, MD_EARN_WNE_P10, INSTNM) %>% na.omit()

mod2 <- lm(MD_EARN_WNE_P10 ~ poly(SATMTMID, 2), data = filt_data)
summary(mod2)
```

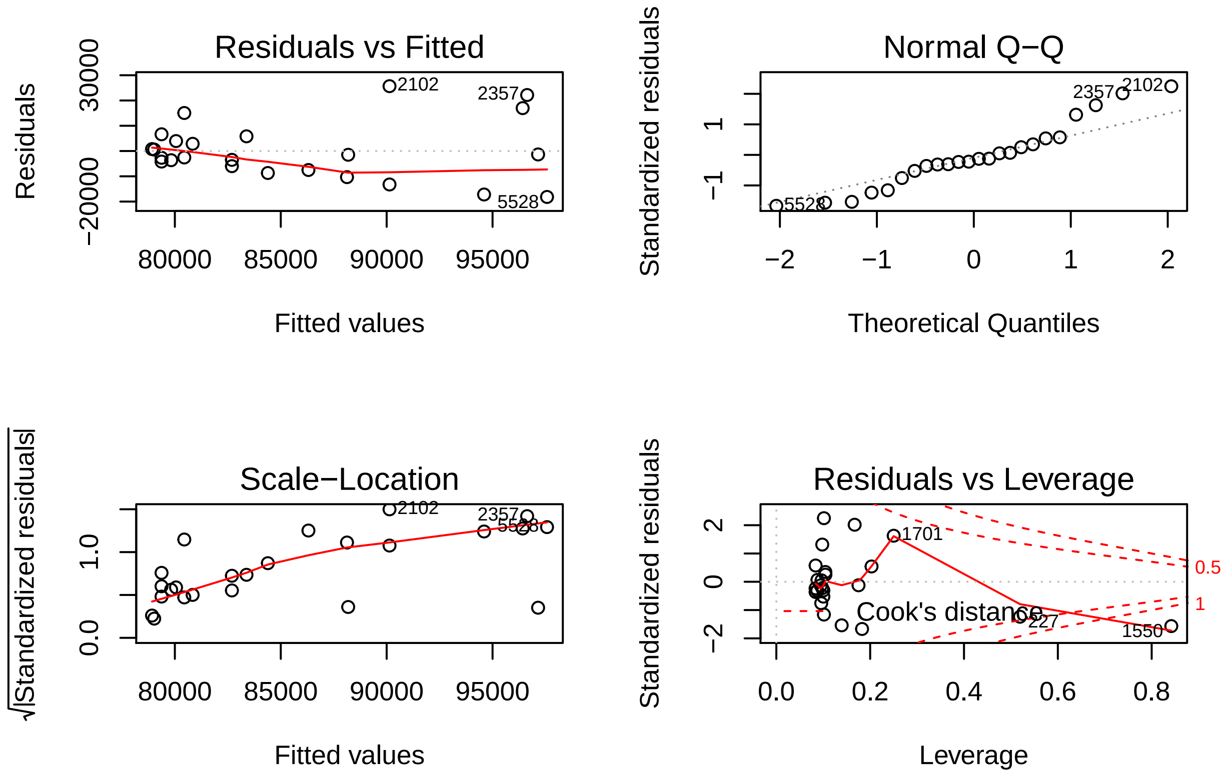
```
##
## Call:
## lm(formula = MD_EARN_WNE_P10 ~ poly(SATMTMID, 2), data = filt_data)
##
## Residuals:
##    Min      1Q Median     3Q    Max
## -17180  -7204  -2562   4307  27509
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)            85671       2573  33.300   <2e-16 ***
## poly(SATMTMID, 2)1    -24129      12604  -1.914   0.0693 .
## poly(SATMTMID, 2)2     -1999      12604  -0.159   0.8755
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 12600 on 21 degrees of freedom
## Multiple R-squared:  0.1495, Adjusted R-squared:  0.06846
## F-statistic: 1.845 on 2 and 21 DF,  p-value: 0.1827
```

```r
mod3 <- lm(MD_EARN_WNE_P10 ~ poly(SATMTMID, 3), data = filt_data)
summary(mod3)
```

```
##
## Call:
## lm(formula = MD_EARN_WNE_P10 ~ poly(SATMTMID, 3), data = filt_data)
##
## Residuals:
##    Min      1Q Median     3Q    Max
## -18172  -6372  -2014   4412  25665
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)           85671       2459  34.833   <2e-16 ***
## poly(SATMTMID, 3)1   -24129      12049  -2.003   0.0590 .
## poly(SATMTMID, 3)2    -1999      12049  -0.166   0.8699
## poly(SATMTMID, 3)3    20793      12049   1.726   0.0998 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12050 on 20 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.1487
## F-statistic: 2.339 on 3 and 20 DF,  p-value: 0.1042
```

```r
# plot residuals and diagnostics
par(mfrow = c(2, 2))
plot(mod3)
```




```r
# Use partial F tests to compare the models
anova(mod, mod2)
```

```
## Analysis of Variance Table
##
## Model 1: MD_EARN_WNE_P10 ~ SATMTMID
## Model 2: MD_EARN_WNE_P10 ~ poly(SATMTMID, 2)
##   Res.Df        RSS Df Sum of Sq      F Pr(>F)
## 1     22 3339816358
## 2     21 3335821103  1   3995256 0.0252 0.8755
```

```r
anova(mod, mod3)
```

```
## Analysis of Variance Table
##
## Model 1: MD_EARN_WNE_P10 ~ SATMTMID
## Model 2: MD_EARN_WNE_P10 ~ poly(SATMTMID, 3)
##   Res.Df        RSS Df Sum of Sq      F Pr(>F)
## 1     22 3339816358
## 2     20 2903463041  2 436353317 1.5029 0.2466
```

d) Cross validation

```r
library(modelr)
cv <- crossv_kfold(filt_data)

model1 <- map(cv$train, ~lm(MD_EARN_WNE_P10 ~ SATMTMID, data =.))
model2 <- map(cv$train, ~lm(MD_EARN_WNE_P10 ~ poly(SATMTMID, 2), data =.))
model3 <- map(cv$train, ~lm(MD_EARN_WNE_P10 ~ poly(SATMTMID, 3), data =.))
```

```r
# Use cross-validation to estimate the squared-error loss of each of your models.
errs1 <- map2_dbl(model1, cv$test, mse)
errs2 <- map2_dbl(model2, cv$test, mse)
errs3 <- map2_dbl(model3, cv$test, mse)

# print mean mse
mean(as.numeric(errs1))
```

```
## [1] 173386251
```

```r
mean(as.numeric(errs2))
```

```
## [1] 265897174
```

```r
mean(as.numeric(errs3))
```

```
## [1] 266892309
```

Whether or not the F tests suggested the polynomials help, you can also compare the pre- dictive performance of each model. Sometimes variables that are not statistically signifi- cant can improve predictive performance. Use cross-validation to estimate the squared- error loss of each of your models. (Fit a new model to each training set.) Compare the results to what you got using F tests. Why could the results differ?

(For K-fold cross-validation, the modelr package has a crossv_kfold function that auto- matically divides your data up into folds, and gives you lists of the training and test sets.

We checked the cross validation RSS and found that the predictive performance of the original linear model is better than the performance of the order 2 polynomial, which is in turn better than performance of the order 3 polynomial. This is different from our conclusion from the F tests.

e) Fit a smoothing spline

For the same model as above we compared the following three spline models:

- cv.fit: spar picked by automatic cross-validation
- half_cv.fit: spar set to be half as big as R picked
- half.fit: spar set to be halfway between cv.fit and 1

```r
library(splines)
# fit the splines
cv.fit <- smooth.spline(filt_data$SATMTMID,
                        filt_data$MD_EARN_WNE_P10)
spar_fit <- cv.fit$spar #spar= 1.499929
spar_fit
```

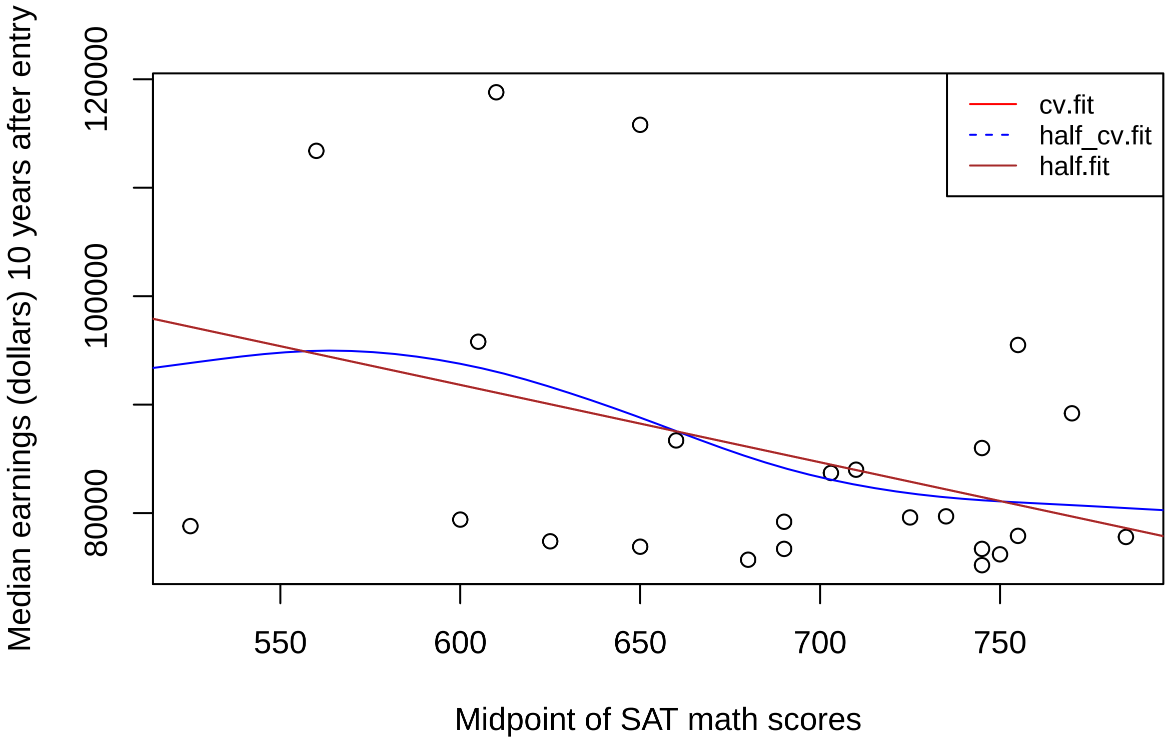
```
## [1] 1.499929
```

```r
half_cv.fit <- smooth.spline(filt_data$SATMTMID,
                             filt_data$MD_EARN_WNE_P10,
                             spar = spar_fit/2.0)
half.fit <- smooth.spline(filt_data$SATMTMID,
                          filt_data$MD_EARN_WNE_P10,
                          spar = (spar_fit-1.0)/2.0 + 1.0)

# Plot predictions from the three models on one scatterplot and compare them visually.
# sat math scores can range from 200 to 800
satmids <- seq(200, 800, length.out=100)

plot(filt_data[,1:2], xlab="Midpoint of SAT math scores",
```

```
     ylab="Median earnings (dollars) 10 years after entry")
lines(satmids, predict(cv.fit, satmids)$y, col="red")
lines(satmids, predict(half_cv.fit, satmids)$y, col="blue")
lines(satmids, predict(half.fit, satmids)$y, col="brown")
legend(x= "topright", legend=c("cv.fit", "half_cv.fit", "half.fit"),
       col=c("red", "blue", "brown"), lty=1:2, cex=0.8)
```



It's clear that whereas half.fit and cv.fit are very similar, half_cv.fit is a less linear model with higher variance.

(f) Use cross-validation to estimate the error of the three splines.

Which is worse, too much bias or too much variance? (Which fit corresponds to high bias, and which corresponds to high variance?)

It seems that high bias wins over high variance in this situation, since the average mse of the half_cv spline (which has the smallest spar and higher variance) is the largest.

```
library(modelr)
# make splits
k=5
folds <- cut(seq(1,nrow(filt_data)),breaks=k,labels=FALSE)
mse1 <- vector(mode = "numeric", length = k)
mse2 <- vector(mode = "numeric", length = k)
mse3 <- vector(mode = "numeric", length = k)
for(i in 1:k){
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- filt_data[testIndexes, 1:2]
  trainData <- filt_data[-testIndexes, 1:2]
  # get 3 model on specific train fold
  cv.fit <- smooth.spline(trainData$SATMTMID,
                          trainData$MD_EARN_WNE_P10,
```

```
                        spar = spar_fit)
  half_cv.fit <- smooth.spline(trainData$SATMTMID,
                              trainData$MD_EARN_WNE_P10,
                              spar = spar_fit/2.0)
  half.fit <- smooth.spline(trainData$SATMTMID,
                           trainData$MD_EARN_WNE_P10,
                           spar = (spar_fit-1.0)/2.0 + 1.0)
  # extract model performance on test fold
  y <- testData$MD_EARN_WNE_P10
  yhat1 <- predict(cv.fit, testData$SATMTMID)$y
  yhat2 <- predict(half_cv.fit,testData$SATMTMID)$y
  yhat3 <- predict(half.fit, testData$SATMTMID)$y

  # get the mse
  mse1[i] <- mean((y-yhat1)^2)
  mse2[i] <- mean((y-yhat2)^2)
  mse3[i] <- mean((y-yhat3)^2)
  }
# Compare the average errors of the three.

mean(as.numeric(mse1))
```

```
## [1] 177254822
```

```
mean(as.numeric(mse2))
```

```
## [1] 210967799
```

```
mean(as.numeric(mse3))
```

```
## [1] 177272169
```