Annotated Code

Cell 1

```
import pandas as pd
import numpy as np
```

Imports the Python libraries used later in the notebook.

Cell 2

```
# Project and client identification

project_name = "project #2: Automation of NDOT Concrete Mix Design"
client = "Nebraska Department of Transportation (NDOT)"
```

Project and client identification

Cell 3

```
# Print project name and client as part of an f-string
print(f"This Project is Designed for: {client}")
```

Print project name and client as part of an f-string Then it prints the result so you can see it.

Cell 4

```
# Try it out for some values

my_number = 7.321212
print(f'{my_number:8.1f}')
```

Try it out for some values Then it prints the result so you can see it.

Cell 5

```
# Define the fundamental constants used throughout the mix design

cubic_yard_ft3 = 27 #cubic feet in one cubic yard
unit_weight_water = 62.4 #lb/ft^3 (unit weight of water)
```

```
# Print the constants as part of an f-string
print(f"Defined Constant: 1 cubic yard = {cubic_yard_ft3} cubic feet")
print(f"Unit weight of water = {unit_weight_water} lb per cubic feet")
```

Define the fundamental constants used throughout the mix design Then it prints the result so you can see it.

Cell 6

```
# define a function to calculate water weight
def calculate_water_weight_Q(cement_A, Fly_ash_B, silica_fume_C,
other_SCM_D, wc_ratio_E):

    # Our first local variable: total cementitious weight
    total_cementious = cement_A + Fly_ash_B + silica_fume_C + other_SCM_D

    # Our second local variable: water weight
    water_weight_Q = total_cementious * wc_ratio_E

    # this returns water weight
    return water_weight_Q
```

define a function to calculate water weight Inputs: cement_A, Fly_ash_B, silica_fume_C, other_SCM_D, wc_ratio_E. Returns: water_weight_Q.

Cell 7

```
# Method 1: pass input values directly into the function call
w_weight_1 = calculate_water_weight_Q(600, 100, 30, 70, 0.42)
# Print the result
print(f"water weight = {w_weight_1} lb/yd^3")
```

Method 1: pass input values directly into the function call Then it prints the result so you can see it.

Cell 8

```
# Method 2:  store input values in variables first and then supplied them
to the function
cement_A = 600
fly_ash_B = 100
silica_fume_C = 30
other_scm_D = 70
wc_ratio_E = 0.42
# Variables
```

```
# Use variables in the function
w_weight_2 = calculate_water_weight_Q(cement_A, fly_ash_B, silica_fume_C,
other_scm_D, wc_ratio_E)
# Print the result
print(f"water weight = {w_weight_2} lb/yd^3")
```

Method 2: store input values in variables first and then supplied them to the function Then it prints the result so you can see it.

Cell 9

```
# Define a function to calculate the volume of cement (R)
def calculate_volume_cement_R(cement_A, sg_cement_J):
    volume_R = cement_A / (sg_cement_J * unit_weight_water)
    return volume_R

# Example input values for demonstration
cement_A = 600
sg_cement_J = 3.5

# Use them in the function
R = calculate_volume_cement_R(cement_A, sg_cement_J)

# Print
print (f"_volume of cement (R); {R:.3f} ft^3")
```

Define a function to calculate the volume of cement (R) Inputs: cement_A, sg_cement_J. Returns: volume_R.

Cell 10

```
# Define a function to calculate fly ash volume

def calculate_fly_ash_volume (fly_ash_B, sg_fly_ash_K):
       return fly_ash_B / (sg_fly_ash_K * unit_weight_water)
fly_ash_B = 100.0
sg_fly_ash_K = 2.3

S = calculate_fly_ash_volume(fly_ash_B, sg_fly_ash_K)

print(f'Fly ash volume (S): {S:.3f} ft/yd^3')


# Print
```

Define a function to calculate fly ash volume Inputs: fly_ash_B, sg_fly_ash_K. Returns: fly_ash_B / (sg_fly_ash_K * unit_weight_water).

Cell 11

```
# Define a function to calculate silica fume volume
def silica_fume_volume (silica_fume_C, sg_silica_fume_L):
       return silica_fume_C / (sg_fly_ash_K * unit_weight_water)
silica_fume_C = 30.0
sg_silica_fume_L = 2.20
T = silica_fume_volume(silica_fume_C, sg_silica_fume_L)

print(f'Silica fume volume (T): {T:.3f} ft/yd^3')
```

Define a function to calculate silica fume volume Inputs: silica_fume_C, sg_silica_fume_L. Returns: silica_fume_C / (sg_fly_ash_K * unit_weight_water).

Cell 12

```
# Define a function to calculate other SCM volume
def calculate_volume_U (other_scm_D, sg_other_scm_M):
       return other_scm_D / (sg_other_scm_M * unit_weight_water)
other_scm_D = 16
sg_other_scm_M = 2.60
U = calculate_volume_U(other_scm_D, sg_other_scm_M)

print(f'SCM volume (U): {U:.3f} ft/yd^3')
```

Define a function to calculate other SCM volume Inputs: other_scm_D, sg_other_scm_M. Returns: other_scm_D / (sg_other_scm_M * unit_weight_water).

Cell 13

```
# Define a function to calculate other air volume
def calculate_other_air (target_air_F):
    return (target_air_F / 100) * 27
target_air_F = 6

F = calculate_other_air(target_air_F)

print(f'Other Air Volume (F): {F:.3f} ft^3')



# Print
```

Define a function to calculate other air volume Inputs: target_air_F. Returns: (target_air_F / 100) * 27.

Cell 14

```
# Define a function to calculate water volume (W) as a function of water
weight (Q)
def calculate_water_volume (weight_water_Q):
    return (weight_water_Q / 62.4)
weight_water_Q = 6
Q = calculate_water_volume(weight_water_Q)

print(f'Water Volume (Q): {Q:.3f} ft^3')
```

Define a function to calculate water volume (W) as a function of water weight (Q) Inputs: weight_water_Q. Returns: (weight_water_Q / 62.4).

Cell 15

```
# Define a function to calculate total aggregate volume (X)
def total_aggregate_volume (volume_R, volume_S, volume_T, volume_U,
volume_V, volume_W):
    volume_X = cubic_yard_ft3 - volume_R - volume_S - volume_T - volume_U -
volume_V - volume_W
    return volume_X

# Example
R = 3.053
S = 0.697
T = 0.219
U = 0.431
V = 1.620
W = 5.385

# Print
X = total_aggregate_volume (R, S, T, U, V, W)
print(f'total aggregate volume (x): {X:.3f} ft^3')
```

Define a function to calculate total aggregate volume (X) Inputs: volume_R, volume_S, volume_T, volume_U, volume_V, volume_W. Returns: volume_X.

Cell 16

```
# User-defined cementitious material weights A, B, C and D
Cement_A = 600.0
fly_ash_B = 100.0
silica_fume_C = 30.0
other_scm_D = 70.0
# Specific gravities J, K, L and M
```

```
sg_cement_J = 3.15
sg_fly_ash_K = 2.30
sg_silica_fume_L = 2.20
sg_other_scm_M = 2.60
wc_ratio_E = 0.42
# Air content
air_content_F = 6.0

# We'll calculate Q from A, B, C, D and E before it is used to calculate W
water_weight_Q = calculate_water_weight_Q(cement_A, fly_ash_B,
silica_fume_C, other_scm_D,wc_ratio_E)

# Now let's use the functions we defined to calculate R, S, T, U, V and W
R = calculate_volume_cement_R(cement_A, sg_cement_J)
S = calculate_fly_ash_volume(fly_ash_B, sg_fly_ash_K)
T = silica_fume_volume(silica_fume_C, sg_silica_fume_L)
U = calculate_volume_U(other_scm_D, sg_other_scm_M)
V = calculate_other_air(air_content_F)
W = calculate_water_volume(water_weight_Q)
# Now we'll use R, S, T, U, V and W to calculate X:
X = total_aggregate_volume (R, S, T, U, V, W)
print(f'total aggregate volume (x): {X:.3f} ft^3')# Let's check our answer
```

User-defined cementitious material weights A, B, C and D Then it prints the result so you can see it.

Cell 17

```
# Define a function to calculate fine aggregate weight (Y)
def aggregate_weight (fine_aggregate_G, specific_gravity_N,
aggregate_volume_X, unit_weight_water):
    weight_W = unit_weight_water * (fine_aggregate_G / 100) *
specific_gravity_N * aggregaate_volume_X
    return weight_W


# Example for demonstration
fine_aggregate_G = 45.0
specific_gravity_N = 2.65
aggregate_volume_X = 15.596
FAW = aggregate_weight(fine_aggregate_G, specific_gravity_N,
aggregate_volume_X, unit_weight_water)

# Print
print(f'fine aggregate weight (x): {FAW:.3f} lb')
```

Define a function to calculate fine aggregate weight (Y) Inputs: fine_aggregate_G, specific_gravity_N, aggregate_volume_X, unit_weight_water. Returns: weight_W.

Cell 18

```
#function for coarce aggregate Weight
def coarse_aggregate (percent_coarse_H, sg_coarse_O, aggregate_volume_X,
unit_weight_water):
    weight_W = unit_weight_water * (percent_coarse_H / 100) * sg_coarse_O *
aggregate_volume_X
    return weight_W


# Example for demonstration
percent_coarse_H = 50.0
sg_coarse_O = 2.70
aggregate_volume_X = 15.596
CAW = coarse_aggregate(percent_coarse_H, sg_coarse_O, aggregate_volume_X,
unit_weight_water)

# Print
print(f'coarse aggregate weight (x): {FAW:.3f} lb')
```

function for coarce aggregate Weight Inputs: percent_coarse_H, sg_coarse_O, aggregate_volume_X, unit_weight_water. Returns: weight_W.

Cell 19

```
# Function for other aggregate weight (AA)
def calculate_other_aggregate_AA(percent_other_I,sg_other_P, volume_X):
    return unit_weight_water * (percent_other_I / 100) * sg_other_P *
volume_X

# Example for demonstration
percent_other_I = 5.0
sg_other_P = 2.60
volume_X = 15.596

AA = calculate_other_aggregate_AA(percent_other_I, sg_other_P, volume_X)

# Print
print(f'Other aggregate weight (AA): {AA:.1f} lb')
```

Function for other aggregate weight (AA) Inputs: percent_other_I, sg_other_P, volume_X. Returns: unit_weight_water * (percent_other_I / 100) * sg_other_P * volume_X.

Cell 20

```
# Integer input example
project_no = int(input("Enter project number: "))    # Example: 404222
```

```
# String input example
concrete_class = input("Enter class of concrete: ")    # Example: Class 47B

# Float input
sg_cement_J = float(input("Enter specific gravity of cement J: "))    #
Example: 3.15

# A message to confirm all inputs have been collected
print("\nAll user inputs collected successfully.")
```

Integer input example Then it prints the result so you can see it.

Cell 21

```
# General info
project_no = int(input("Enter project number: "))
concrete_class = input("enter class of concrete: ")

# Cementitious material inputs
cement_A = float(input("Enter cement weight A (lb per cubic yard): "))
fly_ash_B = float(input("Enter fly ash weight B (lb per cubic yard): "))
silica_fume_C =float(input("Enter silica fume weight (C lb per cubic yard):
"))
other_scm_D = float(input("Enter other SCM weight D (lb per cubic yard):
"))

# Design parameters
water_cement_ratio_E = float(input("Enter target water-cement ratio E: "))
air_content_F = float(input("Enter target air content F (%): "))

# Aggregate proportions
percent_fine_G = float(input("Enter percent fine aggregate G (%): "))
percent_coarse_H = float(input("Enter percent coarse aggregate H (%): "))
percent_other_I = float(input("Enter percent other aggregate I (%): "))

# Specific gravities
sg_cement_J = float(input("Enter specific gravity of cement J: "))
sg_fly_ash_K = float(input("Enter specific gravity of fly ash K: "))
sg_silica_fume_L = float(input("Enter specific gravity of silica fume L:
"))
sg_other_scm_M = float(input("Enter specific gravity of other SCM M: "))

sg_fine_N = float(input("Enter specific gravity of fine aggregate N: "))
sg_coarse_O = float(input("Enter specific gravity of coarse aggregate O:
"))
sg_other_P = float(input("Enter specific gravity of other aggregate P: "))

# A Print message to confirm all inputs have been collected
print("/nAll user inputs collected successfully.")
```

General info Then it prints the result so you can see it.

Cell 22

```
# Step 1: Water weight (Q)
Q = calculate_water_weight_Q(cement_A, fly_ash_B, silica_fume_C,
other_scm_D, water_cement_ratio_E)
```

Step 1: Water weight (Q)

Cell 23

```
# Step 2: Cementitious volumes (R, S, T, U)
R = calculate_volume_R(cement_A, sg_cement_J)
S = calculate_volume_S(fly_ash_B, sg_fly_ash_K)
T = calculate_volume_T(silica_fume_C, sg_silica_fume_L)
U = calculate_volume_U(other_scm_D, sg_other_scm_M)
```

Step 2: Cementitious volumes (R, S, T, U)

Cell 24

```
# Step 3: Air and water volumes (V, W)
V = calculate_air_volume_V(air_content_F)
W = calculate_water_volume_W(Q)
```

Step 3: Air and water volumes (V, W)

Cell 25

```
# Step 4: Total aggregate volume (X)
X = calculate_total_aggregate_volume_X(R, S, T, U, V, W)
```

Step 4: Total aggregate volume (X)

Cell 26

```
# Step 5: Aggregate weights (Y, Z, AA)
Y = calculate_fine_aggregate_Y(percent_fine_G, sg_fine_N, X)
Z = calculate_course_aggregate_Z(percent_coarse_H,sg_coarse_O, X)
AA = calculate_other_aggregate_Z(percent_other_I,sg_other_P, X)
```

Step 5: Aggregate weights (Y, Z, AA)

Cell 27

```
# We'll use --- lines to organize our output
# OPTIONAL: round your values according to the NDOT mix design Excel sheet

# Start with a general explanation
print("\n-------------------------------------------")
print(" NDOT Concrete Mix Design – Weight Summary")
print("         (1 Cubic Yard of Concrete)")

print("-------------------------------------------")

# Project and mix details


print("-------------------------------------------")

# Cementitious material outputs


print("-------------------------------------------")

# Aggregate outputs


print("-------------------------------------------")

# Water output


# End it with a note
print("-------------------------------------------")
```

We'll use --- lines to organize our output OPTIONAL: round your values according to the NDOT mix design Excel sheet Start with a general explanation Then it prints the result so you can see it.

Cell 28

```
(empty cell)
```

Runs the code in this cell to set up variables or perform the next step.

Cell 29

```
(empty cell)
```

Runs the code in this cell to set up variables or perform the next step.

Cell 30

```
(empty cell)
```

Runs the code in this cell to set up variables or perform the next step.

Cell 31

```
# NDOT Concrete Mix Design – Full Integrated Run

# Project metadata


# Cementitious material inputs (lb per cubic yard)


# Aggregate proportions (%)


# Specific gravities


# Calculations


# Output: Final Weight Chart
```

NDOT Concrete Mix Design – Full Integrated Run Project metadata Cementitious material inputs (lb per cubic yard) Aggregate proportions (%) Specific gravities Calculations Output: Final Weight Chart

Cell 32

```
(empty cell)
```

Runs the code in this cell to set up variables or perform the next step.