

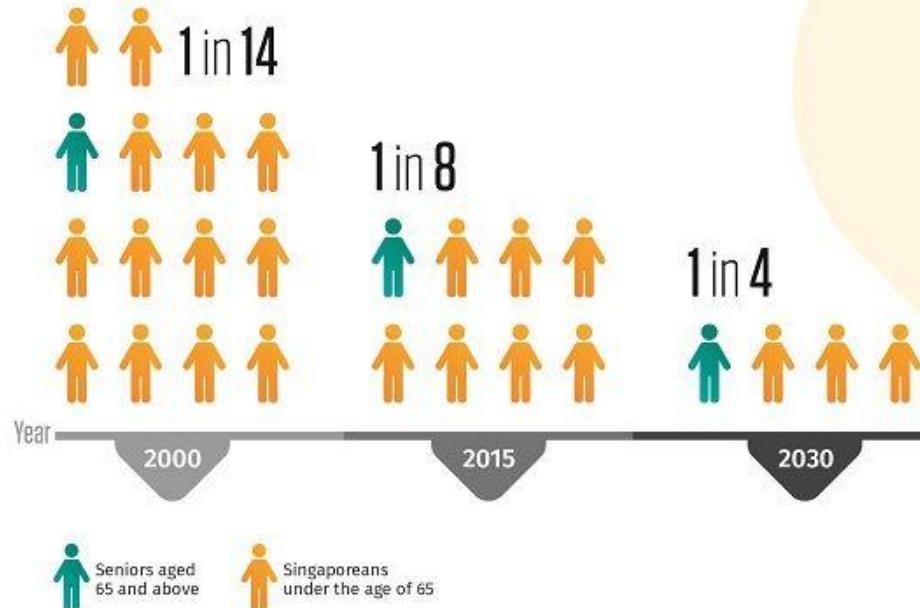
SC1015 Team Project

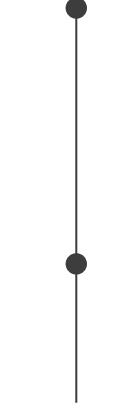
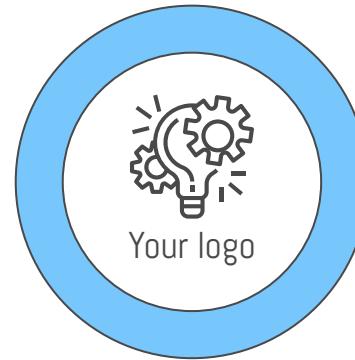
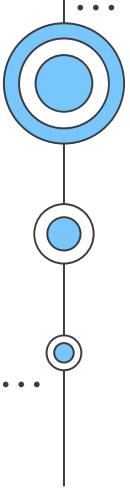
Done by:
Anthony Soon (U2222577H)
Benjamin Fernandez (U2220316G)
Goh Yi Heng (U2222525C)

Introduction

Our Motivation

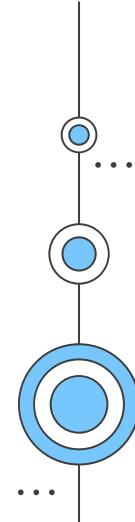
OUR POPULATION IS AGEING RAPIDLY



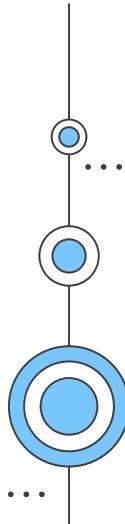
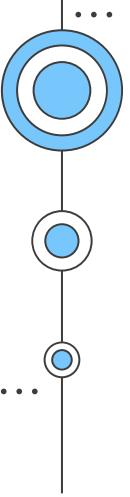


Problem Definition:

How do different factors affect life expectancy?
How can we explain Singapore's aging population
with this?



Methodology



Our Plan/Methodology



Data of All Countries

Explore the life expectancy data of all countries

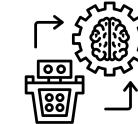
...



Pick 3 main factors

Using correlation, pick 3 best factors to analyse

...



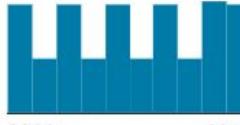
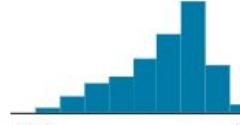
ML techniques

Use a new & different ML tool to further improve accuracy

...

Dataset: Life Expectancy (WHO)

The Global Health Observatory (GHO) data repository under World Health Organization (WHO) keeps track of the health status as well as many other related factors for all countries. The datasets are made available to public for the purpose of health data analysis. The dataset related to life expectancy, health factors for 193 countries has been collected from the same WHO data repository website and its corresponding economic data was collected from United Nation website. Among all categories of health-related factors only those critical factors were chosen which are more representative. It has been observed that in the past 15 years, there has been a huge development in health sector resulting in improvement of human mortality rates especially in the developing nations in comparison to the past 30 years. Therefore, in this project we have considered data from year 2000-2015 for 193 countries for further analysis. The individual data files have been merged together into a single dataset. On initial visual inspection of the data showed some missing

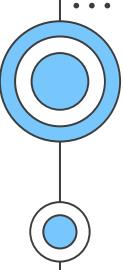
# Country	# Year	# Status	# Life expectancy	# Adult Mortality	# infant
Country	Year	Developed or Developing status	Life Expectancy in age	Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)	Number c per 1000
193 unique values		Developing 83% Developed 17%			
Afghanistan	2015	Developing	65	263	62
Afghanistan	2014	Developing	59.9	271	64

Cleaning of Data

Country	Year	Status	Life expect	Adult Mort	infant dea	Alcohol	percentage	Hepatitis E	Measles	E
Algeria	2002	Developing	71.6	145	20	0.36	148.512		5862	
Algeria	2001	Developing	71.4	145	20	0.23	147.9861		2686	
Algeria	2000	Developing	71.3	145	21	0.25	154.4559		0	
Angola	2015	Developing	52.4	335	66	0	0	64	118	

Removing rows with incomplete data

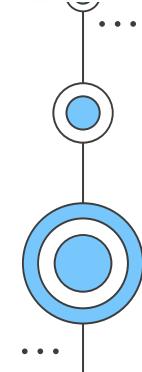
Removing non-numerical data



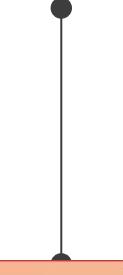
Picking the best factors

Using Correlation Matrix

Life expectancy	0.05	1.00	-0.70	-0.17	0.40	0.41	0.20	-0.07	0.54	-0.19	0.33	0.17	0.34	-0.59	0.44	-0.02	-0.46	-0.46	0.72	0.73
Year																				
Life expectancy																				
Adult Mortality																				
Infant deaths																				
Alcohol																				
percentage expenditure																				
Hepatitis B																				
Measles																				
BMI																				
under-five deaths																				
Polio																				
Total expenditure																				
Diphtheria																				
HIV/AIDS																				
GDP																				
Population																				
thinness 1-19 years																				
thinness 5-9 years																				
Income composition of resources																				
Schooling																				

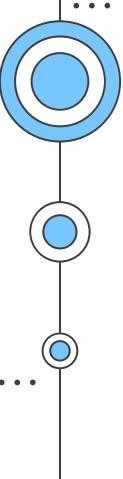


Ranking based on the magnitude of correlation:

1. Schooling (0.73)
 2. Income Composition of resources (0.72)
 3. Adult Mortality (-0.70)
 4. HIV/AIDS (-0.59)
 5. BMI (0.54)
- 

Findings





Train Set and Test Set

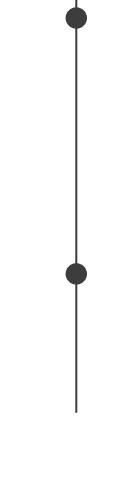
General Steps

Splitting a set of data into 2 groups : “Train” and “Test” where the “Train” group has the larger set of data.

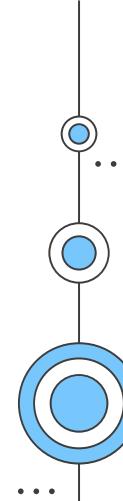
Implementing a statistical technique on the “Train” set and using the resultant predictive values on the “Test” set

Brief Information

To test the accuracy of a prediction method using a set of data where the actual data is already known.



...



Code & Explanation

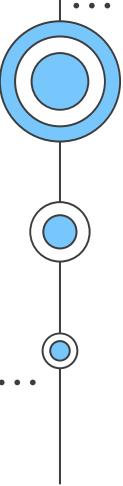
```
# Train Set : 1154 samples
schooling_train = pd.DataFrame(schooling[:1154])
life_exp_train = pd.DataFrame(life_exp[:1154])

# Test Set : 495 samples
schooling_test = pd.DataFrame(schooling[-495:])
life_exp_test = pd.DataFrame(life_exp[-495:])

# Check the sample sizes
print("Train Set :", schooling_train.shape, life_exp_train.shape)
print("Test Set  :", schooling_test.shape, life_exp_test.shape)
```

- Import the panda module
- Input the number of data used to form both the "train" and "test" sets.





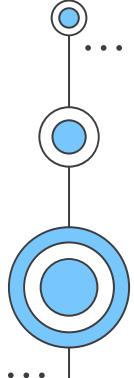
Linear Regression

General Equation

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

Brief Information

Linear Regression is a common statistical technique for modelling linear relationships between 2 variables. A result between -1 to 1 is given where the higher the magnitude, the stronger the correlation. A negative value signifies an inverse relation, a positive value is a positive relation. A magnitude of 1 is a perfect relation, where 0 shows no relation between both variables.



Code & Explanation

```
# Import LinearRegression model from Scikit-Learn
from sklearn.linear_model import LinearRegression

# Create a Linear Regression object
linreg = LinearRegression()

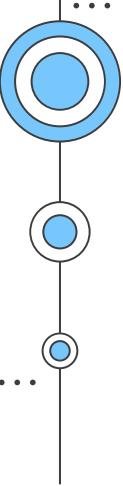
# Train the Linear Regression model
linreg.fit(schooling_train, life_exp_train)

# Coefficients of the Linear Regression line
print('Intercept \t: b = ', linreg.intercept_)
print('Coefficients \t: a = ', linreg.coef_)

# Predict Total values corresponding to Schooling Train
life_exp_train_pred = linreg.predict(schooling_train)
•
# Plot the Linear Regression line
f = plt.figure(figsize=(16, 16))
plt.scatter(schooling_train, life_exp_train)
plt.scatter(schooling_train, life_exp_train_pred, color = "r")
plt.show()
```

- Import the necessary modules and functions
- Input the sets of data required
- Calculate the coefficient of the regression model
- Plot the regression line





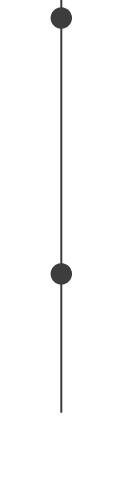
Mean Squared Error

General Equation

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Brief Information

Mean squared error is a technique to calculate deviations between a predicted set of results as compared to the actual results. The higher the MSE, the greater the inaccuracy of the prediction method used. Compared to its alternative, "Mean Absolute Error", MSE places greater weightage for actual values that deviate more from the predicted value. This is because the larger errors are squared to be even larger values before summation.



Code & Explanation

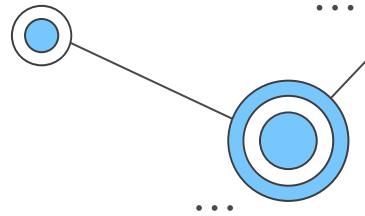
```
# Mean Squared Error (MSE)
def mean_sq_err(actual, predicted):
    '''Returns the Mean Squared Error of actual and predicted values'''
    return np.mean(np.square(np.array(actual) - np.array(predicted)))

mse = mean_sq_err(life_exp_train, life_exp_train_pred)
print("Mean Squared Error (MSE) \t:", mse)
print("Root Mean Squared Error (RMSE) \t:", np.sqrt(mse))
```

- Uses the “numpy” library
- Calculate the difference in actual and predicted values
- Square the absolute error
- Sum all the errors
- Find the mean of the errors



Errors

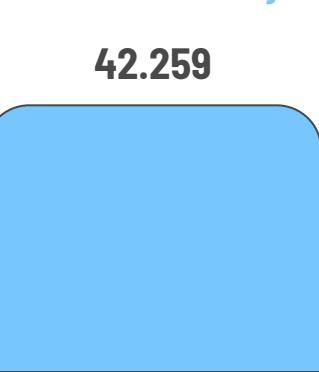
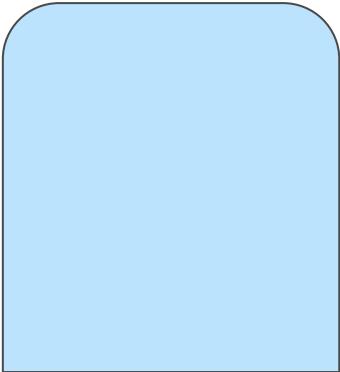


1st Place:
Income
Composition

34.220

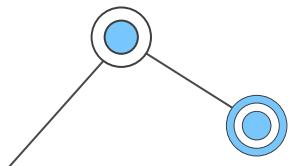


2nd Place:
Schooling
34.274



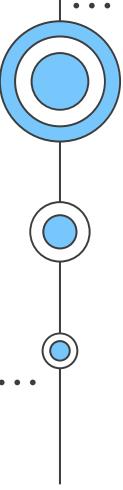
3rd Place:
Adult Mortality

42.259



New ML application

Polynomial Regression

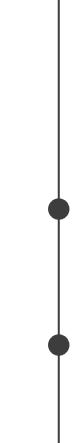


Polynomial Regression



General Equation

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n + \varepsilon.$$



Brief Information

Polynomial Regression is an advanced statistical technique for modeling non-linear relationships between variables. It extends linear regression by incorporating higher-degree polynomial terms, allowing for more accurate predictions. However, it can also present challenges, such as overfitting and increased computational complexity.

Code & Explanation

```
In [24]: from sklearn.preprocessing import PolynomialFeatures  
from sklearn.linear_model import LinearRegression  
from sklearn.metrics import mean_squared_error
```

```
In [25]: # Check the MSE with different Degree value, the lowest the better. Also, the  
# higher score the better  
number_degrees = [1,2,3,4,5,6,7]      # Can add more inside but with higher  
degree, it might overfit the data  
MSE_TrainSet = []        # Empty list to store the MSE for train for each degree  
MSE_TestSet = []          # Empty List to store the MSE for test for each degree  
scores = []                # List to store the accuracy score for test set for each  
degree  
for degree in number_degrees:  
    print("Degree : \t", degree)  
  
    poly_model = PolynomialFeatures(degree = degree)  
    poly_x_train = poly_model.fit_transform(X_train)  
  
    poly_x_test = poly_model.fit_transform(X_test)  
  
    poly_model.fit(poly_x_train, y_train)  
  
    regression_model = LinearRegression()  
  
    model = regression_model.fit(poly_x_train, y_train)  
  
    y_train_pred = model.predict(poly_x_train)  
    y_test_pred = model.predict(poly_x_test)  
  
    train_score = model.score(poly_x_train,y_train)  
    test_score = model.score(poly_x_test,y_test)  
  
    MSETrain = mean_squared_error(y_train, y_train_pred, squared=False)  
    MSETest = mean_squared_error(y_test, y_test_pred, squared=False)
```

- Import the necessary modules and functions
- New function : Polynomial Features to support polynomial regression
- Iterate through different degrees for polynomial equation ; in this case we used 1 - 7
- Have a small range of degrees as using too many has its drawbacks



Code & Explanation

```
scores.append(test_score)
print("MSE train:\t", MSETrain)
print("MSE test:\t", MSETest)
print("train score:\t",train_score)
print("test score: \t",test_score, "\n")

MSE_TrainSet.append(MSETrain)
MSE_TestSet.append(MSETest)

# Print MSE with different degree values
f, axes = plt.subplots(3, figsize=(24, 12))
axes[0].set_title("MSE train Set vs Degree")
axes[0].scatter(number_degrees,MSE_TrainSet, color="green")
axes[0].plot(number_degrees,MSE_TrainSet, color="red")

axes[1].set_title("MSE test Set vs Degree")
axes[1].scatter(number_degrees,MSE_TestSet, color="green")
axes[1].plot(number_degrees,MSE_TestSet, color="red")

axes[2].set_title("Accuracy Score vs Degree")
axes[2].scatter(number_degrees,scores, color="green")
axes[2].plot(number_degrees,scores, color="red")

# auto calculated the lowest MSE degree on test set

# Take note here, this 2 should be the same index in order to determine that
that might be the best degree
Best_Degree_MSE = MSE_TestSet.index(min(MSE_TestSet))+1
print("The lowest MSE degree: \t",Best_Degree_MSE)
Best_Degree_Score = scores.index(max(scores))+1
print("The highest Accuracy degree: \t",Best_Degree_Score)
```

- Create and run train and test sets
- MSE (mean square error) used as indicator of accuracy
- Plot graph for visualization for each polynomial value relation to MSE
- Form logical conclusion based on data



Interpreting Results

Degree : 1
MSE train: 3.2351766285382806
MSE test: 3.053311494607925
train score: 0.8722018696754889
test score: 0.8585441048688139

Degree : 2
MSE train: 2.6919811211900457
MSE test: 2.5757505188193184
train score: 0.9115144149117279
test score: 0.8993331642135977

Degree : 3
MSE train: 2.3414630793407496
MSE test: 2.4047557589656026
train score: 0.933057313772272
test score: 0.9122553243702117

Degree : 4
MSE train: 2.1264679980716825
MSE test: 2.4074684919231517
train score: 0.9447863815605018
test score: 0.9120572484295676

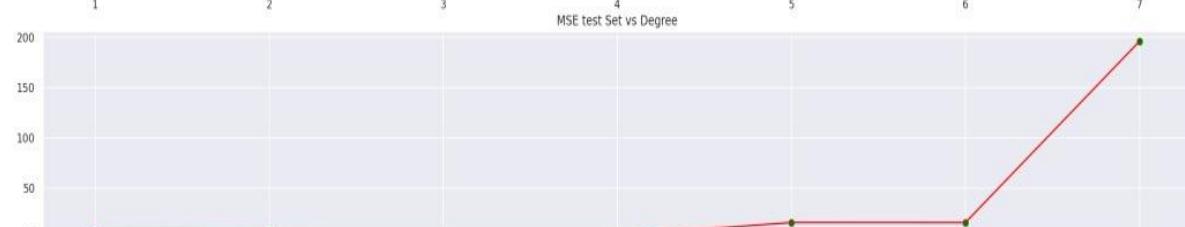
Degree : 5
MSE train: 1.9445165453581863
MSE test: 15.799145002816163
train score: 0.9538038591710387
test score: -2.78744204410231

Degree : 6
MSE train: 1.9057906192464333
MSE test: 15.754122971809277
train score: 0.9556515059499527
test score: -2.765887032822175

Degree : 7
MSE train: 1.7688030733823668
MSE test: 195.9116483156122
train score: 0.961797879025703
test score: -581.3704670735393

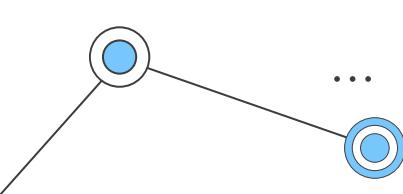
The lowest MSE degree: 3

The highest Accuracy degree: 3

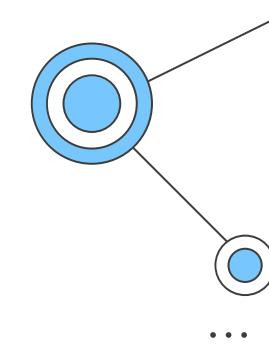


- Train and test score provided for degree (n)

- Model with highest accuracy == lowest MSE is at n = 3



Possible Drawbacks



01

Overfitting

Polynomial regression is prone to overfitting if too many degrees are used to find the best fit - especially when outliers exist due to improper cleaning of data

02

Computational Complexity

Higher-degree polynomials involve more complex calculations, which can lead to increased computational time and resources required.

03

Multicollinearity

Polynomial regression can suffer from multicollinearity especially at higher degrees - this makes it difficult to interpret each coefficients significance and contribution

Conclusion

Factors

- Adult Mortality
- Schooling
- Income composition

01

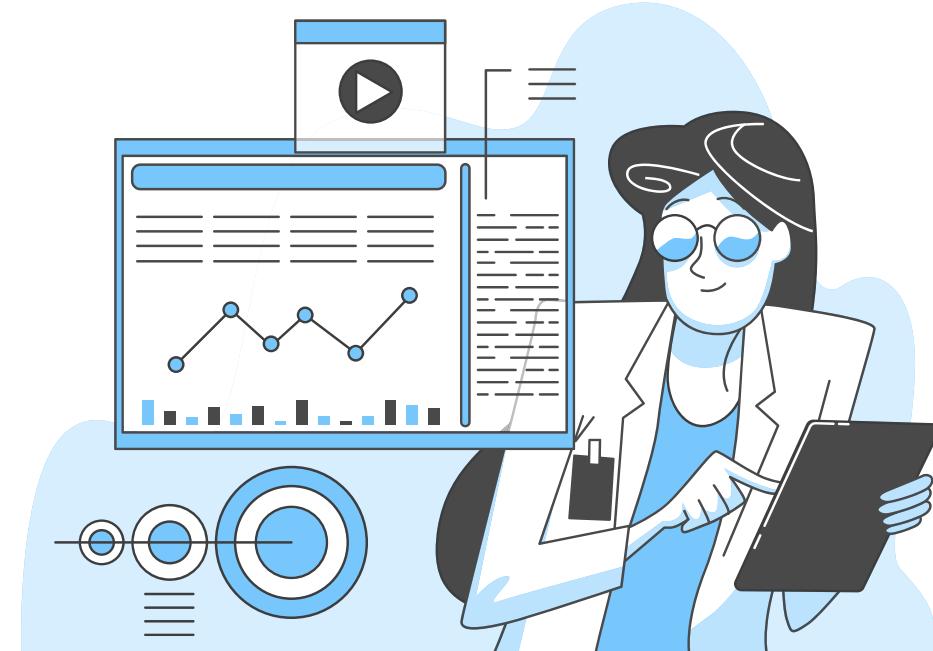
...

02

...

Results in Singapore's context

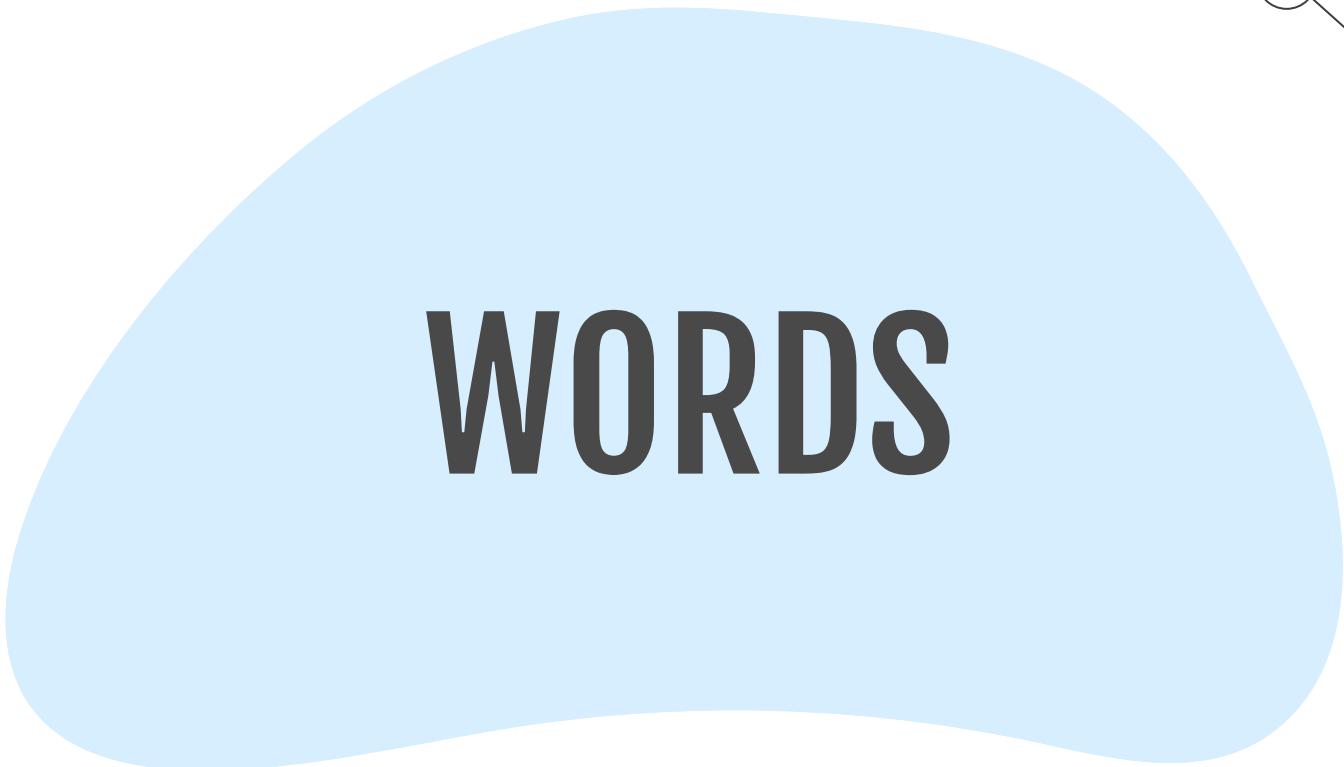
- Schooling, Income has increased
- Adult mortality decreased
- Explains aging population in Singapore



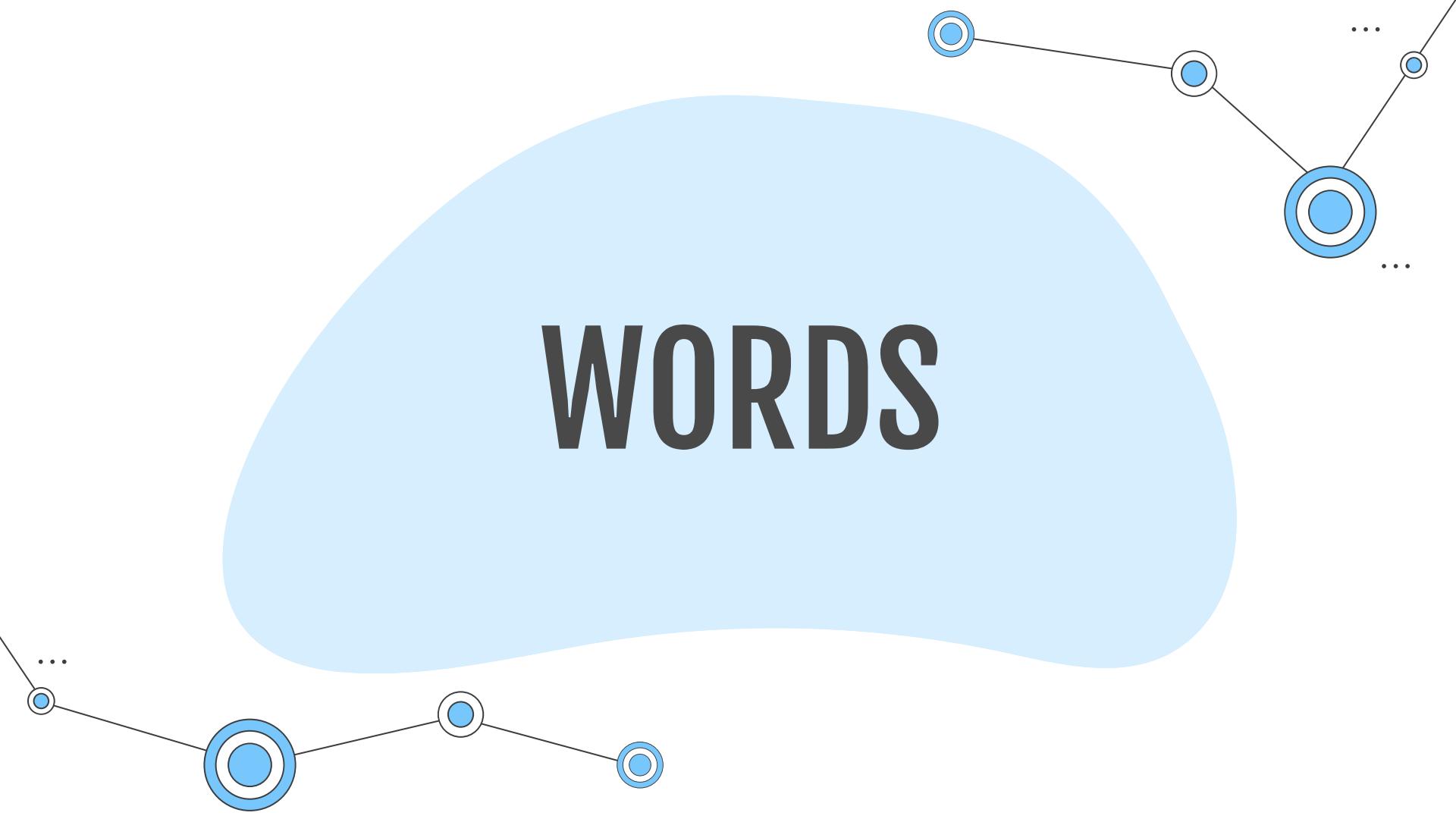


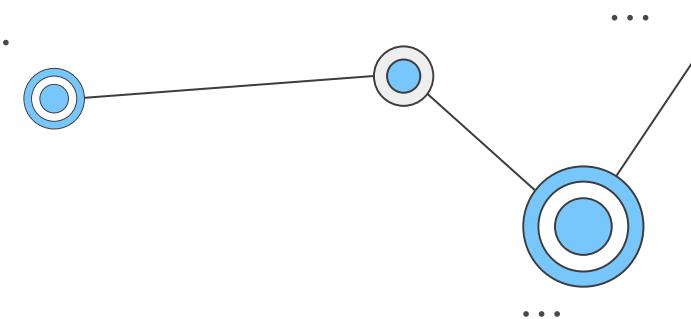
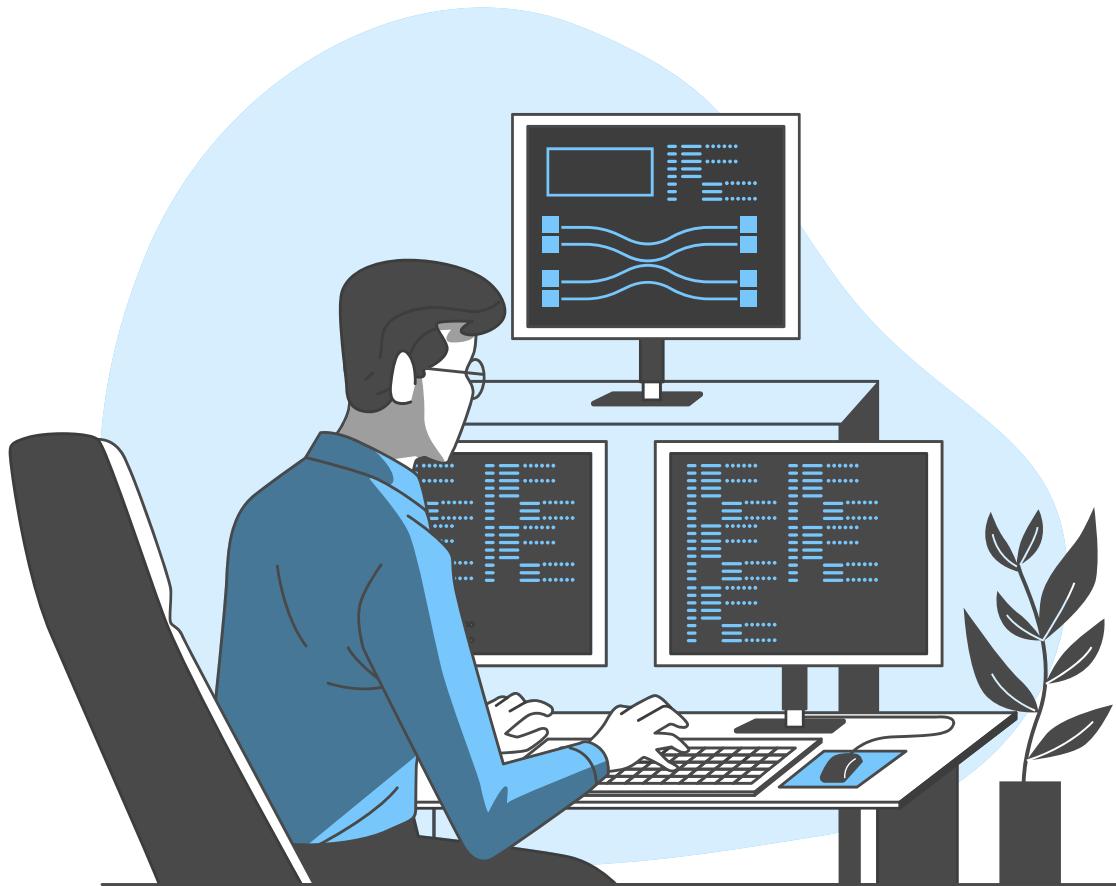
Thank You





WORDS





Technology Consulting

Here is where your
presentation begins

Conclusion

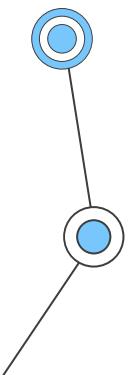
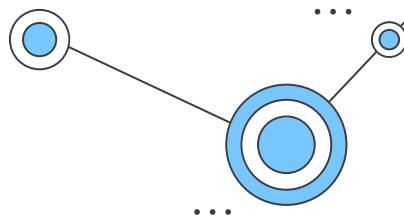


Table of Contents



Problem & Solution

You could describe the topic of the section here



Market Analysis and Target

You could describe the topic of the section here



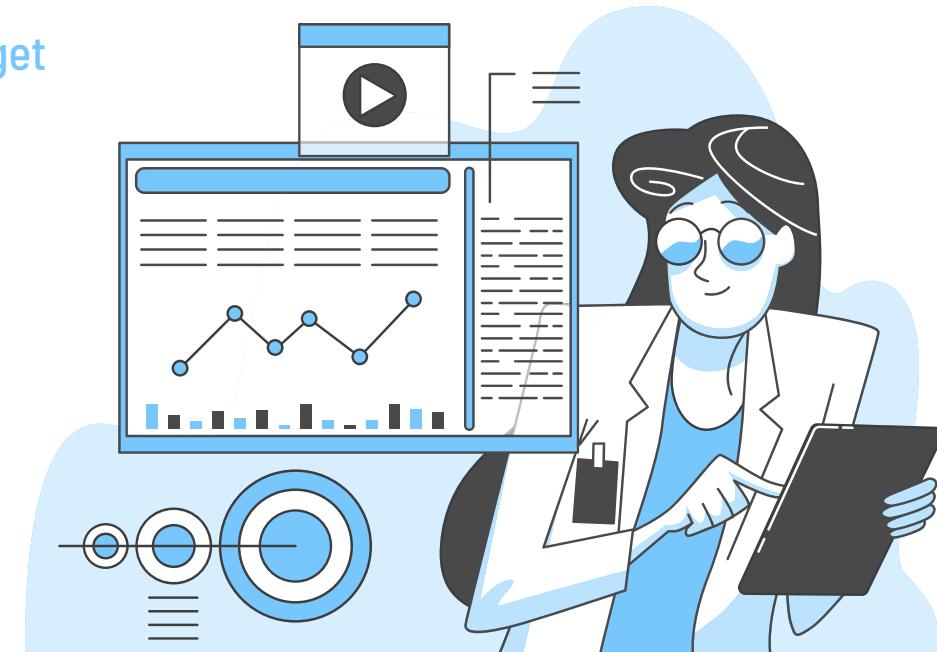
Our Process

You could describe the topic of the section here



Our Consultants

You could describe the topic of the section here



01

Company

Enter a subtitle here if you need it

Understanding the Problem



Neptune

Neptune is the farthest planet from the Sun

...



Mercury

Mercury is the smallest planet in the Solar System

...



Jupiter

Jupiter is the biggest planet in the Solar System

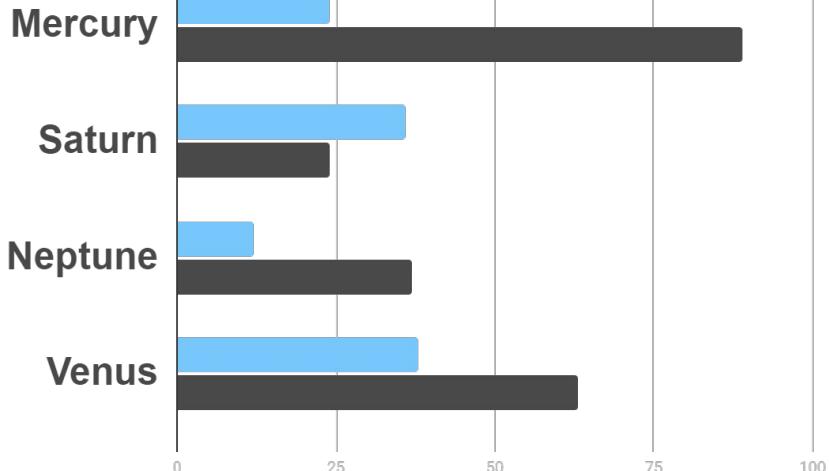
...

Main Competitors

	Mercury	Jupiter	Venus	Neptune
🚀	✖	✖	✖	✖
💰	✖	✓	✖	✓
🏆	✓	✖	✓	✖
coins	✓	✓	✖	✖

Market Research

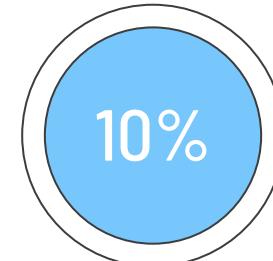
Evolution



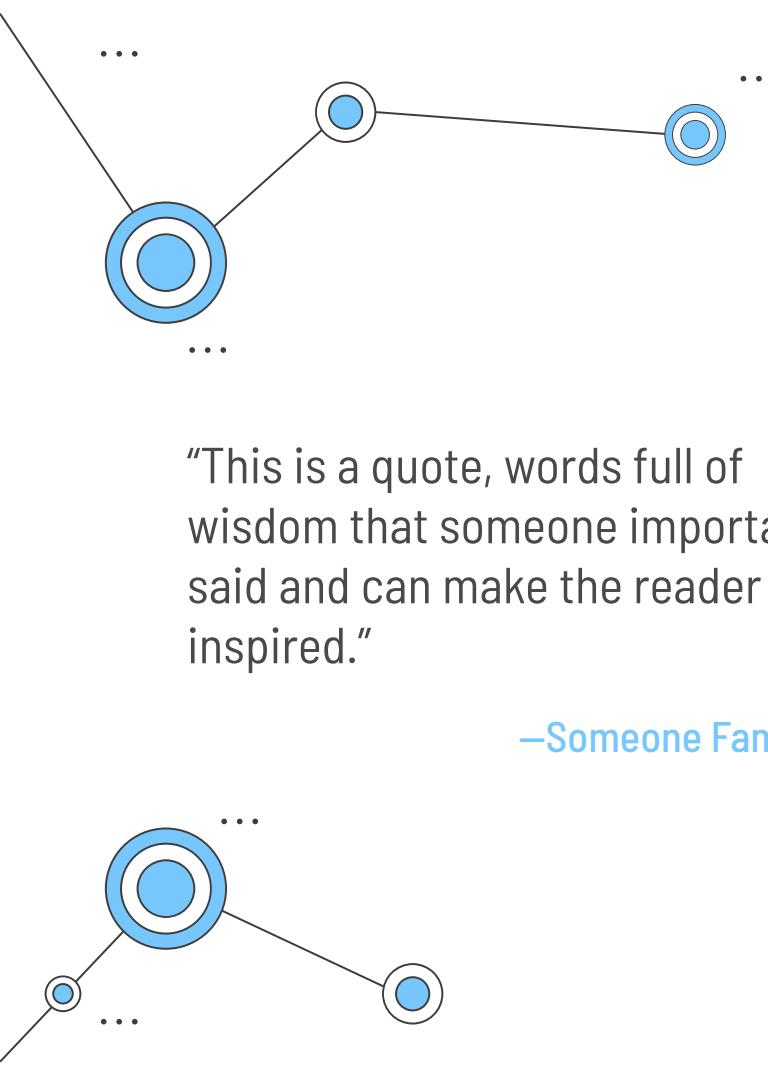
To modify this graph, click on it, follow the link, change the data and paste the resulting graph here



Gross Revenue in the Sector



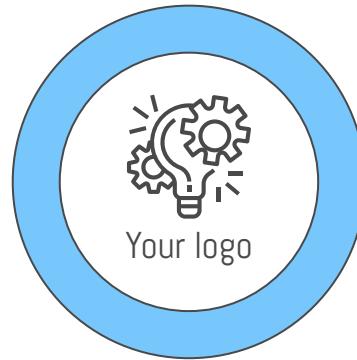
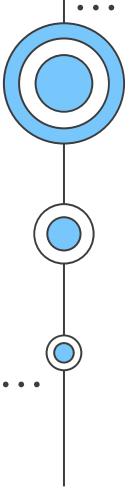
Sector Growth



“This is a quote, words full of wisdom that someone important said and can make the reader get inspired.”

—Someone Famous

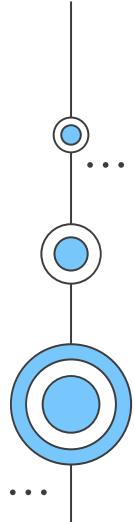




Our Company

Mercury is the closest planet to the sun and the smallest one in the solar system—it's only a bit larger than the Moon. The planet's name has nothing to do with the liquid metal, since Mercury was named after the Roman messenger god

...

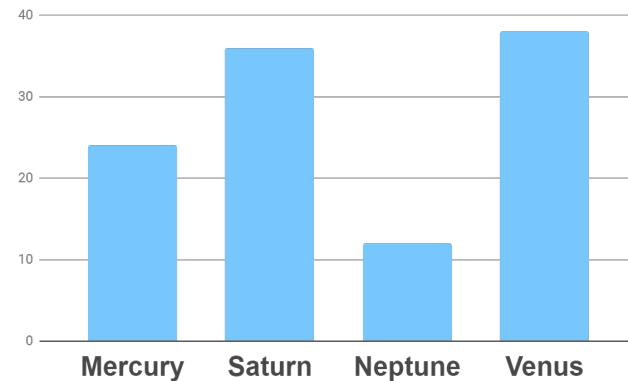


Analysis

Outreach



Top Rated Values



To modify this graph, click on it, follow the link, change the data and paste the new graph here



333,000.00

earths is the sun's mass

24h 37m 23s

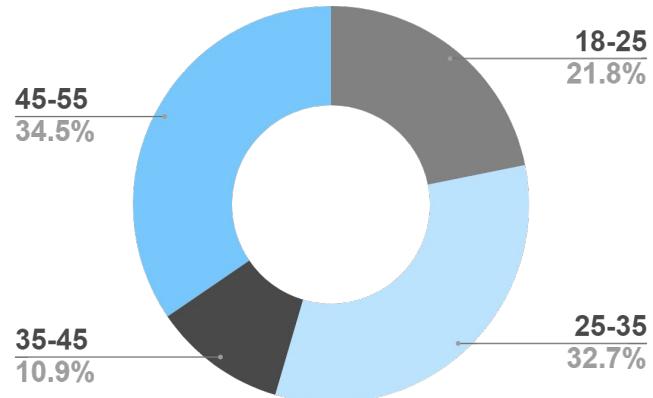
is Jupiter's rotation period

386,000 km

is the distance to the Moon

Target

Age

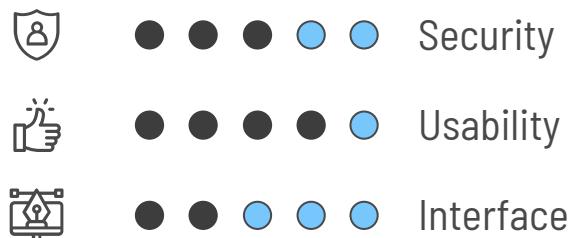


To modify this graph, click on it, follow the link,
change the data and paste the new graph here

Gender



Interests



Our Process

Mercury is the closest planet to the Sun

Venus is the second planet from the Sun

Day 1

Day 2

Day 3

Day 4

Jupiter is the biggest planet in the Solar System

Neptune is the farthest planet from the Sun

Whoa!

This could be the part of the presentation where you can introduce yourself, write your email...

Our Consultants

...
Jane Patterson

Mercury is the
closest planet to
the Sun

...
John James

Venus is the
second planet
from the Sun

...
Joe Doe

Neptune is the
farthest planet
from the Sun



Our Partners



Mercury

Mercury is the closest planet to the Sun



Jupiter

Jupiter is the biggest planet of them all



Neptune

Neptune is the farthest planet from the Sun



Saturn

Saturn is the ringed one and a gas giant



Mars

Despite being red, Mars is actually a cold place



Venus

Venus is the second planet from the Sun

Testimonials

Helena James

"Mercury is the closest planet to the Sun"

Jenna Doe

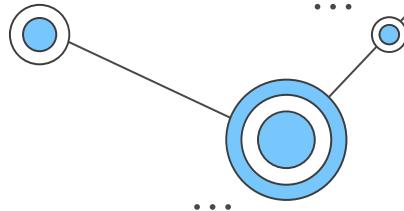
"Jupiter is the biggest planet of them all"

Mary Patterson

"Neptune is the farthest planet from the Sun"



Awards



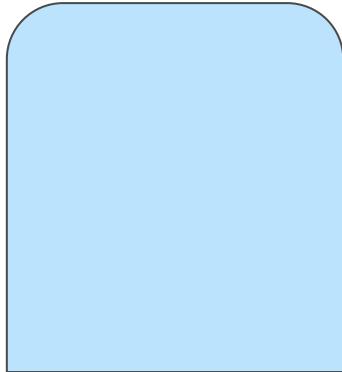
Mercury

It's the closest planet
to the Sun



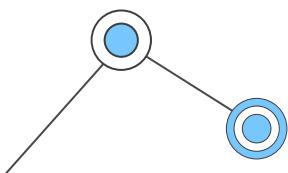
Venus

Venus is the second
planet from the Sun



Jupiter

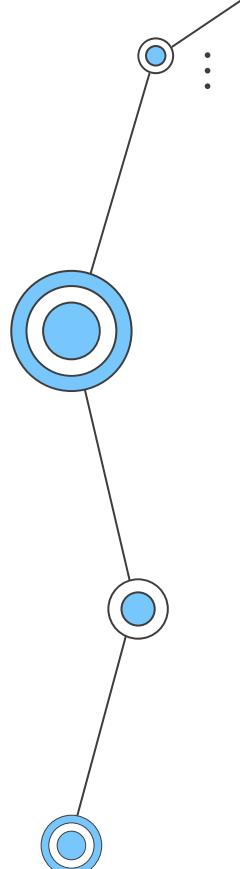
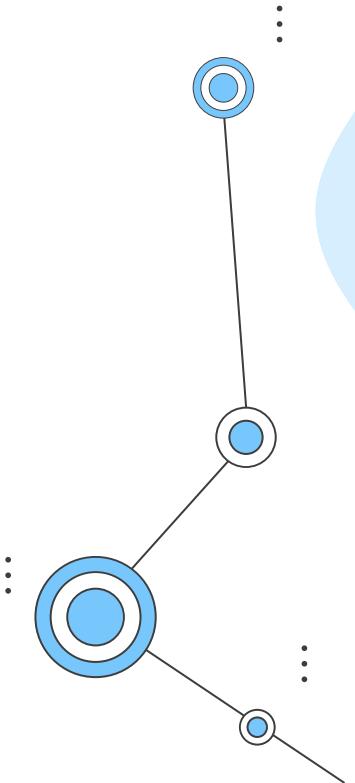
Jupiter is the biggest
planet of them all



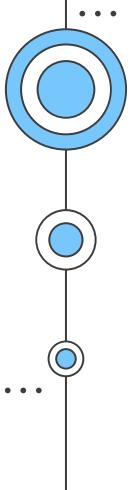


498,300,000

Big numbers catch your
audience's attention



Upcoming Goals



January

February

March

April

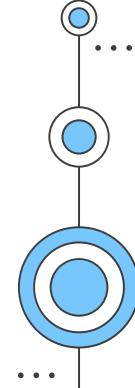
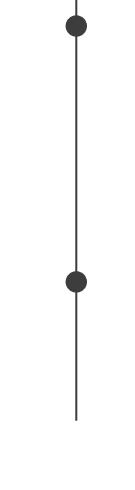
Goal 1



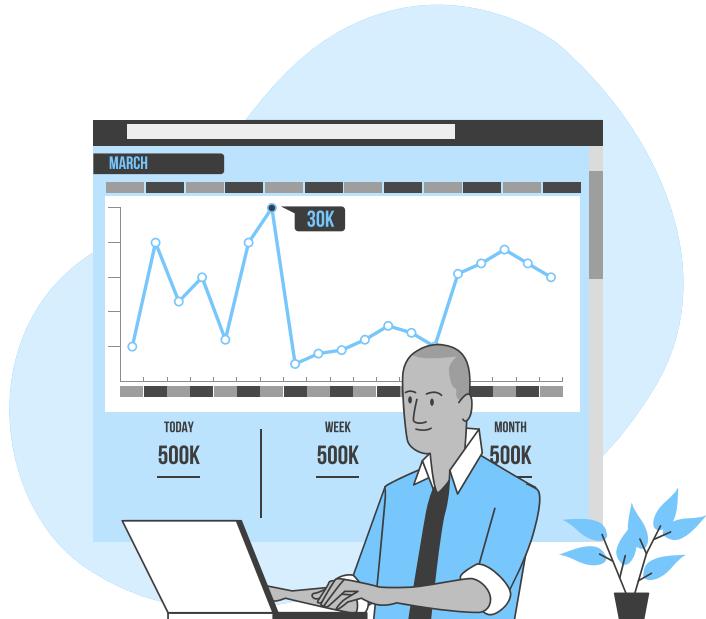
Phase 1



Phase 2



The Slide Title Goes Here!



Do you know what helps you make your point clear? Lists like this one:

- They're simple
- You can organize your ideas clearly
- You'll never forget to buy milk!

And the most important thing: the audience won't miss the point of your presentation

A Picture Is Worth a
Thousand Words



Infographics Make Your Idea Understandable...

Mercury

Mercury is the closest planet to the Sun

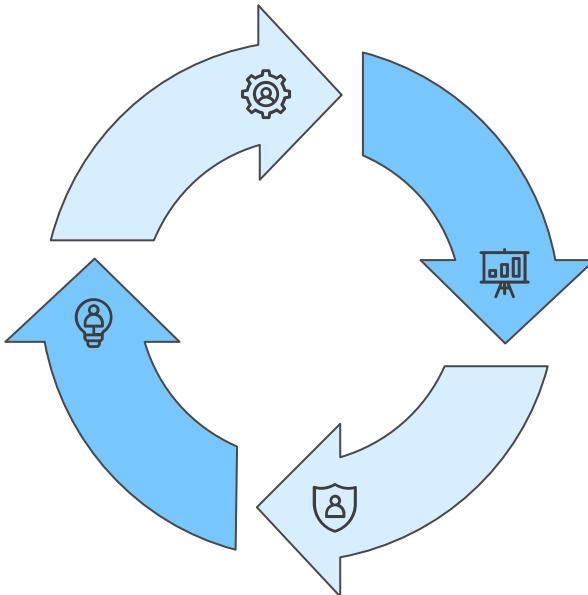
Jupiter

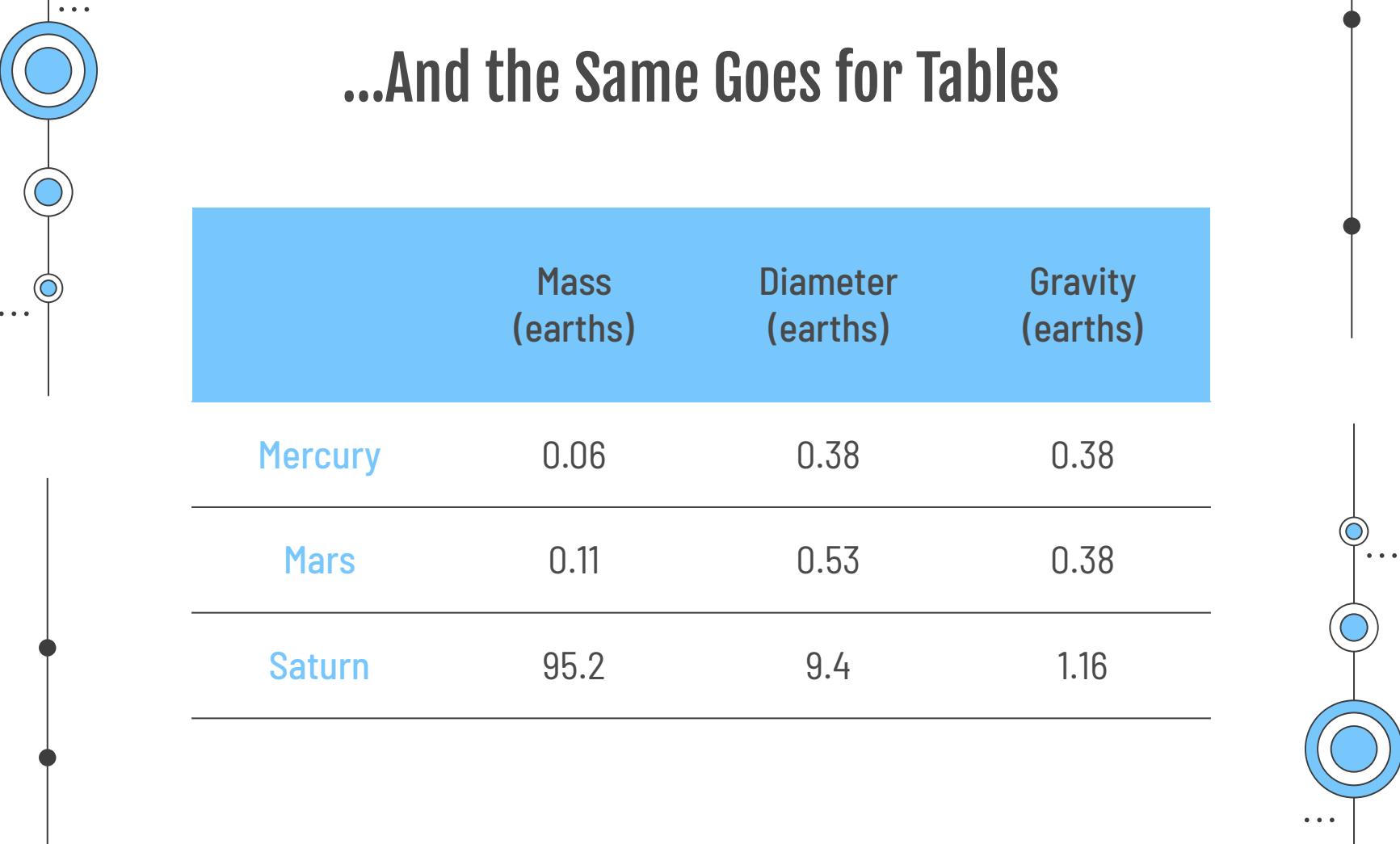
Jupiter is the biggest planet of them all

Neptune is the farthest planet from the Sun

Saturn

Saturn is the ringed one and a gas giant





...And the Same Goes for Tables

	Mass (earths)	Diameter (earths)	Gravity (earths)
Mercury	0.06	0.38	0.38
Mars	0.11	0.53	0.38
Saturn	95.2	9.4	1.16

Maybe You Need to Divide the Content

01

Neptune

Neptune is the farthest planet from the Sun and the fourth-largest in the Solar System

02

Jupiter

Jupiter is the biggest planet in the Solar System and the fourth-brightest one

Desktop Software

You can replace the image on the screen with your own work. Just delete this one, add yours and center it properly



Thanks!

Do you have any questions?

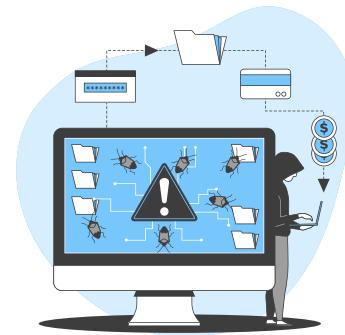
youremail@freepik.com
+91 620 421 838
yourcompany.com



CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#), infographics & images by [Freepik](#) and illustrations by [Stories](#)

Please keep this slide for attribution

Alternative Resources



Find more illustrations like these on [Stories by Freepik](#)

Resources

Vectors

- Programming
- Development
- Mission impossible
- Social networking
- Segmentation
- Site stats
- Responsive
- Meeting
- Analysis
- Cyber attack
- Secure server
- Control panel

Photos

- Close-up of woman using a laptop
- Beautiful young woman holding laptop on pink backdrop

Icons

- Project Management Icon

Find more illustrations like these on
[Stories by Freepik](#)

Instructions for use

In order to use this template, you must credit **Slidesgo** by keeping the **Thanks** slide.

You are allowed to:

- Modify this template.
- Use it for both personal and commercial projects.

You are not allowed to:

- Sublicense, sell or rent any of Slidesgo Content (or a modified version of Slidesgo Content).
- Distribute Slidesgo Content unless it has been expressly authorized by Slidesgo.
- Include Slidesgo Content in an online or offline database or file.
- Offer Slidesgo templates (or modified versions of Slidesgo templates) for download.
- Acquire the copyright of Slidesgo Content.

For more information about editing slides, please read our FAQs or visit Slidesgo School:

<https://slidesgo.com/faqs> and <https://slidesgo.com/slidesgo-school>

Fonts & colors used

This presentation has been made using the following fonts:

Fjalla One

(<https://fonts.google.com/specimen/Fjalla+One>)

Barlow Condensed

(<https://fonts.google.com/specimen/Barlow+Condensed>)

#494949

#477797

#77c6fc

#bbe3fe

#ddf1fe

Storyset

Create your Story with our illustrated concepts. Choose the style you like the most, edit its colors, pick the background and layers you want to be visible and bring them to life with the animator panel. It will boost your presentation for sure! Check out how it works.



Pana



Amico



Bro



Rafiki

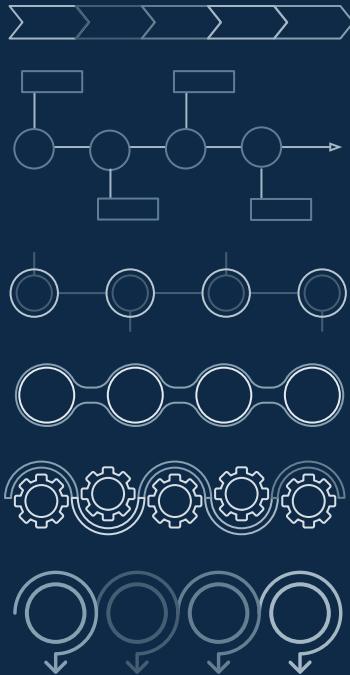
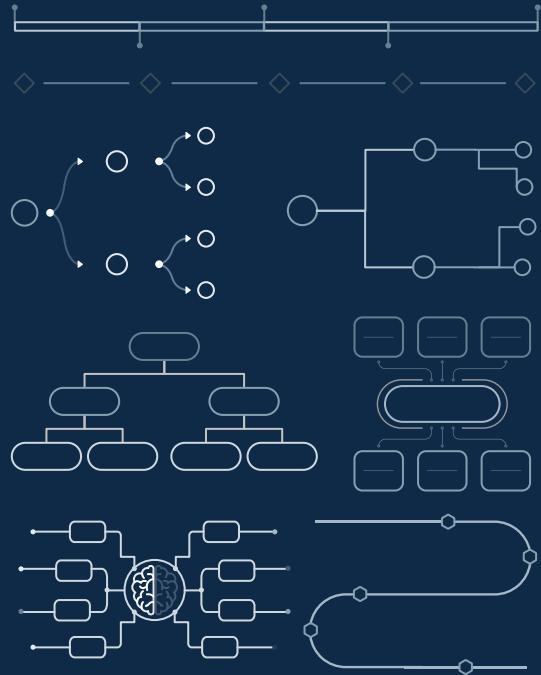
Use our editable graphic resources...

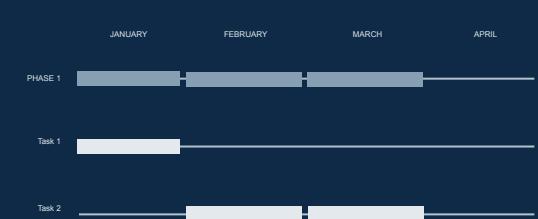
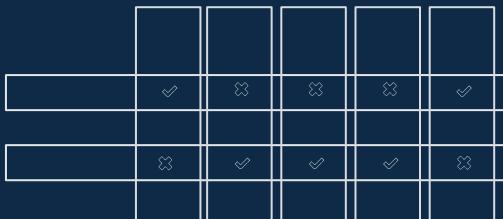
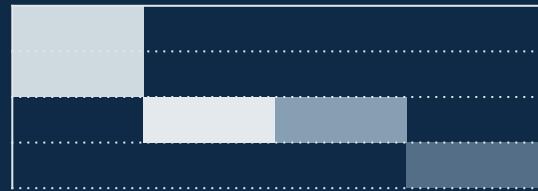
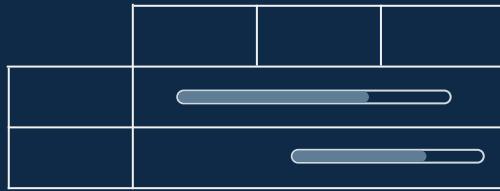
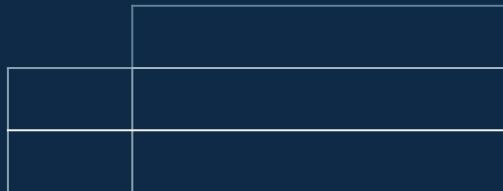
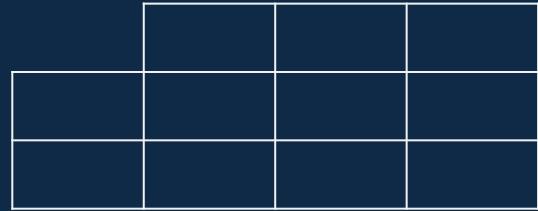
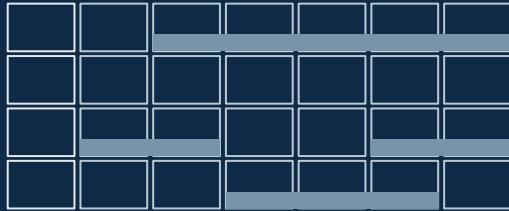
You can easily resize these resources without losing quality. To change the color, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want.

Group the resource again when you're done. You can also look for more infographics on Slidesgo.

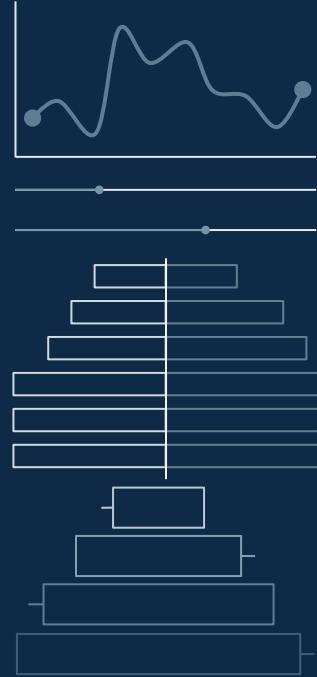
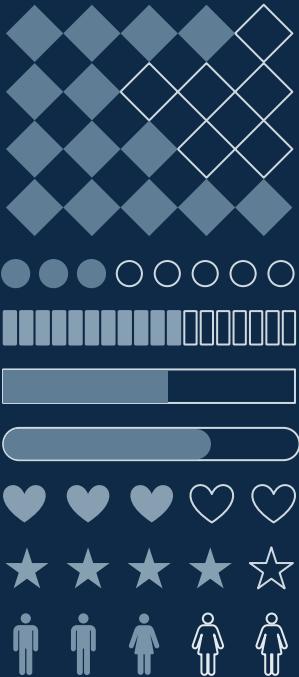
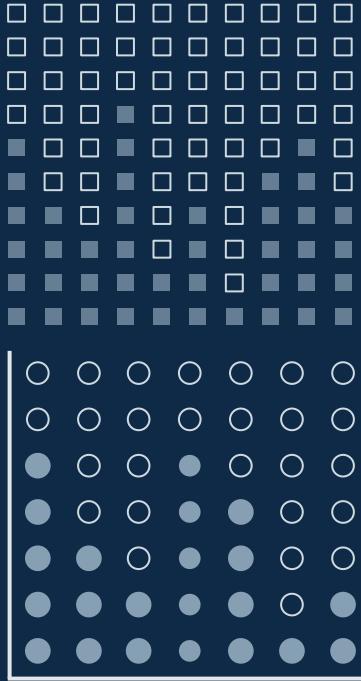












...and our sets of editable icons

You can resize these icons without losing quality.

You can change the stroke and fill color; just select the icon and click on the paint bucket/pen.

In Google Slides, you can also use Flaticon's extension, allowing you to customize and add even more icons.



Educational Icons



Medical Icons



Business Icons



Teamwork Icons



Help & Support Icons



Avatar Icons



Creative Process Icons



Performing Arts Icons



Nature Icons



SEO & Marketing Icons



