# Model Data Input

## Training-set

Train data gives policy’s next year premium. Column names speak for themselves.

## Policy Data

Policy data gives contract level information. One policy is composed of one or more contracts. Each contract differs by insurance coverage, e.g. 16G, 16P, & etc. Part of the column explanation can be found below:

* Insured Amount: Insured amounts are comparable only between contracts of one insurance coverage. Insured amounts are named implicitly within each insurance coverage, but differently across coverages.
* Coverage Deductible If Applied: Same as above.
* Replacement Cost of Insured Vehicle: Replacement cost is in the unit of 10,000. This value is used in premium when the cost of recovering equals to replacement, e.g. theft.
* Distribution Channel: Distribution channels can be categorized into two types, direct and indirect. Direct channel means a multiplier of 1/0.7927 in the premium; indirect channel means a multiplier of 1/0.65 in the premium.

## Claim Data

Claim data has not been examined fully yet.

# Data Preprocess

## Method 1: Group by Main Coverage

This method will group Policy Data and Claim Data at policy number level. Independent variable dimensions will be consistent across policies.

On Policy Data, we aggregated contract level information[[1]](#footnote-1) on main coverage groups, and unstacked the information to columns. Steps are as following:

* Policy Data is split into policy level and contract level information.
* On policy level, we can select any row, specifically the first row, for processed independent variable.
* On contract level, we add 'Main\_Insurance\_Coverage\_Group' to index, and aggregate the other 6 columns by:
  + Sum: 'Insured\_Amount1', 'Insured\_Amount2', 'Insured\_Amount3', 'Premium'
  + Count: 'Insurance\_Coverage'
  + Binary on Existence: 'Coverage\_Deductible\_if\_applied'
* On contract level, unstack index 'Main\_Insurance\_Coverage\_Group' into 18 columns. Contract level information is aggregated at policy level now.

On Claim Data, we aggregated claim level information[[2]](#footnote-2) on main coverage groups, and unstacked the information to columns. Steps are as following:

* On claim level, we add 'Main\_Insurance\_Coverage\_Group' to index, and aggregate the other 7 columns by:
  + Sum: 'Paid\_Loss\_Amount', 'paid\_Expenses\_Amount', 'Salvage\_or\_Subrogation?', 'Deductible', 'number\_of\_claimants'
  + Count: 'Claim\_Number'
  + Mean: 'At\_Fault?'
* On claim level, unstack index 'Main\_Insurance\_Coverage\_Group' into 21 columns. Contract level information is aggregated at policy level now.

## Method2: Group by ID

This method aggregates by policy number on dimensions including insurer, vehicle branding, coverage, and claims.

On Insurer level, following features will be included:

* Insured's\_ID: significance of this column indicates the potential of feature engineering on the insured.
* fsex
* fmarriage
* iage\_lab: separate by 25, 30, and 60 above.
* iassured\_lab: separate Legal from natural person.
* iclaims: number of claims from the insured
* iclaim\_paid\_amount: total amount of expenses and loss paid on accidents by insured
* iclaim\_salvage\_amount: total amount retrieved from salvage and subrogation
* ipolicies: number of policies held by the insured
* ipolicy\_coverage\_avg: average number of coverage of each policy
* ipolicy\_premium\_avg: average premium paid by the insured
* ivehicle\_repcost\_avg: average replacement cost.
* (ipolicy\_premium\_chg: average change of premium)

On Vehicle level, following features will be included:

* Coding\_of\_Vehicle\_Branding\_&\_Type: significance of this column indicates the potential of feature engineering on the vehicle.
* vyear\_lab: separate by 0y, 1y, 2y, 3y, and above every 5 year.
* vregion\_lab: separate by domestic and imported.
* Vehicle\_Make\_and\_Model1
* Vehicle\_Make\_and\_Model2
* Engine\_Displacement\_(Cubic\_Centimeter)

On Coverage level, following features will be included:

* cpremium\_dmg: sum of premium on damage contracts
* cpremium\_lia: sum of premium on liability contracts
* cpremium\_thf: sum of premium on theft contracts
* cbucket: bucket ID for coverage written in policy
* cinsured\_rank\_avg: weighted average of insured amount rank within each coverage

On Policy Data level, following features will be included

* aassured\_zip
* iply\_area
* Distribution\_Channel
* Multiple\_Products\_with\_TmNewa\_(Yes\_or\_No?)
* pdmg\_acc\_adj: adjusted pdmg\_acc according to number of claims received by the insurer/policy
* plia\_acc\_adj: adjusted plia\_acc according to number of claims received by the insurer/policy

qptextracts information from coverage-specific features and generates generic features. Generic features will be generated at coverage level, and aggregated at policy level.

On Policy Data, following generic features will be generated:

* “Premium\_Momentum”
* “Effective\_Insured\_Amount”
* “Distribution\_Cost”

# Feature Engineering

* 跟日期有關的，可以轉換到區間裏
  + 汽車的生產日期
  + 投保人出生日期
* 類別類，training set 和 testing set 不匹配
* Insurance Coverage

# Model Specification and Assumption

1. Contract level information includes 'Main\_Insurance\_Coverage\_Group', 'Insurance\_Coverage', 'Insured\_Amount1', 'Insured\_Amount2', 'Insured\_Amount3', 'Coverage\_Deductible\_if\_applied', 'Premium'. [↑](#footnote-ref-1)
2. Claim level information includes 'Main\_Insurance\_Coverage\_Group', 'Paid\_Loss\_Amount', 'paid\_Expenses\_Amount', 'Salvage\_or\_Subrogation?', 'At\_Fault?', 'Deductible', 'number\_of\_claimants', 'Claim\_Number' [↑](#footnote-ref-2)