Most of the following variables are location independent. Those that are will be specified as such.
Highlighted in yellow indicates that we have direct data available, though quality varies from location to location.
Highlighted in gray indicates that the variable is calculated for display purposes, but is not actually used for simulation purposes.

#==========
1 - Policies & Actions
#==========

ClosureP	'Closure Policy', units = 'unitless', func = Varies from 0 to 1 representing effectiveness, keep value unless changed manually or by conditional rule [Note: The possible values for this depends on the policies used by the specific location] For Rio de Janeiro: {'No Closures' : 1, 'Fase 6' : 0.9, 'Fase 5': 0.8, 'Fase 4' : 0.7, 'Fase 3B': 0.6, 'Fase 3A': 0.5, 'Fase 2': 0.4, 'Fase 1': 0.3, 'Lockdown': 0.2} For Chile and Santiago: {'Paso 5': 1, 'Paso 4' : 0.7, 'Paso 3': 0.5, 'Paso 2': 0.4, 'Paso 1': 0.2}
NewOVents	New Ventilator Orders', units = 'ventilator', func = Zero, unless changed manually or by conditional rule maxval = 1000000 minval = 0
SocialDisP	'Social Distancing Policy', units = 'unitless' func = Varies from 0 to 1 representing effectiveness, keep value unless changed manually or by conditional rule [Note: This is primarily a placeholder variable for locations that we don't more defined policies for] {'No Distancing' : 1, 'Voluntary Social Distancing' : 0.6, 'Mandatory Social Distancing' : 0.1} For Chile and Santiago: {'No Curfew' : 1, 'Unenforced Curfew' : 0.6, 'Enforced Curfew' : 0.1}

#==========
2 - Health Parameters
#==========

BaseContactR	'Base Contact Rate', units = 'people/(day*person)', func = Constant, currently set at 5
ContactR	'Contact Rate', units = 'people/(day*person)', func = ClosureP * SocialDisP * BaseContactR,
Infectivity	'Infectivity', units = 'likelihood of infection per contact', func = Constant, currently set at 0.05 maxval = 1, minval = 0
HosL	'Hospitalization Likelihood, given infection', units = 'probability', func = Constant, currently set at 0.39 maxval = 1, minval = 0
UHML	'Unhospitalized Mortality Likelihood', units = 'probability', init_value = 0.3, func = Constant, currently set at 0.3 maxval = 1, minval = 0
UHRL	'Unhospitalized Recovery Likelihood', units = 'probability' func = 1 - UHML maxval = 1, minval = 0
bHRL	'Base Hospitalized Recovery Likelihood', units = 'probability', func = Constant, currently set at 0.9, maxval = 1, minval = 0
HRL	Hospitalized Recovery Likelihood', units = 'probability', func = If HPop > 5 * Vents, 0.7 * bHRL; If HPop > 5 * Vents & HPop > HBeds, 0.4 * bHRL; else bHRL maxval = 1, minval = 0

HML	'Hospitalized Mortality Likelihood', units = 'probability' func = 1 - HRL.value maxval = 1, minval = 0
RecL	'Recovery Likelihood', units = 'probability' func = (1-HosL) * UHRL + HosL * HRL, maxval = 1, minval = 0
MorL	'Mortality Likelihood', units = 'probability' func = (1-HosL.value) * UHML + HosL * HML, maxval = 1, minval = 0
AvHDur	'Average Hospitalization Duration', units = 'days' func = Constant, currently set at 7 maxval = 300, minval = 0
AvDur	'Average Illness Duration', units = 'days' func = Constant, currently set at 14 maxval = 300, minval = 0

#===
3 - Health Populations
#===

SPop	'Susceptible Population', units = 'people', func = SPop - InfectR * timestep, maxval = 1000000000, minval = 0
IPop	"'True' Unhospitalized Infected Population", units = 'people' func = IPop + (InfectR - UHMR - HosR - UHRR) * timestep, maxval = 100000000, minval = 0 [Note: The historical data for this is based on statistical estimates of the infected population conducted by epidemiologists]
Deaths	'Deaths', units = 'people' func = Deaths + (UHMR + HMR) * timestep, maxval = 100000000, minval = 0
HPop	'Hospitalized Population', units = 'people' func = HPop + (HosR - HMR - HRR) * timestep, maxval = 1000000, minval = 0
RPop	'Known Recovered Population', units = 'people', func = RPop + (UHRR + HRR) * timestep, maxval = 100000000, minval = 0
mIPop	"Measured Unhospitalized Infected Population", units = 'people', func = true_to_measured(IPop, 14, 0.25), maxval = 100000000, minval = 0 [Note: true_to_measured refers to a functions that samples from the left half of a normal curve with a mean equal to IPop of 14 days ago and a standard deviation of 0,25*IPop of 14 days ago. This is intended to simulate both the delay in measuring the infected population and that the measurement is an undercount of the actual population.]
mTotIPop	'Measured Total Infected Population', units = 'people', func = mIPop + HPop, maxval = 100000000, minval = 0

TotIPop	"'True' Total Infected Population", units = 'people', func = IPop + HPop, maxval = 100000000, minval = 0

#===
4 - Health Flows
#===

InfectR	"'True' Infection Rate", units = 'people/day', func = (combos(SPop + IPop) - combos(SPop) - combos(IPop)) / combos(SPop + IPop) * ContactR * (SPop + IPop) * Infectivity, maxval = SPop, minval = 0 [Note: The historical data for this is based on statistical estimates of the infected population conducted by epidemiologists] [Note The above update function is essentially 'ContactR * Infectivity * (IPop + SPop) * likelihood that if two randomly sampled individuals meet, one is infected and the other isn't]
mInfectR	"Measured Infection Rate", units = 'people/day', func = true_to_measured(InfectR, 14, 0.25), maxval = SPop, minval = 0 [Note: true_to_measured refers to a functions that samples from the left half of a normal curve with a mean equal to InfectR of 14 days ago and a standard deviation of 0,25*InfectR of 14 days ago. This is intended to simulate both the delay in measuring the infected population and that the measurement is an undercount of the actual population.]
UHRR	Unhospitalized Recovery Rate', units = 'people/day' func = (1 - HosL) * UHRL * IPop / AvDur, maxval = IPop, minval = 0
UHMR	'Unhospitalized Mortality Rate', units = 'people/day', func = (1 - HosL) * UHML * IPop / AvDur, maxval = IPop, minval = 0
HRR	'Hospital Recovery Rate', units = 'people/day', func = HRL * HPop / AvHDur, maxval = HPop, minval = 0
HMR	Hospital Mortaility Rate', units = 'people/day', func = HML * HPop / AvHDur, maxval = HPop, minval = 0

HosR	'Hospitalization Rate', units = 'people/day', func = HosL * IPop / AvDur, maxval = IPop, minval = 0

#===
5 - Equipment Supplies
#===

HBeds	'Hospital Bed Capacity', units = 'person', func = Constant, currently set at 2000, maxval = 1000000, minval = 0 [Note: This data is not available in most locations, thus it's placeholder value]
Vents	'Available Ventilators', units = 'ventilator', func = Vents + VentAqRate * timestep, maxval = 1000000, minval = 0
OVents	'Ordered Ventilators', units = 'ventilator', func = OVents + NewOVents - (VentAqRate * timestep), maxval = 1000000, minval = 0
PCR	'Daily PCR Tests Conducted', units = 'tests', func = PCR, maxval = 1000000, minval = 0, locations = Chile

#===
6 - Equipment Parameters
#===

VWTP	'Ventilator Willingness to Pay', units = 'dollar/ventilator', func = Constant, currently set at 25000, maxval = 1000000, minval = 0
VDur	'Default Ventilator Delivery Duration', units = 'days', func = Constant, currently set at 30, maxval = 365, minval = 0
VentAqRate	'Ventilator Acquisition Rate', units = 'ventilator/day', func = OVents / VDur * (3 * (1 - exp(-log(3/2) / 25000 * VWTP))), maxval = 1000000, minval = 0

#==========
7 - Environment
#==========

[AirPolluntant]	'Daily Air Pollutant Level', units = 'µg / m3', func = Constant, set at most recent historical value, maxval = 1000, minval = 0 [Note: This is currently a placeholder that needs an actual equation.] We currently have varying amounts of data for the following air pollutants: SO2 NO2 HCNM HCT CH4 CO NOx O3 PM10 PM2.5

#===
8 - Economic
#===

CityEmployment	'Target City Unemployment Rate', units = 'percent', func = CityEmployment + CityEmploymentR * timestep, maxval = 1, minval = 0

CityEmploymentR	'Target City Unemployment Rate of Change',
	units = 'percent',
	func = RioEmploymentR_update(timestep, tind),
	maxval = 0.05,
	minval = -0.05
	[Note below, is the current placeholder function, which is based on the closure policy and total infected population. When nothing is closed and nobody is infected, it trends back to its initial value.]
	if RioEmployment. < RioEmployment[initial]:
	base_EmpR = normal distribution sample(mean = -0.005, std = 0.0025)
	else:
	base_EmpR = normal distribution sample(mean = 0, std = 0.0025)
	EmpR = base_EmpR
	ClosureVal = ClosureP
	
	if 0.7 < ClosureVal < 1:
	EmpR = normal distribution sample(mean = 0.0005, std = 0.0003
	elif 0.3 < ClosureVal <= 0.7:
	EmpR = normal distribution sample(mean = 0.001, std = 0.005)
	elif ClosureVal <= 0.3:
	EmpR = normal distribution sample(mean = 0.0015, std = 0.0055)

NatEmployment	'National Unemployment Rate', units = 'percent', func = NatEmployment + NatEmploymentR * timestep, maxval = 1, minval = 0
NatEmploymentR	'National Unemployment Rate of Change', units = 'percent', func = NatEmploymentR_update(timestep, tind), maxval = 0.05, minval = -0.05 [Note: See CityEmploymentR for an example of the type of placeholder update function currently in use]

GDP	'Gross Domestic Product', units = 'Million Currency', func = Constant, currently set at most recent historical value, maxval = 10000000, minval = 0 [Note: This is currently a placeholder that needs an actual equation.]

Examples of other data currently not included:

- Public Transportation Usage
- Telecoms-Based Mobility Data
- Air Travel Rates (by flights and by passengers)
- Mobility Index