Supercar Topspeeds

Antony Sikorski

SuperCar Topspeeds Project Code

I put the code into a Markdown File so that the graphs and outputs can be seen.

```
#Installing packages, calling necessary libraries
#install.packages("deSolve")
#install.packages("dplyr")
#install.packages("ggplot2")
#install.packages("growthcurver")
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(deSolve)
```

```
## Warning: package 'deSolve' was built under R version 4.2.2
```

```
library(ggplot2)
```

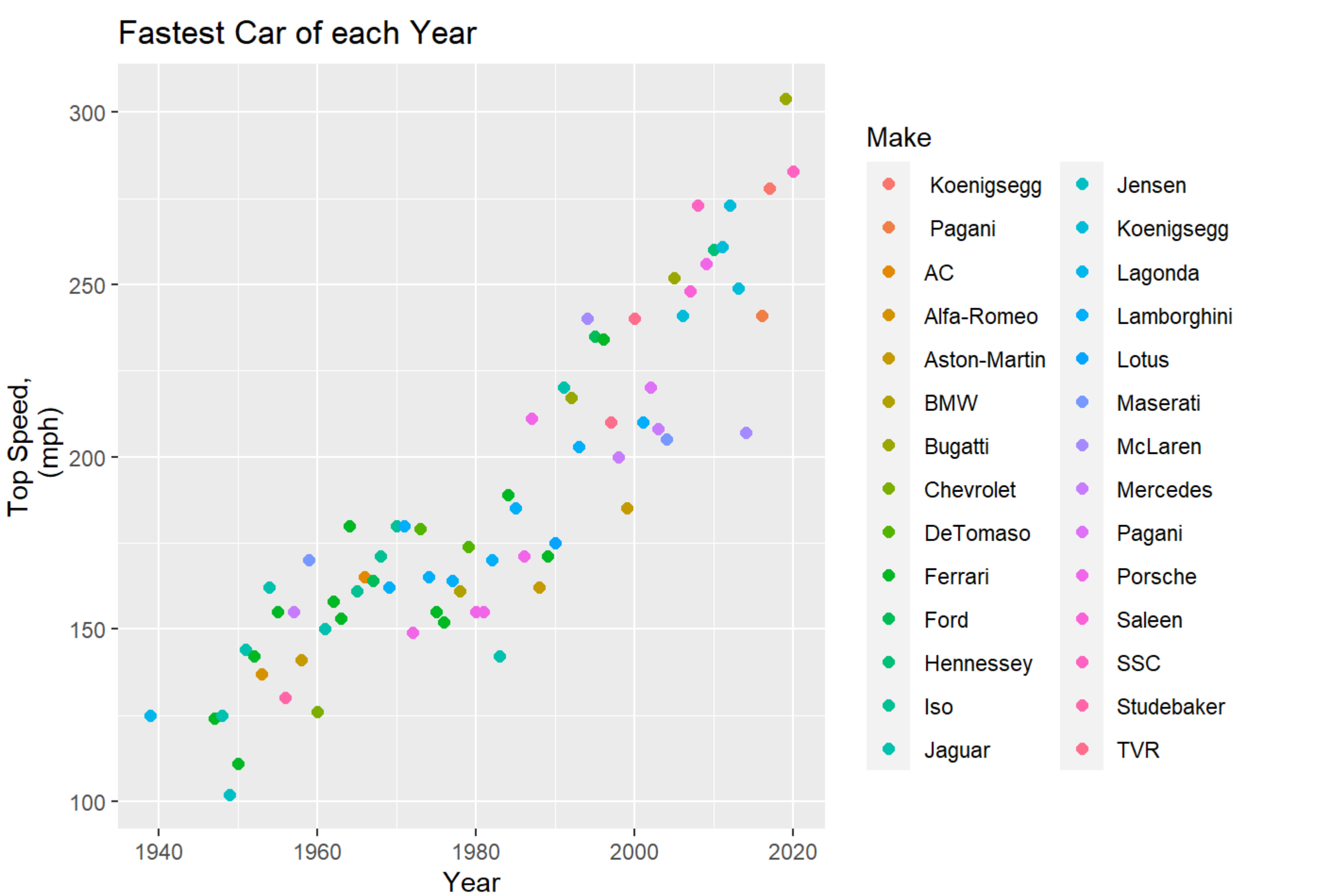
```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
library(growthcurver)
```

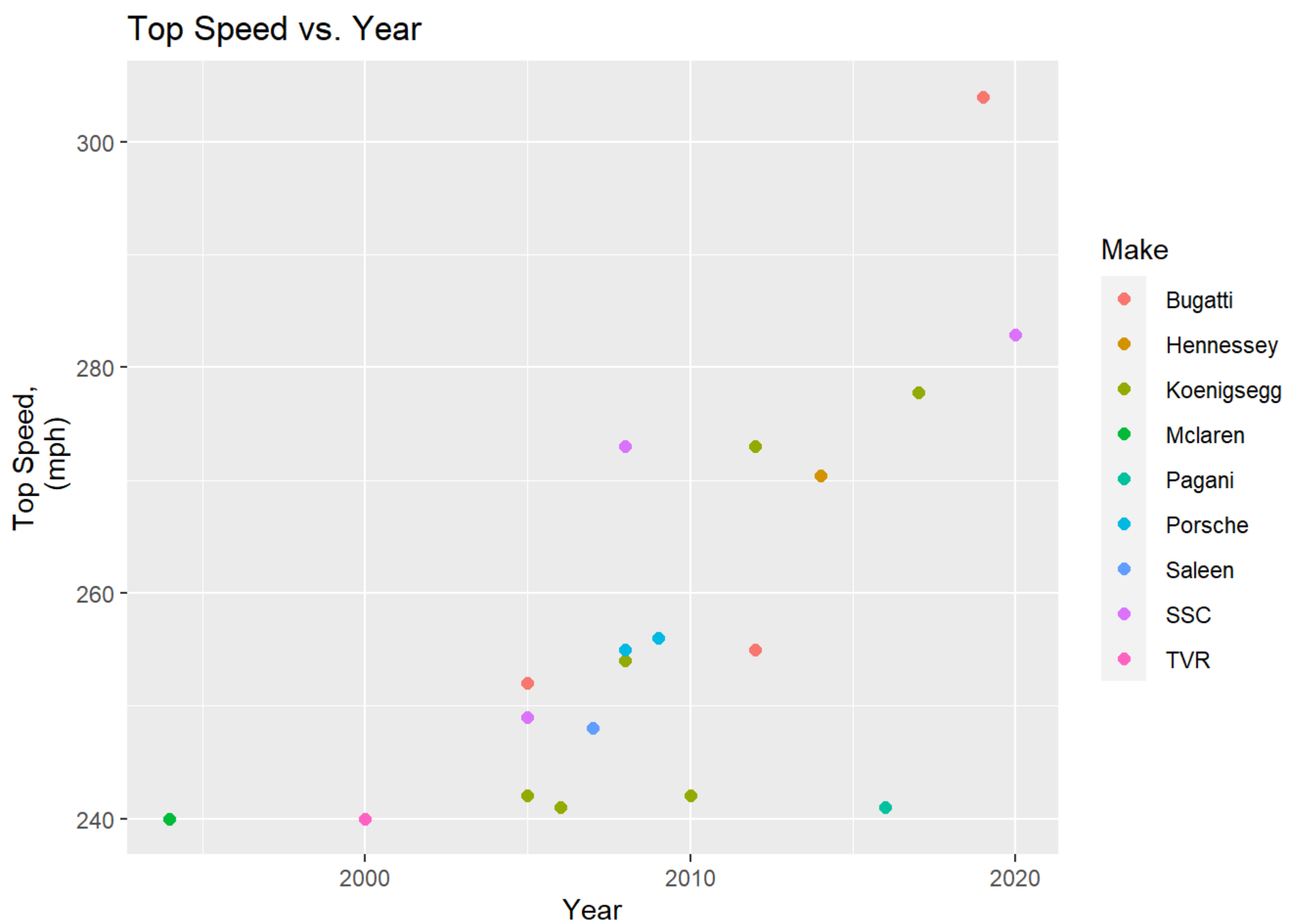
```
## Warning: package 'growthcurver' was built under R version 4.2.2
```

Data cleaning and exploration:

```
#Loading in and filtering original dataset
df <- read.delim("supercardata.txt", sep = ",")
cleaned = df[c(1,2,4,6,9,12, 15)]



#changing to lbs, fixing names
cleaned$car_weight_tons = cleaned$car_weight_tons * 907.185
names(cleaned)[7] <- 'weight_kgs'
names(cleaned)[1] <- 'Car'
names(df)[14] <- 'Make'



#no longer necessary to use this code:
#df$year <- as.factor(df$year)
#df2 <- do.call(rbind, lapply(split(df,df$year), function(x) {return(x[which.max(x$top_speed_mp
h),])}))

df2 <- read.csv("FastestofEachYear.csv")
names(df2)[7] <- 'Make'

ggplot(df2,aes(year, top_speed_mph, color = Make)) + geom_point(size = 2) +
  ggtitle("Fastest Car of each Year") + labs(x = "Year", y = "Top Speed,\n (mph)")
```

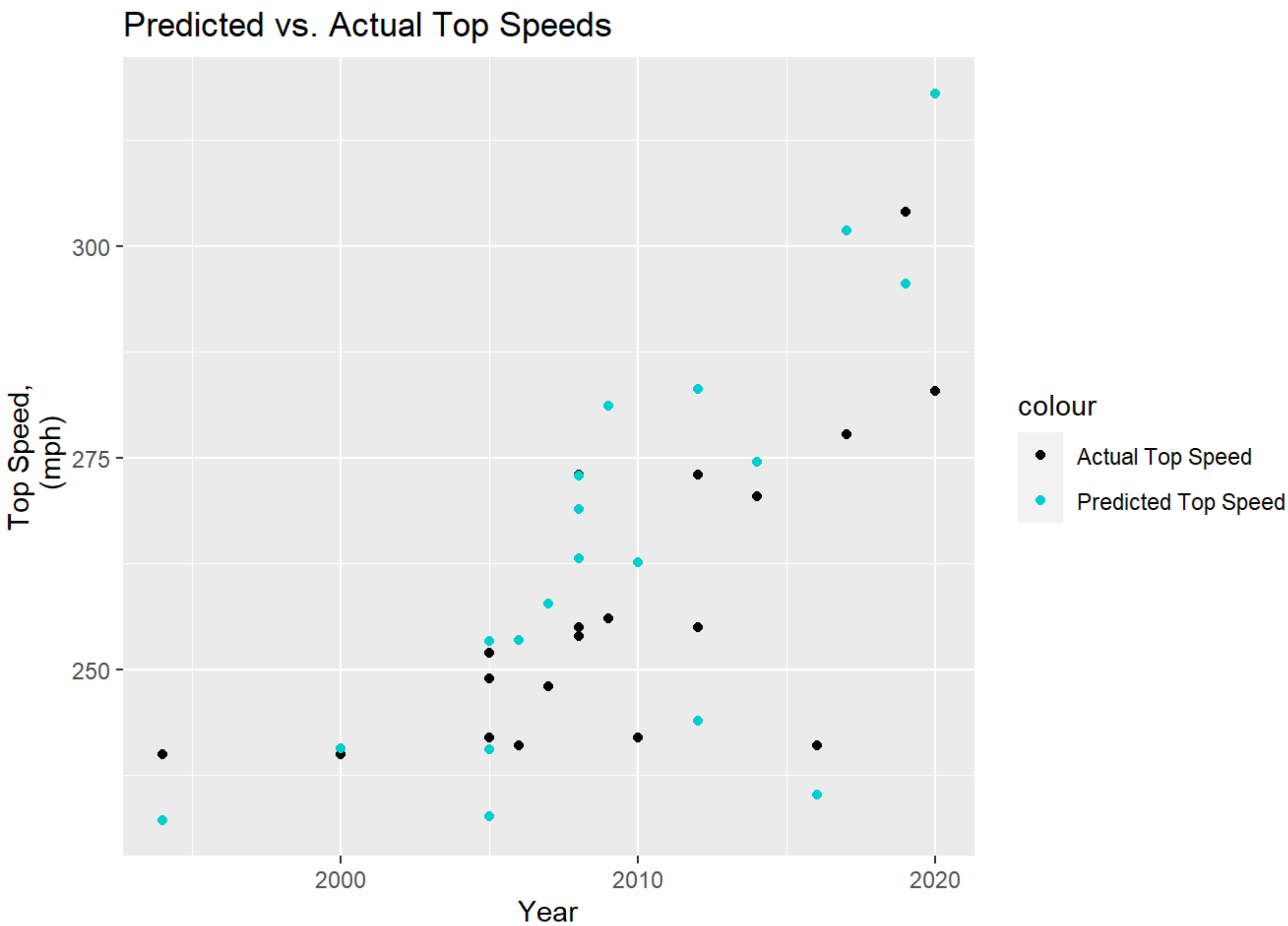


```
#created new csv with cars above 240mph. also added some important cars that the set
#did not include since it is a bit outdated
#further filtering the fastest cars only to necessary specs
#fast_temp.ind <- which(cleaned["top_speed_mph"] >= 240)
#fast_temp <- cleaned[fast_temp.ind,]

#added parameters that the data set did not have for each car (ex: frontal area, drag coef)
#write.csv(fast_temp, "C:\\Users\\anton\\Desktop\\ModFastData.csv", row.names = FALSE)

fast <- read.csv("FinalFastMod.csv")
names(fast)[13] <- 'Make'

#Graph of the final cars that I will use (above 240 mph)
ggplot(fast,aes(year, top_speed_mph, color = Make)) + geom_point(size = 2) +
  ggtitle("Top Speed vs. Year") + labs(x = "Year", y = "Top Speed,\n (mph)")
```



Building difeq model and seeing how well it predicts:

```
#Difeq model to predict top speed of our cars
for(i in 1:19){
  parameters <- c(H = fast[i,2], n = fast[i,8], m = fast[i,7],
                  c = fast[i,9], A = fast[i,10], p = 1.202, r = 0.00923, g = 9.8)

  state <- c(v = 0.000001)

  Velocity <- function(t, state, parameters){
    with(as.list(c(state, parameters)), {
      dv <- (1/(m + 78)) * ((H * 745.7 * n)/v - (0.5 * p * c * A *(v^2)) - (r * m * g))

      list(c(dv))
    })
  }

  times <- seq (0, 60, by = 1)

  out <- ode(y = state, times = times, func = Velocity, parms = parameters)
  head(out)

  fast[i, 11] = max(out[,])*2.23694

}

#plotting our model results vs actual top speeds
p <- ggplot() +
  geom_point(data=fast, aes(x=year, y=top_speed_mph, color = "Actual Top Speed")) +
  geom_point(data=fast, aes(x=year, y=topspd_mph_predict, color = "Predicted Top Speed")) +
  scale_color_manual(values = c("black", "cyan3")) +
  ggtitle("Predicted vs. Actual Top Speeds") + labs(x = "Year", y = "Top Speed,\n (mph)")
print(p)
```



Top speed run of our hypercar, which takes the best parameters from the other, existing vehicles within reason.

```
#means actual and predicted (in order)
mean(fast[,5])
```

```
## [1] 257.6895
```

```
mean(fast[,11])
```

```
## [1] 263.7705
```

```
#checking the statistical significance in the differences between model and actual
for(i in 1:19){

  fast[i, 12] = (((fast[i, 5] - fast[i, 11])^2)/fast[i, 11])
}

chisqstat <- sum(fast[ ,12])
show(chisqstat)
```

```
## [1] 17.02629
```

```
pchisq(q = chisqstat, df = 18, lower.tail = FALSE)
```

```
## [1] 0.5212984
```

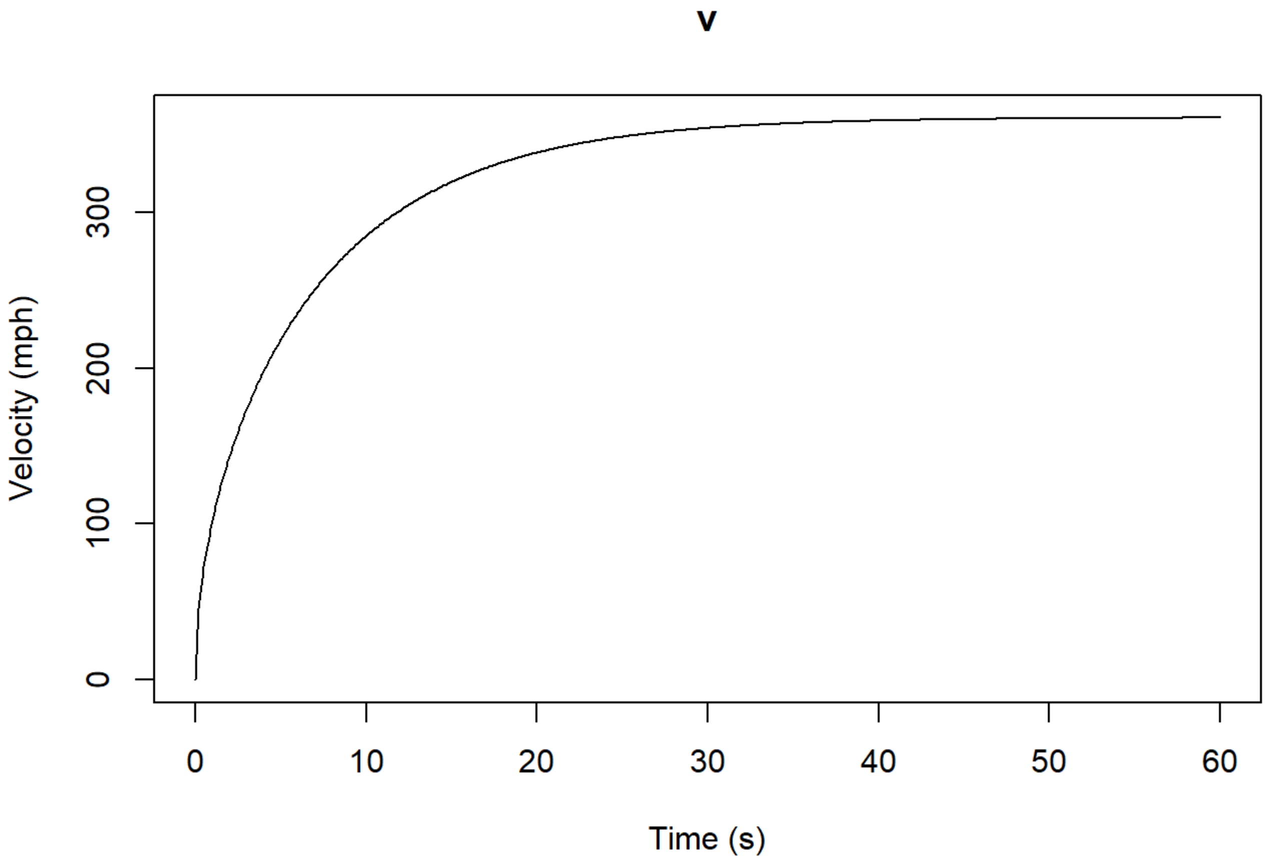
```
#Taking the reasonable best parameter from each car to calculate top speed of ultimate car
mlist <- fast[,7][order(fast[,7])]

mlist[3]
```

```
## [1] 1070.558
```

```
parameters1 <- c(H = max(fast[,2]), n = max(fast[,8]), m = mlist[3],
                 c = min(fast[,9]), A = (min(fast[,10])), p = 1.202, r = 0.00923, g = 9.8)

state1 <- c(v = 0.000001)

Velocity1 <- function(t, state1, parameters1){
  with(as.list(c(state1, parameters1)), {
    dv <- (1/(m + 78)) * ((H * 745.7 * n)/v - (0.5 * p * c * A *(v^2)) - (r * m * g))
    list(c(dv))
  })
}

times <- seq (0, 60, by = 0.1)

out1 <- ode(y = state1, times = times, func = Velocity1, parms = parameters1)
head(out1)
```

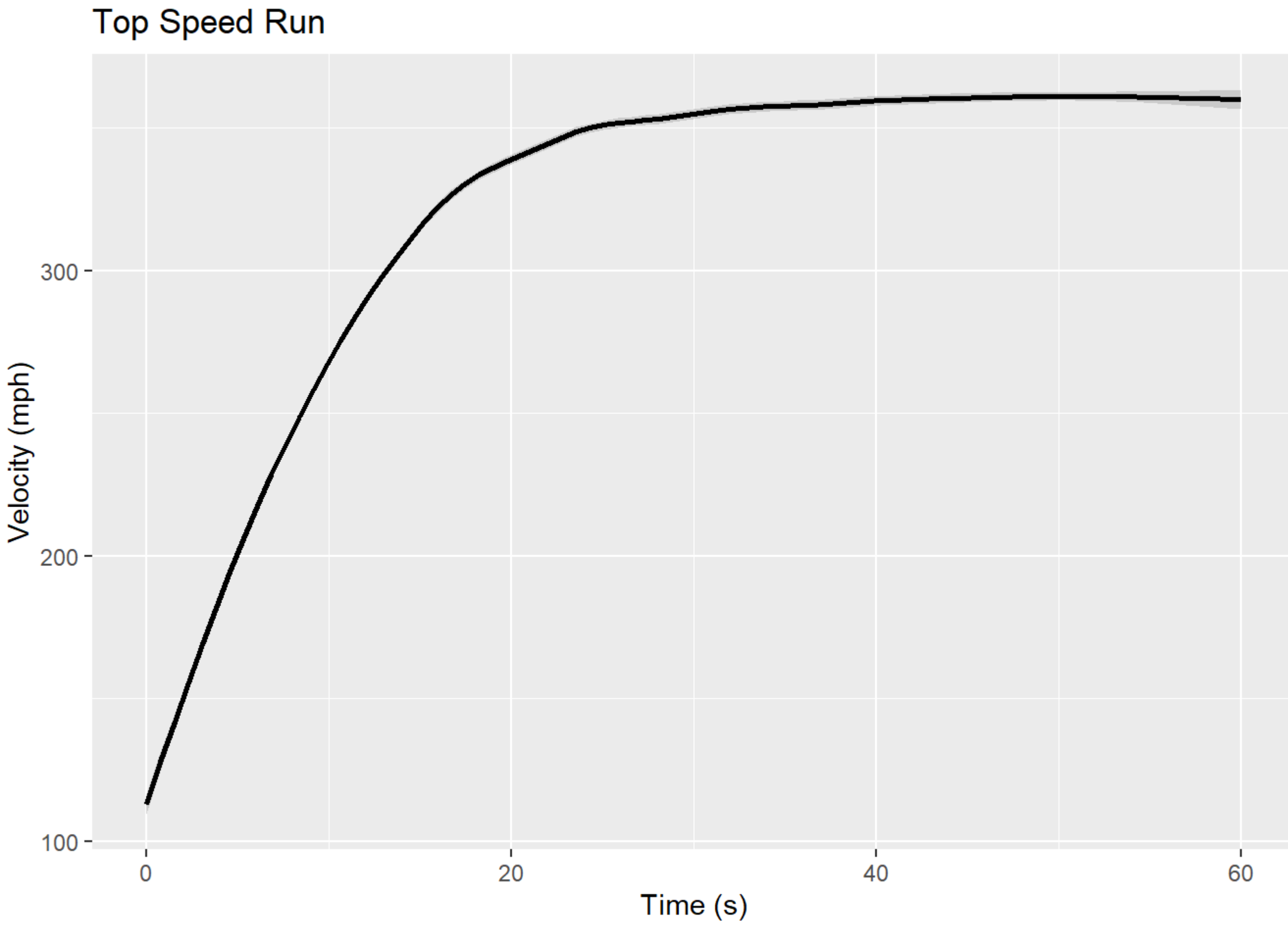
```
##      time         v
## [1,]  0.0  0.000001
## [2,]  0.1 14.529530
## [3,]  0.2 20.539150
## [4,]  0.3 25.143546
## [5,]  0.4 29.018543
## [6,]  0.5 32.425922
```

```
out1[,2] <- out1[,2] * 2.23694

ourcar <- data.frame(time = out1[,1], velocity = out1[,2])
#Plot of our top speed run over time
plot(out1, xlab = "Time (s)", ylab = "Velocity (mph)")
```




```
ggplot() + stat_smooth(data = ourcar, aes(time, velocity), color = "black") +
  ggtitle("Top Speed Run") + labs(x = "Time (s)", y = "Velocity (mph)")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Top Speed Calculations:

```
#Calculation of our top speed
topspeed = max(out1[,])
topspeed
```

```
## [1] 360.7991
```

```
H = max(fast[,2])
n = max(fast[,8])
m = mlist[3]
c = min(fast[,9])
A = (min(fast[,10]))
p = 1.202
r = 0.00923
g = 9.8


#aerodynamics done by hand and checked on Wolfram Alpha
#resulting values are when mph = 361 (161.4 m/s) and 90% power (not peak power at peak speed):

#required front half downforce area
Afront <- 3.701 + 0.091
#required rear half downforce area
Aback <-  4.0104 + 0.091
#Flat top surface area of the car (length * width)
SAcar <- 7.80234
#Maximum added frontal area (inefficient aero design)
Max_added_A <- (Afront + Aback - (SAcar * 0.5)) * sin(12 * pi/180)
#Minimum added frontal area (very efficient aero design)
Min_added_A <- (Afront + Aback - SAcar) * sin(12 * pi/180)

#Running the above differential equation and adding the max and min added frontal areas
#results in these top speeds. (this means adding to the 'A' parameter in parameters1)

#top speed with maximum inefficiency(car body only provides half of the necessary downforce)
topspeedbad <- 319.6126
topspeedgood <- 359.5519
topspeedunsafe <- topspeed
```

Logistic Regression. Now we predict the year in which these different top speeds that we calculated above will be achieved.

```
#S curve regression model (logistic regression)
supercars.ind <- which(df2["year"] >= 1957)
supercars <- df2[supercars.ind,]

supercars$year <- supercars$year - 1957
supercars$top_speed_mph <- supercars$top_speed_mph - 132.941

#Predicting year based on top speed (logistic)
predicto <- function(k, n, r, s){
  time <- (-1/r) * log((n * (k-s))/(s * (k-n)))
  return(time)
}

scurvepredict1 <- predicto(417.965,22.059,0.038,topspeedunsafe - 132.941)
scurvepredict2 <- predicto(417.965,22.059,0.038,topspeedgood - 132.941)
scurvepredict3 <- predicto(417.965,22.059,0.038,topspeedbad - 132.941)

supercars[63,1] = "Bad"
supercars[63,5] = topspeedbad - 132.941
supercars[63,6] = scurvepredict3

supercars[64,1] = "Good"
supercars[64,5] = topspeedgood - 132.941
supercars[64,6] = scurvepredict2

supercars[65,1] = "Unsafe"
supercars[65,5] = topspeedunsafe - 132.941
supercars[65,6] = scurvepredict1

#Plotting our model and predicted points
p <- ggplot(data = supercars, aes(year, top_speed_mph)) + geom_point() +
  ggtitle("Top Speed Logistic Prediction Model") + labs(x = "Years since 1957", y = "Top Speed -
133,\n (mph)")
model.wt <- SummarizeGrowth(supercars$year, supercars$top_speed_mph)
model.wt
```

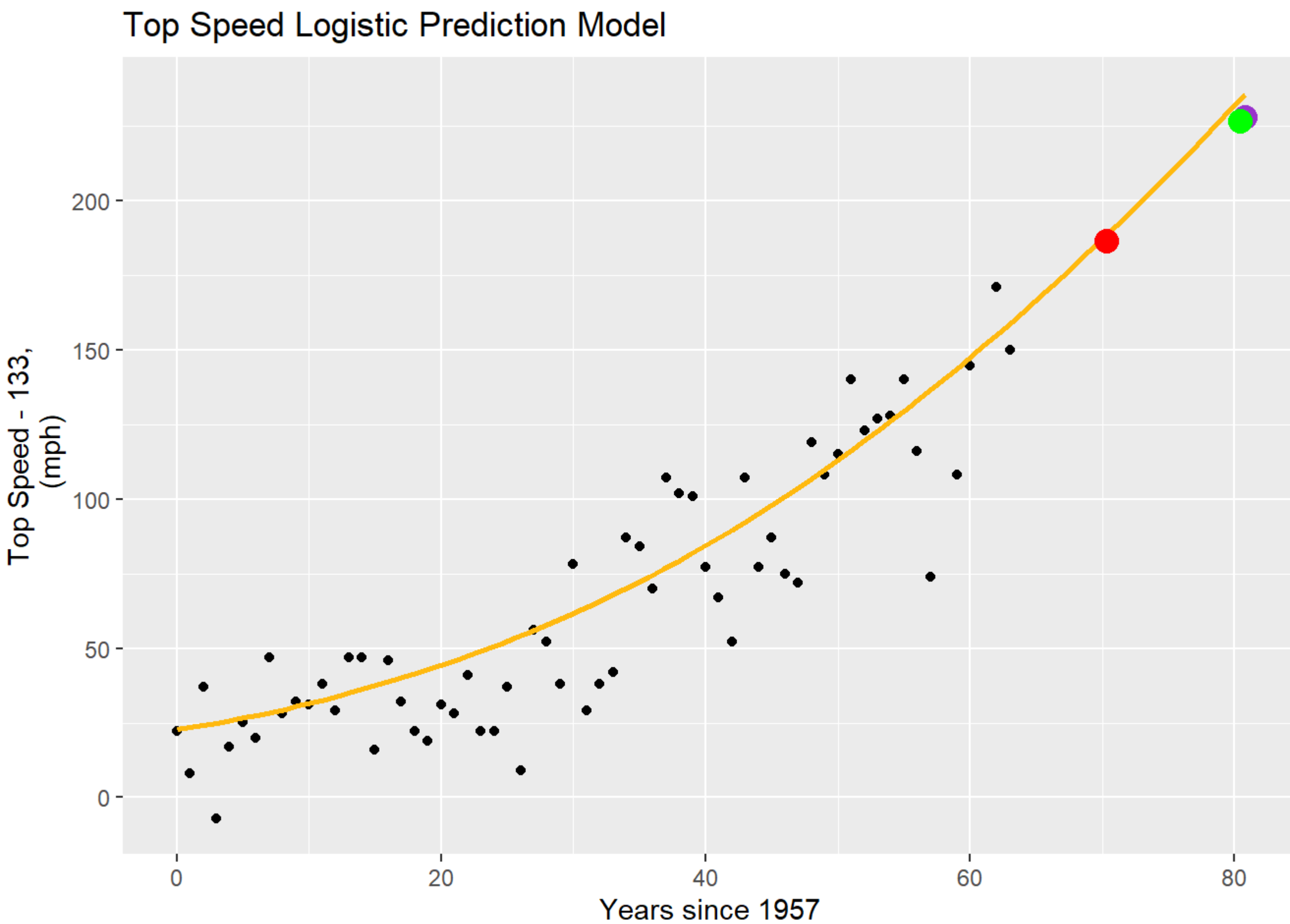
```
## Fit data to K / (1 + ((K - N0) / N0) * exp(-r * t)):
##      K   N0  r
##   val:   486.101 22.392  0.037
##   Residual standard error: 18.734 on 62 degrees of freedom
##
## Other useful metrics:
##   DT 1 / DT  auc_l   auc_e
##   18.87  5.3e-02 8128.17 8128.26
```

```
#Summary of model values
str(model.wt$vals)
```

```
## List of 16
##  $ k    : num 486
##  $ k_se : num 188
##  $ k_p  : num 0.012
##  $ n0   : num 22.4
##  $ n0_se: num 3.23
##  $ n0_p : num 2.86e-09
##  $ r    : num 0.0367
##  $ r_se : num 0.00509
##  $ r_p  : num 9.29e-10
##  $ sigma: num 18.7
##  $ df   : num 62
##  $ t_mid: num 82.5
##  $ t_gen: num 18.9
##  $ auc_l: num 8128
##  $ auc_e: num 8128
##  $ note : chr ""
##  - attr(*, "class")= chr "gcvals"
```

```
#Standard residual error
sigma <- 18.7
#Theoretical top speed that supercars will actually ever achieve before we transition to
#a new technology
Asymptote <- 486

predictedval <- data.frame(time = supercars[,6], pred.wt = predict(model.wt$model))

#Final plot with points, s curve, and predicted points
finalp <- p + stat_smooth(data = predictedval, aes(time, pred.wt), color="darkgoldenrod1") +
  geom_point(aes(x = scurvepredict1, y = topspeedunsafe- 132.941), color = "darkorchid", size =
4) +
  geom_point(aes(x = scurvepredict2, y = topspeedgood- 132.941), color = "green", size = 4) +
  geom_point(aes(x = scurvepredict3, y = topspeedbad- 132.941), color = "red", size = 4)
print(finalp)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Linear Prediction with prediction bands. Same idea, predicting which years we will achieve these top speeds in.

```r
#Linear Regression model
supercarz.ind <- which(df2["year"] >= 1939)
supercarz <- df2[supercarz.ind, ]

#Function that predicts year based on top speed (linear)
predicta <- function(y,a,b){
  x <- (y-a)/b
  return(x)
}

linearpredict1 <- predicta(topspeedunsafe, -3672, 1.947)
linearpredict2 <- predicta(topspeedgood, -3672, 1.947)
linearpredict3 <- predicta(topspeedbad, -3672, 1.947)

supercarz[63,1] = "Bad"
supercarz[63,5] = topspeedbad
supercarz[63,6] = linearpredict3

supercarz[64,1] = "Good"
supercarz[64,5] = topspeedgood
supercarz[64,6] = linearpredict2

supercarz[65,1] = "Unsafe"
supercarz[65,5] = topspeedunsafe
supercarz[65,6] = linearpredict1

linearfit <- lm(supercarz$top_speed_mph ~supercarz$year)
#Summary of model values
summary(linearfit)
```

```
##
## Call:
## lm(formula = supercarz$top_speed_mph ~ supercarz$year)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -46.88 -16.97   2.92  15.74  46.42
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.596e+03  1.736e+02  -20.72   <2e-16 ***
## supercarz$year  1.908e+00  8.746e-02   21.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.81 on 71 degrees of freedom
## Multiple R-squared:  0.8702, Adjusted R-squared:  0.8684
## F-statistic: 476.1 on 1 and 71 DF,  p-value: < 2.2e-16
```

```
corrCoef <- 0.8702

#Final plot with points, line, and predicted points
p1 <- ggplot(data = supercarz, aes(year, top_speed_mph)) + geom_point() +
  ggtitle("Top Speed Linear Prediction Model") + labs(x = "Year", y = "Top Speed,\n (mph)") +
  theme(legend.position = "none") +
  stat_smooth(method = "lm", col = "darkgoldenrod1", fill = "darkgoldenrod1", level = 0.95) +
  geom_point(aes(x = linearpredict1, y = topspeedunsafe), color = "darkorchid", size = 4) +
  geom_point(aes(x = linearpredict2, y = topspeedgood), color = "green", size = 4) +
  geom_point(aes(x = linearpredict3, y = topspeedbad), color = "red", size = 4)
print(p1)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```