Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

Formula Sheet

Standard Deviation $\sigma = \sqrt{\sigma^2}$

Variance $\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$

Permutation $nPr = \frac{n!}{(n-r)!}$

Multinomial Coefficients $N = \frac{n!}{n_1!n_2!\ldots n_k!}$

Combination $\binom{n}{r} = \frac{n!}{r!(n-r)!}$

Conditional Probability $P(A|B) = \frac{P(A \cap B)}{P(B)}$

Independent $P(A|B) = P(A)$

$P(B|A) = P(B)$

$P(A \cap B) = P(A)P(B)$

Probability Mass Function $P(X = x)$

Binomial Distribution $P_x = \binom{n}{x} p^x q^{n-x}$

Geometric Distribution

$P(X = x) = q^{y-1}p$

$E(Y) = \frac{1}{p}$ and $V(Y) = \frac{(1-p)}{p^2}$

Hypergeometric Distribution

$n_A = \binom{r}{y}\binom{N-r}{n-y}$

$\mu = E(Y) = \frac{nr}{N}$ and

$\sigma^2 = V(Y) = n(\frac{r}{N})(\frac{N-r}{N})(\frac{N-n}{N-1})$

Poisson Distribution

$p(y) = \frac{\lambda^y}{y!}e^{-\lambda}$ and $\mu = \lambda$ and $\sigma^2 = \lambda$

Multiplicative Law of Probability

$P(A \cap B) = P(A)P(B|A) = P(B)P(A|B)$

General Addition Rule

$P(A \cup B) = P(A) + P(B) - P(A \cap B)$

Bayes Theorem

$P(A|B) = \frac{P(A|B)\,P(B)}{P(A)}\ w/\ P(A) > 0\ and\ P(B) > 0$

Theorem of Total Probability $if\ 0 < P(B) < 1$

$P(B|A) = \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|B)P(B)}$

Discrete Random Variables Expectations and Variance

$E[Y] = \sum_{y \in Y} yP(y)$ | $V[Y] = E[(Y - \mu)^2]$ | $\mu = E[Y]$

Standard Deviation of Y is $\sqrt{V[Y]}$

Extra Formulas (Derived Geometric Distribution)

$P(X \leq n) = 1 - (1 - p)^n$

$P(X < n) = 1 - (1 - p)^{n-1}$

$P(X \geq n) = (1 - p)^{n-1}$

$P(X > n) = (1 - p)^n$

Negative Binomial Probability Distribution

$p(y) = \binom{y-1}{r-1} p^r q^{y-r}$ and $\mu = \frac{r}{p}$ and $\sigma^2 = \frac{r(1-p)}{p^2}$

Tchebysheff's Theorem $1 - \frac{1}{k^2}$

$k = \frac{the\ "within"\ number}{the\ standard\ deviation}\ and\ k\ must\ be > 1$

Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

Github

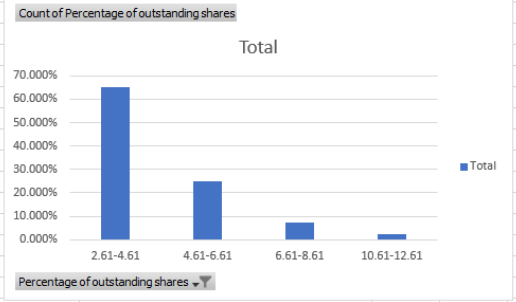
Git is a version control system that is considered the most common that people utilize. A user-friendly workflow is due to its flexibility in how its users can manage various changes, and there is no one way to interact with it or no standardized process. Git allows people to collaborate on projects and produce changes in a flow. Before teaming up, everyone should be on the same page and agree upon a workflow with several types available. It is recommended to use the Git workflow to accomplish the goal of productivity and consistency. The workflow helps to encourage and enhance the effectiveness of the team of developers and users. Before choosing any workflow, make sure it scales to the size of the team, that it is easy to undo mistakes and errors, and whether or not it creates unnecessary cognitive overheads to the team. To begin working with this workflow, users should understand how it functions. First, cloning the central repository will allow users to own a local copy of the project to edit files and commit changes, which will be stored locally. When ready to publish these changes to the official project, the user will have to push their local main branch to the central repository of the main project. The article "Comparing Git workflows: What you should know," by Atlassian, states that cloning will automatically create a shortcut, origin, pointing back to the parent repository. To make a change, the user, as mentioned, will have to commit it, editing, staging, and commit. The commit will be pushed to the central repository, and as the article states, once the local repository has these new changes committed, it will require a push to share with other developers on the project and to the central repository. This central repository is the official project with a history of commits. Sometimes, making changes to what anyone else may have done already will cause the flow to pause the rebasing process and let a developer of the team manually resolve any conflicts. If one commits their changes and pushes them to the origin, other teammates will solve the conflict, get their changes, and integrate with their local changes. Another developer can use git pull, which pulls the entire upstream commit history in their local repository to attempt integration with their local commits. Atlassian states that it is better to rebate so all commits go to the main branch tip, and if not, a merge commit will transfer each local commit to the updated main branch one at a time.

Git also allows collaborators to resolve other merge conflicts by locating where any problem is. To go over all the necessary terminology of git, start with git commit. A commit will take a staged snapshot and commit it to a project history. Combining this with git add, which is the basic workflow. Git merge is another way to integrate changes from divergence branches after the project history combines with the git branch and merge will put it all together again. Git pull is like git fetch but automated. It will download a branch from a remote repository and merge it into the current branch. Lastly is git push, the opposite of fetching, which allows one to move a local branch to another repository, allowing users to publish contributions. It will send a series of commits instead of a single changeset.

Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

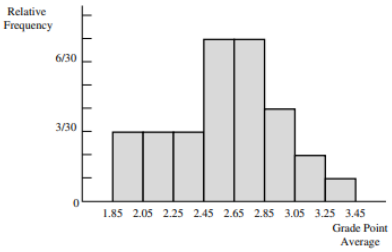
Histogram

Row Labels	Count of Percentage of outstanding shares
2.61-4.61	65.000%
4.61-6.61	25.000%
6.61-8.61	7.500%
10.61-12.61	2.500%
Grand Total	**100.000%**



Histogram | Data | Histogram 1.4 | Data 1.4

1.5 Given here is the relative frequency histogram associated with grade point averages (GPAs) of a sample of 30 students:

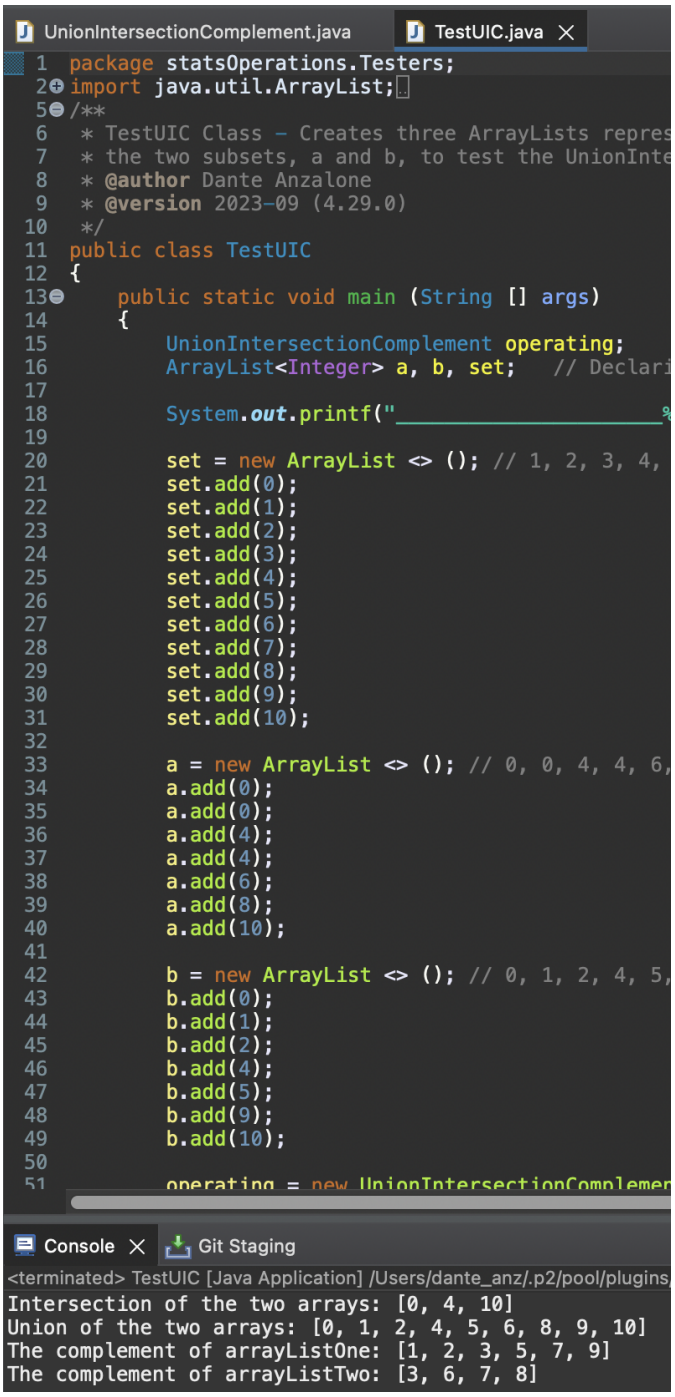


a Which of the GPA categories identified on the horizontal axis are associated with the largest proportion of students?
b What proportion of students had GPAs in each of the categories that you identified?
c What proportion of the students had GPAs less than 2.65?

- a → Categories 2.45-2.85
- b → 14 / 30 students
- c → 16 / 30 students

Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

Set Theory: Union, Intersection, and Complement

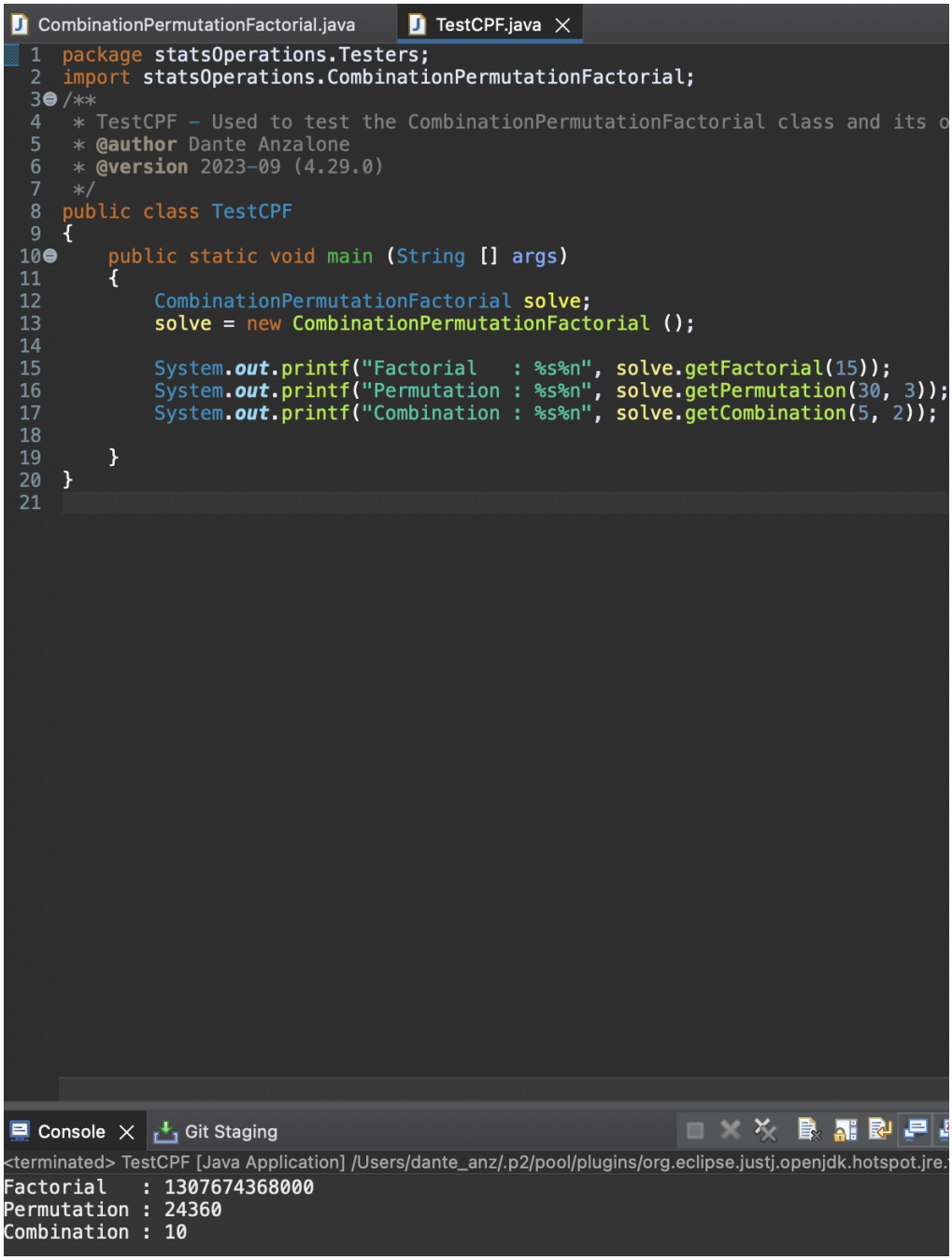


Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

Mean, Median, Mode, Standard Deviation, and Variance

Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

Permutations, Combinations, and Factorial



Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

Determining Independence, and Conditional and Bayes

ConditionalBayesDependence.java | TestCBD.java

```
64          System.out.printf("Finding probability intersection of A and C : %.2f%n", (solve.probIntersection(a, c, totalObjects)));
65
66          a.clear();
67          b.clear();
68          c.clear();
69          /*=============================================================================================================*/
70
71          // For Bayes-Theorem problem
72
73          a.add(2); // Die containing numbers 2, 4, and 5
74          a.add(4);
75          a.add(6);
76
77          b.add(5); // Die containing numbers 5 and 6
78          b.add(6);
79
80          totalObjects = 6;
81          System.out.printf("Bayes Theorem result: %.2f%%%n", solve.bayesTheorem(a, b, totalObjects) * 100);
82          a.clear();
83          b.clear();
84          /*=============================================================================================================*/
85
86          // Testing if two sets are mutually or not mutually exclusive
87
88          a.add(2);
89          a.add(4);
90          a.add(8);
91
92          b.add(1);
93          b.add(3);
94          b.add(5);
95
96          c.add(2);
97          c.add(3);
98
99          totalObjects = 6;
100         System.out.printf("Is the sets A and B mutually exclusive? : %s%n", solve.isMutuallyExclusive(a, b, totalObjects));
101         System.out.printf("Is the sets A and C mutually exclusive? : %s%n", solve.isMutuallyExclusive(a, c, totalObjects));
102
103         System.out.printf("Finding probability union of A and B : %.2f%n", (solve.findProbUnion(a, b, totalObjects)));
104         System.out.printf("Finding probability union of A and C : %.2f%n", (solve.findProbUnion(a, c, totalObjects)));
105
106         /*=============================================================================================================*/
107
```

Console | Git Staging

```
<terminated> TestCBD [Java Application] /Users/dante_anz/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.8.v20230831-1047/jre/bin/java  (Oct 24, 2023, 1
The probability of A given B is: 33.33%
Is the sets A and B independent? : false
Is the sets A and C independent? : true
Finding probability intersection of A and B : 0.00
Finding probability intersection of A and C : 0.17
Bayes Theorem result: 33.33%
Is the sets A and B mutually exclusive? : true
Is the sets A and C mutually exclusive? : false
Finding probability union of A and B : 1.00
Finding probability union of A and C : 0.67
```

Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

Distributions

Distributions.java | TestDistributions.java | Stats Library Classes and Methods

```
60
61          // Testing Geometric Distribution
62
63          p = .5;
64          y = 5;
65
66          System.out.printf("%nThe Geometric Distribution is: %.2f%%", solve.geometric(p, y)*100);
67          System.out.printf("%nThe Geometric Expectation is: %.2f", solve.expectedGeometric(p));
68          System.out.printf("%nThe Geometric Variance is: %.2f%n", solve.varianceGeometric(p));
69
70
71          /*=========================================================================================
72
73          // Testing Poisson Distribution
74
75          int lambda;
76
77          lambda = 1;
78          y = 0;
79
80          System.out.printf("%nThe Poisson Distribution is: %.2f%%", solve.poisson(lambda, y));
81          System.out.printf("%nThe Poisson Expectence is: %.2f", solve.expectedPoisson(lambda));
82          System.out.printf("%nThe Poisson Variance is: %.2f%n", solve.variancePoisson(lambda));
83
84          /*=========================================================================================
85
86          // Testing Tchebysheff's Theorem
87
88          double k;
89
90          k = 2;
91
```

Console | Git Staging

```
<terminated> TestDistributions [Java Application] /Users/dante_anz/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17
The Hypergeometric Distribution is: 1.35%
The Hypergeometric Expectence is: 1.50
The Hypergeometric Variance is: 0.83

The Negative Binomial Probability Distribution is: 3.07%
The Negative Binomial Expectence is: 15.00
The Negative Binomial Variance is: 60.00

The Binomial Distribution is: 20.13%

The Geometric Distribution is: 3.13%
The Geometric Expectation is: 2.00
The Geometric Variance is: 2.00

The Poisson Distribution is: 0.37%
The Poisson Expectence is: 1.00
The Poisson Variance is: 1.00

The Tchebysheff's Theorem result w/ k given is: 75.00%
The Tchebysheff's Theorem result is: 75.00%
```

Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

Birthday Checker Program

Birthday.java | Person.java | Main.java

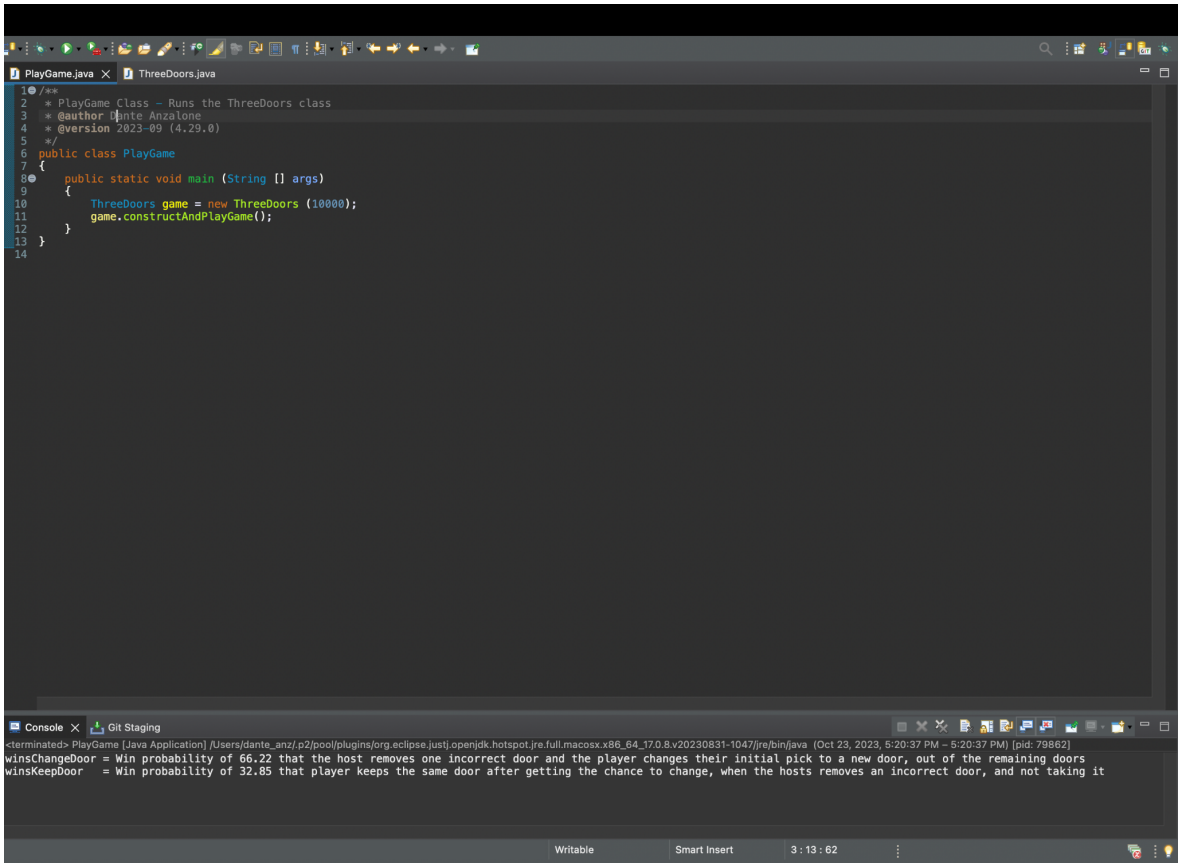
```
1  package birthday;
2  /**
3   * Main Class - Runs the class Birthday
4   * @author Dante
5   * @version 2023-09 (4.29.0)
6   */
7  public class Main
8  {
9      public static void main (String [] args)
10     {
11         Birthday days;
12         days = new Birthday();
13         days.run();
14     }
15 }
16
```

Console | Problems | Debug Shell | Git Repositories | Git Staging

```
<terminated> Main [Java Application] /Users/dante_anz/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.8.v2
How many people are in your class?
30
12 : 12
12 : 10
8 : 13
5 : 4
1 : 24
12 : 25
6 : 6
6 : 6
9 : 23
3 : 20
6 : 10
1 : 10
9 : 24
12 : 29
4 : 29
9 : 30
7 : 10
8 : 18
9 : 8
8 : 15
12 : 24
9 : 20
5 : 18
2 : 9
1 : 29
5 : 6
4 : 23
9 : 28
1 : 29
11 : 27
2.0
The probability that at least two people share the same birthday in class is : 0.06666666666666667
```

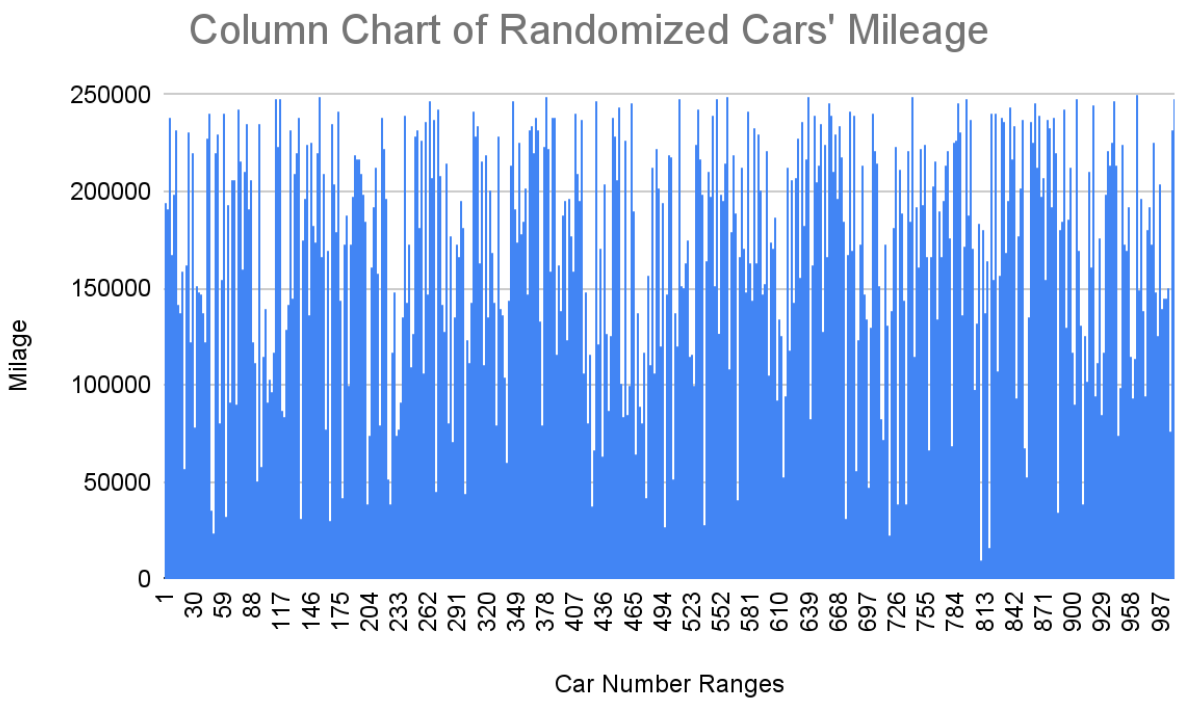
Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

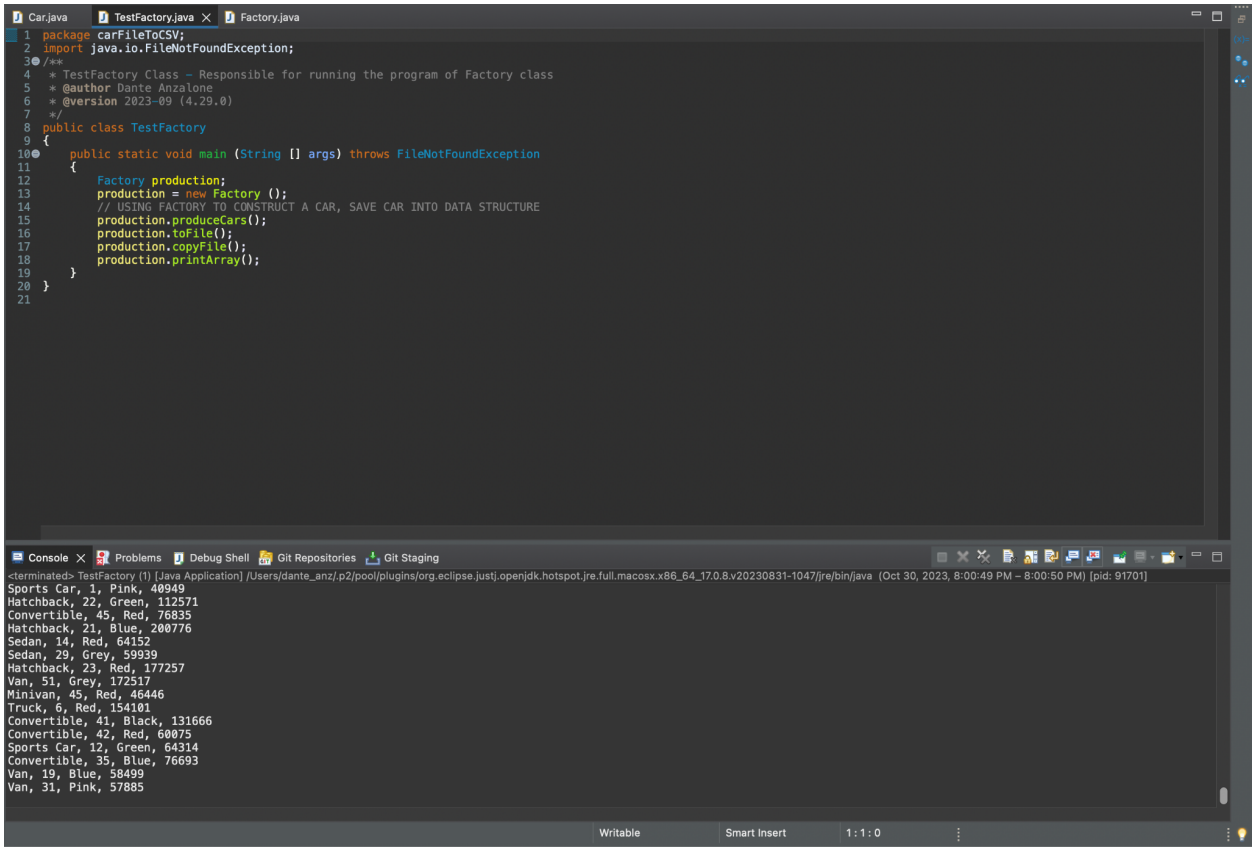
Monte's 3 Doors / Deal or No Deal Program



Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

Car Factory Program





Dante Anzalone's Assignments and Program Screenshots
CSCI 3327: Probability and Applied Statistics

Monte Carlo Simulation (Hands)