QUESTION 2: Evaluating Prediction Accuracy

(a) Using the information in the `toronto.covid` dataframe, construct a new dataframe called `tor` containing the date and daily new COVID-19 case numbers for the period March 1, 2020 to December 31, 2020 inclusive. In particular, `tor` should contain $N = 306$ rows (one for each day) and the following three columns:

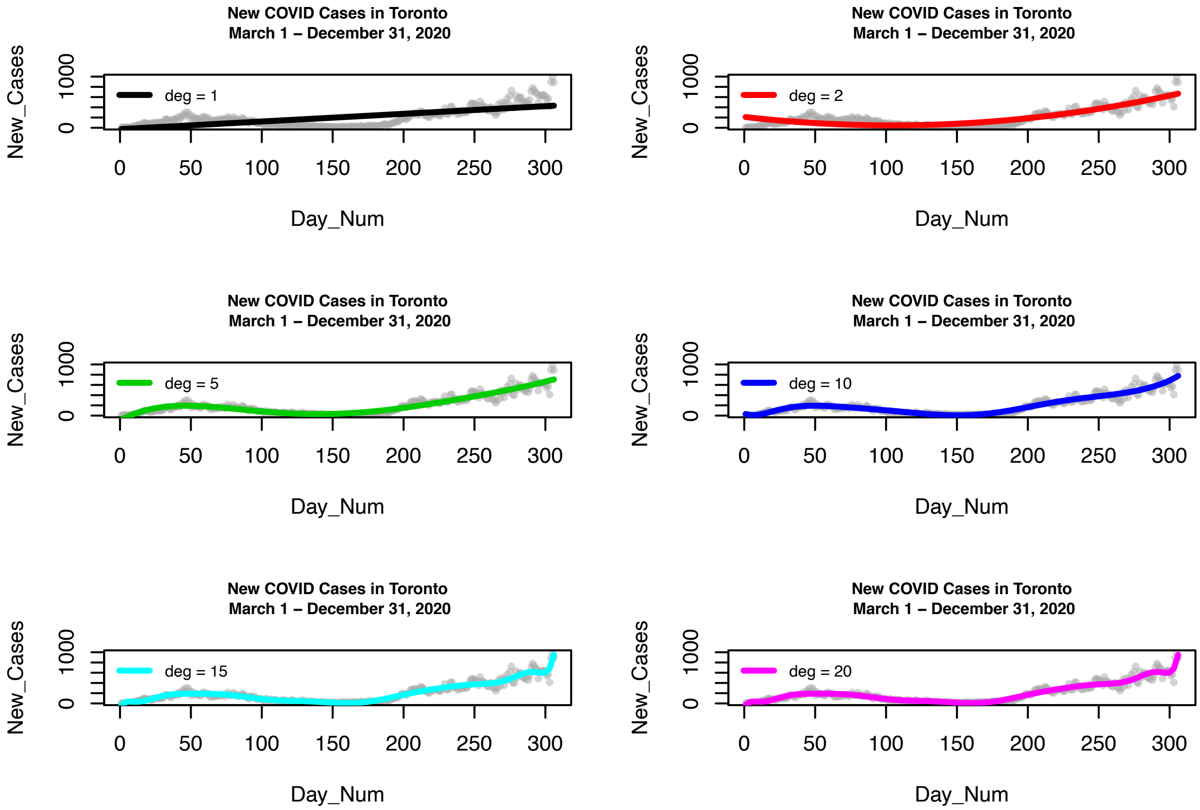
```r
date <- toronto.covid$ï..Episode.Date[2:307]
date <- rev(date)
days <- seq(1, 306)
cases <- toronto.covid$Case.Count[2:307]
cases <- rev(cases)
tor <- data.frame(date,days,cases)
colnames(tor) <- c("Date", "Day_Num", "New_Cases")

head(tor, 15)
```

```
##       Date Day_Num New_Cases
## 1   3/1/20       1         6
## 2   3/2/20       2         7
## 3   3/3/20       3        12
## 4   3/4/20       4         9
## 5   3/5/20       5        10
## 6   3/6/20       6        11
## 7   3/7/20       7         9
## 8   3/8/20       8        10
## 9   3/9/20       9        25
## 10 3/10/20      10        31
## 11 3/11/20      11        32
## 12 3/12/20      12        45
## 13 3/13/20      13        54
## 14 3/14/20      14        61
## 15 3/15/20      15        57
```

(b) Construct a 3×2 grid of scatter plots where each plot is a recreation of the one above, but with polynomials of degrees 1, 2, 5, 10, 15, and 20, respectively overlaid. Use the `getmuhat` function defined on pages 7-9 to estimate these polynomial predictor functions, and use all $N = 306$ observations in `tor` to do so. Use a different colour for each of the different degrees, and use a legend to indicate which degree polynomial is visualized in each plot.

```r
xydata <- data.frame(tor["Day_Num"], tor["New_Cases"])
colnames(xydata) <- c("x", "y")

par(mfrow=c(3,2))

# deg = 1
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
muhat1 <- getmuhat(xydata,1)
curve(muhat1, lwd = 3, add = TRUE)
```

```
legend("topleft", legend = c("deg = 1"), lwd = 3, cex = 0.8, bty = "n")

# deg = 2
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
muhat2 <- getmuhat(xydata,2)
curve(muhat2, add = TRUE, lwd = 3, col = 2)
legend("topleft", legend = c("deg = 2"), lwd = 3, cex = 0.8, bty = "n", col = 2)

# deg = 5
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
muhat5 <- getmuhat(xydata,5)
curve(muhat5, add = TRUE, lwd = 3, col = 3)
legend("topleft", legend = c("deg = 5"), lwd = 3, cex = 0.8, bty = "n", col = 3)

# deg = 10
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
muhat10 <- getmuhat(xydata,10)
curve(muhat10, add = TRUE, lwd = 3, col = 4)
legend("topleft", legend = c("deg = 10"), lwd = 3, cex = 0.8, bty = "n", col = 4)

# deg = 15
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
muhat15 <- getmuhat(xydata,15)
curve(muhat15, add = TRUE, lwd = 3, col = 5)
legend("topleft", legend = c("deg = 15"), lwd = 3, cex = 0.8, bty = "n", col = 5)

# deg = 20
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
muhat20 <- getmuhat(xydata,15)
curve(muhat20, add = TRUE, lwd = 3, col = 6)
legend("topleft", legend = c("deg = 20"), lwd = 3, cex = 0.8, bty = "n", col = 6)
```



(c) Use the following code to generate $M = 50$ samples $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{50}$ of size $n = 100$. Note that doing so will require you to first run the `getSampleComp` and `getXYSample` functions defined on XX. (This is not for points).

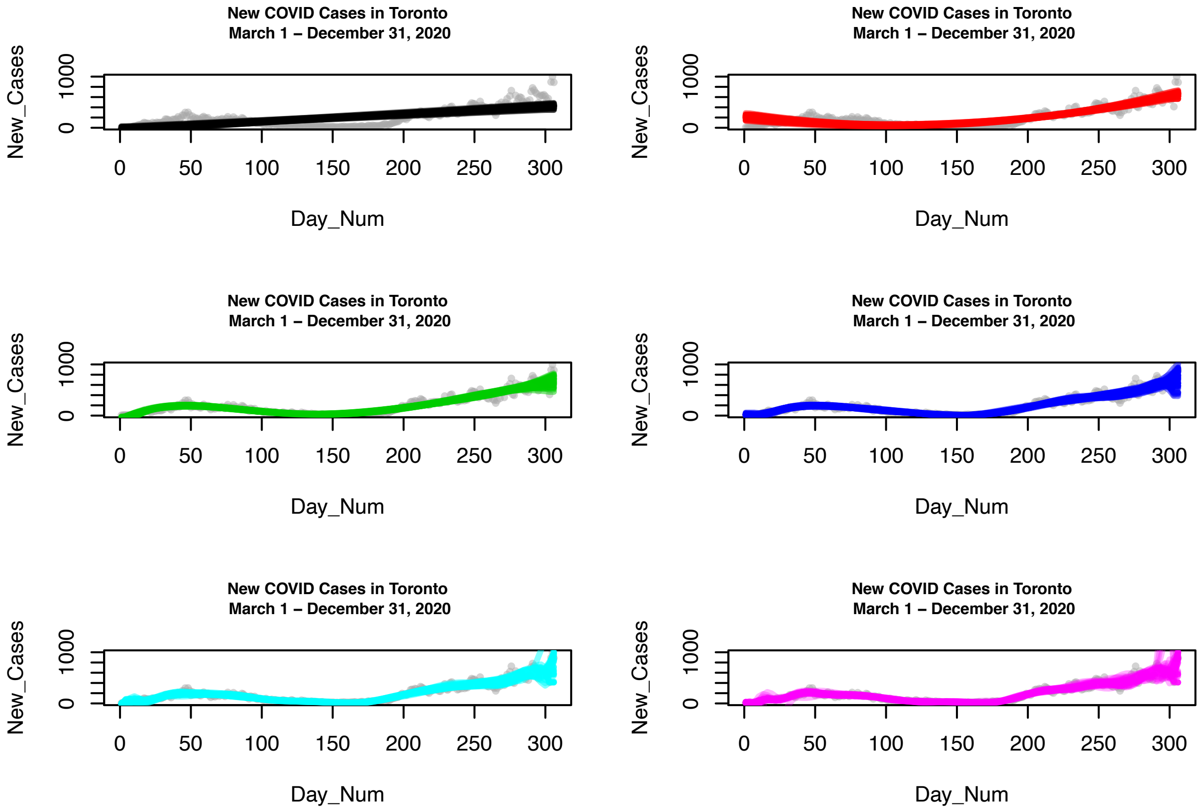
Fit polynomials of degree 1, 2, 5, 10, 15, and 20 to every sample.

```r
M <- 50
n <- 100
set.seed(341)
samps <- lapply(1:M, FUN= function(i){getSampleComp(tor, n)})
Ssamples <- lapply(samps, FUN= function(Si){getXYSample("Day_Num", "New_Cases", Si, tor)})
Tsamples <- lapply(samps, FUN= function(Si){getXYSample("Day_Num", "New_Cases", !Si, tor)})

# Fit a polynomial of the given complexity/degree to every sample
# and save the results in a list
muhats1 <- lapply(Ssamples, getmuhat, complexity = 1)
muhats2 <- lapply(Ssamples, getmuhat, complexity = 2)
muhats5 <- lapply(Ssamples, getmuhat, complexity = 5)
muhats10 <- lapply(Ssamples, getmuhat, complexity = 10)
muhats15 <- lapply(Ssamples, getmuhat, complexity = 15)
muhats20 <- lapply(Ssamples, getmuhat, complexity = 20)
```

(d) Create another 3×2 grid of plots like in part (b), except this time:

```r
par(mfrow=c(3,2))
```

```r
# deg = 1
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
for (i in 1:M) {
  curveFn <- muhats1[[i]]
  curve(curveFn, add = TRUE, lwd = 3, cex = 0.8, bty = "n", col = adjustcolor(1, 0.5))
}

# deg = 2
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
for (i in 1:M) {
  curveFn <- muhats2[[i]]
  curve(curveFn, add = TRUE, lwd = 3, cex = 0.8, bty = "n", col = adjustcolor(2, 0.5))
}

# deg = 5
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
for (i in 1:M) {
  curveFn <- muhats5[[i]]
  curve(curveFn, add = TRUE, lwd = 3, cex = 0.8, bty = "n", col = adjustcolor(3, 0.5))
}

# deg = 10
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
for (i in 1:M) {
  curveFn <- muhats10[[i]]
  curve(curveFn, add = TRUE, lwd = 3, cex = 0.8, bty = "n", col = adjustcolor(4, 0.5))
}

# deg = 15
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
for (i in 1:M) {
  curveFn <- muhats15[[i]]
  curve(curveFn, add = TRUE, lwd = 3, cex = 0.8, bty = "n", col = adjustcolor(5, 0.5))
}

# deg = 20
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
for (i in 1:M) {
  curveFn <- muhats20[[i]]
  curve(curveFn, add = TRUE, lwd = 3, cex = 0.8, bty = "n", col = adjustcolor(6, 0.3))
}
```



(e) Using the $M = 50$ samples of size $n = 100$ generated in part (c), calculate the APSE (and each of its components) for degrees `0:20`. In particular, print out a table that shows for each degree `apse`, `var_mutilde`, `bias2` and `var_y`. **Note:** Use the `apse_all` function defined on pages 7-9. Doing so will require the `getmubar` and `gettauFun` functions also defined on pages 7-9.

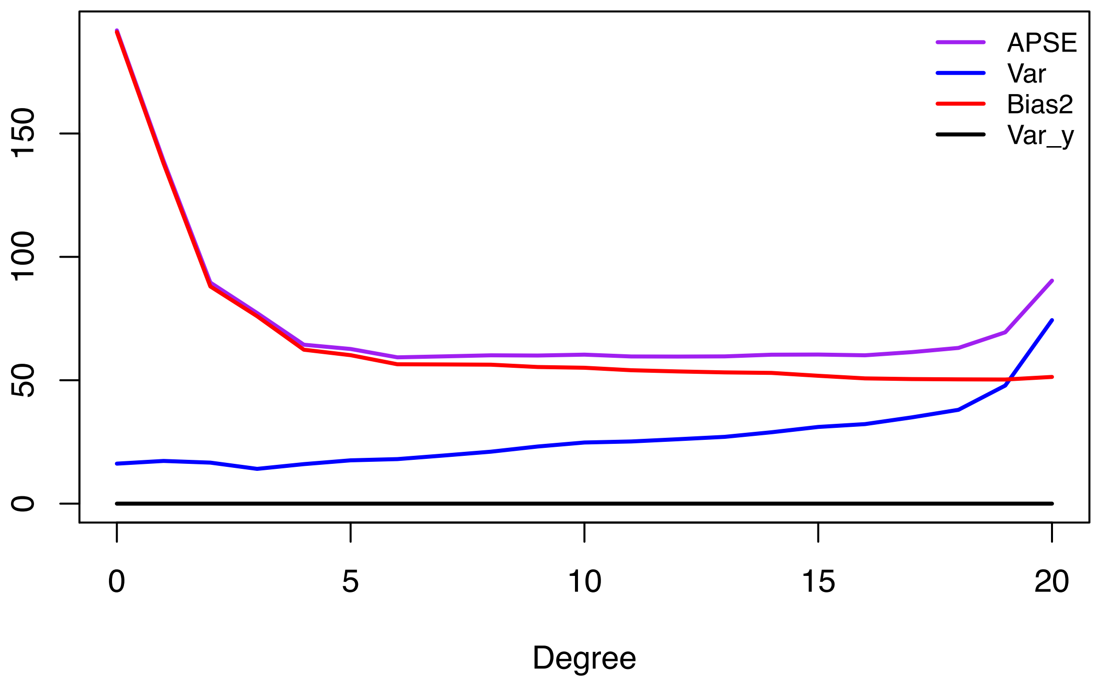
```
tau.x <- gettauFun(xydata, "x", "y")
degrees <- 0:20
apse_vals <- sapply(degrees, FUN = function(complexity) {
  apse_all(Ssamples, Tsamples, complexity = complexity, tau = tau.x)
})

# print results
t(rbind(degrees, aspe = round(apse_vals, 5)))
```

```
##       degrees      apse var_mutilde     bias2 var_y
##  [1,]       0 36780.539    262.6077 36517.932     0
##  [2,]       1 19297.953    299.2158 18998.738     0
##  [3,]       2  8023.492    276.1977  7747.294     0
##  [4,]       3  5966.170    198.5890  5767.581     0
##  [5,]       4  4147.090    256.5894  3890.501     0
##  [6,]       5  3927.015    308.0987  3618.916     0
##  [7,]       6  3514.476    325.7475  3188.728     0
##  [8,]       7  3562.171    380.9529  3181.218     0
##  [9,]       8  3613.195    443.0774  3170.117     0
## [10,]       9  3601.815    535.9507  3065.864     0
## [11,]      10  3645.402    613.6563  3031.746     0
```

```
## [12,]      11  3557.267    634.7055  2922.562     0
## [13,]      12  3552.039    681.6326  2870.407     0
## [14,]      13  3560.835    732.7227  2828.112     0
## [15,]      14  3643.028    836.8560  2806.172     0
## [16,]      15  3650.072    966.1810  2683.891     0
## [17,]      16  3613.607   1038.2253  2575.382     0
## [18,]      17  3769.307   1221.5320  2547.775     0
## [19,]      18  3981.631   1445.2355  2536.396     0
## [20,]      19  4819.604   2289.9750  2529.629     0
## [21,]      20  8167.623   5531.3125  2636.310     0
```

(f) Using your results from part (e) construct a plot whose x-axis is degree and which has four lines: one for each of `apse`, `var_mutilde`, `bias2` and `var_y`. Specifically, and for interpretability, plot `sqrt(apse)`, `sqrt(var_mutilde)`, `sqrt(bias2)` and `sqrt(var_y)` vs. `degree`. Be sure to distinguish the lines with different colours and a legend. Briefly describe the trends you see in the plot.

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

plot(degrees, sqrt(apse_vals[1, ]), xlab = "Degree",
     ylab = "", type = "l", col = "purple", lwd = 2, ylim = c(0, max(sqrt(apse_vals))))
lines(degrees, sqrt(apse_vals[2, ]), xlab = "Degree",
      ylab = "", col = "blue", lwd = 2)
lines(degrees, sqrt(apse_vals[3, ]), xlab = "Degree",
      ylab = "", col = "red", lwd = 2)
lines(degrees, sqrt(apse_vals[4, ]), xlab = "Degree",
      ylab = "", col = "black", lwd = 2)
legend("topright", legend = c("APSE", "Var", "Bias2", "Var_y"),
       col = c("purple","blue","red","black"), lwd = 2, bty = "n", cex = 0.8)
```



APSE looks to be concaved upwards as it is decreasing within [0,5] then starts the increase mostly after

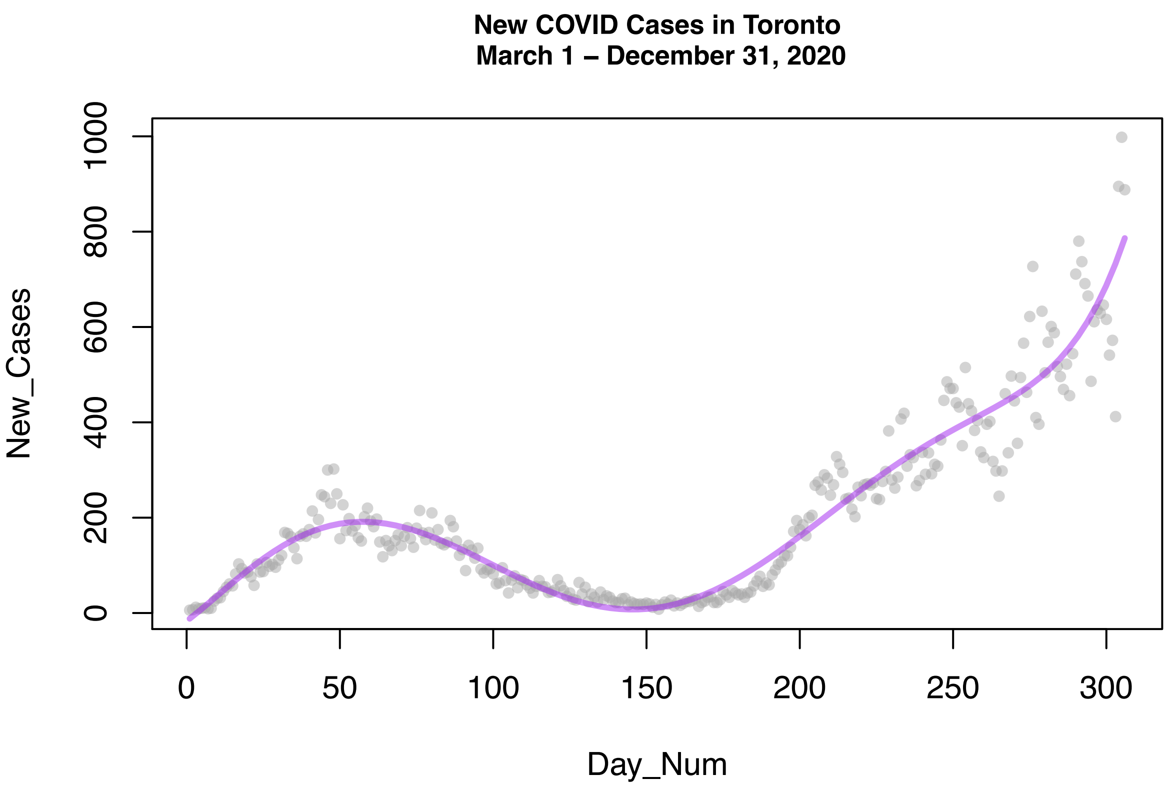
Var looks to have a positive trend as it is increasing mostly

Bias2 looks to have a negative trend as it looks like it is exponentially decreasing

Var_y looks to be constantly at 0

(g) Based on your findings in parts (e) and (f), which degree polynomial has the best predictive accuracy? Construct a scatter plot – like the ones from (b) and (d) – but this time create just one plot, and overlay just the polynomial predictor function with the degree you identified as best. Use all of the data in `tor` to estimate this predictor function.

```
plot(xydata, main = "New COVID Cases in Toronto\n March 1 - December 31, 2020",
     xlab = "Day_Num", ylab = "New_Cases", cex = 0.8, cex.main = 0.8,
     col = adjustcolor("darkgrey", 0.5), pch = 16)
muhat.best <- getmuhat(xydata, 7)
curve(muhat.best, add = TRUE, lwd = 3, cex = 0.8, bty = "n", col = adjustcolor("purple", 0.5))
```



The polynomial of degree 7 looks to have the best predictive accuracy as it has the lowest APSE.