Property Crime and Felony Investigation

Sivan Yair

10/4/2020

Overview

In this document, we first obtain a plausible range of for the number of crimes that would be misdemeanors in the present but would become felonies with Prop 20. We then infer the crime rate below $950 and the average cost of crimes below $950.

New Prison Inmates

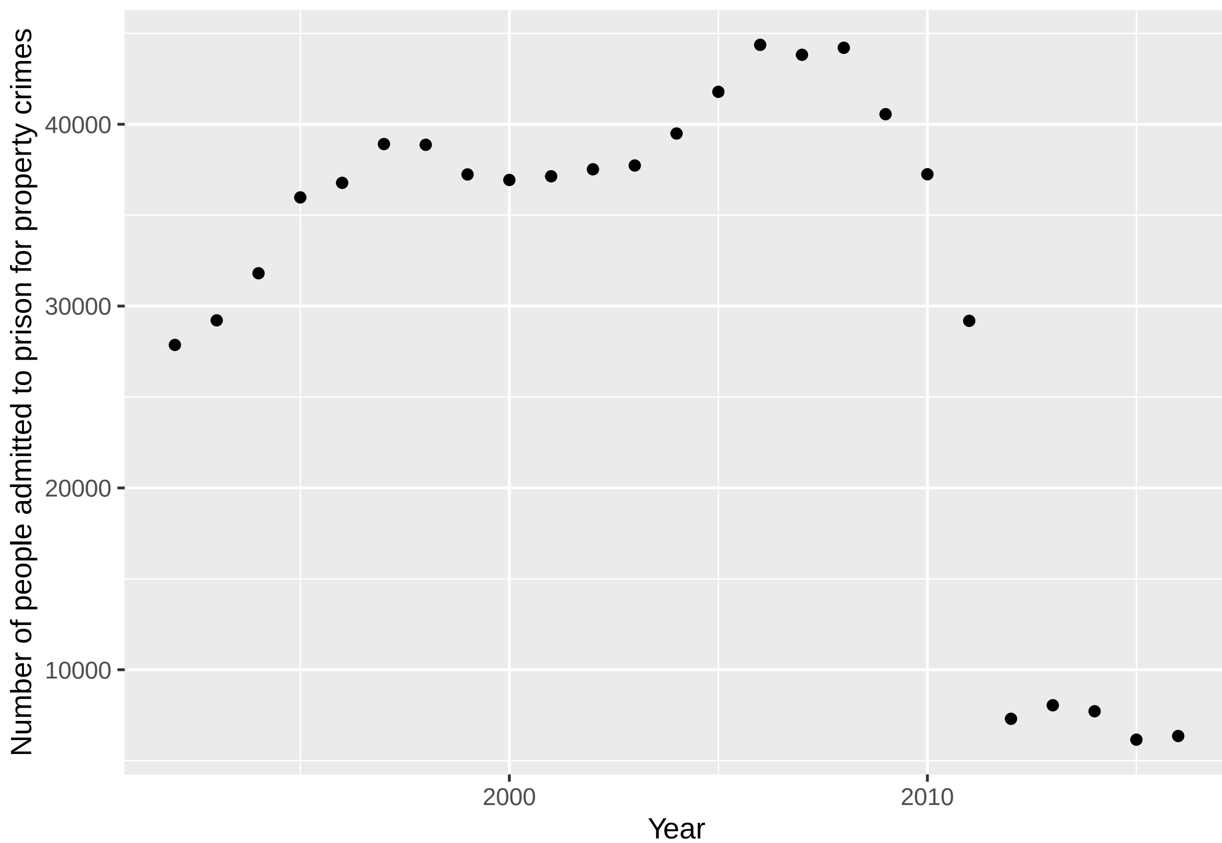
In this section, we identify a 95% confidence interval for the change in the number of felons introduced to prison populations when petty theft criminals are allowed to go to prison. We do this by comparing the number of felons in prison for property crimes from 2011-2014 and 2015-2016. The former range corresponds to the years in which prison sentences were permitted, whereas in the latter range they were not permitted. We do not use data before 2011 because AB109 caused a large decrease in prison populations for reasons outside of the purview of our research questions. We do not have data from after 2016.

Let's start by looking at the trend of property felonies over time.

```
# loads table called propertycrimes, produced by script property_felons_in_prison.R
# we cannot share this because you need to create an account to view the data (see README.md)
# so we just read it in locally
load("~/Downloads/propertycrimes.rda")

felons = data.frame(propertycrimes)
colnames(felons)=c("Year","felons_sum")
felons$Year=as.integer(as.character(felons$Year))
# filter just to 1992 and after because data before that is incomplete
felons = felons %>% filter(Year>=1992)

ggplot(felons,aes(x=Year,y=felons_sum)) + geom_point() + labs(x="Year",y="Number of people admitted to p
```



Now we can test for a significant difference in mean number of felons admitted between 2011-2014 and 2015-2016

```
# new column: prop20 is now a T/F variable. Years 2011-2014 represent prop 20 conditions
felons = felons %>% filter(Year>2011) %>% mutate(prop20=Year<=2014)

# paired t test shows there's a significant difference
t.test(felons$felons_sum~felons$prop20,conf.level=0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  felons$felons_sum by felons$prop20
## t = -6.0381, df = 2.7046, p-value = 0.01218
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2237.1592  -628.8408
## sample estimates:
## mean in group FALSE  mean in group TRUE
##                6254                7687
```

We will save the 95 percent confidence interval representing the plausible range of the number of misdemeanors that would become felonies with Prop 20. We need to reverse the output of the confidence interval (make it positive instead of negative) because the output represents change from having Prop 20 conditions to not having Prop 20 conditions, whereas we want to know about change in the opposite direction.

```r
output_ttest=t.test(felons$felons_sum~felons$prop20,conf.level=0.95)
# reverse output of confidence interval
conf_int_wobblers_entering_prison = c(-output_ttest$conf.int[2],
                      -output_ttest$conf.int[1])
saveRDS(conf_int_wobblers_entering_prison,"../data/wobbler_range.RDS")
```

Property Crime Rates and Average Cost Below $950

We start by loading in the crime data and taking a quick look.

```r
crimes=read.csv("../data/Crimes_and_Clearances_with_Arson-1985-2019.csv")
head(crimes)
```

```
##   Year         County                        NCICCode Violent_sum Homicide_sum
## 1 1985 Alameda County Alameda Co. Sheriff's Department         427            3
## 2 1985 Alameda County                         Alameda         405            7
## 3 1985 Alameda County                          Albany         101            1
## 4 1985 Alameda County                        Berkeley        1164           11
## 5 1985 Alameda County                      Emeryville         146            0
## 6 1985 Alameda County                         Fremont         614            3
##   ForRape_sum Robbery_sum AggAssault_sum Property_sum Burglary_sum
## 1          27         166            231         3964         1483
## 2          15         220            163         4486          989
## 3           4          58             38          634          161
## 4          43         660            450        12035         2930
## 5           5          82             59          971          205
## 6          34          86            491         6053         1786
##   VehicleTheft_sum LTtotal_sum ViolentClr_sum HomicideClr_sum ForRapeClr_sum
## 1              353        2128            122               4              6
## 2              260        3237            205               7              8
## 3               55         418             58               1              3
## 4              869        8236            559               4             32
## 5              102         664             19               0              0
## 6              350        3917            390               2             16
##   RobberyClr_sum AggAssaultClr_sum PropertyClr_sum BurglaryClr_sum
## 1             32                80             409             124
## 2             67               123             889              88
## 3             23                31             166              62
## 4            198               325            1954             397
## 5              4                15              36               9
## 6             27               345            1403             424
##   VehicleTheftClr_sum LTtotalClr_sum TotalStructural_sum TotalMobile_sum
## 1                   7            278                  22               6
## 2                  62            739                  23               4
## 3                  16             88                   2               0
## 4                 177           1380                  72              23
## 5                   8             19                   0               1
## 6                  91            888                  37              26
##   TotalOther_sum GrandTotal_sum GrandTotClr_sum RAPact_sum ARAPact_sum
## 1              3             31              11         22           5
## 2              5             32               7          9           6
## 3              0              2               1          2           2
```

```
## 4              5            100              20          31          12
## 5              0              1               0           4           1
## 6             61            124              14          21          13
##   FROBact_sum KROBact_sum OROBact_sum SROBact_sum HROBnao_sum CHROBnao_sum
## 1          77          22           3          64          59           38
## 2          56          23          11         130         136           48
## 3          23           2           2          31          26           15
## 4         242          71          43         304         351          150
## 5          35          10          11          26          56            9
## 6          38           7           3          38          32           21
##   GROBnao_sum CROBnao_sum RROBnao_sum BROBnao_sum MROBnao_sum FASSact_sum
## 1          23          32          11           3           0          25
## 2           2           2          20           2          10          16
## 3           1           2           6           3           5           3
## 4           0           0          47          21          91          47
## 5           2           0          14           0           1           6
## 6           7           8           9           2           7          47
##   KASSact_sum OASSact_sum HASSact_sum FEBURact_Sum UBURact_sum RESDBUR_sum
## 1          27         111          68         1177         306        1129
## 2          30          90          27          747         242         637
## 3           2          10          23           85          76         100
## 4         103         224          76         2040         890        2015
## 5           8           9          36          161          44          89
## 6          38         120         286         1080         706        1147
##   RNBURnao_sum RDBURnao_sum RUBURnao_sum NRESBUR_sum NNBURnao_sum NDBURnao_sum
## 1          206          599          324         354          216           47
## 2          175          195          267         352          119           46
## 3           33           44           23          61           32           21
## 4          597         1418            0         915          224          691
## 5           32           26           31         116           44           14
## 6          292          485          370         639          274          110
##   NUBURnao_sum MVTact_sum TMVTact_sum OMVTact_sum PPLARnao_sum PSLARnao_sum
## 1           91        233          56          64            5           60
## 2          187        187          33          40           31           20
## 3            8         42           4           9           26            4
## 4            0        559          55         255          133          163
## 5           58         85           9           8            5            4
## 6          255        219          71          60           10           14
##   SLLARnao_sum MVLARnao_sum MVPLARnao_sum BILARnao_sum FBLARnao_sum
## 1          289          930           109          205           44
## 2          664          538           673          516          183
## 3           40          147            62           39           46
## 4         1277         3153           508          611         1877
## 5            1          207           153           16           85
## 6          704         1136           446          360          493
##   COMLARnao_sum AOLARnao_sum LT400nao_sum LT200400nao_sum LT50200nao_sum
## 1            11          475          753             437            440
## 2            53          559          540             622            916
## 3            17           37           84              68            128
## 4            18          496          533             636           2793
## 5            24          169          217             122            161
## 6            27          727          937             607           1012
##   LT50nao_sum
## 1         498
```

```
## 2         1159
## 3          138
## 4         4274
## 5          164
## 6         1361
```

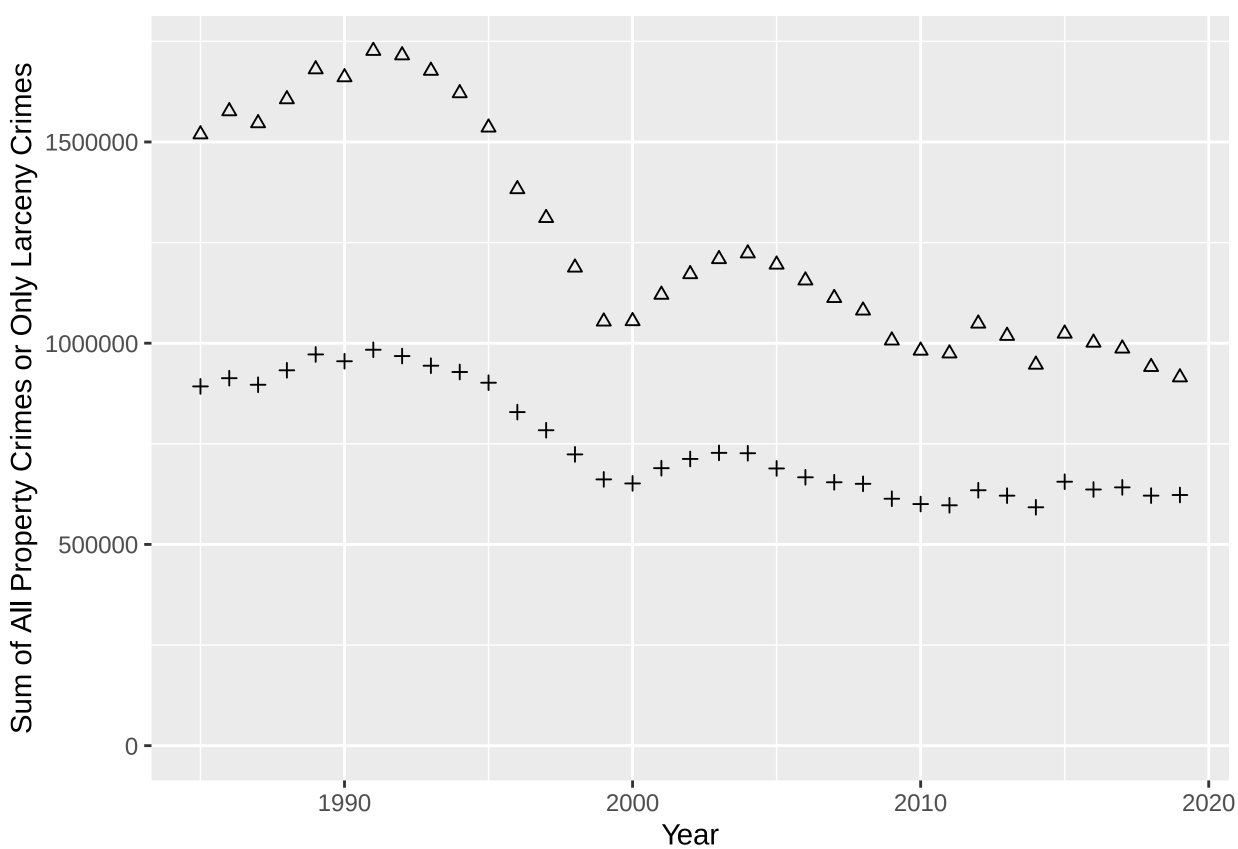
Our git repository contains the pdf describing what all of the column names mean. The ones we are interested in, and their definition, are listed below: - Property__sum: number of property crimes (sum of burglary, motor vehicle, and larceny) - LTtotal__sum: number of larceny-thefts - PropertyClr__sum: number of property crimes cleared aka charged - SLLARnao__sum: number of shoplifting crimes - LT400nao__sum: number of larceny thefts over $400 - LT200400nao__sum: number of larceny thefts from $200-400 - LT50200nao__sum: number of larceny thefts from $50-199 - LT50nao__sum: number of larceny thefts under $50

We remove county-level information by taking the sum of all counts in a category for a given year.

```
crimes_sumYear=crimes %>%
  select(-County,-NCICCode) %>% group_by(Year) %>% summarise_each(funs(sum))
```

Here are the counts of property crimes per year and number of larceny thefts per year (larceny thefts make up a subset of the total property crimes)

```
crimes_sumYear %>%
  ggplot(aes(x=Year))+
  geom_point(aes(y=Property_sum),pch=2)+
  geom_point(aes(y=LTtotal_sum),pch=3) +
  ylim(0,max(crimes_sumYear$Property_sum)) +
  labs(y="Sum of All Property Crimes or Only Larceny Crimes")
```

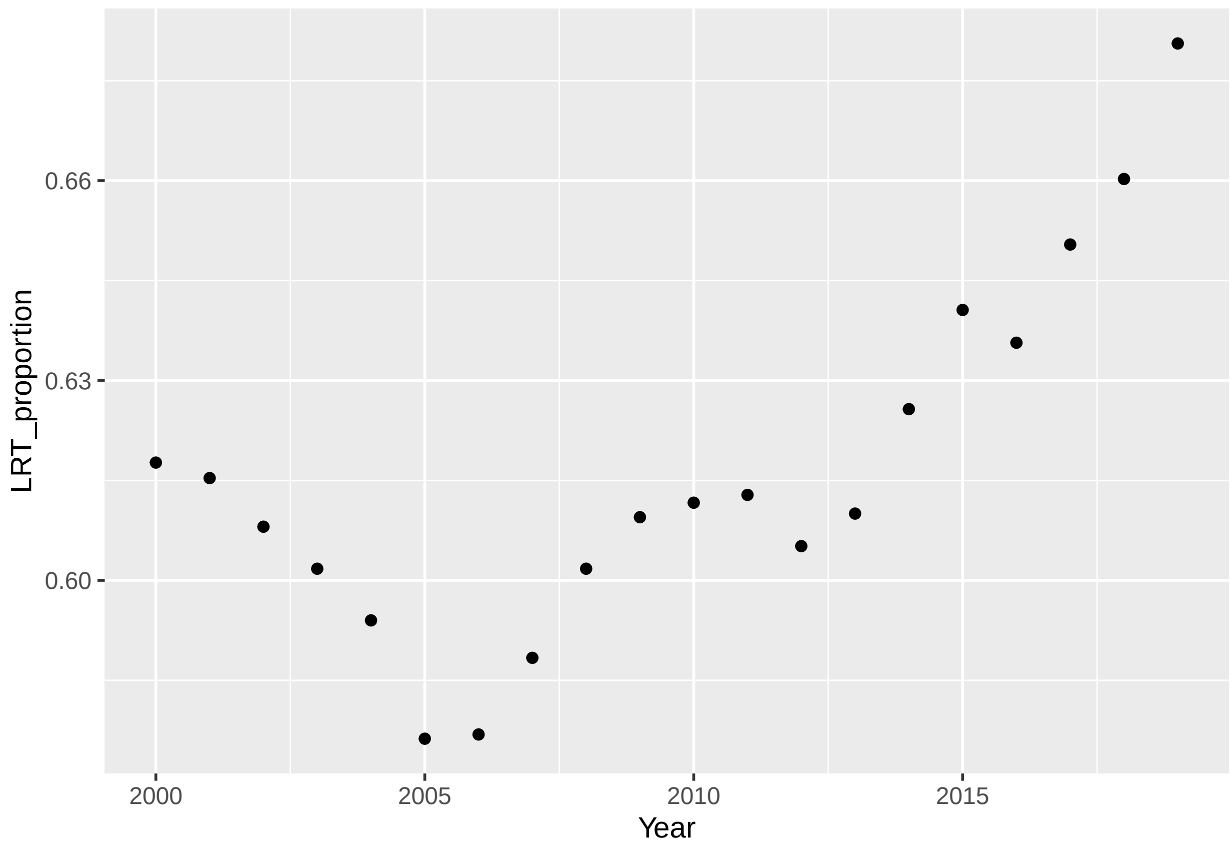


The decline in crime in the 1990s is a phenomenon observed across the United States, and there is no consensus on the cause. We therefore remove all data points before 2000.

```
crimes_sumYear = crimes_sumYear %>% filter(Year>=2000)
```

We need to know the average cost of crimes under $950. We only have data on these values for larceny thefts, but they do make up around 55-70% of property crimes, and seem to be on the rise in recent years. Let's see what the trends are:

```
# proportion of property crimes attributable to larceny
crimes_sumYear %>% mutate(LRT_proportion=LTtotal_sum/Property_sum) %>%
  ggplot(aes(x=Year,y=LRT_proportion))+
  geom_point()
```




```
# average of the above proportion across years
avg_LRT_proportion=crimes_sumYear %>% mutate(LRT_proportion=LTtotal_sum/Property_sum) %>% summarise(avg_
avg_LRT_proportion
```

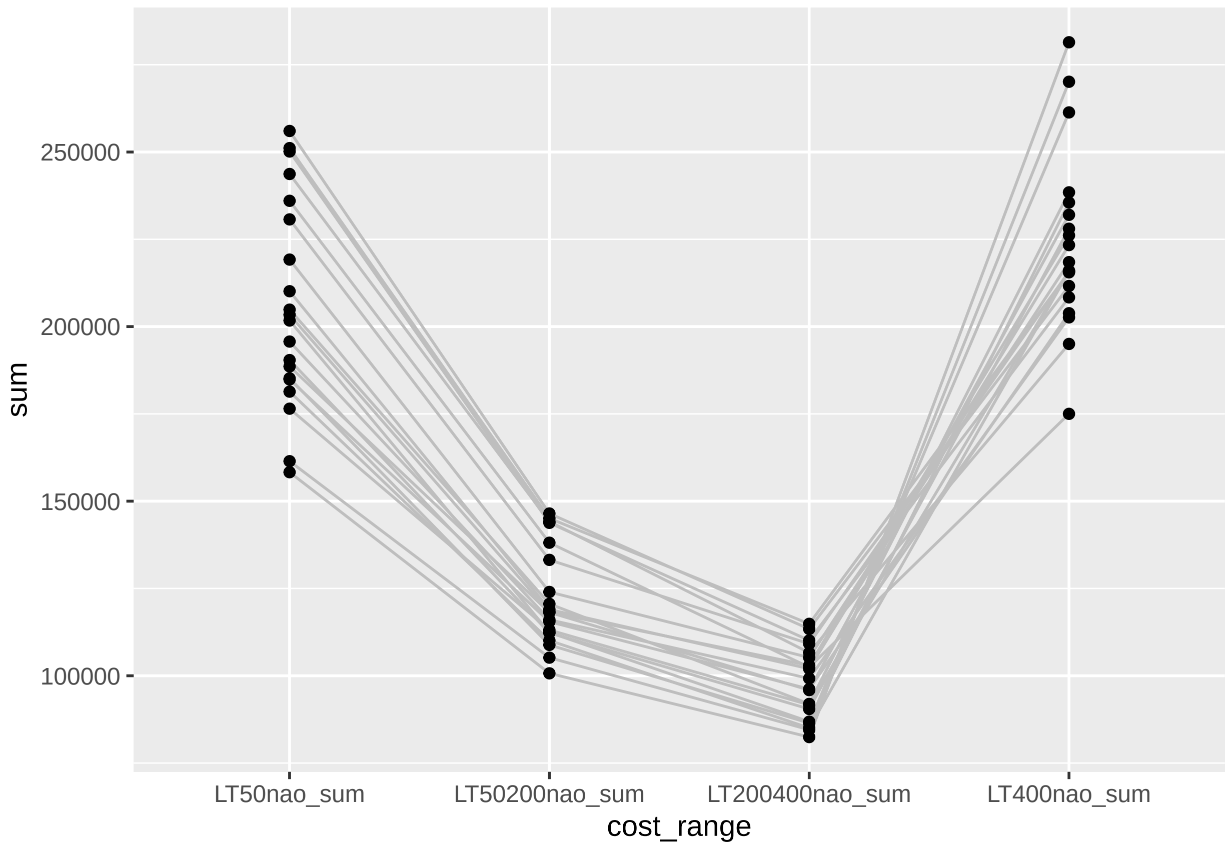
```
## # A tibble: 1 x 1
##   avg_LRT_proportion
##                <dbl>
## 1              0.616
```

Here are the counts of each price range of larceny thefts

```
crime_costs=crimes_sumYear %>% select(Year,LT400nao_sum,LT200400nao_sum,LT50200nao_sum,LT50nao_sum) %>%
  gather(cost_range,sum,LT400nao_sum,LT200400nao_sum,LT50200nao_sum,LT50nao_sum)

# order the categorical data to reflect increasing discrete intervals
crime_costs$cost_range=factor(crime_costs$cost_range, levels=c("LT50nao_sum","LT50200nao_sum","LT200400n

crime_costs %>%
  ggplot(aes(x=cost_range,y=sum,group=Year)) +
  geom_line(col="grey") +
  geom_point()
```



There seems to be a clear decay where most crimes committed are for small values. We are interested in the average cost of crimes below $950, and so we need to build a model that uses the data we do have to continuously predict the number of costs we observe. Since we are working with count data, we use the Poisson to model how the crime count changes with the cost of the crime, among the bounded cost ranges. We spread the count of a crime category across all integer values before proceeding with our models, then parameterize our model, then make predictions for the shiny app.

```
# make a data frame in which the cost of a category is split evenly across its range
cost_range_numeric=list("LT50nao_sum"=1:50,"LT50200nao_sum"=51:200,"LT200400nao_sum"=201:400)

crime_costs_estimation = do.call(rbind,apply(crime_costs[crime_costs$cost_range!="LT400nao_sum",],1,func
  return(data.frame(Year=row["Year"],cost=cost_range_numeric[[row["cost_range"]]],sum=as.integer(row["su

}))
```

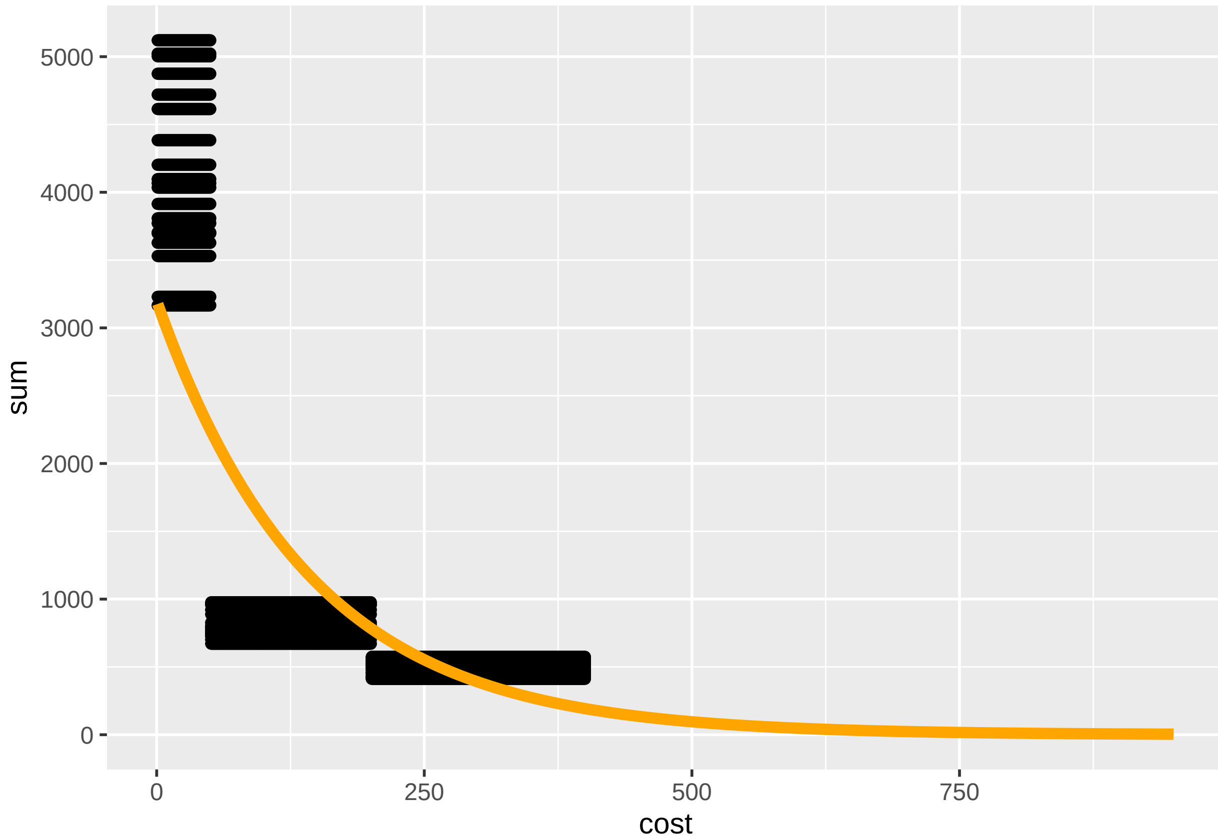
```r
# Poisson model
poisson.model = glm(sum ~ cost, crime_costs_estimation, family = poisson(link = "log"))
```

```r
summary(poisson.model)
```

```
##
## Call:
## glm(formula = sum ~ cost, family = poisson(link = "log"), data = crime_costs_estimation)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -38.909  -10.911   -0.939   11.030   51.729
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  8.071e+00  5.379e-04   15003   <2e-16 ***
## cost        -7.022e-03  3.539e-06   -1984   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 6949092  on 7999  degrees of freedom
## Residual deviance: 2254567  on 7998  degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 4
```

```r
# Show how the Poisson model fits to the data
pred_pois= data.frame(cost=1:950,sum=predict(poisson.model, newdata = data.frame(cost=1:950), type = "re

ggplot() +
  geom_point(data=crime_costs_estimation, aes(x=cost,y=sum,group=Year)) +
  geom_line(data=pred_pois, aes(x=cost,y=sum),col="orange",lwd=2) +
  ylim(0,max(crime_costs_estimation$sum))
```




```
# poisson model parameters
a=unname(coef(poisson.model)[1]) # intercept for log(y)
b=unname(coef(poisson.model)[2]) # slope for log(y)

# VALUES FOR SHINY
# range that we care about
high=950
low=1

# functions
y_count=function(x) exp(a+(b-x)) # number of crimes of value x
y_cost=function(x) x-exp(a+(b-x)) # total cost of all crimes of value x: cost of crime of value x - num

# expected number of crimes in price range provided
total_crime=integrate(f=y_count,lower=low,upper=high)$value
total_crime
```

```
## [1] 1168.613
```

```
# expected total cost of all of these crimes (this accounts for number of crimes of each type)
total_cost=integrate(f=y_cost,lower=low,upper=high)$value
total_cost
```

```
## [1] 450080.9
```

```r
# average cost of each crime
avg_cost=total_cost/total_crime
avg_cost
```

```
## [1] 385.141
```

```r
saveRDS(total_crime/as.numeric(avg_LRT_proportion),"../data/total_crime.RDS") # we inflate the total cr
saveRDS(total_cost,"../data/total_cost.RDS")
saveRDS(avg_cost,"../data/avg_cost.RDS")
```